USB / Bluetooth Media Controller

Interim Report

Name: Sam Dixon

Matriculation No: 40056761

Abstract

This document is an interim report summarising the current state of the "USB / Bluetooth Media Controller". The project consists of the research, development and testing of a microcontroller based media controller and a companion mobile application. The purpose of this report is to catalogue the project's various stages of development and to describe future action.

Table of Contents

1 Introduction

1.1 Background

The rationale for this project was simple - to provide a quick and efficient way for a user to alter the way media is played on a PC.

While many keyboards already offer inbuilt media keys, they can often be in locations that are difficult to access, or they require the usage of awkward key-combinations. These reasons make a stand-alone media controller an attractive option to users, as the controller can be placed in a location that is easy to reach, and a simple analogue interface ensures that the correct function is always selected.

While the technologies that allow users to experience their media have grown ever more advanced, the actual method of controlling this media remains relatively unchanged. The main benefits of an external controller include allowing a user to augment their media without losing access to the keyboard and mouse; haptic feedback on button presses; simplified controls (something which may be particularly useful for elderly users or children); and access to media remotely, or when the PC is locked.

Basing the system around an Arduino microcontroller allows for simple expansion of the project. For instance, a Bluetooth module will allow the media controller to be accessed remotely via a mobile application. The main benefit of this is that it allows a user to control their media remotely. At a party, for instance, the host's computer could be locked to prevent undue access to the system, but the host can still change the current song through the mobile application.

1.2 Aims and Deliverables

The original goal of the project was to create a USB/Bluetooth media controller and mobile application. While this remains the ultimate goal, additional functionality in the form of Windows short cuts or hotkeys is likely to be included in the final build.

The physical device will consist of an Arduino microcontroller with an individual button for each media function. These functions will be as follows - previous track; next track; play/pause; mute. Volume control will also be present in the form of two buttons (one to increase and the other decrease volume). A Bluetooth module will also feature, to allow a mobile device to connect to the system and control it remotely.

The mobile application's goal is to provide the same functionality as the hub, via a Bluetooth connection to the microprocessor. In addition to the media functions present on the hub, the application will include various Windows hotkeys, such as Lock PC and Log Off. At this point in time, it is currently unclear whether these functions will only be accessible via the app, or if the application will change the functionality of the physical buttons.

In order to use the app, the user must first connect to the media control hub via Bluetooth. The user will then have access to several labelled buttons that offer the same functionality of the hub.

The goal for the appearance of the app is to keep it as simple as possible, in order to reduce any potential learning curve involved in usage and keep the experience intuitive.

2 Current Status

2.1 Research and Analysis

From the preliminary research it was clear that this project was feasible, however, achieving the desired functionality without the necessary proprietary software running on the PC would make the task more complex.

By reviewing the datasheets for various microcontrollers [1] [2] and the projects others had produced with them [3], it was determined that an Arduino Leonardo had the capabilities to operate as a plug-and-play USB device. Had this not been the case, a script would have to be produced that would essentially 'listen' for serial input from the microcontroller and respond appropriately.

With the knowledge that a Leonardo could act as a suitable base for the project, further research as to how the microcontroller HID operations began. By examining the Leonardo’s HID, USBAPI, Keyboard and Mouse libraries [4]a better understanding of the device’s USB protocols was attained, as well as providing some base code for the bespoke Media Key library being produced.

Further understanding of how USB-HID devices operated was required to fully comprehend how the Leonardo communicates with a host PC. The official USB developers guidelines [5] provided a wealth of knowledge on the operation of USB devices, specifically HID Usage Tables [6]. In essence, a Usage Table is a list of all possible operations a USB HID device can perform. This list is sent to the PC when the device is plugged in and at any point, the device can request the host to carry out any of the functions provided.

The final step in producing a fully functional Media Key library was to acquire the HEX codes of each command that would be included in HID Usage Table. A three year-old personal blog post that had previously been dismissed for being too outdated provided a sample Usage Table for a host of different media commands [7]. The post in question had been analysed as potential solution to the project during the initial research phase. However, due to the age of the post and the fact the code was designed to interact with an IDE fourteen iterations out of date (as of time of writing), it was deemed unusable. Even a brief conversation on the official Arduino IRC with a lead developer determined that the libraries needed were no longer present in the IDE.

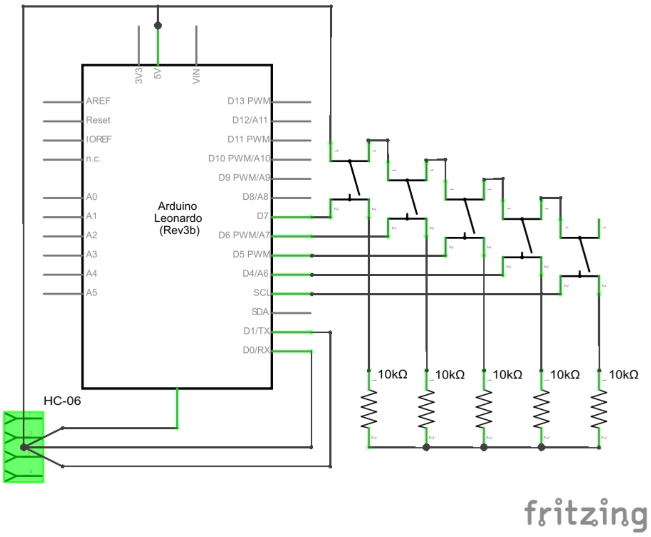

Figure 1- A circuit diagram of the complete system

After producing a working Media Library it was discovered that some of the HEX codes were outdated and were no longer supported by most media systems. This was a simple issue to solve, using the USB HID manual from earlier [6] allowed for quick look-up of the most update HEX codes and how they would be interpreted by the PC.

2.1 Practical and Experimental Work

With the second phase of research complete, the production of custom Media Key library was under taken and completed within the span of working week. This allowed for plenty of time to test, debug and fine tune the library against a number of media systems, including; Google Play Music, Windows Media Player, VLC, Winamp and Spotify.

With the successful completion of the required library, the first major Arduino Sprint has come to a close. As it stands at the time of writing, the system comprises of five push switches, each of which calls a different media control (namely previous, play, next, volume up and volume down). To ensure accurate readings from the switches, an external library was utilized. The Bounce2 library offers accurate edge detection and ensures that a function will only be called once per button press. While it could be said that utilizing an external library reduces the overall complexity of the system, the fact remains that main focus of the project is the media controls it offers. If time allows a bespoke debounce library may be produced but as it stands currently, the Bounce2 library provides the exact functionality required and allowed for more efficient testing.

2.3 Project Management

The previous report highlighted a list of milestones to be completed over the course of the project. The updated list can be seen below:

- ✓ Learn to correctly wire a microcontroller / breadboard device.
- ✓ Using a microcontroller as a basic human interface device. E.g. printing text in a document.
- ✓ Using a microcontroller to send media control signals to a PC. E.g. play/pause music.
- Add basic Bluetooth functionality.
- Develop an Android application to interface with a microcontroller.

A more detailed Gantt chart has been produced, to more display the time frame of the project accurately. The completed tasks are depicted in blue, while those yet to be started are in orange.

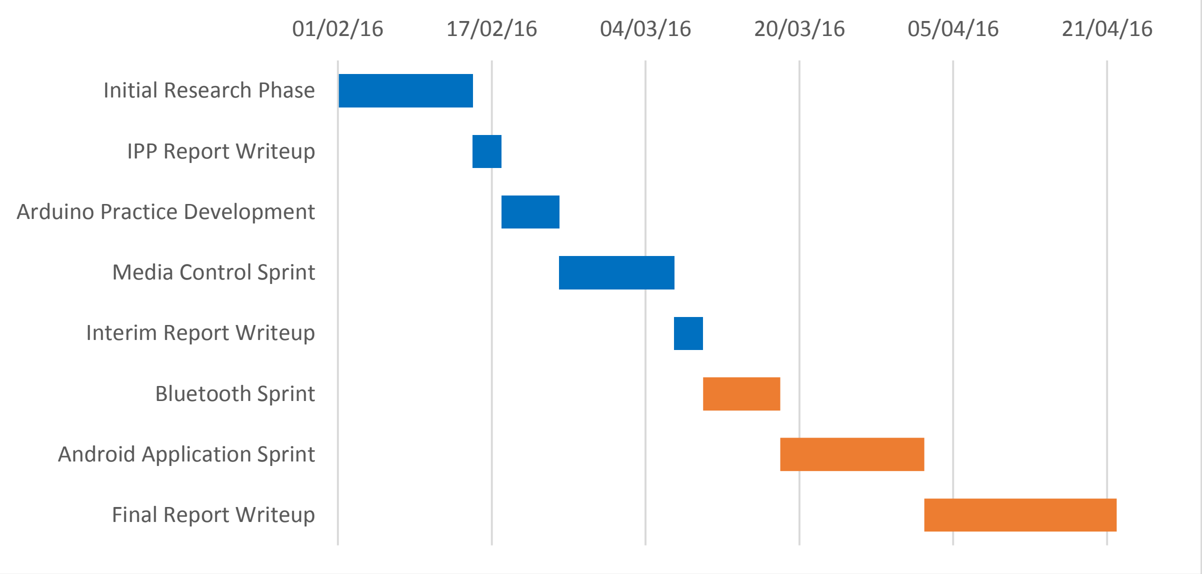

Figure 2 - A Gantt chart indicating the project timeline

As can be seen, the project is well underway and is on time to be complete for the deadline.

3 Future Programme of Work

3.1 Future Sprints

As one can see from the project timeline, two further sprints remain before completion of the project, namely the Bluetooth and Android Application sprints.

The goal of the next development sprint is to set up and configure basic Bluetooth communication with the Media controller. At the time of writing, the sprint has been delayed until the arrival of the required hardware, namely a HC-06 Bluetooth Module. Once the appropriate resources are present, the scheduled sprint should be relatively short and the project will move into its final phase.

Once the Bluetooth module has arrived and been configured, the development of the companion Android application can begin. Mobile development can often be a time consuming process. Fortunately, the application to accompany the media control is of a minimalist design, in the hopes of keeping the operation and development as simple as possible. As seen in the timeline, the application sprint has been scheduled to be the longest. This is to ensure that application can be toughly tested, resulting in an enjoyable user experience.

3.2 Final Report

Once all the design, implementation and testing of the system is complete, all that remains of the project is the final report. At this time, the final report consists of rough notes and diagrams and the final section headers are yet to be finalized. While the report itself may currently be in a state of disarray, the notes being taken at stage of the project are thorough and should allow for a concise and balanced summary of the project.

4 Issues and Concerns

4.1 Hardware Concerns

As it stands, the only issue in terms of hardware is the lack of it. Once the required HC-06 module arrives, it should be a simple task to get it configured, thanks to prior experience with this technology.

4.2 Software Concerns

As previously stated, application development can often be a time consuming affair. The time scale for the project has left ample time to produce an application for the device. However, if more time is required to complete the final report, the overall quality of the application may suffer.

5 Conclusion

5.1 Summary

The project is well under way and is on track to be completed to a high standard for the deadline. The only constraint that currently exists is the time required for the Bluetooth module to arrive. Once Bluetooth implementation is complete, the only concern will be the time required to complete the Android application. While a very basic application can be created in a short amount of time, the usability and features may suffer if any major road-blocks are met during the sprint.

Compared to the time and effort required for the research and development of the project, the final report and presentation should prove to be a relatively simple affair. The hard work put in at the beginning of the project should allow for fast and efficient production of the final documentation.

6 References

6.1 References List

[1] “Arduino - ArduinoBoardLenoardo,” Arduino, [Online]. Available: https://www.arduino.cc/en/Main/ArduinoBoardLeonardo. [Accessed 8 March 20016].

[2] “Arduino - ArduinoBoardUno,” Arduino, [Online]. Available: https://www.arduino.cc/en/Main/ArduinoBoardUno. [Accessed 8 March 2016].

[3] M. Heironimus, “Add USB Game Controller to Arduino Leonardo/Micro,” instructables, [Online]. Available: http://www.instructables.com/id/Add-USB-Game-Controller-to-Arduino-LeonardoMicro/. [Accessed 8 March 2016].

[4] “arduino/Arduino: open-source electronics prototyping platform,” Arduino, [Online]. Available: https://github.com/arduino/arduino. [Accessed 8 March 2016].

[5] “USB.org - HID Tools,” USB Implementers Forum, [Online]. Available: http://www.usb.org/developers/hidpage/. [Accessed 8 March 2016].

[6] “Universal Serial Bus (USB) HID Usage Tables,” 21 October 2004. [Online]. Available: http://www.usb.org/developers/hidpage/Hut1_12v2.pdf. [Accessed 8 March 2016].

[7] S. Jones, “Arduino Leonardo Remote Control and Multimedia Keys Support in 1.0.5 | Some Thoughts,” [Online]. Available: http://stefanjones.ca/blog/arduino-leonardo-remote-multimedia-keys/. [Accessed 8 March 2016].