LibreCube Activities
Review and Outlook

Artur Scholz
LibreCube Initiative
Open Source Space and Earth Exploration

Open Source Everything

Space Standards

Reference Architecture
Build your Mission

LibreCube Sat
Build your Mission

LibreCubeSat

LibreCubeDrone

LibreCubeRover
Stuff done
SLE User
Space Link Extension User in Python

https://gitlab.com/librecube/lib/python-sle-user
Milenko Starcik, Fabian Burger, Artur Scholz
Slide 7

SLE User

Ground Communications Network Using SLE

Forward Link

Return Link

SLE Provider

SLE User

Data Archive

T&C Processing

Ground Networks

SLE User

Image Processing
SLE User
import sle

raf = sle.RafUser(
    service_instance_identifier=os.environ.get('RAF_INST_ID'),
    responder_ip=os.environ.get('SLE_PROVIDER_HOSTNAME'),
    responder_port=int(os.environ.get('SLE_PROVIDER_TM_PORT')),
    auth_level='bind',
    local_identifier=os.environ.get('INITIATOR_ID'),
    peer_identifier=os.environ.get('RESPONDER_ID'),
    local_password=os.environ.get('PASSWORD'),
    peer_password=os.environ.get('PEER_PASSWORD')
)

raf.bind()
raf.start()
time.sleep(5)
raf.stop()
raf.unbind(reason='other')
Linkpredict
Generic and Modular Link Budget in Python

https://gitlab.com/librecube/lib/linkpredict
Jona Saffer, Artur Scholz
```
import linkpredict as lp

channel = lp.Channel(...)
# ...
link = lp.Link(
    channel=channel,
    geometry=geometry,
    transmitter=transmitter,
    transmit_antenna=transmit_antenna,
    receive_antenna=receive_antenna,
    receive_antenna_noise=receive_antenna_noise,
    receiver=receiver,
    medium_losses=medium_losses,
)
result = link.calculate_link_budget()
```
Transmitter (on satellite)

```python
onboard_losses = lp.Device(gain=1)
amplifier_power = 9 # @ dBW = 1 Watt
transmitter = lp.Transmitter(amplifier_power, [onboard_losses])
tx_antenna = lp.OmniDirectionalAntenna(gain=0, linear_polarized=True)
```

Geometry

```python
spacecraft = lp.SpacecraftObject()
nname = "Cubesat-1 (BSUSAT-1)"
line1 = "1 430608J 130933E 13314.15998747 0.00001095 0.00000-0 50587-4 0 9994"
line2 = "2 430608 97.5398 334.9753 0013880 232.5138 215.5169 15.17119642 1849"
spacecraft.set_orbit_from_lto([name, line1, line2])
groundstation = lp.GroundStationObject()
lon, lat, alt = 59.759, 6.216, 275
groundstation.set_location(lat, lon, alt)
groundstation.geometry = lp.GroundStationSpacecraftGeometry(groundstation, spacecraft)
```

Path

```python
atmospheric_loss = lp.SimpleMediumLoss()
medium_losses = [atmospheric_loss]
```

Channel

```python
modulation = lp.FSKNonCoherentNoCoding(bit_rate=9600)
channel = lp.Channel(430.2e6, modulation=modulation)
```

Receiver (groundstation)

```python
line_losses = lp.Device(gain=2.0)
receiver = lp.Receiver.from_noise_figure(noise_figure=2.0, devices=[line_losses])
rx_antenna = lp.MainLobeAntenna(peak_gain=15.5, beam_3db_width=5)
rx_antenna_noise = lp.SimpleAntennaNoise(300)
```
PLUTO to Python
A Parser of ECSS PLUTO scripts

https://gitlab.com/librecube/prototypes/python-pluto
Vidit Jain, Christoph Buchner, Artur Scholz
procedure

preconditions
  wait until value of GyroTemp > 60 degC
end preconditions

main
  initiate and confirm SwitchOnGyro;
end main

confirmation
  wait until value of Gyro5 < 0.2 deg/h
end confirmation

end procedure
from pluto import *
from model import *

class Procedure_test0921(Procedure):
    def preconditions(self):
        if self.wait_until_expression(
            lambda: GyroTemp.get_value() > ureg('60degC')) is False:
            return False

    def main(self):
        act = ActivityCall(self, SwitchOnGyro)
        if self.initiate_and_confirm_activity(act) is False:
            return False

    def confirmation(self):
        if self.wait_until_expression(
            lambda: Gyro5.get_value() < ureg('0.2deg')/h) is False:
            return False
SpaceCAN
Robust and Reliable System Bus

https://gitlab.com/librecube/lib/micropython-spacecan
https://gitlab.com/librecube/lib/cpp-spacecan
Peter Mader, Jesper Schlegel, Nicolas Wolf, Artur Scholz
SpaceCAN

Telecommand

Telemetry
SpaceCAN

CAN Node
- Microcontroller
- CAN Controller
- CAN Transceiver

ISO 11898-2 Network

Stub Length
CAN Node
- Not Terminated At Node

Tx
5V
0V

Can_H

2.5V
d
r
d

V_{off} = 2V

Can_L

0V
2V

Rx
Services

Redundancy management
Synchronization
Time distribution
TC/TM exchange
(Message exchange)
SpaceCAN
Stuff ahead
CDFP in Python
CCSDS File Delivery Protocol
CCSDS File Delivery Protocol
CCSDS File Delivery Protocol
CCSDS File Delivery Protocol
CCSDS File Delivery Protocol
1U Structure
Modular, Extensible 1U CubeSat Structure
1U Structure
1U Structure
1U PCDU
Basic and Robust Power Supply System
Power Control and Distribution Unit
Get on Board
Communication Channels

LibreCube - Open Source Space and Earth Exploration

At LibreCube our vision is to enable everyone to get involved in building systems for exploration of near and remote locations using open source hardware and software. We believe that discovering new worlds and getting scientific insights should be a matter to all humankind.

LibreCube provides the means to it. Be part of it!

Open Source

Everything we do at LibreCube is made available to the public as free and open source. We only use free and open source tools – this way, everybody can get involved!

Free and Open Standards

We rely on proven and tested standards for our system designs, with reference to standards from the space domain.

Reference Architecture

Defining a generic architecture of system of systems that have standardized interfaces makes it possible to combine and reuse elements for various applications.

Chat

info@librecube.org