

Create a WebUI for data mapping and software control in embedded systems

Yash Raj

2022-04-11 12:00:00 UTC+05:30

Name and Contact Information

- name: Yash Raj
- website: <https://tiger-yash.github.io/>
- gitlab username: [@tiger-yash](#)
- timezone: UTC+5:30

Title

Create a WebUI for data mapping and software control in embedded systems

Synopsis

Open Source tools, such as those developed by SAT, have been widely used for a variety of applications by academics and independent users. The tools benefit users a lot and make their work easier, more efficient, and less stressful, but most of them need lengthy setups and installations, which are not very user-friendly.

This GSoC project attempts to integrate tools like SATIE and LivePose into a bespoke WebUI and make them accessible. The UI will be responsible for software configuration, deploying an audio synthesizer on the back-end, etc.

The WebUI will be capable of deploying the tools (start/stop LivePose and SATIE, control Jack Audio parameters, and request reboot/service restart) and basically remotely launching (and possibly integrating and mapping) SAT tools in embedded systems based on the Raspberry Pi generic computer running Linux.

Any program or product's goal is to provide users with a rich and interactive experience; similarly, our WebUI will serve as an entry point to those tools with simple customizations that will make users' life much easier.

Benefits to Community

Metalab(SAT) develops tools for artists and new media which are mostly software-based. Most artists have difficulties installing and using open-source tools, especially if they require extra steps to configure, e.g., compile code or specific configurations.

The Web UI will aid artists and other users right away by allowing them to execute tools remotely, saving them time and effort in setting up, installing, and eventually using the tool for the intended purpose. This will also allow more useful tools and features to be included in such user-friendly UIs, which will be of immediate use to users who are unfamiliar with the internal setups of these tools, allowing them to focus on the creative aspects instead of setting up and compiling the code for the tools.

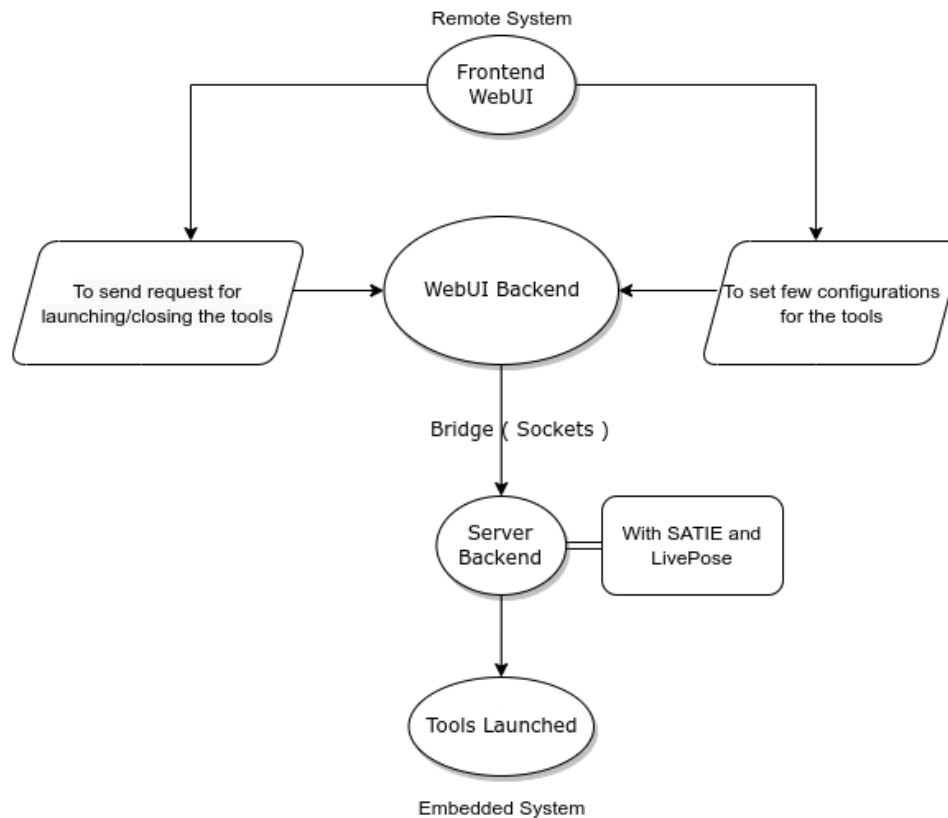
Everyone will be able to launch and access the tools directly through the GUI in only a few clicks, and they will not need to understand the procedure in depth.

Deliverables

Milestones

- Create the Server (Embedded system) backend with the tools set up and operational.
- Create a backend for the WebUI.
- Connect both the backends with a suitable bridge.
- Develop the Frontend UI for launching and configuring the tools.

Architectural Diagram



Timeline

- Pre GSOC Period
 - April 20, 2022 - May 20, 2022
 - * Stay connected with the community and also discuss the tools and project.
 - * Learn more about Open Sound Control (OSC).
 - * Explore the tools LivePose and SATIE in greater depth.
- Community Bonding Period (May 20, 2022 - June 12, 2022)
 - May 20, 2022 - June 4, 2022
 - * Get better acquainted with the mentors and other Contributors.
 - * Discuss approaches to bridge the frontend and the webserver.
 - * Discuss and Finalise the tech stack.
 - June 5, 2022 - June 12, 2022
 - * Discuss and design the basic UI for the frontend.
 - * Lay out an implementation/integration plan.
- Coding Period (June 13, 2022 - September 12, 2022)
 - Week-1 & Week-2 (June 13, 2022 - June 27, 2022)
 - * Set up and install the tools for the embedded backend.
 - * Create functions to integrate and run SATIE.
 - * Start the documentation for the Server Backend.
 - Week-3 (June 27, 2022 - July 4, 2022)
 - * Create the functions to run LivePose.
 - * Update configuration settings code for SATIE.

- Week-4 & Week-5 (July 4, 2022 - July 18, 2022)
 - * Add configuration settings functions for LivePose based on decisions made earlier.
 - * Run and test the tools through the server backend.
- Week-5 (July 18, 2022 - July 25, 2022)
 - * Begin Coding for the backend part of the WebUI.
- Phase-1 Evaluation (July 25, 2022 - July 29, 2022)
- Week-6 & week-7 (July 25, 2022 - August 9, 2022)
 - * Develop a rough UI for the frontend.
 - * Connect the UI with the backend.
 - * Design and discuss the web-socket connection plans for the bridge between the backends.
 - * Start the documentation for the WebUI frontend and backend.
- Week-8 (August 9, 2022 - August 16, 2022)
 - * Add the socket functions at the WebUI backend, based on the discussions with the mentors.
 - * Discuss designs for a final UI and appearance.
- Week-9 & Week-10 (August 16, 2022 - August 30, 2022)
 - * Complete the socket connection at the Server backend.
 - * Test the bridge connection for issues.
 - * Modify and Upgrade the Frontend UI based on the layout decided.
- Week-11 (August 31, 2022 - September 5, 2022)
 - * Give time to bug fixes and implement any pending features.
 - * Test the tools again directly from the system backend.
- Final Week (September 5, 2022 - September 12, 2022)
 - * Test the final code for any issues.
 - * Finish up the documentation.
 - * Submit the final work and the work report.
 - * Discuss with the community the next steps and Post-GSOC work.
- Code Submission and Final Evaluations (September 12, 2022 - September 19, 2022)
- Post GSOC Period : Based on community and core developers' recommendations, work on newer modifications to the project. Stay connected with the community and keep on adding and improving the project.

Related Work

The target of this project is to create a Web User Interface, which will be used to remotely launch tools like [LivePose](#) and [SATIE](#) installed and running on an embedded system. It will also be able to set the configurations for the tools.

This is very similar to an existing open source project by SAT: [Poire](#) which is a web controller for [SATIE](#). The Poire stack is composed of a Back-End and a Front-End, so you must run every part of the stack in order to have a functional Poire setup. This type of setup is similar to what I intend to follow for my project but with quite a few modifications like using the tool not in the browser itself, but on a different system remotely. Also, Poire is not up to date with the current SATIE code base so this project will also be able to achieve a newer and more robust design.

Another existing open source project is [Telluriq](#) which is a graphical user interface that monitors and calibrates haptic floors. It provides the users with an easy way to manipulate a range of D-Box actuators in order to prepare experimental performances with the haptic technology. This project differs from my project in the sense that it is providing an interface to run and use haptic floors but I can still mildly relate to this project as we are also aiming to create a GUI in the form of WebUI for our users to launch the tools.

There are several AI/ML solutions aimed at serving the demands of artists or ordinary users in general, but no big projects or websites exist that provide these technologies with a direct and simple-to-understand launching platform. This project aims to address this issue by making these complex tools accessible to everyone, regardless of prior experience or comprehension of the installation or setup procedure.

Biographical Information

I am Yash Raj, a Computer Science and Engineering undergraduate student at the [Indian Institute of Technology \(BHU \)](#) pursuing a Bachelor of Technology in my second year. I was introduced to the world of programming and

software development in my first year. Since then, I have been very enthusiastic about deep diving into various fields of Computer Science.

The areas that capture my interests are Data Structures, Algorithms, Operating Systems, Web Development, and Information & Security. For most of my programming journey, I have worked primarily with Web-based technologies, my niche, and C/C++ programs, and I have recently been diving a little into neural networks and android development as well.

In addition to my academic activities, I am an active member of the [Software Development Group](#) in the [Club of Programmers](#). As I am also a part of the CyberSec Group of IIT BHU, I have participated in several CTFs, including the CSAW Cybersecurity Games and Conference where I along with my Team became CSAW'21 Capture the Flag Finalist India (AIR 8) and represented India in the Finals. Recently, I worked as a Tech Executive in the tech team of [Technex'22](#), 83rd Edition of Asia's Oldest Techno Management Fest.

I started my open source contribution in Hacktoberfest of 2020, where I contributed to several organizations daily. Apart from the ones I stated above, the major projects I have worked on can be found on my [Github Profile](#).

Skills

- required: basic knowledge of networking protocols, specially OpenSoundControl (OSC)
 - I have implemented several HTTP Servers primarily for use as REST APIs from the ground up using vanilla NodeJS and Python and worked with higher-level APIs of frameworks like Django and Express. While implementing these projects, I have gained insights into working with Web Services like web sockets and internet protocols.
 - I have worked on Web Apps where I needed to write code to implement REST APIs in Django or ExpressJS, and to handle HTTP requests.
 - I have gone through the [OpenSoundControl\(OSC\)](#) networking protocol in-depth and understood its basic functionality and implementation.
- required: passing knowledge of or willingness to learn Python and C++ for adaptations of the embedding backend
 - Created a client-server chat CLI application developed using the Python sockets module. ([Source Code](#))
 - As part of our institute's CyberSec team, I have participated in several CTFs (Capture the Flag) competitions where I have used python scripting for binary exploitation, pwning and, C for reversing tasks. My team also qualified for the CSAW'21 CTF Finals ([Credentials](#)).
 - Developed a website called Sociolib with backend in Python using Django REST Framework. ([Source Code](#))
 - I have been using C/C++ for the past 1.5 years to compete in the sport of Competitive programming and to implement and understand various Data Structures and Algorithms, which makes me pretty familiar with the languages. ([Code](#))
 - I have also used C/C++ for implementing several Parallel Programming and Deadlock Prevention algorithms.
- preferred: willingness to learn NodeJS and JavaScript for the frontend (there is some flexibility on the choice of the WebUI frontend language)
 - Past Internship at [Acadza Technologies](#) as a Frontend Engineer. Worked with ReactJS in the frontend. Also, worked with the NodeJS API backend for the same website. ([Website](#))
 - Developed a website called Sociolib with frontend in NextJS. ([Source Code](#))

Expected size of project

175 hours

Rating of difficulty

medium