

# **STM32 course**

**Preparing for hackathon**

# Going back to USART for a little while

- Synchronous mode
- Half-duplex and full-duplex modes
- Parity bits
- DMA support
- Noise detection
- Programmable transfer data size
- Modem flow control

# Type of communication service



**Simplex**

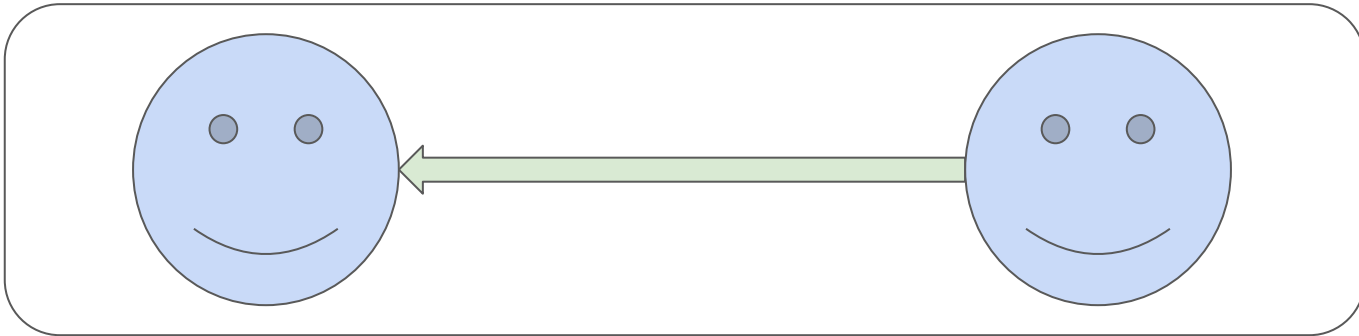
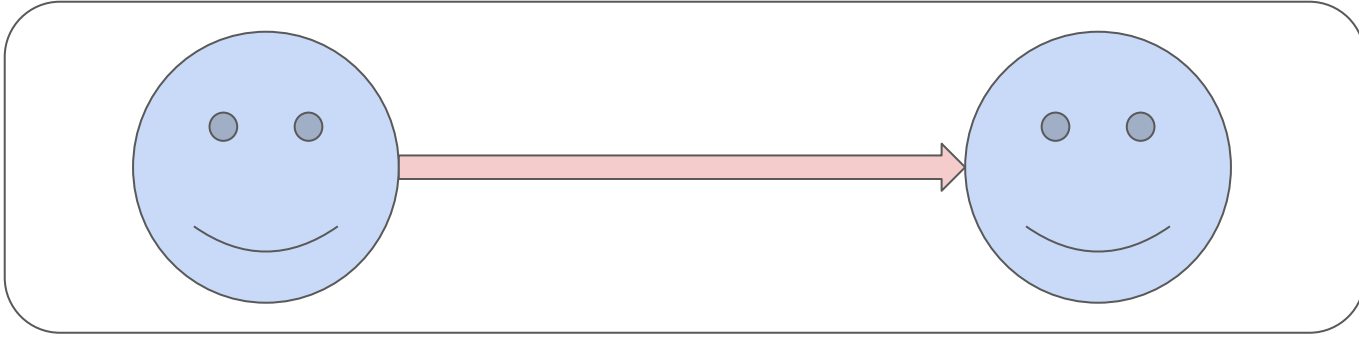
**Duplex**

**Full-duplex**

**Half-duplex**

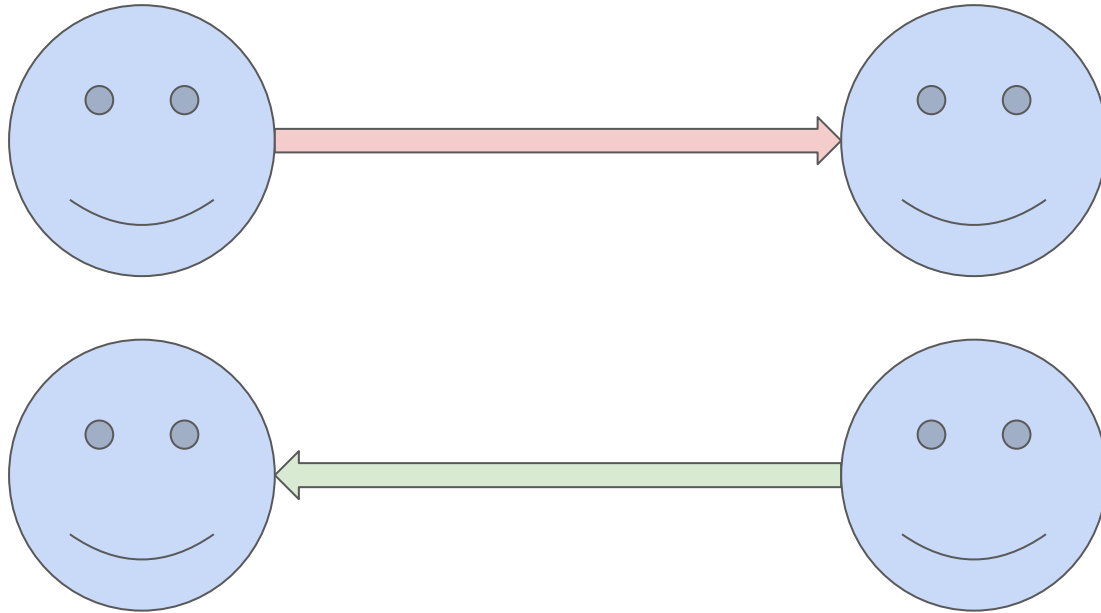
# Simplex communication

**OR**

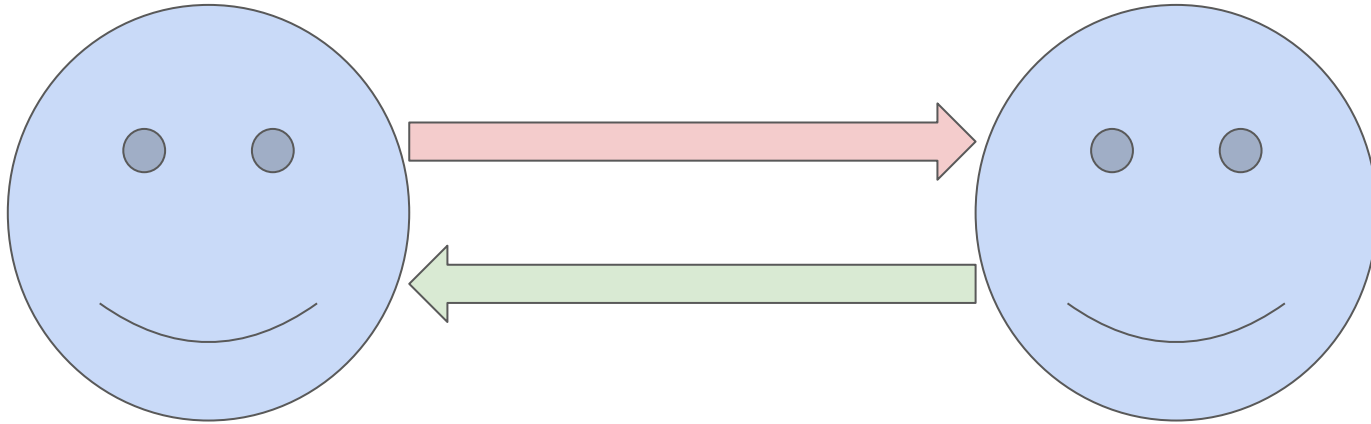


# Half-duplex communication

**OR**



# Full-duplex communication



# Killer-feature of USART

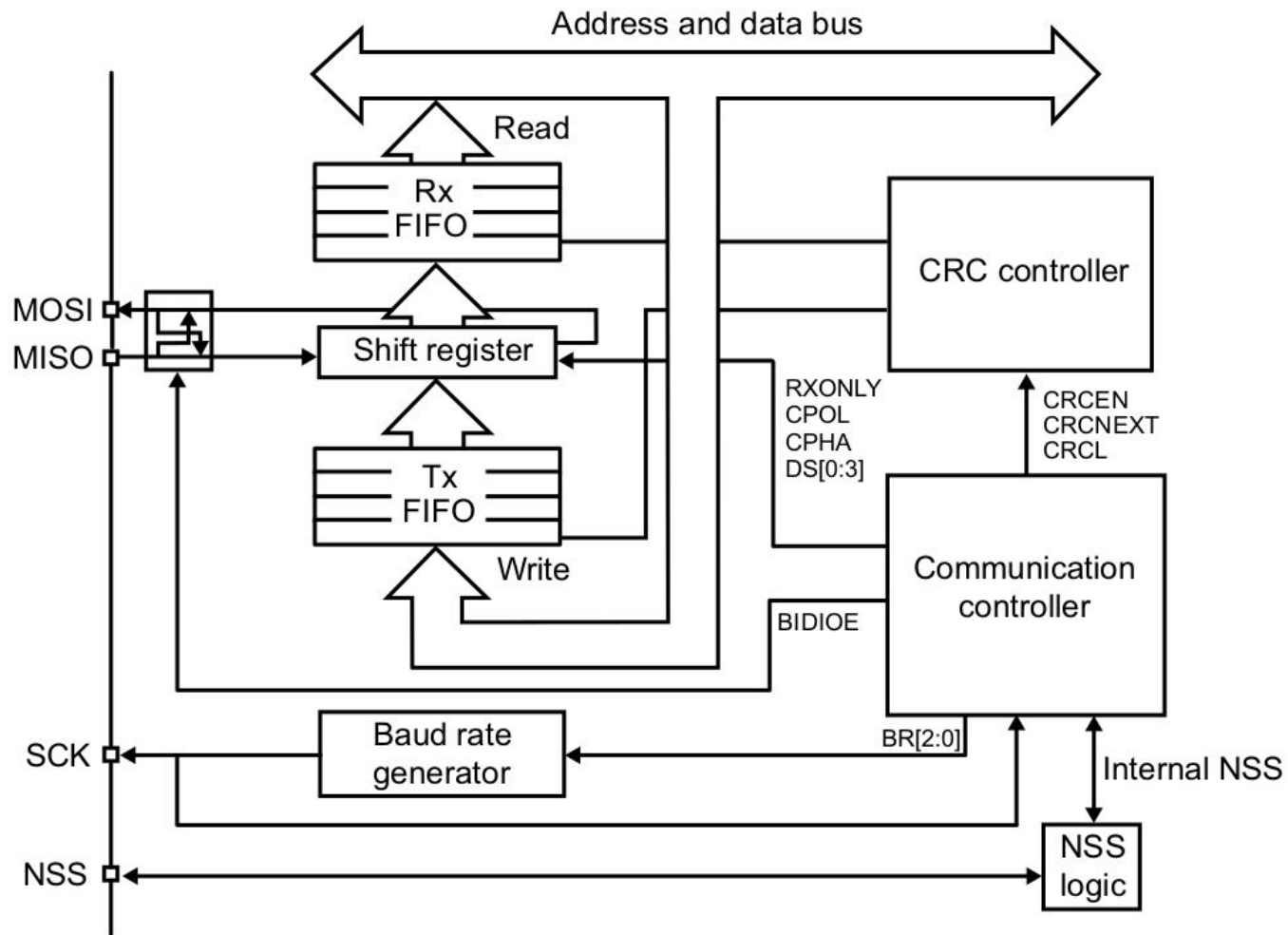
**Debug might become easier and PC-like**

*But we give no warranty*

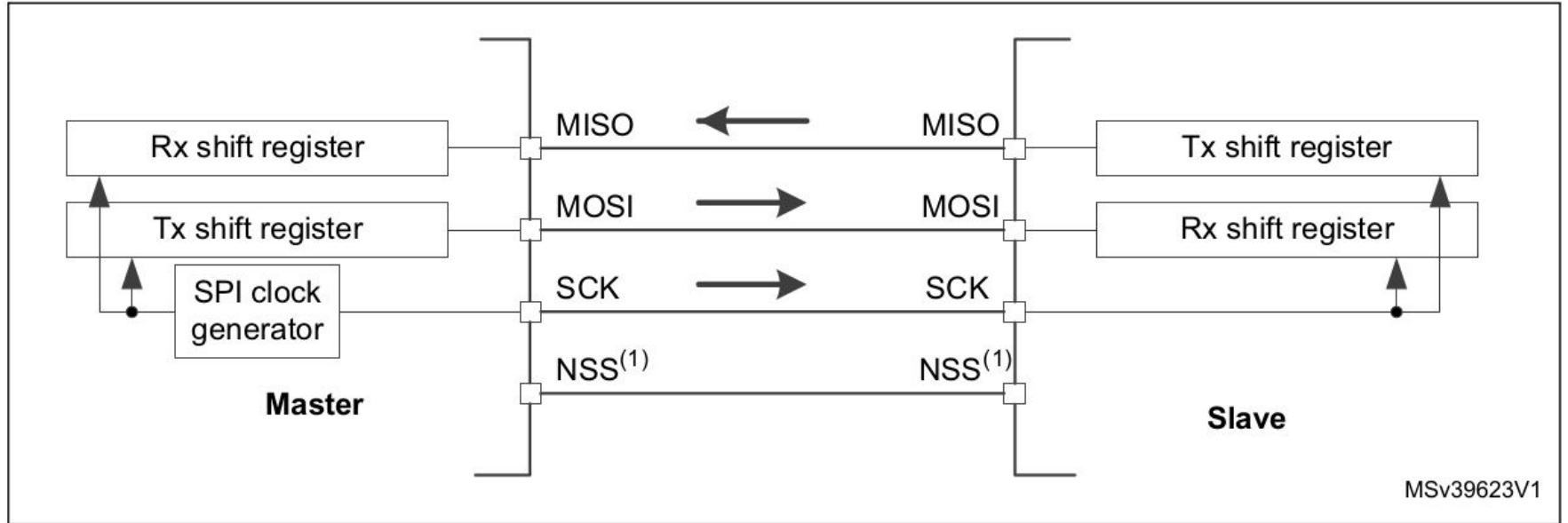
# **SPI. Serial peripheral interface**

- Master or slave operation
- Full-duplex, half-duplex and simplex
- 4-bit to 16-bit data size selection
- Programmable clock polarity and phase
- NSS management by hardware or software

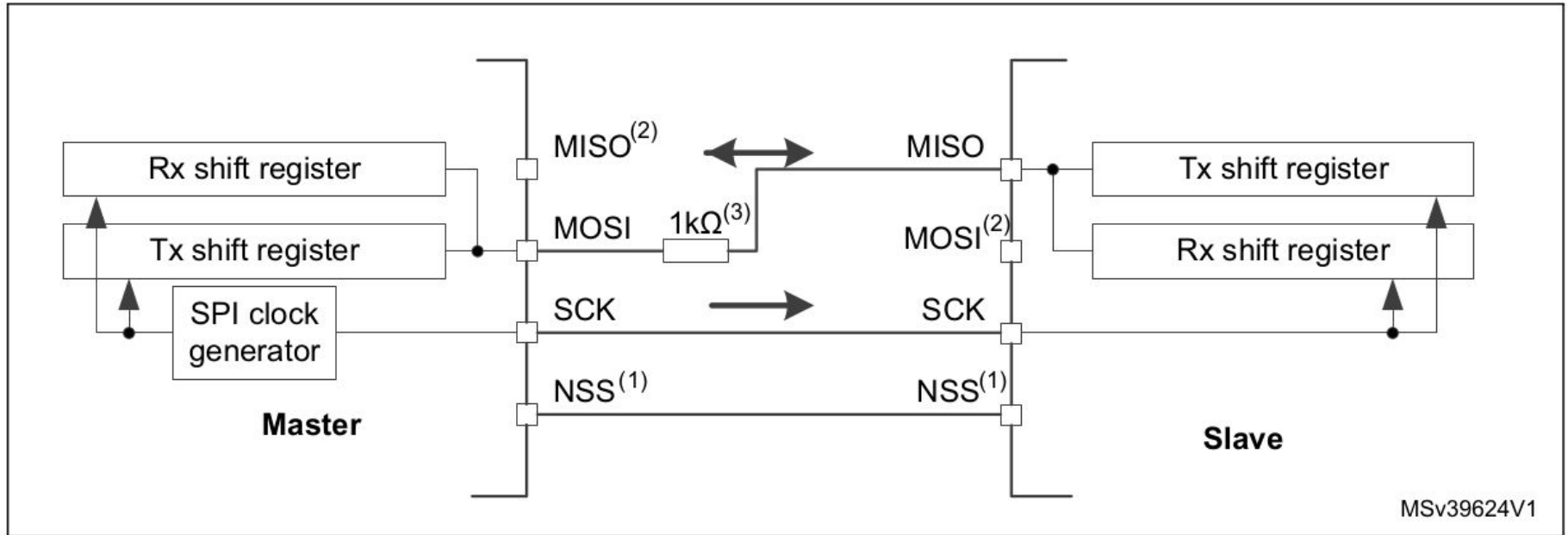




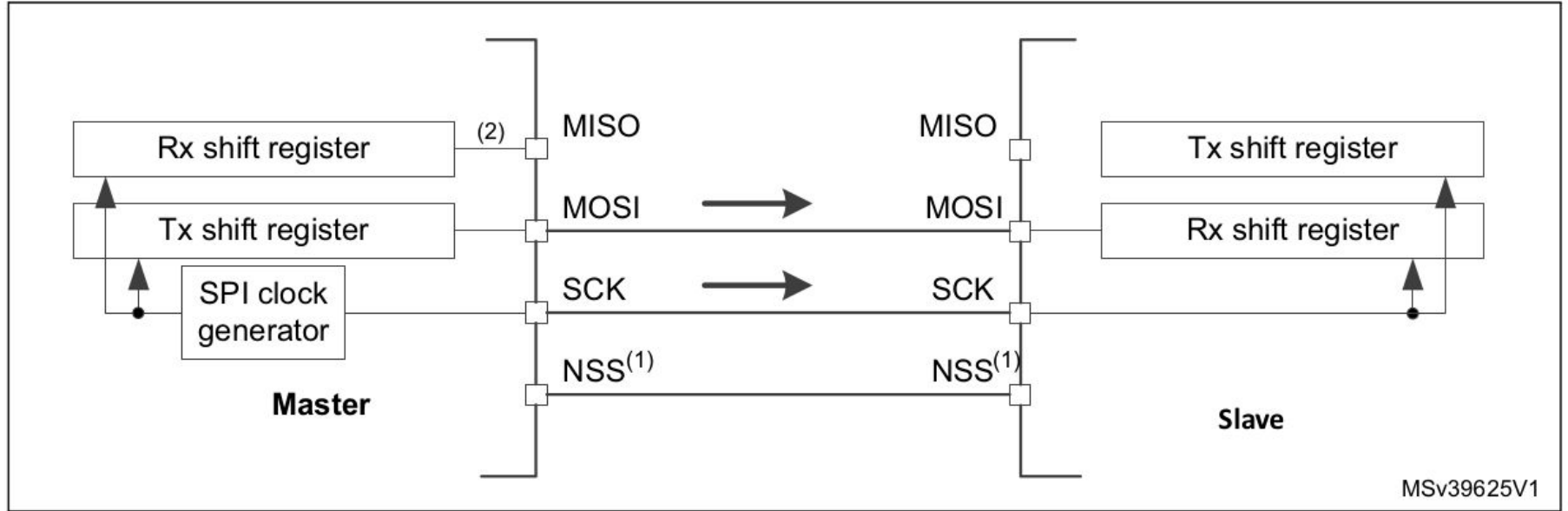
# SPI. Full-duplex



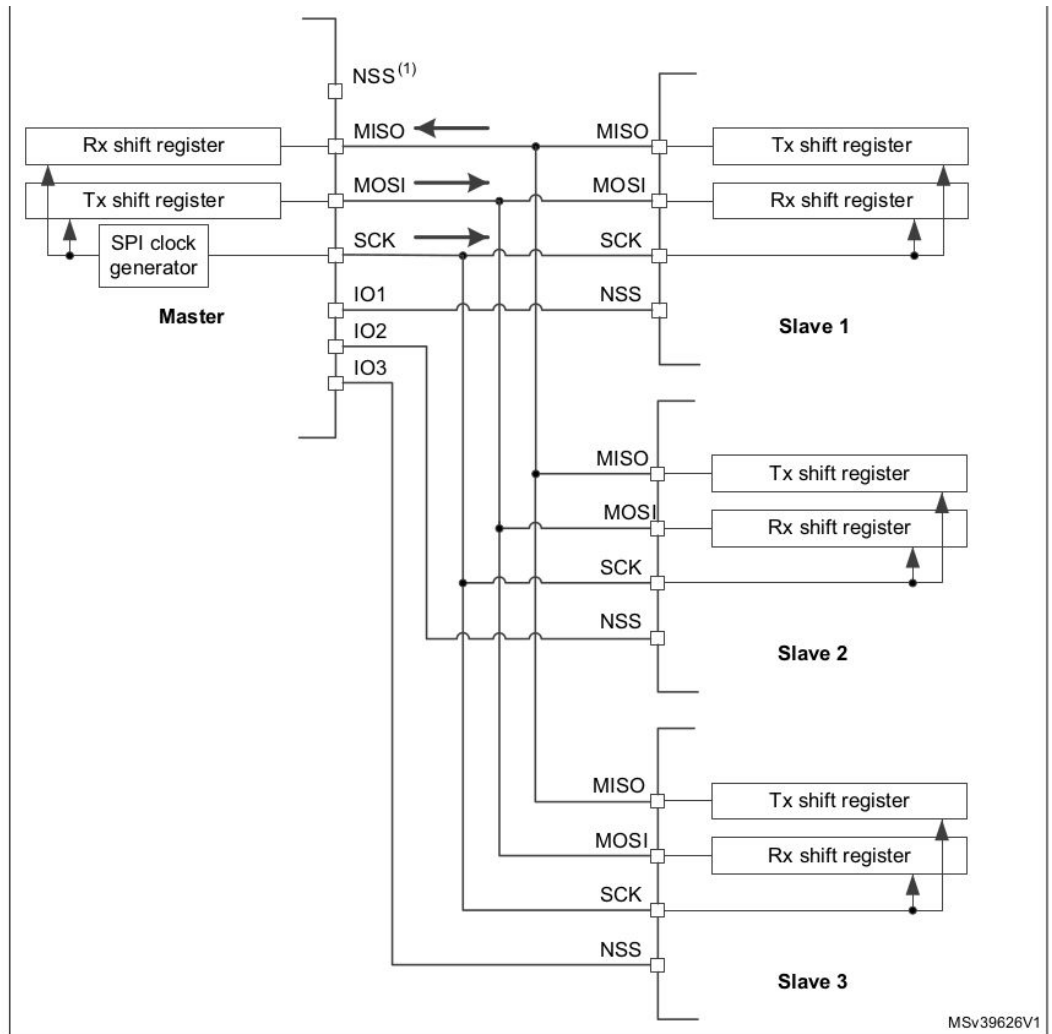
# SPI. Half-duplex



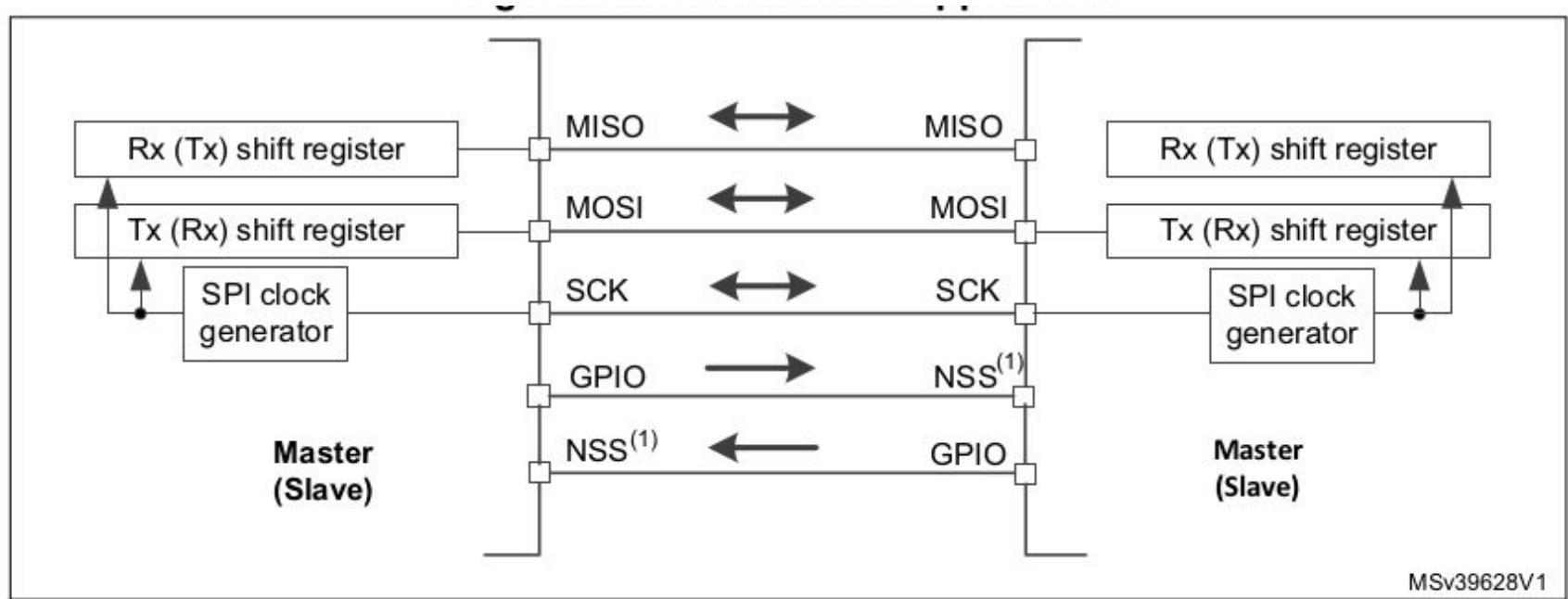
# SPI. Simplex



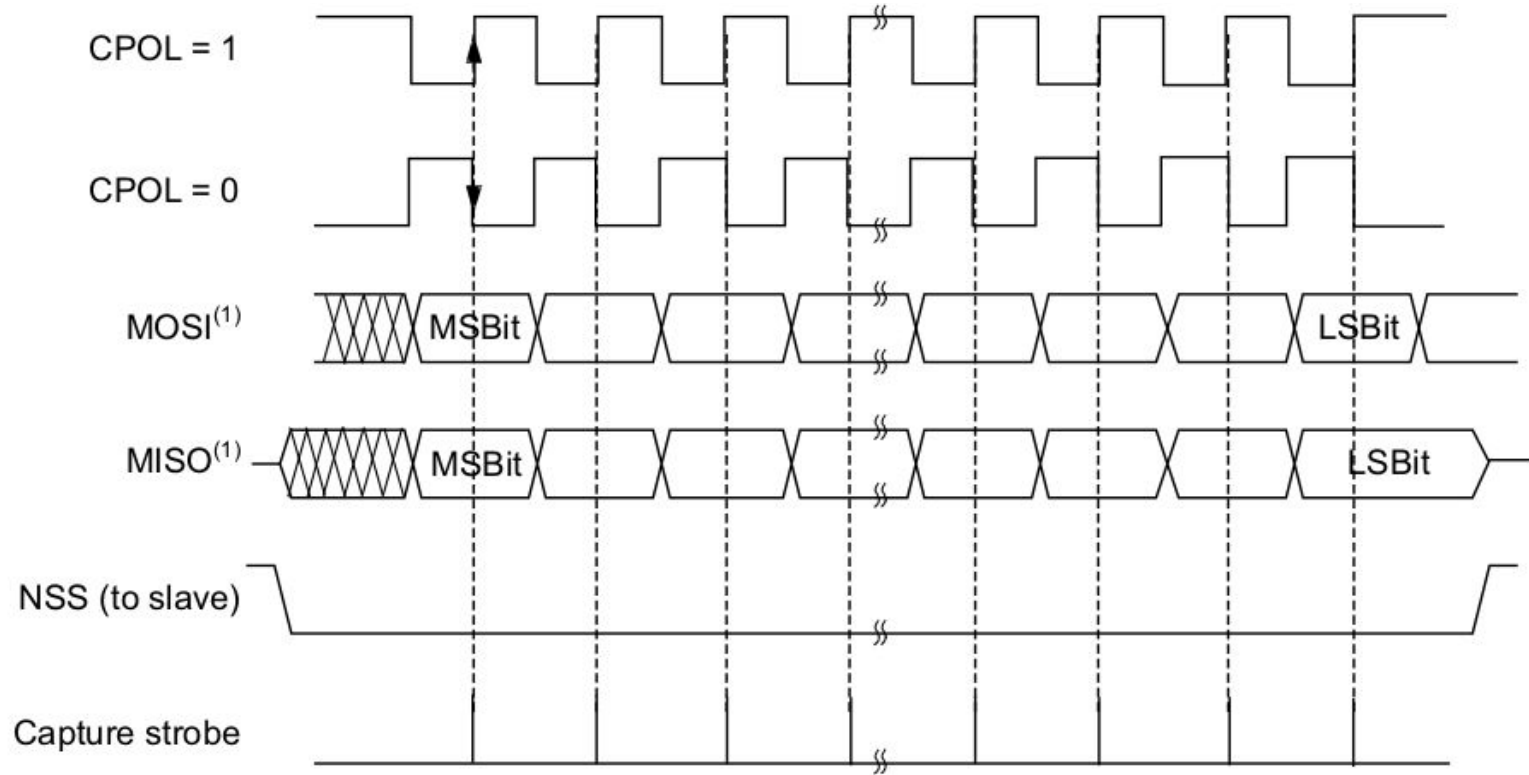
# SPI. Multi slave mode



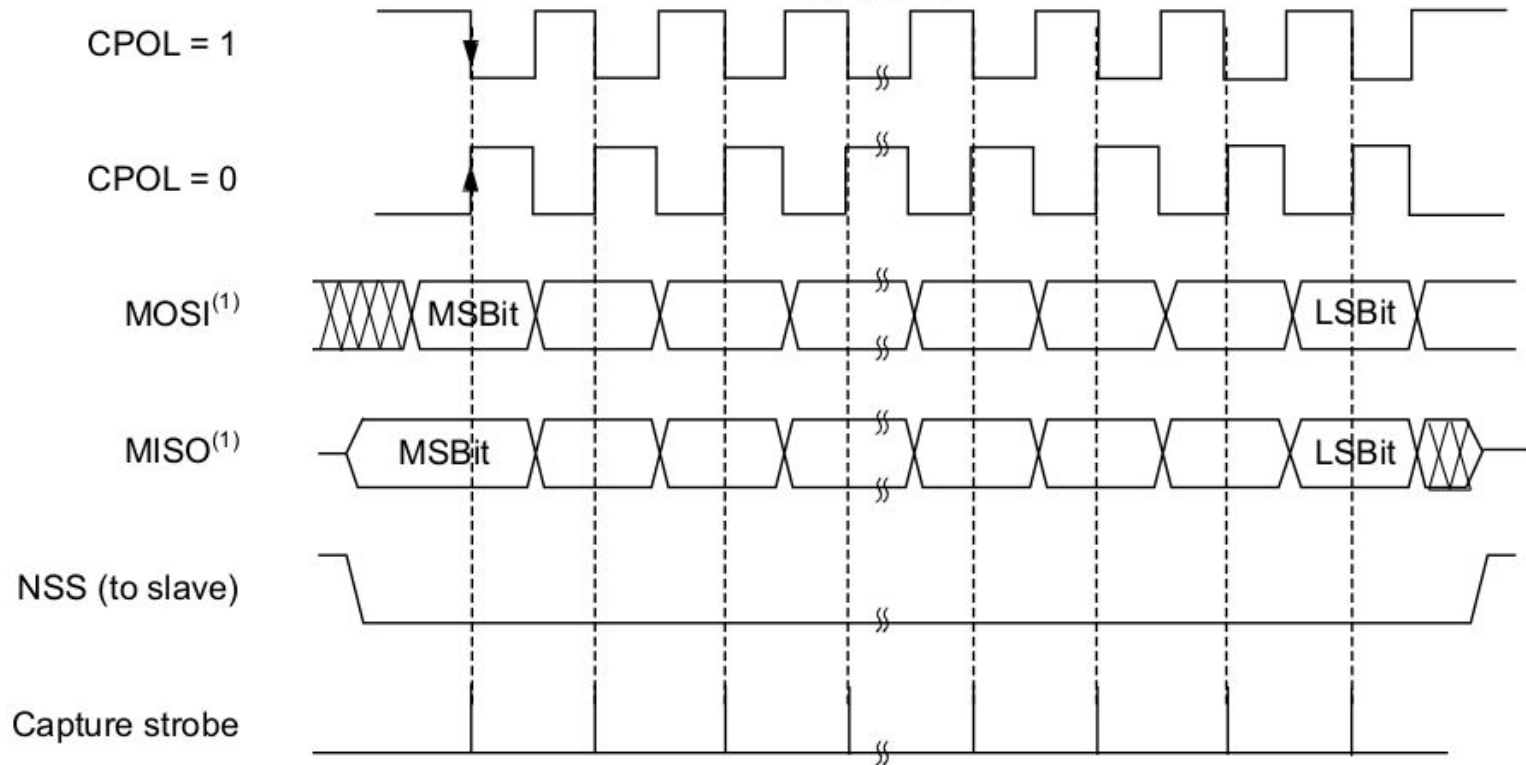
# SPI. Multi master mode



# SPI. CPHA (clock phase) = 1

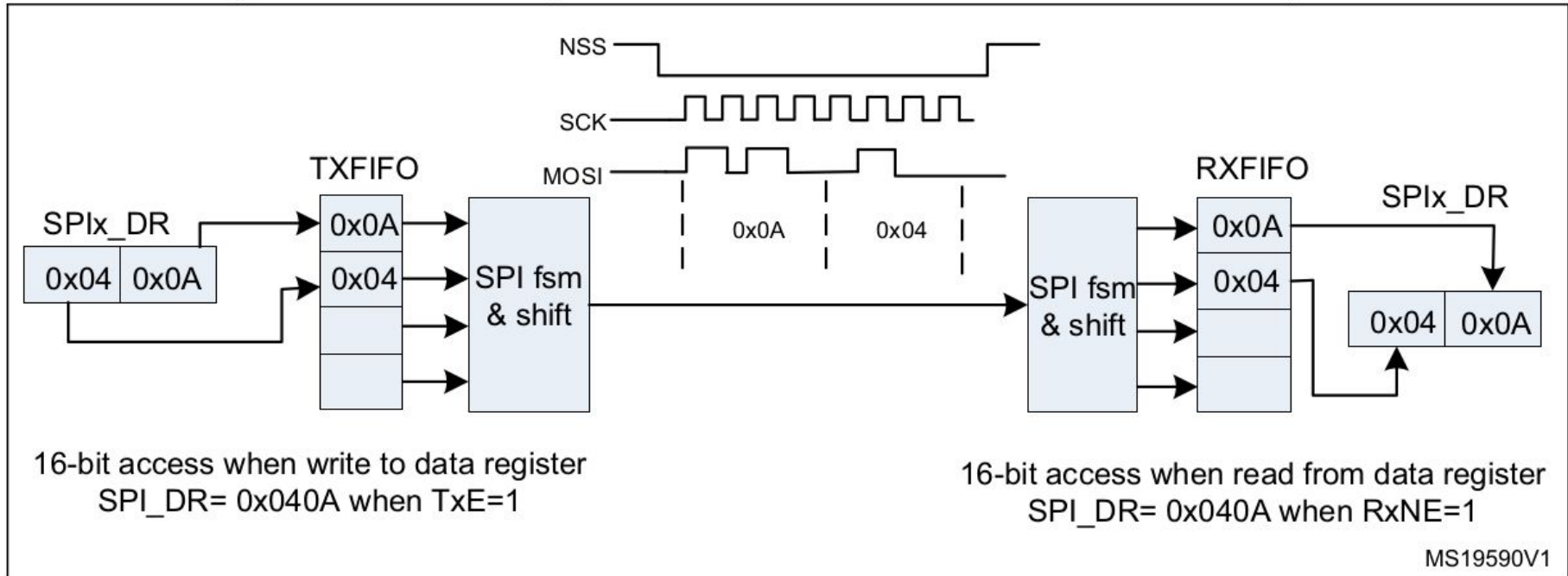


# SPI. CPHA (clock phase) = 0





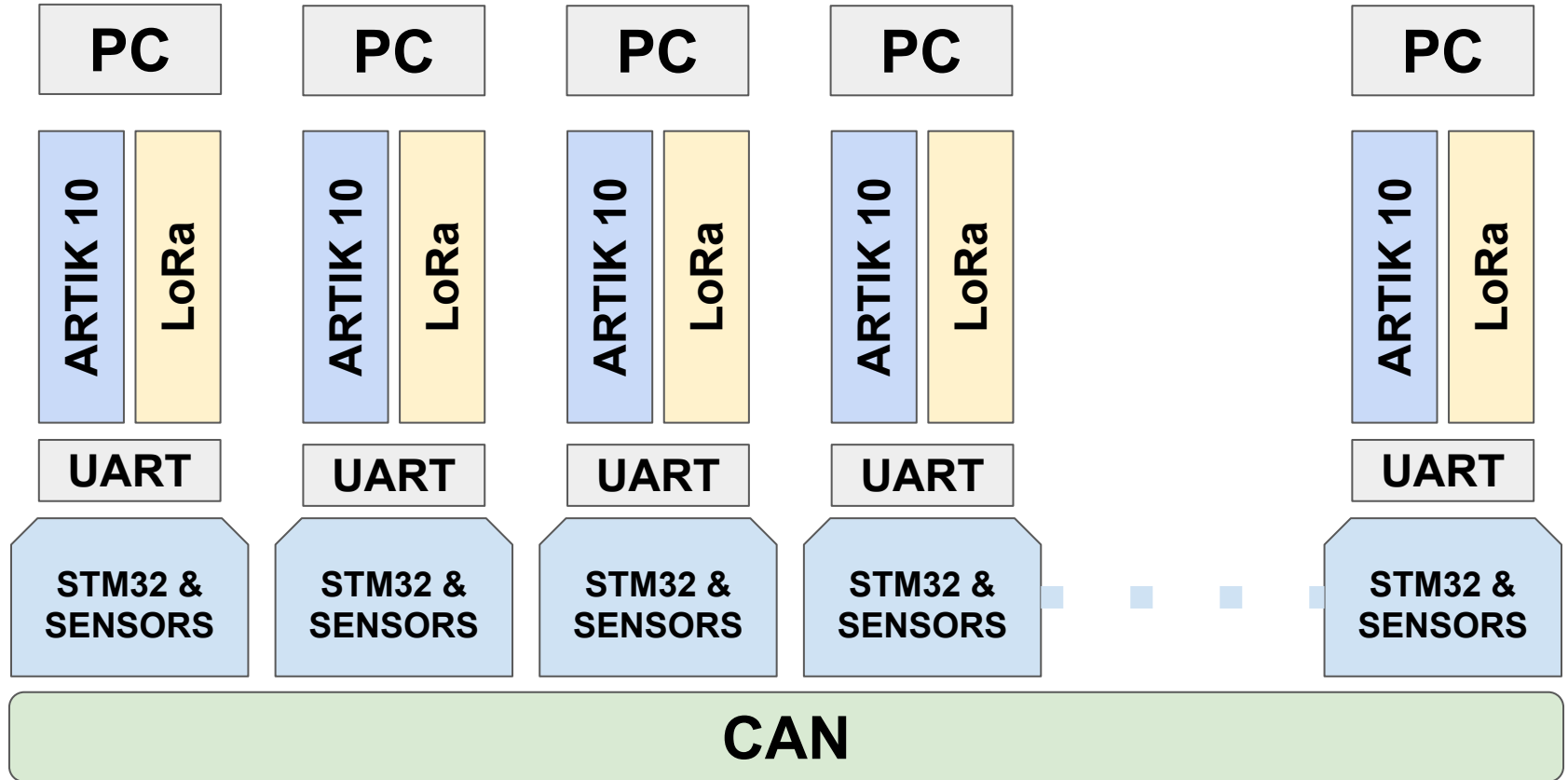
# SPI. RX and TX FIFOs



# SPI. Example

*Switch to example!*

# Hackathon. Diagram



# CAN Underlayer

**GitHub link:** [https://github.com/edosedgar/CAN\\_underlayer](https://github.com/edosedgar/CAN_underlayer)

## **Main sources:**

- main.c
- can\_core.c
- can\_api.c
- can\_callbacks.c

# CAN underlayer. Main idea

