El Yassem Bilal

Garibal Florian

Année 2014-2015

*e*

Sommaire

[1 What is it about ? 3](#_Toc418602094)

[2 What did we have to do ? 4](#_Toc418602095)

[3 With what ? 5](#_Toc418602096)

[4 Project steps 8](#_Toc418602097)

[Conclusion 9](#_Toc418602098)

# What is it about ?

ATTENTE EXPLICATION de Keith

# What did we have to do ?

Le client souhaitait qu’un lien soit établi entre le fait de scanner une carte NFC et l’enregistrement de l’identifiant de cette carte sur un serveur spécifique. Le serveur en question étant, lui, un projet d’un précédent stagiaire qui consistait à gérer \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_. Pour cela il nous a été demandé de réaliser un programme C permettant premièrement de lire un fichier de config où serait stocké l’identifiant et le mot de passe de connexion à l’API du serveur puis deuxièmement de lire l’identifiant d’une carte scannée, préparer et envoyer une requête POST au serveur puis interprêter la réponse. Concernant l’interprétation, si aucune erreur n’était présente, il nous a été demandé d’allumer une DEL verte et dans le cas contraire d’allumer une DEL rouge. En outre, étant donné que c’était un prototype, il était aussi nécessaire de concevoir une boîte capable de contenir l’ensemble du système tout en étant la plus petite possible.

D’autre part, il nous fallait finir ce projet le plus vite possible en raison du retard important qu’il avait prit. De ce fait nous avons mis en place le Gantt prévisionnel suivant :



A SCHEMATISER

Requête POST 1j

XML parser 1/2j

libNFC 1j

Lien 2j

DEL 1j & 1/2

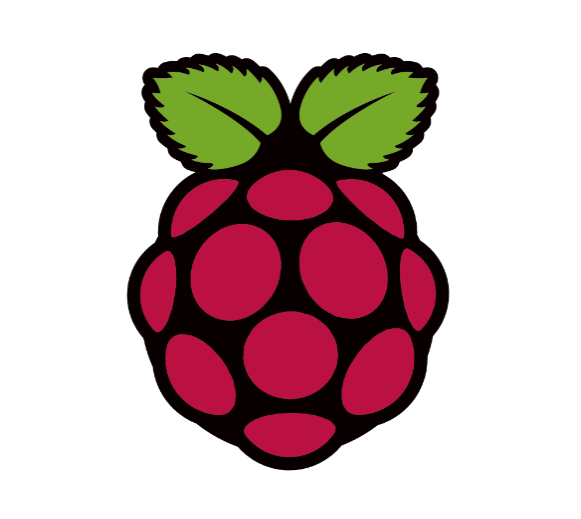
démarrage au boot RPi 1j

# With what means ?

Pour mener à bien ce projet il nous a été confié un certain nombre d’éléments hardware qui sont détaillés ci-dessous.

Pour commencer on nous a fourni un micro-contrôleur, le **Raspberry Pi B+ V1.2**, qui permet de réunir toutes les caractéristiques de base d’un ordinateur. En effet il renferme un processeur, de la RAM, des connectiques (USB, HDMI, etc…) mais aussi un slot pour carte SD qui lui permet de posséder un OS léger, ici Debian (Linux).

Il possède les caractéristiques suivantes :

****

* Dimensions : 85mm x 56mm x 17mm
* Processeur : Broadcom SoC tournant à 700MHz
* RAM : 512MB
* 1 port Ethernet
* 4 ports USB
* 1 port HDMI
* 1 prise d’alimentation microUSB 5V/2A
* 1 connecteur GPIO à 40 broches
* 1 sortie jack audio 3.5mm à 4-poles
* 1 slot pour carte MicroSD
* 4 trous de fixation sur un layout de type rectangulaire (en beige sur le schéma)

Photo et schéma du Raspberry Pi B+ V1.2

On nous a ensuite fourni une carte **Adafruit PN532 Breakout Board**. Cette carte est la plus populaire des cartes NFC et c’est celle qui est intégrée dans presque tous les téléphones ou appareils qui renferment la technologie NFC. En effet elle permet de lire et d’écrire des informations depuis une carte NFC ou même un tag(étiquette) NFC. D’autre part cette carte est utilisée par la librairie **libnfc** qui permet de créer des programmes interagissant avec sur tous systèmes d’exploitation (Windows, Linux, Mac). Deux cartes NFC nous ont par ailleurs été fournies.

Caractéristiques de la carte :

* Dimensions : 51mm x 117.7mm x 1.1mm
* Utilise une adresse I2C 7-bit 0x48

Photo de la carte Adafruit PN532 et d’une carte disposant de la technologie NFC

D’autre part pour créer un feedback, un retour d’information, à l’utilisateur, **deux DEL**, une verte et une rouge, étaient montées sur la breadboard. Pour ne pas endommager ces DEL, il était nécessaire de placer une résistance avant chacune d’entre elle. On nous a conseillé de prendre deux résistances de 220Ω ± 5% chacune.

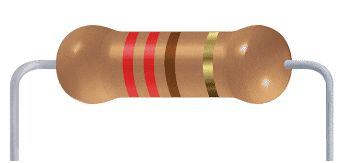


Photo et échelle de deux DEL (à gauche), image 3D d’une résistance 220Ω (à droite)

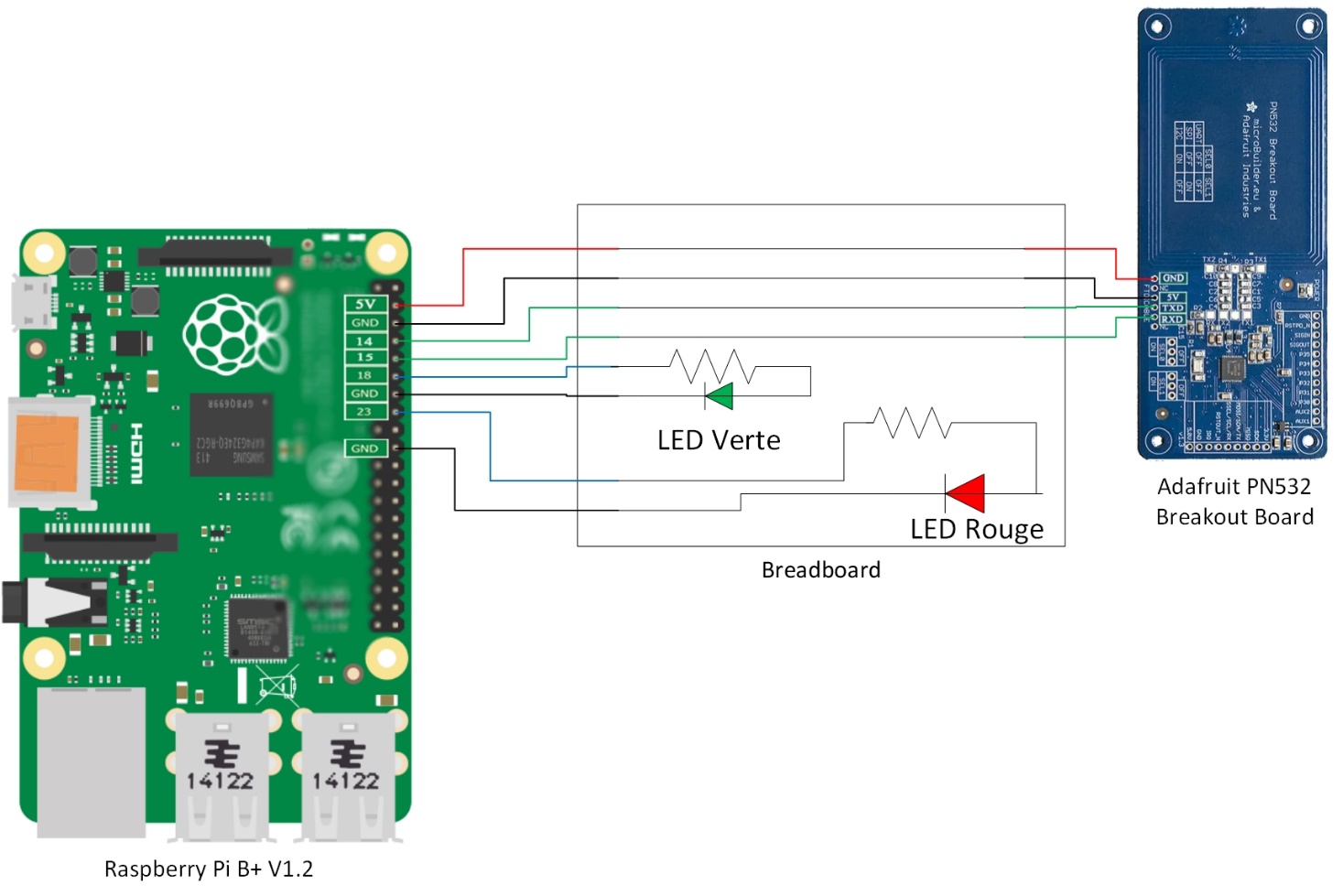
Pour lier ces deux cartes et les DEL, une **breadboard** (ou platine d’expérimentation) et des fils électriques nous ont été confié. Le tout était intégré dans le circuit suivant :

Schéma du circuit électrique réunissant les deux cartes et les DEL

Il nous a, par ailleurs, été demandé de travailler avec le **langage C** et la librairie **libnfc** qui fournissait des méthodes permettant d’interagir avec le lecteur de carte.

Pour réaliser ce projet, nous avons décidé, avec mon camarade Bilal El Yassem, d’utiliser les librairies **curl,** facilitant les échanges réseaux, et **json**, permettant de parser des fichiers json de manière simplifiée.

# Project steps

A REDIGER

• Ecrire un programme qui envoie une requête POST(libcurl)

• Ecrire un programme qui parse un fichier XML (ezxml)

• Comprendre la librairie libNFC (analyser tous les fichiers, recherche sur internet, ...)

Faire des tests pour arriver à récupérer l'ID d'une carte dans une variable

• Lier le fait de lire une carte et d'envoyer une requête POST à un serveur (RequestBin pour le moment car serveur down)

• Traiter la réponse du serveur en essayant de modifier le serveur pour qu'il nous la transmette en XML et non JSON

• Se rendre compte que c'est pas possible donc écrire un programme qui parse du JSON basique

• Modifier tous les parsages du programme de XML vers JSON

• Démonter le circuit des DEL et remonter pour que ça fonctionne

• Intégrer les DEL dans l'interaction avec la carte (allumage rouge puis vert si tout s'est bien passé)

• Faire démarrer le programme au boot du Raspberry

• Assembler les différents élément de la carte grâce à une project board, des fils soudés et deux "\_\_\_\_\_\_\_\_\_\_"

• Designer la boîte grâce à FreeCAD

• Imprimer la boîte

Conclusion