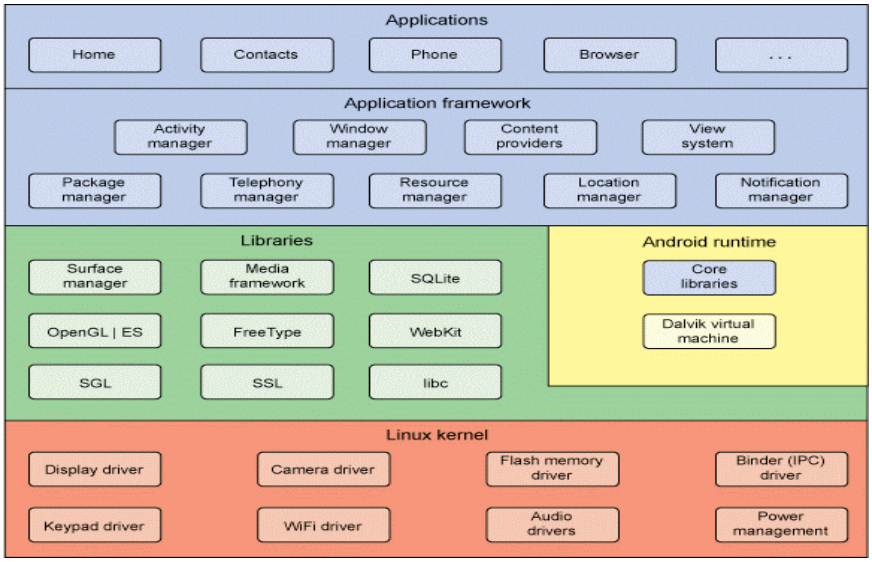
บทที่ 2
แนวคิดและทฤษฎีที่เกี่ยวข้อง

2.1 สถาปัตยกรรมแอนดรอยด์ (Android Architecture)[1]

แอนดรอยด์เป็นซอฟต์แวร์ที่มีโครงสร้างแบบเรียงทับซ้อนหรือแบบสแต็ก (Stack) ซึ่งรวมเอาระบบปฏิบัติการ (Operating System), มิดเดิลแวร์ (Middleware) และแอพพลิเคชันที่สำคัญเข้าไว้ด้วยกัน เพื่อใช้สำหรับทำงานบนอุปกรณ์พกพาเคลื่อนที่ (Mobile Devices) เช่น โทรศัพท์มือถือ เป็นต้น

การทำงานของแอนดรอยด์มีพื้นฐานอยู่บนระบบลีนุกซ์ เคอร์เนล (Linux Kernel) ซึ่งใช้ Android SDK (Software Development Kit) เป็นเครื่องมือสำหรับการพัฒนาแอพพลิเคชันบนระบบปฏิบัติการ Android และใช้ภาษา Java ในการพัฒนา สถาปัตยกรรมของแอนดรอยด์ (Android Architecture) นั้นถูกแบ่งออกเป็นลำดับชั้น ออกเป็น 4 ชั้นหลักๆ



รูปที่ 2.1 สถาปัตยกรรมแอนดรอยด์ (Android Architecture)

---

[1]สถาปัตยกรรมแอนดรอยด์ [ออนไลน์], 25 พฤษภาคม 2558.
แหล่งที่มา http://kadroidz.blogspot.com/android- architecture.html

2.1.1ชั้นแอพพลิเคชัน (Application Layer)[1]

ชั้นนี้จะเป็นชั้นที่อยู่บนสุดของโครงสร้างสถาปัตยกรรม Android ซึ่งเป็นส่วนของแอพพลิเคชันที่พัฒนาขึ้นมาใช้งาน เช่น แอพพลิเคชันรับ/ส่งอีเมล์, SMS, ปฏิทิน, แผนที่, เว็บเบราเซอร์, รายชื่อผู้ติดต่อ เป็นต้น ซึ่งแอพพลิเคชันจะอยู่ในรูปแบบของไฟล์ .apk โดยทั่วไปแล้วจะอยู่ในไดเร็คทอรี data/app

รูปที่ 2.2 ชั้นแอพพลิเคชัน (Application Layer)

2.1.2ชั้นแอพพลิเคชันเฟรมเวิร์ค (Application Framework Layer)[1]

ในชั้นนี้จะอนุญาตให้นักพัฒนาสามารถเข้าเรียกใช้งาน โดยผ่าน API (Application Programming Interface) ซึ่ง Android ได้ออกแบบไว้เพื่อลดความซ้ำซ้อนในการใช้งาน application component โดยในชั้นนี้ประกอบด้วยแอพพลิเคชันเฟรมเวิร์คดังนี้

- View System เป็นส่วนที่ใช้ในการควบคุมการทำงานสำหรับการสร้างแอพพลิเคชัน เช่น lists, grids, text boxes, buttons และ embeddable web browser
- Location Manager เป็นส่วนที่จัดการเกี่ยวกับตำแหน่งของอุปกรณ์พกพาเคลื่อนที่
- Content Provider เป็นส่วนที่ใช้ควบคุมการเข้าถึงข้อมูลที่มีการใช้งานร่วมกัน (Share data) ระหว่างแอพพลิเคชันที่แตกต่างกัน เช่น ข้อมูลผู้ติดต่อ (Contact)
- Resource Manager เป็นส่วนที่จัดการข้อมูลต่างๆ ที่ไม่ใช่ส่วนของโค้ดโปรแกรม เช่น รูปภาพ, localized strings, layout ซึ่งจะอยู่ในไดเร็คทอรี res/
- Notification Manager เป็นส่วนที่ควบคุมอีเวนต์ (Event) ต่างๆ ที่แสดงบนแถบสถานะ (Status bar) เช่น ในกรณีที่ได้รับข้อความหรือสายที่ไม่ได้รับและการแจ้งเตือนอื่นๆ
- Activity Manager เป็นส่วนควบคุม Life Cycle ของแอพพลิเคชัน

---

[1]สถาปัตยกรรมแอนดรอยด์ [ออนไลน์], 25 พฤษภาคม 2558.
แหล่งที่มา http://kadroidz.blogspot.com/android- architecture.html

รูปที่ 2.3 ชั้นแอพพลิเคชันเฟรมเวิร์ค (Application Framework Layer)

2.1.3 ชั้นไลบรารี (Library Layer)[1]

Android ได้รวบรวมกลุ่มของไลบรารีต่างๆ ที่สำคัญและมีความจำเป็นเอาไว้มากมาย เพื่ออำนวยความสะดวกให้กับนักพัฒนาและง่ายต่อการพัฒนาโปรแกรม โดยตัวอย่างของไลบรารีที่สำคัญเช่น

- System C library เป็นกลุ่มของไลบรารีมาตรฐานที่อยู่บนพื้นฐานของภาษา C ไลบรารี (libc) สำหรับ embedded system ที่มีพื้นฐานมาจาก Linux
- Media Libraries เป็นกลุ่มการทำงานมัลติมีเดีย เช่น MPEG4, H.264, MP3, AAC, AMR, JPG, และ PNG
- Surface Manager เป็นกลุ่มการจัดการรูปแบบหน้าจอ การวาดหน้าจอ
- 2D/3D library เป็นกลุ่มของกราฟิกแบบ 2 มิติ หรือ SGL (Scalable Graphics Library) และแบบ 3 มิติ หรือ OpenGL
- FreeType เป็นกลุ่มของบิตแมป (Bitmap) และเวคเตอร์ (Vector) สำหรับการเรนเดอร์ (Render) ภาพ
- SQLite เป็นกลุ่มของฐานข้อมูล โดยนักพัฒนาสามารถใช้ฐานข้อมูลนี้เก็บข้อมูลแอพพลิเคชันต่างๆ ได้
- Browser Engine เป็นกลุ่มของการแสดงผลบนเว็บเบราเซอร์โดยอยู่บนพื้นฐานของ Webkit ซึ่งจะมีลักษณะคล้ายกับ Google Chrome

---

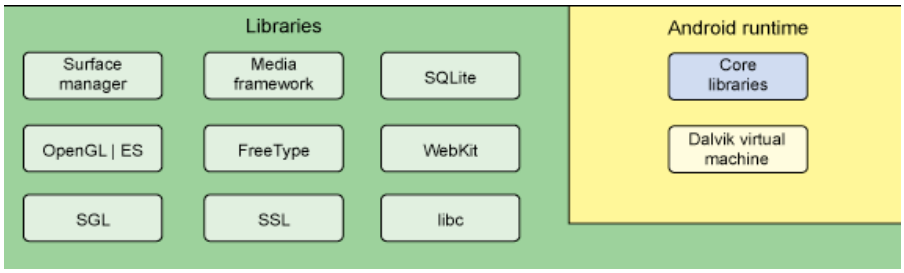
[1]สถาปัตยกรรมแอนดรอยด์ [ออนไลน์], 25 พฤษภาคม 2558.
แหล่งที่มา http://kadroidz.blogspot.com/android- architecture.html

รูปที่2.4 ชั้นไลบรารี (Library Layer)

2.1.3.1 Android Runtime[1]

เป็นชั้นย่อยที่อยู่ในชั้นไลบรารี ซึ่งจะประกอบด้วย 2 ส่วนหลักคือ

- Dalvik VM (Virtual Machine) ส่วนนี้ถูกเขียนด้วยภาษา Java เพื่อใช้เฉพาะการใช้งานในอุปกรณ์เคลื่อนที่ Dalvik VM จะแตกต่างจาก Java VM (Virtual Machine) คือ Dalvik VM จะรันไฟล์ .dex ที่คอมไพล์มาจากไฟล์ .class และ .jar โดยมี tool ที่ชื่อว่า dx ทำหน้าที่ในการบีบอัดคลาส Java ทั้งนี้ไฟล์ .dex จะมีขนาดกะทัดรัดและเหมาะสมกับอุปกรณ์เคลื่อนที่มากกว่า .class เพื่อต้องการใช้พลังงานจากแบตเตอรี่อย่างมีประสิทธิภาพสูงสุด
- Core Java Library ส่วนนี้เป็นไลบรารีมาตรฐาน แต่ก็มีความแตกต่างจากไลบรารีของ Java SE (Java Standard Edition) และ Java ME (Java Mobile Edition)
-

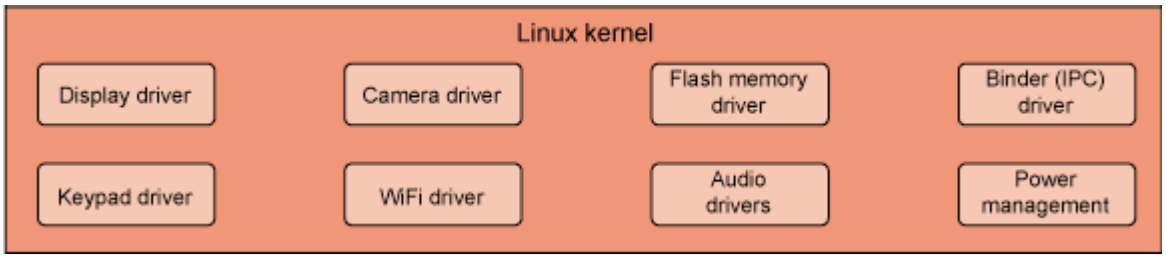


รูปที่ 2.5 Android Runtime

[1]สถาปัตยกรรมแอนดรอยด์ [ออนไลน์], 25 พฤษภาคม 2558.
แหล่งที่มา http://kadroidz.blogspot.com/android- architecture.html

2.1.4 ชั้นลีนุกซ์เคอร์เนล (Linux Kernel Layer)[1]

ระบบ Android นั้นถูกสร้างบนพื้นฐานของระบบปฏิบัติการ Linux โดยในชั้นนี้จะมีฟังก์ชันการทำงานหลายๆ ส่วน แต่โดยส่วนมากแล้วจะเกี่ยวข้องกับฮาร์ดแวร์โดยตรง เช่น การจัดการหน่วยความจำ (Memory Management) การจัดการโพรเซส (Process Management) การเชื่อมต่อเครือข่าย (Networking) เป็นต้น



รูปที่ 2.6 ชั้นลีนุกซ์เคอร์เนล (Linux Kernel Layer)

[1]สถาปัตยกรรมแอนดรอยด์ [ออนไลน์], 25 พฤษภาคม 2558.
แหล่งที่มา http://kadroidz.blogspot.com/android- architecture.html

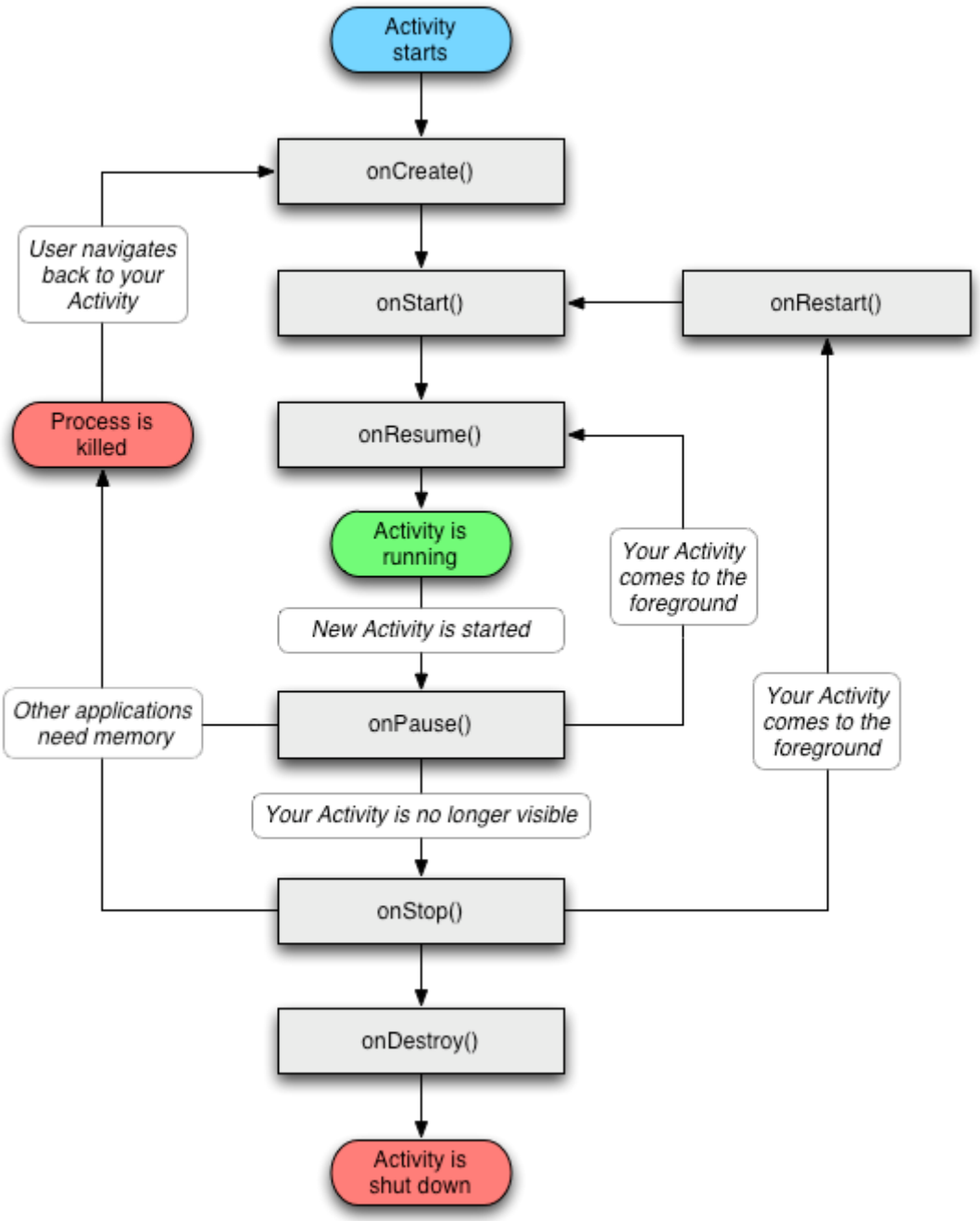
2.2 วงรอบชีวิตของแอพพลิเคชัน (Application Life Cycle)[2]

โดยปกติแอพพลิเคชันจะทำงานแยกกันในแต่ละโพรเซส และแต่ละโพรเซสอาจจะมี Activity/Service ทำงานอยู่มากกว่า 1 Activity/Service ดังนั้น ในแต่ละแอพพลิเคชันอาจจะมีมากกว่า 1 Activity ซึ่งในการเริ่มทำงานของ Activity จะเริ่มด้วย startActivity() สำหรับแบบซิงโครนัส(Synchronous) และจะเริ่มด้วย startSubActivity() สำหรับแบบอะซิงโครนัส (Asynchronous) โดยในแต่ละ Activity จะมีวงรอบชีวิต (Life Cycle) ที่แยกจากกันโดยชัดเจน ซึ่งมีสถานะการทำงานหลักดังนี้

- **onCreate** (Bundle savedInstanceState) ส่วนนี้จะถูกเรียกใช้งานเมื่อเริ่มทำงาน ในกรณีที่มีการเรียกใช้งานเมธอด (Method) นี้ Android Framework จะนำ Bundle object บันทึกไว้ใน Activity ก่อนที่ Activity จะทำงาน จากนั้นจะตามด้วยฟังก์ชั่น onStart()
- **onStart()** ส่วนนี้เป็นการระบุว่า Activity นั้นๆ จะถูกแสดงขึ้นมา จากนั้นสถานะจะถูกย้านไปเป็นสถานะ onResume แต่ถ้า Activity นั้นไม่สามารถทำงานได้ด้วยเหตุผลบางอย่าง สถานะจะถูกย้ายไปเป็นสถานะ onStop
- **onRestart()** ส่วนนี้จะเป็นการระบุว่า Activity นั้นจะถูกแสดงขึ้นมาอีกครั้งหนึ่ง ซึ่งจะตามด้วยสถานะ onStart()
- **onResume()** ส่วนนี้จะถูกเรียกเมื่อ Activity นั้นๆ มีการติดต่อกับผู้ใช้งาน เช่น นักพัฒนาต้องการเรียก Activity นั้นขึ้นมาทำงานอีกรอบหนึ่ง หลังจากที่ Activity นั้นอยู่ในสถานะ onPause
- **onPause()** ส่วนนี้จะถูกเรียกใช้เมื่อ Activity นั้นจะถูกเปลี่ยนไปเป็นการทำงานทางเบื้องหลัง (Background)
- **onStop()** ส่วนนี้จะถูกเรียกใช้งานเมื่อผู้ใช้ไม่ต้องการใช้งาน Activity นั้นๆ ในช่วงระยะเวลาหนึ่งๆ ซึ่งจะตามด้วยสถานะ onRestart() เมื่อต้องการกลับมาทำงานที่ Activity นั้นอีกครั้ง หรือตามด้วยสถานะ onDestroy() เมื่อต้องการปิด Activity นั้นๆ
- **onDestroy()** ส่วนนี้จะถูกเรียกเมื่อมีการปิดการทำงานของแต่ละ Activity

---

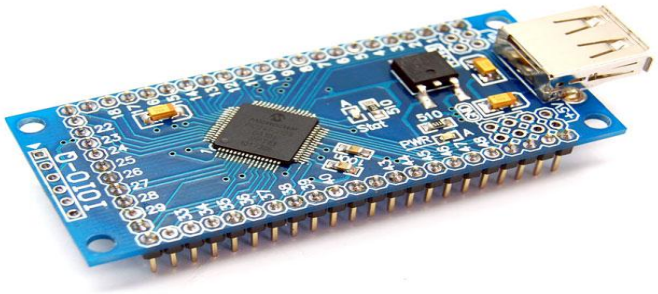
[2](จักรชัย โสอินทร์, พงษ์ศธร จันทร์ย้อย, ณัฐณิชา วีระมงคลเลิศ, 2555, หน้า 5)



รูปที่ 2.7 วงรอบชีวิตของแอพพลิเคชัน

2.3 IOIO-Q I/O ไมโครคอนโทลเลอร์ บอร์ด[3]

เป็นบอร์ดไมโครคอนโทรลเลอร์ที่เดิมทีเกิดมาเพื่อเชื่อมต่อกับแอนดรอยด์ โดยเฉพาะโดยต่างจากไมโครคอนโทรลเลอร์ตัวอื่นๆ เพราะปกติแล้วการเขียนโปรแกรมเชื่อมต่อกับแอนดรอยด์ไม่ว่าจะใช้บอร์ดไมโครคอนโทรลเลอร์ตัวใดก็ตาม จะต้องเขียนโปรแกรม ให้กับไมโครคอนโทรลเลอร์และต้องเขียนแอพพลิเคชั่นบนแอนดรอยด์เพื่อให้สามารถเชื่อมต่อและส่งข้อมูลระหว่างกันได้บอร์ดไมโครทั่วๆ ไป



รูปที่ 2.8 IOIO-Q I/Q Microcontroller Board

ทั้งนี้ขึ้นอยู่กับวิธีในการเชื่อมต่อ ที่ง่ายสุดคือผ่านสัญญาณบลูทูธหรือWi-Fi จึงเป็นบอร์ด IOIO จะแตกต่างจากบอร์ดทั่วๆ ไป เพราะผลิตมาเพื่อเชื่อมต่อและถูกสั่งงานจากแอนดรอยด์ (Real Time) ไม่สามารถทำงานได้ด้วยตัวเอง ต้องรอคำสั่งจากแอนดรอยด์เท่านั้น เนื่องจากการที่พัฒนามาเพื่อแอนดรอยด์ ผู้พัฒนาจึงทำให้ผู้ใช้งานไม่จำเป็นต้องเขียนชุดคำสั่งให้กับบอร์ด IOIO เพราะจะมีชุดคำสั่งใส่มาในบอร์ดให้พร้อมไว้เรียบร้อยแล้วหรือที่เรียกกันว่า เฟิร์มแวร์

ดังนั้นผู้ใช้งานจึงเขียนชุดคำสั่งแค่ฝั่งแอนดรอยด์เท่านั้น โดยผู้ผลิตจะมีไลบรารีของบอร์ด IOIO ให้ใช้ในชุดคำสั่งฝั่งแอนดรอยด์เลย ดังนั้นจึงสามารถสั่งงานบอร์ด IOIO ด้วยคำสั่งในแอพพลิเคชั่น

---

[3] IOIO-Q I/O Basic [ออนไลน์], 28 พฤษภาคม 2558.
แหล่งที่มา http://www.akexorcist.com/2013/11/ioio-board-ioio.html

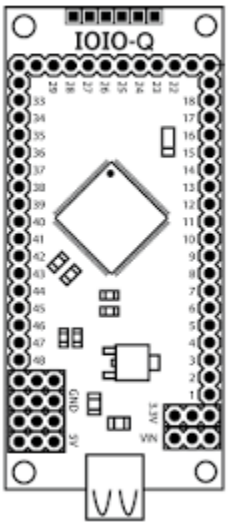
คุณสมบัติของบอร์ด[3]

- ใช้ไมโครคอนโทรลเลอร์เบอร์ PIC24FJ128DA ที่มีโมดูล USB OTG อยู่ภายในจึงสามารถทำงานเป็น USB โฮสต์ได้ และบรรจุเฟิร์มแวร์ IOIO มาพร้อมใช้งานทำให้การพัฒนาแอปพลิเคชั่นกระทำทางฝั่งอุปกรณ์แอนดรอยด์เท่านั้นไม่ต้องเขียนโปรแกรมลงในไมโครคอนโทรลเลอร์อีกมีพอร์ตอินพุตเอาต์พุต 48 ช่อง 13
- มีอินพุตอนาล็อก16 ช่อง ต่อเข้ากับโมดูลแปลงสัญญาณอนาล็อกเป็นดิจิตอลความละเอียด 10 บิตภายในตัวชิป
- มีเอาต์พุต PWM ความละเอียด 10 บิต 9 ช่อง
- มีขาต่อระบบบัส 2 สาย 3 ชุด รองรับการทางานกับบัส I2C
- มีคอนเน็กเตอร์ USB แบบ A ทำให้ใช้สายเชื่อมต่อพอร์ต USB ที่มีมากับอุปกรณ์แอนดรอยด์ในการเชื่อมต่อได้ทันที โดยไม่ต้องดัดแปลงใดๆ
- มี LED แสดงผลการทำงานและ LED แสดงสถานะไฟเลี้ยงไฟเลี้ยง 5 ถึง 12V
- แหล่งจ่ายไฟบนบอร์ด มี 2 ชุดคือ +3.3V สาหรับเลี้ยงวงจรและไมโครคอนโทรลเลอร์ PIC24FJ128 และ +5V 1500mA แบบสวิตชิ่งทำให้นำไปเลี้ยงอุปกรณ์แอนดรอยด์ที่นำมาต่อร่วมด้วยได้
- มีซ็อกเก็ตรองรับบอร์ด IOIO รวมถึงจุดต่อที่ใช้ในการอัพเดตเฟิร์มแวร์ด้วย
- จัดสรรขาพอร์ตใช้งานของ IOIO ทั้งหมดออกมาเป็นคอนเน็กเตอร์ IDC ตัวเมียและตัวผู้เพื่อความสะดวกในการต่อใช้งาน
- มีจุดต่อ PICkit3 สาหรับการอัปเกรดเฟิร์มแวร์ในอนาคต
- มีจุดต่ออะแดปเตอร์ไฟตรง +6.5V ถึ ง +9V พร้อมสวิตช์ เปิดปิด
- มีสวิตช์กดติดปล่อยดับ 1 ตัวต่อกับขาพอร์ต 20 เพื่อการทดสอบอ่านค่าอินพุตดิจิตอล
- มีตัวต้านทานปรับค่าได้ 1 ตัวต่อกับขาพอร์ต AN5 เพื่อการทดสอบอ่านค่าอินพุตอนาล็อก
- มีจุดจ่ายไฟเลี้ยง +3.3V และ +5V 500mA พร้อมกราวด์ สาหรับต่อทดลองอุปกรณ์ภายนอก

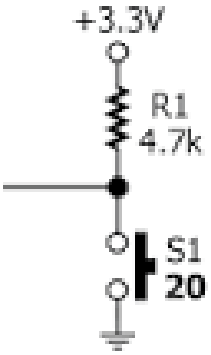
---

[3] IOIO-Q I/O Basic [ออนไลน์], 28 พฤษภาคม 2558.

แหล่งที่มา http://www.akexorcist.com/2013/11/ioio-board-ioio.html



รูปที่ 2.9 โครงสร้างของบอร์ด IOIO/Q IOIO Basic



รูปที่ 2.10 แสดงแหล่งจ่ายไฟเลี้ยงวงจรของบอร์ด

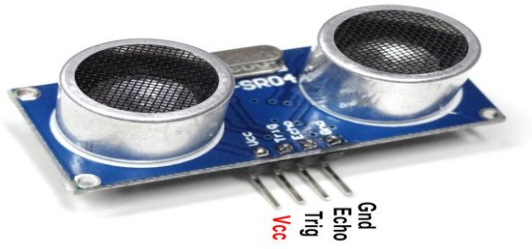
---

[3] IOIO-Q I/O Basic [ออนไลน์], 28 พฤษภาคม 2558.

แหล่งที่มา http://www.akexorcist.com/2013/11/ioio-board-ioio.html

2.4 HC-SR04 โมดูลวัดระยะทางด้วยคลื่นอัลตร้าโซนิก[4]

HC-SR04 เป็นโมดูลวัดระยะทางที่ใช้หลักการสะท้อนของคลื่นอัลตร้าโซนิกโดยตัว HC-SR04 มีแหล่งกำเนิด คลื่นอัลตร้าโซนิกส่งไปสะท้อนกับวัตถุ ที่อยู่ข้างหน้า กลับมายังตัวรับสัญญาณ โดยระยะทางที่วัดได้ จะสัมพันธ์กับระยะเวลาที่คลื่นอัลตร้าโซนิกเคลื่อนที่ไปกระทบวัตถุและสะท้อนกลับมายังตัวรับ เมื่อรู้ระยะเวลาที่คลื่นอัลตร้าโซนิกสะท้อนกลับมา จึงนำมาคำนวณหาเป็นระยะทางระหว่างโมดูล HC-SR04 กับวัตถุได้ โดย โมดูล HC-SR04 วัดระยะทางในช่วง 2 ถึง 500 ซม. (5 เมตร) มีความละเอียดอยู่ที่ 0.3 ซม. ใช้ไฟเลี้ยง +5V



รูปที่ 2.11 HC-SR04 Ultrasonic Module

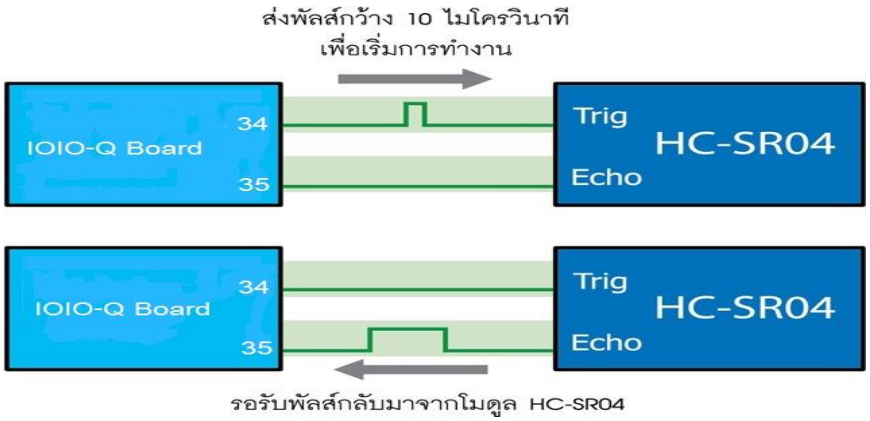
ในการสื่อสารข้อมูลกับโมดูล HC-SR04 ใช้ขาสัญญาณ 2 ขา คือ Trigger และ Echo โดยขา Trigger มีไว้สำหรับสั่งงานให้โมดูล HC-SR04 ส่งคลื่นอัลตร้าโซนิกออกไปข้างหน้า เมื่อคลื่นอัลตร้าโซนิกสะท้อนกลับมาจากวัตถุเป้าหมาย จะส่งสัญญาณพัลส์ออกมาทางขา Echo โดยสัญญาณนี้จะมีความกว้างที่สัมพันธ์กับระยะทางที่วัดได้ ดังนั้น ไมโครคอนโทรลเลอร์จะต้องส่งสัญญาณพัลส์ที่มีความกว้างพัลส์อย่างน้อย 10 ไมโครวินาทีไปยังขา Trigger ของโมดูล HC-SR04 แล้วรอรับสัญญาณพัลส์ที่ส่งกลับมาทางขา Echo เพื่อวัดความกว้างของสัญญาณพัลส์

---

[4]โมดูลวัดระยะทางด้วยอัลตร้าโซนิก [ออนไลน์], 28 พฤษภาคม 2558.
แหล่งที่มา http://doc.inex.co.th/hc-sr04-with- robo-creator/

จุดต่อใช้งานของโมดูล HC-SR04[4]

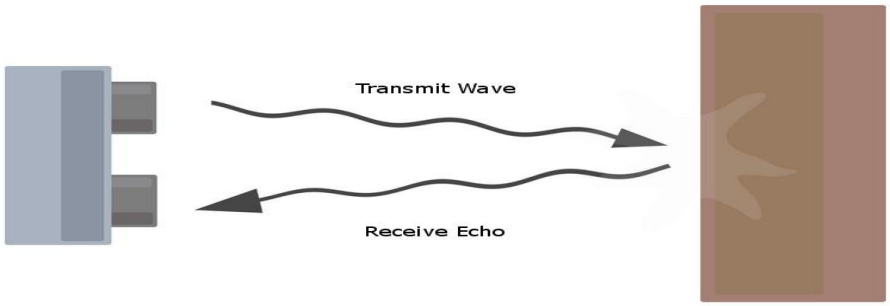
- ขาไฟเลี้ยง (+5V)สำหรับต่อไฟเลี้ยงแรงดัน +5V
- ขา Echo Pulse Output (ECHO) เป็นขาเอาต์พุตสำหรับส่งสัญญาณพัลส์ออกจาก HC-SR04 ซึ่งการใช้งานจะนำขานี้ไปต่อกับพอร์ตอินพุตของ ไมโครคอนโทรลเลอร์ เพื่อตรวจจับความกว้างของสัญญาณพัลส์ที่ส่งออกมาเพื่อแปลความหมายออกมาเป็นระยะทางอีกครั้งหนึ่ง
- ขา Trigger Pulse Input (TRIGGER) เป็นขาอินพุตรับสัญญาณพัลส์ที่มีความกว้างอย่างน้อย 10 ไมโครวินาทีเพื่อกระตุ้นการสร้างคลื่นอัลตร้าโซนิกความถี่ 40kHz ออกสู่อากาศจากตัวส่ง ดังนั้นเมื่อคลื่นความถี่นี้เคลื่อนที่ออกไปกระทบสิ่งกีดขวางที่อยู่เบื้องหน้าก็จะเกิดการสะท้อนกลับเข้ามายังตัวรับ และถูกแปลงเป็นความกว้างของสัญญาณพัลส์ที่จะส่งออกไปทางขา Echo Pulse Output นอกจากนี้ในโหมด 1 สัญญาณ จะใช้จุดนี้เป็นจุดสื่อสารข้อมูลอนุกรมเพื่อรับส่งค่าการวัดกับไมโครคอนโทรลเลอร์
- ขา GND สำหรับต่อกราวด์



รูปที่ 2.12 แสดงการติดต่อระหว่างบอร์ด IOIO-Q กับโมดูล HC-SR04 เพื่อวัดระยะทาง

---

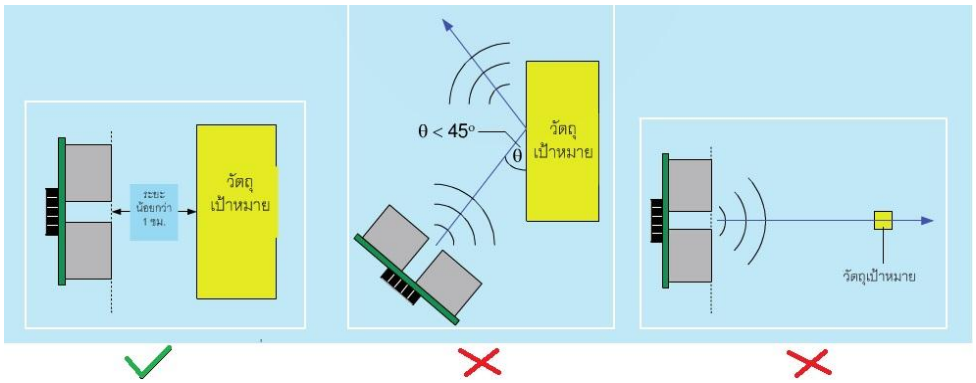
[4]โมดูลวัดระยะทางด้วยอัลตร้าโซนิก [ออนไลน์], 28 พฤษภาคม 2558.
แหล่งที่มา http://doc.inex.co.th/hc-sr04-with- robo-creator/



รูปที่ 2.13 แสดงการทำงานของ Ultrasonic Wave

2.4.1 ข้อจำกัดในการใช้งาน โมดูล HC-SR04

- วัตถุจะต้องอยู่ในตำแหน่งที่ขนาดกับ โมดูล HC-SR04
- วัตถุไม่ควรมีขนาดเล็กกว่า โมดูล HC-SR04
- วัตถุควรมีพื้นผิวเรียบไม่โค้งนูน เพื่อให้คลื่นสะท้อนกลับมีความแม่นยำ



รูปที่ 2.14 ข้อจำกัดในการใช้งาน โมดูล HC-SR04

---

[4]โมดูลวัดระยะทางด้วยอัลตร้าโซนิก [ออนไลน์], 28 พฤษภาคม 2558.
แหล่งที่มา http://doc.inex.co.th/hc-sr04-with- robo-creator/

2.5 Bluetooth Dongle

การเชื่อมต่อสมาร์ทโฟนกับ ไมโครคอนโทรลเลอร์ มีอยู่ 2 รูปแบบ คือ แบบ Wire ผ่านสาย USB และแบบ Wireless ผ่านการเชื่อมต่อสัญญาณ WIFI หรือสัญญาณ Bluetooth ซึ่งการจะใช้วิธีแบบไหนนั้นอาจขึ้นอยู่กับการใช้งาน รวมถึงการรองรับของอุปกรณ์ ถ้าต้องการควบคุมในระยะไกล หรือ ต้องการให้เกิดความสะดวกคล่องตัวมากขึ้นขณะใช้งานก็อาจใช้วิธีการแบบ Wireless ในการเชื่อมต่อ ซึ่งส่วนใหญ่แล้วจะใช้สัญญาณ Bluetooth โดยนิยมใช้ Bluetooth Dongle ซึ่ง Bluetooth Dongle นั้นมีให้เลือกใช้หลากหลาย Version ซึ่งแต่ละ Version นั้นมีคุณสมบัติที่แตกต่างกัน เช่นระยะทางที่จะสามารถติดต่อสื่อสารกับอุปกรณ์อื่นๆ

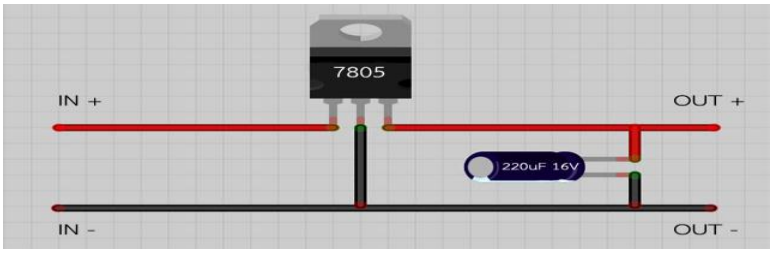
การใช้งาน Bluetooth Dongle แบบการเชื่อมต่อไร้สายระหว่าง ไมโครคอนโทรลเลอร์กับสมาร์ทโฟนนั้น ไม่ได้มีความยุ่งยากแต่ประการใด เนื่องจากทางผู้พัฒนาไมโครคอนโทรลเลอร์อย่าง IOIO-Q Broad นั้นได้เตรียม ไลบารีไว้ให้เรียบร้อยแล้ว ผู้ใช้สามารถเรียกใช้ไลบารีได้เลย

รูปที่ 2.15 Bluetooth Dongle

2.6 วงจรลดแรงดันไฟฟ้ากระแสตรง (DC Step-Down)[5]

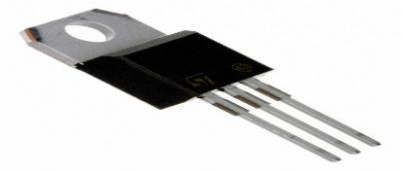
อุปกรณ์อิเล็กทรอนิกส์ยุกต์ดิจิตอลสมัยนี้ต่างนิยมใช้ไฟ 5V กันทั้งนั้นครับ หากเราต้องการต่อพ่วงกับแบตเตอรี่ หรือ อแดปเตอร์ ที่มีไฟสูงกว่าเช่น 12V เราควรจะต้องทำการปรับลดแรงดันไฟฟ้าให้เหมาะสมกับอุปกรณ์ เพื่อไม่ให้เกิดความเสียหายกับตัวอุปกรณ์ ซึ่งจะมีวิธีปรับลดแรงดันไฟฟ้าอยู่ 2 แบบ คือ

- **วิธีการแปลงโดยใช้ IC Linear Regulator** ซึ่งส่วนใหญ่ใช้ IC เบอร์ 7805 ซึ่งง่ายมากในการต่อใช้งาน โดยใช้ ไอซีเบอร์ 7805 พร้อมแผ่นระบายความร้อน 1 ตัวตัวเก็บประจุขนาด 220uF 16V 1 ตัว สายไฟสีดำแดง



รูปที่ 2.16 วิธีการแปลงโดยใช้ IC Linear Regulator

ข้อเสีย ของวงจรระบบ Linear คือหากอุปกรณ์ใช้กระแสสูงตัวไอซีจะร้อน (มาก) วงจรนี้ไฟเข้าไม่ได้จำกัดแค่ 12V แต่สามารถใช้ได้ตั้งแต่ 8 - 24 V โดยประมาณ แต่โวล์ยิ่งสูง ไอซียิ่งร้อน อาจะทำให้ไอซีไหม้ได้



รูปที่ 2.17 แสดงลักษณะของ IC Regulator

---

[5]วงจรลดแรงดันไฟฟ้ากระแสตรง [ออนไลน์], 28 พฤษภาคม 2558
แหล่งที่มา http://thaiconverter.com/category/2/dc-step-down

- **วิธีการแปลงแรงดันโดยใช้ระบบ Switching Regulator** ซึ่งใช้เทคนิคขั้นสูงที่จะทำให้เกิดความร้อนน้อยมาก ความสูญเสียน้อยต่างกับระบบ Linear ที่สูญเสียพลังงานที่ลดทอนแรงดันลงมาในรูปแบบของความร้อน แต่ระบบสวิตชิ่งต้องการอุปกรณ์หลายชิ้น ต้องอาศัยความรู้ทางด้านอิเล็กทรอนิกส์ในการออกแบบและคำนวณ หากออกแบบไม่ดีแล้ว อาจจะสร้างสัญญาณรบกวนอุปกรณ์ต่อพ่วงอื่นได้

รูปที่ 2.18 แสดงวงจร ระบบ Switching Regulator

---

[5]วงจรลดแรงดันไฟฟ้ากระแสตรง [ออนไลน์], 28 พฤษภาคม 2558.
แหล่งที่มา http://thaiconverter.com/category/2/dc-step-down

2.7 รีเลย์ (Relay)[6]

เป็นอุปกรณ์ที่เปลี่ยนพลังงานไฟฟ้าให้เป็นพลังงานแม่เหล็ก เพื่อใช้ในการดึงดูดหน้าสัมผัสของคอนแทคให้เปลี่ยนสภาวะ โดยการป้อนกระแสไฟฟ้าให้กับขดลวด เพื่อทำการปิดหรือเปิดหน้าสัมผัสคล้ายกับสวิตช์อิเล็กทรอนิกส์ ซึ่งเราสามารถนำรีเลย์ไปประยุกต์ใช้ ในการควบคุมวงจรต่าง ๆ ในงานช่างอิเล็กทรอนิกส์มากมาย

จุดต่อใช้งานมาตรฐาน ประกอบด้วย

- จุดต่อ **NC** ย่อมาจาก **normal close** หมายความว่าปกติเปิด หรือ หากยังไม่จ่ายไฟให้ขดลวดเหนี่ยวนำหน้าสัมผัสจะติดกัน โดยทั่วไปเรามักต่อจุดนี้เข้ากับอุปกรณ์หรือเครื่องใช้ไฟฟ้าที่ต้องการให้ทำงานตลอดเวลาเช่น
- จุดต่อ **NO** ย่อมาจาก **normal open** หมายความว่าปกติเปิด หรือหากยังไม่จ่ายไฟให้ขดลวดเหนี่ยวนำหน้าสัมผัสจะไม่ติดกัน โดยทั่วไปเรามักต่อจุดนี้เข้ากับอุปกรณ์หรือเครื่องใช้ไฟฟ้าที่ต้องการควบคุมการเปิดปิดเช่นโคมไฟสนามหนือหน้าบ้าน
- จุดต่อ **C** ย่อมากจาก **common** คือจุดร่วมที่ต่อมาจากแหล่งจ่ายไฟ

รูปที่ 2.19 แสดงลักษณะของรีเลย์

---

[6]รีเลย์ (Relay) [ออนไลน์], 28 พฤษภาคม 2558.
แหล่งที่มา http://www.psptech.co.th.รีเลย์relayคืออะไร-15696.page