

# PicoIP

Depuis peu, PicoIP est opérationnel sur le WEB au travers d'une application PHP installée sur un hébergeur ( Free dans mon cas ). L'accouchement a été long et douloureux mais enfin la plupart des problèmes sont résolus. Parmi ceux-ci on peut citer en vrac:

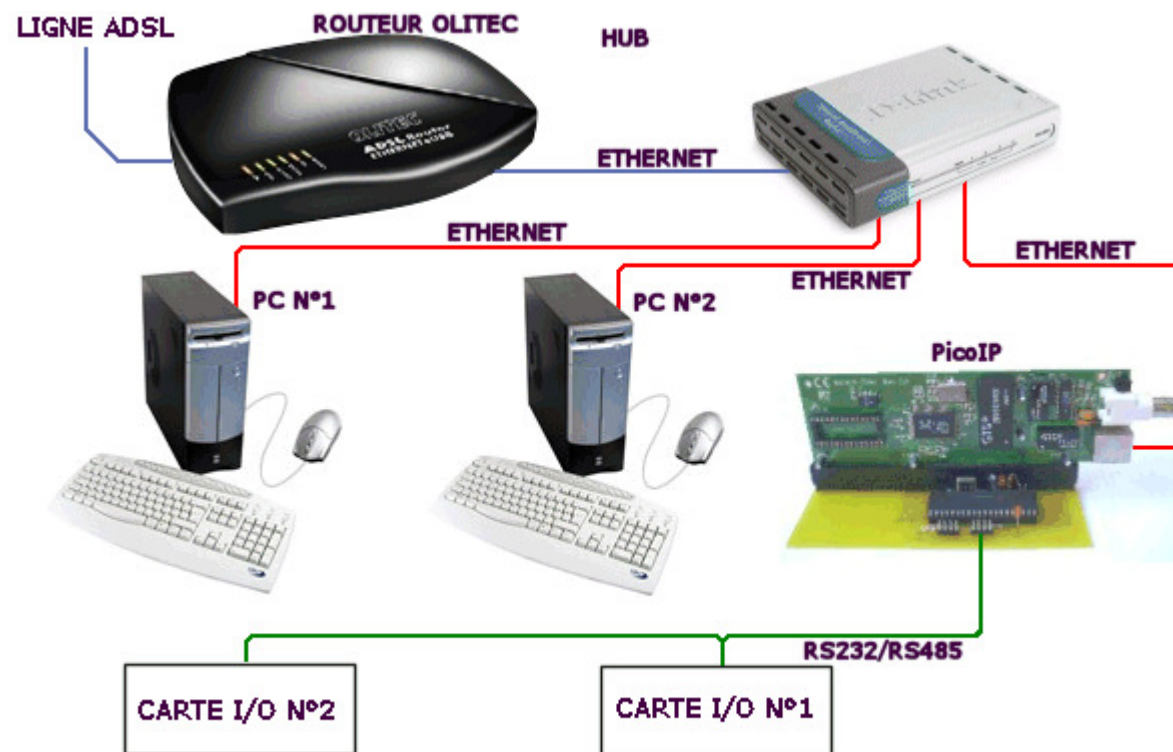
- la gestion d'un routeur ADSL,
- la résolution des adresses IP dynamiques ( comme sur Wanadoo par exemple ),
- l'interfaçage avec des modules externes.

Il reste beaucoup de travail à faire, heureusement vous êtes nombreux à vous être proposés pour résoudre tel ou tel problème. Si je n'ai pas sollicité votre bonne volonté tout de suite, c'est tout simplement parce que je voulais être sûr que le projet aboutirai, et, pour cela il était nécessaire de régler tous les problèmes de la partie réseau en premier. Comme c'est chose faite, l'évolution de ce projet est maintenant entre les mains de tous.

---

## **SYNOPTIQUE DU RESEAU UTILISE POUR LES TESTS**

Je vous propose de découvrir ce que j'ai mis en oeuvre pour faire fonctionner PicoIP. Il ne s'agit pas la d'un phénomène d'autosatisfaction, mais de décrire dans quelles conditions PicoIP fonctionne aujourd'hui. Il y a probablement des cas de figure que je n'ai pas envisagé, il faudra régler ces situations au jour le jour en étudiant chaque cas.



## COMMENT TOUT CELA FONCTIONNE ?

### EN LOCAL

En local, il y a deux possibilités de faire fonctionner PicoIP. Le but est bien sur de récupérer sur l'un des PC du réseau les informations des cartes d'entrées/sorties branchées sur le bus de PicoIP.

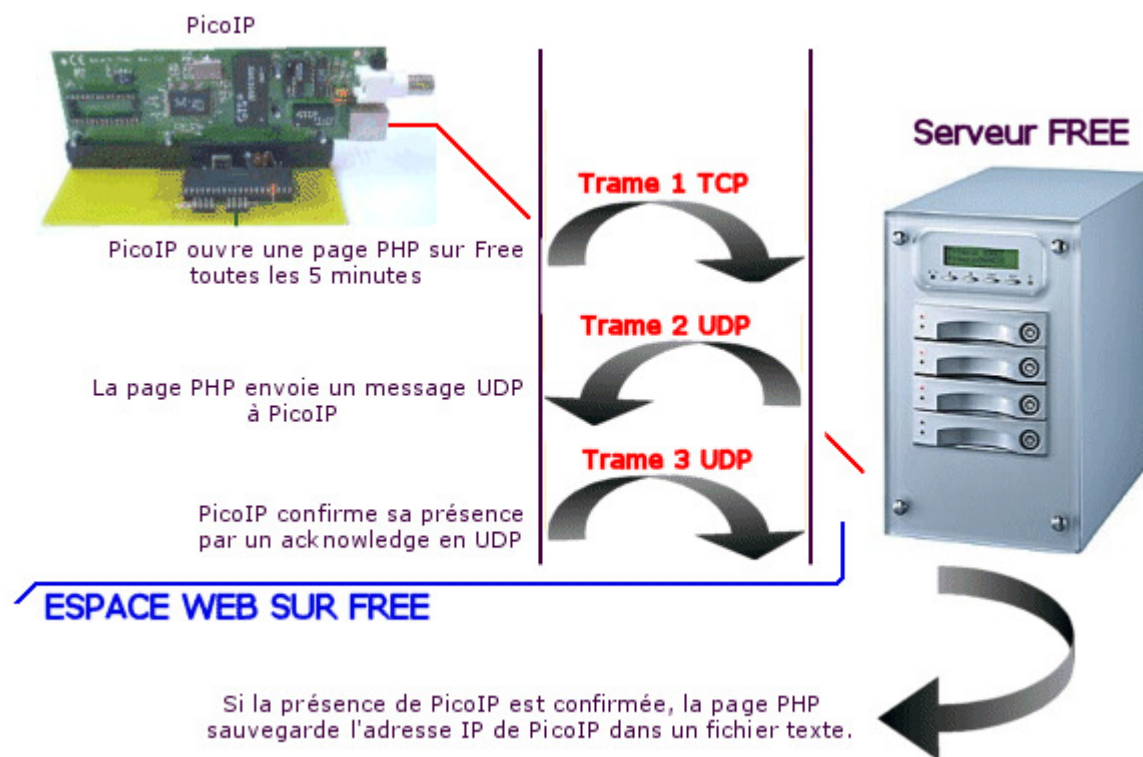
PicoIP dialogue avec l'extérieur en UDP et se comporte comme un port série virtuel qui envoie sur son UART les données qu'il reçoit du réseau.

Pour cela il faut dialoguer avec lui en protocole UDP. Pour y parvenir, soit on écrit une application en C ou en Delphi (c'est mon cas) qui envoie des trames à PicoIP et qui reçoit celle qu'il répond. Dans ce cas, on retrouvera sur l'UART en sortie les trames envoyées en UDP. Pour parvenir à cette configuration il est nécessaire de posséder un minimum de connaissances en programmation réseau.

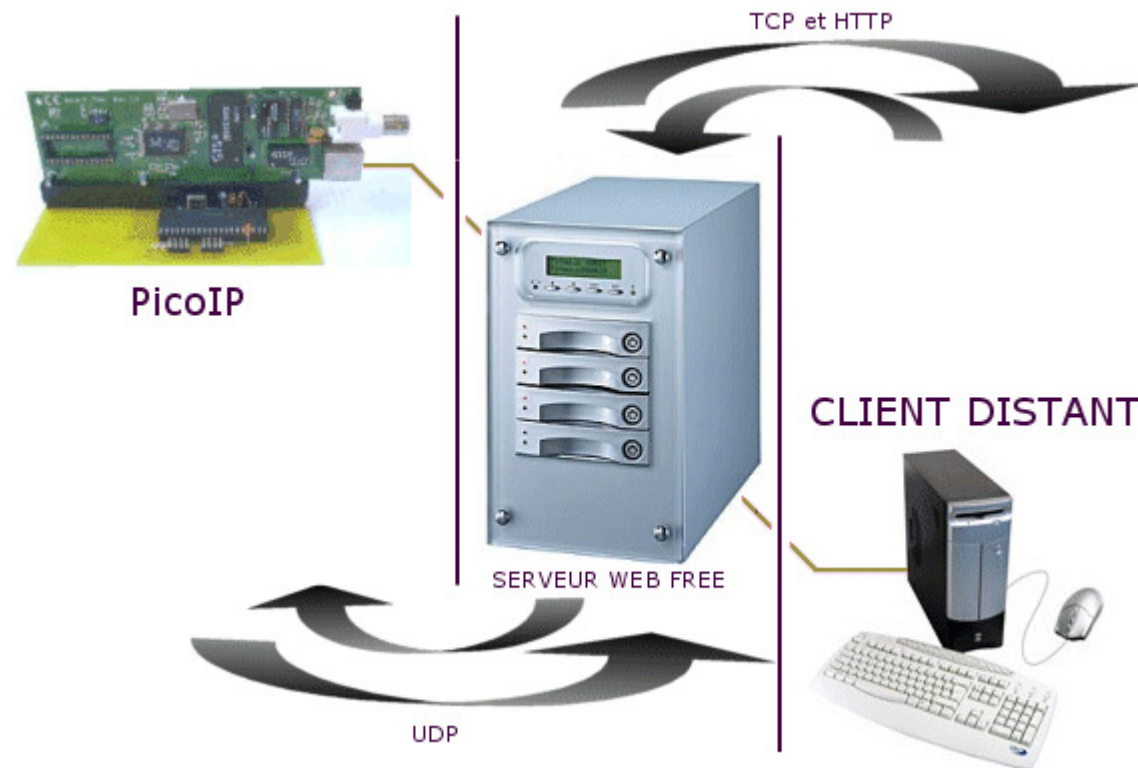
L'autre solution est d'écrire cette application en PHP. Pour cela il faut qu'un des PC du réseau soit un serveur PHP. Si cette solution est choisie, l'interface qui permet d'afficher les informations des cartes d'entrées/sorties pourra être réalisée en HTML et affichée sur n'importe quel navigateur. Je rassure ceux qui lèvent déjà les yeux aux ciel, installer un serveur PHP prend 5 minutes avec EasyPHP et PHP est un langage tout à fait abordable pour écrire de petites applications.

## A DISTANCE

En mode distant, une application sur un hébergeur ( Free dans mon cas ) permet de dialoguer avec PicoIP. Dans un premier temps PicoIP tente, toutes les 5 minutes d'ouvrir une page PHP sur free. Cette page récupère l'adresse IP qui l'a appelée et la stocke dans un fichier texte. Cette astuce permet de régler le problème de l'adresse IP dynamique:



Lorsqu'un client tente de se connecter, il charge une page PHP d'accueil et d'affichage des données. Cette page commence par ouvrir le fichier texte pour récupérer l'adresse IP courante de PicoIP. Les trames sont ensuite envoyées en UDP vers PicoIP. Un miroir parfait de ces trames se retrouve sur l'UART de PicoIP et est diffusé vers les périphériques. Lorsque les périphériques répondent, les trames sont mises en forme et envoyées vers le serveur. L'application PHP traite la réponse et la met en forme pour l'envoyer au client sur son navigateur.



## AVANTAGES ET INCONVENIENTS DU SYSTEME

### INCONVENIENTS

Je ne vois qu'un inconvénient à ce système c'est d'être tributaire du bon fonctionnement de son hébergeur pour accéder à PicoIP depuis n'importe quel endroit de la planète. Il faut donc le choisir pour la fiabilité de ses services et vérifier qu'il autorise le PHP avec la bibliothèque SOCKET installée.

Sur Free, malgré que PHP tourne en Safe Mode, les sockets fonctionnent sur le port 80. Il est donc possible de faire tourner des applications qui utilisent UDP.

### AVANTAGES

Le principal avantage de cette méthode est d'utiliser la puissance d'un serveur pour toute la mise en forme et la présentation des informations. En effet, aujourd'hui la plupart des applications embarquées à PIC intègrent un serveur WEB. Quand on connaît la taille d'une page HTML, on comprend que même un 16F877 avec 8k soit limité. Sans compter qu'il faut quand même un peu de place pour le code de gestion du réseau, environ 4k sur PicoIP.

**Dans le cas de PicoIP, pas de page lourde à stocker dans le PIC, tout est chez FREE.**

Un autre avantage est de permettre la gestion de l'IP dynamique de la plupart des serveurs. En effet, comme la plupart des internautes ont une adresse IP dynamique ( autrement dit qui peut changer une ou plusieurs fois par jour ), il est impossible de connaître de façon sûre l'adresse courante qui vous est affectée.

**Avec PicoIP, vous connaissez votre adresse IP courante** qui est stockée dans un fichier texte dans votre espace WEB.

Enfin PicoIP est utilisable par le plus grand nombre sans outil de développement. Le paramétrage de son adresse IP et de celle du routeur peut être faite par une application UDP par le réseau évitant l'emploi d'un programmeur, dans le futur **il sera même possible de reflasher PicoIP directement par le réseau** via une application PHP sur FreePic.

## QUE FAIRE MAINTENANT ?

Travaillant depuis deux ans sur ce projet, je commence seulement à entrevoir le bout du tunnel et déjà d'autres idées fleurissent. Il faut tout de même passer du stade de bidouille de coin de table à une réalisation semi-professionnelle. Pour cela, je pense qu'il faut:

- redessiner la carte de PicoIP en corrigeant les erreurs du départ sur l'I2C notamment et intégrer un convertisseur RS485,
- Créer une application qui permettra de paramétrer PicoIP (adresse IP, adresse passerelle, masque de sous réseau etc..),
- Créer des cartes d'entrées sorties conforme au protocole PicoIP,
- Créer des applications PHP standart en fonction des cartes d'entrées sorties qui permettent de créer des interfaces client à l'aide de simples copier/coller,
- Réécrire le code de PicoIP en bloc de 2k permettant l'utilisation de TurboCC5X,
- .....

Enfin et c'est probablement le plus important, il faut faire plein de PicoIP et les faire fonctionner pour corriger les bugs, détecter les limites du système et ainsi continuer une base de connaissances et de nouvelles versions.

## EN CONCLUSION

Mon objectif dans les prochains jours est de proposer une page PHP qui affichera les informations de la carte PicoIP abritée chez moi. Je pense afficher quelques températures et des informations tout ou rien récoltées dans mon installation électrique. Je mettrai ensuite en libre téléchargement la nouvelle version de PicoIP ainsi qu'une documentation. L'objectif sera atteint, pour ce qui est du développement futur, les bonnes volontés seront les bienvenues !!

Bonne réalisation à tous.