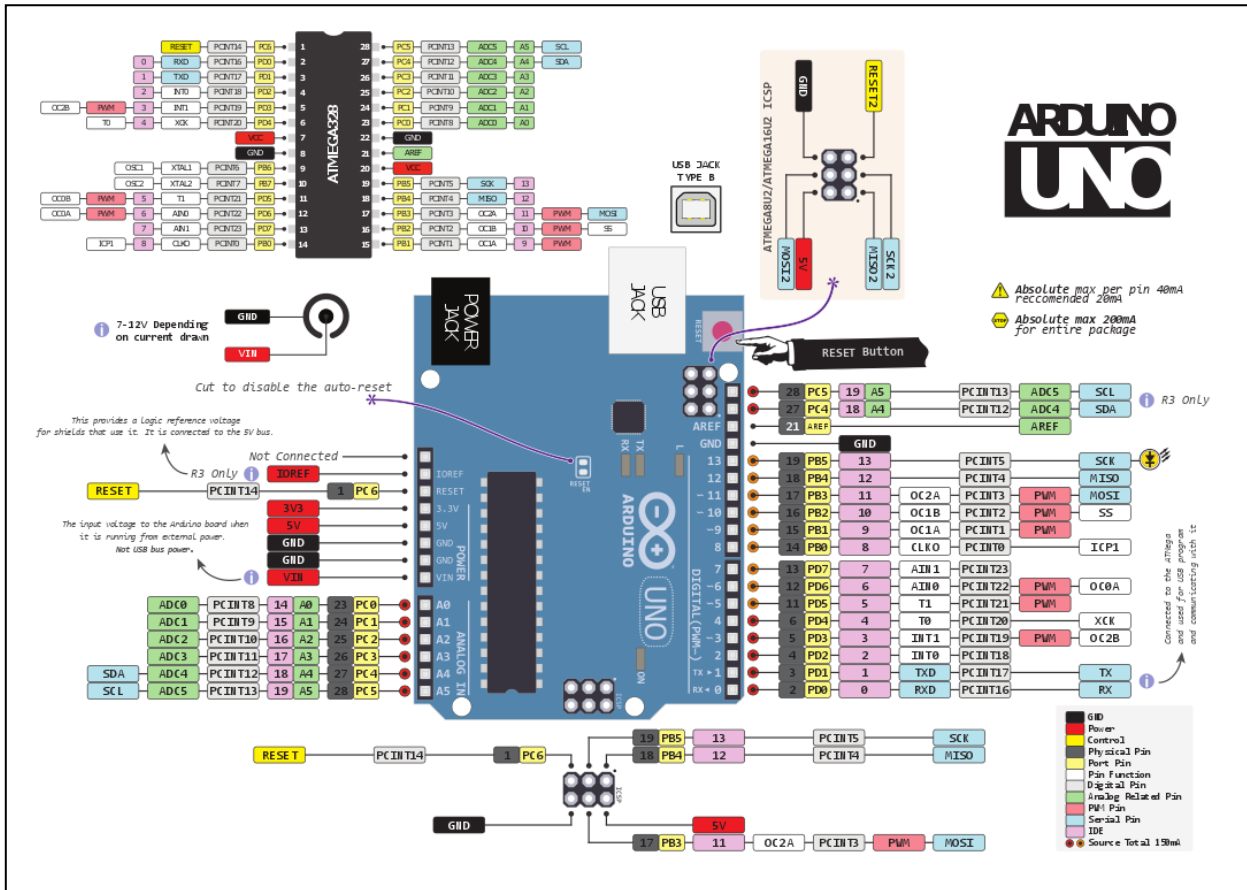


Les Algorithmes

PROGRAMMATION d'un ARDUINO sous « FLOWCODE V9 », simulation avec « Proteus » et test en réel

I) Présentation de la carte Arduino UNO :

Vue d'ensemble :



La carte Arduino Uno est une carte à microcontrôleur basée sur l'ATmega328.

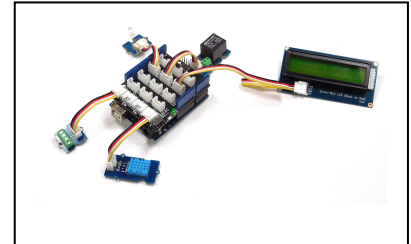
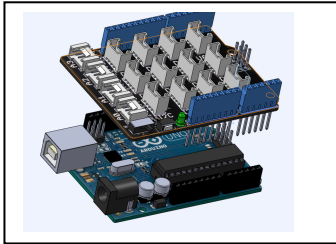
Elle dispose :

- de 14 broches numériques d'entrées/sorties (dont 6 peuvent être utilisées en sorties PWM (largeur d'impulsion modulée)),
- de 6 entrées analogiques (qui peuvent également être utilisées en broches entrées/sorties numériques),
- d'un quartz 16MHz,
- d'une connexion USB,
- d'un connecteur d'alimentation jack,
- d'un connecteur ICSP (programmation "in-circuit"),
- et d'un bouton de réinitialisation (Reset).

Elle contient tout ce qui est nécessaire pour le fonctionnement du microcontrôleur; Pour pouvoir l'utiliser et se lancer, il suffit simplement de la connecter à un ordinateur à l'aide d'un câble USB (ou de l'alimenter avec un adaptateur secteur ou une pile, mais ceci n'est pas indispensable, l'alimentation étant fournie par le port USB).

Les deux rangées de connecteurs, situées de part et d'autre de la carte, permettent sa connexion au monde extérieur et, si l'on utilise les cartes d'interface, ou shields, au format Arduino, il est possible d'enficher directement ces derniers dans la carte de base, sur plusieurs niveaux si nécessaire.

Une de ces cartes d'interfaces que nous utiliserons est une carte Grove. Grove est un système modulaire et simple d'utilisation conçu par Seeed pour connecter facilement un Arduino, à une large gamme de "modules", tels que des capteurs, des actionneurs, etc... C'est donc un excellent moyen de réaliser rapidement des prototypes de projets sans devoir étudier les Datasheet des composants. Mais attention... cela ne sera pas toujours le cas lors des activités en SIN.



Les lignes Digital 0 (IO₀) et Digital 1 (IO₁) ne peuvent pas être utilisées car elles sont réservées pour communiquer avec l'ordinateur.

Les lignes SCL (A₅) et SDA (A₄) sont réservées pour dialoguer avec des périphériques utilisant le protocole I²C.

« Uno » signifie « un » en Italien et ce nom marque la venue prochaine de la version 1.0 du logiciel Arduino. La carte UNO et la version 1.0 du logiciel seront la référence des versions Arduino à venir. La carte Uno est la dernière d'une série de carte USB Arduino, et le modèle de référence des plateformes Arduino.

Le langage de programmation utilisé est le C++, compilé avec avr-g++, et lié à la bibliothèque de développement Arduino, permettant d'utiliser la carte et ses entrées/sorties.

Synthèse des caractéristiques :

Microcontrôleur	ATmega328
Tension de fonctionnement	5V
Tension d'alimentation (recommandée)	7-12V
Tension d'alimentation (limites)	6-20V
Broches E/S numériques	14 (dont 6 disposent d'une sortie PWM)
Broches d'entrées analogiques	6 (utilisables en broches E/S numériques)
Intensité maxi disponible par broche E/S (5V)	40 mA (ATTENTION : 200 mA cumulé pour l'ensemble des broches E/S)
Intensité maxi disponible pour la sortie 3.3V	50 mA
Intensité maxi disponible pour la sortie 5V	Fonction de l'alimentation utilisée - 500 mA max si port USB utilisé seul
Mémoire Programme Flash	32 KB (ATmega328) dont 0.5 KB sont utilisés par le bootloader
Mémoire SRAM (mémoire volatile)	2 KB (ATmega328)
Mémoire EEPROM (mémoire non volatile)	1 KB (ATmega328)
Vitesse d'horloge	16 MHz

Vous pouvez également, pour de plus amples informations consulter le cours sur « Mise en œuvre d'un microcontrôleur » mis à votre disposition sur le site LYCÉE CARNOT – STI2D.

Lien : <http://88.125.120.25:49280/cours/sin/006/Arduino.html>

II) Présentation du logiciel de programmation Flowcode V9

Avant de commencer votre première activité de programmation rendez-vous au lien ci-dessous afin d'obtenir une présentation globale de Flowcode V9

Lien : <http://88.125.120.25:49280/cours/sin/006/FlowcodeV9.html>

Synthèse, analogie et questionnement entre une programmation en langage Blocks et Algorithmique

Vous avez sans doute depuis le collège entendu parler d'Ardublock. Logiciel permettant à tout le monde de programmer la carte Arduino sans avoir besoin d'apprendre un langage de programmation.

Ardublock est un environnement de programmation graphique pour rendre la programmation informatique physique avec Arduino aussi facile que de « glisser-déposer ». C'est un plugin qui s'ajoute à l'IDE Arduino.

Nous allons donc tout simplement faire l'analogie entre ces deux langages de programmation. Le langage Blocks et L'Algorithmie.

Question 1 :

Suite au visionnage de la vidéo de présentation de Flowcode V9 on vous a présenté qu'il est possible de programmer en utilisant différentes méthodes ou langages de programmations.

Citez les... il y en a six.

Nous pouvons maintenant aborder notre premier exercice ensemble, pour cela vous allez suivre le tutoriel suivant :

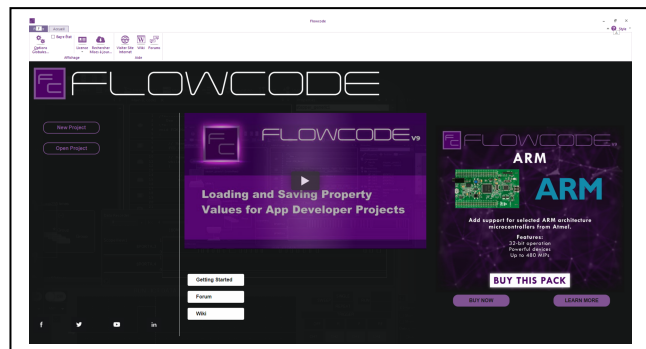
L'idée est de démarrer, avant de réaliser vous-même d'autres programmes, d'un programme conçu par Blocks à l'aide d'Ardublock. Vous n'aurez pas toujours pour la suite des exercices cette aide, le but n'étant pas d'utiliser la programmation par Blocks mais bien par **Algorithmes sous Flowcode**.

Cependant cette aide vous sera précieuse lors des projets d'Innovation Technologique car vous trouverez facilement sur internet comment programmer les composants « Grove » par Blocks et pourrez adapter tout cela sous Flowcode en utilisant la programmation par Algorithmes.

C'est parti ...

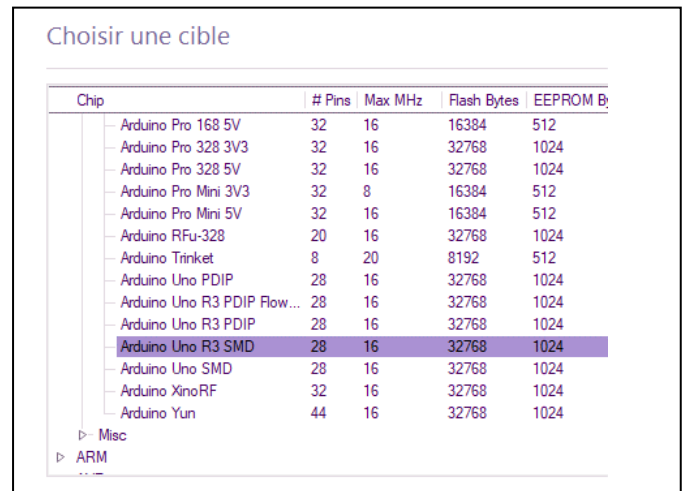
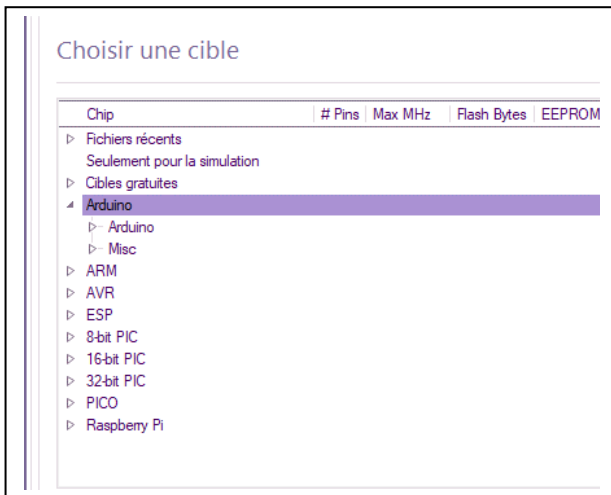
Programme 1 : « Faire clignoter une LED »

- Ouvrir dans un premier temps le fichier [pdf](#) « ardublock-led ».
- Créer un dossier dans votre H:\Travail afin d'y enregistrer vos programmes.
- Exécuter le logiciel « Flowcode V9 ».

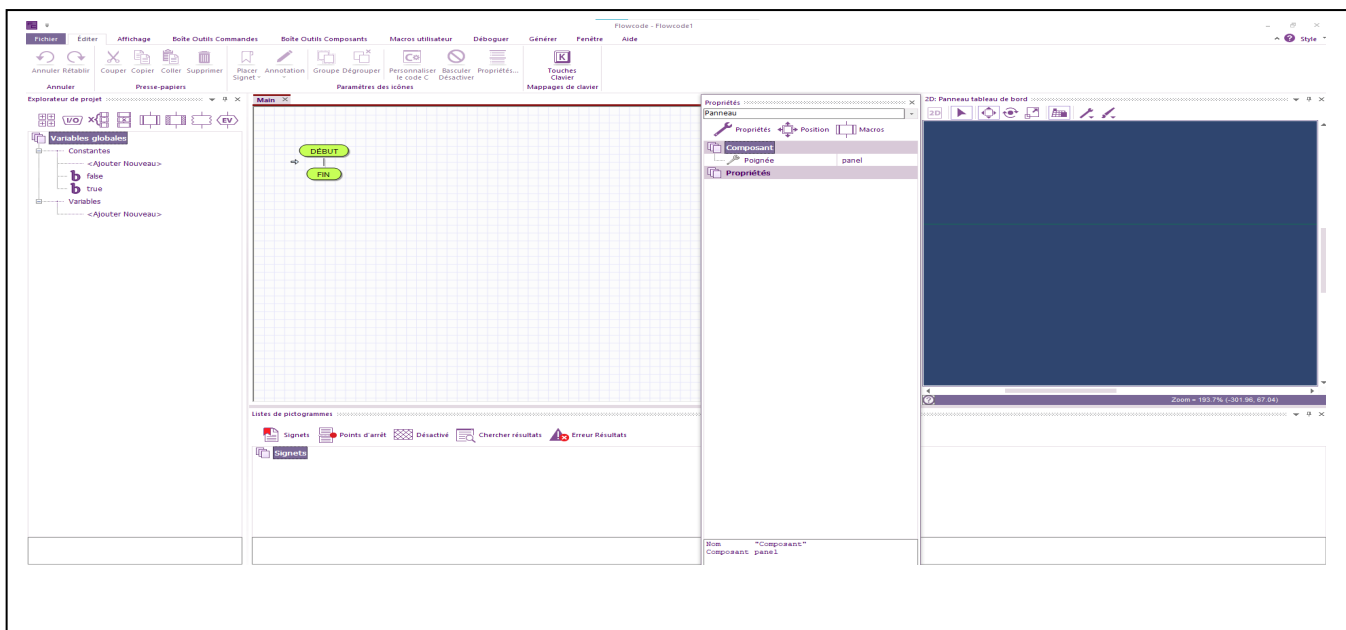


III-1) Paramétrages de Flowcode V9 pour un nouveau projet avec Arduino:

Cliquez sur « New project » et sélectionner le microcontrôleur que l'on va utiliser

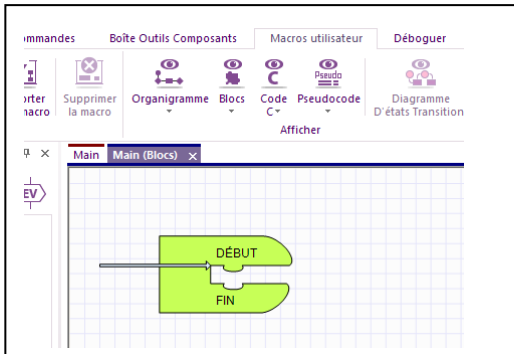


Vous obtenez donc la fenêtre ci-dessous. Celle-ci est composée d'onglets, d'un explorateur de projet situé à gauche de l'écran, d'un onglet « Main » où se trouvera le programme principal, d'un onglet « Propriétés » où nous allons pouvoir vérifier et paramétrer les propriétés des composants et d'un « 2D: Panneau tableau de bord » où nous allons pouvoir y déposer les composants pour la simulation.



Comme on vous l'a présenté dans la vidéo de Flowcode V9 il est possible d'utiliser différents langages de programmation, nous allons donc avant de commencer nos premiers algorithmes commencer par faire une analogie entre le langage Blocks et l'Algorithmie.

Flowcode proposant d'origine la programmation sous forme d'Algorithme nous allons faire apparaître la fenêtre de programmation par Blocks. Procéder comme suit :

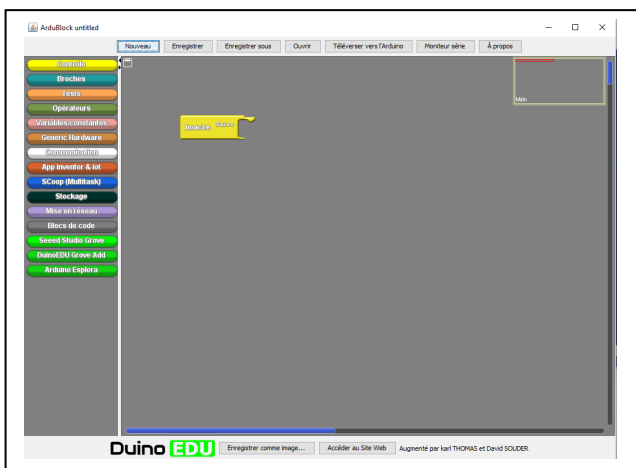


Sélectionner l'onglet « Macros utilisateur » puis « Blocs » et « Main », vous obtenez maintenant une fenêtre de programme principal sous forme de Blocks.

III-2) Décodage du programme Blocks :

Nous allons maintenant décoder le programme blocks fourni dans le fichier pdf « ardublock-led », le retranscrire avec Flowcode dans le « Main Blocs » et ensuite faire l'analogie avec le « Main » Algorithme.

Analysons dans un premier temps le programme Blocks « Ardublock-led » :

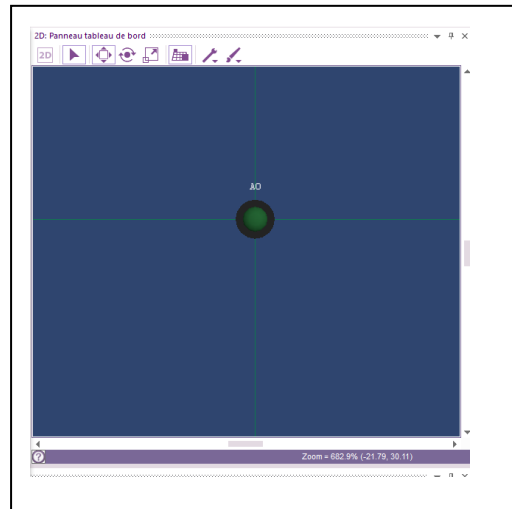
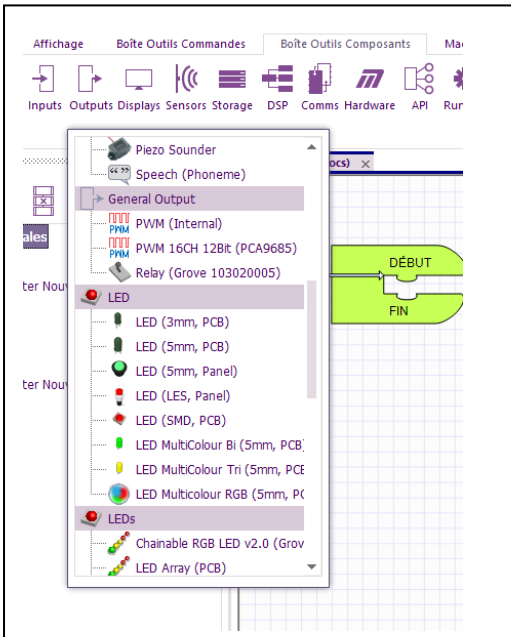


- La première différence est que sous Ardublock, voir la vue d'ensemble du logiciel ci-contre, il n'y a pas de début ni de fin de programme comme sous flowcode. Cela n'est pas gênant nous nous en accommoderons.
- La seconde différence est que sous Flowcode vous ne trouveriez pas les commandes sous forme de Blocks mais uniquement sous forme de symboles algorithmiques. Cependant les appellations sont identiques, une boucle reste une boucle.

En regardant le programme ci-dessous on remarque que l'on fixe une sortie D6 à l'état HAUT (état logique 1), celle où sera connecté la LED. Puis on réalise un Delay (une pause) de 1000ms, cela signifie que la LED restera allumée pendant 1s. Ensuite on passe la sortie D6 au niveau BAS, état logique 0, suivi d'un Delay de 1000ms donc pendant 1s. Comme notre programme se trouve dans une boucle, tant que cette boucle est valable, notre programme bouclera sans s'arrêter et la LED clignotera.

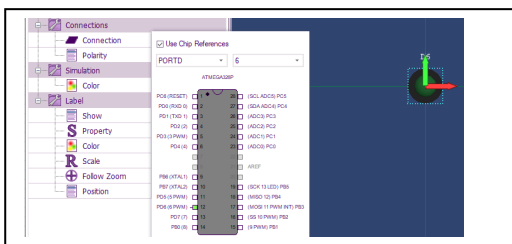
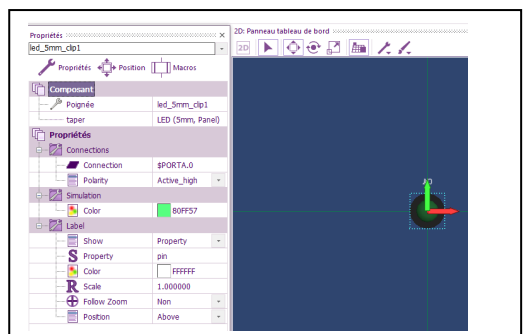
III-3) Réalisation du programme sous Flowcode :

Dans un premier temps nous allons positionner la LED sur le « 2D: Panneau tableau de bord ». Pour cela sélectionner l'onglet « Boîte Outils Composants », comme la LED est un actionneur (L'actionneur étant un dispositif matériel qui transforme l'énergie qui lui fournie en un phénomène physique), vous devriez trouver celle-ci dans « Outputs » et obtenir cela :



Il faut maintenant connecter la LED sur le bon port, en l'occurrence D6.

Sélectionner la LED dans le « 2D: Panneau tableau de bord », vous remarquerez que les propriétés de celle-ci apparaissent et qu'elle est connecté sur le PORT A0.

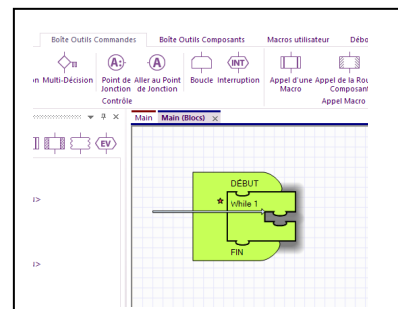


- Il suffit maintenant de sélectionner la sortie, le PORT, sur laquelle nous voulons la connecter.

Comme nous utiliserons qu'un seul composant, la LED, c'est fini pour cette partie.

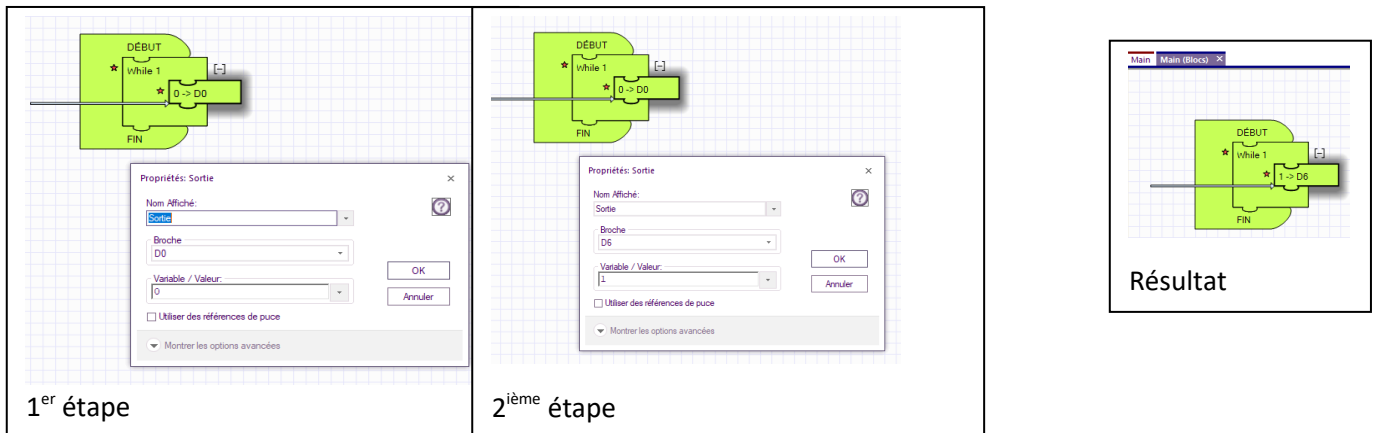
Puis en sélectionnant l'onglet « Boîte Outils Commandes » dans Flowcode vous trouverais les commandes dont vous aurez besoins.

- Insérer donc une boucle sous Flowcode dans le « Main (Blocs) », vous devriez obtenir cela :



Ensuite nous devons positionner une sortie, pour cela sélectionner en restant appuyer sur « Sortie » et glisser, déposer là dans la boucle afin d'obtenir cela :

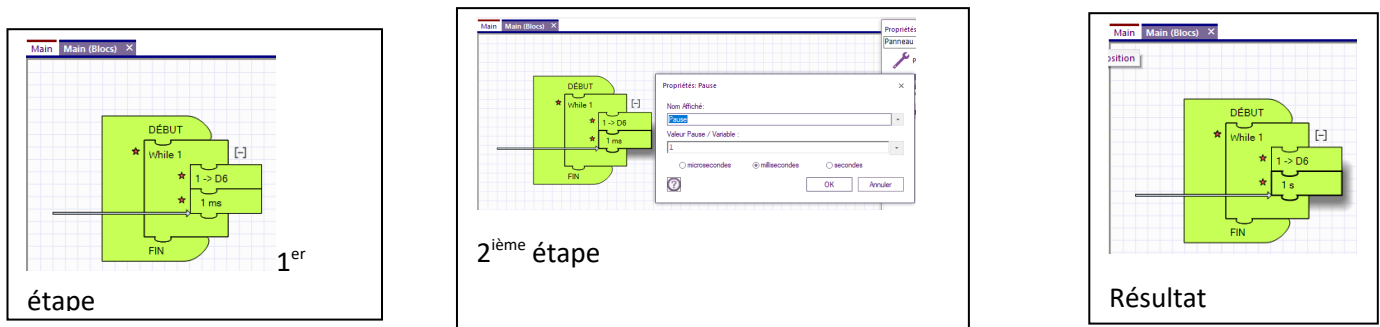
On remarque que cette sortie s'appelle D₀ et que l'on met 0 dans D₀ (0 → D₀). Afin de répondre à notre besoin, c'est-à-dire mettre à 1 la sortie D₆ il suffit de cliquer 2 fois sur le Block « Sortie » et de configurer correctement celle-ci.



Insérons maintenant notre temporisation, Delay ou Pause dans Flowcode... En procédant de la même manière que précédemment (glisser, déposer) vous devriez obtenir cela :

Pour ensuite obtenir le résultat ci-contre →

Procéder en double cliquant sur le block Delay afin de le paramétrer à 1s.



Question 2 : À vous maintenant de terminer le programme sous forme de Blocks et de le fournir dans votre compte rendu.

Question 3 : Vous avez terminé votre programme principal « Main Blocs », vous devez maintenant fournir votre programme sous forme d'Algorithme dans votre compte rendu. Il suffit simplement d'aller dans le « Main » de Flowcode et de voir le résultat.

Nous allons, avant de poursuivre la programmation sous forme d'Algorithmes aux travers de différents exercices dans une autre activité, nous intéresser à un autre langage de programmation qu'il est possible de gérer avec Flowcode V9, le langage C.

Question 4 : Éditez le code C sous Flowcode et fournir celui-ci dans votre compte rendu.

Maintenant que vous avez obtenu les 3 langages sous Flowcode vous pouvez donc faire une analogie entre-eux.

Question 5 : Présentez sous forme d'un tableau les syntaxes utilisés pour les instructions que vous avez utilisées dans votre programme pour les 3 langages de programmation.

Voilà ! l'analogie est faite, vous comprenez maintenant qu'il nous est possible d'utiliser le langage qui nous convient dans Flowcode.

III) Test du programme en simulation :

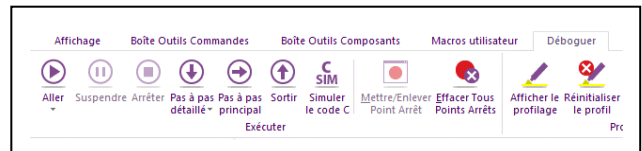
Avant d'effectuer ce test réellement, avec les composants, il est nécessaire de faire une petite mise au point IMPORTANTE.

- Un programme sous Flowcode peut très bien fonctionner en simulation mais pas en réel.
- Il est nécessaire avant de commencer son programme d'élaborer le schéma structurel (schéma avec les composants) à l'aide d'un logiciel de simulation, lui-même issu du schéma fonctionnel (Fonctions de la chaîne d'information). Dans notre cas nous utiliserons Proteus.
- Lorsque le schéma structurel et le programme sont validés en simulation sous Proteus nous pouvons enfin passer au réel (Câblage) mais cela nécessitera en fonction des cas peut être quelques ajustements.

...Le chemin est long du projet à la chose. *Molière*

Revenons donc au test de notre petit programme sous Flowcode, pour le reste votre professeur vous guidera plus tard pour l'élaboration du schéma structurel sous Proteus.

Flowcode nous permet donc de tester notre programme, pour cela il vous suffit dans l'onglet « Débuguer » et de lancer la simulation en cliquant sur l'icône « Aller ».



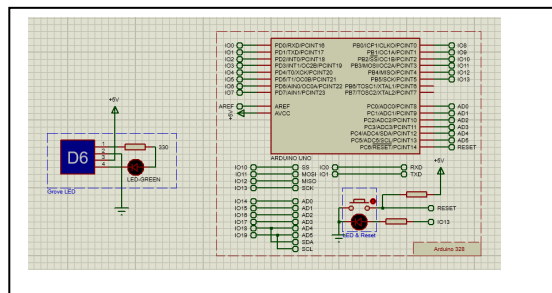
Vous pouvez maintenant vérifier dans le « 2D: Panneau tableau de bord » si la LED clignote.

IV) Validation du programme sous Proteus

Nous allons maintenant tester notre programme en simulation sous Proteus.

Proteus nous permet de simuler le fonctionnement d'un Arduino et des composants qui lui sont associés. Le but de l'activité n'étant pas concentré sur Proteus votre professeur vous guidera sur l'élaboration du schéma structurel de votre projet et vous aidera à mettre en œuvre la simulation.

Vous devriez, sous Proteus, obtenir cela :



Après avoir correctement configuré Proteus avec l'aide de votre professeur vous pouvez maintenant vérifier si la simulation fonctionne.

Question 6 : Spécifiez dans votre compte rendu les différentes étapes permettant d'obtenir le schéma structurel, les étapes de configuration ainsi que le résultat obtenu. Vous pouvez agrémenter votre réponse par des captures d'écrans voire une vidéo.

V) Étape finale : Test en réel.

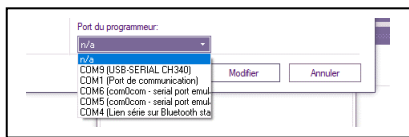
Maintenant il ne vous reste plus qu'à réaliser votre câblage en réel et le tester. Cependant si vous avez tout suivi depuis le début de cette activité il nous manque une étape essentielle...Comment compiler notre programme dans la puce de votre Arduino ?

Pour cela nous allons retourner dans Flowcode V9 et vérifier quelques paramètres avant de réaliser cette action.

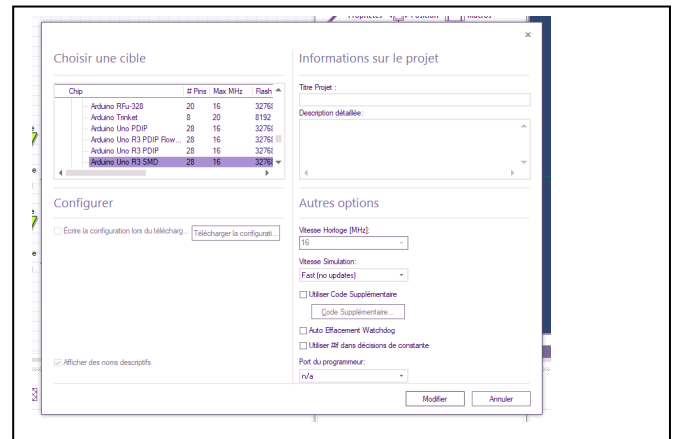
- Connecter le composant Grove « LED » sur le bon Port du BASE SHIELD de l'Arduino,
- Connecter l'Arduino sur le port USB de votre ordinateur,
- Vérifier le Port COM associé à l'Arduino dans le Gestionnaire de périphériques de Windows,
- Suivez enfin la démarche ci-dessous pour le paramétrage de Flowcode V9 et le téléversement dans l'Arduino.

Cliquez dans un premier temps sur l'icône « Options Projet », cette fenêtre devrait apparaître :

En bas de cette fenêtre il faut sélectionner le Port du programmeur.



Attention ce port est différent selon les ordinateurs, vous pouvez repérer l'Arduino il est repéré USB-SERIAL-CH340, donc dans ce cas c'est le COM9. Sélectionner le et cliquer sur « Modifier ».



Le bon Port étant maintenant sélectionné il ne reste plus qu'à compiler votre programme directement vers la puce de l'Arduino, pour ce faire : Cliquer sur l'icône « Compiler → Puce »

« **Félicitations !** » vous avez fini votre premier programme, vous n'avez plus qu'à vérifier son fonctionnement voir modifier la fréquence de clignotement de la LED.