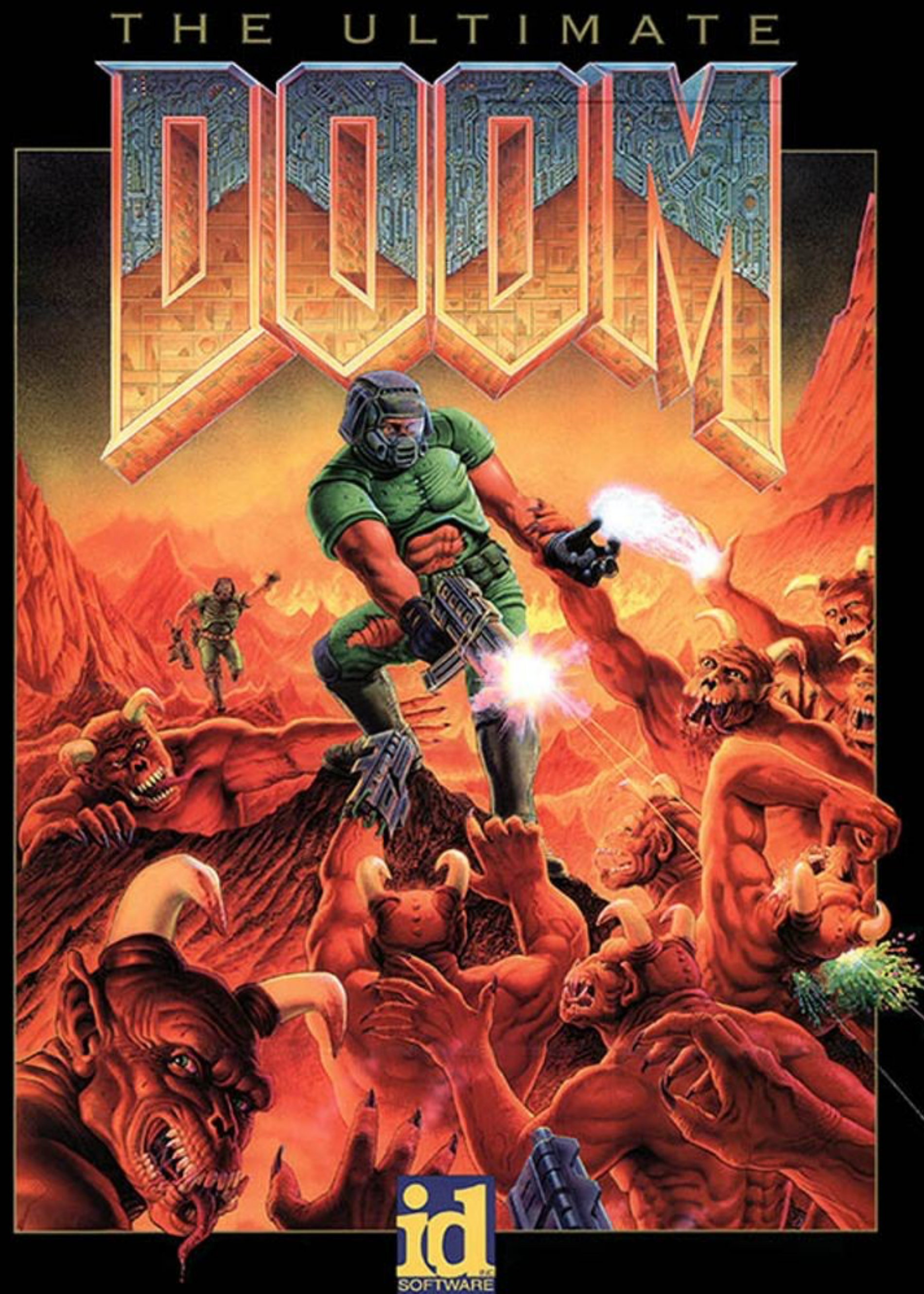


PicoDoom, a fork of GBA Doom

Dr. Olaf Flebbe

Will it run doom ?





# About me

PhD in computational physics  
(Theor. Astrophysik Tübingen)

Former projects: Minix68k (68k FP  
Emulation), Linux libm.so.5 (High  
Precision FP), perl and python for  
epoc, flightgear, msktutil

Member Apache Software  
Foundation, PMC Apache Bigtop

Lead Developer/System Specialist  
Bosch eBike





# Doom

- Was ich nicht erzähle:





# Motivation

- ✦ Embedded lernen
- ✦ Hack



# Microcontroller

- Was unterscheidet Microcontroller von einer CPU / Processer (zB Intel/AMD)?
  - Er ist viel billiger (1 EUR statt 100 EUR)
  - Langsamer (MHz statt GHz)
  - Wenig RAM Arbeitsspeicher (256 kB) auf dem Chip, statt gar keinem, dafür aber externen (8 GB++)
  - Keine virtuelle Speicherverwaltung, keine spezial Befehle für Grafik, Numerik, AI
  - GPIO, PWM, SPI, I2C, DMA (das haben Desktop CPU auch, aber nur indirekt)
  - Brauchen kein Betriebssystem
  - Sind heute leistungsfähiger als ein PC aus den 90 Jahren



# Geschichte ARM

- 1983: Das Team um den BBC Micro blitzen bei den herkömmlichen Herstellern ab: Motorola, Intel
- Acorn RISC Machine: Mit extrem wenig Ressourcen die CPU selbst entworfen : Sophie Wilson
- 1990 Apple braucht eine energiesparende und Schnelle CPU:  
Advanced RISC Machine, Apple Newton (Apple Architektur Lizenz)
- 2005 : 98% aller Handies haben ARM CPU
- Unterschiedliche Lizenzen: Apple verwendet „nur“ den Befehlssatz, Samsung etc verwenden die Funktionsblöcke um eigene CPU machen
- Raspery 2040 : Microcontroller : Dual Core Arm M0+



# PC 1993

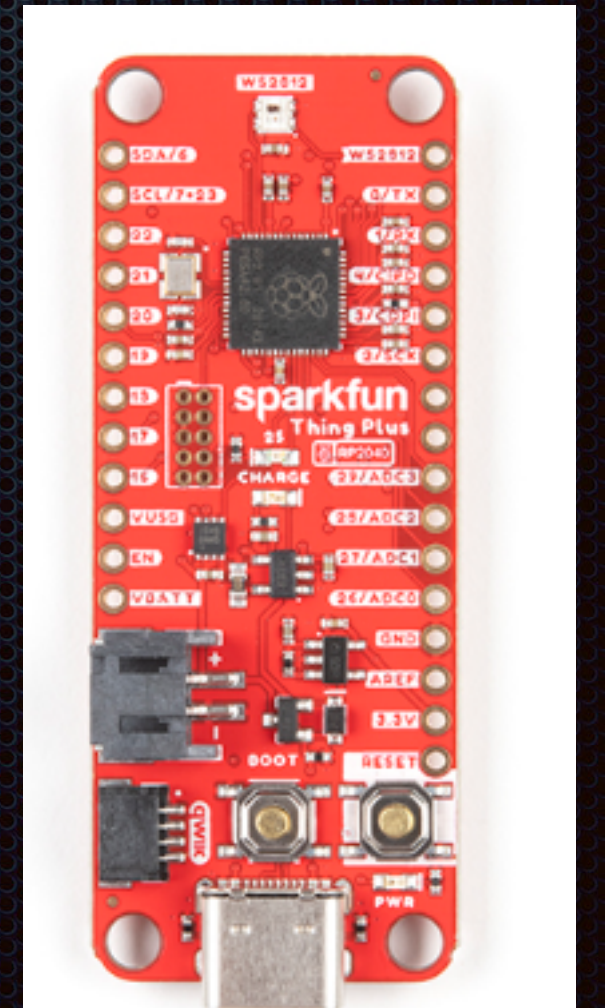
- ✦ 32Bit CPU
- ✦ 486 66Mhz
- ✦ RAM 4MB (8MB)
- ✦ Harddisk 100MB
- ✦ No Floating point
- ✦ VGA (16 colors)
- ✦ MSDOS / 32Bit Extender
- ✦ No HW Acceleration



# RP2040

## Spark thingplus

- ✦ 32 Bit MCU
- ✦ RP2040 133MHz (ARM M0+)
- ✦ Dual Core
- ✦ RAM 264kB
- ✦ Flash ROM 16MB
- ✦ No Floating point
- ✦ ST7789 240x240 16/24 Bit Color
- ✦ Bare Metal
- ✦ PIO, PWM, DMA, SPI





# Doom



Because of the nature of Moore's law, anything that an extremely clever graphics programmer can do at one point can be replicated by a merely competent programmer some number of years later.

— **John Carmack**





# Display ST7789 / LCD Module : Waveshare

The screenshot shows the Waveshare website's product page for a 1.54-inch LCD module. The top navigation bar includes the Waveshare logo, a search bar, and a shopping cart icon showing 0 items for \$0.00. The breadcrumb trail is: Home > Displays > LCD > 0.9" - 2.4" > 18079. A left sidebar lists various product categories, with 'Displays' selected. The main content area features a large image of the LCD module displaying colorful spheres. To the right of the image, the product title is '240x240, General 1.54inch LCD Display Module, IPS, 65K RGB'. Below the title, the SKU is 18079, the part number is '1.54inch LCD Module', and the brand is Waveshare. The price is listed as \$9.49, with a quantity selector set to 1 and an 'ADD TO CART' button. A table below the price shows bulk pricing: \$9.09 for 10+ units, \$8.89 for 50+ units, and \$8.81 for 100+ units. A 'Related Products' section is partially visible at the bottom.

WAVESHARE share awesome hardware

HOME PRODUCTS PAYMENT SHIPPING WIKI SUPPORT IC **NEW**

Search here  0 ITEM(S) - \$0.00

Home > Displays > LCD > 0.9" - 2.4" > 18079

**PRODUCTS**

- Raspberry Pi
- AI
- RISC-V
- Linux Computer
- Displays**
- IoT / Communication
- Misc Modules
- Robotics
- Arduino-Related / Nucleo
- micro:bit
- MCU / ARM
- FPGA
- Sockets / Adapters

**240x240, General 1.54inch LCD Display Module, IPS, 65K RGB**

SKU: 18079  
Part Number: 1.54inch LCD Module  
Brand: Waveshare

**\$9.49** 1

<b>\$9.09</b>	10+
<b>\$8.89</b>	50+
<b>\$8.81</b>	100+

Related Products:

A summary of the product specifications is presented in a grid format. Each specification is accompanied by a representative icon: a diagonal line for size, a grid for resolution, a circle of colored dots for display color, the text 'IPS' for the display panel, a connector pin header for the interface, and a microchip icon for the driver.

<b>Size</b>  1.54"	<b>Resolution</b>  240x240	<b>Display Color</b>  65K RGB
<b>Display Panel</b>  IPS	<b>Interface</b>  SPI	<b>Driver</b>  ST7789



# Tastatur

- Drucktaster Rund
- Dioden N4148





# Piromoni Tufty

- ✦ 8MB Flash
- ✦ st7789v parallel interface
- ✦ 320x240 LED





Planung



# Doom

- 1993 als Shareware für MSDOS veröffentlicht
- Ursprünglich auf NeXTStep entwickelt von John Carmack, John Romero
- Bahnbrechende 3D Grafik und Sound auf PC
- Shareware: WAD (where are all the data)
- 1994-2011: Jugendgefährdend Indiziert in D
- 1999: Quelltext unter GPL ohne Level veröffentlicht



# Welches Doom ?

- Chocolate Doom: Kann man einfach unter Mac, Windows, Linux compilieren. Originales Spiel
- prBoom/prBoom+ -> GBA Doom -> My pico doom (Speicher reduziert, 240x160 Auflösung)
- genericDoom ??
- NICHT rp2040 Doom von „kilogram“ . Wurde wenige Tage später veröffentlicht. DAS IST DER HAMMER!



# SDK Tooling

- ✦ rPi Toolchain (PICO\_SDK)
  - ✦ Checkout eines Git repositories
  - ✦ Arm gcc (zB von arm, oder linux distro, Mac Homebrew)
  - ✦ Multiarch gdb
  - ✦ Arm binutils
- ✦ Cmake
- ✦ Controller als USB Storage uf2 Datei kopieren fertig



# Makefile vs cmake

- ✦ Standard PICO\_SDK example nehmen und einfach alles compilieren

- ✦

```
zB  
cmake -GNinja ..  
ninja
```



# Speichermanagement

- ✦ ELF Segmente
  - ✦ .Text (Programm) -> FLASH (XIP Execute in Place)
  - ✦ R/O Data (const) -> FLASH
  - ✦ Data (Initialisierter Speicher) -> RAM
  - ✦ BSS (mit 0 initialisierter Speicher) -> RAM
  - ✦ Heap (allozierter Speicher) -> RAM



# Speichermanagement

- GBADoom: RAM reduziert, WAD ins Flash
- Tool das WAD Files in C Code convertieren
- PICO\_SDK Anpassung für grösseres Flash



# Auflösung

- 240x160
- Nichts mehr zu tun (GBADoom)



# Debugging

- ✦ printf Debugging:
  - ✦ Stdio konfigurieren als USB Serial oder UART
- ✦ Debug Probe über SWD Interface
  - ✦ Rpi Debug Probe: (RP2040 mit spez Firmware) Openocd
  - ✦ Selber RPI Flashen (Ich hatte nur Ärger)
  - ✦ Segger Jlink EDU mini (hier kaufen)





# st7789 Treiber

- Keinen fertigen Treiber gefunden
- Gestartet mit einem micropython SPI C Treiber  
[https://github.com/devbis/st7789\\_mpy/tree/master](https://github.com/devbis/st7789_mpy/tree/master)
- Dann micropython layer entfernt
- DMA Support hinzugefügt
- Piromoni Treiber für parallele st7789 Variante auf priromoni Tufty
- PIO Support
- Vereinheitlicht (Seitenprojekte Mandelbrot/Filme)



# SPI

- Serial Peripheral Interface
- Interface MCU <-> st7789
- Tut aus der Schachtel ([link](#))



# PIO Programmable I/O

- ✦ MCU Bit Bang Problemen: Ein Protokoll im korrekten Timing sprechen.
- ✦ Unter Linux ist das besonders schwer
- ✦ Wird aber zu einem Problem wenn man nebenbei noch was anderes tun muss
- ✦ Kleine Programme die Input und Output übernehmen



# PIO Programm / State Machine

```
.program st7789_parallel
.side_set 1

.wrap_target
    out pins, 8    side 0
    nop           side 1
.wrap
```

[https://github.com/oflebbe/GBADoom/blob/master/source/st7789\\_parallel.pio](https://github.com/oflebbe/GBADoom/blob/master/source/st7789_parallel.pio)

Paralleler Output mit 8 Pins und einem Side Pin



# DMA (Direct Memory Access)

- Parallel : DMA -> PIO port -> SM -> st7789
- SPI : DMA -> SPI Port -> st7789
- Nice API

```
static void write_blocking_dma(ST7789_t *self, const uint8_t *src, size_t len)
    dma_channel_set_trans_count(self->st_dma, len, false);
    dma_channel_set_read_addr(self->st_dma, src, true);

    dma_channel_wait_for_finish_blocking(self->st_dma);

    sleep_us(10); // We shouldn't overrun the ST7789
}
```



# Tasten:

Problem:

Nicht genug PINs für jeden Taste

Lösung:

Tastatur Matrix, P18-P22 Pulldown

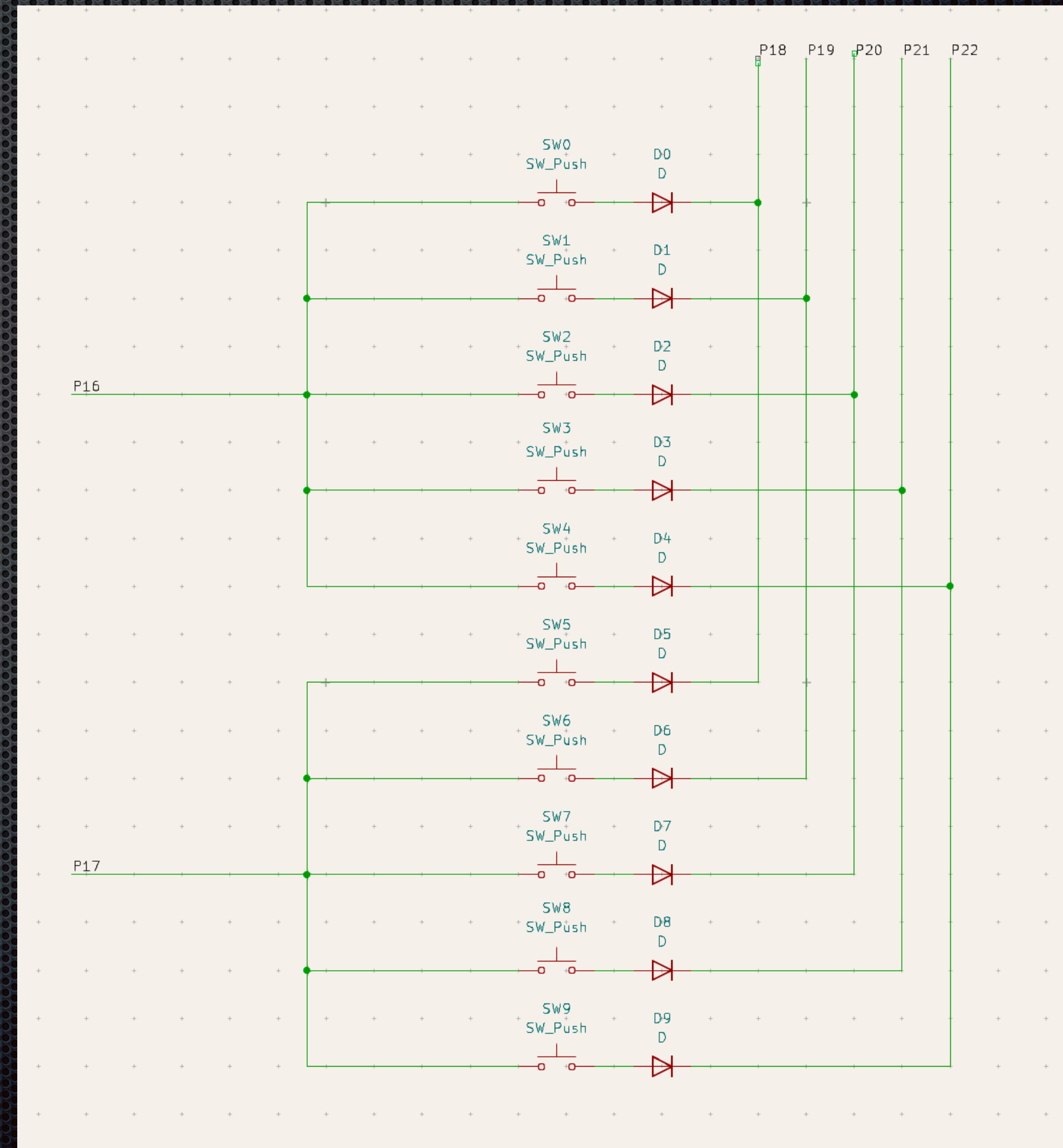
P16/P17 Drive Up/Down

Problem:

gleichzeitigs Drücken mehrerer Tasten

Lösung:

Diode zur Verhinderung von Kurzschlüssen und Scan





# Tastatortreiber

Zeitmultiplex: ([link](#))



# Fertig

<https://github.com/oflebbe/GBADoom>



# Fertig?

- ✦ Sound fehlt
- ✦ Netzwerk fehlt
- ✦ USB HID Tastatur
- ✦ VGA Output
- ✦ Dual Core



# Buchempfehlung:

