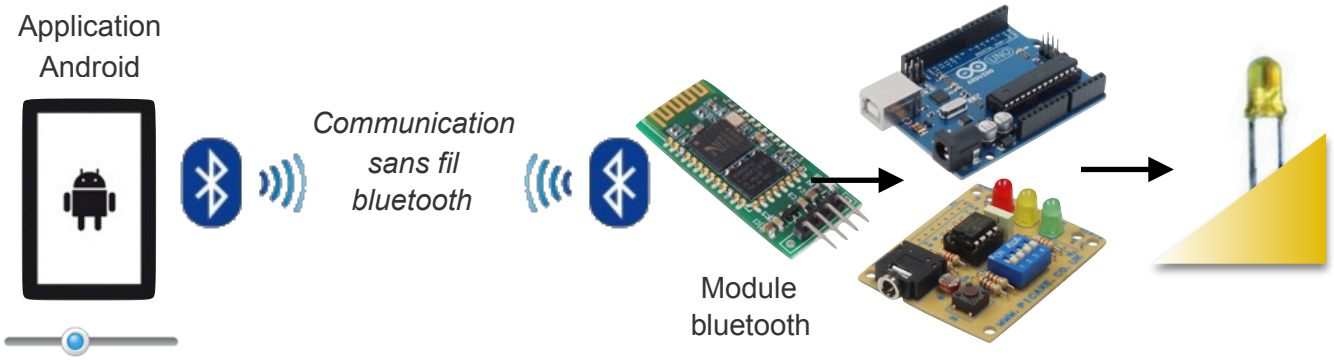
App Inventor : Communication via bluetooth

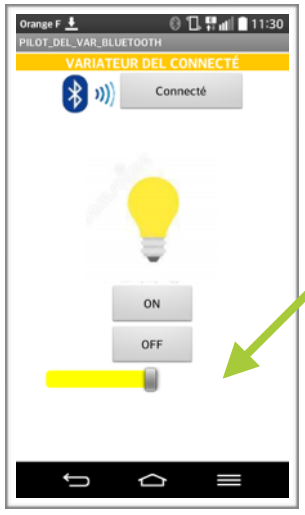
App Inventor



Dans cet exemple il s'agit, de piloter la puissance d'éclairage une del (variation de lumière) depuis le smartphone (application Android)

Pour cela nous allons utiliser les sorties « analogiques » (PWM) des microcontroleurs (Picaxe ou Arduino) pour faire varier la puissance lumineuse de la del. Voir autre ressource pour davantage de précisions.

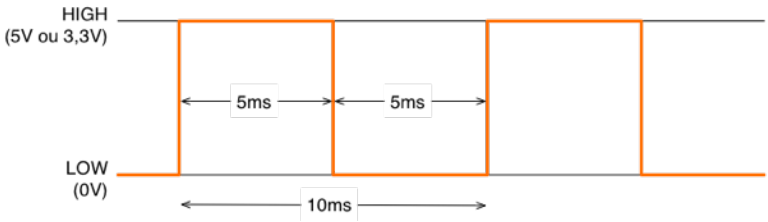
Côté application, nous allons utiliser un curseur qui permet de communiquer une valeur entre 0 et 255 (soit les 256 possibilités en 8 bits) en bluetooth.



Curseur de 0 à 255
et initialement à 128
(valeur milieu)

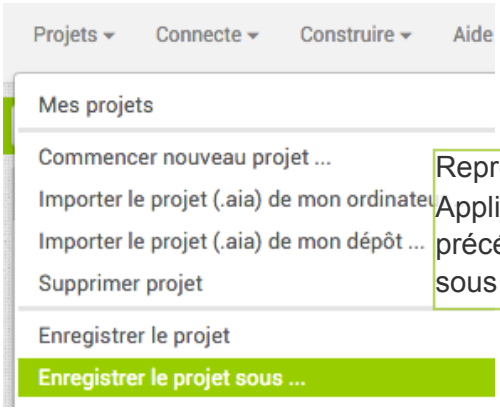
Exemple avec PMW à 50%

La fréquence est de 100Hz, le rapport cyclique de 50%



Une sortie PWM sur un microcontroleur est une sortie Numérique dont les signaux ont toujours une valeur LOW (0 logique) ou HIGH (1 logique). Mais le principe est de construire un signal qui est alternativement LOW et HIGH et de répéter très vite cette alternance en faisant varier la fréquence du signal.

*Dans le cas d'une **DEL**, elle est alternativement allumée et éteinte mais le cycle est tellement rapide que la persistance rétinienne nous donne l'illusion d'une DEL allumée en permanence. Prenons par exemple une période de 10ms, soit une fréquence de 100Hz. Si la DEL est allumée pendant 5ms et éteinte pendant 5ms, comme sur la figure ci-contre, l'impression sera une **luminosité** de 50% de la luminosité maximum.*



1

Reprendre l'application Appli_Lampe_Bluetooth vue précédemment et l'enregistrer sous un nouveau nom de projet

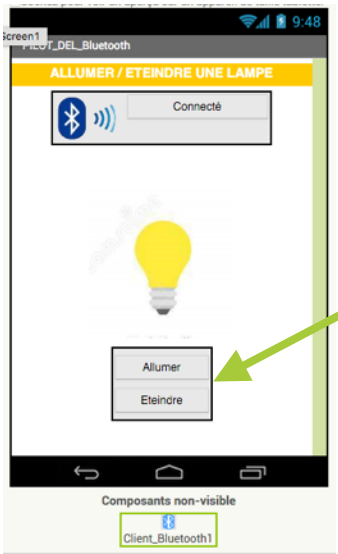
Ajouter les images que l'on va utiliser par la suite :

Image variation et nouveau logo de l'application

2

3

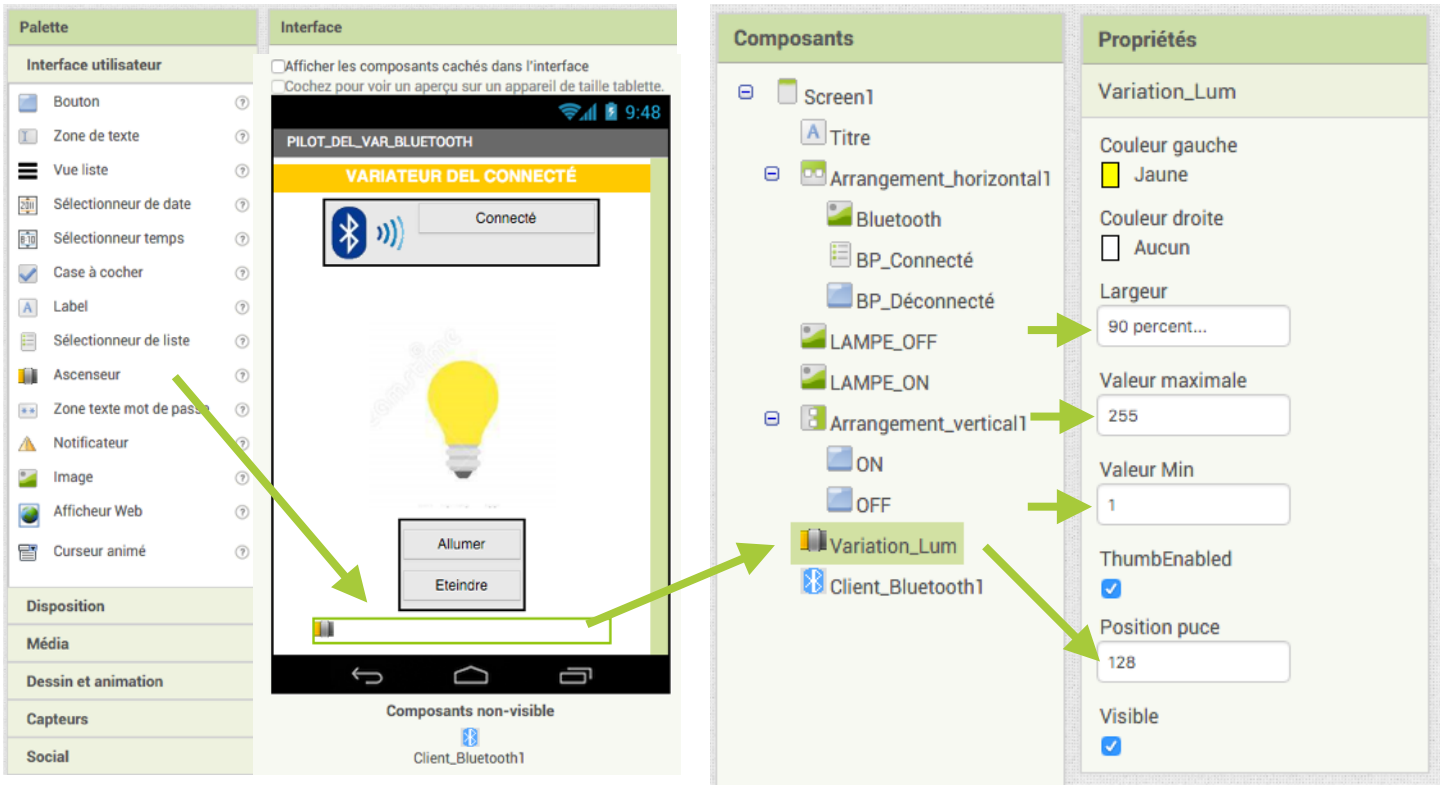
Changez les propriétés de l'application : Logo et nomme l'application



4

Côté interface design de l'application, il suffit d'ajouter un curseur (renommé « Variation_Lum »: Ascenseur en dessous des 2 boutons « Allumer » « Eteindre ».

L'ascenseur doit avoir pour valeur max et min respectivement 255 et 1 (car nous sommes en 8 bits, voir tableau en bas de page). Egalement il peut être initialisé en position milieu (Position puce) soit à 128.

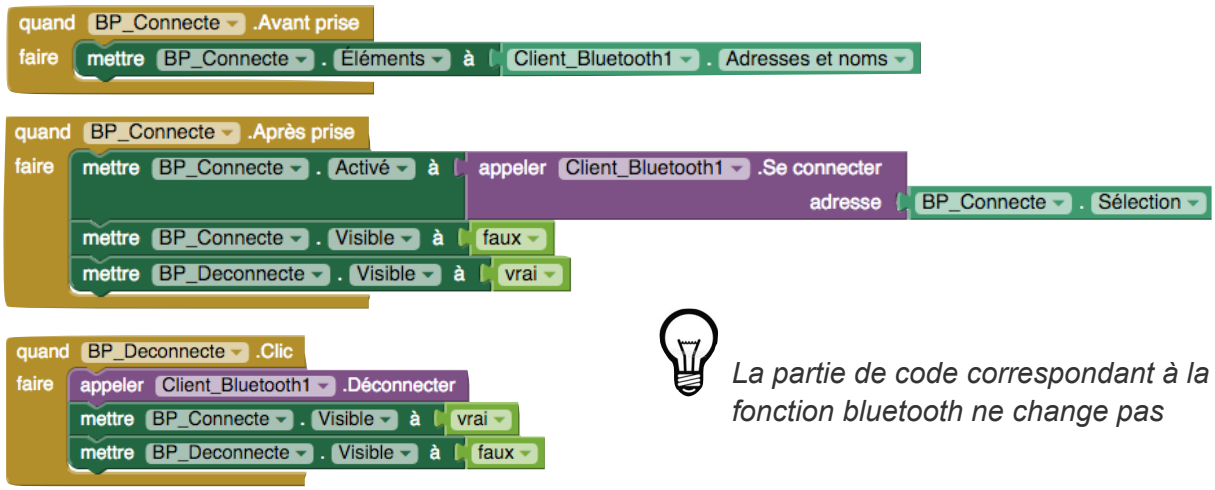


Rapport Cyclique %	0	5	10	15	20	25	30	35	40	45	50	55	60	65	70	75	80	85	90	95	100
Valeur sur 8 bits	0	13	26	38	51	64	77	89	102	115	128	140	153	166	179	191	204	217	230	242	255

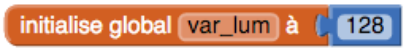
Designer | Blocs

5

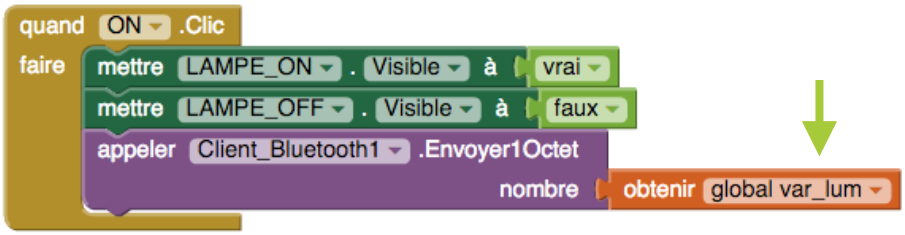
Côté programmation … Quelques ajouts et modifications sont à réaliser pour communiquer les valeurs de l'ascenseur via le bluetooth



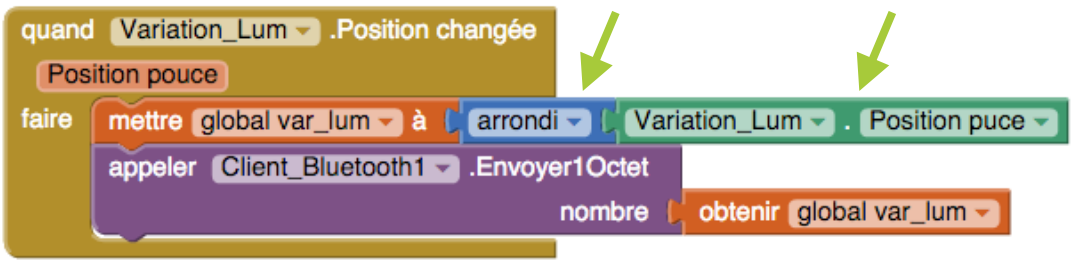
La partie de code correspondant à la fonction bluetooth ne change pas



*Initialiser une variable **var_lum à 128** (valeur milieu du curseur par défaut)*

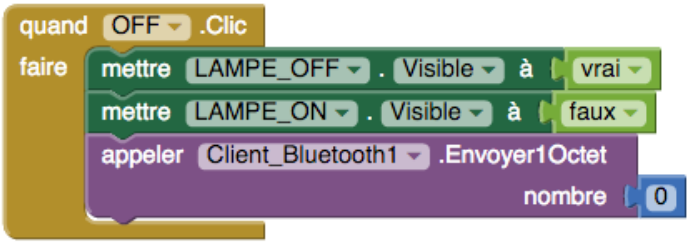


*Le bouton « ON » envoi donc maintenant le contenu de la variable **var_lum***



Quand le curseur de l'ascenseur change de position :

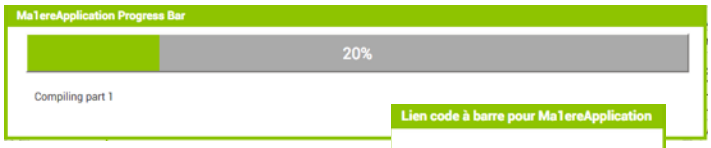
Le Client_Bluetooth envoi la position actuelle du curseur :
*Soit une valeur **entière** entre 1 et 255, d'ou l'utilisation du bloc **arrondi***

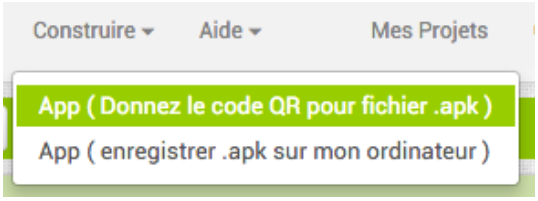


Côté bouton « OFF » rien ne change : il communique 0 via le client bluetooth

6

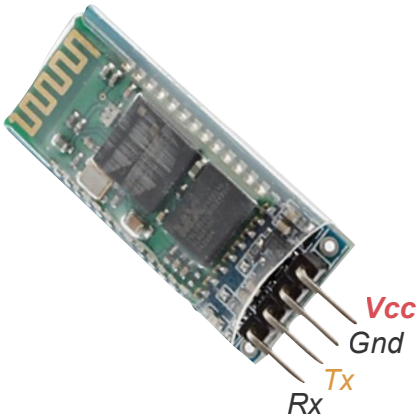
L'application est terminée, vous pouvez la tester et l'installer sur la tablette ou smartphone Android





Il reste maintenant à réaliser un montage électronique qui permet de recevoir en bluetooth les valeurs de 1 à 255 générés par l'application.

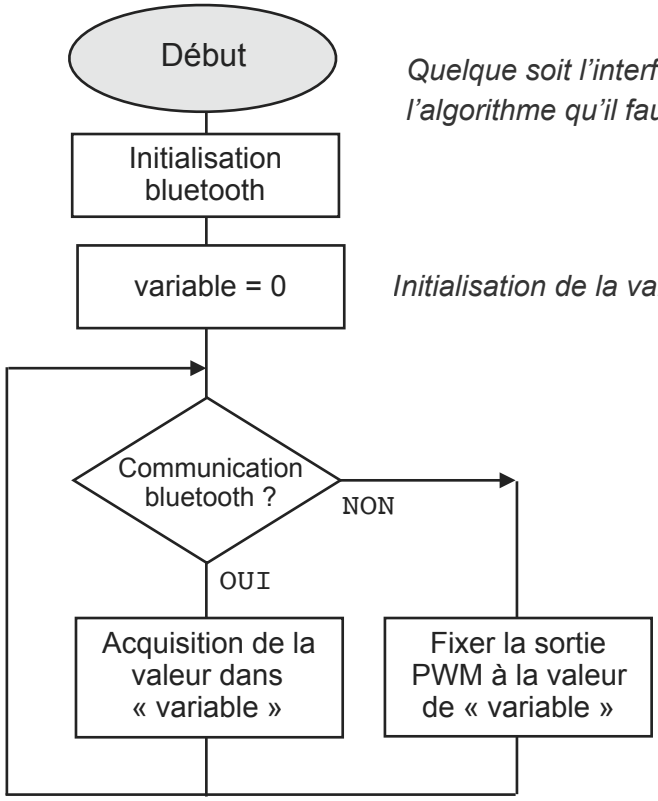
La solution la plus simple étant d'utiliser une interface programmable Picaxe ou Arduino et leurs sorties PWM associées



***Vcc** pour l'alimentation 3,3V ou 5V*

***Gnd** pour l'alimentation : 0V*

***Tx** et **Rx** pour la communication*



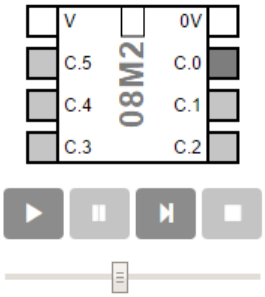
Quelque soit l'interface choisie, voici l'algorithme qu'il faut programmer

Initialisation de la variable à 0

Exemple avec un microcontroleur Picaxe sous Blockly

Projets ▾ Édition ▾ Paramètres ▾ PICAXE ▾ Mode: Blocks Code

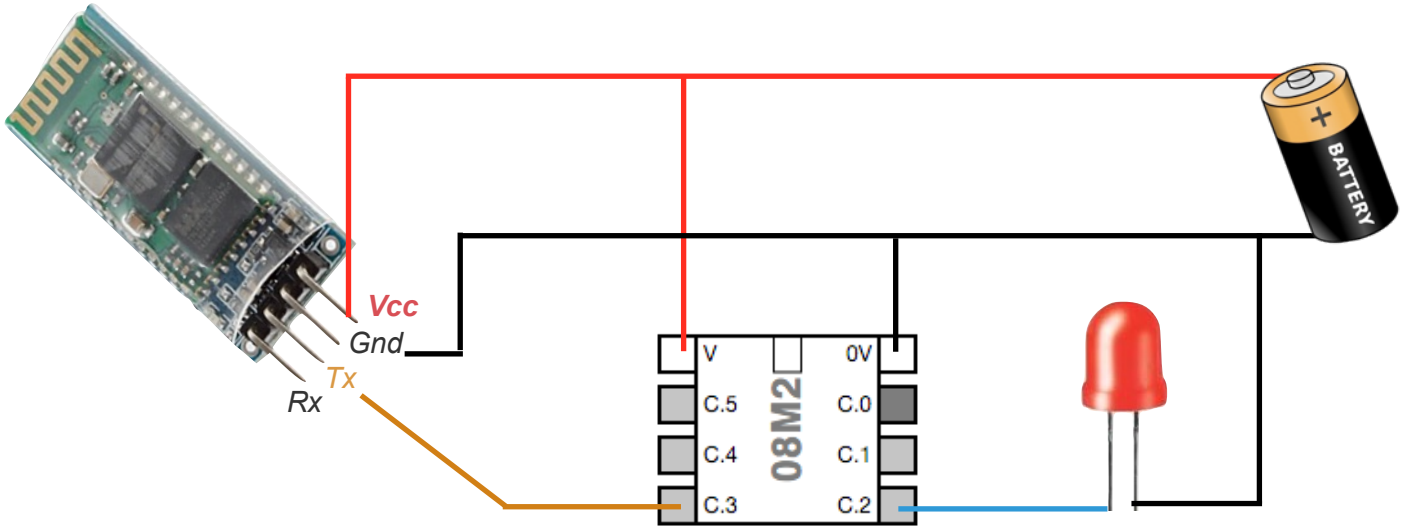
Simulateur | Blocks | PICAXE BASIC | Javascript | XML



- Sorties
- Entrées
- Délais
- Boucles
- Variables
- Maths
- Procédures
- Tâches
- Moteurs
- Liaison série
- Avancé

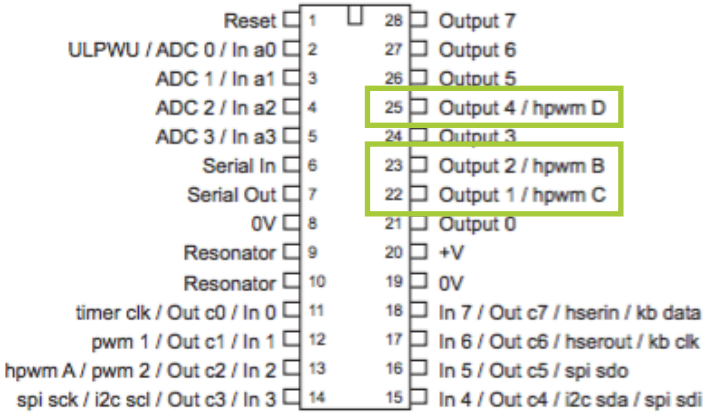
```
début
BASIC setfreq m8
répéter indéfiniment
faire
    fixer varA à 0
    BASIC serin C.3, T9600_8, varA
    si varA <> 0
    faire
        signal pwm de periode 255 rapport cyclique varA sur C.2
    si varA = 0
    faire
        signal pwm de periode 255 rapport cyclique 0 sur C.2
```

Rapport Cyclique %	0	5	10	15	20	25	30	35	40	45	50	55	60	65	70	75	80	85	90	95	100
Valeur sur 8 bits	0	13	26	38	51	64	77	89	102	115	128	140	153	166	179	191	204	217	230	242	255

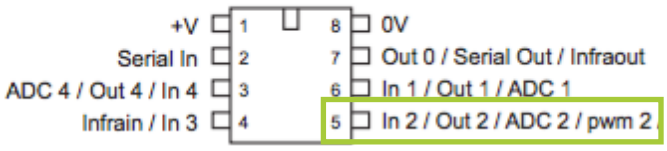


Ne pas oublier une résistance pour limiter le courant dans la del : 300 à 500 Ohms

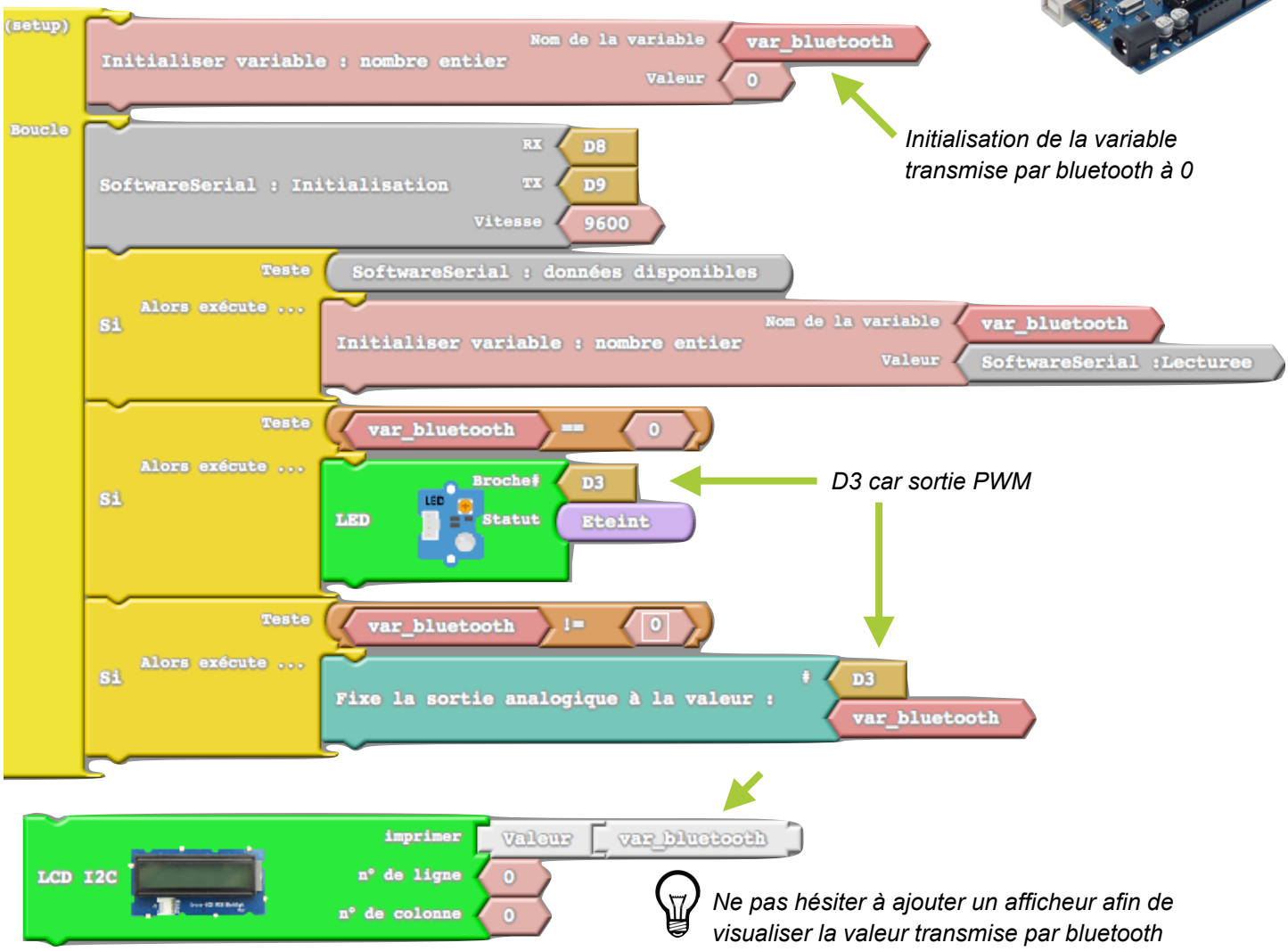
PICAXE-28X1



PICAXE-08M PICAXE-08M2



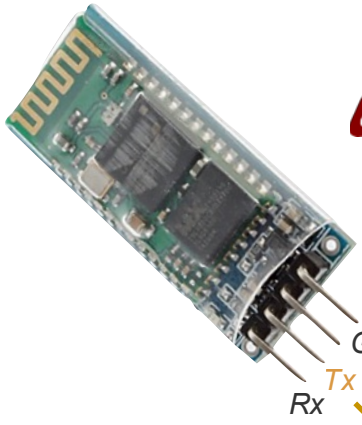
Exemple avec une interface Arduino sous Ardublock



Initialisation de la variable transmise par bluetooth à 0

D3 car sortie PWM

Ne pas hésiter à ajouter un afficheur afin de visualiser la valeur transmise par bluetooth



Attention ici on utile le **port D3** car seuls les **ports 3, 5, 6, 9, 10 et 11** peuvent fournir une sortie analogique (PWM).
Ils sont repérés par le symbole : ~

Ne pas oublier une résistance pour limiter le courant dans la del :
300 à 500 Ohms