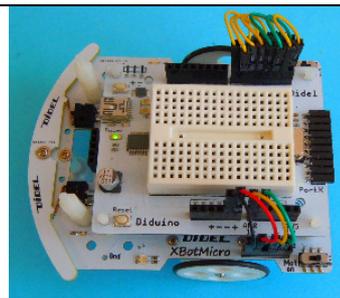
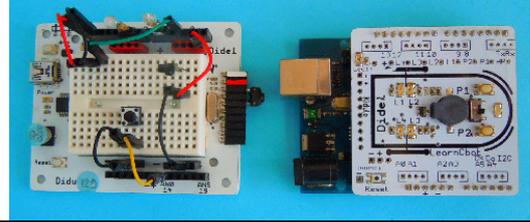




## Xbot – prise en main

Liens aux docs sous <https://www.didel.com/prof/Xbot.html>

On suppose que l'apprentissage du C avec de nombreux exercices ont été fait avec Arduino/Diduino et si possible le LCbot, pour maîtriser les bases du C. Les #define et fichiers inclus ont été pratiqués.



Les élèves découvrent le Xbot. La carte Diduino/Arduino est câblée, ne pas toucher. Le câble USB est connecté au PC, la carte reconnue par le driver. Quel programme charger?

Expliquer le câblage de la carte et le fichier de définition ne semble pas approprié au lycée.

Le premier programme utilise les instructions qui lisent les moustaches et activent les moteurs. Leur définition est cachée mais leur fonctionnalité doit être clairement expliquée.

ObsG et ObsD sont l'équivalent de variables booléennes. Valeur 1 si obstacle à gauche ou à droite, 0 si ouvert. On peut dire que c'est un alias de digitalRead (pinMoustache, LOW).

Il y a des rebonds de contact de 20ms. Quand le contact se fait, la Led proche s'allume. On peut expliquer que la pullup associée au poussoir a une Led en série, si cette notion a été vue.

Si on écrit `if (ObsG) {Action;}` l'action a lieu si la moustache est pressée.

Si on écrit `if (!ObsG) {Action;}` l'action a lieu si la moustache est relâchée.

Pour les moteurs, c'est inutile d'expliquer avant que cela ne soit nécessaire qu'il y a 2 signaux par moteur, donc 4 combinaisons qui agissent différemment électromécaniquement.

Les fonctions Avance(); Recule(); TourneG(); TourneD(); StopGD() sont claires.

(Note : Stop() ne doit plus être utilisé, problème avec l'affichage Oled)

Avec ces fonctions et un délai, on peut proposer des déplacements de plus en plus complexes. L'évitement d'obstacle peut être plus ou moins élégant. Voir des exemples dans le zip.

Ne pas manquer de faire un programme qui attend que l'on agisse sur les moustaches et exécute l'une de 3 actions (TestVitMot.ino). Bien comprendre le poussoir est important, voir :

[www.didel.com/diduino/ActionPoussoir.pdf](http://www.didel.com/diduino/ActionPoussoir.pdf) .

Exercer des applications avec des machines d'état (switch case) est important. TestVitMot est très mal écrit du point de vue structure ; l'usage de `if` dont l'intérieur est bloquant est à proscrire. Le refaire en utilisant des compteurs qui comptent le nombre de passages dans la boucle pour faire évoluer la machine.

Le suivi d'une paroi concave et convexe permet quantités de défis.

Un labyrinthe simple qu'il faut parcourir une 2<sup>e</sup> fois sans toucher les bords doit être possible.

Le contrôle de la vitesse est développé dans prof/PwmPfm.pdf.

Le Xbot devient très intéressant avec ses capteurs, faciles à connecter et interfacer. D'autres capteurs Arduino, analogiques ou I2C sont faciles à connecter.

## Documentation

Ouvrir [www.didel.com/prof/Xbot.html](http://www.didel.com/prof/Xbot.html) pour avoir accès aux liens

XbotMicro.pdf Flyer général

XbotStart.pdf Toute l'info nécessaire pour débiter

XbotDebut.html Plusieurs exemples de programmes

XBotReference.pdf Définitions et fonctions utiles

XbotBase.html Résumé peu utile

Diduibot.pdf Doc des premiers protos du Xbot, qui ont succédés au DDR

Le fichier liste aussi 4 documents en Anglais.

Le Xbot est en vente chez Boxtec, RobotMaker, Tindie et Didel.