Designing a flexible measurement system for research applications

Andreas Rydh

Experimental Condensed Matter Physics

Dept. of Physics, Stockholm University

Outline

- **Motivation**
- **Design requirements**
- **Current system overview**
  - Hardware support
  - Software components
  - Capabilities
- **System structure**
- **Application examples**
  - Nanocalorimetry - thermometry & calibration
  - Intrinsic Josephson Junctions - intensity graphs
- **Summary & conclusions**

EKMF lab

ekmf.physto.se

Stockholm University

QuickTime™ and a
TIFF (Uncompressed) decompressor
are needed to see this picture.

Lab instrumentation layout

~ 5 serial

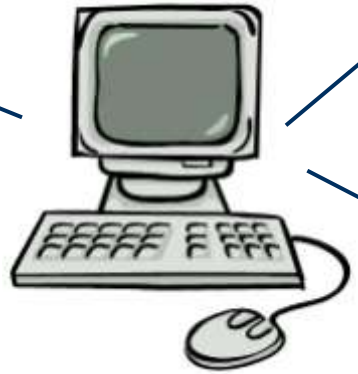
Cryostat

Preamps, connections

FPGA

Logging & supervision

RT

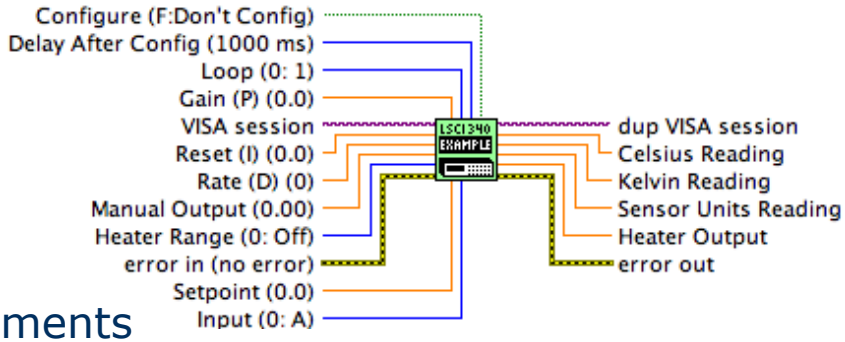


Main system

~ 10 GPIB
Serial, ENET

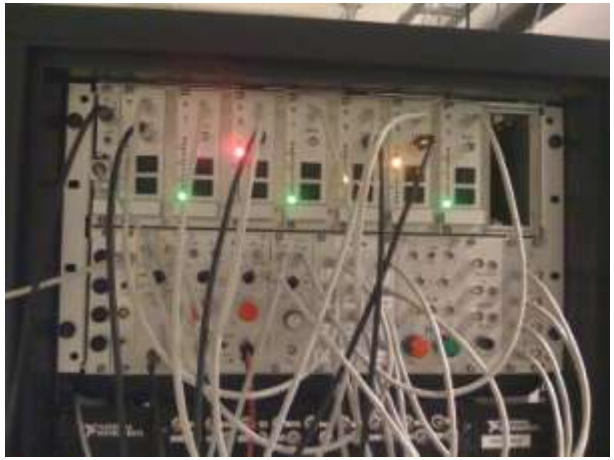
Motivation

- **Typical lab measurement systems**
  - "Hacks" for a particular experiment
  - Fixed set of features
  - Hard to reuse for other measurements
  - Hard to modify if an instrument needs to be replaced
- **Available system components**
  - "Single instrument" examples
  - Hard to interface / connect
  - Missing functionality
- **Desired measurement system**
  - Accommodate new type of experiments
  - Deliver highest possible performance
  - Interface with specialized equipment - including researcher!
  - Direct feedback, immediate results



Design requirements

- **Modular**
  - Easy to add instruments, functionality, etc.
  - Handle missing hardware
- **User-configurable, user friendly**
  - Change experimental setup
  - Configure available instruments - retain configurations
  - Personalize configuration, new experiments
- **Powerful measurements**
  - Minimize dead-time
  - Single samples to high-speed streams
  - Use available hardware functionality
- **Control**
  - Automatic & manual control
  - Direct feedback: data analysis

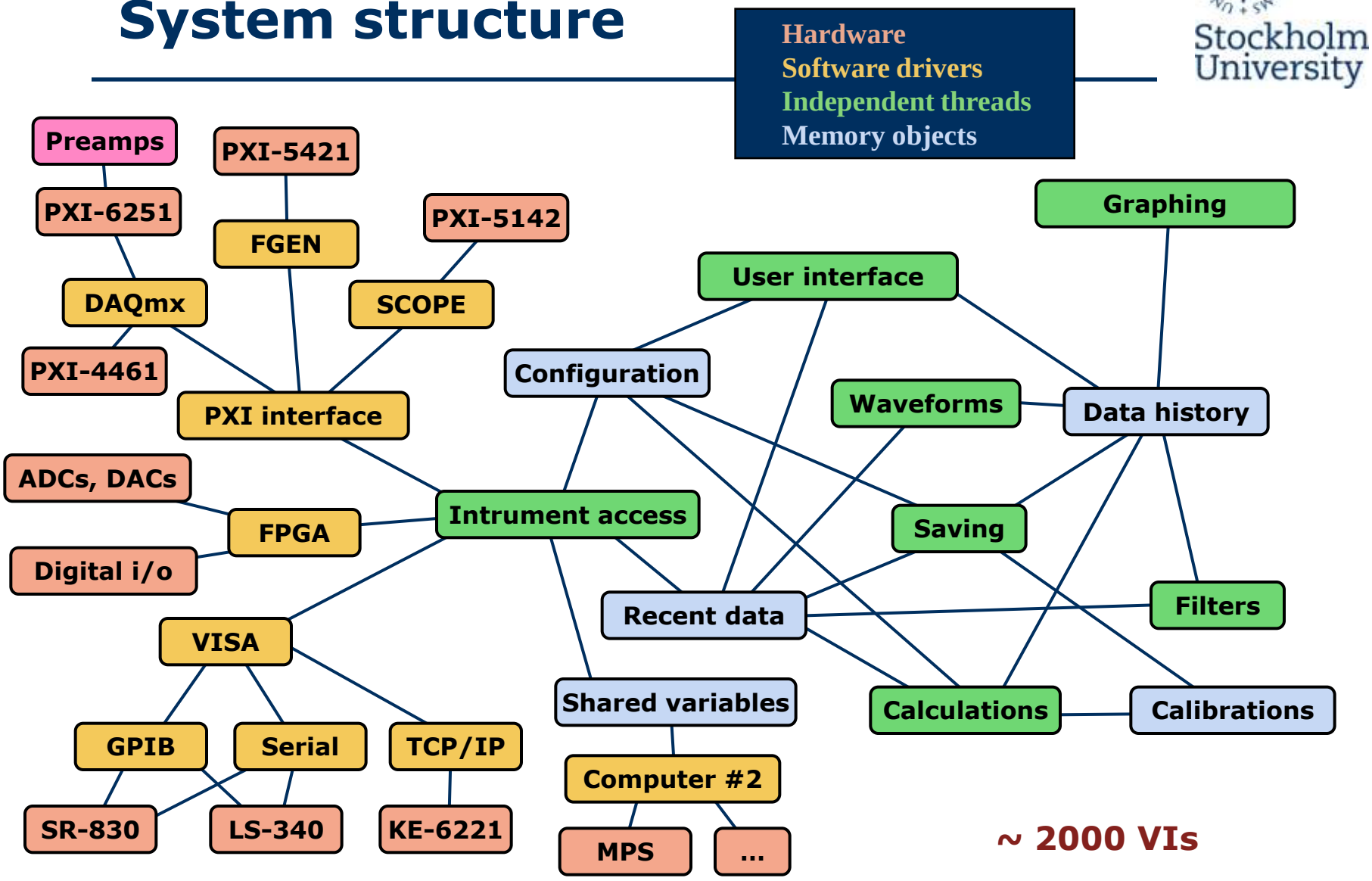


Current system overview

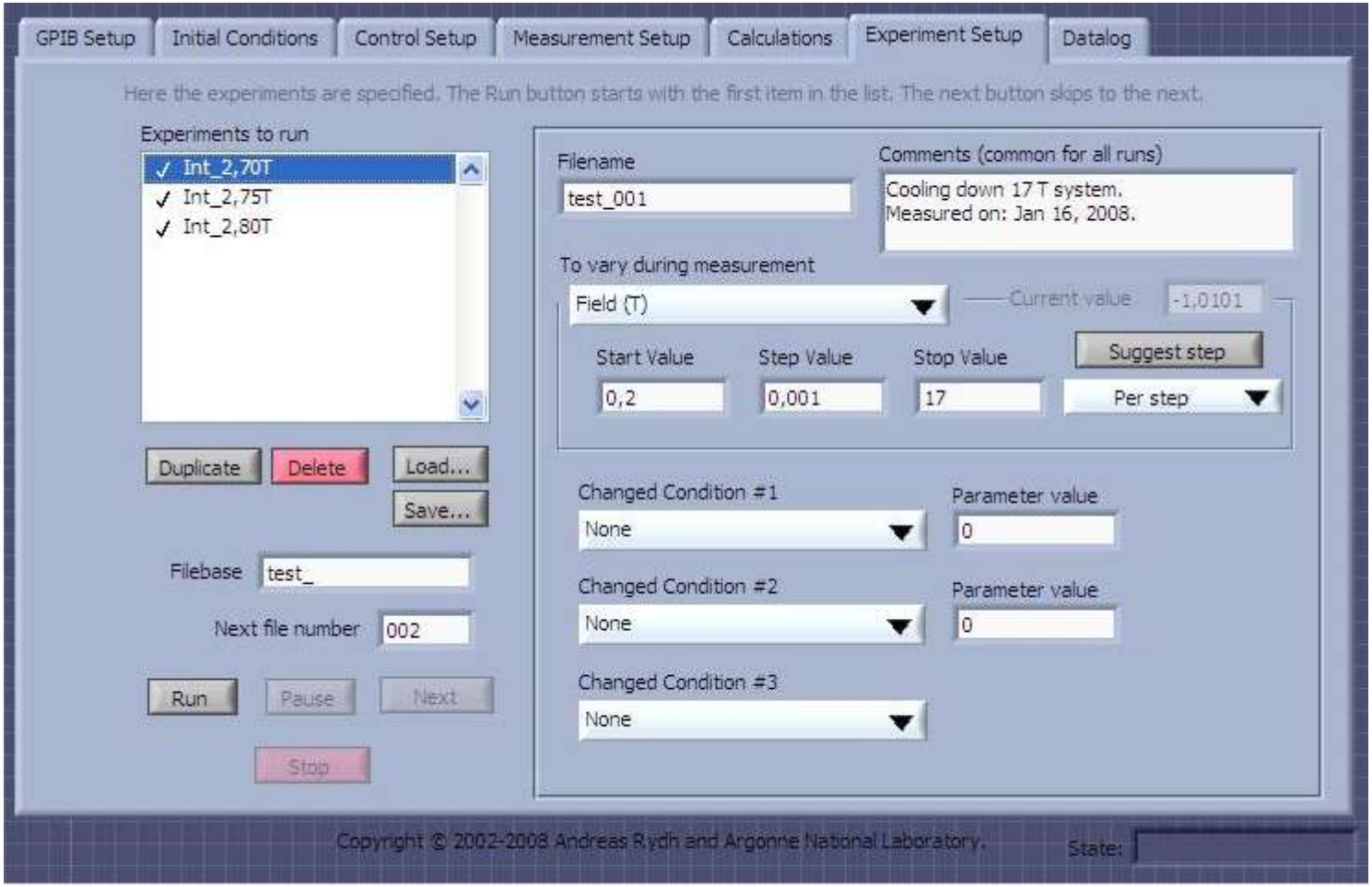
- **LabVIEW-based**
  - Multi-threaded
  - Event based
- **Hardware support**
  - PXI/PCI
    - FGEN (freq. generators)
    - DAQmx (analog & digital input/output)
    - SCOPE (high speed analog input)
  - FPGA
    - DMA streaming
  - VISA (GPIB, Serial, TCP/IP) ~ 30 instruments
  - ActiveX (Compiled windows drivers)
  - Networking
    - Shared variables, Sockets

- **Software components**
  - Basic functionality: logging, plotting, configuring
  - Extended functionality: filters, integration, streaming
  - Automatic control: sequences, feedback loops, conditions
  - Real-time analysis: FFT, intensity graphs, data transformations

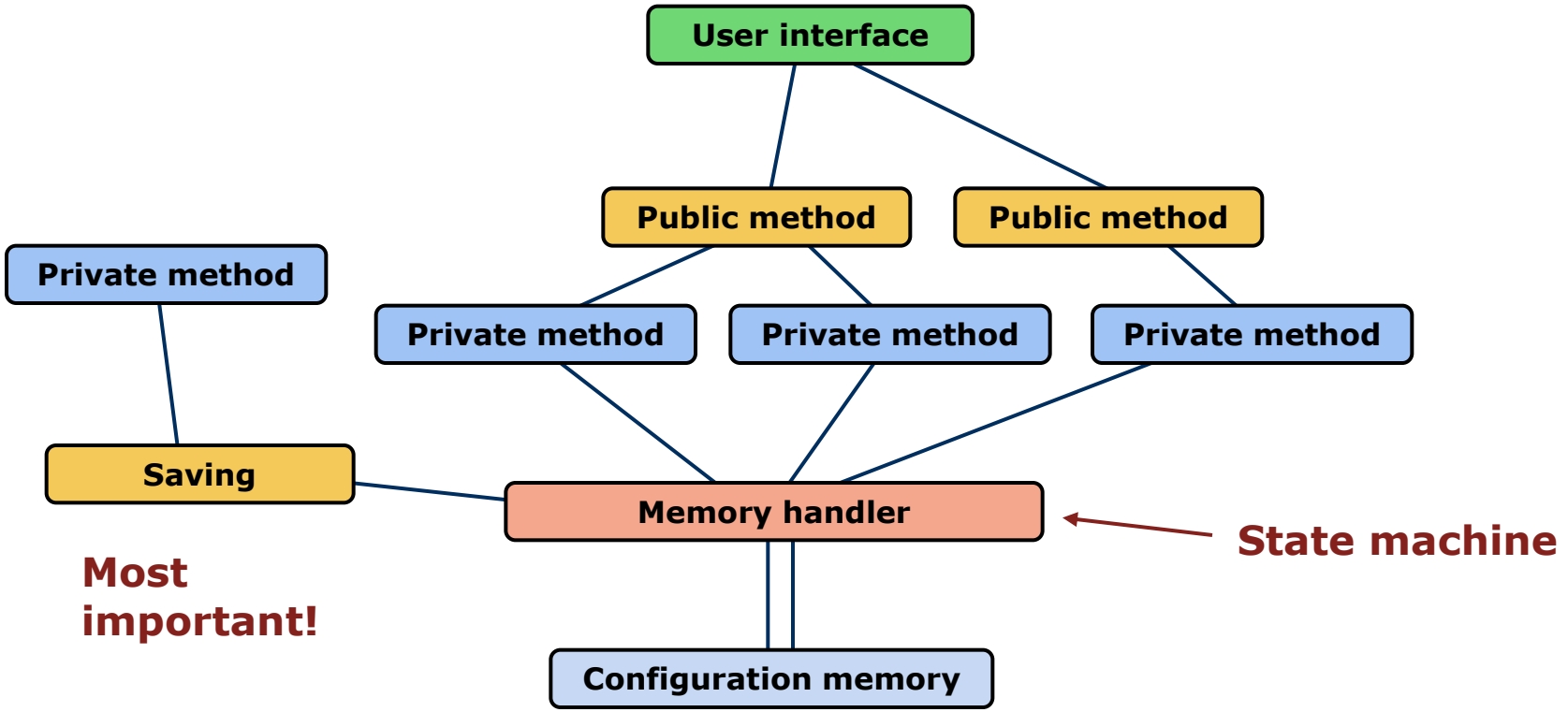
System structure



User interface - remember config!

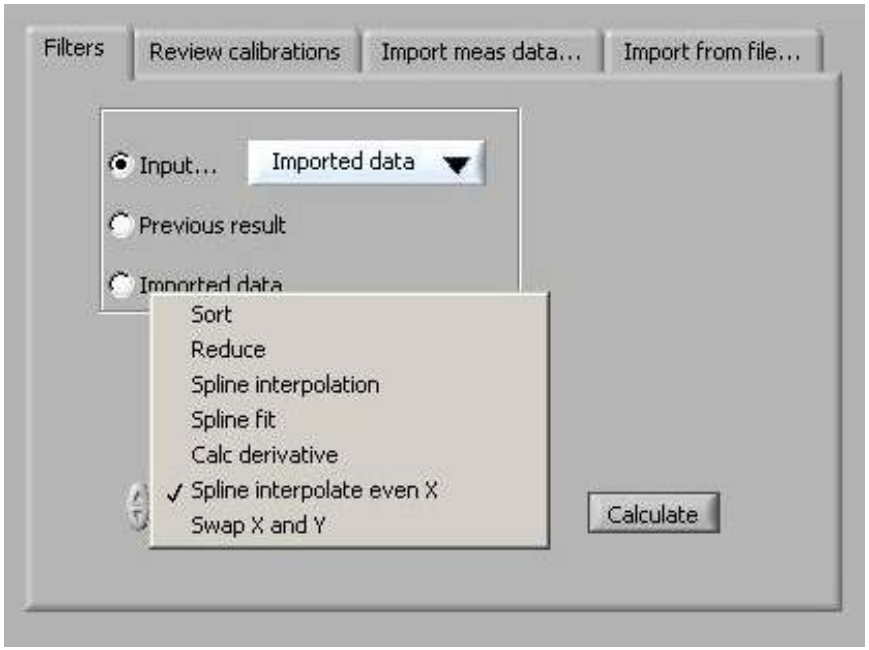


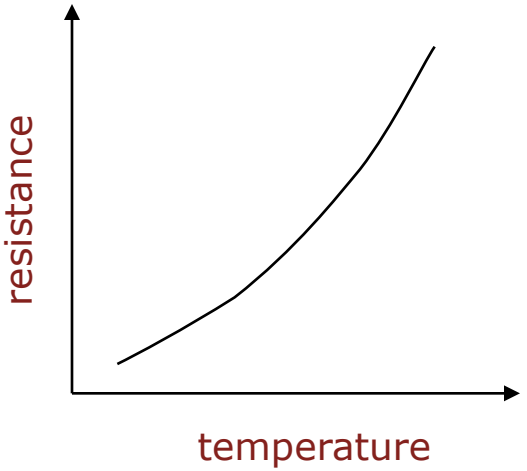
Memory objects



Application examples

- **Nanocalorimetry - thermometry & calibration**
  - Measure resistance, FPGA lock-in
  - Generate calibration
  - Use calibration to obtain temperature
  - Temperature feed-back loop, offset scan, etc.





Application examples

- **Intrinsic Josephson Junctions - intensity graphs**
  - Sample I vs V, streamed data
  - Process / filter data, offset adjust
  - Build-up intensity graph

Graphs not shown

Summary & conclusions

- **Instrument & hardware support**
  - OK drivers for most instruments but not plug-and-play
- **Independent driver objects**
  - NI is doing a great job
  - Other instrument makers: improve!
    Still far from being lvlibs with public and private methods
- **Writing code**
  - Key is user preferences
  - Modular code - object oriented even if no LabVIEW objects
  - Use threading - this is where LabVIEW excels
  - Don't forget data analysis, graphing, interactivity, …
- **Goal**
  - No additional data treatment needed
  - Improve data quality
  - Added functionality: "what the researcher wants"
  - Plug-and-play, any-use instrumentation