

ควบคุมรีเลย์โดยใช้แอปพลิเคชันแอนดรอยด์

RELAY CONTROL BY ANDROID APPLICATION



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ.2557

ควบคุมรีเลย์โดยใช้แอปพลิเคชันแอนดรอยด์

Relay Control by android application



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ.2557

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2557

สาขาวิชา วิศวกรรมอิเล็กทรอนิกส์

คณะ วิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ควบคุมรีเลย์โดยใช้แอปพลิเคชันแอนดรอยด์
Relay Control by android application

ผู้จัดทำ นาย พลวิทย์ ศักดิ์ศรีธนาคุณ รหัสนักศึกษา 54011438

ปริญญานิพนธ์ผ่านการตรวจสอบโดยอาจารย์ที่ปรึกษาแล้ว



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญานิพนธ์ ควบคุมรีเลย์โดยใช้แอปพลิเคชันแอนดรอยด์
นักศึกษา นาย พลวิทย์ ศักดิ์ศรีธนาดุล รหัสนักศึกษา 54011438
ปริญญา วิศวกรรมศาสตรบัณฑิต
สาขาวิชา วิศวกรรมอิเล็กทรอนิกส์
ปีการศึกษา 2557
อาจารย์ที่ปรึกษา ผศ.ดร.แสงระวี บัวแก้ว

บทคัดย่อ

โครงการนี้นำเสนอการควบคุมรีเลย์โดยใช้แอปพลิเคชันบนสมาร์ตโฟน ในระบบปฏิบัติการแอนดรอยด์ ซึ่งจะแบ่งการทำงานออกเป็น3ส่วนหลักๆด้วยกันคือ ส่วนแอปพลิเคชัน ส่วนของฮาร์ดแวร์ และส่วนที่เชื่อมต่อระหว่างแอปพลิเคชันกับส่วนฮาร์ดแวร์คือคือบลูทูธโมดูล ซึ่งแอปพลิเคชันจะทำการเชื่อมต่อกับส่วนฮาร์ดแวร์โดยใช้สัญญาณบลูทูธ เมื่อมีการสั่งงานจากแอปพลิเคชันบลูทูธโมดูลจะส่งสัญญาณไปยังไมโครคอนโทรลเลอร์ และได้รับสัญญาณจากมัลติบลูทูธจะใช้ไมโครคอนโทรลเลอร์เพื่อใช้ในการควบคุมรีเลย์ต่อไป

Thesis Title Relay control by android application

Student Mr. Ponrawit Saksrithanadun Student ID 54011438

Degree Bachelor of Engineering

Program Electronics Engineering

Year 2014

Thesis Advisor Assist.professor Dr. Seangrawee Buakeaw

Abstract

This project presents the application of smartphone based on android's operating for controlling relay. The project consists of three major parts. The first part is application programs. The second parts is the electronic hardware. Which connection between a Bluetooth module. When the execution of applications with a Bluetooth module the data will be send to the microcontroller. After that the received data signals from Bluetooth will be sent to microcontroller for controlling the relay.

กิตติกรรมประกาศ

การทำโครงการ ควบคุมรีเลย์โดยใช้แอปพลิเคชันแอนดรอยด์นี้ ได้รับการสนับสนุนและช่วยเหลือให้คำแนะนำจากอาจารย์ภาควิชาอิเล็กทรอนิกส์ และได้รับความอนุเคราะห์ในด้านอุปกรณ์และเครื่องมือจากภาควิชาอิเล็กทรอนิกส์ โดยเฉพาะอย่างยิ่ง ผศ.ดร.แสงระวี บัวแก้ว ที่ให้แนวคิดและคำปรึกษา การแก้ไขปัญหาและการสนับสนุนในการทำวิจัย รวมไปถึงบิดามารดาของผู้ทำโครงการ ที่ให้การสนับสนุนด้านทุนทรัพย์เพื่อทำการจัดหาอุปกรณ์บางส่วน ซึ่งทางคณะผู้จัดทำรู้สึกซาบซึ้งในความกรุณาของทุกท่านอย่างที่สุดและขอกราบขอบพระคุณเป็นอย่างสูง

พลวิทย์ ศักดิ์ศรีธนาคุณ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

| หัวข้อ | หน้า |
|---|------|
| บทคัดย่อ..... | I |
| Abstract..... | II |
| กิตติกรรมประกาศ..... | III |
| สารบัญรูป..... | VI |
| บทที่ 1 บทนำ..... | 1 |
| 1.1 ความเป็นมาของโครงการ..... | 1 |
| 1.2 วัตถุประสงค์ของโครงการ..... | 1 |
| 1.3 ขอบเขตในโครงการ..... | 2 |
| 1.4 ประโยชน์ที่ได้รับ..... | 2 |
| 1.5 รายละเอียดโดยย่อ..... | 2 |
| บทที่ 2 ทฤษฎีและหลักการ..... | 3 |
| 2.1 ระบบปฏิบัติการแอนดรอยด์..... | 3 |
| 2.2 ทฤษฎีไมโครคอนโทรลเลอร์ตระกูล arduino..... | 5 |
| 2.3 สัญญาณบลูทูท..... | 13 |
| 2.4 Module hc05..... | 16 |
| 2.5 รีเลย์..... | 17 |
| 2.6 วงจรขับรีเลย์ด้วยทรานซิสเตอร์..... | 20 |
| 2.6.1 เเน่ด้านควบคุม..... | 20 |
| บทที่3 การออกแบบ..... | 24 |
| 3.1 บอร์ดไมโครคอนโทรลเลอร์..... | 24 |
| 3.2 แอปพลิเคชันที่ใช้ในการควบคุม..... | 24 |
| 3.3 HC-05 บลูทูธโมดูล..... | 25 |
| 3.4 วงจรที่ใช้ในการขับรีเลย์..... | 26 |
| 3.5 Flow Chart ระบบการทำงาน..... | 28 |
| บทที่4 ผลการทดลอง..... | 29 |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

| | |
|--|----|
| 4.1 Output ที่ได้จากไมโครคอนโทรลเลอร์..... | 29 |
| 4.2 Outputที่ได้จากการวัดค่า NC เทียบกับค่า COM ของรีเลย์..... | 32 |
| บทที่ 5 สรุปผลการทดลอง..... | 33 |
| 5.1 สรุปผลการทดลอง..... | 33 |
| 5.2 ปัญหาและแนวทางการแก้ไข..... | 33 |
| 5.3 แนวทางการพัฒนา..... | 33 |
| บรรณานุกรม..... | 34 |
| ภาคผนวก..... | 35 |
| Code ที่ใช้ในการควบคุมการทำงาน..... | 36 |
| Hardware ที่ได้ทำการออกแบบ..... | 44 |



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 [M]
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

| รูปที่ | หน้า |
|--|------|
| รูปที่ 2. 1 ระบบปฏิบัติการแอนดรอยด์ | 4 |
| รูปที่ 2. 2 บอร์ด Arduino | 7 |
| รูปที่ 2. 3 Arduino Uno R3 | 8 |
| รูปที่ 2. 4 Arduino Uno SMD | 8 |
| รูปที่ 2. 5 Arduino Mega 2560 R3 | 9 |
| รูปที่ 2. 6 Arduino Mega ADK | 9 |
| รูปที่ 2. 7 Arduino Leonardo | 9 |
| รูปที่ 2. 8 Arduino Mini 05..... | 10 |
| รูปที่ 2. 9 Arduino Pro Mini 328 3.3V..... | 10 |
| รูปที่ 2. 10 Arduino Pro Mini 328 5V..... | 10 |
| รูปที่ 2. 11 Arduino Ethernet with PoE module..... | 11 |
| รูปที่ 2. 12 Arduino Ethernet without PoE module..... | 11 |
| รูปที่ 2. 13 Arduino Due | 11 |
| รูปที่ 2. 14 การทำงานของระบบบลูทูธ..... | 14 |
| รูปที่ 2. 15 รูปแบบของบลูทูธแพ็คเก็จ..... | 15 |
| รูปที่ 2. 16 บลูทูธโปรโตคอล..... | 15 |
| รูปที่ 2. 17 module Hc-05..... | 16 |
| รูปที่ 2. 18 รีเลย์กำลัง..... | 17 |
| รูปที่ 2. 19 รีเลย์ควบคุม..... | 18 |
| รูปที่ 2. 20 สัญลักษณ์ของรีเลย์..... | 19 |
| รูปที่ 2. 21 สถานะขดลวดรีเลย์ขณะไม่มีกระแสไฟฟ้า..... | 20 |
| รูปที่ 2. 22 สถานะขดลวดรีเลย์เมื่อมีกระแสไฟฟ้า..... | 20 |
| รูปที่ 2. 23 โครงสร้างการทำงานของทรานซิสเตอร์ NPN..... | 21 |
| รูปที่ 2. 24 โครงสร้างการทำงานของทรานซิสเตอร์ PNP..... | 21 |
| รูปที่ 2. 25 วงจรขับรีเลย์ทรานซิสเตอร์ที่นิยมใช้..... | 22 |
| รูปที่ 3. 1 วงจรสมมูลของบอร์ดไมโครคอนโทรลเลอร์..... | 24 |
| รูปที่ 3. 2 แอปพลิเคชันที่ใช้ในการควบคุมรีเลย์..... | 24 |
| รูปที่ 3. 3 การเชื่อมต่อระหว่าง HC05 module กับบอร์ดไมโครคอนโทรลเลอร์..... | 25 |
| รูปที่ 3. 4 วงจรแปลงไฟฟ้ากระแสสลับเป็นไฟฟ้ากระแสตรง..... | 26 |
| รูปที่ 3. 5 วงจรสมมูลที่ใช้ในการขับรีเลย์..... | 26 |

| | | |
|-------------|---|----|
| รูปที่ 4. 1 | เมื่อยังไม่มีการสั่งเปิดรีเลย์..... | 29 |
| รูปที่ 4. 2 | เมื่อสั่งเปิดรีเลย์ duty cycle เท่ากับ 25% | 30 |
| รูปที่ 4. 3 | เมื่อสั่งเปิดรีเลย์ duty cycle เท่ากับ 50% | 30 |
| รูปที่ 4. 4 | เมื่อสั่งเปิดรีเลย์ duty cycle เท่ากับ 75% | 31 |
| รูปที่ 4. 5 | เมื่อสั่งเปิดรีเลย์ duty cycle เท่ากับ 100% | 31 |
| รูปที่ 4. 6 | ขณะที่ยังไม่สั่งให้รีเลย์ทำงาน | 32 |
| รูปที่ 4. 7 | ขณะที่ยังสั่งให้รีเลย์ทำงาน | 32 |



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 [vii]
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

1.1 ความเป็นมาของโครงงาน

ความเป็นมาและความสำคัญ

ปัจจุบันเทคโนโลยีเกี่ยวกับสมาร์ทโฟนแทบจะไม่มีใครไม่รู้จักคำว่า “แอนดรอยด์”(Android)ซึ่งในตลาดสมาร์ทโฟนในปัจจุบันนั้น แอนดรอยด์ ถือเป็นอีกหนึ่งระบบปฏิบัติการที่ได้รับความนิยมเป็นอย่างมาก ไม่ว่าจะเป็นทางด้านของผู้ผลิตหรือแม้กระทั่งผู้บริโภค ล้วนแล้วก็อยากจะเรียนรู้เพื่อนำมาประยุกต์ใช้งาน เนื่องจากเป็นระบบปฏิบัติการที่แจกให้ใช้กับแบบฟรีๆ (Open Source) ซึ่งสิ่งที่เราสนใจเป็นอย่างมากคือ วิธีการสร้างแอปพลิเคชัน(Application)ที่นำมาใช้ควบคุมอุปกรณ์ภายนอกในรูปแบบของนักอิเล็กทรอนิกส์

สำหรับในโครงงานนี้จะกำหนดให้มีการเชื่อมต่อแบบอัตโนมัติ ซึ่งจะต้องใช้อุปกรณ์ที่จะนำมาใช้งานมาเชื่อมต่อเข้ากับ HC-05 ซึ่งเป็นโมดูลที่เราต้องทำการเชื่อมต่อ หัวใจหลักของโครงงานนี้คือ ไมโครคอนโทรลเลอร์ arduino ซึ่งเป็นไมโครคอนโทรลเลอร์ของตระกูล AVR แบบ Open Source โดยใช้ arduino UNO r3 โดยหน้าที่หลักของ arduino คือจะรับคำสั่งจาก HC-05 ซึ่งเป็นโมดูลบลูทูธ(Bluetooth Module)แบบไร้สาย ที่ใช้การสื่อสารกันด้วย Bluetooth SSP (Serial Port Protocol) โดย Serial Port เป็น Bluetooth V2.0+EDR (Enhance Data Rate) 3Mbps Modulation พร้อมกับความถี่ที่ใช้งาน 2.4GHz ซึ่งอาจจะสามารถนำมาเชื่อมต่อกับอุปกรณ์บลูทูธอื่นๆได้ รวมทั้งเชื่อมต่อไมโครคอนโทรลเลอร์ดังเช่นเดียวกับในโครงงานนี้

1.2 วัตถุประสงค์ของโครงงาน

โครงงานชิ้นนี้มีวัตถุประสงค์ดังต่อไปนี้

1. ศึกษาเกี่ยวกับอุปกรณ์ไมโครคอนโทรลเลอร์เพื่อให้สามารถนำไปใช้ประยุกต์ใช้งานต่างๆได้
2. ศึกษาคำสั่งที่ใช้ในการควบคุมไมโครคอนโทรลเลอร์ เขียนคำสั่ง และสร้างวงจรที่ทำให้วงจรไมโครคอนโทรลเลอร์สามารถทำตามวัตถุประสงค์ที่ต้องการได้
3. ศึกษาระบบปฏิบัติการแอนดรอยด์เพื่อที่จะได้นำมาพัฒนาต่อไปในอนาคต
4. ศึกษาและพัฒนาการสร้างแอปพลิเคชันในระบบปฏิบัติการแอนดรอยด์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. สามารถนำมาประยุกต์ใช้งานในชีวิตประจำวันได้ เช่น ในอนาคตอาจจะนำไปควบคุมอุปกรณ์ภายนอกประเภทอื่นๆต่อไป
6. การออกแบบวงจรที่จะใช้งานตามวัตถุประสงค์ที่ต้องการ

1.3 ขอบเขตในโครงการ

ศึกษาทำความเข้าใจเกี่ยวกับระบบปฏิบัติการแอนดรอยด์ การเขียนแอปพลิเคชันเพื่อใช้ในสมาร์ตหรือแท็บเล็ตที่ใช้ระบบปฏิบัติการแอนดรอยด์เพื่อนำไปควบคุมในการขับรีเลย์

1.4 ประโยชน์ที่ได้รับ

เป็นแนวทางในการพัฒนาการเขียนแอปพลิเคชันในระบบปฏิบัติการแอนดรอยด์ การประยุกต์ใช้คำสั่งต่างๆของไมโครคอนโทรลเลอร์

1.5 รายละเอียดโดยย่อ

บทที่ 1 บทนำ เป็นการกล่าวถึงรายละเอียดความเป็นมาและวัตถุประสงค์ของโครงการ เพื่อให้เห็นภาพรวมก่อนที่จะเข้าสู่บทฤษฎีต่อไป

บทที่ 2 ทฤษฎีและหลักการ กล่าวถึงรายละเอียด ทฤษฎีเบื้องต้นของอุปกรณ์ที่สำคัญ และรายละเอียดต่างๆของวงจร

บทที่ 3 วิธีการและขั้นตอนการดำเนินงานกล่าวถึงรายละเอียดของอุปกรณ์ในส่วนต่างๆและวงจรควบคุม

บทที่ 4 การทดลองและผลการทดลอง จะกล่าวถึงวิธีการทดสอบ และผลการทดสอบหลังการนำอุปกรณ์และวงจรควบคุมที่ได้สร้างจากบทที่ 3 แล้วทำการวิเคราะห์ผลที่เกิดขึ้น

บทที่ 5 สรุปผลการทดลอง จะสรุปผลลัพธ์ที่ได้จากการปฏิบัติโครงการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ทฤษฎีและหลักการ

การควบคุมรีเลย์โดยใช้แอปพลิเคชันบนสมาร์ตโฟน ในระบบปฏิบัติการแอนดรอยด์ ซึ่งจะแบ่งการทำงานออกเป็น 3 ส่วนหลักๆ ด้วยกันคือ ส่วนแอปพลิเคชัน ส่วนของฮาร์ดแวร์ และส่วนที่เชื่อมต่อระหว่างแอปพลิเคชันกับส่วนฮาร์ดแวร์คือคือบลูทูธโมดูล ซึ่งทฤษฎีและหลักการแบ่งออกเป็น 6 หัวข้อ ดังนี้

- 2.1 ระบบปฏิบัติการแอนดรอยด์
- 2.2 ทฤษฎีไมโครคอนโทรลเลอร์ตระกูล arduino
- 2.3 สัญญาณบลูทูธ
- 2.4 Module hc05
- 2.5 รีเลย์
- 2.6 วงจรขับรีเลย์

2.1 ระบบปฏิบัติการแอนดรอยด์

แอนดรอยด์ (Android) กูเกิลแอนดรอยด์ (Google Android) หรือ ระบบปฏิบัติการแอนดรอยด์ (Android Operating System) เป็นชื่อเรียกชุดซอฟต์แวร์ หรือแพลตฟอร์ม (Platform) สำหรับอุปกรณ์อิเล็กทรอนิกส์ ที่มีหน่วยประมวลผลเป็นส่วนประกอบ อาทิเช่น คอมพิวเตอร์, โทรศัพท์ (Telephone), โทรศัพท์เคลื่อนที่ (Cell phone), อุปกรณ์เล่นอินเทอร์เน็ตขนาดพกพา (MID) เป็นต้น แอนดรอยด์นั้น ถือกำเนิดอย่างเป็นทางการในวันที่ 5 พฤศจิกายน 2550 โดยบริษัท กูเกิล จุดประสงค์ของแอนดรอยด์นั้น มีจุดเริ่มต้นมาจากบริษัท Android Inc. ที่ได้นำเอาระบบปฏิบัติการลินุกซ์ (Linux) ซึ่งนิยมนำไปใช้งานกับเครื่องแม่ข่าย (Server) เป็นหลัก นำมาลดทอนขนาดตัว (แต่ไม่ลดทอนความสามารถ) เพื่อให้เหมาะสมแก่การนำไปติดตั้งบนอุปกรณ์พกพาที่มีขนาดพื้นที่จัดเก็บข้อมูลที่จำกัด โดยหวังว่า แอนดรอยด์ นั้นจะเป็นหุ่นยนต์ตัวน้อย ๆ ที่คอยช่วยเหลืออำนวยความสะดวกแก่ผู้ที่พกพามันไปในทุกที่ทุกเวลา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2. 1 ระบบปฏิบัติการแอนดรอยด์

ที่มา : www.techmoblog.com

กูเกิลแอนดรอยด์ เป็นชื่อเรียกอย่างเป็นทางการของเจ้าแอนดรอยด์ เนื่องจากปัจจุบันนี้ บริษัทกูเกิล เป็นผู้ถือสิทธิบัตรในตราสัญลักษณ์ ชื่อ และ รหัสต้นฉบับ (Source Code) ของแอนดรอยด์ ภายใต้เงื่อนไขการพัฒนาแบบ GNL โดยเปิดให้นักพัฒนา (Developer) สามารถนำรหัสต้นฉบับ ไปพัฒนาปรับแต่งได้อย่างเปิดเผย (Open source) ทำให้แอนดรอยด์มีผู้เข้าร่วมพัฒนาเป็นจำนวนมาก และพัฒนาไปได้อย่างรวดเร็ว แอนดรอยด์เปิดตัวอย่างเป็นทางการเมื่อวันที่ 5 พฤษภาคม พุทธศักราช 2550 ปัจจุบันมีผู้ร่วมพัฒนากว่า 52 องค์กร ประกอบด้วยบริษัทซอฟต์แวร์ บริษัทผู้ผลิตอุปกรณ์ บริษัทผู้ผลิตชิ้นส่วนอิเล็กทรอนิกส์ บริษัทผู้ให้บริการเครือข่าย และบริษัทที่เกี่ยวข้องกับการสื่อสาร ฯลฯ

ประเภทของชุดซอฟต์แวร์ เนื่องจากแอนดรอยด์นั้นเปิดให้นักพัฒนาเข้าไปชมรหัสต้นฉบับได้ ทำให้มีผู้พัฒนาจากหลายฝ่ายนำเอารหัสต้นฉบับมาปรับแต่ง และสร้างแอนดรอยด์ในแบบฉบับของตนเองขึ้น เราจึงแบ่งประเภทของแอนดรอยด์ออกได้เป็น 3 ประเภท ดังต่อไปนี้

1. Android Open Source Project (AOSP) เป็นแอนดรอยด์ประเภทแรกที่กูเกิลเปิดให้สามารถนำ “ต้นฉบับแบบเปิด” ไปติดตั้งและใช้งานในอุปกรณ์ต่าง ๆ ได้โดยไม่ต้องเสียค่าใช้จ่ายใด ๆ

2. Open Handset Mobile (OHM) เป็นแอนดรอยด์ที่ได้รับการพัฒนาร่วมกับกลุ่มบริษัทผู้ผลิตอุปกรณ์พกพา ที่เข้าร่วมกับกูเกิลในนาม Open Handset Alliances (OHA) ซึ่งบริษัทเหล่านี้จะพัฒนาแอนดรอยด์ในแบบฉบับของตนเองออกมา โดยรูปร่างหน้าตาการแสดงผล และฟังก์ชันการใช้งาน จะมีความเป็นเอกลักษณ์ และมีลิขสิทธิ์เป็นของตนเอง พร้อมได้รับสิทธิในการมีบริการเสริมต่าง ๆ จากกูเกิล ที่เรียกว่า Google Mobile Service (GMS) ซึ่งเป็นบริการเสริมที่ทำให้แอนดรอยด์มีประสิทธิภาพ เป็นไปตามจุดประสงค์ของแอนดรอยด์ แต่การจะได้มาซึ่ง GMS นั้น ผู้ผลิตจะต้องทำการทดสอบระบบ และขออนุญาตกับทางกูเกิลก่อน จึงจะนำเครื่องออกสู่ตลาดได้

3. Cooking หรือ Customize เป็นแอนดรอยด์ที่นักพัฒนานำเอารหัสต้นฉบับจากแหล่งต่าง ๆ มาปรับแต่ง ในแบบฉบับของตนเอง โดยจะต้องทำการปลดล๊อคสิทธิ์การใช้งานอุปกรณ์ หรือ Unlock เครื่องก่อน จึงจะสามารถติดตั้งได้ โดยแอนดรอยด์ประเภทนี้ถือเป็นประเภทที่มีความสามารถมาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ที่สุด เท่าที่อุปกรณ์เครื่องนั้น ๆ จะรองรับได้ เนื่องจากได้รับการปรับแต่งให้เข้ากับอุปกรณ์นั้น ๆ จากผู้ใช้งานจริง

สิทธิ์ในการใช้งานระบบ เช่นเดียวกับระบบปฏิบัติการทั่วไป ที่มีการจำกัดการใช้งาน และการเข้าถึงส่วนต่าง ๆ ภายในระบบ เพื่อความปลอดภัยของระบบ และ ผู้ใช้งาน อุปกรณ์ที่ติดตั้งระบบแอนดรอยด์จึงมีการจำกัดสิทธิ์ไว้ (เว้นแต่ได้ทำการปลดล๊อคสิทธิ์ หรือ root เครื่องแล้ว) สามารถแบ่งสิทธิ์ของผู้ใช้ในการเข้าถึงระบบคร่าว ๆ ได้ดังต่อไปนี้

1. สิทธิ์ root สิทธิ์การใช้ใช้งานระดับราก ซึ่งถือว่าเป็นรากฐานของระบบ จึงมีความสามารถในการเข้าถึงทุก ๆ ส่วนของระบบ

2. สิทธิ์ ADB (Android Develop Bridge) นักพัฒนาสามารถเข้าถึงส่วนต่าง ๆ ของระบบได้ผ่านสิทธิ์นี้

3. Application & System สิทธิ์ของโปรแกรมในการเข้าถึงระบบ และสิทธิ์ของระบบในการเข้าถึงอุปกรณ์ โดยสิทธิ์เหล่านี้ ตัวระบบจะเป็นตัวจัดการมอบและถอนสิทธิ์ ตามเงื่อนไขที่กำหนด ซึ่งจะถูกแบ่งย่อยออกเป็นหลายหัวข้อ

4. End-user ผู้ใช้งานขั้นสุดท้าย ซึ่งก็คือ คุณ และ คุณ ทั้งหลาย ที่ใช้การเข้าถึงส่วนต่าง ๆ ของระบบผ่านช่องทางสิทธิ์ที่โปรแกรมได้รับอีกที โดยจะถูกจำกัดไม่ให้เข้าถึงในส่วนที่เป็นอันตรายต่อแกนระบบและอุปกรณ์

2.2 ทฤษฎีไมโครคอนโทรลเลอร์ตระกูล arduino

ไมโครคอนโทรลเลอร์ (Microcontroller) เป็นอุปกรณ์อิเล็กทรอนิกส์แบบหนึ่ง ที่รวมเอาหน่วยประมวลผล หน่วยคำนวณทางคณิตศาสตร์และลอจิก วงจรรับสัญญาณอินพุต วงจรส่งสัญญาณเอาต์พุต รวมถึงหน่วยความจำวงจรกำเนิดสัญญาณนาฬิกาไว้ด้วยกัน ทำให้สามารถนำไปใช้งานแทนวงจรอิเล็กทรอนิกส์ที่ซับซ้อนได้เป็นอย่างดี โดยไมโครคอนโทรลเลอร์มาจากคำสองคำรวมกันคือ “ไมโคร” ซึ่งหมายถึงไมโครโปรเซสเซอร์ เป็นอุปกรณ์ประมวลผลข้อมูลขนาดเล็กภายในประกอบด้วยหน่วยประมวลผลกลางหรือ CPU ประกอบด้วยหน่วยคำนวณทางคณิตศาสตร์และลอจิก วงจรเชื่อมต่อหน่วยความจำวงจรกำเนิดสัญญาณนาฬิกา อีกคำหนึ่งคือคำว่า “คอนโทรลเลอร์” หมายถึง อุปกรณ์ควบคุม ดังนั้นไมโครคอนโทรลเลอร์จึงเป็นอุปกรณ์ที่ใช้ในการควบคุม โดยที่สามารถเขียนโปรแกรมเพื่อกำหนดรูปแบบการควบคุมได้อย่างเป็นอิสระ

เอกชัย มะการ (2552) ได้ให้ความเห็นสำหรับ Arduino ไว้ว่า Arduino เป็นภาษา อิตาลี ซึ่งใช้เป็นชื่อของโครงการพัฒนาไมโครคอนโทรลเลอร์ตระกูล AVR แบบ Open Source ที่ได้รับการปรับปรุงมาจากโครงการพัฒนา Open Source ของ AVR อีกโครงการหนึ่งที่มีชื่อว่า “Wiring” แต่เนื่องจากโครงการของ “Wiring” แต่เนื่องจากโครงการของ “Wiring” เลือกใช้ AVR เบอร์ ATmega128 ซึ่งเป็นไมโครคอนโทรลเลอร์ ที่มีจำนวนของหน่วยความจำ และ I/O ค่อนข้างมากมีชิพตัวถังแบบ SMD และบอร์ดมีขนาดใหญ่จึงไม่ได้รับความนิยมเท่าที่ควร แต่หลังจากทางทีมงาน

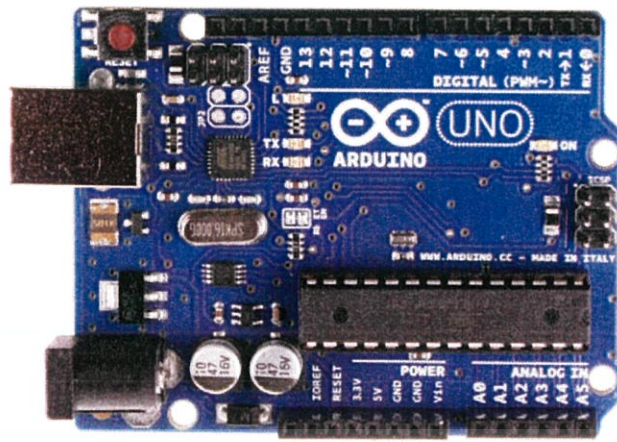
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Arduino นำ Source Code ของ “Wiring” มาพัฒนาปรับปรุงใหม่โดยใช้งานกับไมโครคอนโทรลเลอร์ขนาดเล็กกว่า “Wiring” มากและยังใช้อุปกรณ์น้อยชิ้นทำให้ง่ายต่อการต่อวงจรใช้งานกันนี้เองและยังประหยัดต้นทุนในการสร้างบอร์ดไปได้มาก ทำให้ “Arduino” ได้รับความนิยมจากผู้ใช้งานจากทั่วโลก ในระยะเวลาอย่างรวดเร็ว Arduino มีจุดเด่นในเรื่องของความง่ายในการเรียนรู้และใช้งาน เนื่องจากมีการออกแบบคำสั่งต่าง ๆ ขึ้นมาสนับสนุนการใช้งาน ด้วยรูปแบบที่ง่ายไม่ซับซ้อน ซึ่งถึงแม้ว่า Arduino เองจะมีรูปแบบการใช้งานคล้ายๆ กันกับไมโครคอนโทรลเลอร์อย่าง Basic Stamp ของ Parallax, BX-24 ของ Net medias และ Handy Board ของ MIT แต่ก็มีความโดดเด่นกว่าของรายอื่น ๆ หลายอย่างเป็นต้นว่า

- ราคาไม่แพง เนื่องจากมี Source Code และวงจร แจกให้ฟรี สามารถต่อวงจรขึ้นมาใช้งานได้เอง
- โปรแกรมที่ใช้พัฒนาของ Arduino รองรับการทำงานทั้ง Windows, Linux และ Macintosh OSX
- รูปแบบคำสั่งที่ง่ายต่อการใช้งาน แต่สามารถนำไปใช้งานจริง ๆ ที่มีความซับซ้อนมาก ๆ ได้ และยังสามารสร้างคำสั่งและ Library ใหม่ ๆ ขึ้นมาใช้งานได้เองได้ เมื่อมีความชำนาญมากขึ้นแล้ว
- มีการเปิดเผยวงจรและ Source Code ทั้งหมดทำให้สามารถนำไปพัฒนาต่อยอดเพิ่มเติมได้ตามความต้องการทั้ง Hardware และ Software

Arduino เป็นบอร์ดไมโครคอนโทรลเลอร์ โดยใช้ AVR ขนาดเล็กเป็นตัวประมวลผลและสั่งงานเหมาะสำหรับนำไปใช้ในการศึกษาเรียนรู้ระบบไมโครคอนโทรลเลอร์ และนำไปประยุกต์ใช้งานเกี่ยวกับการควบคุมอุปกรณ์ Input/Output ต่าง ๆ ได้มากมาย ทั้งในรูปแบบที่เป็นการทำงานตัวเดียวอิสระ หรือ เชื่อมต่อสั่งงานร่วมกับอุปกรณ์อื่น ๆ เช่น คอมพิวเตอร์ PC ทั้งนี้ก็เนื่องมาจากว่า Arduino สนับสนุนการเชื่อมต่อกับอุปกรณ์ Input/Output ต่างๆ ได้มากมาย ทั้งแบบ Digital และ Analog เช่น การรับค่าจากสวิทช์ หรือ อุปกรณ์ตรวจจับ (Sensor) แบบต่าง ๆ รวมไปถึงการควบคุมอุปกรณ์ Output ต่าง ๆ ตั้งแต่ LED, หลอดไฟ, มอเตอร์, รีเลย์ ฯลฯ

Arduino เป็นระบบการพัฒนาไมโครคอนโทรลเลอร์แบบ Open Source ด้วยเหตุนี้จึงมีผู้คนทั่วไปให้ความสนใจและนำไปศึกษาทดลองใช้งานกันมากมาย มีการนำไปดัดแปลงและสร้างเป็นโครงการ แบบต่าง ๆ กันเป็นจำนวนมาก



รูปที่ 2. 2 บอร์ด Arduino ที่มา : elec-robo.webiz.co.th

โปรแกรมภาษาของ Arduino จะใช้ภาษา C++ ซึ่งเป็นรูปแบบของโปรแกรมภาษาซีประยุกต์แบบหนึ่ง ที่มีโครงสร้างตัวภาษาโดยรวมใกล้เคียงกันกับ ภาษาซีมาตรฐาน (ANSI-C) อื่น ๆ เพียงแต่ได้มีการปรับปรุงรูปแบบในการเขียนโปรแกรมบางส่วนที่ผิดเพี้ยนไปจาก ANSI-C เล็กน้อย เพื่อช่วยลดความยุ่งยากในการเขียนโปรแกรมและให้ ผู้เขียนโปรแกรมสามารถได้ง่ายและสะดวกมากขึ้นกว่าการเขียนภาษาซีตามแบบมาตรฐานของ ANSI-C โดยตรงภาษาซี สามารถนำไปใช้งานบนระบบฮาร์ดแวร์ที่มีความแตกต่างกันได้หลากหลาย โดยผู้ใช้เพียงแต่เลือกใช้ ตัวแปลคำสั่ง (C-Compiler) ให้ตรงกับระบบฮาร์ดแวร์ที่ใช้งานอยู่ ส่วนเรื่องรูปแบบในการเขียนโปรแกรมจะเป็นมาตรฐานอันเดียวกัน American National Standard Institute (ANSI) จึงได้ทำการตั้งข้อกำหนดมาตรฐานของภาษาซีขึ้น โดยเรียกกันว่า “ANSI-C” เพื่อใช้เป็นข้อบังคับและคงมาตรฐานของภาษาซีไว้ไม่ให้เปลี่ยนแปลงไปจากเดิม โดยทุกผู้ผลิตจะต้องสร้างตัวแปลภาษาให้มีคุณสมบัติการใช้งานขั้นพื้นฐานที่เป็นสิ่งเดียวกัน คำสั่งที่เป็นภาษาซีตามมาตรฐานของ ANSI-C จะมี 32 คำสั่ง คือ

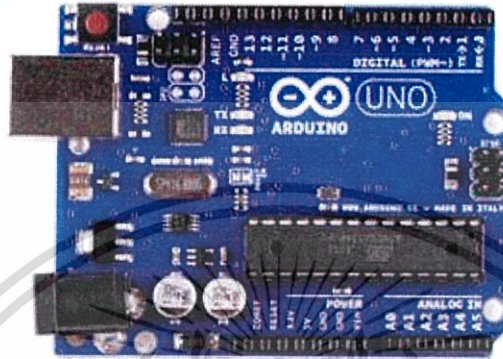
auto break case char const continue default do double else enum extern float
for goto if int long register return short signed sizeof static struct switch type typedef union
unsigned void volatile while

คำสั่งทั้ง 32 คำสั่งนี้ ไม่ว่าจะอยู่บน C-Compiler ตัวใดก็จะต้องมีข้อกำหนดและรูปแบบการใช้งานของคำสั่งที่เหมือนกันทั้งหมด ยกเว้นว่าตัวแปลคำสั่งตัวนั้น ไม่รองรับคำสั่งของ ANSI-C ทั้งหมด Arduino ไม่ใช่ ตัวแปลภาษาโดยตรง แต่ Arduino จะมีลักษณะการทำงานเช่นเดียวกับ Text Editor ของภาษาซี ตัวหนึ่ง โดยจะทำงานร่วมกับ Utility บางส่วนที่ Arduino สร้างขึ้นมารองรับโดย Arduino จะใช้รูปแบบการทำงานของ Text Editor เป็นฉากหน้าในการติดต่อสื่อสารกับผู้ใช้เท่านั้น ส่วนเบื้องหลังจริงๆ นั้น Arduino จะไปเรียกใช้ตัวแปลภาษาซี และ Utility อื่น ๆ ที่ใช้เป็นเครื่องมือพัฒนาโปรแกรมของ ไมโครคอนโทรลเลอร์ ตระกูล AVR อีกทีหนึ่ง โดย Arduino จะเลือกใช้ตัวแปลภาษาของ “GNU AVR-GCC Toolchain” ร่วมกับ Library Function ของ “avr-libc” ส่วน Utility ที่ใช้ในการ Upload Code ให้กับ AVR นั้นก็จะใช้ของ “AVRDude” ดังนั้นผู้ที่เขียนภาษาซีของ AVR เป็นอยู่แล้ว และต้องการประยุกต์ใช้งาน Arduino ให้ได้ประสิทธิภาพการทำงานมากยิ่งขึ้นไปอีก ก็สามารถศึกษาข้อกำหนด และหน้าที่ในการใช้งาน Library และคำสั่งอื่น ๆ ที่บรรจุไว้ใน Library ต่าง ๆ ทั้งจากของ “GNU AVR-GCC Toolchain” และ “avr-libc” เพิ่มเติมอีกเพื่อใช้เป็นแนวทางในการปรับปรุงและประยุกต์ใช้งาน Arduino ในรูปแบบที่สลับซับซ้อนมากขึ้นไปได้อีก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.1 รุ่นของ Arduino

1. Arduino Uno R3 เป็นบอร์ด Arduino ที่ได้รับความนิยมมากที่สุด เนื่องจากราคาไม่แพง ส่วนใหญ่โปรเจกต์และ Library ต่างๆ ที่พัฒนาขึ้นมา Support จะอ้างอิงกับบอร์ดนี้เป็นหลัก และข้อดีอีกอย่างคือ กรณีที่ MCU เสีย ผู้ใช้งานสามารถซื้อมาเปลี่ยนเองได้ง่าย



รูปที่ 2. 3 Arduino Uno R3

2. Arduino Uno SMD เป็นบอร์ดที่มีคุณสมบัติและการทำงานเหมือนกับบอร์ด Arduino UNO R3 ทุกประการ แต่จะแตกต่างที่ Package ของ MCU ซึ่งบอร์ดนี้จะมี MCU ที่เป็น Package SMD (Arduino UNO R3 มี MCU ที่เป็น Package DIP)



รูปที่ 2. 4 Arduino Uno SMD

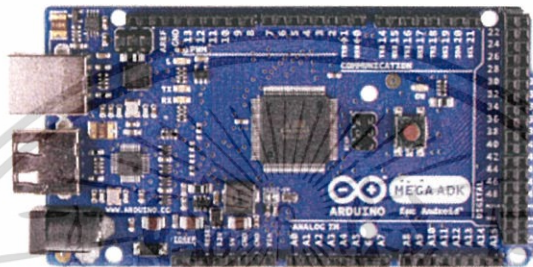
3. Arduino Mega 2560 R3 เป็นบอร์ด Arduino ที่ออกแบบมาสำหรับงานที่ต้องใช้ I/O มากกว่า Arduino Uno R3 เช่น งานที่ต้องการรับสัญญาณจาก Sensor หรือควบคุมมอเตอร์ Servo หลายๆ ตัว ทำให้ Pin I/O ของบอร์ด Arduino Uno R3 ไม่สามารถรองรับได้ ทั้งนี้บอร์ด Mega 2560 R3 ยังมีความหน่วยความจำแบบ Flash มากกว่า Arduino Uno R3 ทำให้สามารถเขียนโค้ดโปรแกรมเข้าไปได้มากกว่า ในความเร็วของ MCU ที่เท่ากัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2. 5 Arduino Mega 2560 R3

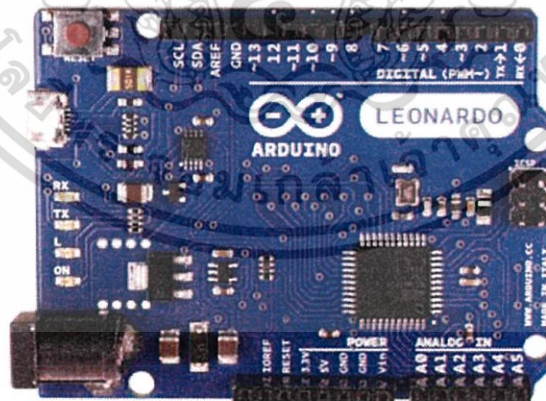
4. Arduino Mega ADK เบนบอร์ดทอออกแบบมาเหบอร์ด Mega 2560 R3 สามารถติดต่อกับอุปกรณ์ Android Device ผ่านพอร์ต USB Host ของบอร์ดได้



รูปที่ 2. 6 Arduino Mega ADK

5. Arduino Leonardo การทำงานจะคล้ายกับบอร์ด Arduino Uno R3 แต่มีการเปลี่ยน MCU ตัวใหม่เป็น ATmega32U4 ซึ่งมีโมดูลพอร์ต USB มาด้วยบนชิป (แตกต่างจากบอร์ด Arduino UNO R3 หรือ Arduino Mega 2560 ที่ต้องใช้ชิป ATmega16U2 ร่วมกับ ATmega328 ในการเชื่อมต่อกับพอร์ต USB)

ข้อควรระวัง: เนื่องจาก MCU เป็นคนละเบอร์กับ Arduino Uno R3 อาจทำให้บอร์ด Shield บางตัวหรือ Library ใช้ร่วมกันกับบอร์ด Arduino Leonardo ไม่ได้ ผู้ใช้งานจำเป็นต้องตรวจสอบก่อนใช้งาน



รูปที่ 2. 7 Arduino Leonardo

6. Arduino Mini 05 เป็นบอร์ด Arduino ขนาดเล็กที่ใช้ MCU เบอร์ ATmega328 เบอร์เดียวกับบอร์ด Arduino UNO R3

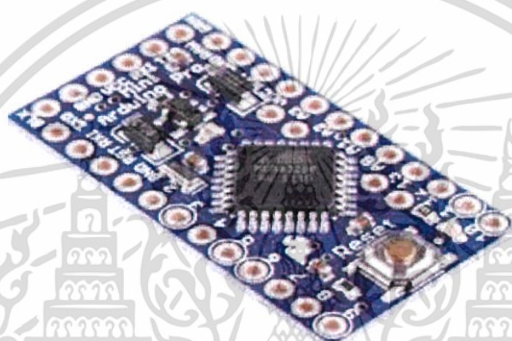
ข้อแตกต่าง: บอร์ด Arduino Mini 05 จะไม่มีพอร์ต USB มาให้ ผู้ใช้งานต้องต่อกับบอร์ด USB to Serial Converter เพิ่มเมื่อต้องการโปรแกรมบอร์ด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



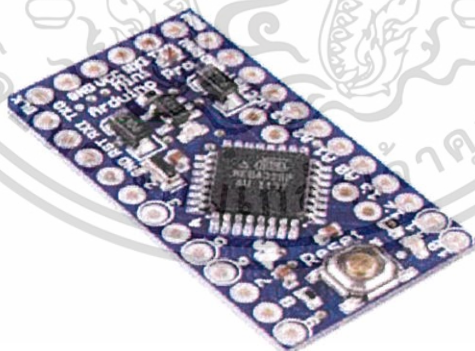
รูปที่ 2. 8 Arduino Mini 05

7. Arduino Pro Mini 328 3.3V เป็นบอร์ด Arduino ขนาดเล็ก ที่ใช้ MCU เบอร์ ATmega328 ซึ่งจะคล้ายกับบอร์ด Arduino Mini 05 แต่บนบอร์ดจะมี Regulator 3.3 V ชุดเดียวเท่านั้น ระดับแรงดันไฟฟ้า I/O คือ 3.3V



รูปที่ 2. 9 Arduino Pro Mini 328 3.3V

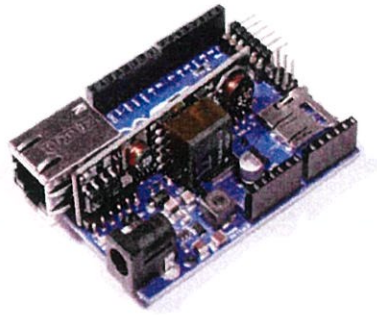
8. Arduino Pro Mini 328 5V เป็นบอร์ด Arduino ขนาดเล็ก ที่ใช้ MCU เบอร์ ATmega328 เช่นเดียวกับบอร์ด Arduino Mini 05 แต่บนบอร์ดจะมี Regulator 5V ชุดเดียวเท่านั้น ระดับแรงดันไฟฟ้า I/O คือ 5V



รูปที่ 2. 10 Arduino Pro Mini 328 5V

9. Arduino Ethernet with PoE module เป็นบอร์ด Arduino ที่ใช้ MCU เบอร์เดียวกับ Arduino Uno SMD ในบอร์ดมีชิป Ethernet และช่องสำหรับเสียบ SD Card รวมทั้งโมดูล POE ทำให้บอร์ดนี้สามารถใช้แหล่งจ่ายไฟจากสาย LAN ได้โดยตรง โดยไม่ต้องต่อ Adapter เพิ่ม แต่บอร์ด Arduino Ethernet with PoE module นี้จะไม่มีพอร์ต USB ทำให้เวลาโปรแกรมต้องต่อบอร์ด USB to Serial Converter

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



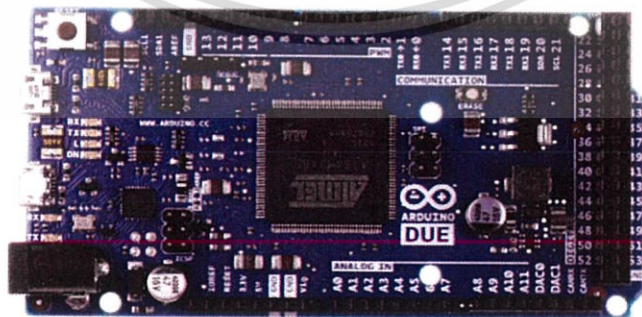
รูปที่ 2. 11 Arduino Ethernet with PoE module

10. Arduino Ethernet without PoE module บอร์ดนี้จะตัดโมดูล POE ออกไป ต้องใช้ไฟจากพอร์ต Power Jack เท่านั้น คุณสมบัติอื่นๆ จะเหมือนกับบอร์ด Arduino Ethernet with PoE module



รูปที่ 2. 12 Arduino Ethernet without PoE module

11. Arduino Due เป็นบอร์ด Arduino ที่เปลี่ยนชิป MCU ใหม่ ซึ่งจากเดิมเป็นตระกูล AVR เปลี่ยนเป็นเบอร์ AT91SAM3X8E(ตระกูล ARM Cortex-M3) แทน ทำให้การประมวลผลเร็วขึ้น แต่ยังคงรูปแบบโค้ดโปรแกรมของ Arduino ที่ง่ายอยู่ ข้อควรระวัง: เนื่องจาก MCU เป็นคนละเบอร์กับ Arduino Uno R3 อาจทำให้บอร์ด Shield บางตัวหรือ Library ใช้ร่วมกันกับบอร์ด Arduino Leonardo ไม่ได้ ผู้ใช้งานจำเป็นต้องตรวจสอบก่อนใช้งาน



รูปที่ 2. 13 Arduino Due

ที่มา : <http://www.thaieasyelec.com>

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.2 โครงสร้างการเขียนโปรแกรมภาษาซีของ Arduino

ภาษาซีของ Arduino จะจัดแบ่งรูปแบบโครงสร้างของการเขียนโปรแกรมออกเป็นส่วนย่อยๆ หลายๆ ส่วน โดยเรียกแต่ละส่วนว่าฟังก์ชัน และเมื่อนำฟังก์ชัน มารวมเข้าด้วยกันก็จะเรียกว่า โปรแกรมโดยโครงสร้างการเขียนโปรแกรมของ Arduino นั้น ทุกๆ โปรแกรมจะต้องประกอบไปด้วย ฟังก์ชัน จำนวนเท่าใดก็ได้ แต่อย่างน้อยที่สุดต้องมี ฟังก์ชัน จำนวน 2 ฟังก์ชัน คือ setup() และ loop() ดังตัวอย่าง

```
#include<Servo.h> //สั่งผนวกไฟล์ ชื่อ Servo.h เข้ามาใช้ในโปรแกรม
int Servo1 = 9; //กำหนดให้ Servo1 แทน Pin Digital-9
Servo myservo; //สร้าง Object ชื่อ myservo เพื่อควบคุม Servo
void setup()
{
myservo.attach(Servo1); //กำหนดให้ขา Digital-9 สร้างสัญญาณควบคุม Servo
}
void loop()
{
myservo.write(180); //กำหนดค่าตำแหน่งให้กับ Servo = 180 องศา
}
```

จะเห็นได้ว่าโครงสร้างพื้นฐานของภาษาซีที่ใช้กับ Arduino นั้น จะประกอบไปด้วย 3 ส่วนใหญ่ ๆ ด้วยกัน คือ

- Header ในส่วนนี้จะมีหรือไม่มีก็ได้ ถ้ามีต้องกำหนดในส่วนเริ่มต้นของโปรแกรม ซึ่งส่วนของ Header ได้แก่ ส่วนที่เป็น Compiler Directive ต่าง ๆ รวมไปถึงส่วนของการประกาศตัวแปร และค่าคงที่ต่าง ๆ ที่จะใช้ในโปรแกรม

- Setup() ในส่วนนี้เป็นฟังก์ชันบังคับที่ต้องกำหนดให้มีในทุกๆ โปรแกรม ถึงแม้ว่าในบางโปรแกรมจะไม่ต้องการใช้งานก็ยังจำเป็นต้องประกาศไว้ด้วยเสมอ เพียงแต่ไม่ต้องเขียนคำสั่งใด ๆ ไว้ในระหว่างวงเล็บปีกกา { } ที่ใช้เป็นตัวกำหนดขอบเขตของ ฟังก์ชัน โดยฟังก์ชัน นี้จะใช้สำหรับบรรจุคำสั่งในส่วนที่ต้องการให้โปรแกรมทำงานเพียงรอบเดียวตอนเริ่มต้นทำงานของโปรแกรมครั้งแรกเท่านั้น ซึ่งได้แก่คำสั่งเกี่ยวกับการ Setup ค่าการทำงานต่าง ๆ เช่น การกำหนดหน้าที่การใช้งานของ PinMode และการกำหนดค่า Baudrate สำหรับใช้งานพอร์ตสื่อสารอนุกรม เป็นต้น

- Loop() เป็นส่วนฟังก์ชัน บังคับที่ต้องกำหนดให้มีในทุกๆ โปรแกรมเช่นเดียวกับ ฟังก์ชัน setup() โดยฟังก์ชัน loop() นี้จะใช้บรรจุคำสั่งที่ต้องการให้โปรแกรมทำงานเป็นวงรอบซ้ำๆ กันไปไม่รู้จบ ซึ่งถ้าเปรียบเทียบกับรูปแบบของ ANSI-C ส่วนนี้คือ ฟังก์ชัน main() นั่นเอง

| | |
|---------|--|
| #header | #include<Servo.h> int Servo1=9; Servo myservo; |
| setup() | Setup() { myservo.attach(9); } |
| loop() | Loop() { myservo.write(180); } |

โครงสร้างและตัวอย่าง โปรแกรมของ Arduino
ที่มา : เอกชัย มะการ,2552

โครงสร้างโปรแกรมของ Arduino ส่วนแรกซึ่งถือเป็นส่วนเริ่มต้นของโปรแกรม ซึ่งจะเรียกว่า Header โดยประกอบด้วยคำสั่ง #include ซึ่งเป็นคำสั่งพิเศษที่เรียกว่า Compiler Directive ซึ่งไม่ใช่คำสั่งสำหรับสั่งงานในโปรแกรมหดงั้นคำสั่งจึงไม่ต้องมีเครื่องหมายเซมิโคลอนปิดท้ายคำสั่งเหมือนคำสั่งอื่นๆ โดย Compiler Directive จะใช้ทำหน้าที่สำหรับบอกให้ Compiler รับรู้เงื่อนไขในการแปลคำสั่งเท่านั้น ซึ่งในกรณีคำสั่ง #include จะใช้สำหรับบอกให้ Compiler รับรู้ว่าในการแปลคำสั่งของโปรแกรมนี้ มีไฟล์ภายนอกใดบ้างที่จำเป็นต้องใช้ร่วมในการแปลคำสั่งให้กับโปรแกรมนี้โดย จากตัวอย่างข้างต้นจะเป็นการบอกให้ Compiler ทำการผนวกไฟล์ชื่อ “Servo.h” เข้ามา ใช้เพื่อเรียกใช้คำสั่งต่างๆ ที่บรรจุไว้ เข้ามาใช้งานในโปรแกรม

2.3 สัญญาณบลูทูท

บลูทูท (Bluetooth) เป็นเทคโนโลยีคลื่นสัญญาณวิทยุระยะสั้นที่สามารถใช้ในการบลูทูท (Bluetooth) เป็นเทคโนโลยีคลื่นสัญญาณวิทยุระยะสั้นที่สามารถใช้ในการเชื่อมต่ออุปกรณ์เคลื่อนที่ต่างๆเข้าด้วยกัน โดยเฉพาะอย่างยิ่งเครือข่ายไร้สายแบบส่วนบุคคลหรือ PANs (Personal Area Networks) ข้อดีของเทคโนโลยีบลูทูท คือ ขนาดเล็กและใช้พลังงานน้อย ระยะเวลาของสัญญาณบลูทูทนั้นขึ้นอยู่กับกำลังส่งของตัวส่งสัญญาณซึ่งจะสัมพันธ์กับพลังงานที่ใช้โดยแบ่งเป็น 3 คลาสดังนี้

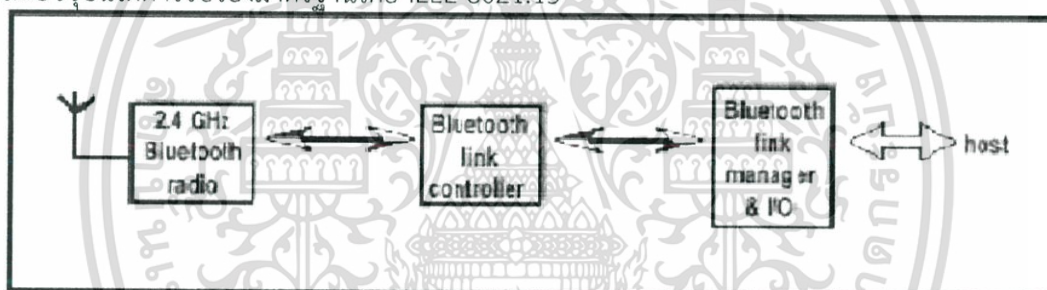
- คลาส 1 (Class 1) จะสามารถรับส่งข้อมูลได้ในระยะรัศมี 100 เมตร แต่จะใช้พลังงานประมาณ 100 mW ซึ่งประมาณครึ่งหนึ่งของอุปกรณ์ WiFi 802.11 ที่ใช้พลังงานประมาณ 250 mW
- คลาส 2 (Class 2) จะสามารถรับส่งข้อมูลได้ในระยะรัศมี 10 เมตร และจะใช้พลังงานประมาณ 2.5 mW ซึ่งเป็นที่นิยมใช้งานค่อนข้างมากเพราะใช้พลังงานค่อนข้างน้อย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- คลาส 3 (Class 3) จะสามารถรับส่งข้อมูลได้ในระยะรัศมี 10 เซนติเมตรถึง 1 เมตรและจะใช้พลังงาน ประมาณ 1 mW โดยคลาสนี้แม้จะใช้พลังงานน้อยที่สุดแต่ไม่เป็นที่นิยมเพราะระยะในการรับส่งข้อมูลค่อนข้างสั้น

วิวัฒนาการของอุปกรณ์บลูทูธเริ่มต้นจากบริษัทอีริคสัน (Ericsson) ได้กำหนดคุณสมบัติเบื้องต้นของอุปกรณ์บลูทูธขึ้นมา และภายหลังได้มีการจัดตั้งกลุ่มผู้สนใจบลูทูธเป็นพิเศษ หรือที่เรียกว่า Bluetooth SIG (Bluetooth Special Interest Group) ซึ่งประกาศเป็นทางการในวันที่ 20 พฤษภาคม 2542 ประกอบด้วยบริษัทโซนี่อีริคสัน (Sony Ericsson), ไอบีเอ็ม (IBM), อินเทล (Intel), โตชิบา (Toshiba) และ โนเกีย (Nokia) ต่อมาได้มีบริษัทผู้ผลิตอุปกรณ์อื่นๆ และองค์กร

มาตรฐานอุปกรณ์เข้าร่วมในการกำหนดคุณสมบัติของอุปกรณ์โดยเวอร์ชันของอุปกรณ์เริ่มจาก 1.0 และ 1.0B ซึ่งเป็นเวอร์ชันแรกๆที่มีข้อจำกัดคือไม่สามารถใช้งานแอปพลิเคชันพร้อมกันบนอุปกรณ์บลูทูธเดียวกันได้และได้รับการแก้ไขต่อมาเป็นเวอร์ชัน 1.1 ซึ่งนิยมใช้งานมากในปัจจุบัน และภายหลังได้มีการพัฒนาต่อเป็นเวอร์ชัน 1.2 และ 2.0 โดยเน้นที่การตัดการรบกวนสัญญาณและใช้พลังงานอย่างประหยัด เพื่อให้การรับส่งมีความเร็ว และรัศมีในการรับส่งข้อมูลเพิ่มขึ้นและปัจจุบันได้มีการรับรองมาตรฐานโดย IEEE 8021.15

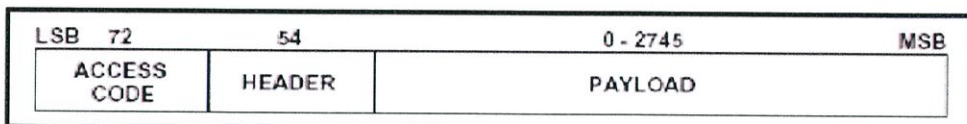


รูปที่ 2. 14 การทำงานของระบบบลูทูธ ; ที่มา : <http://cs.tu.ac.th>

การทำงานของระบบบลูทูธตามรูปที่ 1 นั้นมีส่วนประกอบที่สำคัญคือ ส่วนคลื่นสัญญาณ , ส่วนเชื่อมต่อและควบคุม, ส่วนบริหารการเชื่อมต่อกับ Input/Output และส่วน HCI (Host Controller Interface) ซึ่งก็คือ ส่วนของการติดต่อกับอุปกรณ์ที่มีการใช้งานสัญญาณบลูทูธซึ่งได้แก่ เครื่องคอมพิวเตอร์หรือโทรศัพท์มือถือ เป็นต้น

2.3.1 รูปแบบของบลูทูธแพ็คเกจ (Bluetooth Packet)

ในการส่งข้อมูลนั้นจะเป็นการส่งทีละแพ็คเกจ โดยแต่ละแพ็คเกจจะประกอบไปด้วย 3 ส่วนย่อยๆ ได้แก่ ส่วน Access Code ส่วน Header และ ส่วน Payload สำหรับรูปแบบของแพ็คเกจ และจำนวนบิตที่ใช้ในแต่ละส่วนนั้นแสดงตามรูปที่ 2 โดยขนาดของ Access Code และ Header จะมีขนาดคงที่ (Fixed) คือ 72 และ 54 บิตตามลำดับ ส่วน Payload นั้นมีขนาดขึ้นอยู่กับลักษณะการใช้งานซึ่งอยู่ในช่วงระหว่าง 0 - 2745 บิต

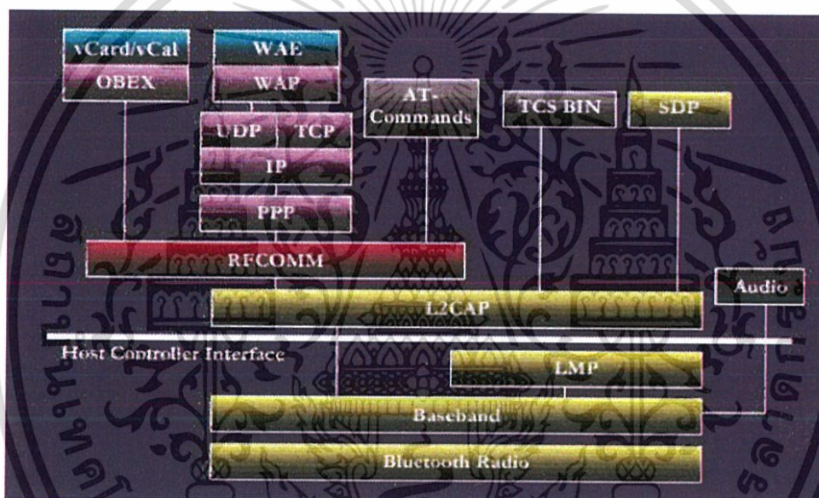


รูปที่ 2. 15 รูปแบบของบลูทูธแพ็คเกจ ; ที่มา : <http://cs.tu.ac.th>

ลักษณะการใช้งานแพ็คเกจจะแบ่งเป็น 2 ประเภท คือแพ็คเกจควบคุม (Control Packet) สามารถมีได้เพียงแค่ Access Code หรือมี Access Code กับ Header โดยไม่ต้องมี Payload ส่วนแพ็คเกจข้อมูลนั้น จำเป็นจะต้องมีครบสมบูรณ์ทั้ง 3 ส่วน

2.3.2 โพรโทคอลของการสื่อสารผ่านทางบลูทูธ (Bluetooth Protocol)

ข้อตกลงในการติดต่อสื่อสารผ่านทางบลูทูธนั้นมีมากมายหลายรูปแบบด้วยกัน โดยการที่ทั้ง 2 ฝ่ายจะสื่อสารกันได้ต้องใช้ข้อตกลงเดียวกัน ซึ่งรูปที่ 3 จะแสดงข้อตกลงแต่ละระดับชั้นของการสื่อสารผ่านทางบลูทูธ



รูปที่ 2. 16 บลูทูธโพรโทคอล

; ที่มา : <http://cs.tu.ac.th>

- Bluetooth Core Protocols
 - Base band และ Link Control ทั้งคู่เป็นส่วนที่ใช้เชื่อมต่อกันระหว่างอุปกรณ์ Bluetooth ในชั้นนี้มีหน้าที่สำคัญในการจับคู่สัญญาณความถี่คลื่นวิทยุ
 - Audio เป็นส่วนที่เชื่อมต่อโดยตรงกันกับ Base band ใช้สำหรับการส่งและรับข้อมูลประเภทเสียง
 - Link Manager Protocol (LMP) ทำหน้าที่เชื่อมต่อและควบคุมการทำงานต่างๆ เช่น เข้ารหัส และการตรวจสอบแพ็คเกจที่มาจาก Base band
 - Logical Link Control and Adaptation Protocol (L2CAP) มีหน้าที่ในการรวมและตรวจสอบความถูกต้องของข้อมูลทีมาจากแต่ละแพ็คเกจ
 - Service Discovery Protocol (SDP) มีหน้าที่ในการสำรวจตรวจสอบข้อมูลและลักษณะพิเศษของอุปกรณ์บลูทูธอื่นๆ
- Cable Replacement Protocol

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

RFCOMM ทำหน้าที่จำลองข้อมูลที่ได้จาก L2CAP เป็นสัญญาณที่สามารถใช้ได้ ในแอปพลิเคชัน

- Telephony Protocol

Telephony Control Protocol-Binary (TCS-BIN) ทำหน้าที่กำหนดสัญญาณการควบคุม สำหรับสร้างข้อมูลเสียง

- Adopted Protocols

- OBEX (Object Exchange) เป็นโปรโตคอลที่ใช้ในการส่งข้อมูลไฟล์

- TCP/UIP/IP เป็นตัวกำหนดวิธีการที่จะให้อุปกรณ์บลูทูธสามารถติดต่อสื่อสารกับอุปกรณ์อื่นๆ ในกรณีที่เป็นการเชื่อมต่ออินเทอร์เน็ต รายละเอียดการใช้จะเป็นTCP/IP/PPP ส่วนในกรณีที่ เป็นสำหรับ WAP จะใช้ UDP/IP/PPP

2.4 Module hc05

HC-05 Bluetooth Module เป็นโมดูลไร้สายที่ใช้สื่อสารกันด้วย Bluetooth SPP (Serial Port Protocol) โดย Serial Port เป็น Bluetooth V2.0+EDR (Enhance Data Rate) 3Mbps Modulation พร้อมกับความถี่ใช้งาน 2.4 GHz สามารถเชื่อมต่อกับอุปกรณ์บลูทูธอื่นๆได้ รวมทั้งเชื่อมต่อกับ Serial Interface ระหว่างคอมพิวเตอร์ด้วยวงจร RS232 เพื่อเข้าสู่ AT Command ในการปรับแต่งค่าต่างๆของโมดูล โมดูลรุ่นนี้สามารถตั้งค่าให้เป็นที่ทั้ง Master และ Slave Mode



รูปที่ 2. 17 module Hc-05

ที่มา : <http://www.synes.co.th/>

คุณสมบัติต่างๆ

- ความตอบสนองสัญญาณ -80dBm
- กำลังส่งคลื่นวิทยุเพิ่มได้ถึง +4dbm
- แรงดัน 1.8-3.6v
- PIO Control
- UART Interface พร้อมกับสามารถปรับแต่ง baud rate ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- เสอาอากาศในตัว

รายละเอียดการทำงานต่างๆ

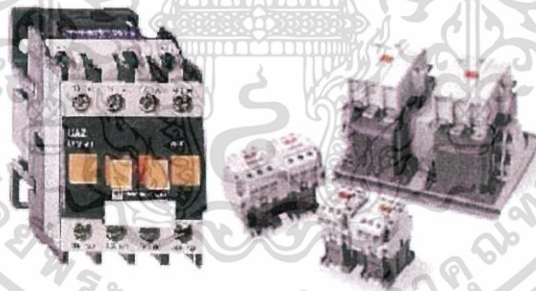
- Baud Rate ปกติ:38400, Data bits:8, Stop Bits:1, Parity:No
- รองรับ Baud Rate : 9600,19200,38400,57600,115200,230400,460800

2.5 รีเลย์

รีเลย์ (อังกฤษ: relay) คือ อุปกรณ์อิเล็กทรอนิกส์ที่ทำหน้าที่เป็นสวิตช์ตัด-ต่อวงจร โดยใช้แม่เหล็กไฟฟ้า[1] และการที่จะให้มันทำงานก็ต้องจ่ายไฟให้มันตามที่กำหนด เพราะเมื่อจ่ายไฟให้กับตัวรีเลย์ มันจะทำให้หน้าสัมผัสติดกัน กลายเป็นวงจรปิด และตรงข้ามทันทีที่ไม่ได้จ่ายไฟให้มัน มันก็จะกลายเป็นวงจรเปิด ไฟที่เราใช้ป้อนให้กับตัวรีเลย์ก็จะเป็นไฟที่มาจาก เพาเวอร์ๆ ของเครื่องเรา ดังนั้นทันทีที่เปิดเครื่อง ก็จะทำให้รีเลย์ทำงานประเภทของรีเลย์ เป็นอุปกรณ์ทำหน้าที่เป็นสวิตช์มีหลักการทำงานคล้ายกับ ขดลวดแม่เหล็กไฟฟ้าหรือโซลินอยด์ (solenoid) รีเลย์ใช้ในการควบคุมวงจร ไฟฟ้าได้อย่างหลากหลาย รีเลย์เป็นสวิตช์ควบคุมที่ทำงานด้วยไฟฟ้า

2.5.1 ลักษณะการใช้งานได้เป็น 2 ประเภทคือ

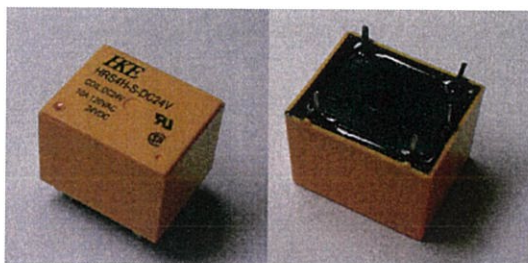
- รีเลย์กำลัง (power relay) หรือมักเรียกกันว่าคอนแทกเตอร์ (Contactor or Magneticcontactor)ใช้ในการควบคุมไฟฟ้ากำลัง มีขนาดใหญ่กว่ารีเลย์ธรรมดา



รูปที่ 2. 18 รีเลย์กำลัง ; ที่มา : motor.lpc.rmutl.ac.th

- รีเลย์ควบคุม (control Relay) มีขนาดเล็กกำลังไฟฟ้าต่ำ ใช้ในวงจรควบคุมทั่วไปที่มีกำลังไฟฟ้าไม่มากนัก หรือเพื่อการควบคุมรีเลย์หรือคอนแทกเตอร์ขนาดใหญ่ รีเลย์ควบคุม บางทีเรียกกันง่าย ๆ ว่า "รีเลย์"

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2. 19 รีเลย์ควบคุม ; ที่มา : www.kmitl.ac.th

2.5.2 ชนิดของรีเลย์

การแบ่งชนิดของรีเลย์สามารถแบ่งได้ 11 แบบ คือชนิดของรีเลย์แบ่งตามลักษณะของคอยล์ หรือ แบ่งตามลักษณะการใช้งาน (Application) ได้แก่รีเลย์ดังต่อไปนี้

- รีเลย์กระแส (Current relay) คือ รีเลย์ที่ทำงานโดยใช้กระแสมีทั้งชนิดกระแสขาด (Under-current) และกระแสเกิน (Over current)
- รีเลย์แรงดัน (Voltage relay) คือ รีเลย์ ที่ทำงานโดยใช้แรงดันมีทั้งชนิดแรงดันขาด (Under-voltage) และ แรงดันเกิน (Over voltage)
- รีเลย์ช่วย (Auxiliary relay) คือ รีเลย์ที่เวลาใช้งานจะต้องประกอบเข้ากับรีเลย์ชนิดอื่น จึงจะทำงานได้
- รีเลย์กำลัง (Power relay) คือ รีเลย์ที่รวมเอาคุณสมบัติของรีเลย์กระแส และรีเลย์แรงดันเข้าด้วยกัน
- รีเลย์เวลา (Time relay) คือ รีเลย์ที่ทำงานโดยมีเวลาเข้ามาเกี่ยวข้องกับ ซึ่งมียูต์ด้วยกัน 4 แบบ คือ

-รีเลย์กระแสเกินชนิดเวลาผกผันกับกระแส (Inverse time over current relay) คือ รีเลย์ ที่มีเวลาทำงานเป็นส่วนกลับกับกระแส

- รีเลย์กระแสเกินชนิดทำงานทันที (Instantaneous over current relay) คือรีเลย์ที่ทำงานทันทีทันใดเมื่อมีกระแสไหลผ่านเกินกว่าที่กำหนดที่ตั้งไว้

- รีเลย์แบบดีฟิไนต์ไทม์แล็ก (Definite time lag relay) คือ รีเลย์ ที่มีเวลาการทำงานไม่ขึ้นอยู่กับความมากมายของกระแสหรือค่าไฟฟ้าอื่นๆ ที่ทำให้เกิดงานขึ้น

- รีเลย์แบบอินเวอร์สดีฟิไนต์ไทม์แล็ก (Inverse definite time lag relay) คือ รีเลย์ ที่ทำงานโดยรวมเอาคุณสมบัติของเวลาผกผันกับกระแส (Inverse time) และ แบบดีฟิไนต์ไทม์แล็ก (Definite time lag relay) เข้าด้วยกัน

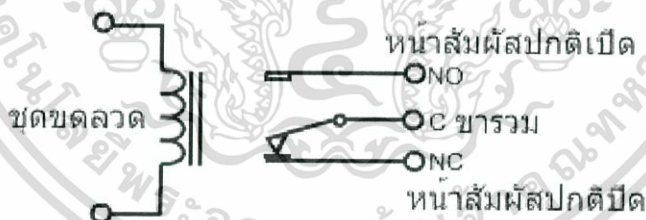
- รีเลย์กระแสต่าง (Differential relay) คือ รีเลย์ที่ทำงานโดยอาศัยผลต่างของกระแส
- รีเลย์มีทิศ (Directional relay) คือรีเลย์ที่ทำงานเมื่อมีกระแสไหลผิดทิศทาง มีแบบรีเลย์กำลังมีทิศ (Directional power relay) และรีเลย์กระแสมีทิศ (Directional current relay)
- รีเลย์ระยะทาง (Distance relay) คือ รีเลย์ระยะทางมีแบบต่างๆ ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- รีแอกแตนซ์รีเลย์ (Reactance relay)
- อิมพีแดนซ์รีเลย์ (Impedance relay)
- โมห์รีเลย์ (Mho relay)
- โอห์มรีเลย์ (Ohm relay)
- โพลาริซมอห์รีเลย์ (Polarized mho relay)
- ออฟเซตมอห์รีเลย์ (Off set mho relay)
- รีเลย์อุณหภูมิ (Temperature relay) คือ รีเลย์ที่ทำงานตามอุณหภูมิที่ตั้งไว้
- รีเลย์ความถี่ (Frequency relay) คือ รีเลย์ที่ทำงานเมื่อความถี่ของระบบต่ำกว่าหรือมากกว่าที่ตั้งไว้บูคโฮลซ์รีเลย์ (Buchholz 's relay) คือรีเลย์ที่ทำงานด้วยก๊าซ ใช้กับหม้อแปลงที่แช่อยู่ในน้ำมันเมื่อเกิด ฟอลต์ ขึ้นภายในหม้อแปลง จะทำให้น้ำมันแตกตัวและเกิดก๊าซขึ้นภายในไปดันหน้าสัมผัส ให้รีเลย์ทำงาน

2.5.3 โครงสร้างและสัญลักษณ์ของรีเลย์

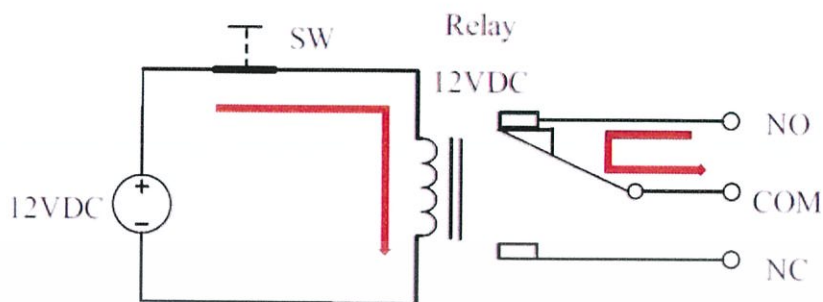
โครงสร้างและสัญลักษณ์ของรีเลย์โครงสร้างภายในของรีเลย์จะประกอบไปด้วยขดลวด 1 ชุด และหน้าสัมผัสซึ่งในหน้าสัมผัส 1 ชุดซึ่งจะประกอบด้วยหน้าสัมผัสแบบปกติปิด (Normally Close หรือ NC) และหน้าสัมผัสแบบปกติเปิด (Normally Open หรือ NO) ในสภาวะปกติขา NC นี้จะต่ออยู่กับขาร่วม(Common หรือ COM) และเมื่อมีการจ่ายกระแสไฟฟ้าในปริมาณที่เพียงพอเข้าที่ขดลวดรีเลย์หรือมีแรงดันตกคร่อมขดลวดจะทำให้เกิดการเปลี่ยนแปลงที่หน้าสัมผัสทำให้จุด COM ต่อกับหน้าสัมผัส NO ในรีเลย์ 1 ตัวอาจมีหน้าสัมผัสมากกว่า 1 ชุดซึ่งขึ้นอยู่กับผู้ผลิต



รูปที่ 2. 20 สัญลักษณ์ของรีเลย์ ; ที่มา : www.atom.rmutphysics.com

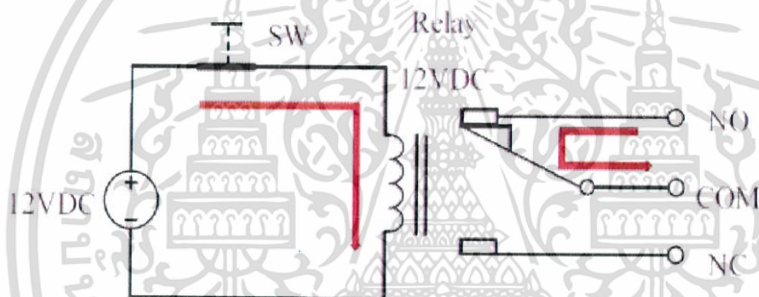
2.5.4 หลักการทำงานของรีเลย์ (Relay)

จากรูปที่ 21 เมื่อไม่จ่ายกระแสไฟฟ้าที่ขดลวดรีเลย์ทำให้จุดร่วมต่ออยู่กับหน้าสัมผัสแบบปกติปิด(NC) แต่เมื่อทำการจ่ายกระแสไฟฟ้าให้กับขดลวดรีเลย์ (Coil) จะทำให้เกิดสนามแม่เหล็กกรอบขดลวดซึ่งอำนาจแม่เหล็กชั่วคราวที่เกิดขึ้นมีค่าเพียงพอที่จะชนะแรงสปริงเกิดการเปลี่ยนแปลงที่หน้าสัมผัสทำให้จุดร่วมต่ออยู่กับหน้าสัมผัสแบบปกติเปิด(NO) ดังแสดงในรูปที่ 2.22 และเมื่อไม่ได้จ่ายกระแสไฟฟ้าสปริงจะดึงหน้าสัมผัสกลับมาต่อกับหน้าสัมผัสแบบปกติปิด(NC) อีกครั้งหนึ่ง



รูปที่ 2. 21 สถานะขดลวดรีเลย์ขณะไม่มีกระแสไฟฟ้า

; ที่มา : <http://archive.lib.cmu.ac.th>



รูปที่ 2. 22 สถานะขดลวดรีเลย์เมื่อมีกระแสไฟฟ้า

; ที่มา : <http://archive.lib.cmu.ac.th>

2.6 วงจรขับรีเลย์ด้วยทรานซิสเตอร์

2.6.1 เต้นด้านควบคุม

- ใช้ไฟเพียงเล็กน้อย กระตุ้นทางขา B ของทรานซิสเตอร์ ควบคุมกำลังไฟฟ้ามาก ๆ ให้ไหลผ่านขา C – E ของมัน มายังโหลดได้ดี
- ควบคุมการเปิด-ปิดไฟให้โหลดทำได้รวดเร็วมาก เพราะไม่ส่วนที่เป็นกลไกเหมือนสวิตช์ทั่วไป ดังนั้นจึงนิยมใช้เป็นวงจรรีเลย์สัญญาณทั้งไฟกระแสตรงและกระแสสลับ

2.6.2 พื้นฐานของการต่อทรานซิสเตอร์

ทรานซิสเตอร์ คือ สวิตซ์อิเล็กทรอนิกส์ที่ดี โดยทรานซิสเตอร์มี 2 ชนิด คือ ชนิด NPN และ ชนิด PNP ซึ่งมีต่อในวงจรอิเล็กทรอนิกส์ทั่วไปต่างกัน

การต่อทรานซิสเตอร์ชนิด NPN ตามรูปที่ 23 โดยระหว่างขา C-E ของทรานซิสเตอร์อาจจะเปรียบได้กับขาของสวิตช์ ในกรณีสวิตช์นั้น สามารถควบคุมการเปิด-ปิดไฟไปยังโหลดด้วยการกดที่ปุ่ม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สวิตช์ แต่สำหรับทรานซิสเตอร์ใช้ไฟค่าน้อยๆ (B2) กระตุ้นที่ขา B ก็ทำให้ระหว่างขา C-E ของทรานซิสเตอร์เปรียบเป็นสวิตช์ต่อไฟจากแหล่งจ่ายไฟ B1 จึงไหลผ่านโหลดได้ โดยขาต่างๆของทรานซิสเตอร์ต้องได้รับการจ่ายไฟที่เหมาะสม ดังนี้

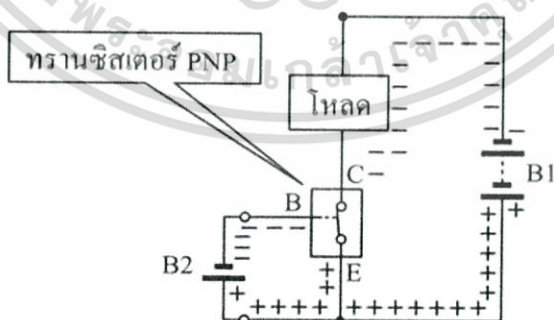
- ไฟกระตุ้นขา B ต้องมีค่าเป็นไฟบวก เมื่อเทียบกับขา E เสมอ และมีค่าแรงดันสูงกว่า 0.7V ด้วย จึงจะสามารถควบคุมให้ทรานซิสเตอร์ทำงานได้
- ขา C ต้องรับไฟบวก และขา E ต้องจ่ายไฟลบเสมอ หากป้อนไฟผิดขั้วทรานซิสเตอร์ก็จะไม่ทำงาน



รูปที่ 2. 23 โครงสร้างการทำงานของทรานซิสเตอร์ NPN

ที่มา : <http://diyelec.blogspot.com/>

- ขา E ต่อกับไฟลบเสมอ เมื่อวงจรจรอิเล็กทรอนิกส์ส่วนใหญ่จะใช้ขากราวด์เป็นไฟลบ จึงทำให้ทรานซิสเตอร์ชนิด NPN เป็นที่นิยมใช้งานกันมาก



รูปที่ 2. 24 โครงสร้างการทำงานของทรานซิสเตอร์ PNP

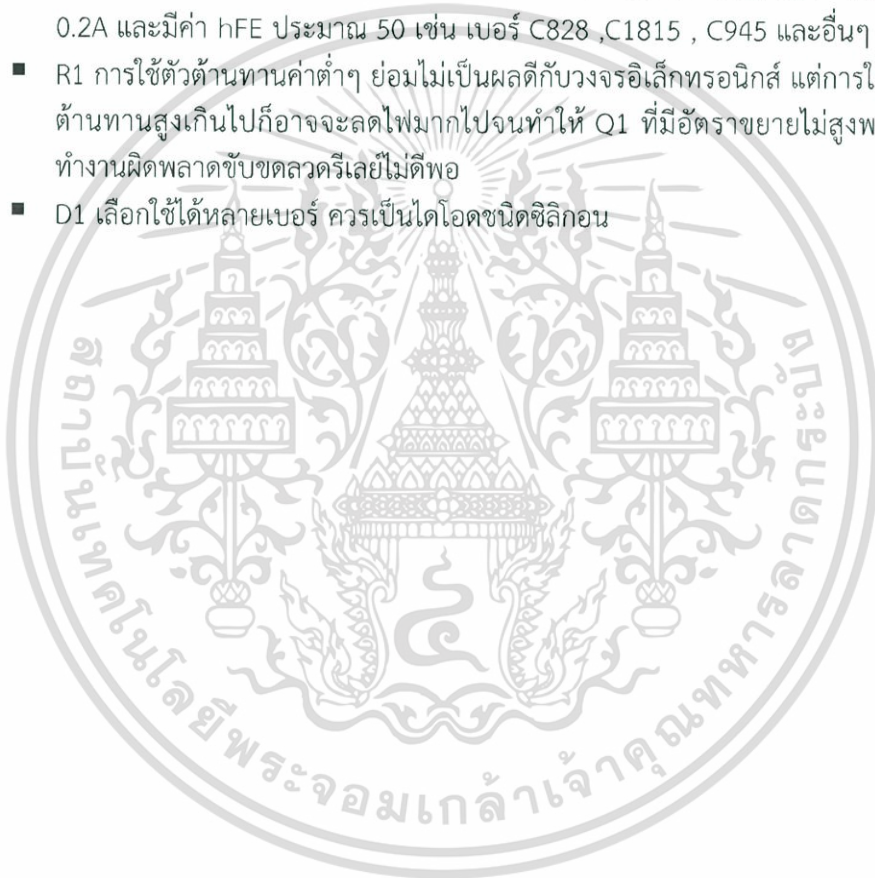
ที่มา : <http://diyelec.blogspot.com/>

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อใช้ไฟเลี้ยงประมาณ 12V จึงใช้รีเลย์ขนาด 12V วงจรอิเล็กทรอนิกส์จ่ายแรงดันออกมาประมาณ 8V(หรือใกล้เคียง) ส่วนกระแสที่จ่ายออกมาไม่เกิน 0.02A ทำให้ไม่สามารถขยับขดลวดรีเลย์ได้ จึงเพิ่มทรานซิสเตอร์ Q1 เบอร์ BC549 เพื่อช่วยขยายกระแสให้สูงขึ้นจนขดลวดรีเลย์ได้ ส่วน R1 ทำหน้าที่ลดไฟจากขาออกของวงจรรีเลย์ให้เหลือประมาณ 0.7V ซึ่งเพียงพอที่ทรานซิสเตอร์จะทำงานได้ปกติ

สำหรับ D1 นั้นใส่ไว้เพื่อป้องกันไฟย้อนกลับจากขดลวดรีเลย์ไหลมาทำอันตรายกับวงจรต่าง อุปกรณ์ที่สามารถเลือกใช้ได้อย่างยืดหยุ่นมาก รายละเอียดดังนี้

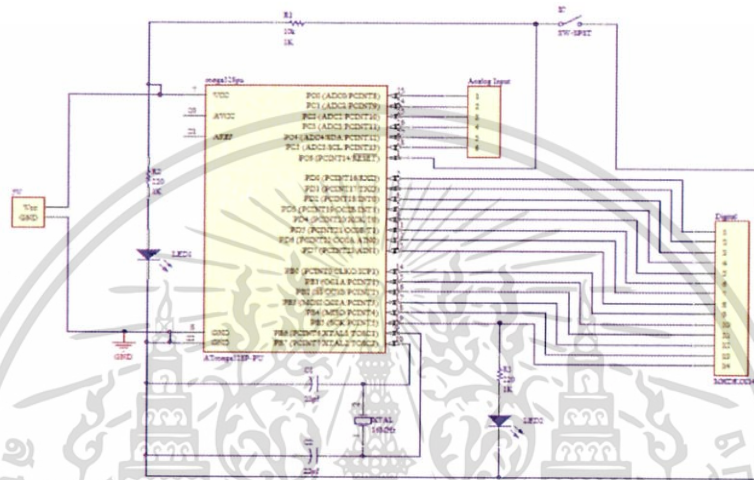
- Q1 เป็นทรานซิสเตอร์ NPN เบอร์ใดก็ได้ ทนแรงดันได้ประมาณ 15V ทนกระแสได้ 0.2A และมีค่า hFE ประมาณ 50 เช่น เบอร์ C828 ,C1815 , C945 และอื่นๆ
- R1 การใช้ตัวต้านทานค่าต่างๆ ย่อมไม่เป็นผลดีกับวงจรรีเลย์อิเล็กทรอนิกส์ แต่การใช้ค่าความต้านทานสูงเกินไปก็อาจจะลดไฟมากไปจนทำให้ Q1 ที่มีอัตราขยายไม่สูงพอ อาจจะทำงานผิดพลาดขดลวดรีเลย์ไม่ดีพอ
- D1 เลือกใช้ได้หลายเบอร์ ควรเป็นไดโอดชนิดซิลิกอน



บทที่ 3

การออกแบบ

3.1 บอร์ดไมโครคอนโทรลเลอร์



รูปที่ 3.1 วงจรสมมูลของบอร์ดไมโครคอนโทรลเลอร์

3.2 แอปพลิเคชันที่ใช้ในการควบคุม



รูปที่ 3.2 แอปพลิเคชันที่ใช้ในการควบคุมรีเลย์

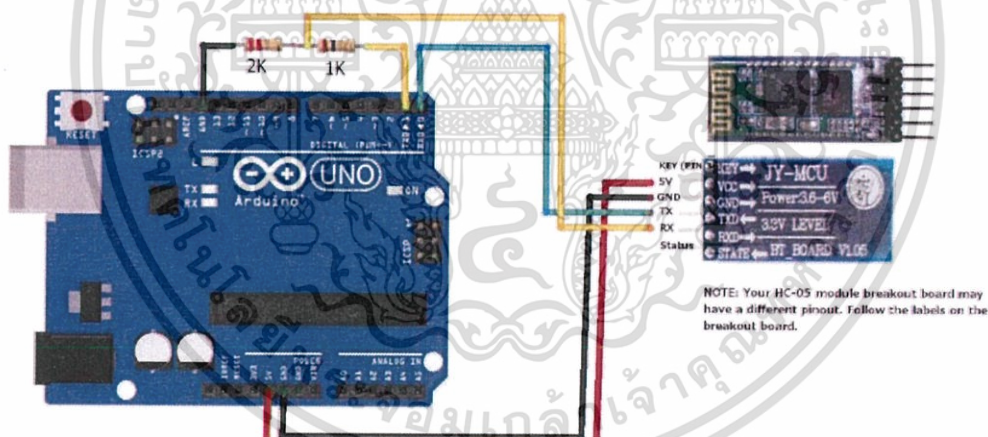
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ArduDroid เป็นแอปพลิเคชันที่เป็นเครื่องมือที่ง่ายเพื่อจะช่วยให้คุณควบคุม Arduino Uno จากโทรศัพท์ Android ผ่านทางโมดูลบลูทูธ HC-05 โมดูลบลูทูธ ArduDroid ทำหน้าที่ต่อไปนี้

- 1) ควบคุมดิจิตอล Arduino และขา PWM
- 2) ส่งคำสั่งข้อความให้กับ Arduino Uno
- 3) รับ / ส่งข้อมูลจาก Arduino Uno โดยใช้บลูทูธ HC05

3.3 HC-05 บลูทูธโมดูล

HC-05 Bluetooth Module เป็นโมดูลไร้สายที่ใช้สื่อสารกันด้วย Bluetooth SPP (Serial Port Protocol) โดย Serial Port เป็น Bluetooth V2.0+EDR (Enhance Data Rate) 3Mbps Modulation พร้อมกับความถี่ใช้งาน 2.4 GHz สามารถเชื่อมต่อกับอุปกรณ์บลูทูธอื่นๆได้ รวมทั้งเชื่อมต่อกับ Serial Interface ระหว่างคอมพิวเตอร์ด้วยวงจร RS232 เพื่อเข้าสู่ AT Command ในการปรับแต่งค่าต่างๆของโมดูล



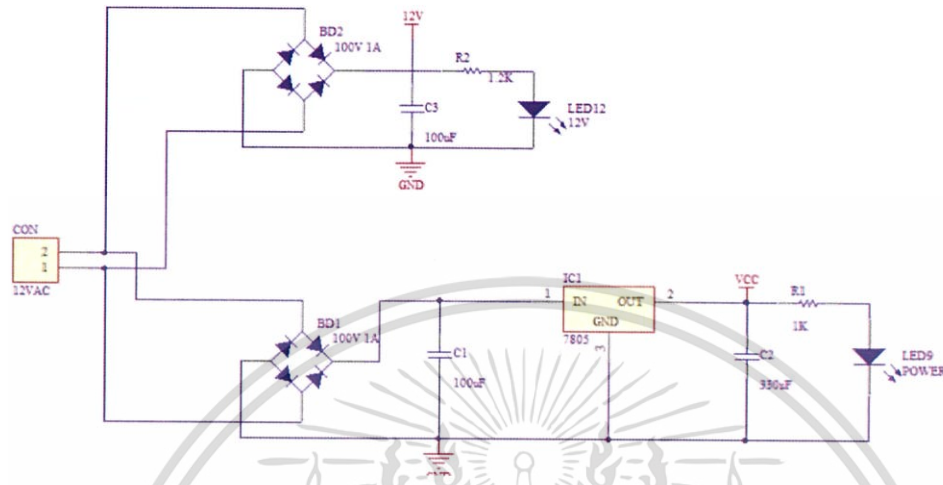
รูปที่ 3.3 การเชื่อมต่อระหว่าง HC05 module กับบอร์ดไมโครคอนโทรลเลอร์

ที่มา : <https://play.google.com>

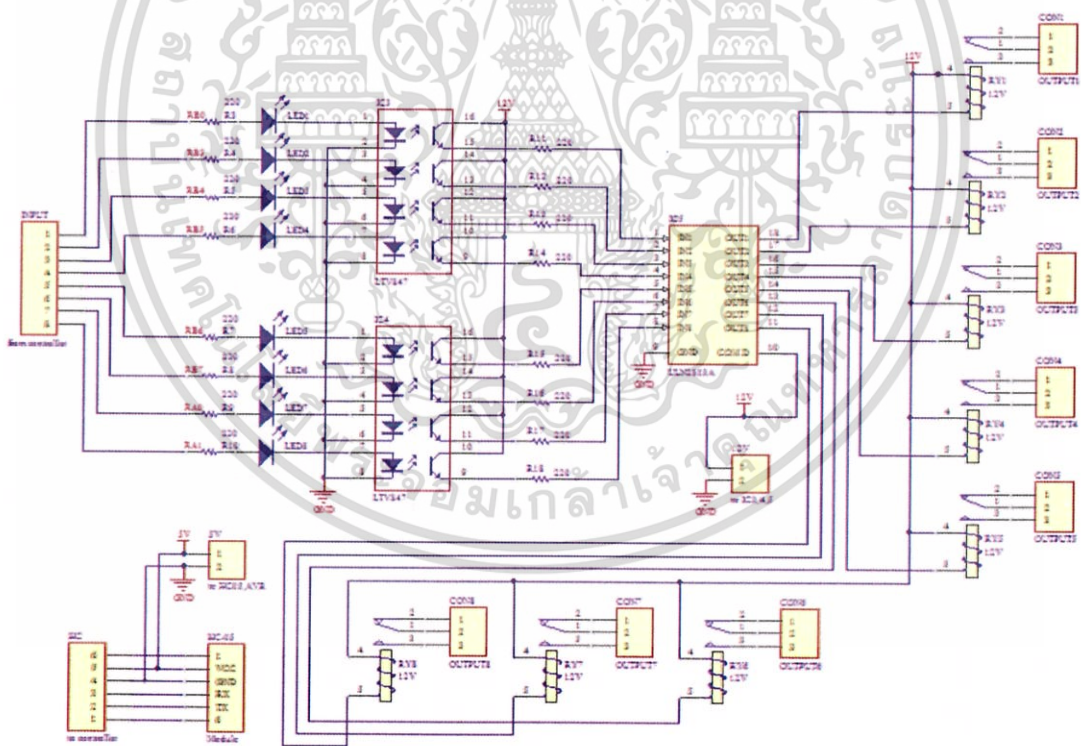
จากรูปที่ 3.3 แสดงจะเห็นได้ว่าขาVccของHC05ต่อกับไฟเลี้ยง5Vจากบอร์ดไมโครคอนโทรลเลอร์ เช่นเดียวกันของGNDของHC05ต่อกับGNDที่บอร์ดไมโครคอนโทรลเลอร์ ขาRx ของHC05ต่อกับoutputระหว่าง port1และGNDโดยมีตัวต้านทาน2ตัวแบ่งแรงดันไฟต่ออยู่ด้วย และสุดท้ายขาTxของHC05ต่อกับoutput port0ของบอร์ดไมโครคอนโทรลเลอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4 วงจรที่ใช้ในการขับรีเลย์



รูปที่ 3. 4 วงจรแปลงไฟฟ้ากระแสสลับเป็นไฟฟ้ากระแสตรง



รูปที่ 3. 5 วงจรสมมูลที่ใช้ในการขับรีเลย์

จากรูปที่ 3.4 BD1 คือ bridge rectifier ซึ่งทำหน้าที่ในการแปลงสัญญาณไฟ 12VAC เป็น 5VDC เพื่อที่จะนำไปใช้เป็นไฟเลี้ยงของบอร์ดไมโครคอนโทรลเลอร์ BD2 คือ bridge rectifier ซึ่งทำหน้าที่ในการแปลงสัญญาณไฟ 12VAC เป็น 12VDC เพื่อนำไปเป็นไฟเลี้ยงให้กับ IC3,IC4 และ IC5 ต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 3.5 IC3,IC4 คือ LTV827 ซึ่งเป็นวงจรรวมที่เป็นอุปกรณ์เชื่อมต่อทางแสง (Optocouplers)

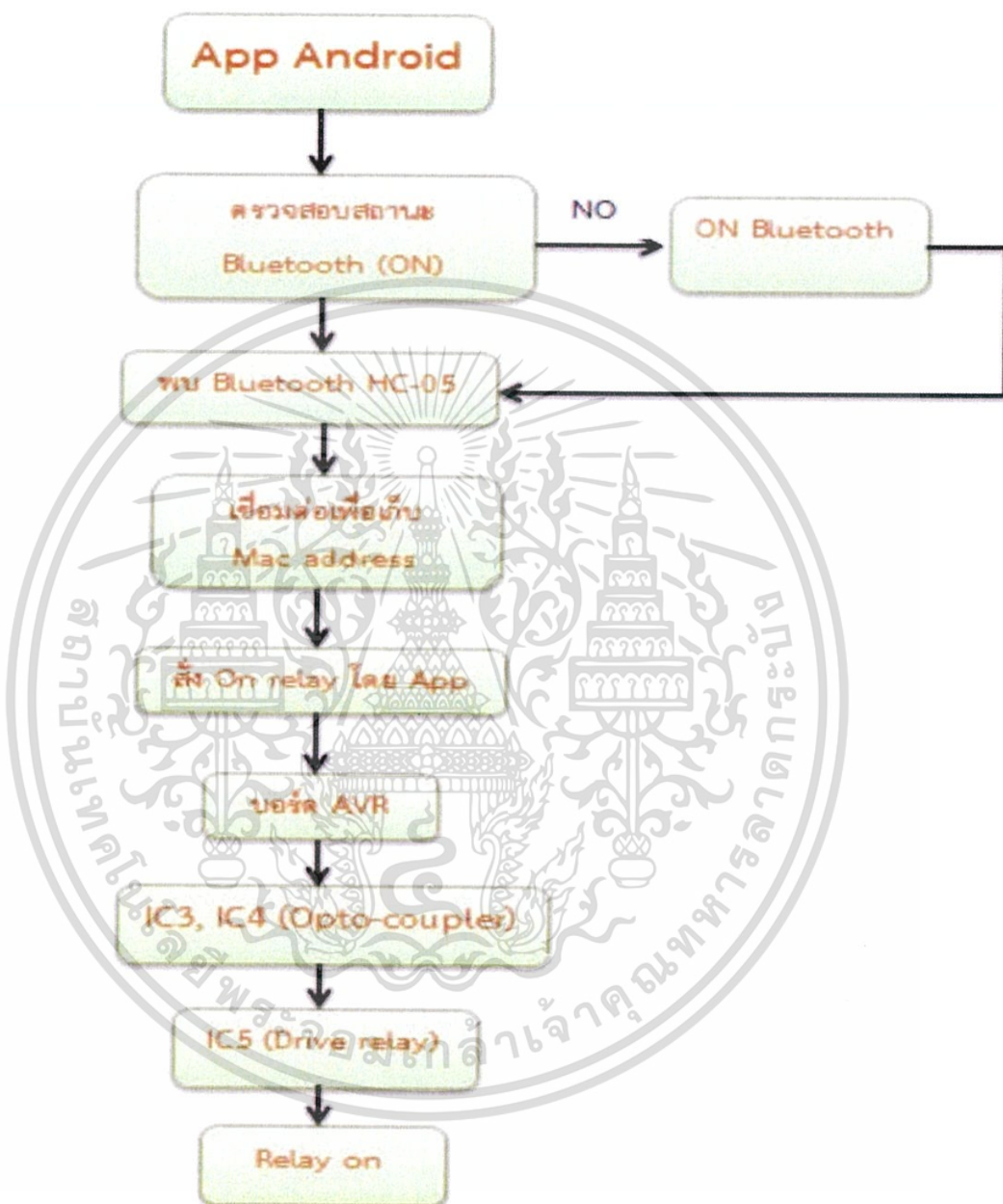
ทำหน้าที่ในการรับ input หรือ output ของไมโครคอนโทรลเลอร์เข้ามา ซึ่งเป็นอุปกรณ์ทางอิเล็กทรอนิกส์สำหรับการเชื่อมต่อทางแสง โดยการเปลี่ยนสัญญาณไฟฟ้าให้เป็นแสงแล้วเปลี่ยนกลับเป็นสัญญาณไฟฟ้าตามเดิมนิยมใช้สำหรับการเชื่อมต่อสัญญาณระหว่างสองวงจร และต้องการแยกกันทางไฟฟ้าโดยเด็ดขาด เพื่อป้องกันการรบกวนกันทางไฟฟ้าระหว่างสองวงจร

IC5 เป็น ULN2803A เป็นไอซีที่ถูกใช้งานกันมากอีกเบอร์หนึ่ง โดยแท้จริงก็คือทรานซิสเตอร์กับไดโอด ที่ถูกเอามาต่อรวมกันในตัวถึงเดียวกัน แต่มันทำให้มีความสะดวกในการใช้งาน ทั้งในการต่อสาย หรือออกแบบ PCB เหมาะสำหรับการใช้ในการเชื่อมต่อกับวงจรที่มีระดับแรงดันไฟฟ้าในการทำงานที่แตกต่างกัน เช่น วงจร Driver LED หรือ มอเตอร์ แต่ก็มีข้อเสียในการออกแบบที่ต้องคำนึงถึง คือ จะต้องถูกขับด้วยแรงดันบวก หากใช้ร่วมกับวงจรที่ต่อกับไมโครคอนโทรลเลอร์บางเบอร์อาจเกิดปัญหาในการทำงาน เช่น ไมโครคอนโทรลเลอร์เบอร์นั้น เมื่อเริ่มการทำงานกำหนดให้ทุกขาเป็นลอจิก 1 ก็จะทำให้ส่วนที่ต่อกับไอซีนี้ทำงานด้วย จึงควรระวังเป็นอย่างมากในการนำไปใช้งาน แต่ถ้ามีความรอบครอบในการออกแบบแล้ว ไอซี ULN2803A ตัวนี้ก็ช่วยลดความยุ่งยากในการออกแบบวงจรไปได้เยอะ

Input 8 channel คือ output ที่ได้จากไมโครคอนโทรลเลอร์เพื่อนำไปเข้าออปโตคัปเบิล IC3,IC4 เพื่อทำการแปลงสัญญาณต่อไป

RY1-RY8 คือรีเลย์ 12V ในโครงงานนี้เป็นจะเป็นส่วน output สุดท้ายเพื่อที่จะนำไปใช้งานภายนอกต่อไป ซึ่งจะทำหน้าที่เป็น switch เพื่อใช้ในการควบคุมอุปกรณ์ภายนอกต่อไป

3.5 Flow Chart ระบบการทำงาน

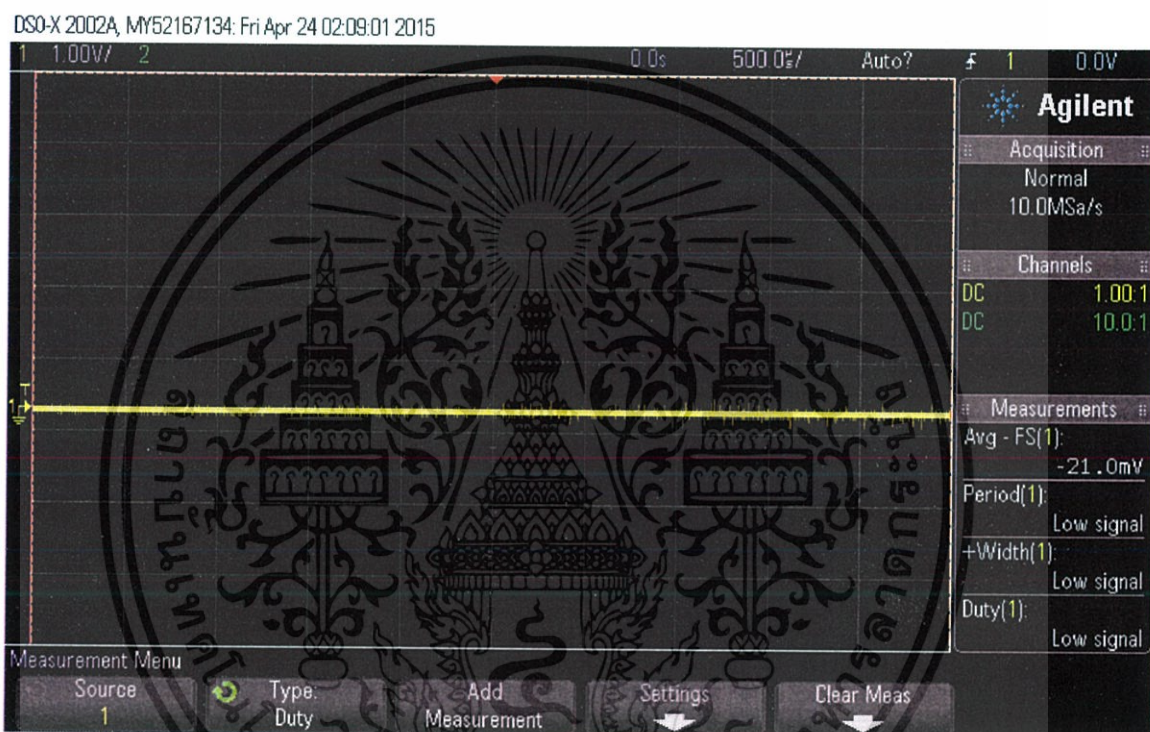


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

ผลการทดลอง

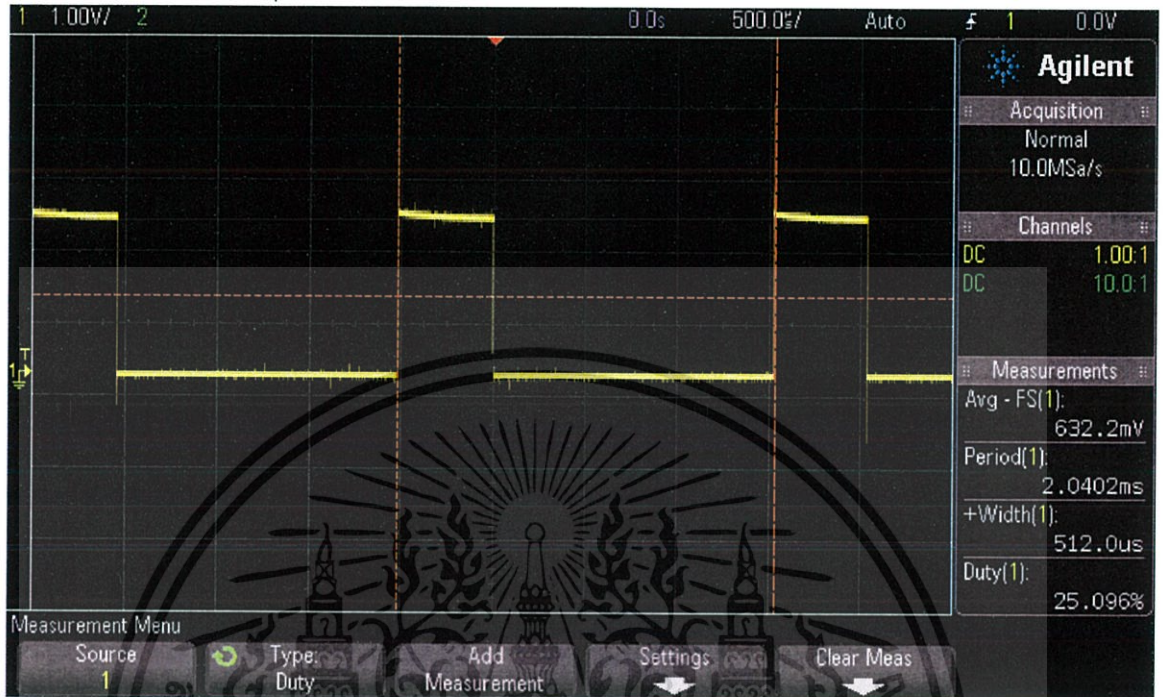
4.1 Output ที่ได้จากไมโครคอนโทรลเลอร์



รูปที่ 4. 1 เมื่อยังไม่มีการสั่งเปิดรีเลย์

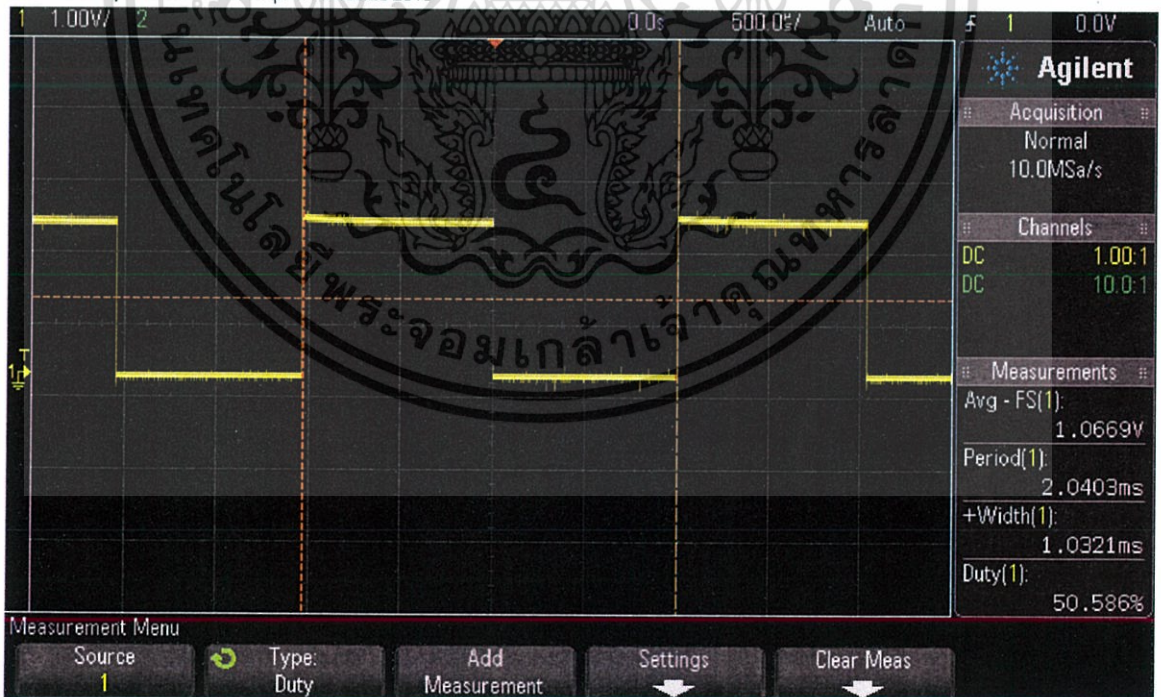
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DSO-X 2002A, MY52167134: Fri Apr 24 02:09:17 2015



รูปที่ 4. 2 เมื่อสั่งเปิดรีเลย์ duty cycle เท่ากับ 25%

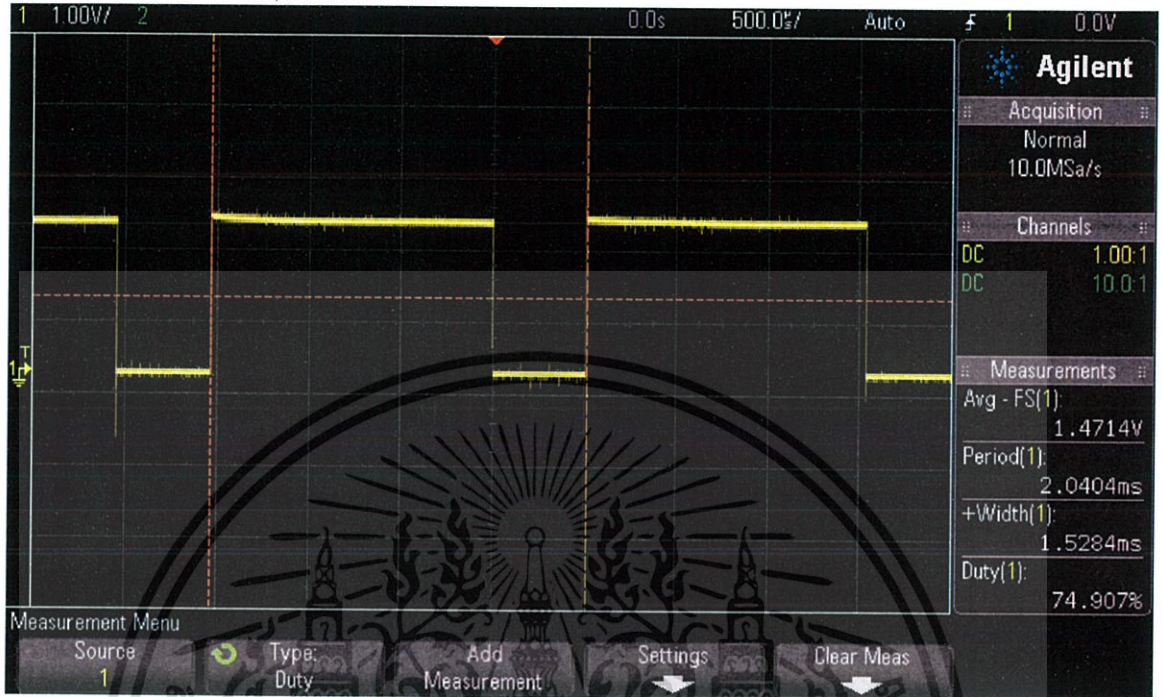
DSO-X 2002A, MY52167134: Fri Apr 24 02:09:25 2015



รูปที่ 4. 3 เมื่อสั่งเปิดรีเลย์ duty cycle เท่ากับ 50%

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DSO-X 2002A, MY52167134, Fri Apr 24 02:09:34 2015



รูปที่ 4. 4 เมื่อสั่งเปิดรีเลย์ duty cycle เท่ากับ 75%

DSO-X 2002A, MY52167134, Fri Apr 24 02:10:18 2015



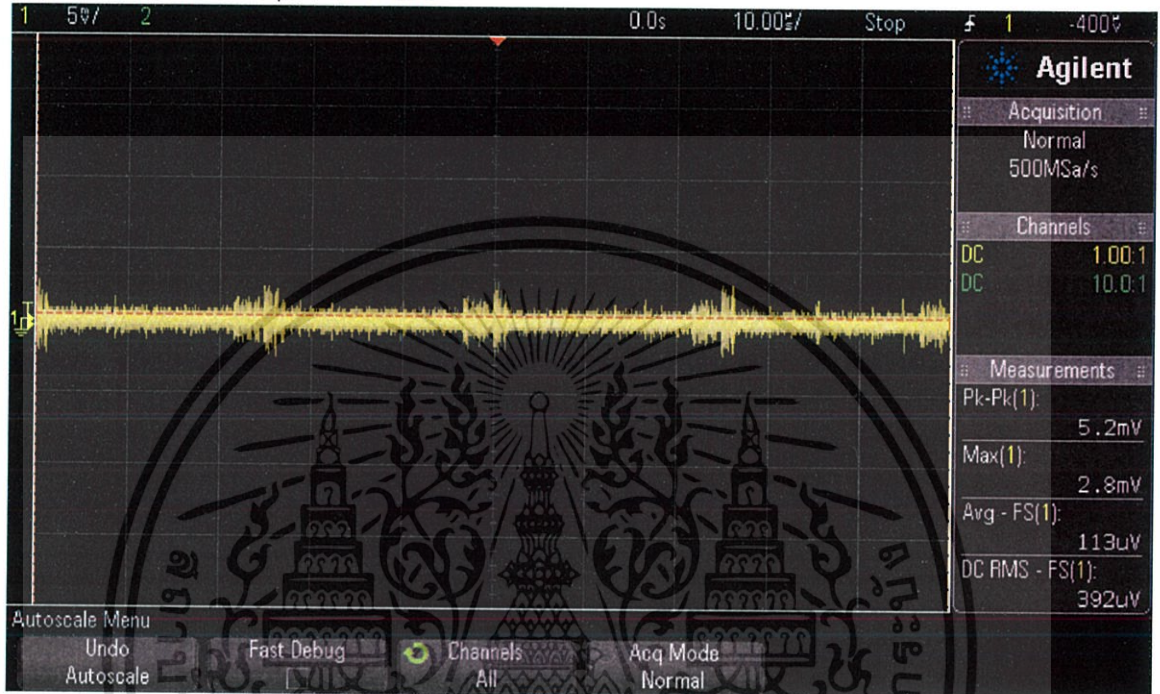
รูปที่ 4. 5 เมื่อสั่งเปิดรีเลย์ duty cycle เท่ากับ 100%

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 Outputที่ได้จากการวัดขา NC เทียบกับขา COM ของรีเลย์

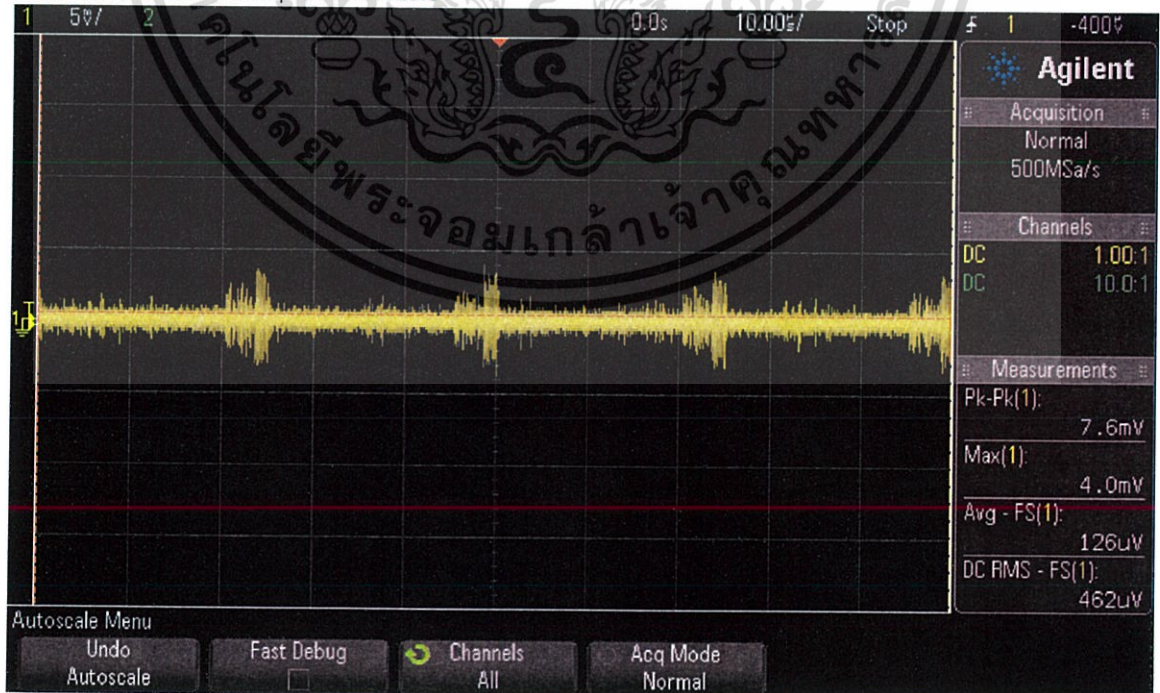
กราฟที่จะแสดงต่อไปนี้เป็น output ของวง เพื่อที่จะนำไปขับอุปกรณ์ภายนอกต่อไป

DSO-X 2002A, MY52167134: Fri Apr 24 03:40:50 2015



รูปที่ 4. 6 ขณะที่ยังไม่สั่งให้รีเลย์ทำงาน

DSO-X 2002A, MY52167134: Fri Apr 24 03:41:10 2015



รูปที่ 4. 7 ขณะที่ยังสั่งให้รีเลย์ทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

สรุปผลการทดลอง

5.1 สรุปผลการทดลอง

- จากผลการทดลองที่แสดงดังในบทที่ 4.1 ได้แสดงให้เห็นสัญญาณ output ของไมโครคอนโทรลเลอร์เพื่อที่จะให้นำสัญญาณนี้ไปเข้าที่ IC3 และ IC4 โดยที่เมื่อมีสัญญาณเข้าจะแสดงโดย LED ซึ่ง IC3 และ IC4 เป็นอุปกรณ์เชื่อมต่อทางแสง หรือที่เรียกอีกอย่างหนึ่งว่า Optocoupler เมื่อได้รับแสงหรืออนุภาคของแสง หรือที่เรียกว่า โฟตอน (Photons) ในปริมาณมากพอ จะทำให้เกิดอนุภาคอิสระที่มีประจุในบริเวณรอยต่อระหว่างเบสและคอลเลคเตอร์ (Base-Collector Region) และให้ผลเหมือนมีกระแสไหลเข้าที่ขาเบสเป็นสัญญาณ output ของ IC3 และ IC4 เพื่อนำไปเข้า IC5 ต่อไป
- เมื่อสัญญาณเข้าที่ IC5 ซึ่งทำหน้าที่ในการขับรีเลย์ก็จะทำการ drive กระแสจากสัญญาณไมโครคอนโทรลเลอร์เพื่อนำไปขับรีเลย์ต่อไป
- จากผลการทดลองที่ 4.2 จะแสดงสัญญาณ Output ที่ได้จากการวัดขา NC เทียบกับขา COM ของรีเลย์ ซึ่งออกมาในลักษณะของแรงดันเพื่อนำไปทำการควบคุมอุปกรณ์ภายนอกต่อไป

5.2 ปัญหาและแนวทางการแก้ไข

- แอปพลิเคชันที่ผู้จัดทำได้ทำการสร้างขึ้นนั้น ไม่สามารถตอบสนองกับความต้องการของผู้จัดทำได้เท่าที่ควร
- ผู้จัดทำยังมีความรู้ในเรื่องของการพัฒนาแอปพลิเคชันไม่มากพอ จึงยังไม่สามารถสร้างแอปพลิเคชันที่ใช้งานในการรับ-ส่งสัญญาณระหว่างแอปพลิเคชันกับตัวโมดูลบลูทูธ HC05 ด้วยตนเองได้
- เพื่อที่จะให้โครงงานนี้ได้ดำเนินการต่อไปได้ ผู้จัดทำจึงจำเป็นต้องนำแอปพลิเคชัน ArduDroid ซึ่งเป็นแอปพลิเคชันที่สามารถรับ-ส่งสัญญาณกับบลูทูธโมดูล HC05 ได้ ซึ่งเป็นแอปพลิเคชันที่มีคุณสมบัติที่ใกล้เคียงกับที่ต้องการ ผู้จัดทำจึงได้นำแอปพลิเคชันนี้มาใช้ในการควบคุมรีเลย์

5.3 แนวทางการพัฒนา

- สร้างแอปพลิเคชันที่ใช้ในการควบคุมรีเลย์ที่สามารถใช้งานได้
- เพิ่มจำนวนรีเลย์ที่ Output ให้มีจำนวนมากขึ้นเพื่อที่จะเป็นทางเลือกในการใช้งานที่จะนำไปใช้ในการควบคุมอุปกรณ์ภายนอกได้จำนวนมากขึ้น
- เปลี่ยนจากการควบคุมผ่านสัญญาณบลูทูธที่ระยะในการควบคุมค่อนข้างใกล้ เป็นการควบคุมด้วยสัญญาณที่มีระยะในการควบคุมที่ไกลขึ้น เพื่อที่จะได้สะดวกในการใช้งานต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

บทความ Arduino [ออนไลน์]. เข้าถึงได้จาก : <http://www.thaieasyelec.com>.

(สืบค้นวันที่ 22 เมษายน 2558).

ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหาร

ลาดกระบัง. การประยุกต์ใช้ไมโครคอนโทรลเลอร์กับวงจรอิเล็กทรอนิกส์[ออนไลน์]. เข้าถึง
ได้จาก: www.ce.kmitl.ac.th. (สืบค้นวันที่ 22 เมษายน 2558).

มหาวิทยาลัยเชียงใหม่. (2555). “การพัฒนาระบบควบคุมอุปกรณ์ไฟฟ้า ผ่านเว็บเบราว์เซอร์”

[ออนไลน์]. เข้าถึงได้จาก : <http://archive.lib.cmu.ac.th>.

(สืบค้นวันที่ 22 เมษายน 2558).

วารสาร รักผกาวงศ์. (2554). บลูทูธเทคโนโลยี [ออนไลน์]. เข้าถึงได้จาก : <http://cs.tu.ac.th>.

(สืบค้นวันที่ 22 เมษายน 2558).

วิกิพีเดีย สารานุกรมเสรี. (2557). รีเลย์ [ออนไลน์]. เข้าถึงได้จาก : <http://th.wikipedia.org>.

(สืบค้นวันที่ 22 เมษายน 2558).

อนุชา ขำขา. (2556). ระบบปฏิบัติการแอนดรอยด์ [ออนไลน์]. เข้าถึงได้จาก :

<http://potinimi.blogspot.com>. (สืบค้นวันที่ 22 เมษายน 2558).

อภิรักษ์ นามแดง. (2558). Android Relay Card [ออนไลน์]. เข้าถึงได้จาก :

<http://www.se-edmagazine.com>. (สืบค้นวันที่ 22 เมษายน 2558).

อภิเชษ กะริภูมิ. (2557). เทคนิค ความรู้ อิเล็กทรอนิกส์[ออนไลน์]. เข้าถึงได้จาก :

<http://diyelec.blogspot.com>(สืบค้นวันที่ 22 เมษายน 2558).

HC-05Bluetooth Module[ออนไลน์]. เข้าถึงได้จาก : <http://www.synes.co.th>.

(สืบค้นวันที่ 22 เมษายน 2558).

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Code ที่ใช้ในการควบคุมการทำงาน

```
#define START_CMD_CHAR '*'

#define END_CMD_CHAR '#'

#define DIV_CMD_CHAR '|'

#define CMD_DIGITALWRITE 10

#define CMD_ANALOGWRITE 11

#define CMD_TEXT 12

#define CMD_READ_ARDUROID 13

#define MAX_COMMAND 20 // max command number code. used for error checking.

#define MIN_COMMAND 10 // minimum command number code. used for error
checking.

#define IN_STRING_LENGTH 40

#define MAX_ANALOGWRITE 255

#define PIN_HIGH 3

#define PIN_LOW 2
```

```
String inText;
```

```
void setup() {
```

```
    Serial.begin(9600);
```

```
    Serial.println("ArduDroid 0.12 Alpha by TechBitar (2013)");
```

```
    Serial.flush();
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}

void loop()
{
  Serial.flush();

  int ard_command = 0;

  int pin_num = 0;

  int pin_value = 0;

  char get_char = ' '; //read serial

  // wait for incoming data
  if (Serial.available() < 1) return; // if serial empty, return to loop().

  // parse incoming command start flag
  get_char = Serial.read();

  if (get_char != START_CMD_CHAR) return; // if no command start flag, return to
loop().

  // parse incoming command type

  ard_command = Serial.parseInt(); // read the command

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

// parse incoming pin# and value

pin_num = Serial.parseInt(); // read the pin

pin_value = Serial.parseInt(); // read the value

// 1) GET TEXT COMMAND FROM ARDUDROID

if (ard_command == CMD_TEXT){

  inText = ""; //clears variable for new input

  while (Serial.available()) {

    char c = Serial.read(); //gets one byte from serial buffer

    delay(5);

    if (c == END_CMD_CHAR) { // if we the complete string has been read

      // add your code here

      break;

    }

    else {

      if (c != DIV_CMD_CHAR) {

        inText += c;

        delay(5);

      }

    }

  }

}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}

// 2) GET digitalWrite DATA FROM ARDUDROID

if (ard_command == CMD_DIGITALWRITE){

    if (pin_value == PIN_LOW) pin_value = LOW;

    else if (pin_value == PIN_HIGH) pin_value = HIGH;

    else return; // error in pin value. return.

    set_digitalwrite( pin_num, pin_value); // Uncomment this function if you wish to
use
return; //return from start of loop()
}

// 3) GET analogWrite DATA FROM ARDUDROID

if (ard_command == CMD_ANALOGWRITE) {

    analogWrite( pin_num, pin_value );

    // add your code here

    return; // Done. return to loop();

}

```

```

// 4) SEND DATA TO ARDUDROID

```

```

if (ard_command == CMD_READ_ARDUDROID) {

```

```

    // char send_to_android[] = "Place your text here." ;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

// Serial.println(send_to_android); // Example: Sending text

Serial.print(" Analog 0 = ");

Serial.println(analogRead(A0)); // Example: Read and send Analog pin value to
Arduino

return; // Done. return to loop();

}

}

// 2a) select the requested pin# for DigitalWrite action
void set_digitalwrite(int pin_num, int pin_value)
{
switch (pin_num) {
case 13:

pinMode(13, OUTPUT);

digitalWrite(13, pin_value);

// add your code here

break;

case 12:

pinMode(12, OUTPUT);

digitalWrite(12, pin_value);

// add your code here

break;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

case 11:

```
pinMode(11, OUTPUT);  
  
digitalWrite(11, pin_value);  
  
// add your code here  
  
break;
```

case 10:

```
pinMode(10, OUTPUT);  
  
digitalWrite(10, pin_value);  
  
// add your code here  
  
break;
```

case 9:

```
pinMode(9, OUTPUT);  
  
digitalWrite(9, pin_value);  
  
// add your code here  
  
break;
```

case 8:

```
pinMode(8, OUTPUT);  
  
digitalWrite(8, pin_value);  
  
// add your code here  
  
break;
```

case 7:

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
pinMode(7, OUTPUT);  
  
digitalWrite(7, pin_value);  
  
// add your code here  
  
break;
```

case 6:

```
pinMode(6, OUTPUT);  
  
digitalWrite(6, pin_value);  
  
// add your code here  
  
break;
```

case 5:

```
pinMode(5, OUTPUT);  
  
digitalWrite(5, pin_value);  
  
// add your code here  
  
break;
```

case 4:

```
pinMode(4, OUTPUT);  
  
digitalWrite(4, pin_value);  
  
// add your code here  
  
break;
```

case 3:

```
pinMode(3, OUTPUT);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
digitalWrite(3, pin_value);

// add your code here

break;

case 2:

pinMode(2, OUTPUT);

digitalWrite(2, pin_value);

// add your code here

break;

// default:

// if nothing else matches, do the default

// default is optional

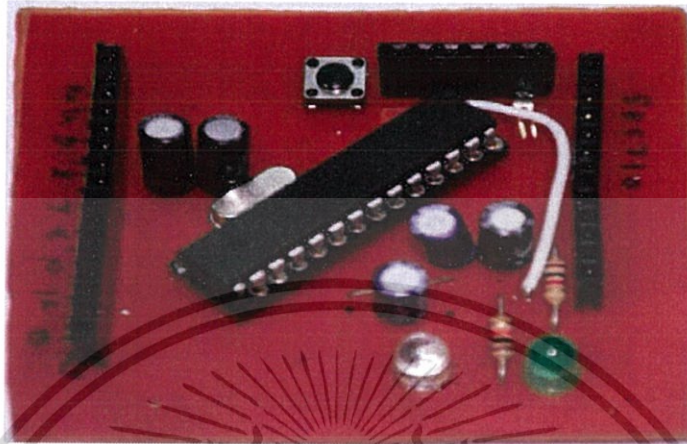
}

}
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Hardware ที่ได้ทำการออกแบบ

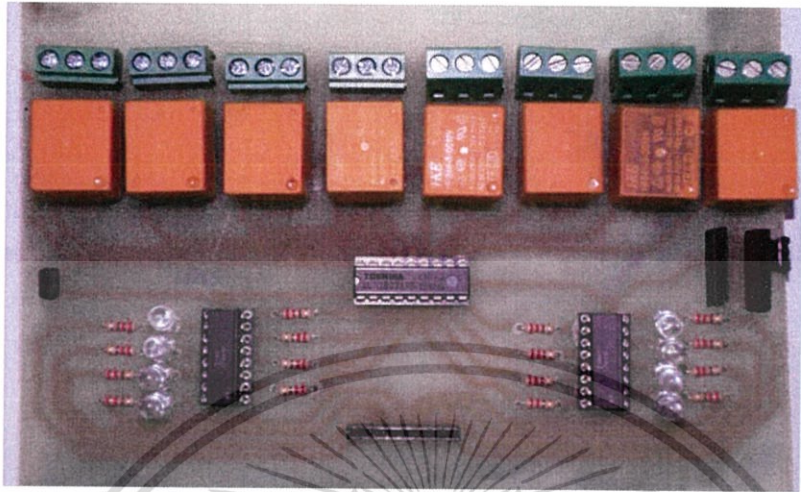


บอร์ดไมโครคอนโทรลเลอร์ AVR

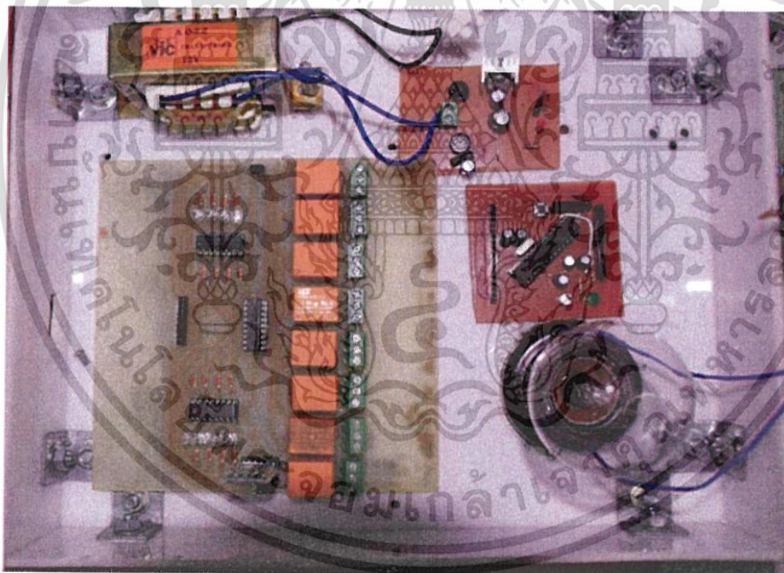


บอร์ดแปลงไฟฟ้ากระแสสลับเป็นไฟฟ้ากระแสตรง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

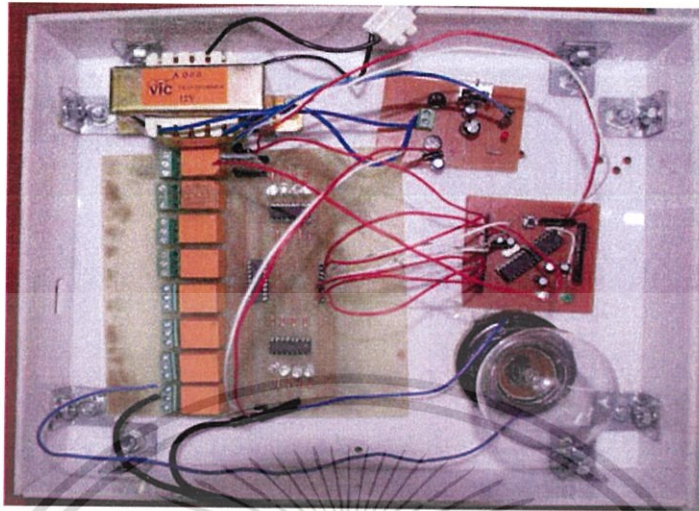


บอร์ดที่ใช้ในการขับรีเลย์



รวมฮาร์ดแวร์ที่ใช้ในโครงการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เชื่อมต่อฮาร์ดแวร์ทั้งหมดเข้าด้วยกัน



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้