Using *LabGenie* software for temperature control, mass spectrometry and TPD

The *LabGenie* software was written to interface with several types of laboratory hardware devices, and to enable quick and straightforward set-up for temperature measurement, analog voltage measurement (e.g. from a baratron), temperature control, mass spectral data acquisition, and temperature programmed desorption (TPD).

The following hardware devices are supported:

- **National Instruments data acquisition boards (NI-DAQ)** such as NI-6211 and NI-6229. *LabGenie* can be configured to read voltages from analog input channels ("ai1" etc.) and write voltages to analog output channels ("ao1" etc.). Voltage ranges are limited to -10V to 10V.
- **National Instruments thermocouple reader NI-9211.** This is a fairly expensive thermocouple (TC) readout device that can read from up to 4 thermocouples. The TC contacts are isolated and open TC circuit shows as maximum calibration temperature for that TC type.
- **Phidgets-1048 thermocouple reader.** This is an affordable 4-thermocouple readout from Phidgets, Inc. The TC inputs are not isolated, and in general an additional USB isolator (such as Phidgets-3036) is required for an independently grounded thermocouple. Open TC circuit results in a reading of approximately 600 K, which can potentially lead to overheating if one TC wire accidentally detaches.
- **Phidgets-1002** is a four channel analog voltage output device (-10V to 10V voltage range). It is a much less expensive option for analog output compared to a NI-DAQ board such as NI-6229.
- **Hiden HAL quadrupole mass spectrometer (QMS).**

Configuring *LabGenie*

The software is configured by editing a text file called **"config.xml"** in the software's directory. The file is written in the XML format and represents a hierarchical structure, or a tree, of **nodes**. Each node begins with an opening tag *(<nodeName>)* and ends with a closing tag *(</nodeName>)*:

```
<nodeName> … Node data and other nodes … </nodeName>
```

XML files may be edited with a specialized XML editor or a simple text editor, however caution must be exercised in the latter case since any errors in node names or node closing tags will result in *LabGenie* not being able to read the configuration file.

Ignoring for the moment the opening and closing tags, the overall structure of the configuration file is given by the following tree of nodes:

```
config
   devices
      device1
      device2
      …
   screens
      screen1
      screen2
      …
```

That is, the main node named *<config>* encloses two other nodes, named *<devices>* and *<screens>.* These three nodes are present in all configurations. The *<devices>* node contains the descriptions of the devices that the user wishes to create. In this case, there are two devices present, as defined by nodes *<device1>* and *<device2>*.

The *<screens>* node lists the different tabs that the software will display on screen. Each device may be assigned to be displayed on one of these tabs.

The order in which devices and screens are defined within their nodes is unimportant. Changing the definition order of will result in the same configuration.

For the sake of example, the configuration listing below specifies a single device called *BaratronReader* that is shown on a screen (tab) named *ScreenBaratron*:

```
<config>
   <devices>
      <BaratronReader>
         <type>NiDaqAnalogReader</type>
         <NIdeviceName>Dev1</NIdeviceName>
         <channel>0</channel>
         <mode>Diff</mode>
         <show>true</show>
         <screen>ScreenBaratron</screen>
      </BaratronReader>
   </devices>

   <screens>
      <ScreenBaratron>
         <label>Baratron reader</label>
      </ScreenBaratron>
   </screens>
</config>
```

Let us walk through this configuration tree node by node. First of all, everything is enclosed in the root node called *<config>*, which has two child nodes - *<devices>* and *<screens>*. The *<devices>* node has a single child called *<BaratronReader>*. The *<BaratronReader>* node has a number of child nodes that serve to describe the device. Its type is *NiDaqAnalogReader*, and it reads from National Instruments device called *Dev1*, voltage channel 0, in differential voltage measurement mode. The *<show>true</show>* child node requests that the device be shown on screen (as devices are not shown by default), and the *<screen>ScreenBaratron</screen>* node requests that the device be shown on a tab named *ScreenBaratron*.

The above configuration file, with the hardware properly connected, produces an analog reader interface shown below:

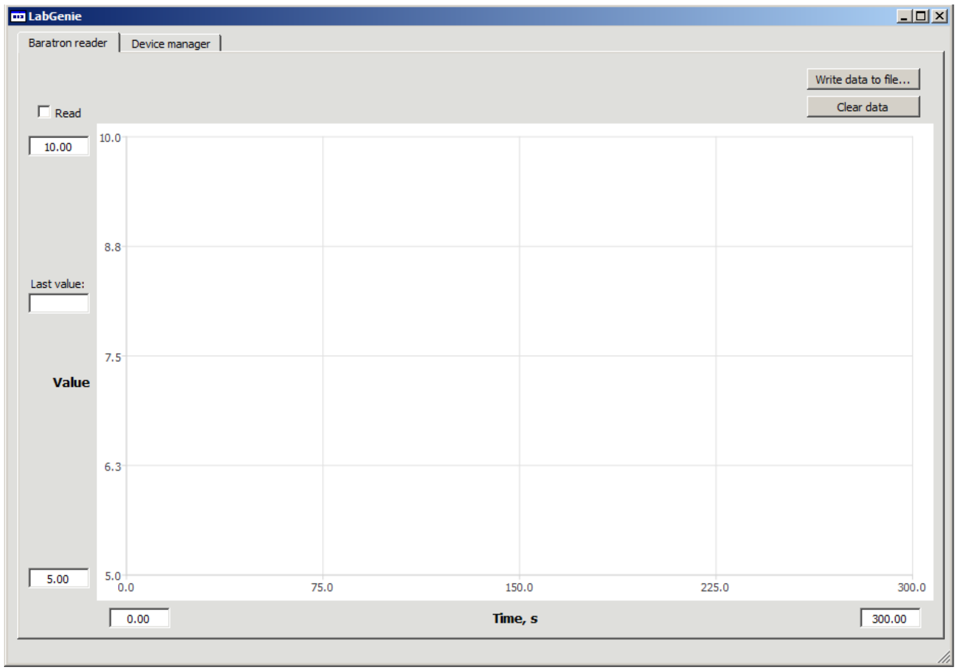


Figure 1. LabGenie interface for an analog reader.

Configuring screen nodes

Screen node configuration is very straightforward. Each of the screen nodes (such as the *<ScreenBaratron>* in the example above) has only a single child node named *<label>*. The label is simply the text that appears at the top of the corresponding tab in the user interface. In the example above, the text is "Baratron reader".

Configuring device nodes: child nodes common to all devices

The configuration options for device nodes are more numerous. The options that are common for all devices are specified in the following child nodes:

- *<type>* - required
- *<show>* - optional, false by default
- *<screen>* - optional unless <show> is set to true
- *<screenX>* - optional, 0 by default
- *<screenY>* - optional, 0 be default
- *<showBox>* - optional, false by default
- *<boxText>* - optional, empty by default

The exact purpose of these child nodes is described in the following sections.

<u><type></u>

This is a required child node for every device. The type string is one of the following:

- *HwNI9211* - for National Instruments NI-9211 reader hardware
- *HwPhidgets1048* - for Phidgets 1048 temperature reader hardware
- *HwPhidgets1002* - for Phidgets 1002 analog voltage output hardware
- *TempReaderNI9211* - a NI-9211 logical reader device (attaches to *HwNI9211*)
- *TempReader1048* - a Phidgets 1048 logical reader device (attaches to *HwPhidgets1048*)
- *WriterPhidgets1002* - a Phidgets 1002 logical writer device (attaches to *HwPhidgets1002*)
- *NiDaqAnalogReader* - a voltage reader for a NI-DAQ board
- *NiDaqAnalogWriter* - a voltage writer for a NI-DAQ board
- *TempController* - a temp controller logical device (needs a temp reader and a voltage writer)
- *QmsHidenHAL* - a Hiden HAL QMS device
- *Tpd* - a TPD device (needs a QMS device and a temperature controller)

An example for type specification would be *<type>QmsHidenHAL</type>*.

<u><show></u>

This option specifies whether the device will be shown in the user interface on one of the screens (tabs). By default, this option is set to false. Only the following device types can be shown: TempReaderNI9211, TempReader1048, NiDaqAnalogReader, TempController, QmsHidenHAL, Tpd. If this option is set for any other device type, it will be ignored.

Example for setting this option: *<show>true</show>*.

<u><screen></u>

This option indicates which screen the device will be shown on if *<show>* is set to true. The corresponding screen must exist in the *<screens>* node.

An example: *<screen>ScreenBaratron</screen>.*

<u><screenX> and <screenY></u>

These two options enable several devices to be shown on a single screen (tab). In case several devices are pointed to the same screen, they should have different values of *<screenX>* and *<screenY>*. Then the screen will be divided between them in a rectangular grid, and each of these devices will be shown at coordinates of the grid corresponding to their *<screenX>* and *<screenY>* values.

For instance, one of the devices could have *<screenX>* set to 0 and *<screenY>* set to 0, while another could have *<screenX>* set to 0 but *<screenY>* set to 1. They devices will then be shown on a vertically split screen one above the other.

By default, both values are set to 0. An example specification for this option:

```
<screenX>0</screenX>
<screenY>1</screenY>
```

<u><showBox></u>

This options indicates whether to draw a box around the device shown on the interface. It is useful for visually separating the devices if more than one of them are shown on the same screen. By default, this option is set to false.

Example: *<showBox>true</showBox>*

<u><boxText></u>

If the *<showBox>* option is set to true, the box around the device on a screen can be labeled with a text. Again, this is useful when several devices are shown on the same screen.

Example: *<boxText>Baratron no. 1</boxText>*

Configuring device nodes: type-specific options

In addition to the options common to all devices, some options are required for specific device types.

<u>HwNI9211</u>

This device enables communication with the NI-9211 hardware connected to the PC. An additional option *<NIdeviceName>* must be specified that indicates which NI device in particular is to be connected to. It can be read from the Devices tab in National Instruments NI-MAX software utility. The device name can also be changed using that utility. Example definition:

```
<MyDevice>
        <type>HwNI9211</type>
        <NIdeviceName>Dev1</NIdeviceName>
</MyDevice>
```

<u>HwPhidgets1048 and HwPhidgets1002</u>

These devices require the serial number of the particular Phidgets 1048 and Phidgets 1002 to be specified. The serial number can be read on the hardware board itself or, if it's not written on the board, from the Phidgets Control Panel utility. Example of device definition:

```
<MyPhidgetsOutput>
        <type>HwPhidgets1002</type>
        <serial>493707</serial>
</MyPhidgetsOutput>
```

TempReaderNI9211 and TempReader1048

For these devices that read temperature from a thermocouple, three additional options have to be specified: the hardware device to connect this reader to *(<hardware>*), the thermocouple type (*<tcType>*), and channel to which the thermocouple in question is attached. Both NI-9211 and Phidgets 1048 have four channels numbered 0 to 3, therefore up to four readers can be simultaneously connected to each hardware device.

These reader devices by default output temperature in Kelvin. If a different output scale is needed, two additional parameters can be specified, named <a> and <b>. The reader will then produce output that is equal to $a * x + b$, where x is the original unmodified reading. By default, <a> is set to 1, and <b> to 0.

The allowable thermocouple types are:

- For NI-9211: J, K, N, R, S, T, B, E
- For Phidgets 1048: J, K, T, E

Therefore a configuration file for a K-type thermocouple reader connected to channel 2 of an NI-9211 device should look as follows:

```
<config>

   <devices>
      <My9211>
         <type>HwNI9211</type>
         <NIdeviceName>Dev1</NIdeviceName>
      </My9211>
      <SampleTempReader>
         <type>TempReaderNI9211</type>
         <hardware>My9211</hardware>
         <channel>2</channel>
         <tcType>K</tcType>
         <a>1</a>
         <b>0</b>
         <show>true</show>
         <screen>TempReaderScreen</screen>
      </SampleTempReader>
   </devices>

   <screens>
      <TempReaderScreen>
         <label>Temperature reader ch 2</label>
      </TempReaderScreen>
   </screens>

</config>
```

This configuration produces a user interface similar to that depicted in Figure 1.

<u>WriterPhidgets1002</u>

This is a logical writer device that connects to a hardware device of type HwPhidgets1002. There are 4 channels on the hardware device, numbered 0 to 3. Therefore up to 4 logical writers may connect to a single hardware device. When specifying the writer, two additional options are needed: the channel to connect to, and the name of the hardware. Below are example definitions of a writer and its hardware:

```
<MyPhidgets1002>
        <type>HwPhidgets1002</type>
        <serial>493756</serial>
</MyPhidgets1002>
<WriterDoser>
        <type>WriterPhidgets1002</type>
        <hardware>MyPhidgets1002</hardware>
        <channel>1</channel>
</WriterDoser>
```

<u>NiDaqAnalogReader</u>

This device reads voltages from National Instruments NI-DAQ cards. Several additional parameters can be specified, two of which are required. If optional parameters aren't specified, default values are used:

- *<NIdeviceName>*, which can be read (and set) from the National Instruments NI-MAX software (required)
- The *<channel>* from which to read the voltage (required). Number of channels varies for different NI-DAQ devices.
- The *<samplingRate>* during data acquisition, in samples per second (default = 20,000 samples/s)
- The number of samples to average per one reading (*<averagingSamples>*); default = 100
- Minimum voltage expected, in volts (*<minVolt>*); default = -10 V
- Maximum voltage expected, in volts (*<maxVolt>*); default = 10 V
- Read period, in seconds - how frequently to read. Default = 0.1 s (10 times a second)
- Acquisition mode, one of:
  - Diff - differential (default)
  - RSE - referenced single-ended
  - NRSE - non-referenced single-ended
  - PseudoDiff - pseudo-differential
- As with other types of readers, it's possible to specify linear constants <a> and <b> that modify the output from the reader according to $a * x + b$, where x is the unmodified output. By default, <a> is set to 1, and <b> is set to 0.

An example of specifying a NI-DAQ analog reader:

```
<AnalogReader>
      <type>NiDaqAnalogReader</type>
      <NIdeviceName>Dev2</NIdeviceName>
      <channel>0</channel>
      <mode>Diff</mode>
      <samplingRate>20000</samplingRate>
      <averagingSamples>100</averagingSamples>
      <minVolt>-10</minVolt>
      <maxVolt>10</maxVolt>
      <period>0.1</period>
      <a>2</a>
      <b>3</b>
</AnalogReader>
```

NiDaqAnalogWriter

This is a device that writes analog voltages to a NI-DAQ board's analog output channels. The *<NIdeviceName>* and the output *<channel>* number must be specified; *<minVolt>* and *<maxVolt>* may also be specified. Their default values are -10 V and 10 V respectively.

Example:

```
<NIwriter>
      <type>NiDaqAnalogWriter</type>
      <NIdeviceName>Dev2</NIdeviceName>
      <channel>0</channel>
      <minVolt>-10</minVolt>
      <maxVolt>10</maxVolt>
</NIwriter>
```

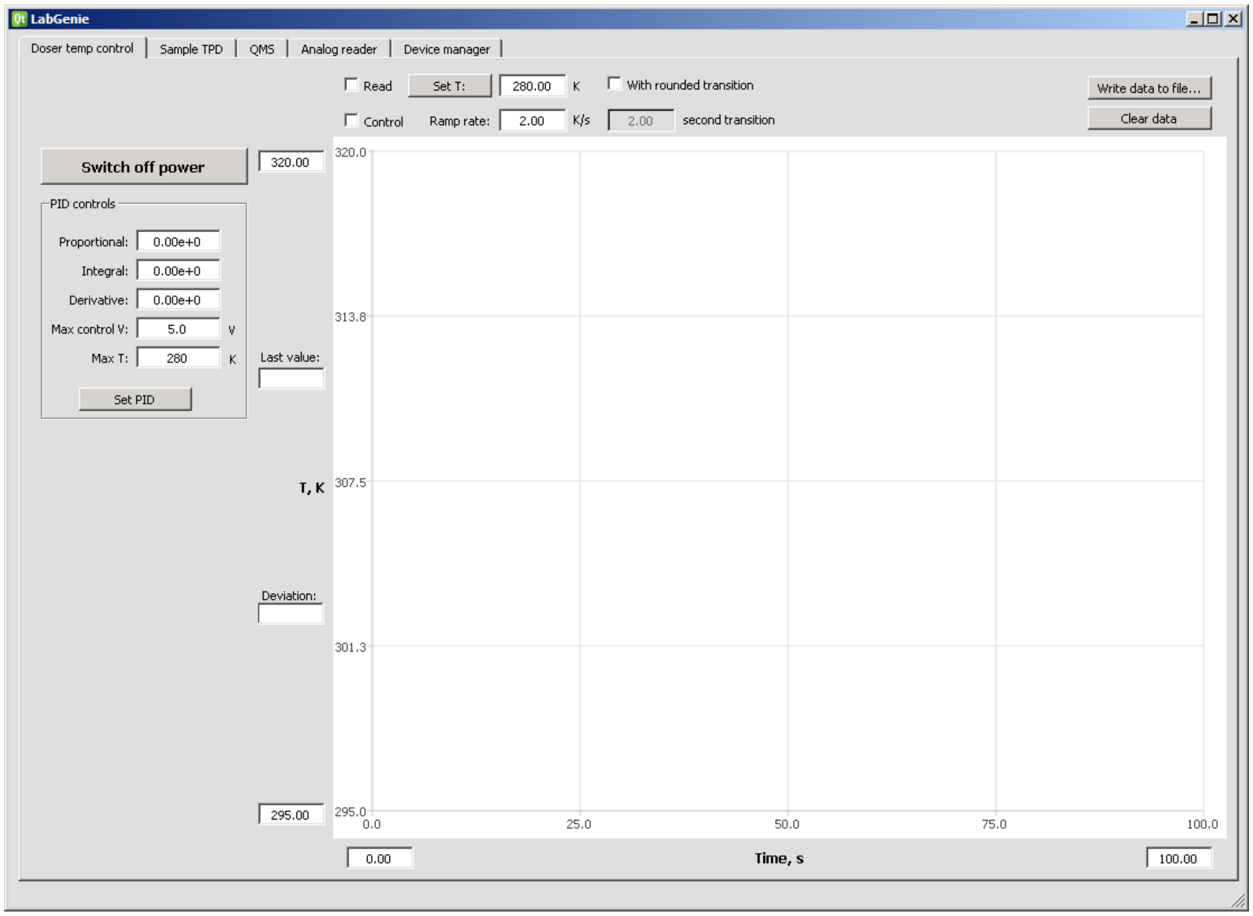
TempController

TempController is a higher-level device that creates a user interface for a PID temperature controller. To function, it requires a temperature reader (either a TempReaderNI9211 or a TempReader1048) and an analog voltage writer (again, either a NiDaqAnalogWriter or a WriterPhidgets1002). The reader and writer for the temperature controller device are specified in the *<reader>* and *<writer>* nodes respectively. An example configuration file for a temperature controller based on a Phidgets 1048 reader and a Phidgets 1002 writer is given below:

```
<config>
   <devices>
      <HwPhidgets1002>
         <type>HwPhidgets1002</type>
         <serial>493703</serial>
      </HwPhidgets1002>

      <HwPhidgets1048>
         <type>HwPhidgets1048</type>
         <serial>502963</serial>
      </HwPhidgets1048>

      <WriterSample>
         <type>WriterPhidgets1002</type>
         <hardware>HwPhidgets1002</hardware>
         <channel>0</channel>
      </WriterSample>

      <ReaderSample>
```

```
         <type>TempReader1048</type>
         <hardware>HwPhidgets1048</hardware>
         <channel>0</channel>
         <tcType>K</tcType>
      </ReaderSample>

      <TempControlSample>
         <type>TempController</type>
         <reader>ReaderSample</reader>
         <writer>WriterSample</writer>
         <show>true</show>
         <screen>Screen1</screen>
      </TempControlSample>
   </devices>

   <screens>
      <Screen1>
         <label>Sample temp control</label>
      </Screen1>
   </screens>
</config>
```

It produces the user interface shown in Figure 2:



Set T: 280.00 K

Ramp rate: 2.00 K/s

2.00 second transition

Proportional: 0.00e+0

Integral: 0.00e+0

Derivative: 0.00e+0

Max control V: 5.0 V

Max T: 280 K

320.00

295.00

0.00

100.00

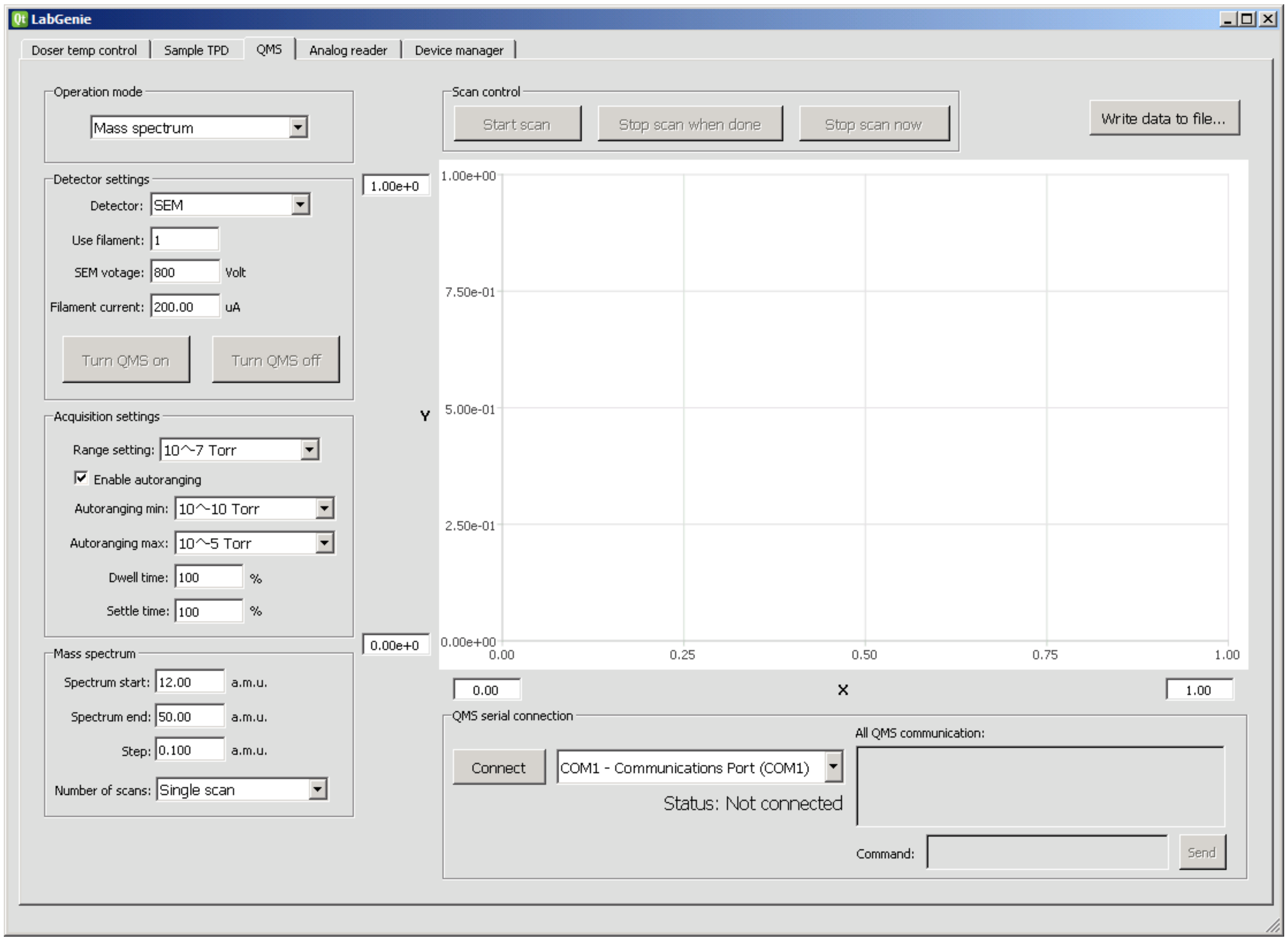
Figure 2. PID temperature control user interface.

QmsHidenHAL

As the name implies, this is a device to control data acquisition from a Hiden HAL Quadrupole Mass Spectrometer (QMS). No additional parameters are needed in device specification. Example:

```
<QmsHiden>
      <type>QmsHidenHAL</type>
      <show>true</show>
      <screen>ScreenQms</screen>
</QmsHiden>
```

The *LabGenie* user interface for the QMS is shown in Figure 3:



Operation mode: Mass spectrum

Detector settings
Detector: SEM
Use filament: 1
SEM votage: 800 Volt
Filament current: 200.00 uA

Acquisition settings
Range setting: 10^-7 Torr
☑ Enable autoranging
Autoranging min: 10^-10 Torr
Autoranging max: 10^-5 Torr
Dwell time: 100 %
Settle time: 100 %

Mass spectrum
Spectrum start: 12.00 a.m.u.
Spectrum end: 50.00 a.m.u.
Step: 0.100 a.m.u.
Number of scans: Single scan

1.00e+0
0.00e+0
0.00
1.00

QMS serial connection
COM1 - Communications Port (COM1)
Status: Not connected
All QMS communication:
Command:

Figure 3. Hiden HAL QMS user interface.

Tpd

Finally, *Tpd* is a composite device that combines a temperature controller and a QMS to enable the acquisition of Temperature Programmed Desorption (TPD) data. The QMS device (*<qms>*) and the temperature control device (*<tempControl>*) need to be specified as child nodes:

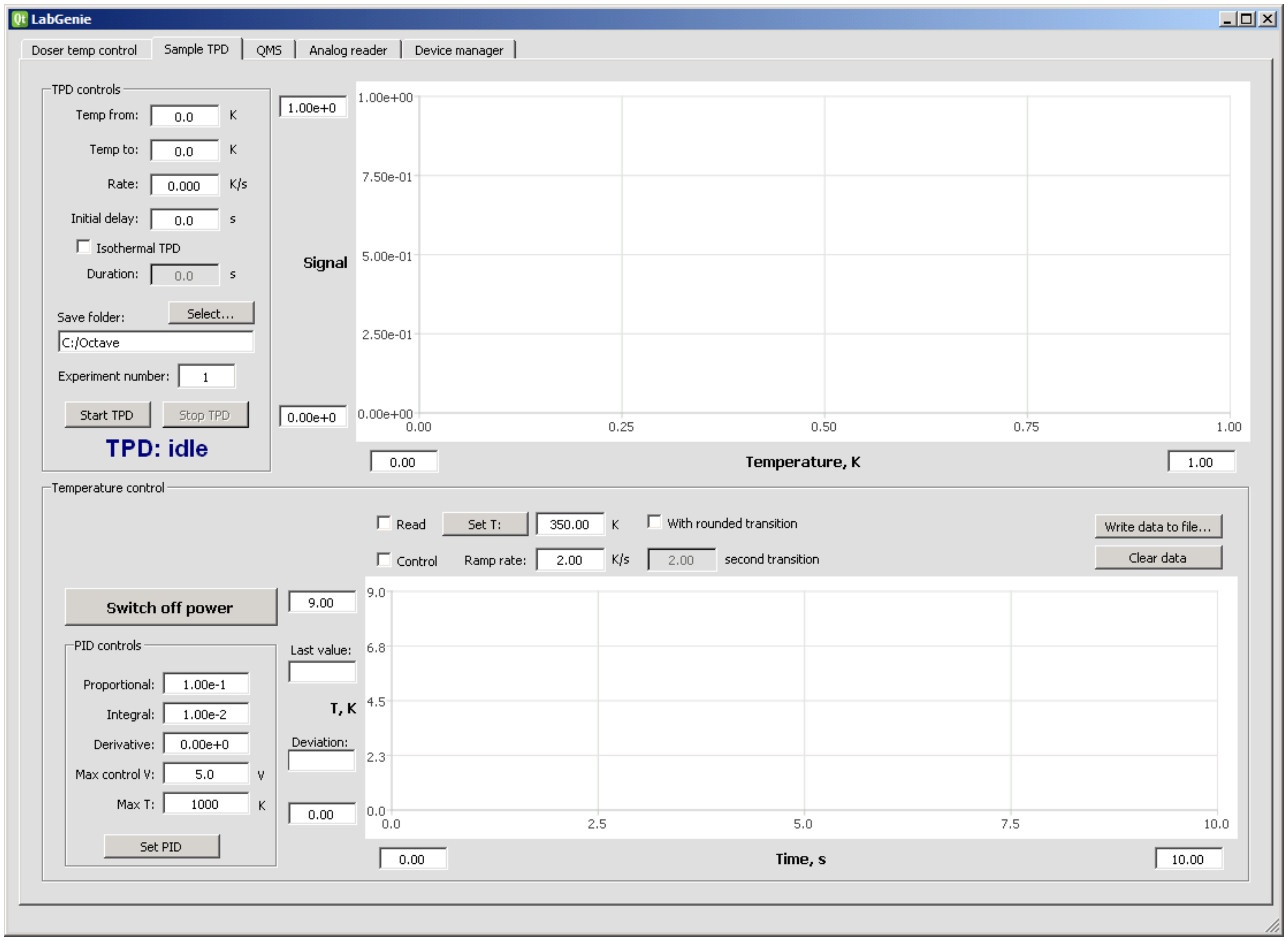
```
<MyTpd>
   <type>Tpd</type>
   <tempControl>TempControlSample</tempControl>
   <qms>QmsHiden</qms>
</Tpd>
```

The configuration below defines a full stack of devices necessary for a TPD system:

- NI-9211 temperature reader hardware
- A reader for thermocouple channel 0
- A Phidgets 1002 voltage output hardware
- A voltage writer for channel 0
- A temperature controller that incorporates the reader and the writer
- A Hiden QMS device
- A TPD device that incorporates the temp controller and the QMS

```
<config>
   <devices>
      <Hw9211>
         <type>HwNI9211</type>
         <NIdeviceName>Dev1</NIdeviceName>
      </Hw9211>

      <SampleTempReader>
         <type>TempReaderNI9211</type>
         <hardware>Hw9211</hardware>
         <channel>0</channel>
         <tcType>K</tcType>
      </SampleTempReader>

      <HwPhidgets1002>
         <type>HwPhidgets1002</type>
         <serial>493756</serial>
      </HwPhidgets1002>

      <WriterSample>
         <type>WriterPhidgets1002</type>
         <hardware>HwPhidgets1002</hardware>
         <channel>0</channel>
      </WriterSample>

      <TempControlSample>
         <type>TempController</type>
         <reader>SampleTempReader</reader>
         <writer>WriterSample</writer>
      </TempControlSample>

      <QmsHiden>
         <type>QmsHidenHAL</type>
         <show>true</show>
         <screen>ScreenQms</screen>
      </QmsHiden>

      <Tpd>
         <type>Tpd</type>
         <tempControl>TempControlSample</tempControl>
         <qms>QmsHiden</qms>
         <show>true</show>
         <screen>ScreenTpd</screen>
      </Tpd>
   </devices>

   <screens>

      <ScreenTpd>
         <label>Sample TPD</label>
      </ScreenTpd>

      <ScreenQms>
         <label>QMS</label>
      </ScreenQms>

   </screens>
</config>
```

This configuration shows a TPD interface in the software:



TPD controls
Temp from: 0.0 K
Temp to: 0.0 K
Rate: 0.000 K/s
Initial delay: 0.0 s
☐ Isothermal TPD
Duration: 0.0 s
Save folder: C:/Octave
Experiment number: 1
TPD: idle

Temperature control
☐ Read
☐ Control
Set T: 350.00 K
Ramp rate: 2.00 K/s
☐ With rounded transition
2.00 second transition

PID controls
Proportional: 1.00e-1
Integral: 1.00e-2
Derivative: 0.00e+0
Max control V: 5.0 V
Max T: 1000 K

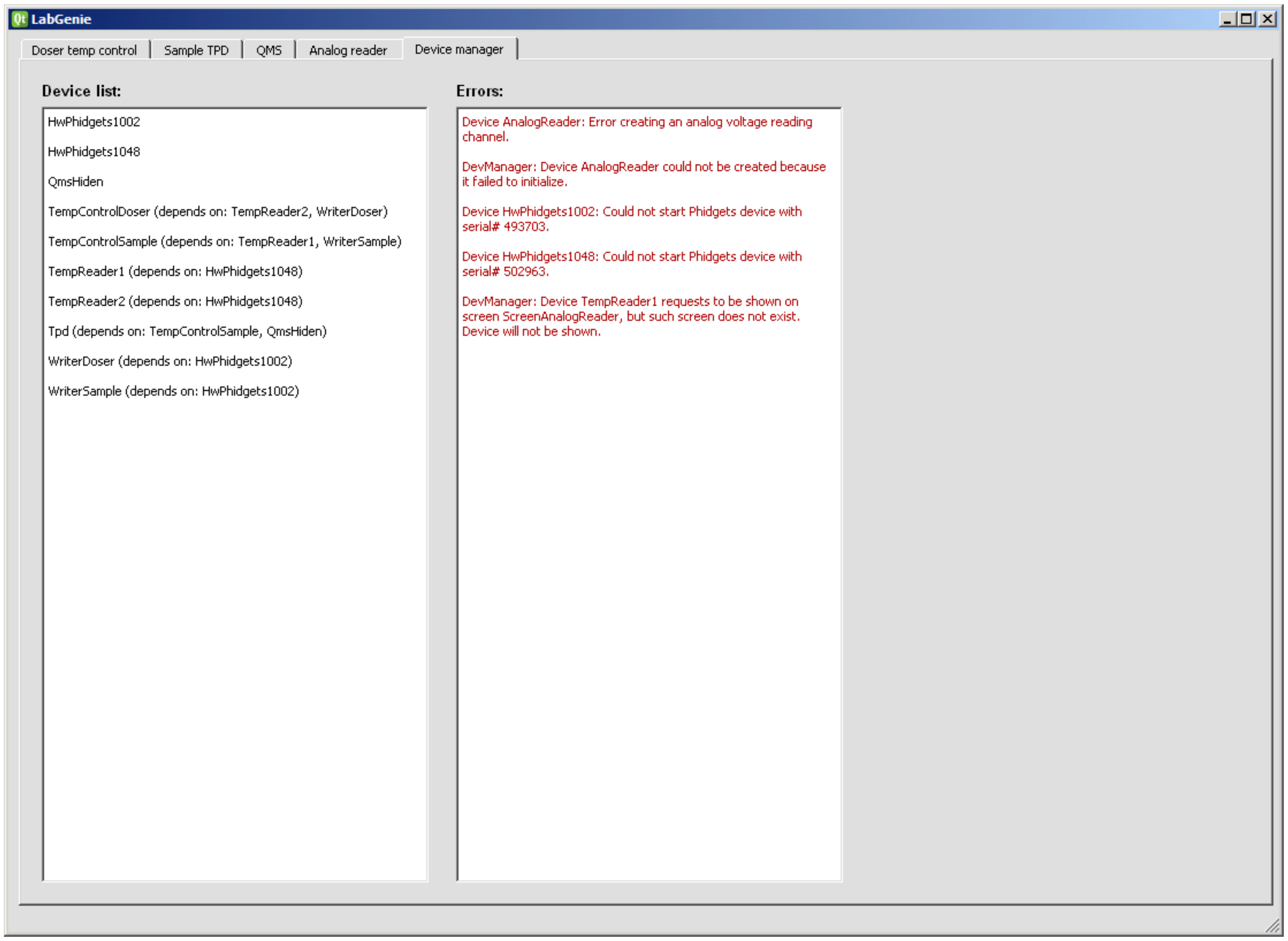
Last value:
Deviation:

Figure 4. LabGenie TPD interface.

Note that even though we did not specify to <*show*> the temperature controller separately, it is shown as a part of a TPD interface.

Device Manager tab

In addition to the screens (tabs) requested in the configuration file, the software always shows an additional tab named "Device Manager":



Device list:

```
HwPhidgets1002
HwPhidgets1048
QmsHiden
TempControlDoser (depends on: TempReader2, WriterDoser)
TempControlSample (depends on: TempReader1, WriterSample)
TempReader1 (depends on: HwPhidgets1048)
TempReader2 (depends on: HwPhidgets1048)
Tpd (depends on: TempControlSample, QmsHiden)
WriterDoser (depends on: HwPhidgets1002)
WriterSample (depends on: HwPhidgets1002)
```

Errors:

```
Device AnalogReader: Error creating an analog voltage reading
channel.

DevManager: Device AnalogReader could not be created because
it failed to initialize.

Device HwPhidgets1002: Could not start Phidgets device with
serial# 493703.

Device HwPhidgets1048: Could not start Phidgets device with
serial# 502963.

DevManager: Device TempReader1 requests to be shown on
screen ScreenAnalogReader, but such screen does not exist.
Device will not be shown.
```

Figure 5. The LabGenie Device Manager tab with some configuration errors shown.

The tab has two fields, the Device List and the Error List. The Device List shows all the devices requested in the configuration file that were successfully created during software initialization. The Error List shows initialization errors; the presence of errors indicates that some devices could not be created as specified. A correctly written configuration file will produce no errors.

The specific error messages may be helpful in figuring out what exactly is misconfigured.