

Conception d'un système embarqué communicant utilisant le SoPC Zynq (Xilinx) et le module wifi ESP8266 : Expérience en Licence Professionnelle CONsEPT

F. Aubépart

Aix-Marseille Université, Pôle CNFM PACA,
Institut Universitaire de Technologie, département GEII
142 traverse Charles Susini - 13013 Marseille
Contact email : fabrice.aubepart@univ-amu.fr

Cet article évoque un travail de conception en électronique embarqué mené sous forme de projet transversal pour des étudiants de Licence Professionnelle (LP). Il est réalisé dans un module d'enseignement de 32 h intitulé « Systèmes sur puce programmable - Système embarqué ». Il permet de renforcer certaines compétences abordées en début d'année, bien particulières au domaine fixé, comme la conception avec le langage VHDL ou la programmation en langage C d'un microcontrôleur 32 bits. Cette réalisation permet aussi d'acquérir de notions en conception de système sur puce programmable et dans celles de l'utilisation d'un module de communication sans fils Wifi afin d'obtenir un « objet connecté ».

I. Introduction

L'internet des Objets (« Internet of Things », l'IoT) est un domaine en pleine expansion et l'on estime à près de 80 milliards d'objets connectés d'ici la fin de cette décennie. Tous les secteurs seront touchés par ce développement (Agriculture, industrie, construction, énergie, transport, santé, finance, télécommunication, distribution, loisirs, etc.), à tel point que certains envisagent une « véritable révolution des modèles économiques » (1).

En tant qu'enseignants, nous ne pouvons pas échapper à ce phénomène et nous devons prendre la mesure de cet engouement dans nos enseignements. Il devient donc nécessaire de créer des projets qui impliquent les techniques et protocoles de communication sans fil. C'est ainsi que nous avons conçu un module d'enseignement, conçu comme la réalisation d'un projet technique, utilisant un système sur puce programmable associé à un module de communication Wifi pour créer un objet connecté. Ce travail est réalisé auprès d'étudiants de la licence professionnelle (LP) CONsEPT (Conception des Systèmes Electroniques et oPToélectroniques) du département Génie Electrique et Informatique Industrielle (GEII) de l'IUT d'Aix-Marseille Université (AMU) (2). 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 (3). Ce composant est disponible sur la carte d'expérimentation Zybo, conçue par la société Digilent, laquelle est distribuée à un prix préférentiel par la Coordination Nationale pour la Formation en Microélectronique et en nanotechnologies (CNFM) (4).

II. Licence Professionnelle « CONsEPT »

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 » (5), 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. Nous avons remarqué des possibilités de débouchés fortement cycliques dans ce secteur et dominé par les gros groupes. 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 réaliser et surtout réussir pour assurer son existence sur le long terme : le non-renouvellement d'une habilitation ministérielle jouant ici le rôle de couperet peu enviable.

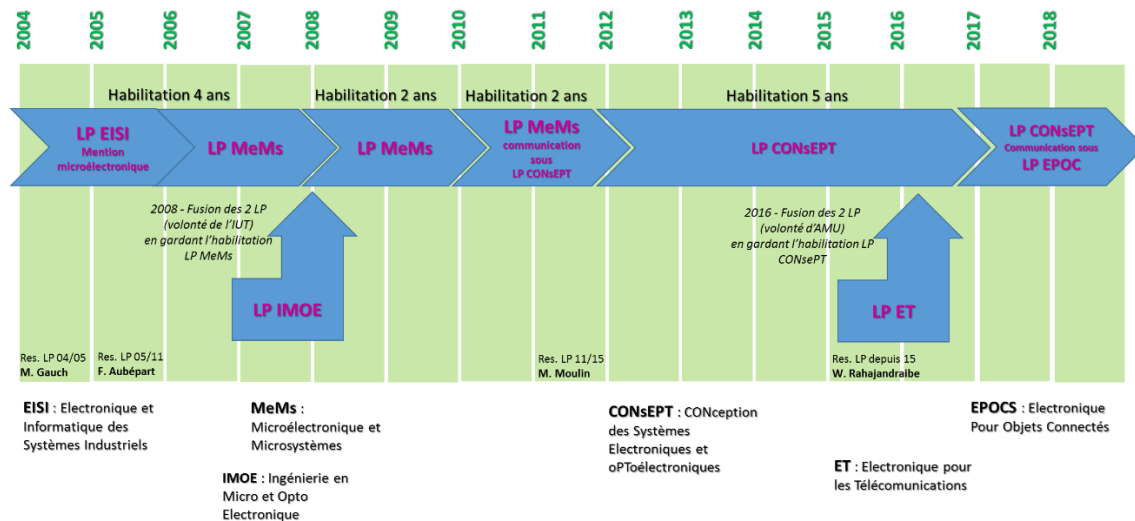


Fig.1. Chronologie de la Licence Professionnelle « CONsEPT » depuis 2004.

Il est donc nécessaire d'évaluer régulièrement les besoins des entreprises régionales dans notre domaine pour adapter le contenu et - si nécessaire - le nom de la formation afin de la rendre plus attractive. 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 technologique » 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 (6).

Suite à une seconde fusion depuis 2016 avec la LP « ET » (Electronique pour les Télécommunications), 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, « EPOC » (Electronique Pour Objets Connectés), devrait être adoptée prochainement, afin d'afficher encore plus clairement le lien avec la réalisation électronique des objets communicants.

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 du Mexique dans le cadre d'un programme national géré par l'association des directeurs des IUT (ADUIT) et, depuis cette année, des étudiants marocains de l'Ecole Supérieure de Technologie de Casablanca (ESTC) par le circuit des bourses Erasmus+.

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

En résumé, la pérennisation d'une formation professionnelle nécessite une évolution ainsi qu'une innovation pédagogique constantes. Celle-ci doit répondre à plusieurs critères comme (i) l'attractivité envers de nouveaux étudiants motivés par les nouvelles technologies, (ii) la réussite d'étudiants aux compétences scientifiques diverses, et surtout (iii) l'adéquation des compétences fournies par la formation aux besoins des entreprises locales. Le projet présenté ici essaie de répondre à ces différents aspects.

III. Le Projet

Ce projet est réalisé dans un module d'enseignement intitulé « Système sur Puce Programmable » pour un volume horaire de 32 heures. Les étudiants sont amenés à travailler par binôme sur la totalité du système décrit ci-dessous. Cet enseignement arrive au second semestre et doit être perçu comme un travail transversal. Effectivement, les étudiants ont l'occasion de renforcer plusieurs compétences abordées en début d'année, comme la conception d'un système électronique à l'aide du langage de description matérielle VHDL, la programmation en langage C d'un microcontrôleur de type ARM, l'utilisation de protocoles de communication, etc. Les nouvelles techniques apportées sont celles de la réalisation du système sur puce (association logique configurable et microcontrôleur ARM via le bus AXI et utilisant des blocs IP), l'apprentissage des outils de CAO (Vivado, SdK-Xilinx), l'utilisation des commandes Hayes (commandes AT), le protocole http, etc.

Nous proposons aux étudiants de créer un système embarqué en utilisant les capacités internes du SoPC Zynq-7010 commercialisé par la société américaine Xilinx. Ce système est associé à un module de communication Wifi externe, piloté par une liaison série asynchrone. L'objectif est la réalisation d'un fréquencemètre à changement de fréquence automatique « connecté » qui permet de recevoir et d'émettre des informations sur un smartphone, une tablette graphique ou sur n'importe quel ordinateur utilisant un navigateur internet. Ce fréquencemètre pourrait le cas échéant être associé à des capteurs fournissant leur mesure sous forme fréquentielle.

Composant « Zynq »

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 et de blocs pour le traitement numérique du signal, figures 2 et 3 (3).

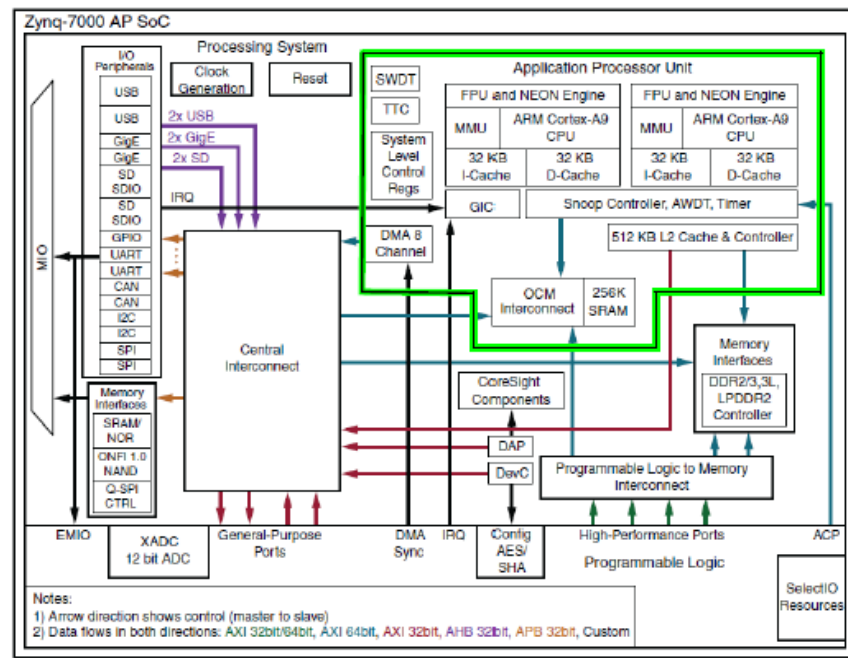


Fig.2. « Processing System » (PS) : c'est la zone sur silicium sur laquelle est implantée le double cœur ARM cortex A9 ainsi que ses différentes fonctionnalités et périphériques. Ces derniers peuvent être choisis et configurés simplement par une interface graphique disponible dans l'outil de CAO Vivado (Xilinx).

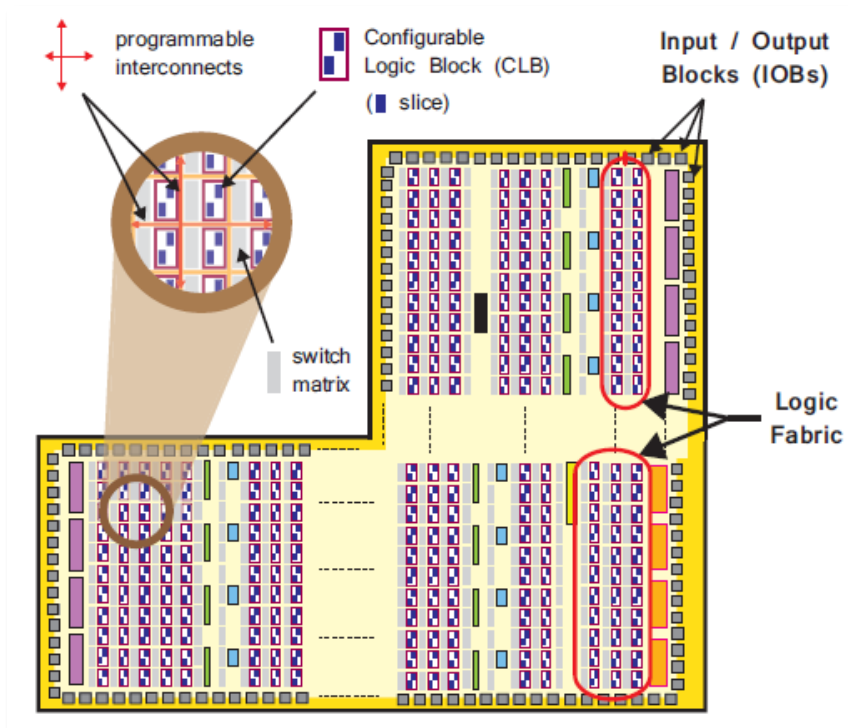


Fig.3. « Programmable Logic » (PL) : il s'agit de la zone équivalente à celles intégrées dans les FPGA Xilinx. Nous y trouvons des bascules D associées à des tables de correspondances (LUT) pour réaliser des fonctions combinatoires et séquentielles, mais aussi des blocs mémoires et des blocs intégrant opérateurs de multiplication et additionneurs/accumulateurs pour les calculs nécessaires au traitement numérique du signal.

Le système embarqué, réalisé par nos étudiants, nécessite l'utilisation du processeur ARM associé à deux périphériques de type série asynchrone (UART) auxquels est ajouté

un périphérique « maison » réalisé par nos étudiants. Il s'agit d'un fréquencemètre, créé à partir du langage de description matérielle 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 ».

Conception VHDL : fréquencemètre

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 (5) et (7). L'architecture choisie inclut un changement de calibre automatique permettant ainsi une mesure de fréquence entre 1 Hz et 99 kHz dans cette version.

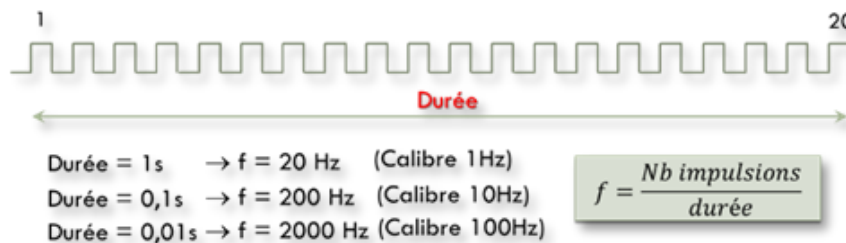


Fig.4. Principe de la détermination de la fréquence d'un signal périodique sur une durée fixe qui peut changer suivant le calibre sélectionné automatiquement.

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 LP. 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 débutant dans ce domaine, figure 5. Les fonctions peuvent 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 » (8) (9).

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

Ce travail sur le fréquencemètre peut également ouvrir d'autres possibilités de projet. Nous l'avons ainsi expérimenté dernièrement sous la forme de projets tutorés pour des étudiants de 2^{ème} année de DUT GEII de l'ESTC 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 de ce fréquencemètre en bloc IP personnalisé n'est pas une démarche complexe. Elle demande toutefois de suivre une méthodologie précise. La préparation y est assez minutieuse. Plusieurs modifications doivent être réalisées sur la description VHDL de niveau le plus élevé pour l'adapter au bloc IP qui sera connecté au bus AXI (10). Par exemple, l'une des modifications consiste à ajouter des registres 32 bits à l'architecture de la figure 5 pour mémoriser le résultat du comptage de la fréquence, etc.

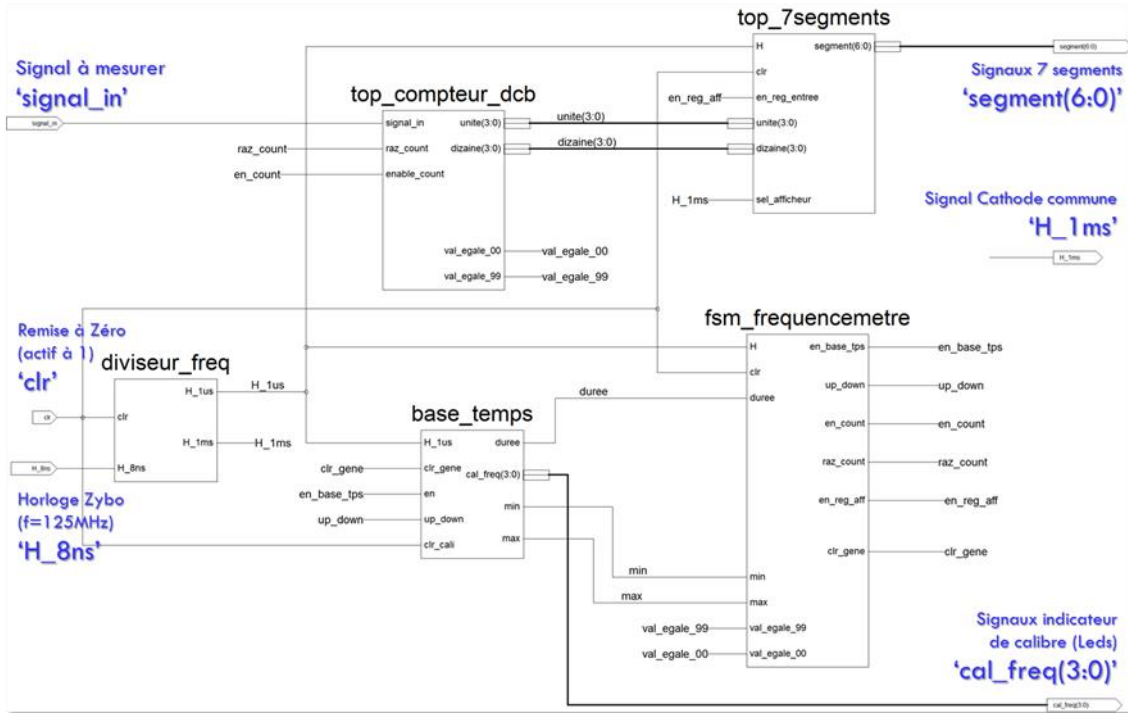


Fig.5. Architecture du fréquencemètre : les impulsions du signal à mesurer sont comptées par des compteurs DCB mis en cascade dans « top_compteur_dcb ». Le résultat est indiqué sur des afficheurs 7 segments via l'interface « top_7segments » assurant à la fois le décodage DCB-7 segments et le multiplexage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

Architecture du Système sur Puce sous Vivado

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.

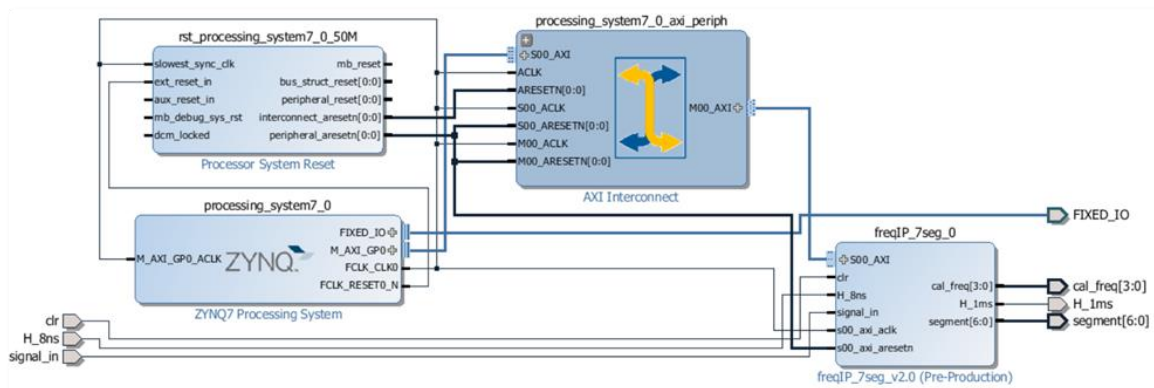


Fig.6. Schéma bloc du système embarqué : Le fréquencemètre encapsulé sous forme de bloc IP « freqIP_7seg_v2.0 » est relié au « ZYNQ7 Processing System » via le bus AXI, symboliquement représenté par « AXI Interconnect ». A cet ensemble est ajouté automatiquement une gestion de remise à zéro pour le processeur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

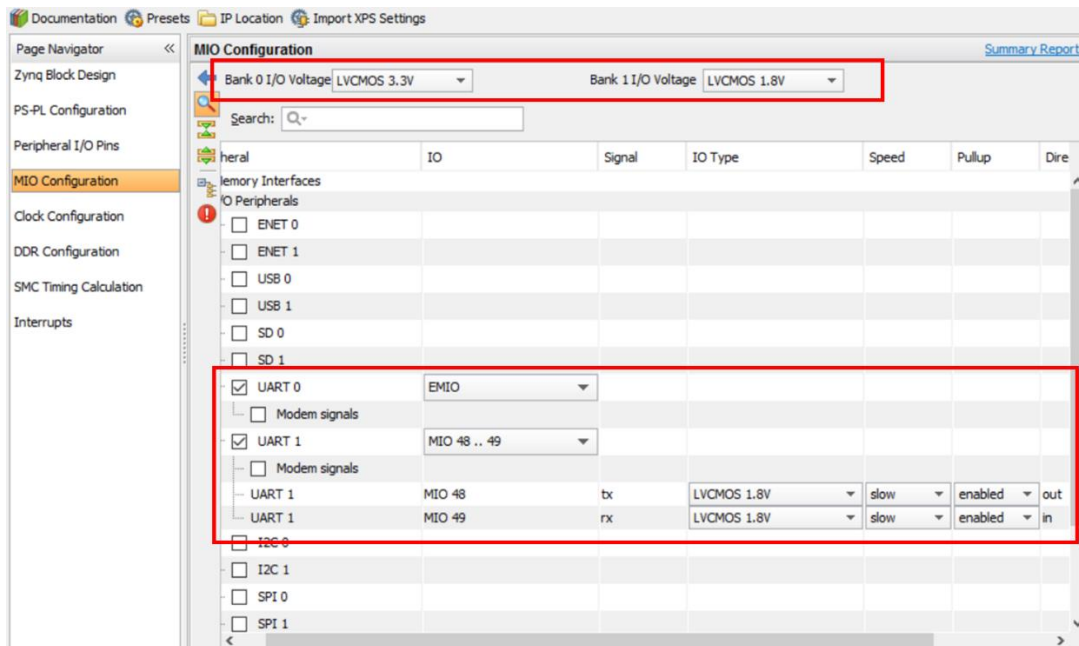


Fig.7. Configuration de l'un de deux UART disponible dans le « Processing System ». Le choix des broches à utiliser sur le Zynq est fonction du câblage de celui-ci sur la carte Zybo

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, conçu par la société chinoise Espressif (11). 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 kit de conception logiciel (« Software Design Kit », 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 autre solution, que nous avons utilisée, consiste à configurer et utiliser le module par un jeu de commandes « AT » (dîtes aussi commandes Hayes) transmises via une liaison série asynchrone (12) (13), figure 8.

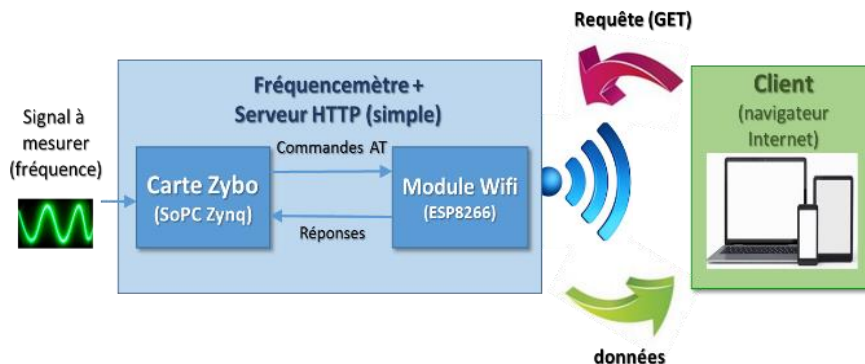


Fig.8. L'ESP8266, disponible sous forme de module, peut être configuré et utilisé par notre système embarqué via une liaison série asynchrone en utilisant des commandes « AT ».

Les commandes envoyées correspondent à une ligne de texte encodée sous forme ASCII. Elles commencent toujours par les caractères « AT » et se terminent par le caractère symbolisant la touche de validation « Carriage Return ». 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 réalise ainsi facilement en utilisant les commandes « AT » adéquates et en traitant les réponses du module par la ligne UART de retour (11).

Programmation en C

La programmation du dispositif se fait en langage C dans l'environnement « Xilinx - SDK » (basé sur l'IDE Eclipse). Nous utilisons les fonctions C disponibles pour piloter les différents périphériques. La figure 9 présente la structure générale du programme mis en œuvre.

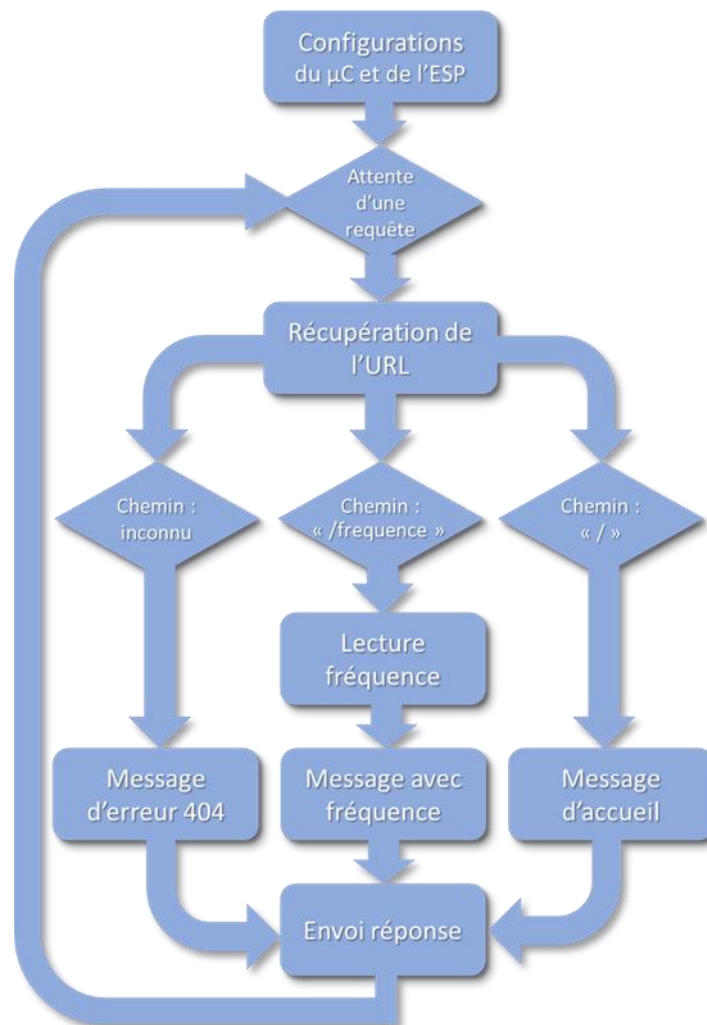


Fig.9. Structure du programme écrit en C et implémenté dans la mémoire du SoPC. Nous remarquerons la boucle principale d'attente d'une requête http qui sera traitée après lecture de l'URL.

La première étape consiste à configurer le µC et les 2 UART qui lui sont associés (un pour le débogage et le second pour la communication avec l'ESP8266). Le module wifi est alors configuré par les commandes AT. Les étapes de configuration sont : (i) un test de la présence du module, (ii) sa configuration en mode « STation », (iii) sa connexion à un réseau Wifi existant (il est nécessaire de lui fournir un nom de réseau et un mot de passe

d'accès. Pour les essais, il est possible de créer un tel réseau à partir d'un smartphone), (iv) fournir une adresse IP disponible sur ce réseau, (v) activer le mode multiple connexions, (vi) configurer le module comme serveur.

Par la suite, un « mini » serveur HTTP tourne en boucle. Il ne traite que des requêtes GET provenant d'un client internet ayant accès au serveur par son adresse IP. Le client doit être sur le même réseau internet que le serveur. Trois possibilités sont alors traitées, une mauvaise saisie de l'adresse IP (chemin inconnu), une demande d'accès directe (adresse IP seule) et une demande de lecture de la fréquence (adresse IP suivie de « /frequence »). Dans chacun des cas, un message sera retourné au navigateur client avec l'envoi de la fréquence et du calibre retournés lors de cette demande. L'accès des données provenant du fréquencemètre est fait par lecture/écriture d'un registre 32 bits utilisé dans le fréquencemètre.

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.

```
+IPD,0,356:GET /frequence HTTP/1.1
Host: 192.168.0.123
User-Agent: Mozilla/5.0 (Windows NT 10.0;
Accept: text/html,application/xhtml+xml,a
Accept-Language: fr,fr-FR;q=0.8,en-US;q=0
Accept-Encoding: gzip, deflate
DNT: 1
Connection: keep-alive
Upgrade-Insecure-Requests: 1
```

Fig.10. Demande de requête qui peut être « espionnée » en provenance de l'ESP8266 (donc d'un client wifi distant) : La requête GET suivi de « /frequence » sont parfaitement reçus par le module et transmis au SoPC pour traitement.

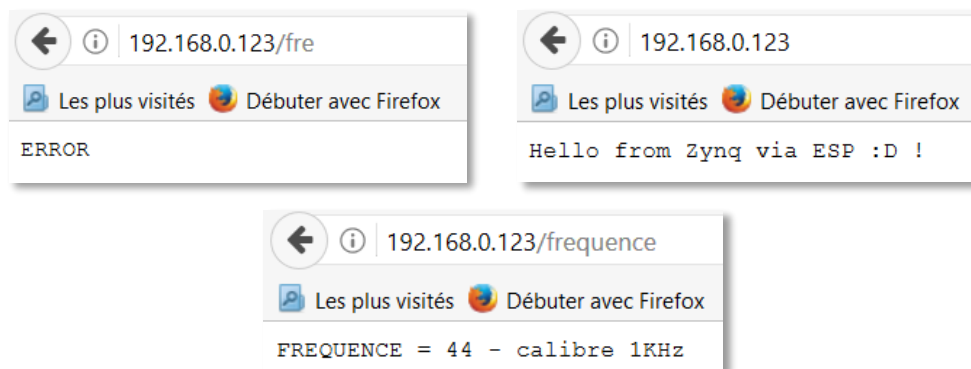


Fig.11. Affichage du résultat dans le navigateur du client : En haut à droite, cas d'une mauvaise saisie ; en haut à gauche, retour d'un message d'accueil ; en bas, demande et affichage de la fréquence et du calibre utilisé.

IV. Conclusion

Cet enseignement, sous forme de projet de développement, met en œuvre la création d'un système embarqué intégré dans un composant très puissant comme le SoPC Zynq. Il exploite au mieux les compétences acquises par les étudiants durant les mois précédents : électronique numérique, architecture des μC 32 bit, programmation en VHDL,

programmation en langage C, protocoles de communication, etc. Il est évident qu'un tel projet serait difficile à mener sans les bases acquises auparavant par les étudiants.

Nous fournissons aux étudiants des sujets bien documentés afin de guider au mieux leur travail sur les différentes méthodologies mises en application. Aussi, grâce à la maturité technique acquise par les étudiants, la bonne structuration du projet, et aux rappels de cours permanents de l'enseignant, les confusions courantes entre les langages C et VHDL sont évitées : L'interaction de ces deux langages est désormais implicite dans la conception des projets logiciels/matériels.

Le retour des étudiants est très positif puisque concret. Ils apprécient notamment la phase finale dans laquelle ils peuvent visualiser une mesure de fréquence sur leur téléphone portable.

Ce projet nécessite encore quelques ajustements et plusieurs améliorations pourraient être apportées notamment sur le développement du site local en utilisant les langages HTML/CSS et JavaScript. De même, nous pourrions aussi envisager d'ajouter un capteur de luminosité ou de couleur dont l'information correspond à une fréquence (capteurs TAOS) : Cela ajouterait une envergure technologique supplémentaire en optoélectronique.

Remerciements

Ce travail a été réalisé avec le support du GIP-CNFM: Groupement d'Intérêt Public - Coordination Nationale pour la formation en Microélectronique et en nanotechnologies et ses Services nationaux de logiciels, et le programme d'investissement d'avenir dans le cadre du projet IDEFI-FINMINA : Initiative d'Excellence - Formation Innovante en Microélectronique et Nanotechnologies, ANR-11-IDFI-001.

Références

1. O. Wyman, *Internet des objets, les business models remis en cause*, Marsh and McLennan Companies, p. 109 (2015)
2. LP COnsEPT au GEII Marseille : Website : <http://iut.univ-amu.fr/diplomes/licence-professionnelle-conception-systemes-electroniques-optoelectroniques-consept>
3. L. H. Crockett, R. A. Elliot, M. A. Enderwitz and R. W. Stewart, *The Zynq Book: Embedded Processing with the ARM Cortex-A9 on the Xilinx Zynq-7000 All Programmable SoC*, First Edition, Strathclyde Academic Media, (2014).
4. Carte Zybo : Website : <https://reference.digilentinc.com/reference/programmable-logic/zybo/start>
5. F. Aubépart, M. Moulin, M. Gauch, S. Favard, *Conception de systèmes numériques pour les circuits. Expériences pédagogiques de la licence EISI, mention microélectronique de l'IUT de Marseille*, Actes des 9^{ième} Journées Pédagogiques de la Coordination Nationale pour la Formation à la Microélectronique et aux nanotechnologies (JPCNFM'06), Saint-Malo (France), 22-24 Novembre 2006, pp. 163-168.
6. Ouvrage collectif, *Livre Blanc - Les Nouveaux Eldorados de l'économie connectée*, ed. Renaissance numérique et institut G9plus, p. 54 (2013).
7. J. Fuan, J. Kaiser, *Fréquence-mètre numérique à constantes de temps variables et en circuits intégrés MECL*, Rapport CEA-R-3507, Service Central de documentation du CEA, p. 36 (1968).
8. S. Moutault, J. Weber, M. Meaudre, *Le langage VHDL – Du langage au circuit, du circuit au langage*, ed. Dunod, p. 276 (2011).
9. P. Simpson, *La conception de systèmes avec FPGA – Bonnes pratiques pour le développement collaboratif*, ed. Dunod, p. 296 (2014).
10. S. Garg, A. Kroh, *Creating AXI-Lite 'Custom IP' in Vivado, lab for COMP4601*, p. 32.
11. <http://www.cse.unsw.edu.au/~cs4601/16s1/labs/custom-ip-lab.pdf>
12. N. Kolban, *Kolban's book on ESP8266*, p. 338 (2016).
13. Les Commandes AT : Website : http://www.technologuepro.com/gsm/commande_at.htm
14. P. André, *La liaison RS232*, éditions techniques et scientifiques françaises, p. 238 (2002).