

# Cahier des charges

↳ WeMeshNetwork.

## Description du document

Titre	2016_CDC_FR_MeshNetwork
Date	2 Mars 2014
Chef de groupe	Jules Solinas
Auteurs	<ul style="list-style-type: none"><li>■ Jules Solinas: Chef de projet, développement de la communication et du marketing</li><li>■ Lucas Philippe: Développeur bas niveau</li><li>■ Jean-Louis Darmon: Développeur iOS</li><li>■ Mathieu Forel: Développeur Android</li><li>■ Justin Brottes: Création du prototype &amp; développeur web</li><li>■ Thomas Matysiak: Développeur bas niveau &amp; création du prototype</li></ul>
Sujet	Cahier des charges
Version du modèle	1.0
Format	PDF

## Tableau des révisions

Indice	Auteur	Sections	Modifications	Date
1	Justin Brottes		Création du document	23/02/2014

## Résumé du document

Ce cahier des charges vise à définir exhaustivement les « spécification de base » de notre projet MeshNetwork.

En interne ce document va servir à formaliser les besoins et à les expliquer aux différents acteurs pour s'assurer que tout le monde est d'accord. En particulier ce cahier des charges va permettre de cadrer les missions des différents membres du groupe, et notamment celles du chef de projet.

En externe ce document va servir de référentiel entre le lab EIP et l'équipe interne. Il est un outil fondamental de communication du chef de projet.

# Sommaire

<b>Partie 1 :</b> Rappels	Page 4
a. L'école Epitech	
b. Qu'est ce qu'un EIP ?	
c. Le projet WeMeshNetwork	Page 5
 <b>Partie 2 :</b> Environnement de réalisation	Page 6
a. Environnement de réalisation	
b. Environnement matériel	Page 7
c. Architecture technique	Page 8
1. Application / Connecteur	
2. Routeur / Application Mobile	Page 9
3. Site internet	Page 10
d. Gestion de la sécurité	Page 11
 <b>Partie 3 :</b> Description des parties à réaliser	Page 12
a. Application Mobile	
1. Partie interface utilisateur	
2. Partie interface / connecteur	Page 13
b. Création du connecteur	Page 15
c. Création du Firmware	Page 15
 <b>Partie 4 :</b> Organisation du projet	Page 16
a. Répartition des tâches	

# Partie 1

↳ Rappels.

## a. L'école Epitech

EPITECH quant à elle, est une école d'expertise informatique. Son cursus et sa pédagogie est innovatrice. En effet, elle se base principalement sur des compétences pratiques plutôt que théoriques comme les autres écoles. Epitech se déroule sur cinq années dont une à l'étranger. Durant la scolarité, les élèves auront aussi la possibilité de faire plusieurs stages et alternances en entreprises.

## b. Qu'est ce qu'un EIP ?

Un EPITECH Inovative Project (EIP) est un projet à faire et à réaliser sur deux années. Il débute dès la troisième année Epitech pour se conclure à la fin de la cinquième et dernière année du cursus. Son but est de créer un projet innovant qui permettra pour certains groupes de le commercialiser ou construire un avenir autour (création d'une start-up par exemple).

## Le projet WeMeshNetwork.

Notre projet se nomme pour l'instant **WeMeshNetwork**. Son but est de permettre aux usagers de pouvoirs communiquer entre eux par le biais d'un réseau de partage d'un téléphone à un autre sans utiliser de réseau de télécommunication. L'utilité de cette idée sera principalement dans les lieux où le réseau téléphonique n'est pas présent ou saturé.

Pour résoudre ce problème, nous avons l'idée de connecter entre-eux les téléphones mobiles présents sur la zone à l'aide d'une application mobile ainsi que d'un connecteur. Ces liaisons servent à créer un réseau de communication semblable à celui d'internet, c'est à dire décentralisé, maillé entre les machines ou plutôt « méché ».

Connectés via l'application sur ce réseau, les utilisateurs pourront, appeler, discuter, partager du contenu via les numéros de téléphone de leurs contacts.

Ce projet est essentiellement consacré à l'événementiel. Nous voulons proposer aux organisateurs de communiquer leur contenu via l'application aux usagers présents. Nous assurons également un service minimum de fonctionnement à l'aide de relais.

Les problèmes majeurs des projets déjà existant sont qu'ils ciblent un nombre trop petit d'utilisateur par rapport à la surface visée. Le connecteur que nous proposons sert à simplifier le fonctionnement. Cela permet d'augmenter la compatibilité entre les téléphones mais également d'augmenter le nombre de personnes ciblées.

# Partie 2

↳ Environnement de réalisation.

## a. Environnement de réalisation

Nous devons réaliser trois éléments bien distincts chacun possédant des caractéristiques techniques et des moyens de conception qui leurs sont propres.

Nous devons dans un premier temps créer une application mobile utilisable sur iOS et Android. Celle-ci est le coeur du projet, elle va permettre aux utilisateurs de communiquer et donc de créer le réseau méché. Cependant, les brevets et autres protections nous empêche de faire correspondre les émetteur wi-fi du téléphone avec l'application.

La deuxième étape est donc de créer un prototype de connecteur jack 3.5 utilisable sur tous les téléphones mobiles. Il s'agit en réalité d'un émetteur / récepteur wi-fi. Ce dernier nous sert à faire communiquer les téléphones en effectuant une conversion analogique du signal (onde wi-fi) en le rendant numérique (binaire).

La dernière étape de la réalisation du projet consiste à développer un logiciel compatible avec n'importe quel routeur. Il communique avec les autres téléphones et assure un service minimum dans le cas ou il n'y aurais pas assez d'utilisateur. Il sert de noeux dans le réseau, il gère l'accès et les requêtes vers des réseaux externes (internet, 3g ...).

## b. Environnement matériel

L'application contient deux couches bien distinctes (application et conversion de signal) et sera développé sur deux plate-formes:

### iOS:

- Nous utilisons le langage objective C et le logiciel XCODE pour développer.
- Afin d'effectuer la couche conversion, nous utilisons le C que nous compilons directement dans l'application à l'aide du compilateur C d'XCODE.
- Pour stocker l'application et la rendre publique lors de la mise en production, nous utilisons l'App Store d'Apple.

### Android:

- Nous utilisons le langage JAVA et le logiciel Eclipse pour développer.
- Afin d'effectuer la couche conversion, nous utilisons à l'aide de la librairie NAG le langage C. ([www.nag.co.uk/IndustryArticles/CallingCLibraryRoutinesfromJava.pdf](http://www.nag.co.uk/IndustryArticles/CallingCLibraryRoutinesfromJava.pdf))
- Pour stocker l'application et la rendre publique lors de la mise en production, nous utilisons le PlayStore de google.

Afin de réaliser le prototype du **connecteur jack 3.5**, nous utilisons plusieurs composants électroniques ainsi que des outils nous servant à les manipuler:

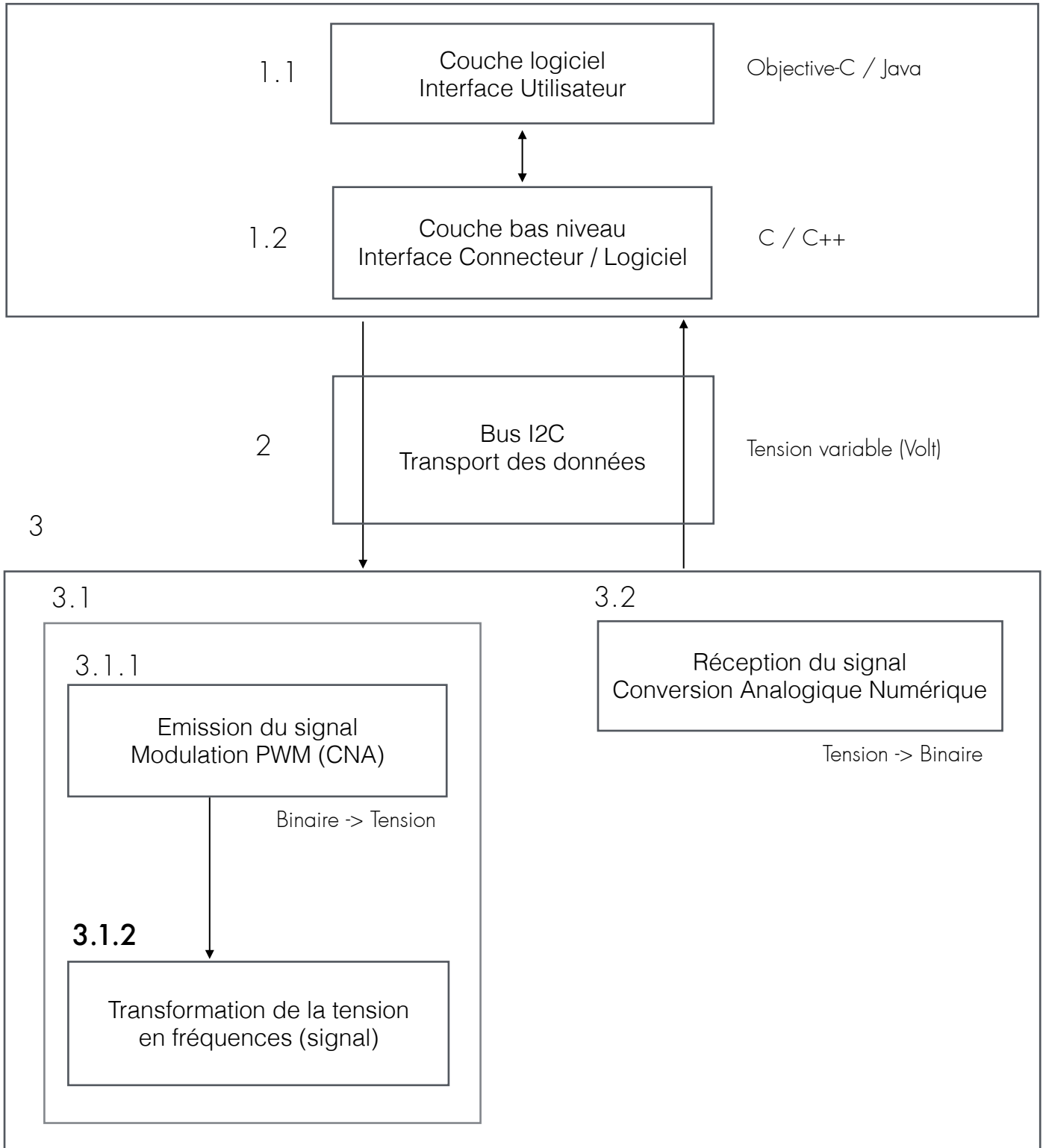
- Un micro émetteur récepteur wifi (Transceiver IC) tel que le Si446x LC Transceivers de Silicon Labs. ([www.silabs.com/products/wireless/EZRadioPRO/Pages/si446x.aspx](http://www.silabs.com/products/wireless/EZRadioPRO/Pages/si446x.aspx))
- Un convertisseur analogique-numérique pour échantillonner le signal.
- Un connecteur jack 3.5.
- Création d'un schéma fonctionnel sous ISIS.
- Routage de la carte sous ARES.
- Fabrication d'un prototype et test de la maquette.
- Nous utilisons un bus I2C pour transmettre les informations entre les composants.
- Un générateur de tension continu 5V.
- Câblages.
- Poste de soudure.

Nous réaliserons également un site internet pour promouvoir notre projet et présenter l'équipe. Celui-ci sera héberger sur un serveur Debian 7, utilisant Nginx comme serveur web et sera développé en HTML, CSS, PHP, SQL et JavaScript.

## c. Architecture technique

### 1. Application / Connecteur:

1

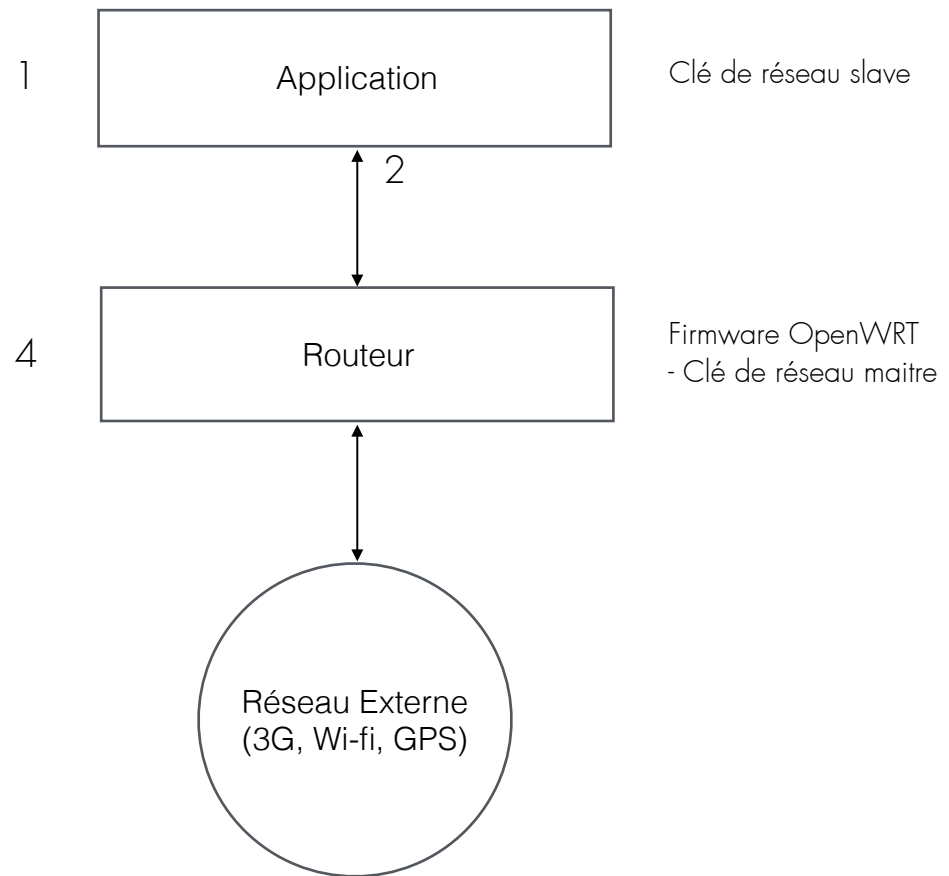


Transceiver IC

1: Téléphone mobile / 2: Connecteur / 3: Emetteur récepteur wifi



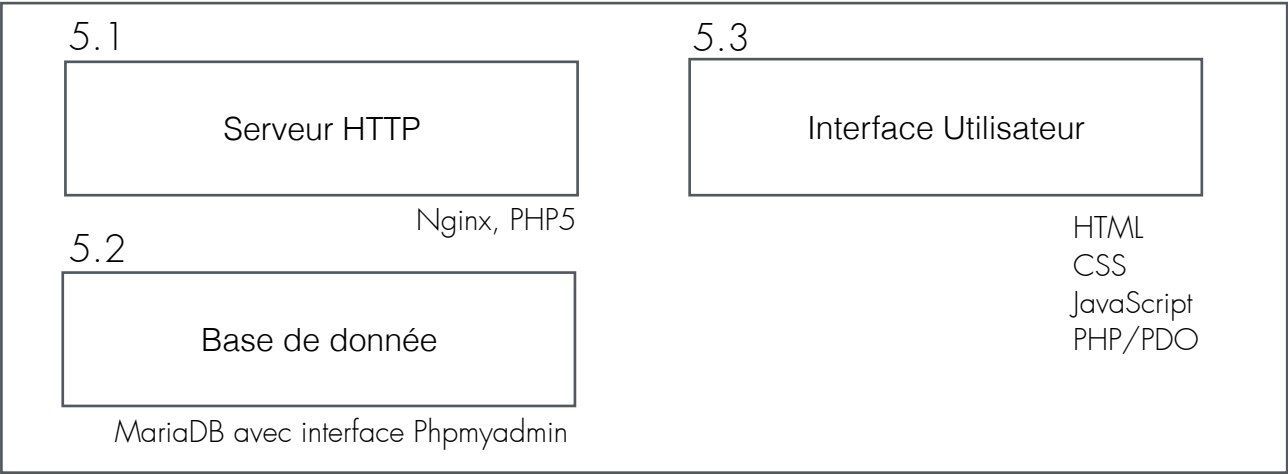
2. Routeur / Application Mobile:



**1: Téléphone mobile / 2: Connecteur / 4: Firmware (Routeur)**

3. Site internet:

5



**5: Serveur web**

## d. Sécurité du projet

Cette application étant orientée client, la sécurité reste un point essentiel afin de protéger les données / informations des utilisateurs et leurs communications. En effet, l'élaboration de ce projet nécessite une certaine rigueur non seulement en terme de performances mais aussi en terme de sécurité d'utilisation. Il est impératif que l'utilisateur ne puisse pas récupérer les données des autres utilisateurs qui transitent par son propre appareil.

Pour cela, nous mettons en place une clé unique générée à chaque installation de l'application par l'utilisateur. Elles peuvent bien entendu être régénérées à n'importe quel moment. Elle sert à identifier le propriétaire de chaque information. Nous utilisons un chiffrement asymétrique et nous chiffons celle-ci à l'aide du protocole GPG (l'open source du PGP). Afin d'identifier la validité de chaque informations (dans le cas d'un man-in-the-middle par exemple), nous vérifions la validité de celle-ci à l'aide de cette clé unique.

La politique de sécurité sur le serveur du site internet sera aussi élevée : firewall, connexions au serveur par clé privées/publiques, détection des bruteforces et scanneur de ports.

# Partie 3

↳ Description des parties à réaliser.

## a. Application mobile

### 1. Partie interface utilisateur:

Cette partie met en place l'application mobile et ses fonctionnalités liées à l'utilisateur. (Toutes à détailler, certaines n'ont pas encore été traitées par le groupe).

- Connexion à l'application à l'aide de son numéro de téléphone (ID unique).
- Génération de clé unique.
- Méthodes de chiffrement.
- Liaison de ce numéro à l'ID du téléphone.
- Gestion des amis (Listes / Ajout / Suppression / Groupe / Importation via les APIs des réseaux sociaux).
- Système de messagerie texte.
- Système de création de groupe de discussion
- Système de création de canaux de discussion lié à un événement.
- Système de création de groupe de discussion selon la position géographique.
- Appel vocaux (Voir vidéo).
- Partage de fichiers.
- Possibilités aux administrateurs d'événements de publier du contenu.
- Historique local sur le mobile.

## 2. Partie interface / connecteur:

Cette partie concerne la conversion des signaux reçu ou a envoyer.

- Utilisation de librairie C en Java.
- Chiffrement de l'information à transmettre (GPG).

### - **Conversion Numérique Analogique par modulation PWM:**

C'est une manière simple et efficace de générer une tension analogique avec un microcontrôleur.

Peu d'entre eux sont équipés d'un convertisseur numérique analogique.

Exemple de génération de PWM en C:

```
int main() {
    DDRD = (1<<PD5);
    char commande=25;
    char temps;
    volatile char i;
    while (1) {
        PORTD |= (1<<PD5);
        for(temps=0; temps<commande; temps++){
            for (i=0; i<50; i++);
        }
        PORTD &= ~(1<<PD5);
        for(temps=0; temps<(100-commande); temps++){
            for (i=0; i<50; i++);
        }
    }
}
```

- Conversion Analogique Numérique.
- **Initialisation et envoi ou réception des données sur le bus I2C:**

Ce bus a été choisi car il permet de simplifier la connectique (2 connexions pour communiquer et 2 connexions pour les alimentations):

- Un signal de données (SDA - Signal Data), est utilisé pour transmettre les données.
- Un signal d'horloge (SCL - Signal Clock) est utilisé pour transmettre un signal d'horloge synchrone (signal qui indique le rythme d'évolution de la ligne SDA).
- Un signal de référence (Masse).

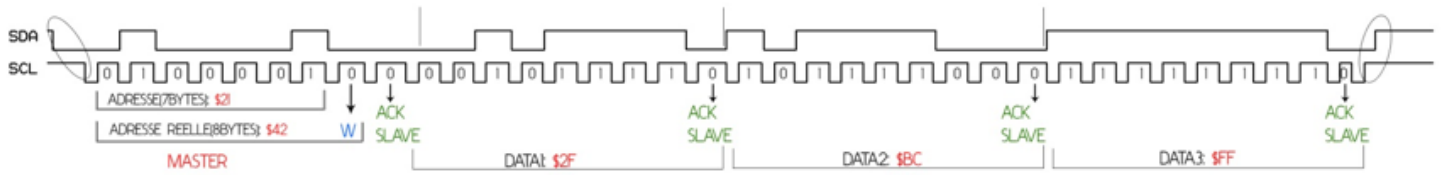
4 modes de transferts possible:

- jusqu'à 100 K bits/s en mode « standard », (standard mode) ;
- jusqu'à 400 K bits/s en mode « rapide », (Fast-mode) ;
- jusqu'à 1 Mégabit/s en mode « rapide plus », (Fast-mode Plus) ;
- jusqu'à 3,4 Mégabits/s en mode « haute vitesse », (High-speed mode).

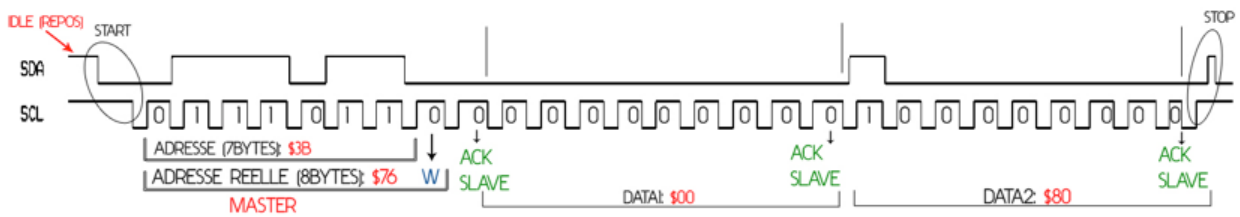
Chaque composant I2C peut être appelé par un autre composant I2C. Pour simplifier l'identification, chaque famille a une adresse spécifique (dont une partie peut être ajustée matériellement ou logiciellement). Le format de l'adressage se fait sur 7 bits.

Il faut utiliser des sorties à collecteur ouvert (ou à drain ouvert pour des circuits CMOS).

Trame n° 1 : Ecriture sur l'afficheur d'adresse 0x76, de deux bits : 0x00 & 0x80



Trame n° 2 : Ecriture à l'adresse 0x42, des trois données 0xBC 0x2F & 0xFF



Algorithme d'une communication sur le bus I2C



## b. Création du connecteur

- Création de la carte
- Choix des composants
- Modélisation du connecteur sous SolidWorks
- Schéma fonctionnel
- Routage
- Soudure des composants

## c. Création du Firmware

- Création d'une clé réseau maître
- Utilisation de la technologie OpenWRT
- Chiffrement asymétrique des informations
- Détection d'un réseau méché
- FireWall
- Gestion et balance des requêtes utilisateurs

# Partie 4

↳ Organisation projet

## a. Répartition des tâches

Nom	Prénom	Fonction
Jules	Solinas	Chef de projet, développement de la communication et du marketing
Forel	Matthieu	Développeur Android, réalisation de l'application en Java
Brottes	Justin	Création du prototype, choix des composants, réalisation de la partie électronique
Philippe	Lucas	Développeur bas niveau en charge du Firmware et de la conversion des signaux sur le connecteur
Matysiak	Thomas	Développeur bas niveau & création du prototype
Darmon	Jean-Louis	Développeur iOS, réalisation de l'application en Objective-C