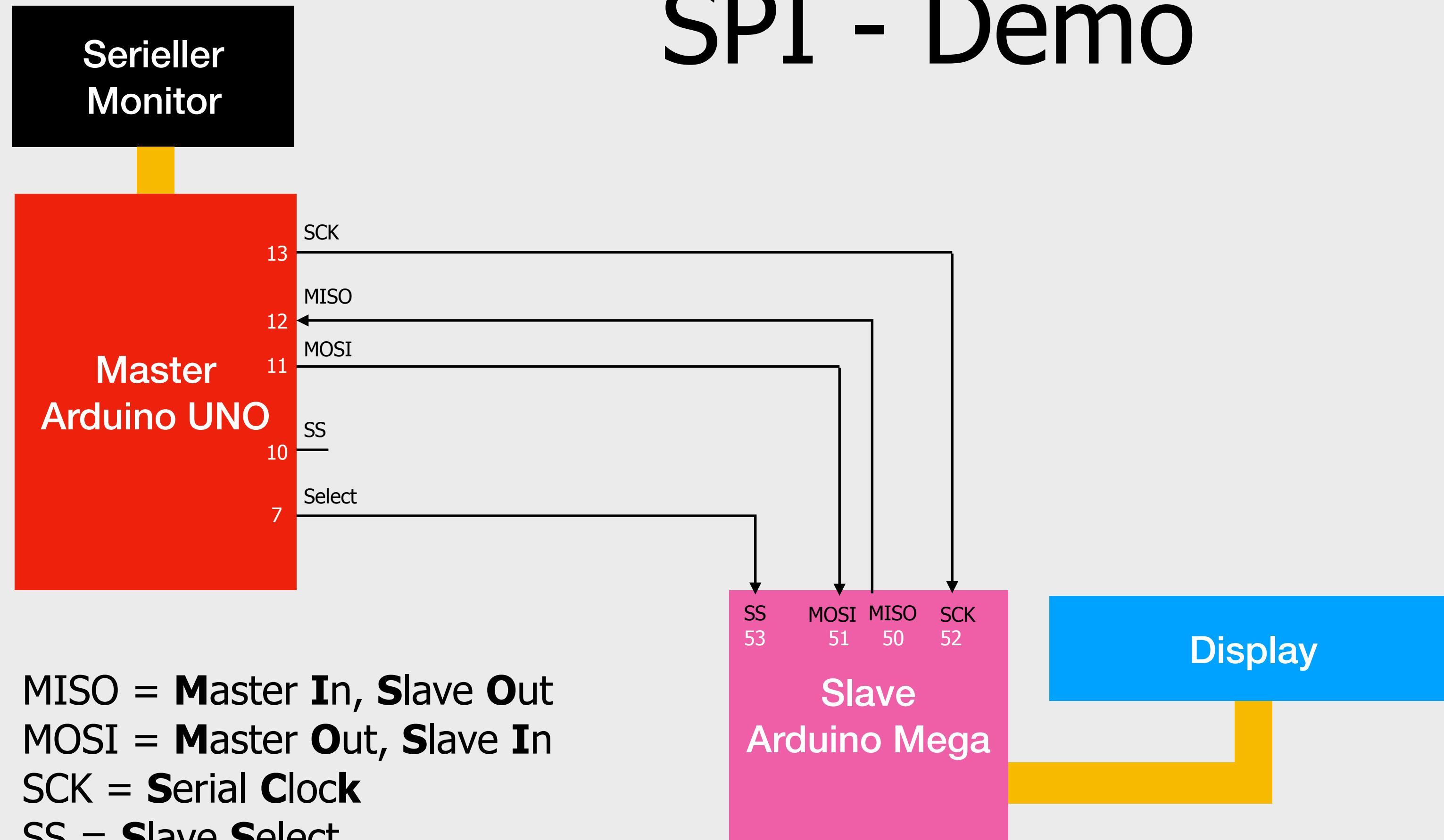


# SPI - Demo



# Das SPI-Protokoll im Detail

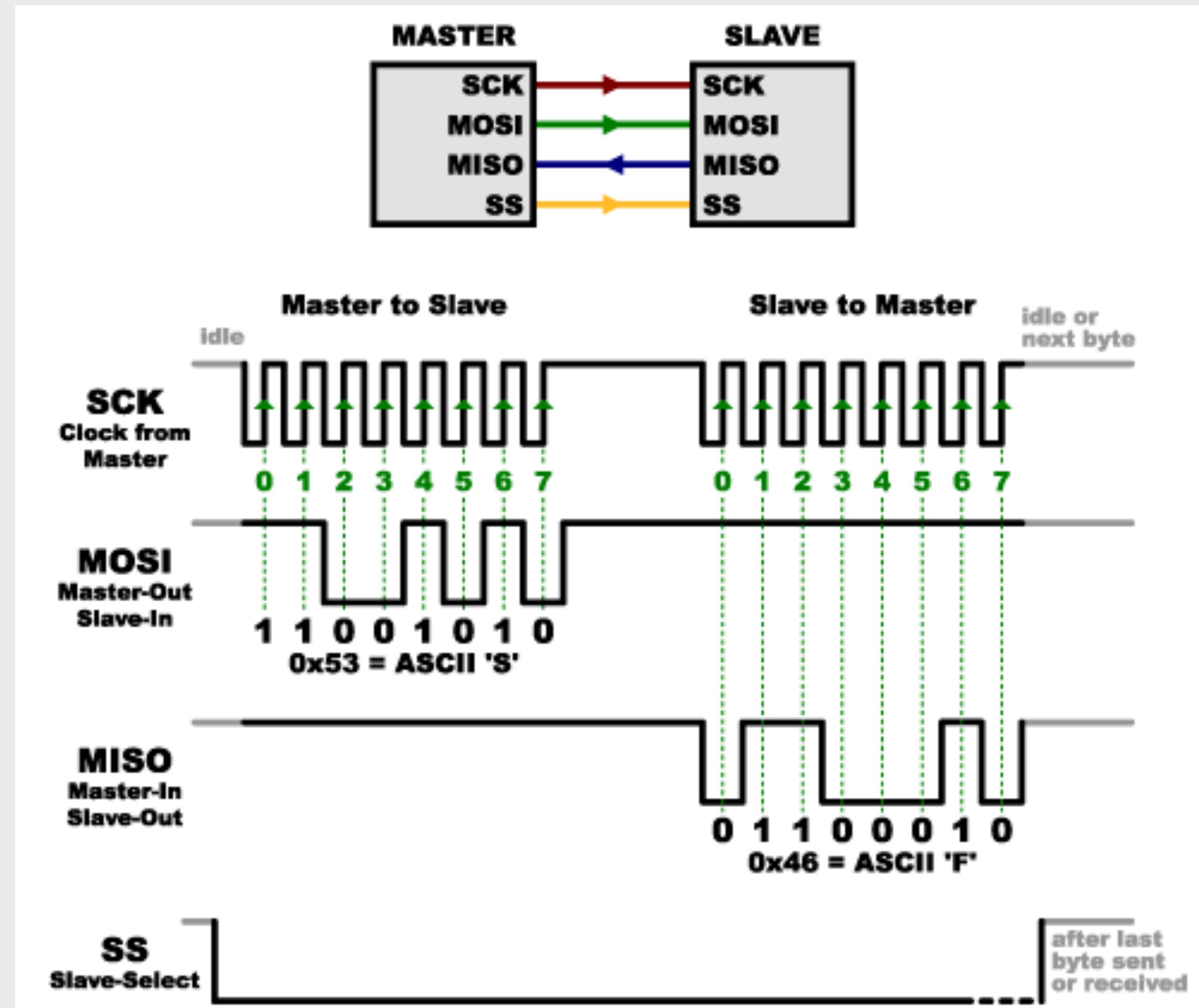


Diagramm aus dem SPI - Tutorial von Sparkfun (<https://learn.sparkfun.com/tutorials/serial-peripheral-interface-spi>)

# Umlaute auf dem LCD - Display

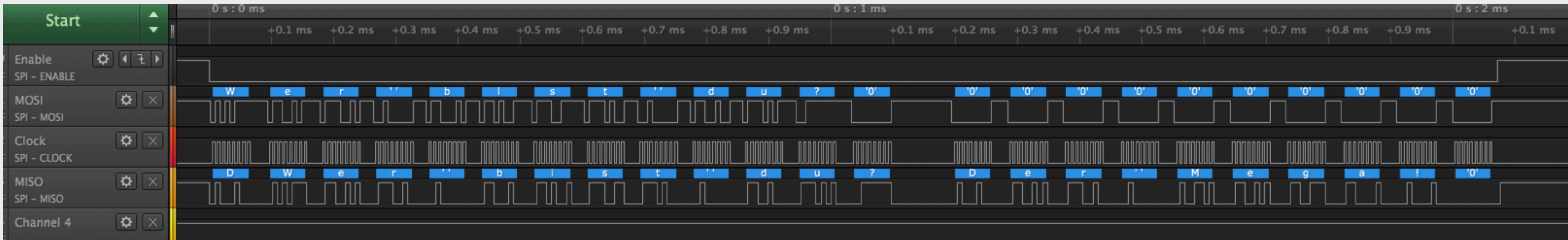
Umlaute sind in den meisten LCD - Displays vorhanden, haben aber einen etwas speziellen Code.

Zeichen	Hexadezimal	Oktal
ä	E1	341
ö	EF	357
ü	F5	365
° (Grad, z.B. °C) <small>Das Zeichen ist allerdings nicht schön. Es wird als kleines Quadrat dargestellt.</small>	DF	337
µ	E4	344
Ω	F4	364

Z. Bsp.: `Lcd.print("Temperatur: 23.5 \xF0C");` oder `Lcd.print("Temperatur: 23.5 \x337C");`  
schreibt Temperatur: 23.5 °C  
Oder in unserem Experiment: `Lcd.print("Der Mega h\xEFrt..")` schreibt 'Der Mega hört..'

Ein grosser Teil der gängigen LCD - Displays funktioniert so.  
Grossbuchstaben stehen aber leider nicht zur Verfügung.

# Übertragungsbeispiel



Der Master fragt: Wer bist du? (MOSI)

Der Slave antwortet: Der Mega! (MISO)

Der Master sendet '0' um zu signalisieren, dass seine Übertragung beendet ist und er auf eine Antwort wartet. Während der Antwort sendet er weiterhin nach jedem erhaltenen Zeichen '0', bis der Slave seine Antwort beendet hat.

Sobald der Slave eine '0' erhält, sendet er ein Zeichen der Antwort. Das macht er, bis die Antwort vollständig ist. Danach sendet er zum Abschluss eine '0'. Der Master weiss jetzt, dass die Antwort komplett ist und beendet die Kommunikation.

# Verschiedene Übertragungsarten

Mode: SPI_MODE0 bis SPI_MODE3	Polarität und Flanken der Signale
Clock Divider: SPI_CLOCK_DIV2 .. SPI_CLOCK_DIV128	Geschwindigkeit
Bitorder: LSBFIRST .. MSBFIRST	Reihenfolge der Bits

Einstellen mit

- setDataMode
- setClockDivider
- setBitOrder

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

Neu jetzt mit

```
SPI.beginTransaction(SPISettings(14000000, MSBFIRST, SPI_MODE0));
```

# Arduino als Slave

Der Slave - Mode wird von der SPI - Library nicht unterstützt!

Das ist kein grosses Problem, denn die Hardware kann diese Protokoll direkt unterstützen. In dem Beispiel auf <https://www.arduino.cc/en/Tutorial/SPIEEPROM> finden wir einige Informationen dazu.

Für uns sind zwei Register wichtig:

SPDR (SPI Data Register): Diese überträgt die Daten (1 Byte)

SPCR (SPI Control Register): Hier aktivieren wir den Slave - Mode

Wir müssen im SPCR - Register nur dir Bits 6 und 7 auf 1 setzen und schon liefert uns der Prozessor die Daten an die Interrupt Service Routine ISR (SPI\_STC\_vect).