|  |  |
| --- | --- |
|  | import time |
|  | import sys |
|  | import ibmiotf.application |
|  | import ibmiotf.device |
|  | import random |
|  |  |
|  | #provide your IBM watson credentials |
|  | organization = "rr1qlb" |
|  | deviceType ="sensordevices" |
|  | deviceId = "swmsfmc1" |
|  | authMethod = "token" |
|  | authToken = "1234567890" |
|  |  |
|  | #Initialize GPIO |
|  | def mycommandCallback(cmd): |
|  | print("Command received: %s" % cmd.data['command']) |
|  | status=cmd.data['command'] |
|  | if status=="alertlighton": |
|  | print("led is on") |
|  | elif status == "alertlightoff": |
|  | print("led is off") |
|  | else: |
|  | print("please send proper command") |
|  |  |
|  | try: |
|  | deviceOptions = {"org": organization, "type": deviceType, "id": deviceId, "auth-method": authMethod, "auth-token": authToken} |
|  | deviceCli = ibmiotf.device.Client(deviceOptions) |
|  | #............................................... |
|  |  |
|  | except Exception as e: |
|  | print("Caught exception connecting device: %s" % str(e)) |
|  | sys.exit() |
|  |  |
|  | # Connect and send a datapoint "hello" with value "world" into thecloud as an event of type "greeting" 10 times |
|  | deviceCli.connect() |
|  |  |
|  | while True: |
|  |  |
|  | sensordata = random.randint(0,200) |
|  |  |
|  | if (sensordata <100 ): |
|  | data= {'Bin is filled' : sensordata} |
|  | else: |
|  | data= {'Bin is not filled' : sensordata} |
|  |  |
|  |  |
|  | #print data |
|  |  |
|  | def myOnPublishCallback(): |
|  | if (sensordata<100): |
|  | print("Bin is filled :%s" %sensordata, "to Ibm Watson") |
|  | else: |
|  | print("Bin is not filled : %s" %sensordata, "to Ibm Watson") |
|  | time.sleep(10) |
|  | success = deviceCli.publishEvent("IoTSensor", "json", data, qos=0, on\_publish=myOnPublishCallback) |
|  | if not success: |
|  | print("Not connected to IoTf") |
|  | time.sleep(10) |
|  |  |
|  | deviceCli.commandCallback = mycommandCallback |
|  |  |
|  | #Disconnect the device and application from the cloud |
|  | deviceCli.disconnect() |