



COMMUNICATION PLUGIN IMPROVEMENT

SOFA-FRAMEWORK

GOOGLE SUMMER OF CODE 2018

Name: Rupesh Harode

Email: rupeshharode@gmail.com

Github: [firedranzer](#)

University: [Shri GS Institute of Technology and Science](#)

Course: B.Tech in Electronics and Telecommunication Engineering

Time Zone: IST (UTC +5:30)

PURPOSE

The purpose of this project is to improve the communication plugin by adding more protocols in it such as vrpn, serial bus communication, and also improve the way it works.

INTRODUCTION

Communication plugins are used in SOFA to send and receive data. Currently two protocols are implemented in SOFA - OSC and ZMQ. Communication plugin can be improved further by introducing new protocols and optimizing them. This project specifically deals with introduction of 2 new protocols namely - VRPN and Serial Bus.

VRPN

The Virtual-Reality Peripheral Network (VRPN) system provides a device-independent and network-transparent interface to virtual-reality peripherals. VRPN's application of factoring by function and of layering in the context of devices produces an interface that is novel and powerful. VRPN also integrates a wide range of known advanced techniques into a publicly-available system.

VRPN is a set of classes within a library and a set of servers that are designed to implement a network-transparent interface between application programs and the set of physical devices (tracker, etc.) used in a virtual-reality (VR) system. The idea is to have a PC or other host at each VR station that controls the peripherals (tracker, button device, haptic device, analog inputs, sound, etc). VRPN provides connections between the application and all of the devices using the appropriate class-of-service for each type of device sharing this link. The application remains unaware of the network topology. Note that it is possible to use VRPN with devices that are directly connected to the machine that the application is running on, either using separate control programs or running all as a single program.

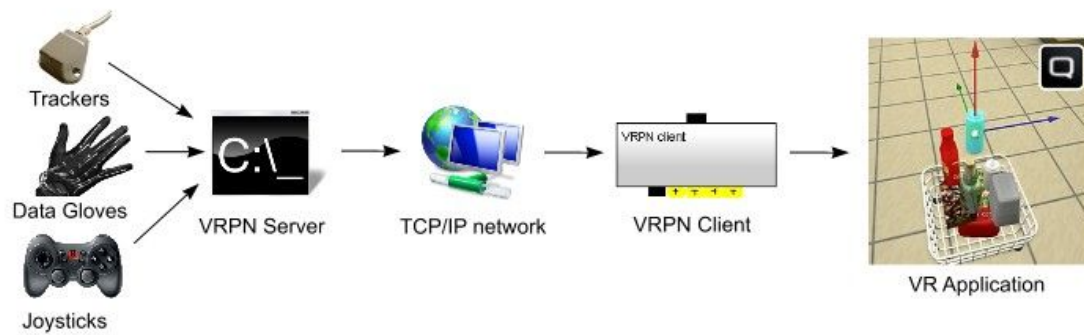


Figure 1. Working of VRPN

Serial Bus Communication

Serial communication is the process of sending data one bit at a time, sequentially, over a communication channel or computer bus. This is in contrast to parallel communication, where several bits are sent as a whole, on a link with several parallel channels.

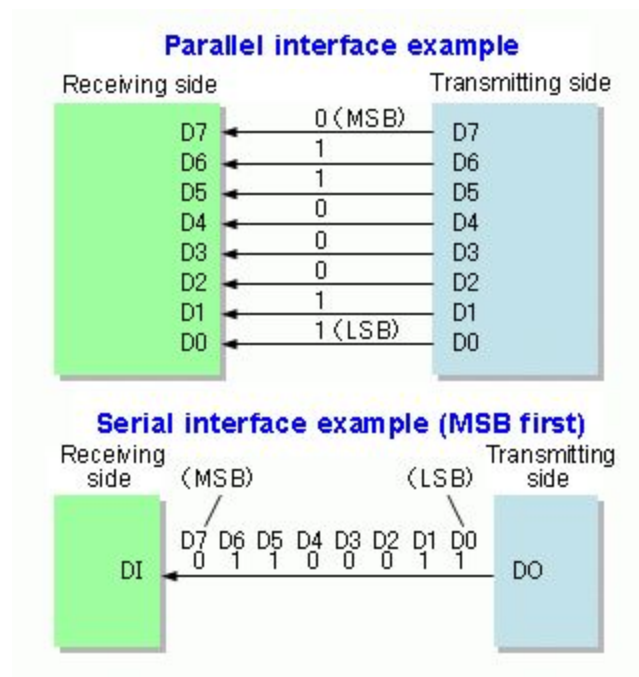


Figure 2. Parallel VS Serial Communication

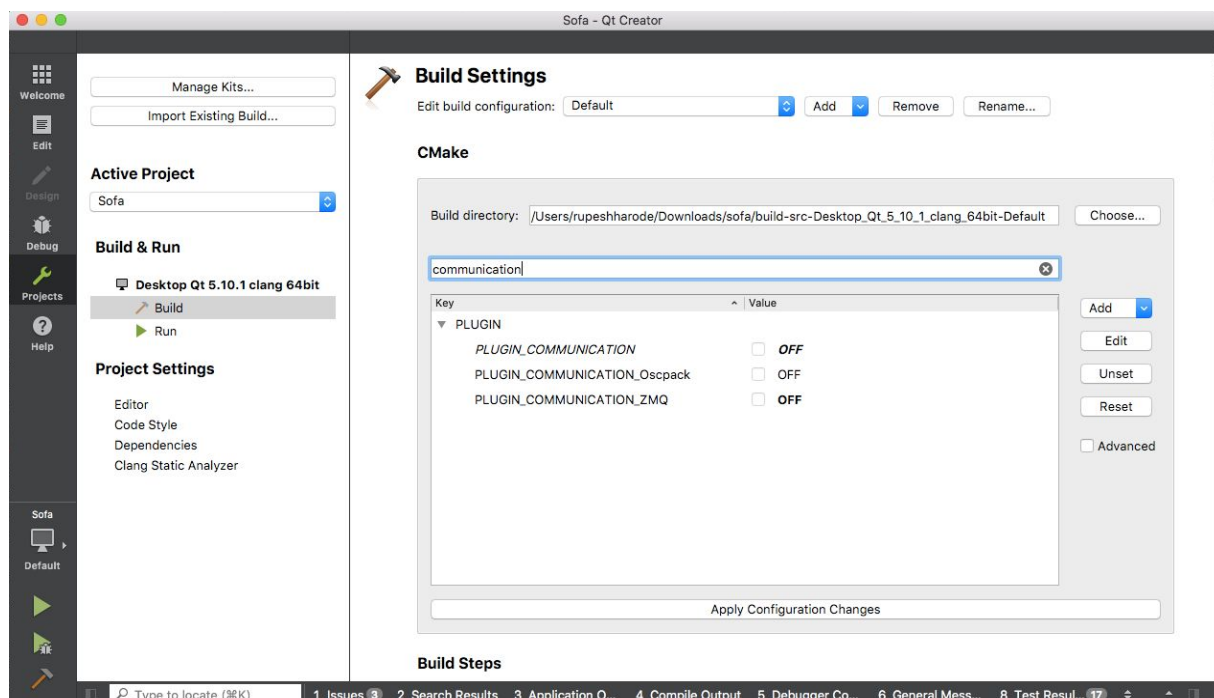
PROPOSED DELIVERABLES

Improve the Communication Plugin.

1. Adding more protocols such as - VRPN and Serial bus protocols.
2. Extending functionalities for different protocols.
3. Adding tests and bug fixes.
4. Increase support for different OS.
5. Adding documentation related to communication plugin

IMPLEMENTATION DETAILS

Currently, the entire code related to the communication plugin resides in [this](#) branch. While the code simplifies the implementation of plugins in Sofa-framework and unifies the protocols, but it is still in development phase and not merged in the stable master branch. The current implementation of the plugin contains OSC and ZMQ.



This screenshot shows current communication plugin in the sofa-framework. This will further be expanded as a part of project.

Also, communication plugin currently supports only Ubuntu and Windows and its support on MacOS is still in progress.

The project requires addition of VRPN and Serial bus protocols in similar fashion.

Communication Plugin

Protocols can be implemented in sofa using Communication Component. It is used to send and receive sofa's data.

ServerCommunication

`ServerCommunication` is an abstract class allowing users to create asynchronous communication class. Actually, there are two implementations of it. One using the OSC protocol and the other one using ZMQ protocol.

Subscriber

`Subscriber` is a class needed to `ServerCommunication`. This will allow user to define which kind of message he wants to send/receive and which sofa's data he wants to bind to.

Currently these implementations are used to send and receive data via OSC and ZMQ protocol.

Implementing new protocols VRPN and Serial Bus

`ServerCommunication` class must be extended to implement a new protocol. This means we have to implement new methods. Some of these virtual methods are -

- `getFactoryInstance`
- `initTypeFactory`
- `sendData`
- `receiveData`
- `getArgumentType`
- `getArgumentValue`

- defaultDataType

Sending and Receiving data

To implement new protocols these functions are need to be implemented.

```
virtual void sendData() override;  
virtual void receiveData() override;
```

Both are ran by the mother class inside a thread. This will be the place where we loop for sending or receiving datas.

For receiving and sending datas we will need to fetch them using the mother class function named `fetchDataFromSenderBuffer` and `saveDatasToReceivedBuffer`.

Note: All the implementation mentioned in this proposal is tentative. I will present a more detailed form of implementation during community bonding period.

TIMELINE

Pre-GSoC + Community Bonding Period (Present - May 14)

Most part of this part would be engaged in my academics as final exams of college are scheduled in this period. I will still try to take out the maximum time for the GSOC project.

During this period, I will enhance my knowledge about communication plugin and learn about various methods used in the plugin including sending and receiving datas.

- I'll spend this time fine tuning the milestones of my deliverables and getting more clear about project implementation with the help of my mentors.
- Dive Deeper in the SOFA codebase and test different features of the framework. Learning more about VRPN, Serial Bus protocol and becoming familiar with the coding style followed.
- Discuss broadly about the design and possible hurdles/problems of implementation of the idea and learning more.

- Communicating with mentor on regular basis to get feedback and discuss various implementations.

Week 1-2:

- Set up [blog](#) to document my progress through the project.
- Setting up trello to manage timeline effectively.
- Starting the implementation of VRPN protocol and learn how it works
- Communication plugin might works on your machine + started to learn how it works

Week 3-4:

- Starting the VRPN protocol support for the communication plugin.
 - Architecture
 - Basic functions, like sendData, receiveData and being able to use them

Phase 1 Evaluation

- Improved architecture of the Communication plugin with VRPN protocol support added in the plugin.

Week 5-6

- Testing serial bus communication protocol and deciding essential features to add in this phase.
- Testing various functionality of plugin and thereby implementing improvements and fixes.

Week 7-8

- Adding serial bus communication protocol in Communication plugin and implementing its various functionality.
- Improving support for the protocols.

Phase 2 Evaluation

- Serial bus communication protocol added to the Communication plugin.
- Improved tests and Bug fixes.

Week 9-10

- Extending support for protocols by testing over different OS.
- Testing new protocols and extending necessary functionality.
- Maintaining documentation.

Week 11-12

- Finalizing the project and Final testing.
- Improving quality of code before final submission.

Final Evaluation

- Communication plugin improved by adding protocols like VRPN and serial bus communication.
- Extended support for communication plugins by writing tests and fixes.
- Improved support for various OS.

NOTES REGARDING TIMELINE

1. I haven't dedicated specific weeks for documentation as I believe documentation is a part of coding itself. I'll try to keep my code as documented as possible while writing the code itself. Also if I miss documentation somewhere, it will be covered as mentioned in timeline.

2. I will be writing weekly blog posts about the milestone at my [blog](#), address of which is mentioned in the proposal.

3. I realise that open source is all about communication and collaboration and I like it that way. Hence, I will be in continuous contact with mentor([@ErwanDouaille](#)) throughout the program describing him the progress and problems I find.

4. The timeline above is tentative. I will prepare a detailed and more fine checklist, which I will host as a Github gist on the above mentioned blog after or during community bonding period. Thus, in this way checklist will be available to everyone with easy accessibility. It will also become easier for mentor to add task directly to that checklist and asses them.

5. I have mentioned my working hour/ environment / availability etcetera later in the proposal.

6. If everything is on schedule or I complete some milestone early, I can proceed to the next task ahead of time.

7. I am very enthusiastic about my work being beneficial to other people in the open source community. I'll keep on seeing the work done by me and fixing issues if they emerge in future.

REQUIREMENTS

- Permission to github repository to push code changes to the stable repository.

Any requirements would be conveyed during pre-GSoC period while fine tuning the milestones.

PERSONAL INFORMATION

Planned Work Schedule, Environment and Absences

- I intend to work on all weekdays (Monday to Friday) and a bit on weekends too, and will easily able to work 40+ hours a week.
- The checklist on the blog will cover all weekly milestones.
- My local time zone throughout the summer will be IST (Indian Standard Time) which is GMT + 05:30.

- As far as the times of the day are concerned, I am pretty flexible, and am willing to adjust with what suits my mentor the best. I have already been working with them for past 1 month.

I will be working from either my home or college hostel. I have access to a stable internet connection at both places along with an active 4G connection. Internet outages are rare, although sometimes I experience momentary disconnections for a few minutes or so but the 4G should suffice at those times.

I have no planned absences over the summer, but in case any arises - I'll be prompt to inform my mentor about the same and use the buffer week to catch up. I shall be available to work full time with Sofa throughout the period of GSoC. My summer vacation will end mid July(the official dates aren't announced yet). For the last weeks of the program, I will also be attending college, but the burden is minimum in initial weeks of the college, so I shall be able to dedicate enough time for project.

PERSONAL DETAILS

I am a 3rd year undergraduate student pursuing Electronics and Telecommunication Engineering at Shri GS Institute of Technology and Science, Indore.

I have been programming for about 4 years now. I started with C and C++. I've used C++ mainly for competitive programming contests where I got comfortable with STL. I started working on Python 2 years back. I like the ease with which one can code in python. I also learnt web-scraping/crawling as a part of internship project. I developed an Optical Character Recognition(OCR) while working on OpenCV. Apart from C++/C and Python, I did projects in Java and JavaScript.

Also, I am Technical Lead at Club Codefoster.Codefoster is the official coding club of the college where we take regular workshops on new technologies in various fields of Computer Science and Electronics.

I am quite comfortable using git/github for version control, been using it for more than a year and have learnt a lot about it whilst opening Pull Requests, creating branches, squashing commits, rebasing etc .

Platform Details

OS: MacOS High Sierra 10.13, Ubuntu 17.10(On VMWare)

Hardware Configuration: i5 5200U/ 8 GB

Editor: QtCreator

ACKNOWLEDGEMENT

- Erwan Douaille([@ErwanDouaille](#))

REFERENCES

- Communication Plugin Documentation
(<https://github.com/SofaDefrost/sofa/tree/sofaCommunication/applications/plugins/Communication>)
- Virtual Reality Peripheral Network - Official Repo
(<https://github.com/vrpn/vrpn/wiki>)
- Serial Communication protocols: The basics
(<https://www.totalphase.com/blog/2017/08/serial-communication-protocols-the-basics/>)