**Abstraction layer of STM32 to development**

**Baremetal:**

Directly writing to MCU registers using their addresses without any abstraction.

**CMSIS:**

Cortex Microcontroller Software Interface Standard. It is essentially baremetal with structure.

It provides :

* Standardized register name and structure.
* Startup code and system initialization.
* Core peripheral access.
* DSP and math function.

It provides portability across all Cortex-M MCUs.

**Low Layer (LL):**

A lightweight low overhead API provides direct control over registers but with some readable functions.

**Hardware abstraction Layer (HAL):**

High-level user-friendly API for configuring and using peripherals. It provides easy-to-use functions but comes with some overhead.

**A simple microcontroller architecture:**

**CPU:**

central processing unit. Executes instruction and processes data.

**Memory:**

(flash, RAM, EEPROM). Stores program code and runtime data.

**System bus** (AHB, APB) **:**

Connects CPU, memory and peripherals.

**Peripherals:**

(IO, timer, ADC, Communication interface). Extends MCU functionality. Peripherals are built-in hardware modules that extend the functionalities of an MCU, allowing it to interact with the external world.

**Peripheral Integration in Microcontroller Architecture**

**Bus:**

* **AHP** advanced high performance bus.
* **APB** Advanced Peripheral Bus.

**Memory Mapped IO:**

Each Peripheral is mapped to a specific address in memory. Allowing the CPU to control it using register read/write operation. Each peripheral has control and data registers mapped to specific addresses in memory.

Interaction with peripherals-

* Configuring registers.
* Writing Reading data.
* Handling status flag.

**Interrupt-driven peripherals:**

Instead of constantly checking (polling) the peripheral registers, MCUs uses interrupts to notify the CPU when a peripheral needs Attention.

**Direct memory access:**

Some high-speed peripherals use direct memory access to transfer data without CPU intervention. (ADC UART SPI)

**Peripheral power and clock management:**

The clock control unit controls the peripheral clock. Allowing peripherals to be enabled or disabled dynamically.

**General purpose Input-output (GPIO) registers of STM32F10x.**

* Port configuration register. (GPIOx\_CRL, GPIOx\_CRH)
* Port input data register. (GPIOx\_IDR)
* Port output data register. (GPIOx\_ODR)
* Port bit reset register. (GPIOx\_BRR)
* Port bit set reset register (GPIOx\_BSRR) [atomic]
* port configuration lock register. (GPIOx\_LCKR)

**To setup the GPIO pin.**

* Enable clock for the port.
* Find the bus that the port is connected to.
* Enable the port of that bus.
* Set pin mode and configuration.