ระบบตรวจจับอุณหภูมิและความชื้นผ่านแอปพลิเคชันแอนดรอยด์

TEMPERATURE AND HUMIDITY DETECTION SYSTEM VIA ANDROID APPLICATION

นางสาวบุษราภรณ์ ศรีคำนวณ

นางสาวพิชญา บัวประดิษฐ์

นางสาวศิวาพร ทองสุกดี

โครงงานวิศวกรรมนี้ เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร

วิศวกรรมศาสตรบัณฑิต

สาขาวิศวกรรมไฟฟ้า แขนงวิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยศรีนครินทรวิโรฒ

พ.ศ. 2557

ระบบตรวจจับอุณหภูมิและความชื้นผ่านแอปพลิเคชันแอนดรอยด์

TEMPERATURE AND HUMIDITY DETECTION SYSTEM VIA ANDROID APPLICATION

นางสาวบุษราภรณ์ ศรีคำนวณ

นางสาวพิชญา บัวประดิษฐ์

นางสาวศิวาพร ทองสุกดี

โครงงานวิศวกรรมนี้ เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร

วิศวกรรมศาสตรบัณฑิต

สาขาวิศวกรรมไฟฟ้า แขนงวิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยศรีนครินทรวิโรฒ

พ.ศ. 2557

หัวข้อโครงงานวิศวกรรม สาขาวิศวกรรมไฟฟ้า

เรื่อง ระบบตรวจจับอุณหภูมิและความชื้นผ่านแอปพลิเคชันแอนดรอยด์

โดย นางสาวบุษราภรณ์ ศรีคำนวณ

นางสาวพิชญา บัวประดิษฐ์

นางสาวศิวาพร ทองสุกดี

ภาควิชา วิศวกรรมไฟฟ้า

อาจารย์ที่ปรึกษา ผู้ช่วยศาสตราจารย์ศิริพงษ์ ฉายสินธ์

คณะวิศวกรรมศาสตร์มหาวิทยาลัยศรีนครินทรวิโรฒอนุมัติให้นับโครงงานวิศวกรรมไฟฟ้าเป็นส่วนหนึ่งของการศึกษา ตามหลักสูตรวิศวกรรมศาสตรบัณฑิต

..... คณบดีคณะวิศวกรรมศาสตร์

(รองศาสตราจารย์ ดร.เวคิน ปียรัตน์)

คณะกรรมการสอบโครงงานวิศวกรรม

..... ประธานกรรมการ

(อาจารย์ธานินทร์ ดวงจันทร์)

..... กรรมการ

(อาจารย์ ดร.พิชญา ชัยปัญญา)

..... กรรมการ

(อาจารย์สุนิศา คุณารักษ์)

..... กรรมการ

(ผู้ช่วยศาสตราจารย์ศิริพงษ์ ฉายสินธ์)

ระบบตรวจจับอุณหภูมิและความชื้นผ่านแอปพลิเคชันแอนดรอยด์
ปีการศึกษา 2557

โดย
นางสาวบุษราภรณ์ ศรีคำนวณ
นางสาวพิชญา บัวประดิษฐ์
นางสาวศิวาพร ทองสุกดี

อาจารย์ที่ปรึกษา
ผู้ช่วยศาสตราจารย์ศิริพงษ์ ฉายสินธ์

บทคัดย่อ

โครงงานวิศวกรรมศาสตร์ชิ้นนี้เป็นโครงงานที่เกี่ยวกับแอปพลิเคชันที่ใช้ตรวจสอบอุณหภูมิและความชื้น โดยมีการส่งข้อมูลผ่านระบบเครือข่ายไร้สายแบบบลูทูธ โดยโครงงานนี้ประกอบไปด้วยตัวตรวจวัดอุณหภูมิ(LM35DZ) และตัวตรวจวัดความชื้นสัมพัทธ์ (HIH4030) จะส่งค่าแรงดันที่แปรเปลี่ยนไปตามอุณหภูมิและความชื้นที่ตรวจวัดได้ให้กับบอร์ด IOIO-Q โดยบอร์ด IOIO-Q จะส่งข้อมูลโดยผ่านระบบเครือข่ายไร้สายแบบบลูทูธไปยังสมาร์ตโฟนระบบปฏิบัติการแอนดรอยด์ที่ได้เขียนแอปพลิเคชันรองรับไว้แล้ว และแอปพลิเคชันนี้จะแจ้งเตือนถ้าหากอุณหภูมิและความชื้นที่ได้รับมีค่ามากหรือน้อยกว่าเงื่อนไขของแอปพลิเคชันที่ผู้ใช้ได้กำหนดไว้

คำสำคัญ: ตัวตรวจวัดอุณหภูมิ(LM35DZ)/ ตัวตรวจวัดความชื้นสัมพัทธ์(HIH4030)/ บอร์ด IOIO-Q/ ระบบเครือข่ายไร้สายแบบบลูทูธ

Temperature And Humidity Detection System Via Android Application

Academic Year 2014

By

Ms. Budsaraporn Srikoumnuaun

Ms. Pichaya Buapradit

Ms. Siwaporn Thongsukdee

Project Advisor

Asst. Prof. Siripong Chaysin

ABSTRACT

This engineering projects is intended to apply using application that used for detect temperature and humidity with data transmission by wireless Bluetooth. In this project, consists of temperature sensor (LM35DZ) and humidity sensors (HIH4030). So we send voltage to vary with temperature sensors and humidity sensors to IOIO-Q board. The IOIO-Q board will sends data via wireless Bluetooth to application that written in the operating android system. This application will alert if the temperature and humidity have been more or less than the value of the application the user is defined.

Keywords: Temperature sensor (LM35DZ)/ Humidity sensors (HIH4030)/ Board IOIO-Q/ Networking Wireless Bluetooth

กิตติกรรมประกาศ

โครงงานนี้จะสำเร็จลุล่วงไม่ได้ถ้าไม่ได้รับการช่วยเหลือจากอาจารย์ที่ปรึกษาโครงงาน ผู้ช่วยศาสตราจารย์ศิริพงษ์ ฉายสินธ์ ที่ช่วยให้คำปรึกษาและหาแนวทางในการแก้ไขปัญหาต่าง ๆ รวมถึงการตรวจสอบความถูกต้องเรียบร้อยของโครงงานจนทำให้โครงงานวิศวกรรมนี้มีความสมบูรณ์และถูกต้อง ตลอดจนขอขอบพระคุณอาจารย์ทุกท่านรวมถึงบุคลากรในภาควิชาวิศวกรรมไฟฟ้ามหาวิทยาลัยศรีนครินทรวิโรฒ ที่ได้ให้ความรู้ให้คำปรึกษาและอำนวยความสะดวกต่าง ๆ ตลอดจนบิดามารดาและผู้มีพระคุณที่ได้ให้การอบรมสั่งสอนและช่วยเหลือในด้านต่าง ๆ ทำให้ทางคณะผู้จัดทำมีกำลังใจและสามารถทำโครงงานฉบับนี้ให้สำเร็จลุล่วงได้ด้วยดี

สารบัญ

	หน้าที่
บทคัดย่อภาษาไทย	ก
บทคัดย่อภาษาอังกฤษ	ข
กิตติกรรมประกาศ	ค
สารบัญ	ง
สารบัญตาราง	ฉ
สารบัญภาพ	ช
รายการสัญลักษณ์	ฎ
ประมวลศัพท์และคำย่อ	ฐ
บทที่ 1 บทนำ	
1.1 ความเป็นมาของโครงงาน	1
1.2 วัตถุประสงค์ของโครงงาน	2
1.3 ขอบเขตโครงงาน	2
1.4 ประโยชน์ที่จะได้รับ	2
บทที่ 2 ทฤษฎีที่เกี่ยวข้อง	
2.1 บอร์ด IOIO-Q	3
2.2 คุณสมบัติทางเทคนิคของบอร์ด IOIO-Q	4
2.3 วงจรของ IOIO-Q	5
2.4 IOIO Activity board	9
2.5 ความแตกต่างระหว่างบอร์ด IOIO กับ บอร์ดไมโครคอนโทรลเลอร์ทั่วไป	10
2.6 การเชื่อมต่อระหว่างบอร์ด IOIO กับอุปกรณ์แอนดรอยด์	12
2.7 การใช้งานพอร์ตอินพุตเอาต์พุตดิจิทัลของ IOIO-Q	12
2.8 IOIO-Q กับการอ่านค่าสัญญาณแอนะล็อก	14
2.9 ตัวตรวจจับความชื้นสัมพัทธ์ HIH4030	16
2.10 ไอซีวัดอุณหภูมิ LM35DZ	18

สารบัญ(ต่อ)

	หน้าที่
2.11 การติดตั้งเครื่องมือพัฒนาแอปพลิเคชันสำหรับอุปกรณ์แอนดรอยด์	19
2.12 ทดสอบการติดตั้งระหว่างอุปกรณ์แอนดรอยด์และบอร์ด IOIO-Q	27
บทที่ 3 หลักการออกแบบ	
3.1 บล็อกไดอะแกรมของระบบตรวจจับอุณหภูมิและความชื้นแบบไร้สาย	37
3.2 การออกแบบระบบตรวจจับอุณหภูมิและความชื้น	38
3.3 การออกแบบการทำงานของแอปพลิเคชัน	43
3.4 แผนผังการทำงานของเซนเซอร์วัดอุณหภูมิและความชื้น	55
3.5 แผนผังการทำงานในส่วนของบอร์ด IOIO-Q กับแอปพลิเคชัน	56
3.6 แผนผังการทำงานในส่วนของแอปพลิเคชัน	57
บทที่ 4 ผลการทดลอง	
4.1 การตั้งค่าและผลการทดลอง	59
4.2 การกำหนดค่าของการรับและส่งข้อมูล	62
4.3 ตารางสรุปผลการทดลอง	68
4.4 ค่าความแม่นยำของเครื่องมือวัด (Accuracy: A)	81
4.5 ค่าความเที่ยงตรงของเครื่องมือวัด (Precision)	85
4.6 ค่าน้อยที่สุดและมากที่สุดในการวัดค่าอุณหภูมิและความชื้นสัมพัทธ์	88
บทที่ 5 สรุปผลและข้อเสนอแนะ	
5.1 สรุปผลของโครงงาน	89
5.2 อุปสรรคและปัญหาของโครงงาน	90
5.3 ข้อเสนอแนะ	90
เอกสารอ้างอิง	91
ภาคผนวก	92
ประวัตินิสิตผู้ทำโครงงาน	110

สารบัญตาราง

ตารางที่	หน้า
2.1 หน้าที่ของขาต่าง ๆ บนบอร์ด IOIO-Q	6
4.1 อุณหภูมิที่วัดได้จาก Thermo-Hygrometer เทียบกับชุดตรวจจับที่อุณหภูมิห้อง	70
4.2 ความชื้นที่วัดได้จากThermo-Hygrometer เทียบกับชุดตรวจจับที่อุณหภูมิห้อง	71
4.3 อุณหภูมิที่วัดได้จาก Thermo-Hygrometer เทียบกับชุดตรวจจับที่อุณหภูมิต่ำ	73
4.4 ความชื้นที่วัดได้จาก Thermo-Hygrometer เทียบกับชุดตรวจจับที่อุณหภูมิต่ำ	74
4.5 อุณหภูมิที่วัดได้จาก Thermo-Hygrometer เทียบกับชุดตรวจจับที่อุณหภูมิสูง	76
4.6 ความชื้นที่วัดได้จาก Thermo-Hygrometer เทียบกับชุดตรวจจับที่อุณหภูมิสูง	77
4.7 อุณหภูมิที่วัดได้จาก Thermo-Hygrometer เทียบกับชุดตรวจจับที่ระยะทางบลูทูธ	79
4.8 ความชื้นที่วัดได้จาก Thermo-Hygrometer เทียบกับชุดตรวจจับที่ระยะทางบลูทูธ	80
4.9 ค่าความแม่นยำของการวัดอุณหภูมิจากThermo-Hygrometer เทียบกับชุดตรวจจับ	82
4.10 ค่าความแม่นยำของการวัดความชื้นจากThermo-Hygrometer เทียบกับชุดตรวจจับ	83
4.11 ค่าความเที่ยงตรงของการวัดอุณหภูมิจากThermo-Hygrometer เทียบกับชุดตรวจจับ	86
4.12 ค่าความเที่ยงตรงของการวัดความชื้นจาก Thermo-Hygrometer เทียบกับชุดตรวจจับ	87
4.13 ค่าน้อยที่สุดและมากที่สุดที่วัดได้จากThermo-Hygrometer และเครื่องตรวจจับ	88

สารบัญรูป

รูปที่	หน้า
2.1 โครงสร้างบอร์ด IOIO ต้นฉบับพัฒนาโดย YTAI	3
2.2 โครงสร้างบอร์ด IOIO-Q ที่ผลิตโดย INEX	4
2.3 ส่วนประกอบที่สำคัญของบอร์ด IOIO-Q	5
2.4 IOIO Activity board	9
2.5 IOIO Activity board ที่ติดตั้งบอร์ด IOIO-Q และบลูทูธดองเกิลพร้อมใช้งาน	10
2.6 การเชื่อมต่อระหว่างบอร์ดIOIO กับแอนดรอยด์	10
2.7 การเชื่อมต่อระหว่างบอร์ดไมโครคอนโทรลเลอร์ทั่วไปกับแอนดรอยด์	11
2.8 การเชื่อมต่อแบบ Bluetooth	12
2.9 การตีความลอจิกจากระดับสัญญาณไฟตรงของบอร์ด IOIO-Q	13
2.10 การเชื่อมต่อ IOIO-Q กับอุปกรณ์ภายนอกที่ใช้ไฟเลี้ยง +5V	13
2.11 การสุ่มสัญญาณแอนะล็อกเพื่อกำหนดข้อมูลดิจิทัล	14
2.12 ระบุตำแหน่งขาอินพุตแอนะล็อกของบอร์ด IOIO-Q	15
2.13 โมดูลวัดความชื้นสัมพัทธ์ HIH4030	16
2.14 โมดูลแผงวงจร HIH4030 ที่พร้อมใช้งาน	17
2.15 ไอซีวัดอุณหภูมิ LM35DZ	18
2.16 ข้อมูลพื้นฐานของไอซีวัดอุณหภูมิ LM35DZ	18
2.17 เว็บไซต์ Oracle ที่ใช้ในการดาวน์โหลด	20
2.18 ขั้นตอนการดาวน์โหลด Java Development Kit	21
2.19 หน้าต่างสำหรับเริ่มติดตั้งโปรแกรม Java Development Kit	21
2.20 การดาวน์โหลด Eclipse IDE	22
2.21 ดาวน์โหลดและติดตั้ง Android SDK	23
2.22 ดาวน์โหลดเวอร์ชันที่ต้องการพัฒนา	23
2.23 ขั้นตอนแรกในการติดตั้ง Plugin เสริมใน Eclipse	23

สารบัญรูป (ต่อ)

รูปที่	หน้า
2.24 การใส่ค่าต่าง ๆ ในโปรแกรม Eclipse	24
2.25 การติดตั้งคลิก Next ไปเรื่อย ๆ จนกว่าจะติดตั้งสำเร็จ	24
2.26 หน้าต่าง Installing Software	25
2.27 หน้าต่าง Java-Eclipse SDK	25
2.28 การตั้งชื่อ AVD และตั้งค่า Emulator	26
2.29 การลองทดสอบ Emulator ที่สร้างขึ้นมา	26
2.30 หน้าต่างแสดงการรัน Emulator โทรศัพท์ Android	27
2.31 ตัวอย่างการเปิดใช้งาน USB Debugging บนแอนดรอยด์เวอร์ชันต่ำกว่า 4.2.1	28
2.32 ตัวอย่างการเปิดเมนู Developer options บนแอนดรอยด์เวอร์ชัน 4.2.1 ขึ้นไป	29
2.33 การเปิดใช้งาน USB Debugging บนแอนดรอยด์เวอร์ชัน 4.2.1 ขึ้นไป	29
2.34 บอร์ด IOIO-Q เมื่อติดตั้งลงบนบอร์ด IOIO-Q Activity	30
2.35 Notification แจ้งการใช้ USB Debugging บนอุปกรณ์แอนดรอยด์	31
2.36 ต่อบลูทูธดองเกิลเข้ากับบอร์ด IOIO-Q ที่ติดตั้งบน IOIO Activity	31
2.37 การค้นหาสัญญาณบลูทูธดองเกิล	32
2.38 การใส่เลขรหัสของการจับคู่สัญญาณกับบลูทูธดองเกิล	33
2.39 อุปกรณ์แอนดรอยด์ปรากฏข้อความแสดงการเชื่อมต่อได้	33
2.40 การแก้ปัญหาแอนดรอยด์ค้นหาบอร์ด IOIO-Q ที่ต่อกับบลูทูธดองเกิลไม่พบ	34
2.41 แผนภาพแสดงขั้นตอนการพัฒนาแอปพลิเคชัน สำหรับอุปกรณ์แอนดรอยด์	35
3.1 บล็อกไดอะแกรมภาคส่ง	37
3.2 บล็อกไดอะแกรมภาครับ	38
3.3 ไอซีวัดอุณหภูมิ LM35DZ	39
3.4 โมดูลแผงวงจร HIH4030 ที่พร้อมใช้งาน	39
3.5 บอร์ด IOIO-Q	40
3.6 การระบุตำแหน่งขาอินพุตแอนะล็อกของบอร์ด IOIO-Q	40

สารบัญรูป (ต่อ)

รูปที่	หน้า
3.7 การเชื่อมต่อระหว่างไอซีLM35DZ และโมดูลHIH4030 เข้ากับบอร์ดIOIO-Q	41
3.8 การเชื่อมต่ออุปกรณ์ในการต่อใช้งานกับแอปพลิเคชัน	41
3.9 การเชื่อมต่อไอซีLM35DZ เข้ากับบอร์ด IOIO-Q	42
3.10 การเชื่อมต่อโมดูลHIH4030 เข้ากับบอร์ด IOIO-Q	43
3.11 การดาวน์โหลดไฟล์ไลบรารีสำหรับบอร์ด IOIO-Q	44
3.12 ขั้นตอนการเตรียมไฟล์โปรเจกต์	44
3.13 วิธีการสร้างโปรเจกต์พื้นฐานของแอปพลิเคชัน	45
3.14 วิธีการเลือกไฟล์ไลบรารีของบอร์ด IOIO ให้กับโปรเจกต์	46
3.15 วิธีการเพิ่มไฟล์ไลบรารีของบอร์ด IOIO ให้กับโปรเจ็กต์	46
3.16 วิธีการสร้างแพ็กเกจของแอปพลิเคชัน	47
3.17 วิธีการตั้งชื่อเพื่อจัดเก็บแพ็กเกจ Java	47
3.18 วิธีการสร้างโปรเจกต์สำหรับจัดเก็บไฟล์ Java	48
3.19 วิธีการตั้งชื่อเพื่อจัดเก็บไฟล์ Java	48
3.20 วิธีการกำหนดฟังก์ชันการทำงานของแอปพลิเคชัน	49
3.21 หน้าต่าง main.xml	50
3.22 การเขียนแบบแสดงภาพกราฟิก (Graphical Layout)	51
3.23 การเขียนแบบคำสั่ง XML	51
3.24 ส่วนย่อยต่าง ๆ ที่นำมาออกแบบแอปพลิเคชัน	52
3.25 ตัวอย่างรูปภาพที่อยู่ใน drawable	52
3.26 การแก้ไขไฟล์ AndroidMainfest.xml	53
3.27 การกำหนดขีดความสามารถของแอปพลิเคชัน	53
3.28 การแก้ไขไฟล์ string.xml	54
3.29 การอ้างอิงรายละเอียดต่าง ๆ ของลักษณะประจำข้อมูล	54
3.30 แผนผังการทำงานของเซนเซอร์วัดอุณหภูมิและความชื้น	55
3.31 แผนผังการทำงานในส่วนของบอร์ด IOIO-Qกับแอปพลิเคชัน	56

สารบัญรูป (ต่อ)

รูปที่	หน้า
3.32 แผนผังการทำงานของแอปพลิเคชันในส่วนที่ 1	57
3.33 แผนผังการทำงานของแอปพลิเคชันในส่วนที่ 2	58
4.1 รูปแอปพลิเคชัน Hello IOIO	59
4.2 แอปพลิเคชันที่ใช้ในการทดลองขณะไม่มีการส่งค่าอุณหภูมิและความชื้นสัมพัทธ์	60
4.3 ส่วนหน้าจอหลักของแอปพลิเคชัน	60
4.4 ส่วนของการกำหนดค่าต่ำที่สุดและสูงที่สุด	61
4.5 หน้าต่างของแอปพลิเคชันเมื่อเสร็จสิ้นการกรอกค่าต่ำที่สุดและสูงที่สุด	62
4.6 ต่อบลูทูธดองเกิลเข้ากับบอร์ด IOIO-Q ที่ติดตั้งบนบอร์ด IOIO Activity	62
4.7 การค้นหาสัญญาณจากบลูทูธดองเกิล	63
4.8 การจับคู่สัญญาณกับบลูทูธดองเกิล	64
4.9 สถานะการเชื่อมต่อสมาร์ตโฟน กับ IOIO-Q แบบไร้สายผ่านบลูทูธ	64
4.10 การกำหนดค่าสูงสุดและต่ำสุดของอุณหภูมิและความชื้น	65
4.11 แอปพลิเคชันแสดงแสดงสถานะ Alert ที่ค่าของอุณหภูมิ	66
4.12 แสดงการกำหนดค่าสูงสุดและต่ำสุดของอุณหภูมิและความชื้น	66
4.13 แอปพลิเคชันแสดงสถานะ Alert ที่ค่าของความชื้น	67
4.14 แสดงการกำหนดค่าสูงสุดและต่ำสุดของอุณหภูมิและความชื้น	67
4.15 แอปพลิเคชันแสดงสถานะ Alert ที่ค่าของอุณหภูมิและความชื้น	68
4.16 การทดลองที่อุณหภูมิห้อง	69
4.17 หน้าจอแอปพลิเคชันแสดงการทำงานในขณะที่ทำการวัดอุณหภูมิห้อง	69
4.18 กราฟอุณหภูมิที่วัดได้จาก Thermo-Hygrometer เทียบกับชุดตรวจจับที่อุณหภูมิห้อง	70
4.19 กราฟความชื้นที่วัดได้จาก Thermo-Hygrometer เทียบกับชุดตรวจจับที่อุณหภูมิห้อง	71
4.20 การทดลองที่อุณหภูมิต่ำ	72
4.21 หน้าจอแอปพลิเคชันแสดงการทำงานในขณะที่ทำการวัดอุณหภูมิต่ำ	72
4.22 กราฟอุณหภูมิที่วัดได้จาก Thermo-Hygrometer เทียบกับชุดตรวจจับที่อุณหภูมิต่ำ	73
4.23 กราฟความชื้นที่วัดได้จาก Thermo-Hygrometer เทียบกับชุดตรวจจับที่อุณหภูมิต่ำ	74

สารบัญรูป (ต่อ)

รูปที่	หน้า
4.24 การทดลองที่อุณหภูมิสูง	75
4.25 หน้าจอแอปพลิเคชันแสดงการทำงานในขณะที่ทำการวัดอุณหภูมิสูง	75
4.26 กราฟอุณหภูมิที่วัดได้จาก Thermo-Hygrometer เทียบกับชุดตรวจจับที่อุณหภูมิสูง	76
4.27 กราฟความชื้นที่วัดได้จาก Thermo-Hygrometer เทียบกับชุดตรวจจับที่อุณหภูมิสูง	77
4.28 หน้าจอแอปพลิเคชันแสดงการทำงานในขณะที่ทำการวัดระยะทางของสัญญาณบลูทูธ	78
4.29 กราฟอุณหภูมิที่วัดได้จาก Thermo-Hygrometer เทียบกับชุดตรวจจับที่ระยะทางบลูทูธ	79
4.30 กราฟความชื้นที่วัดได้จาก Thermo-Hygrometer เทียบกับชุดตรวจจับที่ระยะทางบลูทูธ	80
4.31 การทดลองที่อุณหภูมิห้องเพื่อหาค่าความแม่นยำ	81
4.32 หน้าจอแอปพลิเคชันในขณะที่ทำการวัดอุณหภูมิห้องเพื่อหาค่าความแม่นยำ	82
4.33 กราฟความแม่นยำของอุณหภูมิที่วัดจากThermo-Hygrometerเทียบกับชุดตรวจจับ	83
4.34 กราฟความแม่นยำของความชื้นที่วัดจากThermo-Hygrometerเทียบกับชุดตรวจจับ	84
4.35 การทดลองที่อุณหภูมิห้องเพื่อหาค่าความเที่ยงตรง	85
4.36 หน้าจอแอปพลิเคชันในขณะทำการวัดที่อุณหภูมิห้องเพื่อหาค่าความเที่ยงตรง	85
4.37 กราฟความเที่ยงตรงของอุณหภูมิที่วัดจากThermo-Hygrometer เทียบกับชุดตรวจจับ	86
4.38 กราฟความเที่ยงตรงของความชื้นที่วัดจากThermo-Hygrometer เทียบกับชุดตรวจจับ	87

รายการสัญลักษณ์

Error	**ค่าความคลาดเคลื่อน**	**%**
Accuracy	**ค่าความแม่นยำของเครื่องมือวัด**	**%**
Precision	**ค่าความเที่ยงตรง**	**%**
X_{mea}	**ค่าที่ได้จากการวัด (measure value)**	**%RH, °C**
X_t	**ค่าจริง (true value)**	**%RH,°C**
X_i	**ค่าการวัดแต่ละครั้ง**	**%RH, °C**
X_m	**ค่าเฉลี่ยของการวัด**	**%RH, °C**
n	**จำนวนครั้งของการวัด**	**ครั้ง**
V_{out}	**แรงดันเอาต์พุต**	**V**
V_s	**แรงดันจากขาต่อไฟเลี้ยง**	**V**

ประมวลศัพท์และคำย่อ

V	=	Volts
PWM	=	Pulse-width Modulation
A	=	Ampere
mA	=	Milli-amps
ADB	=	Android Debug Bridge
kΩ	=	Kilo-ohms
Ω	=	Ohms
ADB	=	Android Debug Bridge
AOA	=	Android Open Accessory
ADC	=	Analog to Digital Converter
LED	=	Light-emitting diode
%RH	=	Relative Humidity
µA	=	Micro-ampere (microamps)
GND	=	Ground
mV/°C	=	Millivolt per celsius
PDE	=	Plug-in Development Environment
JDK	=	Java Development Kit
Java VM	=	Java Virtual Machine
Android SDK	=	Android Software Development Kit
AOSP	=	Android Opensource Project
OHM	=	Open Handset Mobile
OHA	=	Open Handset Alliances
GMS	=	Google Mobile Service

บทที่ 1
บทนำ

1.1 ความเป็นมาของโครงงาน

ในปัจจุบันสภาพอากาศมีความไม่แน่นอน ทำให้อุณหภูมิและความชื้นในอากาศมีค่าไม่คงที่การเปลี่ยนแปลงของอุณหภูมิและความชื้นในอากาศจึงเกิดการเปลี่ยนแปลงอยู่ตลอดเวลา การควบคุมจึงเป็นเรื่องสำคัญอย่างหนึ่ง แต่สิ่งสำคัญยิ่งกว่าคือการทราบค่าการเปลี่ยนแปลงอุณหภูมิและความชื้น ดังนั้นการเฝ้าระวังจึงถือเป็นทางเลือกอีกทางหนึ่ง เพราะฉะนั้นอุปกรณ์วัดอุณหภูมิและความชื้นจะต้องมีการรายงานผลที่เที่ยงตรงและสม่ำเสมอ เพื่อที่จะทำให้เราทราบค่าการเปลี่ยนแปลง เมื่อเกิดปัญหาก็จะสามารถแก้ปัญหาได้ทันท่วงที

เมื่อก่อนการส่งข้อมูลระหว่างอุปกรณ์วัดอุณหภูมิและความชื้นกับคอมพิวเตอร์นั้นจะต้องส่งผ่านสายสัญญาณทำให้เราลำบากในด้านของการลากโยงสายระหว่างกันและจะเกิดปัญหามากขึ้นเมื่อสายไม่สามารถเข้าถึงในสถานที่ที่ยากต่อการเข้าถึงได้หรือมีสิ่งกีดขวางที่ลำบากต่อการโยงสาย อีกทั้งยังทำให้เกิดความสิ้นเปลืองของสายสัญญาณอีกด้วย อีกปัญหาหนึ่งที่เราควรคำนึงถึงคือการเคลื่อนย้ายอุปกรณ์วัดอุณหภูมิและความชื้นเพื่อไปวัดในสถานที่อื่นจะต้องมีการเชื่อมโยงสายใหม่ ทำให้เกิดความลำบากยิ่งขึ้น ดังนั้นเราจึงควรหลีกเลี่ยงปัญหาเหล่านี้ โดยการนำเอาการเชื่อมโยงแบบไร้สายด้วยบลูทูธผ่านแอปพลิเคชันที่สร้างขึ้นบนระบบปฏิบัติการแอนดรอยด์มาประยุกต์ใช้ในการส่งข้อมูล ซึ่งสามารถแสดงค่าได้ต่อเนื่องและสามารถตั้งค่าเวลาในการแสดงค่ารวมถึงการแจ้งเตือนได้ ดังนั้นจึงสามารถนำข้อมูลที่ได้นี้ไปประยุกต์ใช้งานต่อไปได้

1.2 วัตถุประสงค์ของโครงงาน

1.2.1 เพื่อสร้างระบบตรวจจับอุณหภูมิ และความชื้นผ่านแอปพลิเคชันบนระบบปฏิบัติการแอนดรอยด์

1.2.2 เพื่อพัฒนาระบบตรวจวัดอุณหภูมิ และความชื้นให้สามารถใช้งานได้กับอุปกรณ์แอนดรอยด์

1.2.3 เพื่อศึกษาและเข้าใจระบบปฏิบัติการแอนดรอยด์

1.3 ขอบเขตโครงงาน

1.3.1 ใช้ระบบ Bluetooth สำหรับรับข้อมูลกับอุปกรณ์แอนดรอยด์

1.3.2 แสดงค่าอุณหภูมิและความชื้นผ่านแอปพลิเคชันบนอุปกรณ์แอนดรอยด์

1.3.3 แอปพลิเคชันสามารถแจ้งเตือนอุณหภูมิ และความชื้นได้ เมื่อข้อมูลที่วัดได้มีค่าสูงกว่าหรือต่ำกว่าค่าที่กำหนดไว้

1.3.4 ระยะทางในการวัดค่าอุณหภูมิและความชื้นไม่เกิน 5 เมตร

1.4 ประโยชน์ที่จะได้รับ

1.4.1 สามารถตรวจวัดอุณหภูมิและความชื้นผ่านบลูทูธด้วยแอปพลิเคชันบนแอนดรอยด์

1.4.2 สามารถใช้แทนกำลังคนในการตรวจวัดอุณหภูมิ และความชื้น เพื่อความสะดวกและลดค่าใช้จ่าย

1.4.3 เรียนรู้และเข้าใจระบบปฏิบัติการแอนดรอยด์ และสามารถเขียนโปรแกรมแอนดรอยด์ได้

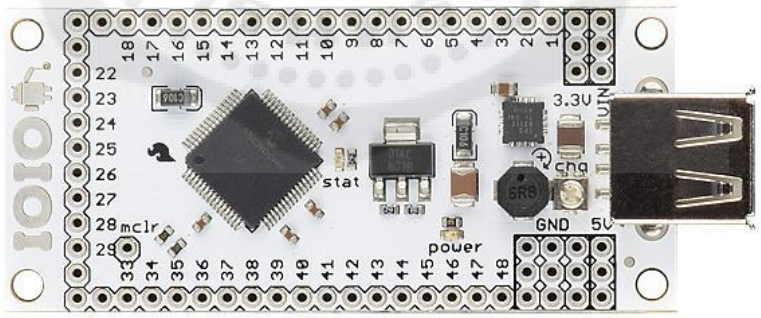
1.4.4 ในระยะที่สัญญาณบลูทูธสามารถส่งข้อมูลไปยังแอปพลิเคชันได้ ผู้ใช้สามารถใช้งานแอปพลิเคชันนี้ได้หลายคนในเวลาเดียวกัน

บทที่ 2
ทฤษฎีที่เกี่ยวข้อง

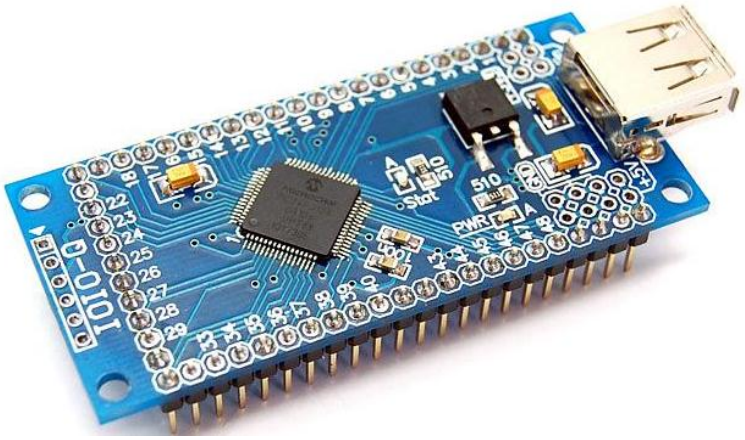
2.1 บอร์ด IOIO-Q

IOIO-Q เป็นบอร์ดไมโครคอนโทรลเลอร์ที่มีการบรรจุเฟิร์มแวร์สำหรับติดต่อกับอุปกรณ์แอนดรอยด์ โดยมีที่มาจากบอร์ด IOIO ซึ่งพัฒนาขึ้นโดยวิศวกรชาวอิสราเอลที่ใช้ชีวิตอยู่ในสหรัฐอเมริกา ซึ่งคุณสมบัติของ IOIO-Q มีคุณสมบัติทางเทคนิคที่เทียบเท่าได้กับ IOIO ทุกประการ

บอร์ด IOIO (อ่านว่า โยโย่) เป็นบอร์ดไมโครคอนโทรลเลอร์ที่พัฒนาขึ้นโดย YTAI เป็นโครงการฮาร์ดแวร์ในลักษณะโอเพนซอร์ส โดยบอร์ด IOIO ทำหน้าที่เป็นบอร์ดอินพุตเอาต์พุตเพื่อช่วยให้อุปกรณ์แอนดรอยด์สามารถติดต่อกับอุปกรณ์ภายนอกผ่านทางพอร์ต USB ได้ โดยตัวบอร์ดใช้ไมโครคอนโทรลเลอร์เบอร์PIC24FJ128DA หรือ PIC24FJ256DA ทำงานเป็น USB โฮสต์ โดยภายในไมโครคอนโทรลเลอร์หลักบรรจุเฟิร์มแวร์สำหรับเชื่อมต่ออุปกรณ์แอนดรอยด์ไว้แล้ว ผู้ใช้งานจึงเพียงพัฒนาโปรแกรมบนอุปกรณ์แอนดรอยด์ โดยไม่ต้องเขียนโปรแกรมสำหรับตัวบอร์ด IOIO อีก ส่งผลให้นักพัฒนาโปรแกรมนำไปประยุกต์หรือสร้างแอปพลิเคชันได้หลากหลายตามต้องการ



รูปที่ 2.1 โครงสร้างบอร์ด IOIO ต้นฉบับพัฒนาโดย YTAI
ที่มา http://androidcontrol.blogspot.com/2012/06/ioio-board-over-bluetooth.html



รูปที่ 2.2 โครงสร้างของบอร์ด IOIO-Q ที่ผลิตโดย INEX
ที่มา http://www.inex.co.th/

2.2 คุณสมบัติทางเทคนิคของบอร์ด IOIO-Q

2.2.1 ใช้ไมโครคอนโทรลเลอร์เบอร์ PIC24FJ128DA หรือ PIC24FJ256DA

2.2.2 ขาอินพุตเอาต์พุตดิจิทัล48 ขา

2.2.3 ขาอินพุตแอนะล็อก16 ขา รับแรงดันได้ 0 ถึง+3.3V ต่อเข้ากับโมดูลแปลงสัญญาณแอนะล็อกเป็นดิจิทัลภายในไมโครคอนโทรลเลอร์ที่มีความละเอียดในการแปลงสัญญาณ 10 บิต

2.2.4 ขาเอาต์พุต PWM 9 ขา ที่สร้างสัญญาณ PWM ด้วยความละเอียดของข้อมูล 10 บิต

2.2.5 ขาสื่อสารข้อมูลอนุกรม UART จำนวน 4 ชุด

2.2.6 ขาสื่อสารข้อมูลอนุกรมผ่านบัส I^2Cจำนวน 3ชุด

2.2.7 คอนเน็กเตอร์ USB แบบAสำหรับต่อกับอุปกรณ์แอนดรอยด์

2.2.8 LED แสดงผลการทำงาน (STATUS) และสถานะไฟเลี้ยง (POWER)

2.2.9 ใช้ไฟเลี้ยงตั้งแต่ +5V ถึง +15V กระแสไฟฟ้า 500mA สำหรับ IOIO

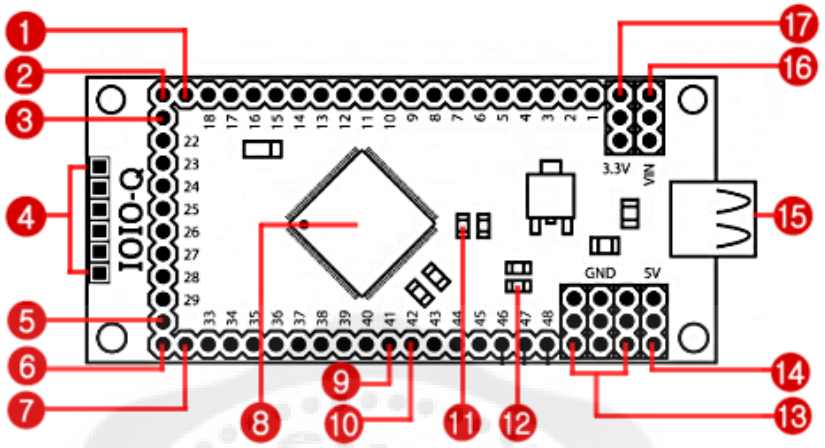
2.2.10 ใช้ไฟเลี้ยงตั้งแต่ +6V ถึง +9V กระแสไฟฟ้า 500mA สำหรับ IOIO-Q

2.2.11มีวงจรควบคุมไฟเลี้ยงคงที่สำหรับวงจรไมโคคอนโทรลเลอร์และอุปกรณ์ภายนอกทั้ง +3.3V และ +5V จ่ายกระแสไฟฟ้าได้ 500 mA ถึง 1A

2.2.12 เชื่อมต่อกับอุปกรณ์แอนดรอยด์แบบ Android Debug Bridge (ADB)

2.2.13 ทำงานร่วมกับอุปกรณ์แอนดรอยด์ที่มีระบบปฏิบัติการ Android ตั้งแต่เวอร์ชัน 1.5 ขึ้นไป

2.3 วงจรของ IOIO-Q



รูปที่ 2.3 ส่วนประกอบที่สำคัญของบอร์ด IOIO-Q
ที่มา หนังสือ Android กับการเชื่อมต่อวงจรอิเล็กทรอนิกส์

หมายเลข 1 ขาพอร์ต 19

หมายเลข 2 ขาพอร์ต 20

หมายเลข 3 ขาพอร์ต 21

หมายเลข 4 จุดต่อเครื่องโปรแกรม PICkit3สำหรับอัปเดตเฟิร์มแวร์

หมายเลข 5 ขาพอร์ต 30

หมายเลข 6 ขาพอร์ต 31

หมายเลข 7 ขาพอร์ต 32

หมายเลข 8 ไมโครคอนโทรลเลอร์ PIC24FJ128DAxx หรือ PIC24FJ256DAxx

หมายเลข 9 ขาพอร์ต 41

หมายเลข 10 ขาพอร์ต 42

หมายเลข 11 LED แจ้งสถานะ (STATUS – ขาพอร์ต 0)

หมายเลข 12 LED แจ้งสถานะไฟเลี้ยง

หมายเลข 13 จุดต่อกราวน์

หมายเลข 14 จุดต่อแรงดันเอาต์พุต +5V

หมายเลข15 คอนเน็กเตอร์USB แบบ A สำหรับต่อกับอุปกรณ์แอนดรอยด์ผ่านทางพอร์ต USB หรือต่อกับบลูทูธดองเกิลในกรณีที่ต้องการติดต่อกับอุปกรณ์ระบบปฏิบัติการแอนดรอยด์แบบไร้สายผ่านบลูทูธ

หมายเลข 16 จุดต่อไฟเลี้ยงเข้า Vin (+6.5 ถึง +9V 500mA)

หมายเลข 17 จุดต่อแรงดันเอาต์พุต +3.3V

ขาหมายเลข 1 ถึง 48 คือขาพอร์ตอินพุตเอาต์พุต สำหรับเชื่อมต่อกับอุปกรณ์อิเล็กทรอนิกส์ภายนอก

ตารางที่ 2.1 หน้าที่ของขาต่าง ๆ บนบอร์ด IOIO-Q

ขา IOIO-Q	อินพุต เอาต์พุต	อินพุต แอนะล็อก	เชื่อมต่อ บัส I^2C	เอาต์พุตติดต่ออุปกรณ์เพอริเฟอรัล	อินพุตติดต่ออุปกรณ์เพอริเฟอรัล	รองรับสัญญาณ +5V	อินพุตวงจรเปรียบเทียบ	ขารองรับการโปรแกรม
1	o							
2	o			o	o	o		
3	o			o	o	o		
4	o		SDA0	o	o	o		
5	o		SCL0	o	o	o		
6	o			o	o	o		
7	o			o	o	o		
8	o						3D	
9	o			o			3C	
10	o			o	o	o		
11	o			o	o	o		
12	o			o	o	o		
13	o			o	o	o		
14	o			o	o	o		
15	o						3B	
16	o						3A	
17	o							
18	o					o		
19	o					o		

ตารางที่ 2.1(ต่อ) แสดงหน้าที่ของขาต่าง ๆ บนบอร์ด IOIO-Q

ขา IOIO-Q	อินพุต เอาต์พุต	อินพุต แอนะล็อก	เชื่อมต่อ บัส I^2C	เอาต์พุตติดต่ออุปกรณ์เพอริเฟอรัล	อินพุต ติดต่ออุปกรณ์เพอริเฟอรัล	รองรับสัญญาณ +5V	อินพุต วงจรเปรียบเทียบ	ขารองรับการโปรแกรม
20	o					o		
21	o					o		
22	o					o		
23	o					o		
24	o					o		
25	o		SCL2			o		
26	o		SDA2			o		
27	o			o	o		1D	
28	o			o	o		1C	
29	o			o	o		2D	
30	o			o	o		2C	
31	o	o		o	o		1A	
32	o	o		o	o		1B	PGC3
33	o	o					2A	PGD3
34	o	o		o	o		2B	
35	o	o		o	o			PGC1
36	o	o		o	o			PGD1
37	o	o		o	o			PGC2
38	o	o		o	o			PGD2
39	o	o		o	o			
40	o	o		o	o			
41	o	o						
42	o	o						
43	o	o						
44	o	o						
45	o	o			o	o		

ตารางที่ 2.1(ต่อ) แสดงหน้าที่ของขาต่าง ๆ บนบอร์ด IOIO-Q

ขา IOIO-Q	อินพุต เอาต์พุต	อินพุต แอนะล็อก	เชื่อมต่อ บัส I^2C	เอาต์พุต ติดต่ออุปกรณ์ เพอริเฟอรัล	อินพุต ติดต่ออุปกรณ์ เพอริเฟอรัล	รองรับ สัญญาณ +5V	อินพุต วงจร เปรียบเทียบ	ขารองรับ การ โปรแกรม
46	o	o			o	o		
47	o			SDA1	o	o		
48	o			SCL1	o	o		
STAT-LED					o	o	o	
MCLR								Vpp

2.3.1 ขาอินพุตเชื่อมต่ออุปกรณ์เพอริเฟอรัล

2.3.1.1 ขารับข้อมูลอนุกรมของ UART โมดูลสื่อสารข้อมูลอนุกรม

2.3.1.2 ขาติดต่อบัส SPI

2.3.1.3 ขาอินพุตตรวจจับสัญญาณ (input capture)

2.3.2 ขาเอาต์พุตเชื่อมต่ออุปกรณ์เพอริเฟอรัล

2.3.2.1 ขารับข้อมูลอนุกรมของ UART โมดูลสื่อสารข้อมูลอนุกรม

2.3.2.2 ขาติดต่อบัส SPI

2.3.2.3 ขาเอาต์พุตวงจรเปรียบเทียบ (comparator)

2.3.2.4 เอาต์พุตสัญญาณ PWM

2.3.3 จุดที่ IOIO-Q แตกต่างกับ IOIO ต้นฉบับ

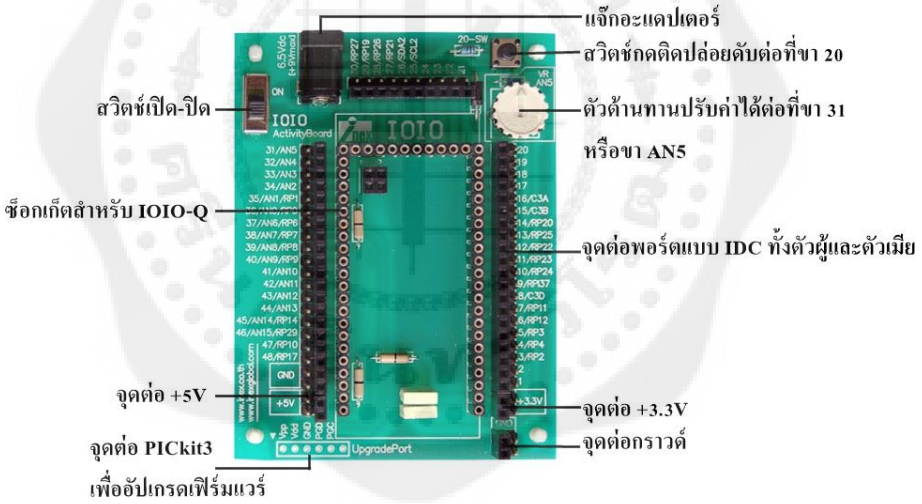
วงจรภาคจ่ายไฟในวงจรนี้เลือกใช้ ไอซีเบอร์ LM1117-5.0 เพื่อควบคุมไฟเลี้ยง +6V ถึง +9V ที่เข้ามาให้มีค่าคงที่ +5V สำหรับจ่ายไปยังอุปกรณ์แอนดรอยด์ผ่านคอนเน็กเตอร์ K1 ซึ่งเป็นพอร์ต USB แบบ A ตัวเมีย นอกจากนั้นแรงดัน +5V ยังถูกส่งไปยังวงจรเรกูเลเตอร์ +3.3V ซึ่งใช้ IC3 เบอร์ LM1117-3.3 ทำให้ได้ไฟเลี้ยง +3.3V สำหรับเลี้ยง IC1

LED1 ทำหน้าที่แสดงสภาวะไฟเลี้ยง ส่วน LED2 สีเหลืองต่อกับขา RD4 ของ IC1 ทำงานด้วยลอกจิกต่ำ ใช้เป็น LED แสดงผลอเนกประสงค์ของบอร์ด IOIO-Q นี้ โดยขาพอร์ตที่ใช้ต่อกับ LED1 คือ ขาพอร์ต 0 ตามการกำหนดโดยเฟิร์มแวร์ของ IOIO ที่บรรจุใน IC1

บอร์ด IOIO-Q มีขาพอร์ตให้ใช้งานรวม 48 ขา โดยมรอินพุตอนาล็อก 16 ขา ซึ่งต่อเข้ากับโมดูลแปลงสัญญาณอะนาลอกเป็นดิจิทัล มีความละเอียด 10 บิต ซึ่งภายในตัว IC1 มีขาพอร์ต ที่ทำหน้าที่เป็นเอาต์พุต PWM ความละเอียด 10 บิต รวม 9 ขา มีขาพอร์ตสื่อสารข้อมูลอนุกรม หรือ UART อีก 4 ชุด และมีขาพอร์ตสำหรับเชื่อมต่อระบบบัส I2C อีก 3 ชุด ที่เหลือจะเป็นขาพอร์ตอินพุตเอาต์พุตดิจิทัล ซึ่งมีการกำหนดตำแหน่งขาเหมือนกับบอร์ด IOIO ต้นฉบับ

2.4 IOIO Activity board

แผงวงจรเอนกประสงค์ สำหรับนำ IOIO บอร์ดไมโครคอนโทรลเลอร์ สำหรับการเชื่อมต่ออุปกรณ์แอนดรอยด์มาติดตั้ง เพื่อใช้ในการทดลองและใช้งานสะดวกขึ้น โดยเตรียมจุดต่อขาพอร์ตเพื่อให้การต่อวงจรทำได้สะดวกรวมถึงมีอุปกรณ์รองรับการทดลองขั้นต้นไว้พร้อมใช้งาน และ มีจุดต่อสำหรับการอัปเดตเฟิร์มแวร์ของตัว IOIO ที่อาจมีในอนาคต



รูปที่ 2.4 IOIO Activity board
ที่มา http://www.inex.co.th/

2.4.1 คุณสมบัติทางเทคนิคของ IOIO Activity board

2.4.1.1 มีซ็อกเก็ตรองรับบอร์ด IOIO-Q รวมถึงจุดต่อ MCLR ที่ใช้อัปเดตเฟิร์มแวร์ด้วย

2.4.1.2 จัดสรรขาพอร์ตใช้งานของ IOIO-Q ทั้งหมดออกมาเป็นคอนเน็กเตอร์ IDC ตัวเมียและตัวผู้เพื่อความสะดวกในการต่อใช้งาน

2.4.1.3 มีจุดต่อ PICkit3 สำหรับการอัปเดตเฟิร์มแวร์ในอนาคต

2.4.1.4 มีจุดต่ออะแดปเตอร์ไฟตรง +6.5V ถึง +9V พร้อมสวิตช์เปิด-ปิด

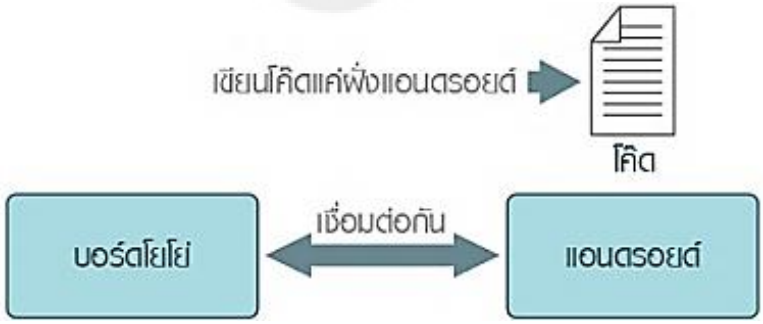
2.4.1.5 มีสวิตช์กดติดปล่อยดับ 1 ตัวต่อกับขาพอร์ต 20 เพื่อทดสอบอ่านค่าอินพุตดิจิทัล

2.4.1.6 มีตัวต้านทานปรับค่าได้ 1 ตัวต่อกับขาพอร์ต AN5 เพื่อการทดสอบอ่านค่าอินพุตแอนะล็อก

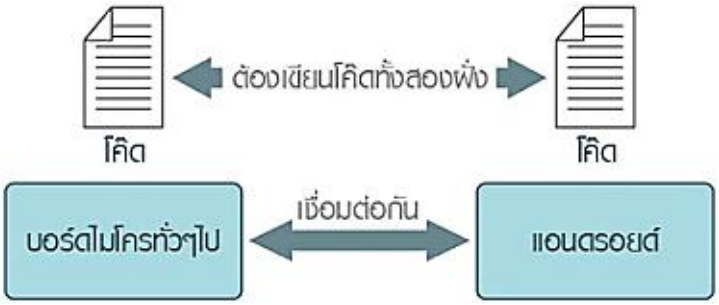
2.4.1.7 มีจุดจ่ายไฟเลี้ยง +3.3V และ +5V 500mA พร้อมกราวด์สำหรับต่ออุปกรณ์ภายนอก

รูปที่ 2.5 IOIO Activity boardที่ติดตั้งบอร์ด IOIO-Qและบลูทูธดองเกิลพร้อมใช้งาน
ที่มา http://www.inex.co.th/

2.5 ความแตกต่างระหว่างบอร์ด IOIO กับ บอร์ดไมโครคอนโทรลเลอร์ทั่วไป



รูปที่ 2.6 การเชื่อมต่อระหว่างบอร์ด IOIO กับแอนดรอยด์
ที่มา http://www.akexorcist.com/2013/11/ioio-board-ioio.html



รูปที่ 2.7 การเชื่อมต่อระหว่างบอร์ดไมโครคอนโทรลเลอร์ทั่วไปกับแอนดรอยด์
ที่มา http://www.akexorcist.com/2013/11/ioio-board-ioio.html

สำหรับบอร์ด IOIO นั่นเป็นบอร์ดไมโครคอนโทรลเลอร์ที่เดิมทีเกิดมาเพื่อเชื่อมต่อกับแอนดรอยด์โดยเฉพาะ โดยต่างจากไมโครคอนโทรลเลอร์ตัวอื่น ๆ เพราะปกติแล้วการเขียนโปรแกรมเชื่อมต่อกับแอนดรอยด์ ไม่ว่าจะใช้บอร์ดไมโครคอนโทรลเลอร์ตัวใดก็ตามจะต้องเขียนโปรแกรมให้กับไมโครคอนโทรลเลอร์และต้องเขียนแอปพลิเคชันบนแอนดรอยด์เพื่อให้สามารถเชื่อมต่อและส่งข้อมูลระหว่างกันได้

บอร์ดไมโครคอนโทรลเลอร์ทั่ว ๆ ไปก็รวมไปถึง Arduino ด้วยเช่นกัน ผู้ใช้งานส่วนใหญ่เข้าใจว่าบอร์ด ไมโครคอนโทรลเลอร์ที่เชื่อมต่อกับแอนดรอยด์ได้นั้น จะมีแค่บอร์ด IOIO และ Arduinoเท่านั้น ซึ่งจริง ๆ แล้วไม่ใช่เลย บอร์ดไมโครคอนโทรลเลอร์ทั่วไปก็สามารถทำได้ ทั้งนี้ขึ้นอยู่กับวิธีในการเชื่อมต่อ ถ้าจะง่ายสุดก็เป็นบลูทูธ จึงเป็นความเข้าใจแบบผิด ๆ ว่าต้อง Arduino หรือ IOIO เท่านั้น บอร์ดทั่ว ๆ ไปนี้จะหมายถึงบอร์ดที่ไม่ได้เกิดมาเพื่อแอนดรอยด์โดยตรง

บอร์ด IOIO จะแตกต่างจากบอร์ดทั่ว ๆ ไปตรงจุดนี้นั่นเอง เพราะเกิดมาเพื่อเชื่อมต่อและถูกสั่งงานจากแอนดรอยด์ ไม่สามารถทำงานได้ด้วยตัวเอง ต้องรอคำสั่งจากแอนดรอยด์เท่านั้น เนื่องจากการที่เกิดมาเพื่อแอนดรอยด์ ผู้พัฒนาจึงทำให้ผู้ใช้งานไม่จำเป็นต้องเขียนโค๊ดให้กับบอร์ด IOIO เลย เพราะจะมีโค๊ดใส่มาในบอร์ดให้พร้อมไว้เรียบร้อยแล้ว (หรือที่เรียกกันว่าเฟิร์มแวร์นั่นเอง)

ดังนั้นผู้ใช้งานจึงเขียนโค๊ดแค่ฝั่งแอนดรอยด์เท่านั้น โดยผู้ผลิตจะมีไลบรารีของ บอร์ด IOIO ให้ใช้ในโค๊ดฝั่งแอนดรอยด์เลย ดังนั้นจึงสามารถสั่งงานบอร์ด IOIO ด้วยคำสั่ง ในแอปพลิเคชันด้วยจุดดีข้อนี้ จึงทำให้ง่ายต่อการนำไปใช้งาน ไม่ต้องวุ่นวายกับการเชื่อมต่อ เพราะเฟิร์มแวร์ทำให้หมด และจะทำงานทันทีที่เชื่อมต่อกับแอนดรอยด์ แต่นั้นก็กลายเป็น

ข้อเสียอีกอย่างหนึ่งเช่นกัน เพราะว่าจะไม่สามารถทำงานด้วยตัวเองได้ (Standalone) เนื่องด้วยวิธีการทำงานของบอร์ด IOIO นั่นเองที่ทำงานแบบ Realtime คือต้องรอแอนดรอยด์สั่งงานทุกครั้ง ไม่ได้รับโค๊ดทั้งหมดจากแอนดรอยด์แล้วมาทำงานเองทั้งหมด แต่จะรอคำสั่งจากแอนดรอยด์แล้วทำคำสั่งนั้น ๆ ทีละคำสั่งเรื่อย ๆ เมื่อทำคำสั่งนั้น ๆ เสร็จแล้ว แอนดรอยด์ก็จะสั่งให้ทำคำสั่งต่อไปเรื่อย ๆ

จึงสรุปได้ว่าบอร์ด IOIO เป็นบอร์ดไมโครคอนโทรลเลอร์ที่ใช้สำหรับเชื่อมต่อกับ แอนดรอยด์แล้วทำงานตามคำสั่งในแอนดรอยด์ สำหรับคำสั่งที่จะสั่งงานผู้ใช้ก็ต้องเขียนขึ้นมาเป็นแอปพลิเคชันแทน

2.6 การเชื่อมต่อระหว่างบอร์ด IOIO กับอุปกรณ์แอนดรอยด์

บลูทูธ (Bluetooth) เป็นการเชื่อมต่อผ่านสัญญาณบลูทูธ โดยบอร์ด IOIO สามารถเสียบ Bluetooth Dongleได้ แล้วให้ฝั่งแอนดรอยด์เชื่อมต่อบลูทูธ ก็ใช้งานผ่านบลูทูธได้แล้ว (ต้องเพิ่มไลบรารีสำหรับเชื่อมต่อผ่านบลูทูธด้วย)

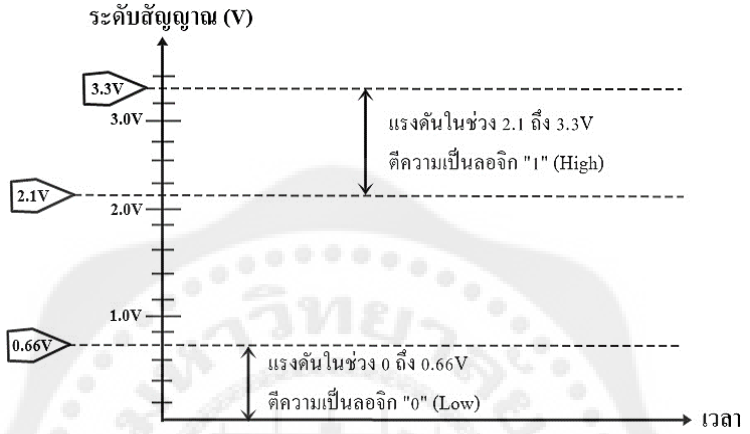
รูปที่ 2.8 การเชื่อมต่อแบบ Bluetooth
ที่มา http://www.akexorcist.com/

2.7 การใช้งานพอร์ตอินพุตเอาต์พุตดิจิทัลของ IOIO-Q

สำหรับการติดต่อเพื่อส่งและอ่านค่าจากขาพอร์ตอินพุตเอาต์พุตดิจิทัล ถือว่าเป็นพื้นฐานที่สุดสำหรับการใช้งานไมโครคอนโทรลเลอร์

2.7.1 การตีความลอจิกของ IOIO-Q

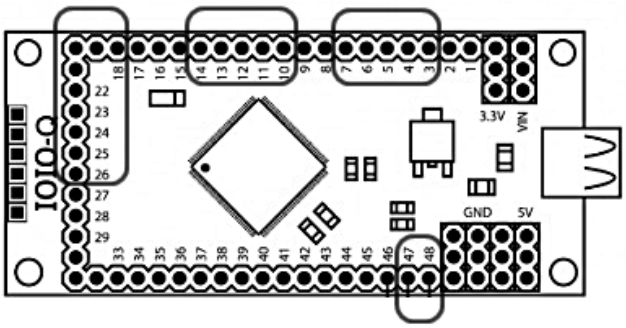
สำหรับบอร์ด IOIO-Q มีการกำหนดระดับสัญญาณเพื่อตีความว่าเป็นสัญญาณลอจิก “0” หรือ “1” ด้วยการอ้างอิงกับไฟเลี้ยง +3.3V โดยสัญญาณลอจิก “0” มีระดับแรงดันในช่วง 0 ถึง 0.66V ส่วนสัญญาณลอจิก “1” มีระดับแรงดันในช่วง 2.1 ถึง 3.3V โดยข้องมูลนี้อ้างอิงจากข้อมูลคุณสมบัติของไมโครคอนโทรลเลอร์ PIC24FJ128DA และ PIC24FJ256DA



รูปที่ 2.9 การตีความลอจิกจากระดับสัญญาณไฟตรงของบอร์ด IOIO-Q
ที่มา หนังสือ Android กับการเชื่อมต่อวงจรอิเล็กทรอนิกส์

ในรูปที่ 2.9 แสดงการตีความลอจิกของบอร์ด IOIO-Q จะเห็นว่าระดับสัญญาณในช่วง 0.7 ถึง 2.0V อาจเป็นช่วงที่ตีความหมายเป็นลอจิกที่แน่นอนไม่ได้ ในการใช้งานจริงจึงต้องคำนึงถึงข้อตกลงลองสัญญาณลอจิกนี้ร่วมด้วย

2.7.2 ความสามารถในการรับแรงดันของขาพอร์ตของบอร์ด IOIO-Q



รูปที่ 2.10 การเชื่อมต่อ IOIO-Q กับอุปกรณ์ภายนอกที่ใช้ไฟเลี้ยง +5V
ที่มา หนังสือ Android กับการเชื่อมต่อวงจรอิเล็กทรอนิกส์

เนื่องจาก IOIO-Q สร้างขึ้นจากไมโครคอนโทรลเลอร์ที่ต้องการไฟเลี้ยง +3.3V ดังนั้นความสามารถในการขับแรงดันและการรับกระแสไฟฟ้าของขาพอร์ตปกติจึงไม่เกิน +3.3V

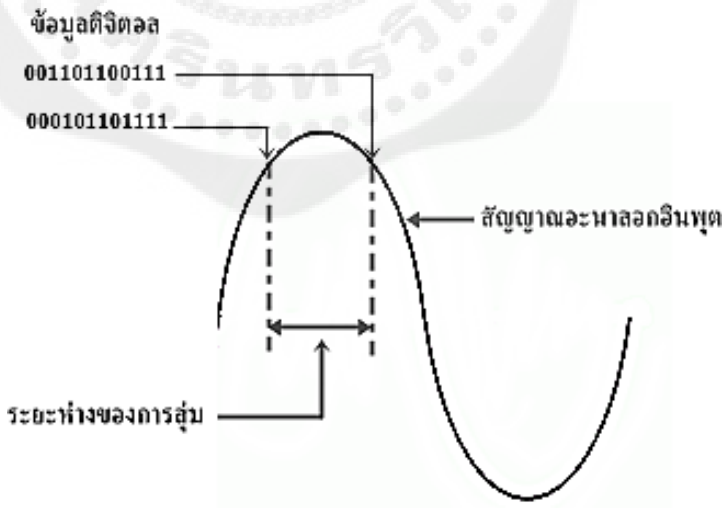
แต่ในบางขาพอร์ตของ IOIO-Q มีความสามารถที่พิเศษเพิ่มขึ้นนั่นคือ การรับแรงดันอินพุตที่ป้อนเข้ามาของขาพอร์ตดิจิทัลที่สูงได้ถึง +5V ในกรณีที่ต้องใช้งานร่วมกับอุปกรณ์หรือตัวตรวจจับที่ใช้แรงดัน +5V สำหรับลอจิก “1” จะต้องเชื่อมต่อเข้ากับขาพอร์ตของ IOIO-Q ที่มีการทำงานเป็นแบบเดรนเปิด (open drain) นั่นคือ ขา 3 ถึง 7, 10 ถึง 14, 18 ถึง 26 และ 47 ถึง 48 เท่านั้น หากต่อขาอื่นจะทำให้บอร์ด IOIO-Q เกิดความเสียหายได้

ในรูปที่ 2.10 แสดงลักษณะการต่อวงจรของขาพอร์ตแบบเดรนเปิดของ IOIO-Q กับอุปกรณ์ที่ใช้ไฟเลี้ยง +5V

2.8 IOIO-Q กับการอ่านค่าสัญญาณแอนะล็อก

การจัดการกับสัญญาณแอนะล็อกของบอร์ด IOIO-Q นั้น มีอินพุตรับแอนะล็อก16 ขา รับแรงดันไฟตรงได้ย่าน 0 ถึง +3.3V มีความละเอียดของแปลงสัญญาณ 10 บิต หรือ 3mV สำหรับในกรณีที่ใช้กับแรงดัน +5V แนะนำให้ต่อผ่านวงจรแบ่งแรงดันก่อนที่จะป้อนเข้าที่ขาอินพุตแอนะล็อก สำหรับขาพอร์ตของ IOIO-Q ที่ใช้เป็นอินพุตแอนะล็อกคือ ขา 31 ถึง 46 รวม16 ขา

2.8.1 การแปลงสัญญาณแอนะล็อกเป็นดิจิทัล(ADC: Analog to Digital Converter)



รูปที่ 2.11 การสุ่มสัญญาณแอนะล็อกเพื่อกำหนดข้อมูลดิจิทัล

ที่มา หนังสือ Android กับการเชื่อมต่อวงจรอิเล็กทรอนิกส

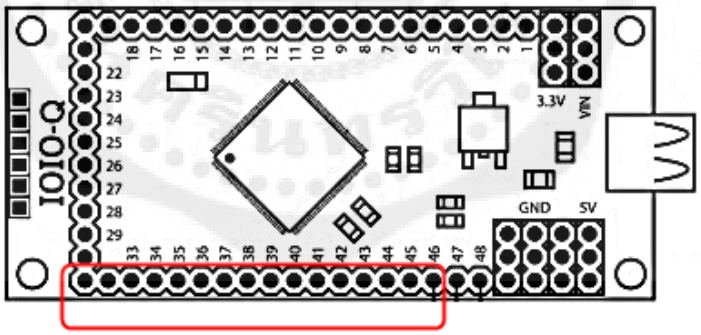
การแปลงสัญญาณแอนะล็อกเป็นดิจิตอล สัญญาณจะได้รับการแปลงเป็นจำนวนดิจิทัลโดยการสุ่มหรือแซมปลิ้ง (Sampling) ดังในรูป ถ้าหากวงจร ADC มีเอาต์พุต 8 เส้น จะมีความแตกต่างทางรหัสเลขฐานสองทั้งหมด 2^8 หรือ 256 รหัส

ค่าความละเอียดของตัวแปลงสัญญาณแอนะล็อกเป็นดิจิทัลถูกอธิบายเป็นระยะห่างที่น้อยที่สุดของค่าแรงดันทางอินพุตที่เพิ่มขึ้น ซึ่งถูกกำหนดโดยตัวแปลงสัญญาณนั้น ระยะห่างยิ่งน้อยเท่าใด ค่าความละเอียดยิ่งสูงขึ้น โดยค่าความละเอียดแปรผันตรงกับจำนวนของบิตเอาต์พุต ยกตัวอย่างเช่น ถ้าตัวแปลงสัญญาณมีความแตกต่างทางรหัสเอาต์พุต 256 ระดับสัญญาณอินพุตถูกแทนเป็นรหัสเลขฐานสองจาก 00000000 ถึง 11111111 ถ้าย่านอินพุตเริ่มต้นจาก 0 ถึง +3.3V ดังนั้นความละเอียดเท่ากับ $\frac{3.3}{256} = 0.01289\text{V}$

ถ้ารหัสเลขฐานสองเป็น 00000001 แทนแรงดัน 0.01289V ในทำนองเดียวกัน ถ้าเอาต์พุตรหัสเลขฐานสองเป็น 00000010 ก็จะแทนแรงดัน 0.02578V ดังนั้นข้อมูลของแรงดัน 1.50V จะมีค่าเท่ากับ $\frac{1.5}{0.01289} = 116_{10}$

ทำการแปลงเป็นเลขฐานสองหรือข้อมูลดิจิทัลจะได้ค่าเท่ากับ 01110100_2

2.8.2 อินพุตแอนะล็อกของ IOIO-Q



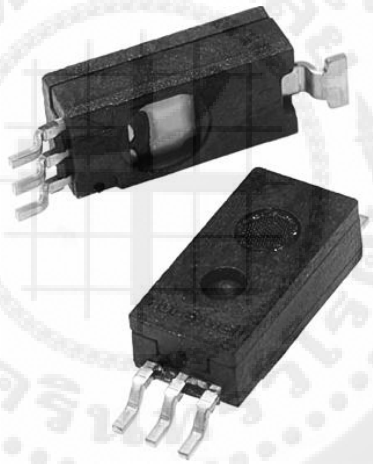
รูปที่ 2.12 ระบุตำแหน่งขาอินพุตแอนะล็อกของบอร์ด IOIO-Q
ที่มา หนังสือ Android กับการเชื่อมต่อวงจรอิเล็กทรอนิกส์

บอร์ด IOIO-Q มีอินพุตแอนะล็อกให้เลือกใช้งานทั้งสิ้น 16 ขาคือ ขา 31 ถึง 46 โดยแต่ละขาอินพุตรับแรงดันไฟตรงได้ย่าน 0 ถึง +3.3V หากสัญญาณอินพุตที่เข้ามามีระดับสัญญาณสูงกว่า +3.3V จะต้องใช้วงจรลดทอนแรงดันเข้ามาช่วย ความละเอียดในการแปลงสัญญาณคือ 10 บิต

มีค่าข้อมูล 0 ถึง 1,023 รวม 1,024 ค่า เมื่อเทียบกับไฟเลี้ยง +3.3V ค่าข้อมูล 1 บิต จึงเท่ากับแรงดัน 0.003V โดยประมาณ

2.9 ตัวตรวจจับความชื้นสัมพัทธ์HIH4030

HIH4030 เป็นตัวตรวจจับและวัดความชื้นสัมพัทธ์ในอากาศที่ให้ผลการทำงานเป็นแรงดันไฟตรงที่สัมพันธ์กับค่าความชื้นสัมพัทธ์ HIH4030 ผลิตโดย Honeywell ผู้ผลิตอุปกรณ์ตรวจจับระดับโลกที่มีใช้งานในอุตสาหกรรมอิเล็กทรอนิกส์ทั่วโลก ด้วยการให้ผลการทำงานเป็นแรงดันไฟตรงของ HIH4030 ทำให้ช่วยลดความซับซ้อนในการเขียนโปรแกรมเพื่ออ่านค่าความชื้นสัมพัทธ์ เพียงต่อเข้ากับอินพุตของโมดูลแปลงสัญญาณแอนะล็อกเป็นดิจิทัล แล้วอ่านค่ามาคำนวณ ก็จะได้ค่าความชื้นสัมพัทธ์ออกมา



รูปที่ 2.13 โมดูลวัดความชื้นสัมพัทธ์ HIH4030

ที่มา http://www.digitalbees.info/pmwiki.php/Main/HIH-4030

รูปที่ 2.14 โมดูลแผงวงจร HIH4030 ที่พร้อมใช้งาน

ที่มา http://www.rpelectronics.com/sen-09569-humidity-sensor-hih-4030-breakout.html

2.9.1 คุณสมบัติเด่นของโมดูล HIH4030

2.9.1.1 เป็นโมดูลวัดความชื้นสัมพัทธ์ (%RH) ที่ให้ผลการทำงานเป็นแรงดันไฟตรงที่แปรผันตรงกับค่าความชื้นสัมพัทธ์ในแบบใกล้เคียงเชิงเส้นมาก

2.9.1.2 ย่านวัดความชื้นสัมพัทธ์ 0 %RHถึง 100 %RHเทียบกับค่าแรงดันเอาต์พุตประมาณ 0.958Vที่ไฟเลี้ยง +5V

2.9.1.3 มีความแม่นยำในการวัดสูง มีความผิดพลาด ±3.5%RHที่ไฟเลี้ยง +5V และอุณหภูมิห้อง 25°C

2.9.1.4 ใช้ไฟเลี้ยง4ถึง 5.8V ต้องการกระแสไฟฟ้าเพียง200 μA

2.9.1.5 มีขาต่อใช้งาน *3* ขา คือ กราวด์ สัญญาณเอาต์พุต และไฟเลี้ยง

2.9.1.6 มีความเที่ยงตรงสูง และความไวในการทำงานสูง

2.9.1.7 ขนาด0.75×0.30 นิ้ว (19.05 × 7.62 มิลลิเมตร)

2.9.2 การใช้งาน HIH4030กับ IOIO-Q

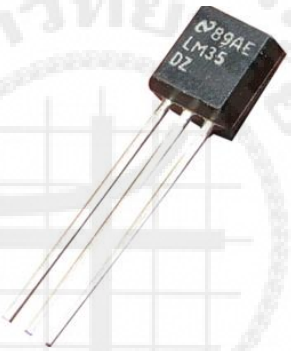
เอาต์พุตที่ได้จาก HIH4030 อยู่ในรูปของแรงดันไฟตรงที่สัมพันธ์กับค่าความชื้นสัมพัทธ์ในอากาศ คำนวณได้จากความสัมพันธ์ดังนี้

$$\%RH = \left(\frac{V_{out} - 0.958}{0.0307}\right)$$

โดยที่ V_{out} คือ แรงดันเอาต์พุตที่ได้จาก HIH4030 ในหน่วยโวลต์ (V)

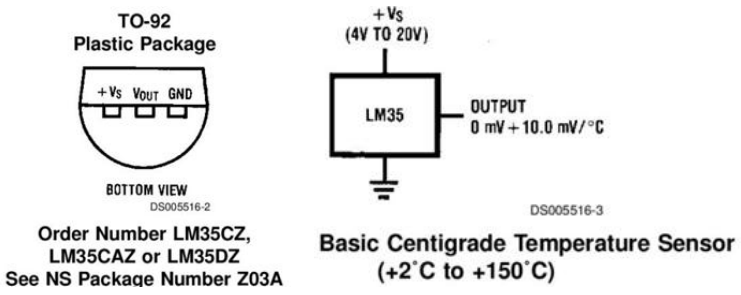
เมื่อนำไปใช้กับ IOIO-Qต้องเชื่อมต่อขาสัญญาณเอาต์พุตของ HIH4030เข้ากับอินพุตแอนะล็อกโดยจะต้องระมัดระวังในการใช้งาน เนื่องจาก HIH4030ทำงานที่ไฟเลี้ยง +5V และเอาต์พุตได้ในช่วง 0.958V ถึง 4.065V ซึ่งสูงเกินกว่าที่อินพุตแอนะล็อกของ IOIO-Q จะรับได้ แต่ที่ค่าความชื้นสัมพัทธ์ 75 %RH โมดูล HIH4030 จะให้แรงดันไฟตรงออกมาประมาณ 3.268V ในสภาพห้องปกติค่าความชื้นสัมพัทธ์มีค่าประมาณ 25 %RH ถึง 60%RH จึงนำ HIH4030 มาใช้งานกับ IOIO-Q ได้โดยไม่ต้องใช้วงจรแบ่งแรงดัน แต่จะต้องแน่ใจว่าในพื้นที่ ที่จะนำ IOIO-Q และ HIH4030 ไปใช้งานร่วมกันควรมีค่าความชื้นสัมพัทธ์ไม่เกิน 75 %RH

2.10 ไอซีวัดอุณหภูมิ LM35DZ



รูปที่ 2.15 ไอซีวัดอุณหภูมิ LM35DZ

ที่มา http://www.mikroelectron.com/lm35-temperature-sensor



รูปที่ 2.16 ข้อมูลพื้นฐานของไอซีวัดอุณหภูมิ LM35DZ

ที่มา http://www.alldatasheet.com/datasheet-pdf/pdf/8875/NSC/LM35DZ.html

ไอซีวัดอุณหภูมิแบบแอนะล็อก เบอร์ LM35DZ ผลิตโดยบริษัท National Semiconductor ตัวถังแบบ TO-92 มีสามขาคือ $+V_S$ V_{out} และ GND ตามรูปที่ 2.16 โดยเป็นเซนเซอร์วัดอุณหภูมิที่มีความถูกต้องแม่นยำสูง (ไม่เกิน ±0.5°C) ให้เอาต์พุตเป็นแรงดันที่มีความสัมพันธ์กับอุณหภูมิในหน่วยเซลเซียส (Celsius) แบบเชิงเส้น มีอัตราขยายเท่ากับ 10 mV/°C (หมายความว่า ถ้าอุณหภูมิเพิ่มขึ้นหรือลดลง 1°C จะทำให้แรงดันที่ขาเอาต์พุตเพิ่มหรือลดลง 10มิลลิโวลต์) และสามารถวัดอุณหภูมิได้ในช่วง 0°C ถึง 100°C ดังนั้นอุณหภูมิที่ 0°C จะได้แรงดันเท่ากับ 0 มิลลิโวลต์ และอุณหภูมิที่ 100°C จะได้ 1000 มิลลิโวลต์ หรือ 1 โวลต์ และกล่าวได้ว่า แรงดันเอาต์พุตจะอยู่ในช่วง 0 ถึง 1 โวลต์ เท่านั้น

2.11 การติดตั้งเครื่องมือพัฒนาแอปพลิเคชันสำหรับอุปกรณ์แอนดรอยด์

2.11.1 ซอฟต์แวร์สำหรับการพัฒนาแอปพลิเคชัน

ในการพัฒนาแอปพลิเคชั่นผู้พัฒนาต้องดาวน์โหลดซอฟต์แวร์ซึ่งเป็นเครื่องมือสำคัญจากอินเทอร์เน็ตเพื่อนำมาติดตั้งลงในคอมพิวเตอร์ จากนั้นจึงเริ่มต้นพัฒนาแอปพลิเคชั่นให้กับอุปกรณ์แอนดรอยด์ต่อไป โดยซอฟแวร์ที่สำคัญทั้งหมดมีดังนี้

2.11.1.1 Eclipseเป็นซอฟต์แวร์พัฒนาโปรแกรมภาษาจาวาที่รองรับการพัฒนาแอปพลิเคชันสำหรับอุปกรณ์แอนดรอยด์เป็นที่นิยมมากที่สุด เพราะเป็นโอเพ่นซอร์สซอฟต์แวร์ที่พัฒนาขึ้นเพื่อใช้งานโดยนักพัฒนาเอง ทำให้ Eclipse มีการพัฒนาอย่างต่อเนื่องและรวดเร็ว Eclipse มีจุดเด่นที่เรียกว่า Plug-inDevelopment Environment (PDE) ที่เพิ่มความสามารถในการพัฒนาซอฟแวร์ได้มากขึ้น ทั้งยังรองรับการนำ plug-in มาติดตั้งเพิ่มให้กับ Eclipse ได้ ซึ่งชุดพัฒนาอุปกรณ์แอนดรอยด์ก็เป็นการนำplug-in สำหรับพัฒนาแอปพลิเคชันมาติดตั้งลงใน Eclipse เช่นกัน นอกจากนี้ยังรองรับการทำงานได้กับไฟล์หลายชนิด อาทิ HTML JAVA ภาษา C และ XML เป็นต้น และที่สำคัญที่สุดก็คือใช้งานกับระบบปฏิบัติการ Windows Linux และ Mac OS ได้

2.11.1.2 Java Development Kit (JDK) เป็นชุดคำสั่งในการพัฒนาโปรแกรมด้วยภาษาจาวา และบรรจุเครื่องมือในการพัฒนาโปรแกรม Java Compiler, Java Debugger หรือ Java VM(Java Virtual Machine) เป็นต้นอันเป็นหัวใจสำคัญสำหรับการพัฒนาโปรแกรมด้วยภาษาจาวา

2.11.1.3 Android Software Development Kit (Android SDK) เป็นชุดโปรแกรมที่ Google ได้พัฒนาขึ้นมาเพื่อแจกจ่ายให้ผู้พัฒนาแอปพลิเคชันโดยบรรจุ โปรแกรมและไลบรารีต่าง ๆ

ที่จำเป็นต่อการพัฒนาแอปพลิเคชันสำหรับอุปกรณ์แอนดรอยด์ เช่น อีมูเลเตอร์ (Emulator) หรือ ตัวจำลองการทำงานที่ทำให้ผู้พัฒนาสามารถสร้างแอปพลิเคชันและทดลองบนตัวจำลองซึ่งมีลักษณะการทำงานเหมือนกับอุปกรณ์ แอนดรอยด์จริง ๆ ก่อนที่จะทำการติดตั้งลงในอุปกรณ์แอนดรอยด์จริงต่อไป

2.11.2 การติดตั้งเครื่องมือทางซอฟต์แวร์บนระบบปฏิบัติการ Windows

สำหรับการติดตั้งซอฟต์แวร์บนวินโดวส์ที่นำมาอธิบายเป็น Windows 7 รุ่น 64 บิต

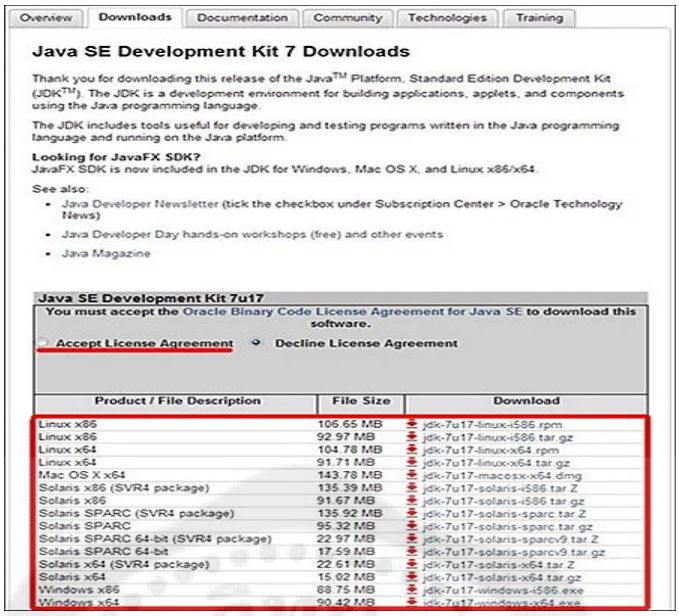
ขั้นตอนที่ 1 : ดาวน์โหลด และติดตั้ง Java Development Kit (JDK)

1. เข้าไปดาวน์โหลดที่เว็บไซต์ Oracle และเลือกดาวน์โหลด "Java Platform (JDK) 7u17"

รูปที่ 2.17 เว็บไซต์ Oracle ที่ใช้ในการดาวน์โหลด

ที่มา http://m.thaiware.com/reviewDetail.php?review_id=384

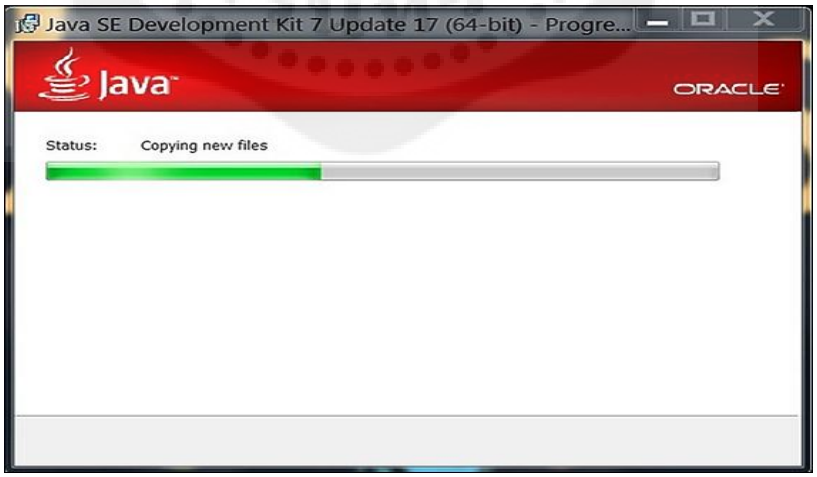
2. หลังจากนั้นให้เลือก "ยอมรับเงื่อนไข" และเลือกดาวน์โหลดให้ตรงกับระบบปฏิบัติการที่คอมพิวเตอร์ใช้อยู่ (Windows 7รุ่น 64 bit)



รูปที่ 2.18 ขั้นตอนการดาวน์โหลด Java Development Kit

ที่มา http://m.thaiware.com/reviewDetail.php?review_id=384

3. หลังจากดาวน์โหลดตัวติดตั้งแล้วรันไฟล์ติดตั้งจะพบหน้าต่างสำหรับเริ่มติดตั้งโปรแกรมปรากฏขึ้นมาคลิกปุ่ม Next จนกระทั่งดาวน์โหลดและติดตั้งโปรแกรมสำเร็จ



รูปที่ 2.19 หน้าต่างสำหรับเริ่มติดตั้งโปรแกรม Java Development Kit

ที่มา http://m.thaiware.com/reviewDetail.php?review_id=384

ขั้นตอนที่ 2 : ดาวน์โหลด และติดตั้ง Eclipse IDE

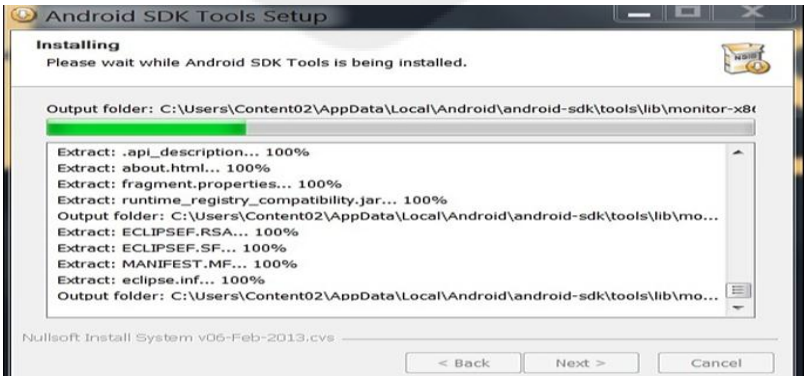
ดาวน์โหลด Eclipse IDE ที่ http://www.eclipse.org/downloads/ และเลือกดาวน์โหลด "Eclipse Classic" 64 bit หลังจากที่ดาวน์โหลดเสร็จเรียบร้อย จะได้ไฟล์ .ZIP มา จากนั้นแตกไฟล์ออกมาไว้ที่ D:\Eclipse

รูปที่ 2.20 การดาวน์โหลด Eclipse IDE

ที่มา http://m.thaiware.com/reviewDetail.php?review_id=384

ขั้นตอนที่ 3 : ดาวน์โหลด และติดตั้ง Android SDK

1.ดาวน์โหลด Android SDK และติดตั้งโปรแกรม



รูปที่ 2.21 ดาวน์โหลดและติดตั้ง Android SDK

ที่มา http://m.thaiware.com/reviewDetail.php?review_id=384

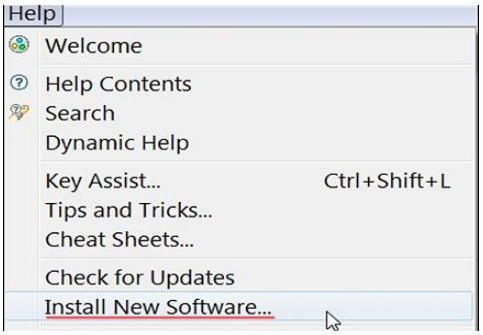
2. เมื่อติดตั้งเสร็จสมบูรณ์ ให้เปิด Android SDK Manager (Start > Android SDK Tools > SDK Manager และเลือกดาวน์โหลดเวอร์ชันที่เราต้องการจะพัฒนา หลังจากคลิก Install Packages แล้วจะมีป็อปอัพเด้งขึ้นมาให้เลือก "Accept All" (เหตุผลที่เลือกดาวน์โหลด API หลายตัว เพราะเวลาเราพัฒนาแอพพลิเคชั่นไปเรื่อย ๆ จะเจอ Errors หรือปัญหาอื่น ๆ ที่เกิดขึ้นเพราะเราไม่ได้โหลด API นั้น ๆ มา เพราะฉะนั้นจึงดาวน์โหลด API ที่จำเป็นไว้ด้วย)

รูปที่ 2.22 ดาวน์โหลดเวอร์ชันที่ต้องการพัฒนา

ที่มา http://m.thaiware.com/reviewDetail.php?review_id=384

ขั้นตอนที่ 4 : ติดตั้ง Plugin เสริมใน Eclipse

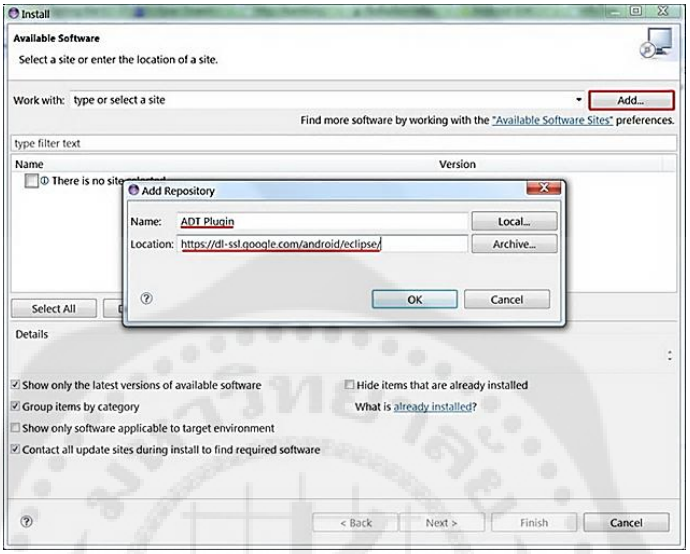
1. เปิดโปรแกรม Eclipse ขึ้นมา หลังจากนั้นไปที่ Help > Install New Hardwar



รูปที่ 2.23 ขั้นตอนแรกในการติดตั้ง Plugin เสริมใน Eclipse

ที่มา http://m.thaiware.com/reviewDetail.php?review_id=384

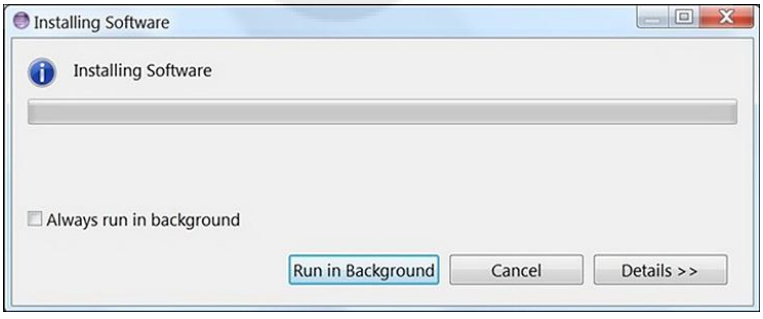
2. คลิกที่ Add และใส่ "ADT Plugin" ในช่อง Name ส่วนช่อง Location ให้ใส่ https://dl-ssl.google.com/android/eclipse/



รูปที่ 2.24 การใส่ค่าต่าง ๆ ในโปรแกรม Eclipse

ที่มา http://m.thaiware.com/reviewDetail.php?review_id=384

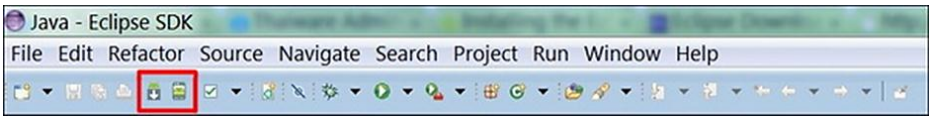
3. หลังจากนั้นเลือก "Developer Tools" แล้วคลิก Next ไปเรื่อย ๆ จนกว่าจะติดตั้งสำเร็จ



รูปที่ 2.25 หน้าต่าง Installing Software

ที่มา http://m.thaiware.com/reviewDetail.php?review_id=384

4. เมื่อติดตั้งเสร็จ โปรแกรม Eclipse ก็จะถามขึ้นมาว่าต้องการรีสตาร์ทเครื่องไหม แนะนำให้รีสตาร์ท หลังจากนี้เราจะสังเกตไอคอน Android สองตัวเล็ก ๆ นั่นคือ "Android SDK Manager" และ "Android Virual Device Manager"

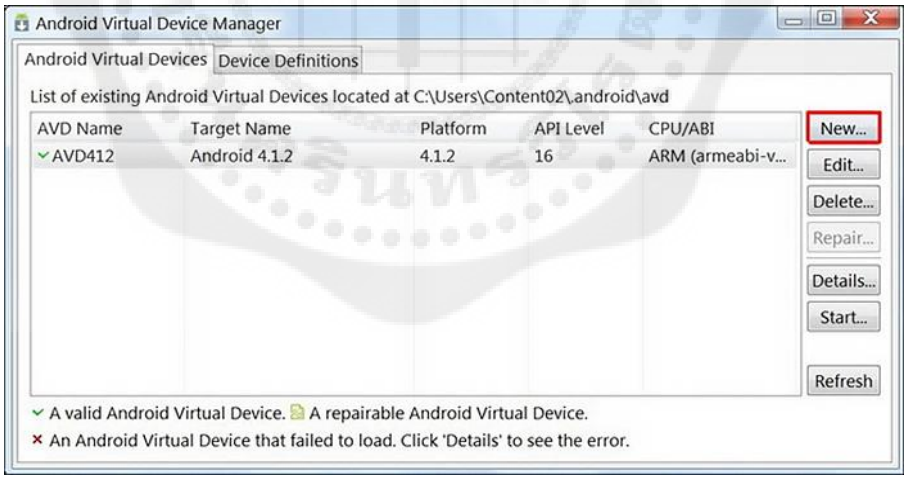


รูปที่ 2.26 หน้าต่าง Java-Eclipse SDK

ที่มา http://m.thaiware.com/reviewDetail.php?review_id=384

ขั้นตอนที่ 5 : ติดตั้ง และตั้งค่า Emulator

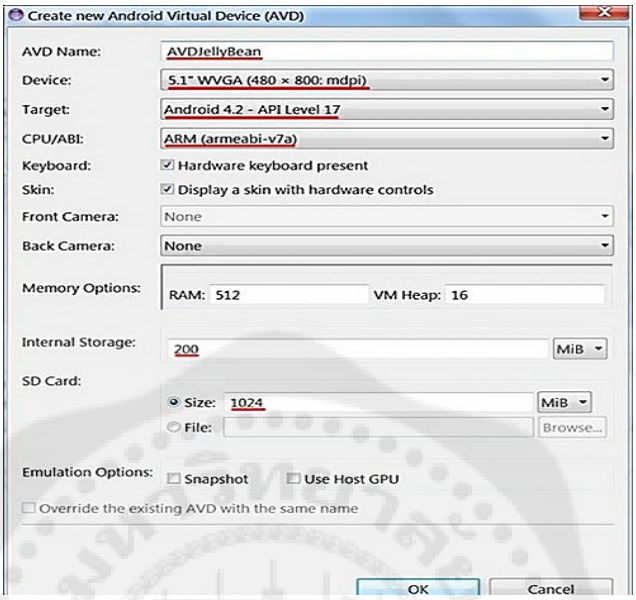
1. เลือกไอคอน "Android Virtual Device Manager" และคลิกที่ "New"



รูปที่ 2.27 หน้าต่าง Android Virtual Device Manager

ที่มา http://m.thaiware.com/reviewDetail.php?review_id=384

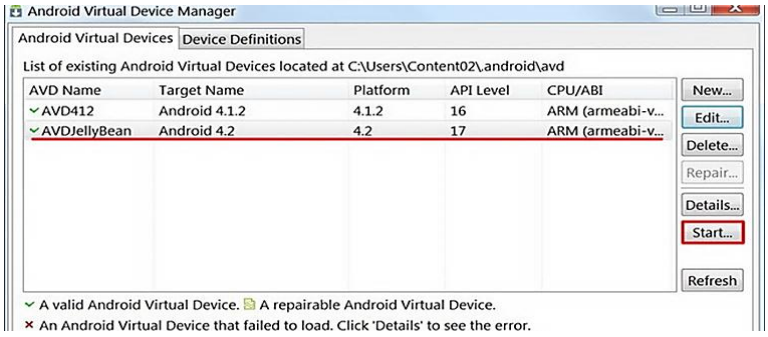
2. ให้เราตั้งชื่อ AVD และตั้งค่าของ Emulator ที่เราต้องการ



รูปที่ 2.28 การตั้งชื่อ AVD และตั้งค่า Emulator

ที่มา http://m.thaiware.com/reviewDetail.php?review_id=384

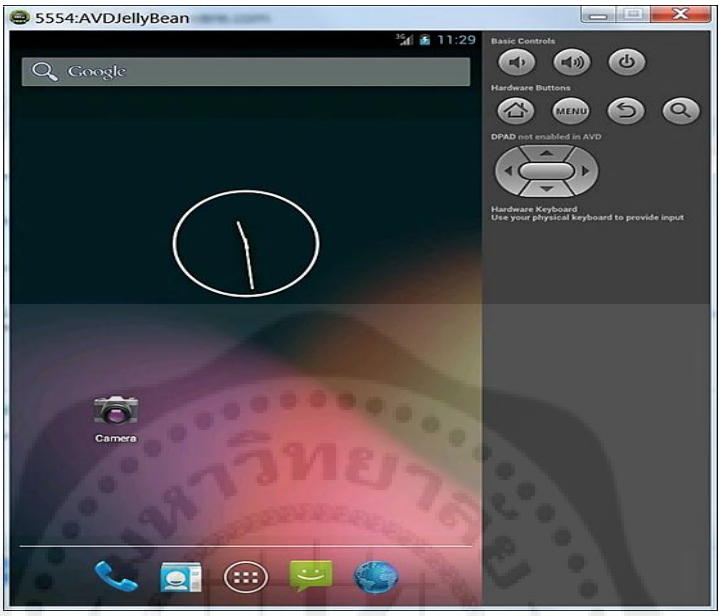
3. เสร็จเรียบร้อยคราวนี้เราลองทดสอบ Emulator ที่สร้างขึ้นมา โดยเลือก "AVDJellyBean" และคลิก "Start"



รูปที่ 2.29 การลองทดสอบ Emulator ที่สร้างขึ้นมา

ที่มา http://m.thaiware.com/reviewDetail.php?review_id=384

4. จะเห็นว่าเราสามารถรัน Emulator โทรศัพท์ Android ได้



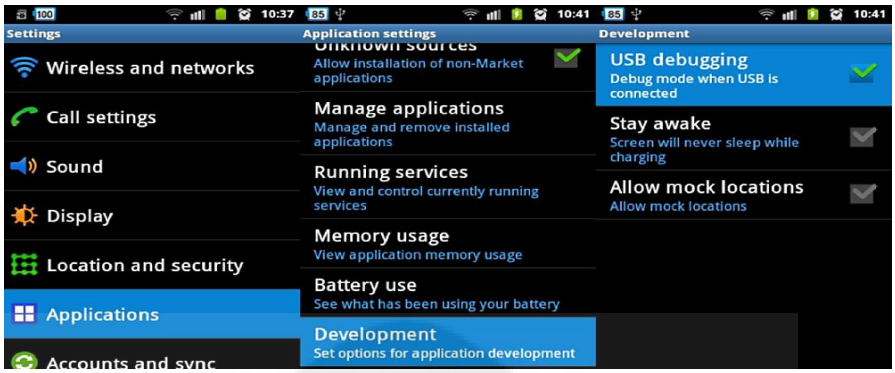
รูปที่ 2.30 หน้าต่างแสดงการรัน Emulator โทรศัพท์ Android

ที่มา http://m.thaiware.com/reviewDetail.php?review_id=384

2.12 ทดสอบการติดตั้งระหว่างอุปกรณ์แอนดรอยด์และบอร์ด IOIO-Q

การทดสอบการทำงานของบอร์ด IOIO-Q แบ่งออกเป็น 2 กรณีคือ เชื่อมต่อบอร์ด IOIO-Q กับอุปกรณ์แอนดรอยด์ผ่านพอร์ต USB โดยใช้สาย microUSBและเชื่อมต่อบอร์ด IOIO-Q กับอุปกรณ์แอนดรอยด์แบบไร้สายผ่านบลูทูธ

2.12.1 เตรียมการอุปกรณ์แอนดรอยด์



รูปที่ 2.31 ตัวอย่างการเปิดใช้งาน USB Debugging บนแอนดรอยด์เวอร์ชันต่ำกว่า 4.2.1
ที่มา หนังสือ Android กับการเชื่อมต่อวงจรอิเล็กทรอนิกส์

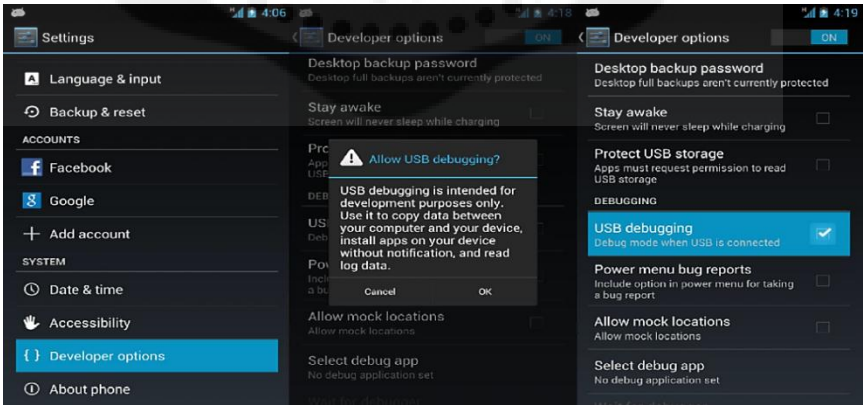
ไม่ว่าการเชื่อมต่อกับบอร์ด IOIO-Q ของอุปกรณ์แอนดรอยด์จะเป็นแบบใช้สายหรือไร้สาย ก็ตามที่อุปกรณ์แอนดรอยด์จะต้องมีการเตรียมความพร้อมเพื่อรองรับการติดต่อกับบอร์ด IOIO-Q เสียก่อนทั้งนี้เนื่องจากการติดต่อของบอร์ด IOIO-Q กับอุปกรณ์แอนดรอยด์เป็นแบบ ADB หรือ Android Debug Bridge ขั้นตอนของการเตรียมการอุปกรณ์แอนดรอยด์ก่อนเชื่อมต่อกับ IOIO-Q ง่ายมากเพียงเลือกเข้าไปยังเมนู Settings > Applications > Development ให้คลิกหรือแตะเลือกเพื่อทำเครื่องหมายถูกที่ช่อง USB Debugging ดังรูปที่ 2.31

2.12.2 การเปิดใช้งาน USB Debugging บนอุปกรณ์แอนดรอยด์เวอร์ชัน 4.2.1 ขึ้นไป

ในอุปกรณ์แอนดรอยด์เวอร์ชัน 4.2.1 เป็นต้นมาระบบได้ทำการซ่อนเมนู Developer options ไว้หากต้องการเปิดจึงต้องมีขั้นตอนเพิ่มเติมโดยเข้าไปที่เมนู About phone > Build number แล้วแตะเลือกเป็นจำนวน 3 ถึง 4 ครั้งตามรูปจนกระทั่งมีข้อความ You are now a developer! แสดงขึ้นมาดังรูปที่ 2.32

รูปที่ 2.32 ตัวอย่างการเปิดเมนู Developer options บนอุปกรณ์แอนดรอยด์เวอร์ชัน 4.2.1 ขึ้นไป ที่มา หนังสือ Android กับการเชื่อมต่อวงจรอิเล็กทรอนิกส์

จากนั้นเมนู Developer options ปรากฏขึ้นมาในหน้าเมนู Setting หลักทำการแตะเลือกเข้าไปเพื่อตั้งค่า USB debugging โดยแตะเลือกเพื่อทำเครื่องหมายถูกที่ ช่อง USB Debugging จะแสดงหน้าต่าง Allow USB debugging? ขึ้นมาจากนั้นแตะเลือกปุ่ม OK ก็จะเป็นการเปิดใช้งาน USB Debugging บนอุปกรณ์แอนดรอยด์เวอร์ชั่น 4.2.1 ขึ้นไปที่มีการซ่อนเมนู Developer options เอาไว้ ดังรูปที่ 2.33



รูปที่ 2.33 การเปิดใช้งาน USB Debugging บนอุปกรณ์แอนดรอยด์เวอร์ชัน 4.2.1 ขึ้นไป ที่มา หนังสือ Android กับการเชื่อมต่อวงจรอิเล็กทรอนิกส์

2.12.3 การเชื่อมต่ออุปกรณ์แอนดรอยด์ กับ IOIO-Q แบบใช้สาย microUSB

รูปที่ 2.34 บอร์ด IOIO-Q เมื่อติดตั้งลงบนบอร์ด IOIO-Q Activity

ที่มา หนังสือ Android กับการเชื่อมต่อวงจรอิเล็กทรอนิกส์

2.12.3.1 ติดตั้งบอร์ด IOIO-Q ลงบนบอร์ด IOIO Activity ดังรูปที่ 2.34

2.12.3.2 จ่ายไฟในย่าน +6.5V ถึง +9V ให้กับบอร์ด IOIO-Q เนื่องจากในการเชื่อมต่อผ่านสาย microUSB อุปกรณ์แอนดรอยด์ไม่ได้จ่ายไฟเลี้ยงให้กับบอร์ด IOIO-Q แต่ตัว IOIO-Q เป็นฝ่ายจ่ายไฟเลี้ยงไปยังอุปกรณ์แอนดรอยด์แทน ทั้งยังเป็นการประจุแรงดันให้กับแบตเตอรี่ภายในอุปกรณ์แอนดรอยด์ด้วย

2.12.3.3 ต่อสาย microUSBระหว่างบอร์ด IOIO-Q กับอุปกรณ์แอนดรอยด์ที่แถบ Notification ของอุปกรณ์แอนดรอยด์จะมีการแจ้งว่ากำลงประจุแบตเตอรี่อยู่และมีการเชื่อมต่อ USB Debugging ในกรณีที่ไม่มีการแจ้งต้องตรวจสอบการเปิดใช้งาน USB Debugging ใหม่อีกครั้ง

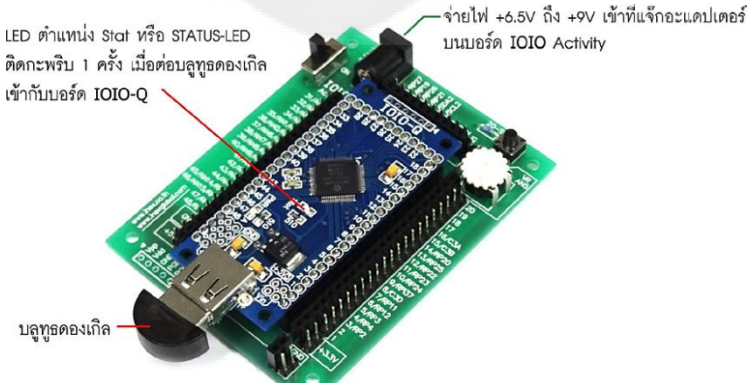
2.12.3.4 ในกรณีที่ใช้บอร์ด IOIO ต้นฉบับ (ไม่ใช่ IOIO-Q) ติดต่อกับอุปกรณ์แอนดรอยด์ไม่ได้ให้หมุนตัวต้านทานปรับค่าได้ที่ตำแหน่ง CHG บนบอร์ด IOIO เพื่อปรับแรงดันไฟฟ้าที่จ่ายให้อุปกรณ์แอนดรอยด์มากขึ้น ในการปรับตัวต้านทานต้องกระทำอย่างระมัดระวังเพื่อป้องกันความเสียหายที่อาจเกิดขึ้นกับตัวบอร์ดได้สังเกตที่บอร์ด IOIO จะมี LED สีเหลืองกะพริบหนึ่งครั้งนั้นหมายความว่า บอร์ด IOIO เชื่อมต่อกับอุปกรณ์แอนดรอยด์เป็นที่เรียบร้อยแล้วแต่ถ้าหากใช้บอร์ด IOIO-Q จะไม่พบปัญหาในลักษณะนี้

นอกจากนั้นในอุปกรณ์แอนดรอยด์บางรุ่นเมื่อใช้การติดต่อผ่าน USB Debugging ไประยะหนึ่งเมื่อต่อกับบอร์ด IOIO ในครั้งต่อไปจะไม่แสดงการเปิดใช้งาน USB Debugging แต่ยังคงใช้งานได้ตามปกติ

รูปที่ 2.35 Notification แจ้งการใช้ USBDebugging บนอุปกรณ์แอนดรอยด์

ที่มา หนังสือ Android กับการเชื่อมต่อวงจรอิเล็กทรอนิกส์

2.12.4 การเชื่อมต่ออุปกรณ์แอนดรอยด์กับ IOIO-Q แบบไร้สายผ่านบลูทูธ



รูปที่ 2.36 ต่อบลูทูธดองเกิลเข้ากับบอร์ด IOIO-Q ที่ติดตั้งบน IOIO Activity

ที่มา หนังสือ Android กับการเชื่อมต่อวงจรอิเล็กทรอนิกส์

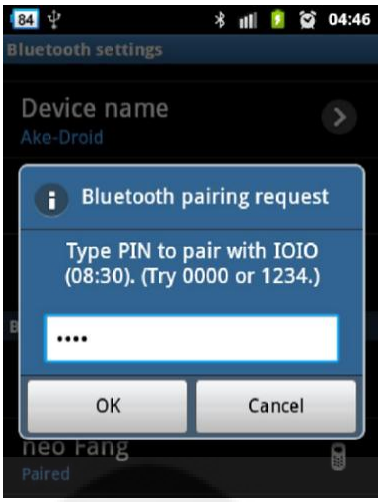
2.12.4.1 ต่อบลูทูธดองเกิลเข้ากับบอร์ด IOIO-Q ดังรูปที่ 2.38 จะเห็น LED แสดงสถานะ (STATUS-LED) ติดกะพริบหนึ่งครั้งเป็นการแจ้งให้ทราบว่า IOIO-Q ติดต่อกับ บลูทูธดองเกิลได้แล้ว

2.12.4.2 ที่อุปกรณ์แอนดรอยด์ทำการเลือกไปที่ Setting >Wireless and Networks > Bluetooth Settingsเลือกเครื่องหมายถูกที่ Bluetooth เพื่อเปิดการใช้งานบลูทูธรอสักครู่เพื่อให้อุปกรณ์ค้นหาสัญญาณจากบลูทูธดองเกิลหากพบจะแสดงเป็นชื่ออุปกรณ์นำหน้าด้วย IOIO แล้วต่อด้วยหมายเลขประจำตัวของบลูทูธดองเกิลจากนั้นทำการจับคู่ (pairing) กับบลูทูธดองเกิล

รูปที่ 2.37 การค้นหาสัญญาณบลูทูธดองเกิล

ที่มา หนังสือ Android กับการเชื่อมต่อวงจรอิเล็กทรอนิกส์

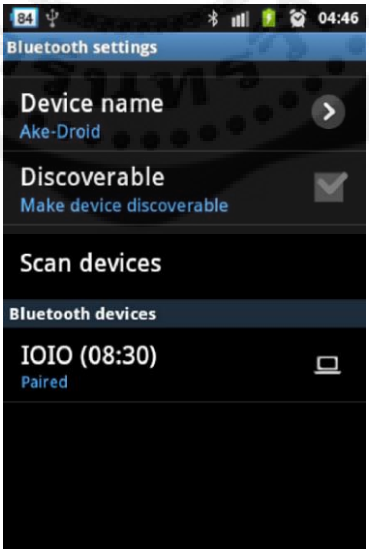
2.12.4.3 ในการจับคู่สัญญาณกับบลูทูธดองเกิลระบบจะให้ใส่เลขรหัสของการจับคู่ ซึ่งก็คือ 4545 จากนั้นเลือกปุ่ม OK



รูปที่ 2.38 การใส่เลขรหัสของการจับคู่สัญญาณกับบลูทูธดองเกิล

ที่มา หนังสือ Android กับการเชื่อมต่อวงจรอิเล็กทรอนิกส์

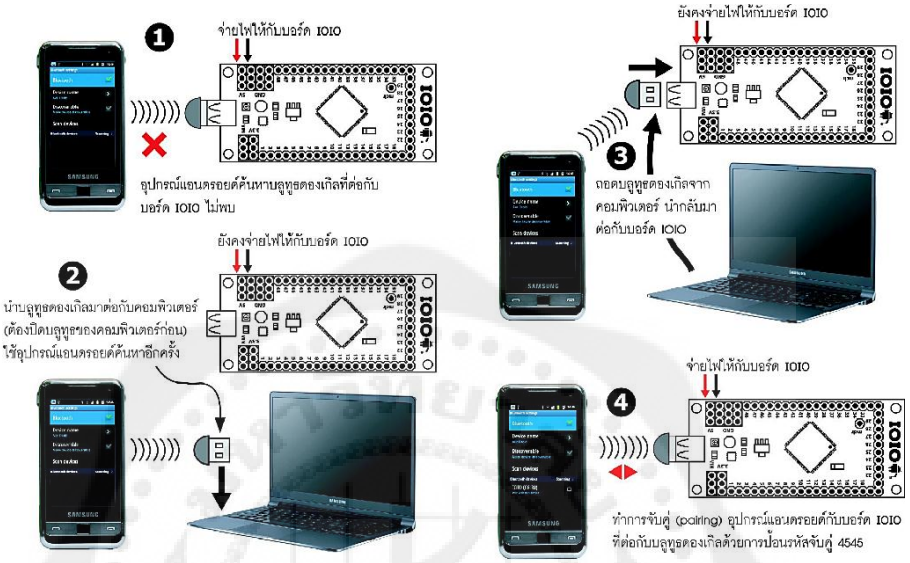
2.12.4.4 ถ้าเชื่อมต่อได้แล้วจะปรากฏข้อความว่า Pairedแต่อุปกรณ์แอนดรอยด์บางรุ่นจะแสดงข้อความ Paired but not connectedแทนซึ่งก็หมายความว่าเชื่อมต่อได้แล้วเช่นกัน



รูปที่ 2.39 อุปกรณ์แอนดรอยด์ปรากฏข้อความแสดงการเชื่อมต่อได้

ที่มา หนังสือ Android กับการเชื่อมต่อวงจรอิเล็กทรอนิกส์

2.12.5 กรณีที่อุปกรณ์แอนดรอยด์ค้นหาสัญญาณของบลูทูธดองเกิลไม่พบ มีขั้นตอนการแก้ไขปัญหาดังนี้



รูปที่ 2.40 การแก้ปัญหาแอนดรอยด์ค้นหาบอร์ด IOIO-Q ที่ต่อกับบลูทูธดองเกิลไม่พบ
ที่มา หนังสือ Android กับการเชื่อมต่อวงจรอิเล็กทรอนิกส์

2.12.5.1 ถอดบลูทูธดองเกิลออกจากบอร์ด IOIO-Q

2.12.5.2 นำบลูทูธดองเกิลไปต่อกับคอมพิวเตอร์แทนถ้าคอมพิวเตอร์ที่ใช้ยังไม่ได้ติดตั้งไดรเวอร์สำหรับบลูทูธดองเกิลต้องทำการติดตั้งให้เรียบร้อยก่อนปิดบลูทูธที่มากับคอมพิวเตอร์ (ถ้ามี)

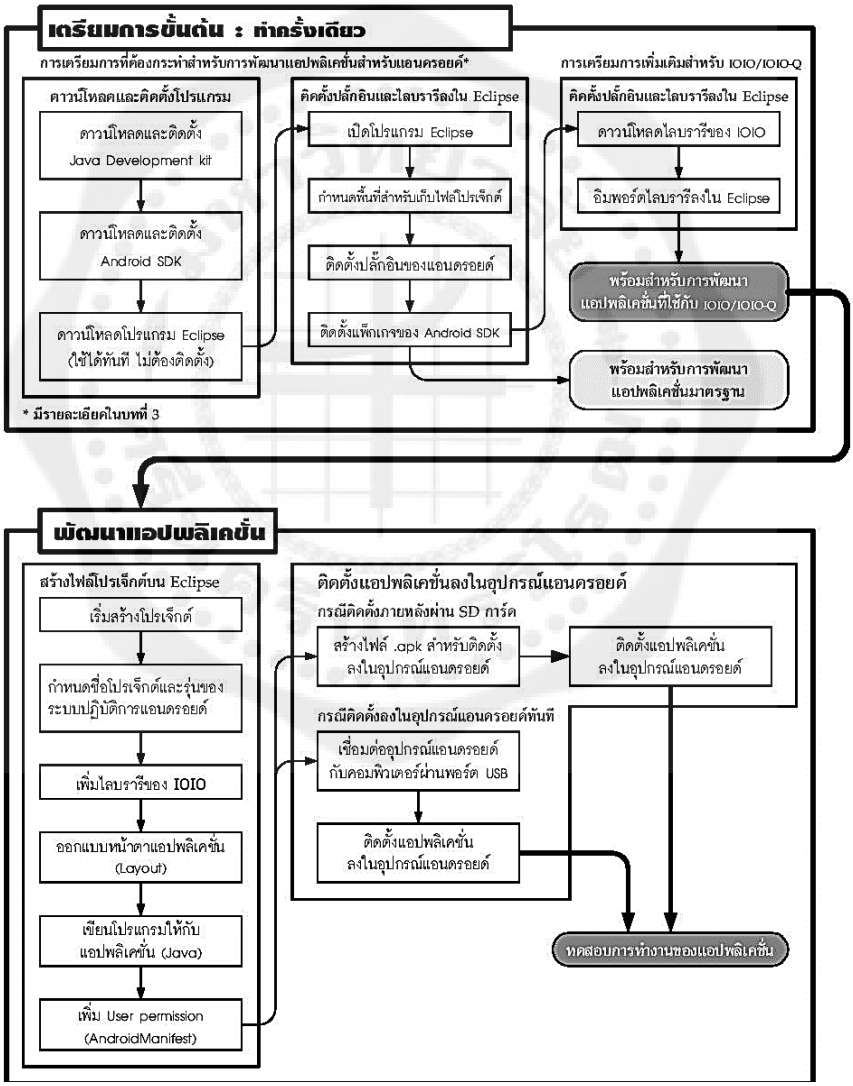
2.12.5.3 จากนั้นใช้อุปกรณ์แอนดรอยด์ค้นหาบลูทูธดองเกิลอีกครั้งหากไม่มีอะไรผิดพลาดจะพบชื่อของอุปกรณ์บลูทูธดองเกิลนี้เป็นชื่อคอมพิวเตอร์ที่เชื่อมต่ออยู่ในกรณีที่อุปกรณ์แอนดรอยด์ตัวนั้น ๆ เคยเชื่อมต่อบลูทูธกับคอมพิวเตอร์ตัวที่ใช้งานนี้มาก่อนให้ยกเลิกการจับคู่เสียก่อน

2.12.5.4 เมื่ออุปกรณ์แอนดรอยด์ค้นหาบลูทูธดองเกิลพบแล้วอย่าเพิ่งจับคู่ให้ถอดบลูทูธดองเกิลออกจากคอมพิวเตอร์แล้วนำไปต่อกับบอร์ด IOIO-Q ก่อน (ทุกขั้นตอนในหัวข้อนี้

ต้องดำเนินการอย่างต่อเนื่องและต้องจ่ายไฟเลี้ยงแก่บอร์ด IOIO-Q ตลอดเวลา) จากนั้นจึงดำเนินการจับคู่กับบลูทูธดองเกิลอีกครั้งในระหว่างการเชื่อมต่อชื่อของบลูทูธดองเกิลยังคงเป็นชื่อคอมพิวเตอร์อยู่เมื่อจับคู่เสร็จแล้วก็จะเปลี่ยนเป็นชื่อของบอร์ด IOIO-Q ในภายหลัง

ในรูปที่ 2.40 แสดงขั้นตอนโดยสรุปของการแก้ปัญหาในกรณีที่อุปกรณ์แอนดรอยด์ค้นหาบอร์ด IOIO-Q ที่ต่อกับบลูทูธดองเกิลไม่พบ

2.12.6 การสร้างแอปพลิเคชัน



รูปที่ 2.41 แผนภาพแสดงขั้นตอนการพัฒนาแอปพลิเคชันสำหรับอุปกรณ์แอนดรอยด์
ที่มา หนังสือ Android กับการเชื่อมต่อวงจรอิเล็กทรอนิกส์

การสร้างโปรเจ็กต์ใน Eclipse ก็คือ การสร้างพื้นที่สำหรับสร้างแอปพลิเคชันเพราะแอปพลิเคชันแต่ละตัวไม่ได้มีเพียงไฟล์เดียวแต่จะมีไฟล์ย่อยมากมายตามความซับซ้อนของโปรแกรมรวมถึงการเก็บไฟล์ภาพหรือเสียงที่ใช้ในแอปพลิเคชันด้วยไฟล์ทั้งหมดได้รับการเก็บแยกไว้ให้เป็นกลุ่มเมื่อต้องการทำแอปพลิเคชันไปเผยแพร่หรือติดตั้งจึงนำข้อมูลในโปรเจ็กต์นั้น ๆ มาทำให้เป็นไฟล์หนึ่งไฟล์ที่มีนามสกุลเป็น .apk ในภายหลัง เนื่องจากการสร้างแอปพลิเคชันสำหรับอุปกรณ์แอนดรอยด์มีขั้นตอนพอสมควร จึงได้ทำการสรุปเป็นแผงผังถึงกระบวนการต่าง ๆ ดังรูป ที่ 2.41 โดยแบ่งออกเป็น 2 กระบวนการหลักคือ

1. การเตรียมการขั้นต้นซึ่งกระทำเพียงครั้งเดียวหากไม่มีการอัพเกรดเวอร์ชันของซอฟต์แวร์โดยในขั้นตอนนี้ประกอบด้วยการดาวน์โหลดและติดตั้งโปรแกรมที่ใช้ในการพัฒนาซึ่งส่วนใหญ่ได้อธิบายไปแล้ว ที่จะมีเพิ่มเติมในบทนี้ก็คือ การดาวน์โหลดและนำเข้าหรือ อิมพอร์ตไลบรารีของIOIO-Q

2. การพัฒนาแอปพลิเคชันกระบวนการนี้มีขั้นตอนมากพอสมควรซึ่งต้องดำเนินการตามลำดับอย่างเคร่งครัดในกระบวนการนี้ประกอบด้วย การเริ่มต้นสร้างไฟล์โปรเจ็กด้วย Eclipse กำหนดชื่อและเวอร์ชันของระบบปฏิบัติการแอนดรอยด์ที่ต้องการให้แอปพลิเคชันที่สร้างขึ้นนี้ทำงานได้ เพิ่มไลบรารีของ IOIO-Q เพื่อใช้เชื่อมต่อกับบอร์ด IOIO-Q ออกแบบหน้าตาของแอปพลิเคชันเขียนโปรแกรมภาษาจาวาซึ่งเป็นโปรแกรมหลักของแอปพลิเคชันและการเพิ่มเติมโปรแกรมเพื่อสร้างการเข้าถึงระบบการเชื่อมต่อของอุปกรณ์แอนดรอยด์ผ่านพอร์ต USB หรือบลูทูธ (User permission) ไปจนถึงการติดตั้งแอปพลิเคชันที่พัฒนาแล้วลงในอุปกรณ์แอนดรอยด์เพื่อทดสอบการทำงาน

บทที่ 3

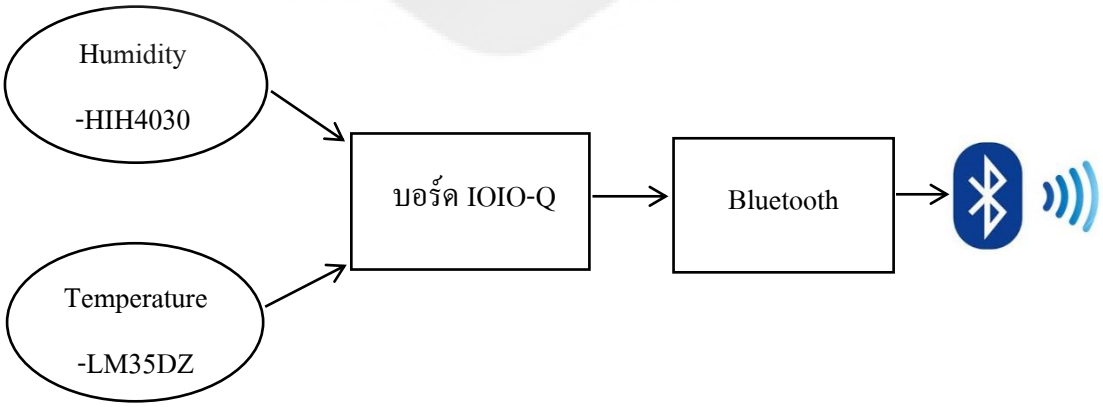
หลักการออกแบบ

ในบทนี้จะกล่าวถึงการสร้างระบบตรวจจับอุณหภูมิและความชื้นสัมพัทธ์แบบไร้สาย โดยใช้หลักการทำงานของการรับและส่งสัญญาณ ระหว่างชุดตรวจจับกับสมาร์ตโฟน ด้วยการเชื่อมต่อสัญญาณบลูทูธมาประยุกต์ใช้ในโครงงานโดยแยกการออกแบบ และการสร้างออกเป็นส่วนหลัก ๆ ได้ดังนี้

3.1 บล็อกไดอะแกรมของระบบตรวจจับอุณหภูมิและความชื้นสัมพัทธ์แบบไร้สาย

3.1.1 การออกแบบและการสร้างส่วนของฮาร์ดแวร์

จากแนวคิดของการสื่อสารด้วยระบบเครือข่ายไร้สายบลูทูธ เราจะนำมาประยุกต์ใช้ในโครงงานนี้ ซึ่งจะทำให้เกิดความสะดวกและรวดเร็วในการติดตั้งหรือเคลื่อนย้าย โดยใช้สัญญาณบลูทูธในการส่งข้อมูลแบบไร้สายส่วนประกอบของภาคส่งจะแสดงดังรูปที่ 3.1 ซึ่งการทำงานจะเริ่มต้นด้วยตัวตรวจวัดอุณหภูมิ (LM35DZ) และความชื้นสัมพัทธ์ (HIH4030) จะส่งค่าแรงดันที่แปรเปลี่ยนไปตามอุณหภูมิและความชื้นที่ตรวจวัดได้ให้กับบอร์ด IOIO-Q โดยบอร์ด IOIO-Q



รูปที่ 3.1 บล็อกไดอะแกรมภาคส่ง

ส่วนประกอบภาครับจะแสดงดังรูปที่ 3.2 วงจรจะทำการรับสัญญาณจากภาคส่ง โดยบอร์ด IOIO-Q จะส่งข้อมูลโดยผ่านระบบเครือข่ายไร้สายแบบบลูทูธไปยังสมาร์ตโฟนระบบปฏิบัติการแอนดรอยด์ที่ได้เขียนแอปพลิเคชันรองรับไว้แล้ว และแอปพลิเคชันนี้จะแจ้งเตือนถ้าหากอุณหภูมิและความชื้นที่ได้รับมีค่ามากหรือน้อยกว่าเงื่อนไขของแอปพลิเคชันที่กำหนดไว้

รูปที่ 3.2 บล็อกไดอะแกรมภาครับ

3.2 การออกแบบระบบตรวจจับอุณหภูมิและความชื้น

3.2.1 การเลือกใช้อุปกรณ์ที่จะนำมาใช้ตรวจวัดอุณหภูมิ

เนื่องจากเครื่องตรวจจับอุณหภูมิและความชื้นสัมพัทธ์ ต้องการวัดอุณหภูมิโดยจะต้องทำงานร่วมกับบอร์ด IOIO-Q และจะต้องให้เอาต์พุตเป็นแรงดันที่มีความสัมพันธ์กับอุณหภูมิ มีความถูกต้องแม่นยำสูง ดังนั้นจึงเลือกใช้ไอซีวัดอุณหภูมิ เบอร์LM35DZ ซึ่งมีคุณสมบัติ ดังนี้

3.2.1.1 ไอซีวัดอุณหภูมิแบบแอนะล็อก

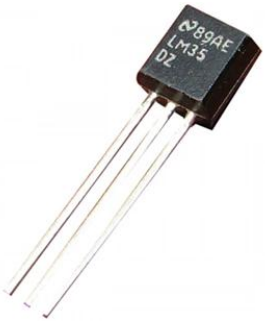
3.1.2.2 มีขาต่อใช้งาน 3 ขา คือ $+V_s$ V_{out}และ GND

3.2.1.3 เป็นเซนเซอร์วัดอุณหภูมิที่มีความถูกต้องแม่นยำสูง มีความผิดพลาดไม่เกิน ±0.5°C

3.2.1.4 ให้เอาต์พุตเป็นแรงดันที่มีความสัมพันธ์กับอุณหภูมิในหน่วยเซลเซียส (°C)

3.2.1.5 มีอัตราขยายเท่ากับ 10 mV/°C สามารถวัดอุณหภูมิในช่วง 0°C ถึง 100°C

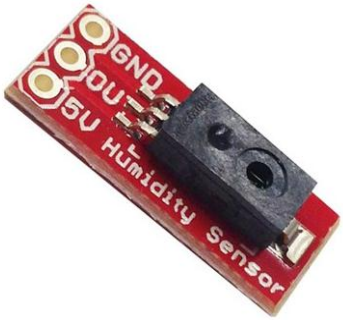
3.2.1.6 แรงดันเอาต์พุตจะอยู่ในช่วง 0 ถึง 1 โวลต์



รูปที่ 3.3 ไอซีวัดอุณหภูมิ LM35DZ

3.2.2 การเลือกใช้อุปกรณ์ที่จะนำมาใช้ตรวจวัดค่าความชื้นสัมพัทธ์

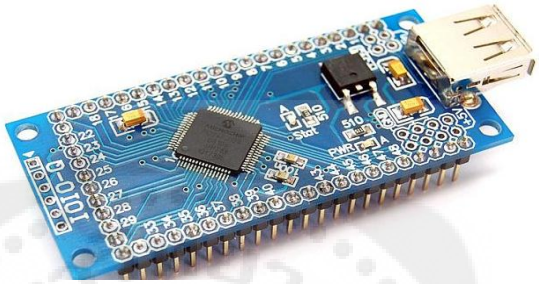
เนื่องจากเครื่องตรวจจับอุณหภูมิและความชื้นสัมพัทธ์ ต้องการวัดความชื้นสัมพัทธ์ โดยจะต้องทำงานร่วมกับบอร์ด IOIO-Q และจะต้องให้ผลการทำงานเป็นแรงดันไฟตรงที่สัมพันธ์กับค่าความชื้นสัมพัทธ์ เพื่อที่จะทำให้ช่วยลดความซับซ้อนในการเขียนโปรแกรม ซึ่งคุณสมบัติดังกล่าวตรงกับโมดูล HIH4030 ที่ให้ค่าออกมาเป็นแรงดันไฟตรงที่เป็นสัญญาณแอนะล็อก มีค่าอยู่ในช่วง 0.958 ถึง4.065 V ที่ค่าแรงดันสูงกว่า +3.3V ซึ่งเท่ากับความชื้นสัมพัทธ์ 80%RH ถึง 100% RH เป็นระดับแรงดันที่สูงเกินกว่าอินพุตแอนะล็อกของบอร์ด IOIO-Q จะรับได้ แต่ที่ค่าความชื้นสัมพัทธ์ ประมาณ 75%RH โมดูล HIH4030 จะให้แรงดันออกมาประมาณ +3.268V ซึ่งห้องปกติ ค่าความชื้นสัมพัทธ์จะมีค่าประมาน 25 ถึง 60%RH ดังนั้นจึงใช้งาน HIH4030 กับ IOIO-Q ได้เลย แต่จะใช้งานได้ในช่วงความชื้นสัมพัทธ์ เท่ากับ 0 ถึง 75%RH



รูปที่ 3.4 โมดูลแผงวงจร HIH4030 ที่พร้อมใช้งาน

3.2.3 การออกแบบหน่วยประมวลผลตัวตรวจวัดอุณหภูมิและความชื้น

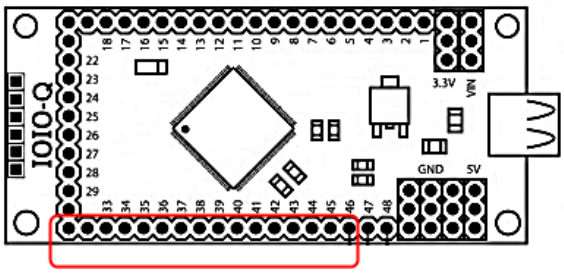
เนื่องจากเอาต์พุตตัวตรวจจับอุณหภูมิและความชื้นสัมพัทธ์ อุปกรณ์ทั้ง 2 จะต้องต่อใช้งานร่วมกับบอร์ด IOIO-Qได้ ดังนั้นจึงเลือกใช้บอร์ด IOIO-Q ซึ่งมีอินพุตอนาล็อกที่สามารถรับแรงดันไฟตรงจากตัวตรวจจับอุณหภูมิและความชื้นสัมพัทธ์ได้



รูปที่ 3.5 บอร์ด IOIO-Q

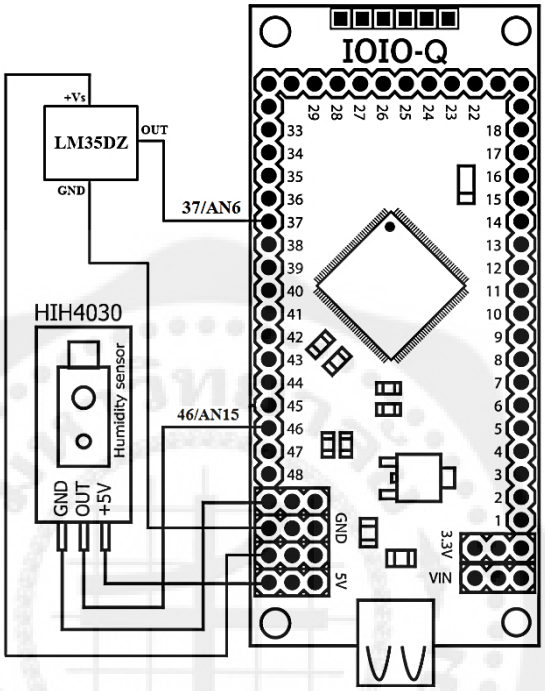
3.2.4 การออกแบบบอร์ด IOIO-Q

เนื่องจากอุปกรณ์ตรวจจับอุณหภูมิและความชื้นสัมพัทธ์เอาต์พุตของอุปกรณ์ทั้ง 2 จะต้องทำการต่อใช้งานร่วมกับกับบอร์ด IOIO-Q ได้โดยตรง ซึ่งสามารถทำได้โดยนำขา V_{out} ของไอซี LM35DZ และขาV_{out}ของHIH4030 มาต่อเข้ากับขาสัญญาณอินพุตแอนะล็อกของบอร์ด IOIO-Q ที่มีให้เลือกใช้งานทั้งสิ้น 16 ขาคือ ขา 31 ถึง 46 โดยแต่ละขาอินพุตรับแรงดันไฟตรงได้ในย่าน 0 ถึง +3.3V ความละเอียดในการแปลงสัญญาณคือ 10 บิต มีค่าข้อมูล 0 ถึง 1,023 รวม 1,024 ค่า เมื่อเทียบกับไฟเลี้ยง +3.3V ค่าข้อมูล 1 บิต จึงเท่ากับแรงดัน 0.003V โดยประมาณ

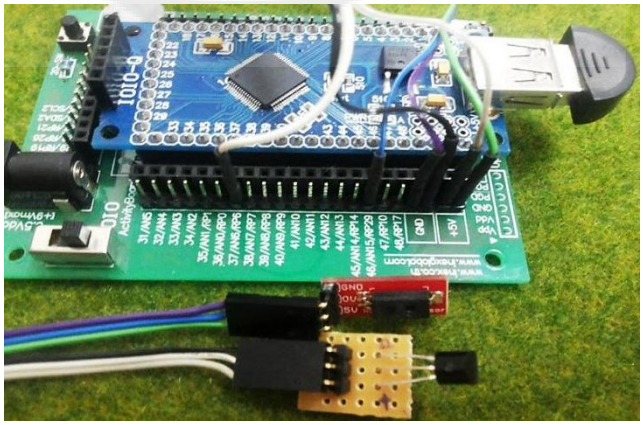


รูปที่ 3.6 การระบุตำแหน่งขาอินพุตแอนะล็อกของบอร์ด IOIO-Q

3.2.5 การออกแบบการเชื่อมต่อระหว่างไอซีวัดอุณหภูมิ(LM35DZ)และโมดูลวัดความชื้นสัมพัทธ์ (HIH4030) เข้ากับบอร์ด IOIO-Q



รูปที่ 3.7 การเชื่อมต่อระหว่างไอซี LM35DZ และโมดูลHIH4030 เข้ากับบอร์ด IOIO-Q



รูปที่ 3.8 การเชื่อมต่ออุปกรณ์ในการต่อใช้งานกับแอปพลิเคชัน

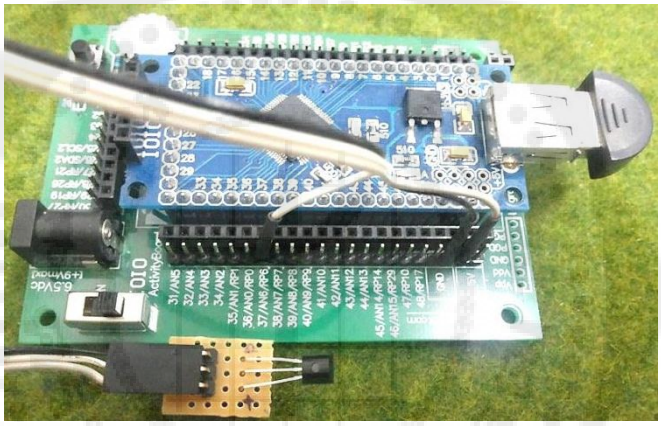
3.2.5.1 การเชื่อมต่อไอซีวัดอุณหภูมิ (LM35DZ)เข้ากับบอร์ด IOIO-Q

ไอซีLM35DZ มีทั้งหมด 3 ขา โดยที่ลักษณะการเชื่อมต่อเป็นดังนี้

- ขา V_{out} ของ ไอซีLM35DZ จะต่อเข้าที่ช่องสัญญาณ 37/AN6/RP6ของบอร์ด IOIO-Q ซึ่งเป็นขาสัญญาณอินพุตแอนะล็อกแบบ ADCที่มีขนาดความละเอียดในการแปลงสัญญาณ 10 บิต มีทั้งหมด 16 ขาและสามารถรับแรงดันได้ 0 ถึง +3.3V

-ขา $+V_s$ ของไอซีLM35DZ จะต่อเข้าที่ช่องสัญญาณ +5V ของบอร์ด IOIO-Q

-ขา GND ของไอซีLM35DZ จะต่อเข้าที่ช่องสัญญาณ GND ของบอร์ดIOIO-Q



รูปที่ 3.9 การเชื่อมต่อไอซีLM35DZ เข้ากับบอร์ด IOIO-Q

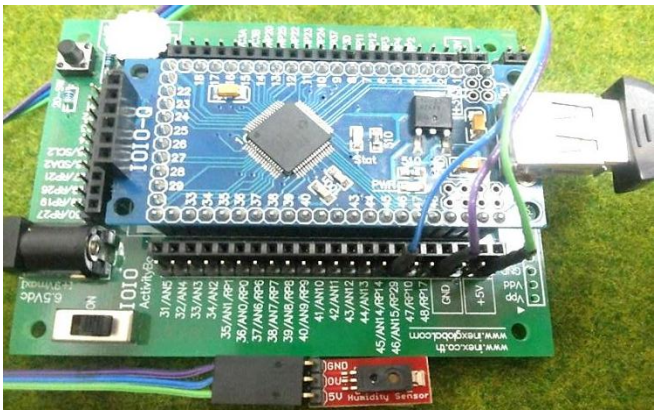
3.2.5.2 การเชื่อมต่อโมดูลวัดความชื้นสัมพัทธ์(HIH4030)เข้ากับบอร์ด IOIO-Q

โมดูล HIH4030 มีทั้งหมด 3 ขา โดยที่ลักษณะการเชื่อมต่อเป็นดังนี้

- ขาV_{out} ของHIH4030 จะต่อเข้าที่ช่องสัญญาณ 46/AN15/RP29 ของบอร์ด IOIO-Q ซึ่งเป็นขาสัญญาณอินพุตแอนะล็อกแบบ ADC ที่มีขนาดความละเอียดในการแปลงสัญญาณ 10 บิต มีทั้งหมด 16 ขาและสามารถรับแรงดันได้ 0 ถึง +3.3V

-ขา $+V_s$ของHIH4030 จะต่อเข้าที่ช่องสัญญาณ +5V ของบอร์ด IOIO-Q

-ขา GND ของHIH4030 จะต่อเข้าที่ช่องสัญญาณ GND ของบอร์ด IOIO-Q



รูปที่ 3.10 การเชื่อมต่อโมดูลHIH4030 เข้ากับบอร์ด IOIO-Q

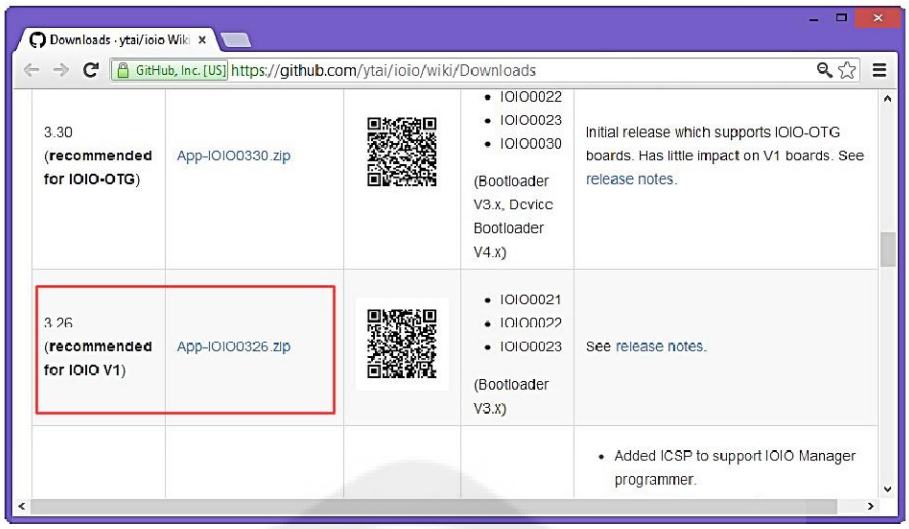
3.3 การออกแบบการทำงานของแอปพลิเคชัน

การสร้างโปรเจกต์ใน Eclipse ก็คือการสร้างพื้นที่สำหรับสร้างแอปพลิเคชัน เพราะแอปพลิเคชันแต่ละตัวไม่ได้มีเพียงไฟล์เดียวแต่จะมีไฟล์ย่อยมากมายตามความซับซ้อนของโปรแกรม ทำให้ขั้นตอนการสร้างแอปพลิเคชันสำหรับอุปกรณ์แอนดรอยด์ มีขั้นตอนพอสมควรจึงได้ทำการสรุปโดยแบ่งออกเป็น 3กระบวนการหลัก คือ

3.3.1 การนำเข้าไฟล์ไลบรารี

การสร้างไฟล์โปรเจกต์เพื่อใช้ในการสร้างแอปพลิเคชันเบื้องต้นเพื่อใช้งานกับบอร์ด IOIO-Q เริ่มต้นจากการนำเข้า หรืออิมพอร์ตไลบรารีของบอร์ด IOIO-Q

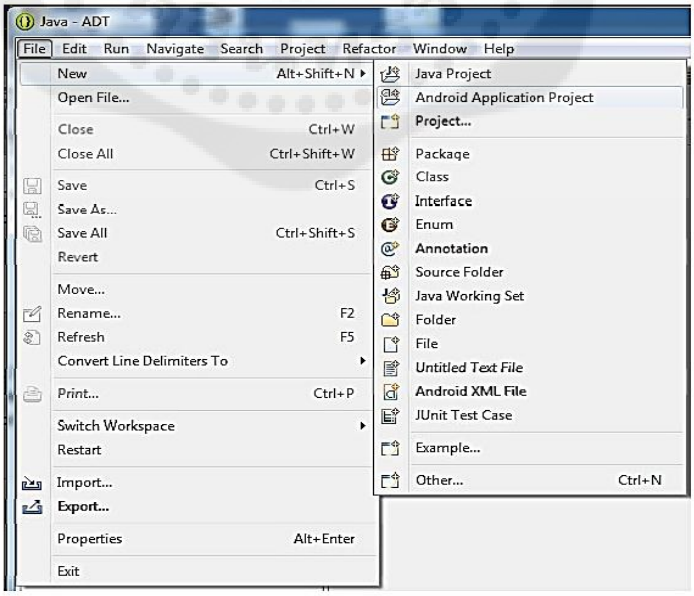
3.3.1.1 ดาวน์โหลดไฟล์ไลบรารีสำหรับบอร์ด IOIO-Q เพื่อนำมาติดตั้งลงในโปรแกรม Eclipse จาก https://github.com/ytai/ioio/wiki/Downloads ให้ดาวน์โหลดเวอร์ชัน 3.26 เนื่องจากเข้ากันได้ดีที่สุดหากดาวน์โหลดเวอร์ชันอื่นต้องแก้ไขข้อผิดพลาดมากและมีขั้นตอนที่ซับซ้อน



รูปที่ 3.11 การดาวน์โหลดไฟล์ไลบรารีสำหรับบอร์ด IOIO-Q

3.3.2 เตรียมไฟล์โปรเจกต์

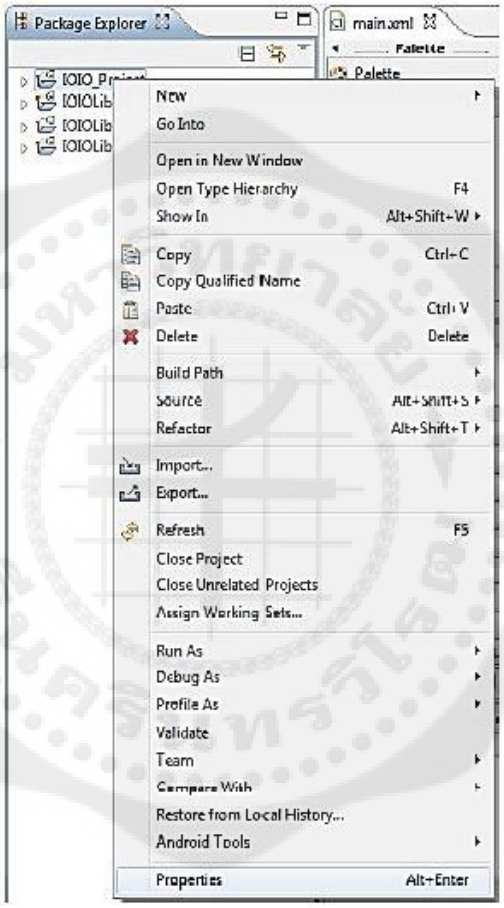
3.3.2.1 ที่หน้าต่างของโปรแกรม eclipse เลือกที่เมนู File > New > Android Application Project



รูปที่ 3.12 ขั้นตอนการเตรียมไฟล์โปรเจกต์

3.3.2.2 จะพบกับหน้าต่าง New Android Application ที่ใช้ตั้งค่าพื้นฐานของแอปพลิเคชัน ที่จะสร้างขึ้น

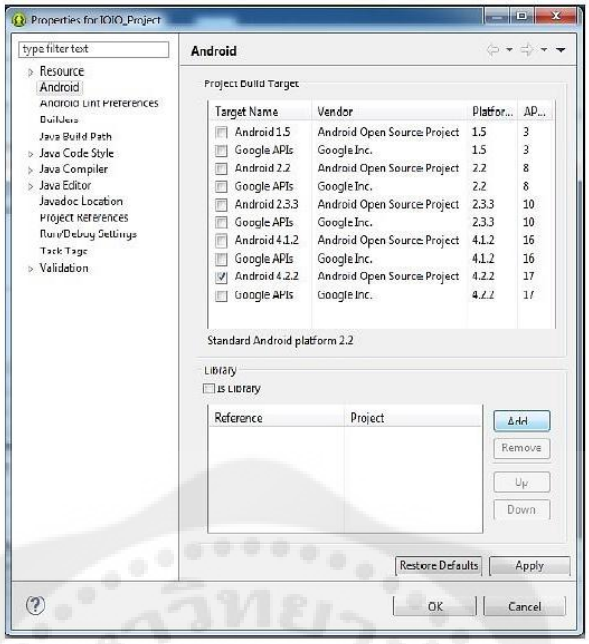
3.3.2.3 สังเกตที่หน้าต่าง Package Explorer จะพบว่ามีโปรเจกต์ที่ได้สร้างขึ้นมาแสดงอยู่ในหน้าต่างแล้วคลิกเมาส์ปุ่มขวาที่IOIO_Projectแล้วเลือกที่ Properties



รูปที่ 3.13 วิธีการสร้างโปรเจกต์พื้นฐานของแอปพลิเคชัน

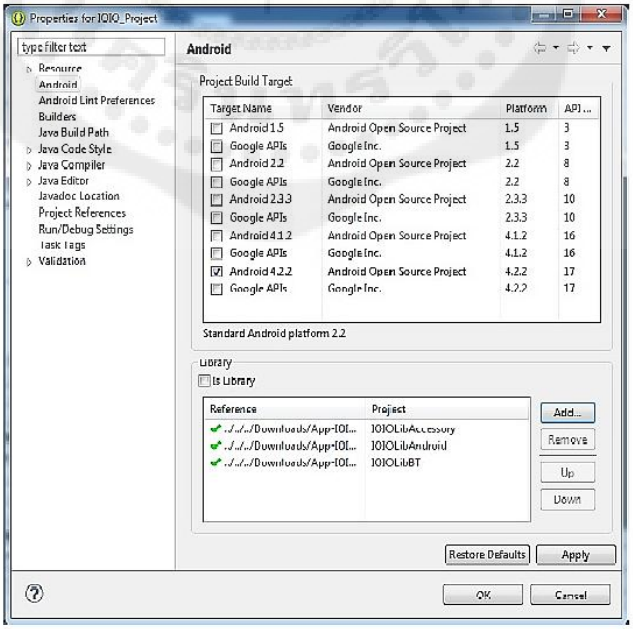
3.3.2.4 เลือกที่ Android แล้วคลิกปุ่ม Add..เพื่อเพิ่มไฟล์ไลบรารีของบอร์ด IOIO ให้กับโปรเจกต์

3.3.2.5 เลือกไฟล์ไลบรารีแล้วคลิกปุ่ม OK



รูปที่ 3.14 วิธีการเลือกไฟล์ไลบรารีของบอร์ด IOIO ให้กับโปรเจกต์

3.3.2.6 เพิ่มไลบรารีให้ครบทั้งหมดจากนั้นคลิกปุ่ม OK เพื่อปิดหน้าต่าง Properties

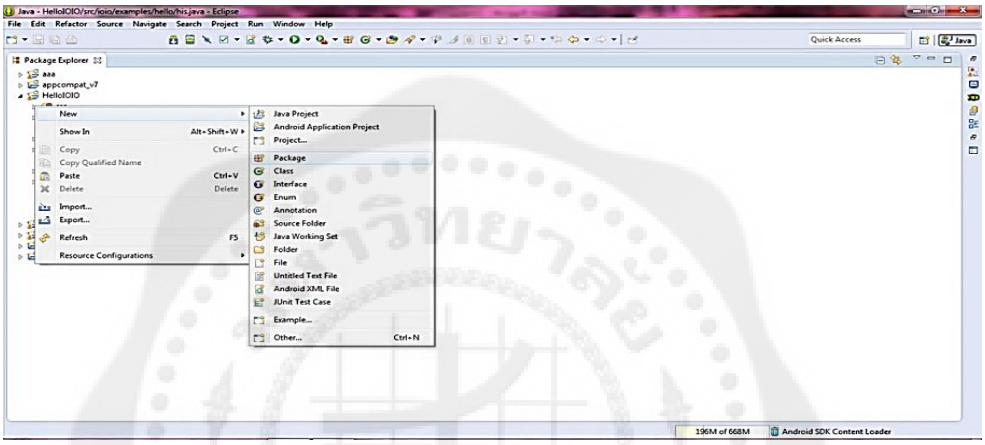


รูปที่ 3.15 วิธีการเพิ่มไฟล์ไลบรารีของบอร์ด IOIO ให้กับโปรเจ็กต์

3.3.3 การสร้างแอปพลิเคชัน

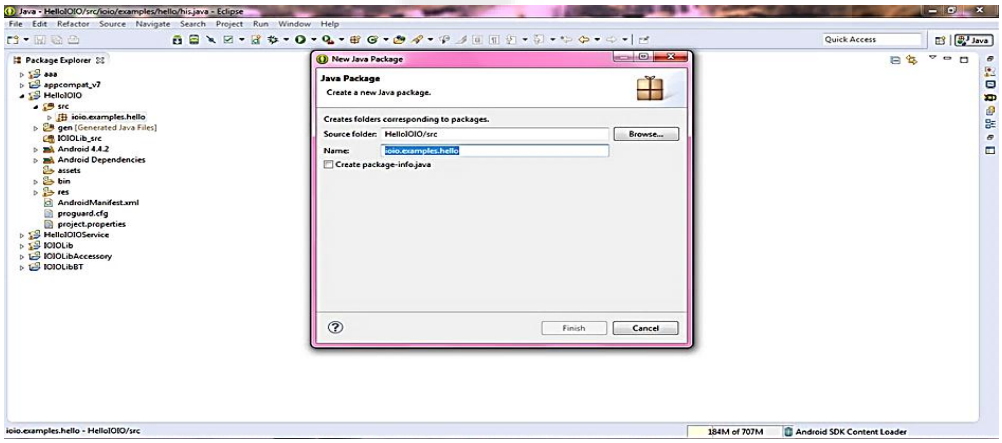
หลังจากที่ได้เตรียมการสร้างไฟล์โปรเจกต์เรียบร้อยแล้วต่อไปเป็นการทดลองสร้างแอปพลิเคชัน เพื่อควบคุมการตรวจจับอุณหภูมิและความชื้นสัมพัทธ์บนบอร์ด IOIO-Q

3.3.3.1 ที่หน้าต่าง Project Explorer ทางซ้ายมือของโปรแกรม Eclipse ให้ดับเบิ้ลคลิกที่ src>New> Package



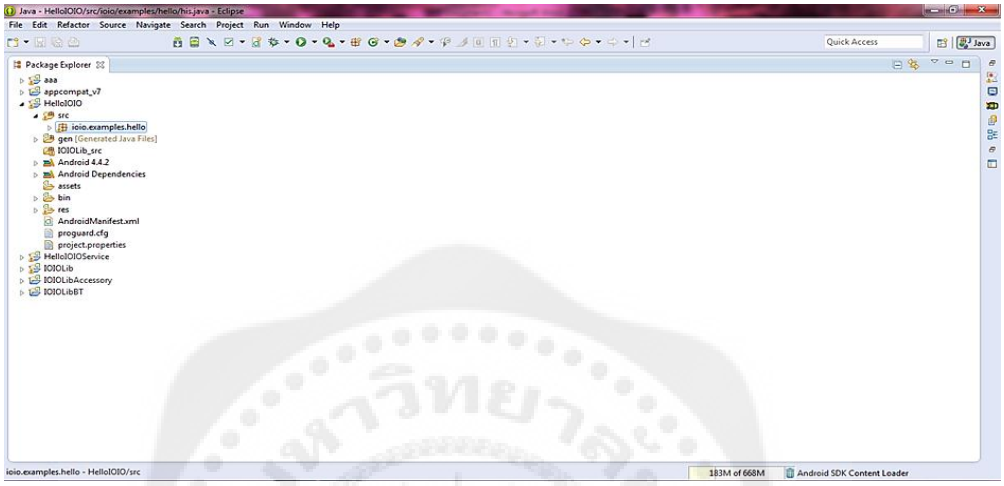
รูปที่ 3.16วิธีการสร้างแพ็กเกจของแอปพลิเคชัน

3.3.3.2 ที่หน้าต่าง New Java Package > ในช่อง Name: ให้ทำการตั้งชื่อเพื่อทำการจัดเก็บแพ็กเกจ Java โดยตั้งชื่อว่า “ioio.examples.hello”



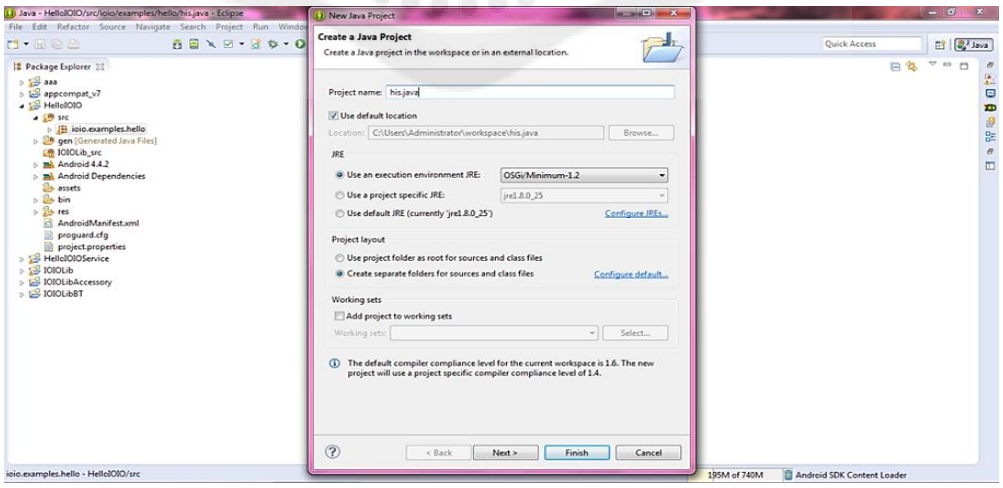
รูปที่ 3.17วิธีการตั้งชื่อเพื่อจัดเก็บแพ็กเกจ Java

3.3.3.3 หลังจากได้ทำการสร้างแพ็กเกจเรียบร้อยแล้วให้ทำการเปิดไฟล์ดับเบิ้ลคลิกที่ src>ioio.examples.hello เพื่อการสร้างโปรเจกต์สำหรับจัดเก็บไฟล์ Java



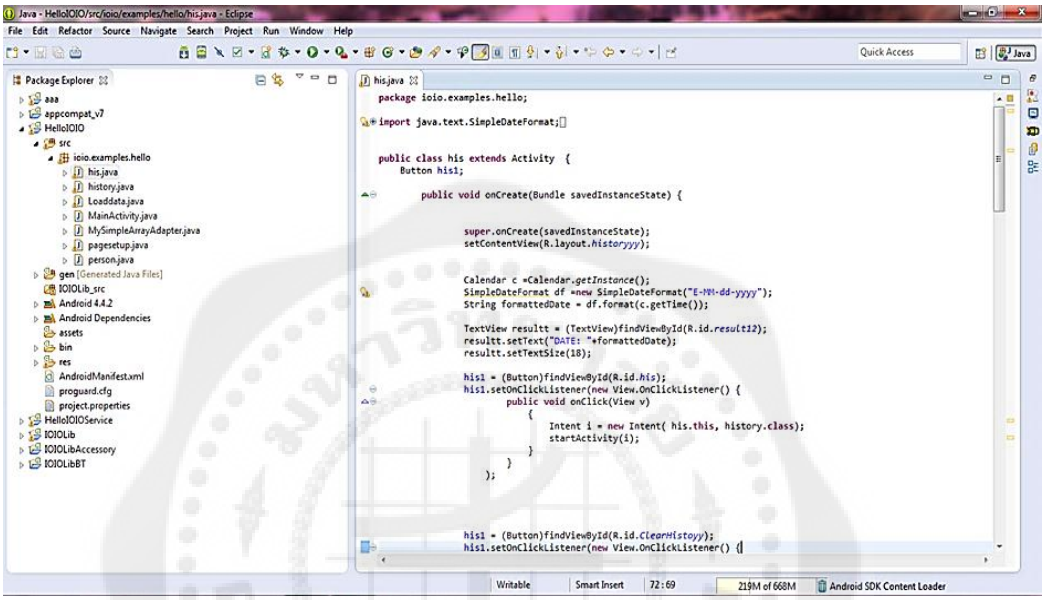
รูปที่ 3.18 วิธีการสร้างโปรเจกต์สำหรับจัดเก็บไฟล์ Java

3.3.3.4 ที่หน้าต่าง New Java Project>ในช่อง Project name:ให้ทำการตั้งชื่อเพื่อทำการจัดเก็บไฟล์ Java โดยตั้งชื่อว่า “his.java” ก็จะได้แพ็กเกจสำหรับจัดเก็บไฟล์ Java



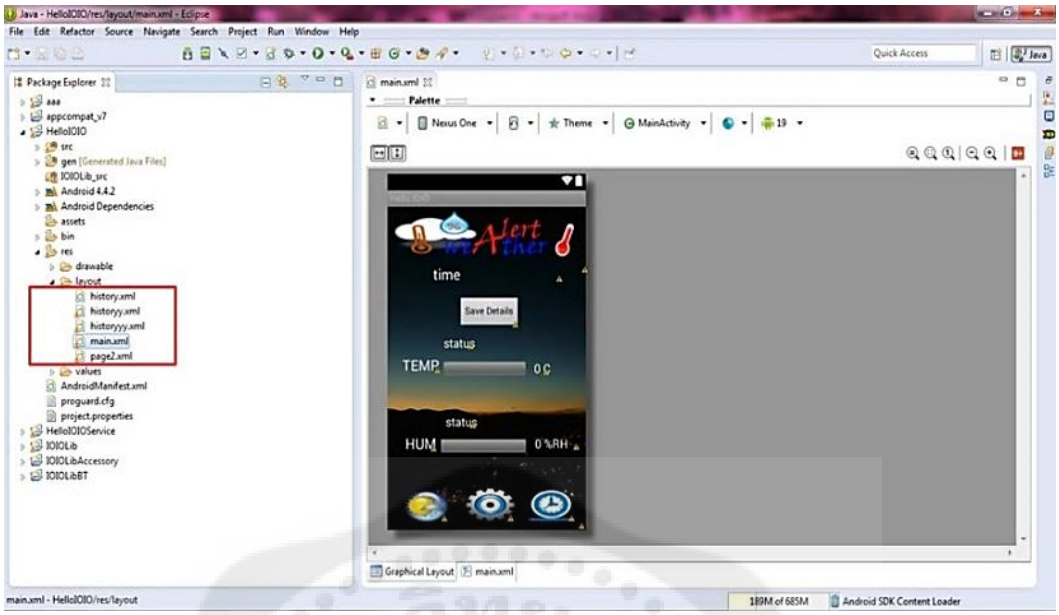
รูปที่ 3.19 วิธีการตั้งชื่อเพื่อจัดเก็บไฟล์ Java

3.3.3.5 หลังจากได้ทำการสร้างโปรเจกต์เรียบร้อยแล้วให้ดับเบิ้ลคลิกไฟล์ his.java ในไดเรกทอรี src>ioio.examples.hello> his.javaเพื่อเปิดไฟล์ขึ้นมาแล้วทำการกำหนดฟังก์ชันการทำงานของแอปพลิเคชันที่เราต้องการ



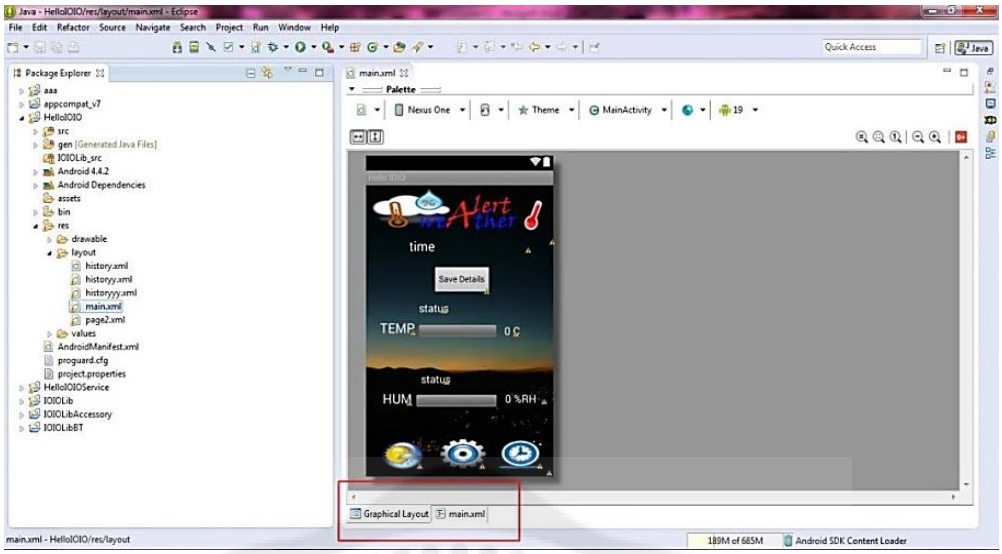
รูปที่ 3.20 วิธีการกำหนดฟังก์ชันการทำงานของแอปพลิเคชัน

3.3.3.6 หน้าต่าง Project Explorer อยู่ทางซ้ายมือของโปรแกรม Eclipseให้เลือกที่ Project name > res > layout > main.xml โดยไฟล์ main.xml จะเป็นไฟล์สำหรับจัดการกับหน้าตาของแอปพลิเคชันในหนึ่งแอปพลิเคชันมีได้มากกว่าหนึ่งไฟล์โดยแอปพลิเคชันที่เราเขียนขึ้นมานั้นมีทั้งหมด 5 ไฟล์ ได้แก่ history.xml, historyy.xml, historyyy.xml, main.xml และ page2.xml

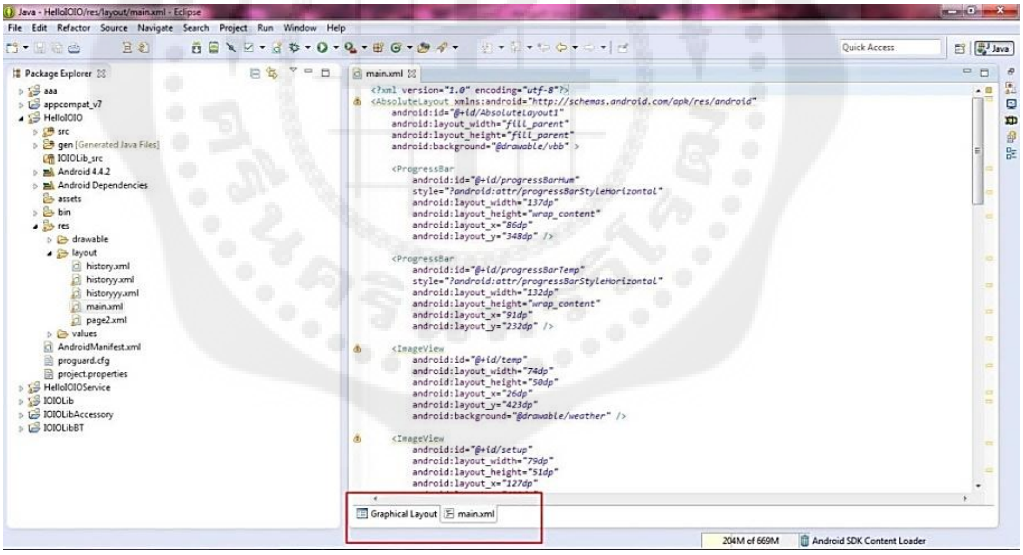


รูปที่ 3.21 หน้าต่าง main.xml

3.3.3.7 เมื่อเปิดขึ้นมาก็จะพบกับหน้าต่างสำหรับออกแบบหน้าตาแอปพลิเคชันที่เราได้ทำการออกแบบเอาไว้แล้วซึ่งในหน้านี้ออกแบบและปรับปรุงการแสดงผลได้ 2 แบบคือ แบบแสดงภาพกราฟิก (Graphical Layout) โดยเลือกอุปกรณ์ต่าง ๆ จากช่อง Palette ทางซ้ายมือมาวางลงในพื้นที่หน้าจอได้ทันที อีกแบบหนึ่งคือ แบบคำสั่ง XML ซึ่งเป็นการสร้างภาพกราฟิกโดยใช้คำสั่ง XMLโดยทั้งสองคำสั่งนี้มีความสัมพันธ์กันเมื่อสร้างใน Graphical Layout แล้วไปที่หน้าคำสั่ง XMLจะพบว่า มีคำสั่งเพิ่มขึ้นมาตามที่สร้างไว้ใน Graphical Layout หากมีการแก้ไขคำสั่งXML ที่หน้า Graphical Layout ก็จะเปลี่ยนไปตามที่แก้ไขด้วย

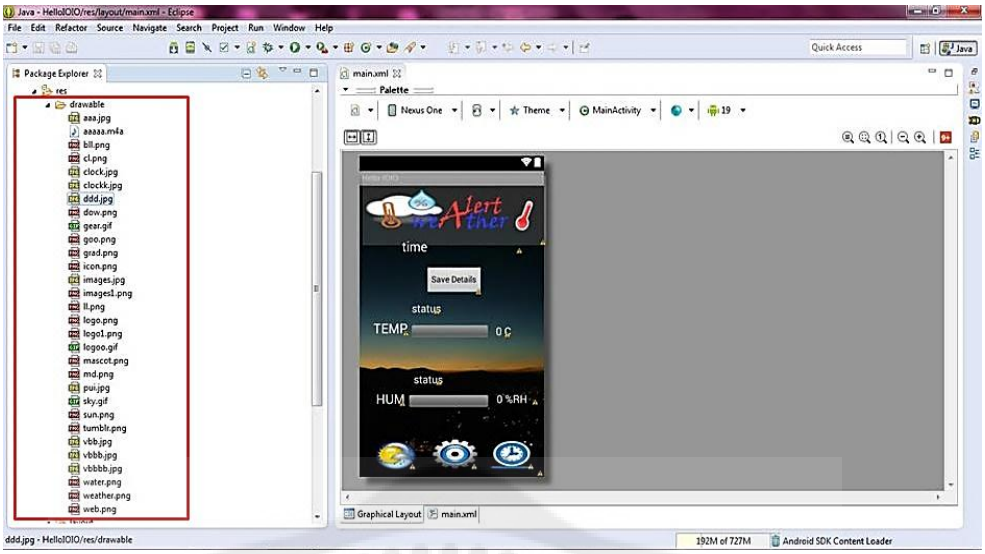


รูปที่ 3.22 การเขียนแบบแสดงภาพกราฟิก (Graphical Layout)

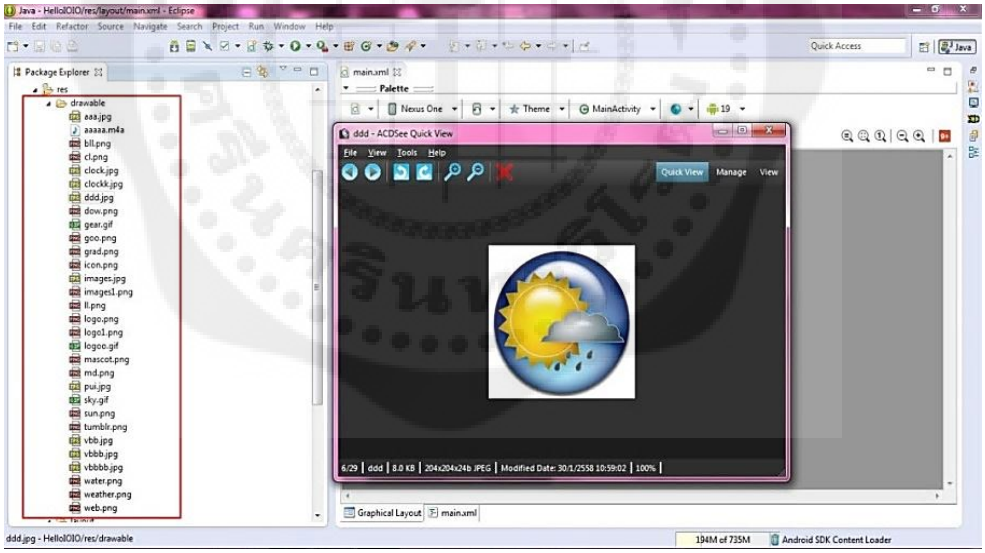


รูปที่ 3.23 การเขียนแบบคำสั่ง XML

3.3.3.8 res>drawable> คลิกที่รูปที่เราจะนำมาใช้ในการทำหน้า layout โดย drawable เป็นส่วนของการออกแบบหน้าตาแอปพลิเคชันนี้ เหมือนเป็นแหล่งเก็บรูปภาพที่เรานำมาตกแต่งในการทำหน้า layout



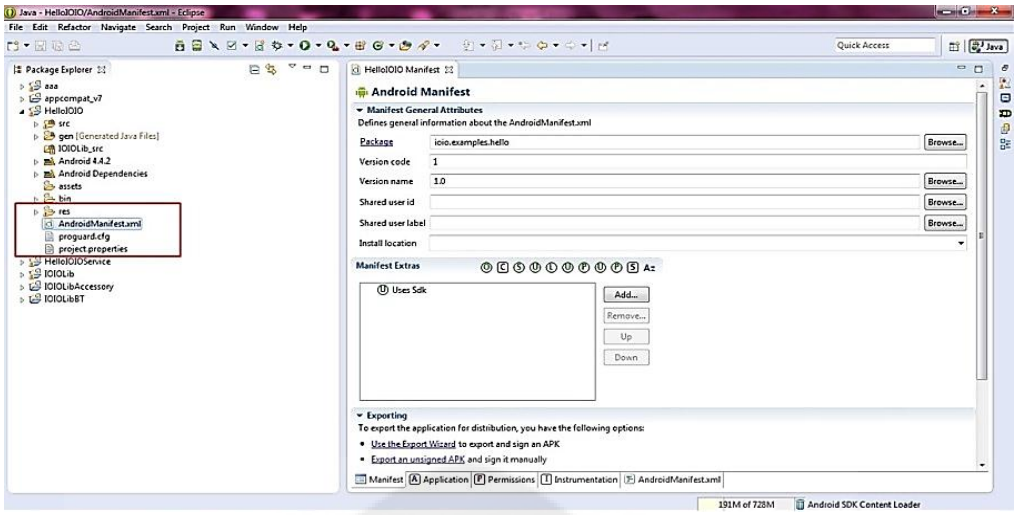
รูปที่ 3.24 ส่วนย่อยต่าง ๆ ที่นำมาออกแบบแอปพลิเคชัน



รูปที่ 3.25 ตัวอย่างรูปภาพที่อยู่ในdrawable

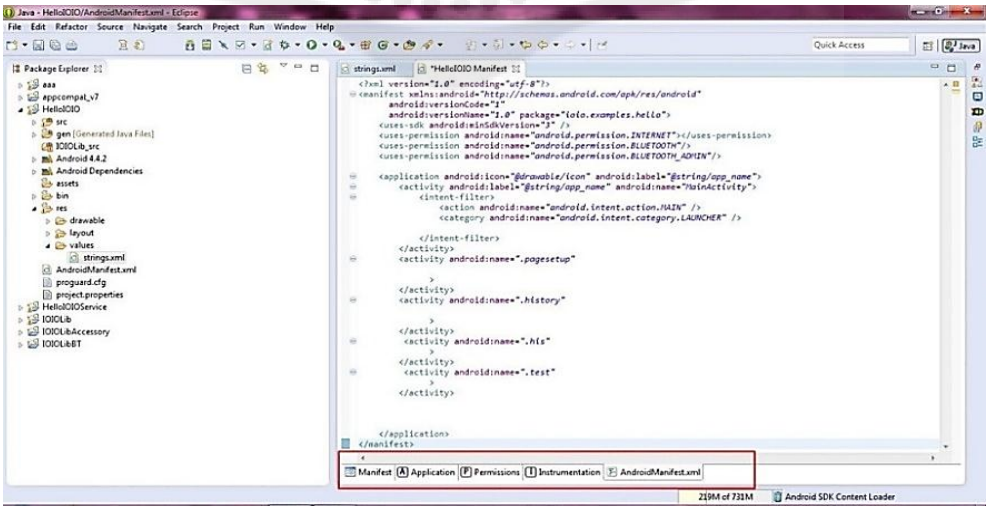
3.3.3.9 การแก้ไขไฟล์ AndroidMainfest.xml

หน้าต่าง Package Explorer ให้เลือกที่ HelloIOIO > AndroidManifest.xml หน้าต่างของ AndroidManifest.xml จะแสดงขึ้นมา



รูปที่ 3.26 การแก้ไขไฟล์ AndroidMainfest.xml

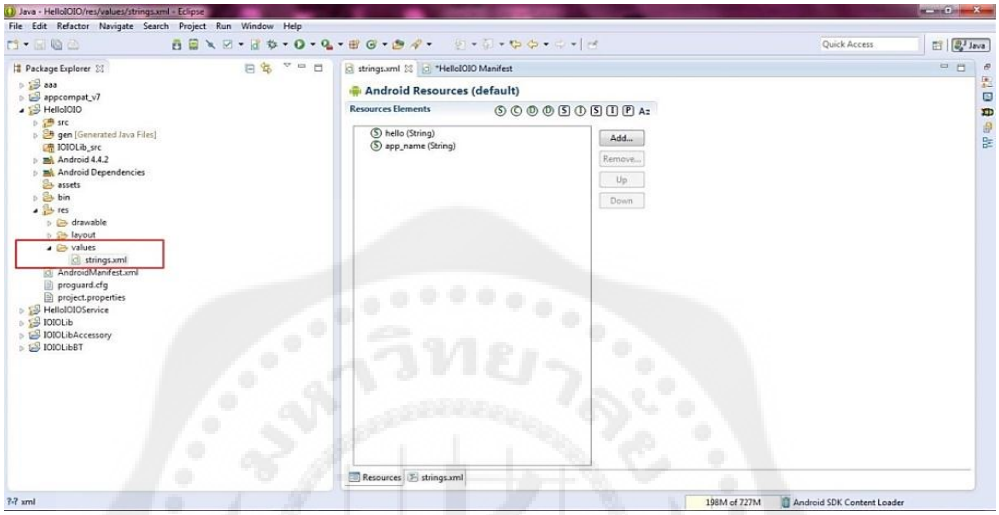
โดยการแก้ไขไฟล์ AndroidMainfest.xml เป็นไฟล์ที่ใช้กำหนดค่าต่าง ๆ ให้กับแอปพลิเคชันเป็นการกำหนดให้แอปพลิเคชันมีขีดความสามารถต่าง ๆ เช่นรูปไอคอนหรือชื่อแอปพลิเคชันการอนุญาตให้แอปพลิเคชันสามารถใช้งานอินเทอร์เน็ตได้โดยเพิ่ม permission เป็นต้น จะสังเกตเห็นว่ามีแถบอยู่ด้านล่างจำนวนหลายแถบให้เลือกไปที่แถบ AndroidManifest.xml (แถบสุดท้าย) ดังรูปที่ 3.27 ซึ่งเป็นการกำหนดค่าด้วยคำสั่ง XML



รูปที่ 3.27 การกำหนดขีดความสามารถของแอปพลิเคชัน

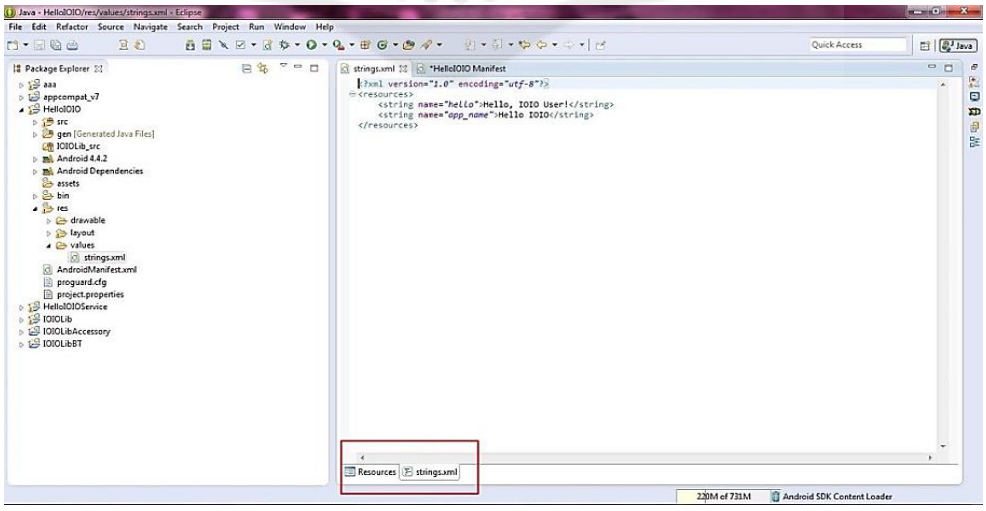
3.3.3.10 การแก้ไขไฟล์ string.xml

หน้าต่าง Package Explorer ให้เลือกที่ HelloIOIO> res > values > string.xml หน้าต่างของ string.xml จะแสดงขึ้นมา ดังรูปที่ 3.28



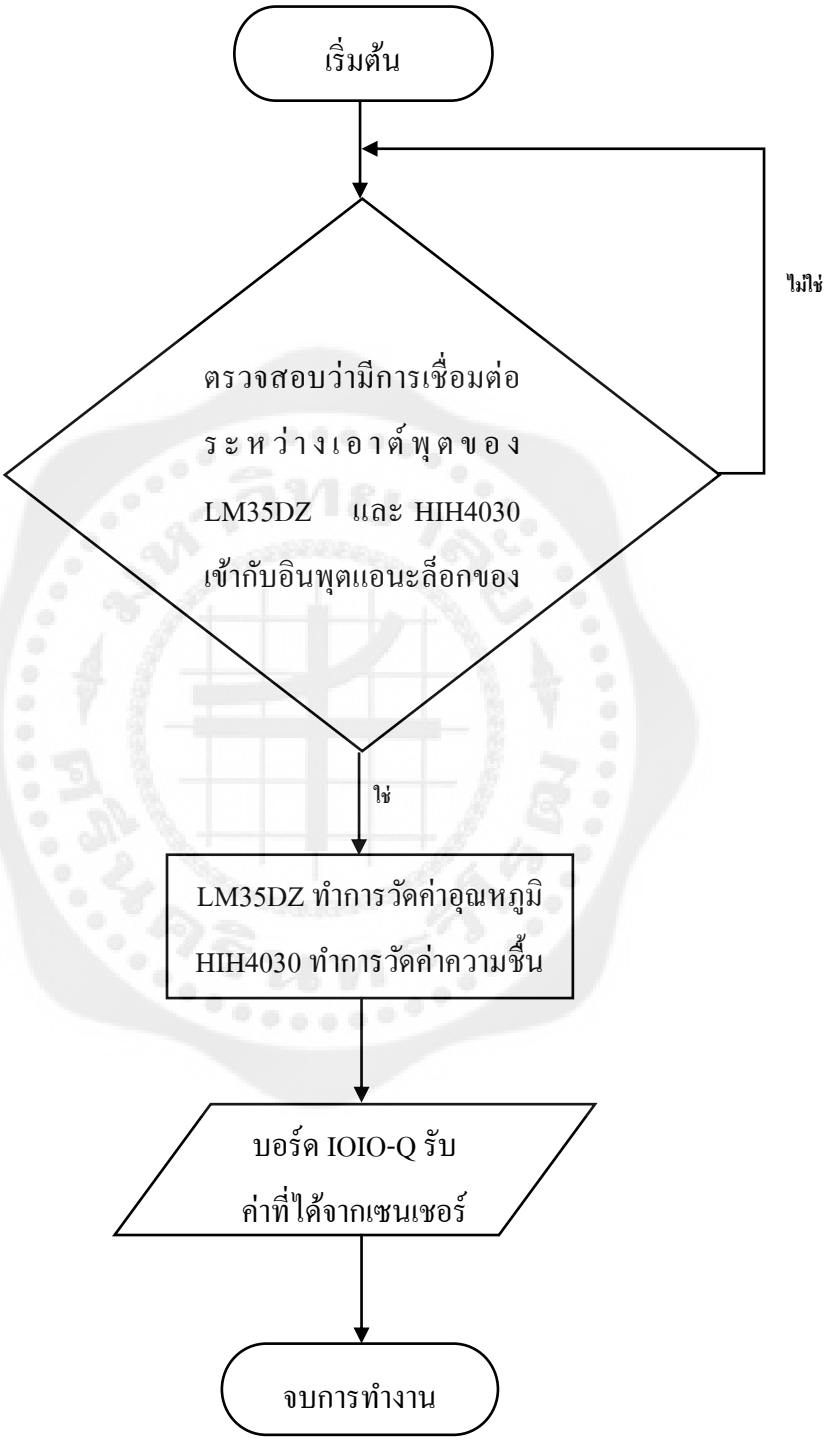
รูปที่ 3.28 การแก้ไขไฟล์ string.xml

โดยในส่วนนี้เป็นส่วนเพิ่มเติมสำหรับช่วยในการอ้างอิงถึงรายละเอียดต่าง ๆ ของลักษณะประจำข้อมูลให้มีความสะดวกในการเรียกใช้งาน .xml



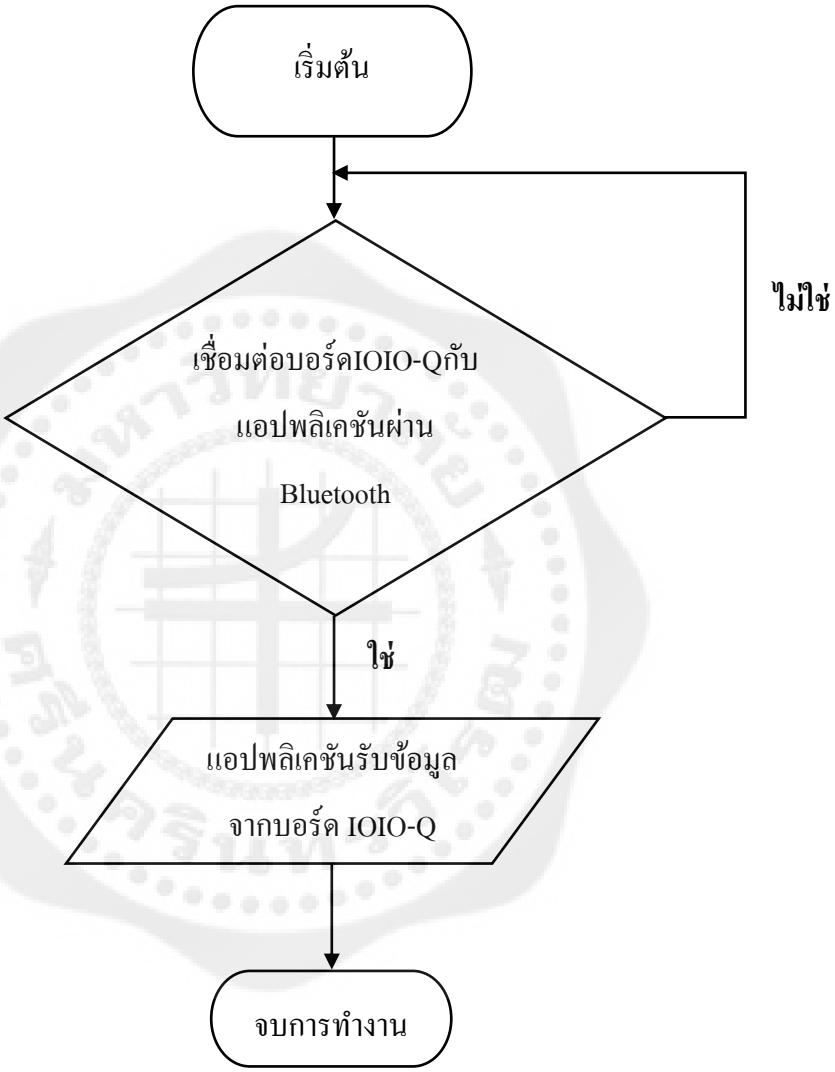
รูปที่ 3.29 การอ้างอิงรายละเอียดต่าง ๆ ของลักษณะประจำข้อมูล

3.4 แผนผังการทำงานของเซนเซอร์วัดอุณหภูมิและความชื้น



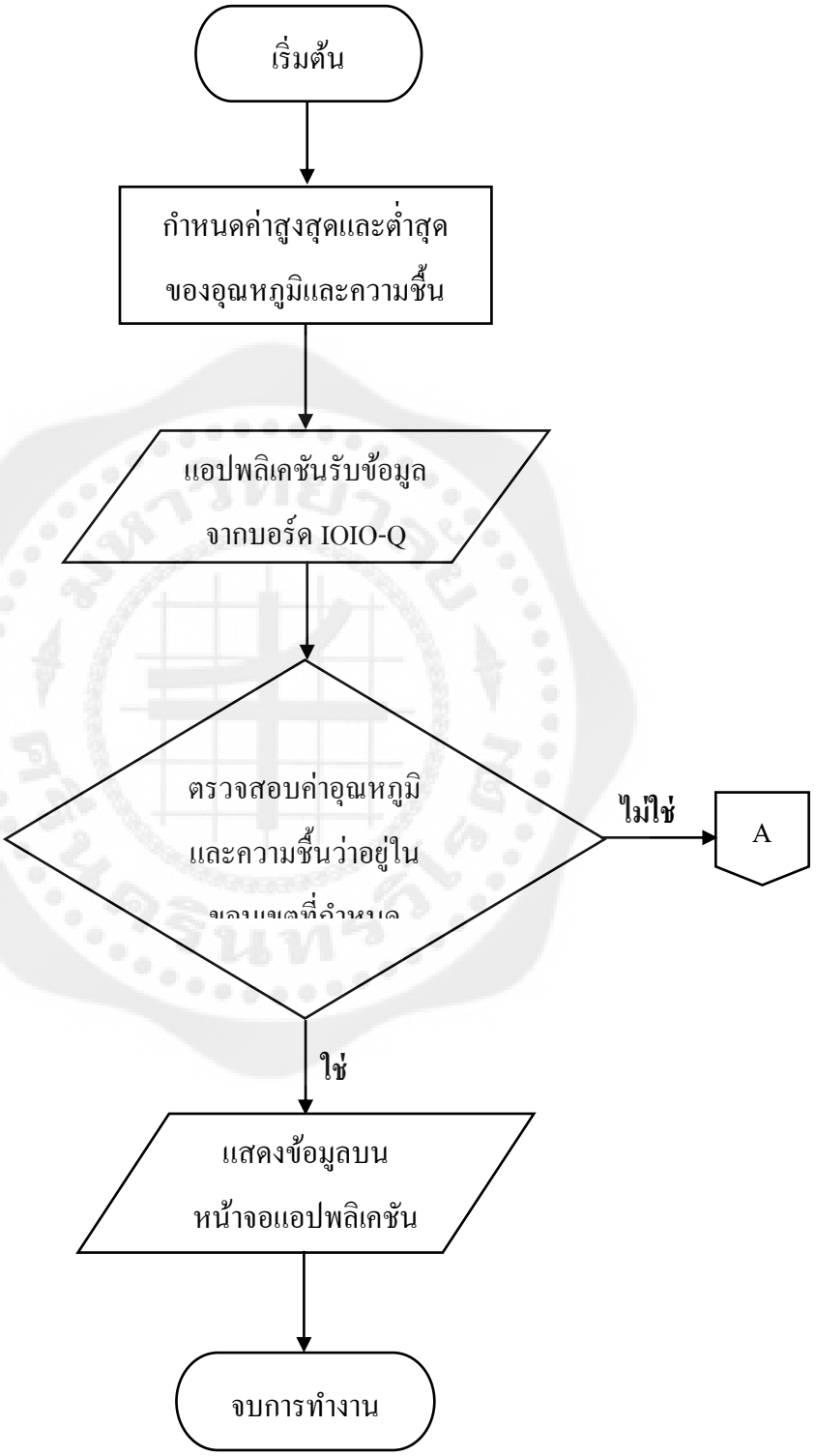
รูปที่ 3.30 แผนผังการทำงานของเซนเซอร์วัดอุณหภูมิและความชื้น

3.5 แผนผังการทำงานในส่วนของบอร์ด IOIO-Q กับแอปพลิเคชัน

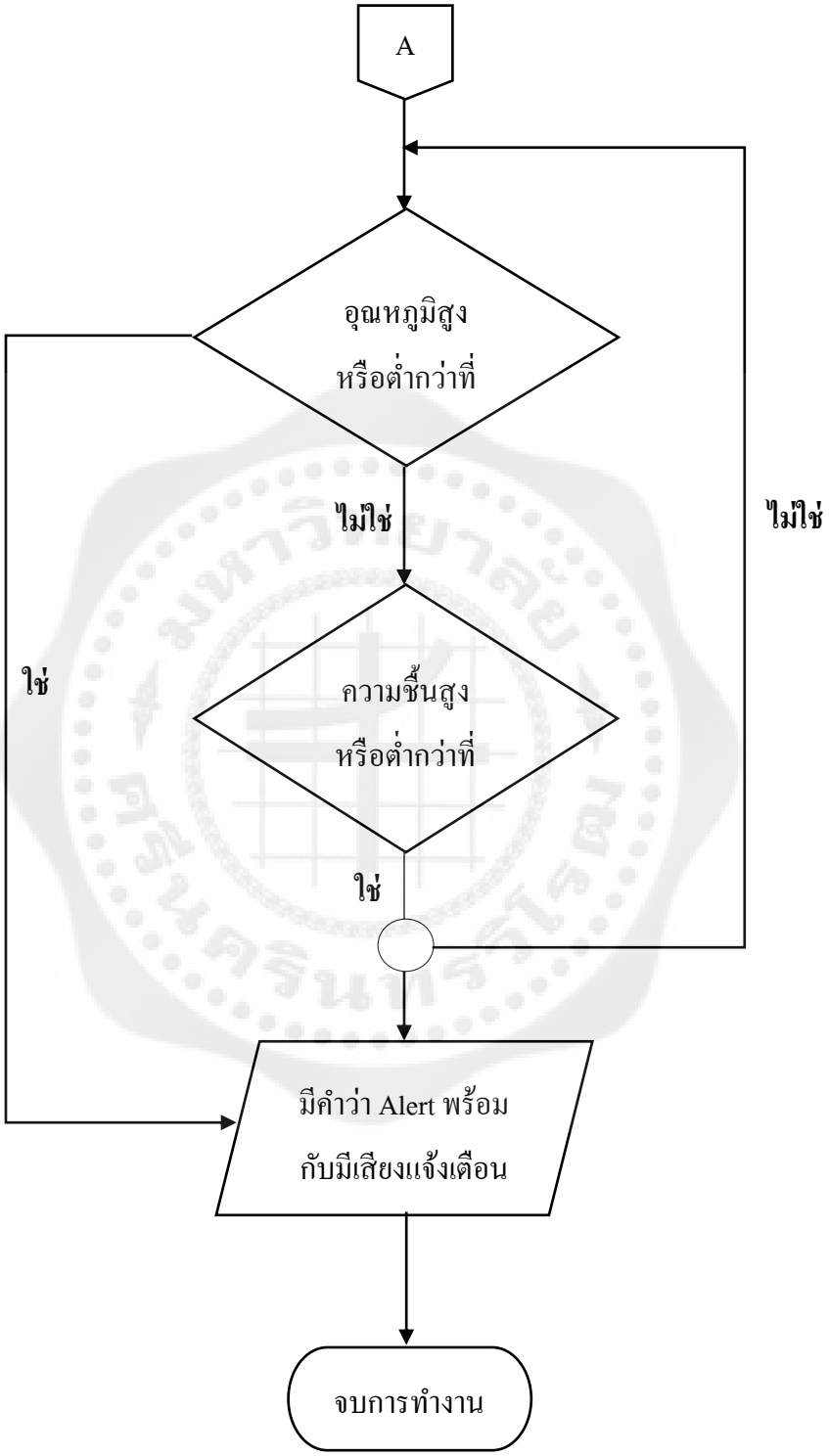


รูปที่ 3.31 แผนผังการทำงานในส่วนของบอร์ด IOIO-Qกับแอปพลิเคชัน

3.6 แผนผังการทำงานในส่วนของแอปพลิเคชัน



รูปที่ 3.32 แผนผังการทำงานของแอปพลิเคชันในส่วนที่ 1



รูปที่ 3.33 แผนผังการทำงานของแอปพลิเคชันในส่วนที่ 2

บทที่ 4

ผลการทดลอง

ในบทนี้เป็นผลการทดลองชุดตรวจจับอุณหภูมิและความชื้นสัมพัทธ์ผ่านบอร์ด IOIO-Q โดยค่าที่ได้จะเป็นค่าตัวเลขอุณหภูมิในหน่วยองศาเซลเซียส (°C) และความชื้นสัมพัทธ์ในหน่วย Relative Humidity (%RH)ในรูปแบบดิจิทัลซึ่งจะส่งค่าแรงดันที่แปรเปลี่ยนไปตามอุณหภูมิและความชื้นที่ตรวจวัดได้ให้กับบอร์ด IOIO-Q เพื่อส่งข้อมูลไปตามสัญญาณบลูทูธมายังแอปพลิเคชัน Hello IOIO ของสมาร์ตโฟนระบบปฏิบัติการแอนดรอยด์ โดยจะมีรายละเอียดดังนี้

4.1 การตั้งค่าและผลการทดลอง

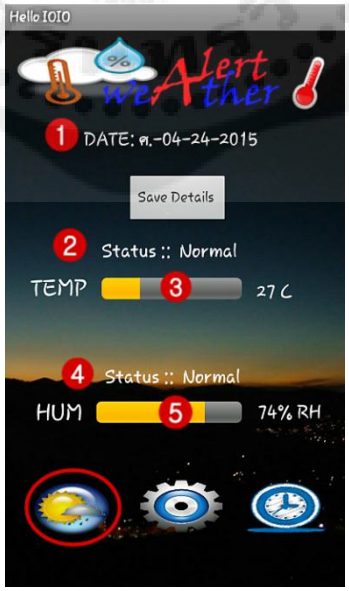
หลังจากที่ได้ทำการออกแบบแอปพลิเคชันด้วยโปรแกรมอีคิปส์ (Eclipse) ซึ่งเป็นเครื่องมือในการพัฒนาแอปพลิเคชันแอนดรอยด์โดยใช้ภาษาจาวาจะได้แอปพลิเคชันดังรูปที่ 4.1 โดยมีชื่อแอปพลิเคชันว่า “Hello IOIO”

รูปที่ 4.1 รูปแอปพลิเคชัน Hello IOIO

รูปที่ 4.2 แอปพลิเคชันที่ใช้ในการทดลองขณะไม่มีการส่งค่าอุณหภูมิและความชื้นสัมพัทธ์

โดยในแอปพลิเคชัน Hello IOIO ถูกแบ่งการทำงานออกเป็น 2 ส่วน คือ

4.1.1 ส่วนของหน้าจอหลัก



รูปที่ 4.3 ส่วนหน้าจอหลักของแอปพลิเคชัน

เมื่อทำการเรียกแอปพลิเคชันขึ้นมาใช้งานจะปรากฏหน้าจอหลักของการใช้งานแอปพลิเคชัน ดังแสดงในรูปที่ 4.3 ในหน้าจอหลักนั้นจะประกอบไปด้วย

หมายเลข 1 แสดงเวลาและวันที่จริง ณ ปัจจุบัน

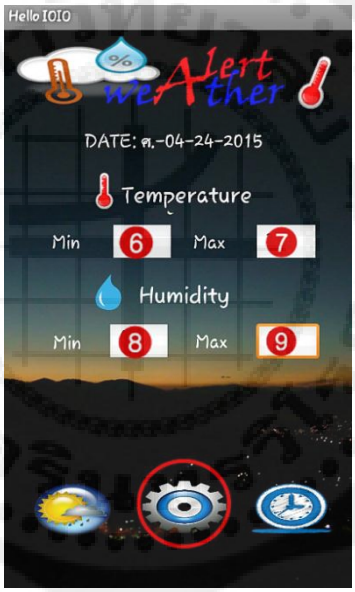
หมายเลข 2 แสดงสถานะของอุณหภูมิที่วัดได้

หมายเลข 3 แทบแสดงผลค่าอุณหภูมิที่วัดด้วยชุดตรวจจับ

หมายเลข 4 แสดงสถานะของความชื้นสัมพัทธ์ที่วัดได้

หมายเลข 5 แทบแสดงผลค่าความชื้นสัมพัทธ์ที่วัดด้วยชุดตรวจจับ

4.1.2 ส่วนของการกำหนดค่าต่ำที่สุดและสูงที่สุด ของอุณหภูมิและความชื้นสัมพัทธ์



รูปที่ 4.4 ส่วนของการกำหนดค่าต่ำที่สุดและสูงที่สุด

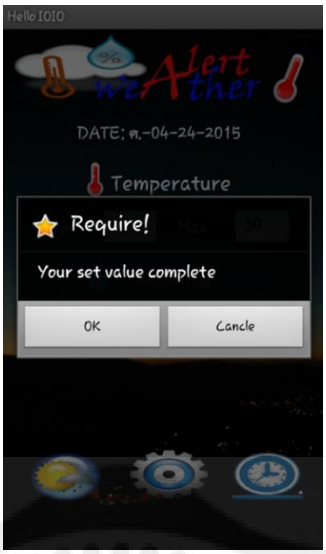
หมายเลข 6 ช่องกำหนดค่าอุณหภูมิต่ำที่สุด

หมายเลข 7 ช่องกำหนดค่าอุณหภูมิสูงที่สุด

หมายเลข 8 ช่องกำหนดค่าอุณหภูมิต่ำที่สุด

หมายเลข 9 ช่องกำหนดค่าอุณหภูมิสูงที่สุด

โดยในการกรอกค่าต่ำที่สุดและสูงที่สุด เพื่อเป็นการควบคุมอุณหภูมิและความชื้นสัมพัทธ์ที่ต้องการวัดในสถานที่นั้น ๆ



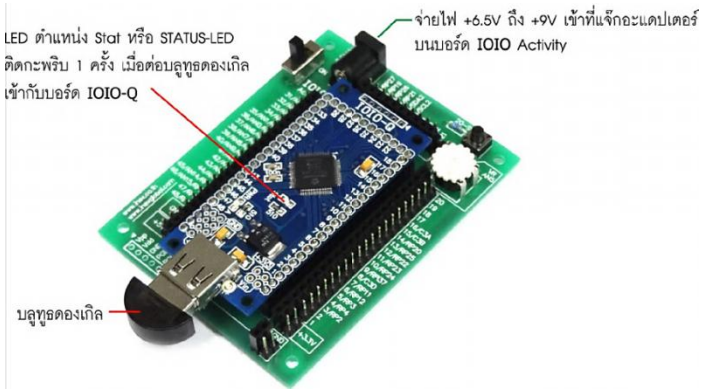
รูปที่ 4.5 หน้าต่างของแอปพลิเคชันเมื่อเสร็จสิ้นการกรอกค่าต่ำที่สุดและสูงที่สุด

เมื่อทำการกรอกค่าต่ำที่สุดและสูงที่สุดของอุณหภูมิและความชื้นสัมพัทธ์เสร็จแล้ว แอปพลิเคชันจะแสดงหน้าต่างนี้ดังรูปที่ 4.5 ให้ทำการกดปุ่ม OK

4.2 การกำหนดค่าของการรับและส่งข้อมูล

ในการกำหนดการรับค่าการทำงานของเครื่องตรวจจับอุณหภูมิและความชื้นสัมพัทธ์ จะมีการติดต่อกับบอร์ดIOIO-Qโดยที่เราต้องทำการเปิดบลูทูธของสมาร์ตโฟน

4.2.1 การเชื่อมต่อสมาร์ตโฟนกับ IOIO-Q แบบไร้สายผ่านบลูทูธ



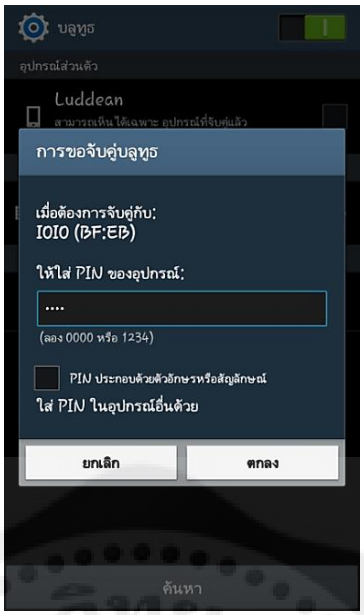
รูปที่ 4.6 ต่อบลูทูธดองเกิลเข้ากับบอร์ด IOIO-Q ที่ติดตั้งบนบอร์ด IOIO Activity

4.2.1.1 ต่อบลูทูธดองเกิลเข้ากับบอร์ดIOIO-Q ดังรูปที่ 4.7 จะเห็น LED แสดงสถานะ (STATUS-LED) ติดกะพริบ 1 ครั้งเป็นการแจ้งให้ทราบว่า IOIO-Q ติดต่อกับบลูทูธดองเกิลได้แล้ว

4.2.1.2 ที่อุปกรณ์แอนดรอยด์ทำการเลือกไปที่ Setting > Wireless and Networks > Bluetooth Settings เลือกเครื่องหมายถูกที่ Bluetooth เพื่อเปิดการใช้งานบลูทูธรอสักครู่ เพื่อให้อุปกรณ์ค้นหาสัญญาณจากบลูทูธดองเกิล หากพบจะแสดงเป็นชื่ออุปกรณ์นำหน้าด้วย IOIO แล้วต่อด้วยหมายเลขประจำตัวของบลูทูธดองเกิล ในที่นี้ชื่ออุปกรณ์คือ “IOIO (BF: EB)” ดังแสดงในรูปที่ 4.8 จากนั้นทำการจับคู่ (pairing) กับบลูทูธดองเกิล

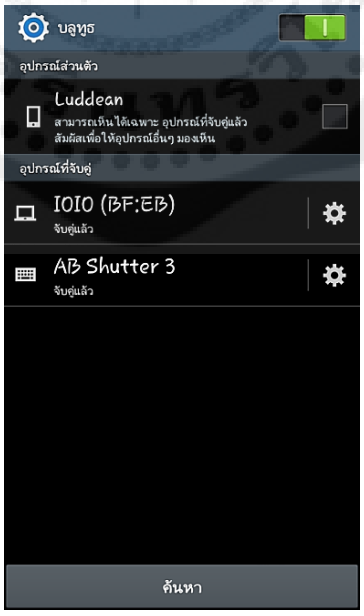
รูปที่ 4.7 การค้นหาสัญญาณจากบลูทูธดองเกิล

4.2.1.3 ในการจับคู่สัญญาณกับบลูทูธดองเกิลระบบจะให้ใส่เลขรหัสของการจับคู่ ซึ่งก็คือ 4545 จากนั้นเลือกปุ่ม OK ดังแสดงในรูปที่ 4.8



รูปที่ 4.8 การจับคู่สัญญาณกับบลูทูธดองเกิล

4.2.1.4 ถ้าเชื่อมต่อได้แล้วจะปรากฏข้อความว่า “จับคู่แล้ว” แสดงว่าเชื่อมต่อได้แล้ว



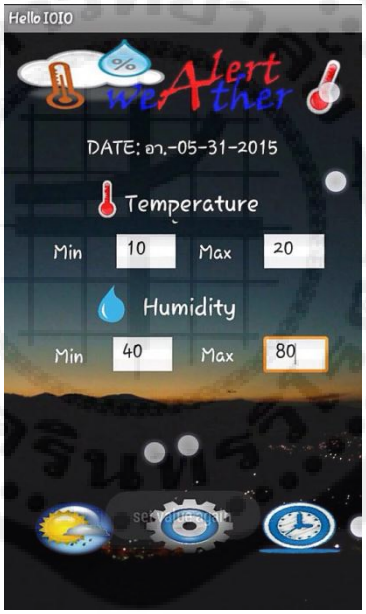
รูปที่ 4.9 สถานะการเชื่อมต่อสมาร์ตโฟนกับ IOIO-Q แบบไร้สายผ่านบลูทูธ

4.2.2 การทำงานและการแจ้งเตือนของแอปพลิเคชัน

การทำงานของแอปพลิเคชันนี้ จะขึ้นอยู่กับการกำหนดขอบเขตของอุณหภูมิและความชื้นสัมพัทธ์ที่ต่ำที่สุดและสูงที่สุดสำหรับการควบคุมตามสถานที่นั้น ๆ ถ้าอุณหภูมิหรือความชื้นสัมพัทธ์มีค่าสูงกว่าหรือต่ำกว่าที่กำหนดนั้น แอปพลิเคชันก็จะมีการส่งค่าการแจ้งเตือนเป็นข้อความแสดงที่หน้าจอสมาร์ตโฟนพร้อมทั้งมีการแจ้งเตือนในรูปแบบของเสียงอีกด้วย

ตัวอย่างการแจ้งเตือนผ่านแอปพลิเคชันโดยได้ทำการกำหนดขอบเขตของอุณหภูมิและความชื้นสัมพัทธ์ที่ต่ำที่สุดและสูงที่สุด โดยการแจ้งเตือนแบ่งออกเป็น 3 กรณี ดังนี้

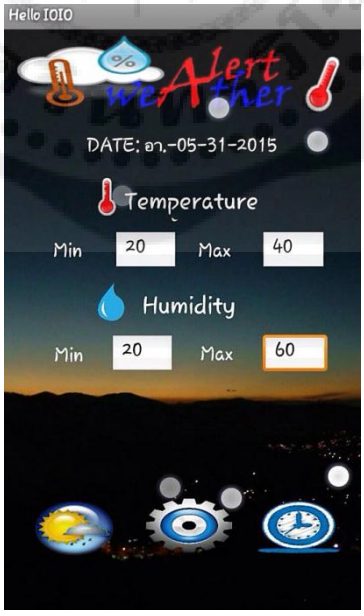
4.2.2.1 ค่าอุณหภูมิที่วัดได้อยู่นอกเหนือจากค่าที่กำหนดไว้



รูปที่ 4.10 การกำหนดค่าสูงสุดและต่ำสุดของอุณหภูมิและความชื้น

รูปที่ 4.11 แอปพลิเคชันแสดงแสดงสถานะ Alert ที่ค่าของอุณหภูมิ

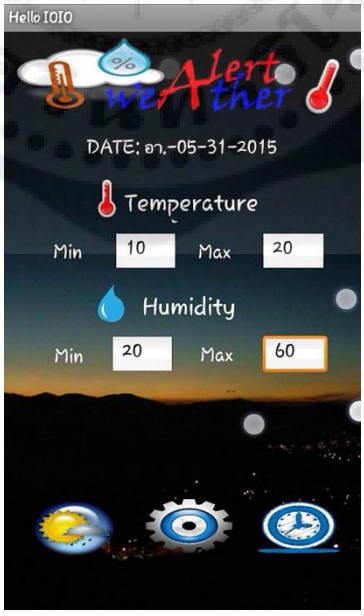
4.2.2.2 ค่าความชื้นที่วัดได้อยู่นอกเหนือจากค่าที่กำหนดไว้



รูปที่ 4.12 แสดงการกำหนดค่าสูงสุดและต่ำสุดของอุณหภูมิและความชื้น

รูปที่ 4.13 แอปพลิเคชันแสดงสถานะ Alert ที่ค่าของความชื้น

4.2.2.3 ค่าอุณหภูมิและความชื้นที่วัดได้อยู่นอกเหนือจากค่าที่กำหนดไว้



รูปที่ 4.14 แสดงการกำหนดค่าสูงสุดและต่ำสุดของอุณหภูมิและความชื้น

รูปที่ 4.15 แอปพลิเคชันแสดงสถานะ Alert ที่ค่าของอุณหภูมิและความชื้น

4.3 ตารางสรุปผลการทดลอง

ในการทดลองครั้งนี้เราได้ทำการทดลองวัดอุณหภูมิและความชื้นสัมพัทธ์ โดยแบ่งเป็น 4 การทดลองคือ การทดลองที่อุณหภูมิห้อง การทดลองที่อุณหภูมิต่ำ การทดลองที่อุณหภูมิสูง และการทดลองระยะทางของสัญญาณบลูทูธ โดยการทดลองทุกครั้งจะต้องวัดค่าอุณหภูมิและความชื้นสัมพัทธ์ที่ได้จากชุดตรวจจับ เทียบกับ Thermo-Hygrometer เพื่อที่จะได้ทราบค่าความคลาดเคลื่อน ความแม่นยำ และความเที่ยงตรง นำมาเปรียบเทียบกันแล้วพิจารณาว่าค่าที่ได้จากชุดตรวจจับ ใกล้เคียงกับค่าที่ได้จาก Thermo-Hygrometer มากน้อยเพียงใด

4.3.1 การทดลองที่อุณหภูมิห้อง

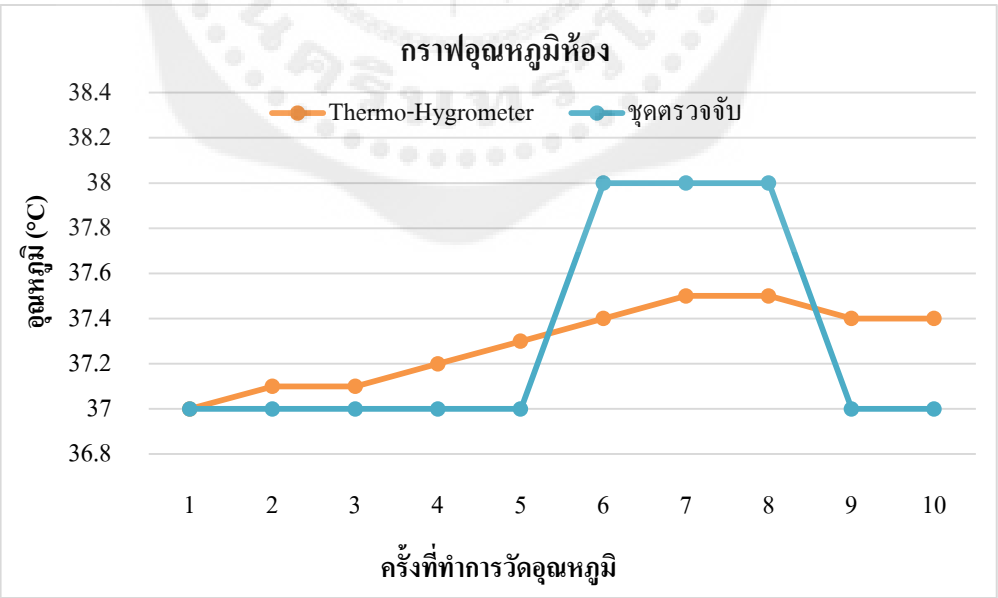
ทำการทดลองโดยนำชุดตรวจจับและเซนเซอร์ของ Thermo-Hygrometer มาใส่ลงในกล่องจำลอง แล้วทำการวัดค่าอุณหภูมิและความชื้นสัมพัทธ์ ทำการวัดทั้งหมด 10 ครั้ง ช่วงเวลาห่างกันครั้งละ 1 นาที โดยอุณหภูมิห้องอยู่ที่ 34 °C และความชื้นสัมพัทธ์ 60%RH

รูปที่ 4.16 การทดลองที่อุณหภูมิห้อง

รูปที่ 4.17 หน้าจอแอปพลิเคชันแสดงการทำงานในขณะที่ทำการวัดอุณหภูมิห้อง

ตารางที่ 4.1 อุณหภูมิที่วัดได้จาก Thermo-Hygrometer เทียบกับชุดตรวจจับที่อุณหภูมิห้อง

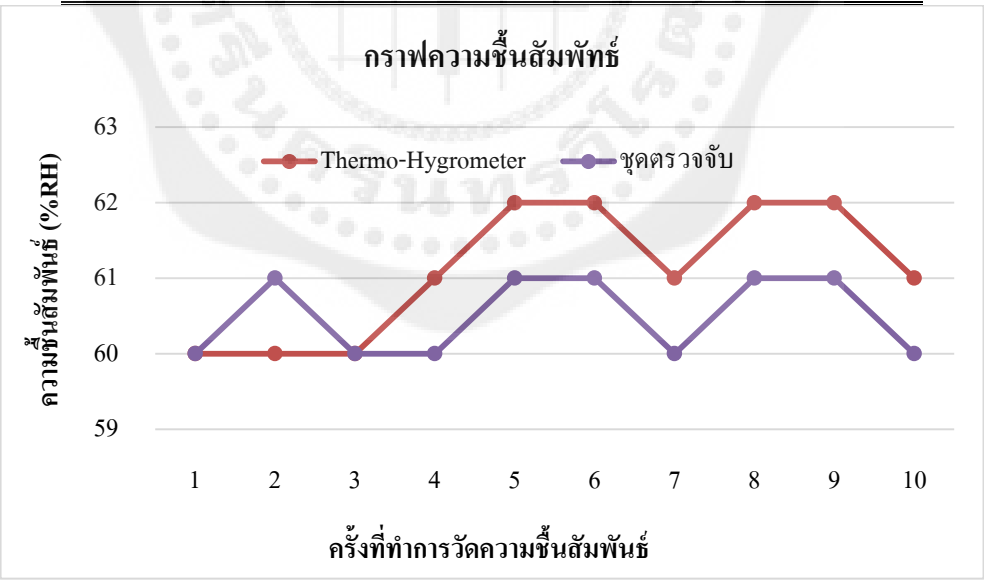
ครั้งที่	อุณหภูมิ (°C)		ความคลาดเคลื่อน (%)
	Thermo-Hygrometer	ชุดตรวจจับ	
1	37.0	37.0	0.00
2	37.1	37.0	0.27
3	37.1	37.0	0.27
4	37.2	37.0	0.53
5	37.3	37.0	0.80
6	37.4	38.0	1.60
7	37.5	38.0	1.33
8	37.5	38.0	1.33
9	37.4	37.0	1.07
10	37.4	37.0	1.07



รูปที่ 4.18 กราฟอุณหภูมิที่วัดได้จาก Thermo-Hygrometer เทียบกับชุดตรวจจับที่อุณหภูมิห้อง

ตารางที่ 4.2 ความชื้นที่วัดได้จาก Thermo-Hygrometer เทียบกับชุดตรวจจับที่อุณหภูมิห้อง

ครั้งที่	ความชื้นสัมพัทธ์ (%RH)		ความคลาดเคลื่อน (%)
	Thermo-Hygrometer	ชุดตรวจจับ	
1	60	60	0.00
2	60	61	1.67
3	60	60	0.00
4	61	60	1.64
5	62	61	1.61
6	62	61	1.61
7	61	60	1.64
8	62	61	1.61
9	62	61	1.61
10	61	60	1.64

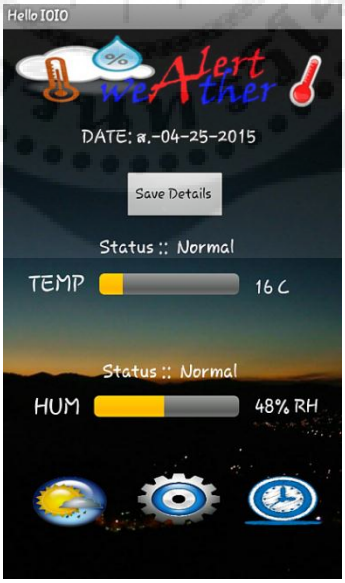


รูปที่ 4.19 กราฟความชื้นที่วัดได้จาก Thermo-Hygrometer เทียบกับชุดตรวจจับที่อุณหภูมิห้อง

4.3.2 การทดลองที่อุณหภูมิต่ำ

ทำการทดลองโดยนำชุดตรวจจับและเซนเซอร์ของ Thermo-Hygrometer วางลงในถังที่บรรจุน้ำแข็ง แล้วทำการวัดค่าอุณหภูมิและความชื้นสัมพัทธ์ ทำการวัดทั้งหมด 10 ครั้ง ช่วงเวลาห่างกันครั้งละ 1 นาที โดยอุณหภูมิห้องอยู่ที่ 16°C และความชื้นสัมพัทธ์ 48%RH

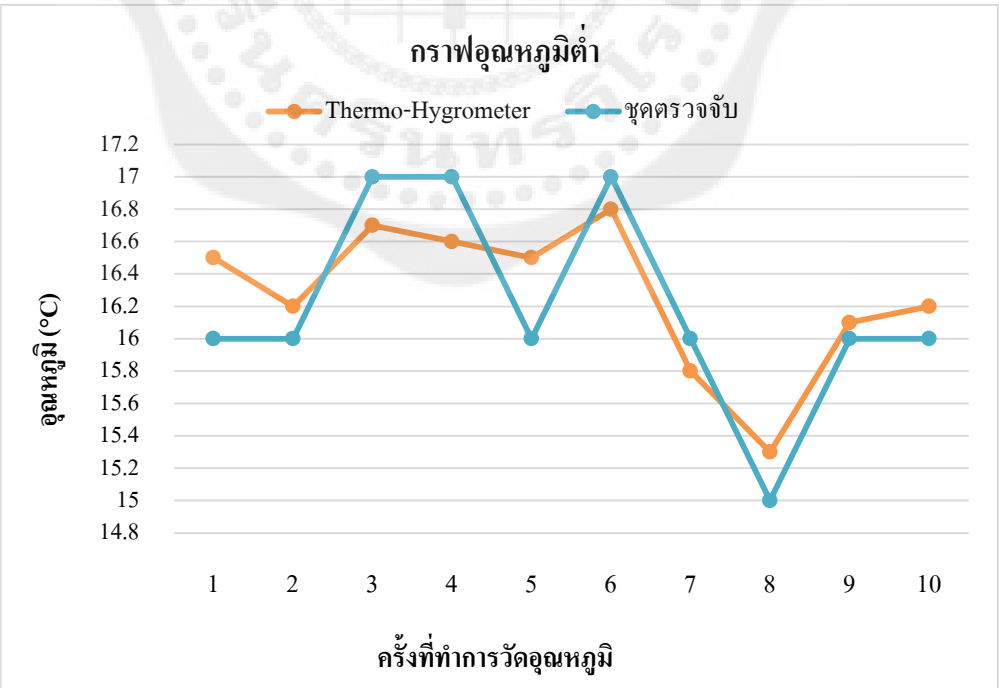
รูปที่ 4.20 การทดลองที่อุณหภูมิต่ำ



รูปที่ 4.21 หน้าจอแอปพลิเคชันแสดงการทำงานในขณะที่ทำการวัดอุณหภูมิต่ำ

ตารางที่ 4.3 อุณหภูมิที่วัดได้จาก Thermo-Hygrometer เทียบกับชุดตรวจจับที่อุณหภูมิต่ำ

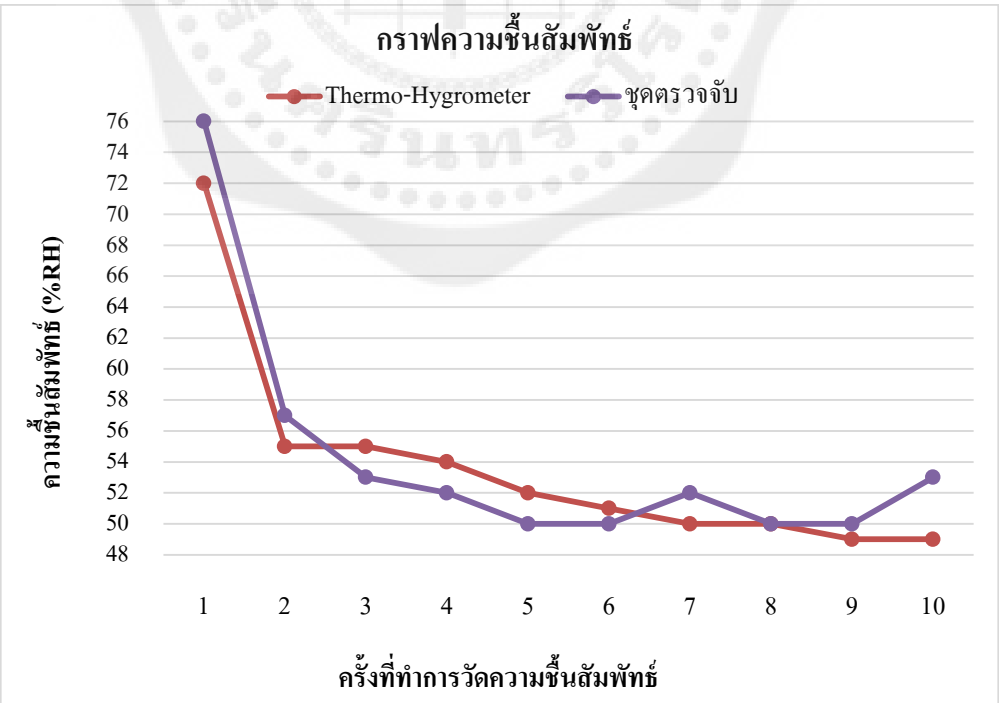
ครั้งที่	อุณหภูมิ (°C)		ความคลาดเคลื่อน (%)
	Thermo-Hygrometer	ชุดตรวจจับ	
1	16.5	16.0	3.03
2	16.2	16.0	1.24
3	16.7	17.0	1.80
4	16.6	17.0	2.41
5	16.5	16.0	3.03
6	16.8	17.0	1.20
7	15.8	16.0	1.27
8	15.3	15.0	1.96
9	16.1	16.0	0.62
10	16.2	16.0	1.24



รูปที่ 4.22 กราฟอุณหภูมิที่วัดได้จาก Thermo-Hygrometer เทียบกับชุดตรวจจับที่อุณหภูมิต่ำ

ตารางที่ 4.4 ความชื้นที่วัดได้จาก Thermo-Hygrometer เทียบกับชุดตรวจจับที่อุณหภูมิต่ำ

ครั้งที่	ความชื้นสัมพัทธ์ (%RH)		ความคลาดเคลื่อน (%)
	Thermo-Hygrometer	ชุดตรวจจับ	
1	72	76	5.56
2	55	57	3.64
3	55	53	3.64
4	54	52	3.70
5	52	50	3.85
6	51	50	1.96
7	50	52	4.00
8	50	50	0.00
9	49	50	2.04
10	49	53	8.16



รูปที่ 4.23 กราฟความชื้นที่วัดได้จาก Thermo-Hygrometer เทียบกับชุดตรวจจับที่อุณหภูมิต่ำ

4.3.3 การทดลองที่อุณหภูมิสูง

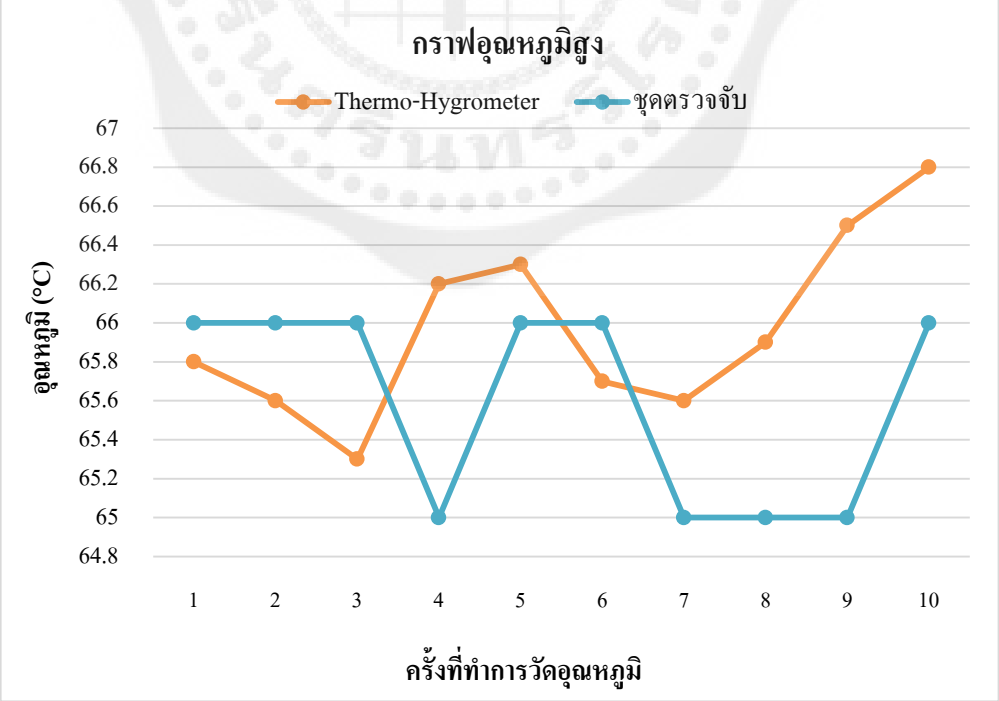
ทำการทดลองโดยนำชุดตรวจจับและเซนเซอร์ของ Thermo-Hygrometer มาใส่ลงในกล่องจำลอง เป่าลมร้อนจากไดร์เป่าผมเข้าไปภายในกล่อง แล้วทำการวัดค่าอุณหภูมิ และความชื้นสัมพัทธ์ ทำการวัดทั้งหมด 10 ครั้ง ช่วงเวลาห่างกันครั้งละ 1 นาที โดยอุณหภูมิห้องอยู่ที่ 66°C และความชื้นสัมพัทธ์ 25%RH

รูปที่ 4.24 การทดลองที่อุณหภูมิสูง

รูปที่ 4.25 หน้าจอแอปพลิเคชันแสดงการทำงานในขณะที่ทำการวัดอุณหภูมิสูง

ตารางที่ 4.5 อุณหภูมิที่วัดได้จาก Thermo-Hygrometer เทียบกับชุดตรวจจับที่อุณหภูมิสูง

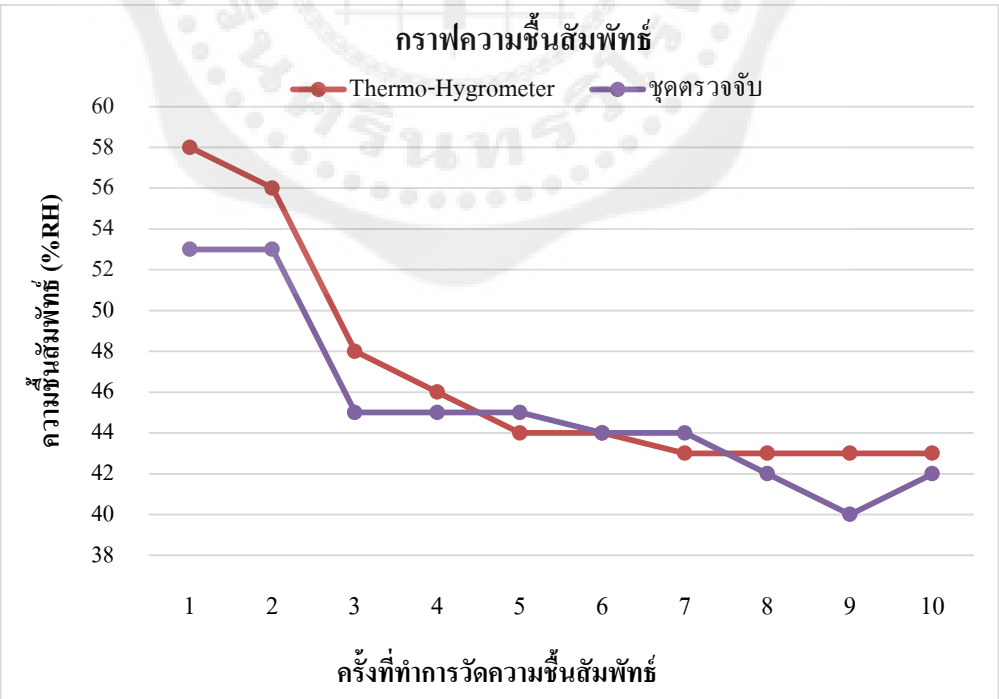
ครั้งที่	อุณหภูมิ (°C)		ความคลาดเคลื่อน (%)
	Thermo-Hygrometer	ชุดตรวจจับ	
1	65.8	66.0	0.31
2	65.6	66.0	0.61
3	65.3	66.0	1.07
4	66.2	65.0	1.81
5	66.3	66.0	0.45
6	65.7	66.0	0.46
7	65.6	65.0	0.76
8	65.9	65.0	0.76
9	66.5	65.0	2.26
10	66.8	66.0	1.20



รูปที่ 4.26 กราฟอุณหภูมิที่วัดได้จาก Thermo-Hygrometer เทียบกับชุดตรวจจับที่อุณหภูมิสูง

ตารางที่ 4.6 ความชื้นที่วัดได้จาก Thermo-Hygrometer เทียบกับชุดตรวจจับที่อุณหภูมิสูง

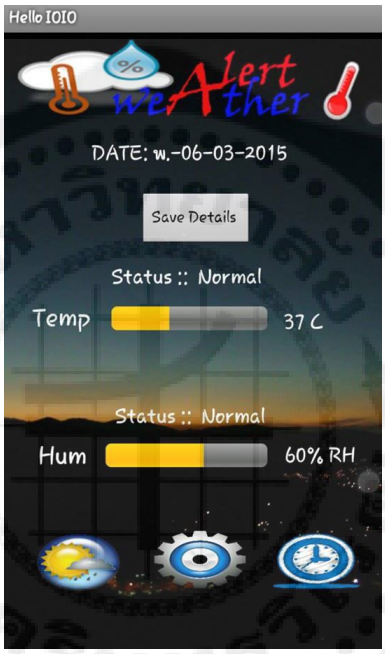
ครั้งที่	ความชื้นสัมพัทธ์ (%RH)		ความคลาดเคลื่อน (%)
	Thermo-Hygrometer	ชุดตรวจจับ	
1	58	53	8.62
2	56	53	5.36
3	48	45	6.25
4	46	45	2.17
5	44	45	2.27
6	44	44	0.00
7	43	44	2.33
8	43	42	2.33
9	43	40	6.98
10	43	42	2.33



รูปที่ 4.27 กราฟความชื้นที่วัดได้จาก Thermo-Hygrometer เทียบกับชุดตรวจจับที่อุณหภูมิสูง

4.3.4 การทดลองระยะทางของสัญญาณบลูทูธ

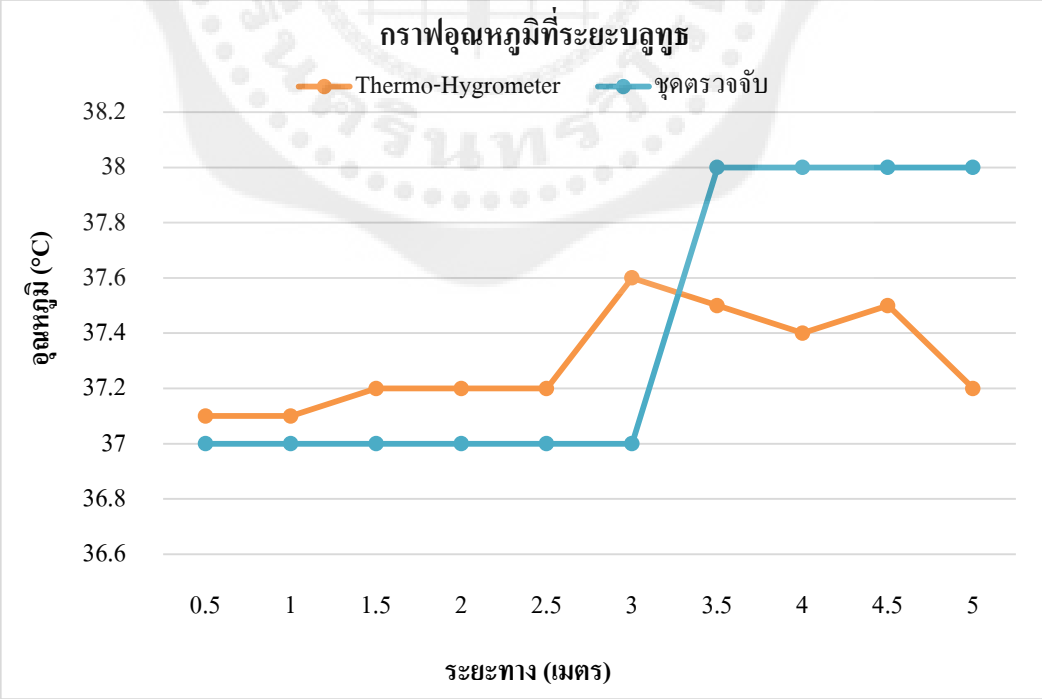
ทำการทดลองโดยนำชุดตรวจจับและเซนเซอร์ของ Thermo-Hygrometer มาใส่ลงในกล่องจำลอง วางอุปกรณ์แอนดรอย์และกล่องจำลองให้ห่างกันในระยะทางของสัญญาณบลูทูธต่าง ๆ ที่กำหนดไว้ในตารางที่ 4.7 –4.8 จากนั้นทำการวัดค่าอุณหภูมิและความชื้นสัมพัทธ์ โดยอุณหภูมิห้องอยู่ที่ 37°C และความชื้นสัมพัทธ์ 60%RH



รูปที่ 4.28 หน้าจอแอปพลิเคชันแสดงการทำงานในขณะที่ทำการวัดระยะทางของสัญญาณบลูทูธ

ตารางที่4.7 อุณหภูมิที่วัดได้จาก Thermo-Hygrometer เทียบกับชุดตรวจจับที่ระยะทางบลูทูธ

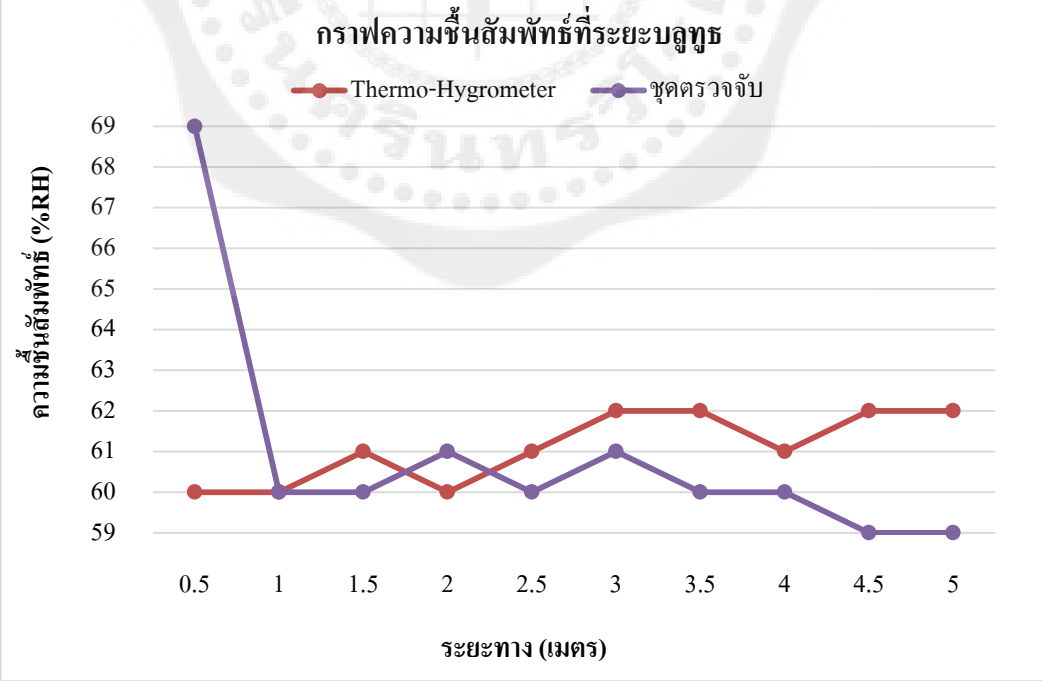
ระยะทาง (เมตร)	อุณหภูมิ (°C)		ความคลาดเคลื่อน (%)
	Thermo-Hygrometer	ชุดตรวจจับ	
0.5	37.1	37.0	0.27
1.0	37.1	37.0	0.27
1.5	37.2	37.0	0.53
2.0	37.2	37.0	0.53
2.5	37.2	37.0	0.53
3.0	37.6	37.0	1.59
3.5	37.5	38.0	1.33
4.0	37.4	38.0	1.60
4.5	37.5	38.0	1.33
5.0	37.2	38.0	2.15



รูปที่ 4.29 กราฟอุณหภูมิที่วัดได้จาก Thermo-Hygrometer เทียบกับชุดตรวจจับที่ระยะทางบลูทูธ

ตารางที่ 4.8 ความชื้นที่วัดได้จาก Thermo-Hygrometer เทียบกับชุดตรวจจับที่ระยะทางบลูทูธ

ระยะทาง (เมตร)	ความชื้นสัมพัทธ์ (%RH)		ความคลาดเคลื่อน (%)
	Thermo-Hygrometer	ชุดตรวจจับ	
0.5	60	69	1.67
1.0	60	60	0.00
1.5	61	60	1.64
2.0	60	61	1.67
2.5	61	60	1.67
3.0	62	61	1.62
3.5	62	60	3.30
4.0	61	60	1.67
4.5	62	59	3.27
5.0	62	59	3.27



รูปที่ 4.30 กราฟความชื้นที่วัดได้จาก Thermo-Hygrometer เทียบกับชุดตรวจจับที่ระยะทางบลูทูธ

ค่าที่ได้จะเปรียบเทียบตามตารางที่ 4.1 - 4.8 จะสังเกตได้ว่าค่าที่วัดได้จากชุดตรวจจับเมื่อเทียบกับThermo-Hygrometerจะมีค่าใกล้เคียงกันเมื่อคิดเป็นเปอร์เซ็นความคลาดเคลื่อนของอุณหภูมิและความชื้นสัมพัทธ์

โดยค่าความคลาดเคลื่อน(Error)หาได้จากสูตร

$$\%\text{Error} = \left|\frac{X_{mea}-X_t}{X_t}\right| \times 100$$

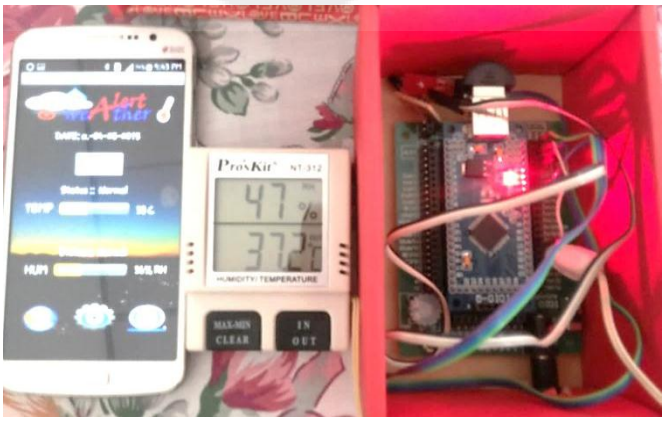
เมื่อ X_{mea} คือ ค่าที่ได้จากการวัด (measure value) ในการทดลองนี้คือค่าจากชุดตรวจจับ

X_t คือ ค่าจริง (true value) ในการทดลองนี้คือค่าที่ได้จากThermo-Hygrometer

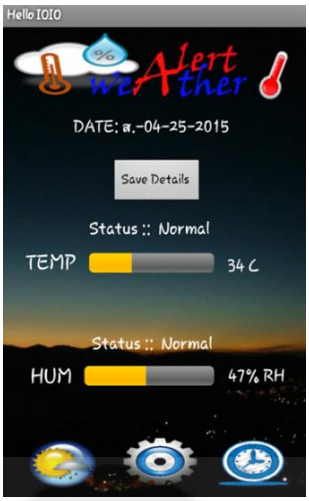
4.4 ค่าความแม่นยำของเครื่องมือวัด (Accuracy: A)

ค่าความแม่นยำหรือค่าความถูกต้อง คือ ค่าที่วัดได้หรือทดลองได้ในแต่ละครั้งมีค่าเข้าใกล้ค่าที่แท้จริงหรือค่าทฤษฎีมากน้อยเพียงใด

การทดลองหาค่าความแม่นยำ ได้ทำการทดลองโดยนำชุดตรวจจับและเซนเซอร์ของ Thermo-Hygrometer มาใส่ลงในกล่องจำลอง แล้วทำการวัดค่าอุณหภูมิและความชื้นสัมพัทธ์ที่อุณหภูมิห้อง ทำการวัดทั้งหมด 15 ครั้ง ช่วงเวลาห่างกันครั้งละ 1 นาที โดยอุณหภูมิห้องอยู่ที่ 34 °C และความชื้นสัมพัทธ์ 47%RH



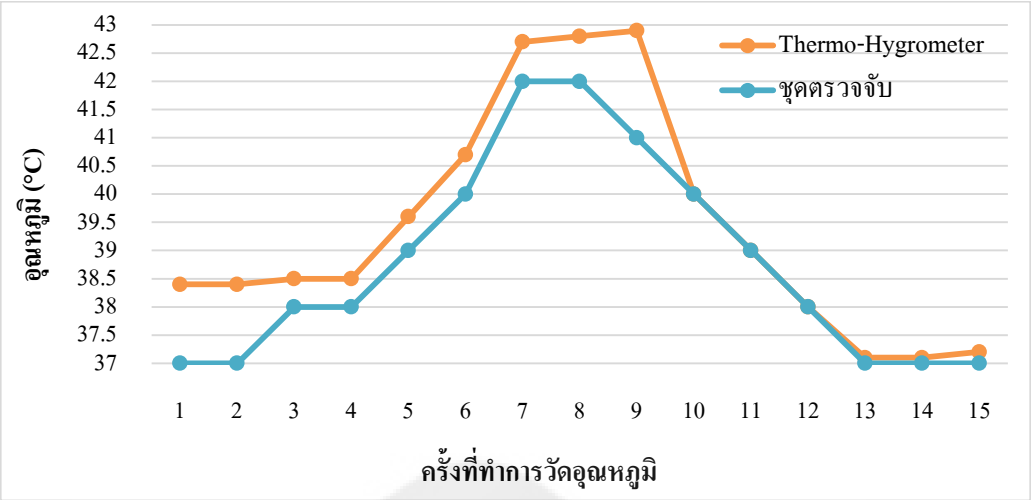
รูปที่ 4.31 การทดลองที่อุณหภูมิห้องเพื่อหาค่าความแม่นยำ



รูปที่ 4.32 หน้าจอแอปพลิเคชันในขณะที่ทำการวัดอุณหภูมิห้องเพื่อหาค่าความแม่นยำ

ตารางที่ 4.9 ค่าความแม่นยำของการวัดอุณหภูมิจากThermo-Hygrometerเทียบกับชุดตรวจจับ

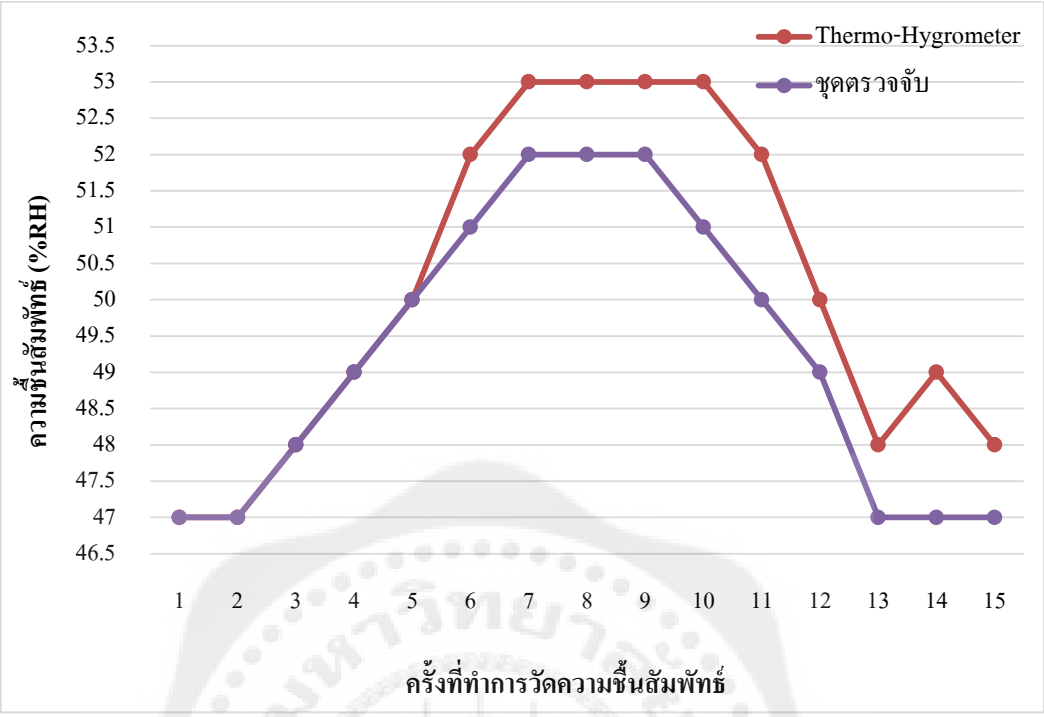
ครั้งที่	อุณหภูมิ (°C)		ความแม่นยำ (%)
	Thermo-Hygrometer	ชุดตรวจจับ	
1	38.4	37.0	96.05
2	38.4	37.0	96.05
3	38.5	38.0	98.59
4	38.5	38.0	98.59
5	39.6	39.0	98.31
6	40.7	40.0	98.04
7	42.7	42.0	98.04
8	42.8	42.0	97.77
9	42.9	41.0	97.49
10	40.0	40.0	100.00
11	39.0	39.0	100.00
12	38.0	38.0	100.00
13	37.1	37.0	99.72
14	37.1	37.0	99.72
15	37.2	37.0	99.45



รูปที่ 4.33 กราฟความแม่นยำของอุณหภูมิที่วัดได้จาก Thermo-Hygrometer เทียบกับชุดตรวจจับ

ตารางที่ 4.10 ค่าความแม่นยำของการวัดความชื้นจาก Thermo-Hygrometer เทียบกับชุดตรวจจับ

ครั้งที่	ความชื้นสัมพัทธ์ (%RH)		ความคลาดเคลื่อน (%)
	Thermo-Hygrometer	ชุดตรวจจับ	
1	47.0	47.0	100.00
2	47.0	47.0	100.0
3	48.0	48.0	100.00
4	49.0	49.0	100.00
5	50.0	50.0	100.00
6	52.0	51.0	97.83
7	53.0	52.0	97.83
8	53.0	52.0	97.83
9	53.0	52.0	97.83
10	53.0	51.0	95.65
11	52.0	50.0	95.65
12	50.0	49.0	97.78
13	48.0	47.0	97.78
14	49.0	47.0	95.65
15	48.0	47.0	97.78



รูปที่ 4.34 กราฟความแม่นยำของความชื้นที่วัดได้จาก Thermo-Hygrometer เทียบกับชุดตรวจจับ

จากตารางที่ 4.9 - 4.10 ได้แสดงถึงค่าความแม่นยำของการวัดอุณหภูมิและความชื้นสัมพัทธ์จากThermo-Hygrometerเทียบกับชุดตรวจจับซึ่งวัดจากอุณหภูมิห้องทั้งหมด 15ครั้งโดยมีค่าความแม่นยำมากกว่า 90% จึงถือว่าเครื่องมือที่ใช้ทั้งสองมีความแม่นยำเพียงพอสำหรับการวัดอุณหภูมิและความชื้นสัมพัทธ์

โดยค่าความแม่นยำ(Accuracy) หาได้จากสูตร

$$\%Accuracy = 100 - \%Error$$

โดยที่

$$\%Error = \left|\frac{X_{mea}-X_t}{X_t}\right| \times 100$$

เมื่อ X_{mea} คือ ค่าที่ได้จากการวัด (measure value) ในการทดลองนี้คือค่าที่ได้จากชุดตรวจจับ

X_t คือ ค่าจริง (true value) ในการทดลองนี้คือค่าที่ได้จากThermo-Hygrometer

4.5 ค่าความเที่ยงตรงของเครื่องมือวัด (Precision)

ค่าความเที่ยงตรงคือค่าวัดได้ใกล้เคียงกันมากน้อยเพียงใดจากการวัดตัวแปรเดียวกันซ้ำกันหลาย ๆ ครั้ง

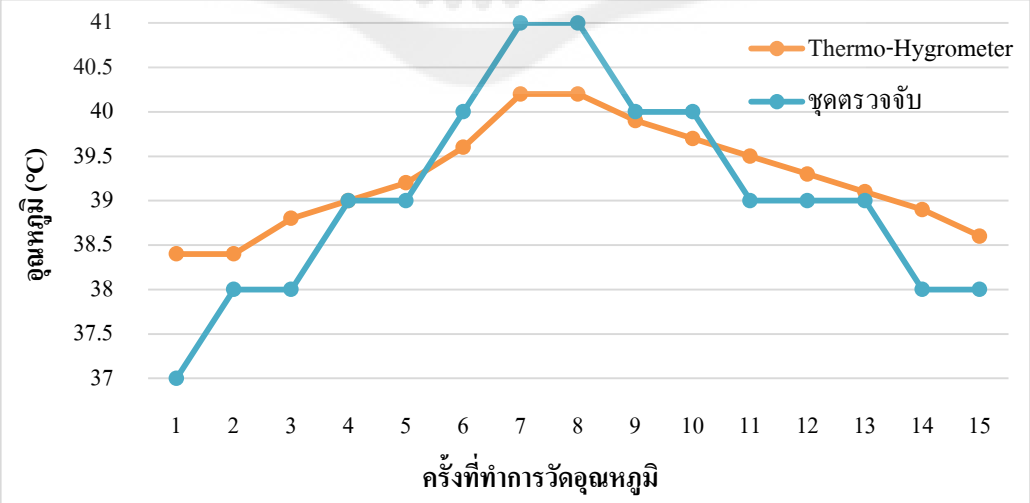
การทดลองหาค่าความเที่ยงตรง ได้ทำการทดลองโดยนำชุดตรวจจับและเซนเซอร์ของ Thermo-Hygrometerมาใส่ลงในกล่องจำลอง แล้วทำการวัดค่าอุณหภูมิและความชื้นสัมพัทธ์ที่อุณหภูมิห้อง ทำการวัดทั้งหมด 15 ครั้ง ช่วงเวลาห่างกันครั้งละ 1 นาที โดยอุณหภูมิห้องอยู่ที่ 37°C และความชื้นสัมพัทธ์ 47%RH

รูปที่ 4.35 การทดลองที่อุณหภูมิห้องเพื่อหาค่าความเที่ยงตรง

รูปที่ 4.36 หน้าจอแอปพลิเคชันในขณะทำการวัดที่อุณหภูมิห้องเพื่อหาค่าความเที่ยงตรง

ตารางที่ 4.11 ค่าความเที่ยงตรงของการวัดอุณหภูมิจาก Thermo-Hygrometer เทียบกับชุดตรวจจับ

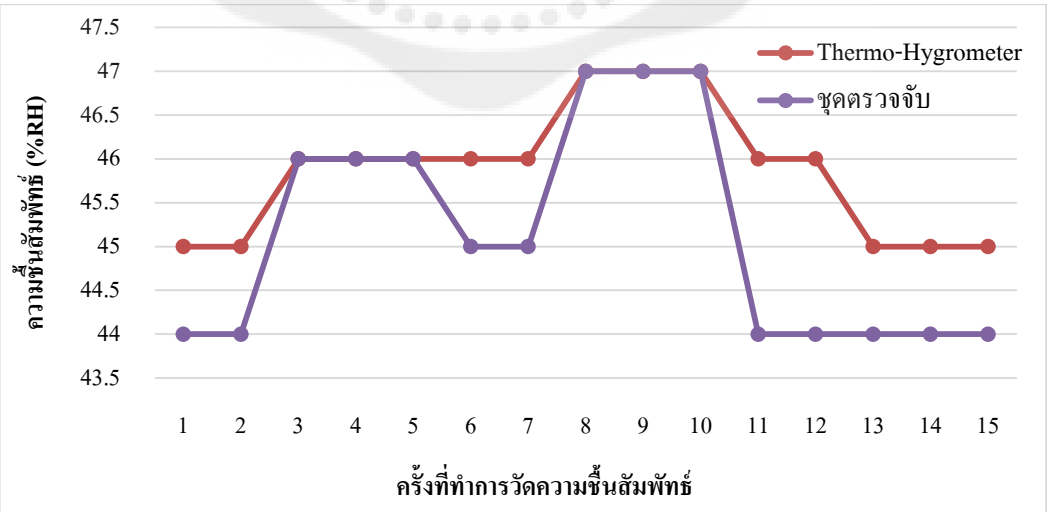
ครั้งที่	อุณหภูมิ (°C)			
	Thermo-Hygrometer(°C)	ความเที่ยงตรง(%)	ชุดตรวจจับ(°C)	ความเที่ยงตรง (%)
1	38.4	97.82	37.0	94.52
2	38.4	97.82	38.0	94.52
3	38.8	98.92	38.0	97.30
4	39.0	99.47	39.0	99.92
5	39.2	99.97	39.0	99.92
6	39.6	98.87	40.0	97.10
7	40.2	97.21	41.0	94.36
8	40.2	97.21	41.0	94.36
9	39.9	98.04	40.0	97.14
10	39.7	98.60	40.0	97.14
11	39.5	99.14	39.0	99.92
12	39.3	99.70	39.0	99.92
13	39.1	99.75	39.0	99.92
14	38.9	99.20	38.0	97.30
15	38.6	98.37	38.0	97.30



รูปที่ 4.37 กราฟความเที่ยงตรงของอุณหภูมิที่วัดได้จาก Thermo-Hygrometer เทียบกับชุดตรวจจับ

ตารางที่ 4.12 ค่าความเที่ยงตรงของการวัดความชื้นจากThermo-Hygrometerเทียบกับชุดตรวจจับ

ครั้งที่	ความชื้นสัมพัทธ์ (%RH)			
	Thermo-Hygrometer(%RH)	ความเที่ยงตรง(%)	ชุดตรวจจับ(%RH)	ความเที่ยงตรง(%)
1	45	98.62	44	99.77
2	45	98.62	44	99.77
3	46	99.19	46	95.22
4	46	99.19	46	95.22
5	46	99.19	46	95.22
6	46	99.19	45	97.49
7	46	99.19	45	97.49
8	47	97.00	47	92.94
9	47	97.00	47	92.94
10	47	97.00	47	92.94
11	46	99.19	44	99.77
12	46	99.19	44	99.77
13	45	98.62	44	99.77
14	45	98.62	44	99.77
15	45	98.62	44	99.77



รูปที่ 4.38 กราฟความเที่ยงตรงของความชื้นที่วัดได้จาก Thermo-Hygrometer เทียบกับชุดตรวจจับ

จากตารางที่ 4.11 – 4.12ได้แสดงถึงค่าความเที่ยงตรงของการวัดอุณหภูมิและความชื้นสัมพัทธ์ จากThermo-Hygrometerเทียบกับชุดตรวจจับซึ่งวัดจากอุณหภูมิห้องทั้งหมด 15ครั้ง โดยมีค่าความเที่ยงตรงมากกว่า 90% จึงถือว่าเครื่องมือที่ใช้ทั้งสองมีความเที่ยงตรงเพียงพอสำหรับการวัดอุณหภูมิและความชื้นสัมพัทธ์

โดยค่าความเที่ยงตรง (Precision)หาได้จากสูตร

$$\%\text{Precision}= \left(1- \left|\frac{X_i - X_m}{X_m}\right|\right) \times 100$$

โดยที่

$$X_m = \frac{1}{n}\sum_{i=1}^{n} X_i$$

เมื่อ X_i คือ ค่าการวัดแต่ละครั้ง

X_m คือ ค่าเฉลี่ยของการวัด

n คือ จำนวนครั้งของการวัด

4.6 ค่าน้อยที่สุดและมากที่สุดในการวัดค่าอุณหภูมิและความชื้นสัมพัทธ์

ค่าที่น้อยที่สุดและมากที่สุดของอุณหภูมิและความชื้นสัมพัทธ์ที่วัดได้จาก Thermo-Hygrometer และเครื่องตรวจจับเป็นดังตารางที่ 4.13

ตารางที่ 4.13 ค่าน้อยที่สุดและมากที่สุด ที่วัดได้จาก Thermo-Hygrometer และเครื่องตรวจจับ

เครื่องมือวัด	อุณหภูมิ (°C)	ความชื้นสัมพัทธ์ (%RH)
Thermo-Hygrometer	-10 ถึง +55	20 ถึง 99
เครื่องตรวจจับ	0 ถึง 100	0 ถึง 75

บทที่ 5
ผลสรุปและข้อเสนอแนะ

5.1 สรุปผลของโครงงาน

โครงงานนี้ได้ทำการออกแบบและสร้างแอปพลิเคชันด้วยภาษาจาวาสำหรับวัดอุณหภูมิและความชื้นผ่านบอร์ด IOIO-Qโดยส่งผ่านข้อมูลแบบบลูทูธ ซึ่งเราจะแบ่งช่วงของอุณหภูมิได้ทั้งหมด 3 แบบ คือ อุณหภูมิต่ำกว่าที่ตั้งไว้ สูงกว่าที่ตั้งไว้ และอุณหภูมิปกติคืออยู่ในขอบเขตที่เราได้กำหนดไว้ หากอุณหภูมิที่กำลังวัดอยู่นั้นเกิดขอบเขตที่ได้กำหนดไว้แอปพลิเคชันจะมีการทำงานโดยมีการแจ้งเตือนด้วยข้อความพร้อมเสียงปรากฏที่หน้าจอสมาร์ตโฟน

จากการทดลองในบทที่ 4 ส่วนของการวัดค่าและควบคุมอุณหภูมิความชื้น จากการทดสอบจะเห็นว่าอุณหภูมิจะแปรผกผันกับความชื้น กล่าวคือเมื่ออุณหภูมิสูงขึ้นความชื้นในอากาศจะลดลงอันเนื่องมาจากไดร์เป่าผมให้ความร้อน ทำให้ความชื้นในอากาศระเหยไป และเมื่อเพิ่มความชื้น(ไอน้ำ)ภายในห้องจะทำให้อุณหภูมิในห้องลดลง ค่าที่ได้จากแอปพลิเคชั่นจะมีค่าที่ใกล้เคียงกับค่ามาตรฐานเมื่อเทียบกับ Thermo-Hygrometer รุ่น NT-312 กล่าวคือ เมื่อนำค่าที่ได้จากเครื่องวัดอุณหภูมิมาหาค่าความคลาดเคลื่อน ผลปรากฏว่ามีความคลาดเคลื่อนอยู่ที่ระหว่าง 0 – 10 เปอร์เซ็นเท่านั้น ซึ่งถือว่ามีความคลาดเคลื่อนเพียงเล็กน้อยเท่านั้น โดยความคลาดเคลื่อนที่เกิดขึ้น เกิดจากการที่เราไม่สามารถทำการควบคุมอุณหภูมิโดยรอบตลอดการทดลองได้ และระยะห่าง ของการวัดของแต่ละสถานที่หรืออุปกรณ์ไม่เท่ากัน

ผู้ใช้สามารถนำบอร์ด IOIO-Q ไปใช้ตามสถานที่ต่างๆได้ตามความเหมาะสม สามารถเรียกดูข้อมูลค่าอุณหภูมิและความชื้นได้สะดวกขึ้น เนื่องจากสามารถดูได้ผ่านทางแอปพลิเคชันบน สมาร์ตโฟนซึ่งง่ายต่อการพกพาติดตัวทำให้การควบคุมอุณหภูมิและความชื้นเป็นไปได้อย่างสะดวกรวดเร็วข้อดีของระบบคือผู้ใช้งานไม่จำเป็นต้องตรวจสอบอุณหภูมิและความชื้นอยู่ตลอดเวลา เพราะผู้ใช้งานสามารถตั้งค่าอุณหภูมิและความชื้นที่เหมาะสมไว้บนแอปพลิเคชันได้เลย

5.2 อุปสรรคและปัญหาของโครงงาน

5.2.1 ระยะในการรับและส่งข้อมูลจะไปได้ใกล้หรือไกลขึ้นอยู่กับระยะสัญญาณบลูทูธที่ใช้ ซึ่งถ้าสัญญาณบลูทูธสามารถส่งสัญญาณได้เพียงระยะสั้น ๆ เราก็ไม่สามารถวัดอุณหภูมิและความชื้นที่ระยะเกินกว่านั้นได้

5.2.2 ในการเชื่อมต่อข้อมูลจากบอร์ด IOIO-Q มายังแอปพลิเคชันอาจจะมีการหน่วงเวลาอยู่บ้าง ทำให้การส่งข้อมูลมีความล่าช้าเล็กน้อย

5.3 ข้อเสนอแนะ

ก่อนจะทำการวัดอุณหภูมิและความชื้นเราต้องทำการตรวจสอบระยะสัญญาณบลูทูธด้วยว่ามีระยะในการรับส่งเท่าไร จะได้ไม่เกิดข้อพลาดขณะทำการควบคุมอุณหภูมิและความชื้นสำหรับพื้นที่ที่ต้องการควบคุมอุณหภูมิและความชื้นอยู่ตลอดเวลา ความล่าช้าที่เกิดขึ้นจากการเชื่อมต่อหรือแม้กระทั่งการรับส่งข้อมูล อาจแก้ไขได้โดยการกำหนดอัตราการหน่วงเวลาให้น้อยลงจากเดิม แต่ทั้งนี้ทั้งนั้นก็ต้องขึ้นอยู่กับขีดจำกัดของอุปกรณ์นั้น ๆ ด้วย

เอกสารอ้างอิง

สมเกียรติ กิจวงศ์วัฒนะ. (2550). **Android กับการต่อวงจรอิเล็กทรอนิกส์.** กรุงเทพฯ : บริษัท อินโนเวตีฟ เอ็กเพอริเมนต์ จำกัด.

นางสาวณัฐธิดา มากสังข์ และนางสาวจินดารัตน์ วรรณคุณ. (2557). **ระบบตรวจสอบอุณหภูมิด้วยสมาร์ทโฟนผ่านระบบปฏิบัติการแอนดรอยด์**. ปริญญานิพนธ์ วศ.บ. (วิศวกรรมไฟฟ้า) กรุงเทพฯ : บัณฑิตวิทยาลัย. มหาวิทยาลัยศรีนครินทรวิโรฒ

สมเกียรติ กิจวงศ์วัฒนะ. (2555, กุมภาพันธ์). IOIO กับการติดตั้งอุปกรณ์แอนดรอยด์แบบไร้สายผ่านบลูทูธ. **The Prototype Electronics Magazine**. (26): 42-44.

สมเกียรติ กิจวงศ์วัฒนะ. (2554, กรกฎาคม). IOIO บอร์ดอินพุตเอาต์พุตสำหรับสร้างอุปกรณ์เชื่อมต่อกับอุปกรณ์แอนดรอยด์. **The Prototype Electronics Magazine.** (21): 48-52.

สมเกียรติ กิจวงศ์วัฒนะ. (2558). **รู้จักกับบอร์ด IOIO กันแล้วหรือยัง**. สืบค้นเมื่อ 20 เมษายน 2558, จาก http://www.akexorcist.com/2013/11/ioio-board-ioio.html

ภาคผนวก

ภาคผนวก ก โค้ดที่ใช้ในการพัฒนาแอปพลิเคชัน ส่วนของหน้าจอหลัก ชื่อไฟล์ว่า Main.xml

รูปที่ ก1 หน้าจอหลักของแอปพลิเคชัน

1. MainActivity.java

```
packageioio.examples.hello;
importjava.text.SimpleDateFormat;
public class MainActivity extends IOIOActivity  {
TextViewtxtTemp, txtHum, stTemp, stHum, result;
ProgressBarprogressTemp,progressHum;
publicImageViewtempp, setup, hourly;
staticint tempmin,tempmax,hummin,hummax,a=0,b=0;
public String tempmin1,tempmax1,hummin1,hummax1,formattedDate1,TIME1;
        String datatemp, datahum ;
AnalogInputintemp, inhum;
publicMediaPlayer play1, play2, play3;
public Button add;
publicvoidonCreate(Bundle savedInstanceState)
```

```
super.onCreate(savedInstanceState);
setContentView(R.layout.main);
tempmin1 = getIntent().getStringExtra("vamintemp");
        tempmax1 = getIntent().getStringExtra("vamaxtemp");
        hummin1 = getIntent().getStringExtra("vaminhum");
        hummax1 = getIntent().getStringExtra("vamaxhum");

stTemp = (TextView)findViewById(R.id.stTemp);
    stHum = (TextView)findViewById(R.id.stHum);
    txtHum = (TextView)findViewById(R.id.rehum);
    txtTemp = (TextView)findViewById(R.id.retemp);
    progressTemp = (ProgressBar)findViewById(R.id.progressBarTemp);
    progressHum = (ProgressBar)findViewById(R.id.progressBarHum);
    progressTemp.setMax(100);
    progressHum.setMax(100);
play1 = MediaPlayer.create(MainActivity.this, R.drawable.a1);
play2 = MediaPlayer.create(MainActivity.this, R.drawable.a2);
play3 = MediaPlayer.create(MainActivity.this, R.drawable.a3);

  Calendar cc =Calendar.getInstance();
SimpleDateFormatdsf =newSimpleDateFormat("E-MM-dd-yyyy");
    String formattedDate1 = dsf.format(cc.getTime());

    Calendar b =Calendar.getInstance();
    SimpleDateFormat bb =newSimpleDateFormat("HH:mm");
    TIME1 =  bb.format(b.getTime());

    TextView result = (TextView)findViewById(R.id.result);
    result.setText("DATE: "+formattedDate1);
    result.setTextSize(18)
```

```
if(tempmin1 != null)
        {

tempmin = (Integer.parseInt(tempmin1))/10 ;
tempmax = (Integer.parseInt(tempmax1))/10 ;
hummin = (Integer.parseInt(hummin1))/10;
hummax = (Integer.parseInt(hummax1))/10 ;
                    }

tempp = (ImageView)findViewById(R.id.temp);
tempp.setEnabled( false );

setup = (ImageView)findViewById(R.id.setup);
setup.setOnClickListener(newView.OnClickListener() {
        publicvoidonClick(View v)
                Intent i = newIntent(MainActivity.this,pagesetup.class);
        startActivity(i);
                    }
 );
                hourly = (ImageView)findViewById(R.id.hourly);
                hourly.setOnClickListener(newView.OnClickListener() {
                        publicvoidonClick(View v)
                   {
                     Intent i = newIntent(MainActivity.this, his.class);
                startActivity(i);
                   }
             }
);

add = (Button)findViewById(R.id.add);
```

```
}
class Looper extends BaseIOIOLooper {
public void setup() throws ConnectionLostException, InterruptedException {
                    // Assign analog input with port on IOIO board
                              intemp = ioio_.openAnalogInput(37);
                              inhum = ioio_.openAnalogInput(46);
                                   }
public void loop() throws ConnectionLostException,  InterruptedException{
                    runOnUiThread(new Runnable() {
public void run() {
          try {
float value = intemp.getVoltage();
float value1 = inhum.getVoltage();

int temp =(int)(value*100);
int hum = (int)(((value1 - 0.958) / 0.0307)*(1));

datatemp = String.valueOf(temp);
datahum = String.valueOf(hum);

txtHum.setText(hum + "% RH");
progressHum.setProgress(hum);
                    txtHum.setText(hum + "% RH");
progressHum.setProgress(hum);

if(hum >= hummin&& hum <= hummax)
               {
stHum.setText("Status ::  Normal");
               }
elseif(hummin == 0 || hummax == 0)
```

```
            {
stHum.setText("Status ::  Normal");
}


elseif(hum <= hummin || hum >= hummax)
            {
stHum.setText("Status ::  Alert");
a = 1;
            }

txtTemp.setText( temp + " C");

        progressTemp.setProgress(temp);

if(temp >= tempmin&& temp <= tempmax)
            {
                stTemp.setText("Status ::  Normal");
            }

elseif(tempmin == 0 || tempmax == 0)
            {
                stTemp.setText("Status ::  Normal");
            }

elseif(temp <= tempmin || temp >= tempmax)
            {
stTemp.setText("Status ::  Alert");
b = 1;
            }
```

```
if(a+b == 2)
                {
play3.start();
a=0;
b=0;
                }
elseif(a==1){
play1.start();
a=0;
               }
elseif(b==1){
play2.start();
b=1;
               }
add.setOnClickListener(newView.OnClickListener() {
publicvoidonClick(View v){
//ShowDialogg();
try {
          String id = "null";

ArrayList<NameValuePair>nameValuePairs = new
          ArrayList<NameValuePair>();
nameValuePairs.add(newBasicNameValuePair("isAdd", "true"));
nameValuePairs.add(newBasicNameValuePair("ID", id));
nameValuePairs.add(newBasicNameValuePair("DATE", formattedDate1));
nameValuePairs.add(newBasicNameValuePair("TIME", TIME1));
nameValuePairs.add(newBasicNameValuePair("TEMP", datatemp));
nameValuePairs.add(newBasicNameValuePair("HUM", datahum));
HttpClienthttpclient = newDefaultHttpClient();
```

```
HttpPosthttppost = newHttpPost("http://192.168.1.134/addDATA.php");
httpclient.execute(httppost);
Toast.makeText(MainActivity.this,"Save Complete"  ,
Toast.LENGTH_LONG).show()
   }
catch (Exception e) {

Log.d("log_err", "Error in http connection " + e.toString());

          }
        }
      }
   );

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

 }
catch (ConnectionLostException e) {
e.printStackTrace();
 }
   }
    });
       }
          publicvoid disconnected() {
          publicvoid incompatible() {
                                    }
                }
          publicIOIOLoopercreateIOIOLooper() {
          returnnewLooper();
```

```
        }
}
```

2. ไฟล์ Main.xml ในส่วนของหน้าจอหลัก

```
<?xml version="1.0" encoding="utf-8"?>
<AbsoluteLayoutxmlns:android="http://schemas.android.com/apk/res/android"
android:id="@+id/AbsoluteLayout1"
android:layout_width="fill_parent"
android:layout_height="fill_parent"
android:background="@drawable/vbb">

<ProgressBar
android:id="@+id/progressBarHum"
style="?android:attr/progressBarStyleHorizontal"
android:layout_width="137dp"
android:layout_height="wrap_content"
android:layout_x="86dp"
android:layout_y="348dp" />

<ProgressBar
android:id="@+id/progressBarTemp"
style="?android:attr/progressBarStyleHorizontal"
android:layout_width="132dp"
android:layout_height="wrap_content"
android:layout_x="91dp"
android:layout_y="232dp" />

<ImageView
android:id="@+id/temp"
android:layout_width="74dp"
```

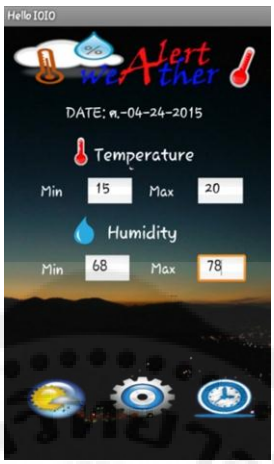
```
android:layout_height="50dp"
android:layout_x="26dp"
android:layout_y="423dp"
android:background="@drawable/weather" />

<ImageView
android:id="@+id/setup"
android:layout_width="79dp"
android:layout_height="51dp"
android:layout_x="127dp"
android:layout_y="422dp"
android:background="@drawable/mascot" />

<ImageView
android:id="@+id/hourly"
android:layout_width="79dp"
android:layout_height="52dp"
android:layout_x="225dp"
android:layout_y="421dp"
android:background="@drawable/ll" />

<TextView
android:id="@+id/retemp"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_x="237dp"
android:layout_y="232dp"
android:text="0 C"
android:textAppearance="?android:attr/textAppearanceMedium" />
```

ภาคผนวก ข โค้ดที่ใช้ในการพัฒนาแอปพลิเคชัน ส่วนของการกำหนดค่าต่ำที่สุดและสูงที่สุดของอุณหภูมิและความชื้นสัมพัทธ์



รูปที่ ข1 หน้าจอของแอปพลิเคชัน สำหรับกำหนดค่าต่ำที่สุดและสูงที่สุดในการวัด

1. Pagesetup.java

```
packageioio.examples.hello;
importjava.text.SimpleDateFormat;
public class pagesetup extends Activity  {
                EditTextmintemp, maxtemp, minhum, maxhum;
                static String vamintemp, vamaxtemp, vaminhum, vamaxhum;
                ImageView setup1, hourly1;
                TextView alert;

public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
                setContentView(R.layout.page2);

                        mintemp=(EditText)findViewById(R.id.mintemp);
                        maxtemp=(EditText)findViewById(R.id.maxtemp);
                        minhum=(EditText)findViewById(R.id.minhum);
```

```
                maxhum=(EditText)findViewById(R.id.maxhum);

                Calendar c =Calendar.getInstance();
                SimpleDateFormatdf =new SimpleDateFormat("E-MM-dd-yyyy");
                String formattedDate = df.format(c.getTime());
                TextView result1 = (TextView)findViewById(R.id.result1);
                result1.setText("DATE: "+formattedDate);
                result1.setTextSize(18);

                setup1 = (ImageView)findViewById(R.id.setup1);
                setup1.setEnabled ( false );
                 hourly1 = (ImageView)findViewById(R.id.hourly1);
                hourly1.setOnClickListener(new View.OnClickListener() {
                public void onClick(View v)
                        {
                            Intent i = new Intent(pagesetup.this, his.class);
                            startActivity(i);
                        }
                    }
        ImageView  temp1 = (ImageView)findViewById(R.id.temp1);
        temp1.setOnClickListener(new View.OnClickListener() {

public void onClick(View v)
                {
                    vamintemp = mintemp.getText().toString()+"0";
                    vamaxtemp = maxtemp.getText().toString()+"0";
                    vaminhum  =minhum.getText().toString()+"0";
                    vamaxhum  =maxhum.getText().toString()+"0";
                    int check = Integer.parseInt(vamintemp);
            if(check == 0 )
```

```
                    {
        Toast.makeText(pagesetup.this,"please set value

                    showdialog();
                    }
                else
                    {
                    showdialog1();
                    }
                );
public void showdialog(){

        finalAlertDialog.BuilderdDialog = new AlertDialog.Builder(this);
                dDialog.setTitle("Require! ");
                dDialog.setMessage("Your don't set value Please set value");
                dDialog.setIcon(android.R.drawable.btn_star_big_on);
dDialog.setPositiveButton("Set", new DialogInterface.OnClickListener() {
        public void onClick(DialogInterface dialog, intwhichButton) {
    }
        });
dDialog.setNegativeButton("Don't Set",newDialogInterface.OnClickListener(){
        public void onClick(DialogInterface dialog, intwhichButton) {

Intent setup = new Intent(getApplicationContext(),          MainActivity.class);
                setup.putExtra("vamintemp", vamintemp);
                setup.putExtra("vamaxtemp", vamaxtemp);
                setup.putExtra("vaminhum", vaminhum);
                setup.putExtra("vamaxhum", vamaxhum);
                startActivity(setup);
            }
```

```
dDialog.show();
        Public void showdialog1(){
        finalAlertDialog.BuilderdDialog = new AlertDialog.Builder(this);
                        dDialog.setTitle("Require! ");
                        dDialog.setMessage("Your set value complete");
                        dDialog.setIcon(android.R.drawable.btn_star_big_on);

dDialog.setPositiveButton("OK", new DialogInterface.OnClickListener() {
public void onClick(DialogInterface dialog, intwhichButton) {
   Intent setup = new Intent(getApplicationContext(), MainActivity.class);
                setup.putExtra("vamintemp", vamintemp);
                setup.putExtra("vamaxtemp", vamaxtemp);
                setup.putExtra("vaminhum", vaminhum);
                setup.putExtra("vamaxhum", vamaxhum);
                startActivity(setup);
         }
    });

dDialog.setNegativeButton("Cancle", new DialogInterface.OnClickListener() {
        public void onClick(DialogInterface dialog, intwhichButton) {
                Toast.makeText(pageseupt.this,"set value again",
                        Toast.LENGTH_LONG).show();
          }
        });
dDialog.show();
}
}
```

2. ไฟล์ page2.xml ส่วนการกำหนดค่าต่ำสุด-สูงสุด ของอุณหภูมิและความชื้นสัมพัทธ์

```
<?xml version="1.0" encoding="utf-8"?>
<AbsoluteLayoutxmlns:android="http://schemas.android.com/apk/res/android"
android:id="@+id/AbsoluteLayout1"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:background="@drawable/vbb"
android:focusable="false">

<ImageView
android:id="@+id/temp1"
android:layout_width="74dp"
android:layout_height="50dp"
android:layout_x="26dp"
android:layout_y="423dp"
android:background="@drawable/weather" />

<ImageView
android:id="@+id/setup1"
android:layout_width="79dp"
android:layout_height="51dp"
android:layout_x="127dp"
android:layout_y="422dp"
android:background="@drawable/mascot" />

<ImageView
android:id="@+id/hourly1"
android:layout_width="79dp"
android:layout_height="52dp"
android:layout_x="225dp"
```

```
android:layout_y="421dp"
android:background="@drawable/ll" />

<ImageView
android:id="@+id/imageView1"
android:layout_width="match_parent"
android:layout_height="117dp"
android:layout_x="6dp"
android:layout_y="-17dp"
android:src="@drawable/goo" />

<TextView
android:id="@+id/result1"
android:layout_width="227dp"
android:layout_height="wrap_content"
android:layout_x="74dp"
android:layout_y="91dp"
android:text="time"
android:textAppearance="?android:attr/textAppearanceLarge" />

<TextView
android:id="@+id/textView1"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_x="108dp"
android:layout_y="138dp"
android:text="Temperature"
android:textAppearance="?android:attr/textAppearanceLarge" />

<ImageView
```

```
android:id="@+id/imageView2"
android:layout_width="66dp"
android:layout_height="40dp"
android:layout_x="59dp"
android:layout_y="132dp"
android:src="@drawable/grad" />

<EditText
android:id="@+id/mintemp"
android:layout_width="58dp"
android:layout_height="38dp"
android:layout_x="98dp"
android:layout_y="181dp"
android:ems="2"
android:inputType="number">

<requestFocus />
</EditText>

<TextView
android:id="@+id/textView2"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_x="124dp"
android:layout_y="232dp"
android:text="Humidity"
android:textAppearance="?android:attr/textAppearanceLarge" />

<ImageView
android:id="@+id/imageView3"
```

```
android:layout_width="24dp"
android:layout_height="34dp"
android:layout_x="84dp"
android:layout_y="228dp"
android:src="@drawable/water" />

<TextView
android:id="@+id/textView5"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_x="176dp"
android:layout_y="279dp"
android:text="Max"
android:textAppearance="?android:attr/textAppearanceMedium" />
<TextView
android:id="@+id/textView4"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_x="44dp"
android:layout_y="186dp"
android:text="Min"
android:textAppearance="?android:attr/textAppearanceMedium" />
<TextView
android:id="@+id/textView3"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_x="174dp"
android:layout_y="186dp"
android:text="Max"
android:textAppearance="?android:attr/textAppearanceMedium" />
```

ประวัติย่อนิสิตผู้ทำโครงงาน

ชื่อ-สกุล นางสาวบุษราภรณ์ ศรีคำนวน

วัน เดือน ปี 24 ตุลาคม 2535

สถานที่เกิด จังหวัดปทุมธานี

สถานที่อยู่ปัจจุบัน 32/1 หมู่.9 ต.บางคูวัด อ.เมือง
จ.ปทุมธานี 12000

โทรศัพท์ 089-173-2162

E-mail kapangyaii@gmail.com

ประวัติการศึกษา

พ.ศ.2548 ระดับมัธยมศึกษาตอนต้น โรงเรียนปทุมวิไล จังหวัดปทุมธานี

พ.ศ.2551 ระดับมัธยมศึกษาตอนปลาย โรงเรียนปทุมวิไล จังหวัดปทุมธานี

พ.ศ.2554 แขนงวิศวกรรมโทรคมนาคม ภาควิชาไฟฟ้า คณะวิศวกรรมศาสตร์ มหาวิทยาลัยศรีนครินทรวิโรฒ

ประวัติย่อนิสิตผู้ทำโครงงาน

ชื่อ-สกุล นางสาวพิชญา บัวประดิษฐ์

วัน เดือน ปี 3 มิถุนายน 2536

สถานที่เกิด จังหวัดพระนครศรีอยุธยา

สถานที่อยู่ปัจจุบัน 11/17 หมู่บ้านเพชรศิริวิลล่า หมู่ 3
ตำบลเจ้าเจ็ด อำเภอเสนา
จังหวัดพระนครศรีอยุธยา 13110

โทรศัพท์ 087-0027679

E-mail pichaya.bpd@gmail.com

ประวัติการศึกษา

พ.ศ.2548	ระดับมัธยมศึกษาตอนต้น โรงเรียนปทุมวิไล จังหวัดปทุมธานี
พ.ศ.2551	ระดับมัธยมศึกษาตอนปลาย โรงเรียนปทุมวิไล จังหวัดปทุมธานี
พ.ศ.2554	แขนงวิศวกรรมโทรคมนาคม ภาควิชาไฟฟ้า คณะวิศวกรรมศาสตร์ มหาวิทยาลัยศรีนครินทรวิโรฒ

ประวัติย่อนิสิตผู้ทำโครงงาน

ชื่อ-สกุล นางสาวศิวาพร ทองสุกดี

วัน เดือน ปี 1 พฤศจิกายน 2534

สถานที่เกิด จังหวัดนครสวรรค์

สถานที่อยู่ปัจจุบัน 32/7 หมู่ 8 ต.หนองนมวัว อ.ลาดยาว
จ.นครสวรรค์ 60150

โทรศัพท์ 080-0828533

E-mail siwaporn.sd@gmail.com

ประวัติการศึกษา

พ.ศ.2547	ระดับมัธยมศึกษาตอนต้น โรงเรียนสตรีนครสวรรค์ จ.นครสวรรค์
พ.ศ.2550	ระดับมัธยมศึกษาตอนปลาย โรงเรียนสตรีนครสวรรค์ จ.นครสวรรค์
พ.ศ.2554	แขนงวิศวกรรมโทรคมนาคม ภาควิชาไฟฟ้า คณะวิศวกรรมศาสตร์ มหาวิทยาลัยศรีนครินทรวิโรฒ