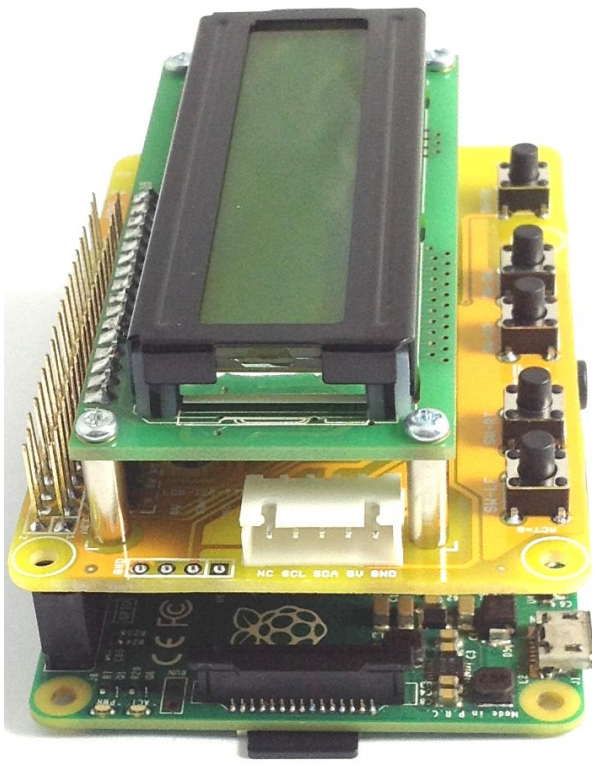
ET- LCD SW HAT

ET-LCD SW HAT เป็นบอร์ด I/O ที่ได้ออกแบบมาสำหรับใช้เชื่อมต่อกับบอร์ด Raspberry Pi ทางขั้วต่อ 40 Pin ซึ่งตัวบอร์ดจะประกอบไปด้วยอุปกรณ์ LCD 16x2 ,SW , Buzzer , RTC DS3231 พร้อมขั้วต่อ I2C เวลาใช้งานผู้ใช้สามารถนำบอร์ด ET-LCD SW HAT ไปเสียบไว้บนบอร์ด Raspberry Pi ได้เลย จากนั้นก็ฝึกเขียนโปรแกรมควบคุมอุปกรณ์ I/O ที่อยู่บนบอร์ดได้เลยโดยไม่ต้องต่อวงจรอะไรให้ยุ่งยาก ตำแหน่งขา I/O ที่จะใช้ควบคุมอุปกรณ์บนบอร์ดสามารถดูได้จากวงจรที่อยู่ท้ายคู่มือ รวมทั้งตัวอย่างการควบคุมอุปกรณ์ที่อยู่บนบอร์ดก็มีให้ในCD ซึ่งจะเขียนด้วยภาษา Python สามารถ Copy ไปทดสอบได้

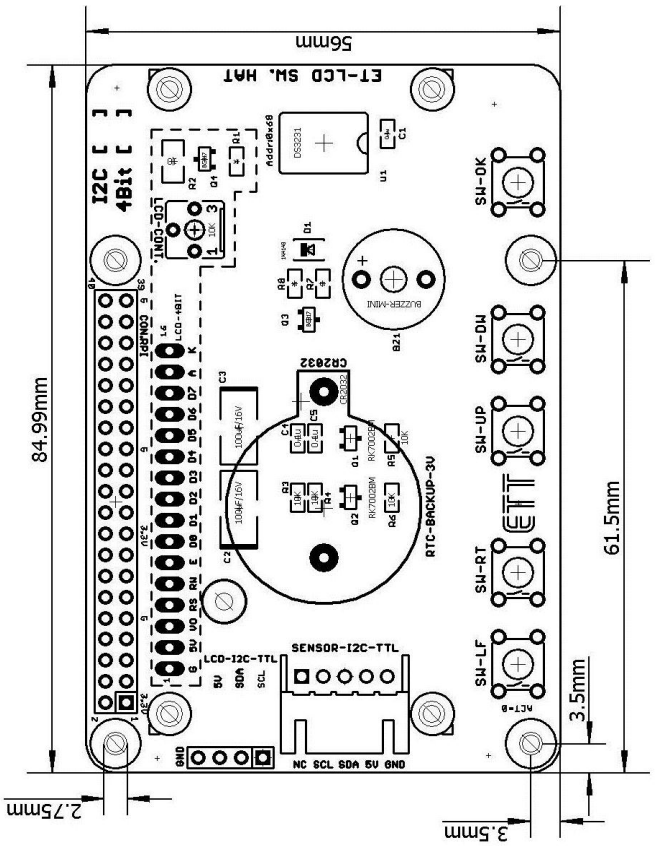
1. คุณสมบัติของบอร์ด ET-LCD SW HAT

- สามารถนำไปต่อใช้กับบอร์ด Raspberry Pi Model B+ , Pi2 Model B ที่เป็นขั้วต่อแบบ 40 Pin ได้
- ใช้ไฟเลี้ยง 5 V และ 3.3 V ที่ต่อจากบอร์ด Raspberry Pi โดยตรงทาง Connector 40 Pin ในการเลี้ยงบอร์ด
- Connector ที่ใช้เชื่อมต่อกับบอร์ด Raspberry Pi เป็นแบบ 2x20 Pin Stackable 25.5 mm ทำให้ Pin I/O ของบอร์ด RPi ที่ไม่ถูกใช้งานสามารถเชื่อมต่อไปใช้งานภายนอกได้อีก
- อุปกรณ์ I/O ทีต่อไว้ให้ใช้งานบนบอร์ด ประกอบไปด้วย
    * LCD 16x2 Backlight ต่อ Control แบบ Data 4 Bit พร้อมวงจรปรับ Contrast
    * SW.แบบกดติดปล่อยดับ จำนวน 5 ตัว ทำงานที่ Logic 0
    * Real Time Clock -RTC DS3231 (ชั่วโมง,นาที,วินาที,วันของสัปดาห์,วันที่,เดือน,ปี ค.ศ.และ Alarm) กำหนด RTC Address= 0x68
    * Buzzer ทำงานที่ Logic 1
    * ขั้วต่อ I2C แบบ Box 5 Pin สามารถต่อใช้งานกับ Temp Sensor ของ ETT รุ่น SHT31 ได้เลย ซึ่ง Pin I2C (SCL, SDA) จากบอร์ด RPi จะถูกต่อผ่านวงจรปรับระดับแรงดันบนบอร์ด ET-LCD SW HAT ก่อนที่จะต่อออกมาให้ใช้งานที่ขั้วต่อ Box 5 Pin ดังนั้นจึงหมดปัญหาเรื่องระดับแรงดันไม่เท่ากัน

2. การต่อใช้งานบอร์ด ET-LCD SW HAT และขนาดบอร์ด

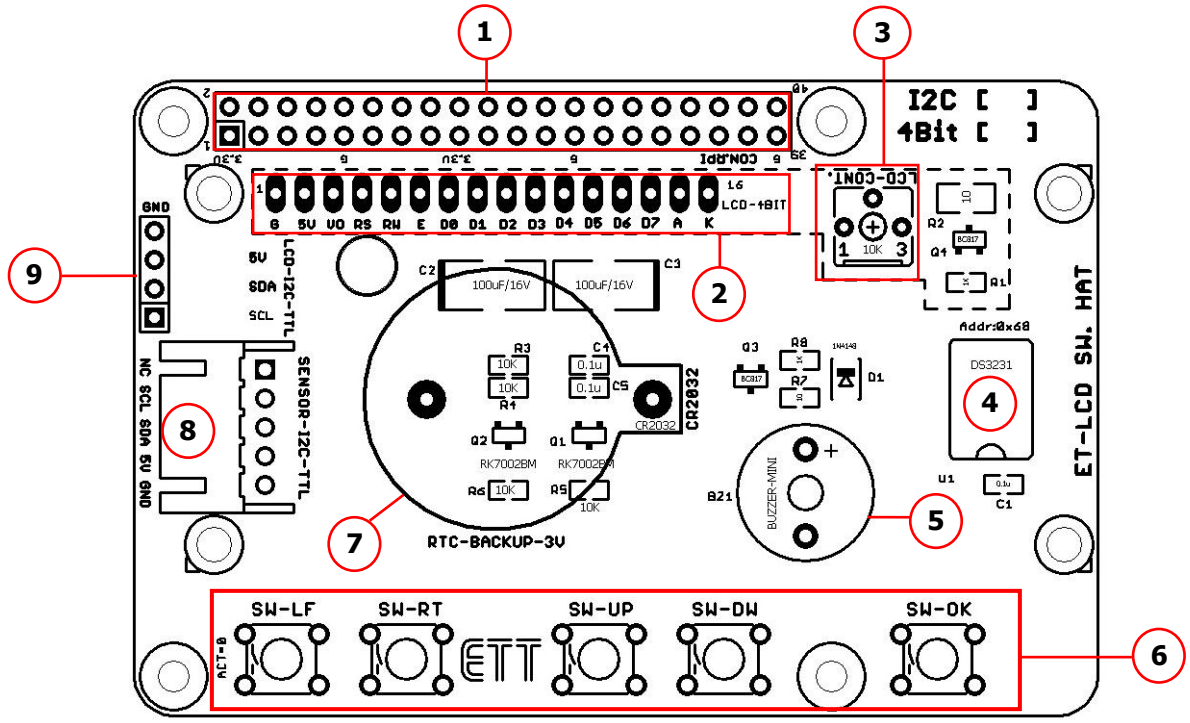


รูปการต่อใช้งานบอร์ด ET-LCD SW HAT กับบอร์ด RPi-2B



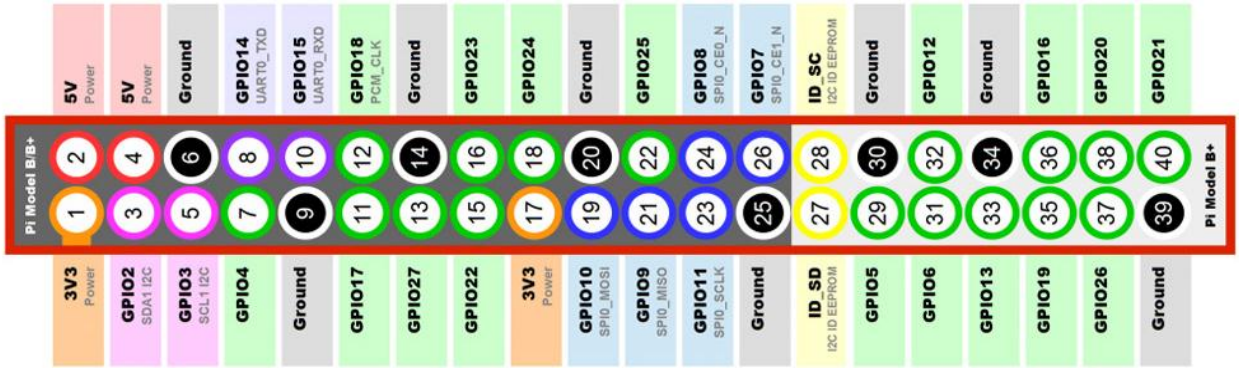
รูปขนาดบอร์ด ET-LCD SW HAT

3. ตำแหน่งขั้วต่อและอุปกรณ์ บนบอร์ด ET-LCD SW HAT

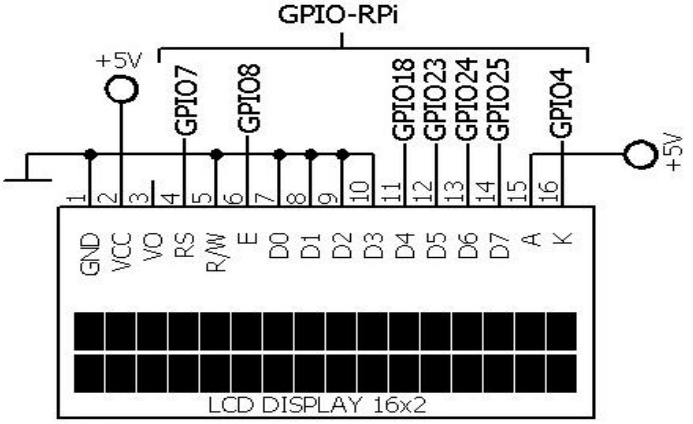


รูปแสดงตำแหน่งขั้วต่อและอุปกรณ์บนบอร์ด ET-LCD SW HAT

(1) CON-RPi : เป็น Connector 2x20 Pin Stackable 25.5 mm ใช้สำหรับต่อเข้ากับบอร์ด Raspberry Pi มีการจัดเรียงขาเหมือนกับบอร์ด Raspberry Pi ทุกอย่างดังรูป



(2) LCD-PAD HOLD : เป็น PAD 16 รู สำหรับต่อ LCD ขนาด 16x2 ซึ่ง LCD จะถูกต่อไว้ให้เรียบร้อยแล้วบนบอร์ด การจัดวงจรควบคุม LCD จะเป็นแบบ 4 Bit + Backlight โดยขา GPIO ที่ถูกต่อมาควบคุม LCD จะเป็นดังรูปด้านล่าง โดย GPIO4 จะใช้ควบคุมการทำงานของ Backlight ซึ่งจะ Active ที่ Logic 1 = ON



(3) LCD-Contrast : เป็น VR ใช้สำหรับปรับค่าความเข้มให้กับตัวอักษรบนจอ LCD

(4) RTC : เป็น IC Real Time Clock (RTC) #DS3231 ซึ่งจะถูกต่อ Interface แบบ I2C โดยมี I2C Address = 0x68 สามารถอ่านเขียนเวลาได้ทั้ง ชั่วโมง,นาที,วินาที,วันของสัปดาห์,วันที่,เดือน,ปี ค.ศ.และ Alarm โดยขา GPIO ที่ถูกต่อมาควบคุม RTC ได้แก่

GPIO2 (SDA1) → **SDA ของ RTC**

GPIO3 (SCL1) → **SCL ของ RTC**

(5) Buzzer : เป็น Buzzer สำหรับสร้างเสียง Beep ซึ่งจะทำงานที่ Logic 1 = ON โดยขา GPIO ที่ถูกต่อมาควบคุม Buzzer ได้แก่ **GPIO17**

(6) SW : เป็น Switch แบบกดติดปล่อยดับจำนวน 5 ตัวทำหน้าที่เป็น Input รับการทำงานจากผู้ใช้ เมื่อกด จะให้ Logic เป็น 0 ซึ่ง SW. ทั้ง 5 ตัว ไม่ได้ต่อ R-Pull Up ภายนอกไว้ดังนั้นเวลาเขียนโปรแกรมใช้งานผู้ใช้จะต้องกำหนดให้ GPIO ที่ต่อใช้งาน เป็น Input Pull Up ด้วย โดยคำสั่งที่ใช้สามารถดูได้จากตัวอย่างโปรแกรม " LCD4bit_SW_Buz.py " ส่วน ขา GPIO ที่ถูกต่อมารับสัญญาณจาก SW. ได้แก่

GPIO27 → **SW-LF**

GPIO22 → **SW-RT**

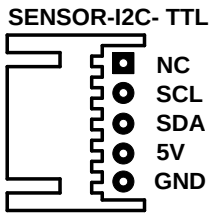
GPIO10 → **SW-UP**

GPIO9 → **SW-DW**

GPIO11 → **SW-OK**

(7) RTC-Backup : เป็นรางถ่านกลมใส่แบตเตอรี่ 3V (CR2032) สำหรับ Backup RTC

(8) CON-I2C : เป็นขั้วต่อแบบ Block 5 Pin ซึ่งจัดไว้สำหรับต่อ Temp Sensor ของ ETT รุ่น ET-SENSOR SHT31 หรือผู้ใช้จะใช้ต่ออุปกรณ์ I2C อย่างอื่นก็ได้ แต่อาจจะต้องดัดแปลงการต่อใหม่ถ้าขั้วต่อเข้ากันไม่ได้ โดยการจัดเรียง Pin ขั้วต่อแสดงดังรูป

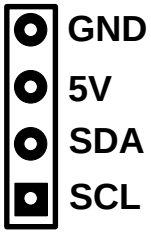


Pin ขั้วต่อ SCL และ SDA ที่ขั้วต่อนี้ได้ถูกต่อผ่านวงจรปรับระดับแรงดันไว้แล้วไม่ได้ต่อไปยัง GPIO ของ RPi โดยตรงดังนั้นจึงสามารถต่อเข้ากับอุปกรณ์ I2C ที่ทำงานด้วยแรงดัน 5V ได้ โดยอุปกรณ์ I2C ที่นำมาต่อไม่จำเป็นต้องต่อ R-Pull Up ที่ขา SDA และ SCL เนื่องจากที่วงจรปรับระดับแรงดันบนบอร์ด มีการ Pull Up ไปในตัวแล้ว โดยขา GPIO ที่ถูกต่อมาควบคุม ขั้วต่อ I2C ได้แก่

GPIO2 (SDA1) → **SDA**

GPIO3 (SCL1) → **SCL**

(9) I2C-PAD HOLD : เป็น PAD 4 รู ใช้สำหรับบัดกรีต่อสายไปควบคุมอุปกรณ์ I2C อื่นๆภายนอกเหมือนกับขั้วต่อหมายเลข 8 ทุกประการ หรือใช้ต่อกับ LCD-I2C เป็นต้น โดยมีการจัดเรียง Pin ตามรูป



4. ตัวอย่างโปรแกรมและการ Run โปรแกรม

สำหรับตัวอย่างโปรแกรมที่ให้มานี้จะเขียนด้วยภาษา Python มีอยู่ด้วยกัน 3 ตัวอย่าง ซึ่งจะคลอบคุมการใช้งานของอุปกรณ์ I/O ที่อยู่บนบอร์ด โดยเก็บไว้ใน Folder *" Ex_ET-LCD SW HAT /Ex_LCD_4Bit "* โดยในแต่ละตัวอย่างจะมีการทำงานตามหัวข้อ 4.1 ส่วนการ Run โปรแกรมตัวอย่างให้ดูในหัวข้อ 4.2 เพราะก่อน Run จะต้องมีการติดตั้ง Library บางตัวให้กับบอร์ด Raspberry Pi ก่อน Run โปรแกรมด้วย

4.1 ตัวอย่างโปรแกรม

สำหรับตัวอย่างโปรแกรมที่ให้มาใน CD จะมีอยู่ด้วยกัน 3 ตัวอย่าง เขียนด้วยภาษา Python ซึ่งจะครอบคลุมการ Control อุปกรณ์ต่างๆ ที่อยู่บนบอร์ดทั้งหมด ในแต่ละตัวอย่างจะใช้ฟังชั่นในส่วนของ RPi.GPIO ซึ่งติดตั้งมาพร้อมกับ OS Raspbian-jessie อยู่แล้ว และจะต้องติดตั้งเพิ่มเติมในส่วนของ Library SMBUS ซึ่งจะมีฟังชันก์เกี่ยวกับการ Interface กับ I2C ให้เรียกใช้งาน การทำงานของแต่ละตัวอย่างมีดังนี้

1) LCD4bit_SW_Buz.py : สำหรับตัวอย่างนี้จะเป็นตัวอย่างในการ Control LCD แบบ 4 Bit ,Buzzer และ อ่านค่า SW.ทั้ง5ปุ่ม จะเรียกใช้งานในส่วนของ ฟังชันก์ Library RPi.GPIO เท่านั้น การทำงานจะวนอยู่ในส่วนของฟังชันก์ main() คือจะวนรอรับค่าการกด SW. ทั้ง 5ตัว เมื่อมีการกด SW.ใด (Logic=0) ก็จะมีเสียง Beep ดังขึ้น และแสดงชื่อ SW.ที่ถูกกดบนหน้าจอ LCD เมื่อปล่อย SW.ชื่อ SW. นั้นก็จะหายไปและ ขึ้นข้อความ “Press SW.” มาแทนเป็นต้น

2) LCD4bit_RTC3231.py : ในตัวอย่างนี้จะเป็นการอ่านเขียน RTC และนำค่าเวลามาแสดงบนจอ LCD ซึ่งก่อน Run ตัวอย่างนี้ผู้ใช้ต้องแน่ใจว่า OS ที่ใช้ได้มีการติดตั้ง Library SMBUS เรียบร้อยแล้วและ I2C มีการเปิดใช้งาน มิฉะนั้นเวลา Run จะเกิด Error ได้ การทำงานจะวน Loop อยู่ในส่วนของฟังก์ชัน main() เริ่มต้นโปรแกรมจะเขียนค่าเวลา 12:00:00 และค่าวันที่ 29/9/16 เข้าไปเก็บไว้ใน RTC ก่อน จากนั้นก็จะวน Loop อ่านค่าเวลาและวันที่ออกมาจาก RTCและนำไปแสดงผลบนจอ LCD โดยบรรทัดบนจะเป็นค่าเวลาและบรรทัดล่างจะเป็นค่าวันที่ ให้สังเกตุวินาทีจะต้องเดินตลอดเวลา

3) LCD4bit_SHT31.py : ในตัวอย่างนี้จะเป็นการใช้งานขั้วต่อ RTC แบบบล็อก 5 Pin โดยในตัวอย่างจะใช้ Temp Sensor SHT31 ที่ทางอีทีทีจัดจำหน่ายมาต่อทดลองเนื่องจากมีขั้วต่อที่เสียบกันได้เลย ซึ่ง Sensor ตัวนี้จะ Interface แบบ I2C เหมือนกับ RTC ดังนั้นผู้ใช้ต้องแน่ใจว่า OS ที่ใช้ได้มีการติดตั้ง Library SMBUS เรียบร้อยแล้วและ I2C มีการเปิดใช้งาน มิฉะนั้นเวลา Run จะเกิด Error ได้ การทำงานจะวน Loop อยู่ในส่วนของฟังก์ชัน main() เช่นเดิม เริ่มต้นโปรแกรมก็จะ write I2C Address ไปยัง Sensor จากนั้นก็จะทำการอ่านค่าอุณหภูมิและความชื้นมาเก็บไว้ในตัวแปรเพื่อทำการคำนวณค่าอุณหภูมิและความชื้นจริงๆออกมา แล้วจึงส่งค่าที่คำนวณได้ไปแสดงบนหน้าจอ LCD โดยในบรรทัดบนจะแสดงค่าอุณหภูมิหน่วยเป็นองศาเซลเซียส (C) ส่วนบรรทัดล่างจะแสดงค่า เปอร์เซ็นต์ความชื้น(RH) เป็นต้น

4.2 การ Run ตัวอย่างโปรแกรม

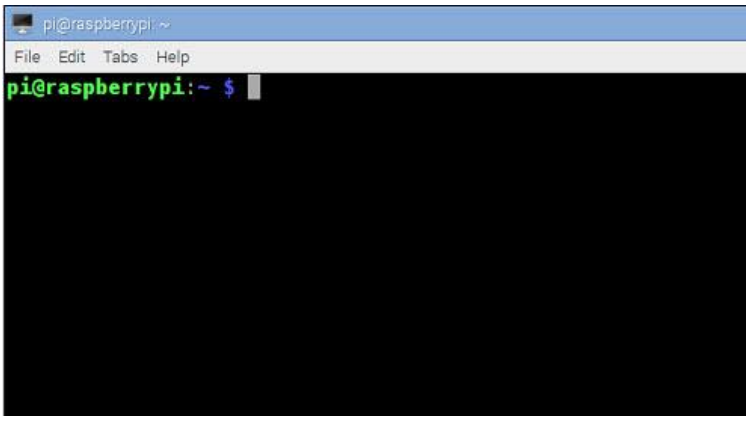
การ Run ตัวอย่างโปรแกรมผู้ใช้จะต้องทำการ Copy ตัวอย่างทั้งหมด ลงใน Micro SD ที่ติดตั้ง OS ไว้แล้ว เวลา Runโปรแกรมถ้าเกิด Error ขึ้นโปรแกรมไม่ Run ส่วนหนึ่งอาจมาจาก OS เก่าเกินไป ทำให้หาฟังชั่นใน Libary RPi.GPIO หรือ SMBUS ไม่เจอ ผู้ใช้จะต้องทำการ Upgrade OS ใหม่ซึ่งสามารถหาวิธีการ Upgrade OS ได้จาก Google ไม่ขอกล่าวถึงในที่นี้ สำหรับไฟล์ตัวอย่างจะเป็นแบบไฟล์เดียวนามสกุล .py ไม่ต้องมีการติดตั้ง Library ใดๆเพิ่มเติมนอกจาก RPI.GPIO(มากับ OS) และ SMBUS(ลงเพิ่ม) สำหรับ Compiler Python จะติดมากับ OS อยู่แล้วไม่ต้องลงเพิ่ม

ก่อนจะ Run โปรแกรมผู้ใช้จะต้องทำการเข้าใช้งานบอร์ด RPi เสียก่อนซึ่งทำได้ 3 แบบหลักๆดังนี้ แล้วแต่ผู้ใช้จะเลือกใช้งาน

- ***แบบที่1 :*** เข้าใช้งานแบบ Remote Command Line (SSH) ซึ่งจะเป็นการเข้าใช้งานบนวินโดว์ด้วยโปรแกรม PuTTy (อยู่ในFolder “Tool / SSH_RemoteCommand_Rpi”) การเข้าใช้งานวิธีนี้ บอร์ด RPi และ PC ของผู้ใช้จะต้องเชื่อมต่ออยู่ในเครือข่าย LAN เดียวกันโดยใช้ Router เป็นตัวกลาง
- ***แบบที่2 :*** เข้าใช้งานแบบ Remote Desktop (VNC) ซึ่งจะเป็นการเข้าใช้งานบนวินโดว์ด้วยโปรแกรม TightVNCServer (อยู่ในFolder “Tool / VNC_RemoteDesktop_RPi”) การเข้าใช้งานวิธีนี้ บอร์ด RPi และ PC ของผู้ใช้จะต้องเชื่อมต่ออยู่ในเครือข่าย LAN เดียวกันโดยใช้ Router เป็นตัวกลางเหมือนแบบที่ 1 แต่จะต่างกันตรงที่สามารถเข้าใช้งานในแบบของ Desktop ได้

- แบบที่3 : เข้าใช้งานโดยตรงจากบอร์ด RPi โดยต้องต่อมอนิเตอร์ที่ช่อง HDMI ต่อ Mouse และ Keyboard ที่ช่อง USB จากนั้นก็ใช้งานเหมือนเป็นคอมพิวเตอร์เครื่องหนึ่ง โดย Micros SD จะเหมือนเป็นฮาร์ดดิสก์ที่ลง OS ของ RPi ไว้แล้ว สามารถ Copy File ต่างๆผ่าน USB Drive ที่ต่อเข้าไปยังบอร์ด RPi ได้เลย

เมื่อเข้าใช้งานบอร์ด RPi ได้แล้วผู้ใช้จะได้หน้าต่าง Command Prompt แสดงดังรูปด้านล่างพร้อมใช้งาน ซึ่งเป็นหน้าต่างสำหรับให้ผู้ใช้ส่ง Command Line ไปคุยกับบอร์ด RPi ซึ่งจะเหมือนกับตอนที่เราใช้งานระบบ DOS Command บนวินโดว์นั่นเอง ส่วน Command Line ของ RPi มีอะไรบ้างสามารถหาข้อมูลได้จาก Google เป็นต้น



รูป Command Prompt of RPi

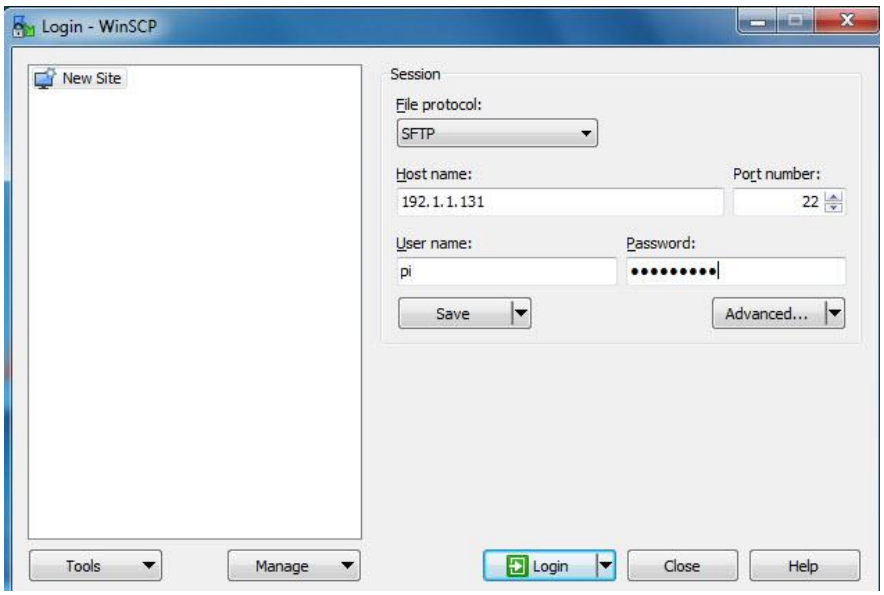
ในกรณีเข้าใช้งานบอร์ด RPi แบบที่ 2 หรือ 3 ผู้ใช้จะต้อง Run โปรแกรม Terminal บน Desktop [ ] ถึงจะแสดงหน้าต่างนี้ขึ้นมา

เริ่มต้นขั้นตอนการ Run โปรแกรม

1) ประกอบบอร์ด Raspberry Pi เข้ากับบอร์ด ET-LCD SW HAT ให้เรียบร้อยเหมือนรูปในหน้าแรกของคู่มือ และจ่ายไฟเลี้ยงให้บอร์ด RPi

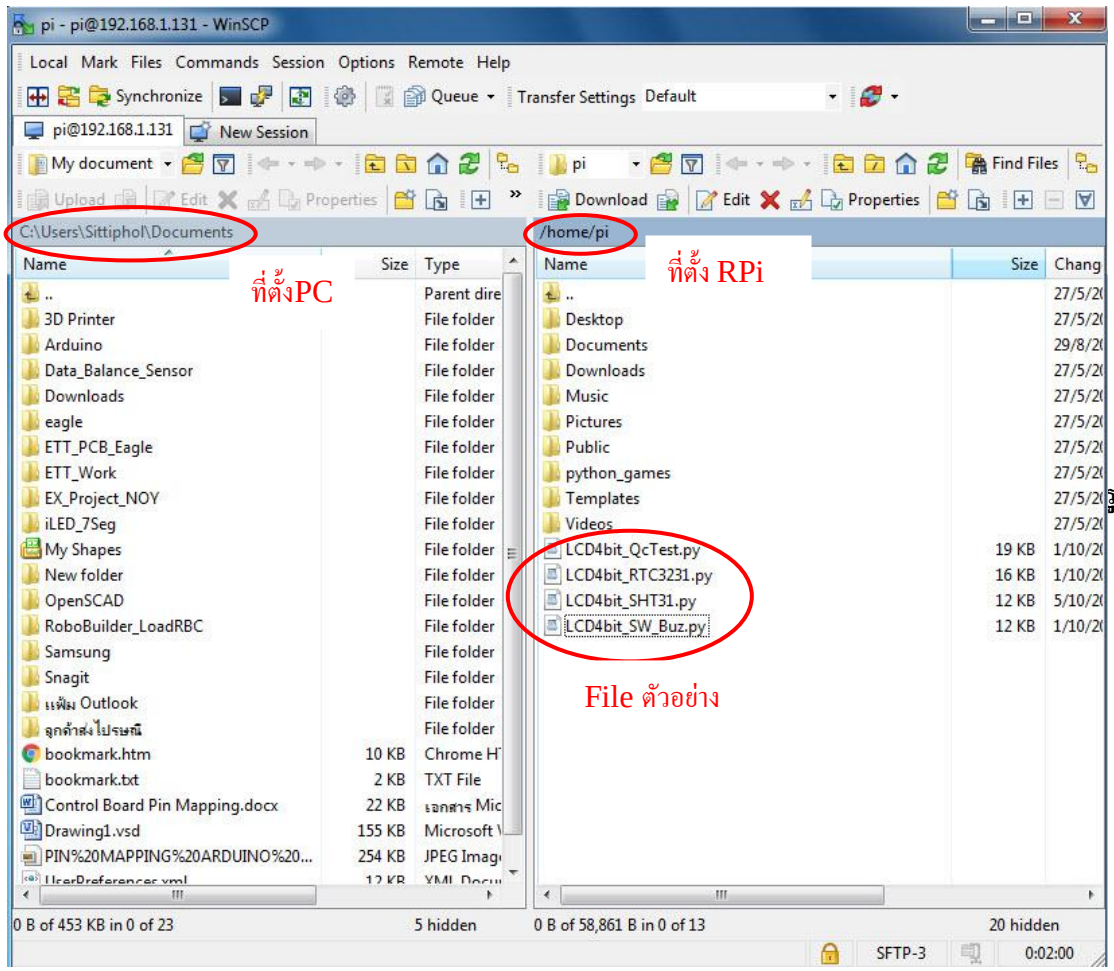
2) Copy File ตัวอย่างโปรแกรมที่ต้องการ Run ลงใน Micro SD Card ที่ลง OS-RPi ไว้แล้ว ซึ่งการ Copy File สามารถทำได้ หลายวิธี ขึ้นอยู่กับการเข้าใช้งานตัวบอร์ด RPi ในที่นี้จะขอแนะนำ 2 วิธีดังนี้

วิธีที่1 Copy File จาก Window ลงใน Micro SD Card ที่อยู่บนบอร์ด RPi ที่ติดตั้ง OS ไว้แล้ว โดยใช้โปรแกรม WinSCP (อยู่ใน Folder "Tool/VinSCP_Copy_File") ซึ่งวิธีนี้บอร์ด RPi และ PC ของผู้ใช้จะต้องเชื่อมต่ออยู่ในเครือข่าย LAN เดียวกันโดยใช้ Router เป็นตัวกลาง เมื่อเชื่อมตัว RPi และ PC เรียบร้อยแล้วจากนั้น บน PC ให้ Run โปรแกรม WinSCP [ ] ขึ้นมาจะได้หน้าต่างดังรูปที่1 ในช่อง Host name: ให้ใส่ IP Address ของบอร์ด Rpi ซึ่งจะต้องเข้าไปดูใน Router ที่ผู้ใช้ใช้อยู่ในหัวข้อ DHCP/Status ในช่อง User name: ให้ใส่ pi ในช่อง Password: ใส่ raspberry แล้วคลิก Login



รูปที่1

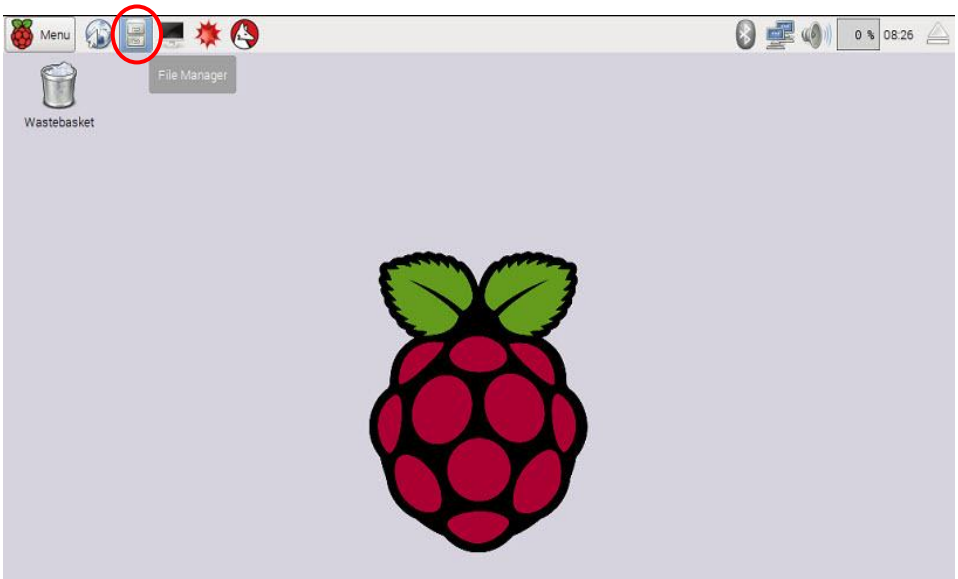
ถ้าโปรแกรม WinSCP ติดต่อกับบอร์ด RPi สำเร็จ ก็จะได้หน้าต่างขึ้นมาดังรูปที่2 โดยหน้าต่างด้านซ้ายจะเป็นพื้นที่ของ PC และหน้าต่างด้านขวาจะเป็นพื้นของ Rpi ในหน้าต่างของ PC ให้ผู้ใช้ Link เข้าไปยังที่เก็บ File ตัวอย่าง จากนั้นลาก File ตัวอย่างที่ต้องการจะ Run โยนมาไว้ฝั่งหน้าต่างของ Rpi ก็จะเห็นไฟล์ตัวอย่างดังรูปที่ 2 เมื่อ Copy File เสร็จแล้วให้ปิดโปรแกรม WinSCP และทำขั้นตอนต่อไป



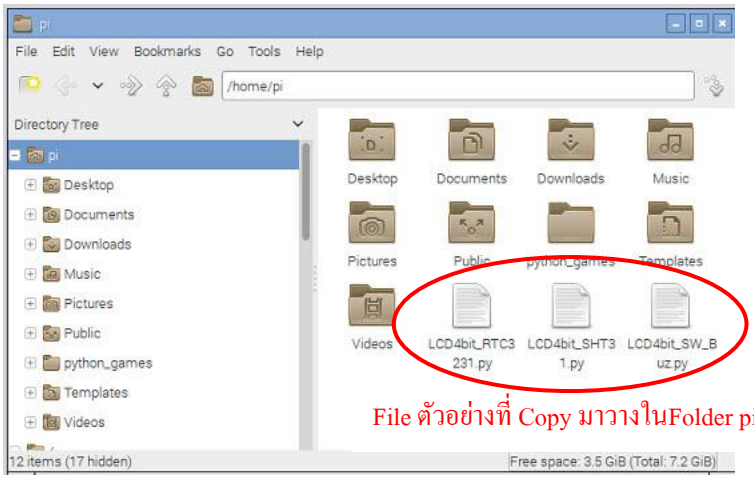
รูปที่2

วิธีที่2 Copy File ตัวอย่างจาก Window ลงใน USB Drive ปกติ จากนั้นนำ USB Drive ไปเสียบบนบอร์ด RPi ซึ่งเราจะมองบอร์ด RPi เป็น PC เครื่องหนึ่งโดยที่บอร์ด RPi จะต้องต่อ Monitor ,Mouse,Keyboard ให้เรียบร้อย จากนั้นทำการจ่ายไฟให้กับบอร์ด RPi และ Monitor ที่หน้าจอก็จะปรากฏ Desktop ของ RPi ขึ้นมาดังรูป (เข้าใช้งานบอร์ด RPi โดยตรงแบบ Desktop) จากนั้นทำการ Copy ไฟล์ตัวอย่างจาก USB Drive ไปวางไว้ในใน Folder /home/pi

โดยให้คลิกที่ Icon File Manager [ ] ก็จะปรากฏหน้าต่าง Folder pi ดังรูปด้านล่าง ให้ทำการวาง File ที่ Copy ไว้ ลงไป



รูป Desktop RPi



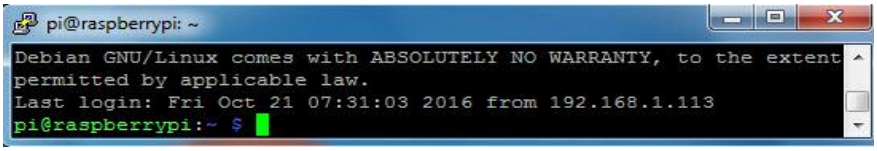
File ตัวอย่างที่ Copy มาวางในFolder pi

รูปหน้าต่าง Folder pi

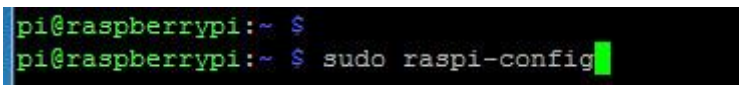
3) เมื่อ Copy File ไปไว้ใน Micro SD Card เรียบร้อยแล้ว ให้ทำการติดตั้ง Library SMBUS สำหรับใช้งาน I2C ดังนี้ (ถ้า Run ตัวอย่างที่ไม่มีการใช้งานอุปกรณ์ I2C หรือลง Library SMBUS และเคยเปิดใช้งาน I2Cแล้ว สามารถข้ามขั้นตอนนี้ไปได้)

3.1) ต่อ Internet ให้กับบอร์ด RPi ทางขัวต่อ LAN เพื่อใช้ ติดตั้ง Library ผ่านทาง Internet

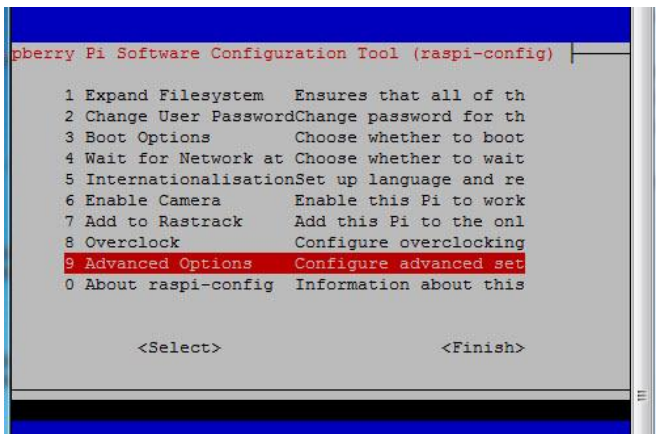
3.2) เปิดหน้าต่าง Command Prompt ของ RPi ขึ้นมาสำหรับพร้อมพิมพ์คำสั่งดังรูป

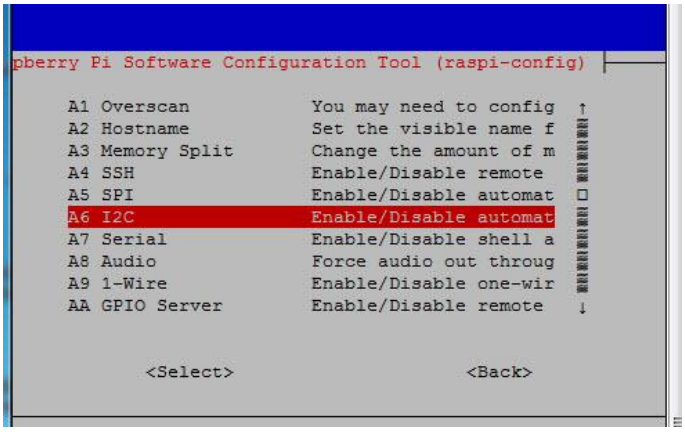


3.3) ทำการ Enable i2C Port ใน OS Raspbian ก่อนจะใช้งาน (ทำเมื่อใช้งานครั้งแรก) โดยพิมพ์คำสั่งดังนี้

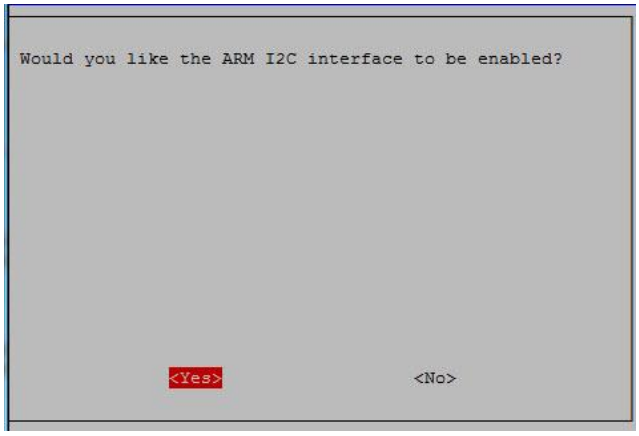


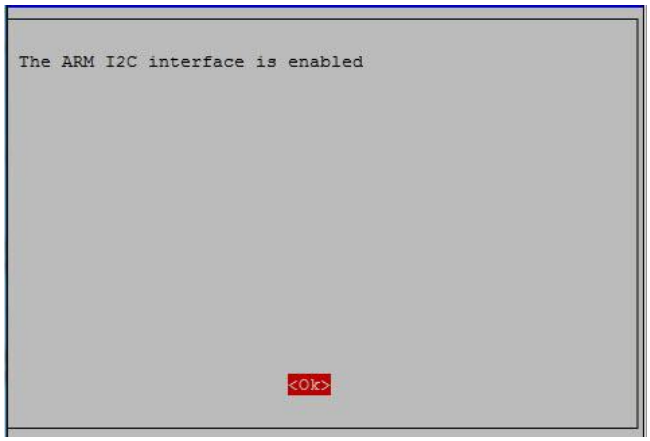
หน้าจอก็จะปรากฏรายการดังรูปด้านล่าง ให้เลือกไปที่ “ 9 Advanced Option” จากนั้นเลือกไปที่ “ A6 I2C ”



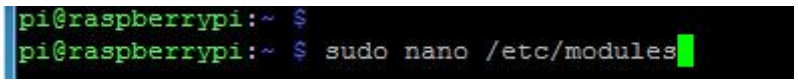


จะปรากฏหน้าต่างถามว่าต้องการ Enable I2C หรือไม่ ให้เลือก <Yes> ต่อมาก็จะมีหน้าต่างบอกว่า I2C ถูก Enable แล้ว ให้เลือก <OK> ก็จะกลับมาที่หน้าเมนูแรกอีกครั้ง ให้เลือก <Finish> เพื่อกลับเข้าสู่ Command Prompt เป็นการ Enable I2C เรียบร้อยแล้ว





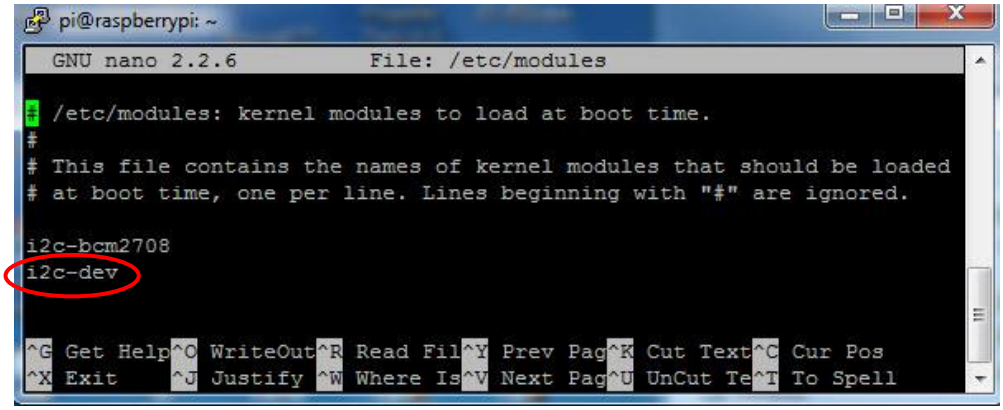
3.4) เข้าไปทำการแก้ไข Module file โดยพิมพ์คำสั่งดังนี้



จากนั้นจะได้หน้าต่าง nano ขึ้นมาแสดงดังรูปด้านล่าง และให้พิมพ์บรรทัดนี้ลงไป

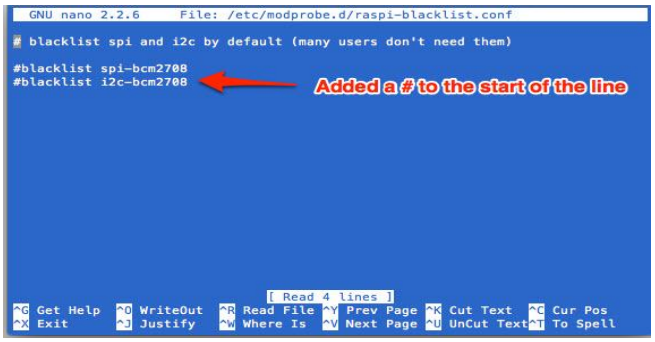
i2c-dev

ถ้ามีการพิมพ์ไว้แล้วก็ไม่ต้องพิมพ์ จากนั้นกด Control X เพื่อออก และ Save ทับไฟล์ เดิมลงไป แล้วจะกลับเข้าสู่หน้า Command Prompt เหมือนเดิม



3.5) ต่อไปทำการปิดค่า Config บางตัว โดยพิมพ์คำสั่งตามรูปด้านล่าง

ก็จะได้หน้าต่าง nano ขึ้นมาถ้าพบ บรรทัด blacklist i2c-bcm2708 ให้ใส่เครื่องหมาย # ด้านหน้าเพื่อปิดไม่ใช้งาน ดังรูป ถ้าไม่พบก็ไม่ต้องใส่อะไร จากนั้นกด Control X เพื่อออก และ Save ทับไฟล์เดิมลงไป แล้วจะกลับเข้าสู่หน้า Command Prompt เหมือนเดิม



3.6) ทำการติดตั้ง Library I2C-SMBUS ซึ่งจะต้องต่อ Internet ผ่าน Port LAN ไว้ด้วย โดยให้พิมพ์คำสั่งตามรูปด้านล่าง

```
pi@raspberrypi:~ $
pi@raspberrypi:~ $ sudo apt-get install python-smbus i2c-tools
```

จากนั้น RPi ก็จะทำการ Download และ Install Library เมื่อเรียบร้อยแล้วที่หน้าจอก็จะแสดง Command Prompt เหมือนเดิม ดังรูป

```
pi@raspberrypi:~ $ sudo apt-get install python-smbus i2c-tools
Reading package lists... Done
Building dependency tree
Reading state information... Done
Suggested packages:
  libi2c-dev
The following NEW packages will be installed:
  i2c-tools python-smbus
0 upgraded, 2 newly installed, 0 to remove and 0 not upgraded.
Need to get 60.8 kB of archives.
After this operation, 286 kB of additional disk space will be used.
Get:1 http://archive.raspberrypi.org/debian/ jessie/main i2c-tools armhf 3.1.1+s
vn-2 [51.3 kB]
Get:2 http://archive.raspberrypi.org/debian/ jessie/main python-smbus armhf 3.1.
1+svn-2 [9,462 B]
Fetched 60.8 kB in 3s (18.3 kB/s)
Selecting previously unselected package i2c-tools.
(Reading database ... 118476 files and directories currently installed.)
Preparing to unpack .../i2c-tools_3.1.1+svn-2_armhf.deb ...
Unpacking i2c-tools (3.1.1+svn-2) ...
Selecting previously unselected package python-smbus.
Preparing to unpack .../python-smbus_3.1.1+svn-2_armhf.deb ...
Unpacking python-smbus (3.1.1+svn-2) ...
Processing triggers for man-db (2.7.0.2-5) ...
Setting up i2c-tools (3.1.1+svn-2) ...
/run/udev or .udevdb or .udev presence implies active udev.  Aborting MAKEDEV in
vocation.
Setting up python-smbus (3.1.1+svn-2) ...
pi@raspberrypi:~ $
```

3.7) ทำการปิดไฟเลี้ยงบอร์ด RPi และปิดโปรแกรมที่ใช้ส่ง Command Line แล้วเปิดมันทั้งหมดขึ้นมาใหม่เพื่อเป็นการ Reset ระบบ ซึ่งตอนนี้ I2C Pin จะสามารถใช้งานได้แล้ว

3.8) ทำการทดสอบการเชื่อมต่อ I2C เมื่อ Reboot ระบบขึ้นมาใหม่แล้วให้ไปที่หน้า Command Prompt แล้วพิมพ์คำสั่งตามรูปด้านล่าง โดยเลข 1 ท้ายคำสั่งจะเป็นหมายเลขลำดับ I2C ที่เราใช้งานซึ่งปกติจะเขียนกำกับไว้หลังชื่อ Pin I2C สำหรับ RPi2 นี้จะเป็น I2C-1 ส่วน RPi1 จะเป็น I2C-0

```
pi@raspberrypi:~ $
pi@raspberrypi:~ $ sudo i2cdetect -y 1
```

เมื่อคำสั่งนี้ถูกกระทำ ก็จะแสดงหมายเลข I2C Address (Hex) ของอุปกรณ์ I2C ทั้งหมดที่ต่ออยู่ในระบบ BUS I2C ให้ผู้ใช้เห็น ดังรูป

```
pi@raspberrypi:~ $ sudo i2cdetect -y 1
     0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:          -- -- -- -- -- -- -- -- -- -- -- -- --
10: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
20: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
30: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
40: -- -- -- -- 44 -- -- -- -- -- -- -- -- -- -- --
50: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
60: -- -- -- -- -- -- -- -- 68 -- -- -- -- -- -- --
70: -- -- -- -- -- -- -- --
pi@raspberrypi:~ $
```

ซึ่งจากรูป 0x44 คือ Address ของ Temp Sensor SHT31 ส่วน 0x68 คือ Address ของ RTC DS3231 นั่นเอง ในกรณีที่ส่งคำสั่งนี้ไปแล้วมีการฟ้องกลับจากระบบว่า Command not found นั่นแสดงว่ายังติดตั้ง Library I2C ไม่สำเร็จ หรือ ไม่ได้ Enable I2C ไว้ตามที่กล่าวไปข้างต้น

ในกรณีที่เราไม่รู้ว่า I2C ของ RPi ที่เราใช้มีหมายเลขลำดับเท่าไหร่สามารถใช้คำสั่ง ls -l /dev/i2c* แสดงตามรูปด้านล่าง

```
pi@raspberrypi:~ $ ls -l /dev/i2c*
crw-rw---- 1 root i2c 89, 1 Sep 27 02:51 /dev/i2c-1
```

หมายเลข I2C

4) ทำการ Run โปรแกรมตัวอย่างโดยไปที่หน้า Command Prompt แล้วพิมพ์คำสั่ง ตามรูปด้านล่าง

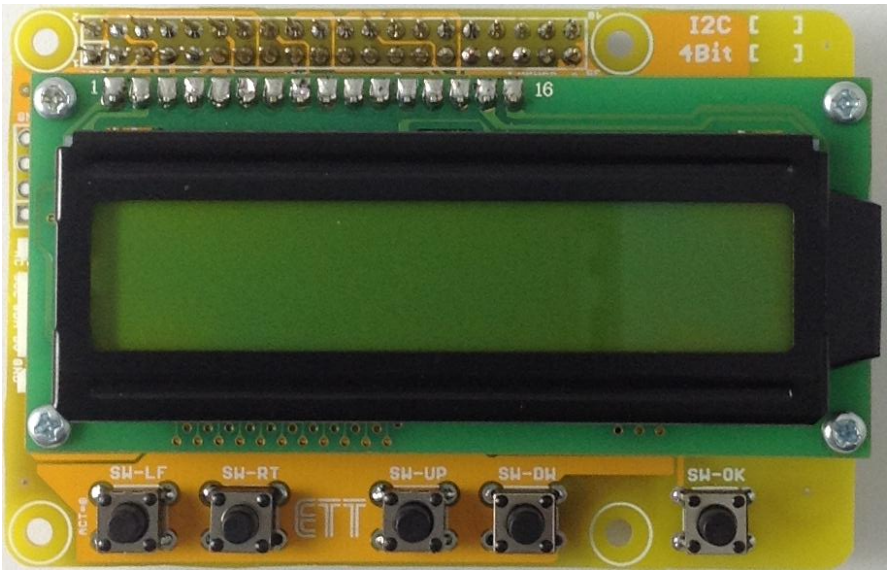
รูปแบบ Command : sudo python ชื่อไฟล์.py

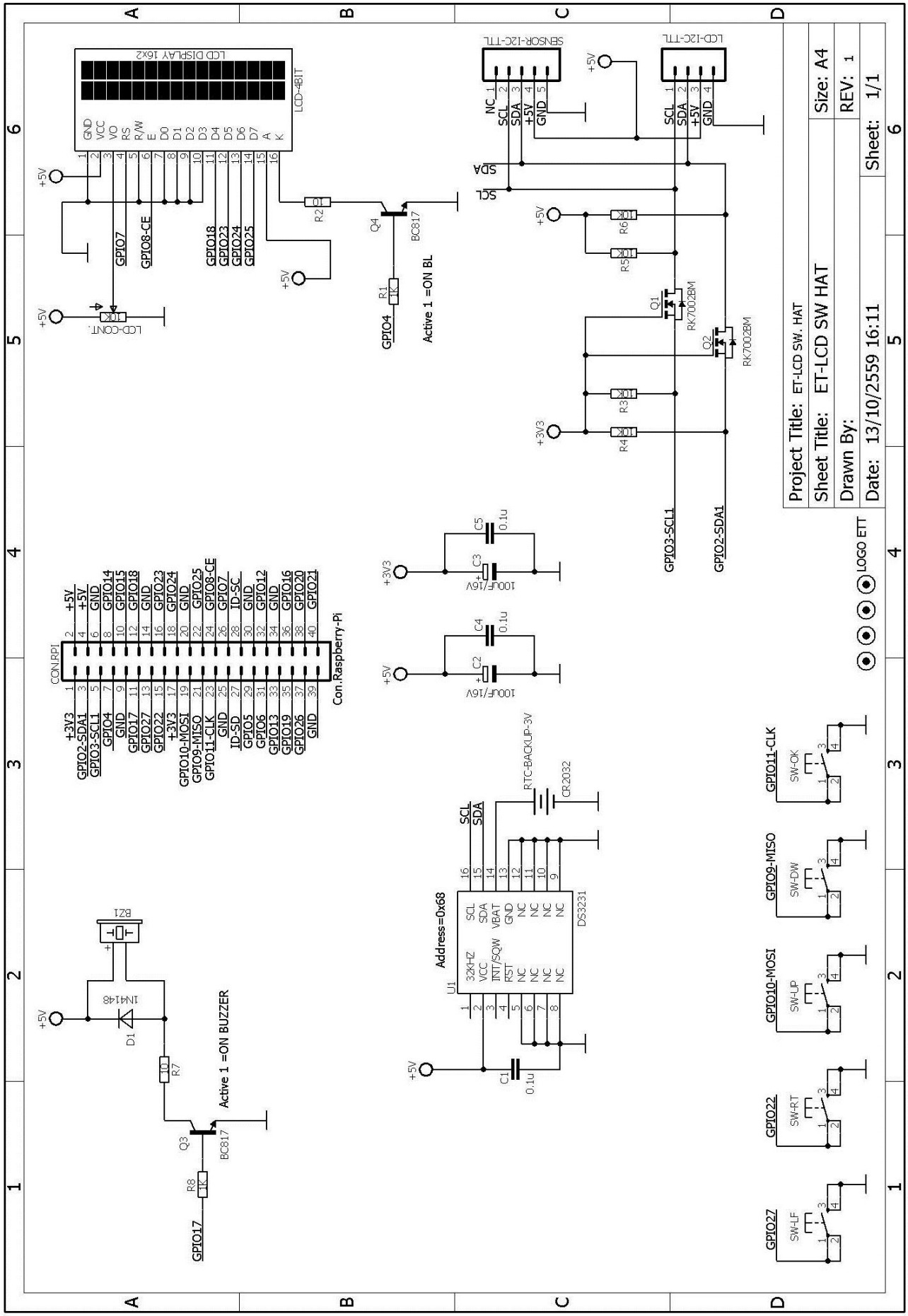
```
pi@raspberrypi:~ $ sudo python LCD4bit_RTC3231.py
```

เนื่องจากโปรแกรมตัวอย่างนั้นจะ Run วน Loop ไปเรื่อยๆ ถ้าต้องการออกจากโปรแกรม มาอยู่ที่ Command Prompt ให้กด Ctrl Z เพื่อ Stop โปรแกรมที่ Run อยู่

ให้สังเกตว่า File ตัวอย่าง ที่ให้ Copy ไปวางตามที่กล่าวไว้ในหัวข้อข้างต้นนั้นจะถูกไปวางไว้ที่ Folder home/pi ดังนั้นจึงสามารถใช้คำสั่ง Run จากชั้นแรกของ Command Prompt ได้ทันที

หมายเหตุ ขั้นตอนและวิธีการต่างๆในการใช้งานบอร์ด Raspberry Pi ที่กล่าวไปทั้งหมดนี้เป็นเพียงวิธีการหนึ่งเท่านั้นเพื่อใช้อธิบายให้ผู้ใช้เข้าใจ ซึ่งอาจจะมีอีกหลายวิธีในการใช้งาน อยู่ที่ความถนัดของผู้ใช้งานจะเลือกใช้ ไม่จำเป็นต้องยึดตามขั้นตอนที่กล่าวไปข้างต้นทั้งหมด เพราะขั้นตอนอาจเปลี่ยนแปลงได้ตามเวอร์ชัน OS ใหม่ๆที่มีการเปลี่ยนแปลง





วงจร ET-LCD SW HAT