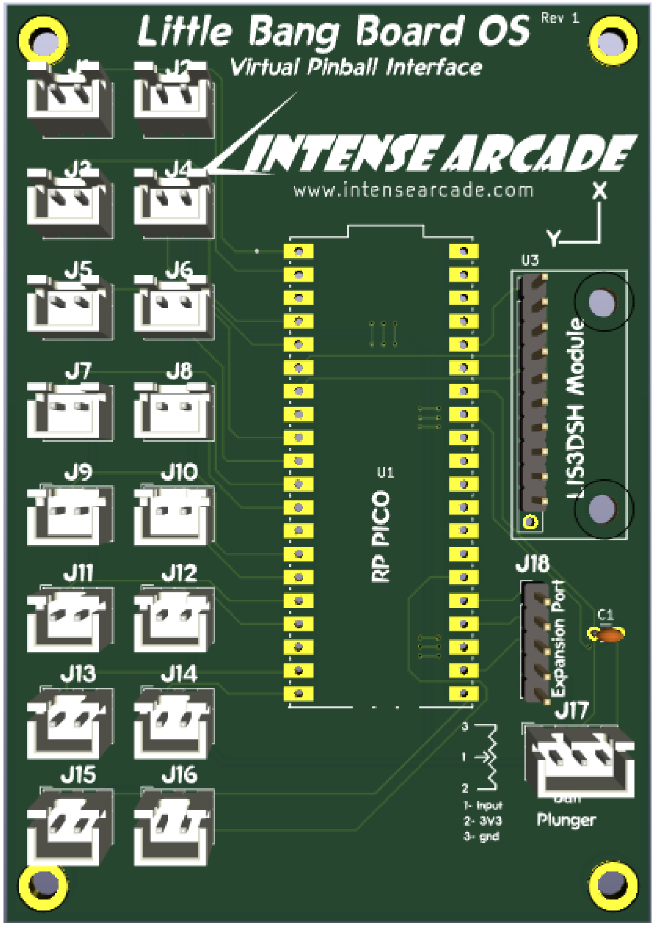
Little Bang OS Virtual Pinball Controller

Version 2.00
Author: Rickey M. Horwitz

Introduction

This document details the assembly of a game controller specifically used to interface with a virtual pinball machine. The document provides detailed instructions on theory of operation, assembly, test, and implementation.



Description

This project revolves around a Raspberry Pi Pico module. The Raspberry Pi Pico is a system on a chip contain a dual core ARM Cortex 11 processor that provides a peripheral interface between the human and host PC using USB, like a keyboard, mouse, or joystick. These are commonly referred to as USB HID devices. Devices such as mechanical buttons, accelerometer, and a ball plunger are connected to the Pico via Printed Circuit Board (PCB). The signals from these devices are processed in the Pico and sent to the host PC using a Joystick and Keyboard HID interface to ultimately control a virtual pinball application.

Specifications

Interface:	USB 1.1
USB HID Types:	Joystick/Gamepad and Keyboard on a single USB
Button inputs:	16 each, debounced at .03 seconds (30 milliseconds).
Button latency:	.003 seconds (3 milliseconds)
Ball Plunger:	16-bit, sampled at 333 Hz (.003 seconds)
Accelerometer:	LIS3DSH, 10-bit, X/Y axis sampled at 100 Hz (.01 seconds)

Project Summary

The Little Bang Board OS is a DIY kit that comes complete with all components except button cables. The following will be in the kit:

1. Little Bang PCB	qty 1
2. LIS3DSH Module	qty 1
3. RP Pico Module	qty 1
4. JST XH 2 pin header	qty 16
5. JST XH3 Pin header	qty 1
6. 11 pin header	qty 1
7. 20 pin header	qty 2
8. 5 pin header	qty 1
9. .1ufd Capacitor	qty 1
10. USB A to C, 1m	qty 1

The project requires soldering components to a PCB. A soldering iron and rosin core solder are required. Assembling this project should not take longer than a half hour to complete. The Raspberry Pi Pico is the most powerful and versatile hobby processor on the market today. The extremely low cost makes it an ideal choice for almost any project. The LIS3DSH accelerometer module was chosen as it is the most popular model available, and the performance is very good. The PCB design uses JST XH 2.5mm pitch connector for all inputs. This allows either JST XH or Dupont connector terminated cables making the design neat and clean.

Instructions

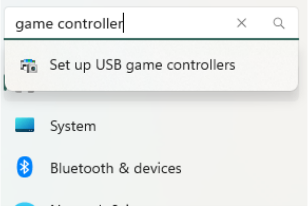
- Locate both 20-pin, single-row headers for the Pico and install these headers into the PCB (but not soldered).
- Place the Pico module on top of the pin headers. Orient the Pico so that the USB port faces towards the top of the PCB.
- Once the pin headers are fully aligned with the holes on both module and PCB, the pins are ready to be soldered. Try soldering the four corners first so that adjustments can be made without damaging the assembly.

- For the LIS3DSH module, install the pin header first, ensure that the pins are aligned squarely with the PCB before soldering them.
- When soldering the LIS3DSH module to the pin header **it is important that both module and PCB are parallel.** Failure to do so will result in nudging errors and the virtual playfield not being level with the cabinet.
- Solder all the JST 2-pin header connectors (J1-J16). It is best to solder a single pin on each of these connectors so that that can all be aligned correctly before soldering.
- Solder header connector J17 and J18.
- C1 is a .1ufd ceramic capacitor used for filtering the Plunger input. Add this component if not using an Intense Arcade Plunger/Ball Shooter assy.
- Inspect the bottom of your PCBA for any solder bridging. Any bridging may roach your board.
- A USB C to standard A cable is required. This design uses the older USB 1.1 specification and as such the cable length can be 2 meters long without any issues.

Note

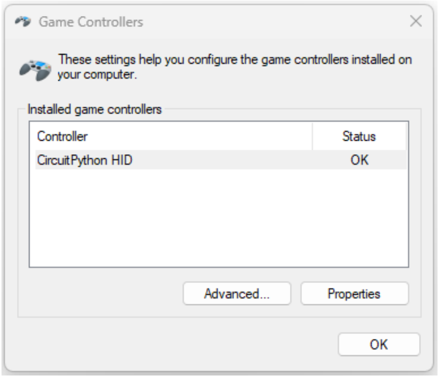
The RP Pico comes pre-programmed with a default configuration script installed.

- On your PC, bring up Device Manager and look for an extra HID Compliant Consumer Control Device, HID Compliant Game Controller, and a USB Input Device.



- In Windows settings, select '*Set up USB game controllers*':

- A list should appear in a window that looks like:

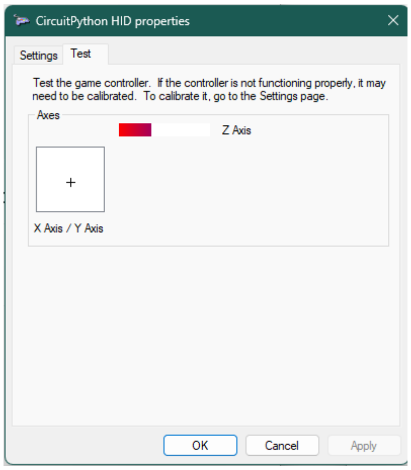


These settings help you configure the game controllers installed on your computer.

Controller	Status
CircuitPython HID	OK

- Select the CircuitPython HID and click on *'Properties'*

- The following window should appear:



Test the game controller. If the controller is not functioning properly, it may need to be calibrated. To calibrate it, go to the Settings page.

- Shake the Little Bang Board and ensure that the X/Y axis moves accordingly.
- Pull the ball Plunger back and forth and ensure that the Z axis reacts accordingly.
- Click on OK and close the CircuitPython HID properties.
- Open the notepad app on the PC, in the session press the buttons on the Little Bang Controller that correspond to the keystrokes on the keyboard. Ensure that they all work accordingly.

Installing the little bang board assembly

Mounting the assembly should follow these requirements:

- The assembly should be mounted within reach of the button wires, plunger wires, and USB cable
- The USB plug requires at least 40 mm length allowance for strain relief.
- The mounting area needs to be flat and level.
- Route plunger wires away from other wires. This will reduce electrical noise and reduce plunger jitter.
- After mounting, check the level by running the windows game controller utility.
- If the X/Y accelerometer is picking up too much vibration from the cabinet during no activity, the dead band may need to be increased. This can be accomplished by modifying the configuration script.

Configuration Editing

As mentioned, the Bang Board uses a texted based configuration file that is located on the RP Pico. When the board's USB is plugged into a PC, it will appear as a flash drive. You can browse the files on this drive and find the 'config.txt' file. At the time of this publication, this file needs to be edited manually and only the first 24 lines of this file are relevant. So, let's explore them.

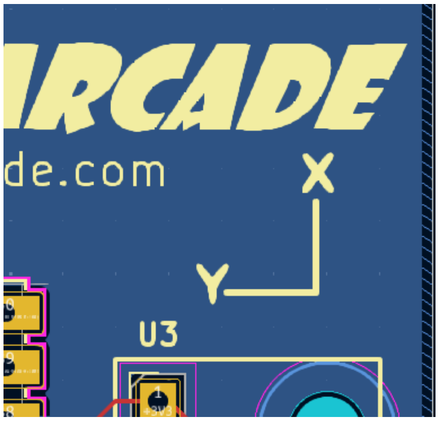
Note
Do not delete the configuration file!!

Accelerometer Dead Band

Noise and vibration will cause the accelerometer to fluctuate, showing up as a jitter. Increasing the dead band value reduces this affect. Placing the value to too high will make the accelerometer unresponsive to slight nudging. Set this value to 500-1000.

X_ Accelerometer Gain and Y_ Accelerometer Gain

If nudging is not producing the correct response, try adjusting this value. By default, both X and Y are set to 3500.



Accelerometer Orientation

When mounting the Bang Board in your Pinball cabinet, be mindful of the orientation of the mounting of the PCBA. The PCBA has an in indicator showing the X and Y orientation".   Orientation assumes a right-angle placement. So, there is 0, 90, 180, and 270 degrees which are enumerated as 0, 1, 2, and 3 respectively.

Plunger Offset

This value is not yet implemented in this software and doubtfully will it ever be.   The offset is the value that

Plunger Gain

A virtual pinball plunger uses a linear potentiometer that has an output voltage directly proportional to the position of the wiper. Some virtual pinball plunger implementations lack full range from their slider potentiometer. In the case of the Bang Board, the voltage range of this pot is 0 to 3.3V which is the full extent with an Intense Arcade Plunger. The Plunger gain is a value that modifies the output to compensate for limitations. If you use an Intense Arcade shooter leave this value at default.

Button Values

As stated, the Little Bang Board supports up to 16 button inputs. These buttons correspond to any keystroke on a keyboard. The names of each keystroke use the naming conventions in Python. Below is an excerpt of an actual configuration script.

```
acc_deadband = 1000
x_acc_gain = 3500
y_acc_gain = -3500
acc_orientation = 1
plunger_offset = 1
plunger_gain = 1
button_01 = LEFT_SHIFT
button_02 = RIGHT_SHIFT
button_03 = LEFT_CONTROL
button_04 = RIGHT_CONTROL
button_05 = ONE
button_06 = FIVE
button_07 = FOUR
button_08 = ENTER
button_09 = Q
button_10 = EQUALS
```

button_11 = MINUS
button_12 = F11
button_13 = T
button_14 = SPACE
button_15 = FOUR
button_16 = ENTER

Saving The Configuration

Sometimes the Pico may prevent you from saving the configuration on the device. In cases where this occurs simply press the RST key on the Pico module and wait for the flash drive to reappear and proceed to save the file.

RP Pico Variants

The RP Pico comes in two variants for this project, each having a USB C interface and each functioning identically. The first variant is non castellated through hole mounted and includes an onboard Neo Pixel (RGB) LED. The control for this LED is GP23 (pin 37). To add an external Neo Pixel (WS2812B), remove the RGB solder jumper and tap pins 37 (GP23), 40 (5V) and 38 (GND).

The alternate variant sticks to the RP Pico module standard. It is suggested to not attempt to solder it directly to the PCB, as the USB cable my have issues with clearance. This variant does not have the Neo Pixel GP23.

Expansion Port

The expansion port is designed for adding PWM controlled output driver board using a SPI port and 3.3V.

1- GP20
2 - GP18
3 - GP19
4 - 3.3V
5 - GND

Contact Me

This is an open-source project I created and can be accessed on GitHub at
https://github.com/rmhorwitz/Little_Bang_Board/
If you have any questions, contact me at rick.horwitz@yahoo.com
I also have a website www.IntenseArcade.com where I sell small VP cabinets and affordable parts.