```
#!/usr/bin/env python
import cayenne.client, datetime, time, serial, logging

# Cayenne authentication info. This should be obtained from the Cayenne Dashboard.
MQTT_USERNAME  = "6375a470-cff9-11e7-86d0-83752e057225"
MQTT_PASSWORD  = "26e1dc13f900da7b30b24cad4b320f9bc6dd0d78"
MQTT_CLIENT_ID = "900747c0-532e-11ec-bbfc-979c23804144"

# Default location of serial port on pre 3 Pi models
#SERIAL_PORT =  "/dev/ttyAMA0"

# Default location of serial port on Pi models 3 and Zero
SERIAL_PORT =   "/dev/ttyS0"

#This sets up the serial port specified above. baud rate is the bits per second timeout seconds
#port = serial.Serial(SERIAL_PORT, baudrate=2400, timeout=5)

#This sets up the serial port specified above. baud rate and WAITS for any cr/lf (new blob of
data from picaxe)
port = serial.Serial(SERIAL_PORT, baudrate=2400)

client = cayenne.client.CayenneMQTTClient()
client.begin(MQTT_USERNAME, MQTT_PASSWORD, MQTT_CLIENT_ID, loglevel=logging.INFO)
#client.begin(MQTT_USERNAME, MQTT_PASSWORD, MQTT_CLIENT_ID, loglevel=logging.DEBUG)
# For a secure connection use port 8883 when calling client.begin:
# client.begin(MQTT_USERNAME, MQTT_PASSWORD, MQTT_CLIENT_ID, port=8883, loglevel=logging.INFO)

qos = 1
qos_good = 1
qos_bad = 1
timestamp = 1

while True:
  client.loop()
  try:
    rcv = port.readline() #read buffer until cr/lf
    #print("Serial Readline Data = " + rcv)
    rcv = rcv.rstrip("\r\n")
    synch,node,channel,data,cs = rcv.split(",")
    #print("rcv.split Data = : " + node + " " + channel + " " + data + " " + cs)
    #time.sleep(1)
    #Pacing delay to control rate of upload data

    checkSum = int(node) + int(channel) + int(data) %256
    cs = int(cs)
    #print(checkSum,cs)

    time.sleep(.1)
    #Wait a bit for Serial Ports
    port.write(str(checkSum) + '\r\n')
    #Send something back from the Pi serial com port

    if checkSum == cs:
```

```
      qos_good = qos_good + 1
    else :
      qos_bad = qos_bad + 1

    if (time.time() > timestamp):
      qos = float(qos_good) / (qos_good + qos_bad)      #Calculate error rate ratio
      qos = round(qos,2) * 100                  #Convert qos ratio to Percent
      client.virtualWrite(25,qos,"analog_sensor", "null")
      qos_good  = 1
      qos_bad   = 1
      timestamp = time.time() + 300                 #Every 300 seconds = 5 minutes

    if checkSum == int(cs) :
    #if cs = Check Sum is good then do the following

      if channel == '1':
        data = float(data)/1
        client.virtualWrite(1, data, "analog_sensor", "null")

      if channel == '2':
        data = float(data)/1
        client.virtualWrite(2, data, "analog_sensor", "null")

      if channel == '3':
        data = float(data)/1
        client.virtualWrite(3, data, "analog_sensor", "null")

      if channel == '4':
        data = float(data)/1
        client.virtualWrite(4, data, "analog_sensor", "null")

      if channel == '5':
        data = float(data)/1
        client.virtualWrite(5, data, "analog_sensor", "null")

      if channel == '6':
        data = float(data)/1
        client.virtualWrite(6, data, "analog_sensor", "null")

      if channel == '7':
        data = float(data)/1
        client.virtualWrite(7, data, "analog_sensor", "null")

      if channel == '8':
        data = float(data)/1
        client.virtualWrite(8, data, "analog_sensor", "null")

      if channel == '9':
        data = float(data)/1
        client.virtualWrite(9, data, "analog_sensor", "null")

      if channel == '10':
        data = float(data)/1
```

```
      client.virtualWrite(10, data, "analog_sensor", "null")

    if channel == '11':
      data = float(data)/10
      client.virtualWrite(11, data, "analog_sensor", "null")

    if channel == '12':
      data = float(data)/10
      client.virtualWrite(12, data, "analog_sensor", "null")

    if channel == '13':
      data = float(data)/10
      client.virtualWrite(13, data, "analog_sensor", "null")

    if channel == '14':
      data = float(data)/10
      client.virtualWrite(14, data, "analog_sensor", "null")

    if channel == '15':
      data = float(data)/1
      client.virtualWrite(15, data, "analog_sensor", "null")

    if channel == '16':
      data = float(data)/1
      client.virtualWrite(16, data, "analog_sensor", "null")

    if channel == '17':
      data = float(data)/1
      client.virtualWrite(17, data, "analog_sensor", "null")

    if channel == '18':
      data = float(data)/1
      client.virtualWrite(18, data, "analog_sensor", "null")

    if channel == '19':
      data = float(data)/1
      client.virtualWrite(19, data, "analog_sensor", "null")

    if channel == '20':
      data = float(data)/1
      client.virtualWrite(20, data, "analog_sensor", "null")

    if channel == '21':
      data = float(data)/1
      client.virtualWrite(21, data, "analog_sensor", "null")

    if channel == '22':
      data = float(data)/1
      client.virtualWrite(22, data, "analog_sensor", "null")

    if channel == '23':
      data = float(data)/1
      client.virtualWrite(23, data, "analog_sensor", "null")
```

```
    if channel == '24':
      data = float(data)/1
      client.virtualWrite(24, data, "analog_sensor", "null")

    if channel == '25':
      data = float(data)/1
      client.virtualWrite(25, data, "analog_sensor", "null")

    if channel == '26':
      data = float(data)/1
      client.virtualWrite(26, data, "analog_sensor", "null")

    if channel == '30':
      data = float(data)/1
      client.virtualWrite(30, data, "analog_sensor", "null")

    if channel == '31':
      data = float(data)/1
      client.virtualWrite(31, data, "analog_sensor", "null")

    if channel == '32':
      data = float(data)/1
      client.virtualWrite(32, data, "analog_sensor", "null")

    if channel == '33':
      data = float(data)/1
      client.virtualWrite(33, data, "analog_sensor", "null")

except ValueError:
  #error = error + 10
  #error = float(error)/1
  #client.virtualWrite(22,error)/1
  #if Data Packet corrupt or malformed then...
  print("Data Packet corrupt or malformed")
```