Universal Game Controller

ECE 445 Mock Design Review Document

Evan Lee, Peter Van Fossan, and Neil Singh
Group 48
TA: Eric Clark
2/22/2017

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.

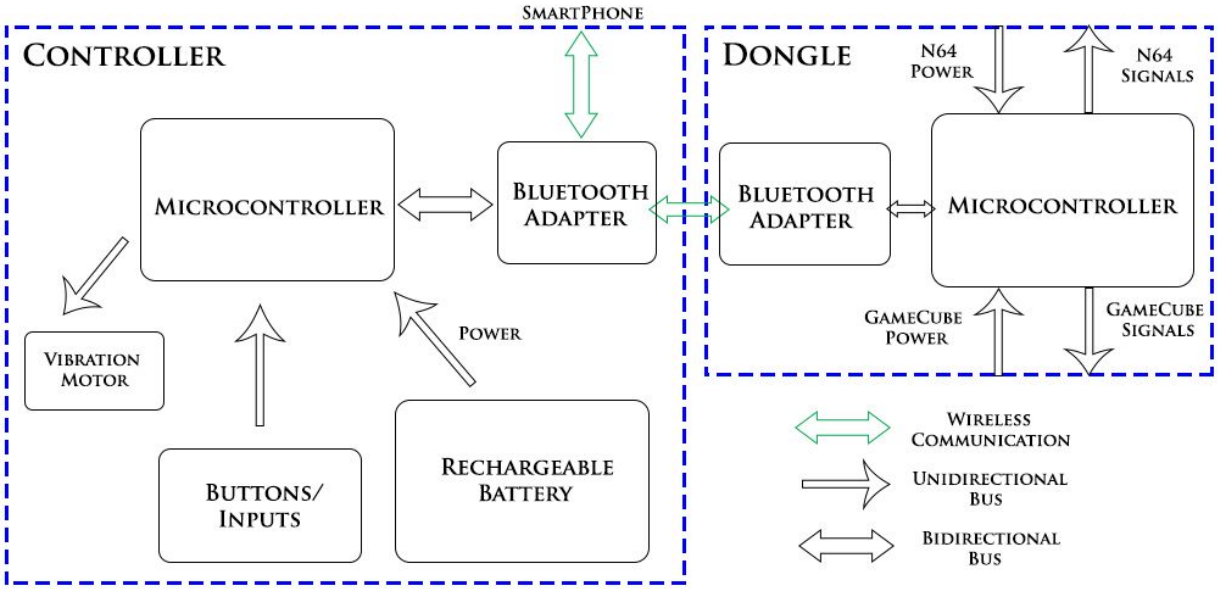

Figure 1

2 Circuit Schematic

This is a schematic of our joystick, which uses two potentiometers for two axises of freedom. Namely, moving the joystick along the X-axis corresponds to one potentiometer, and moving the joystick along the Y-axis corresponds to the other. On top of this, there is a push button in the joy stick, so that you are able to push down on the joystick, which is controlled by a standard switch.

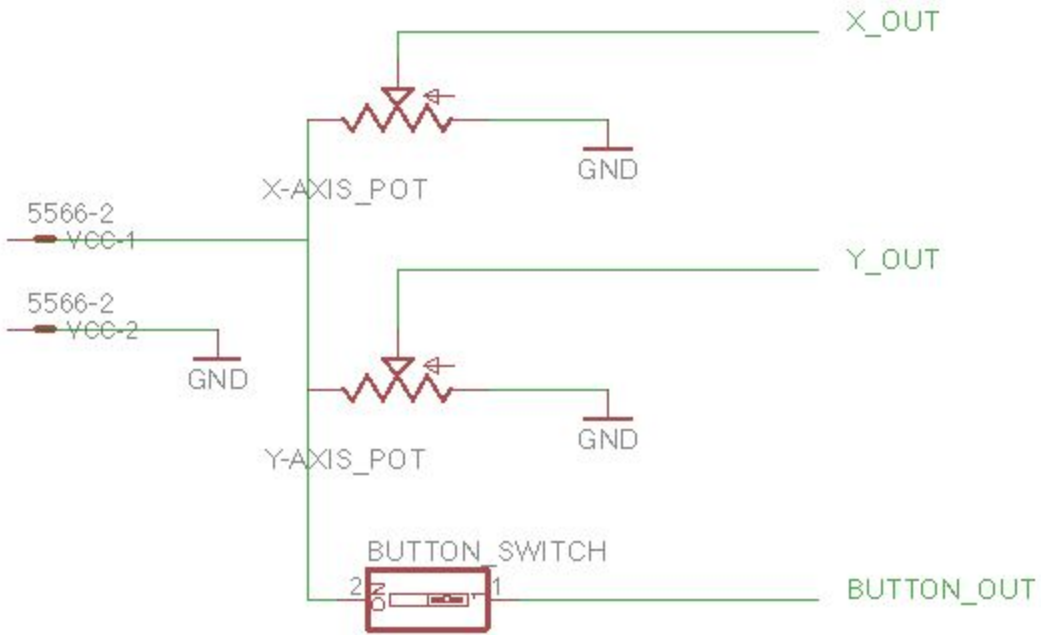

Figure 2

3 Calculation

The microcontroller must be able to have microsecond precision to allow supporting the controller protocols. The protocols have a bitrate of 250Kb/s, which translates to 4 microseconds per bit [1]. However, due to the protocols using an inferred clock from the data line, a low bit is denoted as 3 microseconds of low and 1 microsecond of high, while a high bit is denoted as 1 microseconds of low and 1 microsecond of high. Thus, our microcontroller must have 4 microseconds / 4 = 1 microsecond of precision. This is possible because the microcontroller we have selected has a 24 MHz timer, which would translate to a precision of 1 / 24MHz = about 42 ns [2].

4 Plot

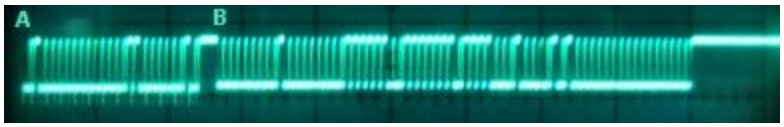

Figure 3 [1]

This is a plot of the 24 bit command word (A) sent by the GameCube console, as well as the 64 bit response (B) from the controller. You can see that a high bit is represented as 1 low bit then 3 high bits, and that a low bit is represented as 3 low bits and 1 high bit. Also it shows that when the response is finished, the line remains high to signal that there is no more data needed. This is because all bits start by going low for at least one cycle, and therefore when the line goes low, the console and controller will know that data is being transmitted.

5 Block Description (with R&V) - Buttons / Inputs

We will be using the Wii U Pro Controller for our controller shell.

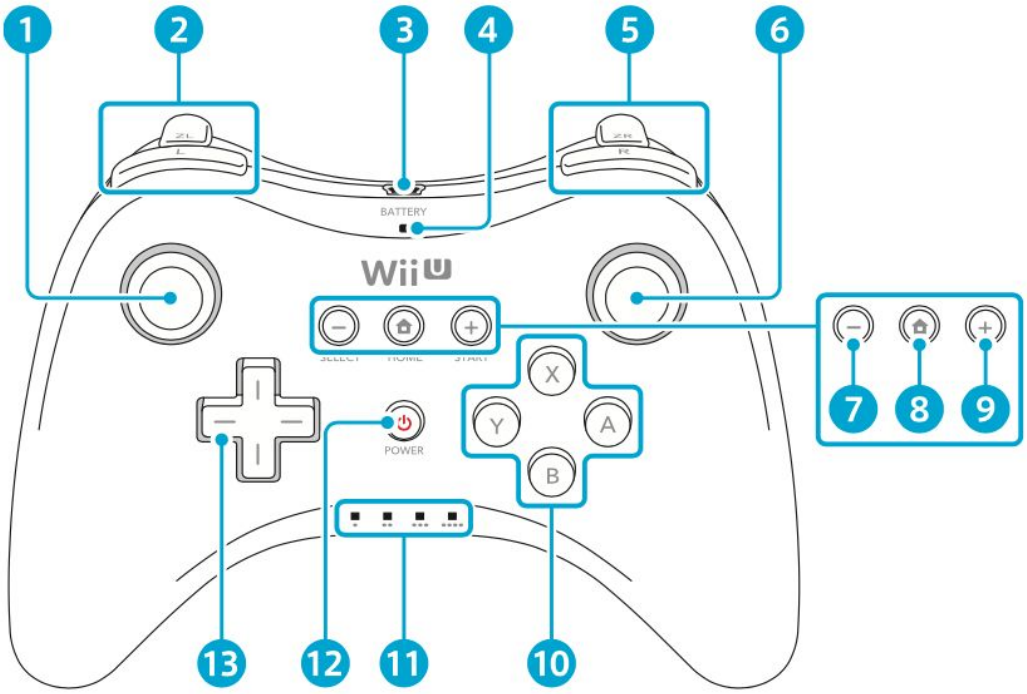

Figure 4 [3]

All the inputs feed into the microcontroller, separated by only either a potentiometer or digital button. The left analog stick (1 in Figure 2) and the right analog stick (6) will each be connected to a dual axis

potentiometer that adjusts two voltages based on how the analog stick is moved. The ZL trigger (top of 2) and the ZR trigger (top of 5) will be connected to single axis potentiometer that will adjust a single voltage based on how far down the trigger is. The directional pad's (13) four directions will be connected as digital buttons, along with the A, X, Y, and B buttons (10), the select, start, and home buttons (7, 8, 9 respectively), the L and R buttons (bottom of 2, bottom of 5), and power button (12). The LED's (11) will be connected to digital output pins on the microcontroller, separated by a resistor. The battery LED (4) will be connected to a multi-colored LED, separated by a resistor.

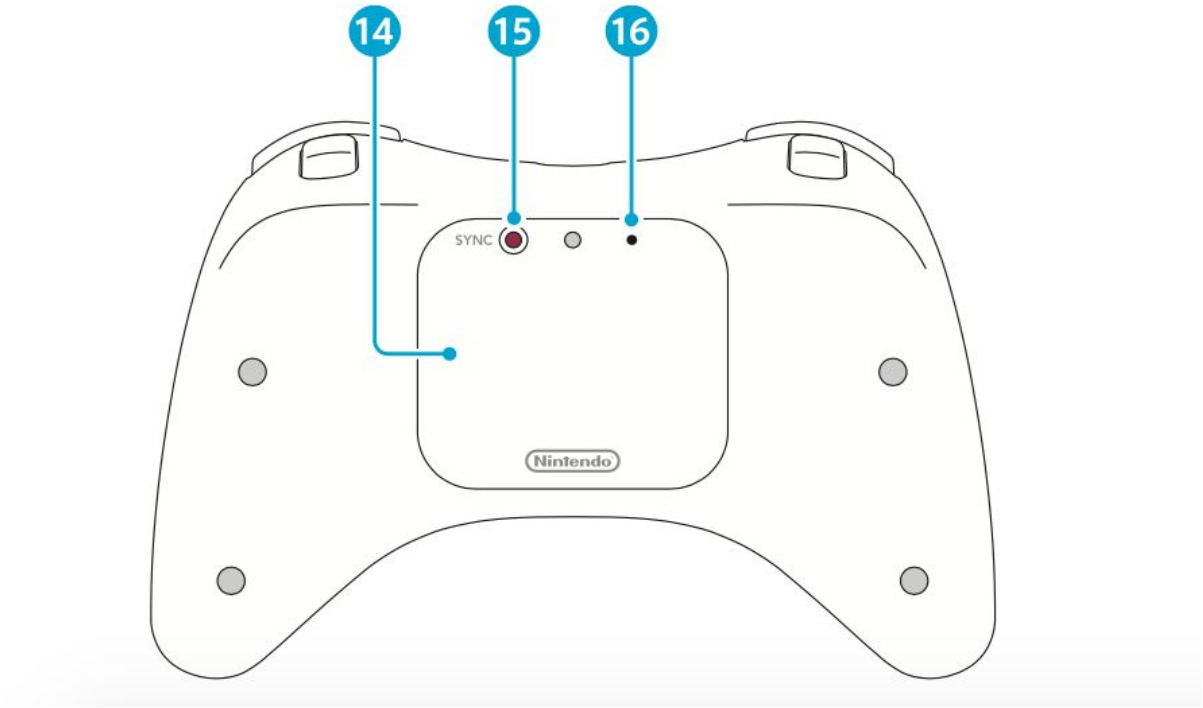

Figure 5 [3]

The sync button (15 on figure 3) will be connected by a digital button to the microcontroller, as well as the reset button (16).

Requirements	Verification
1. The joysticks must be able to send a variable amount of voltage ranging from 0-3.53V based on their positions to the microcontroller. Pushing down on the joysticks must also send a "high" voltage signal to the microcontroller. 2. The buttons must be able to send a voltage signal of approximately 1.5V to the microcontroller.	1. a. Measure the output voltages of the joysticks connected to the protoboard using an oscilloscope and ensure that the voltages are within 0-3.7V when moving. b. Measure the voltages of the joysticks connected to the protoboard using an oscilloscope and ensure that the voltage

	signals are within 5% of 1.5V when pressed. 2. Measure the voltage outputs of the buttons using an oscilloscope and ensure the voltage outputs are within 5% of 1.5V when pressed.

6 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. However, 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” [4]. To prevent this we will use a microcontroller that can not output more than 5V DC.

7 Citations

[1] James, "Nintendo Gamecube controller Pinout," 2004. [Online]. Available: http://www.int03.co.uk/crema/hardware/gamecube/gc-control.html. Accessed: Feb. 17, 2017.

[2] T. Instruments, "CC2640 SimpleLink Bluetooth Wireless MCU Datasheet," in *Texas Instruments*, 2016. [Online]. Available: http://www.ti.com/lit/ds/symlink/cc2640.pdf. Accessed: Feb. 20, 2017.3

[3] Nintendo, "Wii U Operations Manual," in *Nintendo*. [Online]. Available: https://www.nintendo.com/consumer/downloads/wiiu_operations_manual_en_la.pdf. Accessed: Feb. 20, 2017.

[4] "IEEE IEEE code of ethics," in *IEEE*, 2017. [Online]. Available: http://www.ieee.org/about/corporate/governance/p7-8.html. Accessed: Feb. 15, 2017.