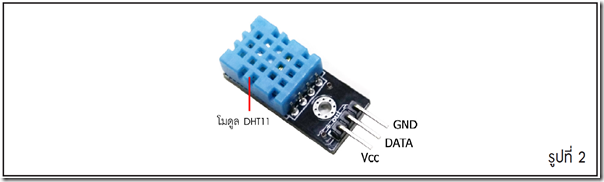
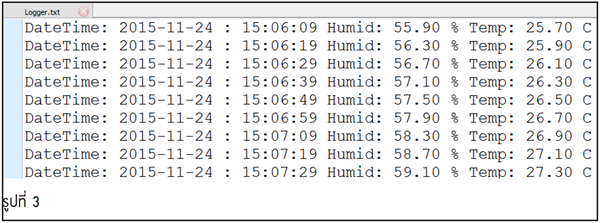
**2. สร้างระบบ   Data  Logger  ในโรงเรือน**

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

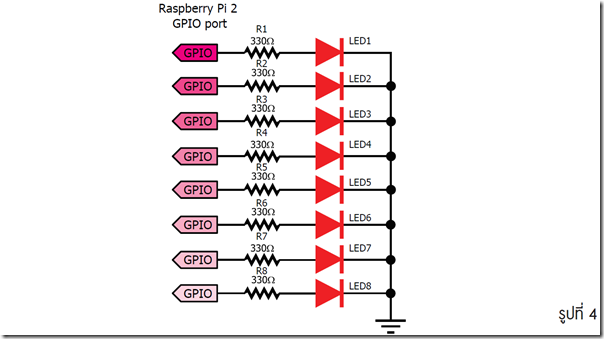
แสดงค่าความเปลี่ยนแปลงความชื้นและอุณหภูมิทุกๆ 10 วินาที โดยทำตามขั้นตอนดังนี้ ทำการต่อโมดูล DHT11 เข้ากับบอร์ด Raspberry Pi จากนั้นเขียนโปรแกรมเพื่ออ่านค่าความชื้นและอุณหภูมิเก็บไว้ในไฟล์ชื่อ Logger.txt โดยรูปแบบการเก็บข้อมูลใน 1 บรรทัดประกอบด้วย วันที่ – เวลา – ค่าความชื้น – ค่าอุณหภูมิ รูปแบบของไฟล์ Text เมื่อเปิดขึ้นมามีรูปแบบดังนี้[](http://doc.inex.co.th/wp-content/uploads/2015/12/image1.png)  
          (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)**

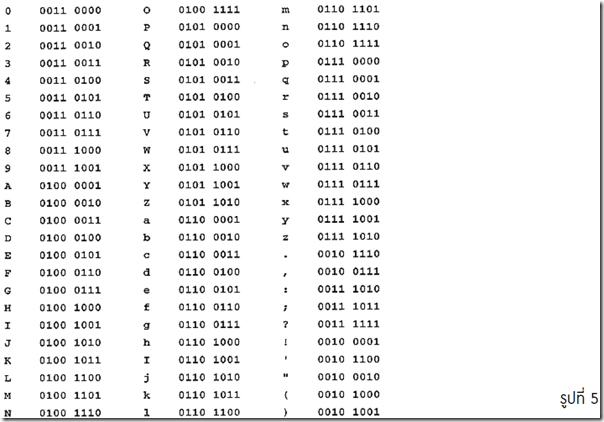
**จากโจทย์เขียนเป็นโปรแกรมได้ดังนี้**

1. **import** time
2. **import** Adafruit\_DHT
3. **import** datetime
4. Sensor = Adafruit\_DHT.DHT11
5. GPIO = 17
6. h=0
7. t=0
8. **while** True:
9. humidity, temperature = Adafruit\_DHT.read\_retry(Sensor,GPIO)
10. **if** humidity **is** **not** None **and** temperature **is** **not** None:
11. *#print('Temp={0:0.1f}\*C Humidity={1:0.1f}%'.format(temperature,humidity))*
12. h=humidity
13. t=temperature
14. **else**:
15. **print**('Failed to get reading. Try again!')
16. date=datetime.datetime.now()
17. **with** open("Logger.txt", "a") **as** text\_file:
18. text\_file.write("DetaTime: %s Humid: %.2f %% Temp: %.2f C**\n**" %(date.strftime("%Y-%M-%d : %H:%M:%S"),h,t))
19. **print**("DetaTime: %s Humid: %.2f %% Temp: %.2f C" % (date.strftime("%Y-%M-%d : %H:%M:%S"),h,t))
20. time.sleep(10)

**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**

1. **import** RPi.GPIO **as** GPIO
2. GPIO.setmode(GPIO.BCM)
3. my\_pin=[6,12,13,19,16,26,20,21]
4. **for** pin **in** my\_pin:
5. GPIO.setup(pin,GPIO.OUT)
6. **while**(1):
7. ascii=input("Aacii=")
8. bin2str=bin(ord(ascii)+256)
9. *#0b1xxxxxxxx*
10. **print**("bin=",bin2str[3:11])
11. GPIO.output(my\_pin[0],int(bin2str[2]))
12. GPIO.output(my\_pin[1],int(bin2str[3]))
13. GPIO.output(my\_pin[2],int(bin2str[5]))
14. GPIO.output(my\_pin[3],int(bin2str[6]))
15. GPIO.output(my\_pin[4],int(bin2str[7]))
16. GPIO.output(my\_pin[5],int(bin2str[8]))
17. GPIO.output(my\_pin[6],int(bin2str[9]))
18. GPIO.output(my\_pin[7],int(bin2str[10]))

หัวใจของโปรแกรมนี้คือการระบุตำแหน่งของ LED ให้อยู่ในรูปแบบของ list หรือ array เพื่อระบุตำแหน่งบิตได้สะดวก

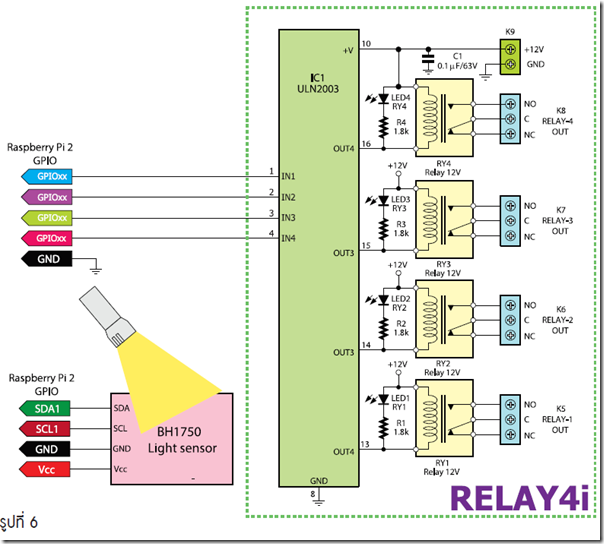
**my\_pin=[6,12,13,19,16,26,20,21]**

อีกคำสั่งคือ ORD สำหรับการเปลี่ยนค่ารหัสแอสกี้ที่อ่านได้จากการกดคีย์บอร์ดให้อยู่ในรูป integer

bin2str=bin(ord(ascii)+256)

**หมายเหตุ** 256 ที่บวกเพิ่มเพื่อให้เวลาแสดงเลขฐาน 2 แล้วเลข 0 ด้านหน้าไม่ถูกตัดหายไป

**4. สวิตช์รหัสกระตุ้นด้วยแสง**

[](http://doc.inex.co.th/wp-content/uploads/2015/12/image4.png)

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)

**จากโจทย์เขียนเป็นโปรแกรมได้ดังนี้**

[SEE ORIGINAL](http://pastebin.com/8YzmGw0f)

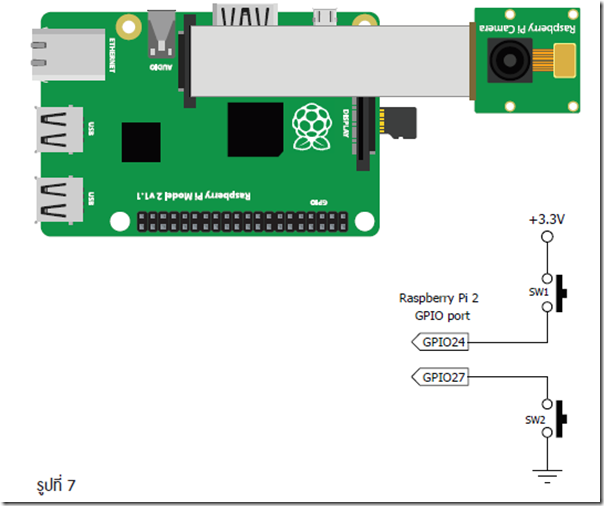
1. *#!/usr/bin/python*
2. *# -\*- encoding: utf-8 -\*-*
3. **import** time
4. **import** smbus
5. bus = smbus.SMBus(1)
6. **import** RPi.GPIO **as** GPIO
7. **import** time
8. GPIO.setwarnings(False)
9. GPIO.setmode(GPIO.BCM)
10. addr = 0x23 *# i2c adress*
11. my\_pin=[16,26,20,21]
12. **for** pin **in** my\_pin:
13. GPIO.setup(pin,GPIO.OUT)
14. st1=0
15. st2=0
16. st3=0
17. st4=0
18. st5=0
19. count=0
20. timer=0
21. **while** True:
22. time.sleep(0.2)
23. data = bus.read\_i2c\_block\_data(addr,0x11)
24. lum=(data[1] + (data[0]<<8) / 1.2)
25. *#print ("Luminosity " ,lum,"lx")*
26. **while** (lum<100):
27. timer=0
28. count+=1
29. data = bus.read\_i2c\_block\_data(addr,0x11)
30. lum=(data[1] + (data[0]<<8) / 1.2)
31. time.sleep(0.2)
32. **while** (lum<100):
33. data = bus.read\_i2c\_block\_data(addr,0x11)
34. lum=(data[1] + (data[0]<<8) / 1.2)
35. time.sleep(0.2)
36. **print**(count)
37. timer+=1
38. *#print ("Timer=",timer)*
39. **if** (timer>10):
40. **if**(count==1):
41. st1=~st1
42. GPIO.output(my\_pin[0],st1)
43. **elif**(count==2):
44. st2=~st2
45. GPIO.output(my\_pin[1],st2)
46. **elif**(count==3):
47. st3=~st3
48. GPIO.output(my\_pin[2],st3)
49. **elif**(count==4):
50. st4=~st4
51. GPIO.output(my\_pin[3],st4)
52. **elif**(count==5):
53. st5=~st5
54. st1=st2=st3=st4=st5
55. GPIO.output(my\_pin[0],st5)
56. GPIO.output(my\_pin[1],st5)
57. GPIO.output(my\_pin[2],st5)
58. GPIO.output(my\_pin[3],st5)
59. 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)

**จากโจทย์เขียนเป็นโปรแกรมได้ดังนี้**

1. **import** RPi.GPIO **as** GPIO
2. **import** time
3. **import** picamera
4. **import** datetime
5. GPIO.setwarnings(False)
6. GPIO.setmode(GPIO.BCM)
7. sw\_1 = 24
8. sw\_2 = 27
9. count= 0
10. GPIO.setup(sw\_1,GPIO.IN,pull\_up\_down=GPIO.PUD\_DOWN)
11. GPIO.setup(sw\_2,GPIO.IN,pull\_up\_down=GPIO.PUD\_UP)
12. GPIO.add\_event\_detect(sw\_1,GPIO.RISING)
13. GPIO.add\_event\_detect(sw\_2,GPIO.FALLING)
14. camera = picamera.PiCamera()
15. **while**(1):
16. **if** GPIO.event\_detected(sw\_1):
17. **print**("SW1")
18. img\_name="/home/pi/Desktop/SW1/"
19. img\_name+=str(datetime.datetime.now())
20. img\_name+=".jpg"
21. camera.capture(img\_name)
22. **if** GPIO.event\_detected(sw\_2):
23. count+=1
24. **print**("SW2")
25. img\_name="/home/pi/Desktop/SW2/Image"
26. img\_name+= str(count).zfill(2)
27. img\_name+=".jpg"
28. 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)

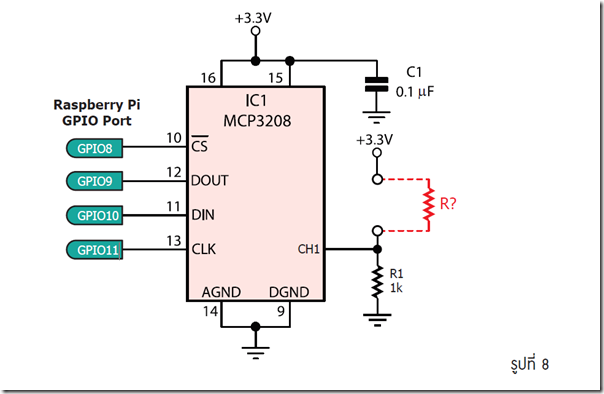
การตรวจสอบการกดสวิตช์ก็สามารถทำผ่านคำสั่ง

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)

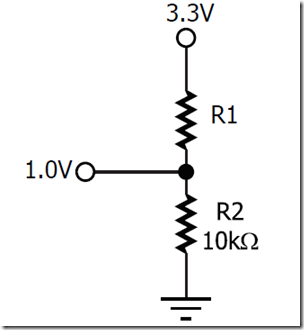
ต่อวงจรแปลงสัญญาณอะนาลอกเป็นดิจิตอลเข้ากับบอร์ด Raspberry Pi โดยที่เอาต์พุต CH1 ของ MCP2308 ต่อกับวงจรแบ่งแรงดัน โดยกำหนดค่า R1 คงที่ไว้ที่ 1 กิโลโอห์ม ส่วนค่า R? คือจุดที่ใช้ตรวจวัด โดยจะมีการสุ่มตัวต้านทานที่ใช้ในการวัดจำนวน 5 ค่า มีค่าอยู่ระหว่าง 100 โอห์ม ถึง 10 กิโลโอห์ม ค่าที่อ่านได้จะต้องมีค่าความผิดพลาดไม่เกิน 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**

1. **import** spidev
2. **import** time
3. analog\_ch = 1
4. spi = spidev.SpiDev()
5. spi.open(0,0)
6. Vin=3.3
7. R1=1000
8. **def** readADC(adcnum):
9. **if** adcnum > 7 **or** adcnum < 0:
10. **return** -1;
11. r = spi.xfer2([4 | 2 | (adcnum >> 2), (adcnum & 3) << 6, 0])
12. adcout = ((r[1] & 15) << 8) + r[2]
13. **return** adcout
14. **while** True:
15. value = readADC(analog\_ch)
16. voltage = value\*3.3/4096
17. **if** voltage > 0:
18. ohm = (R1 \* (Vin-voltage))/voltage
19. *#ohm = ((Vin\*R1)/voltage)-R1*
20. **print**("R? = %d"  % ohm)
21. time.sleep(0.3)

**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