LabVIEW

ISOTDAQ 2024

Adriaan Rijllart & Gary Boorman

20th –June – 2024

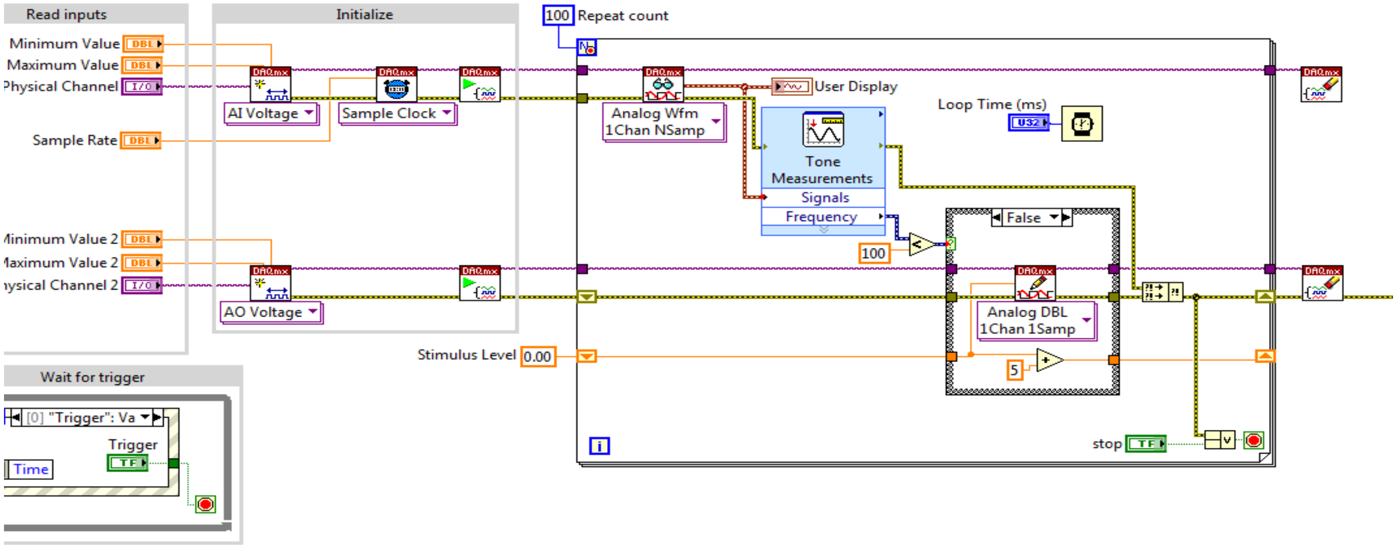
Agenda

- Introduction to LabVIEW
- Instrumentation and Data Acquisition
- Application Development
- LabVIEW for Accelerators and Detectors
- Other Research Applications

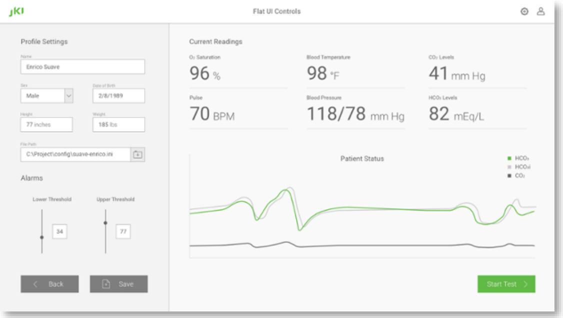
What is LabVIEW?

Application development

- Program as you think

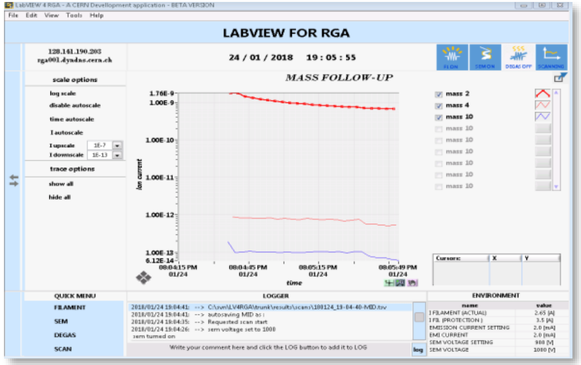


Graphical interface



Name: Enrico Suave
Sex: Male
Date of Birth: 2/8/1989
Height: 77 inches
Weight: 185 lbs
File Path: C:\Project\config\suave-enrico.ini
Lower Threshold: 34
Upper Threshold: 77

O₂ Saturation: 96 %
Blood Temperature: 98 °F
CO₂ Levels: 41 mm Hg
Pulse: 70 BPM
Blood Pressure: 118/78 mm Hg
HCO₃ Levels: 82 mEq/L



128.141.190.203
rga001.dyndns.cern.ch

24 / 01 / 2018   19 : 05 : 55

I upscale: 1E-7
I downscale: 1E-13

☑ mass 2
☑ mass 4
☑ mass 10
□ mass 10
□ mass 10
□ mass 10
□ mass 10
□ mass 10
□ mass 10

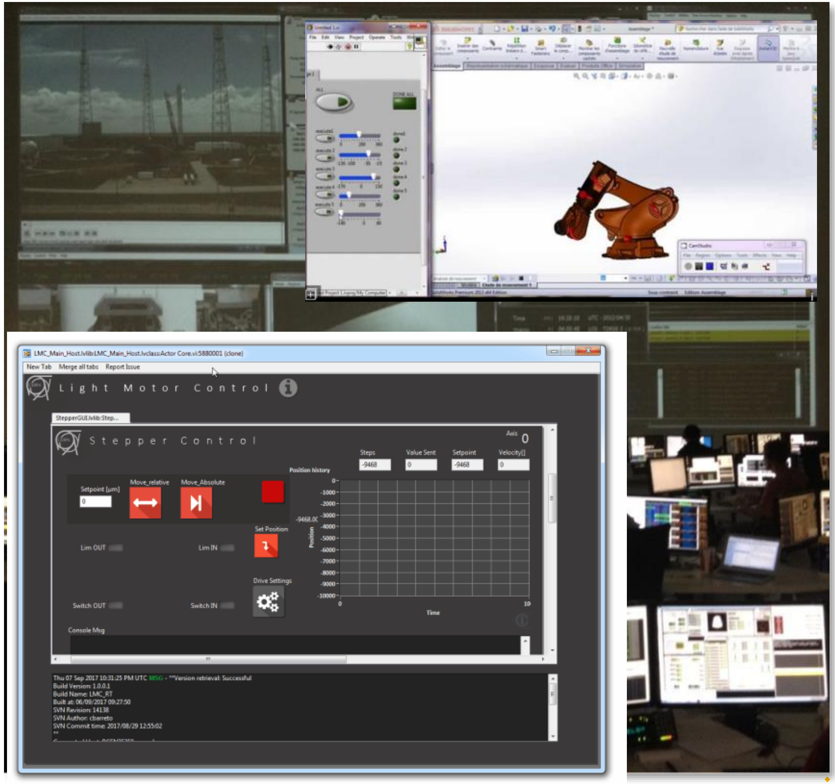
LOGGER

```
2018/01/24 19:04:41:  --> C:\svn\LV4RGA\trunk\results\scans\180124_19-04-40-MID.tsv
2018/01/24 19:04:41:  --> autosaving MID as :
2018/01/24 19:04:35:  --> Requested scan start
2018/01/24 19:04:26:  --> sem voltage set to 1000
 sem turned on
```

Write your comment here and click the LOG button to add it to LOG

ENVIRONMENT

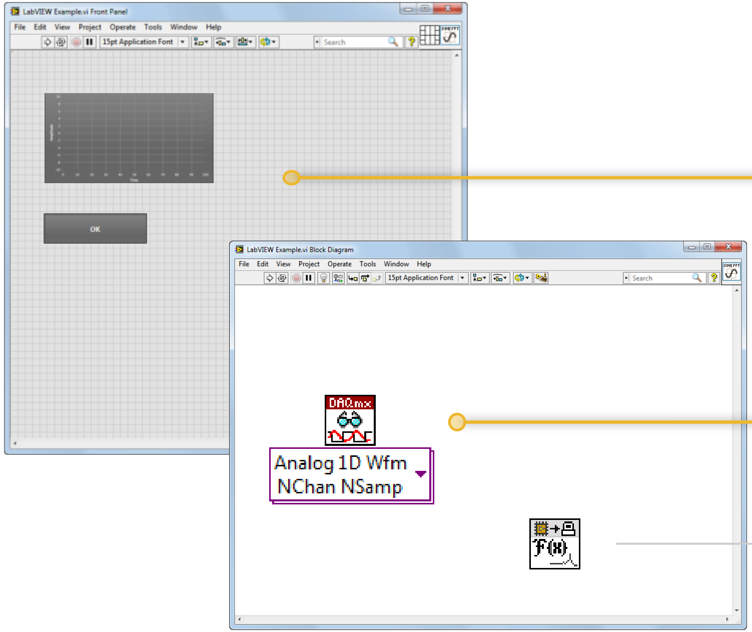
name	value
I FILAMENT (ACTUAL)	2.65 [A]
I FIL (PROTECTION)	3.5 [A]
EMISSION CURRENT SETTING	2.0 [mA]
EMI CURRENT	2.0 [mA]
SEM VOLTAGE SETTING	900 [V]
SEM VOLTAGE	1000 [V]



Axis 0
Steps: -9468
Value Sent: 0
Setpoint: -9468
Velocity[]: 0
Setpoint [µm]: 0
-9468.0C

```
Thu 07 Sep 2017 10:31:25 PM UTC MSG - **Version retrieval: Successful
Build Version: 1.0.0.1
Build Name: LMC_RT
Built at: 06/09/2017 09:27:50
SVN Revision: 14138
SVN Author: cbarreto
SVN Commit time: 2017/08/29 12:55:02
**
```

Application development


LabVIEW Example.vi Front Panel
File Edit View Project Operate Tools Window Help
15pt Application Font
Search
OK
LabVIEW Example.vi Block Diagram
File Edit View Project Operate Tools Window Help
15pt Application Font
Search
DAQmx
Analog 1D Wfm
NChan NSamp

LabVIEW Front Panel

The user interface of a VI

LabVIEW Block Diagram

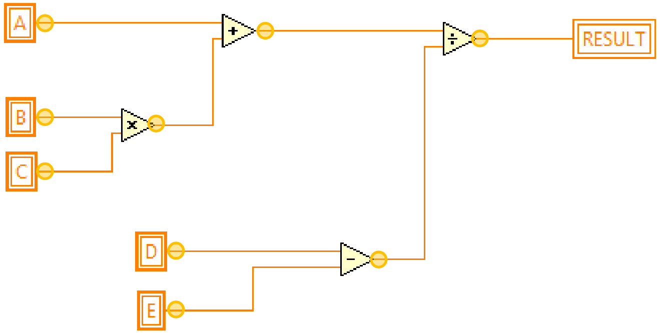
The source code of a VI

Functions:

Virtual
Instruments

Dataflow

- Data driven execution

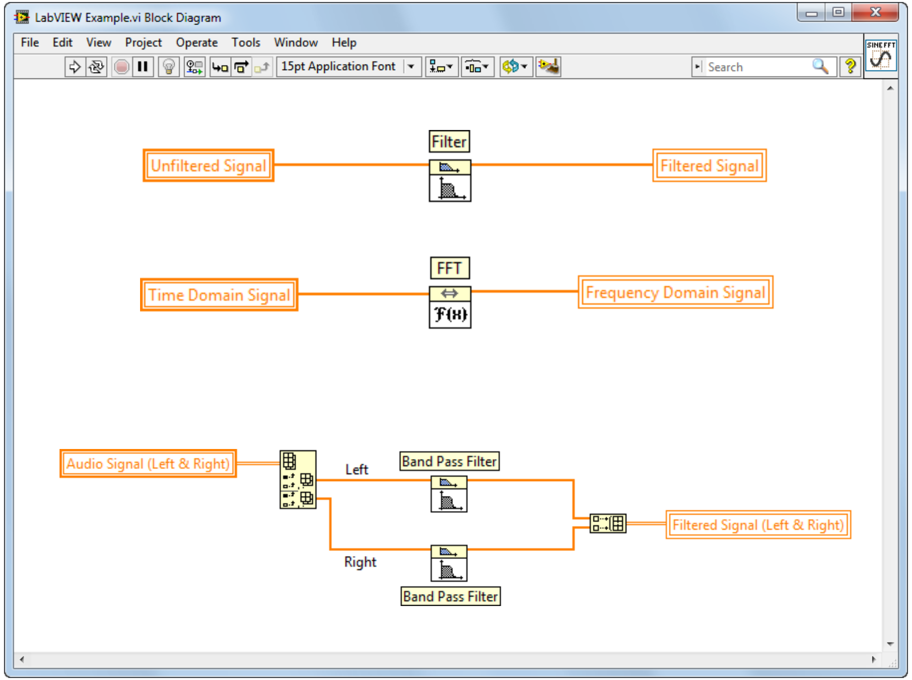


Intrinsic **Parallelism**

Intrinsic **Synchronisation**

Dataflow

- Data driven execution

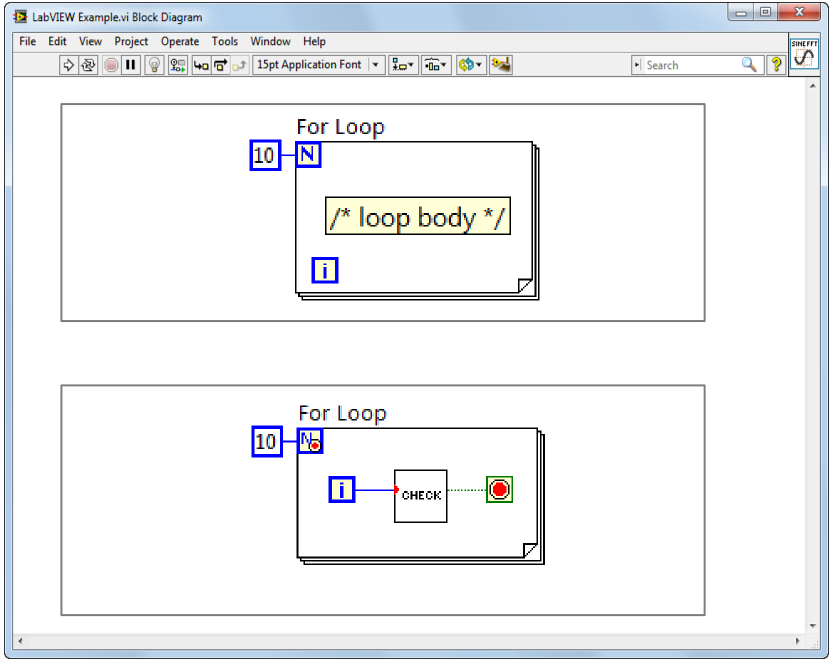

LabVIEW Example.vi Block Diagram

Intrinsic Parallelism

Comparison with text

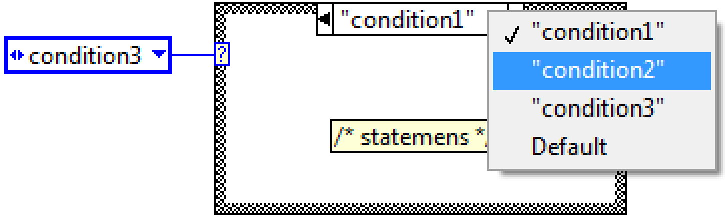
```
for (i = 0; i < 10; i++)
{
    /* loop body */
}
```

```
for (i = 0; i < 10; i++)
{
  if(check(i)) break;
}
```

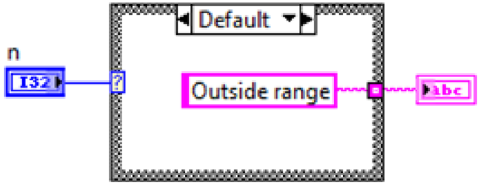

LabVIEW Example.vi Block Diagram
File Edit View Project Operate Tools Window Help
15pt Application Font
Search
For Loop
10
N
/* loop body */
i
For Loop
10
N
i
CHECK

Comparison with text

```
if condition1 then
  -- statements;
elseif condition2 then
  -- more statements
elseif condition3 then
  -- more statements;
else
  -- other statements;
end if
```

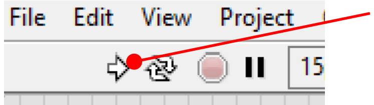


```
switch (n) {
  case 5:
    printf(“Small number.");
    break;
  case 100:
    printf(“Large number.");
    break;
default:
    printf(“Outside range”);
    break;
}
```



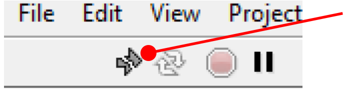
The LabVIEW Compiler I

- The LabVIEW environment continually parses the block diagram
  - ○ Valid code ->



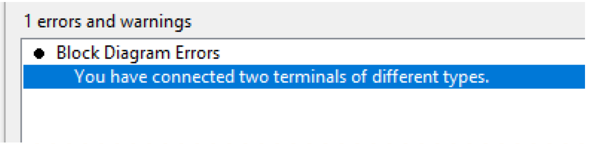
Solid RUN button

  - ○ Invalid/incomplete code ->



Broken RUN button

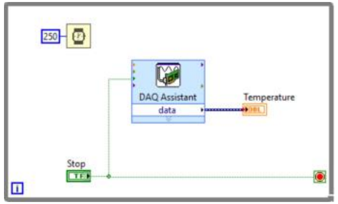
- If code is valid, clicking on the RUN button causes LabVIEW to compile the code and then execute it
- Click on a broken RUN button to get detailed information on the error



1 errors and warnings

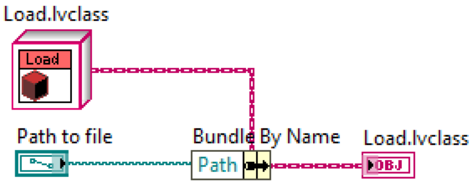
- Block Diagram Errors
  You have connected two terminals of different types.

The LabVIEW Compiler II

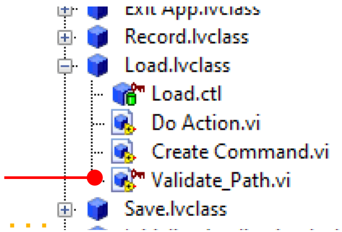


- When developing/debugging LabVIEW code it can be run and tested within the LabVIEW environment
- Once the code is working as desired it can be compiled into an executable (.exe etc), then launched like any other program
  - LabVIEW supports both 32 and 64-bit OS: Windows, Linux and MacOS
- Code can also compile into a windows library (.DLL) or Linux library (.SO)
  - Calls to DLL or SO require knowledge of the function prototypes - LabVIEW will generates the appropriate documentation
- LabVIEW can call functions within other DLL and SO libraries

LabVIEW OOP



- LabVIEW has object-oriented capabilities – encapsulation & inheritance
- But **BEWARE**
  - LabVIEW is a by-value language, including its objects
    - Most other OO environments use by-reference objects
  - All data is private
    - Explicit accessor methods must be used to access the data
- Methods are public by default but can be made private (called by class's methods only) or protected (called by child classes too)
- LabVIEW objects are supported on Desktop, RT and FPGA



Private method

DAQ & Instrumentation

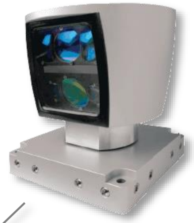
Measurement challenges

- Conflicting programming approaches
- Disparate drivers
- Timing, triggering, and synchronization
- Fixed soft/hardware
- Changing requirements
- …

temperature

storage

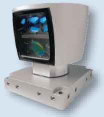
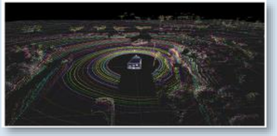
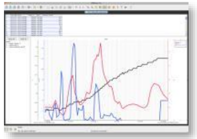
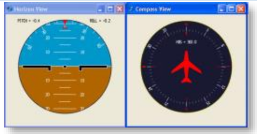
Heterogeneous systems



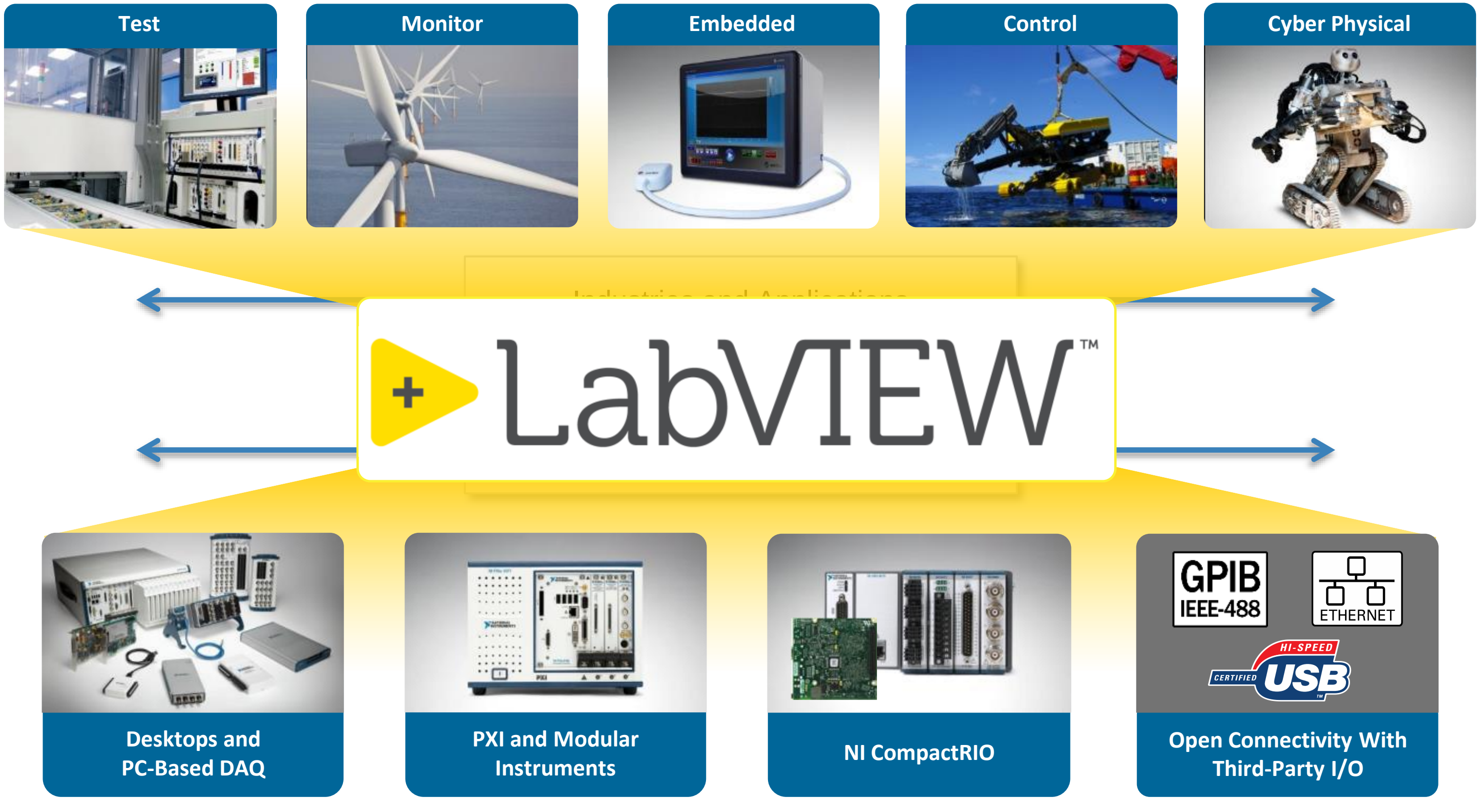
LiDAR

RGA

Measurement challenges

Sensor	Interface	Conditioning?	Software
		no	
		yes	
		yes	-
		no	

Heterogeneous systems



Test

Monitor

Embedded

Control

Cyber Physical

LabVIEW™

Desktops and PC-Based DAQ

PXI and Modular Instruments

NI CompactRIO

Open Connectivity With Third-Party I/O

Modular Instruments

Compact DAQ

PXI

PXI/PXIe modules

chassis

Compact RIO

DAQ Comparison

Software Used for Data Acquisition and Instrument Control

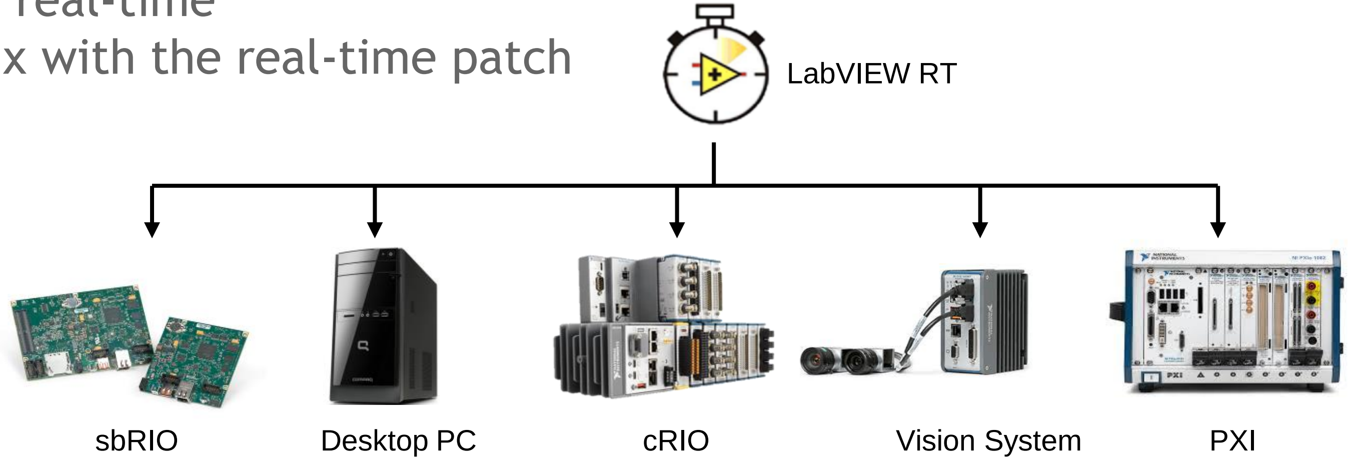
OPTIONS	C++/C#/JS/VB	LabVIEW	MATLAB	DASYLab
Ease of programming (novice)	Difficult	Easy	Medium	Easy
Programming Community size	Very large	Large	Large	Medium
Complex Applications	Yes	Yes	No	No
Built-in DAQ Support	No	Yes	Some	Yes
Built-in Analysis	No	Yes	Yes	Yes

Embedded Systems

Extending the LabVIEW environment

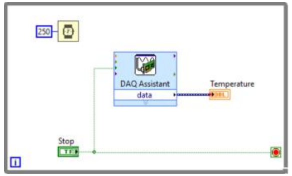
Real-time Systems

- Deterministic code operation
- Distributed control/test/acquisition systems
- LabVIEW real-time
  - Linux with the real-time patch



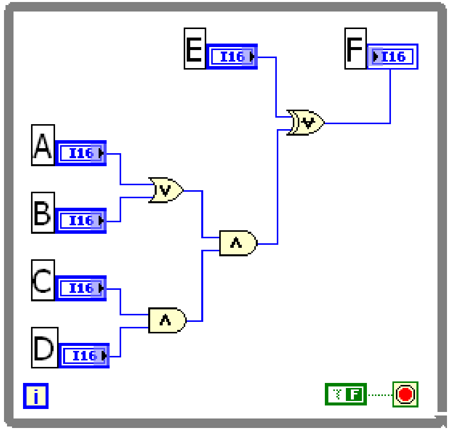
Compiling LabVIEW for RT Systems

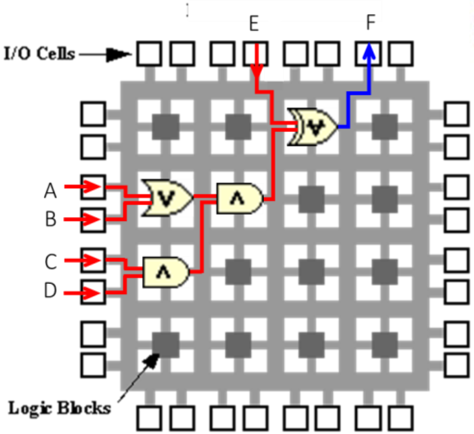
- LabVIEW can run RT code within the development environment
    - Code is executed on the RT system
    - User interface is on the desktop/development system
- Code can usually be run on different RT targets with only minimal changes (file paths, hardware interfaces etc)
- Once the code is running as expected, compile the code into an RT executable
    - Executable can be deployed on RT system
    - Executable starts running once the RT has powered up and loaded its operating system
    - Code is usually designed for running 24/7



LabVIEW on FPGA

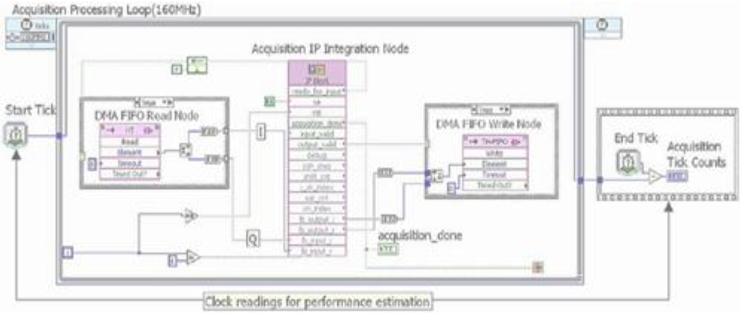
- LabVIEW FPGA



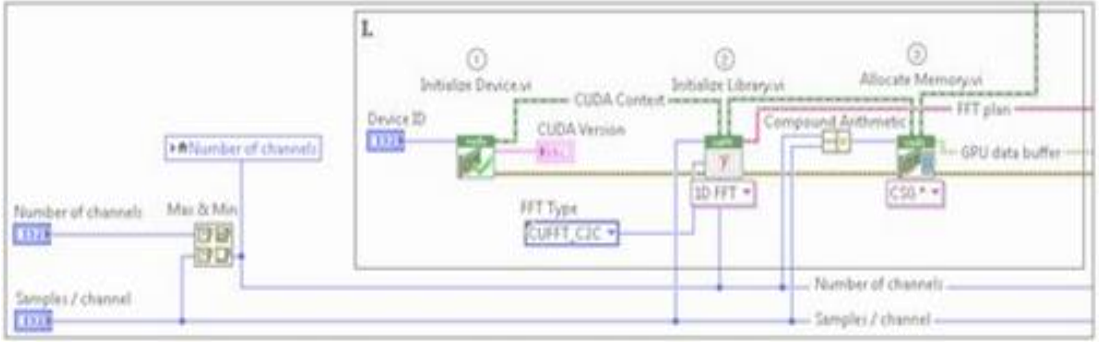


LabVIEW on FPGA

- Xilinx FPGA
  - IP integration
  - Vivado Export
- NVIDIA CUDA GPU



Acquisition Processing Loop(160MHz)
Acquisition IP Integration Node
Start Tick
DMA FIFO Read Node
DMA FIFO Write Node
End Tick
Acquisition Tick Counts
acquisition_done
Clock readings for performance estimation



Compiling LabVIEW for FPGA

- Many LabVIEW functions are available for FPGA
  - *Some exceptions*:
    - Unbound arrays, queues, strings
    - Double precision numbers (Single is permitted)
    - Non-homogeneous arrays of objects
- LabVIEW FPGA code needs to be compiled – automatically launches and uses the Xilinx Vivado environment. Can add existing VHDL IP
- The RT system accesses the FPGA using:
  - Front panel controls and indicators (latency of ms)
  - Direct memory access, DMA (very fast, up to GB/s depending on backplane)
  - Interrupts (latency of μs)

LabVIEW for Accelerators and Detectors

LabVIEW at CERN

550 LabVIEW Users

30+ Project clients

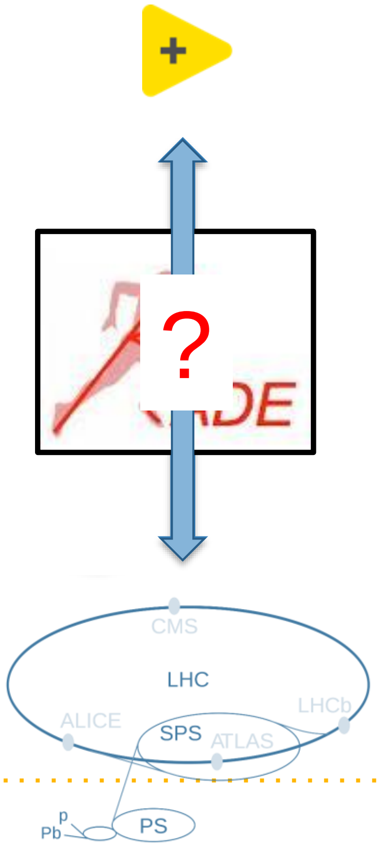
CERN LabVIEW

Support

LabVIEW™

Center of Excellence

The access challenge



Custom hardware

PXI

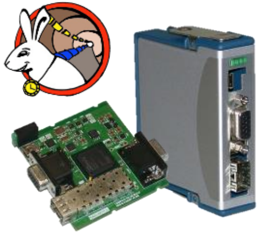
Accelerator timing
CTRP-PMC
(CERN)

PMC carrier
(Kontron)

Fine delay-FMC
(CERN)

FMC carrier
(INCAA)

cRIO

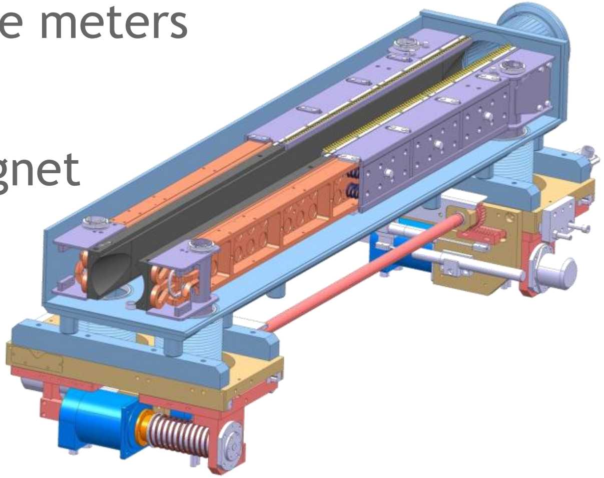


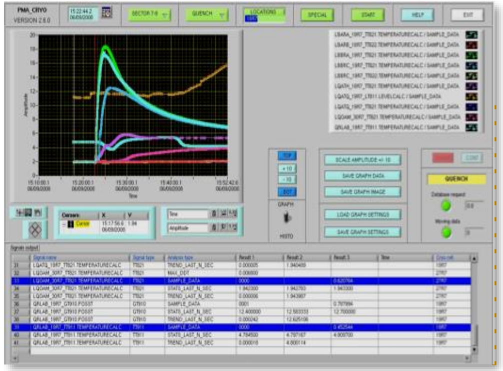
White rabbit timing (CERN)

Fibre-based triggering
(ANGARA Technology)

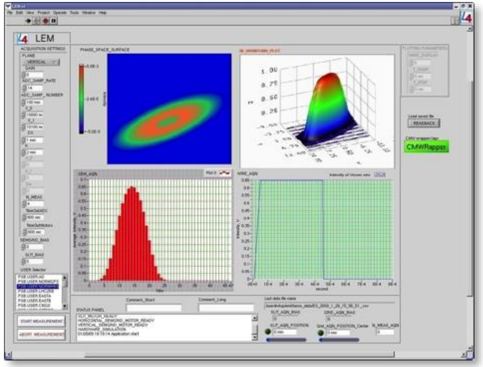
Example applications

- LHC collimators
- LINAC4 emittance meters
- Kicker Magnets
- Experiments magnet protection
- AWAKE
- ISOLDE
- CLIC RF test
- …





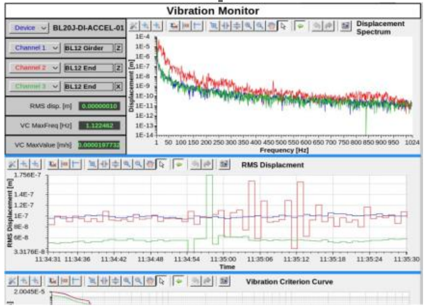
Post-Mortem analysis



Linac4

LabVIEW and Middleware

- EPICS support built-in
  - Create EPICS IOCs to run (usually) on Embedded systems
  - Create EPICS Clients on both Embedded and Desktop systems
  - Several third-party solutions that improve performance or the scope of data-types (LNLS, ANL etc)
- CMW (Controls Middleware) at CERN
  - The MTA group has created RADE
    - Embedded systems running LabVIEW can read/ write to the standard CERN tools/databases
- TANGO
  - Third-party support from some European and US labs



Vibration Monitor

Device BL20J-DI-ACCEL-01

Channel 1 BL12 Girder Z

Channel 2 BL12 End Z

Channel 3 BL12 End X

RMS disp. [m] 0.00000010

VC MaxFreq [Hz] 1.122462

VC MaxValue [m/s] 0.0000197732

Other Applications

LabVIEW Web Module

- Compile LabVIEW and run within web-page (Javascript)
- View compiled code on any device
- Try *www.webvi.io*

Support for emerging technology

- Extensive HW and SW support of RF
  - Vector Signal Transceiver (VST) with accessible FPGA
  - 6G research and metrology (1 Tbps)
- Autonomous vehicles
- Industrial Internet of Things (IIoT)

Thank you

Credits

- NI (now part of Emerson)
- CERN BE-CEM group

BE-CEM