Sidharth Shambu
sidharthshambu00@gmail.com
Timezone: India Standard Time (IST) UTC + 05:30

Proposal for CCSDS Space Packet / USDLP Transport Layer for Python and MicroPython

GSoC Project proposal for LibreCube

Personal Details and Contact Information

Email: sidharthshambu00@gmail.com
GitHub Handle: shambu2k
University: National Institute of Technology, Tiruchirappalli
Timezone: India Standard Time (IST) UTC + 05:30
Address: GB, Rosslyn Place No.5, Dr.Guruswamy Road, Chetpet, Chennai, Tamil Nadu, India (600031)
Phone Number: +91 94457 37949
Element.io handle: @shambu2k:matrix.org
Linkedin: sidharth-shambu
Resume: Link

Synopsis

The goal of this project is to **implement the Space Packet Protocol (SPP) and the Unified Space Data Link Protocol (USDLP) in python and micropython and ultimately demonstrate the protocols by transferring data between a Micropython supported development board (Pyboard or ESP32) and a Laptop/PC**.

The SPP and USDLP is used to efficiently transfer space application data through different space-links such as ground-to-space, space-to-ground, space-to-space, or on-board communication links.
As of now there are very limited available implementations of SPP and USDLP. Implementing these protocols will be of great help to amateurs and students who find it difficult to get their hands on an open source solution with community support.

My goals for this project is to implement these two protocols in Python and Micropython and demonstrate the data transfer using ESP32.
I will be working with the mentors - Milenko and Artur for this project as mentioned in librecube.org

Benefits to the Community

As mentioned before, I couldn't find many proper implementations of these protocols. I aim to build a complete python module where anyone can easily use them in their own projects and missions.

- Python being a very popular programming language, implementing these protocols in python and micropython will encourage more enthusiasts to find it easier to approach and use them in applications and missions. It might also bring more people to contribute to this project.
- This project will bring LibreCube one step closer in being a complete open source community for space data system softwares and hardwares.
- The project will be of great use to amateur enthusiasts or students in their missions. In the future, this project can also evolve to be used officially by different space agencies.

Current Status of the Project

There is no existing repository for this project and has to be implemented from scratch. The protocols are to be implemented in Python and later ported to Micropython to be demonstrated using an external development board (Pyboard or ESP32).
The prototype I made has python classes for data packets involved in SPP as per the specifications mentioned in this document.
Link to sample code - here
I will be updating this repository during the application review period.

Findings so far

I have been reading the official CCSDS documents for SPP and USDLP.
Let me summarize what I have learnt so far.

SPP

This forms the data layer as per CCSDS communications stack of satellite communications.
Much like the Internet Protocol (IP) the space packet protocol is a data unit that contains an Application Process ID to identify its contents, headers and data fields.
A packet mainly consists of 4 parts - Packet Primary Header, Packet Secondary Header, Packet Data Field and User Data Field.

Each of these further is made up of different variables. For example the Packet Primary Header consists of Packet version number holder, Packet sequence control flags, packet name holder, packet identification and packet data length.
All of these specifications including how much octet each variable should occupy are mentioned in detail in the CCSDS docs for SPP.

USDLP

This is basically the data link layer which facilitates transfer of data between the physical channels and the network layer for space links. It provides functions to help in transferring the spp data units (transfer frames).
One main feature of a space data link protocol is virtual channels (VC). As far as I understood from the CCSDS docs, if we use multiple VCs to a physical channel, it will enable the physical channel to be used by multiple data streams. Each of the transfer frames mentioned above belongs to one of the Virtual Channels.
This protocol also allows data units from different sources to be joined together (multiplexed) into one Virtual Channel using Multiplexer Access Points (MAPs). This is done using MAP IDs which tells the Space Data Units to go to a Service Access Point (SAP) with that MAP ID.
The overview of the implementation of this protocol is best represented here.

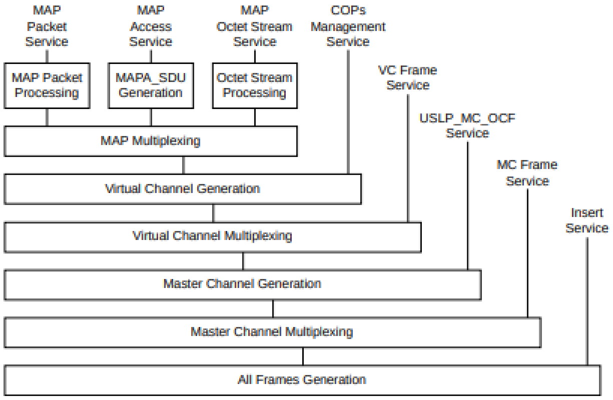

Figure 2-6: Internal Organization of Protocol Entity (Sending End)

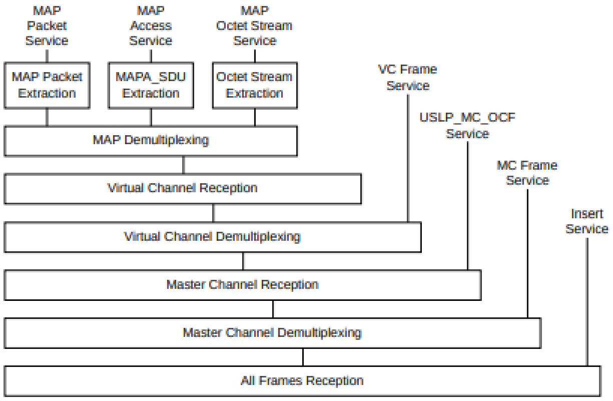

Figure 2-7: Internal Organization of Protocol Entity (Receiving End) [1]

[1] Taken from page 37 of USDLP

Micropython/ESP32

I am completely new to microcontroller programming but I have bought a module and have tried running small programs in Micropython in the ESP32. I am yet to figure out the complete demonstration once I have coded the protocols.

UDP Transport layer

For the demonstration, I can implement UDP using sockets as the underlying transport layer. ZeroMQ cannot be used since it is not supported in Micropython.

My Prototypes

The prototype I made has python classes for data packets involved in SPP as per the specifications in https://public.ccsds.org/Pubs/133x0b2e1.pdf (SPP).
Link to sample code - here
I will be updating this repository during the application review period.

Goals

These are rough milestones that I have set for myself for the completion of this project.

1. Implement Space Packet Protocol in Python and Micropython

The CCSDS specifications will be kept as a reference to code the complete structure of the data packet along with their utility functions.

2. Implement functions for the USDLP

The data link layer protocol needs to be implemented by creating various functions for multiplexing the data packets into VCs, generating channels and transfer frames.
The same has to be extracted and demultiplexed in the receiving end.

3. Ensure complete support in Micropython

Here I will ensure complete support in Micropython and will also ensure minimum external modules are used

4. Demonstrate the library in an ESP32 module.

Set up the ESP32 module to demonstrate the transfer of data packets using UDP as the underlying transfer layer.

5. Write tests and documentation.

Write test cases and provide proper documentation so that it will be helpful for other open source enthusiasts looking forward to contributing to this project.

Deliverables

1. Before first Evaluation - SPP & USDLP module

- Before the first evaluation on **July 12th 2021**, I will be able to make a complete usable module for Space packet protocol and the Data Link Protocol in Python.
- It will provide utility functions to generate the packets and also provisions to multiplex and demultiplex the data units into virtual channels.
- Also this module will be ported to Micropython.

2. Before Final Evaluation - Demo, Tests and Documentation

- Before my final evaluation on **August 16th 2021**, I will use the above implemented module and demonstrate the transfer of data between my ESP32 module and my laptop using TCP sockets in ZMQ.
- Proper tests for the module will be written.
- This module will also be thoroughly documented to help users and future contributors.

Expected Results

1. A complete python module of the Space packet protocol and the Unified Space Data Link Protocol. This python module can be used by enthusiasts and students in their own projects or missions.
2. 100% coverage of tests and documentation.
3. Successful demonstration of the module using an external microcontroller (ESP32)

Approach

- CCSDS standards are new to me. But I had already started looking into it almost a month back. I have a Network and Communications course as part of my minor program in computer science in my institute. So I was able to understand the nitty-gritty details in the CCSDS specifications for SPP and USDLP.
- I have summarized my findings above.
- I will further look more into these protocols, their uses and applications in space missions. This will provide me with a better understanding and motivation to implement this project.

- ## SPP
  - First I will write data unit classes for the Space Packet Protocol and its utility functions in python. Simultaneously while also ensuring that it is also supported in Micropython. The SPP packet class will have the following fields:
    - Packet Primary Header
    - Packet Secondary Header
    - Packet Data Field
    - User Data Field
  - Apart from this, we need
    - Packet Assembly function
    - Packet Transfer function
    - Packet Extraction function
    - Packet Reception function
- ## USDLP
  - Next, As part of the USDLP, I will have to implement the required functions for sending end such as:
    - MAP Packet processing function
    - MAPA_SDU generation functions
    - MAP octet street processing function
    - Multiplexing function
    - Virtual channel generation and multiplexing function
    - Master channel generation and multiplexing function
    - Frames generation function
  - Similarly, at the receiving end we need:
    - MAP Packet extraction function
    - MAPA_SDU extraction function
    - MAP octet stream extraction function
    - MAP demultiplexing function
    - Virtual channel reception and demultiplexing function
    - Master channel reception and demultiplexing function
    - Frames reception function
- ## ESP32 setup and UDP Transport Layer
  - Here I need to implement UDP using sockets or (usockets) which will be used as the underlying data transport layer for the demonstration with ESP32
- ## Tests and Documentation
  - Upon completion of the module, the rest of my time will be spent in fixing any bugs(if any).
  - Also I will try to write tests aiming for 100% coverage.
  - A thorough documentation will also be written.

The above to my knowledge is more or less an extensive todo list for this project. All along the process I am aware that I have to ensure support in micropython and also use minimum external dependencies as possible.

Timeline

This section shows how I am planning to spend my time during the GSoC programme.

Period	Task
After proposal submission [April 13 - May 17]	- Look more into USDLP's multiplexing and Virtual Channels - Code Prototype for USDLP - Interact with the community to know more.
Community bonding period [May 17 - June 7]	- Get to know more about everything that librecube does - Get to know about the people of Librecube and mentors - Start coding!
Coding Period - 1 [June 7 - July - 12]	- Write python module for SPP - Write python module for USDLP - Ensure Micropython portability
Evaluation - 1 [July 12 - July 16]	- Get feedback from mentors - Fix bugs if any - Make any necessary changes mentioned by mentors
Coding Period - 2 [July 17 - August 16]	- Setup the module in micropython for demo. - Implement UDP for the transport layer - Write tests and documentation
Final Evaluation [August 16 - August 23]	- Submit code for review - Make final changes if any - Start writing a blog post documenting the whole experience in GSoC with Librecube

About Me

Hi, I am **Sidharth Shambu**, a 3rd year undergraduate majoring in Chemical Engineering with a minor in Computer Science from India. I am also a self-taught programmer, and I am an active member of the coding club of our Institute for the past 2.5+ years. We maintain the official college website, develop sites and apps for various college fests, undertake projects from our college admin, etc. I am also a member of the Astronomy and Science club of our college - Nakshatra.

Relevant Skills and Interests

- I have a very good experience in **python**. I have been making **backend applications using Flask and Django (python frameworks for backend) and also have made a bot (also in python)** such as the one which posts information about upcoming rocket missions, NASA's APODs and much more. You can find those automated posts here.
- I have also done image manipulation coding using OpenCV in python.
- I have been a space and astronomy enthusiast since my childhood, I have participated in many astronomy quiz competitions in high school.
- After joining college, I have picked up a hobby of creating beautiful images of nebulas and star clusters using DS9 SAO software using telescopic FITS data.
- I have also contributed to Opensource projects for OpenMifos and also in DWoC.
- Other software skills
    - **C, C++**
    - **Kotlin, Java**
    - **Javascript**
    - **Linux environment**
    - **VS Code**
    - **Android Studio**

Availability

- I am currently a student at National Institute of Technology Tiruchirappalli, India and I am eligible for this programme.
- I have not written any other proposals for any other organizations.
- I would also like to mention that I am completely free this summer and **won't** be taking up any intern programs.
- Since I have **no other commitments** I will be able to devote 100% of my time for this project. I will be able to put in 40-50 hours per week.