DESN2000
(Computer Engineering)

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
- Single master, multiple slaves
- Slave cannot communicate with slave directly
- Higher throughput than I2C

**SPI Master**

- SCLK: serial clock
- MOSI: master out slave in
- MISO: master in slave out
- SS1, SS2: slave select (active low)

**SPI Slave 1**

- SCLK: serial clock
- MOSI: master out slave in
- MISO: master in slave out
- SS: slave select (active low)

**SPI Slave 2**

- SCLK: serial clock
- MOSI: master out slave in
- MISO: master in slave out
- SS: slave select (active low)
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.
• Combination of CPOL and CPHA determines the clock edge for transmitting and receiving.

• CPOL = 0 $\rightarrow$ SCLK is pushed to low during idle. Otherwise, pulled to high during idle.

• CPHA = 0 $\rightarrow$ 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

Clock Phase (CPHA)
Clock Polarity (CPOL)
CPHA = 0
CPOL = 0
CPHA = 1
CPOL = 1
Sampling Edge
Toggling Edge
Sampling Edge
Toggling Edge
Sampling Edge
Sampling Edge
Sampling Edge
Sampling Edge
Sampling Edge
Sampling Edge
Sampling Edge
Gyro Sensors

Z (yaw)

Y (Pitch)

X (Roll)

SPI Master (STM32L4)
- SPI2_Clock (PD1)
- SPI2_MOSI (PD4)
- SPI2_MISO (PD3)
- GPIO PD7
- GPIO PD2
- GPIO PE8
- GPIO PE0
- GPIO PC0

SPI Slave (L3GD20 Gyro)
- MEMS_SCK
- MEMS_MOSI
- MEMS_MISO
- GYRO_CS
- GYRO_INT1
- GYRO_INT2
- XL_CS
- MAG_CS

GPIO PD7
GPIO PD2
GPIO PE8
GPIO PE0
GPIO PC0

GPIO PE0
GPIO PC0

SPI Master
SPI2_Clock (PD1)
SPI2_MOSI (PD4)
SPI2_MISO (PD3)
GPIO PD7
GPIO PD2
GPIO PE8
GPIO PE0
GPIO PC0

SPI Slave
MEMS_SCK
MEMS_MOSI
MEMS_MISO
GYRO_CS
GYRO_INT1
GYRO_INT2
XL_CS
MAG_CS
On coaST Board

ACCELEROMETER

UNSW coaST education board

DirectBTNs &LEDs
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 I²C protocol. However, fees are still required to obtain I²C slave addresses allocated by NXP (acquired Philips).

• Characteristics
  • Serial, byte-oriented
  • Multi-master, multi-slave
  • Two bidirectional open-drain lines, plus ground
    • Serial Data Line (SDA)
    • Serial Clock Line (SCL)
    • SDA and SCL need to pull up with resistors
Inter-Integrated Circuit (I2C)

- 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.
- SDA and SCL have to be open-drain
- Each Device has a unique address (7, 10 or 16 bits). Address 0 used for broadcast
- STM32L’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).
• 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.
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 \textit{TxE flag} in the status register if an acknowledge pulse is received from the slave.
- During receiving, the I2C hardware then automatically sets the \textit{RxNE flag} in the status register if a byte has been successfully received.
Interfacing Serial Digital Thermal Sensor
On coast Board

https://www.sparkfun.com/qwiic
Tutorials for HAL

- [https://wiki.st.com/stm32mcu/wiki/Getting_started_with_I2C](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

<table>
<thead>
<tr>
<th>Advantages</th>
<th>SPI</th>
<th>I2C</th>
</tr>
</thead>
<tbody>
<tr>
<td>Faster speed.</td>
<td>• Can be 10 Mbps or more.</td>
<td>• Simplicity. Commonly used as a 2-wire bus.</td>
</tr>
<tr>
<td></td>
<td>• Full duplex</td>
<td>• Adding new slave is easy</td>
</tr>
</tbody>
</table>