



State of the Celestial Sphere

Ongoing Developments in Constellation

Simon Spannagel, DESY
EDDA Project Meeting
05/02/2024

Overview

- Many things developed over past months
- Most infrastructure things solved, repositories on DESY Gitlab in action
- CI set up and running
- Implementation for micro controllers (ESPx)
- Stephan is back! – working @ DESY since last week

& implement!

Satellite Configuration

Satellite Module

- load
- unload
- reload
- launch
- land
- reconfigure
-

UI Concepts

Hanno
FSM

with::
Satellite

Commander
(aka Controller)

Listener

Log Service

Heartbeat Service

HeartbeatCheck Service

Data Sender

Data Receiver

CHIRP Service

Message

Log Message

Command Message

Heartbeat Message

Data Message

Stephan [Hanno]

CHIRP Message

documentation of state

Separate C++ & Python implementations of full stack?

TESTS!
CI!
HOSTING?

2N god:
current knowledge is documented

@Hathiakhor

Chin-Lia

Leue Kristian

Anniko

Low-Level

Repositories @ DESY GitLab

The screenshot displays the GitLab interface for the 'Constellation' repository. The top section shows the repository name, a search bar, and navigation links for 'Subgroups and projects' and 'Shared projects'. Below this, a list of subgroups and projects is shown, including 'C++ snippets', 'CHIRP Draft', 'EDDA Hackathon 1', 'MicroSat', 'spdlog_test' (highlighted), 'Constellation', 'Build Images', and 'Website'.

The main section shows the repository details for 'Constellation', including the number of commits (358), branches (3), tags (0), and project storage (3.1 GiB). It also displays badges for 'Latest Release' (none), 'License' (EUPL 1.2), 'REUSE' (compliant), 'coverage' (98.40%), and 'pipeline' (passed).

Below the repository details, a list of issues is shown, filtered by 'Open' status. The issues listed are:

- Logging: simplify interface** (#25) - created 2 days ago by Simon Spannagel. Status: C++ core. Updated 1 day ago.
- Notes on Documentation** (#22) - created 2 weeks ago by Simon Spannagel. Status: docs manual. 0 comments.
- Follow-up from "Add first CMDP Implementation": Logs start appearing late on CMDP interface** (#21) - created 2 weeks ago by Simon Spannagel. Status: C++ bug. 1 of 1 checklist item completed. Updated 1 week ago.
- Allow 2D Arrays in Metrics** (#19) - created 1 month ago by Simon Spannagel. Status: protocols. 0 of 1 checklist item completed. 0 comments.

Continuous Integration

Pipeline Needs Jobs 19 Tests 25

Group jobs by Stage Job dependencies

Test coverage: 98%

build

- build:alma9-cern
- build:debian-clang
- build:debian-gcc
- build:fedora-latest
- build:ubuntu-22.04

test

- test:alma9-cern
- test:debian-clang
- test:debian-gcc
- test:fedora-latest
- test:ubuntu-22.04

lint

- lint:clang-scan-build
- lint:clang-tidy
- lint:coverage
- lint:coverity-scan

format

- format:clang-format
- format:codespell
- format:reuse

docs

- docs:html

Reference generation

Supported OS:

- Alma Linux 9
- Debian Testing
- Fedora 39
- Ubuntu 22.04

Static code analysis:

- Coverity
- Clang Tidy
- Clang Scan-Build

Form & style:

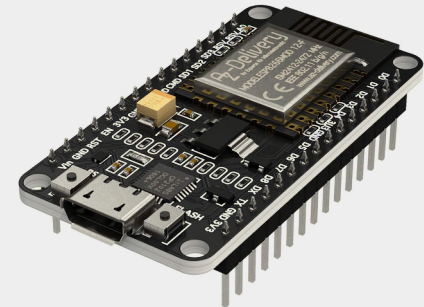
- Code formatting
- Spelling / typos
- Licensing checks

MicroSat – Constellation on Microcontrollers

- Had some ESP8266 lying around
 - Use them regularly for temperature/humidity monitoring
 - Thought: that would be cool to feed into Constellation
- Issue: no ZMQ implementation for microcontrollers (too heavy!)
Solution: implement our own minimalist implementation
- Now have minimal **ZMTP** + pub-only **CMDP** + offer-only **CHIRP**!
fully compliant with Constellation protocols



~3x6 cm
~5 EUR



MicroSat in Action

```
void setup() {  
    // hostname of this machine, optionally a NTP  
    CMDPPublisher::init(hostname, &timeClient);  
    // Starting the publisher  
    CMDPPublisher::begin();  
}
```

```
void loop() {  
    // Needs to be called on every loop iteration to process new clients and subscriptions/unsubscriptions:  
    CMDPPublisher::update();  
  
    // Logging a message:  
    CMDPLOG(INFO, "This is a log message that will be distributed via CMDP");  
  
    // Optionally, a logger topic can be set:  
    CMDPLOG(WARNING, "SENSOR", "Something is up with the sensor reading");  
  
    // For TRACE log levels, code location information is automatically attached  
    CMDPLOG(TRACE, "There is a bug at this location that needs fixing");  
}
```

```
[INFO] Connected to WiFi network MYWIFISSID  
[INFO] To connect to CMDP, send a CHIRP request beacon to obtain the endpoint  
[INFO] Alternatively, connect directly to 192.168.179.12 : 53080  
[DEBUG] CHIRP: Preparing to send CHIRP message of type 2 for service 3  
...  
[DEBUG] ZMTP: Received connection from client at 192.168.179.3 : 39432  
[DEBUG] ZMTP: Verified ZMTP signature match  
[DEBUG] ZMTP: Verified ZMTP version match  
[DEBUG] ZMTP: Verified that remote peer is SUB socket, accepting.  
[INFO] CMDP: Accepted new connection from client 0 at 192.168.179.3 : 39432  
[INFO] CMDP: Client 0 subscribed to topic: LOG/WARNING
```


Next steps (for me)

- Pick up on writing conceptual design documentation
- Documentation will be divided into 4 parts:
 - Tutorials: teach how to use Constellation
 - Concepts: write up thoughts behind the framework structure
 - How-To Guides: concise answers how to achieve a specific goal
 - Reference: detailed documentation one can consult to get info on inner workings, parameters, etc
- See <https://diataxis.fr/> for some theoretical background on this

Things we could tackle already

- Chinchia presented some tools & ideas for GUI design
- We should identify tools / frameworks in which we would like to implement Uis
 - Python: Textual (<https://textual.textualize.io/>)?
 - C++: Qt6? Azul? FTXUI?
- Check out and add your ideas to the issue:
<https://gitlab.desy.de/constellation/constellation/-/issues/13>
- Start drawing some ideas so we can iterate?

