



Elizabeth PAZ
Salem HARRACHE

Easy eLua

**eLua et approche Arduino sur
STM32F4-DISCOVERY**

Projet Innovant, Avril 2012

Table des matières

Contents	i
List of Figures	ii
List of Tables	iii
Introduction	1
Approche Arduino	1
1 Approche Arduino	3
1.1 Qu'est-ce qu'Arduino?	3
1.2 Avantages de l'utilisation de Arduino	4
1.3 Utilisation d'Arduino	4
2 elua sur STM32F4-DISCOVERY	7
2.1 Qu'est-ce que Lua?	7
2.2 Qu'est-ce qu'eLua?	8
2.3 Avantages de eLua	8
3 Designer l'esprit Arduino en LUA	11
4 Avancement du projet	13
5 Difficultés	15
6 Demo	17
7 Conclusion	19

Bibliographie	20
Appendix	21

Table des figures

1.1	Logo Arduino	3
1.2	Carte Arduino Uno	4
1.3	Environnement de travail : choix de la carte et du port	5
2.1	Logo Lua	7

Liste des tableaux

1.1	Création d'une nouvelle fonction en Arduino	5
1.2	Exemple Blink pour Arduino	5

Introduction

Présentation de Lua

Présentation de Arduino

Présentation de carte STM32F4-DISCOVERY

Approche Arduino

1.1 Qu'est-ce qu'Arduino ?



Figure 1.1 – Logo Arduino

Le système Arduino¹ est une plateforme *open-source* d'électronique programmée basée sur une carte à microcontrôleur de la famille AVR², et sur un environnement de développement intégré qui permet d'écrire, compiler et transférer un programme vers la carte. Ce logiciel utilise la technique du Processing/Wiring³.

Avec le système Arduino on peut réaliser divers tâches, par exemple le développement des objets interactifs indépendants : le prototypage rapide. Aussi, la domotique, qui grâce aux différents interrupteurs/capteurs permet de contrôler plusieurs sorties matérielles : contrôle des appareils domestiques, pilotage de robot, etc ... ou même charger des batteries par l'analyse et la production des signaux électriques. De plus, il peut communiquer avec des logiciels tournant sur l'ordinateur tel que Macromedia Flash, Processing, MaxMSP, etc

Le langage de programmation Arduino est une implémentation de Wiring, une plateforme de développement similaire, qui est basée sur l'environnement multimédia de programmation Processing.

Les cartes électroniques sont accessibles à tous, elles peuvent être achetées pré-assemblées ou être fabriquées manuellement, tout en ayant la totalité des informations nécessaire à l'assemblage.

1. Le projet Arduino a reçu un titre honorifique à l'Ars Electronica 2006, dans la catégorie Digital Communities

2. Cœur du processeur et famille de microcontrôleurs

3. Processing est un langage de programmation et un environnement de développement

1.2 Avantages de l'utilisation de Arduino

Le système Arduino a simplifié la façon de travailler avec les microcontrôleurs, en offrant plusieurs avantages pour les enseignants, les étudiants et les amateurs intéressés par d'autres systèmes. Ce système prend en charge des détails compliqués de la programmation des microcontrôleurs et les intègre dans une présentation facile à utiliser. Voici, plusieurs avantages qui propose Arduino :

Peu coûteux : En comparaison a d'autres plateformes les cartes Arduino sont peu coûteuses, les moins chères sont les versions qui peuvent être assemblées à la main.

Multi-plateforme : Le logiciel de programmation des modules Arduino est une application Java, libre et qui peut être tourné dans plusieurs systèmes d'exploitation tel que Linux, Windows et Mac.

Environnement de programmation clair et simple : Le logiciel est facile à utiliser pour les débutants (nous même l'ayant utiliser suite au cours d'initiation à l'Arduino) ; de plus, il reste flexible pour des utilisateurs avancés.

Logiciel Open Source et extensible : Le logiciel et le langage Arduino sont publiés sous licence Open Source qui est donc disponible à tous, ce qui permet donc la possibilité d'être complété et amélioré par des programmeurs plus expérimentés. Le langage Arduino, qui utilise le langage C++, peut être étendu grâce à l'aide des bibliothèques du C++.

Matériel Open Source et extensible : La version sur plaque d'essai de la carte Arduino peut être achetée à très bas coût et peut être réalisée par tout utilisateur, elle a pour but comprendre comment la carte fonctionne. De plus, tous les schémas de modules Arduino sont publiés alors les utilisateurs plus expérimentés en circuits peuvent apporter des améliorations à leur cartes.

1.3 Utilisation d'Arduino

Dans cette partie on va décrire certains points importants pour l'utilisation du logiciel, de la carte et du langage.



Figure 1.2 – Carte Arduino Uno

Un programme en langage Arduino doit obligatoirement être composé de ces deux fonctions :

- la fonction d'initialisation *setup()* qui est exécutée une seule fois au démarrage.
- la fonction "boucle sans fin" *loop()* qui est exécutée en boucle une fois que la fonction *setup()* a été exécutée une fois.

Puis, si besoin, autres fonctions peuvent être créées en suivant ce schéma :

```
1 type nom_fonction (arguments) {
2     // ici le code de la fonction
3 }
```

Table 1.1 – Création d'une nouvelle fonction en Arduino

```
1 /*
2  Blink
3  Turns on an LED on for one second, then off for one second, ...
4  repeatedly.
5  */
6 void setup() {
7     // initialize the digital pin as an output.
8     // Pin 13 has an LED connected on most Arduino boards:
9     pinMode(13, OUTPUT);
10 }
11 void loop() {
12     digitalWrite(13, HIGH);    // set the LED on
13     delay(1000);              // wait for a second
14     digitalWrite(13, LOW);    // set the LED off
15     delay(1000);              // wait for a second
16 }
```

Table 1.2 – Exemple Blink pour Arduino

L'utilisation du logiciel est facile et très clair, ce qui est très important à retenir c'est de bien vérifier avant d'envoyer le programme à la carte, si on utilise le bon port et la bonne carte.

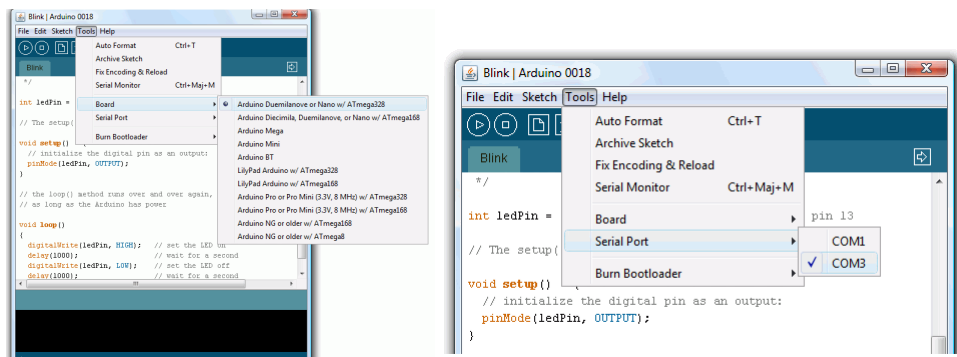


Figure 1.3 – Environnement de travail : choix de la carte et du port

elua sur STM32F4-DISCOVERY

2.1 Qu'est-ce que Lua ?

Lua est un langage de script libre, réflexif et impératif.

Créé en 1993, il est conçu de manière à pouvoir être embarqué au sein d'autres applications afin d'étendre celles-ci, comme en témoigne la compacité de son interpréteur (95 à 185 Ko pour la version 5.0.2, selon le compilateur utilisé et le système cible). Lua (qui signifie lune en portugais) a été développé par Luiz Henrique de Figueiredo, Roberto Ierusalimsky et Waldemar Celes, membres du groupe de recherche TeCGraf, de l'université de Rio de Janeiro au Brésil.

Lua est écrit en langage C ANSI strict, et de ce fait est compilable sur une grande variété de systèmes. Il est souvent utilisé dans des systèmes embarqués où sa compacité est très appréciée. Il profite de la compatibilité que possède le langage C avec un grand nombre de langages pour s'intégrer facilement dans la plupart des projets.

Il est particulièrement apprécié pour le développement de jeux vidéo comme pour l'interface du jeu World of Warcraft de Blizzard Entertainment, Garry's Mod ou encore Far Cry de Crytek Studios, SimCity 4 et prochainement dans Natural Selection 2 de Unknown Worlds Entertainment qui a entre autres développé Decoda, un IDE pour Lua. Lua a été porté dernièrement sur la console portable de Sony, la PSP. Une version Playstation 2 est en cours de portage. Ce langage est également utilisé pour la programmation de jeux de Nintendo DS, pour la programmation Roblox et récemment pour la TI-Nspire avec l'OS 3.0 ou ultérieur.



Figure 2.1 – Logo Lua

2.2 Qu'est-ce qu'eLua ?

elua adopte le langage de programmation Lua pour faire une complète implémentation dans le monde de l'embarqué, elua ajoute des caractéristiques spécifiques pour une efficacité, portabilité et développement de logiciels embarqués. eLua propose la totalité des caractéristiques de la version de bureau de Lua, et c'est important de remarquer que elua utilise les mécanismes de base pour pouvoir l'étendre avec des fonctionnalités de développement de l'embarqué optimisés et spécifiques.

2.3 Avantages de eLua

Complet contrôle de la plateforme : eLua runs on the "bare-metal". There is no Operating System between your programs and the microcontroller.

Portabilité du code : Like in Lua, you program in C, Lua or a mixture of both and your program runs in a wide varied of (sometimes radically different) platforms and architectures supported.

Transform hardware in commodity : Design and code your products for eLua and make them hardware-independent. Upgrade or completely change your hardware in the future and save time and money invested on the previous code development.

Development on targets : Fully functional Lua and a dedicated command shell on the microcontroller itself. No need to install development environments on the PC side, other than a serial or ethernet console/terminal emulator. Use any text editor, save your programs in sd/mmc and other media and use them directly in your platforms.

Flexibilité des produits : Add modern high level script-language capabilities to your projects, resulting in highly adaptable, field-programable and reconfigurable designs. Efficient (and cheap!) future evolution to your systems.

Shorter TTM : Optimizes Time to Market, shorter time to revenue, improved ability to hit critical market windows, agility to survive in turbulent market conditions

Embedded RAD : Prototype and experiment on a Rapid Application Development model. Test your ideas directly on the target platforms and cheap development kits. No need for simulators or future code adaptations.

Ready to use kits : A big (and growing) number of Open Source hardware and commercially available platforms are supported (see here). Prototype cheap and fast and design your final hardware later using the produced code.

Long cycle de vie : Add user configuration and scripting capabilities to your projects, making them adaptable to the always changing contexts of industrial processes, evolving engineering, automation standards, field optimizations etc...

Apprendre l'embarqué : Simple interactive and interpreted experimenting cycle. Use your desktop programming skills to become an embedded systems developer in no time and with a lot of fun.

Open Source : elua est libre, gratuit, et open source logiciel, comme Lua, elle possède une licence MIT qui permet l'utilisation de eLua dans des codes

“closed” source. Il n’y a aucune permission à demander, ils demandent juste de faire circuler l’information au monde : on utilise eLua !

CHAPITRE 3

Designer l'esprit Arduino en LUA

CHAPITRE 4

Avancement du projet

CHAPITRE 5

Difficultés

CHAPITRE 6

Demo

Conclusion

Bibliographie
