



อุปกรณ์แสดงผลข้อความสั้นเป็นอักษรเบรลล์

Refreshable Braille Display for SMS



นาย ตามพงศ์ ชำนาญพุด รหัส 51371291

นาย อรรถพล ทรัพย์สิทธิถาวร รหัส 51371666

ห้องสมุดคณะวิศวกรรมศาสตร์
 วันที่รับ.....12/ก.ย. 2556.....
 เลขทะเบียน.....13675819.....
 เลขเรียกหนังสือ.....พร.....
 มหาวิทยาลัยนเรศวร ๓341 ๑

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
2๕๕๕

สาขาวิชาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

ปีการศึกษา 2555




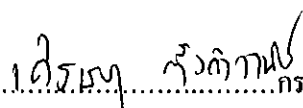
ใบรับรองปริญญาโท

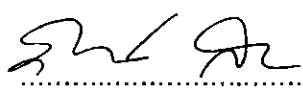
ชื่อหัวข้อ โครงการงาน อุปรกรณ์แสดงผลข้อความสั้นเป็นอักษรเบรลล์
 ผู้ดำเนิน โครงการงาน นาย ตามพงศ์ ชำนาญพุด รหัส 51371291
 นาย อรรถพล ทรัพย์สิทธิถาวร รหัส 51371666
 ที่ปรึกษาโครงการงาน อ.รัฐภูมิ วรานุสาสน์
 สาขาวิชา วิศวกรรมคอมพิวเตอร์
 ภาควิชา วิศวกรรมไฟฟ้าและคอมพิวเตอร์
 ปีการศึกษา 2555

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี อนุมัติให้ปริญญาโทฉบับนี้เป็นส่วนหนึ่ง
 ของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิตสาขา วิชาวิศวกรรมคอมพิวเตอร์


ที่ปรึกษา โครงการงาน
 (อ.รัฐภูมิ วรานุสาสน์)


กรรมการ
 (ผู้ช่วยศาสตราจารย์ ดร. พนมขวัญ ธิษะมงคล)


กรรมการ
 (อ.เสรษฐา ตั้งคำวานิช)


กรรมการ
 (อ.กาญจนาพงศ์ สอนคม)

ชื่อหัวข้อ โครงการงาน	อุปกรณ์แสดงผลข้อความสั้นเป็นอักษรเบรลล์		
ผู้ดำเนินโครงการงาน	นาย ตามพงศ์ ชำนาญพุด	รหัส	51371291
	นาย อรรถพล ทรัพย์สิทธิ์ถาวร	รหัส	51371666
ที่ปรึกษาโครงการงาน	อ.รัฐภูมิ วรรณสาสน์		
สาขาวิชา	วิศวกรรมคอมพิวเตอร์		
ภาควิชา	วิศวกรรมไฟฟ้าและคอมพิวเตอร์		
ปีการศึกษา	2555		

บทคัดย่อ

โครงการนี้จัดทำขึ้นเพื่อที่จะสร้างส่วนติดต่อกับผู้ใช้บนอุปกรณ์โทรศัพท์มือถือระบบแอนดรอยด์สำหรับผู้พิการทางสายตา อุปกรณ์ที่สร้างขึ้นนี้สามารถแสดงผลข้อความเอสเอ็มเอสจากโทรศัพท์มือถือออกมาในรูปแบบของอักษรเบรลล์ ข้อความเอสเอ็มเอสในโทรศัพท์จะถูกประมวลผลโดยโปรแกรมส่วนอุปกรณ์โทรศัพท์มือถือ และจะถูกส่งผ่านพอร์ตอนุกรมไปยังโปรแกรมบนอุปกรณ์แสดงผลอักษรเบรลล์ อักษรเบรลล์แต่ละตัวที่แสดงนั้นจะประกอบไปด้วยแท่งโลหะที่สามารถเคลื่อนที่ได้จำนวน 6 แท่ง แท่งโลหะจะสามารถเคลื่อนที่ขึ้นหรือลงตามสัญญาณไฟฟ้าที่ได้รับ โดยใช้หัวขับโซลินอยด์ แท่งโลหะดังกล่าวเป็นการจำลองผลให้เหมือนกับผลของจุดที่นูนขึ้นของอักษรเบรลล์ที่แสดงบนกระดาษ โดยผู้ที่ใช้นั้นสามารถอ่านตัวอักษรตรงแท่งโลหะที่นูนขึ้นมาแทนการสัมผัสปุ่มในกระดาษที่ละตัวอักษร ผู้ใช้สามารถคีย์บบนอุปกรณ์แสดงผลเพื่อให้อุปกรณ์แสดงผลตัวอักษรตัวถัดไป

Project title Refreshable Braille Display for SMS
Name Mr. Tampong Chumnanpood ID. 51371291
Mr. Attapol Supsititaworn ID. 51371666
Project advisor Mr.Rattapoom Waranusast
Major Computer Engineering
Department Electrical and Computer Engineering
Academic year 2012

Abstract

This project aimed to create a user interface for the visually impaired on **Android mobile phone**. The developed Braille device is able to display an SMS from a mobile phone **in the form of Braille**. The SMS is processed by the phone-side program and transferred to the **Braille-device-side program** via the phone's serial port. Each displayed Braille character is composed of 6 movable pins. The pins can be raised or stayed flat by electrical signal using solenoid actuators. The pins act as the raised dots in the traditional Braille on papers. The user can read a **Braille character** by the raised pins instead of the raised dot. The user can also shift to next character by a **button** on the device.

กิตติกรรมประกาศ

วิทยานิพนธ์นี้สำเร็จลงได้ด้วยความกรุณาอย่างสูงจากท่าน อ.รัฐภูมิ วรรณสาส์น ที่กรุณา
เป็นอาจารย์ที่ปรึกษาวิทยานิพนธ์ โดยให้คำปรึกษา แนะนำ ช่วยหาวิธีการแก้ไขปัญหาอุปสรรค
ต่างๆ ที่เกิดขึ้น ไม่ว่าจะเป็นปัญหาทางการศึกษา หรือปัญหาทางการงาน รวมทั้งให้กำลังใจ
ผู้เขียนด้วยความเมตตา ตั้งแต่เริ่มทำวิทยานิพนธ์ฉบับนี้จนสำเร็จ ขอกราบขอบพระคุณเป็นอย่างสูง

ผู้เขียนขอกราบขอบพระคุณท่าน ผู้ช่วยศาสตราจารย์ ดร. พนมขวัญ ธิษะมงคล อ.เศรษฐา
ตั้งคำวานิช และ อ.ภาณุพงศ์ สอนคม ที่กรุณา รับเป็นกรรมการวิทยานิพนธ์ และได้สละเวลา
ดำเนินการสอบวิทยานิพนธ์นี้ ทั้งได้กรุณา ให้คำแนะนำและชี้แนะแนวทางจนวิทยานิพนธ์นี้มีความ
สมบูรณ์มากยิ่งขึ้น

ผู้เขียน

นาย ตามพงศ์ ชำนาญพุด

นาย อรรถพล ทรัพย์สิทธิถาวร

สารบัญ

หน้า

ใบรับรองปริญญาโท.....	ก
บทคัดย่อภาษาไทย.....	ข
บทคัดย่อภาษาอังกฤษ.....	ค
กิตติกรรมประกาศ.....	ง
สารบัญ.....	จ
สารบัญตาราง.....	ช
สารบัญรูป.....	ซ
บทที่ 1 บทนำ.....	1
1.1 ความเป็นมาและความสำคัญของโครงการ.....	1
1.2 วัตถุประสงค์.....	1
1.3 ขอบเขตของโครงการ.....	2
1.4 แผนการดำเนินงาน.....	2
1.5 ผลที่คาดว่าจะได้รับ.....	2
1.6 งบประมาณ.....	3

สารบัญ (ต่อ)

บทที่ 2 ทฤษฎีที่เกี่ยวข้อง.....	4
2.1 อักษรเบรลล์.....	4
2.2 ระบบปฏิบัติการแอนดรอยด์.....	6
2.3 เครื่องมือในการพัฒนาแอนดรอยด์.....	6
2.4 IOIO for Android.....	7
บทที่ 3 การวิเคราะห์และออกแบบ.....	10
3.1 การออกแบบขั้นตอนการใช้งานอุปกรณ์.....	10
3.2 การแปลงข้อความสั้น.....	11
3.3 การแสดงผลข้อความสั้น.....	13
3.4 การพัฒนาโปรแกรมควบคุมบอร์ด โยโย่.....	15
บทที่ 4 การดำเนินงานและการทดลอง.....	20
4.1 การพัฒนาแอปพลิเคชันควบคุมอุปกรณ์.....	20
4.2 การติดตั้งแอปพลิเคชันที่พัฒนาบน โทรศัพท์แอนดรอยด์.....	23
4.3 การทดสอบแอปพลิเคชันกับบอร์ด โยโย่.....	25
4.4 การประกอบและทดลองอุปกรณ์แสดงผลข้อความสั้น.....	26

สารบัญ (ต่อ)

บทที่ 5 สรุปผลและข้อเสนอแนะ	28
5.1 สรุปผลการทดลอง.....	28
5.2 ปัญหาที่พบและวิธีการแก้ไขปัญหา.....	28
5.3 ข้อเสนอแนะ	29
เอกสารอ้างอิง	30
ภาคผนวก	31
การติดตั้งโปรแกรมสำหรับพัฒนาแอนดรอยด์.....	31
ประวัติผู้ดำเนินโครงการ	38

สารบัญตาราง

ตารางที่	หน้า
1.1 แผนการดำเนินงานทั้งหมดของโครงการ.....	12
3.1 ตารางข้อมูลอักษรเบรลล์แบบดิจิทัล	12

สารบัญรูป

รูปที่	หน้า
2.1 รูปแบบและตัวอย่างการใช้อักษรเบรลล์ภาษาอังกฤษ.....	4
2.2 รูปแบบและตัวอย่างการใช้อักษรเบรลล์ตัวเลขและสัญลักษณ์	5
2.3 บอร์ด IOIO for Android ด้านหน้า	9
2.4 บอร์ด IOIO for Android ด้านหลัง	9
3.1 หลักการทำงานของอุปกรณ์แสดงผลข้อความสั้น	10
3.2 ตำแหน่งบิตของอักษรเบรลล์.....	11
3.3 โซลินอยด์แบบไฟฟ้ากระแสตรง 6 V.....	13
3.4 หน้าที่การทำงานของหมายเลข PIN บนบอร์ดโยโย่.....	14
3.5 องค์ประกอบอุปกรณ์แสดงผล.....	14
3.6 ในโครงการนี้ใช้ตัว Java Platform Eclipse.....	15
3.7 ในโครงการนี้ใช้ตัว Eclipse IDE for Java EE Developers	16
3.8 ในโครงการนี้ใช้ไฟล์ .Zip สำหรับ ระบบปฏิบัติการ Windows.....	16
3.9 ระบบแอนดรอยด์บนคอมพิวเตอร์	17
3.10 Graphical Layout.....	18
3.11 แอปพลิเคชันรับข้อความ SMS เป็นอักษรเบรลล์.....	18

สารบัญรูป (ต่อ)

3.12 ส่วนประกอบของโปรแกรมควบคุมอุปกรณ์.....	19
4.1 การเพิ่มแพ็คเกจการอนุญาตการให้สิทธิเพื่อได้รับ SMS ได้.....	20
4.2 การแบ่งหน้าที่การทำงานบนส่วนติดต่อกับผู้ใช้.....	21
4.3 การ Add Project IOIOLib.....	22
4.5 เปรียบเทียบโค้ดส่วนติดต่อกับผู้ใช้และส่วนควบคุมบอร์ด.....	23
4.6 ไฟล์ .apk ที่ได้ทำการ Export.....	24
4.7 การติดตั้งไฟล์ .apk บนโทรศัพท์แอนดรอยด์.....	24
4.8 การตั้งค่า USB debugging.....	25
4.9 วงจรทดสอบแอปพลิเคชัน.....	25
4.10 วงจรภายในอุปกรณ์แสดงผลข้อความสั้น.....	26
4.11 ภายในอุปกรณ์แสดงผลข้อความสั้น.....	27
4.12 ภายนอกอุปกรณ์แสดงผลข้อความสั้น.....	27

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของโครงการ

ในยุคปัจจุบันนี้ต้องยอมรับว่าในเรื่องของเทคโนโลยีนั้น ได้มีการเติบโตขึ้นอย่างรวดเร็ว จนเทคโนโลยีนั้นเข้ามามีบทบาทภายในชีวิตประจำวันของเราไปอย่างไม่รู้ตัว โดยเฉพาะอย่างยิ่ง การติดต่อสื่อสารกันทางโทรศัพท์มือถือกำลังเป็นที่นิยมและเข้ามามีบทบาทความสำคัญต่อการดำเนินชีวิตประจำวันมากขึ้นทุกขณะ ซึ่งผู้พิการทางสายตาเหล่านี้ถึงแม้ว่าจะสามารถใช้โทรศัพท์มือถือสื่อสารพูดคุยได้ตามปกติก็ตาม แต่ยังมีข้อจำกัดในการติดต่อสื่อสารทางข้อความสั้น (SMS) จึงเป็นที่มาทำให้เกิดโครงการนี้ขึ้น

โครงการนี้เป็นโครงการเพื่อพัฒนาระบบติดต่อกับผู้ใช้ (User Interface) สำหรับผู้พิการทางสายตา ซึ่งอยู่ในรูปของอุปกรณ์ที่สามารถแปลงข้อความสั้นทางโทรศัพท์มือถือ ให้แสดงผลในรูปแบบของอักษรเบรลล์ (Braille) ที่คนพิการทางสายตา สามารถอ่านเข้าใจได้ โดยอุปกรณ์นี้จะเปิดอ่านข้อความสั้นในโทรศัพท์มือถือ และส่งข้อมูลตัวอักษรที่อ่านได้ไปให้อุปกรณ์แสดงผลซึ่งเป็นกลไกทางกลศาสตร์ เพื่อแสดงผลตัวอักษรเบรลล์ขึ้นมาให้ผู้พิการทางสายตาอ่านได้ ซึ่งโครงการนี้จะช่วยเพิ่มโอกาสในการได้รับข่าวสารและการติดต่อสื่อสารของผู้พิการทางสายตาได้

1.2 วัตถุประสงค์

1. เพื่อสร้างอุปกรณ์แสดงผลข้อความสั้นเป็นอักษรเบรลล์
2. เพื่อช่วยพัฒนาศักยภาพของผู้พิการทางสายตาให้เท่าเทียมกับคนสายตาปกติมากขึ้น
3. เพื่อศึกษาทฤษฎีและเทคโนโลยีการพัฒนาซอฟต์แวร์บนโทรศัพท์เคลื่อนที่ การเชื่อมต่อกับอุปกรณ์ควบคุมด้วยไมโครคอนโทรลเลอร์ และกลไกการแสดงผลอักษรเบรลล์

1.3 ขอบเขตของโครงการ

1. อุปกรณ์แสดงผลข้อความสั้นนี้ถูกพัฒนาขึ้นมาเพื่อใช้กับระบบโทรศัพท์ที่เป็นระบบปฏิบัติการ Android โดยเชื่อมต่อทางสาย USB
2. สามารถแสดงผลอักษรเบรลล์ อังกฤษ Aa-Zz และตัวเลข 0-9 ได้
3. แสดงผลอักษรเบรลล์ได้ครั้งละ 1 ตัวอักษร

1.4 แผนการดำเนินงาน

ตาราง 1.1 การดำเนินงานทั้งหมดของโครงการ

กิจกรรม	เดือน / ปีการศึกษา 2554 (ภาคการศึกษาต้นและปลาย)							
	ก.ค.	ส.ค.	ก.ย.	ต.ค.	พ.ย.	ธ.ค.	ม.ค.	ก.พ.
1. ศึกษาและพัฒนาโปรแกรมบนระบบปฏิบัติการ Android	■	■						
2. ศึกษาบอร์ดไมโครคอนโทรลเลอร์สำหรับ Android (IOIO for Android)	■	■	■					
3. เขียนโปรแกรมควบคุมบอร์ดไมโครคอนโทรลเลอร์			■	■	■			
4. ออกแบบและพัฒนาอุปกรณ์แสดงผลข้อความสั้น					■	■	■	
5. ทดสอบและแก้ไขจุดบกพร่อง							■	
6. สรุปการทำงานและจัดทำเอกสาร								■

1.5 ผลที่คาดว่าจะได้รับ

1. ผู้พิการทางสายตา สามารถนำเอาอุปกรณ์แสดงผลอักษรเบรลล์ไปใช้งานจริงได้
2. เป็นอุปกรณ์ตัวอย่างเพื่อให้บุคคลทั่วไปหันมาใส่ใจผู้พิการเพิ่มมากขึ้น
3. ได้ความรู้เกี่ยวกับ Android OS และบอร์ดไมโครคอนโทรลเลอร์ IOIO for Android

1.6 งบประมาณ

บอร์ด IOIO for Android	1950	บาท
สายเชื่อมต่อ USB กับ โทรศัพท์มือถือ	50	บาท
รวมทั้งหมด	<u>2000</u>	บาท

หมายเหตุ ถัวเฉลี่ยทุกรายการ

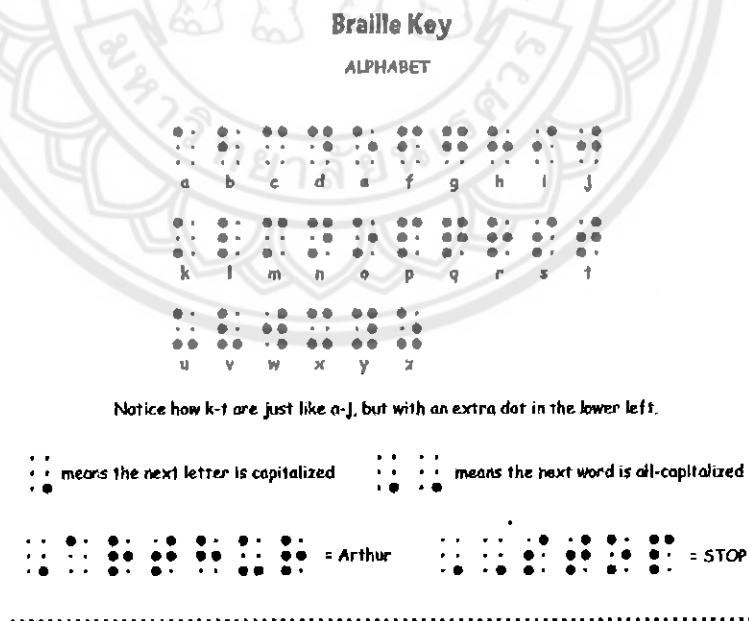


บทที่ 2

ทฤษฎีที่เกี่ยวข้อง

2.1 อักษรเบรลล์

อักษรเบรลล์ คือ อักษรสำหรับใช้ในการเขียนหนังสือของผู้พิการทางสายตา ซึ่งมีลักษณะเป็นจุดนูนบนกระดาษอ่านโดยใช้การสัมผัสด้วยปลายนิ้วมือ ผู้ประดิษฐ์คิดค้นอักษรเบรลล์ คือ หลุยส์ เบรลล์ (Louis Braille) ครูตาบอดชาวฝรั่งเศส อักษรเบรลล์ได้กลายเป็นที่รู้จักกันอย่างกว้างขวางในปี พ.ศ. 2367 อักษรเบรลล์ประกอบด้วยจุดนูนบนกระดาษแข็ง เรียกว่า โซโนกราฟี (Sonography) จะมีลักษณะเป็นจุดนูนๆเล็กๆ ซึ่งจุดเหล่านี้จะอยู่เป็นกลุ่ม โดยใน 1 กลุ่มนั้นจะประกอบไปด้วยจุดเล็กๆ 6 จุด การเขียนอักษรเบรลล์นั้นจะต้องใช้เครื่องมือเฉพาะเรียกว่า สเลท(Slate) และดินสอ (Stylus) ส่วนในการพิมพ์อักษรเบรลล์นั้นจะใช้เครื่องพิมพ์ที่เรียกว่า เบรลล์เตอร์ (Brailier) ในการพิมพ์และกระดาษที่ใช้ต้องมีขนาดหนาเท่ากระดาษวาดรูป จากที่กล่าวมาข้างต้นจะเห็นรูปแบบและตัวอย่างอักษรเบรลล์ ได้จากรูปที่ 2.1 และ 2.2



รูปที่ 2.1 รูปแบบและตัวอย่างการใช้อักษรเบรลล์ภาษาอังกฤษ[1]

NUMBERS

The numbers 1-0 are the same as a-j, but with a number sign before it.

⠠ means the next "letter" or "word" is a number

⠠1	⠠2	⠠3	⠠4	⠠5
⠠6	⠠7	⠠8	⠠9	⠠0

⠠ is a decimal point

⠠479	=	⠠3.14
------	---	-------

PUNCTUATION

⠠.	⠠,	⠠'	⠠-	⠠/
period .	comma ,	apostrophe '	hyphen -	slash /
⠠!	⠠?	⠠\$	⠠:	
exclamation point !	question mark ?	dollar sign \$	colon :	
⠠"	⠠"	⠠(⠠)	
open quote "	close quote "	open parenthesis (close parenthesis)	

รูปที่ 2.2 รูปแบบและตัวอย่างการใช้อักษรเบรลล์ตัวเลขและสัญลักษณ์ [1]

2.2 ระบบปฏิบัติการแอนดรอยด์

แอนดรอยด์ (Android) เป็นระบบปฏิบัติการที่ทำงานบนลินุกซ์ เคอร์เนล สำหรับอุปกรณ์พกพา เช่น โทรศัพท์มือถือ แท็บเล็ตคอมพิวเตอร์ เริ่มพัฒนาโดยบริษัทแอนดรอยด์ จากนั้นบริษัทแอนดรอยด์ถูกซื้อ โดย Google และนำแอนดรอยด์ไปพัฒนาต่อ ภายหลังจากพัฒนาในนามของ Open Handset Alliance[2] ทาง Google ได้เปิดให้นักพัฒนาสามารถแก้ไขโค้ดต่างๆ ด้วยภาษาจาวา และควบคุมอุปกรณ์ผ่านทางชุด Java libraries ที่ Google พัฒนาขึ้น แอนดรอยด์ได้เป็นที่รู้จักต่อสาธารณชนเมื่อวันที่ 5 พฤศจิกายน พ.ศ.2550 โดยทาง Google ได้ประกาศก่อตั้ง Open Handset Alliance[3] กลุ่มบริษัทฮาร์ดแวร์ ซอฟต์แวร์ และการสื่อสาร 48 แห่ง ที่ร่วมมือกันเพื่อพัฒนามาตรฐานเปิด สำหรับอุปกรณ์มือถือ ลิขสิทธิ์ของโค้ดแอนดรอยด์นี้จะใช้ในลักษณะของซอฟต์แวร์เสรี [2]

2.3 เครื่องมือในการพัฒนาแอนดรอยด์

2.3.1 Android SDK

ข้อมูลจาก Android Software Development Kit ซึ่งเป็นชุดโปรแกรมที่ทาง Google พัฒนาออกมาเพื่อแจกจ่ายให้นักพัฒนาแอปพลิเคชัน หรือผู้สนใจทั่วไปดาวน์โหลดไปใช้กันโดยไม่มีค่าใช้จ่าย ซึ่งนี่ก็เป็นหนึ่งในปัจจัยที่ทำให้แอปพลิเคชันบนแอนดรอยด์นั้นเพิ่มขึ้น อย่างรวดเร็ว ซึ่งในชุด SDK นั้นจะมีโปรแกรมและไลบรารีต่างๆ ที่จำเป็นต่อการพัฒนาแอปพลิเคชันบนแอนดรอยด์ อย่างเช่น Emulator ซึ่งทำให้ผู้ใช้สามารถสร้างแอปพลิเคชันและนำมาทดลอง รันบนตัวอีมูเลเตอร์ ก่อน โดยมีสถานะแวดล้อม เหมือนมือถือที่รันระบบปฏิบัติการแอนดรอยด์จริงๆ

2.3.2 Java JDK

เนื่องจากระบบปฏิบัติการแอนดรอยด์นั้น ได้ถูกพัฒนาขึ้นด้วยภาษา Java ดังนั้นในการเขียนโปรแกรมเพื่อพัฒนาซอฟต์แวร์บนแอนดรอยด์ ต้องทำให้ระบบรู้จักกับ Java โดยการลง Java JDK

2.3.3 Eclipse IDE

เป็นโปรแกรมประเภท IDE (Integrated Development Environments) ซึ่งเป็นเครื่องมืออีกอันหนึ่งในการพัฒนา Java Application ที่มีคุณภาพ โดยจะเป็น Freeware ซึ่งตัว eclipse สามารถใช้กับเทคโนโลยี java ในระดับสูงได้

2.3.4 ADT plug-in for Eclipse

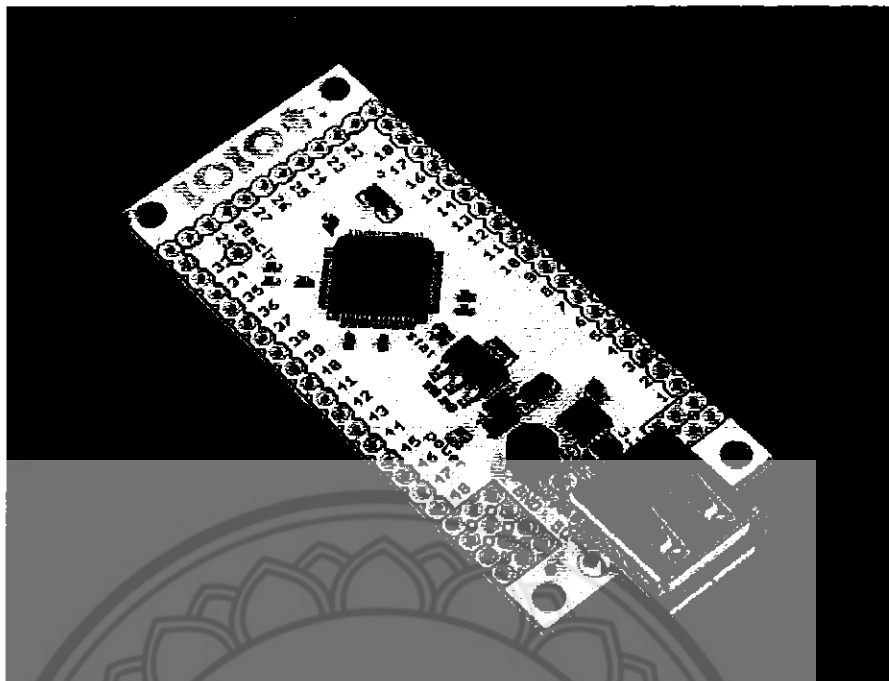
ในการพัฒนา Application บน Android จะใช้ภาษา Java โดยต้องทำการติดตั้งส่วนเสริมที่มีชื่อว่า ADT (Android Development Tools) ซึ่งเป็นส่วนเสริมของ IDE ที่ใช้ในการเขียนโปรแกรม ซึ่ง ADT plug-in นี้จะรวมอยู่เป็นส่วนหนึ่งของ Android SDK

2.4 IOIO for Android

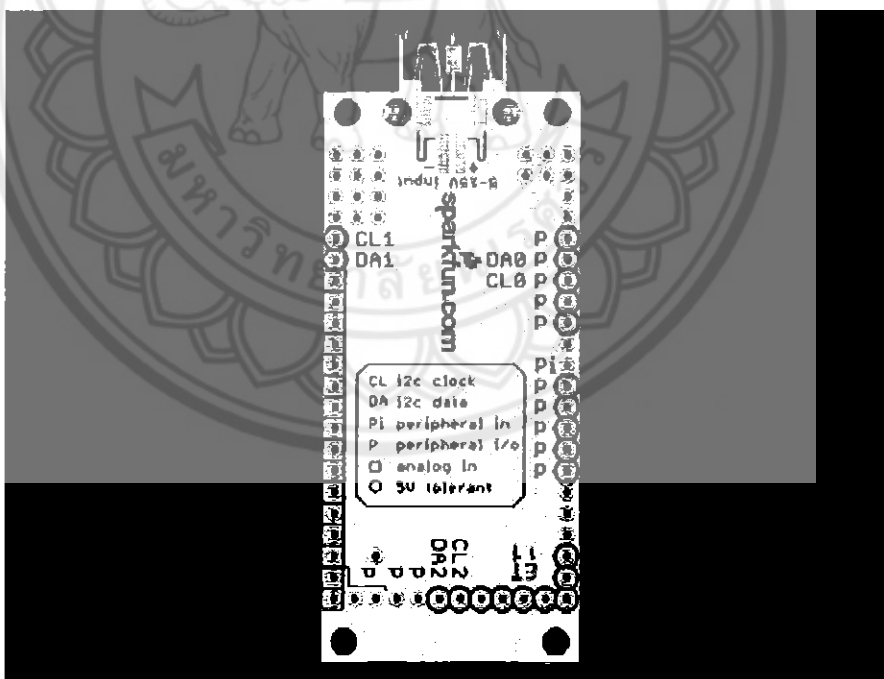
IOIO (อ่านว่า โยโย) เป็นบอร์ดไมโครคอนโทรลเลอร์สำหรับใช้กับ Android Device ทุกประเภท ไม่ว่าจะเป็น Smart Phone , Tablet PC หรือ อุปกรณ์อื่น ๆ เพื่อเป็น Hardware Input และ Output (สามารถใช้ Android OS ตั้งแต่ versions 1.5 เป็นต้นไป) ด้วยอุปกรณ์นี้จึงทำให้สามารถดึงความสามารถของ Android Device ออกมาทำงานทางด้าน Embedded ได้ ภายในบอร์ด IOIO จะโปรแกรม Firmware สำเร็จสำหรับติดต่อผ่าน USB Port การติดต่อระหว่าง Android Device กับ IOIO จะเป็น Protocol ผ่านสาย USB (Protocol นี้เป็น Firmware Open Source สามารถนำไปพัฒนาต่อได้)

คุณสมบัติทางเทคนิคที่สำคัญ :

- ใช้ไมโครคอนโทรลเลอร์เบอร์ PIC24FJ128DA ที่มีโมดูล USB OTG อยู่ภายในจึงสามารถทำงานเป็น USB โฮสต์ได้ และบรรจุเฟิร์มแวร์ IOIO มาพร้อมใช้งานทำให้การพัฒนาแอปพลิเคชันกระทำทางฝั่งอุปกรณ์แอนดรอยด์เท่านั้น ไม่ต้องเขียน โปรแกรมลงในไมโครคอนโทรลเลอร์อีก
- มีพอร์ตอินพุตเอาต์พุต 48 ช่อง
- มีอินพุตอนาล็อก 16 ช่องต่อเข้ากับ โมดูลแปลงสัญญาณอนาล็อกเป็นดิจิตอลความละเอียด 10 บิตภายในตัวชิป
- มีเอาต์พุต PWM ความละเอียด 10 บิต 9 ช่อง
- มีพอร์ตสื่อสารอนุกรม UART 4 ชุด
- มีขาต่อระบบบัส 2 สาย 3 ชุด รองรับการทำงานกับบัส I2C
- มีคอนเน็คเตอร์ USB แบบ A ทำให้ใช้สายเชื่อมต่อพอร์ต USB ที่มีมากับอุปกรณ์แอนดรอยด์ในการเชื่อมต่อได้ทันทีโดยไม่ต้องดัดแปลงใดๆ
- มี LED แสดงผลการทำงาน และ LED แสดงสถานะไฟเลี้ยง
- ไฟเลี้ยง 5 ถึง 12V
- แหล่งจ่ายไฟบนบอร์ดมี 2 ชุดคือ +3.3V สำหรับเลี้ยงวงจรและไมโครคอนโทรลเลอร์ PIC24FJ128 และ +5V 1500mA แบบสวิตซ์ซึ่งทำให้นำไปเลี้ยงอุปกรณ์แอนดรอยด์ที่นำมาต่อร่วมด้วยได้
- เป็นบอร์ดที่พัฒนาขึ้นโดย YTAI (<http://ytai-mer.blogspot.com>) ในแบบโอเพ่นซอร์สที่เปิดรายละเอียดของวงจรอิเล็กทรอนิกส์และแบบของแผ่นวงจรพิมพ์ตลอดจนเฟิร์มแวร์ที่ใช้ควบคุมเพื่อให้ผู้สนใจสามารถนำไปดัดแปลงและต่อยอดด้วยตนเองได้
- การเชื่อมต่อกับอุปกรณ์แอนดรอยด์ใช้รูปแบบการสื่อสารแบบ ADB
- ออกแบบให้สามารถทำงานกับอุปกรณ์แอนดรอยด์ที่ติดตั้งระบบปฏิบัติการแอนดรอยด์ตั้งแต่เวอร์ชัน 1.5



รูปที่ 2.3 บอร์ด IOIO for Android ด้านหน้า [3]



รูปที่ 2.4 บอร์ด IOIO for Android ด้านหลัง [3]

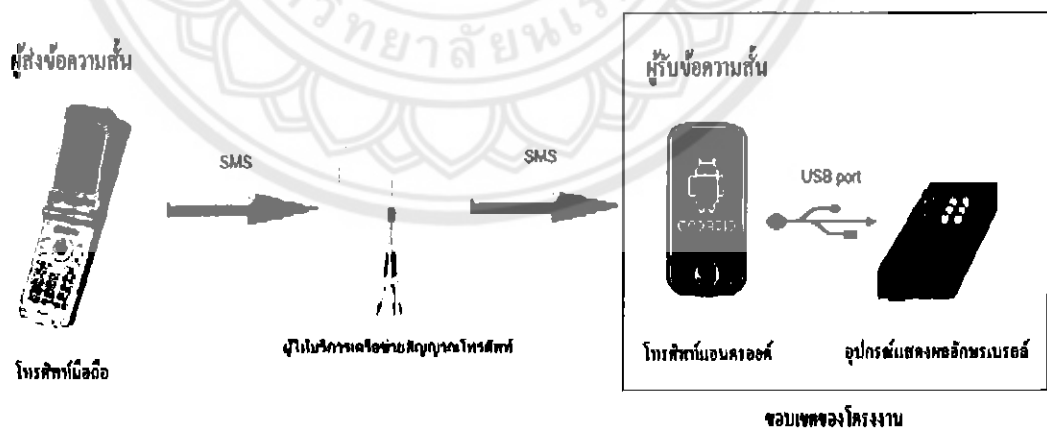
บทที่ 3

การวิเคราะห์และออกแบบ

อุปกรณ์แสดงผลข้อความสั้นถูกพัฒนาเพื่อใช้แสดงตัวอักษรเบรลล์จากข้อความสั้นที่ถูกส่งมายังโทรศัพท์แอนดรอยด์ที่เชื่อมต่อกับตัวอุปกรณ์ การออกแบบสามารถแบ่งออกเป็น 2 ส่วนหลักคือส่วนของซอฟต์แวร์ควบคุมอุปกรณ์ และส่วนของอุปกรณ์แสดงผล

3.1 การออกแบบขั้นตอนการใช้งานอุปกรณ์

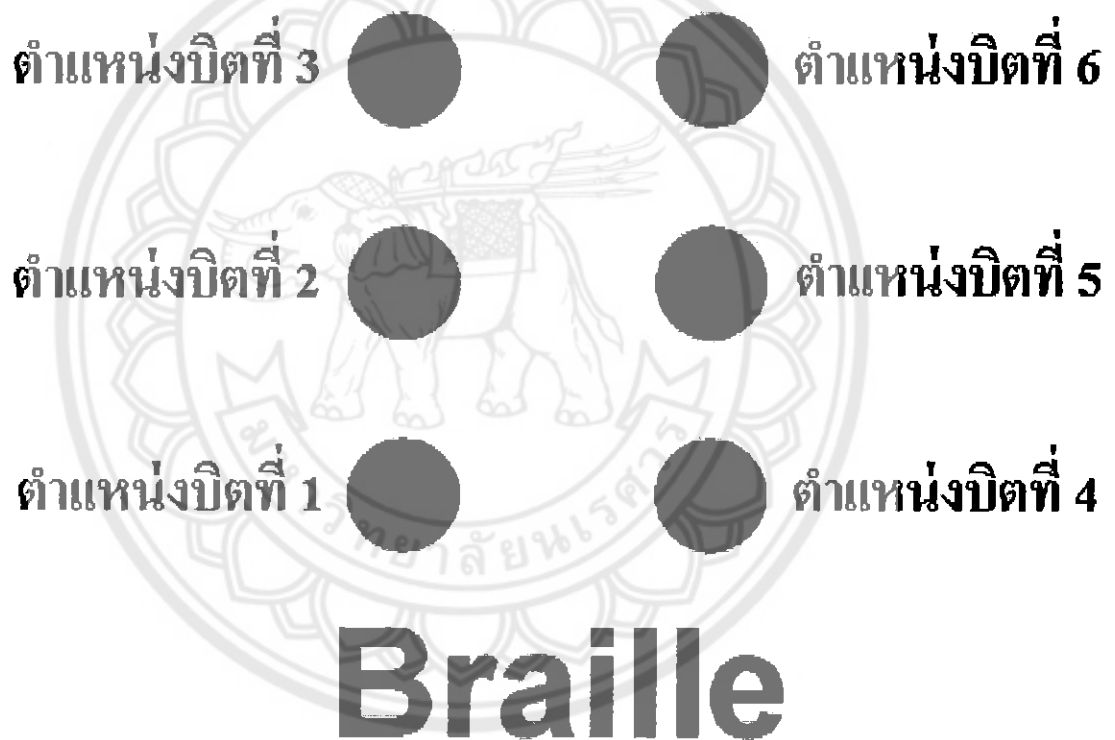
หลักการทำงานของอุปกรณ์แสดงผลอักษรเบรลล์จากข้อความสั้น (SMS) ย่อมาจากคำว่า Short Message Service เป็นบริการส่งข้อความสั้นๆ ลักษณะการใช้งานคล้ายกับการส่งอีเมลล์ แต่สามารถส่งข้อความได้ไม่เกิน 160 ตัวอักษรผ่านทางโทรศัพท์มือถือ โดยเมื่อข้อความถูกส่งมายังโทรศัพท์แอนดรอยด์ของผู้รับที่เชื่อมต่อกับอุปกรณ์แสดงผลอักษรเบรลล์ผ่านสาย USB Port ซอฟต์แวร์ที่ถูกพัฒนาในโทรศัพท์แอนดรอยด์จะทำการควบคุมการทำงานของอุปกรณ์ ให้แสดงผลตามอักขระตัวอักษรของข้อความสั้นที่ได้รับ



รูปที่ 3.1 หลักการใช้งานของอุปกรณ์แสดงผลข้อความสั้น

3.2 การแปลงข้อความสั้น

อักษรเบรลล์ประกอบด้วยจุด 6 ตำแหน่งใน 1 ตัวอักษร มีการเรียงลำดับตำแหน่งจุดของอักษรเบรลล์ดังรูปที่ 3.2 ซึ่งอักษรเบรลล์แต่ละตัวอักษรจะมีการแสดงตำแหน่งจุดแต่ละตำแหน่งแตกต่างกันไปตามตัวอักษร การออกแบบการแปลงตัวอักษรจากข้อความสั้นไปเป็นการแสดงผลยังตำแหน่งของจุดในอักษรเบรลล์ จึงมีการกำหนดค่าในรูปเลขฐานสอง 6 ตำแหน่ง แทนค่าตำแหน่งของจุดในอักษรเบรลล์ ดังตารางที่ 3.1



รูปที่ 3.2 ตำแหน่งบิตของอักษรเบรลล์

ตารางที่ 3.1 ตารางข้อมูลอักษรเบรลล์แบบดิจิทัล

CASE	BRAILLE DIGITAL					
	1	2	3	4	5	6
A,1	0	0	1	0	0	0
B,2	0	1	1	0	0	0
C,3	0	0	1	0	0	1
D,4	0	0	1	0	1	1
E,5	0	0	1	0	1	0
F,6	0	1	1	0	0	1
G,7	0	1	1	0	1	1
H,8	0	1	1	0	1	0
I,9	0	1	0	0	0	1
J,0	0	1	0	0	1	1
K	1	0	1	0	0	0
L	1	1	1	0	0	0
M	1	0	1	0	0	1
N	1	0	1	0	1	1
O	1	0	1	0	1	0
P	1	1	1	0	0	1
Q	1	1	1	0	1	1
R	1	1	1	0	1	0
S	1	1	0	0	0	1
T	1	1	0	0	1	1
U	1	0	1	1	0	0
V	1	1	1	1	0	0
W	0	1	0	1	1	1
X	1	0	1	1	0	1
Y	1	0	1	1	1	1
Z	1	0	1	1	1	0
.	0	1	0	1	1	0
,	0	1	0	0	0	0
?	1	1	0	1	0	0
!	1	1	0	0	1	0
\$	0	1	0	1	1	0
:	0	1	0	0	1	0
+	1	1	0	0	1	0
-	0	1	0	0	1	0
*	1	1	0	1	0	0
/	0	1	0	1	1	0
<	0	1	1	1	0	0
>	1	0	0	0	1	1
=	1	1	0	1	1	0

BRAILLE DIGITAL DATA TABLE

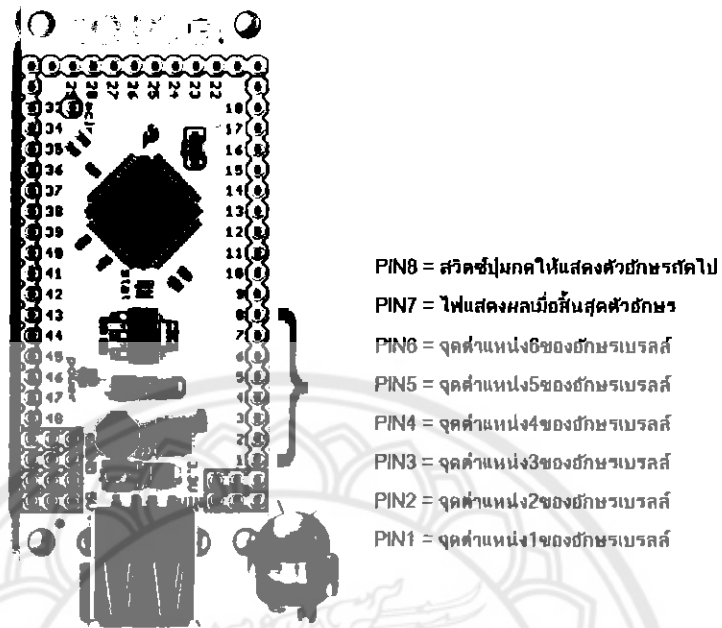
3.3 การแสดงผลข้อความสั้น

จุดตำแหน่งอักษรเบรลล์ที่ใช้แสดงในรูปของแท่งโลหะค้ำขึ้นออกมาจากตัวอุปกรณ์ จะใช้การทำงานของโซลินอยด์แบบไฟฟ้ากระแสตรงเป็นตัวควบคุมแท่งโลหะทั้ง 6 ตำแหน่ง ให้สามารถผลักค้ำแท่งโลหะขึ้นและลดลงตามตำแหน่งการทำงานของจุด ตำแหน่งนั้นๆ โดยโซลินอยด์แบบไฟฟ้ากระแสตรงมีการทำงานเมื่อจ่ายไฟ แท่งโลหะจะเคลื่อนที่ออกมาและเมื่อหยุดจ่ายไฟแท่งโลหะจะเคลื่อนที่กลับเข้าไปยังตำแหน่งเดิม

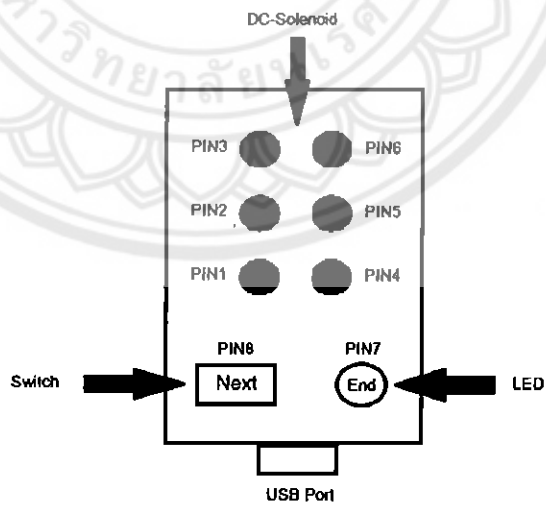


รูปที่ 3.3 โซลินอยด์แบบไฟฟ้ากระแสตรง 6 V[5]

เนื่องจากอุปกรณ์แสดงผลข้อความสั้นมีขีดจำกัดแสดงผลอักษรเบรลล์ได้ทีละ 1 ตัวอักษร จึงต้องเพิ่มปุ่มสวิตช์กดให้อุปกรณ์แสดงผลข้อความสั้นทำการแสดงตัวอักษรในตัวถัดไป เมื่ออุปกรณ์ทำการแสดงผลตัวอักษรตัวสุดท้ายของข้อความสั้น หลอด LED ที่ใช้บอกการสิ้นสุดของข้อความจะทำงาน โดยการเขียนแอสพลีเคชันสั่งงานบอร์ดโยโย่ให้สามารถควบคุมการทำงานเหล่านี้ ต้องกำหนดหมายเลข PIN บนบอร์ดโยโย่ โดยมีการกำหนดไว้ดังรูปที่ 3.4



รูปที่ 3.4 หน้าที่การทำงานของหมายเลข PIN บนบอร์ดไอซี



รูปที่ 3.5 องค์ประกอบอุปกรณ์แสดงผล

3.4 การพัฒนาโปรแกรมควบคุมบอร์ดโยโย่

การเขียนแอปพลิเคชันสำหรับใช้งานบนโทรศัพท์ระบบแอนดรอยด์นั้นค่อนข้างสะดวกเพราะเป็นระบบปฏิบัติการที่กำลังเป็นที่นิยมในยุคสมัยปัจจุบันและถูกพัฒนาขึ้นมาให้ทุกคนสามารถทำการพัฒนาแอปพลิเคชันเองได้

3.4.1 เครื่องมือในการพัฒนา

JDK (Java Development Kit) เนื่องจากแอปพลิเคชันบนแอนดรอยด์ถูกพัฒนาด้วยภาษา Java ในการพัฒนาแอปพลิเคชันเองนั้นจึงจำเป็นต้องทำการติดตั้ง JDK (Java Development Kit) เพื่อให้ระบบปฏิบัติการที่ใช้ในขั้นตอนการพัฒนาแอปพลิเคชันรู้จักกับภาษา Java สามารถโหลดได้จาก <http://www.oracle.com/technetwork/java/javase/downloads/index.html>



รูปที่ 3.6 ในโครงการนี้ใช้ตัว Java Platform Eclipse

Eclipse คือโปรแกรมที่ใช้สำหรับพัฒนาภาษา Java ซึ่งใช้ในการพัฒนา Application Server ได้อย่างมีประสิทธิภาพ เนื่องจาก Eclipse เป็นซอฟต์แวร์ Open Source ที่ใช้ในการพัฒนาซอฟต์แวร์ ทำให้ความก้าวหน้าในการพัฒนาของ Eclipse เป็นไปอย่างต่อเนื่องและรวดเร็ว สามารถโหลดได้จาก <http://www.eclipse.org/downloads/>



Eclipse Downloads

Compare Packages Older Versions Eclipse Indigo (3.7.2) Packages (Windows)

- Eclipse IDE for Java EE Developers**, 212 MB
Downloaded 3,865,244 Times Details Windows 32 Bit Windows 64 Bit
- Eclipse Classic 3.7.2**, 174 MB
Downloaded 2,877,513 Times Details Other Downloads Windows 32 Bit Windows 64 Bit
- Eclipse IDE for Java Developers**, 128 MB
Downloaded 1,452,028 Times Details Windows 32 Bit Windows 64 Bit
- Eclipse IDE for C/C++ Developers (Includes Incubating components)**, 228 MB
Downloaded 1,452,028 Times Details Windows 32 Bit Windows 64 Bit

รูปที่ 3.7 ในโครงการนี้ใช้ตัว Eclipse IDE for Java EE Developers

Android SDK (Android Software Development Kit) คือโปรแกรมสำหรับพัฒนาแอปพลิเคชันบนระบบปฏิบัติการ Android ซึ่งสนับสนุนการพัฒนาด้วยภาษา Java โดยบริษัท Google Inc.สามารถโหลดได้จาก <http://developer.android.com/sdk/index.html>

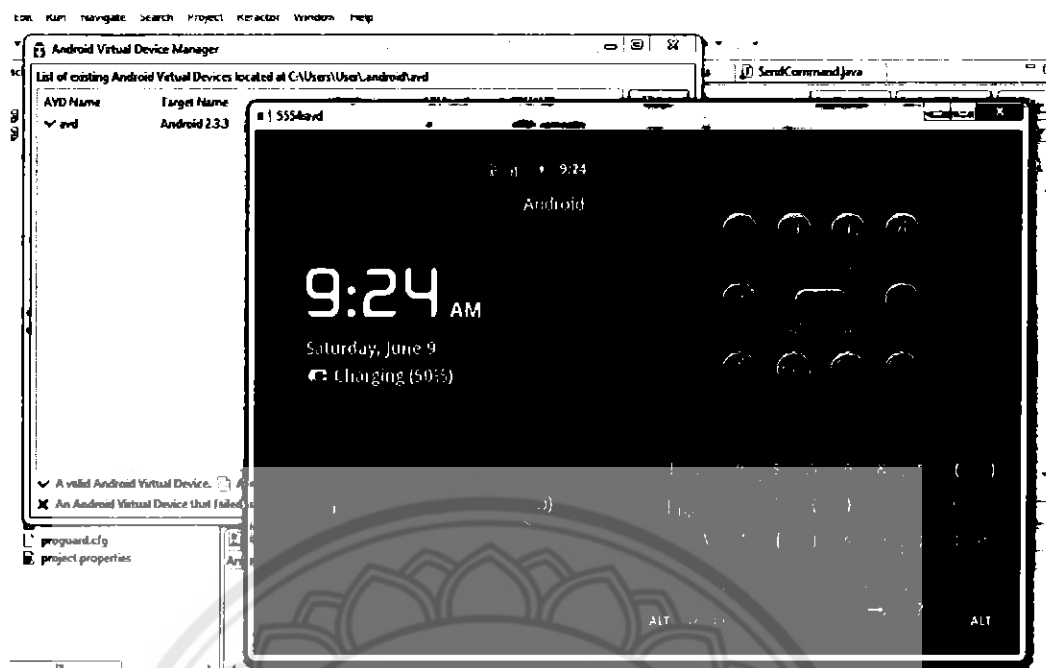
Download the Android SDK

Welcome Developers! If you are new to the Android SDK, please read the steps below, for an overview of how to set up the SI

If you're already using the Android SDK, you should update to the latest tools or platform using the *Android SDK and AVD Manager*. See [Adding SDK Components](#).

Platform	Package	Size	MD5 Checksum
Windows	android-sdk_r18-windows.zip	37448775 bytes	b7fd18b2d0fdecc2a621544d706fa98
	installer_r18-windows.exe (Recommended)	37456234 bytes	48b1fe7b431afe6b9c8a992b75dd898
Mac OS X (intel)	android-sdk_r18-macosx.zip	33903758 bytes	8328e8a5531c9d6f6f1a0261cb97a136
Linux (386)	android-sdk_r18-linux.tgz	29731463 bytes	6cd716d0e04624b865fed3c25b3485c

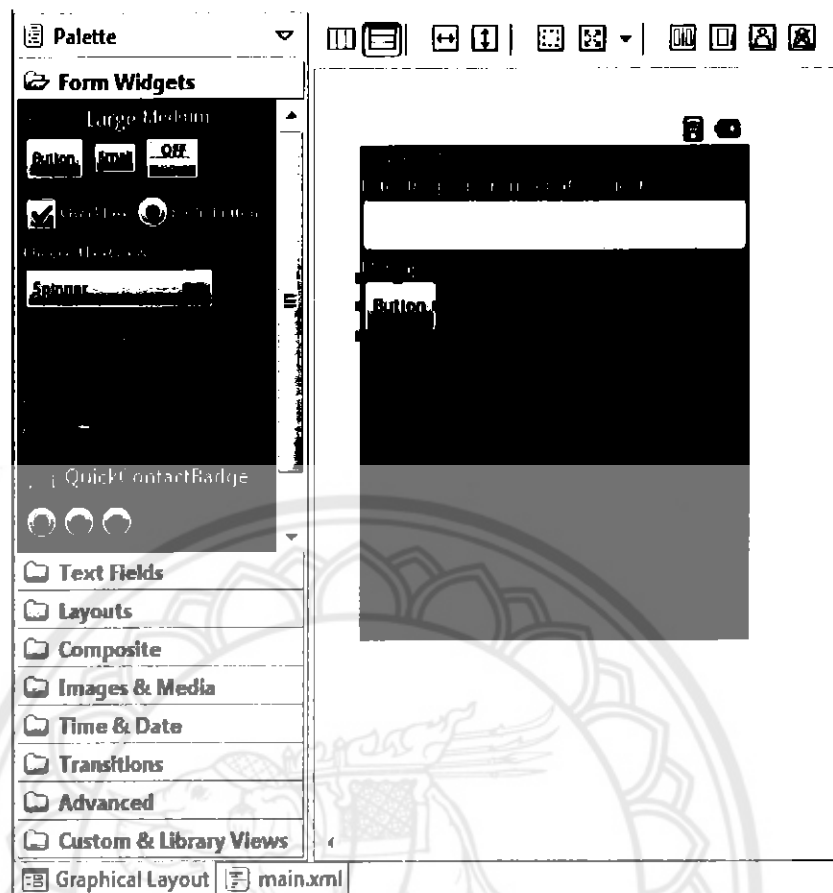
รูปที่ 3.8 ในโครงการนี้ใช้ไฟล์ .Zip สำหรับ ระบบปฏิบัติการ Windows



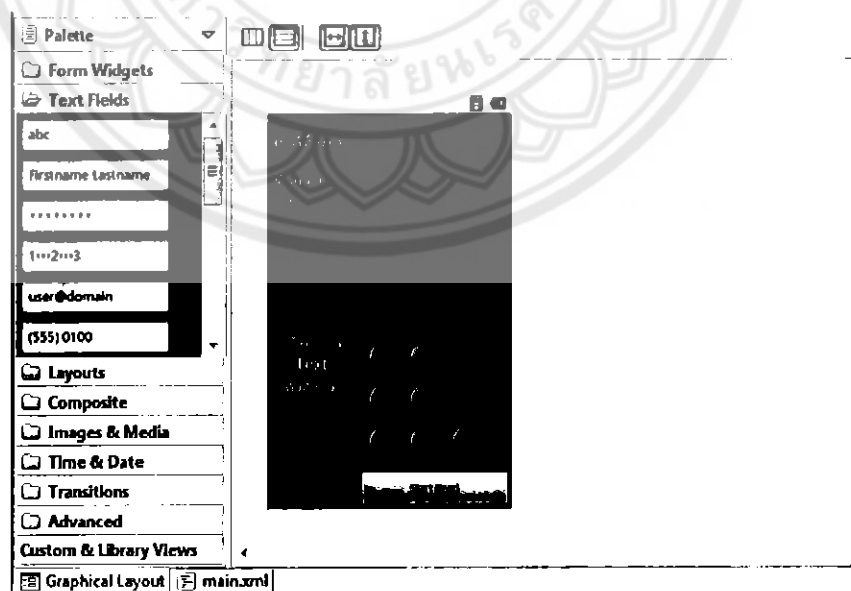
รูปที่ 3.9 ระบบแอนดรอยด์บนคอมพิวเตอร์

3.4.2 การออกแบบแอปพลิเคชัน

เครื่องมือที่ใช้พัฒนาแอปพลิเคชันมีส่วนเสริมสำหรับการออกแบบหน้าจอหลักให้สามารถใช้งานด้วยความสะดวกสบาย การพัฒนาแอปพลิเคชันทำได้โดยทำการสร้าง Android Project ในโปรแกรม Eclipse ที่ได้ทำการลงส่วนเสริมในการพัฒนาไว้แล้ว Eclipse จะทำการสร้างไฟล์ main.xml ให้อัตโนมัติซึ่งสามารถทำการสร้างตกแต่งส่วนติดต่อกับผู้ใช้ของโปรแกรมได้จากสองส่วนคือการแก้ไขไฟล์ main.xml และจากส่วน Graphical Layout ซึ่งในส่วนนี้สามารถทำได้ง่ายและสะดวกเพราะมีฟังก์ชันในการตกแต่งรูปแบบสำหรับแอปพลิเคชันที่ต้องการจะพัฒนาดังรูปที่ 3.10 และ 3.11



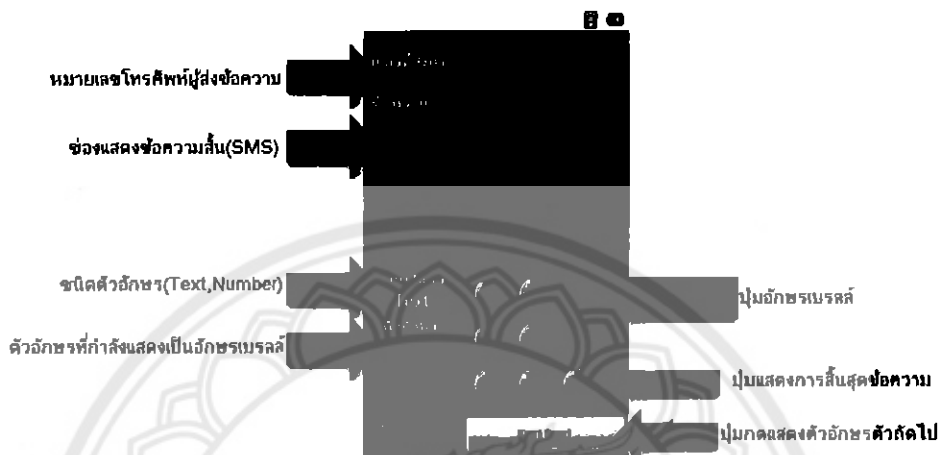
รูปที่ 3.10 Graphical Layout



รูปที่ 3.11 แอปพลิเคชันรับข้อความ SMS เป็นอักษรเบรลล์

3.4.3 การทำงานของโปรแกรมควบคุมอุปกรณ์แสดงผลข้อความสั้น

จากโปรแกรมที่ได้ทำการออกแบบไว้ จะแบ่งหน้าที่การทำงานของแต่ละส่วนในตัวโปรแกรมไว้ดังรูปที่ 3.12



รูปที่ 3.12 ส่วนประกอบของโปรแกรมควบคุมอุปกรณ์

จากรูปที่ 3.12 จะเห็นส่วนติดต่อกับผู้ใช้โปรแกรมควบคุมอุปกรณ์แสดงผลข้อความสั้น ตำแหน่งจุดวงกลมสีเขียว นั้นเป็นตัวแทนตำแหน่งจุด 6 จุดของอักษรเบรลล์และจุดการสิ้นสุดของข้อความ การทำงานของโปรแกรมนั้นคล้ายกับ โปรแกรมรับข้อความทั่วไปที่มีอยู่ในโทรศัพท์มือถืออยู่แล้ว แต่มีการเพิ่มเติมในส่วนการแปลงค่าตัวอักษรให้อยู่ในรูปของอักษรเบรลล์ซึ่งสามารถส่งค่าอักษรเบรลล์ไปยังบอร์ด โยโย่เพื่อ ไปควบคุมการทำงานของอุปกรณ์แสดงผลข้อความสั้น

บทที่ 4

การดำเนินงานและการทดลอง

4.1 การพัฒนาแอปพลิเคชันควบคุมอุปกรณ์

ขั้นตอนแรกในการพัฒนาแอปพลิเคชันทุกครั้งคือการสร้างโปรเจกต์ ซึ่งสามารถทำได้โดยการเปิดโปรแกรม Eclipse คลิกเมนู File > New > Android Project โปรแกรม Eclipse จะทำการสร้าง Package Explorer ขึ้นมาอัตโนมัติ จากนั้นทำการแก้ไขไฟล์ AndroidManifest.xml เพิ่มเติมส่วนการกำหนดสิทธิ์ให้แอปพลิเคชันสามารถรับและส่ง SMS ได้



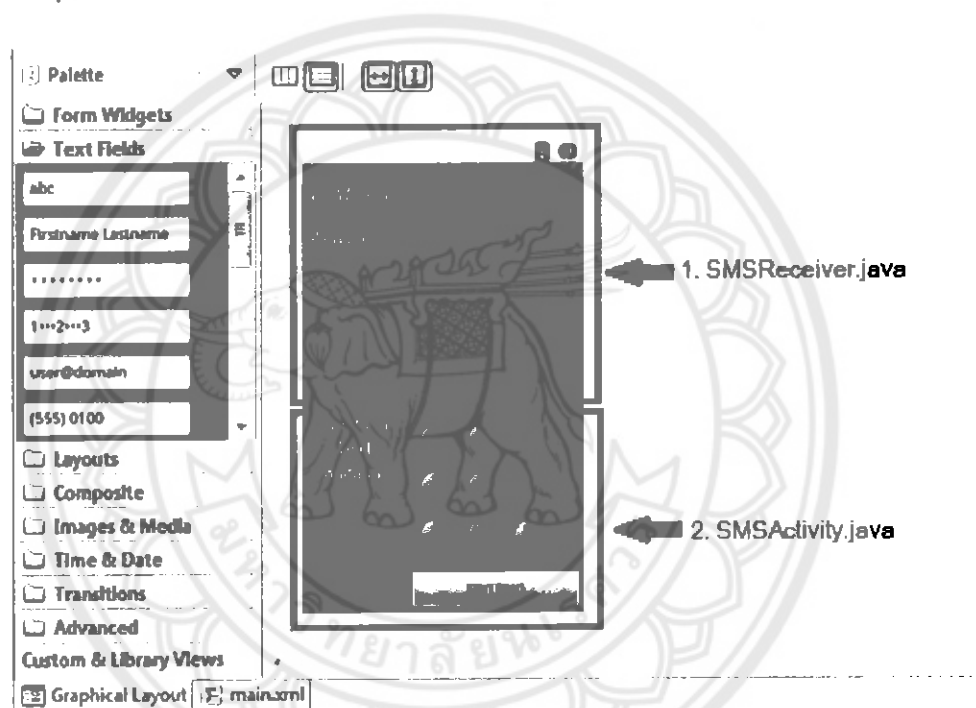
```
1 <?xml version="1.0" encoding="utf-8"?>
2 <manifest xmlns:android="http://schemas.android.com/apk/res/android"
3     package="bet.learn2develop.SMS"
4     android:versionCode="1"
5     android:versionName="1.0" >
6
7     <uses-sdk android:minSdkVersion="8" />
8     <uses-permission android:name="android.permission.SEND_SMS"></uses-permission>
9     <uses-permission android:name="android.permission.RECEIVE_SMS"></uses-permission>
10    <uses-permission android:name="android.permission.INTERNET"></uses-permission>
11
12    <application
13        android:icon="@drawable/ic_launcher"
14        android:label="@string/app_name" >
15        <activity
16            android:name=".SMSActivity"
17            android:label="@string/app_name" >
18            <intent-filter>
19                <action android:name="android.intent.action.MAIN" />
20
21                <category android:name="android.intent.category.LAUNCHER" />
22            </intent-filter>
23        </activity>
24
25        <receiver android:name=".SMSReceiver">
26            <intent-filter>
27                <action android:name="android.provider.Telephony.SMS_RECEIVED" />
28
29                </intent-filter>
30        </receiver>
31
32    </application>
```

รูปที่ 4.1 การเพิ่มแท็กรับการอนุญาตการให้สิทธิ์เพื่อรับ SMS ได้

จากนั้นทำการแก้ไขเพิ่มเติมส่วนติดต่อผู้ใช้แอปพลิเคชัน โดยแก้ไขไฟล์ main.xml เพื่อปรับเปลี่ยนรูปแบบขององค์ประกอบของส่วนติดต่อผู้ใช้ ให้ครบถ้วนตรงตามการออกแบบ

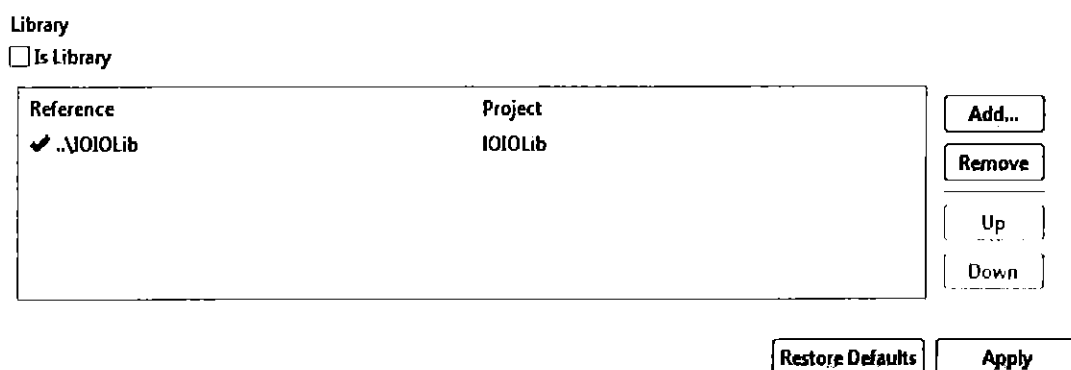
เมื่อได้องค์ประกอบบนส่วนติดต่อกับผู้ใช้ครบถ้วน ทำการกำหนดหน้าที่การทำงานขององค์ประกอบในส่วนติดต่อกับผู้ใช้ จะแบ่งการทำงานส่วนติดต่อกับผู้ใช้ดังนี้

1. ส่วนใช้แสดงอักขระข้อความและเบอร์ผู้ส่ง ส่วนนี้เหมือนโปรแกรมรับข้อความบนโทรศัพท์มือถือทั่วไป ส่วนนี้อยู่ครึ่งบนหน้าจอส่วนติดต่อกับผู้ใช้
2. ส่วนการบอกอักขระ ชนิดตัวอักษรของข้อความและส่วนตำแหน่งจุดวงกลมที่ใช้แทนตำแหน่งจุดอักษรเบรลล์ ซึ่งจะทำงานตำแหน่งเดียวกับการทำงานของโซลินอยด์บนอุปกรณ์แสดงผล



รูปที่ 4.2 การแบ่งหน้าที่การทำงานบนส่วนติดต่อกับผู้ใช้

การแก้ไขเพิ่มเติมการทำงานของแอปพลิเคชันในสองส่วนนี้ ทำโดยแก้ไขไฟล์ .java ในไอดีรีคทอรี `src > net.learn2develop.SMS` เมื่อกำหนดการทำงานในส่วนติดต่อกับผู้ใช้แล้ว ต่อไปคือการทำให้แอปพลิเคชันสามารถควบคุมการส่งงานบอร์ดโฮโยได้ โดยทำการดาวน์โหลด Project IOIOLib จากเว็บไซต์ <https://github.com/ytai/ioio/wiki/Downloads> เพื่อให้โปรแกรม Eclipse รู้จักกับส่วนคำสั่งของการควบคุมบอร์ดโฮโย เมื่อดาวน์โหลดเสร็จสิ้นทำการ Import โฟลเดอร์ Project IOIOLib ใน Package เดียวกับแอปพลิเคชันในโปรแกรม Eclipse จากนั้นคลิกขวาที่ Project แอปพลิเคชันที่พัฒนาเลือก Properties -> Android ในหัวข้อ Library คลิก Add และเลือก IOIOLib ที่ทำการ Import เข้ามา



รูปที่ 4.3 การ Add Project IOIOLib

เมื่อโปรแกรมรู้จักคำสั่งควบคุมบอร์ดแล้ว เขียนโค้ดเพิ่มเติมส่วนการควบคุมบอร์ดในไฟล์

SMSActivity.java

```

DigitalOutput led = ioio_.openDigitalOutput(0, true);
DigitalOutput led1 = ioio_.openDigitalOutput(1, true);
DigitalOutput led2 = ioio_.openDigitalOutput(2, true);
DigitalOutput led3 = ioio_.openDigitalOutput(3, true);
DigitalOutput led4 = ioio_.openDigitalOutput(4, true);
DigitalOutput led5 = ioio_.openDigitalOutput(5, true);
DigitalOutput led6 = ioio_.openDigitalOutput(6, true);
DigitalOutput led7 = ioio_.openDigitalOutput(7, true);
DigitalInput sw1 = ioio_.openDigitalInput(8);

```

กำหนด PIN1-8
เป็นตัวควบคุมโซลินอยด์

กำหนด PIN7
ไฟแสดงผลเมื่อสิ้นสุดข้อความ

กำหนด PIN8
เป็นสวิตช์ไม่Next

รูปที่ 4.4 การกำหนดเลขPINบนบอร์ดไฮโอ


```
case 'A' :
```

```
led1.setImageResource(R.drawable.light);
led2.setImageResource(R.drawable.black);
led3.setImageResource(R.drawable.black);
led4.setImageResource(R.drawable.black);
led5.setImageResource(R.drawable.black);
led6.setImageResource(R.drawable.black);
```

← การแสดงผลตัวอักษร "A"
บนหน้าจอแอปพลิเคชัน

```
case 'A' :
```

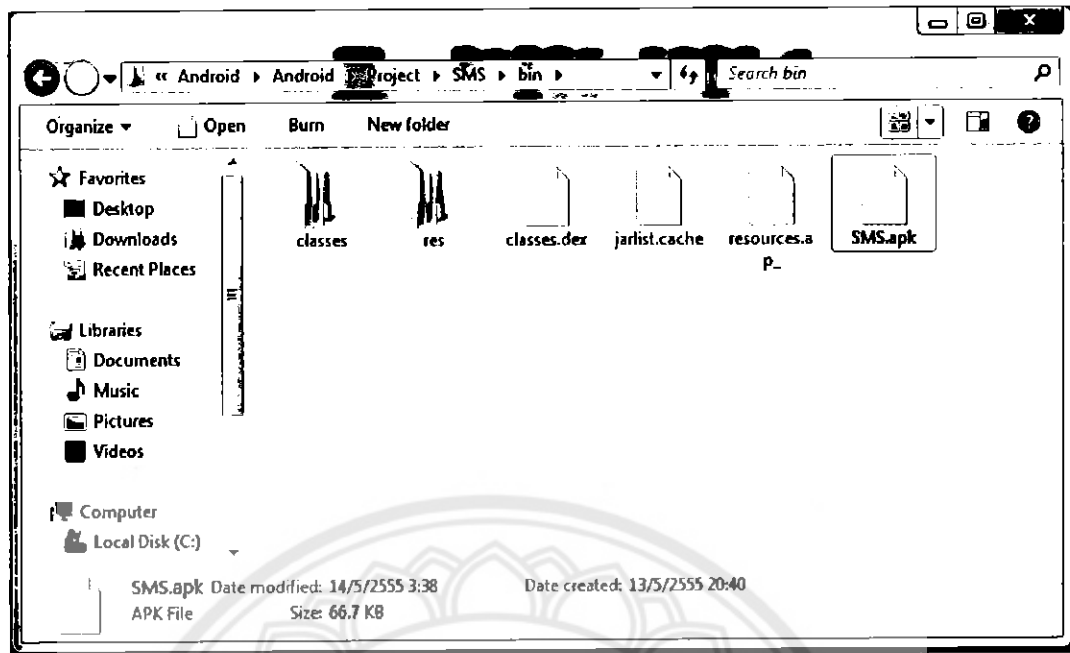
```
led1.write(true); // LED
led2.write(false);
led3.write(false);
led4.write(false);
led5.write(false);
led6.write(false);
led7.write(false);
```

← การแสดงผลตัวอักษร "A"
บนบอร์ดโมโย

รูปที่ 4.5 เปรียบเทียบโค้ดส่วนติดต่อผู้ใช้และส่วนควบคุมบอร์ด.

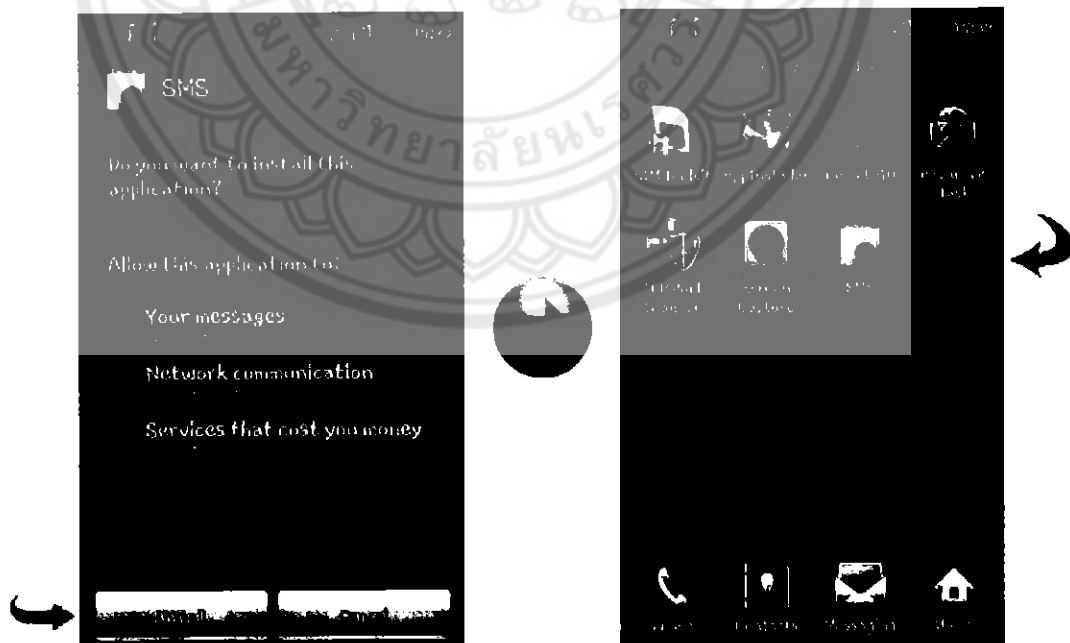
4.2 การติดตั้งแอปพลิเคชันที่พัฒนาบนโทรศัพท์แอนดรอยด์

เมื่อพัฒนาแอปพลิเคชันและทดสอบการทำงานจาก Emulator แล้ว การนำแอปพลิเคชันที่พัฒนามาใช้งานบนโทรศัพท์แอนดรอยด์จริงต้องทำการส่งออก (Export) ไฟล์ .apk เพื่อนำไปติดตั้งบนโทรศัพท์ วิธีสร้างไฟล์ .apk คลิกขวาที่โปรเจกต์เลือกคำสั่ง Android Tool > Export Signed Application Package เมื่อปรากฏหน้าต่าง Export Android Application ทำการตั้งชื่อและกำหนดพื้นที่จัดเก็บไฟล์ .apk



รูปที่ 4.6 ไฟล์ .apk ที่ได้ทำการ Export

จากนั้นคัดลอกไฟล์ .apk ที่ได้ไปจัดเก็บยังพื้นที่ของโทรศัพท์แอนดรอยด์หรือ SD Card เพื่อทำการติดตั้งไฟล์ .apk



รูปที่ 4.7 การติดตั้งไฟล์ .apk บนโทรศัพท์แอนดรอยด์

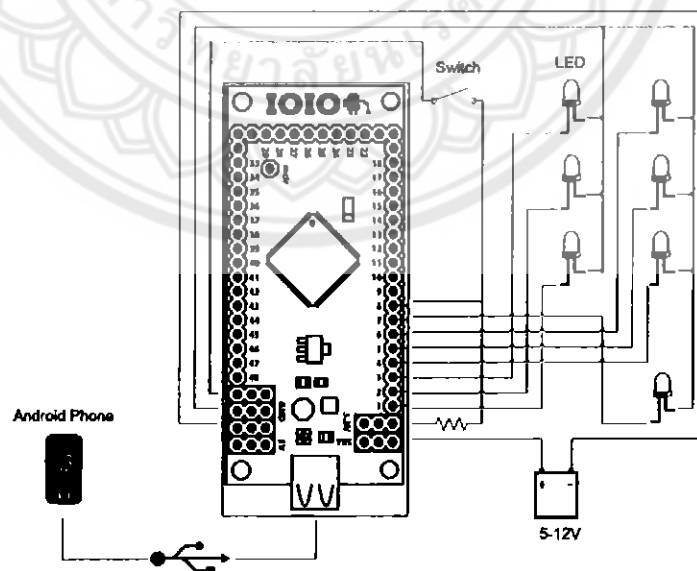
4.3 การทดสอบแอปพลิเคชันกับบอร์ดโยโย่

การเชื่อมต่อโทรศัพท์แอนดรอยด์เข้ากับบอร์ดโยโย่ต้องทำการตั้งค่า USB debugging โดยการเข้าเมนู Settings > Applications > Development แล้วทำเครื่องหมายถูกที่ USB debugging



รูปที่ 4.8 การตั้งค่า USB debugging

การทดสอบแอปพลิเคชันกับบอร์ดโยโย่เบื้องต้น ต่อหลอด LED กับบอร์ดโยโย่แทนโซลินอยด์เพื่อง่ายและสะดวกในการทดสอบแอปพลิเคชันว่าสามารถควบคุมการทำงานที่กำหนดของ PIN บนบอร์ดโยโย่ โดยต่อวงจรในการทดสอบดังรูป 4.9



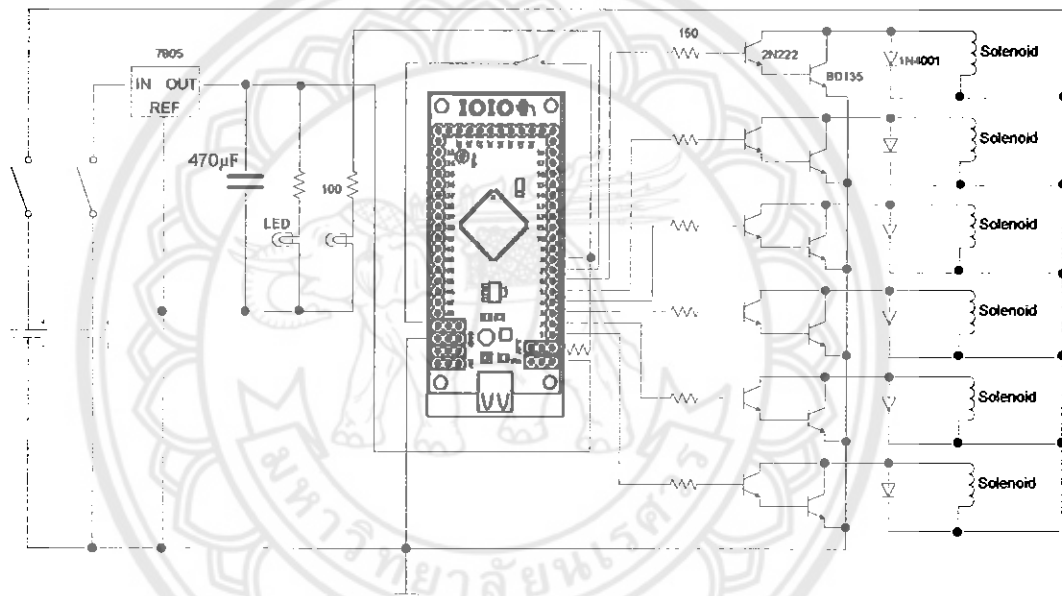
รูปที่ 4.9 วงจรทดสอบแอปพลิเคชัน

ทดสอบผลลัพธ์ของแอปพลิเคชันโดยสังเกตผ่านทางตำแหน่งหลอด LED ว่าได้ผลลัพธ์ตรงตามตำแหน่งตัวอักษรเบรลล์และหน้าที่หมายเลข PIN ที่กำหนดหรือไม่

4.4 การประกอบและทดลองอุปกรณ์แสดงผลข้อความสั้น

หลังจากทดสอบแอปพลิเคชันกับบอร์ดโยโย่และตรวจสอบผลการทำงานแล้ว การประกอบอุปกรณ์แสดงผลข้อความสั้นมีการเพิ่มเติมส่วนของสวิทช์ปิดเปิดและการแทนที่หลอด LED ด้วยโซลินอยด์เพื่อใช้แทนเข็มตัวอักษรเบรลล์ ซึ่งจะมีวงจรภายในอุปกรณ์แสดงผลดังรูปที่

4.10

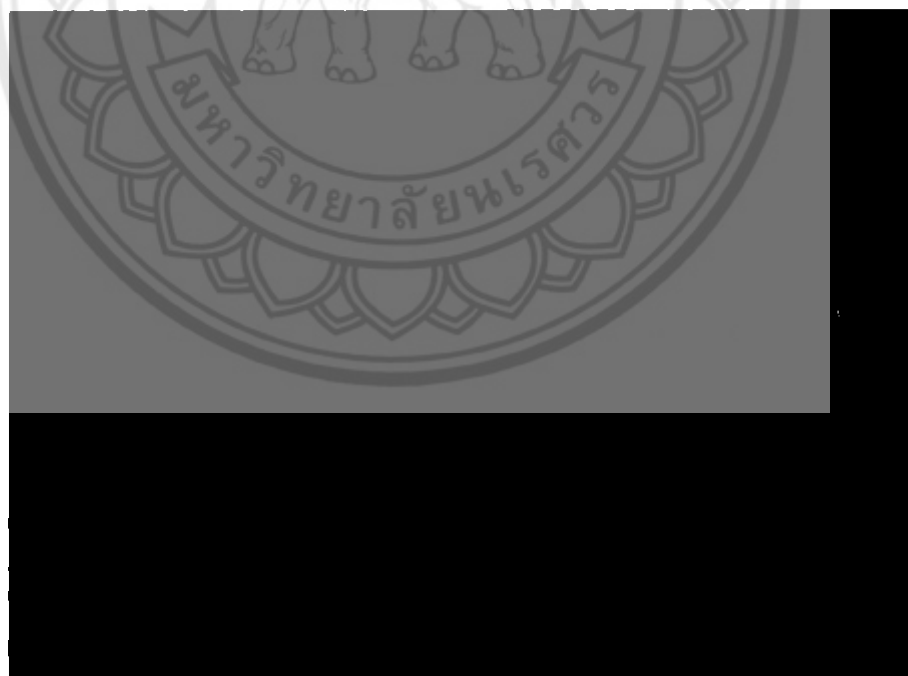


รูปที่ 4.10 วงจรภายในอุปกรณ์แสดงผลข้อความสั้น

หลังจาก ทำการประกอบวงจรภายในเสร็จสิ้น จะได้อุปกรณ์แสดงผลข้อความสั้นที่พร้อมใช้งานจริง โดยรูปที่ 4.11 และ 4.12 จะแสดงภาพภายในและภายนอกของอุปกรณ์แสดงผลข้อความสั้น



รูปที่ 4.11 ภายในอุปกรณ์แสดงผลข้อความสั้น



รูปที่ 4.12 ภายนอกอุปกรณ์แสดงผลข้อความสั้น

บทที่ 5

สรุปผลและข้อเสนอแนะ

บทนี้กล่าวถึงบทสรุป ปัญหาที่พบและข้อเสนอแนะของโครงการงาน “อุปกรณ์แสดงผลข้อความสั้นเป็นอักษรเบรลล์” (Refreshable Braille Display for SMS) เพื่อให้เกิดความเข้าใจและสามารถนำโครงการนี้ไปพัฒนาต่อไป

5.1 สรุปผลการทดลอง

โครงการนี้จัดทำขึ้นเพื่อนำเทคโนโลยีโทรศัพท์มือถือในปัจจุบันมาทำการประยุกต์เพื่อเสริมประสิทธิภาพในการใช้งานให้สามารถเข้าถึงกลุ่มผู้ใช้ที่เป็นกลุ่มคนพิการ เพื่อให้เกิดความหัดเทียมในการใช้เทคโนโลยีโทรศัพท์มือถือรับข้อความข่าวสารและสื่อต่างๆ โดยใช้โทรศัพท์ที่เป็นระบบปฏิบัติการ Android โดยเชื่อมต่อทางสาย USB สามารถแสดงผลอักษรเบรลล์จากข้อความสั้น ทีละ 1 ตัวอักษร เป็นภาษาอังกฤษ Aa-Zz และตัวเลข 0-9

5.2 ปัญหาที่พบและวิธีการแก้ไขปัญหา

จากการทดลองโครงการนี้ทำให้พบปัญหาใหญ่ๆ 2 ส่วนได้แก่การศึกษาและใช้งานไลบรารีต่างๆที่ใช้ในโครงการนี้ และขั้นตอนการทดลอง

ปัญหา : ปัญหาการใช้งานไลบรารีสั่งงานบอร์ดโฮโย่ ซึ่งเป็นเรื่องที่ค่อนข้างใหม่ ต้องใช้เวลาพอสมควรในการศึกษา เนื่องจากไม่เคยใช้ไลบรารีเหล่านี้มาก่อนและยังมีแหล่งข้อมูลให้ศึกษาค่อนข้างน้อย

การแก้ไข : การศึกษาหาข้อมูลในเรื่องที่ค่อนข้างใหม่อย่างบอร์ดโฮโย่ ถ้าข้อมูลที่ทำการศึกษาค้นคว้ามาได้มีน้อยหรือไม่เข้าใจในเนื้อหา สามารถหาข้อมูลเพิ่มเติมจากการตั้งคำถามผ่านเว็บไซต์ของผู้พัฒนาผลิตภัณฑ์บอร์ดโฮโย่ได้

ปัญหา : การทดลองบอร์ดโยโย่กับโทรศัพท์แอนดรอยด์นั้น แม้ว่าบอร์ดโยโย่ได้ถูกพัฒนาเพื่อใช้งานกับโทรศัพท์แอนดรอยด์โดยตรง แต่ในความเป็นจริงบอร์ดโยโย่ไม่สามารถเชื่อมต่อกับระบบแอนดรอยด์ได้ทุกเวอร์ชัน

การแก้ไข : ขั้นตอนการศึกษาข้อมูลจากเว็บไซต์ ควรดูส่วนการคอมเม้นจากผู้ใช้งานเพิ่มเติมด้วย สามารถใช้ Android OS ตั้งแต่ versions 1.5 เป็นต้นไป เพื่อเป็น Hardware Input / Output ของบอร์ดโยโย่

ปัญหา : การทดสอบแอปพลิเคชันที่พัฒนากับบอร์ดโยโย่ ไม่สามารถทำการทดสอบจากคอมพิวเตอร์โดยตรงได้ ต้องทำการทดสอบผ่านโทรศัพท์แอนดรอยด์เครื่องจริงเท่านั้น

การแก้ไข : ลงแอปพลิเคชันเสริม (KssAir) ในโทรศัพท์แอนดรอยด์ที่สามารถทำการเชื่อมต่อโทรศัพท์แอนดรอยด์กับคอมพิวเตอร์โดยไม่ต้องผ่านสาย USB เพื่อความสะดวกในการถ่ายโอนข้อมูลระหว่างโทรศัพท์แอนดรอยด์และคอมพิวเตอร์

ปัญหา : การจ่ายไฟภายในอุปกรณ์แสดงผลข้อความสั้นไม่เพียงพอ เนื่องจาก โซลีนอยด์ค่อนข้างใช้กระแสไฟฟ้าในการทำงานมาก

การแก้ไข : ใช้แหล่งจ่ายไฟอื่นในการป้อนไฟให้กับบอร์ดโยโย่ เพื่อป้องกันไม่ให้ โยโย่บอร์ดหยุดการทำงานชั่วคราวเนื่องจาก โซลีนอยด์กินกระแสมากเกินไป สังเกตได้จาก รูป 4.10 จะมีแหล่งจ่ายไฟ 2 แหล่งจ่าย

5.3 ข้อเสนอแนะ

จากโครงการอุปกรณ์แสดงผลข้อความสั้นเป็นอักษรเบรลล์นี้ ถ้ามีโอกาสถูกนำไปพัฒนาต่อยอดเพื่อให้สามารถนำไปใช้งานจริงได้ในอนาคต มีข้อเสนอแนะควรจัดหาอุปกรณ์ซึ่งมีขนาดเล็กเพื่อสะดวกในการใช้งานและพกพา อาจทำการออกแบบให้ตัวอุปกรณ์สามารถประกบยึดติดกับโทรศัพท์ได้ โดยให้อุปกรณ์ใช้แหล่งจากไฟจากตัวโทรศัพท์เอง ซึ่งจะสามารถช่วยให้อุปกรณ์แสดงผลข้อความสั้นเป็นอักษรเบรลล์นี้ ง่ายต่อการใช้งานและเป็นที่ยอมรับในอนาคต

เอกสารอ้างอิง

[1] “Braille Key”.(PBS Kids).สืบค้นเมื่อ 12 ตุลาคม 2554,จาก

http://pbskids.org/arthur/print/braille/braille_key.html

[2] “แอนดรอยด์ (ระบบปฏิบัติการ)”. (วิกิพีเดีย สารานุกรมเสรี).สืบค้นเมื่อ 20 ตุลาคม 2554,จาก

[http://th.wikipedia.org/wiki/แอนดรอยด์_\(ระบบปฏิบัติการ\)](http://th.wikipedia.org/wiki/แอนดรอยด์_(ระบบปฏิบัติการ))

[3] “IOIO for Android”.(Sparkfun ELECTRONICS).สืบค้นเมื่อ 17 ตุลาคม 2554,จาก

<https://www.sparkfun.com/products/10748>

[4] “Title IOIO”.(Cloudfront).สืบค้นเมื่อ 9 ธันวาคม 2554,จาก

<http://dlnmh9ip6v2uc.cloudfront.net/datasheets/Widgets/IOIO-v15.pdf>

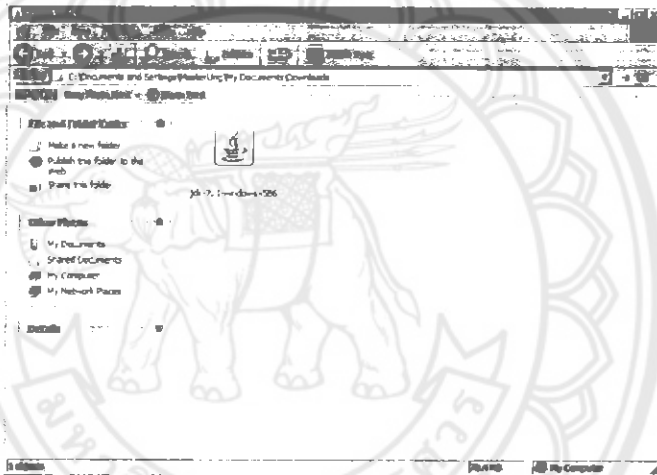
[5] “Tubular solenoid” .(Direct industry).สืบค้นเมื่อ 24 มกราคม 2555,จาก

<http://www.directindustry.com/prod/isliker-magnete-ag/tubular-solenoids-13153-378141.html>

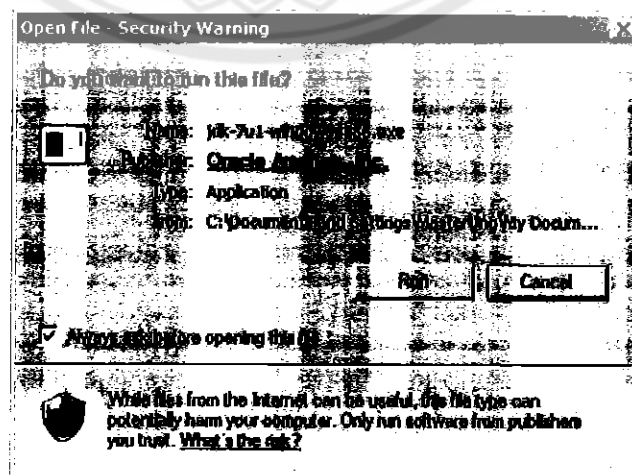
ภาคผนวก

การติดตั้งโปรแกรมสำหรับพัฒนาแอนดรอยด์

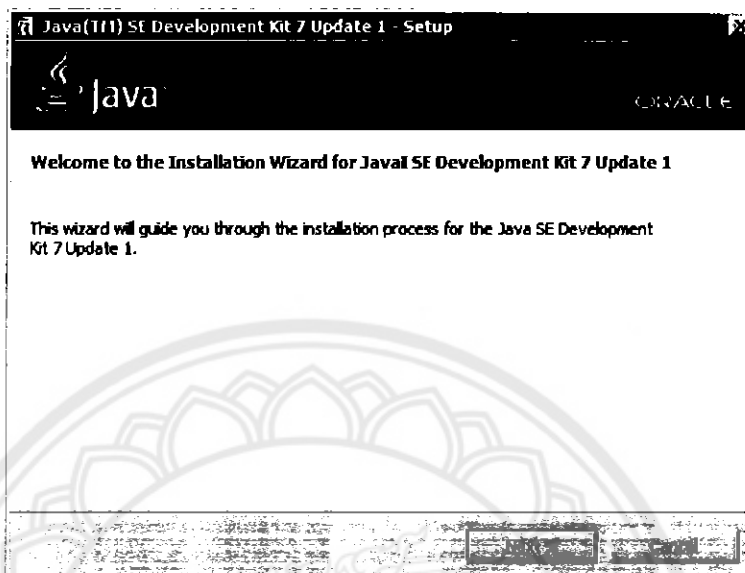
วิธีติดตั้ง JDK (Java Development Kit)



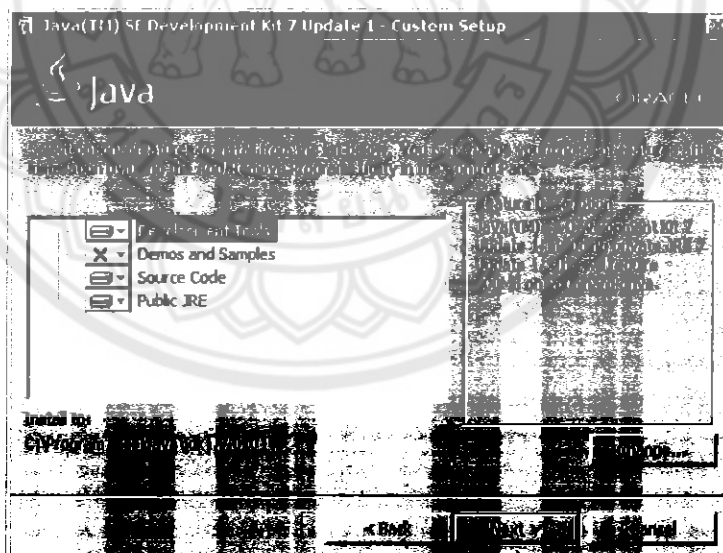
1. ทำการรันไฟล์ที่ดาวน์โหลดมา



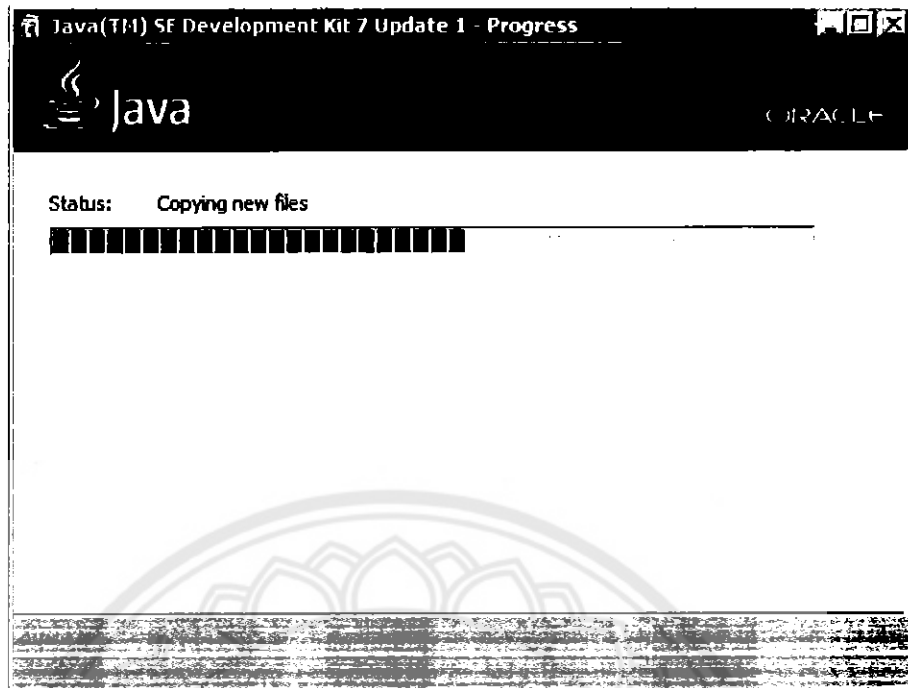
2. คลิก Run เพื่อเข้าสู่กระบวนการติดตั้ง



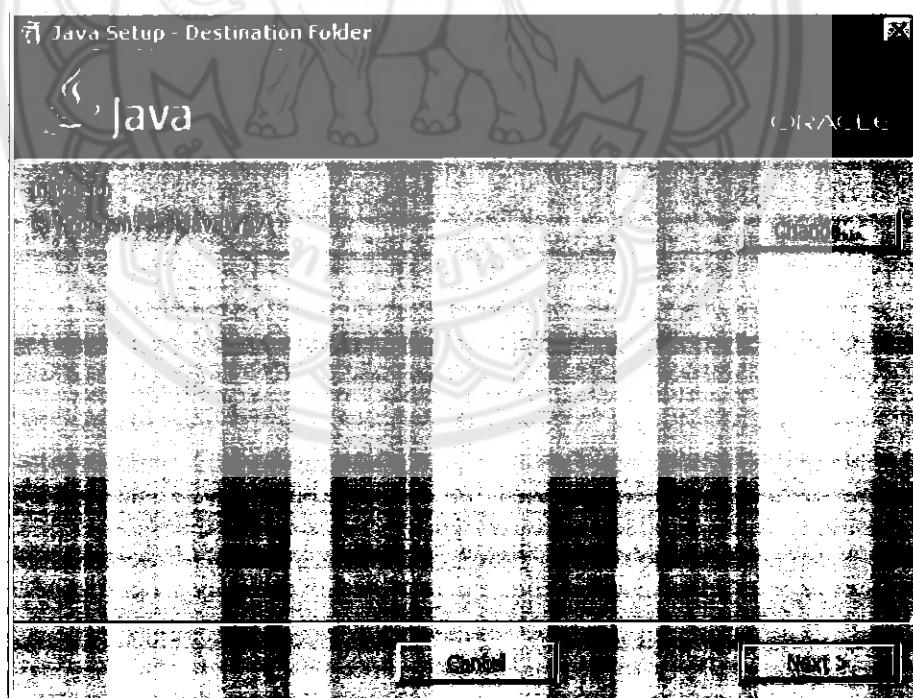
3. คลิก Next เพื่อเข้าสู่กระบวนการติดตั้งถัดไป



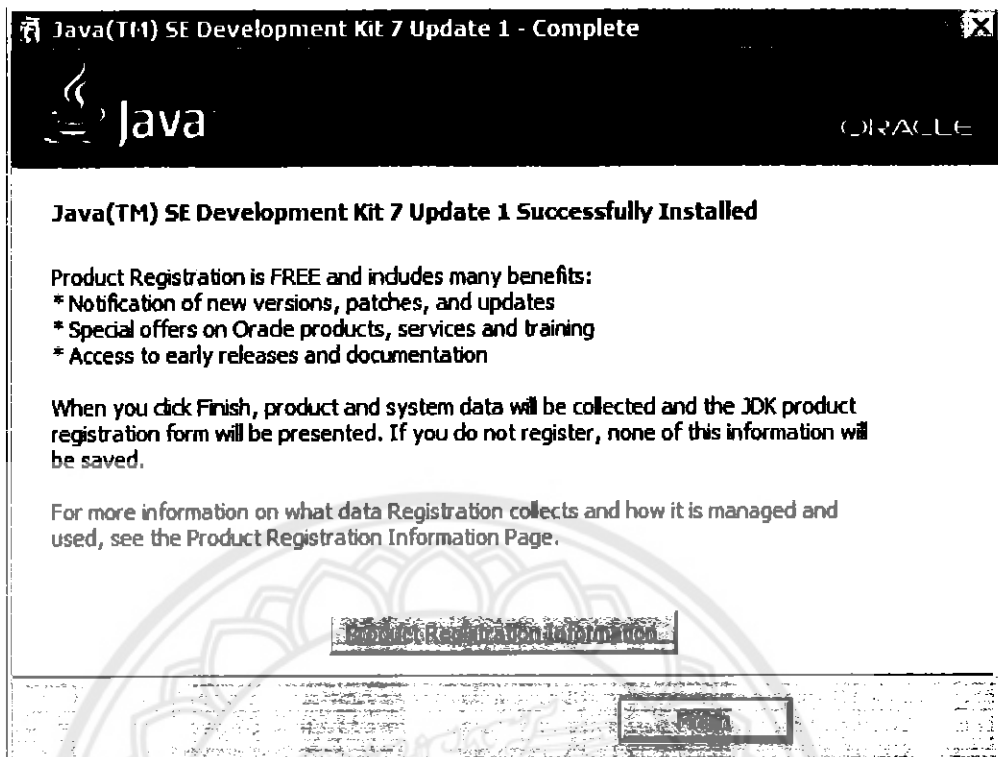
4. เลือกส่วนประกอบปลั๊กอินของโปรแกรม เมื่อเสร็จสิ้นแล้ว คลิก Next เพื่อเข้าสู่กระบวนการติดตั้งถัดไป



5. โปรแกรมจะเริ่มทำการคัดลอกข้อมูล เพื่อเตรียมความพร้อมในการติดตั้งโปรแกรม

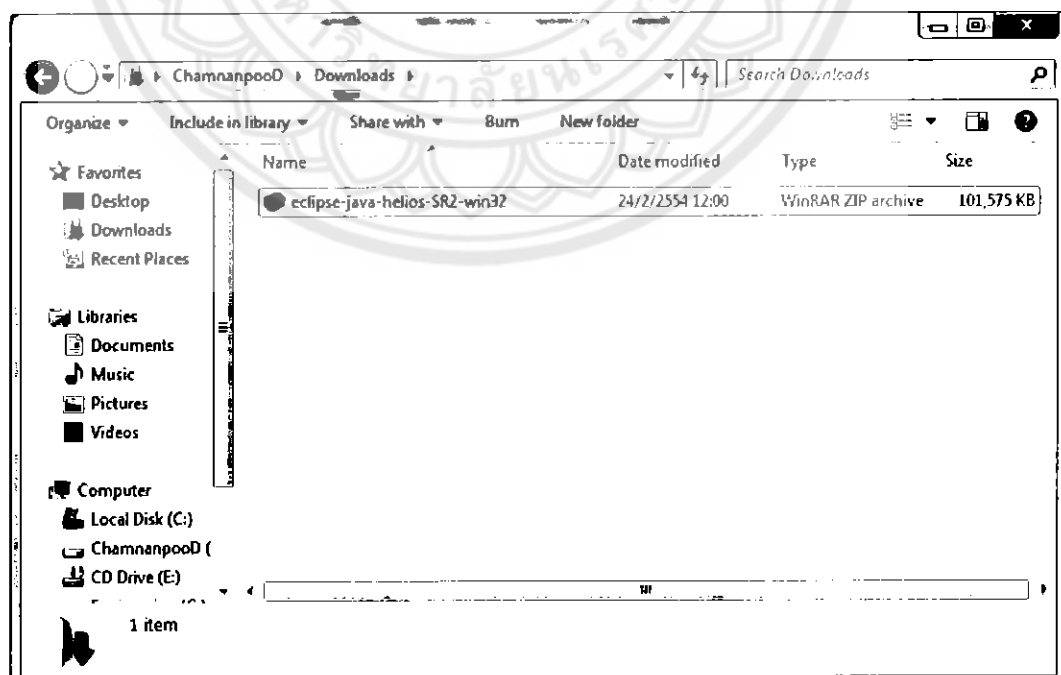


6. ทำการเลือกพื้นที่บนฮาร์ดดิสก์ ที่ต้องการติดตั้งโปรแกรมที่ปุ่ม Change เมื่อเสร็จสิ้นแล้วคลิก Next

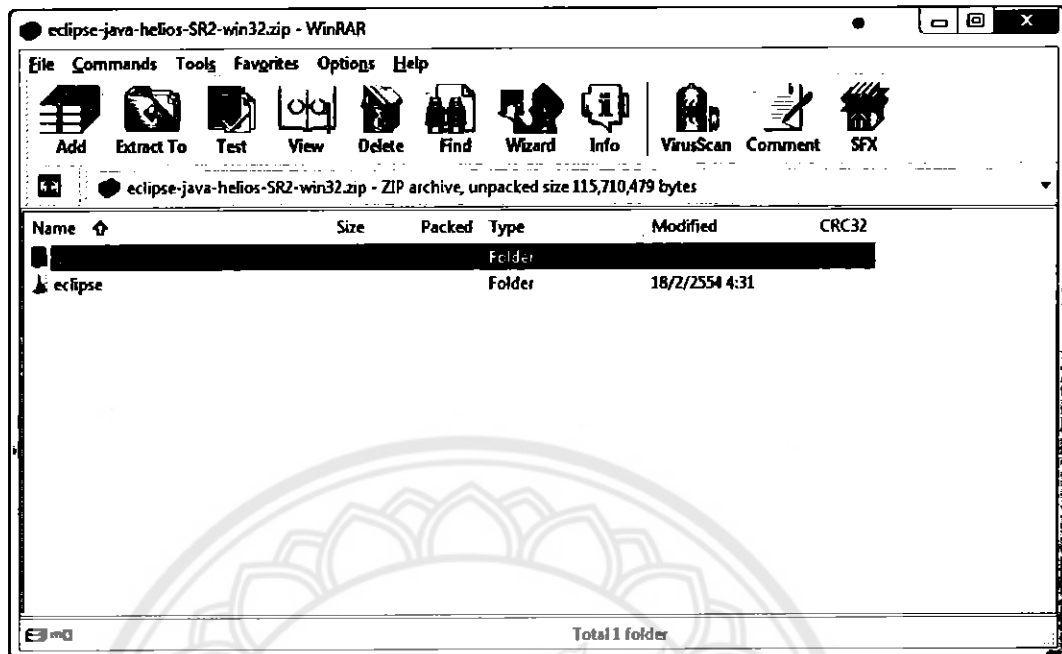


7. กด Finish เพื่อเสร็จสิ้นกระบวนการติดตั้ง

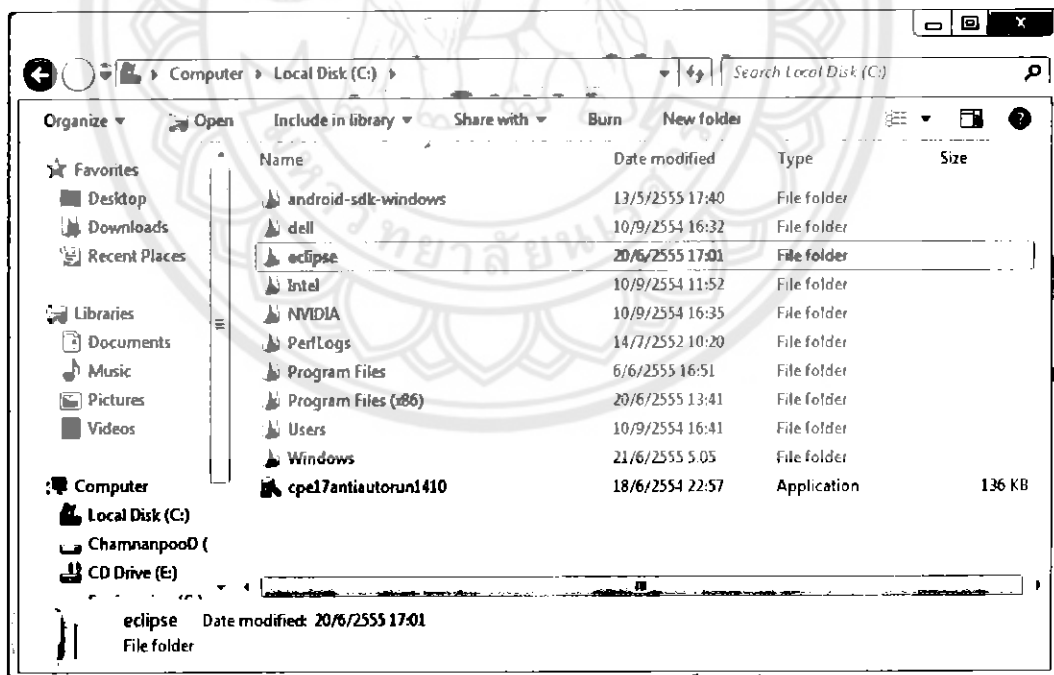
วิธีติดตั้ง Eclipse



1. เมื่อทำดาวโหลดเสร็จสิ้นแล้วจะได้ไฟล์ .zip

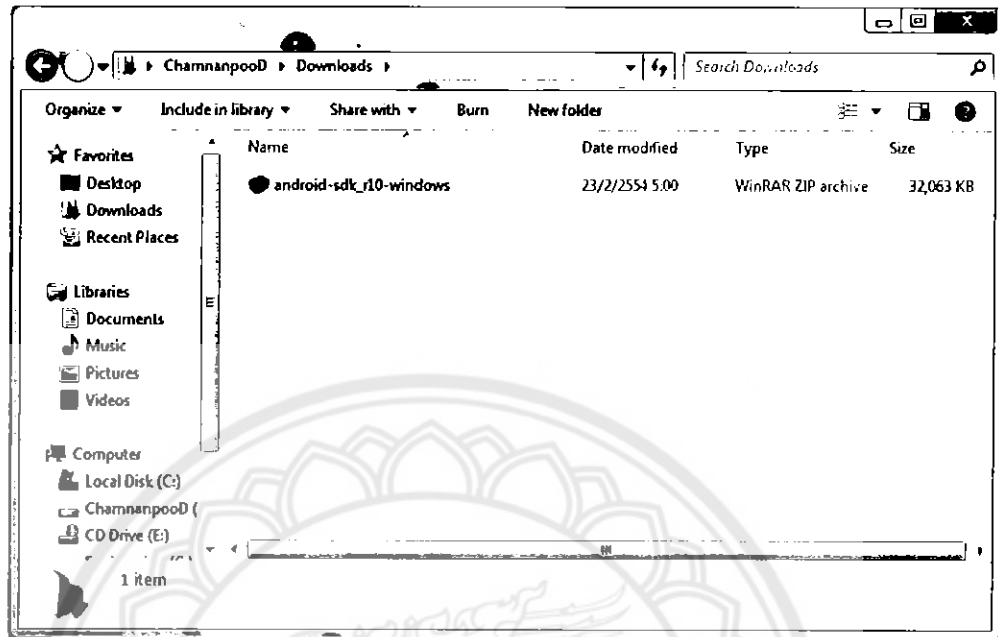


2. ทำการ Unzip ไฟล์เคอร์ eclipse ไว้ที่ไดเรกทอรี C:\eclipse

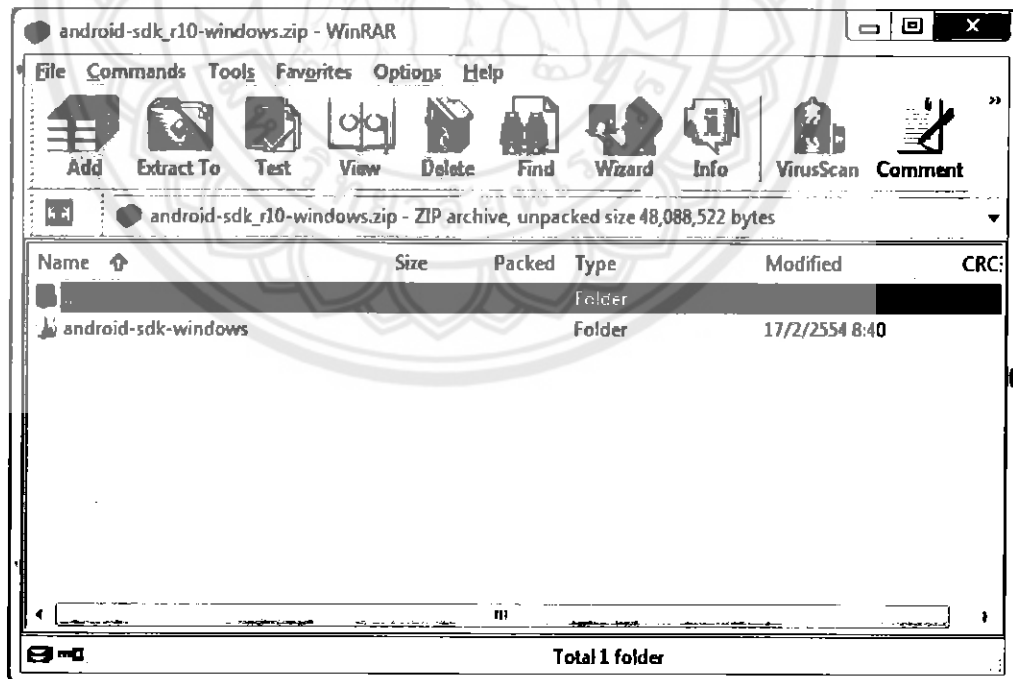


3. เสร็จสิ้นการติดตั้ง Eclipse

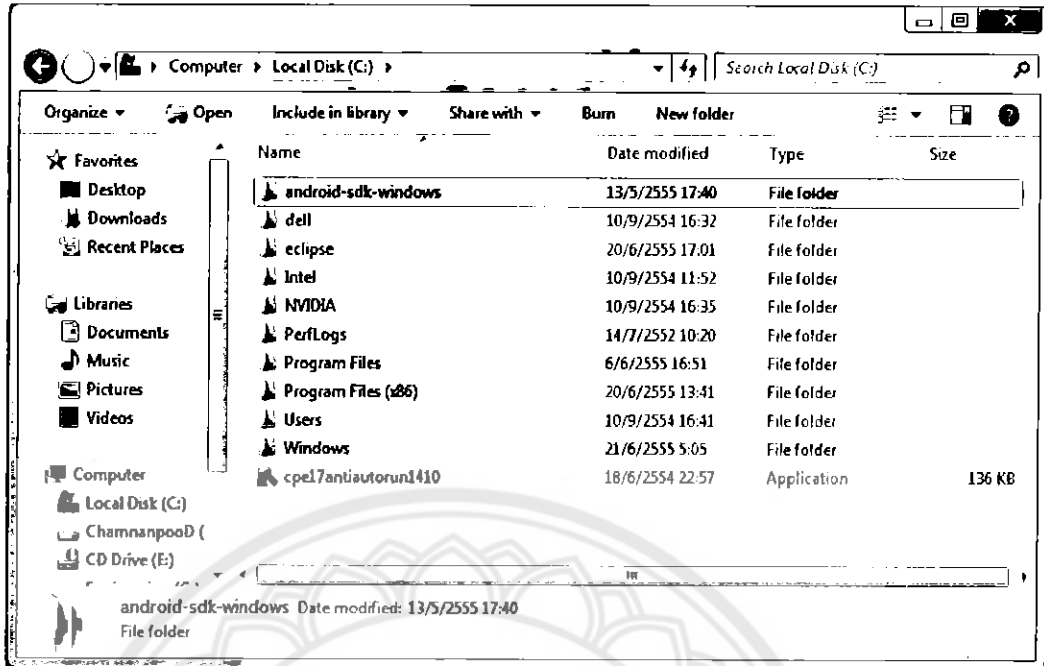
วิธีติดตั้ง Android SDK (Android Software Development Kit)



1. เมื่อทำดาวโหลดเสร็จสิ้นแล้วจะได้ไฟล์ .zip



2. ทำการ Unzip โฟลเดอร์ android-sdk-windows ไว้ที่ไดเรกทอรี C:\android-sdk-windows



3. เสร็จสิ้นการติดตั้ง

