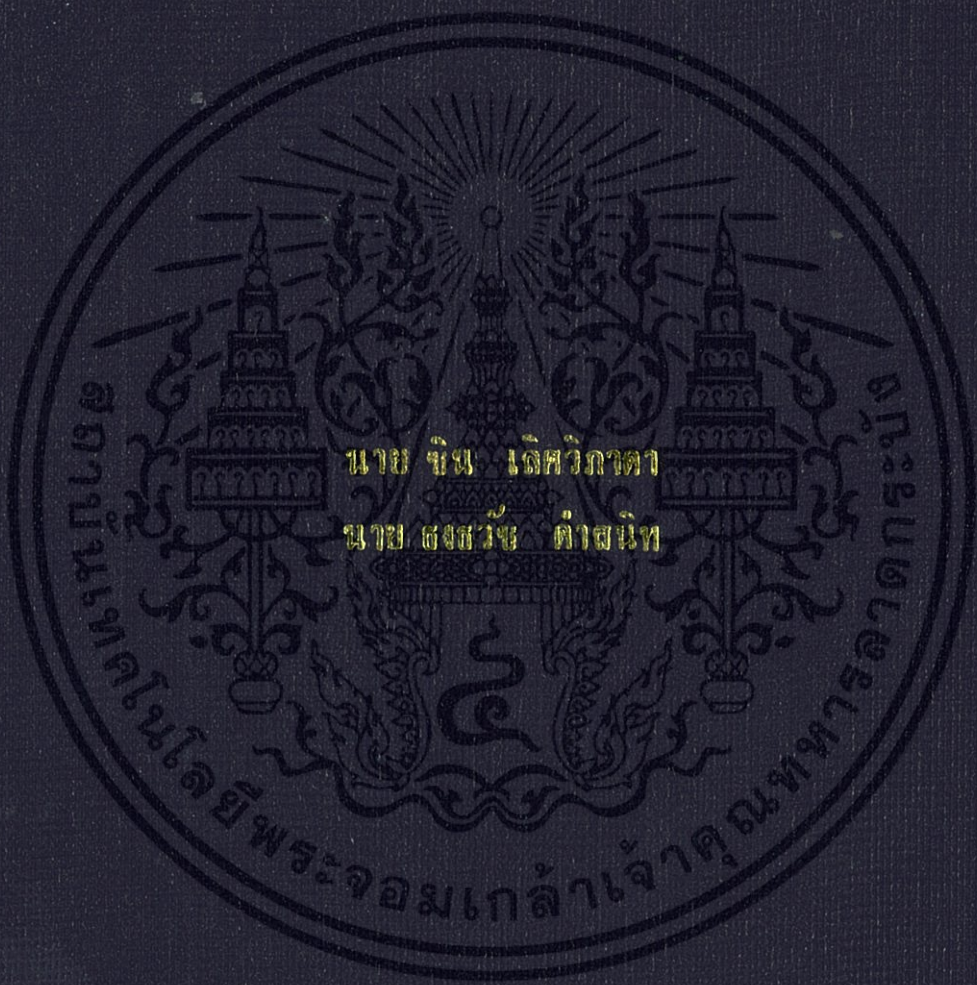
ระบบควบคุมรถหุ่นยนต์ด้วยสมาร์ทโฟน (Android) ผ่านระบบเครือข่ายไร้สาย

Designing and Implementing a WLAN Robot Car via Android Smartphone



นาย จิน เฉิดวิภาดา

นาย ธนวัช ต่ำผนิท

บัญญัติเป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต

สาขาวิชาวิทยาการคอมพิวเตอร์

คณะวิทยาศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2556

ระบบควบคุมรถหุ่นยนต์ด้วยสมาร์ทโฟน (Android) ผ่านระบบเครือข่ายไร้สาย

Designing and Implementing a WLAN Robot Car via Android Smartphone

นาย ชิน เลิศวิภาดา 53050968
นาย ธงธวัช ดำสนิท 53050993

ปัญหาพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต
สาขาวิทยาการคอมพิวเตอร์
คณะวิทยาศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2556

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Designing and Implementing a WLAN Robot Car via Android Smartphone

CHIN LERTVIPADA

THONGTAWAT DUMSANIT

A SPECIAL PROJECT SUBMITTED IN PARTIAL FULFILLMENT

OF THE REQUIRMENT FOR DEGREE OF BACHELOR OF SCIENCE

DEPARTMENT OF COMPUTER SCIENCE

FACULTY OF SCIENCE

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ACADEMIC YEAR 2013

หัวข้อปัญหาพิเศษ	ระบบควบคุมรถหุ่นยนต์ด้วยสมาร์ทโฟน (Android) ผ่านระบบเครือข่ายไร้สาย Designing and Implementing a WLAN Robot Car via Android Smartphone		
ชื่อนักศึกษา	นายชิน	เลิศวิภาดา	รหัสนักศึกษา 53050968
	นายธงธวัช	ดำสนิท	รหัสนักศึกษา 53050993
สาขาวิชา	วิทยาการคอมพิวเตอร์		
อาจารย์ที่ปรึกษา	รองศาสตราจารย์ธีรวัฒน์ ประกอบผล		

สาขาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง อนุมัติให้นับปัญหาพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร วิทยาศาสตรบัณฑิต สาขาวิชา วิทยาการคอมพิวเตอร์ ประจำปีการศึกษา 2556

คณะกรรมการสอบ	ลายมือชื่อ
ผู้ช่วยศาสตราจารย์ธีระ ศิริธีรากุล ประธานกรรมการ	
ดร.สันทิฎฐ์ นรบิน กรรมการ	
รองศาสตราจารย์ธีรวัฒน์ ประกอบผล กรรมการและอาจารย์ที่ปรึกษา	

ลิขสิทธิ์ของภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปัญหาพิเศษ	ระบบควบคุมรถหุ่นยนต์ด้วยสมาร์ทโฟน (Android) ผ่านระบบเครือข่ายไร้สาย Designing and Implementing a WLAN Robot Car via Android Smartphone
เสนอโดย	นายชิน เลิศวิภาดา รหัสนักศึกษา 53050968 นายธงธวัช ดำสนิท รหัสนักศึกษา 53050993
ปริญญา	วิทยาศาสตรบัณฑิต
สาขาวิชา	วิทยาการคอมพิวเตอร์
ปีการศึกษา	2556
อาจารย์ที่ปรึกษา	รองศาสตราจารย์ธีรวัฒน์ ประกอบผล

บทคัดย่อ

ในปัจจุบันเทคโนโลยีต่างๆได้ถูกพัฒนาขึ้นมาอย่างมากมาย เพื่อตอบสนองการใช้ชีวิตประจำวัน หรือเพื่อตอบสนองการใช้งานความต้องการเฉพาะทางของมนุษย์ เช่น เครื่องมือสื่อสาร คอมพิวเตอร์ส่วนบุคคล เครื่องจักรต่างๆ หรือเทคโนโลยีการรับส่งข้อมูล สิ่งเหล่านี้ล้วนแต่มีความทันสมัย มีความสะดวกสบาย ใช้งานง่าย และทำงานได้อย่างรวดเร็ว ดังที่จะเห็นได้จากโทรศัพท์มือถือในปัจจุบันที่เราเรียกกันว่าสมาร์ทโฟนที่ไม่ได้เป็นแค่เพียงโทรศัพท์อีกต่อไป โดยสมาร์ทโฟนมีฟังก์ชันการใช้งานและแอปพลิเคชันมากมายให้เลือกใช้งาน เช่น เทคโนโลยี Wi-Fi สำหรับการเชื่อมต่อกับเครือข่ายหรืออินเทอร์เน็ตสำหรับการรับส่งข้อมูล เทคโนโลยีระบบปฏิบัติการแอนดรอยด์ของบริษัทกูเกิ้ลซึ่งเป็นระบบปฏิบัติการของสมาร์ทโฟนที่มีการใช้งานอย่างกว้างขวาง เทคโนโลยีภาษาจาวาที่ใช้สำหรับการเขียนแอปพลิเคชันบนสมาร์ทโฟนระบบปฏิบัติการแอนดรอยด์ เทคโนโลยีไมโครคอนโทรลเลอร์สำหรับการทำงานระดับฮาร์ดแวร์ ผู้จัดทำจึงมีความคิดริเริ่มในการนำเทคโนโลยีเหล่านี้มาพัฒนาเข้าด้วยกัน เพื่อสร้างแอปพลิเคชันบนสมาร์ทโฟนระบบปฏิบัติการแอนดรอยด์มาทำการควบคุมไมโครคอนโทรลเลอร์เพื่อให้ไมโครคอนโทรลเลอร์สั่งการการทำงานของชุดอุปกรณ์ที่ประกอบขึ้นเป็นรถหุ่นยนต์ ในทิศทางการเคลื่อนที่เดินหน้า ถอยหลัง เลี้ยวซ้าย เลี้ยวขวา ผ่านเทคโนโลยี Wi-Fi โดยการควบคุมบนสมาร์ทโฟนมีรูปแบบต่างๆ ที่หลากหลาย ซึ่งจะมีการส่งสัญญาณ Wi-Fi ไปยังบอร์ดไมโครคอนโทรลเลอร์ เพื่อให้โปรแกรมบนบอร์ดไมโครคอนโทรเลอร์ที่ถูกเขียนด้วยภาษาซีทำการประมวลผลแล้วควบคุมการทำงานของมอเตอร์เพื่อขับเคลื่อนรถหุ่นยนต์ และใช้งานเซนเซอร์ต่างๆ เพื่อตรวจสอบค่าและส่งกลับค่าของเซนเซอร์ไปยังสมาร์ทโฟน เพื่อให้สามารถนำไปใช้งานได้ทันที หรือเพื่อนำไปต่อยอดสำหรับใช้งานในพื้นที่ปฏิบัติงานเฉพาะทางต่างๆ เพื่อลดความเสี่ยงที่อาจจะเกิดขึ้นในขณะปฏิบัติงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Title	Designing and Implementing a WLAN Robot Car via Android Smartphone
Student	Mr.Chin Lertvipada 53050968
	Mr.ThongtawatDumsanit 53050993
Degree	Bachelor of Science
Major Program	Computer Science
Academic Year	2013
Advisor	Associate Professor Teerawat Prakobphon

ABSTRACT

In this day, a number of technologies have been developed tremendously to respond people's daily life or people's need; For instance, communication technology, personal computer, machine or data transmission technology. These technologies are up-to-date, comfortable, easy to use and quick at processing. As mobile phone that we call smartphone these days have many functions and applications; For instance, wifi technology for communicate network or internet for data transmission, android operating system of google inc. which is broadly usable, java language technology for build android application, microcontroller technology for task on hardware. All the technologies mentioned above have been gathered together so as to be able to success fully design android application to control the microcontroller that command robot car's kit to movement, forward, backward, left and right via wifi technology by many type for control on smartphone. Signals are sent to the wifi microcontroller board; In order for the program in board to be written with C language. Such process will accordingly control the motor which is connected to the robot car and be usable sensors to examine values and return to smartphone. This project can using for working immediately or adapt to using in dangerous task for decreasing risks.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ในการจัดทำปัญหาพิเศษเรื่อง ระบบควบคุมรถหุ่นยนต์ด้วยสมาร์ทโฟน (Android) ผ่านระบบเครือข่ายไร้สาย นี้สำเร็จลุล่วงได้ด้วยดี คณะผู้จัดทำขอขอบพระคุณ รองศาสตราจารย์ธีรวัฒน์ ประกอบผล อาจารย์ที่ปรึกษาปัญหาพิเศษ ที่ได้เสียสละเวลาให้คำปรึกษา คำแนะนำ ตรวจแก้ไขปัญหาต่างๆ และเอาใจใส่ตลอดระยะเวลาทั้งหมดที่ทำปัญหาพิเศษนี้

ขอขอบคุณ สาขาวิชาวิทยาการคอมพิวเตอร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง สำหรับสิ่งอำนวยความสะดวกมากมายที่มีให้บริการ ทั้งห้องทำงาน และอินเตอร์เน็ตเพื่อการค้นคว้าข้อมูลต่างๆ ที่จำเป็นสำหรับการทำปัญหาพิเศษนี้

ขอขอบพระคุณคณาจารย์ สาขาวิทยาการคอมพิวเตอร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบังทุกท่าน ที่ได้ประสิทธิ์ประสาทวิชาให้แก่คณะผู้จัดทำ

สุดท้ายนี้ต้องขอกราบขอบพระคุณบุคคลที่สำคัญที่สุดที่ทำให้มีวันนี้คือ บิดา มารดา อันเป็นที่เคารพรักของคณะผู้จัดทำ ซึ่งได้เลี้ยงดูมาเป็นอย่างดี พร้อมทั้งให้โอกาสการศึกษาอย่างเต็มที่ และยังให้กำลังใจ เอาใจใส่เสมอมา ในทุกๆ ด้านอันหาที่เปรียบมิได้ รวมถึงขอบคุณพี่ๆ เพื่อนๆ และน้องๆ ที่ให้กำลังใจ คำปรึกษา และความช่วยเหลือไว้ ณ ที่นี้

คุณค่าและประโยชน์อันพึงมาจากปัญหาพิเศษฉบับนี้ คณะผู้จัดทำขอมอบแด่ผู้มีพระคุณทุกท่าน

คณะผู้จัดทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

บทที่	หน้า
บทคัดย่อภาษาไทย	i
บทคัดย่อภาษาอังกฤษ	ii
กิตติกรรมประกาศ	iii
สารบัญ	iv
สารบัญภาพ	viii
สารบัญตาราง	xiv
บทที่ 1 บทนำ	1
1.1 ความสำคัญและที่มาของปัญหา	1
1.2 วัตถุประสงค์	2
1.3 ขอบเขตของปัญหาพิเศษ	2
1.4 ขั้นตอนการดำเนินงาน	2
1.5 ประโยชน์ที่คาดว่าจะได้รับ	3
1.6 อุปกรณ์ที่ต้องใช้	3
บทที่ 2 ทฤษฎีและหลักการที่เกี่ยวข้อง	4
2.1 ระบบสื่อสารไร้สาย	4
2.1.1 เทคโนโลยี Wi-Fi	4
2.1.2 โปรโตคอล	6
2.2 Android	15
2.2.1 Google Android	16
2.2.2 คุณลักษณะพิเศษของระบบปฏิบัติการแอนดรอยด์	16
2.2.3 ชนิดของโปรแกรมประยุกต์	18
2.2.4 Activity Life Cycle	18
2.3 Java	21
2.3.1 ข้อดีของภาษา Java	21
2.3.2 ข้อเสียของภาษา Java	22
2.3.2 เบื้องหลังการทำงานของโปรแกรมภาษา Java	22
2.3.4 หลักการเขียนโปรแกรมด้วยภาษา Java	23

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

บทที่	หน้า
2.3.5 Object-Oriented Programming	23
2.4 Java Develop Kit	24
2.5 Software Develop Kit	24
2.6 ชุดหุ่นยนต์ POP-BOT XT Lite	25
2.6.1 ไมโครคอลโทรลเลอร์	25
2.6.2 หุ่นยนต์ POP-BOT XT Lite	28
2.7 ชุดอุปกรณ์ RN-XV WiFly Module – Wire Antenna	35
2.7.1 คุณสมบัติของ RN-XV WiFly Module – Wire Antenna	36
2.7.2 Configuration	36
2.8 โปรแกรม Arduino	40
2.8.1 โครงสร้างโปรแกรมของ Arduino	40
2.8.2 ส่วนประกอบของหน้าจอโปรแกรม Arduino 1.0	42
2.9 บทความที่เกียวข้อง	44
2.9.1 Designing and Implementing a WLAN Robot Car	44
2.9.2 System Direction Control for Controller Car via Bluetooth on Smartphone	45
บทที่ 3 การวิเคราะห์และออกแบบระบบ	48
3.1 โครงสร้างและภาพรวมระบบ	48
3.2 การออกแบบระบบ	50
3.2.1 แผนภาพการทำงานของผู้ใช้ระบบ	50
3.2.2 แผนภาพขั้นตอนการทำงานของระบบ	52
3.2.3 แผนภาพกิจกรรมของระบบงาน	55
3.3 สิ่งที่จำเป็นในการพัฒนาระบบ	56
3.3.1 Eclipse Development Tools and Java Development Kit	56
3.3.2 Android Development Tools (ADT) Plugin for Eclipse	60
3.3.3 Android Software Development Kit	66
3.3.4 Android Virtual Device Manager (Emulator)	74

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

บทที่	หน้า
3.4 ขั้นตอนการประกอบรถหุ่นยนต์ POP-BOT XT Lite	79
3.5 ออกแบบส่วนติดต่อกับผู้ใช้งาน	86
3.5.1 การออกแบบส่วนของแอปพลิเคชั่นบนสมาร์ทโฟน	86
3.6 เซนเซอร์ในแอนดรอยด์	97
3.6.1 เซนเซอร์วัดระยะทาง	97
3.6.2 เซนเซอร์ตรวจจับแสงสว่าง	98
3.7 การเรียกใช้เซนเซอร์วัดการเอียง	99
3.7.1 แผนภาพขั้นตอนการทำงานของการควบคุมด้วยการเอียงสมาร์ทโฟน	101
3.7.2 การเรียกใช้งานเซนเซอร์วัดการเอียง	101
3.8 หน้าจอแสดงผลบนไมโครคอนโทรลเลอร์	104
3.9 การตั้งค่าอุปกรณ์ RN-XV WiFly Module	108
บทที่ 4 การทดสอบและผลการดำเนินงาน	109
4.1 การเอ็กพอร์ต (Export) แอปพลิเคชั่นจากโปรแกรม Eclipse เป็นไฟล์ .apk	109
4.1.1 ขั้นตอนการสร้างคีย์สโตร์และเอ็กพอร์ตไฟล์ .apk	109
4.1.2 ขั้นตอนการเอ็กพอร์ตไฟล์ .apk กรณีที่มีคีย์สโตร์แล้ว	115
4.2 การทดลองบนสมาร์ทโฟนระบบปฏิบัติการแอนดรอยด์	118
4.2.1 การติดตั้งแอปพลิเคชัน	118
4.2.2 การทดสอบใช้งานแอปพลิเคชัน	121
4.3 การทดสอบบนรถหุ่นยนต์	128
4.3.1 การทดสอบการตั้งค่าอุปกรณ์รับส่งสัญญาณ Wi-Fi	128
4.3.2 การติดตั้งโปรแกรมลงสู่ไมโครคอนโทรลเลอร์	130
บทที่ 5 สรุปผลการดำเนินงาน และข้อเสนอแนะ	136
5.1 สรุปผลการดำเนินงาน	136
5.1.1 สรุปผลการดำเนินงาน	136
5.1.2 การพัฒนาระบบ	136
5.1.3 ประสิทธิภาพและคุณสมบัติของระบบ	137

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

บทที่	หน้า
5.1.4 ข้อจำกัดของโปรแกรม	137
5.2 ข้อเสนอแนะ	138
รายการอ้างอิง	139

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ

ภาพที่	หน้า
2.1 ขั้นตอนการ Encapsulation และDemultiplexing	7
2.2 โครงสร้าง TCP/IP	8
2.3 IP Header	9
2.4 ICMP Header	11
2.5 UDP Header	12
2.6 TCP Header	13
2.7 Activity Life Cycle	19
2.8 Activity Life Cycle Flowchart	19
2.9 ขั้นตอนการใช้งาน Source Code ภาษาจาวา	22
2.10 โครงสร้างพื้นฐานของไมโครคอนโทรลเลอร์	25
2.11 โครงสร้างและส่วนประกอบหลักของซีพียู	26
2.12 ลักษณะของชุดหุ่นยนต์ POP-BOT XT Lite	28
2.13 แสดงส่วนประกอบของแผงวงจรควบคุม POP-BOT XT	30
2.14 วงจรสมบูรณ์ของแผงวงจรควบคุม POP-BOT XT	31
2.15 มอเตอร์ไฟตรงพร้อมชุดเฟืองขับ	32
2.16 รูปร่างและวงจรของแผงวงจรไฟแสดงผล ZX-LED	33
2.17 รูปร่างและวงจรของแผงวงจรสวิตช์ ZX-01	33
2.18 รูปร่างและวงจรของแผงวงจรตรวจจับแสงสะท้อนอินฟราเรด ZX-03	34
2.19 สาย JST3AA-8	34
2.20 สาย USB-miniB	35
2.21 RN-XV WiFly Module - Wire Antenna	35
2.22 แสดงส่วนประกอบของโปรแกรม Arduino1.0	43
2.23 รถหุ่นยนต์ของโครงงาน Designing and Implementing a WLAN Robot Car	44
2.24 ET-WiFly GSX	44
2.25 หน้าจอส่วนติดต่อกับผู้ใช้เพื่อควบคุมรถหุ่นยนต์	45
2.26 รถหุ่นยนต์ของโครงงาน System Direction Control for Controller Car via Bluetooth on Smartphone	45

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ (ต่อ)

ภาพที่	หน้า
2.27 ZX-Bluetooth	46
2.28 โทรศัพท์มือถือ Nokia	46
2.29 หน้าจอส่วนติดต่อกับผู้ใช้งาน	47
3.1 ระบบการควบคุมรถหุ่นยนต์	49
3.2 Use case Diagram ระบบควบคุมรถหุ่นยนต์ด้วยสมาร์ทโฟนผ่านเครือข่ายไร้สาย	51
3.3 Flow Chart ของระบบควบคุมรถหุ่นยนต์ด้วยสมาร์ทโฟนผ่านเครือข่ายไร้สาย	52
3.4 Flow Chart ของสมาร์ทโฟนระบบปฏิบัติการแอนดรอยด์	53
3.5 Flow Chart ของไมโครคอนโทรลเลอร์	54
3.6 แผนภาพกิจกรรมของระบบ	55
3.7 โลโก้โปรแกรม Eclipse	56
3.8 การดาวน์โหลดโปรแกรม Eclipse Classic	57
3.9 ทำการแตกไฟล์ .zip ไว้ที่ C:\eclipse	57
3.10 เลือกดาวน์โหลดไฟล์ JDK	58
3.11 เลือกดาวน์โหลดให้ถูกต้องกับระบบปฏิบัติการของคอมพิวเตอร์ที่ต้องการติดตั้งโปรแกรม	59
3.12 ทำการติดตั้ง JDK	59
3.13 หน้าจอเสร็จสิ้นการติดตั้ง JDK	60
3.14 คลิกที่ Installing the Eclipse Plugin	61
3.15 เลือกเมนู Help > Install New Software	61
3.16 คลิกที่ปุ่ม Add	62
3.17 ระบุ Name และ Location ให้กับปลั๊กอิน	62
3.18 คลิกเลือก Developer Tools และ Next	63
3.19 แสดงรายละเอียดส่วนประกอบของปลั๊กอินที่จะทำการติดตั้ง	64
3.20 หน้าจอแสดงเงื่อนไขข้อตกลงการใช้งาน	65
3.21 โปรแกรม Eclipse จะทำการดาวน์โหลดและติดตั้งปลั๊กอิน	65
3.22 ทำการเริ่มโปรแกรม Eclipse ใหม่	66
3.23 Download the SDK for Windows	66

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ (ต่อ)

ภาพที่	หน้า
3.24 ทำการติดตั้ง Android Software Development Kit ที่ดาวน์โหลดมา	67
3.25 กำหนดพื้นที่ติดตั้งไว้ที่ C:\Android\android-sdk\	67
3.26 แสดงความคืบหน้าในการติดตั้ง	68
3.27 เลือก Start SDK Manager แล้วกดปุ่ม Finish	68
3.28 ทำการดาวน์โหลด Tools และ Android Version	69
3.29 เลือก Accept และกดปุ่ม Install เพื่อติดตั้ง	69
3.30 แสดงความคืบหน้าการดาวน์โหลด	70
3.31 แสดงข้อความแจ้งว่าการติดตั้งเสร็จสิ้นแล้ว	70
3.32 คลิกที่ Open the Android SDK Manager	71
3.33 กล่องข้อความเตือนเมื่อเกิดข้อผิดพลาด	71
3.34 เลือก Window > Preferences	72
3.35 กำหนด path ที่เราได้ติดตั้ง Android SDK ไว้	73
3.36 การทำงานที่ถูกต้องของโปรแกรม	74
3.37 ตัวอย่างหน้าจอของ Emulator	75
3.38 คลิกที่ Open the Android Virtual Device Manager	75
3.39 กดปุ่ม New เพื่อสร้าง Emulator	76
3.40 หน้าจอกำหนดค่าของ Emulator	76
3.41 เลือกรายการการจำลอง Emulator ที่ต้องการ	77
3.42 หน้าจอแสดงรายละเอียด Emulator ให้กด Launch เพื่อเริ่มการจำลอง	77
3.43 หน้าจอของ Emulator เมื่อพร้อมใช้งาน	78
3.44 ยึดมอเตอร์เข้ากับกระบะถ่าน	79
3.45 ติดเสารองโลหะ	79
3.46 ติดล้อเข้ากับมอเตอร์	80
3.47 ชุดมอเตอร์ยึดเข้ากับฐานพลาสติกของหุ่นยนต์	80
3.48 แผงวงจรตรวจจับแสงสะท้อนอินฟราเรด ZX-03	81
3.49 ต่อแผงวงจรตรวจจับแสงสะท้อน	81
3.50 ต่อแท่งต่อ 5 รู	82

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ (ต่อ)

ภาพที่	หน้า
3.51 ต่อวงจรสวิตช์	82
3.52 ต่อแผงวงจรสวิตช์กับฐานของหุ่นยนต์	83
3.53 ต่อแผงวงจร POP-XT วางลงกล่องรองกระบะถ่าน	83
3.54 ต่อสายมอเตอร์	84
3.55 ต่อสายของวงจรเซนเซอร์ต่างๆ	84
3.56 POP-BOT XT เสร็จสมบูรณ์	85
3.57 หน้าเมน	86
3.58 หน้าการควบคุมด้วยปุ่มกด	87
3.59 ส่วนของการแสดงผลค่าของเซนเซอร์	88
3.60 ส่วนของการควบคุมรถหุ่นยนต์ในหน้าการควบคุมด้วยปุ่มกด	89
3.61 ส่วนของการเปลี่ยนรูปแบบการควบคุมในหน้าการควบคุมด้วยปุ่มกด	90
3.62 หน้าการควบคุมด้วยการเอียงสมาร์ทโฟน	91
3.63 การแสดงผลเมื่อเอียงสมาร์ทโฟนไปด้านหน้าและด้านขวาตามลำดับ	92
3.64 ส่วนของการเปลี่ยนรูปแบบการควบคุมในหน้าการควบคุมด้วยการเอียงสมาร์ทโฟน	93
3.65 หน้าการควบคุมด้วยคันบังคับ	94
3.66 ส่วนของพื้นที่คันบังคับสำหรับใช้ควบคุม	95
3.67 ส่วนของการเปลี่ยนรูปแบบการควบคุมในหน้าการควบคุมด้วยคันบังคับ	96
3.68 Proximity Sensor	97
3.69 Light Sensor	98
3.70 แกนสำหรับอุปกรณ์ที่มีขนาด 7 นิ้วขึ้นไป	99
3.71 แกนสำหรับอุปกรณ์ที่มีขนาด 7 นิ้วลงมา	99
3.72 แสดงค่าของแต่ละแกนเมื่อเอียงเครื่องไปในทิศทางต่างๆ	100
3.73 Flow chart ของการควบคุมด้วยการเอียงสมาร์ทโฟน	101
3.74 ไมโครคอนโทรเลอร์กำลังเชื่อมต่อกับ Wi-Fi Hotspot	104
3.75 ไมโครคอนโทรเลอร์เชื่อมต่อกับ Wi-Fi Hotspot ไม่สำเร็จ	105
3.76 ไมโครคอนโทรเลอร์เชื่อมต่อกับ Wi-Fi Hotspot สำเร็จ	106
3.77 สมาร์ทโฟนเชื่อมต่อกับไมโครคอนโทรลเลอร์สำเร็จ	107
3.78 โปรแกรม Tera Term	108

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ (ต่อ)

ภาพที่	หน้า
4.1 เลือกโปรเจคที่จะทำการเอ็กพอร์ต	109
4.2 เลือกประเภทของการเอ็กพอร์ต	110
4.3 หน้าจอสำหรับเลือกโปรเจคที่ต้องการจะเอ็กพอร์ต	110
4.4 หน้าจอสำหรับระบุไฟล์คีย์สโตร์	111
4.5 ตั้งชื่อให้กับไฟล์คีย์สโตร์	111
4.6 กำหนดรหัสผ่านสำหรับคีย์สโตร์	112
4.7 กำหนดค่าให้กับคีย์สโตร์	113
4.8 กำหนดพื้นที่สำหรับจัดเก็บไฟล์ .apk	113
4.9 เลือกพื้นที่จัดเก็บและตั้งชื่อไฟล์ .apk	114
4.10 กดปุ่ม Finish เพื่อทำการเอ็กพอร์ตไฟล์ .apk	114
4.11 ไฟล์คีย์สโตร์และไฟล์ .apk ที่ทำการเอ็กพอร์ต	115
4.12 เลือก Use existing keystore เพื่อเลือกไฟล์คีย์สโตร์ที่มีอยู่แล้ว	115
4.13 เลือกไฟล์คีย์สโตร์ที่มีอยู่	116
4.14 ระบุรหัสผ่านของคีย์สโตร์	116
4.15 เลือก Alias และระบุรหัสผ่าน	117
4.16 แสดงรายละเอียดวันหมดอายุของคีย์สโตร์และ Certificate fingerprints	118
4.17 การเข้าสู่เมนูตั้งค่าระบบป้องกัน	119
4.18 การตั้งค่าเพื่อให้สมาร์ทโฟนสามารถติดตั้งแอพลิเคชันจากแหล่งที่ไม่รู้จักได้	119
4.19 ไฟล์ .apk ของแอพพลิเคชัน	120
4.20 แสดงการขออนุญาตเข้าถึงระบบต่าง ๆ ในสมาร์ทโฟนก่อนทำการติดตั้ง	120
4.21 รอการติดตั้งแอพลิเคชัน	121
4.22 การตั้งค่า Wi-Fi Hotspot บนสมาร์ทโฟน	122
4.23 การทดสอบการทำงานของหน้าเมน	122
4.24 การทดสอบเมื่อการเชื่อมต่อสำเร็จ	123
4.25 การทดสอบเมื่อการเชื่อมต่อไม่สำเร็จ	123
4.26 การทดสอบเมื่อผู้ใช้งานระบุ IP Address ไม่ถูกต้องตามหลัก IPv4	124

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ (ต่อ)

ภาพที่	หน้า
4.27 การทดสอบเมื่อผู้ใช้งานไม่ได้ทำการระบุหมายเลข PORT	124
4.28 การทดสอบด้วยการใช้ปุ่มควบคุม	125
4.29 การทดสอบด้วยการเอียงสมาร์ทโฟน	126
4.30 การทดสอบด้วยการใช้คันบังคับควบคุม	126
4.31 การทดสอบการใช้งานเซนเซอร์วัดแสงและเซนเซอร์ตรวจจับการชน	127
4.32 ตั้งค่าการเชื่อมต่ออุปกรณ์ RN-XV WiFly Module กับคอมพิวเตอร์	128
4.33 การเข้า Command Mode	129
4.34 ตั้งค่าลงบน RN-XV WiFly Module	129
4.35 หน้าแรกโปรแกรม Arduino	131
4.36 ขั้นตอนการเลือกเบอร์ของไมโครคอนโทรลเลอร์	132
4.37 การกำหนดพอร์ตอนุกรมที่ใช้ในการติดต่อกับโมดูล	132
4.38 ปุ่มคอมไพล์โปรแกรม	133
4.39 การคอมไพล์เสร็จสมบูรณ์	134
4.40 การต่อสาย USB	134
4.41 อัพโหลดโปรแกรมลงบอร์ดไมโครคอนโทรลเลอร์	135

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่	หน้า
2.1 Flag ทั้งหมด 6 บิต	14
3.1 รายละเอียดค่าของการเอียง	100
3.2 แสดงค่าความเร็วที่เปลี่ยนแปลงตามการเอียงของสมาร์ทโฟน	103

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความสำคัญและที่มาของปัญหา

เนื่องจากในปัจจุบันเทคโนโลยีที่เกี่ยวข้องกับอุปกรณ์อิเล็กทรอนิกส์และหุ่นยนต์ได้มีบทบาทและความสำคัญมากขึ้นทั้งในชีวิตประจำวัน ในการทำงานและในการปฏิบัติการต่างๆ ที่ไม่จำเป็นต้องพึ่งพามนุษย์ ซึ่งเทคโนโลยีของอุปกรณ์อิเล็กทรอนิกส์และหุ่นยนต์นี้ได้ก้าวไปไกลอย่างมากเมื่อเทียบกับในอดีต และมีการพัฒนาให้มนุษย์สามารถเข้าถึงได้ง่ายและมากยิ่งขึ้น ตัวอย่างเช่น สมาร์ทโฟน (Smartphone) หรือ แท็บเล็ต (Tablet) ซึ่งอุปกรณ์เหล่านี้มีความสามารถที่ใกล้เคียงกับคอมพิวเตอร์ เทคโนโลยีการพัฒนาอุปกรณ์อิเล็กทรอนิกส์และหุ่นยนต์มีหลากหลายรูปแบบในปัจจุบัน ไม่ว่าจะเป็นเครื่องจักรในโรงงานอุตสาหกรรม ระบบการเปิดและปิดเครื่องใช้ไฟฟ้าหรืออุปกรณ์อิเล็กทรอนิกส์ตามบ้านเรือนโดยสั่งการด้วยสมาร์ทโฟนผ่านระบบเครือข่ายไร้สาย การสั่งการหุ่นยนต์ต่างๆ ด้วยคอมพิวเตอร์หรืออุปกรณ์ควบคุมเฉพาะตัวเช่น หุ่นยนต์กู้ภัย หุ่นยนต์เตะฟุตบอล รถหุ่นยนต์สำรวจ ซึ่งการควบคุมหุ่นยนต์และอุปกรณ์อิเล็กทรอนิกส์เหล่านี้จำเป็นต้องมีการเขียนโปรแกรมลงบนอุปกรณ์ควบคุมที่เรียกว่าไมโครคอนโทรลเลอร์ (Microcontroller) เพื่อให้สามารถใช้งานได้

“รถหุ่นยนต์” เป็นเทคโนโลยีหนึ่งที่มีประโยชน์ในหลากหลายรูปแบบ ไม่ว่าจะเป็นรถบังคับของเล่น รถหุ่นยนต์สำรวจ รถหุ่นยนต์ลำเลียงสิ่งของ ตามแต่ที่ผู้พัฒนาต้องการจะใช้งาน แม้ว่าในปัจจุบันจะมีโครงงานที่เกี่ยวกับการควบคุมรถหุ่นยนต์แล้วแต่ส่วนใหญ่มักจะเป็นการควบคุมผ่านโปรแกรมบนคอมพิวเตอร์หรืออุปกรณ์ควบคุมเฉพาะเพื่อสั่งการ ซึ่งก่อให้เกิดความไม่สะดวกในการใช้งานนอกสถานที่และนอกจากนี้ยังใช้การบังคับทิศทางด้วยปุ่มกดซึ่งเป็นส่วนติดต่อกับผู้ใช้ที่ล้าสมัย ผู้ใช้จะรู้สึกว่าไม่ได้มีส่วนร่วมกับระบบมากเท่าใดนัก และนอกจากนี้โครงงานส่วนมากยังได้ใช้งานสายสื่อกลางหรือระบบเครือข่ายไร้สายแบบบลูทูธเพื่อส่งคำสั่งการใช้งานซึ่งทำให้ระยะทางในการใช้งานค่อนข้างจำกัด

โครงงานนี้จึงนำเสนอการควบคุมรถหุ่นยนต์ด้วยการควบคุมแบบต่างๆ (การกดปุ่ม, การเอียง และการใช้จอยลากนิ้ว) โดยใช้สมาร์ทโฟนระบบปฏิบัติการแอนดรอยด์ (Android) เป็นตัวควบคุมการทำงานผ่านระบบเครือข่ายไร้สายสัญญาณ wifi เนื่องจากว่าเพื่อให้สามารถควบคุมรถหุ่นยนต์จากระยะไกล และสามารถใช้งานได้สะดวกทั้งภายนอกและภายในสถานที่ สามารถใช้งานได้ในระยะทางที่กว้างมากขึ้น และสมาร์ทโฟนระบบนี้ยังมีผู้ใช้งานอยู่เป็นจำนวนมากและราคาไม่สูงจนเกินไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.2 วัตถุประสงค์

1. ออกแบบและพัฒนาการควบคุมรถหุ่นยนต์ด้วยการกดปุ่ม การเอียง และการใช้จอยลากนิ้ว โดยใช้สมาร์ทโฟนระบบปฏิบัติการแอนดรอยด์ผ่านระบบเครือข่ายไร้สาย
2. เรียนรู้การเขียนแอปพลิเคชัน (Application) บนระบบปฏิบัติการแอนดรอยด์และการเขียนโปรแกรมลงบนไมโครคอนโทรลเลอร์
3. สามารถนำรถหุ่นยนต์ไปต่อยอดเพื่อใช้งานในลักษณะต่างๆ ได้ในอนาคต

1.3 ขอบเขตของปัญหาพิเศษ

1. สามารถควบคุมรถหุ่นยนต์ด้วยการกดปุ่ม การเอียง และการใช้จอยลากนิ้ว ผ่านทางสมาร์ทโฟนระบบปฏิบัติการแอนดรอยด์โดยใช้ระบบเครือขายไร้สายสัญญาณ wifi
2. ใช้ภาษาจาวา (Java Language) ในการเขียนแอปพลิเคชันบนระบบปฏิบัติการแอนดรอยด์
3. ใช้ภาษาซี (C Language) ในการเขียนโปรแกรมลงบอร์ดไมโครคอนโทรลเลอร์
4. ใช้คำสั่งเฉพาะในการเขียนการตั้งค่าลงบนอุปกรณ์รับส่งสัญญาณ wifi ที่ติดต่อกับบอร์ดไมโครคอนโทรลเลอร์

1.4 ขั้นตอนการดำเนินงาน

1. ศึกษาข้อมูลของไมโครคอนโทรลเลอร์และอุปกรณ์ที่จะนำมาใช้พัฒนารถหุ่นยนต์
2. ศึกษาข้อมูลของการเชื่อมต่อระหว่างไมโครคอนโทรลเลอร์กับเครือข่ายไร้สาย
3. ออกแบบการทำงานของหุ่นยนต์
4. ศึกษาวิธีในการเขียนโปรแกรมลงบนไมโครคอนโทรลเลอร์
5. ศึกษาวิธีในการเขียนแอปพลิเคชันด้วยภาษาจาวาบนระบบปฏิบัติการแอนดรอยด์
6. ออกแบบโปรแกรม
7. ทำการเขียนโปรแกรมและแอปพลิเคชันพร้อมทดสอบการใช้งาน
8. ทดลองนำรถหุ่นยนต์ไปใช้งานจริง
9. ปรับปรุงแก้ไขส่วนบกพร่อง
10. จัดทำเอกสารประกอบการพัฒนาปัญหาพิเศษ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1. ได้รับความรู้ ความเข้าใจในเขียนโปรแกรมลงบนไมโครคอนโทรลเลอร์เพื่อควบคุมรถหุ่นยนต์
2. ได้รับความรู้ ความเข้าใจในการเขียนแอปพลิเคชันบนระบบปฏิบัติการแอนดรอยด์
3. สามารถนำรถหุ่นยนต์ไปพัฒนาต่อยอดในอนาคตได้

1.6 อุปกรณ์ที่ต้องใช้

1.6.1 ฮาร์ดแวร์ (Hardware)

1. สมาร์ทโฟนระบบปฏิบัติการแอนดรอยด์ จำนวน 2 เครื่อง
2. เครื่องคอมพิวเตอร์พกพา จำนวน 2 เครื่อง
3. Microcontroller Atmel ATmega32U4
4. รถหุ่นยนต์
5. Wire Antenna - RN-XV WiFly Module
6. ADX-XBee 5V
7. ZX-XBeeU

1.6.2 ซอฟต์แวร์ (Software)

1. Eclipse IDE
2. TeraTeam Pro
3. Arduino

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ทฤษฎีและหลักการที่เกี่ยวข้อง

2.1 ระบบสื่อสารไร้สาย (Wireless Telecommunication) [2]

ปัจจุบันนี้โลกของเราเข้าสู่ยุคดิจิตอล เนื่องจากสัญญาณดิจิตอลสามารถส่งถ่ายข้อมูลชนิดต่างๆจากที่หนึ่งไปยังอีกที่หนึ่งในเวลาอันรวดเร็วโดยสามารถคงความถูกต้องของข้อมูลไว้ได้ การส่งข้อมูลดิจิตอลนั้นสามารถส่งผ่านตัวกลางได้หลายชนิด โดยตัวกลางที่ใช้งานได้ง่ายและถูกที่สุดคือสายส่งสัญญาณที่มีตัวนำไฟฟ้าเป็นสื่อกลางหรือที่เรียกว่าสายเคเบิล การส่งข้อมูลผ่านสายเคเบิลนั้นถูกนำมาใช้งานกันอย่างแพร่หลาย เพราะไม่ต้องเปลี่ยนสัญญาณข้อมูลไปอยู่ในรูปแบบอื่นก่อน อีกทั้งยังสามารถรองรับความเร็วในการส่งสัญญาณได้สูง แต่ปัญหาจากการเชื่อมต่อสายเคเบิลระหว่างอุปกรณ์มากกว่าสองชิ้นเข้าด้วยกันนั้นมีความยุ่งยาก ไม่สะดวกในการใช้งานอุปกรณ์ที่ไม่ได้ติดตั้งอยู่กับที่ จึงมีแนวคิดหาสื่อกลางชนิดอื่นเข้ามาแทนสายเคเบิล และในปัจจุบันมีการรับส่งข้อมูลมากขึ้นจึงทำให้การส่งข้อมูลนั้นจำเป็นต้องใช้ความเร็วที่เพิ่มมากขึ้น ดังนั้นสื่อกลางที่จะนำมาทำหน้าที่แทนนั้นต้องสามารถรองรับความเร็วในการส่งข้อมูลที่สูงนี้ได้ด้วย

ระบบสื่อสารไร้สาย (Wireless Telecommunication) หรือ ระบบเครือข่ายไร้สาย (Wireless Local Area Network) คือระบบการสื่อสารข้อมูลที่มีความคล่องตัวสูง ซึ่งนำมาใช้ทดแทนหรือเพิ่มเติมกับระบบเครือข่ายแลนที่ใช้สายแบบดั้งเดิม โดยใช้การส่งคลื่นความถี่วิทยุในย่าน คลื่นความถี่วิทยุ (Radio Frequency) และ คลื่นอินฟราเรด (Infrared) ในการรับและส่งข้อมูลระหว่างคอมพิวเตอร์แต่ละเครื่องผ่านอากาศ ทะลุกำแพง เพดานหรือสิ่งก่อสร้างอื่นๆ โดยไม่จำเป็นต้องติดตั้งสาย [1]

2.1.1 เทคโนโลยี Wi-Fi

Wi-Fi ย่อมาจาก Wireless Fidelity หมายถึงชุดผลิตภัณฑ์ต่างๆ ที่สามารถใช้ได้กับมาตรฐานเครือข่ายคอมพิวเตอร์แบบไร้สาย (WLAN) ซึ่งอยู่บนมาตรฐาน IEEE 802.11 เดิมที Wi-Fi ออกแบบมาใช้สำหรับอุปกรณ์พกพาต่างๆ และใช้เครือข่าย LAN เท่านั้น แต่ปัจจุบันนิยมใช้ Wi-Fi เพื่อต่อกับอินเทอร์เน็ต โดยอุปกรณ์พกพาต่างๆ สามารถเชื่อมต่อกับอินเทอร์เน็ตได้ผ่านอุปกรณ์ที่เรียกว่า Access Point และบริเวณที่ระยะทำการของ Access Point ครอบคลุมเรียกว่า Hotspot แต่เดิมคำว่า Wi-Fi เป็นชื่อที่ตั้งแทนตัวเลข IEEE 802.11 ซึ่งง่ายกว่าในการจดจำ โดยนำมาจากเครื่องขยายเสียง Hi-Fi อย่างไรก็

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตามในปัจจุบันใช้เป็นคำย่อของ Wireless Fidelity โดยมีแสดงในเว็บไซต์ของ Wi-Fi Alliance โดยใช้ชื่อ Wi-Fi เป็นเครื่องหมายการค้า

ในทางปฏิบัติมักมีความเข้าใจกันว่าเทคโนโลยี Wi-Fi กับมาตรฐาน WLAN เป็นสิ่งเดียวกัน แต่แท้จริงแล้ว WLAN มีความหมายถึงการให้บริการสื่อสารข้อมูลในลักษณะแบ่งกันใช้แบนด์วิธท์ ระหว่างเครื่องลูกข่าย ซึ่งส่วนใหญ่เป็นเครื่องคอมพิวเตอร์กับเครือข่ายสื่อสารไร้สาย โดยผ่านทางอุปกรณ์สถานีฐานหรือจุดเชื่อมต่อ ทั้งนี้ไม่มีการกำหนดมาตรฐานการเชื่อมต่อทางเทคนิคให้ตายตัว นอกเหนือจากเทคโนโลยี Wi-Fi แล้ว ยังมีเทคโนโลยีอื่นๆ ที่เข้าข่ายให้บริการแบบ WLAN ไม่ว่าจะเป็นเทคโนโลยี WiMAX มาตรฐานการสื่อสารแบบ Bluetooth เทคโนโลยี Home RF หรือแม้กระทั่งเทคโนโลยี HiperLAN ซึ่ง 2 เทคโนโลยีหลังนั้นยังไม่ได้รับการยอมรับใช้งานอย่างแพร่หลายเท่าใดนักในปัจจุบัน

2.1.1.1 ลักษณะการเชื่อมต่อของอุปกรณ์

Wi-Fi ได้กำหนดลักษณะการเชื่อมต่อของอุปกรณ์ภายในเครือข่ายแลนไว้ 2 ลักษณะ คือ โหมด Infrastructure และโหมด Ad-Hoc หรือ Peer-to-Peer

1. โหมด Infrastructure

โดยทั่วไปแล้วอุปกรณ์ในเครือข่าย Wi-Fi จะเชื่อมต่อกันในลักษณะของโหมด Infrastructure ซึ่งเป็นโหมดที่อนุญาตให้อุปกรณ์ภายใน LAN สามารถเชื่อมต่อกับเครือข่ายอื่นได้ ในโหมด Infrastructure นี้จะประกอบไปด้วยอุปกรณ์ 2 ประเภทได้แก่ สถานีผู้ใช้ (Client Station) ซึ่งก็คืออุปกรณ์คอมพิวเตอร์ (เดสก์ท็อป, แล็ปท็อป, สมาร์ทโฟน หรือ PDA ต่างๆ) ที่มีอุปกรณ์ Client Adapter เพื่อใช้รับส่งข้อมูลผ่าน Wi-Fi และสถานีแม่ข่าย (Access Point) ซึ่งทำหน้าที่ต่อเชื่อมสถานีผู้ใช้เข้ากับเครือข่ายอื่น (ซึ่งโดยปกติจะเป็นเครือข่าย IEEE 802.3 Ethernet LAN) การทำงานในโหมด Infrastructure มีพื้นฐานมาจากระบบเครือข่ายโทรศัพท์มือถือ กล่าวคือสถานีผู้ใช้จะสามารถรับส่งข้อมูลโดยตรงกับสถานีแม่ข่ายที่ให้บริการ แก่สถานีผู้ใช้นั้นอยู่เท่านั้น ส่วนสถานีแม่ข่ายจะทำหน้าที่ส่งต่อ (Forward) ข้อมูลที่ได้รับจากสถานีผู้ใช้ไปยังจุดหมายปลายทางหรือส่งต่อข้อมูลที่ได้ รับจากเครือข่ายอื่นมายังสถานีผู้ใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. โหมด Ad-Hoc (Peer-to-Peer)

เครือข่าย Wi-Fi ในโหมด Ad-Hoc (Peer-to-Peer) เป็นเครือข่ายที่ปิดคือไม่มีสถานีแม่ข่ายและไม่มีการเชื่อมต่อกับเครือข่ายอื่น บริเวณของเครือข่าย Wi-Fi ในโหมด Ad-Hoc จะเรียกว่า Independent Basic Service Set (IBSS) ซึ่งสถานีผู้ใช้หนึ่งสามารถติดต่อสื่อสารข้อมูลกับสถานีผู้ใช้อื่นๆในเขต IBSS เดียวกันได้โดยตรงโดยไม่ต้องผ่านสถานีแม่ข่าย แต่สถานีผู้ใช้จะไม่สามารถรับส่งข้อมูลกับเครือข่ายอื่นๆ ได้

2.1.2 โปรโตคอล TCP/IP [6]

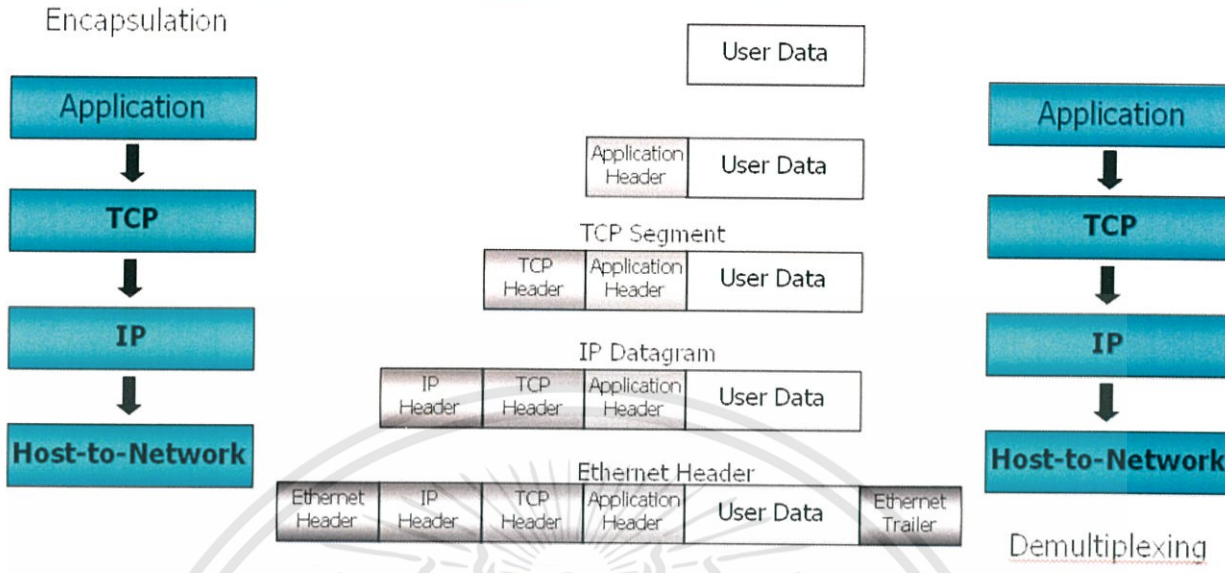
TCP/IP (Transmission Control Protocol/Internet Protocol) เป็นชุดของโปรโตคอลที่ถูกใช้ในการสื่อสารผ่านเครือข่ายอินเทอร์เน็ต โดยมีวัตถุประสงค์เพื่อให้สามารถใช้สื่อสารจากต้นทางข้ามเครือข่ายไปยังปลายทางได้ และสามารถหาเส้นทางที่จะส่งข้อมูลไปได้เองโดยอัตโนมัติ ถึงแม้ว่าในระหว่างทางอาจจะผ่านเครือข่ายที่มีปัญหา โปรโตคอลก็ยังคงหาเส้นทางอื่นในการส่งผ่านข้อมูลไปให้ถึงปลายทางได้

TCP/IP มีจุดประสงค์ของการสื่อสารตามมาตรฐาน สามประการคือ

- เพื่อใช้ติดต่อสื่อสารระหว่างระบบที่มีความแตกต่างกัน
- ความสามารถในการแก้ไขปัญหาที่เกิดขึ้นในระบบเครือข่าย เช่นในกรณีที่ผู้ส่งและผู้รับยังคงมีการติดต่อกันอยู่ แต่โหนดกลางที่ใช้เป็นผู้ช่วยรับ-ส่งเกิดเสียหายใช้การไม่ได้ หรือสายสื่อสารบางช่วงถูกตัดขาด กฎการสื่อสารนี้จะต้องสามารถจัดหาทางเลือกอื่นเพื่อทำให้การสื่อสารดำเนินต่อไปได้โดยอัตโนมัติ
- มีความคล่องตัวต่อการสื่อสารข้อมูลได้หลายชนิดทั้งแบบที่ไม่มีความเร่งด่วน เช่น การจัดส่งแฟ้มข้อมูล และแบบที่ต้องการรับประกันความเร่งด่วนของข้อมูล เช่น การสื่อสารแบบ real-time และทั้งการสื่อสารแบบเสียง (Voice) และข้อมูล (data)

การส่งข้อมูลผ่านในแต่ละเลเยอร์ แต่ละเลเยอร์จะทำการประกอบข้อมูลที่ได้รับมา กับข้อมูลส่วนควบคุมซึ่งถูกนำมาไว้ในส่วนหัวของข้อมูลเรียกว่า Header ภายใน Header จะบรรจุข้อมูลที่สำคัญของโปรโตคอลที่ทำการ Encapsulate เมื่อผู้รับได้รับข้อมูล ก็จะเกิดกระบวนการทำงานย้อนกลับคือโปรโตคอลเดียวกัน ทางฝั่งผู้รับก็จะได้รับข้อมูลส่วนที่เป็น Header ก่อนและนำไปประมวลและทราบว่าข้อมูลที่ตามมามีลักษณะอย่างไร ซึ่งกระบวนการย้อนกลับนี้เรียกว่า Demultiplexing

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



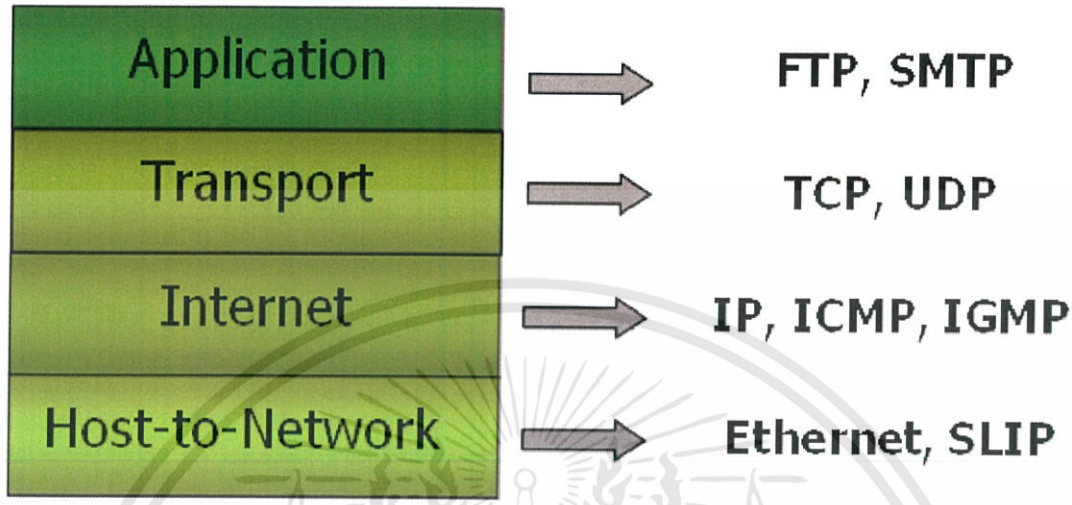
รูปที่ 2.1 ขั้นตอนการ Encapsulation และ Demultiplexing

ข้อมูลที่ผ่านการ Encapsulate ในแต่ละเลเยอร์มีชื่อเรียกแตกต่างกัน ดังนี้

- ข้อมูลที่มาจาก User หรือก็คือข้อมูลที่ User เป็นผู้ป้อนให้กับ Application เรียกว่า User Data
- เมื่อแอปพลิเคชั่นได้รับข้อมูลจาก user ก็จะนำมาประกอบกับส่วนหัวของแอปพลิเคชั่น เรียกว่า Application Data และส่งต่อไปยังโปรโตคอล TCP
- เมื่อโปรโตคอล TCP ได้รับ Application Data ก็จะนำมารวมกับ Header ของ โปรโตคอล TCP เรียกว่า TCP Segment และส่งต่อไปยังโปรโตคอล IP
- เมื่อโปรโตคอล IP ได้รับ TCP Segment ก็จะนำมารวมกับ Header ของ โปรโตคอล IP เรียกว่า IP Datagram และส่งต่อไปยังเลเยอร์ Host-to-Network Layer
- ในระดับ Host-to-Network จะนำ IP Datagram มาเพิ่มส่วน Error Correction และ flag เรียกว่า Ethernet Frame ก่อนจะแปลงข้อมูลเป็นสัญญาณไฟฟ้า ส่งผ่านสายสัญญาณที่เชื่อมโยงอยู่ต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในแต่ละเลเยอร์ของโครงสร้าง TCP/IP สามารถอธิบายได้ดังนี้



รูปที่ 2.2 โครงสร้าง TCP/IP

2.1.2.1 ชั้นโฮสต์-เครือข่าย (Host-to-Network Layer)

โพรโตคอลสำหรับการควบคุมการสื่อสารในชั้นนี้เป็นสิ่งที่ไม่มีการกำหนดรายละเอียดอย่างเป็นทางการ หน้าที่หลักคือการรับข้อมูลจากชั้นสื่อสาร IP มาแล้วส่งไปยังโหนดที่ระบุไว้ในเส้นทางเดินข้อมูลทางด้านผู้รับก็จะทำงานในทางกลับกัน คือรับข้อมูลจากสายสื่อสารแล้วนำส่งให้กับโปรแกรมในชั้นสื่อสาร

2.1.2.2 ชั้นสื่อสารอินเทอร์เน็ต (The Internet Layer)

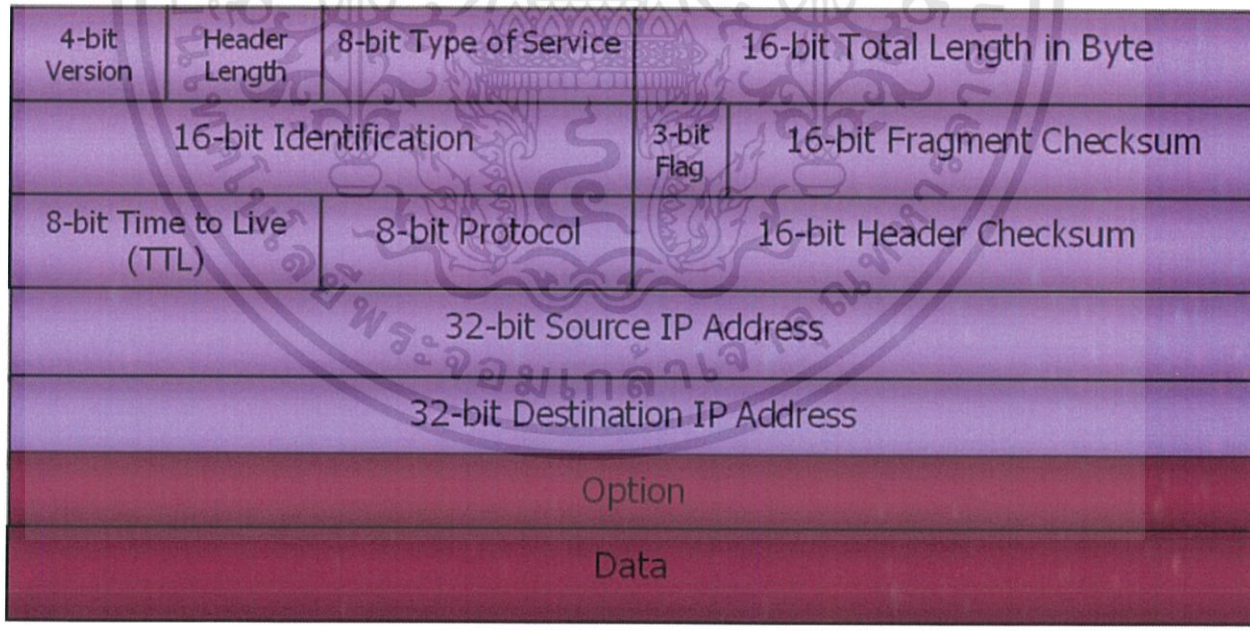
ใช้ประเภทของระบบการสื่อสารที่เรียกว่า ระบบเครือข่ายแบบสลับช่องสื่อสารระดับแพ็กเก็ต (packet-switching network) ซึ่งเป็นการติดต่อแบบไม่ต่อเนื่อง (Connectionless) หลักการทำงานคือการปล่อยให้ข้อมูลขนาดเล็กที่เรียกว่า แพ็กเก็ต (Packet) สามารถไหลจากโหนดผู้ส่งไปตามโหนดต่างๆ ในระบบจนถึงจุดหมายปลายทางได้โดยอิสระ หากว่ามีการส่งแพ็กเก็ตออกมาเป็นชุดโดยมีจุดหมายปลายทางเดียวกันในระหว่างการเดินทางในเครือข่าย แพ็กเก็ตแต่ละตัวในชุดนี้ก็จะเป็นอิสระแก่กันและกัน ดังนั้น แพ็กเก็ตที่ส่งไปถึงปลายทางอาจจะไม่เป็นไปตามลำดับก็ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

➢ IP (Internet Protocol)

IP เป็นโปรโตคอลในระดับเน็ตเวิร์คเลเยอร์ ทำหน้าที่จัดการเกี่ยวกับแอดเดรสและข้อมูล และควบคุมการส่งข้อมูลบางอย่างที่ใช้ในการหาเส้นทางของแพ็กเก็ต ซึ่งกลไกในการหาเส้นทางของ IP จะมีความสามารถในการหาเส้นทางที่ดีที่สุด และสามารถเปลี่ยนแปลงเส้นทางได้ในระหว่างการส่งข้อมูล และมีระบบการแยกและประกอบดาต้าแกรม (datagram) เพื่อรองรับการส่งข้อมูลระดับ data link ที่มีขนาด MTU (Maximum Transmission Unit) ที่แตกต่างกัน ทำให้สามารถนำ IP ไปใช้บนโปรโตคอลอื่นได้หลากหลาย เช่น Ethernet ,Token Ring หรือ Apple Talk

การเชื่อมต่อของ IP เพื่อทำการส่งข้อมูล จะเป็นแบบ connectionless หรือเกิดเส้นทางการเชื่อมต่อในทุกๆครั้งของการส่งข้อมูล 1 ดาต้าแกรม โดยจะไม่ทราบถึงข้อมูลดาต้าแกรมที่ส่งก่อนหน้าหรือส่งตามมา แต่การส่งข้อมูลใน 1 ดาต้าแกรม อาจจะเกิดการส่งได้หลายครั้งในกรณีที่มีการแบ่งข้อมูลออกเป็นส่วนย่อยๆ (fragmentation) และถูกนำไปรวมเป็นดาต้าแกรมเดิมเมื่อถึงปลายทาง



รูปที่ 2.3 IP Header

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

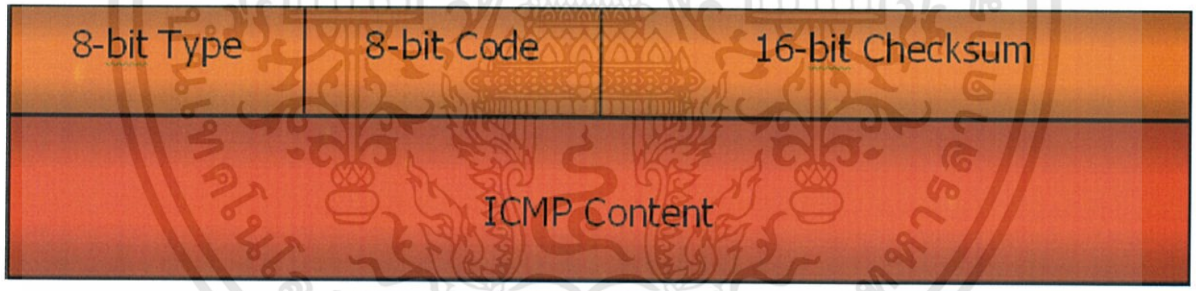
เฮดเดอร์ของ IP โดยปกติจะมีขนาด 20 bytes ยกเว้นในกรณีที่มีการเพิ่ม option บางอย่าง ฟิลด์ของเฮดเดอร์ IP จะมีความหมายดังนี้

- **Version** : หมายเลขเวอร์ชันของโปรโตคอล ที่ใช้งานในปัจจุบันคือ เวอร์ชัน 4 (IPv4) และเวอร์ชัน 6 (IPv6)
- **Header Length** : ความยาวของเฮดเดอร์ โดยทั่วไปถ้าไม่มีส่วน option จะมีค่าเป็น 5 (5*32 bit)
- **Type of Service (TOS)** : ใช้เป็นข้อมูลสำหรับเราเตอร์ในการตัดสินใจเลือกการเราต์ข้อมูลในแต่ละดาต้าแกรม แต่ในปัจจุบันไม่ได้มีการนำไปใช้งานแล้ว
- **Length** : ความยาวทั้งหมดเป็นจำนวนไบต์ของดาต้าแกรม ซึ่งด้วยขนาด 16 บิตของฟิลด์ จะหมายถึงความยาวสูงสุดของดาต้าแกรม คือ 65535 byte (64k) แต่ในการส่งข้อมูลจริง ข้อมูลจะถูกแยกเป็นส่วนๆตามขนาดของ MTU ที่กำหนดในลิงค์เลเยอร์ และนำมารวมกันอีกครั้งเมื่อส่งถึงปลายทาง แอปพลิเคชั่นส่วนใหญ่จะมีขนาดของดาต้าแกรมไม่เกิน 512 byte
- **Identification** : เป็นหมายเลขของดาต้าแกรมในกรณีที่มีการแยกดาต้าแกรมเมื่อข้อมูลส่งถึงปลายทางจะนำข้อมูลที่มี identification เดียวกันมารวมกัน
- **Flag** : ใช้ในกรณีที่มีการแยกดาต้าแกรม
- **Fragment Offset** : ใช้ในการกำหนดตำแหน่งข้อมูลในดาต้าแกรมที่มีการแยกส่วน เพื่อให้สามารถนำกลับมาเรียงต่อกันได้อย่างถูกต้อง
- **Time to Live (TTL)** : กำหนดจำนวนครั้งที่มากที่สุดที่ดาต้าแกรมจะถูกส่งระหว่าง hop (การส่งผ่านข้อมูลระหว่างเน็ตเวิร์ค) เพื่อป้องกันไม่ให้เกิดการส่งข้อมูลโดยไม่สิ้นสุด โดยเมื่อข้อมูลถูกส่งไป 1 hop จะทำการลดค่า TTL ลง 1 เมื่อค่าของ TTL เป็น 0 และข้อมูลยังไม่ถึงปลายทาง ข้อมูลนั้นจะถูกยกเลิก และเราเตอร์สุดท้ายจะส่งข้อมูล ICMP แจ้งกลับมายังต้นทางว่าเกิด time out ในระหว่างการส่งข้อมูล
- **Protocol** : ระบุโปรโตคอลที่ส่งในดาต้าแกรม เช่น TCP ,UDP หรือ ICMP
- **Header Checksum** : ใช้ในการตรวจสอบความถูกต้องของข้อมูลในเฮดเดอร์
- **Source IP Address** : หมายเลข IP ของผู้ส่งข้อมูล
- **Destination IP Address** : หมายเลข IP ของผู้รับข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Data : ข้อมูลจากโปรโตคอลระดับบน
- ICMP (Internet Control Message Protocol) : ICMP เป็นโปรโตคอลที่ใช้ในการตรวจสอบและรายงานสถานภาพของดาต้าแกรม (Datagram) ในกรณีที่เกิดปัญหากับดาต้าแกรม เช่น เราเตอร์ไม่สามารถส่งดาต้าแกรมไปถึงปลายทางได้ ICMP จะถูกส่งออกไปยังโฮสต้นทางเพื่อรายงานข้อผิดพลาด ที่เกิดขึ้น อย่างไรก็ดี ไม่มีอะไรรับประกันได้ว่า ICMP Message ที่ส่งไปจะถึงผู้รับจริงหรือไม่ หากมีการส่งดาต้าแกรมออกไปแล้วไม่มี ICMP Message ฟ้อง Error กลับมา ก็แปลความหมายได้สองกรณีคือ ข้อมูลถูกส่งไปถึงปลายทางอย่างเรียบร้อย หรืออาจจะมีปัญหา ในการสื่อสารทั้งการส่งดาต้าแกรม และ ICMP Message ที่ส่งกลับมาก็มีปัญหาระว่างทางก็ได้ ICMP จึงเป็นโปรโตคอลที่ไม่มีความน่าเชื่อถือ (unreliable) ซึ่งจะเป็นหน้าที่ของ โปรโตคอลในระดับสูงกว่า Network Layer ในการจัดการให้การสื่อสารนั้นๆ มีความน่าเชื่อถือ

ในส่วนของ ICMP Message จะประกอบด้วย Type ขนาด 8 บิต Checksum ขนาด 16 บิต และส่วนของ Content ซึ่งจะมีขนาดแตกต่างกันไปตาม Type และ Code ดังรูป



รูปที่ 2.4 ICMP Header

2.1.2.3 ชั้นสื่อสารนำส่งข้อมูล (Transport Layer)

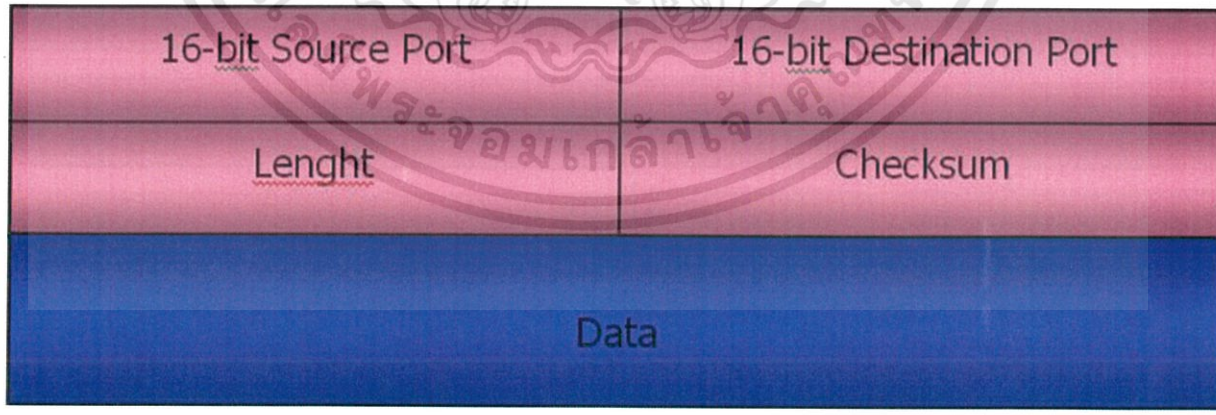
แบ่งเป็นโพรโตคอล 2 ชนิดตามลักษณะ ลักษณะแรกเรียกว่า Transmission Control Protocol (TCP) เป็นแบบที่มีการกำหนดช่วงการสื่อสารตลอดระยะเวลาการสื่อสาร (connection-oriented) ซึ่งจะยอมให้มีการส่งข้อมูลเป็นแบบ Byte stream ที่ไว้ใจได้โดยไม่มีข้อผิดพลาด ข้อมูลที่มีปริมาณมากจะถูกแบ่งออกเป็นส่วนเล็กๆ เรียกว่า message ซึ่งจะถูกส่งไปยังผู้รับผ่านทางชั้นสื่อสารของอินเทอร์เน็ต ทางฝ่ายผู้รับจะนำ message มาเรียงต่อกันตามลำดับเป็นข้อมูลตัวเดิม TCP ยังมีความสามารถในการควบคุมการไหลของข้อมูลเพื่อป้องกันไม่ให้ผู้ส่ง ส่งข้อมูลเร็วเกินกว่าที่ผู้รับจะทำงานได้ทันอีกด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรโตคอลการนำส่งข้อมูลแบบที่สองเรียกว่า UDP (User Datagram Protocol) เป็นการติดต่อแบบไม่ต่อเนื่อง (connectionless) มีการตรวจสอบความถูกต้องของข้อมูลแต่จะไม่มีการแจ้งกลับไปยังผู้ส่ง จึงถือได้ว่าไม่มีการตรวจสอบความถูกต้องของข้อมูล อย่างไรก็ตาม วิธีการนี้มีข้อดีในด้านความรวดเร็วในการส่งข้อมูล จึงนิยมใช้ในระบบผู้ให้และผู้ใช้บริการ (client/server system) ซึ่งมีการสื่อสารแบบ ถาม/ตอบ (request/reply) นอกจากนั้นยังใช้ในการส่งข้อมูลประเภทภาพเคลื่อนไหวหรือการส่งเสียง (voice) ทางอินเทอร์เน็ต

➢ UDP (User Datagram Protocol)

เป็นโปรโตคอลที่อยู่ใน Transport Layer เมื่อเทียบกับโมเดล OSI โดยการส่งข้อมูลของ UDP นั้นจะเป็นการส่งครั้งละ 1 ชุดข้อมูล เรียกว่า UDP datagram ซึ่งจะไม่มีความสัมพันธ์กันระหว่างดาต้าแกรมและจะไม่มีกลไกการตรวจสอบความสำเร็จในการรับส่งข้อมูล กลไกการตรวจสอบโดย checksum ของ UDP นั้นเพื่อเป็นการป้องกันข้อมูลที่อาจจะถูกแก้ไข หรือมีความผิดพลาดระหว่างการส่ง และหากเกิดเหตุการณ์ดังกล่าว ปลายทางจะได้รู้ว่ามีข้อผิดพลาดเกิดขึ้น แต่มันจะเป็นการตรวจสอบเพียงฝ่ายเดียวเท่านั้น โดยในข้อกำหนดของ UDP หากพบว่า Checksum Error ก็ให้ผู้รับปลายทางทำการทิ้งข้อมูลนั้น แต่จะไม่มีการแจ้งกลับไปยังผู้ส่งแต่อย่างใด การรับส่งข้อมูลแต่ละครั้งหากเกิดข้อผิดพลาดในระดับ IP เช่น ส่งไม่ถึง, หมดเวลา ผู้ส่งจะได้รับ Error Message จากระดับ IP เป็น ICMP Error Message แต่เมื่อข้อมูลส่งถึงปลายทางถูกต้อง แต่เกิดข้อผิดพลาดในส่วนของ UDP เอง จะไม่มีการยืนยัน หรือแจ้งให้ผู้ส่งทราบแต่อย่างใด



รูปที่ 2.5 UDP Header

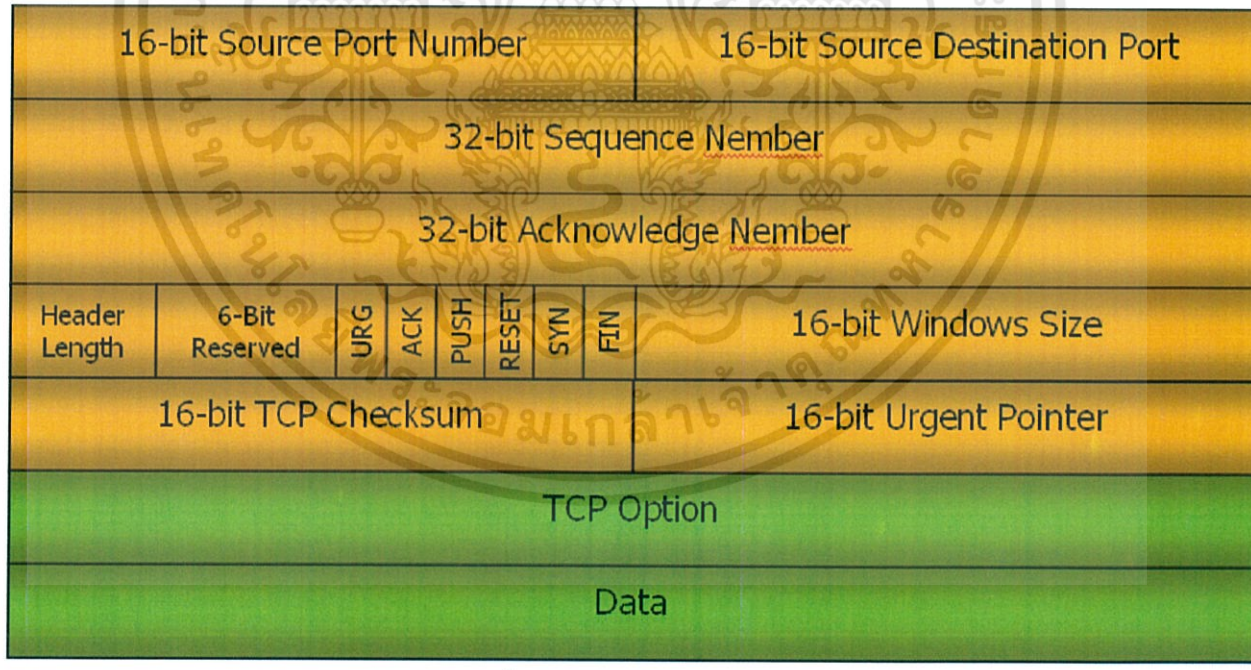
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มีรายละเอียดดังนี้

1. Source Port Number : หมายเลขพอร์ตต้นทางที่ส่งดาต้าแกรมนี้
2. Destination Port Number : หมายเลขพอร์ตปลายทางที่จะเป็นผู้รับดาต้าแกรม
3. UDP Length : ความยาวของดาต้าแกรม ทั้งส่วน Header และ data นั่นหมายความว่า ค่าที่น้อยที่สุดในฟิลด์นี้คือ 8 ซึ่งเป็นขนาดของ Header
4. Checksum : เป็นตัวตรวจสอบความถูกต้องของ UDP datagram และจะนำข้อมูลบางส่วนใน IP Header มาคำนวณด้วย

➢ TCP (Transmission Control Protocol)

อยู่ใน Transport Layer เช่นเดียวกับ UDP ทำหน้าที่จัดการและควบคุมการรับส่งข้อมูล ซึ่งมีความสามารถและรายละเอียดมากกว่า UDP โดยดาต้าแกรมของ TCP จะมีความสัมพันธ์ต่อเนื่องกัน และมีกลไกควบคุมการรับส่งข้อมูลให้มีความถูกต้อง (reliable) และมีการสื่อสารอย่างเป็นกระบวนการ (connection-oriented)



รูปที่ 2.6 TCP Header

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มีรายละเอียด ดังนี้

1. Source Port Number : หมายเลขพอร์ตต้นทางที่ส่งดาต้าแกรมนี้
2. Destination Port Number : หมายเลขพอร์ตปลายทางที่จะเป็นผู้รับดาต้าแกรม
3. Sequence Number : ฟิลด์ที่ระบุหมายเลขลำดับอ้างอิงในการสื่อสารข้อมูลแต่ละครั้ง เพื่อใช้ในการแยกแยะว่าเป็นข้อมูลของชุดใด และนำมาจัดลำดับได้ถูกต้อง
4. Acknowledgment Number : ทำหน้าที่เช่นเดียวกับ Sequence Number แต่จะใช้ในการตอบรับ
5. Header Length : โดยปกติความยาวของเฮดเดอร์ TCP จะมีความยาว 20 ไบต์ แต่อาจจะมากกว่านั้น ถ้ามีข้อมูลในฟิลด์ option แต่ต้องไม่เกิน 60 ไบต์
6. Flag : เป็นข้อมูลระดับบิตที่อยู่ในเฮดเดอร์ TCP โดยใช้เป็นตัวบอกคุณสมบัติของแพ็กเก็ต TCP ขณะนั้นๆ และใช้เป็นตัวควบคุมจังหวะการรับส่งข้อมูลด้วย ซึ่ง Flag มีอยู่ทั้งหมด 6 บิต แบ่งได้ดังนี้

ตารางที่ 2.1 Flag ทั้งหมด 6 บิต

Type	Description
URG	ใช้บอกความหมายว่าเป็นข้อมูลด่วน และมีข้อมูลพิเศษมาด้วย (อยู่ใน Urgent Pointer)
ACK	แสดงว่าข้อมูลในฟิลด์ Acknowledge Number นำมาใช้งานได้
DSH	เป็นการแจ้งให้ผู้รับข้อมูลทราบว่าควรจะส่งข้อมูล Segment นี้ไปยัง Application ที่กำลังรออยู่โดยเร็ว
RST	ยกเลิกการติดต่อ (Reset) เนื่องจากในกรณีที่เกิดการสับสนขึ้นด้วยเหตุผลต่างๆ เช่น โฮสต์มีปัญหา ให้เริ่มสื่อสารใหม่
SYN	ใช้ในการเริ่มต้นขอติดต่อกับปลายทาง
FIN	ใช้ส่งเพื่อแจ้งให้ปลายทางทราบว่ายุติการติดต่อ

Flag ในเฮดเดอร์ของ TCP มีความสำคัญในการกำหนดการทำงานของ TCP segment เนื่องจากข้อมูลในเฮดเดอร์ของ TCP จะมีข้อมูลครบถ้วนทั้งการรับและการส่งข้อมูล ซึ่งในการทำงานแต่ละอย่างจะมีการใช้งานฟิลด์ไม่เหมือนกัน flag จะเป็นตัวกำหนดว่าให้ใช้งานฟิลด์ไหน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เช่น ฟิลด์ Acknowledgment number จะไม่ถูกใช้ในขั้นตอนการเริ่มต้นการเชื่อมต่อ แต่จะมีข้อมูลในฟิลด์ซึ่งเป็นข้อมูลที่ไม่มีความหมายใดๆ ซึ่งถ้าไม่มี flag เป็นตัวกำหนดก้ออาจมีการนำข้อมูลมาใช้และก่อให้เกิดความผิดพลาดได้

2.1.2.4 ชั้นสื่อสารการประยุกต์ (Application Layer)

มีโพรโตคอลสำหรับสร้างจอเทอร์มินัลเสมือน เรียกว่า TELNET โพรโตคอลสำหรับการจัดการแฟ้มข้อมูล เรียกว่า FTP และโพรโตคอลสำหรับการให้บริการจดหมายอิเล็กทรอนิกส์ เรียกว่า SMTP โดยโพรโตคอลสำหรับสร้างจอเทอร์มินัลเสมือนช่วยให้ผู้ใช้สามารถติดต่อกับเครื่องโฮสต์ที่อยู่ไกลออกไปโดยผ่านอินเทอร์เน็ต และสามารถทำงานได้เสมือนกับว่ากำลังนั่งทำงานอยู่ที่เครื่องโฮสต์นั้น โพรโตคอลสำหรับการจัดการแฟ้มข้อมูลช่วยในการคัดลอกแฟ้มข้อมูลมาจากเครื่องอื่นที่อยู่ในระบบเครือข่ายหรือส่งสำเนาแฟ้มข้อมูลไปยังเครื่องใดๆก็ได้ โพรโตคอลสำหรับให้บริการจดหมายอิเล็กทรอนิกส์ช่วยในการจัดส่งข้อความไปยังผู้ใช้ในระบบ หรือรับข้อความที่มีผู้ส่งเข้ามา

2.2 Android [3]

คอมพิวเตอร์ทุกเครื่องจะใช้งานได้ก็ต่อเมื่อมีระบบปฏิบัติการ (Operating System) เช่น ระบบปฏิบัติการของคอมพิวเตอร์ที่เป็นที่นิยมมากที่สุด คือ ระบบปฏิบัติการ Windows

สมาร์ทโฟนถือเป็นคอมพิวเตอร์ขนาดเล็กชนิดหนึ่งจึงจำเป็นที่จะต้องมีระบบปฏิบัติการเช่นเดียวกัน โดยระบบปฏิบัติการที่เป็นที่นิยมนิยมในสมาร์ทโฟนมีมากมายจากหลากหลายผู้พัฒนา ซึ่งระบบปฏิบัติการแอนดรอยด์นั้นถูกคิดค้นโดยบริษัทแอนดรอยด์ (Android, Inc.) ซึ่งต่อมาถูกซื้อไปพัฒนาต่อโดยบริษัทกูเกิ้ล (Google, Inc.) โดยแอนดรอยด์เป็นระบบปฏิบัติการแบบ Open Source สามารถนำไปใช้งานได้ฟรี ทำให้ได้รับความนิยมและมีผู้ใช้งานทั่วโลกมากที่สุด โดยระบบปฏิบัติการนี้ถูกปล่อยออกมาใช้งานอย่างเป็นทางการครั้งแรกเมื่อวันที่ 28 กันยายน ค.ศ 2008

แอนดรอยด์เป็นซอฟต์แวร์ที่มีโครงสร้างแบบเรียงทับซ้อนหรือแบบสแต็ก (Stack) ซึ่งรวมเอาระบบปฏิบัติการ, มิดเดิลแวร์ (Middleware) และแอปพลิเคชันที่สำคัญเข้าไว้ด้วยกัน เพื่อใช้สำหรับทำงานบนอุปกรณ์พกพาเคลื่อนที่ (Mobile Devices) เช่น โทรศัพท์มือถือ เป็นต้น การทำงานของ แอนดรอยด์มีพื้นฐานอยู่บนระบบลีนุกซ์ เคอร์เนล (Linux Kernel) ซึ่งใช้ Android SDK (Software

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Development Kit) เป็นเครื่องมือสำหรับการพัฒนาแอปพลิเคชันบนระบบปฏิบัติการแอนดรอยด์และใช้ภาษา Java ในการพัฒนา

2.2.1 Google Android

กูเกิลแอนดรอยด์ (Google Android) คือ ระบบปฏิบัติการที่เป็นซอฟต์แวร์แพลตฟอร์มบนมือถือสร้างขึ้นมาจากระบบปฏิบัติการลีนุกส์ (Powered by the Linux kernel) พัฒนาขึ้นมาโดยกูเกิล กูเกิลแอนดรอยด์นั้นได้เปิดให้นักพัฒนาสามารถเข้ามาจัดการเขียนโค้ดต่างๆได้ด้วยภาษาจาวา และเขียนควบคุมอุปกรณ์ต่างๆ ผ่านทาง จาวาไลบรารี่ ที่ทางกูเกิลพัฒนาขึ้นโดยเฉพาะ (Google-developed Java libraries) โปรแกรมต่างๆ ที่รันบนกูเกิลแอนดรอยด์สามารถเขียนได้ด้วยภาษาซี (C) และภาษาอื่นๆ

แอนดรอยด์ (Android) คือ ระบบปฏิบัติการหรือแพลตฟอร์ม ที่จะใช้ควบคุมการทำงานบนอุปกรณ์อิเล็กทรอนิกส์ต่างๆ สำหรับโทรศัพท์มือถือ และอุปกรณ์พกพาโดยมี บริษัทกูเกิล, ที-โมบาย (T-Mobile) , เอชทีซี (HTC) , ควอลคอมม์ (Qualcomm) , โมโตโรลา (motorola) และบริษัทชั้นนำอีกมากมายร่วมพัฒนา โดยใช้องค์ประกอบที่เป็นโอเพนซอร์สหลายอย่าง เช่น Linux Kernel, SSL, OpenGL, FreeType, SQLite, WebKit และเขียนไลบรารีเฟรมเวิร์คของตัวเองเพิ่มเติม

2.2.2 คุณลักษณะพิเศษของระบบปฏิบัติการแอนดรอยด์

ระบบปฏิบัติการแอนดรอยด์นั้น มีคุณลักษณะพิเศษที่ทำให้แตกต่างจากระบบปฏิบัติการของสมาร์ทโฟนอื่นๆ อยู่มาก และยังเป็นเหตุผลที่ทำให้ระบบปฏิบัติการแอนดรอยด์เป็นที่น่าสนใจเป็นอย่างมาก ณ เวลาช่วงปัจจุบัน ซึ่งมีตัวอย่างคุณลักษณะเด่นๆ ดังนี้

2.2.2.1 ความเปิดกว้างในการพัฒนา

หนึ่งองค์ประกอบหลักที่ทำให้ระบบปฏิบัติการแอนดรอยด์มีชื่อเสียงโด่งดัง และมีการพัฒนาอย่างรวดเร็ว เป็นที่นิยมอย่างแพร่หลาย โดยทางบริษัทกูเกิล มีแนวคิดที่เปิดให้ผู้คนทั่วไปสามารถเข้าถึงและพัฒนาโปรแกรมบนแอนดรอยด์ได้ ไม่ว่าจะไปจัดการกับส่วนระบบของสมาร์ทโฟนหรือการจัดการต่างๆ เช่น การเข้าถึงระบบการโทรได้ การจัดการส่วนของการแสดงผล หน้าตาของส่วนการใช้งาน จัดการกับกล้อง และอื่นๆ อีกมากมาย อีกทั้งยังมีการเผยแพร่ซอร์สโค้ด (Source Code) ของโปรแกรมต่างๆ เพื่อเป็นแนวทางในการพัฒนาให้กับนักพัฒนาทั้งหลาย และยังเป็นสิ่งที่ช่วยผลักดันให้ระบบปฏิบัติการแอนดรอยด์ก้าวหน้าไปอย่างรวดเร็วอีกด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.2.2 การรองรับเทคโนโลยีการสื่อสารระยะใกล้ (NFC)

Near Field Communication (NFC) เป็นเทคโนโลยีการสื่อสารไร้สายระยะสั้นที่ใช้ได้ดีกับโครงสร้างพื้นฐานแบบไร้การสัมผัส ช่วยสนับสนุนรองรับการสื่อสารระหว่างเครื่องมืออิเล็กทรอนิกส์ในระยะใกล้ๆ NFC ถูกพัฒนาขึ้นโดย Sony และ NXP โดยใช้คลื่นความถี่ 13.56 MHz บนพื้นฐานมาตรฐาน ISO 14443 (Philips MIFARE and Sony's FeliCa) ปัจจุบันบริษัททั้งสองได้ร่วมมือกับบริษัทผู้ลิตและพัฒนาโทรศัพท์เคลื่อนที่จัดตั้งเป็น NFC Forum เพื่อให้เกิดการใช้งานในรูปแบบต่างๆมากขึ้น ในระยะเริ่มแรกมีบริษัทโทรศัพท์มือถือชั้นนำของโลกประกาศนำเทคโนโลยีนี้มาใช้กับโทรศัพท์มือถือแล้ว เช่น Nokia, Samsung, Motorola เป็นต้น การประยุกต์ใช้งานส่วนใหญ่มักนำ NFC มาใช้กับการชำระเงินที่ต้องการความรวดเร็วและมีมูลค่าไม่สูง ซึ่งจะทำให้โทรศัพท์เคลื่อนที่สามารถใช้เพื่อการชำระเงิน โดยวิธีการแตะบนเครื่องอ่านหรือเครื่องชำระเงิน เช่น การให้บริการในร้านอาหารจานด่วน ร้านขายสินค้า ระบบการซื้อขายตั๋ว และระบบการแลกเปลี่ยนข้อมูลแบบ Peer-to-Peer เช่น เพลง เกม และรูปภาพ การชำระเงินค่าโดยสารในระบบขนส่งมวลชน เป็นต้น การชำระเงินแบบไร้สัมผัสนี้ก่อให้เกิดการชำระเงินที่ง่ายและรวดเร็ว ลดการเข้าคิวชำระเงินในร้านค้า ห้างสรรพสินค้า และร้านสะดวกซื้อต่างๆ ซึ่งในปัจจุบันเทคโนโลยี NFC สามารถต่อต่อสื่อสารได้ในระยะทั่วไปที่มีประสิทธิภาพที่ดีที่สุดคือ 4 เซนติเมตร แต่ระยะการทำงานสามารถใช้งานได้ตั้งแต่ 10 ถึง 20 เซนติเมตร

2.2.2.3 การนำโปรแกรมที่พัฒนานำไปเผยแพร่สู่ Google Play

จากความเปิดกว้างในการพัฒนาดังที่ได้อธิบายในข้างต้น ทำให้นักพัฒนาโปรแกรมสนใจและทำการพัฒนาโปรแกรมบนระบบปฏิบัติการแอนดรอยด์ขึ้นมามากมาย เมื่อโปรแกรมที่พัฒนาเสร็จสิ้นแล้วก็สามารถนำมาเผยแพร่สู่สาธารณชนได้ผ่านทาง Google Play ซึ่งทำหน้าที่ในการนำเสนอและเผยแพร่โปรแกรมต่างๆ ให้สามารถดาวน์โหลดไปใช้งานฟรี รวมถึงโปรแกรมที่คิดค่าการใช้บริการซึ่งในการนำโปรแกรมที่พัฒนาแล้ว ลงสู่ Google Play นั้น สามารถทำได้ง่ายมากเพียงแค่สมัครสมาชิก http://play.google.com/apps/publish/ โดยสามารถใช้รหัสของ Google Mail ได้ จากนั้นก็ทำการอัพโหลดโปรแกรมลงสู่ Google Play แต่ในการอัพโหลดโปรแกรมลง Google Play นั้น จำเป็นที่จะต้องทำการแปลงไฟล์โปรเจกต์ของโปรแกรมให้เป็นนามสกุล .apk (Android Package) และ Sign โปรแกรมเสียก่อน โดยการ Sign โปรแกรมนั้นคือการสร้างหลักฐานหรือรายละเอียดด้านผู้พัฒนาให้กับโปรแกรม รวมถึงอายุการใช้งานของโปรแกรมนั้นๆอีกด้วย เพียงเท่านี้ก็สามารถนำโปรแกรมที่พัฒนาแล้วขึ้นไปบน Google Play ได้อย่างง่ายดาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.3 ชนิดของโปรแกรมประยุกต์

2.2.3.1 Foreground Activity

คือ โปรแกรมประยุกต์ที่จะทำงานและติดต่อกับผู้ใช้ในขณะที่มองเห็นได้เท่านั้น (บน Foreground) ซึ่งเมื่อไม่เห็นโปรแกรมหรือการปิดโปรแกรม ก็จะสิ้นสุดการทำงานพูดง่ายๆได้ว่าทำงานเมื่อมองเห็นได้เท่านั้น

2.2.3.2 Background Service

คือ โปรแกรมประยุกต์ที่มีการติดต่อกับผู้ใช้น้อยมาก อาจมีบ้างในการตั้งค่าการทำงานและวงจรการทำงานส่วนใหญ่ของโปรแกรมชนิดนี้จะซ่อนตัวอยู่ นั้นคือการทำงานโดยที่เราไม่เห็นหน้าตาของโปรแกรม เช่นโปรแกรมการแสดงภาพของผู้ติดต่อที่โทรเข้าหรือโทรออก หรือโปรแกรมในการส่งข้อความอัตโนมัติ (การทำงานจะทำงานโดยที่เราไม่จำเป็นต้องเห็น ถ้าโดนปิดไป เมื่อมันเริ่มทำงานใหม่เราก็ไม่รู้เพราะทำงานอยู่ที่ background)

2.2.3.3 Intermittent Activity

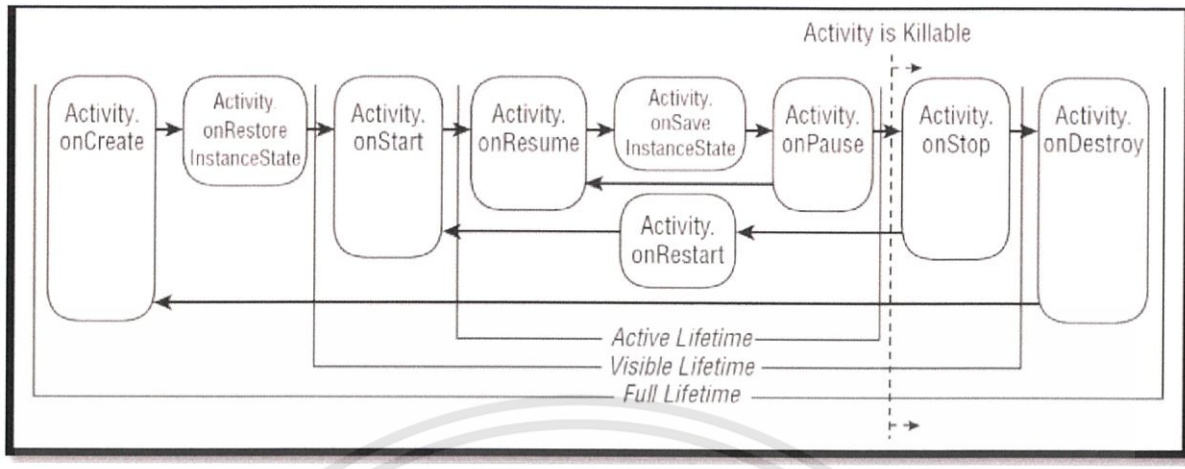
คือ โปรแกรมประยุกต์ที่มีการติดต่อกับผู้ใช้บ้าง แต่การทำงานส่วนใหญ่จะทำใน background ซึ่งโดยส่วนใหญ่แล้วโปรแกรมเหล่านี้จะทำงานอย่างเงียบๆ และอาจมีการแจ้งเตือนเมื่อมีความเคลื่อนไหวของโปรแกรม ตัวอย่างที่เห็นได้ชัดคือ โปรแกรม media player ซึ่งอาจมีการติดต่อกับผู้ใช้บ้าง เช่นเลือกเพลงต่างๆ จากนั้นเมื่อผู้ใช้ปิดไป จะทำการซ่อนตัวเองไปทำงานในชั้น background

2.2.4 Activity Life Cycle

Activity มีหน้าที่ในการสร้าง และควบคุมการทำงานของหน้าจอ เพื่อโต้ตอบกับผู้ใช้งานซึ่งจะมีได้แค่ Activity เดียวเท่านั้นที่โต้ตอบกับผู้ใช้งานในขณะใดขณะหนึ่ง Activity แต่ละตัวจะมีวงจรชีวิตเป็นของตนเอง โดยแบ่งเป็นสถานะดังนี้

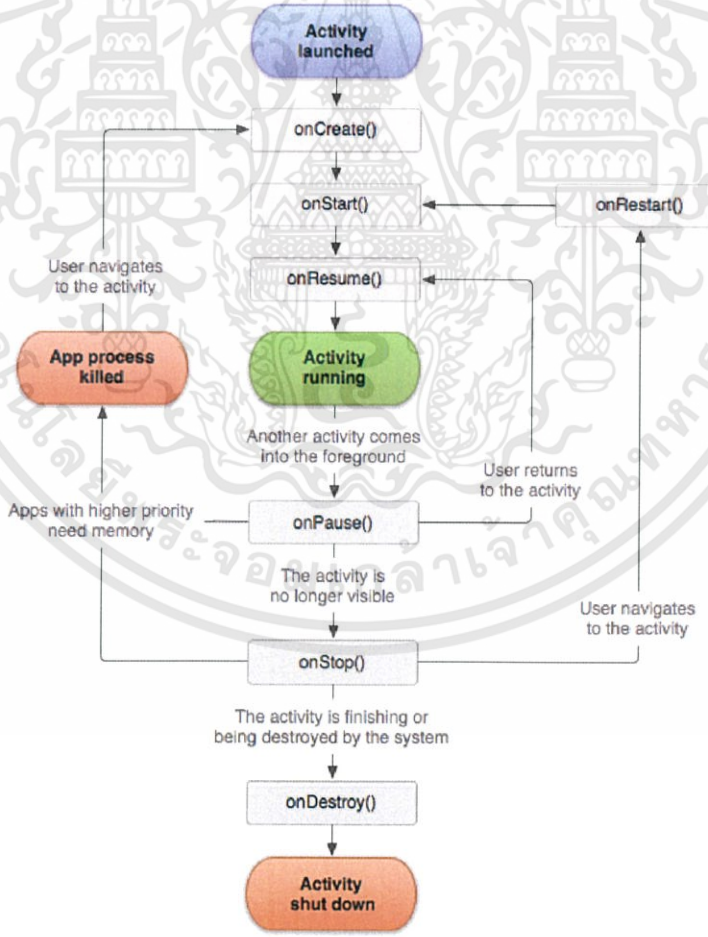
1. Running เป็นสถานะที่ Activity กำลังได้ทำงานในขณะนั้น
2. Pause เป็นสถานะที่ Activity ปรากฏอยู่บนหน้าจอ แต่ไม่ได้ทำงานในขณะนั้น
3. Stop เป็นสถานะที่ Activity ไม่ปรากฏบนหน้าจอ
4. Kill เป็นสถานะที่ Activity ถูกทำลายไปแล้วเมื่อ Activity อยู่ในสถานะ Pause หรือ Stop Activity สามารถถูกทำลายได้โดยระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.7 Activity Life Cycle

จะแสดงขั้นตอนการทำงานและการเปลี่ยนวงจร Lift Cycle เป็น Flowchart ดังรูปที่ 2.8



รูปที่ 2.8 Activity Life Cycle Flowchart

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. onCreate(Bundle)

   จะถูกเรียกเมื่อ Activity มีการสร้างตัวเองครั้งแรก เป็น method สำคัญในการสร้างหน้าจอการทำงานต่างๆ หรือกำหนดสถานะต่างๆ ของ Control ที่จะใช้งาน

2. onRestart()

   จะถูกเรียกหลังจากที่ Activity กลับมาจากสถานะ Stop เพื่อเริ่มทำงานใหม่

3. onStart()

   จะถูกเรียกเมื่อ Activity เริ่มแสดงให้ผู้ใช้งานเห็น

4. onResume()

   จะถูกเรียกเมื่อ Activity สามารถที่จะโต้ตอบกับผู้ใช้งานได้

5. onPause()

   จะถูกเรียกเมื่อมี Activity อื่นกำลังทำงาน เป็น method สำคัญในการเก็บข้อมูลที่สำคัญ หรือทำการปล่อยหน่วยความจำเพื่อให้ Activity อื่นได้ใช้งาน

6. onStop()

   จะถูกเรียกเมื่อ Activity ไม่ปรากฏบนหน้าจอ แล้วจะเข้าสู่สถานะ Stop

7. onDestroy()

   จะถูกเรียกเมื่อ Activity กำลังจะถูกทำลาย

8. onSaveInstanceState(Bundle)

   จะถูกเรียกก่อน onPause() เพื่อเก็บสถานะต่างๆ ของ Activity ไว้ใช้เมื่อ Activity นี้ได้กลับมาทำงานอีกครั้ง

9. onRestoreInstanceState(Bundle)

   ถูกเรียกก่อน onResume() เพื่อนำสถานะต่างๆ ของ Activity กลับมาใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 Java

Java หรือ Java programming language คือภาษาโปรแกรมเชิงวัตถุ พัฒนาโดย เจมส์ กอสลิง และวิศวกรคนอื่นๆ ที่บริษัท ซัน ไมโครซิสเต็มส์ ภาษานี้มีจุดประสงค์เพื่อใช้แทนภาษา C++ โดยรูปแบบที่เพิ่มเติมขึ้นคล้ายกับภาษา Objective-C แต่เดิมภาษานี้เรียกว่า ภาษาโอ๊ก (Oak) ซึ่งตั้งชื่อตามต้นโอ๊กใกล้ที่ทำงานของ เจมส์ กอสลิง แล้วภายหลังจึงเปลี่ยนไปใช้ชื่อ "จาวา" ซึ่งเป็นชื่อกาแฟแทน จุดเด่นของภาษา Java อยู่ที่ผู้เขียนโปรแกรมสามารถใช้หลักการของ Object-Oriented Programming มาพัฒนาโปรแกรมของตนเองด้วย Java ได้ โปรแกรมที่เขียนขึ้นถูกสร้างภายในคลาส ดังนั้นคลาสคือที่เก็บเมทอด (Method) หรือพฤติกรรม (Behavior) ซึ่งมีสถานะ (State) และรูปพรรณ (Identity) ประจำพฤติกรรม (Behavior)

2.3.1 ข้อดีของภาษา Java

1. ภาษา Java เป็นภาษาที่สนับสนุนการเขียนโปรแกรมเชิงวัตถุแบบสมบูรณ์ ซึ่งเหมาะสำหรับพัฒนาระบบที่มีความซับซ้อน การพัฒนาโปรแกรมแบบวัตถุจะช่วยให้เราสามารถใช้คำหรือชื่อต่างๆ ที่มีอยู่ในระบบงานนั้นมาใช้ในการออกแบบโปรแกรมได้ ทำให้เข้าใจได้ง่าย
2. โปรแกรมที่เขียนขึ้นโดยใช้ภาษา Java จะมีความสามารถทำงานได้ในระบบปฏิบัติการที่แตกต่างกัน ไม่จำเป็นต้องดัดแปลงแก้ไขโปรแกรม เช่น หากเขียนโปรแกรมบนเครื่อง Sun โปรแกรมนั้นก็สามารถถูก compile และ run บนเครื่องคอมพิวเตอร์ธรรมดาได้
3. ภาษาจาวามีการตรวจสอบข้อผิดพลาดทั้งตอน compile time และ runtime ทำให้ลดข้อผิดพลาดที่อาจเกิดขึ้นในโปรแกรม และช่วยให้ debug โปรแกรมได้ง่าย
4. ภาษาจาวามีความซับซ้อนน้อยกว่าภาษา C++ เมื่อเปรียบเทียบ code ของโปรแกรมที่เขียนขึ้นโดยภาษา Java กับ C++ พบว่า โปรแกรมที่เขียนโดยภาษา Java จะมีจำนวน code น้อยกว่าโปรแกรมที่เขียนโดยภาษา C++ ทำให้ใช้งานได้ง่ายกว่าและลดความผิดพลาดได้มากขึ้น
5. ภาษาจาวาถูกออกแบบมาให้มีความปลอดภัยสูงตั้งแต่แรก ทำให้โปรแกรมที่เขียนขึ้นด้วยจาวามีความปลอดภัยมากกว่าโปรแกรมที่เขียนขึ้นด้วยภาษาอื่น เพราะ Java มี security ทั้ง low level และ high level ได้แก่ electronic signature, public and private key management, access control และ certificates

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6. มี IDE, application server, และ library ต่างๆ มากมายสำหรับจาวาที่เราสามารถใช้งานได้โดยไม่ต้องเสียค่าใช้จ่าย ทำให้เราสามารถลดค่าใช้จ่ายที่ต้องเสียไปกับการซื้อ tool และ s/w ต่าง ๆ

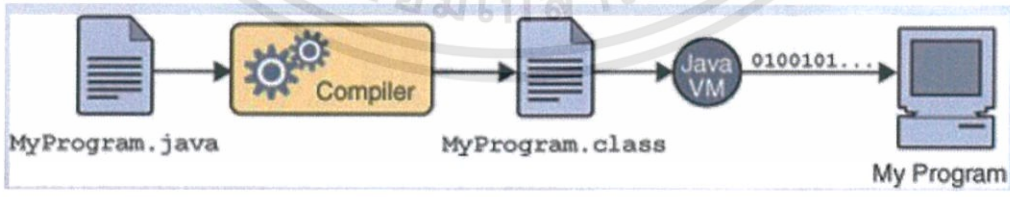
2.3.2 ข้อเสียของภาษา Java

1. ทำงานได้ช้ากว่า native code (โปรแกรมที่ compile ให้อยู่ในรูปของภาษาเครื่อง) หรือโปรแกรมที่เขียนขึ้นด้วยภาษาอื่น อย่างเช่น C หรือ C++ ทั้งนี้ก็เพราะว่าโปรแกรมที่เขียนขึ้นด้วยภาษาจาวาจะถูกแปลงเป็นภาษากลาง ก่อน แล้วเมื่อโปรแกรมทำงานคำสั่งของภาษากลางนี้จะถูกเปลี่ยนเป็นภาษาเครื่องอีก ทีหนึ่ง ทีละคำสั่ง (หรือกลุ่มของคำสั่ง) ณ runtime ทำให้ทำงานช้ากว่า native code ซึ่งอยู่ในรูปของภาษาเครื่องแล้วตั้งแต่ compile โปรแกรมที่ต้องการความเร็วในการทำงานจึงไม่นิยมเขียนด้วยจาวา
2. tool ที่มีในการใช้พัฒนาโปรแกรมจาวามักไม่ค่อยเก่ง ทำให้หลายอย่างโปรแกรมเมอร์จะต้องเป็นคนทำเอง ทำให้ต้องเสียเวลาทำงานในส่วนที่ tool ทำไม่ได้ ถ้าเราดู tool ของ MS จะใช้งานได้ง่ายกว่า และพัฒนาได้เร็วกว่า (แต่เราต้องซื้อ tool ของ MS และก็ต้องรันบน platform ของ MS)

2.3.3 เบื้องหลังการทำงานของโปรแกรมภาษา Java

การเขียนโปรแกรมของภาษา Java เราจะได้ Source Code ซึ่งเป็นไฟล์ที่มีนามสกุลเป็น .java จากนั้นเรานำ Source Code ไปคอมไพล์ให้กลายเป็นเป็น Java Byte Code (จะเก็บอยู่ในไฟล์ .class)

เวลาที่ทำงานจริงในคอมพิวเตอร์ หรืออุปกรณ์ใดๆ ก็ตาม Java Byte Code จะถูกคอมไพล์อีกครั้งให้เป็นภาษาเครื่องเฉพาะคอมพิวเตอร์ หรืออุปกรณ์นั้นๆ เข้าใจ การคอมไพล์ครั้งนี้จะใช้ Java Virtual Machine คอมไพล์และรัน



รูปที่ 2.9 ขั้นตอนการใช้งาน Source Code ภาษาจาวา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.4 หลักการเขียนโปรแกรมด้วยภาษา Java

การเขียน Java Syntax กับระบบปฏิบัติการแอนดรอยด์ด้วย Eclipse IDE for Java Developer นั้นจะใช้ Java Platform ในการพัฒนาและเขียนคำสั่งให้โปรแกรมทำงานแบบ Object-Oriented Programming (OOP) และ API Library ต่างๆ ที่อยู่ใน Android Framework ที่สามารถเรียกใช้งานได้จะมีคำสั่งสำหรับเรียกใช้งานมากมายตามวัตถุประสงค์ โดยจะแยกเป็น Package หรือ API Class Library ต่าง ๆ จัดเก็บไว้เป็นหมวดหมู่เพื่อความสะดวกและง่ายต่อการนำมาใช้ เมื่อต้องการใช้งาน Package หรือ Class ใดก็ค่อย Import เข้ามาใน Class ที่ต้องการใช้งาน Package หรือ Class เหล่านั้น

โครงสร้างการเขียนโปรแกรมบนระบบปฏิบัติการแอนดรอยด์โดยทั่วไปจะเป็นการทำงานของ XML Layout ซึ่งจะผสานการทำงานร่วมกันระหว่าง XML กับ Java โดย XML จะถูกออกแบบให้เป็นส่วน Graphical User Interface (GUI) และใช้ XML Syntax ในการวาง Layout ต่าง ๆ ของ Widgets หรือ Element ต่าง ๆ ส่วนภาษา Java จะเป็นชุดคำสั่งที่จะควบคุมการทำงานของโปรแกรมและหน้าจอที่แสดงผลให้อยู่ในรูปแบบของ XML Layout

2.3.5 Object-Oriented Programming (OOP)

การเขียนโปรแกรมแบบ OOP มีรายละเอียดค่อนข้างมาก ซึ่งมีจุดเริ่มต้นที่การพูดถึงส่วนประกอบพื้นฐานที่สำคัญ 3 อย่างคือ Object, Class และ Method ซึ่งมีรายละเอียดดังนี้

1. Object

   เปรียบได้กับสิ่งของที่มีคุณสมบัติเฉพาะ (Attribute) และความสามารถในการทำงาน (Method)

2. Class

   เปรียบได้กับแม่แบบหรือพิมพ์เขียวที่ใช้สร้าง Object โดย Object ที่สร้างจากคลาสเดียวกันอาจจะมี Attribute ที่เหมือนหรือต่างกันก็ได้

3. Method

   เปรียบเทียบได้กับความสามารถในการทำงานของ Object แต่ละตัว ถูกสร้างขึ้นมาเพื่อรองรับการกระทำต่างๆ ที่เกิดจาก Object ที่สร้าง Class นั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4 Java Develop Kit (JDK)

Java Development Kit หรือ JDK คือชุดของเครื่องมือ (tools) ที่ใช้ในการพัฒนาโปรแกรม JAVA ของบริษัทซัน ไมโครซิสเต็มส์ ซึ่งใครก็ตามที่ต้องการจะพัฒนาโปรแกรมโดยใช้ภาษา Java อย่างเช่น Java compiler, Java debugger, Java doc และ Java interpreter หรือ Java VM จะต้องลง JDK นี้ มิฉะนั้นจะไม่สามารถ compile และ run java ได้ เวอร์ชันปัจจุบันของ JDK คือเวอร์ชั่น 7 ประกอบไปด้วยโปรแกรมต่างๆ อาทิเช่น โปรแกรมคอมไพเลอร์ (javac.exe) ,โปรแกรมอินเตอร์พรีเตอร์ (java.exe) , โปรแกรมดีบักเกอร์ แต่จะไม่มีโปรแกรมอีดิเตอร์

ชุดพัฒนาโปรแกรม JDK ประกอบด้วย 3 รุ่นย่อยดังนี้

1. Java SE (Standard Edition) สำหรับพัฒนาโปรแกรมบนคอมพิวเตอร์ตั้งโต๊ะทั่วไป ถือว่าเป็นรูปแบบแรกของภาษา Java

2. Java ME (Micro Edition) สำหรับพัฒนาโปรแกรมบนอุปกรณ์พกพา เช่น โทรศัพท์มือถือ หรือพีดีเอ ส่วนมากใช้เขียนโปรแกรมเกม

3. Java EE (Enterprise Edition) สำหรับพัฒนาโปรแกรมในองค์กรใหญ่ๆ หรือมีขอบเขตของโครงการกว้างมาก

2.5 Software Develop Kit (SDK)

Android SDK ย่อมาจาก Android Software Development Kit ซึ่งเป็นชุดโปรแกรมที่ทาง Google พัฒนาออกมาเพื่อแจกจ่ายให้นักพัฒนาแอปพลิเคชันหรือผู้สนใจทั่วไปสามารถดาวน์โหลดไปใช้งานโดยไม่เสียค่าใช้จ่าย ซึ่งเป็นหนึ่งในปัจจัยที่ทำให้แอปพลิเคชันบนแอนดรอยด์นั้นเพิ่มขึ้นอย่างรวดเร็ว ซึ่งในชุด SDK นั้นจะมีโปรแกรมและไลบรารี่ต่างๆ ที่จำเป็นต่อการพัฒนาแอปพลิเคชั่นบนแอนดรอยด์ อย่างเช่น อีมูเลเตอร์ (Emulator) ซึ่งทำให้ผู้ใช้สามารถสร้างแอปพลิเคชันและนำมาทดลองรันบนตัวอีมูเลเตอร์ก่อน โดยมีสภาวะแวดล้อมเหมือนสมาร์ทโฟนที่รันระบบปฏิบัติการแอนดรอยด์จริง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

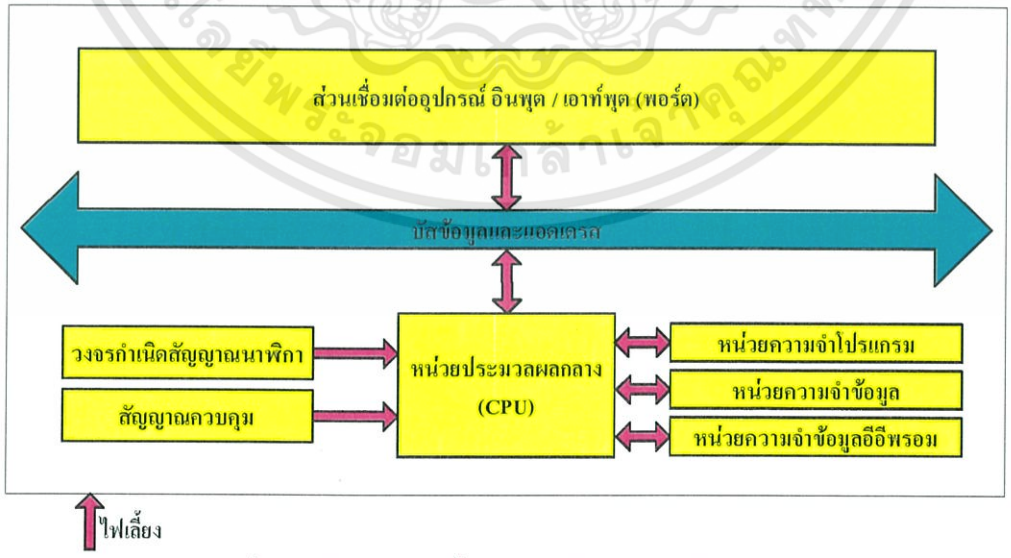
2.6 ชุดหุ่นยนต์ POP-BOT XT Lite

2.6.1 ไมโครคอนโทรลเลอร์

ไมโครคอนโทรลเลอร์ (Microcontroller) มาจากคำ 2 คำ คือ ไมโคร (Micro) หมายถึง ขนาดเล็ก และคำว่า คอนโทรเลอร์ (Controller) หมายถึง ตัวควบคุมหรืออุปกรณ์ควบคุม ดังนั้นไมโครคอนโทรลเลอร์ จึงหมายถึง อุปกรณ์ควบคุมขนาดเล็ก ซึ่งเป็นอุปกรณ์ประเภทสารกึ่งตัวนำ มีหน้าที่เป็นหน่วยประมวลผลกลาง โดยรวบรวมคำสั่งการทำงานต่างๆไว้ในตัวเอง ทำให้สามารถโปรแกรมให้ทำงานตามเงื่อนไขที่กำหนด และสามารถสร้างสัญญาณส่งออกหรือรับสัญญาณเข้าเพื่อติดต่อกับอุปกรณ์ต่างๆภายนอกได้ มักถูกนำไปเป็นหน่วยประมวลผลภายในอุปกรณ์อิเล็กทรอนิกส์สำเร็จรูปต่าง ๆ

2.6.1.1 สถาปัตยกรรมของไมโครคอนโทรลเลอร์

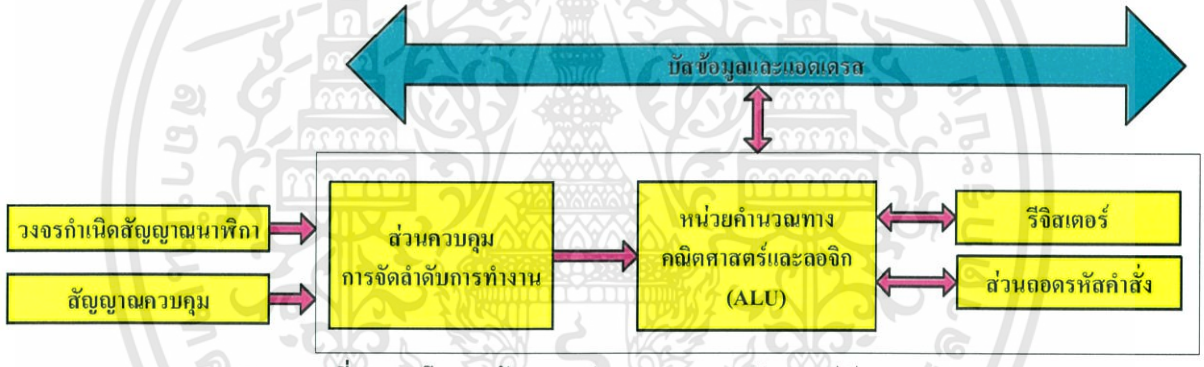
ไมโครคอนโทรลเลอร์เป็นอุปกรณ์พิเศษที่รวมซีพียู, หน่วยความจำ, และพอร์ต อินพุต/เอาท์พุต บรรจุอยู่ภายใต้ตัวถังเดียวกัน สามารถทำงานได้เมื่อจ่ายไฟเลี้ยงและต่อวงจรกำเนิดสัญญาณนาฬิกา จากนั้นซีพียูจะติดต่อกับหน่วยความจำโปรแกรมเพื่ออ่านคำสั่งที่ระบุไว้ โดยทำการอ้างตำแหน่งของหน่วยความจำผ่านสายสัญญาณที่เรียกว่า บัสแอดเดรส (Address Bus) แล้วทำการอ่านข้อมูลคำสั่งออกมาจากหน่วยความจำโปรแกรมในแอดเดรสนั้นๆ จากนั้นทำการประมวลผล โดยมีหน่วยความจำข้อมูลแรมเป็นที่พักของข้อมูลที่อยู่ระหว่างการประมวลผล ข้อมูลในการประมวลผลจะส่งผ่านสายสัญญาณที่เรียกว่า บัสข้อมูล (Data Bus) แล้วส่งต่อไปยังอุปกรณ์ภายนอกผ่านทางขาพอร์ตอินพุต เอาท์พุต ไมโครคอนโทรลเลอร์ประกอบด้วย



รูปที่ 2.10 โครงสร้างพื้นฐานของไมโครคอนโทรลเลอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. หน่วยประมวลผลกลางซีพียู (CPU : Central Processing Unit) : ซีพียูทำหน้าที่ประมวลผลข้อมูลที่เข้ามาในระบบ แล้วทำการส่งต่อไปยังส่วนต่างๆเพื่อควบคุมการทำงานต่อไป จากรูป 2.5 จะเห็นว่าหัวใจหลักของซีพียูคือหน่วยคำนวณทางคณิตศาสตร์และลอจิก (ALU: Arithmetic and Logic Unit) ซึ่งได้รับการกำหนดจังหวะการทำงานจากส่วนควบคุมลำดับการทำงาน โดยจังหวะการทำงานจะสัมพันธ์กับสัญญาณนาฬิกา เมื่อซีพียูทำการติดต่อหน่วยความจำ สิ่งที่ปรากฏขึ้นบนบัสข้อมูลภายในซีพียูคือ รหัสคำสั่ง (Instruction Code) ซึ่งต้องผ่านการทำงานของส่วนถอดรหัสคำสั่ง (Instruction Decode) เพื่อแปลงเป็นข้อมูลคำสั่งที่ซีพียูเข้าใจและสามารถดำเนินการต่อได้ หลังจากที่หน่วยคำนวณทางคณิตศาสตร์และลอจิกประมวลผลแล้ว จะส่งข้อมูลมายังส่วนเชื่อมต่อรีจิสเตอร์ภายในซีพียู เพื่อติดต่อกับส่วนอื่นๆ ต่อไป



รูปที่ 2.11 โครงสร้างและส่วนประกอบหลักของซีพียู

การทำงานของซีพียูมี 2 จังหวะคือ เฟตช์ (Fetch) และ เอ็กซีคิวต์ (Execute) โดยการทำงานจะเริ่มจากการเฟตช์ คือการเรียกหรือการเข้าถึงคำสั่ง และทำการถอดรหัสเป็นภาษาเครื่องเพื่อเตรียมประมวลผล จากนั้นจะเป็นจังหวะเอ็กซีคิวต์ คือการกระทำตามคำสั่งที่กำหนดให้เสร็จสิ้น

2. หน่วยความจำ : ไมโครคอนโทรเลอร์มีหน่วยความจำ 2 ประเภท คือ

   - หน่วยความจำอีอีพรอม (EEPROM Memory) หรือ หน่วยความจำโปรแกรม (Program Memory) เป็นที่เก็บโปรแกรมและข้อมูลสำหรับการประมวลผลที่ไม่ต้องการให้สูญหายในกรณีที่ไม่มีไฟเลี้ยงวงจร มีการแบ่งออกเป็นส่วนๆ เรียกว่า เพจ เพื่อเก็บโปรแกรมสำหรับทำงานที่แตกต่างและไม่เกี่ยวข้องกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- หน่วยความจำข้อมูลแรม (RAM Data Memory) แบ่งเป็น 2 กลุ่ม คือ
  - หน่วยความจำข้อมูลสำหรับเก็บค่าตัวแปร (Variable RAM) หรือ รีจิสเตอร์ (Register) ใช้เก็บค่าตัวแปรเพื่อทำการประมวลผล และข้อมูลทั่วไปชั่วคราว
  - หน่วยความจำแรมสแครตช์แพด (Scratch pad RAM) เป็นที่เก็บข้อมูลชั่วคราวกรณีที่มีการกระโดดข้ามไปทำงานในโปรแกรมที่อยู่ต่างเพจกัน ซึ่งการเก็บข้อมูลจะเก็บในลักษณะของสแต็ก (Stack) และในการดึงข้อมูลจะเป็นแบบ FIFO

ข้อมูลในหน่วยความจำจะหายไป เมื่อไม่มีการจ่ายไฟให้แก่ไมโครคอนโทรเลอร์ และเมื่อมีการจ่ายไฟให้ ค่าต่างๆ ในหน่วยความจำจะกลายเป็น 0

3. ส่วนติดต่ออุปกรณ์ภายนอก : ส่วนติดต่ออุปกรณ์ภายนอกหรือเรียกว่าพอร์ต (Port) มีด้วยกัน 2 ลักษณะคือ พอร์ตรับสัญญาณหรือพอร์ตอินพุต (Input Port) และพอร์ตส่งสัญญาณหรือพอร์ตเอาต์พุต (Output Port) ส่วนนี้มีความสำคัญมาก เนื่องจากใช้ในการเชื่อมต่อกับอุปกรณ์ภายนอก ที่เป็นสื่อกลางในการติดต่อกับมนุษย์ ยกตัวอย่าง พอร์ตอินพุตใช้ต่อกับสวิตช์เพื่อรับข้อมูลที่ผู้ใช้งานกดป้อนเข้ามา หรือต่อกับหน่วยความจำเพื่อเพิ่มพื้นที่ในการเก็บข้อมูล ส่วนพอร์ตเอาต์พุตก็คือส่วนที่ต่อกับจอแสดงผลแอลซีดี (LCD) เป็นต้น

4. บัสข้อมูลและแอดเดรส : การติดต่อแลกเปลี่ยนสัญญาณข้อมูลระหว่างซีพียู หน่วยความจำและพอร์ต จะกระทำบนสายสัญญาณจำนวนมาก เรียกว่า เส้นทางสัญญาณหรือบัส โดยแบ่งเป็นบัสข้อมูล (Data Bus), บัสแอดเดรส (Address Bus) และบัสควบคุม (Control Bus)

   - บัสข้อมูลเป็นสายสัญญาณที่บรรจุข้อมูลสำหรับการประมวลผลทั้งหมดขนาดของบัสจะขึ้นอยู่กับความสามารถในการประมวลผลของซีพียูและเทคโนโลยีของไมโครคอนโทรลเลอร์ตัวนั้นๆ
   - บัสแอดเดรสเป็นสายสัญญาณที่บรรจุค่าตำแหน่งของหน่วยความจำ โดยการติดต่อกับหน่วยความจำนั้น ซีพียูต้องกำหนดตำแหน่งที่ต้องการอ่านหรือเขียนก่อน ซึ่งก็คือการกำหนดค่าแอดเดรส จำนวนสายสัญญาณของบัสแอดเดรส จึงต้องมีจำนวนมาก และถ้ายิ่งมีมากเท่าใด จะเป็นการแสดงถึงความจุของหน่วยความจำ ที่ไมโครคอนโทรลเลอร์ตัวนั้นสามารถติดต่อได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- บัสควบคุมเป็นกลุ่มของสายสัญญาณควบคุมการติดต่อทั้งหมดของซีพียูกับหน่วยความจำและพอร์ต สำหรับสายสัญญาณควบคุมหลักได้แก่ สายสัญญาณเลือกอ่าน-เขียนหน่วยความจำ สายสัญญาณเลือกอ่าน-เขียนข้อมูลกับพอร์ต

5. วงจรกำเนิดสัญญาณนาฬิกา : การทำงานทั้งหมดในไมโครคอนโทรลเลอร์จะขึ้นอยู่กับการกำหนดจังหวะโดยใช้สัญญาณนาฬิกาหากสัญญาณนาฬิกามีความถี่สูงจังหวะในการทำงานของไมโครคอนโทรลเลอร์ก็จะถี่และมีมากตามส่งผลให้ไมโครคอนโทรลเลอร์นั้นมีความเร็วในการประมวลผลสูงตามไปด้วย

2.6.2 ชุดหุ่นยนต์ POP-BOT XT Lite [5]

รูปที่ 2.12 ลักษณะของชุดหุ่นยนต์ POP-BOT XT Lite

POP-XT เป็นแผงวงจรไมโครคอนโทรลเลอร์ที่ใช้ไมโครคอนโทรลเลอร์ตระกูล AVR เบอร์ ATmega32U4 ของ Atmel (www.atmel.com) มีส่วนเชื่อมต่อพอร์ต USB เพื่อใช้ในการสื่อสารข้อมูลและดาวน์โหลดโปรแกรมได้ในตัวโดยไม่ต้องใช้สายสัญญาณหรืออุปกรณ์แปลงสัญญาณใดๆเพิ่มเติมจึงทำให้การใช้งานง่ายและสะดวกมาก รวมถึง POP-XT ได้เลือกใช้ฮาร์ดแวร์ที่เข้ากันได้กับฮาร์ดแวร์ของโครงการไมโครคอนโทรลเลอร์ระบบเปิด (โอเพนซอร์ส: Open Source) ที่ชื่อ Arduino (www.arduino.cc/en) ใน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

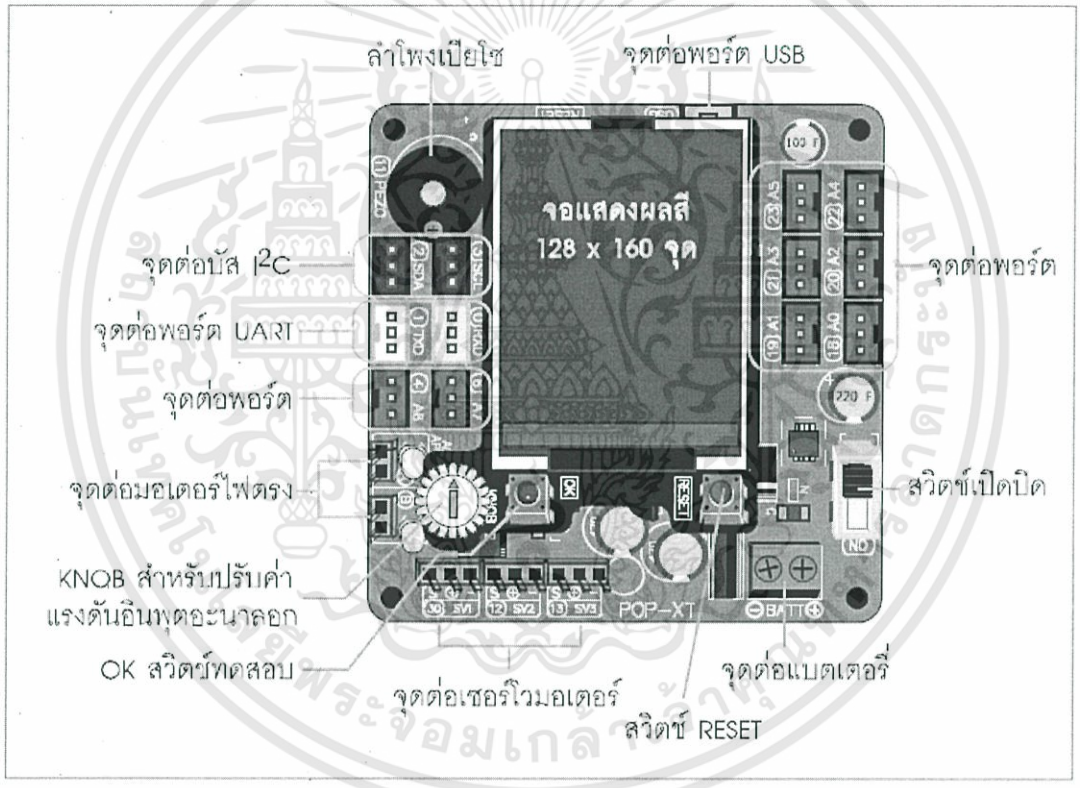
รุ่น Arduino Leonardo จึงทำให้สามารถนำชุดพัฒนาของ Arduino1.0 มาใช้งานได้ภายในชุดพัฒนาของ Arduino1.0 มีไลบรารีฟังก์ชันภาษาซีสำหรับติดต่อกับฮาร์ดแวร์จำนวนมากไว้ ให้ทำเขียนโปรแกรมสั่งงานอุปกรณ์ต่างๆได้ง่ายโดยไม่จำเป็นต้องศึกษาลงไปในรายละเอียดของไมโครคอนโทรลเลอร์มากนัก แต่ถ้าหากมีความต้องการพัฒนาในระดับที่สูงขึ้นก็สามารถนำ POP-XT ไปใช้ ร่วมกับเครื่องมือพัฒนาโปรแกรมรวมถึงคอมไพเลอร์อื่นๆ ได้เช่นกัน

2.6.2.1 คุณสมบัติของชุดหุ่นยนต์ POP-BOT XT Lite

- ใช้ไมโครคอนโทรลเลอร์ 8 บิตเบอร์ ATmega32U4 ของ Atmel ภายในมีโมดูลแปลงสัญญาณอนาลอกเป็นดิจิตอลความละเอียด 10 บิต 12 ช่อง มีหน่วยความจำโปรแกรมแบบแฟลช 32 กิโลไบต์ โปรแกรมซ้ำได้ 10,000 ครั้ง มีหน่วยความจำข้อมูลอีอีพรอม 1024 ไบต์ หน่วยความจำข้อมูลแรม 2.5 กิโลไบต์ หรือ 2,560 ไบต์ ใช้สัญญาณนาฬิกา 16MHz จากเซรามิกเรโซเนเตอร์
- จุดต่อพอร์ตใช้งาน 25 จุด และจัดสรรเป็นจุดต่อแบบ JST 3 ขา จำนวน 12 จุด สำหรับต่ออุปกรณ์ตรวจจับและอุปกรณ์ต่อพ่วงภายนอกต่างๆ
- มีไฟแสดงสถานะไฟเลี้ยงและไฟทดสอบพอร์ต
- มีสวิตช์ RESET
- มีจุดต่อพอร์ต USB สำหรับดาวน์โหลดโปรแกรมและสื่อสารข้อมูลกับคอมพิวเตอร์
- มีจุดต่อไฟเลี้ยง (DC INPUT) ผ่านทางจุดต่อสายแบบขันสกรู สามารถรับไฟเลี้ยงได้ตั้งแต่ 4.8 ถึง 7.2V
- มีวงจรควบคุมไฟเลี้ยง +5V แบบสวิตชิ่งสำหรับรักษาระดับไฟเลี้ยงให้แก่ไมโครคอนโทรลเลอร์
- จุดต่อพอร์ตอินพุตเอาต์พุตดิจิตอลหรืออนาลอก 8 ช่อง คือ A0 ถึง A7 (ตรงกับขา 18 ถึง 23 สำหรับขา A0 ถึง A5, 4 และ 6 สำหรับขา A6 และ A7)
- จุดต่อพอร์ตดิจิตอลรองรับระบบบัส I^2C 1 ชุด คือ จุดต่อ 2 (SDA) และ 3 (SCL)
- มีจุดต่อพอร์ตสื่อสารข้อมูลอนุกรม UART 1 ชุดคือ จุดต่อ 0 (RxD) และ 1 (TxD)
- มีวงจรขับมอเตอร์ไฟตรง 2 ช่อง พร้อมไฟแสดงผล
- มีจุดต่อขาพอร์ตของไมโครคอนโทรลเลอร์สำหรับขับเซอร์โวมอเตอร์ 3 ช่อง

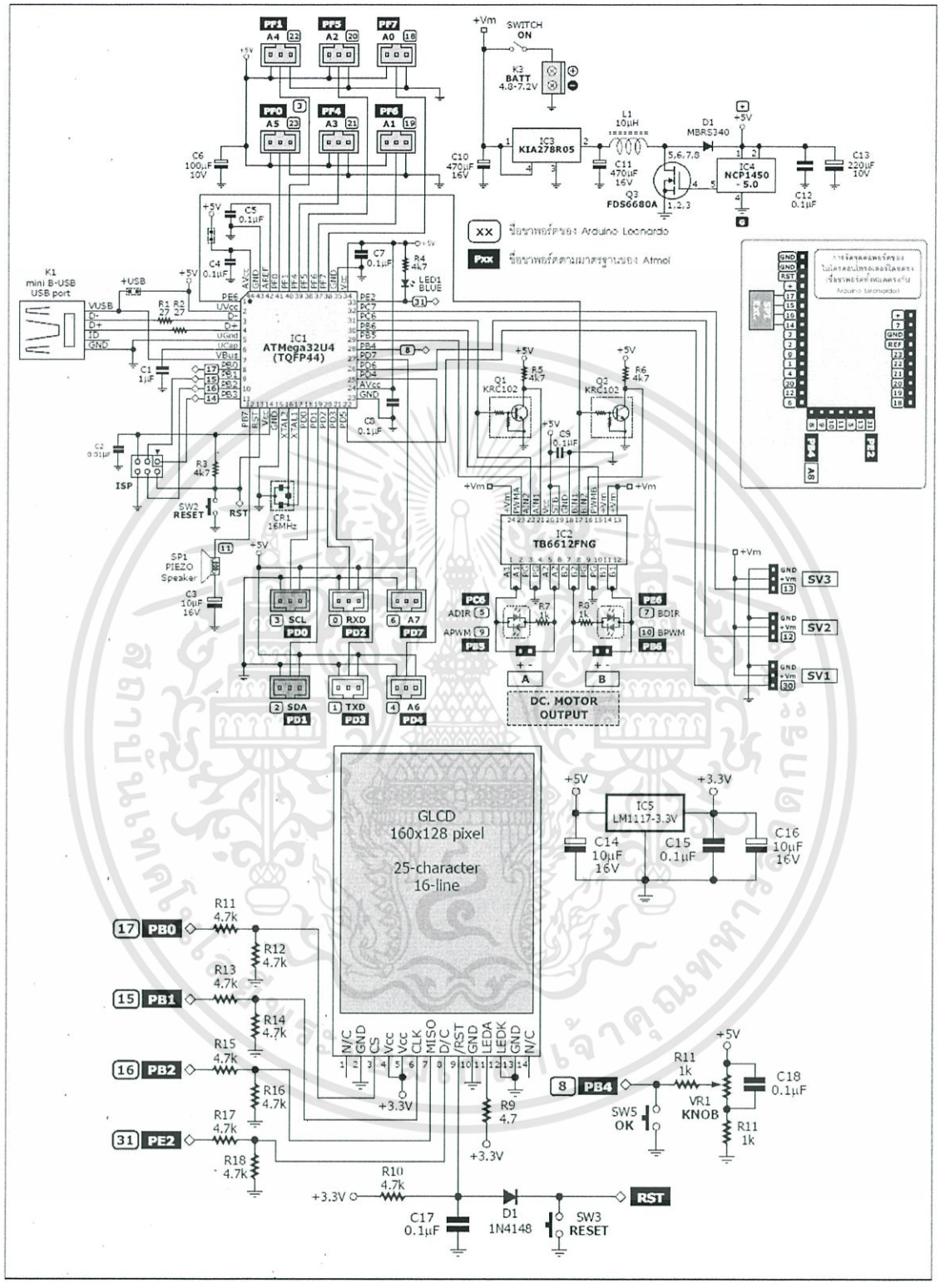
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- มีลำโพงเปียโซสำหรับขับเสียง
- มีจุดต่อ ISP สำหรับอัพเกรดเฟิร์มแวร์ โดยใช้ชุดโปรแกรมแบบ ISP เพิ่มเติม
- มีโมดูลแสดงผลแบบกราฟฟิกสีความละเอียด 128x160 จุด แสดงภาพกราฟิกลายเส้นและสีพื้น (ไม่รองรับไฟล์รูปภาพใดๆ) พร้อมไฟส่องหลัง แสดงผลเป็นตัวอักษรขนาดปกติ (5x7 จุด) ได้ 21 ตัวอักษร 16 บรรทัด (21x16)
- มีสวิตช์กดติดปล่อยดับพร้อมใช้งาน (สวิตช์ OK) 1 จุด โดยต่อร่วมกับตัวต้านทานปรับค่าได้ (KNOB) ซึ่งเชื่อมต่อไปยังขาพอร์ต 8 ทำให้อ่านค่าสัญญาณดิจิตอลและอนาลอกได้ในขาพอร์ตเดียวกัน



รูปที่ 2.13 แสดงส่วนประกอบของแผงวงจรควบคุม POP-BOT XT

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.14 วงจรสมบูรณ์ของแผงวงจรควบคุม POP-BOT XT

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

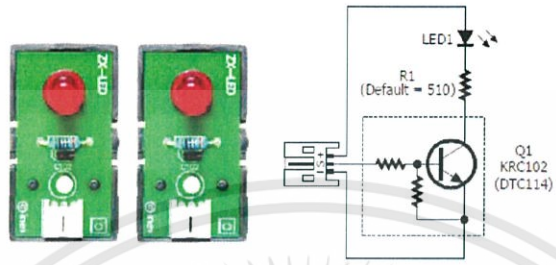
2.6.2.2 คุณสมบัติของอุปกรณ์เอาต์พุต

- มอเตอร์ไฟตรงพร้อมชุดเฟืองขับ เป็นชุดมอเตอร์พร้อมเฟืองขับรุ่น BO-2 อัตราทด48:1 มีสายต่อ 2 เส้นคุณสมบัติทางเทคนิคที่สำคัญมีดังนี้
  - ต้องการไฟเลี้ยงในย่าน +3 ถึง +9 Vdc
  - กินกระแสไฟฟ้า 130mA (ที่ไฟเลี้ยง +6V และไม่มีโหลด)
  - ความเร็วเฉลี่ย 170 ถึง 250 รอบต่อนาที (ที่ไฟเลี้ยง +6V และไม่มีโหลด)
  - น้ำหนัก 30 กรัม
  - แรงบิดต่ำสุด 0.5 กิโลกรัม-เซนติเมตร
  - ติดตั้งเข้ากับตัวยึดพลาสติกแบบมีพุกทองเหลืองสำหรับยึดในตัว
  - ขนาด(กว้าง x ยาว x สูง) 42 x 45 x 22.7 มิลลิเมตร

รูปที่ 2.15 มอเตอร์ไฟตรงพร้อมชุดเฟืองขับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

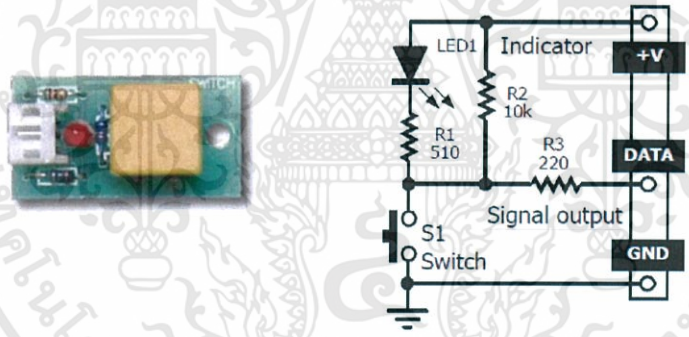
- แผงวงจรไฟแสดงผล : ZX-LED ใช้ LED ขนาด 8 มิลลิเมตรต้องการลอจิก “1” ในการขับให้สว่างมีวงจรแสดงในรูปที่ 2.10



รูปที่ 2.16 รูปร่างและวงจรของแผงวงจรไฟแสดงผล ZX-LED

2.6.2.3 คุณสมบัติของชุดอุปกรณ์ตรวจจับสัญญาณ

- แผงวงจรสวิตช์ : ZX-01 ประกอบด้วยสวิตช์พร้อมไฟแสดงผล ถ้ากดสวิตช์จะส่งลอจิก “0” ไฟสีแดงติด



รูปที่ 2.17 รูปร่างและวงจรของแผงวงจรสวิตช์ ZX-01

- แผงวงจรตรวจจับแสงสะท้อนอินฟราเรด : ZX-03

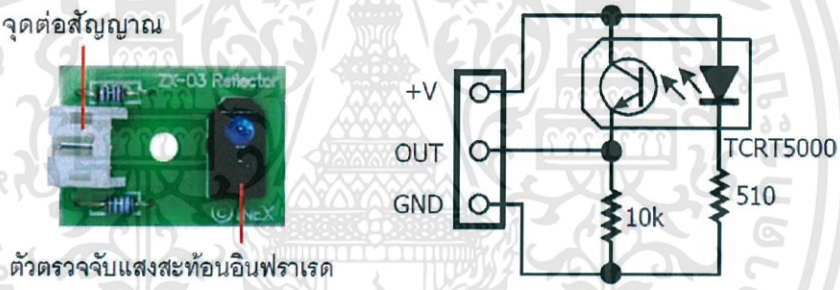
เป็นแผงวงจรที่ใช้ในการตรวจสอบการสะท้อนของแสงอินฟราเรดของตัวตรวจจับแสงสะท้อนอินฟราเรดซึ่งรวมตัวส่งและตัวรับไว้ในตัวถังเดียวกัน โดยตัวตรวจจับแสงสะท้อนอินฟราเรดที่นำมาใช้คือ TCRT5000

เมื่อจ่ายไฟเลี้ยง LED อินฟราเรดภายในตัวโมดูล TCRT5000 จะเปล่งแสงออกมาตลอดเวลาส่วนตัวรับซึ่งเป็นโฟโต้ทรานซิสเตอร์จะได้รับแสงอินฟราเรดจากการสะท้อนกลับ โดยปริมาณของแสงที่ได้รับจะมากหรือน้อยขึ้นอยู่กับว่ามีวัตถุมากีดขวางหรือไม่และวัตถุนั้นมี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ความสามารถในการสะท้อนแสงอินฟราเรดได้ดี เพียงไรซึ่งขึ้นกับลักษณะพื้นผิวและสีของวัตถุ โดยวัตถุสีขาวผิวเรียบจะสะท้อนแสงอินฟราเรดได้ดี ทำให้ตัวรับแสงอินฟราเรดได้รับแสงสะท้อนมากส่งผลให้แรงดันที่เอาต์พุตของวงจรสูงตามไปด้วย ในขณะที่วัตถุสีดำสะท้อนแสงอินฟราเรดได้น้อย ทำให้ตัวรับอินฟราเรดส่งแรงดันออกมาต่ำ ด้วยคุณสมบัติดังกล่าวจึงนิยมนำแผงวงจรตรวจจับแสงสะท้อนอินฟราเรดนี้มาใช้ในการตรวจจับพื้นหรือเส้นโดยต้องติดตั้งไว้ด้านล่างของโครงหุ่นยนต์

เนื่องจากแผงวงจรตรวจจับแสงสะท้อนอินฟราเรด ZX-03 ให้ผลการทำงานเป็นแรงดันไฟตรง ดังนั้นในการใช้งานกับ POP-BOT XT จึงต้องต่อเข้ากับช่องอินพุตอนาลอกของแผงวงจรหลัก จากนั้นใช้ความรู้จากการอ่านค่าสัญญาณอนาลอกเพื่ออ่านค่าจากแผงวงจรตรวจจับแสงสะท้อนอินฟราเรดนำไปสู่การตรวจจับเส้นต่อไป



รูปที่ 2.18 รูปร่างและวงจรของแผงวงจรตรวจจับแสงสะท้อนอินฟราเรด : ZX-03

2.6.2.4 ข้อมูลของสายสัญญาณที่ใช้ในชุดหุ่นยนต์ POP-BOT XT

- สาย JST3AA-8 ใช้เชื่อมต่อระหว่างแผงวงจรควบคุม POP-XT กับแผงวงจรตรวจจับและแผงวงจรอุปกรณ์ต่างๆ เป็นสายแพ 3 เส้น ยาว 8 นิ้ว ปลายสายทั้งสองด้านติดตั้งคอนเน็กเตอร์ (Connector) แบบ JST 3 ขา ตัวเมีย ระยะห่างระหว่างขา 2 มิลลิเมตร

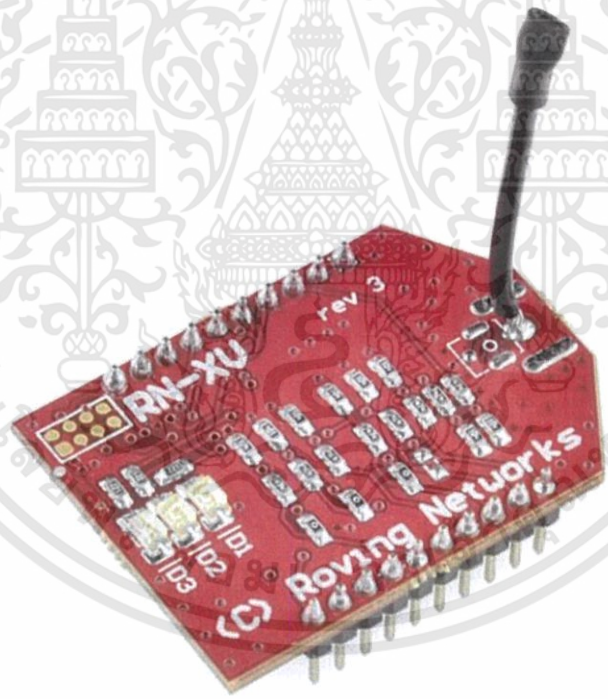
รูปที่ 2.19 สาย JST3AA-8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- สาย USB-miniB เป็นสายสัญญาณสำหรับเชื่อมต่อระหว่างพอร์ต USB ของคอมพิวเตอร์กับแผงวงจร POP-BOT XT

รูปที่ 2.20 สาย USB-miniB

2.7 ชุดอุปกรณ์ RN-XV WiFly Module - Wire Antenna [8]



รูปที่ 2.21 RN-XV WiFly Module - Wire Antenna

RN-XV WiFly Module หรือเรียกสั้นๆ ว่า WiFly เป็นโมดูลสำหรับเชื่อมต่อสัญญาณ Wi-Fi มาตรฐาน 802.15.4 และ 802.11b/g ประมวลผลด้วยระบบ 32 บิต ช่วยให้จัดการกับการเชื่อมต่อผ่านระบบ Wi-Fi ได้ง่าย โดยตัวโมดูลมีลักษณะคล้ายกับ XBee โมดูลสื่อสารข้อมูลอนุกรมไร้สายผ่านคลื่นวิทยุ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4GHz จึงใช้ซ็อกเก็ต (Socket) หรือแผงวงจรเชื่อมต่อร่วมกันได้ WiFly ใช้การเชื่อมต่อแบบ UART จึงทำให้นำ WiFly ไปใช้แทนที่ XBee ได้ทันที ทำให้ระบบหรือบอร์ดไมโครคอนโทรลเลอร์เดิมที่ใช้ XBee เปลี่ยนระบบมาเป็นการเชื่อมต่อผ่าน Wi-Fi ได้ทันที โดยไม่ต้องดัดแปลงใดๆ ทั้งสิ้น

2.7.1 คุณสมบัติของ RN-XV WiFly Module - Wire Antenna

- เชื่อมต่อสัญญาณ Wi-Fi มาตรฐาน 802.15.4
- ใช้กระแสไฟฟ้า 38mA ในขณะทำงาน และ 4 uA เมื่ออยู่ในโหมดสลีป
- WiFly มี TCP/IP Stack อยู่ในตัว รองรับการใช้งาน DHCP, UDP, DNS, ARP, ICMP, HTTP Client, FTP Client และ TCP
- กำหนดกำลังส่งได้ระหว่าง 0 dBm ถึง 12 dBm
- เชื่อมต่อโดยพอร์ต TTL UART
- กำหนดอัตราเร็วในการสื่อสารข้อมูลผ่าน UART ได้สูงสุดถึง 464 กิโลบิตต่อวินาที
- รองรับการทำงานแบบ Adhoc และ Network Infrastructure
- พอร์ตอินพุตเอาต์พุตดิจิตอล 8 ขา
- พอร์ตอินพุตเอาต์พุตอนาล็อก 3 ขา
- Real time clock ในตัวสำหรับการใช้งานในโหมด Time-stamping, Auto-sleep และ Auto-wake
- ใช้ Power Supply 3.3 VDC

2.7.2 Configuration

WiFly module จะมี 2 รูปแบบการทำงาน คือ โหมดข้อมูล (data mode) และโหมดคำสั่ง (command mode) โดยโหมดข้อมูลเป็นโหมดที่พร้อมกับการเชื่อมต่อหรือยกเลิกการเชื่อมต่อการตั้งค่าพารามิเตอร์หรือดูค่าที่ตั้งไว้อยู่แล้ว WiFly module ต้องเข้าสู่โหมดการทำงานในเวลาใดก็ตาม WiFly module จะอยู่ในโหมดคำสั่งหรือโหมดข้อมูล

1. คำสั่งสำหรับเข้าสู่ Command Mode

เมื่อตัว wiFly เริ่มทำงานจะอยู่ในโหมดข้อมูล หากต้องการเข้าสู่โหมดคำสั่งต้องพิมพ์สัญลักษณ์พิเศษ (กำหนด $$$ เป็นค่าเริ่มต้น) โดยอุปกรณ์จะมีการตอบสนองกับ CMD เพื่อแสดงให้เห็นว่าอยู่ในโหมดคำสั่งแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยคำสั่งออกจาก Command Mode คำสั่งนี้จะใช้ในการออกจาก Command Mode โดยการป้อนคำสั่ง exit และตามด้วยกดปุ่ม Enter จากนั้นโมดูลจะส่งข้อความ EXIT เพื่อให้รู้ว่าได้ออกจาก Command Mode เรียบร้อยแล้ว

2. คำสั่งสำหรับตั้งค่า IP address

รูปแบบคำสั่งคือ set ip address <addr> ในกรณีที่ต้องการตั้งค่า IP address ทำได้โดยป้อนคำสั่ง set ip address ตามด้วย IP address ที่ต้องการและตามด้วยกดปุ่ม Enter จากนั้นโมดูลจะส่งข้อความ AOK ตอบกลับมา

ตัวอย่าง : set ip a 192.168.1.1

3. คำสั่งสำหรับตั้งค่า Baudrate

รูปแบบของของคำสั่งนี้คือ set uart baud <rate> ตามด้วยค่า Baudrate ที่ต้องการ (2400, 4800, 9600, 19200, 38400, 57600, 115200, 230400, 460800, 921600) และตามด้วยกดปุ่ม Enter จากนั้นโมดูลจะส่งข้อความ AOK ตอบกลับมา

ตัวอย่าง : set uart baud 9600

4. คำสั่งสำหรับเปิดปิดการใช้งาน DHCP Mode

เปิด/ปิด โหมด DHCP ถ้าเปิด ip address, gateway, netmask และ DNS เซิฟเวอร์ จะถูกร้องขอและตั้งค่าเมื่อมีการเชื่อมต่อกับ access point ทุกค่าของ ip จะถูกเขียนทับ รูปแบบของคำสั่งคือ set ip dhcp <value> โดยค่าแต่ละค่าจะแตกต่างกันไป เลข 0 คือปิด DHCP เลข 1 คือเปิด DHCP เลข 2 คือ ออโต้ไอพีโดยทั่วไปจะใช้กับ ADHOC เลข 3 คือ แคชโหมดจะใช้ไอพีล่าสุด เลข 4 คือ สำรองไว้ใช้ในอนาคต

ตัวอย่าง : set ip dhcp 0

5. คำสั่งสำหรับตั้งค่า netmask

ตั้งค่าเครือข่าย mask ถ้า DHCP เปิดอยู่ net mask จะถูกกำหนดค่าและเขียนทับระหว่างเชื่อมต่อกับ access point รูปแบบของคำสั่งคือ set ip netmask <addr> ในกรณีที่ต้องการตั้งค่า netmask ทำได้โดยป้อนคำสั่ง set ip netmask ตามด้วยค่า netmask ที่ต้องการและตามด้วยกดปุ่ม Enter จากนั้นโมดูลจะส่งข้อความ AOK ตอบกลับมา

ตัวอย่าง : set ip netmask 255.255.255.0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6. คำสั่งสำหรับตั้งค่า gateway

รูปแบบของคำสั่งคือ set ip gateway <addr> ในกรณีที่ต้องการตั้งค่า gateway ทำได้โดยป้อนคำสั่ง set ip gateway ตามด้วยค่า gateway ที่ต้องการและตามด้วยกดปุ่ม Enter จากนั้นโมดูลจะส่งข้อความ AOK ตอบกลับมา

ตัวอย่าง : set ip gateway 192.168.1.1

7. คำสั่งสำหรับตั้งหมายเลขพอร์ตของโมดูล

รูปแบบของคำสั่ง set ip localport <num> คำสั่งนี้จะใช้ตั้งค่าหมายเลขพอร์ตที่ต้องการจะเชื่อมต่อด้วย โดยปกติแล้วค่ามาตรฐานจะใช้พอร์ตหมายเลข 2000 ในกรณีที่ผู้ใช้ต้องการเปลี่ยนไปใช้พอร์ตอื่น ก็ทำได้โดยป้อนคำสั่ง set ip localport ตามด้วยหมายเลขพอร์ตที่ต้องการตามด้วยกดปุ่ม Enter จากนั้นโมดูลจะส่งข้อความ AOK ตอบกลับมา

ตัวอย่าง : set ip localport 9999

8. คำสั่ง set wlan ssid <string>

รูปแบบของคำสั่ง set wlan ssid <string> คำสั่งนี้จะมี 2 หน้าที่ คือ ถ้าผู้ใช้เชื่อมต่อโมดูลในรูปแบบ Adhoc Mode คำสั่งนี้จะใช้สาหรับตั้งชื่อ SSID ของโมดูล แต่ถ้าเชื่อมต่อในรูปแบบ Infrastructure คำสั่งนี้จะใช้สำหรับเลือกชื่อ SSID ของ Access Point ที่ต้องการจะเชื่อมต่อด้วย เช่น ถ้าต้องการเชื่อมต่อกับ SSID ชื่อใดก็ทำได้โดยป้อนคำสั่ง set wlan ssid ตามด้วยชื่อ SSID ที่ต้องการ โดยใช้ตัวอักษรหรือตัวเลข 1-32 ตัว และกดปุ่ม Enter จากนั้นโมดูลจะส่งข้อความ AOK ตอบกลับมา

ตัวอย่าง : set wlan ssid yellow ในกรณีที่ชื่อ SSID มีการเว้นวรรค เช่น yellow brick road ให้แทนการเว้นวรรคด้วยตัวอักษร : $ เมื่อป้อนคำสั่งจะได้เป็น yellow$brick$road เป็นต้น

9. คำสั่งสำหรับตั้งค่า passphrase

รูปแบบของคำสั่ง set wlan phrase <string> คำสั่งนี้จะใช้สาหรับตั้งค่า passphrase ของ Access Point ซึ่งใช้ระบบรักษาความปลอดภัยแบบ WPA หรือ WPA2 ที่จะทำการเชื่อมต่อการใช้งาน คำสั่งนี้ทำได้โดยป้อนคำสั่ง set wlan phrase ตามด้วยตัวอักษรหรือตัวเลข 1-64 ตัว และกดปุ่ม Enter จากนั้นโมดูลจะส่งข้อความ AOK ตอบกลับมา

ตัวอย่าง : set wlan phrase 123456789

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

10. คำสั่งสำหรับการเลือกช่องสัญญาณ

รูปแบบของคำสั่ง set wlan channel <value> คำสั่งนี้จะมี 2 หน้าที่ คือ ถ้าผู้ใช้เชื่อมต่อโมดูลในรูปแบบ Adhoc Mode คำสั่งนี้จะใช้สำหรับตั้งช่องสัญญาณของโมดูล แต่ถ้าเชื่อมต่อในรูปแบบ Infrastructure คำสั่งนี้จะใช้สำหรับเลือกช่องสัญญาณที่ต้องการเชื่อมต่อด้วย ค่านี้จะมีให้เลือกตั้งแต่ช่อง 1-13 แต่ถ้าใช้ค่านี้เป็น 0 จะเป็นการสแกนหาทุกช่องสัญญาณ ในกรณีที่ต้องการเลือกช่องสัญญาณ ก็ทำได้โดยใช้คาสั่ง set wlan channel ตามด้วยช่องสัญญาณที่ต้องการ และกดปุ่ม Enter จากนั้นโมดูลจะส่งข้อความ AOK

ตัวอย่าง : set wlan channel 0

11. คำสั่งสำหรับเลือกรูปแบบการเชื่อมต่อ

รูปแบบของคำสั่ง set wlan join <value> คำสั่งนี้จะใช้เลือกรูปแบบการเชื่อมต่อกับ network ซึ่งค่าสามารถตั้งได้ 5 ค่า คือ

- 0 ค่านี้จะเป็นการเชื่อมต่อแบบ manual ผู้ใช้ต้องเลือกที่จะต่อกับ Access Point เองทุกครั้งที่เริ่มจ่ายไฟเข้าโมดูลหรือเริ่มการทำงานใหม่
- 1 ค่านี้จะเป็นการเชื่อมต่อกับ Access Point ตามค่า SSID ที่กำหนดแบบอัตโนมัติ ทุกครั้งที่เริ่มจ่ายไฟเข้าโมดูลหรือเริ่มการทำงานใหม่แนะนำให้ใช้ค่านี้
- 2 ค่านี้จะเป็นการให้โมดูลทำการเชื่อมต่อกับ Access Point ตัวที่มีสัญญาณแรงที่สุดทุกครั้งที่เริ่มจ่ายไฟเข้าโมดูลหรือเริ่มการทำงานใหม่
- 3 ค่านี้ยังไม่สามารถใช้งานได้
- 4 ค่านี้จะเป็นการให้โมดูลทำงานในรูปแบบ Adhoc Mode ตัวอย่างเช่น ถ้าต้องการเชื่อมต่อกับ Access Point ตามค่า SSID, IP Address และ netmask

ตัวอย่าง : set wlan join 0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.8 โปรแกรม Arduino

ในการเขียนโปรแกรมสำหรับแผงวงจร POP-XT และหุ่นยนต์ POP-BOT XT จะต้องเขียนโปรแกรมโดยใช้ภาษา C/C++ ของ Arduino (Arduino programming language) เวอร์ชัน 1.0 ขึ้นไปซึ่งตัวภาษาของArduino เองก็นำเอาโอเพนซอร์สโปรเจกต์ชื่อ wiring มาพัฒนาต่อ ภาษาของ Arduino แบ่งได้เป็น 2 ส่วนหลักคือ

- โครงสร้างภาษา (Structure) ตัวแปรและค่าคงที่
- ฟังก์ชั่น (Function)

ภาษาของ Arduino จะอ้างอิงตามภาษา C/C++ จึงอาจกล่าวได้ว่าการเขียนโปรแกรมสำหรับ Arduino (ซึ่งก็รวมถึง POP-MCU) ก็คือการเขียนโปรแกรมภาษาซี โดยเรียกใช้ฟังก์ชั่นและไลบรารีที่ทาง Arduino ได้เตรียมไว้ให้แล้วซึ่งสะดวกและทำให้ผู้ที่ไม่มีความรู้ด้านไมโครคอนโทรลเลอร์อย่างลึกซึ้งสามารถเขียนโปรแกรมสั่งงานได้

2.8.1 โครงสร้างโปรแกรมของ Arduino

โปรแกรมของ Arduino แบ่งได้เป็นสองส่วนคือ void setup() และ void loop() โดยฟังก์ชั่น setup() เมื่อโปรแกรมทำงานจะทำคำสั่งของฟังก์ชั่นนี้เพียงครั้งเดียวใช้ในการกำหนดค่าเริ่มต้นของการทำงานส่วนฟังก์ชั่น loop() เป็นส่วนทำงานโปรแกรมจะทำคำสั่งในฟังก์ชั่นนี้ต่อเนื่องกัน ตลอดเวลา โดยปกติใช้กำหนดโหมดการทำงานของขาต่างๆ กำหนดการสื่อสารแบบอนุกรม ฯลฯ ส่วนของ loop() เป็นโค้ดโปรแกรมที่ทำงานเช่น อ่านค่าอินพุตประมวลผลสั่งงานเอาต์พุต ฯลฯ โดยส่วนกำหนดค่าเริ่มต้นเช่นตัวแปรจะต้องเขียนที่ส่วนหัวของโปรแกรมก่อนถึงตัวฟังก์ชั่น นอกจากนั้นยังต้องคำนึงถึงตัวพิมพ์เล็ก-ใหญ่ของตัวแปรและชื่อฟังก์ชั่นให้ถูกต้อง

- **ส่วนของฟังก์ชัน setup()**

  ฟังก์ชั่นนี้จะเขียนที่ส่วนต้นของโปรแกรมทำงานเมื่อโปรแกรมเริ่มต้นเพียงครั้งเดียว ใช้เพื่อกำหนดค่าของตัวแปรโหมดการทำงานของขาต่างๆ เริ่มต้นเรียกใช้ไลบรารี่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่าง

```
int buttonPin = 3;
void setup()
{       beginSerial(9600);
        pinMode(buttonPin, INPUT);
}
void loop()
{       if (digitalRead(buttonPin) == HIGH)
                serialWrite('H');
        else
                serialWrite('L');
        delay(1000);
}
```

ในขณะที่โปรแกรมภาษา C มาตรฐานที่เขียนบน AVR GCC (เป็นโปรแกรมภาษา C ที่ใช้ C คอมไพเลอร์แบบ GCC สำหรับไมโครคอนโทรลเลอร์ AVR) จะเขียนได้ดังนี้

```
int main (void)
{
        init();
        setup();
        for ( ; ; )
                loop();
        return;
}
```

ตรงกับ void setup()

ตรงกับ void loop()

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- **ส่วนของฟังก์ชั่น loop()**

หลังจากที่เขียนฟังก์ชั่น setup() ที่กำหนดค่าเริ่มต้นของโปรแกรมแล้วส่วนถัดมาคือฟังก์ชั่น loop() ซึ่งมีการทำงานตรงตามชื่อคือจะทำงานตามฟังก์ชั่นนี้วนต่อเนื่องตลอดเวลา ภายในฟังก์ชั่นนี้จะมีโปรแกรมของผู้ใช้เพื่อรับค่าจากพอร์ตประมวลผลแล้วสั่งเอาต์พุตออกขาต่างๆ เพื่อควบคุมการทำงานของบอร์ด

ตัวอย่าง

```
int buttonPin = 3; // setup initializes serial and the button pin
void setup()
{
        beginSerial(9600);
        pinMode(buttonPin, INPUT);
}
// loop checks the button pin each time and will send serial if it is pressed
void loop()
{
        if (digitalRead(buttonPin) == HIGH)
                serialWrite('H');
        else
                serialWrite('L');
        delay(1000);
}
```

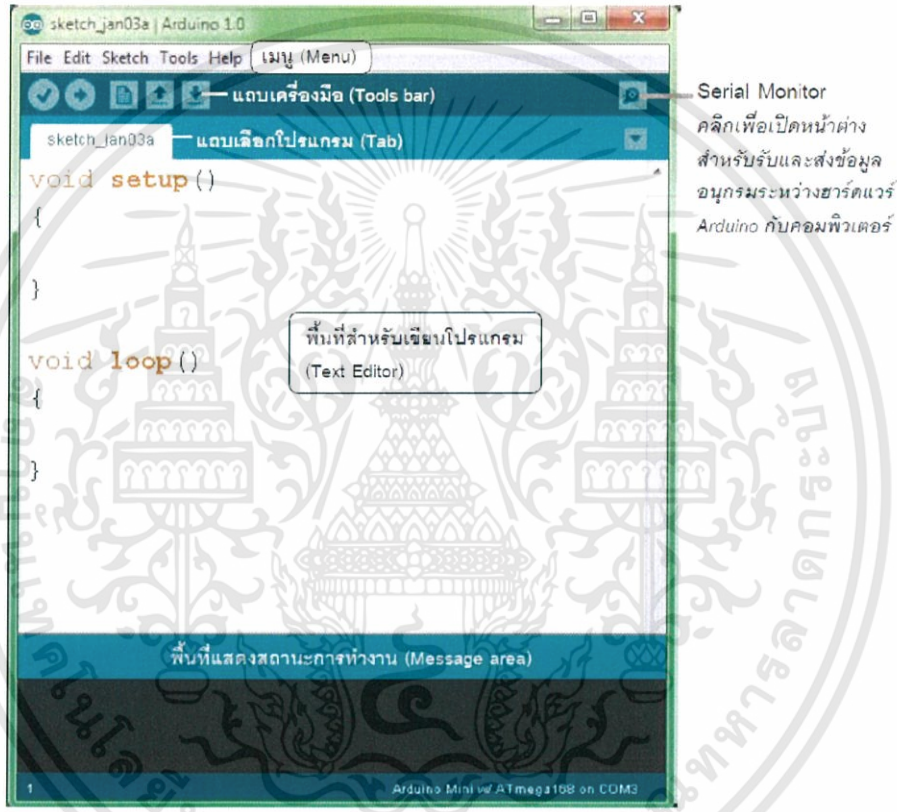
2.8.2 ส่วนประกอบของหน้าจอโปรแกรม Arduino1.0

เมื่อเรียกให้โปรแกรมทำงานจะมีหน้าตาดังรูปที่ 2.16 ตัวโปรแกรมประกอบด้วยส่วนต่างๆดังนี้

- **เมนู (Menu)** ใช้เลือกคำสั่งต่างๆ ในการใช้งานโปรแกรม
- **แถบเครื่องมือ (Toolbar)** เป็นการนำคำสั่งที่ใช้งานบ่อยๆ มาสร้างเป็นปุ่มเพื่อให้เรียกใช้คำสั่งได้รวดเร็วขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- **แถบเลือกโปรแกรม (Tabs)** เป็นแถบที่ใช้เลือกไฟล์โปรแกรมแต่ละตัว (กรณีที่เขียนโปรแกรมขนาดใหญ่ ประกอบด้วยไฟล์หลายตัว)
- **พื้นที่เขียนโปรแกรม (Text editor)** เป็นพื้นที่สำหรับเขียนโปรแกรมภาษา C/C++
- **พื้นที่แสดงสถานะการทำ งาน (Message area)** เป็นพื้นที่โปรแกรมใช้แจ้งสถานะการทำงานของโปรแกรม เช่น ผลการคอมไพล์โปรแกรม



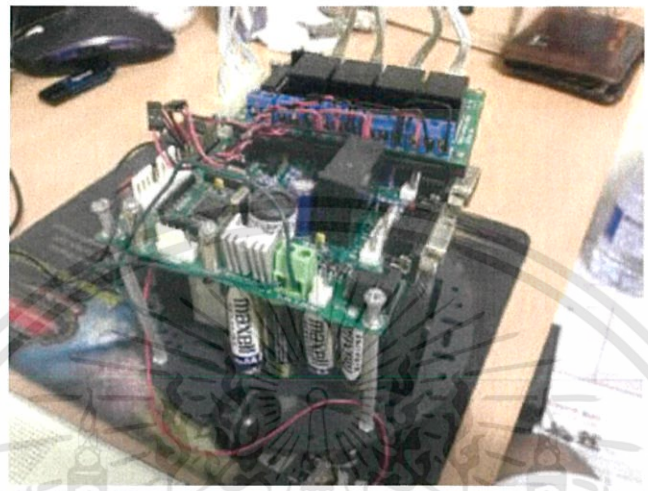
รูปที่ 2.22 แสดงส่วนประกอบของโปรแกรม Arduino1.0

- **พื้นที่แสดงข้อมูล** (Text area) ใช้แจ้งว่าโปรแกรมที่ผ่านการคอมไพล์แล้วมีขนาดกี่ไบต์
- **ปุ่มสำหรับเปิดหน้าต่าง Serial Monitor** ปุ่มนี้จะอยู่ทางมุมบนด้านขวามือ คลิกปุ่มนี้เมื่อต้องการเปิดหน้าต่างสื่อสารและแสดงข้อมูลอนุกรมโดยต้องมีการต่อฮาร์ดแวร์ Arduino และเลือกพอร์ตการเชื่อมต่อให้ถูกต้องก่อน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.9 บทความที่เกี่ยวข้อง

2.9.1 Designing and Implementing a WLAN Robot Car [4]



รูปที่ 2.23 รถหุ่นยนต์ของโครงงาน Designing and Implementing a WLAN Robot Car

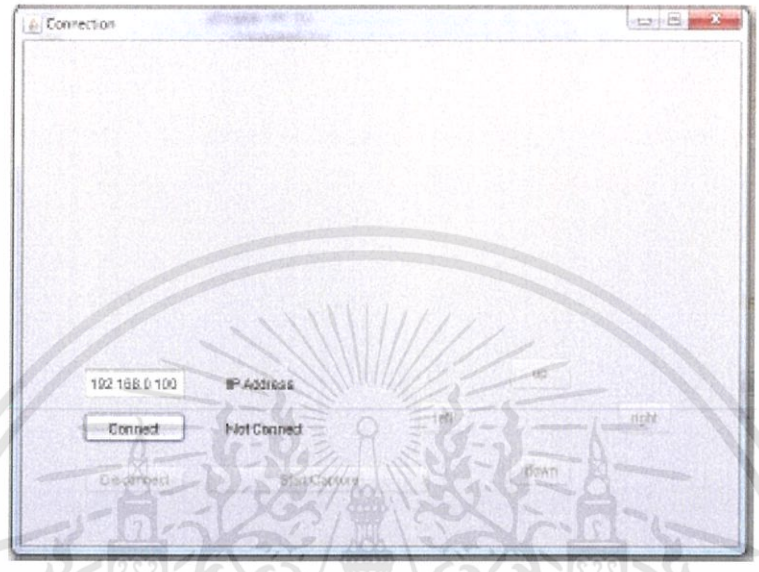
เป็นโครงงานที่นำเสนอเกี่ยวกับการบังคับรถหุ่นยนต์ด้วยเครื่องคอมพิวเตอร์ผ่านเทคโนโลยีไร้สาย Wi-Fi โดยใช้ไมโครคอนโทรลเลอร์ EB-7220 ซึ่งเป็นบอร์ดไมโครคอนโทรลเลอร์อเนกประสงค์ ใช้ชิพเบอร์ 82G516A ของ megawin ในตระกูล MCS-51 มีพอร์ท I/O รองรับการใช้งานได้หลากหลาย และสามารถขยายเพิ่มเติมได้

ส่วนของการเชื่อมต่อไร้สายใช้งาน ET-WiFly GSX ซึ่งเป็นอุปกรณ์ที่ใช้แปลงข้อมูลระหว่างระบบ Wireless LAN และ RS232 โดยด้านหนึ่งจะใช้การติดต่อสื่อสารด้วยพอร์ตอนุกรม (RS232) ส่วนอีกด้านหนึ่งจะสื่อสารผ่านระบบ WirelessLAN โดยที่ ET-WiFly GSX จะทำหน้าที่แปลงข้อมูลของการสื่อสารระหว่างอุปกรณ์ที่เป็น RS232 กับอุปกรณ์ที่เป็นระบบ Wireless LAN

รูปที่ 2.24 ET-WiFly GSX

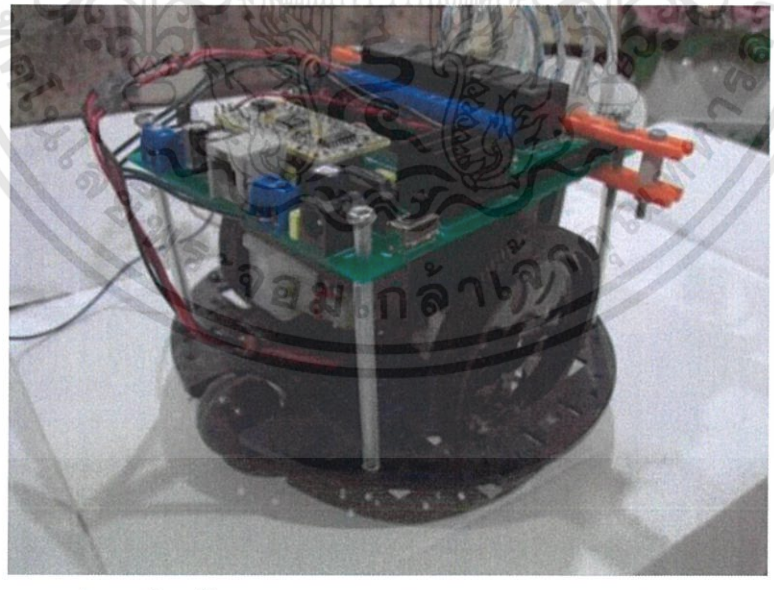
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- โดยใช้การเชื่อมต่อแบบ Infrastructure ซึ่งจำเป็นต้องมีแอคเซสพ้อยท์เป็นศูนย์กลางในการเชื่อมต่อ และการควบคุมรถหุ่นยนต์นั้นจะใช้งานคอมพิวเตอร์ระบบปฏิบัติการ Windows



รูปที่ 2.25 หน้าจอส่วนติดต่อกับผู้ใช้เพื่อควบคุมรถหุ่นยนต์

2.9.2 System Direction Control for Controller Car via Bluetooth on Smartphone [7]



รูปที่ 2.26 รถหุ่นยนต์ของโครงงาน System Direction Control for Controller Car via Bluetooth on Smartphone

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็นโครงงานเป็นโครงงานที่นำเสนอเกี่ยวกับการควบคุมรถหุ่นยนต์ด้วยโทรศัพท์มือถือระบบปฏิบัติการ Symbian ของโนเกีย (Nokia) ผ่านเทคโนโลยีไร้สายบลูทูธ (Bluetooth) โดยมีการใช้งานไมโครคอนโทรลเลอร์เบสิกแสตมป์ (Basic Stamp) รุ่น i-stamp 2P24 ซึ่งง่ายต่อการศึกษา โดยเบสิกแสตมป์เป็นไมโครคอนโทรลเลอร์ที่ผลิตโดยบริษัท Parallax Inc. มีตัวแปรภาษาเบสิก (Basic Interpreter) อยู่ภายในตัวเอง ในการพัฒนาเบสิกแสตมป์จะใช้ชุดคำสั่งภาษาเบสิกที่เรียกว่า พีเบสิก (PBASIC)

ส่วนของการเชื่อมต่อไร้สายมีการใช้งานแผงวงจรสื่อสารข้อมูลไร้สายผ่านโมดูลบลูทูธแบบสเลฟ (ZX-Bluetooth) ซึ่งมีรัศมีทำการสูงสุด 10 เมตร

รูปที่ 2.27 ZX-Bluetooth

โดยการควบคุมมีการใช้งานโทรศัพท์มือถือ Nokia ที่มีระบบปฏิบัติการ Symbian

รูปที่ 2.28 โทรศัพท์มือถือ Nokia

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.29 หน้าจอส่วนติดต่อกับผู้ใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

การวิเคราะห์และออกแบบระบบ

3.1 โครงสร้างและภาพรวมระบบ

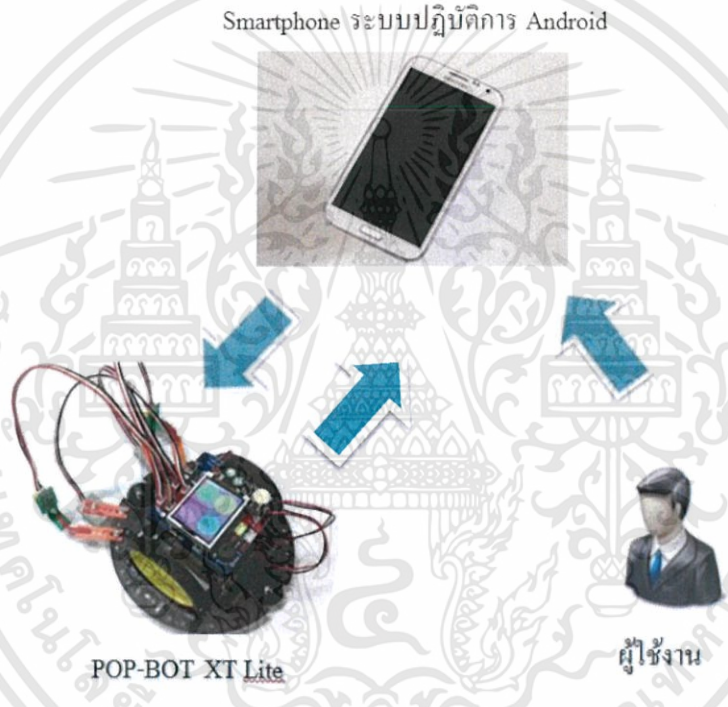
โครงงานนี้ได้แบ่งการพัฒนาระบบออกเป็น 3 ส่วนคือ ส่วนของการโปรแกรมลงบนสมาร์ทโฟนระบบปฏิบัติการแอนดรอยด์ ส่วนของการโปรแกรมลงบนรถหุ่นยนต์ และส่วนของการโปรแกรมเพื่อตั้งค่าอุปกรณ์สำหรับเชื่อต่อ Wi-Fi โดยในส่วนของการโปรแกรมบนสมาร์ทโฟนระบบปฏิบัติการแอนดรอยด์จะใช้โปรแกรม Eclipse IDE ที่มีการติดตั้ง ADT (Android Develop Tool) โดยใช้ภาษา Java และส่วนของการโปรแกรมลงบนรถหุ่นยนต์ จะใช้โปรแกรม Anduino เขียนโปรแกรมลงบนรถหุ่นยนต์เพื่อสั่งการให้เคลื่อนที่และใช้งานเซนเซอร์ต่างๆ ด้วยภาษา C/C++ และยังต้องมีการตั้งค่าการใช้งานอุปกรณ์สำหรับเชื่อมต่อ Wi-Fi ที่รถหุ่นยนต์ โดยการตั้งค่าต้องใช้งานโปรแกรม TeraTeam Pro โดยใช้คำสั่งบางส่วนที่ระบุไว้ในบทที่ 2

การใช้งานแอปพลิเคชันในการควบคุมรถหุ่นยนต์จะแบ่งออกเป็น 3 รูปแบบ คือ ใช้การกดปุ่มบนหน้าจอสมาร์ทโฟน ใช้การเอียงสมาร์ทโฟน และใช้การลากคันบังคับ (Joystick) บนหน้าจอสมาร์ทโฟน โดยภาพรวมในการใช้งานระบบเป็นดังนี้

1. ผู้ใช้ทำการเปิด Wi-Fi Hotspot ที่สมาร์ทโฟนระบบปฏิบัติการแอนดรอยด์จากนั้นทำการตั้งค่า Wi-Fi Hotspot ดังนี้
   - Network SSID : Jae
   - Security : WPA2 PSK
   - Password : 123456789
2. ทำการเปิดรถหุ่นยนต์ รอจนรถหุ่นยนต์แจ้งหมายเลข IP Address และหมายเลข port
3. เปิดแอปพลิเคชันบนสมาร์ทโฟน จากนั้นระบุ IP Address และ port ที่ปรากฏบนหน้าจอของรถหุ่นยนต์ แล้วทำการเชื่อมต่อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. ทำการควบคุมรถหุ่นยนต์ (โดยสามารถเปลี่ยนรูปแบบการควบคุมได้)
   - ใช้การกดปุ่มบนสมาร์ทโฟน
   - ใช้การเอียงสมาร์ทโฟน
   - ใช้การลากคันบังคับ
5. หุ่นยนต์ทำการส่งค่าของเซนเซอร์กลับมาที่สมาร์ทโฟน
6. ผู้ใช้ออกจากแอปพลิเคชัน



รูปที่ 3.1 ระบบการควบคุมรถหุ่นยนต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

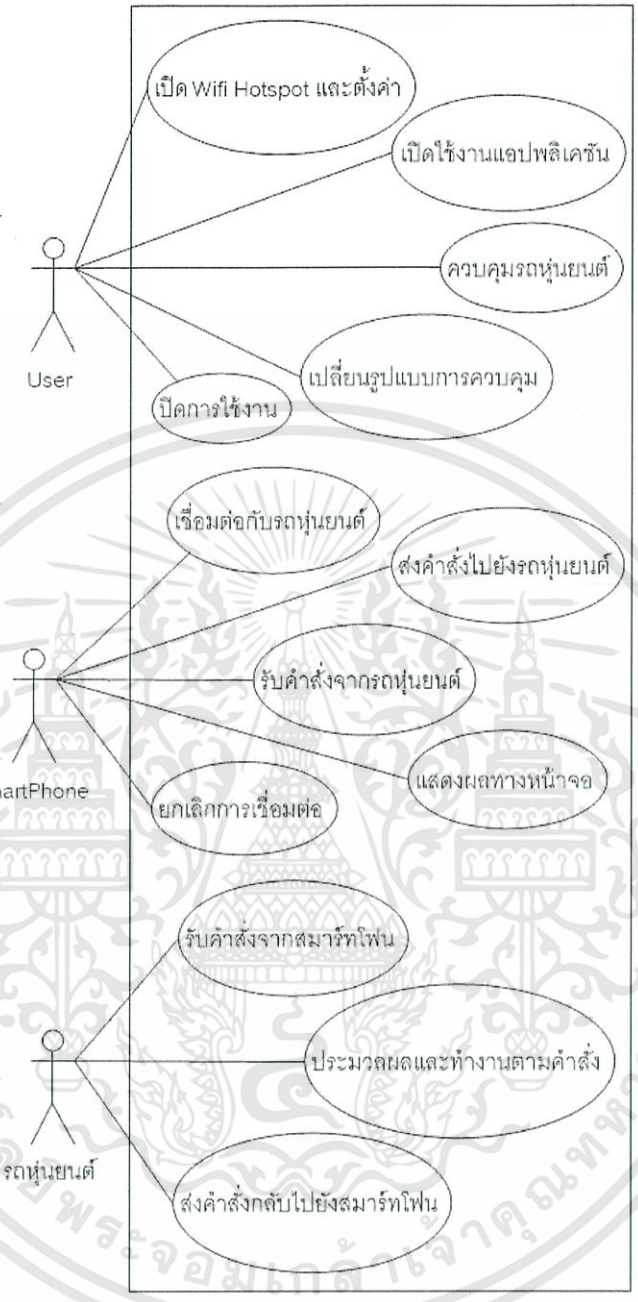
3.2 การออกแบบระบบ

ในการออกแบบระบบจะทำการสร้างแผนภาพการทำงานของผู้ใช้ระบบ (Use Case Diagram) แผนขั้นตอนการทำงานระบบ (Flow chart) แผนภาพกิจกรรมของระบบงาน (Activity Diagram) เพื่อทำให้เข้าใจระบบได้ดียิ่งขึ้น

3.2.1 แผนภาพการทำงานของผู้ใช้ระบบ

แผนภาพนี้จะแสดงกระบวนการทำงานของผู้ใช้งาน สมาร์ทโฟน และรถหุ่นยนต์ว่าแต่ละส่วนสามารถทำสิ่งใดในระบบได้ โดยขั้นตอนการทำงานขั้นแรกผู้ใช้จะต้องทำการเปิด Wi-Fi Hotspot สมาร์ทโฟน จากนั้นทำการเชื่อมต่อสมาร์ทโฟนกับรถหุ่นยนต์โดยเปิดแอปพลิเคชันที่ใช้ควบคุมแล้วกำหนดค่า IP Address และ port ของ RN-XV WiFly Module ได้รับจากสมาร์ทโฟนลงบนแอปพลิเคชันให้ตรงกันจากนั้นทำการเชื่อมต่อ ถ้าการเชื่อมต่อระหว่างสมาร์ทโฟนและรถหุ่นยนต์สำเร็จแอปพลิเคชันจะเข้าสู่หน้าจอการควบคุมด้วยการกดปุ่ม จากนั้นผู้ใช้งานจะสามารถควบคุมการทำงานของรถหุ่นยนต์ได้ โดยใช้สัญญาณ Wi-Fi เป็นสื่อกลางในการส่งคำสั่ง ซึ่งรถหุ่นยนต์จะเคลื่อนที่ตามที่ผู้ใช้งานสั่งผ่านสมาร์ทโฟน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.2 Use case Diagram ระบบควบคุมรถหุ่นยนต์ด้วยสมาร์ทโฟนผ่านเครือข่ายไร้สาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.2 แผนภาพขั้นตอนการทำงานของระบบ

หลักการทำงานโดยรวมของระบบสามารถแบ่งออกได้เป็นการทำงานสองส่วนย่อย คือ ส่วนของการทำงานบนสมาร์ทโฟนที่ทำหน้าที่เป็นตัวควบคุมการใช้งานรถหุ่นยนต์ และส่วนการทำงานของรถหุ่นยนต์ที่ดำเนินการตามคำสั่ง มีการใช้งานเซนเซอร์ต่างๆ และส่งค่ากลับ ซึ่งภาพรวมของการทำงานทั้งระบบสามารถแสดงเป็นแผนภาพได้ดังต่อไปนี้

รูปที่ 3.3 Flow Chart ของระบบควบคุมรถหุ่นยนต์ด้วยสมาร์ทโฟนผ่านเครือข่ายไร้สาย

3.2.2.1 แผนภาพขั้นตอนการทำงานของสมาร์ทโฟน

การทำงานของระบบบนส่วนของสมาร์ทโฟนระบบปฏิบัติการแอนดรอยด์หลังจากเปิดใช้งาน Wi-Fi Hotspot และตั้งค่าตามที่ระบุไว้ข้างต้น เมื่อผู้ใช้เปิดแอปพลิเคชันขึ้นมาผู้ใช้จะต้องทำการระบุ IP Address และ port ที่ปรากฏบนรถหุ่นยนต์แล้วกดปุ่มเชื่อมต่อ จากนั้นสมาร์ทโฟนจะ

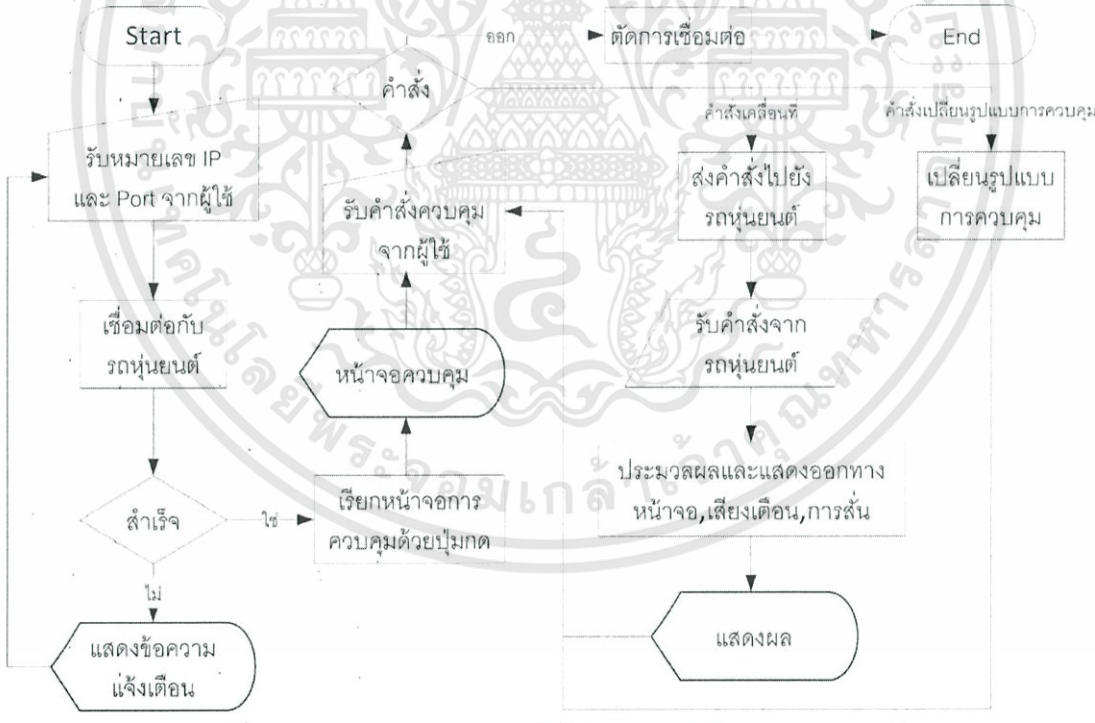
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทำการเชื่อมต่อกับไมโครคอนโทรลเลอร์ของรถหุ่นยนต์ โดยผลลัพธ์ของการเชื่อมต่อสามารถแบ่งออกเป็นสองกรณี

- กรณีที่การเชื่อมต่อสำเร็จจะเข้าสู่หน้าจอสำหรับควบคุมรถหุ่นยนต์ด้วยปุ่มกดเพื่อให้ผู้ใช้ทำการควบคุมรถหุ่นยนต์ตามที่ต้องการ
- กรณีที่การเชื่อมต่อไม่สำเร็จสามารถแบ่งเป็นกรณีย่อยได้ดังนี้

1. ผู้ใช้งานระบุ IP Address ไม่ถูกต้องตามหลัก IPv4 จะทำการแสดงข้อความ Illegal IP Address
2. ผู้ใช้งานไม่ได้ระบุ port จะทำการแสดงข้อความ Please enter PORT
3. เชื่อมต่อไม่สำเร็จเนื่องจากปัญหาด้านเครือข่าย จะทำการแสดงข้อความ Connect Fail

โดยการทำงานของสมาร์ทโฟนหลังจากทำการปล่อย Wi-Fi Hotspot และเข้าสู่แอปพลิเคชันสามารถแสดงเป็นแผนภาพได้ดังต่อไปนี้

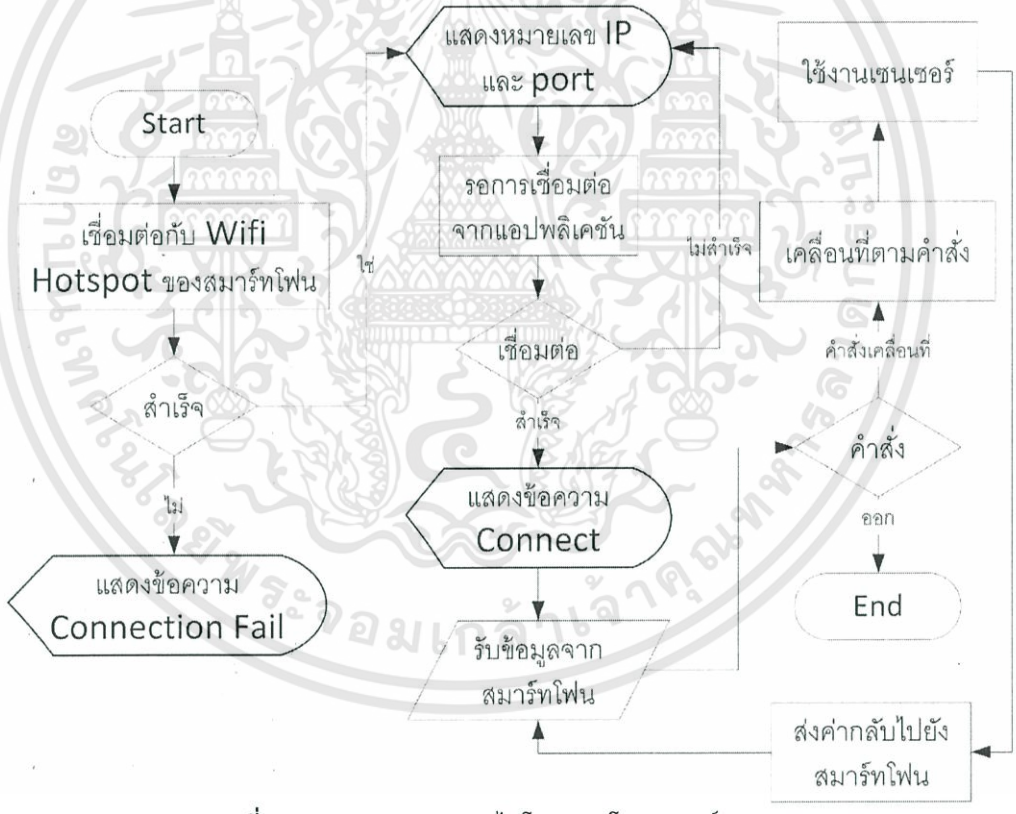


รูปที่ 3.4 Flow Chart ของสมาร์ทโฟนระบบปฏิบัติการแอนดรอยด์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.2.2 แผนภาพขั้นตอนการทำงานของรถหุ่นยนต์

การทำงานของระบบในส่วนของรถหุ่นยนต์ แผนภาพจะแสดงให้เห็นว่าหลังจากที่รถหุ่นยนต์ทำการเชื่อมต่อกับ Wi-Fi Hotspot ของสมาร์ทโฟนเพื่อขอรับ IP Address แล้ว รถหุ่นยนต์จะแสดง IP Address ที่ได้รับผ่านทางหน้าจอเพื่อรอการเชื่อมต่อจากสมาร์ทโฟนโดยการระบุ IP Address และ port ตามที่ปรากฏบนรถหุ่นยนต์ หากการเชื่อมต่อสำเร็จหน้าจอของรถหุ่นยนต์จะแสดงคำว่า Connect จากนั้นจะรอรับคำสั่งที่ผู้ใช้ทำการสั่งจากสมาร์ทโฟน เมื่อได้รับคำสั่งแล้วจะทำการประมวลผลคำสั่งเพื่อดำเนินการในขณะเดียวกันก็จะมีการส่งข้อมูลของเซนเซอร์ที่เชื่อมต่อกับรถหุ่นยนต์กลับไปยังสมาร์ทโฟน โดยจะทำซ้ำอย่างนี้เรื่อยๆ จนกว่าผู้ใช้จะออกจากโปรแกรม โดยการทำงานของรถหุ่นยนต์หลังจากทำการเปิดใช้งานสามารถแสดงเป็นแผนภาพได้ดังต่อไปนี้

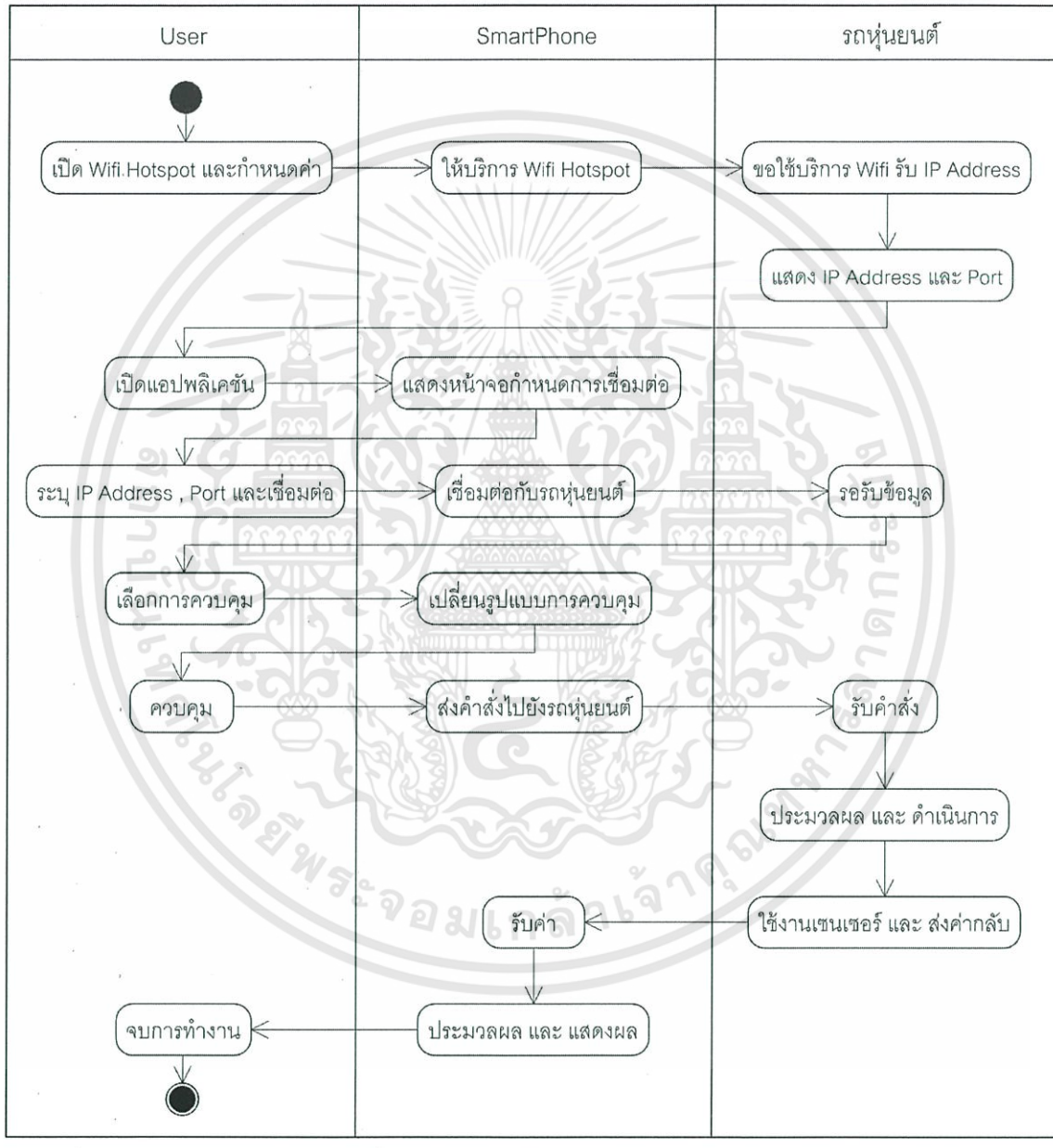


รูปที่ 3.5 Flow Chart ของไมโครคอนโทรลเลอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.3 แผนภาพกิจกรรมของระบบงาน

แผนภาพกิจกรรมของระบบจะแสดงขั้นตอนการทำงานของระบบทั้งหมดว่าเริ่มต้นจากที่ใด สิ้นสุดที่ใด แต่ละกิจกรรมใครเป็นผู้กระทำ มีความเกี่ยวข้องและทำงานร่วมกันอย่างไรทั้งผู้ใช้งาน สมาร์ทโฟน และรถหุ่นยนต์ โดยมีขั้นตอนกิจกรรมของระบบงาน ดังรูปที่ 3.6



รูปที่ 3.6 แผนภาพกิจกรรมของระบบ

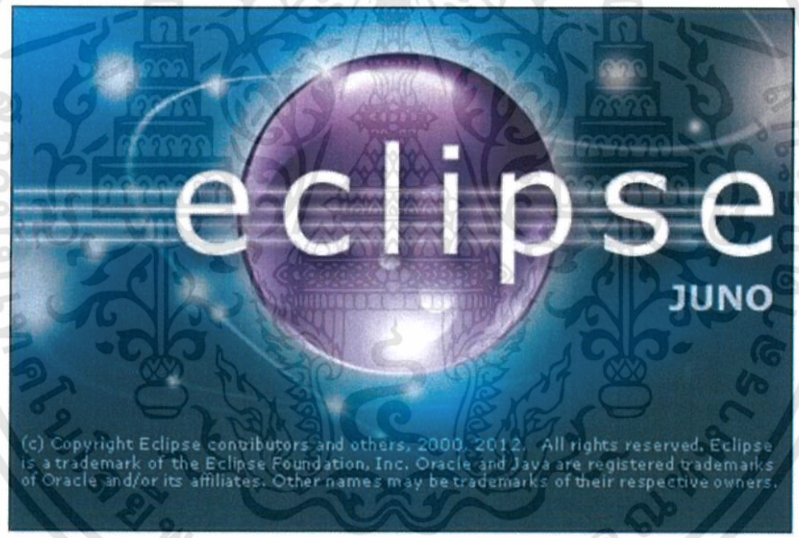
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 สิ่งที่จำเป็นในการพัฒนาระบบ

ในการพัฒนาโครงงานนี้ต้องมีการจัดเตรียมเครื่องมือสำหรับใช้ในการพัฒนาแอปพลิเคชันบนสมาร์ทโฟนระบบปฏิบัติการแอนดรอยด์ที่พร้อมและสามารถใช้งานได้อย่างมีประสิทธิภาพบนคอมพิวเตอร์ระบบปฏิบัติการวินโดว์ (Windows) ซึ่งผู้ที่ต้องการจะพัฒนาแอปพลิเคชันบนระบบปฏิบัติการแอนดรอยด์สามารถดาวน์โหลดและติดตั้งเครื่องมือได้ตามรายละเอียดดังนี้

3.3.1 Eclipse Development Tools and Java Development Kit (JDK)

สำหรับโปรแกรม Eclipse ปกติแล้วจะนิยมสำหรับเขียนโปรแกรมจาวาแต่ทั้งนี้เราสามารถติดตั้ง Plugin สำหรับเขียนโปรแกรมบนแอนดรอยด์ได้ โดยการติดตั้ง Eclipse จะต้องติดตั้ง Java Development Kit ด้วยเพื่อให้สามารถเขียนโปรแกรมบนแอนดรอยด์ได้



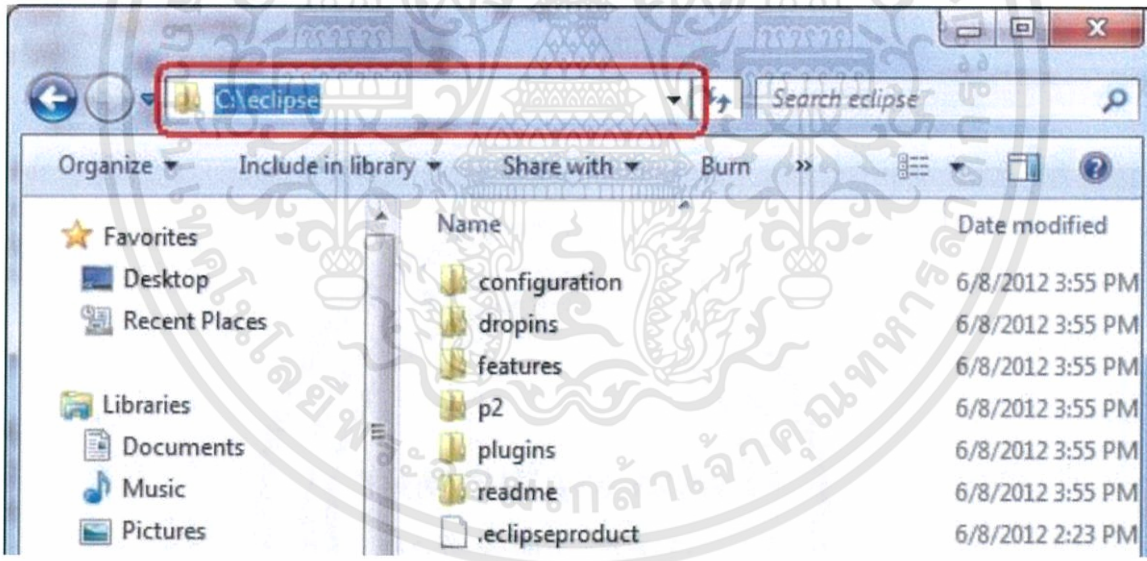
รูปที่ 3.7 โลโก้โปรแกรม Eclipse

โปรแกรม Eclipse ถือเป็นโปรแกรมขนาดเล็กโดยมีขนาดประมาณ 200-300 MB โดยสามารถดาวน์โหลดไฟล์ที่เป็น .zip มาใช้ได้เลยโดยไม่ต้องทำการติดตั้ง สามารถดาวน์โหลด Eclipse ได้ที่ http://www.eclipse.org/downloads โดยให้เลือกดาวน์โหลด Eclipse Classic

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.8 การดาวน์โหลดโปรแกรม Eclipse Classic

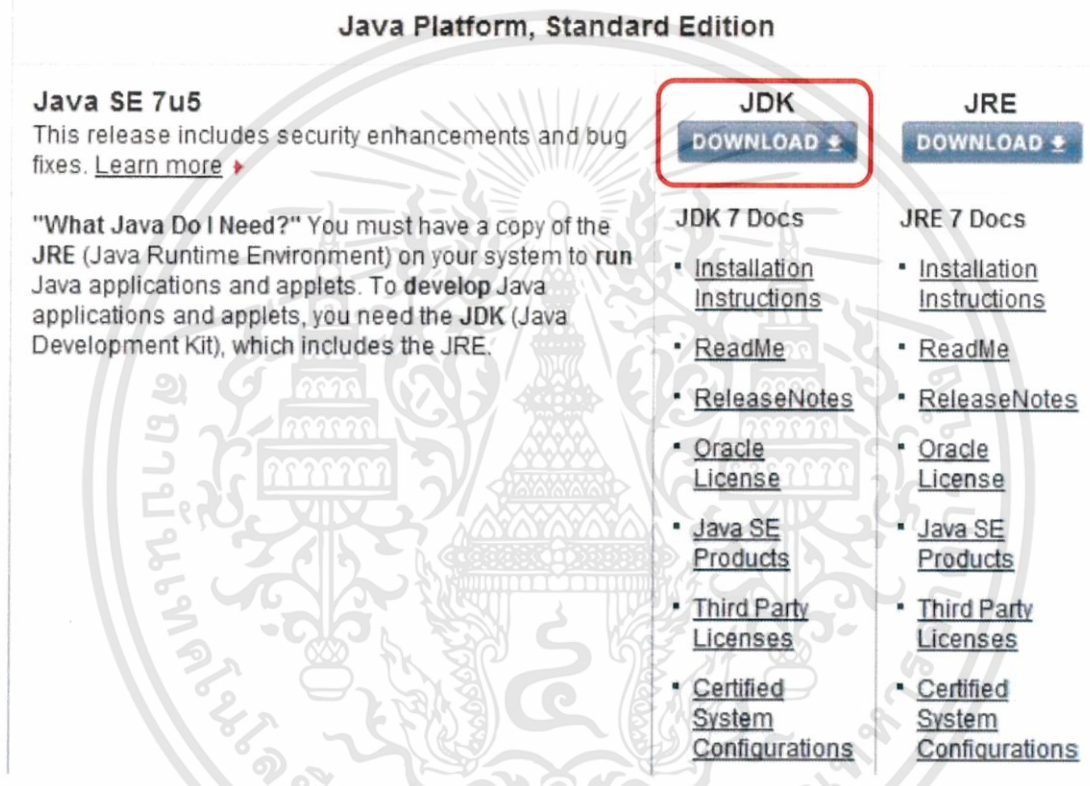
หลังจากดาวน์โหลดไฟล์ .zip มาแล้วไม่ต้องทำการติดตั้ง สามารถทำการแตกไฟล์และสามารถใช้งานได้ทันทีโดยในตัวอย่างนี้ทำการแตกไฟล์ไว้ที่ C:\eclipse



รูปที่ 3.9 ทำการแตกไฟล์ .zip ไว้ที่ C:\eclipse

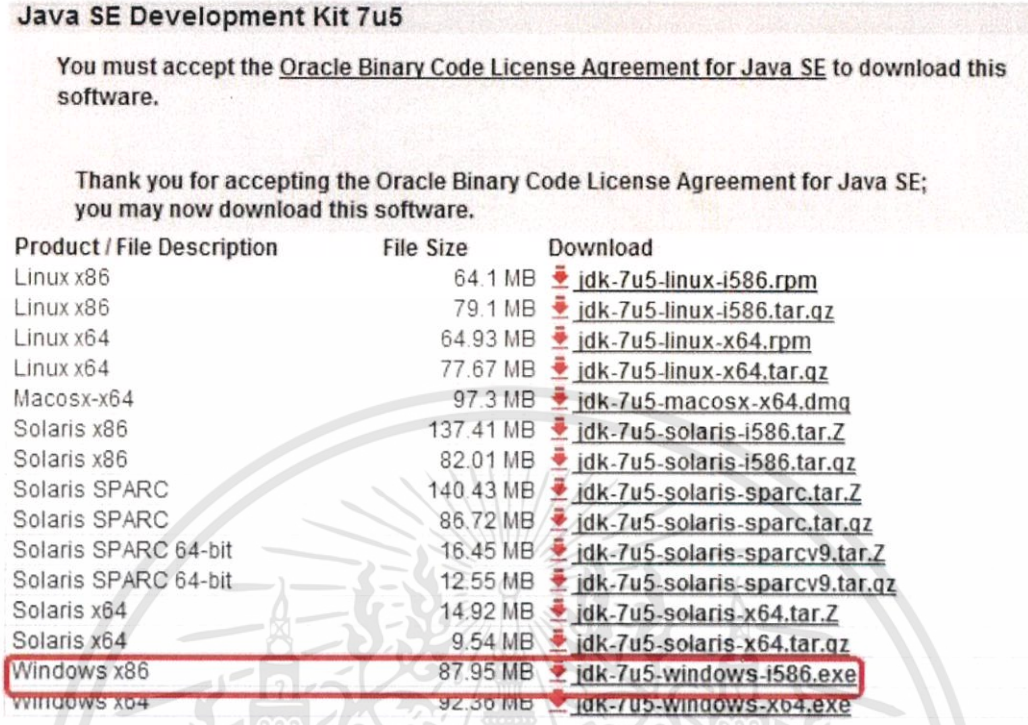
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อทำการแตกไฟล์โปรแกรมเสร็จแล้ว ถ้าคอมพิวเตอร์ยังไม่ได้ทำการติดตั้ง Java Development Kit โปรแกรม Eclipse จะเกิดข้อผิดพลาด (Error) เมื่อเปิดโปรแกรมขึ้นมาใช้งาน ดังนั้นจึงต้องทำการติดตั้ง Java Development Kit ด้วย สามารถดาวน์โหลด Java Development Kit (JDK) ได้ที่ http://www.oracle.com/technetwork/java/javase/downloads/index.html ทำการดาวน์โหลดไฟล์ที่เป็น JDK



รูปที่ 3.10 เลือกดาวน์โหลดไฟล์ JDK

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Java SE Development Kit 7u5

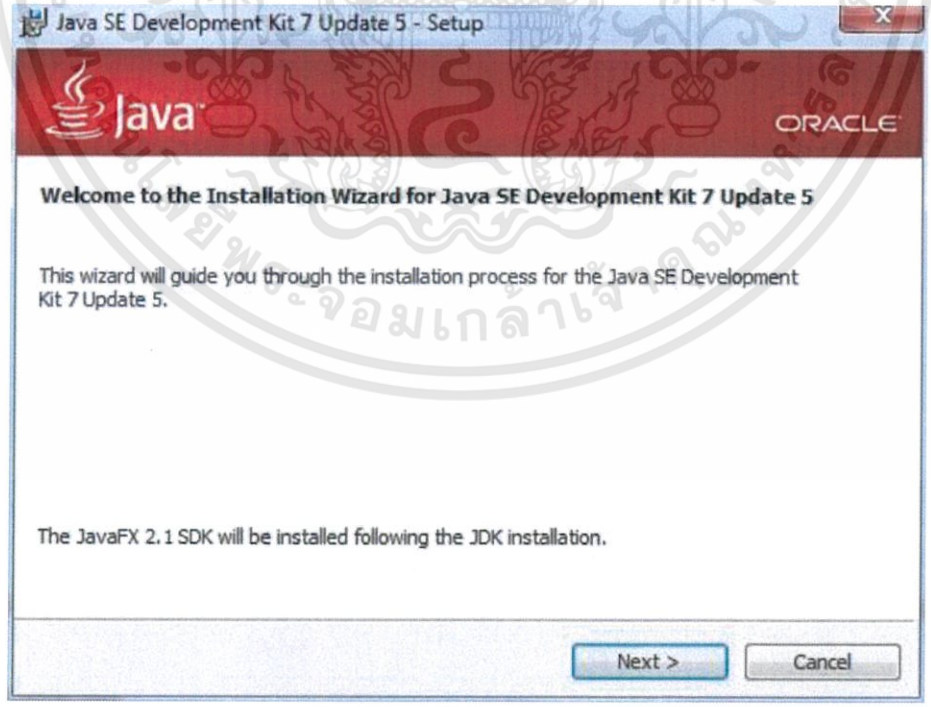
You must accept the Oracle Binary Code License Agreement for Java SE to download this software.

Thank you for accepting the Oracle Binary Code License Agreement for Java SE; you may now download this software.

Product / File Description	File Size	Download
Linux x86	64.1 MB	jdk-7u5-linux-i586.rpm
Linux x86	79.1 MB	jdk-7u5-linux-i586.tar.gz
Linux x64	64.93 MB	jdk-7u5-linux-x64.rpm
Linux x64	77.67 MB	jdk-7u5-linux-x64.tar.gz
Macosx-x64	97.3 MB	jdk-7u5-macosx-x64.dmg
Solaris x86	137.41 MB	jdk-7u5-solaris-i586.tar.Z
Solaris x86	82.01 MB	jdk-7u5-solaris-i586.tar.gz
Solaris SPARC	140.43 MB	jdk-7u5-solaris-sparc.tar.Z
Solaris SPARC	86.72 MB	jdk-7u5-solaris-sparc.tar.gz
Solaris SPARC 64-bit	16.45 MB	jdk-7u5-solaris-sparcv9.tar.Z
Solaris SPARC 64-bit	12.55 MB	jdk-7u5-solaris-sparcv9.tar.gz
Solaris x64	14.92 MB	jdk-7u5-solaris-x64.tar.Z
Solaris x64	9.54 MB	jdk-7u5-solaris-x64.tar.gz
Windows x86	87.95 MB	jdk-7u5-windows-i586.exe
Windows x64	92.36 MB	jdk-7u5-windows-x64.exe

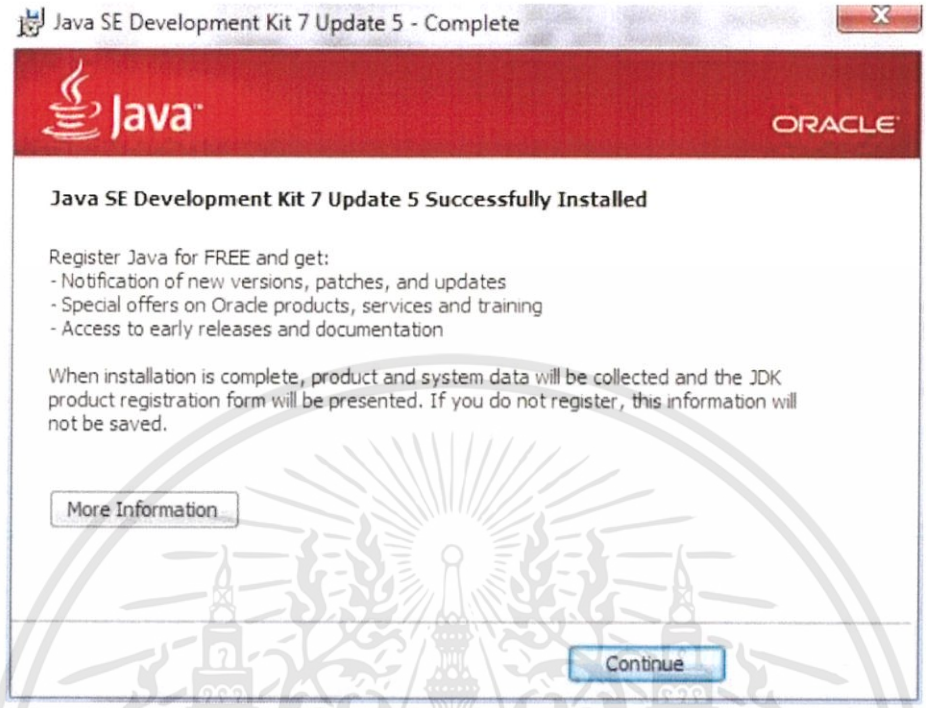
รูปที่ 3.11 เลือกดาวน์โหลดให้ถูกต้องกับระบบปฏิบัติการของคอมพิวเตอร์ที่ต้องการติดตั้งโปรแกรม

เมื่อดาวน์โหลดเสร็จสิ้นให้ทำการติดตั้งตามปกติจนสำเร็จ



รูปที่ 3.12 ทำการติดตั้ง JDK

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.13 หน้าจอเสร็จสิ้นการติดตั้ง JDK

3.3.2 Android Development Tools (ADT) Plugin for Eclipse

Android Development Tools Plugin for eclipse เป็นปลั๊กอิน (Plugin) สำหรับติดตั้งบนโปรแกรม Eclipse เพื่อรองรับการเขียนแอปพลิเคชันบนระบบปฏิบัติการแอนดรอยด์ โดยปลั๊กอิน Android Development Tools จะมีหน้าที่ในการปรับสภาพแวดล้อมของโปรแกรม Eclipse ปรับแต่งส่วนประกอบต่างๆ เพื่อเพิ่มประสิทธิภาพในการเขียนแอปพลิเคชันบนระบบปฏิบัติการแอนดรอยด์

สำหรับการติดตั้ง Android Development Tools จะต้องติดตั้งผ่านโปรแกรม Eclipse โดยขั้นแรกให้ไปที่ http://developer.android.com/tools/sdk/eclipse-adt.html จากนั้นคลิกที่ Installing the Eclipse Plugin ดังภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ADT Plugin

SEE ALSO

Installing the Eclipse Plugin

Android Development Tools (ADT) is a plugin for the Eclipse IDE that is designed to give you a powerful, integrated environment in which to build Android applications.

ADT extends the capabilities of Eclipse to let you quickly set up new Android projects, create an application UI, add packages based on the Android Framework API, debug your applications using the Android SDK tools, and even export signed (or unsigned) .apk files in order to distribute your application.

Developing in Eclipse with ADT is highly recommended and is the fastest way to get started. With the guided project setup it provides, as well as tools integration, custom XML editors, and debug output pane, ADT gives you an incredible boost in developing Android applications.

This document provides step-by-step instructions on how to download the ADT plugin and install it into your Eclipse development environment. Note that before you can install or use ADT, you must have compatible versions of both the Eclipse IDE and the Android SDK installed. For details, make sure to read Installing the Eclipse Plugin.

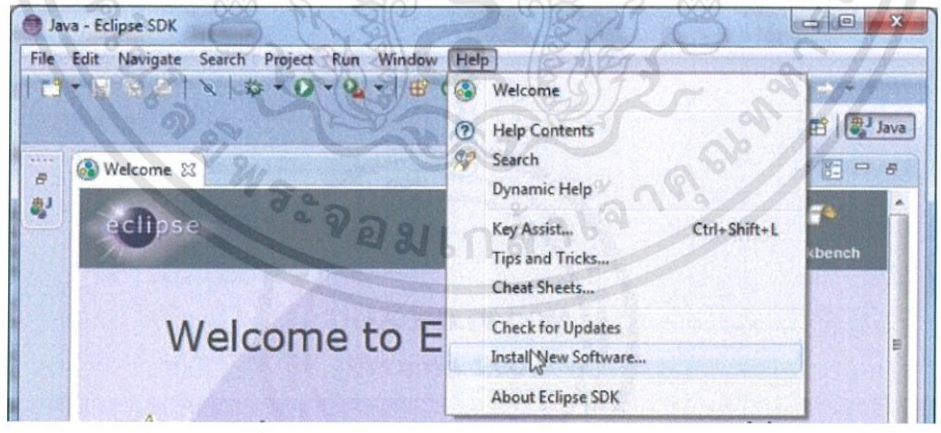
If you are already using ADT, this document also provides instructions on how to update ADT to the latest version or how to uninstall it, if necessary.

For information about the features provided by the ADT plugin, such as code editor features, SDK tool integration, and the graphical layout editor (for drag-and-drop layout editing), see the Android Developer Tools document.

รูปที่ 3.14 คลิกที่ Installing the Eclipse Plugin

จากนั้นเปิดโปรแกรม Eclipse แล้วทำตามขั้นตอนต่อไปนี้

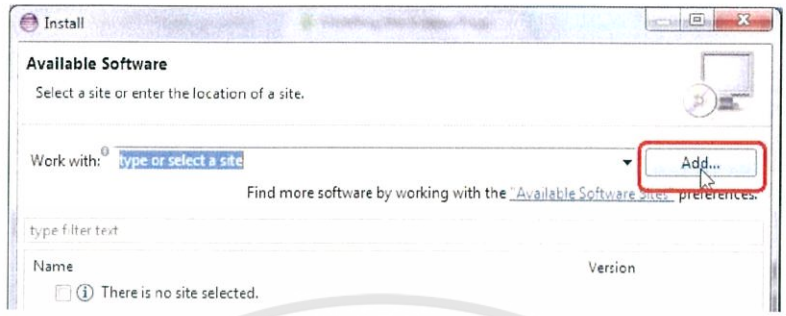
- เลือก Help > Install New Software



รูปที่ 3.15 เลือกเมนู Help > Install New Software

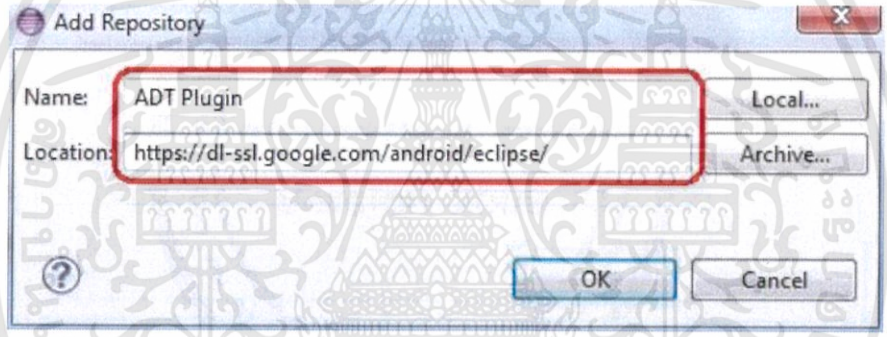
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

➢ คลิกปุ่ม Add



รูปที่ 3.16 คลิกที่ปุ่ม Add

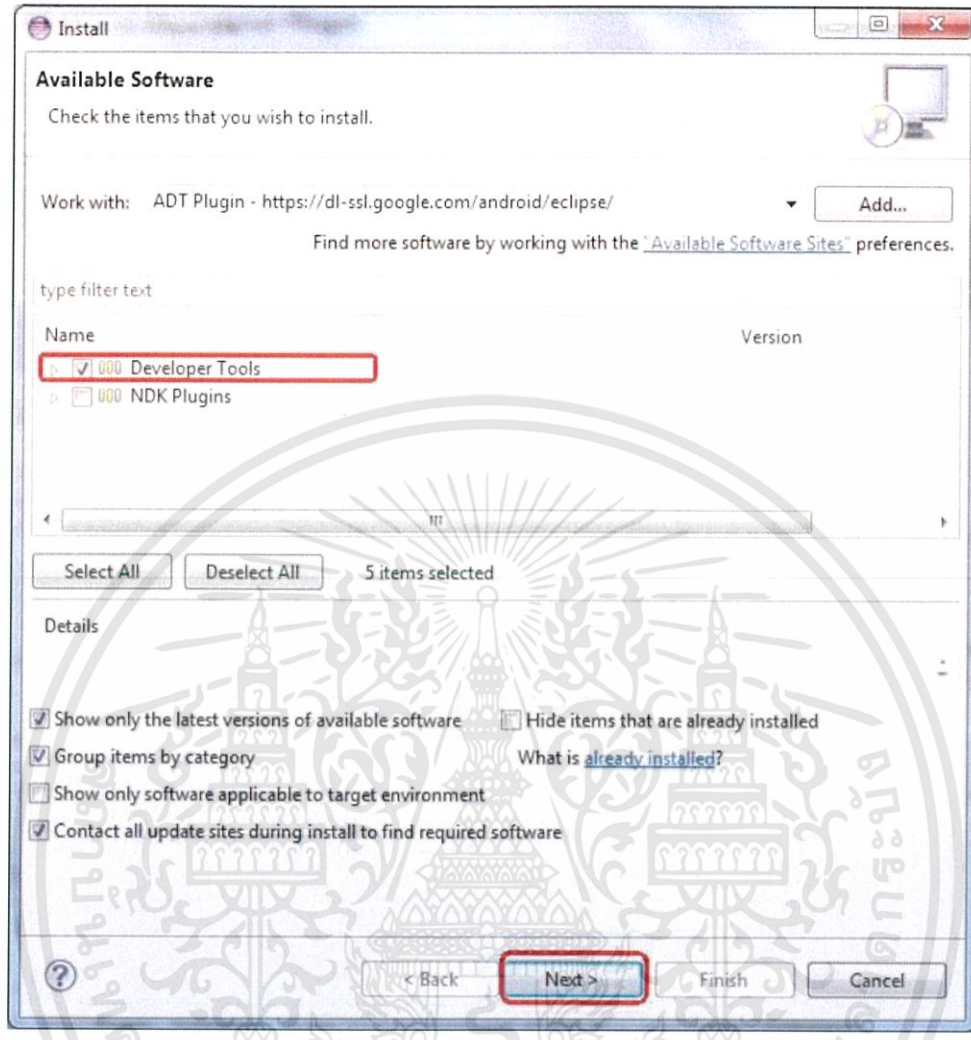
➢ ระบุ Name และ Location ให้กับปลั๊กอินดังภาพ จากนั้นกดปุ่ม OK



รูปที่ 3.17 ระบุ Name และ Location ให้กับปลั๊กอิน

➢ เลือก Developer Tools จากนั้นกดปุ่ม Next

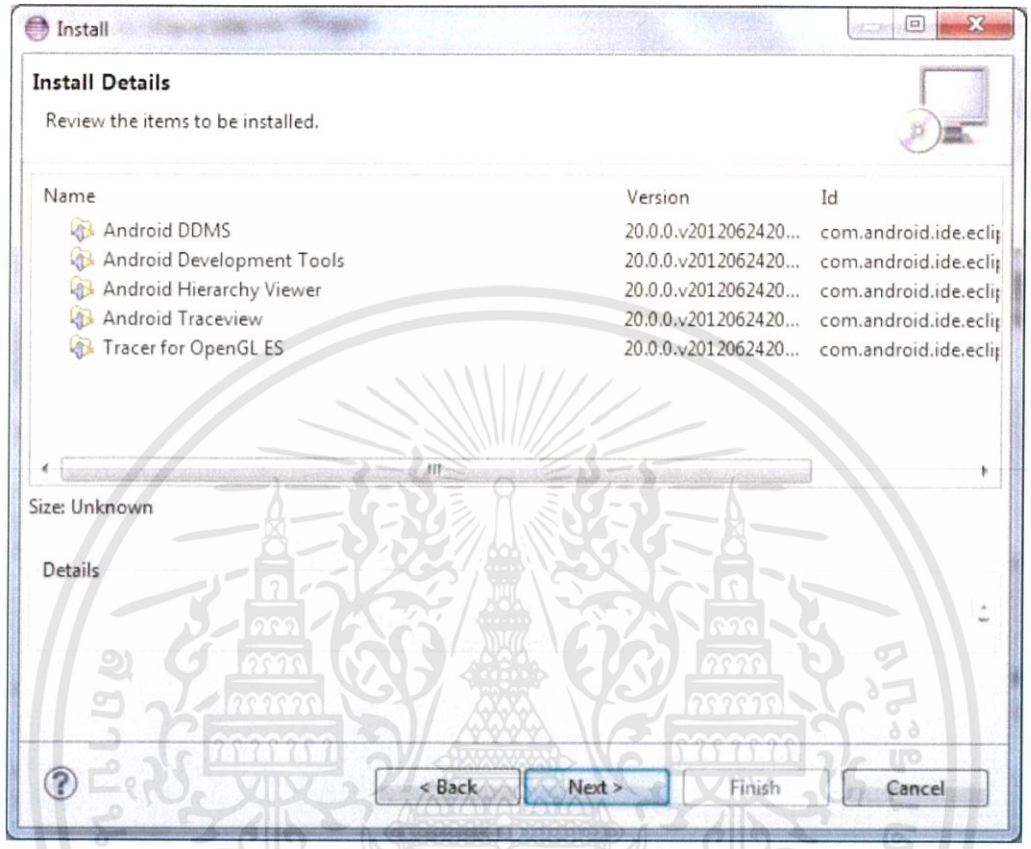
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.18 คลิกเลือก Developer Tools และ Next

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

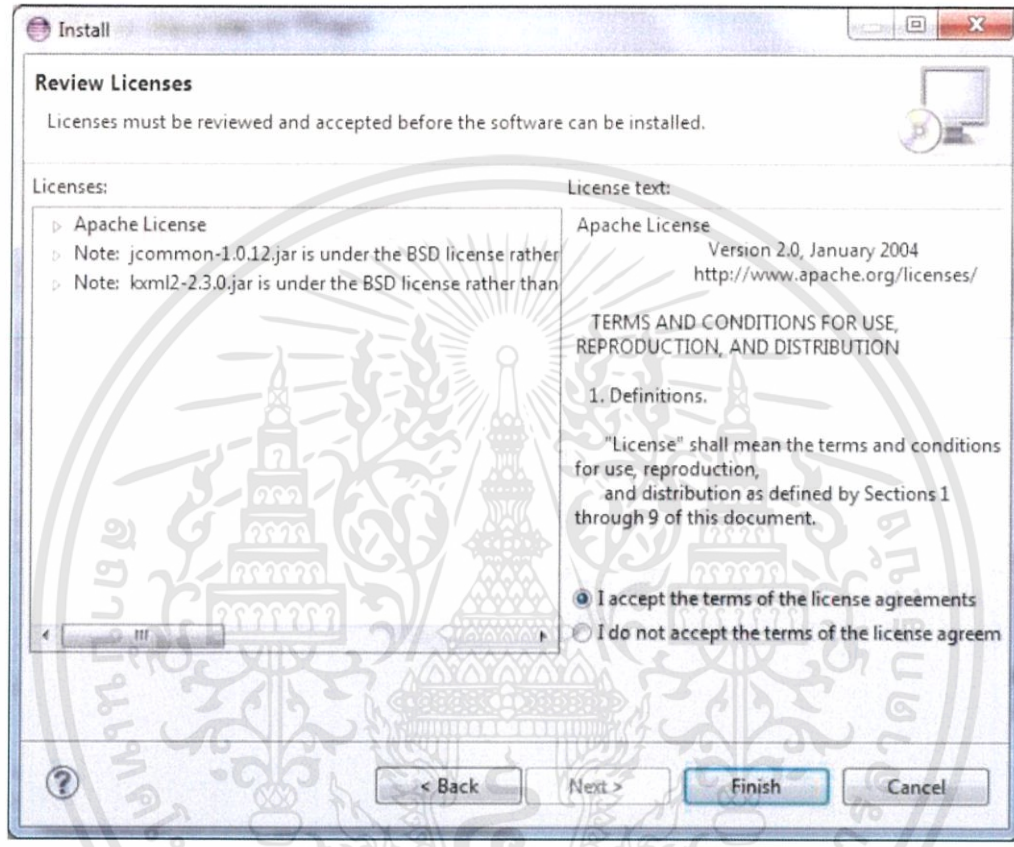
➢ กดปุ่ม Next เพื่อติดตั้งปลั๊กอินให้กับโปรแกรม Eclipse



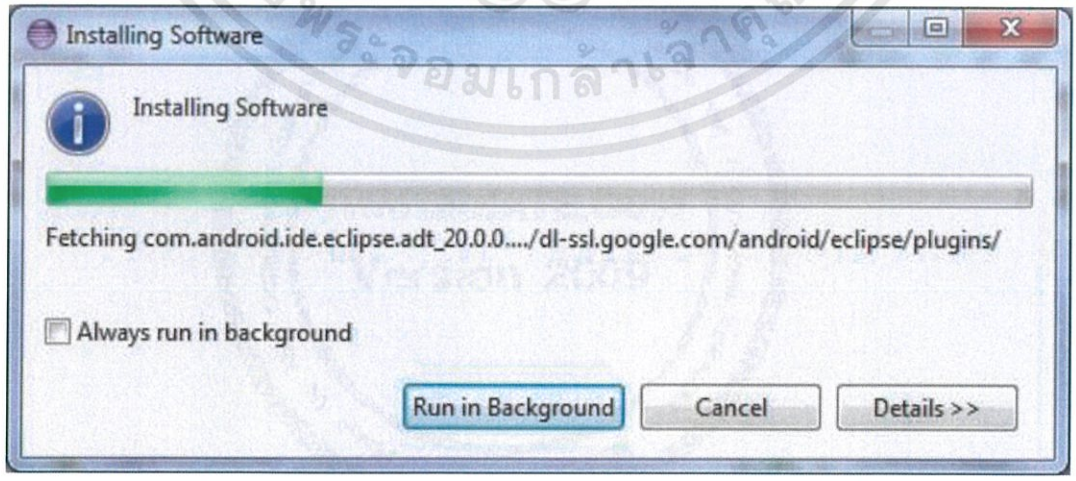
รูปที่ 3.19 แสดงรายละเอียดส่วนประกอบของปลั๊กอินที่จะทำการติดตั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- คลิกเลือก I accept the terms of license agreements เพื่อยอมรับเงื่อนไขการใช้งาน และกดปุ่ม Finish ตามลำดับ จากนั้นโปรแกรม Eclipse จะทำการดาวน์โหลดและติดตั้ง Android Development Tools



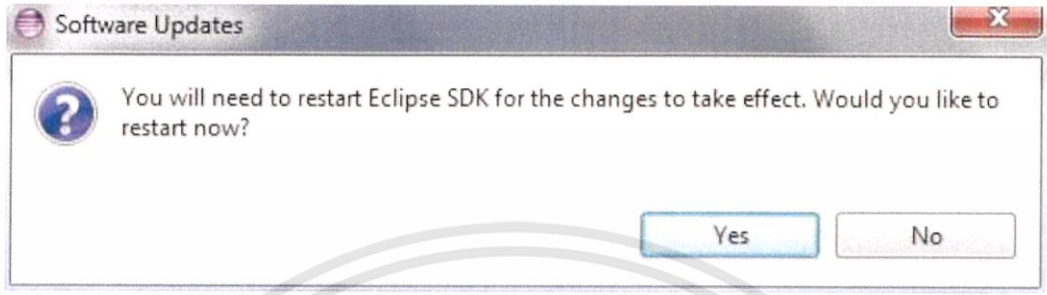
รูปที่ 3.20 หน้าจอแสดงเงื่อนไขข้อตกลงการใช้งาน



รูปที่ 3.21 โปรแกรม Eclipse จะทำการดาวน์โหลดและติดตั้งปลั๊กอิน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- เมื่อการติดตั้งปลั๊กอินเสร็จสิ้นให้ทำการเริ่มโปรแกรม Eclipse ใหม่อีกครั้ง จากนั้นสามารถใช้งานได้ตามปกติ



รูปที่ 3.22 ทำการเริ่มโปรแกรม Eclipse ใหม่

3.3.3 Android Software Development Kit (Android SDK)

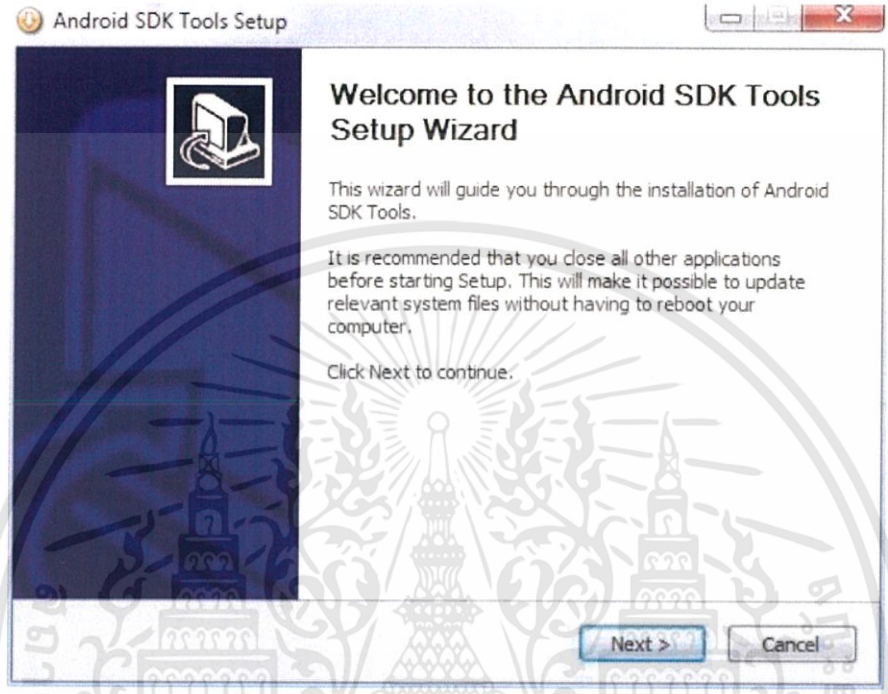
สำหรับ Android Software Development Kit นั้นเป็นไลบรารี่ (Library) ที่จำเป็นในการสร้างและเขียนแอปพลิเคชันสำหรับระบบปฏิบัติการแอนดรอยด์ ดังนั้นจึงเป็นต้องดาวน์โหลดมาติดตั้งด้วย

การติดตั้ง Android Software Development Kit สามารถติดตั้งจากโปรแกรม Eclipse ได้โดยตรงหรือจะติดตั้งด้วยวิธีที่นำเสนอนี้ก็ได้ โดยสามารถดาวน์โหลดได้จาก http://developer.android.com/sdk/index.html

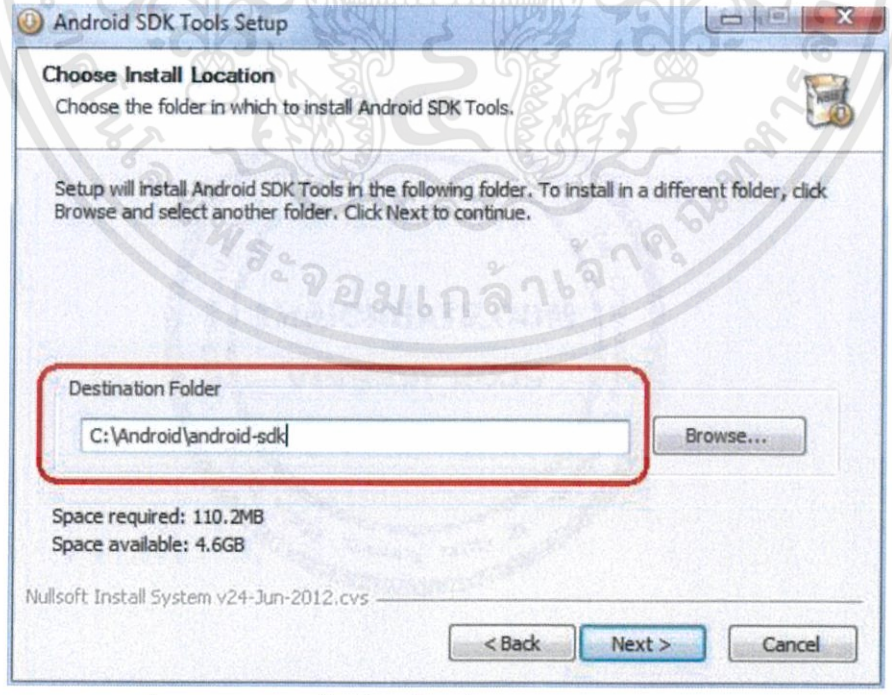
รูปที่ 3.23 Download the SDK for Windows

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อทำการดาวน์โหลดเสร็จสิ้นให้ทำการติดตั้งโปรแกรมตามปกติ โดยในการกำหนดพื้นที่สำหรับติดตั้งให้เลือกไปที่ C:\Android\android-sdk\ (จะมีการนำ Path ที่ติดตั้งไปใช้งานในภายหลัง)

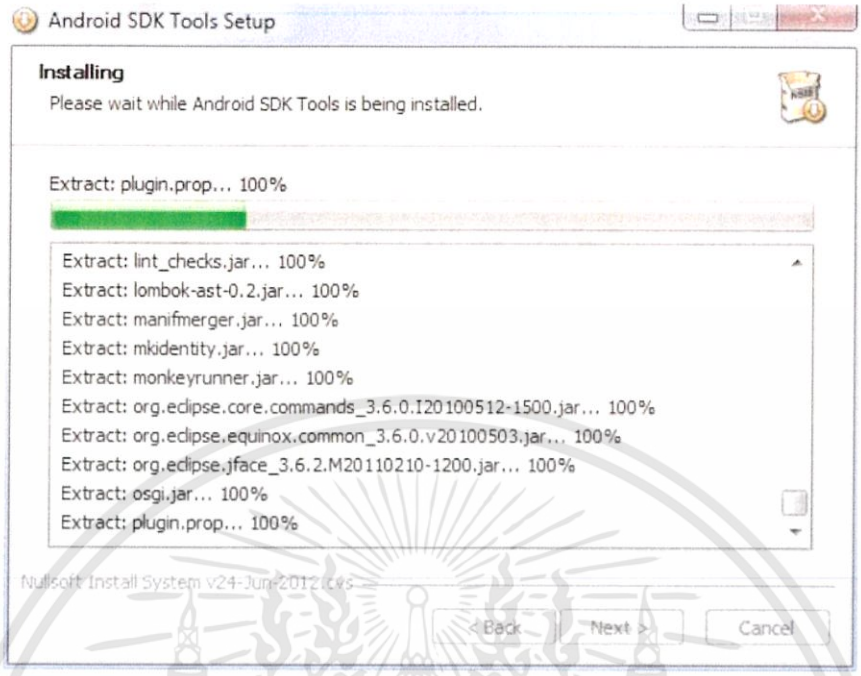


รูปที่ 3.24 ทำการติดตั้ง Android Software Development Kit ที่ดาวน์โหลดมา



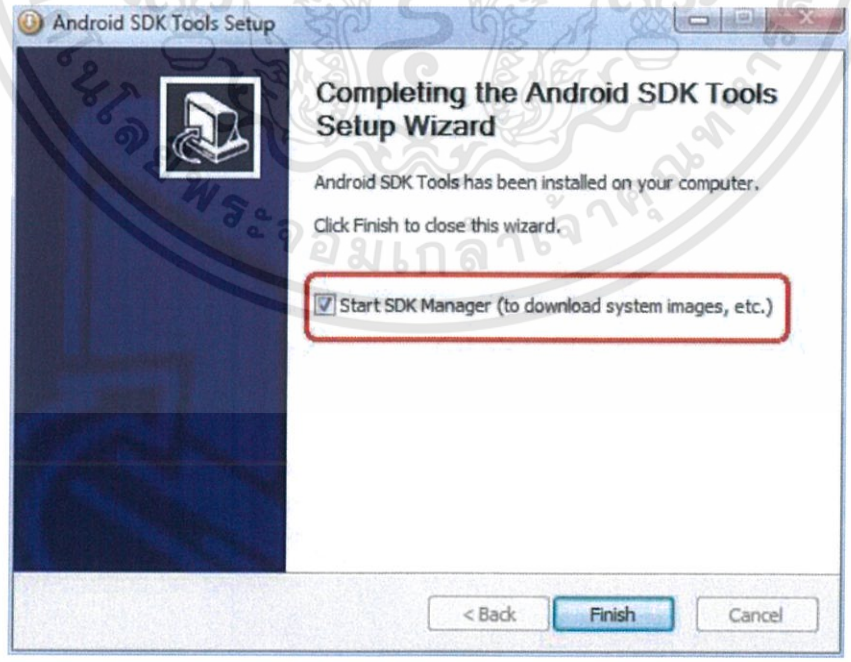
รูปที่ 3.25 กำหนดพื้นที่ติดตั้งไว้ที่ C:\Android\android-sdk\

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.26 แสดงความคืบหน้าในการติดตั้ง

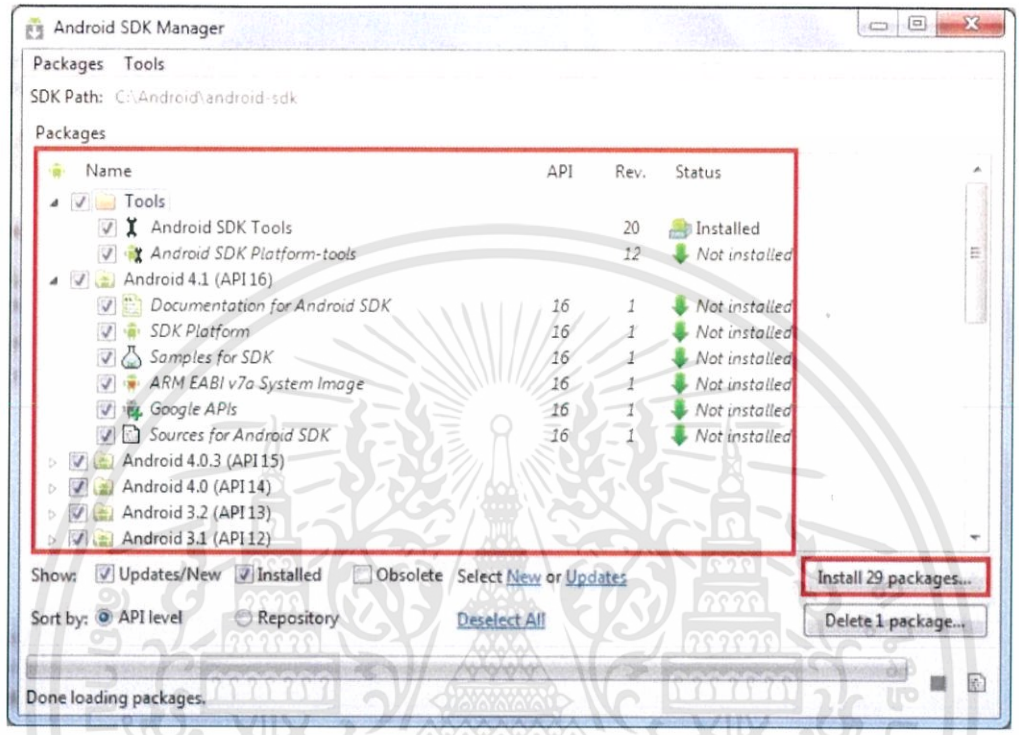
เมื่อการติดตั้งเสร็จสิ้นให้เลือก Start SDK Manager ก่อนกดปุ่ม Finish เพื่อเปิดโปรแกรมสำหรับปรับแต่ง Android SDK Manager (หรือสามารถเปิดได้จาก C:\Android\android-sdk\SDK Manager.exe)



รูปที่ 3.27 เลือก Start SDK Manager แล้วกดปุ่ม Finish

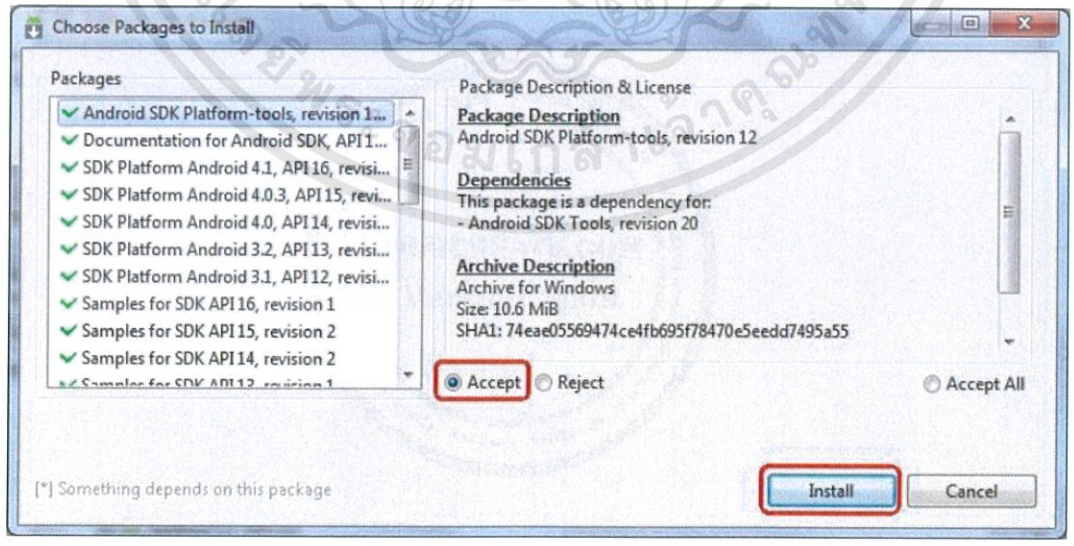
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เลือกรายการใน Tools ทุกหัวข้อดังในภาพ ส่วนการเลือก Android นั้นขึ้นอยู่กับว่าต้องการจะพัฒนาแอปพลิเคชันบนระบบปฏิบัติการแอนดรอยด์รุ่น (Version) ใดบ้าง จากนั้นกดปุ่ม Install



รูปที่ 3.28 ทำการดาวน์โหลด Tools และ Android Version

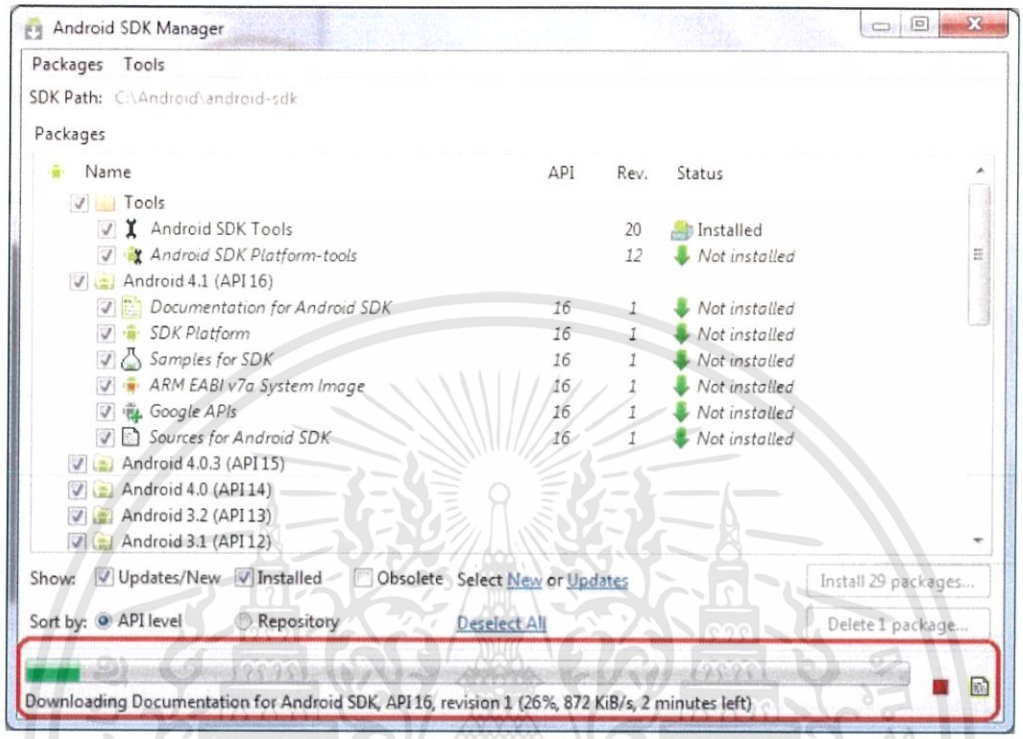
เลือก Accept เพื่อยอมรับเงื่อนไขข้อตกลงการใช้งาน จากนั้นกด Install



รูปที่ 3.29 เลือก Accept และกดปุ่ม Install เพื่อติดตั้ง

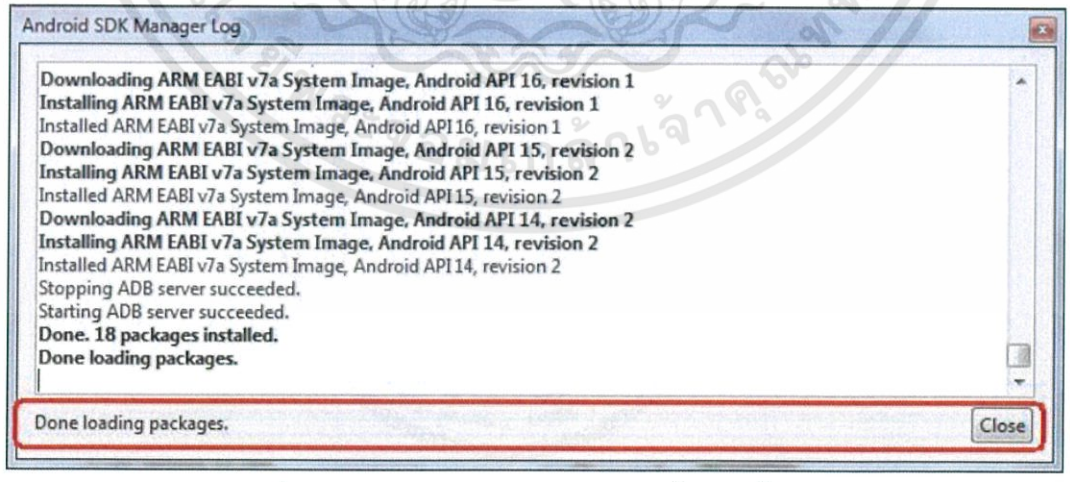
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากนั้นโปรแกรมจะทำการดาวน์โหลดสิ่งที่เราเลือกไว้



รูปที่ 3.30 แสดงความคืบหน้าการดาวน์โหลด

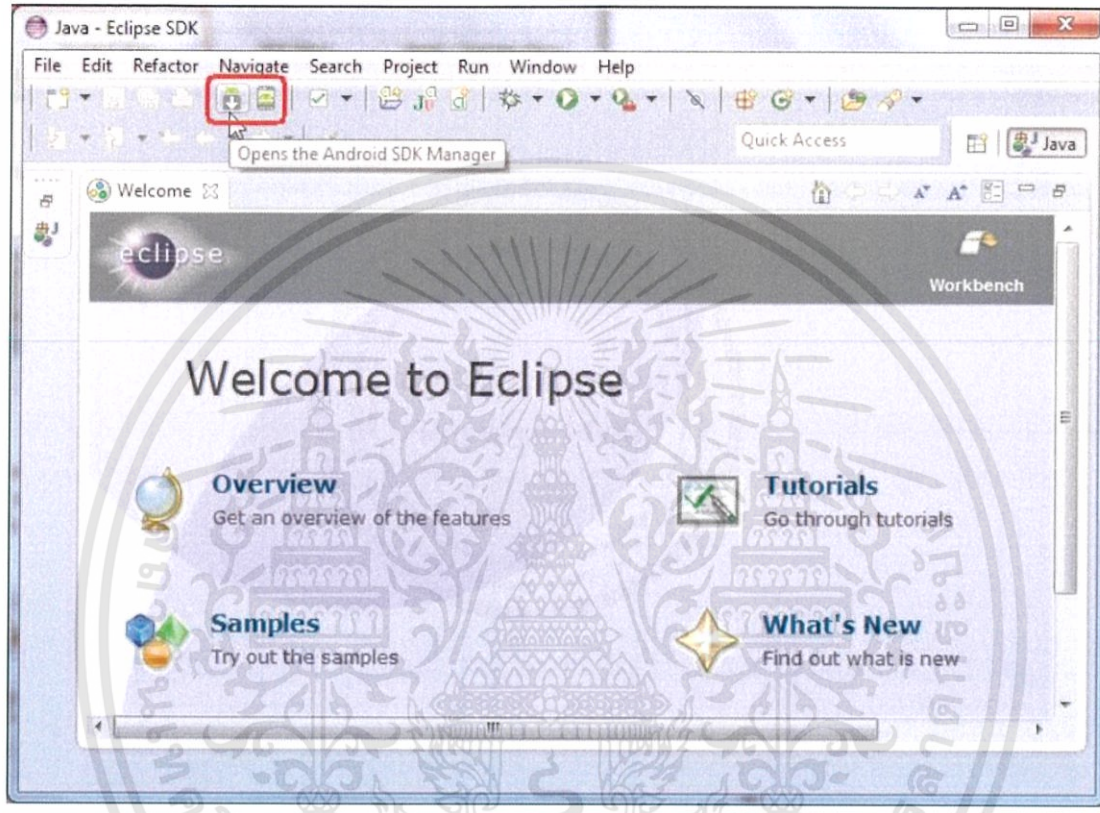
เมื่อการดาวน์โหลดและติดตั้งเสร็จสิ้นจะแสดงคำว่า Done loading packages จากนั้นสามารถทำการปิดหน้าจอนี้ได้ทันที



รูปที่ 3.31 แสดงข้อความแจ้งว่าการติดตั้งเสร็จสิ้นแล้ว

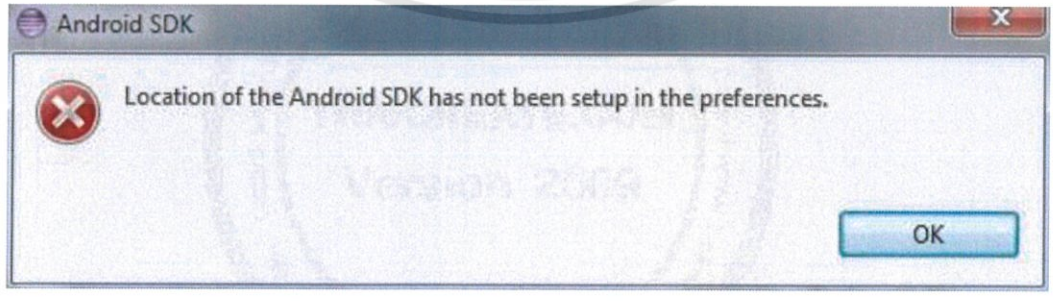
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อทำการติดตั้งเสร็จเรียบร้อยต้องทำการปรับแต่ง Environment ของ Android SDK บนโปรแกรม Eclipse โดยการเปิดโปรแกรม Eclipse แล้วเลือกที่ Open the Android SDK Manager ดังภาพ



รูปที่ 3.32 คลิกที่ Open the Android SDK Manager

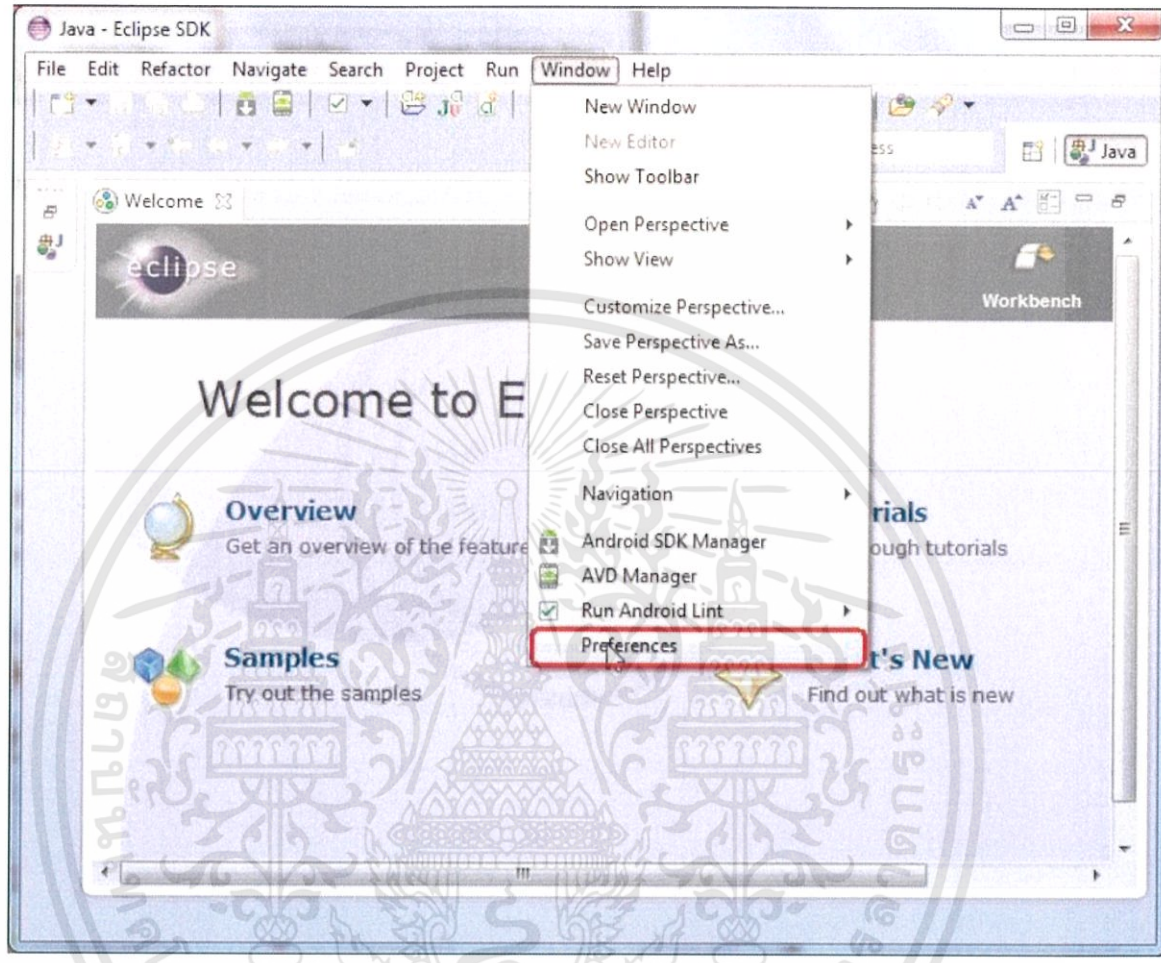
กรณีที่ข้อผิดพลาด “Location of the Android SDK has not been set up in the preferences” ให้ทำการแก้ไขปัญหาดังนี้



รูปที่ 3.33 กล่องข้อความเตือนเมื่อเกิดข้อผิดพลาด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

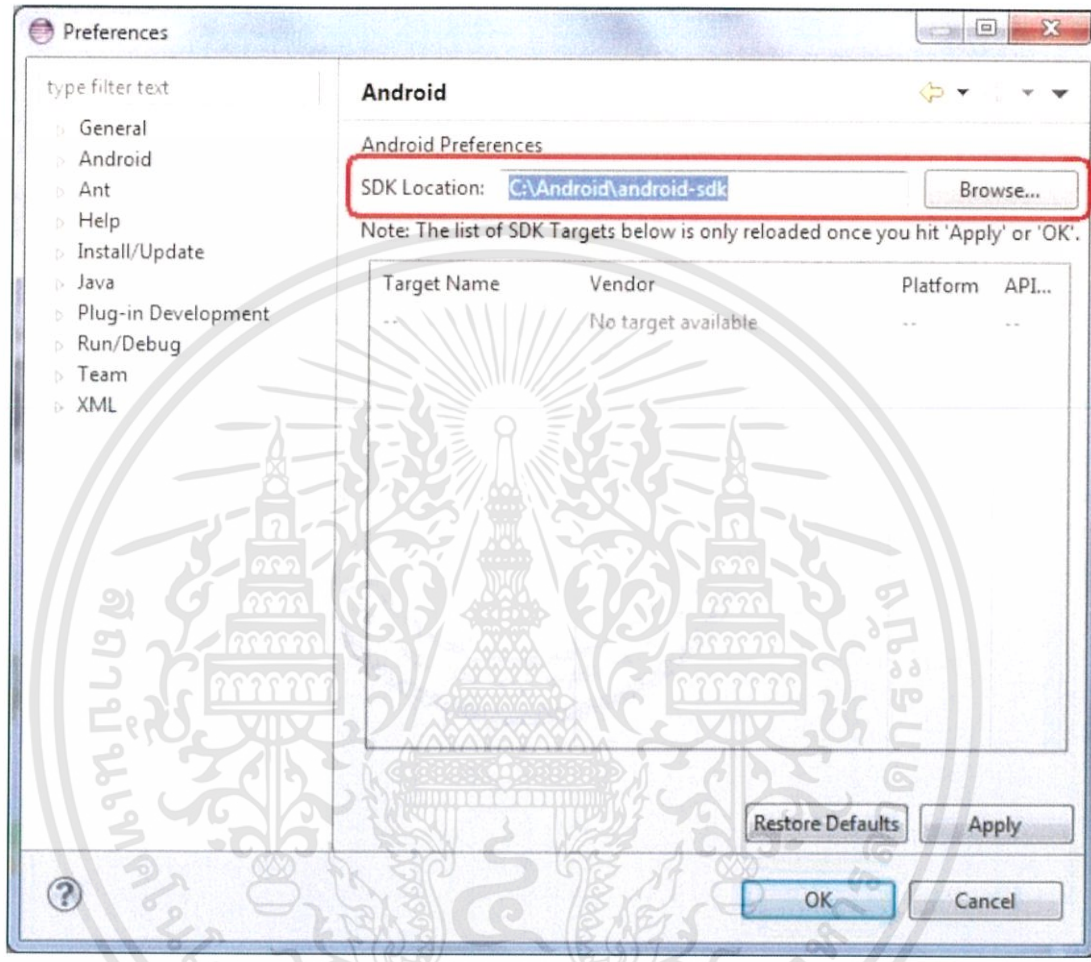
➢ เลือกที่ Window > Preferences



รูปที่ 3.34 เลือก Window > Preferences

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

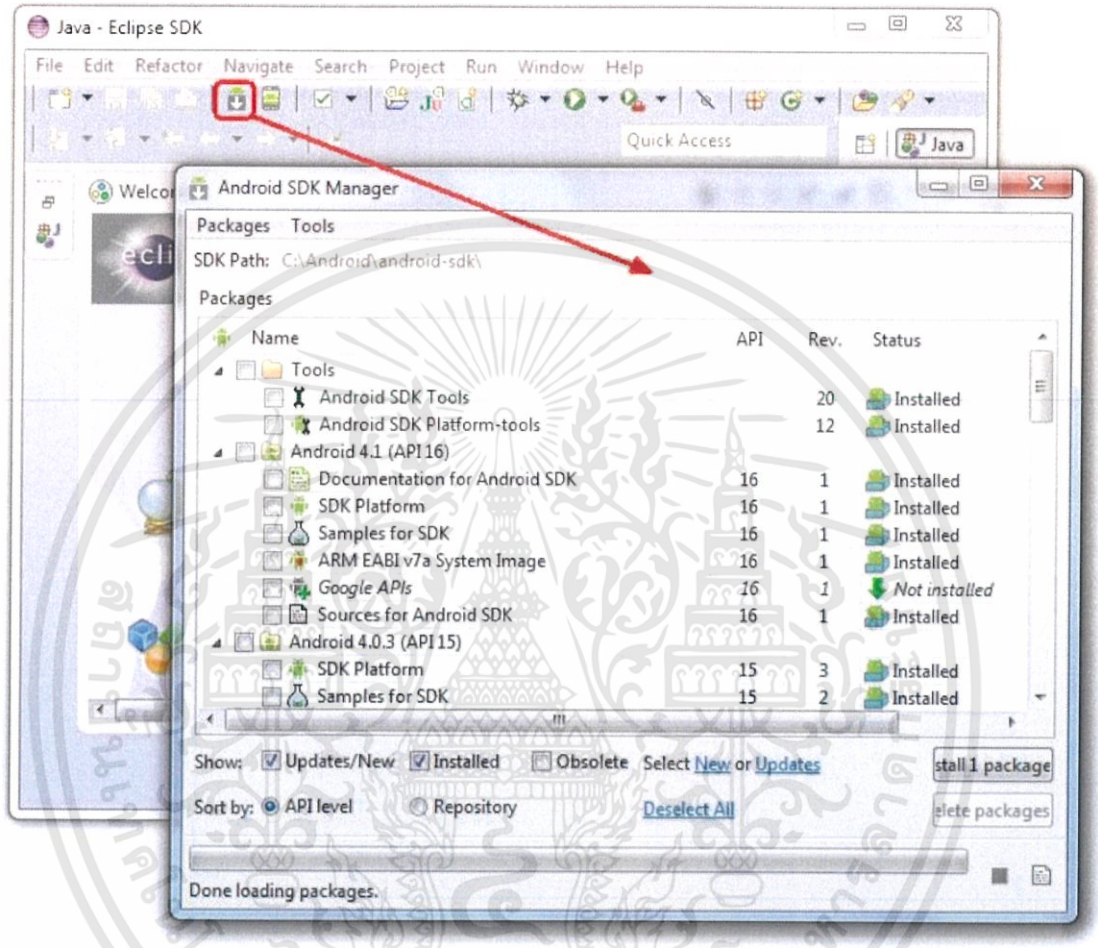
- กำหนด path ที่ได้ทำการติดตั้ง Android SDK ไว้ ในที่นี้คือ C:\Android\android-sdk\ จากนั้นกดปุ่ม OK



รูปที่ 3.35 กำหนด path ที่เราได้ติดตั้ง Android SDK ไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยเมื่อทำตามขั้นตอนเสร็จสิ้นแล้ว โปรแกรม Eclipse ก็จะพร้อมใช้งานได้ตามปกติ โดยปรากฏหน้าจอของ Android SDK Manager ขึ้นมา

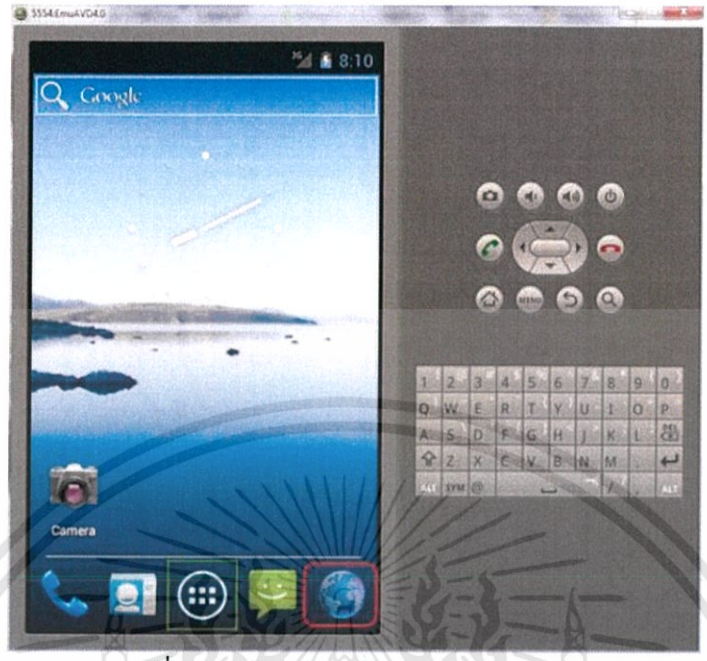


รูปที่ 3.36 การทำงานที่ถูกต้องของโปรแกรม

3.3.4 Android Virtual Device Manager (Emulator)

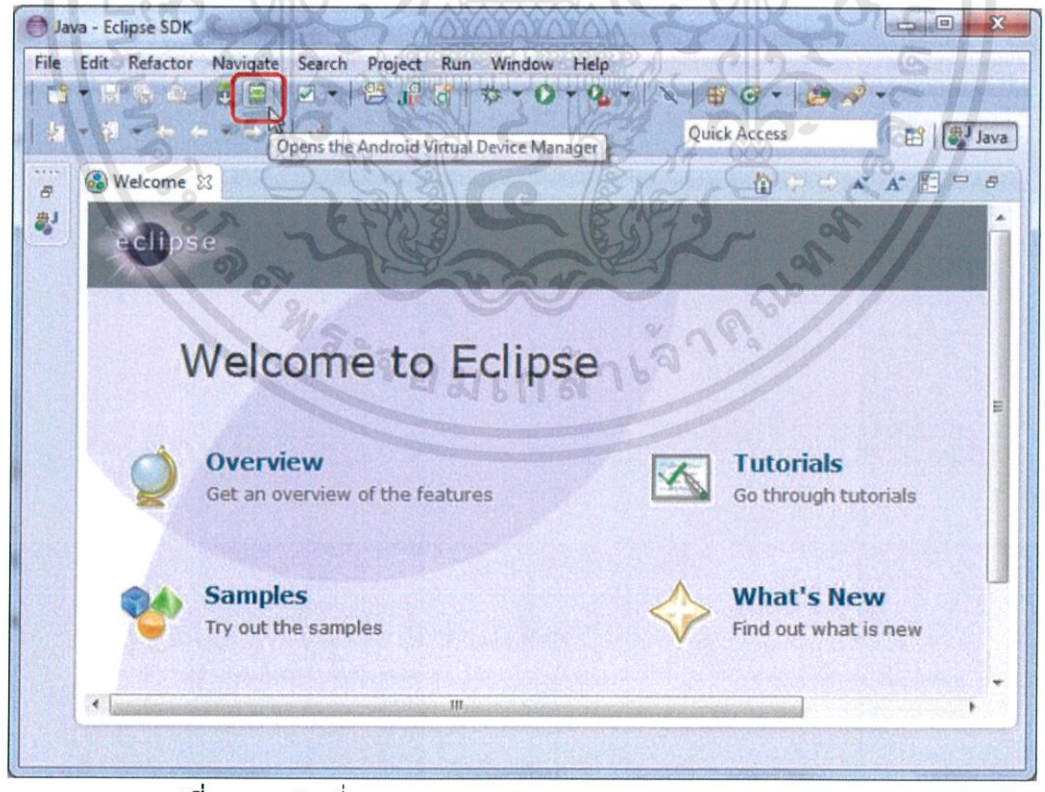
เป็นการจำลองสมาร์ทโฟนระบบปฏิบัติการแอนดรอยด์สำหรับใช้ในการทดสอบแอปพลิเคชันที่เขียนขึ้น โดยตัว Emulator จะจำลอง Virtual OS ของแอนดรอยด์ในรุ่นต่างๆ มาไว้บนเครื่องคอมพิวเตอร์ที่ใช้งาน และเราสามารถใช้ Emulator ของแอนดรอยด์นี้ในการทดสอบแอปพลิเคชัน โดย Android Virtual Device Manager นี้เราได้ทำการติดตั้งมาพร้อมกับการติดตั้งของ Android SDK ซึ่งสามารถเรียกใช้งานได้ในทันที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.37 ตัวอย่างหน้าจอของ Emulator

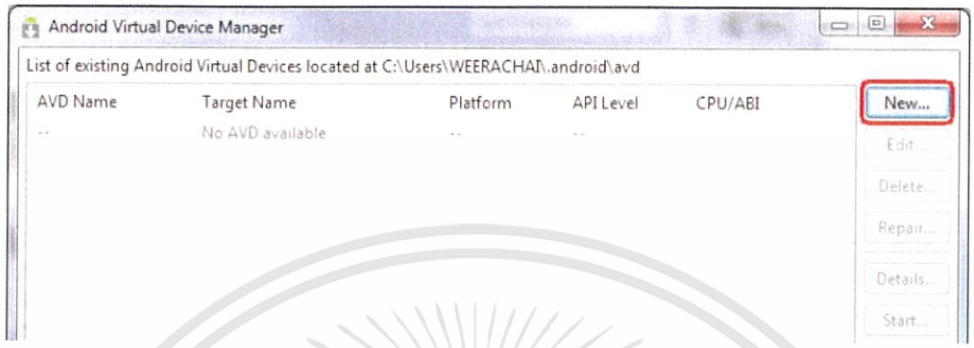
การเรียกใช้งาน Android Virtual Device Manager ให้คลิกที่ Open the Android Virtual Device Manager ดังภาพ



รูปที่ 3.38 คลิกที่ Open the Android Virtual Device Manager

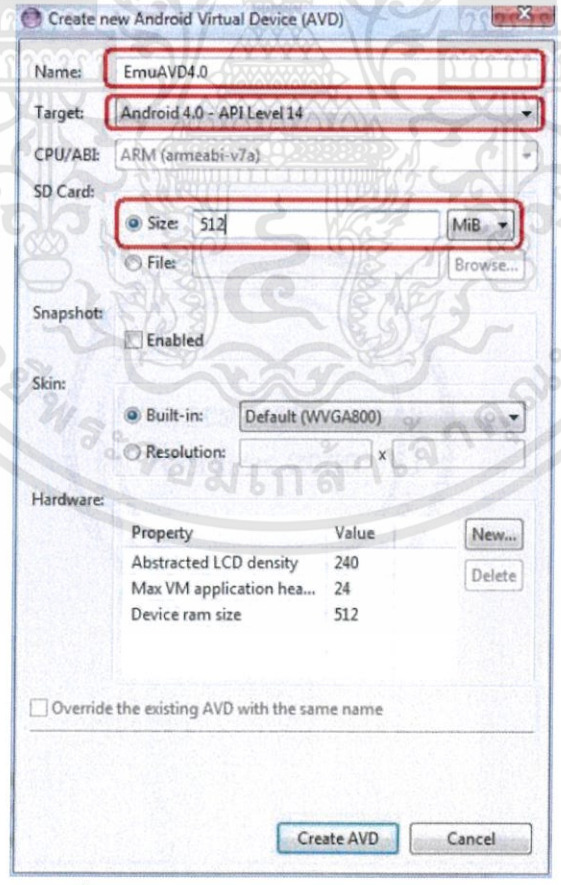
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อเข้าสู่หน้าจอของ Android Virtual Device Manager ให้กดที่ปุ่ม New เพื่อสร้าง Emulator ขึ้นมาใหม่



รูปที่ 3.39 กดปุ่ม New เพื่อสร้าง Emulator

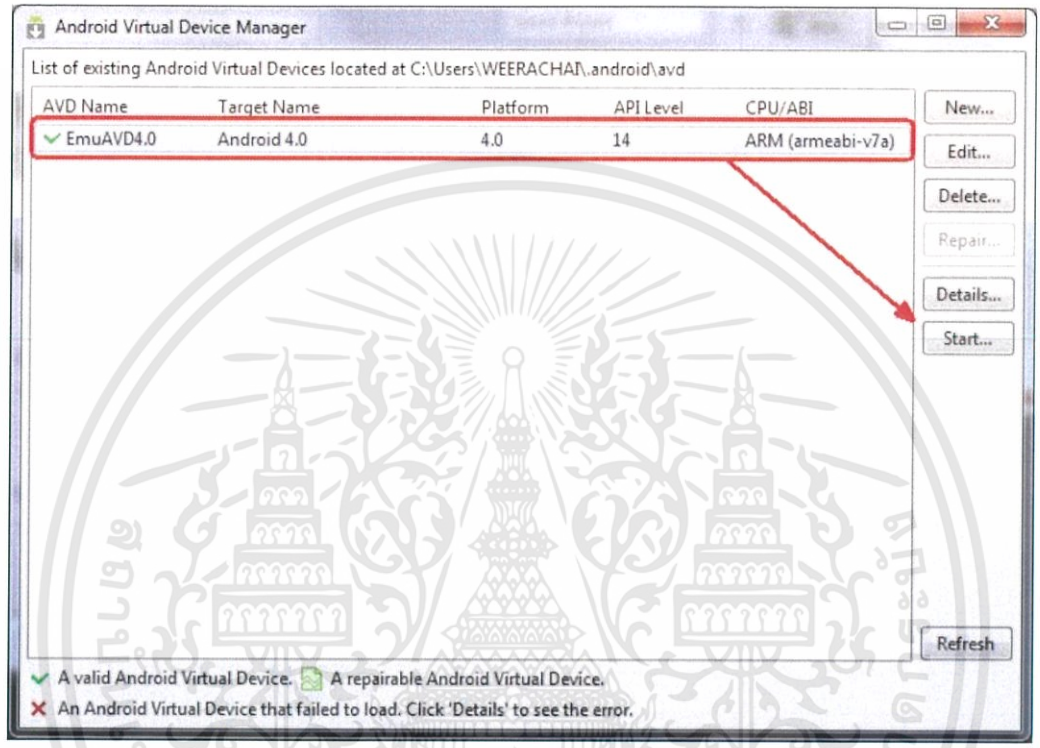
ในช่องแรกให้กำหนดรุ่นของ Android Virtual Device จากนั้นในช่องที่สองจะเป็นรุ่นของระบบปฏิบัติการแอนดรอยด์ จากนั้นทำการกำหนดค่าอื่นๆ เมื่อกำหนดเสร็จสิ้นให้กดปุ่ม Create AVD



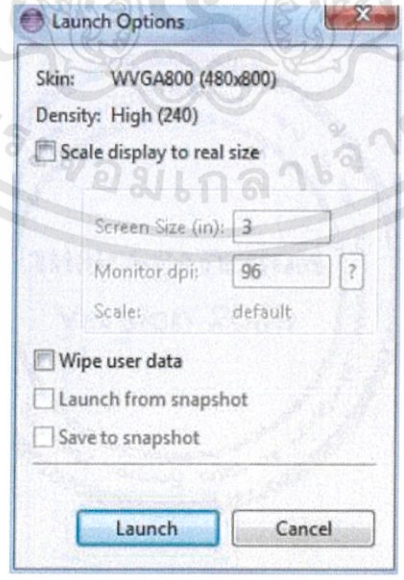
รูปที่ 3.40 หน้าจอกำหนดค่าของ Emulator

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะได้ Android Virtual Device หรือ Emulator ของแอนดรอยด์ขึ้นมาดังภาพ การใช้งานให้เลือก Android Virtual Device ที่ต้องการจำลองจากนั้นกดปุ่ม Start จะมีหน้าจอแสดงรายละเอียดของ Emulator ให้กดปุ่ม Launch



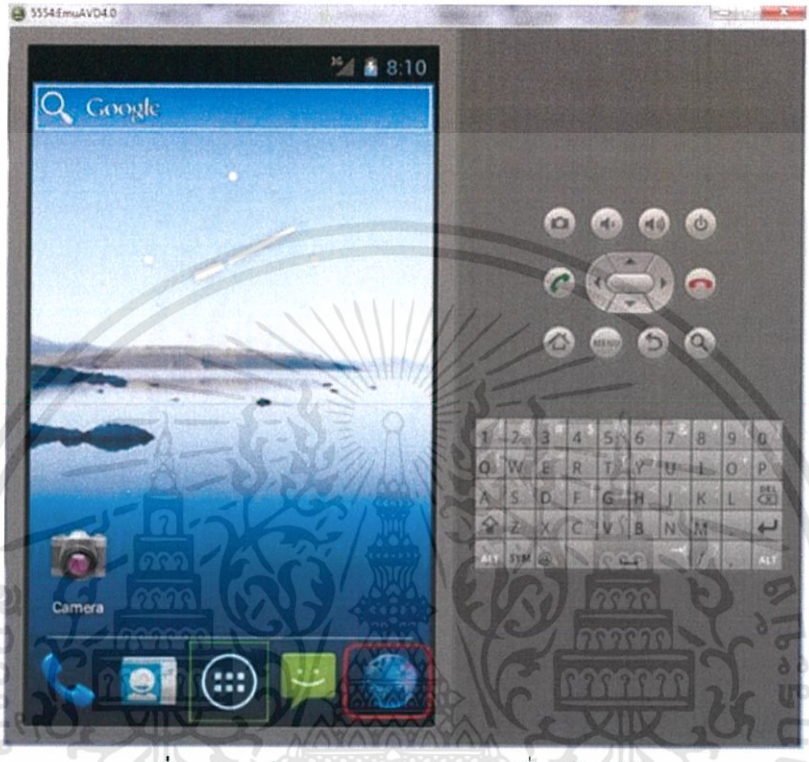
รูปที่ 3.41 เลือกรายการการจำลอง Emulator ที่ต้องการ



รูปที่ 3.42 หน้าจอแสดงรายละเอียด Emulator ให้กด Launch เพื่อเริ่มการจำลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลังจากการโหลดข้อมูลเสร็จสิ้นจะปรากฏหน้าจอดังรูป ซึ่งสามารถทำการทดสอบแอปพลิเคชัน หรือทดสอบอื่นๆ ได้ตามต้องการเสมือนการทำงานบนสมาร์ทโฟนระบบปฏิบัติการแอนดรอยด์



รูปที่ 3.43 หน้าจอของ Emulator เมื่อพร้อมใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4 ขั้นตอนการประกอบรถหุ่นยนต์ POP-BOT XT Lite

1. นำชุดเฟืองขับมอเตอร์ยึดเข้ากับถาดรองกระบะถ่านโดยใช้สกรูหัวตัดขนาด 3x8 มม. ขันยึดในตำแหน่งดังรูป

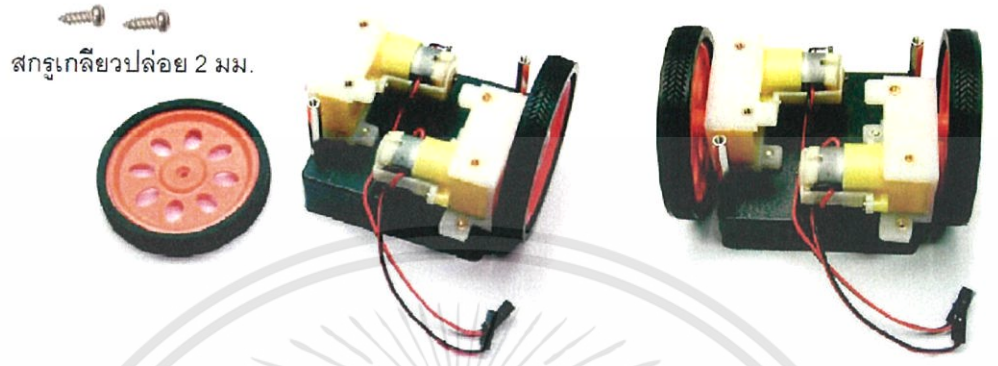
รูปที่ 3.44 ยึดมอเตอร์เข้ากับกระบะถ่าน

2. นำเสารองโลหะขนาด 32 มม. ยึดเข้ากับถาดรองกระบะถ่านแล้วใช้สกรูหัวตัดขนาด 3x8 มม. ขันยึดในตำแหน่งดังรูป

รูปที่ 3.45 ติดเสารองโลหะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. นำล้อพลาสติกพร้อมยางเสียบเข้ากับแกนมอเตอร์สีชมพูแล้วใช้สกรูเกลียวปล่อย 2 มม. ขันยึด



รูปที่ 3.46 ติดล้อเข้ากับมอเตอร์

4. นำชุดมอเตอร์ยึดเข้ากับฐานพลาสติกของหุ่นยนต์ แล้วใช้สกรูขนาด 3x6 มม.ขันยึดในตำแหน่งดังรูปจะสังเกตได้ว่าการวางมอเตอร์ทั้งสองด้านไม่สมมาตรกันแต่ล้อพลาสติกจะอยู่ตรงกลางพอดี

รูปที่ 3.47 ชุดมอเตอร์ยึดเข้ากับฐานพลาสติกของหุ่นยนต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. นำแผงวงจรตรวจจับแสงสะท้อนอินฟราเรด ZX-03 ร้อยด้วยสกรู 3x15 มม. ยึดเข้ากับแท่งต่อ 5 รู ทำเหมือนกัน 2 ชุดสำหรับด้านซ้ายและขวาของหุ่นยนต์

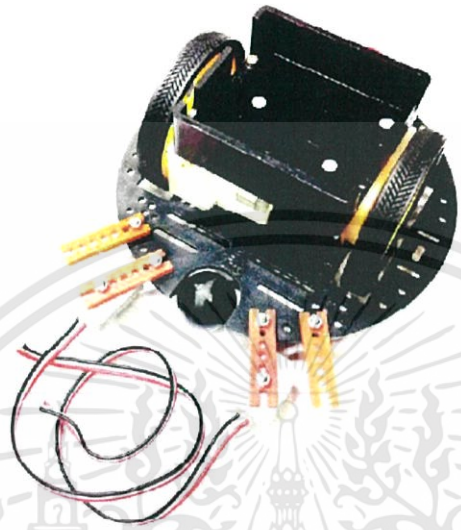
รูปที่ 3.48 แผงวงจรตรวจจับแสงสะท้อนอินฟราเรด ZX-03

6. ยึดแผงวงจรตรวจจับแสงสะท้อนอินฟราเรดจากขั้นตอนที่ 5 เข้าที่ด้านหน้าของหุ่นยนต์ทั้งทางซ้ายและขวาโดยใช้สกรูขนาด 3x10 มม. ขันยึด

รูปที่ 3.49 ต่อแผงวงจรตรวจจับแสงสะท้อน

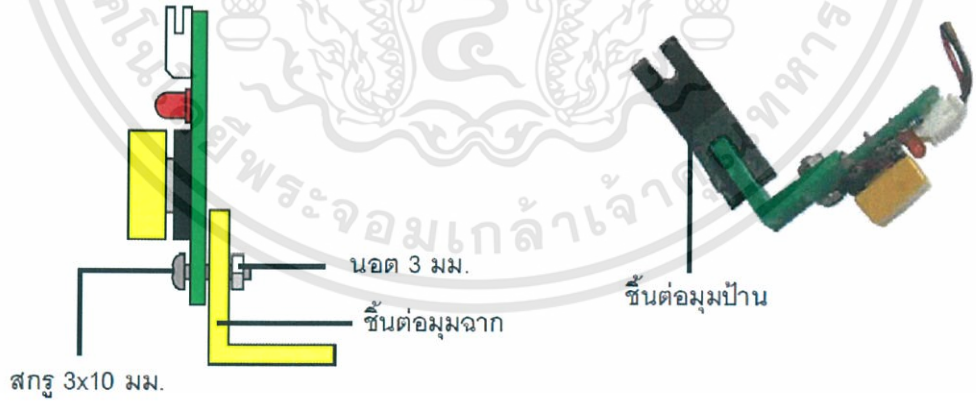
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7. นำแท่งต่อ 5 รูอีกสองอันมายึดเข้าที่ด้านหน้าของฐานหุ่นยนต์ดังรูป โดยใช้สกรู 3x10 มม.ขันยึดติดตั้ง ทั้งสองข้างของหุ่นยนต์ดังรูป



รูปที่ 3.50 ต่อแท่งต่อ 5 รู

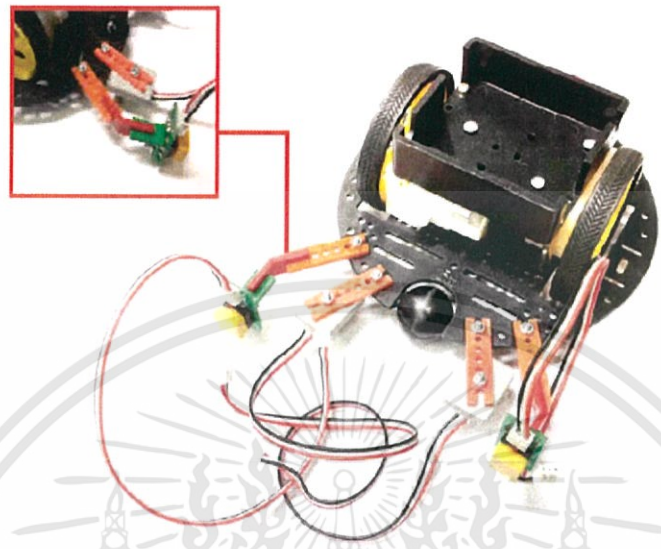
8. นำแผงวงจรสวิตช์ยึดเข้ากับชิ้นต่อมุมฉากโดยใช้สกรู 3x10 มม.และนอต 3 มม.ขันยึด จากนั้นนำชิ้นต่อมุมป้านเสียบเข้ากับปลายอีกด้านหนึ่งของชิ้นต่อมุมฉากทำเหมือนกัน 2 ชุด



รูปที่ 3.51 ต่อวงจรสวิตช์

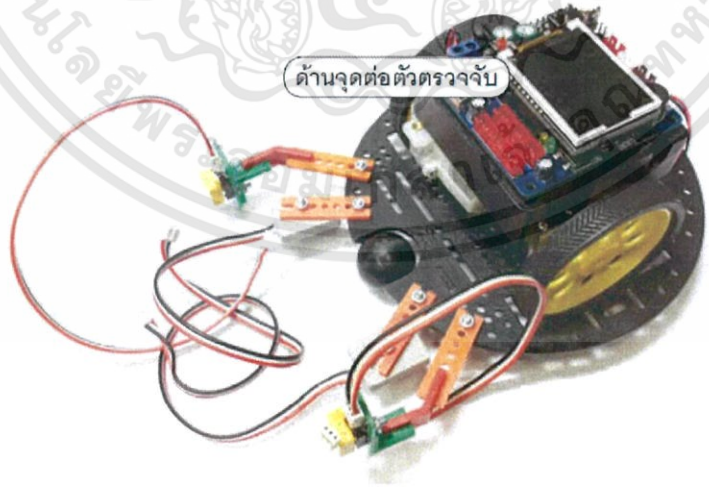
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

9. นำชุดแผงวงจรสวิตช์ติดตั้งเข้ากับฐานของหุ่นยนต์โดยเสียบเข้าไปที่แท่งต่อดังรูป



รูปที่ 3.52 ต่อแผงวงจรสวิตช์กับฐานของหุ่นยนต์

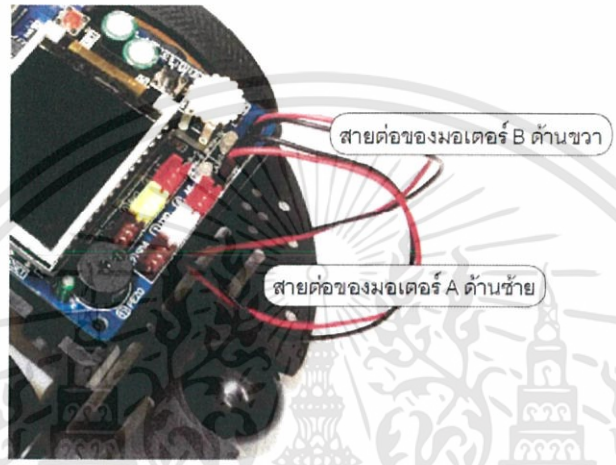
10. นำแผงวงจร POP-XT วางลงบนกล่องรองกระบะถ่าน โดยหันด้านจุดต่อตัวตรวจจับไว้ด้านหน้าของหุ่นยนต์ดังรูป



รูปที่ 3.53 ต่อแผงวงจร POP-XT วางลงกล่องรองกระบะถ่าน

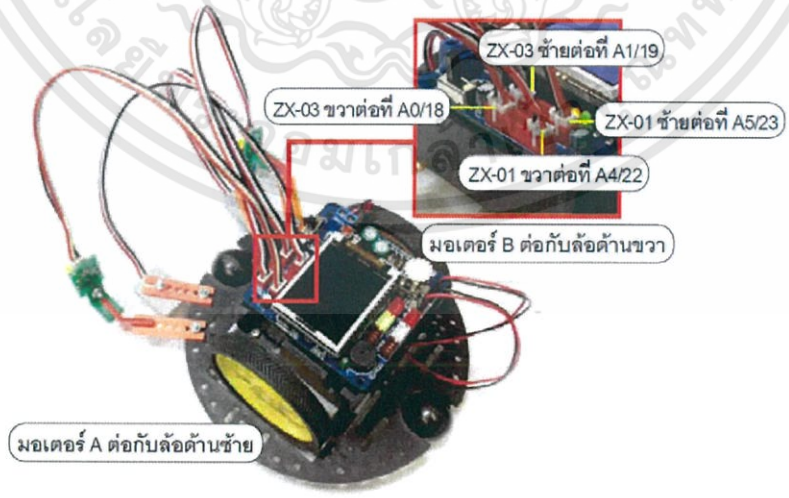
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

11. นำสายมอเตอร์ต่อเข้ากับจุดต่อมอเตอร์โดยให้ตรวจสอบขั้วของมอเตอร์จากการหมุนล้อเมื่อต่อสายแล้ว หมุนล้อไปด้านหน้า LED แสดงสถานะของมอเตอร์ควรติดเป็นสีเขียวหมุนล้อกลับไปทางด้านหลัง ในทิศตรงข้าม LED ควรติดเป็นสีแดง ถ้าไม่ใช่ให้สลับขั้วต่อสายมอเตอร์ จนกระทั่ง LED แสดงสีที่ถูกต้องทั้งสองล้อ



รูปที่ 3.54 ต่อสายมอเตอร์

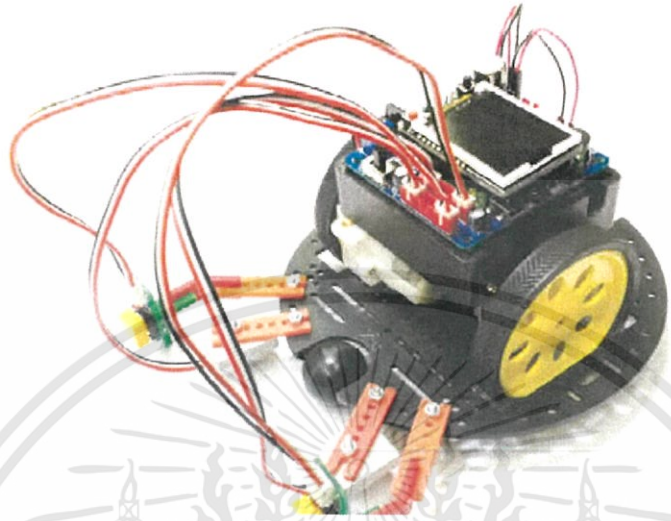
12. จากนั้นต่อสายของแผงวงจรตรวจจับแสงสะท้อน ZX-03 ด้านซ้ายเข้าที่จุดต่อ A1/19 และต่อ ZX-03 ทางด้านขวาเข้าที่จุดต่อ A0/18 ต่อสายจากแผงวงจรสวิตช์ด้านซ้ายเข้ากับจุดต่อ A5/23 และต่อสายจากแผงวงจรสวิตช์ด้านขวาเข้ากับจุดต่อ A4/22



รูปที่ 3.55 ต่อสายของวงจรเซนเซอร์ต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

13. จะได้หุ่นยนต์ POP-BOT XT ที่พร้อมสำหรับการเขียนโปรแกรมต่อไป



รูปที่ 3.56 POP-BOT XT เสร็จสมบูรณ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

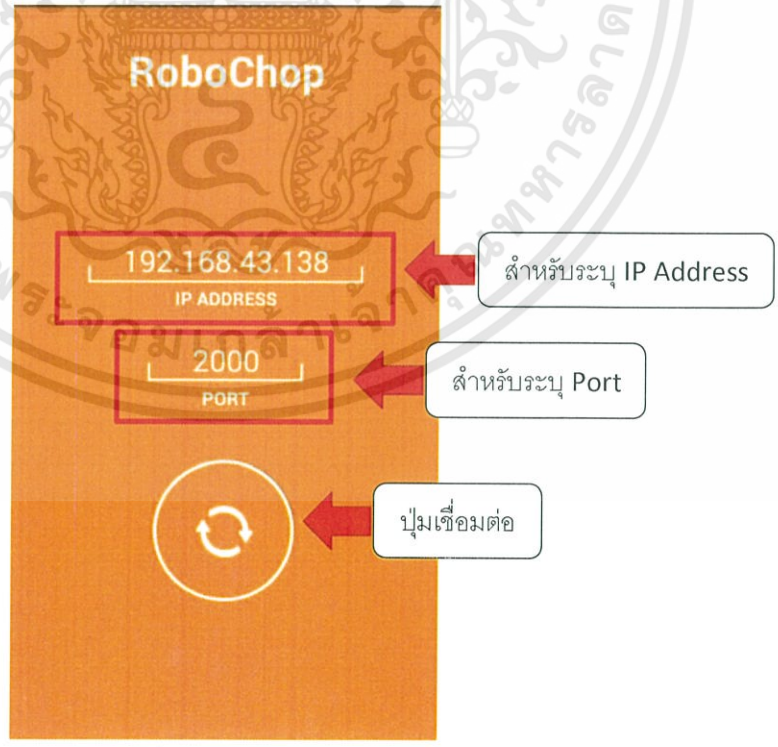
3.5 การออกแบบส่วนติดต่อกับผู้ใช้งาน (User Interface)

ในโครงงานปัญหาพิเศษนี้จะมีส่วนติดต่อกับผู้ใช้งานอยู่สองส่วนคือ ส่วนของแอปพลิเคชันสำหรับการควบคุมบนสมาร์ทโฟนระบบปฏิบัติการแอนดรอยด์ และส่วนของรถหุ่นยนต์ที่มีหน้าจอ LCD สำหรับใช้แสดงสถานะและข้อมูลต่างๆ

3.5.1 การออกแบบส่วนของแอปพลิเคชันบนสมาร์ทโฟน

3.5.1.1 หน้าเมน (Main)

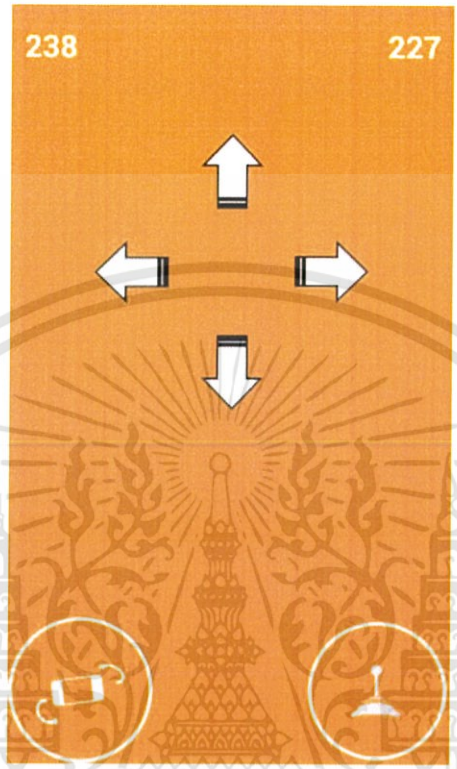
หน้าเมน (Main) นี้จะเป็นหน้าจอแสดงผลหน้าแรกหลังจากที่ทำการเปิดแอปพลิเคชันขึ้นมา โดยหน้าเมนจะใช้สำหรับการกำหนดค่า IP Address และ Port สำหรับการเชื่อมต่อระหว่างสมาร์ทโฟนกับรถหุ่นยนต์ โดยค่าของ IP Address และ Port ที่นำมากำหนดค่าสามารถดูได้จากหน้าจอของรถหุ่นยนต์ และเมื่อผู้ใช้ได้ทำการกำหนดค่าครบถ้วนแล้ว ผู้ใช้สามารถกดปุ่ม “เชื่อมต่อ” เพื่อทำการเชื่อมต่อได้ทันที โดยถ้าการเชื่อมต่อสำเร็จแอปพลิเคชันเข้าสู่หน้าการควบคุมด้วยปุ่มกดต่อไป แต่หากการเชื่อมต่อไม่สำเร็จจะมีการแจ้งเตือนผู้ใช้งานถึงสาเหตุและจะรอรับการเชื่อมต่ออีกครั้ง



รูปที่ 3.57 หน้าเมน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5.1.2 หน้าการควบคุมด้วยปุ่มกด (Button)



รูปที่ 3.58 หน้าการควบคุมด้วยปุ่มกด

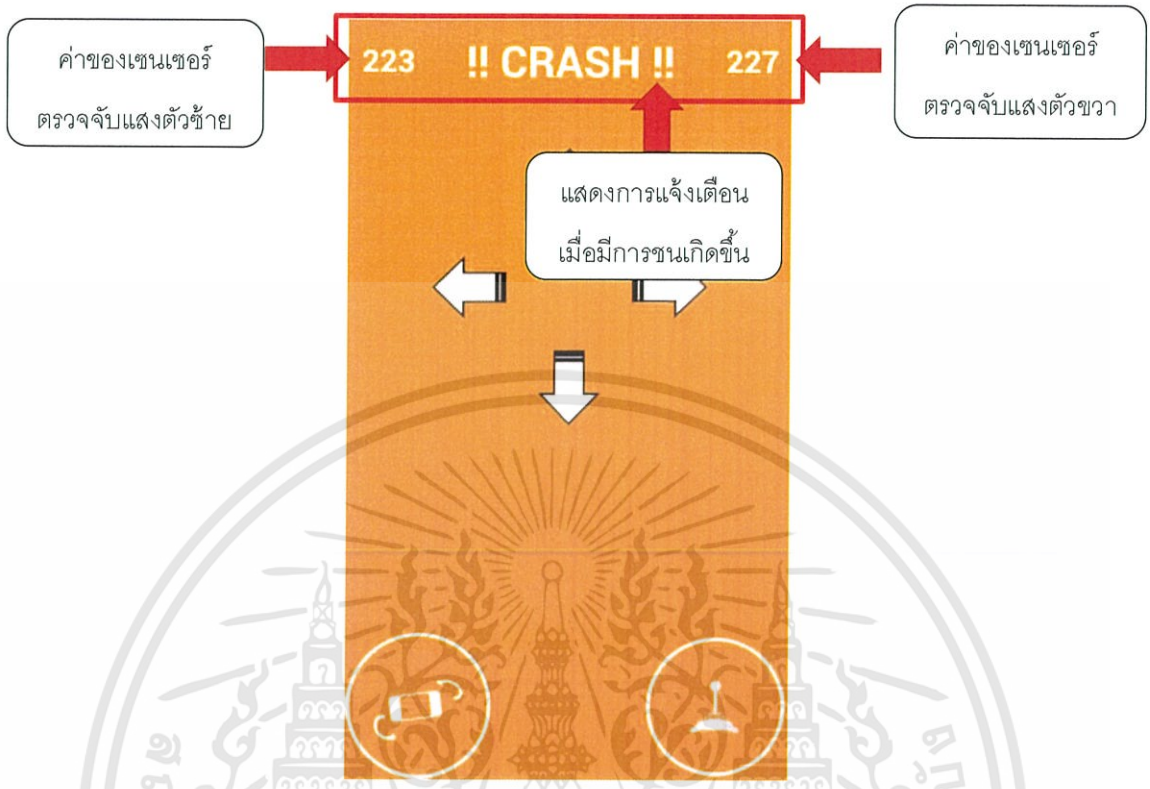
หลังจากทำการเชื่อมต่อที่หน้าเมนสำเร็จได้มีการออกแบบให้แอปพลิเคชันเข้าสู่หน้าจอของการควบคุมด้วยปุ่มกด ในรูปแบบการควบคุมด้วยปุ่มกดหน้าจอของการควบคุมรถหุ่นยนต์จะถูกแบ่งออกเป็น 3 ส่วนหลักๆ คือ

- ส่วนแสดงผลค่าของเซนเซอร์ที่รับมาจากรถหุ่นยนต์

  ส่วนของการแสดงผลค่าของเซนเซอร์ตรวจจับแสงที่ติดอยู่กับตัวรถหุ่นยนต์ โดยจะแสดงผลค่าของเซนเซอร์ตัวซ้ายที่มุมบนซ้ายของหน้าจอสมาร์ทโฟน และแสดงผลค่าของเซนเซอร์ตัวขวาที่มุมบนขวาของหน้าจอสมาร์ทโฟน

  ส่วนของการแสดงผลค่าของเซนเซอร์ตรวจจับการชนที่ติดอยู่กับตัวรถหุ่นยนต์ โดยเมื่อเซนเซอร์ตรวจจับการชนเกิดการเปลี่ยนแปลงค่าจะทำการแจ้งเตือนผู้ใช้งานโดยแสดงข้อความ “CRASH” ที่ตรงกลางด้านบนของหน้าจอสมาร์ทโฟน นอกจากนี้ยังแจ้งเตือนด้วยการสั่นสมาร์ทโฟนและมีเสียงเตือนอีกด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

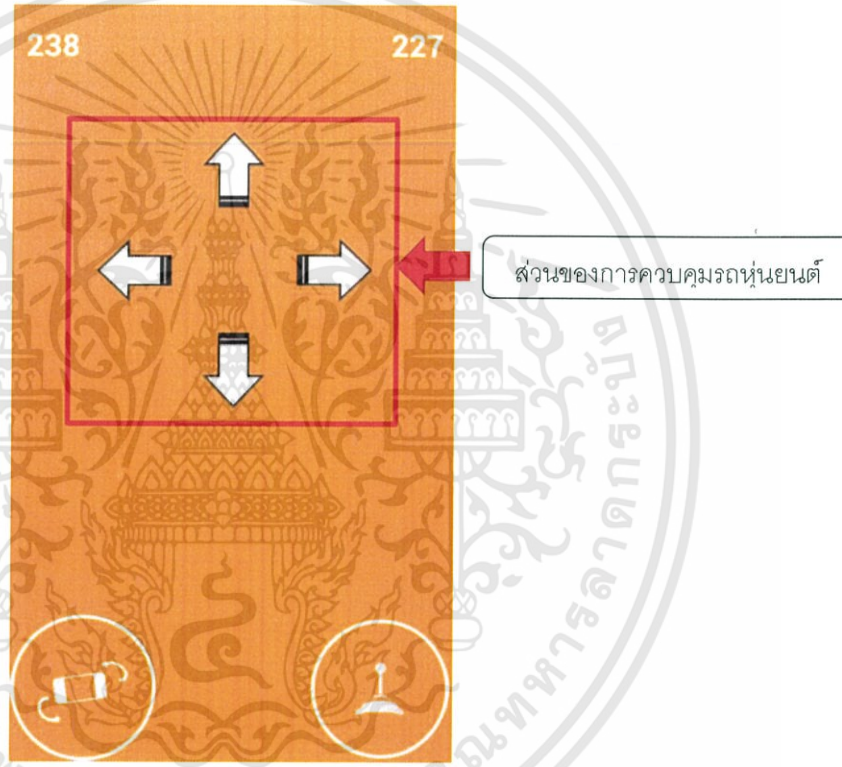


รูปที่ 3.59 ส่วนของการแสดงผลค่าของเซนเซอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

➢ ส่วนของปุ่มสำหรับใช้ควบคุมรถหุ่นยนต์

ส่วนของการควบคุมจะเป็นปุ่มกดรูปลูกศรแบบ 4 ทิศทาง คือ ขึ้น ลง หมุนซ้าย และหมุนขวา โดยปุ่มขึ้นจะเป็นการควบคุมให้รถหุ่นยนต์เคลื่อนที่ไปทางด้านหน้า ปุ่มลงจะเป็นการควบคุมให้รถหุ่นยนต์เคลื่อนที่ไปทางด้านหลัง ปุ่มหมุนซ้ายจะเป็นการควบคุมให้รถหุ่นยนต์หมุนไปทางด้านซ้าย และปุ่มหมุนขวาจะเป็นการควบคุมให้รถหุ่นยนต์หมุนไปทางด้านขวา

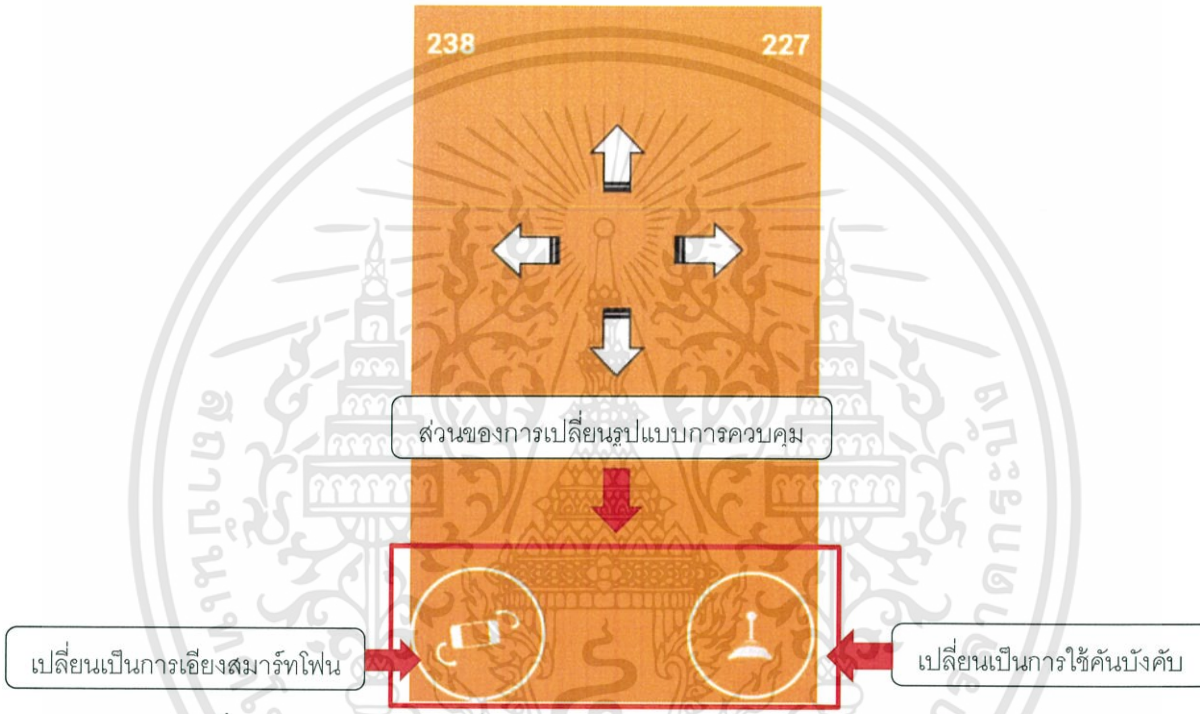


รูปที่ 3.60 ส่วนของการควบคุมรถหุ่นยนต์ในหน้าการควบคุมด้วยปุ่มกด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

➢ ส่วนของการเปลี่ยนรูปแบบการควบคุม

ส่วนนี้เป็นส่วนที่ใช้สำหรับเปลี่ยนรูปแบบการควบคุมเป็นแบบอื่นตามที่ผู้ใช้งานต้องการ โดยในรูปแบบการควบคุมด้วยปุ่มกดสามารถเปลี่ยนรูปแบบการควบคุมเป็นแบบอื่นได้ 2 แบบ คือ การควบคุมด้วยการเอียงสมาร์ทโฟน และการควบคุมด้วยการใช้คันบังคับ



รูปที่ 3.61 ส่วนของการเปลี่ยนรูปแบบการควบคุมในหน้าการควบคุมด้วยปุ่มกด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5.1.3 หน้าการควบคุมด้วยการเอียงสมาร์ทโฟน (Tilt)

รูปที่ 3.62 หน้าการควบคุมด้วยการเอียงสมาร์ทโฟน

ผู้ใช้งานจะเข้าสู่หน้าจอการควบคุมด้วยการเอียงสมาร์ทโฟนได้ด้วยการเลือกรูปแบบการควบคุมนี้จากหน้าการควบคุมด้วยปุ่มกดหรือจากหน้าการควบคุมด้วยคันบังคับ ในรูปแบบการควบคุมด้วยการเอียงสมาร์ทโฟนนี้จะใช้การตรวจจับความลาดเอียงของเซนเซอร์ในสมาร์ทโฟนเป็นตัวควบคุม รถหุ่นยนต์ หน้าจอของการควบคุมรถหุ่นยนต์จะถูกแบ่งออกเป็น 3 ส่วนหลักๆ คือ

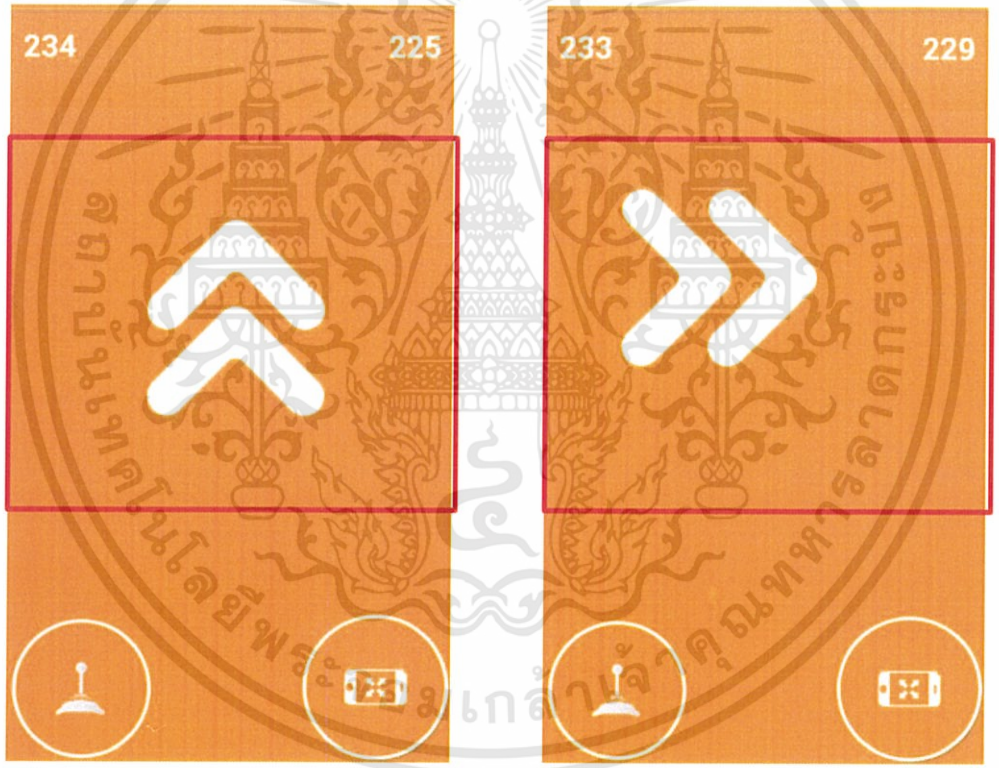
- ส่วนแสดงผลค่าของเซนเซอร์ที่รับมาจากรถหุ่นยนต์

  เป็นส่วนการแสดงผลค่าของเซนเซอร์ตรวจจับแสง และเซนเซอร์ตรวจจับการชน เช่นเดียวกันกับหน้าการควบคุมด้วยปุ่มกด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

➢ ส่วนของการแสดงผลทิศทางที่ใช้ควบคุมรถหุ่นยนต์

ส่วนของการควบคุมจะเป็นใช้การเอียงสมาร์ทโฟนแบบ 4 ทิศทาง คือ เอียงไปด้านบน เอียงไปด้านล่าง เอียงไปด้านซ้ายและเอียงไปด้านขวา โดยการเอียงไปด้านบนจะเป็นการควบคุมให้รถหุ่นยนต์เคลื่อนที่ไปทางด้านหน้า การเอียงไปด้านล่างจะเป็นการควบคุมให้รถหุ่นยนต์เคลื่อนที่ไปทางด้านหลัง การเอียงไปด้านซ้ายจะเป็นการควบคุมให้รถหุ่นยนต์หมุนไปทางด้านซ้าย และการเอียงไปด้านขวาจะเป็นการควบคุมให้รถหุ่นยนต์หมุนไปทางด้านขวา โดยเมื่อเอียงสมาร์ทโฟนมากขึ้นความเร็วของรถหุ่นยนต์จะมากขึ้นตามไปด้วย และจะมีรูปภาพเคลื่อนไหวแสดงบอกทิศทางที่กำลังควบคุมอยู่

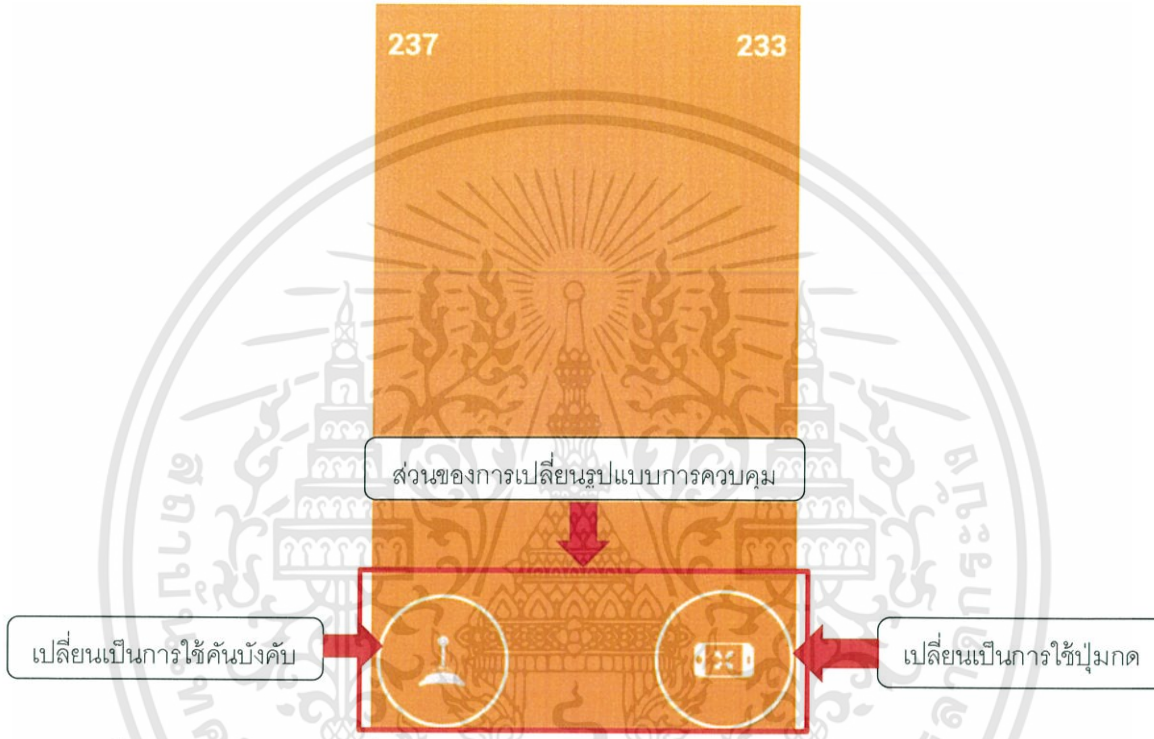


รูปที่ 3.63 การแสดงผลเมื่อเอียงสมาร์ทโฟนไปด้านหน้าและด้านขวาตามลำดับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

➢ ส่วนของการเปลี่ยนรูปแบบการควบคุม

ส่วนนี้เป็นส่วนที่ใช้สำหรับเปลี่ยนรูปแบบการควบคุมเช่นเดียวกับหน้าการควบคุมด้วยปุ่มกด แต่ในรูปแบบการควบคุมด้วยการเอียงสมาร์ทโฟนจะสามารถเปลี่ยนรูปแบบการควบคุมเป็นการควบคุมด้วยการใช้ปุ่มกด และการควบคุมด้วยการใช้คันบังคับ



รูปที่ 3.64 ส่วนของการเปลี่ยนรูปแบบการควบคุมในหน้าการควบคุมด้วยการเอียงสมาร์ทโฟน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5.1.4 หน้าการควบคุมด้วยคันบังคับ (Joystick)

รูปที่ 3.65 หน้าการควบคุมด้วยคันบังคับ

ผู้ใช้งานจะเข้าสู่หน้าจอการควบคุมด้วยคันบังคับได้ด้วยการเลือกรูปแบบการควบคุมนี้จากหน้าการควบคุมด้วยปุ่มกดหรือจากหน้าการควบคุมด้วยการเอียงสมาร์ทโฟน ในรูปแบบการควบคุมด้วยคันบังคับนี้จะใช้การลากนิ้วไปในทิศทางต่างๆบนคันบังคับเพื่อทำการควบคุมรถหุ่นยนต์ หน้าจอของการควบคุมรถหุ่นยนต์จะถูกแบ่งออกเป็น 3 ส่วนหลักๆ คือ

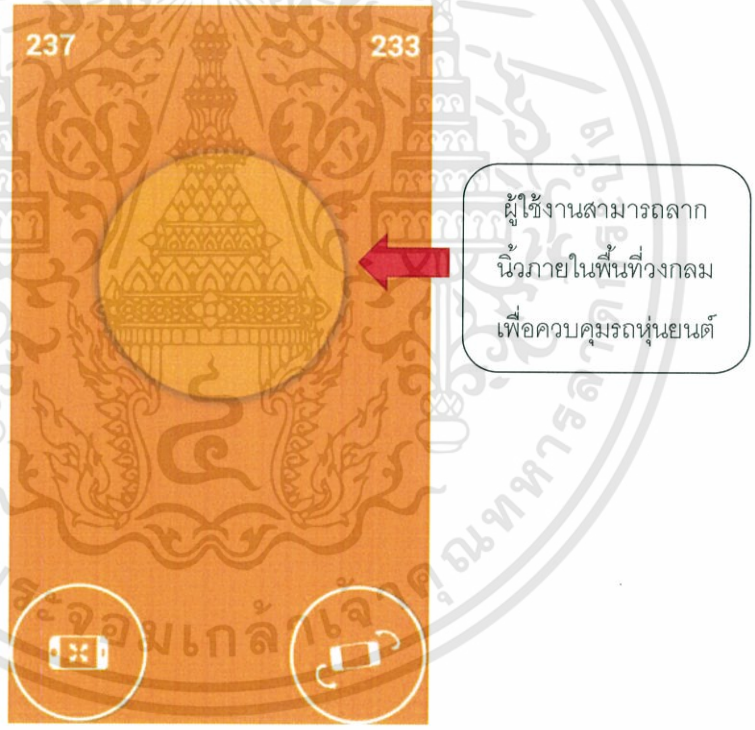
- ส่วนแสดงผลค่าของเซนเซอร์ที่รับมาจากรถหุ่นยนต์

  เป็นส่วนการแสดงผลค่าของเซนเซอร์ตรวจจับแสง และเซนเซอร์ตรวจจับการชน เช่นเดียวกันกับหน้าการควบคุมด้วยปุ่มกดและหน้าการควบคุมด้วยการเอียงสมาร์ทโฟน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

➢ ส่วนของพื้นที่คันบังคับเพื่อใช้ควบคุมรถหุ่นยนต์

ส่วนของการควบคุมด้วยคันบังคับจะใช้การลากนิ้วบนพื้นที่ที่จัดเตรียมไว้สำหรับคันบังคับบนหน้าจอสมาร์ทโฟนซึ่งมีการตรวจจับแบบ 4 ทิศทาง คือ ลากนิ้วไปด้านบน ลากนิ้วไปด้านล่าง ลากนิ้วไปด้านซ้าย และลากนิ้วไปด้านขวา โดยการลากไปด้านบนจะเป็นการควบคุมให้รถหุ่นยนต์เคลื่อนที่ไปทางด้านหน้า การลากไปด้านล่างจะเป็นการควบคุมให้รถหุ่นยนต์เคลื่อนที่ไปทางด้านหลัง การลากไปด้านซ้ายจะเป็นการควบคุมให้รถหุ่นยนต์หมุนไปทางด้านซ้าย และการลากไปด้านขวาจะเป็นการควบคุมให้รถหุ่นยนต์หมุนไปทางด้านขวา โดยถ้าผู้ใช้ไม่ได้ทำการลากคันบังคับหรือแตะนิ้วอยู่ที่จุดศูนย์กลางคันบังคับรถหุ่นยนต์จะไม่เคลื่อนไหว

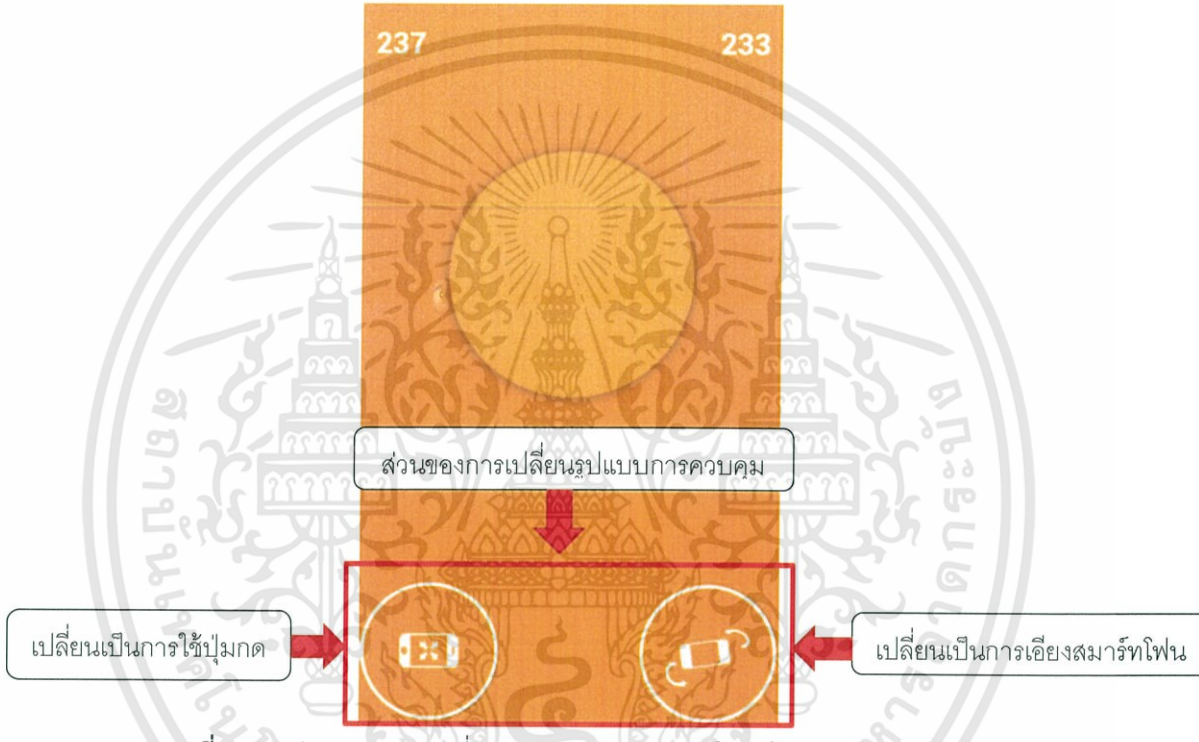


รูปที่ 3.66 ส่วนของพื้นที่คันบังคับสำหรับใช้ควบคุม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

➢ ส่วนของการเปลี่ยนรูปแบบการควบคุม

ส่วนนี้เป็นส่วนที่ใช้สำหรับเปลี่ยนรูปแบบการควบคุมเช่นเดียวกับหน้าการควบคุมด้วยปุ่มกดและหน้าการควบคุมด้วยการเอียงสมาร์ทโฟน แต่ในรูปแบบการควบคุมด้วยคันบังคับจะสามารถเปลี่ยนรูปแบบการควบคุมเป็นการควบคุมด้วยการใช้ปุ่มกด และการควบคุมด้วยการเอียงสมาร์ทโฟน



รูปที่ 3.67 ส่วนของการเปลี่ยนรูปแบบการควบคุมในหน้าการควบคุมด้วยคันบังคับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

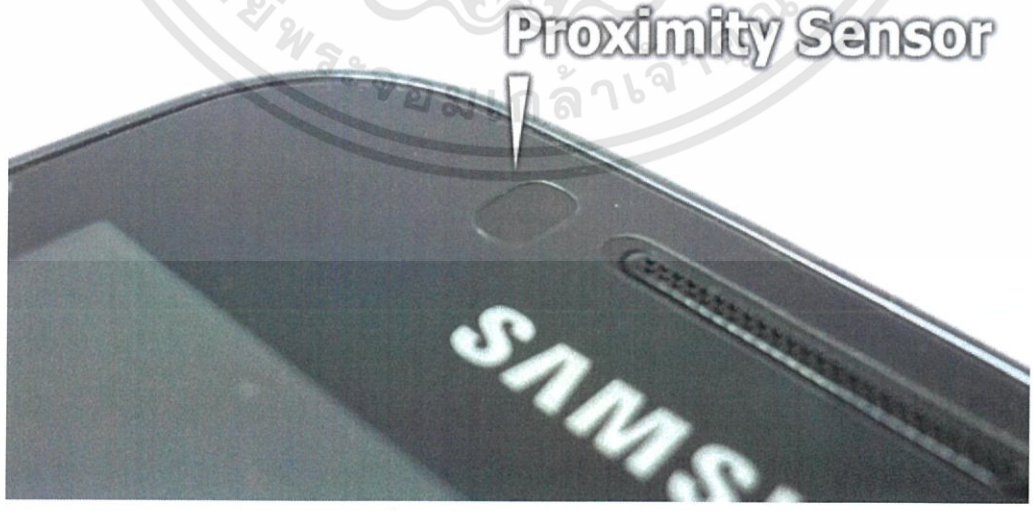
3.6 เซนเซอร์ในแอนดรอยด์

อุปกรณ์แอนดรอยด์จะมีเซนเซอร์ต่างๆที่สามารถให้ผู้เรียกใช้งานได้ ซึ่งการพัฒนาระบบควบหุ่นยนต์จะใช้เซนเซอร์เพื่อบังคับหุ่นยนต์ตามที่เราต้องการ เซนเซอร์ที่ผู้ใช้สารมารถใช้งานได้มีดังนี้

3.6.1 เซนเซอร์วัดระยะทาง (Proximity)

Proximity เป็นเซนเซอร์วัดระยะทางแบบหนึ่งที่ใช้หลักการทางแสงที่จะใช้การสะท้อนของแสงกับวัตถุ โดยจะมีแหล่งกำหนดแสงและตัวรับแสงจากแหล่งกำเนิด ที่รับแสงที่สะท้อนกลับมาโดยบนอุปกรณ์แอนดรอยด์ ก็จะใช้เป็นแสงอินฟราเรด

Proximity มีไว้เวลาแนบหูคุยโทรศัพท์ ทำให้เครื่องพักหน้าจออัตโนมัติเพื่อป้องกันหน้าไปกดโดนจอสำหรับแทบเลตที่ไม่สามารถโทรได้ก็จะไม่มี Proximity ให้ใช้ ค่าที่อ่านได้จาก Proximity จะมีค่าอยู่แค่สองค่าคือ 0 และ X โดย X คือค่าใดๆของ Proximity แต่ละตัวบนแต่ละเครื่อง ซึ่งค่าที่ได้จาก Proximity จะมีหน่วยเป็น เซนติเมตร และ Proximity บนแต่ละเครื่องจะมีค่า X ต่างกันออกไป ตัวอย่างเช่น Proximity ของ Galaxy Nexus จะมีระยะ 5 cmเมื่อมีวัตถุอยู่ในระยะ 5 เซนติเมตร ค่าที่ได้ก็จะได้ออกมาเป็น 0 และถ้านอกเหนือจากนั้น ค่าที่ได้ก็จะได้ออกมาเป็น 5 แต่สำหรับบน Galaxy Gio จะมีระยะแค่ 1 เซนติเมตร เท่านั้น ดังนั้นการใช้งานจริงๆควรจเช็คว่าวัดได้ที่ระยะเท่าไร หรือเช็คจากค่าเป็น 0 ก็ได้ ถ้าค่าเป็น 0 คือมีวัตถุบังอยู่ และถ้าค่ามากกว่า 0 ก็หมายความว่าไม่มีวัตถุใดบังอยู่

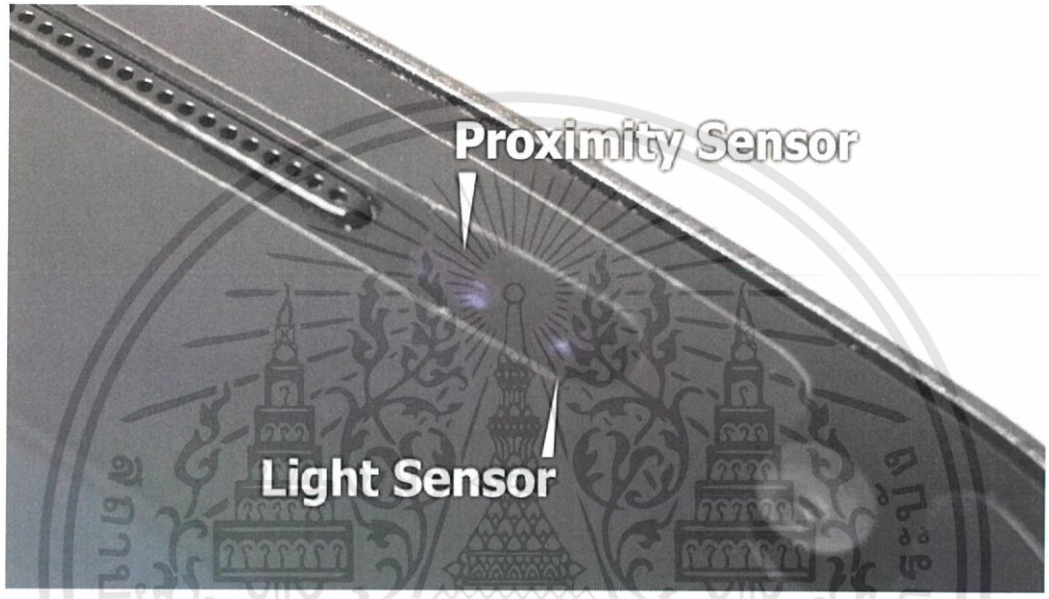


รูปที่ 3.68 Proximity Sensor

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.6.2 เซนเซอร์ตรวจจับแสงสว่าง (Light)

Light Sensor จะใช้การวัดค่า Lux ของแสงที่วัด ซึ่งค่า Lux คือหน่วยของความสว่างของแสง ถ้าอุปกรณ์แอนดรอยด์รุ่นใดมี Light Sensor จะสามารถปรับแสงหน้าจอแบบอัตโนมัติได้ โดยสังเกตได้จากช่องเดี่ยวที่อยู่ข้างหน้าฝั่งบนของจอ



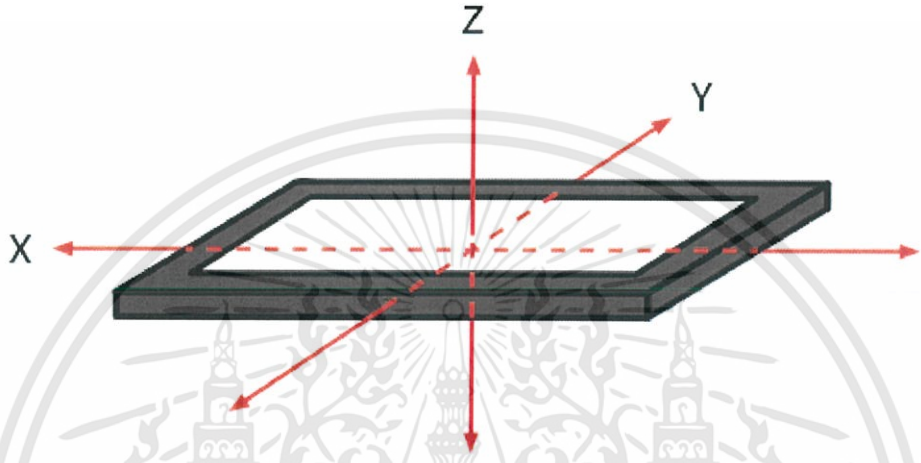
รูปที่ 3.69 Light Sensor

ค่า Lux ของ Light Sensor จะมีค่าอยู่ระหว่าง 0 จนถึงความเข้มแสง ณ เวลาที่ทำการวัด โดยถ้าความเข้มแสงน้อย (แสงสว่างน้อย) ค่าของ Lux จะน้อย และถ้าความเข้มแสงมาก (แสงสว่างมาก) ค่า Lux จะมากขึ้นตามไปด้วย

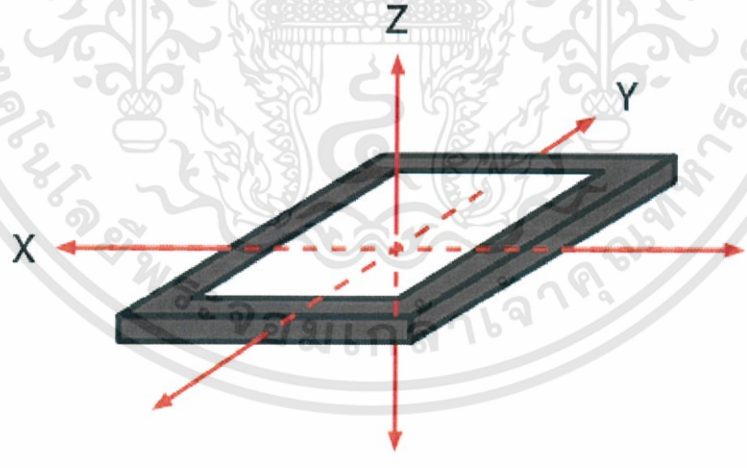
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.7 การเรียกใช้เซนเซอร์วัดการเอียง (Accelerometer) [9]

เซนเซอร์วัดการเอียงจะวัดความเร่งในการเอียงเครื่องทั้ง 3 ทิศ สำหรับแกน XYZ บนอุปกรณ์แอนดรอยด์ โดยจะมีตำแหน่งของแกนในการวัดดังนี้



รูปที่ 3.70 แกนสำหรับอุปกรณ์ที่มีขนาด 7 นิ้วขึ้นไป



รูปที่ 3.71 แกนสำหรับอุปกรณ์ที่มีขนาด 7 นิ้วลงมา

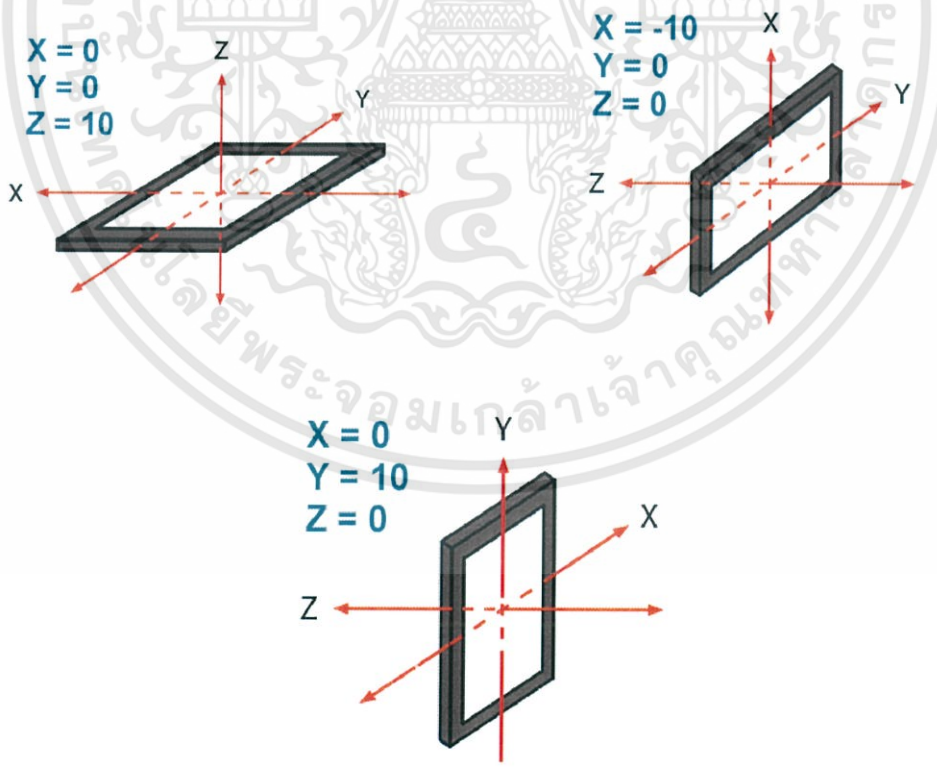
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปจะเห็นได้ว่าขนาดหน้าจอของอุปกรณ์ที่แตกต่างกันจะมีแนวของแกนวัดที่แตกต่างกัน เนื่องจากหน้าจอขนาดที่เล็กกว่า 7 นิ้วจะเน้นการใช้งานในแนวตั้งเป็นหลัก ส่วนอุปกรณ์ที่มีขนาด 7 นิ้วขึ้นไปจะเน้นการใช้งานในแนวนอน

เซนเซอร์วัดการเอียงจะวัดความเร่งในแต่ละแกน เวลาที่เครื่องอยู่นิ่งไม่มีการขยับ ค่าของแกน x และ y จะเป็น 0 แต่ค่าของแกน z จะไม่เป็น 0 เพราะมีแรงโน้มถ่วงของโลก ดังนั้นถ้าอุปกรณ์หยุดนิ่ง ค่าแกน x และ y จะเป็น 0 ส่วนค่าของแกน z จะมีค่าเป็นแรงโน้มถ่วงโลกคือประมาณ 10

ตารางที่ 3.1 รายละเอียดค่าของการเอียง

มุม / แกน	-90 องศา	0 องศา	90 องศา
x	-10	0	10
y	-10	0	10
z	0	10	0

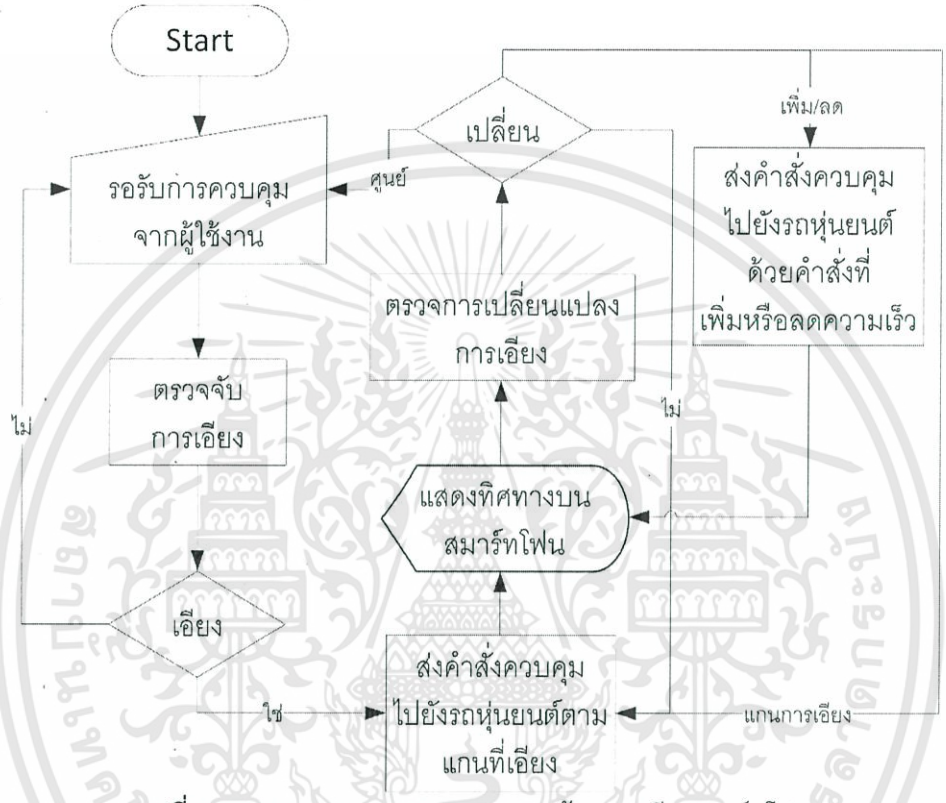


รูปที่ 3.72 แสดงค่าของแต่ละแกนเมื่อเอียงเครื่องไปในทิศทางต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.7.1 แผนภาพขั้นตอนการทำงานของการควบคุมด้วยการเอียงสมาร์ทโฟน

ในการพัฒนาแอปพลิเคชันเพื่อควบคุมรถหุ่นยนต์ด้วยรูปแบบการเอียงสมาร์ทโฟนนั้นได้มีการออกแบบขั้นตอนการทำงานในการตรวจสอบค่าของการเอียงและการเปลี่ยนแปลงต่างๆ ในรูปแบบนี้ ดังแผนภาพต่อไปนี้



รูปที่ 3.73 Flow chart ของการควบคุมด้วยการเอียงสมาร์ทโฟน

3.7.2 การเรียกใช้งานเซนเซอร์วัดการเอียง

สำหรับการใช้งานเซนเซอร์ใดๆ ในอุปกรณ์แอนดรอยด์ จะต้องมีคำสั่งเพื่อขอใช้งานเซนเซอร์ตัวนั้นๆด้วย ส่วนการใช้งานเซนเซอร์วัดการเอียงจะมีคำสั่งเพื่อขอใช้งานดังนี้

```
SensorManager sensorManager = (SensorManager)getSestymService(Context.SENSOR_SERVICE);
Sensor sensor = sensorManager.getDefaultSensor(Sensor.TYPE_ACCELEROMETER);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากคำสั่งในด้านบน จะให้ sensorManager ดึงการให้บริการ (Service) จำพวกเซนเซอร์มาเก็บไว้แล้วให้ sensor ซึ่งเป็นคลาสของ Sensor ดึงค่าของ Accelerometer จากตัวแปร sensorManager มีผลทำให้ตัวแปร sensor กลายเป็น Accelerometer Sensor

การใช้งานนั้นจะต้องสั่งการผ่านคำสั่ง registerListener ซึ่งเป็นคำสั่งของ SensorManager ที่กำหนดให้อ่านค่าจากเซนเซอร์ โดยปกติแล้วเซนเซอร์วัดการเอียงจะถูกเรียกใช้งานอยู่แล้ว จึงไม่ต้องทำการสั่งให้เซนเซอร์เริ่มทำงาน ทำเพียงแค่กำหนด Listener เพื่อรับค่าจากเซนเซอร์

```
sensorManager.registerListener(Listener,Sensor,SensorManager.SENSOR_DELAY_NORMAL);
```

การกำหนด Listener ให้กับเซนเซอร์ ที่เป็น registerListener จะต้องมีการหยุดใช้งาน Listener ซึ่งจะใช้คำสั่ง unregisterListener โดยจะใช้คำสั่งนี้ก็ต่อเมื่อไม่ต้องการจะใช้งานเซนเซอร์แล้ว

สำหรับ registerListener จะนิยมเรียกใช้ใน onResume() เพื่อให้แอปพลิเคชันสามารถใช้งาน Accelerometer Sensor ได้ทันทีที่ Activity นั้นกลับมาทำงาน และสำหรับ unregisterListener จะเรียกใช้งานใน onPause() เพื่อที่เมื่อทำการปิดหรือหยุด Activity ที่เรียกใช้งาน หรือเมื่อทำการย่อแอปพลิเคชันจะให้ Listener หยุดทำงาน

```
public void onResume()
{  super.onResume();
   sensorManager.registerListener(Listener,Sensor,SensorManager.SENSOR_DELAY_NORMAL);
}
public void onStop
{  super.onStop();
   sensorManager.unregisterListener(Listener);
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนของ Listener ที่ใช้กับเซนเซอร์จะใช้เป็น Listener แบบ SensorEventListener ซึ่งจะประกาศเป็นตัวแปรที่กำหนดค่าให้ทันที เพราะจะต้องมีการเรียกใช้ตัวแปรนี้จากคำสั่ง registerListener และ unregisterListener นั่นเอง

```
SensorEventListener accelListener = new SensorEventListener()
{  public void onSensorChanged(SensorEvent event)
   {  float x = event.values[0];
      float y = event.values[1];
      float z = event.values[2];
   }
};
```

ฟังก์ชัน onSensorChanged เป็นฟังก์ชันที่ทำงานเมื่อเซนเซอร์มีการอ่านค่าใหม่เมื่อมีการเปลี่ยนแปลง โดยตัวแปรที่ชื่อ event ที่ทำการส่งเข้ามาในฟังก์ชันนี้เป็นตัวแปรของคลาส SensorEvent ที่ส่งค่าเข้ามาจากเซนเซอร์บนสมาร์ทโฟน ดังนั้นเวลาอ่านค่าจากเซนเซอร์จึงต้องอ่านจากตัวแปรนี้ สำหรับ Accelerometer Sensor จะส่งค่ามาในตัวแปรนี้ทั้งสามแกน โดยจะเก็บไว้ในอาร์เรย์ช่อง X Y Z ตามลำดับ เวลาจะอ่านค่าจากเซนเซอร์ก็อ่านจากคำสั่งดังนี้

```
float value = event.values[int];
```

โดย int คือลำดับของอาร์เรย์ที่ต้องการอ่านค่า

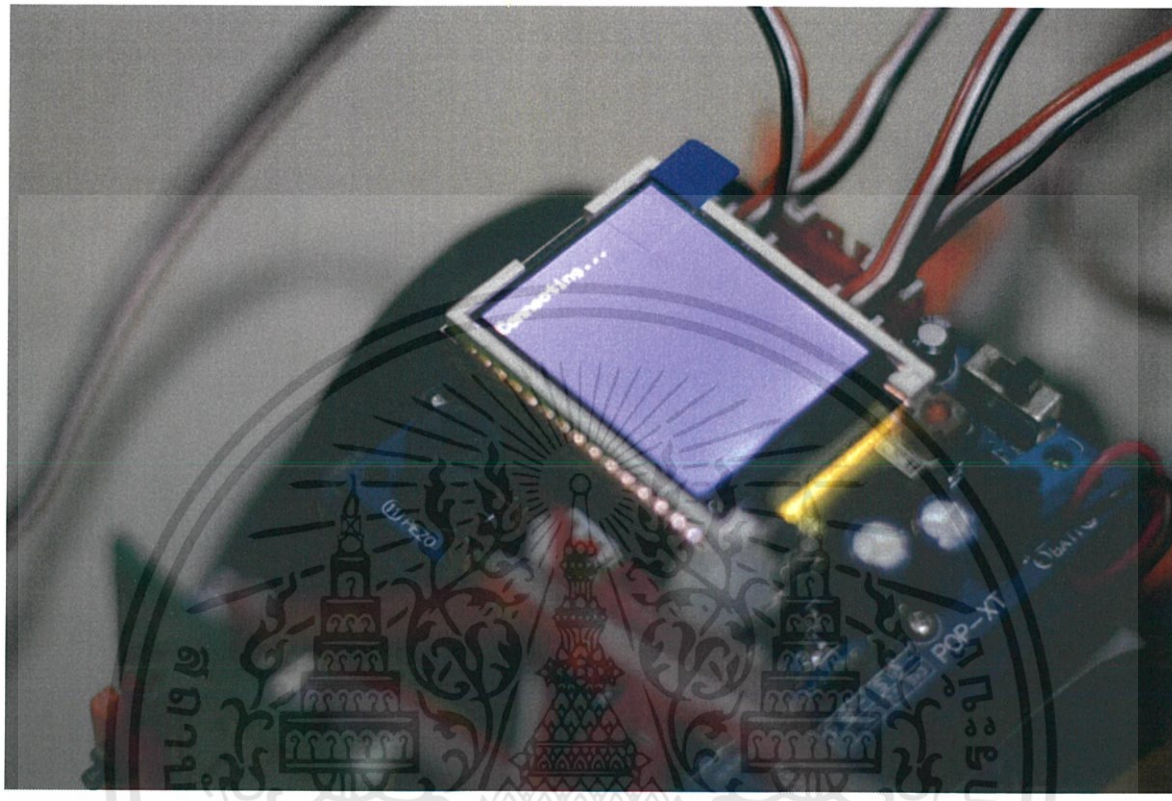
การออกแบบการเอียงของสมาร์ทโฟนเพื่อใช้ในการบังคับหุ่นยนต์นั้น เราได้ออกแบบยิ่งการเอียงยิ่งมากค่าความเร็วในการเคลื่อนที่ของหุ่นยนต์จะมากขึ้น และถ้าเอียงน้อยลงความเร็วในการเคลื่อนที่จะลดลงตามไปด้วย

ตารางที่ 3.2 แสดงค่าความเร็วที่เปลี่ยนแปลงตามการเอียงของสมาร์ทโฟน

ค่าการเอียง / แกน	น้อยกว่า -6	-5 ถึง -4	-3 ถึง -2	-1 ถึง 1	2 ถึง 3	4 ถึง 5	มากกว่า 6
x	ขวา 90	ขวา 50	ขวา 20	หยุด	ซ้าย 20	ซ้าย 50	ซ้าย 90
y	หน้า 90	หน้า 50	หน้า 20	หยุด	หลัง 20	หลัง 50	หลัง 90

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

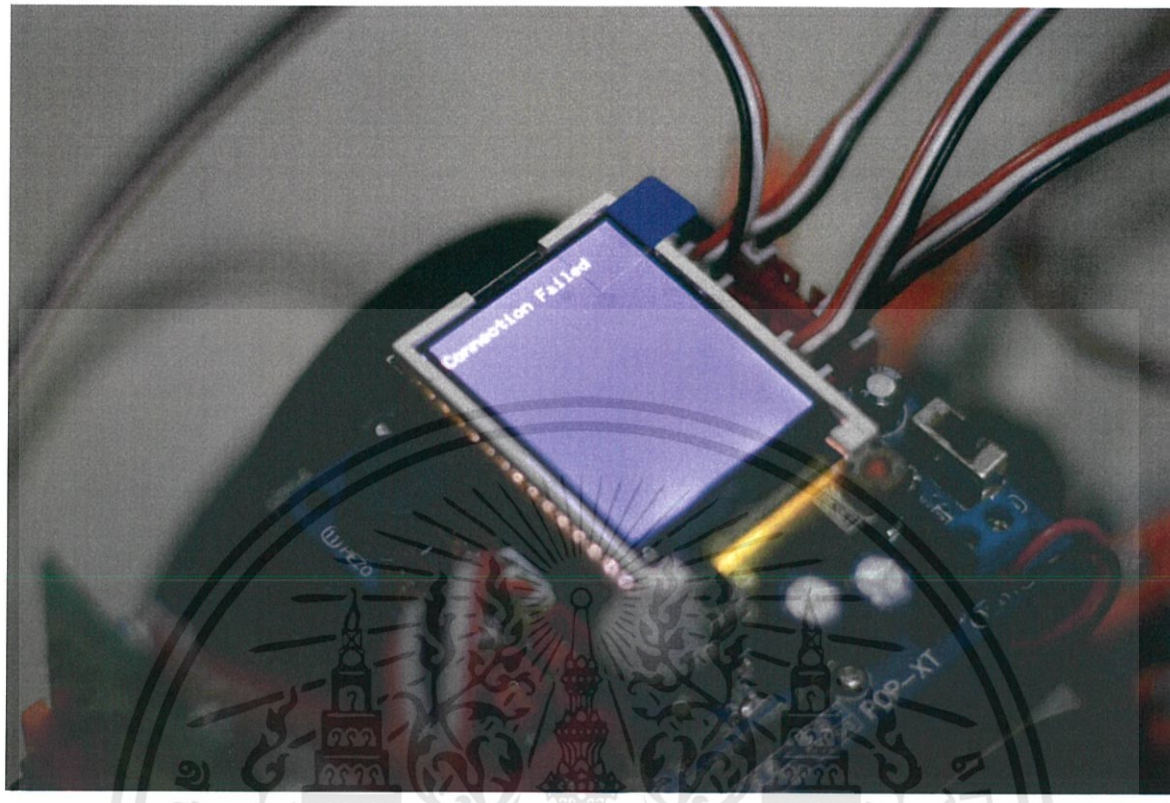
3.8 หน้าจอแสดงผลบนไมโครคอนโทรเลอร์



รูปที่ 3.74 ไมโครคอนโทรเลอร์กำลังเชื่อมต่อกับ Wi-Fi Hotspot

เมื่อเปิดสวิตซ์ของรถหุ่นยนต์ หน้าจอของไมโครคอมโทรเลอร์จะแสดงผลบนหน้าจอด้วยคำว่า Connecting... เพื่อแสดงสถานะว่ากำลังเชื่อมต่อกับ Wi-Fi Hotspot ที่เราได้เปิดไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

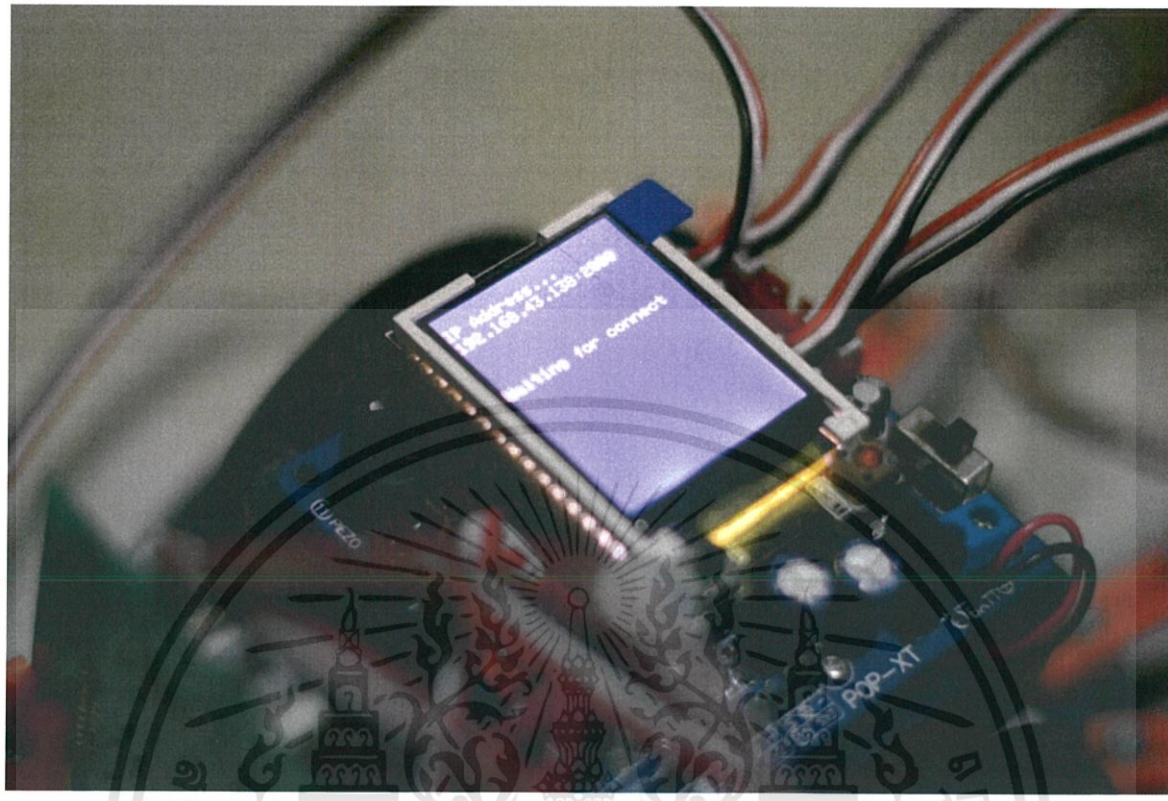


รูปที่ 3.75 ไมโครคอนโทรเลอร์เชื่อมต่อกับ Wi-Fi Hotspot ไม่สำเร็จ

หลังจากไมโครคอนโทรเลอร์ทำการเชื่อมต่อกับ Wi-Fi Hotspot ถ้าการเชื่อมต่อไม่สำเร็จจะขึ้นข้อความว่า Connection Failed ซึ่งถ้าการเชื่อมต่อไม่สำเร็จอาจมาจากสาเหตุดังนี้

- ผู้ใช้งานไม่ได้ทำการเปิด Wi-Fi Hotspot บนสมาร์ทโฟน
- การกำหนดค่าของ Wi-Fi Hotspot บนสมาร์ทโฟนและ RN-XV WiFly Module บนรถหุ่นยนต์ไม่ตรงกัน เช่น ชื่อ SSID เป็นต้น

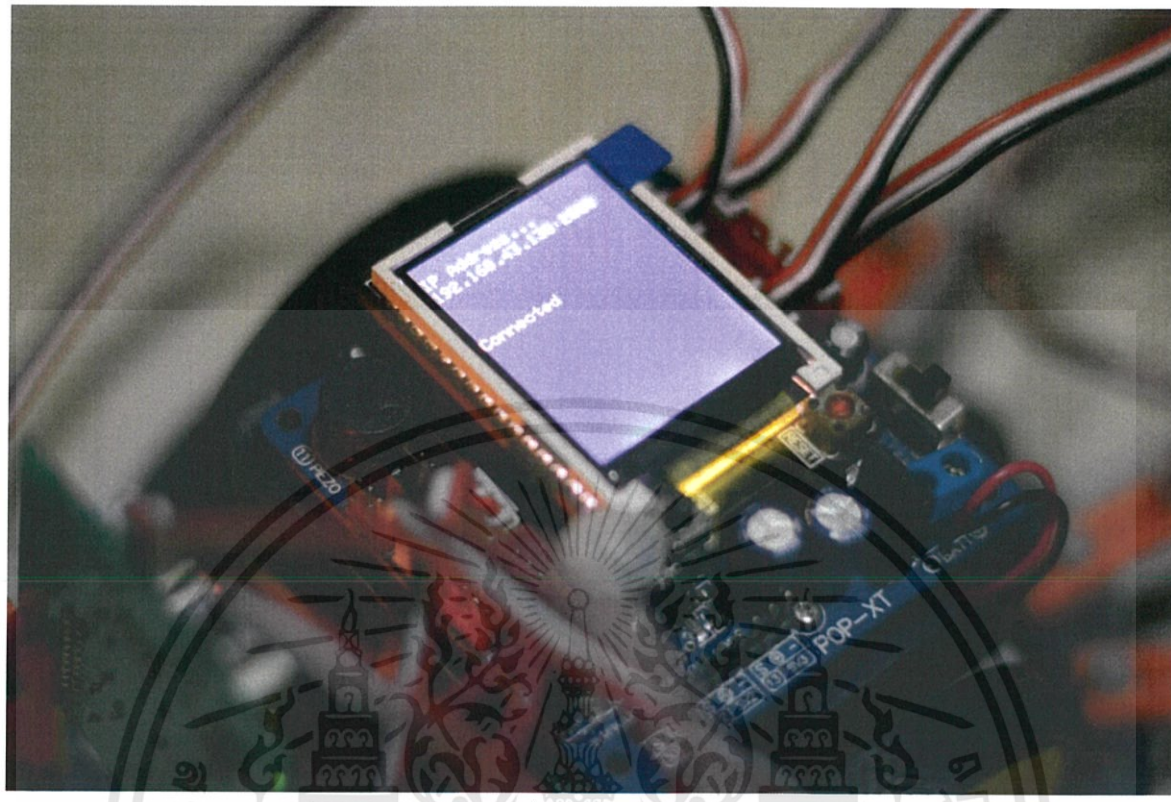
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.76 ไมโครคอนโทรเลอร์เชื่อมต่อกับ Wi-Fi Hotspot สำเร็จ

หลังจากไมโครคอนโทรเลอร์ทำการเชื่อมต่อกับ Wi-Fi Hotspot ถ้าการเชื่อมต่อสำเร็จหน้าจอของไมโครคอนโทรลเลอร์จะแสดง IP Address และ port พร้อมกับข้อความ Waiting for connect เพื่อรอการเชื่อมต่อจากสมาร์ทโฟน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



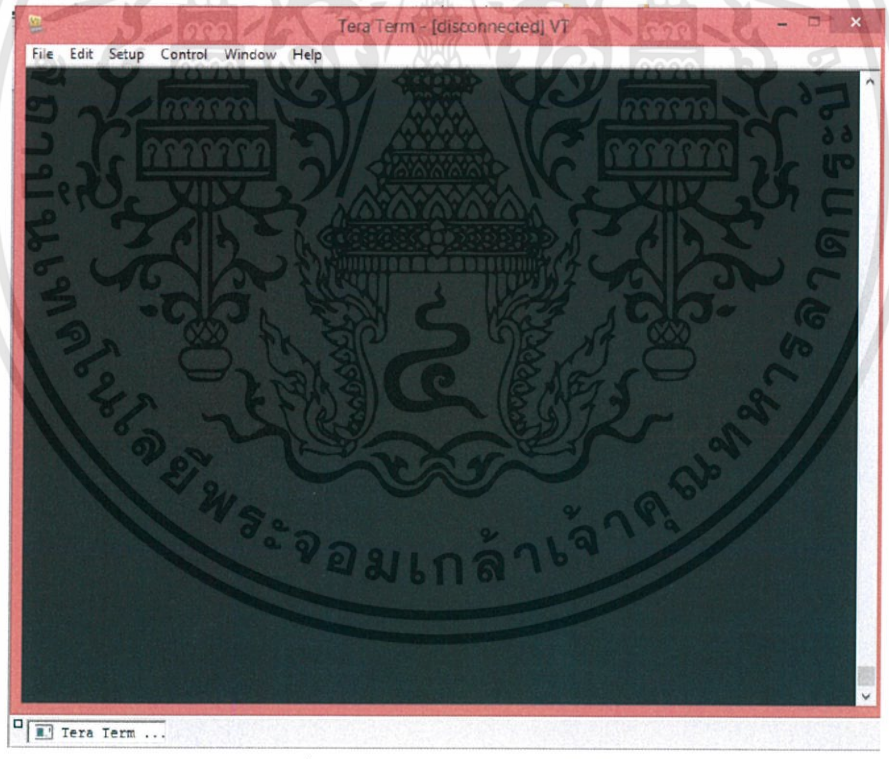
รูปที่ 3.77 สมาร์ทโฟนเชื่อมต่อกับไมโครคอนโทรเลอร์สำเร็จ

เมื่อทำการเชื่อมต่อแอปพลิเคชันบนสมาร์ทโฟนกับไมโครคอนโทรเลอร์และการเชื่อมต่อสำเร็จ หน้าจอของไมโครคอนโทรลเลอร์จะแสดงข้อความ Connected เพื่อบอกว่าการเชื่อมต่อสำเร็จและพร้อมใช้งานกับสมาร์ทโฟน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.9 การตั้งค่าอุปกรณ์ RN-XV WiFly Module

ในการเชื่อมต่อระหว่างรถหุ่นยนต์กับสมาร์ทโฟนจำเป็นต้องมี Module ในการทำให้รถหุ่นยนต์และสมาร์ทโฟนสามารถสื่อสารกันผ่านเครือข่ายไร้สายได้ ดังนั้นจึงต้องมีการใช้งานอุปกรณ์ RN-XV WiFly Module ซึ่งเป็นอุปกรณ์ที่สามารถใช้รับส่งข้อมูลผ่านเครือข่ายไร้สาย Wi-Fi ได้ ซึ่งการเชื่อมต่อของสมาร์ทโฟนกับรถหุ่นยนต์จะใช้ลักษณะการเชื่อมต่อเป็นแบบ Adhoc Mode โดยจะทำการตั้งค่า SSID , password และ รูปแบบการเข้ารหัส password ของอุปกรณ์ RN-XV WiFly Module ให้ตรงกันกับชื่อ SSID , password และรูปแบบการเข้ารหัส password ของ Wi-Fi Hotspot ที่ทำการตั้งค่าไว้บนสมาร์ทโฟน ส่วนการตั้งค่า IP Address และการตั้งค่าอื่นๆ จะกำหนดให้รับค่าตามที่ Wi-Fi Hotspot บนสมาร์ทโฟนกำหนดให้โดยอัตโนมัติในแต่ละครั้ง โดยการตั้งค่าของ RN-XV WiFly Module นั้นจะใช้งานโปรแกรม Tera Term



รูปที่ 3.78 โปรแกรม Tera Term

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การทดสอบและผลการดำเนินงาน

ในโครงงานปัญหาพิเศษนี้ได้แบ่งการทดสอบระบบออกเป็น 2 ส่วน คือ ส่วนของสมาร์ทโฟนระบบปฏิบัติการแอนดรอยด์ที่ใช้เป็นอุปกรณ์ควบคุม และส่วนของรถหุ่นยนต์ซึ่งประกอบด้วยบอร์ดไมโครคอนโทรลเลอร์ เซนเซอร์ มอเตอร์ และอุปกรณ์รับส่งสัญญาณ Wi-Fi เพื่อสังเกตลักษณะการทำงานและผลลัพธ์ที่เกิดขึ้น

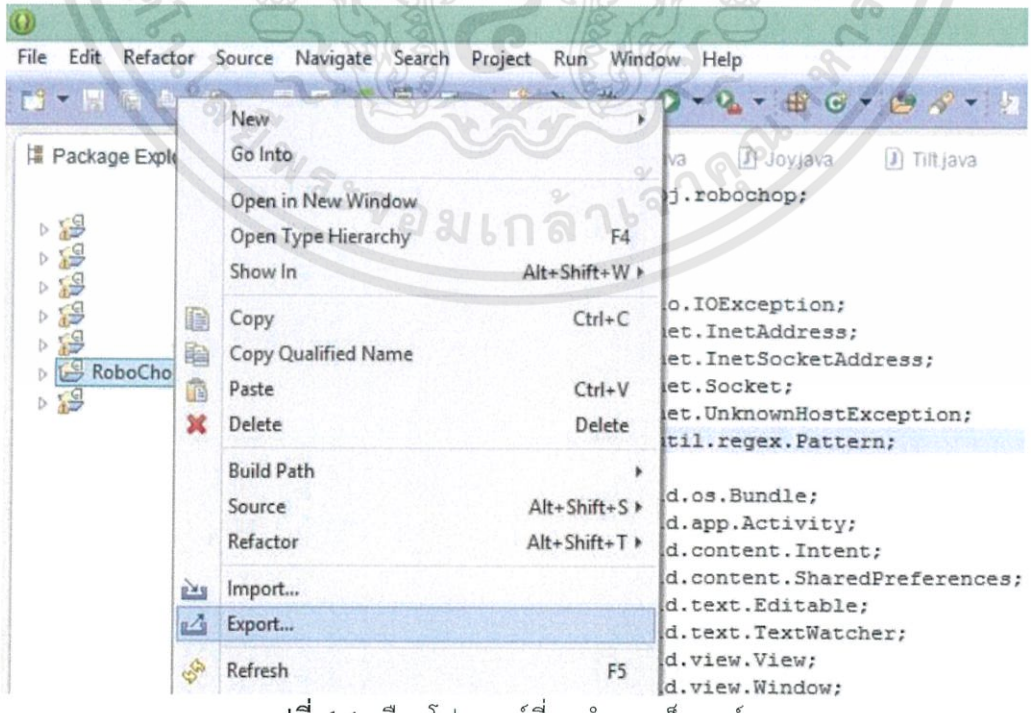
4.1 การเอ็กพอร์ต (Export) แอปพลิเคชันจากโปรแกรม Eclipse เป็นไฟล์ .apk

เมื่อทำการพัฒนาแอปพลิเคชันบนระบบปฏิบัติการแอนดรอยด์เรียบร้อยแล้ว และต้องการที่จะนำแอปพลิเคชันที่พัฒนาออกจากโปรแกรม Eclipse มาติดตั้งบนสมาร์ทโฟนหรือนำขึ้นสู่เพลสโตร์ (Play Store) เพื่อทำการจำหน่ายจะต้องทำการเอ็กพอร์ตโปรเจกต์ออกมาเป็นไฟล์ .apk

สิ่งสำคัญที่สุดในการเอ็กพอร์ตไฟล์ .apk ออกมา คือ ไฟล์คีย์สโตร์ (keystore) ซึ่งเป็นไฟล์ที่เปรียบเสมือนบัตรประจำตัวของผู้พัฒนาแอปพลิเคชันตัวนั้นๆ ซึ่งในแอปพลิเคชันแต่ละตัวจะมีการระบุไฟล์คีย์สโตร์ของผู้พัฒนาแอปพลิเคชันลงไปเพื่อใช้ระบุว่านักพัฒนาคนใดเป็นผู้พัฒนาแอปพลิเคชัน

4.1.1 ขั้นตอนการสร้างคีย์สโตร์และเอ็กพอร์ตไฟล์ .apk

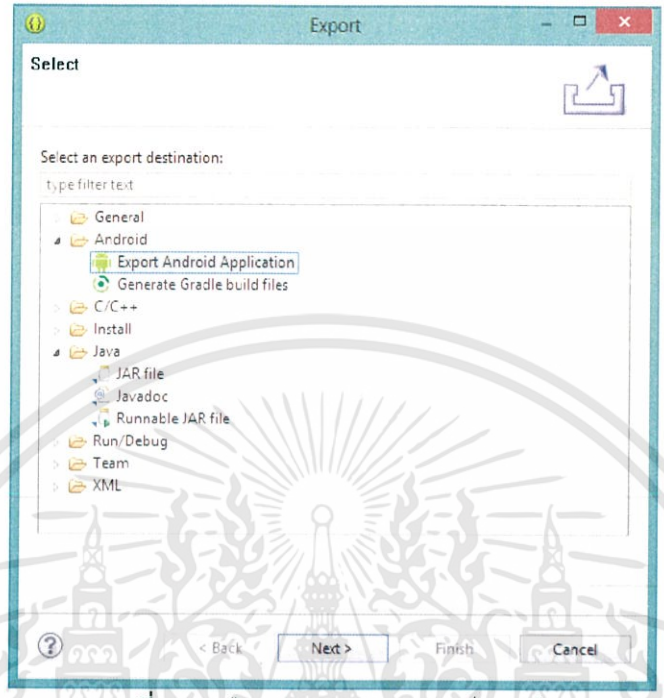
1. ทำการคลิกขวาที่โปรเจกต์ที่ต้องการเอ็กพอร์ต จากนั้นเลือก Export



รูปที่ 4.1 เลือกโปรเจกต์ที่จะทำการเอ็กพอร์ต

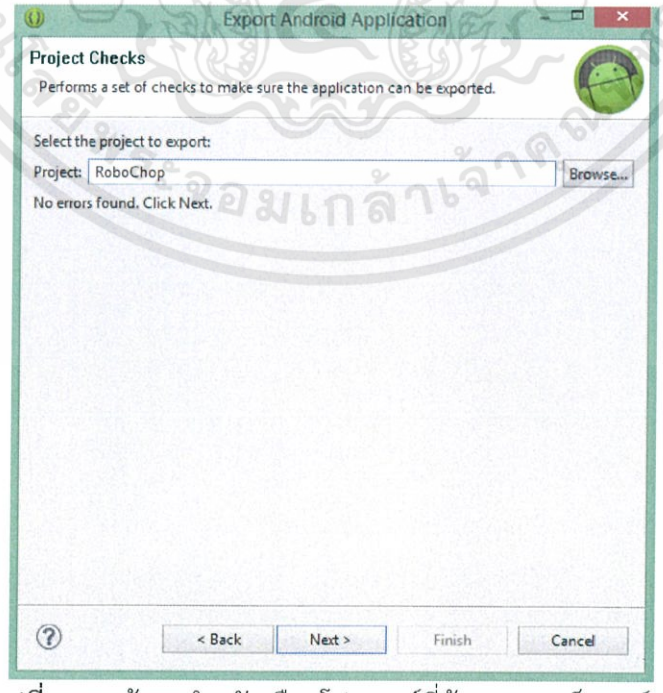
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. จะมีหน้าต่าง Export แสดงขึ้นมา ให้เลือกที่ Android > Export Android Application จากนั้นกดปุ่ม Next



รูปที่ 4.2 เลือกประเภทของการเอ็กพอร์ต

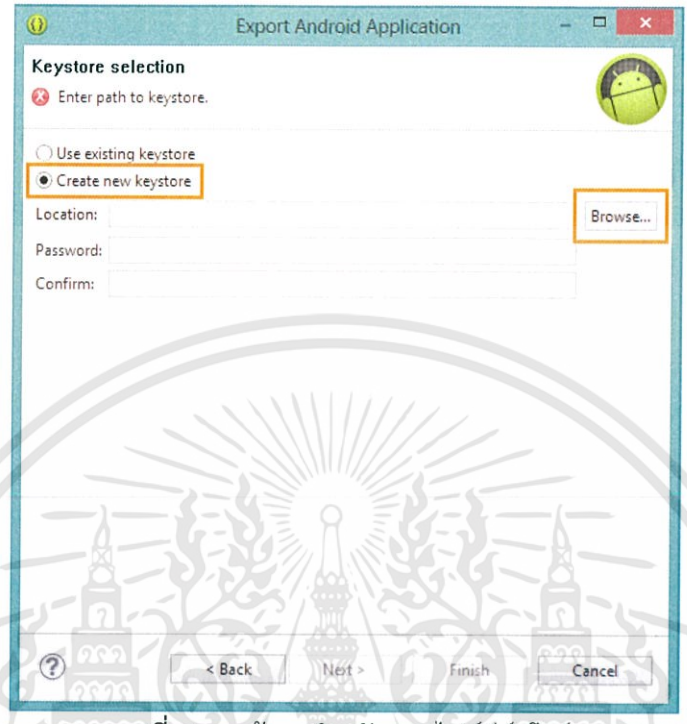
3. จากนั้นจะมีหน้าต่างสำหรับเลือกโปรเจกต์ที่ต้องการเอ็กพอร์ต แต่เนื่องจากเราได้ทำการคลิกขวาแล้วเลือก Export ที่โปรเจกต์ที่เราต้องการเอ็กพอร์ตแล้วจึงไม่ต้องทำการเลือกอีกโดยสามารถกด Next ได้เลย



รูปที่ 4.3 หน้าจอสำหรับเลือกโปรเจกต์ที่ต้องการจะเอ็กพอร์ต

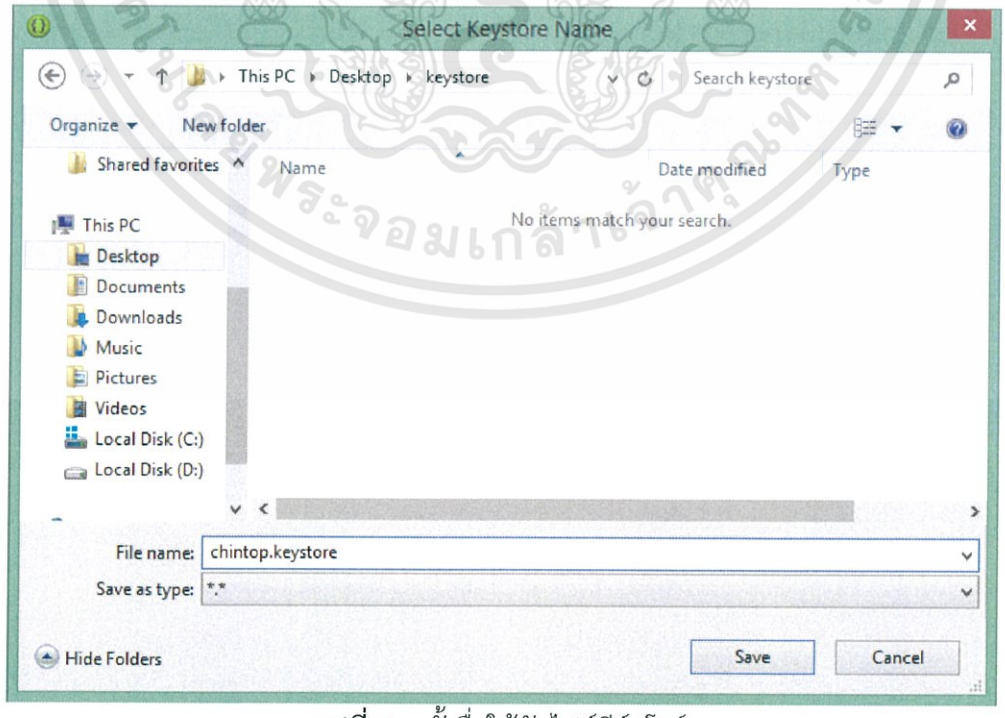
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. หน้าต่างถัดมาจะเป็นหน้าสำหรับกำหนดคีย์สโตร์ที่จะใช้ทำการเอ็กพอร์ต แต่เนื่องจากว่ายังไม่มีไฟล์คีย์สโตร์จึงต้องทำการสร้างขึ้นมาใหม่โดยเลือกที่ Create new keystore แล้วกดปุ่ม Browse..



รูปที่ 4.4 หน้าจอสำหรับระบุไฟล์คีย์สโตร์

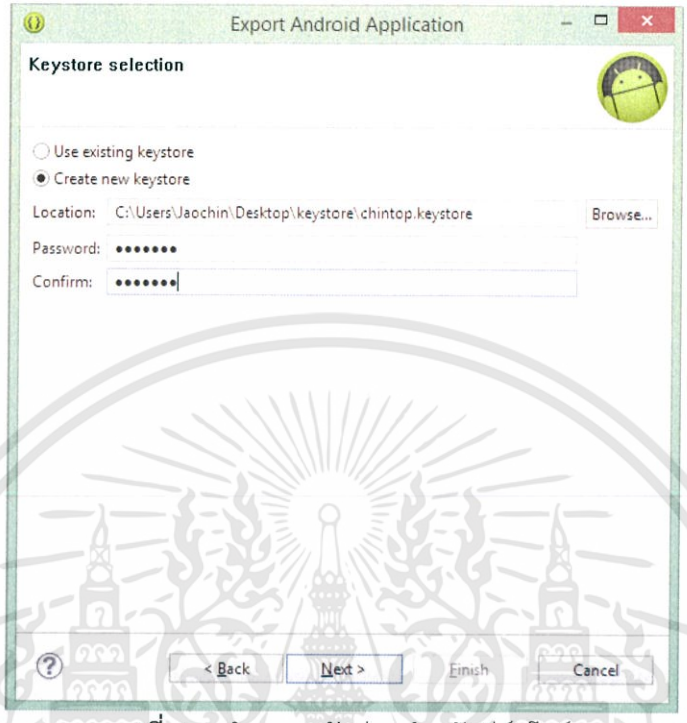
5. ทำการเลือกพื้นที่ที่ต้องการสร้างและจัดเก็บไฟล์คีย์สโตร์ จากนั้นทำการตั้งชื่อไฟล์คีย์สโตร์ ในที่นี้ตั้งชื่อไฟล์ว่า chintop.keystore แล้วกด Save



รูปที่ 4.5 ตั้งชื่อให้กับไฟล์คีย์สโตร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6. เมื่อกำหนดพื้นที่จัดเก็บและชื่อไฟล์คีย์สโตร์เรียบร้อยแล้ว จะต้องทำการกำหนดรหัสผ่านซึ่งรหัสผ่านนี้จำเป็นต้องใช้ทุกครั้งที่ทำการเอ็กพอร์ตโปรเจกต์ด้วยคีย์สโตร์นี้



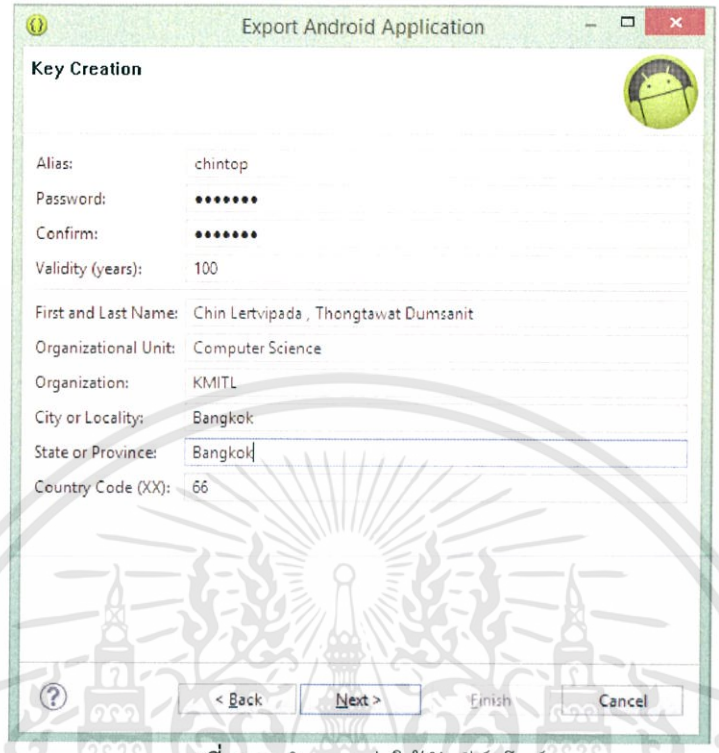
รูปที่ 4.6 กำหนดรหัสผ่านสำหรับคีย์สโตร์

7. หน้าต่างถัดมาจะเป็นการกำหนดรายละเอียดของคีย์สโตร์ที่เราทำการสร้างขึ้นโดยมีรายละเอียดต่างๆ ที่ต้องกำหนดดังนี้

- Alias : เป็นชื่อที่เปรียบเสมือน User Name สามารถกำหนดได้ตามต้องการ โดยให้ใช้ตัวอักษรเป็นตัวพิมพ์เล็กเนื่องจากว่าถึงจะใช้งานตัวพิมพ์ใหญ่ตัวโปรแกรมก็จะแปลงเป็นตัวพิมพ์เล็ก
- Password : เป็นรหัสผ่านของ Alias สามารถกำหนดได้ตามต้องการ
- Validity (years) : ระยะเวลาของคีย์สโตร์ที่จะมีผลกับแอปพลิเคชันที่จะทำการเอ็กพอร์ตเป็นระยะเวลาในหน่วยปีตามที่กำหนด โดยสามารถกำหนดจำนวนปีได้ตั้งแต่ 30–1000 ปี
- First and Last Name : ใช้กำหนดชื่อและนามสกุลของเจ้าของคีย์สโตร์
- Organizational Unit และ Organization : ใช้ระบุข้อมูลของบริษัทหรือองค์กร (ไม่จำเป็นต้องกำหนด)
- City or Locality : ชื่อเมืองหรือชื่อท้องถิ่น (ไม่จำเป็นต้องกำหนด)
- State or Province : ชื่อรัฐหรือชื่อจังหวัด (ไม่จำเป็นต้องกำหนด)
- Country Code (XX) : ใช้สำหรับระบุรหัสประเทศ สำหรับประเทศไทยใช้รหัส 66 (ไม่จำเป็นต้องกำหนด)

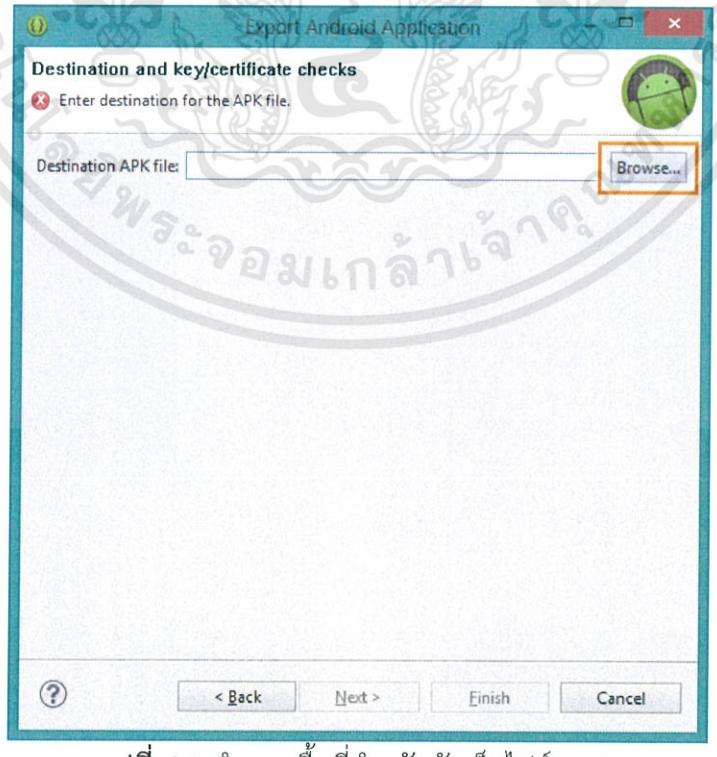
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อกำหนดค่าต่างๆ เรียบร้อยแล้วให้กดปุ่ม Next



รูปที่ 4.7 กำหนดค่าให้กับคีย์สโตร์

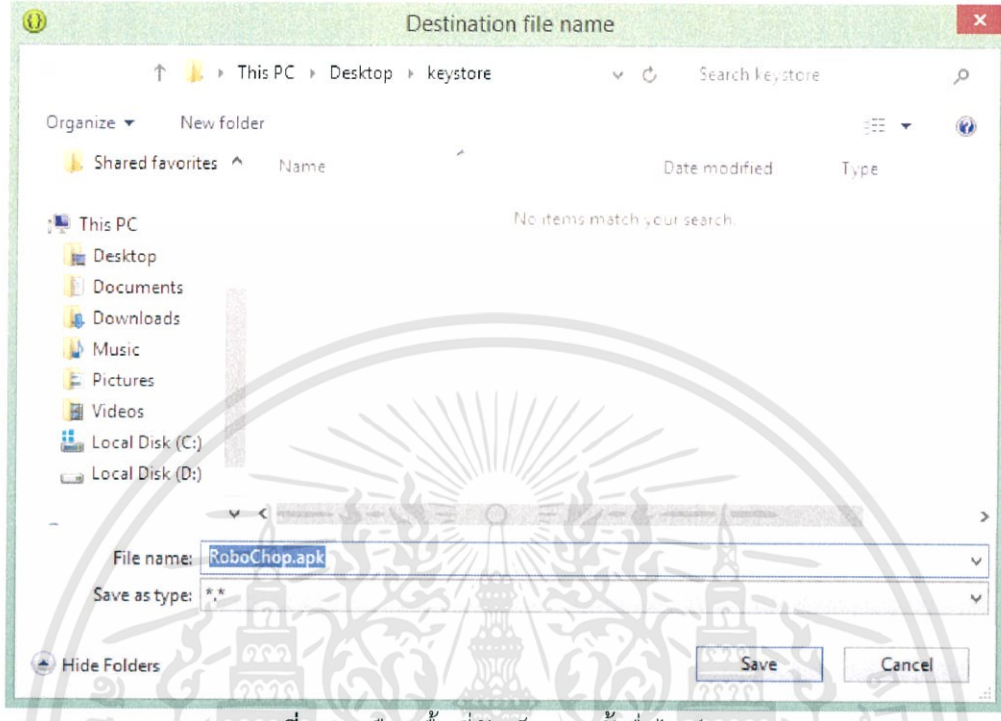
8. หน้าต่างถัดมาเป็นการกำหนดพื้นที่ที่ต้องการจัดเก็บไฟล์ .apk ของโปรเจกต์ที่กำลังทำการเอ็กพอร์ต โดยกดปุ่ม Browse...



รูปที่ 4.8 กำหนดพื้นที่สำหรับจัดเก็บไฟล์ .apk

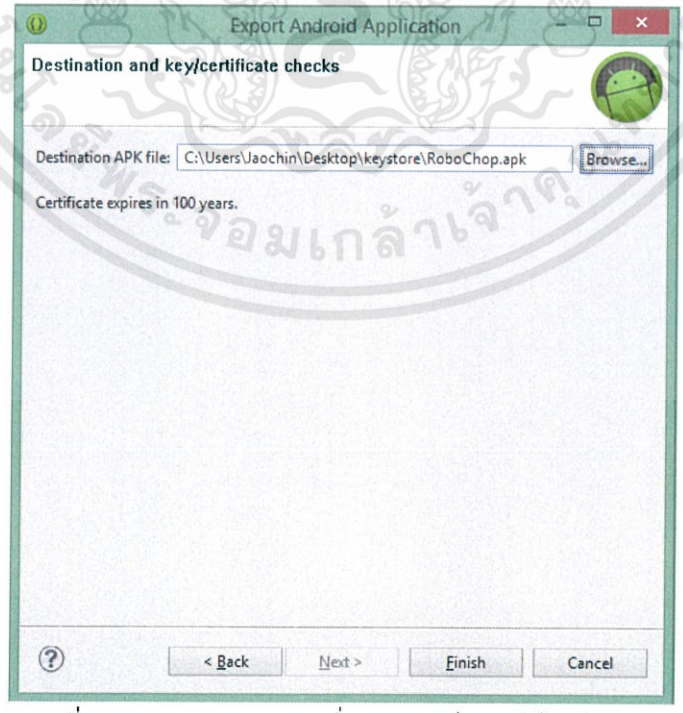
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

9. หลังจากกดปุ่ม Browse... จะสามารถเลือกพื้นที่จัดเก็บไฟล์ .apk ได้ตามต้องการ จากนั้นทำการตั้งชื่อไฟล์ .apk โดยให้ทำการใส่นามสกุล .apk ต่อท้ายชื่อไฟล์แล้วกดปุ่ม Save



รูปที่ 4.9 เลือกพื้นที่จัดเก็บและตั้งชื่อไฟล์ .apk

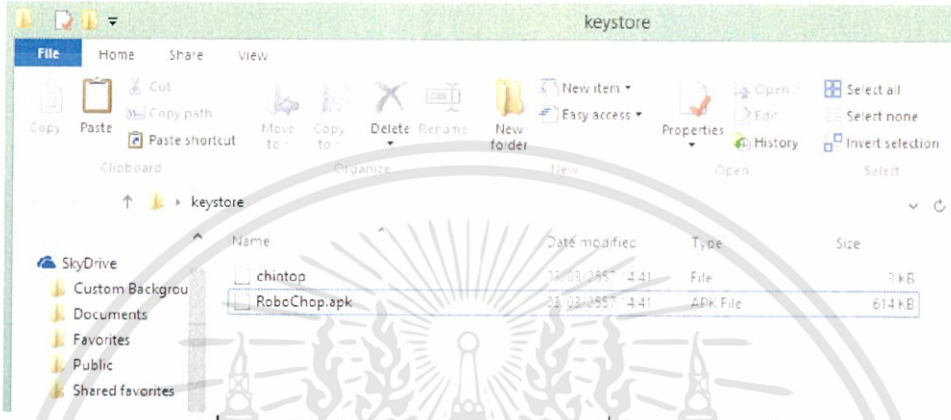
10. เมื่อกำหนดพื้นที่จัดเก็บของไฟล์ .apk เรียบร้อยแล้วจะเห็นข้อความแจ้งจำนวนปีที่ใบรับรองจะหมดอายุ (ตามที่กำหนดใน Validity) จากนั้นกดปุ่ม Finish เพื่อทำการเอ็กพอร์ต



รูปที่ 4.10 กดปุ่ม Finish เพื่อทำการเอ็กพอร์ตไฟล์ .apk

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

11. เมื่อเอ็กพอร์ตเสร็จเรียบร้อยแล้วให้เข้าไปยังโฟลเดอร์ที่กำหนดให้จัดเก็บไฟล์ .apk ก็จะเห็นไฟล์ .apk โดยสามารถเอาไฟล์ .apk นี้ขึ้นสู่เพลสโตร์หรือนำไปติดตั้งบนสมาร์ทโฟนระบบปฏิบัติการแอนดรอยด์ได้ทันที ส่วนไฟล์คีย์สโตร์ให้เก็บไว้เพื่อใช้ในการเอ็กพอร์ตครั้งต่อๆ ไป เพราะเมื่อนำไฟล์ .apk ขึ้นสู่เพลสโตร์แล้ว เวลาที่มีการอัพเดทจำเป็นต้องเอ็กพอร์ตไฟล์ .apk ด้วยไฟล์คีย์สโตร์ตัวเดิม

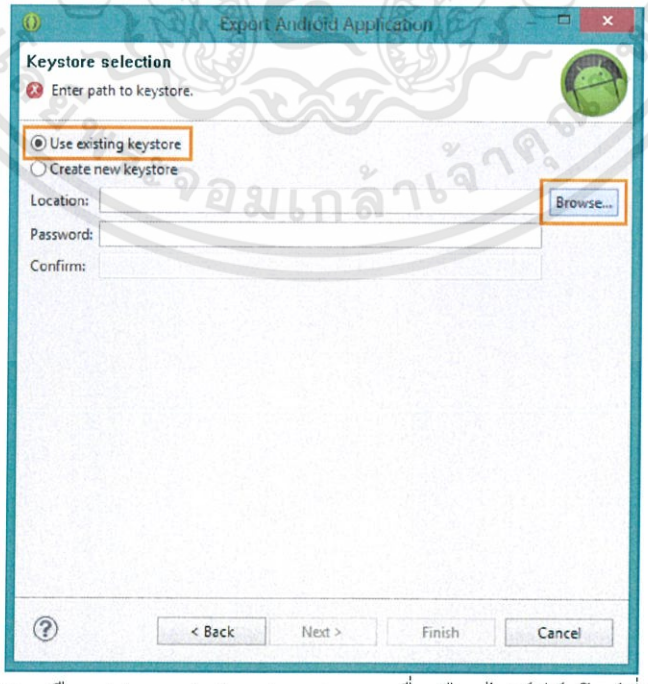


รูปที่ 4.11 ไฟล์คีย์สโตร์และไฟล์ .apk ที่ทำการเอ็กพอร์ต

4.1.2 ขั้นตอนการเอ็กพอร์ตไฟล์ .apk กรณีที่มีคีย์สโตร์แล้ว

1. ทำตามขั้นตอนข้อ 1-3 ในหัวข้อ **4.1.1 ขั้นตอนการสร้างคีย์สโตร์และเอ็กพอร์ตไฟล์ .apk**

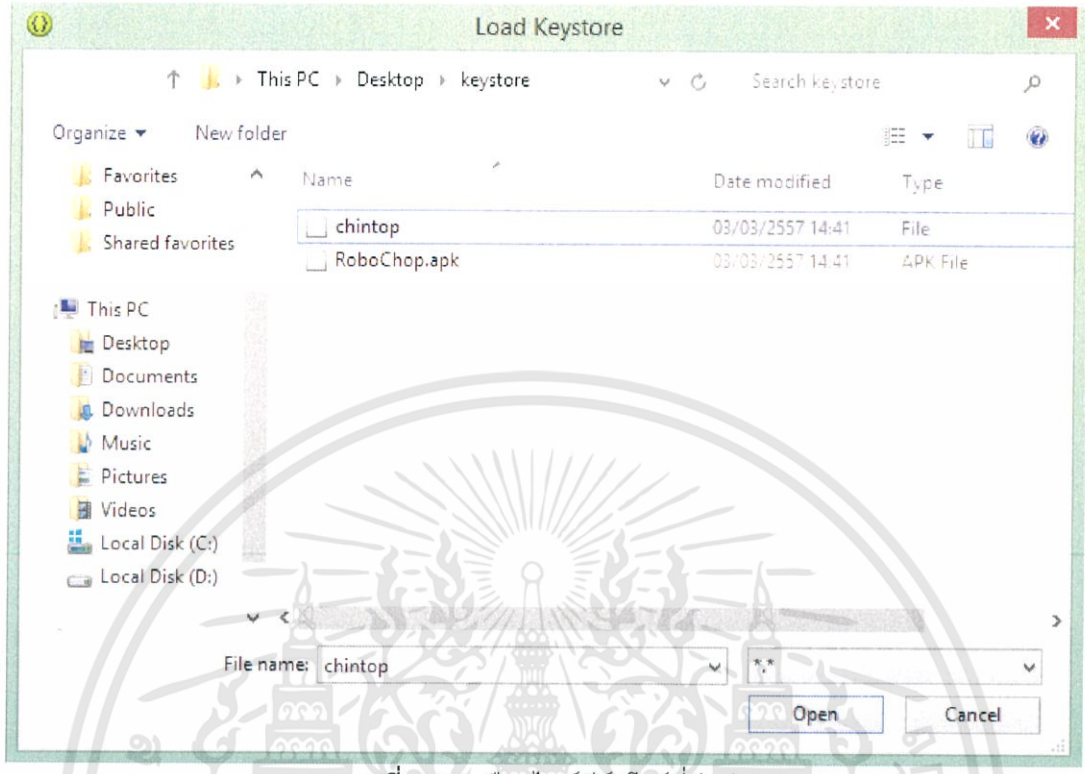
2. ในหน้าเลือกไฟล์คีย์สโตร์ที่จะนำมาใช้ในการเอ็กพอร์ตให้เลือก Use existing keystore แล้วกดที่ปุ่ม Browse...



รูปที่ 4.12 เลือก Use existing keystore เพื่อเลือกไฟล์คีย์สโตร์ที่มีอยู่แล้ว

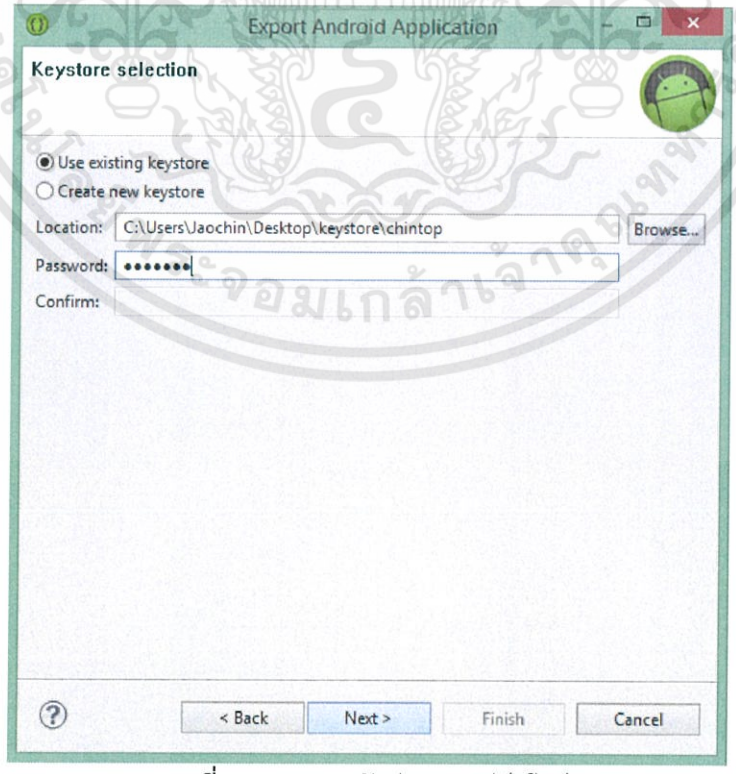
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. เลือกไฟล์คีย์สโตร์ที่ต้องการใช้เพื่อเอ็กพอร์ต จากนั้นกดปุ่ม Open



รูปที่ 4.13 เลือกไฟล์คีย์สโตร์ที่มีอยู่

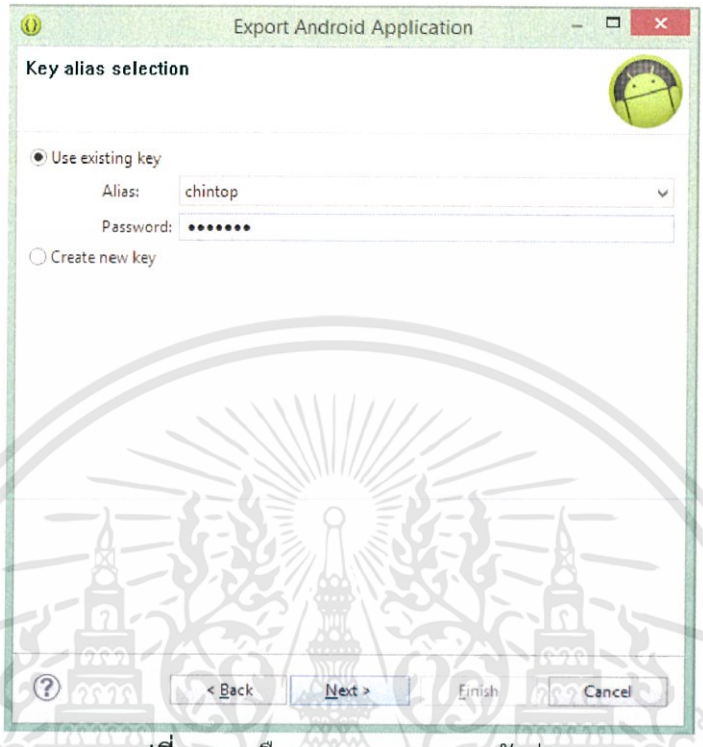
4. เมื่อทำการเลือกคีย์สโตร์เสร็จแล้ว ให้ใส่รหัสผ่านของคีย์สโตร์ แล้วกดปุ่ม Next



รูปที่ 4.14 ระบุรหัสผ่านของคีย์สโตร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. หน้าต่างถัดมาจะเป็นหน้าสำหรับเลือก Alias ของคีย์สโตร์ โดยสามารถใช้ Alias ที่มีอยู่แล้ว หรือจะสร้างขึ้นใหม่ก็ได้ หลังจากทำการเลือกแล้วให้ระบุรหัสผ่านของ Alias นั้นๆ แล้วกดปุ่ม Next



รูปที่ 4.15 เลือก Alias และระบุรหัสผ่าน

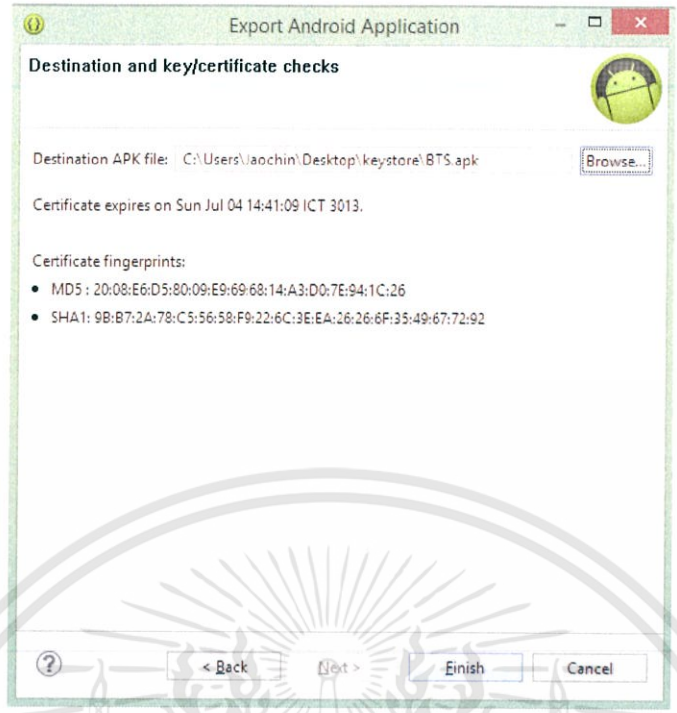
6. จากนั้นทำตามขั้นตอนข้อ 8-9 ในหัวข้อ **4.1.1 ขั้นตอนการสร้างคีย์สโตร์และเอ็กพอร์ตไฟล์ .apk**

7. เมื่อกำหนดชื่อไฟล์ .apk และพื้นที่จัดเก็บเรียบร้อยแล้วจะมีข้อความแสดงให้เห็นโดยข้อมูลมีดังนี้

- วันที่ใบรับรองหมดอายุ (อายุขึ้นอยู่กับการกำหนดใน Validity ตอนสร้างคีย์สโตร์)
- รหัส Certificate fingerprints ของคีย์สโตร์แบบ MD5 และ SHA1

ให้ทำการกดปุ่ม Finish เพื่อเริ่มทำการเอ็กพอร์ต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.16 แสดงรายละเอียดวันหมดอายุของคีย์สโตร์และ Certificate fingerprints

4.2 การทดสอบบนสมาร์ทโฟนระบบปฏิบัติการแอนดรอยด์

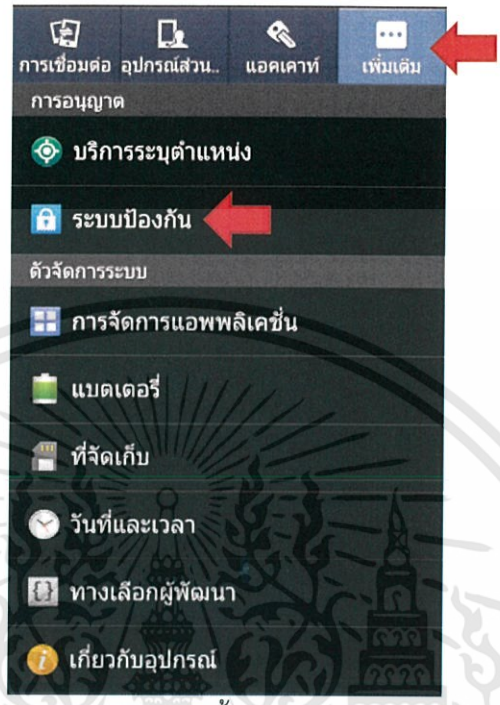
4.2.1 การติดตั้งแอปพลิเคชัน

4.2.1.1 การเตรียมสมาร์ทโฟนสำหรับติดตั้งแอปพลิเคชัน

เนื่องจากว่าในโครงงานปัญหาพิเศษนี้ไม่ได้นำแอปพลิเคชันที่สร้างขึ้นเข้าสู่เพลสโตร์ ดังนั้นจึงต้องมีการกำหนดค่าสมาร์ทโฟนเพื่อให้สามารถทำการติดตั้งแอปพลิเคชันจากไฟล์ .apk บนพื้นที่จัดเก็บข้อมูลของสมาร์ทโฟนได้ ซึ่งการกำหนดค่านี้อาจจะแตกต่างไปในสมาร์ทโฟนแต่ละรุ่น โดยในโครงงานปัญหาพิเศษนี้ได้ใช้สมาร์ทโฟนรุ่น Samsung Galaxy Note2 (Android 4.3) ซึ่งมีขั้นตอนการกำหนดค่าดังนี้

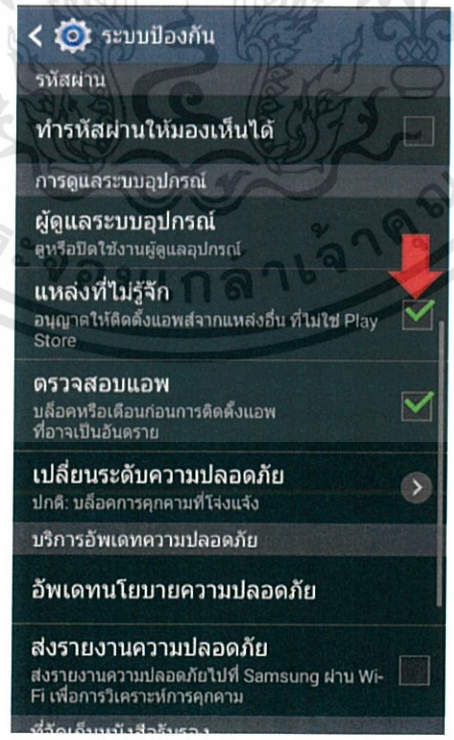
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. เข้าไปที่เมนูตั้งค่าของสมาร์ทโฟน “เลือกเพิ่มเติม” จากนั้น เลือก “ระบบป้องกัน”



รูปที่ 4.17 การเข้าสู่เมนูตั้งค่าระบบป้องกัน

2. เลือก “แหล่งที่ไม่รู้จัก”

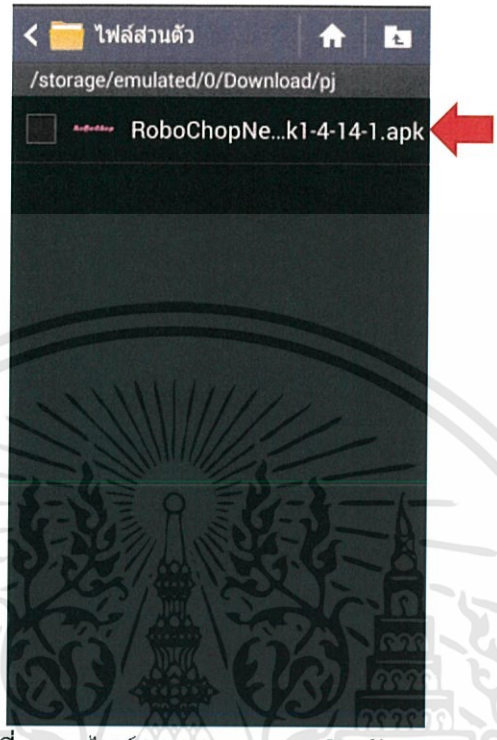


รูปที่ 4.18 การตั้งค่าเพื่อให้สมาร์ทโฟนสามารถติดตั้งแอพลิเคชันจากแหล่งที่ไม่รู้จักได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

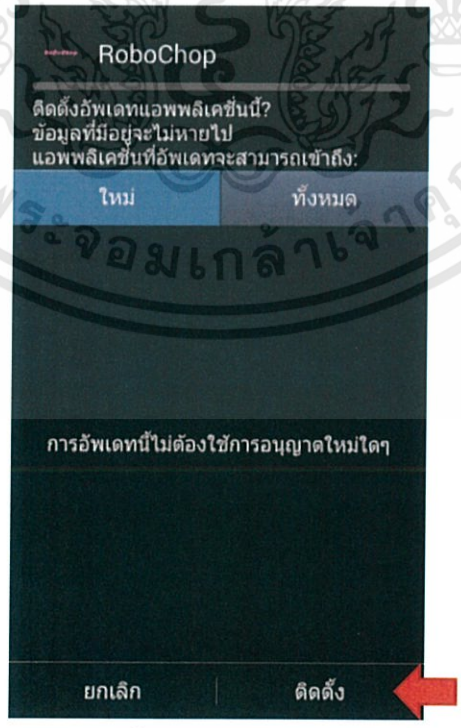
4.2.1.2 การติดตั้งแอปพลิเคชันบนสมาร์ทโฟน

1. เลือกไฟล์ .apk ของแอปพลิเคชัน



รูปที่ 4.19 ไฟล์ .apk ของแอพพลิเคชัน

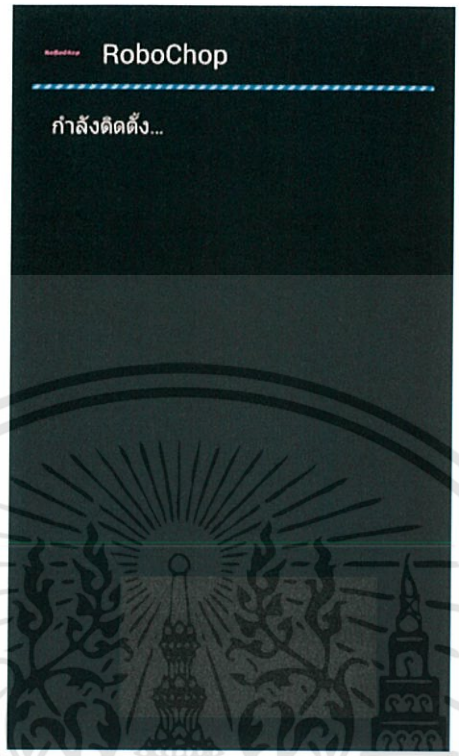
2. เลือกติดตั้ง



รูปที่ 4.20 แสดงการขออนุญาตเข้าถึงระบบต่างๆ ในสมาร์ทโฟนก่อนทำการติดตั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. จากนั้นระบบจะทำการติดตั้งแอปพลิเคชัน



รูปที่ 4.21 รอการติดตั้งแอพลิเคชัน

4.2.2 การทดสอบใช้งานแอปพลิเคชัน

1. การจ่าย IP Address แก่รถหุ่นยนต์ ทำการเปิด Wi-Fi Hotspot ที่สมาร์ทโฟนเพื่อให้รถหุ่นยนต์ทำการขอ IP Address สำหรับการเชื่อมต่อ จากนั้นทำการตั้งค่า Hotspot ดังนี้
   - Network SSID : Jae
   - Security : WPA2 PSK
   - Password : 123456789

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

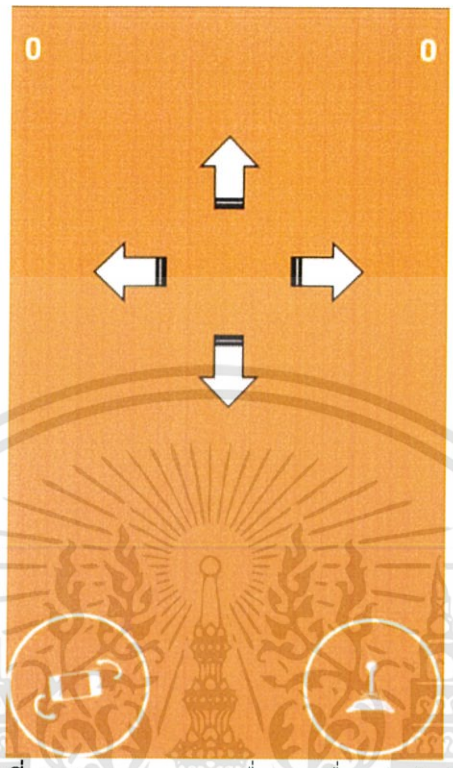
รูปที่ 4.22 การตั้งค่า Wi-Fi Hotspot บนสมาร์ทโฟน

2. การเชื่อมต่อสมาร์ทโฟนกับรถหุ่นยนต์ เปิดแอปพลิเคชันเพื่อใช้งาน โดยต้องทำการระบุ IP Address และ Port ตามที่รถหุ่นยนต์ได้รับจาก Wi-Fi Hotspot แล้วทำการแสดงบนหน้าจอ จากนั้นกดปุ่มเชื่อมต่อ

รูปที่ 4.23 ทดสอบการทำงานของหน้าเมน

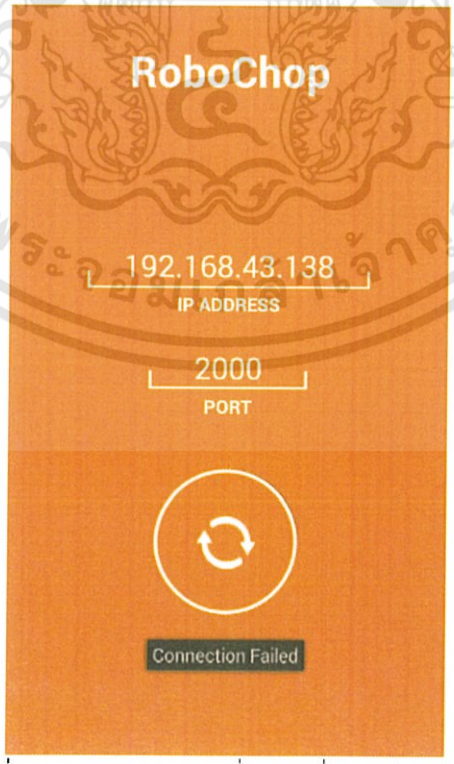
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

➢ กรณีเชื่อมต่อสำเร็จ แอปพลิเคชันจะเข้าสู่หน้าจอควบคุมโดยการใช้ปุ่ม



รูปที่ 4.24 การทดสอบเมื่อการเชื่อมต่อสำเร็จ

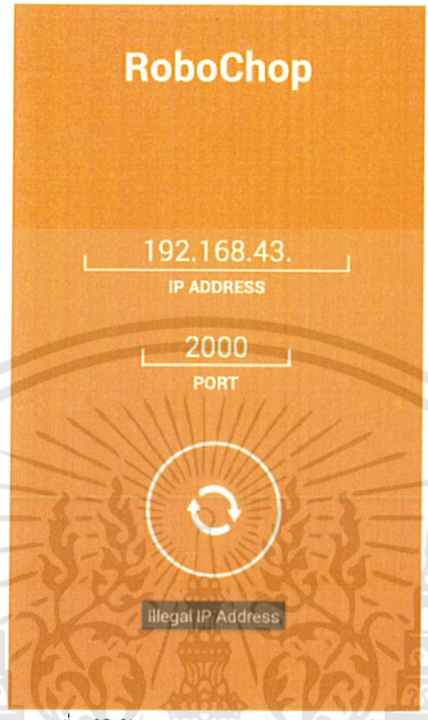
➢ กรณีเชื่อมต่อไม่สำเร็จจะขึ้นข้อความแจ้งว่า “Connection Fail”



รูปที่ 4.25 การทดสอบเมื่อการเชื่อมต่อไม่สำเร็จ

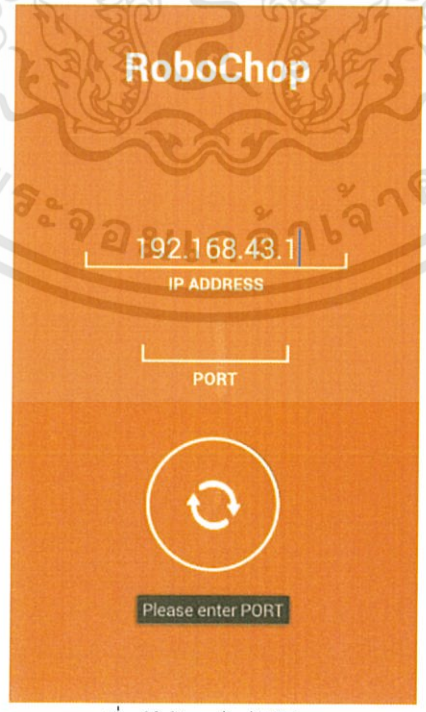
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- กรณีที่ผู้ใช้งานระบุ IP Address ไม่ถูกต้องตามหลัก IPv4 จะขึ้นข้อความแจ้งว่า “Illegal IP Address”



รูปที่ 4.26 การทดสอบเมื่อผู้ใช้งานระบุ IP Address ไม่ถูกต้องตามหลัก IPv4

- กรณีที่ผู้ใช้งานไม่ได้ทำการระบุหมายเลข PORT จะขึ้นข้อความแจ้งว่า “Please enter PORT”

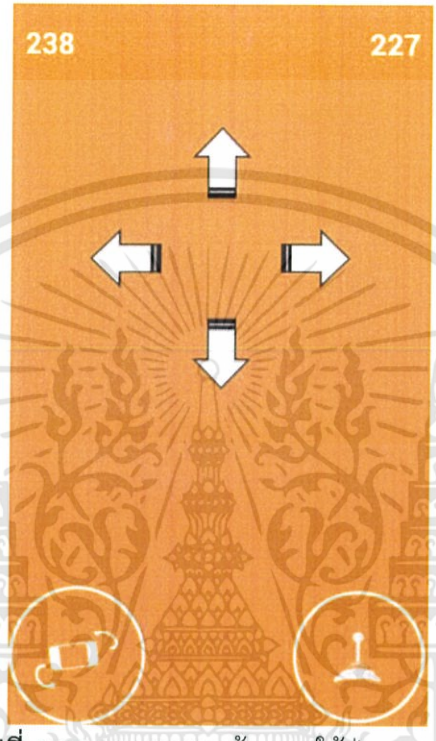


รูปที่ 4.27 การทดสอบเมื่อผู้ใช้งานไม่ได้ทำการระบุหมายเลข PORT

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. การควบคุมรถหุ่นยนต์ ทำการควบคุมรถหุ่นยนต์ผ่านสมาร์ทโฟนโดยเมื่อมีการควบคุมสมาร์ทโฟนจะส่งคำสั่งไปยังรถหุ่นยนต์เพื่อให้เกิดการเคลื่อนที่ โดยสามารถเปลี่ยนรูปแบบการควบคุมได้จากปุ่มทางซ้ายล่างหรือขวาล่าง ทั้งหมด 3 รูปแบบ ดังนี้

   - ใช้ปุ่มกดควบคุม



รูปที่ 4.28 การทดสอบด้วยการใช้ปุ่มควบคุม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ใช้การเอียงสมาร์ทโฟนควบคุม โดยเมื่อทำการเอียงสมาร์ทโฟนมากขึ้นจะทำให้รถหุ่นยนต์เคลื่อนที่ด้วยความเร็วที่เพิ่มขึ้น

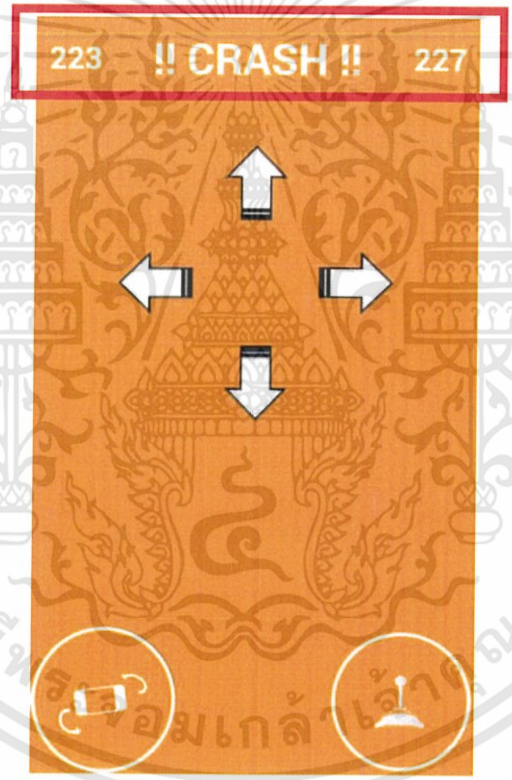
รูปที่ 4.29 การทดสอบด้วยการเอียงสมาร์ทโฟน

- ใช้คันบังคับควบคุม

รูปที่ 4.30 การทดสอบด้วยการใช้คันบังคับควบคุม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. การรับค่าจากรถหุ่นยนต์ ค่าที่สมาร์ทโฟนทำการรับจากรถหุ่นยนต์จะมีอยู่สองค่า คือ ค่าของเซ็นเซอร์ตรวจจับการชน และ ค่าตัวเลขของเซนเซอร์ตรวจจับแสง
   - เซนเซอร์ตรวจจับการชน เมื่อสถานะของเซนเซอร์ตรวจจับการชนของรถหุ่นยนต์เกิดการเปลี่ยนแปลง บอร์ดไมโครคอนโทรเลอร์จะทำการส่งค่ากลับมายังสมาร์ทโฟน จากนั้นสมาร์ทโฟนจะทำการอ่านค่าและแจ้งเตือนผู้ใช้งาน โดยการสั่น ส่งเสียงเตือน และแสดงผลบนหน้าจอสมาร์ทโฟน
   - เซนเซอร์ตรวจจับแสง เมื่อค่าที่เซนเซอร์ตรวจจับแสงวัดได้เกิดการเปลี่ยนแปลง บอร์ดไมโครคอนโทรลเลอร์จะทำการส่งค่ากลับมายังสมาร์ทโฟน จากนั้นสมาร์ทโฟนจะทำการอ่านค่าและแสดงผลตัวเลขเพื่อให้ผู้ใช้งานทราบ



รูปที่ 4.31 การทดสอบการใช้งานเซนเซอร์วัดแสงและเซนเซอร์ตรวจจับการชน

5. การออกจากแอปพลิเคชัน การออกจากแอปพลิเคชันสามารถทำได้โดยการกดปุ่ม “กลับ” บนเครื่องสมาร์ทโฟนได้โดยตรง โดยจะมีการเก็บค่าของ IP Address และ Port ที่ใช้ในการเชื่อมต่อครั้งล่าสุดไว้ เมื่อมีการเปิดใช้งานแอปพลิเคชันในครั้งถัดไปก็จะนำข้อมูลที่เก็บไว้มาระบุเป็นค่าเบื้องต้นให้โดยอัตโนมัติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

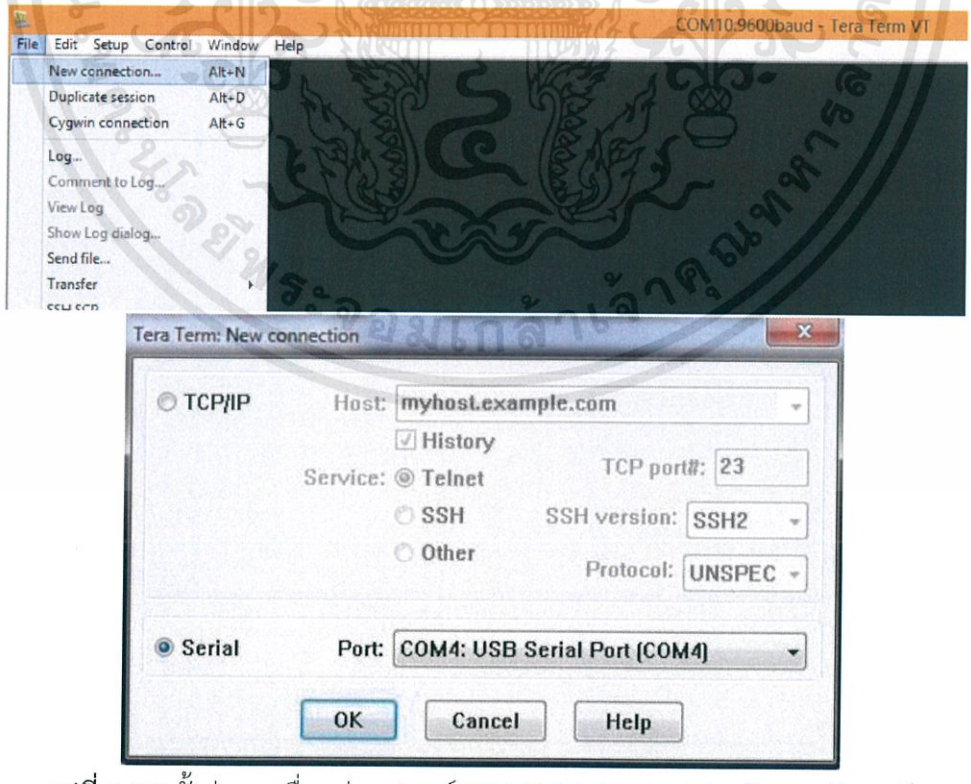
4.3 การทดสอบบนรถหุ่นยนต์

รถหุ่นยนต์ในโครงงานปัญหาพิเศษนี้มีส่วนประกอบคือ บอร์ดไมโครคอนโทรลเลอร์ เซนเซอร์ตรวจจับแสง เซนเซอร์ตรวจจับการชน มอเตอร์ ตัวถังรถหุ่นยนต์ และอุปกรณ์รับส่งสัญญาณ Wi-Fi โดยก่อนที่จะทำการใช้งานรถหุ่นยนต์จะต้องมีการติดตั้งโปรแกรมลงสู่ไมโครคอนโทรลเลอร์ และอุปกรณ์รับส่งสัญญาณ Wi-Fi เพื่อให้สามารถเชื่อมต่อและรับส่งข้อมูลกับสมาร์ทโฟนได้

4.3.1 การทดสอบการตั้งค่าอุปกรณ์รับส่งสัญญาณ Wi-Fi

ในการเชื่อมต่อระหว่างรถหุ่นยนต์กับสมาร์ทโฟนจำเป็นต้องมีโมดูล (Module) ในการเชื่อมต่อกัน ดังนั้นจึงใช้งานอุปกรณ์ RN-XV WiFly Module ซึ่งการเชื่อมต่อของสมาร์ทโฟนกับรถหุ่นยนต์จะใช้ลักษณะการเชื่อมต่อเป็นแบบ Adhoc Mode โดยจะทำการตั้งค่า SSID และพาสเวิร์ดให้ตรงกันกับชื่อและพาสเวิร์ดของ Wi-Fi Hotspot ที่เปิดไว้ที่สมาร์ทโฟน ส่วนการตั้งค่าไอพีจะกำหนดให้รับค่า IP Address จาก Wi-Fi Hotspot โดยจะใช้ค่า IP Address ตามที่ Wi-Fi Hotspot ทำการกำหนดให้ในแต่ละครั้ง โดยการตั้งค่าของ RN-XV WiFly Module นั้นจะใช้งานโปรแกรม Tera Term ดังนี้

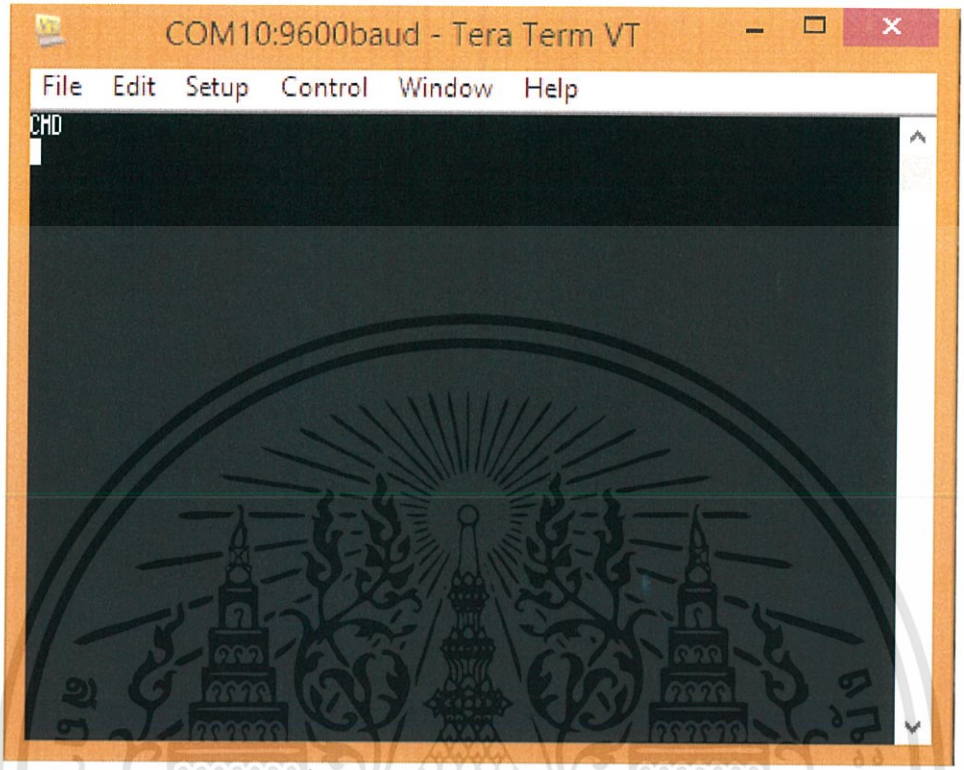
1. ทำการเปิดโปรแกรม Tera Term ขึ้นมาจากนั้นเลือก File -> New Connection เพื่อเป็นการเลือก port ที่ต้องการจะเชื่อมต่อ โดยเลือก Serial จะมีพอร์ตที่เราได้ทำการเชื่อมต่ออยู่



รูปที่ 4.32 ตั้งค่าการเชื่อมต่ออุปกรณ์ RN-XV WiFly Module กับคอมพิวเตอร์

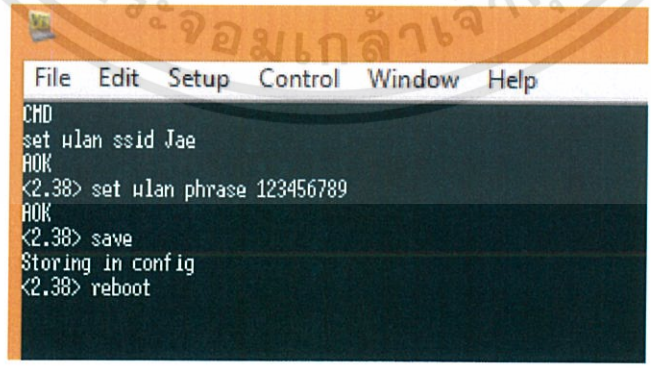
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. จากนั้นทำการเข้าโหมดการตั้งค่า Command Mode โดยให้ป้อนคาสั่ง $$$ โปรแกรมจะตอบกลับมาเป็น CMD ดังรูป



รูปที่ 4.33 การเข้า Command Mode

3. จากนั้นทำการตั้งค่า SSID ด้วยคำสั่ง set wlan ssid ตามด้วยชื่อของ Wi-Fi Hotspot และพาสเวิร์ด ด้วยคำสั่ง set wlan phrase ตามด้วยรหัสของ Wi-Fi Hotspot ให้ตรงกันกับชื่อและพาสเวิร์ดของ Wi-Fi Hotspot ที่เปิดไว้ที่สมาร์ทโฟน ทำการบันทึกการตั้งค่าโดยใช้คำสั่ง save และทำการ reboot ก็ตั้งค่าเสร็จเรียบร้อย

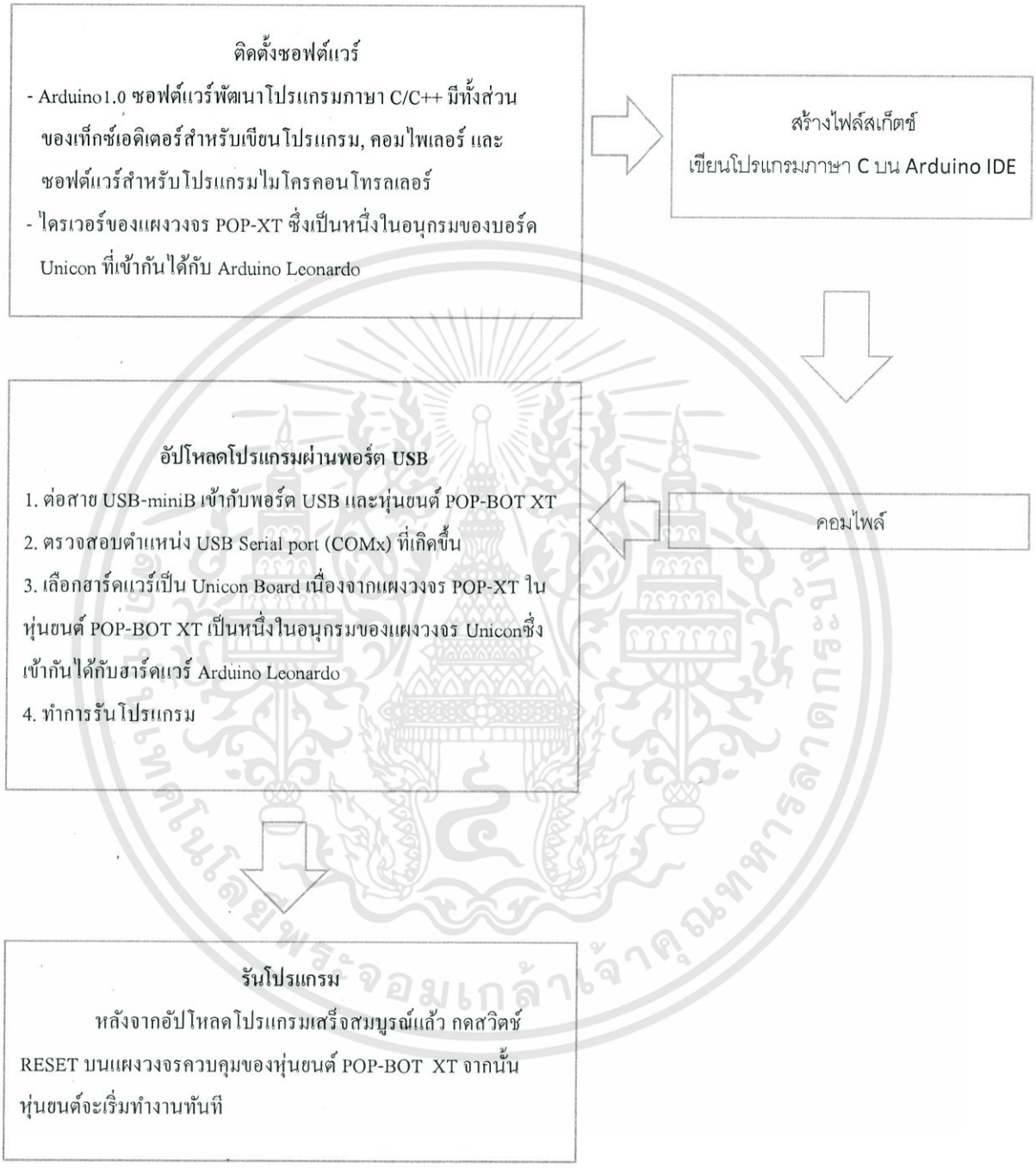


รูปที่ 4.34 ตั้งค่าลงบน RN-XV WiFly Module

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

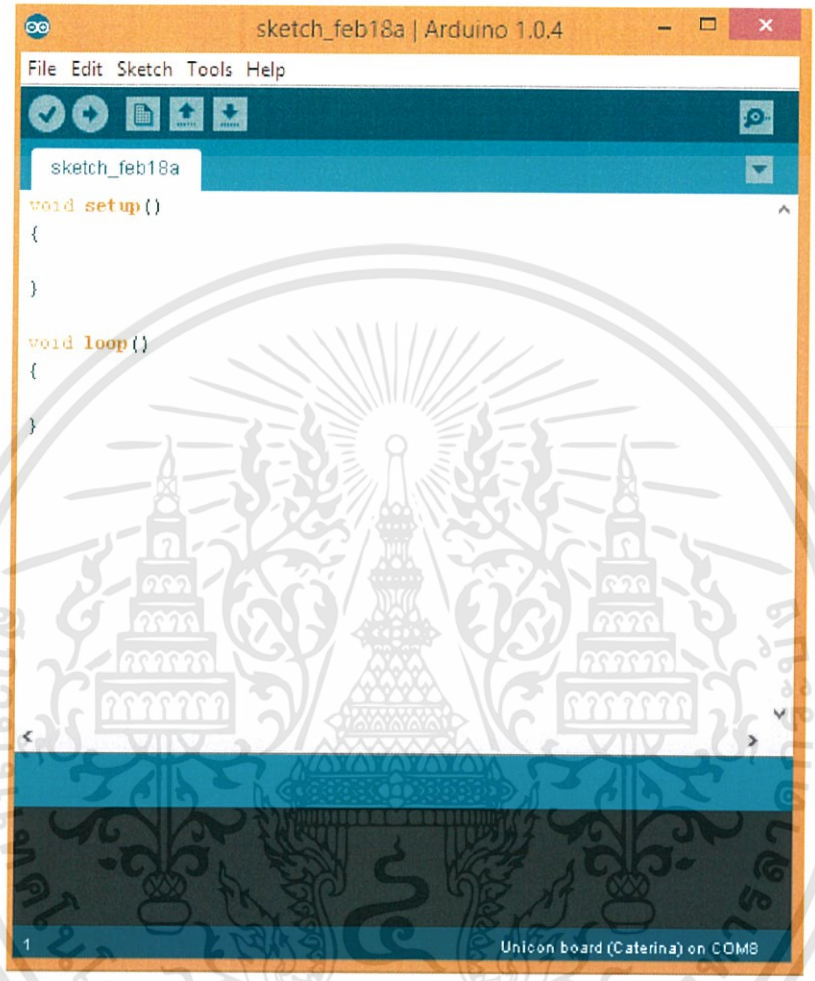
4.3.2 การติดตั้งโปรแกรมลงสู่ไมโครคอนโทรเลอร์

1. ขั้นตอนการพัฒนาโปรแกรมภาษา C เพื่อควบคุมหุ่นยนต์ POP-BOT XT



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลังจากที่ติดตั้งโปรแกรมและไดรเวอร์ของ POP-BOT XT อันเป็นแผงวงจรควบคุมการทำงานของหุ่นยนต์ POP-BOT XT เรียบร้อยแล้ว เรียกโปรแกรม Arduino1.0 ขึ้นมาใช้งาน เมื่อ Arduino ทำงานในครั้งแรกจะมีหน้าจอดังรูป



รูปที่ 4.35 หน้าแรกโปรแกรม Arduino

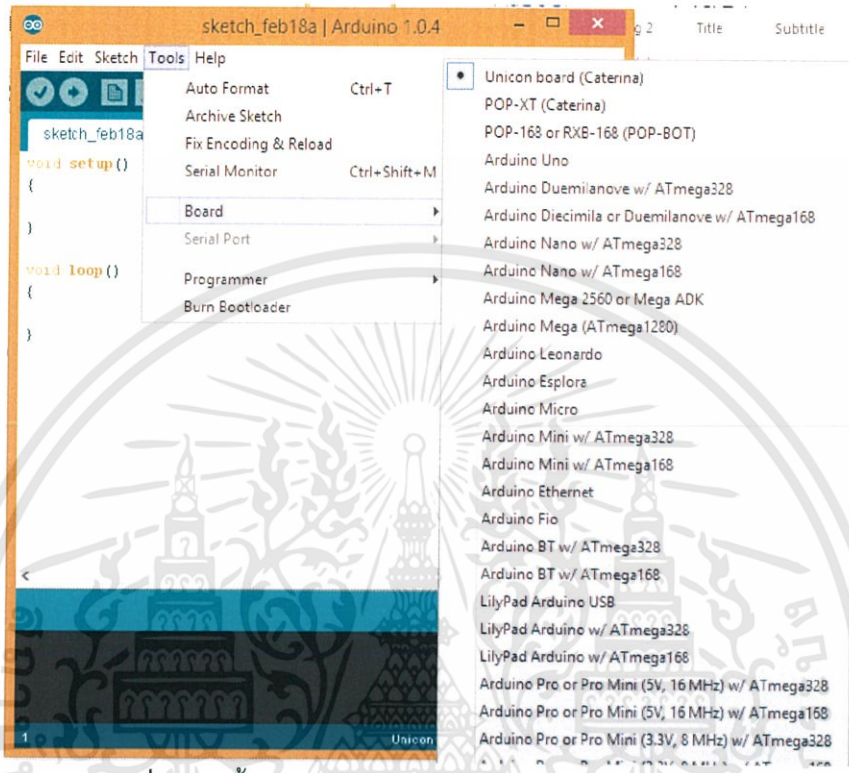
2. กำหนดค่าทางฮาร์ดแวร์เพื่อใช้กับหุ่นยนต์ POP-BOT XT

ในการใช้งานโปรแกรม Arduino ครั้งแรกจะต้องกำหนดค่าของฮาร์ดแวร์ที่ใช้งานร่วมด้วย ซึ่งประกอบด้วยการเลือกแผงวงจรไมโครคอนโทรลเลอร์ที่ใช้และเลือกพอร์ตอนุกรมของคอมพิวเตอร์ที่ต้องการเชื่อมต่อเมื่อกำหนดค่าแล้วครั้งต่อไปที่ทำการเปิดโปรแกรมโปรแกรมจะนำค่าที่กำหนดไว้มาใช้งานทันที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

➢ เลือกเบอร์ของไมโครคอนโทรลเลอร์

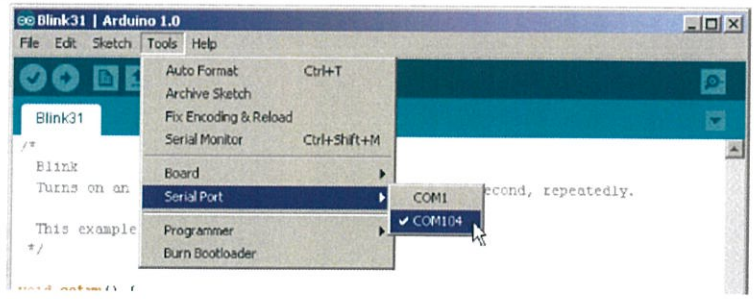
เลือกเมนู Tools -> Board -> POP-XT หรือ Unicon Board (ใช้ได้ทั้งสองรุ่น) ดังรูป



รูปที่ 4.36 ขั้นตอนการเลือกเบอร์ของไมโครคอนโทรลเลอร์

➢ กำหนดพอร์ตอนุกรมที่ใช้ในการติดต่อกับโมดูล

การอัปโหลดโปรแกรมจาก Arduino ไปยังหุ่นยนต์ POP-BOT XT จะกระทำผ่านพอร์ตอนุกรม ซึ่งจะต้องกำหนดหมายเลขพอร์ตที่ใช้ดังนี้เลือกเมนู Tools -> Serial Port โปรแกรมจะแสดงพอร์ตอนุกรมที่มีในคอมพิวเตอร์ ให้ผู้ใช้เลือกพอร์ตตามที่ต้องการ พอร์ตอนุกรมที่ใช้กับหุ่นยนต์ POP-BOT XT เป็นพอร์ตอนุกรมเสมือน (virtual COM port) ที่ เกิดจากการติดตั้งไดรเวอร์ปกติคือพอร์ต COM ที่มีหมายเลขมากกว่า 2 ขึ้นไป

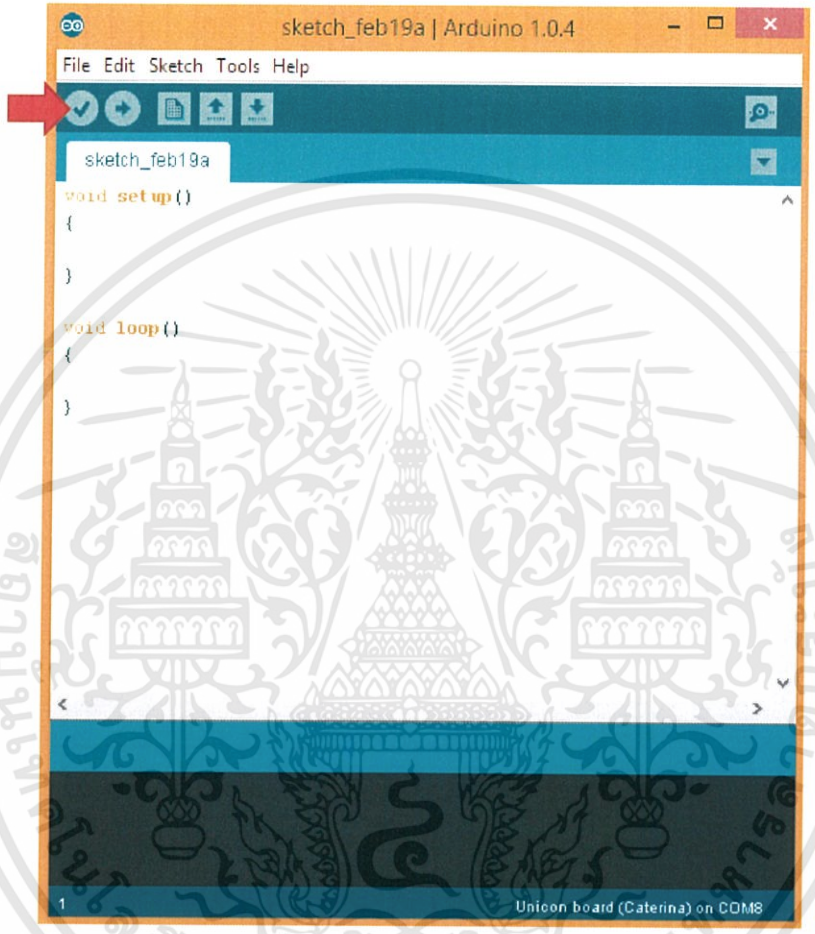


รูปที่ 4.37 การกำหนดพอร์ตอนุกรมที่ใช้ในการติดต่อกับโมดูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

➢ การคอมไพล์โปรแกรมลงไมโครคอนโทรลเลอร์

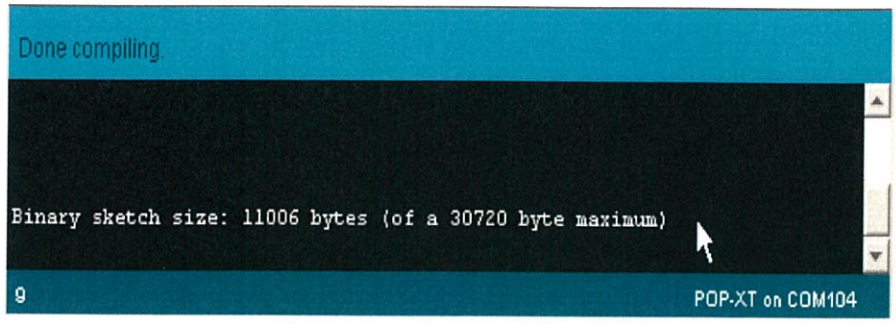
เมื่อทำการเขียนโปรแกรมลงไมโครคอนโทรลเลอร์เสร็จแล้ว จะต้องทำการคอมไพล์เพื่อให้โปรแกรมเป็นภาษาที่ไมโครคอนโทรลเลอร์เข้าใจ สำหรับการคอมไพล์นั้นสามารถกดปุ่มเครื่องหมายถูกเพื่อคอมไพล์ได้เลย



รูปที่ 4.38 ปุ่มคอมไพล์โปรแกรม

เมื่อคอมไพล์โปรแกรมแล้ว ที่แถบแสดงสถานะและหน้าต่างแสดงผลการคอมไพล์ซึ่งเป็นหน้าต่างสีดำอยู่ด้านล่างของโปรแกรมที่แถบแสดงสถานะจะปรากฏข้อความ Done compilling และหน้าต่างแสดงผลแสดงข้อความบอกขนาดของโปรแกรมและขนาดของหน่วยความจำแฟลชของไมโครคอนโทรลเลอร์ทั้งหมดที่ใช้งานได้ เช่น Binary sketch size: 11006 bytes (of a 30720 byte maximum) ดังรูป แสดงว่าโปรแกรมภาษาเครื่องที่ได้จากการคอมไพล์มีขนาด 11,006 ไบต์ จากขนาดของหน่วยความจำแฟลชของไมโครคอนโทรลเลอร์ทั้งหมดที่ใช้งานได้ 30,720 ไบต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

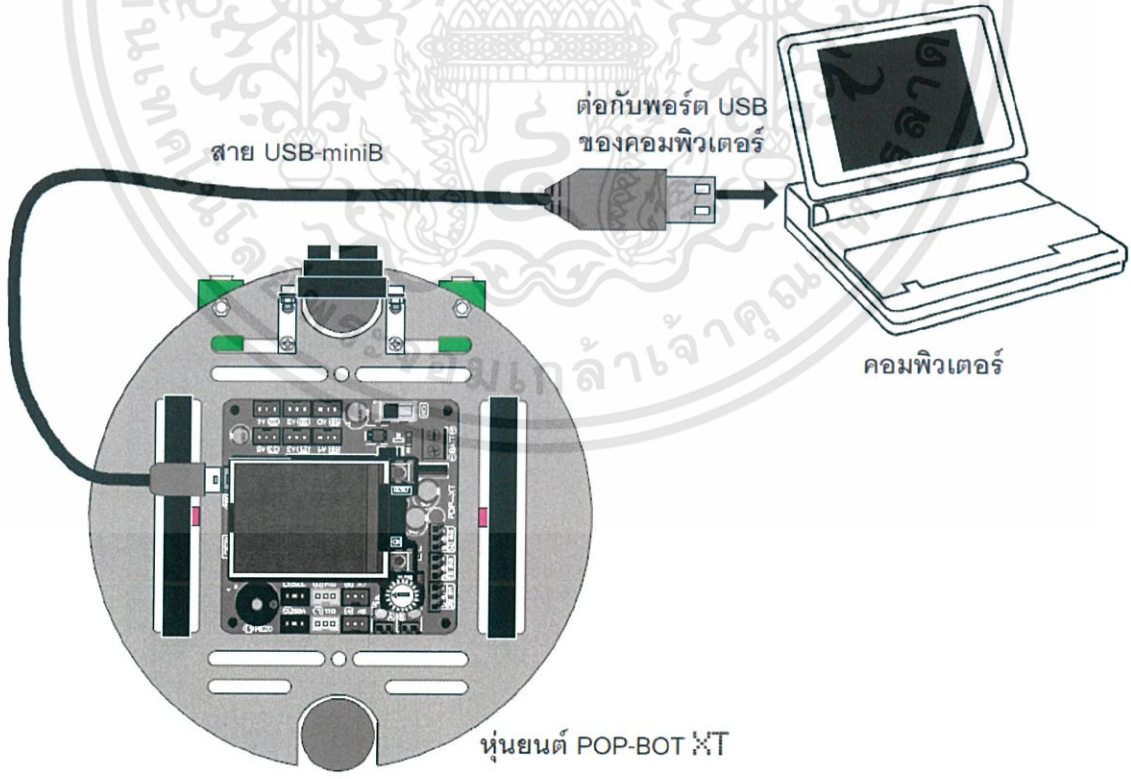


รูปที่ 4.39 การคอมไพล์เสร็จสมบูรณ์

➢ อัปโหลดโปรแกรมไปยังหุ่นยนต์ POP-BOT XT

หลังจากที่คอมไพล์โปรแกรมเสร็จสมบูรณ์แล้วขั้นตอนต่อมาเป็นการอัปโหลดโปรแกรมภาษาเครื่องไปยังหุ่นยนต์ POP-BOT XT ในซอฟต์แวร์ Arduino เรียกกระบวนการการส่งข้อมูลของโปรแกรมภาษา C ที่คอมไพล์แล้วไปยังหุ่นยนต์ POP-BOT XT ว่าอัปโหลด (upload) ซึ่งแตกต่างจากการพัฒนาโปรแกรมของไมโครคอนโทรลเลอร์อื่นๆ ซึ่งเรียกว่าการดาวน์โหลด (download) มีขั้นตอนโดยสรุปดังนี้

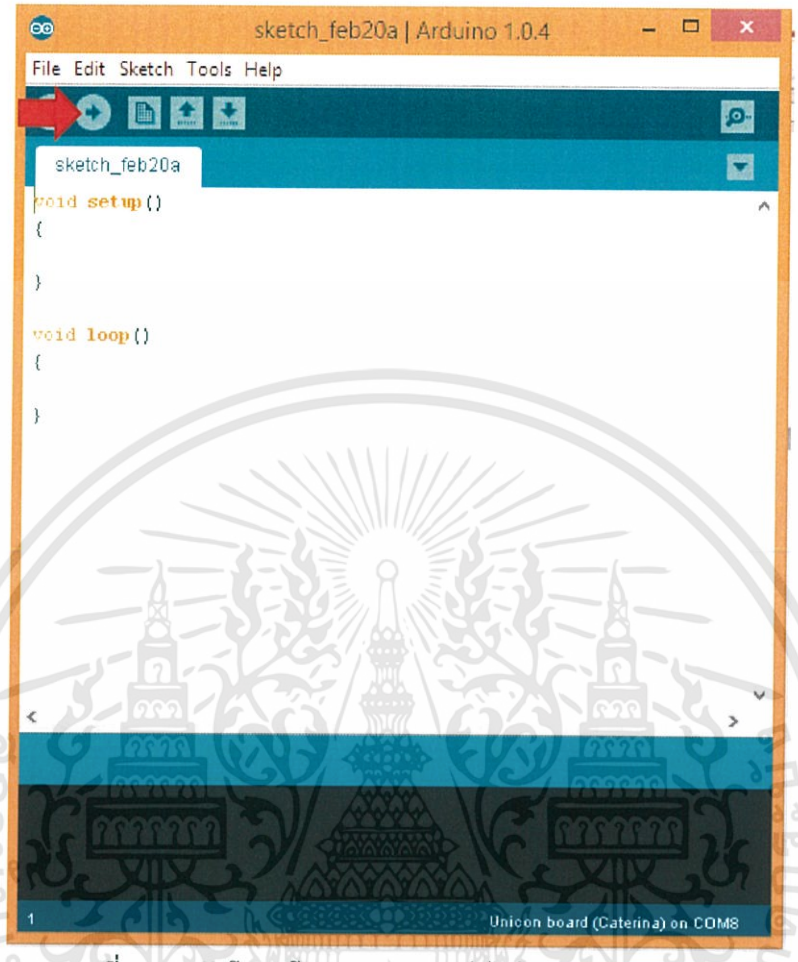
1. เปิดสวิตช์ของหุ่นยนต์ POP-BOT XT รอให้ตัวหุ่นยนต์พร้อมทำงานประมาณ 10 วินาทีจากนั้นต่อสาย USB เข้ากับคอมพิวเตอร์



รูปที่ 4.40 การต่อสาย USB

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. อัปโหลดโปรแกรมไปยังหุ่นยนต์ POP-BOT XT โดยคลิกที่ปุ่มเครื่องหมายลูกศร



รูปที่ 4.41 อัพโหลดโปรแกรมลงบอร์ดไมโครคอนโทรลเลอร์

3. รอจนการอัปโหลดเสร็จสิ้น โปรแกรมจะเริ่มทำงานทันทีหรือกดสวิตช์ RESET อีกครั้งหลังจากที่ได้ลงโปรแกรมบนสมาร์ทโฟนและไมโครคอนโทรลเลอร์ รวมถึงการตั้งค่าต่างๆ ระบบจะพร้อมใช้งาน ซึ่งเราสามารถบังคับหุ่นยนต์ได้หลากหลายรูปแบบ เช่น การเอียง การลากนิ้ว รวมถึงการใช้งานเซนเซอร์ต่างๆ ของไมโครคอนโทรลเลอร์ด้วย อีกทั้งยังใช้สัญญาณ Wi-Fi ในการเชื่อมต่อได้อย่างสมบูรณ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
สรุปผลการดำเนินงานและข้อเสนอแนะ

5.1 สรุปผลการดำเนินงาน

จากการดำเนินงานพัฒนาระบบควบคุมรถหุ่นยนต์ด้วยสมาร์ทโฟนระบบปฏิบัติการแอนดรอยด์ผ่านเครือข่ายไร้สาย สามารถสรุปผลการดำเนินงานในการพัฒนาระบบได้ดังนี้

5.1.1 สรุปผลการดำเนินงาน

การดำเนินงานพัฒนาระบบควบคุมรถหุ่นยนต์ด้วยสมาร์ทโฟนระบบปฏิบัติการแอนดรอยด์ผ่านเครือข่ายไร้สาย เป็นการพัฒนาแอปพลิเคชันด้วยภาษา java ร่วมกับ android API เพื่อทำการควบคุมรถหุ่นยนต์ในรูปแบบต่างๆ โดยโปรแกรมบนไมโครคอนโทรเลอร์ของรถหุ่นยนต์พัฒนาด้วยภาษา C โดยเมื่อผู้ใช้งานสั่งการผ่านสมาร์ทโฟน สมาร์ทโฟนจะส่งคำสั่งไปยังรถหุ่นยนต์เพื่อให้ทำการเคลื่อนไหวในทิศทางต่างๆ โดยที่รถหุ่นยนต์จะมีการใช้งานเซนเซอร์ต่างๆ พร้อมทั้งทำการส่งค่ากลับไปยังสมาร์ทโฟน ซึ่งการสื่อสารกันระหว่างสมาร์ทโฟนและรถหุ่นยนต์จะใช้งานอุปกรณ์ RN-XV WiFly Module ในการเชื่อมต่อกัน การใช้งานระบบเริ่มจากผู้ใช้เปิด Wi-Fi Hotspot บนสมาร์ทโฟนและทำการตั้งค่าตามที่กำหนด จากนั้นเปิดแอปพลิเคชันแล้วระบุ IP Address และ port ที่ปรากฏอยู่ที่รถหุ่นยนต์แล้วทำการเชื่อมต่อ จากนั้นผู้ใช้ควบคุมรถหุ่นยนต์ด้วยรูปแบบการควบคุมต่างๆ โดยจะมีการแจ้งเตือนบนหน้าจอสมาร์ทโฟนเมื่อเซนเซอร์บนรถหุ่นยนต์เกิดการเปลี่ยนแปลง

5.1.2 การพัฒนาระบบ

การดำเนินงานพัฒนาเริ่มจากการศึกษาทฤษฎีและหลักการที่เกี่ยวข้องเพื่อนำมาประยุกต์ใช้ในการพัฒนาระบบ โดยแบ่งออกเป็นการศึกษาการพัฒนาแอปพลิเคชันบนสมาร์ทโฟนระบบปฏิบัติการแอนดรอยด์รวมไปถึงการออกแบบส่วนติดต่อกับผู้ใช้งานบนหน้าจอสมาร์ทโฟน การศึกษาการพัฒนาโปรแกรมบนไมโครคอนโทรลเลอร์รวมไปถึงการศึกษาและเลือกซื้ออุปกรณ์สำหรับใช้ในการพัฒนารถหุ่นยนต์ และการศึกษาเกี่ยวกับการเชื่อมต่อระหว่างสมาร์ทโฟนกับรถหุ่นยนต์ผ่านเครือข่ายไร้สายตลอดจนการออกแบบการเชื่อมต่อและการเลือกใช้อุปกรณ์ในการเชื่อมต่อ จากนั้นจึงเริ่ม

การพัฒนาโปรแกรมบนรถหุ่นยนต์ใช้งานภาษา C ในการพัฒนา โดยเริ่มจากการสร้างโปรแกรมที่มีการใช้งานฟังก์ชันที่ทางผู้ผลิตไมโครคอนโทรลเลอร์มีการจัดเตรียมไว้ จากนั้นจึงทำการสร้างโปรแกรมสำหรับใช้งานจริงโดยฟังก์ชันต่างๆ สำหรับการใช้งานได้มีการประยุกต์ใช้ฟังก์ชันที่มีอยู่เดิมมาปรับปรุงรวมกับฟังก์ชันที่ทำการพัฒนาขึ้นเองเพื่อติดตั้งบนรถหุ่นยนต์ เพื่อให้สามารถควบคุมการหมุนของมอเตอร์และการใช้งานเซนเซอร์ต่างๆ ได้ การพัฒนาแอปพลิเคชันบนสมาร์ทโฟนเริ่มจากการออกแบบส่วนติดต่อกับผู้ใช้งาน จากนั้นจึงเริ่มทำการพัฒนาแอปพลิเคชันเพื่อให้สามารถควบคุมการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เคลื่อนที่ของรถหุ่นยนต์และรับค่าของเซนเซอร์ต่างๆ จากรถหุ่นยนต์ได้ การตั้งค่าอุปกรณ์ RN-XV WiFly Module ต้องมีการออกแบบรูปแบบการเชื่อมต่อระหว่างสมาร์ทโฟนและรถหุ่นยนต์ โดยได้ออกแบบและตั้งค่าให้อุปกรณ์ RN-XV WiFly Module คอยเชื่อมต่อกับ Wi-Fi Hotspot ตามการตั้งค่าที่กำหนดเพื่อทำการสื่อสารระหว่างรถหุ่นยนต์และสมาร์ทโฟน โดยการตั้งค่ามีการระบุรหัสผ่านเพื่อความปลอดภัยไว้ด้วย

เมื่อทำการพัฒนาระบบเรียบร้อยแล้วได้มีการทดสอบการใช้งานโดยเริ่มจากทำการทดสอบการทำงานตามฟังก์ชันต่างๆ ทั้งฟังก์ชันหลักและฟังก์ชันย่อยของโปรแกรมแล้วนำไปปรับแก้ไขจุดบกพร่องต่างๆ จนกระทั่งได้ระบบที่เสร็จสมบูรณ์ ซึ่งผลการทดสอบสรุปได้ว่าการทำงานาสามารถดำเนินการได้ถูกต้องตามที่ได้ออกแบบไว้ และสามารถใช้งานได้อย่างมีประสิทธิภาพ

5.1.3 ประสิทธิภาพและคุณสมบัติของระบบ

1. รถหุ่นยนต์จะเชื่อมต่อกับ Wi-Fi Hotspot ที่กำหนดไว้เท่านั้น เพื่อป้องกันไม่ให้บุคคลอื่นลักลอบเข้าใช้งานรถหุ่นยนต์
2. แอปพลิเคชันสามารถตรวจสอบความถูกต้องของ IP Address ตามหลัก IPv4 และ port ที่ผู้ใช้งานระบุได้
3. สามารถควบคุมการเคลื่อนที่ของรถหุ่นยนต์ได้ 3 รูปแบบ คือ การกดปุ่ม การเอียงสมาร์ทโฟน และการใช้คันบังคับ
4. รถหุ่นยนต์สามารถใช้งานเซนเซอร์ต่างๆ ได้ (ตรวจจับการชน, ตรวจวัดแสง, ไฟ LED) และสามารถส่งค่าของเซนเซอร์กลับไปยังสมาร์ทโฟน
5. รถหุ่นยนต์มีระบบป้องกันการตกจากที่สูง
6. เมื่อรถหุ่นยนต์เกิดการชนจะมีการแจ้งเตือนผู้ใช้งานผ่านสมาร์ทโฟน พร้อมทั้งรถหุ่นยนต์จะทำการเคลื่อนที่ไปยังทิศทางอื่น

5.1.4 ข้อจำกัดของโปรแกรม

1. การเชื่อมต่อระหว่างสมาร์ทโฟนกับรถหุ่นยนต์เพื่อทำการควบคุมเป็นการเชื่อมต่อแบบ Ad-hoc ดังนั้นสมาร์ทโฟนที่จะนำมาใช้งานจึงต้องสามารถทำการปล่อย Wi-Fi Hotspot ได้
2. ข้อจำกัดของระยะทางที่ Wi-Fi Hotspot สามารถปล่อยได้ โดยในขณะที่ทำการควบคุมรถหุ่นยนต์ รถหุ่นยนต์จะต้องอยู่ภายในอาณาเขตที่ Wi-Fi Hotspot ครอบคลุมถึง ซึ่งในโครงงานนี้ใช้สมาร์ทโฟนรุ่น Samsung Galaxy Note 2 และ Samsung Galaxy Nexus ซึ่งสามารถดำเนินงานได้ภายใต้รัศมีประมาณ 20-25 เมตร ในพื้นที่โล่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2 ข้อเสนอแนะ

1. สามารถนำระบบไปพัฒนาต่อให้มีอาณาเขตสำหรับการใช้งานที่กว้างขึ้นและสะดวกมากขึ้น โดยอาจใช้งาน router เป็นตัวกลางในการสื่อสาร หรือใช้อินเทอร์เน็ตและเซิร์ฟเวอร์มาประยุกต์เพื่อสั่งการผ่านทางเครือข่ายอินเทอร์เน็ตเพื่อควบคุมระยะไกล
2. สามารถนำแอปพลิเคชันไปพัฒนาบนสมาร์ทโฟนระบบปฏิบัติการหรืออุปกรณ์อื่นๆ เพื่อความหลากหลายในการใช้งาน เช่น iOS , Windows Phone , Google glass เป็นต้น
3. พัฒนารถหุ่นยนต์ให้สามารถติดตั้งอุปกรณ์เสริมต่างๆ เช่น กล้องถ่ายวีดีโอ แขนกลหยิบจับ เพื่อนำไปประยุกต์ใช้ในการทำงานต่างๆ เช่น หุ่นยนต์สำรวจพื้นที่อันตราย หุ่นยนต์ลำเลียงของในพื้นที่ขนาดเล็ก เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

[1] *Wireless Network Technology ทันสมัยด้วยเทคโนโลยีไร้สายแห่งอนาคต*. (ออนไลน์). แหล่งที่มาจากเว็บไซต์กูเกิ้ล: https://sites.google.com/site/sitesample/wireless-technology. 22 สิงหาคม 2556

[2] *ระบบเครือข่ายไร้สาย (Wireless LAN)*. (ออนไลน์). แหล่งที่มาจากเว็บไซต์สำนักคอมพิวเตอร์ มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี: http://www.cc.kmutt.ac.th/wireless/about.html. 22 สิงหาคม 2556

[3] Mr. Prajak Boonjaratcha. *โครงสร้างสถาปัตยกรรมของแอนดรอยด์*. (ออนไลน์). แหล่งที่มาจาก: http://mobiledep.blogspot.com/2011/09/blog-post_11.html 23 สิงหาคม 2556

[4] นายปรีชากร คุปติมิตร และนายพรเทพ ปั้นสมสกุล. 2012. *รถหุ่นยนต์ติดกล้องผ่านระบบเครือข่ายไร้สาย*

[5] *POP-BOT XT Lite*. (ออนไลน์). แหล่งที่มาจากเว็บไซต์ INEX: http://www.inex.co.th/inexstore/index.php?page=shop.product_details&flypage=flypage.tpl&product_id=479&category_id=6&option=com_virtuemart&Itemid=11#.UhfkRhunqgw

[6] นายชวลิต ทินกรสูติบุตร. *ความรู้พื้นฐานเกี่ยวกับโปโตคอล TCP/IP*. (ออนไลน์). แหล่งที่มาจากเว็บไซต์ T-NET: http://www.tnetsecurity.com/content_basic/tcp_ip_knowledge.php

[7] นางสาวศศิธร ขำศิริ และนางสาวสิทธิรศน์ หงษ์สุวรรณ. 2010. *ระบบควบคุมทิศทางรถบังคับด้วยเทคโนโนยีบลูทูธผ่านโทรศัพท์เคลื่อนที่*

[8] *User Manual and Command Reference 802.11 b/g Wireless LAN Modules*. (ออนไลน์). แหล่งที่มาจากเว็บไซต์ INEX: http://dlnmh9ip6v2uc.cloudfront.net/datasheets/Wireless/WiFi/WiFly-RN-UM.pdf

[9] *การใช้งาน Accelerometer*. (ออนไลน์). แหล่งที่มาจากเว็บไซต์ akexorcist: http://www.akexorcist.com/2013/03/android-code-accelerometer.html

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

[10] *Android API Guide*. (ออนไลน์). แหล่งที่มาจากเว็บไซต์ Android: http://developer.android.com/guide/index.html

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้