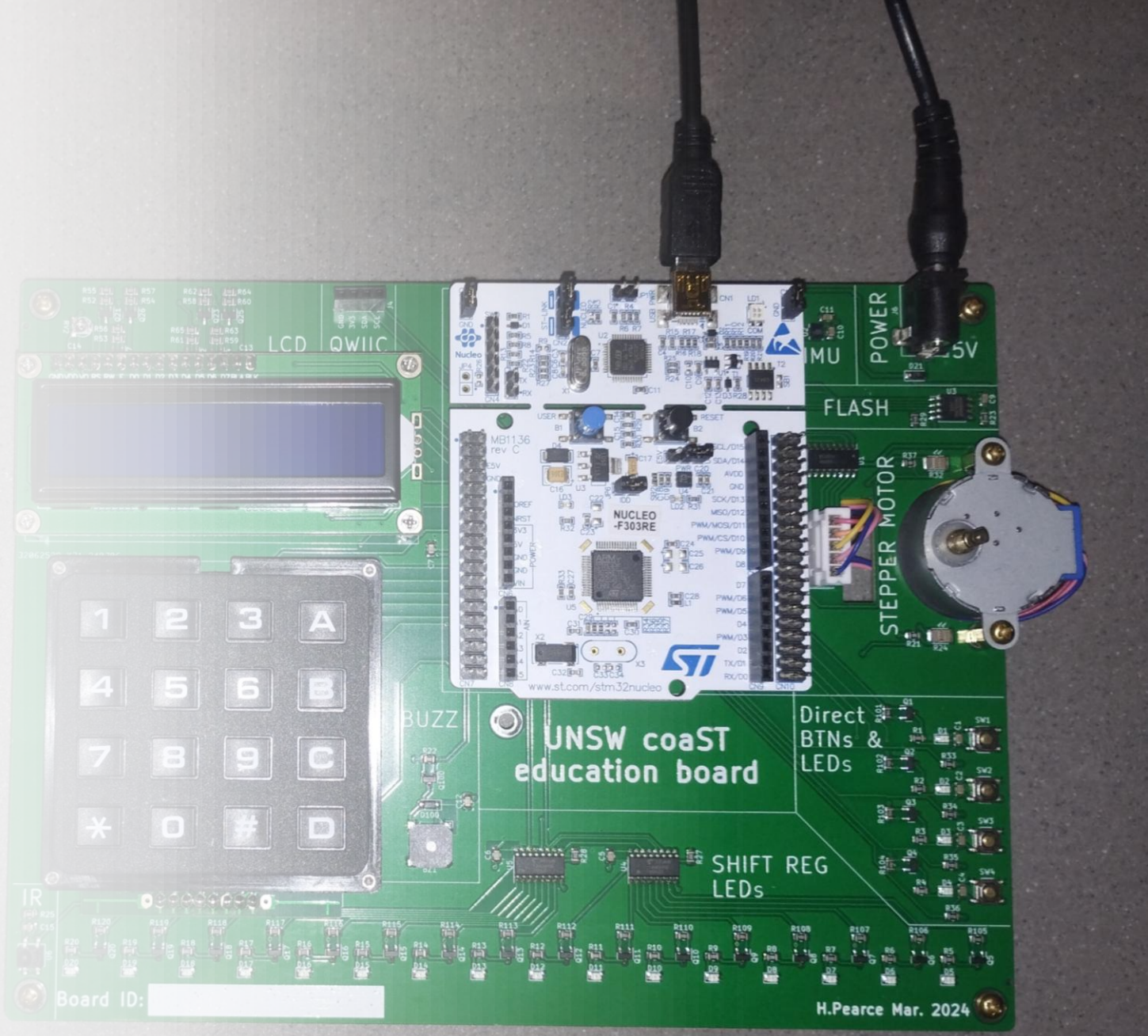
DESN2000 (Computer Engineering) 2025 T2

Serial Communication- SPI and I2C

Hasindu Gamaarachchi



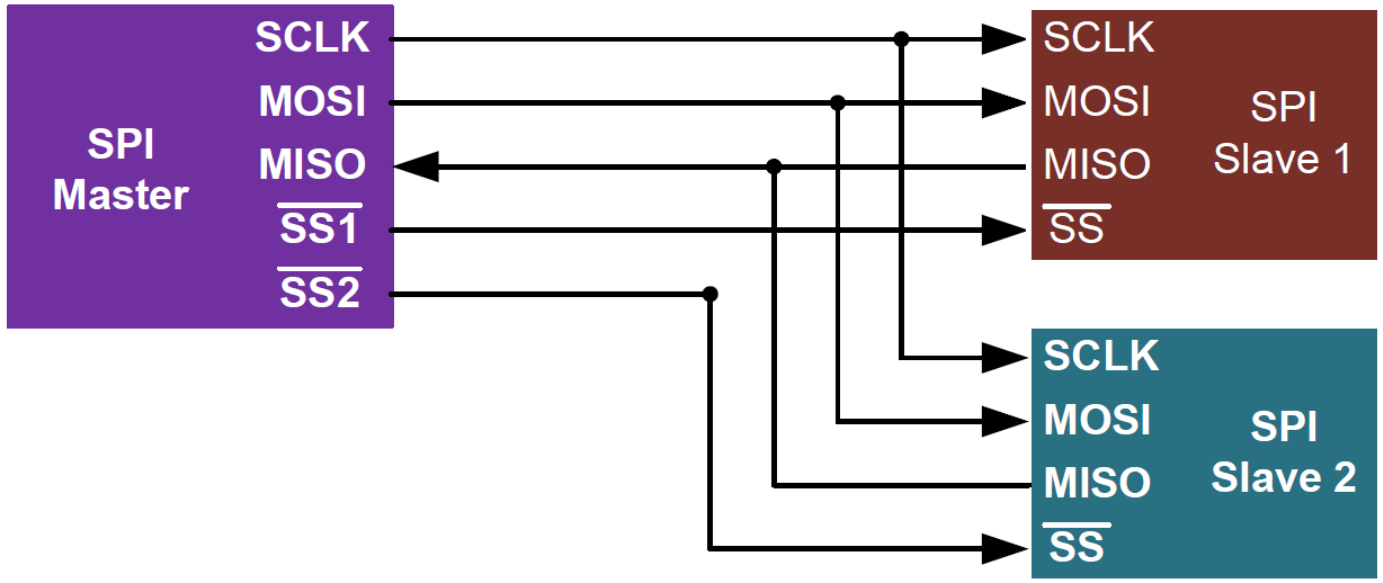
Learning Resources

The upcoming slides are adapted from “Embedded Systems with ARM Cortex-M Microcontrollers in Assembly Language and C (Fourth Edition)” – Yifeng Zhu

- Serial communication protocols- chapter 19

Serial Peripheral Interface (SPI)

- Synchronous full-duplex communication
- Higher throughput
- Single master, multiple slaves
- Slave cannot communicate with slave directly



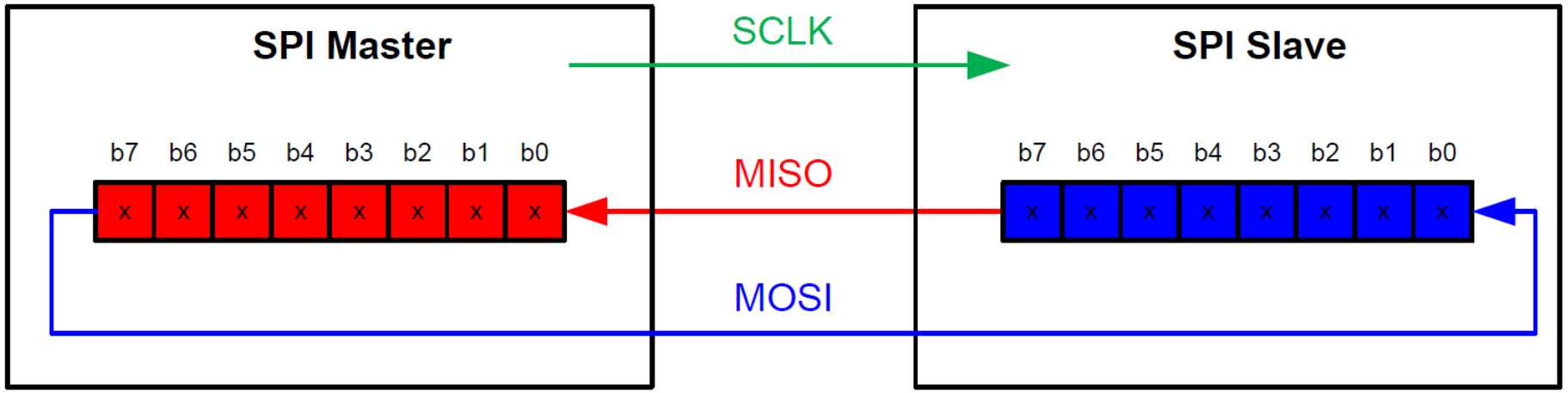
SCLK: serial clock

SS: slave select (active low)

MOSI: master out slave in

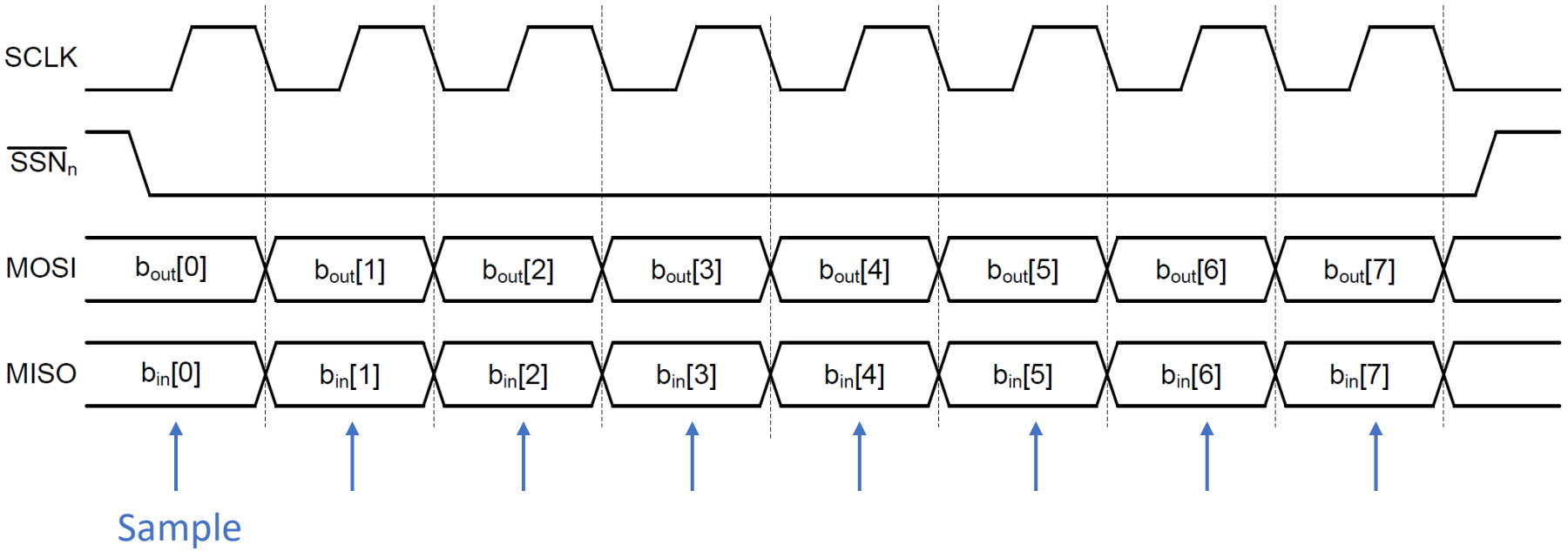
MISO: master in slave out

SPI Synchronous Data Exchange

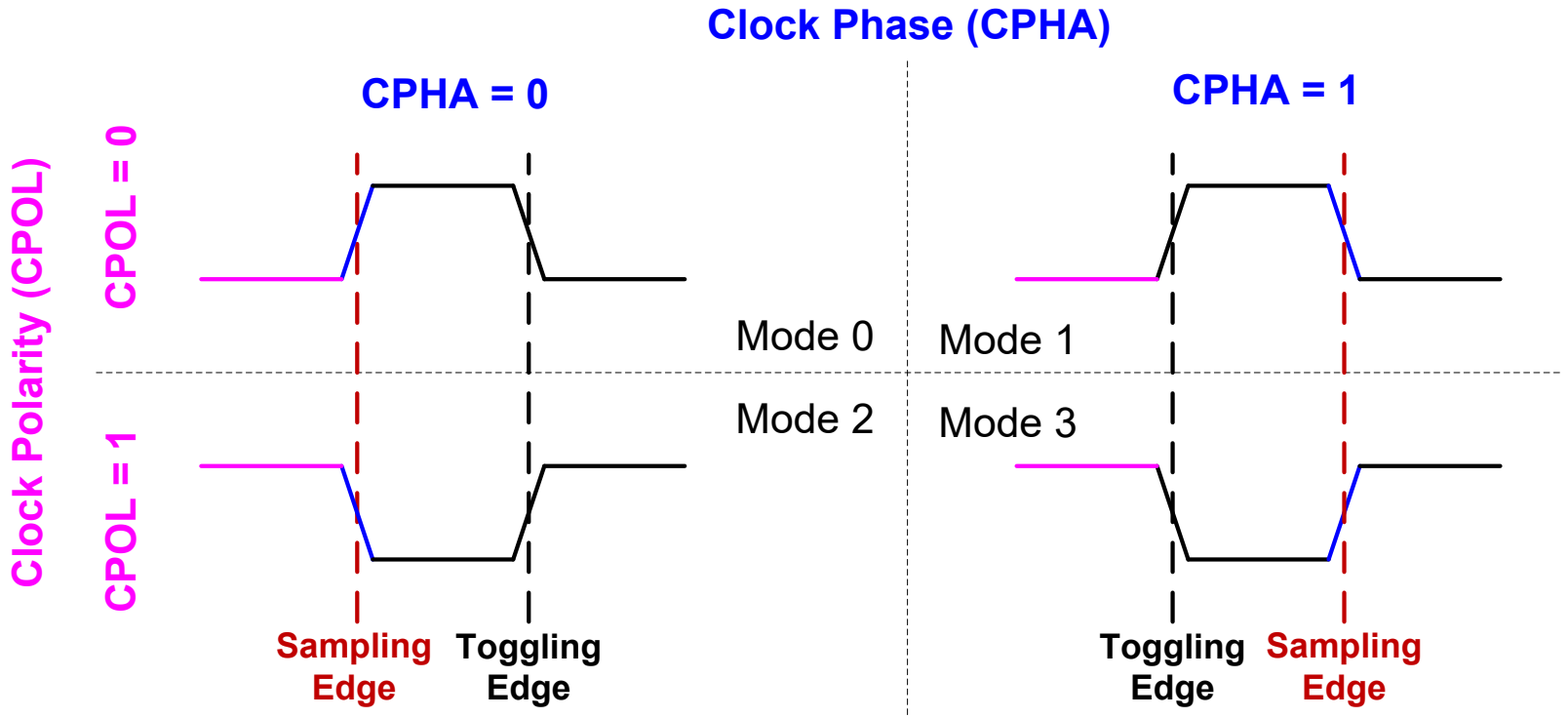


- Master has to provide clock to slave
- *Synchronous exchange*
  - Master shift out a bit to slave, and shifts in a bit from slave.
- Only master can start the data transfer.

SPI Timing Diagram

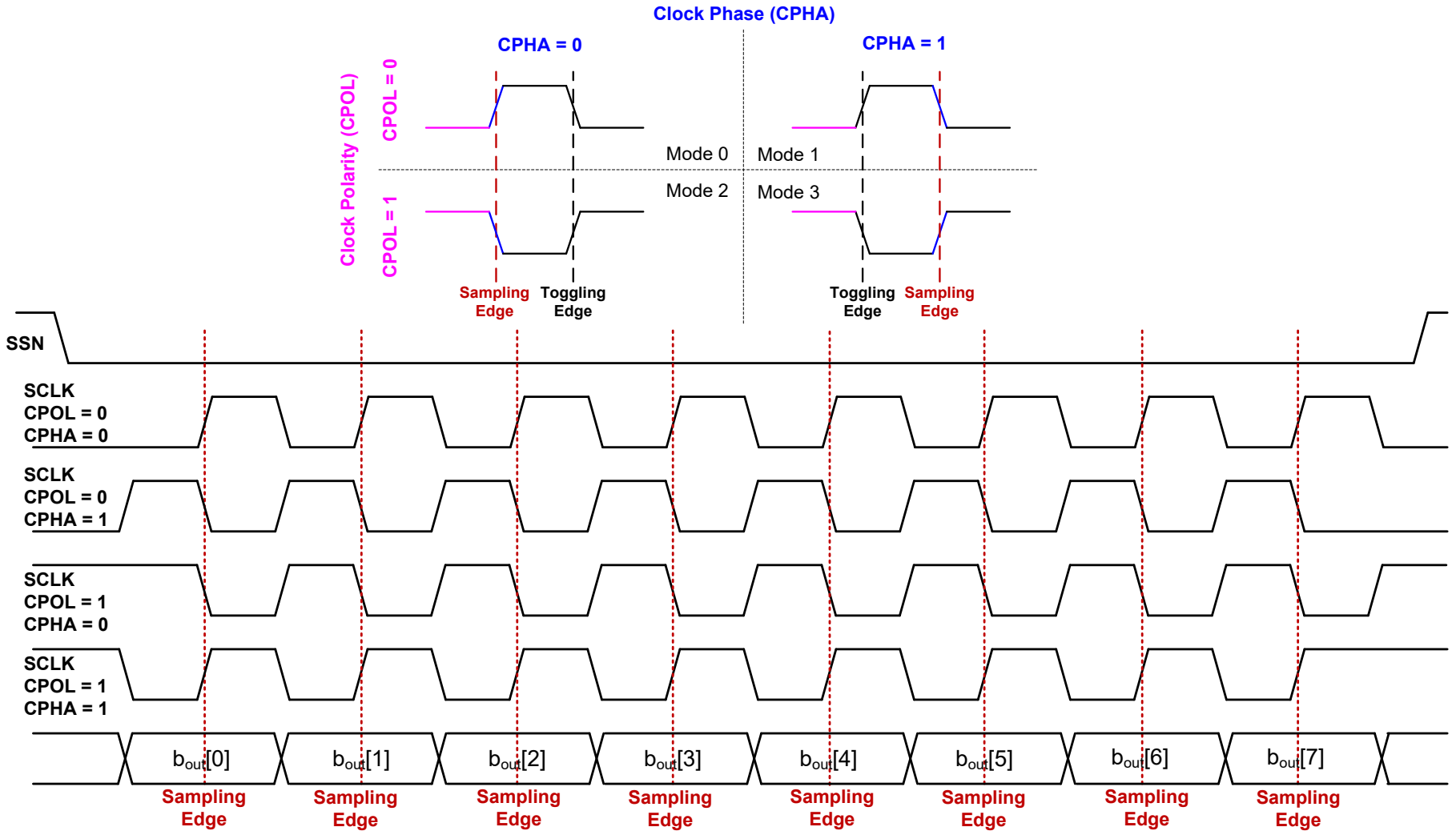


SPI Clock Phase and Polarity

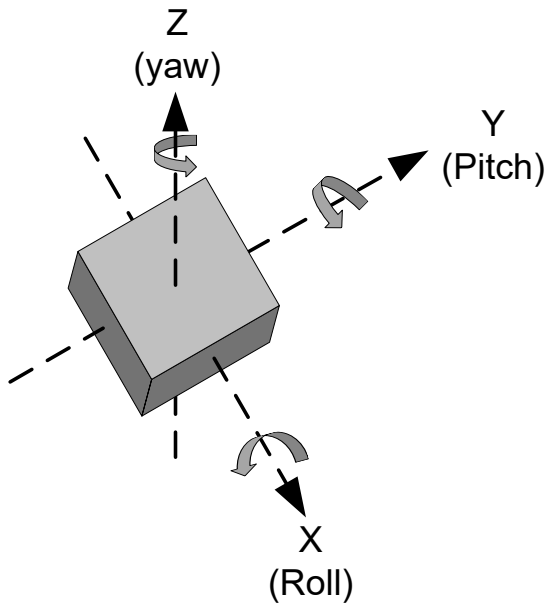


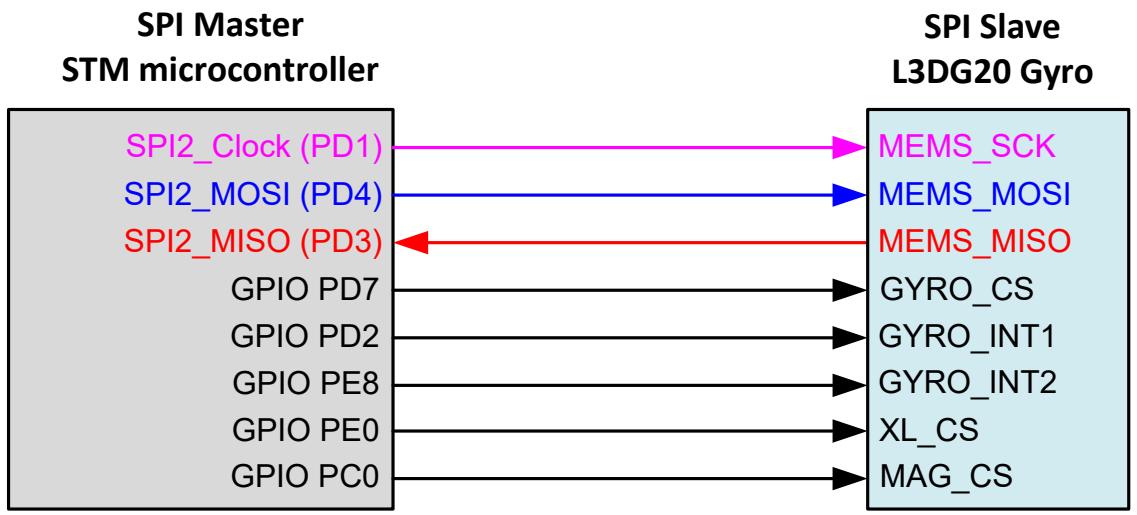
- Combination of CPOL and CPHA determines the clock edge for transmitting and receiving.
- CPOL = 0 ⟶ SCLK is pushed to low during idle. Otherwise, pulled to high during idle.
- CPHA = 0 ⟶ the first clock transition (either rising or falling) is the first data capture edge. Otherwise, the second clock transition is the first data capture edge.

SPI Clock Phase and Polarity

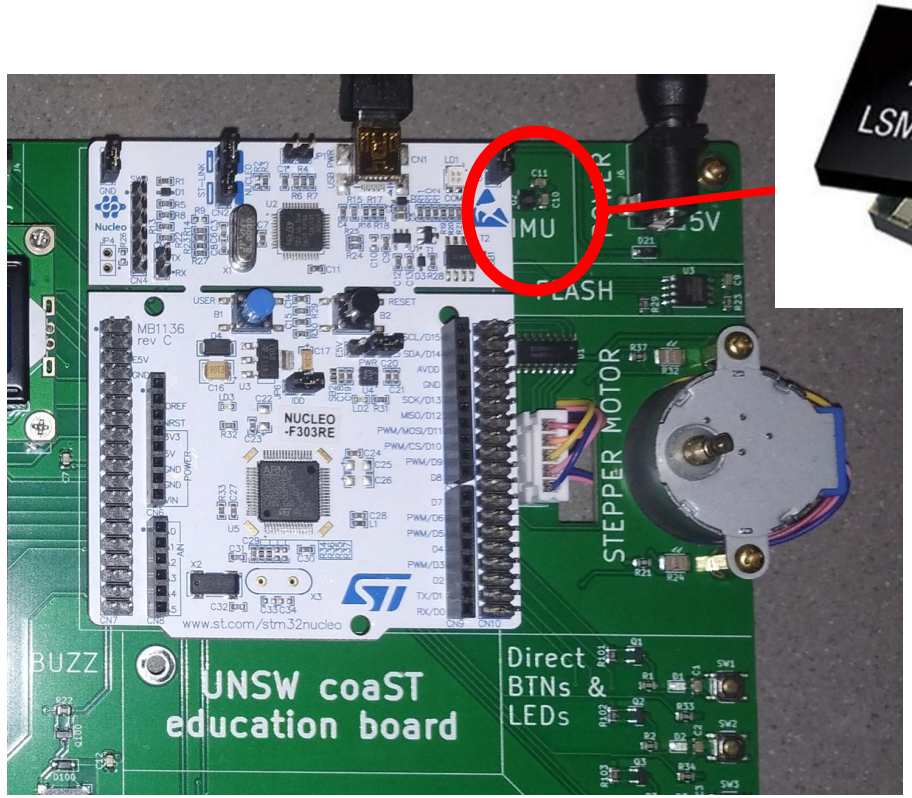


Gyro Sensors





On coaST Board – STLSM6DS3



Accelerometer	Microcontroller PIN
ACC_SCK	PA5
ACC_MISO	PA6
ACC_MOSI	PA7
ACC_CS	PC12

Tutorials for HAL

- https://wiki.st.com/stm32mcu/wiki/Getting_started_with_SPI
- https://01001000.xyz/2020-08-09-Tutorial-STM32CubeIDE-SD-card/

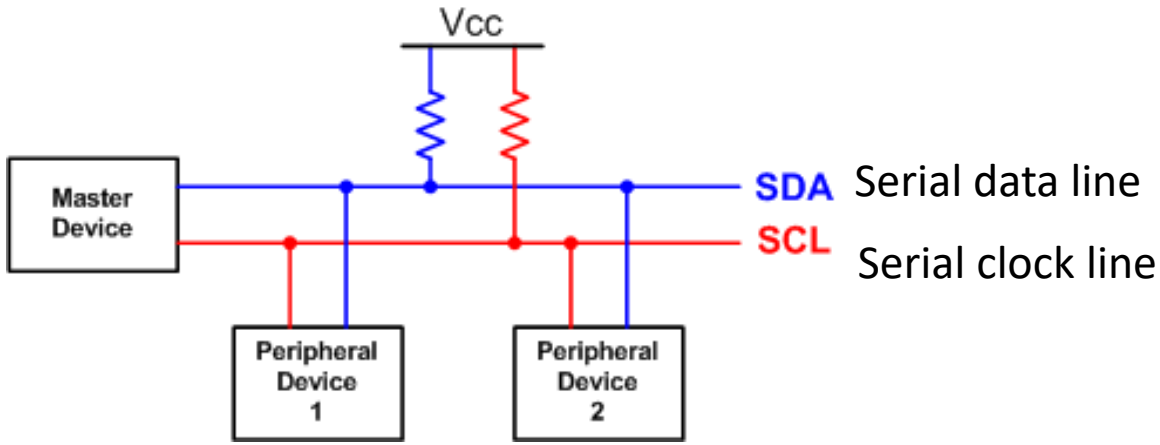
Inter-Integrated Circuit (I2C)

- Designed for low-cost, medium data rate applications by Philips in the early 1980's
  - Original purpose: connect a CPU to peripheral chips in a TV-set
  - Today: a de-facto standard for 2-wire communications
  - Since October 10, 2006, no licensing fees are required to implement the I^2C protocol.
  - However, fees are still required to obtain I^2C slave addresses allocated by NXP (acquired Philips).

Inter-Integrated Circuit (I2C)

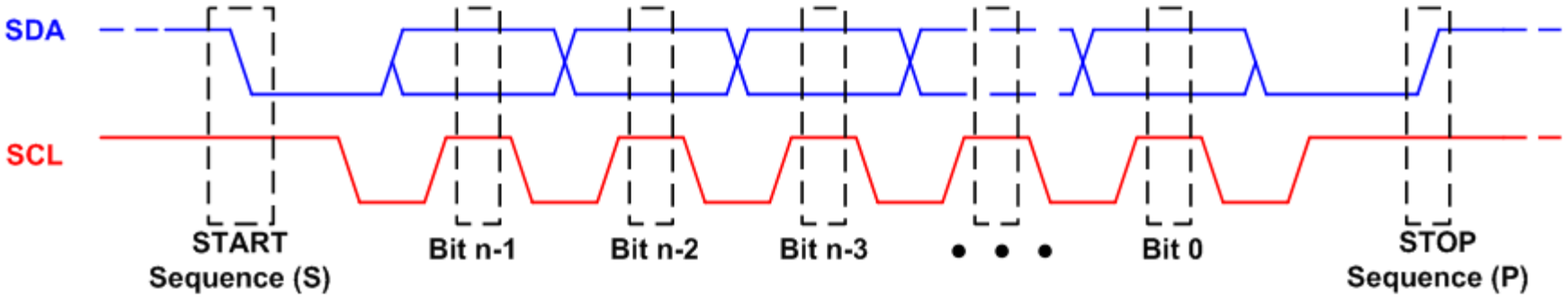
- Characteristics
  - Serial
  - Multi-master, multi-slave
  - Each Device has a unique address (7, 10 or 16 bits)
    - Address 0 used for broadcast
  - Two bidirectional lines, plus ground
    - Serial Data Line (SDA)
    - Serial Clock Line (SCL)
  - Up to 100 kbit/s in the standard mode, up to 400 kbit/s in the fast mode, and up to 3.4 Mbit/s in the high-speed mode.

Inter-Integrated Circuit (I2C)



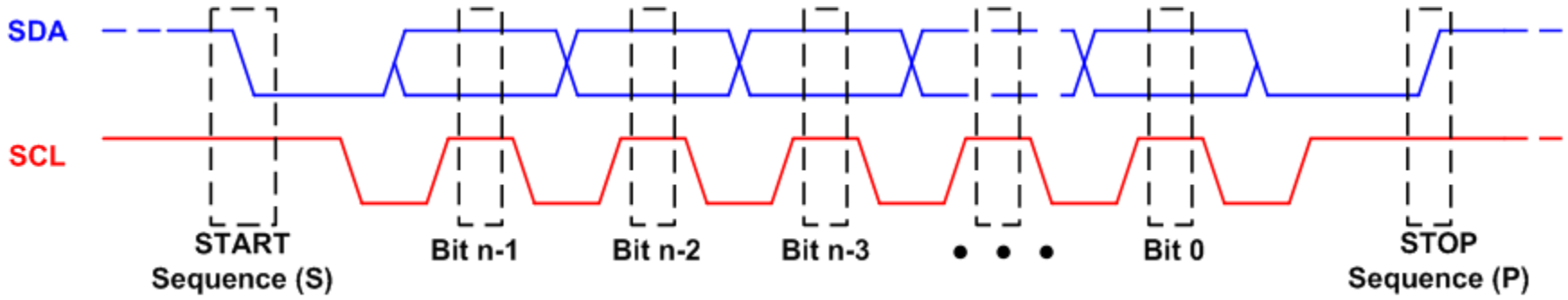
- SDA and SCL need to pull up with resistors
- SDA and SCL have to be open-drain
- STM32's internal pull-up is too weak (internal 100KΩ)
- External pull-up (4.7 kΩ for low speed, 3 kΩ for standard mode, and 1 kΩ for fast mode).

Timing Diagram



- A **START** condition is a high-to-low transition on SDA when SCL is high.
- A **STOP** condition is a low to high transition on SDA when SCL is high.
- The address and the data bytes are sent most significant bit first.
- Master generates the clock signal and sends it to the slave during data transfer
- Data on SDA can be changed only when SCL is low.

Multiple Masters



- "Wired-AND" bus: A sender can pull the lines to low, even if other senders are trying to drive the lines to high
- In single master systems, arbitration is not needed.
- Arbitration for multiple masters:
  - During data transfer, the master constantly checks whether the SDA voltage level matches what it has sent.
  - When two masters generate a START setting concurrently, the first master which detects SDA low while it has intended to set SDA high will lose the arbitration and let the other master complete the data transfer.

Slave Addressing

- 7-bit, 8-bit, 10-bit I2C slave addressing
- Slave's address is often programmable
- Some reserved for special function

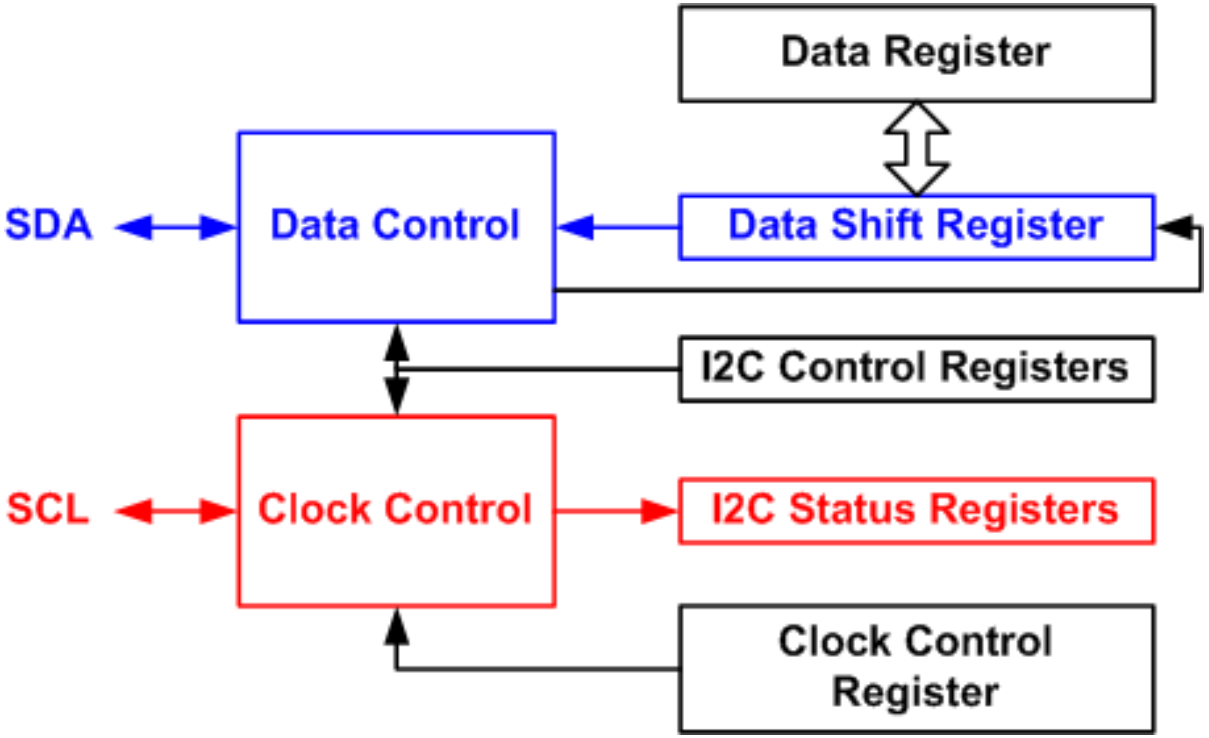

```
0 = write to slave
1 = read from slave
```

Slave Address	R/W bit	Description
0000 000	0	General call address
0000 000	1	START byte
0000 001	X	CBUS address
0000 010	X	Reserved for different bus format
0000 011	X	Reserved for future purposes
0000 1xx	X	Hs-mode master code
1111 1xx	X	Reserved for future purposes
1111 0xx	X	10-bit slave addressing

Working Modes

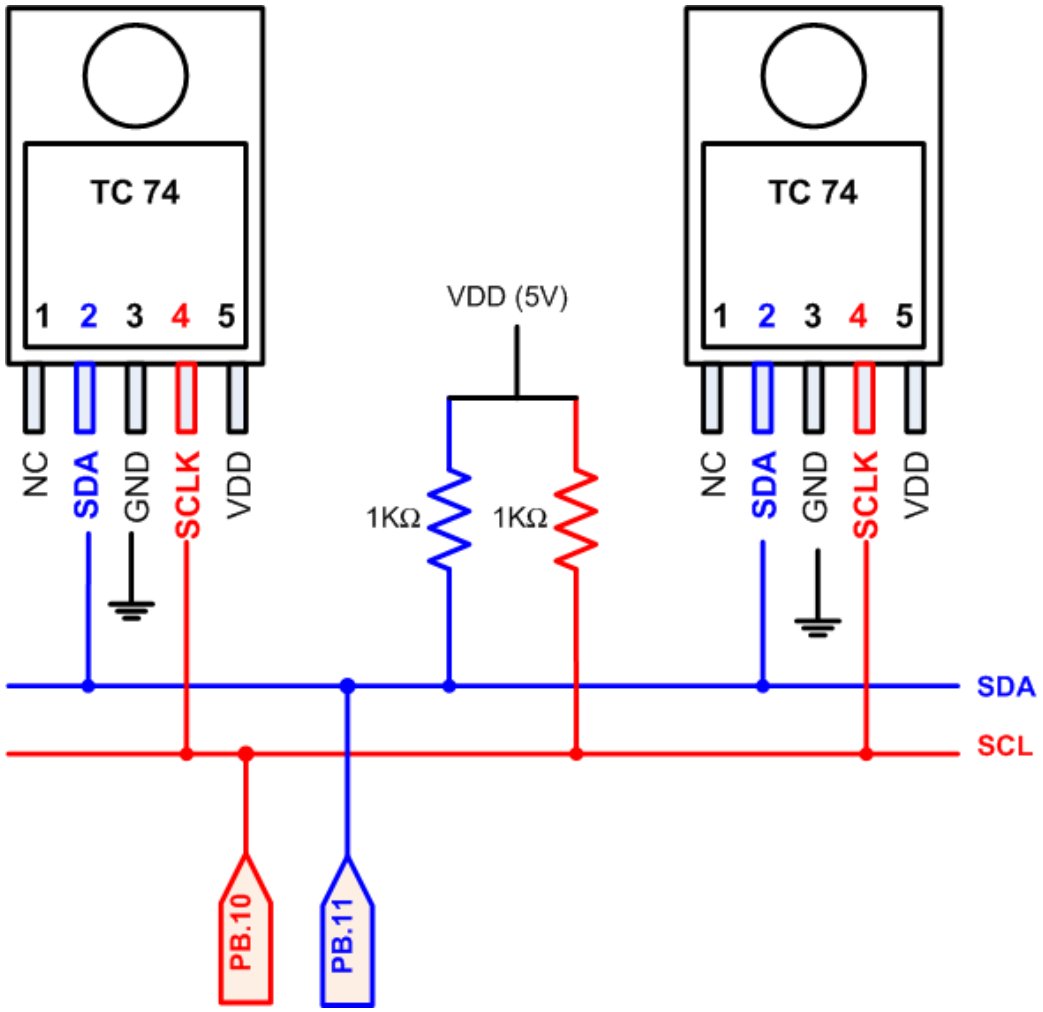
- Master-sender
  - Master issues START and ADDRESS, and then transmits data to the addressed slave device
- Master-receiver
  - Master issues START and ADDRESS, and receives data from the addressed slave device
- Slave-sender
  - Master issues START and the ADDRESS of the slave, and then the slave sends data to the master
- Slave-receiver
  - Master issues START and the ADDRESS of the slave, and then the slave receives data from the master.

STM32L I2C Module

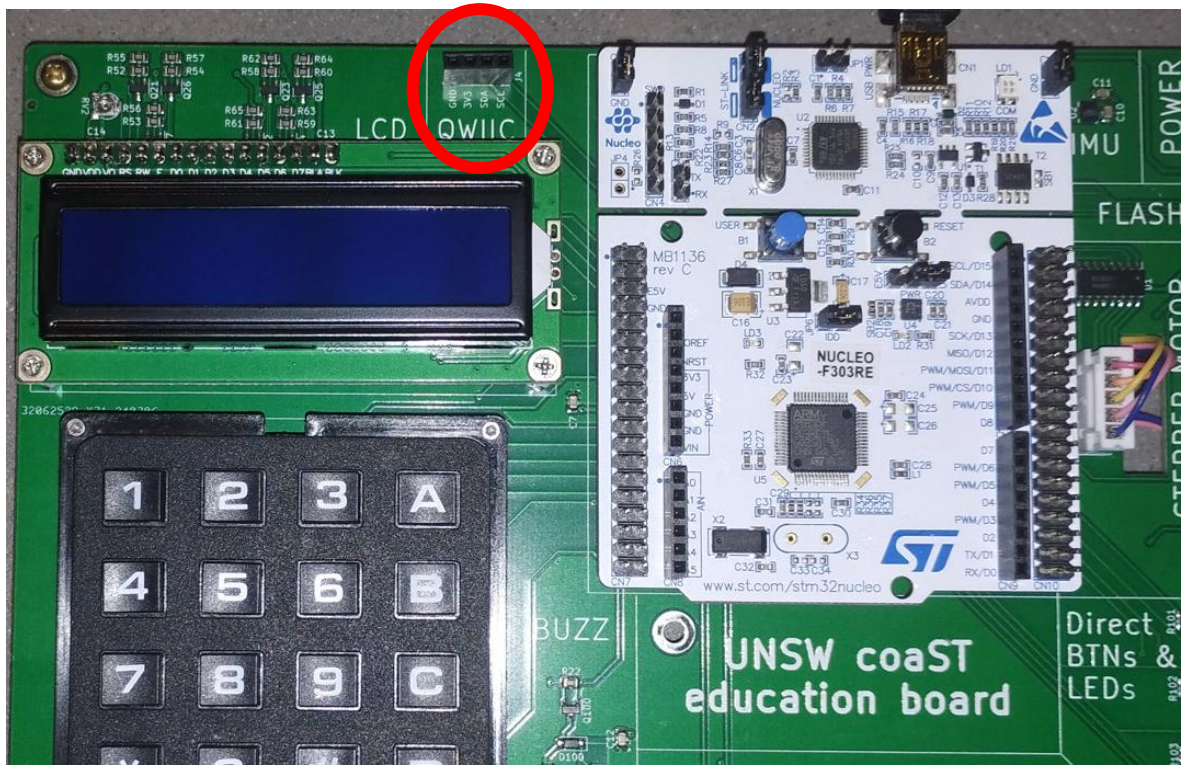


- During sending, the I²C hardware automatically sets the *TxE* flag in the status register if an acknowledge pulse is received from the slave.
- During receiving, the I²C hardware then automatically sets the *RxNE* flag in the status register if a byte has been successfully received.

Interfacing Serial Digital Thermal Sensor



On coast Board



QWIIC	Microcontroller PIN
QWIIC_SCL	PB8
QWIIC_SDA	PB9

https://www.sparkfun.com/qwiic

Tutorials for HAL

- https://wiki.st.com/stm32mcu/wiki/Getting_started_with_I2C

SPI vs I2C

- SPI by Motorola and I2C by Philips
- Both are synchronous protocols for short distance communications
- Generally, operates on 3.3 V or 5V

	SPI	I2C
Advantages	• Faster speed. Can be 10 Mbps or more. • Full duplex	• Simplicity. Commonly used as a 2-wire bus. • Adding new slave is easy