



Architecture des systèmes à microprocesseur

Informatique embarquée

- **Chapitre 1 :**
 - Brève introduction aux micro-contrôleurs
 - Entrées-sorties génériques – GPIO
- **Chapitre 2 :** GPIO, Interruptions, timers
- **Chapitre 3 :** BUS de communication
 - Liaison série : UART
- **Chapitre 4 :** Périphériques de conversion
 - Conversion analogique-numérique — CAN
 - Conversion numérique-analogique — CNA



☐ Séances de TP (28h- 7×4h)

- TP 1 : Introduction à l'environnement de développement Keil μ vision
- TP 2 : GPIO
- TP 3 : Interruptions / Scrutation
- TP 4 : Timer
- TP 5 : Liaison Série - DMA
- TP 6 : ADC, DAC

- Mini-projet



□ Règles du jeu

- Notes de DS : DS 1h, DS 2h
- Notes de TP
- Note de TP Contrôle



❑ Les documents

- Les documents papiers seront distribués en temps utile pendant les séances
 - Diapositives feront office de poly et seront distribués en CM
 - Textes de TP
- Documents fortement conseillés pour les séances de TP et projet

❑ Les sources : **tout se trouve en ligne (ou presque) !**

- Site de STMicroelectronics (www.st.com)
- Site de ARM (www.arm.com)
- Documentations annexes :
 - Getting started with STM32 Nucleo board software development tool
 - *Mastering STM32* de Carmine Noviello



- Comprendre les bases liées à l'utilisation d'un micro-contrôleur, et ce qui le différencie d'un micro-processeur,
- Capable de mettre en œuvre une grande partie des périphériques internes/externes nécessaires à un objet connecté,
- Savoir utiliser l'environnement de développement Keil uVision / STM32CubeIDE (compilation, debug,...) pour concevoir/tester/valider votre code





Chapitre 1

1. Micro-processeurs, micro-contrôleurs
2. STM32
3. Outils

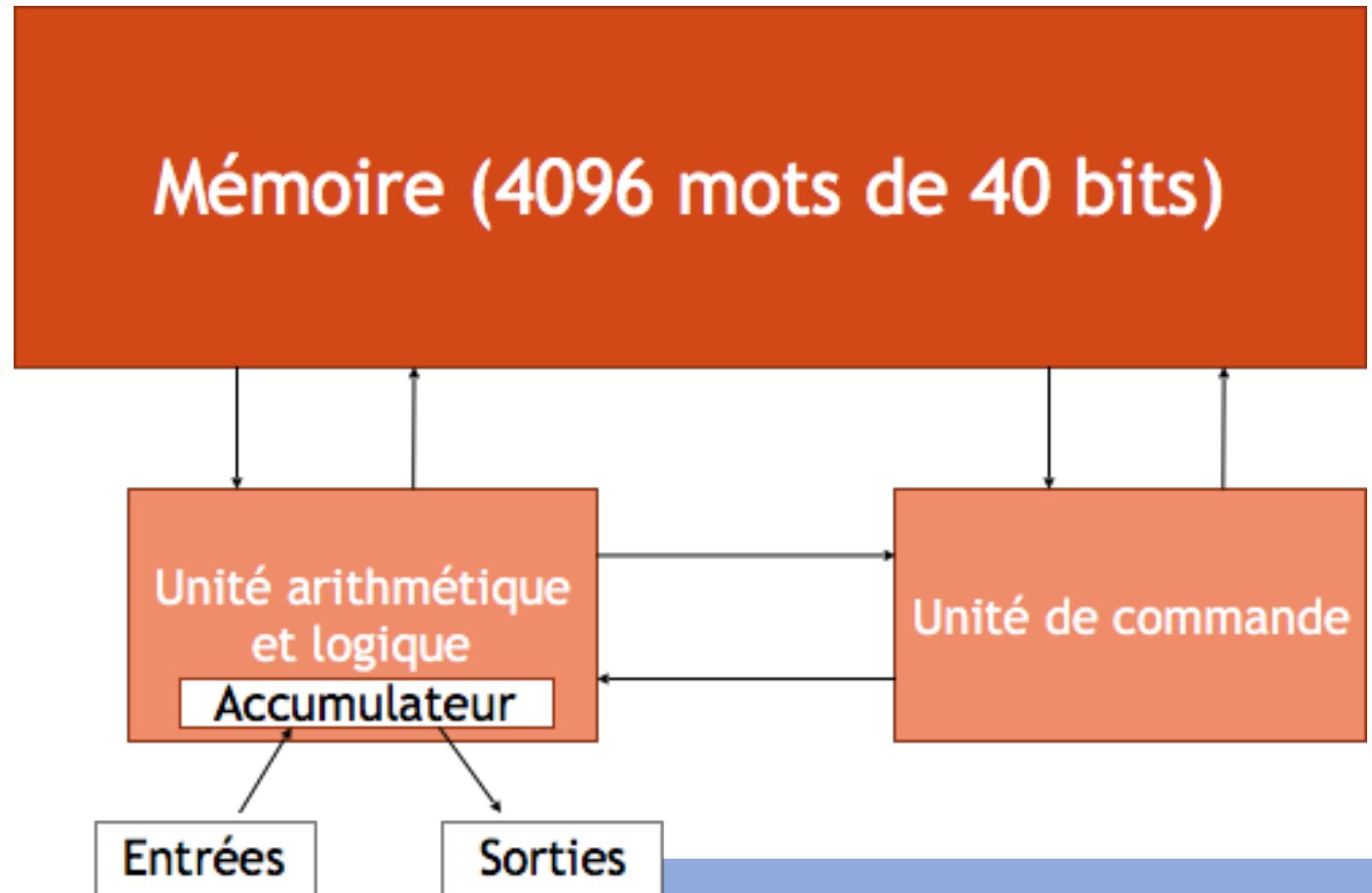


1. Historique

□ Schéma de l'architecture

- 5 parties :

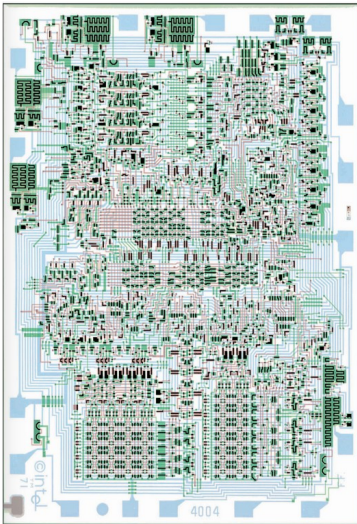
.L'UAL,
.L'unité de
commande,
.La mémoire
.Les entrées
.Les sorties



1. Historique

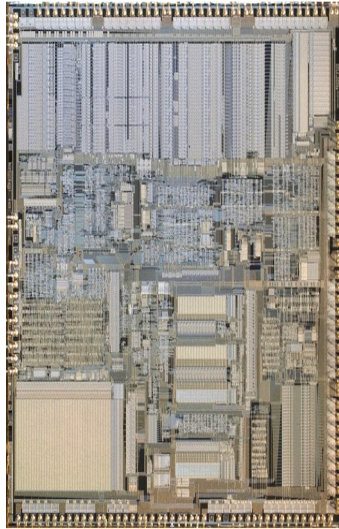
□ Augmentation de la densité d'intégration

1970's
8 bits uP



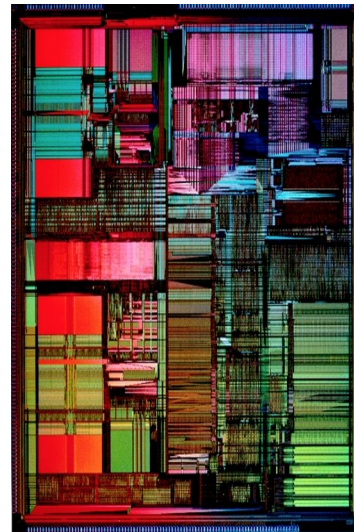
Intel 4004
2300 transistors

1980's
32 bits uP



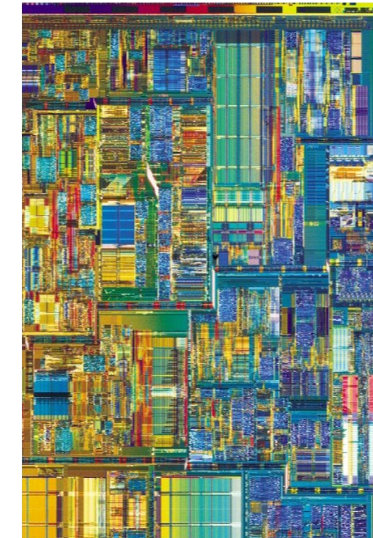
Intel 80386
275k transistors

1990's
32 bits uP



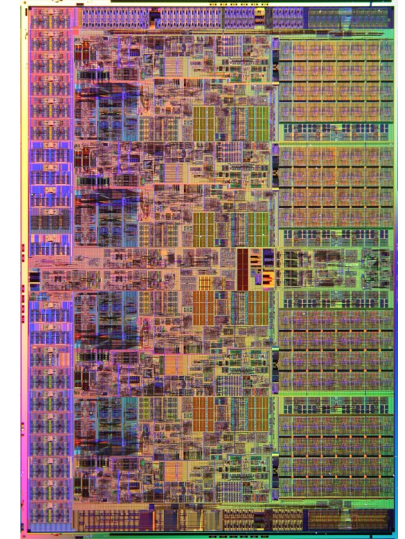
Pentium (1993)
3,1M transistors

2000's
64 bits uP



Pentium IV
[42-184]M transistors

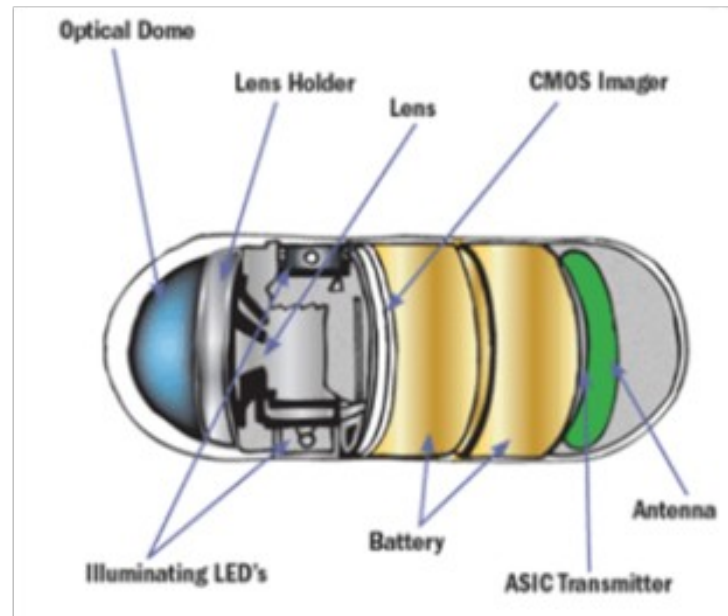
2010's
64 bits uP



Core i7
[731M et + (>Milliard)
transistors

Aujourd'hui ?

- Processeurs fabriqués en masse et à faible coût
- Ils sont omniprésents dans nos vies et ils sont le cœur des objets connectés pour « **l'internet des objets** »



□ Fonctionnalités d'un microprocesseur

- Acquisition de signaux issus du monde 'extérieur'
 - Convertisseur analogique-numérique
- Traitement des données
 - processeur
- Envoie de commandes extérieures
 - GPIO, convertisseur numérique-analogique
- Communication
 - Liaison I2C, SPI, USART, USB, Ethernet, ...
- Gestion de périphériques internes
 - Mémoires, Timer, ...

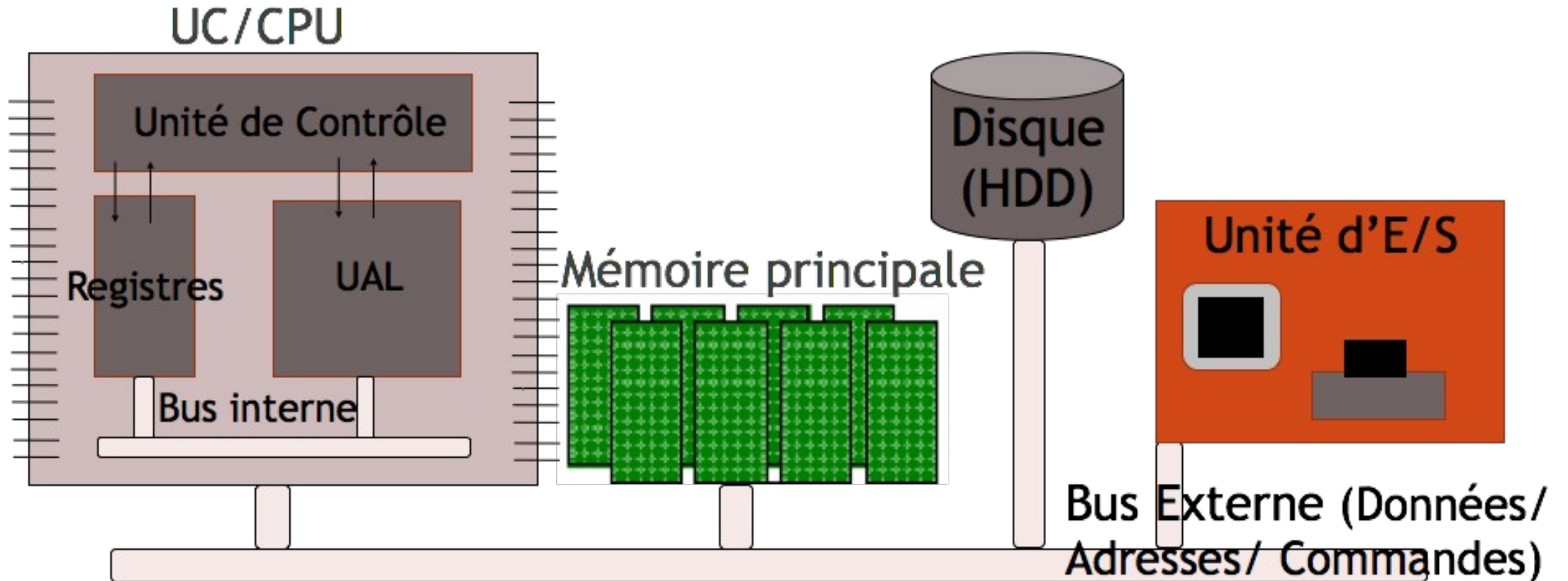
2. Taxonomie d'un processeur

2) Taxonomie d'un processeur

- Architecture actuelle de l'ordinateur (aspects externe/interne)
- Définition des différents composants élémentaires (ALU, registre, mémoire, bus)

3. Architecture

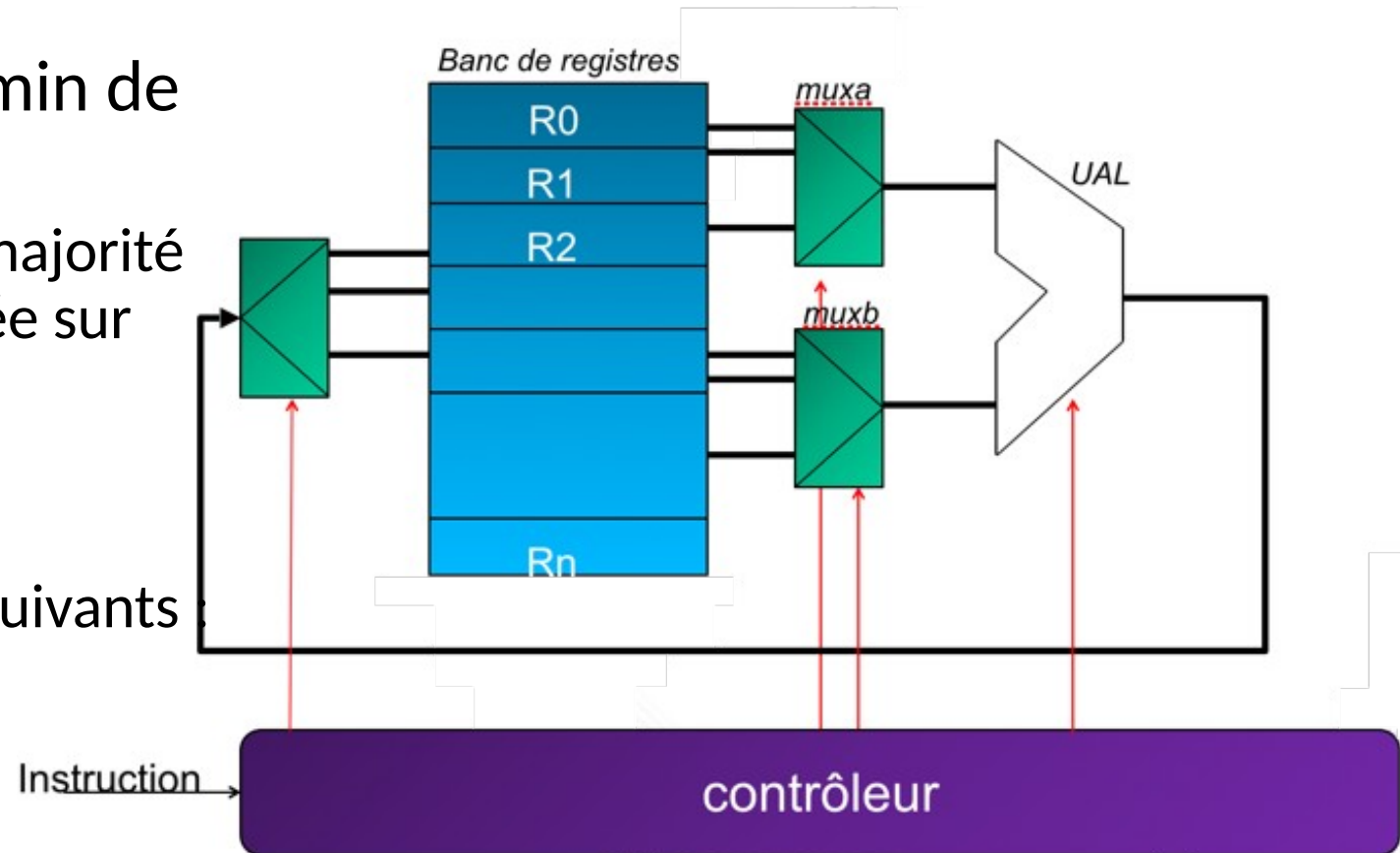
- ❑ Architecture actuelle de l'ordinateur : aspect externe



2. Architecture

□ Aspect interne du microprocesseur : le chemin de données

- Depuis les années 80, la majorité des architectures est basée sur une architecture à chargement/rangement
- Utilisation des éléments suivants :
 - banc de registres
 - UAL
 - multiplexeurs



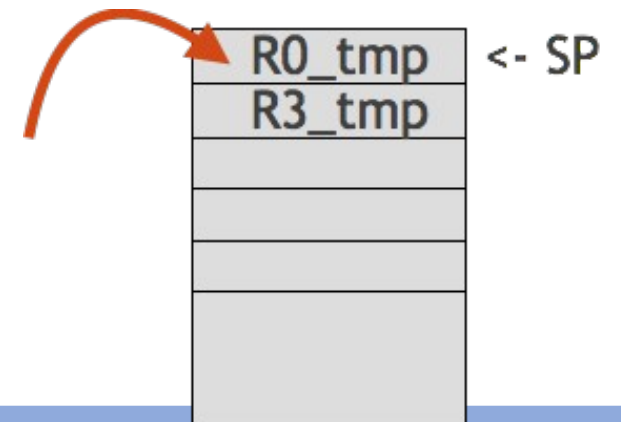
2. La pile

□ La pile

- La pile est un emplacement mémoire dédié pour la sauvegarde de l'état des registres
- Cet emplacement mémoire doit être connu du processeur

➤ On utilise un pointeur SP (Stack Pointeur) qui stocke l'adresse du dernier mot stocké

- La pile fonctionne comme une mémoire de type LIFO
- Elle est généralement créée en RAM
- Le 'sommet' de la pile est repéré par le pointeur de pile et évolue au fil des accès



3. Famille de processeur

3. Famille de processeur

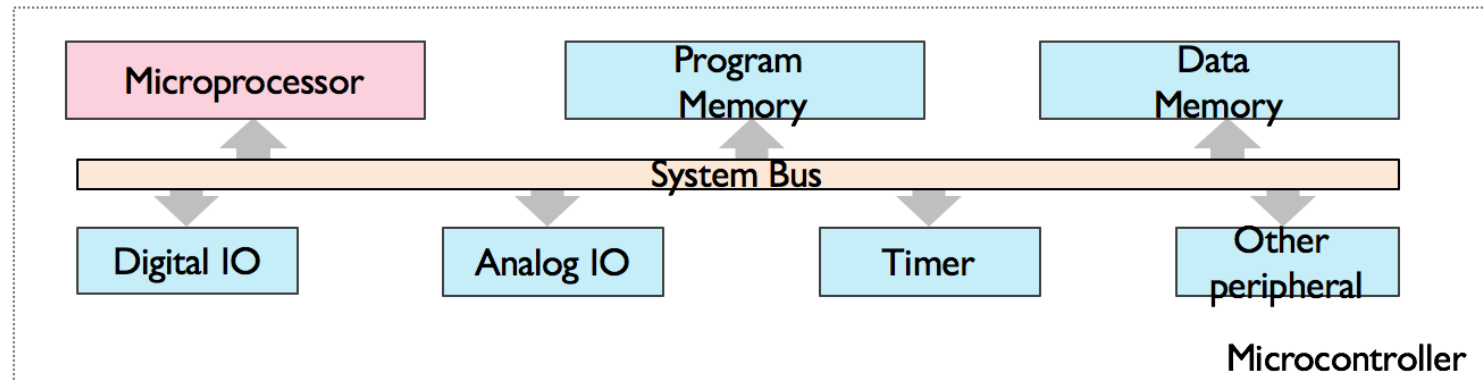
- Microcontrôleur vs Microprocesseur
- RISC vs CISC

3. Famille de processeur

❑ Microcontrôleur vs Microprocesseur

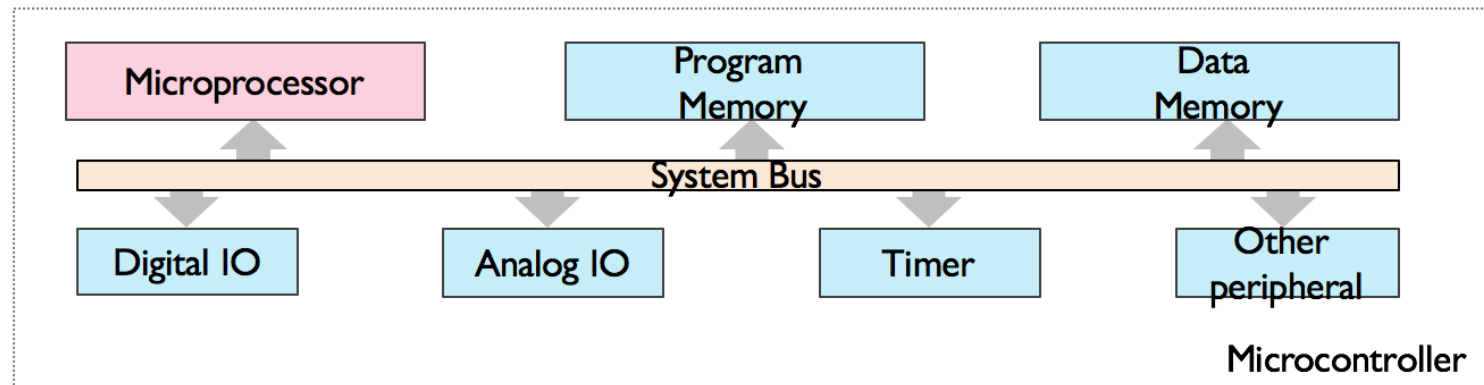
- Microcontrôleur :

- C'est un circuit intégré qui contient un microprocesseur, des mémoires, des entrées/sorties ainsi que des périphériques additionnels. Le microcontrôleur contient tous les éléments d'un mini-ordinateur sur une même puce.



3. Famille de processeur

- Microprocesseur :
 - élément clé permettant d'exécuter les instructions d'un programme. A noter qu'un microprocesseur seul ne peut fonctionner. Il a besoin de mémoires (ROM/RAM) ainsi que de périphériques extérieurs.



3. Famille de processeur

□ Architecture RISC vs CISC

- **CISC : *Complex Instruction Set Computers***

Jeu étendu d'instructions complexes où chaque instruction peut effectuer plusieurs opérations élémentaires (chargement mémoire, opération arithmétique,...). Le temps d'exécution varie en fonction de la taille et le format de l'instruction.

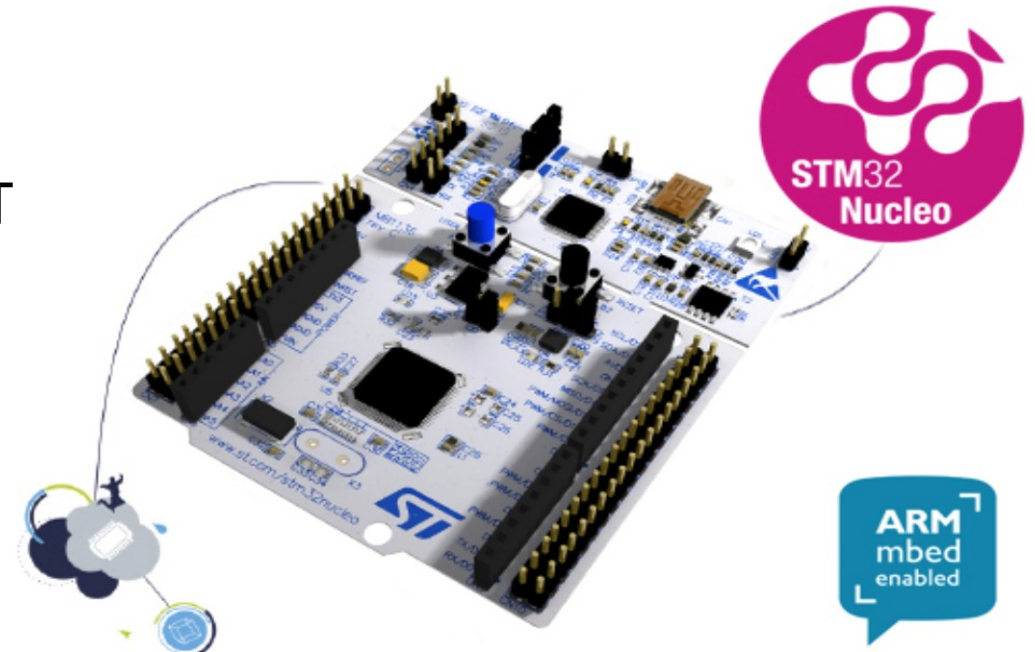
- **RISC : *Reduced Instruction Set Computers***

Jeu d'instruction réduit avec uniquement des instructions simples pouvant effectuer une opération élémentaire. Chaque instruction prend en général 1 cycle d'horloge et elles sont codées avec le même nombre de bits.

3. Famille de processeur

□ Pour les travaux pratiques, la plateforme que nous utiliserons est la **nucleo board 64** qui contient un microcontrôleur **STM32F4xx** de STMicroelectronics (ST)

- Architecture du cœur : Cortex-M4 d'ARM
- Périphériques et plateforme conçus par ST
 - Connecteurs Arduino, ST Morpho
 - 3 Leds, 2 boutons poussoirs, ...
- Programmation/Debug via ST-LINK/V2-1
- Compilateur en ligne Mbed possible
- Prix ≈ 12\$



3. Famille de processeur

❑ Environnements possibles de développement

- **µVision de Keil**

- Utilisé en TP

- Version gratuite mais limitée en taille de code (32ko)

- Possède un très bon débbuger !

- Coocox

- IAR

- Compilateur en ligne avec Mbed

❑ OS disponibles

- PowerPac (IAR)

- FreeRTOS

- uClinus sur Cortex-M4

- ARTX-ARM (Keil)

- C/OS-333 (Micrium)

- embOS (Segger)

3. Famille de processeur

ARM

- Conçoit et vend des architectures de processeurs RISC depuis les années 80 (pas de circuit) :

« An ARM architecture is a set of specifications regarding the instruction set, the execution model, the memory organization and layout, the instruction cycles and more, which describes precisely a machine that will implement said architecture » [Mastering STM32, Carmine Noviello]

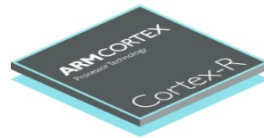
ARM Cortex Family



Cortex-A

Highest performance
Optimized for rich operating systems

Application



Cortex-R

Fast response
Optimized for high-performance, hard real-time applications

Real-Time



Cortex-M

Smallest/lowest power
Optimized for discrete processing and microcontroller

eMbedded



SecurCore

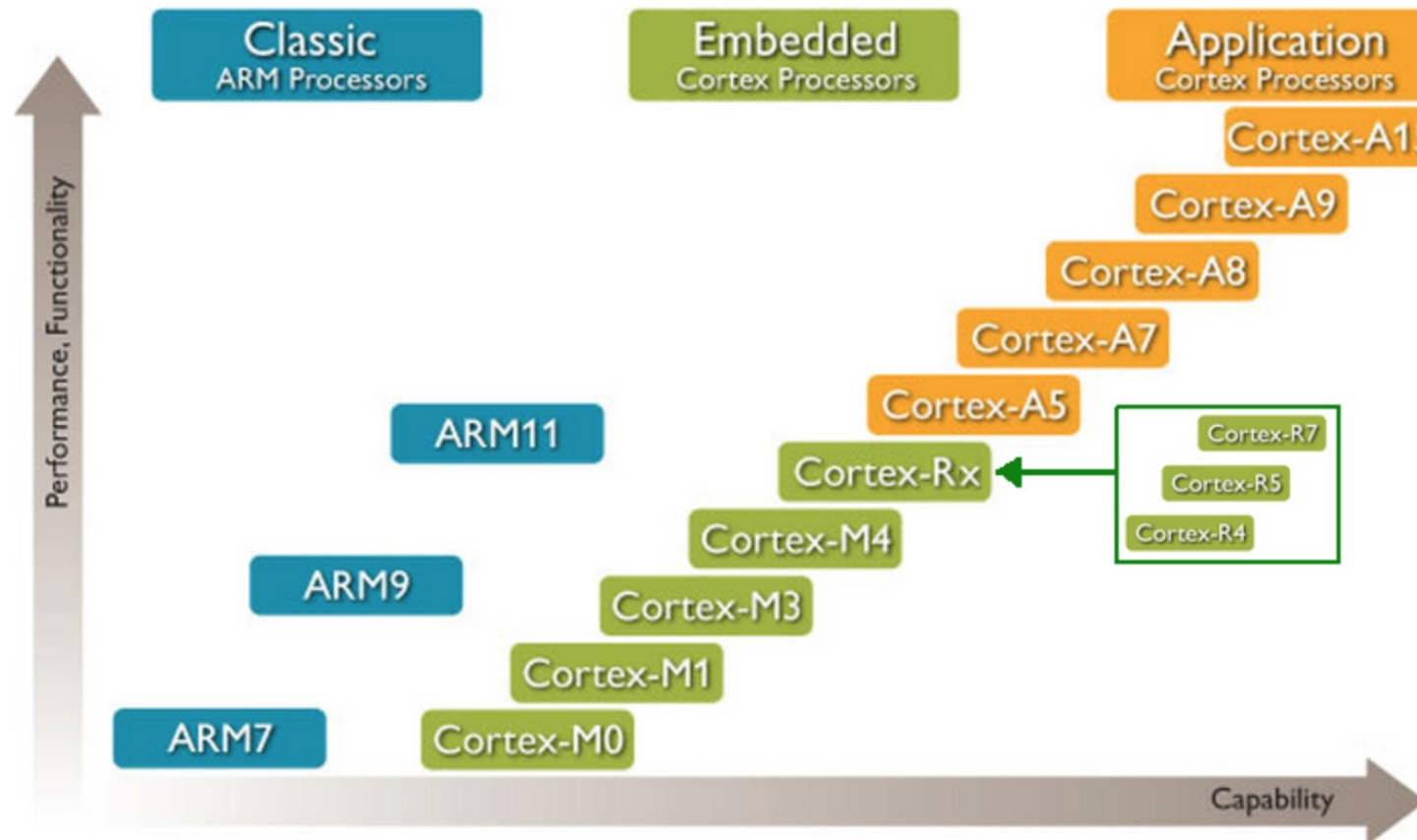
Tamper resistant
Optimized for security applications

Security

[source :
[arm.com](https://www.arm.com)]

3. Famille de processeur

□ La famille Cortex d'ARM

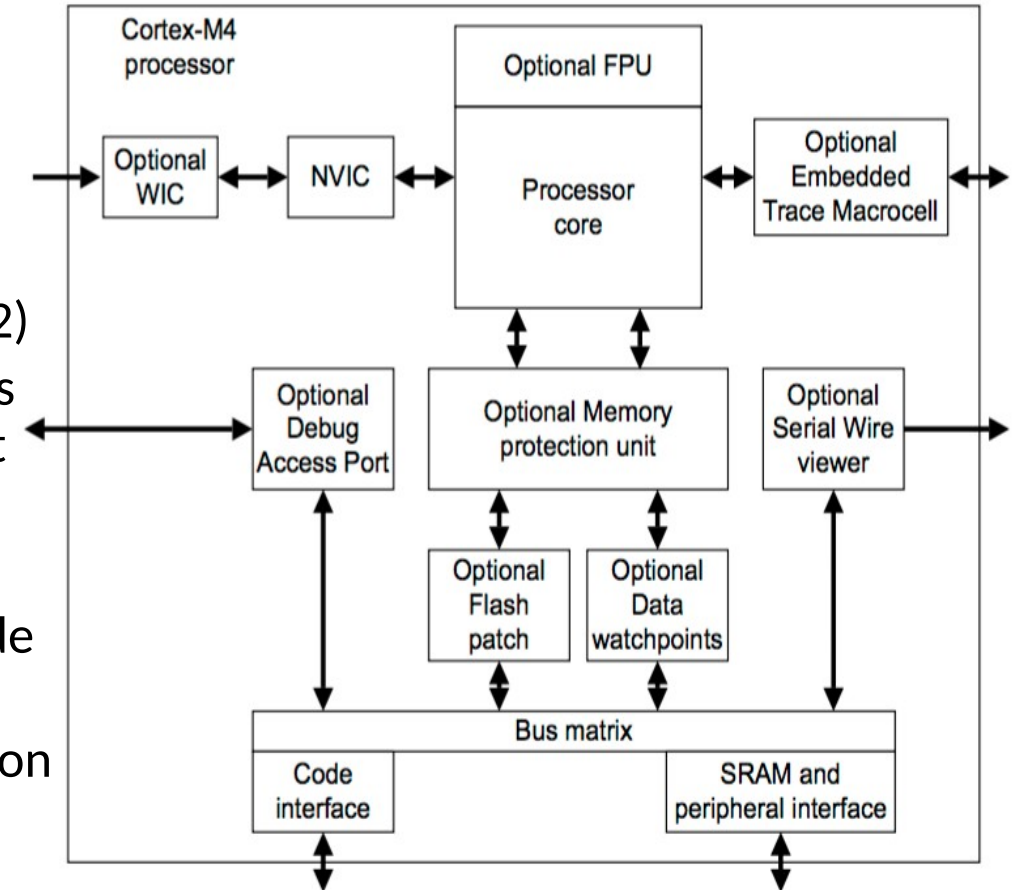


<http://www.emcu.it/>

3. Cortex M4

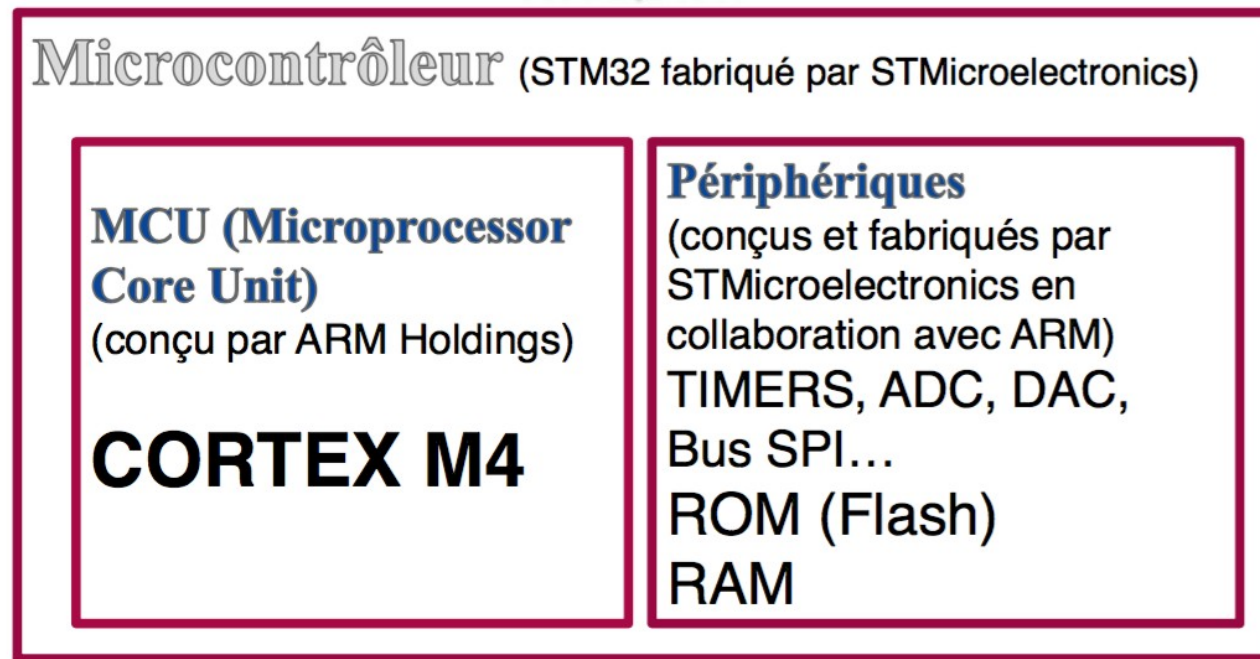
□ Cortex-M4

- Architecture de processeur RISC 32 bits
- Architecture de Harvard (bus de données et d'instructions ≠),
- 3 étages de pipeline
- 2 Jeux d'instructions sur 32/16 bits (ARM/Thumb2)
- Cortex-M4F : contient une unité de traitement des nombres flottants (FPU) + capacités de traitement (DSP)
- Contrôleur d'interruption (NVIC) à faible latence de traitement
- Une unité de protection de mémoire (MPU) -option
- Une solution faible coût de debug



3. STM32F4xx

□ Positionnement du Core ARM dans le microcontrôleur STM32

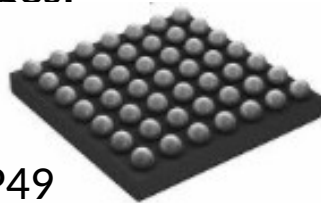


3. STM32F4xx

Principales caractéristiques du STM32F446RE

- Core ARM Cortex-M4: Fréquence maximale de 84MHz
- Mémoires : up to 512Ko Flash // up to 96Ko SRAM
- Operating Voltage : 1.7 to 3.6V
- Plusieurs sources d'horloge :
 - 4-26 MHz crystal oscillator
 - Internal 16MHz RC (default CPU clock on reset)
 - 32kHz, ...
- Consommation: Run (146uA/MHz), StandBy (2.4uA) + Deep power down mode
- Up to 11 Timers : génération de PWM,...
- Up to 81 I/O ports (5V-tolerant)
- Interface de communication (I2C, USART, SPI, USB...)
- Les types de packages:

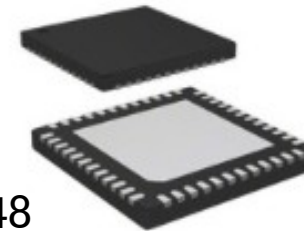
WLCSP49



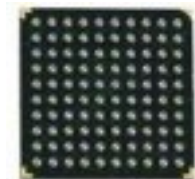
LQFP64/100



UFQFPN48

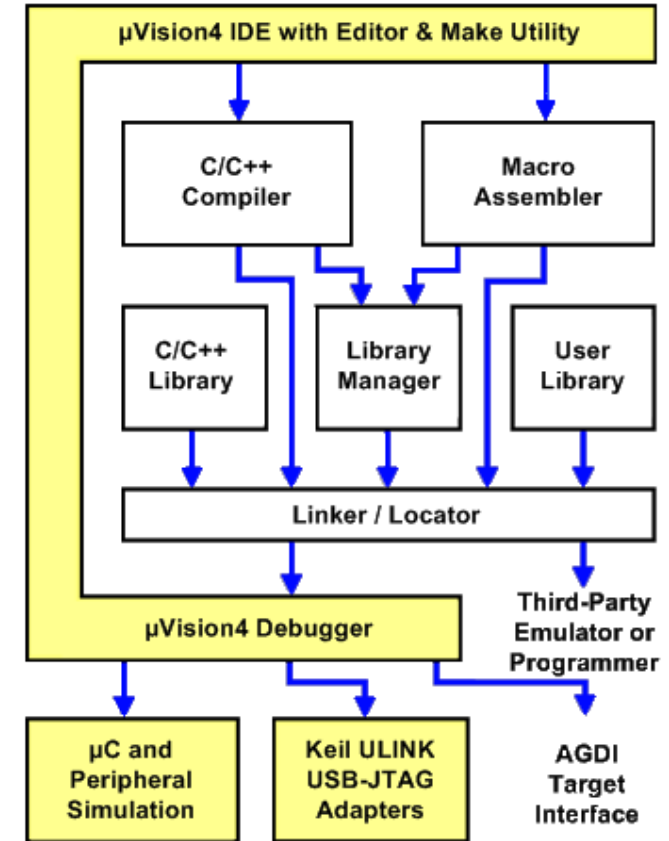


UFBGA100



❑ Cycle de développement logiciel

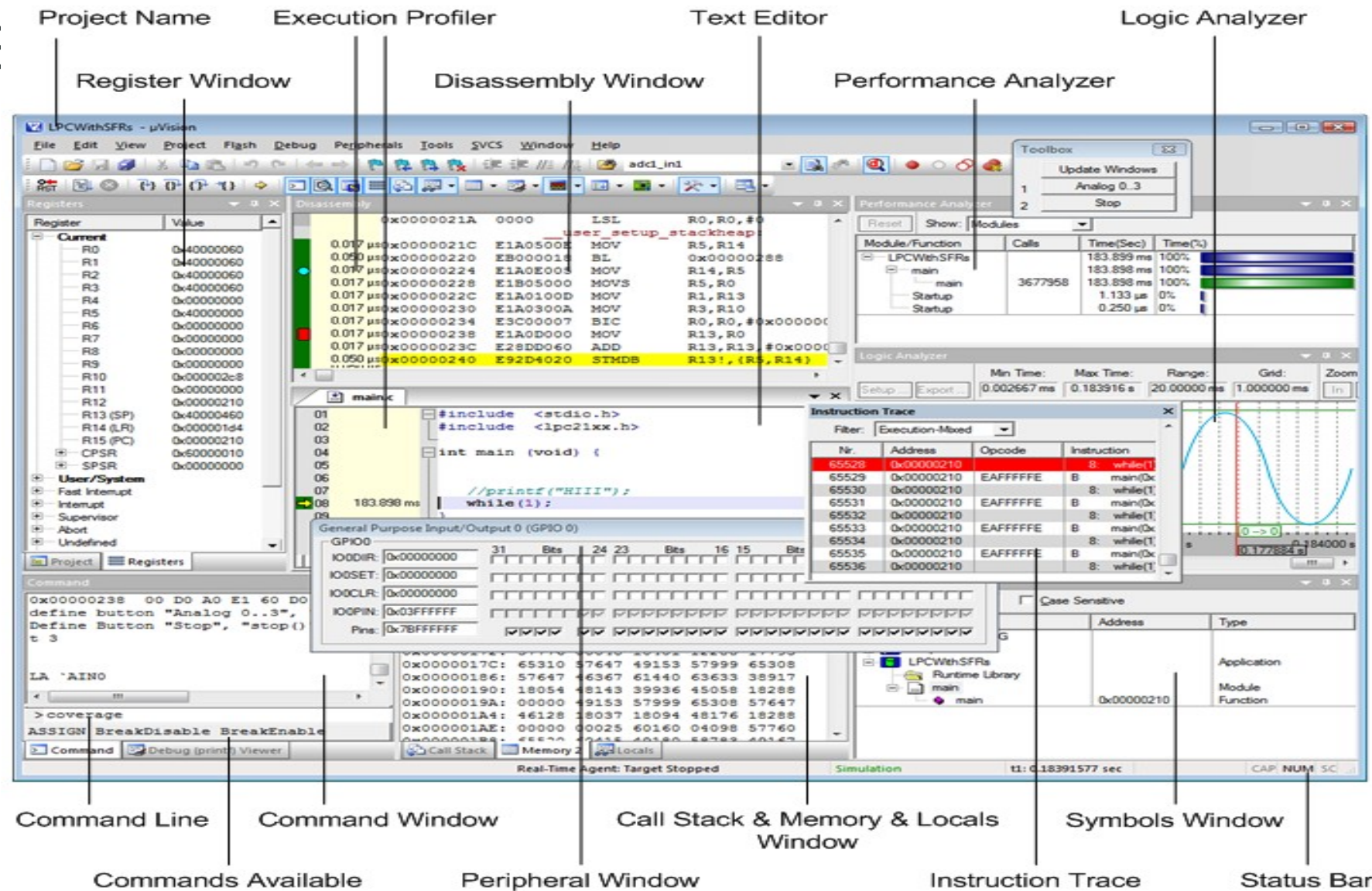
- L'IDE μ Vision intègre:
 - Project manager (configuration wizard..)
 - μ Vision Editor (*completion*, vérification de syntaxe dynamique..)
- C/C++ Compilateur et assembleur de macro
 - Fichiers sources -> fichiers objets (code machine)
- Library manager
 - Permet la création de fichiers objets
- Linker / Editeur de lien
 - Création de l'exécutable (.hex) à partir des fichiers objets et des bibliothèques pré-compilées
 - code fixé en mémoire / cible précise
- Debugger
 - Incluant un simulateur (microprocesseur + périphériques)



keil.com



IDE

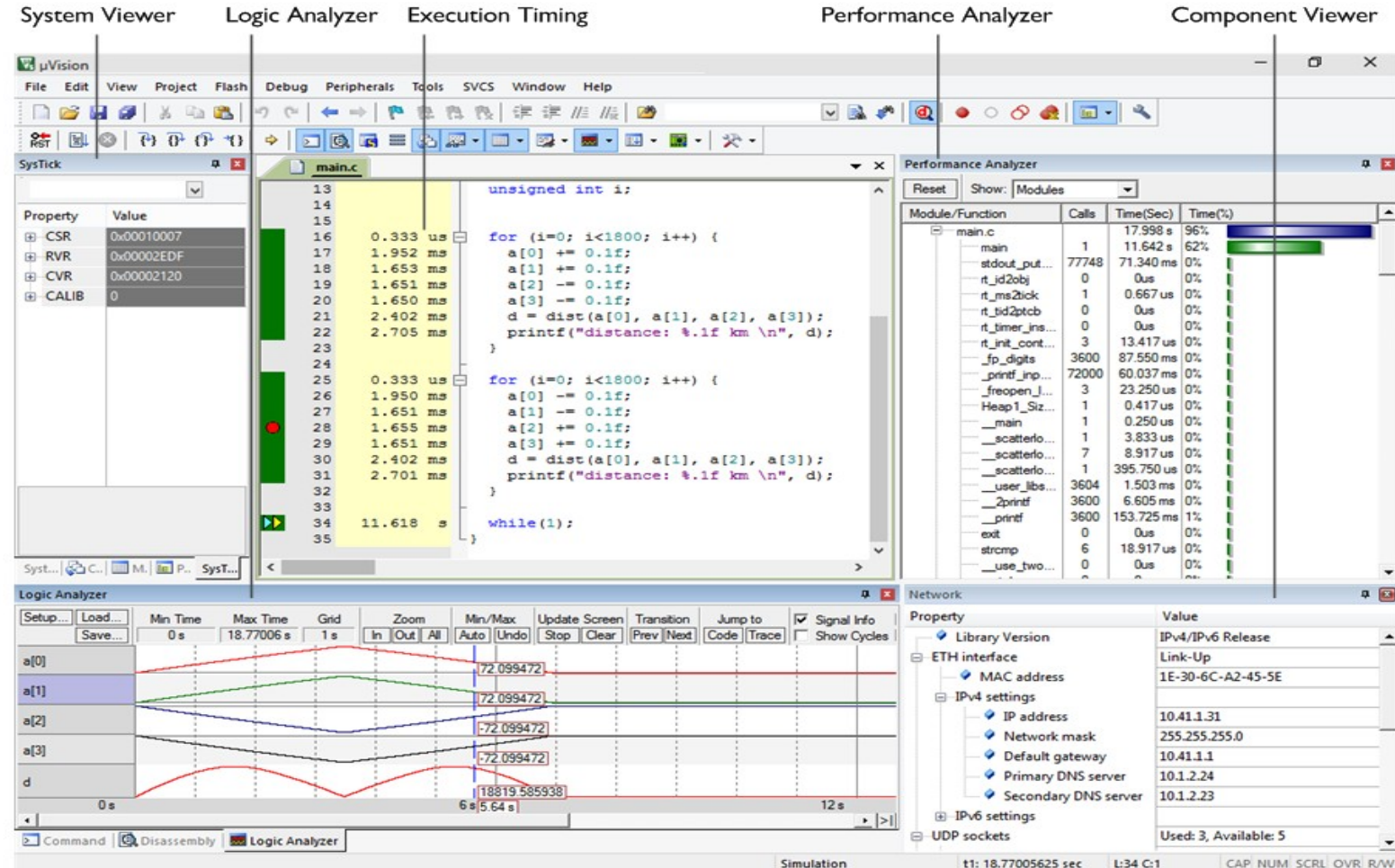


keil.com

3. Outils de développement SW

Le debugger

- Permet de tester, vérifier et optimiser le code de l'application
- 2 modes de fonctionnement :
 - **Simulator Mode** : vérification purement logicielle -> Pas besoin de cible matérielle
 - **Target Mode** : vérification via connexion avec la cible matérielle
- Breakpoints/Watch supportés



□ En résumé

- Evolution considérable des microprocesseurs depuis les dernières décennies
- Omniprésence du cœur ARM dans les microcontrôleurs de nouvelle génération
 - Utilisation en TP du μ C STM32F446RE basé sur le cœur ARM Cortex-M4
 - Utilisation de l'environnement de développement μ Vision de Kiel
- Pour les séances suivantes, on abordera le mécanisme d'interruption ainsi que le contrôle de périphériques essentiels d'un microcontrôleur, pour mettre un œuvre un objet connecté

- ❑ Pour commencer à jouer : *STM32CubeMX* tool et *STM32CubeIDE*
- Utilitaire graphique (gratuit) permettant de générer le code d'initialisation du processeur, des périphériques (internes/externes), de l'arbre d'horloge...
 - Il génère aussi le projet, incluant la librairie de drivers HAL (Hardware Abstraction Layer)

