EPITECH. TECHNOLOGY

# T-DEV-811 / IOT

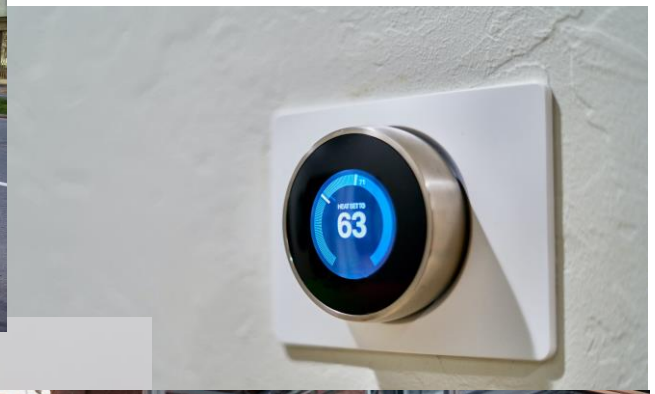
*Smart trash cans*

# IOT+VIR

- **Example : industrial maintenance**

# IOT

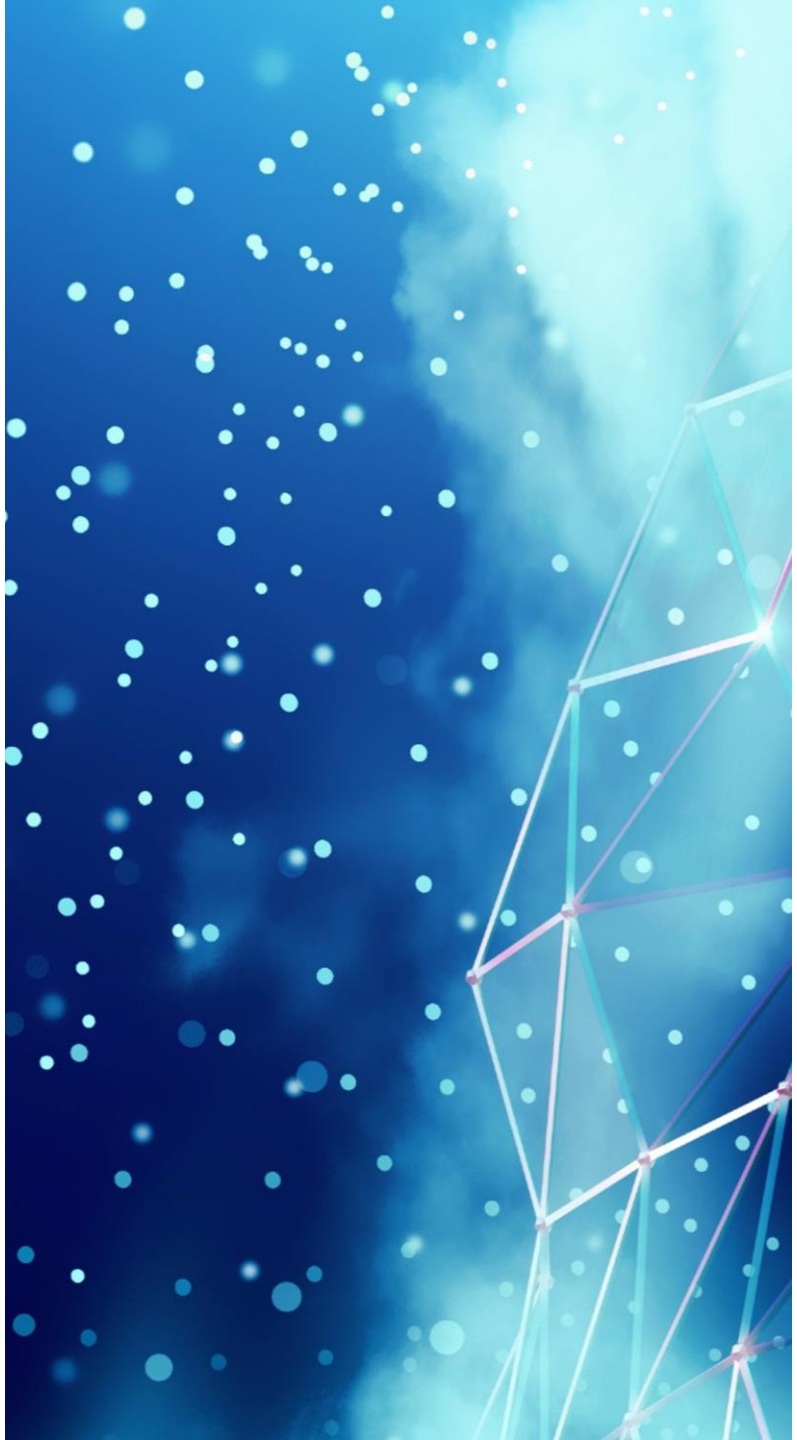
- Internet of the Things
- Extremely wide range of applications


# How many devices ?

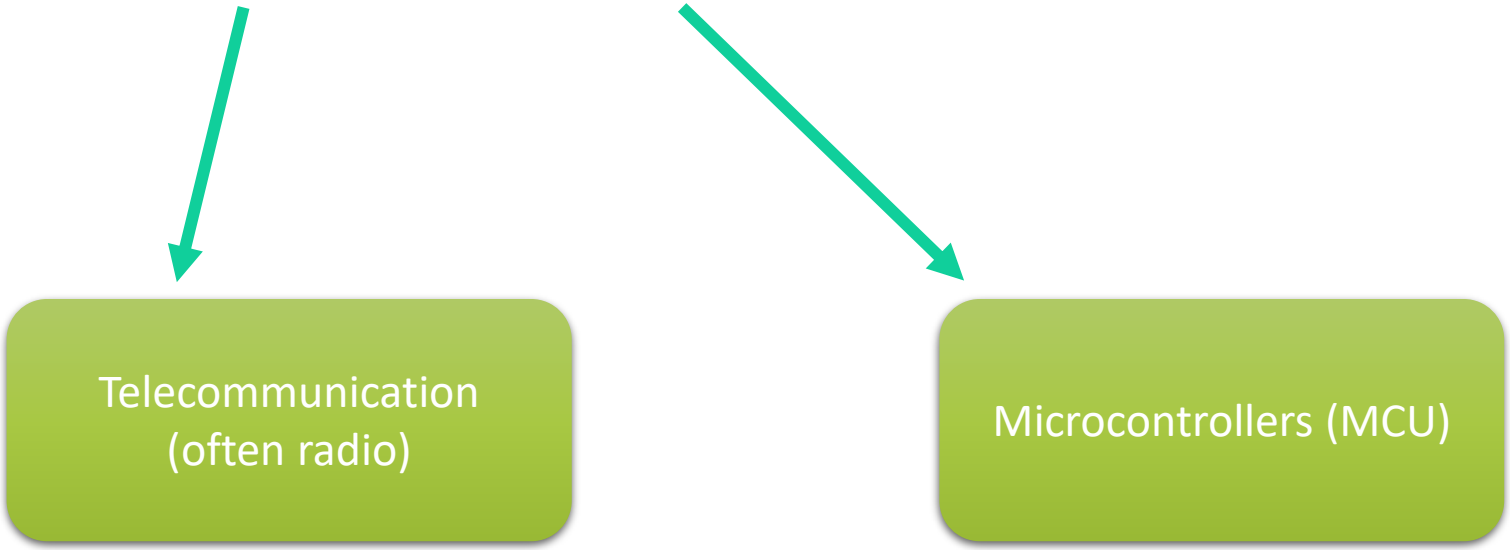
- **2018 : 7 billions devices**
- **2025 : 21 billion devices (estimated)**

- **Far more connected objects than humans using Internet**

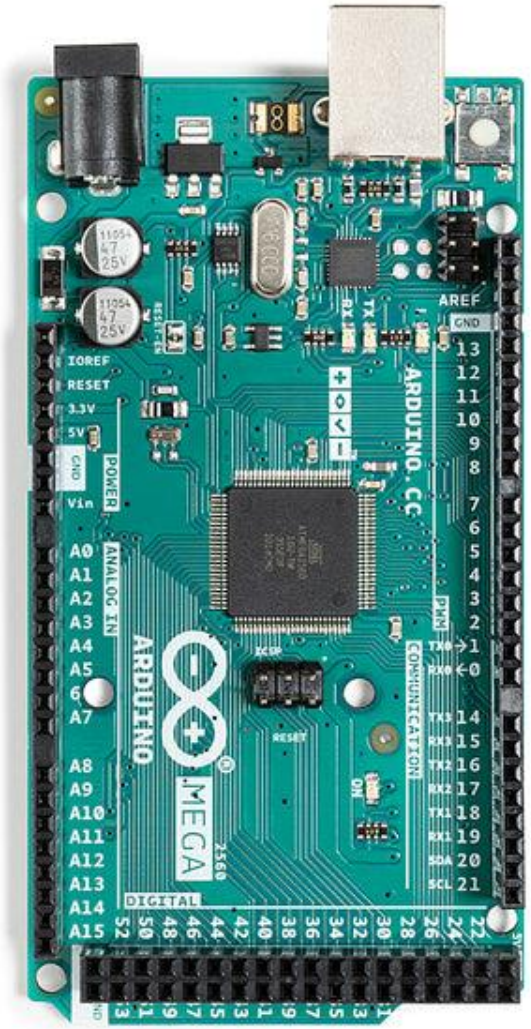


# IOT

- **Internet of the Things**

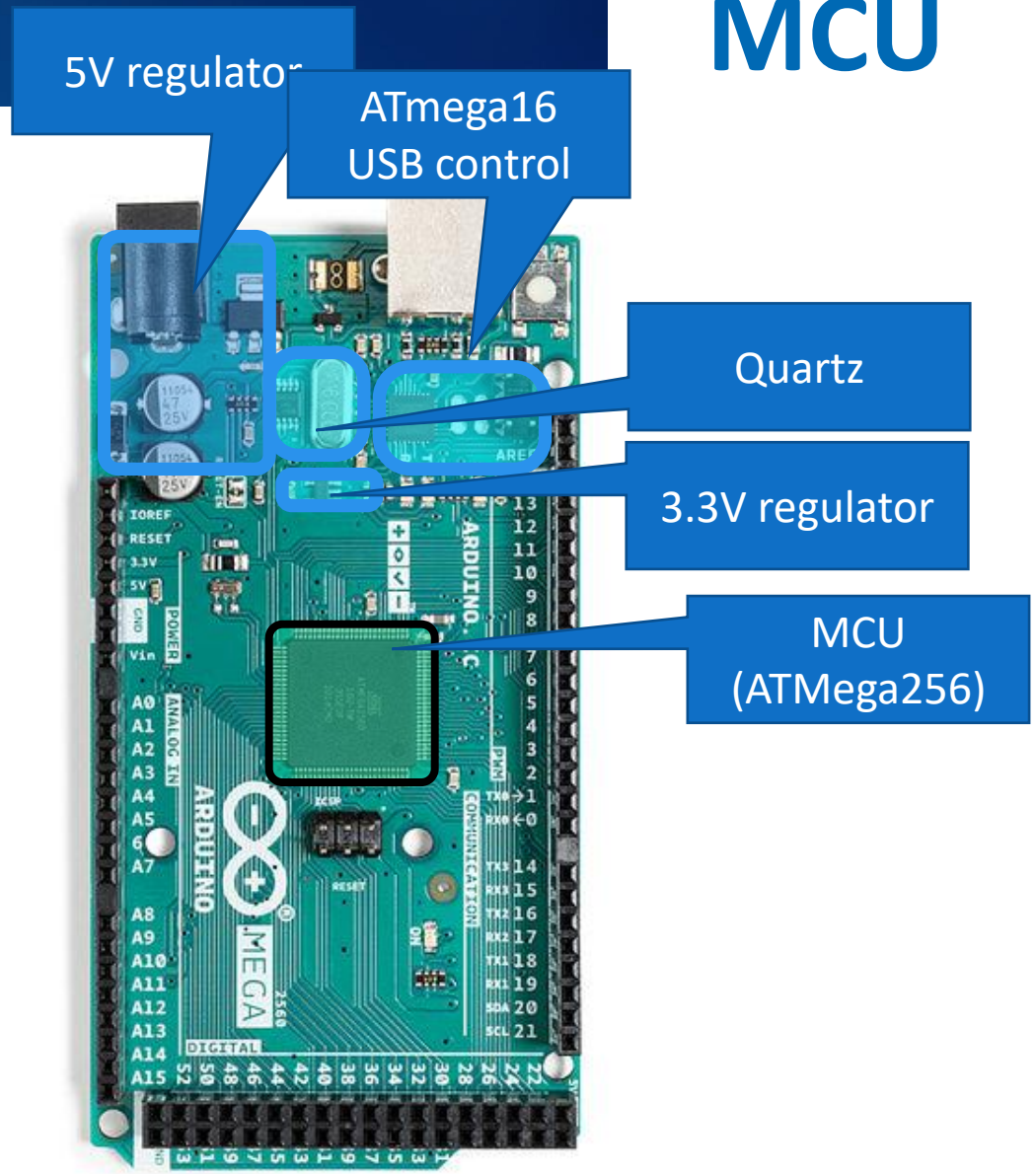


# MCU



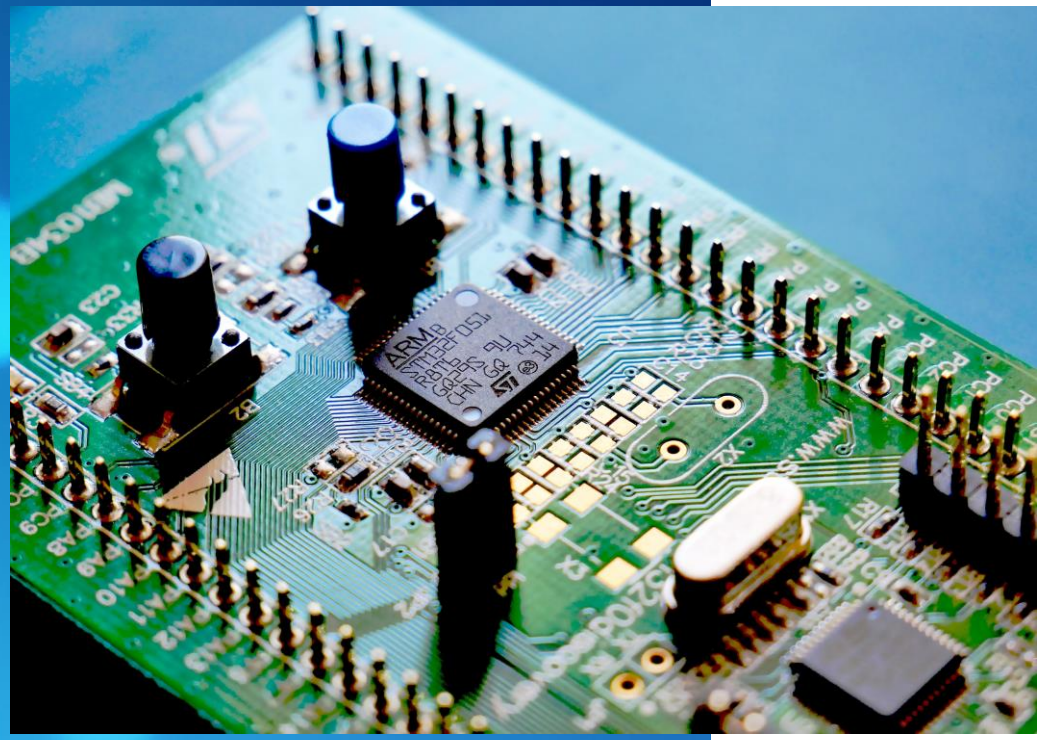

- **« Everything on a single chip »**
- **Very few side components needed**
  - Even quartz is optional

# MCU

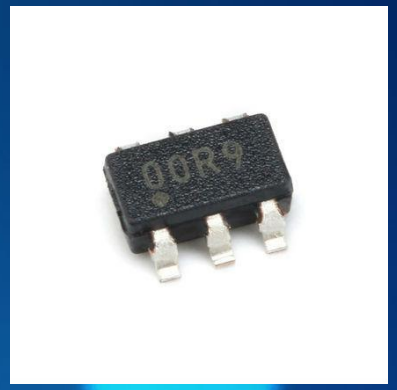



- **« Everything on a single chip »**
- **Very few side components needed**
  - Even quartz is optional

# MCU



- **Integrates all parts of computer** (CPU, Flash, RAM, IOs…)
- **Mostly : single chip board** (only the MCU + power)
  - Even quartz is optionnal
- **Low range CPU :**
  - 8 bits are common
  - Low MHz (low power)
- **Few kB of RAM and Flash**
- **But many IOs**




# MCU

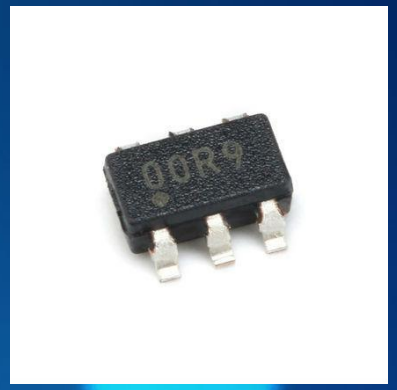
- **Mass market**

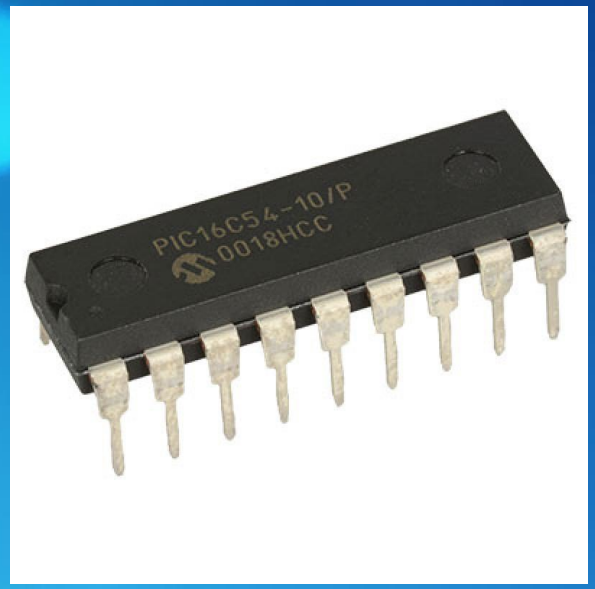
  (billions of units / years)

- **Cost effective**

  (dozen of units in a car ; multiply unit cost by number of produced car…)

- **Very wide range of MCU**
  - Cost effective
- **Time to market**

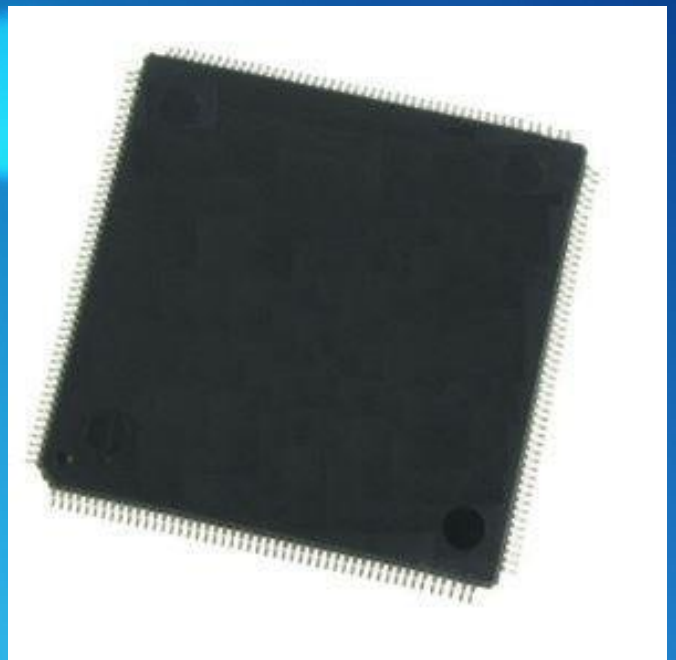



# MCU examples

- **PIC10F200**
  - 6 pins package
  - 256 words FLASH for programs
  - 16 bytes RAM
  - 4MHz
  - datasheet

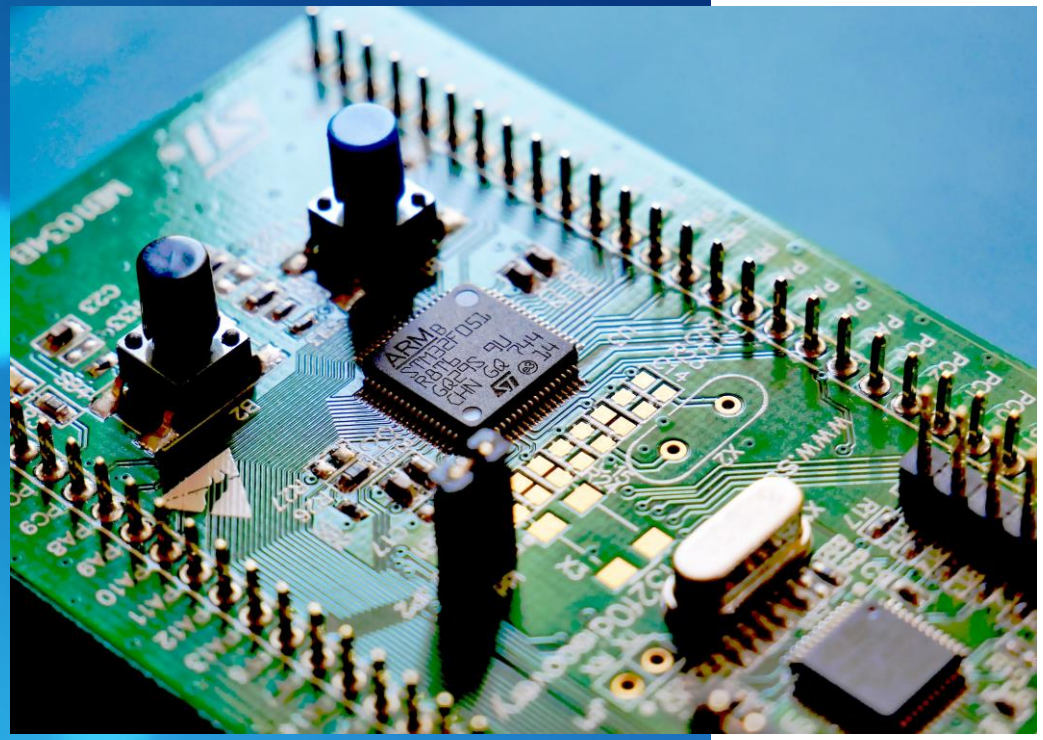
- **PIC16C54**
  - 18 pins package
  - 8bit (12 bits instructions), 40 MHz
  - ROM : 512 words, RAM 20 bytes
  - 12 IO pins
  - PIC16C54 | Microchip Technology

# MCU examples

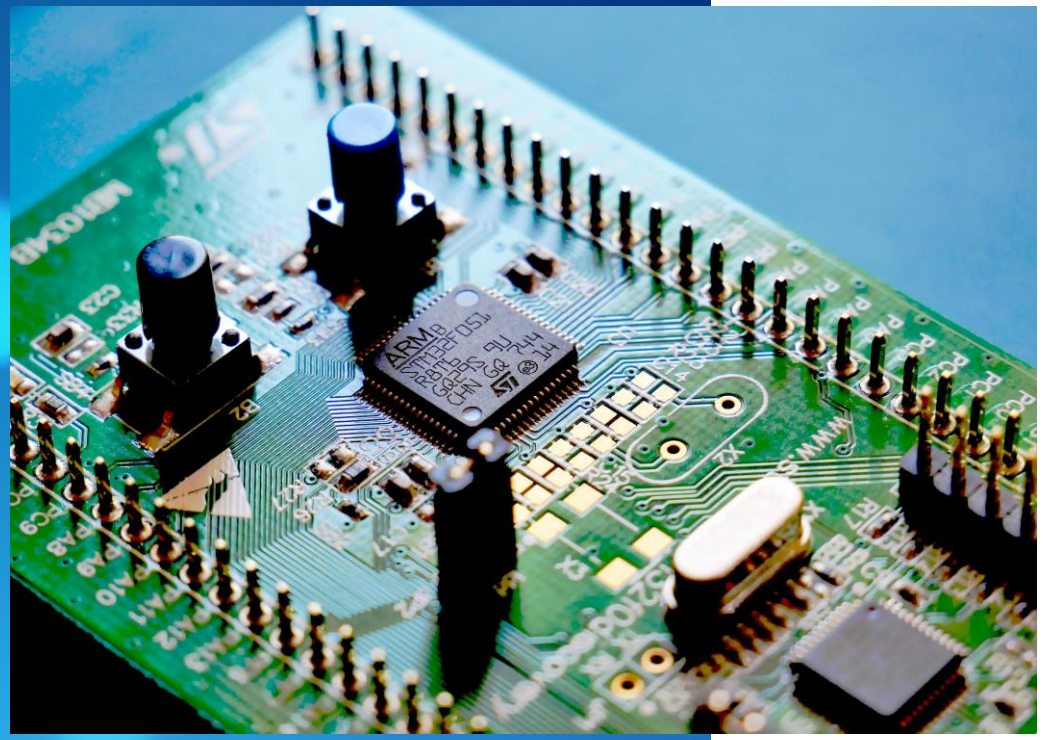
- **STM32F103**
  - 68K flash, 20K RAM
  - ARM 32 bits, 72MHz
  - 80 GPIO, 7 timers, 9 communication interfaces, 16 ADC channels, 7 DMA channels
  - datasheet
- **STM32H757**
  - Dual ARM Cortex-M5, 480 MHz
  - DP FPU
  - JPEG codec, crypto hardware
  - 2MB FLASH, 1MB RAM
  - datasheet

# MCU



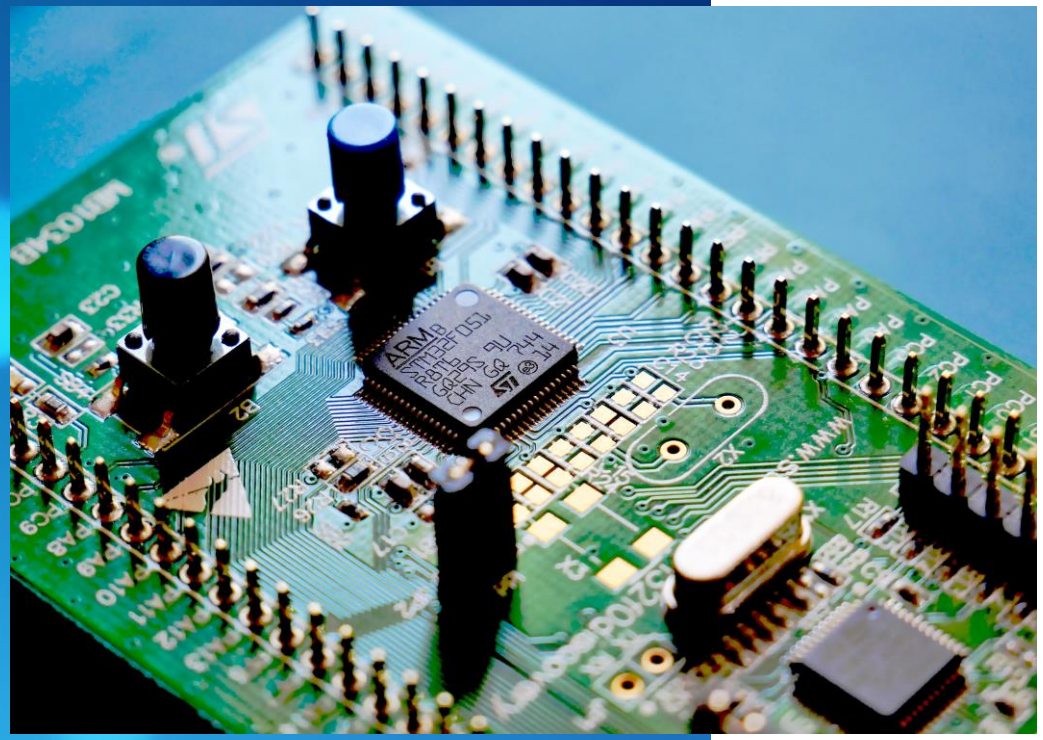
- **Low power usage**
  - For battery powered devices or solar powered
  - MCU often have low consumption
  - Configurable frequencies
  - Advanced low power modes

# Inputs/Outputs



- **GPIO**
  - General purpose IO, can be configured as input or output
- **Timers**
  - Some can be used to generate PWM, or as counter
- **ADC**
  - Generally multichannel, 12 bits
- **DAC**
  - Seldom present, PWM is suitable for many uses
- **Watchdog**

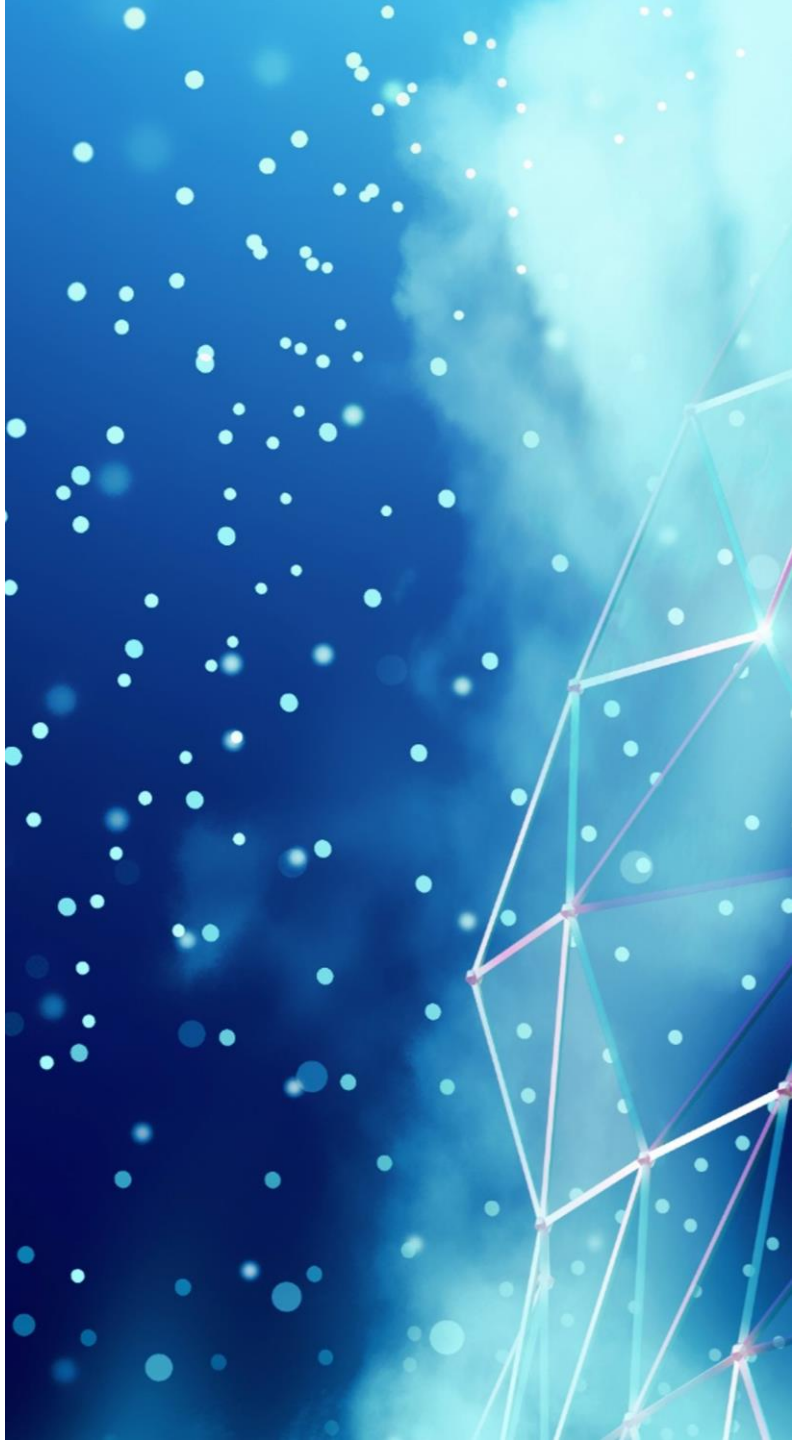
# Communication channels



- **I2C**
  - Communication with sensors / actuators
- **CAN**
  - Communication bus for multiple MCU (automotive)
- **UART**
  - « old style» serial port, asynchronous or synchronous
- **SPI**
  - High speed serial port (serial flash, etc..)
- **USB, support for ethernet, wifi, display, etc...**

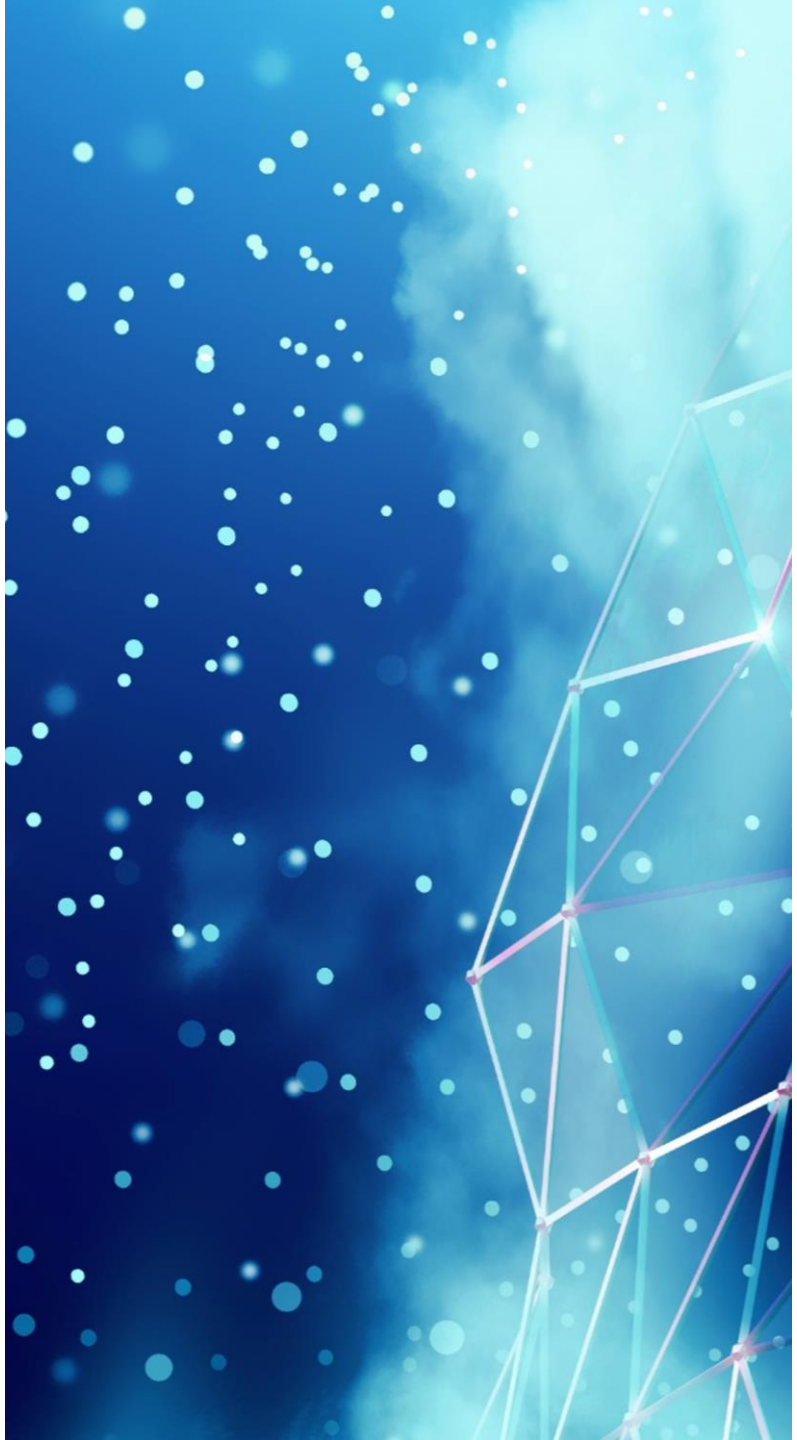
# Sensors and actuators

- **Interaction with real words**
- **Sensors**
  - Get information from the world
  - Can be simple (contact, IR barrier, etc…)
  - MEMS : inertial sensor, pressure measure…
- **Actuators**
  - Act on the world
  - Generally require external power !
  - Motors (DC, step, servo), relay
  - Display, LED, audio…
  - MEMS (ex: inkjet), DLP…



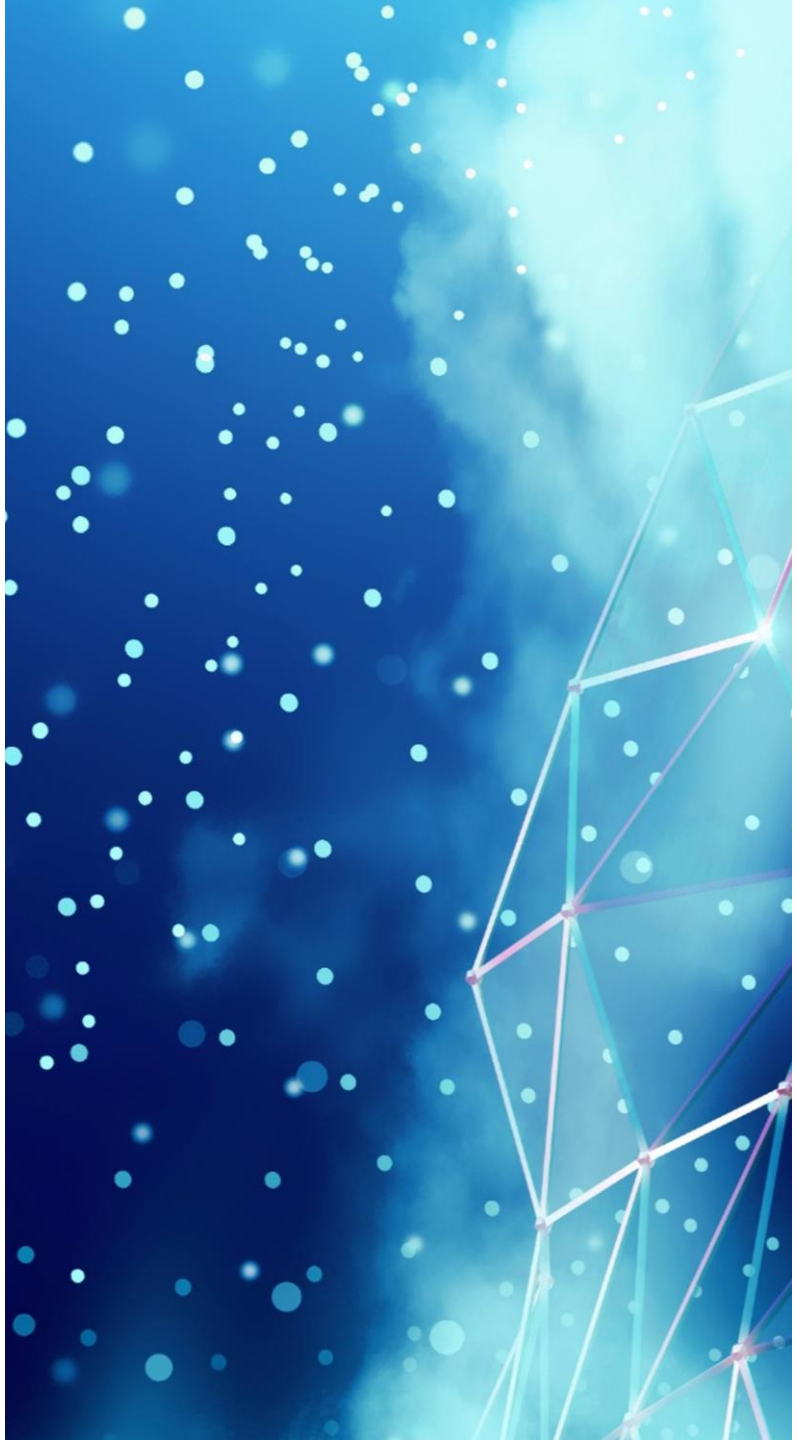
# MCU difficulties

- **Low memory**
  - Low level languages
  - Design constraints (algorithm choice)
- **Interaction with real world**
  - Problems cannot be easily reproduced
  - Breakpoint : can’t continue after a breakpoint
  - Unit tests are more complex
- **Hardware/Software**
  - Both developed simultaneously (often)
  - Investigate if problem is hardware or software (or connection)
- **But often very high reliability expected !**



# MCU difficulties

- **Hopefully small programs**
- **Strict methodology**
  - Never change hardware and software at the same time
  - QA standards for automotive, aviation, space
- **Test environments**
  - Dedicated hardware that simulate external world
  - Cross compilation on PC for tests / debug
  - Possibly MCU emulation (eg. QEMU)
  - …



# For this project

- **Discovering of MCU**
- **Discovering of sensors and actuators**
- **Discovering MCU development with both hardware and software**

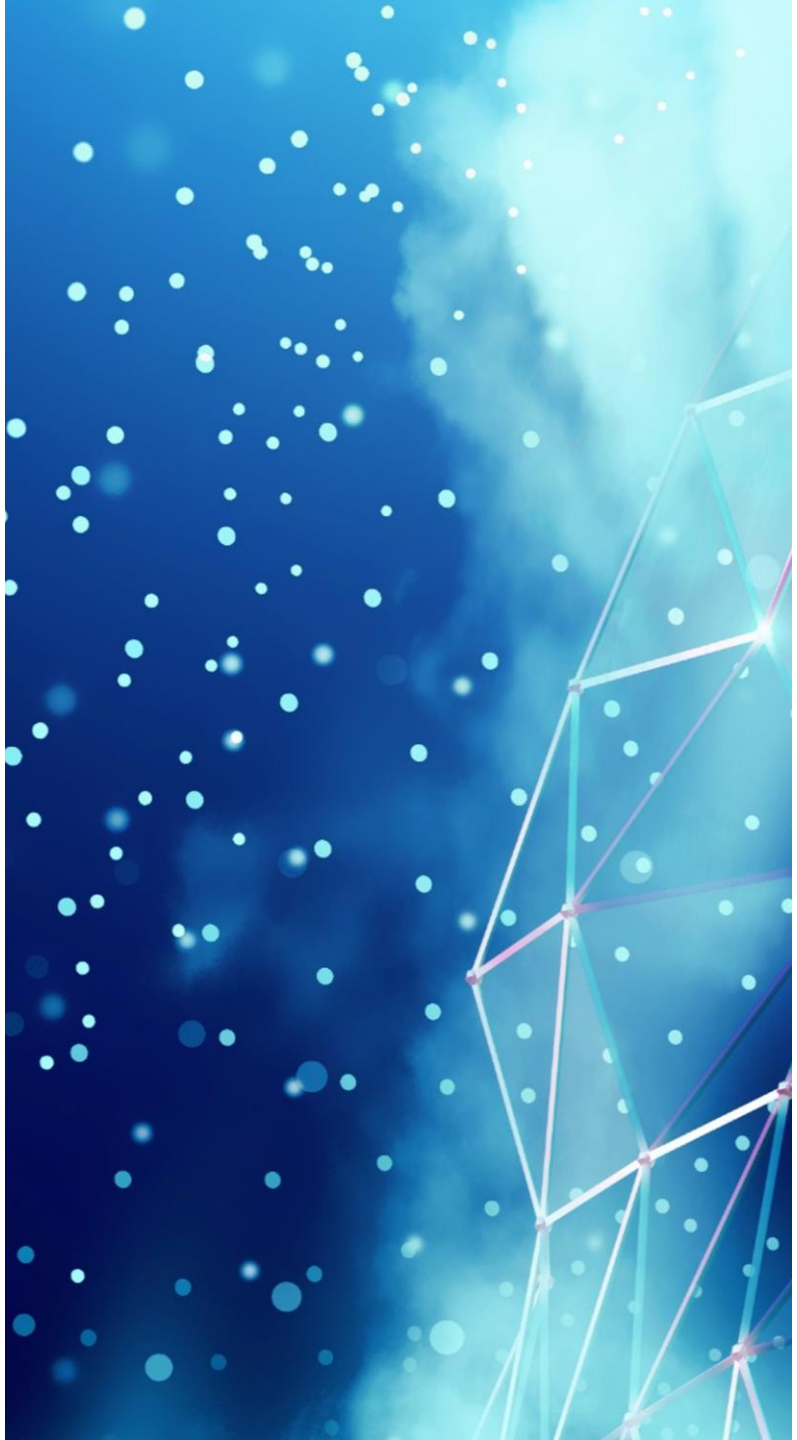
- **Many of your issues : cable connection, sensor misuse**
- **Wide knowledge requirements :**
  - HW and SW are closely linked
  - Datasheets contains a lot of information, one need to get use to find relevant part.

# Communication

- **Shannon theorem**

$$C = B\ \boldsymbol{log}(1 + \frac{S}{N})$$

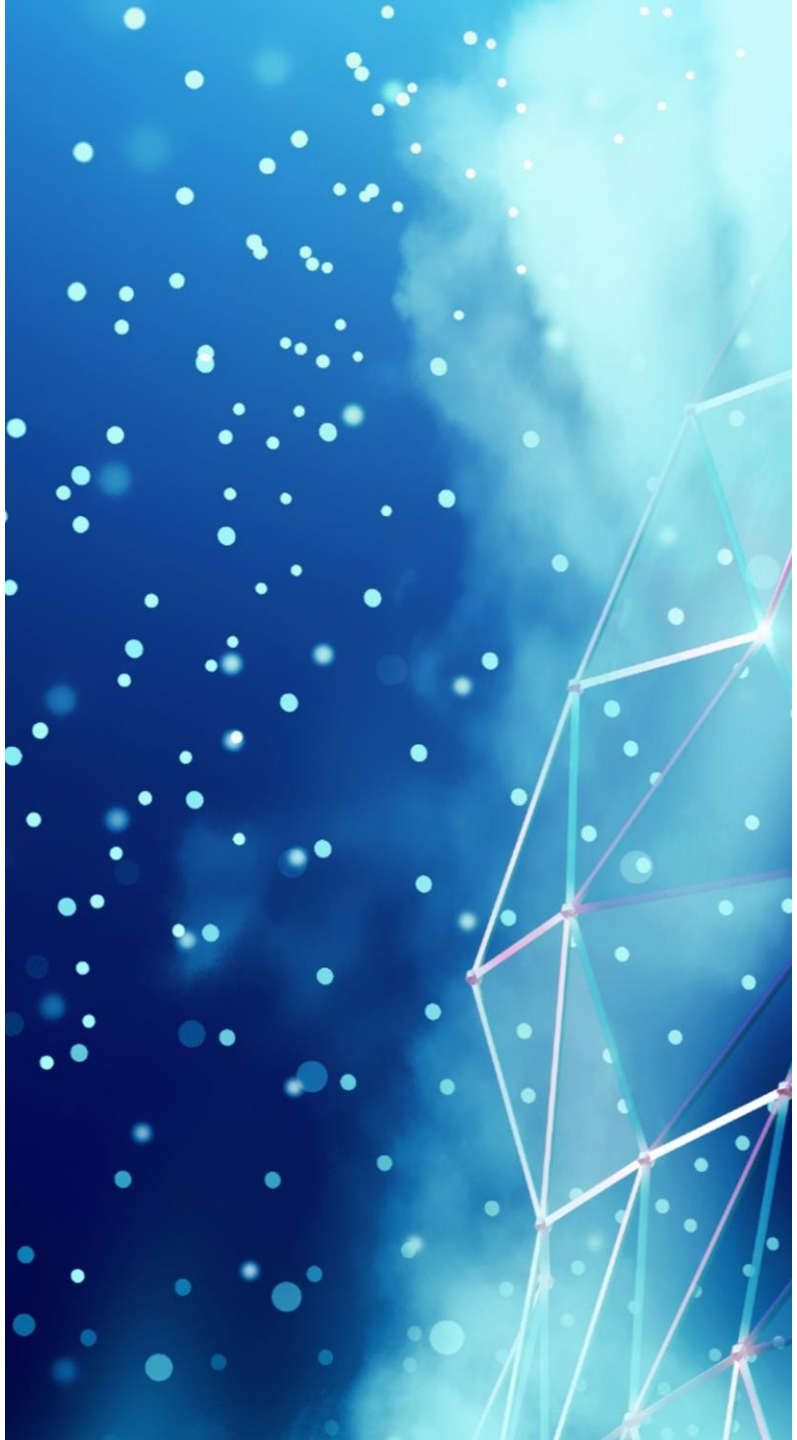
  - C channel capacity
  - B bandwidth
  - S signal, N noise
- **Noise cannot be suppressed (cosmic noise)**
- **Bandwidth is expensive + power consumtion**
- **Signal can be increase but radio Tx consume a lot**
- **It is not possible to have high data rate, low power and low cost**



# Communication

- **Most of the time, IoT only need to transmit**
  - Small messages
  - A few messages / day
  - Often long distance

- **Specific radio needs**
  - Possibly on existing technology (e.g. GSM), possibly adapted (LTE-M, NB-IoT)
  - Or dedicated technology (eg: LoRa)



# For this project

- **Wi-Fi communication**
  AT commands
- **No power consumption constraints**

# AT commands

- **Widely used for modems in the 80s - 90s**
- **Single UART channel**
  - Commands
  - Data
- **Still used for many modules (interfaced through UART)**
  - Bluetooth/BLE
  - Wifi
  - Etc… (allow command/data switching)

  - Hayes AT command set - Wikipedia

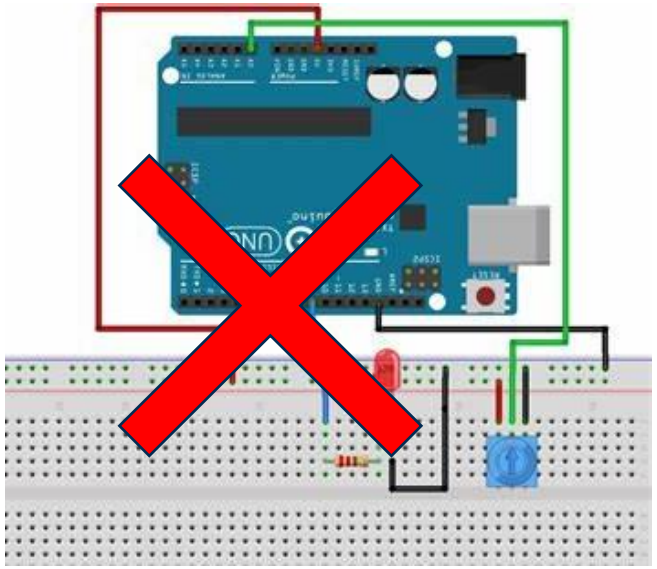
# AT commands

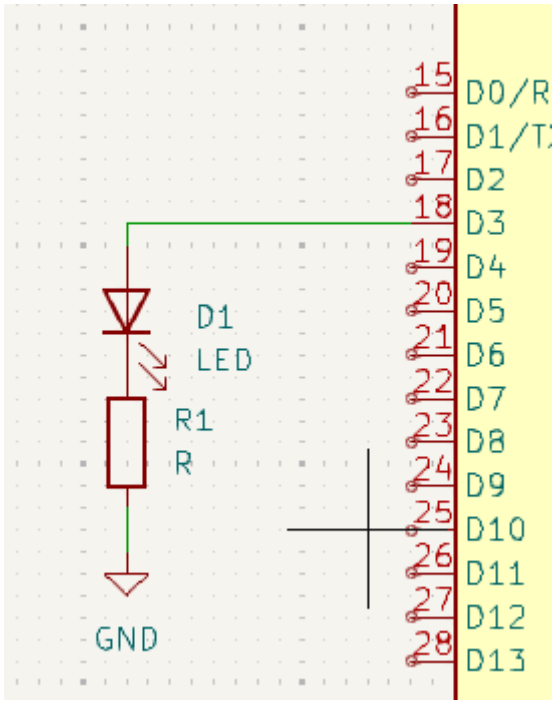
- **AT**
  OK
- **ATH**
- **ATDT…**
- **+++**
- **ATZ**
- **Etc…**

- **More or less standard – rather less**
- **Made for human, not for machines**

# Schematics

- **Clear schematics in your documentation (yes, you DO have a documentation)**
- **Use f.i. Kicad**





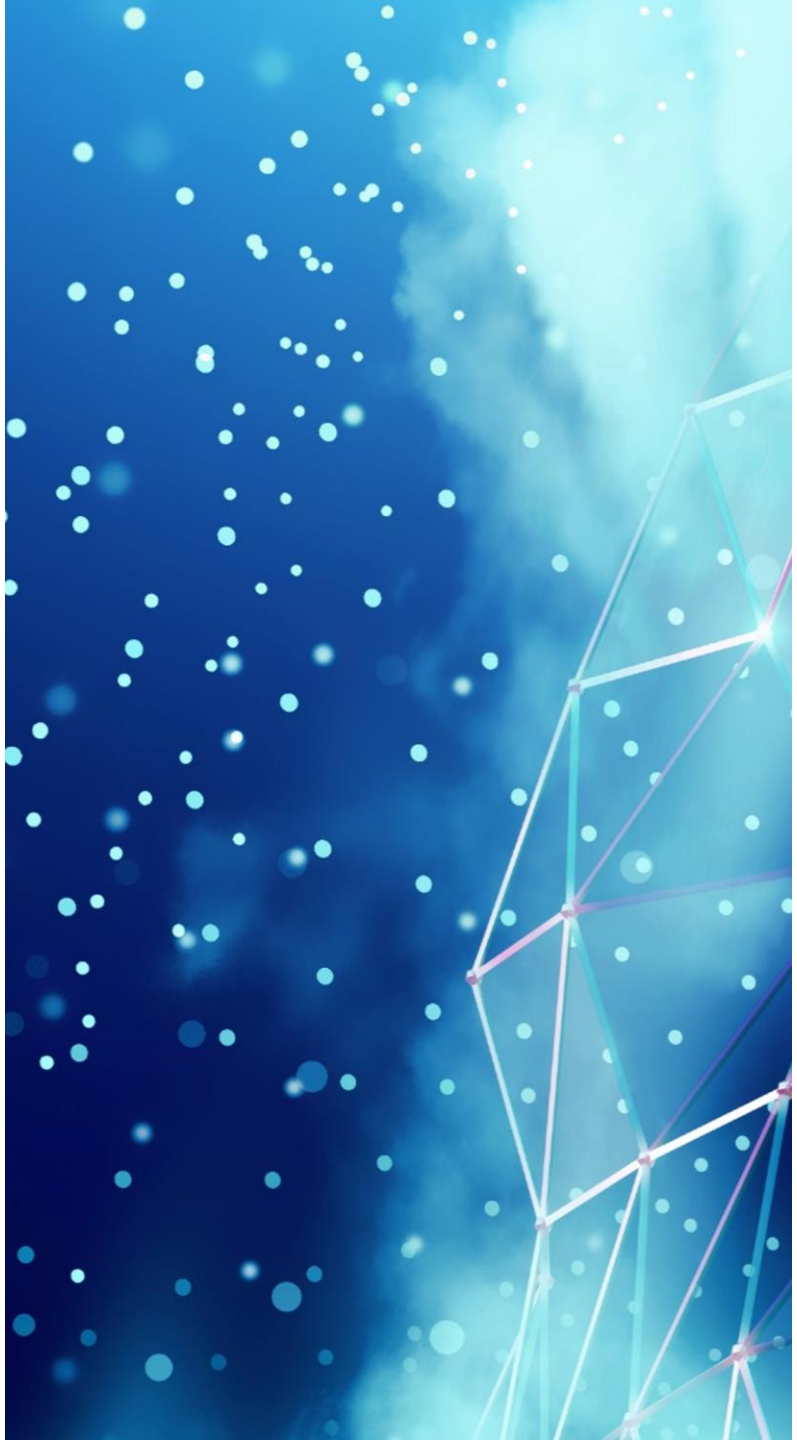
# Don’t burn it

EPITECH. TECHNOLOGY

**(…if possible)**

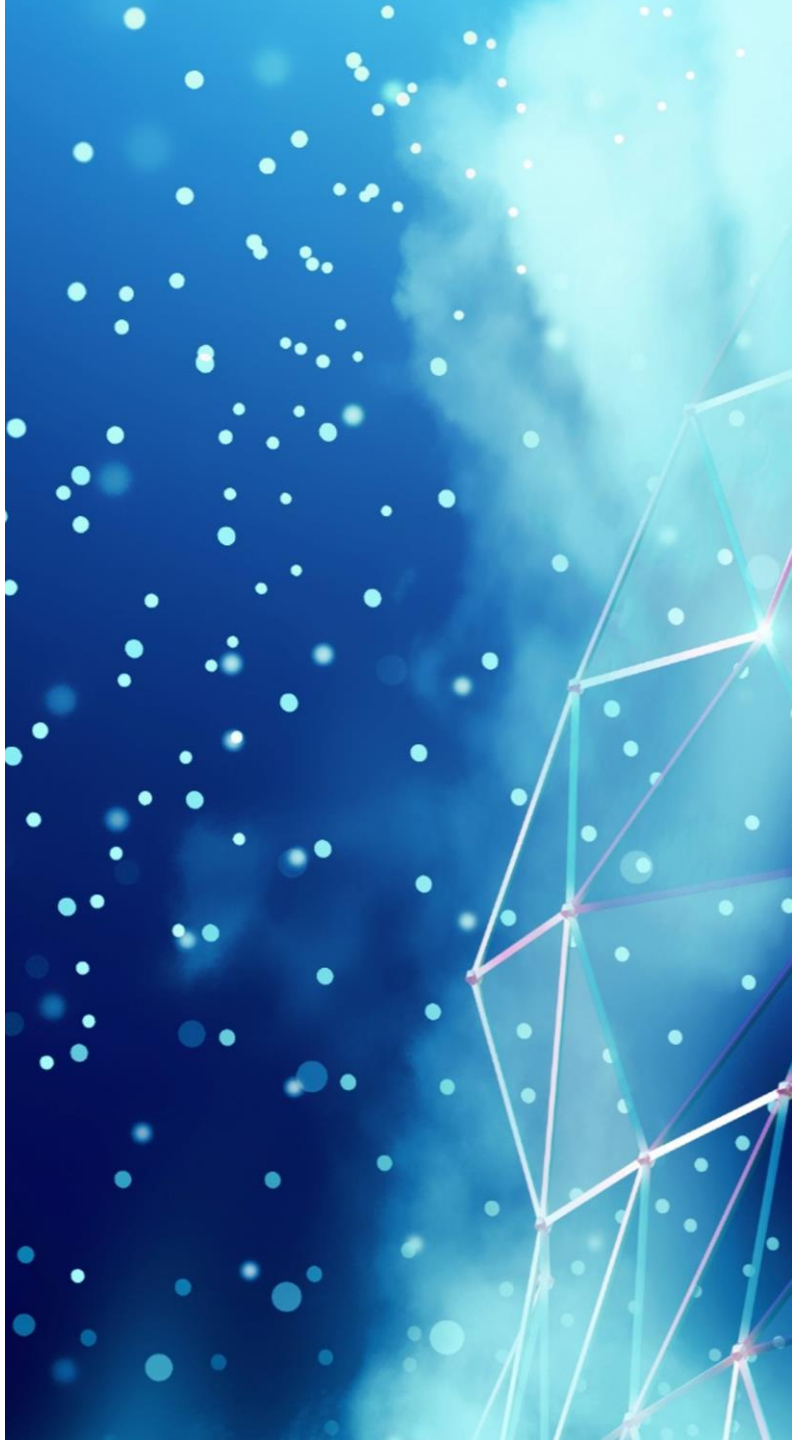
- **Reverse polarity**
- **Board on metalic table**
- **5V on 3.3V module (wifi module)**
- **Inductive load on GPIO !**

- **Low risk !!**
- **Low cost !!**



# Enhance the project

- **Mandatory :**
  - Clean code and generic code for any sensor
- **Normal :**
  - Are sensor data noisy ? Need some low-pass filtering or other preprocessing ?
  - Unit tests ! (cross compilation)
- **Plus :**
  - DC motor actuator (control direction and speed)
  - With feedback for speed control (PID)
- **Even better +++:**
  - RTOS (FreeRTOS, ThreadX) with ESP32 or (if available) STM32




EPITECH.
TECHNOLOGY
Questions ?