Universal Game Controller

Neil Singh, Evan Lee, and Charles Van Fossan
ECE 445 Project Proposal - Spring 2017
TA: Eric Clark

1 Introduction

1.1 Objective

With the advancement of technology, the world of gaming is continuously expanding. This can be observed just by looking at the technical specifications of video game consoles over the years. One of the earliest video consoles, the Nintendo Entertainment System, started the industry out with a CPU that ran at 1.79 MHz [3]. When comparing it to the modern day version of the console, the Wii U, which clocks in at around 3 GHz, the difference is astounding [4]. While this is great for the progression of gaming, having new hardware released every couple of years starts to have its monetary impact. With each new console, an individual must buy new controllers that tend to be quite expensive. Unfortunately, there is no way to avoid this situation because previous controllers tend to be incompatible with new consoles. On top of that, with multiple consoles, there would be many different controllers that both take up space and time to acclimate to the change in controller layout.

Our goal is to eliminate this problem by providing a universal game controller that connects to many different consoles. This eliminates the need for the storage of the plethora of controllers for the various consoles and also the adaptation period. However, this ultimately allows consumers to only have to purchase one type of controller that can be used with all of their consoles and thus saving them money. We plan to develop a single controller that can be connected (wirelessly over bluetooth) to various dongles, which then can be plugged into the console themselves. In addition, we also intend to create an application for mobile devices such that the user can define custom mappings of the universal controller's buttons to the specific console’s. For right now, the only two consoles that will be supported are the Nintendo GameCube and Nintendo 64, but hopefully in the future more could be developed.

1.2 Background

For any avid gamer, having multiple systems are a must, but the expense of having four controllers for each might prevent some from realizing their console’s true potential. By having a single controller that could work on multiple systems, the cost of being able to utilize the console the way it is supposed to be greatly diminishes. Our controllers should not exceed the cost of a normal commercial controller and the various dongles should be as inexpensive as possible so that buying new dongles does not have a large financial impact on the consumer. This way,

consumers do not need to worry as much about the cost of getting the maximum amount of controllers for their consoles.

1.3 High Level Requirements

- Controller must control the console the same as commercial controllers for the GameCube and N64 consoles
- Controller must work wirelessly to connect to dongles that plug into the consoles
- Controller must be able to have custom button mapping profiles which are able to be set from another device such as a laptop or smartphone

2 Design

2.1 Block Diagram

There are two major components to our design: the controller and dongle. The controller will be powered by an onboard rechargeable battery and have a microcontroller unit (MCU) that will process all I/O between the physical buttons, vibration motor and the bluetooth adapter which will likely be built into the MCU. The dongle will have a similar MCU, powered by each console and data output through a single data pin. Lastly, there will be an application interface between the controller MCU and a smartphone that will allow the user to reprogram the mappings of each button.

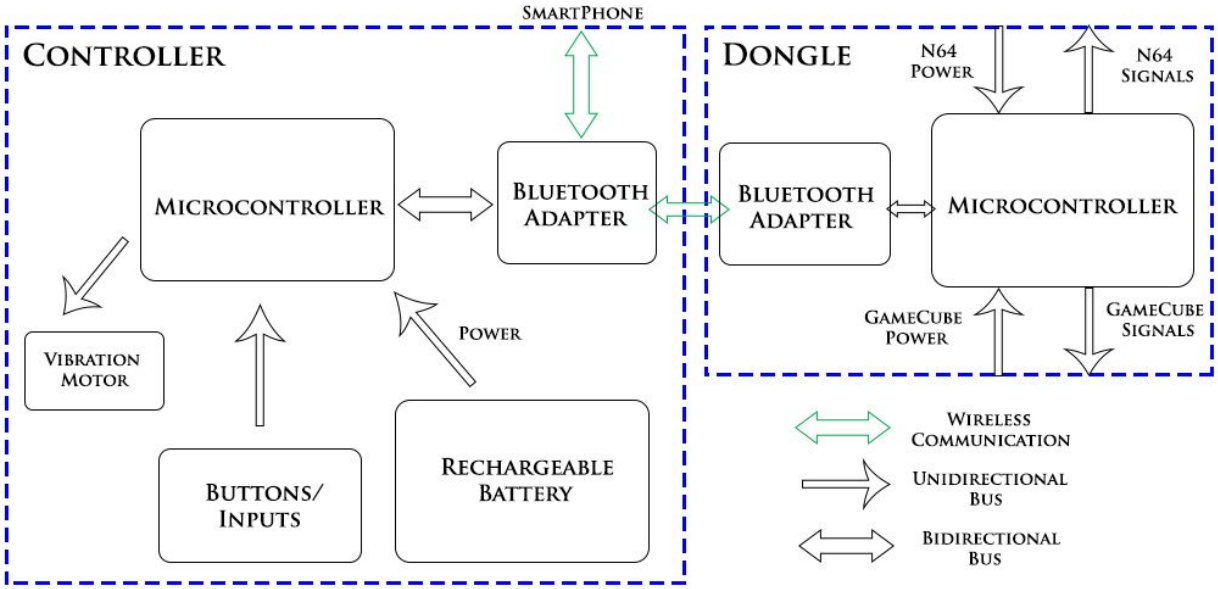

Figure 1

2.1.1 Microcontroller (Controller)

The MCU for the controller handles the interpretation of the inputs from the physical controller and relays the information to the bluetooth adapter when requested. It also allows the bluetooth adapter to be connected to either the dongle or the smartphone. In the case of a connection to the smartphone, the microcontroller enables the controller to have custom profiles dictated by the smartphone.

Requirement 1: Upon receiving a relay request from the bluetooth adapter (controller), the microcontroller must respond to the request with the correct input information.
Requirement 2: Upon receiving an update profile request from the bluetooth adapter (controller), the microcontroller must update its internal profile and respond with a confirmation to the smartphone.

2.1.2 Buttons/Inputs

The buttons and joysticks of the controller will be connected to the microcontroller and will receive inputs from the user and relay the information to the microcontroller. The console (through the bluetooth dongle) will poll the controller for information, to which the controller will respond with the states of the buttons and the positions of the joysticks. Therefore, the buttons and the joysticks must be able to continuously send their states and positions to the microcontroller to allow the microcontroller to quickly respond to the console's poll with the button/joystick information.

Requirement 1: The buttons and joysticks must be able to replicate the tactile feel of the original controllers.
Requirement 2: The buttons and joysticks must continuously send the button state and joystick position to the microcontroller.

2.1.3 Bluetooth Adapter (Controller)

The purpose of the controller's bluetooth adapter will be two-fold. First, it will be used to receive information from the dongle and transmit information regarding the states of the controller's buttons and the position of the joysticks to the paired dongle to respond to the console's poll. Second, it will used to receive button mapping settings from a smartphone. Because the controller's bluetooth will be connected to multiple bluetooth devices (dongle and smartphone), the controller's bluetooth adapter will act as the master in the piconet. We will set the bluetooth power to be 2.5 mW to have a 10 meter radius, which we think will be adequate for our purposes.

Requirement 1: The bluetooth adapter must be able to connect to the wireless dongle in order to send button information and receive polls and other information.

Requirement 2: The bluetooth adapter must be able to connect to a smartphone to receive button mapping settings.

2.1.4 Motor

The motor will be housed within the controller and can be turned on or off by the microcontroller, which receives this information from the dongle. The motor is used as part of the gaming experience.

Requirement 1: The motor must be able to be turned on and off by the microcontroller.

2.1.5 Rechargeable Battery

We are planning on housing our microcontroller in an official Nintendo Wii U Pro Controller (see Physical Design below), so we are planning on reusing the rechargeable battery that comes with it. This is a 1300 mAh battery that is charged through a Mini USB B connector cable.

Requirement 1: The battery must be able to be recharged and provide power to the rest of the controller

2.1.6 Bluetooth Adapter (Dongle)

This bluetooth adapter will be used to send polls and other useful information to the controller and to receive button and joystick information from the controller. As stated above, the dongle must be able to connect with the controller's bluetooth adapter, but will act as a slave device in the piconet.

Requirement 1: The bluetooth adapter must be able to send and receive data to and from the controller.

2.1.7 Microcontroller (Dongle)

This microcontroller will handle button and joystick information from the controller (from the bluetooth adapter) and will transmit this information directly to the console. In order to do this, it will need to map the controller information to the appropriate communication protocol based on what console it is connected to. The microcontroller must also use the same communication protocol to receive polls and other requests from the console and forward this information to the controller. The microcontroller must be able to discern what type of console (Nintendo64 or GameCube) it is connected to and use the corresponding communication protocol.

Requirement 1: The microcontroller must be able to determine what type of console it is connected to and use the appropriate communication protocol to communicate with it.

Requirement 2: The microcontroller must be able to receive polls and other types of information from the console and forward it to the controller.
Requirement 3: The microcontroller must be able to receive controller information from the controller and forward it to the console using the appropriate communication protocol.

2.2 Physical Diagram

Figure 2

We will be shelling a modern day game controller (Wii U Pro controller pictured) [7] for our implementation. The modern style controller is significantly different than both the N64 and Gamecube controller layout, but there are sufficient buttons on the contemporary controller for mapping all “vintage” controller buttons.

2.5 Risk Analysis

We have a heavy reliance on unofficial specifications for the protocols we will be using to communicate with the consoles. If the documentation we are using is incorrect in any way we will have to manually measure signals created by the controllers we are attempting to replace. This would require analyzing bit patterns on an oscilloscope to reverse engineer the protocols ourselves.

While this project is not necessarily hardware intensive, reverse engineering two separate console protocols could significantly add to the complexity and time of completion of our project. The interface between console and controller and the ability to swap protocols is what makes this project unique and it is imperative that our interface is accurate and complete.

3 Ethics and Safety

There is very little safety concern with our project as there are not any mechanical parts and we will be reusing the battery from the controller shell. Our main concern with this project is to avoid infringing on any patents or intellectual property ethical codes.

In reference to the ACM Code of Ethics, Section 1.6, “Computing professionals are obligated to protect the integrity of intellectual property. Specifically, one must not take credit for other's ideas or work, even in cases where the work has not been explicitly protected by copyright, patent, etc”[1]. The protocol of communication between the game controllers and game consoles was not created by us, nor did we reverse engineer the protocols ourselves. We will not take credit for the technology behind the protocols.

It is also important that we are careful in regards to voltages that we are manipulating on the dongles. It is possible to cause harm to both the Nintendo 64 and Gamecube through neglectful design of our devices. It is our responsibility to prevent damage to the property of others according to the ACM Code of Ethics, Section 1.2, “Avoid harm to others. "Harm" means injury or negative consequences, such as undesirable loss of information, loss of property, property damage, or unwanted environmental impacts”[2]. To prevent this we will most likely use a microcontroller that can not output more than 5V DC.

4 References

[1]"ACM Code of Ethics and Professional Conduct," in *acm.org*, 2017. [Online]. Available: https://www.acm.org/about-acm/acm-code-of-ethics-and-professional-conduct#sect1. Accessed: Feb. 9, 2017.

[2]"IEEE code of ethics," in *ieee.org*, 2017. [Online]. Available: http://www.ieee.org/about/corporate/governance/p7-8.html. Accessed: Feb. 9, 2017.

[3] P. Diskin, "NES Documentation," in *NESDev*, 2004. [Online]. Available: http://nesdev.com/NESDoc.pdf. Accessed: Feb. 7, 2017.

[4] "Wii U System Specs," in *Nintendo Today*, NintendoToday, 2011. [Online]. Available: http://nintendotoday.com/wii-u-system-specs. Accessed: Feb. 7, 2017.

[5] "Bluetooth basics," in *SparkFun*. [Online]. Available: https://learn.sparkfun.com/tutorials/bluetooth-basics. Accessed: Feb. 8, 2017.

[6] "Nintendo Gamecube Controller Pinout,". [Online]. Available: http://www.int03.co.uk/crema/hardware/gamecube/gc-control.html. Accessed: Feb. 7, 2017.

[7]Alibaba, "Wii U Pro Controller," in *alibaba.com*, 2016. [Online]. Available: https://guide.alibaba.com/shop/2014-new-black-classic-dual-analog-wireless-bluetooth-remote-u-pro-game-controller-gamepad-for-nintendo-wii-u_55640912.html. Accessed: Feb. 9, 2017.