

BASE & LabVIEW Object Oriented Data Logger

Axel Pontén

Supervisor: Jack Devlin, PhD

8/5/19

What is BASE?

- Baryon Antibaryon Symmetry Experiment
- Principal Investigator: Dr. Stefan Ulmer
- Small group



Best beard at CERN

What do we measure?

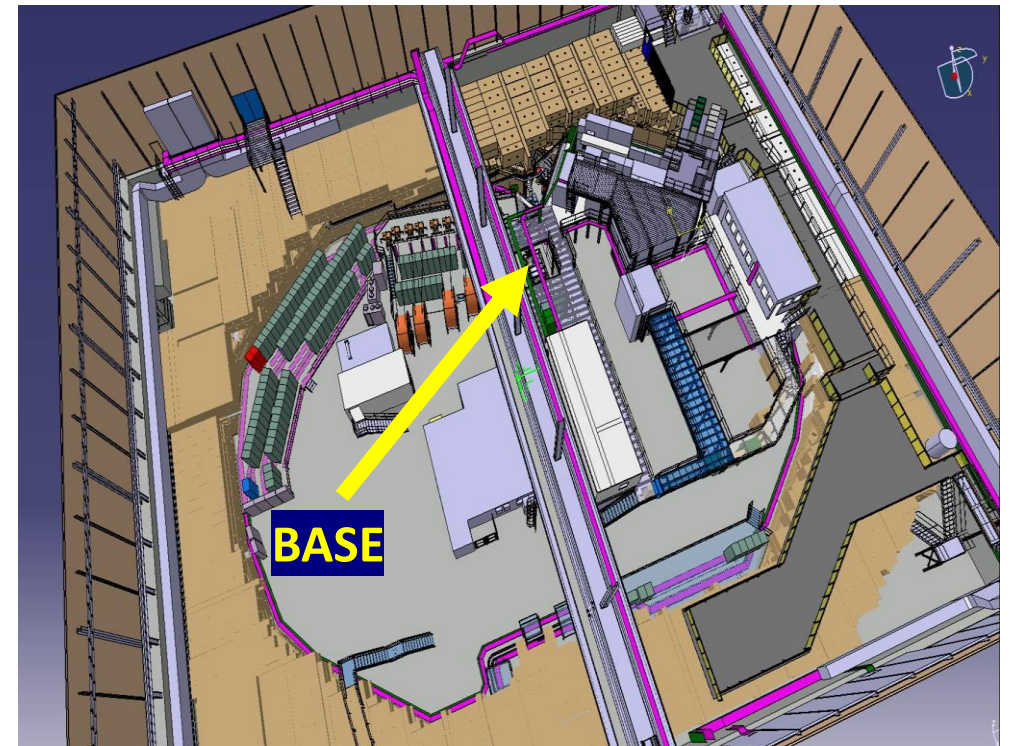
Fundamental properties of **protons** and **antiprotons**:

- *charge/mass ratio*
- *g-factor*

Why do we measure?

CPT symmetry says they should be the **same**! Is that the case? Let's measure these properties to very high precision to find out!

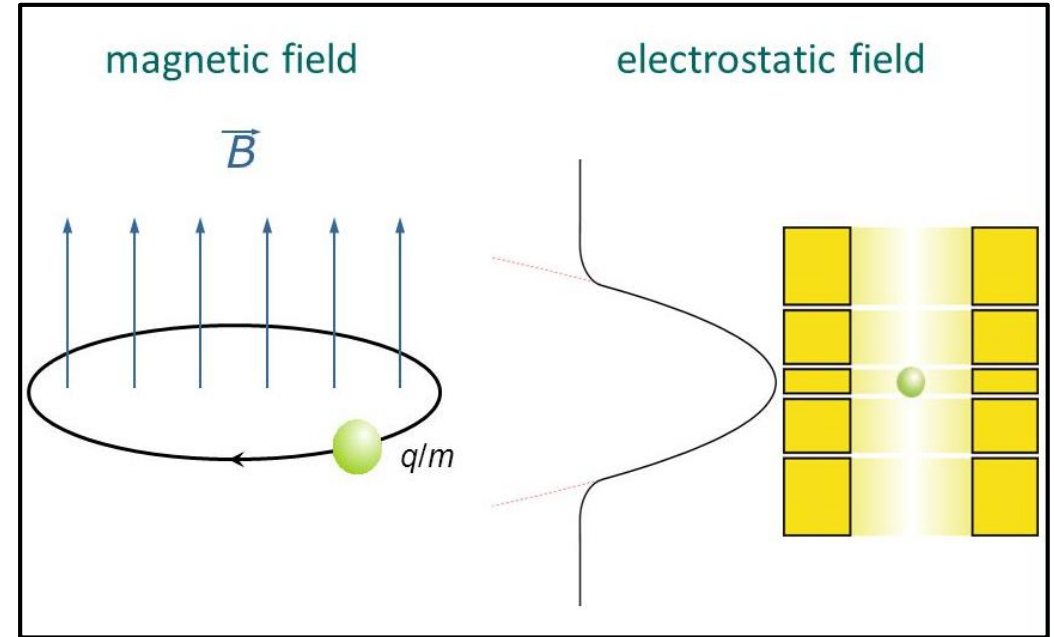
Could a difference explain matter/antimatter imbalance?



How does BASE test CPT?

- Trap individual *low energy* protons and antiprotons in a **Penning Trap**
- **Magnetic & Electrostatic fields** trap particle radially and axially
- Measure cyclotron frequency and Larmor frequency (Continuous Stern-Gerlach Effect)

Principle of a Penning Trap

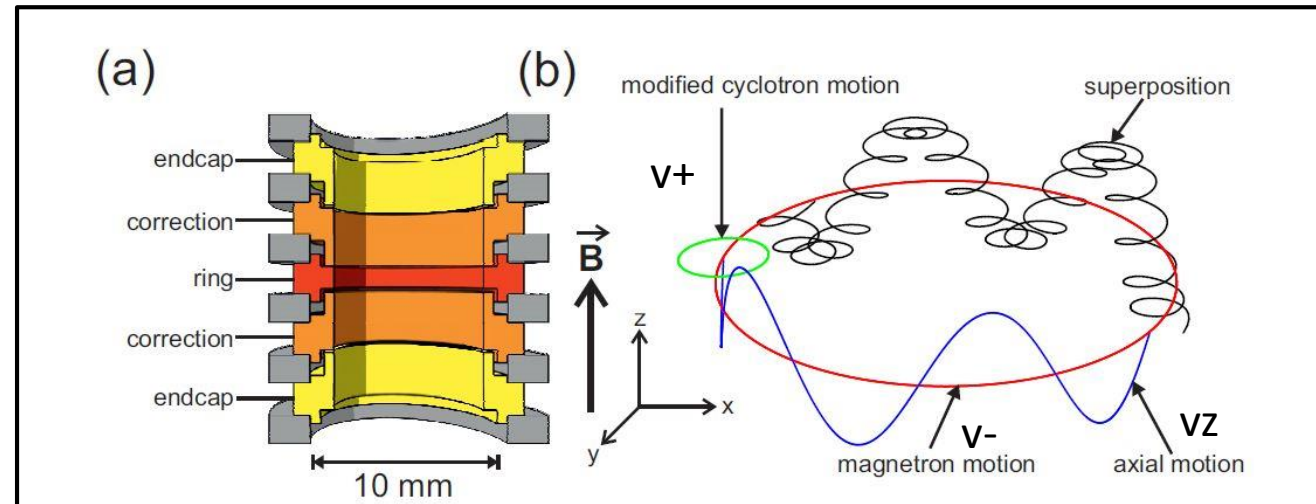


Fundamental Properties Tested:

MASS/CHARGE: Free Cyclotron Frequency

MAGNETIC MOMENT: Cyclotron Frequency & Larmor Frequency (Spin Precession)

LIFETIME: How long can we trap, lower limit on lifetime

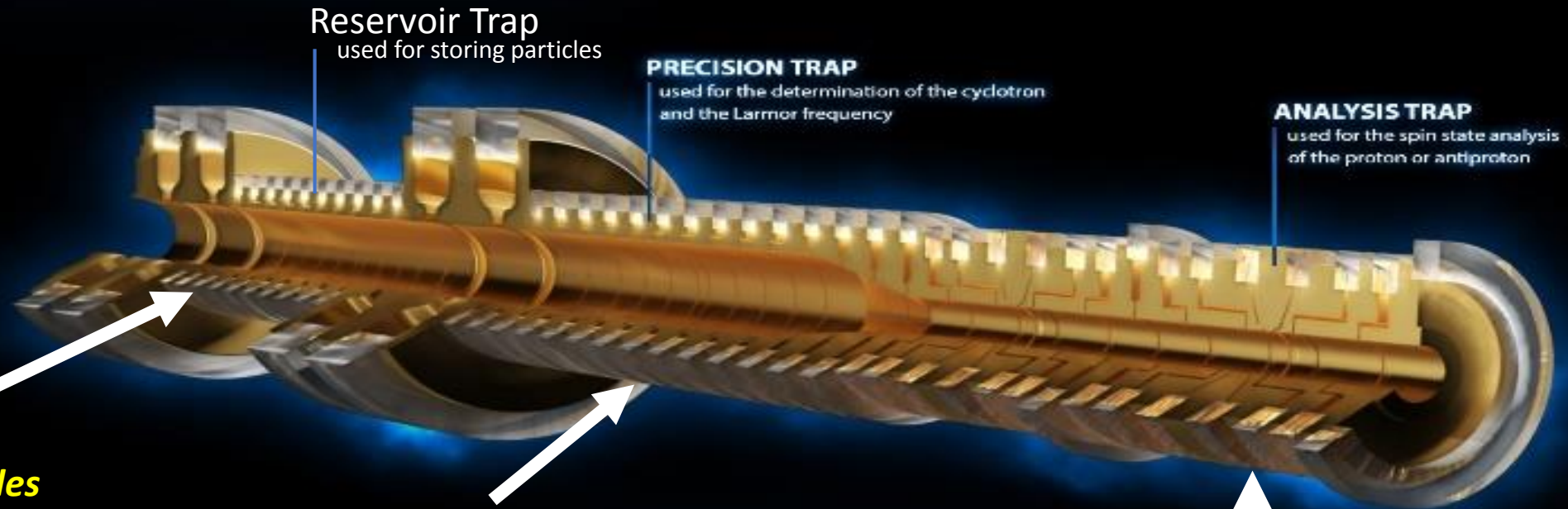


BASE measured lifetime of pbar > 10.2 a with a confidence level of 68%

A. Ponten

Free cyclotron freq. $\rightarrow \nu_c^2 = \nu_+^2 + \nu_-^2 + \nu_z^2$ 3

BASE Multi-Penning Trap



Reservoir Trap
used for storing particles

PRECISION TRAP
used for the determination of the cyclotron
and the Larmor frequency

ANALYSIS TRAP
used for the spin state analysis
of the proton or antiproton

Store Particles

Have stored
particles for
405 days!

Measure Charge/Mass

Current precision of
 $\sim 10^{-12}$ (ppt)

Measure Magnetic Moment

Current precision of $\sim 10^{-9}$ (ppb)



Antilbaryon Baryon Symmetry Experiment

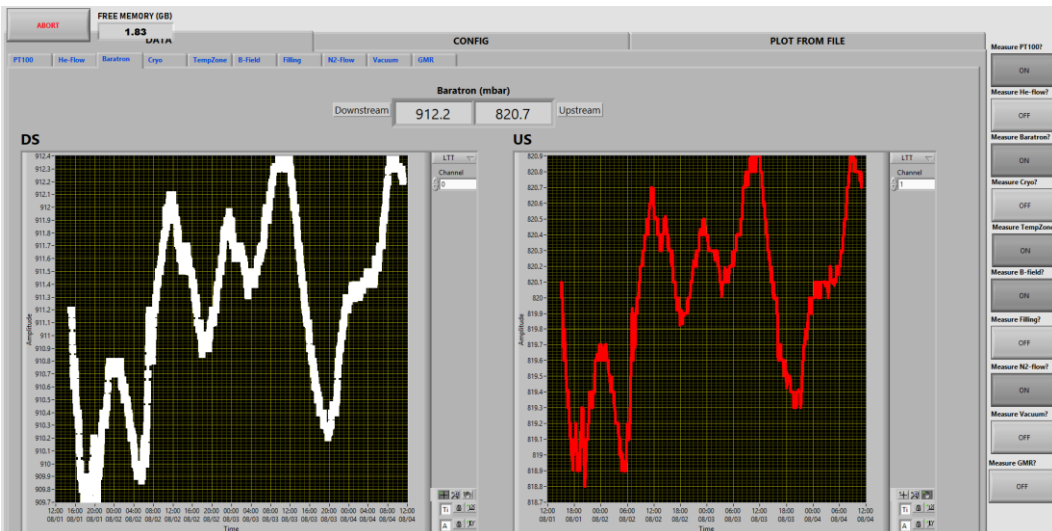
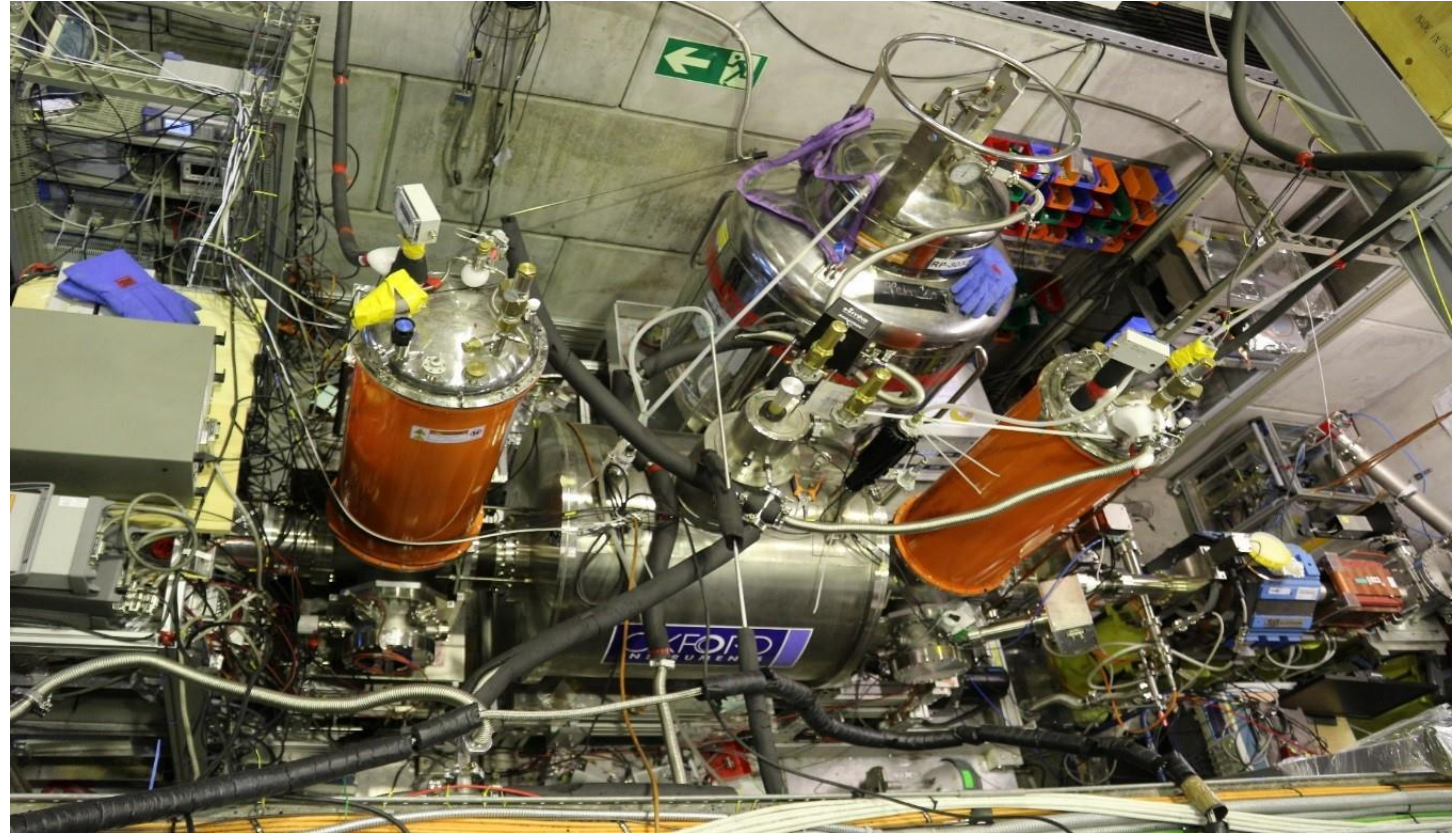
High precision → Sensitive to environmental fluctuations

- Need to keep track of stability in temperature, pressure, vacuum, etc.
- Can alter data significantly!
- Robust data logger needed

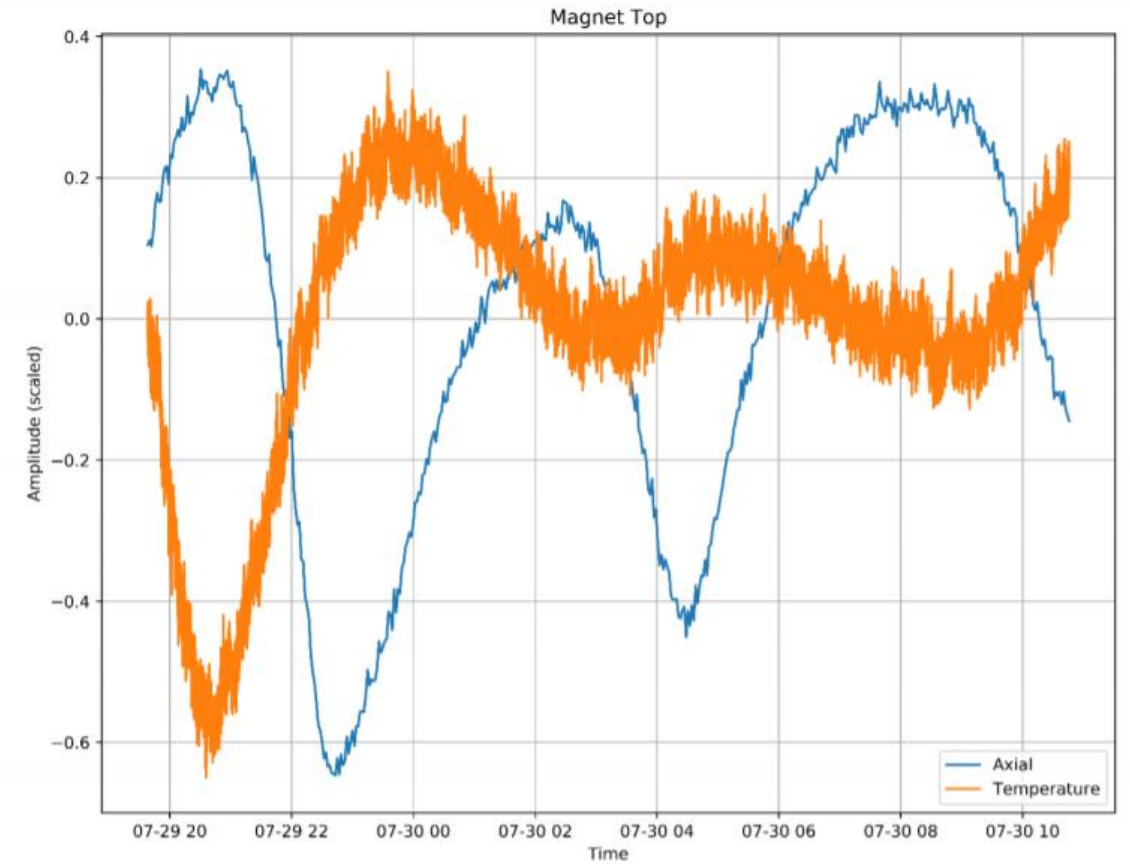
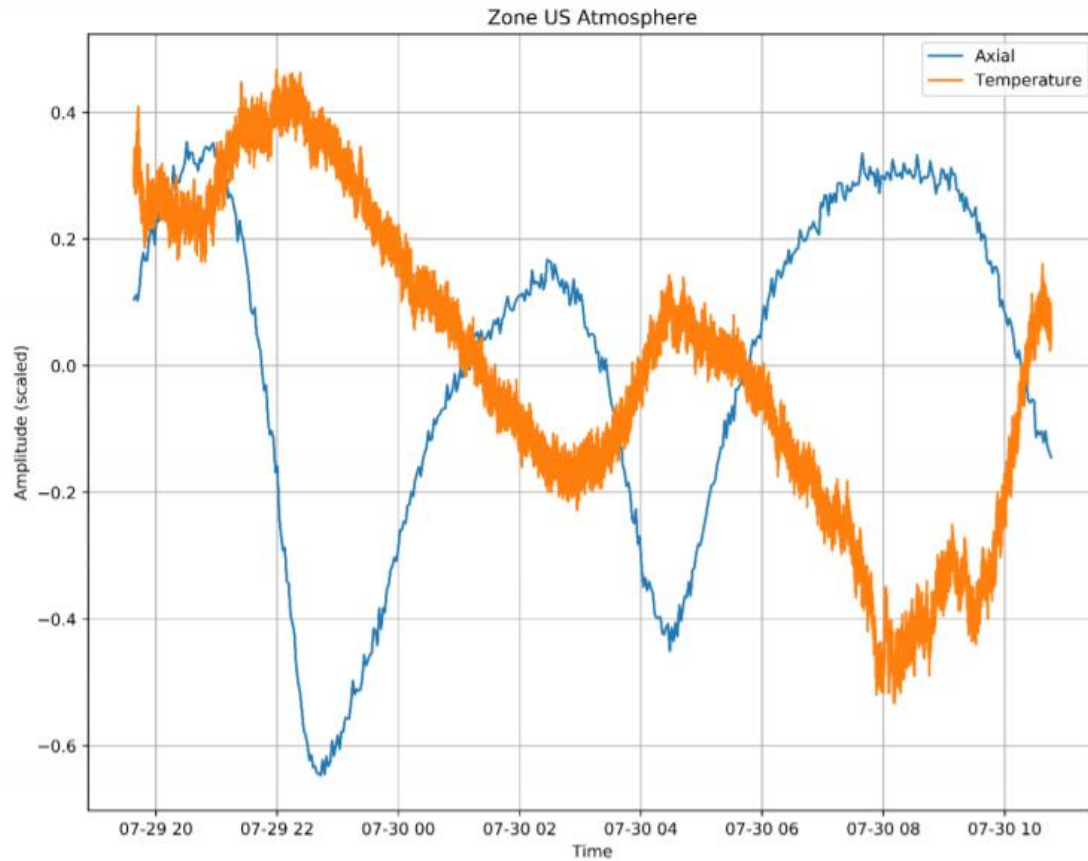
Small Experimental Zone:

- Keep datalogger on one computer

BASE uses a **LabVIEW** data logger, written by the experimentalists themselves.



Example: Temperature dependency of axial frequency in Penning Trap



Purpose of logger: *Where can improvements be made to reduce environmental fluctuations?*

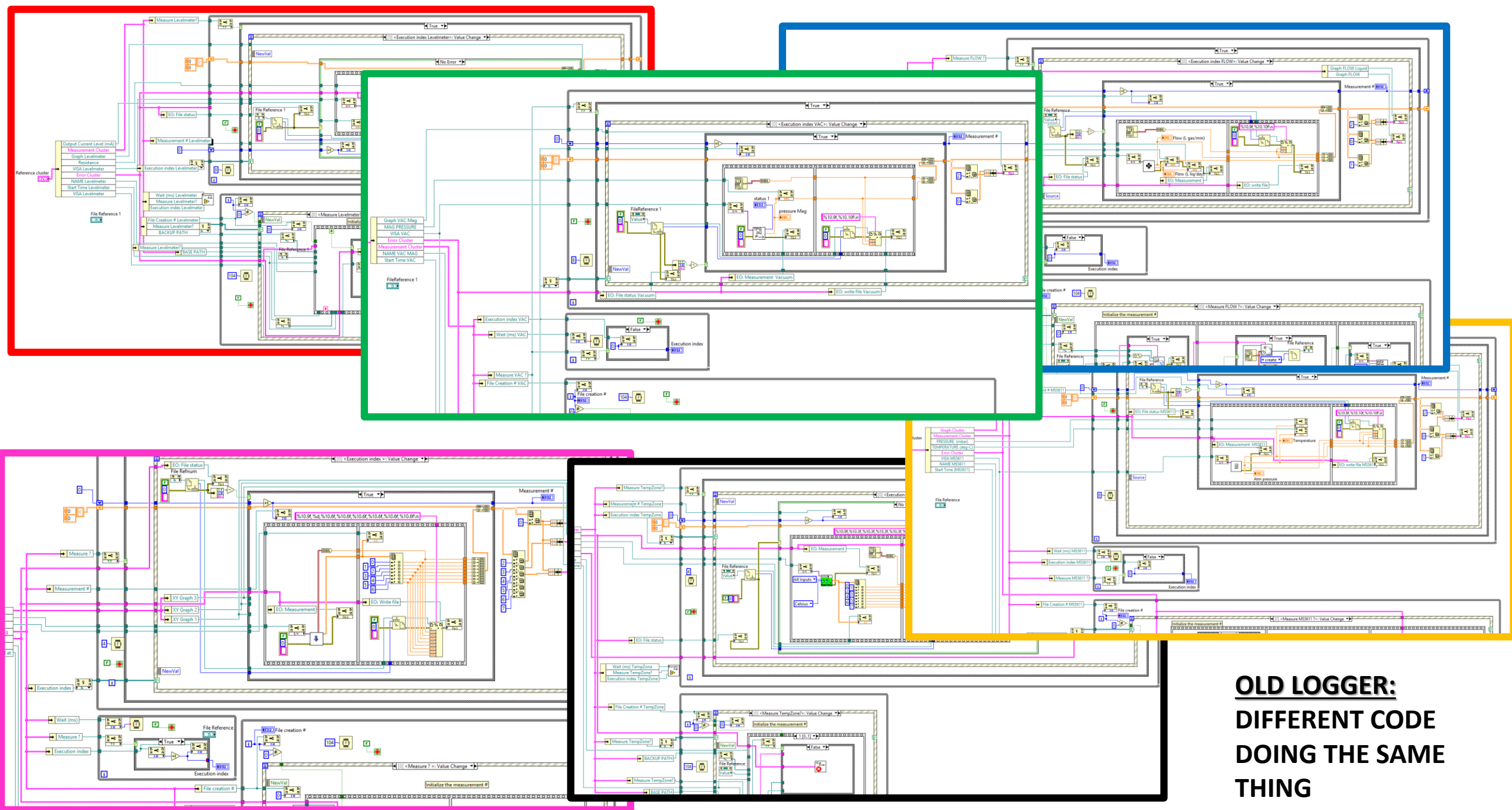
My work at BASE: Rewrite Data Logger

Old Logger Limitations:

- Same tasks for each instrument (Read Data, Plot Data, Save to File) but limited reuse of code.
- No encapsulation. If bug or error, don't know from where. **Example: Memory leak**
- Inserting new instrument -> write new code for every aspect

New Logger Solutions:

- **LabVIEW Object Oriented Programming**
- Create **classes**: File, Graph, Instrument, etc.
- Each instrument **inherits** from a parent *Instrument* class and **overrides** *Measure Data*-function.
- This allows for good reuse of common functions and encapsulates the code (e.g. if error with file writing, only need to look at File-class)
- Inserting new instrument -> only write new *Measure Data* function



**OLD LOGGER:
DIFFERENT CODE
DOING THE SAME
THING**

CLASS HIERARCHY

File

Contains a file path & file refnum. Has VIs to initialize and write to file.

Measurement

Data members:

- *Instrument* [OBJECT]
- *File* [OBJECT]
- *Logger Graphs* [OBJECT]
- *Logger Graph Arrays* [OBJECT]
- *Logger Errors* [OBJECT]
- *Measurement Config* [CLUSTER]

Logger Graph Arrays

Contains two 2D-arrays: Real Time (RT) and Long Time Trend (LTT). Columns in the array represents data channel. LTT performs moving averages to keep array size constant. The arrays are sent to Logger Graphs "Plot.vi".

Logger Errors

Contains error indicators for config, device and file.

Instrument

Abstract class. Each device will inherit and override the "Measure Data.vi"

Logger Graphs

Contains a "Logger Graph.ctf" which is a cluster of an XY-graph, channel selector and RT/LTT enum. Will plot data from Logger Graph Arrays.

Temp. probe

Flowmeter

Baratron

Etc.

MAIN VI

Front Panel

Data

- Tab page for each instrument
- Logger Graph Control
 - Real Time
 - Long Time Trend (Continuous Averaging)
- Last Values Array

Config

- Config control:
 - File path
 - Sample Time
 - Duration of real time plots
 - Comment to be saved as header in data csv file
- Error Cluster

Initialization parameters

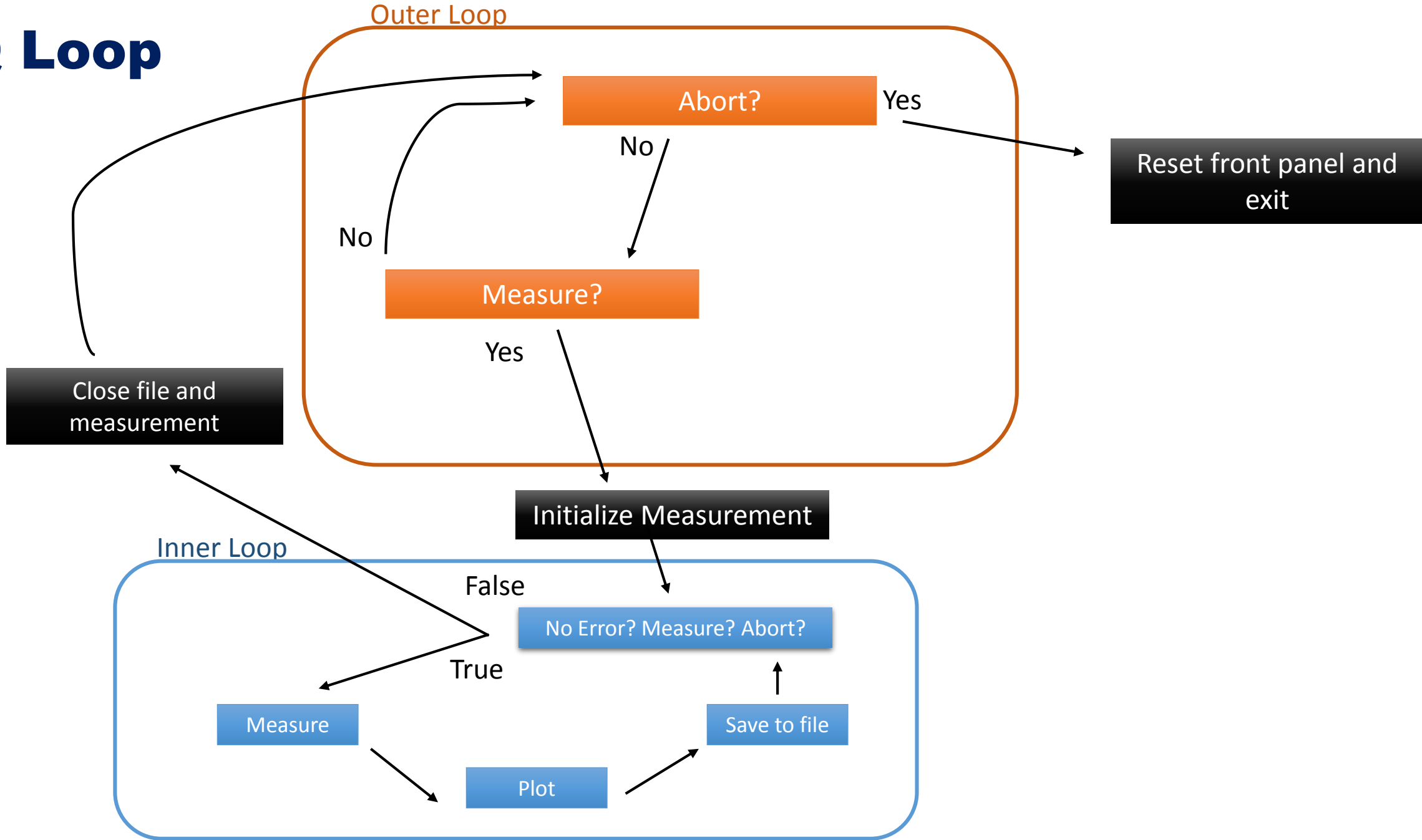
- References:
 - Config ctrl
 - Logger Graph ctrls
 - Measure?-button
 - Abort?-button
 - Last Values Array
 - Error Cluster
- Objects:
 - Instrument object (select from child classes)

Block Diagram

DAQ Loop

- Each device calls reentrant VI that contains the DAQ Loop
- Dynamically overrides "Measure Data.vi" depending on which instrument child the device belongs to

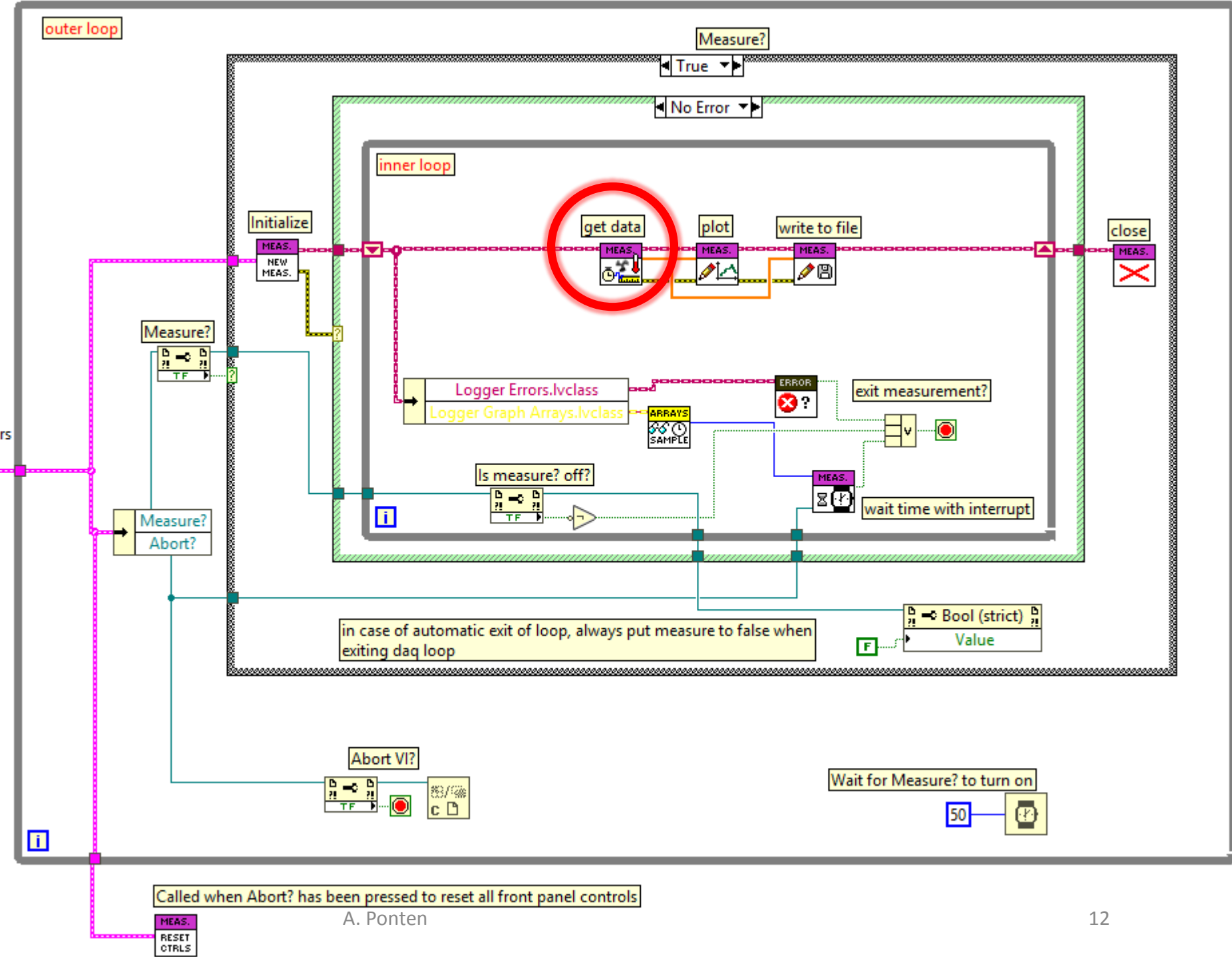
DAQ Loop



Structure:

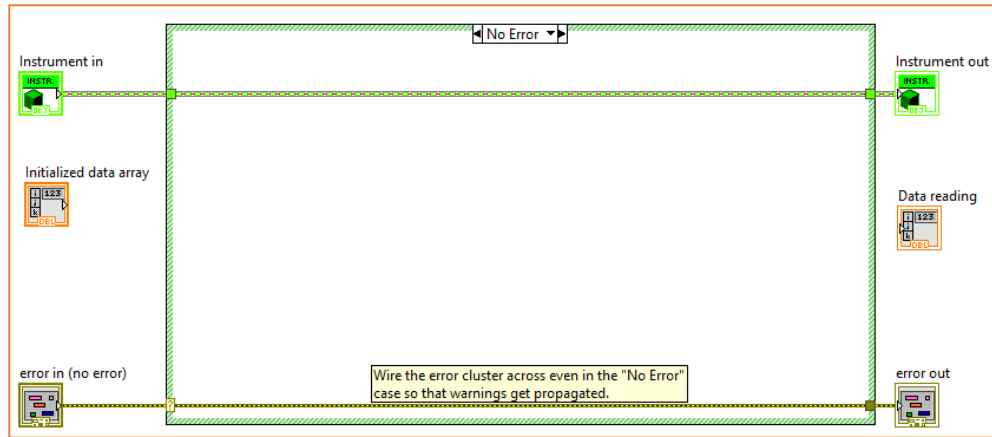
1. Measure?
2. If true, enter inner loop
3. Gather data, plot, write to file
4. Repeat 3. until exit measurement?
5. Go to 1.

This diagram the same for all devices! LabVIEW knows which instrument is being wired into the functions and dynamically calls the right version of Measure Data.vi



Overriding *Measure Data.vi* examples

Instrument parent class (empty)

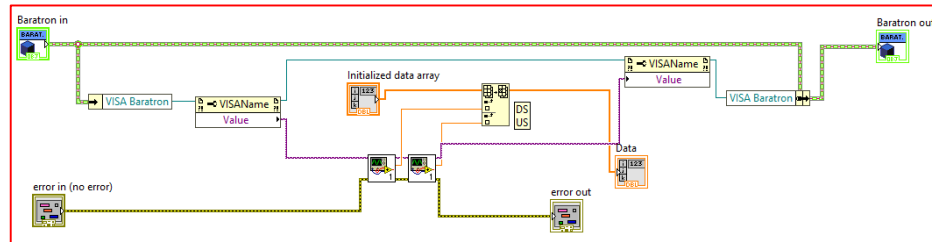


Override

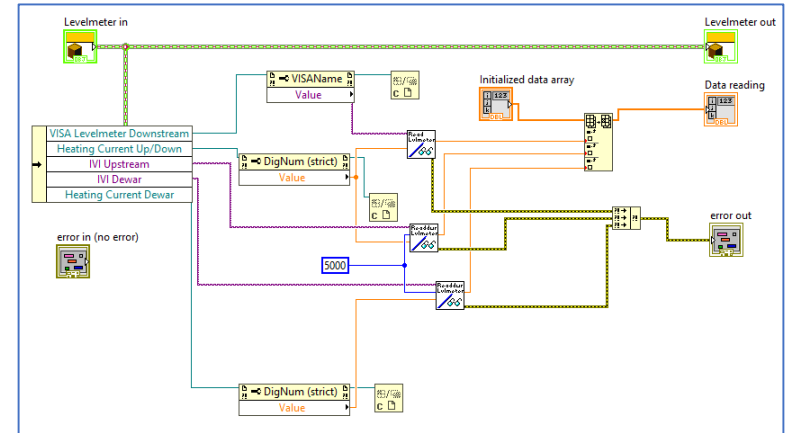
Override

Override

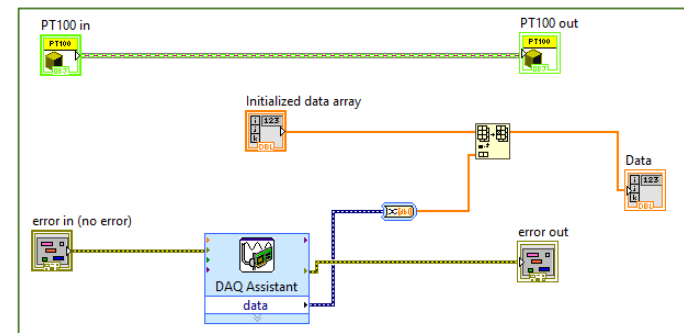
Baratron



Helium Levelmeter



Temperature Probe



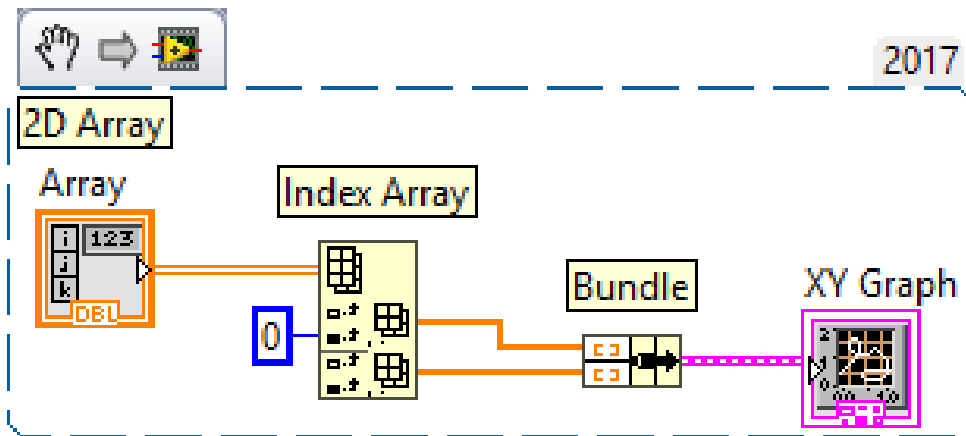
Results

- Adding new instruments is much quicker
- Troubleshooting is much quicker
- Can still customize every function if need be, due to inheritance

Backup

LabVIEW: A Graphical Programming Language

1. Easy to learn and use
 - Good for quickly setting up measurements etc.
2. Dataflow Language. Data is passed through a set of instructions.
 - Analogy: Electrical signals being sent through a circuit



Example of LabVIEW code to plot a 2D-array

Image: <https://knowledge.ni.com/KnowledgeArticleDetails?id=kA00Z0000004BQISA2>