รายงานสหกิจศึกษาฉบับสมบูรณ์

การสั่งงานบอร์ดสวิตช์ด้วยเว็บ remotemux

นางสาวณิชชารีย์ ศรีลาวัณย์

ภาควิชาอิเล็กทรอนิกส์
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2560

รายงานสหกิจศึกษาฉบับสมบูรณ์

การสั่งงานบอร์ดสวิตซ์ด้วยเว็บ
remotemux

นางสาว ณิชชารีย์ ศรีลาวัณย์

ภาควิชาอิเล็กทรอนิกส์
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2560

หัวข้อโครงงาน	การสั่งงานบอร์ดสวิตซ์ด้วยเว็บ
นักศึกษา	นางสาว ณิชชารีย์ ศรีลาวัณย์ 57010474
ปริญญา	วิศวกรรมศาสตรบัณฑิต
ภาควิชา	วิศวกรรมอิเล็กทรอนิกส์
ปีการศึกษา	2560
อาจารย์ที่ปรึกษาโครงการ	อาจารย์ชินภัทร นันทจิวากรชัย,อาจารย์เกรียงไกรสุขสุด

บทคัดย่อ

การศึกษาโครงงานสหกิจศึกษาที่บริษัท แม็กซิมอินทริเกรดเต็ด โปรดักส์ (ประเทศไทย) จำกัด ในครั้งนี้มีวัตถุประสงค์เพื่อเรียนรู้กระบวนการทำงานของวิศวกรทดสอบ (Test Development Engineer) ซึ่งทำหน้าที่ในการออกแบบระบบ(Test system) ที่ใช้ในการทดสอบวงจรรวมเพื่อให้แน่ใจว่าได้ส่งวงจรรวมที่ดีมีคุณภาพไปให้แก่ลูกค้าและได้ออกแบบบอร์ดรีเลย์ในการสับสวิตซ์เพื่อวัดแรงดันเอาท์พุทวัดได้ 32 pin ,4 channel โดยการสั่งงานผ่านเว็บสร้างขึ้น

Project Title	Remotemux
Student	Ms. Nitcharee Srilawan Student ID 57010474
Degree	Bachelor of Engineering
Program	Electronics Engineering
Year	2017
Project Advisor	Dr.Chinnapat Nantajiwakornchai,Dr.Kriangkrai Sooksood

ABSTRACT

The purpose of this COOP training project, at Maxim Integrated Products Thailand,is to learn about working processof Test Development Engineer which an engineer who design a test system to determine whether products are good or not for deliver only good products to customerand designed relay switch board to measure the output voltage measured 32 pins, 4 channels by the web.

กิตติกรรมประกาศ

การทำโครงงานในครั้งนี้สามารถสำเร็จลุล่วงไปได้ด้วยดี ด้วยความช่วยเหลือของนายโกวิท พิมพ์ทอง ซึ่งทำหน้าที่เป็นผู้นิเทศงาน และพนักงานท่านอื่นๆ ในบริษัทแม็กซิมอินทริเกรทเต็ดโปรดักส์ (ประเทศไทย) จำกัด ซึ่งได้ให้คำแนะนำและข้อคิดเห็นต่าง ๆ อันเป็นประโยชน์อย่างยิ่งในการออกแบบระบบทดสอบชิ้นงาน อีกทั้งยังช่วยแก้ปัญหาต่าง ๆ ที่เกิดขึ้นระหว่างการดำเนินงานจนทำให้ระบบทดสอบชิ้นงานเสร็จสมบูรณ์ ขอขอบคุณพี่ๆ พนักงานในบริษัทแม็กซิมอินทริเกรทเต็ดโปรดักส์ (ประเทศไทย) จำกัด และคณะอาจารย์ในสถาบันฯ ที่ได้เปิดโอกาส และให้ข้อแนะนำและความช่วยเหลือในทุก ๆ ด้านตลอดการทำโครงงานในครั้งนี้ นอกจากนี้ขอขอบคุณเพื่อนๆทุกคนที่เป็นกำลังใจ และให้ความช่วยเหลือในการโครงงานในครั้งนี้

สุดท้ายนี้ ผู้วิจัยขอขอบคุณบิดามารดา และครอบครัว ซึ่งเปิดโอกาสให้ได้รับการศึกษาเล่าเรียน ตลอดจนคอยช่วยเหลือและให้กำลังใจเสมอมาจนสามารถสำเร็จโครงงานได้

ณิชชารีย์ ศรีลาวัณย์

สารบัญ

บทคัดย่อ.....I

ABSTRACT.....II

กิตติกรรมประกาศ.....III

สารบัญ.....IV

บทที่ 1 บทนำ.....1

1.1 ที่มาและความสำคัญของโครงงาน.....1

1.2 วัตถุประสงค์ของโครงงาน.....1

1.3 สมมติฐานของการศึกษา.....2

1.4 ขอบเขตของโครงงาน.....2

1.5 ประโยชน์ที่คาดว่าจะได้รับ.....2

บทที่ 2 ทฤษฎีและหลักการ.....3

2.1 หลักการ.....3

2.1.1 ส่วนประกอบพื้นฐานของวงจรบอร์ด remotemux.....3-4

2.1.2หลักการพื้นฐานของบอร์ด remotemux

.....**Error! Bookmark not defined.**

2.2 ทฤษฎี.....6

2.2.1 แหล่งจ่ายไฟ (Power supply) 6

2.2.2 แบตเตอรี่ (battery).....7

2.2.3 ราสเบอรี่พาย (raspberry pi).....8-9

2.2.4 WEBUI.....10-26

2.2.5 รีเลย์ (Relay driver).....27-34

บทที่ 3 วิธีการดำเนินการ.....35

3.1 การออกแบบหน้าเว็บสำหรับการควบคุมบอร์ด remotemux.....35

3.2แผนผังการเขียน web ของ บอร์ด remotemuxลงใน raspberry pi 36

3.3 วิธีการลง linux.....36

3.3.1 ติดตั้งระบบปฏิบัติการ Linux ให้กับ Raspberry Pi.....36

3.3.2 เตรียมSoftwareสำหรับติดตั้งระบบปฏิบัติการLinuxลงบนบอร์ดRaspberry Pi.....37-38

3.3.3 ขั้นตอนการติดตั้งระบบปฏิบัติการ Raspbian ให้กับบอร์ด Raspberry Pi.....39

3.4 สรุปวิธีการลง linux 40-42

3.5วิธีการติดตั้ง webmin.....43

3.6 วิธีติดตั้ง LAMP(linux,apache,php,mysql).....44

3.7 วิธี install phpmyadmin raspberry pi.....44

3.8 วิธีทำให้ LED ที่บอร์ด remotemux ติด.....45

3.9 วิธี ลง i2C.....46

บทที่ 4 การทดลองและผลการทดลอง.....47

4.1 การทดลองและผลการทดลอง การสั่งงานด้วยเว็บของ บอร์ด remotemux 47

4.1.1 ให้แต่ละ channel เปิดได้ทีละ CH 47

4.1.2 มีข้อความในช่อง text ขึ้นมาอัตโนมัติตามคำที่เราตั้งไว้ 47

4.1.3 สามารถเพิ่มข้อความใน text ไปเก็บใน sql database ได้.....47

4.1.4 กดปุ่ม reset แล้วสวิตซ์ปิด.....48

4.1.5 สามารถวัดแรงดันที่เอาท์พุทได้โดยกดปุ่ม ADC ON /ADC OFF 48

4.1.6 ส่วนของเวลาการใช้งานของหน้าเว็บ.....48

4.2 คุณสมบัติของบอร์ด remotemux 49-53

บทที่ 5 สรุปผลการวิจัยและข้อเสนอแนะ.....54

เอกสารอ้างอิง.....55

ภาคผนวก 56-73

สารบัญรูป

รูปที่ **หน้า**

รูปที่ 2.1 แผนผังการใช้บอร์ดremotemux.....3

รูปที่ 2.2 แสดงบล๊อคไดอะแกรมของ บอร์ด remotemux.....5

รูปที่ 2.3 แบตเตอรี่อัลคาไลน์.....7

รูปที่ 2.4raspberry pi zero w.....8

รูปที่ 2.5 พอร์ต GPIO ของ raspberry pi zero w 10

รูปที่ 2.6 โครงสร้างภาษา HTML.....17

รูปที่ 2.7 ผลลัพธ์การทำงานในการรันโปรแกรม first.py.....22

รูปที่ 2.8 รูปตัวอย่างรีเลย์และสัญลักษณ์รีเลย์.....28

รูปที่ 2.9การทำงานของส่วนหน้าสัมผัส.....29

รูปที่ 2.10รูปบล๊อกไดอะแกรมของมัลติเพล็กซ์เซอร์.....34

รูปที่ 2.11รูปแสดงการมัลติเพล็กซ์แบบแบ่งเวลา.....34

รูปที่ 3.2 Sandisk Micro SD Ultra 8GB 30MB/s Class10 with Adapter.....38

รูปที่ 3.3ตัวอย่างหน้าต่างโปรแกรม SD Formatter Version 4.0.....39

รูปที่ 3.4ตัวอย่างโปรแกรม Win32 Disk Imager.....39

รูปที่ 3.5รันไฟล์ Win32DiskImager หน้าต่าง 40

รูปที่ 3.6 หลังจากรันโปรแกรมจะปรากฏโปรแกร.ม.....40

รูปที่ 3.7 write ไฟล์ลงใน SD CARD 41

รูปที่ 3.8 รูปหลังจาก boot เสร็จเรียบร้อย 41

รูปที่ 3.9 สำหรับ Enable Boot to Desktop/Scratch.....42

รูปที่ 3.10 เพื่อให้สามารถ log in ที่ desktop ได้.....42

รูปที่ 3.11 เปิดใช้งานโมดูลกล้อง.....43

รูปที่ **หน้า**

รูปที่ 4.1 เปิดCHได้ทีละสวิตช์เท่านั้น.....47

รูปที่ 4.2 ข้อความใน text ขึ้นตามตัวอักษร.....47

รูปที่ 4.3 เก็บข้อความไว้ใน sql database 48

รูปที่ 4.4 ช่องดูค่าแรงดันเอาท์พุท ADC.....48

รูปที่ 4.5 ส่วนของเวลาการใช้งานเว็บ.....48

รูปที่ 4.6 วัดแรงดันที่เอาท์พุทที่ CH1 ของทั้ง 4 channels และหน้าเว็บ.....49

รูปที่ 4.7 วัดแรงดันที่เอาท์พุทที่ CH2 ของทั้ง 4 channels และหน้าเว็บ.....50

รูปที่ 4.8 วัดแรงดันที่เอาท์พุทที่ CH3 ของทั้ง 4 channels และหน้าเว็บ.....50

รูปที่ 4.9 วัดแรงดันที่เอาท์พุทที่ CH4ของทั้ง 4 channels และหน้าเว็บ.....51

รูปที่ 4.10วัดแรงดันที่เอาท์พุทที่ CH5 ของทั้ง 4 channels และหน้าเว็บ.....51

รูปที่ 4.11 วัดแรงดันที่เอาท์พุทที่ CH6ของทั้ง 4 channels และหน้าเว็บ.....52

รูปที่ 4.12 วัดแรงดันที่เอาท์พุทที่ CH7 ของทั้ง 4 channels และหน้าเว็บ.....52

รูปที่ 4.13 วัดแรงดันที่เอาท์พุทที่ CH8ของทั้ง 4 channels และหน้าเว็บ.....53

บทที่ 1

บทนำ

1.1 ที่มาและความสำคัญของโครงงาน

ในปัจจุบันมนุษย์นั้นมักจะคิดเทคโนโลยีใหม่ๆออกมาตลอดเวลา เพื่อความทันสมัย สะดวกสบายยิ่งขึ้น นวัตกรรมเหล่านี้ส่งผลให้วิถีการใช้ชีวิตของมนุษย์เปลี่ยนแปลงไปจากเดิม และความต้องการนวัตกรรมที่มีความทันสมัยมีแนวโน้มที่จะเพิ่มมากขึ้นตามความต้องการของมนุษย์ที่เพิ่มขึ้น

ดังนั้นเราจึงได้คิดออกแบบวงจรบอร์ดรีเลย์ในการสับสวิตซ์มาเพื่อความสะดวกและความรวดเร็วของงานที่ต้องการวัดแรงดันเอาท์พุทของชิ้นงานที่ต้องการวัดได้ เพราะเราสามารถสั่งงานได้ผ่านเว็บที่สร้างขึ้นโดยวัดขาชิ้นงานได้มากสุด 32 ขาโดยแบ่งเป็น 4 channels วัดได้ทีละขา

1.2 วัตถุประสงค์ของโครงงาน

1.2.1 ฝึกการทำงานภาคทฤษฎีและปฏิบัติของภาควิชาอิเล็กทรอนิกส์

1.2.2 เพื่อศึกษาการทดลองออกแบบวงจรและทำเว็บไซต์

1.2.3 เพื่อฝึกทักษะในกระบวนการคิด วิเคราะห์ และแก้ไขปัญหาที่เกิดขึ้น

1.2.4เพื่อเป็นการนำความรู้ที่ได้ศึกษามาประยุกต์ใช้งานจริงในสายอาชีพ

1.2.5เพื่อเป็นการเพิ่มความรู้ และประสบการณ์ในสายงานทางด้านวิศวกรรมอิเล็กทรอนิกส์

1.3 สมมติฐานของการศึกษา

สามารถเข้าใจถึงวิธีการออกแบบวงจรอิเล็กทรอนิกส์โดยผ่านการออกแบบ เข้าใจหลักการทำงานของวงจรบอร์ด remotemux และเข้าใจถึงการทำเว็บไซต์โดยใช้ภาษาต่างๆ เช่น python,php,java,html สามารถนำความรู้ที่ได้จากการเรียนในห้องเรียนโดยเฉพาะวิชาElectronic Engineering มาใช้ในการลงมือปฏิบัติจริง และสามารถใช้งานบอร์ด remotemux ได้จริง

1.4 ขอบเขตของโครงงาน

1.4.1 มีสัญญาณ **input 32** สัญญาณ สามารถเชื่อมต่อกับสัญญาณ **input** ของ **oscilloscope** ได้

1.4.2 สามารถเชื่อมต่อ **Web user interface** ได้

1.4.3 การทำงานของแบตเตอรี่ขนาด **5000 mAh** สามารถใช้งานได้เกิน **10** ชั่วโมง

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1.5.1 เข้าใจกระบวนการทำงานของวงจรในแต่ละส่วน

1.5.2 เข้าใจหลักการของการเขียนเว็บในแต่ละภาษา

1.5.3 เข้าใจถึงผลกระทบที่เกิดจากโค้ด และสามารถแก้ไขปัญหาได้

1.5.4 สามารถออกแบบเว็บสำหรับการใช้งานให้เกิดประโยชน์สูงสุด

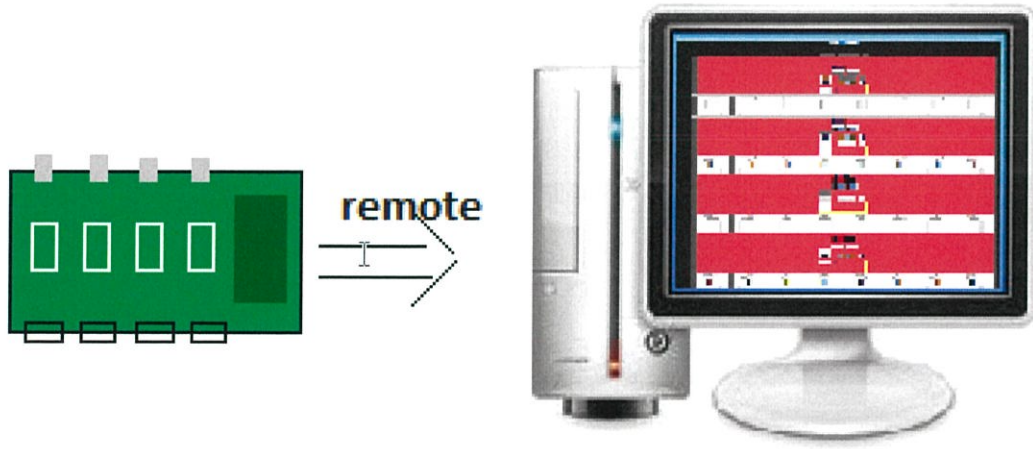
1.5.5 สามารถที่จะนำความรู้และประสบการณ์ที่ได้รับไปประยุกต์ใช้กับการทำงานในอนาคตได้

บทที่ 2

ทฤษฎีและหลักการ

2.1 หลักการ

สำหรับองค์ประกอบของระบบทดสอบวงจรรวมสามารถเขียนเป็นแผนผังได้ดังนี้



รูปที่ 2.1แผนผังการใช้งานบอร์ด remotemux

2.1.1 ส่วนประกอบพื้นฐานของวงจรบอร์ด remotemux

1.วงจร GPIO CONNECTOR

เป็น GPIO ของบอร์ด raspberry pi คือเราสามารถ คอนโทรลให้เป็น “1” หรือ “0” ได้ตามใจชอบ โดยที่จะสามารถควบคุมได้แต่ละ pin เหมือนไมโครคอนโทรลเลอร์

2. วงจร MUX MODULE

มัลติเพล็กซ์เซอร์ (Multiplexer:Mux) หรือเรียกอีกชื่อหนึ่งว่า ตัวเลือกข้อมูล (Data selector) ซึ่งเป็นตัวที่ทำหน้าที่เลือกช่องสัญญาณที่มีข้อมูลช่องหนึ่งจากหลาย ๆช่องสัญญาณมาเป็นอินพุทและต่อช่องสัญญาณที่มีข้อมูลนั้นเข้าเป็ฯสัญญาณเอาท์พุทเพียงเอาท์พุทเดียว

3.วงจรรักษาแรงดันคงที่ (Voltage Regulator)

วงจรที่จะต่ออยู่ระหว่างแหล่งจ่ายไฟกระแสตรงกับโหลด ทำหน้าที่ในการจ่ายไฟตรงให้กับโหลดและสามารถรักษาระดับแรงดันให้คงที่ได้ หรือเราสามารถนำตัววงจรรักษาระดับแรงดันนี้มาใช้ต่อให้กับวงจรเราได้นั่นเอง

4.วงจร Power Protection

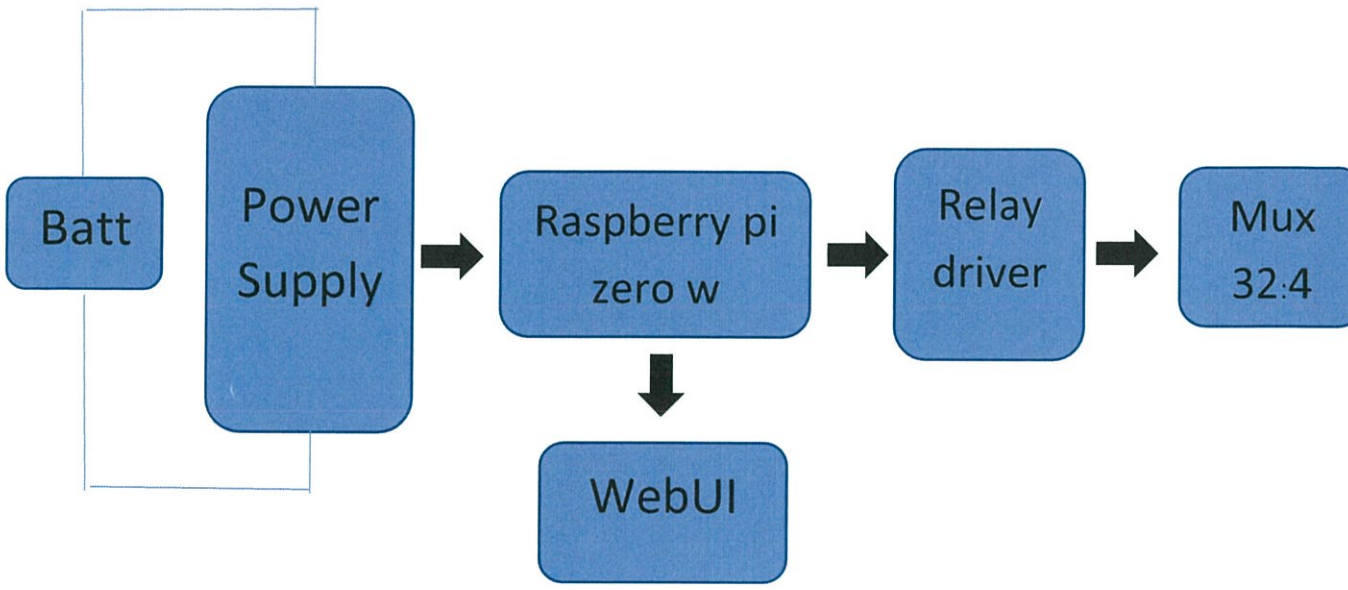
เป็นวงจรที่เพิ่มเติมขึ้นมาเพื่อป้องกันไม่ให้แหล่งจ่ายไฟฟ้าเสียหายอันเนื่องมาจากหการลัดวงจร (ShortCircuit) หรือกระแสเกิน (Overload) ที่เอาท์พุท ซึ่งอาจจะทำให้อุปกรณ์บางตัวอาจจะเสียหายได้

5.ราสเบอรี่พาย ซีโร่ (Raspberry pi zero w)

เป็นคอมพิวเตอร์ขนาดเล็ก ไว้สำหรับการเขียนโปรแกรมติดต่อกับอินพุต เอาท์พุต (GPIO) และการสื่อสารผ่านพอร์ตอนุกรม (Serial Port) ทั้งแบบ I2C และ SPI ในตัวมี wireless lan และ Bluetooth

2.1.2 หลักการพื้นฐานของบอร์ด remotemux

บอร์ด remotemux สามารถแบ่งออกเป็นส่วนๆได้ดังแผนภาพ



รูปที่ 2.2 แสดงบล็อคไดอะแกรมของ บอร์ด remotemux

-ส่วนที่ 1แรงดันไฟตรงที่เข้ามา ใช้แรงดันจ่ายไฟมา 5 v สำหรับจ่ายไฟให้ raspberry pi

-ส่วนที่ 2 raspberry pi บอร์ดคอมพิวเตอร์ขนาดเล็กที่สามารถเชื่อมต่อกับจอมอนิเตอร์ คีย์บอร์ด และเมาส์ได้ สามารถนำมาประยุกต์ใช้ในการทำโครงงานทางด้านอิเล็กทรอนิกส์ การเขียนโปรแกรม หรือเป็นเครื่องคอมพิวเตอร์ตั้งโต๊ะขนาดเล็ก ไม่ว่าจะเป็นการทำงาน Spreadsheet Word Processing ท่องอินเทอร์เน็ต ส่งอีเมล หรือเล่นเกมส์ อีกทั้งยังสามารถเล่นไฟล์วีดีโอความละเอียดสูง (High-Definition) ได้อีกด้วย

-ส่วนที่ 3 WebUI ทำหน้าที่ในการสั่งงานให้บอร์ด remotemux ว่าจะให้ relay ตัวไหน on/off สำหรับการสับสวิตช์ของในแต่ละขา โดยสั่งงานได้ผ่านเว็บที่เราสร้างขึ้นมา

-**ส่วนที่ 4** Relay driver เป็นส่วนของ relay สำหรับการสับสวิตซ์ในการวัดแรงดันที่เอาท์พุทของงานที่เราต้องการทราบค่า

-**ส่วนที่** 5 มัลติเพล็กซ์เซอร์ (Multiplexer : MUX) หรือเรียกอีกชื่อหนึ่งว่า ตัวเลือกข้อมูล(Data Selector) ซึ่งเป็นตัวที่ทำหน้าที่เลือกช่องสัญญาณที่มีข้อมูลช่องหนึ่งจาก 32 ช่องสัญญาณมาเป็นอินพุตและต่อช่องสัญญาณที่มีข้อมูลนั้นเข้าเป็นสัญญาณเอาต์พุตเพียง 4 ช่อง

2.2 ทฤษฎี

2.2.1 แหล่งจ่ายไฟ (Power supply)

เป็นอุปกรณ์ที่จ่ายพลังงานไฟฟ้าให้กับโหลดไฟฟ้า. เป็นคำที่ใช้กันมากที่สุด ในการแปลงพลังงานไฟฟ้าจากรูปแบบหนึ่ง ไปเป็นอีกรูปแบบหนึ่ง แม้ว่ามันจะยังอาจหมายถึง อุปกรณ์ที่แปลงพลังงานรูปแบบหนึ่ง (เช่นพลังงานกล, พลังงานเคมี, พลังงานแสงอาทิตย์) ให้เป็นพลังงานไฟฟ้า. แหล่งจ่ายไฟแบบควบคุมได้(อังกฤษ: regulated power supply)สามารถควบคุม แรงดันหรือกระแสเอาต์พุตให้มีค่าที่คงที่แน่นอน แม้ว่าโหลดจะมีการเปลี่ยนแปลงหรือมีการเปลี่ยนแปลงที่พลังงานที่อินพุทก็ตาม

แหล่งจ่ายไฟทุกตัวต้องได้รับพลังงานจากแหล่งพลังงานภายนอกเพื่อจ่ายให้โหลดและการบริโภคพลังงานของตัวมันเองในขณะที่ปฏิบัติงาน แหล่งพลังงานภายนอกจะขึ้นอยู่กับการออกแบบ. แหล่งจ่ายไฟอาจจะได้รับพลังงาน จาก:

- ระบบสายส่งพลังงานไฟฟ้า อาจเป็นกระแสสลับ หรือกระแสตรงที่ได้จากอุปกรณ์ไฟฟ้าที่แปลงแรงดันไฟฟ้า กระแสสลับให้เป็นกระแสตรง
- อุปกรณ์จัดเก็บพลังงาน เช่นแบตเตอรี่ และ เซลล์เชื้อเพลิง
- ระบบเครื่องกลไฟฟ้า เช่นเครื่องกำเนิดไฟฟ้าจากพลังงานกล เช่น Generator และ Alternator
- พลังงานแสงอาทิตย์

แหล่งจ่ายไฟอาจถูกนำมาใช้แบบแยกส่วน หรือเป็นอุปกรณ์ส่วนหนึ่งของโหลด เช่น แหล่งจ่ายไฟในคอมพิวเตอร์เดสก์ทอป และอุปกรณ์อิเล็กทรอนิกส์ทั่วไปสำหรับผู้บริโภค

แหล่งจ่ายไฟอาจถูกนำมาใช้แบบแยกส่วน หรือเป็นอุปกรณ์ส่วนหนึ่งของโหลด เช่นแหล่งจ่ายไฟในคอมพิวเตอร์เดสก์ทอป และอุปกรณ์อิเล็กทรอนิกส์ทั่วไปสำหรับผู้บริโภค

คุณลักษณะเฉพาะที่ระบุไว้บนแหล่งจ่ายไฟ ได้แก่:

- ปริมาณของแรงดันและกระแสที่จะสามารถจ่ายให้กับโหลดได้
- วิธีการที่จะทำให้แรงดันหรือกระแสเอาต์พุตมีเสถียรภาพ ภายใต้เงื่อนไขที่กระแสไฟฟ้าอินพุทและสภาวะของโหลดที่เปลี่ยนแปลง
- ระยะเวลาการใช้งานได้นานเท่าใดโดยไม่ต้องเติมเชื้อเพลิงหรือหรือชาร์จประจุใหม่ (เฉพาะอุปกรณ์ไฟฟ้าที่ใช้แหล่งพลังงานแบบพกพา)

2.2.2 **แบตเตอรี่** (power bank)

รูปที่ 2.3แบตเตอรี่อัลคาไลน์

แบตเตอรี่เป็นอุปกรณ์ที่แปลงพลังงานเคมีที่เก็บไว้ให้เป็นพลังงานไฟฟ้า แบตเตอรี่ถูกใช้เป็น แหล่งพลังงานในครัวเรือนจำนวนมากและประยุกต์ใช้ในอุตสาหกรรม

แบตเตอรี่มีสองชนิด ได้แก่แบตเตอรรี่ปฐมภูมิ (แบตเตอรี่ที่ใช้แล้วทิ้ง) ซึ่งถูกออกแบบมาให้ ใช้เพียงครั้งเดียวแล้วทิ้ง และแบตเตอรี่ทุติยภูมิ (แบตเตอรี่ชาร์จใหม่ได้) ซึ่งถูกออกแบบมาเพื่อ ชาร์จใหม่ได้หลายครั้ง แบตเตอรี่มีหลายขนาดจากเซลล์ขนาดเล็กที่ใช้ในเครื่องช่วยฟังและนาฬิกาข้อมือ ไปจนถึงขนาดห้องเป็นกลุ่มของแบตเตอรี่ (อังกฤษ: battery bank) ที่ทำหน้าที่เป็นเครื่องกำเนิดไฟฟ้าสำรอง ในชุมสายโทรศัพท์และศูนย์ข้อมูลคอมพิวเตอร์

2.2 raspberry pi

บทความการพัฒนาโปรแกรมบน Raspberry Pi ด้วย Qt

บทความนี้จะแนะนำให้รู้จักกับ Raspberry Pi บอร์ด Embedded OS ที่กำลังได้รับความนิยมอย่างมาก ตั้งแต่ส่วนประกอบจนถึงการติดตั้งระบบปฏิบัติการ Linux เพื่อเริ่มต้นเรียนรู้และใช้งานกับบอร์ด จากนั้นกล่าวถึงการติดตั้งและใช้งาน Qt ซึ่งเป็นเฟรมเวิร์คสำหรับการพัฒนาแอพพลิเคชั่นติดต่อผู้ใช้งาน และเชื่อมต่อกับฮาร์ดแวร์ของบอร์ดผ่านไลบรารี่ Wiring Pi พร้อมตัวอย่างการเขียนโปรแกรมติดต่อกับอินพุต-เอาท์พุต (GPIO) และการสื่อสารผ่านพอร์ตอนุกรม (Serial Port) ทั้งแบบ UART I2C และ SPI

2.2.3 ราสเบอรี่พาย (raspberry pi)

Raspberry Pi คืออะไร?

บอร์ดคอมพิวเตอร์ขนาดเล็กที่สามารถเชื่อมต่อกับจอมอนิเตอร์ คีย์บอร์ด และเมาส์ได้ สามารถนำมาประยุกต์ใช้ในการทำโครงงานทางด้านอิเล็กทรอนิกส์ การเขียนโปรแกรม หรือเป็นเครื่องคอมพิวเตอร์ตั้งโต๊ะขนาดเล็ก ไม่ว่าจะเป็นการทำงาน Spreadsheet Word Processing ท่องอินเทอร์เน็ต ส่งอีเมล หรือเล่นเกมส์ อีกทั้งยังสามารถเล่นไฟล์วีดีโอความละเอียดสูง (High-Definition) ได้อีกด้วย

บอร์ด Raspberry Pi รองรับระบบปฏิบัติการลินุกซ์ (Linux Operating System) ได้หลายระบบ เช่น Raspbian (Debian) Pidora (Fedora) และ Arch Linux เป็นต้น โดยติดตั้งบน SD Card บอร์ด Raspberry Pi นี้ถูกออกแบบมาให้มี CPU GPU และ RAM อยู่ภายในชิปเดียวกัน มีจุดเชื่อมต่อ GPIO ให้ผู้ใช้สามารถนำไปใช้ร่วมกับอุปกรณ์อิเล็กทรอนิกส์อื่นๆ ได้อีกด้วย

รูปที่ 2.4 raspberry pi zero w

คุณสมบัติทางเทคนิคของบอร์ด raspberry pi zero w

บอร์ด Raspberry Pi Zero W เป็นบอร์ด Zero รุ่นที่มี Wireless LAN และ Bluetooth 4.1 BLE เพิ่มเติมขึ้นมา ใช้ชิพ Broadcom BCM2835 ARM11 ตัวเดียวกับ Raspberry Pi Model B/A+/B+ รันที่ความเร็ว 1 GHz (ปกติรันที่ 700 MHz) มี RAM LPDDR2 ขนาด 512 MB ช่องใส่ micro SD card พอร์ต mini HDMI พอร์ต micro USB สำหรับจ่ายไฟและ micro USB OTG / Host สำหรับต่ออุปกรณ์ พร้อมขา GPIO 40 ขาของ Raspberry Pi และพอร์ต CSI Camera Interface สำหรับต่อกล้อง Raspberry Pi Camera (เป็นพอร์ตขนาดเล็กต้องใช้สายแปลง)

ประกอบด้วย

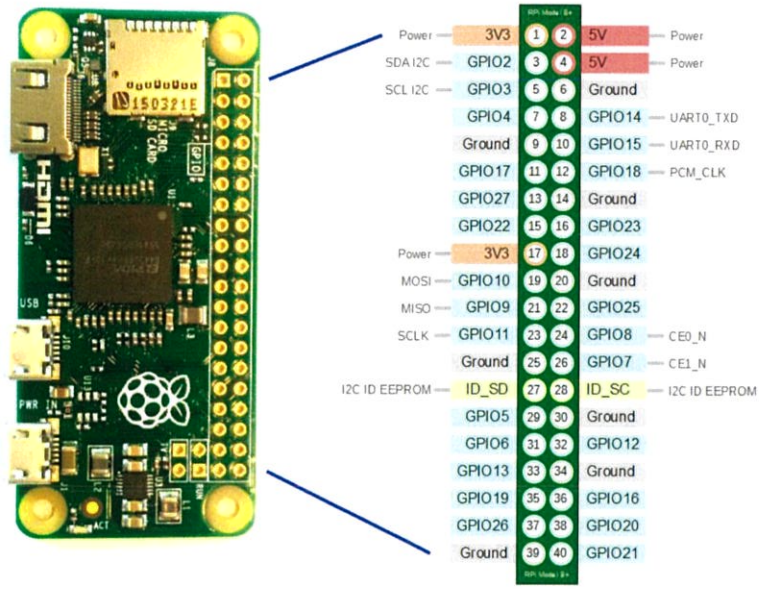
- 802.11 b/g/n wireless LAN
- Bluetooth 4.1
- Bluetooth Low Energy (BLE)
- Like the Pi Zero, it also has:
- 1GHz, single-core CPU
- 512MB RAM
- Mini HDMI and USB On-The-Go ports
- Micro USB power
- HAT-compatible 40-pin header
- Composite video and reset headers
- CSI camera connector

To get started

(not included) you'll need:

- MicroSD card with NOOBS
- Micro USB power supply

ส่วนประกอบของบอร์ด raspberry pi zero w



รูปที่ 2.5 พอร์ต GPIO ของ raspberry pi zero w

2.1.4 WEBUI

ภาษาและโปรแกรมที่ใช้ในการพัฒนาโฮมเพจ

การสร้างเว็บเพจเมื่อเรามีพื้นฐานด้านภาษา HTML ดีแล้วการจะใช้เครื่องมือสำเร็จรูปมาช่วยสร้างก็จะเป็นการประหยัดเวลาได้มาก โปรแกรมสร้างเว็บเพจจะมีอยู่ 2 ประเภท คือ

1. โปรแกรมในกลุ่ม Text Editor หมายถึงโปรแกรมที่เน้นในการใช้ภาษา HTML เป็นหลัก ผู้ใช้จะต้องมีความเข้าใจในการใช้งาน จดจำคำสั่ง รูปแบบของคำสั่งต่างๆ ได้เป็นอย่างดี โปรแกรมในกลุ่มนี้ได้แก่ Notepad, Homesite, CoffeeCup, EditPlus, HotDog Pro, 1st Page 2000 เป็นต้น คุณสมบัติที่ดีของโปรแกรมกลุ่มนี้คือขนาดไฟล์ที่ได้จะเล็ก และตรงกับความต้องการของผู้ออกแบบมากที่สุด

2. โปรแกรมในกลุ่ม WYSIWYG (What You See It What You Get) หมายถึงโปรแกรมที่คุณมองเห็นอย่างไรในขณะกำลังสร้างก็จะได้ผลของเว็บเพจอย่างนั้น ผู้ใช้งานไม่ต้องรู้ภาษา HTML มาก

สามารถนำโปรแกรมในกลุ่มแรกมาช่วยในการแก้ไขไฟล์ในส่วนที่ซ้ำซ้อน ไม่จำเป็นออกไป ทำให้ไฟล์มีขนาดเล็กลงได้

ภาษาที่ใช้ในการเขียนเว็บเพจ

Java

ภาษา Java เป็นภาษาในการเขียนโปรแกรมสำหรับใช้กับเครื่องคอมพิวเตอร์ มีลักษณะคล้ายกับภาษา C++ โดยเป็นการเขียนโปรแกรมแบบเชิงวัตถุ (Object-Oriented Programming : OOP) คิดค้นโดยบริษัท Sun Microsystems ในปี 1991 โดยทางบริษัทมีเป้าหมายที่จะสร้างผลิตภัณฑ์อิเล็กทรอนิกส์สำหรับผู้บริโภคที่ง่ายต่อการใช้ง่าย มีค่าใช้จ่ายต่ำ ไม่มีข้อผิดพลาด และสามารถใช้กับเครื่องใด ๆ ก็ได้

ภาษา Java มีลักษณะของ Applets เป็นระบบ Client/Sever ระหว่างตัว Web Browsers ที่ฝั่ง Client และ Web Sever ที่ฝั่ง Server เหมาะสำหรับเขียนขึ้นใช้งานบน Internet (World Wide Web) ในลักษณะ Close Platform โดยไม่ขึ้นกับ Platform ของระบบปฏิบัติการ(Operation System : OS)ใดๆ เช่น Windows 95/98, Windows NT, Macintosh หรือ Unix

โครงสร้างของภาษา Java

August 15, 2016 · 6,611 views

ในบทนี้ คุณจะได้เรียนโครงสร้างพื้นฐานของภาษา Java และรูปแบบต่างๆ ในการใช้งานของภาษา เราจะมาเริ่มต้นกับโปรแกรมสุดคลาสสิค Hello World Program

Hello word program

Hello World Program เป็นโปรแกรมแรกที่โปรแกรมเมอร์ทุกคนจะได้เขียน มันเป็นโปรแกรมที่แสดงข้อความว่า "Hello World!" ออกทางหน้าจอคอมพิวเตอร์ของคุณ ถ้าคุณสร้างโปรเจ็คของคุณแล้ว ตอนนี้คัดลอกโปรแกรมข้างล่างไปรันได้เลย

```
// Hello World Program
publicclassHelloWorld{
publicstaticvoid main (String[] args){
```

```
// Hello World Program
publicclassHelloWorld{
publicstaticvoid main (String[] args){
System.out.println("Hello World!");
}
}
```

Note: ในการสร้างคลาส ชื่อไฟล์ของคุณต้องตรงกันกับคลาสในโปรแกรม เช่น ตัวอย่างคลาสมีชื่อว่า HelloWorld ไฟล์โปรแกรมของคุณต้องมีชื่อว่า HelloWorld.java

หลังจากที่คุณรันโปรแกรมใน IDE คุณควรจะได้ผลลัพธ์ดังนี้

```
Hello World!
```

ต่อไปลองเปลี่ยนข้อความเป็นข้อความใหม่ที่คุณต้องการ เพื่อดูผลลัพธ์ว่ามันน่าทึ่งแค่ไหน และจากตัวอย่างของโปรแกรมข้างบน มันมีส่วนประกอบที่สำคัญดังนี้

Package: เป็นกลุ่มของคลาสหรือไลบรารี่มาตรฐานของภาษา Java ที่มีฟังก์ชันต่างๆ ให้ใช้มากมาย ในตัวอย่างนั้นไม่มี เราจะพูดเกี่ยวกับเรื่องนี้ในภายหลัง

Class: ในส่วนของการประกาศคลาส จะต้องประกาศคลาสให้ชื่อตรงกับไฟล์เสมอ นอกจาก Inner คลาสที่อยู่ในคลาสเดียวกัน โดยชื่อคลาสนั้นควรจะขึ้นต้นด้วยตัวใหญ่ และถ้ามีหลายคำให้ใช้ตัวพิมพ์ใหญ่แบ่งดังตัวอย่างด้านล่าง

```
publicclassClassName{
...
}
```

Method: หลังจากคลาสสร้างแล้ว จะเป็นประกาศเมธอดภายในคลาส โดยในการที่จะรันโปรแกรมได้จะต้องมีเมธอดที่ชื่อว่า Main ดังตัวอย่างในโปรแกรมด้านบน มันเป็นที่แรกที่โปรแกรมจะเริ่มทำงาน

```
publicstaticvoid main (String[] args){
...
}
```

การใช้ Comment

คอมเมนต์ คือการทำเครื่องหมายสำหรับให้โปรแกรมเมอร์เข้าใจโปรแกรมของพวกเขามากขึ้น หรือป้องกันในกรณีที่อาจจะเกิดการลืมได้ การคอมเมนต์นั้นจะเป็นคำสั่งที่ไม่มีผลต่อการทำงานของโปรแกรม และคอมไพเลอร์จะเพิกเฉยต่อคอมเม้นต์ มันใช้สำหรับมนุษย์เท่านั้น นอกจากนี้คอมเมนต์ยังมีประโยชน์มากในกรณีที่โค้ดโปรแกรมมีจำนวนมาก ข้างล่างเป็นตัวอย่างการคอมเม้นต์ในภาษา Java

```
// Single line comment

/* Multiline comment1
Multiline comment2 */
```

การคอมเม้นต์มีสองแบบคือ Single line และ Multiline โดยแบบ Single line ใช้เครื่องหมาย // และตามสิ่งที่ต้องการคอมเม้นต์ และสำหรับการคอมเม้นต์แบบหลายบรรทัดใช้เครื่องหมาย /* ตามด้วยคอมเม้นต์ และสิ้นสุดที่ */

Semicolon

ทุกคำสั่งการทำงานของโปรแกรมในภาษา Java จะจบด้วยเครื่องหมาย Semicolon (;) นั่นหมายความว่าคุณสามารถเขียนโปรแกรมแบบไหนก็ได้ โดยคอมไพเลอร์จะทราบอัตโนมัติว่าสิ้นสุดคำสั่งที่ไหน เช่น

```
int a =1;
int b =2;int c =3;

System.out.println(
    a + b + b
);
```

แต่เราขอแนะนำให้คุณเขียนคำสั่งละ 1 บรรทัด เพราะมันเป็นวิธีที่ดี และทำให้โค้ดของคุณอ่านง่ายและเป็นระเบียบ

White space

ในภาษา Java คุณสามารถใช้ White space ได้อย่างอิสระตามที่คุณต้องการ โดย White space จะประกอบไปด้วย Space bar Tab และ Enter (return) เพราะว่าคอมไพเลอร์ตรวจการสิ้นสุดของคำสั่งด้วย ; ใช้ while space ทำให้โค้ดอ่านเข้าใจง่าย และเป็นระเบียบ ตัวอย่าง

```
int x =1;
int y=2;
int z =3;
```

จากตัวอย่างนั้นเป็นการประกาศตัวแปรที่ถูกต้อง แต่เราแนะนำแบบตัวแปร x ต่อไปให้คุณสังเกตุการใช้งาน while space ที่ดีในตัวอย่างในบทต่อๆ ไป

Literals

Literals คือค่าของข้อมูลใดๆ ที่กำหนดให้กับตัวแปรได้ เราเรียกมันว่า Constant Literals ทุกค่าที่เป็นไปได้ เช่น "MarcusCode" เป็น String Literals 10 เป็น Integer Literals หรือ true เป็น Boolean Literals โดย Literals เป็นได้แค่ Primitive data type เท่านั้น ตัวอย่างการกำหนดค่าหรือ Literals ให้กับตัวแปร

```
int age =18;
String name ="Marcus";
float pi =3.14f;
double money =45.2;
```

Expressions

Expression เป็นการกระทำระหว่างตัวแปรกับตัวดำเนินการเพื่อให้ได้ผลลัพธ์ใหม่ เช่น 4 + 3เป็น expression ของการบวกเลขและได้ได้ผลลัพธ์เท่ากับ 7 หรือ 1 == 1 เป็น expression ของการเปรียบเทียบระหว่างค่าสองค่าว่าเท่ากันหรือไม่ และได้ผลลัพธ์เป็น true คุณจะได้เรียนเกี่ยวกับตัวดำเนินการในบทต่อไป

Keywords in Java

Keyword คือคำที่สงวนไว้ในภาษา Java นั่นหมายความว่าคุณไม่สามารถนำคำเหล่านี้ไปประกาศเป็นชื่อตัวแปร เมธอด หรือว่าคลาสได้ เพราะว่า Keyword ถูกใช้โดยคอมไพเลอร์เพื่อให้มันทำงานได้สมบูรณ์ ข้างล่างนี้เป็นลิสต์ของ Keyword ในภาษา Java

abstract	continue	for	new	switch
assert	default	goto	package	synchronized
boolean	do	if	private	this
break	double	implements	protected	throw
byte	else	import	public	throws
case	enum	instanceof	return	transient
catch	extends	int	short	try
char	final	interface	static	void
class	finally	long	strictfp	volatile
const	float	native	super	while

Keywords list in Java

HTML หรือ HyperText Markup Language

เป็นภาษาคอมพิวเตอร์รูปแบบหนึ่ง ที่มีลักษณะเป็นภาษาในเชิง การบรรยายเอกสารไฮเปอร์มีเดีย (Hypermedia Document Description Language) เพื่อนำเสนอเอกสารนั้น เผยแพร่ในระบบเครือข่ายใยแมงมุม WWW (World Wide Web) มีโครงสร้างการเขียน ที่อาศัยตัวกำกับ เรียกว่า แท็ก (Tag) ควบคุมการแสดงผลของข้อความ, รูปภาพ หรือวัตถุอื่นๆ ผ่านโปรแกรมเบราเซอร์(Browser)

เผยแพร่ในระบบเครือข่ายใยแมงมุม WWW (World Wide Web) มีโครงสร้างการเขียน ที่อาศัยตัวกำกับ เรียกว่า แท็ก (Tag) ควบคุมการแสดงผลของข้อความ, รูปภาพ หรือวัตถุอื่นๆ ผ่านโปรแกรมเบราเซอร์ (Browser)

โครงสร้างของ HTML จะประกอบไปด้วยส่วนของคำสั่ง 2 ส่วน คือ ส่วนที่เป็น ส่วนหัว (Head) และส่วนที่เป็นเนื้อหา (Body)

โดยมีรูปแบบคำสั่งดังนี้

รูปที่ 2.6 โครงสร้างภาษา HTML

การจัดโครงสร้างแฟ้มเอกสาร

ในความง่ายของภาษา HTML นั้นเพราะภาษานี้ไม่มีโครงสร้างใด ๆ มากำหนดนอก จากโครงสร้างพื้นฐานเท่านั้น

หรือ แม้แต่จะไม่มีโครงสร้าง พื้นฐานอยู่ โปรแกรมที่เขียนขึ้นมานั้นก็สามารถทำงานได้เสมือนมีโครงสร้างทั่งนี้เป็นเพราะว่าตัวโปรแกรม
เว็บเบราเซอร์ จะมองเห็นทุกสิ่งทุกอย่างในโปรแกรม HTML เป็นส่วนเนื้อหาทั้งสิ้น

ยกเว้นใน ส่วนหัว ที่ต้อง มีการกำหนด แยกออกไปให้ เห็นชัดเท่านั้น จะเขียน คำสั่ง หรือ ข้อความที่ ต้องการ ให้แสดง อย่างไรก็ได้
เป็นเสมือนพิมพ์งานเอกสารทั่ว ๆ ไปเพียง แต่ ทำตำแหน่ง ใดมีการ ทำตำแหน่ง พิเศษขึ้นมา เว็บเบราเซอร์ถึงจะแสดงผล ออกมาตามที่
ถูกกำหนด โดยใช้คำสั่งให้ตรงกับ รหัสที่กำหนดเท่านั้น

การแสดงผลที่เว็บเบราเซอร์

หลังจากมีการพิมพ์โปรแกรมนี้เสร็จเรียบร้อยแล้ว ให้บันทึกเป็น ไฟล์ที่มีนามสกุล .htm หรือ .html จากนั้นให้เรียกโปรแกรมเว็บเบราเซอร์ขึ้นมาทำการทดสอบ ข้อมูลที่เราสร้างจะถูกนำมาที่ออกมาแสดงที่จอภาพ ถ้าไม่เขียนอะไรผิด บนจอภาพก็จะแสดงผลตามนั้น
ถ้าเรามีการปรับปรุงแก้ไขข้อมูลในโปรแกรมเดิม ให้อยู่ในรูปของ โปรแกรมใหม่ ก็จำ เป็นต้องโหลดโปรแกรมขึ้นมาใหม่ เพียงแต่เลื่อนเมาส์
ไปคลิกที่ปุ่ม Refresh โปรแกรมก็จะทำการ ประมวลผลและแสดงผลออกมาใหม่ ในคำสั่ง HTML ส่วนใหญ่ใช้ตัวเปิด เป็นเครื่องหมายน้อยกว่า
<ตามด้วยคำสั่ง และปิดท้ายด้วยเครื่องหมายมากกว่า>และมีตัวปิดที่มีรูปแบบเหมือนตัวเปิดเสมอ เพียงแต่จะมีเครื่อง หมาย / อยู่หน้าคำสั่งนั้นๆ เช่น คำสั่ง<BODY>จะมี</BODY>เป็นคำสั่งปิด
เมื่อใดที่ผู้เขียนลืมหรือพิมพ์คำสั่งผิด จะส่งผลให้การทำงานของโปรแกรมผิดพลาดทันที

คำสั่งเริ่มต้นสำหรับ HTML

คำสั่งหรือ Tag ที่ใช้ในภาษา HTML ประกอบไปด้วยเครื่องหมายน้อยกว่า<ตามด้วย ชื่อคำสั่งและปิดท้ายด้วยเครื่องหมายมากกว่า
>เป็นส่วนที่ทำหน้าที่ตกแต่งข้อความ เพื่อ การแสดงผลข้อมูล โดยทั่วไปคำสั่งของ HTML ส่วนใหญ่จะอยู่เป็นคู่ มีเพียงบาง คำสั่งเท่านั้น
ที่มีรูปแบบคำสั่งอยู่เพียงตัวเดียว ในแต่ละคำสั่ง จะมีคำสั่งเปิดและปิด คำสั่งปิดของแต่ละ คำสั่งจะมีรูปแบบเหมือนคำสั่งเปิด เพียงแต่จะเพิ่ม /

(Slash) นำหน้าคำสั่ง ปิดให้ดู แตกต่าง เท่านั้น และในคำสั่งเปิดบางคำสั่ง อาจมีส่วนขยายอื่นผสมอยู่ด้วย ในการเขียน ด้วยตัวอักษร
เล็กหรือใหญ่ ทั้งหมดหรือเขียนปนกันก็ได้ ไม่มีผลอะไร

คำสั่งเริ่มต้น

รูปแบบ <HTML>.....</HTML>

คำสั่ง <HTML>เป็นคำสั่งเริ่มต้นในการเขียนโปรแกรม และ</HTML>เป็นคำสั่งจุดสิ้นสุดโปรแกรม เหมือนคำสั่ง Beign และ End ใน Pascal

คำสั่งการทำหมายเหตุ

รูปแบบ <!-- ..... -->

ตัวอย่าง <!-- END WEBSTAT CODE --> ข้อความที่อยู่ในคำสั่งจะปรากฏอยู่ในโปรแกรมแต่ไม่ถูกแสดง บนจอภาพ

ส่วนหัว

รูปแบบ <HEAD>.....</HEAD>

ใช้กำหนดข้อความ ในส่วนที่เป็น ชื่อเรื่อง ภายในคำสั่งนี้ จะมีคำสั่งย่อย อีกหนึ่งคำสั่ง คือ<TITLE>

กำหนดข้อความในไตเติลบาร์

รูปแบบ <TITLE>.....</TITLE>

ตัวอย่าง <TITLE>บทเรียน HTML </TITLE>

เป็นส่วนแสดงชื่อของเอกสาร จะปรากฏ ขณะที่ไฟล์ HTML ทำงานอยู่ ข้อความ ที่กำหนด ในส่วนนี้จะไม่ถูกนำไปแสดง ผลของ เว็บเบราเซอร์แต่จะปรากฏในส่วนของไตเติบาร์ (Title bar) ที่เป็นชื่อของวินโดว์ข้างบนไม่ควรให้ยา เกินไป เพียงให้รู้ว่าเว็บเพจที่กำลัง ใช้งานอยู่เกี่ยวข้องกับอะไร

ส่วนของเนื้อหา

รูปแบบ <BODY>.....</BODY>

ส่วนเนื้อหาของโปรแกรมจะเริ่มต้นด้วย คำสั่ง<BODY>และจบลงด้วย</BODY>ภายในคำสั่งนี้ คือ ส่วนที่จะ แสดงทางจอภาพ

Tag

Tag เป็นลักษณะเฉพาะของภาษา HTML ใช้ในการระบุรูปแบบคำสั่ง หรือการลงรหัสคำสั่ง HTML ภายในเครื่องหมาย less-than bracket (<) และ greater-than bracket (>) โดยที่ Tag HTML แบ่งได้ 2 ลักษณะ คือ

- Tagเดี่ยว
  เป็น Tag ที่ไม่ต้องมีการปิดรหัส เช่น <P>, <HR> เป็นต้น
- Tagเปิด/ปิด
  เป็น Tag ที่ประกอบด้วย Tag เปิด และ Tag ปิด โดย Tag ปิด จะมีเครื่องหมาย slash (/) นำหน้าคำสั่งใน Tag นั้นๆ เช่น <B>...</B>, <P>...</P> เป็นต้น

Attributes

Attributes เป็นส่วนขยายความสามารถของ Tag จะต้องใส่ภายในเครื่องหมาย <> ในส่วน Tag เปิดเท่านั้น Tag คำสั่ง HTML แต่ละคำสั่ง จะมี Attribute แตกต่างกันไป และมีจำนวนไม่เท่ากัน การระบุ Attribute มากกว่า 1 Attribute ให้ใช้ช่องว่างเป็นตัวคั่น เช่น Tag ควบคุมเกี่ยวกับ รูปภาพ <IMG> มี Attributes ดังนี้

<IMG SRC="filename" WIDTH="n" HEIGHT="n" ALT="text" BORDER="n">

โดย SRC เป็น Attribute ควบคุมชื่อไฟล์ภาพที่จะนำมาแสดงผล

WIDTH เป็น Attribute ควบคุมความกว้างของภาพ

ไฟล์เอกสาร HTML เป็นไฟล์ข้อความรูปแบบหนึ่ง (Text File) ที่เก็บชุดคำสั่ง HTML ดังนั้นการสร้างเอกสาร HTML จึงสามารถใช้ Text Editor ตัวใดก็ได้ เช่น NotePad หรือ WordPadThai

โครงสร้างเอกสาร HTML

ไฟล์เอกสาร HTML ประกอบด้วยส่วนประกอบสองส่วนคือ Head กับ Body โดยสามารถเปรียบเทียบได้ง่ายๆ ก็คือ ส่วน Head จะคล้ายกับส่วนที่เป็น Header ของหน้าเอกสารทั่วไป หรือบรรทัด Title ของหน้าต่างการทำงานในระบบ Windows สำหรับส่วน Body จะเป็นส่วนเนื้อหาของเอกสารนั้นๆ โดยทั้งสองส่วนจะอยู่ภายใน Tag <HTML>...</HTML>

โครงสร้างไฟล์ HTML

PHP

PHP ย่อมาจาก " Hypertext Preprocessor " เป็นภาษา Server-Side Script อีกภาษาหนึ่งเช่นเดียวกันกับ ASP ที่มีการทำงานที่เครื่องคอมพิวเตอร์ฝั่ง Server ซึ่งรูปแบบในการเขียนคำสั่งการทำงานนั้นจะมีลักษณะคล้ายกับภาษา Perl หรือภาษา C และสามารถใช้ร่วมงานกันกับ ภาษา HTML ได้อย่างมีประสิทธิภาพ

หมายเหตุ ในการสร้างเว็บจะใช้ Script อยู่ 2 รูปแบบคือ

Server-Side Script เป็นลักษณะการทำงานบนเครื่อง Server และแปลออกมาเป็นภาษา HTML เช่น ASP, CGI

Client-Side Script เป็นลักษณะการทำงานบนเครื่อง Client (เครื่องผู้ใช้) เช่น JavaScript, VBScript

PHP สามารถทำอะไรได้บ้าง

ความสามารถของ PHP นั้นสามารถที่จะทำงานเกี่ยวกับ Dynamic Web ได้ทุกรูปแบบเหมือนกับ CGI หรือ ASP ไม่ว่าจะเป็นการจัดการดูแลระบบฐานข้อมูล ระบบรักษาความปลอดภัย การรับ – ส่ง Cookies โดยที่ PHP นั้นสามารถที่จะติดต่อกับโปรแกรมจัดการฐานข้อมูลที่มีอยู่มากมาย ดังนี้

Adabas D	InterBase Solid	Microsoft Access
DBase	mSQL	Sybase
Empress	MySQL	Velocis
FilePro	Oracle &	

PYTHON

โครงสร้างของภาษา Python

ในบทนี้ คุณจะได้เรียนรู้และทำความเข้าใจในโครงของภาษา Python ในภาษาคอมพิวเตอร์นั้นก็มีโครงสร้างของภาษาเช่นเดียวกกับภาษามนุษย์ ซึ่งสิ่งเหล่านี้ถูกกำหนดเพื่อเป็นรูปแบบและวิธีในการเขียนโปรแกรมในภาษา Python มันใช้สำหรับควบคุมวิธีที่คุณจะเขียนโค้ดของคุณเพื่อให้เข้าใจโดยตัวแปรภาษาหรือคอมไพเลอร์

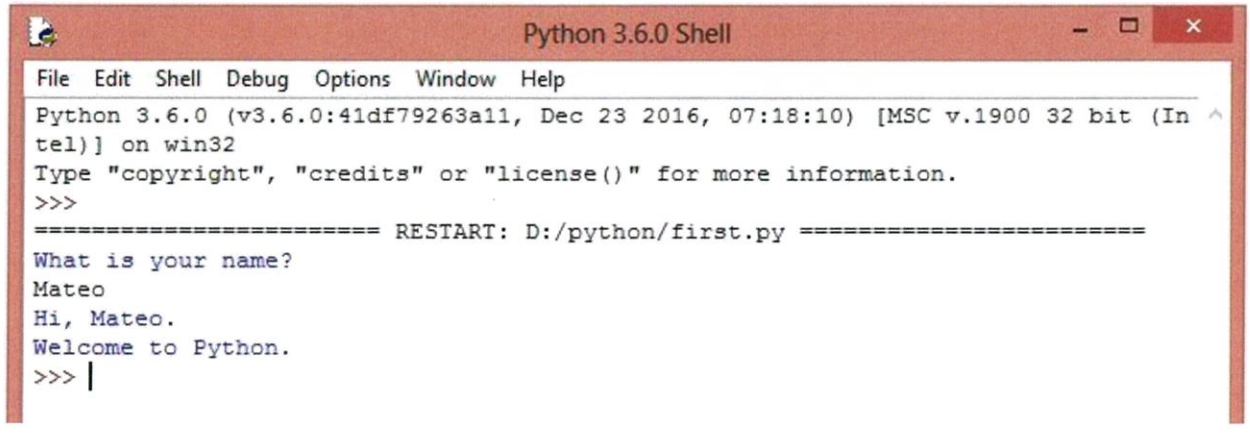
ในบทนี้ คุณจะได้เรียนรู้และทำความเข้าใจในโครงของภาษา Python ในภาษาคอมพิวเตอร์นั้นก็มีโครงสร้างของภาษาเช่นเดียวกกับภาษามนุษย์ ซึ่งสิ่งเหล่านี้ถูกกำหนดเพื่อเป็นรูปแบบและวิธีในการเขียนโปรแกรมในภาษา Python มันใช้สำหรับควบคุมวิธีที่คุณจะเขียนโค้ดของคุณเพื่อให้เข้าใจโดยตัวแปรภาษาหรือคอมไพเลอร์

Simple Python program

เพื่อเริ่มต้นการเรียนรู้ในภาษา Python มาดูตัวอย่างของโปรแกรมอย่างง่าย โดยเป็นโปรแกรมที่ถามชื่อผู้ใช้และแสดงข้อความทักทายทางหน้าจอ มาเริ่มเขียนโปรแกรมแรกในภาษา Python ของคุณ ให้คัดลอกโปรแกรมข้างล่างแล้วนำไปรันใน IDE

```
# My first Python program
name = input('What is your name?\n')
print ('Hi, %s.' % name)
print ('Welcome to Python.')
```

ในตัวอย่างเป็นโปรแกรมในการรับชื่อและแสดงข้อความทักทายออกทางหน้าจอ ในการรันโปรแกรมคุณสามารถรันได้หลายวิธี แต่ที่แนะนำคือการใช้ Python shell ให้คุณเปิด Python shell ขึ้นมาแล้วกดสร้างไฟล์ใหม่โดยไปที่ *File -> New File* จะปรากฏกล่อง Text editor ของภาษา Python ขึ้นมา เพื่อรันโปรแกรม *Run -> Run Module* หรือกด *F5* โปรแกรมจะเปลี่ยนกลับไปยัง Python shell และเริ่มต้นทำงาน



รูปที่ 2.7 ผลลัพธ์การทำงานในการรันโปรแกรม first.py

Module

ในตัวอย่างโปรแกรมรับชื่อของเรา เป็นโปรแกรมแรกของเราในบทเรียน Python นี้ และเราได้บันทึกเป็นไฟล์ที่ชื่อว่า *first.py* ซึ่งไฟล์ของภาษา Python นั้นจะเรียกว่า Module ซึ่ง Module จะประกอบไปด้วยคลาส ฟังก์ชัน และตัวแปรต่างๆ และนอกจากนี้เรายังสามารถ import โมดูลอื่นเข้ามาในโปรแกรมได้ ซึ่งโมดูลอาจจะอยู่ภายใน package ซึ่งเป็นเหมือน directory ของ Module ในตัวอย่าง *first.py* จึงเป็นโมดูลของโปรแกรมแรกของเรา

Comment

คอมเมนต์ในภาษา Python นั้นเริ่มต้นด้วยเครื่องหมาย # คอมเมนต์สามารถเริ่มต้นที่ตำแหน่งแรกของบรรทัดและหลังจากนั้นจะประกอบไปด้วย Whilespace หรือโค้ดของโปรแกรม หรือคำอธิบาย ซึ่งโดยทั่วไปแล้วคอมเมนต์มักจะใช้สำหรับอธิบายซอสโค้ดที่เราเขียนขึ้นและมันไม่มีผลต่อการทำงานของโปรแกรม นี่เป็นตัวอย่างการคอมเมนต์ในภาษา Python

```
# My first Python program

'''
This is a multiline comment
'''

print ('Hello Python.') # Inline comment
```

ในตัวอย่าง เราได้คอมมเมนต์สามแบบด้วยกัน แบบแรกเป็นการคอมเมนต์แบบ single line แบบที่สองเป็นการคอมเมนต์แบบ multiline line และแบบสุดท้ายเป็นการคอมมเมนต์แบบ inline หรือการคอมเมนต์ภายในบรรทัดเดียวกัน

Statement

Statement คือคำสั่งการทำงานของโปรแกรม แต่ละคำสั่งในภาษา Python นั้นจะแบ่งแยกด้วยการขึ้นบรรทัดใหม่ ซึ่งจะแตกต่างจากภาษา C และ Java ซึ่งใช้เครื่องหมายเซมิโคลอนสำหรับการจบคำสั่งการทำงาน แต่อย่างไรก็ตาม ในภาษา Python นั้นคุณสามารถมีหลายคำสั่งในบรรทัดเดียวกันได้โดยการใช้เครื่องหมายเซมิโคลอน ;

```
name = input('What is your name?\n')
print ('Hi, %s.' % name);
print ('Welcome to Python.'); print ('Do you love it?')
```

ในตัวอย่าง เรามี 4 คำสั่งในโปรแกรม สองบรรทัดแรกเป็นคำสั่งที่ใช้บรรทัดใหม่ในการจบคำสั่ง ซึ่งเป็นแบบปกติในภาษา Python และบรรทัดสุดท้ายเรามีสองคำสั่งในบรรทัดเดียวที่คั่นด้วยเครื่องหมาย ; สำหรับการจบคำสั่ง

Indentation and while space

ในภาษา Python นั้นใช้ Whilespace และ Tab สำหรับกำหนดบล็อคของโปรแกรม เช่น คำสั่ง If Else For หรือการประกาศฟังก์ชัน ซึ่งคำสั่งเหล่านี้นั้นเป็นคำสั่งแบบบล็อค โดยจำนวนช่องว่างที่ใช้นั้นต้องเท่ากัน มาดูตัวอย่างของบล็อคคำสั่งในภาษา Python

```
n = int(input ('Input an integer: '))

if (n > 0):
print ('x is positive number')
print ('Show number from 0 to %d' % (n - 1))

else:
print ('x isn\'t positive number')

for i in range(n):
print(i)
```

ในตัวอย่าง เป็นบล็อคของโปรแกรมจากท 3 คำสั่ง ในคำสั่งแรกคือ If ในบล็อคนี้มีสองคำสั่งย่อยอยู่ภายใน ที่หัวของบล็อคนั้นจะต้องมีเครื่องหมาย : กำหนดหลังคำสั่งในการเริ่มต้นบล็อคเสมอ อีกสองบล็อคสุดท้ายนั้นเป็นคำสั่ง Else และ For ซึ่งมีหนึ่งคำสั่งย่อยอยู่ภายใน ในภาษา Python นี้เข้มงวดกับช่องว่างภายในบล็อคมาก นั้นหมายความว่าทุกคำสั่งย่อยภายในบล็อคนั้นต้องมีจำนวนช่องว่างเท่ากันเสมอ

```
n = int(input ('Input an integer: '))

# Invalid indent
if (n > 0):
print ('x is positive number')
print ('Show number from 0 to %d' % (n - 1))

# Valid indent
else:
print ('x isn\'t positive number')

# Valid indent
for i in range(n):
print(i)
```

นี่เป็นตัวอย่างการใช้งานช่องว่างที่ถูกต้องและไม่ถูกต้องภานในบล็อค ใสคำสั่ง If นั้นไม่ถูกเพราะทั้งสองคำสั่งมีจำนวนช่องว่างที่ไม่เท่ากัน สำหรับในคำสั่ง Else และ For นั้นถูกต้อง

Literals

ในการเขียนโปรแกรม Literal คือเครื่องหมายที่ใช้แสดงค่าของค่าคงที่ในโปรแกรม ในภาษา Python นั้นมี Literal ของข้อมูลประเภทต่างๆ เช่น Integer Floating-point number และ String หรือแม้กระทั่งตัวอักษรและ boolean นี่เป็นตัวอย่างของการกำหนด Literal ให้กับตัวแปรในภาษา Python

```
a = 1
b = -1.64E3
```

```
c = True
d = "marcuscode.com"
e = 'A'
```

ในตัวอย่าง เป็นการกำหนด Literal ประเภทต่างๆ ให้กับตัวแปร ในค่าที่เป็นแบบตัวเลขนั้นสามารถกำหนดค่าลงไปโดยตรงได้ทันทีและสามารถกำหนดในรูปแบบสั้นได้อย่างในตัวแปร b และสำหรับ boolean นั้นจะเป็น True ส่วน String หรือ Character นั้นจะต้องอยู่ภายในเครื่องหมาย double quote หรือ single quote เสมอ

Expressions

Expression คือการทำงานร่วมกันระหว่างค่าตั้งแต่หนึ่งไปจนถึงหลายค่า โดยค่าเหล่านี้จะมีตัวดำเนินการสำหรับควบคุมการทำงาน ในภาษา Python นั้น Expression จะมีสองแบบคือ Boolean expression เป็นการกระทำกันของตัวแปรและตัวดำเนินการและจะได้ผลลัพธ์เป็นค่า Boolean โดยทั่วไปแล้วมักจะเป็นตัวดำเนินการเปรียบเทียบค่าและตัวดำเนินการตรรกศาสตร์ และ Expression ทางคณิตศาสตร์ คือการกระทำกันกับตัวดำเนินการและได้ค่าใหม่ที่ไม่ใช้ Boolean นี่เป็นตัวอย่างของ Expressions ในภาษา Python

```
a = 4
b = 5

# Boolean expressions
print(a == 4)
print(a == 5)
print(a == 4 and b == 5)
print(a == 4 and b == 8)

# Non-boolean expressions
print(a + b)
print(a + 2)
```

```
# Boolean expressions
print(a == 4)
print(a == 5)
print(a == 4 and b == 5)
print(a == 4 and b == 8)

# Non-boolean expressions
print(a + b)
print(a + 2)
print(a * b)
print(((a * a) + (b * b)) / 2)
print("Python " + "Language")
```

ในตัวอย่าง เรามีตัวแปร a และ b และกำหนดค่าให้กับตัวแปรเหล่านี้และทำงานกับตัวดำเนินการประเภทต่างๆ ที่แสดง Expression ในรูปแบบของ Boolean expression ที่จะได้ผลลัพธ์สุดท้ายเป็นเพียงค่า True และ False เท่านั้น ส่วน Non-Boolean expression นั้นสามารถเป็นค่าใดๆ ที่ไม่ใช่ Boolean

```
True
False
True
False
9
6
20
20.5
Python Language
```

นี่เป็นผลลัพธ์การทำงานของโปรแกรมในการทำงานของ Expression ในภาษา Python

continue	def	del	elif
else	except	finally	for
from	global	if	import
in	is	lambda	nonlocal
not	or	pass	raise
return	try	while	with
yield			

2.2.5 รีเลย์ (Relay driver)

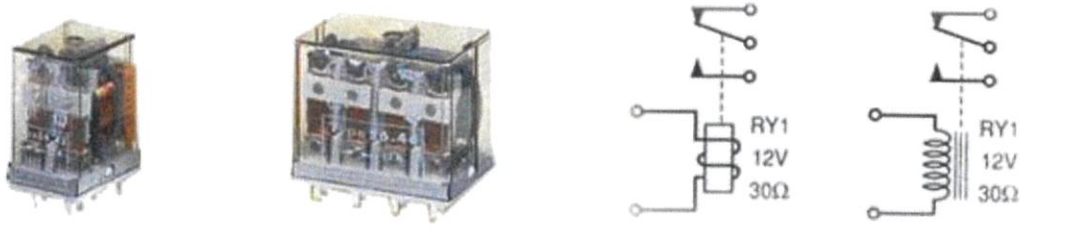
รีเลย์ (Relay)

เป็นอุปกรณ์ที่เปลี่ยนพลังงานไฟฟ้าให้เป็นพลังงานแม่เหล็ก เพื่อใช้ในการดึงดูดหน้าสัมผัสของคอนแทคให้เปลี่ยนสภาวะ โดยการป้อนกระแสไฟฟ้าให้กับขดลวด เพื่อทำการปิดหรือเปิดหน้าสัมผัสคล้ายกับสวิตช์อิเล็กทรอนิกส์ ซึ่งเราสามารถนำรีเลย์ไปประยุกต์ใช้ ในการควบคุมวงจรต่าง ๆ ในงานช่างอิเล็กทรอนิกส์มากมาย

2.2.5 รีเลย์ (Relay driver)

รีเลย์ (Relay)

เป็นอุปกรณ์ที่เปลี่ยนพลังงานไฟฟ้าให้เป็นพลังงานแม่เหล็ก เพื่อใช้ในการดึงดูดหน้าสัมผัสของคอนแทคให้เปลี่ยนสภาวะ โดยการป้อนกระแสไฟฟ้าให้กับขดลวด เพื่อทำการปิดหรือเปิดหน้าสัมผัส คล้ายกับสวิตช์อิเล็กทรอนิกส์ ซึ่งเราสามารถนำรีเลย์ไปประยุกต์ใช้ ในการควบคุมวงจรต่าง ๆ ในงานช่างอิเล็กทรอนิกส์มากมาย



รูปร่างของรีเลย์ที่มีตัวถังเป็นพลาสติกใสป้องกันฝุ่น | สัญลักษณ์แบบลวดพัน | สัญลักษณ์แบบตัวเหนี่ยวนำพันแกนเหล็ก

รูปที่ 2.8รูปตัวอย่างรีเลย์และสัญลักษณ์รีเลย์

รีเลย์ ประกอบด้วยส่วนสำคัญ 2 ส่วนหลักก็คือ

1. ส่วนของขดลวด (coil) เหนี่ยวนำกระแสต่ำ ทำหน้าที่สร้างสนามแม่เหล็กไฟฟ้าให้แกนโลหะไปกระทุ้งให้หน้าสัมผัสต่อกัน ทำงานโดยการรับแรงดันจากภายนอกต่อคร่อมที่ขดลวดเหนี่ยวนำนี้ เมื่อขดลวดได้รับแรงดัน(ค่าแรงดันที่รีเลย์ต้องการขึ้นกับชนิดและรุ่นตามที่ผู้ผลิตกำหนด) จะเกิดสนามแม่เหล็กไฟฟ้าทำให้แกนโลหะด้านในไปกระทุ้งให้แผ่นหน้าสัมผัสต่อกัน

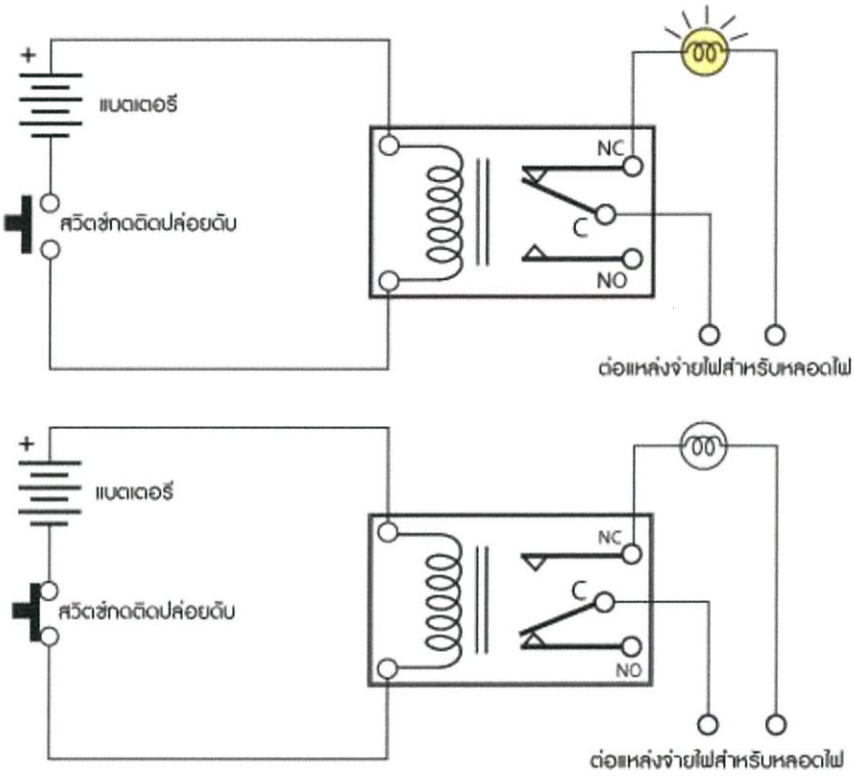
2. ส่วนของหน้าสัมผัส (contact) ทำหน้าที่เหมือนสวิตช์จ่ายกระแสไฟให้กับอุปกรณ์ที่เราต้องการนั่นเอง

จุดต่อใช้งานมาตรฐาน ประกอบด้วย

จุดต่อ NC ย่อมาจาก **normal close** หมายความว่าปกติดปิด หรือ หากยังไม่จ่ายไฟให้ขดลวดเหนี่ยวนำหน้าสัมผัสจะติดกัน โดยทั่วไปเรามักต่อจุดนี้เข้ากับอุปกรณ์หรือเครื่องใช้ไฟฟ้าที่ต้องการให้ทำงานตลอดเวลาเช่น

จุดต่อ NO ย่อมาจาก normal open หมายความว่าปกติเปิด หรือหากยังไม่จ่ายไฟให้ขดลวดเหนี่ยวนำหน้าสัมผัสจะไม่ติดกัน โดยทั่วไปเรามักต่อจุดนี้เข้ากับอุปกรณ์หรือเครื่องใช้ไฟฟ้าที่ต้องการควบคุมการเปิดปิดเช่นโคมไฟสนามหนือหน้าบ้าน

จุดต่อ C ย่อมากจาก common คือจุดร่วมที่ต่อมาจากแหล่งจ่ายไฟ



รูปที่ 2.9การทำงานของส่วนหน้าสัมผัส

ข้อคำถึงในการใช้งานรีเลย์ทั่วไป

1. แรงดันใช้งาน หรือแรงดันที่ทำให้รีเลย์ทำงานได้ หากเราดูที่ตัวรีเลย์จะระบุค่า แรงดันใช้งานไว้ (หากใช้ในงานอิเล็กทรอนิกส์ ส่วนมากจะใช้แรงดันกระแสตรงในการใช้งาน) เช่น 12VDC คือต้องใช้แรงดันที่ 12 VDC เท่านั้นหากใช้มากกว่านี้ ขดลวดภายใน ตัวรีเลย์อาจจะขาดได้ หรือหากใช้แรงดันต่ำกว่ามาก รีเลย์จะไม่ทำงาน ส่วนในการต่อวงจรนั้นสามารถต่อขั้วใดก็ได้ครับ เพราะตัวรีเลย์ จะไม่ระบุขั้วต่อไว้ (นอกจากชนิดพิเศษ)

ชนิดของรีเลย์

รีเลย์ที่นิยมใช้งานและรู้จักกันแพร่หลาย 4 ชนิด

1.อาร์เมเจอร์รีเลย์ (Armature Relay)

2.รีดรีเลย์ (Reed Relay)

3.รีดสวิตช์ (Reed Switch)

4.โซลิดสเตตรีเลย์ (Solid-State Relay)

ประเภทของรีเลย์

เป็นอุปกรณ์ทำหน้าที่เป็นสวิตช์มีหลักการทำงานคล้ายกับ ขดลวดแม่เหล็กไฟฟ้าหรือโซลินอยด์ (solenoid) รีเลย์ใช้ในการควบคุมวงจร ไฟฟ้าได้อย่างหลากหลาย รีเลย์เป็นสวิตช์ควบคุมที่ทำงานด้วยไฟฟ้า แบ่งออกตามลักษณะการใช้งานได้เป็น 2 ประเภทคือ

1.รีเลย์กำลัง (power relay) หรือมักเรียกกันว่าคอนแทกเตอร์ (Contactor or Magneticcontactor) ใช้ในการควบคุมไฟฟ้ากำลัง มีขนาดใหญ่กว่ารีเลย์ธรรมดา

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

ชนิดของรีเลย์

การแบ่งชนิดของรีเลย์สามารถแบ่งได้ 11 แบบ คือ

ชนิดของรีเลย์แบ่งตามลักษณะของคอยล์ หรือ แบ่งตามลักษณะการใช้งาน (Application) ได้แก่ รีเลย์ดังต่อไปนี้

1.รีเลย์กระแส (Current relay) คือ รีเลย์ที่ทำงานโดยใช้กระแสมีทั้งชนิดกระแสขาด (Under-current)
และกระแสเกิน (Over current)
2.รีเลย์แรงดัน (Voltage relay) คือ รีเลย์ ที่ทำงานโดยใช้แรงดันมีทั้งชนิดแรงดันขาด (Under-voltage) และ แรงดันเกิน (Over voltage)
3.รีเลย์ช่วย (Auxiliary relay) คือ รีเลย์ที่เวลาใช้งานจะต้องประกอบเข้ากับรีเลย์ชนิดอื่น จึงจะทำงานได้
4.รีเลย์กำลัง (Power relay) คือ รีเลย์ที่รวมเอาคุณสมบัติของรีเลย์กระแส และรีเลย์แรงดันเข้าด้วยกัน5.รีเลย์เวลา (Time relay) คือ รีเลย์ที่ทำงานโดยมีเวลาเข้ามาเกี่ยวข้องด้วย ซึ่งมีอยู่ด้วยกัน 4 แบบ คือ
5.1 รีเลย์กระแสเกินชนิดเวลาผกผันกับกระแส (Inverse time over current relay) คือ รีเลย์ ที่มีเวลาทำงานเป็นส่วนกลับกับกระแส
5.2 รีเลย์กระแสเกินชนิดทำงานทันที (Instantaneous over current relay) คือรีเลย์ที่ทำงานทันทีทันใดเมื่อมีกระแสไหลผ่านเกินกว่าที่กำหนดที่ตั้งไว้
5.3 รีเลย์แบบดิฟฟินิตไทม์เล็ก (Definite time lag relay) คือ รีเลย์ ที่มีเวลาการทำงานไม่ขึ้นอยู่กับความมากน้อยของกระแสหรือค่าไฟฟ้าอื่นๆ ที่ทำให้เกิดงานขึ้น
5.4 รีเลย์แบบอินเวอสดิฟฟินิตมินิมั่มไทม์เล็ก (Inverse definite time lag relay) คือ รีเลย์ ที่ทำงานโดยรวมเอาคุณสมบัติของเวลาผกผันกับกระแส (Inverse time) และ แบบดิฟฟินิตไทม์แล็ก (Definite time lag relay) เข้าด้วยกัน
6.รีเลย์กระแสต่าง (Differential relay) คือ รีเลย์ที่ทำงานโดยอาศัยผลต่างของกระแส
7.รีเลย์มีทิศ (Directional relay) คือรีเลย์ที่ทำงานเมื่อมีกระแสไหลผิดทิศทาง มีแบบรีเลย์กำลังมีทิศ (Directional power relay) และรีเลย์กระแสมีทิศ (Directional current relay)
8.รีเลย์ระยะทาง (Distance relay) คือ รีเลย์ระยะทางมีแบบต่างๆ ดังนี้
- รีแอกแตนซ์รีเลย์ (Reactance relay)
- อิมพีแตนซ์รีเลย์ (Impedance relay)
- โมห์รีเลย์ (Mho relay)
- โอห์มรีเลย์ (Ohm relay)
- โพลาไรซ์โมห์รีเลย์ (Polaized mho relay)

- ออฟเซทโมห์รีเลย์ (Off set mho relay)

9.รีเลย์อุณหภูมิ (Temperature relay) คือ รีเลย์ที่ทำงานตามอุณหภูมิที่ตั้งไว

10.รีเลย์ความถี่ (Frequency relay) คือ รีเลย์ที่ทำงานเมื่อความถี่ของระบบต่ำกว่าหรือมากกว่าที่ตั้งไว้

11.บูคโฮลซ์รีเลย์ (Buchholz 's relay) คือรีเลย์ที่ทำงานด้วยก๊าซ ใช้กับหม้อแปลงที่แช่อยู่ในน้ำมัน เมื่อเกิด ฟอลต์ ขึ้นภายในหม้อแปลง จะทำให้น้ำมันแตกตัวและเกิดก๊าซขึ้นภายในไปดันหน้าสัมผัสให้รีเลย์ทำงาน

ความรู้ทั่วไปเกี่ยวกับรีเลย์

1.1 หน้าที่ของรีเลย์ คือ เป็นอุปกรณ์ไฟฟ้าที่ใช้ตรวจสอบสภาพการณ์ของทุกส่วน ในระบบกำลังไฟฟ้าอยู่ตลอดเวลาหากระบบมีการทำงานที่ผิดปกติ รีเลย์จะเป็นตัวสั่งการให้ตัดส่วนที่ลัดวงจรหรือส่วนที่ทำงานผิดปกติ ออกจากระบบทันทีโดยเซอร์กิตเบรกเกอร์จะเป็นตัวที่ตัดส่วนที่เกิดฟอลต์ออกจากระบบจริงๆ

1.2 ประโยชน์ของรีเลย์

1.ทำให้ระบบส่งกำลังมีเสถียรภาพ (Stability) สูงโดยรีเลย์จะตัดวงจรเฉพาะส่วนที่เกิดผิดปกติ ออกเท่านั้น ซึ่งจะเป็นการลดความเสียหายให้แก่ระบบน้อยที่สุด

2.ลดค่าใช้จ่ายในการซ่อมแซมส่วนที่เกิดผิดปกติ

3.ลดความเสียหายไม่เกิดลุกลามไปยังอุปกรณ์อื่นๆ

4.ทำให้ระบบไฟฟ้าไม่ดับทั้งระบบเมื่อเกิดฟอลต์ขึ้นในระบบ

1.3 คุณสมบัติที่ดีของรีเลย์

1.ต้องมีความไว (Sensitivity) คือมีความสามารถในการตรวจพบสิ่งที่ผิดปกติเพียงเล็กน้อยได้

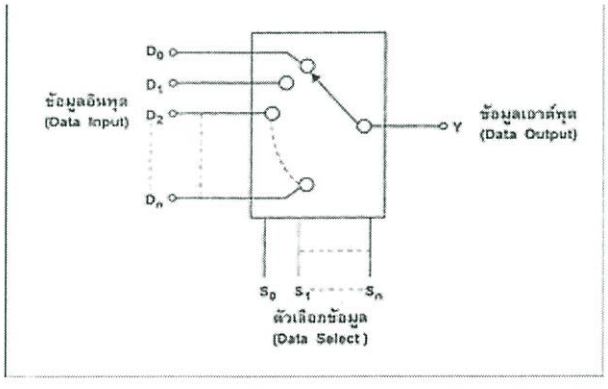
2.มีความเร็วในการทำงาน (Speed) คือความสามารถทำงานได้รวดเร็วทันใจ ไม่ทำให้เกิดความเสียหายแก่อุปกรณ์และไม่กระทบกระเทือนต่อระบบ โดยทั่วไปแล้วเวลา ที่ใช้ในการตัดวงจรจะขึ้นอยู่กับระดับของแรงดันของระบบด้วย
ระบบ 6-10 เควี จะต้องตัดวงจรภายในเวลา 1.5-3.0 วินาที

ระบบ 100-220 เควี จะต้องตัดวงจรภายในเวลา 0.15-0.3 วินาที

ระบบ 300-500 เควี จะต้องตัดวงจรภายในเวลา 0.1-0.12 วินาที

2.2.6**มัลติเพล็กซ์เซอร์ (Multiplexer : MUX)**

มัลติเพล็กซ์เซอร์ (Multiplexer : MUX) หรือเรียกอีกชื่อหนึ่งว่า ตัวเลือกข้อมูล(Data Selector) ซึ่งเป็นตัวที่ทำหน้าที่เลือกช่องสัญญาณที่มีข้อมูลช่องหนึ่งจากหลายๆช่องสัญญาณมาเป็นอินพุตและต่อช่องสัญญาณที่มีข้อมูลนั้นเข้าเป็นสัญญาณเอาต์พุตเพียงเอาต์พุตเดียว ดังรูป

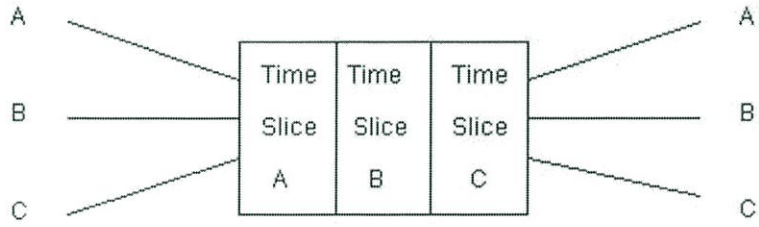


รูปที่ 2.10รูปบล๊อกไดอะแกรมของมัลติเพล็กซ์เซอร์

จากรูปทางด้านซ้ายนั้นได้แสดงอินพุตจำนวน n ขา ตั้งแต่ D0 ถึง Dn ขาซึ่งใช้สำหรับเลือกข้อมูล S0 ถึง Sm และขาเอาต์พุต Y ส่วนขาของ S0 ถึง Sm ใช้สำหรับการเลือกให้อินพุตใดจากอินพุต D0 ถึง Dn ผ่านออกไปที่เอาต์พุต Y และจำนวนขาของ S นั้นก็ยังมีความสัมพันธ์กับจำนวนขาของอินพุต D คือ เช่น ถ้ามีจำนวนอินพุตที่จะถูกเลือกจำนวน 4 อินพุต จำนวนบิตของขาสำหรับเลือกข้อมูลจะเท่ากับ 2 ($2^2 = 4$) ถ้ามีจำนวนอินพุตที่จะถูกเลือกจำนวน 8 อินพุต จำนวนบิตของขาสำหรับเลือกข้อมูลจะเท่ากับ 3 ($2^3 = 8$) เป็นต้น

ซึ่งวิธีการรวมข้อมูลจากหลายๆ จุด แล้วส่งผ่านไปตามสายส่งเพียงสายเดียวนั้น เรียกว่า **multiplex** ซึ่งสามารถแบ่งได้ออกเป็น 2 รูปแบบคือ

1) การมัลติเพล็กซ์แบบแบ่งเวลา (Time Division Multiplexer หรือ TDM) เป็นวิธีที่เพิ่งจะได้รับการพัฒนาขึ้นมาได้ไม่นานนักการมัลติเพล็กซ์แบบแบ่งเวลาจะใช้เส้นทางเพียงเส้นทางเดียวและคลื่นพาห์ความถี่เดียวเท่านั้นแต่ ผู้ใช้แต่ละคนนั้นจะได้รับการจัดสรรเวลาในการเข้าใช้ช่องสัญญาณที่ต้องการนั้นเพื่อส่งข้อมูลไปยังปลายทางที่ตนต้องการ



รูปที่ 2.11รูปแสดงการมัลติเพล็กซ์แบบแบ่งเวลา

2) การมัลติเพล็กซ์แบบแบ่งความถี่ (Frequency Division Multiplexer หรือ FDM) เป็นวิธีที่ใช้กันทั้งระบบที่มีสายและระบบคลื่นวิทยุ ซึ่งหลักการของการมัลติเพล็กซ์แบบแบ่งความถี่นี้คือการนำสัญญาณจากแหล่งต่างๆมารวมกัน ให้อยู่ในคลื่นพาห์เดียวกันที่ความถี่ต่างๆ และสัญญาณเหล่านี้สามารถที่จะใช้เส้นทางร่วมกันได้ซึ่งต่างจากแบบแรกที่ไม่สามารถใช้ร่วมกันได้

บทที่ 3

วิธีการดำเนินการ

3.1 การออกแบบหน้าเว็บสำหรับการควบคุมบอร์ด remotemux

-ส่วนที่ 1 ออกแบบให้ มี 4 channel แต่ละ channel มี 8 ช่อง โดยให้แต่ละ channel เปิดสวิตซ์ checkbox ได้เพียง 1 ช่องเท่านั้น

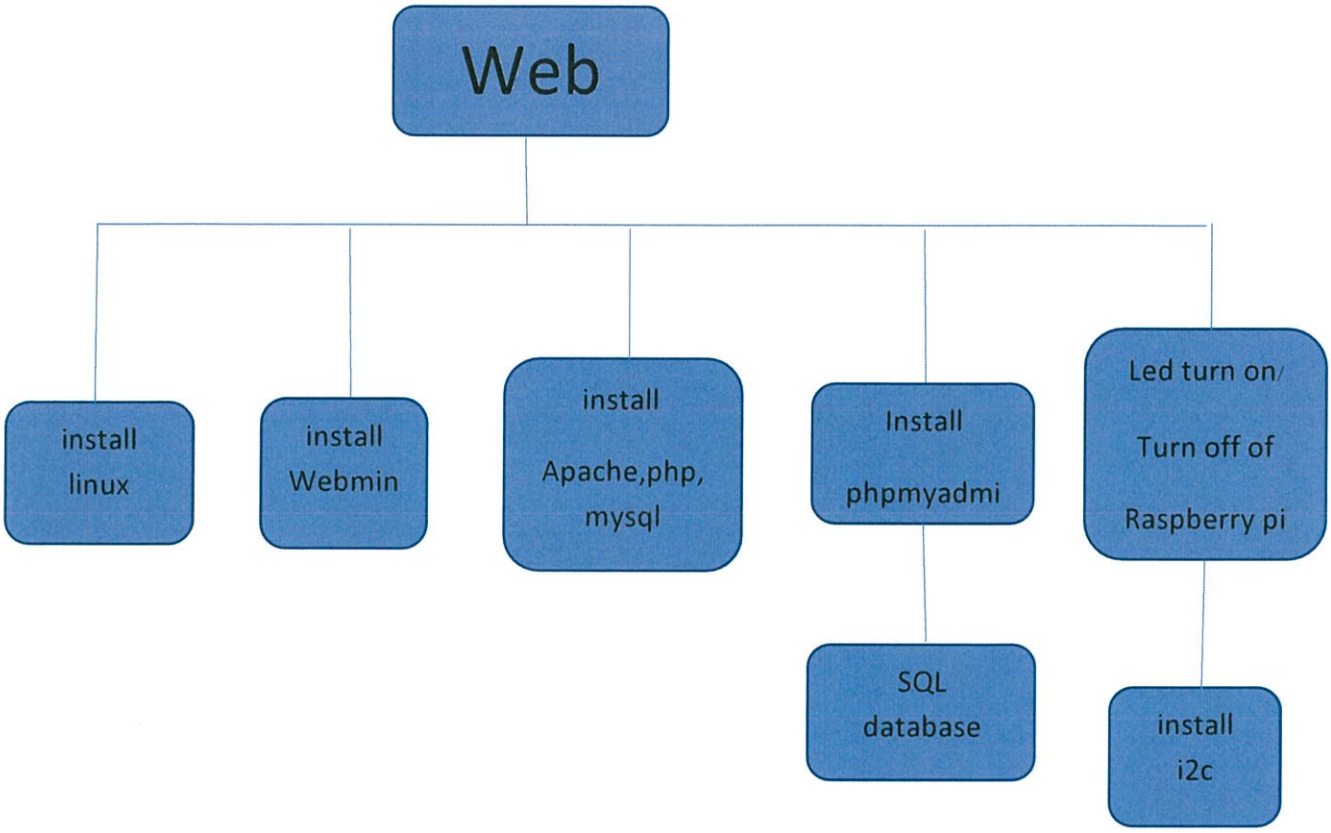
-ส่วนที่ 2 ให้ช่อง text สามารถเก็บข้อความไว้ใน sql database ได้

-ส่วนที่ 3 ทำช่องADC (การแปลงจากสัญญาณอนาลอกเป็นสัญญาณดิจิตอล)ในการวัดค่าแรงดันที่เอาท์พุท

-ส่วนที่ 4 ส่วนของเวลาในการเข้าใช้งานหน้าเว็บ

-ส่วนที่ 5 กดปุ่ม reset แล้วสวิตซ์ปิด

3.2 แผนผังการเขียน web ของ บอร์ดremotemuxลงใน raspberry pi



3.3 วิธีการลง linux

3.3.1 ติดตั้งระบบปฏิบัติการ Linux ให้กับ Raspberry Pi

- **เริ่มต้นการติดตั้งระบบปฏิบัติการ**

ก่อนเริ่มต้นการใช้งานบอร์ด Raspberry Pi จำเป็นที่จะต้องติดตั้งระบบปฏิบัติการให้กับบอร์ดก่อนเนื่องจากบอร์ดไม่มีหน่วยความจำแบบแฟลชเมมโมรี่มาบนบอร์ดด้วย ดังนั้น จำเป็นที่จะต้องเตรียมอุปกรณ์ต่างๆ ให้พร้อมเพื่อให้สามารถใช้งานบอร์ดได้ ซึ่งมีรายละเอียดอุปกรณ์ดังนี้

1. บอร์ด Raspberry Pi
2. SD Card สำหรับติดตั้งระบบปฏิบัติการ Linux ต้องมีความจุมากกว่า 2GB ขึ้นไป แต่

2. SD Card สำหรับติดตั้งระบบปฏิบัติการ Linux ต้องมีความจุมากกว่า 2GB ขึ้นไป แต่แนะนำให้ใช้ ขนาด 4GB หรือมากกว่า สำหรับคู่มือฉบับนี้จะใช้ขนาด 8GB ควรเลือกใช้การ์ดที่มีความเร็วสูงอย่าง Class 10 เพื่อประสิทธิภาพการทำงานของระบบโดยรวม

รูปที่ 3.2 Sandisk Micro SD Ultra 8GB 30MB/s Class10 with Adapter

3. เมาส์และคีย์บอร์ดแบบ USB

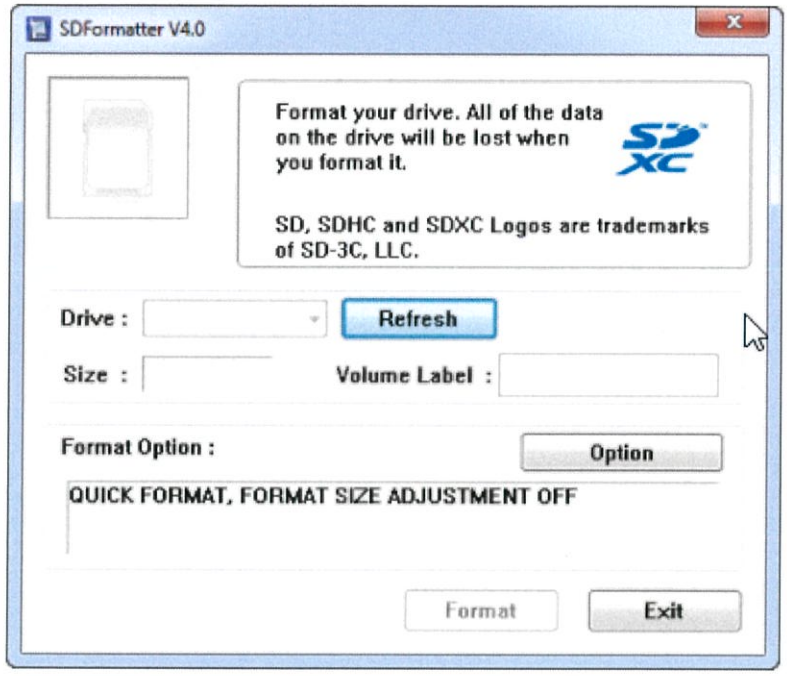
4. สาย Micro USB เพื่อจ่ายไฟเลี้ยงวงจร สามารถเลือกใช้แหล่งจ่ายไฟจากพอร์ต USB ของเครื่องคอมพิวเตอร์ได้

5. สาย HDMI เพื่อเชื่อมต่อกับจอแสดงผล หากเลือกใช้จอ Monitor ที่ไม่มีพอร์ต HDMI รองรับต้องใช้ตัวแปลง HDMI to VGA ด้วย หรือเชื่อมต่อสายวีดีโอ RCA ก็ได้เช่นเดียวกัน (เลือกอย่างใดอย่างหนึ่ง)

3.3.2 เตรียม Software สำหรับติดตั้งระบบปฏิบัติการ Linux ลงบนบอร์ด Raspberry Pi

คู่มือฉบับนี้จะจัดเตรียมซอฟต์แวร์ที่รองรับระบบปฏิบัติการ Windows 7 เป็นหลัก และต้องติดตั้งลงบนเครื่องคอมพิวเตอร์ดังนี้

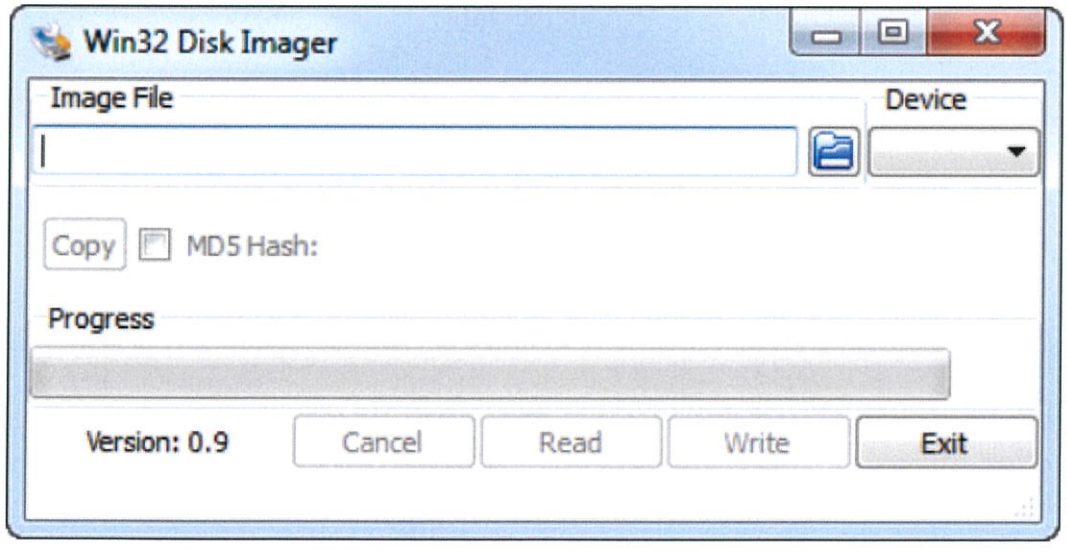
1. โปรแกรม SD Formatter 4.0 ใช้สำหรับ Format Disk สามารถดาวน์โหลดได้จากลิงค์ https://www.sdcard.org/downloads/formatter_4/eula_windows/



รูปที่ 3.3ตัวอย่างหน้าต่างโปรแกรม SD Formatter Version 4.0

2. โปรแกรม Win32 Disk Imager ใช้สำหรับเขียนไฟล์ระบบปฏิบัติการที่เป็นไฟล์ Image (*. img) ลงบน SD Card สามารถดาวน์โหลดได้จาก

ลิงค์ http://sourceforge.net/projects/win32diskimager/

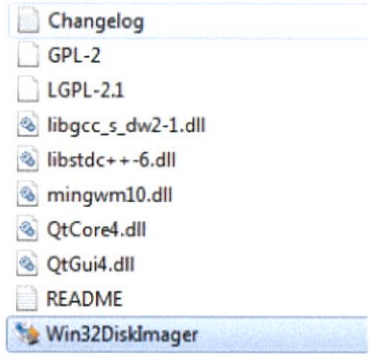


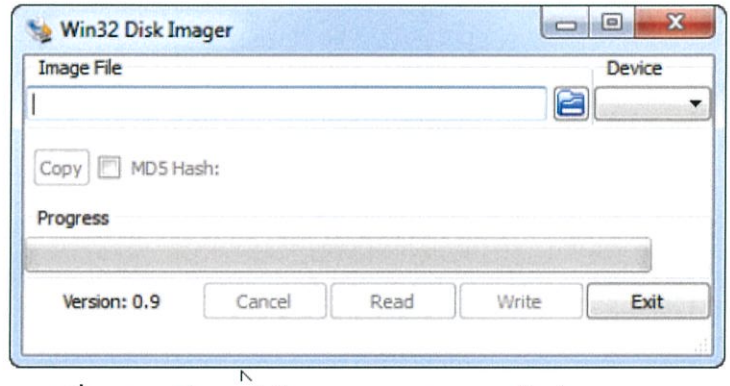
รูปที่ 3.4ตัวอย่างโปรแกรม Win32 Disk Imager

3. ไฟล์ระบบปฏิบัติการ **คู่มือนี้ติดตั้งระบบปฏิบัติการ Raspbian** เป็นระบบปฏิบัติการ Debian Wheezy ที่ถูกปรับแต่งให้ใช้สำหรับบอร์ด Raspberry Pi โดยเฉพาะ เป็น Linux ที่ให้ใช้งานได้ฟรี สามารถดาวน์โหลดได้จากลิงค์ http://www.raspberrypi.org/downloads

3.3.3 ขั้นตอนการติดตั้งระบบปฏิบัติการ Raspbian ให้กับบอร์ด Raspberry Pi

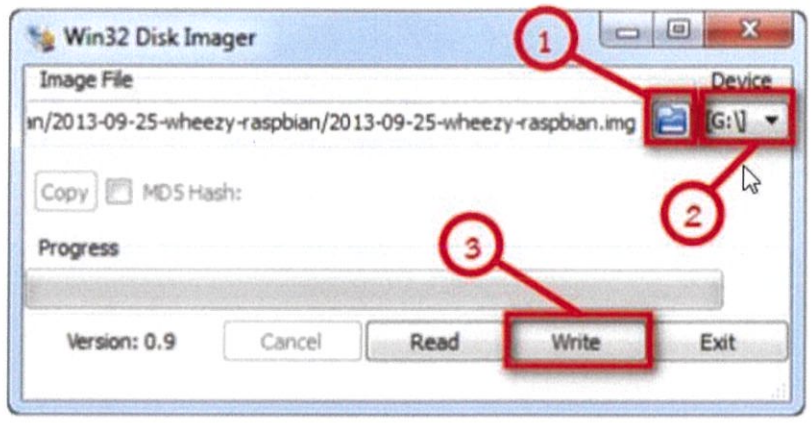
1. หากมีข้อมูลอยู่ใน SD Card ให้ทำการ Format ด้วยโปรแกรม SD Formatter 4.0 หรือโปรแกรมอื่นๆ ก็ได้ ถ้าหาก Format แล้วให้ข้ามขั้นตอนนี้ได้เลย
2. เมื่อดาวน์โหลดไฟล์ระบบปฏิบัติการ Raspbian มาแล้วจะได้เป็นไฟล์ Zip ให้แตกไฟล์จะได้เป็นไฟล์ Image (*img)
3. เมื่อดาวน์โหลดโปรแกรม Win32 Disk Imager มาแล้วจะได้เป็นไฟล์ Zip ให้แตกไฟล์และรันโปรแกรมแสดงดังรูป





รูปที่ 3.5รันไฟล์ Win32DiskImager **รูปที่ 3.6** หลังจากรันโปรแกรมจะปรากฏหน้าต่างโปรแกรม

4. ให้ Browse ไฟล์ Image ระบบปฏิบัติการ Raspbain (*img) และเลือก Device ให้ถูกต้องแล้วคลิกปุ่ม Write แสดงดังรูป และจะปรากฏหน้าต่างยืนยัน ให้คลิกปุ่ม Yes



รูปที่ 3.7write ไฟล์ลงใน SD CARD

5. รอจนกว่า Progress Bar ครบ 100% และปรากฏหน้าต่างแสดงดังรูป แล้วให้กดปุ่ม OK และ Exit

6. หากเชื่อมต่อบอร์ด Raspberry Pi กับจอคอมพิวเตอร์ผ่านอุปกรณ์แปลง HDMI-to-VGA ให้แก้ไขไฟล์ config.txt ตามเอกสารในลิงค์

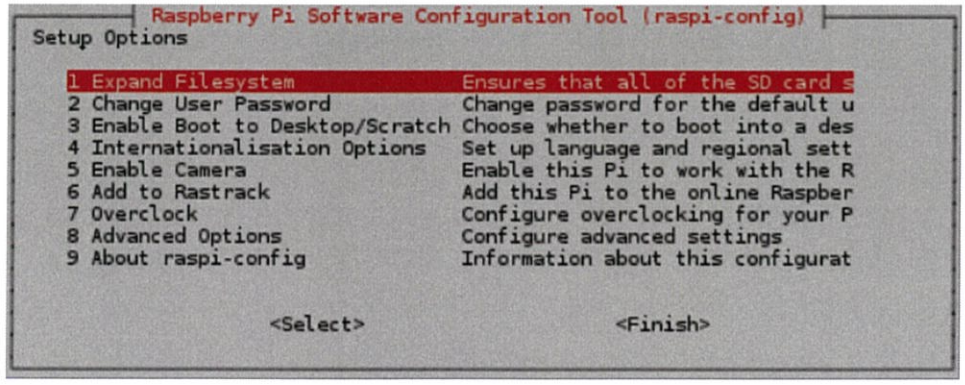
http://www.thaieasyelec.com/downloads/EACC027/HDMI_to_VGA_Cofiguration.zip

7. จากนั้นถอด SD Card ออกจากเครื่องคอมพิวเตอร์ แล้วนำไปเสียบที่บอร์ด Raspberry Pi

8. หลังจากเสียบ SD Card เรียบร้อยแล้ว ให้เสียบเมาส์ คีย์บอร์ด สายต่อจอแสดงผล HDMI หรือ RCA สายไฟเลี้ยงวงจรบอร์ด Micro USB และอื่นๆ

9. หลังจากนั้นบอร์ด Raspberry Pi ก็จะเริ่มทำงาน และเริ่ม Boot

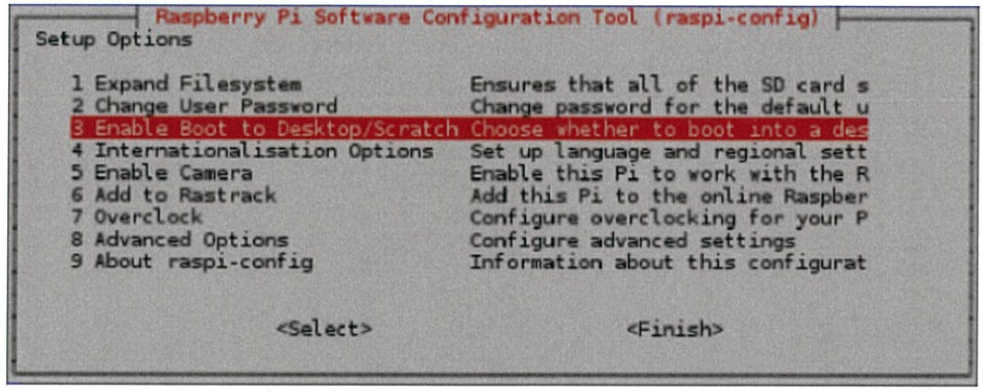
10. หลังจากระบบปฏิบัติการ Boot เสร็จเรียบร้อยจะปรากฏหน้าต่างแสดงดังรูป ให้เลือกเมนู 1 Expand Filesystem เพื่อขยายพื้นที่บน SD Card ให้ใช้งานได้เต็มความจุ เลือกด้วยลูกศรขึ้นลง แล้วกด Enter



รูปที่ 3.8รูปหลังจาก boot เสร็จเรียบร้อย

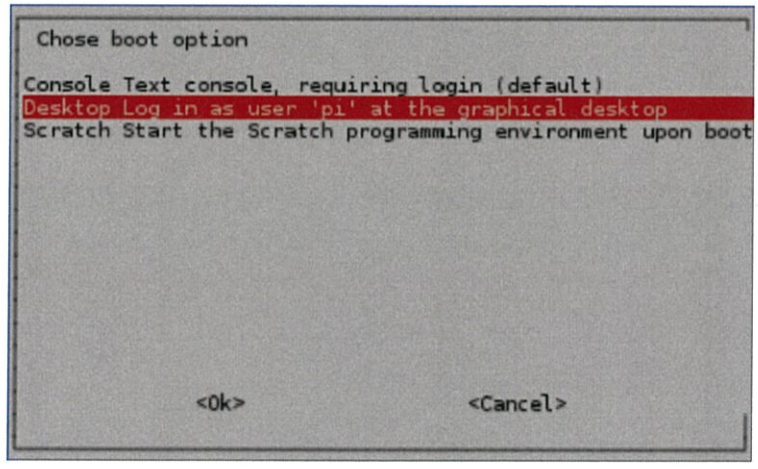
11. จะปรากฏหน้าต่างแสดง ให้กด Enter อีกครั้ง

12. กำหนดรูปแบบการใช้งานระบบปฏิบัติการให้ใช้งานในโหมด Graphic ให้เลือกเมนู 3 Enable Boot to Desktop/Scratch แล้วกดแป้นพิมพ์ Enter



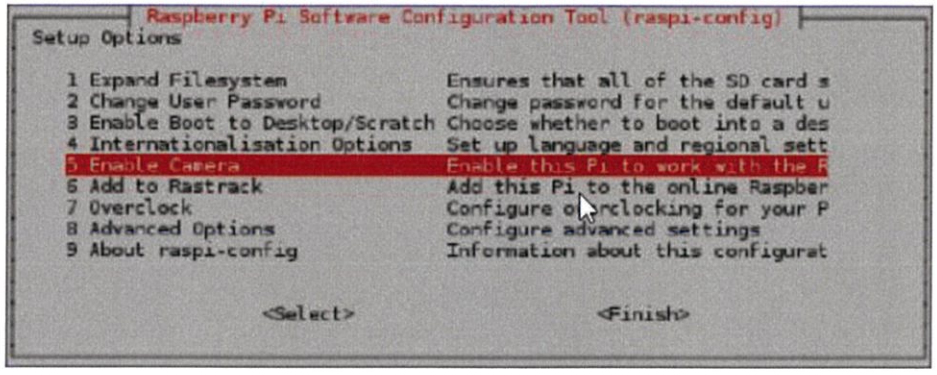
รูปที่ 3.9สำหรับ Enable Boot to Desktop/Scratch

13. หลังจากนั้นจะปรากฏตัวเลือกมาทั้งหมด 3 ตัวเลือก ให้เลือก Desktop Log in as user 'pi' at the graphical desktop แล้วกดแป้นพิมพ์ Enter



รูปที่ 3.10 เพื่อให้สารถ log in ที่ desktop ได้

14. หากต้องการใช้งานโมดูลกล้อง (Raspberry Pi Camera Module) ต้องเปิดฟังก์ชันการใช้งานนี้ด้วย โดยเลือกเมนู 5 Enable Camera แล้วกด Enter แต่ถ้าหากไม่ต้องการใช้ให้ข้ามขั้นตอนนี้ไป



รูปที่ 3.11 เปิดใช้งานโมดูลกล้อง

15. หลังจากนั้นให้เลือก Enable แล้วกดแป้นพิมพ์ Enter
16. สุดท้ายให้เลื่อนไปที่ Finish แล้วกด Enter เพื่อจบการตั้งค่าระบบและ Reboot ระบบใหม่
17. ระบบจะถามย้ำอีกครั้งว่าต้องการจะ Reboot ระบบใหม่ตอนนี้เลยหรือไม่ เลือก Yes แล้วกด Enter ระบบก็จะ Reboot ใหม่ทันที
18. หลังจากที่ระบบ Reboot ใหม่เสร็จเรียบร้อยแล้ว จะแสดงหน้า Desktop ของ Raspbian

3.4สรุปวิธีการลง linux

- โหลด win32
- โหลด Raspbian jessie zip
- เปิด win32 แล้วเลือกไฟล์ Raspbian jessie zip
- เสียบ SD card แล้วก็กด write เพื่อ write งานลง SD card
- เข้าไปที่ boot folder แล้วสร้างไฟล์ชื่อ ssh โดยไม่ต้องมีนามสกุล
- นำไฟล์ wpa-supplicant.conf มาลงใน boot
- เข้า putty เข้าระบบ putty ด้วย username:pi / password: pi
- ใช้คำสั่ง passwd ในการเปลี่ยน password
- วิธีเปลี่ยน hostname

➔ ใช้คำสั่ง : sudo nano /etc/hosts (เปลี่ยนคำว่า raspberry pi เป็น remoetmux0)แล้วก็กด ctrl+x

➔ ใช้คำสั่ง : sudo nano /etc/hostname (เปลี่ยนคำว่า raspberry pi เป็น remoetmux0)

➔ พิมพ์ sudo /etc/init.d/hostname.sh

➔ พิมพ์ sudo reboot

3.5 วิธีการติดตั้ง webmin

- เข้าไปที่ www.webmin.com/deb.html
- พิมพ์ใน putty
  - ➔ Wget http://prdownloads.sourceforge.net/webadmin/webmin_1.860_all.deb
  - ➔ sudo dpkg--install webmin_1.860_all.deb
  - ➔ sudo apt-get install perl libnet-ssleay-perl openssl libauthen-pam-perl libpam-runtime libio-pty-perl apt-show-versions python
  - ➔ sudo nano /etc/apt/sources.list
  - ➔ แล้วพิมพ์ `deb http://download.webmin.com/download/repository sarge contrib`
  - ➔ cd /root
  - ➔ `wget http://www.webmin.com/jcameron-key.asc`
  - ➔ `apt-key add jcameron-key.asc`
  - ➔ `apt-get update`
  - ➔ `apt-get install apt-transport-https`
  - ➔ `apt-get install webmin`
- เข้า webmin ได้ที่ https://remotemux0:10000

3.6วิธีติดตั้ง LAMP(linux,apache,php,mysql)

- เข้าเว็บ http://www.php5dp.com/get-mysql-and-php-to-work-together-in-respberry-pi/
- พิมพ์ใน putty
  - ➔ sudo apt-get update
  - ➔ sudo apt-get install apache2 php5 php5-mysql mysql-sever

3.7วิธี install phpmyadmin raspberry pi

- พิมพ์ใน putty
  - ➔ sudo bash
  - ➔ sudo apt-get install phpmyadmin (เลือก apache2)
  - ➔ nano /etc/apache2/apache2.conf
  - ➔ ctrl+v (jump page) แล้วเพิ่ม Include /etc/phpmyadmin/apache.conf เข้าไป
  - ➔ /etc/init.d/apache2 restart
  - ➔ มี username : root / password : winfast2

3.8 วิธีทำให้ LED ที่บอร์ด remotemux ติด

- พิมพ์ใน putty ใช้คำสั่ง sudo nano turn-on.py
  - ➔ Import Rpi.GPIO as GPIO
    GPIO.setwarnings(False)
    GPIO.setmode(GPIO.BCM)
    GPIO.setup(13,GPIO.OUT)
    GPIO.output(13,true)
  - ➔ sudo apt-get install python-dev
  - ➔ sudo apt-get install python-rpi.gpio
  - ➔ python turn-on.py

- พิมพ์ใน putty ใช้คำสั่ง sudo nano turn-off.py

  ➔ Import Rpi.GPIO as GPIO

  GPIO.setwarnings(False)

  GPIO.setmode(GPIO.BCM)

  GPIO.setup(13,GPIO.OUT)

  GPIO.output(13,False)

  ➔ sudo i2cdetect -y 1 (เช็ค i2c)

  ➔ sudo nano /etc/network/interface

  ➔ post-up python /var/www/html/turn-on.py

  ➔ sudo cd turn-on.py /var/www/html

  ➔ cd /var/www/html

  ➔ ls

  ➔ sudo reboot

  ➔ nano relay_control.php

  ➔ ls -1

  ➔ sudo chmod 777 adc.py

  ➔ sudo chmod 777 relay_control.py

  ➔ ls -1

  ➔ nano relay_control.php

- check LED ที่บอร์ด

  ➔ cd /var/www/html/test1

  ➔ python relay.control.py 33 0 0 (ex 32 0 254)

➔ cd /var/www/html/test1

➔ python relay.control.py 33 0 0 (ex 32 0 254)

3.9วิธี ลง i2c

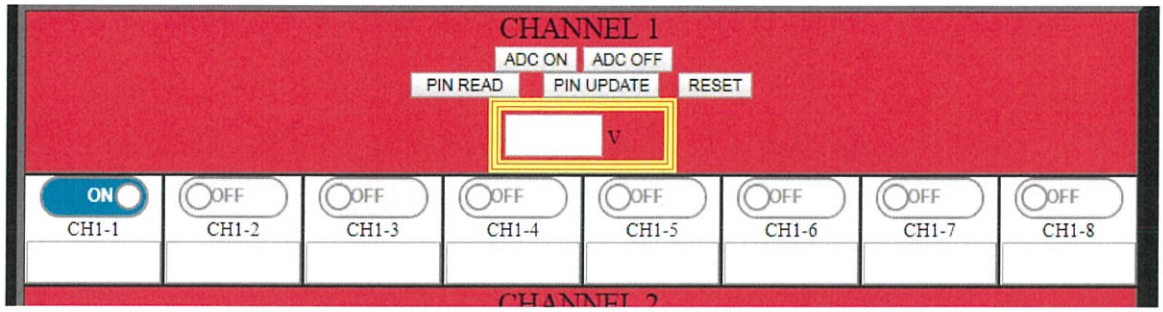
- https://learn.adafruit.com/adafruits-raspberry-pi-lesson-4-gpio-setup/configuring-i2c
- พิมพ์ใน putty
    - ➔ sudo apt-get install -y python-smbus
    - ➔ sudo apt-get install -y i2c-tools
    - ➔ sudo raspi-config (interfacing option -> i2c)
- ให้ led สั่งงานด้วย web ได้
    - ➔ sudo visudo (add : www-data ALL=(ALL) NOPASSWD : ALL)

บทที่ 4

การทดลองและผลการทดลอง

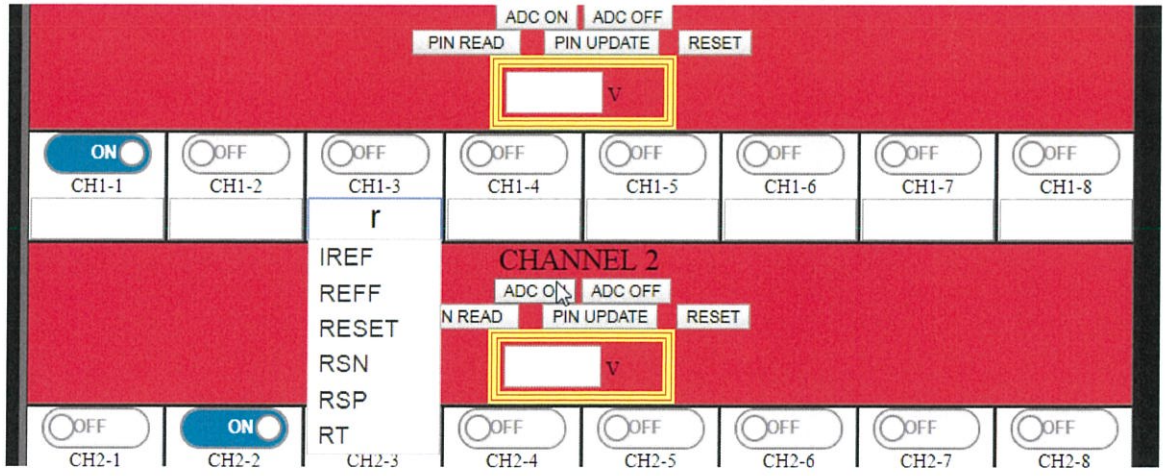
4.1 การทดลองและผลการทดลอง การสั่งงานด้วยเว็บของ บอร์ด remotemux

4.1.1 ให้แต่ละchannel เปิดได้ทีละ CH



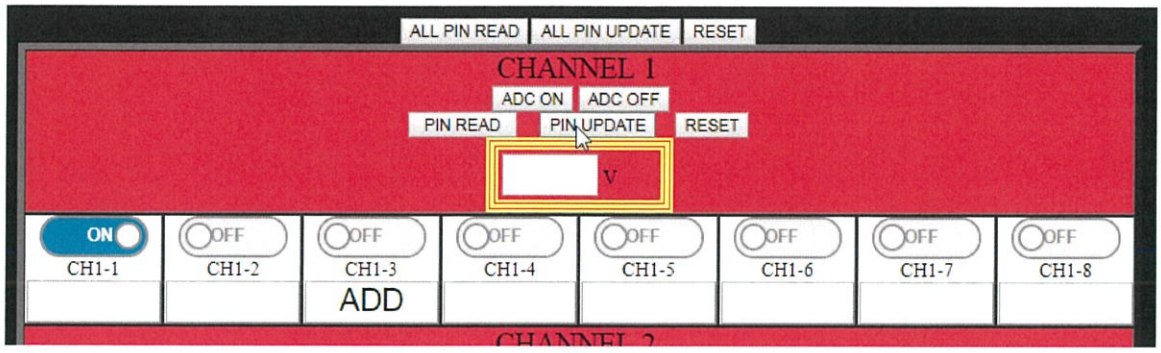
รูปที่ 4.1เปิดCH ได้ทีละสวิตช์เท่านั้น

4.1.2 มีข้อความในช่อง text ขึ้นมาอัตโนมัติตามคำที่เราตั้งไว้



รูปที่ 4.2 ข้อความใน text ขึ้นตามตัวอักษร

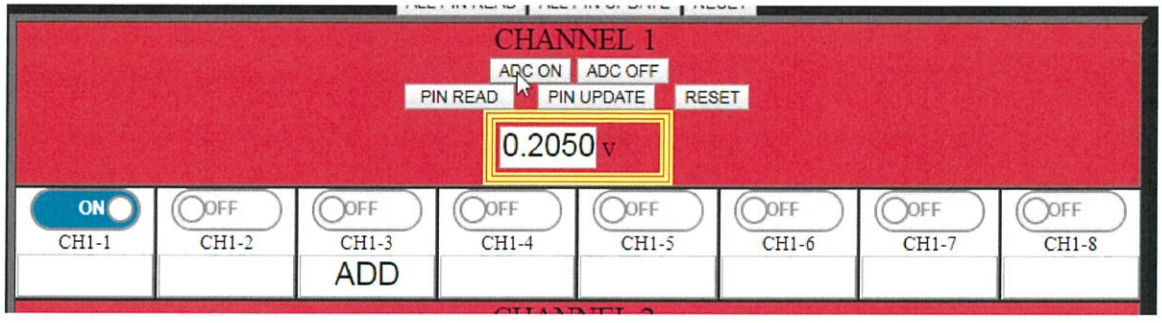
4.1.3 สามารถเพิ่มข้อความใน text ไปเก็บใน sql database ได้ โดยกดปุ่ม PIN UPDATE เพื่อเอาข้อความไปเก็บใน SQL แล้วกด PIN READ เพื่ออ่านค่า



รูปที่ 4.3 เก็บข้อความไว้ใน sql database

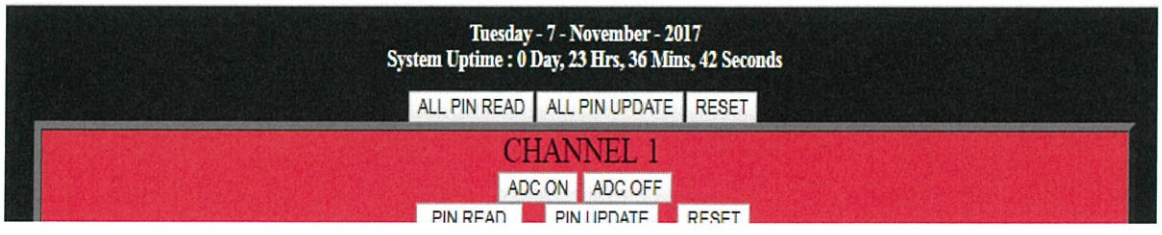
4.1.4 กดปุ่ม reset แล้วสวิตช์ปิด

4.1.5 สามารถวัดแรงดันที่เอาท์พุทได้โดยกดปุ่ม ADC ON /ADC OFF โดยปรากฏที่ช่องแสดง



รูปที่ 4.4 ช่องดูค่าแรงดันเอาท์พุท ADC

4.1.6 ส่วนของเวลาการใช้งานของหน้าเว็บ



รูปที่ 4.5 ส่วนของเวลาการใช้งานเว็บ

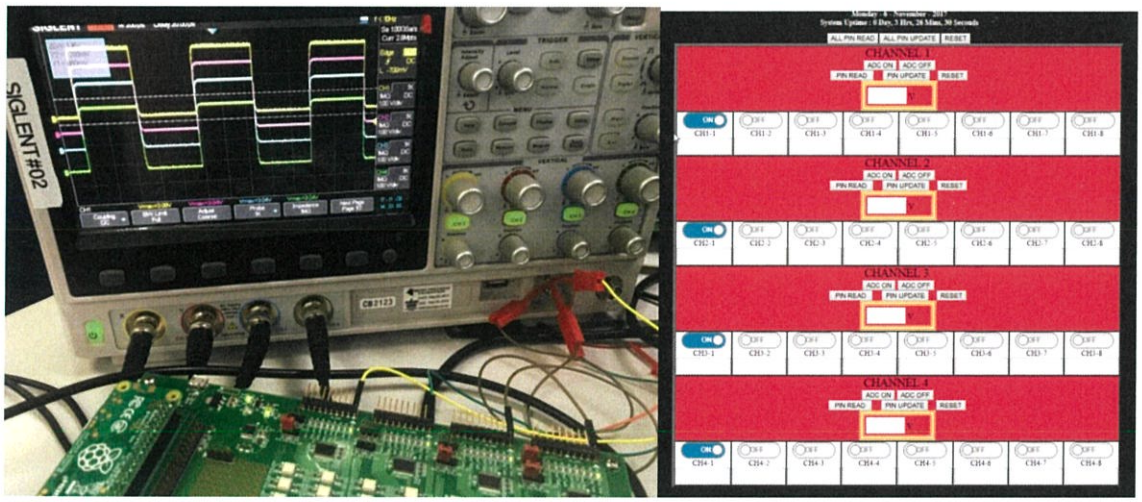
4.2 คุณสมบัติของบอร์ด remotemux

-สามารถวัดแรงดันที่เอาท์พุทได้

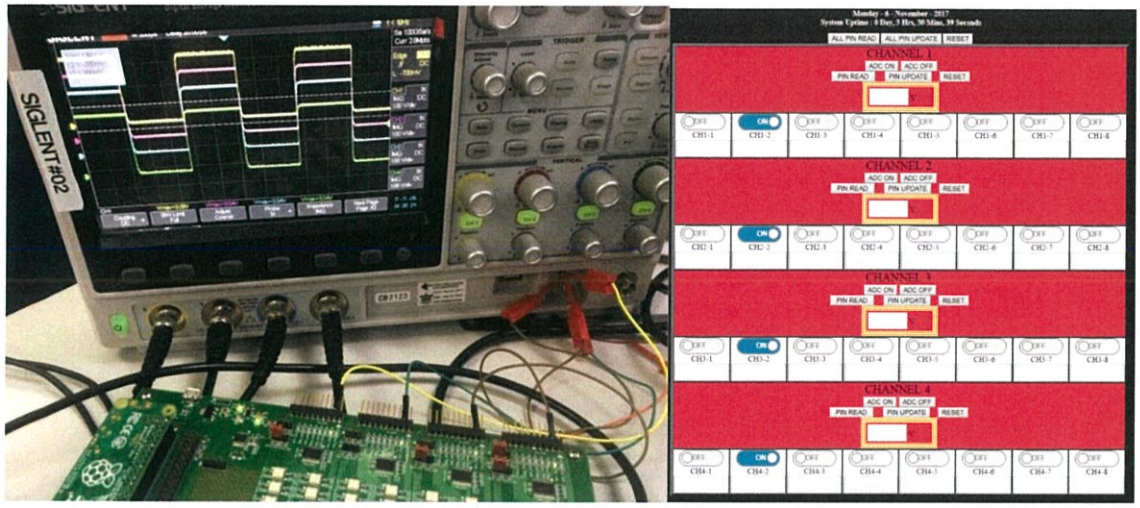
- noise อยู่ในค่าที่รับได้

- ระหว่างการสลับสวิตซ์ไม่มีการชนกันของกระแสและแรงดัน

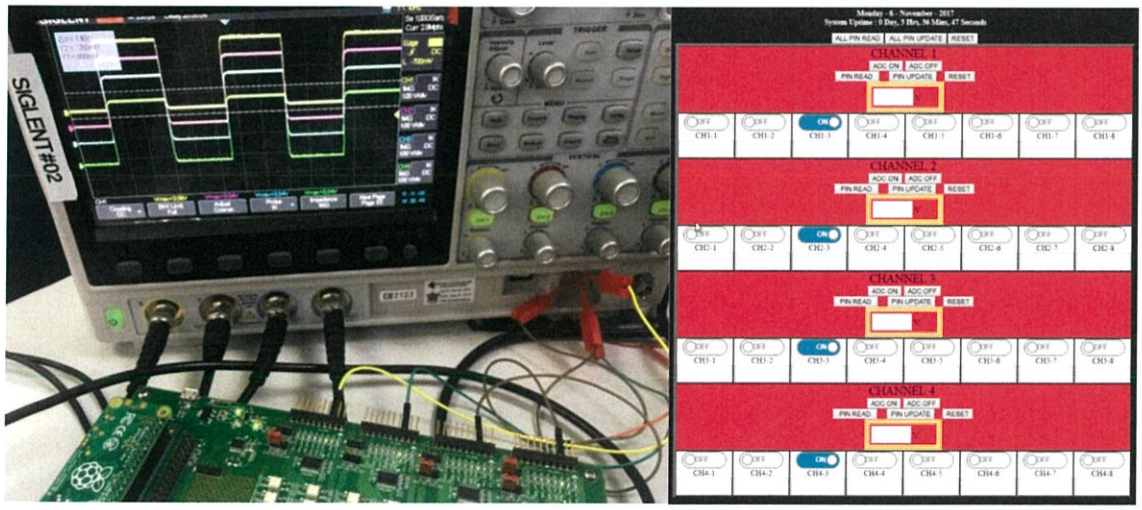
Channel 1 , Channel 2 , Channel 3 Channel 4



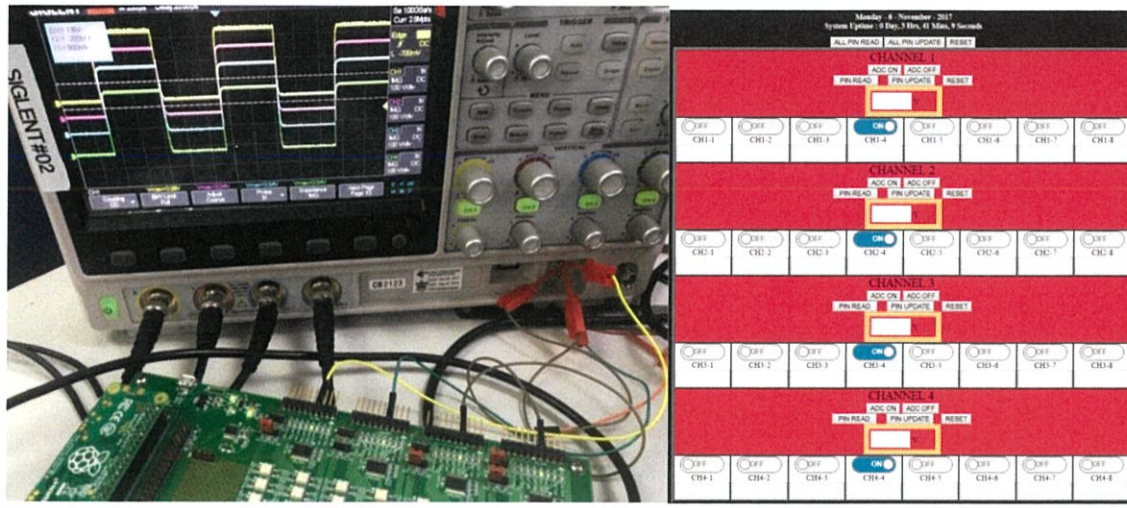
รูปที่ 4.6 วัดแรงดันที่เอาท์พุทที่ CH1 ของทั้ง 4 channelsและหน้าเว็บ



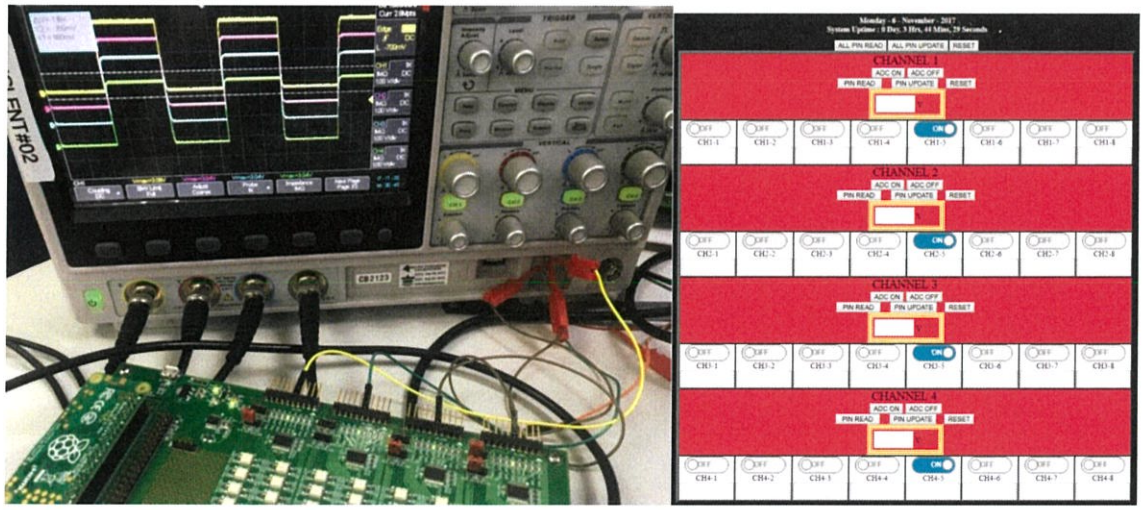
รูปที่ 4.7 วัดแรงดันที่เอาท์พุทที่ CH2 ของทั้ง 4 channelsและหน้าเว็บ



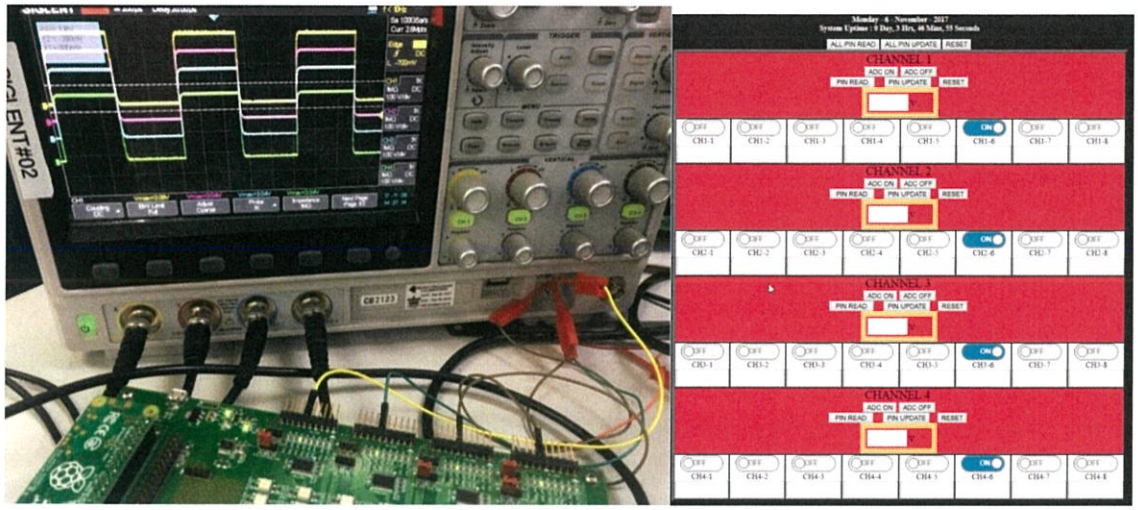
รูปที่ 4.8 วัดแรงดันที่เอาท์พุทที่ CH3 ของทั้ง 4 channels และหน้าเว็บ



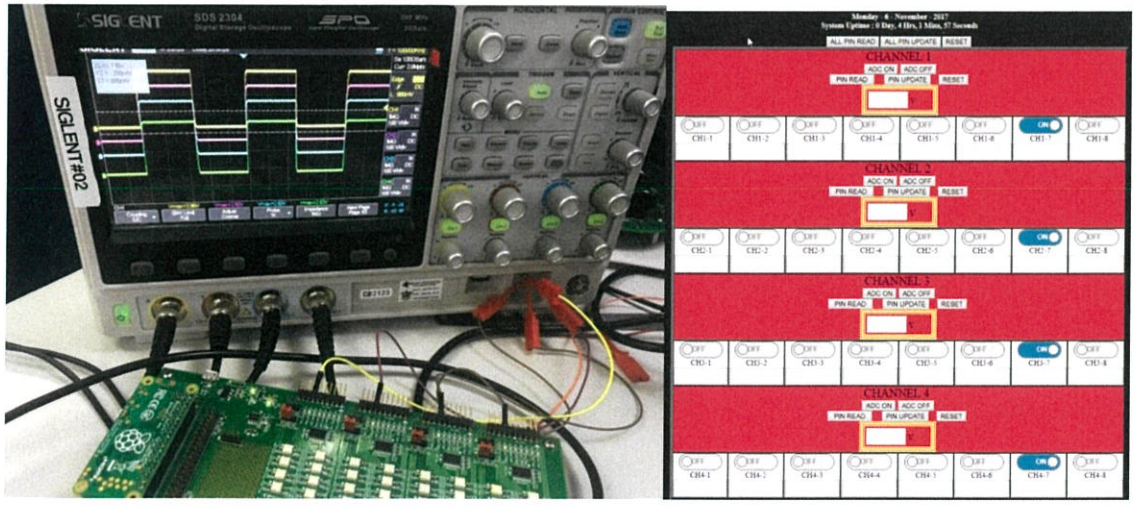
รูปที่ 4.9 วัดแรงดันที่เอาท์พุทที่ CH4ของทั้ง 4 channels และหน้าเว็บ



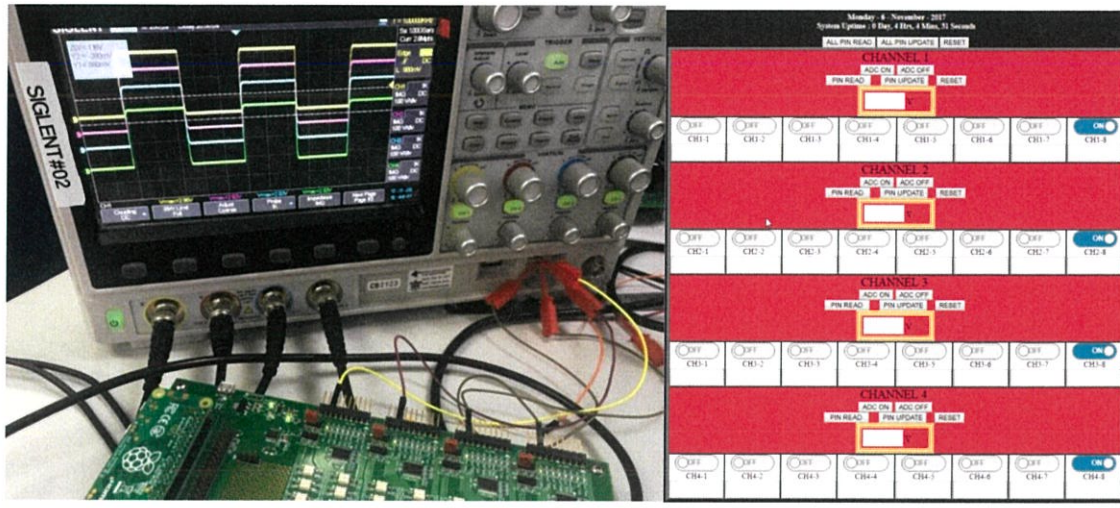
รูปที่ 4.10 วัดแรงดันที่เอาท์พุทที่ CH5 ของทั้ง 4 channels และหน้าเว็บ



รูปที่ 4.11 วัดแรงดันที่เอาท์พุทที่ CH6 ของทั้ง 4 channels และหน้าเว็บ



รูปที่ 4.12 วัดแรงดันที่เอาท์พุทที่ CH7 ของทั้ง 4 channels และหน้าเว็บ



รูปที่ 4.13วัดแรงดันที่เอาท์พุทที่ CH8ของทั้ง 4 channels และหน้าเว็บ

บทที่ 5

สรุปผลการวิจัยและข้อเสนอแนะ

จากการทดลองรีโมทระหว่างบอร์ด remotemux กับ เว็บเพจสามารถสรุปได้ดังนี้

การใช้งานเว็บนั้นใช้งานได้ 4 channels โดยแต่ละ channel เปิดสวิตซ์ได้เพียงอันเดียวเท่านั้นจาก 8 CH เท่านั้น เราจะวัดแรงดันที่เอาท์พุทได้เพื่อให้ดูง่ายโดยเราจะเปลี่ยนสัญญาณอนาลอกเป็นสัญญาณดิจิตอลกดที่ปุ่ม ADC ON/ADC OFF และสามารถเก็บค่า SQL database ซื่อขาที่เราวัดค่ามาไว้ที่ช่อง text ให้กดปุ่ม pin updateให้กดปุ่ม pin read เพื่ออ่านค่าที่เก็บไว้ใน SQL databaseมีปุ่ม reset เพื่อปิดสวิตซ์ และยังสามารถดูว่าเราใช้งานเว็บไปนานเท่าไหร่อีกด้วย

เอกสารอ้างอิง

https://www.raspberrypi.org/documentation/installation/installing-images/

http://www.webmin.com/deb.html

https://www.stewright.me/2012/09/tutorial-install-phpmyadmin-on-your-raspberry-pi/

https://makezine.com/projects/tutorial-raspberry-pi-gpio-pins-and-python/

https://learn.adafruit.com/adafruits-raspberry-pi-lesson-4-gpio-setup/configuring-i2c

ภาคผนวก

โค้ดสำหรับการทำงานของหน้าเว็บในแต่ละส่วน

โค้ดของ adc ภาษา python

```
#!/usr/bin/python


import smbus

import sys

import time



bus = smbus.SMBus(1)

bus.write_byte_data(0x33, int(sys.argv[1]), int(sys.argv[2]))

time.sleep(0.1)

#print hex(bus.read_word_data(0x33,1))

print bus.read_word_data(0x33,int(sys.argv[1]))
```

โค้ดของ adc ภาษา php

```
<?php

if (!empty($_GET[ch])) {

    $ch = $_GET[ch];
```

```
}
else{
    $ch = 1;
}
$bit1 = 0x61 + ($ch-1)*2;
//0x61; //CH1
//0x63; //CH2
//0x65; //CH3
//0x67; //CH4
$python_path = "/var/www/html" .dirname($_SERVER['SCRIPT_NAME']) .
"/adc.py";
//echo $python_path;
$ADC_OUT = exec("sudo python $python_path $bit1 210");
$temp1  = ($ADC_OUT & 0x0F) << 8;
$temp2 = ($ADC_OUT & 0xFF00) >> 8;
$FinalValue = $temp1+$temp2;
echo $FinalValue * 70 / 4096;
?>
```

โค้ดสำหรับให้สวิตซ์ทั้ง **channel** เปิดได้ **CH** เดียว

```
//chechbox
    $(function() {
```

```
        $(".onoffswitch-checkbox").click(function() {
            if ($(this).prop("checked") == true) {
                var indexObj = $(this).index(".onoffswitch-checkbox");
                CH1_LED_ON = indexObj + 1;

                var xhttp = new XMLHttpRequest();
xhttp.open("GET", PHP_RELEY_CONTROL_URL + "?ch=32&led=" +
CH1_LED_ON + "&adc=" + CH1_ADC_Status, true);
xhttp.send();
                $(".onoffswitch-checkbox").not(":eq(" + indexObj +
")").prop("checked", false);
            } else {
                var xhttp = new XMLHttpRequest();
xhttp.open("GET", PHP_RELEY_CONTROL_URL + "?ch=32&led=0&adc="
+ CH1_ADC_Status, true);
xhttp.send();
                CH1_LED_ON = 0;

            }
        });
    });
```

โค้ดสำหรับ **SQL**

```
var sql_ch = 0;


    function CH1_SQL() {
        sql_ch = 0;
SQLR();
    }
    //Read all CH
    function SQLR() {

        for (i = 8 * sql_ch + 1; i <= 8 * sql_ch + 8; i++) {
            var xhttp = new XMLHttpRequest();



xhttp.onreadystatechange= function() {
                if (this.readyState== 4&& this.status == 200) {
                $("#tags"+i).val(this.responseText);
                }
            };


xhttp.open("GET", "mydb.php?des=" + i + "&read=1", false);
xhttp.send();
```

```
    }

}

//Update SQL
function SQLW() {

    var xhttp = new XMLHttpRequest();
    var get_url = "";

    for (i = 1; i <33; i++) {

        //if ($("#inputtext" + i).val() === "") {
        //      get_url = get_url + "K" + i;
        //} else {
            get_url = get_url + $("#tags" + i).val();
        //}

        if (i <32) get_url = get_url + ",";
    }
```

```
        //alert(get_url);


    xhttp.onreadystatechange= function() {};
    xhttp.open("GET", "mydb.php?des=1&relay=" + get_url, true);
    xhttp.send();


    }
```

โค้ดสำหรับการ reset สวิตซ์และปิด LED

```
    function CH1_RESET(){
        $(".onoffswitch-checkbox").attr('checked',false);
        var xhttp = new XMLHttpRequest();
    xhttp.open("GET", PHP_RELEY_CONTROL_URL + "?ch=32&led=0&adc="
    + CH1_ADC_Status, true);
    xhttp.send();
        CH1_LED_ON = 0;
    }
```

โค้ดส่วนของ ADC

```
    function CH1_ADC_ON() {
        var xhttp = new XMLHttpRequest();
        CH1_ADC_Status = 1;
```

```
xhttp.open("GET", PHP_RELEY_CONTROL_URL + "?led=" +
CH1_LED_ON + "&ch=32&adc=" + CH1_ADC_Status, true);
xhttp.send();
    }

    function CH1_ADC_OFF() {
        var xhttp = new XMLHttpRequest();
        CH1_ADC_Status = 0;
xhttp.open("GET", PHP_RELEY_CONTROL_URL + "?led=" +
CH1_LED_ON + "&ch=32&adc=" + CH1_ADC_Status, true);
xhttp.send();
    }
```

ส่วนที่แสดงหน้าเว็บ html

```
<center>
<h2><div id="Date" ></div><div id="uptime"></div></h2>
<button id="all_sql_read" onclick="ALL_SQL();" >ALL PIN READ </button>
<button id="all_sql_write" onclick="SQLW();">ALL PIN UPDATE </button>
<button
onclick="CH1_RESET();CH2_RESET();CH3_RESET();CH4_RESET();">RES
ET</button>
<br>
```

```
<TABLE border=5  width="40%"  >

<TR>

<!--Row 1 -->

<TD align=center colspan=8 BGcolor=#FF3366>

<font size=5> CHANNEL 1</font>

<br>

<button onclick="CH1_ADC_ON();ADC1();"> ADC ON </button><button
onclick="CH1_ADC_OFF()"> ADC OFF </button><br>


<button  class=onoffswitch  align=left   onclick="CH1_SQL()">PIN
READ</button>   

<button onclick="SQLW();"> PIN UPDATE </button>  

<button onclick="CH1_RESET()">RESET</button>

<table width=17% cellpadding="4" cellspacing="0" style="border:double5px
#FFFF99;">

<tr><br>

<td style="border:double5px #FFFF99;">

<input type=text size= 5  style="width:80px;height:35px;text-align:center;font-
size:25"  id="adc_show1" > V </td>

</tr>

</table>
```

```
</TD>

</TR>

<TR>


<div>CH1-1</div>


<div class="ui-widget">


                                        <input type= text id="tags1"
style="text-align: center;width:110px;height:35px;font-size:25;" class="ui-
widget" onChange="javascript:this.value=this.value.toUpperCase();" >

                                                </div>

</TD>

<TD align=center style="cursor:pointer" BGcolor =#FFFFFF  >

<div class="onoffswitch">

<input type="checkbox" name="onoffswitch2"    class="onoffswitch-checkbox"
id="myonoffswitch2">

<label class="onoffswitch-label" for="myonoffswitch2">

<span class="onoffswitch-inner"></span>

<span class="onoffswitch-switch"></span>

</label>

</div>
```

โค้ดส่วนของข้อความคำอัตโนมัติในช่อง text

```
var availableTags =
["BATT","BIAS","BST","COMP","CSN","CSP","DH","DL","EN","FAULT","FB","I
N","IREF","LX","OUT","OSC","PGOOD","PV","REFF","RESET","RSN","RSP",
"RT","SCL","SDA","SET","SPS","SUP","SUPSW","SYNC","VCC"];



   var getSQL = function (query, callback) {
    var req = new XMLHttpRequest();
req.onreadystatechange= function () {
        if (req.readyState== 4&& req.status == 200) {
            callback(req.responseText);
        }
    };
    var param = "?q=" + encodeURIComponent(query);
req.open("GET", "yourdb.php" + param, true);
req.send();
}



function myGetSQL(){


getSQL('x', function(json_expr){
```

```
    availableTags = JSON.parse(json_expr);
        alert(availableTags);
});



}

$( function() {
$("#tags1,#tags2,#tags3,#tags4,#tags5,#tags6,#tags7,#tags8,#tags9,#tags10
,#tags11,tags12,#tags13,#tags14,#tags15,#tags16,#tags17,#tags18,#tags19,
#tags20,#tags21,#tags22,#tags23,#tags24,#tags25,#tags26,#tags27,#tags28
,#tags29,#tags30,#tags31,#tags32" ).autocomplete({
    source: availableTags
  } );
 } );
```

โค้ดสำหรับ **SQL** ภาษา **php**

```
<?php
include 'DB.php';
$host = "localhost";
```

```
$user = "root";
$pass = "winfast2";
$databaseName = "win";
$tableName     = "SWITCH";
if (!empty($_GET[des])) {
    $led = $_GET[des];
}
if(!empty($_GET[read])){
    //--------------------------------------------------------------------
    // 1) Connect to mysql database
    //--------------------------------------------------------------------
    $con = mysql_connect($host, $user, $pass);
    $dbs = mysql_select_db($databaseName, $con);

    //--------------------------------------------------------------------
    // 2) Query database for data
    //--------------------------------------------------------------------
    $result = mysql_query("SELECT DESCRIPTION FROM
$tableName where ID=$led"); //query
    $array  =mysql_fetch_row($result); //fetch result

    //--------------------------------------------------------------------
```

```
    // 3) echo result as json
    //--------------------------------------------------------------------------
    //echo json_encode($array);
    echo $array[0];

    mysql_close($con);



}
if (!empty($_GET[update])) {
        $name = $_GET['update'];
                //echo $name;
                //echo "<br>";
        $con = mysql_connect($host, $user, $pass);
        $dbs = mysql_select_db($databaseName, $con);

        $result = mysql_query("UPDATE SWITCH SET
DESCRIPTION=$name WHERE ID=$led"); //query
                //echo $result;

        mysql_close($con);
```

```
}
if (!empty($_GET[relay])) {
        //$ch =split($_GET[relay],",");
        $ch =explode(',', $_GET[relay]);
        $con = mysql_connect($host, $user, $pass);
        $dbs = mysql_select_db($databaseName, $con);

        $i=1;
foreach($ch as $a){
            $result = mysql_query("UPDATE SWITCH SET
DESCRIPTION='$a' WHERE ID=$i"); //query
            $i++;
if($i > 32) goto for_exit;
        }
for_exit:
          mysql_close($con);


}
?>
```

โค้ดสำหรับการควบคุมรีเลย์ ภาษา **php**

```
<?php

if (!empty($_GET[led])) {
    $LED = $_GET[led];
} else {
    $LED = 0;
}

if (!empty($_GET[adc])) {
    if ($_GET[adc] != 0) {
        $ADC = 1;
    }
} else {
    $ADC = 0;
}

if (!empty($_GET[ch])) {
    $ch = $_GET[ch];
} else {
    $ch = 32;
}
```

```
$bit1        = 255 - (1 <<($LED - 1));

$bit2        = 255 - $ADC;


$python_path = "/var/www/html"
.dirname($_SERVER['SCRIPT_NAME']) . "/relay_control.py";


//echo $python_path . $ch .$bit1 .$bit2;


exec("sudo python $python_path $ch $bit1 $bit2");


?>
```

โค้ดสำหรับควบคุมรีเลย์ ภาษา **python**

```
#!/usr/bin/python


import smbus

import sys

import time

#import RPi.GPIO as GPIO
```

```
bus = smbus.SMBus(1)    # 0 = /dev/i2c-0 (port I2C0), 1 = /dev/i2c-1
(port I2C1)

DEVICE_ADDRESS = int(sys.argv[1]) #0x20     #7 bit address (will
be left shifted to add the read write bit)

DEVICE_REG_MODE1 = int(sys.argv[2])



#Write a single register

time.sleep(1)

bus.write_byte_data(DEVICE_ADDRESS, DEVICE_REG_MODE1,
int(sys.argv[3]))
```

โค้ดสำหรับการเปิด/ปิด **led** ที่บอร์ด

```
import RPi.GPIO as GPIO

GPIO.setwarnings(False)

GPIO.setmode(GPIO.BCM)

GPIO.setup(13,GPIO.OUT)

GPIO.output(13,true)
```

โค้ดสำหรับเวลาการใช้งานหน้าเว็บ

```
<?php
```

```
$str   = @file_get_contents('/proc/uptime');

$num   = floatval($str);

$secs  =round(fmod($num, 60)); $num = (int)($num / 60);

$mins  =$num % 60;       $num = (int)($num / 60);

$hours = $num % 24;      $num = (int)($num / 24);

$days  =$num;


echo "System Uptime : " .  $days . " Day, " . $hours . " Hrs, " . $mins . "
Mins, " . $secs . " Seconds";


?>
```