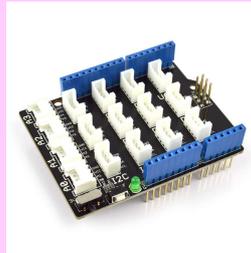


Programmation d'un clignotement de diode sous [Blockly@duino](#)

Matériel nécessaire : carte arduino uno , 1 « base shield », 1 diode.

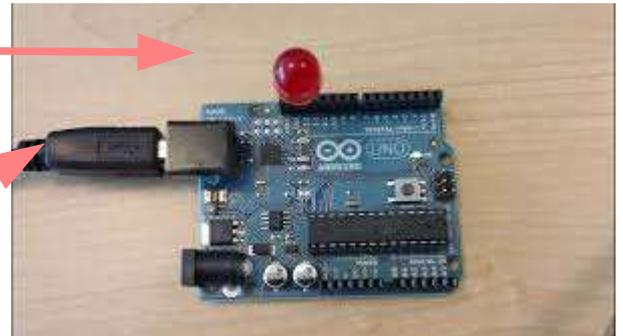


Montage

1 - Montage :

La diode est connecté entre la sortie digitale 13 et la masse (gnd)

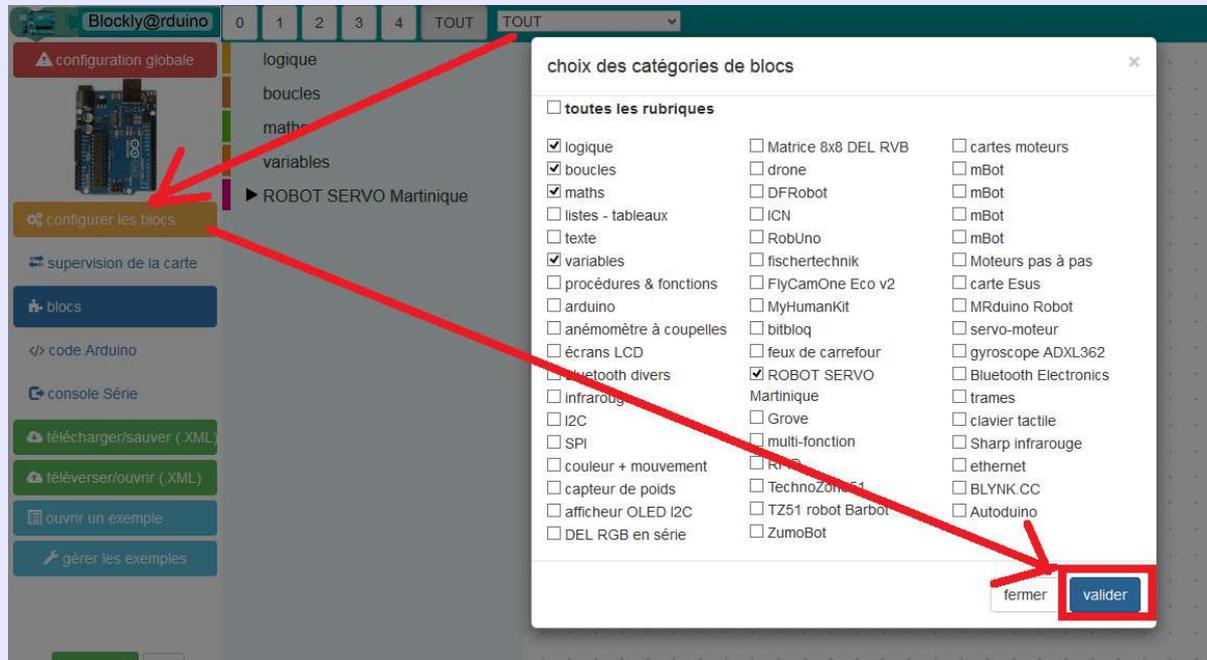
2 – Connecter la carte arduino
À l'ordinateur par la prise usb



3 – Ouvrir [Blockly@duino](#) sur le bureau



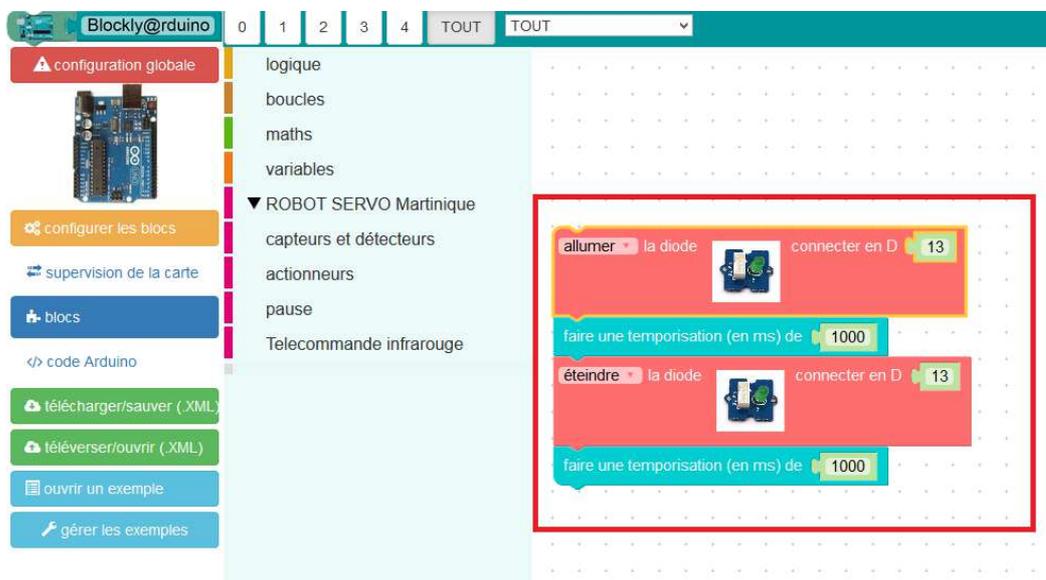
4 - Sélectionner « TOUT », cliquer sur configurer les blocs et valider



5 – Ouvrir l'IDE arduino



6 - Réaliser le programme de clignotement de diode en utilisant le menu «Robot Martinique »



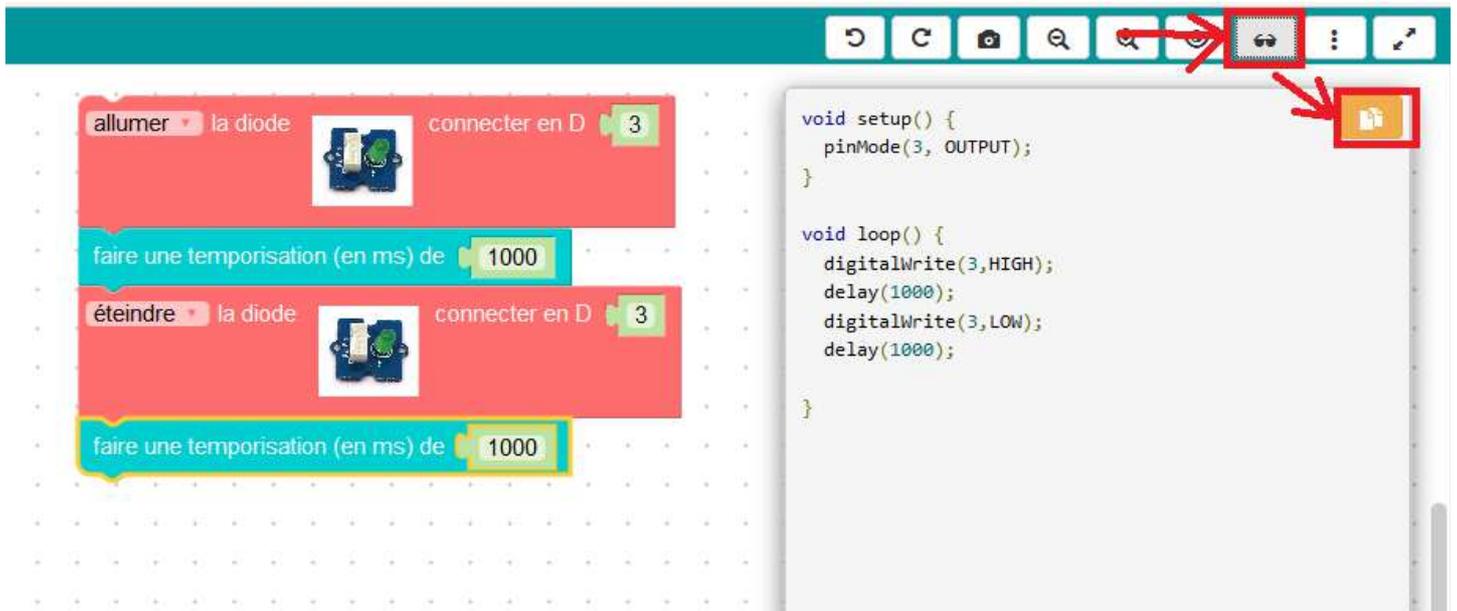
7 - Copier le programme en c++ cliquant sur les deux icones suivantes



(affiche le programme en c++)



(copier)

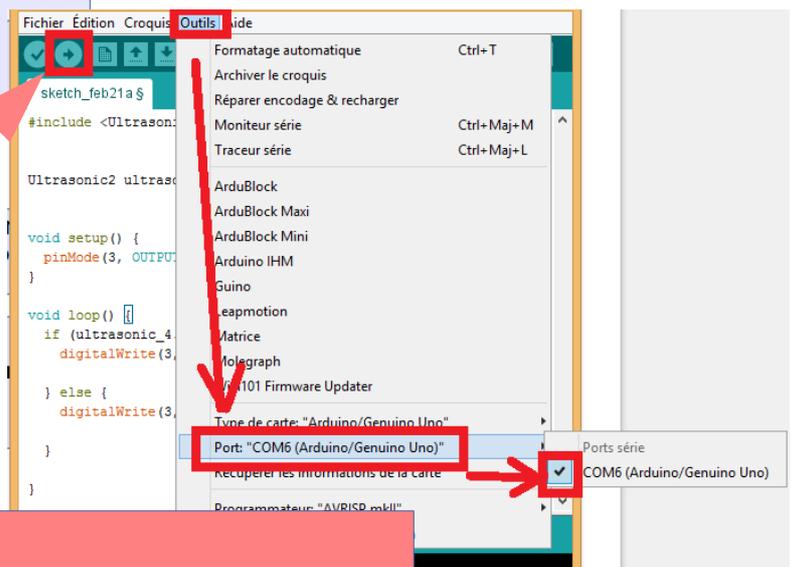


8 - Coller dans l'IDE d'arduino

→ ctrl A

→ ctrl V

9 - Téléverser le programme



Si le téléversement ne fonctionne pas vérifier que le port COM est bien sélectionner dans le menu « Outils »

activité N°2		Domotique/Robotique
Clignotement de diode		scratch / blockly

Résultat du programme :

→ La diode clignote toutes les 2 secondes !

Changer les pauses : 100ms, 50 ms , 20 ms; 10ms et téléverser , et observer ce qu'il se passe.

Programmation avec « scratch version makeblock » (sans fermer Blocl@arduino)

C'est qui « makeblock » ?

La société « Makeblock » est située à Shenzhen, en Chine . Sa carte électronique utilise « arduino uno » qui est open source.

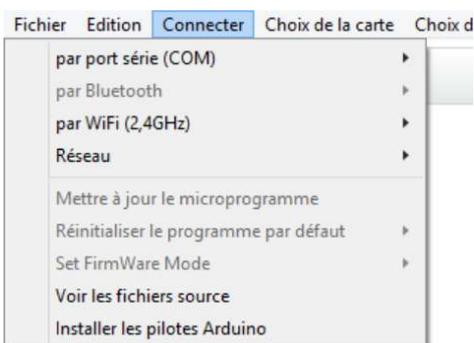
D'après makeblock, tous ses produits sont « open source » et les codes sources sont disponibles ici :

<https://github.com/Makeblock-official?tab=repositories>

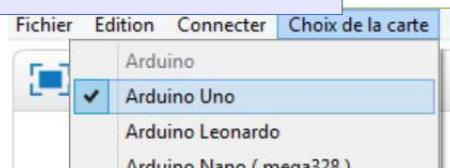
1 – Ouvrir le programme « mBlock » en cliquant sur l'icône dans le bureau



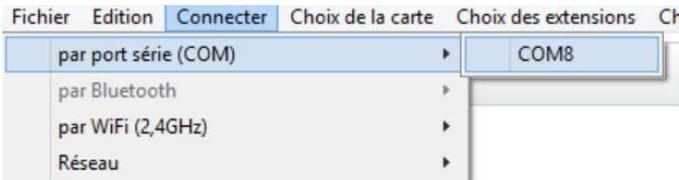
2 – Cliquer le menu « connecter », puis « installer les drivers » (si nécessaire)



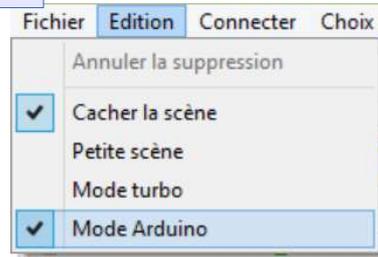
3 – Choix de la carte



4 – Connecter la carte en sélectionnant le port com.



5



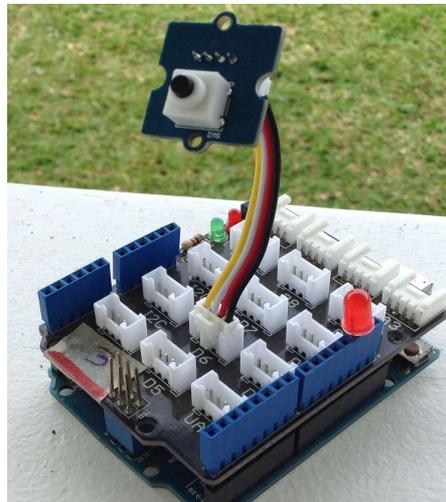
Faire le montage et programmer le cahier des charges suivant sous « mBlock », et téléverser.

Cahier des charges :

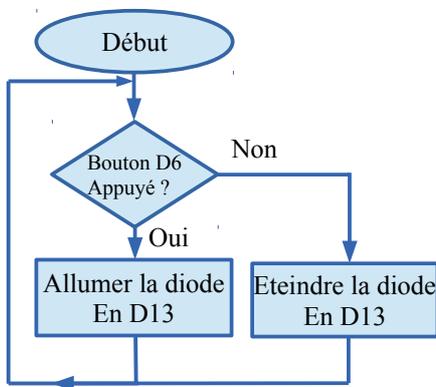
Cahier des charges :

lorsque j'appuie sur le bouton la diode s'allume, sinon elle s'éteint.

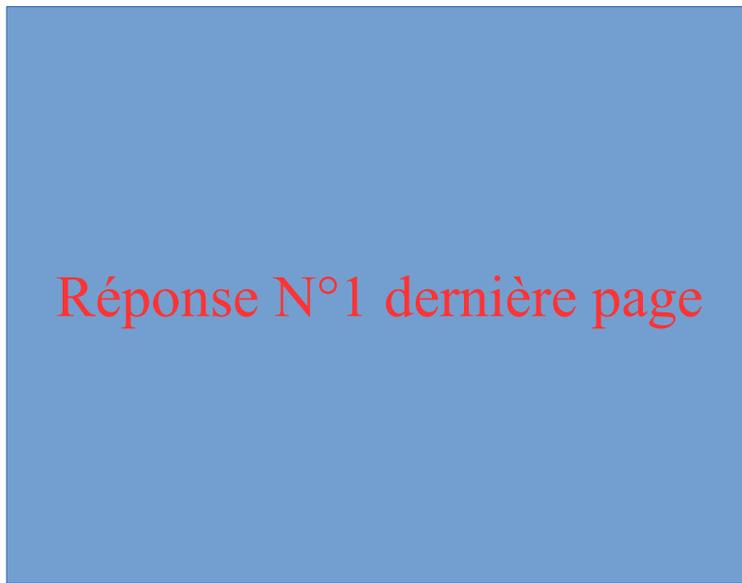
Montage : Remplacer la diode par l'interrupteur en D6:



Organigramme :



Programmation sous « scratch »



A partir de maintenant, vous allez choisir l'environnement que vous préférez « ardublock » ou « scratch ».



**Nous nous sommes rendus compte pendant le stage que « Mblock » n'était pas parfaitement optimisé pour les cartes « arduino » engendrant parfois des bugs
IL VAUT MIEUX DONC UTILISER
« Mblock » avec le robot « MBOT »**

Dernier programme de l'activité !

Cahier des charges :

Si j'appuie une fois sur le bouton poussoir alors la LED s'allume et reste allumé !
Si j'appuie une deuxième fois sur le bouton poussoir, alors la LED s'éteint et reste éteinte !
Ainsi de suite !

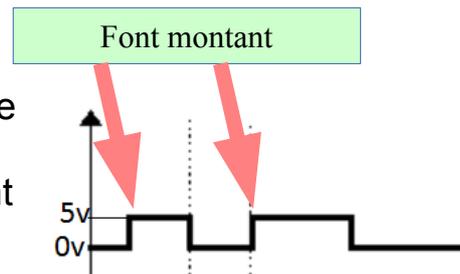
Facile non ?

Le problème posé à l'air facile.... Il ne l'est pas !

Etape 1 : il faut introduire plusieurs variables qui vont permettre de détecter un front montant.

Qu'est-ce qu'un front montant ?

Quand j'appuie sur l'interrupteur, l'entrée digitale reçoit 5V, sinon elle reçoit 0V. Quand l'entrée digitale passe de 0V à 5V, c'est un front montant



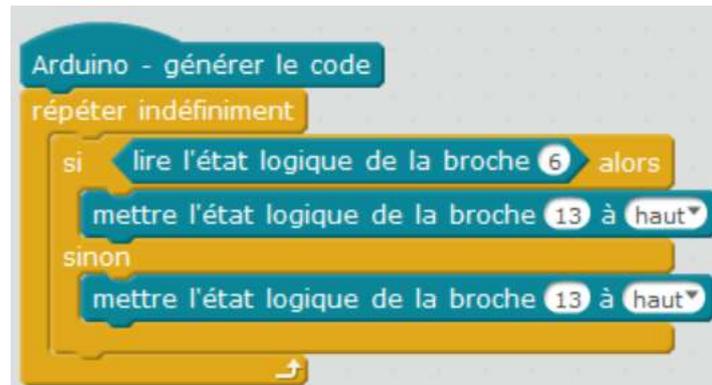
ETAPE 2 : Créer 3 variables :

« Etat bouton », « Etat bouton antérieur », et « Etat diode »

Réponse N°2 dernière page

Solution de Daniel
Parade en dernière page
qui est beaucoup
plus simple.

Réponse N°1



Réponse N°2

initialisation (setup)

- mettre la variable Etat bouton à 0
- mettre la variable Etat bouton antérieur à 0
- mettre la variable Etat bouton antérieur à 0

si Bouton poussoir appuyé connecter en D 6

alors

- mettre la variable Etat bouton à 1
- si Etat Diode = 0 et Etat bouton antérieur = 0
 - alors
 - allumer la diode connecter en D 13
 - mettre la variable Etat Diode à 1
 - sinon
 - si Etat Diode = 1 et Etat bouton antérieur = 0
 - alors
 - éteindre la diode connecter en D 13
 - mettre la variable Etat Diode à 0
- sinon mettre la variable Etat bouton à 0

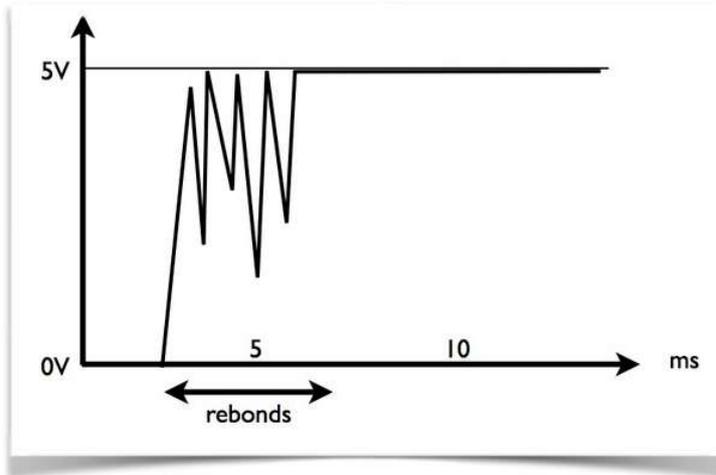
mettre la variable Etat bouton antérieur à Etat bouton



Le programme fonctionne bien, sauf si vous appuyer très rapidement sur l'interrupteur. A ce moment il peut se produire un phénomène de rebond de l'interrupteur mécanique :
SOLUTION ajouter une petite temporisation une fois que la diode est allumée ou éteinte ou utiliser la méthode logicielle qui suit.

Un BP, ça rebondit énormément !

Quand le BP est appuyé, l'état change sur la broche de l'Arduino, il passe de 0 à 5V ou inversement selon le type de BP ou le type de branchement. Mais entre les deux états, la transition n'est pas instantanée et il y a une phase de rebonds qui dure quelques millisecondes et qui est due au choc entre les contacts métalliques un peu élastiques.



D'où des changements d'états très rapides qui sont captés par l'Arduino, et qui peuvent entraîner des erreurs. Dans le cadre d'un compteur, pour le réglage de l'heure par exemple, un seul appui pourra être compté comme plusieurs à cause de ce phénomène.

Pour l'éviter, un condensateur de 100 nF en parallèle sur le BP, pourra être employé. Mais la valeur du condensateur doit être déterminée expérimentalement, les BP étant tous différents sur ce plan.

Il existe heureusement une autre méthode purement logicielle. Elle consiste à lire l'état de la broche et la stocker dans une variable, d'attendre quelques millisecondes et de lire une nouvelle fois la broche en la comparant à la valeur stockée. Si la valeur est la même, l'appui est confirmé.

SOLUTION de Daniel Parade

Tant que le bouton n'est pas relâché, on boucle