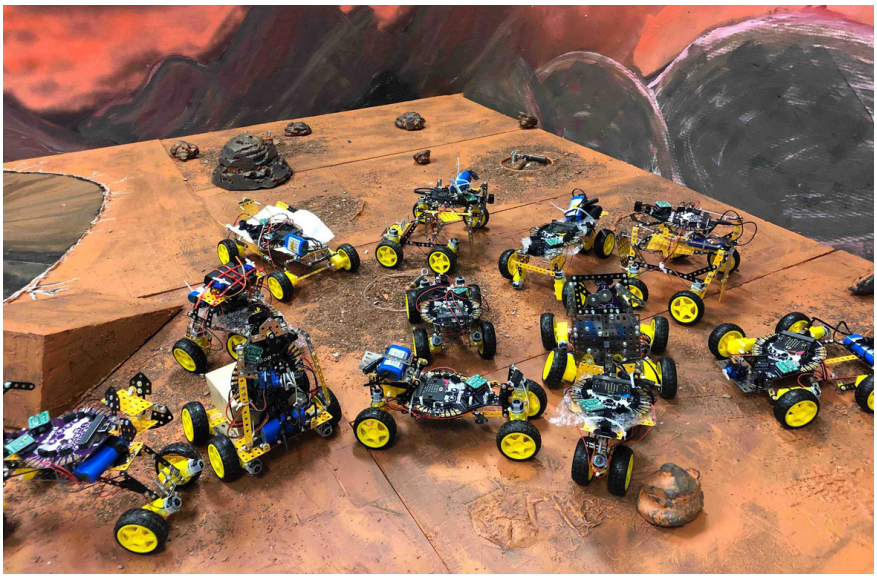
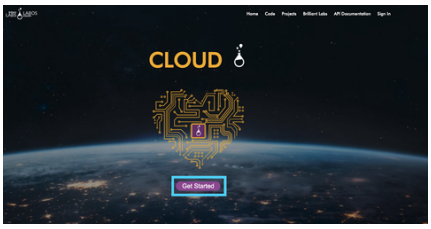
Se brancher au nuage

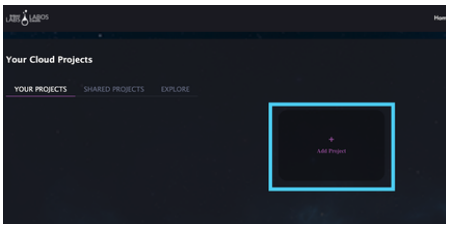
avec votre rover



La connexion au nuage des Labos Créatifs permet aux utilisateurs d'avoir une connectivité directe point à point. Cette connexion offre aux makers une connexion sécurisée pour créer. Il ne s'agit pas seulement d'un service de cybersécurité judicieux, mais il permet aux rovers de Mission:Mars de bénéficier de capacités IoT. Cela signifie que vous pouvez contrôler le rover que vous avez conçu depuis votre maison ou n'importe où avec une connexion Internet !

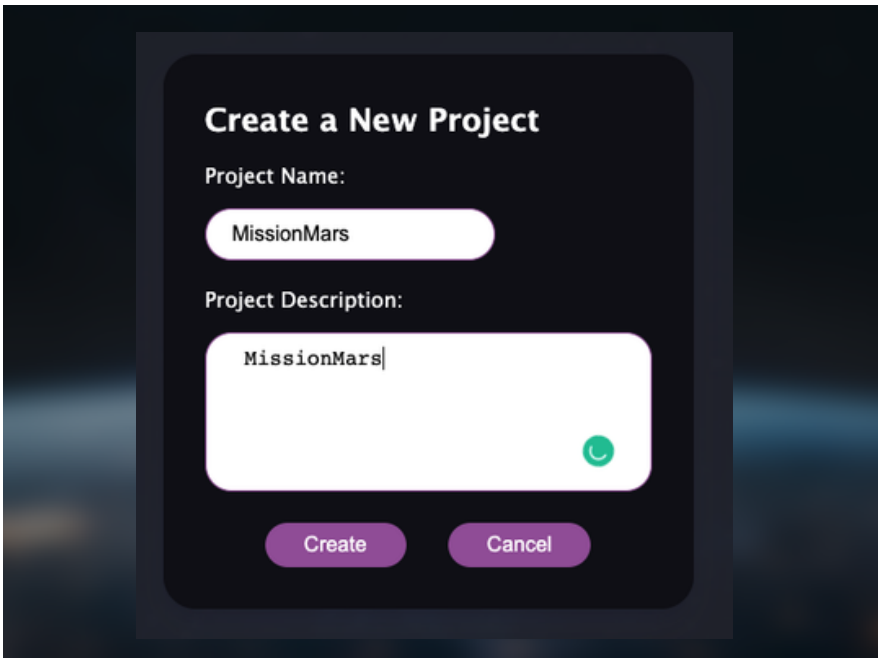
Créez un noauveau projet



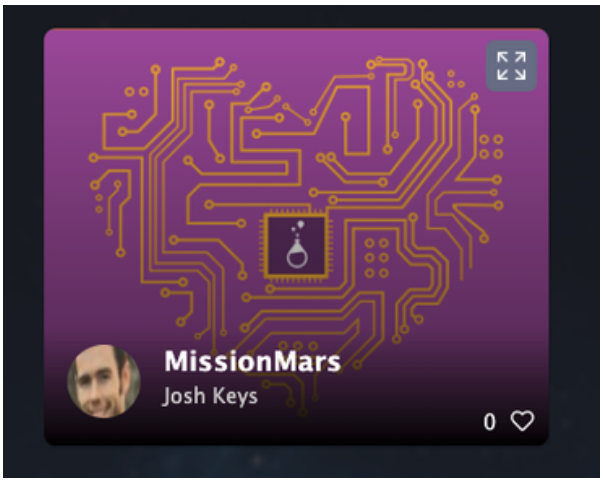


Dirigez-vous: cloud.brilliantlabs.ca et cliquez “Get Started” :

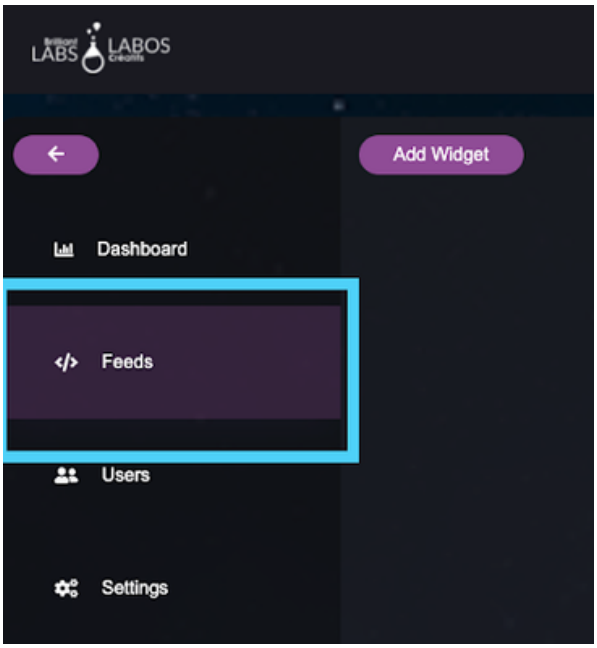
Donnez un nom et une description à votre projet (obligatoire)



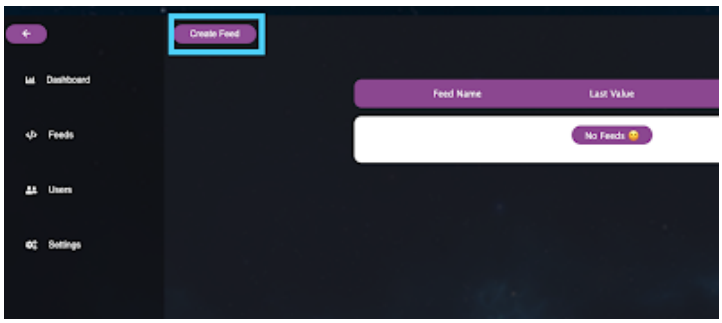
Ouvrez votre nouveau projet



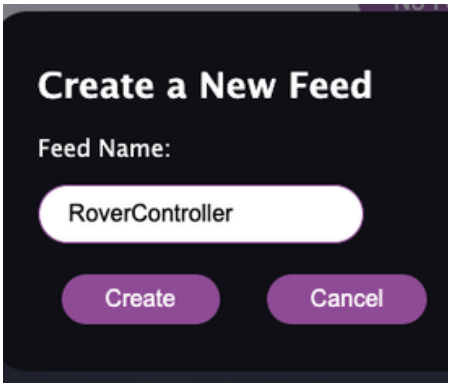
Sélectionnez l'onglet “Feeds”
Feed = flux



Sélectionnez “Create Feed”

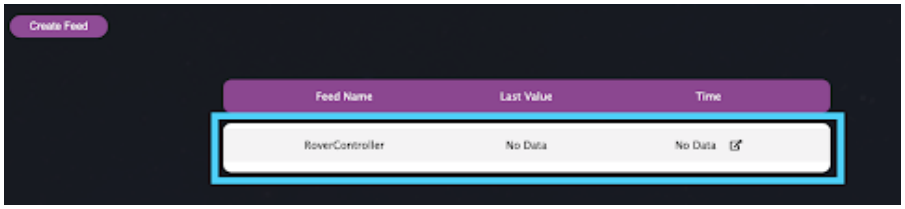


Nommez votre "feed" (flux) et cliquez “Create”

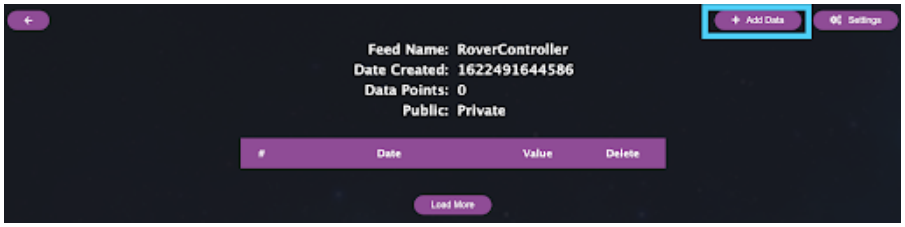


Félicitations ! Vous avez réussi à créer votre premier flux. Un flux est un endroit où des données (des chiffres comme la température, l'accélération ou l'humidité, etc.) peuvent être stockées. Nous allons utiliser ce flux pour stocker les commandes de notre Rover.

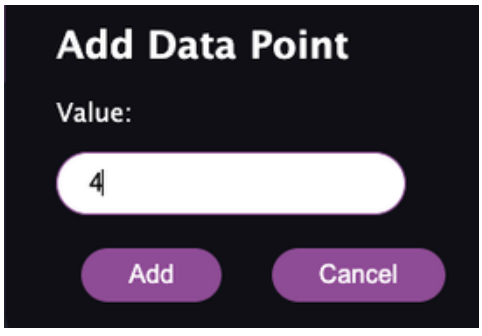
Nous pouvons ajouter des données à notre flux, une valeur à la fois, en cliquant dessus.



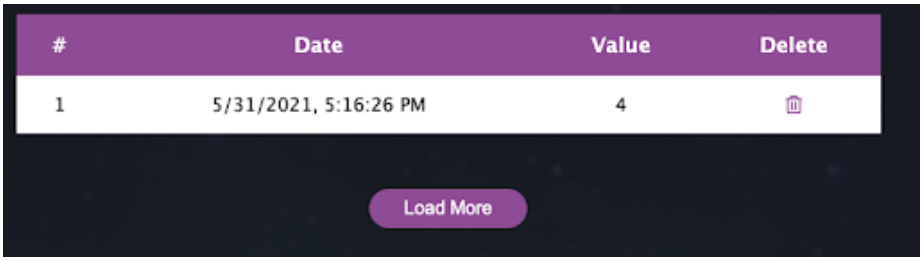
Sélectionnez ensuite le bouton "+ Add Data" dans le coin supérieur droit.



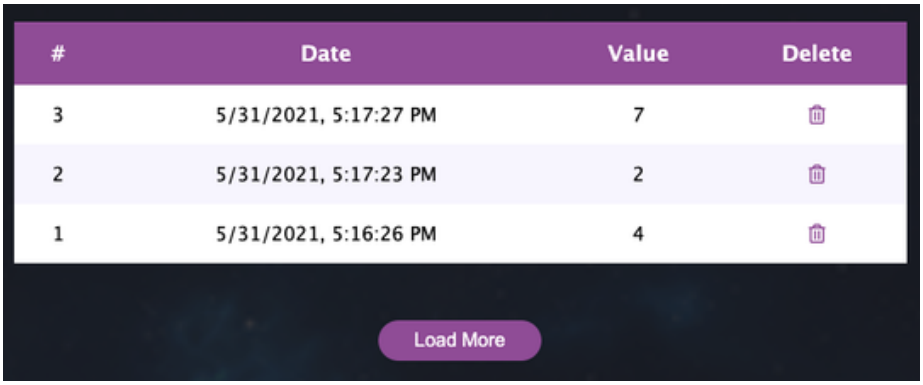
Ici, une valeur de 4 est ajoutée à nos données en entrant "4" et en sélectionnant "Add".



Comme vous pouvez le voir, la valeur a été ajoutée à notre flux.

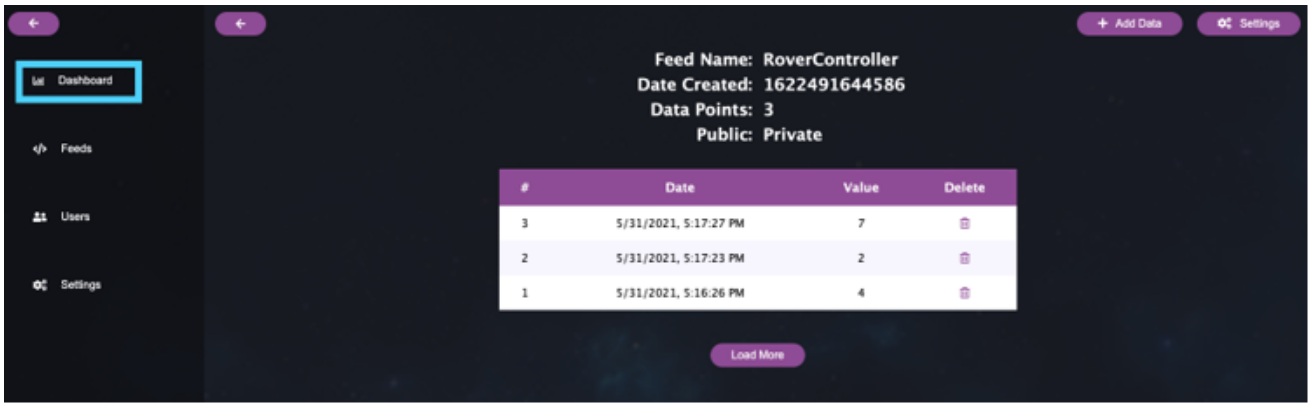


Ctrl



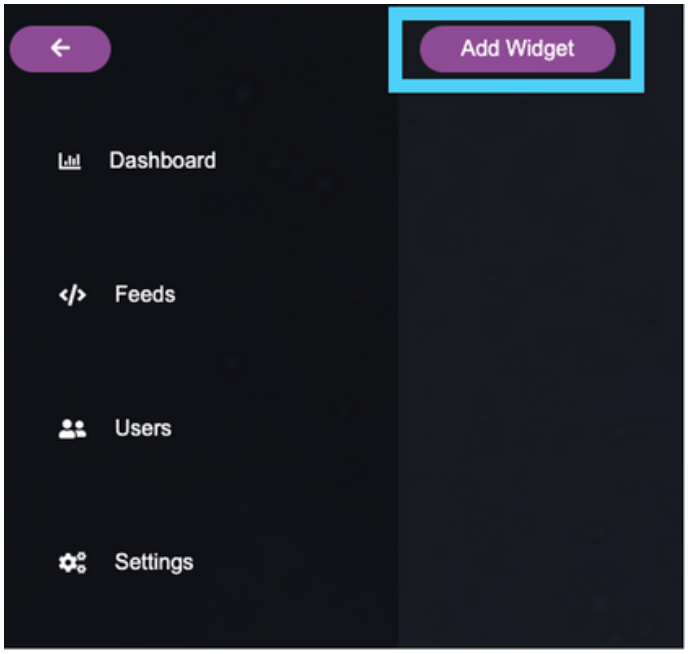
Chaque valeur est stockée dans le nuage en fonction du moment où elle a été envoyée. L'ajout de données de ce type peut s'avérer long et difficile, surtout lorsqu'on essaie d'utiliser ces données pour faire quelque chose comme contrôler un Rover, alors examinons un moyen plus simple !

Sélectionnez “Dashboard” sur le côté gauche de votre écran :

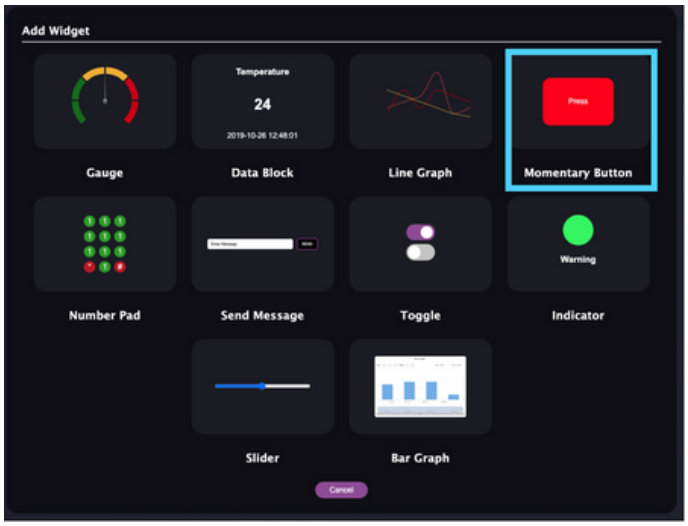


Le tableau de bord (dashboard) nous permet de regarder nos données et même de les contrôler avec des visuels beaucoup plus jolis et plus faciles à comprendre. Pour ce faire, nous utilisons ce que l'on appelle des "widgets".

Sélectionnez “Add Widget” en haut de votre écran :



Une fois ouvert, vous verrez une variété d'excellentes façons de visualiser et de contrôler vos données. Pour nous aider à piloter notre rover, nous allons sélectionner le “Momentary Button" en haut à droite :



Vous trouverez ici quelques options pour configurer votre ou vos boutons :

La première option est le "Feed". Comme nous voulons utiliser ces boutons pour contrôler notre rover, nous allons sélectionner le flux RoverController que nous venons de créer :

Pour la disposition des boutons, la disposition 1x2x1 est recommandée, mais c'est vous qui décidez.

Ci-dessous, nous avons également nommé notre Widget "Rover Remote Control" :

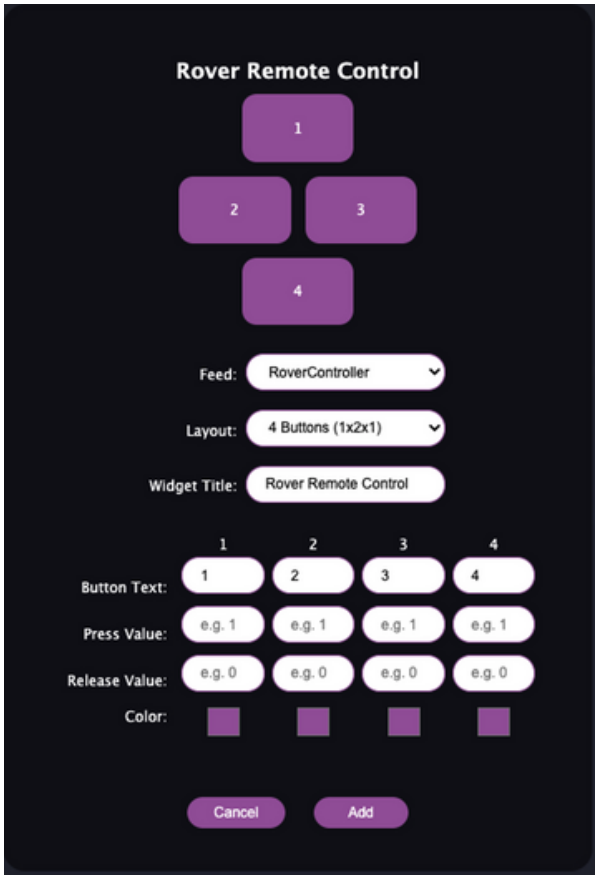
Nous avons maintenant 4 paramètres que nous pouvons définir pour chacun des boutons que nous avons :

Button Text: Le nom du bouton. Appelez-le comme vous le souhaitez.

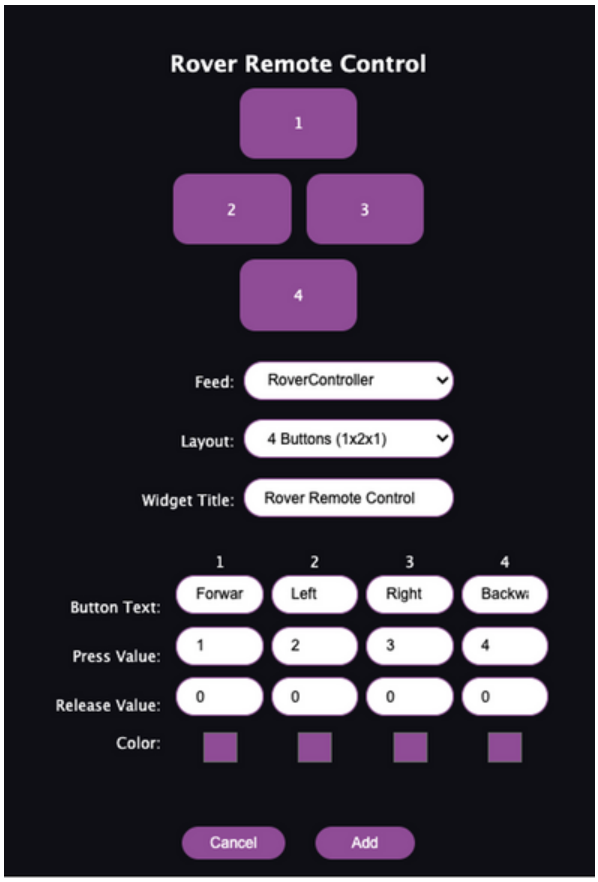
Press Value: La valeur que vous souhaitez que le widget écrive dans votre flux lorsque vous appuyez sur ce bouton.

Release Value: La valeur que vous souhaitez que le widget écrive dans votre flux dès que vous relâchez ce bouton.

Colour: La couleur de votre bouton.



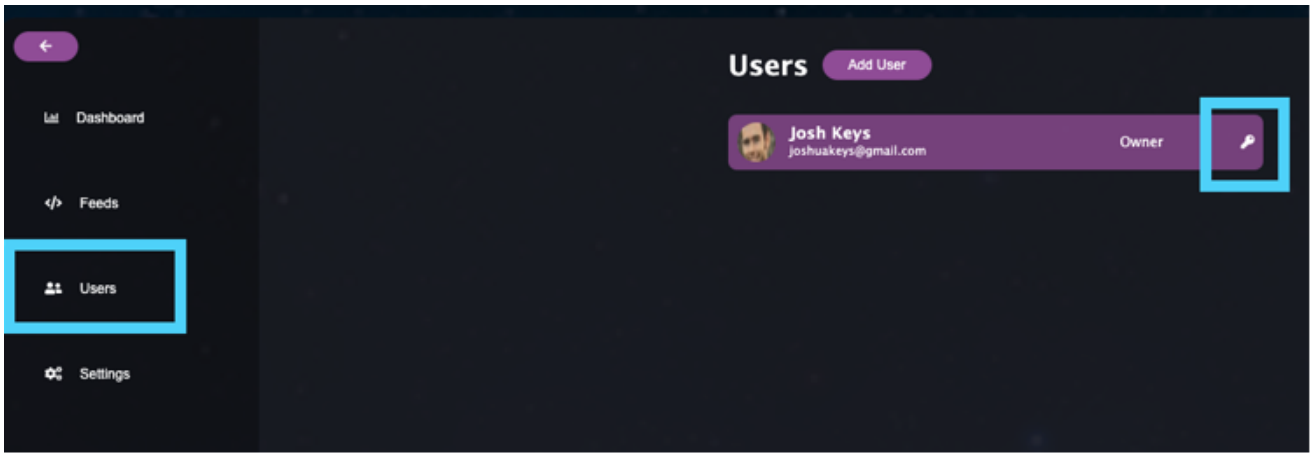
Voici un exemple de configuration pour contrôler notre rover :



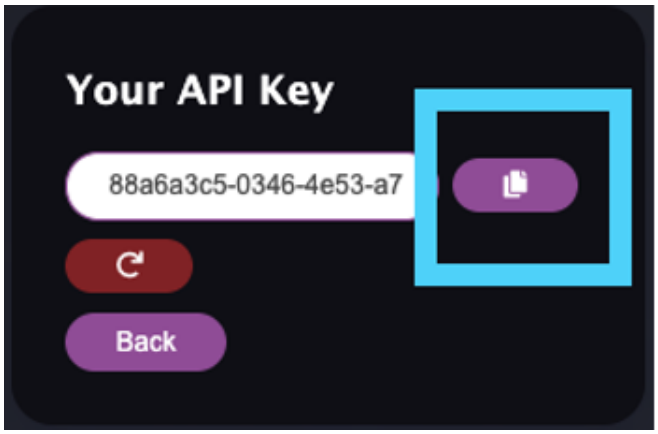
Lorsque le bouton "Forward" (Avancer) est enfoncé, la valeur "1" est inscrite dans l'alimentation du Rover Controller. Lorsque nous décidons de lâcher le bouton, la valeur "0" est inscrite. Lorsque l'on appuie sur le bouton "Left", une valeur de "2" sera écrite dans le flux du RoverController. Lorsque nous lâcherons le bouton, il écrira 0, et ainsi de suite.

Lorsque vous avez terminé la configuration de votre widget, sélectionnez "Add" et c'est tout. Vous avez maintenant votre contrôleur WiFi construit et prêt à être utilisé en appuyant sur le bouton directement dans votre tableau de bord :

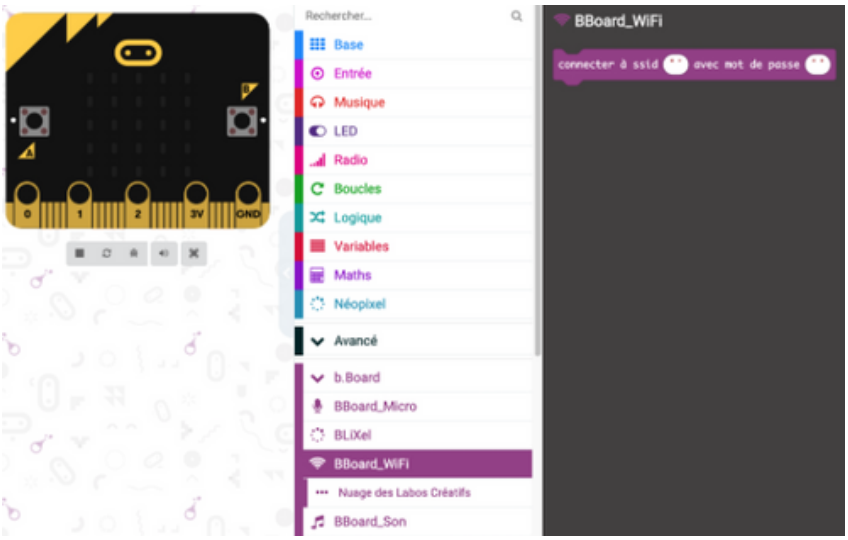
Avant de construire notre code, nous allons avoir besoin d'une information supplémentaire. Même si vous avez un nom unique pour votre widget et votre flux, il y a probablement beaucoup d'autres élèves qui utilisent exactement le même nom. Nous avons besoin d'un moyen de faire la distinction entre votre widget/feed et tous les autres. Pour ce faire, chaque compte dans notre nuage possède un mot de passe secret qui permet à son bBoard de se connecter à son nuage. Nous appelons cela une clé API et vous pouvez la trouver en naviguant dans l'onglet "Users" (utilisateurs) et en sélectionnant l'icône de la clé :



Une fois sélectionné, une fenêtre s'ouvrira avec votre clé API. Appuyez sur l'icône de copie comme indiqué ci-dessous pour la copier dans votre presse-papiers. Conservez-la dans un endroit sûr car nous en aurons besoin plus tard.



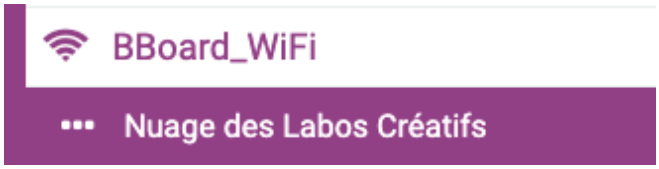
Il est maintenant temps de configurer le code sur le b.Board qui est connecté à notre Rover pour qu'il bouge lorsque ces boutons sont pressés.



Pour commencer, allez sur code.brilliantlabs.ca et ouvrez votre projet rover. Une fois sur place, naviguez jusqu'à la boîte à outils "b.Board" et sélectionnez la catégorie "BBoard_WiFi" :

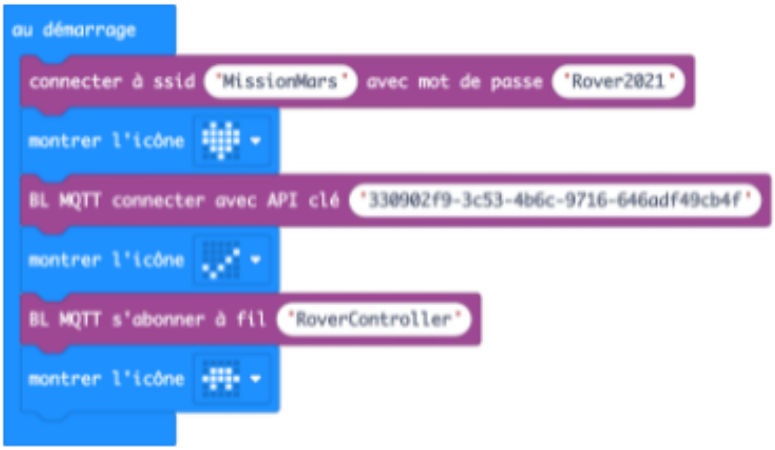
Sélectionnez le bloc "connect to ssid..." et déposez-le dans votre "au démarrage" et mettez votre SSID et mot de passe pour la connexion internet à laquelle vous voulez que votre rover se connecte pendant que vous le testez.
Avant de l'expédier, assurez-vous de le changer en SSID : MissionMars et Password : Rover2021 afin qu'il se connecte au réseau que nous avons mis en place sur Mars.

Maintenant que notre rover peut se connecter à Internet, nous devons le connecter à notre nuage Labos Créatifs afin d'utiliser le flux que nous avons créé précédemment. Pour ce faire, sélectionnez la catégorie "BBoard_WiFi" puis la sous-catégorie "Brilliant Labs Cloud" (Nuage Labos Créatifs) située en dessous.



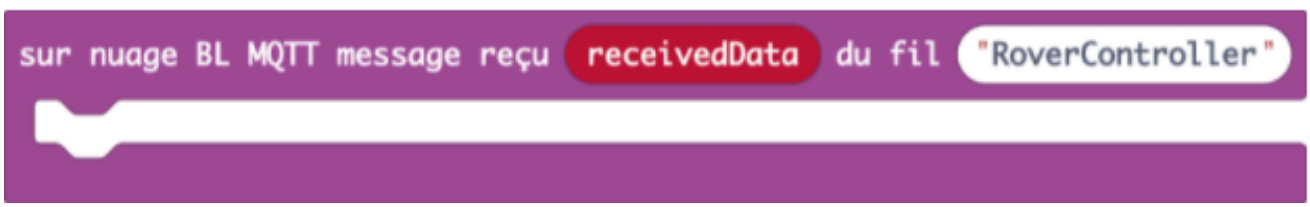
Il y a 4 blocs ici pour connecter et contrôler vos données via le nuage.

Nous allons nous connecter à notre BL Cloud (Nuage Labos Créatifs) personnel en mettant notre clé API de tout à l'heure dans le bloc BL MQTT connect. Ensuite, nous allons dire au rover d'écouter tous les messages/données qui proviennent du flux "RoverController" que nous avons créé plus tôt. Dans l'exemple ci-dessous, nous avons placé quelques icônes à afficher sur le micro:bit entre chaque étape pour nous indiquer ce qui se passe.

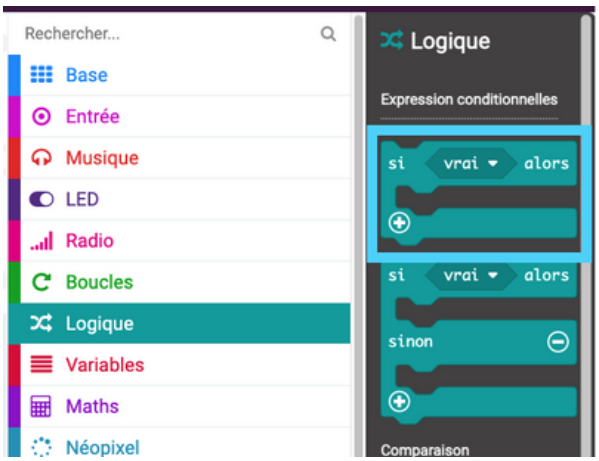


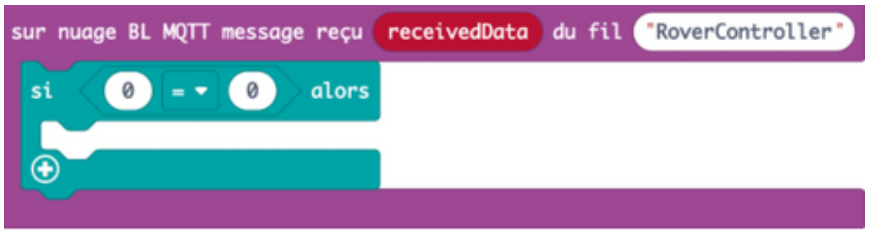
Votre rover est maintenant prêt à se connecter à votre nuage et à écouter les messages du flux RoverController. Mais nous avons besoin d'UNE dernière chose pour que tout cela fonctionne. Que se passe-t-il lorsque de nouveaux messages sont reçus par votre rover ? Nous devons ajouter un bloc qui reçoit ces messages et indique ensuite à votre rover ce qu'il doit faire !

Prenez le bloc "on BLCloud MQTT received..." et déposez-le n'importe où dans votre code

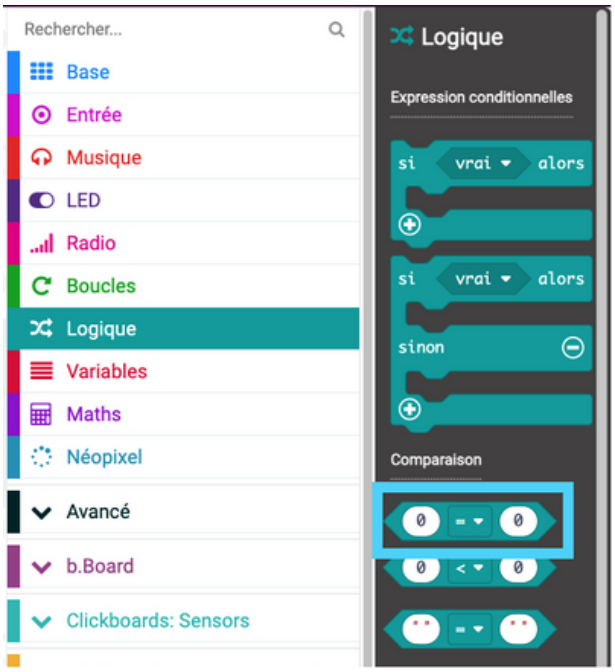


Nous allons placer tout ce dont nous avons besoin pour contrôler notre rover dans ce bloc. Nous devons appliquer une logique à notre rover afin qu'il puisse décider de ce qu'il doit faire en fonction du message qu'il reçoit du flux RoverController. Pour ce faire, prenez le bloc "If true then" (si vrai) et insérez-le dans votre bloc MQTT.

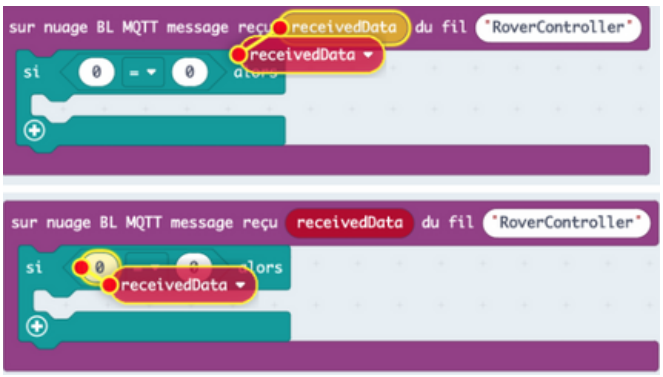




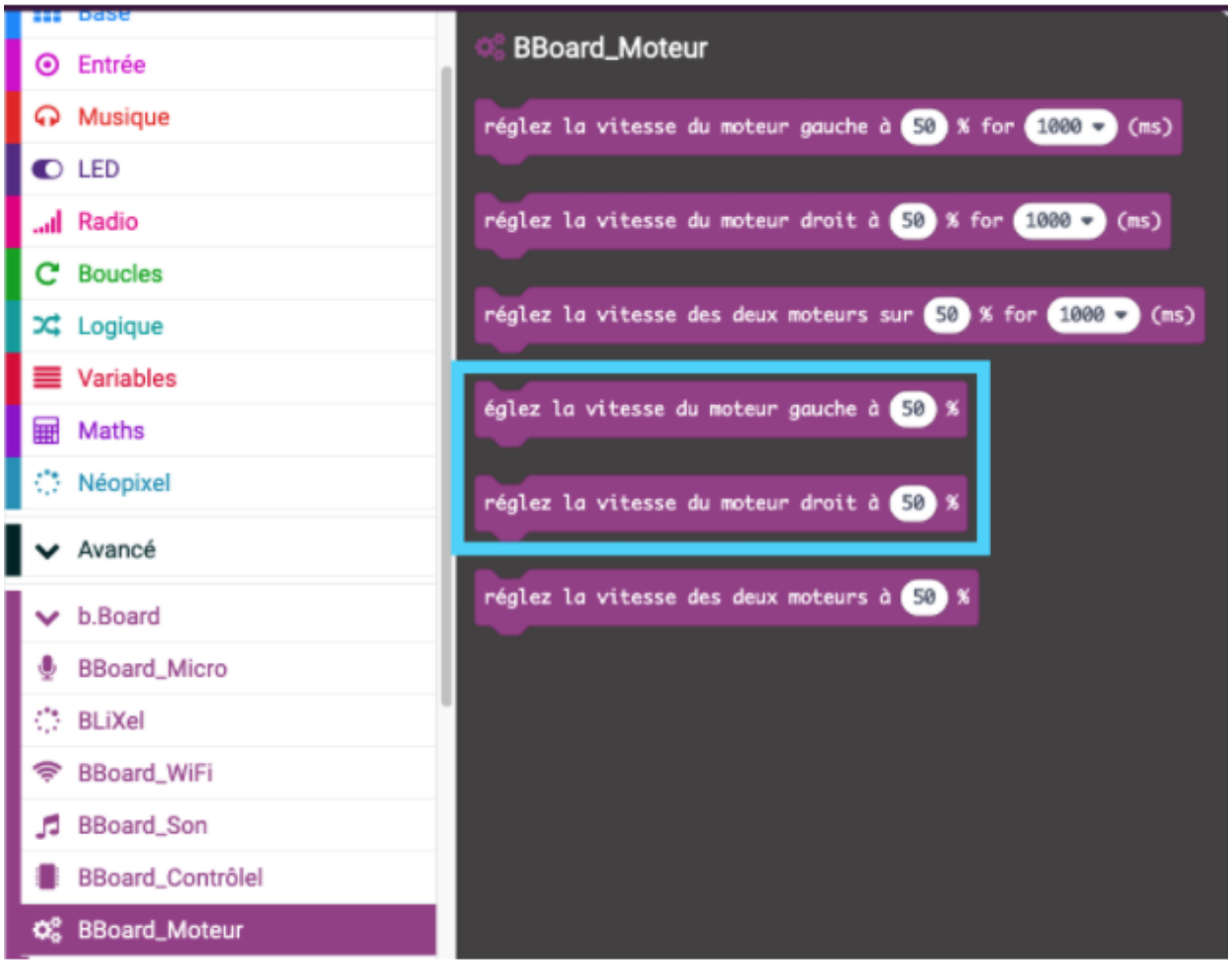
Ensuite, prenez le bloc "comparaison" et déposez-le dans le bloc "if" (si) que vous venez d'ajouter.

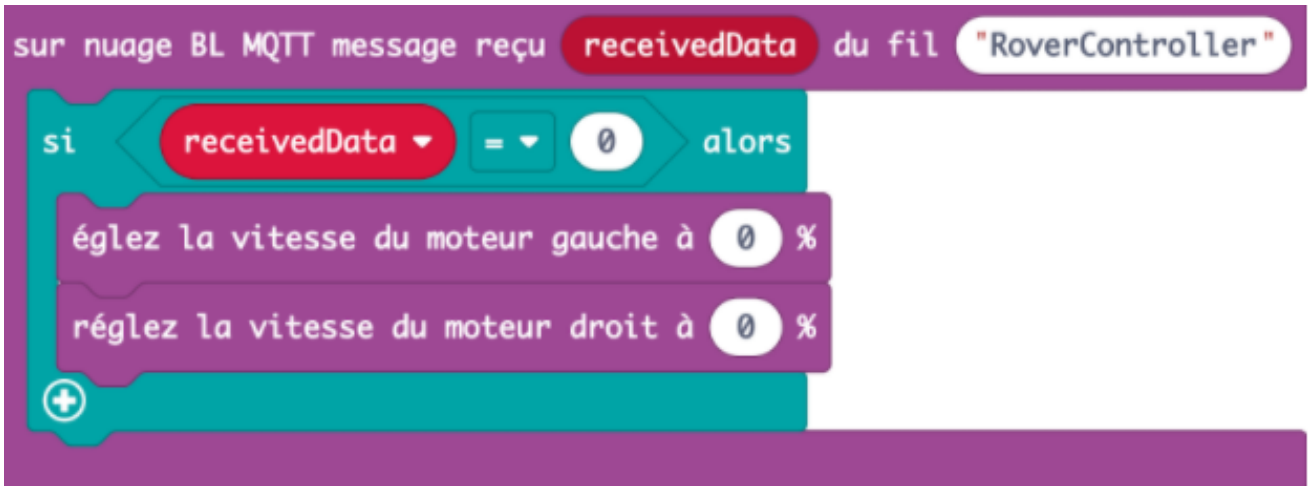


Et maintenant, prenez la variable "receivedData" et faites-la glisser dans votre bloc de comparaison :



Votre rover a maintenant son premier morceau de code logique du nuage. Il dit "Si le message que nous venons de recevoir du nuage (recievedData) est égal à 0, alors exécutez le code ci-dessous". Si vous vous souvenez, nous avons dit à notre widget Rover Controller d'envoyer un 0 à chaque fois qu'un bouton était relâché, alors programmons notre rover pour qu'il s'arrête chaque fois qu'un 0 est reçu. Pour ce faire, accédez à la catégorie BBoard_Motor et récupérez les blocs "set left motor to speed to ..." et "set right motor speed to ...", placez-les dans notre bloc "if" et réglez leur vitesse sur 0 pour l'arrêter :





Pour terminer notre contrôleur, dupliquez votre bloc "if" et couvrez chaque condition pour une valeur que le flux RoverController peut envoyer. Dans l'exemple ci-dessous, nous avons fourni du code pour que les moteurs avancent, reculent, tournent à gauche, à droite et s'arrêtent en fonction des valeurs envoyées par nos widgets lorsque les boutons sont pressés. N'hésitez pas à utiliser la même configuration pour votre rover, ou à faire quelque chose de complètement différent, c'est vous qui décidez!