Project 1 - PC Compatible Wii Guitar

Project Summary:

This project aims to create a device that will take the controller data from a Wii Guitar, sans Wii Remote, and transmit that data to a computer as an HID device. This is to avoid using Third-Party Drivers that are not Windows certified to facilitate the communication between a Wii Remote and a Windows PC. In addition, this device will be able to communicate with the computer using Bluetooth or USB.

Engineering Requirements:

- The controller can connect to PC via Bluetooth and USB without the use of unverified drivers.
- The controller will able to last at least 8 hours on battery power.
- The input delay of the controller is low enough that there is no significant delay between physically pressing a button and the input being recorded by the computer.
- The circuit will be assembled on a custom PCB and verified by my project mentor

Testing Requirements:

- Connect the controller to the PC and test that all inputs register correctly.
- Turn on the controller, connected via Bluetooth, and record the time it takes for the system to shut off.
- The controller will be tested and verified by at least two other students and they will comment on how responsive the controller feels.
- PCB Layout will be approved by my project Mentor (Don Heer).

Project Design:

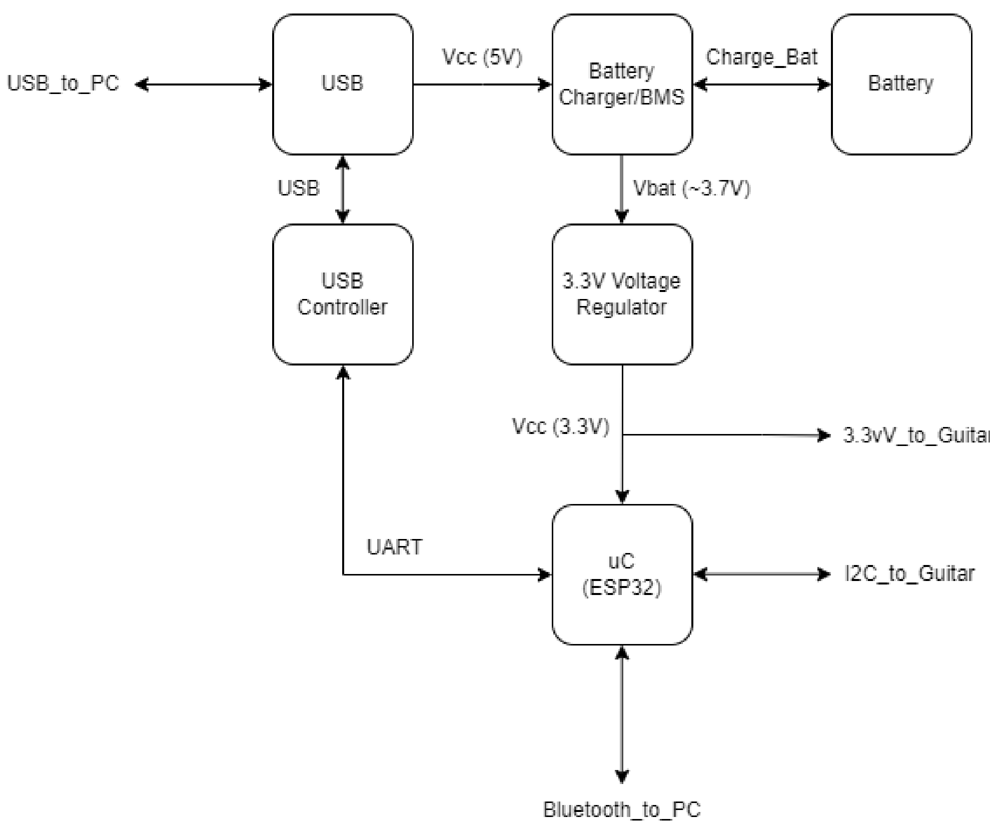
A lot of information exists on how the Wii works including how the Wii Remote communicates with extension devices such as the Nunchuk or the Wii Guitar. The communication protocol used is a 3.3V I2C signal(s) with nearly all extension devices having an address of 0x52. By default, the data is encrypted but more ‘modern’ extension devices, such as the Wii Guitar, can be set in a decrypted communication mode using the following I2C sequence:

Start, 0xF0, 0x55, Stop, Start, 0xFB, 0x00, Stop

The data is formatted into 6 bytes with the following bit structure with the default bit value being 1 for all of the buttons, changing to 0 when the button is pressed:

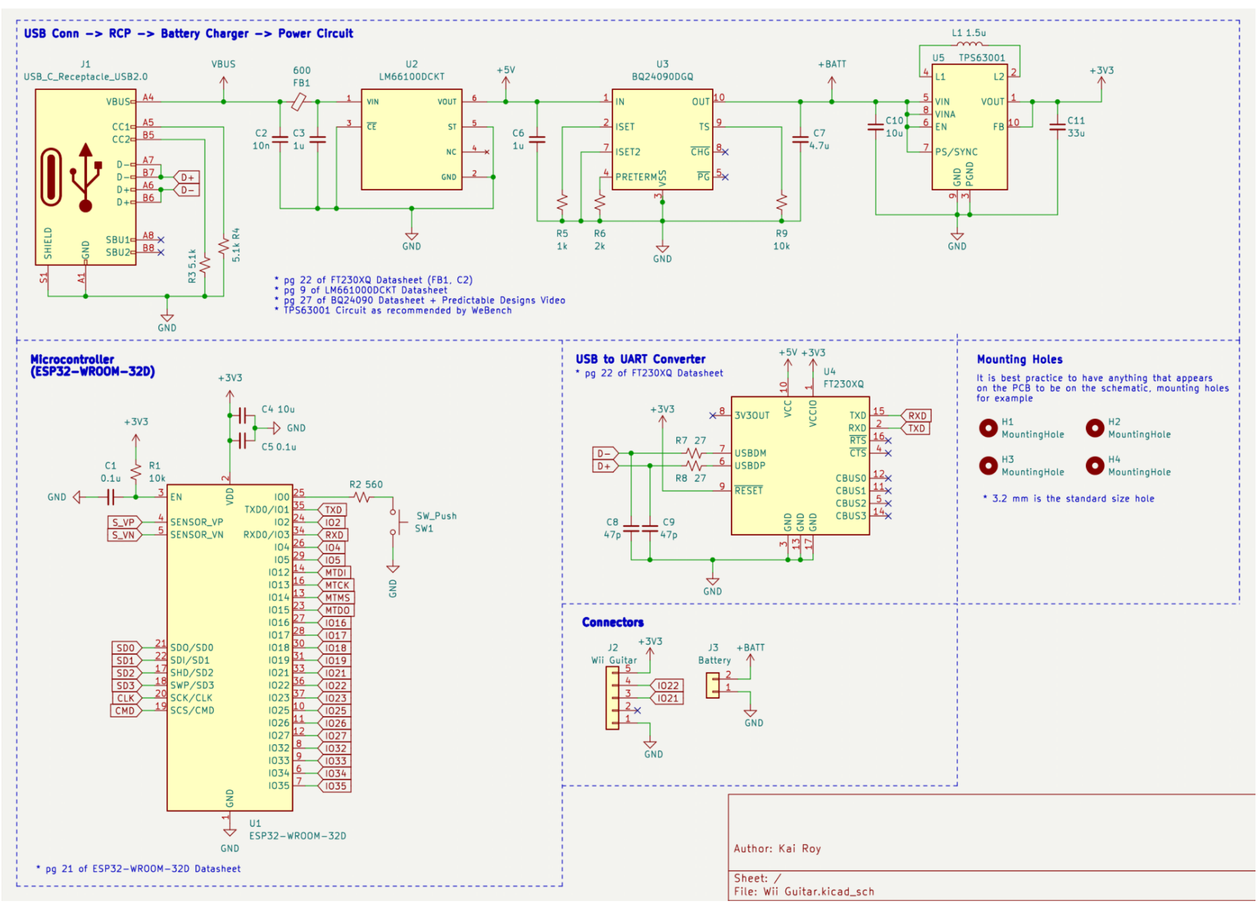
	Bit							
Byte	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
0	GH3	GH3	**SX**					
1	GH3	GH3	**SY**					
2	0	0	0	**TB**				
3	0	0	0	**WB**				
4	1	**BD**	1	**B-**	1	**B+**	1	1
5	**BO**	**BR**	**BB**	**BG**	**BY**	1	1	**BU**

SX and SY refer to the x and y-axis of the analog stick.
WB refers to the analog whammy bar.
BD and BU refer to the strum bar.
BO, BR, BB, BG, and BY refer to the Fretbuttons.
B- and B+ are the final two buttons on the
TB refer to the analog touch bar which is only found on some Wii Guitars

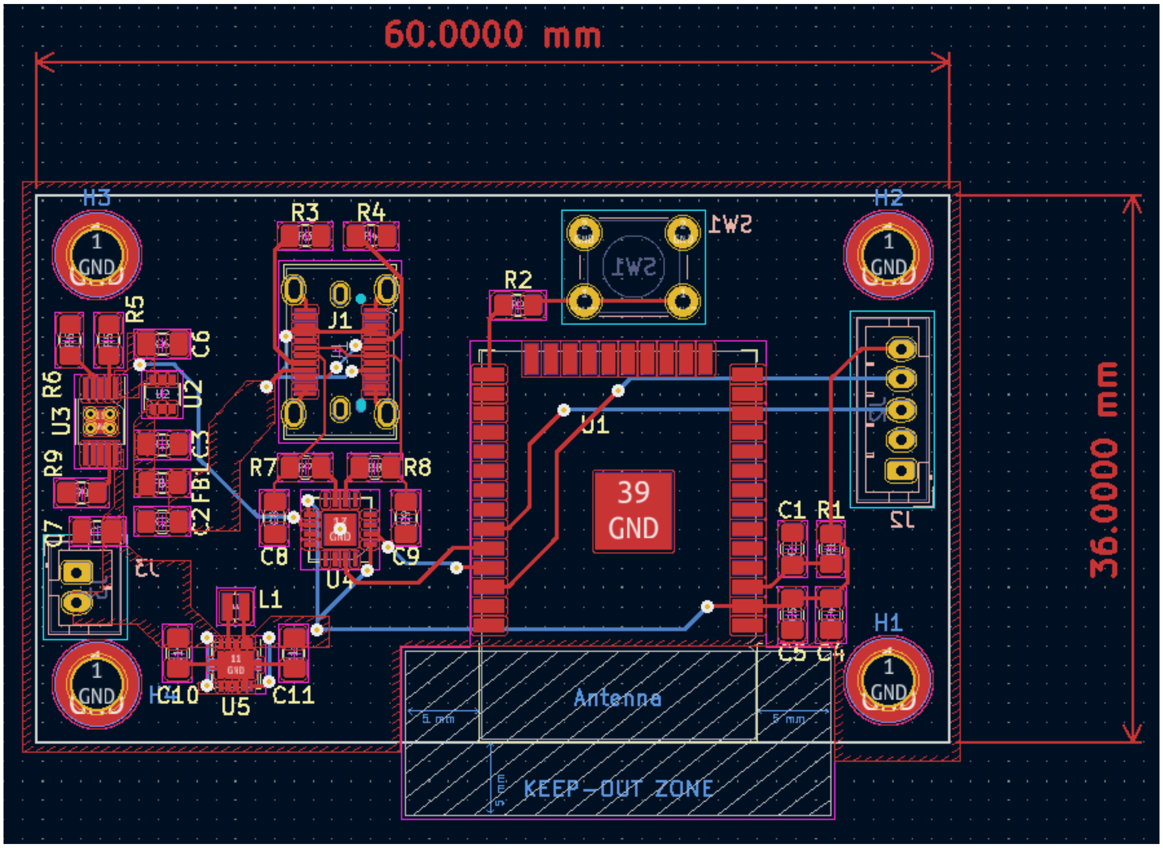


The block diagram above shows the basic outline for the circuit that will take in the data from the Wii Guitar and output via Bluetooth. The necessary power systems for battery power are also included.

The ESP32, specifically the ESP32-WROOM-32D module, was chosen for the microcontroller because of its inbuilt Bluetooth Low Energy (BLE) Capability as well as my prior experience with designing custom boards for it from my Senior Design Class. My previous design already had the USB Controller (USB-to-UART Converter) and the ESP32, so the only thing that needed to be added was a Battery Management System (BMS) and a 3.3V Regulator. While the old 3.3V regulator Circuit would have worked, I did not know that at the time so I used TI's WeBench to make a circuit capable of inputting 3.5V-5V and outputting a stable 5V. The circuit that was recommended used the TPS63001. The BQ24090 was chosen for the BMS circuit as one of the guides that I found used that IC for their own custom microcontroller. Another circuit that was added to the schematic was a reverse current protection circuit using the LM66100.



KiCad schematic of the project board



KiCad PCB Layout of the project Board

I chose to use a Vertical USB-C connector for two reasons. One is because USB-C is becoming much more common. The second reason is so that in the future, I can attach the board to a piece of acrylic and have the connector flush or near flush with the acrylic once the board is mounted. This idea of mounting is also why I have the programming button and JST connectors (for the battery and connection to the Wii Guitar) on the bottom of the board. Since the USB-C connector is an SMD component, I decided to have all of the SMD components on the same side, but in order for the USB-C to be placed correctly, I needed the other connectors to sit on the other side. Since this was being assembled by hand having all the SMD components on the same side would make it easier to assemble.

Three libraries are needed to program the board. The first is the Wire library which is needed to communicate with the Wii Guitar via I2C. The second library is LemmingDev's ESP32-BLE-Gamepad library. This library makes it very easy to turn an ESP32 into a BLE Gamepad that can connect and send data to the computer. The final library that is needed is a dependence for the ESP32-BLE-Gamepad, the NimBLE-Arduino Library.

Status at the end of the Term/406 Course:

By the end of the term, I have met three of the four requirements for this project. The only requirement I was not able to meet was the PCB requirement. This is for two reasons. One, the requirement says I will get the design verified by Don, which I did not do. The second reason is that the board that I ordered did not work. The reason is because of a bridge between 3.3V and GND caused by a 3.3V via connecting to the GND plane. I had run a DRC Check on the Board and fixed all the issues that had come up. I believe that this via was accidentally moved after that and the planes were not refilled causing an intersection between 3.3V and GND. This occurred right next to the Voltage Regulator circuit, making it very difficult to cut the traces and splice in a wire. While I did attempt to fix this and splice in a wire, I sadly do not have the materials or tools to make the necessary splice. I did have a backup plan in the form of my Senior Design Tech Demo Board and a battery charger module from Adafruit. Since the Tech Demo Microcontroller is nearly identical to the board that I designed, the code I made would have functioned on the board if I had caught that mistake.

With my backup system, I was able to demonstrate Bluetooth connectivity to the PC on battery power as well as a low response time between pressing a button on the Wii Guitar and the input registering on a windows computer. These are demonstrated in the videos attached to the project showcase page for this project.

What I would do differently/Future Plans:

Most of the issues that I have faced with this project and my second project this term came from poor planning and scheduling. Most of the work on all of my projects did not start until midway through the term. With three projects, I could not spend the necessary time focusing on each project in high detail. Because of this, I made mistakes, such as the mistake with the via being moved. In the future, I will be sure to run a DRC immediately before exporting or ordering any boards in the future.

My future plans for this project include recording the board that I had designed and fully implementing it into both of my Wii Guitars. While I may make some changes to the board, I will definitely be making an acrylic mounting plate that will fit where the Wii Remote is supposed to be in the Guitar as well as properly measuring the current draw of the system and getting an appropriately sized battery.