



## Documentation "prof"

Liens aux documents de base sous <https://www.didel.com/prof/Prof.html>

Vous êtes enseignant et vous avez des heures à passer avec des élèves pour faire de "l'informatique". Les approches sont multiples et dépendent de votre expérience et de vos conseillers. La séquence idéale pour comprendre l'informatique suivrait le chemin électronique – logique – architecture – instructions de base – outils de développement - langages temps réel – langages objets – outils d'applications. La culture informatique doit inclure les concepts de base des langages procéduraux, la pratique du "temps réel" et l'interaction avec l'environnement technique et humain.

Pour les élèves, il faut des étapes motivantes. Pour l'enseignant, il faut une documentation facile à adapter.

Arduino a mis en place un environnement de développement du C facilement maîtrisable et une gamme de produits abondante avec des nouveautés toutes fascinantes. Plus de performance et de fonctionnalité n'est pas nécessaire dans une première étape pédagogique.

Didel propose du matériel pédagogique simple d'utilisation. Le *Diduino-Edu* pour interagir avec des composants électroniques, le *LearnCbot* pour aller en profondeur dans le C et les interfaces d'un microcontrôleur, le *Edu-C* avec des entrées-sorties plus riches, les *Kidules* orientés applications spécifiques temps réel. Des cartes et accessoires sont développés pour des ateliers, pour des projets individuels, pour intéresser de jeunes enfants - la technologie offre tant de ressources à exploiter !

Le robot Xbot avec sa carte Diduino ou Arduino est une solution pédagogique qui a fait ses preuves, mais la documentation est trop abondante et "désordre". Xbot est un "robot" parce qu'il a des roues, mais son intérêt est dans l'étude des capteurs et le test de comportements intelligents. Notre complément de doc voudrait le rendre plus simple à mettre en oeuvre.

Pour apprendre le C et comprendre les possibilités des microcontrôleurs et de l'environnement C/Arduino, le MOOC EPFL a été complété par les 7 chapitres LC1..LC7.pdf

La documentation pour le Edu-C s'appuie sur un modèle logiciel des entrées-sorties et est plus ludique. Les anciens documents Diduino font la transition avec les composants électroniques.

Pour programmer des applications, on se réfère à des fonctions, des bibliothèques logicielles et matérielles dont il est difficile de faire une liste exhaustive. Ce qui se trouve sur le Web est souvent naïf ou difficile. Nos documents veulent compléter ce que vous trouvez.

### Documents dans le dossier prof

Un "robot" est formé de modules qu'il faut présenter successivement, tout en allant plus loin dans la maîtrise de la programmation. Des documents de 2 pages couvrent ces modules et donnent l'information essentielle, que le prof doit ensuite mettre en forme pour son enseignement. Les références données et la recherche sur le Web permettent de donner l'ampleur voulue au chapitre qui sera traité.

### Accès aux fichiers pdf et zip

Depuis 20 ans, jdn crée avec Word des fichiers .pdf contenant des liens à des fichiers .pdf qu'il était facile de cliquer dans les browsers. La combinaison word/pdf reste la plus efficace pour transmettre les résultats de notre travail. Nos documents sont optimisés pour être imprimés sans gaspiller le papier et peuvent être consultés sur tablette.

Pour faciliter leur accès, des fichiers html listent nos fichiers par groupes de produits ou groupe de notions. En première étape (août 2018), les documents sur le Xbot et ses capteurs ont été complétés et listé sous [www.didel.com/prof/Xbot.html](http://www.didel.com/prof/Xbot.html)

Evidemment, la majorité de ces documents sont mauvais et devraient être éliminés ou refaits.

Éliminer signifie lien mort, et personne n'aime tomber dessus.

Refaire est un énorme travail. Quelques documents importants sont retravaillés et les exemples logiciels mis au goût du jour.

Ajouter vos documents ? Pourquoi pas. En tout cas donner le lien à tout ce que je considère, avec l'avis d'amis, comme ayant une valeur pédagogique.

## Classement

Les groupes de liens (il y a beaucoup de documents) sont classés selon les thèmes ci-dessous. Chaque ligne est associée sous <https://www.didel.com/prof/Prof.html> à un lien vers une liste de liens qui recouvrent le sujet.

- 1 Concepts et opinions
- 2 Programmer en C/Arduino
  - MOOC EPFL** Modules Edu-C Clair Diduino
- 3 Savoir faire en C
  - Résumé C Erreurs Fonctions Arduino Interpolation et moyennage
- 4 Interruptions et librairies
  - Temps partagé Interruptions LibX
- 5 Mise au point et affichages
  - Observation des signaux Terminal série **Tell Oled** Strips
- 6 Cartes
  - Diduino DuiFamily Sam Microdules
- 7 Robots
  - Xbot Capteurs Xbot** Drawbot Wirebot
- 8 Capteurs
  - I2C Optique Autres
- 9 Asservissement
  - PwmPfm** Encodage
- 10 Kits et composants
  - Pour makers Kids Catalogue expliqué

*En bleu, les thèmes qui ont déjà de l'information dans Prof.html*

Les noms des fichiers s'efforcent d'être cohérents. Nous aimons les noms en "dos de chameau". Les dossiers commencent par une minuscule, les fichiers par une majuscule. La plupart de nos documents anciens se retrouvent en cherchant sous Google "Didel nomassezprécis".

## Remarques générales

La documentation de base des produits Didel n'a pas de prétention pédagogique pour des élèves. Le but est de décrire le produit et donner des exemples de programmation. Une documentation pédagogique qui dépend du niveau des élèves et du temps à disposition doit chaque fois être réalisée.

Concrétiser ce qui se passe, faciliter le dépannage, gérer les variantes de programmes, utiliser des librairies suppose une succession d'étapes qui ne peut pas être prédéfinie.

Les documents prof/Xxxx sont une tentative pour apporter l'information un peu différemment, en espérant que cela facilite l'établissement des documents que chaque enseignant doit se créer.

Les différences d'approche de chacun sont notables, aucun conseil ne peut être donné.

En programmation de base, il y a en gros trois approches avec

- le C sur écran (base de données)
- le temps réel (interaction avec du matériel)
- le Python (algorithmes).

Expérimenter avec les imperfections du matériel et les problèmes de mise au point, de temps de réponse, de récupération d'erreurs est un passage obligé de la culture informatique.

Comprendre ce qui se passe dans le processeur, avec quoi et comment il agit sur son environnement est fondamental. Arduino a popularisé cet aspect, en ajoutant des fonctions et librairies qui facilitent l'écriture de programmes et permettent des applications "vendables" à moindre effort. Il faut aller plus loin, plus profond.

## Arduino/C

Utiliser l'environnement de développement (IDE) d'Arduino pour écrire des programmes C variés avec des contraintes temps réel et un souci d'observer les paramètres est notre choix.

Nous critiquons certains aspects d'Arduino, notre style de programmation ne convient pas à tous. Que chacun fasse au mieux avec l'information à disposition sur le Web ; notre documentation essaye d'être plus précise, plus complète ; elle s'efforce de voir les différents aspects d'un problème et ses solutions.

### **Documents "prof"**

Le MOOC EPFL nous a fait réfléchir à ce qu'il était important d'avoir bien compris pour faire des applications microcontrôleurs. L'enseignement au lycée/collège a d'autres contraintes et il faut "écrémer" en gardant le plus possible les notions fondamentales. Les documents "prof" font quelques commentaires dans ce sens, et donnent des conseils pour la mise en œuvre. Critiques et contributions sont bienvenues.

### **Documents "C" "AVR328" "Debug" et "LibX"**

Comprendre, réaliser donne envie de documenter. Il en résulte quantités de documents que nous essayons de lister, avec l'idée de les améliorer une fois. Mais il est plus rapide et motivant de créer du nouveau. Nos documents sont toujours datés, mais cela ne permet pas de savoir s'ils sont encore à jour.

### **Rôle des robots**

Un robot ne doit pas être vu comme un objet qui se déplace. C'est un système informatique avec des capteurs qui demande à réfléchir au monde réel, comprendre la physique et les technologies, saisir toutes les contraintes de l'application, bref développer sa pensée computationnelle qui conduira au meilleur algorithme et à une programmation correcte, si possible élégante. L'imprécision du monde réel par rapport à tout ce que l'on peut simuler sur écran est un élément important.

Avec des fichiers HTML qui ont une adresse courte, nous espérons vous faciliter l'accès à notre information. Dans ces fichiers html, il y a un lien cliquable et un minimum d'information pour aider votre choix.