LabVIEW Exercises: Part 3

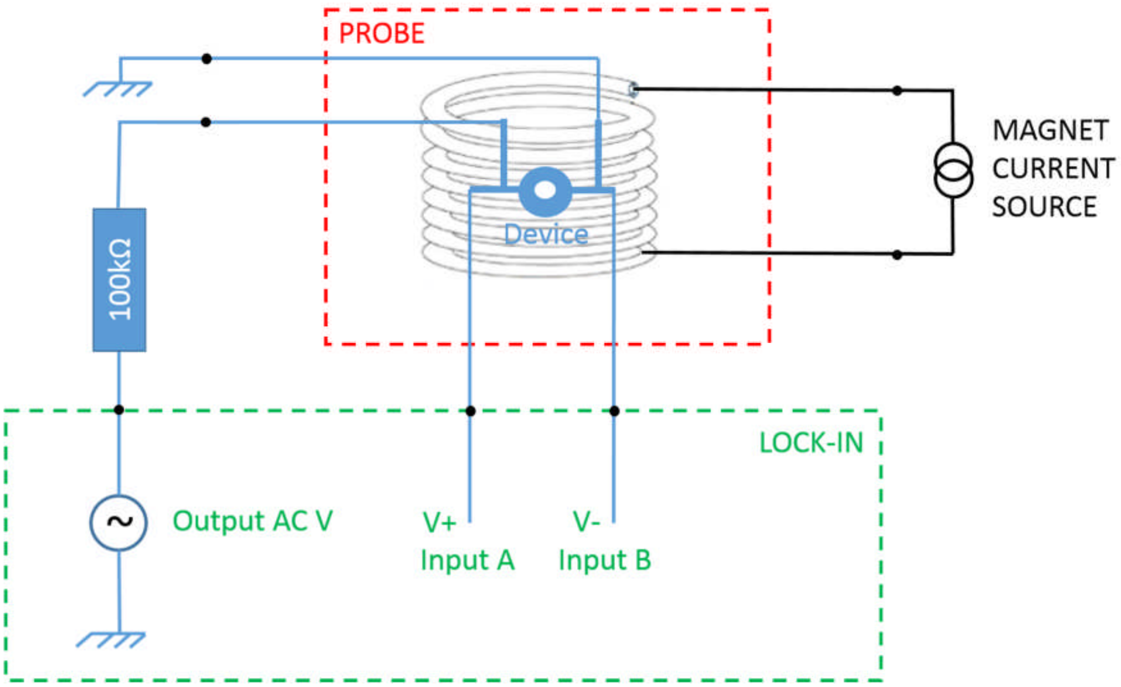
In this task, you will create the VI your group will use in the lab exercise to collect data at low temperature from the device you will have fabricated in the cleanroom. You should develop the VI as a group – it might be most efficient if you decide as a group to do certain parts of the coding separately and combine them later, although you will certainly need to debug the VI together. (For example, you could divide up the tasks of designing how the program should run and the task of finding out which commands need to be sent to the instruments.)
In order to avoid confusion with the VIs of other groups, save the VI you create with exactly the following filename, InstrumentControlExercises_YourNames and save any subVIs or controls you make including the suffix _YourNames – for example: Control1_YourNames.ctl.

Instrument control VI task

For the lab experiment you will need to carry out low-temperature resistance measurements as a function of magnetic field for the device which you will have made in the cleanroom, and also to find its superconducting critical temperature, by measuring the resistance as a function of temperature. This exercise is fairly realistic compared with a real lab situation, where you might well have some new instruments, know how you want to use them in an experiment and need to develop LabVIEW code for achieving this.

To measure the device resistance, you will use a lock-in amplifier to improve noise rejection. The lock-in amplifier can be thought of as a sort of ac voltmeter. It sources a low-frequency ac signal (tens of hertz) and then looks for components at that frequency in the measured voltage, thereby rejecting noise at other frequencies. (It also detects the phase of the signal with respect to the source signal – or a reference signal.)

A schematic of the set-up is shown below.



You will connect the ac output voltage V_ac from the lock-in to a resistor R_series which you will put in series with the sample. Since the total sample resistance is expected to be tens of ohms (you will verify this during the experiment), the series resistor means that the current through the sample is given to a good approximation by I=V_ac/R_series, independent of the sample resistance, so since you set V_ac you will know I.

The magnetic field at the device has already been calibrated with respect to the current in the coil. The magnetic field B is related to the current I_coil in the coil by B=0.072*I_coil - 0.0004 where B is in Tesla and I_coil is in A.

The instruments you will use are an SR830 lock-in amplifier to measure the device resistance and a Keithley 2400 current/voltage source to provide the current to the magnetic-field coil. These instruments are both equipped with GPIB interfaces and so can be interfaced to a PC by a GPIB interface and therefore controlled by LabVIEW.

Temperature control for the system is already taken care of by another VI, which will be running in the background. This VI reads the temperature from the probe's sensors and logs this to a file. The same VI is also be used to control the temperature. It periodically updates a global variable for the system temperature. The VI which you write therefore does not need to talk to the system's temperature controller, only to read the value of the relevant global variable. A copy of the relevant global variables VI is included in the Moodle folder. Do not modify that VI! You should save your VIs so that they are compatible with LabVIEW 2013 (the version of the software installed on the machines in the CDT cluster is LabVIEW 2014). When you bring your VIs to test in the lab, you should bring them on a USB key – you should scan it for viruses before bringing it to the lab.

You will need to do the following:
-Record the variation of the sample resistance with temperature.
You will use this to determine the superconducting Tc of your sample.
-Record the variation of the sample resistance with magnetic field.

You could do this using a single program (the most elegant solution!), or by using two similar programs.

Your program should calculate the sample resistance from the sourced ac voltage and the voltage measured by the lock-in and calculate the magnetic field from the generated magnet current. The values used for this calculation should be included in a header to each saved data file along with any other relevant settings.

By looking at the documentation for the instruments (find their manuals online) and examining the instrument manufacturers' sample VIs, work out how to control the instruments. I recommend starting by locating online and downloading the drivers and example VIs for these instruments. Open some of the example VIs so that you can see how to control the instruments. NB. These VIs may be more or less useful for what you want to do. You do not need to use the example VIs if they do not do what you want them to do – the alternative is to use lower level VIs and the fundamental GPIB text-control commands which you will find in the instrument manuals.

The VI(s) which you write should:
-display the present temperature, magnetic field and ac current prominently on the front panel (use a font size larger than the default!) Obtain the sample temperature by reading the current value of the relevant global variable.

-initialises the instruments when first run. This should include putting the Keithley 2400 into current-source mode (in case it was in voltage-source mode) and setting the correct phase mode and gain mode for the SR830 (start with auto-phase and auto-gain mode).
-Each lock-in measurement should carry out a user-specified amount of averaging (determined by the integration time).
-For both sorts of measurement, the data should be saved point by point as they are collected into a datafile (for the R(B) sweeps, record the temperature for each point as well) and also plotted onto an R(T) or R(B) graph (as appropriate) on the front panel.

All configuration and control of the instruments should be done from the LabVIEW front panel – the user should not need to touch the physical front panel of the instruments once the power is on!

For the magnet-current sweep measurement:
-The sweep should begin when the user clicks a “Start R(B) Measurement” button
-The sweep of the magnet current should start from a user-specified value (of field – but the current should also be displayed) to a second user-specified value with a user-specified number of intermediate points and then back down to the first specified value (which will usually be zero). It would be sensible to automatically repeat field sweeps at both positive and negative fields to check there is no difference between the properties.
-It should not be possible to specify a magnet current magnitude above 1A. The software should check that the start and end of the range are within these limits and the sweep should be limited to this range.
-At each value of the current, the lock-in measurement should begin after a user-specified delay (to allow time for any transients to settle after the current is changed.)

For the resistance-temperature measurement:
-The measurement should begin when the user clicks a “Start R(T) Measurement” button
-The program should then read the lock-in voltage and the temperature at sensible intervals. Since the temperature will be changing during the measurements it is important to make these readings close together. Remember that the value of the global variable is only the up-to-date value of the temperature when it is first read by the temperature-logger VI. Therefore, to avoid a time lag between the R and T readings which introduces a systematic error, a good practice would be:
-Wait for the temperature reading to be updated (in practice since the temperature will be changing, this can be achieved simply by waiting for the value of the global variable to change) read the global temperature variable, then read the voltage and then read the global temperature variable again (in case it has changed in the meantime) and then either average the two temperature readings or record both of them. (You could also add a time-out condition and/or user-skip button so that the VI only waits a certain time for the global variable to change/until the user clicks before going ahead with the measurement.)

-If a “Stop” button is pressed, any open files should be closed and the instruments de-initialised before the program exits.
-Your program could contain a button which will save an image of a relevant graph when it is clicked. This can be useful for instant analysis (since you do not need to open the relevant datafile and plot it again). You can save an image of a graph using an Invoke Node, which you create by right-clicking on the graph in the block diagram and choosing Create > Invoke Node > Export Image. You will need to to wire constants into the File Type and Target inputs for the Invoke Node (right click and choose Create > Constant and keep the respective default values of .BMP and File) and also wire a filepath

into the Path input – this could be from a control or from a user dialogue or could be a modified version of the filename into which the data are being saved. Check what the saved image looks like (unfortunately it's not exactly as on screen!) It is possible to programmatically (via a Property Node) change properties of the graph if you want to improve the look of the saved image.