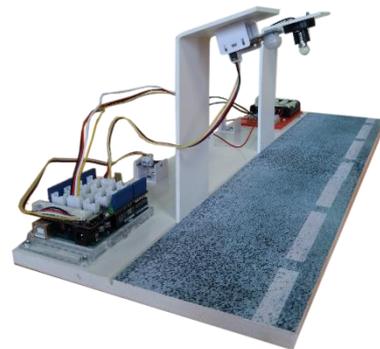


Simulation d'un éclairage urbain

ATTENTION aux maquettes !

Elles sont fragiles et doivent être manipulées avec précaution.



Objectif : Piloter une maquette réelle avec capteurs, actionneurs et effecteurs

Première partie : Découverte de la maquette

La maquette représente une demi-rue.

Elle est contrôlée par une carte électronique appelée carte ARDUINO.



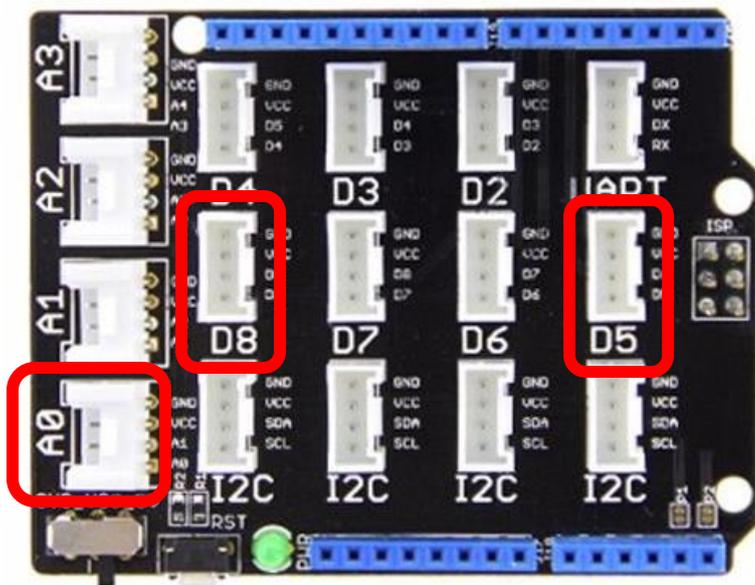
Figure 1 : carte de commande Arduino Uno avec shield Grove

Elle dispose également de capteurs, d'actionneurs et d'effecteurs.

Photo	Nom	Fonction	Capteur / Actionneur / Effecteur
	Un relai	Distribuer le courant de puissance vers l'effecteur. <i>(Il joue le rôle d'un interrupteur commandé.)</i>	Actionneur
	Un capteur de luminosité	Acquérir une quantité de lumière reçue (et la transformer en valeur analogique). <i>(Il capte l'intensité lumineuse extérieure.)</i>	Capteur
	Un détecteur de mouvement	Acquérir la présence d'un mouvement, ou non, (et la transformer par un code binaire). <i>(Il détecte un mouvement.)</i>	Capteur
	Une ampoule	Transformer l'énergie électrique en énergie lumineuse. <i>(Elle éclaire la rue.)</i>	Effecteur
	Batterie/pile	Alimenter en énergie électrique la partie opératrice (les effecteurs)	Alimentation

Ces éléments sont branchés sur la carte de commande Arduino à des emplacements précis.

Composant de la maquette	Repère sur le Shield Grove	Entrée/sortie	Type
Le relais (relié à l'ampoule) :	D8	Sortie	Digitale (0 ou 1)
Le détecteur de présence :	D5	Entrée	Digitale (0 ou 1)
Le détecteur de luminosité :	A0	Entrée	Analogique (infinité de valeur)



Remarque

Le numéro des connecteurs n'a pas d'importance (sauf pour le programme).

Mais le type oui ! Attention donc à la différence entre D / A / UART / I2C.

Figure 2 : Shield Grove



Pour info :

Les connecteurs Dx (2 à 8) sont des entrées/sorties de type « digitales ». Cela signifie que l'information qui passe par ces connecteurs est un **état logique** : 1 ou 0.

Les connecteurs Ax (0 à 3) sont des entrées de type « analogiques ». Cela signifie que l'information qui passe par ces connecteurs peut prendre une infinité de valeur entre 0 et 5V (alimentation électrique de la carte).

Seconde partie : programmation

Préparation de l'ordinateur :

La carte Arduino va être programmée à l'aide d'un logiciel : mBlock (qui ressemble beaucoup à Scratch). Le logiciel mBlock possède des blocs supplémentaires par rapport à Scratch :

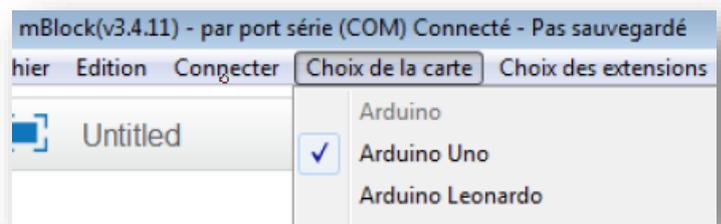


Figure 3 : exemple de blocs "nouveaux"

Ce sont ces « nouveaux » blocs que vous allez utiliser pour programmer la carte ARDUINO.

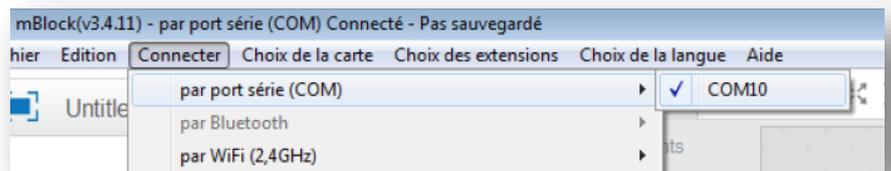
Suivre la procédure :

- 1) Sélectionner la carte Arduino dans le menu « Choix de la carte » (*par défaut cela risque d'être le robot mBot qui est actif*)

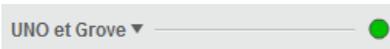


- 2) Relier la carte Arduino à l'ordinateur avec un cordon USB.

- 3) Dans le menu « Connecter », sélectionner le port COM.



Dans le bloc « Pilotage », une pastille verte doit alors apparaître.

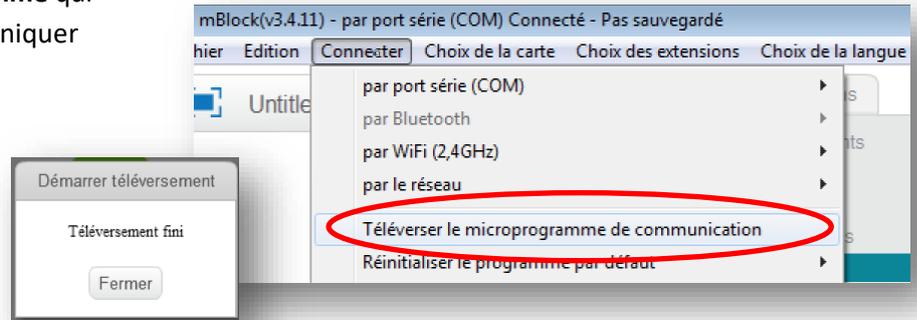


- 4) Pour la suite, deux possibilités s'offrent à vous :

2 modes possibles :	Différences	Avantages Inconvénients
Mode connecté : 	permet de modifier le programme et l'exécuter en live. (La carte ARDUINO devient alors une interface de programmation)	++ : rapidité / modification du programme en live / pas besoin de batterie pour la carte Arduino / les blocks Scratch fonctionnent -- : téléverser le programme de communication / la carte reste toujours connecté à l'ordinateur
Mode offline : 	le programme doit être chargé dans la carte Arduino pour être exécuté. (La carte Arduino devient une vraie carte de commande)	++ : autonomie, pas besoin d'ordinateur lors du fonctionnement -- : doit être téléverser à chaque modification / les blocks scratch ne fonctionnent plus (sauf Contrôle et Opérateurs) / nécessite une batterie

Je vous conseille très fortement le mode « Connecté ».

Donc, il faut **téléverser le microprogramme** qui permettra à la carte Arduino de communiquer avec le logiciel.

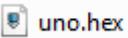
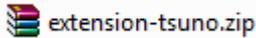


5) Plus compliqué ...

mBlock est nativement destiné à piloter les robots mBot. Il faut donc lui rajouter les blocs de pilotage de la carte Arduino mais surtout des modules Grove (car la maquette utilise cette technologie). Cela se fera en ajoutant une extension au logiciel mBlock.

L'extension utilisée pour cette maquette est celle de technologie-Service.

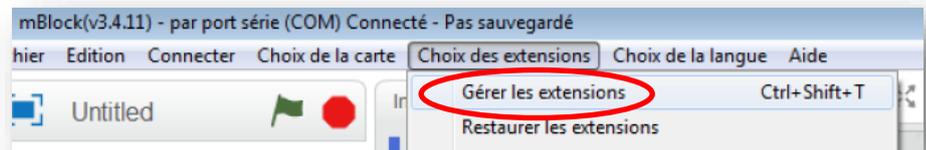
<https://www.technologieservices.fr/mblock-extensions-ts-3-4-11-ress-175416.html>

Seuls deux fichiers : « uno.hex » et « extension-tsuno.zip » nous seront utiles.  

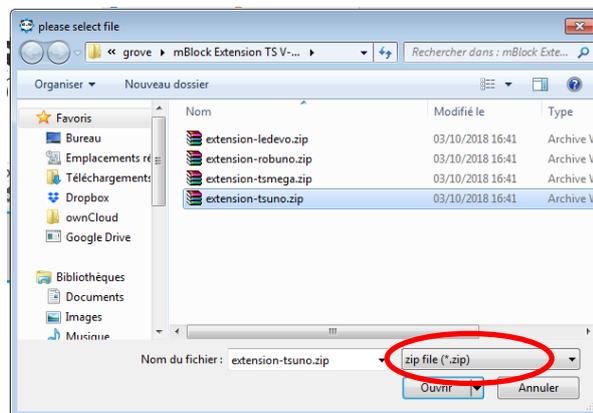
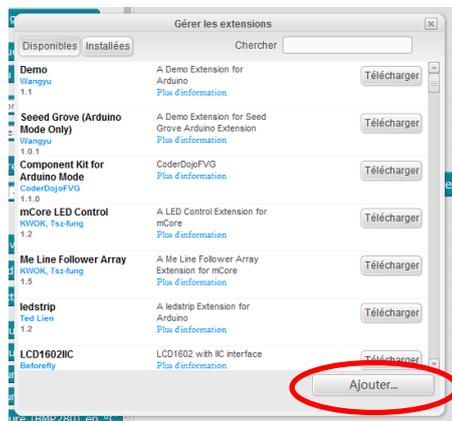
a) Fermer mBlock. Remplacer le fichier « uno.hex » par le nouveau dans le dossier « C:\Program Files (x86)\mBlock\tools\hex ». Relancer mBlock

b) Pour ajouter l'extension :

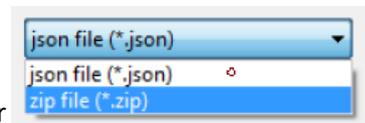
Cliquez sur « Gérer les extensions » du menu « Choix des extensions ».



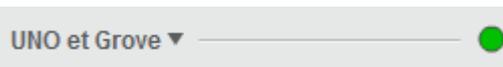
Cliquer sur « Ajouter » en bas à droite de la fenêtre.



Puis modifier le format du fichier à ajouter (en bas à droite) pour mettre (*.zip), sélectionner extension-tsuno.zip » et cliquer sur « ouvrir ».



L'extension sera maintenant visible dans les blocs « Pilotages »



Et voilà, tout est prêt à s'amuser avec la maquette !

Avant de programmer en entier la maquette vous allez passer par des étapes afin de mieux comprendre.

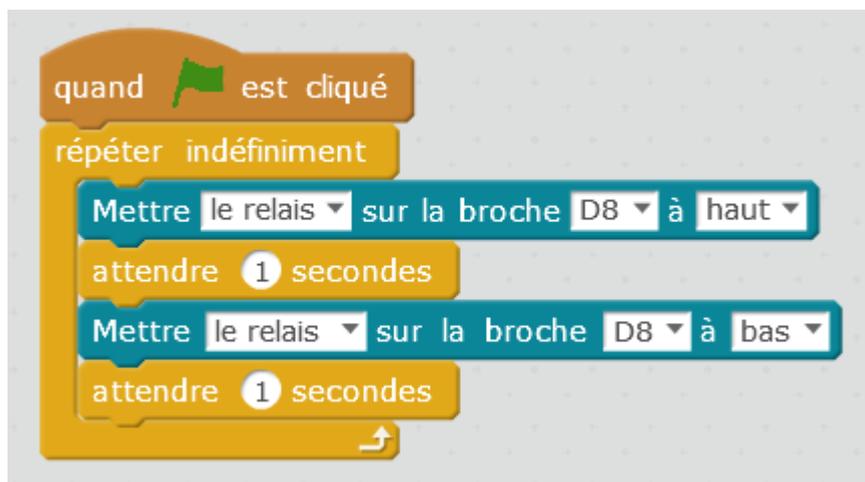
Etape 1 : Allumer et éteindre l'ampoule



Objectif : prendre en main mBlock et montrer l'interaction programme-maquette.

	description	Action sur la maquette
Eléments utilisés	le relais - l'ampoule	
Bloc(s) utilisé(s)		Active (haut) ou désactive (bas) l'actionneur branché sur la broche.
Bloc(s) à adapté(s)	 	Active le relais connecté sur la sortie D8. Désactive le relais connecté sur la sortie D8

Il est possible de réaliser le programme suivant pour tester le bloc :



Retour/bilan possible :

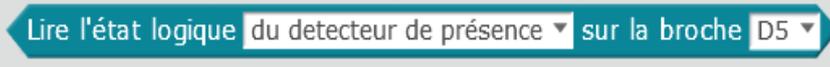
Le programme agit sur la réalité grâce à la carte de commande Arduino. Le relais (actionneur) s'active et laisse passer le courant (qui n'est pas celui de la carte Arduino) qui allume la lampe (effecteur).

Possibilité de parler de chaîne d'énergie

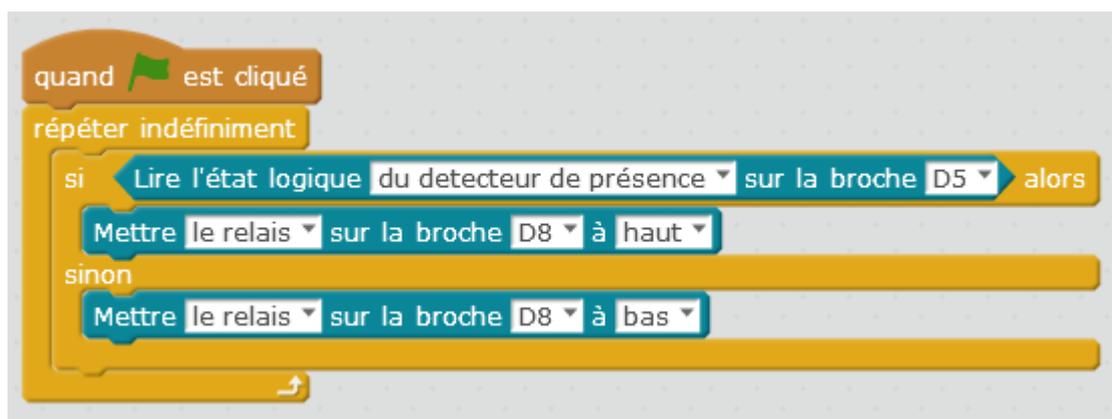
Etape 2 : Détecter le mouvement



Objectif : Utiliser le détecteur de mouvement afin d'allumer l'ampoule si passage d'un piéton.

	description	Action sur la maquette
Eléments utilisés	Détecteur de mouvement – relais - ampoule	
Bloc(s) utilisé(s)	   (mais pas obligatoire) .	<p>La forme de ce bloc doit aider les élèves à trouver dans quel autre bloc il peut s'utiliser.</p> <p>Le Bloc SI teste : soit la validité d'une opération, soit la présence d'un état logique haut (1)</p>
Bloc(s) à adapté(s)	 <p>Il faut comprendre ce bloc par : « l'état logique renvoyé par le détecteur de présence est ... (1 ou 0) »</p>	<p>Le capteur de mouvement envoie au programme une valeur : 1 ou 0 <i>1 s'il y a mouvement, 0 s'il n'y a pas de mouvement.</i></p>

Il est possible de réaliser le programme suivant pour tester le bloc :



Retour/bilan possible :

Une action (lampe qui s'allume) est conditionnée par un évènement réel (présence de mouvement). Le détecteur ne donne pas d'information sur la quantité de mouvement, mais juste présence d'un mouvement.

Possibilité de parler de chaine d'information

Etape 3 : Découvrir le capteur de luminosité



Objectif : découvrir la valeur analogique renvoyée par le capteur de luminosité

	description	Action sur la maquette
Eléments utilisés	Capteur de luminosité	
Bloc(s) utilisé(s)	 	
Bloc(s) à adapté(s)	 	La création d'une variable va permettre d'afficher à l'écran la valeur retournée par le capteur.

Il est possible de réaliser le programme suivant pour tester le bloc :



Sur la scène de mBlock, la valeur renvoyée par le capteur s'affiche.

Il est alors possible de prendre deux valeurs de référence : une quand il fera jour (lumière de la salle allumée par exemple). Une autre quand il fera nuit (lumière et volet fermés par exemple).

L'élève aura à déterminer une valeur médiane qui permettra de juger s'il fait nuit ou s'il fait jour (l'un OU l'autre)

Retour/bilan possible : le capteur de luminosité envoie une information qui varie en fonction de la luminosité ambiante. Cette information peut prendre une infinité (ou presque) de valeur. C'est une information analogique. Il est possible de mettre cette valeur dans une variable. Une variable est une boîte dans laquelle on peut y stocker un nombre afin de l'utiliser plus tard (ou de l'afficher sur mBlock). Le nom de la variable n'est pas important mais il faut qu'elle soit explicite.

Etape 4 : Faire fonctionner la maquette en mode jour/nuit



Objectif : programmer la carte afin que la lumière s'allume la nuit s'il y a du mouvement, mais pas le jour.

	description	Action sur la maquette
Eléments utilisés	Capteur de luminosité – détecteur de mouvement – relais - ampoule	
Bloc(s) utilisé(s)		La forme du bloc montre qu'il n'est pas possible d'intégrer la valeur dans une condition SI.
Bloc(s) à adapté(s)		

Il est possible de réaliser le programme suivant pour tester le bloc :

Retour/bilan possible : C'est top ! ;-) c'est mieux que la simulation de Scratth...

Annexe : Les chaînes d'information et d'énergie de la maquette

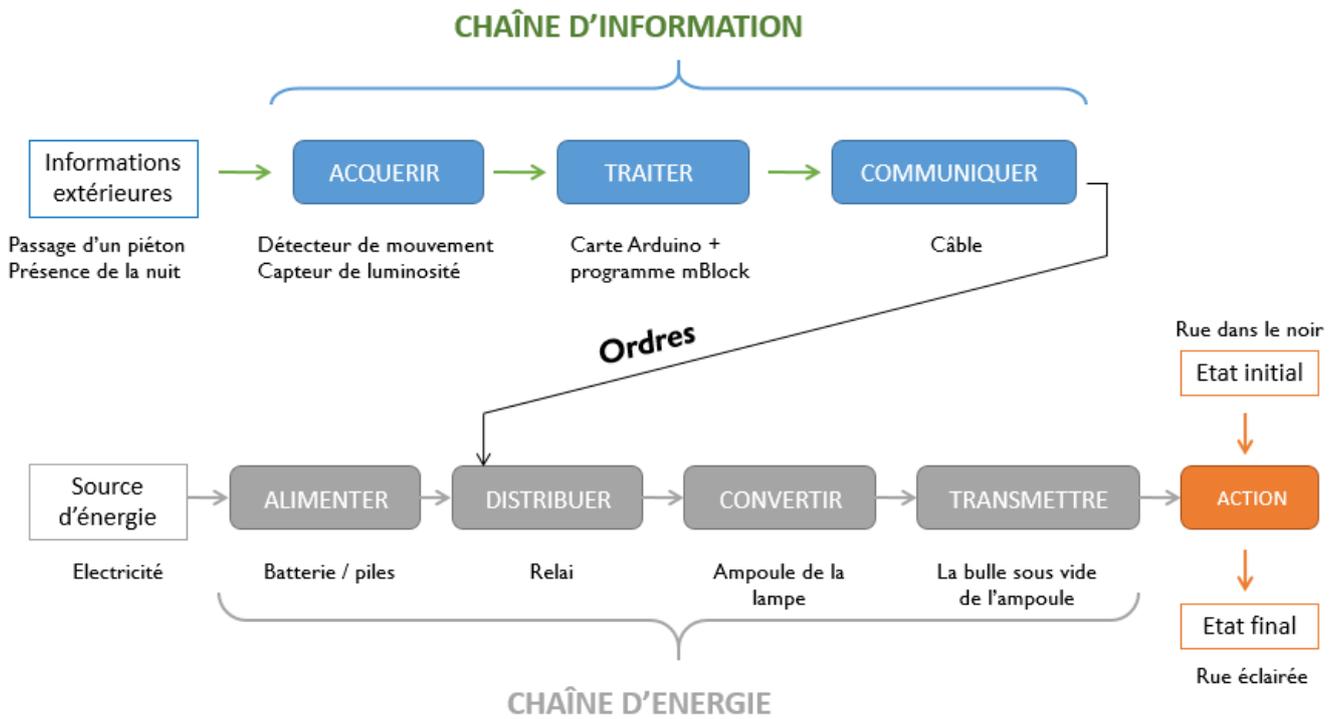


Figure 4 : chaînes d'information et d'énergie de la maquette de rue

Le fichier pptx permet de modifier le schéma ci-dessus.

