

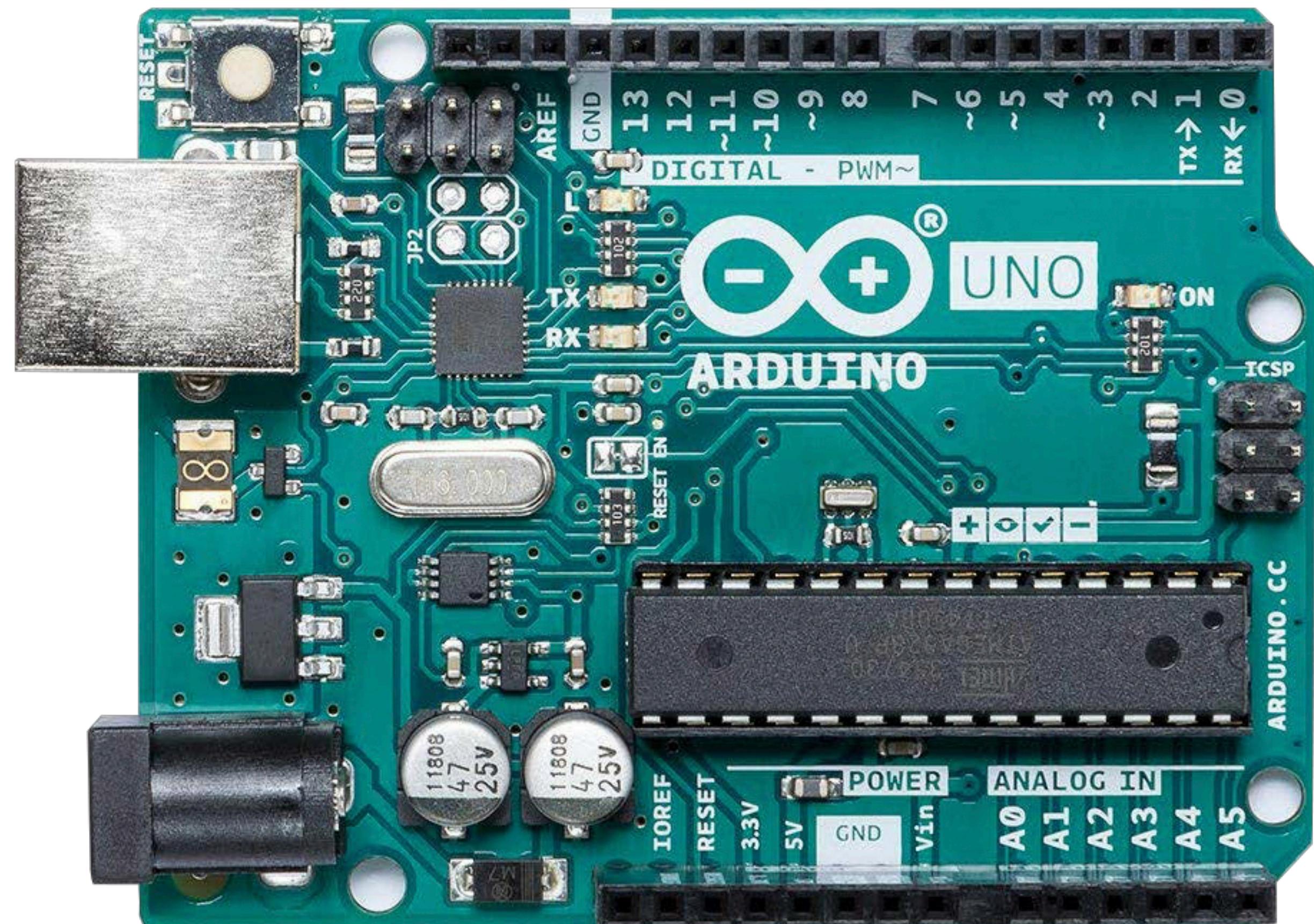
Serielle Kommunikation

UART

I2C

SPI

Aufgabe



Serielle Kommunikation

LEDs in Benutzung
"Kommunikation" über
UART

UART

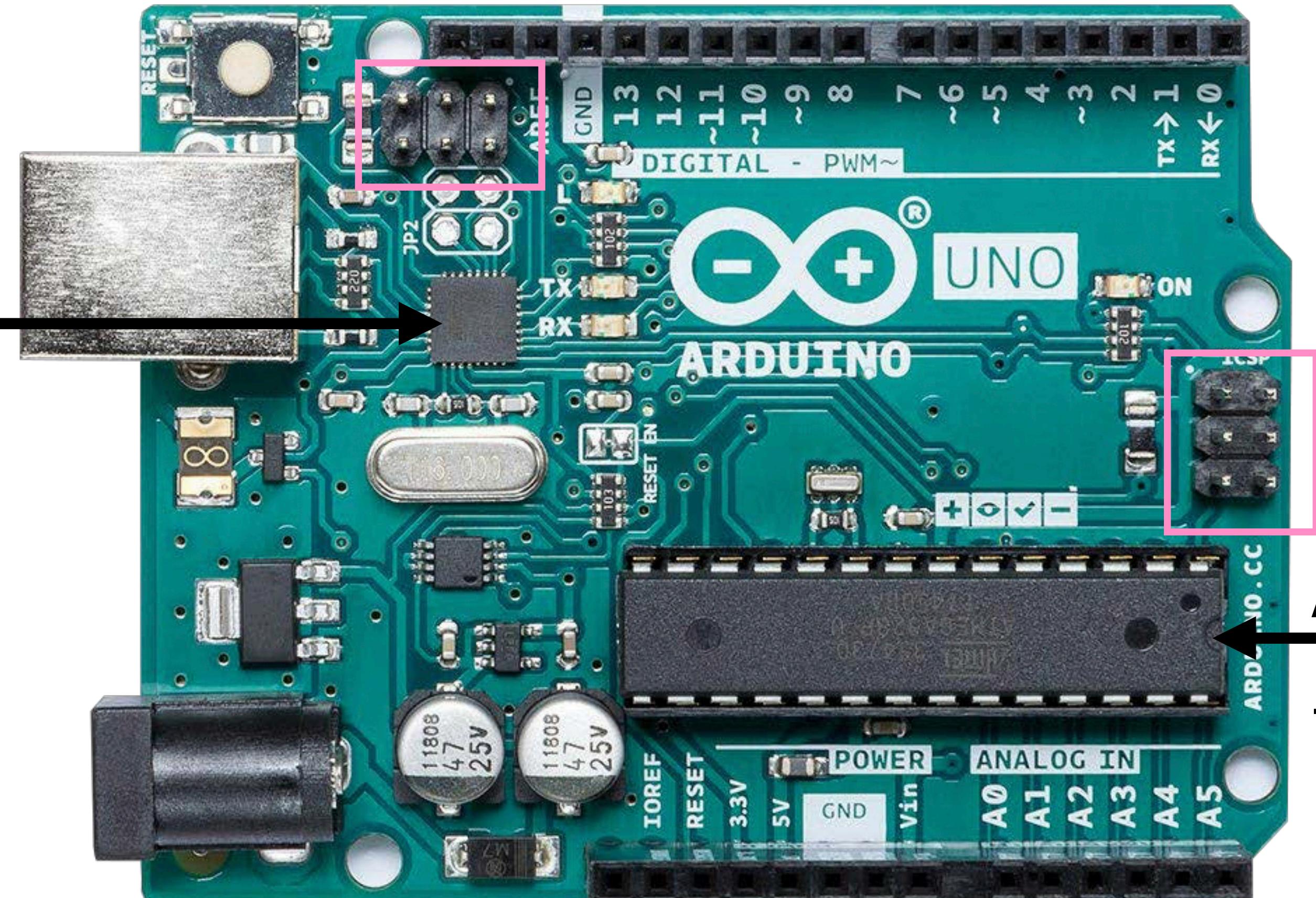


Serielle Kommunikation

ISP → In-System Programming

ICSP → In-Circuit Serial Programming

ATmega16U2 oder ATmega8U2

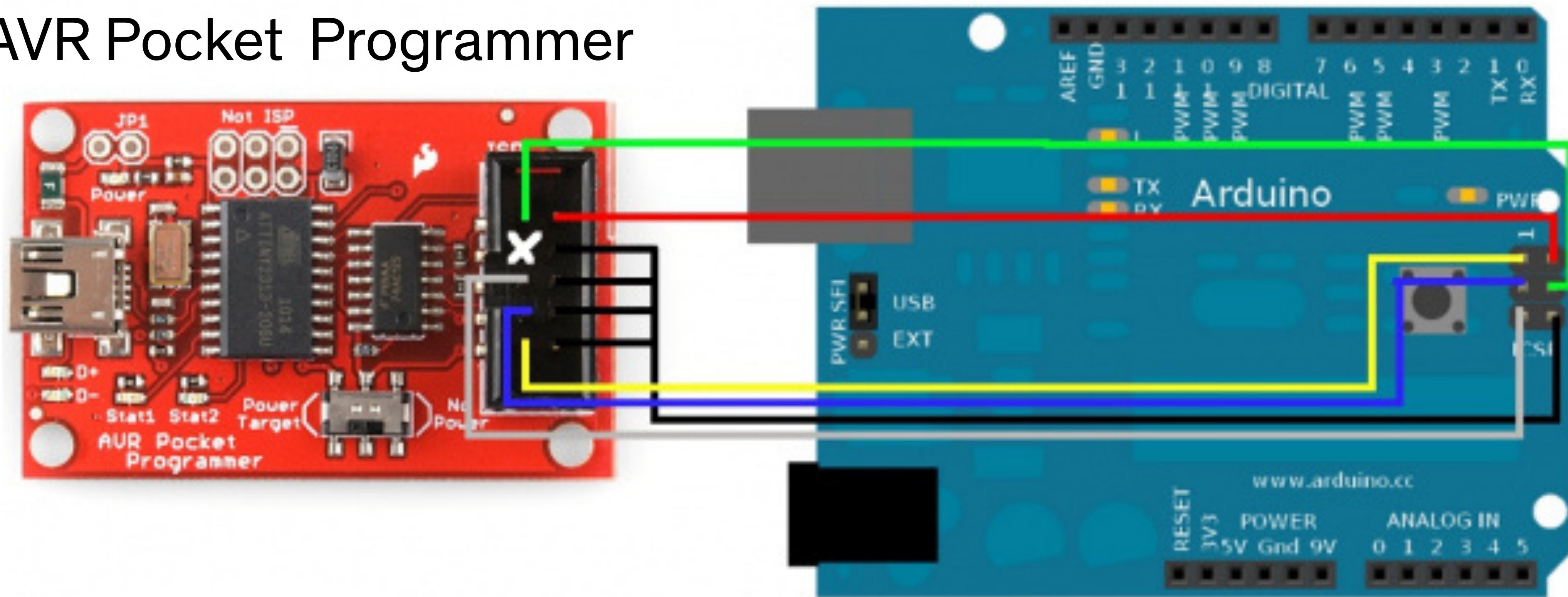


ATmega328p

- Bootloader aufspielen

Serielle Kommunikation

AVR Pocket Programmer



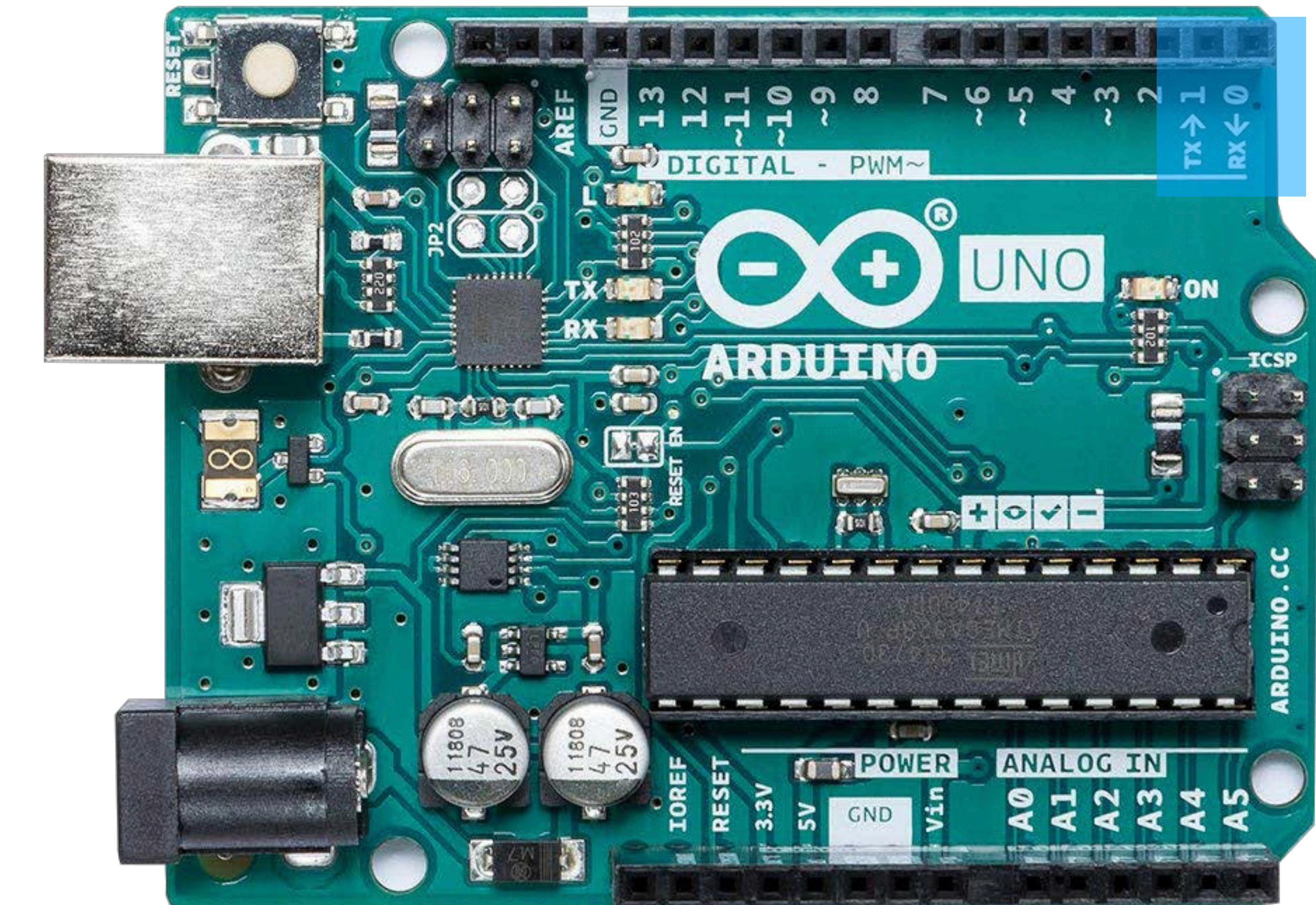
MOSI 1 2 5V
3 4
RST 5 6 } GND
SCK 7 8
MISO 9 10

MISO 1 2 5V
3 4 MOSI
RST 5 6 GND

Serielle Kommunikation

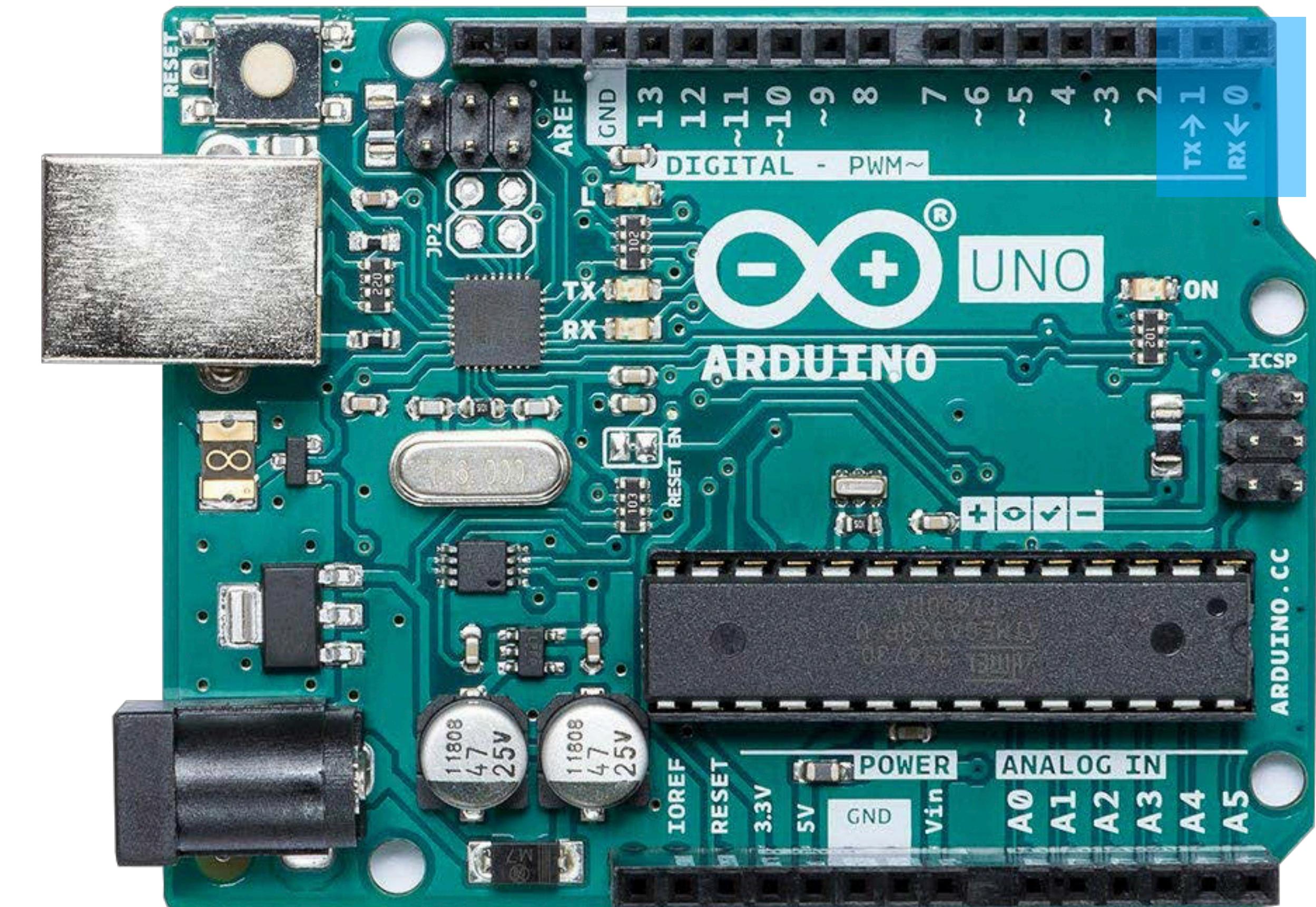
Universal Asynchronous Reception and Transmission

UART

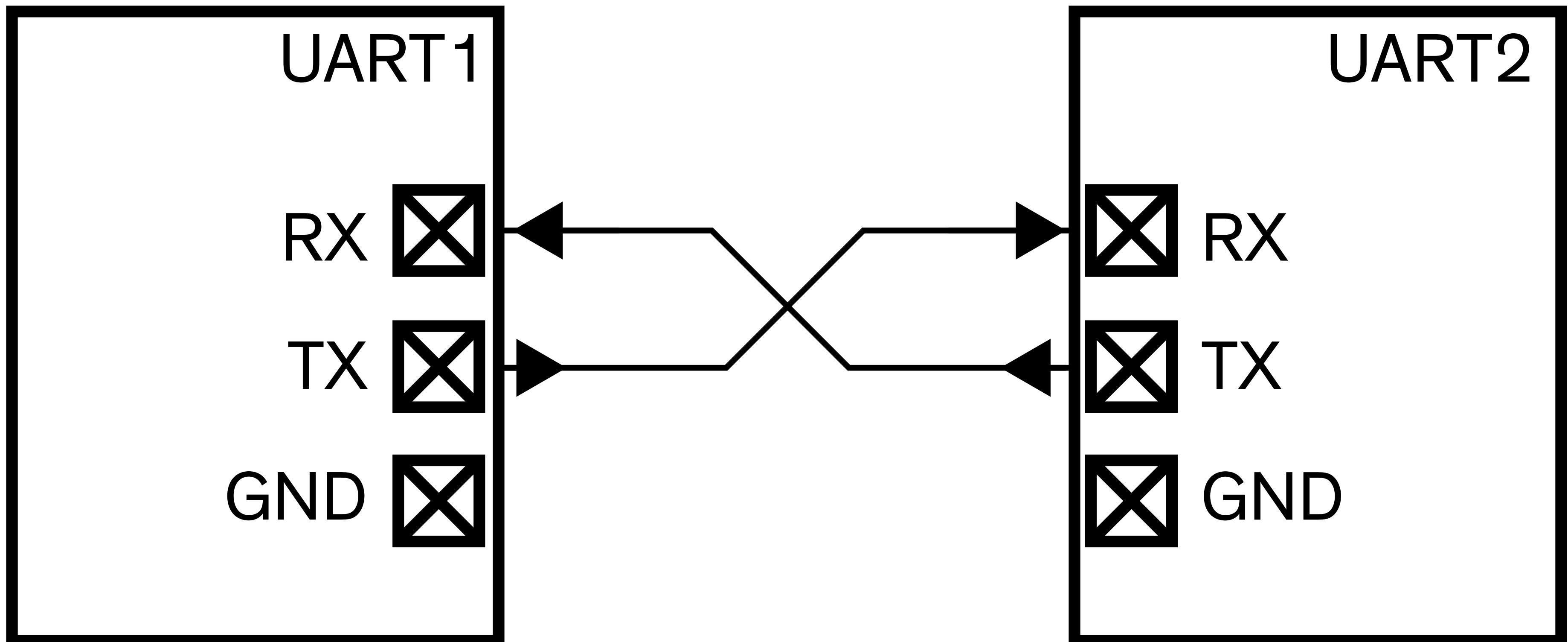


Universal Asynchronous Reception and Transmission

Tx → Transmit
Rx → Receive

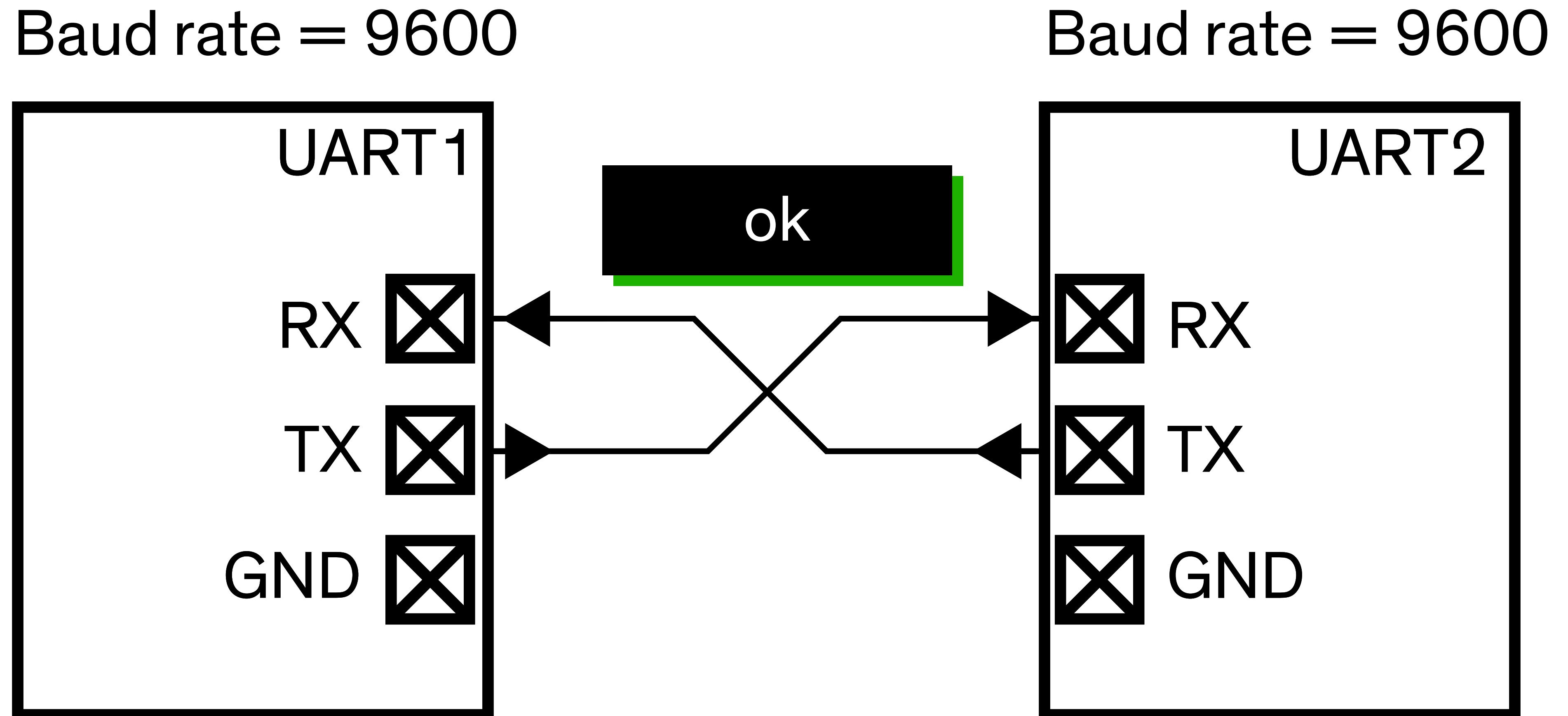


UART



UART

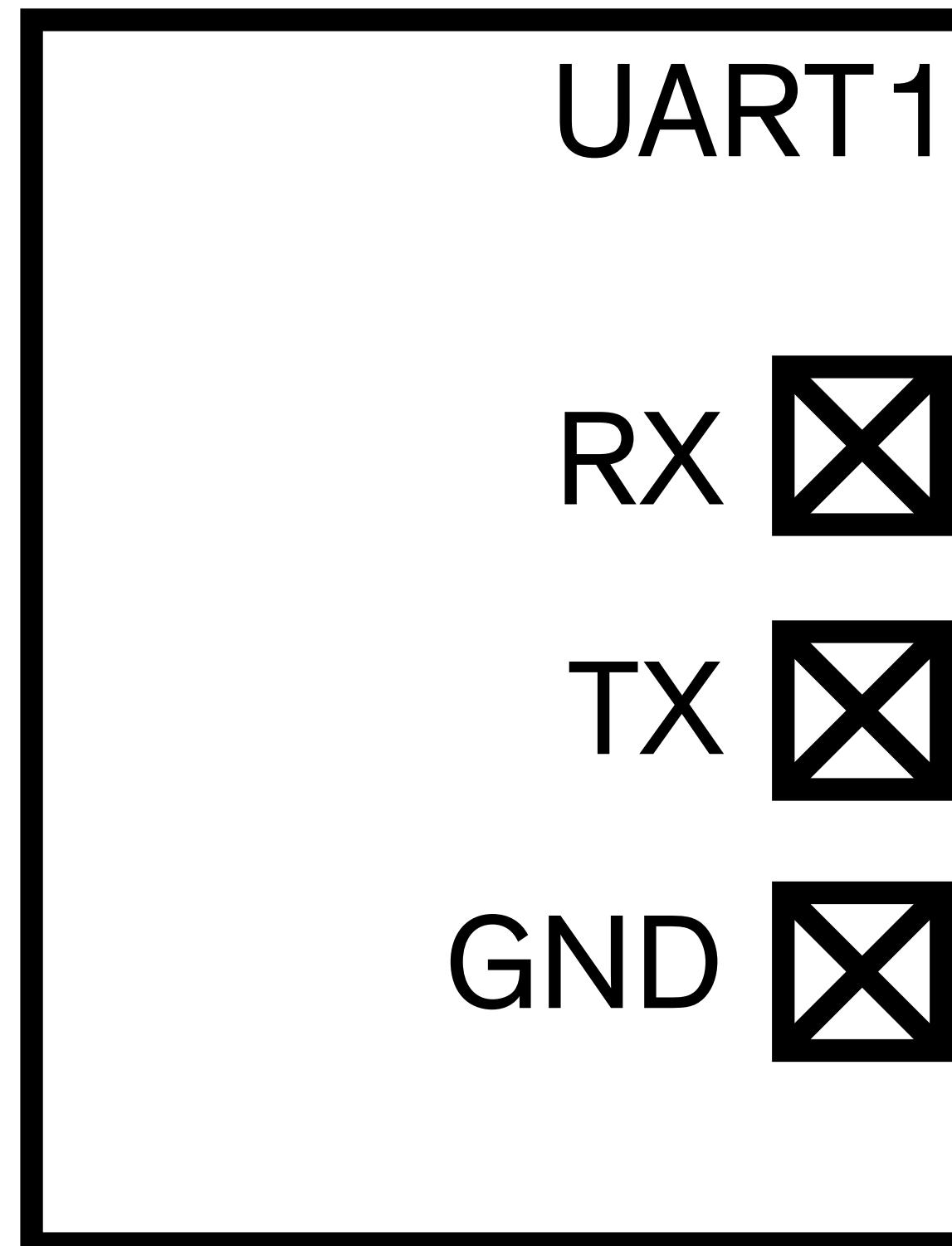
Bits (Symbole) pro Sekunde



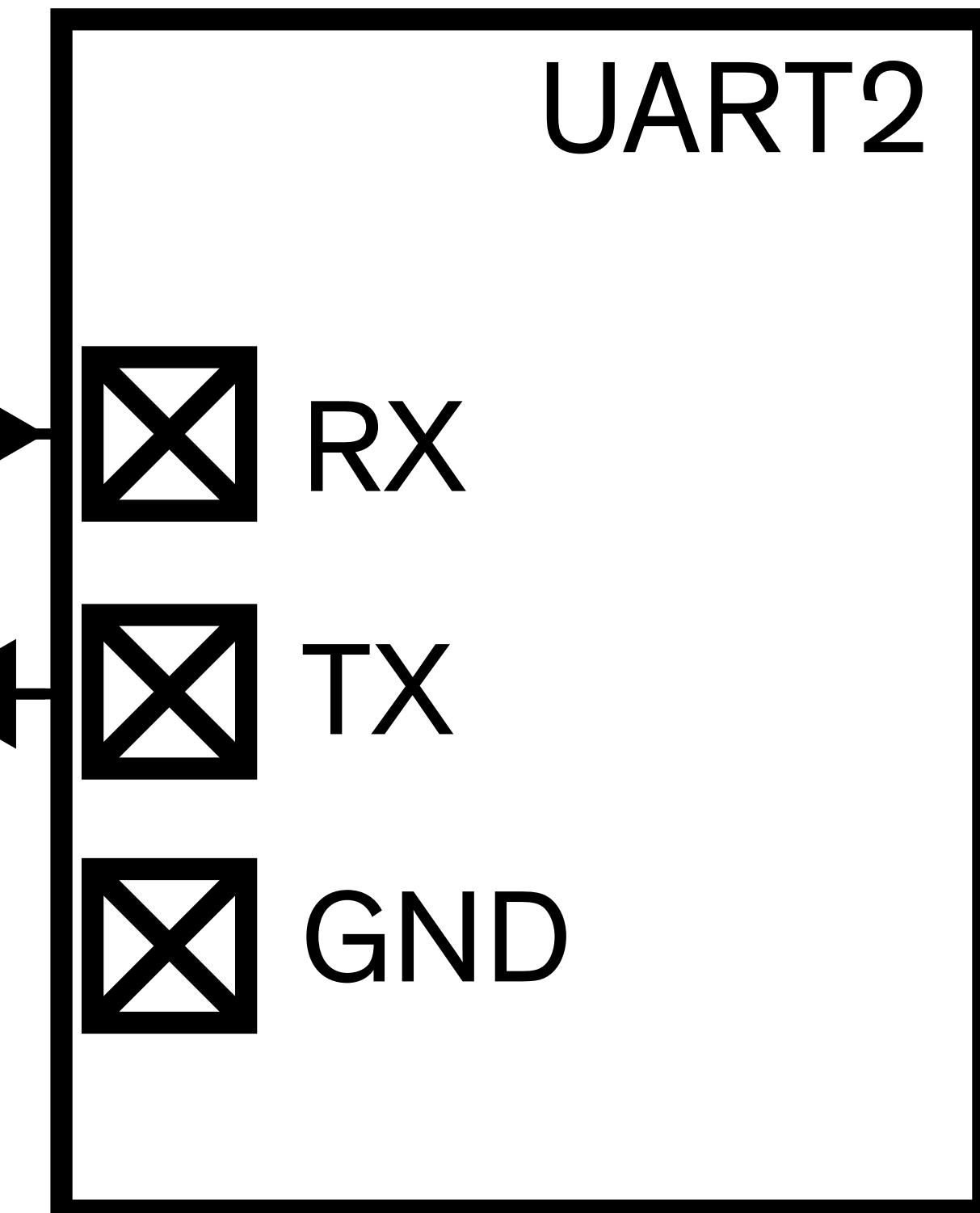
UART

Bits (Symbole) pro Sekunde

Baud rate = 9600



Baud rate = 115,200



UART

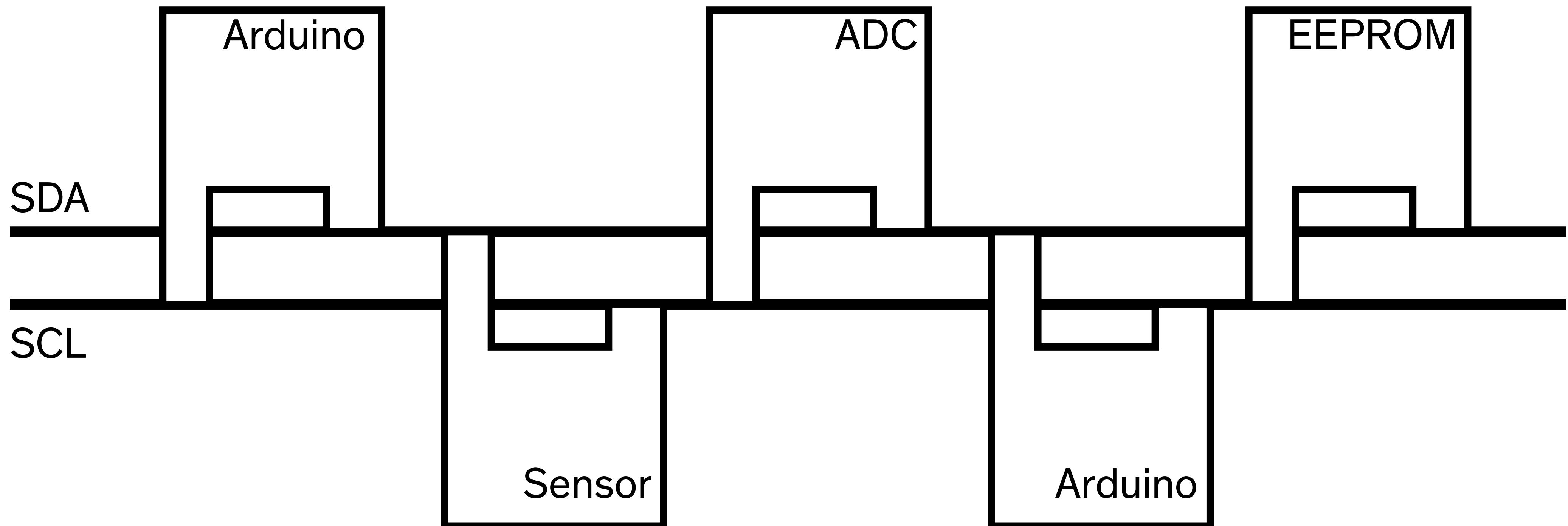
Vorteile

- Simpel aufzubauen (z.B. zwei Arduinos miteinander kommunizieren lassen)
- Online gut dokumentiert
- Braucht keine Clock

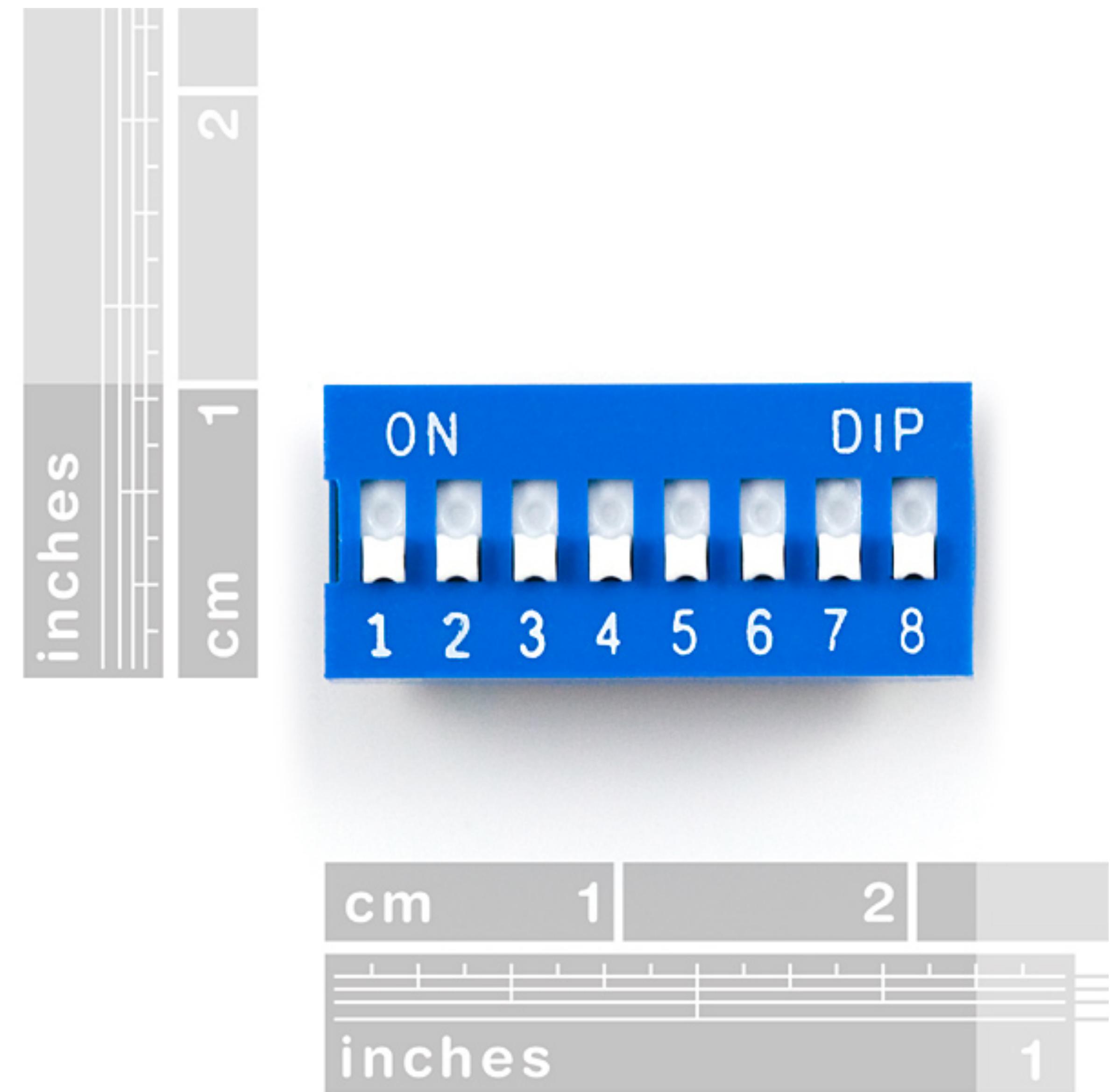
Nachteile

- Ein Datenpaket (Frame) kann nur 9 Bits lang sein
- Overhead durch Start/Stopp Bits
- Nur eine Master / Slave Konfiguration
- Baud rates müssen "gleich" sein (10% Abweichung ok)
- Langsam

UART



I²C (oder I²C)



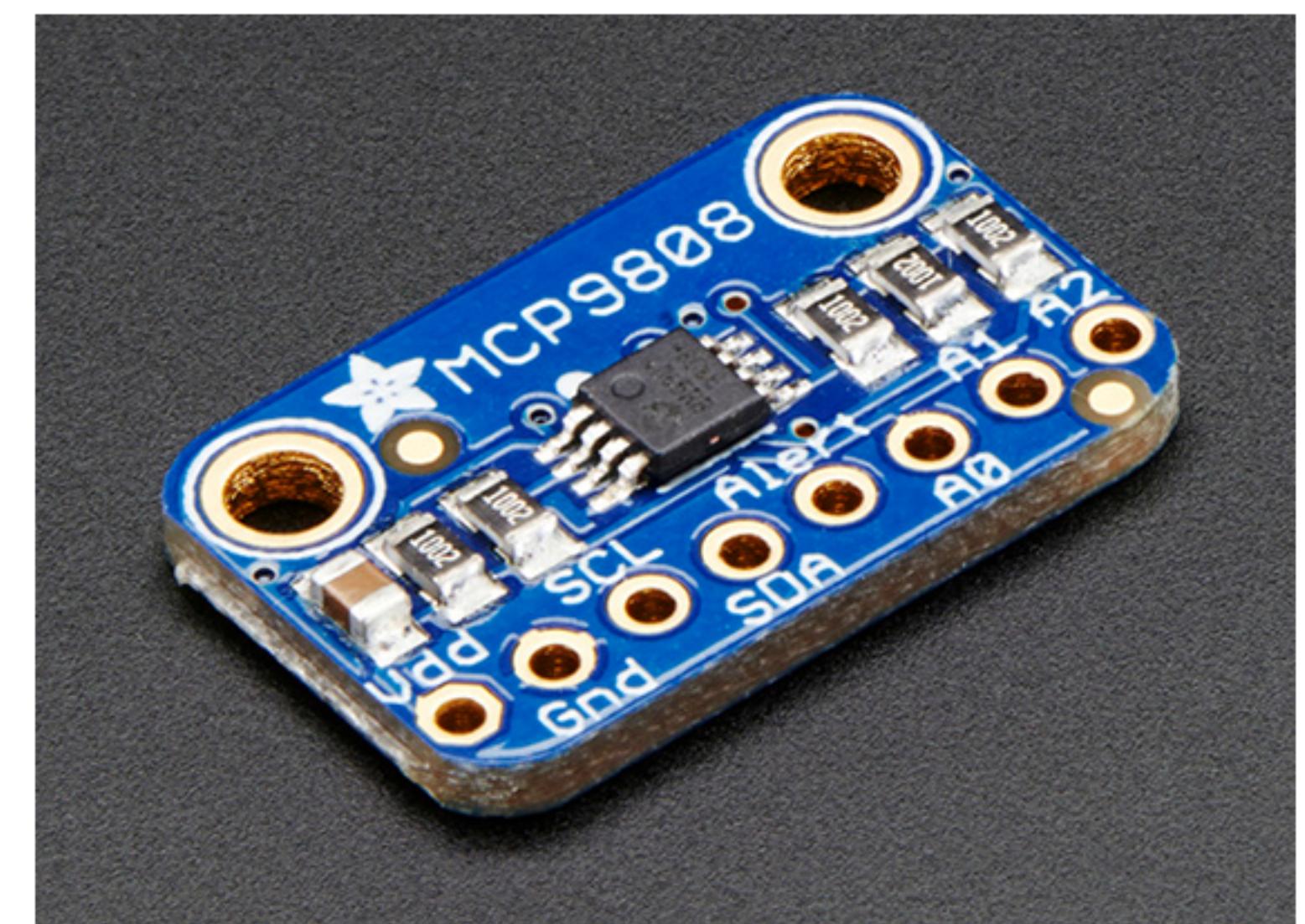
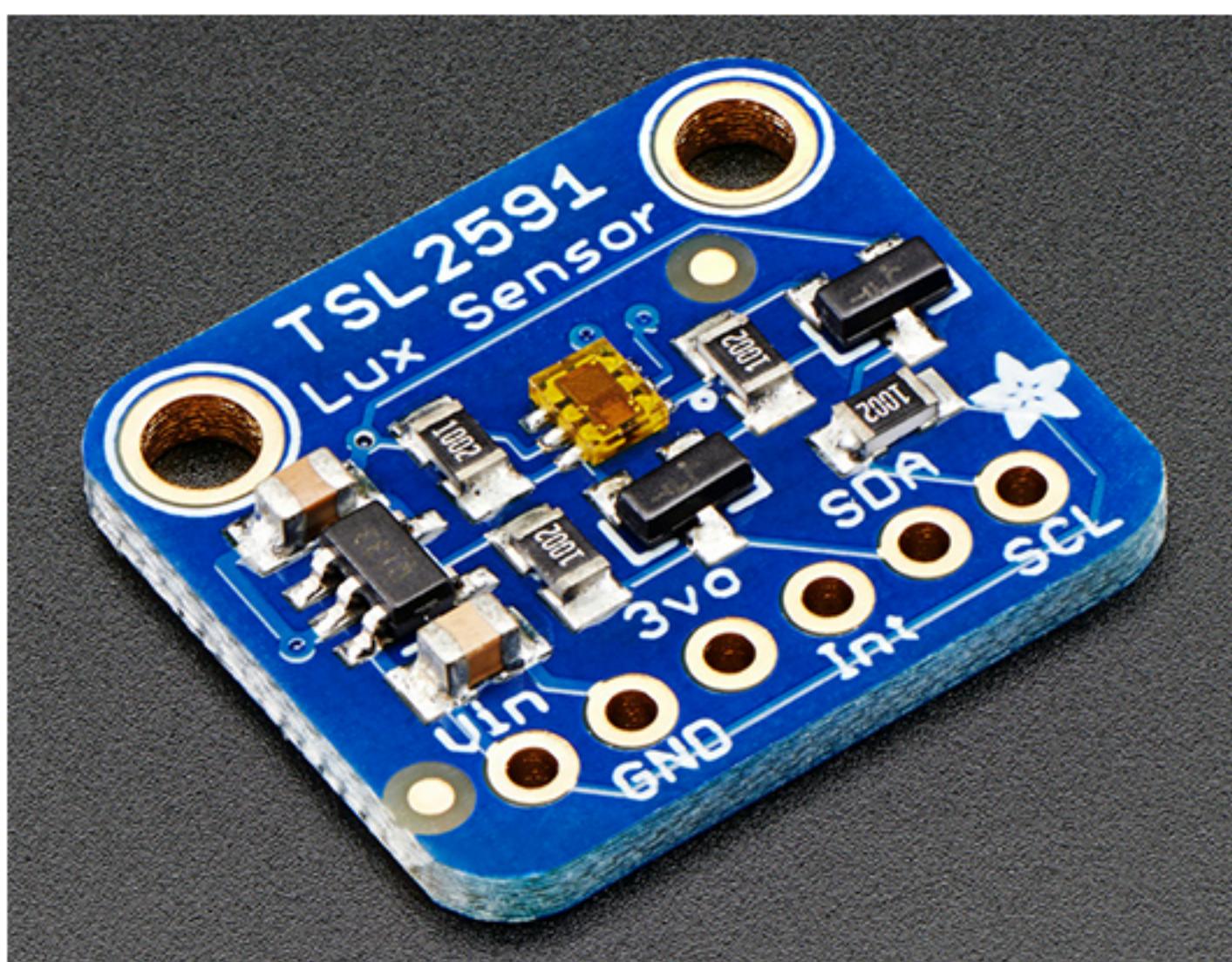
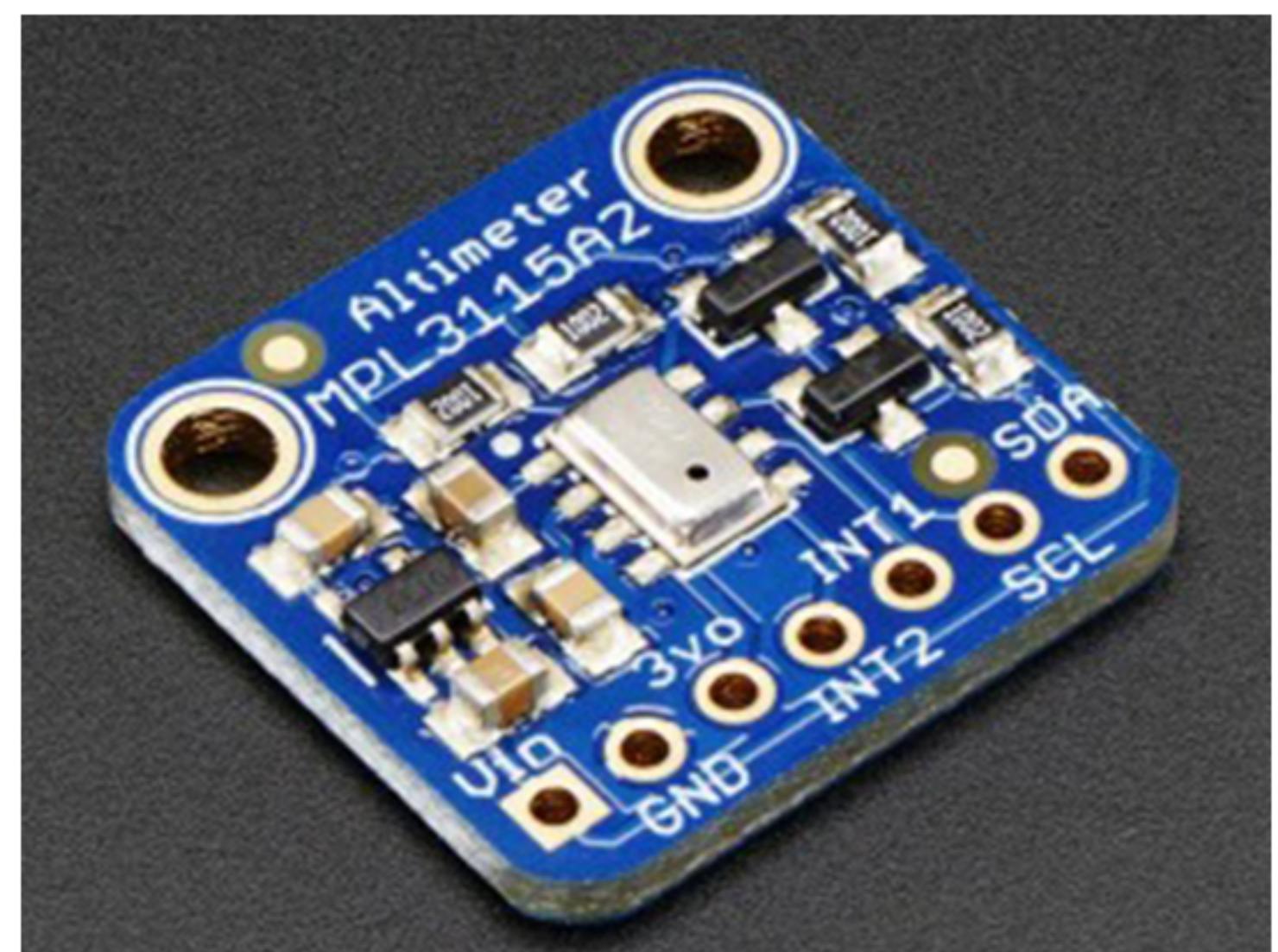
I²C (oder I²C)

Vorteile

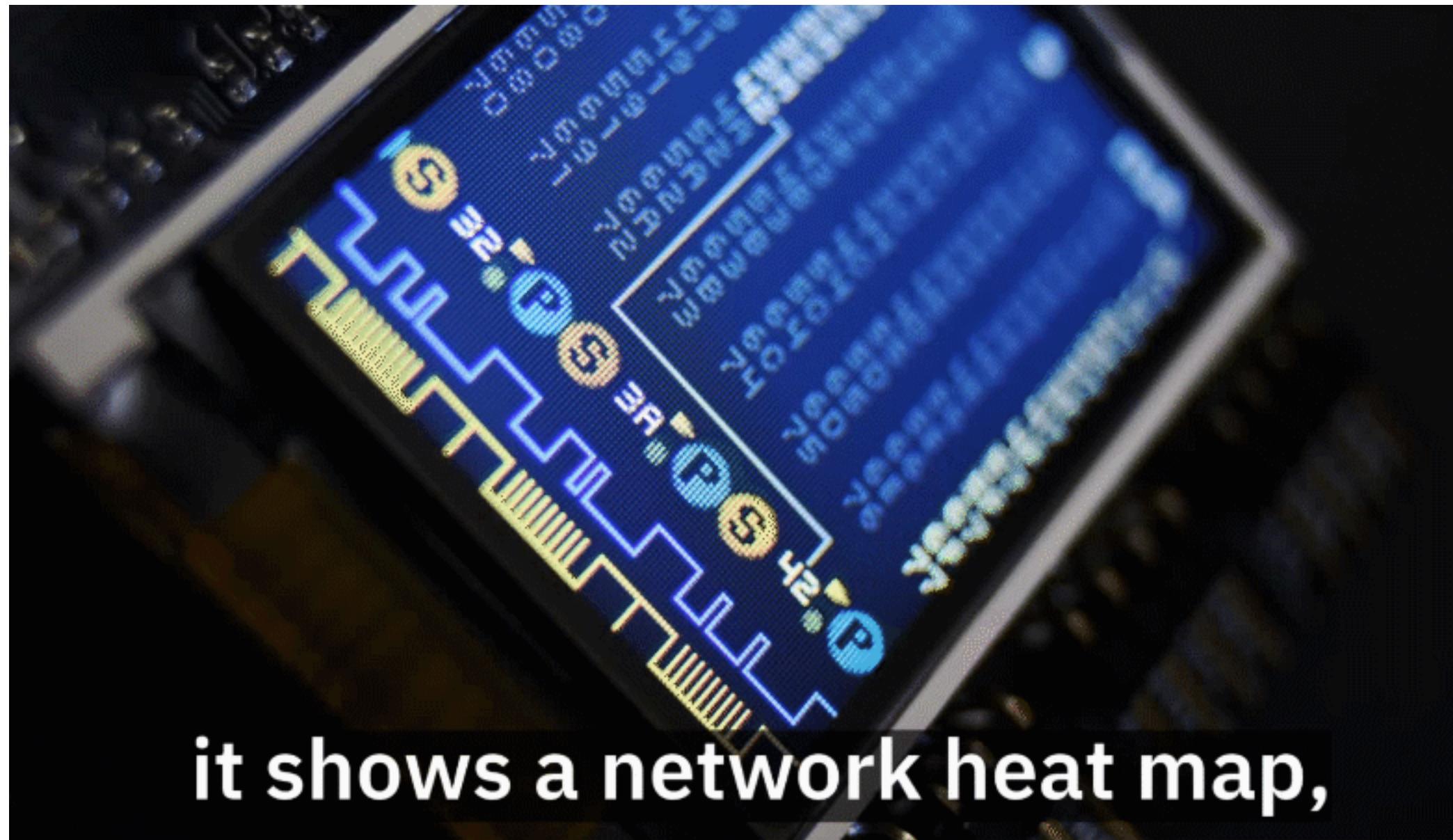
- Simpler Aufbau
- Wenig Pins in Benutzung
- Adressierungssystem (7-Bit, 8-Bit oder 10-Bit)
- Bis zu 128 Geräte addressierbar
- Klarer Kommunikationsweg
- Multiple Master

Nachteile

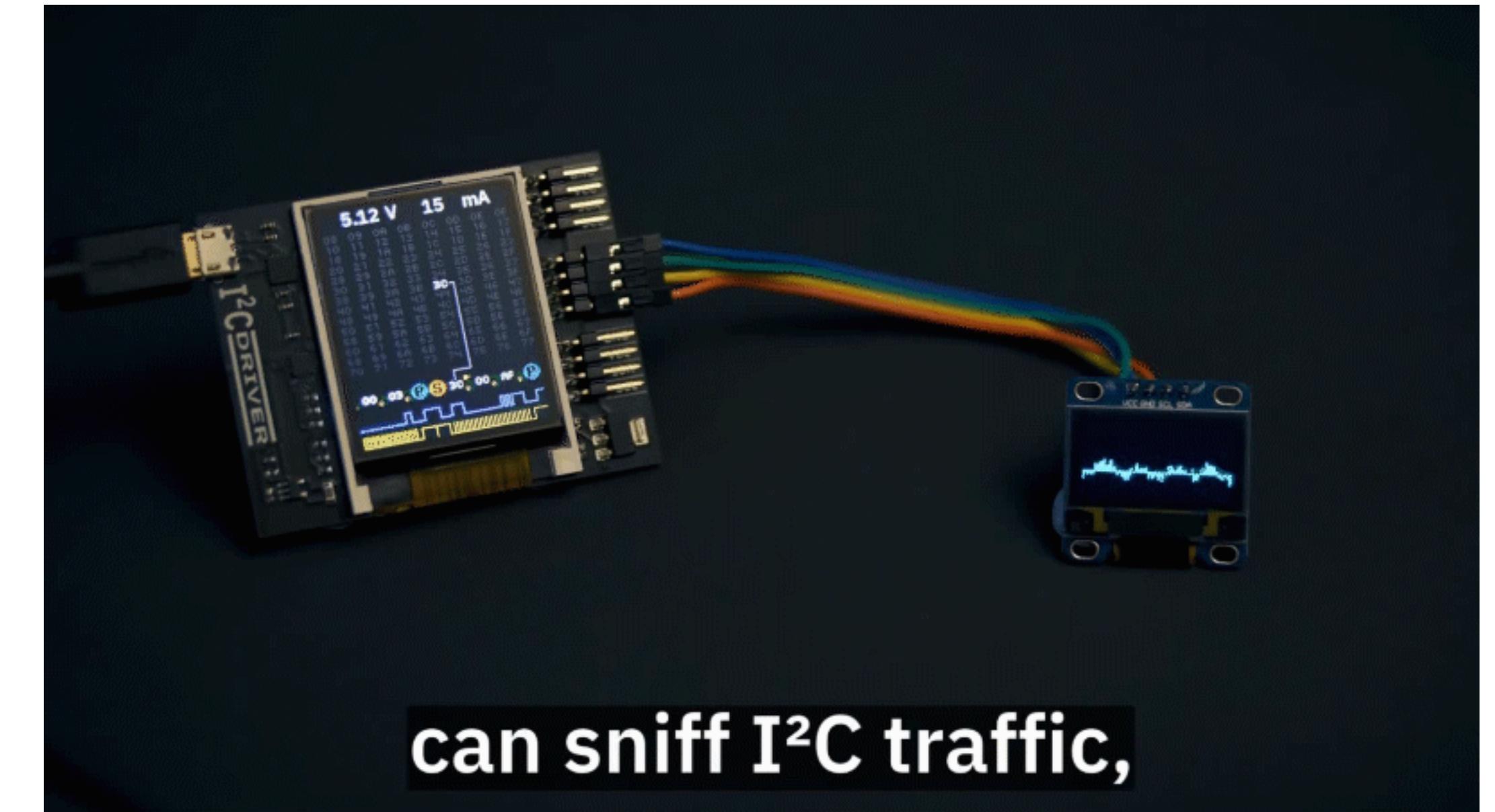
- Langsamer als SPI
- Geschwindigkeit hängt auch von Wahl/Qualität der Kabel, Datenvolumen und externem Rauschen ab
- Komplexität nimmt mit steigender Geräte Anzahl zu



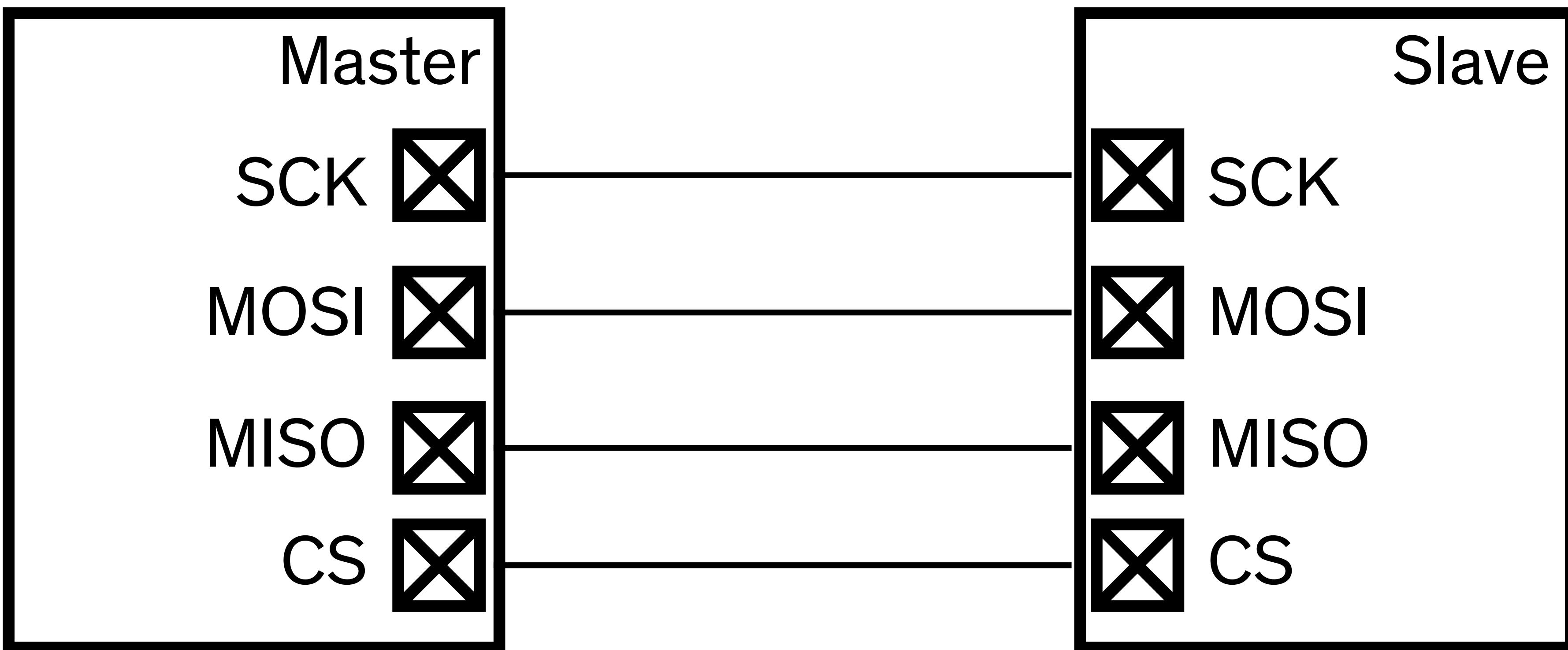
I2C



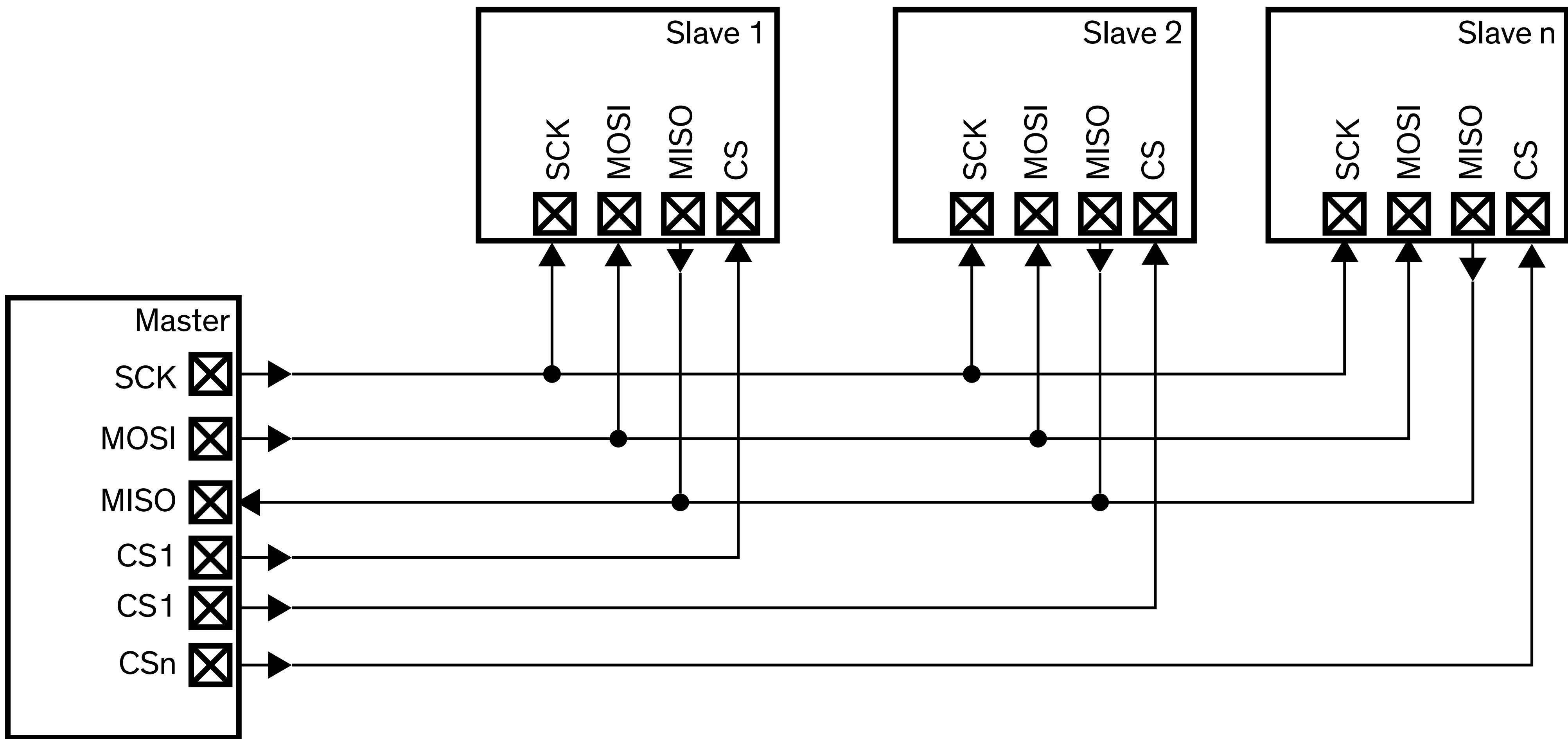
it shows a network heat map,



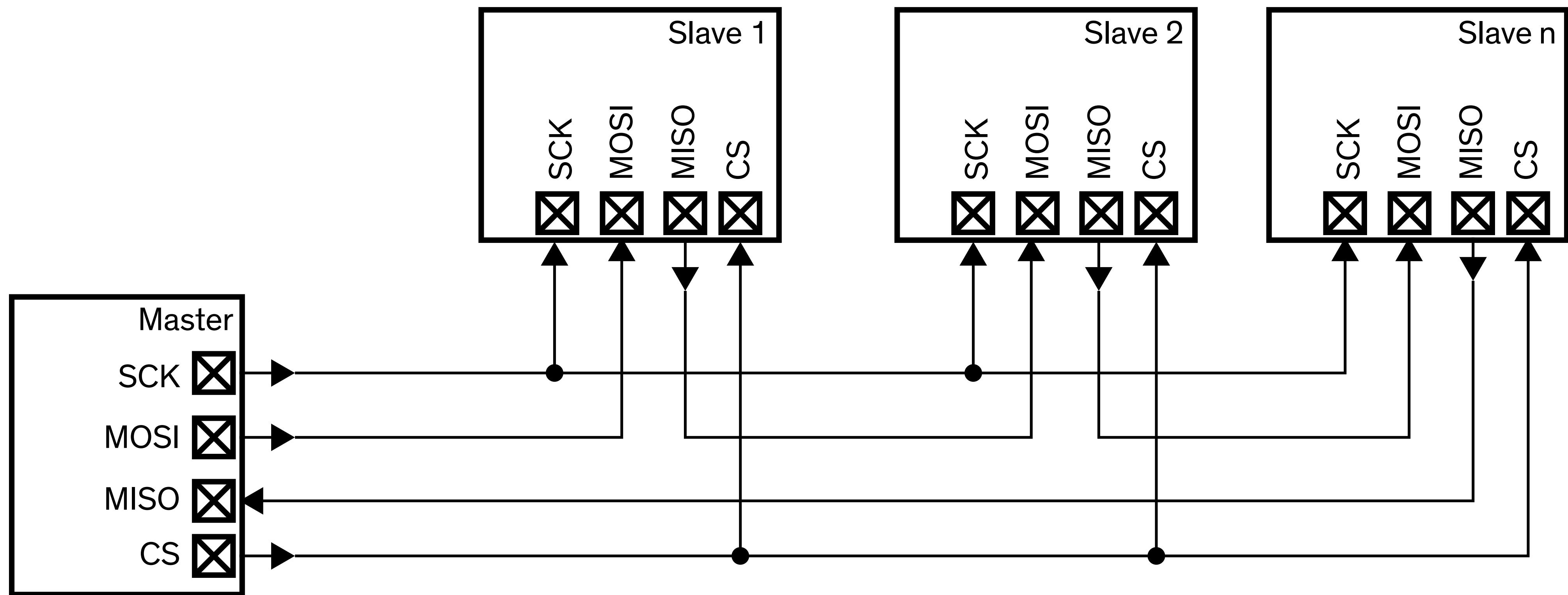
can sniff I²C traffic,



SPI



SPI



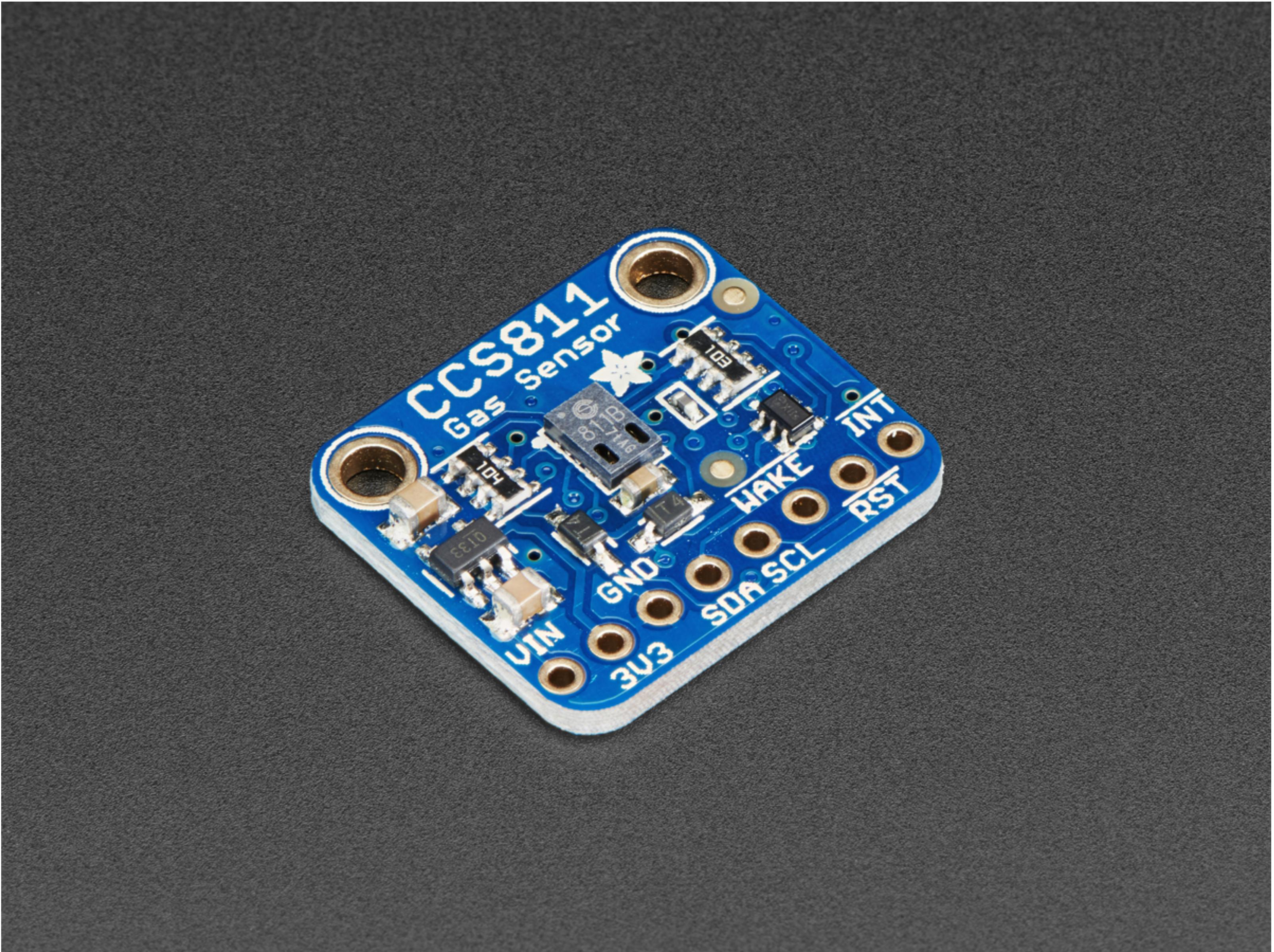
Vorteile

- Simples Protokoll
- Kein Adressierungssystem wie bei I2C
- Kein Limit an Geräten im Bus
- Wenig Pins in Benutzung
- Schnell: 8Mbits und mehr (Schneller als UART oder I2C)
- MISO und MOSI Leitungen sind separat (Gleichzeitig Daten empfangen und senden)

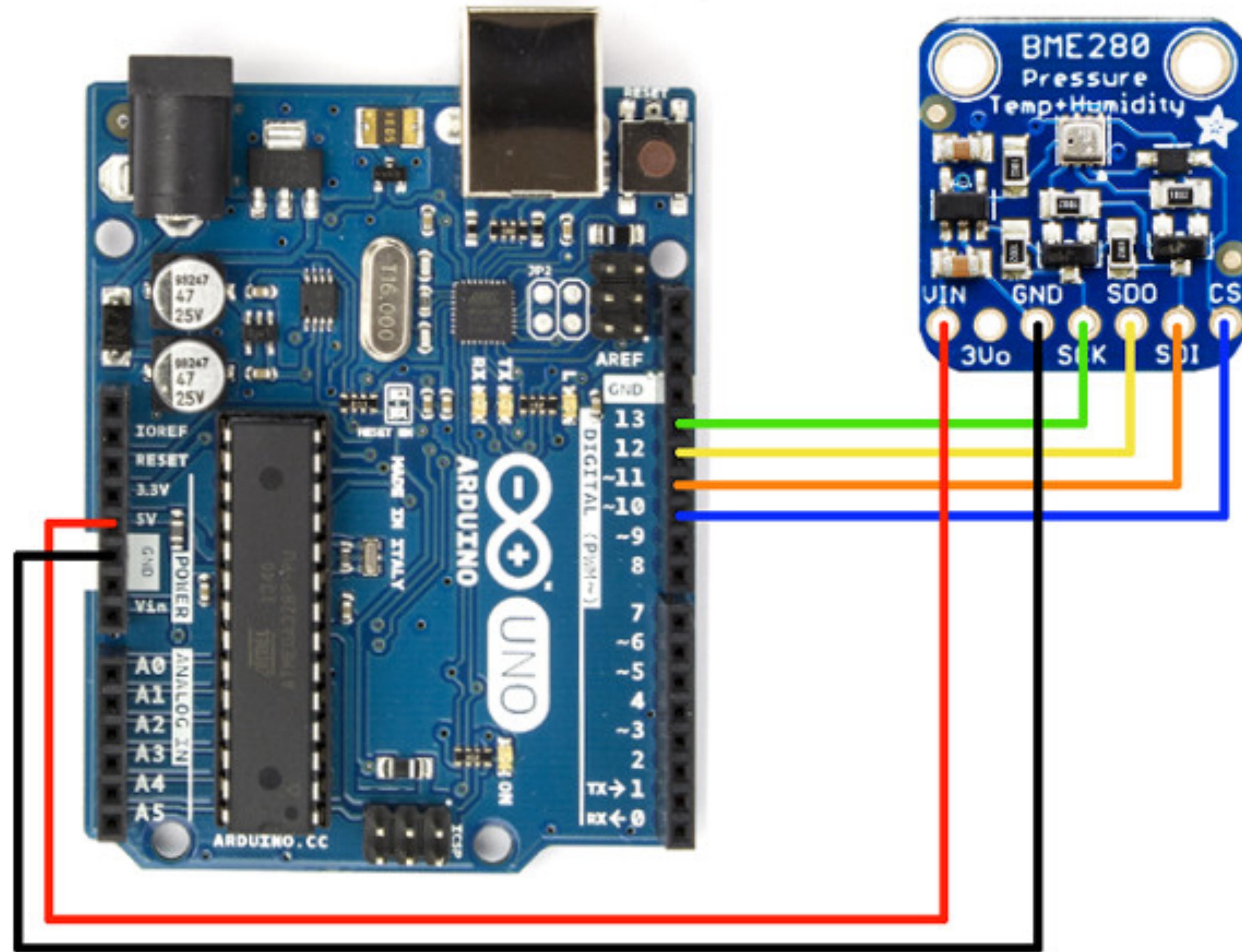
Nachteile

- Limitierung durch CS Pins (Außer bei Daisy-Chaining)
- Mehr Pins in Benutzung und "praktisches" Limit
- Single Master

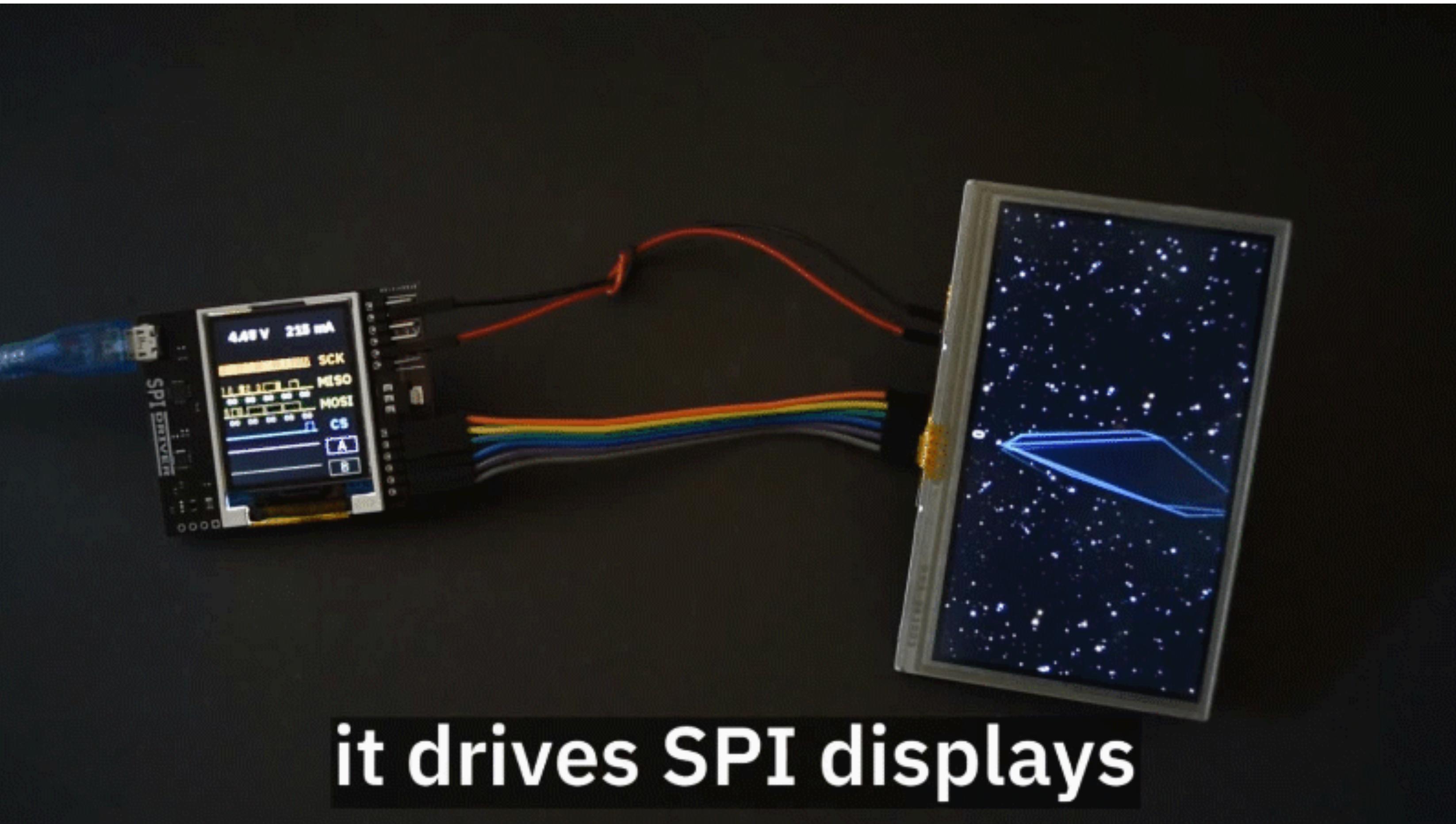
I2C



I²C



I2C



I2C

<https://www.arduino.cc/en/Reference/SPI>

<https://learn.sparkfun.com/tutorials/i2c>

<https://learn.sparkfun.com/tutorials/serial-communication/all>

<https://learn.sparkfun.com/tutorials/serial-peripheral-interface-spi>

UART vs I2C vs SPI

Aufgabe

Bringt beim nächsten mal eure Arduino Kits mit.

Fragen?