

Expérience pédagogique en Licence Professionnelle CONsEPT de conception d'un système embarqué communicant utilisant le SoPC Zynq (Xilinx)

Fabrice Aubépart

Aix-Marseille Université, Institut Universitaire de Technologie, département Geii
142 traverse Charles Susini - 13013 Marseille
fabrice.aubepart@univ-amu.fr

Le travail présenté ici s'inscrit dans le cadre d'un cours sur les systèmes intégrés programmables donné aux étudiants de la licence professionnelle « Conception des Systèmes Electroniques et optoélectroniques » (LP « CONsEPT ») de l'IUT d'Aix-Marseille Université (AMU).

Il consiste à réaliser un système embarqué en utilisant le système sur puce programmable (« *System On Programmable Chip* » - SoPC) Zynq-7010 de la société Xilinx. Ce composant est disponible sur la carte d'expérimentation Zybo, conçue par la société Digilent, laquelle est distribuée à un prix préférentiel par le CNFM.

La formation

La licence professionnelle CONsEPT vise à former des assistants ingénieurs (BAC+3). Elle a été l'une des premières LP créée en IUT en 2004 sous une appellation - peu précise - « EISI » (Electronique et Informatique des Systèmes Industriels), avec mention « Microélectronique », figure 1. Le manque de lisibilité avec ce nom générique, imposé au niveau ministériel, a vite montré des limites. Deux années plus tard, il nous a été permis de le modifier en LP « MeMs » (Microélectronique et Microsystèmes).

A l'origine notre formation était donc prévue pour former des assistants ingénieurs en conception, test, validation ou maintenance pour les entreprises de la microélectronique. Il nous a toutefois apparu un manque de débouchés dans ce secteur fortement cyclique et dominé par de grosses entreprises. Une première fusion avec une LP portée par le département Mesures Physiques voisin, nous a permis une ouverture vers des domaines de conception en électronique et optoélectronique touchant plus particulièrement des PME et PMI régionales. Un nouveau nom devait donc évoquer ces changements pour un affichage plus cohérent. C'est ainsi que la formation a été renommée LP « CONsEPT ».

La figure 1 montre bien la vie de ce type de formation et présente les différentes adaptations (subies ou voulues) qu'une formation professionnelle doit endurer pour assurer son existence sur le long terme : Le non-renouvellement d'une habilitation ministérielle jouant ici le rôle de couperet peu enviable.

Il est donc nécessaire d'évaluer régulièrement les besoins des entreprises régionales dans notre domaine pour adapter régulièrement le contenu et - si nécessaire - le nom de la formation pour plus d'attrait. Différents outils sont disponibles pour mesurer cela, tels que des enquêtes réalisées sur nos anciens diplômés, des rencontres avec les professionnels du secteur, etc. Les « effets de mode » sont également importants et nous insufflent de nouvelles perspectives. C'est actuellement le cas sur le développement des objets communicants touchant de nombreux secteurs industriels.

Suite à une seconde fusion cette année avec la LP « ET » (Electronique pour les Télécommunication), nous proposons une nouvelle version du programme pédagogique adaptée aux besoins des entreprises dans le domaine de la conception en électronique et optoélectronique, mais aussi à celles nécessitant des compétences en systèmes de télécommunication associés aux objets communicants. Une nouvelle appellation, LP « SELECT » (Systèmes Electroniques Embarqués CommunicanTs) est validée et sera effective à partir de 2017.

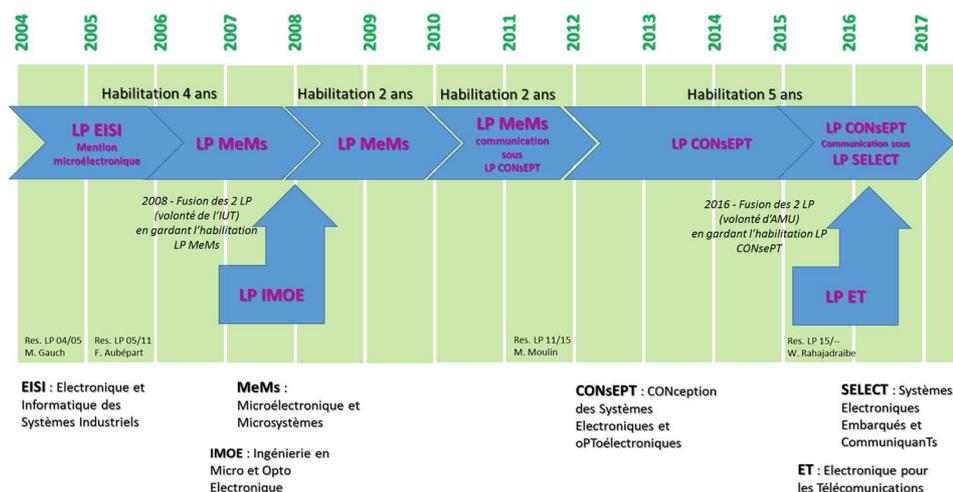


Figure 1 – Chronologie de la Licence Professionnelle « CONsEPT » depuis 2004.

Nos étudiants sont issus de formation de type BTS (système numérique, système optique ou TPIL), de DUT (Geii, MP, informatique option SI) ou de cycle Licence universitaire. Certains sont en formation par alternance sous forme de contrats de professionnalisation. Par ailleurs, nous accueillons depuis plusieurs années des étudiants mexicains dans le cadre d'un programme national d'accueil et, depuis cette année trois étudiants marocains de l'Ecole Supérieure de Technologie de Casablanca par le circuit des bourses Erasmus+.

La grosse difficulté consiste donc à fournir des enseignements les plus adaptés à des étudiants de provenance variée, mais ayant toutefois un socle de compétences et de cultures technologiques similaire.

En résumé, la « survie » d'une formation professionnelle nécessite une évolution constante dont l'un des mécanismes est l'innovation pédagogique. Celle-ci doit répondre à plusieurs critères comme (i) l'attrance de nouveaux étudiants de plus en plus sensibles aux « modes » technologiques, (ii) la réussite d'étudiants aux compétences scientifiques diverses, et surtout (iii) l'intérêt des compétences acquises par les diplômés pour les entreprises régionales. Le projet pédagogique présenté ici essaie de répondre à ces différents aspects.

Le projet

Nous proposons aux étudiants de créer un système embarqué en utilisant les capacités internes du SoPC Zynq-7010 commercialisé par Xilinx. Ce système est associé à un module de communication Wifi externe, piloté par commandes AT via une liaison série asynchrone. Nous obtenons ainsi un « objet connecté » qui permet de recevoir et d'émettre des informations sur téléphone portable, tablette graphique ou sur poste informatique utilisant un navigateur internet. Ce projet pédagogique n'a été réalisé qu'une seule fois et nécessite encore quelques ajustements et amélioration, notamment en ce qui concerne la programmation de la page internet côté « client » (utilisateur).

Le Zynq est particulièrement intéressant dans ce contexte car il intègre à la fois un processeur double cœur ARM cortex A9, un gestionnaire d'interruptions, un convertisseur Analogique-Numérique et d'autres périphériques, dans une zone nommée « *Processing System* » (PS). Ce composant dispose d'une seconde zone nommée « *Programmable Logic* » (PL), constituée de cellules logiques programmables, de mémoires, de blocs pour le traitement du signal numérique (figure 2 et 3).

Le système embarqué, réalisé par les étudiants, nécessite l'utilisation du processeur ARM associé à deux périphériques de type série asynchrone UART auquel est ajouté un « périphérique maison » bien particulier. Il s'agit d'un fréquencemètre, créé à partir du langage VHDL, et encapsulé sous forme de bloc IP (« *Intellectual Property* »). Ce dernier est connecté au processeur via un bus AXI-Lite. Ce bus permet la communication entre le système à processeur « PS » et le bloc IP réalisé et intégré dans la zone à logique programmable « PL ».

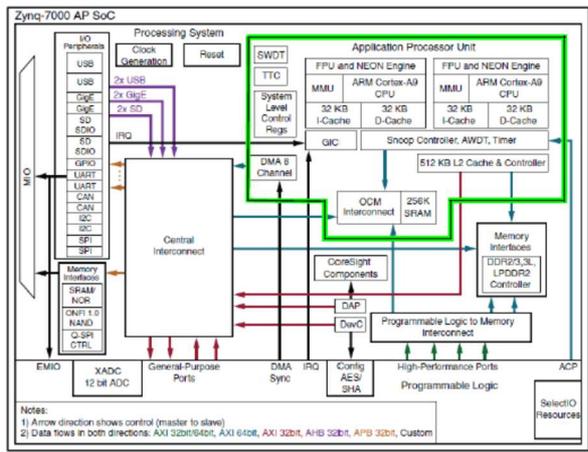


Figure 2 – Le « Processing System » (PS)

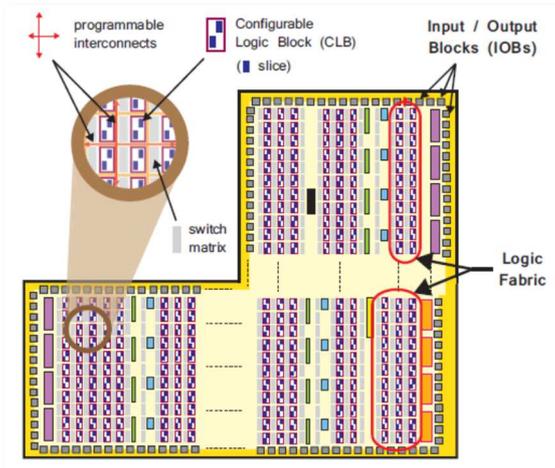


Figure 3 – La « Programmable Logic » (PL)

Le fréquencemètre est une application souvent utilisée dans les travaux pratiques en conception numérique. Il en existe différents principes et architectures. Nous avons choisi ici le principe du comptage des périodes du signal en mesure sur une durée de temps fixe (figure 4). L'architecture choisie inclut un changement de calibre automatique permettant ainsi une mesure de fréquence entre 01 Hz et 99 kHz dans cette version.

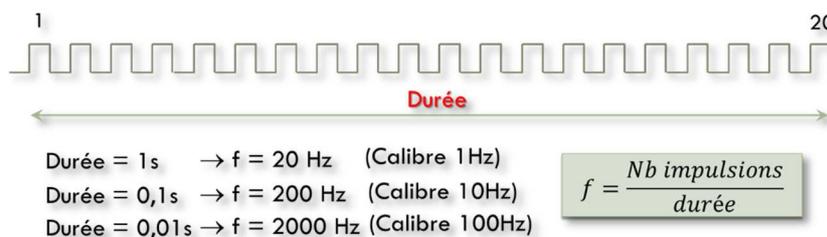


Figure 4 – Principe de la détermination de la fréquence d'un signal périodique sur une durée fixe qui peut changer suivant le calibre sélectionné (automatiquement).

Au sein de notre département Geii, c'est une application que nous apprécions bien lors de l'apprentissage du langage VHDL tant au niveau du DUT qu'en Licence professionnelle. Effectivement, le principe de fonctionnement est facile à comprendre et l'architecture est constituée de fonctions combinatoires et séquentielles simples à coder en VHDL même pour des étudiants débutants dans ce domaine (figure 5). Les fonctions peuvent ainsi être décrites en utilisant la gamme complète de la syntaxe et du style possible à partir de ce langage de conception matérielle. Par exemple, l'aspect algorithmique de certaines fonctions est facile à coder par une instruction concurrente en utilisant un « processus ». Celle-ci peut aussi être écrite de manière différente, et équivalente, par une instruction conditionnelle « when...else » ou une instruction sélective « with...select ».

Au-delà de cet aspect concernant les descriptions comportementales, d'autres notions importantes sont également vues : codage d'une machine d'état, descriptions structurales (c'est-à-dire l'équivalent d'un schéma), écriture en VHDL de « testbench », comparaison entre simulations comportementales et post-placement-routing, etc.

Ce travail sur le fréquencemètre peut également ouvrir d'autres possibilités. Nous l'avons ainsi expérimenté dernièrement sous la forme de projets tutorés pour des étudiants de DUT de l'école supérieure de technologie (EST) de Casablanca lors d'une collaboration internationale. Certains projets ont, par exemple, été étendus à l'électronique de conditionnement pour le traitement et la mise en forme du signal analogique à mesurer.

L'encapsulation du fréquencemètre en bloc IP personnalisé n'est pas une démarche complexe. Elle requiert toutefois de suivre une méthodologie un peu « lourde » où de la rigueur est demandée aux étudiants. La préparation y est assez minutieuse. Plusieurs modifications doivent être réalisées sur la

description VHDL de niveau la plus élevée pour l'adapter au bloc IP qui sera connecté au bus AXI. Par exemple, l'une des modifications consiste à ajouter des registres 32bits à l'architecture de la figure 6 pour mémoriser le résultat du comptage de la fréquence. Mais ce n'est pas la seule modification et les étudiants peuvent être vite déconcertés dans l'arborescence des fichiers d'une conception hiérarchisée de ce type, et dans les descriptions VHDL modifiés en conséquence pour le bloc IP.

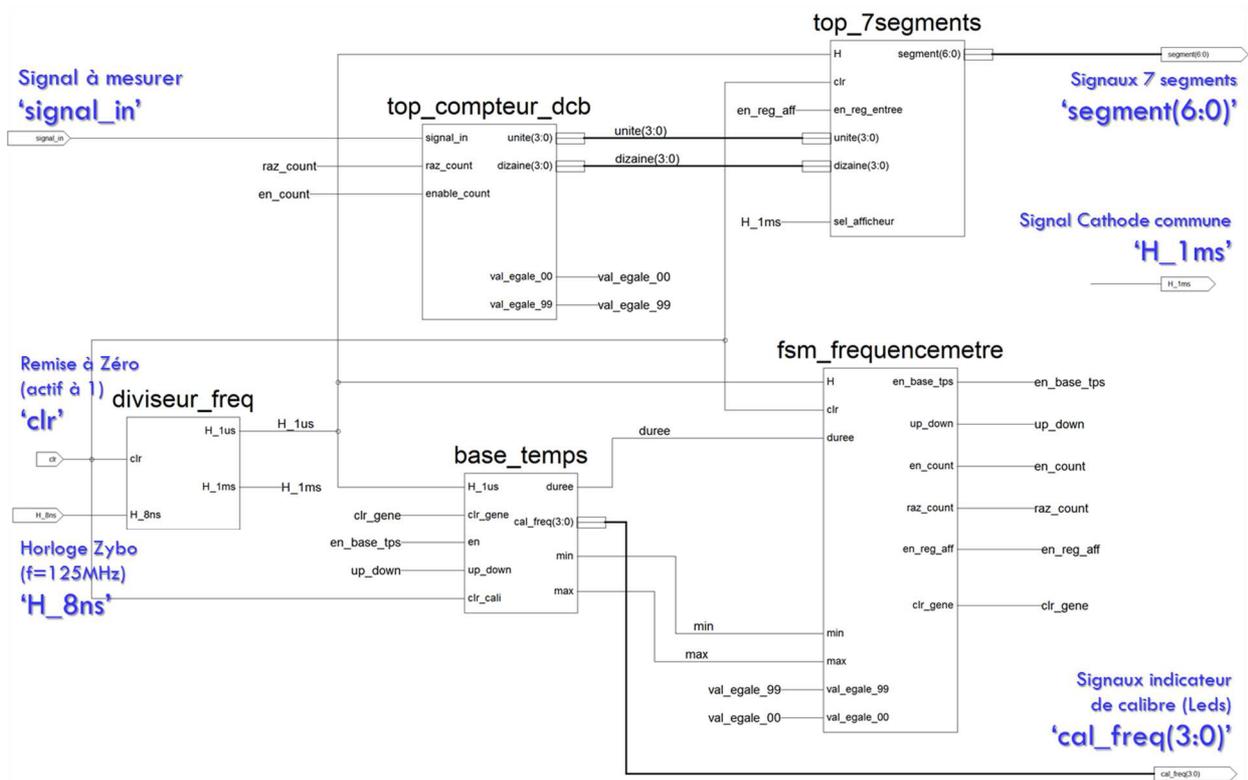


Figure 5 – Architecture du fréquencemètre : Les impulsions du signal à mesurer sont comptées par des compteurs DCB mis en cascade dans « top_compteur_dcb » et le résultat affichées sur des afficheurs 7 segments via une interface « top_7segments » assurant à la fois le décodage DCB-7 segments et le multiplexage des afficheurs. Une machine à états finis de type Moore « fsm_frequecemetre » assure la coordination de l'ensemble, mais aussi le changement automatique de la durée (donc le calibre) dans « base_temps ». Enfin le « diviseur_freq » assure des signaux d'horloge compatibles à notre application (périodes de 1 μ s et 1 ms) à partir du signal d'horloge (trop rapide) disponible sur la carte Zybo.

Au final, l'assemblage du processeur et du périphérique « fréquencemètre » se fait sous forme d'assemblage de blocs graphiques, grâce aux outils automatisés de l'environnement Vivado (figure 6).

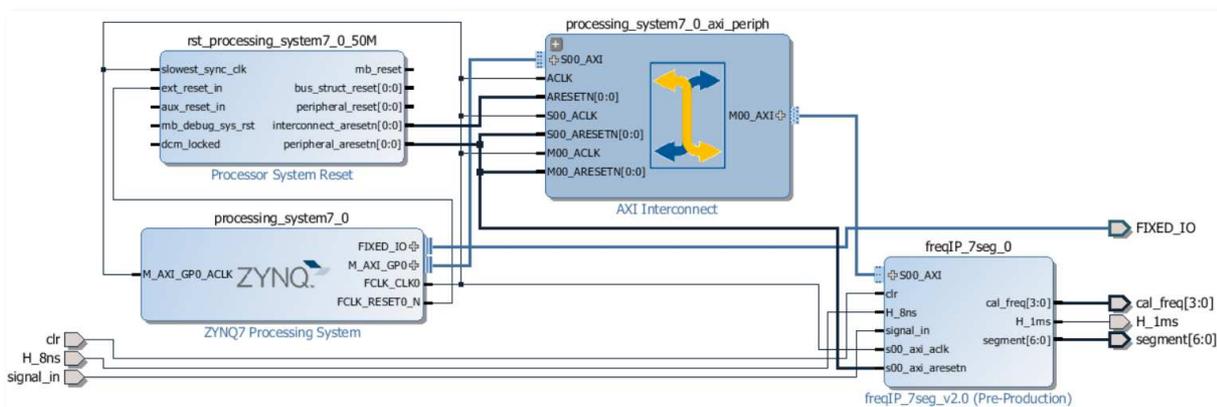


Figure 6 – Schéma bloc du système embarqué : Le fréquencemètre encapsulé sous forme de bloc IP « freqIP_7seg_v2.0 » est relié au « ZYNQ7 Processing System » via le bus AXI, symboliquement représenté par « AXI Interconnect ». A cet ensemble est ajouté automatiquement une gestion de remise à zéro pour le processeur « Processor System Reset ».

Les périphériques secondaires, les UART, nécessaires à notre système embarqué, sont ceux disponibles dans la zone PS (figure 2). Ils n'apparaissent donc pas sur la figure 6. Leur configuration est réalisée très simplement à l'aide de fenêtres de sélection (figure 7).

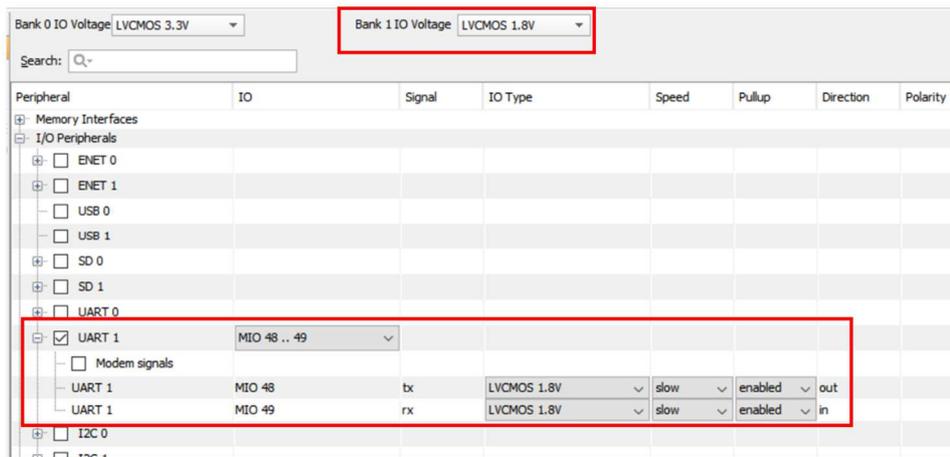


Figure 7 – Configuration de l'un de deux UART disponible dans le PS. Le choix des broches se fait en fonction de la carte Zybo.

A l'extérieur du Zynq, un module Wifi est branché via l'un des connecteurs de la carte Zybo. Ce module utilise un système sur puce ESP8266 et conçu par la société Espressif. Ce SoC est construit autour d'un microprocesseur Tensilica L106 32 bits, de la mémoire flash, divers périphériques, mais surtout des fonctionnalités pour une communication Wifi au standard 802.11. Le fabricant fournit un SDK pour la programmation de son circuit. Mais cette solution est assez fastidieuse pour la programmation des couches TCP/IP et n'est pas l'utilisation recherchée ici. Une seconde solution consiste à configurer et utiliser le module par un jeu de commandes AT transmises via une liaison série asynchrone (figure 8). L'ESP8266 est utilisé en mode « STation », c'est-à-dire dans le mode qui permet de se connecter à un réseau wifi existant. La configuration du module se fait par commandes AT et par la surveillance des réponses du module par la ligne de retour.

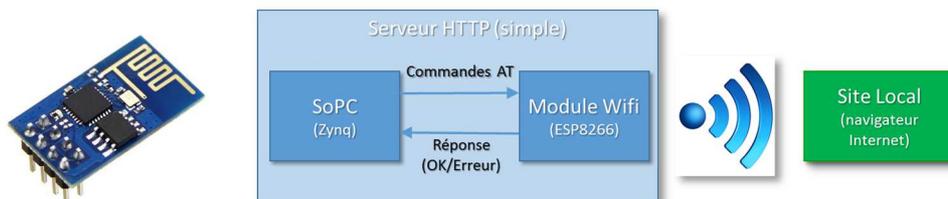


Figure 8 – L'ESP8266, disponible sous forme de module, peut être configuré et utilisé par notre système embarqué via une simple liaison série asynchrone en utilisant des commandes AT.

La programmation du dispositif se fait en langage C dans l'environnement SDK (basé sur l'IDE Eclipse) fourni également par Xilinx. Nous utilisons les fonctions C disponibles pour piloter les périphériques « standards ». L'accès des données provenant du fréquencemètre est fait par lecture/écriture de registres 32 bits. La plus grosse difficulté consiste à programmer un « mini » serveur http simplifié pour ne considérer que des requêtes GET et gérer qu'un seul client à la fois, (figure 9). Plusieurs fonctions en C doivent donc être écrites par les étudiants. Par la suite, plusieurs tests sont réalisés en surveillant le trafic des échanges entre le SoPC et le module Wifi, (figures 10 et 11).

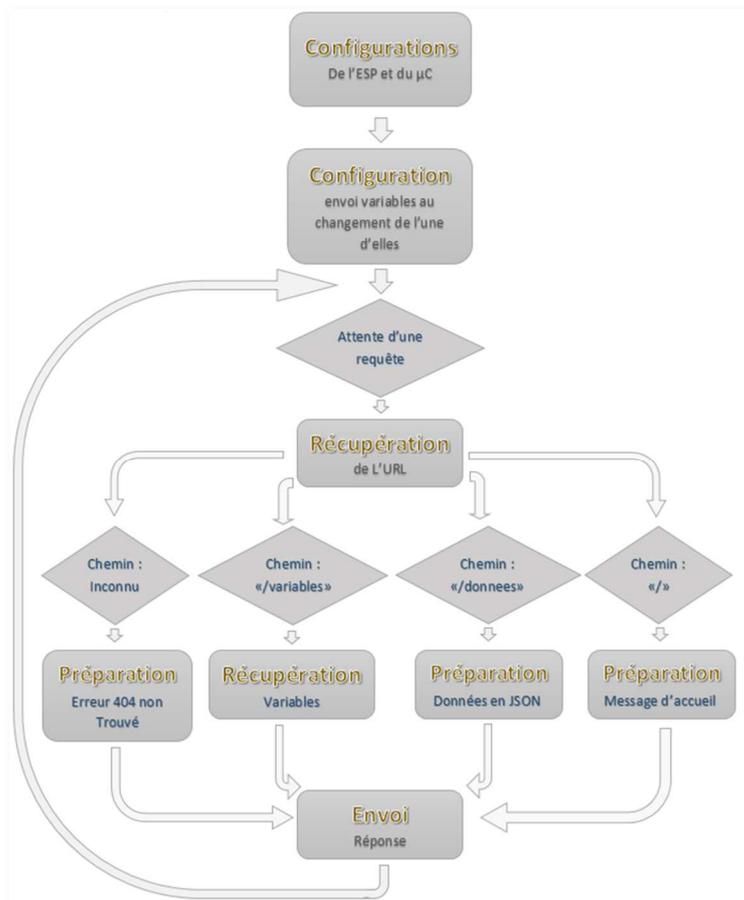


Figure 9 – structure du programme écrit en C et implémenté dans le SoPC.

```

+IPD,0,380:GET /donnees HTTP/1.1
Host: 192.168.0.123
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:49.0) Gecko/20100101 Firefox/49.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: fr,fr-FR;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
DNT: 1
Connection: keep-alive
Upgrade-Insecure-Requests: 1
Cache-Control: max-age=0
  
```

Figure 10 – Demande de requête - « espionné » - en provenance de l'ESP8266 (donc d'un client wifi distant) pour le SoPC.

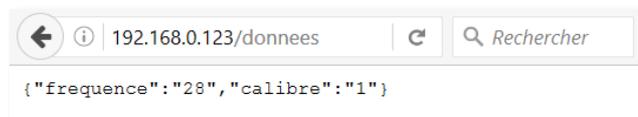


Figure 11 – Affichage du résultat dans le navigateur du client... Bien entendu, une mise en forme s'impose !

Conclusion

Cet enseignement sous forme de projet met en œuvre la création d'un système embarqué intégré dans un composant très puissant comme le Zynq. Il exploite au mieux d'autres compétences acquises par les étudiants durant les mois de formation : Electronique numérique, programmation en VHDL, programmation en langage C, protocoles de communication, etc.

Plusieurs améliorations doivent être apportées notamment sur le développement du site local en utilisant les langages HTML/CSS et JavaScript. A l'entrée du fréquencemètre, nous pourrions aussi envisager d'ajouter un capteur de luminosité ou de couleur dont l'information correspond à une fréquence (capteurs TAOS). Cela donnera une envergure supplémentaire en optoélectronique à ce projet.

Ce travail peut également être renforcé lors des projets tutorés. C'est particulièrement le cas de certains étudiants qui travaillent sur la réalisation d'une partie de l'électronique d'un nano-satellite en collaboration avec la Laboratoire d'Astrophysique de Marseille (LAM) et utilisant le composant Zynq.