Embedded Security Continued

Selected Topics on Hardware for Security (NWI-IMC065)

Ileana Buhan, November 2021

EOS N 01.760, Tuesdays 10.30-12.15

Radboud University

Recap previous lecture

1. Who is TOM?
2. What is Hardware Security?
3. What do we look for when analyzing a PCB?
4. What is the UART?
5. How many PINs for UART?

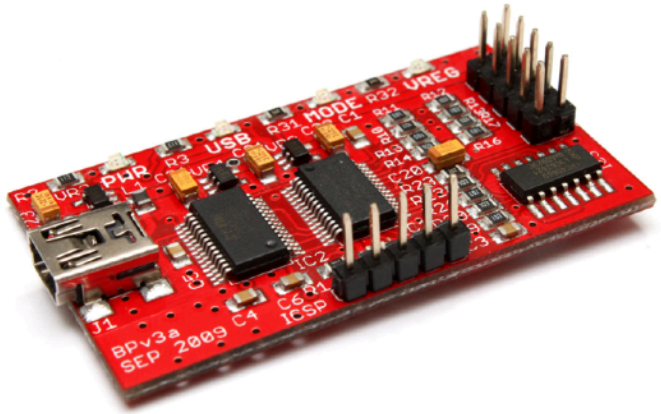
This lecture

- More serial communication:
  - JTAG
  - SPI
  - I2C

- Survival in a hostile environment
  - Built-in secure hardware
  - The bootloader

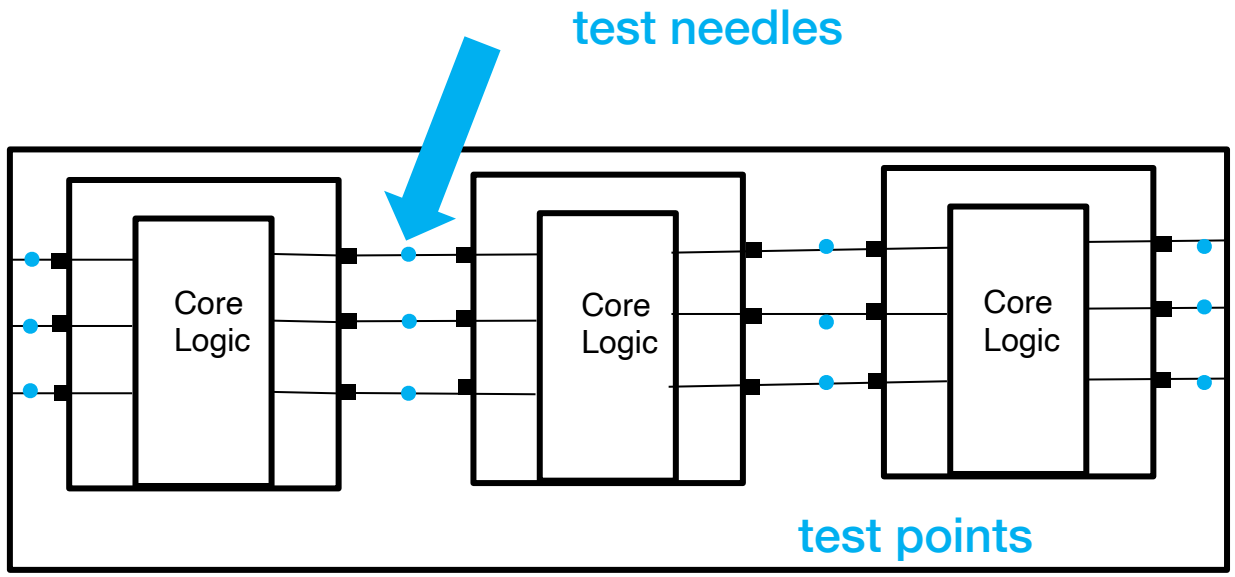
TOOLS OF THE TRADE

A selection of tools

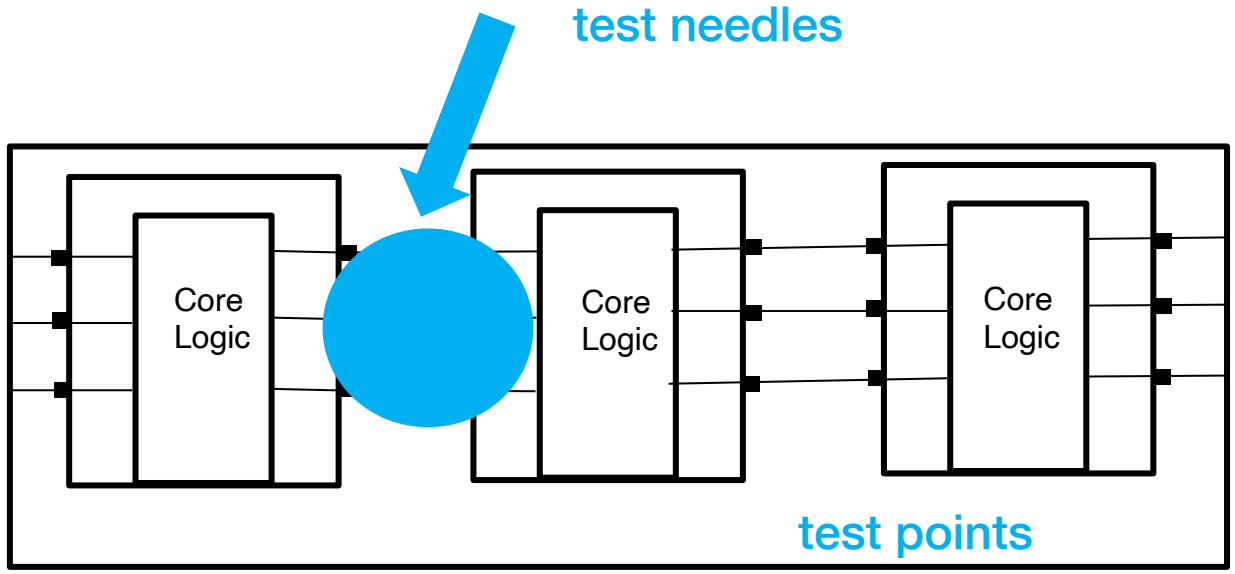


CAN WE TALK?
JTAG

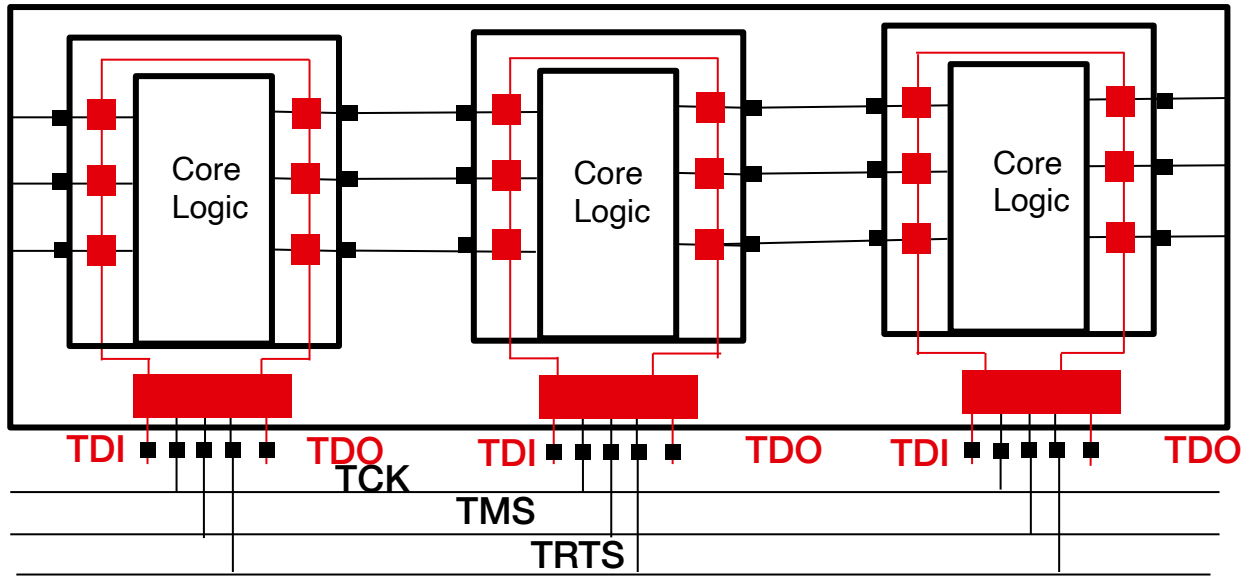
JTAG - Boundary scan



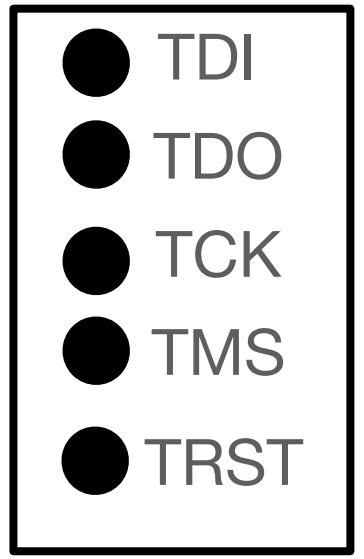
JTAG - Boundary scan



JTAG (Joint Test Action Group)

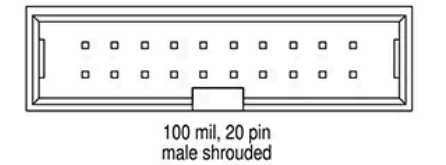


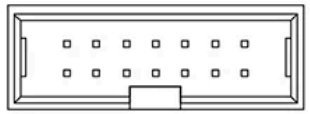
JTAG TAP

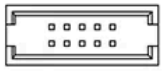


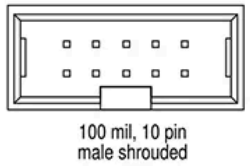
1. **TDI** (Test Data In)
2. **TDO** (Test Data Out)
3. **TCK** (Test Clock)
4. **TMS** (Test Mode Select)
5. **TRST** (Test Reset) optional.

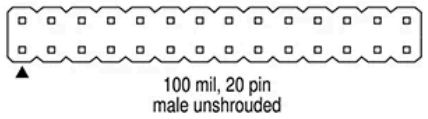
JTAG (Joint Test Action Group)

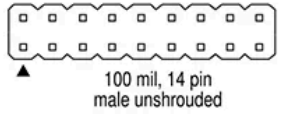


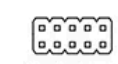


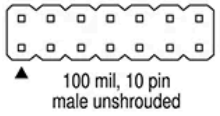


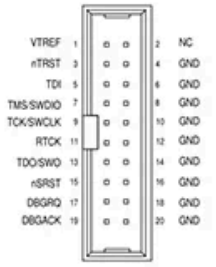


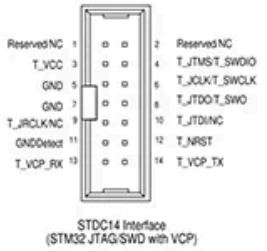


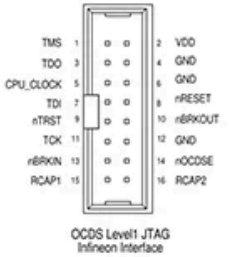


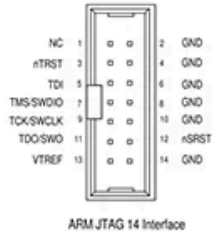


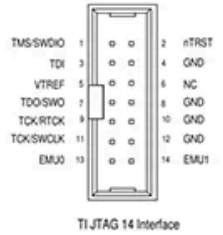


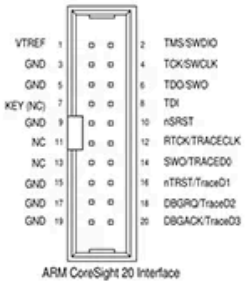


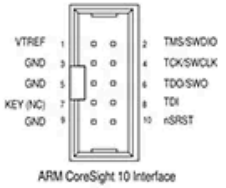












No standard connection for JTAG!

Debug interfaces

UART

Pros:
+ test/debug interface
+ well documented
+ “easy” to recognize
+ only two wires
+ no clock

Cons:
-slow

JTAG

Pros:
+ test/debug interface micron
+ circuitry is standardized
+ very powerfull

Cons:
- hard to recognize
- can be locked
- protocol implementation is not standard

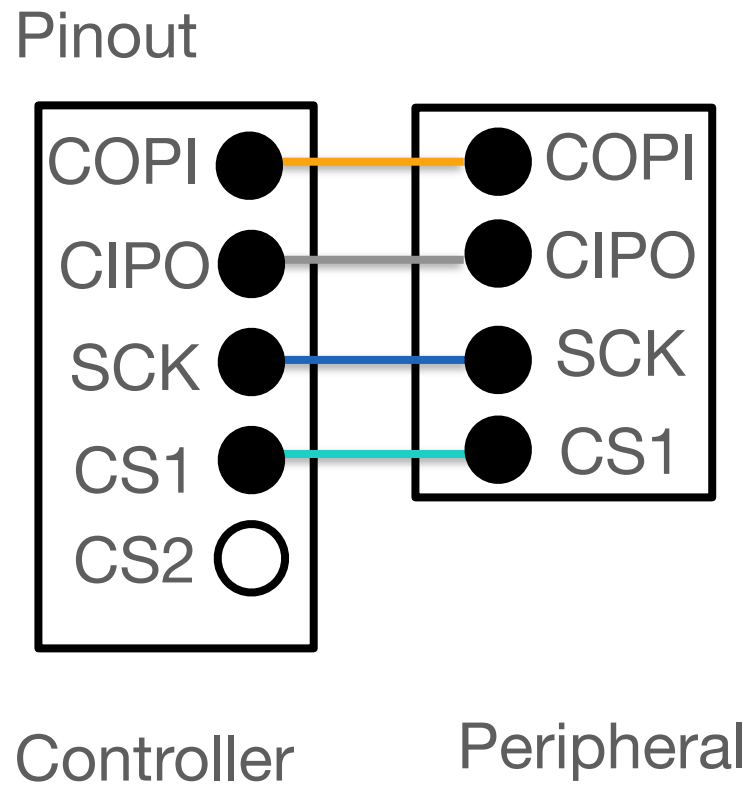
SPI and I2C
More Protocols

Serial Periferal Interface (SPI)

- [x] Serial
- [ ] Parallel
- [x] Synchronous
- [ ] Asynchronous

Serial Periferal Interface (SPI)

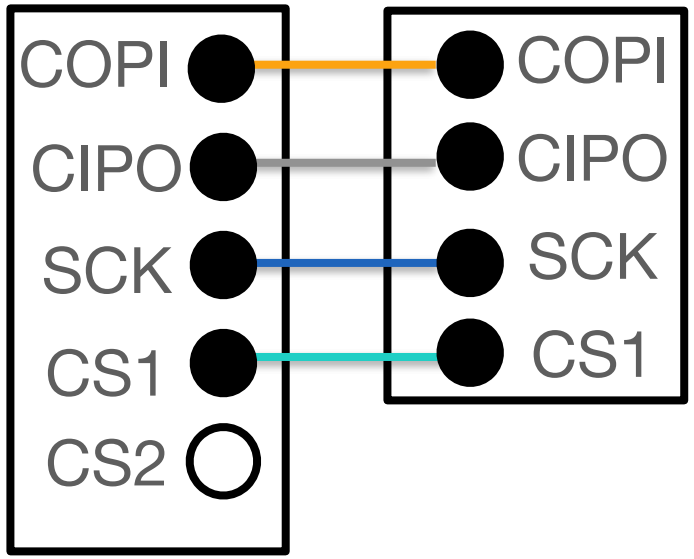
- [x] Serial
- [ ] Parallel
- [x] Synchronous
- [ ] Asynchronous



Serial Periferal Interface (SPI)

- [x] Serial
- [ ] Parallel
- [x] Synchronous
- [ ] Asynchronous

Pinout



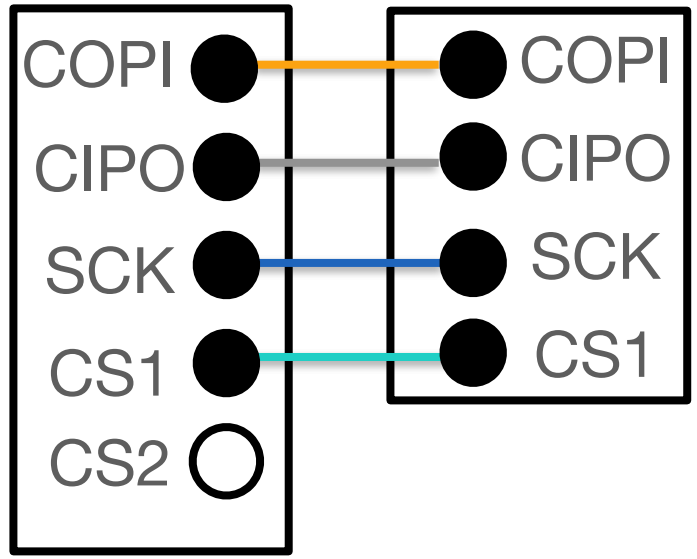
Pro's

- faster than asynchronous protocols
- can support multiple peripherals
- very common

Serial Periferal Interface (SPI)

- [x] Serial
- [ ] Parallel
- [x] Synchronous
- [ ] Asynchronous

Pinout



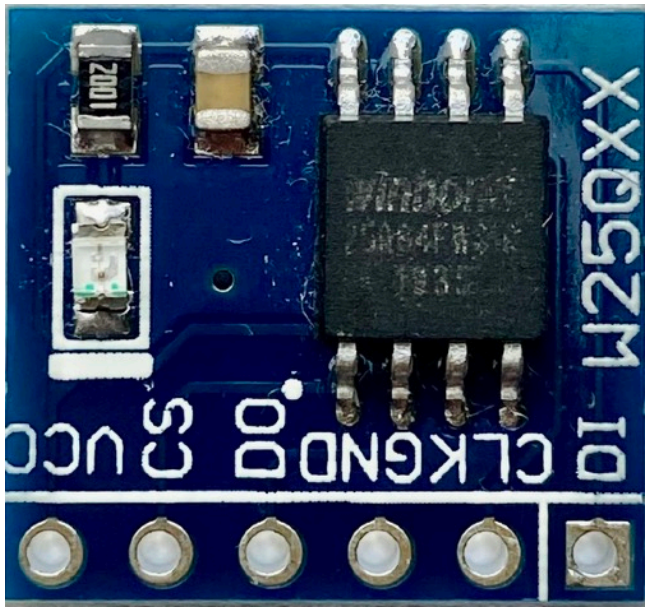
Pro's

- faster than asynchronous protocols
- can support multiple peripherals
- very common

Con's

- more wires than other protocols
- separate line for each peripheral

SPI in the wild



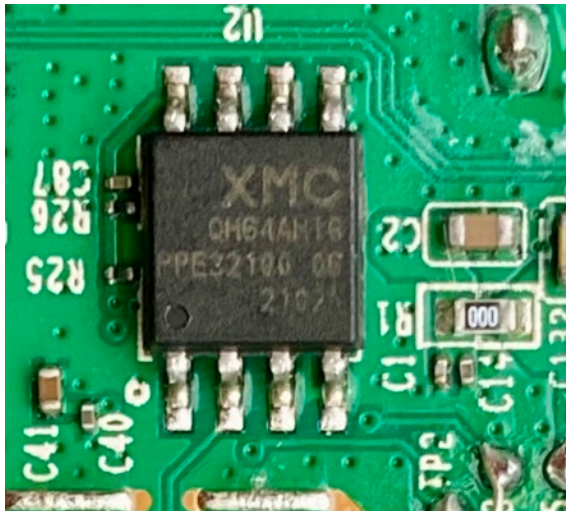
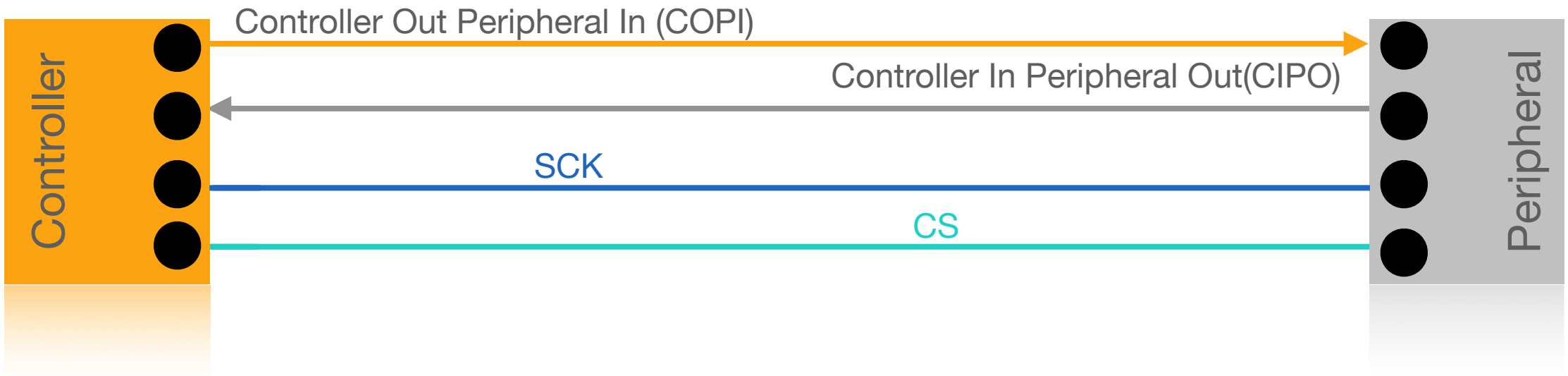


Image source: author collection

SPI communication

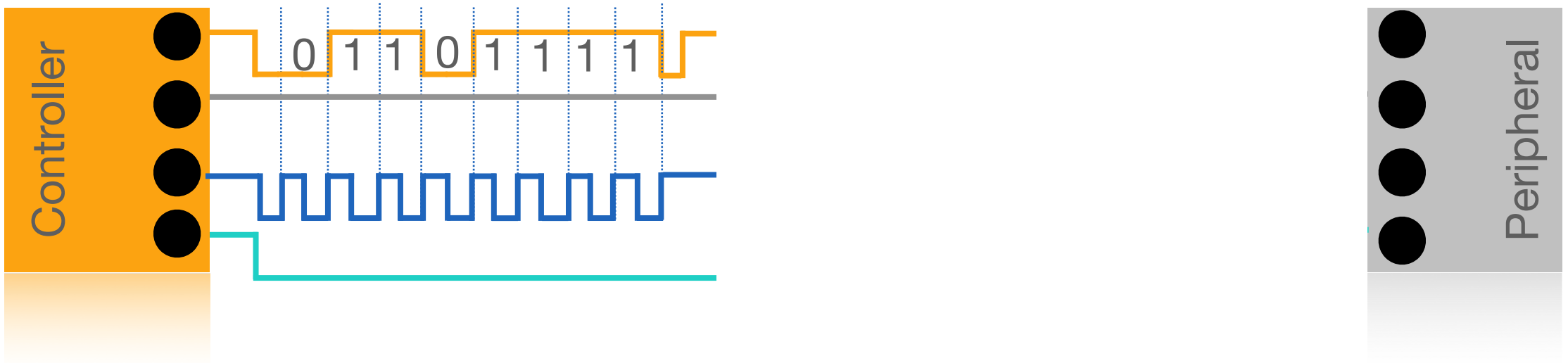


COPI=MOSI=DO
CIPO=MISO=DI

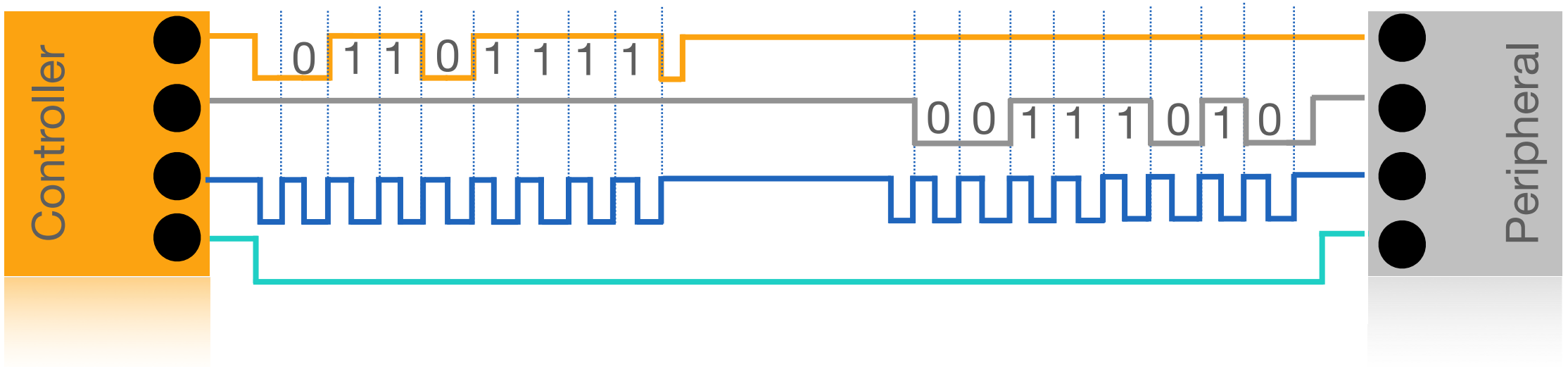
Controller/Master
Peripheral/Slave

SPI communication

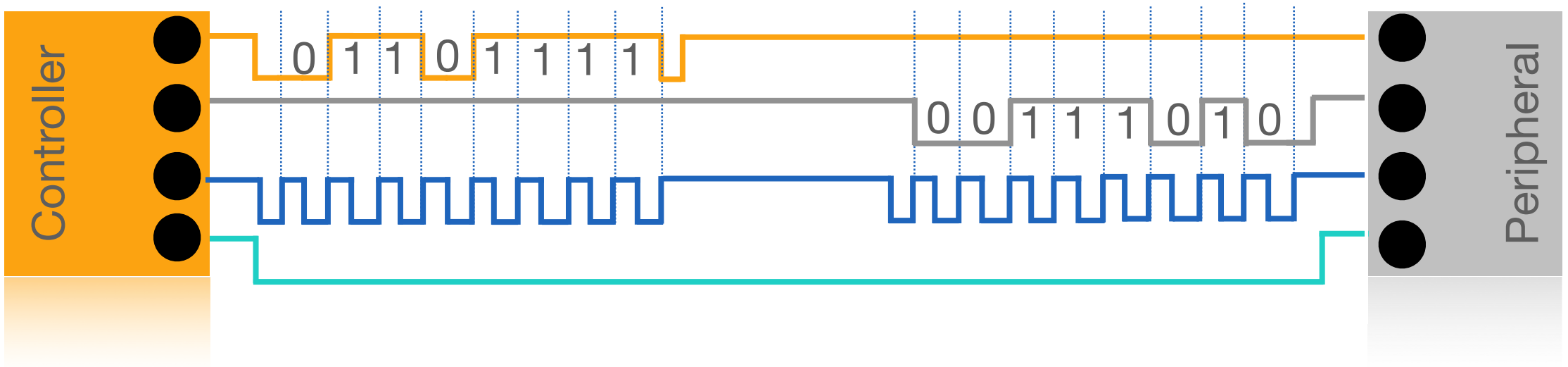
SPI communication



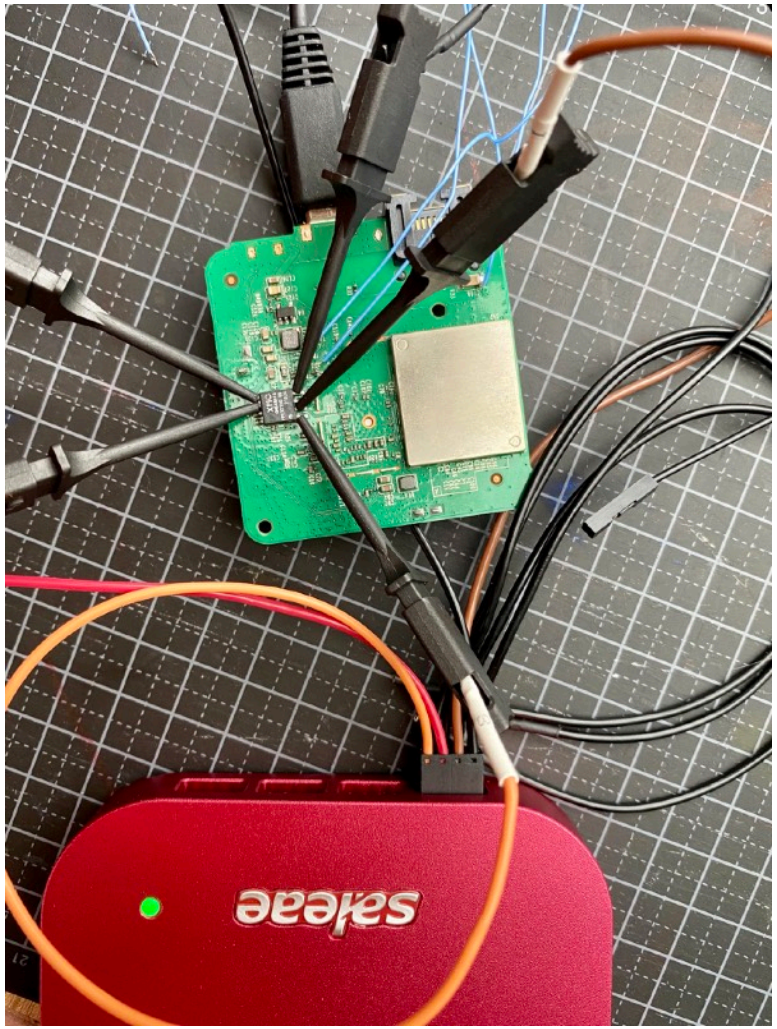
SPI communication



SPI communication



SPI communication (logic analyzer)



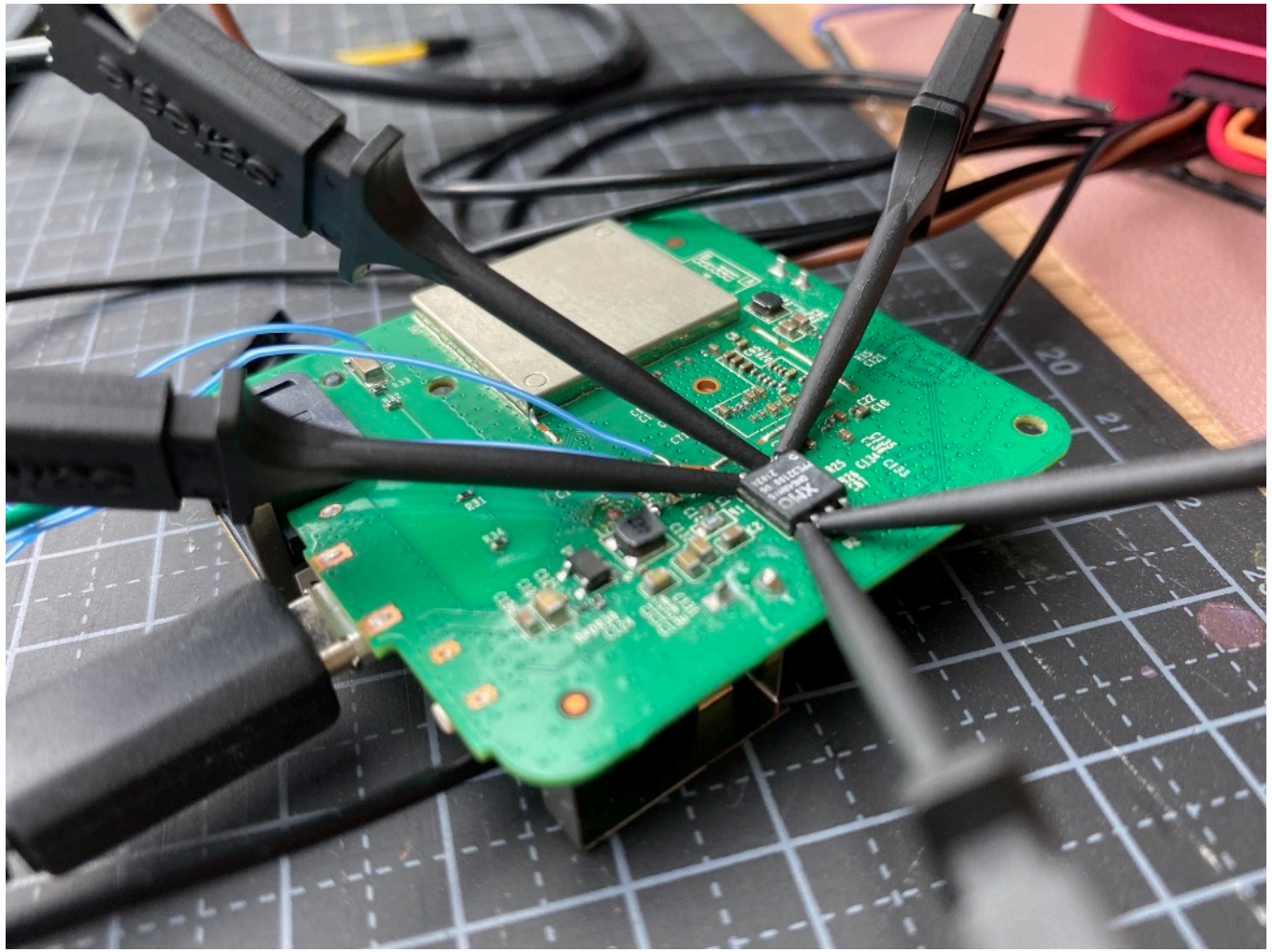


Image source: author collection

SPI communication (logic analyzer)

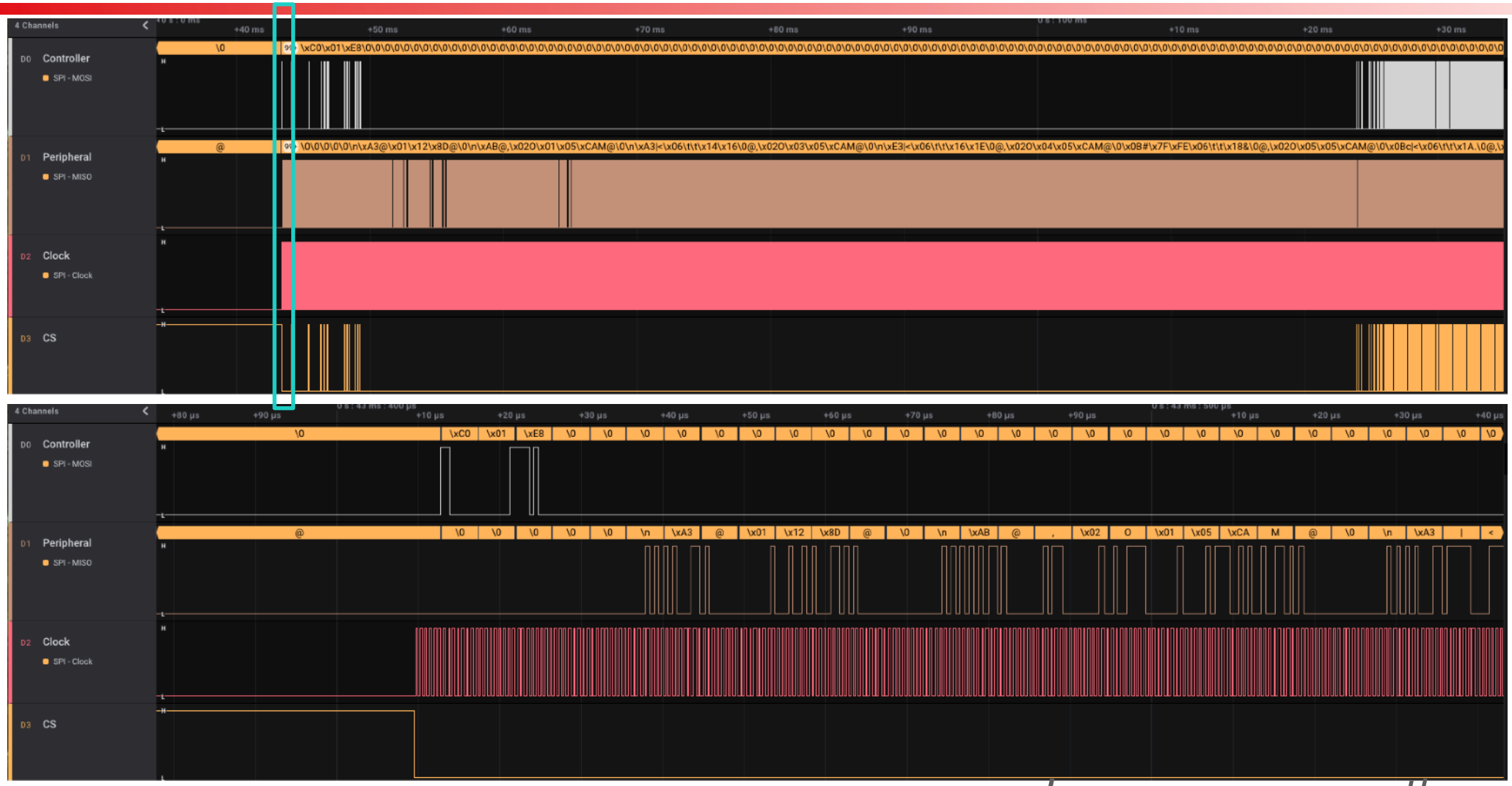
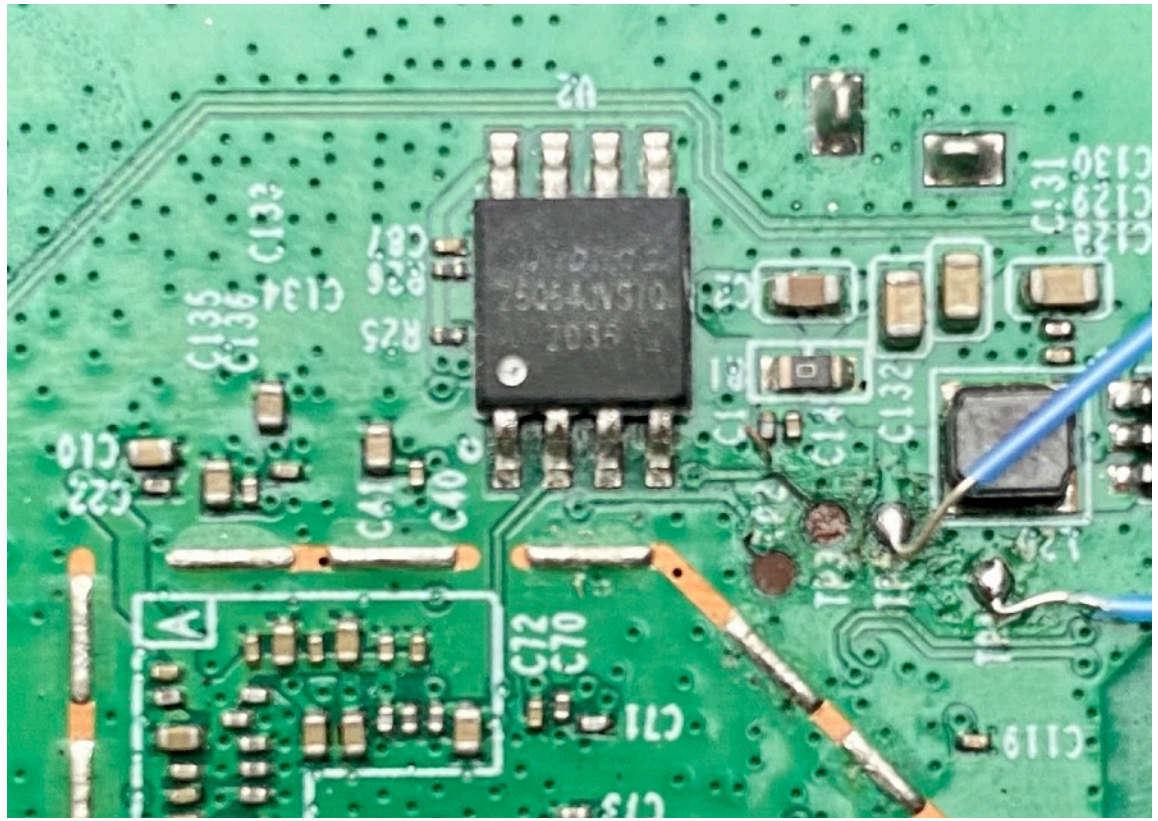


Image source: author collection

SPI Demo (firmware extraction)



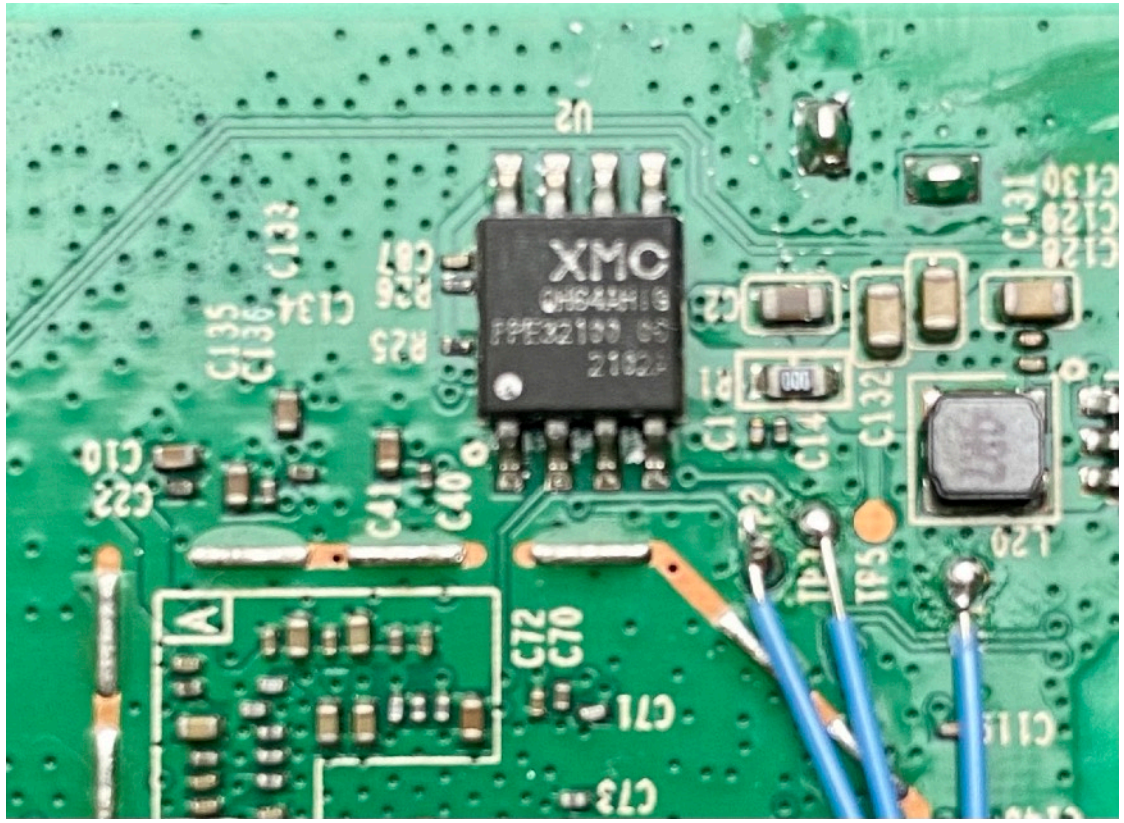


Image source: author collection

SPI Demo (firmware extraction)

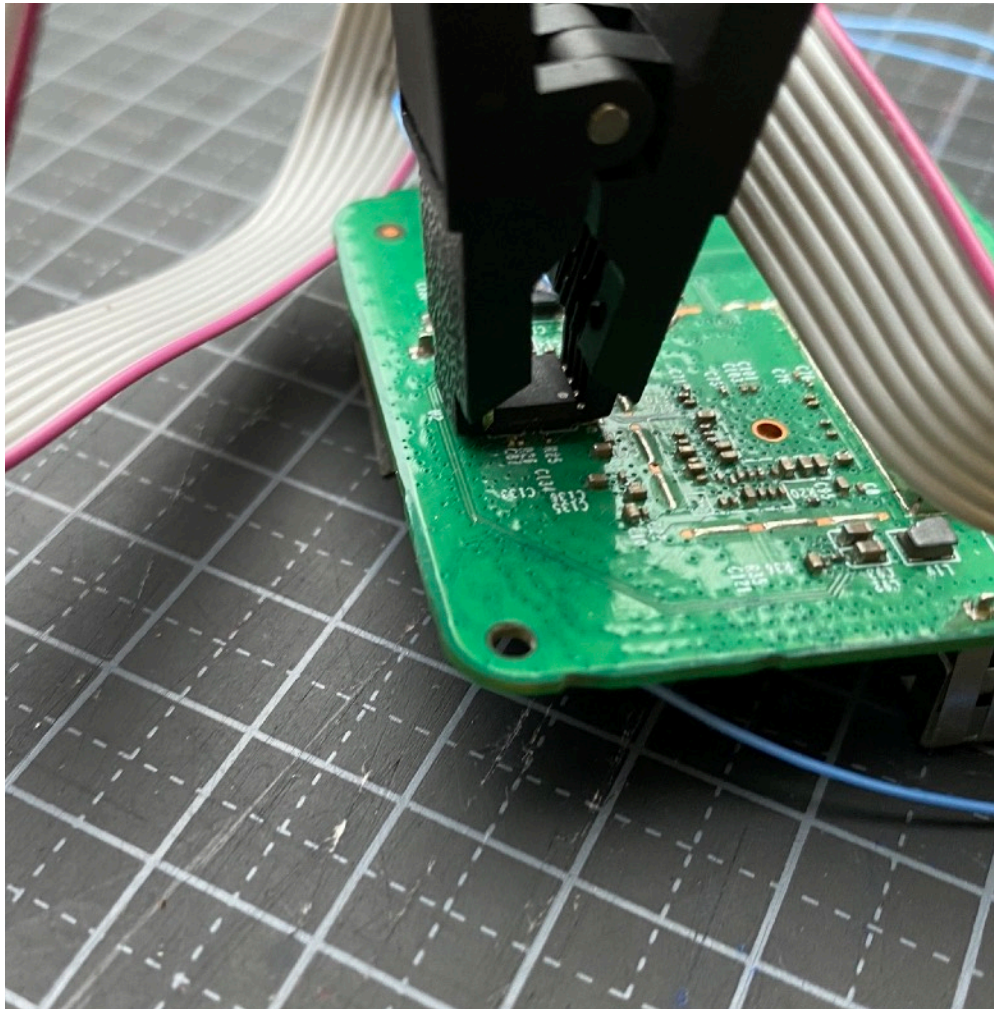
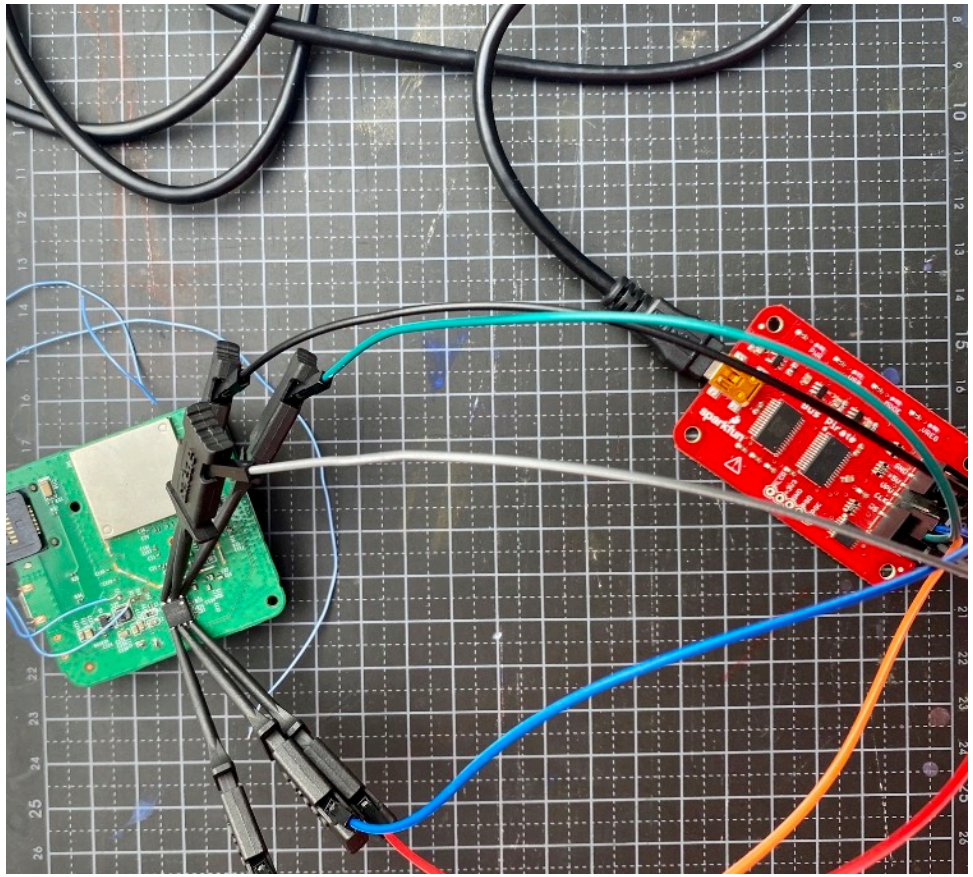


Image source: author collection

SPI Demo (firmware extraction)



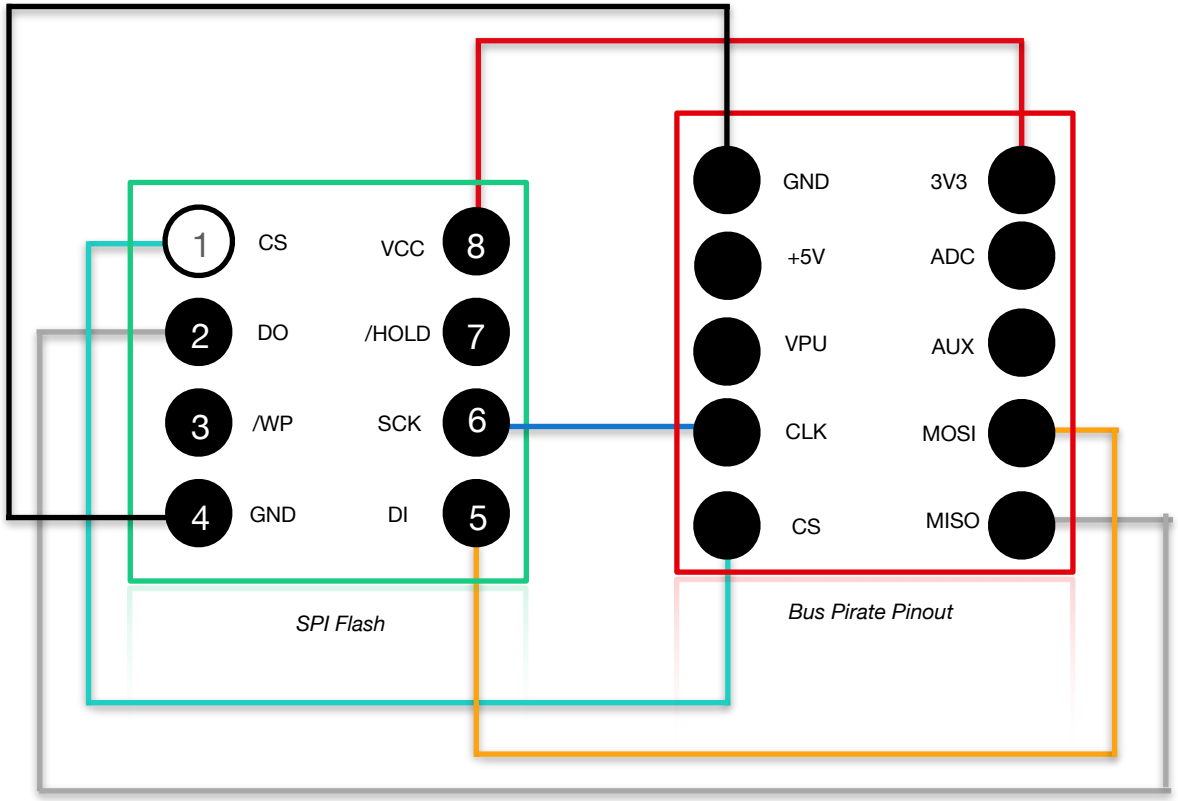
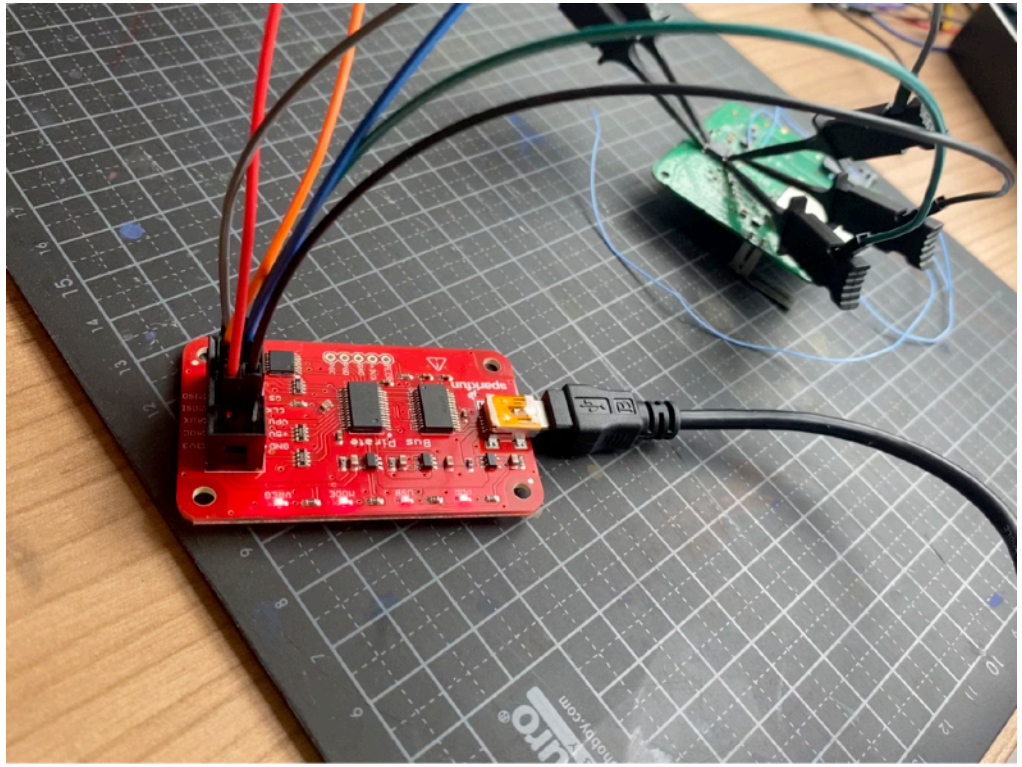


Image source: author collection

SPI Demo (firmware extraction)



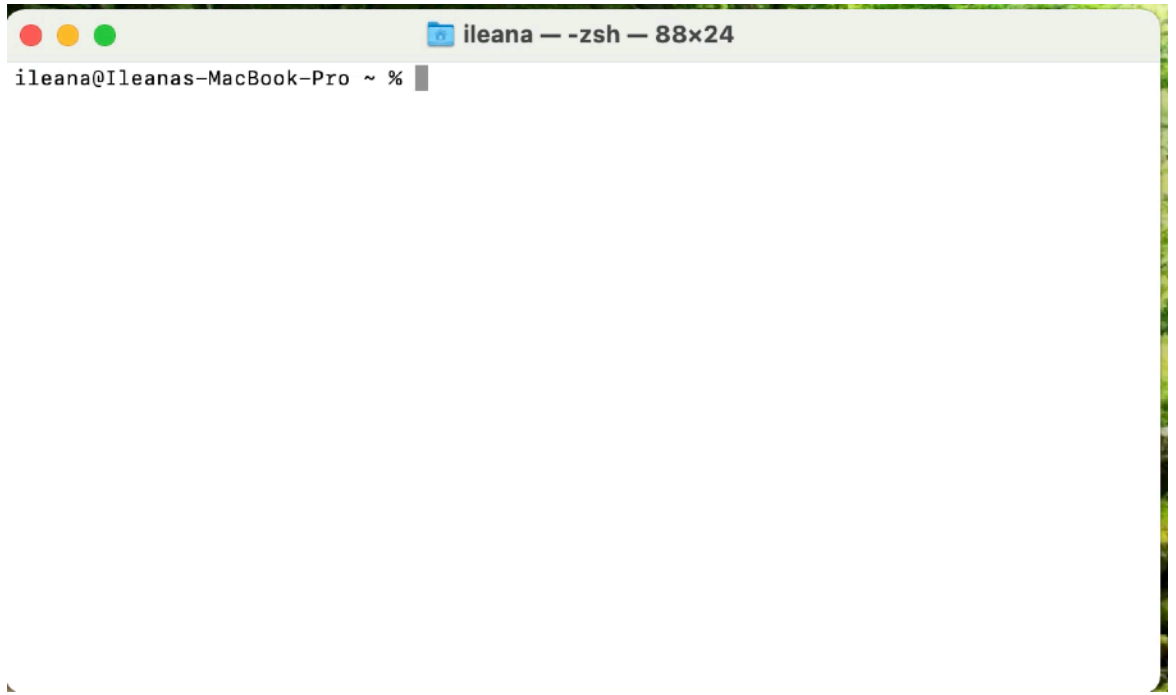
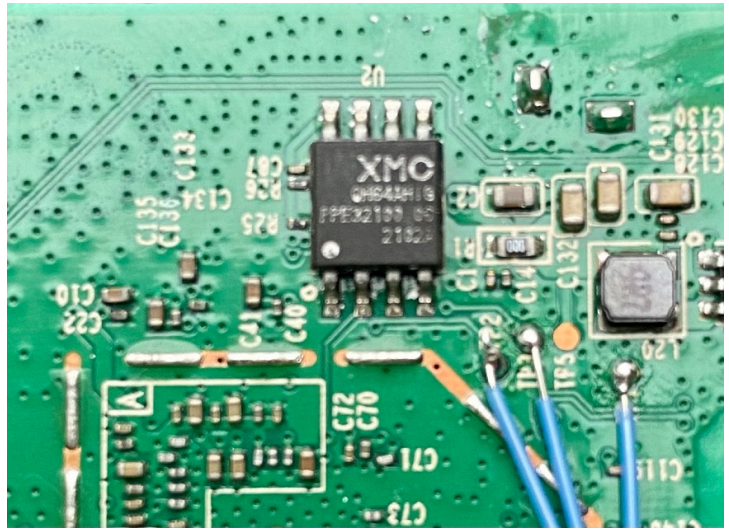


Image source: author collection

SPI Demo (firmware extraction) - troubleshooting

Flashroom:

- downgrade flashrom
- XMC not supported by default
- One defective clip



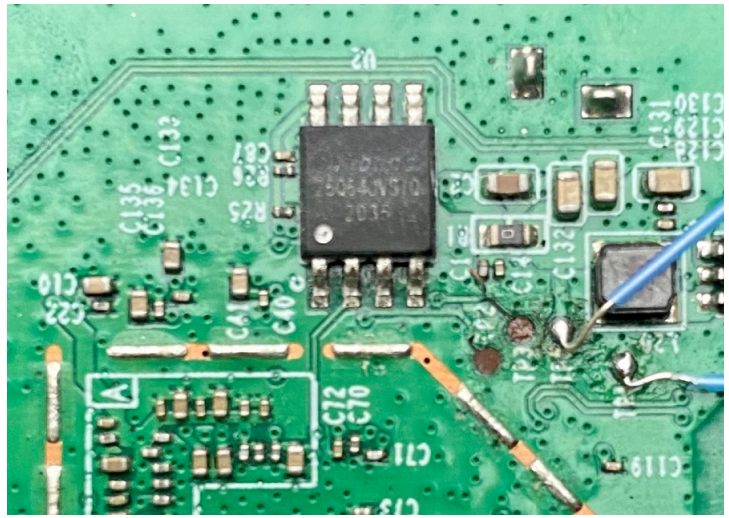
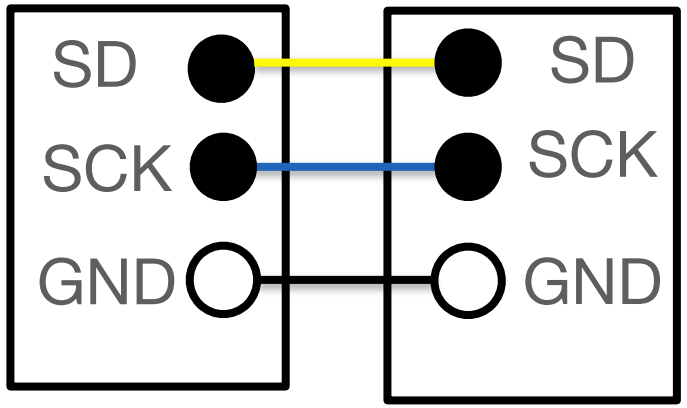


Image source: author collection

Inter-Integrated Circuit (I2C)

- [x] Serial
- [ ] Parallel
- [x] Synchronous
- [ ] Asynchronous

Pinout



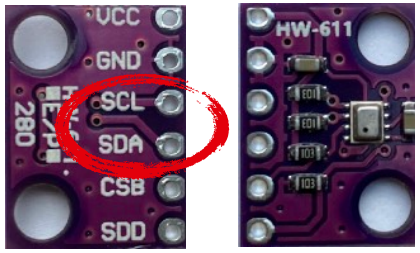
Pro's

- very simple
- support multiple peripherals
- very common (sensors)

Con's

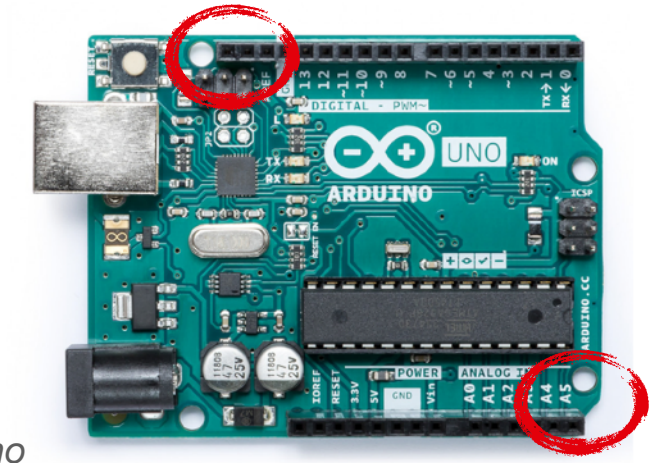
- Lower speed (SPI)
- separate line for each peripheral

I2C in the wild

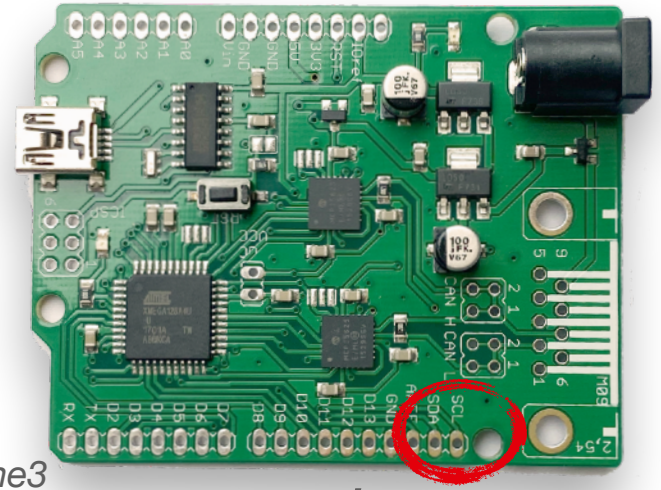


(bottom) *(top)*

Barometer pressure sensor



Arduino Uno



Riscurino RHme3

Image source: author collection

Inter-Integrated Circuit (I2C)

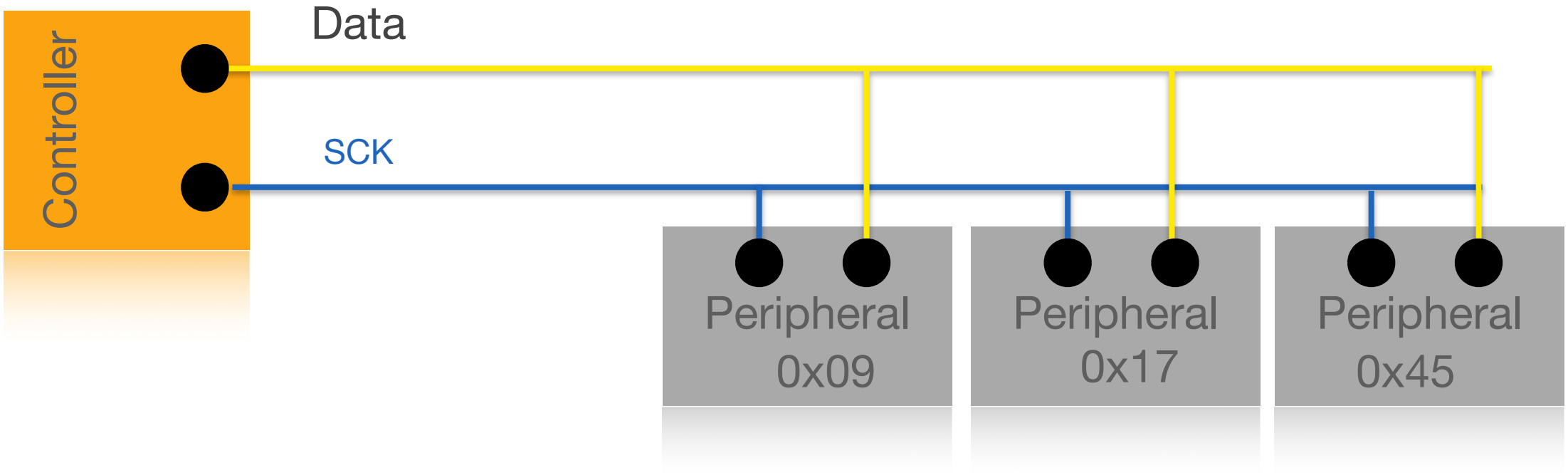


Image source: author collection

Inter-Integrated Circuit (I2C)

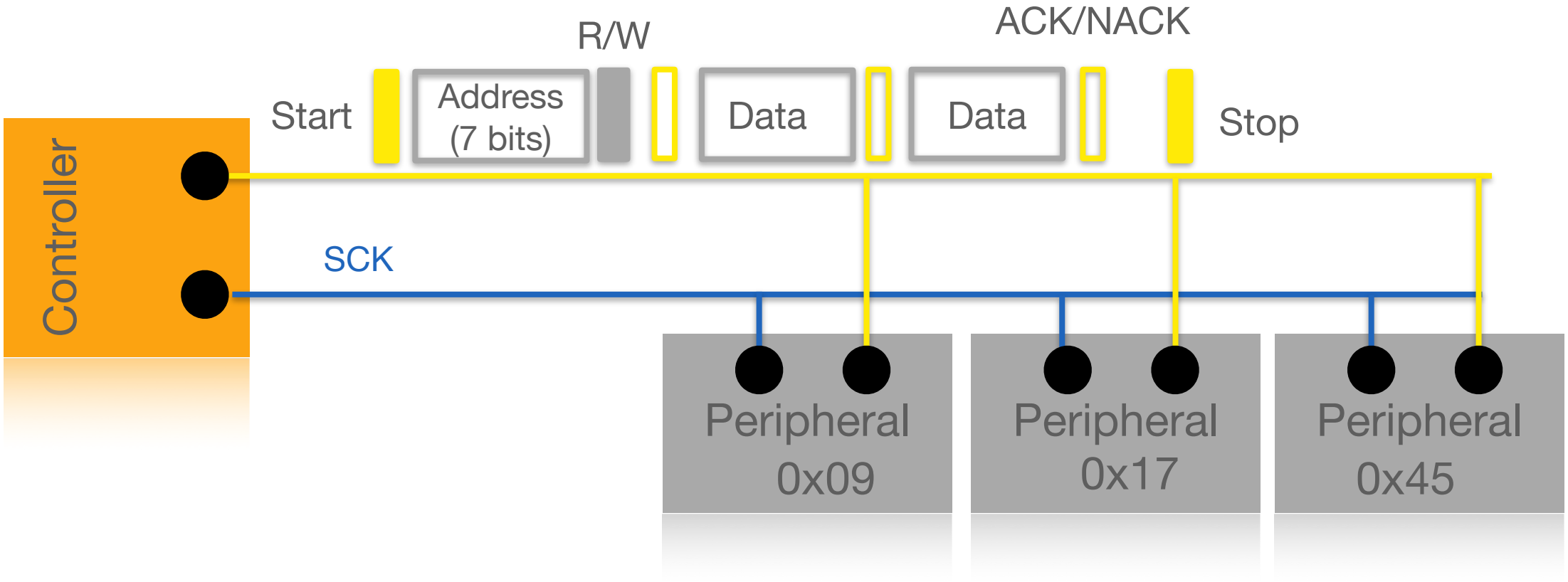


Image source: author collection

I2C Demo implementation

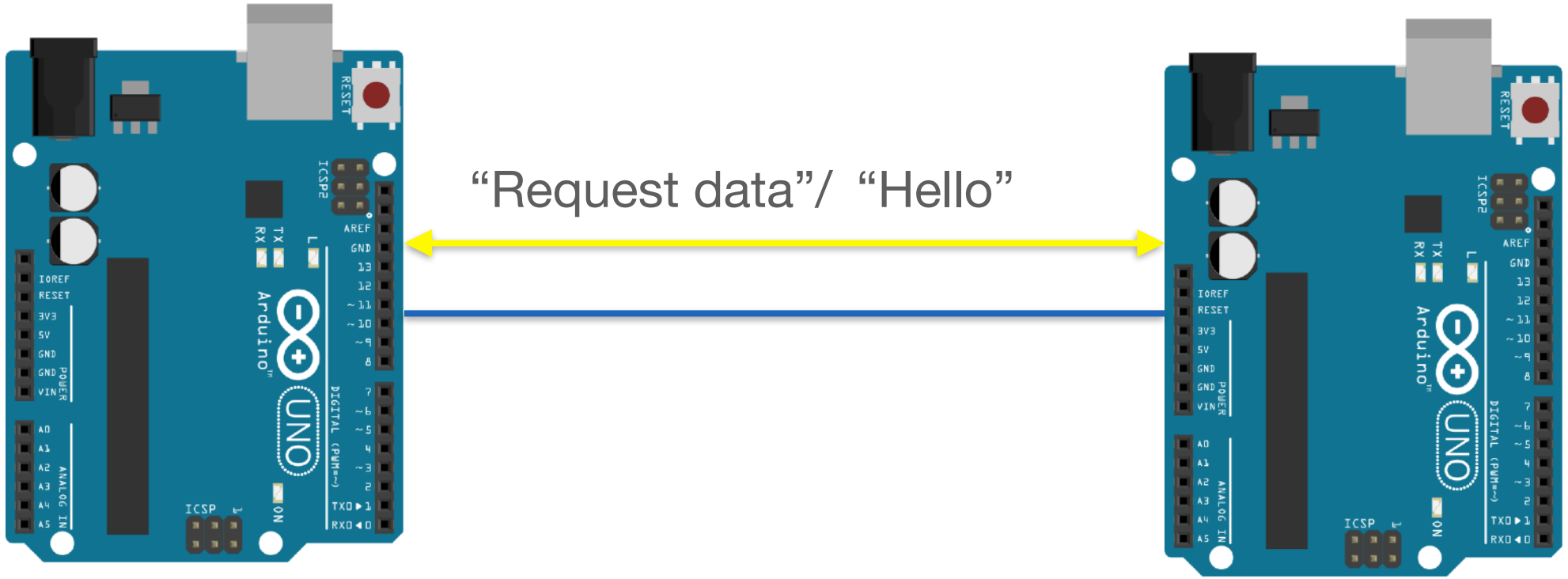
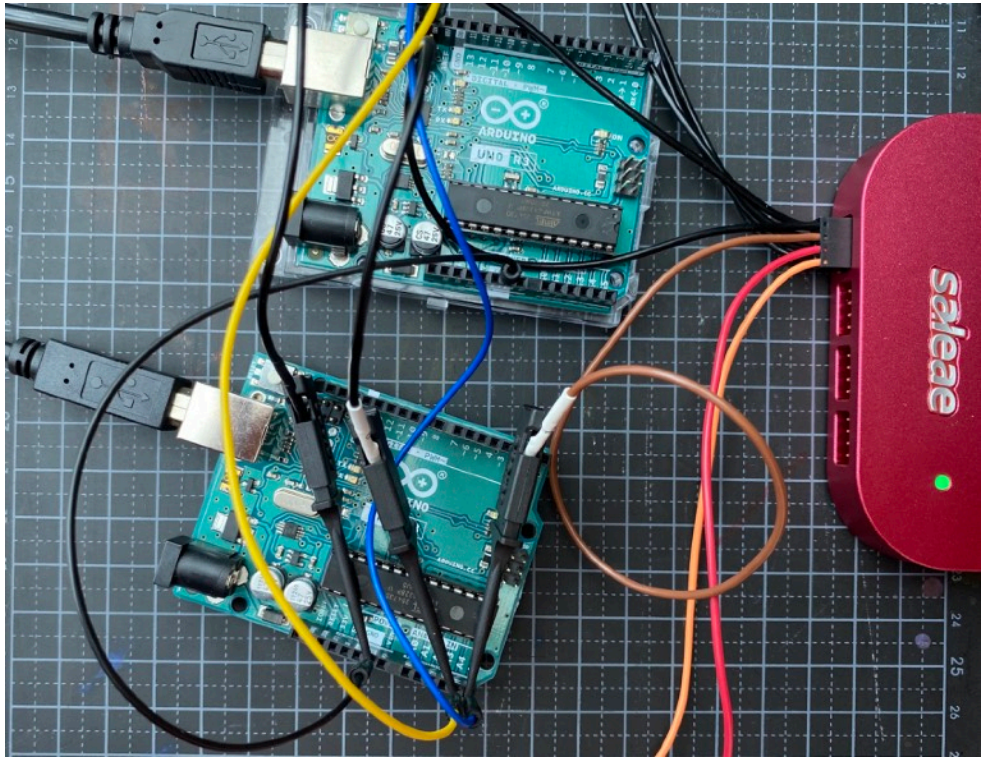


Image source: author collection

I2C communication



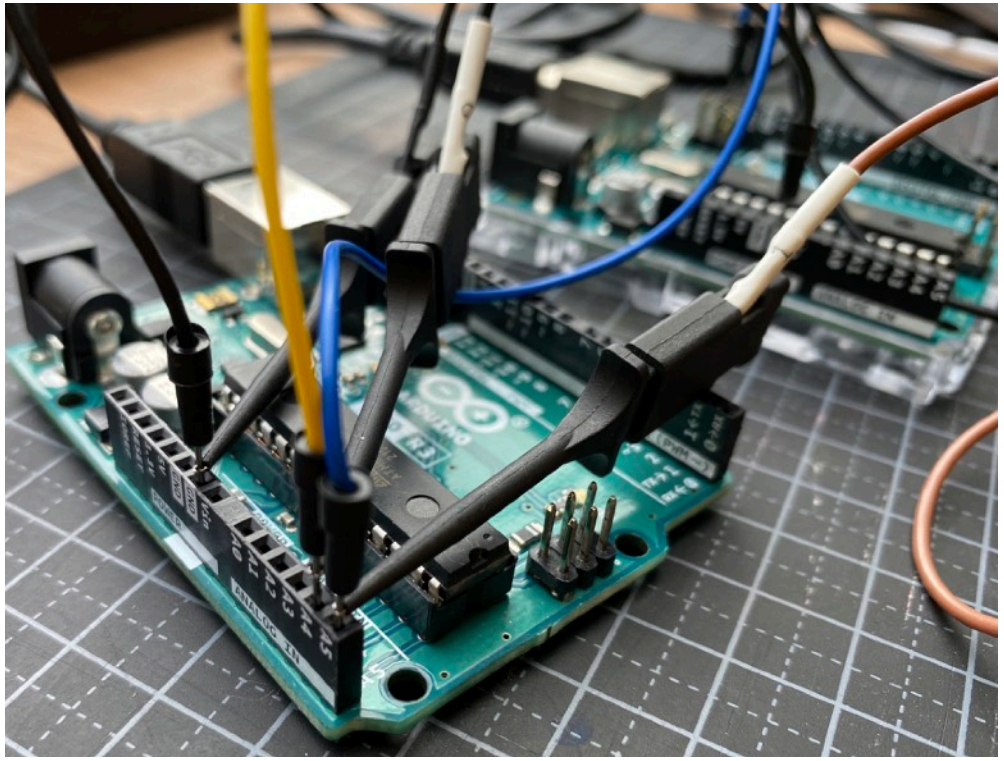


Image source: author collection

I2C communication

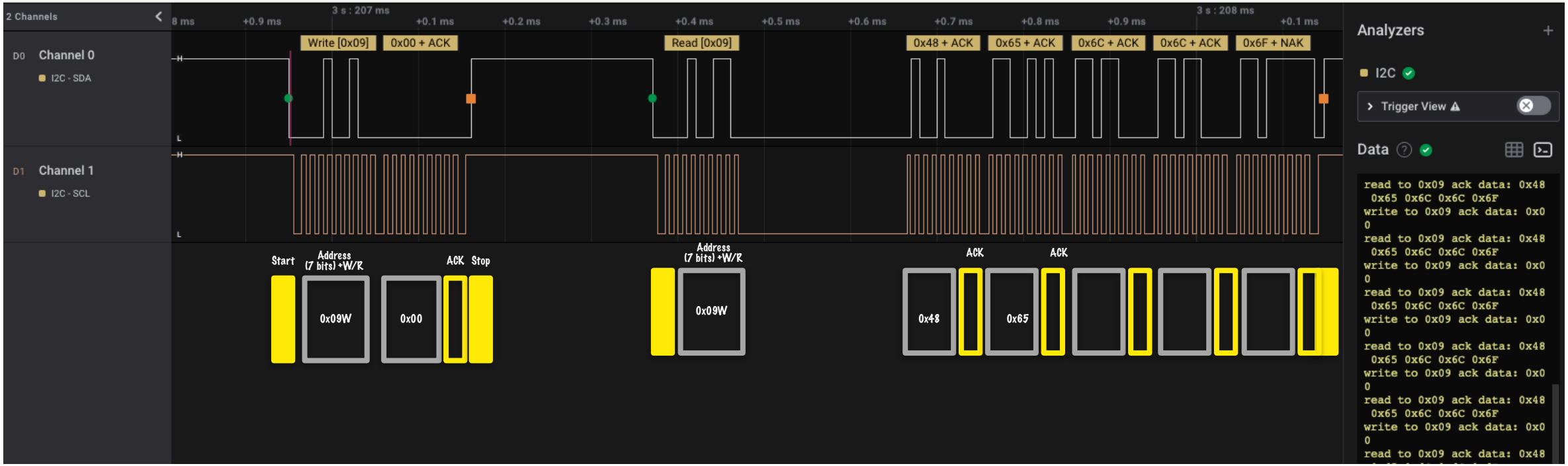
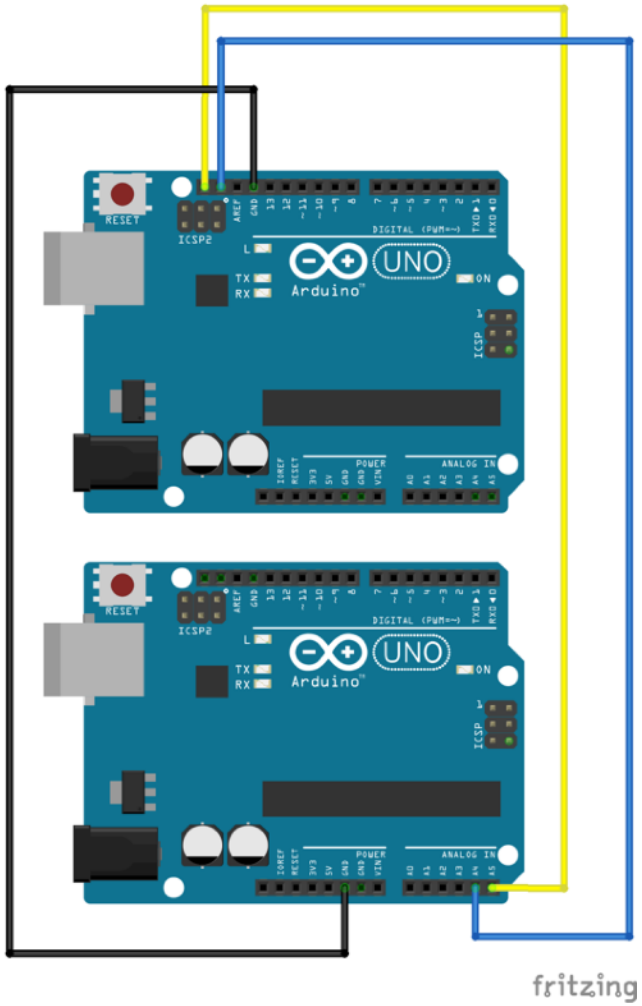


Image source: author collection

I2C communication



GND

CLK

Data

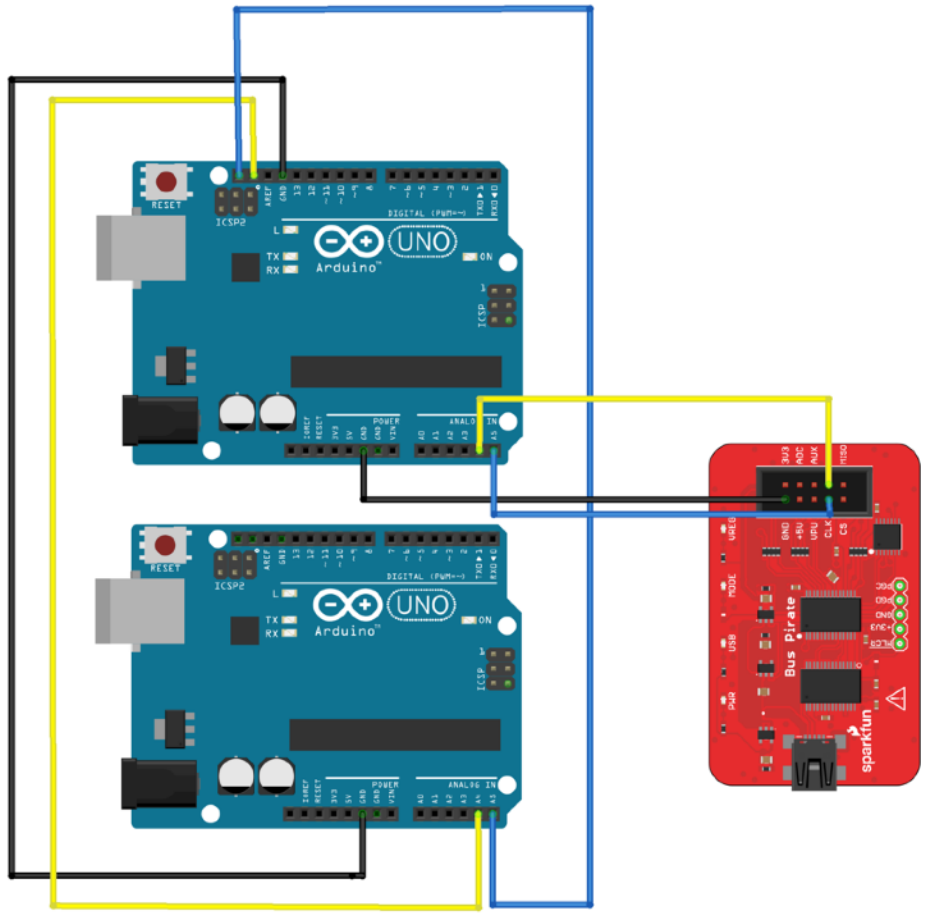
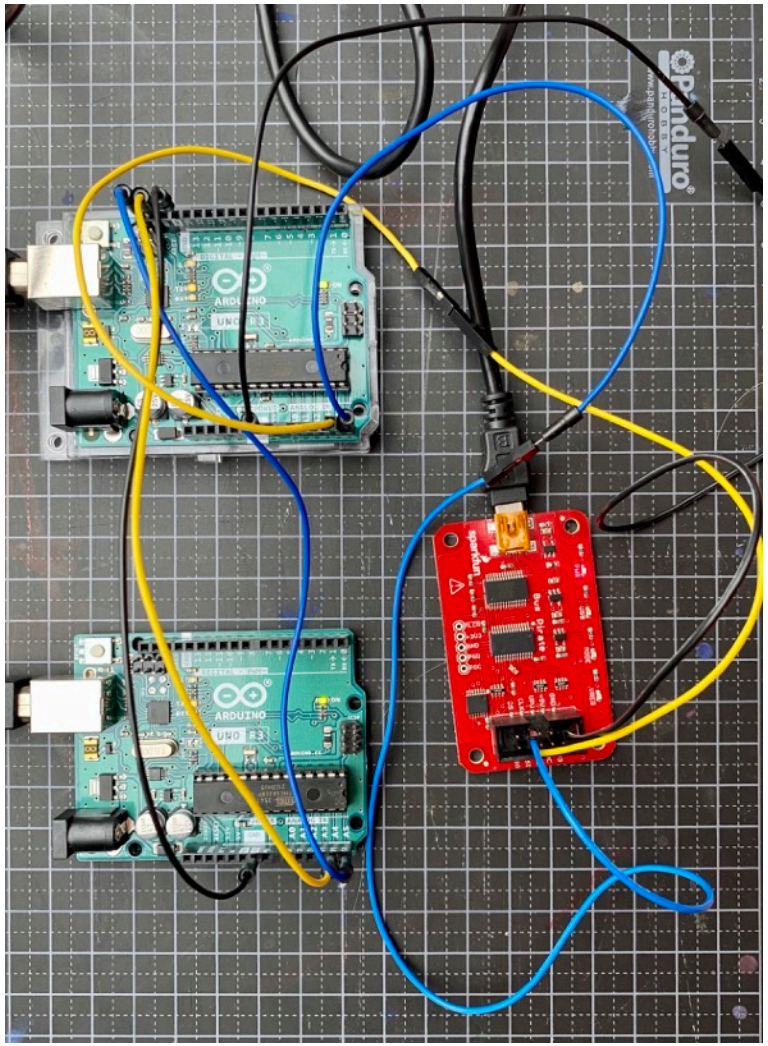


Image source: author collection

I2C sniffing Bus Pirate



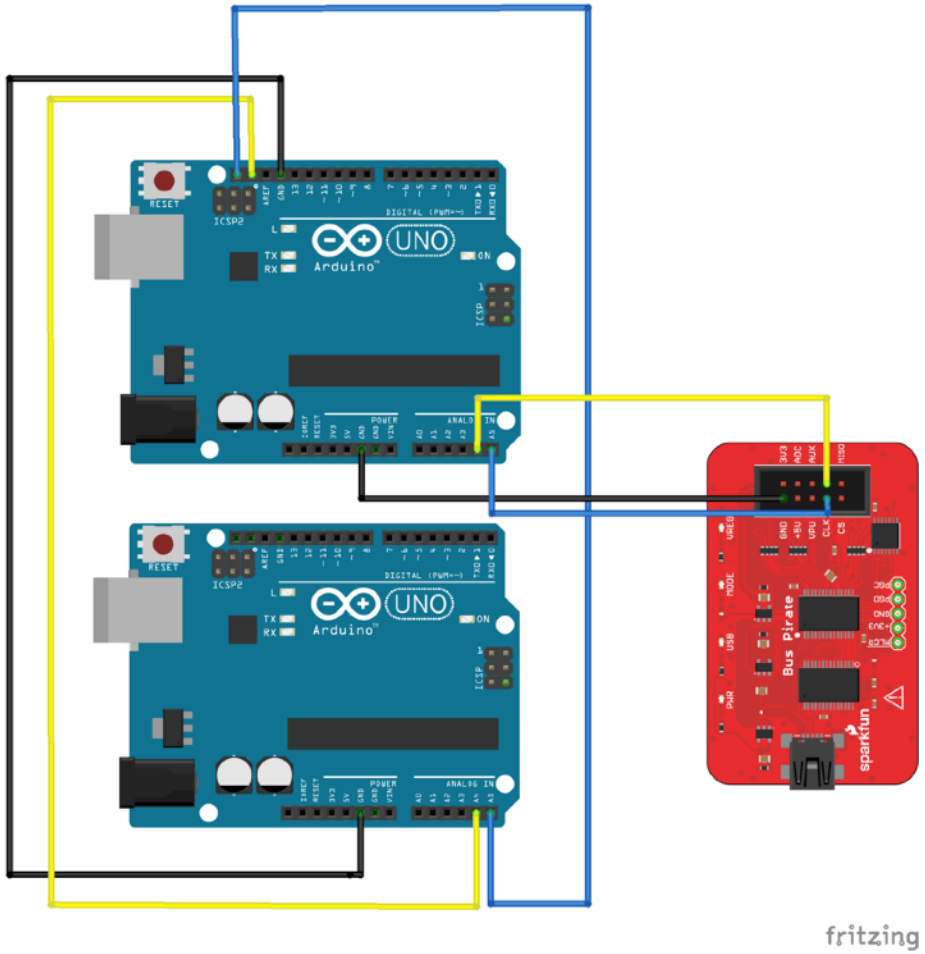
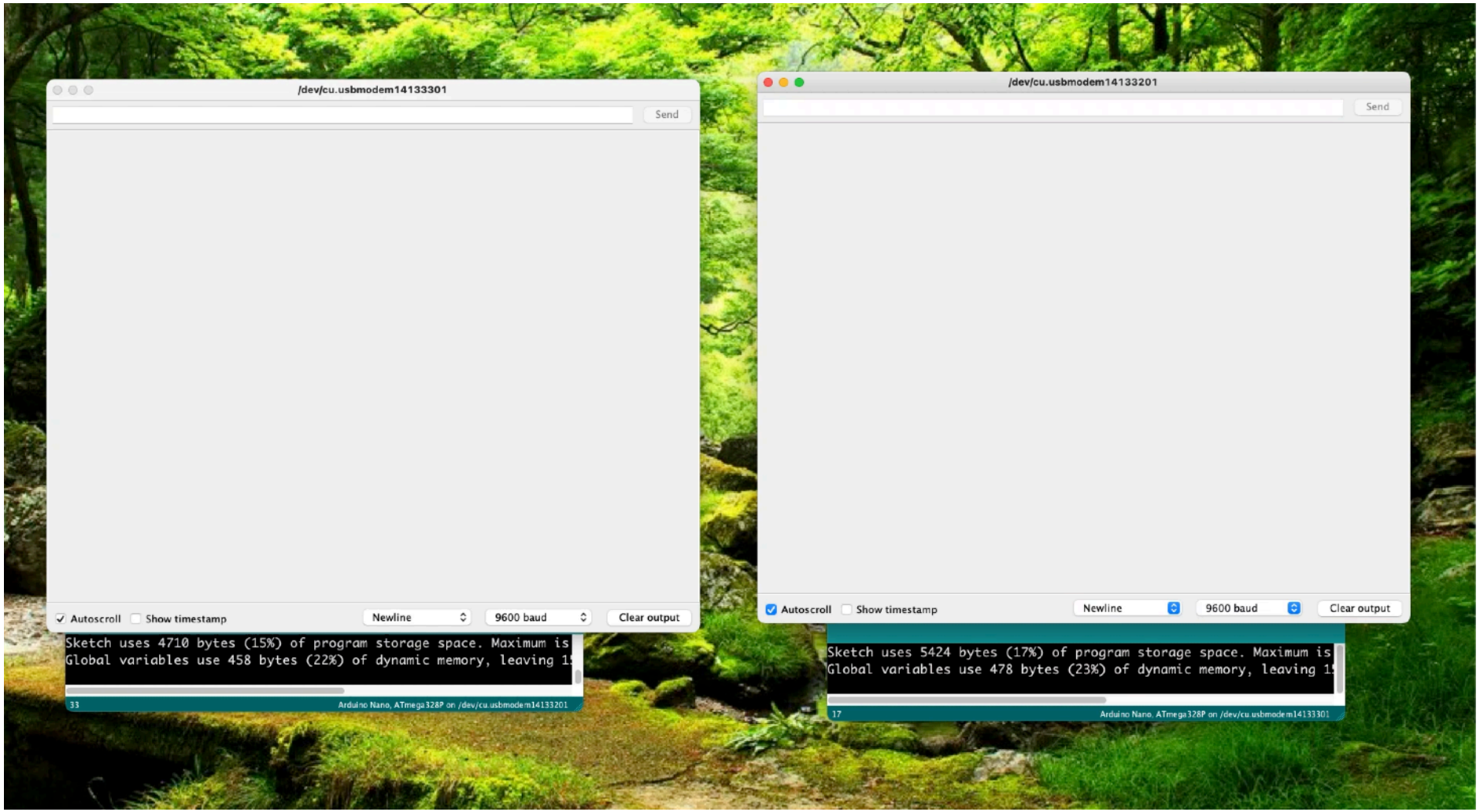


Image source: author collection

I2C communication

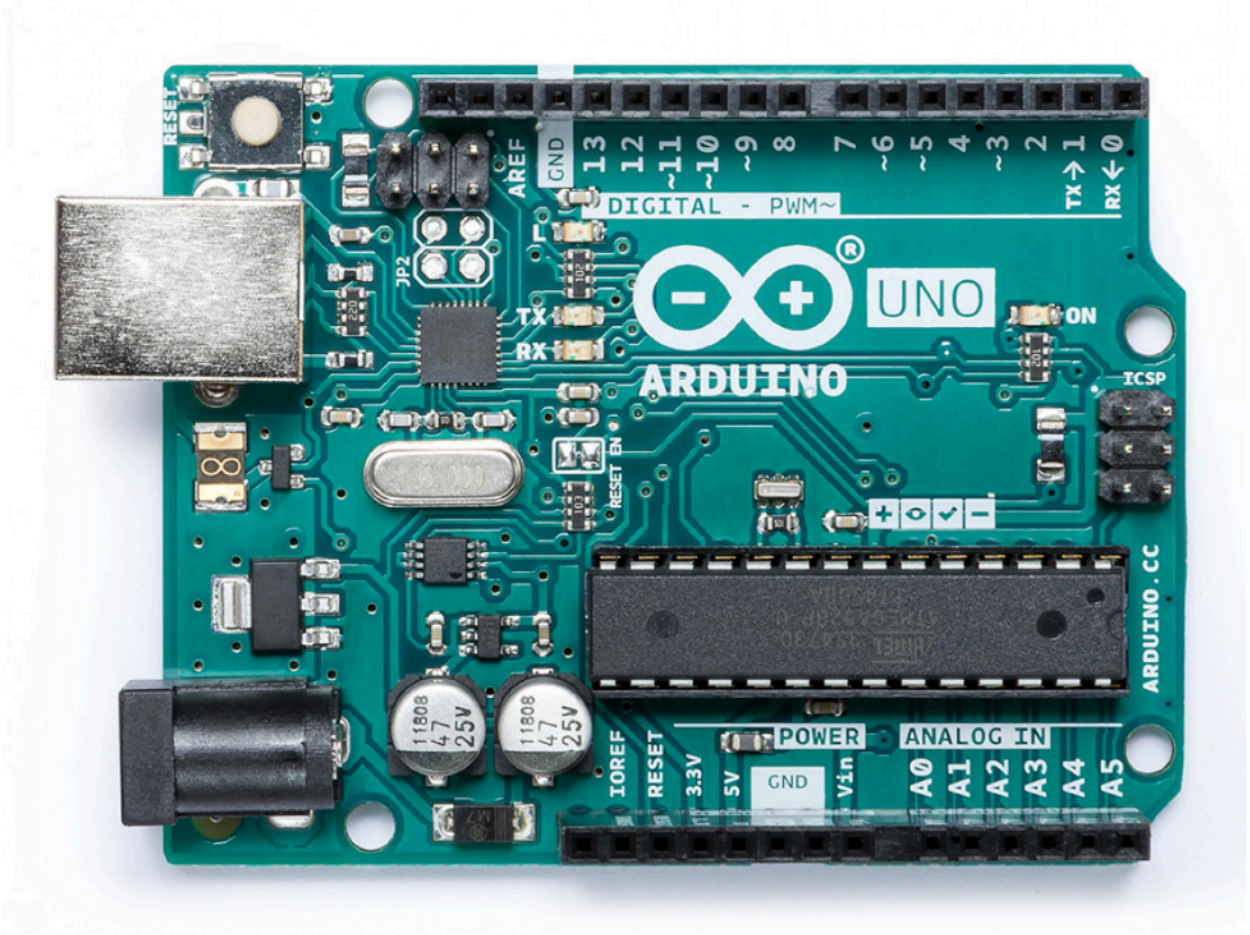


I2C

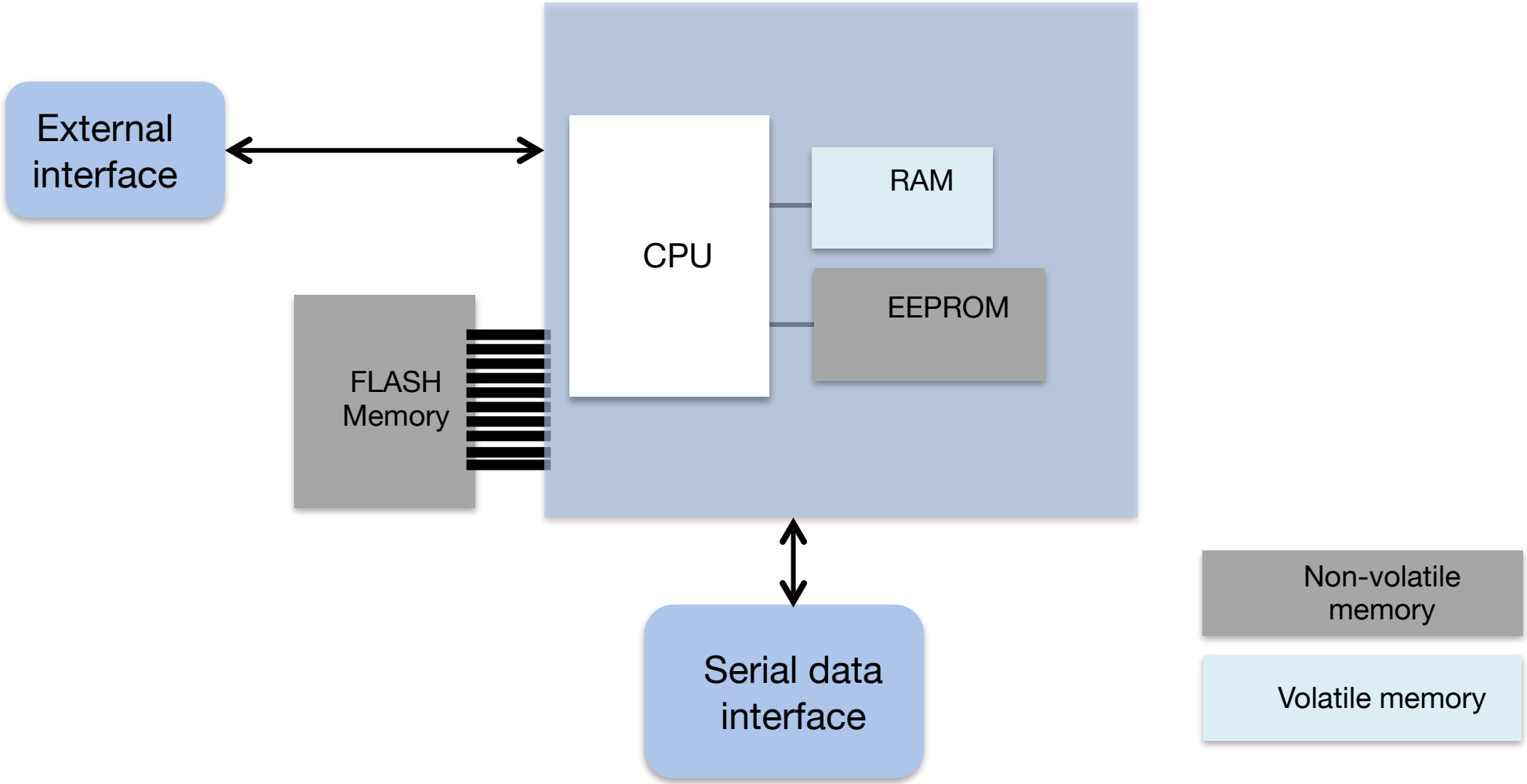
TAKE AWAY ?

SURVIVAL IN HOSTILE ENVIRONMENT

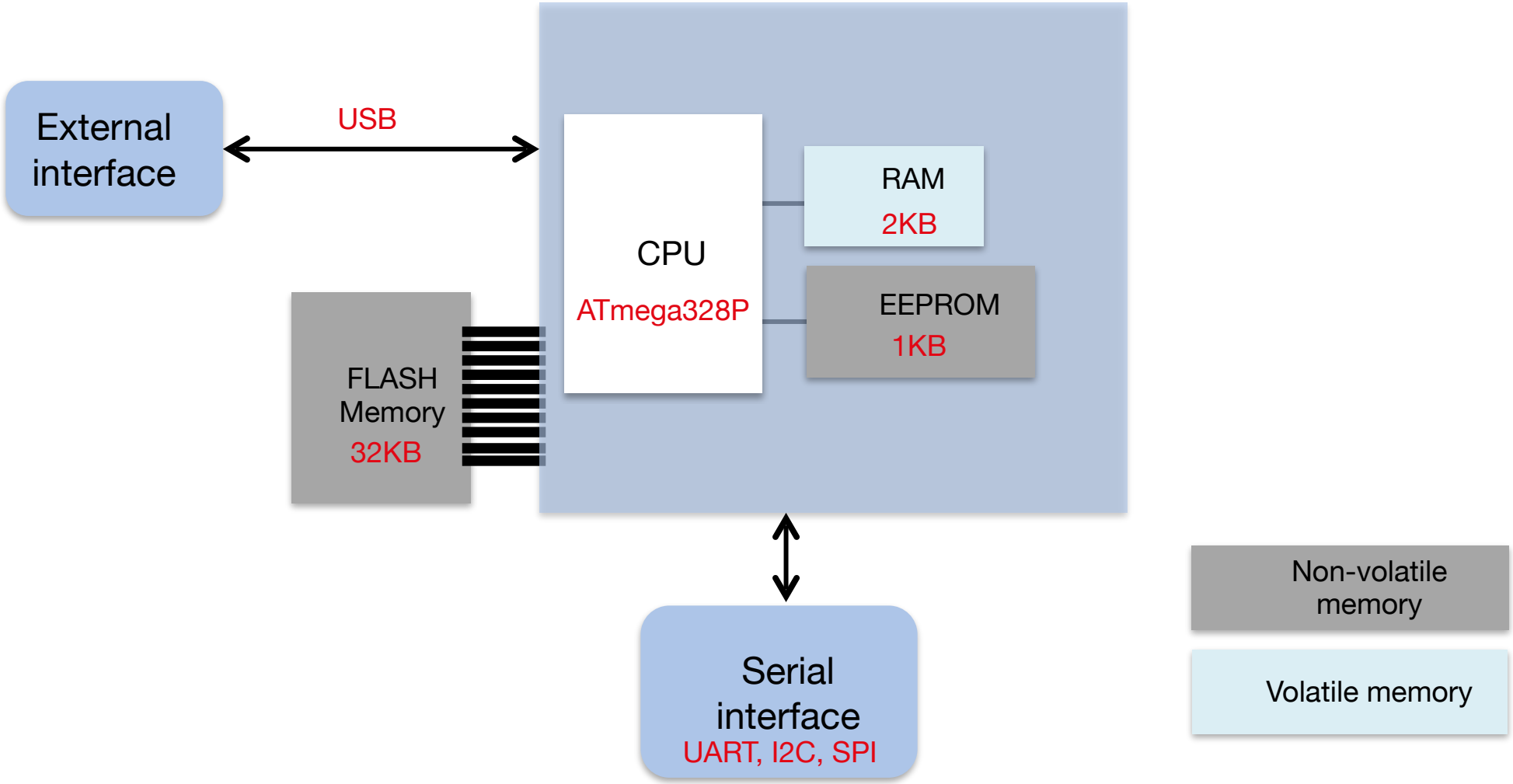
Devices in hostile environment



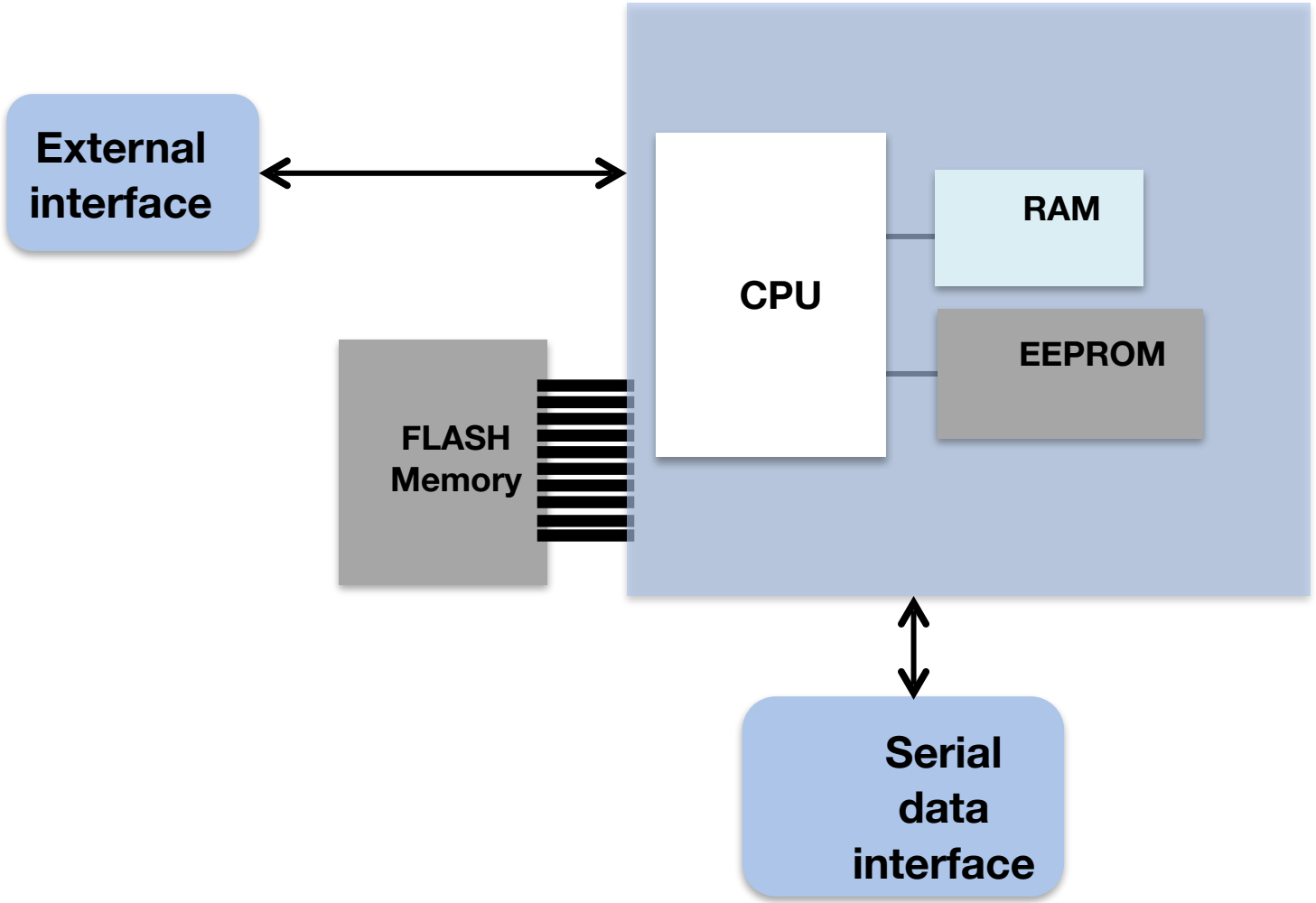
Anatomy of a COTS device



Anatomy of a COTS device



AES encryption on COTS

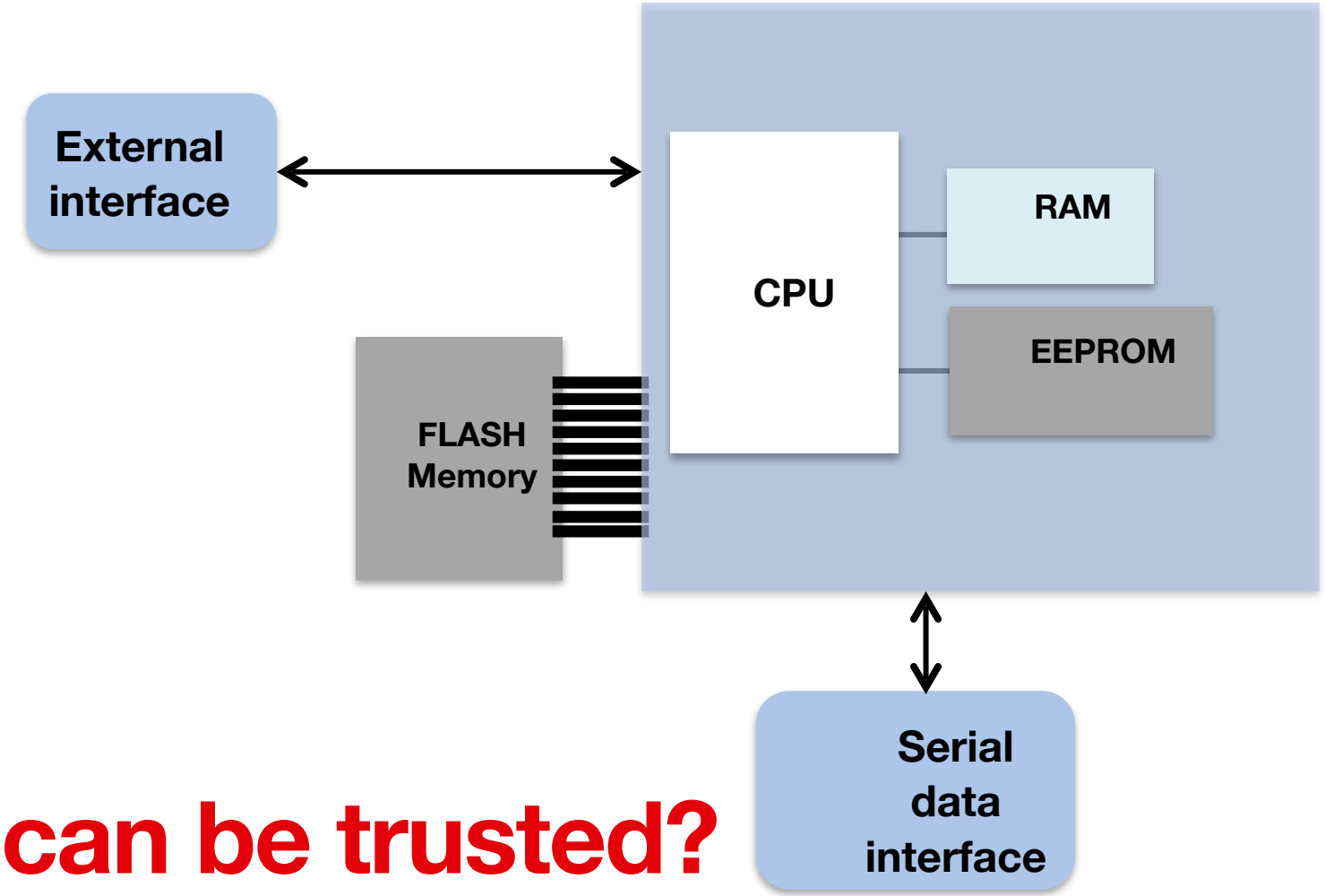



```
AddRoundKey(state, &w[0])
for i = 1 step 1 to 9
    SubBytes(state)
    ShiftRows(state)
    MixColumns(state)
    AddRoundKey(state, &w[i*4])
end for
SubBytes(state)
ShiftRows(state)
```

Non-volatile memory

Volatile memory

AES encryption on COTS



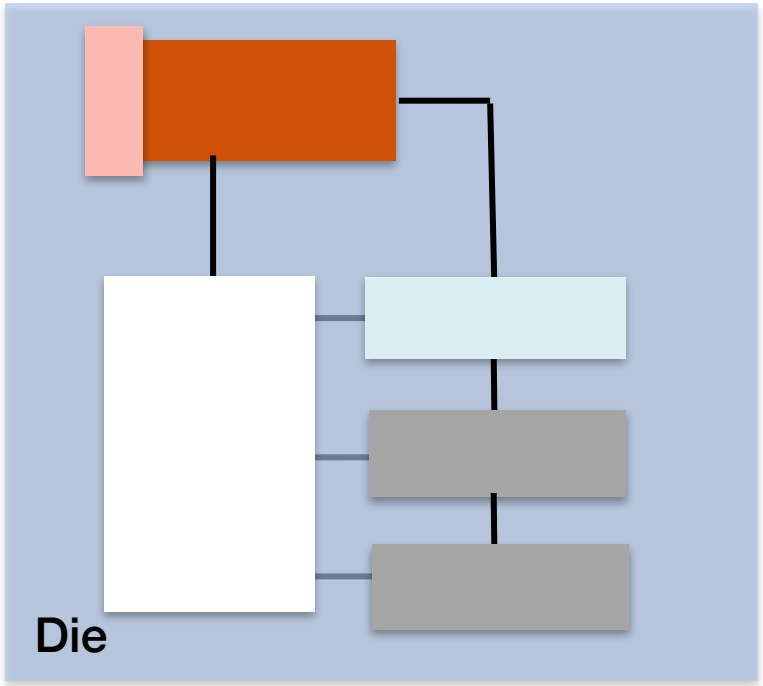

```
AddRoundKey(state, &w[0])
for i = 1 step 1 to 9
    SubBytes(state)
    ShiftRows(state)
    MixColumns(state)
    AddRoundKey(state, &w[i*4])
end for
SubBytes(state)
ShiftRows(state)
```

Non-volatile memory

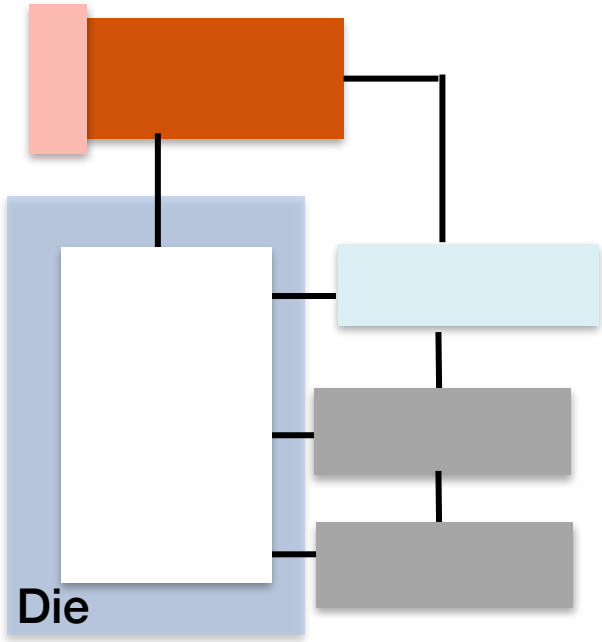
Volatile memory

What can be trusted?

MICROPROCESSORS VS MICROCONTROLLERS



Microcontroller (MCU)



Microprocessor (CPU)

A (typical) smartcard

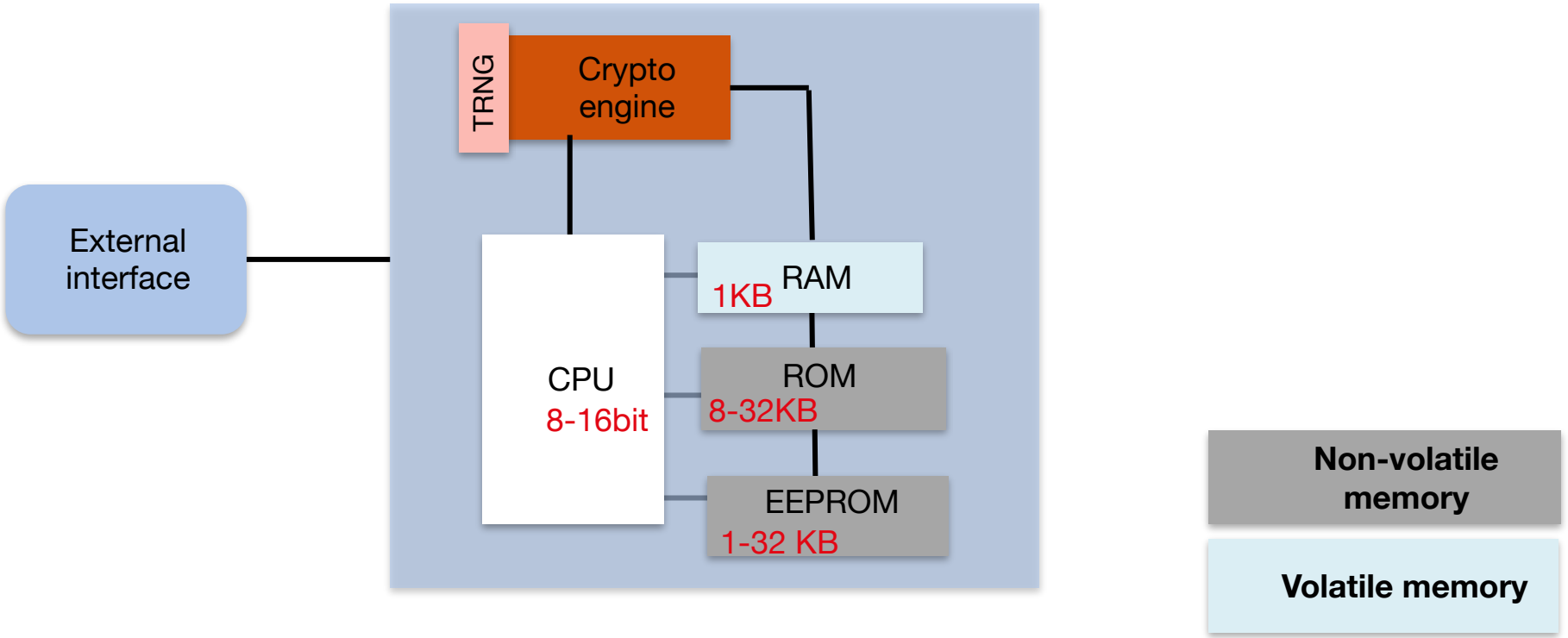


Image source: author collection

Foundation for secure software execution

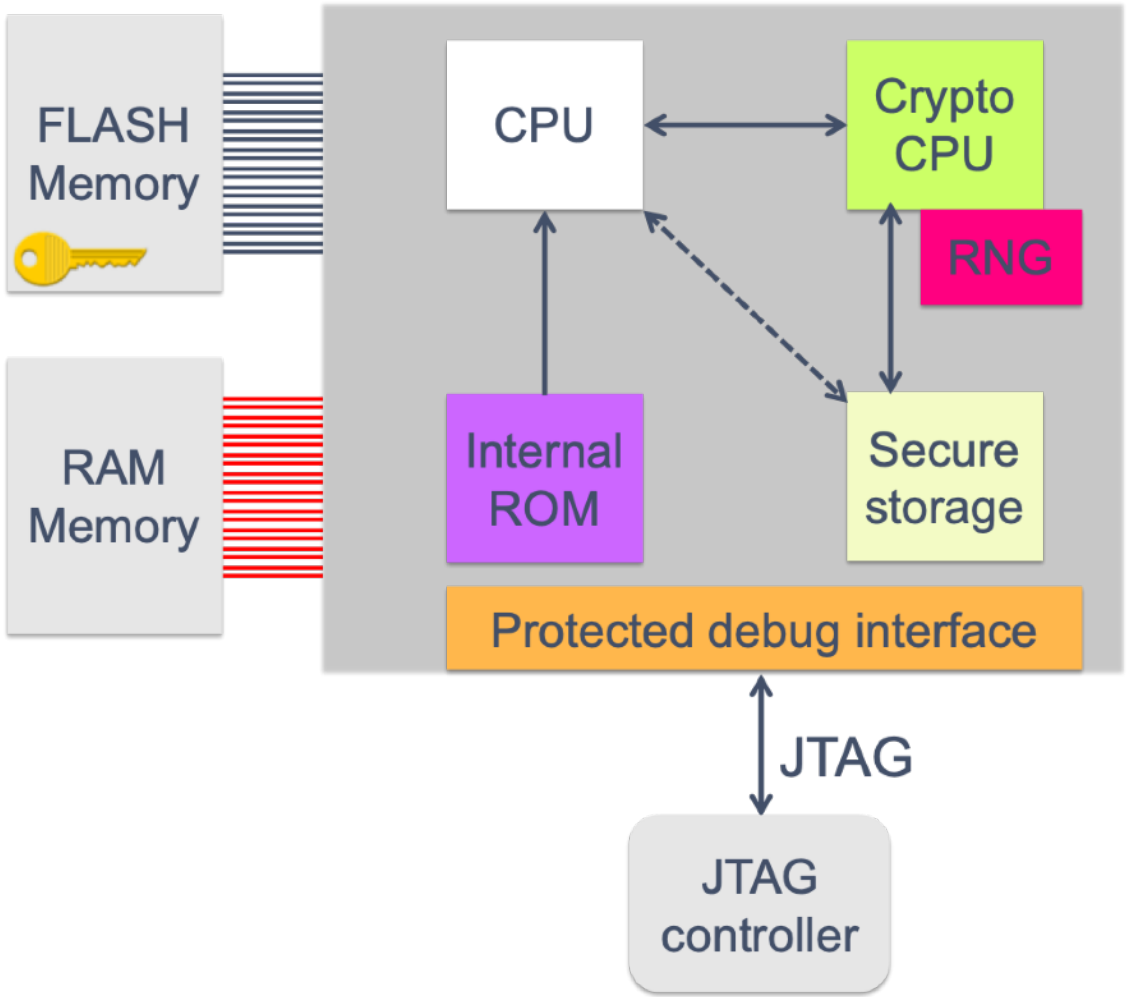
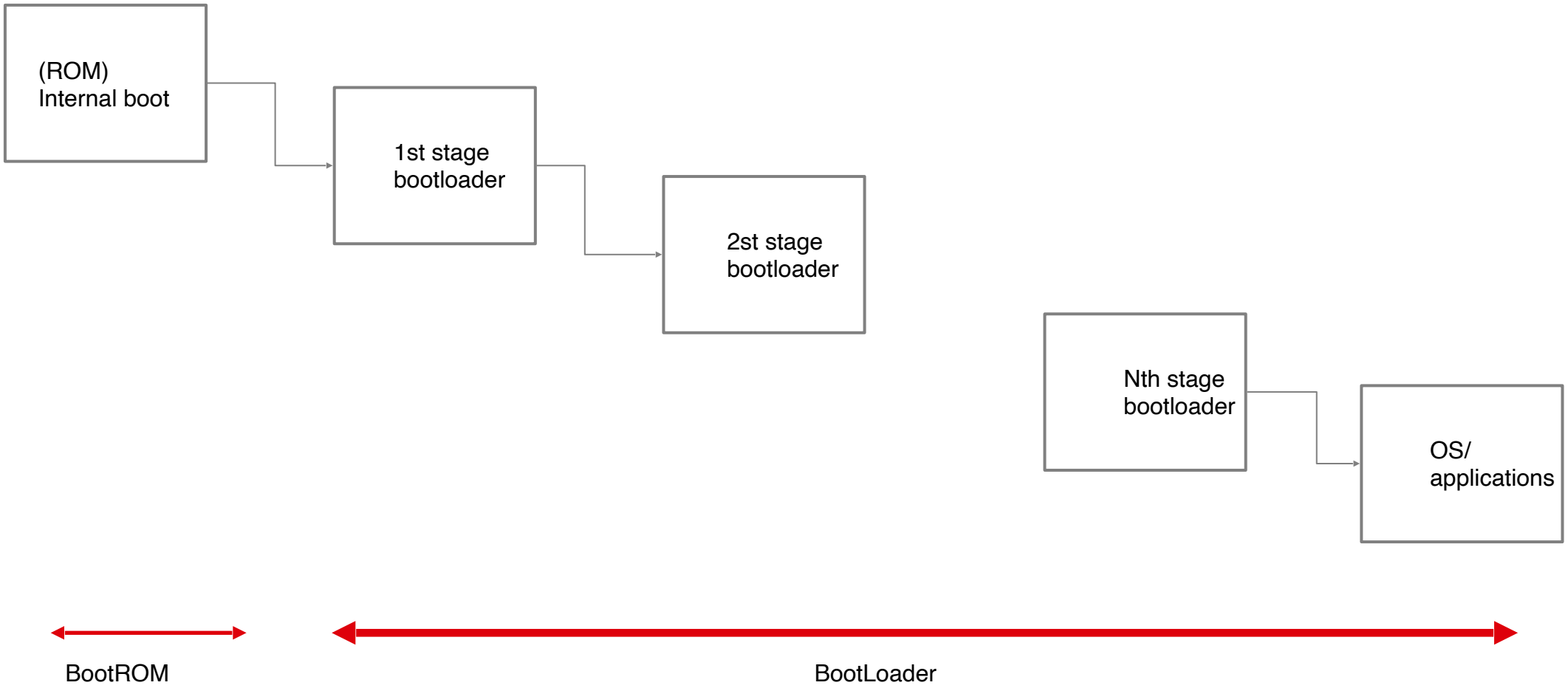


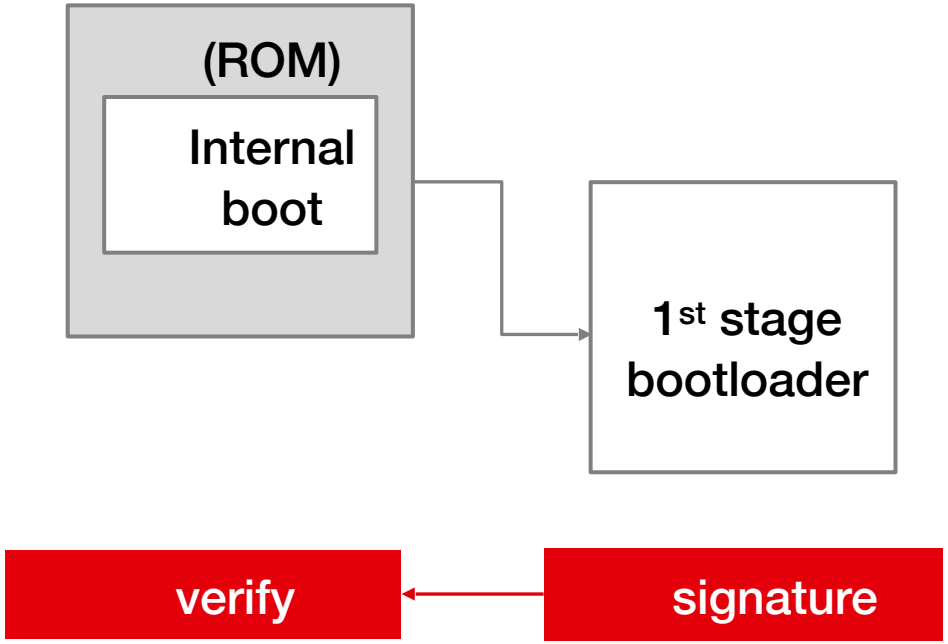
Image source: author collection

Hardware is the foundation which enables the secure execution of software (but it is not enough)

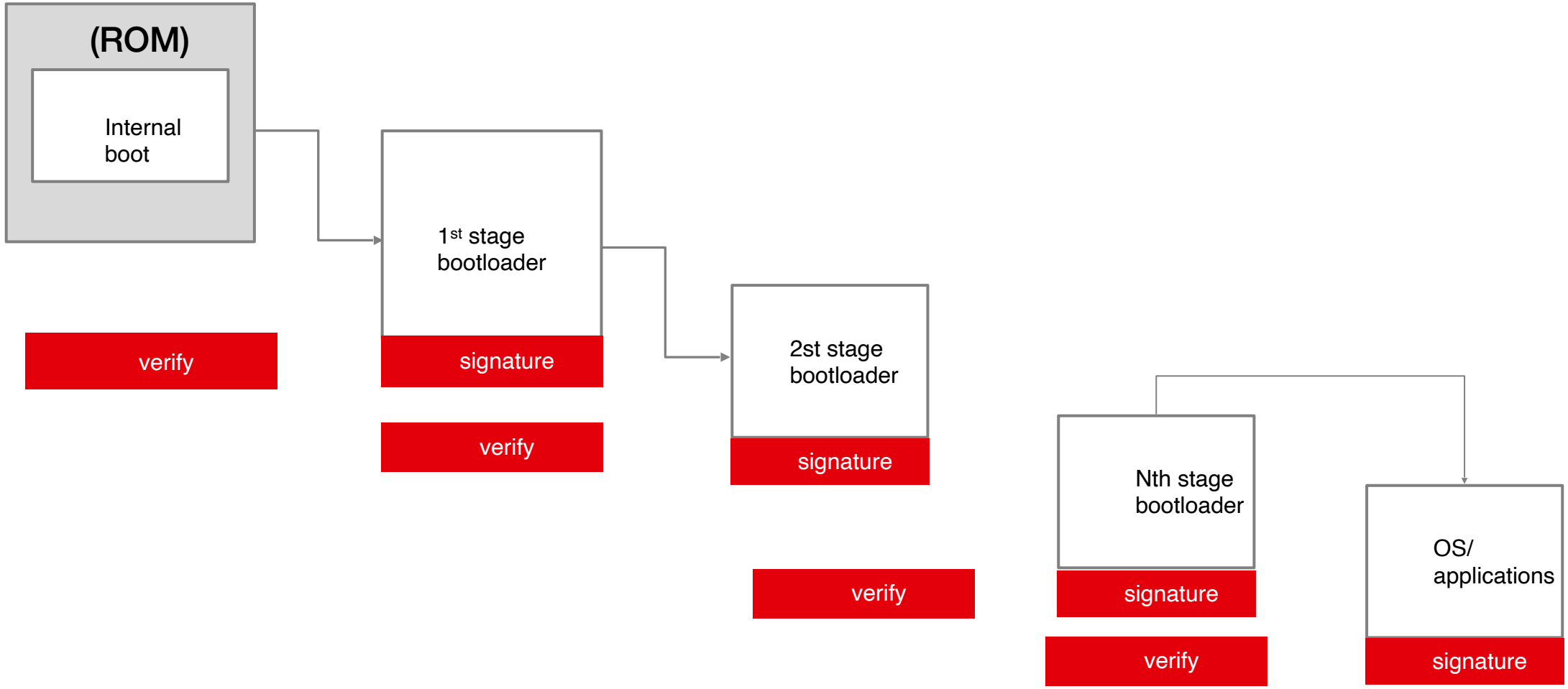
The Bootrom and the Bootloader



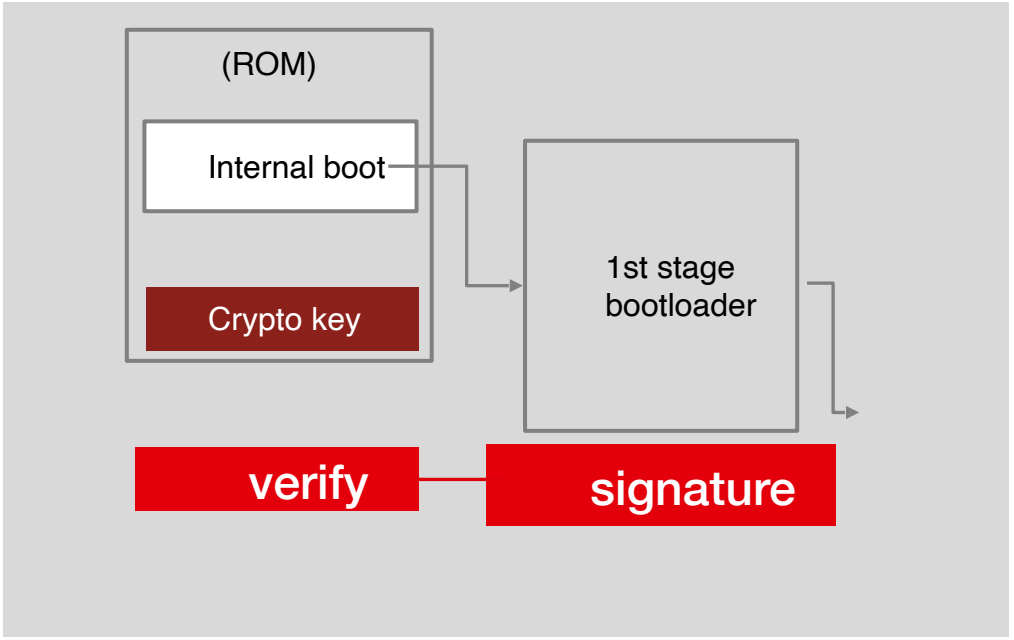
Secure Boot



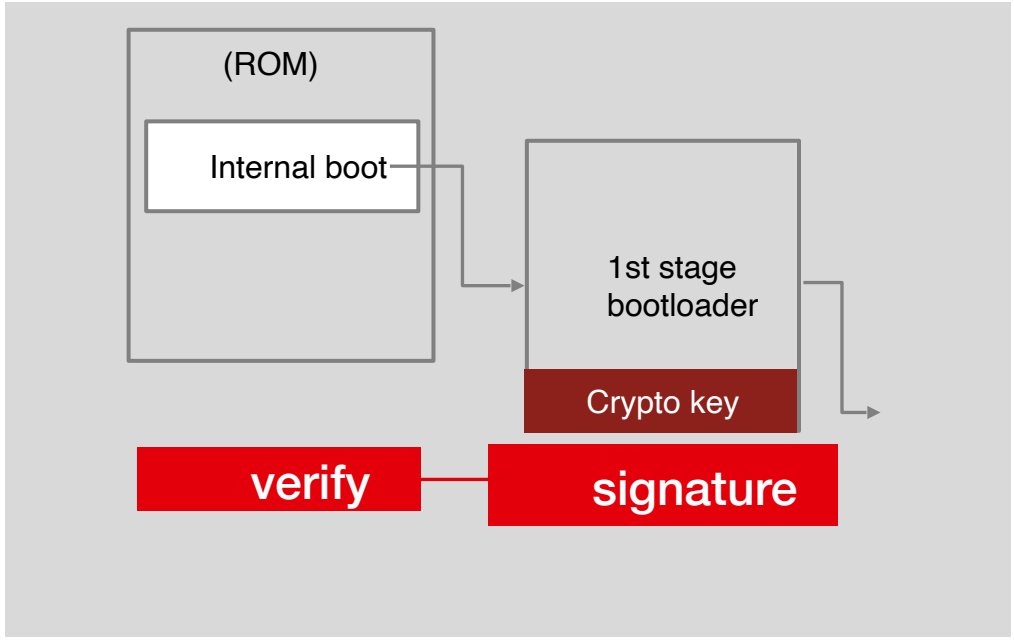
Secure Boot


(ROM)
Internal boot
verify
1st stage bootloader
signature
verify
2st stage bootloader
signature
verify
Nth stage bootloader
signature
verify
OS/ applications
signature

Design Decisions



Option A



Option B

Recommened reading: 20 ways to bypass secure boot, Job de Haas

Take aways

1. Extracting assets from unprotected hardware is easy
2. Hardware is the foundation for secure software execution
3. What can be trusted?

Lets play a game?

In which category you think this devices fits?

COTS ▲ Secure system

Lets play a game

ICS › 43 › 43.040 › 43.040.15

ISO/SAE 21434:2021

Road vehicles — Cybersecurity engineering

ABSTRACT PREVIEW

This document specifies engineering requirements for cybersecurity risk management regarding concept, product development, production, operation, maintenance and decommissioning of electrical and electronic (E/E) systems in road vehicles, including their components and interfaces.

A framework is defined that includes requirements for cybersecurity processes and a common language for communicating and managing cybersecurity risk.

This document is applicable to series production road vehicle E/E systems, including their components and interfaces, whose development or modification began after the publication of this document.

This document does not prescribe specific technology or solutions related to cybersecurity.

GENERAL INFORMATION

Status : Published
Publication date : 2021-08
Edition : 1
Number of pages : 81
Technical Committee : ISO/TC 22/SC 32 Electrical and electronic components and general system aspects
ICS : 43.040.15 Car informatics. On board computer systems

COTS ▲ Secure system

QUESTIONS?