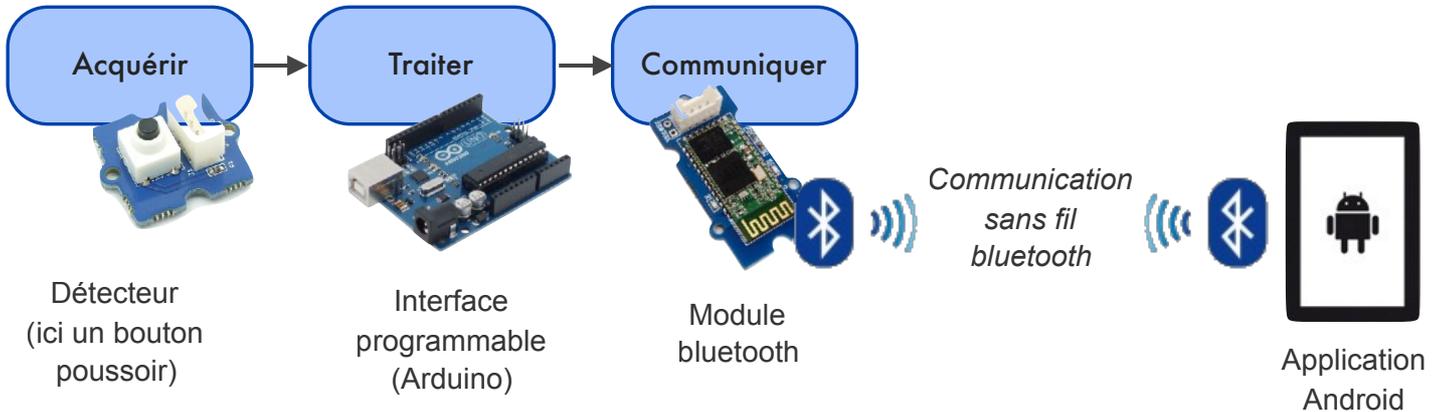
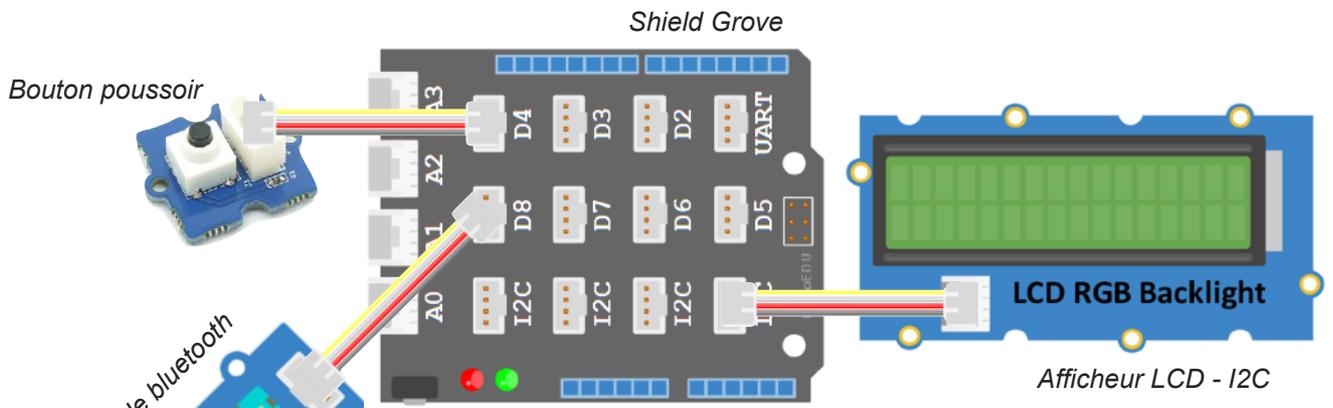


APP INVENTOR



Dans cet exemple il s'agit, d'afficher l'état d'un détecteur sur l'écran d'un smartphone (application Android)



La carte Arduino traite le signal envoyé par le bouton (1 ou 0), puis le communique au module bluetooth (et l'affiche aussi sur un afficheur LCD).

Une application nomade réceptionne le signal bluetooth et affiche la valeur numérique du bouton poussoir.

Voici le programme sous mBlock (avec la librairie spécifique Uno et Grove développée par TechnologieService)

1

UNO et Grove - générer le code

mettre Acquisition_Bouton à 0

Effacer le texte

Afficher le texte Etat bouton sur la ligne 0

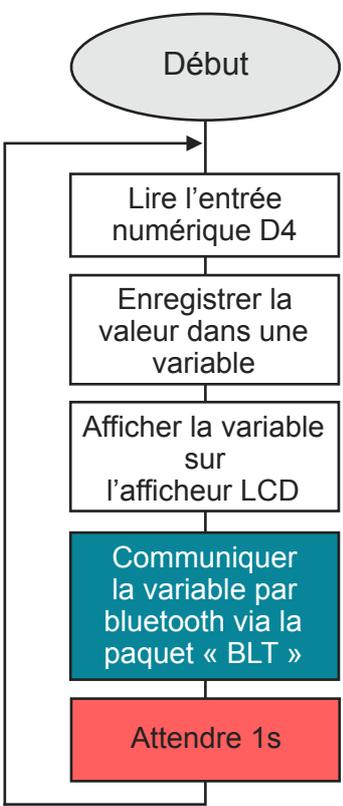
répéter indéfiniment

mettre Acquisition_Bouton à Lire l'état logique du bouton poussoir sur la broche D4

Afficher le texte Acquisition_Bouton sur la ligne 1

BT: envoyer BLT de valeur Acquisition_Bouton sur le port D8

attendre 1 secondes



Création d'une variable « Acquisition_Bouton » qui contient la l'état du bouton poussoir connecté sur l'entrée numérique D4.
Affichage de cette variable sur l'afficheur LCD.

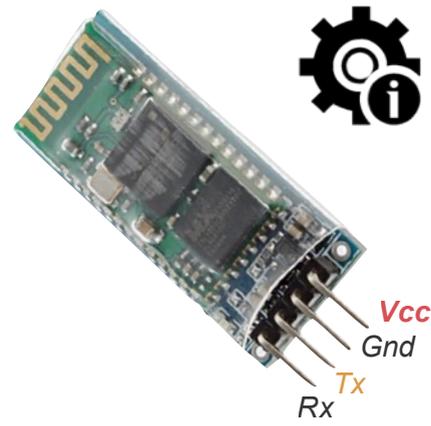
Puis transmission du contenu de la variable par le paquet ici nommé « BLT » via le module bluetooth.

Avant de boucler une pause de 1 seconde est imposée.

Une fois le programme fonctionnel au niveau de la carte Arduino, il faut réaliser l'application Android qui va réceptionner la variable :

« Acquisition_Bouton »
via le paquet
« BLT »

2



Pour rappel : mBlock utilise une vitesse de 115200 Bauds

	Prise	Module bluetooth
Port D8	Noir	Gnd -
	Rouge	Vcc +
	Blanc	Rx
	Jaune	Tx

Une simple zone de texte (Label) est suffisante pour venir afficher la valeur de la variable reçu par bluetooth.

4

Interface utilisateur

- Bouton
- Case à cocher
- Sélecteur de date
- Image
- Label



Texte

En attente de valeur

Afin que la zone de texte ne soit pas vide, il est possible d'indiquer un texte de ce style dans les propriétés du composant.

La partie connectivité en bluetooth, est identique aux tutoriaux précédents.

3

Connectivité

- Déclencheuractivité
- Client Bluetooth
- Serveur Bluetooth
- Web

Capteurs

- Accéléromètre
- Scanneur code à barre
- Horloge
- GyroscopeSensor
- Capteur position
- Champ proche
- Capteur orientation
- Pedometer
- ProximitySensor

Composants non-visible

- Client Bluetooth
- Horloge1

Il est important d'ajouter une Horloge pour se synchroniser avec le module bluetooth de la carte Arduino. En indiquant le même intervalle de temps soit 1 seconde !

5

Propriétés

Horloge1

- TimerAlwaysFires
- ChronomètreActivé
- IntervalleChronomètre: 1000

1s = 1000ms

Côté programmation ... Quelques ajouts et modifications sont à réaliser pour réceptionner la communication via le bluetooth

6

```

quand BP_Connecte .Avant prise
faire
mettre BP_Connecte . Eléments à Client_Bluetooth1 . Adresses et noms

```



Quand BP_Connecte (le sélectionneur_de_liste) cliqué :
Afficher le nom des connexions disponibles en bluetooth

```

quand BP_Connecte .Après prise
faire
mettre BP_Connecte . Activé à appeler Client_Bluetooth1 .Se connecter
adresse BP_Connecte . Sélection
mettre BP_Connecte . Visible à faux
mettre BP_Deconnecte . Visible à vrai

```



Quand BP_Connecte (le sélectionneur_de_liste) a été sélectionné :
Se connecter au bluetooth sélectionné
Ne plus afficher le bouton qui permet d'afficher cette liste de bluetooth : BP_Connecte
Afficher le bouton « se déconnecter » : BP_Deconnecte

```

quand BP_Deconnecte .Clic
faire
appeler Client_Bluetooth1 .Déconnecter
mettre BP_Connecte . Visible à vrai
mettre BP_Deconnecte . Visible à faux

```



Quand BP_Deconnecte cliqué :
Se déconnecter du bluetooth
Afficher le bouton qui permet d'afficher cette liste des bluetooth disponibles : BP_Connecte
Ne plus afficher le bouton « se déconnecter » : BP_Deconnecte

```

quand Horloge1 .Chronomètre
faire
si Client_Bluetooth1 . Est connecté et Client_Bluetooth1 . Disponible
alors
mettre Affichage_Valeur . Texte à appeler Client_Bluetooth1 .Recevoir texte
nombre d'octets appeler Client_Bluetooth1 .Octets disponibles pour la réception

```



A chaque intervalles de temps de l'horloge :
Si le bluetooth est activé et disponible :
Alors afficher la donnée reçue par bluetooth dans le Label « Affichage_valeur »

Mais l'application affiche le contenu du paquet reçu, c'est à dire : « BLT:0.00 »

Il est possible d'améliorer l'affichage en enregistrant le paquet réceptionnée par bluetooth dans une variable ici « Variable_Reception » puis de la comparer aux 2 valeurs réceptionnées.

7



```
initialise global Variable_Reception à " "
```

```

quand Horloge1 .Chronomètre
faire
  si Client_Bluetooth1 . Est connecté et Client_Bluetooth1 . Disponible
  alors
    mettre global Variable_Reception à appeler Client_Bluetooth1 .Recevoir texte
    nombre d'octets appeler Client_Blu
  si comparer textes obtenir global Variable_Reception = " BLT:1.00 "
  alors
    mettre Affichage_Valeur . Couleur de fond à [vert]
    mettre Affichage_Valeur . Texte à " 1 "
  si comparer textes obtenir global Variable_Reception = " BLT:0.00 "
  alors
    mettre Affichage_Valeur . Couleur de fond à [rouge]
    mettre Affichage_Valeur . Texte à " 0 "

```



Dans cet exemple, la couleur de l'affichage change également en fonction de la donnée reçue (vert pour 1 et rouge pour 0).

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

8

