

LabVIEW scripts for sensor QC

Christoph Klein
QA-QC Meeting
06/11/2019



Goals

- preparation for sensor pre-production
- same tests at every QC site \Rightarrow reproducibility of results
- same format of data files
- compatibility with database upload
- adaptable to different local setups (e.g. varying GPIB addresses, data directories, etc.)
- easy to include additional/new instruments
- general user-friendliness



Prerequisites

Instrument hardware:

- SMUs
 - LCR meter(s)
 - Arduino with T/RH readout (+ touchdown feedback)
 - switching matrix/multiplexers
 - probestation
- } for strip testing

Software:

- NI LabVIEW 2013 or newer

Instrument Control



Why class structure?

Case structure

- simple implementation
- tried and tested in old R&D LabVIEW code
- good if only few cases and calling Vis
- including new instruments
⇒ all callers need updating

vs.

Class structure

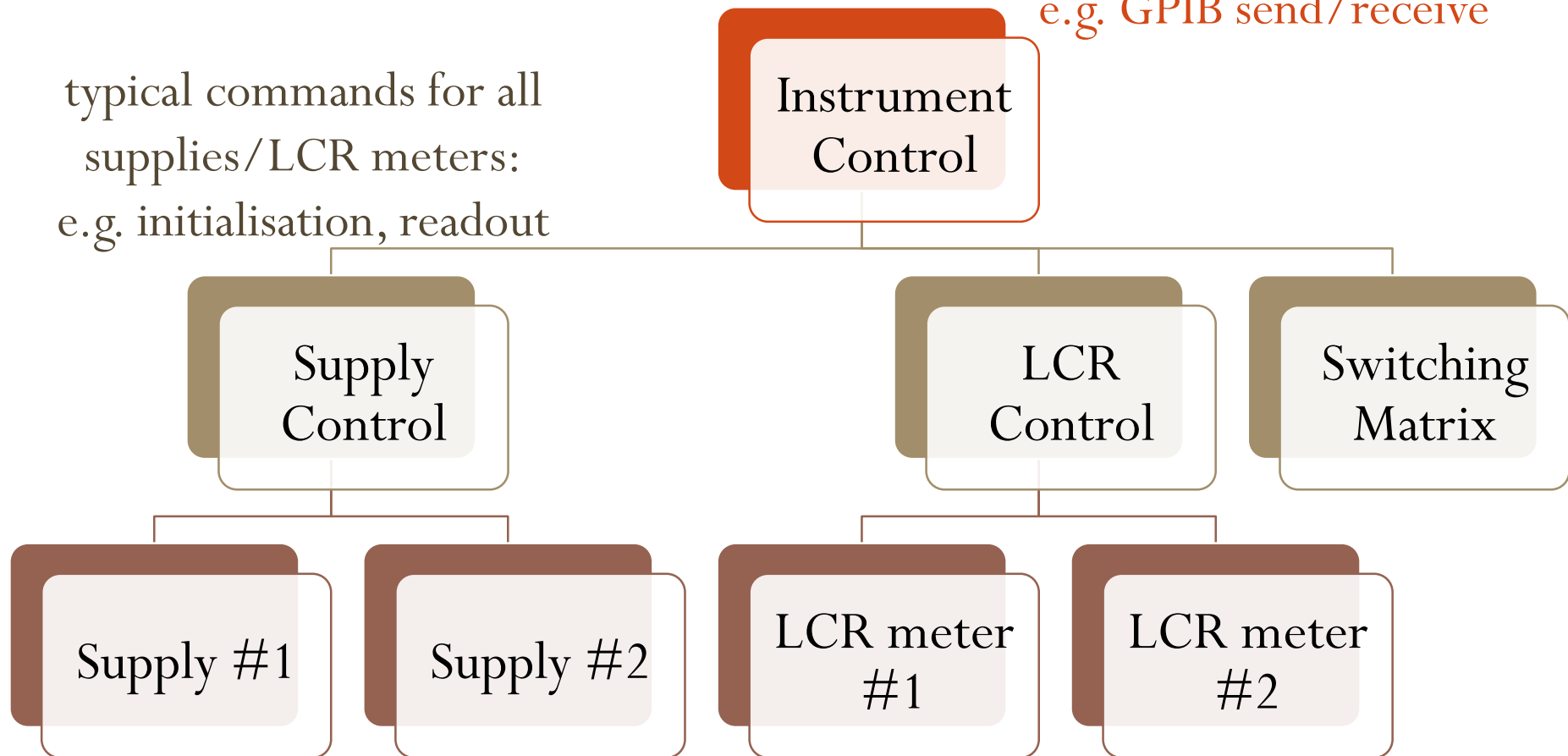
- more complicated to get started
- automatic polymorphism
- including new instruments
⇒ independent from measurement scripts
- multiples of same instrument without additional class
- “must override” option makes sure all descendants implement their own versions



Instrument control: class hierarchy

low-level commands:
e.g. GPIB send/receive

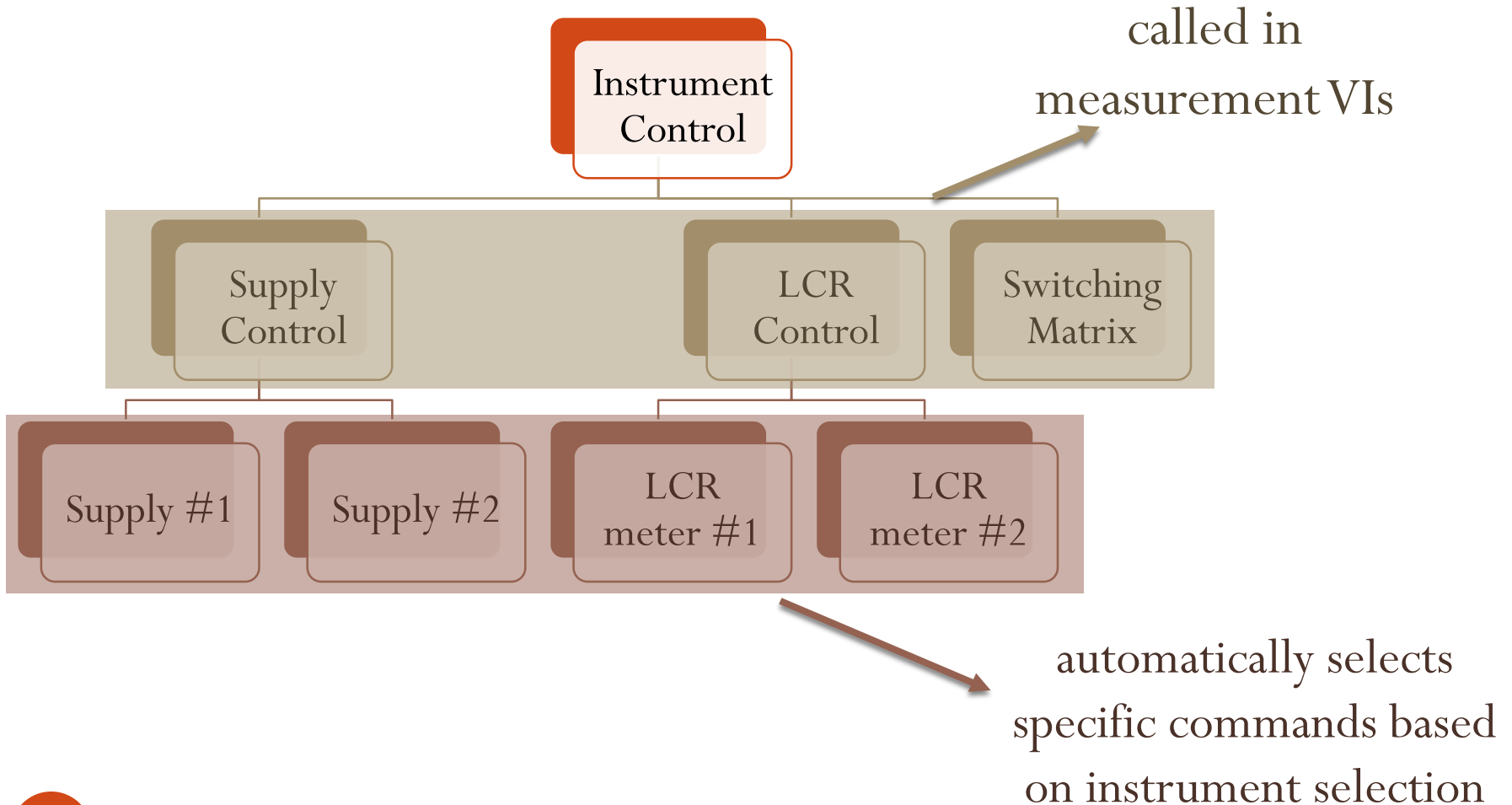
typical commands for all
supplies/LCR meters:
e.g. initialisation, readout



specific variations of
existing commands



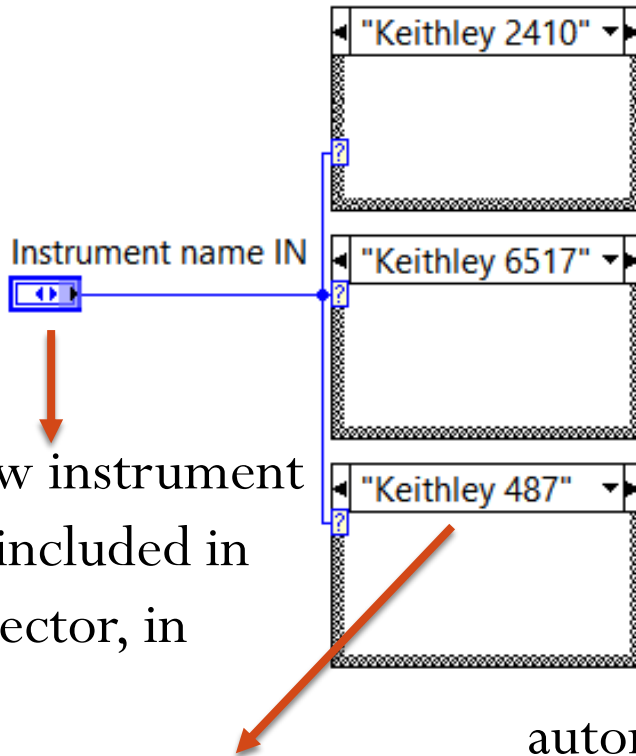
Polymorphism





Why class structure?

Case structure

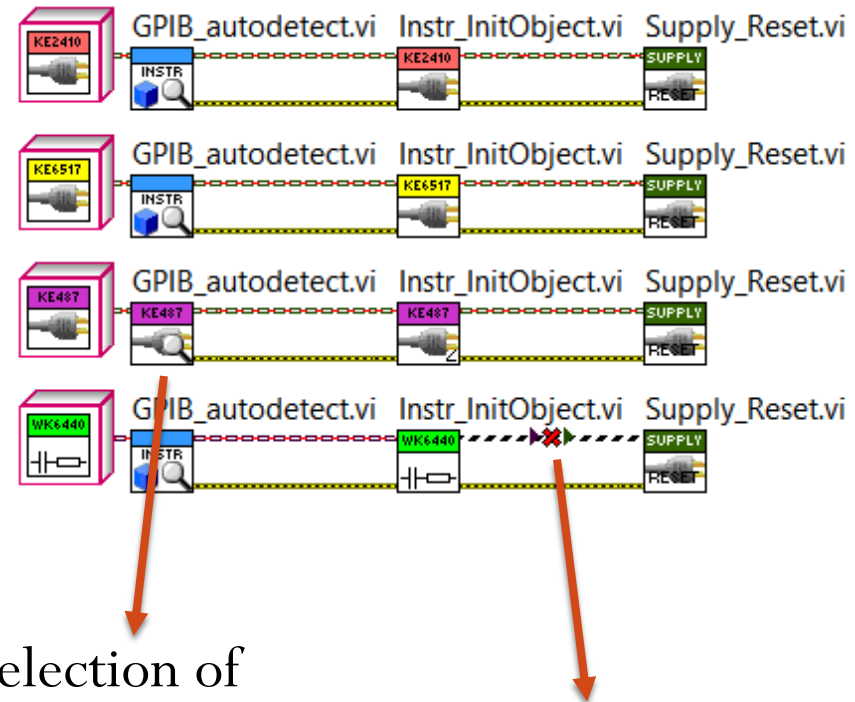


every new instrument must be included in every selector, in every VI

every new instrument needs new case

vs.

Class structure



automatic selection of "special cases" where parent class VIs are overwritten

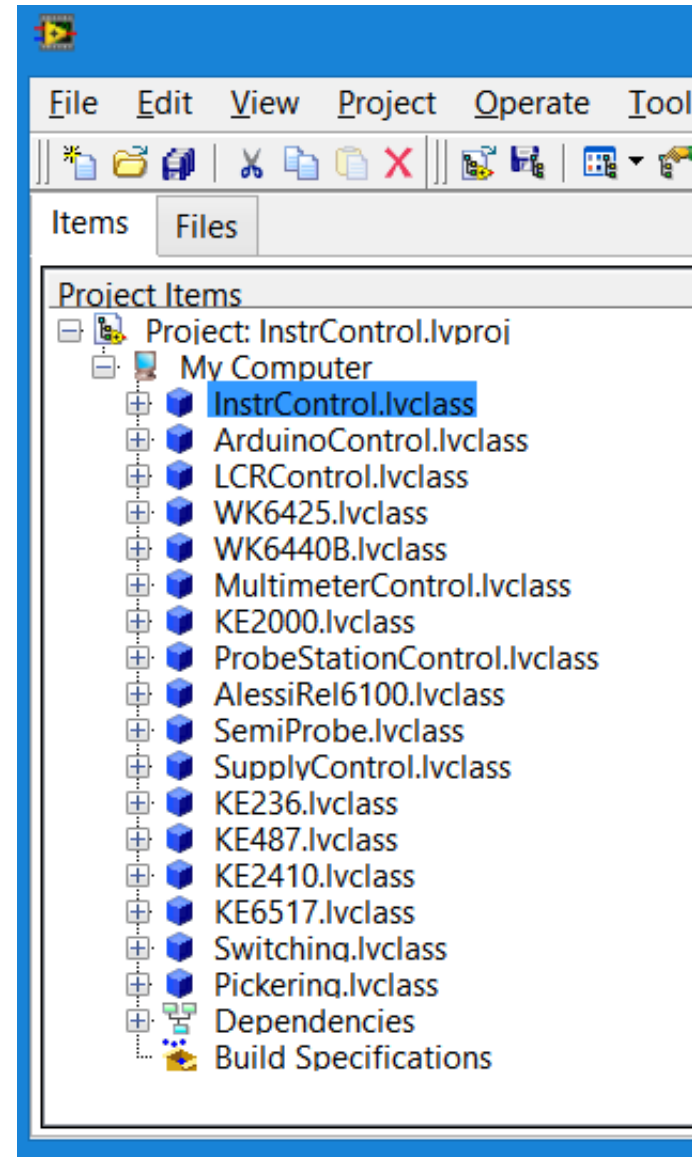
LCR meter not descendant of "Supply"



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)

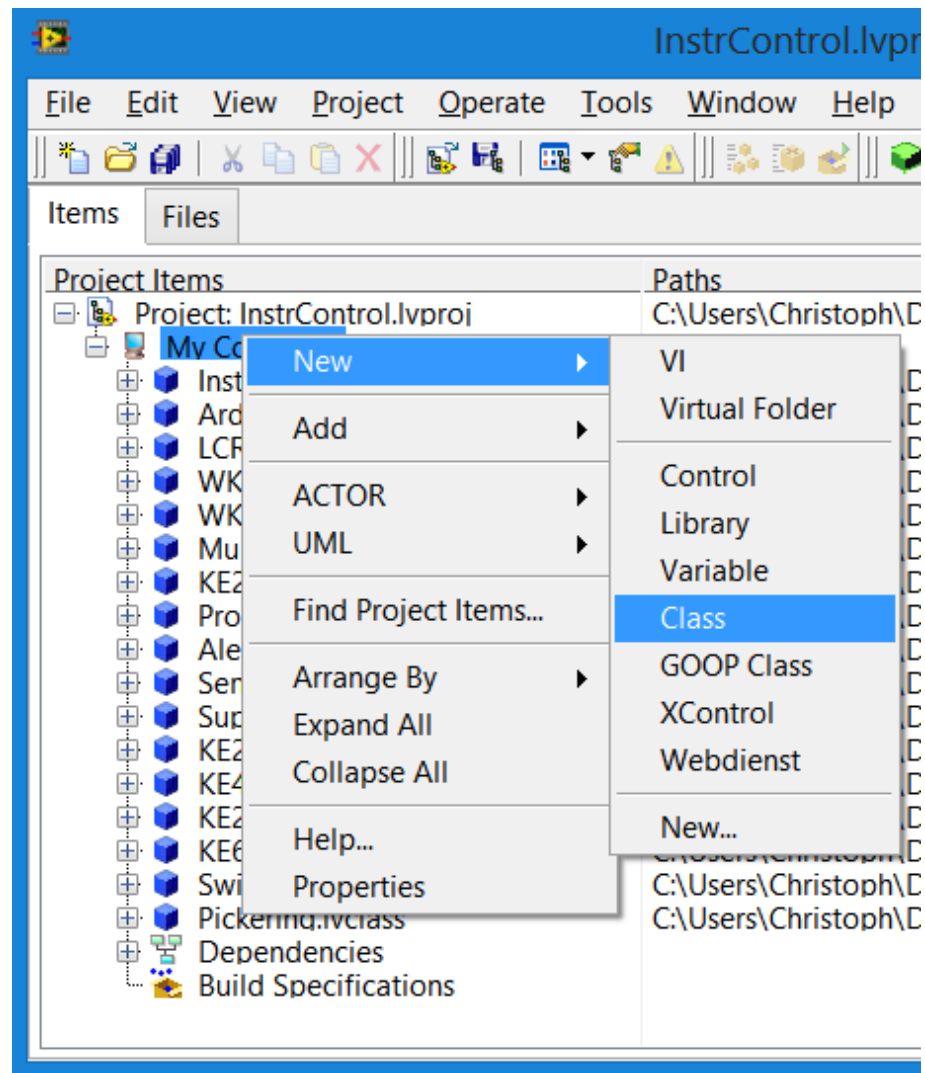




How to include a new instrument

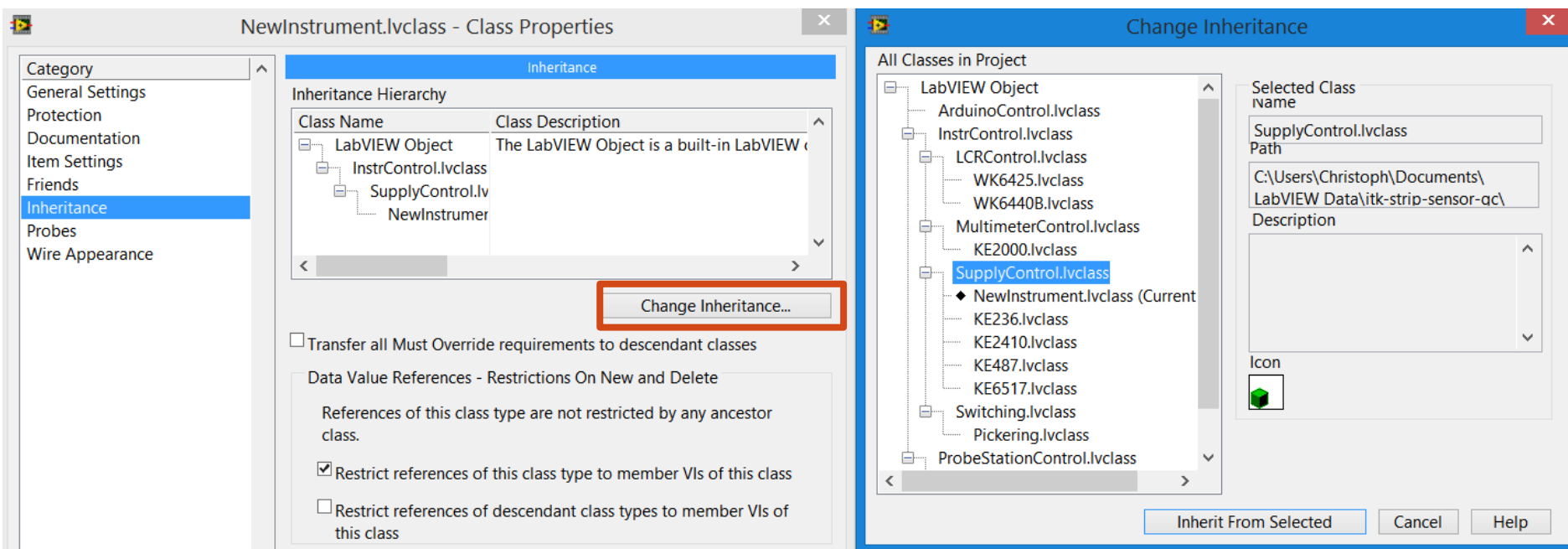
2. Right-click “My Computer”, select “New → Class”

(options may look slightly different depending on the LV version and installed add-ons)





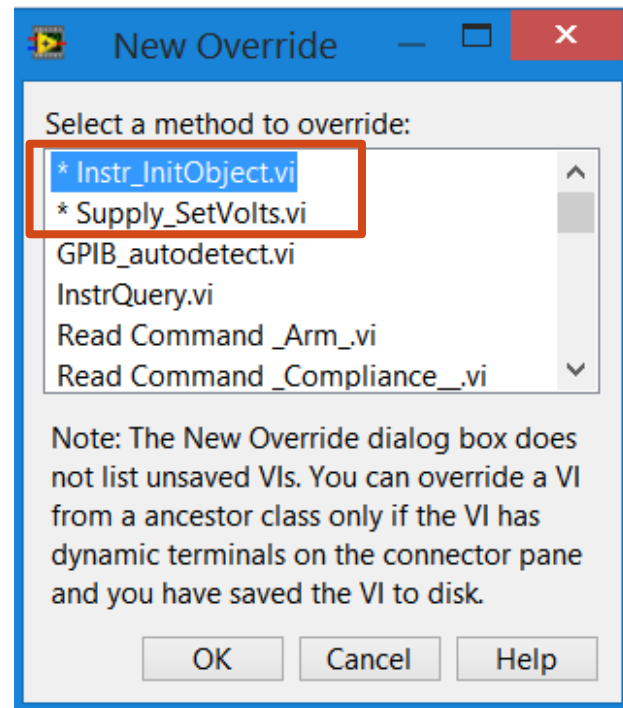
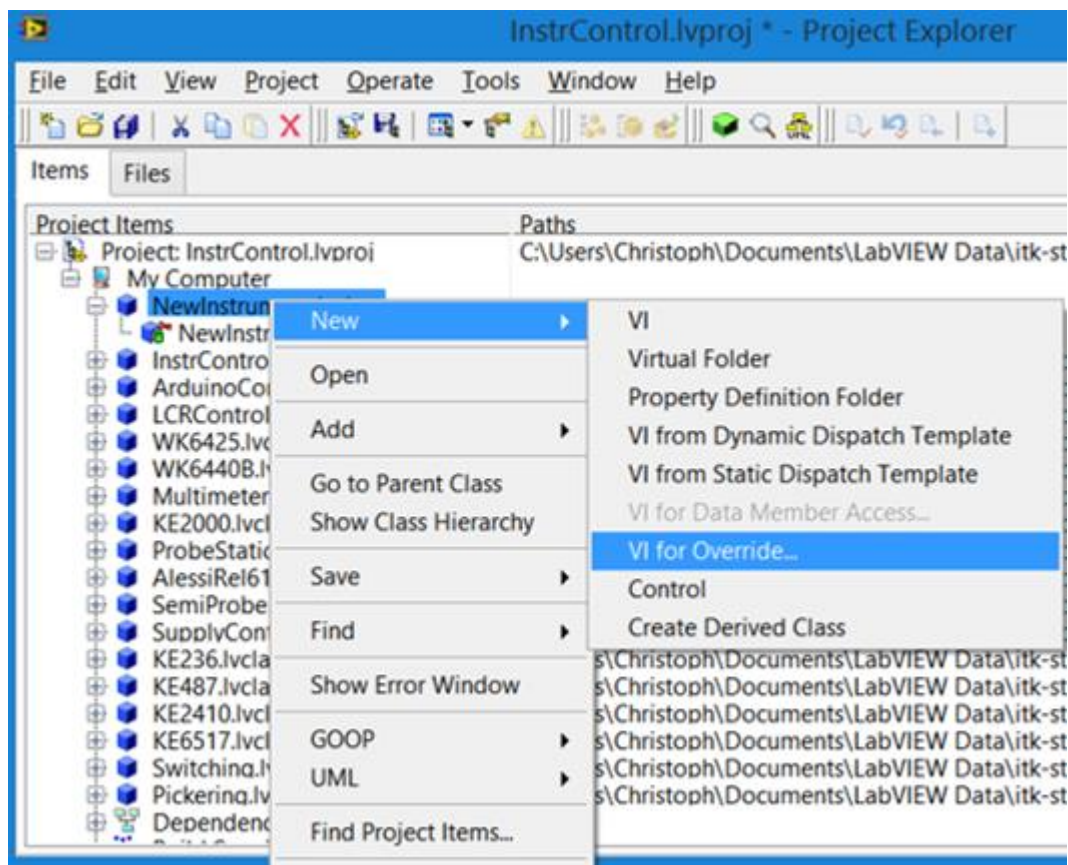
How to include a new instrument



3. give your class a name
4. Right-click new class, select "Properties → Inheritance" and change inheritance accordingly



How to include a new instrument



5. Right-click new class, select "New → VI for Override"
6. VIs with * must be overwritten; all others can be, if necessary



How to include a new instrument

PROPERTY CLUSTERS + DEFAULT VALUES FOR INSTRUMENT SETUP

Keithley 2410 Configurations

Current Limit
1E-4
KE2410

Keithley 236 Configurations

current compliance level
0.0001
compliance/ measurement range (auto:0)
0.1 mA / 110 V
KE236

Keithley 2000 Configurations

Range
Auto
Range Value
100
Function
DC Voltage
KE2000

Keithley 487 Configurations

Compliance
20µA
Range:
Autorange
KE487

Keithley 487.2 Configurations

Compliance
20µA
Range:
Autorange
KE487

Keithley 6517 Configurations

Current Limit
ON
KE6517

Arduino Configuration

VISA address
COM24
T sensor
SHT
Arduino

WayneKerr 6440B Configurations

RC Configuration
serial
AC amplitude [V]
0.1
AC frequency [Hz]
1000
WK6440B

WayneKerr 6425 Configurations

RC Configuration
serial
AC amplitude [V]
0.1
AC frequency [Hz]
1000
WK6425

Picking MUX & Matrix Configurations

Matrix address
1
Mux1 address
4
Mux2 address
5
Matrix used?
☒
Picking
Matrix X Channels
Bias Supply / SMU LO
5
Bias Supply / SMU HI
6
CV Box LO
3
CV Box HI
4
CV Box LO IN
7
CV Box HI IN
8
Volt meter LO
1
Volt meter HI
2
Mux1 C1
9
Mux1 C2

Instrument GPIB addresses

KE 2410

Instrument name	GPIB address	Instrument type
Keithley 2410	24	Supply

KE 6517

Instrument name	GPIB address	Instrument type
Keithley 6517	27	Supply

KE 487

Instrument name	GPIB address	Instrument type
Keithley 487	22	Supply

KE 487.2

Instrument name	GPIB address	Instrument type
Keithley 487.2	19	Supply

KE 236

Instrument name	GPIB address	Instrument type
Keithley 236	16	Supply

WK 6440B

Instrument name	GPIB address	Instrument type
WayneKerr 6440B	23	LCR

WK 6425

Instrument name	GPIB address	Instrument type
WayneKerr 6425	17	LCR

Picking Matrix

Instrument name	GPIB address	Instrument type
Picking Matrix	10	Switching

KE 2000

Instrument name	GPIB address	Instrument type
Keithley 2000	19	Multimeter

7. “InstrumentSetup” global: add default properties cluster and add instrument (name, address, type) in GPIB address cluster
(not GPIB instrument: use arbitrary address, but still give name + type)



How to include a new instrument

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

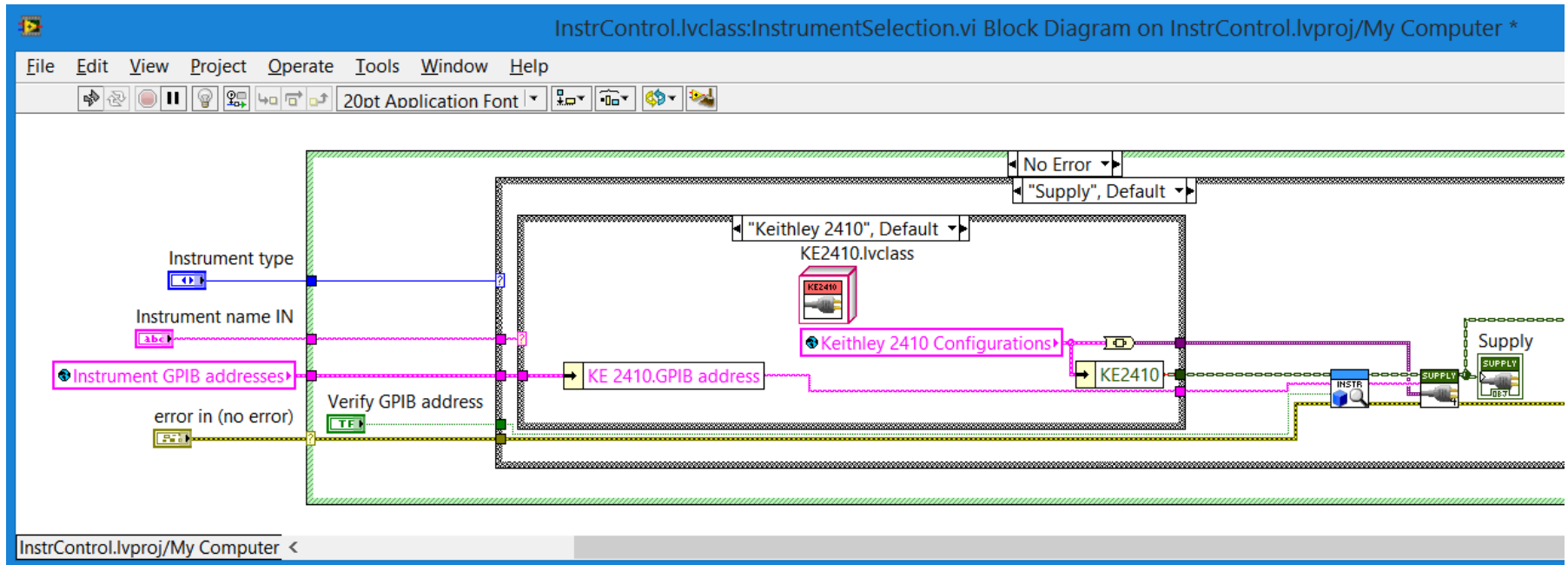
Hardware Configuration

Supply Configuration	Multimeter Configuration
Keithley 236 Keithley 2410 Keithley 487 Keithley 6517	Keithley 2000
GPIO Address KE2410 <input type="text"/> Current Limit [µA] <input type="text" value="100"/>	GPIO Address KE2000 <input type="text"/> Range <input type="text" value="Auto"/> Function <input type="text" value="DC Voltage"/> Range Value (0.1 - 100) <input type="text" value="100"/>
LCR Configuration 	Arduino Configuration
WayneKerr 6440B WayneKerr 6425	Arduino address <input type="text" value="COM24"/>
GPIO Address WK6440B <input type="text"/> RC Configuration <input type="text" value="serial"/> (Only for initialisation) AC amplitude [V] <input type="text" value="0.1"/> AC frequency [Hz] <input type="text" value="1k"/>	

8. add instrument with relevant parameters in “HardwareConfiguration.vi”



How to include a new instrument



9. add instrument in “InstrumentSelection.vi” with correct type
+ name

QC Measurements



Getting started

local configuration file:

- slight differences between different sites
- read local configuration from simple text file

⇒ change accordingly, file in directory

[...]\QCtests\general_VIs\LocalConfig

Exception: directory
for screenshots and
automatic data backup

no need to create folders for all data files:

- LabVIEW code creates file structure automatically starting from parent directory given in local config
- [parent directory] \ [sensor type] \ [batch] \ [wafer number] \ [HPK or database serial no.]



Getting started

measurement VIs ready to be used:

- IV, CV, C_{is} , R_{is} , striptest
- data file header for all measurements according to QC document

notes:

- VIs are intended to be used for “QC only” (of course, can be used for R&D, as well)
- minimal user input necessary for IV, CV, C_{is} , R_{is}
- therefore:
 - ideally, user just needs to type in serial no.
 - measurement settings according to QC specs used by default
 - user has to actively change scan params to deviate from QC specs



Example: IV scan

University of Cambridge

Strip Sensor QC: IV

ATLAS ITK

IV Setup

DUT & Test Description

Sensor Serial No.:

Wafer Serial No.:

Batch No.:

DB Serial No.:

Manufacturer:

Device Category:

User:

Hardware

IV Supply:

Temperature:

Measurement Configuration

Initial volts [V]:

Final volts [V]:

step [V]:

step interval [s]:

hold steps:

V_breakdown limit [V]:

Set current limit? ☐

Current Limit [μ A]:

ATLAS Serial No.

DB Serial No.:

Sensor Serial No.:

Wafer Serial No.:

Batch No.:

Manufacturer:

Device Category:

Current

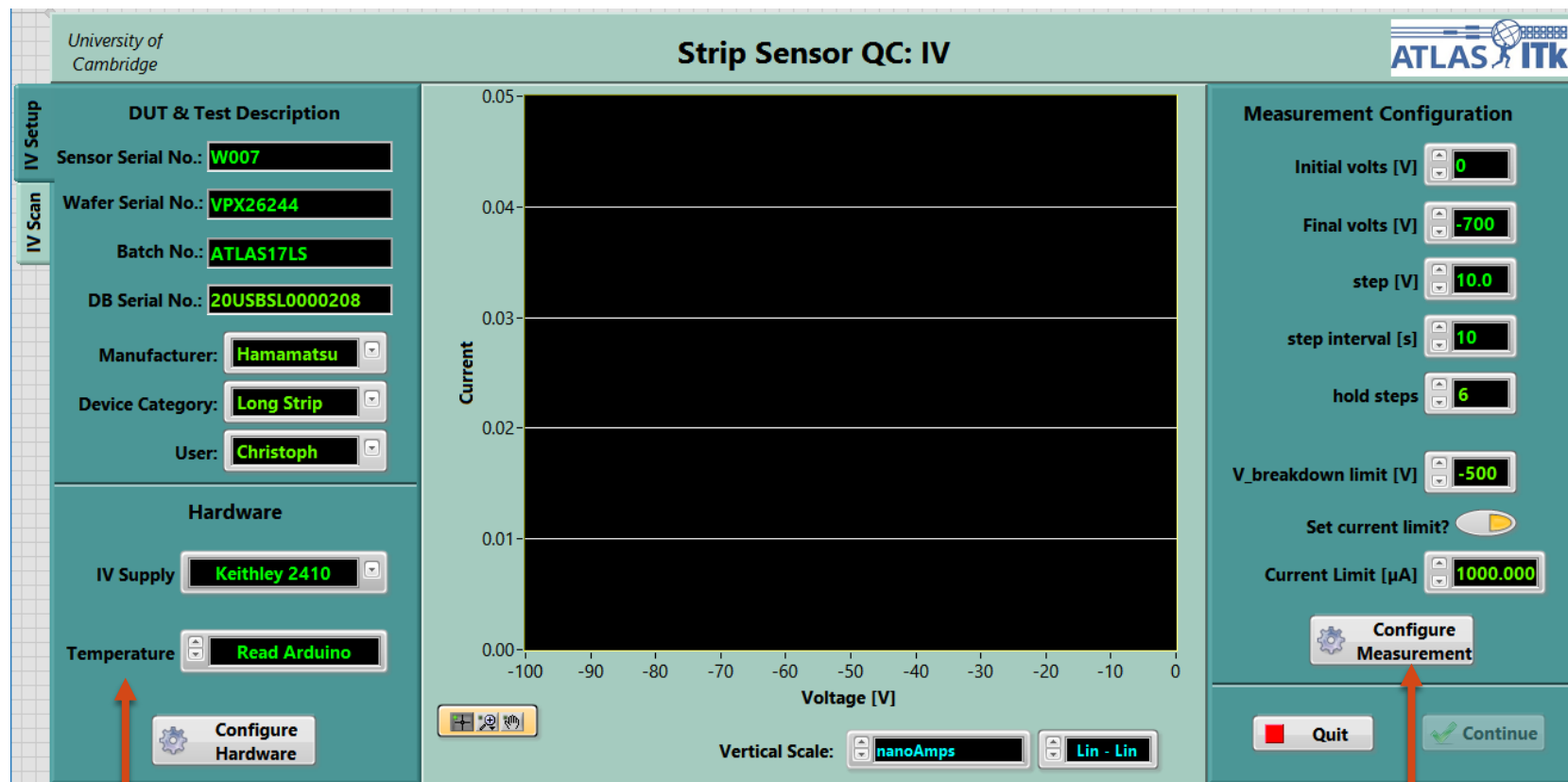
Vertical Scale:

setup step before start of scan:

- enter serial number, etc. in prompt (barcode scanner?)



Example: IV scan

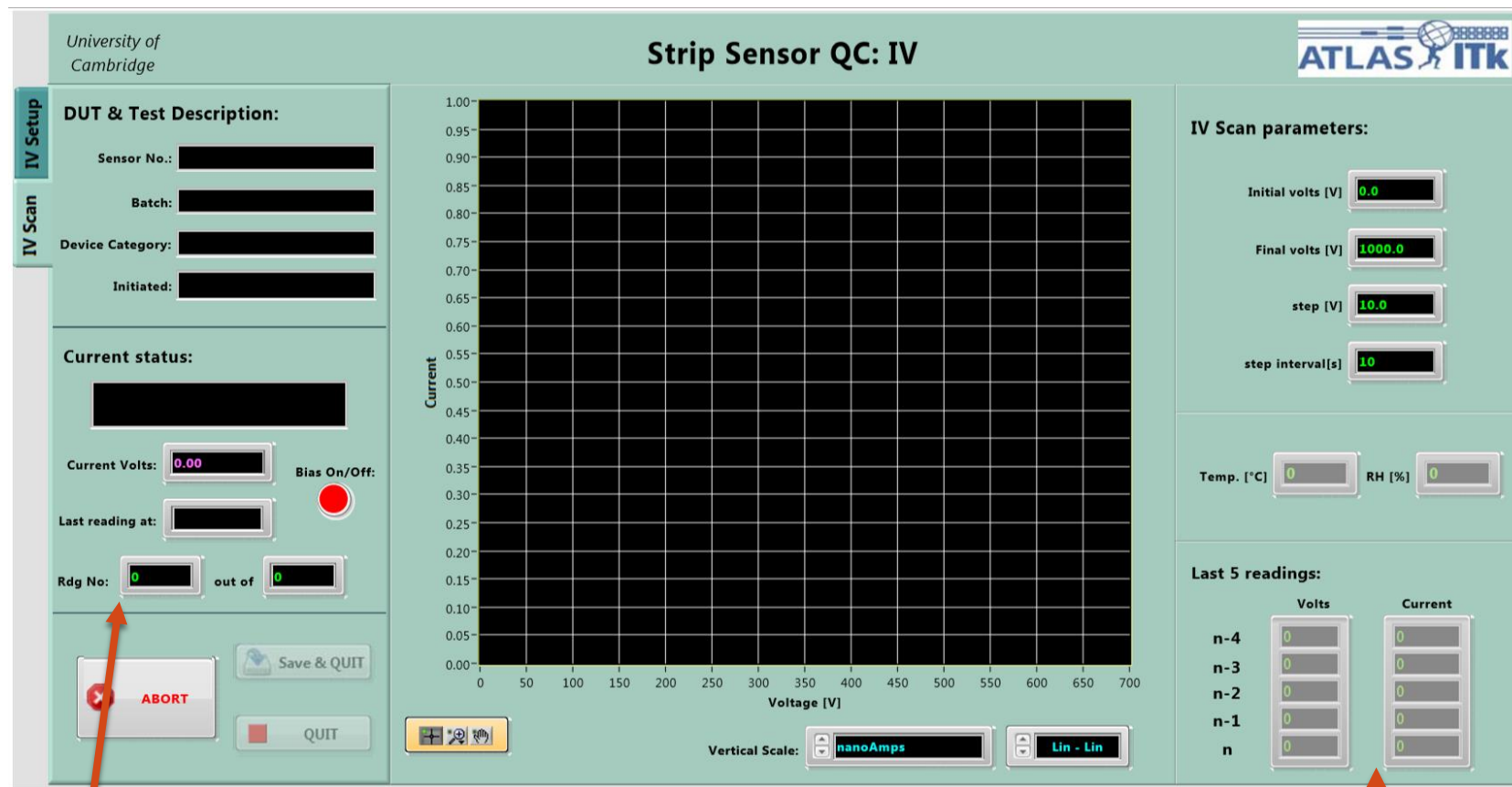


setup step before start of scan:

- select hardware, change configuration
- change scan parameters (only for non-QC measurements!)



Example: IV scan



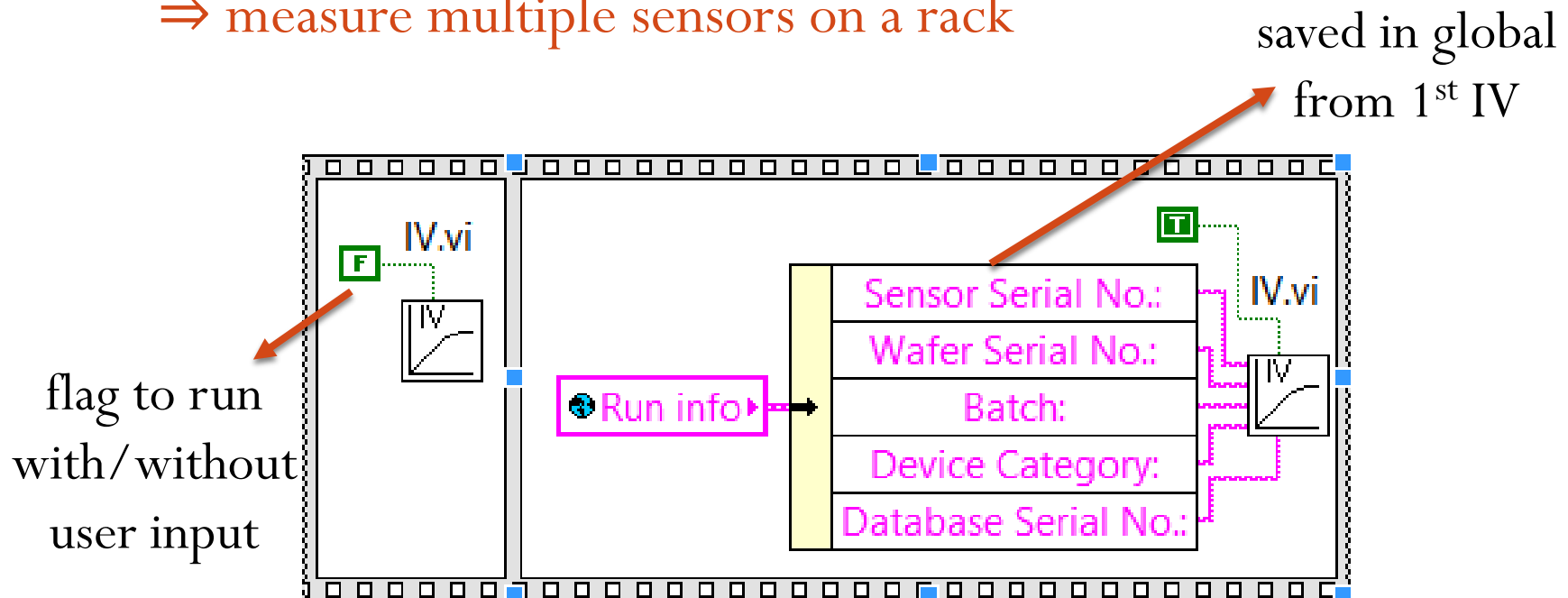
during scan:

- information about status and progress of measurement
- display IV curve and last readings of current + T/RH



DIY measurement sequence

- all QC measurement VIs are standalone scripts
 - they can be used in top-level VIs without the need for any user input (exception: striptest)
- ⇒ chain QC measurements for test sequence, e.g. IV → CV
- ⇒ measure multiple sensors on a rack





Summary and Outlook

- implementation of class structure in instrument control allows for easy addition of all types of new instruments
- this includes:
 - instruments with communication protocols other than GPIB
 - replacements for legacy hardware (e.g. Pickering Matrix)
 - different probestations at individual QC sites
- measurement VIs are ready to be used
- ongoing optimisation and adaption to changing requirements
 - distribute to QC sites for testing and feedback

<https://github.com/chtklein/itk-strip-sensor-qc>

