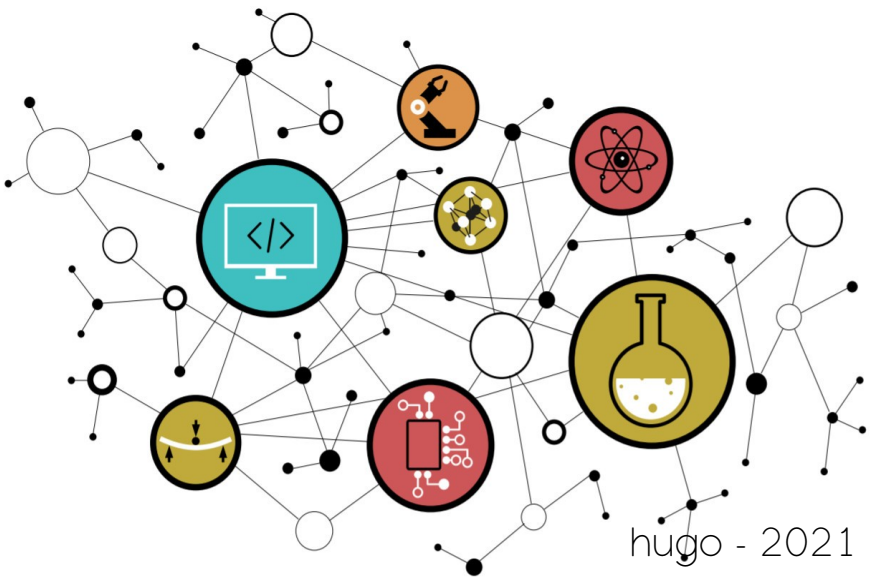
TRAVAUX PRATIQUES

PRÉLUDE



hugo - 2021

SOMMAIRE

La trame de TP minimale que nous considérons comme être les compétences minimales à acquérir afin d'accéder aux métiers de base du domaine suit le séquencement suivant : chapitres 1, 2, 3, 4, 5 et 8. Le reste de la trame ne sera pas évalué et représente une extension au jeu de compétences minimales relatif au domaine en cours d'étude (* devant chapitres facultatifs voire complémentaires). Libre à vous d'aller plus loin selon votre temps disponible et votre volonté de mieux comprendre et maîtriser ce domaine !

1. PRÉLUDE

1.1. Présentation des Systèmes Embarqués
1.2. Objectifs pédagogiques
1.3. Quelques ressources internet

2. MODULE DE BROCHE GPIO ET ASSEMBLEUR PIC18

2.1. Introduction : *Module GPIO ou General Purpose Input Output*
2.2. Introduction : *Module GPIO sur PIC18*
2.3. Introduction : *Configuration des GPIO sur PIC18*
2.4. Introduction : *Assembleur ou langage d'assemblage PIC18*
2.5. My first MPLABX project from scratch
2.6. Analyse assembleur et debug
2.7. BSP et fonctions pilotes C/ASM
2.8. Gestion des boutons poussoirs
2.9. Délais logiciel en assembleur

3. MODULE DE COMPTAGE TIMER ET GESTION DES INTERRUPTIONS

3.1. Introduction : *Interruption matérielle*
3.2. Introduction : *Source et requête d'interruption IRQ*
3.3. Introduction : *Logique et démasquage d'interruption*
3.4. Introduction : *Vecteur d'interruption*
3.5. Introduction : *Fonction d'interruption ISR*
3.6. Introduction : *Gestion des interruptions sur PIC18*
3.7. Introduction : *Gestion du RESET sur PIC18*
3.8. Introduction : *Module périphérique de comptage Timer*
3.9. Introduction : *Module périphérique Timer0 sur PIC18*
3.10. Introduction : *Configuration du Timer0 sur PC18*
3.11. Configuration du Timer0 et interruption
3.12. Analyse assembleur et commutation de contexte
3.13. Mise en veille du CPU

4. MODULE DE COMMUNICATION UART ET LIAISON SÉRIE

4.1. Introduction : *Protocole de communication d'une liaison série asynchrone*
4.2. Introduction : *Module périphérique UART*
4.3. Introduction : *Norme RS232*
4.4. Introduction : *Module périphérique UART sur PIC18*
4.5. Introduction : *Configuration du module UART sur PIC18*
4.6. Module périphérique UART1 en transmission
4.7. Terminal de communication série sur ordinateur
4.8. Transmission de chaînes de caractères
4.9. Module périphérique UART1 en réception
4.10. Buffer circulaire de réception
4.11. Contrôle de flux logiciel
4.12. Réception de chaînes de caractères
4.13. Module périphérique UART2
4.14. Bridge de communication UART1 vers UART2

5. MODULE AUDIO BLUETOOTH EXTERNE

- 5.1. Introduction : *Bluetooth*
- 5.2. Configuration du module Audio Bluetooth RN52

* 6. MODULE DE COMMUNICATION I2C ET AFFICHEUR LCD

* 7. MODULE DE CONVERSION ADC

8. CONCEPTION D'UNE APPLICATION ET ORDONNANCEMENT

- 8.1. Introduction : *Application, ordonnancement et philosophie Unix*
- 8.2. Conception et ordonnancement de l'application
- 8.3. Cahier des charges du POC (Proof Of Concept)
- 8.4. Développement du POC (Proof Of Concept)
- 8.5. Evolutions et améliorations

* 9. DOCUMENTATION TECHNIQUE ET LIVRABLES

- 9.1. Introduction : *Livrables et documentation technique*
- 9.2. Doxygen et documentation d'une bibliothèque
- 9.3. Documentation technique

SEQUENCEMENT PÉDAGOGIQUE

Le plan et le séquencement de la trame de Travaux Pratiques correspondent au chemin proposé afin d'atteindre les objectifs pédagogiques fixés. Ces objectifs ont été choisis au regard des attentes et exigences demandées par les marchés de l'industrie du logiciel et des couches basses des systèmes : systèmes embarqués, systèmes temps réel, développement de systèmes d'exploitation, développement de bibliothèques spécialisées, développement de chaînes de compilation, attaque et sécurité des systèmes, etc, tous des métiers dans divers domaines actuellement exercés par certains de nos anciens élèves. Il s'agit d'un séquencement conseillé qui n'a en aucune façon volonté à être imposé (étudiant comme enseignant encadrant). Pour se dérouler sous les meilleurs hospices, il serait cependant préférable dès le début de l'enseignement de rythmer 1 à 2 heures de travail personnel à la maison en dehors des séances en présentiels avec enseignant. Voici une proposition de séquencement (2h par créneau de TP). Lire à minima l'introduction de chaque TP avant de venir en séance :

2. Module de broche GPIO et assembleur PIC18

- En session de TP : *2.5 / 2.6 (TP n°1) 2.7 / 2.8 (TP n°2) 2.9 (TP n°3)*
- Hors session de TP : *Lire le document prélude et introduction au TP (avant TP n°1 . 2.6 (avant TP n°2) proposition pour 2.9 (avant TP n°3)*

3. Module de comptage Timer et gestion des interruptions

- En session de TP : *3.11 / 3.12 / 3.13 (TP n°4 et n°5)*
- Hors session de TP : *Lire introduction au TP n°3 et proposition pour 3.11 (avant TP n°4)*

4. Module de communication UART et liaison série

- En session de TP : *4.5 / 4.7 (TP n°6) 4.8 / 4.9 (TP n°7) 4.10 / 4.11 (TP n°8) 4.11/ 4.12 (TP n°9) 4.13/ 4.14 (TP n°10)*
- Hors session de TP : *Lire introduction au TP n°4 et proposition pour 4.5 (avant TP n°6)*

5. Module Audio Bluetooth externe

- En session de TP : *5.2 (TP n°11 et n°12)*
- Hors session de TP : *Lire introduction au TP n°5 et proposition pour 5.2 (avant TP n°11)*

8. Conception d'une Application et ordonnancement

- En session de TP : *8.2 / 8.3 / 8.4 / 8.5 / etc (TP n°13 jusqu'à la fin)*
- Hors session de TP : *Lire introduction au TP n°8 et proposition pour 8.2 (avant TP n°13)*

9. Documentation et livrables

- Hors session de TP : *Lire introduction au TP n° 9 et 9.2/ 9.3*

PRÉLUDE

1. PRÉLUDE

1.1. Présentation des Systèmes Embarqués

Un système embarqué peut-être vu comme le système (ensemble physique, électronique et informatique) embarqué dans le produit (souris, clavier, montre, voiture, fusée spatiale, carte bancaire, etc la liste est très longue). A l'image des produits, la conception d'un système embarqué peut offrir de multiples facettes. Il peut être communicant (téléphone mobile, Ebooks, montre connectée, IOT, etc) en utilisant divers protocoles de communication sans fil ou filaire (WIFI, Bluetooth, 2/3/4G, LORA, Ethernet, USB, etc), autonome sur réserve par stockage d'énergie électrique (tablette, souris, voiture, etc), portable dans la main (carte bancaire, clé de voiture, etc), faiblement encombrant (lecteur MP3, carte monétique, etc) comme très encombrant (fusée spatiale, avion, voiture, etc). Un système embarqué peut être vu comme l'ensemble des sous composantes ou sous systèmes suivants. Ce ''tout'' forme un système embarqué complet de stockage, d'échange et de traitement de l'information :

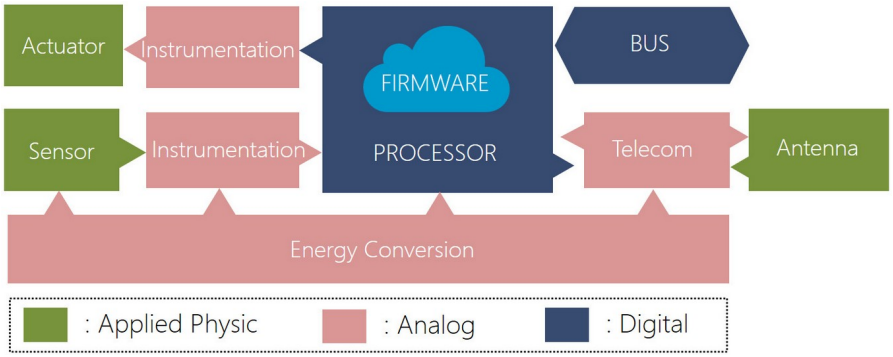
- *Système Physique (hardware/matériel)* : Interfaces avec l'environnement extérieur au produit (capteurs, actionneurs et antennes). Ponts entre l'environnement physique et le périmètre d'action du produit répondant à un besoin (capteurs de température, pression, humidité, luminosité, bouton poussoir, buzzer, LED, antennes radio, WIFI, Bluetooth, etc)

- *Système Électronique Analogique (hardware/matériel)* : Conditionnement et mise en forme des signaux de mesure et de contrôle (chaîne d'instrumentation) ainsi que des chaînes de communication (wire/filaire et wireless/sans fil).

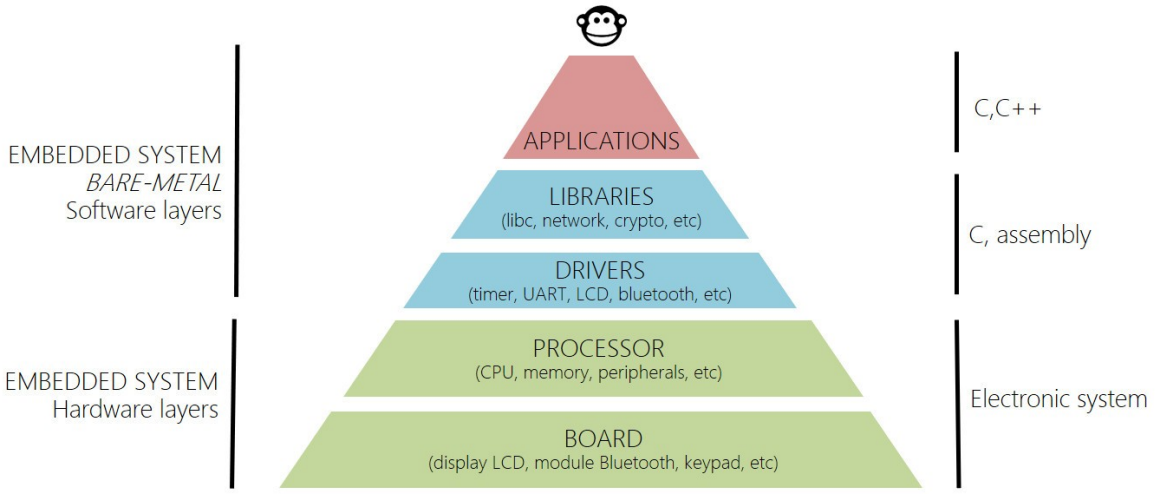
- *Système Électronique de Puissance (hardware/matériel)* : Mise en forme et dimensionnement des entrées électriques d'énergie (redresseur, hacheurs flyback, forward, etc). Stockage de l'énergie électrique sur système embarqué autonome (batterie).

- *Système Électronique Numérique (hardware/matériel)* : Stockage numérique des données (mémoire donnée) et des programmes (mémoire programme). Traitement des données par exécution du programme puis mise en forme de l'information (processeur). Interfaces numériques de communication et d'échange avec l'extérieur (fonctions périphériques)

- *Système Informatique (software/logiciel et firmware/micrologiciel)* : Couche système (système d'exploitation du matériel), couche bibliothèque (utilitaires et fonctionnalités pour les applications) et couche applicative (missions de supervision du système appliquées à des besoins spécifiques voire problématiques de calcul)

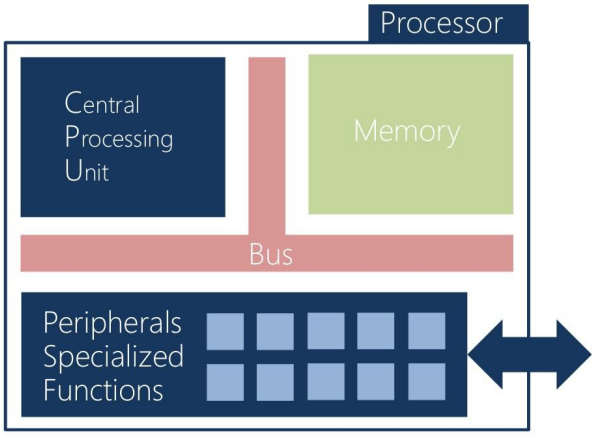


Au cœur de la très grande majorité des produits électroniques actuels se trouvent enfouis un (souris, montre, carte monétique, etc) voire plusieurs processeurs (voiture, avion, etc). Un système embarqué, peut d'ailleurs interagir avec plusieurs sous systèmes embarqués (un ordinateur et une souris par exemple). Différentes familles de processeurs cohabitent sur le marché (MCU, GPP/MPU, AP/MPU, DSP, MPPA, FPGA, GPU, etc), chacune répondant à des besoins et exigences (application/supervision ou calcul/algorithmique, consommation, coût, encombrement, performance, etc). Durant cet enseignement, nous nous intéresserons à l'architecture processeur la plus répandue en volume sur le marché, celle des MCU (Micro Controller Unit ou microcontrôleur).



Un système numérique de traitement de l'information peut souvent être représenté en couches matérielles comme logicielles. Ce modèle représente grossièrement les dépendances des différentes fonctionnalités matérielles et logicielles entre elles (application, bibliothèques, pilotes, périphériques internes et externes, etc) et donc le chemin de l'information dans le système (montante/entrante ou descendante/sortante). L'objectif premier restant de développer une application logicielle, répondant à un besoin et supervisant le système matériel dédié. Contrairement à un ordinateur cherchant la généricité (système d'exploitation multi-applicatifs, interfaces utilisateur génériques et standards, etc), un système embarqué est développé pour répondre spécifiquement et de façon optimale à un besoin. Une souris n'est pas une montre !

Nous allons tout au long de cet enseignement nous efforcer de comprendre puis maîtriser au mieux les différentes couches de ce modèle. Les solutions matérielles de prototypage ayant été spécifiées (MCU 8bits PIC18F27K40 et carte curiosity HPC de Microchip), une analyse attentive des documentations techniques (datasheet) sera nécessaire afin de développer les couches pilotes (drivers) puis applicative répondant à nos besoins. L'application terminale de l'enseignement visant à concevoir et développer un application audio Bluetooth. Cependant, une grande partie de notre travail consistera à développer un BSP spécifique (Board Support Package, bibliothèque de fonctions pilotes dédiées à notre carte et MCU).



Un MCU (Micro Controller Unit) ou microcontrôleur est une famille de processeur numérique présente sur le marché de l'électronique. Le premier MCU produit et commercialisé date de 1971 par Texas Instruments. Cette famille de processeur possède notamment comme particularité d'intégrer sur une même puce de silicium (on chip) l'ensemble des éléments (CPU, bus, mémoires et périphériques) faisant d'elle un processeur complet sur puce, contrairement aux ordinateurs où les composants silicium électroniques (GPP/MPU, mémoires vive DRAM et de masse HDD/SSD, chipset, ec) sont distincts et portés sur une carte mère (circuit imprimé).

Les MCU sont dédiés et spécialisés aux applications exigeant un faible encombrement spatial (processeur complet intégré sur puce), une faible consommation énergétique (souvent mono CPU à qq10-100MHz) et un faible coût financier (entre de qq0,1€ à qq1€ l'unité). Il s'agit des processeurs les plus fabriqués en volumes dans le monde chaque année avec près de 25 milliards d'unités en 2020 (source IC Insigths). Il s'agit de composants numériques de stockage,d'échange et de traitement de l'information. Rappelons les rôles de chaque entité d'un processeur architecturé autour d'un CPU :

- *CPU (Central Processing Unit) – Traiter l'information :* Composant chargé de récupérer séquentiellement par copie (fetch) une à une les instructions du programme binaire exécutable (firmware) présent en mémoire. Une fois récupérée, chaque instruction est décodée (decode), exécutée (execute) puis le résultat sauvé dans la machine (writeback). Hormis lorsqu'il est forcé en veille, un CPU réalisera sans arrêt les étapes séquentielles suivantes Fetch/Decode/Execute/WriteBack. Le pipeline matériel d'un CPU correspond à sa capacité à réaliser ces traitements en parallèle. Il existe plusieurs modèles architecturaux de fonctionnement de CPU (RISC/CISC, Harvard/VonNeumann/Hybrid, Scalar/Superscalar/VLIW, etc). Tous seront découverts en formation dans la suite du cursus.

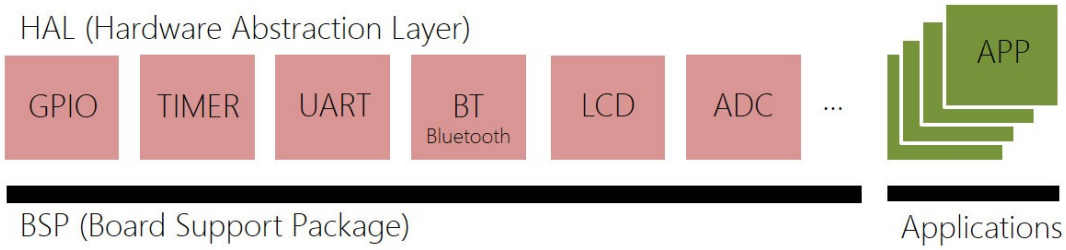
- *Mémoire – Stocker l'information :* La mémoire est chargée de stocker l'information. L'information peut être de deux natures différentes dans la machine, instructions du programme (code) ou données (data)

  - *Mémoire programme :* mémoire non-volatile (persistante), elle sera souvent nommée historiquement mémoire Flash sur MCU. Elle sera le plus souvent accessible en lecture seule par le CPU à l'usage (ReadOnly). Les technologies les plus répandues sont les mémoires flash NOR et flash NAND.

  - *Mémoire donnée :* mémoire volatile le plus souvent de technologie SRAM sur MCU. Elle sera accessible en Lecture et écriture par le CPU à l'usage (Read/Write)

- *Périphériques – Échanger/Convertir/Traiter l'information :* Les fonctions matérielles spécialisées périphériques à l'ensemble CPU/Mémoire, plus communément nommées ''périphériques'', jouent généralement l'un des trois rôles suivants dans le système : *communiquer ou échanger l'information* (par protocole de communication), *convertir l'information* (du domaine analogique vers numérique) ou *traiter l'information* (fonction de traitement matérielle spécialisée)

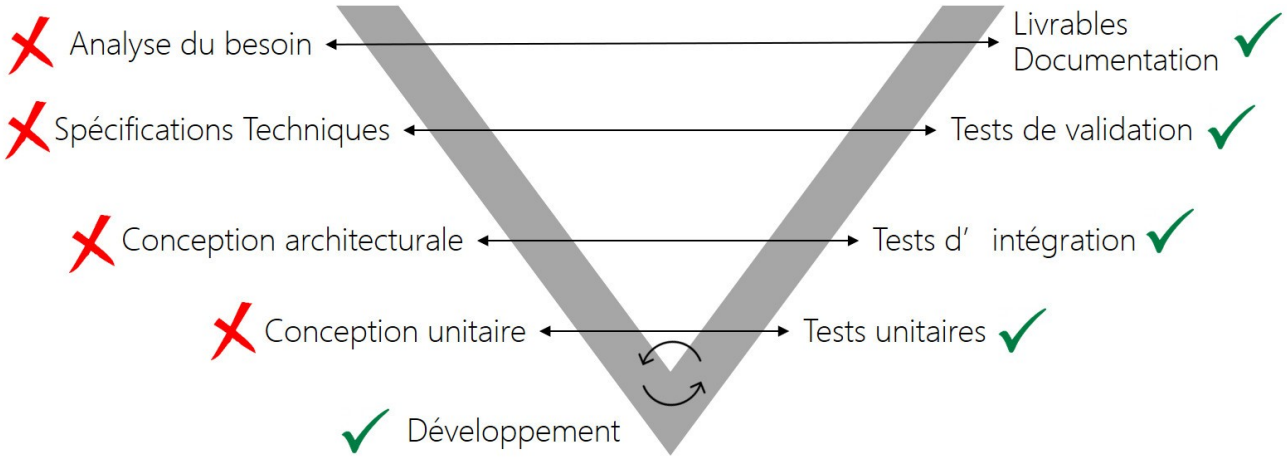
1.2. Objectifs pédagogiques



Les objectifs de cet enseignement sont multiples. Pour une grande partie des réalisations, nous aurons à développer un *BSP (Board Support Package)* ou *HAL (Hardware Abstraction Layer) from scratch* (en partant de rien) et en travaillant à l'étage registre du processeur (plus bas niveau de développement sur machine). En résumé, nous allons tout développer de A à Z. Dans notre cas, le BSP doit être vu comme une collection de fonctions logicielles pilotes (drivers) assurant le contrôle des fonctions matérielles périphériques internes (GPIO, Timer, UART, I2C, etc) voire externes (LED, switch, module Audio Bluetooth, potentiomètre, afficheur alphanumérique LCD, etc).

Le BSP sera dédié à notre processeur de travail (MCU 8bits PIC18F27K40 Microchip) et notre carte (Curiosity HPC Microchip). Une migration de technologie processeur et/ou carte nécessiterait des ajustements et de nouveaux développements. Une fois le BSP développé, testé, validé, documenté et la bibliothèque statique générée, nous développerons une application audio Bluetooth *bare-metal*. Dans le domaine de l'embarqué, *Baremetal* signifie nu sans système d'exploitation (OS ou Operating System ni RTOS ou Real Time OS). L'application implémentera un *scheduler offline*, ce point sera vu plus en détail dans la suite de la trame. En fin d'enseignement, nous ne pourrons alors qu'imaginer l'infini potentiel créatif s'ouvrant devant nos yeux !

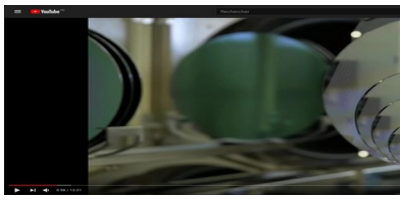
La chronologie de la trame de TP est construite autour d'un workflow typique rencontré en industrie autour de ce type de développements. Avant de développer tout applicatif, nous aurons à valider unitairement toutes les fonctionnalités et interfaces de l'application (matérielles et logicielles). Nous pourrons ensuite nous focaliser sur les phases d'intégration successives. Même si des méthodologies agiles de gestion de projet seront utilisées lors de projets en équipe en 2ième et 3ième année en majeure, celui découvert ici présente un cycle en V (V et agilité peuvent cohabiter). De même, il ne serait pas raisonnable ni pertinent de demander à un élève ingénieur en formation 1ière année de réaliser la conception d'un projet de cette taille sans avoir le recul suffisant sur le domaine. Les spécifications techniques et les conceptions ont donc été réalisées (matériel et logiciel, architecturales et unitaires), vous aurez donc à vous focaliser sur les phases de développement, de test unitaire, de test d'intégration et de validation. La conception sera découverte en majeure durant la 2ième année et la 3ième année. Ce point nécessite une très bonne assise des compétences de 1ière année.

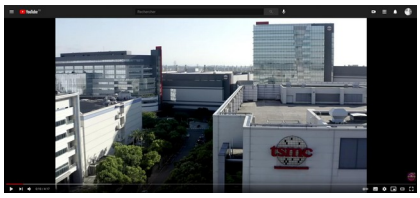


1.3. Quelques ressources internet

Voici quelques ressources en ligne pouvant vous aider à une meilleure contextualisation du domaine, des acteurs, compréhension des processus de fabrication et des outils que nous utiliserons durant cet enseignement.

How microchips are made – Infineon (13mn) and TSMC foundry world leader (4mn)

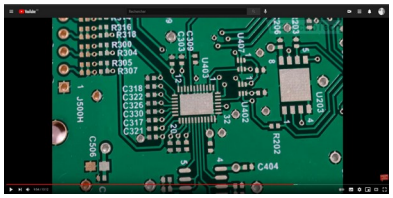




https://www.youtube.com/watch?v=bor0qLifjz4

https://www.youtube.com/watch?v=Hb1WDxSoSec

How PCB and MotherBoards are made – PCBWay (13mn) and GigaByte (2mn)



https://www.youtube.com/watch?v=_GVk_hEMjzs&t=637s

https://www.youtube.com/watch?v=bR-DOeAm-PQ

Microchip Company – Microchip (5mn)

https://www.youtube.com/watch?v=-N20QMIgh4Q