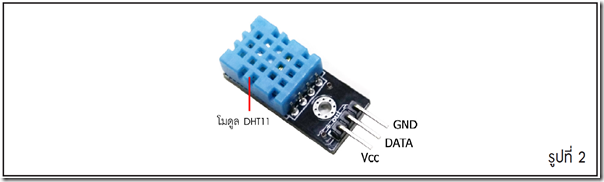
**เฉลยโจทย์ การแข่งขันทักษะระดับภาคข้อ 2-6**

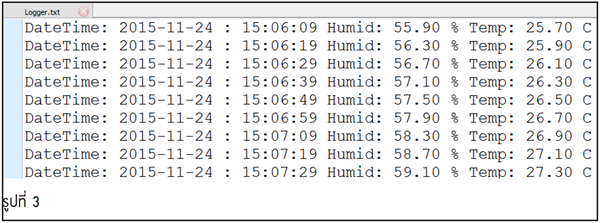
**1. การส่งอีเมล์และ ค้นหาข้อมูลในเวปไซต์ต่าง ๆ**

**ข้อนี้ไม่จำเป็นต้องเฉลย**

**2. สร้างระบบ   Data  Logger  ในโรงเรือน**



แสดงค่าความเปลี่ยนแปลงความชื้นและอุณหภูมิทุกๆ 10 วินาที โดยทำตามขั้นตอนดังนี้ ทำการต่อโมดูล DHT11 เข้ากับบอร์ด Raspberry Pi จากนั้นเขียนโปรแกรมเพื่ออ่านค่าความชื้นและอุณหภูมิเก็บไว้ในไฟล์ชื่อ Logger.txt โดยรูปแบบการเก็บข้อมูลใน 1 บรรทัดประกอบด้วย วันที่ – เวลา – ค่าความชื้น – ค่าอุณหภูมิ รูปแบบของไฟล์ Text เมื่อเปิดขึ้นมามีรูปแบบดังนี้



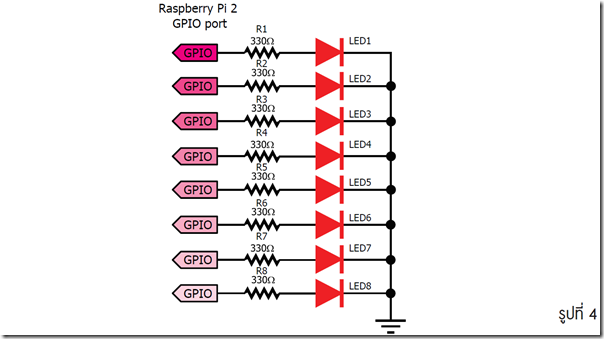
          (2.1) สามารถสร้างไฟล์มีค่า ปี/เดือน/วัน/ชั่วโมง/นาทีและวินาที ถูกต้อง **(2 คะแนน)**   
          (2.2) สามารถตัดการแสดงผลหน่วยไมโครวินาที ออกจากชุดเวลาได้      **(2 คะแนน)**  
          (2.3) สามารถแสดงค่า Humidity ได้ถูกต้อง **( 2 คะแนน)**   
          (2.4) สามารถแสดงเครื่องหมาย “%” (เปอร์เซ็นต์) ได้ **(2 คะแนน)**  
          (2.5) สามารถแสดงค่า Temparature ได้ถูกต้อง **(2 คะแนน)**

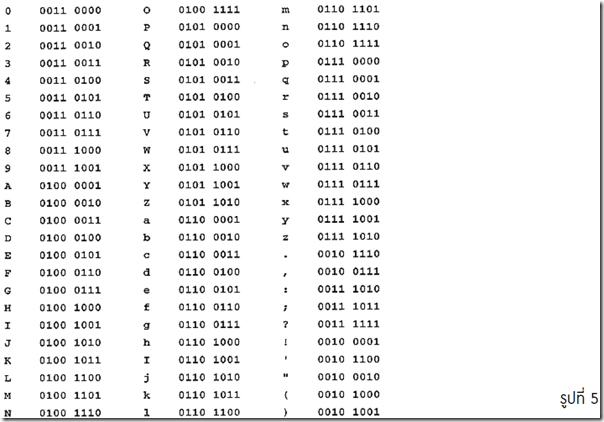
**จากโจทย์เขียนเป็นโปรแกรมได้ดังนี้**



**ข้อ (2.1)** จะตอบโจทย์ด้วย คำสั่ง **date=datetime.datetime.now()**  
**ข้อ (2.2)** จะตอบโจทย์ได้ด้วยคำสั่ง **date.strftime(“%Y-%M-%d : %H:%M:%S”)** เพื่อกำหนดรูปแบบการแสดง วันเดือนปี และเวลา  
**ข้อ (2.3)** และ **(2.5)** จะตอบโจทย์ด้วยการอ่านค่า **humidity, temperature = Adafruit\_DHT.read\_retry(Sensor,GPIO)**  
**ข้อ (2.4)** เครื่องหมาย % แสดงด้วยคำสั่ง **%%**

**3. รหัสแอสกี้และเลขฐานสอง**

[](http://doc.inex.co.th/wp-content/uploads/2015/12/image2.png)

[](http://doc.inex.co.th/wp-content/uploads/2015/12/image3.png)

เขียนโปรแกรมรับค่าจากคีย์บอร์ดที่หน้าต่าง Terminal นำค่าที่ได้แสดงผลเป็นค่าเลขฐานสองตามรหัสแอสกี้ที่เกิดขึ้น โดยมีรูปแบบตามรูปที่ 5 ลอจิก “1” LED ติด ลอจิก “0” LED ดับ

(3.1) สามารถแสดงค่าเลขฐานสองที่หน้าต่าง Terminal ได้ถูกต้อง ( 4 คะแนน)   
(3.2) สามารถให้ LED ติดดับตามค่าข้อมูลเลขฐานสองได้ถูกต้อง ( 6 คะแนน)

**จากโจทย์เขียนเป็นโปรแกรมได้ดังนี้**

****

หัวใจของโปรแกรมนี้คือการระบุตำแหน่งของ LED ให้อยู่ในรูปแบบของ list หรือ array เพื่อระบุตำแหน่งบิตได้สะดวก

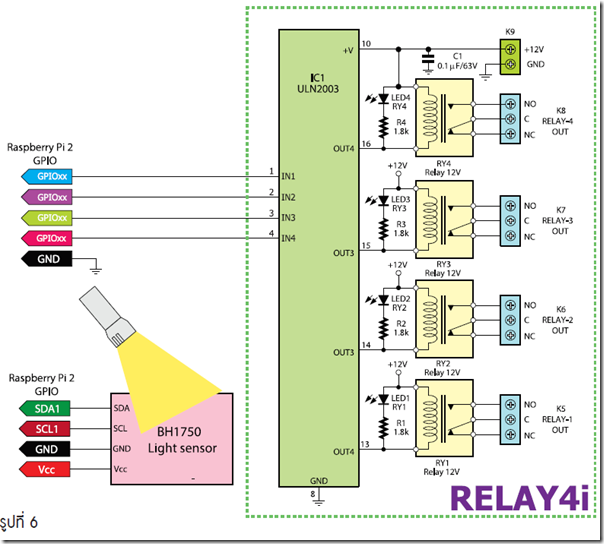
**my\_pin=[6,12,13,19,16,26,20,21]**

อีกคำสั่งคือ ORD สำหรับการเปลี่ยนค่ารหัสแอสกี้ที่อ่านได้จากการกดคีย์บอร์ดให้อยู่ในรูป integer

bin2str=bin(ord(ascii)+256)

**หมายเหตุ** 256 ที่บวกเพิ่มเพื่อให้เวลาแสดงเลขฐาน 2 แล้วเลข 0 ด้านหน้าไม่ถูกตัดหายไป

**4. สวิตช์รหัสกระตุ้นด้วยแสง**



 สั่งการเปิดปิดรีเลย์ด้วยการสั่งงานด้วยแสง โดยสภาวะปกติ BH1750 จะได้รับแสงปกติของห้อง  
     (4.1) ใช้กระดาษบังแสงให้ BH1750 แล้วเอาออกรอ 3 วินาที RELAY1 ทำงาน ทำอีกครั้ง RELAY1 หยุดทำงานสลับกันไปเรื่อย ๆ (Toogle) (4 คะแนน)  
     (4.2) ใช้กระดาษบังแสงให้ BH1750 แล้วเอาออก 2 ครั้ง รอ 3 วินาที RELAY2 ทำงาน ทำอีกครั้ง RELAY2 หยุดทำงานสลับกันไปเรื่อย ๆ (Toogle) (4 คะแนน)  
     (4.3) ใช้กระดาษบังแสงให้ BH1750 แล้วเอาออก 3 ครั้ง รอ 3 วินาที RELAY3 ทำงาน ทำอีกครั้ง RELAY3 หยุดทำงาน สลับกันไปเรื่อย ๆ (Toogle) (4 คะแนน)  
     (4.4) ใช้กระดาษบังแสงให้ BH1750 แล้วเอาออก 4 ครั้ง รอ 3 วินาที RELAY4 ทำงาน ทำอีกครั้ง RELAY4 หยุดทำงาน สลับกันไปเรื่อย ๆ (Toogle) (4 คะแนน)  
     (4.5) ใช้กระดาษบังแสงให้ BH1750 แล้วเอาออก 5 ครั้ง รอ 3 วินาที RELAY ทุกตัวทำงาน ทำอีกครั้ง RELAY ทุกตัวหยุดทำงาน สลับกันไปเรื่อย ๆ (Toogle) (4 คะแนน)

**import** time

**import** smbus

bus **=** smbus**.**SMBus**(**1**)**

**import** RPi**.**GPIO **as** GPIO

**import** time

GPIO**.**setwarnings**(False)**

GPIO**.**setmode**(**GPIO**.**BCM**)**

addr **=** 0x23 # i2c adress

my\_pin**=[**16**,**26**,**20**,**21**]**

**for** pin **in** my\_pin**:**

GPIO**.**setup**(**pin**,**GPIO**.**OUT**)**

st1**=**0

st2**=**0

st3**=**0

st4**=**0

st5**=**0

count**=**0

timer**=**0

**while** **True:**

time**.**sleep**(**0.2**)**

data **=** bus**.**read\_i2c\_block\_data**(**addr**,**0x11**)**

lum**=(**data**[**1**]** **+** **(**data**[**0**]<<**8**)** **/** 1.2**)**

#print ("Luminosity " ,lum,"lx")

**while** **(**lum**<**100**):**

timer**=**0

count**+=**1

data **=** bus**.**read\_i2c\_block\_data**(**addr**,**0x11**)**

lum**=(**data**[**1**]** **+** **(**data**[**0**]<<**8**)** **/** 1.2**)**

time**.**sleep**(**0.2**)**

**while** **(**lum**<**100**):**

data **=** bus**.**read\_i2c\_block\_data**(**addr**,**0x11**)**

lum**=(**data**[**1**]** **+** **(**data**[**0**]<<**8**)** **/** 1.2**)**

time**.**sleep**(**0.2**)**

**print(**count**)**

timer**+=**1

#print ("Timer=",timer)

**if** **(**timer**>**10**):**

**if(**count**==**1**):**

st1**=~**st1

GPIO**.**output**(**my\_pin**[**0**],**st1**)**

**elif(**count**==**2**):**

st2**=~**st2

GPIO**.**output**(**my\_pin**[**1**],**st2**)**

**elif(**count**==**3**):**

st3**=~**st3

GPIO**.**output**(**my\_pin**[**2**],**st3**)**

**elif(**count**==**4**):**

st4**=~**st4

GPIO**.**output**(**my\_pin**[**3**],**st4**)**

**elif(**count**==**5**):**

st5**=~**st5

st1**=**st2**=**st3**=**st4**=**st5

GPIO**.**output**(**my\_pin**[**0**],**st5**)**

GPIO**.**output**(**my\_pin**[**1**],**st5**)**

GPIO**.**output**(**my\_pin**[**2**],**st5**)**

GPIO**.**output**(**my\_pin**[**3**],**st5**)**

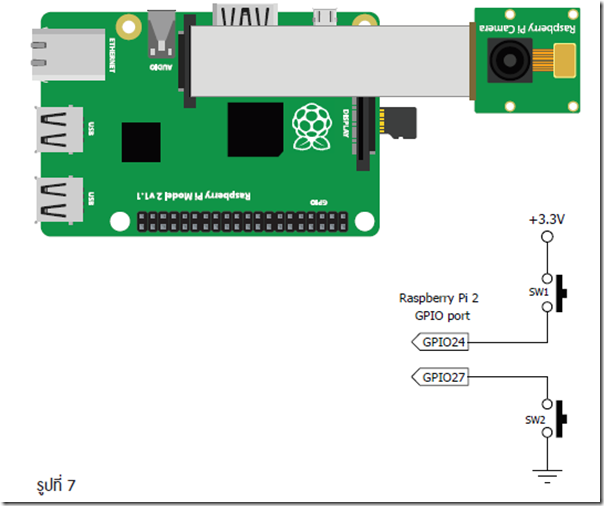
count**=**0

**การทำงานของโปรแกรม**

โปรแกรมนี้ใช้ while ในการตรวจสอบ   
while (lum<100)  ตรวจสอบว่าบังแสง  ถ้าใช่ เพิ่มค่าการนับ count  
while (lum<100)  รอจนกระทั่งเอากระดาษบังแสงออก

ตัวแปร timer จะคอยตรวจสอบว่า ไม่มีการบังแสงนานเกินกว่าเวลาที่กำหนด (2-3 วินาที) ถ้าเกินก็ไปสั่งตรวจสอบ ค่า count เพื่อสั่ง RELAY เปิดหรือปิด

**5. ระบบถ่ายภาพด้วยสวิตช์ 2 รูปแบบ**

[](http://doc.inex.co.th/wp-content/uploads/2015/12/image5.png)

ใช้สวิตช์ 2 ตัวเพื่อถ่ายภาพ โดยการต่อสวิตช์ตามรูปเท่านั้น (ห้ามต่อตัวต้านทาน)  
 (5.1) ถ้ากดสวิตช์ SW1 ให้ถ่ายภาพและเก็บไฟล์ภาพไว้ที่โฟลเดอร์ home/pi/Desktop/SW1 โดยมีรูปแบบชื่อไฟล์เป็น Imagexx.jpg โดยค่า xx คือค่าตัวเลขที่ต้องเพิ่มขึ้นทีละ 1 ค่าไปเรื่อยๆ เช่น Image01.jpg Image02.jpg Image03.jpg ( 5 คะแนน)  
 (5.2) ถ้ากดสวิตช์ SW2 ให้ถ่ายภาพและเก็บไฟล์ภาพไว้ที่โฟลเดอร์ home/pi/Desktop/SW2 โดยมีรูปแบบชื่อไฟล์เป็นวัน/เดือน/ปีและเวลา (5 คะแนน)

**import** RPi**.**GPIO **as** GPIO

**import** time

**import** picamera

**import** datetime

GPIO**.**setwarnings**(False)**

GPIO**.**setmode**(**GPIO**.**BCM**)**

sw\_1 **=** 24

sw\_2 **=** 27

count**=** 0

GPIO**.**setup**(**sw\_1**,**GPIO**.**IN**,**pull\_up\_down**=**GPIO**.**PUD\_DOWN**)**

GPIO**.**setup**(**sw\_2**,**GPIO**.**IN**,**pull\_up\_down**=**GPIO**.**PUD\_UP**)**

GPIO**.**add\_event\_detect**(**sw\_1**,**GPIO**.**RISING**)**

GPIO**.**add\_event\_detect**(**sw\_2**,**GPIO**.**FALLING**)**

camera **=** picamera**.**PiCamera**()**

**while(**1**):**

**if** GPIO**.**event\_detected**(**sw\_1**):**

**print(**"SW1"**)**

img\_name**=**"/home/pi/Desktop/SW1/"

img\_name**+=**str**(**datetime**.**datetime**.**now**())**

img\_name**+=**".jpg"

camera**.**capture**(**img\_name**)**

**if** GPIO**.**event\_detected**(**sw\_2**):**

count**+=**1

**print(**"SW2"**)**

img\_name**=**"/home/pi/Desktop/SW2/Image"

img\_name**+=** str**(**count**).**zfill**(**2**)**

img\_name**+=**".jpg"

camera**.**capture**(**img\_name**)**

**การทำงานของโปรแกรม**

หลัก ๆ ของโปรแกรมนี้คือให้ใช้การพูลอัพ และพูลดาวน์ภายในตัว Raspberry Pi ด้วยคำสั่ง

GPIO.setup(sw\_1,GPIO.IN,pull\_up\_down=GPIO.PUD\_DOWN)  
 GPIO.setup(sw\_2,GPIO.IN,pull\_up\_down=GPIO.PUD\_UP)

เพื่อให้การทำงานเกิดการตรวจสอบได้อย่างรวดเร็ว อาจจะใช้การสร้าง Event เพื่อตรวจสอบการกดสวิตช์

GPIO.add\_event\_detect(sw\_1,GPIO.RISING)  
 GPIO.add\_event\_detect(sw\_2,GPIO.FALLING)

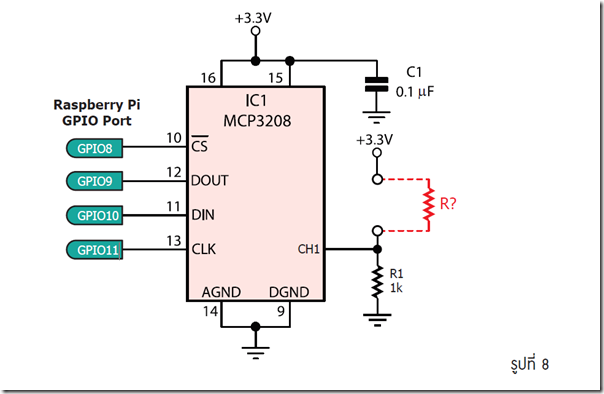
การตรวจสอบการกดสวิตช์ก็สามารถทำผ่านคำสั่ง

if GPIO.event\_detected(sw\_1):

if GPIO.event\_detected(sw\_2):

นอกนั้นที่เหลือ ก็เป็นการสั่งงานถ่ายรูปปกติ  โดยผลลัพธ์ที่เกิดขึ้น จะเก็บในโฟลเดอร์ที่ต่างกัน

**6. โอห์มมิเตอร์ เครื่องวัดค่าความต้านทาน**

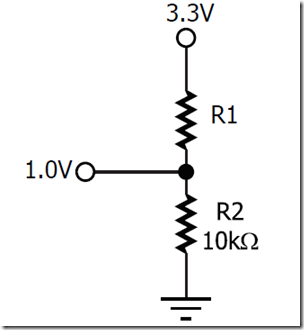
[](http://doc.inex.co.th/wp-content/uploads/2015/12/image6.png)

  ต่อวงจรแปลงสัญญาณอะนาลอกเป็นดิจิตอลเข้ากับบอร์ด Raspberry Pi โดยที่เอาต์พุต CH1 ของ MCP2308 ต่อกับวงจรแบ่งแรงดัน โดยกำหนดค่า R1 คงที่ไว้ที่ 1 กิโลโอห์ม ส่วนค่า R? คือจุดที่ใช้ตรวจวัด โดยจะมีการสุ่มตัวต้านทานที่ใช้ในการวัดจำนวน 5 ค่า มีค่าอยู่ระหว่าง 100 โอห์ม ถึง 10 กิโลโอห์ม ค่าที่อ่านได้จะต้องมีค่าความผิดพลาดไม่เกิน 10%   
     6.1) ค่าความต้านทานเท่ากับ        อ่านค่าได้        (2 คะแนน)  
     6.2) ค่าความต้านทานเท่ากับ        อ่านค่าได้       (2 คะแนน)  
     6.3) ค่าความต้านทานเท่ากับ        อ่านค่าได้        (2 คะแนน)  
     6.4) ค่าความต้านทานเท่ากับ        อ่านค่าได้        (2 คะแนน)  
     6.5) ค่าความต้านทานเท่ากับ        อ่านค่าได้        (2 คะแนน)

**จากโจทย์เขียนเป็นโปรแกรมได้ดังนี้**

**การทำงานของโปรแกรม**

จากกฎของวงจรแบ่งแรงดัน

[](http://doc.inex.co.th/wp-content/uploads/2015/12/image7.png)

**R1 = (R2 \* (Vin – Vo))/Vo**

ค่าของ Vo จะมาจากการอ่านค่าอะนาลอกจาก MCP3208 ซึ่งเป็นแบบ12 บิต แรงดันไฟเลี้ยง 3.3V

**voltage = value\*3.3/4096**

การวัดค่าความต้านในย่าน 1-10 กิโลโอห์มก็ทำได้ไม่ยาก