



# Embedded Systems



How to write code for Embedded Systems

# Outline

---

- ▶ Know the Hardware
- ▶ Writing “Embedded” code





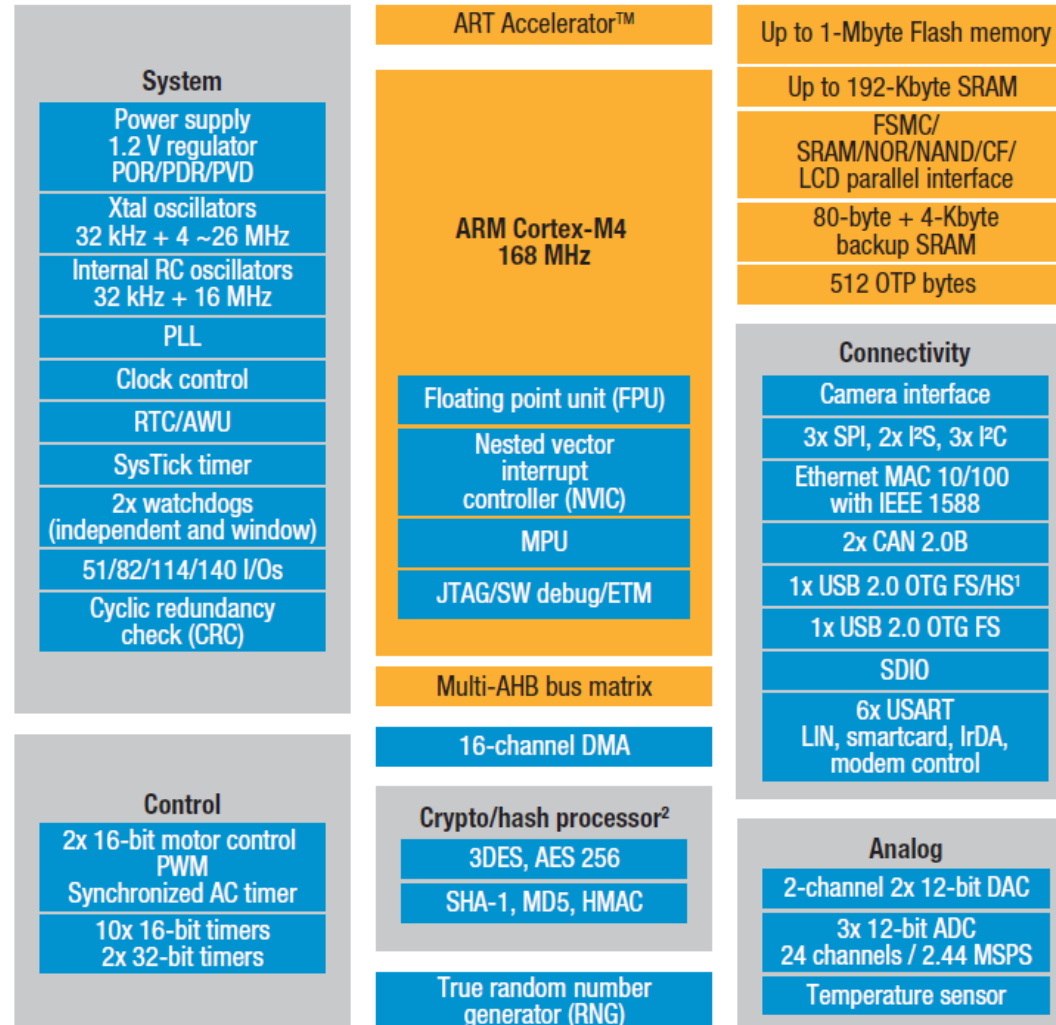
Know the Hardware

# STM32F4DISCOVERY

- ▶ STM32F407VGT6 microcontroller
  - ▶ 32-bit ARM Cortex-M4F core
    - ▶ 168 MHz
    - ▶ single cycle DSP MAC
    - ▶ floating point unit
  - ▶ 1 MB Flash,
  - ▶ 192 KB RAM
  - ▶ USART: 10.5 Mbit/s
  - ▶ SPI: 37.5 Mbit/s
  - ▶ ADC: 2.44 MSPS



# STM32F4DISCOVERY



# The Big Picture

---

- ▶ **Understand the system**
  - ▶ What are the major functionalities of this system?
  - ▶ What are the inputs/outputs interfaces?
  - ▶ What are the provided HW modules/peripherals?
- ▶ **Read the documents !**
  - ▶ Product description/brochure
  - ▶ User guide
  - ▶ Programmer guide
  - ▶ Reference manual
  - ▶ Application notes
  - ▶ Schematics
  - ▶ Errata



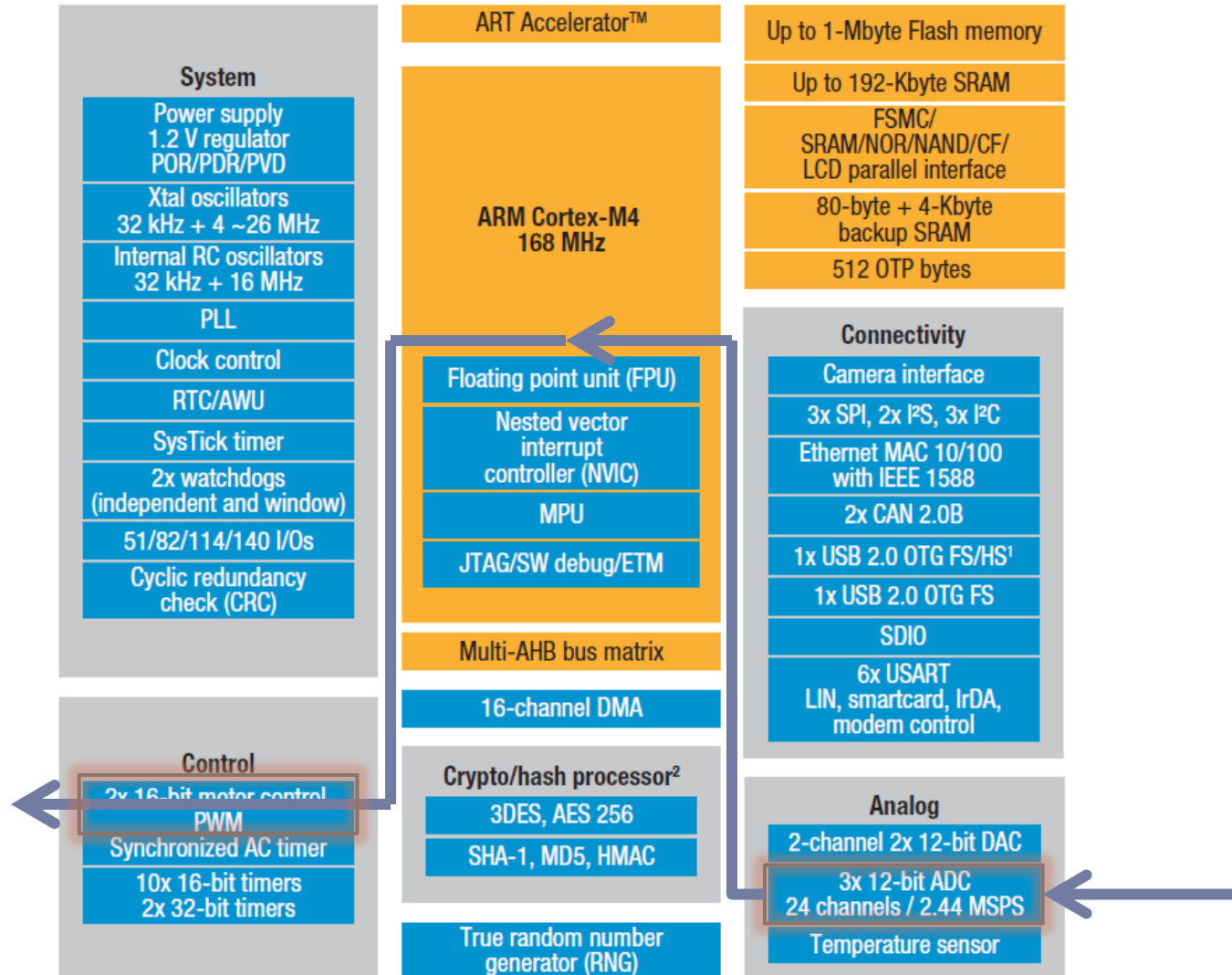
# The Big Picture

---

- ▶ **Observe the community**
  - ▶ Opensource ?
- ▶ **Get a board !**
  - ▶ Application fits the board?
    - ▶ Processing requirements
    - ▶ Data flow



# Data Flow



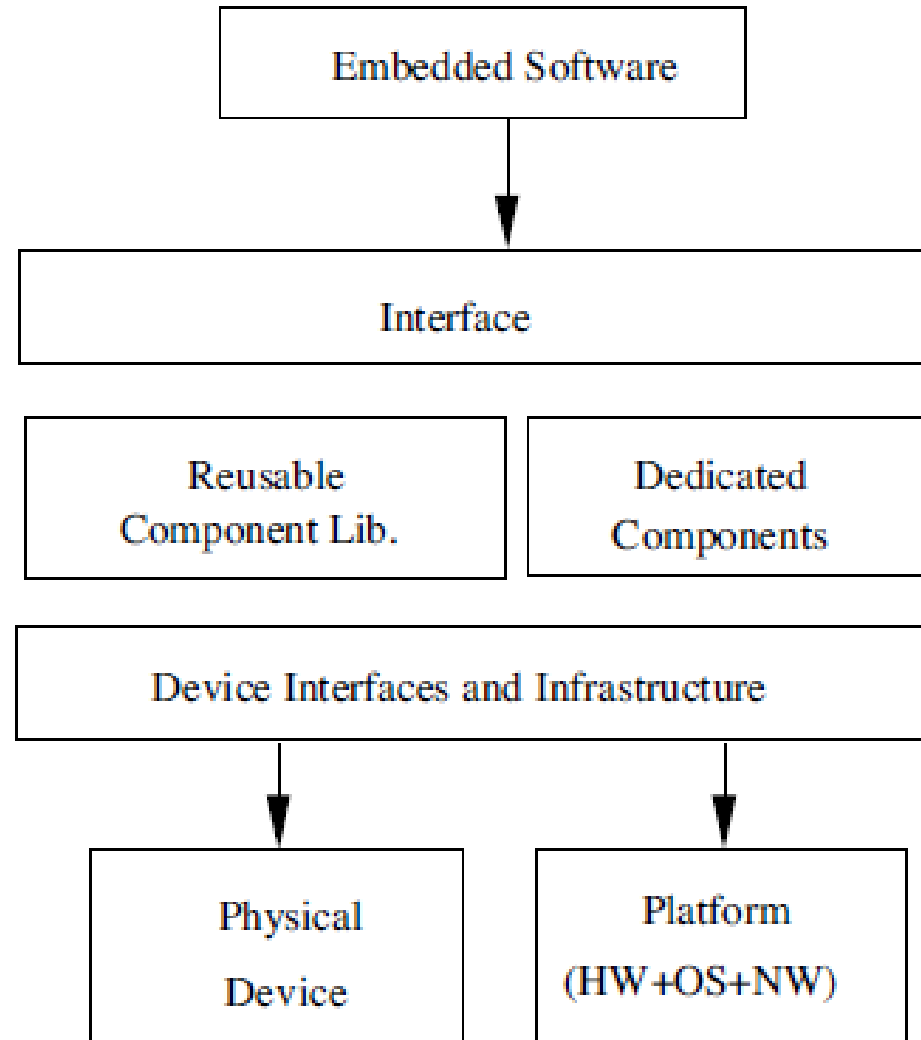




## Writing “Embedded” code

# Components

---



# Embedded Software Architecture

---

- ▶ **Simple Control Loop**
  - ▶ Single function
  - ▶ FSM
- ▶ **Barebone**
  - ▶ Interrupt-controlled
  - ▶ Cooperative multitasking
  - ▶ Preemptive multitasking
- ▶ **Operating System**



# Constrains on Embedded Processing

---

- ▶ **On Operation**

- ▶ Stand-alone
- ▶ Error-handling

- ▶ **On Recourses**

- ▶ Memories
- ▶ Processing
- ▶ Power consumption
- ▶ Peripherals

- ▶ **On Execution**

- ▶ RT constraints

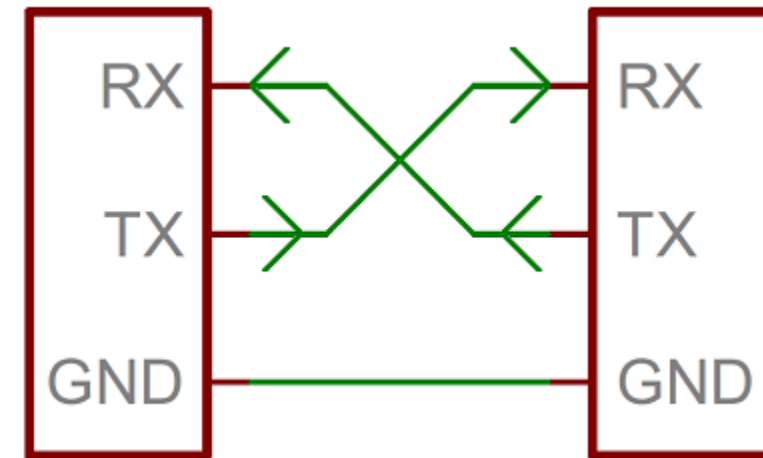


# Supplementary Info

---

## ▶ UART

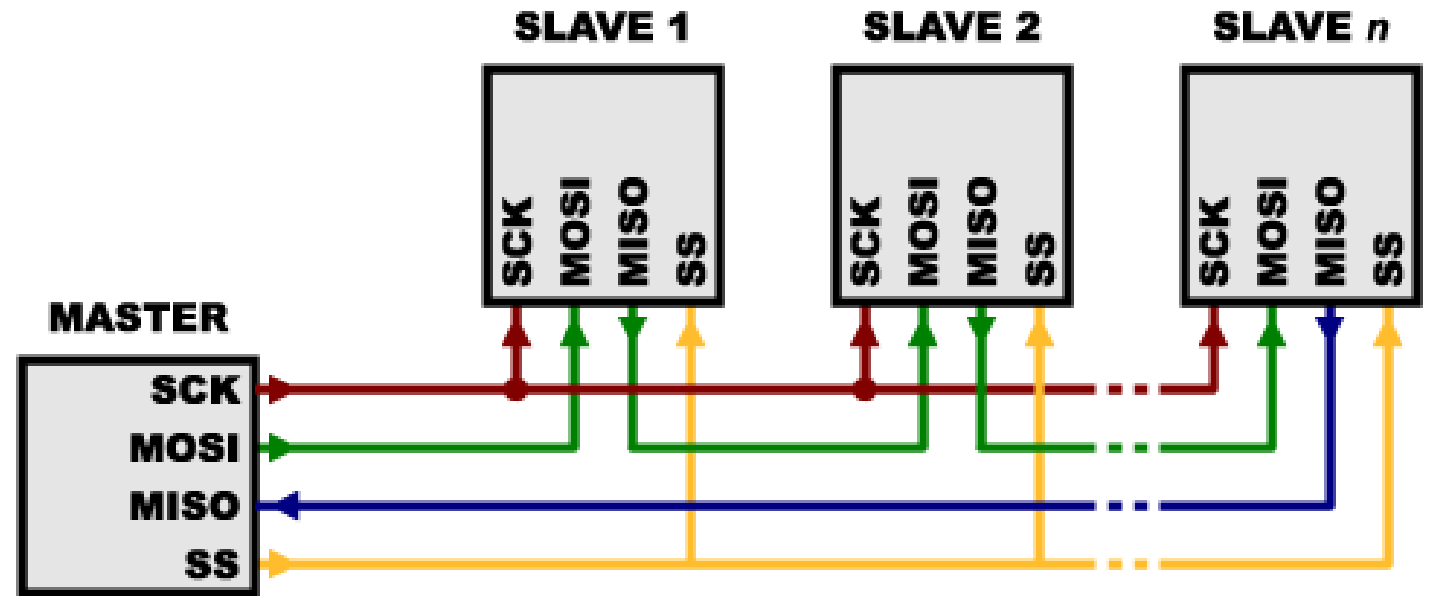
- ▶ Universal Asynchronous Receiver/Transmitter
- ▶ Data are sent sequentially as individual bits.
- ▶ Sender and receiver have to agree on transmission characteristics:
  - ▶ Baud Rate: how fast data is sent over a serial line
  - ▶ Framing: number of bits
  - ▶ Synchronization: start and stop sequences
  - ▶ Error detection: parity bits
  - ▶ Flow control



# Supplementary Info

---

- ▶ SPI
  - ▶ Serial Peripheral Interface
- ▶ Targets asynchronous problems in serial communication
  - ▶ Uses separate lines for data and clock signals



# Memory Map

---

- ▶ **Where things are !**
  - ▶ On-chip memory
  - ▶ Off-chip memory
  - ▶ Registers
  - ▶ Peripherals
- ▶ **Memory-mapped I/O**
  - ▶ CPU uses read/write operations on different I/O and peripherals on the chip



# Memory Map

---

Reserved	0xE010 0000 - 0xFFFF FFFF
CORTEX-M4 internal peripherals	0xE000 0000 - 0xE00F FFFF
Reserved	0xA000 1000 - 0xDFFF FFFF
AHB3	0xA000 0FFF
Reserved	0x6000 0000
Reserved	0x5006 0C00 - 0x5FFF FFFF
AHB2	0x5006 0BFF
Reserved	0x5000 0000
Reserved	0x4008 0000 - 0x4FFF FFFF





# Walkthrough

---

- ▶ STM32FDISCOVERY UART

