```
import paho.mqtt.publish as publish
import psutil
import serial
import time

arduinoData = serial.Serial("/dev/tty.usbmodem14101",9600,write_timeout=2,timeout=2);
channelID = "1072373"
apiKey = "75Y#########4"
useUnsecuredTCP = False
useUnsecuredWebsockets = False
useSSLWebsockets = True
mqttHost = “mqtt.thingspeak.com"
topic = "channels/" + channelID + "/publish/" + apiKey

if useUnsecuredTCP:
    tTransport = "tcp"
    tPort = 1883
    tTLS = None

if useUnsecuredWebsockets:
    tTransport = "websockets"
    tPort = 80
    tTLS = None

if useSSLWebsockets:
    import ssl
    tTransport = "websockets"
    tTLS = {'ca_certs':"/etc/ssl/certs/ca-certificates.crt",'tls_version':ssl.PROTOCOL_TLSv1}
    tPort = 443

def realizar_mensuração(comando):
  arduinoData.write()
  medida = arduinoData.readline().decode('ascii')
  return medida

def espera(segundos):
    time.sleep(segundos)

while(True):
    temperatura = realizar_mensuração(b't')
    pressao = realizar_mensuração(b'p')
    umidade = realizar_mensuração(b'u')
    dir_vento = realizar_mensuração(b'd')
    vel_vento = realizar_mensuração(b'a')

    tPayload = "field1=" + str(temperatura) + "&field2=" + str(pressao) + "&field3=" + str(dir_vento) +  "&field4=" +
str(temperatura) +  "&field5=" + str(vel_vento)

    try:
        publish.single(topic, payload=tPayload, hostname=mqttHost, port=tPort, tls=tTLS, transport=tTransport)
        espera(30)
    except (KeyboardInterrupt):
        break
    except:
        print (“Erro ao publicar os dados")
```