STM32 GPIO Alternate Functions

- GPIO pins are multiplexed with other peripherals.
- Each GPIO pin can have up to 16 alternate functions.
  - Refer STM32 data sheet for the details.
- Alternate function (0 to 15) can be set in GPIO AFR register.
  - GPIOx[7:0] alternate functions to be set in GPIOx_AFRL (4-bits for each pin).
  - GPIOx[15:8] alternate functions to be set in GPIOx_AFRH (4-bits for each pin).
- GPIO pin mode should be set to Alt Function [10] into MODER register.

Basic communication types

- Parallel
- Serial

Serial communication

- Synchronous
- Asynchronous

Serial Protocols

- peer to peer/end to end/Ad-hoc
- bus
- star

Protocol types

- Simplex
- Half Duplex
- Full Duplex

USART vs UART

- USART -- Universal Synchronous Asynchronous Receiver Transmitter
  - Single byte or multi-byte communication (in a data frame)
- UART -- Universal Asynchronous Receiver Transmitter
  - Single byte communication (in a data frame)

RS-232

- Physical characteristics

  - Type

    - Full-duplex
    - Peer-to-peer

  - Connections/Wires/Pins

- Tx: Transmit
- Rx: Receive
- Gnd

- Connector

    - Old: DB-25
    - Standard: DB-9

- Half Serial Cable: Only 3 wires connected -- Rx, Tx, Ground.

- Full Serial Cable: 9 wires connected

- Voltage/current levels

    - NRZ (Not Returning Zero)
    - CMOS Voltage levels
        - +3V to +25V -- Logic 0 -- Space
        - -3V to -25V -- Logic 1 -- Mark *TTL Voltage (0V-0 and 5V-1) ---> RS-232 Line Driver ---> CMOS Voltage levels
    - TTL <--> MAX-232 <--> CMOS

- Bit-rate / Baud rate

    - Standard baud rates: 9600, 38400, 115200, ...
- Logical characteristics
    - Data frame
        - Start bit --> Always 0
        - Data bits --> 5 to 9 bits -- LSB Tx First
        - Parity bit --> Even/Odd/1/0/No
        - Stop bit(s) --> Always 1
    - Error conditions
        - Partity error: Parity mismatched on receiver end (w.r.t. config)
        - Frame error: Stop bit is 0
        - Read overrun: The uC is processing some data and not read the data transmitted.
        - write overrun:
        - Noise error: Detected in STM32 with oversampling.

STM32 USART

- STM32F407xx has 4 USART and 2 UART.

- Features

    - Full-duplex, synchronous & asynchronous communication
    - NRZ standard format
    - Flexible configuration
    - Fractional baud rate * Programmable word length, stop bits, parity....
    - Transfer detection flags
    - Error detection flags

- Parity, Overrun, Frame and Noise error
- Transfer and Error interrupts
- Programmable hardware control

- STM32 Oversampling

  - To detect noise while receiving data over-sampling technique is employed in STM32.
  - This technique samples receiver pin signal multiple times (based on OVER8 bit).
    - For OVER8=1, max baud = PCLK/8 but receiver tolerance to clock deviation is less.
    - For OVER8=0, max baud = PCLK/16 but receiver tolerance to clock deviation is more. Preferred in noisy environment.

- STM32 UART baud rate calculation

- USART registers

  - USART_DR: Composed of two registers TDR & RDR.
  - USART_BRR: USART divisor
  - USART_CR1, CR2, CR3: Control registers
  - USART_SR: Hardware control, interrupts and error flags.

USART Tx and Rx with polling

- USART Initialization

  - Configure GPIO
    - Enable clock
    - Set Alternate function & mode
    - Config output type & pull-up/down
  - Configure USART
    - Enable clock
    - Comm. attributes (CRx)
    - Baud rate setting
  - Enable UART

- USART Send

  - Wait for last char to transmit (TXE flag)
  - Write the char into DR

- UART Receive

  - Wait for a char to be received (RXNE flag)
  - Read the received char from DR

STM32 Interrupt Handling

- USART Initialization
  - Configure GPIO
    - Enable clock
    - Set Alternate function & mode

- - - Config output type & pull-up/down
    - Configure USART
      - Enable clock
      - Comm. attributes (CRx)
      - Baud rate setting
    - Enable interrupt in NVIC (ISER)
    - Enable UART
- USART Send String
  - Wait for last string to transmit
  - Get ready for new transmission
    - string address
    - start index = 0
  - Enable TXEIE interrupt
- UART ISR
  - Verify the source of the interrupt
  - If string is not completed, send next char
  - Otherwise stop the transmission and mark the end of current transmission.