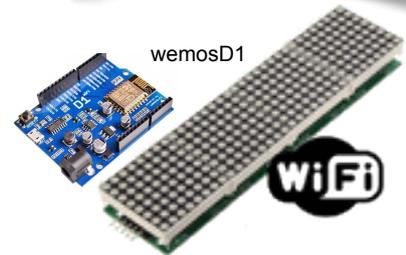


Poster un message sur un afficheur dynamique à DELS.



Ce tutoriel montre comment concevoir une application qui permet de poster un message vers l'afficheur dynamique à DELS. Il décrit aussi le programme ARDUBLOCK à téléverser dans la carte wemos D1. Pour terminer, un plan de câblage du système est décliné en deux versions.

Principe de fonctionnement :  <https://youtu.be/jvIHh38bpa4>



Les deux terminaux (smartphone, wemosD1+afficheur) sont sur le même réseau local. L'application envoie une requête HTTP qui inclut le texte vers l'afficheur

- l'adresse IP doit pouvoir être renseignée par l'utilisateur ; - message est le nom de la variable qui stockera le texte saisi par l'utilisateur ; - le texte est une chaîne de caractères.

Design de l'APP.

1

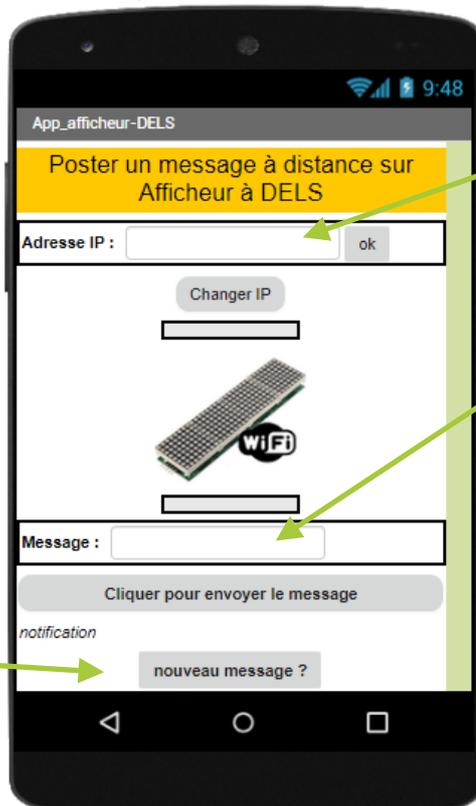
A l'aide de la bibliothèque des composants, réalises l'interface qui suit:



La notification et bouton nouveau message ne sont pas visibles au démarrage de l'application.

Il apparaissent lorsque que l'utilisateur clique pour envoyer un premier message.

Rappel : le composant Web1 est disponible dans la bibliothèque Connectivité



Acquisition_IP

Acquisition_TEXTE



Fichier AIA téléchargeable depuis la rubrique APP INVENTOR du site (les blocs sont à programmer) :

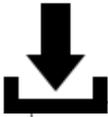
<http://moncoursdetechno.ovh/appinventor.html>

Composants non-visible

Web1 Horloge1

2

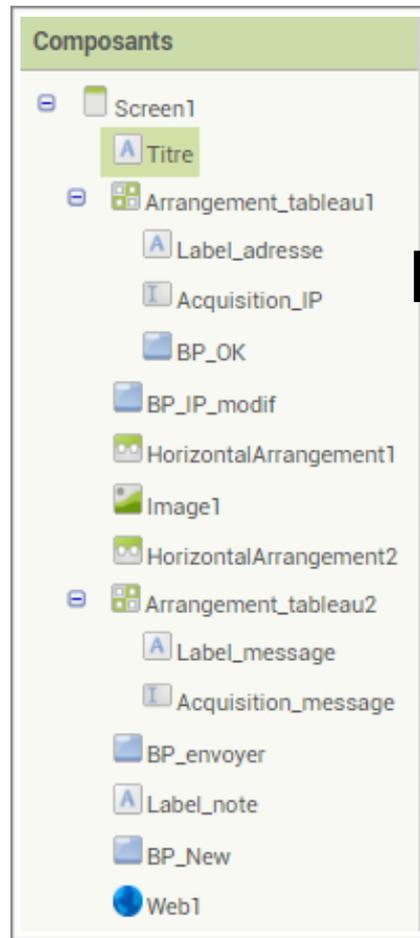
Nommer les composants



L'image est téléchargeable depuis la rubrique* APP INVENTOR du site :

<http://moncoursdetechno.ovh/appinventor.html>

**Dans cette rubrique, rendez-vous dans la partie Communication AppInventor/D1 pour la trouver.*



Programmation de L'APP.

3

Créer une variable pour stocker l'adresse IP et une seconde pour stocker le message

initialise global adresseIP à " "

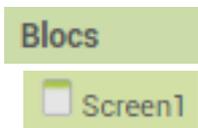
initialise global message à " "



Pour initialiser ces deux variables au démarrage de l'APP, on associe une case vide de texte.

4

Programmer le composant écran



quand Screen1 .Initialise
faire
mettre Label_note . Visible à faux
mettre BP_New . Visible à faux



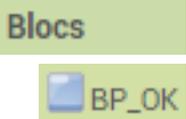
On met le paramètre visible à Faux pour que les composants Label_note et BP_New soit cachés au démarrage de l'APP.

5

Programmer le composant BP_OK



Lorsque que l'utilisateur appuie sur ce bouton alors l'adresse IP qu'il a saisi dans le formulaire est affiché en vert et le bouton pour changer l'adresseIP apparaît.



quand BP_OK .Clic
faire
mettre Acquisition_IP . Couleur texte à [vert]
mettre BP_IP_modif . Visible à vrai

5

Programmer le composant BP_IP_modif



Lorsque que l'utilisateur presse ce bouton alors le formulaire est vidé de l'adresse IP pour être remplacé le texte par défaut (valeur de départ indiquée dans le design)

Blocs

BP_IP_modif

quand BP_IP_modif .Clic

faire
mettre Acquisition_IP . Couleur texte à [noir]
mettre Acquisition_IP . Texte à Obtenir valeur de départ

Nuance

Saisir IP de l'afficheur

6

Programmer le composant BP_envoyer



Quand l'utilisateur clique sur ce bouton alors la variable adresseIP stocke la valeur de l'adresse IP saisie par l'utilisateur et la variable message stocke le texte saisi par l'utilisateur.

Blocs

BP_envoyer

quand BP_envoyer .Clic

faire
mettre global adresseIP à Acquisition_IP . Texte
mettre global message à Acquisition_message . Texte
mettre Web1 . Url à joint " http://" obtenir global adresseIP
"/?message=" obtenir global message
appeler Web1 . Obtenir
mettre Label_note . Visible à vrai
mettre Label_note . Couleur texte à [jaune]
mettre Label_note . Texte à joint " message envoyé à " obtenir global adresseIP
obtenir global message
mettre Acquisition_message . Texte à obtenir global message
mettre BP_New . Visible à vrai



La requête HTTP (Url) est concaténée pour être envoyée via le Web.



Une notification apparaît en jaune en indiquant la requête envoyée.



Le bouton pour effacer et saisir un nouveau message apparaît.

7

Programmer le composant BP_New

Blocs

BP_New



La notification est cachée. La zone de saisie du message est actualisée avec le texte par défaut.

quand BP_New .Clic

faire
mettre Label_note . Visible à faux
mettre Acquisition_message . Texte à Obtenir valeur de départ
mettre BP_New . Visible à faux

Nuance

Tapez ici votre message

Programmation de la carte D1.

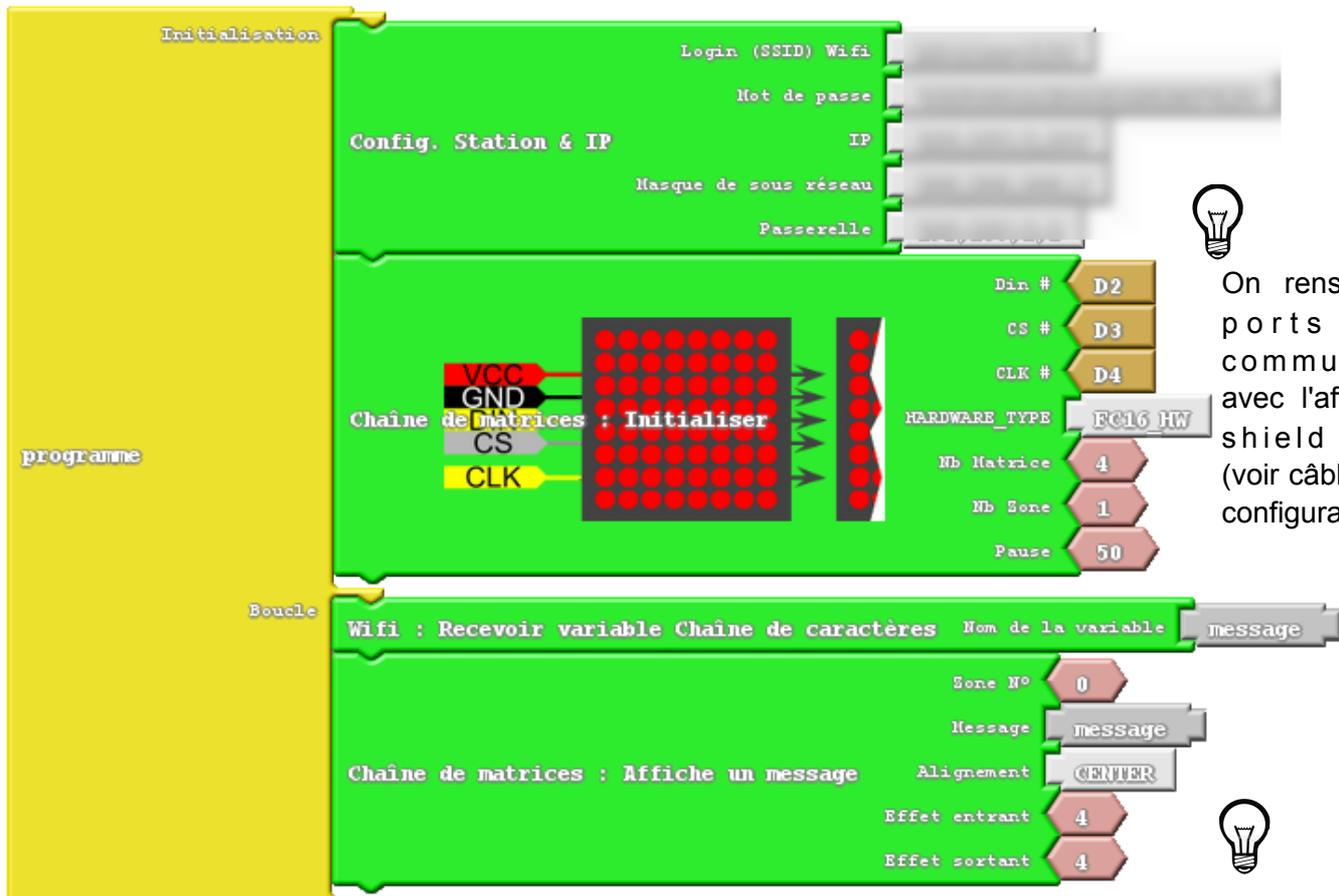
8

Rédiger le programme avec Arduino Augmenté

<http://duinoedu.com/telecharger.html>



La carte d1 est configurée comme Station de notre réseau local.



On renseigne les ports de communication avec l'afficheur du shield GROVE (voir câblage) et sa configuration.

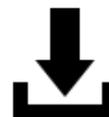


On reçoit la variable message postée depuis l'APP et on l'affiche avec un effet de défilement de la droite vers la gauche..

Câblage de l'afficheur dynamique.

Un support pour l'afficheur a été réalisé avec une CNC. Les fichiers de fabrication sont en libre téléchargement :

http://moncoursdetechno.ovh/coin_maker/maquettes.html



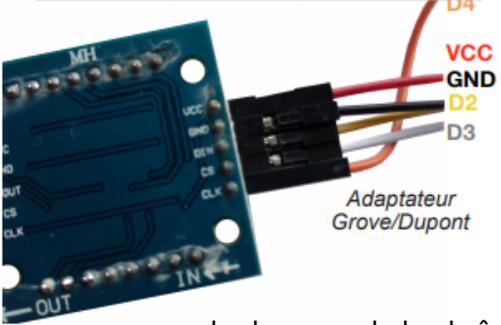
Deux solutions pour câbler la chaîne de matrice à DELS sur la Wemos D1:

- une version avec une connectique de type dupont plus délicate à faire avec des élèves ;
- une autre version avec une connectique de type GROVE, facilite le câblage .

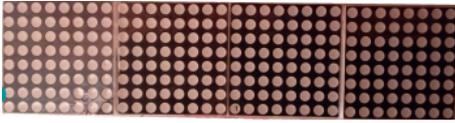
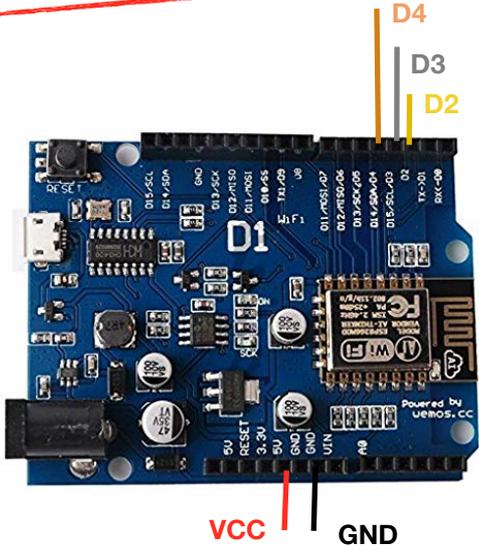


Câblage "dupont"

Prise	Chaine-matrices.	
Prise D2	Noir	GND -
	Rouge	VCC +
	Blanc	CS
Prise D4	Jaune	DIN
	Marron	CLK

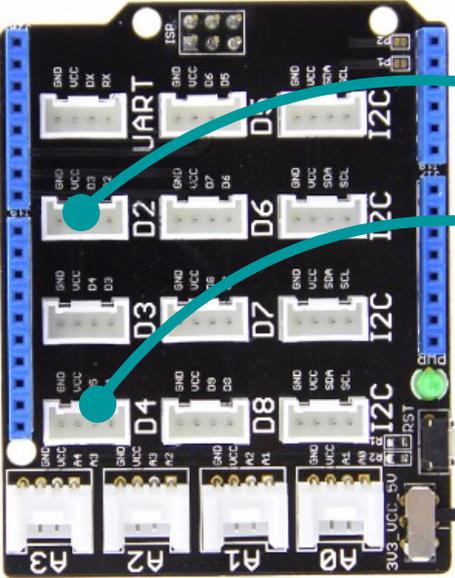


Adaptateur Grove/Dupont

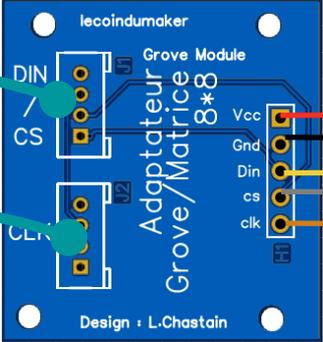


vue de dessous de la chaîne de matrice à del

Câblage "grove"



Shield Grove assemblé sur carte WEMOSD1



Adaptateur à fabriquer : http://moncoursdetechno.ovh/coin_maker/modules_electroniques.html

