```
#!/usr/bin/python
#
#simple app to read string from serial port
#and publish via MQTT
#
#uses the Python MQTT client from the Mosquitto project
#http://mosquitto.org
#
#Andy Piper http://andypiper.co.uk
#2011/09/15

import serial
import mosquitto
import os

serialdev = '/dev/ttyUSB0'
broker = "127.0.0.1"
port = 1883


#MQTT callbacks

def on_connect(rc):
    if rc == 0:
    #rc 0 successful connect
        print "Connected"
    else:
        raise Exception


def on_publish(val):
    print "Published ", val


#called on exit
#close serial, disconnect MQTT

def cleanup():
    print "Ending and cleaning up"
    ser.close()
    mqttc.disconnect()


try:
    print "Connecting... ", serialdev
    #connect to serial port
    ser = serial.Serial(serialdev, 9600, timeout=20)
except:
    print "Failed to connect serial"
    #unable to continue with no serial input
    raise SystemExit


try:
    ser.flushInput()
    #create an mqtt client
    mypid = os.getpid()
    client_uniq = "arduino_pub_"+str(mypid)
    mqttc = mosquitto.Mosquitto(client_uniq)

    #attach MQTT callbacks
    mqttc.on_connect = on_connect
    mqttc.on_publish = on_publish

    #connect to broker
    mqttc.connect(broker, port, 60, True)

    #remain connected to broker
```

```
    #read data from serial and publish
    while mqttc.loop() == 0:
        line = ser.readline()
        #split line as it contains V,temp
        list = line.split(",")
        #second list element is temp
        temp = list[1].rstrip()
        mqttc.publish("arduino/temp", temp)
        pass


# handle list index error (i.e. assume no data received)
except (IndexError):
    print "No data received within serial timeout period"
    cleanup()
# handle app closure
except (KeyboardInterrupt):
    print "Interrupt received"
    cleanup()
except (RuntimeError):
    print "uh-oh! time to die"
    cleanup()
```