

Der eigene digitale (offline) Sprachassistent - das Ziel ist (fast) erreicht

FrOSCon 2023

Jürgen Pabel



Startup-Animation

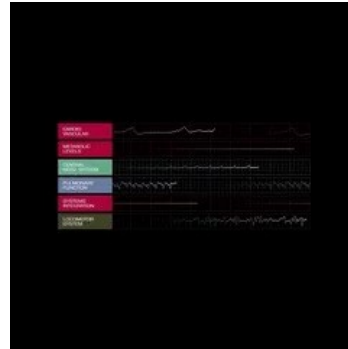


Bildmaterial: <https://github.com/armel/HAL9000/>
<http://www.jcheudin.fr/playground/playground/hal9000/>

Was bisher geschah....

- 2018 – 2022
 - Gestartet als reines 3D-Druck-Projekt (mit evtl. einer roten LED)...
 - <nichts/>
 - ...und auf einmal war es ein Hardware- & Software-Projekt
- FrOSCon 2022
 - Vorstellung des Projektes (mit Fokus auf technische Aspekte)
 - Aber: Teilweise grundlegende Einstiegs-/Übersichtsaspekte übersprungen
- 2022 - 2023
 - Bug fixes, code refactoring & new features
 - Projekt review => Projekt-Ziele

Shutdown-Animation

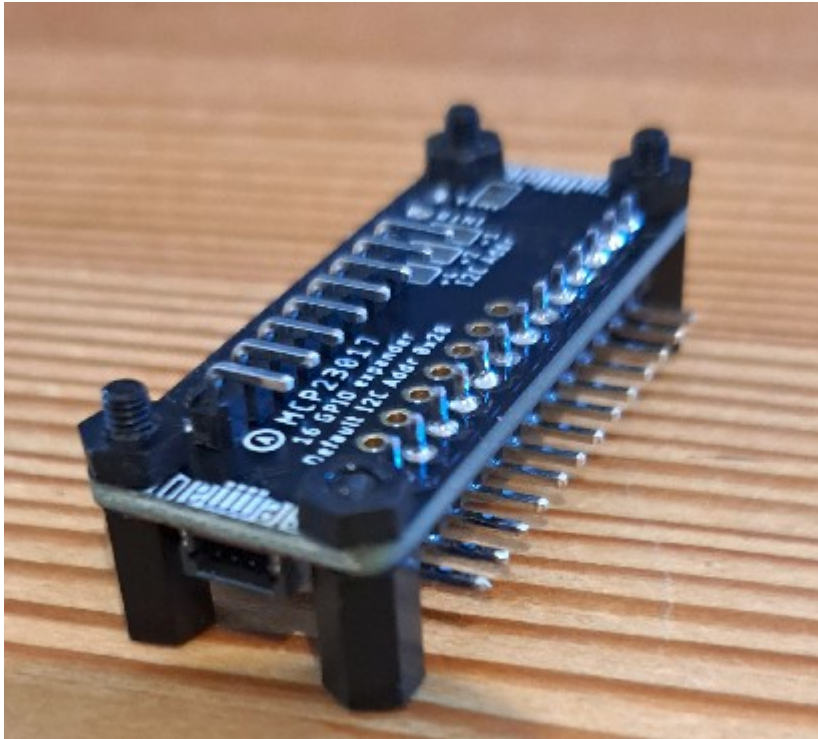


Bildmaterial: <https://github.com/armel/HAL9000/>

Projekt-Ziele

- **Qualität:** Marktfähiges Produkt (zumindest für Benutzer)
- **Usability:** Maker
 - 3D Druck-Designs sind auf gute Druckbarkeit optimiert
 - Elektronik sollte möglichst ohne löten/crimpen zu müssen aufbaubar sein
 - Software soll als Raspberry-Image bereitgestellt werden (inkl. geführter/automatisierter Firmware-Installation auf dem Microcontroller)
- **Usability:** User
 - Vollständige Konfiguration per WLAN (Web-Interface, keine App)
 - Fehlersituationen mittels Hilfe-Seiten (über QRcodes) abfedern
- **Kosten:** Explizit kein Projekt-Ziel (aber gerade noch akzeptabel, ca. 120-200€)

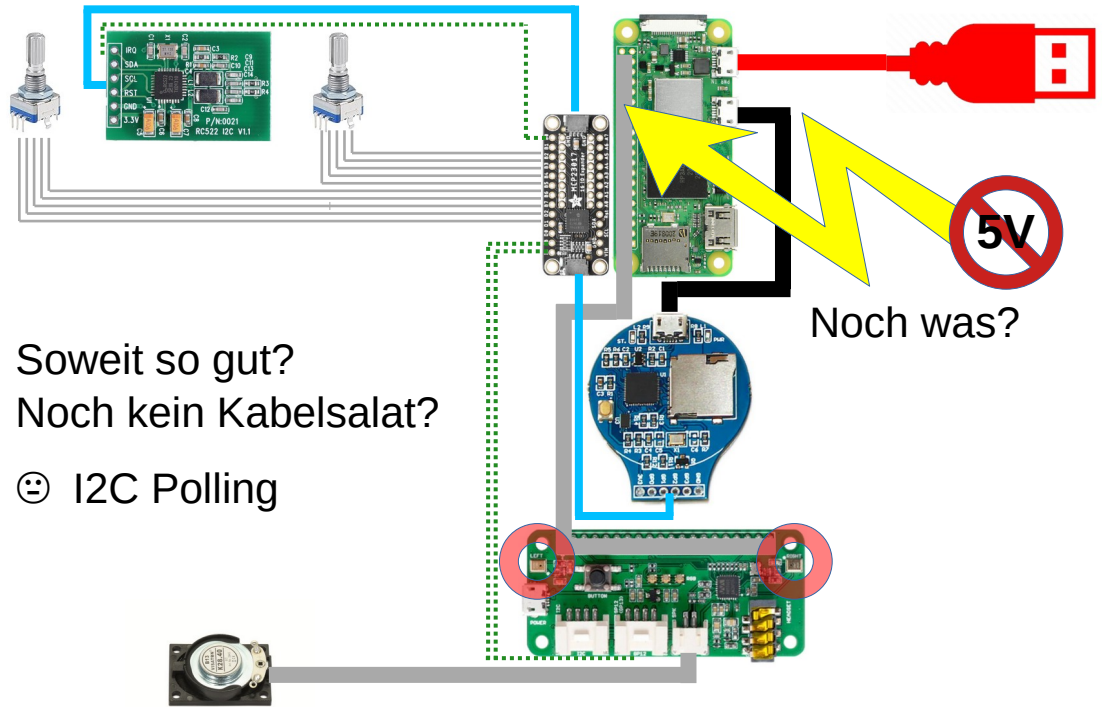
Hardware Flashback: MCP23017



- I/O Multiplexer über I2c
 - 2x8 GPIO-Bänke mit Pullups
A[0-7] & B[0-7]
 - 2 Interrupt PINs (☺ I2C Polling)

Diese Folie wurde aus
mindestens 80%
recycle-ten Bits erstellt.

Hardware: System-Aufbau



Hardware: MCU-Boards

- SBC RoundyPi
 - 4 GPIOs (I2C + 2 GPIOs)
- M5stack Core2
 - Einfachste Verkabelung (USB + Grove)
 - 520KB RAM & 8MB PSRAM
 - USB-Serial immer online (CH9102 Chip)
- Waveshare RP2040-LCD-1.28
 - USB-C
 - USB-Serial immer online (CH9102 Chip)
 - 30 GPIOs....als 1.27mm GPIO-Stiftleisten (statt 2.54mm)

Software: Überblick

- **Linux** (Raspberry Pi Zero2)
 - Kalliope: Digitaler Sprachassistent (nur Patches und Plugins beigesteuert)
 - Enclosure
 - Brain => Zustand und Logik
 - Arduino => Ansteuerung des Microcontrollers
 - Console => Web-Interface zur Konfiguration
- **Microcontroller** (RP2040, ESP32, ...)
 - Firmware: I/O Dienstleister für den Raspberry (über USB)
 - Outputs => Display
 - Inputs => Drehknöpfe, RFID-Leser, Motion-Detection, ... (one MCP23X17 to rule them all)

Software: Runtime

- **Linux** (Raspberry Pi Zero2)
 - Applikationsserver: uWSGI mit Python3 plugin in full-dynamic mode (mules)
- **Microcontroller** (RP2040, ESP32, ...)
 - PlatformIO
 - C/C++ nur mit static memory allocations (=> 264KB RAM @ RP2040)
 - PIO Libraries: ETL, TFT_eSPI, JPEGDEC, ArduinoJSON, qr-code-generator-library, Adafruit BusIO & MCP23017, Time
 - M5stack Core2: += XpowersLib
 - Threads
 - Core #2: MCP23X17 handler (☺ I2C polling)
 - Core #1: Alles andere

Software: Kommunikation Linux <=> MCU

```
USB(D->H): ["application/runtime", {"status":"booting"}]
USB(D->H): ["application/runtime", {"status":"configuring"}]
USB(H->D): ["device/mcp23X17", {"init": {"i2c-bus": 0, "i2c-address": 32}}]
USB(H->D): ["device/mcp23X17", {"config": {"device": {"type": "rotary", "name": "control", "inputs": [{"pin":
    "A1", "label": "sigA"}, {"pin": "A0", "label": "sigB"}]}}}]]
USB(H->D): ["device/mcp23X17", {"config": {"device": {"type": "rotary", "name": "volume", "inputs": [{"pin":
    "A6", "label": "sigA"}, {"pin": "A5", "label": "sigB"}]}}}]]
USB(H->D): ["device/mcp23X17", {"config": {"device": {"type": "button", "name": "control", "inputs": [{"pin":
    "A2", "label": "sigX"}], "events": {}}}}]]
USB(H->D): ["device/mcp23X17", {"config": {"device": {"type": "toggle", "name": "volume", "inputs": [{"pin":
    "A7", "label": "sigX"}], "events": {"high": "on", "low": "off"}}}}]]
USB(H->D): ["device/mcp23X17", {"config": {"device": {"type": "switch", "name": "motion", "inputs": [{"pin":
    "B1", "label": "sigX"}], "events": {}}}}]]
USB(H->D): ["device/mcp23X17", {"start": true}]
USB(H->D): ["", ""]
USB(D->H): ["application/runtime", {"status":"running"}]
USB(H->D): ["gui/screen", {"idle": {}}]
```

Änderungen seit der FrOSCon 2022 (1/2)

- Enclosure
 - Brain
 - Zustands-Tracking (activity, context, ...)
 - Erwartbare ALSA Lautstärkeregelung
 - Arduino
 - async python
 - Console
 - flet.dev basiertes Webinterface für Konfiguration & Status

Änderungen seit der FrOSCon 2022 (2/2)

- Microcontroller-Firmware
 - Startup & shutdown animations
 - Fehler-Codes mit Hilfe-URL per Qrcode
 - MCU-Abstraktions-Layer (RoundyPi, M5Stack-Core2, Waveshare RP2040, ...)
 - Migrationen auf ETL-Framework
 - Minimierung der Dependencies (`pngdec`, `pngle`, `simplewebserial`, `bsonpp`, ...)
 - Timestamp-Recovery bei Restarts

Ausblick

- Enclosure software, release 1.0 noch in 2022 (angestrebt)
- Diverse pull-requests für kalliope in der Mache
- Wake-word Lösung finden

FAIL

Diese Folie wurde aus
100% recycle-ten Bits
erstellt.

Ausblick (nun 2023)

- Release 1.0
 - Dokumentation
 - Entwickler
 - Maker
 - Benutzer (hauptsächlich Error-/Hilfe-Seiten)
 - Enclosure:Console vervollständigen
 - WLAN Konfiguration (comitup)
 - Dies und das fertigmachen (code cleanup, syslog, python async, ...)
 - RPI Installationsimage

Zum Schluss: Diverse Anekdoten

- Stromversorgung über USB2-Hub beim Systemstart nicht ausreichend
- systemd: time sync'ing (on boot & uwsgi-restart)
- Fritzing: „Nur wenige“ RPI & Arduino Parts
- Quellen für (mehr als 50) Kinderwitze?
- RIP mycroft.ai

Ende

Danke!
Fragen? Fragen!

Backup

...nur für den Fall, dass die Demo-Götter ungnädig sind...

Enclosure: Console

