Spring 2023

Fingerless Xbox Controller

Brooke Snow

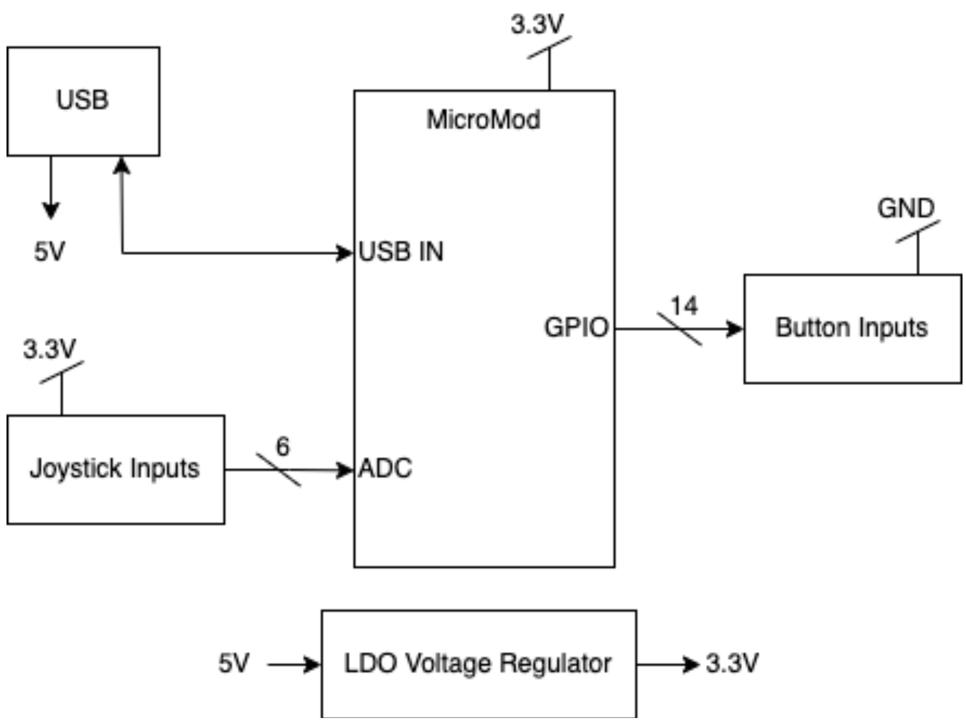
FINGERLESS XBOX CONTROLLER

Brooke Snow

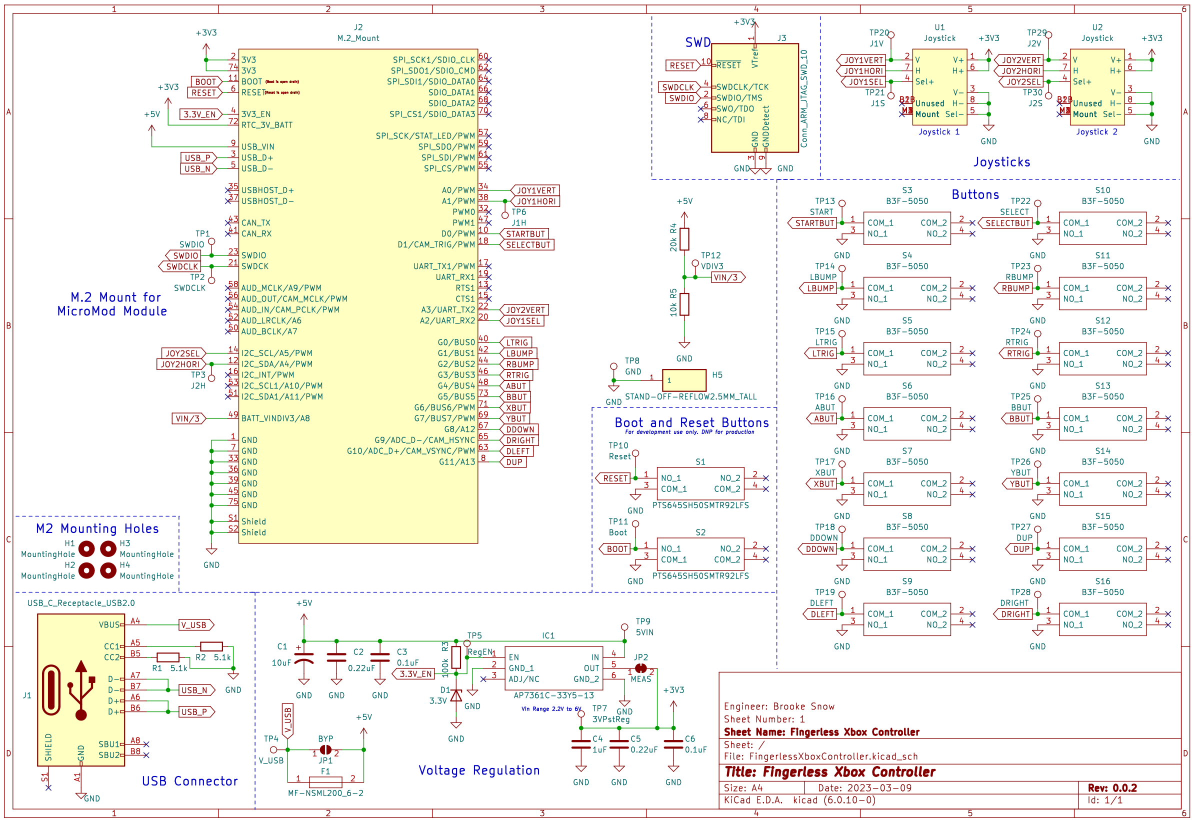
2023

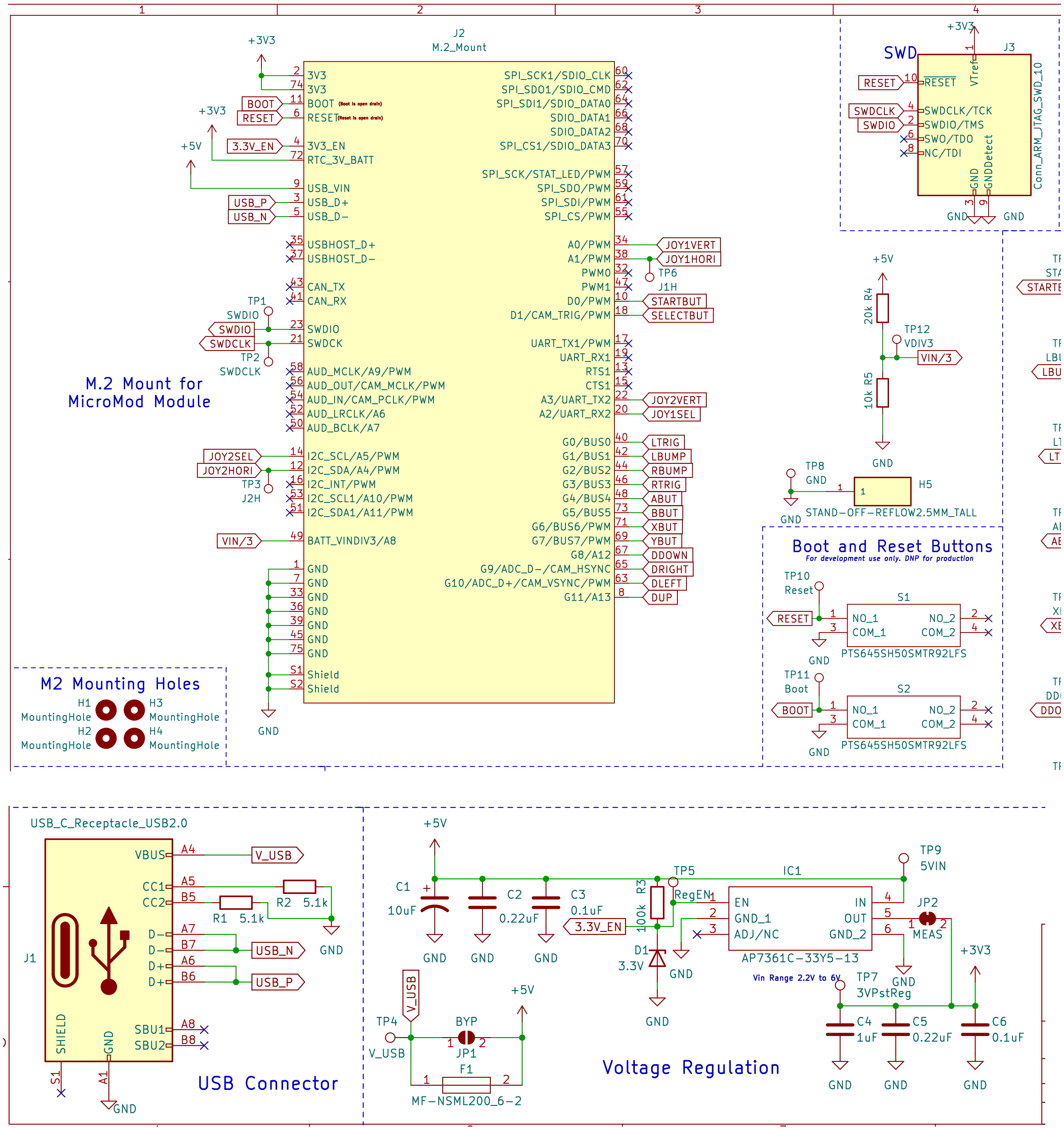
System Design

The Fingerless Xbox Controller (FXC) aims to make gaming more accessible to those who lack the full use of all their fingers by providing a better layout for the inputs on an ordinary Xbox controller. The FXC communicates with a computer that operates on Windows OS via a USB 2.0 and needs a USB-C cable to make that connection. That signal goes directly to the MicroMod, and the 5V supply from the USB gets run through a linear regulator to become a regulated 3.3 volts. That supplies power to the MicroMod and all of the buttons and joysticks. The buttons are attached to an internal pullup resistor in the MicroMod, as are R3 and L3 in the joysticks. Pushing a button or joystick pulls the GPIO (or ADC in the case of the joysticks) to ground, giving an active low signal. The joysticks x and y axes are based off potentiometers in the joysticks that act as variable voltage dividers and send that voltage to the ADC of the MicroMod. The MicroMod takes these signals from the user inputs and translates them into Xinput standard USB communication, which then gets sent to the computer to process. All these inputs are laid out on one flat surface so that you do not need to hold the FXC while playing with it.

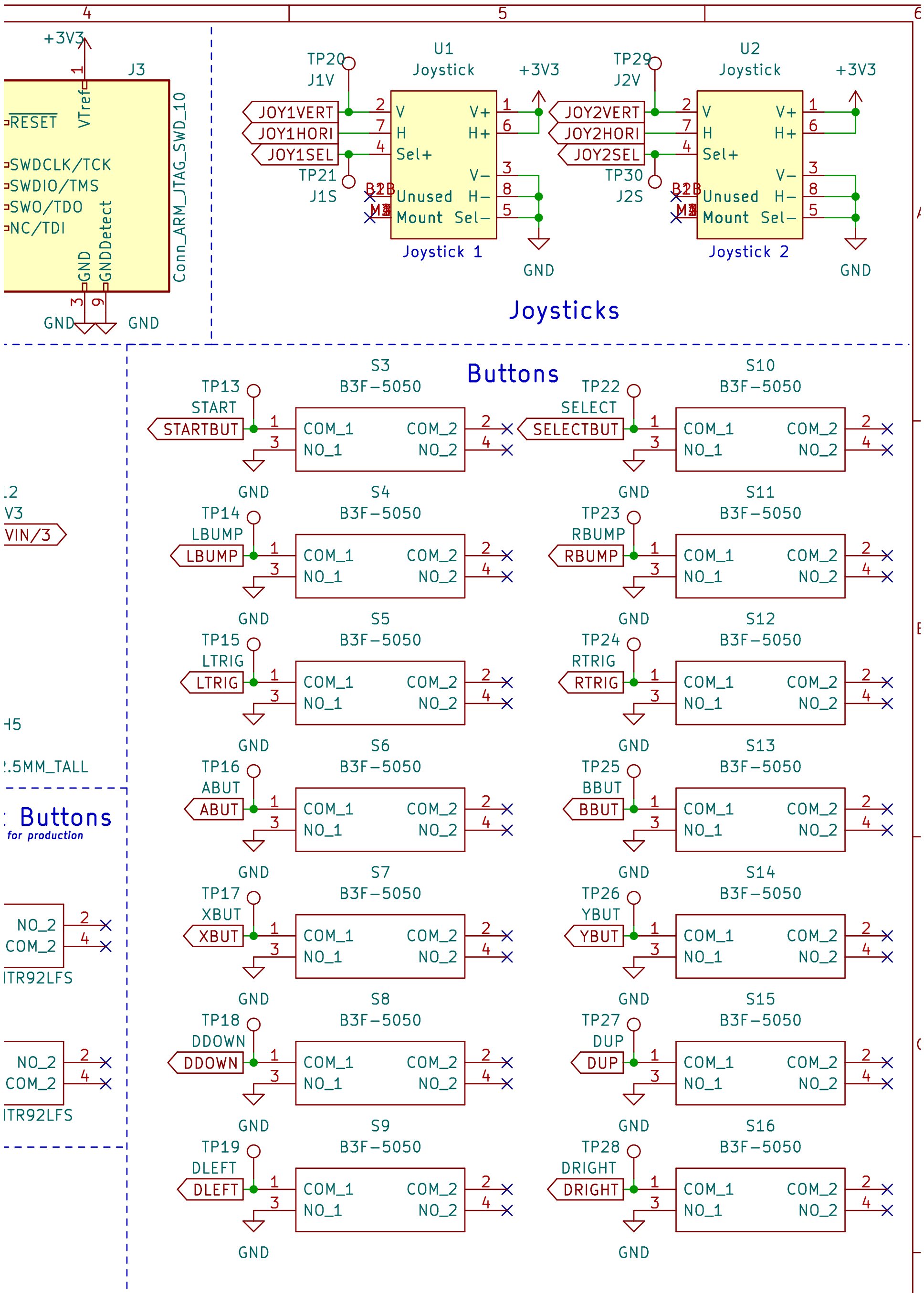


System Schematic





M.2 Mount for MicroMod Module
J2
M.2_Mount
3V3
BOOT
RESET
3V3_EN
RTC_3V_BATT
USB_VIN
USB_D+
USB_D−
USBHOST_D+
USBHOST_D−
CAN_TX
CAN_RX
SWDIO
SWDCK
AUD_MCLK/A9/PWM
AUD_OUT/CAM_MCLK/PWM
AUD_IN/CAM_PCLK/PWM
AUD_LRCLK/A6
AUD_BCLK/A7
I2C_SCL/A5/PWM
I2C_SDA/A4/PWM
I2C_INT/PWM
I2C_SCL1/A10/PWM
I2C_SDA1/A11/PWM
BATT_VINDIV3/A8
GND
Shield
SPI_SCK1/SDIO_CLK
SPI_SDO1/SDIO_CMD
SPI_SDI1/SDIO_DATA0
SDIO_DATA1
SDIO_DATA2
SPI_CS1/SDIO_DATA3
SPI_SCK/STAT_LED/PWM
SPI_SDO/PWM
SPI_SDI/PWM
SPI_CS/PWM
A0/PWM
A1/PWM
PWM0
PWM1
D0/PWM
D1/CAM_TRIG/PWM
UART_TX1/PWM
UART_RX1
RTS1
CTS1
A3/UART_TX2
A2/UART_RX2
G0/BUS0
G1/BUS1
G2/BUS2
G3/BUS3
G4/BUS4
G5/BUS5
G6/BUS6/PWM
G7/BUS7/PWM
G8/A12
G9/ADC_D−/CAM_HSYNC
G10/ADC_D+/CAM_VSYNC/PWM
G11/A13
JOY1VERT
JOY1HORI
STARTBUT
SELECTBUT
JOY2VERT
JOY1SEL
LTRIG
LBUMP
RBUMP
RTRIG
ABUT
BBUT
XBUT
YBUT
DDOWN
DRIGHT
DLEFT
DUP
JOY2SEL
JOY2HORI
VIN/3
USB_P
USB_N
+3V3
+5V
M2 Mounting Holes
H1 MountingHole
H2 MountingHole
H3 MountingHole
H4 MountingHole
SWD
J3
Conn_ARM_JTAG_SWD_10
VTref
RESET
SWDCLK/TCK
SWDIO/TMS
SWO/TDO
NC/TDI
GNDDetect
R4 20k
R5 10k
TP12 VDIV3
TP8 GND
H5
STAND−OFF−REFLOW2.5MM_TALL
Boot and Reset Buttons
For development use only. DNP for production
TP10 Reset
TP11 Boot
S1
S2
NO_1
COM_1
NO_2
COM_2
PTS645SH50SMTR92LFS
USB_C_Receptacle_USB2.0
J1
VBUS
CC1
CC2
D−
D+
SBU1
SBU2
SHIELD
R1 5.1k
R2 5.1k
V_USB
USB Connector
C1 10uF
C2 0.22uF
C3 0.1uF
R3 100k
TP5 RegEN
D1 3.3V
IC1
AP7361C−33Y5−13
EN
GND_1
ADJ/NC
IN
OUT
GND_2
Vin Range 2.2V to 6V
TP9 5VIN
JP2 MEAS
TP7 3VPstReg
C4 1uF
C5 0.22uF
C6 0.1uF
TP4 V_USB
BYP JP1
F1
MF−NSML200_6−2
Voltage Regulation



Engineer: Brooke Snow
Sheet Number: 1
Sheet Name: Fingerless Xbox Controller

Sheet: /
File: FingerlessXboxController.kicad_sch

Title: Fingerless Xbox Controller

Size: A4	Date: 2023-03-09	**Rev: 0.0.2**
KiCad E.D.A. kicad (6.0.10-0)		Id: 1/1

PCB Design

Figure 1: Top View of Entire PCB, Top copper prominent

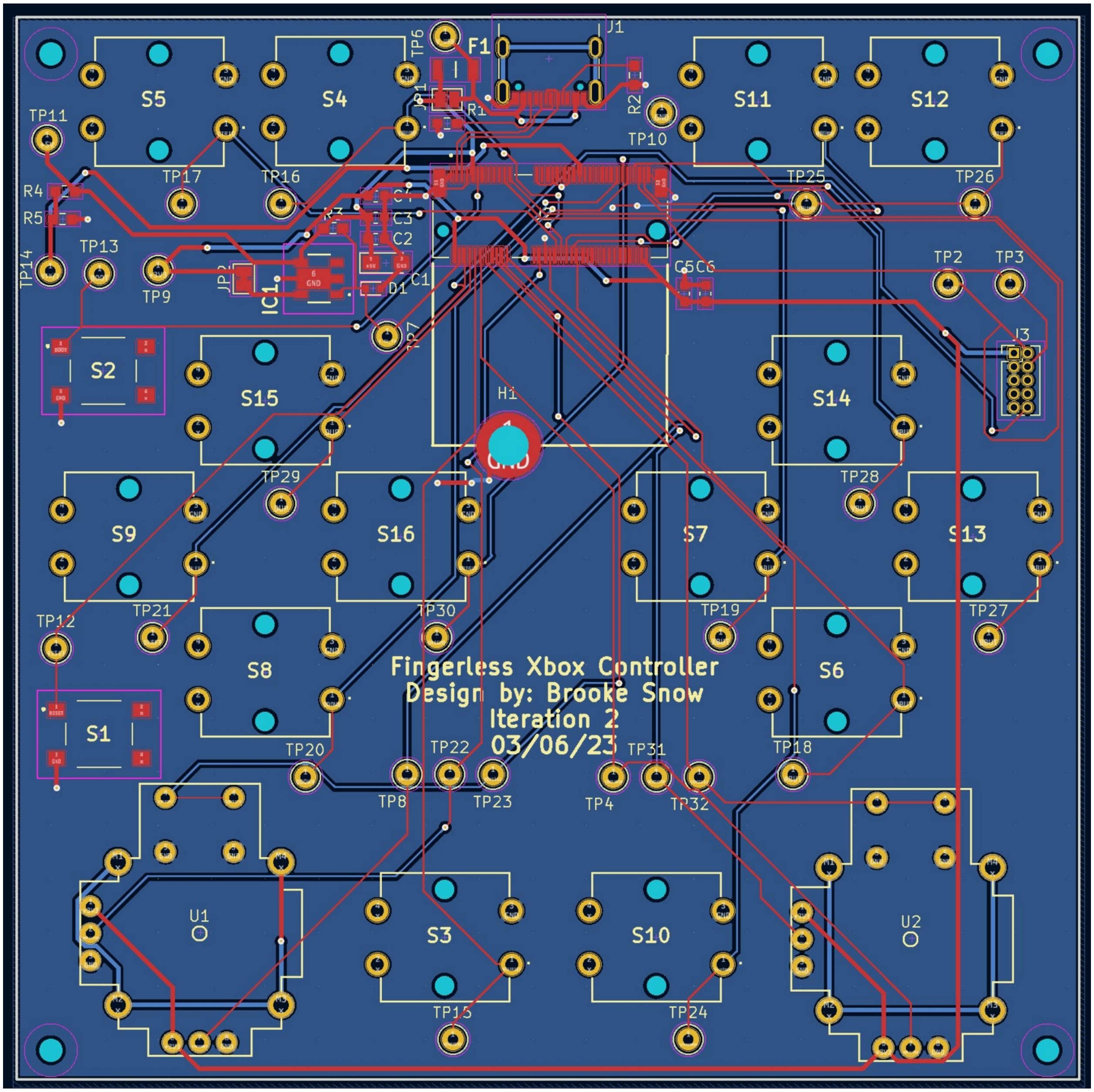


Figure 2: Flipped view of Entire PCB, Bottom copper prominent

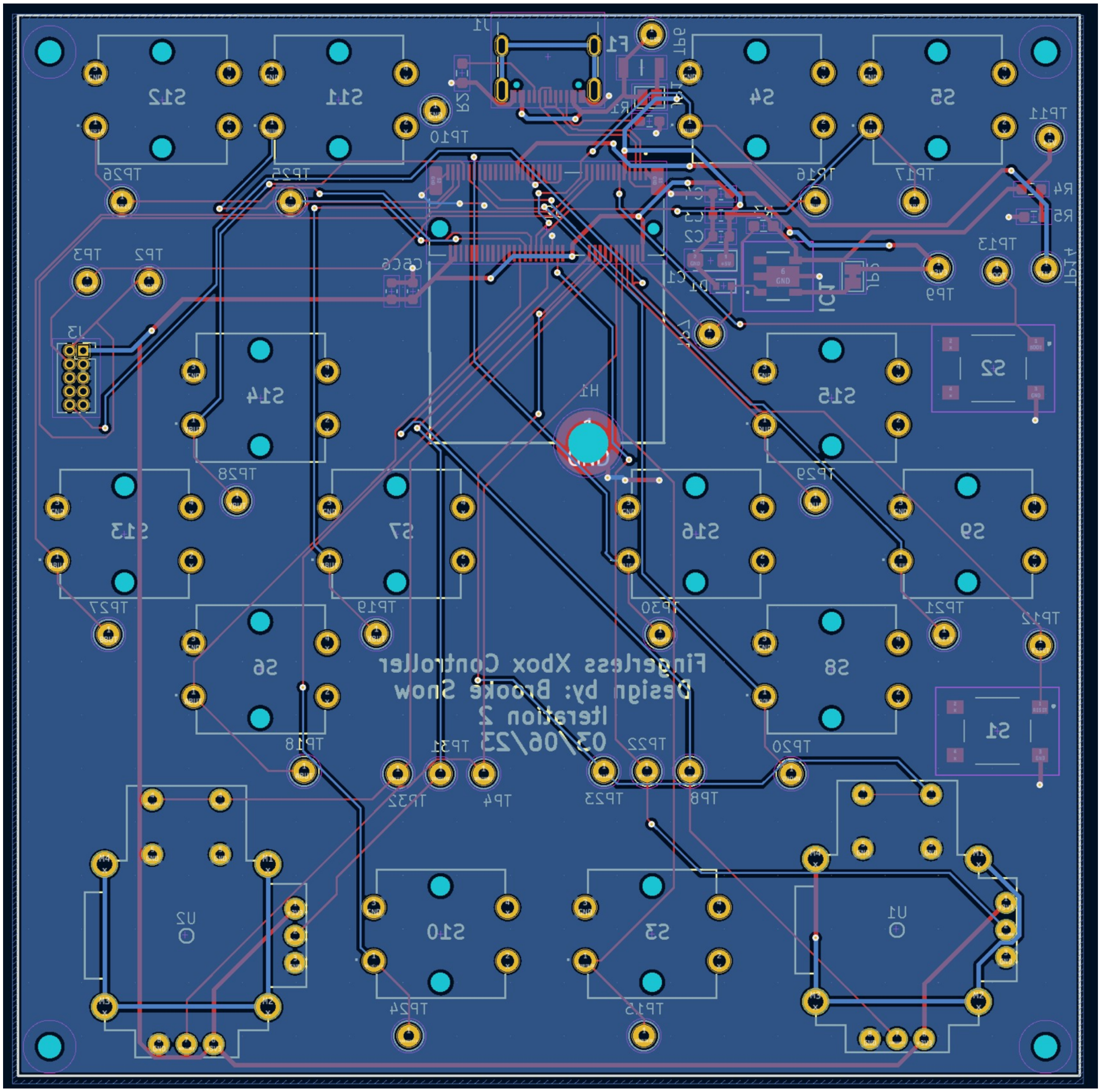


Figure 3: Zoomed in view of USB-C header, fuse, cut-trace, and USB-C resistors wiring.

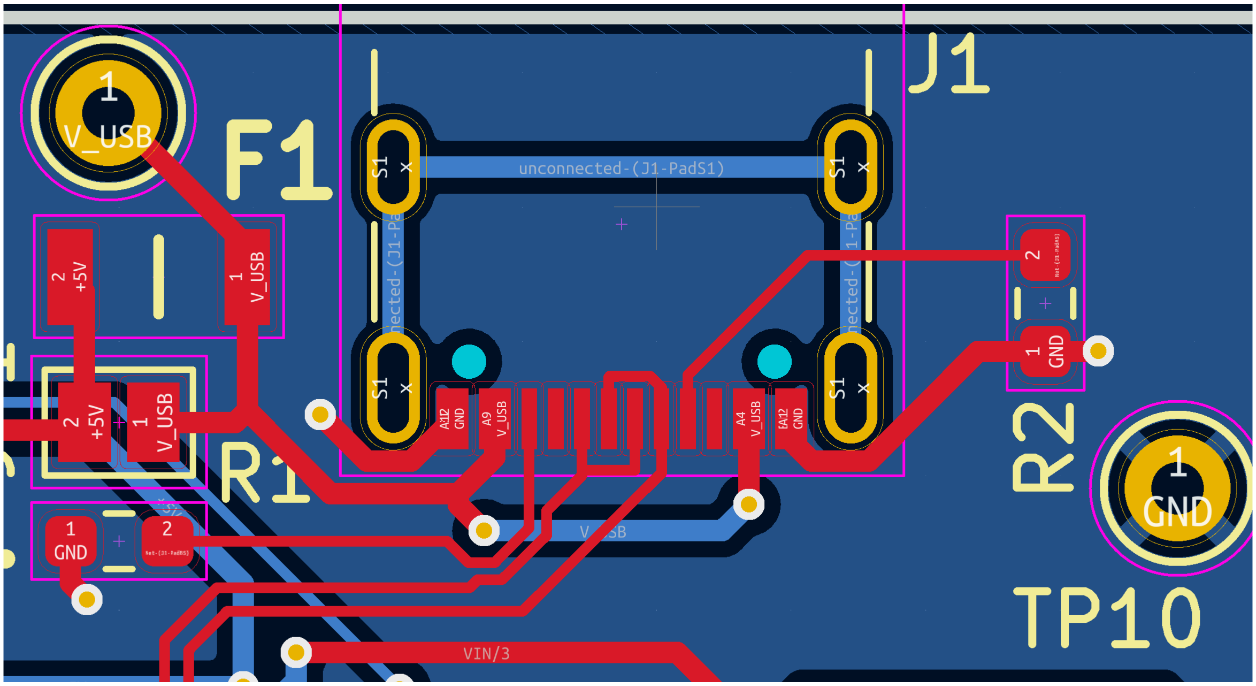


Figure 4: Zoomed in view of M.2 Mount for MicroMod

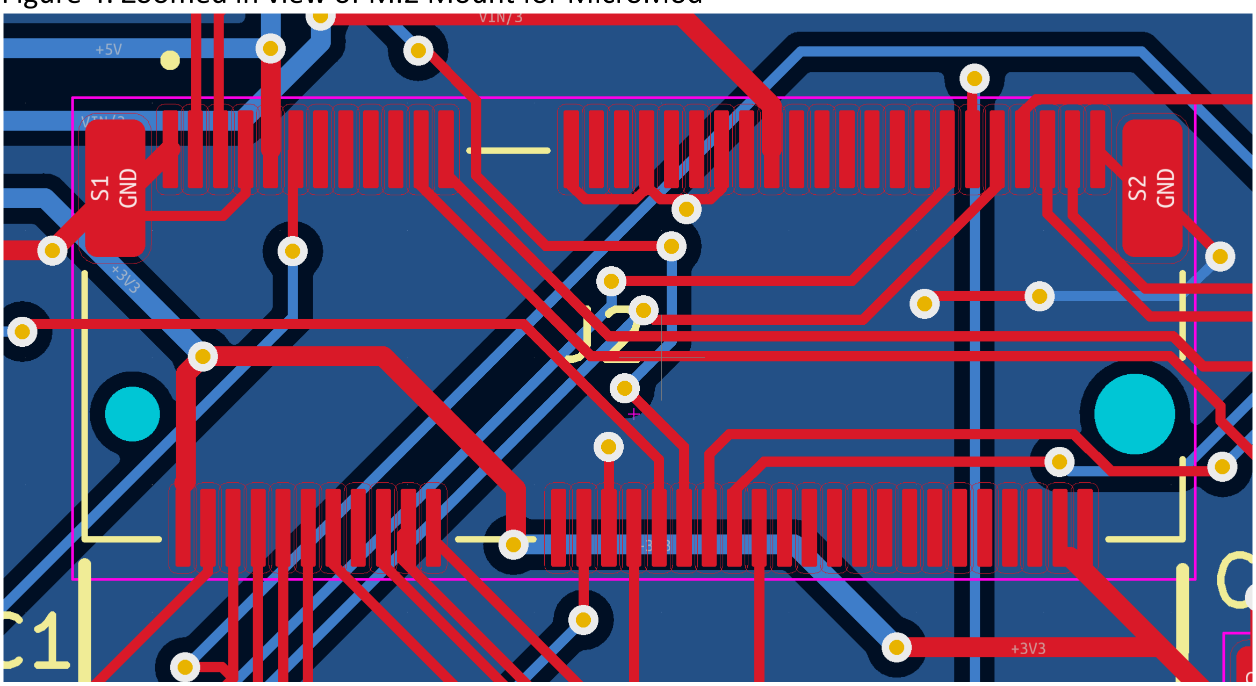


Figure 5: Zoomed in view of LDO and surrounding circuitry to set correct output voltage.

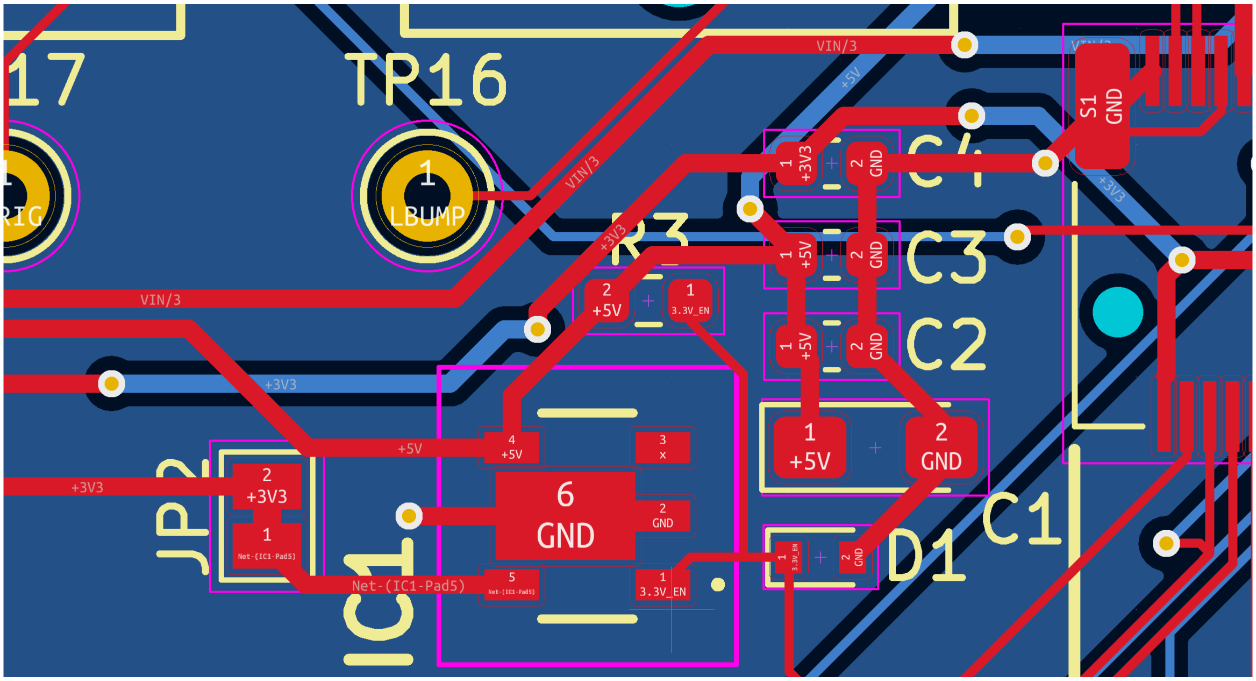


Figure 6: Zoomed in view of example buttons, SWD, and through-hole setup for M.2 Mount

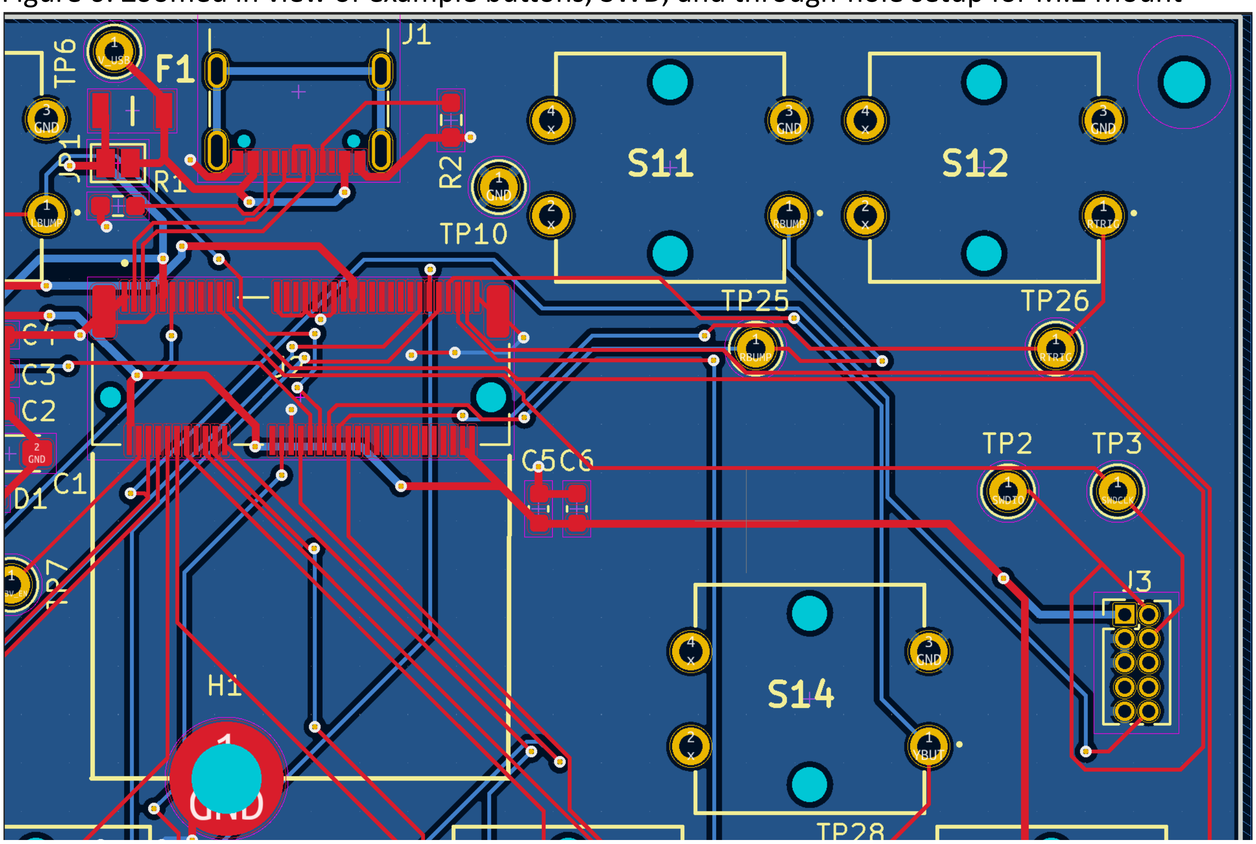


Figure 7: Zoomed in view of Reset button, other buttons nearby, and joystick.

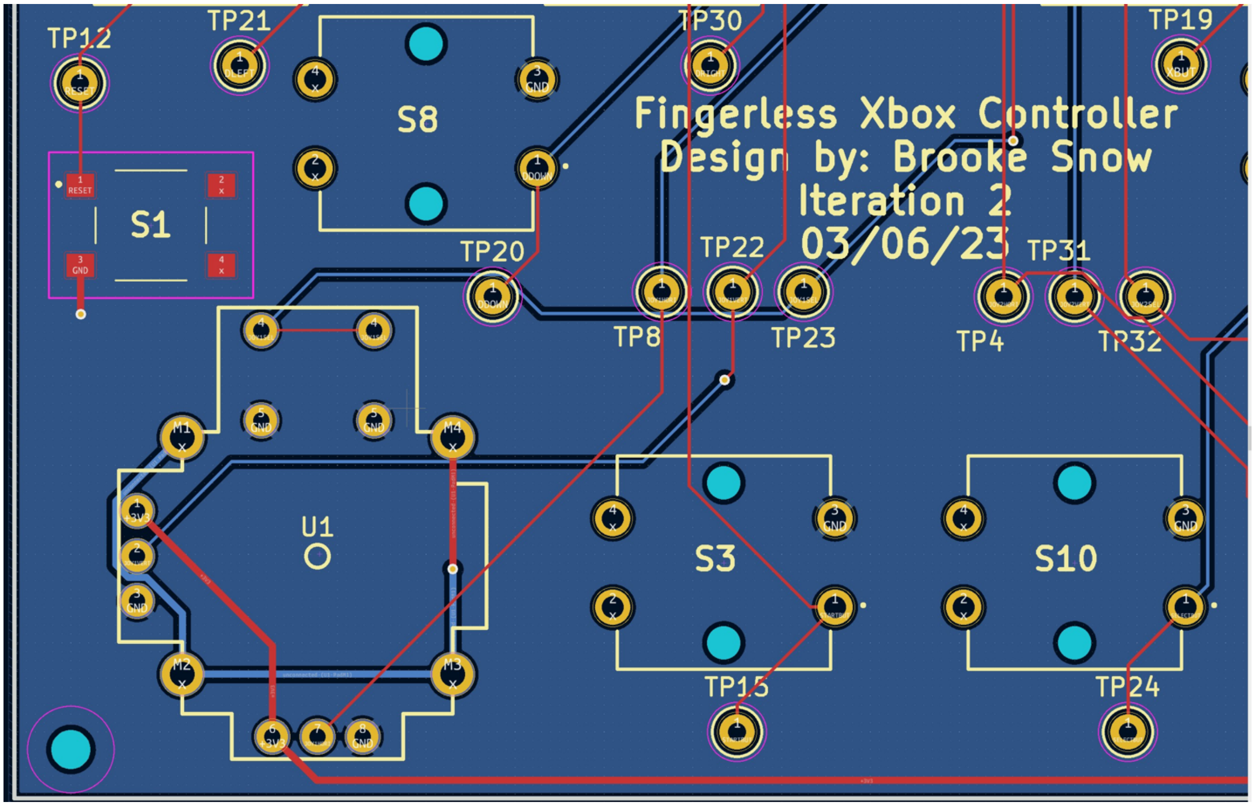
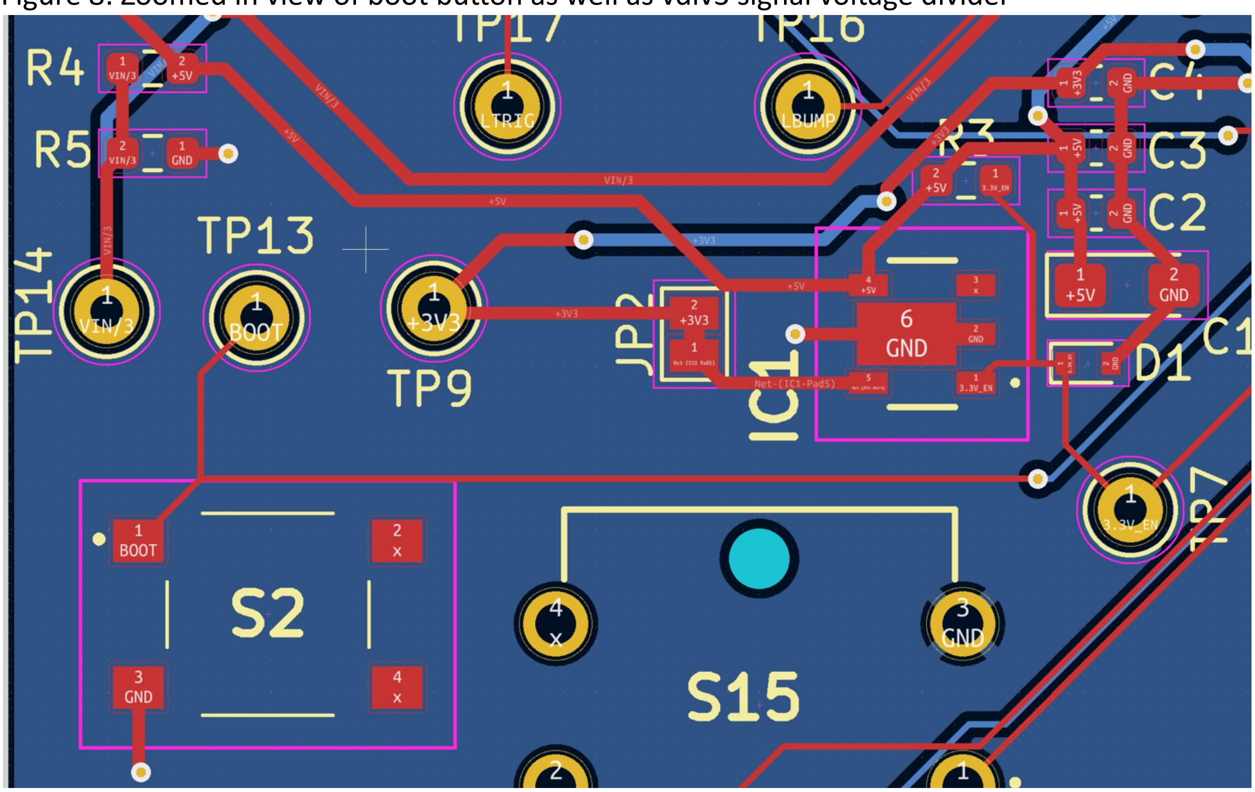


Figure 8: Zoomed in view of boot button as well as vdiv3 signal voltage divider



Bill of Materials

Fingerless Xbox Controller BOM

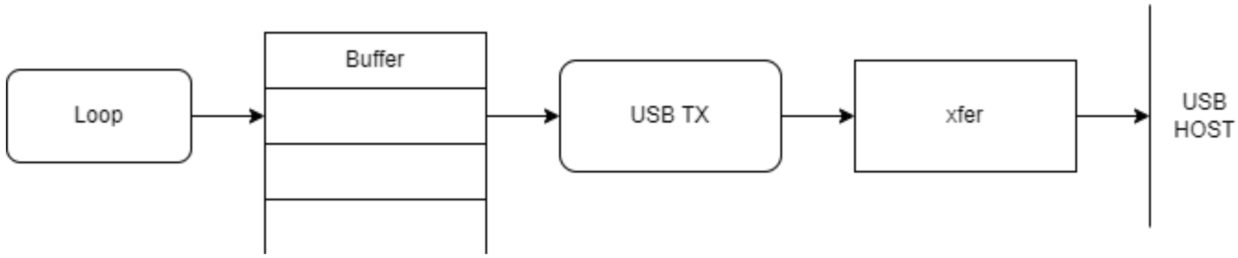
Component	Designator	Description	Footprint	Manufacturer	Quantity	Value
5001	TP2,TP3,TP4,TP6,TP7,TP8,TP9,TP10,TP11,TP12,TP13, TP14,TP15,TP16,TP17,TP18,TP19,TP20,TP21,TP22,TP23 TP24,TP25,TP26,TP27,TP28,TP29,TP30,TP31,TP32	Test Point	TestPoint_Keystone_5000-5004_Miniature	Keystone Electronics	30	
AP7361C-33Y5-13	IC1	LDO Voltage Regulator	AP7361C33Y513	Diodes Incorporated	1	
B3F-5050	S16,S15,S14,S13,S12,S11,S10,S9,S8,S7,S6,S5,S4,S3	Tactile switch	B3F-4055	Omron Electronics	14	
BZX584C3V3-HG3-08	D1	Diode	D_SOD-523	Vishay Semiconductors	1	3.3V
Cap	C1	Electrolytic Capacitor	CP_EIA-3216-10_Kemet-I_Pad1.58x1.35mm_HandSolder	Generic	1	10uF
Cap	C2,C5	Capacitor	C_0603_1608Metric	Generic	2	0.22uF
Cap	C4	Capacitor	C_0603_1608Metric	Generic	1	1uF
Cap	C6,C3	Capacitor	C_0603_1608Metric	Generic	2	0.1uF
COM-09032	U2,U1	Joystick	COM-09032	Sparkfun	2	
COM-15111	J1	Female USB-C Header	USB_C_Receptacle_GCT_USB4105-xx-A_16P_TopMnt_Horizontal	Sparkfun	1	
Conn_ARM_JTAG_SWD_10	J3	SWD Header	PinHeader_2x05_P1.27mm_Vertical	Generic	1	
KIT-16549	J2	M2 Mount	TE_21992304	Sparkfun	1	
MF-NSML200_6-2	F1	2A-6V Resettable Fuse	FUSC3216X60N	Bourns	1	
PTS645SH50SMTR92LFS	S2,S1	Tactile switch	PTS645SL43SMTR92LFS	C&K	2	
Res	R1,R2	Resistor	R_0603_1608Metric_Pad0.98x0.95mm_HandSolder	Generic	2	5.1k Ohms
Res	R4	Resistor	R_0603_1608Metric_Pad0.98x0.95mm_HandSolder	Generic	1	20k Ohms
Res	R5,R3	Resistor	R_0603_1608Metric_Pad0.98x0.95mm_HandSolder	Generic	2	10k Ohms

Assembly and Test Procedure

1. Solder J1, R1 and R2. Connect a USB-C cable to J1 and measure the voltage across TP 6 and TP10 to verify it is 5V.
2. Solder F1, IC1, R3, D1, C1, C2, C3, C4, C5 and C6. With USB-C connected to J1, measure the voltage across TP 9 and TP 10 to verify it is 3.3V.
3. Solder S1 and S2. While S1 is pressed and USB-C connected to J1, measure voltage across TP12 and TP10 to verify it is 0V. While 21 is pressed, measure voltage across TP13 and TP10 to verify it is 0V.
4. Solder J2 and H1. Insert MicroMod module. With USB-C connected to J1 and the other end of the cable connected to a PC, open Arduino (with Teensyduino installed) and upload a blank sketch. Press S2 when sketch is processed to "Boot" the Micro Mod and verify communication with the MicroMod. Computer should respond to the boot signal through Teensyduino's interface as well as loading code onto MicroMod.
5. Solder J3. With USB-C connected to J1, use a Segger J-Link to connect to a PC from J3. Open J-Link Commander and follow steps to validate successful connection to J-Link.
6. Solder R4 and R5. Measure voltage across TP14 and TP10 to verify it is 1.67V.
7. Solder S3. With USB-C connected to J1, while pressing S3, measure voltage across TP15 and TP10 to verify it is 0V. Release S3 and verify that it is no longer 0V.
8. Solder S4. With USB-C connected to J1, while pressing S4, measure voltage across TP16 and TP10 to verify it is 0V. Release S4 and verify that it is no longer 0V.
9. Solder S5. With USB-C connected to J1, while pressing S5, measure voltage across TP17 and TP10 to verify it is 0V. Release S5 and verify that it is no longer 0V.
10. Solder S6. With USB-C connected to J1, while pressing S6, measure voltage across TP18 and TP10 to verify it is 0V. Release S6 and verify that it is no longer 0V.

11. Solder S7. With USB-C connected to J1, while pressing S7, measure voltage across TP19 and TP10 to verify it is 0V. Release S7 and verify that it is no longer 0V.
12. Solder S8. With USB-C connected to J1, while pressing S8, measure voltage across TP20 and TP10 to verify it is 0V. Release S8 and verify that it is no longer 0V.
13. Solder S9. With USB-C connected to J1, while pressing S9, measure voltage across TP21 and TP10 to verify it is 0V. Release S9 and verify that it is no longer 0V.
14. Solder S10. With USB-C connected to J1, while pressing S10, measure voltage across TP24 and TP10 to verify it is 0V. Release S10 and verify that it is no longer 0V.
15. Solder S11. With USB-C connected to J1, while pressing S11, measure voltage across TP25 and TP10 to verify it is 0V. Release S11 and verify that it is no longer 0V.
16. Solder S12. With USB-C connected to J1, while pressing S12, measure voltage across TP26 and TP10 to verify it is 0V. Release S12 and verify that it is no longer 0V.
17. Solder S13. With USB-C connected to J1, while pressing S13, measure voltage across TP27 and TP10 to verify it is 0V. Release S13 and verify that it is no longer 0V.
18. Solder S14. With USB-C connected to J1, while pressing S14, measure voltage across TP28 and TP10 to verify it is 0V. Release S14 and verify that it is no longer 0V.
19. Solder S15. With USB-C connected to J1, while pressing S15, measure voltage across TP29 and TP10 to verify it is 0V. Release S15 and verify that it is no longer 0V.
20. Solder S16. With USB-C connected to J1, while pressing S16, measure voltage across TP30 and TP10 to verify it is 0V. Release S16 and verify that it is no longer 0V.
21. Solder U1. With USB-C connected to J1, while pressing the joystick in, measure voltage across TP23 and TP10 to verify it is 0V. Release the joystick and verify that it is no longer 0V.
22. Measure voltage between TP8 and TP10. While gathering this measurement, move the joystick all the way to the left, then all the way to the right to verify that one direction results in 3.3V and the other results in 0V. Repeat this step with the vertical axis and the voltage between TP22 and TP10. The direction which the voltage is at is max or min is irrelevant as that can be fixed in code.
23. Solder U2. With USB-C connected to J1, while pressing the joystick in, measure voltage across TP32 and TP10 to verify it is 0V. Release the joystick and verify that it is no longer 0V.
24. Measure voltage between TP4 and TP10. While gathering this measurement, move the joystick all the way to the left, then all the way to the right to verify that one direction results in 3.3V and the other results in 0V. Repeat this step with the vertical axis and the voltage between TP32 and TP10. The direction which the voltage is at is max or min is irrelevant as that can be fixed in code.

Software Dataflow Diagram



Verification of Functionality

The following video contains verification that the Fingerless Xbox Controller inputs communicate with the computer and therefore work as intended:
https://youtu.be/2p_fDrv6Go8

Presentation Poster

The following page contains a copy of my final poster used to present to colleagues and industry experts.

Presentation Video

In this video I give a presentation very similar to what I gave during demo day, noise included. I captioned the video, so feel free to turn on closed captioning if necessary.
YouTube Video of Demo Day: https://youtu.be/Q-L18eyKbQ0

Fingerless Xbox Controller

CONTACT INFORMATION
Brooke Snow
Brooke.Lenae.Snow@gmail.com

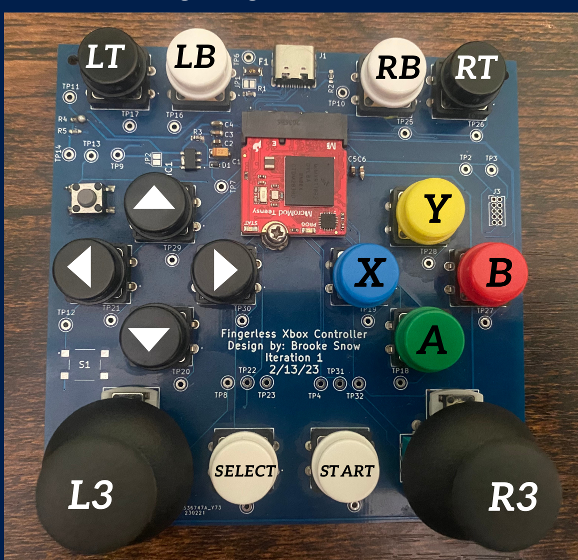
Objectives

In designing the Fingerless Xbox Controller, I wanted to create a device that:

1) is easier to use for someone who had hands that are not the same shape as most.
2) is satisfying to use.
3) lets the user know if they pressed a button regardless of what the game is showing them.
4) allows the user to play games on their Windows OS device.

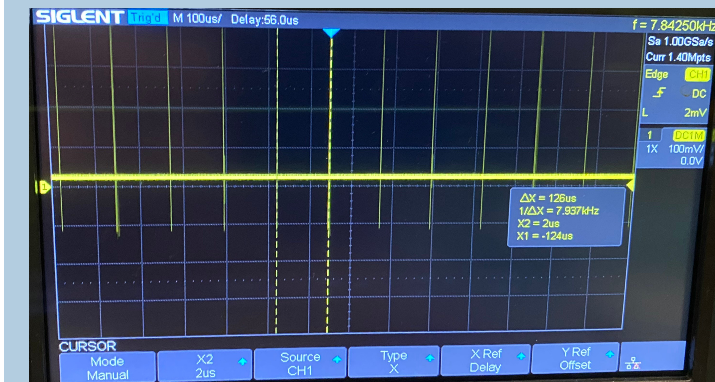
Button Diagram

This picture shows the layout of the buttons, which is intentionally laid out similarly to a traditional Xbox controller. I'm choosing to trust the research that game designers do when creating the games.



Summary of Results

The Fingerless Xbox Controller transmits data to the computer every 126 microseconds, or almost 8000 times a second, as seen being measured below. This makes the response time to user input unnoticeable to the user.



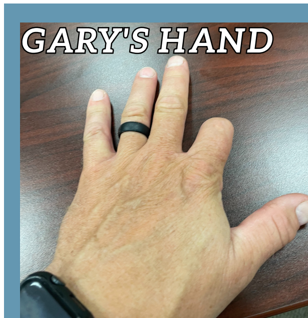
The final board (seen in button diagram) is a 10 cm x 10 cm (3.9 in x 3.9in) square. This board is too small for the requirements. While someone with a mild amputation (see Gary's hand in the motivation section) may have an easier time using this controller over a traditional controller, someone with Symbrachydactyly (See Nicki's hand in the motivation section) may have more difficulty navigating around the joysticks and close buttons.

Conclusion

The Fingerless Xbox Controller does most of what it aims to. It provides a way for those who do not have normal use of their fingers to play games and reach all of the inputs. However, after some testing, I've determined that, in order for all the buttons to be laid out with enough space to press and maneuver with ease, the joysticks should be off board via jumpers.

Abstract

The Fingerless Xbox Controller (FXC) aims to make gaming more accessible to those who lack the full use of all their fingers by providing a better layout for the inputs on an ordinary Xbox controller. The FXC communicates with a computer that operates on Windows OS via a USB. That signal goes directly to the MicroMod, and the 5V supply from the USB gets run through a linear regulator to become a regulated 3.3 volts. That supplies power to the MicroMod and all of the buttons and joysticks. Pushing a button or joystick pulls the GPIO (or ADC in the case of the joysticks) to ground, giving an active low signal. The joysticks x and y axes are based off potentiometers in the joysticks that act as variable voltage dividers and send that voltage to the ADC of the MicroMod. The MicroMod takes these signals from the user inputs and translates them into Xinput standard USB communication, which then gets sent to the computer to process. All these inputs are laid out on one flat surface so that you do not need to hold the FXC while playing with it.



Motivation

The Fingerless Xbox Controller was made for users with a wide range of hands. From personal experience, family game night can be difficult when one player cannot reach all the buttons. The Fingerless Xbox Controller was designed to make gaming more fun and accessible for everyone.