

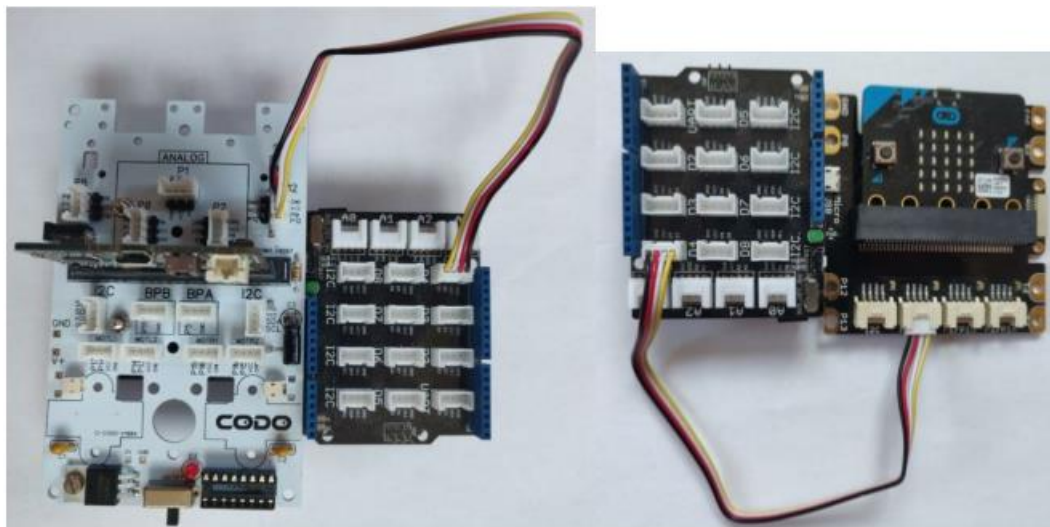
Valider des compétences réseau au collège à l'aide d'une carte Microbit et une programmation par blocs facilement.

Cet article propose une solution pour permettre à nos élèves de réaliser des activités autour des compétences de réseau.

Pour cela l'utilisation de l'outil de développement Makecode en programmation par blocs est très simple avec la carte Microbit.

La solution proposée est basée autour d'une carte Microbit connectée à une carte D1R32 (ESP32).

Il y a plusieurs solutions. Connecter la carte Microbit à une carte Codo elle-même connectée à un shield grove + D1R32, ou en utilisant un autre shield Microbit. Au choix.



La carte Microbit ne propose pas de solution de connexion Wifi. C'est la carte ESP32 qui fera le pont Wifi de façon totalement transparente pour nos élèves.

L'objectif est de proposer des commandes très épurées et très simples pour nos élèves, afin de leur permettre d'envoyer, et de recevoir une donnée, sur un réseau sans fils TCP/IP, le plus simplement possible.

Une extension est à ajouter dans Makecode pour qu'ils puissent programmer avec très peu de blocs un système pour envoyer et recevoir des données. Puis il suffira d'utiliser les commandes proposées pour envoyer et recevoir une donnée sur un réseau IP.

1. Mise en place (qui ne se fait qu'une seule fois !).
 - a) Téléchargement d'un programme dans la carte D1R32.

Commencez par connecter la carte D1R32 via un câble USB au PC.

Lancez le logiciel Arduino. Vous pouvez le télécharger ici :

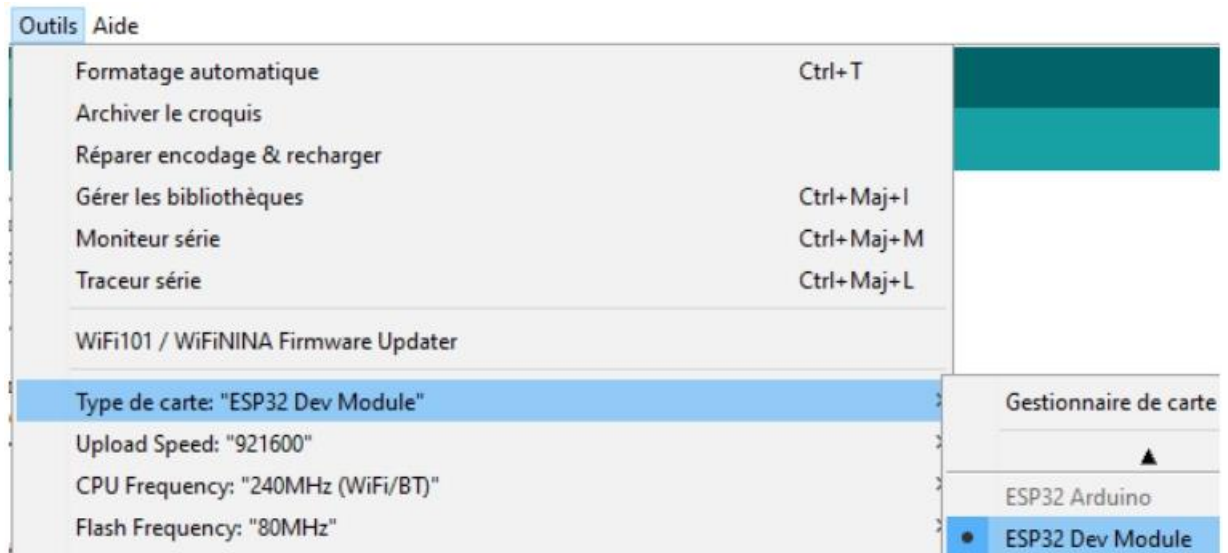
<https://www.arduino.cc/en/software>

Téléchargez le programme esp32_iot.ino disponible ici :

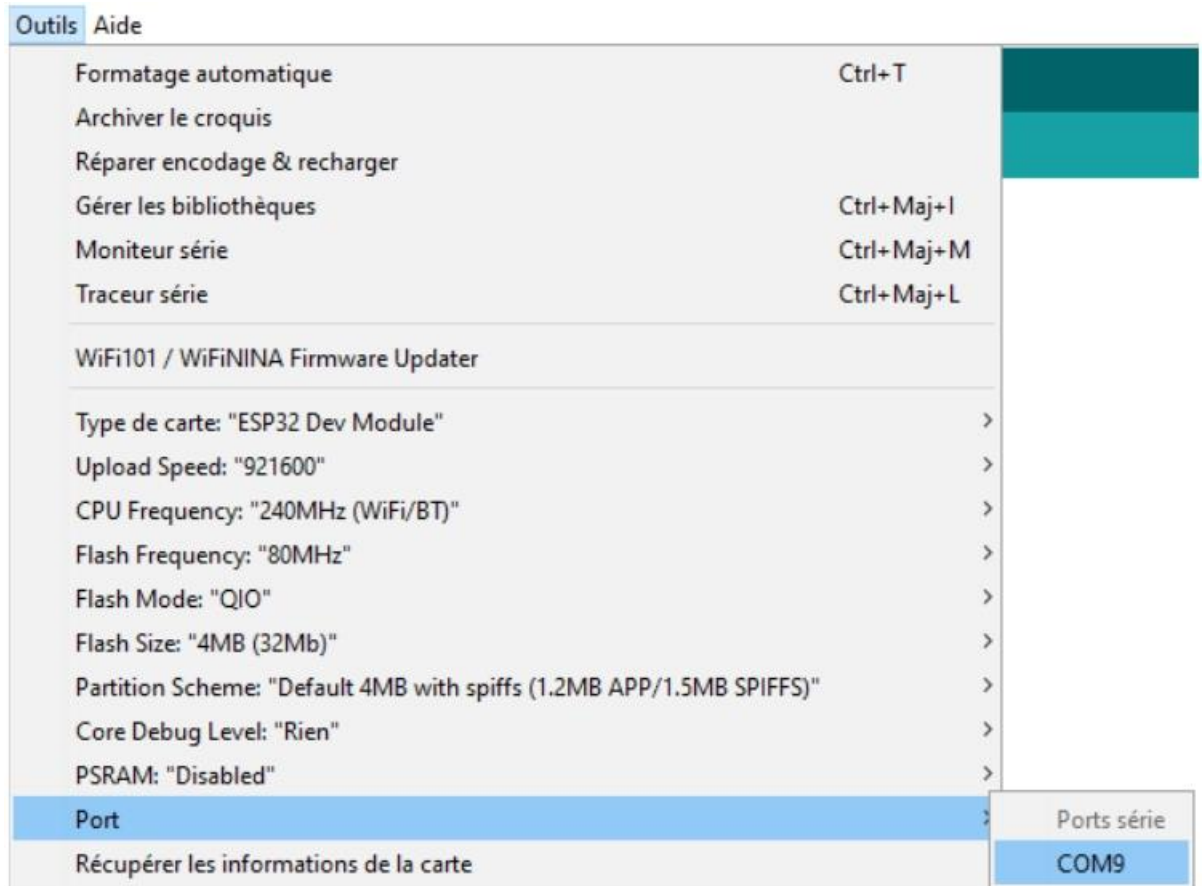
technologiecollege.atwebpages.com/images/domaines/informations/reseaux/wifi_microbit/esp32_iot.ino

Puis cliquez sur fichier / ouvrir, et ouvrir le fichier téléchargé.

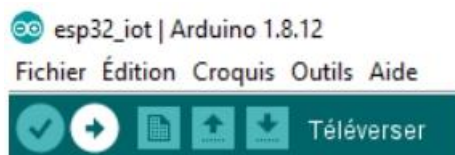
Cliquez sur outils/type de carte/ESP dev module et sélectionner la carte



Cliquez sur outil/port/ et choisir le port COM (il n'y en a qu'un seul en général)

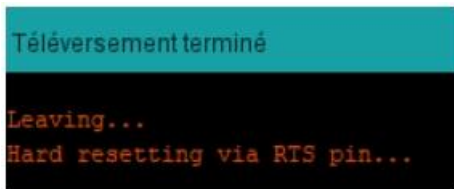


Cliquez sur la flèche téléverser. Le fichier se téléverse dans la carte D1R32.



Cela prend un peu de temps !

La fin du téléversement est indiqué par :

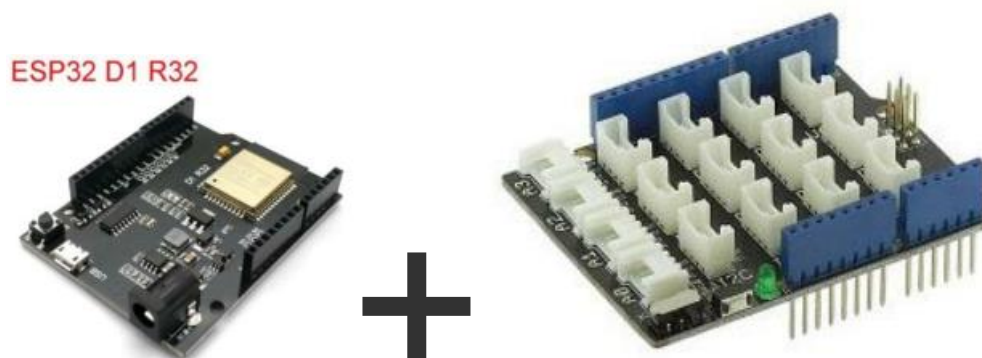


Il ne sera plus nécessaire de faire cette manipulation, le programme sera toujours présent dans votre carte D1R32, et les élèves pourront toujours l'utiliser sans soucis. Donc cette manipulation n'est qu'à faire au départ.

b) Connexion carte Microbit -> D1R32.

Connectez une carte d'extension comme la carte Codo ou une autre carte d'extension à la carte Microbit. Cette carte va permettre de faire une liaison avec la carte D1R32 et de permettre de connecter des capteurs et actionneurs supplémentaires à la carte Microbit.

Connectez le Shield Grove type Arduino à la carte D1R32.



Connecter un câble sur le port D4 du Shield Grove qui est déjà connecté à la carte D1R32.

Si vous utilisez une carte Codo : connecter l'autre bout du câble au port P12.

Si vous utiliser une carte d'extension basique Grove. Connecter alors l'autre bout du câble au port P0/P14.

Votre système est près maintenant à être utilisé par vos élèves. Vous pouvez mettre ces 2 cartes dans un même boîtier par exemple (projet de 4eme ou 3eme au passage).

Les élèves doivent penser que la carte Microbit peut dialoguer en Wifi sur un réseau.

2. Rappels sur le fonctionnement d'envoi et de réception de données sur un réseau IP.

Pour envoyer et recevoir une donnée sur un réseau IP on utilise en général la structure client / serveur.

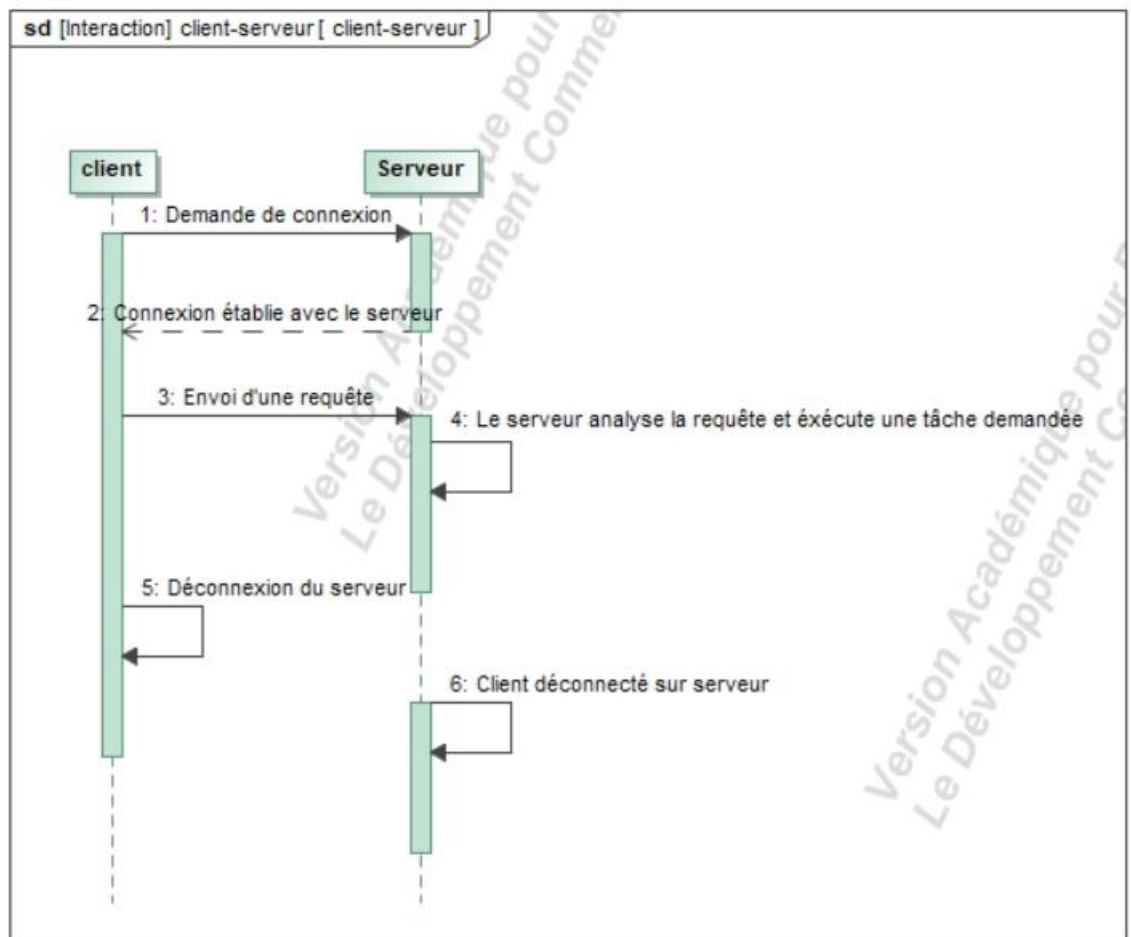
Lorsque vous allez au restaurant, vous sollicitez un serveur pour commander votre repas. Donc vous envoyez une requête (demande) au serveur. Le serveur vous répond en vous apportant ensuite le repas demandé. Le serveur est toujours à votre écoute et disponible. De plus un même serveur peut servir plusieurs clients.



© CanStockPhoto.com - csp44772836

Sur un réseau, c'est la même chose, il faudra définir qui est le client et qui est le serveur.

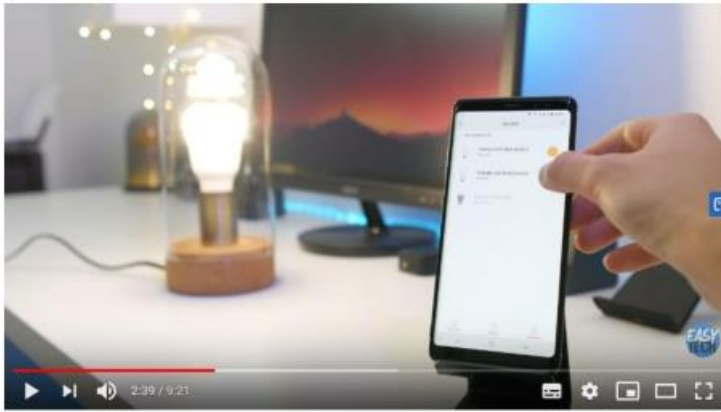
Diagramme de séquence typique de dialogue entre un client et un serveur.



3. Exemple concret d'application

On souhaite réaliser une lampe connectée.

<https://youtu.be/NaMcHT884rU?t=136>



Pour cela la lampe pourra être commandée à distance sur un réseau IP à l'aide d'un interrupteur connecté.

Pédagogiquement, la lampe sera la carte Microbit (+ ESP32) qui comporte un panneau à LED que l'on pourra allumer ou éteindre. L'interrupteur commandé sera une autre carte Microbit (+ ESP32) sur laquelle on utilisera le bouton A pour allumer et B pour éteindre, elle jouera le rôle d'un smartphone qui commande la lampe.

4. Schéma du réseau.

Vous aurez besoin d'une carte Microbit + ESP32 qui jouera le rôle de la lampe connectée



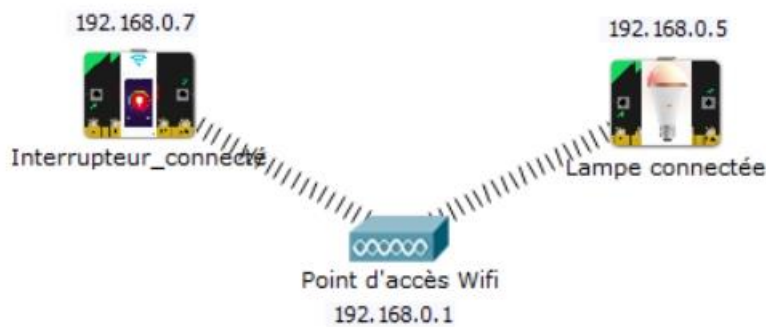
Vous aurez besoin d'une carte Microbit + ESP32 qui jouera le rôle de l'interrupteur connecté.



Il ne faudra pas oublier le point d'accès Wifi.



<https://www.a4.fr/routeur-wi-fi-alimentation-usb.html>



5. Identifier qui est le client, et qui est le serveur.

Le client sera l'interrupteur, le serveur sera la lampe qui attend des commandes pour obéir. Attention, un serveur n'est pas obligé de répondre à un client, il peut simplement exécuter une tâche sans rendre compte.

Donc l'objectif est que le client envoie une donnée à son serveur. Sans que le serveur réponde.

6. Notion de protocole

Mais quelle donnée sera envoyée par l'interrupteur connecté ? On pourra faire réfléchir les élèves à la notion de protocole de communication. Que doit envoyer comme donnée l'interrupteur à la lampe ?

Exemple on pourra envoyer « ON » pour l'allumer et « OFF » pour l'éteindre par exemple, mais on pourrait aussi choisir d'envoyer 0 ou 1, il y a plusieurs solutions. Voir ce que vont proposer les élèves.

Vous pouvez aussi tester ce protocole en mode débranché avec vos élèves. Vous distribuez un protocole différent à vos élèves par îlot. Vous leur demandez d'effectuer ce qui est noté sur la fiche protocole en fonction des commandes que vous allez donner à l'oral.

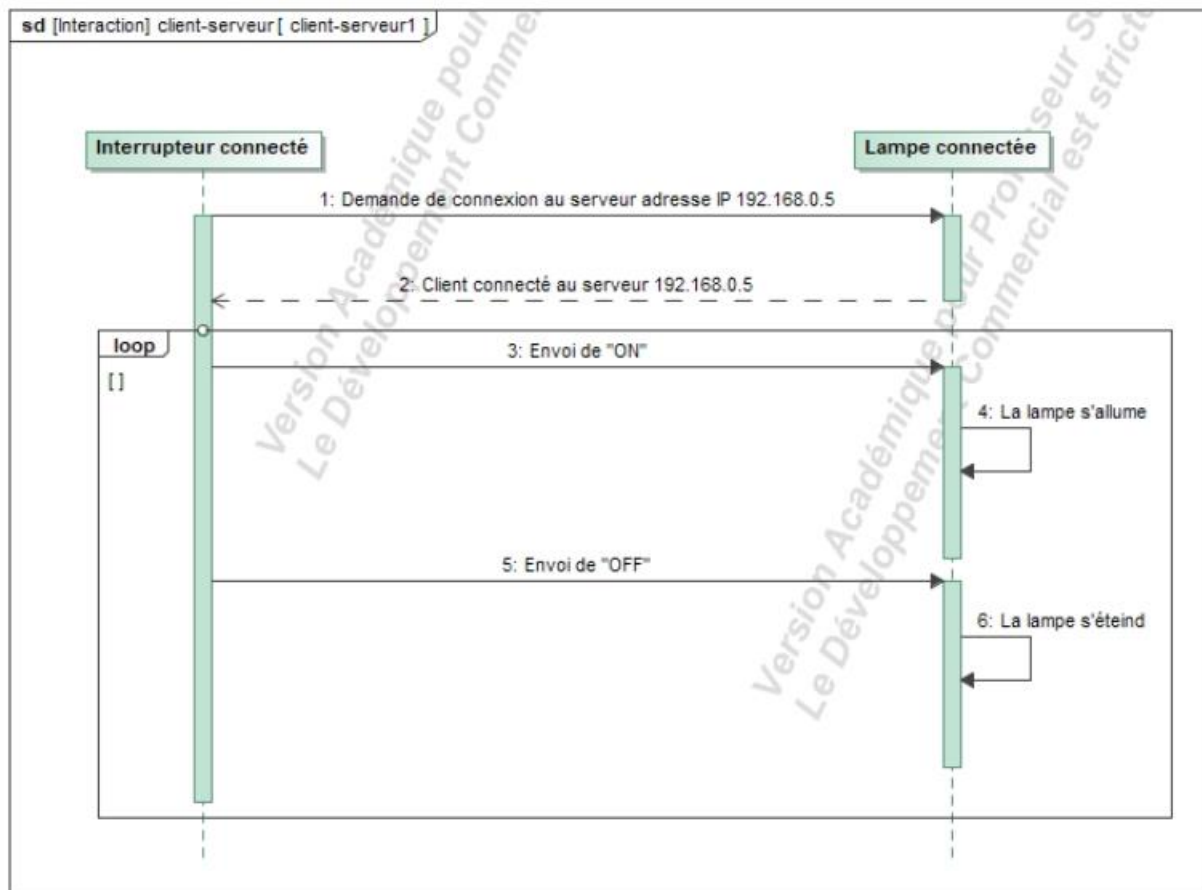
Puis vous dictez une commande, l'îlot doit interpréter la commande et exécuter la tâche. Chaque îlot doit voir si tous les îlots de la classe font la même chose ? Les élèves doivent noter ce qu'il se passe pour chaque commande envoyée. Attention, bien envoyer la même commande à chaque îlot les uns après les autres tranquillement, pour que les élèves aient le temps de noter ce que fait le serveur. De plus un client ne se connecte pas à plusieurs serveurs en même temps !

Commande envoyée par le prof (client)	Tâche effectuée par l'îlot 1 (serveur)	Tâche effectuée par l'îlot 2 (serveur)	Tâche effectuée par l'îlot 3 (serveur)	Tâche effectuée par l'îlot 4 (serveur)
A				
12				
H1				

Les élèves verront bien ainsi que si le client et le serveur n'ont pas les mêmes protocoles, ils ne pourront pas échanger des données et se comprendre.

Nous avons vu que le serveur devait être disponible avant le client. Donc il faudra être très vigilant sur le fait que le serveur soit lancé avant le client ! Cela peut faire l'objet d'une petite activité pédagogique pour le montrer aux élèves. Testez les deux cas. Démarrez le client, puis ensuite le serveur et montrer que rien ne fonctionne. Démarrez d'abord le serveur, puis ensuite le client, et là le serveur reçoit bien la demande du client.

Diagramme de séquence du système.



7. Programmation du serveur (lampe) et du client (interrupteur connecté)

Passons aux choses concrètes, la programmation des deux systèmes.

a) Programmation de la lampe (serveur).

L'algorithme est très simple.

DEBUT

Se connecter au point d'accès Wifi SSID et mot de passe

Fixer l'adresse IP de la lampe à 192.168.0.5

REPETER toujours

Donnee_recue ← Attendre une donnée d'un client

SI donnee_recue = « ON » ALORS

Allumer panneau à led

SINON SI donnee_recue = « OFF » ALORS

Eteindre panneau à LED

FIN SI

FIN REPETER

FIN

Il suffit maintenant de traduire cet algorithme en programme Blockly avec Makecode.

Lancez makecode.microbit.org

Allez dans avancé / extension

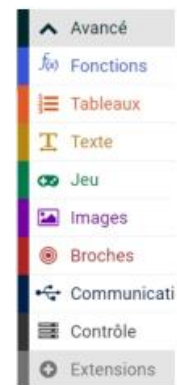
Puis saisissez l'adresse de l'extension suivante :

Pour les personnes qui utilisent le shield grove basique

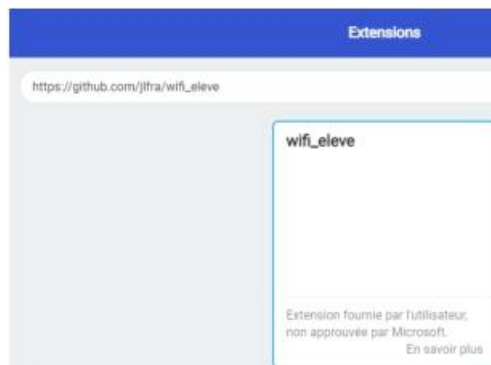
https://github.com/jlfra/wifi_eleve

Pour ceux qui utilisent la carte Codo :

https://github.com/jlfra/wifi_eleve_codo

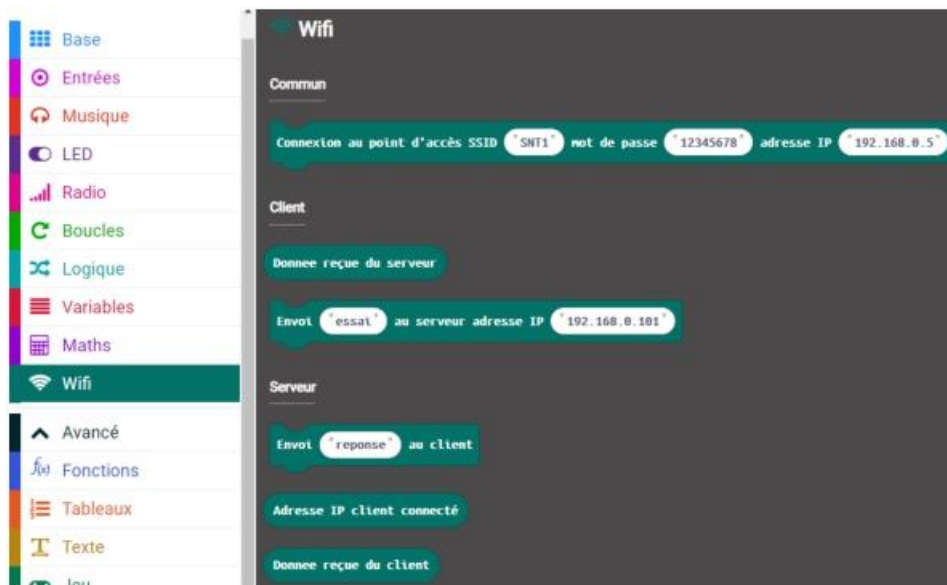


L'extension wifi apparait alors dans la liste des blocs.

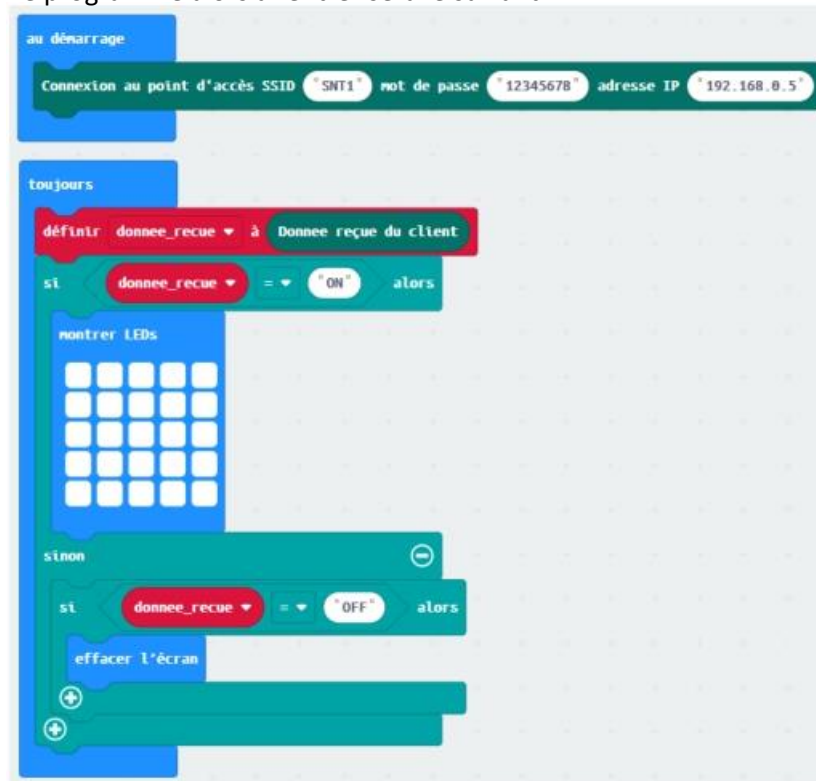


Vous pouvez maintenant utiliser les blocs suivants pour programmer vos objets connectés très facilement.

Pour ceux qui utilisent la carte Codo, il faudra utiliser le connecteur P12.



Le programme alors à rentrer sera le suivant :



Ce programme lampe_connectee_serveur.hex est aussi fourni.

b) Programmation de l'interrupteur connecté (client)

L'algorithme est là encore simple :

DEBUT

Se connecter au point d'accès Wifi SSID et mot de passe

Fixer l'adresse IP de la lampe à 192.168.0.7

REPETER toujours

SI bouton A appuyé ALORS

Envoyer « ON » au serveur 192.168.0.5

FIN SI

SI bouton A appuyé ALORS

Envoyer « OFF » au serveur 192.168.0.5

FIN SI

FIN REPETER

FIN

Le programme sur Microbit.

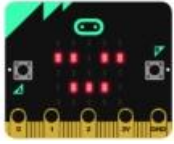


8. Essais

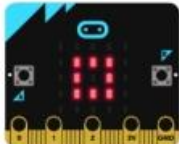
Vérifiez bien que votre point d'accès est bien activé (vérifiez-la à l'aide de votre smartphone en vérifiant qu'il est bien visible et que vous pouvez vous connectez avec votre smartphone). Vérifiez l'adresse IP de votre smartphone pour voir l'adresse réseau et ne pas faire d'erreur sur les adresses IP de votre lampe connectée ainsi que pour votre interrupteur connecté.

Lancez la lampe connectée. Pour cela, alimentez la carte D1R32, par exemple avec une prise USB. Inutile de connecter aussi la carte Microbit, car l'alimentation passera de la carte D1R32 vers la carte Microbit.

Vous devez voir plusieurs icônes défiler sur l'afficheur de la carte Microbit.

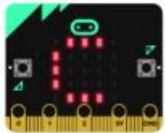


Indique le démarrage de l'application



Indique que le système cherche à se connecter au point d'accès demandé. Si cet icône reste toujours visible, vous avez peut-être fait une erreur dans les connexions entre les deux cartes, ou que le SSID, n'est pas correct, ou le mot de passe.

Puis visualiser au final, une fois connectée, un « C » comme connecté.



Lancez l'interrupteur connecté. Alimentez la deuxième carte Microbit.

Appuyez sur le bouton A, vous devriez voir la lampe connectée s'allumer.

Appuyez sur le bouton B, vous devriez voir la lampe connectée s'éteindre.

9. Tests

Vous pouvez aussi utiliser une tablette ou smartphone pour simuler un client ou un serveur sur un réseau en utilisant deux application Android très pratiques :

Application TCP client pour jouer le rôle de l'interrupteur connecté par exemple.



<https://play.google.com/store/apps/details?id=com.sollae.eztcpclient&hl=fr>

Application TCP serveur pour jouer le rôle de la lampe connectée



<https://play.google.com/store/apps/details?id=com.mightyIT.gops.tcpserver&hl=fr&gl=US>

10. Pédagogie

En général, pour des questions de sécurité, il n'est pas possible d'utiliser le réseau pédagogique de votre salle de technologie.

Avec le système proposé, il vous sera alors possible de créer votre propre réseau dans votre classe.

Il sera alors possible d'utiliser un simple point d'accès



Puis d'utiliser un système par îlot, et de demander à deux îlots de jouer, l'un le rôle de client, et l'autre de serveur.

Il faudra faire réfléchir les élèves sur les adresses IP à choisir dans leur îlot. Par exemple, chaque îlot pourra proposer son adresse IP en imposant une adresse réseau et en laissant le choix de l'adresse périphérique. Chaque îlot pourra alors noter son adresse sur le tableau pour éviter les doublons.

Avant d'utiliser les systèmes Microbit, des activités peuvent être menées avec de simples smartphones ou tablettes. Par exemple réaliser un petit réseau social avec un petit tchat.

Il est aussi possible d'utiliser un système IOT complet et fonctionnel pour faire réfléchir les élèves sur les adresses IP par exemple et visualiser les adresses IP utilisées sur le réseau.

Les élèves pourront choisir leur protocole dans un premier temps, puis visualiser le code reçu sur la partie lampe connectée et valider ainsi leur protocole de communication.

Une commande bloc permet aussi d'afficher l'adresse IP du client connecté au serveur.

On pourra fournir un programme complet où on demandera aux élèves de modifier par exemple : l'adresse IP, le protocole, l'affichage de l'IP du client, modifier les actions (par exemple pilote le robot Codo), etc..

Vos élèves pourront suite créer d'autres objets connectés, comme le contrôle à distance de la température de son réfrigérateur.

Vous avez à votre disposition d'autres commandes de blocs dans l'extension Makecode afin que le serveur réponde au client, et que le client lise la réponse du serveur.

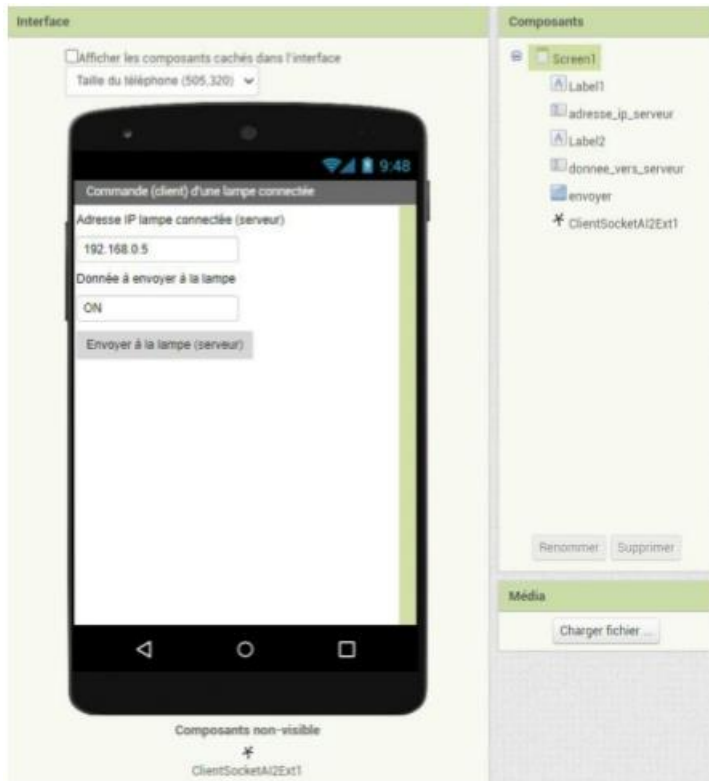
Dans le cas où vous voulez récupérer des données d'un capteur connecté, il faudra que votre capteur soit positionné en mode serveur, et qu'un autre périphérique vienne l'interroger en mode client. Typiquement, vous pouvez faire réaliser une application cliente avec Appinventor et faire réaliser une station météo avec la carte microbit en mode serveur.

11. Commander la lampe avec Appinventor

Il est assez simple de créer maintenant une application Android qui va commander cette même lampe.

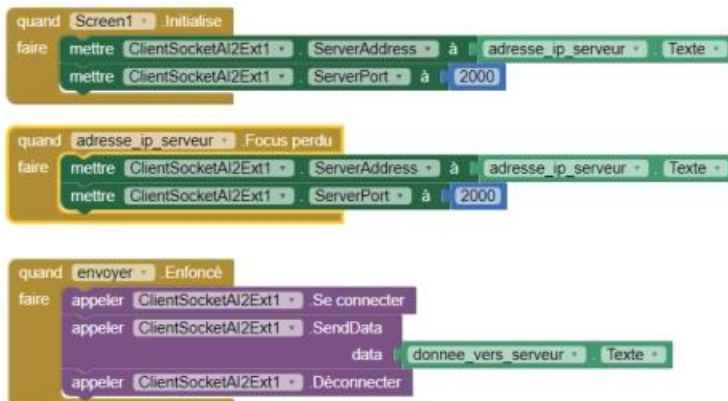
Il suffit d'utiliser une extension client TCP/IP pour Appinventor qui permet de se connecter à un serveur et d'envoyer une donnée.

Partie IHM



Partie programmation.

Vous remarquerez que l'on se connecte au serveur, on envoie la donnée au serveur, puis on ferme la connexion avec le serveur.



12. Les fichiers sources sont disponibles ici :

<http://technologiecollege.atwebpages.com/index.php/11-domaines/information/les-reseaux/2-wifi-avec-la-carte-microbit>