

Build Your Own Portable Gaming System

Paul Pagel

Build Your Own Portable Gaming System

If you have not already done so:

Download the pre-requisites instructions and complete all downloads and installations

<http://prereqs.codemash.org/>

Download at least one Doom1 WAD to your laptop

<https://www.wad-archive.com/wad/Doom-Shareware-v1.9>

Slide deck download (PDF):

<https://github.com/DigiTorus86/Teensy-R4ge-Pro/tree/master/documents>

Session Objectives

Load and run different emulators and games on the hardware.

Learn how to use the Teensy Audio Library and hardware peripherals in custom games and other programs.

Troubleshoot and resolve any lingering hardware issues.

Pre-Requisites

- Working Teensy R4ge Pro with 8MB PSRAM installed
- VS Code with Platformio and Teensy platform
- Arduino IDE with Teensyduino
- Micro SD card formatted for FAT32
- Clone of <https://github.com/Jean-MarcHarvengt/MCUME/>
- Clone of <https://github.com/DigiTorus86/Teensy-R4ge-Pro/>

IDE Setup and Verification

Are you ready?

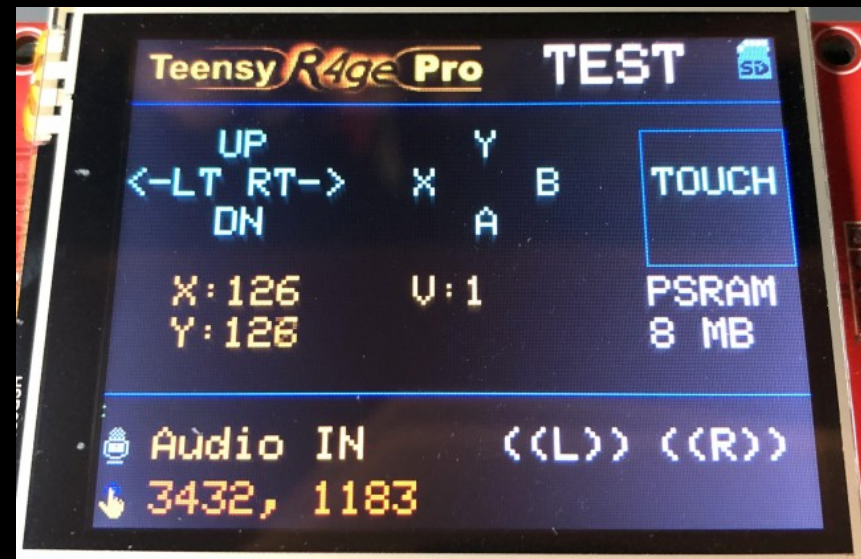
Arduino IDE

1. Ensure Teensy is connected to your laptop via USB
2. Launch the Arduino IDE
3. Open the [teensy-r4ge-pro-test](#) sketch
4. Under Tools → Board, select “Teensy4.1”
5. CPU Speed: “600 MHZ”
6. Select the Port that the Teensy is on (this will vary by system and OS)
7. Programmer: “AVR ISP” or “AVR ISP mkII”
8. Go to Sketch → Upload (or click the right arrow button)
9. Should compile and the Teensyduino app should launch



Verification

- Ensure that all peripherals are still working as expected. Use the instructions in the header comments if needed.
- Change line 155 to some other (family-friendly) short word.
- Re-upload and verify that the title text changed accordingly.
- The test app is a good resource for simple example code to use the hardware and peripherals.



Doom

What to do when all hell breaks loose



A Brief History

- Originally released in December 1993
- Source code released in 1997
- Many sequels and spin-offs



Prepare the SD Card



1. Create a local folder named "MCUME-SD"
2. Extract the [sd.zip](#) file from the MCUME repo to it.
3. Copy the [DOOM1.WAD](#) file to the [data](#) folder.
4. Copy the contents of MCUME-SD to the micro SD card root.
(The [data](#) folder should be at the SD root.)
5. Insert the SD card into the Teensy.

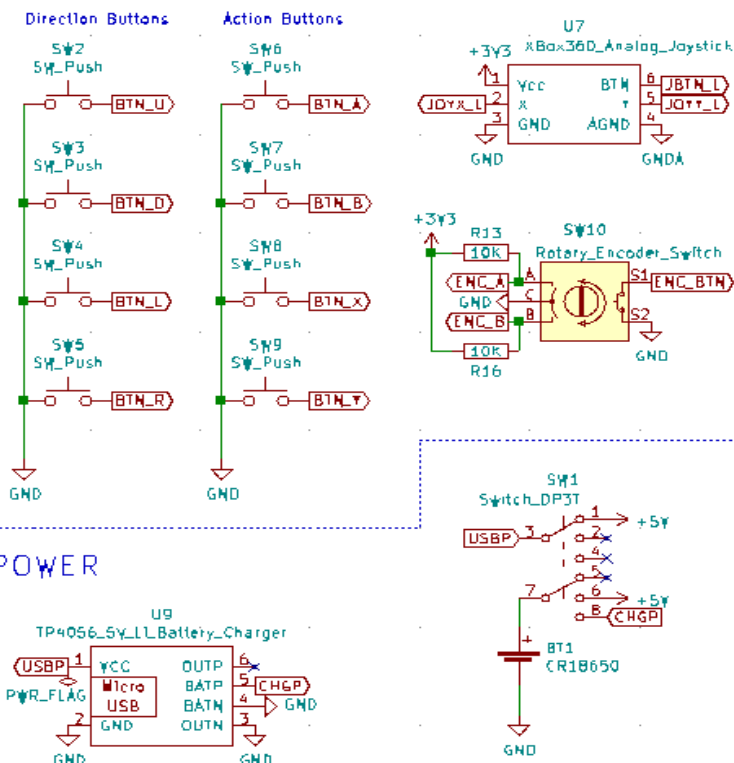
Source and Settings

1. In the ArduinoIDE, open the
Teensy-R4ge-Pro/MCUME/teensydoom/ folder.
2. Under Tools → Optimize, select “Faster”
3. Upload the sketch to the Teensy.

Doom Modifications — iopins.h

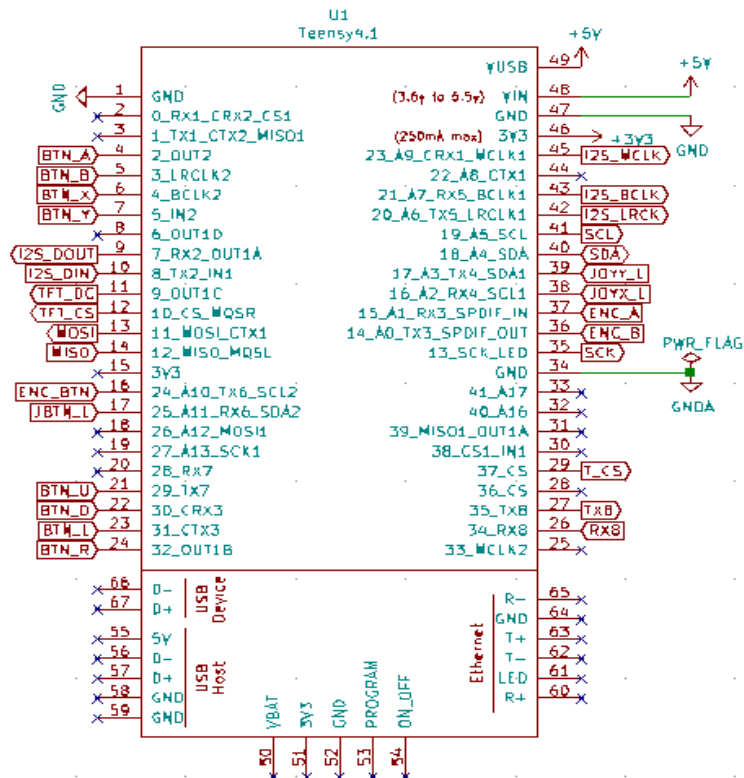
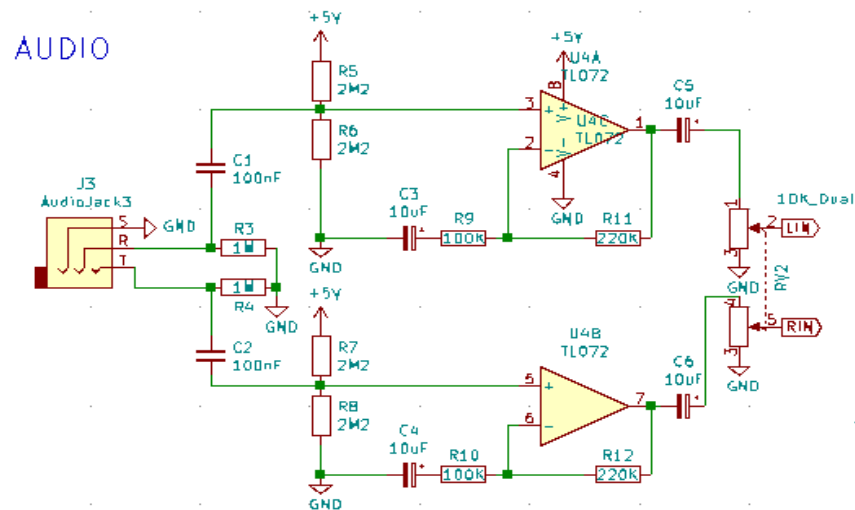
```
#define TFT_CS      10                // Second joystick
#define TFT_RST     255               #define PIN_JOY1_BTN    24 // Encoder
                                     button
#define PIN_JOY2_A1X 16               #define PIN_JOY1_1     29 // UP
#define PIN_JOY2_A2Y 17               #define PIN_JOY1_2     30 // DOWN
#define PIN_JOY2_BTN 25               #define PIN_JOY1_3     32 // RIGHT
#define PIN_KEY_USER1 2 // BTN A      #define PIN_JOY1_4     31 // LEFT
#define PIN_KEY_USER2 3 // BTA B
#define PIN_KEY_USER3 4 // BTN X
#define PIN_KEY_USER4 5 // BTN Y
```

CONTROLLER INPUT



POWER

AUDIO



Doom Modifications - platform

platform_config.h changes:

```
#define INVX      1
```

```
#define INVY      1
```

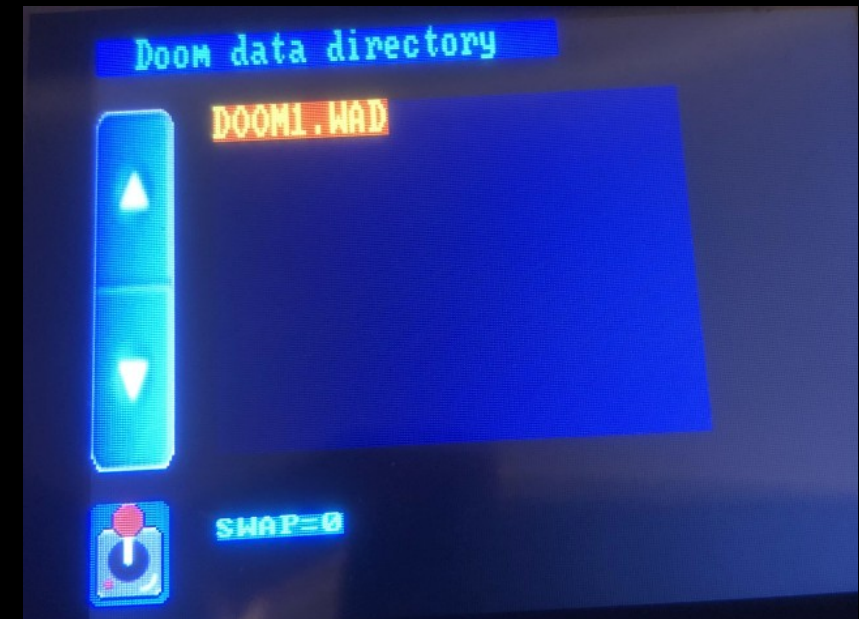
```
#define ILI9341    1
```

Running the Emulator

- Use the Up/Down buttons or joystick to change selection.
- Press rotary encoder button to select file.
- Press Y button to reset emulator.



If screen is blank, re-insert SD card and reboot hardware.



Playing Doom™

- Change menu selection: Up/Down buttons or joystick
- Select menu, get item: A button
- Fire: Rotary encoder button
- Door/switch: B button
- Show Map: X button



Unfortunately, no sound available.



Other Emulators

Running old systems on new hardware

General Instructions

- Obtain emulator game files.
- Copy game files to corresponding emulator folder on SD card.
- Copy the iopins.h and platform_config.h files from the [Digitorus86/Teensy-R4ge-Pro/MCUME](#) folder to the [MCUME/MCUME_teensy41/](#) folder.
- Load the emulator into ArduinoIDE.
- Select the appropriate Tools->Optimize setting.
- Upload to the Teensy.

Castaway — Atari ST

1. Save .st game files to “st” folder on SD card.
2. Copy the iopins.h and platform_config.h files from the [Teensy-R4ge-Pro/MCUME/teensycastaway41/](#) to the [MCUME/MCUME_teensy41/teensycastaway41/](#) folder.
3. Load into ArduinoIDE and select “Smallest Code” option.

Castaway — Verified Games

- 1943 (with sound)
- 23 Games
 - Battlezone
 - Joust
- Arkanoid
- Caves of Rigel
- Crossbow
- Defender of the Crown
- Dragon Spirit
- Flying Shark
- Fusion
- Ivanhoe
- Karate King
- Slightly Magic
- Super Sprint
- Twin World



Ultimate Amiga Emulator

1. Save .ADF or .HDF files to “amiga” folder on SD card.
2. Copy the iopins.h and platform_config.h files from the [Teensy-R4ge-Pro/MCUME/teensyuae41/](#) to the [MCUME/MCUME_teensy41/teensyuae41/](#) folder.
3. Load into ArduinoIDE and select “Smallest Code” option.

Nofrendo — NES

1. Save files to “nes” folder on SD card.
2. Copy the iopins.h and platform_config.h files from the `Teensy-R4ge-Pro/MCUME/teensynofrendo/` to the `MCUME/MCUME_teensy/teensynofrendo/` folder.
3. Load into ArduinoIDE and select “Faster” option.

Nofrendo — Verified Games

- Super Mario Brothers
- Tetris
- Donkey Kong
- Donkey Kong Jr.
- Elevator Action
- Galaga
- Faxanadu
- Ms. Pac Man
- Q*Bert
- Xevious



USB Keyboard

Teensy can act as a USB host for peripherals, such as a keyboard. Some games (especially for computer emulators) will require a keyboard.

To use:

- Attach host cable to Teensy with black wire to left, red to right.
- Attach USB device to host cable.

UAE — Verified Games

- Battle Squadron
- Elite
- SturmTruppen
- Turrigan
- Turrigan 3



Custom Sketches

Creative DIY Hackery

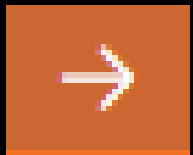
VS Code + PlatformIO

If you plan to do any custom development, VS Code + PlatformIO is **STRONGLY** recommended!

- Modern IDE
- Much better developer experience
- Excellent industry support

VS Code + PlatformIO

1. Launch VS Code.
2. File → Open folder:
Teensy-R4ge-Pro/PlatformIO/teensy-r4ge-pro-test
3. Make sure the Teensy is connected to your laptop.
4. Click the upload button in the bottom toolbar.



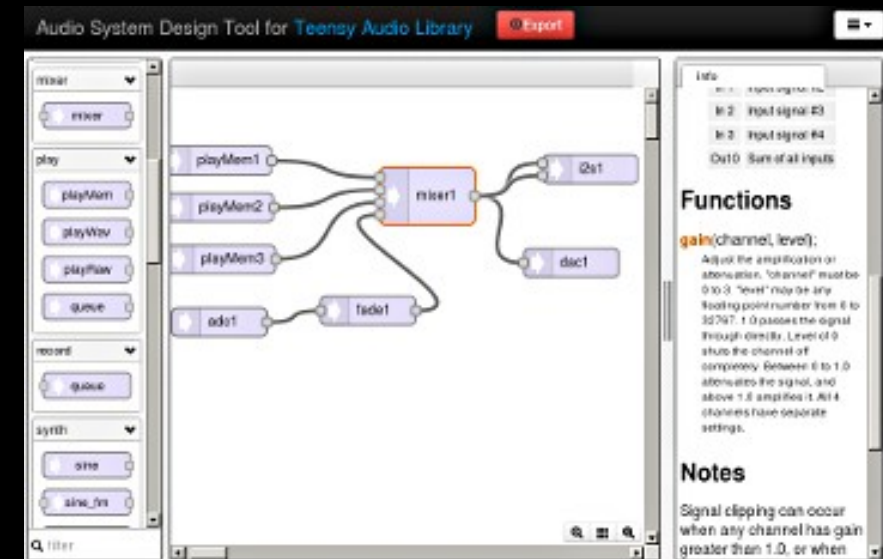
Teensy Audio

Teensy 4.1 is capable of sophisticated, high-quality audio generation, capture, and processing.

Online Audio Design GUI

<https://www.pjrc.com/teensy/gui/>

Imports and exports C code



Teensy Audio Network

Basic Structure:

Source(s) → Effects/Filters/Mixers → Output

Synthesizers

Chorus/Reverb

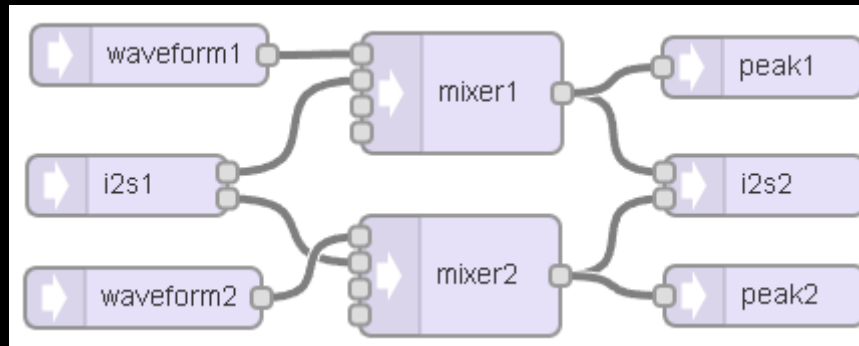
WAV files → Low Pass → I2S DAC (Inter IC Sound)

I2S ADC Channel Mix

Teensy Audio – Import

- 1) Launch Teensy Audio GUI in browser:
<https://www.pjrc.com/teensy/gui/?info=AudioMixer4>
- 2) Copy the code between the `//GUItool` comments in test sketch
- 3) Click IMPORT on the Audio GUI and paste into the textbox

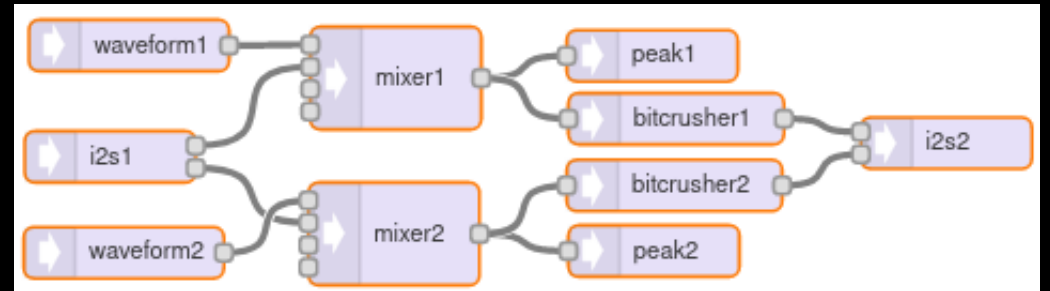
Sources



Output

Teensy Audio – Export

- 1) Modify the signal chain as shown:
- 2) Export to clipboard and paste back into sketch.



- 3) Add the following at line 185 in the sketch:

```
bitcrusher1.bits(4);
```

```
bitcrusher2.bits(8);
```

- 4) Upload the sketch and hear the difference by pressing the left and right buttons.
- 5) Experiment with other bit values and/or filters.


```
// GUItool: begin automatically generated code
```

```
AudioInputI2S          i2s1;          //xy=668.75,348
AudioSynthWaveform      waveform2;     //xy=676.75,404
AudioSynthWaveform      waveform1;     //xy=679.75,283
AudioMixer4             mixer2;        //xy=834.75,393
AudioMixer4             mixer1;        //xy=835.75,302
AudioAnalyzePeak        peak1;         //xy=985.75,289
AudioAnalyzePeak        peak2;         //xy=986.75,403
AudioEffectBitcrusher    bitcrusher2;  //xy=998.75,364.75
AudioEffectBitcrusher    bitcrusher1;  //xy=999.75,325.75
AudioOutputI2S          i2s2;          //xy=1157.75,340
AudioConnection          patchCord1(i2s1, 0, mixer1, 1);
AudioConnection          patchCord2(i2s1, 1, mixer2, 1);
AudioConnection          patchCord3(waveform2, 0, mixer2, 0);
AudioConnection          patchCord4(waveform1, 0, mixer1, 0);
AudioConnection          patchCord5(mixer2, peak2);
AudioConnection          patchCord6(mixer2, bitcrusher2);
AudioConnection          patchCord7(mixer1, peak1);
AudioConnection          patchCord8(mixer1, bitcrusher1);
AudioConnection          patchCord9(bitcrusher2, 0, i2s2, 1);
AudioConnection          patchCord10(bitcrusher1, 0, i2s2, 0);
```

Teensy Graphics

- Use the Teensy-specific (xxx_t3) graphic libraries for most applications.
- Use the direct memory access (DMA) libraries for situations where fastest performance is required.
https://github.com/KurtE/ILI9341_t3n
- Graphics performance limited by speed of SPI bus between Teensy and the display.

Teensy Graphics — Demo

- Load the sketch:
[Teensy-R4ge-Pro/PlatformIO/teensy-r4ge-pro-shooter-demo](#)
- Demonstrates full screen scrolling, transparent drawing, and simple rotational movement.



Tombstone City

- [Teensy-R4ge-Pro/PlatformIO/teensy-r4ge-tombstone](https://teensy-r4ge-pro/platformio/teensy-r4ge-tombstone)
- An emulation of the classic TI-99/4A game by John Plaster for Texas Instruments
- Tile-based graphics



Gravitack

- [Teensy-R4ge-Pro/PlatformIO/teensy-r4ge-gravitack](#)
- Vector-based homage to the arcade console game



Ideas and Next Steps

Where do you go from here?

Topics

- More emulators
- Sensors and other peripherals
- Common Protocols (I2C, SPI, I2S, UART, CAN bus)
- Motor and servo control
- VGA Output
- Making your own PCBs
- Surface mount soldering

Project-Driven Discovery

Find something you want to do (real-world application)

...something that excites you

...something that you can accomplish in less than 1 month

...and figure out how to do it!

CodeMash Contest

Categories

- Audio Alchemy
- Maker Magic
- Technical Wizardry



Bring submissions to the Maker Space before noon on Friday for consideration. CodeMash staff will take a short video, along with your name and email. Must be present Friday afternoon to win.

Questions?



Resources

- Adafruit GFX Library
<https://learn.adafruit.com/adafruit-gfx-graphics-library/overview>
- PJRC Forum
<https://forum.pjrc.com/>
- Amp Hour Podcast / EEV Blog

Contact Info for Paul Pagel
pjpagel86@gmail.com

The words "THANK YOU" are displayed in a stylized, glowing neon font. The letters are outlined in a bright blue color with a double-line effect, giving them a three-dimensional, illuminated appearance. The text is set against a solid black background, which makes the glowing letters stand out prominently. The font is a clean, sans-serif style, with the letters slightly spaced out.