

RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE
MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITÉ IBN-KHALDOUN DE TIARET

FACULTÉ DES SCIENCES APPLIQUÉES
DÉPARTEMENT DE GENIE ELECTRIQUE



MEMOIRE DE FIN D'ETUDES

Pour l'obtention du diplôme de Master

Domaine : Sciences et Technologie

Filière : Electronique

Spécialité : Electronique des systèmes embarqués

THÈME

*Développement d'un système embarqué pour la
commande d'une matrice de commutation*

Préparé par : Ahmed Salima
Abdellah Fatima

Devant le Jury :

Nom et prénoms	Grade	Université	Qualité
MAASKRI Mustapha	MAA	U-TIARET	Président
BENABID Houari	MAA	U-TIARET	Examineur
Adda BENATTIA Abderrahmane	MAA	U-TIARET	Encadreur

PROMOTION 2018 /2019

Remerciement

NOUS TENONS TOUT D'ABORD À
REMERCIER DIEU LE TOUT PUISSANT
ET MISÉRICORDIEUX, QUI NOUS A
DONNÉ LA FORCE ET LA PATIENCE
D'ACCOMPLIR CE MODESTE TRAVAIL.

EN SECOND LIEU, NOUS TENONS À
REMERCIER NOTRE ENCADREUR MR :
ADDA BENATTIA ABDERRAHMANE,
SON PRÉCIEUX CONSEIL ET SON AIDE
DURANT TOUTE LA PÉRIODE
DU TRAVAIL.



NOUS TENONS À EXPRIMER NOS
SINCÈRES REMERCIEMENTS À TOUS LES
PROFESSEURS QUI NOUS ONT ENSEIGNÉ
ET QUI PAR LEURS COMPÉTENCES NOUS
ONT SOUTENU DANS LA POURSUITE DE
NOS ÉTUDES.

ENFIN, NOUS TENONS ÉGALEMENT À
REMERCIER TOUTES LES PERSONNES
QUI ONT PARTICIPÉ DE PRÈS OU DE LOIN
À LA RÉALISATION DE CE TRAVAIL.



Sommaire

Introduction générale

Chapitre I: Généralités sur objets connectés et systèmes embarqués

1	Introduction	2
2	Internet des Objets.....	2
2.1	Application de l'internet des Objets.....	2
2.2	Avantage de l'internet des Objets	2
3	Les systèmes embarqués.....	3
3.1	Définition :	3
3.2	Avantage des systèmes embarqués	4
3.3	Application des systèmes embarqués	4
3.4	Des exemples des systèmes embarqués	5
3.4.1	PcDuino	5
3.4.2	Arduino.....	6
3.4.3	Raspberry.....	7
3.4.4	Red Pitaya.....	8
4.	Conclusion.....	9

Chapitre II: Conception d'une manipulation à distance

1.	Introduction	10
2.	Matrice de commutation.....	10
2.1	Switch numérique	10
2.1.1	Les type de switch	11
2.2	Relais.....	12
2.2.1	Relais électromécaniques	12
2.3	Rôle de la matrice de commutation.....	12

2.3.1	Switch ADG202	12
2.4	La commande de la matrice de commutation	13
3.	Architecture Hard	14
4.	Architecture logicielle (software)	15
5.	Outil de développement Hard	16
5.1	PcDuino.....	16
5.2	Arduino	17
5.3	Communication de arduino	17
5.4	RedPitaya	18
6.	Software : Architecture logicielle	20
6.1	Front end	20
6.1.1	JAVA Script	20
6.1.2	HTML.....	20
6.1.3	CSS : feuille de style	21
6.2	Back end.....	21
6.2.1	Node.js.....	21
6.3	Communication entre pcDuino et Windows	22
6.3.1	WinSC	22
6.3.2	PuTTY.....	23
7.	Conclusion.....	24

Chapitre III: Application et Réalisation

1.	Introduction :	25
2.	Objectif.....	25
3.	Mode opératoire	25
3.1	Première patrie : L'interface de pcdduino	25
3.2	Deuxième partie : L'interface de commutation	26

3.3	Troisième partie : Le choix entre les arduinos.....	28
3.4	Quatrième partie : La transmission des trames.....	30
3.5	Cinquième partie : L'analyse des trames.....	31
3.6	Sixième partie : Validation des paramètres	33
4.	Conclusion.....	33

Conclusion générale

Liste des figures

Figure 1-1. Architecture d'un Système embarqué	3
Figure 1-2. Le montage de réalisation	5
Figure 1-3. La mesure analogique obtenue par la réponse de la requête Ajax	5
Figure 1-4. Montage de réalisation d'application avec arduino	6
Figure 1-5. L'interface de l'application d'android communiquant en bluetooth	6
Figure 1-6. Les étapes d'allumage des LEDs par Raspberry	7
Figure 1-7. Les étapes d'extinction des LEDs par Raspberry	7
Figure 1-8. Montage de l'application avec red pitaya, et traceur de bode (circuit RC)	8
Figure 1-9. Les circuits connectés au red pitaya	8
Figure 2-1. Schéma d'une matrice de commutation.....	10
Figure 2-2. Les types de switch.....	11
Figure 2-3. Configuration des pins d'ADG202A	13
Figure 2-4. Schéma de Montage.....	14
Figure 2-5. Architecture Logicielle	15
Figure 2-6. Architecture interne d'un pcDuino	16
Figure 2-7. Composants nécessaires pour lancer un pcDuino.....	17
Figure 2-8. Architecture interne d'Arduino.....	17
Figure 2-9. Composants de lame pour une transmission série	18
Figure 2-10. Redpitaya	18
Figure 2-11. Interface web de Redpitaya avec icones des applications	19
Figure 2-12. La configuration de WinSCP	22
Figure 2-13. Les fichiers de pcDuino et d'ordinateur	22
Figure 2-14. La configuration de PuTTY	23
Figure 2-15. Le terminal de PuTTY	23
Figure 3-1. Interface de pcDuino dans un ordinateur.....	25
Figure 3-2. Les fichiers de pcDuino affichés sous WinScp	26
Figure 3-3. Le code source du serveur de commutation.....	26
Figure 3-4. L'interface qui commande la matrice de commutation	27
Figure 3-5. Le programme qui génère l'interface de commutation.....	27
Figure 3-6. Le composant du montage	28
Figure 3-7. Le programme du choix des arduinos.....	28

Liste des figures

Figure 3-8. L'interface web de manipulation	29
Figure 3-9. Le code source de l'interface web en HTML	29
Figure 3-10. Le Red Pitaya et son interface web.....	30
Figure 3-11. Les programmes des arduinos.....	30
Figure 3-12. La trame transmis par le premier Arduino.....	31
Figure 3-13. Analyse de la première trame	31
Figure 3-14. La trame transmise par le deuxième Arduino	32
Figure 3-15. Analyse de la deuxième trame	32
Figure 3-16. Les réponses correctes des Arduino.....	33

Introduction

Introduction générale

Les technologies de l'information et de la communication représentent la révolution la plus innovante qui marque ces dernières années. En particulier de nombreuses applications correspondent à des systèmes dits embarqués, qui généralement sont composés de sous-systèmes. De la même manière, beaucoup de systèmes actuels présentent plusieurs comportements au cours du temps, afin de réaliser différentes tâches.

Ce travail vise la conception et le développement d'un système embarqué pour la commande d'une matrice de commutation afin de réaliser une manipulation à distance en utilisant différentes techniques, applications et outils tels que pduino, redpitaya et arduino du côté hardware et pour le développement web en utilisant JavaScript, html, et css du côté software.

Dans ce travail, nous voulons répondre à différentes questions, à savoir :

- Qu'est-ce que c'est une matrice de commutation ?
- Comment commander un circuit à distance ?
- C'est quoi un pduino et un Redpitaya ?
- Comment développer une interface web et quel est son rôle ?

Pour répondre à ces questions, nous avons pensé à réaliser une application sous forme d'un système embarqué.

La suite du mémoire est organisée selon les chapitres suivants

- Chapitre 1 est consacré à la définition des objets connectés, et les systèmes embarqués, ses applications et leurs avantages, nous avons aussi cité quelques exemples des systèmes embarqués.
- Chapitre 2 représente la conception de la gestion de la matrice de commutation, l'architecture matérielle et logicielle et les outils de développement.
- Chapitre 3 focalise sur les différentes étapes de réalisation de l'application, le traitement et les résultats d'une manipulation (cas de TP).

Chapitre I

1 Introduction aux objets connectés et systèmes embarqués

Les objets connectés sont des systèmes matériels communicants qui supportent un logiciel applicatif embarqué [1].

En transformant les objets physiques en objets numériques, intelligents, autonomes et communicants, les technologies des systèmes embarqués modifient notre environnement. Facteurs de compétitivité, elles permettent d'évoluer vers des systèmes plus intelligents et plus efficaces dans de nombreux domaines d'applications (transports, défense, télécommunications...).

Aujourd'hui, Internet ne se limite plus aux ordinateurs, Smartphones et tablettes. Il existe désormais une multitude d'objets connectés qui forment un réseau étendu : l'Internet des objets. Le champ d'application des objets connectés est très large : énergie (capteurs), sécurité (caméras de vidéosurveillance), santé (bracelet connecté) etc [2].

2 Internet des Objets

L'Internet des objets est une concrétisation technique de l'informatique ubiquitaire où la technologie est intégrée naturellement aux objets du quotidien. Très prometteur ce concept ouvre la voie vers une multitude de scénarios basés sur l'interconnexion entre le monde physique et le monde virtuel : domotique, e-santé, ville intelligente, logistique, sécurité, ... etc [3].

2.1 Application de l'internet des Objets

L'Internet des Objets (IoT) est un réseau des réseaux (représentant également l'expansion du Web et de l'industrie), qui permet via des systèmes d'identification électronique normalisés et unifiés – connectés, interconnectés avec les humains et interopérabilité entre robots et humains-robots/IA) – et des dispositifs mobiles sans fil, d'identifier directement des entités numériques ou physiques et ainsi de pouvoir récupérer, stocker, transférer et traiter sans discontinuité, entre les mondes physiques et virtuels, les données s'y rattachant. [4]

2.2 Avantage de l'internet des Objets

Aujourd'hui, de nombreux secteurs comme la banque, les télécoms ou l'énergie considèrent déjà l'Internet des objets comme un levier important de fidélisation client. Grâce aux informations recueillies via les objets connectés, ils peuvent désormais proposer des services spécifiques et influencer le comportement des clients sur un point de vente, qu'il soit physique ou virtuel.

Mieux, l'objet connecté n'est pas qu'un moyen de communication, puisqu'il remplit une fonction particulière: il permet de valider immédiatement une modification de comportement du consommateur. Il devient possible de vérifier, par exemple, la pertinence d'un partenariat entre un constructeur de réfrigérateurs connectés et un service de courses en ligne lorsque des commandes sont effectuées grâce à la connectivité de l'équipement ménager. [5]

3 Les systèmes embarqués

Un système embarqué est un système complexe qui intègre du logiciel et du matériel conçus ensemble afin de fournir des fonctionnalités données. Il contient généralement un ou plusieurs microprocesseurs destinés à exécuter un ensemble de programmes définis lors de la conception et stockés dans des mémoires.

3.1 Définition :

Quelle que soit la nature et la complexité du système, on décompose un système embarqués en :

- le système contrôlé
- le système de contrôle

Le système contrôlé = environnement (procédé) équipé d'une instrumentation qui réalise l'interface avec le système de contrôle

Le système de contrôle = éléments matériels (microprocesseurs..) et logiciels dont la mission est d'agir sur le procédé via les actionneurs en fonction de l'état de ce procédé indiqué par les capteurs de manière maintenir ou conduire le procédé dans un état donné.

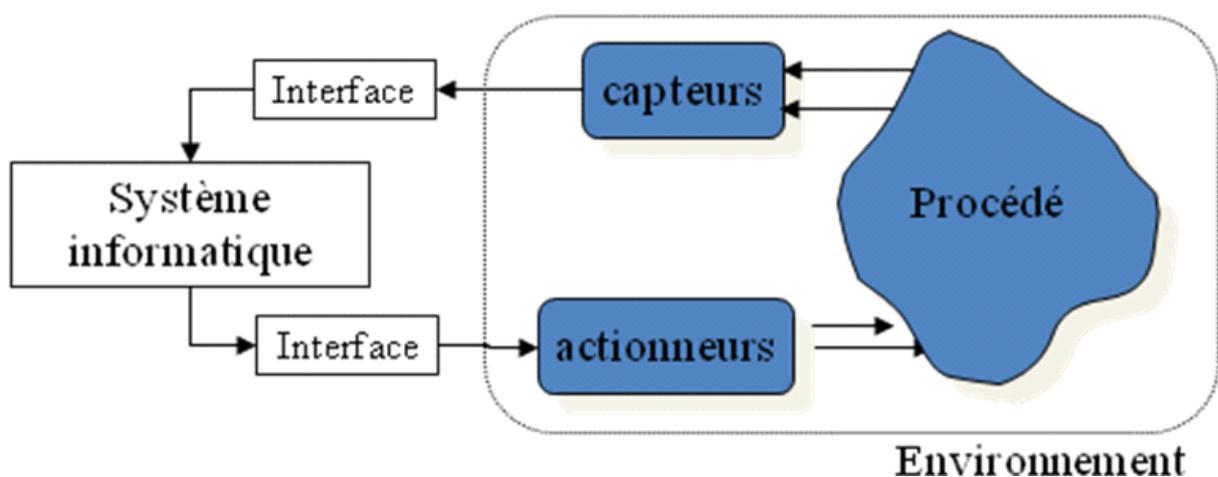


Figure 1.1 Architecture d'un Système embarqué

3.2 Avantage des systèmes embarqués

Le système matériel et l'application (logiciel) sont intimement liés et immergés dans le matériel et ne sont pas aussi facilement discernables comme dans un environnement de travail classique de type ordinateur de bureau PC (*Personal Computer*).

Un système embarqué est autonome et ne possède pas des entrées/sorties standards tels qu'un clavier ou un écran d'ordinateur. Contrairement à un PC, l'interface IHM (Interface Homme machine) d'un système embarqué peut être aussi simple qu'une diode électroluminescente LED (*Light Emitter Diode*) qui clignote ou aussi complexe qu'un système de vision de nuit en temps réel ; les afficheurs à cristaux liquides LCD (*Liquid Crystal Display*) de structure généralement simple sont couramment utilisés.[6]

3.3 Application des systèmes embarqués

L'électronique embarquée est introduite dans divers domaines, à savoir :

- **Le domaine grand public :**
Smart phone, console de jeux, appareil photos, lecteur audio
- **Les moyens de transport :**
Ordinateur de bord, GPS, système de navigation, automobiles, avions, trains, bateau
- **Les équipements médicaux:**
Imagerie (rayon X, ultra-sons, IRM) endoscopie, caméra, monitoring, perfusion, lasers, chirurgie, stimulateur cardiaque
- **Les équipements de télécommunication :**
Station mobile, routeur, gateway, satellite
- **Les équipements industriels :**
Productions automatisées, systèmes de commande d'énergie, équipements de stockage
- **Les équipements de bureautiques :**
Répondeurs, copieurs, imprimante.,,
- **Les équipements de bâtiment :**
Ascenseurs, système de surveillance, contrôle d'accès, systèmes d'éclairage... [7]

3.4 Des exemples des systèmes embarqués

Les systèmes embarqués les plus basiques ne disposent pour seule interface utilisateur que de simples boutons ou LED.

Il existe aussi d'autres systèmes embarqués tel que Pcdduino, Arduino, Raspberry et Redpitya

3.4.1 Pcduino

Un exemple d'un pcduino qui est connecté au réseau, et la sortie variable de la résistance variable est connecté à la broche analogique A2, les deux autres broches de la résistance variable étant connectées entre le 0V et le 3.3V.

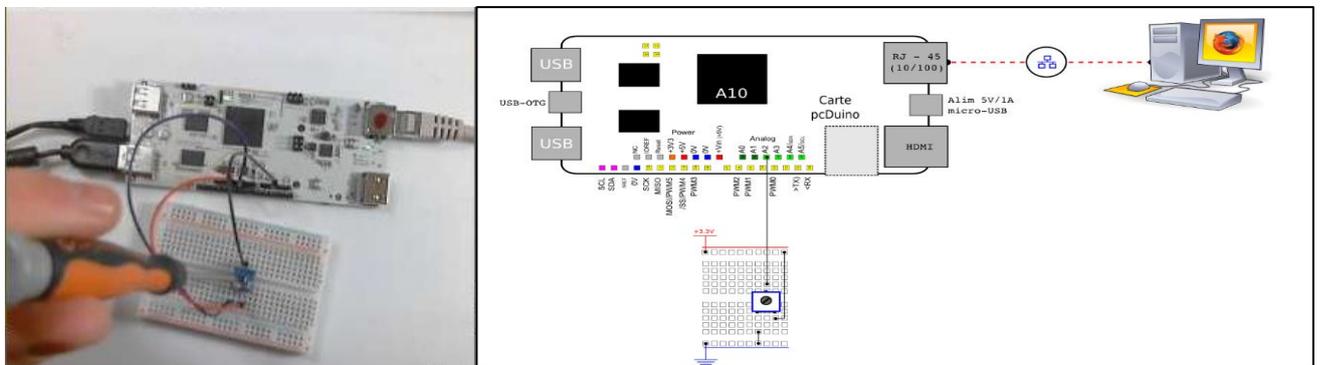


Figure 1.2 Le montage de réalisation

Le code est écrit en Python qui intègre lui-même la page HTML qui contient un code JavaScript qui sera exécuté côté client, celui-ci gère l'envoi de la requête Ajax et la gestion de la présentation de la réponse sous la forme graphique [8].

Le serveur TCP gère les requêtes GET et les requêtes Ajax à partir des clients html.

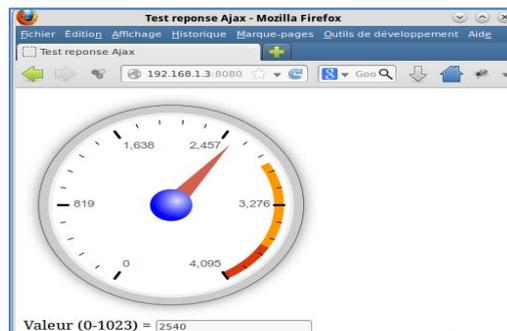


Figure 1.3 la mesure analogique obtenue par la réponse de la requête Ajax

3.4.2 Arduino

Exemple d'une application Android pour communiquer en bluetooth. Une application de ce genre pourrait permettre de contrôler de l'éclairage ou des moteurs (ex. un robot) au moyen d'un appareil Android, ou encore de lire au moyen d'un appareil Android l'état de capteurs externes branchés à l'Arduino.

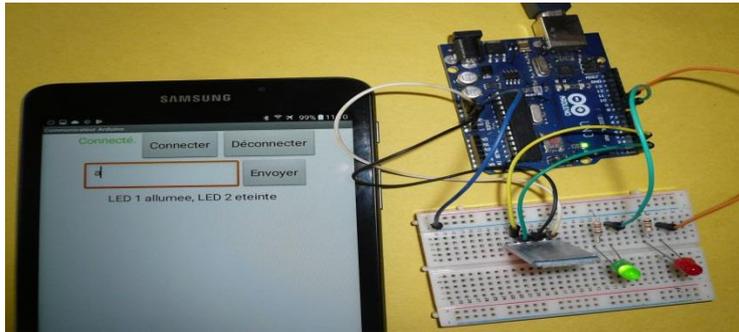


Figure 1.4 Montage de réalisation d'application avec arduino

Il s'agit d'une application très générale, qui se divise en trois parties principales:

1) Dans le haut de la fenêtre, deux boutons permettent d'établir ou d'interrompre la connexion avec l'Arduino. Si aucune connexion bluetooth n'est active, le mot "Déconnecté" apparaît en rouge.

Si une connexion bluetooth est active, le mot "Connecté" apparaît en vert.

2) Un peu plus bas, une zone de texte accompagnée d'un bouton permet à l'utilisateur d'envoyer à l'Arduino un message de son choix

3) Finalement, un label affiche le plus récent message reçu en provenance de l'Arduino. [9]

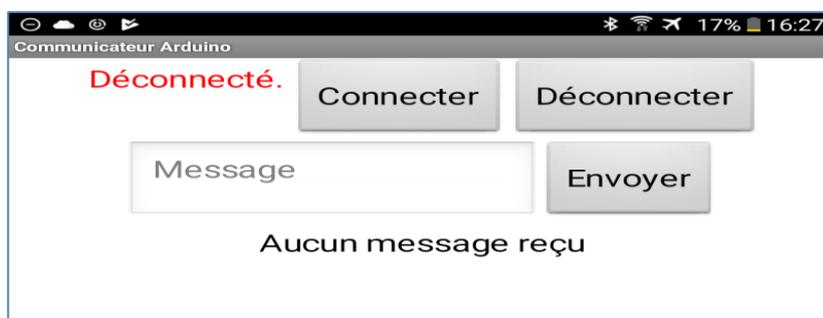


Figure 1.5 L'interface de l'application d'android communiquant en bluetooth

3.4.3 Raspberry

Application pour commander deux DEL (Led) à partir d'un Smartphone avec le Raspberry Pi3 avec Bluetooth et Wifi intégrés.

Le Raspberry pi fait l'objet de serveur, le smartphone est en point d'accès wifi, la communication est établie avec une adresse IP.

Cette première application est simple, deux boutons, rouges au départ, il suffit d'appuyer sur un des deux, il passe au vert et la LED s'allume



Figure 1.6 Les étapes d'allumage des LEDs par Raspberry

Ensuite, on appuie sur un des boutons verts, la led concernée s'éteint

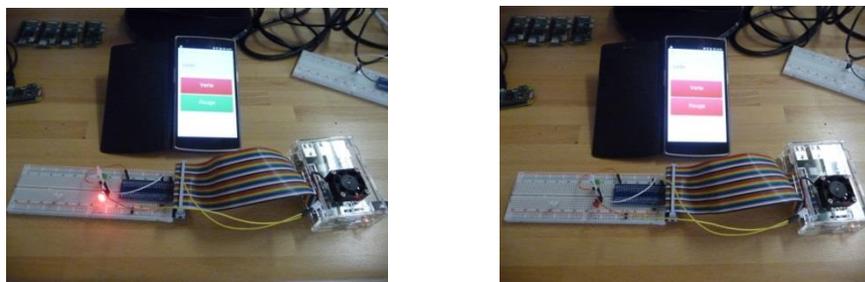


Figure 1.7 les étapes d'extinction des LEDs par Raspberry

La suite sera donc la réalisation de l'application, à partir d'une page de type Web sur laquelle il y a un choix des fonctions qu'elles doivent être agencées sur la page.

Cela nécessite d'avoir une borne Wifi, qui sera dédiée à ce travail et simulera un réseau Internet sans en avoir les défauts (navigation inappropriée, débit trop lent voire accès impossible)

La deuxième étape consiste à mettre en place des capteurs et d'avoir le retour d'information sur le Smartphone [10]

3.4.4 Red Pitaya

L'objectif principal de cet exemple est de permettre aux utilisateurs de pouvoir effectuer des mesures sur les circuits connectés au Red Pitaya. Pour ce faire, le Red Pitaya serait connecté à Internet, ce qui permettrait aux utilisateurs d'accéder au Red Pitaya n'importe où, tant qu'ils sont connectés à Internet.

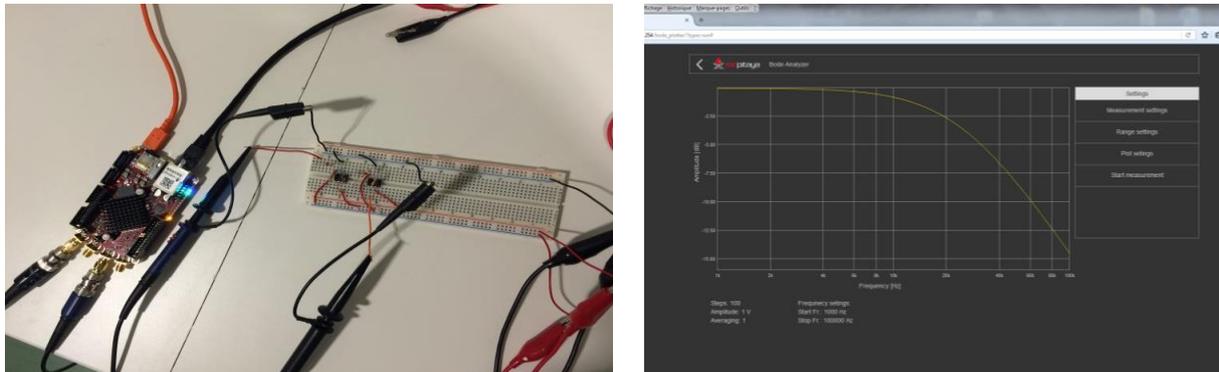


Figure 1.8 Montage de l'application avec red pitaya, et traceur de bode (circuit RC)

En ce qui concerne les circuits à connecter au Red Pitaya, il est fourni avec un circuit de filtrage passe-bande et un circuit de filtrage coupe de bande, comme indiqué dans les illustrations ci-dessous. Ici, on illustre la conception des fréquences de coupure définies avec les valeurs de résistance obtenue. Sur la photo, il montre également la réponse en fréquence qui indique les plages de fréquences qui pourront passer à travers les filtres, ainsi que le gain [11].

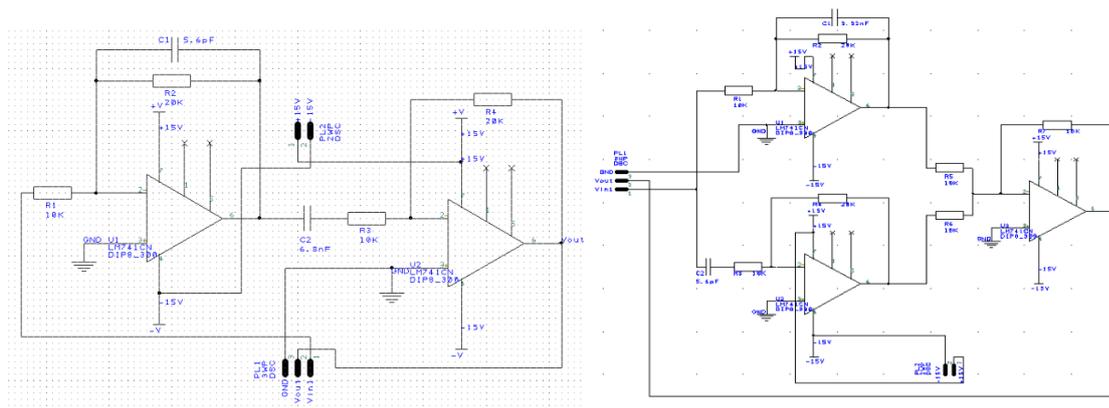


Figure 1.9 les circuits connectés au red pitaya

4 Conclusion

Dans ce chapitre, nous avons présenté les aspects généraux liés aux systèmes embarqués et l'internet des objets avec leurs avantages ensuite nous avons mentionné quelques domaines d'application des systèmes embarqués nous avons aussi donné des exemples des systèmes embarqués

Dans le chapitre suivant, nous allons procéder à la conception de notre application ou nous allons détailler les différentes étapes de réalisation.

Chapitre II

1. Introduction

Les outils de commutation ouvrent une large gamme de besoin, Dans notre application la matrice de commutation est utilisée pour effectuer un choix entre les dispositifs connectés afin de reconfigurer le circuit d'étude.

L'objectif de notre application qui utilise la matrice de commutation consiste à faire le choix entre deux Arduino qui font une transmission série, puis de déterminer les paramètres de cette transmission série à savoir le décodage binaire de la trame envoyée et de mesurer la vitesse de transmission, à travers un oscilloscope embarqué sur la carte Redpitaya.

1. Matrice de commutation

La matrice de commutation à relais (switches) est une carte contrôlant le raccordement des composants aux composants, des instruments et des modules. Il agit comme un outil de câblage de circuits, conçu pour les expériences à base de circuits électroniques. La matrice de commutation contient un certain nombre de relais (switches) : 4, 8 16, ... suivant la complexité de montage.

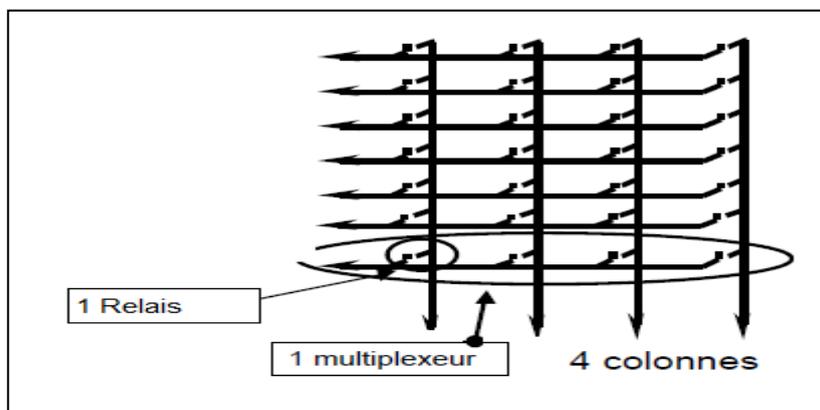


Figure 2.1. Schéma d'une matrice de commutation

2.1 Switch numérique

Un commutateur est un dispositif utilisé pour établir et rompre la connexion dans un circuit électrique. Un commutateur peut également être appelé un relais. Bien que ces deux termes soient souvent utilisés de manière interchangeable, il est généralement fait référence

à un commutateur individuel en tant que relais et un commutateur est une combinaison de relais.

Un commutateur a au moins deux terminaux: un pour le signal entrant et un pour le signal sortant. Les pôles d'un commutateur sont les bornes d'entrée; ceux-ci définissent le nombre de circuits distincts que le commutateur peut contrôler. Les jets d'un commutateur sont les bornes de sortie; ils définissent le nombre de connexions de sortie différentes auxquelles chaque pôle de commutateur peut connecter son entrée. Un commutateur est classé par son nombre de lancers et de pôles [12].

2.1.1 Les type de switch

- **ON-OFF Single Pole, Single Throw = SPST**

Les commutateurs normalement ouverts (N.O.) ferment le circuit quand ils sont activés

Les commutateurs normalement fermés (N.C.) ouvrent le circuit lorsqu'ils sont activés.

Quand des switches deviennent plus complexes, ils peuvent ouvrir une connexion et en fermer une autre lorsqu'ils sont activés.

Ce type d'interrupteur est un interrupteur unipolaire à double orientation (SPDT)

- **ON-ON Single Pole, Double Throw = SPDT**

Cet interrupteur peut être activé dans les deux positions, allumant un appareil séparé dans chaque cas. On l'appelle souvent un commutateur inverseur..[13]

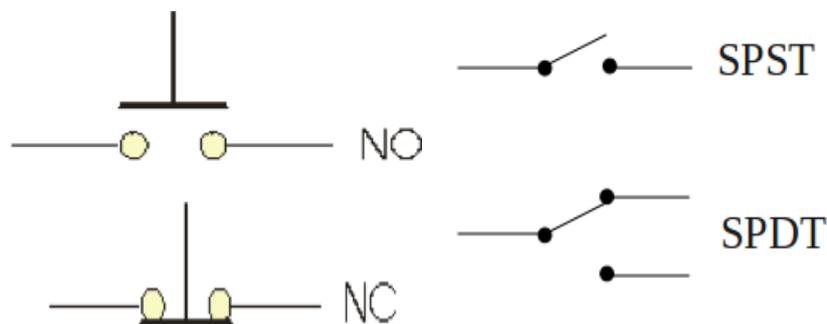


Figure 2.2.Les types de switch

2.2 Relais

Un relais est un appareil dans lequel un phénomène électrique (courant ou tension) contrôle la commutation On / Off d'un élément mécanique (on se trouve alors en présence d'un relais électromécanique) ou d'un élément électronique (on a alors affaire à un relais statique). C'est en quelque sorte un interrupteur que l'on peut actionner à distance, et où la fonction de coupure est dissociée de la fonction de commande.

2.2.1 Relais électromécaniques

Un relais électromécanique est doté d'un bobinage en guise d'organe de commande. La tension appliquée à ce bobinage va créer un courant, ce courant produisant un champ électromagnétique à l'extrémité de la bobine (il ne s'agit ni plus ni moins que d'un électro-aimant). Ce champ magnétique va être capable de faire déplacer un élément mécanique métallique monté sur un axe mobile, qui déplacera alors des contacts mécaniques [14].

2.3 Rôle de la matrice de commutation

Le rôle de la matrice de commutation c'est de choisir une configuration et personnaliser le circuit d'étude.

Nous avons choisi de concevoir la matrice de commutation à base d'un Switch numérique(ADG202A), qui dispose de quatre (04) commutateurs numériques.

2.3.1 Switch ADG202

ADG201A et ADGG202A sont des dispositifs CMOS monolithiques comprenant quatre commutateurs sélectionnables indépendamment. Ils sont conçus sur un LC2MOS processus amélioré qui donne une capacité de traitement du signal accrue de ± 15 V.

Le ADG201A et ADGG202A se composent de quatre commutateurs SPST¹. Ces commutateurs offrent également des vitesses de commutation élevées et une faible R_{on} .

Ils ne diffèrent que par le fait que la logique de commande numérique est inversée. Tous les appareils présentent une coupure avant d'effectuer une action de commutation. Inhérente à la conception est une injection à faible charge pour des transitoires minimum lors de la commutation des entrées numériques [15].

¹http://resources.uwcsea.edu.sg/UWCSEA_DT_East/Electronics/Electronics/Switches.html

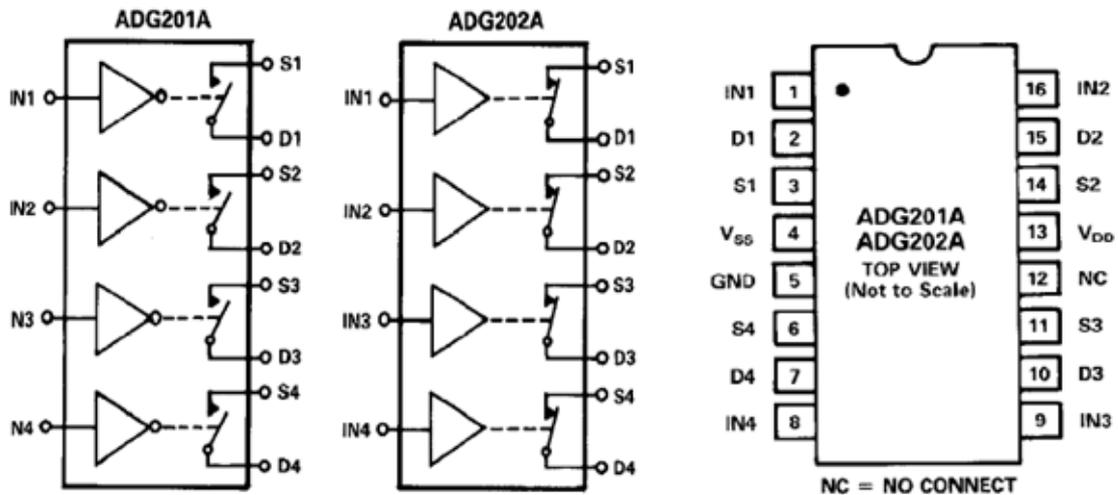


Figure 2.2 Configuration des pins d'ADG202A

2.4 La commande de la matrice de commutation

La commande de la matrice de commutation se fait par un PcDuino qui est un système embarqué accessible à distance. En outre, c'est un ordinateur de type carte qui fonctionne sous le système Linux ubuntu. L'éditeur d'Arduino permet d'écrire des programmes qui manipulent les entrées/sorties (analogique/numériques) et de les exécuter directement sur la même carte du PcDuino.

3. Architecture Hard

L'architecture matérielle est représentée par la figure suivante :

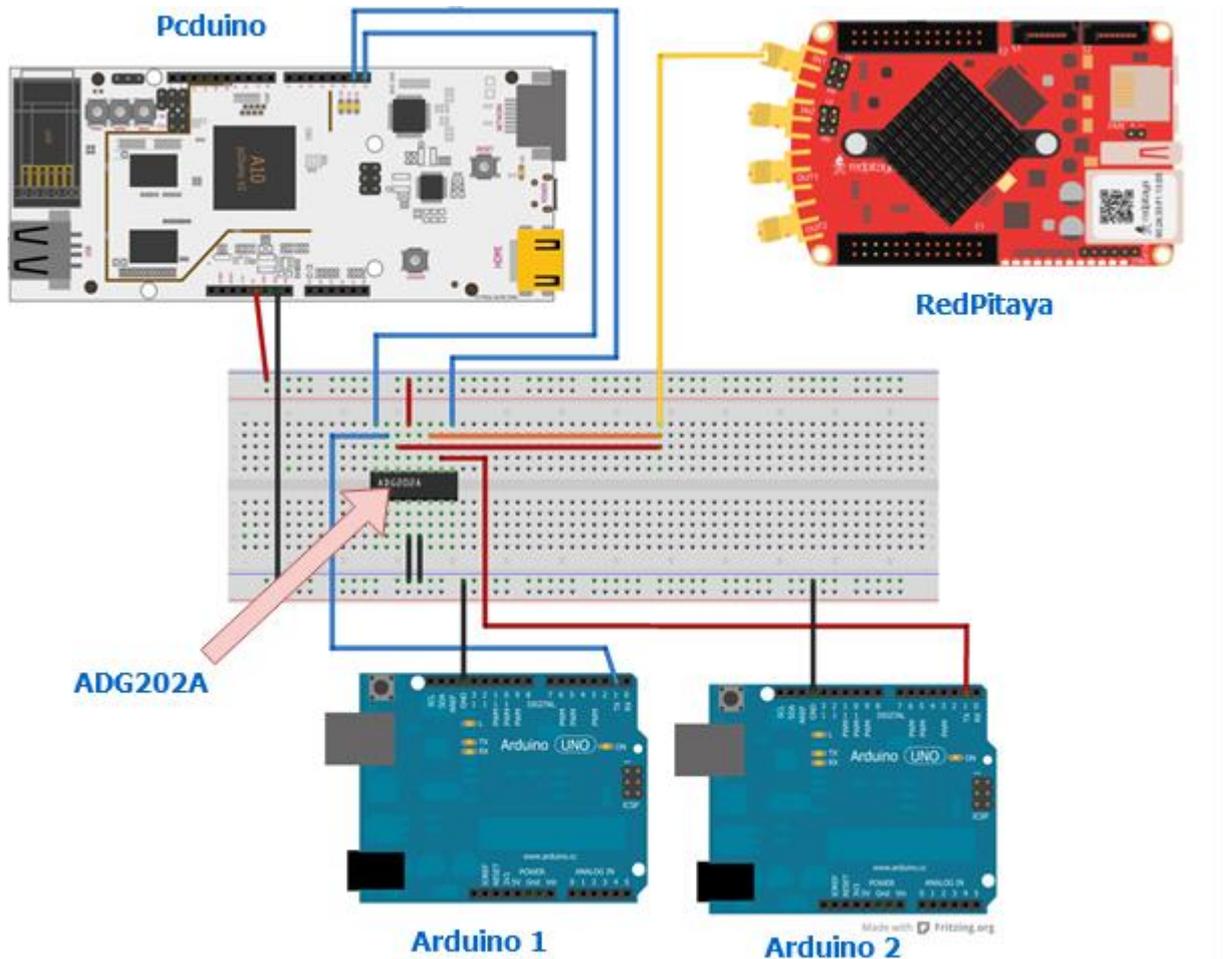


Figure 2.3 Schéma de Montage

4. Architecture logicielle (software)

L'architecture logicielle est donnée par la figure suivante :

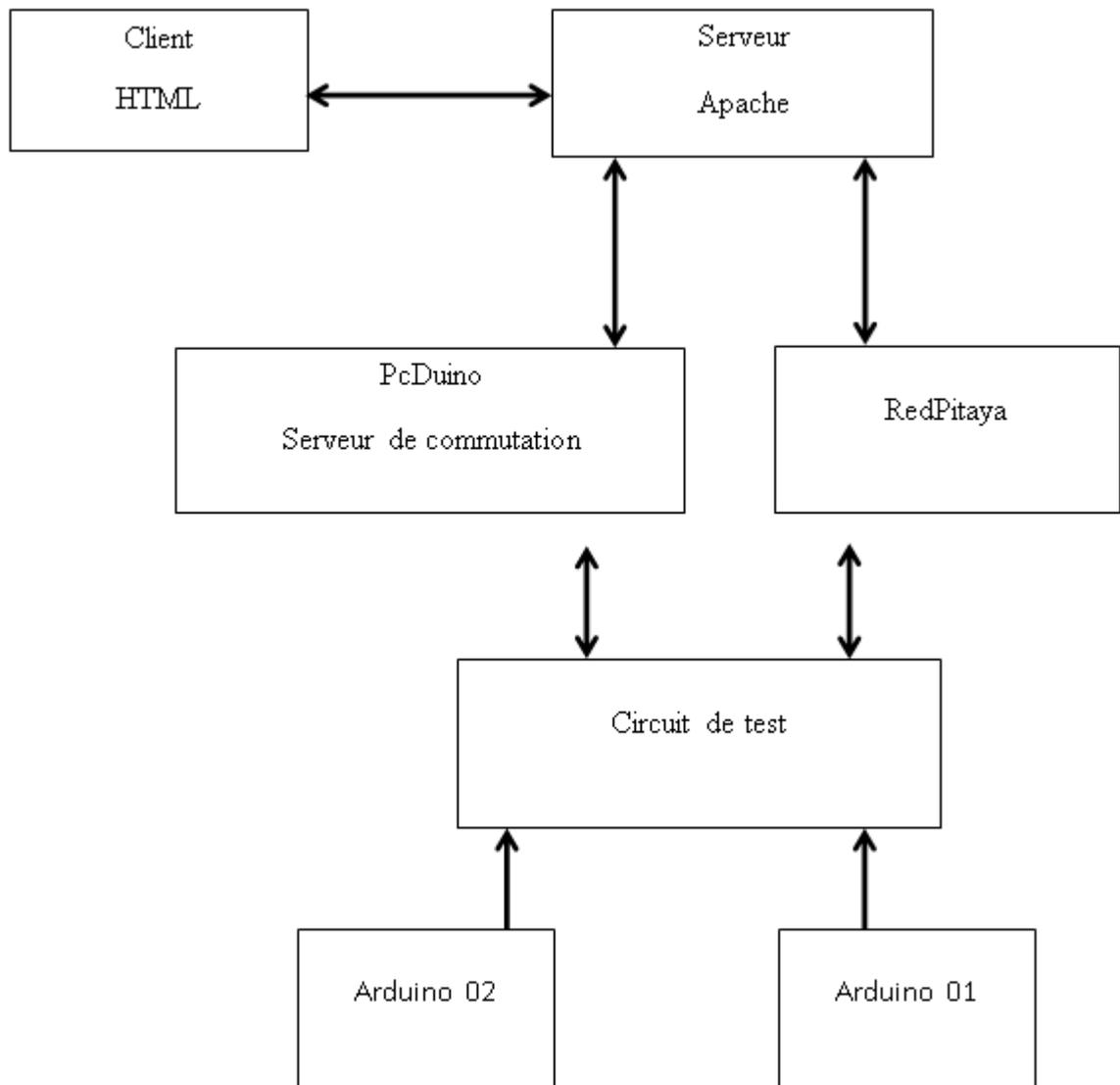


Figure 2.4 Architecture Logicielle

5. Outil de développement Hard

5.1 PcDuino

Le pcDuino, est un mini-PC très performante à faible coût qui tient sur une mini-carte électronique [14]. Elle supporte les systèmes d'exploitation tels qu'Ubuntu Linux et Android. Le pcDuino3 dispose également d'une interface HDMI pour un affichage sur tout type d'écran supportant l'HDMI. Elle supporte le multi-format 1080p avec un décodeur 60 image/seconde et 1080p 30 images par seconde en H.264 et encodage vidéo MPEG4 à l'aide d'un processeur vidéo intégré.

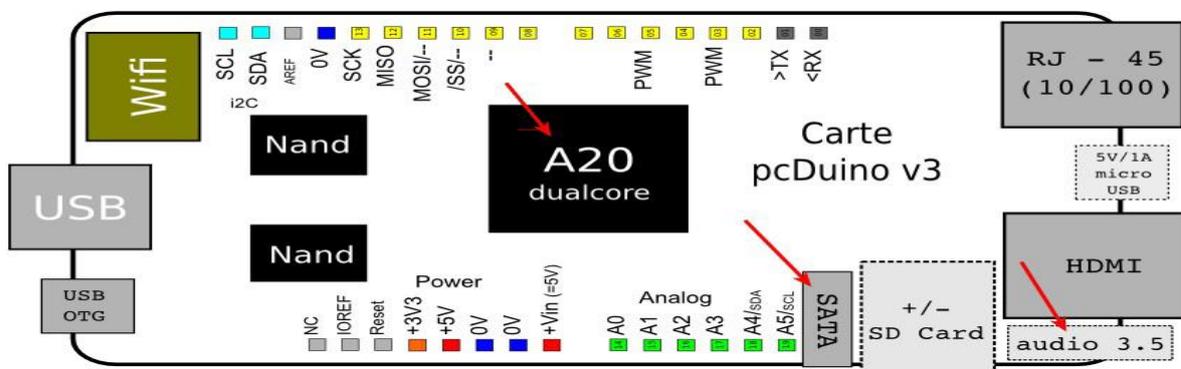


Figure 2.5 Architecture interne d'un pcDuino

PcDuino dispose d'une suite d'outils facile à utiliser et compatible avec le système Arduino comme la compatibilité électronique pin à pin avec les shields Arduino.

Le pcDuino v3 se distingue par les caractéristiques suivantes :

- Une connectique « Arduino-like » sur borines droites femelles associant :
 - 14 broches E/S digital
 - 6 broches analogiques dont 4 en 12 bits
 - 2 broches PWM
 - Communication 1xUART, 1xI2C, 1xSPI...

Pour lancer un pcDuino, on a besoin des composants suivants :

- d'une alimentation 5V/1000mA micro-USB
- d'un câble HDMI et d'un écran avec entrée HDMI
- d'une souris USB, clavier USB et d'un Hub USB. [16]



Figure 2.6 Composants nécessaires pour lancer un pcDuino

5.2 Arduino

Une carte Arduino est une petite (5,33 x 6,85 cm) carte électronique équipée d'un Microcontrôleur.

Le microcontrôleur permet, à partir d'évènements détectés par des capteurs, de programmer et commander des actionneurs ; la carte Arduino est donc une interface programmable.

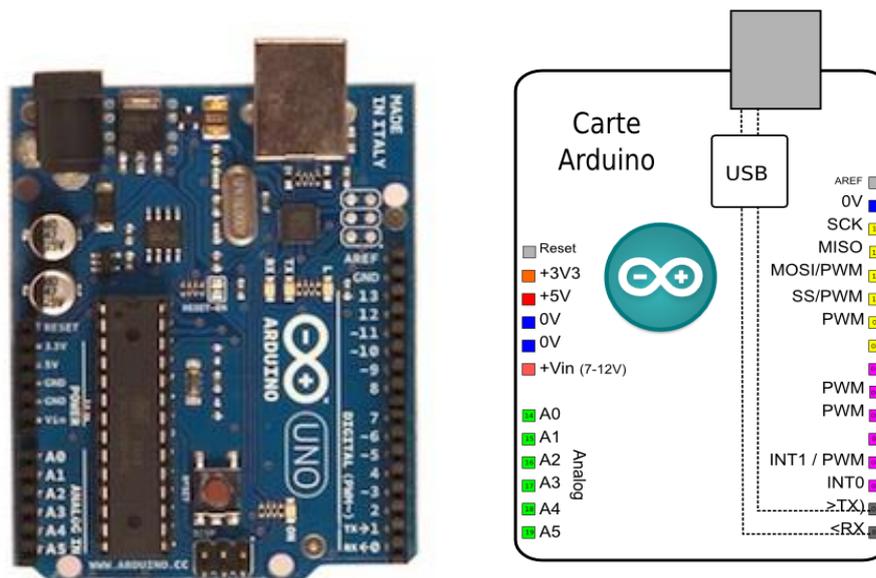


Figure 2.7 Architecture interne d'Arduino

5.3 Communication de arduino

La carte Arduino Uno dispose de toute une série de facilités pour communiquer avec un ordinateur, une autre carte Arduino, ou avec d'autres microcontrôleurs.

L'ATmega 328 dispose d'une UART (UniversalAsynchronousReceiverTransmitter ou émetteur-récepteur asynchrone universel en français) pour communication série de niveau TTL (5V) et qui est disponible sur les broches 0 (RX) et 1 (TX) [17].

L'UART c'est le composant utilisé pour faire la liaison entre l'ordinateur et le port série. L'ordinateur envoie les données en parallèle (autant de fils que de bits de données). Il faut par conséquent transformer ces données pour les faire passer à travers une liaison série qui

utilise un même fil pour faire passer l'ensemble des bits de données [18].

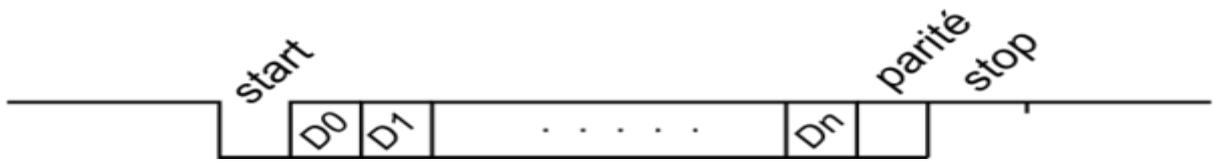


Figure 2.8 Composants de lame pour une transmission série

- Bit de Start toujours à 0 (dominant) : permettant de la synchronisation du récepteur
- Les données (D_0, D_1, \dots, D_n): la taille peut fluctuer (généralement entre 5 et 9 bits)
- Bit de parité : peut-être un bit de parité paire ou impaire
- Bit de stop toujours à 1 (état au repos).

5.4 RedPitaya

RedPitaya est une plateforme de mesure open source, petite, compacte, simple, intuitive, puissante, polyvalente, configurable, compatible avec des applications prêtes à l'emploi comme *Matlab* et *LabView* ou développées soi-même en utilisant des langage de programmation comme C, javascript et Python.

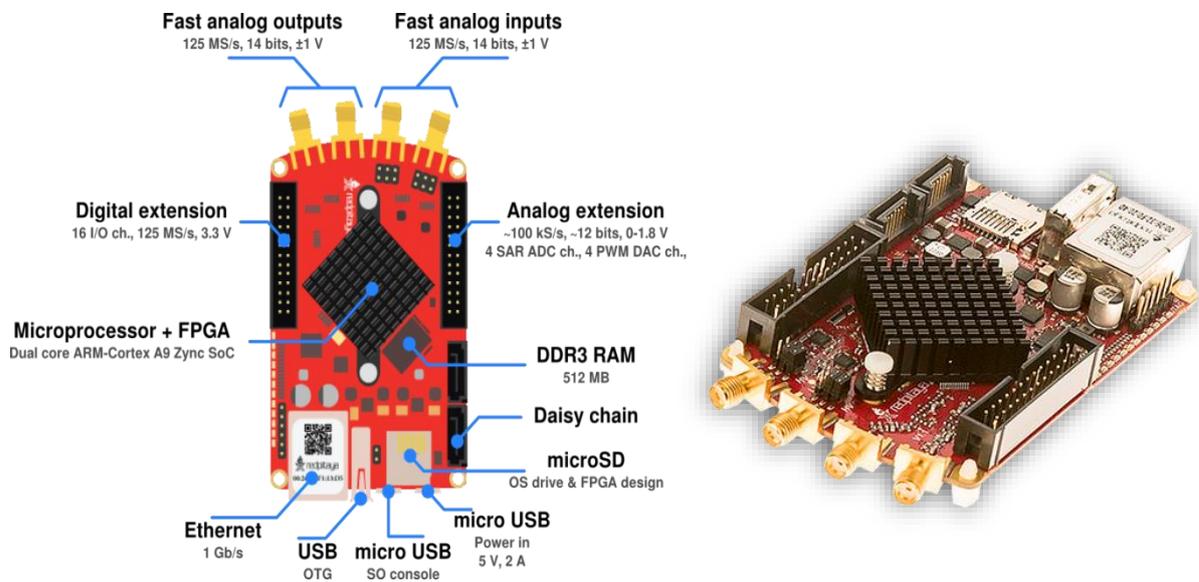


Figure 2.9 Redpitaya

Mise en service

La mise en service est aisée grâce à l'excellent guide de démarrage du site RedPitaya². On copie l'image du système pour le RedPitaya sur une carte microSD que l'on l'insère ensuite dans le connecteur SD du module. Il suffit alors de relier le Pitaya à un réseau LAN et de le mettre sous tension pour inaugurer le premier amorçage.

Suivre le processus d'amorçage avec le port COM USB d'un PC permet de voir l'adresse IP affectée au module Pitaya. L'amorçage terminé, on peut communiquer directement avec Linux via le port COM USB ou, méthode préférée, passer par Ethernet pour établir d'autres communications, en utilisant le protocole HTTP (interface web) ou SSH (client ssh) [19].

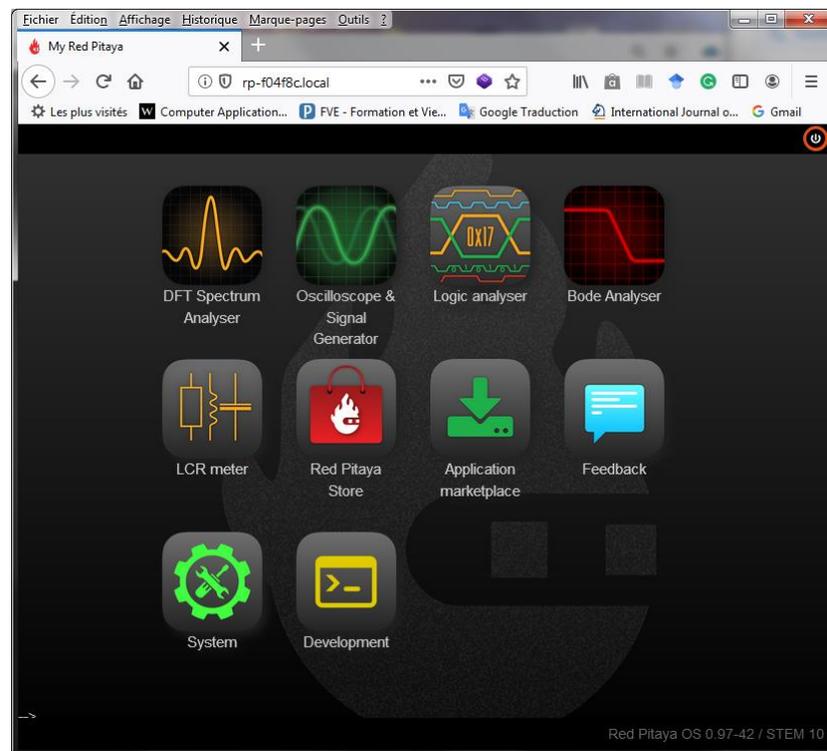


Figure 2.10 Interface web de Redpitaya avec icônes des applications

²<https://redpitaya.readthedocs.io/en/latest/>

6. Software : Architecture logicielle

L'architecture logicielle se compose de différentes parties, à savoir le back end, le front end et le système de communication entre les deux.

Le développement du back end consiste à développer le serveur embarqué qui est chargé d'écouter les commandes en provenance du front-end qui est l'interface utilisateur..

6.1 Front end

Le développement de l'interface utilisateur se fait en utilisant des outils web à base de html, javascript et CSS.

6.1.1 JAVA Script

JavaScript est un langage de script orienté-objet utilisé pour le développement d'applications internet afin de rendre ce langage accessible à des navigateurs.

Les programmes JavaScript s'intègrent dans le code HTML d'une page web. L'intérêt d'un langage comme JavaScript est de pouvoir contrôler dynamiquement le comportement d'une page web : on peut par exemple dans notre cas vérifier la page web qui effectue le choix d'Arduino et qui vérifie aussi si le code transmis par cet Arduino est correct.

JavaScript est utilisé côté client, la charge d'exécuter le code JavaScript incombe au client, c'est-à-dire à un navigateur web. Le serveur envoie le code HTML avec du code JavaScript au client qui l'interprète dès qu'il est chargé. Ce type de fonctionnement s'oppose aux langages orientés serveurs qui exécutent le programme pour produire du HTML et l'envoient au client [20].

6.1.2 HTML

HTML signifie « *HyperText MarkupLanguage* » qu'on peut traduire par « langage de balises pour l'hypertexte ». Il est utilisé afin de créer et de représenter le contenu d'une page web et sa structure. D'autres technologies sont utilisées avec HTML pour décrire la présentation d'une page (CSS) et/ou ses fonctionnalités interactives (JavaScript).

HTML fonctionne grâce à des « balises » qui sont insérées au sein d'un texte normal. Chacune de ces balises indique la signification de telle ou telle portion de texte dans le site. On parle d'« hypertexte » en référence aux liens qui connectent les pages web entre elles. C'est la mécanique originelle du « World Wide Web » que nous connaissons aujourd'hui. En écrivant et publiant des pages web, nous devenons un acteur du Web dès que notre site est accessible en ligne.

HTML permet d'inclure des images et d'autres contenus dans les pages web. Chacun peut créer des sites web aussi bien statiques que dynamiques. HTML est le langage qui permet de décrire la structure et le contenu d'un document web. Ce contenu est balisé par des éléments HTML

comme `<head>`, `<title>`, `<body>`, `<header>`, `<footer>`, `<article>`, `<section>`, `<p>`, `<div>`, ``, `` et bien d'autres encore. Ces éléments forment les blocs utilisés pour construire une page web [21].

6.1.3 CSS : feuille de style

Cascading Style Sheets (CSS) est un langage de feuille de style utilisé pour décrire la présentation d'un document écrit en HTML ou en XML (on inclut ici les langages basés sur XML comme SVG ou XHTML). CSS décrit la façon dont les éléments doivent être affichés à l'écran, sur du papier, en vocalisation, ou sur d'autres supports.

```
body {  
background-color: powderblue;  
}  
h1 {  
color: blue;  
}  
p {  
color: red;  
}
```

6.2 Back end

6.2.1 Node.js

Node.js est une plateforme de développement JavaScript. Ce n'est pas un serveur, ce n'est pas un framework, c'est juste le langage Javascript avec des bibliothèques permettant de réaliser des actions comme écrire sur la sortie standard, ouvrir/fermer des connections réseau ou encore créer un fichier.

Il est souvent confondu avec un serveur car c'est son origine : Node.js a été créé par Ryan Dahl dans le but de pouvoir créer des applications temps réel où le serveur est capable de pousser de l'information au client. C'est dans ce but qu'il utilise la bibliothèque libuv pour réaliser son modèle d'entrée sortie non bloquante [22]

6.3 Communication entre pcDuino et Windows

6.3.1 WinSC

WinSCP est un client SFTP (Secure FTP) graphique pour Windows. Il est open source et utilise le protocole SSH et le protocole SCP est également supporté. Le but de ce programme est de permettre la copie sécurisée de fichiers entre un ordinateur local et un ordinateur distant [23].

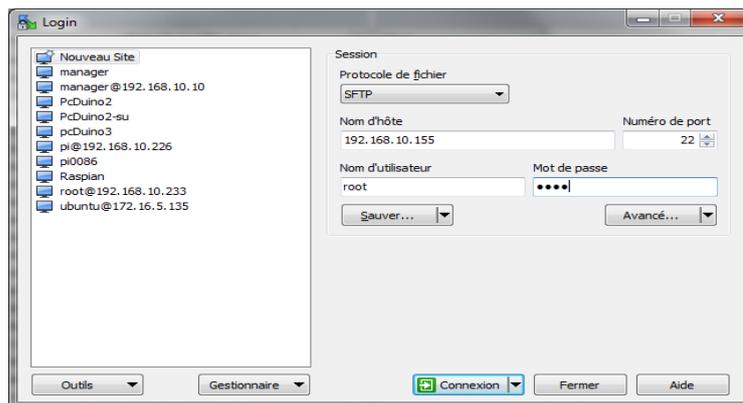


Figure 2.12 La configuration de WinSCP

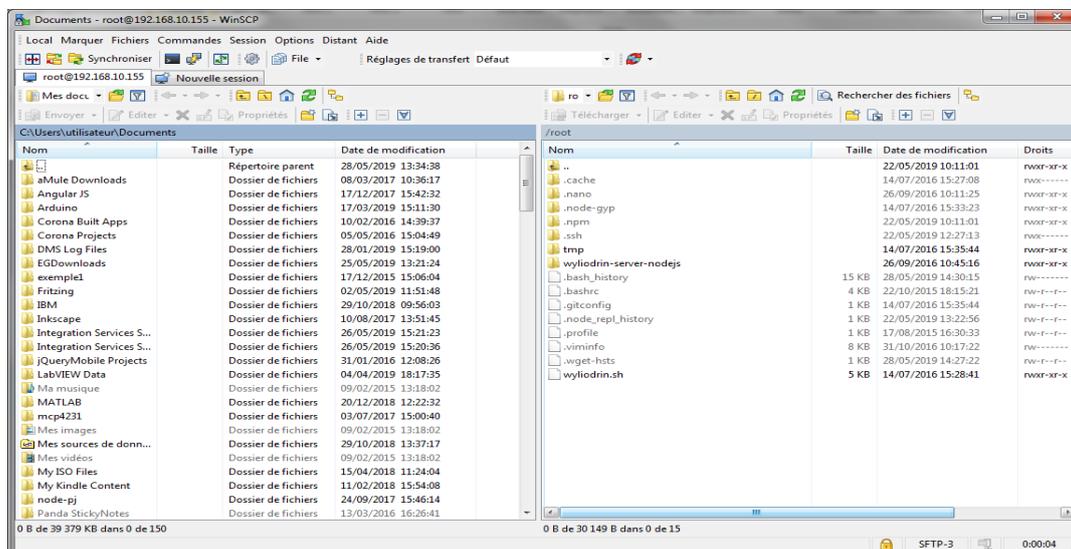


Figure 2.11 les fichiers de pcDuino et d'ordinateur

6.3.2 PuTTY

PuTTY est un petit logiciel développé à l'origine par Simon Tatham pour Windows qui permet d'effectuer des connexions SSH. Il s'agit d'un protocole de communication sur réseau TCP/IP (port 22). Peut sécuriser, les informations sont communiquées en clair (les mots de passe sont visible via des logiciels d'analyse de réseaux) [24].

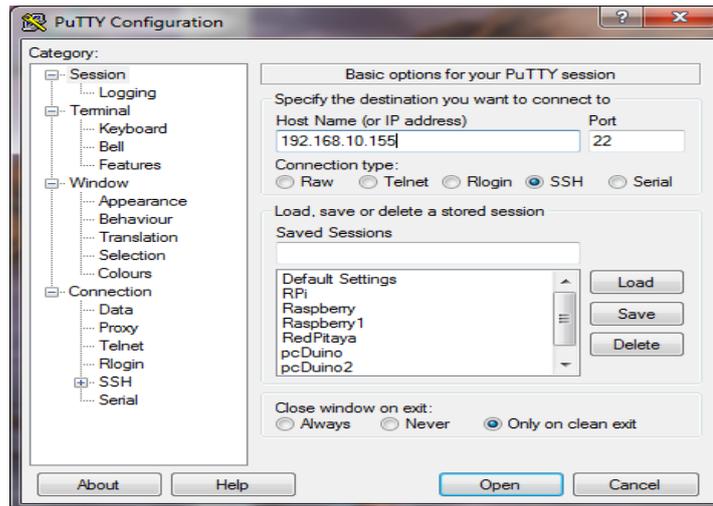


Figure 2.13 La configuration de PuTTY

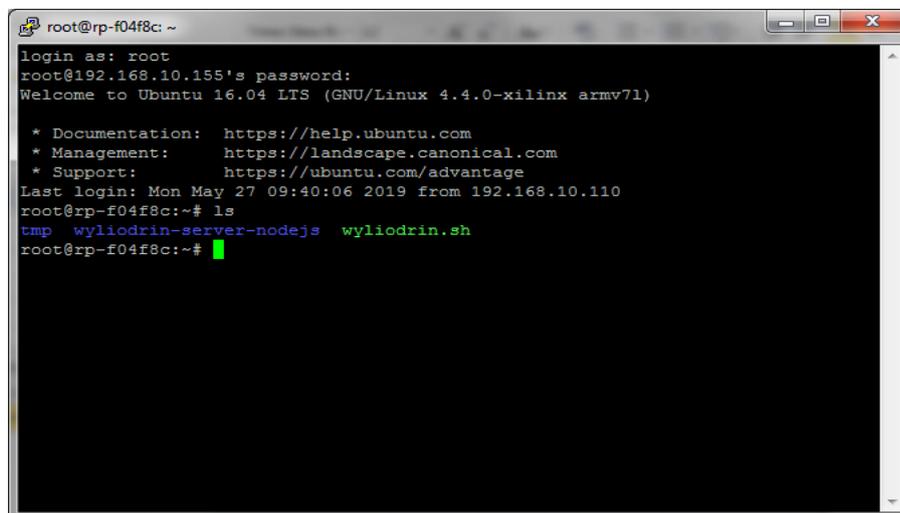


Figure 2.14 . Le terminal de PuTTY

7. Conclusion

Dans ce chapitre nous avons détaillé et expliqué les différents aspects liés à l'architecture matérielle et logiciel nécessaire pour la conception de notre application.

Dans le troisième chapitre nous allons détailler notre travail et les différentes étapes pour réaliser notre application, et les démarches de mise en œuvre et de manipulation.

Chapitre III

1. Introduction :

Dans ce chapitre nous abordons une description générale de notre application ou nous allons identifier toutes les étapes de réalisation

Notre application présente un TP de communication série dont le but de faire transmettre une trame binaire

2. Objectif

L'objectif de notre TP c'est de décoder la trame transmis en binaire de l'Arduino vers le Red pitaya et de calculer la vitesse de transmission, c'est-à-dire déterminer les paramètres de transmission série asynchrone.

3. Mode opératoire

3.1 Première patrie : L'interface de pcduino

Dans la première étape nous allons connecter le pcduino à un ordinateur pour afficher ses fichiers en utilisant les logiciels PuTTY¹ et WinSCP²

Les deux composants doivent être connectés au même réseau

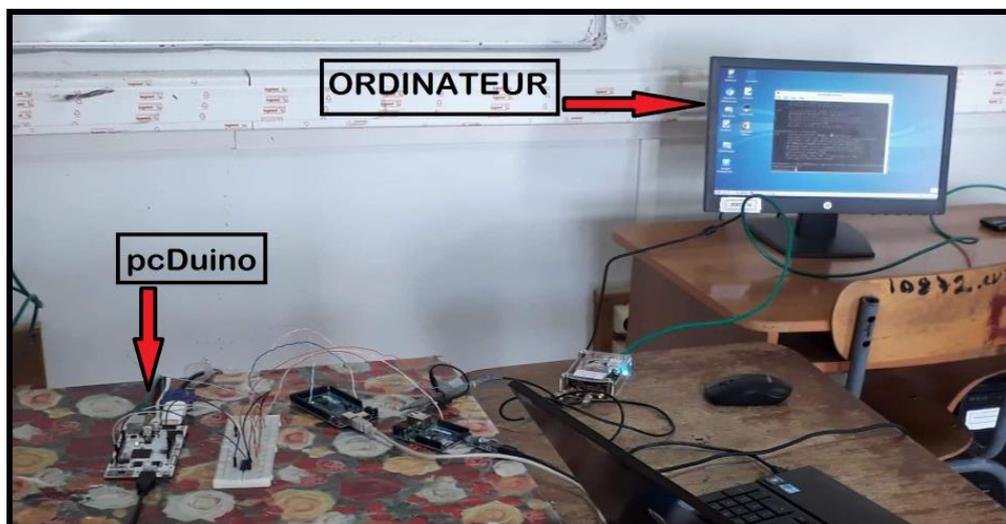


Figure 3-1. Interface de pcduino dans un ordinateur

¹ <https://www.putty.org/>

² <https://winscp.net/>

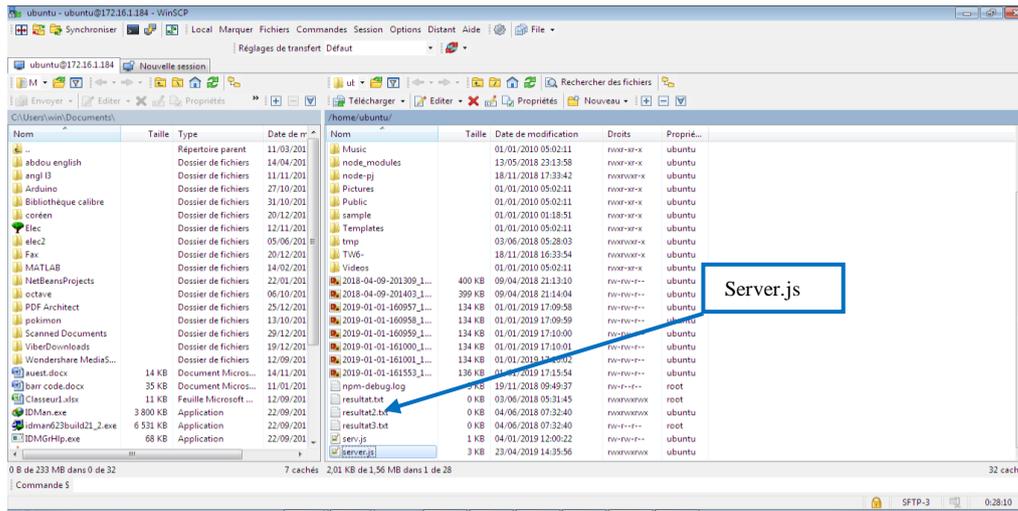


Figure 3-2. Les fichiers de pcduino affichés sous WinScp

Nous allons créer un serveur de commutation node.js dans le pcduino qui commande les GPIO de ce pcduino.

```

1  var etat ;
2  var s1,s2,s3,s4=0;
3  var http = require('http');
4  var url = require('url');
5  var fs = require('fs');
6  var querystring = require('querystring');
7  http.createServer(function (req, res) {
8    res.writeHead(200, {'Content-Type': 'text/plain'});
9    var params = querystring.parse(url.parse(req.url).query);
10   s1=params['x1'];
11   s2=params['x2'];
12   s3=params['x3'];
13   s4=params['x4'];
14   ledonoff();
15   res.end('s1 s2 s3 s4 \n'+s1+s2+s3+s4);
16 }) .listen(8080);
17 console.log(" démarrage du serveur sur le port 8080");
18 //les fonctions ON/OFF GPIO :
19 function ledonoff() {
20   fs.writeFile('/sys/devices/virtual/misc/gpio/mode/gpio1', '1', function (err) {
21     if(err) throw err;
22     // console.log('Le switch est ouvert');
23   });
24   fs.writeFile('/sys/devices/virtual/misc/gpio/mode/gpio2', '1', function (err) {
25     if(err) throw err;
26     // console.log('Le switch est ouvert');
27   });
28   fs.writeFile('/sys/devices/virtual/misc/gpio/pin/gpio1', s1, function (err) {
29     // if(err) throw err;
30     console.log('Le gpio1 est en etat : '+s1);
31   });
32   fs.writeFile('/sys/devices/virtual/misc/gpio/pin/gpio2', s2, function (err) {
33     // if(err) throw err;
34     console.log('Le gpio2 est en etat : '+s2);
35   });
36   console.log('*****');
37 }
38 ;

```

Figure 3-3. Le code source du serveur de commutation

3.2 Deuxième partie : L'interface de commutation

Cette étape consiste à développer une interface web pour gérer les différentes fonctions et commander à distance la matrice de commutation de switch (ADG202A)

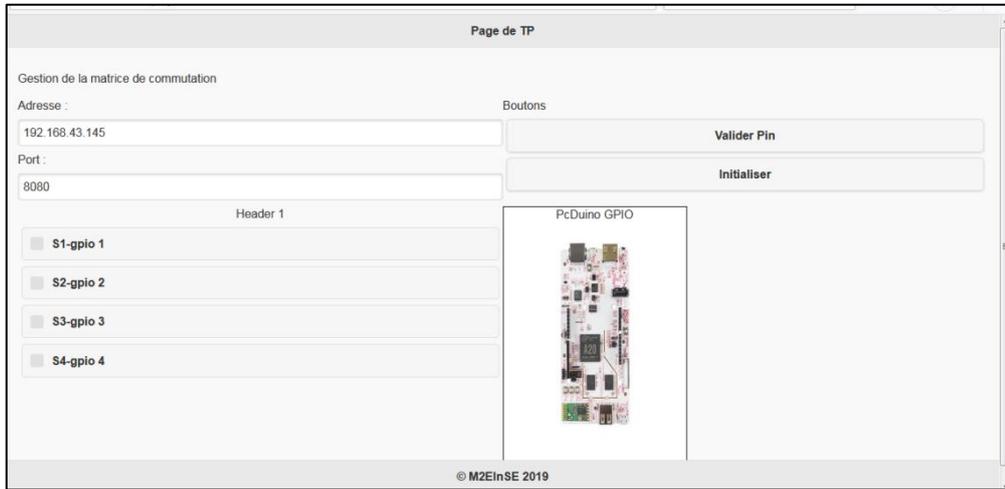


Figure 3-4. L'interface qui commande la matrice de commutation

Cette interface web est développée à l'aide d'un langage JavaScript

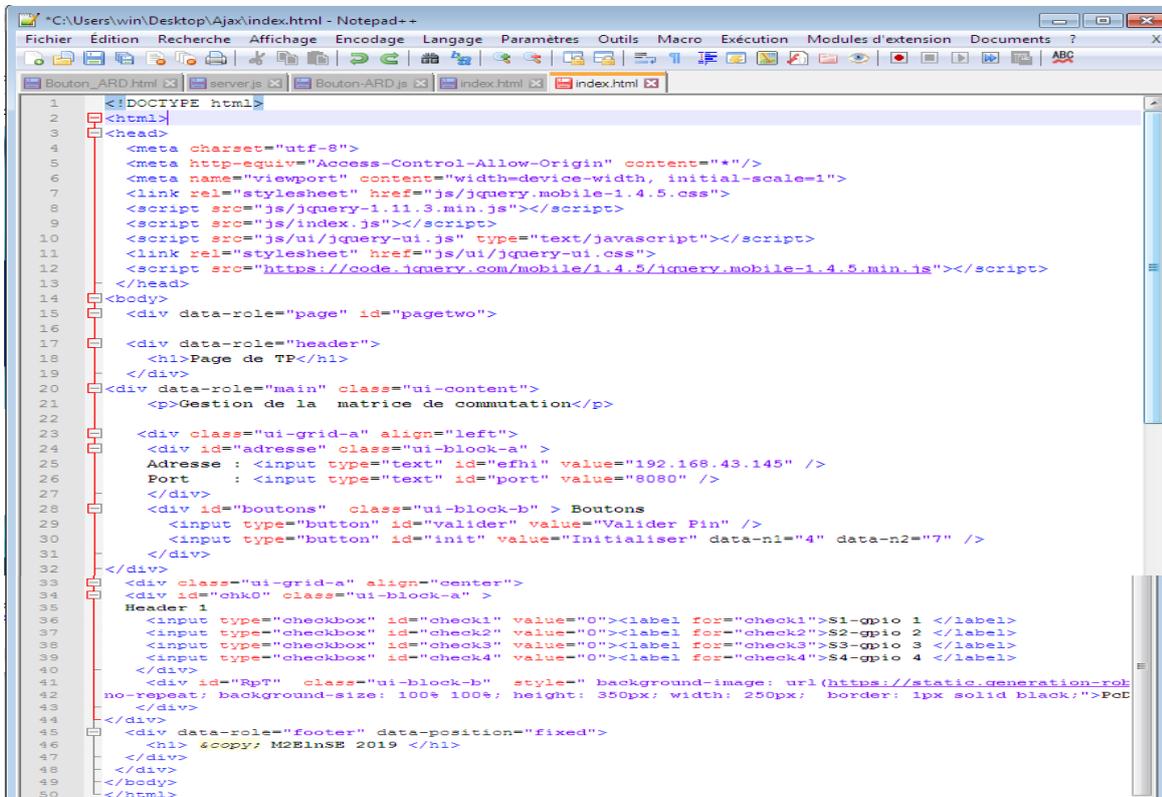


Figure 3-5. Le programme qui génère l'interface de commutation

3.3 Troisième partie : Le choix entre les arduinos

La matrice de commutation (ADG202A) est liée à deux Arduino

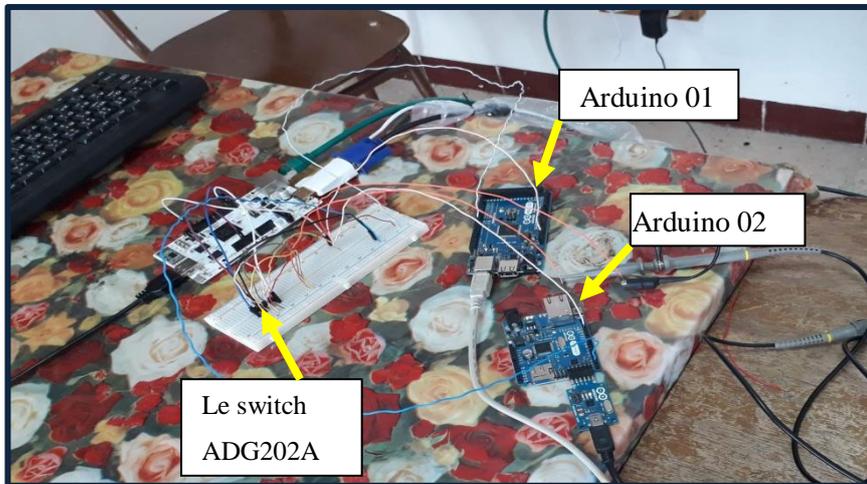


Figure 3-6. Le composant du montage

Le choix entre Arduino 01 et Arduino 02 se fait à l'aide du programme suivant

```

1  var r1 = r2 = r3 = r4 = 0;
2  var provider='',if1="25";
3  var ipfree;
4  var etat_s;
5  var arduino=0,car='',vitesse=0;
6  function ajaxFunc(){
7  $.ajax({
8      url:provider+etat_s,
9      //url:'http://192.168.10.226:8080/?'+etat_s,
10     success: function(response){
11         $('#dim1').html(response); //affiche l'etat des pins sur Rpi
12     },
13     type: 'GET'
14 });
15 }
16 $(document).ready(function(){
17     $('#ard1').click(function(){
18         arduino=1;
19         provider='http://'+$('#efh1').val()+":"+$('#port').val()+"/?";
20         r1=r2=0;r3=0;r4=0;
21         etat_s = "r1="+r1+"&r2="+r2+"&r3="+r3+"&r4="+r4;
22         ajaxFunc(); // requete PcDuino
23     });
24     $('#ard2').click(function(){
25         arduino=2;
26         provider='http://'+$('#efh1').val()+":"+$('#port').val()+"/?";
27         r1=0;r2=1;r3=0;r4=0;
28         etat_s = "r1="+r1+"&r2="+r2+"&r3="+r3+"&r4="+r4;
29         ajaxFunc(); // requete PcDuino
30     });
31     $('#valider').click(function(){
32         car=$('#car').val();
33         vitesse=$('#vitesse').val();
34         if(arduino==1) {
35             if(car=='01100001' && vitesse==5600) alert('Reponse correcte');
36             else alert('Reponse fausse');
37         }
38         else if(car=='01100011' && vitesse==115200) alert('Reponse correcte');
39         else alert('Reponse fausse');
40         //alert('arduino = '+arduino);
41     });
42     $('#check1').change(function(){
43         if ($(this).is(':checked')) { r1=1; } else { r1=0; }
44         $('#Connecter ARDUINO 01').button('enable');
45         $('#Connecter ARDUINO 02').button('enable');
46     });
47     $('#check2').change(function(){
48         if ($(this).is(':checked')) { r2=1; } else { r2=0; }
49         $('#Connecter ARDUINO 01').button('enable');
50         $('#Connecter ARDUINO 02').button('enable');
51     });
52 });

```

Figure 3-7. Le programme du choix des arduinos

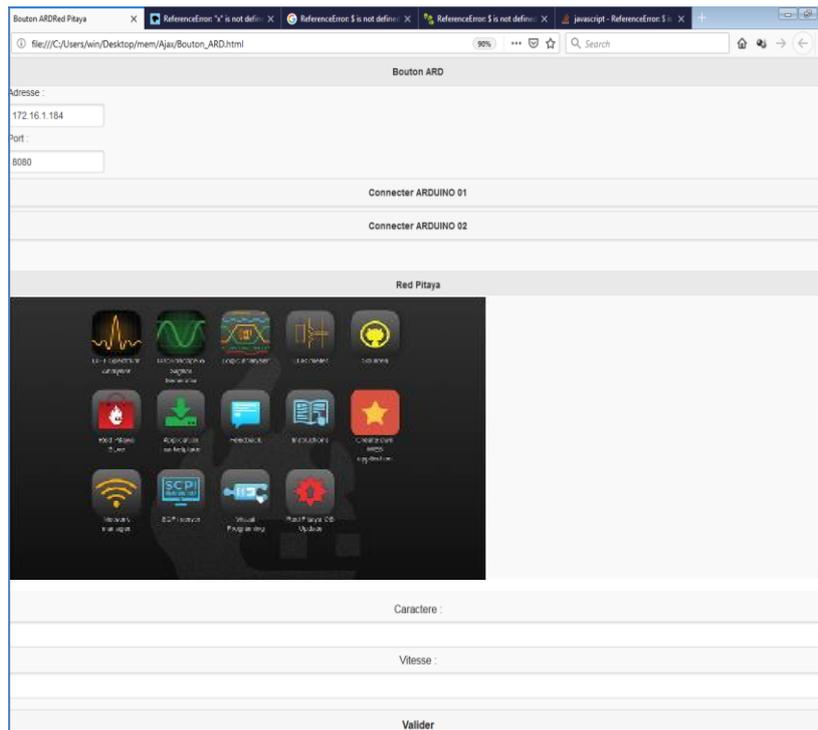


Figure 3-8. L’interface web de manipulation

Cette interface web est développée à l’aide du langage JavaScript

```

1 <!DOCTYPE html>
2 <html>
3 <head>
4 <meta charset="utf-8">
5 <title>Bouton-ARD</title>
6 <meta http-equiv="Access-Control-Allow-Origin" content="*" />
7 <meta name="viewport" content="width=device-width, initial-scale=1">
8
9 <link rel="stylesheet" href="js/jquery.mobile-1.4.5.css">
10 <script type="text/javascript" src="//ajax.googleapis.com/ajax/libs/jquery/2.0.0/jquery.min.js"></script>
11 <script src="js/jquery-1.11.3.min.js"></script>
12 <script src="https://code.jquery.com/jquery-1.11.3.min.js"></script>
13 <!--script src="js/ui/jquery-ui.js" type="text/javascript"></script-->
14 <!--link rel="stylesheet" href="js/ui/jquery-ui.css">
15 <script src="https://code.jquery.com/mobile/1.4.5/jquery.mobile-1.4.5.min.js"></script>
16 <script src="js/Bouton-ARD.js"> </script>
17 </head>
18 <div data-role="header">
19 <h1>Bouton ARD</h1>
20 </div>
21
22 <div id="adresse" class="ui-block-a">
23 Adresse : <input type="text" id="efhi" value="172.16.1.184" />
24 Port : <input type="text" id="port" value="8080" />
25 </div>
26
27 <div id="boutons" class="bouton-aligne">
28
29 <button type="button" id="ard1" onclick="toggle_text1();">Connecter ARDUINO 01</button>
30 <span id="span_txt1"></span>
31 <button type="button" id="ard2" onclick="toggle_text2();">Connecter ARDUINO 02</button>
32 <span id="span_txt2"></span>
33 </div>
34
35 <div data-role="header">
36 <h1>Red Pitaya</h1>
37 </div>
38 <iframe id="inlineFrameExample"
39 title="Inline Frame Example"
40 width="900"
41 height="600"
42 src="http://172.16.1.186/scopesgenpro/?type=run">
43 </iframe>
44 <h2> </h2>
45
46 <div class="ui-grid-b" align="center">
47 Caractere :<input type="text" id="car" name="car" required
48 minlength="4" maxlength="8" size="10"/>
49 </div>
50 <div class="ui-grid-b" align="center">
51 Vitesse : <input type="text" id="vitesse" name="vitesse" required
52 minlength="4" maxlength="8" size="10"/>
53 </div>
54 <div id="boutons" class="ui-grid-b" align="center">
55 <input type="button" id="valider" value="Valider " />
56 </div>
57 </html>
58
59

```

Figure 3-9. Le code source de l’interface web en HTML

3.4 Quatrième partie : La transmission des trames

L'entrée oscilloscope du Red Pitaya est liée à travers la matrice de commutation avec les deux Arduinos

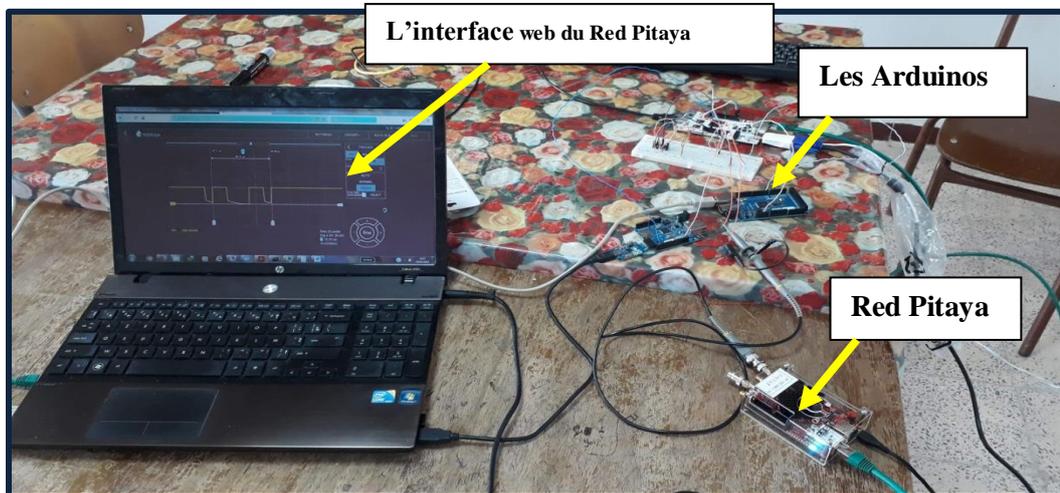


Figure 3-10. Le Red Pitaya et son interface web

L'Arduino sélectionné envoie une trame vers le Red Pitaya ou elle sera capturée, ce qui nous permet de décoder la trame transmise en binaire et mesurer sa vitesse de transmission

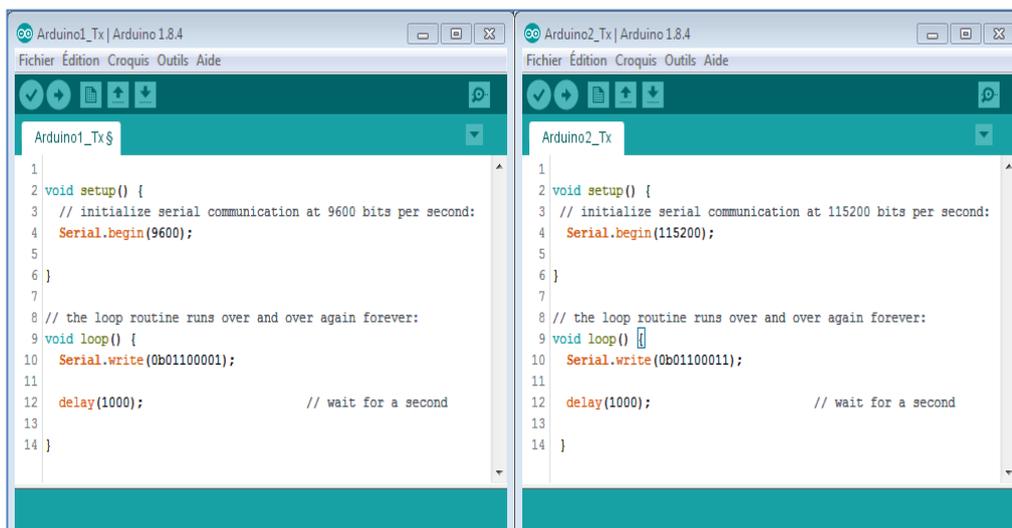


Figure 3-11. Les programmes des arduinos

3.5 Cinquième partie : L'analyse des trames

Dans cette partie nous allons analyser les trames transmises par les arduino en utilisant l'oscilloscope embarqué dans Red-Pitaya

- Pour l'Arduino 01



Figure 3-12. La trame transmis par le premier Arduino

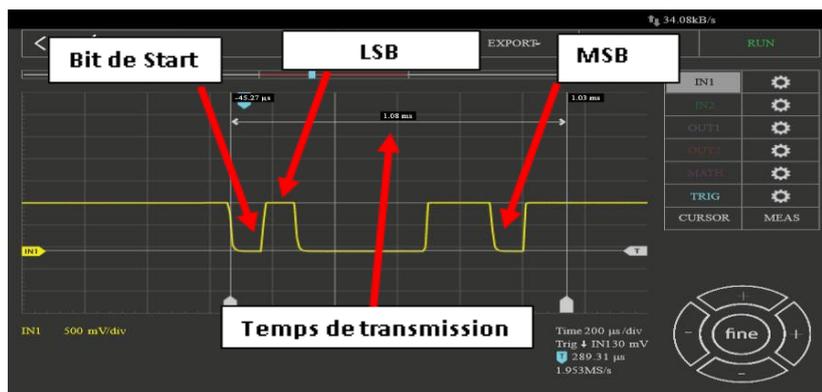


Figure 3-13. Analyse de la première trame

Le décodage binaire de la trame '01100001'

Et c'est le caractère « a »

Le temps de transmission c'est 1.08 ms

La durée de transmission de 10 bit $\rightarrow 1.08 \times 10^{-3}$ s

La durée de transmission d'un seul bit $\rightarrow dt = 0.108 \times 10^{-3}$ s

La vitesse de transmission d'un seul bit est : $1/dt = 9259 \approx 9600$

• Pour Arduino 02

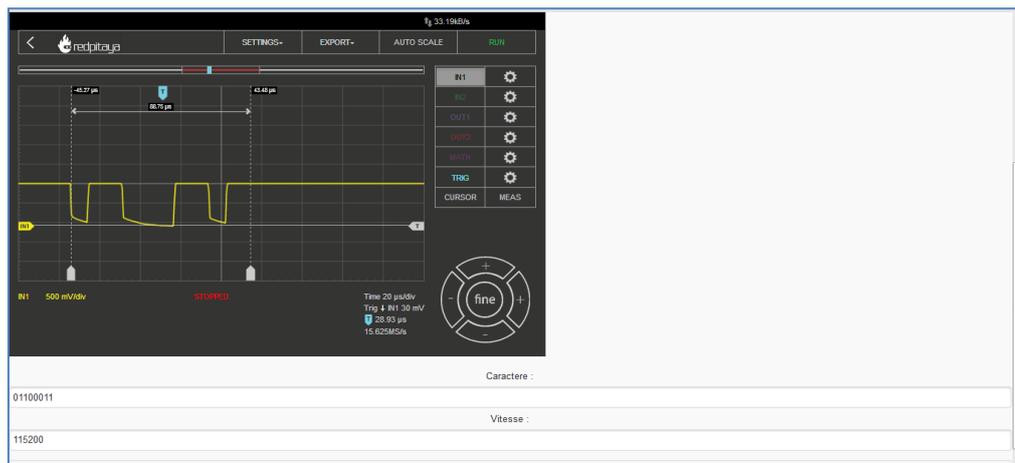


Figure 3-14. La trame transmise par le deuxième Arduino

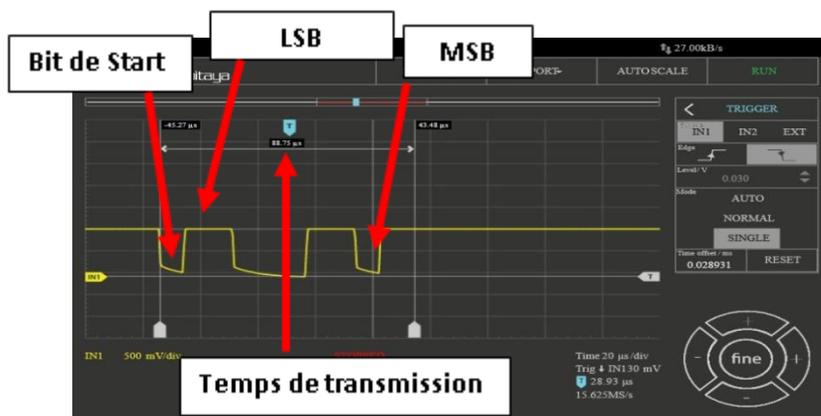


Figure 3-15. Analyse de la deuxième trame

Le décodage binaire de la trame '01100011'

Et c'est le caractère « c »

Le temps de transmission c'est 88.75us (Star bit + trame + stop bit)

La durée de transmission de 10 bit → 88.75×10^{-6} s

La durée de transmission d'un seul bit → $dt = 8.875 \times 10^{-6}$ s

La vitesse de transmission d'un seul bit est $1/dt = 112676 \approx 115200$

3.6 Sixième partie : Validation des paramètres

Après le décodage de la trame à partir d'un Arduino choisi par la matrice de commutation, et le calcul de sa vitesse de transmission, on doit valider ces deux paramètres avec un système d'autocorrection qui vérifie si les deux paramètres sont corrects en les comparant avec celle qui sont fixées dans les programmes JavaScript implémentés dans l'interface web.

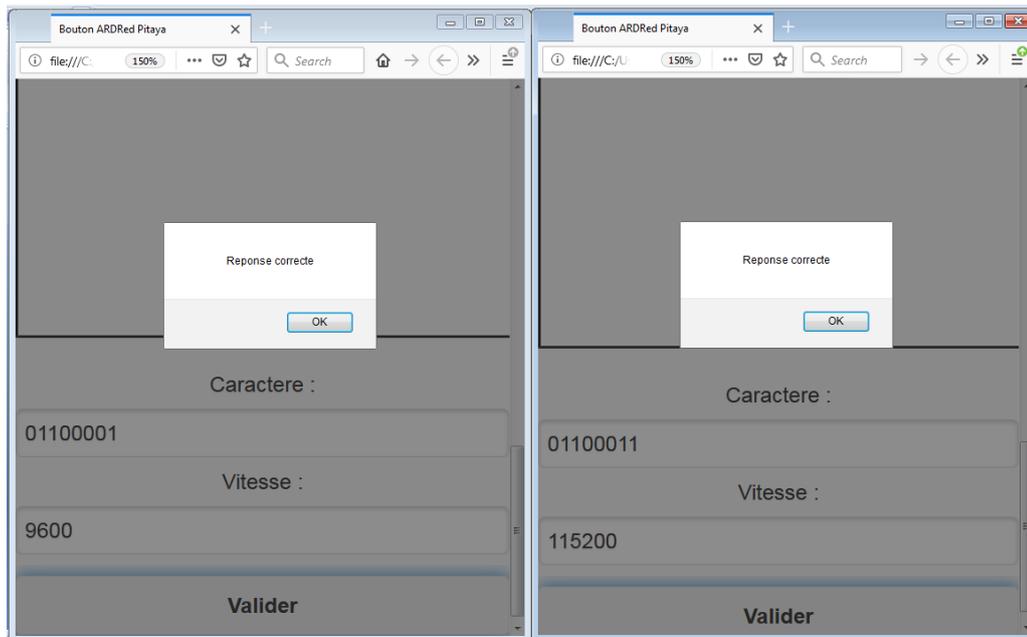


Figure 3-16. Les réponses correctes des Arduino

4. Conclusion

Nous avons présenté dans ce chapitre la phase de conception de notre projet qui contient les interfaces et les programmes commandant notre travail, qui nous ont aidés à décrire d'une façon détaillée, le fonctionnement de système dans le but de faciliter la réalisation.

Conclusion

Conclusion générale

Le travail exposé dans ce mémoire, s'intéresse à commander une matrice de commutation à distance basée sur un système embarqué.

Nous avons au fil des pages introduit des généralités sur celle-ci et des notions de base sur les outils de développement des interfaces web.

Dans ce projet nous avons réalisé des programmes qui implémentent un serveur embarqué, et qui génèrent des pages web commandant des composants électroniques (switch), ainsi que des programmes arduino pour initier la communication série.

Nous avons aussi détaillé la communication série qui fait la liaison entre la sortie de l'arduino et le port d'entrée d'un oscilloscope embarqué sur red pitaya, et l'outil utilisé pour analyser et décoder une trame binaire émise par arduino.

Bibliographie

Bibliographie

- [1] Introduction aux objets connectés et systèmes embarqués -<https://www.neotech-assurances.fr/seminaire-captronic-cybersecurite-systemes-embarques/> 24/05/2019
- [2] Introduction aux objets connectés et systèmes embarqués -<https://systematic-paris-region.org/fr/technologies-cles/systeme-embarques-internet-des-objets/> 24/05/2019
- [3] BENJAMIN Billet, 2015, *Système de gestion de flux pour l'Internet des objets intelligents*. Thèse de doctorat .Informatique-systèmes distribués. Yvelines. Université de Versailles.
- [4] Application de l'internet des Objets -<https://gladiacteur.com/iot-linternet-des-objets/> 24/05/2019
- [5] Avantage de l'internet des Objets- <https://www.relationclientmag.fr/Thematique/techno-ux-1256/Breves/Les-avantages-Internet-objets-dans-relation-client-320638.htm#W2ddHsOPPCO334Hr.97> 24/05/2019
- [6] les systèmes embarqués-<https://www.technologuepro.com/cours-systemes-embarques/cours-systemes-embarques-introduction.htm> 24/05/2019
- [7] Application des systèmes embarqués-<http://elyzee-consortium.com/domaines-dapplication-de-lelectronique-embarquee/> 24/05/2019
- [8]Exemple d'un système embarqué pcdduino - http://www.mon-club-elec.fr/pmwiki_mon_club_elec/pmwiki.php?n=MAIN.PCDUINOAppESSysReseauServeurTCPHTMLJSAjaxCanvaRGraphAnalogx1 18/06/2019
- [9] Exemple Arduino- <http://electroniqueamateur.blogspot.com/2017/06/programmer-une-appli-android-pour.html> 18/06/2019
- [10]Exemple Raspberry- <http://blogpeda.ac-poitiers.fr/col-chauvigny-techno-show/2016/07/01/raspberry-pi-3-application-smartphone/> 18/06/2019
- [11] Exemple Red Pitaya- <https://www.rs-online.com/designspark/virtual-lab-project-journey> 19/06/2019
- [12] Switch numérique-<http://www.ni.com/product-documentation/3960/en/> 10/06/2019
- [13] Type de Switch-
http://resources.uwcsea.edu.sg/UWCSEA_DT_East/Electronics/Electronics/Switches.html 15/06/2019
- [14] Relais- https://www.sonelec-musique.com/electronique_theorie_relais.html 09/06/2019
- [15] Fiche Technique d'ADG202A, <https://www.analog.com/en/products/adg202a.html>
- [16] X. HINAULT .*Le pcDuino v3 + Pyduino : Juillet 2014* 26/05/2019

Bibliographie

- [17] Arduino-http://www.mon-cluelec.fr/pmwiki_reference_arduino/pmwiki.php?n=Main.MaterielUno 26/05/201
- [18] L'UART-<http://www.composelec.com/uart.php> :09/06/2019
- [19] Red Piyata -<https://www.elektormagazine.fr/articles/red-pitaya-pulpeux-et-juteux> 26/05/2019 [20] Stephane Genaud *.Introduction a JavaScript* .2007
- [21] HTML-<https://developer.mozilla.org/fr/docs/Web/HTML> 27/05/2019
- [22] Node.js-<https://makina-corpus.com/blog/metier/2014/introduction-a-nodejs> 27/05/2019
- [23] Winscp-<https://winscp.net/eng/docs/lang:fr> 05/05/2019
- [24]PuTTY-<https://www.supinfo.com/articles/single/4093-presentation-outil-putty> 05/05/2019

Résumé

Les systèmes embarqués sont des systèmes électroniques et informatiques autonome sans entrées/sorties standards. Ils sont pilotés dans la majorité des cas par un logiciel (serveur), qui est complètement intégré au système qu'il contrôle, ce serveur est parfois accessible à distance via une interface web.

Le travail effectué dans ce mémoire porte sur la commande d'une matrice de commutation avec un aperçu sur le développement d'une interface web pour effectuer une manipulation à distance (commande à distance) en utilisant les éléments suivants : PcDuino, arduino, redpitaya.

Enfin, notre système a été appliqué au cas d'une manipulation d'une plateforme matérielle à distance, pour effectuer un TP de communication série.

Mots clés : Systèmes à commutation, manipulation à distance, interface web, PcDuino, arduino, redpitaya.

المخلص

الأنظمة المدمجة هي أنظمة إلكترونية و معلوماتية مستقلة بدون مداخل / مخارج. يتم التحكم فيها في معظم الحالات بواسطة برنامج (خادم) ، وهو متكامل تمامًا مع النظام الذي يتحكم فيه ، ويمكن الوصول إلى هذا الخادم في بعض الأحيان عن بُعد عبر واجهة ويب.

يتناول العمل المنجز في هذه المذكرة التحكم في مصفوفة التبديل مع نظرة عامة على تطوير واجهة ويب للمعالجة عن بُعد (التحكم عن بُعد) باستخدام العناصر التالية راسم الذبذبات redpitay كمبيوتر مصغر بيبي دوينو ,متحكم اردوينو

في الجزء الأخير تم تطبيق نظامنا على حالة معالجة النظام الأساسي للأجهزة عن بُعد، للقيام بعمل تطبيقي في الاتصال التسلسلي

الكلمات المفتاحية: أنظمة التبديل ، المعالجة عن بُعد ، واجهة الويب ،بيبي دوينو ، اردوينو ،ريد بيتايا