Adafruit Joy Bonnet for Raspberry Pi

Created by Phillip Burgess

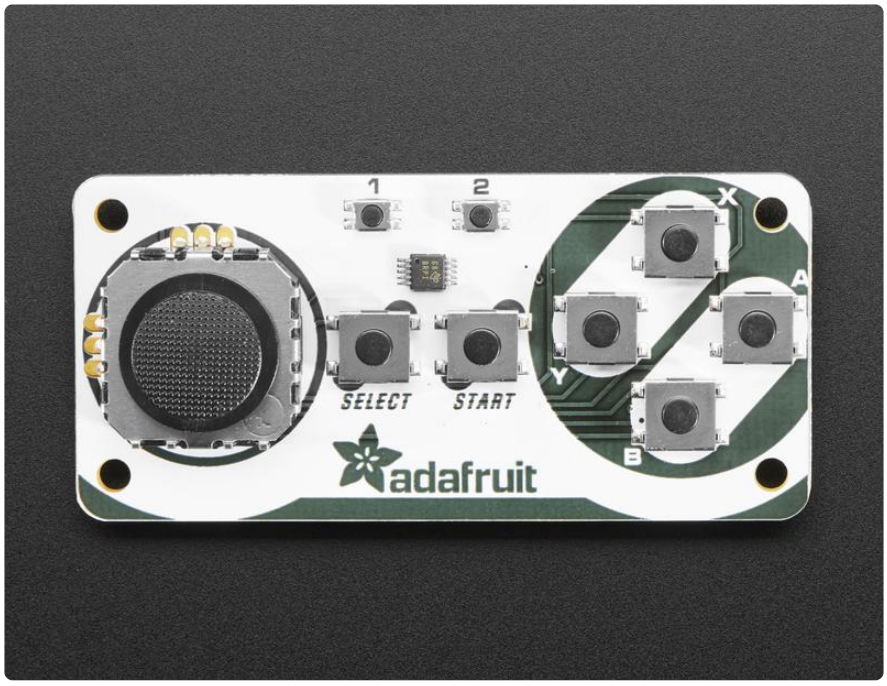
https://learn.adafruit.com/adafruit-joy-bonnet-for-raspberry-pi

Last updated on 2023-09-22 06:59:48 PM EDT

Table of Contents

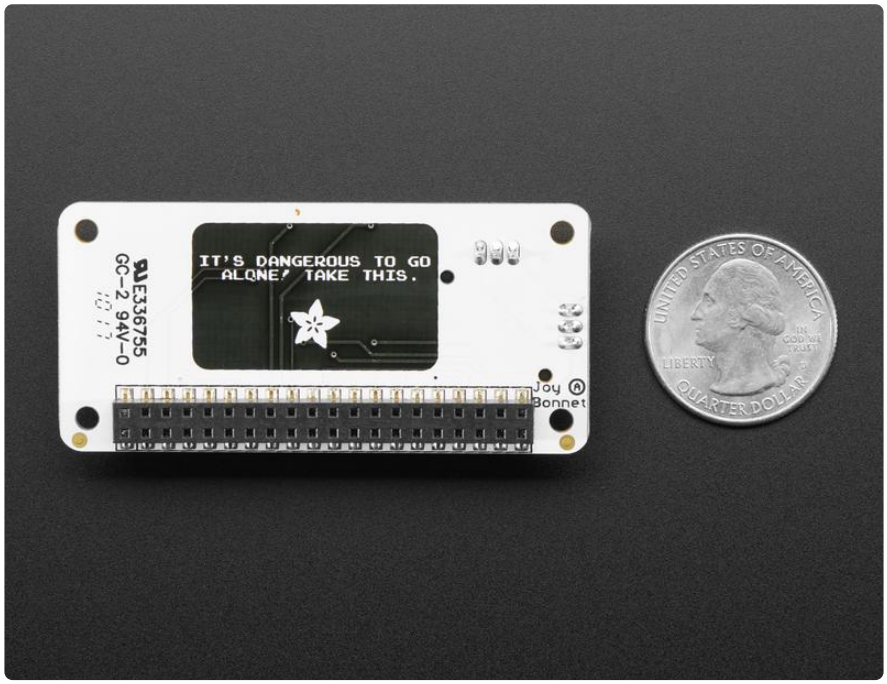
Overview

Pocket-sized fun is the name of this game, with the Joy Bonnet - our most fun Bonnet ever (no we didn't even think that was possible, either!) This Bonnet fits perfectly on top of your Raspberry Pi Zero (any kind) and gives you adorable hand-held arcade controls. Once you install our script onto your Pi, the controls will act like a keyboard, for easy use with any emulator or media player.



Personally, we found this Bonnet to work best with RetroPie/EmulationStation. On a Pi Zero we could emulate NES and MAME game (our favorites!) but other emulators that

don't require more than 1GHz speeds will work OK too, e.g. a N64 emulator won't work, it needs way more power!



Best of all, this Bonnet is fully assembled - no soldering at all is required. You may need to solder headers onto your Pi Zero (or use press-fit headers) but once that's done you're ready to rock. In theory you could also use this with one of the full-size Pi models, but it wouldn't be very hand-held

Since you'll likely grip the Pi during play, we strongly recommend a comfortable enclosure to go along with it! The Adafruit Pi Zero case works well and is nice and rounded for easy holding

Adafruit Raspberry Pi Zero Case
This is a basic, classic Pi Zero enclosure with a black base and a clear top. The case is as minimal as it gets, coming in just two pieces of polycarbonate that...
https://www.adafruit.com/product/3252

Friendly forum-member cvadillo suggests () slipping the extra piece of plastic underneath the bonnet to support it

A Mini HDMI cable will also make it easy to plug-and-play into your HDMI screen

Mini HDMI to HDMI Cable - 5 feet
Connect a device with a Mini HDMI port to a regular sized HDMI port together with this basic HDMI cable. It has nice molded grips for easy installation, and is 1.5 meter long...
https://www.adafruit.com/product/2775

Install & Use

Software

Software installation for the Joy Bonnet requires an internet connection. That's a frequent topic already covered in other Pi getting-started guides, so we'll assume

here that your Pi is already booted and networked, running Raspbian or a gaming-ready OS like RetroPie. (RetroPie is what we use, and recommend!) RetroPie has a nice First Installation guide ()

You can set up the controls however you like, its easy to change them up later. Just get logged into RetroPie, and setup WiFi. Then get into a terminal either on the HDMI console or by ssh'ing in.

You may find this easiest if ssh is enabled on the Pi, and then log in with a terminal app. ()This lets you copy-and-paste the commands that follow, as they're very exact about spelling.

Install!

Enter the following few lines to install support for the buttons and joystick:

```
cd ~
sudo apt-get install python3-pip
sudo pip3 install --upgrade adafruit-python-shell
wget https://raw.githubusercontent.com/adafruit/Raspberry-Pi-Installer-Scripts/
master/joy-bonnet.py
sudo python3 joy-bonnet.py
```

When run, this script will offer a couple of options:

- Disable overscan? If you answer "Y" this removes the black border around the screen (common on HDMI monitors). Takes effect on next boot.
- Install gpio-halt utility? Linux-based systems like the Raspberry Pi don't like it when you just switch off power...there's a specific shutdown process, else the SD card may get corrupted. The gpio-halt tool lets you add one more button between ground and any unused GPIO pin to initiate an orderly shutdown. Gaming OSes like RetroPie already include a shutdown command among their menu options, so you probably do not need this, unless you want a dedicated button for it.

When the script finishes you'll be asked whether to reboot. Answer "N" if you plan on installing other software. Either way, sudo reboot when done to get the joy bonnet software activated!

```
    arm-linux-gnueabihf-gcc -pthread -shared -Wl,-O1 -Wl,-Bsymbolic-functions -W
l,-z,relro -fno-strict-aliasing -DNDEBUG -g -fwrapv -O2 -Wall -Wstrict-prototype
s -D_FORTIFY_SOURCE=2 -g -fstack-protector-strong -Wformat -Werror=format-securi
ty -Wl,-z,relro -D_FORTIFY_SOURCE=2 -g -fstack-protector-strong -Wformat -Werror
=format-security build/temp.linux-armv6l-2.7/evdev/ecodes.o -o build/lib.linux-a
rmv6l-2.7/evdev/_ecodes.so

Successfully installed evdev
Cleaning up...
Installing Adafruit code in /boot...
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  6982  100  6982    0     0  37135      0 --:--:-- --:--:-- --:--:-- 37537
Configuring system...
Done.

Settings take effect on next boot.

REBOOT NOW? [y/N] y
Reboot started...
```

That's it! The script to generate keycodes is now installed and will be run every time the Pi boots

Once RetroPie has restarted, go to the MENU and step down to CONFIGURE INPUT and select it to re-start the input configuration. The left joystick can be used for up/down/left/right. The right buttons are A B X Y. Middle buttons are Select and Start as labeled. Then you get two more buttons you can set up as left/right shoulder or trigger buttons.

Note that pressing Select+Start will quit out of games, other than that the controls are used in the emulators by generating 'key presses' - see below for how to change the keypresses!

Although the Joy Bonnet has an analog stick, the software installed above makes this act like a digital 4-way stick emitting key presses. This is on purpose and by design, as Retropie on a single-stick controller like this expects digital inputs.

Advanced Usage

If you need to change the key assignments: edit the file /boot/joyBonnet.py say with `sudo nano /boot/joyBonnet.py`

Near the top of the code you'll see this table:

```
KEYS= { # EDIT KEYCODES IN THIS TABLE TO YOUR PREFERENCES:
        # See /usr/include/linux/input.h for keycode names
        # Keyboard        Bonnet          EmulationStation
        BUTTON_A: e.KEY_LEFTCTRL, # 'A' button
        BUTTON_B: e.KEY_LEFTALT,  # 'B' button
        BUTTON_X: e.KEY_Z,        # 'X' button
        BUTTON_Y: e.KEY_X,        # 'Y' button
        SELECT:   e.KEY_SPACE,    # 'Select' button
        START:    e.KEY_ENTER,    # 'Start' button
        PLAYER1:  e.KEY_1,        # '#1' button
        PLAYER2:  e.KEY_2,        # '#2' button
        1000:     e.KEY_UP,       # Analog up
        1001:     e.KEY_DOWN,     # Analog down
        1002:     e.KEY_LEFT,     # Analog left
        1003:     e.KEY_RIGHT,    # Analog right
}
```

'#' lines are human comments and do nothing for the code. The first eight actual elements in the table correspond to the button inputs, while the last four are the joystick directions.

Downloads

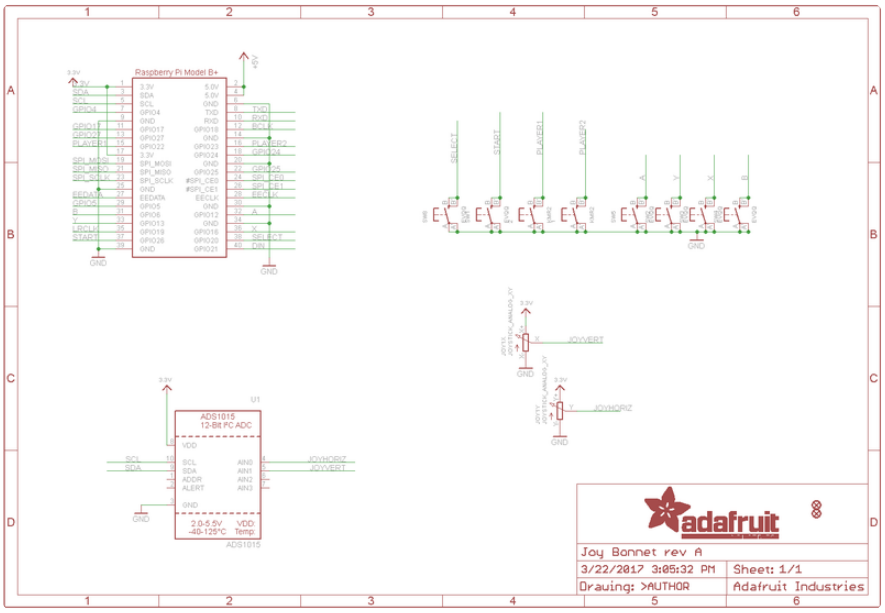
Source Code

- Joy Bonnet installer () and key-press python script () available on github!

Files

- EagleCAD PCB files on GitHub ()

Schematic


Raspberry Pi Model B+
ADS1015
12-Bit I²C ADC
Joy Bonnet rev A
3/22/2017 3:05:32 PM
Sheet: 1/1
Drawing: >AUTHOR
Adafruit Industries