**QC LabVIEW Documentation**

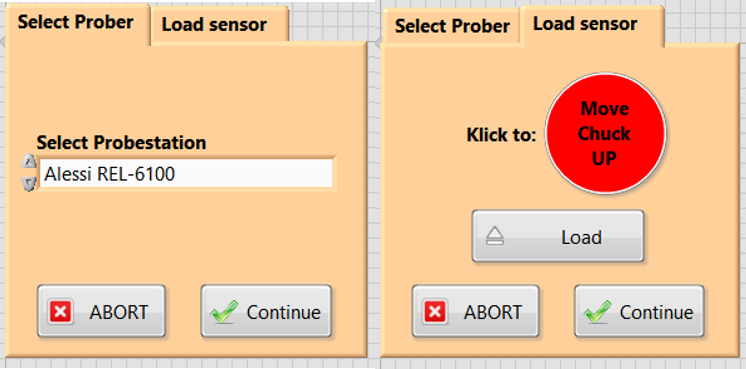
# Getting started

# IV measurements

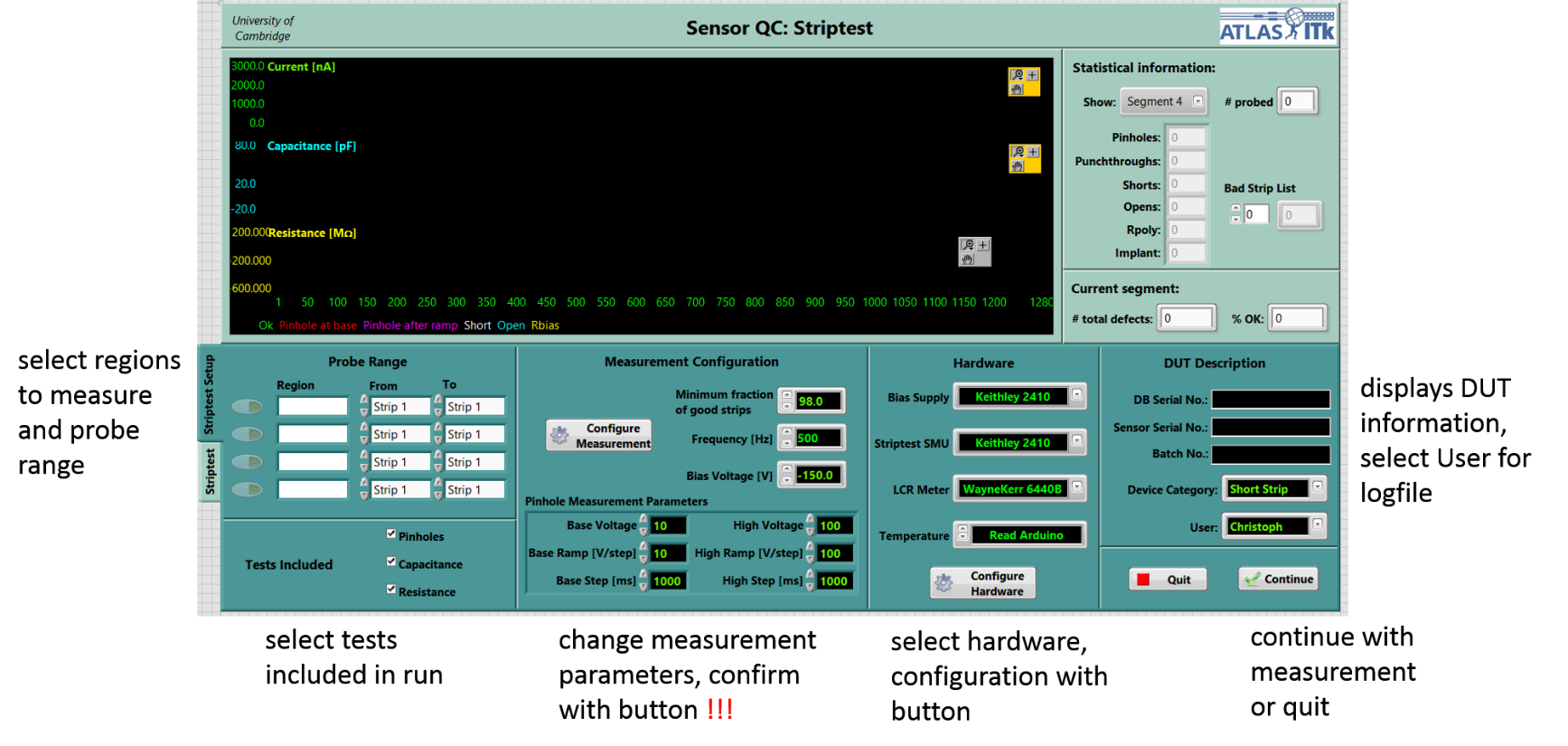
# CV & Cis measurements

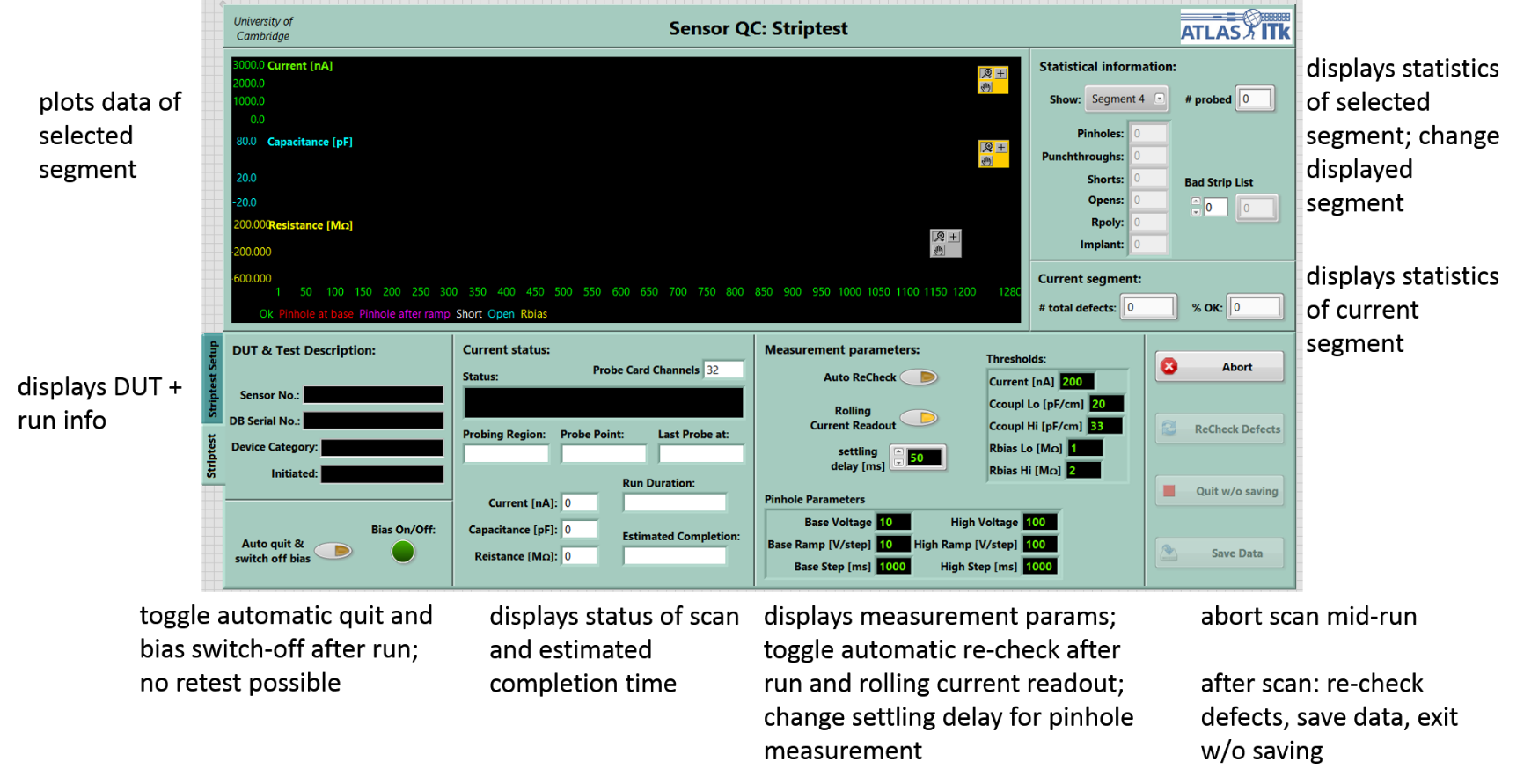
# Ris measurements

# Striptests

Upon running the striptest VI, the probe station selection automatically opens, which also includes the selection of some parameters e.g. for the edgesense. Currently there are 2 probe stations implemented, the Alessi REL-6100 and the SemiProbe. The initialisation VIs are shown below. After the probe station is selected and initialised, the open VI is meant to facilitate loading of sensors by being able to move the chuck to a pre-defined load position and back to the centre, as well as allowing the chuck to be moved between the *Separate* and *Contact* position.

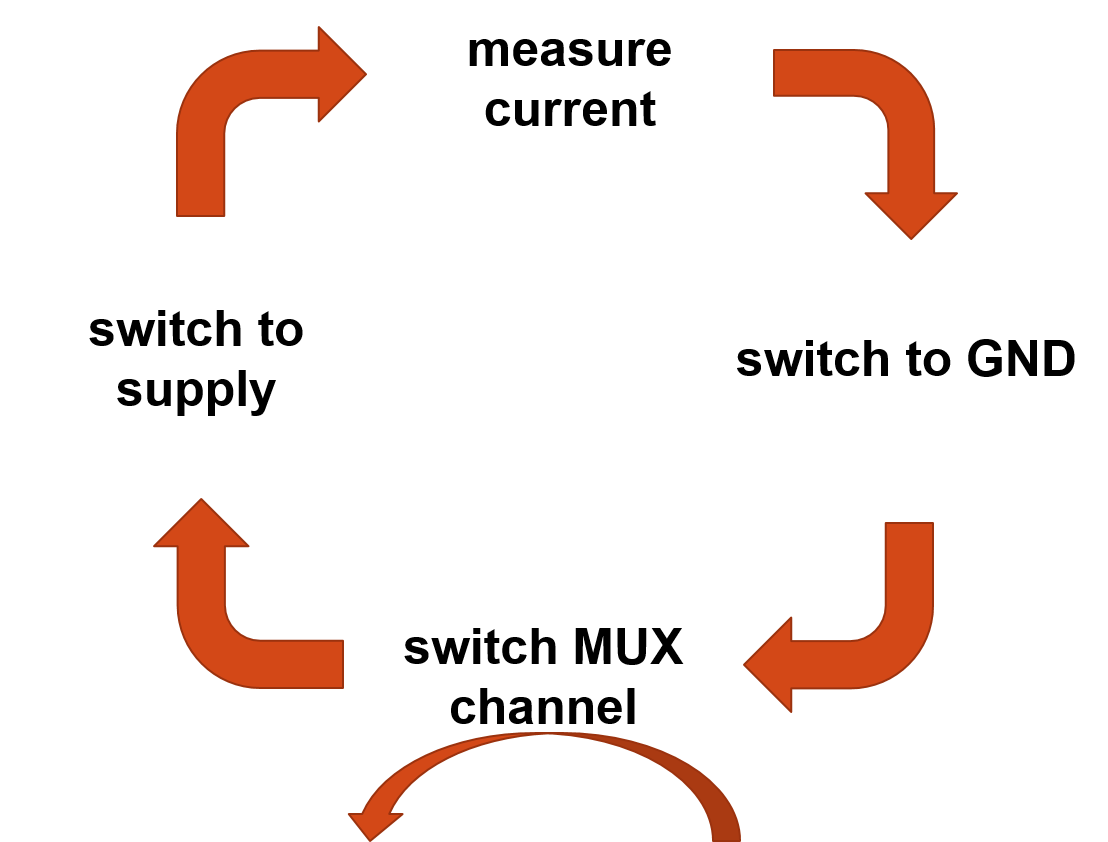
The next step includes the loading of the probeplan, either the alignment of chuck in *theta* or the rotation of the probeplan coordinates to fit the chuck rotation, and the option to test the loaded coordinates and alignments by jumping to the AC measurement pads.

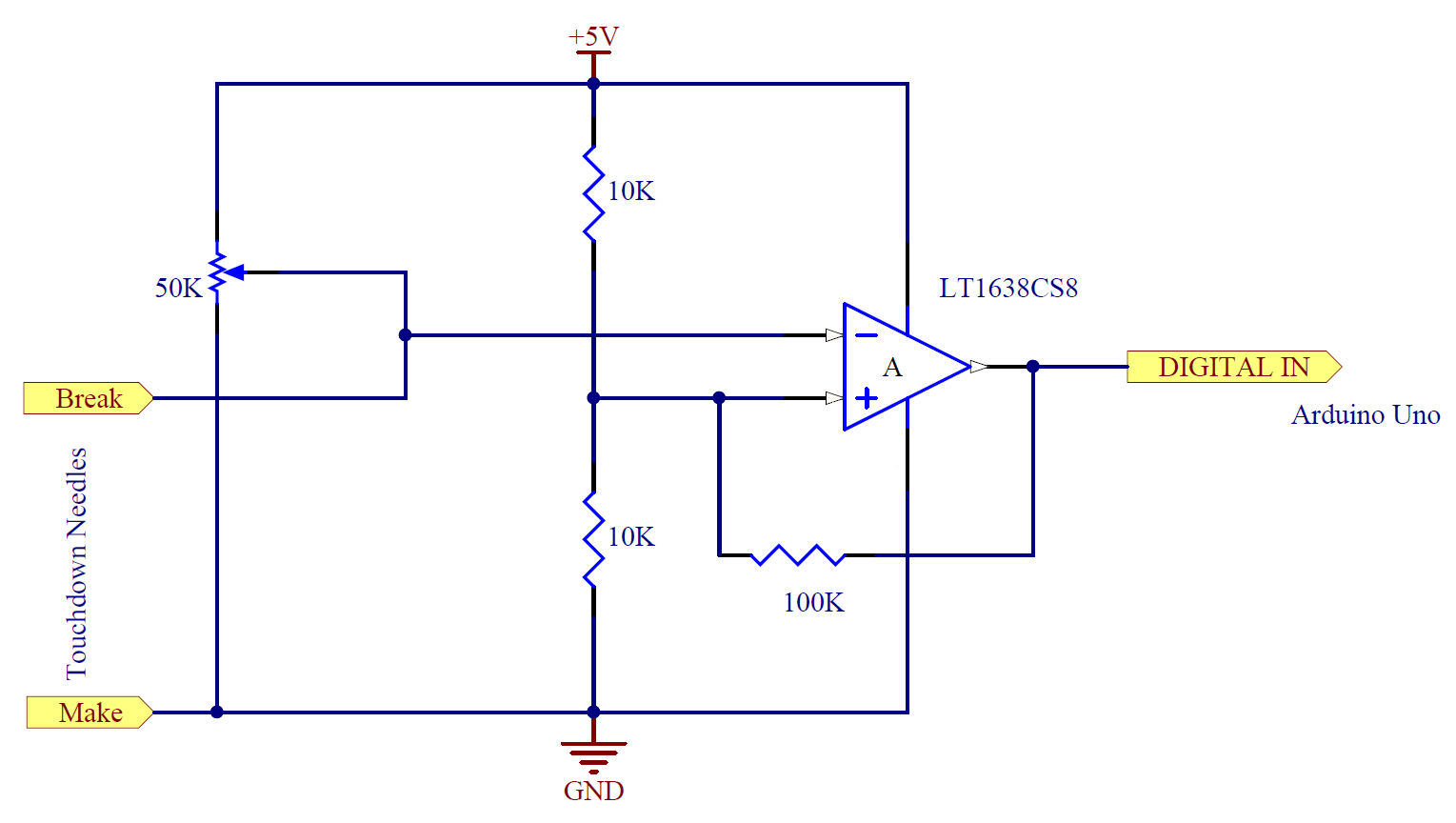




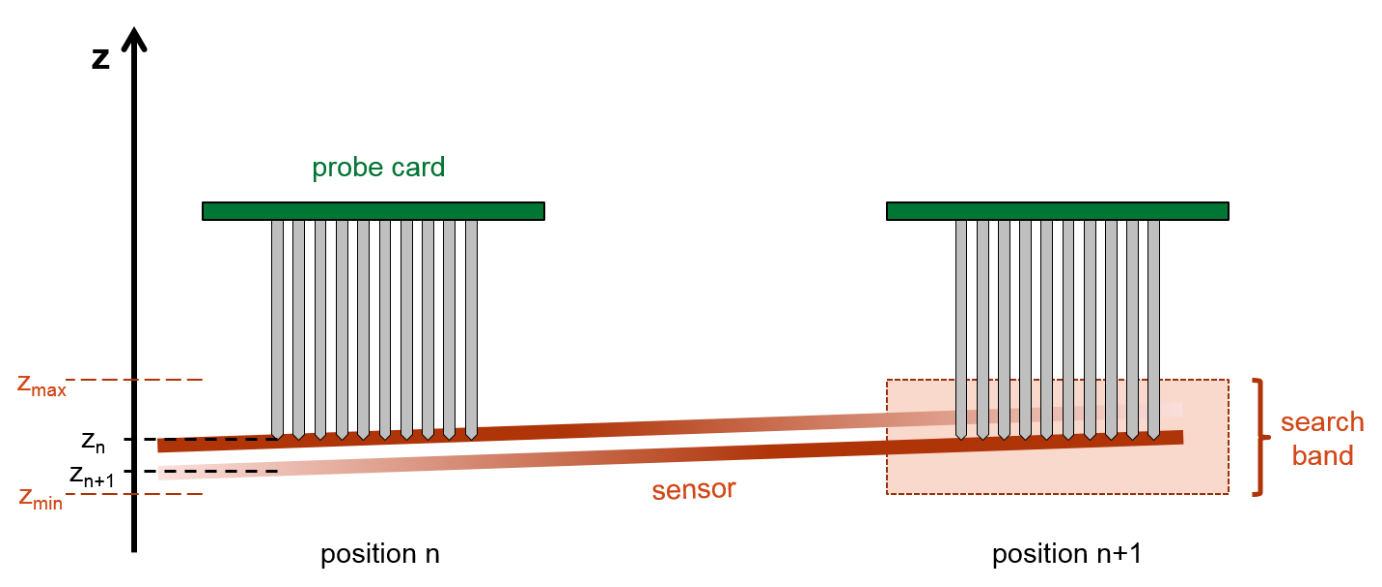
## Special features using a probecard (barrel sensors)

The main reason to use a probe card is the opportunity to measure multiple strips without having to move the chuck to a new position for each new strip. Hence the strip test procedure was edited in order to minimise the time needed for the full strip test. Since hot-switching at 100V can yield difficulties due to the RC behaviour of the cables and the probe card itself and resulting in current spikes originating from discharge processes while switching the card channels, these effects have to be accounted for with a reasonable settling time.

Due to the two 8x2 multiplexers leading to 4 simultaneously usable probe card channels, a specific readout procedure has been devised in order to minimise the time necessary for the pinhole measurement. According to the scheme displayed there are 4 steps in the current measurement used to determine pinholes. Having one MUX output present in each step at any given time during the measurement allows for the necessary delay to be reduced, since the switching processes and the SMU readout itself introduce a ``natural'' delay, during which the current of channels, which are switched to high voltage/GND, are allowed to settle.

The probe card also provides two touchdown needles in addition to the 32 signal channels used for measurement. These needles can be used in a custom-made touchdown feedback using *make* or *break* contact configuration, i.e. one needle touches down on the sensor surface in alignment with the measurement needles, whereas the second one loses the contact to the first needle upon touching the surface. The edgesense circuit uses the break contact configuration in conjunction with a *Schmitt trigger* on a digital input pin of the Arduino to provide the necessary touchdown information. The reference voltage is set to 2.5V with the Arduino providing the asymmetric supply +5V/GND for the amplifier. The make needle is pulled to GND potential resulting in the amplifier sourcing +5V without surface contact and 0V upon touchdown due to the break needle being at floating potential.

For the edgesense to work, the *initial z* coordinate -- i.e. the chuck height at the previous position -- has to be known, as well as the *search band*, defining the limits around the initial z within which the new z value is searched, the *step size*, and the *overdrive*. The general concept of the edgesense is as follows:

1. when moving to a new position, the chuck is automatically lowered to the *Separation* state
2. upon arriving at the target x and y coordinates, the chuck is risen to the lower limit of the search band, i.e. the initial z minus half the search band
3. the output of the Arduino is checked for contact
4. if no contact is established, the chuck is moved up by the step size value
5. the last two steps are repeated until either a contact has been established or the upper limit of the search band is reached
6. the chuck is moved up by the overdrive value in order to ensure a good contact for all needles
7. in the case of no contact until the previous step, the Arduino is checked one last time and if the result is still negative, an error message is shown; if there is a contact, the z coordinate minus the overdrive is saved for moving to the next position

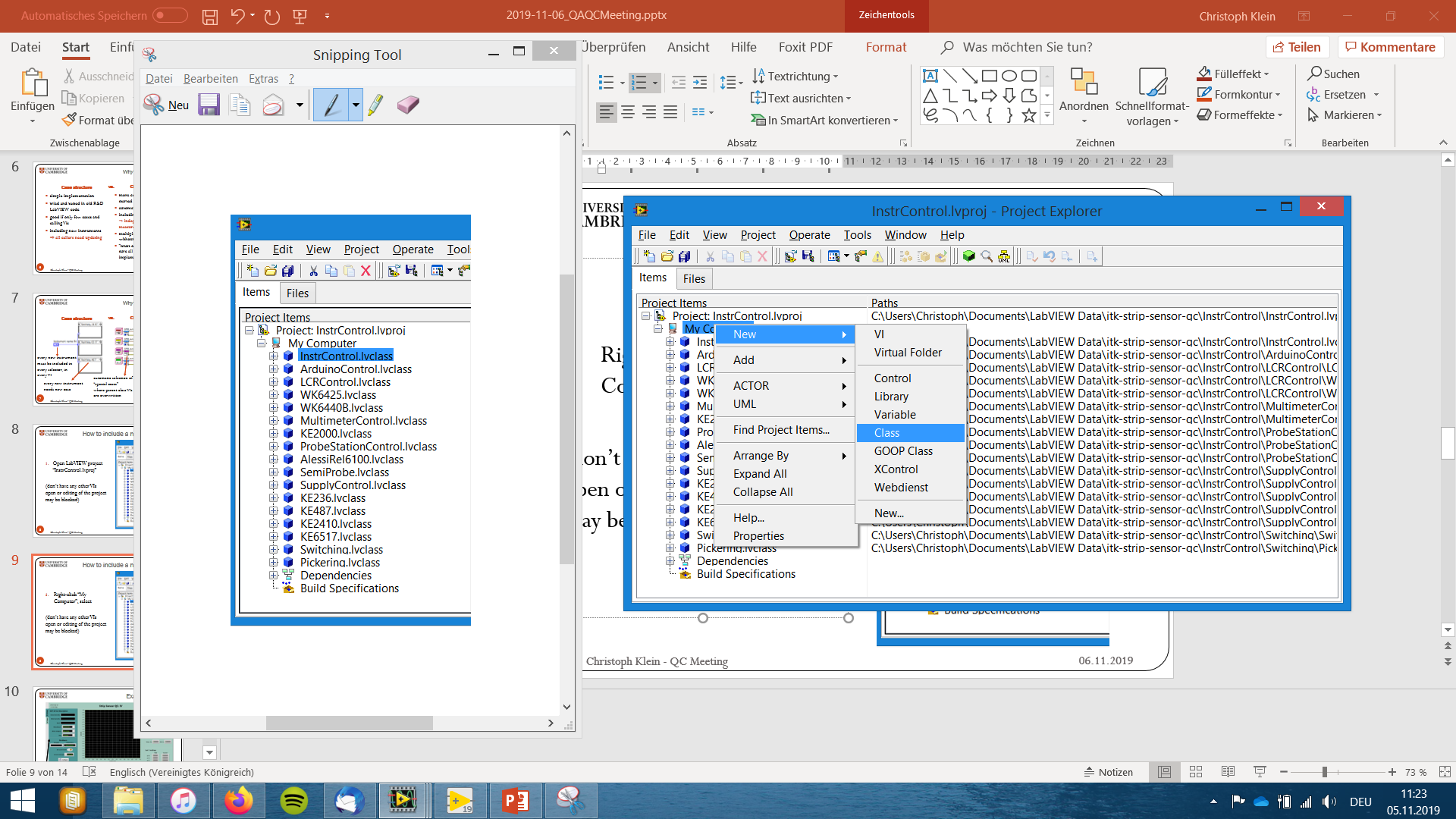
# How to include a new instrument

1. Open LabVIEW project “InstrControl.lvproj”.

Don’t have any other VIs open or editing of the project may be blocked.

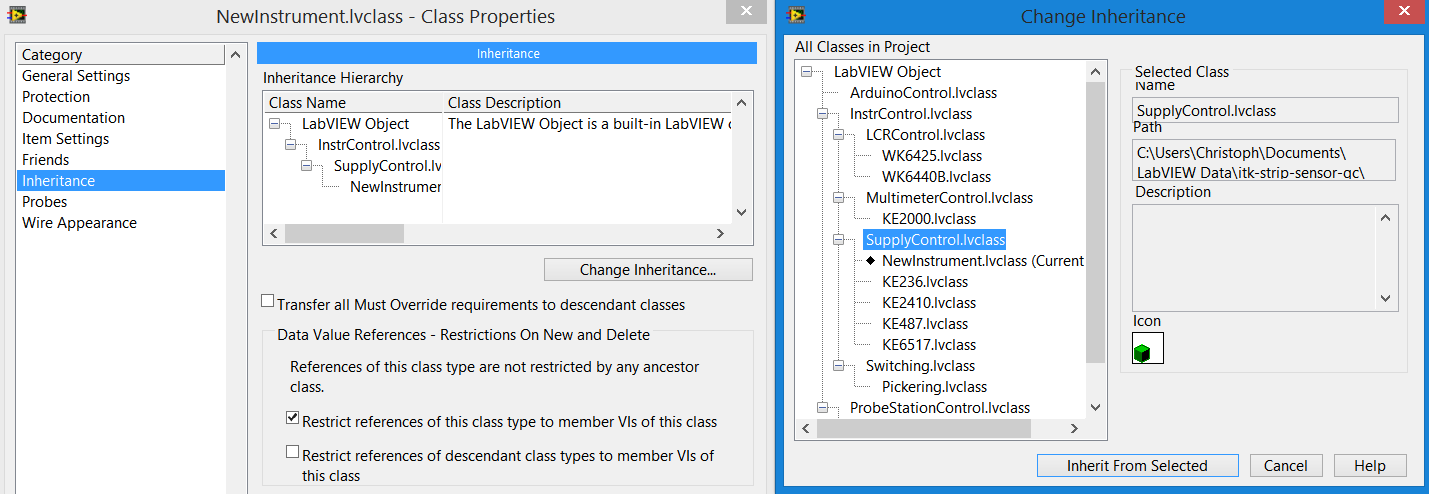
The project lists all instrument classes present.

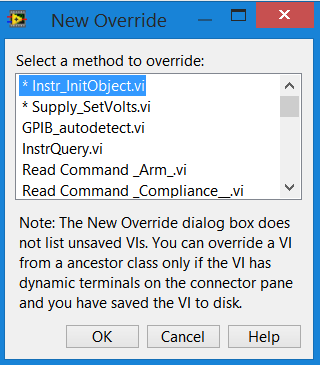
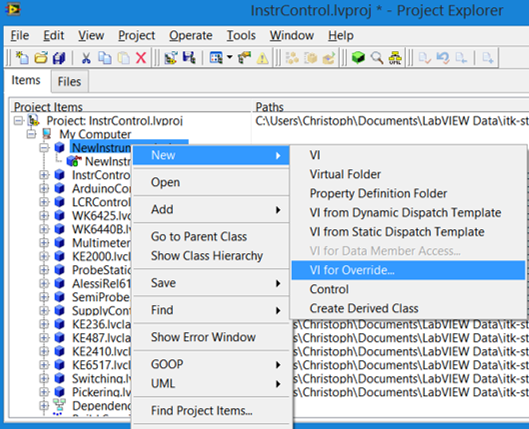
1. Right-click “My Computer”, select “New → Class”.

Options upon right-clicking may look slightly different depending on the LV version and installed add-ons.

1. Give your class a name.

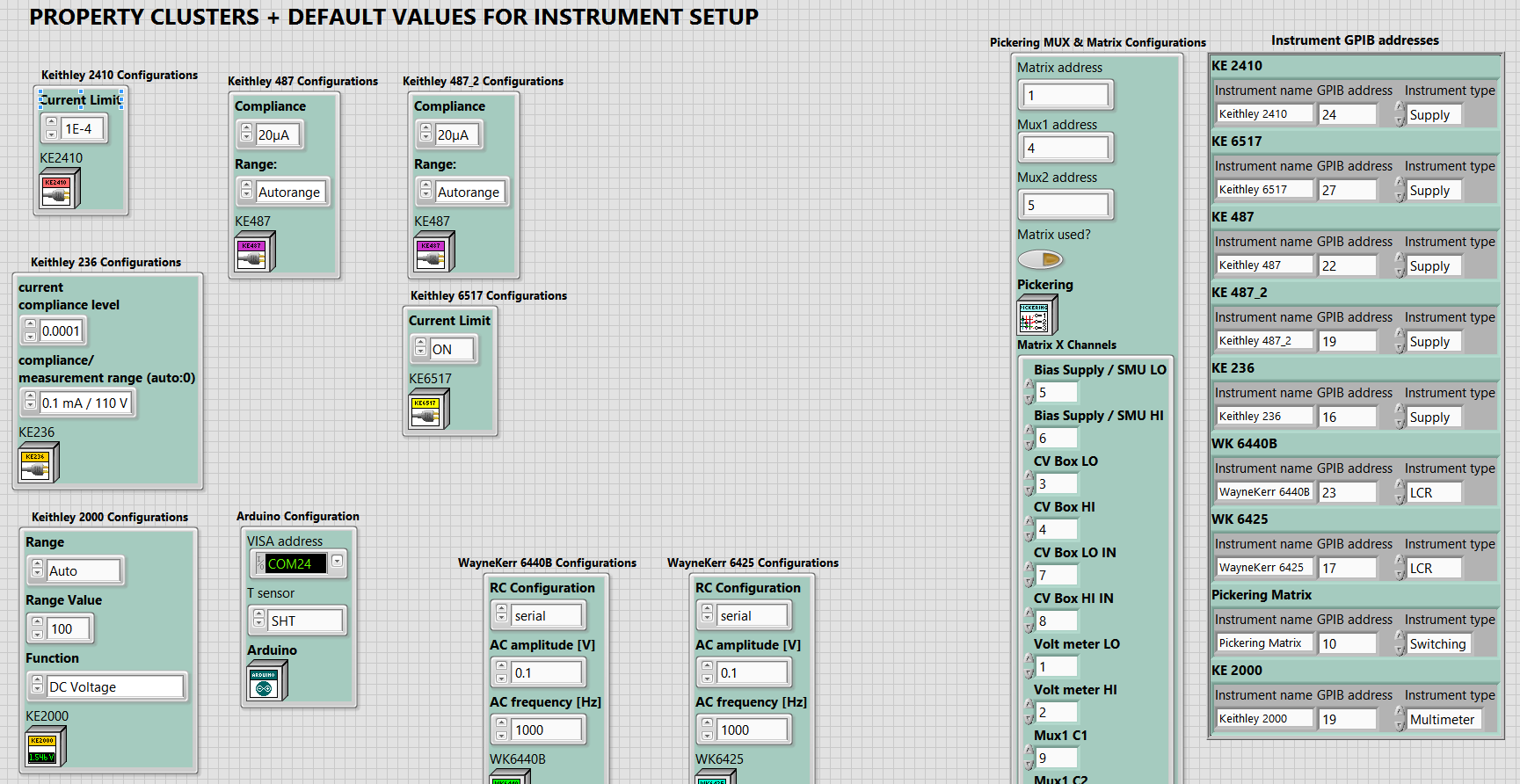
Choose one which allows easy identification of the instrument it is being used for.

1. Right-click new class, select “Properties → Inheritance” and change inheritance accordingly.
2. Right-click new class, select “New → VI for Override”.



VIs with \* must be overwritten; all others can be, if necessary.

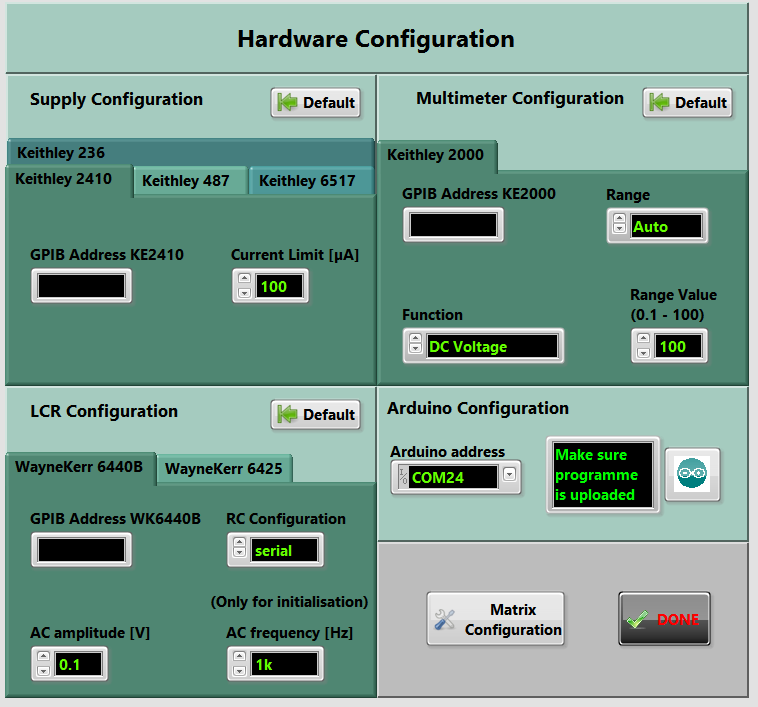
Usually, only VIs which require procedures specific for each instrument or need to be unique – and not just differ in GPIB commands used – are denoted as “override required”.

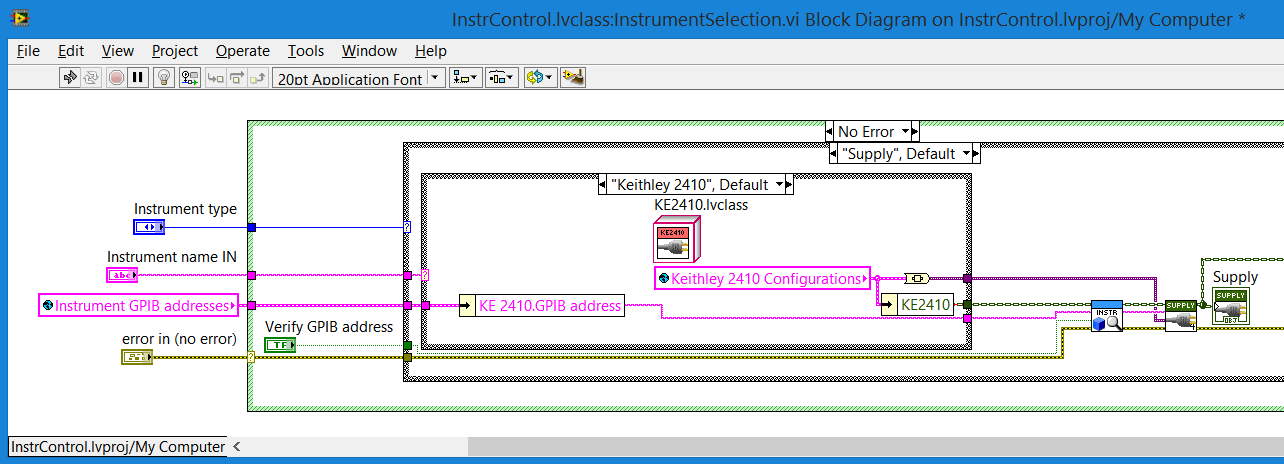
1. “InstrumentSetup” global: add default properties cluster.

Add instrument (name, address, type) in GPIB address cluster. If new instrument is not a GPIB instrument, use NI-VISA address, but still give name + type like other GPIB instrument.

1. Add instrument with relevant parameters in “HardwareConfiguration.vi”.

Not all parameters which *can* be configured are also *relevant*. Select only those necessary to be configured by the user, all others can be hardcoded in the instrument initialisation VI.



1. Add instrument in “InstrumentSelection.vi” with correct type + name.

The Instrument selection VI is important to create an object with the correct address from the instrument classes and store that newly created object in the corresponding measurement configuration global.

E.g. during IV scans the selected supply from the specific instrument daughter classes will be saved as the more generic “Supply” class object from its parent class in the configuration cluster to be referenced during the measurement script.