



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
3 Designer l'esprit Arduino en LUA	9
4 Avancement du projet	11
5 Difficultés	13
6 Demo	15
7 Conclusion	17
Bibliographie	18

Table des figures

1.1	Logo Arduino	3
1.2	Carte Arduino Uno	4
1.3	Environnement de travail : Logiciel Arduino	5

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.

3.

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

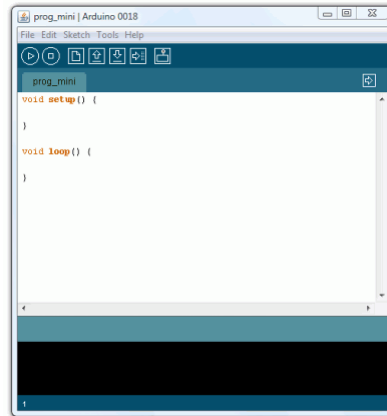


Figure 1.3 – Environnement de travail : Logiciel Arduino

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, ...
      repeatedly.
4   */
5  void setup() {
6    // initialize the digital pin as an output.
7    // Pin 13 has an LED connected on most Arduino boards:
8    pinMode(13, OUTPUT);
9  }
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

CHAPITRE 2

Elua sur STM32F4-DISCOVERY

CHAPITRE 3

Designer l'esprit Arduino en LUA

CHAPITRE 4

Avancement du projet

Difficultés

CHAPITRE 6

Demo

CHAPITRE 7

Conclusion

Bibliographie
