JRP ACQ-PRO

Towards the propagation of AC Quantum Voltage Standards

WP 2: Good Practice Guide on the operation of AC quantum voltage standards

Activity A2.1.9, Report: Requirements to use common software for the AC quantum voltage infrastructures

CMI

February 2016

Contents

1 Introduction

The following report is on the definition of the requirements to use common software for the AC quantum voltage infrastructures. This involves monitoring and control of the AC quantum voltage standard (and automation of the measurements taken by the system under calibration (including RMS and direct sampling).

The document describes requirements, which were identified based on the experience with PTB and NPL systems and on the experience of developing metrology grade software in CMI.

The requirements can be divided into main two parts. Requirements are based on the:

- Measurement system - requirements based on the hardware of the standard and on the common operations, such as calibrations and maintenance.
- Good coding practice - requirements based on a set of informal rules that the software development community has learned over time which can help improve the quality of software.

Due to large hardware differences, the requirements are also specified according the measurement system. Following measurement systems are considered:

- Programmable Josephson System (PJVS) used either as a voltage generator or a voltmeter.
- Josephson Arbitrary Waveform System (JAWS).

2 Requirements based on the Measurement system

2.1 Required controlled devices

Common software should be able to operate a hardware commonly used in metrological laboratories. The following list is based on systems operated in NIST and PTB.

For the case of PJVS:

- Microwave source:
  - Jülicher SQUID Miniature Millimeter Wave Synthesizer
  - Phase Matrix EIP 54X counter with Gunn diode
- Bias sources:
  - Active Technologies (LeCroy) Arbitrary Waveform Generator AT-AWG 1102/1104

  - Agilent E8257D-528

- Null meters:
  - Keysight (Agilent) 3458A
  - National Instruments 5922
  - National Instruments 6230

For the case of JAWS:

- Pattern Generator:
  - Sympuls Aachen Pattern Generator BPG 30G-TER
- Compensation source:
  - Keysight (Agilent) 33220A
- Spectrum analyzer:
  - National Instruments 5922
- Clock source:
  - Keysight (Agilent) E8247C

2.2 Required operations with devices

Common software should be able to do following operations with a hardware commonly used in metrological laboratories. The

For the PJVS system:

- Microwave source:
  - Switch on/off.
  - Set output power (attenuator).
- Bias source:
  - Set current for selected segment.
  - Load waveform.
- Null meter:
  - Sample data.

  - Set sampling frequency.

For the JAWS system:

- Pattern generator:
  - Load pre-calculated waveform.
  - Change pulse amplitude.
  - Select no pulses.
- Compensation generator:
  - Load pre-calculated waveform.
  - Change output amplitude.
  - Set triggering mode and shift phase.
- Clock source:
  - Set clock frequency.
  - Switch on/off.
- Spectrum analyzer:
  - Measure spectrum in the defined bandwidth.
  - Measure time series of output voltage.

2.3 Required tasks

Common software should to be able not only to do simple operations with measurement devices, but also to do specific tasks and operate the whole system as a unit. E.g. to measure properties of the chip, to check stability of devices under test or to do a maintenance of the system.

Following tasks are required. For the case of PJVS:

- Measure chip IV characteristics with and without applied microwave power.
- Measure critical current of the chip.
- Measure step width of all segments.
- Start/stop generating a required waveform.
- Generate DC voltage.

- Determine transients ringing time.
- Calibrate linearity of multimeter/digitizer DC/AC.
- Calibrate RMS of calibrator DC/AC.

For the case of JAWS:

- Measure chip IV characteristics with and without applied microwave power.
- Start/stop generating a required waveform.
- Determine optimal pulse power and compensation current.
- Generate DC voltage.
- Calibrate linearity of multimeter/digitizer DC/AC.
- Calibrate RMS of calibrator DC/AC.

3 Requirements based on the Good coding practice

3.1 Extensive documentation

The documentation is the key for any software being developed by more than one man for more than one day. This has to be stressed for any scientific software, thus the software could be thoroughly checked and reviewed.

Folowing should be part of documentation:

- Descriptions of all functions and modules.
- Flowcharts of operations, tasks and algorithms.
- Document about how to use the software.
- Description of abstraction layers.

3.2 Verifiability

If the common software will be used in calibrations, it must be possible to verify the software calculates results correctly.

It can be achieved by implementing simulation of measurement devices with predefined "measured" values and checking that the software gives the same result as calculated externally.

3.3 Modularity

The common software has to be modular. Because the measurement system evolves, components are added, removed or changed, one needs an easy way to add, remove or change parts of the software without affecting the rest. This is the key requirement to keep the software being used as the time goes on.

3.4 Hardware abstraction layer

The common software needs to be used with a variety of measurement devices. For example as a null meter several digitizers can be used. However the same actions are required with the device: setup the device, start the sampling, acquire samples. With a Hardware abstraction layer it is easy to change the software to use different device for the same operation.

3.5 Operations abstraction layer

An abstraction layer between operations and tasks enhance the modularity. It will also simplify creation of new tasks for the case of implementation of new measurement methods.

3.6 Application programming interface

For day to day calibrations a variety of systems are available to maintain track of devices under test, issue calibration certificates and manage calibration services. These systems are usually unique for a calibration institute. . Such systems

3.7 Open source software

Whole software should be open source, or at least source code should be accessible to end-users. Typically a petrological software of primary standards have to be modified to fit current hardware or methods of use. Also end-user require the possibility to check all calculations are correct otherwise the traceability of the system cannot be assured.

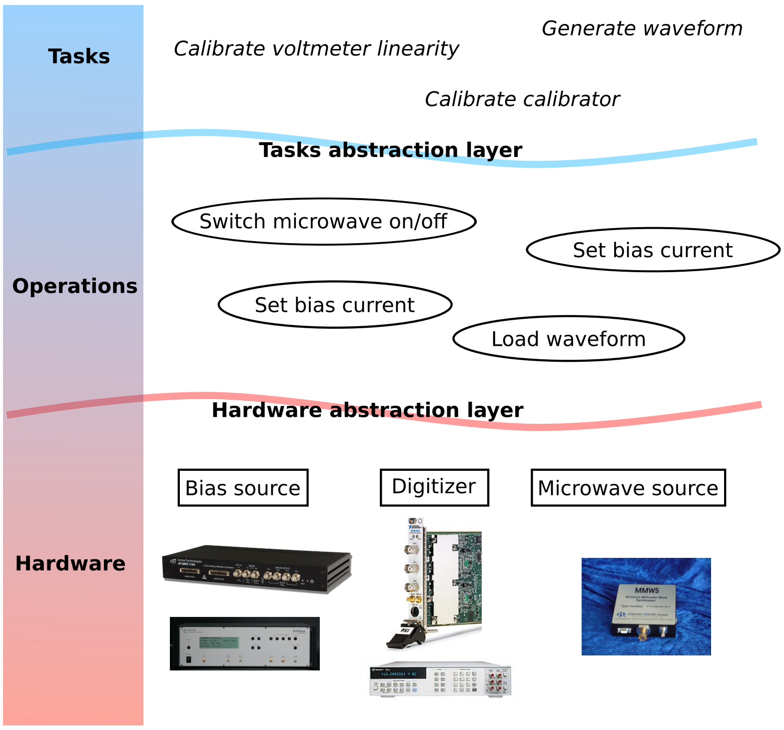
3.8 Language dependent recommendation

For every programming language a set of good programming recommendations can be found. For example good variable naming reduces the effort needed to read and understand source code, and National Instruments provide this recommendation for Labview at `http://www.ni.com/white-paper/5560/en/#toc5`.

4 Proposals

4.1 General scheme

The proposed general scheme of common software should contain previously mentioned two abstraction layers:



4.2 Programming language

Part of EMPIR project TracePQM aims at joint software development of electrical power measurement system. After a discussion it was found only 2 from 13 european national metrological institutes requires different programming language than LabVIEW. This language have some defficiencies, however is very easy to use for a non-professional programmers.

http://www.acqpro.cmi.cz

This document was prepared during project ACQ-PRO. The project ACQ-PRO has received funding from the EMPIR programme co-financed by the Participating States and from the European Union's Horizon 2020 research and innovation programme. This collection reflects only the author's view and EURAMET is not responsible for any use that may be made of the information it contains.

The EMPIR initiative is co-funded by the European Union's Horizon 2020 research and innovation programme and the EMPIR Participating States

Requirements to use common software for
the AC quantum voltage infrastructures
Martin Šíra
2016
Typeset in pdfLaTeX
Published by Czech Metrology Institute