Activities　　euRun　　　　　　　　　　　　Mär 20 17:08

EUDAQ Log Collector (on fhlrcduranta)

Level: 0-DEBUG　From: All　Search:

Received	Sent	Level	Text
17:06:06.718	17:06:06.716	4-INFO	TLU VERBOSITY SET TO: 0
17:06:06.718	17:06:06.716	4-INFO	TLU DELAY START SET TO: 0 ms
17:06:06.722	17:06:06.722	4-INFO	DataCollector.c
17:06:06.722	17:06:06.722	4-INFO	TRIGGER VETO
17:06:06.790	17:06:06.790	4-INFO	Producer.aida_m
17:06:16.193	17:06:16.193	4-INFO	Producer.ni_mi
17:06:17.476	17:06:17.476	4-INFO	Processing Sta
17:06:17.577	17:06:17.577	4-INFO	RUN #6335 is
17:06:17.577	17:06:17.577	4-INFO	RUN #6335 is
17:06:18.677	17:06:18.677	4-INFO	RUN #6335 is
17:06:18.679	17:06:18.679	4-INFO	DataReceiver
17:06:18.680	17:06:18.680	4-INFO	RUN #6335 is
17:06:19.778	17:06:19.777	4-INFO	RUN #6335 is
17:06:19.781	17:06:19.781	4-INFO	DataReceiver
17:06:19.781	17:06:19.780	4-INFO	RUN #6335 is
17:06:20.883	17:06:20.878	4-INFO	DataReceiver
17:06:20.883	17:06:20.882	4-INFO	RUN #6335 is
17:06:20.883	17:06:20.882	4-INFO	TLU START con
17:06:20.885	17:06:20.885	4-INFO	TLU SET TO RU
17:06:20.886	17:06:20.886	4-INFO	TRIGGER VETO
17:08:01.770	17:08:01.770	4-INFO	increase high

Data Collector Trigger (on fhlrcduranta)

[DataCollector one_dc] [2024-03-20 17:06:18.677] (INFO) [
5 is to be started...
[DataCollector one_dc] [2024-03-20 17:06:18.680] (INFO) [
5 is started.
[DataCollector one_dc] [2024-03-20 17:06:19.780] (INFO) [
iver: Connection from <ConnectionTCP
<FID>tcp://127.0.0.1:38060</FID>

NI/Mimosa Producer (on fhlrcduranta)

```
<Configuration>
  <Section title="">
    <DataCollector.one_dc>tcp://127.0.0.1:42501</DataCollector.one_dc>
    <LogCollector.log>tcp://127.0.0.1:43555</LogCollector.log>
    <Monitor.StdEventMonitor>tcp://127.0.0.1:45249</Monitor.StdEventMonitor>
    <Name>/home/teleuser/ship202403/all_standard-mode-one-DC.conf</Name>
  </Section>
  <Section title="Producer.ni_mimosa">
    <DISABLE_PRINT>1</DISABLE_PRINT>
    <EUDAQ_DC>one_dc</EUDAQ_DC>
  </Section>
</Configuration>
```

ucer ni_mimosa] [2024-03-20 17:06:06.670] (INFO) [unknown sender]Producer.n
osa is to be configured...
ducer ni_mimosa] [2024-03-20 17:06:16.193] (INFO) [unknown sender]Producer.n
osa is configured.
ucer ni_mimosa] [2024-03-20 17:06:19.777] (INFO) [unknown sender]RUN #6335
o be started...
ducer ni_mimosa] [2024-03-20 17:06:19.780] (INFO) [unknown sender]RUN #6335
tarted.
ucer ni_mimosa] [2024-03-20 17:08:01.770] (INFO) [unknown sender]increase h
7bits of trigger number, last_tg_l15(32767) tg_l15(0)

MIMOSA26
　Sensor 0
　Sensor 1

AidaTluProducer (on fhlrcduranta)

[Producer aida_tlu] [2024-03-20 17:06:06.716] (INFO) [unknown sender]TLU VERBOSI
TY SET TO: 0
[Producer aida_tlu] [2024-03-20 17:06:06.717] (INFO) [unknown sender]TLU DELAY S
TART SET TO: 0 ms
[Producer aida_tlu] [2024-03-20 17:06:06.722] (INFO) [unknown sender]TRIGGER VET
0: triggers are disabled
　Clock output on LEMO is disabled
　TRIGGER PATTERN (for external triggers) SET TO 0x0 --- 0x8 (Two 32-bit words)
　TRIGGER POLARITY (for external triggers) SET TO 0x3f
　Required internal trigger frequency: 0 Hz
[Producer aida_tlu] [2024-03-20 17:06:06.790] (INFO) [unknown sender]Producer.ai
da_tlu is configured.
[Producer aida_tlu] [2024-03-20 17:06:20.878] (INFO) [unknown sender]RUN #6335 i
s to be started...
[Producer aida_tlu] [2024-03-20 17:06:20.882] (INFO) [unknown sender]TLU START c
ommand received
[Producer aida_tlu] [2024-03-20 17:06:20.883] (INFO) [unknown sender]RUN #6335 i
s started.
[Producer aida_tlu] [2024-03-20 17:06:20.885] (INFO) [unknown sender]TLU SET TO
RUN
[Producer aida_tlu] [2024-03-20 17:06:20.886] (INFO) [unknown sender]TRIGGER VET

Online Monitor (on fhlrcduranta)

ll/lib/libXMLParser_rdict.pcm

Initialising MonitorPerformance Collection
Initialising ParaMonitor Collection
OnlineMon Configuration

General Settings
Configuration File　　　: ..
Configuration File　　　: ../snapshots/
SnapShot Directory　　　: ../snapshots/
SnapShot Format　　　　 : .pdf

Correlation Settings
MinClusterSize　　　　　: 1
Planes to skip　　　　　:
Clusterizer Settings
HotPixelFinder Settings
HotPixelCut　　　　　　 : 0.01

Mimosa26 Settings
Mimosa26_max_sections　　　: 4
Mimosa26_section_boundary : 288

Configure: /home/teleuser/ship202403/all_standard-mode-one-DC.conf

eudaq Run Control v2.5.2 (on fhlrcduranta)

State:

Current State: Running

Control

|			
Init file:	/home/teleuser/ship202403/telescope.ini	Load	Init
Config file:	/home/teleuser/ship202403/all_standard-mode_one-DC.conf	Load	Config
Next RunN:		Start	Stop
	0%	Reset	Terminate
Log:		Log	☑ LogConfigs
ScanFile	/home/teleuser/hgtd102023/config_Feb2024/HGTD_batch1700.scan	Load	Start Scan

Run Number:　6335　　　　　　　　　　　　　　　　one_dc:DataCollector:　　49698 Events
aida_tlu:Producer:　49899 Events　　　　　　　　　　ni_mimosa:Producer:　　49899 Events
StdEventMonitor:Monitor:　4991 Events

Connections

type	name	state	connection	message	information
LogCollector	log	RUNNING	tcp://127.0.0....	Started	<_SERVER> tcp://43555
DataCollector	one_dc	RUNNING	tcp://127.0.0....	Started	<EventN> 49698　<MonitorEventN> 4969.000000　<_SERVER> tcp://42501
Producer	aida_tlu	RUNNING	tcp://127.0.0....	Started	<EventN> 49899　<Freq. (avg.) [kHz]> 0.327206　<IDTrig> 49902　<Particles> 59379　<Run duration [s]> 152.509202　<Scaler> 810341:121267:0:0:0:0
Producer	ni_mimosa	RUNNING	tcp://127.0.0....	Started	<EventN> 49899
Monitor	StdEventMon...	RUNNING	tcp://127.0.0....	Started	<EventN> 4991　<_SERVER> tcp://45249
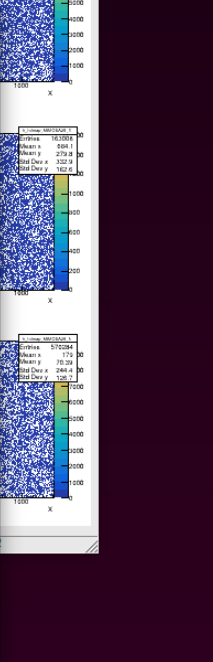
Constellation

Autonomous Control and Data Acquisition System for Dynamic Experimental Setups

Stephan Lachnit, DESY
Simon Spannagel, DESY

12th BTTB Workshop
Edinburgh, UK
2024-04-15

What do we expect from a "flexible DAQ system"?

- **Useful to control single laboratory setup**
  (e.g. radioactive source measurement)

- **Possibility to integrate multiple setups**
  (Detector DAQ, TCT laser control)

- **Lab supervision mode**
  (multiple setups monitored but control not ceded)

- **Synchronized operations**
  (test beam environment, coordinated start/stop, central control)

- **Scalability for small experiments**
  (many detectors, multiple data endpoints & monitors)
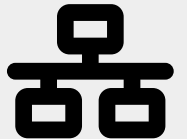
Introducing Constellation (in 30 sec)

- Project goals:

  - **Easy** to use, easy & fast to integrate new systems

  - **Stable** operation, reliable error handling

  - **Flexible** and applicable for many use cases

- Solid foundation: well-defined communication protocols between components

- Participants are called **satellites** (eudaq: Producers/Collectors)

  - Operation is governed by a **finite state machine**

  - Satellites can operate **autonomously** without active user interface

Network Discovery and Resiliance

- Current testbeam software often involves

  - assigning **fixed IP adresses & ports**

  - a **central control software** (single point of failure)

- Industry has long moved to more flexible and reliable systems, like

  - automatic **service discovery** in local network

  - **autonomous operation** with stateless control interfaces (REST)

- Constellation features both as **core design principle**

Live Demo: Network Discovery & Logging

- Starting a **logger** on Stephan's laptop that listens for all satellites

- Starting a 1st **satellite** (also on Stephan's Laptop)
  - Logger **discoveres** the satellite
  - Logger **starts listening** to messages from the satellite

- Starting a 2nd satellite **on Simon's laptop** (on the same network)
  - Has **dynamically assigned** IP address and port
  - Logger discoveres 2nd satellite
 - Logger starts listening to messages from Simon's satellite

\

Controlling the Finite State Machine of a Satellite

S. Lachnit & S. Spannagel - 12th BTTB - Constellation   15/04/2024
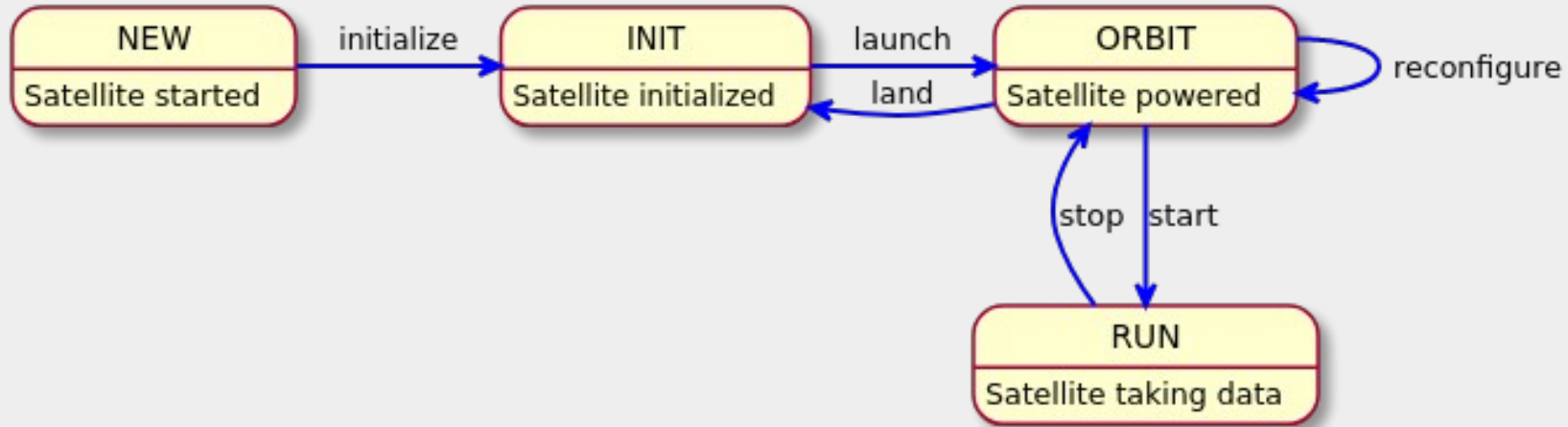
Live Demo: Controlling a Satellite

- Satellite starts in **NEW** state

- Sending **initialize** command
  → Satellite transitions to **INIT** state

- Sending **launch** command
  → Satellite transitions to **ORBIT** state

- Sending **launch** command **again**
  → Get response that this is an **invalid command** for the current state

- Sending **land** command
  → Satellite transitions to **INIT** state

\<live demo /\>
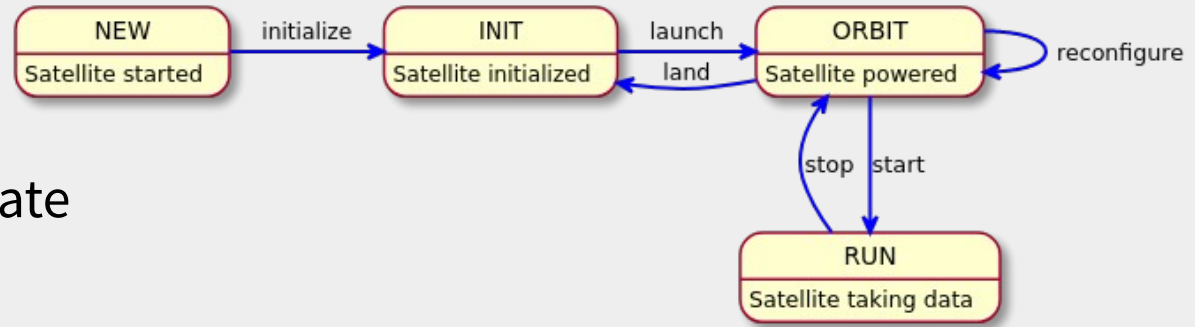
How to implement a Satellite

- C++ example

```cpp
18
19    void bttb12::starting(std::uint32_t run_number) {
20        frame_number_ = 0;
21        LOG(STATUS) << "Starting run " << run_number;
22    }
23
24    void bttb12::running(const std::stop_token& stop_token) {
25        while(!stop_token.stop_requested()) {
26            auto data = getDataFromDetector();
27            LOG(STATUS) << "Got frame " << frame_number_;
28            sendData(data);
29            frame_number_++;
30        }
31    }
```

Let's log something

- Note: separate Python implementation also available

Live Demo: Starting a Run and Disconnect

- Sending **initialize**, **launch** and **start** commands to start a run

- Satellite calls **user function**
  - Logs frame number while running

- **Disconnect** controller
  - Satellite **continues running**

- **Reconnect** controller
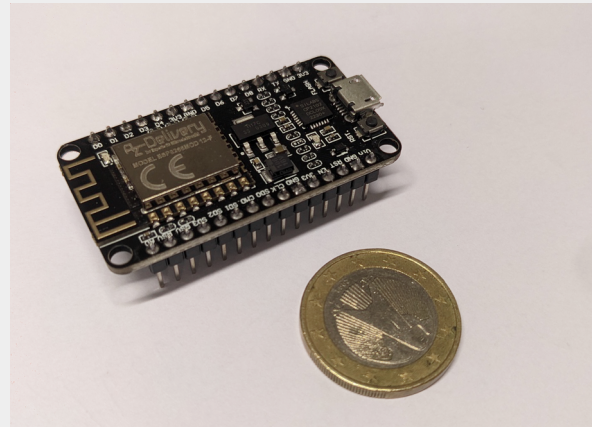 - Get state of satellite to see it is still running

Live Demo: MicroSat

...well-defined protocols means independence from implementation

- Let's implement Constellation on a 5 EUR, low power **ESP8266 microcontroller**

- Possible applications:

  - Adding an RS232 interface and controlling a Keithley

  - Providing temperature & humidity data

  - ...

Where are we now?

- Implementing a new control & DAQ system bottom-up

- Many core concepts and features already implemented

  - Network discovery

  - Controlling & Logging over the network

  - Interface for Satellite implementation ready!

- Documentation in the process of being written, check it out at
    constellation.pages.desy.de

- All code is open source and available at
    gitlab.desy.de/constellation

What's next?

- Implementation of last core components underway

- Next big item up: (graphical) user interfaces!

- We are holding a (2nd) Hackathon in May in Hamburg

  - Working towards User Interfaces

  - Continue writing user & developer documentation

- Do you have suggestions?

  - Chat with us!

  - Send us a mail!

  - Open a ticket on GitLab!