Rocket Lab USA, Inc.
25 Levene Place
Mount Wellington
Auckland 1060, NZ

P: +64 9 373 2721
rocketlabusa.com

Production Automation Coding Test – Test Devices

Introduction

This test involves a couple of simple tasks related to communicating with test devices. We will be looking at coding style, consistency, correctness, robustness, usability, and adherence to the requirements. We prefer that you don't use 3rd-party libraries unless necessary.

Requirements/Information

Assume that the applications run on a Linux system.

We added the source code for the device simulator (device.c) and a makefile.
Compile: *make*
Usage: *device [-H Host] [-P Port]*
Command line options:

-H	--host	<host>	Hostname/IP to bind (default: 0.0.0.0)
-P	--port	<u16>	UDP port to bind (default: 0)
-M	--model	<str>	Model number (default: M001)
-S	--serial	<str>	Serial number (default: SN0123456)
	--mcast-addr	<ip>	IP for multicast group to join (default: 224.3.11.15)
	--mcast-port	<u16>	UDP port for multicast socket (default: 31115)
	--mv	<float>	DUT reported mV (default: 4500.000000)
	--ma	<float>	DUT reported mA (default: 100.000000)
	--deterministic		Remove DUT mV/mA randomness (default: false)
-v	--verbose		Debug logging
-h	--help		Show this help text

Task: Communicate with a test measurement device

Write a GUI program using Python 3 and PyQt5. It should allow the user to interact with the delivered test device:

- discover devices on the network. See Appendix for description.
- select a device
- define test duration
- start a test
- force a test to stop early
- see live plot of measured values during the test
- support testing multiple devices independently

Rocket Lab USA, Inc.
25 Levene Place
Mount Wellington
Auckland 1060, NZ

P: +64 9 373 2721
rocketlabusa.com

Deliverables

Your deliverables are:

- Source code for test program
- Instructions for building and running the test program
- Justification and considerations for use of any 3rd-party library
- Usage notes to allow running the test program

These may be delivered via zipped Git repo, published Github repo, or any other suitable mechanism.

Appendix: Device protocol and behaviour

- The device listens for UDP packets. Unknown messages are ignored by the device.
- Messages other than Discovery are ignored if they arrive to the multicast address.
- The device sends responses to the IP address and port number from which the request was received.
- All messages are strings consisting of keywords and values separated by semicolons. All numerical values are integers. The strings are encoded using ISO-8859-1 (Latin 1).

Packet descriptions are as follows (uppercase indicates literal values, lowercase indicates values to be filled in as appropriate):

A. Discovery

A discovery message is sent to the multicast address 224.3.11.15 port 31115, or to the address:port of a specific device.

"ID;"

Any device that receives a discovery message will respond with its model ID and serial number:

"ID;MODEL=m;SERIAL=n;"

B. Testing

Start a test of the given duration, with status reporting at the specified rate:

"TEST;CMD=START;DURATION=s;RATE=ms;"

DURATION is test duration, in seconds
RATE is how often the device should report status during the test, in milliseconds

The test will stop after the given duration, or when device receives the stop command:

"TEST;CMD=STOP;"

The start and stop commands will result in one of the following responses:

"TEST;RESULT=STARTED;" - *the test was started successfully*
"TEST;RESULT=STOPPED;" - *the test was stopped successfully*
"TEST;RESULT=error;MSG=reason;" - *a test was already running, or was already stopped*

Rocket Lab USA, Inc.
25 Levene Place
Mount Wellington
Auckland 1060, NZ

P: +64 9 373 2721
rocketlabusa.com

C. Status

While the test is running, the device will send status messages at the specified rate:

"STATUS;TIME=ms;MV=mv;MA=ma;"

TIME is milliseconds since test start
MV, MA are millivolts and milliamps, respectively.

(it's not important what values Volts/Amps the devices sends back – it can be configured to use random or constant values --deterministic option)

After the test has finished (or if the test was stopped), the device will send one final status message:

"STATUS;STATE=IDLE;"