

AUTOMATISATION ÉTUVE RAGT SEMENCES

RAPPORT DE PROJET BTS SE 2015



ESCAMEZ Michel
Lycée Charles Carnus
BTS SE2 2014 – 2015



Sommaire

Remerciements	1
Présentation du candidat	2
Introduction	3
I - Cahier des charges	
1. L'entreprise RAGT Semences	5
Présentation de l'entreprise	5
Quelques chiffres de l'entreprise	5
2. Présentation du projet	6
Objectif du projet	6
Étude de l'existant	7
Les objectifs recherchés	8
Gestion des tâches	8
Mon rôle dans ce projet	9
II - Déroulement du projet	
1. Phase de recherche	11
Informations à transmettre	11
Choix technologique	12
Étude du module USR-WIFI232-T	17
Manipulation envisagée	19
2. Phase de développement	20
Choix des composants	20
Choix des broches du PSoC	20
Réalisation des schémas	20
Réalisation des PCB	23
Programmation Carte HF et Télécommande	25
Conclusion Générale	28
Annexes	29
Glossaire des Acronymes	29
1. Organisation du thème 2015	30
2. Programmation carte HF et Télécommande	31

Remerciements

Dans un premier temps je tiens à remercier tout particulièrement nos enseignants techniques, Bernard Pouget, Hervé Dijols et Joël Marot, qui nous ont beaucoup aidés durant le déroulement de notre projet. Je les remercie également pour tous leurs investissements personnels et les innombrables heures passées en dehors des cours pour la recherche et la résolution de nos problèmes techniques.

Ensuite, je tiens également à remercier le groupe RAGT Semences sans qui ce merveilleux projet n'aurait jamais eu lieu mais aussi pour la confiance qu'ils nous ont accordée pour la réalisation de ce projet.

Pour finir, je remercie notre professeur de français Catherine Rey pour l'aide qu'elle m'a apporté lors de la correction de ce rapport de projet.



Présentation du candidat

Étudiant en électronique

Nom : ESCAMEZ

Prénom : Michel Jean Angelo

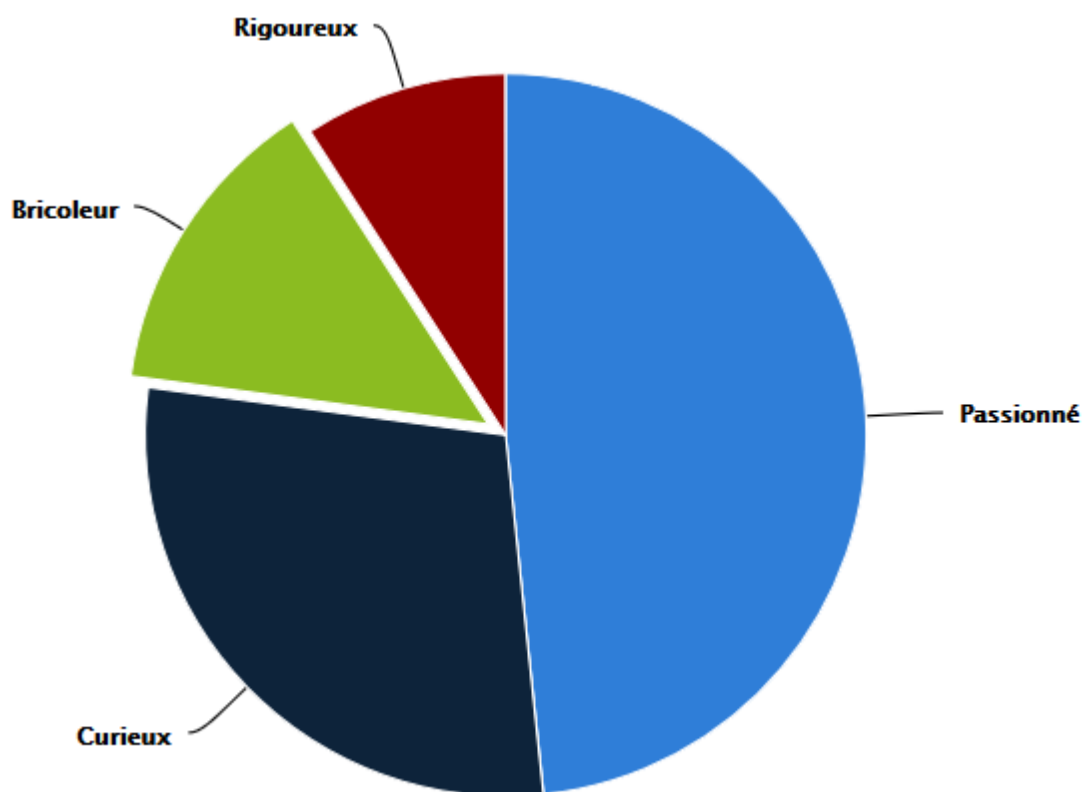
Date de naissance : 16 Novembre 1994

Lieu de naissance : BEZIERS

Adresse : VILLEFRANCHE DE PANAT

Scolarité : BTS SE1

Établissement : CHARLES CARNUS



Introduction

Dans le cadre de notre formation au sein du BTS Système Électronique du lycée Charles Carnus de Rodez, un projet électronique est effectué par notre promotion afin de mettre en pratique nos connaissances acquises au cours de notre formation.

Nous sommes en partenariat avec l'entreprise RAGT Semences pour réaliser une solution électronique de séchage de semences dans des étuves. Ce projet doit respecter un cahier des charges défini à l'avance par l'entreprise.

L'objectif de ce rapport est de décrire le travail que j'ai effectué durant ces cinq mois consacrés au projet.

Ce rapport de projet est constitué de trois parties. La première est une présentation du projet et de ses objectifs. La deuxième partie présente le travail effectué. Enfin, la dernière partie détaille les outils et méthodes utilisés pour mener à bien ce projet.



I - Première Partie

Cahier des charges du projet



1. L'entreprise RAGT Semences

Présentation de l'entreprise

Le groupe **RAGT Semences** est une société de production de **semences** et de conseil et commerce autour de l'agriculture. Le siège de l'entreprise se situe à **Rodez**. **RAGT Semences** a été créé en **1919** par des agriculteurs aveyronnais, aujourd'hui partiellement actionnaires.



Quelques chiffres de l'entreprise

RAGT Semences consacre **14%** de son chiffre d'affaire pour la recherche et le développement de nouvelles semences agricoles. Aujourd'hui, **RAGT Semences** réalise la majorité (**53%**) de son chiffre d'affaire à l'étranger. Ce dernier s'élève à plus de **235** millions d'euros. L'entreprise compte **17** filiales commerciales dans le monde avec plus de **1000** salariés.



2. Présentation du projet

Objectif du projet

Le but de ce projet de fin d'année d'études de **BTS SE** est la reprise d'un système existant. Nous avons un **contrat** avec la **RAGT Semences** pour finaliser un **système** de **mesures** et de **transmission** de paramètres physiques pour l'**automatisation** d'**étuves** de **séchages** pour différentes semences. Les objectifs de ce projet sont dans un premier temps de **fiabiliser** le premier prototype du module de gestion de l'étuve et dans un second temps la **modernisation** des solutions structurelles par le remplacement d'un **PIC** par un **PSOC**. L'électronique intervient dans ce milieu agricole pour répondre au mieux aux besoins suivants :

- Respecter un **temps de séchage précis** ;
- Respecter un **taux d'humidité** et une **température précise** dans l'étuve ;
- L'envoi de programmes de séchages à l'étuve avec une **transmission sans fil** ;
- Divers programmes de séchages pour diverses semences.

Nous devons réaliser plusieurs cartes électroniques afin d'envoyer les programmes de séchage à l'aide d'une télécommande et pour contrôler des actionneurs placés sur l'étuve de séchage. A la fin de notre projet, il sera primordial de **fournir** tous les **documents** nécessaires pour une compréhension intégrale de notre système par les techniciens de **RAGT Semences**. Le but pour l'entreprise RAGT Semences est de **produire** de **nombreuses** cartes basées sur notre projet. C'est pour cela qu'il est impératif de documenter au mieux nos recherches et de fournir un **travail fiable** et **reproductible** par l'entreprise.



Étude de l'existant

Comme nous l'avons vu précédemment, ce projet est la **reprise** d'un système existant. Ce système est composé d'un système de mesure d'archivage et d'alerte « **Testo Saveris** » avec des sondes radio **868MHz** pour la transmission sans fil. Les étuves de séchage sont équipées d'un **clapet électronique** et d'une carte avec un **PIC** ainsi qu'une **télécommande** pour la transmission filaire des programmes. Les serres de **RAGT Semences** sont aussi équipées d'une **station météo** composée d'un **anémomètre** et d'une **girouette** avec un **pluviomètre**.



Figure 1 et 2 : Station météo avec girouette, anémomètre et pluviomètre

Le lycée **Charles CARNUS** a déjà eu plusieurs contrats avec **RAGT Semences** en 2010, 2012, 2014 et maintenant en 2015 pour intervenir sur le système d'automatisation d'étuve de séchage. Le thème du projet pour le **BTS SE 2010** était la mise en place d'un **système de mesures de paramètres** tels que l'**humidité** relative dans l'air, la **température**, le taux de **dioxyde de carbone** (CO₂) et le taux de **monoxyde de carbone** (CO). Le thème de l'année **2012** était la mesure de **pluviométrie** dans la station météo. Pour l'année **2014**, le thème portait sur le **contrôle** et la **gestion** du fonctionnement de l'étuve de séchage avec la mise en place de capteurs d'**humidité** et de **température**.

Nous disposons pour ce **projet** des multiples **ressources** du thème de l'année précédente. Parmi ces ressources, les différents **schémas électroniques** ainsi que certains exemples de **programmations** nous seront utiles.



Nous réutiliserons certains éléments comme le **clapet** d'extraction d'air qui sert à réduire la consommation électrique, un **afficheur LCD 20 x 4** et divers éléments. Nous allons maintenant étudier les objectifs recherchés par l'entreprise **RAGT Semences** et le rôle de chaque **équipe** dans le déroulement du **projet**.

Les objectifs recherchés

Les objectifs recherchés par **RAGT Semences** sont premièrement de répondre au mieux aux **besoins** exprimés par les **utilisateurs** qui sont les techniciens, les gestionnaires et les chercheurs.

Deuxièmement il faudra **finaliser** la réalisation de **prototypes** permettant de s'adapter aux contraintes de fonctionnement.

Pour finir il sera important de **produire** les **documents** d'études et de réalisation fiables et utilisables par **RAGT Semences**.

Gestion des tâches

Le thème de cette année s'organise en **5 équipes**. Nous allons voir le rôle de chaque équipe dans le déroulement du projet.

- **Équipe 1** : Liaison HF et architecture complète du code de la télécommande.
- **Équipe 2** : Réalisation de la télécommande avec encodeur rotatif et afficheur LCD.
- **Équipe 3** : Réalisation de la carte module HF avec lecture et écriture dans l'EEPROM en I²C.
- **Équipe 4** : Réalisation de la partie encodeur, commande relais, lecture et écriture en liaison I²C dans l'EEPROM.
- **Équipe 5** : Alimentation, relais résistances chauffantes et ventilateur avec interface DAC et ADC.

Un diagramme complet de l'organisation du thème 2015 est disponible en Annexe 1.



Mon rôle dans ce projet

Dans ce projet d'automatisation d'étuve de séchage, je suis **seul** pour gérer la **transmission sans fil** entre la **télécommande** et le **module HF**.

Dans un premier temps, je vais devoir étudier des solutions pour déterminer une **technologie** à utiliser pour réaliser cette **transmission sans fil**, le choix des broches à utiliser pour la fonction **UART** et le choix des broches pour les modules **PSoc5** de la télécommande et du module HF.

Dans un second temps je mettrai en œuvre le module HF que j'aurai choisi et j'effectuerai les **configurations** nécessaires avec les modules **PSoc5**.

Enfin, j'aurai la responsabilité de la **structure générale** du **programme** du pupitre standard HF aussi appelé télécommande.



II - Deuxième Partie

Déroulement du projet



1. Phase de recherche

Informations à transmettre

Les données à transmettre proviennent principalement de la télécommande et transitent vers le module HF à l'aide de deux modules **WiFi** vu précédemment. Ces données sont principalement des programmes destinés à l'étuve de séchage pour les entrées dans un mémoire **EEPROM**. Par la suite le technicien opérant sur l'étuve aura la possibilité de sélectionner plusieurs programmes de séchage en fonction du produit dans l'étuve. Le détail des données à transmettre est représenté ci-dessous.

- N° Programme : de **00** à **99** -> **1** octet ;
- T° à consigner : de **000** à **250°C** -> **1** octet (1 bit par degré) ;
- Nom du produit : **10 caractères** -> **10** octets ;
- Temps Max de séchage (en heures) : de **00** à **99** -> **1** octet.


Nous obtenons donc un mot de **13 octets**. Cette transmission de données n'est pas unidirectionnelle. En effet, le technicien a la possibilité de modifier un programme à l'aide de la télécommande. La transmission de données est donc bidirectionnelle. La télécommande envoie une requête au module HF en demandant le programme avec un certain numéro de programme. En réponse à cette requête, le module HF envoie à la télécommande le bon programme.

Les programmes de séchage ne sont pas les seules données à transmettre au module HF. En effet, j'ai dans l'idée de valider la connexion entre les deux modules **WiFi** par la transmission de caractères au lancement des modules. Le but est de valider la communication entre les deux modules et d'afficher sur l'écran de la télécommande « *Connexion Établie* ».

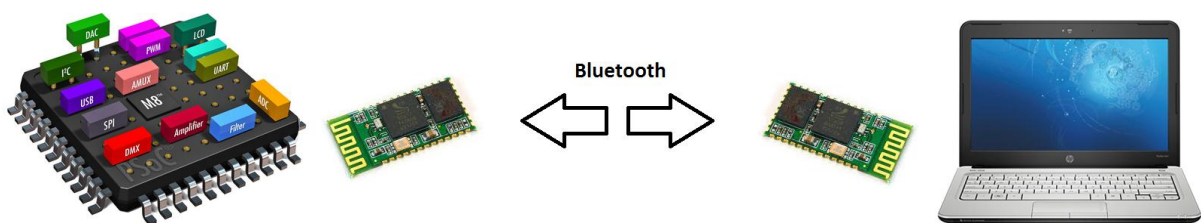


Choix technologique

Lors de la première phase de recherche, j'ai eu à déterminer quelle serait la meilleure technologie à employer pour réaliser la transmissions des données entre la télécommande et la carte module HF. Cette transmission doit être fiable, de portée suffisante et simple à réaliser pour permettre la reproduction des prototypes par l'entreprise **RAGT Semences**.

Dans un premier temps je me suis dirigé vers la technologie **Bluetooth**. Ce dernier est un standard de  **Bluetooth**® communication permettant l'échange bidirectionnel de données à très courte distance et utilisant des ondes radio **UHF** (de **300MHz** à **3GHz**). Dans le cas suivant, j'ai utilisé un module avec une fréquence de **2.4GHz** et la norme **IEEE 802.15.1**.

J'ai réalisé une connexion sans fil entre le module **UART** du **PSoC** et mon ordinateur portable à l'aide de deux modules **Bluetooth HC-05**.

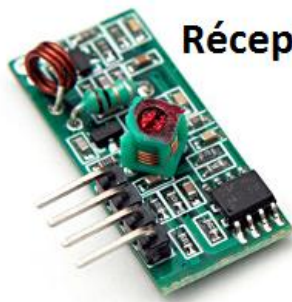


Dans un périmètre de quelques mètres, la communication est établie sans aucun problème et avec une vitesse de transmission de **115200** baud. Dans la documentation du module **Bluetooth**, le constructeur indique que la portée maximale est d'environ **40** mètres, mais lors de mes tests, la transmission coupe à environ **20** mètres sans obstacles et à presque **10** mètres avec des obstacles et des cloisons.

La facilité de configuration et le faible prix de ce module fait de cette technologie un atout important pour la réalisation de notre projet mais la faible portée de communication m'oblige à réfléchir à une nouvelle technologie de transmission sans fil.



Dans la continuité de ma démarche de recherche d'une technologie de communication haute fréquence, j'ai réfléchi sur un module de transmission de données sur une fréquence de **433MHz**. Ce module est composé d'une carte émettrice et d'une autre réceptrice comme représenter ci-dessous.



Récepteur

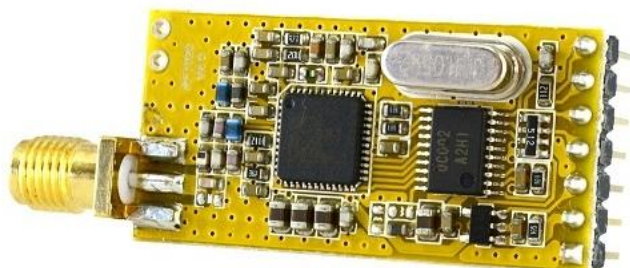


Émetteur

Le cahier des charges indique que la transmission de données se fait sur les deux cartes **PSoC** et dans les deux sens. C'est à dire que sur chacune de ces deux cartes doivent être présents les deux modules radio, émetteur et récepteur, pour assurer cette transmission bidirectionnelle. Ces modules sont miniatures avec une dimension de **30 x 14 x 7mm** pour le récepteur et de **19 x 19mm** pour l'émetteur. Mais il faut rajouter une antenne par module pour assurer la portée de la transmission. La longueur de cette dernière se calcule de la façon suivante :

- $f = 433\text{MHz}$ (fréquence du module)
- $c = 3.10^8 \text{ m.s}^{-1}$ (célérité de la lumière dans l'air)
- d'où la longueur d'onde $\lambda = \frac{c}{f}$ soit 0.69284 m
- soit pour une antenne quart d'onde $l = \frac{\lambda}{4} = 17.32 \text{ cm}$.

Deux antennes de **17.32cm** par carte rendraient la réalisation trop compliquée pour que l'ensemble s'insère correctement dans un boîtier. Il existe aussi un module radio **433MHz** qui fait à la fois émetteur et récepteur mais ce dernier à une dimension trop importante pour être implanté dans le boîtier de la carte **PSoC**. Une photo de ce dernier est présente ci-dessous.



La technologie que j'envisage d'implanter dans la télécommande et dans la carte module HF est le **ZigBee**. Ce dernier est un protocole de haut niveau permettant la communication de petites radios, à consommation réduite, basée sur la norme **IEEE 802.15.4** et sur la fréquence de **2.4GHz**. Sur la photo ci-dessous est représenté le module **XBee Pro**.



- Puissance : **10 mW (+10 dBm)** ;
- Portée int/ext : **90 m / 1.6 km** ;
- Dimensions : **2.438 cm x 3.294 cm** ;
- Vitesse de transmission : **115.2 Kbps** ;
- Tension d'alimentation : **2.8 – 3.4V** ;

Ce module est disponible à l'achat sur le site **www.digikey.com** pour le prix de **45 €** sans les frais de port. Dans notre situation de projet de fin d'études il nous faudrait 6 modules **XBee Pro**, 2 modules pour les prototypes de la télécommande et de la carte HF et les 4 autres pour réaliser nos manipulations devant le jury lors de notre passage à l'oral. Le coût d'achat pour ces modules est d'environ **300 €**. Nous ne disposons pas d'un tel budget pour l'achat de la partie télécommunication sans fil. De plus, ce projet reste un **prototype**, ce qui veut dire que nous pouvons fournir une technologie de télécommunication identique mais avec un coût plus faible et si notre projet est accepté par l'entreprise **RAGT Semences**, nous leur fournirons toutes nos recherches et les améliorations possibles de ces prototypes pour un fonctionnement professionnel et optimal pour les étuves de séchages.

La technologie utilisée lors de ce projet est similaire à la technologie évoquée plus tôt. J'ai choisi le **WiFi** pour réaliser la transmission de données sans fil entre la télécommande et le module HF. Cette technologie utilise la fréquence **2.4GHz** et la norme **IEEE 802.11**. Le **WiFi** est un des protocoles de communication le plus utilisé dans le monde. Tous nos appareils mobiles et nos ordinateurs disposent d'une carte **WiFi** ce qui en fait un avantage pour l'utilisation de ce protocole dans notre projet de fin d'année.



En quelques mots le **WiFi** est un standard international possédant la norme **IEEE 802.11**. Le nom **Wi-Fi** est la contraction de *Wireless Fidelity*. La **Wi-Fi Alliance** est l'organisme chargé de maintenir l'interopérabilité entre les matériels répondant à la norme **802.11**. Ci-dessous est représenté le logo du protocole.



Le module utilisé dans ce projet est **USR-WIFI232-T** qui est fabriqué par le constructeur **USR IOT**. Ce dernier est spécialisé dans l'internet des objets et propose depuis plus de 6 ans des produits certifiés et fabriqués en **Chine**. Le module choisi fait partie d'une série de plusieurs produits à faible consommation d'énergie. Le rôle de ces derniers est de remplacer une connexion filaire série **RS232** par une transmission de données sans fil à l'aide d'une liaison **UART**. Ci-dessous est représenté le fonctionnement du module **USR-WIFI232-T**.

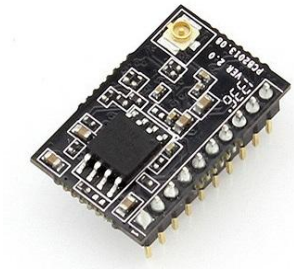


Ce module est plutôt récent dans le milieu des télécommunications **WiFi** et de l'internet des objets. J'ai entendu parler de ce module et de toutes les possibilités de configuration sur une communauté de l'internet des objets. Il existe de nombreux modules **WiFi** miniatures de ce style mais ce dernier est le seul à fournir une communication sans fil série à l'aide d'une connexion **UART** et avec une configuration minimale requise de quelques lignes seulement.

Le module est livré avec une antenne **PCB** miniature qui permet tout de même de fournir une transmission de données sur une portée de plus de **100m** comme il est indiqué dans la documentation du constructeur.



Le module **USR-WIFI232-T** est disponible sur le site du constructeur chinois **usriot.com** pour le prix de **7 €** environ. Le coût total pour l'achat de 6 modules **WiFi** est d'environ **50 €** avec les frais de port. Ce prix est tout à fait correct pour la réalisation de la transmission sans fil des prototypes. Les caractéristiques principales du module **WiFi** sont indiquées ci-dessous.



- Dimensions : **22 x 13.5 x 6mm** ;
- Tension d'alimentation : **3.3V** ;
- Interface **UART / PWM / GPIO** ;
- Standards : **IEEE 802.11b/g/n** ;
- Portée intérieur : **50 m**.

Ce module est idéal pour la réalisation de la transmission des données sans fil pour la télécommande. En effet, sa miniaturisation ainsi que la portée de transmission sont deux éléments convaincants pour le choix de ce module. D'une part les dimensions du module ne gêneront pas pour son implantation et d'autre part la portée de **50 m** assurée par le constructeur n'est pas négligeable dans notre projet où la télécommande ne sera peut être pas à côté de l'étuve de séchage lors de l'envoi de programmes.

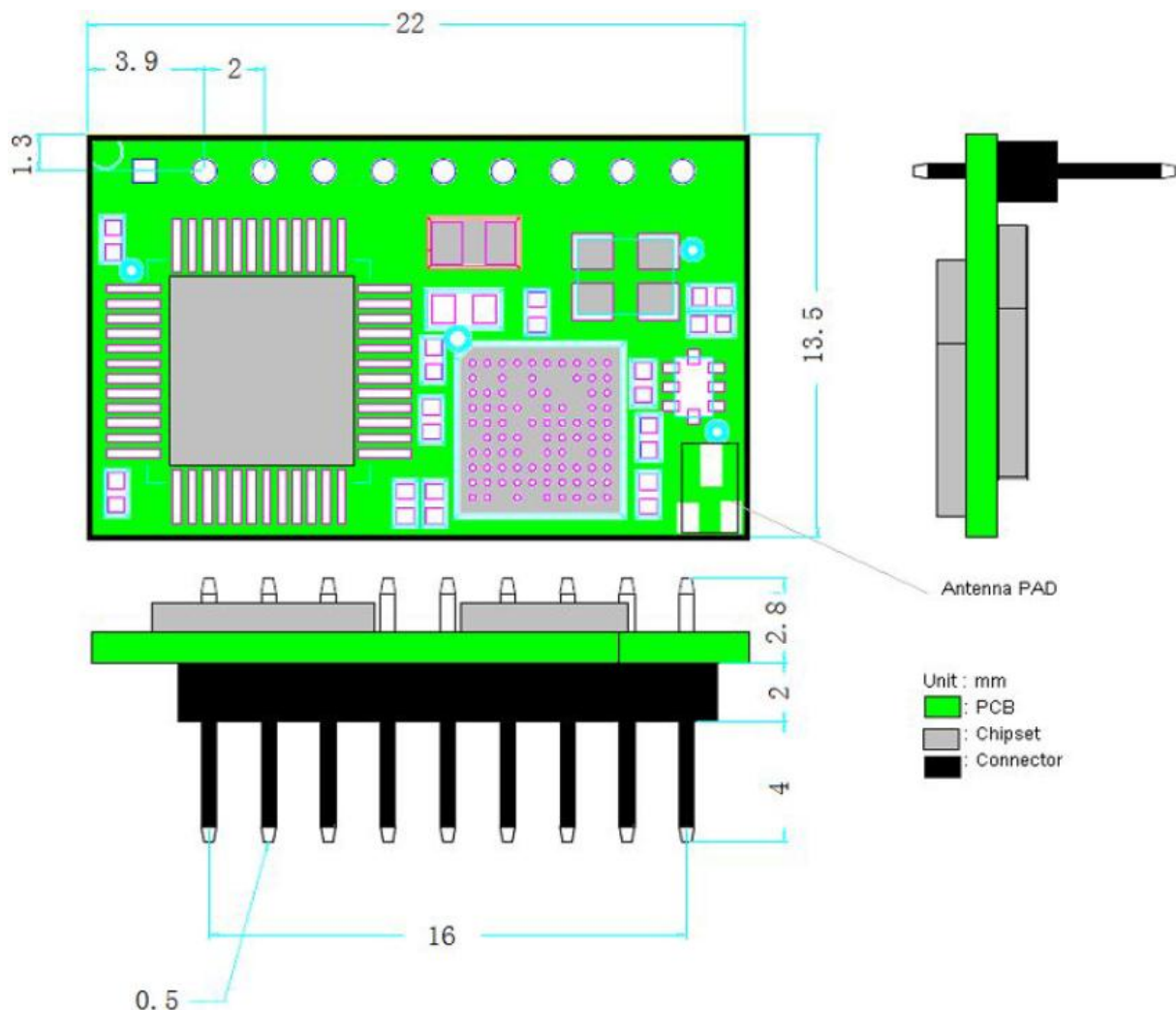
La figure ci-dessous montre les paramètres matériels du module **USR-WIFI232-T**.

Hardware Parameters	Data Interface	UART
		PWM, GPIO
	Operating Voltage	2.8~3.6V
	Operating Current	Peak [Continuous TX]: ~200mA Normal [WiFi ON/OFF, DTIM=100ms]: Average. ~12mA, Peak: 200mA Standby [WiFi Shutdown]: <200uA Power Down Switch: <10uA
	Operating Temp.	-40°C- 85°C
	Storage Temp.	-45°C- 125°C
	Dimensions and Size	22mm x 13.5mm x 6mm
	External Interface	1x10, 2mm DIP



Étude du module USR-WIFI232-T

Nous allons maintenant étudier les caractéristiques du module **USR-WIFI232-T** pour définir son implantation dans la conception de la télécommande et du module HF. La figure ci-dessous représente les dimensions du module en mm. Ce dernier est relativement petit ce qui facilite son implantation.



Ce module est utilisé dans de multiples domaines comme la télémétrie, la sécurité, le contrôle industriel, la domotique mais aussi dans les appareils médicaux. J'ai choisi ce module car il offre les mêmes fonctionnalités que la carte **XBee Pro** mais avec un prix dérisoire. Je vais détailler certaines fonctionnalités importantes qui m'ont fait choisir ce module en particulier.

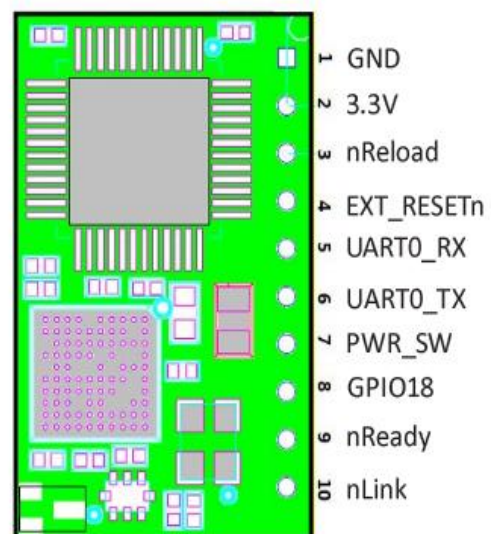


Pour commencer, le module **USR-WIFI232-T** dispose d'une fonction très intéressante qui est la transmission de donnée en mode transparent. Il s'agit en fait d'une communication port série virtuelle. C'est comme si nous avions un câble série mais sans fil. C'est cette principale fonction que je vais utiliser dans ce projet pour réaliser la communication entre la télécommande et le module HF.

Ensuite, ce module dispose de **10** broches avec un espacement de **2mm** entre chacune d'entre elles. Dans ces dernières sont comprises l'alimentation, la masse, une broche pour la restauration de configuration d'origine, une broche pour redémarrer le module, deux broches pour l'émission et la réception de données et quatre **GPIO** qui ont la fonction **PWM**. Ces quatre dernières broches sont intéressantes pour commander des relais ou pour faire de la domotique directement relié au **WiFi**. Je n'utiliserais pas ces **GPIO** dans le projet. La figure ci-dessous représente le nom des broches du module **USR-WIFI232-T**.

Ce module dispose aussi d'une interface web pour accéder plus facilement aux différentes configurations. Lors de l'alimentation du module un réseau WiFi apparaît avec comme **SSID** le modèle du module et sans sécurité. Il suffit alors de se connecter à ce réseau et d'accéder à l'adresse **10.10.100.254** sur le port **80**.

Ci-dessous sont représentées les caractéristiques électriques du module.



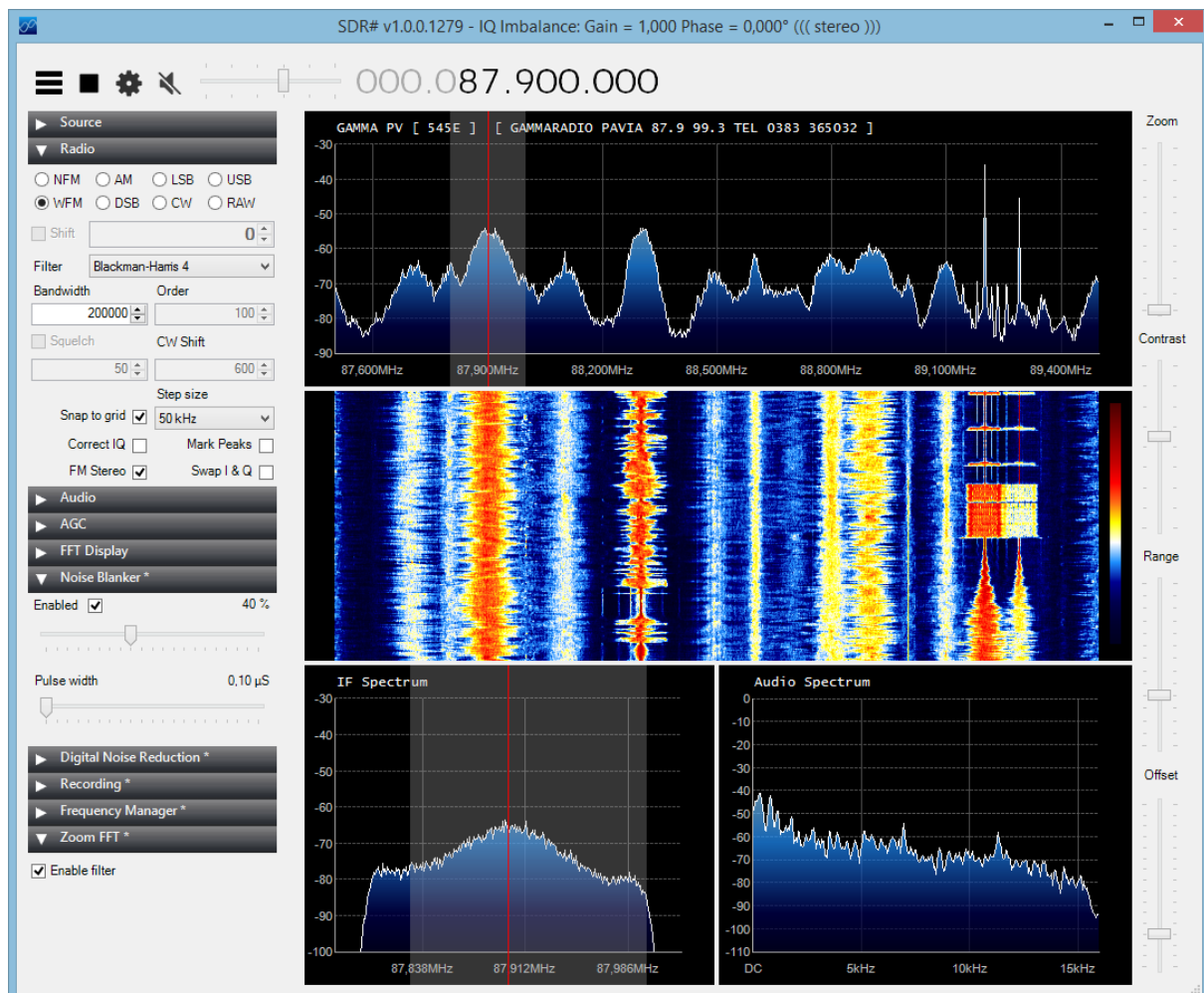
Parameter	Condition	Min.	Typ.	Max.	Unit
Operating Supply voltage		2.8	3.3	3.8	V
Supply current, peak	Continuous Tx		200		mA
Supply current, IEEE PS	DTIM=100ms		12		mA
Output high voltage	Sourcing 6mA	2.8			V
Output low voltage	Sinking 6mA			0.2	V
Input high voltage		2.2			V
Input low voltage				0.8	V



Manipulation envisagée

Pour mon oral de fin d'années d'études, j'ai une manipulation à faire en rapport au projet effectué. J'ai choisi de démontrer les effets d'un boîtier en métal sur la transmission de données. Pour cela je vais avoir besoin d'une boîte métallique dans laquelle je vais placer le module WiFi. Je vais tout d'abord calculer le gain dans des bonnes conditions, c'est-à-dire sans obstacles apparents. Ensuite, je vais calculer le gain dans cette boîte métallique.

Cette manipulation va servir à déterminer les influences d'une boîte avec un matériau qui ne laisse passer que très peu d'ondes électromagnétiques. Je vais utiliser un analyseur de spectre et un ordinateur équipé d'un module **SDR** qui me servira à étudier les influences du boîtier métallique. La figure ci-dessous est une capture du module **SDR** que je vais utiliser lors de ma manipulation. La capture d'écran montre une analyse de la bande **FM** à la fréquence **87.9 MHz**.



2. Phase de développement

Choix des composants

Après la phase de recherche, j'ai eu à choisir les composants électroniques à utiliser pour réaliser les deux cartes. Dans un premier temps je me suis occupé de la partie alimentation. J'ai choisi deux régulateurs **3.3V LD1086V33** pour l'alimentation des modules **WiFi**. Je me suis référé à la documentation de ce dernier pour déterminer les valeurs des condensateurs de régulation de tension qui sont de **10 μ F**. Les cartes télécommande et module HF sont équipées des mêmes composants pour m'a partie télécommunication.

Le coût total des composants pour ma partie télécommunication s'élève à environ **20€** en comptant les deux modules **USR-WIFI232-T**, les deux régulateurs **3.3V**, les condensateurs et les résistances de pull-up de **10K Ω** .

Choix des broches du PSoC

J'ai eu à déterminer les broches du **PSoC** à utiliser sur les cartes télécommande et module HF pour la réalisation de la transmission de données. J'ai choisi les mêmes broches pour la transmission **UART** sur les deux cartes.

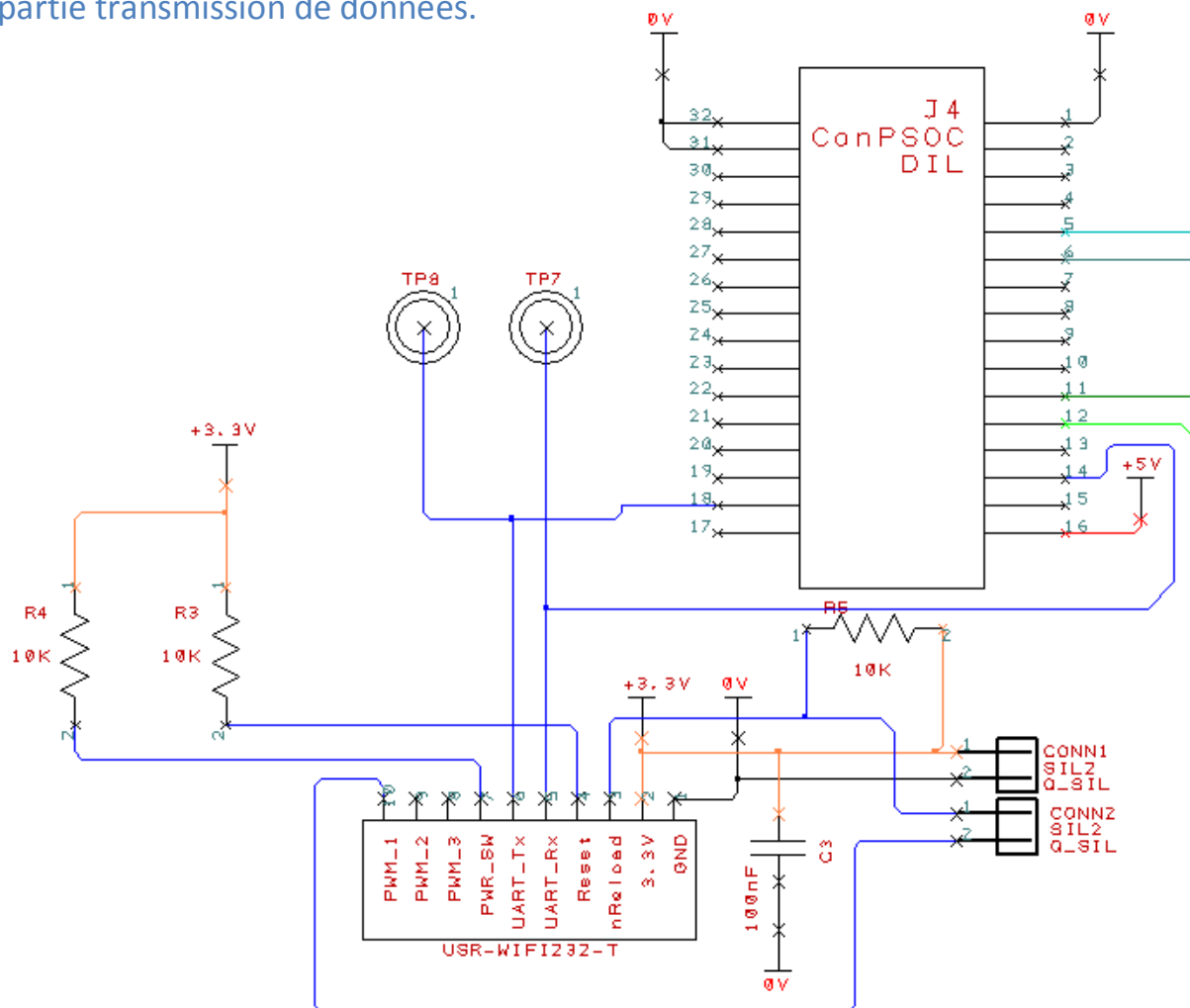
Fonction	Broche PSoC	Broche USR-WIFI232-T
Transmission	P3_7 (52) Tx	Pin 5 Rx
Réception	P3_5 (49) Rx	Pin 6 Tx

Réalisation des schémas

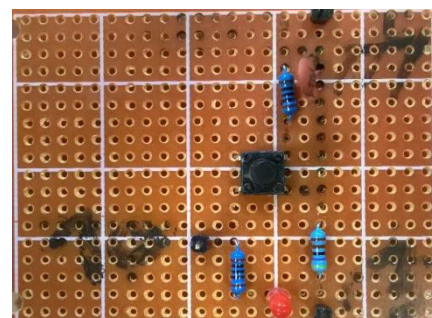
Pour réaliser les schémas des cartes de la télécommande et de la carte module HF nous avons utilisé le logiciel de **CAO DesignSpark PCB 7.0**. Ce dernier a été développé par les entreprises *RS Components* et *Allied Electronics*.



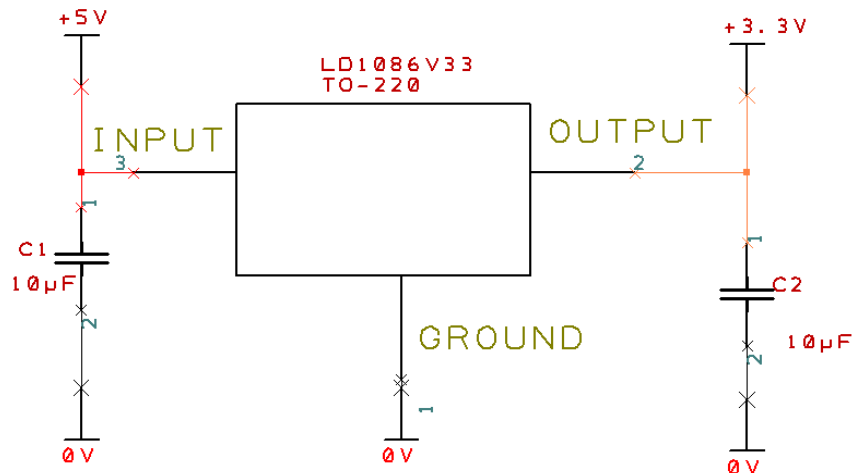
Premièrement je me suis référer à la documentation du module **USR-WIFI232-T** pour déterminer les broches à utiliser. Je vais en utiliser quatre pour l'alimentation et le module **UART**. Le reste des broches ne sont pas utilisées et sont en pull-up par défaut dans le module. La figure ci-dessous montre le schéma électronique de la télécommande et de la carte module HF pour la partie transmission de données.



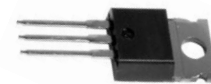
J'ai ajouté des points test pour vérifier le bon fonctionnement du module. Ces derniers indiquent si le module est bien connecté avec l'autre grâce à une LED rouge branchée sur une carte externe que j'ai conçue. Cette carte est représentée ci-dessous. Elle est constituée d'une LED rouge avec un bouton reset avec un système anti-rebond.



Je vais maintenant vous montrer le schéma pour la partie alimentation du module qui est composé d'un régulateur **LD1086V33** et de deux condensateurs de **10 μ F**.



La tension d'entrée provient d'un régulateur **5V**. Le boîtier du composant **LD1086V33** est un **TO-220**. Ce régulateur est linéaire.



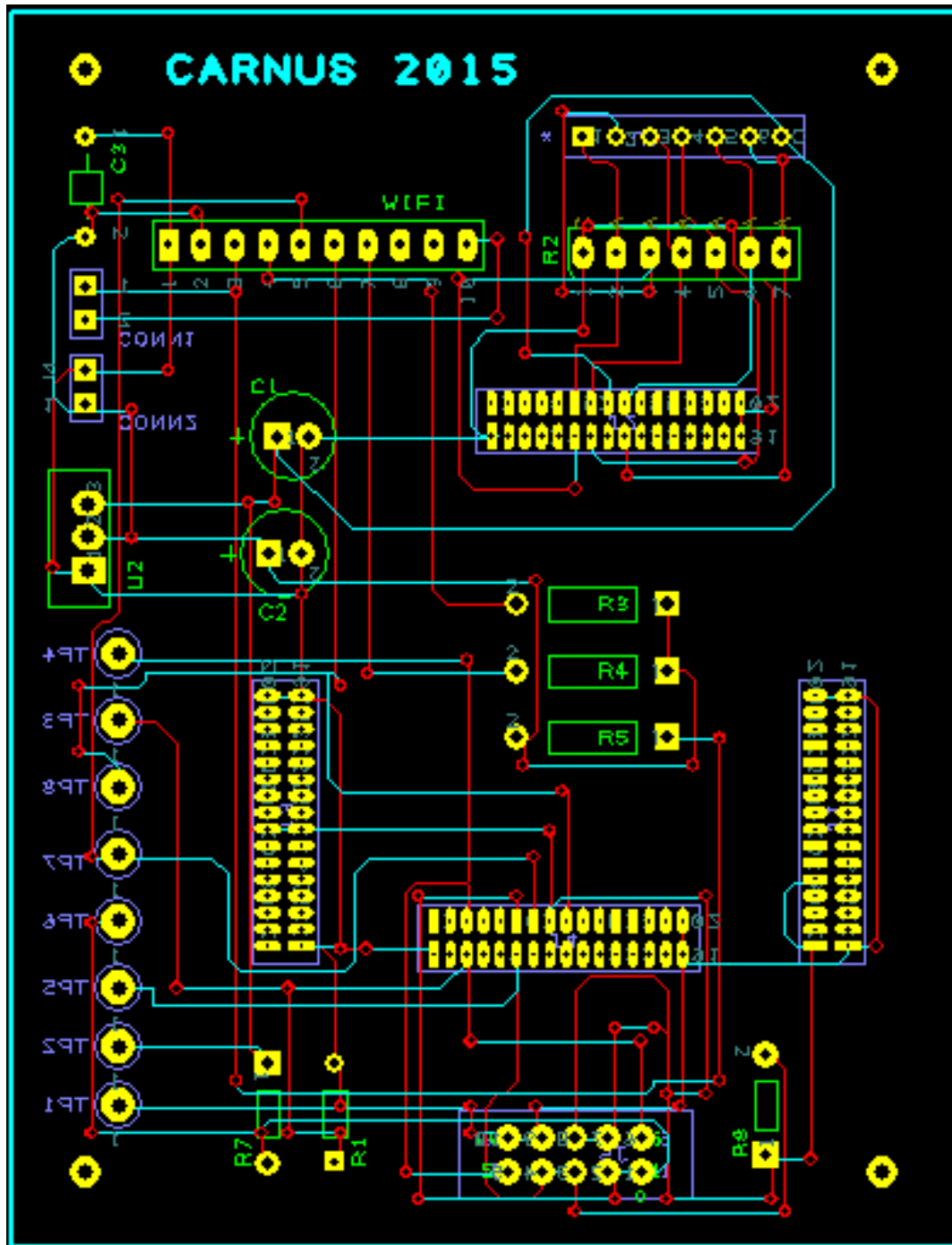
Le tableau ci-dessous montre les caractéristiques électriques du composant **LD1086V33**.

Symbol	Parameter	Test condition	Min.	Typ.	Max.	Unit
V_O	Output voltage (1)	$I_O = 0 \text{ mA}, T_J = 25^\circ\text{C}$	3.267	3.3	3.333	V
		$I_O = 0 \text{ to } 1.5 \text{ A}, V_I = 4.9 \text{ to } 30 \text{ V}$	3.234	3.3	3.366	V
ΔV_O	Line regulation	$I_O = 0 \text{ mA}, V_I = 4.9 \text{ to } 18 \text{ V}, T_J = 25^\circ\text{C}$		0.5	6	mV
		$I_O = 0 \text{ mA}, V_I = 4.9 \text{ to } 18 \text{ V}$		1	6	mV
ΔV_O	Load regulation	$I_O = 0 \text{ to } 1.5 \text{ A}, T_J = 25^\circ\text{C}$		1	10	mV
		$I_O = 0 \text{ to } 1.5 \text{ A}$		7	25	mV
V_d	Dropout voltage	$I_O = 1.5 \text{ A}$		1.3	1.5	V
I_q	Quiescent current	$V_I \leq 30 \text{ V}$		5	10	mA
I_{sc}	Short-circuit current	$V_I - V_O = 5 \text{ V}$	1.5	2		A
		$V_I - V_O = 25 \text{ V}$	0.05	0.2		A
	Thermal regulation	$T_A = 25^\circ\text{C}, 30 \text{ ms pulse}$		0.008	0.04	%/W
SVR	Supply voltage rejection	$f = 120 \text{ Hz}, C_O = 25 \mu\text{F}, I_O = 1.5 \text{ A}, V_I = 8.3 \pm 3 \text{ V}$	60	79		dB
eN	RMS output noise voltage (% of V_O)	$T_A = 25^\circ\text{C}, f = 10 \text{ Hz to } 10 \text{ kHz}$		0.003		%
S	Temperature stability			0.5		%
S	Long term stability	$T_A = 125^\circ\text{C}, 1000 \text{ Hrs}$		0.5		%

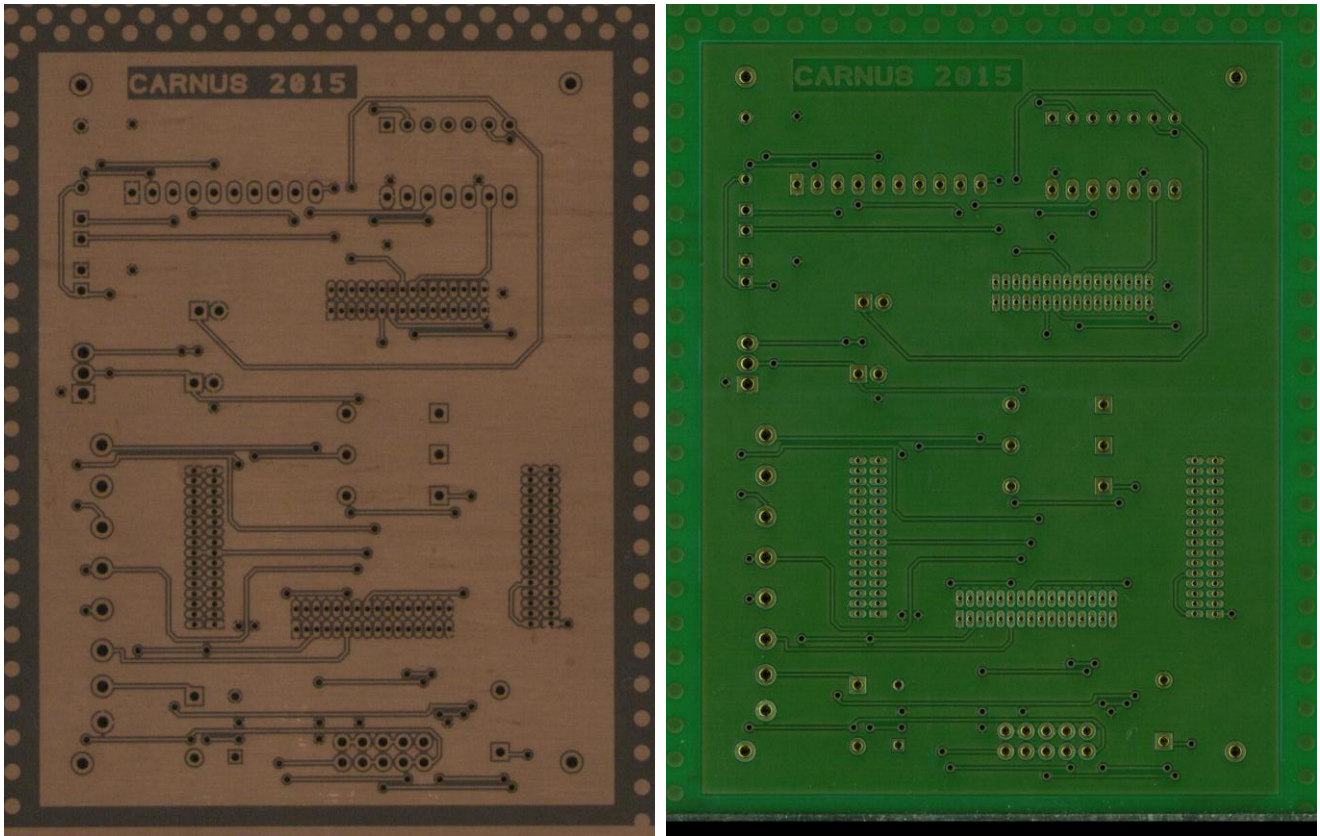


Réalisation des PCB

Nous avons réalisé le routage à l'aide du même logiciel utilisé pour les schémas électroniques. Le routage de la carte module HF est représenté ci-dessous. J'ai placé l'empreinte du module **USR-WIFI232-T** dans le coin gauche en haut pour que l'antenne ne traîne pas sur les composants.



Lorsque nous avons fini le routage, nous l'avons envoyé à une entreprise Irlandaise pour créer les cartes électroniques. Une quinzaine de jours plus tard ils nous ont retourné nos cartes **PCB** finis. Les figures ci-dessous représentent les cartes **PCB** en cours de traitement et les finitions avec le vernis.



Ces cartes **PCB** sont double couches en fibre de verre avec liant **époxy**. Les cartes mesurent **1.6 mm** d'épaisseur. Premièrement, la carte est percée sur du cuivre nu de **35 microns** d'épaisseur. Ensuite, à l'aide d'une solution chimique, le cuivre est supprimé pour laisser place aux pistes dessinées sur le schéma. L'avant dernière étape est la sérigraphie de la carte. Pour finir, la carte **PCB** est vernie puis découpée. Le résultat final est de très bonne qualité.



Programmation Carte HF et Télécommande

Le programme ci-dessous est la librairie **WiFi** que j'ai écrit pour configurer le module **USR-WIFI232-T** dans la carte HF.

```

/*****
* Nom Fichier: WIFI.c
*
* Fonctions: - Liaison série sans fil
*             - Mode point d'accès Wifi
*
* Connexion: Module GPIO5 (Rx)    -> PSoc P3[7] (Tx)
*             Module GPIO6 (Tx)    -> PSoc P3[5] (Rx)
*             Module GPIO1 (GND)   -> PSoc GND
*             Module GPIO2 (DVDD) -> PSoc 3.3V (max)
*
* Paramètres: - Bits per second    = 115200
*             - Data bits           = 8
*             - Parity               = None
*             - Stop Bits            = 1
*             - Flow Control         = None
*
* Informations: Voir document PDF "USR-WIFI232-T.pdf"
*
*****/
* CARNUS 2015, Escamez Michel - Soler Donovan - Verdier Jordan
*****/

/**** Déclarations & Directives ****/
#include "Application.h"

/**** Déclarations Variables ****/
char *AT[NUM_COM] = {
    "+++", //Entre en mode commande
    "a", //Valide la demande
    "AT+E", //Arrête l'affichage de la commande saisie
    "AT+UART=115200,8,1,None,NFC", //Définition des paramètres de transmissions
    "AT+WMODE=AP", //Mode point d'accès
    "AT+WAP=11BGN,RAGT_ETUVE,CH11", //Nom du réseau auquel le module se connecte
    "AT+WKEY=OPEN,NONE", //Paramètres de sécurité
    "AT+WADHCP=off", //Service DHCP = OFF
    "AT+LANN=10.10.100.254,255.255.255.0", //Assignment IP
    "AT+NETP=TCP,Server,8899,10.10.100.254", //Paramétrage Server TCP
    "AT+ENTM" //Entre en mode transmission invisible "Serial Wifi"
};

extern uint8 ConfigEnd; //La configuration du module est finie
extern uint8 Linked; //Client connecté ? lu oui; 0u non

uint8 command;

/**** Entrer en mode commande ****/
void ATModeCommande()
{
    UART_PutString(AT[0]);
    CyDelay(10);
    UART_PutString(AT[1]);
    CyDelay(500); }

```



```

/**** Configuration module AP ****/
void ConfigAP() { for (i = 2; i < NUM_COM; i++) { SendATCommand(AT[i]); }
}

/**** Envoi Commande UART ****/
void SendATCommand(char *at)
{
    UART_PutString(at);
    UART_PutCRLF(0x0d);
    CyDelay(100);
}

/**** Configuration Wifi ****/
void ConfigWifi()
{
    ATModeCommande();
    ConfigAP();
    CyDelay(1000);
    UART_ClearRxBuffer();
    ConfigEnd = 1u;
}

/* [] END OF FILE */

```

Je ne vais pas détailler ma programmation à l'écrit car je pense qu'elle est assez claire et lisible vu que je l'ai commentée. J'ai utilisé une interruption sur la broche P3_5 Rx du module **PSoC** pour détecter la réception d'une trame de données. Ce programme est représenté ci-dessous.

```

/*****
* Nom Fichier: Interruptions.c
*
* Fonctions: - Interruption UART Rx
*             - Interruption
*
*****/
* CARNUS 2015, Escamez Michel - Soler Donovan - Verdier Jordan
*****/

/**** Déclarations & Directives ****/
#include "Application.h"

/**** Déclarations Variables ****/
extern uint8 RxData;
extern uint8 ConfigEnd; //La configuration du module est finie
extern uint8 Linked; //Client connecté ? 1u oui; 0u non

uint8 inData[14];
uint8 inChar = -1;
uint8 indexx = 0;

```




```
/**** Interruption UART Rx ****/
CY_ISR(Rx_ISR)
{
    while (UART_GetByte())
    {
        if (indexx < 13)
        {
            inChar = UART_GetByte();
            inData[indexx] = inChar;
            indexx++;
            inData[indexx] = '\0';
        }
    }
}

/**** Initialisation Interruptions ****/
void Init_Interruptions()
{
    CYGlobalIntEnable;
    Rx_Int_Start();
    Rx_Int_SetVector(Rx_ISR);
}

/* [] END OF FILE */
```

Les codes source complet pour la programmation de la carte HF et de la télécommande sont disponibles sur mon **GitHub** à l'adresse suivant <https://github.com/Eih3/Projet-RAGT-2015> ou en flashant ce **QRCode** ci-dessous. La programmation est toujours en cours d'amélioration. C'est pour cela qu'il ne faudra pas entièrement se fier aux pages de codes présents ici et en annexe car elles peuvent être obsolètes lors de l'oral de projet.



Conclusion Générale

Ce projet de fin d'années d'études de BTS Systèmes Électroniques m'a permis de mettre en pratique mes connaissances acquises durant ces deux années d'étude.

Cette étude a également mis en évidence l'importance du travail en équipe dans la réalisation d'un projet pour une entreprise. De plus cette réalisation répond à un réel besoin de l'entreprise RAGT Semences et débouchera sur la fabrication d'un système de séchage avec télécommande sans fil pour des étuves déjà présentes dans l'entreprise. J'ai également pu prendre conscience des difficultés et des contraintes de la conception d'un produit industriel et j'ai pris un réel plaisir à rechercher des solutions et des améliorations nécessaires.

Au cours de ce projet, j'ai utilisé différentes technologies de communication comme le *Bluetooth* et le *WiFi* et j'ai aussi utilisé des logiciels de conception assisté par ordinateur tel que *DesignSpark* et *PSoC Creator 3.1*. Lors de la phase de développement j'ai rencontré certains problèmes matériels que nous avons résolus en équipe.

Pour conclure, cette dernière année de BTS SE a été fructueuse dans la découverte du fonctionnement de réalisation d'un projet électronique pour une entreprise à partir d'un cahier des charges. Ce fut également l'occasion de mettre en pratique mes connaissances personnelles en électronique et en programmation.



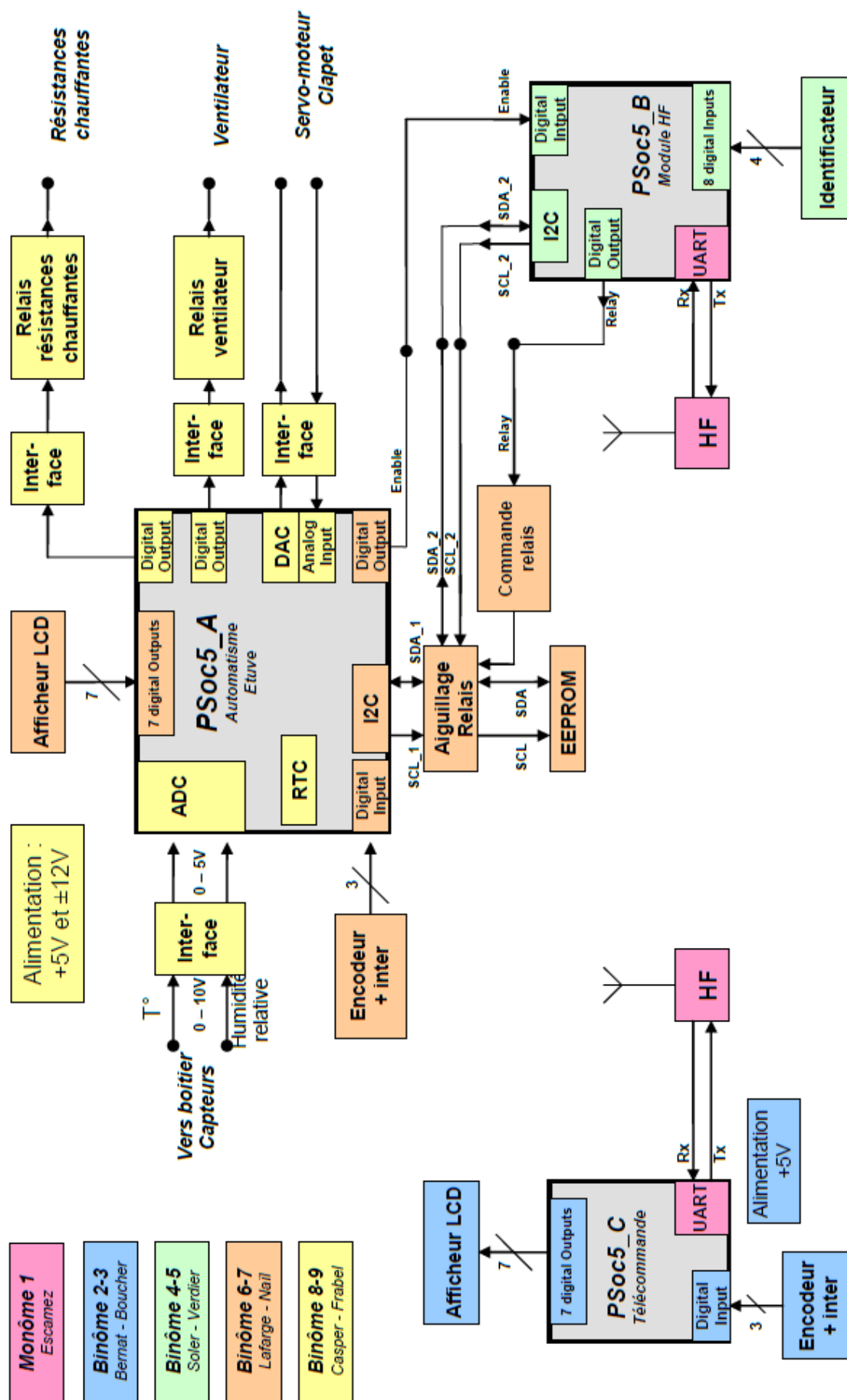
ANNEXES

Glossaire des Acronymes

EEPROM	Electrically-Erasable Programmable Read-Only Memory
Wi-Fi	Wireless Fidelity - ASFI en français (accès sans fil à l'internet)
IEEE	Institute of Electronics Engineers
PCB	Printed Circuit Board - Circuit imprimé
UART	Universal Asynchronous Receiver Transmitter
UHF	Ultra Hautes Fréquences de 300MHz à 3GHz
PIC	Peripheral Interface Controller
PSoC	Programmable System on Chip
GPIO	General Purpose Input / Output
PWM	Pulse Width Modulation
SSID	Service Set Identifier
CAO	Conception Assisté par Ordinateur
LED	Light-Emitting Diode
SDR	Software Defined Radio
FM	Frequency Modulation



Organisation Thème 2015



```
/******  
* Nom Fichier: main.c  
*  
* Programme: Carte Module HF  
*  
* Fonctions: - Liaison HF  
*             - Liaison I²C  
*             - Commande Relai  
*             - R/W EEPROM  
*             - AP Wifi  
*  
* Dernière MàJ: 11/05/2015  
*  
*****  
* CARNUS 2015, Escamez Michel - Soler Donovan - Verdier Jordan  
*****/  
  
/**** Déclarations & Directives ****/  
#include <Application.h>  
  
/**** Base du programme ****/  
int main() { Application(); }  
  
/* [] END OF FILE */
```



```

/*****
* Nom Fichier: Application.c
*
* Fonctions: - Lecture EEPROM
*             - Ecriture EEPROM
*
*****/
* CARNUS 2015, Escamez Michel - Soler Donovan - Verdier Jordan
*****/

/**** Déclarations & Directives ****/
#include <Application.h>

/**** Déclarations Variables ****/
uint8 i;
uint8 adresse[2] = {0x00};
uint8 dataBuffer[BUFFER_SIZE] = {0x00};
uint8 writeData[BUFFER_SIZE];
uint8 readData[BUFFER_SIZE];
uint8 RxData = 0x20;
uint8 ConfigEnd; //La configuration du module est finie
uint8 Linked; //Client connecté ? 1u oui; 0u non

/**** Fonction de base de l'application ****/
void Application()
{
    Initialisations();
    //Configurations();

    LCD_ClearDisplay();

    fonctionTest();

    i=33;

    for(;;)
    {
        if (!SW2_Read()) LCD_ClearDisplay();/*
        if (!SW3_Read()) UART_PutChar(0x40);

        if (Linked) wifiTest();
        else { LCD_Position(0, 0); LCD_PrintString("Attente de STA"); }
        */
        CyDelay(1000u);
    }
}

/**** Fonction Configurations ****/
void Configurations()
{
    LCD_PrintString("  Module HF");
    LCD_Position(1u, 1u);
    LCD_PrintString("Configuration");

    ConfigWifi();

    UART_ClearRxBuffer();
}

```



```

/**** Initialisations ****/
void Initialisations()
{
    Init_Interruptions();

    Init_I2C();

    Init_UART();

    Init_LCD();
}

/**** Initialisation LCD ****/
void Init_LCD()
{
    LCD_Start();
    LCD_PrintString("  Module AP");
    LCD_Position(1u, 1u);
    LCD_PrintString("Initialisation");
    CyDelay(1000);
    LCD_ClearDisplay();
}

/**** Initialisation UART ****/
void Init_UART()
{
    UART_Start();
    UART_ClearRxBuffer();
}

/**** Initialisation I2C ****/
void Init_I2C()
{
    I2C_Start();
    I2C_MasterClearStatus();
}

/**** Fonction Test Divers ****/
void fonctionTest()
{
    LCD_Position(0, 0);

    //writeData = 10 octets de 0 à 9
    for (i = 0; i < 13; i++) { writeData[i] = 0x30 + i; };

    /** Ecriture I2C 10 caractères **/
    //EcritureEEPROM(5, writeData, 13);

    /** Lecture I2C 10 caractères **/
    LectureEEPROM(3, readData, 5);

    //Affiche 10 caractères
    for (i = 0; i < 5; i++) { LCD_PutChar(readData[i]); };
}

void wifiTest()
{
    LCD_Position(0, 0);
    LCD_PrintString("Char rec: ");
    LCD_Position(0, 14);
    LCD_PrintString("AP");
}
```




```
LCD_Position(1, 0);  
LCD_PrintString("Char env: ");  
  
UART_PutChar(i);  
LCD_Position(1, 10);  
LCD_PutChar(i);  
i++;  
if(i==127) i=33;  
}  
  
/* [] END OF FILE */
```



```

/*****
* Nom Fichier: WIFI.c
*
* Fonctions: - Liaison série sans fil
*             - Mode point d'accès Wifi
*
* Connexion: Module GPIO5 (Rx)    -> PSoc P3[7] (Tx)
*             Module GPIO6 (Tx)    -> PSoc P3[5] (Rx)
*             Module GPIO1 (GND)   -> PSoc GND
*             Module GPIO2 (DVDD) -> PSoc 3.3V (max)
*
* Paramètres: - Bits per second    = 115200
*             - Data bits           = 8
*             - Parity               = None
*             - Stop Bits            = 1
*             - Flow Control        = None
*
* Informations: Voir document PDF "USR-WIFI232-T.pdf"
*
*****/
* CARNUS 2015, Escamez Michel - Soler Donovan - Verdier Jordan
*****/

/**** Déclarations & Directives ****/
#include "Application.h"

/**** Déclarations Variables ****/
char *AT[NUM_COM] = {
    "+++", //Entre en mode commande
    "a",   //Valide la demande
    "AT+E", //Arrête l'affichage de la commande saisie
    "AT+UART=115200,8,1,None,NFC", //Définition des paramètres de
transmissions
    "AT+WMODE=AP", //Mode point d'accès
    "AT+WAP=11BGN,RAGT_ETUVE,CH11", //Nom du réseau auquel le module se
connecte
    "AT+WKEY=OPEN,NONE", //Paramètres de sécurité
    "AT+WADHCP=off", //Service DHCP = OFF
    "AT+LANN=10.10.100.254,255.255.255.0", //Assignment IP
    "AT+NETP=TCP,Server,8899,10.10.100.254", //Paramétrage Server TCP
    "AT+ENTM" //Entre en mode transmission invisible "Serial Wifi"
};

extern uint8 ConfigEnd; //La configuration du module est finie
extern uint8 Linked; //Client connecté ? lu oui; 0u non

uint8 command;

/**** Entrer en mode commande ****/
void ATModeCommande()
{
    UART_PutString(AT[0]);
    CyDelay(10);
    UART_PutString(AT[1]);
    CyDelay(500);
}

/**** Configuration module AP ****/
void ConfigAP() { for (i = 2; i < NUM_COM; i++) { SendATCommand(AT[i]); }
}

```



```
/***** Envoi Commande UART *****/  
void SendATCommand(char *at)  
{  
    UART_PutString(at);  
    UART_PutCRLF(0x0d);  
    CyDelay(100);  
}  
  
/***** Configuration Wifi *****/  
void ConfigWifi()  
{  
    ATModeCommande();  
    ConfigAP();  
    CyDelay(1000);  
    UART_ClearRxBuffer();  
    ConfigEnd = 1u;  
}  
  
/***** Fonction Transmission TCP OK *****/  
void TCPOK()  
{  
    UART_PutChar(0x40);  
    LCD_ClearDisplay();  
    LCD_PrintString("Transmission TCP");  
    LCD_Position(1u, 7u);  
    LCD_PrintString("OK");  
    CyDelay(1000);  
    LCD_ClearDisplay();  
    Linked = 1u;  
    LCD_Position(1u, 14u);  
    LCD_PrintString("ok");  
}  
  
/* [] END OF FILE */
```



```
/* *****  
* Nom Fichier: Interruptions.c  
*  
* Fonctions: - Interruption UART Rx  
*             - Interruption  
*  
*****  
* CARNUS 2015, Escamez Michel - Soler Donovan - Verdier Jordan  
*****/  
  
/**** Déclarations & Directives ****/  
#include "Application.h"  
  
/**** Déclarations Variables ****/  
extern uint8 RxData;  
extern uint8 ConfigEnd; //La configuration du module est finie  
extern uint8 Linked; //Client connecté ? 1u oui; 0u non  
  
uint8 inData[14];  
uint8 inChar=-1;  
uint8 indexx = 0;  
/**** Interruption UART Rx ****/  
CY_ISR(Rx_ISR)  
{  
  
    while (UART_GetByte())  
    {  
        if (indexx < 13)  
        {  
            inChar = UART_GetByte();  
            inData[indexx] = inChar;  
            indexx++;  
            inData[indexx] = '\0';  
        }  
    }  
}  
  
/**** Initialisation Interruptions ****/  
void Init_Interruptions()  
{  
    CYGlobalIntEnable;  
    Rx_Int_Start();  
    Rx_Int_SetVector(Rx_ISR);  
}  
  
/* [] END OF FILE */
```



```
/* *****  
* Nom Fichier: Application.h  
*****  
* CARNUS 2015, Escamez Michel - Soler Donovan - Verdier Jordan  
***** */  
  
#if !defined(APPLICATION_H)  
  
#define APPLICATION_H  
  
/**** Déclarations & Directives ****/  
#include <project.h>  
#include "EEPROM.h"  
#include "WIFI.h"  
#include "Interruptions.h"  
  
#define BUFFER_SIZE      (254u)  
  
/**** Déclarations Fonctions ****/  
void Application();  
  
void Initialisations();  
  
void Configurations();  
  
void Init_LCD();  
  
void Init_UART();  
  
void Init_I2C();  
  
void Init_Interruptions();  
  
/* Temporaire */  
void fonctionTest();  
void wifiTest();  
  
#endif  
  
/* [] END OF FILE */
```



```
/* *****  
* Nom Fichier: WIFI.h  
* *****  
* CARNUS 2015, Escamez Michel - Soler Donovan - Verdier Jordan  
* *****/  
  
/**** Déclarations & Directives ****/  
#include "Application.h"  
  
#define NUM_COM          (11u)  
  
/**** Déclarations Fonctions ****/  
void ATModeCommande();  
  
void SendATCommand(char *at);  
  
void ConfigAP();  
  
void ConfigWifi();  
  
void TCPOK();  
  
/* [] END OF FILE */  
  
/* *****  
* Nom Fichier: Interruptions.h  
* *****  
* CARNUS 2015, Escamez Michel - Soler Donovan - Verdier Jordan  
* *****/  
  
/**** Déclarations & Directives ****/  
#include "Application.h"  
  
/* [] END OF FILE */
```

