

Alexandre Berard, Julien Liottard, Marie Vialle

Client : Jean-Pierre Ceysson, CARI Electronic

Tuteur : André Lagrèze

Projet de deuxième année de DUT Informatique

IUT de Valence

**Compte rendu de travail du 2 au 6  
mars 2020**

07 mars 2020 (version 1)

# 1 Programmation de la carte

## 1.1 Création d'une fonction "printf"

Dans le but de pouvoir prendre conscience des valeurs obtenues lorsque nous lisons sur le pin analogique, nous devons trouver un moyen d'afficher un nombre décimal représentant la tension d'une carte à tester. Ces valeurs devront ensuite être transmises au réseau de neurones pour calculer le label de cette carte.

La fonction `printf` est une fonction utilitaire incluse dans les bibliothèques par défaut du langage C<sup>1</sup>. Ici, elle nous permettra de convertir le binaire lu sur le pin analogique en chaîne de caractère et de l'afficher en sortie sur une console.

Nous disposons d'un lien USB pour envoyer des programmes sur la carte, ce qui peut également servir de lien pour obtenir la chaîne transférée par notre "printf". Nous avons premièrement suivi les indications de ce site<sup>2</sup> pour savoir comment procéder. Il y est préconisé de vérifier la présence d'un port COMXX dans les périphériques installés sur l'ordinateur. Pour cela, il aura fallu installer le driver STSW\_LINK009 qui permet le lien entre PC et carte. L'IDE utilisé sur le site étant différent du nôtre, nous avons pas à notre disposition le fichier d'en-tête "mbed.h", et disposons donc pas toutes les fonctions utilisées pour générer notre "printf".

En parallèle, nous cherchions aussi comment afficher les caractères reçus depuis le programme "printf" de la carte. Notre tuteur nous a conseillé d'utiliser un logiciel que nous avons déjà eu l'occasion d'essayer en TP, RealTerm<sup>3</sup>. Après quelques minutes de configuration (débit : 115200 bauds, type de caractères : ASCII, port : 3), nous avons réussi à afficher quelques caractères qui correspondent au démarrage de la carte.

---

<sup>1</sup>Plus d'informations : <http://manpagesfr.free.fr/man/man3/printf.3.html>.

<sup>2</sup><https://os.mbed.com/teams/ST-Americas-mbed-Team/wiki/Creating-a-printf-console-output>

<sup>3</sup>Lien officiel : <https://realterm.sourceforge.io/>

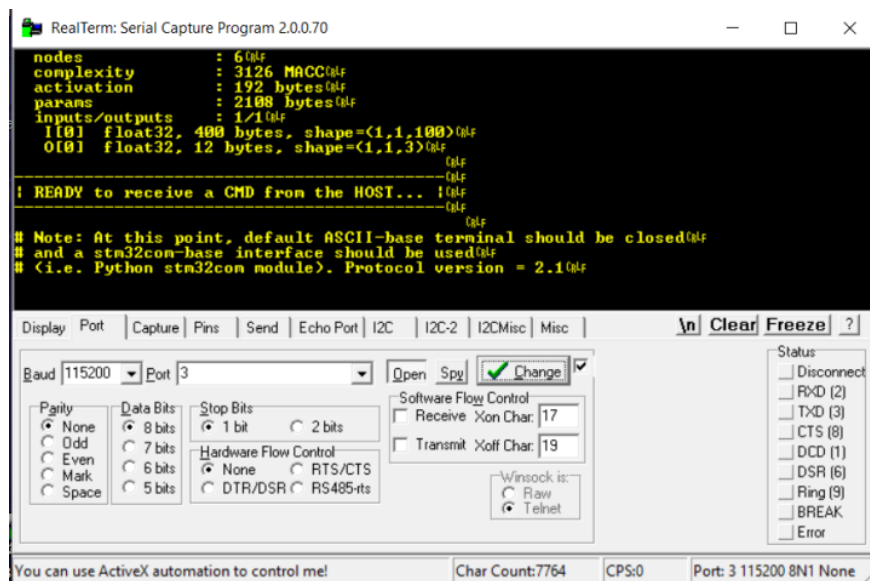


Figure 1 : Caractères reçus après allumage de la carte depuis RealTerm

## 2 Difficultés

Nous avons cependant rencontré quelques difficultés qui nous ont retardé dans notre exploitation de la carte.

### 2.1 Création de projet hors de l'IDE

Il est apparemment impossible de compiler un projet créé hors de l'IDE STM32CubeIDE. En effet, tous nos projets créés via STM32CubeMX<sup>4</sup> n'étaient pas lisibles dans STM32CubeIDE, pourtant capable de compiler ce genre de code.

### 2.2 GDB

Une erreur principale signalée par l'IDE d'ST était que le GDB était manquant dans le dossier utilisé pour le debug et que nous n'avions pas les droits pour le remettre. Une solution apportée à ce problème a donc été de l'ajouter via le terminal Ubuntu présent sur Windows en `su`<sup>5</sup>. Nous allons donc être plus vigilants pour les tests à venir, afin d'essayer de comprendre pourquoi cette erreur est apparue et comment y remédier si elle persiste dans les nouveaux projets que nous allons créer.

<sup>4</sup>Interface graphique ne générant que le code correspondant aux paramètres choisis

<sup>5</sup>Super-User

## 2.3 Branchement de la carte

Lorsque nous lançons le programme une fois la compilation terminée, le terminal nous affiche "Debugger connection lost".

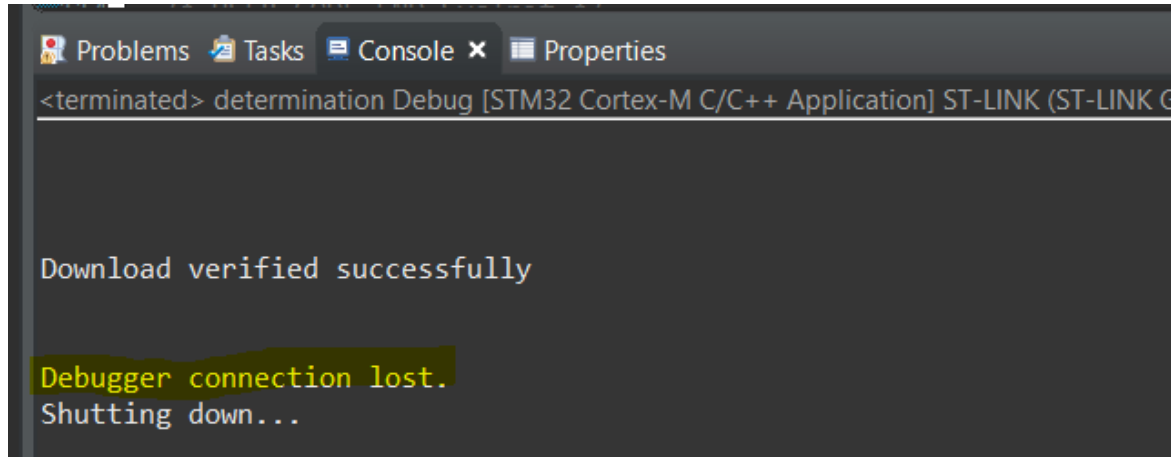


Figure 2 : Message de la console lors de l'exécution du programme sur la carte

Nous ne savons pas encore si cela est vraiment un problème ou si le programme à juste fini de s'exécuter.

## 3 Prévisions de travail pour la semaine du 9 au 13 mars

Cette semaine, sur les conseils de M. Lagrèze, nous allons tester la sortie des 3 USART de la carte. En effet, nous avons seulement testé la sortie de l'USART 3 pour le moment, nous allons donc voir les résultats sur le 1 et le 2 assez rapidement afin d'en exploiter les résultats et déterminer quelle sortie utiliser le plus rapidement possible.

Ce qui est sûr, c'est que d'après la Figure 1, on peut constater que les fonctions d'initialisation par défaut de l'IDE permettent d'afficher des caractères vers l'USB. En analysant de près ces fonctions, nous pourrions sûrement trouver un moyen d'envoyer des caractères sur ce lien. Reste à savoir si on peut utiliser le format '%f' de la fonction printf, qui convertit les variables de type 'float'<sup>6</sup> en chaîne de caractères, ou s'il faudra écrire à la main cette fonction. (Ce problème a déjà été posté dans de nombreux forums).<sup>7</sup>

<sup>6</sup>Représentation sous 16 bits d'un nombre décimal

<sup>7</sup><https://stackoverflow.com/questions/905928/using-floats-with-sprintf-in-embedded-c> "[...] on an embedded platform, it's quite possible that you don't have the full range of capabilities from the printf()-style functions."