**SMART PUBLIC RESTROOM**

**import requests**

**import time**

**import serial**

**import re**

**import urllib2**

**import datetime**

**import os**

**import sys**

**import RPi.GPIO as GPIO**

**import tm1637**

**ser = serial.Serial(**

**port='/dev/ttyAMA0',**

**baudrate=9600,**

**parity=serial.PARITY\_NONE,**

**stopbits=serial.STOPBITS\_ONE,**

**bytesize=serial.EIGHTBITS,**

**timeout=1**

**)**

**ser.flushInput()**

**CONT = 1**

**send = 0**

**string1 = '{"$id":"Group6\_smart\_toilet","room\_temp":'**

**string2 = ',"room\_lux":'**

**string3 = ',"slot1\_num\_of\_p":'**

**string4 = ',"slot1\_lux":'**

**string5 = ',"slot2\_num\_of\_p":'**

**string6 = ',"slot2\_lux":'**

**string7 = ',"slot3\_num\_of\_p":'**

**string8 = ',"slot3\_lux":'**

**Display = tm1637.TM1637(23, 24, tm1637.BRIGHT\_TYPICAL)**

**Display.Clear()**

**Display.SetBrightnes(5)**

**a = 0**

**b = 0**

**c = 0**

**d = 0**

**while 1:**

**try:**

**data\_string = ser.readline() #DL read data from node 1**

**data\_num = re.findall('\d+(?:\.\d+)?', data\_string) #DL divide string data and allocate to each data array**

**NID = data\_num[1]**

**except:**

**print "serial read error and will try again"**

**CONT = 1**

**send = 0**

**continue #DL when data fail to send the data, this loop return to first of while loop**

**try:**

**if (NID == "1") & (CONT == 1): #YL pass the data from board 1 firstly**

**Temperature1 = data\_num[3] #YL assign value to the Temperature1 from the data\_num[3]**

**Light1 = data\_num[5] #YL assign value to the Light1 from the data\_num[5]**

**if (data\_num[7] == "1") & (len(data\_num[3]) < 9) & (len(data\_num[5]) < 9):#YL receive data only when the CRC=1 and correct length of data**

**print "NODE :", NID, "Temperature :", Temperature1, " Light :", Light1**

**CONT = CONT + 1 #YL Self-added 1 to make sure received data is from board 2**

**elif (NID == "2") & (CONT == 2):#YL pass the data from board 2 secondly**

**User1 = data\_num[3] #YL value the User1 from the data\_num[3]**

**Light2 = data\_num[5] #YL value the Light2 from the data\_num[5]**

**if (data\_num[7] == "1") & (len(data\_num[3]) < 9) & (len(data\_num[5]) < 9):#YL receive data only when the CRC=1 and correct length of data**

**print "NODE :", NID, "Number of people :", User1, " Light :", Light2**

**CONT = CONT + 1**

**elif (NID == "3") & (CONT == 3):#YL pass the data from board 3 thirdly**

**User2 = data\_num[3] #YL value the User2 from the data\_num[3]**

**Light3 = data\_num[5] #YL value the Light3 from the data\_num[5]**

**if (data\_num[7] == "1") & (len(data\_num[3]) < 9) & (len(data\_num[5]) < 9):#YL receive data only when the CRC=1 and correct length of data**

**print "NODE :", NID, "Number of people :", User2, " Light :", Light3**

**CONT = CONT + 1**

**elif (NID == "4") & (CONT == 4):#YL pass the data from board 4 in the end**

**User3 = data\_num[3] #YL value the User3 from the data\_num[3]**

**Light4 = data\_num[5] #YL value the Light4 from the data\_num[5]**

**if (data\_num[7] == "1") & (len(data\_num[3]) < 9) & (len(data\_num[5]) < 9):#YL receive data only when the CRC=1 and correct length of data**

**print "NODE :", NID, "Number of people :", User3, " Light :", Light4**

**CONT = CONT + 1**

**send = 1 #YL value the send=1 when a group of data is received**

**elif NID == "5": #YL CRC=0**

**print "CRC incorrect (You really need to check your CRC)"**

**elif (len(data\_num[3]) > 9) & (len(data\_num[5]) > 9):#YL a series of incorrect and long length data will stop the program**

**print "You realy need reboot your microcontrolers :", NID**

**print "But program will try again "**

**CONT = 1**

**send = 0 #YL initialization of send signal**

**continue**

**#JL the Indexerror usually came when the raspberrypi read the data,**

**# it does not read it from the beginning of each print out but from the middle of it**

**except IndexError:**

**print 'Index error, will flush serial and try again' #YL give a hint as the data stop presenting if IndexError exist**

**ser.flushInput()**

**try:**

**json\_string = string1 + str(Temperature1) + string2 + str(Light1) \**

**+ string3 + str(User1) \**

**+ string4 + str(Light2) \**

**+ string5 + str(User2) \**

**+ string6 + str(Light3) \**

**+ string7 + str(User3) \**

**+ string8 + str(Light4) + '}'**

**except:**

**print('Json\_string incorrect Read again')**

**continue**

**if send == 1:**

**try: #DL upload on cloud(devicepilot) to use a specific url**

**resp = requests.post("https://api.devicepilot.com/devices",**

**headers={"Authorization": "Token 4f07894ac4e9351140f190d3d0dc0696",**

**"Content-Type": "application/json"},**

**data=json\_string)**

**except: #DL when data fail to upload on the cloud, this loop return to first of while loop**

**print "There is a Connection error or error while sending data to cloud"**

**continue**

**print 'Data have been ' + resp.reason + ' by cloud'**

**print (datetime.datetime.now())**

**int\_user1 = int(float(User1))**

**int\_user2 = int(float(User2))**

**int\_user3 = int(float(User3))**

**sum\_of\_p = int\_user1 + int\_user2 + int\_user3**

**if sum\_of\_p<10:**

**d = sum\_of\_p**

**if sum\_of\_p > 9:**

**x = str(sum\_of\_p)**

**c = int(x[0])**

**d = int(x[1])**

**if sum\_of\_p > 99:**

**x = str(sum\_of\_p)**

**b = int(x[0])**

**c = int(x[1])**

**d = int(x[2])**

**if sum\_of\_p > 999:**

**x = str(sum\_of\_p)**

**a = int(x[0])**

**b = int(x[1])**

**c = int(x[2])**

**d = int(x[3])**

**display\_num = [a, b, c, d]**

**Display.Show(display\_num)**

**print str(sum\_of\_p) + ' People had used this toilet'**

**print '==================================='**

**print 'Program will Sleep 2 second '**

**time.sleep(2)**

**ser.flushInput()**

**print 'Serial Flushed'**

**os.system('clear')**

**CONT = 1**

**send = 0**

**a = 0**

**b = 0**

**c = 0**

**d = 0**