

# Documentation de la suite logicielle GBMProject3A

Guillaume Gibert

9 mars 2020

## 1 Introduction

Ce document décrit les différents éléments de la suite logicielle GBMProject3A.

### 1.1 Côté PC

Le code fourni est écrit en C++ avec comme unique dépendance l'API Qt. Il se compose d'un ensemble de classes qui permettent :

- de communiquer avec un système Arduino que ce soit en réception ou en émission ;
- de traiter des signaux (filtrage, FFT) ;
- d'afficher des signaux ;
- de faire des requêtes sur une base de données MySQL.

### 1.2 Côté Arduino

Un code de test est fourni qui permet d'envoyer sous forme de chaîne de caractères standardisée les valeurs lues sur les 5 canaux analogiques de la carte Arduino. Une fonction de lecture d'événements est également mise en place pour lire, en parallèle, les données sur le port série provenant du PC.

Une série d'exemples permet d'implémenter chaque fonctionnalité. Ils vont être décrits dans la suite du document.

## 2 Installation

### 2.1 Qt

Pour utiliser le code fourni dans ce projet, il faut installer l'API Qt. Pour ce faire, il faut :

- aller sur le site de Qt : <https://www.qt.io/>

- télécharger l’installateur pour la version Open Source (Onglet ”Download. Try .Buy”, puis ”Go for Open Source”, enfin ”Download the Qt Online Installer”);
- lancer l’installateur (vous devrez créer un compte Qt si vous n’en possédez pas) et choisir la version Open Source;
- choisir une version de Qt pour un compilateur donné (par exemple MinGW 64-bit)
- finaliser l’installation en suivant les instructions.

Une fois l’installation terminée, vous trouverez l’outil `qtcreator.exe` dans `<QTDIR >/Tools/QtCreator/bin` où `<QTDIR >` est le répertoire où vous avez choisi d’installer Qt.

## 2.2 Arduino

Pour pouvoir utiliser la carte Arduino et charger vos programmes dans celle-ci, il faut installer l’IDE Arduino. Pour ce faire, il faut :

- aller sur le site d’Arduino : <https://www.arduino.cc/>
- télécharger l’installateur (Onglet ”Software” puis ”Downloads”);
- lancer l’installateur;
- finaliser l’installation en suivant les instructions.

## 2.3 Github

Pour récupérer le code de ce projet, deux options sont possibles :

- Télécharger les sources
  - aller sur le site : <https://github.com/GuillaumeGibert/GBMProject3A>;
  - cliquer sur ”Clone or download”;
  - cliquer ”Download ZIP”.
- Cloner les sources
  - aller sur le site : <https://desktop.github.com/>
  - cliquer ”Download for Windows”;
  - installer Github Desktop;
  - cloner le projet.

## 2.4 GBMProject3A

Il faut charger dans la carte Arduino le projet `testArduino.ino` qui se trouve dans `<GBMProject3ADIR >/testArduino` où `<GBMProject3ADIR >/` est le répertoire où vous avez téléchargé les sources du projet. Un programme Arduino est composé de 2 fonctions minimum :

- une fonction `setup()` dans laquelle les variables sont initialisées qui est appelée une seule fois au début du programme;
- une fonction `loop()` qui est appelée régulièrement dès qu’elle est terminée.

Il faut ouvrir le fichier `GBMProject3A.pro` à l’aide du logiciel `qtcreator.exe`. Il faut ensuite éditer le fichier `GBMProject3A.pro` et changer la dernière ligne `src/ex8.cpp` par l’exemple que vous voulez lancer (choix possibles entre `ex1`,

ex2, ex3, ex4, ex5, ex6, ex7 et ex8). Pour créer votre propre programme, il vous suffit de créer un fichier `main.cpp` (par exemple) qui doit contenir une fonction `main` et d'éditer le fichier `GBMProject3A.pro` et changer la dernière ligne par `src/main.cpp`.

## 3 Code Arduino

### 3.1 Variables globales

Le code Arduino est composé de variables globales :

- `_nbPinIn` : nombre de pins analogiques connectés ;
- `_portBaud` : taux de transfert sur le port série ;
- `_fps` : fréquence de rappel de la fonction `loop()`.

### 3.2 `setup()`

La fonction `setup()` initialise le port série avec le taux de transfert spécifié dans la variable `_portBaud`.

### 3.3 `loop()`

La fonction `loop()` lit les valeurs sur les ports analogiques de la carte grâce à la commande `analogRead(int)`. Elle crée une chaîne de caractères de la forme :

```
Sensor:|<input1>|<input2>|<input3>|<input4>|<input5>|\n
```

Puis elle envoie cette chaîne de caractère sur le port série grâce à la fonction `Serial.println(String)`.

### 3.4 `serialEvent()`

La fonction `serialEvent()` est lancée entre deux appels à la fonction `loop()` si des données sont reçues sur le port série. Cela permet de récupérer sur la carte Arduino des informations en provenance du PC.

## 4 Exemple 1 : Récupérer des données depuis un Arduino (liaison filaire)

Côté PC, deux objets sont instantiés (voir Figure 1). Un objet de type `SerialPortManager` récupère les données envoyées par l'Arduino sur le port série. Il envoie ensuite, via un signal/slot Qt, les données vers un objet de type `Spy` qui les affiche à l'écran.



FIGURE 1 – Les données de l’Arduino sont captées par l’objet SerialPortManager qui les envoie à l’objet Spy.

## 5 Exemple 2 : Récupérer des données depuis un Arduino (liaison Bluetooth)

Cet exemple est similaire au précédent (cf. Figure 1). La seule différence réside dans le type de connexion entre l’Arduino et le PC. Dans l’exemple précédent, L’Arduino était connecté avec un câble USB au PC alors que dans cet exemple, il est connecté en Bluetooth via un module HC-05 par exemple.

## 6 Exemple 3 : Récupérer des données depuis un générateur de signaux

## 7 Exemple 4 : Afficher des signaux temporels

## 8 Exemple 5 : Filtrer des signaux temporels

## 9 Exemple 6 : Calculer une transformée de Fourier rapide (FFT)

## 10 Exemple 7 : Envoyer une chaîne de caractères à l’Arduino

## 11 Exemple 8 : Gérer une base de données ”patient” depuis une interface graphique

Il faut installer wamp. Attention il faut installer la même version que le compilateur de Qt (soit 32, soit 64bits). Une fois installé, il faut compiler le plugin mysql qui n’est pas présent par défaut dans Qt. Il faut ouvrir une invite de commande, puis taper :

```
cd %QTDIR%\qtdbase\src\plugins\sqldrivers
```

```
qmake — MYSQL_INCDIR=<WAMP_DIR>\bin\mysql\mysql5.7.28\include”  
MYSQL_LIBDIR=<WAMP_DIR>\bin\mysql\mysql5.7.28\lib”  
nmake sub-mysql  
nmake install
```

Si vous n'utilisez pas le compilateur de Visual Studio, il faudra remplacer nmake  
par mingw32-make