

---

# Hardware Hacking 101

Christopher Scheuring

---

# /whoami

- Christopher „Chris“ (Scheuring)
  - Hacker, Security Researcher & Analyst
  - Offiziell IT/OT/Embedded Security Experte und freiberuflicher Dozent an der DHBW Mosbach und Karlsruhe
- Mehr auf der Offensive-Seite...
  - Hauptfokus: IoT / OT&ICS / HW / Multi-Tier-Umgebungen / Mobile und SDR Stuff
  - Ebenso Automotive und andere Embedded Systeme
- Kontakt:
  - [chris+ccc@aucmail.de](mailto:chris+ccc@aucmail.de)
  - [@0x4045494650@chaos.social :: @0x4045494650](https://@0x4045494650@chaos.social)

---

# Warum das Ganze?

---

# Wieso ein Hardware (HW) Hacking 101?

- Embedded und IoT Hardware wird immer günstiger und einfach zu Programmieren
  - (Meist) kein Assembler notwendig - Programmierung in LUA, MicroPython, C/C++
  - Arduino, ESP usw.
- Kaum jemand macht sich Gedanken über Security
  - Oder: Viele Entwickler denken, dass ja keiner der HW öffnet und/oder es ist ja nur ein einfaches Stück HW für eine dedizierte Anwendungen...
- Die Hacking-Hardware ist teuer...
- Was kann schon schief gehen ;-)

---

# Was spricht für mehr Security?

---

# Gummihammer

Sehr zuverlässig bei geklebten  
Gehäusen - und richtet (meist)  
wenig Schaden an :-)



---

# Dremel / Mini-Flex / Puk-Säge

Wenn der Gummihammer nicht funktioniert -  
dann geht es mit ein wenig "Gewalt" ;-)



---

# Am Ende geht es um eine Platine :-)

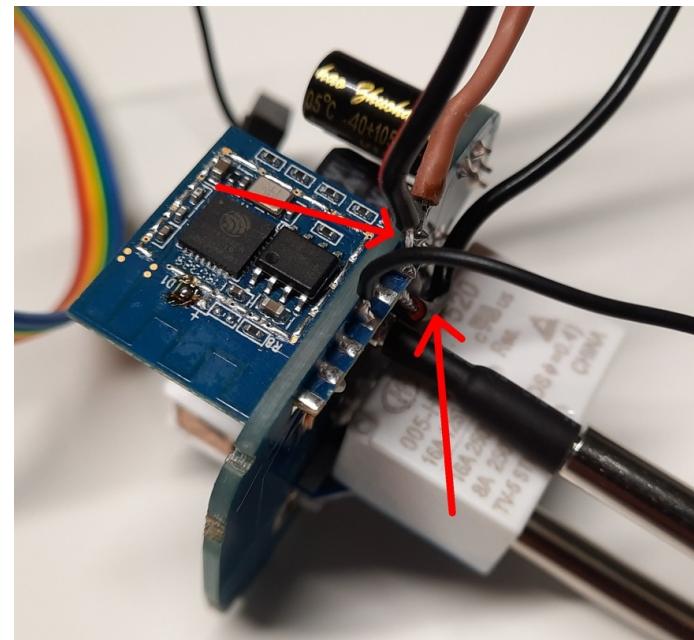
Wo Kabel angelötet werden können...

Firmware usw. ausgelesen werden kann...

Um z.B. an Credentials zu gelangen...

Oder die Funktion zu verstehen...

Oder eigene Firmware aufzuspielen...



---

# Die Idee des Test-Boards

---

# Ziele und Motivation (1)

- Das ganze ist als Idee für eine IT-Security Vorlesung an der DHBW entstanden
- Erkennen von Schnittstellen und Schwachstellen
  - Hardware-Design
  - Software-Implementierung
- Warum?
  - Aus Security-Sicht muss auch die HW betrachtet werden
  - Schlechtes SW-Design kann zu erfolgreichem Hack führen

---

## Ziele und Motivation (2)

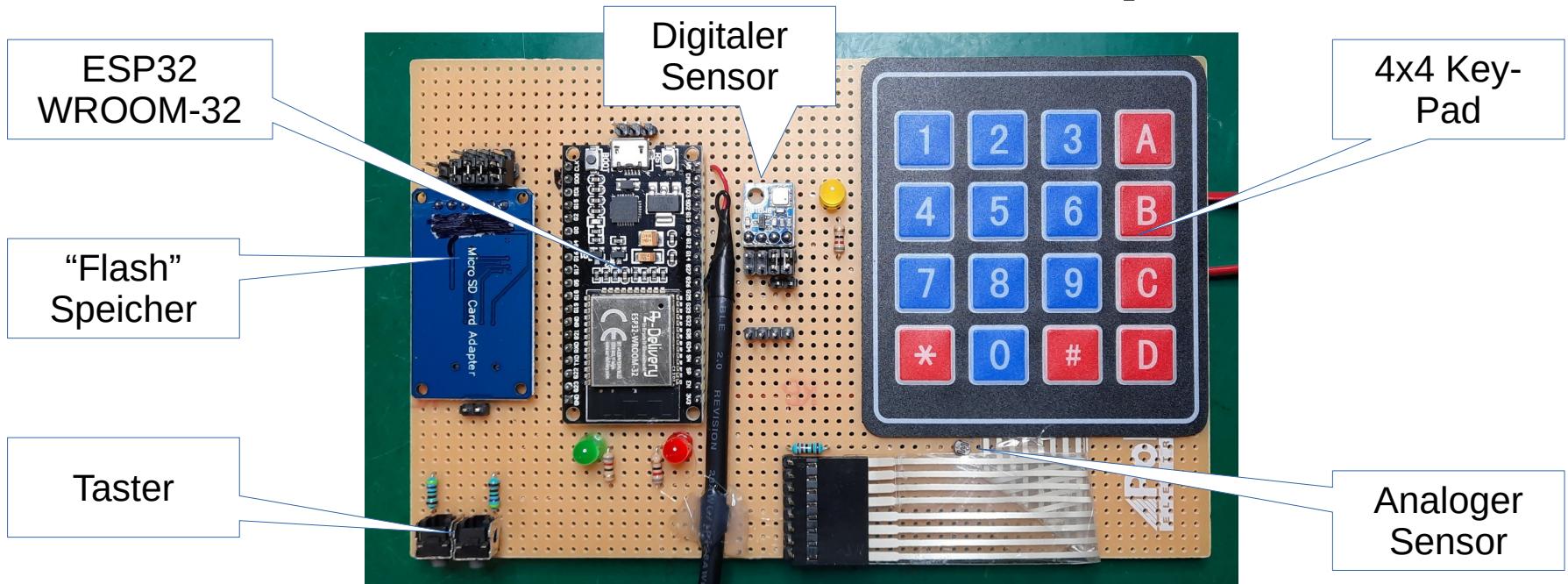
- Umgang mit
  - Multimeter
  - Oszilloskop
  - Logic Analyzer
  - \$Adapter (TTY, CAN, RS485 etc.)
- Ausnutzen von Schwachstellen / Designfehlern / Hidden Functions

---

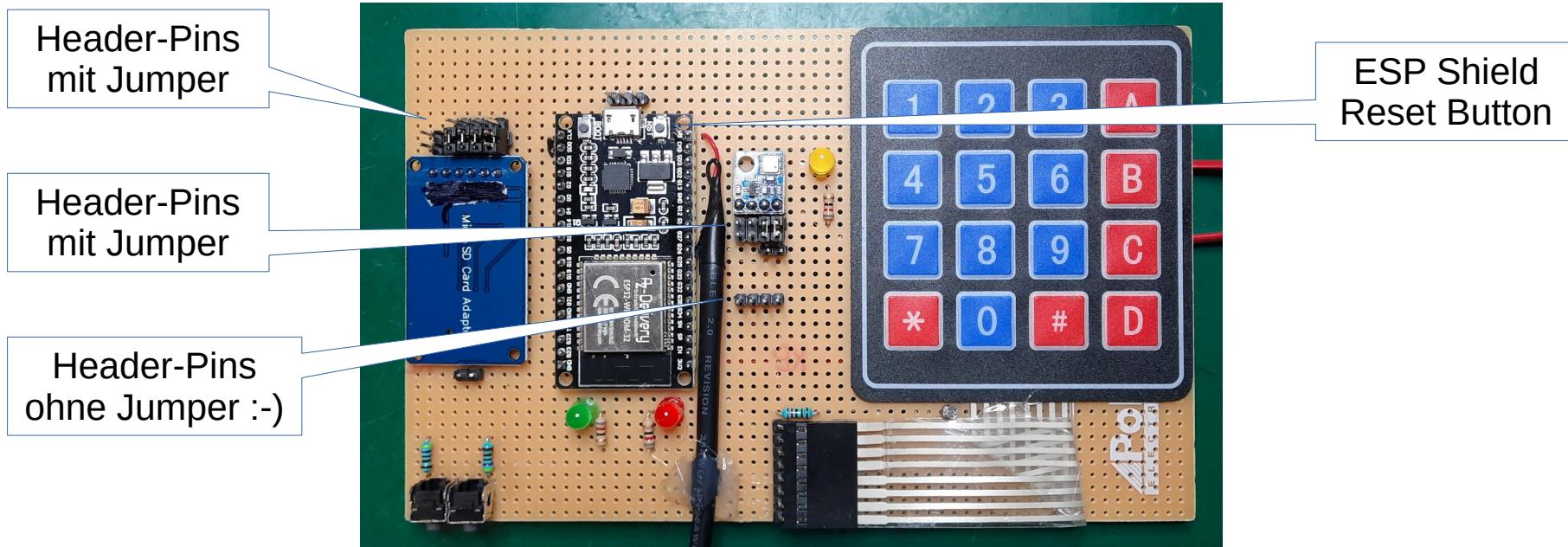
# Ziele und Motivation (3)

- Standard-Tools gibt es für kleine Geld und es wird kein Hightech-Equipment benötigt:
  - Multimeter => 15,- bis 25,- EUR
  - Logic Analyzer => China Saleae 10,- bis 15,- EUR (reicht für das meiste aus)
  - \$Adapter (TTY, CAN, RS485 etc.), Jumper-Kabel => je 2,- bis 6,- EUR
  - Lötkolben – nicht zwingen, aber macht das Leben leichter => 50,- EUR
  - Optional: „kleines“ Digital-Oszilloskop => 100,- bis 300,- EUR
- **Alles zusammen gibts ab ~100,-EUR + 100 EUR für optionales Oszi**

# Das Test-Board und seine Komponenten



# Das Test-Board und seine Komponenten



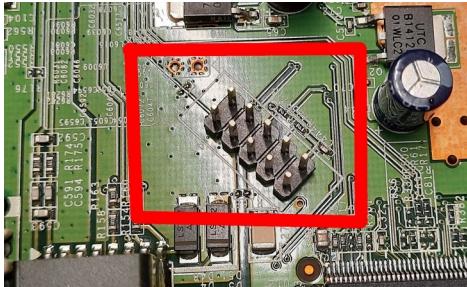
---

# Auffinden von Schnittstellen und Zugriff darauf

---

# UART usw. auf Platinen / Embedded Geräten

- Meist ist keine Sub-D Stecker wie die RS-232 bei einem PC verfügbar
- Wenn man Glück hat existiert ein Pfostenstecker oder zumindest eine entsprechende Vorsehung auf der Platine
- Ansonsten muss man die ICs auf der Platine analysieren und ggf. Kabel direkt an die Pins der ICs oder andere taktisch gut gelegene Stellen auf der Platine anlöten.



---

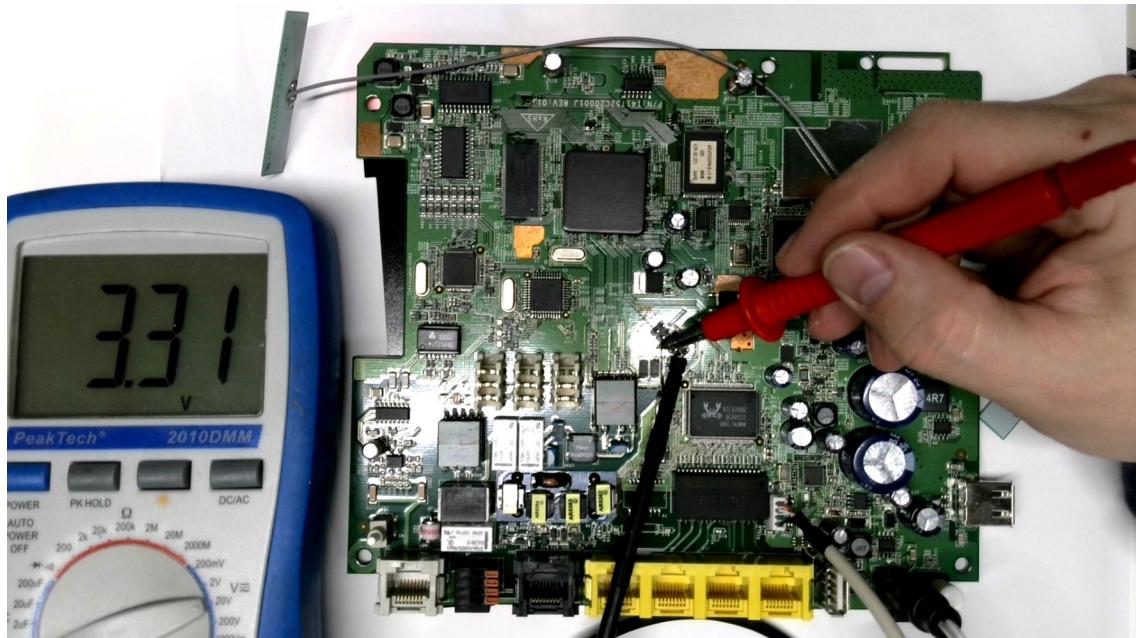
# UART auf Platinen / Embedded Geräten

- Mit das wichtigste Tool für den UART Zugriff sind entsprechende Adapter
- RS-232 arbeitet mit mind +/-12 V => daher kein USB-TTL Adapter nutzen!
- Ansonsten sollte man auf darauf achten, mit welchen Pegel gearbeitet wird.
  - TTL Standard nutzt 5,0 V
  - Neuere Systeme nutzen oft 3,3 V oder sogar 1,8 V
  - Daher: Eine zu hohe Spannung für TX (transmit data) kann den UART-Anschluss auf dem System oder des Adapters zerstören.

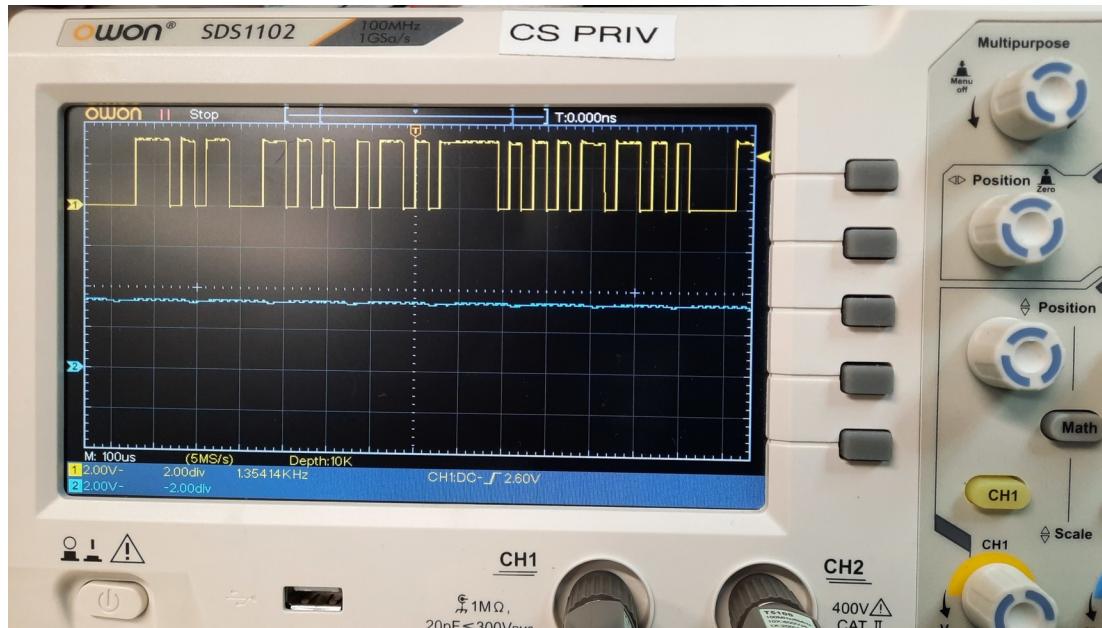


---

# Messung der Spannung mit Multimeter

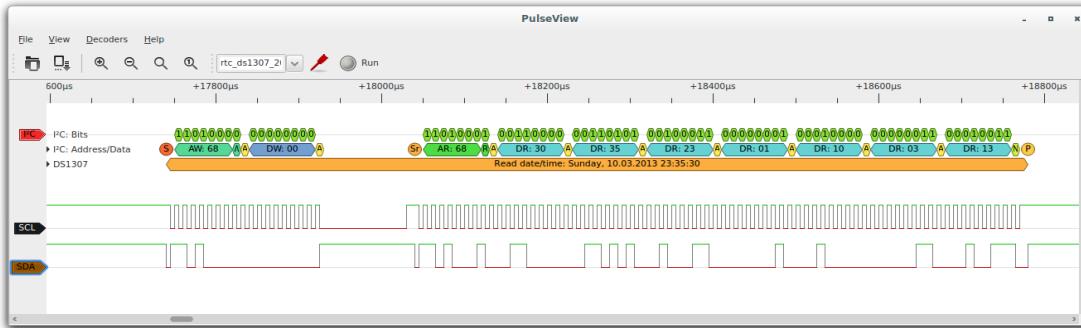


# Analyse des Signals mit Oszilloskop

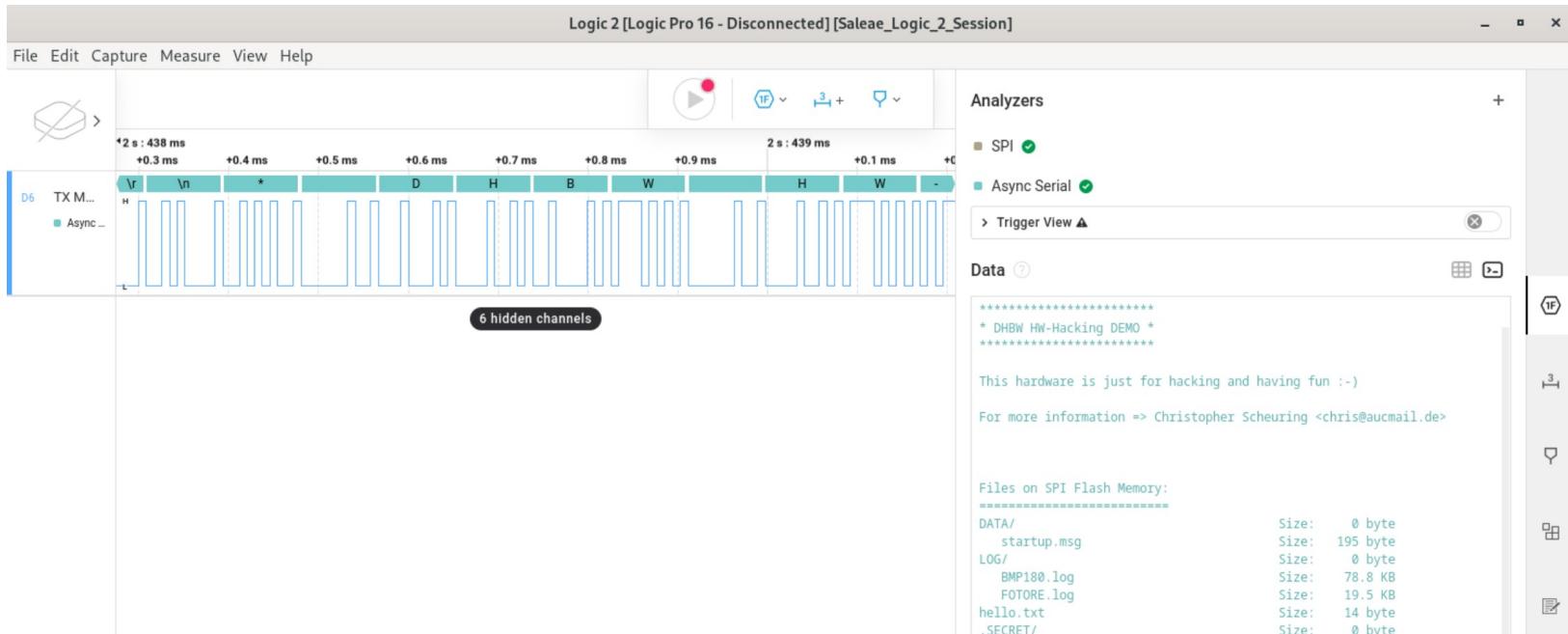


# Analyse des Signals mit Logic-Analyser

- Da ein Oszilloskop nicht gerade günstig ist, bietet sich die kostengünstige Version Spannungsmessung und “15,- Euro” 24 MHz Logic-Analyser an (China Saleae Fake) an.
- Unter Linux kann dieser Logic-Analyser mit Hilfe der Sigrok-Tools und PulseView einfach genutzt werden
  - Sigrok-cli: <https://sigrok.org/wiki/Sigrok-cli>
  - PulseView (GUI): <https://sigrok.org/wiki/PulseView>



# Analyse des Signals mit Logic-Analyser



---

# Die Challanges

---

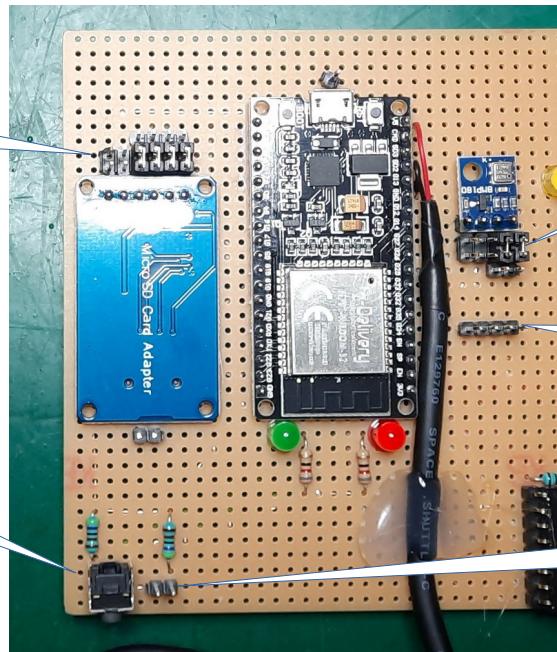
# Challenge: HW und Funktionsanalyse

- Ermitteln der Pins und deren Funktion mit Hilfe von
  - Multimeter
  - Logic-Analyzer => welche Funktion haben die verschiedenen Pins?
    - Erstellen Sie sich eine Skizze mit den Pins und deren Funktion/Belegung.
  - TTY USB Adapter => Welche Details liefert die Serielle Schnittstelle?
- Welches Verhalten kann beim Ziehen einzelner Jumper identifiziert werden?
- Welche Funktionen haben die verschiedenen Sensoren und Taster?
  - Welches Verhalten können Sie erkennen?

# HW-Analyse - Pins und deren Funktion

SPI Schnittstelle  
Ist auf jeden Fall  
wichtig.

Taster  
Löst Meldung in  
Serieller-Konsole aus



I2C  
Schnittstelle

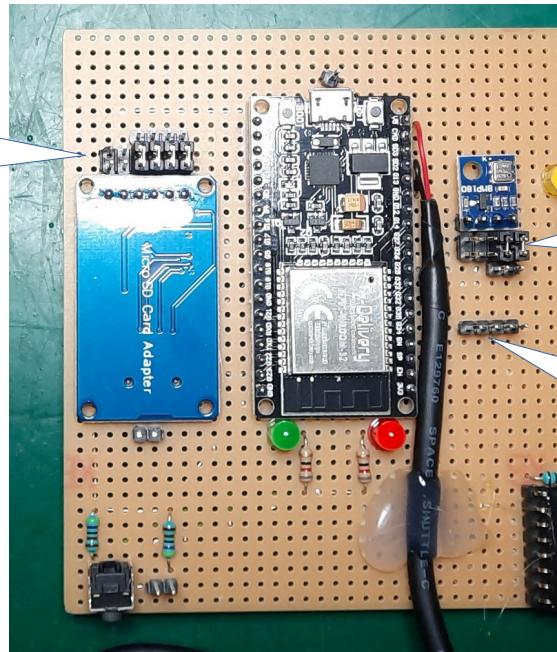
UART/TTY  
Serielle Schnittstelle  
nur TX-Data

Macht  
erstmal nix

# HW-Analyse - Pin Belegung (links nach rechts)

1 GND  
2 Vcc 5V  
3 MISO  
4 MOSI  
5 SCLK  
6 CS

**ACHTUNG:**  
3.3V Pegel



1 Vcc 3.3V  
2 GND  
3 SCL  
4 SDA

3.3V Pegel

1 n.c.  
2 TX Data  
3 GND  
4 GND

3.3V Pegel

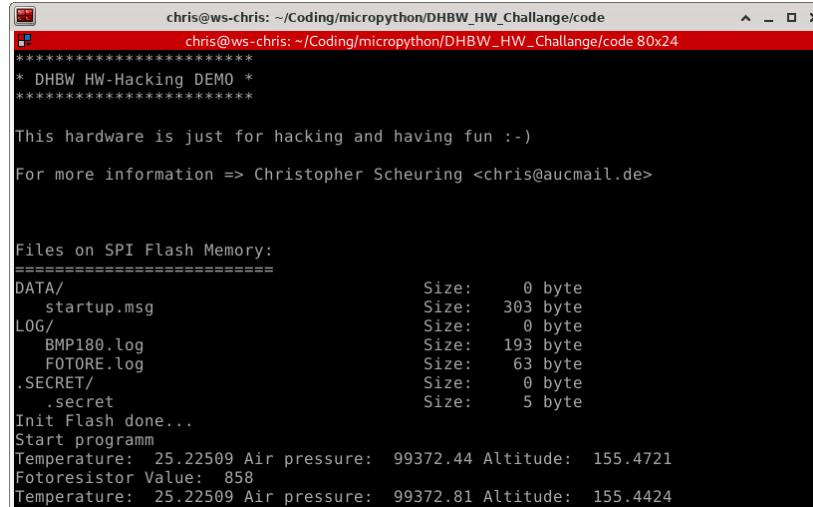
X-Ver:  
1 GND  
2 GND  
3 TX Data  
4 n.c.

---

# Funktions-Analyse

- SPI Schnittstelle dient zum Datenaustausch zwischen dem ESP und Flash-Speicher (Standard SD-Card Zugriff auf FAT-Ebene)
- I2C Schnittstelle dient zum Auslesen der BMP180 Sensorwerte (Temperatur, Luftdruck und relative Höhe)
- Werden einzelne Jumper von der SPI und I2C Schnittstelle entfernt, erscheint eine Fehlermeldung in der Seriellen Konsole.
- Serielle Schnittstelle stellt eine “read-only” Konsole zur Verfügung, die verschiedene Systemzustände anzeigt.
- Der analoge Sensor ist ein Helligkeitssensor und gibt Werte von 0 (absolut dunkel) bis 4095 (max. hell) aus.
- Der Taster erzeugt folgende Meldung in der Seriellen Konsole: „Case is open - no Pin entry possible!“
- Der Jumper Header neben dem Taster scheint Funktionslos zu sein, wenn man diesen überbrückt.

# Funktions-Analyse Serielle Konsole



```
chris@ws-chris: ~/Coding/micropython/DHBW_HW_Challange/code
chris@ws-chris: ~/Coding/micropython/DHBW_HW_Challange/code 80x24
*****
* DHBW HW-Hacking DEMO *
*****


This hardware is just for hacking and having fun :-)

For more information => Christopher Scheuring <chris@aucmail.de>

Files on SPI Flash Memory:
=====
DATA/
    startup.msg
LOG/
    BMP180.log
    FOTORE.log
.SECRET/
    .secret
Size:      0 byte
Size:  303 byte
Size:      0 byte
Size:  193 byte
Size:   63 byte
Size:      0 byte
Size:      5 byte

Init Flash done...
Start programm
Temperature: 25.22509 Air pressure: 99372.44 Altitude: 155.4721
Fotoresistor Value: 858
Temperature: 25.22509 Air pressure: 99372.81 Altitude: 155.4424
```

- Nach dem anlegen der Betriebsspannung startet das Systems, es erscheint eine Begrüßungsmeldung und Details zum angebunden Flash-Filesystem.
- Danach werden alle 2 Sekunden die Werte des BMP180 und dem Lichtsensor ausgegeben.
- Aktionen lösen weitere Meldungen in der Konsole aus.

---

## Challenge: Key-Pad und die PIN

- Finden Sie einen Weg, um das Key-Pad zu aktivieren.
- Finden Sie einen Weg, um an die PIN zu gelangen.
- Hinweis: Achten Sie auf die Ausgabe der Seriellen Konsole.
  - Was können Sie daraus schließen, um an die PIN zu gelangen?
  - Denke Sie auch an die Übungen zur Analyse von über SPI übertragene Daten.

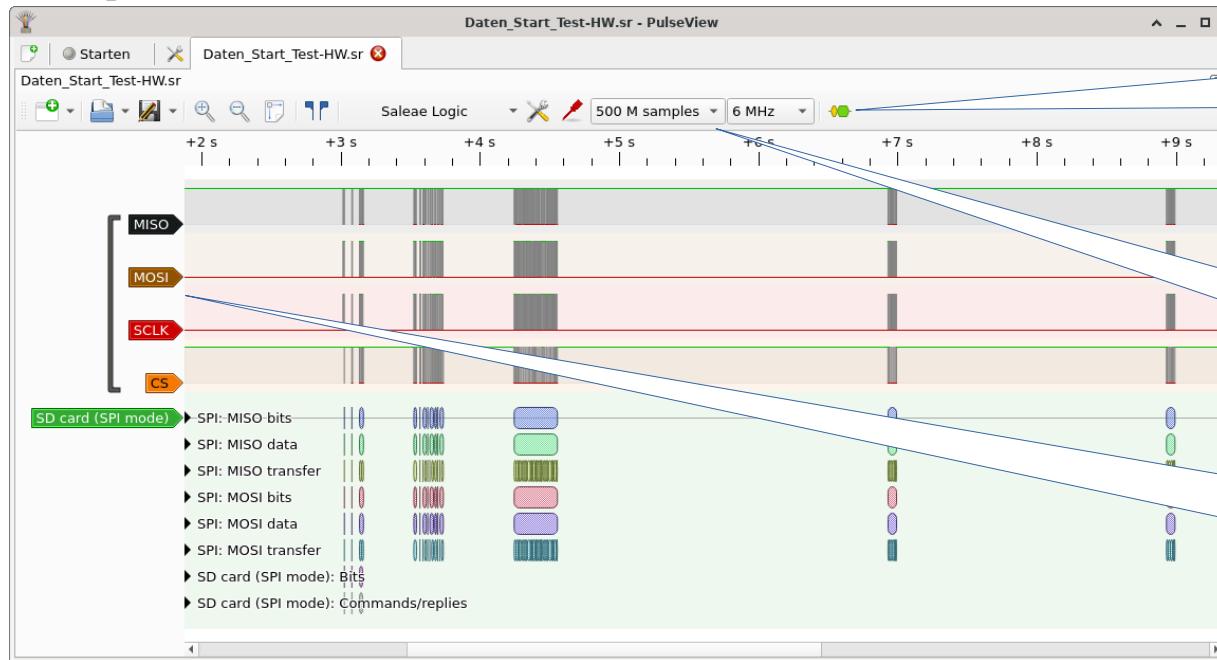
---

# Key-Pad und die PIN

- Finden Sie einen Weg, um das Key-Pad zu aktivieren.
  - Es muss der Taster gedrückt und gleichzeitig muss der Lichtsensor komplett abgedunkelt werden. Es erscheint die Meldung:

```
Case is closed :: Start KeyPad activation for 10 seconds.  
Enter Pin (Confirm with #-Key):
```
- Finden Sie einen Weg, um an die PIN zu gelangen.
  - Nutzen Sie *Pulseview* um die SPI Kommunikation mit zu schneiden und extrahieren Sie mit Hilfe von *sigrok-cli* die übertragenen Daten.
  - Die PIN wird beim starten neu erzeugt und in die Datei `.SECRET/.secret` abgelegt.

# Key-Pad und die PIN - SPI sniffen



Den „SD Card (SPI Mode)“ Decoder auswählen

500 M Samples + 6 MHz Abtastrate reichen aus

Am besten die Kanäle Beschriften wie im Pin-Layout identifiziert.

---

# SPI Analyse der Daten

- Die mit Hilfe von *Pulseview* aufgezeichneten Daten beim Starten des Systems (am besten warten, bis mind. 1-2 Sensorwerte in der Konsole angezeigt wurden) in eine Datei zur weiteren Verarbeitung mit *sigrok-cli* abspeichern (z.B. *Daten\_Start-Test-HW.sr*).
- Folgende Befehle extrahieren die Klartext Information:
  - Daten Senden Flash-Speicher:  
`$ sigrok-cli -i Daten_Start-Test-HW.sr -P  
spi:wordsize=8:miso=MISO:mosi=MOSI:clk=SCLK:cs=CS -B spi=mosi | strings`
  - Daten Empfangen Flash-Speicher  
`$ sigrok-cli -i Daten_Start-Test-HW.sr -P  
spi:wordsize=8:miso=MISO:mosi=MOSI:clk=SCLK:cs=CS -B spi=miso | strings`
  - Anstatt `| strings` kann auch `| hexdump -c` genutzt werden (zum besseren Vergleich in *Pulseview* z.B.)

# PIN und PIN Eingabe

- Die gesendeten Daten enthalten folgenden String:

..

**SECRET~1**

**ECRET~1SWP**

**5777D <= Das ist die erzeugte  
und auf dem Flash  
gespeicherte PIN.**

```
chris@ws-chris: ~/Coding/micropython/DHBW_HW_Challange/code
chris@ws-chris: ~/Coding/micropython/DHBW_HW_Challange/code 80x24
Fotoresistor Value: 0
Case is closed :: Start KeyPad activation for 10 seconds
Enter Pin (Confirm with #-Key):
3
31
313
3133
3133D
KeyPad Timeout
3133D

Entered Pin OK ::: Entering Admin Mode

You have found the correct switch usage and identified to cover the light
sensor, to enable the PIN Entry Mode.

Congratz: You have solved the Admin-Menu Challange :-)

Nothing more to do here - just wait 2 seconds
Temperature: 25.41686 Air pressure: 99388.31 Altitude: 154.196
Fotoresistor Value: 582
Temperature: 25.41686 Air pressure: 99390.52 Altitude: 154.0183
Fotoresistor Value: 597
```

---

# Challenge: Versteckte Funktion

- Der Jumper Header zeigt bisher beim Überbrücken mit z.B. einem Kabel vom Logik-Analyse keine Reaktion.
- Gibt es einen Weg, um darüber in Kombination mit einer weiteren Aktion eine versteckte Funktion auszulösen?
- Hint: Manchmal muss man in die Extreme gehen – kein Feuerzeug, Lötkolben, Gummihammer usw. ;-)

---

# Analyse Versteckte Funktion

- Sobald der Jumper Header gebrückt ist und der Lichtsensor mit einer sehr hellen Lichtquelle (Smartphone Blitz-LED z.B.) steigt der Lichtsensor-Wert auf das Maximum von 4095 und eine versteckte Funktion wird aktiviert:

```
Entering hidden REPL prompt (micropython console)...
```

```
You have found the correct switch usage and identified to flash the light
sensor, to enable this hidden function.
```

```
Congratz: You have solved the Hidden-Function Challenge :-)
```

```
You will get an REPL console.
```

```
For restart, just press the reset button of the ESP32 Wroom Shield.
```

```
Enhanced Challenge:
```

```
Try to enable the RX-Data Pin for getting complete REPL access!
```

```
Hint: Maybe you need a third "hand" and need to ask for additional stuff ;-)
```

---

# Weitere Challanges

- RX-Data ist nicht mit der Seriellen Konsole verbunden; daher kann die Micropython REPL nicht direkt genutzt werden.

**Challange:** Finden Sie einen Weg, RX-Date für die Eingabe von Befehlen in der REPL Eingabe nutzen zu können

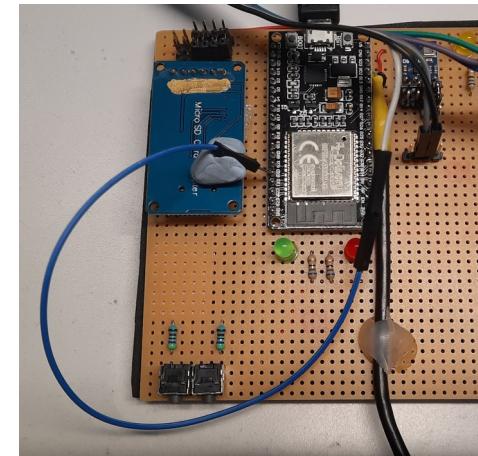
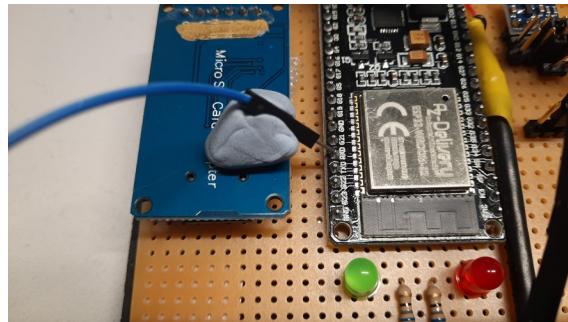
- Welche zusätzlichen Probleme im Code können Sie mit dem Konsolen-Zugriff noch identifizieren?
- Beim Starten des Anwendung existiert eine Race-Condition mit der das Programm zum Absturz gebracht werden kann und in einem undefinierten Zustand gelangt.

**Challange:** Finden Sie den Race-Condition – wie kann man diese mit Hilfe der Konsolen-Ausgabe und dem SPI Mitschnitt zeitlich eingrenzen?

---

# Challenge RX-Data und Konsolenzugriff

- Aufgrund des ESP32-Shield Design ist der Zugriff auf die Serielle Konsolen RX-Data Line einfach möglich. Der Pin mit der Bezeichnung RX muss einfach mit TX des USB-TTY Adapters verbunden werden.
- Sobald Sie der RX-Line nutzen können, kann das laufende Programm jederzeit mit CTRL-C abgebrochen werden! Sie erhalten so auch Zugriff auf die REPL



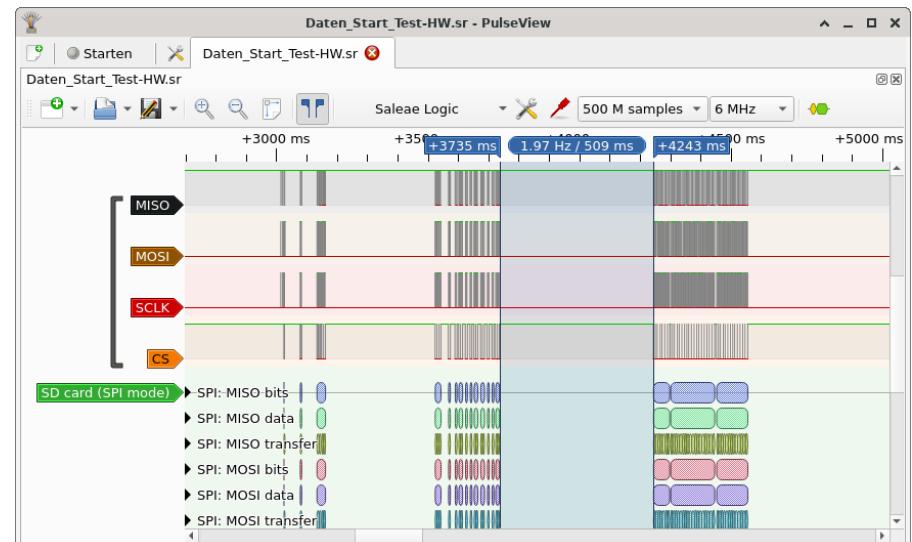
# Challenge Race-Condition (1)

- In der Konsole kann beim Starten eine kleine Verzögerung zwischen den beiden Ausgaben

**Init Flash done...**  
**Start programm**

beobachtet werden.

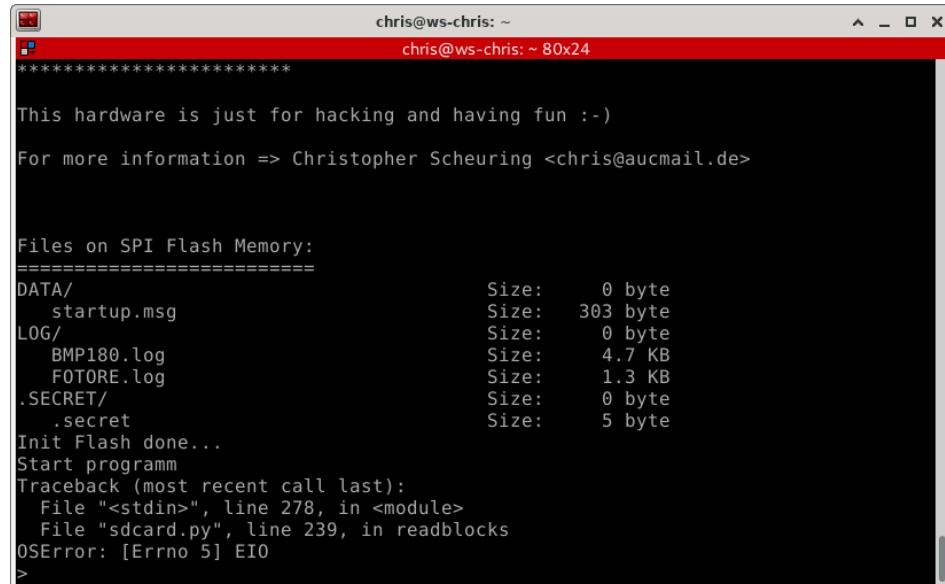
- Im SPI Mitschnitt kann diese Stelle auch erkannt werden, da hier eine Verzögerung von 500ms zwischen den Auslesen der Start-Nachricht und dem anlegen der Log-Daten zu erkennen ist.



---

# Challenge Race-Condition (2)

- Diese Verzögerung von 500ms könnte auf ein „Warten“ auf den Flash-Speicher etc. sein.
- Durch gezieltes ziehen eines Jumpers im SPI-Block innerhalb dieser Verzögerung, wird eine nicht abgefangene Exception erzeugt, die das Programm zum abstürzen bringt. Der Micropython Interpreter befindet sich dann in einem undefinierten Zustand.



chris@ws-chris: ~

chris@ws-chris: ~ 80x24

```
*****
This hardware is just for hacking and having fun :)

For more information => Christopher Scheuring <chris@aucmail.de>

Files on SPI Flash Memory:
=====
DATA/                           Size:    0 byte
    startup.msg                  Size: 303 byte
LOG/                            Size:    0 byte
    BMP180.log                   Size:   4.7 KB
    FOTORE.log                   Size:   1.3 KB
.SECRET/                         Size:    0 byte
    .secret                      Size:    5 byte
Init Flash done...
Start programm
Traceback (most recent call last):
  File "<stdin>", line 278, in <module>
    File "sdcard.py", line 239, in readblocks
      OSError: [Errno 5] EIO
>
```

---

Only DEMO  
Hidden Functions...

---

# Weiter Hidden Functions - DEMO

- Beim BMP180 können die Temperatur-Extreme dazu genutzt werden, zwei unterschiedlichen versteckte Funktionen aufzurufen.
- Bei der Durchführung muss man sehr vorsichtig vorgehen – es könnte dabei kaputt gehen!
- Extreme Hitze über 80°C startet eine versteckter Funktion.
- Extreme Kälte unter -30°C startet ebenfalls eine versteckter Funktion.
  - Da der BMP180 Sensor eine kleine Öffnung für die Luftdruckmessung besitzt, fällt der Sensor nach dem Einfrieren eine Weile aus, bis das sich gebildete Kondenswasser aus dem Sensor entwichen ist.

---

# Fragen? Diskussion?

---

# Wo finde ich mehr dazu?

- Details zur Hardware, Sourcecode, mehr Doku, Übungen usw:
  - <https://gitfoo.0x17sec.de/DHBW-Stuff/hw-hacking-101>
  - Alles (soweit möglich) ist unter GPL und CC veröffentlicht
- Oder eMail an mich ;-)
  - [chris+ccc@aucmail.de](mailto:chris+ccc@aucmail.de)
- Link zu den Slides
  - [https://gitfoo.0x17sec.de/DHBW-Stuff/hw-hacking-101/src/branch/main/Hardware\\_Hacking\\_Board\\_Vorstellung\\_DE.pdf](https://gitfoo.0x17sec.de/DHBW-Stuff/hw-hacking-101/src/branch/main/Hardware_Hacking_Board_Vorstellung_DE.pdf)