

Doc TestXamConnection

Le projet TestXamConnections propose une couche d'abstraction pour se connecter à des appareils bluetooth depuis un appareil Android. La librairie a été développée pour fonctionner avec les appareils suivants :

- Balance Teo de la marque Balea
- Appareil Android C-One² proposant le scan de puce RFID grâce à agrident Wedge
- Lecteur barcode HoneyWell
- *Barcode interne ?*
- *Outil externe agrident ?*

Pour ces appareils, il existe dans la librairie des classes dédiées qui proposent les fonctions pour les utiliser.

Pour la balance Teo :

Il s'agit de la classe TeoBalance qui s'initialise en prenant en argument un objet de type Device (2 champs: Nom et Adresse MAC). Cette classe propose les fonctions suivantes :

- **Task<bool> ConnectToTeoAsync()** : permet d'établir une connexion bluetooth entre l'appareil et la balance Teo. Important : le bluetooth doit être activé, la balance allumée et les appareils doivent déjà être appairés. Renvoie true si la connexion a réussi (i.e. un socket RFCOMM pour la balance Teo a été ouvert), false sinon.
- **Task<bool> SendCommandToTeoAsync(TeoCommandType)** : permet d'envoyer une commande à la balance pour le programme INRAE. L'ensemble des commandes est disponible dans l'enum TeoCommandType. Retourne true si la commande a été envoyée, false sinon.

La classe présente également des événements auxquels il est possible de s'abonner :

- **TeoPeseReceivedEvent<ReponsePesee>** : déclenché à la réception d'une pesée, disponible au format ReponsePesee.
- **TeoDataReceivedEvent<string>** : déclenché à la réception de données autre qu'une pesée. Ces données sont disponibles au format string

L'objet ReponsePesee :

- *int NumPlateau* : numéro du plateau
- *int TypePlateau* : type du plateau
- *string PoidsMesure* : chaîne de caractère contenant la valeur de la pesée
- *bool IsNegative* : indique si le poids est faible ou non (true : poids négatif)

Pour le lecteur de RFID Agrident présent dans le C-One :

La classe AgridentReader propose les événements suivants :

- **AgridentDataReceivedEvent<string>** : Événement déclenché à la réception de données par le lecteur RFID.
- **AgridentNORFIDEvent** : Événement déclenché lorsque le scan de tag RFID a échoué.

Les fonctions suivantes :

- **Task<bool> EnableServiceAsync()** : Initialise la lecture de tag RFID.
Retourne true la connexion a réussi, false sinon.
- **Task<bool> StartScanAsync()** : Lance un scan de tag RFID.
Retourne true si le scan s'est lancé correctement, false sinon.

La structure de la librairie est visible sur le diagramme de classe suivant :

Voir SicpaDeviceConnectionsUML.drawio.svg (dans le rep. Git, à ouvrir dans un navigateur Internet)

Il est aussi possible d'utiliser les composants de la librairie pour se connecter à un nouvel appareil.

Le fonctionnement repose sur le **ConnectionServiceProvider** qui fournit un **ConnectionService** en fonction du type de connexion demandé.

Ces classes proposent toutes les même fonctionnalité :

- La connexion avec la fonction `Connect(params)`. Le passage des arguments se fait à l'aide d'un `Dictionary<string,string>`:

- En interne, il s'agit de l'enregistrement des intents qui vont être reçus en les passant en arguments. La fonction attend un champ "receivable_intents" (ConnectionConstants.RECEIVABLE_INTENTS_KEY) avec une liste d'intent séparées par une virgule (csv)
- En bluetooth il s'agit de l'ouverture d'un socket RFCOMM en fonction de l'adresse MAC passée en argument. La fonction attend un champ "mac_address" avec l'adresse MAC passé en string ("XX:XX:XX...").

Dans le cas où les paramètres sont erronés, la fonction Connect renvoie false.

- L'envoi d'une commande :
 - En interne par une chaîne de caractère.
 - En bluetooth avec un tableau d'octets.
- Un événement DataReceivedEvent<byte[]>, déclenché à la réception de données. Présente les données sous forme d'un tableau d'octets.

Cas du lecteur de code barre HoneyWell :

Il suffit de créer une page BluetoothReaderView (voir projet XamConnections) qui utilise seulement la fonction de connexion ainsi que l'évènement BluetoothDataReceived du BluetoothConnectionService.