

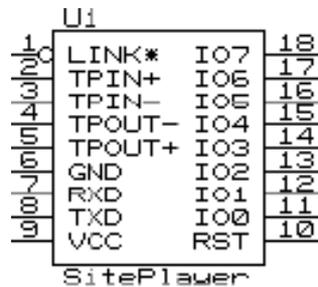
## Table des matières

Module SitePlayer :.....	3
Applications typiques :.....	3
Caractéristiques techniques :.....	3
Kit de test :.....	3
Première prise en main :.....	3
Installation matérielle :.....	3
Programmation :.....	3
Installation des programmes nécessaires :.....	3
Création d'un fichier binaire :.....	4
Création d'une page Web :.....	4
Chargement des fichiers dans SitePlayer :.....	4
Test du SitePlayer :.....	4
Prise en main complète :.....	5
Il faut toujours travailler dans cet ordre :.....	5
Utilisation des ports du module SitePlayer :.....	5
Pour afficher sur une page Web l'état logique d'une entrée de SitePlayer :.....	5
Pour remplacer les 0 ou les 1 par une image de led rouge ou de led verte :.....	6
Pour permuter l'état logique d'une sortie de SitePlayer depuis une page Web :.....	6
Pour forcer une sortie de SitePlayer à un état logique 0 ou 1 depuis une page Web :.....	6
Pour utiliser le port complet de SitePlayer en une entrée numérique de 8 bits :.....	6
Manuel du logiciel :.....	7
Section Définition :.....	8
\$DHCP [on, off] on par défaut.....	8
\$Devicename [nom_ou_commentaire].....	8
\$DownloadPassword [mot_de_passe].....	8
\$Include [nom_fichier].....	8
\$InitialIP [xxx.xxx.xxx.xxx].....	9
\$Parse [extention1].....	9
\$PostIRQ [on, off] off par défaut.....	9
\$Sitefile [nom_de_fichier].....	9
\$Sitepath [path].....	9
Section Objet :.....	9
DBIT [valeur] – Définit une variable d'un Bit.....	10
DB[valeur] – Définit une variable d'un octet ou d'une chaîne de caractères.....	10
DHEX [valeur] – Définit une variable d'un octet.....	10
DW [valeur] – Définit une variable de un mot de 16 bits.....	10
DD [valeur] – Définit une variable double de 32 bits.....	10
DS [valeur] – Définit une variable de dimension comprise entre 1 et 255 octets.....	10
ORG [valeur] – Définissez Emplacement d'Objets.....	10
Objet d'utilisation du port série (défini dans pcafile.inc).....	10
Objet BAUD RATE (défini dans pcafile.inc).....	10
Objet Interface série (SPI).....	11
Objet HalfSec OFF1Fh.....	11
Objet modificateur ExitIf0.....	12
Objet UDPsend OFF1Eh.....	12
\$OutputOnly.....	12
\$Bidirectional.....	12
Section Exportation :.....	12
\$ExportFormatFile [filename] none.....	12
\$ExportFile [outputfilename].....	13
\$ExportHeaderFile [headerfilename].....	13
\$ExportFooterFile [footerfilename].....	13
\$Export.....	13
Création de fichiers Page Web.....	14
SiteObjects en HTML.....	14
Modification de SiteObject.....	14
Sélectionner le Chiffre Particulier d'un Objet.....	14
Calculs mathématiques simples sur un objet.....	15
Choix d'un bit particulier dans un objet.....	15

Éviter qu'un objet soit en conflit avec le texte.....	<a href="#">15</a>
Pages Web de SitePlayer.....	<a href="#">15</a>
Fichier d'interface de SitePlayer.....	<a href="#">15</a>
Créer un fichier d'interface de SitePlayer.....	<a href="#">15</a>
Envoi de données d'un lien vers SitePlayer.....	<a href="#">16</a>
Envoi de données par formulation vers SitePlayer.....	<a href="#">16</a>
Quelques règles de HTML :.....	<a href="#">18</a>
Structure d'un programme html :.....	<a href="#">18</a>
La balise INPUT :.....	<a href="#">18</a>
La balise TEXTAREA :.....	<a href="#">18</a>
La balise SELECT :.....	<a href="#">19</a>
Création des tableaux.....	<a href="#">19</a>
Créer un lien :.....	<a href="#">19</a>
Les formulaires :.....	<a href="#">20</a>
Exemples de syntaxe sur les entrées sorties :.....	<a href="#">20</a>
Exemples non liés :.....	<a href="#">20</a>
Redirection ou rafraîchissement :.....	<a href="#">20</a>
Aides :.....	<a href="#">21</a>
Transmission et réception série :.....	<a href="#">22</a>
La trame :.....	<a href="#">22</a>
Les codes de commande série :.....	<a href="#">22</a>
NOP ( commande 00h) :.....	<a href="#">22</a>
Status ( commande 10h) :.....	<a href="#">22</a>
Reset ( commande 20h) :.....	<a href="#">23</a>
ComParams ( commande 33h) :.....	<a href="#">23</a>
Write or WriteX ( commande 80h ou 90h) :.....	<a href="#">23</a>
Read or ReadX (commande 0C0h ou 0D0h) :.....	<a href="#">23</a>
UDPSend (commande 50h) :.....	<a href="#">23</a>
Reading, Writing, et Toggling Bit Variables (commande 0E0h ou 0A0h ou 0B0h) :.....	<a href="#">23</a>
Envoi de plus d'un objet avec une commande :.....	<a href="#">24</a>
Annexes :.....	<a href="#">25</a>
Miniserveur Internet pour gérer son habitat.....	<a href="#">27</a>
Le principe :.....	<a href="#">27</a>
La photo de la carte :.....	<a href="#">27</a>
Utilisation des entrées 20mA :.....	<a href="#">28</a>
Sortie à relais :.....	<a href="#">28</a>
Attention au DANGER du raccordement des connecteurs sous 240V :.....	<a href="#">28</a>
Utilisation des entrées 240V :.....	<a href="#">28</a>
Utilisation des sorties 240V :.....	<a href="#">29</a>
Les schémas :.....	<a href="#">30</a>
Liste des composants :.....	<a href="#">32</a>
Équivalence des entrées sorties :.....	<a href="#">33</a>
Le fichier binaire du module SitePlayer Site.spd :.....	<a href="#">33</a>
Le fichier Web du module SitePlayer index.htm :.....	<a href="#">34</a>
Programme de rafraichissement page Web Site.spi :.....	<a href="#">36</a>
Le programme du microcontrôleur Site.c :.....	<a href="#">36</a>
Bibliographie :.....	<a href="#">38</a>
RAB de traduction :.....	<a href="#">39</a>

## Module SitePlayer :

SITEPLAYER est un très petit des serveur Web Ethernet. Il permet de réaliser à lui tout seul le coeur d'un serveur Web Server. Il suffit de lui ajouter un simple et ordinaire microcontrôleur et il est prêt à fonctionner. Sur le minuscule circuit imprimé de ce module, il y a un contrôleur Ethernet 10base T, une mémoire Flash dans laquelle on sauvegarde les pages Web et un dispositif d'interface série. Il est programmable en html, mais n'accepte pas le php.



### Applications typiques :

Elles sont domestiques, contrôles industriels, contrôles de processus, dispositifs de test, etc..

C'est un coprocesseur qui rend possible la gestion des protocoles Web et des paquets Ethernet avec n'importe quel microcontrôleur. La communication entre SitePlayer et le microcontrôleur se fait à travers une liaison série RS232 standard à deux fils ou une liaison SPI (liaison série synchrone).

Il peut aussi fonctionner en mode Standalone; Dans ce cas on peut gérer directement les lignes d'entrées et de sorties du module.

Il dispose d'un puissant système de gestion d'objets appelé SITEOBJECT. Ces objets peuvent être des images graphiques, textes, musiques, raccordements, boutons et qui changent en fonction des données que le microcontrôleur de gestion envoie à SitePlayer. Ainsi, les pages Web peuvent contenir des objets graphiques comme, des boutons boutons ou des interrupteurs qui se modifient ou qui actionnés changent les variables de SitePlayer et qui pourront être lus par un microcontrôleur.

### Caractéristiques techniques :

- Un serveur Web Ethernet complet sur un circuit imprimé de 23 x 33 mm.
- Des Programmes en JAVA, C, C++, et Visual Basic peuvent piloter ou contrôler SitePlayer.
- La gestion en temps réel des graphiques permet de réaliser des afficheurs, une barre de défilement, des boutons et des interrupteurs.
- Jusqu'à 768 octets de SITEOBJECTS™.
- En mode Standalone SitePlayer permet de gérer des ressources externes sans ajouter de microcontrôleur.
- Les pages Web peuvent être créées avec la plupart des programmes comme FrontPage ou Mozilla.
- Il se connecte directement à une prise RJ45 filtrée.
- Jusqu'à 48K octets de pages Web peuvent être descendues dans la mémoire Flash de SitePlayer via Ethernet.
- Le logiciel peut être mis à jour via Ethernet.
- Il supporte les protocoles suivants: ARP, ICMP, IP, UDP, TCP, DHCP.
- Gestion des adresses IP fixes ou dynamiques données par des serveurs DHCP.
- Il dispose d'un port série pour s'interfacer au processeur de contrôle ou à un ordinateur.

### Kit de test :

Il comprend le module SitePlayer, le connecteur Ethernet, les logiciels de contrôle et de chargement ....

### Première prise en main :

#### Installation matérielle :

L'application comprenant SitePlayer doit être alimentée et raccordée à un réseau informatique, ou à un Hub, ou à un switch par un câble réseau, ou encore à un ordinateur par un câble réseau croisé.

#### Programmation :

#### Installation des programmes nécessaires :

Pour faciliter les explications on suppose qu'un lecteur [P:](#) est disponible sur le réseau informatique. Si vous n'avez pas ce lecteur, vous pouvez tout aussi bien le remplacer par [C:](#) dans tous les exemples proposés.

1. Récupérer le logiciel sur le site : [http://www.siteplayer.com/docs/001212/siteplayer\\_complete\\_101003.zip](http://www.siteplayer.com/docs/001212/siteplayer_complete_101003.zip)
2. Installer en cliquant sur SETUP.EXE (fichier compressé dans un fichier ZIP). Les logiciels s'installent automatiquement dans le répertoire [C:\Program Files\SitePlayer](#)
3. Dans l'ordinateur, créer le répertoire [P:\Site](#)
4. Récupérer le logiciel Mozilla sur le site <http://www.clubic.com/telecharger-fiche10797-mozilla.html>
5. Installer Mozilla sur votre ordinateur.

## Création d'un fichier binaire :

Avec le logiciel [Bloc-notes](#) accessible dans tous les ordinateurs faites un copier coller du texte suivant :

```
; Les variables essentielles à placer dans votre système
;
;$Devicename :          Indique le nom ou la description du produit
$Devicename "Test du SitePlayer"

;$DHCP on :            SitePlayer trouve une adresse IP automatiquement sur DHCP
$DHCP on

;$DownloadPassword :   Demande un mot de passe avant de charger la première page Web
$DownloadPassword ""

;$InitialIP            Force une adresse IP au SitePlayer Exemple 192.168.0.250
$InitialIP "192.168.0.250"

;$PostIRQ on off      SitePlayer doit générer une IRQ niveau bas sur la broche 11
$PostIRQ off

;$Sitefile             Donne le nom et chemin du fichier binaire à charger
$Sitefile "P:\Site\Test.spb"

;$Sitepath            Donne le chemin des fichiers des pages Web à charger
$Sitepath "P:\Site"
```

Vous lisez, comprenez et modifiez à votre goût et sauvez ce fichier sous le nom Test.spd dans le répertoire [P:\Site](#).

## Création d'une page Web :

Vous démarrez [Mozilla](#) en ouvrant une fenêtre du composeur.

Vous écrivez un texte dans cette fenêtre et vous sauvez le fichier dans P:\Site. Mozilla vous demande un titre de page : donnez **Index** et un nom de fichier : écrivez **Index.htm**. (attention pas .html)

Après sa sauvegarde ce fichier peut être lu par [Bloc-Notes](#) et vous obtenez :

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
<head>
  <meta content="text/html; charset=ISO-8859-1"
    http-equiv="content-type">
  <title>index</title>
  <meta content="Guy" name="author">
</head>
<body style="color: rgb(0, 0, 0);" alink="#ee0000" link="#0000ee"
  vlink="#551a8b">
<div style="text-align: center; background-color: rgb(255, 255, 153);"><br>
<big style="font-weight: bold; color: rgb(255, 0, 0);"><big><big><big><big>BONJOUR<br>
<small>vous &ecirc;tes bien sur le serveur WEB de la maison<br>
<span style="color: rgb(51, 204, 0);">En ce moment tout est normal</span><br>
<br>
<br>
<br>
<br>
<br>
<br>
<br>
</small></big></big></big></big></big></div>
</body>
</html>
```

## Chargement des fichiers dans SitePlayer :

1. Lancez l'application dans le menu de Windows : Démarrer\Programmes\SitPlayer\SiteLinker
2. Dans l'application SiteLinker :
  - Cliquez sur File\Open et ouvrez le fichier P:\Site\Test.spd
  - Cliquez sur Download\Make Download File F3 ce qui compile le fichier source en Test.spb dans [P:\Site](#)
  - S'il n'y a pas d'erreur de détectée, cliquez sur Download\SitePlayer F4 ce qui télécharge le fichier binaire et la page Web dans le SitePlayer. Attendez un peu, en général le logiciel vous prévient qu'il n'y a pas de mot de passe, ce qui n'est pas grave et pour l'instant on laisse comme ça.

## Test du SitePlayer :

Avec votre navigateur Internet favori (*choisissez Mozilla c'est une application opensource, donc elle ne comprend aucun moucharde*) ouvrez le site Internet <http://192.168.0.250> et vous avez la page de votre miniserveur WEB SitePlayer. Vous pouvez aussi ne taper que l'adresse IP : [192.168.0.250](http://192.168.0.250) et ça marche de la même façon.

## Prise en main complète :

### Il faut toujours travailler dans cet ordre :

1. Créer un répertoire vide qui contiendra tous les fichiers nécessaires à SitePlayer.
2. Faire le fichier binaire xxxx.spd avec [Bloc-notes](#).
3. Faire les images .jpg avec un logiciel adéquat qui peut être OpenOffice Draw ou ...
4. Faire complètement la page Web avec [Mozilla](#) en fenêtre composeur avec toutes la structure voulue et insérer les images éventuelles dans un fichier obligatoirement nommé Index.htm.
5. Modifier Index.htm avec [Bloc-notes](#) pour insérer les lignes de contrôle des entrées/sorties. Après ce stade, si vous reprenez le fichier avec le composeur Mozilla, le fichier sera altéré dans ses lignes de contrôle et vous devrez repasser sur Bloc-notes.
6. Utiliser l'application [SiteLinker](#) pour compiler le fichier xxxx.spd et charger tous les fichiers dans SitePlayer. Pour aller plus vite vous pouvez faire F12 puis F5 pour charger le fichier xxxx.spd précédemment utilisé, le compilateur et le chargeur SitePlayer.

### Utilisation des ports du module SitePlayer :

**Attention** en cas de rupture d'alimentation, SitePlayer fait automatiquement un reset, ce qui signifie que toutes les sorties se repositionnent au 1 logique. **La programmation est donc perdue.** Pour éviter ce problème il faut adjoindre une pile (consommation 75mA typique) ou un microcontrôleur avec EEPROM interne comme le PIC16F628 par exemple.

Sur l'application sur laquelle est montée SitePlayer, il y a 8 entrées ou sorties sur les broches 11 à 18. Il s'agit maintenant d'ajouter la possibilité :

- De forcer des sorties logiques du port.
- De lire des entrées logiques du port.

Les deux fichiers Index.htm et Test.cpd doivent être cohérents et une entrée ou sortie indiquée à l'un doit aussi être indiquée à l'autre par le même nom.

La fin du fichier Test.cpd doit être modifiée comme suit :

```
; Les variables essentielles à placer dans votre système
;
;$Devicename :          Indique le nom ou la description du produit
$Devicename "Test du SitePlayer"

;$DHCP on :            SitePlayer trouve une adresse IP automatiquement sur DHCP
$DHCP on

;$DownloadPassword :   Demande un mot de passe avant de charger la première page Web
$DownloadPassword ""

;$InitialIP            Force une adresse IP au SitePlayer Exemple 192.168.0.250
$InitialIP "192.168.0.250"

;$PostIRQ on off       SitePlayer doit générer une IRQ niveau bas sur la broche 11
$PostIRQ off

;$Sitefile             Donne le nom et chemin du fichier binaire à charger
$Sitefile "P:\Site\Test.spb"

;$Sitepath             Donne le chemin des fichiers des pages Web à charger
$Sitepath "P:\Site"

; Définitions nécessaires pour utiliser directement les ports du SitePlayer seul.
; Page 38 sur 39 du SitePlayer software Manuel on a la table mémoire du module
; On voit que le Port 1 est à l'adresse 0FF00h et que les Bits du Port 1
; vont de 0FF11h à 0FF18h.
    org 0ff00h
p1          ds 1          ; Port 1 les 8 bits

    org 0ff11h
io0         ds 1          ; Port 1 Bit num 0
io1         ds 1          ; Port 1 Bit num 1
io2         ds 1          ; Port 1 Bit num 2
io3         ds 1          ; Port 1 Bit num 3
io4         ds 1          ; Port 1 Bit num 4
io5         ds 1          ; Port 1 Bit num 5
io6         ds 1          ; Port 1 Bit num 6
io7         ds 1          ; Port 1 Bit num 7
```

### Pour afficher sur une page Web l'état logique d'une entrée de SitePlayer :

Vous démarrez [Mozilla](#) en ouvrant une fenêtre du composeur.

Vous ouvrez le fichier **Index.htm** et ajoutez la ligne : Le bit 5 du port1 vaut XX logique.

Sauvez le fichier et fermez Mozilla.

Ouvrez le fichier Index.htm avec [Bloc-notes](#) et remplacez le XX par ^io5. L'accent circonflexe qui précède io5 n'est pas une erreur.

## Par exemple cela donne :

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
<head>
  <meta content="text/html; charset=ISO-8859-1"
  http-equiv="content-type">
  <title>index</title>
  <meta content="Guy" name="author">
</head>
<body style="color: rgb(0, 0, 0);" alink="#ee0000" link="#0000ee"
  vlink="#551a8b">
<div style="text-align: center; background-color: rgb(255, 255, 153);"><br>
<big style="font-weight: bold; color: rgb(255, 0, 0);"><big><big><big><big>BONJOUR<br>
<small>vous &ecirc;tes bien sur le serveur WEB de la maison<br>
<span style="color: rgb(51, 204, 0);">En ce moment tout est normal</span><br>
Le bit 5 du port1 vaut ^io5 logique<br>
<br>
<br>
<br>
</small></big></big></big></big></div>
</body>
</html>
```

## Pour remplacer les 0 ou les 1 par une image de led rouge ou de led verte :

Comme précédemment vous devrez utiliser l'application [SiteLinker](#) pour compiler le fichier Test.spd puis charger les 2 fichiers Test.spb et Index.htm dans SitePlayer.

Avec votre navigateur Internet vous pouvez voir le résultat de l'état logique du bit 5 du port 1. Si vous changez l'état de cette entrée il faut rafraîchir la page Web pour observer le changement (Sur Mozilla Menu Affichage\Actualiser ou icône Actualiser la page courante).

On peut remplacer le 1 ou 0 logique que calcule ^io5 par des images ou des textes différents des nombres 1 ou 0. Il suffit de procéder comme précédemment mais il faut remplacer le XX par `` et créer deux images, une pour le 0 logique que l'on va nommer Led0.jpg ● dans notre exemple, et l'autre Led1.jpg ● pour le 1 logique. (Pour récupérer ces dessins de Led verte ou rouge, faites un copier/coller dans un logiciel de dessins et sauvez les dans un fichier). Ici la variable ^io5:1 sera remplacée par la valeur de l'état logique correspondant pour former le nom du fichier image appelé. Dans l'exemple les deux images sont construites avec un logiciel de dessin quelconque et mesurent 50 pixels de hauteur pour 50 pixels de largeur, mais dans le fichier on peut changer les dimensions comme dans l'exemple ou on a pris 30 et 70.

## Pour permuter l'état logique d'une sortie de SitePlayer depuis une page Web :

Avec le logiciel [Bloc-notes](#) accessible faites un copier coller du texte suivant :

```
HTTP/1.0 302 FOUND
LOCATION:/Index.htm
```

Faites en sorte que le curseur se retrouve en dessous de la deuxième ligne pour que le texte comporte les deux lignes écrites plus une ligne vide.

Sauvegardez ce fichier dans le répertoire [P:\Site](#) sous le nom Test.spi.

Toujours avec [Bloc-notes](#) vous ouvrez Index.htm et ajoutez la ligne :

```
<a href="Test.spi?io5=^io5~1">port1 bit 5 = ^io5</a>
```

Cette ligne fait appel au fichier Test.spi et permute io5 en lisant l'état du bit 5, en l'inversant, et en le remplaçant, puis permet de rafraîchir la page Web pour visualiser le résultat.

Évidemment comme précédemment il faut utiliser [SiteLinker](#) pour charger SitePlayer.

## Pour forcer une sortie de SitePlayer à un état logique 0 ou 1 depuis une page Web :

Pour forcer une sortie à un état logique bien défini, il suffit de garder le fichier Test.spi précédent et d'insérer la bonne ligne dans le fichier Index.htm avec [Bloc-notes](#).

Pour forcer à l'état logique 1 insérez : `<a href="Test.spi?io5=1">port1 bit 5 = 1</a>`

Pour forcer à l'état logique 0 insérez : `<a href="Test.spi?io5=0">port1 bit 5 = 0</a>`

Utilisez [SiteLinker](#) pour charger SitePlayer.

## Pour utiliser le port complet de SitePlayer en une entrée numérique de 8 bits :

Comme le port1 du SitePlayer fait 8 bits, on peut l'utiliser comme une entrée numérique sur 1 octet. Le poids fort pour le bit 7 et le poids faible pour le bit 0. Il suffit d'écrire les lignes suivantes dans Index.htm avec [Bloc-notes](#) :

```

```

```

```

```

```

et de copier/coller les fichiers suivants depuis [C:\Program Files\SitePlayer\Graphic\\_Library](#) vers [P:\Site](#) :

```
0_BLK.JPG 1_BLK.JPG 2_BLK.JPG 3_BLK.JPG 4_BLK.JPG
```

```
5_BLK.JPG 6_BLK.JPG 7_BLK.JPG 8_BLK.JPG 9_BLK.JPG
```

C'est encore le principe d'insertion d'image pour afficher des informations. Les images x\_BLK.JPG font partie de ce que fournit le kit de SitePlayer.

Utilisez [SiteLinker](#) pour charger SitePlayer.

## Manuel du logiciel :

Le fichier de définition (extension .spd) contient les directives pour le programme SiteLinker qui crée le fichier image binaire (extension .spb) lequel est téléchargé à SitePlayer.

Ce fichier contient tout le code pour SitePlayer et inclut les paramètres de démarrage, de pages de web, et autres objets et prévoit aussi l'exportation vers d'autres formats de dossier tels que HTML, C, et Visual Basic.

Ce fichier ressemble à un fichier assembleur de microprocesseur, c'est simplement un autre langage.

### Il a trois sections :

- La section Définition qui donne les paramètres de démarrage de SitePlayer.
- La section Objet qui contient les commandes et visualisations des Entrées/Sorties.
- La section Exportation qui dit où et comment les fichiers seront manipulés pendant l'exécution dans SitePlayer.

### Exemple typique :

```
;DEFINITION SECTION
$definition1 argument ;comment
$definition2 argument
...
$definitionx argument
;SECTION OBJET
org 0h
object1      db 14 ;nom objet1, taille 1 octet, valeur initiale 14.
object2      dw 0   ;nom objet2, taille 1 mot = 2 octets, valeur initiale 0.
object3      dd 0   ;nom objet3, taille 1 long = 4 octets, valeur initiale 0.
object4      ds 47 ;nom objet4, taille 47 octets, pas de valeur initiale.
object5      db "String with some data"
object       dbit 1;nom objet, taille 1 bit, valeur initiale 1 logique.
;SECTION EXPORTATION
$ExportFormatFile "filename" ;défini le format export
$ExportFile       "filename" ;défini le nom du fichier export
$Export           ;instruction export
```

Un commentaire commence par un point-virgule. Tous les nombres peuvent être écrit en décimal, en hexadécimal (h), en binaire (b), ou en octal (o). Si aucune lettre ne suit le nombre, alors SitePlayer considère que le nombre est décimal.

### Exemple du récepteur stéréo SitePlayer :

```
;
; Récepteur stéréo
;
$devicename "NetMedia Audio Receiver Demo"
$DHCP on
$DownloadPassword ""
$InitialIP "192.168.1.250"
$PostIRQ off
$sitefile "C:\Program Files\SitePlayer\rcvr.spb"
$sitepath "C:\Program Files\SitePlayer\SP_Root"
$Include "C:\Program Files\SitePlayer\pcadef.inc"
;
; OBJECTS du récepteur
;
org 0h
select      db 2   ;initialisation à FM
volume      db 0   ;Volume à l'arrêt
channel     dw 921 ;Canal 92.1

org 0FF15h
power       db 1   ;LED verte

org 248
AM_page     dbit 0 ;AM
FM_page     dbit 0 ;FM
Tape_page   dbit 0 ;Tape
CD_page     dbit 0 ;CD
Aux_page    dbit 0 ;Aux
; autres OBJECTS tests
;
org 30h
xytest_x    db 0
xytest_y    db 0
;
; fichiers de réception EXPORT
;
$ExportFormatFile contient les définitions au format $ExportFile
$ExportFormatFile "C:\Program Files\SitePlayer\makeHTML.def"

;$ExportHeaderFile contient les informations d'entête pour $ExportFile
$ExportHeaderFile "C:\Program Files\SitePlayer\htmlheader.htm"
```

```

; $ExportFooterFile contient les informations de fin pour $ExportFile
$ExportFooterFile "C:\Program Files\SitePlayer\htmlfooter.htm"

; $ExportFile donne le nom du fichier d'exportation créé pour ces définitions
$ExportFile "C:\Program Files\SitePlayer\SP_Root\receiver_export.htm"

; $Export      commande qui crée le HTML $ExportFile et remise à 0 des paramètres
$Export

; Export définitions pour un fichier formaté Visual Basic
$ExportFormatFile "C:\Program Files\SitePlayer\makevb.def"
$ExportFile "C:\Program Files\SitePlayer\rcvrdefs.bas"
$Export

; Export définitions pour un fichier d'entête en langage C
$ExportFormatFile "C:\Program Files\SitePlayer\makec.def"
$ExportFile "C:\Program Files\SitePlayer\rcvrdefs.h"
$Export

; Export définitions pour un fichier en assembleur
$ExportFormatFile "C:\Program Files\SitePlayer\makeasm.def"
$ExportFile "C:\Program Files\SitePlayer\rcvrdefs.asm"
$Export

```

### **Section Définition :**

La section Définition détermine les paramètres de démarrage de SitePlayer. Ils sont précédés du signe dollar (\$) et sont des contrôles spécifique pour SitePlayer. Aucun espace ou tabulation n'est permis entre le signe du dollar et le corps du contrôle.

#### **Par exemple celle du récepteur est :**

```

; $devicename Indique le nom ou la description de l'application
$devicename "NetMedia Audio Receiver Demo"
; $DHCP on SitePlayer trouve une adresse IP automatiquement sur DHCP
$DHCP on
; $DownloadPassword Demande un mot de passe avant de charger la première page Web
$DownloadPassword ""
; $InitialIP Force une adresse IP au SitePlayer si le serveur n'est pas en DHCP
$InitialIP "192.168.1.250"
; $PostIRQ SitePlayer doit générer une IRQ niveau bas sur la broche 11
$PostIRQ off
; $Sitefile Donne le nom et chemin du fichier binaire à charger
$sitefile "C:\Program Files\SitePlayer\rcvr.spb"
; $Sitepath Donne le chemin des fichiers des pages Web à charger
$sitepath "C:\Program Files\SitePlayer\SP_Root"
; $Include Donne le nom du fichier à inclure au moment de l'exécution du processus
$Include "C:\Program Files\SitePlayer\pcadef.inc"

```

### **Les descriptions des variables sont :**

#### **\$DHCP [on, off] on par défaut**

- Pour fixer une adresse IP fixe écrire \$DHCP off.
- Pour une adresse IP donnée automatiquement par le serveur de réseau DHCP écrire \$DHCP on. Si SitePlayer ne trouve pas le serveur, il prendra l'adresse IP par défaut, si vous utilisez la fonction \$InitialIP

**Exemple :** \$DHCP on

#### **\$Devicename [nom\_ou\_commentaire]**

On peut donner un nom de 64 caractères qui peut être un nom de produit ou une description de produit.

Chaque SitePlayer peut avoir un nom de 64 caractères. Ce peut être un nom du produit, ou une quelconque description qui caractérisera votre application sur un réseau où il y en a plusieurs.

**Exemple :** \$Devicename "Récepteur audio de démonstration"

#### **\$DownloadPassword [mot\_de\_passe]**

Si vous ne voulez pas qu'une tierce personne télécharge un nouveau site Web dans SitePlayer, vous pouvez spécifier un mot de passe jusqu'à 16 caractères qui sera systématiquement demandé à chaque modification. Ce mot peut être changé à chaque téléchargement. Pour plus de sécurité SitePlayer vérifie que le téléchargement se fait depuis le réseau local, même si des pirates peuvent imiter un réseau local (ce n'est pas très facile).

Si vous ne voulez pas de mot de passe écrivez : \$DownloadPassword ""

**Exemple 1 :** \$DownloadPassword "mon\_mot\_de\_passe"

**Exemple 2 :** \$DownloadPassword "" pour annuler le mot de passe

#### **\$Include [nom\_fichier]**

Vous pouvez avoir une bibliothèque d'objets, ou vouloir utiliser une Bibliothèque d'objets de NetMedia. Vous pouvez les inclure comme si vous les aviez copiés directement le fichier. Un fichier inclus peut pointer un autre fichier à inclure et ceci jusqu'à 8 niveaux de fichier. Les chemins de fichier inclus doivent être complet depuis la racine.

**Exemple :** "C:\Program Files\SitePlayer\pcadef.inc"

### **\$InitialIP [xxx.xxx.xxx.xxx]**

Cette fonction permet d'affecter une adresse IP fixe par défaut à SitePlayer. Vous devez utiliser cette fonction même si vous demandez une adresse à un serveur DHCP.

**Exemple :** \$InitialIP 192.168.0.250

Si vous donnez l'adresse **IP 0.0.0.0** cela permet de fixer l'adresse IP à travers un PING.

Premièrement, faites une table d'entrée ARP statique utilisant la MAC adresse de votre SitePlayer (voir le programme de test du port série pour trouver la MAC adresse) et les adresses IP voulues.

**Exemple :** ARP -S 192.168.1.240 00-03-75-00-05-3A

Puis faites un PING à l'adresse du SitePlayer.

**Exemple :** PING 192.168.1.240

Par sécurité, une fois que l'adresse est affectée par un PING, elle ne peut être modifiée que par un Reset ou une mise sous tension.

### **\$Parse [extention1]**

Les fichiers recherchés sont toujours avec l'extension htm, html, xml, et wml. Si vous voulez en ajouter cette fonction le permet.

**Exemple :** \$Parse zip permet de rechercher les fichiers \*.zip.

### **\$PostIRQ [on, off] off par défaut**

Si un utilisateur envoie des données à SitePlayer à travers une forme ou en appuyant sur des boutons spéciaux, SitePlayer peut provoquer une transition sur la broche 11 (ou io0) du module quand la données arrive. Cela permet d'endormir les appareils et de ne les réveiller que lorsque cette interruption se produit.

Si **\$PostIRQ est on**, alors la broche 11 sera mise au niveau bas (0 Volts) puis au niveau haut (+5 Volts) pour créer une impulsion de 5 microsecondes. Un processeur verra alors un front descendant au lieu d'un niveau logique et pourra interpréter cette interruption.

Si **\$PostIRQ est off**, alors la façon seule de savoir si les données sont arrivées ou pas, est d'utiliser le registre d'état et de commande du port série et d'aller voir si la donnée est arrivée. Souvent ce n'est pas nécessaire pour des applications comme un thermostat où il suffit de faire une lecture de temps en temps et d'utiliser une ancienne valeur pour prendre des décisions. Voir le registre d'état et de commande du port série dans la partie transmission et réception pour plus d'informations.

**Exemple :** \$PostIRQ off

### **\$Sitefile [nom\_de\_fichier]**

Quand le logiciel SiteLinker ouvre un répertoire (voir \$SitePath) pour faire un site Web à téléchargé dans SitePlayer, il fait d'abord un fichier image binaire. Ce fichier d'extension .spb est téléchargé dans la mémoire flash de SitePlayer. Il est portable et peut être envoyé à d'autres personnes. Par exemple, si vous avez une mise à jour de l'image d'un de vos produits, vous pouvez envoyer un fichier binaire à vos clients qui peuvent le télécharger dans leur SitePlayer. Ceci vous évite de donner le détail du projet.

**Exemple :** \$SiteFile "C:\Program Files\SitePlayer\image47.spb"

### **\$Sitepath [path]**

SitePath donne au logiciel SiteLinker le chemin des fichiers de données qui composent le Serveur SitePlayer. C'est en quelque sorte la racine de référence des fichiers.

Quand un utilisateur appelle le fichier http://xxx.xxx.xxx.xxx/index.htm il accède à des données dans C:\Program Files\SitePlayer\SP\_Root\index.htm. Donc pour chaque projet il faut un répertoire de stockage différent.

**Exemple :** \$Sitepath "C:\Program Files\SitePlayer\SP\_Root"

### **Section Objet :**

Dans cette section on déclare des objets en leur donnant un nom, un type, une dimension et une valeur initiale. Ces variables permettent la communication entre les pages HTML et le fichier binaire. Elles doivent avoir le même nom dans tous les fichiers. Ces objets sont formés des caractères A à Z, 0 à 9 et le caractère souligné \_, ne doivent pas dépasser 32 caractères de long et ne pas appartenir aux noms réservés listés dans la partie Symboles réservés.

Pour **créer un objet ça ressemble à :**

Mon\_Objet db 5 ; je commente

Cette ligne crée un objet de **1 octet** s'appelant **Mon\_Objet** initialisé à la valeur **5**. (db = define byte)

Pour utiliser des coordonnées graphique x/y le principe est modifié par l'ajout d'un caractère souligné \_ suivi de la coordonnée x ou y. objet.x sera nommé objet\_x dans SitePlayer.

**Par exemple** la Section Objet du récepteur FM ressemble à :

org 0h

select db 2 ;nom select, taille 1 octet, valeur initiale 2 pour FM

volume db 0 ;Par défaut le volume est bas.

channel dw 921 ;Ce mot donne le canal 92.1 après un Reset.

org 0FF15h

power db 1 ;Au départ c'est à l'arrêt, l'adresse est io4 donc la LED verte

org 248 ;nombre décimal parce que non terminé par h, b ou o

AM\_page dbit 0 ;AM

FM\_page dbit 0 ;FM

Tape\_page dbit 0 ;Enregistreur  
 CD\_page dbit 0 ;CD  
 Aux\_page dbit 0 ;Auxiliaire

org 30h  
 xytest\_x db 0 ;coordonnée x du graphique de xytest  
 xytest\_y db 0 ;coordonnée y du graphique de xytest

#### Autres exemples possibles :

#### DBIT [valeur] – Définit une variable d'un Bit

Exemple : NomObjet DBIT 1 ;1 bit initialisé à 1 logique

#### DB[valeur] – Définit une variable d'un octet ou d'une chaîne de caractères

La chaîne de caractère doit être délimitée par des guillemets. Vous pouvez utiliser les codes hexadécimaux (%HH) ou (%20) = espace ou autres .... pour les caractères spéciaux en HTML.

Exemples : Objet DB 47h ;Un octet initialisé à 47 en hexadécimal  
 Phrase DB "C'est une phrase%0D%0A" ;Chaîne de 18 caractères de longueur

#### DHEX [valeur] – Définit une variable d'un octet

Les objets définis sont affichées et reçues à travers les pages Web comme des nombres hexadécimaux.

Exemple : Nom DHEX 47h ;Définit un objet hexadécimal initialisé à 47h.

#### DW [valeur] – Définit une variable de un mot de 16 bits

Exemple : Nom DW 0FF12h ;Nom prend la valeur initiale de 0FF12 en hexadécimal

#### DD [valeur] – Définit une variable double de 32 bits

Exemple : Nom DD 2435A734h ;Nom est initialisé à 2435A734 en hexadécimal

#### DS [valeur] – Définit une variable de dimension comprise entre 1 et 255 octets

La valeur initiale n'est pas spécifiée. Si l'objet est défini pour 1 ce sera un octet, pour 2 ce sera un mot, pour 4 un Double et pour plus une chaîne de caractères.

Exemple : Nom DS 12 ;Chaîne de 12 caractères  
 Nom2 DS 2 ;nombre entier sur 2 octets

#### ORG [valeur] – Définissez Emplacement d'Objets

Précise l'emplacement dans la mémoire de SitePLAYER où les objets seront créés.

Exemple : ORG 200h  
 Object1 db 0 ;à l'adresse 200 en hexadécimal  
 Object2 db 0 ;à l'adresse 201 en hexadécimal

#### Objet d'utilisation du port série (défini dans pcaodef.inc)

L'objet COM permet d'envoyer des chaînes de données sur la sortie du port série de SitePlayer ce qui est utile quand on veut qu'il dialogue avec un système. C'est le même principe que pour un objet chaîne de caractères, sauf qu'au lieu de placer les données dans la RAM, on les envoie sur la ligne TX du port série. La forme de la page Web sera la même. Le protocole est de 8 bits, aucune parité, un stop bit, sans poignée de main et à la vitesse de transmission courante. Voir BAUD RATE pour changer la vitesse de transmission à partir d'une page Web.

Cet exemple HTML envoie la chaîne "Coucou tout le monde" sur la broche TX du port série quand on clique sur le lien Coucou! Dans la page Web.

Exemple : [Coucou](comtest.spi?com=Coucou%20tout%20le%20monde)

Tout octet de 0 à 255 peut être envoyé. Comme ce sont des caractères non-imprimables vous devez écrire :

Exemple : [Send Ctrl-A Ctrl-C](comtest.spi?com=%01%03)

où les deux nombres après le signe pourcent sont des nombres hexadécimaux de commande. Pour faire un retour à la ligne, on utilise généralement %0D%0A. Si vous utilisez une chaîne de plusieurs lignes formatées par un composeur de page Web, les retours à la ligne sont déjà intégrés sous la forme %XX. Le fonctionnement normal du port série instaurera des commandes (ReadObject, WriteObject...) et c'est à vous d'éviter des conflits de lecture et écriture simultanées.

#### Objet BAUD RATE (défini dans pcaodef.inc)

Cet objet est un nombre entier chargé dans le registre de l'horloge le l'UART de SitePlayer pour produire une vitesse de transmission correcte. La formule qui calcule cet objet BAUD RATE est:

65536 - (1250000 / baudrate)

Voici quelques valeurs couramment utilisées :

Vitesse	Objet baudrate						
110	54172	1200	64494	7200	65362	38400	65503
300	61369	2400	65015	9600	65406	57600	65514
600	63453	4800	65276	19200	65471	115200	65525

Exemple de vitesse à 9600 baud : [Cliquer pour 9600 baud](comtest.spi?baud=65406)

## Objet Interface série (SPI)

C'est un bus série à haute vitesse qui permet le dialogue avec des composants électroniques comme des convertisseurs analogiques, des registres à décalage, des bascules d'entrées/sorties, des contrôleurs de moteurs, etc ... Dans ce mode, SitePlayer est le maître et les périphériques doivent s'adapter notamment à sa vitesse. Le microcontrôleur PIC16F876 ne va pas assez vite.

Définition des broches pour l'objet SPI	
Broches	Description
IO0	SCK Horloge de synchronisme (Serial Clock)
IO1	MOSI Sortie Maître, Entrée esclave (Master Output Slave Input)
IO2	MISO Entrée Maître, Sortie Esclave (Master Input Slave Output)
IO3	Sélection boîtier (Option)
IO4	Sélection boîtier (Option)
IO5	Sélection boîtier (Option)
IO6	Sélection boîtier (Option)
IO7	Sélection boîtier (Option)

SitePlayer fournit 5 sélections de boîtier. Si on clique sur un dessin ou un lien, la broche de sélection boîtier choisie passe à l'état bas puis remonte. (Voir les exemples). Les trames de 8 bits sont transmises avec le MSB en premier. Les données sortent de MOSI et entre par MISO en même temps. Les entrées et sorties sont dépendantes, si vous envoyez 5 octets, vous recevez aussi 5 octets. Si tous les octets ne vous intéressent pas vous devrez quand même les gérer. Par exemple le CAN 24bits LTC2415 envoie 32 bit de trame chaque fois qu'il est sollicité, vous devrez les recevoir en 4 octets et envoyer aussi 4 octets à 0.

La donnée entrante sera placée en interne et pointé par un unique objet appelé SPIaddr. Ceci rassemble les 5 composants dans 5 adresses mémoires et transmet l'intégralité dans une page Web.

### Exemple d'un lien de page Web qui appelle la SPI :

```
<a href="spitest.spi?IO0=0&SPIaddr=%80&IO3=0&SPIBus=%47%98&IO3=1">SPI Test</a>
```

Détail des fonctions des broches :

#### Au départ :

- IO0=0 signifie que SitePlayer démarre l'horloge SCK avec un 0 logique puis bascule vers le 1 logique et reviens vers le 0 logique pour chaque bit envoyé.
- IO0=1 signifie que SitePlayer démarre l'horloge SCK avec un 1 logique puis bascule vers le 0 logique et reviens vers le 1 logique pour chaque bit envoyé.

Ceci permet d'adapter votre composant SPI à l'horloge de synchronisation.

Le circuit LTC2415, mentionné auparavant, spécifie que l'horloge débute par un état bas. Une fois que la broche SCK a été mise à un état de départ, vous ne devez pas la maintenir à cet état, mais la laisser faire. Si vous avez 5 composants qui débutent avec une horloge à l'état bas, vous ne mettrez la broche IO0 qu'une seule fois à 0 logique au moment du départ.

SPIaddr=%80 signifie que c'est l'adresse qui sera utilisée pour stocker la donnée SPI dans la mémoire de SitePlayer et qu'elle est à l'adresse 80h. SPIaddr ne traite qu'un octet et vous devrez envoyer les données SPI dans un des 256 octets de RAM de SitePlayer.

IO3=0 signifie que SitePlayer sélectionne le composant SPI commandé par la broche IO3.

SPIBus=%47%98 signifie que deux trames SPI de 8 bits enverront leur MSB en premier, et que 2 trames de 8 bits seront reçues aux adresses 80h et 81h (voir SPIaddr expliqué précédemment). 47h sortira le premier en commençant par son MSB puis 98h en commençant aussi par son MSB.

IO3=1 signifie que la transaction SPI est finie sur la broche IO3.

Pour envoyer une grande quantité de données vous pouvez utiliser Visual Basic, C ou Java ce qui peut être plus lisible.

Suivant votre application, vous pouvez avoir besoin de résistances de tirage vers le niveau haut sur certaines broches. Vous pouvez vous référer à la documentation du P89C51 sur le CD ou sur le Site Internet de SitePlayer pour connaître le courant (en mA) disponible sur les broches d'entrées sorties.

## Objet HalfSec OFF1Fh

C'est un octet qui est décrementé toutes les 0,50135 secondes. Cela peut être utilisé comme un timer par un microcontrôleur, ou peut être utilisé pour basculer une page Web après un certain temps. Voir l'objet modificateur ExitIF0 pour plus d'information.

**Exemple :** ^object:ExitIF0

## Objet modificateur ExitIf0

Si cet octet = 0, la page Web arrête son ouverture.

Si cet octet ≠ 0, la page s'affiche normalement.

Cette commande peut être utilisée pour sécuriser une page Web. Si un processeur détermine qu'une personne peut regarder une page, alors il met une valeur différente de 0 et la page pourra être vue.

**Exemple :** ^object:ExitIf0

Utilisé conjointement avec l'objet HalfSec, SitePlayer peut ouvrir l'accès aux pages pour un certain temps sans nécessiter un processeur supplémentaire. Pour une minute il faut le régler à 120.

**Exemple: Les pages que vous voulez protéger devront commencer par :** ^HalfSec:ExitIf0

Quand le timer décompte = 0, les pages ne peuvent pas être affichées si un processeur ne fait pas un reset.

## Objet UDPsend 0FF1Eh

Quelque soit la donnée envoyée à cet objet, SitePlayer le comprend comme un paquet envoyé au format UDP et comme s'il avait été reçu sur le port COM. Voir réception et transmission UDP pour plus d'information.

## \$OutputOnly

Cette commande placée devant la définition d'un objet, impose les objets suivants seulement comme des sorties. Ils ne peuvent plus être utilisés pour entrer des données. Pour un thermostat, par exemple, vous pourrez lire la température mais pas modifier le réglage.

**Exemple :**

\$OutputOnly	
température	db 72
\$bidirectional	
reglage	db 72

Ici la variable température est une sortie tandis que reglage est bidirectionnelle.

## \$Bidirectional

Cette commande placée devant la définition d'un objet, impose les objets suivants comme étant simultanément des entrées ou des sorties.

**Exemple :**

\$OutputOnly	
température	db 72
\$bidirectional	
reglage	db 72

Ici la variable température est une sortie tandis que reglage est bidirectionnelle.

## Section Exportation :

Elle définit comment et où les fichiers seront envoyés au moment du processus de SiteLinker de SitePlayer. C'est utile pour créer des fichiers de données qui seront passés à des Assembleur, compilateur C, Visual Basic, ou adapter une page Web au format SiteLinker.

### La section Exportation du récepteur FM ressemble à :

```
;$ExportFormatFile contient les définitions au format $ExportFile
$ExportFormatFile "C:\Program Files\SitePlayer\makeHTML.def"
;$ExportHeaderFile contient les informations d'entête pour $ExportFile
$ExportHeaderFile "C:\Program Files\SitePlayer\htmlheader.htm"
;$ExportFooterFile contient les informations de fin pour $ExportFile
$ExportFooterFile "C:\Program Files\SitePlayer\htmlfooter.htm"
;$ExportFile donne le nom du fichier d'exportation créé pour ces définitions
$ExportFile "C:\Program Files\SitePlayer\SP_Root\receiver_export.htm"
;$Export instruction qui crée le HTML $ExportFile et remise à 0 des paramètres
$Export
;Export définitions pour un fichier formaté Visual Basic
$ExportFormatFile "C:\Program Files\SitePlayer\makevb.def"
$ExportFile "C:\Program Files\SitePlayer\rcvrdefs.bas"
$Export
;Export définitions pour un fichier d'entête en langage C
$ExportFormatFile "C:\Program Files\SitePlayer\makec.def"
$ExportFile "C:\Program Files\SitePlayer\rcvrdefs.h"
$Export
;Export définitions pour un fichier en assembleur
$ExportFormatFile "C:\Program Files\SitePlayer\makeasm.def"
$ExportFile "C:\Program Files\SitePlayer\rcvrdefs.asm"
$Export
```

Des fichiers multiples peuvent être exportés. Chacun aura son propre format et appellera la fonction \$Export. L'exemple définit et crée des fichiers HTML, Visual Basic et assembleur.

## \$ExportFormatFile [filename] none

Définit le fichier à utiliser par la commande \$Export.

Le format du fichier contient des pointeurs sur les objets que SiteLinker utilise. Les objets suivants sont définis pour l'utilisation pendant l'exportation.

^name – renvoie le nom de l'objet.  
^address – renvoie l'adresse de l'objet.  
^type – renvoie le type de l'objet.  
^direction – entrée, sortie ou bidirectionnel.

`^default` – renvoie la valeur de l'objet par défaut.  
`^secure` – Si cette variable est sécurisée ou non.

Par exemple si les objets suivants avaient été définis :

```
ORG 200h  
Object1 db 0  
Object2 dw 47h
```

une définition de fichier avec les états suivants :

```
#define ^name 0x^address /* ^type */
```

serait traduit par un fichier contenant ce texte :

```
#define Object1 0x0200 /* byte */  
#define Object2 0x0201 /* integer */
```

Cela pourrait être utilisé dans un programme C pour localiser les objets de SitePlayer sans toute une dactylographie. Si vous déplacez les objets aux alentours, le programme SiteLinker exportera un nouveau dossier au moment où vous faites le nouveau fichier binaire pour SitePlayer.

Le format du fichier de réception (makeHTML.def) crée une sortie HTML avec cette définition :

```
^name = ^^name<br>
```

Deux accents circonflexes, `^^`, sont nécessaires quand vous voulez qu'une flèche soit écrite dans le fichier d'exportation.

### **\$ExportFile [outputfilename]**

Définit le fichier qui sera utilisé pour sortir les objets créés par la fonction `$ExportFormatFile`. Ce fichier sera seulement créé si il n'y a pas d'erreur et que la fonction `$Export` est exécutée.

Vous pourriez créer le fichier HTML simplement avec les définitions de `ExportHeaderFile`, `ExportFormatFile`, et `ExportFooterFile` :

```
<html>  
<body>  
power = ^power<br>  
select = ^select<br>  
volume = ^volume<br>  
channel = ^channel<br>  
</body>  
</html>
```

### **\$ExportHeaderFile [headerfilename]**

Définit un fichier qui sera copié au début de `$ExportFile` quand le début du processus d'exportation commence.

```
$ExportHeaderFile "c:\definitions\header.htm"
```

Adds c:\definitions\header.htm to the beginning of the \$ExportFile

Example contents of the header file might include the following:

```
<html>  
<body>
```

### **\$ExportFooterFile [footerfilename]**

Définit un dossier qui sera copié à la fin du `$ExportFile` quand le processus de l'exportation se complète. Par exemple, cela pourrait être utilisé pour ajouter un `<</html>` à la fin d'un dossier.

```
$ExportFooterFile "c:\definitions\footer.htm"
```

Ajoute : c:\definitions\footer.htm à la fin de \$ExportFile

Par exemple un fichier pourrait inclure ce qui suit :

Exemple contents of the footer file might include the following:

```
</body>  
</html>
```

### **\$Export**

Commence le processus de création de `$ExportFile` qui utilise le fichier `$ExportFormatFile` comme guide. Les variables de l'exportation seront effacées toutes les fois que la commande `$Export` sera rencontré pour que les multiples formats puissent être exportés par différentes définitions d'exportation.

```
$Export
```

## Création de fichiers Page Web

Les pages Web sont l'interface pour votre application avec SitePlayer. Il n'est pas nécessaire d'avoir JAVA ou le script de Visual Basic pour faire la mise à jour des données. Vous utilisez les techniques classiques pour concevoir les pages Web, et pour rendre vos pages active, vous remplacez les représentations statiques par les noms d'Objets que vous avez créé dans le dossier de définition de SitePlayer. Ce sont ces SiteObjects qui font que SitePlayer est si simple et votre application si puissante et flexible. Vous pouvez utiliser SitePlayerPC pour voir et tester vos pages sans connecter votre application.

1. Construisez un site Web standard avec votre éditeur HTML favori.
2. Changez les codes HTML pour obtenir des données dynamiques en les remplaçant des pointeurs sur les objets qui les représentent.

Le fichier interface (.spi) de SitePlayer ajoute le contrôle de votre site.

### SiteObjects en HTML

Quand vous faites un fichier HTML, vous pouvez créer les données dynamiques en utilisant les propriétés de modification de la page. Ces données sont des variables ayant un type, une dimension, et une valeur initiale. Elles sont les mêmes objets que vous avez défini auparavant dans le fichier de définition de SitePlayer.

Pour utiliser un objet, vous signalez son usage par un accent circonflexe (“^”) suivi par le nom de l'objet et des modificateurs facultatifs. L'exemple suivant définit un objet appelé “debit” (le nom des données n'a jamais d'accent) comme un nombre entier de 2 octets avec une valeur initiale à 0. Si SitePlayer rencontre les caractères “^debit” au moment de l'émission d'une page Web, il substituera immédiatement la valeur de debit par son pointeur. Donc, dans une page Web, vous pourriez écrire cette simple phrase :

**En ce moment le débit binaire est ^debit.**

Si, à ce moment précis, le débit vaut 567, alors SitePlayer vous écrira : **En ce moment le débit binaire est 567.** SitePlayer converti les 2 octets par leur représentation ASCII.

### Modification de SiteObject

Des modificateurs sont disponibles pour autoriser certaines manipulation d'objet par SitePlayer quand il envoie des données. Ces caractères modificateurs sont seulement 0 à 9, S, et P. Ces objets sont modifiés uniquement dans SitePlayer et ne modifient pas l'HTML. Quand SitePlayer produit une page, l'action sera celle décrite dans la table suivante.

Usage de l'objet	Description de l'utilisation
^objet	Affiche l'objet
^objet:n	Affiche n objets numériques depuis la droite vers la gauche, ou n caractères d'une chaîne depuis la gauche vers la droite. S ajoute le signe - ou un espace. P ajoute le signe + ou le signe -. M ajoute “M” pour le signe moins ou “P” pour le signe plus.
^objet+n	Ajoute la quantité n à l'objet puis affiche le résultat.
^objet-n	Soustrait la quantité n à l'objet puis affiche le résultat.
^objet*n	Multiplie l'objet par n puis affiche le résultat.
^objet/n	Divise l'objet par n puis affiche le résultat.
^objet&n	Effectue un ET logique entre n et l'objet puis affiche le résultat.
^objet n	Effectue un OU logique entre n et l'objet puis affiche le résultat.
^objet~n	Effectue un OU exclusif entre n et l'objet puis affiche le résultat.
^objet#n	Effectue un ET logique entre n et l'objet et affiche la vérification si le résultat vaut 0 sinon n'affiche rien.
^objet\$n	Si l'objet = n alors affiche la vérification, sinon rien.
^objet'n	Obtient le nème bit de l'objet compté à parti de la droite, le N° 0 est le 1er bit.

### Sélectionner le Chiffre Particulier d'un Objet

^debit:3 choisit le 3ème chiffre de debit, en commençant par la droite. Ainsi si debit vaut 567 alors ^debit:3 donne le caractère ASCII “5”. En isolant des chiffres c'est ensuite facile de simuler des afficheurs avec des images.

#### Exemple pour :

```
<img Src="/images/LED^debit:3.gif">
```

```
<img Src="/images/LED^debit:2.gif">
```

```
<img Src="/images/LED^debit:1.gif">
```

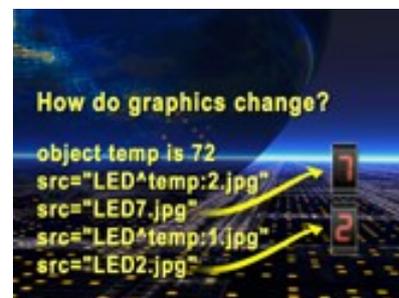
#### SitePlayer écrira la chaîne :

```
<img Src="/images/LED5.gif">
```

```
<img Src="/images/LED6.gif">
```

```
<img Src="/images/LED7.gif">
```

Pour afficher des images de chiffre de 0 à 9, (qui sont d'ailleurs fournies avec le kit de développement de SitePlayer), vous pourrez afficher des nombres qui changent.



### **Calculs mathématiques simples sur un objet.**

`^debit+5` ajoute 5 à la valeur et l'émet. Dans l'exemple `debit+5` donnerait 572.

Il y a les opérateurs mathématiques :

- addition +
- soustraction -
- multiplication \*
- division /
- ET &
- OU |
- OU exclusif ~

### **Choix d'un bit particulier dans un objet**

`^debit'13` choisit le bit 13 d'un objet en comptant le bit 0 pour le poids le plus faible et en allant vers la gauche.

`^debit'30` n'est pas valide et renverra la valeur 0 si la donnée n'a que 16 bits. Les numéros de bits ne sont valables qu'entre 0 et 31.

### **Éviter qu'un objet soit en conflit avec le texte**

Il y a des cas où il y a un risque de confusion entre le texte et les modificateurs d'objet. L'accent circonflexe comme les guillemets sont utilisés pour arrêter l'analyse chaque fois que SitePlayer pourrait confondre. Les fichiers sont un bon exemple où cela arrive fréquemment.

Exemple où `^debit` vaut 544:

```
/^debit/32/file.xxx      donne      /17/file.xxx
```

Comment SitePlayer peut-il savoir si `/32` est un diviseur de du débit ou un sous répertoire ? En fait vous ne pouvez pas le dire en regardant le texte. Dans ce cas SitePlayer suppose que vous avez l'intention de diviser le débit par 32 à moins que vous ne forciez l'objet à terminer avec un autre accent circonflexe.

```
/^debit^/32/file.xxx      donne      /544/32/file.xxx
```

Le même problème se produit avec des chiffres qui vont ensemble.

```
^debit+57856      donne      58400
```

```
^debit+5^7856      donne      5497856
```

Deux accents circonflexes, `^^`, sont nécessaires quand vous voulez en afficher un seul, `^`.

SiteObject `^^débit` = `^debit`      donne      SiteObject `^débit` = 544 (ici `débit` peut prendre un accent quand il n'est pas une variable mais un texte à afficher.)

### **Pages Web de SitePlayer**

Créez des pages Web standards pour agir sur votre application. SitePlayer inclut des simples fichiers HTML et des contrôles graphiques pour vous aider à créer un site Web attirant et actif. Les exemples inclus donnent une idée des fichiers et des techniques de SitePlayer. La racine principale des fichiers pour votre application est définie par `$$Sitepath` quand vous assemblez le site. Tous les fichiers composant le site doivent être inclus dans ce répertoire devenu racine principale. Pour chaque nouveau projet il vous faut un nouveau répertoire racine différent. A l'assemblage des erreurs sont déclarées quand il y a une différence entre les fichiers .HTM, .HTML, .XML, et .SPI déclarés et les fichiers existant dans le répertoire racine. Le fichier de première page Web doit s'appeler `index.htm`. Au moment de la conception, bien penser qu'il n'y a que 48K d'espace mémoire. La taille est affichée par SiteLinker quand il a fait le fichier SPB. Chaque caractère, espace, retour à la ligne, retour chariot, et nom de fichier feront une différence sur le total.

### **Fichier d'interface de SitePlayer**

Le fichier d'interface (.SPI) contient la méthode d'envoi des données de navigateur Web pour SitePlayer. Vous pouvez lui donner n'importe quel nom de fichier. Utilisez le dans vos pages Web avec vos fichiers d'interfaces « Common Gateway Interface file (.CGI) ». Placez des hyperliens avec les noms des objets SitePlayer que vous voulez pour attirer le regard, ou agir sur un nom de fichier qui devra être activé. Il peut aussi contenir des instructions HTML pour rediriger votre navigateur par l'appel d'un lien. Utilisez SiteObjets pour des redirections autorisées.

### **Créer un fichier d'interface de SitePlayer**

Vous pouvez faire le fichier `sample.spi` qui contient la prochaine page Web à afficher après que le point actif ait été modifié. Ceci redirige le navigateur vers la page qui vient juste d'être faite avec la nouvelle information mise à jour.

Pour faire une page de redirection, les commandes suivantes doivent être utilisées dans votre page `sample.spi` :

```
HTTP/1.0 302 Found
```

```
Location: /URL
```

```
<cr><lf>
```



Dans le cas d'un bouton radio, un champs non caché est nécessaire mais vous devez toutefois savoir si le bouton a été sélectionné ou pas. Si l'objet est une valeur binaire, les boutons renverront une sélection, autrement, vous devez placer le # ou "\$" SiteObject dans l'étiquette HTML du bouton radio :

```
<input type="radio" name="volume" value="0" ^volume$0>
```

Sélectionner une liste/menus provoque toujours quelque chose. Si elle n'est pas choisie, la première option choisie dans la liste sera transmise. Pour s'assurer que la valeur de l'objet de votre application n'a pas changé, sauf si vous voulez faire autrement, vous devriez utiliser l'option suivante, valeur fréquemment utilisée par SiteObjects :

```
<select name="channel" size="1">
<option value="^channel">^channel:3^^channel:2.^channel:1</option>
<option value="929">KWFm</option>
</select>
```

Quand la liste du menu « List/Menu » est affichée, le premier choix montrera le canal par défaut. Dans la prochaine section il y a des détails supplémentaires de SiteObjects. Quand vous définissez une forme, vous créez un nom de données et style particulier. Un grand avantage est que SitePlayer ne se soucie pas de comment arrivent les données du moment que le format lui convient. Vous pouvez essayer les démonstrations du Récepteur audio contenus dans le Kit de SitePlayer pour voir les différentes possibilités d'interaction avec une application. Voici un échantillon qui montre la forme et les concepts de SiteObject :

```
<html>
<head>
<title>Untitled Document</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
</head>
```

```
<body bgcolor="#CCFFFF">
<form method="get" action="fi.spi" name="forminput">
<h1>^channel:3^^channel:2.^channel:1</h1>
<input type="hidden" name="power" value="0">
<br>
```

**Arrêt du récepteur :**

```
<input type="checkbox" name="power" value="1" ^power#1>
<br>
<br>
```

Volume 0 through 7<br>

```
<input type="radio" name="volume" value="0" ^volume$0>
<input type="radio" name="volume" value="1" ^volume$1>
<input type="radio" name="volume" value="2" ^volume$2>
<input type="radio" name="volume" value="3" ^volume$3>
<input type="radio" name="volume" value="4" ^volume$4>
<input type="radio" name="volume" value="5" ^volume$5>
<input type="radio" name="volume" value="6" ^volume$6>
<input type="radio" name="volume" value="7" ^volume$7>
<br>
<br>
```

**Choisissez votre station :**

```
<select name="channel" size="1">
<option value="^channel">^channel:3^^channel:2.^channel:1</option>
<option value="929">KWFm</option>
<option value="937">KRQQ</option>
<option value="949">KMXZ</option>
<option value="961">KLPX</option>
</select>
```

```
<br>
<br>
<input type="radio" name="select" value="1" ^select#1> AM<br>
<input type="radio" name="select" value="2" ^select#2> FM<br>
<input type="radio" name="select" value="4" ^select#4> Tape<br>
<input type="radio" name="select" value="8" ^select#8> CD<br>
<input type="radio" name="select" value="16" ^select#16> Aux<br>
<br>
```

```
<input type="submit" name="Submit" value="Submit">
<br>
</form>
</body>
</html>
```

## Quelques règles de HTML :

### Structure d'un programme html :

Les balises sont décalées pour savoir où on en est dans le programme.

Il y a une balise de début et une balise de fin pour encadrer chaque bloc de programme.

```
<html>
  <head>
    <meta>
    <title></title>
  </head>
  <body>
    <h1></h1>
    <p><em></em></p>
    <ul>
      <li></li>
      <li></li>
    </ul>
    <h2></h2>
    <p><abbr></abbr><em></em></p>
  </body>
</html>
```

### La balise INPUT :

La balise INPUT est la balise essentielle des formulaires, car elle permet de créer un bon nombre d'éléments "interactifs". La syntaxe de cette balise est la suivante :

```
<INPUT type="Nom du champ" value="Valeur par défaut" name="Nom de l'élément">
```

### Quelques attributs :

**name** : est l'attribut essentiel car il permettra au script CGI (*Common Gateway Interface - Interface générale de passerelle*) de connaître le champ associé à la paire nom/valeur, c'est-à-dire que le nom du champ sera suivi du caractère "=" puis de la valeur entrée par l'utilisateur, ou dans le cas contraire de la valeur par défaut repéré par l'attribut value.

**maxlength="Valeur"** (integer) attribut de "text" ou "password" définit le nombre maximum de caractères du champ de saisie.

**size="Valeur"** définit la longueur du champ de saisie.

**type** : est l'attribut qui permet de préciser le type d'élément que représente la balise INPUT.

### Voici les valeurs que " type " peut prendre :

- **checkbox**: il s'agit de cases à cocher pouvant admettre deux états : checked (coché) et unchecked (non coché). Lorsque la case est cochée la paire nom/valeur est envoyée au CGI
- **hidden**: il s'agit d'un champ caché. Ce champ non visible sur le formulaire permet de préciser un paramètre fixe qui sera envoyé au CGI sous forme de paire nom/valeur
- **file**: il s'agit d'un champ permettant à l'utilisateur de préciser l'emplacement d'un fichier qui sera envoyé avec le formulaire. Il faut dans ce cas préciser le type de données pouvant être envoyées grâce à l'attribut ACCEPT de la balise FORM
- **image**: il s'agit d'un bouton de soumission personnalisé, dont l'apparence est l'image situé à l'emplacement précisé par son attribut SRC
- **password**: il s'agit d'un champ de saisie, dans lequel les caractères saisis apparaissent sous forme d'astérisques afin de camoufler la saisie de l'utilisateur
- **radio**: il s'agit d'un bouton permettant un choix parmi plusieurs proposés (l'ensemble des boutons radios devant porter le même attribut name. La paire nom/valeur du bouton radio sélectionné sera envoyé au CGI. Un attribut checked pour un des boutons permet de préciser le bouton sélectionné par défaut
- **reset**: il s'agit d'un bouton de remise à zéro permettant uniquement de rétablir l'ensemble des éléments du formulaire à leurs valeurs par défaut
- **submit**: il s'agit du bouton de soumission permettant l'envoi du formulaire. Le texte du bouton peut être précisé grâce à l'attribut value
- **text**: il s'agit d'un champ de saisie permettant la saisie d'une ligne de texte. La taille du champ peut être définie à l'aide de l'attribut size et la taille maximale du texte saisi grâce à l'attribut maxlength

### La balise TEXTAREA :

La balise TEXTAREA permet de définir une zone de saisie plus vaste par rapport à la simple ligne de saisie que propose la balise INPUT. Cette balise possède les attributs suivants :

- **cols**: représente le nombre de caractères que peut contenir une ligne
- **rows**: représente le nombre de lignes
- **name**: représente le nom associé au champ, c'est le nom qui permettra d'identifier le champ dans la paire nom/valeur
- **readonly**: permet d'empêcher l'utilisateur de modifier le texte entré par défaut dans le champ

- **value:** représente la valeur qui sera envoyée par défaut au script si le champ de saisie n'est pas modifié par une frappe de l'utilisateur

### La balise SELECT :

La balise SELECT permet de créer une liste déroulante d'éléments (précisés par des balises OPTION à l'intérieur de celle-ci). Les attributs de cette balise sont :

- **name:** représente le nom associé au champ, c'est le nom qui permettra d'identifier le champ dans la paire nom/valeur
- **disabled:** permet de créer une liste désactivée, c'est-à-dire affichée en grisée
- **size:** représente le nombre de lignes dans la liste (cette valeur peut être plus grande que le nombre d'éléments effectifs dans la liste)
- **multiple:** marque la possibilité pour l'utilisateur de choisir plusieurs champs dans la liste

### Création des tableaux

La création des tableaux se passe autour de 4 balises :

- **<table>** : Définit au navigateur la création d'un tableau
- **<th>** : Définit l'en-tête de chaque colonne
- **<tr>** : Définit une ligne
- **<td>** : Définit une cellule

A noter que la balise <caption> qui affiche le titre du tableau possède 2 attributs :

- **top** : Titre placé au-dessus du tableau (par défaut)
- **bottom** : Titre placé en dessous du tableau

Bien entendu, toutes ces balises doivent être fermées pour indiquer la fin du tableau, d'une colonne ou d'une cellule.

**Note :** Si vous voulez définir une police de caractère pour le tableau, vous êtes obligé d'en définir pour chaque cellule même si elle est identique, exemple ci-dessous.

```
<td><font face="arial" size=1>Cellule 1</font></td>
```

### Exemple d'un tableau avec son code source :

```
<table border="1"><caption><span class="texte"><b> Légende du
tableau</b></span></caption><tbody><tr><th><span class="texte">Colonne 1</span></th><th><span
class="texte">Colonne 2</span></th></tr><tr><td><span class="texte">Cellule 1</span></td><td><span
class="texte">Cellule 2</span></td></tr><tr><td><span class="texte">Cellule 3</span></td> <td><span
class="texte">Cellule 4</span></td></tr></tbody></table>
```

### Code :

```
<table border=1>
  <caption>Légende du tableau</caption>
  <tr>
    <th>Colonne 1</th>
    <th>Colonne 2</th>
  </tr>
  <tr>
    <td>Cellule 1</td>
    <td>Cellule 2</td>
  </tr>
  <tr>
    <td>Cellule 3</td>
    <td>Cellule 4</td>
  </tr> </table>
```

### Empêcher le passage à la ligne dans les cellules :

```
<th nowrap>cellule d'entête</th>
<td nowrap>cellule de données</td>
```

### Créer un lien :

#### lien vers un fichier à télécharger :

```
<A HREF="fichier.exe"> cliquez ici pour télécharger </A>
```

#### lien e-mail :

```
<A HREF="mailto:oli.ert@caramail.com">adresse_e_mail</A>
```

avec sujet :

```
<A HREF="mailto:oli.ert@caramail.com?subject=renseignement">adresse</A>
```

avec sujet et texte dans le corps du message :

```
<A HREF="mailto:oli.ert@caramail.com?subject=renseignement&BODY=votre message">adresse_e_mail :</A>
```

avec sujet, envoyé à 2 personnes :

```
<A HREF="mailto:oli.ert@caramail.com&cc=webmaster@lemel.fr &subject=renseignement">adresse_e_mail</A>
```

## Les formulaires :

Ils sont délimités par la balise <form> ... </form>; C'est une balise qui permet de regrouper plusieurs éléments de formulaire (boutons, champs de saisie,...) et qui possède les attributs obligatoires suivants :

**METHOD** indique sous quelle forme seront envoyées les réponses.

- « **post** » est la valeur qui correspond à un envoi de données stockées dans le corps de la requête.
- « **get** » correspond à un envoi des données codées dans l'URL.

**ACTION** indique l'adresse d'envoi (script CGI ou adresse email exemple: (mailto:adresse.email@machine))

**ENCTYPE** est un attribut facultatif qui spécifie le codage des données dans l'URL; il n'est toutefois pas nécessaire de le préciser car la valeur attribuée par défaut (application/x-www-form-urlencoded) est la seule valeur valide.

**ACCEPT** est un attribut facultatif qui permet de définir les types MIME des données pouvant être envoyées par le formulaire.

## Exemples de formulaires récupérés sous forme de courrier électronique avec une structuration prédéfinie :

```
<form method="post" ACTION="mailto:o.ertzscheid@lemel.fr">
votre nom : <BR>
<INPUT TYPE="text" NAME="nom"> <BR>
votre adresse : <BR>
<TEXTAREA NAME="adresse" ROWS=2 COLS=35></TEXTAREA> <BR>
<INPUT TYPE="submit" VALUE="OK">
<INPUT TYPE="reset" VALUE="cancel">
</form>
```

## Exemples de syntaxe sur les entrées sorties :

Affichage des états logiques

```
^io3 ^io2 ^io1 ^io0
```

Affichage avec des LED0);"

```




```

Bouton de permutation

```
<a href="Test.spi?io3=^io3~1"></a>
<a href="Test.spi?io2=^io2~1"></a>
<a href="Test.spi?io1=^io1~1"></a>
<a href="Test.spi?io0=^io0~1"></a>
```

Bouton de forçage au 1

```
<a href="Test.spi?io3=1"></a>
<a href="Test.spi?io2=1"></a>
<a href="Test.spi?io1=1"></a>
<a href="Test.spi?io0=1"></a>
```

Bouton de forçage au 0

```
<a href="Test.spi?io3=0"></a>
<a href="Test.spi?io2=0"></a>
<a href="Test.spi?io1=0"></a>
<a href="Test.spi?io0=0"></a>
```

## Exemples non liés :

```
<input type="submit" value="Modifier">
<input name="Voie1" size="2" maxlength="2" value="^Voie1 ^Voie1 type="text">Bassin
```

## Redirection ou rafraîchissement :

Pour rediriger ou rafraîchir une page à intervalle régulier on utilise la balise <meta http-equiv="refresh">

Pour recharger automatiquement une page toutes les X secondes, et provoquer ainsi un rafraîchissement régulier, c'est la balise <meta http-equiv="refresh">.

Par exemple : <meta http-equiv="refresh" content="3"> pour 3 secondes.

Cette balise doit être insérée entre les balises <HEAD> et /HEAD> de la page.

Pour rediriger les visiteurs vers une autre adresse URL on utilise la balise <meta http-equiv="refresh"> de la manière suivante:

```
<meta http-equiv="refresh" content="0;URL=http://www.le nouveausite.com">
```

Le "0" juste avant "URL" détermine, en secondes, le temps d'attente avant la redirection. Ainsi <meta http-equiv="refresh" content="3;URL=http://www.le nouveausite.com">Redirigera votre visiteur sur le site http://www.le nouveausite.com au bout de 3 secondes.

Cette balise doit être insérée entre les balises <HEAD> et /HEAD> de la page. Cette pratique n'est cependant pas très appréciée par les robots des sites de référencement.

**A essayer aussi ce bout de programme récupéré sur une page web :**

```
<script language="javascript">
var timeoutID,stop_ok=0;
function LoadPage(){
    window.document.location.reload();
}
function Timer(){
    timeoutID = window.setTimeout("LoadPage()",5000*60);
}
function Stop(){
    if (stop_ok==0){
        window.clearTimeout(timeoutID);
        stop_ok=1;
    }
}
</script>
```

**Aides :**

[http://www.htmlcodetutorial.com/forms/\\_INPUT.ht](http://www.htmlcodetutorial.com/forms/_INPUT.ht)

<http://fr.selfhtml.org/navigation/html.htm>

<http://www.allhtml.com/articles/detail/369>

## Transmission et réception série :

C'est simple. Ce port permet toutes les transmissions dont vous avez besoin. Si votre système n'a pas de port série, vous pouvez sortir la broche pour une extension future. Il faut moins de 128 octets si le code est dans le 8051 de SitePlayer ou un microcontrôleur externe muni d'un port série. La vitesse par défaut ou après une initialisation est 9600 Bauds. En utilisant ComParams on peut choisir de 110 à 115 200 bauds. Les vitesses classiques sont disponibles.

## **ATTENTION** cette erreur bloque définitivement SitePlayer :

Ne faites jamais de transmission série pendant le téléchargement d'un nouveau programme Ethernet ou d'une nouvelle page Web. Si vous avez besoin de le faire, le client doit interrompre la transmission série pendant cette mise à jour. Cela peut être un arrêt des transmission et réception, ou la déconnexion par cavaliers de la liaison, entre le microcontrôleur et SitePlayer.

### La trame :

Une trame comporte :

Une commande (1 octet) + Une adresse (1 ou 2 octets) + des données (0 à 16 octets)

### Principe d'une commande série :

L'octet de commande est constitué en 2 parties. Les 4 bits de poids forts sélectionnent une des 16 fonctions et les 4 bits de poids faibles indiquent le nombre d'octets qui seront transmis ou reçus.

Octet de commande							
Valeur de commande				Nombre d'octets envoyé/demandé			
Jusqu'à 16 commandes différentes				1 à 16 octets (0 pour 1, 1 pour 2, ..... , 15 pour 16)			
D7	D6	D5	D4	D3	D2	D1	D0

### Les codes de commande série :

Commandes de SitePlayer		
Commande	Octet de commande	Description
NOP	00h	Ne fait rien, Commande d'initialisation recommandée
Status	10h	Renvoie l'état de SitePlayer
Reset	20h	Remise à 0 du chien de garde
ComParams	33h	Positionne la vitesse de transmission baud et le retard de l'UART
UDPSend	50h	Envoi un paquet UDP
Read	0C0h	Lecture d'un objet de SitePlayer
Write	80h	Écriture d'un objet de SitePlayer
ReadX	0D0h	Lecture utilisant 2 octets en adressage étendu
WriteX	90h	Écriture utilisant 2 octets en adressage étendu
ReadBit	0E0h	Lecture d'un bit , Adresse de 1 octet
WriteBit	0A0h	Écriture d'un bit , Adresse de 1 octet
ToggleBit	0B0h	Permutation d'un bit , Adresse de 1 octet

### **NOP ( commande 00h) :**

Cette commande n'exécute aucune opération. L'octet vaut 00h.

Si on envoie 20 octets NOP, ceci permet à SitePlayer d'être sur une fin de message, donc de se synchroniser. Il est recommandé d'initialiser les communications de SitePlayer avec les commandes NOP.

### **Status ( commande 10h) :**

Cette commande envoie le registre d'état de SitePlayer. Il faut envoyer l'octet 10h puis recevoir l'octet du registre d'état. Après l'envoi, ce registre est remis à 0 dans SitePlayer.

Registre d'état transmis							
Submit	Bit 254	Bit 253	Bit 252	Bit 251	Bit 250	Bit 249	Bit 248
D7	D6	D5	D4	D3	D2	D1	D0

Le bit de poids le plus fort est mis à 1 quand une donnée a été soumise à SitePlayer sans se soucier si un objet a été modifié ou pas. Les autres bits peuvent être activés un par un par votre page Web pour aider un processus courant.

### Exemple :

Premièrement créez un objet à l'une des adresses du bit d'état.

Org 248

AM\_page            dbit 0    ; active le bit d'état avec AM\_page

Alors, placez-le dans un champ caché de votre page de Web.

<input type= »hidden » name= »AM\_page » value= »1 »>

Maintenant, toutes les fois que vous vérifiez le registre d'état, vous pouvez déterminer si toutes les données AM\_page ont été envoyés et agir en conséquence.

### **Reset ( commande 20h ) :**

Cette commande réinitialise SitePlayer. Elle arrête les boucles d'interruption et le timer du chien de garde. Tout se passe comme à une nouvelle mise sous tension. Cela suppose que SitePlayer est toujours actif, sinon il faut utiliser la broche Reset avec un microcontrôleur.

### **ComParams ( commande 33h ) :**

Cette commande initialise la vitesse et le temps de réponse. Quand un processeur envoie puis reçoit des données, il peut être nécessaire qu'il y ait un temps d'attente entre les deux pour ne pas manquer de caractères. Toutes les fois que SitePlayer est réinitialisé, il se règle à 9600 baud et 300 microsecondes d'attente. Le temps d'attente maximum est 19.66 millisecondes. ComParams permet de modifier l'un ou l'autre des paramètres. La commande nécessite 4 octets. Les deux premiers pour la vitesse (les bauds) avec l'octet de poids faible en premier puis l'octet de poids fort. Utiliser cette équation pour le réglage :

65536 - (1250000 / Vitesse en Baud)

Pour 115200 baud on a :

65536 - (1250000 / 115200) = 65525 = 0FFF5h

Les 2 octets suivants règlent le temps d'attente est sont déterminés par l'équation : 65536 - (temps en µS \* 3.333333)

Pour une attente de 1000 µS on a :

65536 - (1000 \* 3.333333) = 62203 = 0F2F6h

La commande suivante est complète et initialise à une vitesse de 115200 baud et une attente de 1000 µS.

33h 0F5h 0FFh 0F6h 0F2h

### **Write or WriteX ( commande 80h ou 90h ) :**

Cette commande écrit des données dans SitePlayer. Write utilise un seul octet pour couvrir des adresses de 0 à 255 (00FFh), tandis que WriteX utilise 2 octets ce qui permet des adressages étendus de 0 à 65535 (0FFFFh). Les différents modèles de SitePlayer ont différentes tailles de mémoire objet. Les plus petits ont 768 octets dont les adresses vont de 0 à 2FFh. Les commandes Write ou WriteX contiennent aussi le nombre d'octets à transmettre depuis 1 à 16 dans les 4 bits de poids faible de l'octet de commande. Pour la donnée 0 il y aura 1 octet et pour la donnée 15 il y en aura 16.

#### **Exemple de transmission :**

Envoi du nombre 1023 (03FFh) à l'objet d'adresse 47h. La commande d'écriture est :

Write+2-1=81h            47h            0FFh            03h

Donc la commande est 80h à laquelle on ajoute 2 pour 2 octets à transmettre et à laquelle on enlève 1 pour que la donnée 0 ait la valeur 1.

Plus attentivement on a sur le quartet de poids fort la commande 8 et sur le quartet de poids faible le compteur des octets avec le principe : (0=1, 1=2,...15=16). Ceci donne 81h.

Puis les données sont envoyées en commençant par les poids faibles. 1023 donne en hexadécimal 3FFh, soit 3 \* 256 + 255.

La donnée pourrait aussi être transmise par 5 octets comme cette écriture en mode étendu.

On envoie donc le nombre 03FFh à l'adresse 0047h :

WriteX+2-1=91h            47h            00h            0FFh            03h

Les adresses sont aussi transmises par l'octet de poids faible en premier.

### **Read or ReadX ( commande 0C0h ou 0D0h ) :**

Reçoit les données dans SitePlayer par la broche d'entrée série, générée localement, telles que les adresses IP ou par des données antérieurement envoyées à SitePlayer par les commandes Write ou WriteX. Comme pour les commandes Write ou WriteX, Read ou ReadX ont un compteur de caractères de 1 à 16 associé à 4 bits de commande pour lire 1 ou 2 octets de donnée. Pour lire 2 octets à l'adresse 47h la commande doit être :

Read+2-1 = C1h            47h

et SitePlayer répondra par les 2 octets depuis l'adresse 47h, la plus petite adresse en premier. Donc les 2 octets envoyés sont ceux d'adresses 47h et 48h.

### **UDPsend ( commande 50h ) :**

Instruit le processeur du circuit pour envoyer un message UDP à toute les combinaisons d'adresse MAC/IP. Cela rend un circuit équipé de SitePlayer plus flexible pour diffuser des messages à un système informatique. Cette commande peut diffuser des conditions d'alarme, des flux de données, ou périodiquement des paramètres sans avoir besoin d'obtenir des demandes par un système informatique éloigné. Voir la section UPD Send et Receive pour plus d'informations.

### **Reading, Writing, et Toggling Bit Variables ( commande 0E0h ou 0A0h ou 0B0h ) :**

SitePlayer a aussi des variables de 1 bit qui sont intéressantes pour les boîtes de contrôle (checkboxes), les boutons radio (radio buttons), et les relais ou interrupteurs. Au lieu de pénaliser le système par des calculs pour travailler

avec 1 bit dans un octet, SitePlayer possède 256 bits de variables internes. Par exemple si vous aviez dû changer le bit dans une application sans ces variables, vous auriez dû lire l'octet, modifier le bit que vous vouliez, et alors renvoyer l'octet. Que se passerait-il si l'utilisateur d'un navigateur changeait une de ces variables de 1 bit dans l'octet avant que vous ayez eu le retour de l'octet modifié ? Vous auriez alors n'importe quoi ! Donc SitePlayer exécute les lectures, les modifications et les écritures avec un blocage des interruptions pour qu'il n'y ait aucun problème de ce genre. Physiquement, les variables de 1 bit sont situées aux adresses de 02E0h à 02FFh. SiteLinker alloue automatiquement 32 octets de mémoire dans cette région pour les variables de 1 bit. Si ces variables de 1 bit ne sont pas utilisées, alors cette mémoire peut être utilisée par autre chose. Les commandes bit sont les mêmes que celles de Read et Write, sauf que seule l'adresse de l'octet est convertie en une adresse de bit, avec le bit 0 au poids faible de l'adresse 02E0h et le bit 255 au poids fort de l'adresse 02FFh.

Read renvoie un seul octet à 0 ou 1 suivant l'état du bit.

Write demande un argument 0 ou 1 et active le bit considéré.

ToggleBit nécessite un octet argument et une adresse et il inverse le bit de l'adresse. L'octet argument de ToggleBit est simplement sa position et peut avoir n'importe quelle valeur, mais elle doit être fournie.

#### **Envoi de plus d'un objet avec une commande :**

SitePlayer vous permet de transmettre 1 à 16 octets par commande d'écriture ou de lecture. Cela veut dire que vous pourriez en transférer seize, par une simple commande de 1 octet, ou huit par une commande de deux octet. Ceci permet de réduire les octets de commande mais complique aussi le programme. Si vous avez un grand nombre d'octet à envoyer, le temps peut être le même en utilisant une commande groupée ou des commandes séparées.

**Annexes :**

<b>Table mémoire des objets de SitePlayer</b>	
<b>Adresses</b>	<b>Description</b>
0000h	adresses d'objets de 1 ou 2 octets 02D0h – 02DFh peut aussi envoyer des objets UDP. (Voir table)
02DFh	
02E0h	adresses d'objets de 1 bit dans un octet ou adresses d'objets de 2 octets
02FFh	
0FF00h	Fonctions spéciales de SitePlayer. (Voir table)
0FFFFh	

<b>Table mémoire des fonctions spéciales</b>		
<b>Adresse</b>	<b>Nom</b>	<b>Description</b>
0FF00h	P1	Port 1 - Port 8 bit complet
0FF01h	CMOD	Mode compteur PCA
0FF02h	CCON	Contrôle compteur PCA
0FF03h	CH	Poids fort Compteur PCA
0FF04h	CL	Poids faible Compteur PCA
0FF05h	CCAPM0	Mode du module 0
0FF06h	CCAPM1	Mode du module 1
0FF07h	CCAPM2	Mode du module 2
0FF08h	CCAPM3	Mode du module 3
0FF09h	CCAP0H	Capture poids forts du module 0
0FF0Ah	CCAP1H	Capture poids forts du module 1
0FF0Bh	CCAP2H	Capture poids forts du module 2
0FF0Ch	CCAP3H	Capture poids forts du module 3
0FF0Dh	CCAP0L	Capture poids faibles du module 0
0FF0Eh	CCAP1L	Capture poids faibles du module 1
0FF0Fh	CCAP2L	Capture poids faibles du module 2
0FF00h	CCAP3L	Capture poids faibles du module 3
0FF11h	IO0	Bit numéro 0 du Port 1
0FF12h	IO1	Bit numéro 1 du Port 1
0FF13h	IO2	Bit numéro 2 du Port 1
0FF14h	IO3	Bit numéro 3 du Port 1
0FF15h	IO4	Bit numéro 4 du Port 1
0FF16h	IO5	Bit numéro 5 du Port 1
0FF17h	IO6	Bit numéro 6 du Port 1
0FF18h	IO7	Bit numéro 7 du Port 1
0FF19h	COM	Sortie de l'objet Port série
0FF1Ah	Baud	Objet vitesse de transmission Baud
0FF1Ch	SPIbus	Objet SPI
0FF1Dh	SPIaddr	Pointeur de l'adresse mémoire de la donnée SPI entrante
0FF1Eh	UDPsend	Objet transmission UDP
0FF1Fh	HalfSec	Objet HalfSec
0FFE0h à 0FFE5h	MAC Adress	6 octets MAC Adress Ethernet
0FFE6h à 0FFE9h	Current IP Adress	Adresse courante IP

<b>Table mémoire des envois UDP de SitePlayer</b>		
<b>Adresses</b>	<b>Noms</b>	<b>Description</b>
02D0h	UDPMAC	1er des 6 octets pour une destination MAC adress
02D1h	UDPMAC2	2ème des 6 octets pour une destination MAC adress
02D2h	UDPMAC3	3ème des 6 octets pour une destination MAC adress
02D3h	UDPMAC4	4ème des 6 octets pour une destination MAC adress
02D4h	UDPMAC5	5ème des 6 octets pour une destination MAC adress
02D5h	UDPMAC6	6ème des 6 octets pour une destination MAC adress
02D6h	UDPIP	1er des 4 octets pour une destination adresse IP
02D7h	UDPIP2	2ème des 4 octets pour une destination adresse IP
02D8h	UDPIP3	3ème des 4 octets pour une destination adresse IP
02D9h	UDPIP4	4ème des 4 octets pour une destination adresse IP
02DAh	UDPPORT	2 octets pour une destination d'adresse de PORT
02DCh	UDPADDR	2 octets pour une adresse de départ d'objets à transmettre
02DEh	UDPCOUNT	2 octets pour compter les octets à envoyer 1 à 768

<b>Octet de commande</b>							
<b>Valeur de commande</b>				<b>Nombre d'octets envoyé/demandé</b>			
Jusqu'à 16 commandes différentes				1 à 16 octets (0 pour 1, 1 pour 2, ..... , 15 pour 16)			
D7	D6	D5	D4	D3	D2	D1	D0

<b>Commandes de SitePlayer</b>		
<b>Commande</b>	<b>Octet de commande</b>	<b>Description</b>
NOP	00h	Ne fait rien, Commande d'initialisation recommandée
Status	10h	Renvoie l'état de SitePlayer
Reset	20h	Remise à 0 du chien de garde
ComParams	33h	Positionne la vitesse de transmission baud et le retard de l'UART
UDPsend	50h	Envoi un paquet UDP
Read	0C0h	Lecture d'un objet de SitePlayer
Write	80h	Écriture d'un objet de SitePlayer
ReadX	0D0h	Lecture utilisant 2 octets en adressage étendu
WriteX	90h	Écriture utilisant 2 octets en adressage étendu
ReadBit	0E0h	Lecture d'un bit , Adresse de 1 octet
WriteBit	0A0h	Écriture d'un bit , Adresse de 1 octet
ToggleBit	0B0h	Permutation d'un bit , Adresse de 1 octet

<b>Structure de réception UDP de SitePlayer</b>		
<b>Nom</b>	<b>Taille</b>	<b>Description</b>
Nombre d'octets	1 octet	Nombre d'octets dans une donnée
Félicitation du nombre d'octets ?????	1 octet	Celui qui félicite le nombre d'octets dans une donnée
Adresse donnée	2 octets	Pointeur sur l'adresse d'un objet dans SitePlayer. Octet de poids faible en premier
Données	1 à n octets	Les octets de donnée

## Miniserveur Internet pour gérer son habitat.

### Le principe :

Cela consiste à interroger ou à piloter des appareils électriques dans une habitation à partir d'un réseau Internet. L'accès se fait sur une ou plusieurs pages Web avec ou sans mot de passe.

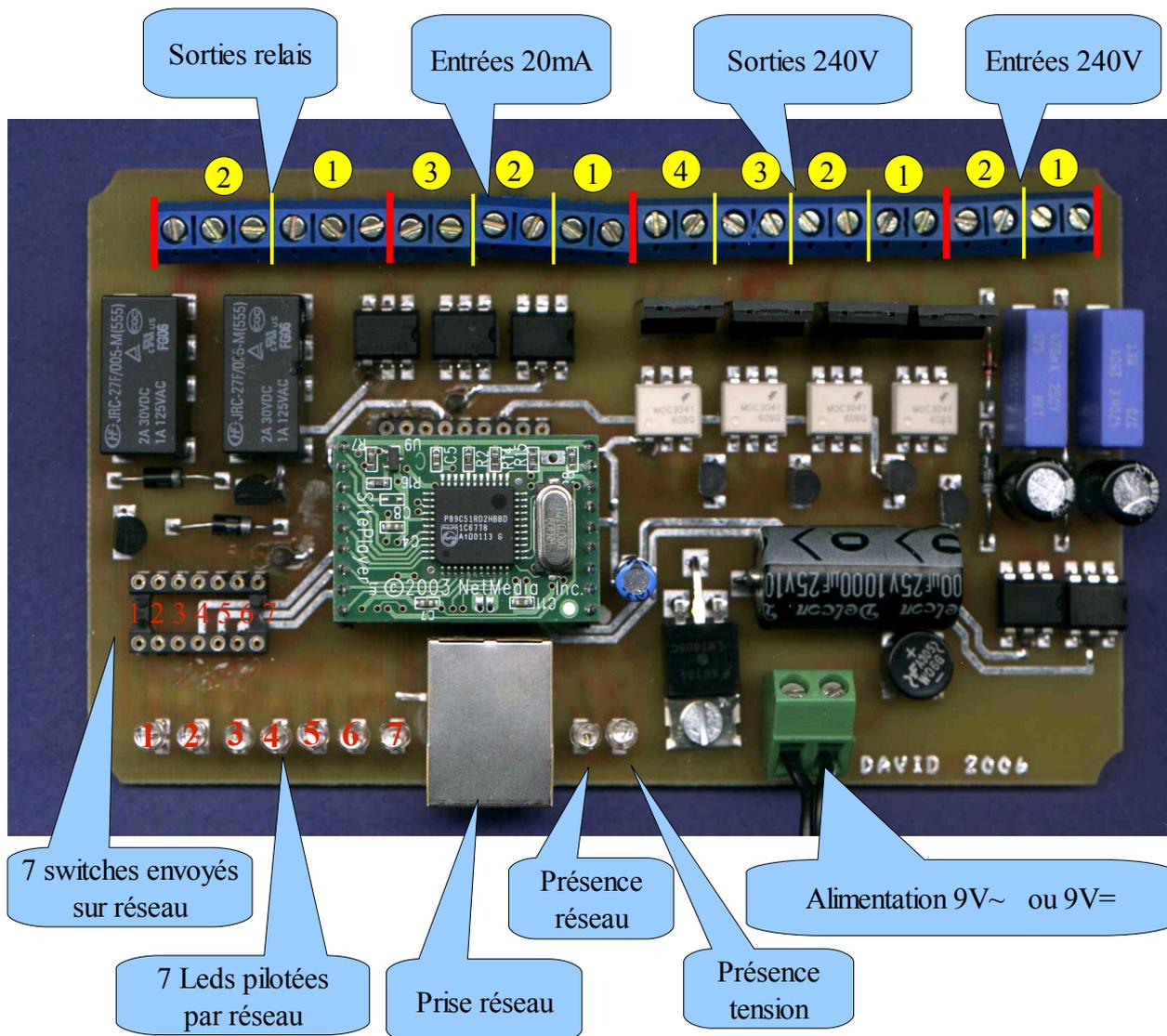
Une carte électronique utilisant le module SitePlayer que l'on peut se procurer ces [Comelec](#) ou chez [TotalRobot](#) dialogue avec microcontrôleur PIC16F628 qui conserve l'état des sorties en cas de rupture d'alimentation momentanée. On peut aussi demander au microcontrôleur de réagir automatiquement en cas d'appel par la page Web, par exemple si on clique sur un bouton de la page Web, on peut couper le courant pendant quelques secondes sur une prise puis la remettre sous tension pour rebooter une machine.

La carte comprend :

- 4 sorties 240V 5A.
- 2 sorties relais inverseurs sur contact sec.
- 2 entrées sous 240V.
- 3 entrées sur LED 20mA boucle de courant Tout ou rien.

Toutes ces entrées et sorties sont isolées galvaniquement les unes par rapport aux autres et par rapport au module SitePlayer.

### La photo de la carte :

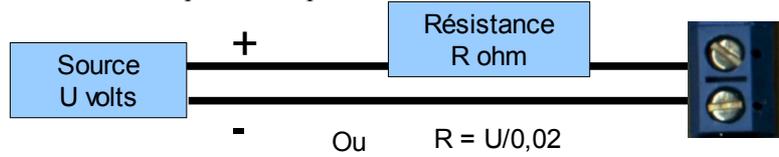


### Utilisation des entrées 20mA :

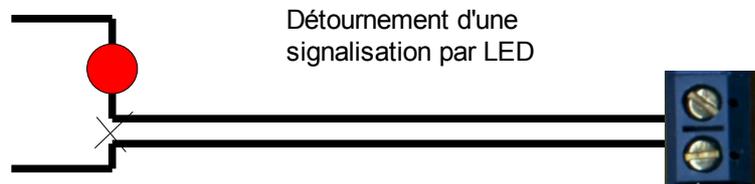
Il faut qu'il passe 20mA soit 0,02 A dans le connecteur de cette entrée pour que la Led du photocoupleur intégré sur la carte électronique détecte que l'entrée est active. **Il y a un sens de raccordement** et si c'est le mauvais sens qui est choisi ça ne fonctionne pas mais il n'y a pas de destruction de la carte électronique. Alors on essaye.

1. On peut alors utiliser une source de tension (pile, tension d'un appareil électronique ou bloc secteur) avec une limitation du courant par résistance. Exemple si la pile fait 1,5V il faut une résistance de  $R = U/I = 1,5/0,02 = 75 \text{ Ohm}$ .

Comme les valeurs normalisées sont 56 ou 82 ohm on choisit celle qu'on a. Les valeurs peuvent être approximatives.



2. On peut aussi dévier le courant d'une signalisation à Led vu que souvent le courant qui la traverse vaut autour de 20mA. Mais cela peut ne pas fonctionner si le courant clignote à grande vitesse, ce qui ne se voit pas à l'oeil mais qui économise de l'énergie ou si le courant est faible comme cela se produit avec les led à faible consommation. Faut essayer sans endommager son appareil électronique.

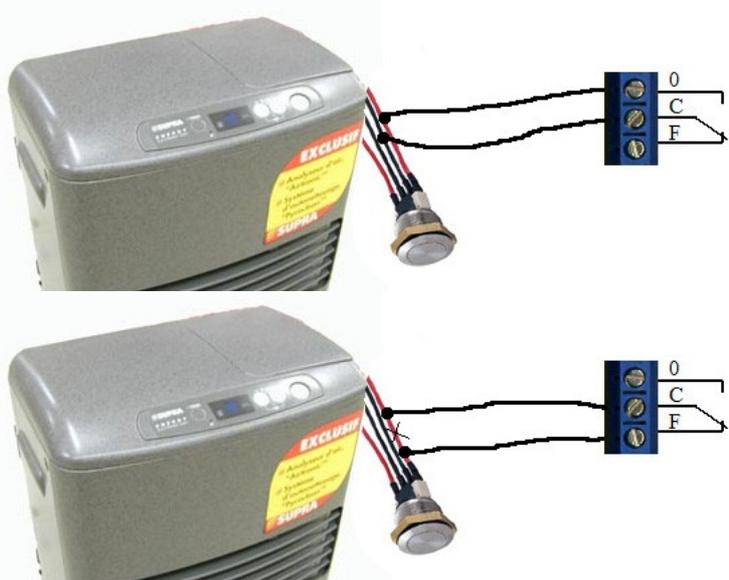


### Sortie à relais :

Cette sortie peut piloter des faibles tensions comme par exemple un bouton poussoir de mise en route d'un poêle à pétrole.

Il y a en général 2 sortes de bouton poussoir :

- Les boutons de mise en route et c'est alors un branchement en parallèle qu'il faut faire. C'est à dire qu'il faut amener les 2 broches du contact à fermeture du relais sur les 2 broches de contact du bouton poussoir. C'est la première photo.
- Les boutons d'arrêt et c'est alors un branchement série qu'il faut faire. C'est à dire qu'il faut couper un fil du bouton poussoir et amener les deux bouts de fils créés sur le contact à ouverture du relais. C'est la deuxième photo.



### Attention au DANGER du raccordement des connecteurs sous 240V :

**C'est super pratique mais hyper dangereux. Il faut absolument faire les raccordements hors tension, et fermer le boîtier avant de remettre le courant. En plus du risque de l'électrocution, il y a le risque d'un flash électrique en cas de mauvais raccordement ce qui pulvérise du métal liquide aux alentours, et attention aux yeux, aux brûlures ...**

**Les schémas proposés montrent des fils comme conducteur, mais dans la réalité il faut utiliser des câbles qui sont une association de fils protégés mécaniquement et électriquement par une gaine extérieure.**

Manipulé avec prudence, ça reste vraiment intéressant.

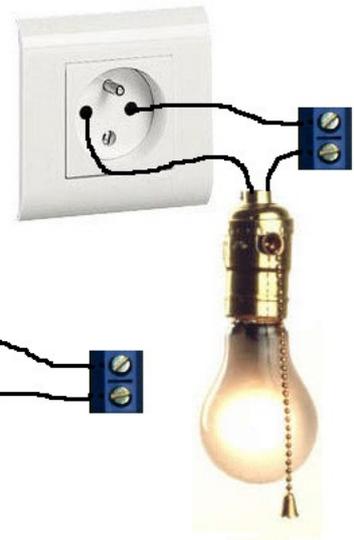
### Utilisation des entrées 240V :

Cette entrée peut directement recevoir du 240V du réseau électrique. Elle peut se brancher directement en parallèle sur une ampoule, ou aux bornes d'une prise de courant pour savoir s'il y a de la tension. Il n'y a pas de risque de court-circuit quand on branche ces 2 fils n'importe où.

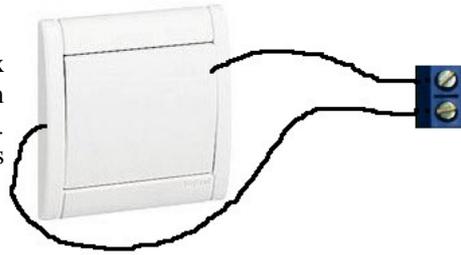


### **Utilisation des sorties 240V :**

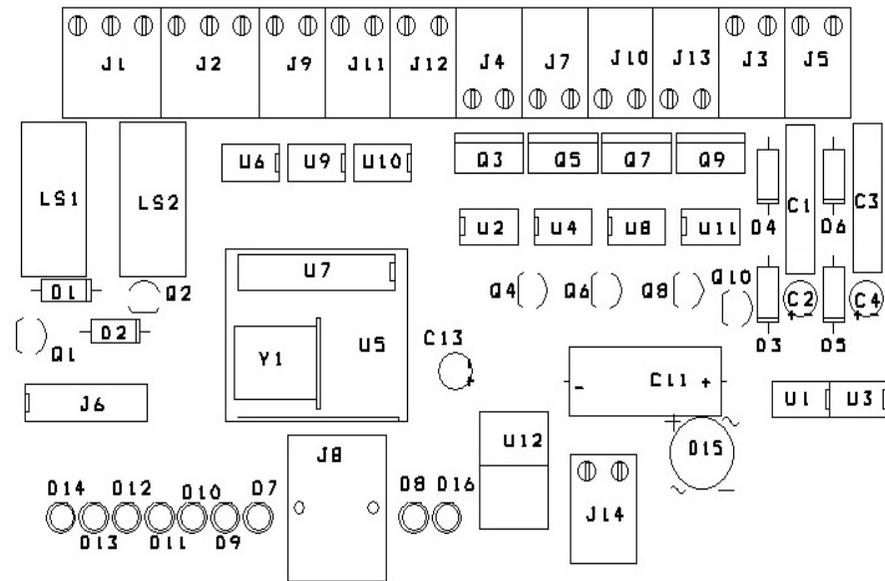
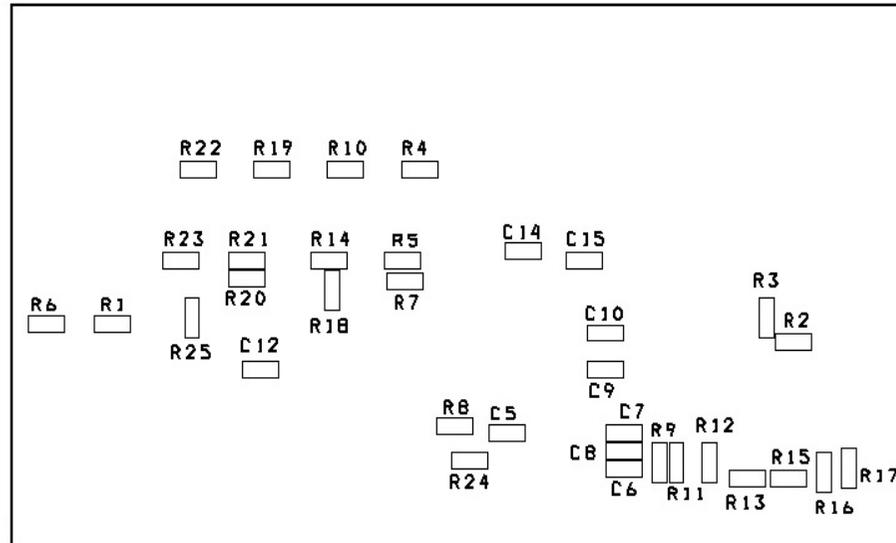
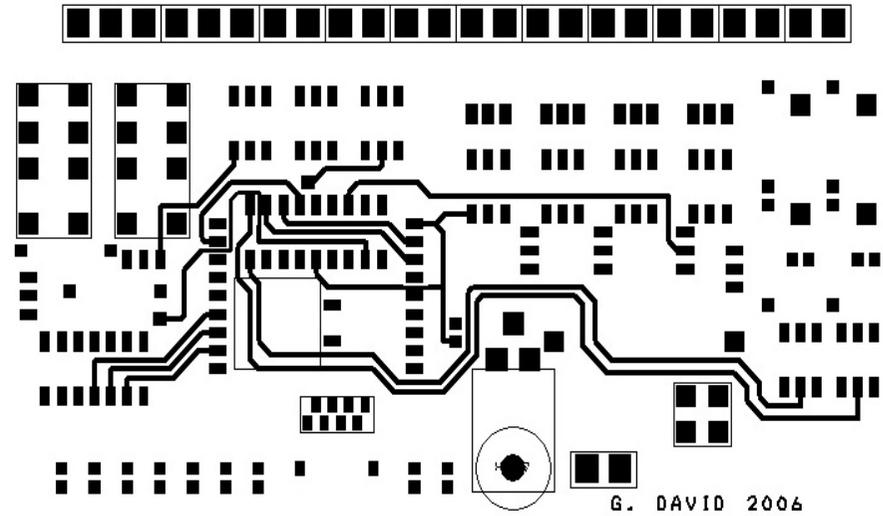
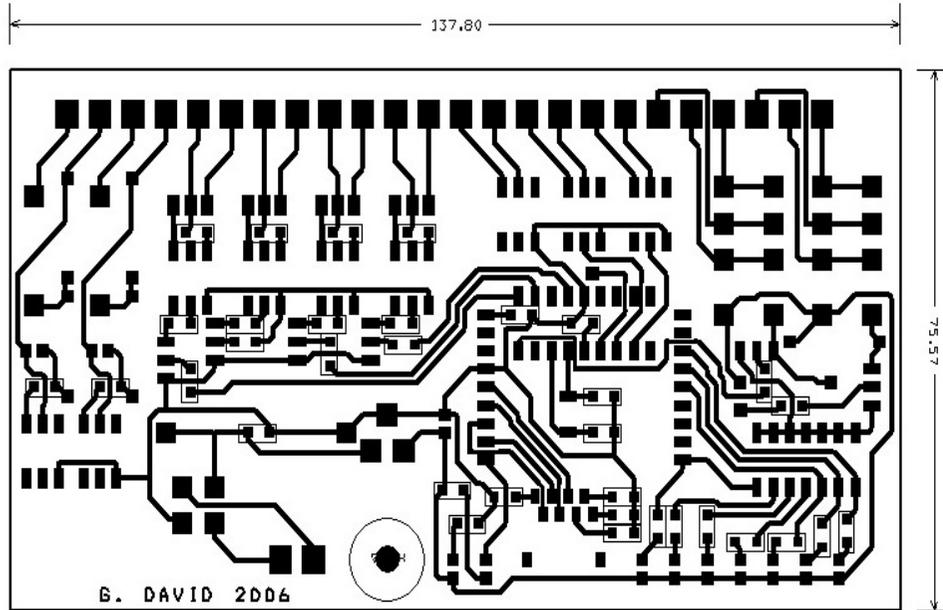
La sortie doit impérativement avoir un récepteur en série comme le montre la photo, sinon c'est le court jus, l'explosion de la carte, la satellisation de la boîte. Cette sortie se comporte comme un interrupteur, elle peut donc allumer ou éteindre la lampe, mais ce peut être une cafetière électrique, ou la télé. Attention ne dépassez pas 800W, sinon ça va chauffer. On peut coller un dissipateur sur les triacs pour aller jusqu'à 1500W.

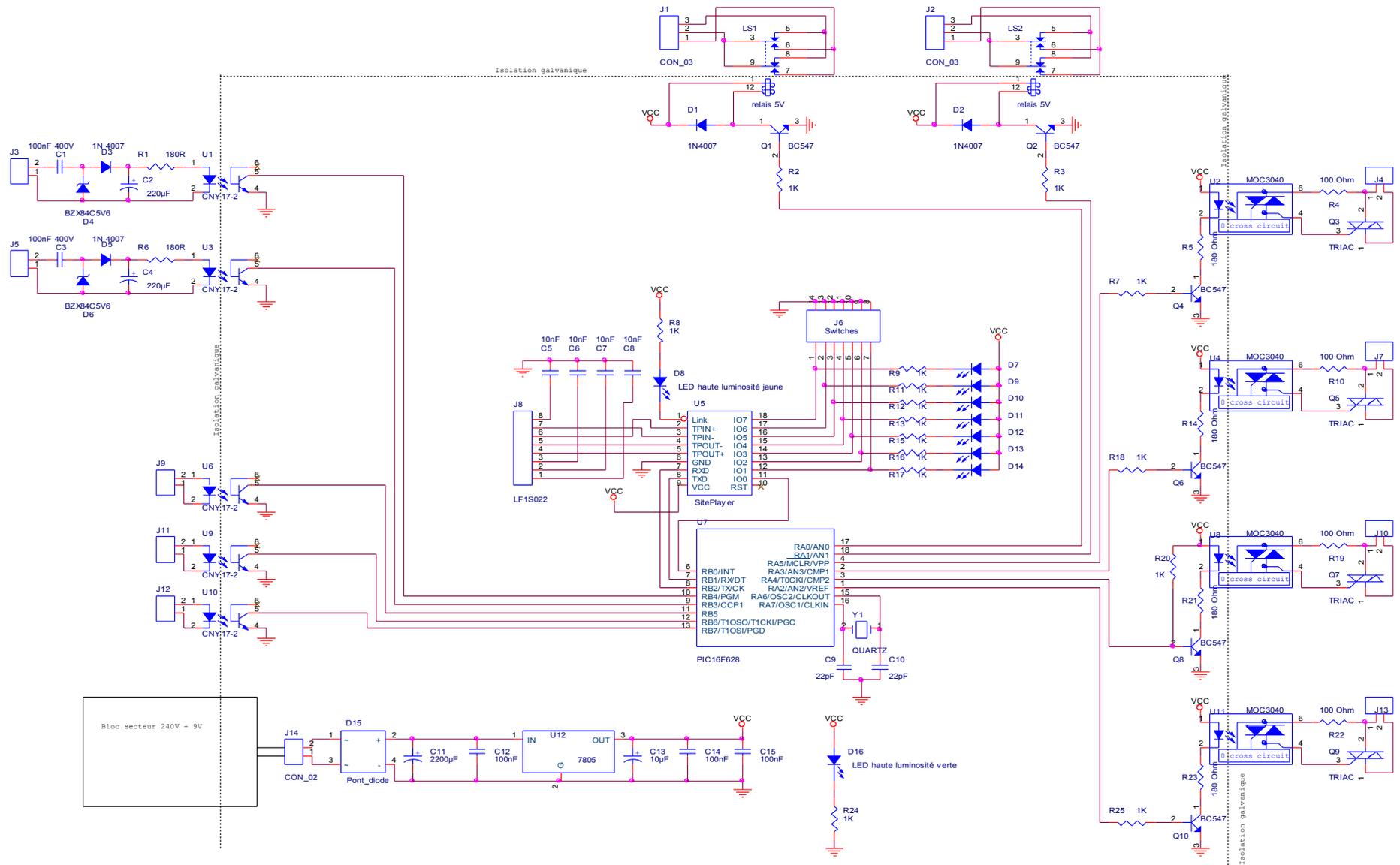


On peut aussi raccorder directement les deux bornes de l'entrée 240V sur les deux bornes d'un interrupteur pour allumer des lampes à distance. Dans ce cas la lampe est déjà comprise dans l'installation.



Les schémas :





Title	<Title>		
Size	Document Number		
A3	<Doc>		<Rev Code>
Date:	Sunday, September 17, 2006	Sheet	1 of 1

**Liste des composants :**

Quantité	Désignation	Composant	Chez	Coût unitaire	Coût total
1		Boîtier ABS 160x95x55mm	SE	4,9€	4,9
10	J3,J4,J5,J7,J9,J10,J11,J12,J13,J14	Bornes empilables pour CI (2b)	ED	0,45€	4,5
2	J2,J1	Bornes empilables pour CI (3b)	ED	0,65€	1,3
2	C1,C3	Condensateur 100nF 400V	ED	0,80€	1,6
1	C13	Condensateur 10µF radial 16V	ED	0,30€	0,3
1	C11	Condensateur 2200µF axial	ED	2,00€	2
2	C4,C2	Condensateur 220µF radial 16V	ED	0,40€	0,8
3	C12,C14,C15	Condensateur CMS 100nF	ED	0,03€	0,09
4	C5,C6,C7,C8	Condensateur CMS 10nF	ED	0,03€	0,12
2	C9,C10	Condensateur CMS 22pF	ED	0,03€	0,06
4	D1,D2,D3,D5	Diode 1N4007	ED	0,50€	2
2	D4,D6	Diode zener BZX5V6	ED	0,30€	0,6
1	D8	LED 2000mcd jaune 5mm	SE	0,20€	0,2
7	D7,D9,D10,D11,D12,D13,D14	LED 2000mcd rouge 5mm	SE	0,20€	1,4
1	D16	LED 2000mcd verte 5mm	SE	0,80€	0,8
1	J6	Micro inter unipolaire 7 contacts	SE	1,90€	1,9
1	U7	Microcontrôleur PIC16F628	SE	4,60€	4,6
1	U5	Module SitePlayer	TR	35,00€	35
5	U1,U3,U6,U9,U10	Photocoupleur CNY17-2	SE	0,40€	2
4	U2,U4,U8,U11	Photocoupleur MOC3041	SE	0,90€	3,6
1	D15	Pont diode 1A	SE	0,40€	0,4
1	J8	Prise Ethernet LF1S022	TR	7,50€	7,5
1	Y1	Quartz 20MHz	SE	1,10€	1,1
1	U12	Régulateur 7805	SE	0,50€	0,5
2	LS2,LS1	Relais 5V	SE	1,90€	3,8
4	R4,R10,R19,R22	Résistance CMS 100Ω	ED	0,03€	0,12
2	R6,R1	Résistance CMS 180Ω	ED	0,03€	0,06
4	R5,R14,R21,R23	Résistance CMS 180Ω	ED	0,03€	0,12
7	R15,R16,R17,R18,R20,R24,R25	Résistance CMS 1KΩ	ED	0,03€	0,21
15	R2,R3,R7,R8,R9,R11,R12,R13,	Résistance CMS 1KΩ	ED	0,03€	0,45
2	U5	SIP 9 broches double mâles	SE	1,15€	2,3
1	U7	Support tulipe 28 broches	SE	0,80€	0,8
9	U1,U3,U6,U9,U10,U2,U4,U8,U11	Support tulipe 6 broches	SE	0,16€	1,44
6	Q1,Q2,Q4,Q6,Q8,Q10	Transistor BC547	SE	0,15€	0,9
4	Q3,Q5,Q7,Q9	TRIAC TO220 isolé	SE	1,2€	4,8
1		Circuit imprimé double face	IP	34,42€	34,42
4		Fiche multiprise	CR	2,00€	8
1		Câble 2x1mm <sup>2</sup> souple	CR	10,00€	10
					0
10	Heures	Main d'oeuvre			0
				Total	144,69

SE = selectronic

ED = électronique diffusion

TR = Total Robot

IP = Imprelec

CR = Carrefour

**Équivalence des entrées sorties :**

Carte	PIC	SitePlayer	index.htm Site.spd	Site.c
Entrée 240V N°1 N°2	RB3 RB4		e24_1 e24_2	E240_1 E240_2
Entrée 20mA N°1 N°2 N°3	RB7 RB6 RB5		e20_1 e20_2 e20_3	E20_1 E20_2 E20_3
Sortie 240V N°1  N°2  N°3  N°4	RA2  RA4  RA3  RA5		s24_1  s24_2  s24_3  s24_4	S240_1_0 S240_1_1 S240_2_0 S240_2_1 S240_3_0 S240_3_1 S240_4_0 S240_4_1
Sortie relais N°1  N°2	RA1  RA0		sre_1  sre_2	SR1_0 SR1_1 SR2_0 SR2_1
Entrées Sorties Led/Switch N°1  N°2  N°3  N°4  N°5  N°6  N°7		I/O1  I/O2  I/O3  I/O4  I/O5  I/O6  I/O7	Io1  io2  io3  io4  io5  io6  io7	IO1 IO1_0 IO1_1 IO2 IO2_0 IO2_1 IO3 IO3_0 IO3_1 IO4 IO4_0 IO4_1 IO5 IO5_0 IO5_1 IO6 IO6_0 IO6_1 IO7 IO7_0 IO7_1
Liaisons série  Interruption	TX RX RB0 int	RxD TxD I/O0 int		

**Le fichier binaire du module SitePlayer Site.spd :**

```

;
; Site
;
$Devicename "Serveur Web Domotique"
$DHCP off
$DownloadPassword ""
$SitePassword ""
$InitialIP "192.168.0.10"
$PostIRQ off
$Sitefile "C:\Documents and Settings\Guy\Mes
documents\Travail\Alarme\Sylvain\Site\Site.spb"
$Sitepath "C:\Documents and Settings\Guy\Mes documents\Travail\Alarme\Sylvain\Site"
;
    org 0FF1Ah
baudrate dw 65015                ; Vitesse liaison série 9600 baud
;
    org 0ff00h
port1 ds 1                        ; Port 1 all 8 bits
;

```







```

#define S240_3_0  output_low (pin_A3)
#define S240_3_1  output_high(pin_A3)

#define S240_4    input (pin_B0)
#define S240_4_0  output_low (pin_B0)
#define S240_4_1  output_high(pin_B0)

#define SR1       input (pin_A1)           // sorties relais
#define SR1_0     output_low (pin_A1)
#define SR1_1     output_high(pin_A1)

#define SR2       input (pin_A0)
#define SR2_0     output_low (pin_A0)
#define SR2_1     output_high(pin_A0)

// commande de SitePlayer
#define NOP        0
#define Status    0x10
#define Reset     0x20
#define ComParams 0x33
#define UDPsend   0x50
#define Read      0xC0
#define Write     0x80
#define ReadX     0xD0
#define WriteX    0x90
#define ReadBit   0xE0
#define WriteBit  0xA0
#define ToggleBit 0xB0

int8 IO; // sauve entrées/sorties SitePlayer

void Put_Site(int8 data)
// putc() avec attente
{
    putc(data);
    delay_ms(1);
}

int8 Get_Site(void)
//Getc regroupées pour simulation
{
    return(getc());
}

void SitePlayer(void)
// dialogue Entrées Sorties avec SitePlayer.
{
    int8 i, GBit;

    for(i=0;i<20;i++) Put_Site(NOP); // 20 NOP, synchro SitePlayer

    Put_Site(WriteBit); Put_Site(0); Put_Site(E240_1); // Entrées
    Put_Site(WriteBit); Put_Site(1); Put_Site(E240_2);
    Put_Site(WriteBit); Put_Site(2); Put_Site(E20_1);
    Put_Site(WriteBit); Put_Site(3); Put_Site(E20_2);
    Put_Site(WriteBit); Put_Site(4); Put_Site(E20_3);

    Put_Site(ReadBit); Put_Site( 5); GBit = Get_Site(); // Sorties
    if((GBit&1)!=0) S240_1_0; else S240_1_1;
    Put_Site(ReadBit); Put_Site( 6); GBit = Get_Site();
    if((GBit&1)!=0) S240_2_0; else S240_2_1;
    Put_Site(ReadBit); Put_Site( 7); GBit = Get_Site();
    if((GBit&1)!=0) S240_3_0; else S240_3_1;
    Put_Site(ReadBit); Put_Site( 8); GBit = Get_Site();
    if((GBit&1)!=0) S240_4_0; else S240_4_1;
    Put_Site(ReadBit); Put_Site( 9); GBit = Get_Site();
    if((GBit&1)!=0) SR1_0; else SR1_1;
    Put_Site(ReadBit); Put_Site(10); GBit = Get_Site();
    if((GBit&1)!=0) SR2_0; else SR2_1;

    // IO de SitePlayer
    Put_Site(Readx); Put_Site(0x00); Put_Site(0xFF); IO=Get_Site();
}

void Sauve(void)
/* Sauve les sorties en EEPROM pour garder les états logiques en cas de panne
d'alimentation. */
{
    int8 i, Data=0, GBit;

```

```

for(i=0;i<6;i++)
{
    Put_Site(ReadBit); Put_Site(i+5); GBit = Get_Site();
    if((GBit&1)!=0) Bit_set (Data, i);
    else      Bit_clear(Data, i);
}
write_eeprom (0,Data);
write_eeprom (1,IO);
}

void Recupere(void)
/* Récupère les sorties sauvées dans l'EEPROM et les place sur les sorties du
PIC au moment de la remise sous tension. */
{
    int8 i, Data=0, GBit;

    // sorties dans SitePlayer
    Data = read_eeprom(0);
    for(i=0;i<6;i++)
    {
        GBit = Bit_test (Data, i);
        Put_Site(WriteBit); Put_Site(i+5); Put_Site(GBit);
    }
    // IO dans SitePlayer
    IO = read_eeprom(1);
    Put_Site(Writex); Put_Site(0x00); Put_Site(0xFF); Put_Site(IO);
}

void main(void)
{
    char Version[]={"Site DAVID 2006 Version 1.0"};
    int i;

    setup_timer_0(RTCC_INTERNAL);setup_wdt(WDT_2304MS);
    setup_timer_1(T1_DISABLED);
    setup_timer_2(T2_DISABLED,0,1);
    setup_comparator(NC_NC_NC_NC);
    setup_vref(FALSE);

    port_b_pullups(TRUE);
    // Grave version en EEPROM
    for(i=0;i<strlen(Version);i++) write_eeprom (5+i,Version[i]);
    delay_ms(500);
    restart_wdt();

    for(i=0;i<20;i++) Put_Site(NOP); // synchro SitePlayer
    Put_Site(Reset); // Reset SitePlayer
    delay_ms(500);
    for(i=0;i<20;i++) Put_Site(NOP); // synchro SitePlayer
    Put_Site(Writex+3); Put_Site(0xE6); Put_Site(0xFF); // IP=192.168.0.10
    Put_Site(192); Put_Site(168); Put_Site(0); Put_Site(10);
    Recupere();

    while(1)
    {
        SitePlayer();
        Sauve();
        restart_wdt();
    }
}

```

### **Bibliographie :**

<http://fr.selfhtml.org>

<http://laluiste.com/document.php?url=http://laluiste.com/courshtml/sommaire.html>

[http://www.htmlcodetutorial.com/forms/\\_INPUT.ht](http://www.htmlcodetutorial.com/forms/_INPUT.ht)

<http://fr.selfhtml.org/navigation/html.htm>

## RAB de traduction :

Programme de test du Port Série :

Le programme de test du port série, **SitePlayerSerialDemo.exe**, est inclus dans les logiciels de SitePlayer. Il offre une communication PC avec SitePlayer afin de développer et de tester les applications de toutes sortes. Il fournit aussi un moyen d'obtenir ou de mettre l'adresse IP de SitePlayer toutes les fois que c'est pas possible par le réseau Ethernet. Si nécessaire, changez le protocole du port COM de votre PC pour vous adapter à SitePlayer : 9600 baud; 8 bits de données; Aucune parité; 1 Stop bit.

1. Connecter une extrémité du câble série sur la carte SitePlayer (Sur la carte de développement c'est le connecteur J9, si c'est une autre application il faut interfacier avec un circuit MAX233, ou un MAX232 + 4 condensateurs).
2. Connecter l'autre extrémité du câble série sur le ports série de votre ordinateur.
3. Démarrer le programme de test SitePlayerSerialDemo.
4. Choisir le port COM du PC dans le menu Comm.
5. Choisir les fonctions telles que Get IP Address ou Set IP Address.
6. Entrer l'adresse mémoire hexadécimale et le type de donnée obtenir et donner des valeurs.
7. Envoyer le message UDP.

