Department of Computer Science & Engineering
The University of Texas at Arlington

Project Charter
CSE 4316: Senior Design I
Fall 2020

Stream Hoppers
The Stream Hopper

Seth Jaksik
Justin Erdmann
Kevin Chawla
Dominic Kotzer
Alexander Isaula

Revision History

Revision	Date	Author(s)	Description
0.1	10.05.2020	SJ, JE, KC, DK, AI	Document creation

Contents

List of Figures

1 Problem Statement

Streaming has become an extremely popular past-time and career in the recent years, with Twitch.tv being at the center of the revolution. While Twitch and other streaming providers have tools that connect actions in the stream, such as donations and subscriptions, to software events such as pop-ups or sounds to be played in the stream, there is very little to no support for these triggers to be used on physical devices around the streamer.

2 Methodology

We will build an Internet of Things (IOT) hub to be connected to the streamer's home network and various other IOT devices the streamer may have. The hub will allow the streamer to set triggers for specific actions in the stream, which will then trigger a device connected either through USB or on a subnet specific to the hub to perform an action. The hub will have a GUI that will allow the streamer to configure which triggers they want active as well as what action they will cause, and also includes presets for varying streaming setups.

3 Value Proposition

Our IOT device will provide immense value to any user that utilizes our device. The art of streaming relies on viewer retention and entertainment in order to convince advertisers to invest in you. Any added content to an individuals stream will inevitably provide increased entertainment value. The IOT device will provide lights and noises that will seamlessly entertain the viewer and add to the overall stream experience. In addition to the physical benefit, viewers will be able to interact with this device providing an overall more active viewer experience.

4 Development Milestones

This list of core project milestones including all major documents, demonstration of major project features, and associated deadlines. Any date that has not yet been officially scheduled at the time of preparing this document has be listed by month. Dates are subject to change.

List of milestones and completion dates:

- Project Charter first draft - October 5, 2020
- System Requirements Specification - October 26, 2020
- Demonstration of preliminary website GUI controller - December 2020
- Detailed Design Specification - December 7, 2020
- Demonstration of preliminary integration to Twitch triggered Events - December 2020
- Architectural Design Specification - February, 2020
- Demonstration of initial GUI control of triggers - February 2021
- Demonstration of Twitch action triggering event - March 2021
- CoE Innovation Day poster presentation - April 2021
- Demonstration of GUI input impacting Twitch triggered event response - April 2021
- Demonstration of GUI preset feature - April 2021

- Demonstration of full product - May 2021
- Final Project Demonstration - May 2021

5 Background

Although several services exist that allow for Twitch events to be triggered by actions that occur during the stream, there is an apparent lack of support for similar automation of physical devices, such as turning on or off lights, changing the color or pattern of lights, and various other dynamic effects in the area around the streamer. Some streams use these sort of hardware setups, but whenever the appropriate action occurs in stream they have to divert their attention off the game and to manually change the device configuration as requested. This product will reduce the need for streamers to micromanage these hardware components on their stream and allow them to focus on the quality of the performance they give.

Our solution is to build an IOT hub that will run on the streamer's home network, allowing them to set up custom triggers to their hardware devices that are prompted through the Twitch API. The streamer will be able to set up a variety of hardware interactions that can be an additional draw for viewers retention and entertainment, without the added pressure of having to manually perform the changes themselves during a stream. Since interaction with the viewers is a key factor in streaming, this automation of actions based off triggered stream events will allow the streamer to focus more on the quality of their stream and less about micromanaging their interactions with live viewers. The primary expenses for our project include a Raspberry Pi, router, and the various IOT devices that will be configurable to trigger on stream events.

Our sponsor, Dr. Shawn Gieser, has a particular interest in this project so that he can utilize it on his Twitch stream. Our existing relationship with Dr. Gieser is as our professor in the Senior Design course.

6 Related Work

There are very few solutions to the current basis of our project but the one's related are as follows:

Azure IOT hub is available commercially. It could be used for the same purpose trigger devices through stream but it would require some modifications and the reason of it not being as effective as our concept is that it lacks usability because it doesn't have any user interface available. This device would not have a good user experience. [4]

Confluent.io hub is available commercially. It is used to control IOT devices through the cloud but it doesn't have the ability to be connected with the twitch API and is not cost effective even if modified further. [5]

Elgato Stream Deck is available commercially. It has 15 LCD keys poised to launch unlimited actions eliminate the need to map and memorize keyboard shortcuts. One-touch, tactile operation lets you switch scenes, launch media, adjust audio and more, while visual feedback confirms your every command. Traditionally this level of control was exclusive to mainstream entertainment broadcasters. As useful as it is, it can not be used to achieve the goals of giving the users control of the streaming environment. [1]

Streamlabs is a live streaming software that integrates Open Broadcaster Software with viewer interactions, chat management, and tip donations. [2] It provides the streamer with an interface to their stream and allows them to activate digital notifications and alerts using a stream overlay. Although this can be very useful and entertaining, it does not allow for external hardware to be activated via Twitch. It is strictly capable of providing digital entertainment which is why our IOT hub is a necessary product for the streaming industry. We plan to build on Streamlab's interface and bring the streamers notifications to life through In-Real-Life alerts.

AutoHotKey is a free script based software that allows for the creation of custom keyboard macros to be created. AutoHotKey allows for the user to write a simple script that is triggered on the press of a macro key as assigned in the software, which then triggers an action, similar to macro keys built into modern keyboards or the Elgato Stream Deck. Many streamers create custom AutoHotKey scripts with that coincide with their streams to mimic the functionality of tools similar to the Elgato Stream Deck without the cost of the hardware. These scripts can be used to change scenes of a stream, paste a message in chat, or even change an image on the streamers stream, but are unable to control physical devices. The scripts are also not connected to the Twitch API, meaning the streamer has to physically trigger the macros when an action occurs, rather than a piece of software triggering the action. [3]

7 System Overview

Our device will be a universal hub where multiple IOT devices can connect and interface with the Twitch website and activate on various Twitch events. First, an event will happen on the user's Twitch channel, such as a follow, donation or subscription, and through the API, a message will be sent to the Raspberry Pi. Based on the message from Twitch, an event will be triggered through the General-Purpose Input/Output pins on the Pi or via the internet to an IOT device. The user will be able to customize these devices through a Graphical User Interface and they will be able to add or delete their own unique electronic devices. In addition to customizing the devices, the user will be able to decide what events will trigger their devices.

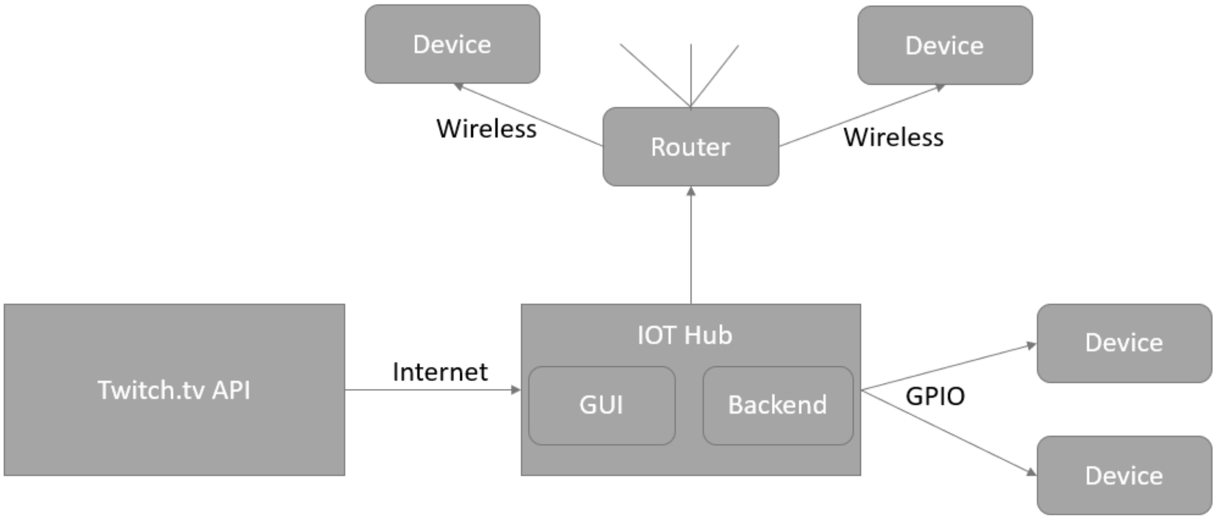

Figure 1: High Level Overview Diagram

8 Roles & Responsibilities

For this project, the stakeholder will be Dr. Shawn Gieser as this device will be used particularly on his Twitch stream. The point of contact for the project will be Justin Erdmann and he will be the bridge between the customer and the development team. Justin Erdmann, Seth Jaksik and Alexander Isaula will work on developing the hardware and connecting the IOT devices to the internet. In addition, Seth, Justin and Alexander will research and implement an internet connection via a router so the IOT devices can be activated via the internet. Kevin Chawla will develop the Graphical User Interface and the Front End to allow the user to add, delete and edit the IOT devices on the Raspberry Pi. Dominic Kotzer will develop the code that will connect our devices to Twitch so the IOT devices can activate on Twitch Events. This includes researching and understanding the Twitch API. The scrum master will be new each scrum so we can distribute the work load throughout the group.

9 Cost Proposal

The project is funded by the CSE Department at University of Texas at Arlington. Our Major expenses will include a Raspberry Pi(with touch screen), router and all the IOT devices that we will trigger using our Pi. We will be using these to host our application and create a local network to provide the Stream Hopper functionality.

9.1 Preliminary Budget

Product	Estimate cost
Raspberry Pi(with touch screen)	$150
Router	$30
Smart Plug	$30
Smart lights	$150
Speakers	$100

Table 1: Preliminary Budget Overview

9.2 Current & Pending Support

The main funding source of the project would be the CSE department, which will be providing an overall of $800 to complete the project. To utilize this fund we will have to go through the department's ticketing system. Additional financial support may be requested from the sponsor pending the cost of additional devices to trigger with the product.

10 Facilities & Equipment

For the Project a reserved space is needed, with a lab space the team will be able to utilize an internet network and better assemble the hardware and troubleshoot any issues. The lab space will also allow the teams to test the equipment and peripherals with ample space, network capabilities and available outlets. The equipment considered for the project include one of the versions of the Raspberry Pi platforms, depending on the amount of output devices, we will decide on the appropriate Raspberry Pi platform to use as the central IOT Hub to connect with the Twitch API. Other equipment that is needed for network establishment and connectivity is a Router. To create an interactive experience for the twitch streamer we will add peripherals to add effects to the streamer's environment such as smart lights changing colors when a twitch subscriptions occurs. Other peripherals include network cabling and USB cabling to connect network hardware and other peripherals. A touchscreen is also needed to provide accessibility to the Twitch Streamer. To obtain the equipment and peripherals, purchasing through the ticketing system is the easiest and cheapest option.

11 Assumptions

The following list contains critical assumptions related to the implementation and testing of the project.

- Lab Space is available throughout the Fall and Spring Semesters.
- UTA network will allow TCP network traffic on port 1883
- Equipment shipping will not be delayed
- Current pandemic does not get worse causing lock downs and or campus shutdowns
- Connectivity to the network is stable and not overpopulated.

12 Constraints

The following list contains key constraints related to the implementation and testing of the project.

- Final prototype demonstration must be completed by May 1st, 2021
- The IOT devices must have a public API
- Deliverables must be ready at the end of each sprint
- Total development costs must not exceed $800
- All IOT devices must be FDA approved
- Meet or exceed the standards established in the Association for Computing Machinery Code of Ethics
- Meet or exceed the standards established by the International Electrotechnical Commission

13 Risks

The following high-level risk census contains identified project risks with the highest exposure. Mitigation strategies will be discussed in future planning sessions.

Risk description	Probability	Loss (days)	Exposure (days)
Delays on shipping Raspberry Pi	0.60	8	4.8
Lab closed due to Covid	0.20	14	2.8
Internet access not available at installation site	0.20	4	0.8
Delays obtaining equipment from Department	0.30	7	2.1
Delay due to budget overshoot	0.30	14	4.2

Table 2: Overview of highest exposure project risks

14 Documentation & Reporting

14.1 Major Documentation Deliverables

14.1.1 Project Charter

The Project Charter will be maintained and updated by the team after each sprint is completed to reflect current progression of the project. The document will also be updated contingently on whether there are major changes to dates or other section of the document. The initial version of the Project Charter will be available on October 5, 2020, with the final version being delivered on completion of the project in May 2021.

14.1.2 System Requirements Specification

The System Requirements Specification will be maintained and updated by the team after each sprint is completed to reflect changes to the requirements of the system. The document will also be updated contingently on whether there are major changes to other aspects of the document. The initial version of the System Requirements Specification will be available on October 26, 2020, with the final version being delivered on completion of the project in May 2021.

14.1.3 Architectural Design Specification

The Architectural Design Specification will be maintained and updated by the team after each sprint is completed to reflect changes to the architecture design of the system. The document will also be updated contingently on whether there are major changes to other aspects of the document. The initial version of the Architectural Design Specification will be available on November 16, 2020, with the final version being delivered on completion of the project in May 2021.

14.1.4 Detailed Design Specification

The Detailed Design Specification will be maintained and updated by the team after each sprint is completed to reflect changes to the detailed design elements of the system. The document will also be updated contingently on whether there are major changes to other aspects of the document. The initial version of the Detailed Design Specification will be available in February 2020, with the final version being delivered on completion of the project in May 2021.

14.2 Recurring Sprint Items

14.2.1 Product Backlog

Items will be added to the product backlog from the SRS during Sprint Planning sessions and as new requirements are added to the SRS. These items will be prioritized based off of what is needed to satisfy the current customer wants for the sprint to ensure that tasks are completed in time. The decision will be made as a team with input from the technical expert for that section of the project as to what needs to be included to ensure progress is made. Trello will be used to maintain and share the product backlog with the team members and stakeholders.

14.2.2 Sprint Planning

Each sprint will be planned as a team following the Sprint Review for the previous sprint. There will be 8 sprints in total, with 4 occurring during Fall 2020 and 4 occurring during Spring 2021.

14.2.3 Sprint Goal

The Sprint Goal will be decided as a team during the Sprint Planning section prior to the start of a sprint. When meeting with the customer, a preliminary set of goals will be set for between the meeting date and the next available meeting opportunity, and the team will establish the goals they seem fit in order to meet the deadlines set by the customer.

14.2.4 Sprint Backlog

The team will decide what needs to be placed in the Sprint Backlog from the Product Backlog at the Sprint Planning session prior to the start of a new sprint. The backlog will be maintained through the Trello scrum board software using tags to separate backlog items based off task area and person.

14.2.5 Task Breakdown

Individuals will assign themselves tasks from the Sprint Backlog based off the technical area of the product they are working on. Individuals tasks will be marked with their account in the Trello software to show ownership of the task. Time spent on tasks will also be controlled through the Trello card system using a number in the story title to identify time spent.

14.2.6 Sprint Burn Down Charts

Seth Jaksik will be responsible for generating the burn down charts for the sprint. They will be able to access the total amount of effort expended by each individual team member through the Trello software on each members assigned cards. The burn down charts will follow the format below:

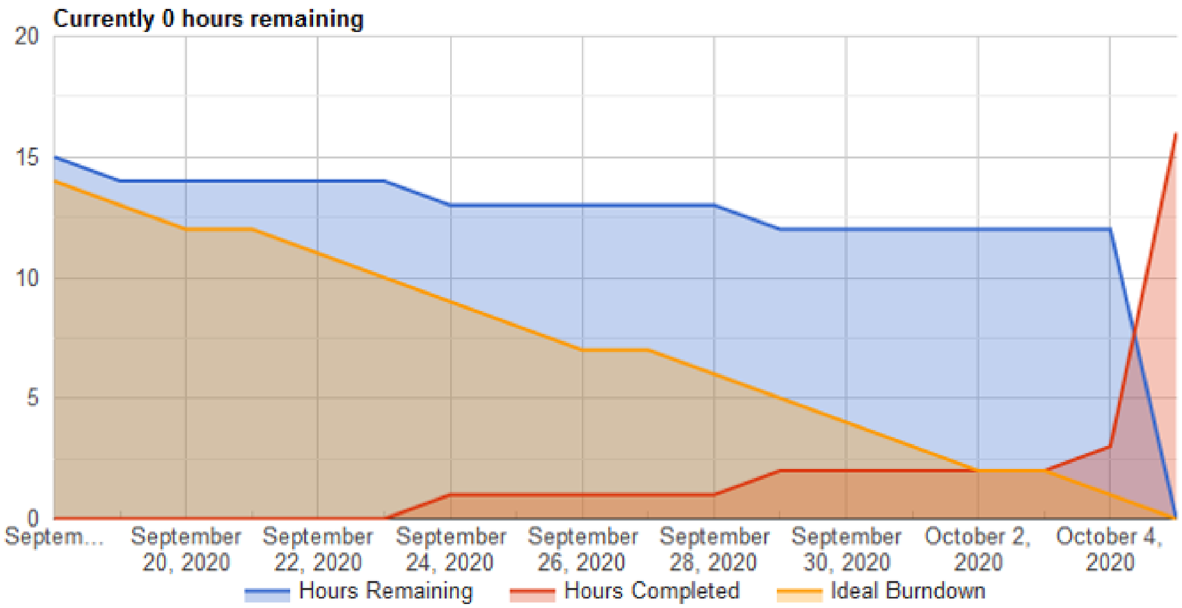

Figure 2: Example sprint burn down chart

14.2.7 Sprint Retrospective

The Sprint Retrospective will take place prior directly before the Sprint Planning session of the next sprint. The current Scrum Master will lead the Sprint Retrospective discussion as well as construct the charts from the discussion. Each Sprint Retrospective document will include our Sprint Goal, Product Backlog Items, the Sprint Burn Down Chart with an explanation, what we accomplished this sprint and how we did it, and a demonstration of something completed this sprint.

14.2.8 Individual Status Reports

Each member's status report will include what they have worked on since their last report, what they plan to work on for the next two days, and anything that is blocking their progress. Team members progress will also be tracked through the team's Trello board.

14.2.9 Engineering Notebooks

The engineering notebook will be updated at a minimum of 2 times a sprint by each team member. Each interval should include a minimum of 2 pages completed, and each interval will consist of a unique sprint. Each team member will be kept accountable through checks of the engineering notebook during Sprint Planning and Review sessions, as well as reminders throughout the duration of the sprint.

14.3 Closeout Materials

14.3.1 System Prototype

The final system prototype will include the IoT device, software to operate the device, and the router and sample devices to be triggered with the IoT hub. The final system prototype will be demonstrated in May 2021 at the Final Project Demonstration day either in person or virtually, depending on the current status of the Covid-19 pandemic and social distancing concerns. there will be a Prototype Acceptance Test (PAT) to be completed witht the customer prior to the Final Project Demonstration day in May 2021. There will be no off-site demonstration of the product.

14.3.2 Project Poster

The product poster will include the Team Name, Timeline, Team Members, Sponsor, and Abstract for the project, as well as supporting figures. The product poster will be in the standard poster size of 28x2in and be delivered before the Final Project Demonstration day in May 2021.

14.3.3 Web Page

The project web page will include the following elements:

- Team Name
- Timeline
- Team Members
- Sponsor
- Abstract
- Background
- Project Requirements
- System Overview
- Results
- Future Work
- Project Files
- References

The web page will be available to the public through the UTA CSE Senior Design blog. It will be updated throughout the project as major milestones are reached, and finally just before closeout.

14.3.4 Demo Video

Demo Videos will include demonstrations of our technology and progress throughout our respective sprints. They will show our developments and advancements in our software and hardware and will provide explanations of our difficulties and successes. B-Reel footage will include our current hardware and potentially a demonstration of our software. Videos will be approximately two minutes long as to allow adequate time to show and explain our progress.

14.3.5 Source Code

The source code for the project will be maintained using a Github repository. The customer will be provided binaries loaded onto the IoT device as well as the source code. The project will be open sourced to the general public using the MIT license. The license will be located in a single readme file in the root directory of the repository on Github.

14.3.6 Source Code Documentation

The project will include system documentation in each file using the Doxygen documenting standards. Doxygen will be used to generate the final documentation, which will be provided as a browsable HTML.

14.3.7 Hardware Schematics

The project will include networking schematics that identify the various networking components, such as the IoT device, router, and devices that need to be triggered.

14.3.8 Installation Scripts

The project will include instillation scripts in order to install the software onto newly made IoT devices in order to help the customer with instillation. The instillation script will consist of only one phase that initializes both the front-end GUI and the back-end networking components.

14.3.9 User Manual

The customer will be provided with a digital user manual in the form of a readme in the Github repository. The user manual will walk the user through the initial instillation of the software onto the IoT device, setting up the device as well as networking components, and give an overview of how to use the GUI to configure triggers.

References

[1] Elgato stream deck.

[2] Streamlabs. http://streamlabs.com. Accessed: 2020-10-04.

[3] Chris Mallett and S Gray. Autohotkey. *AutoHotkey Foundation, LLC, Charleswood, IN, USA*, 2009.

[4] Robinsh. Azure iot hub device streams. *Azure IoT Hub device streams* | *Microsoft Docs*.

[5] Jesse Yates. Stream processing with iot data: Best practices techniques. *Confluent*, Jun 2020.