**Connexion au compteur Linky,  
sortie de télé-information - liaison série**

[I. Document ERDF-NOI-CPT\_44E 2](#_Toc386442249)

[I.1. Caractéristiques physiques des signaux de télé-information client 2](#_Toc386442250)

[I.2. Document Compteur\_energie\_zcd126\_02 2](#_Toc386442251)

[II. Schéma de démodulation 3](#_Toc386442252)

[III. Composition d’une trame 3](#_Toc386442253)

[III.1. Cheksum 3](#_Toc386442254)

[III.2. Etiquette d’une trame 4](#_Toc386442255)

[IV. Connexion à la Raspery Pi 5](#_Toc386442256)

[V. Algorithme 5](#_Toc386442257)

[VI. Installations sur R-Pi 6](#_Toc386442258)

[VI.1. Utilisation de Microcom comme terminal rs232 sur R-Pi 6](#_Toc386442259)

[VI.2. Utilisation du compilateur Gcc 6](#_Toc386442260)

[VI.3. Utilisation de Bcm2835 comme bibliothèque hardware 6](#_Toc386442261)

[VI.4. Configuration de l’UART 6](#_Toc386442262)

[VI.4.1. Retirer l’utilisation du périphérique au noyau pour pouvoir s’en servir : 6](#_Toc386442263)

[VI.4.2. Utiliser le terminal minicom 7](#_Toc386442264)

[VI.4.3. Programme (en PY ?) pour interagir avec l’UART 7](#_Toc386442265)

[VI.4.4. Différents codes pour accéder aux GPIO 7](#_Toc386442266)

[VI.4.4.A. Advanced Serial Port Control 7](#_Toc386442267)

[VII. Installation d’une bibliothèque pour la liaison série 7](#_Toc386442268)

[VIII. Connection à la R-Pi via VNC 7](#_Toc386442269)

[IX. Dongle Wifi TP-Link TL-WN725N 8](#_Toc386442270)

[IX.1. Installer le driver sous R-Pi 8](#_Toc386442271)

[IX.2. Configurer le dongle en point d’accès Wifi 8](#_Toc386442272)

1. Document ERDF-NOI-CPT\_44E

# Caractéristiques physiques des signaux de télé-information client

Soulever le cache vert pour accéder aux borniers.

**2.3.1 Télé-information filaire sous couvre-bornes client**

La transmission pour l’interface sous couvre-bornes a les caractéristiques suivantes :

* binaire,
* unidirectionnelle,
* vitesse de modulation 1200 bauds (mode historique) et 9600 bauds (mode standard) +/-1% ,
* durée égale des bits à « 0 » et à « 1 » ,
* fréquence de la **porteuse 50kHz** +/-3% ,
* **logique de codage négative**, c’est-à-dire que si la porteuse est présente alors le bit vaut « 0 », sinon le bit vaut « 1 ».

# Document Compteur\_energie\_zcd126\_02

6.2 SORTIE TELEINFORMATION CLIENT

**6.2.2 Caractéristiques de la liaison**

Le principe de la communication est une modulation d'amplitude en tout ou rien d’une **porteuse à 50 kHz**. La **logique est négative.**

- un bit émis à "0" correspond à la présence de porteuse pendant le temps correspondant.

- un bit émis à "1" correspond à l'absence de porteuse pendant le temps correspondant.

Après démodulation, on retrouve une liaison asynchrone classique dont les caractéristiques sont les suivantes:

- vitesse de transmission 1200 bits/s

- codage de toutes les informations sous forme ASCII (affichable)

-> 7 bits pour représenter un caractère ASCII

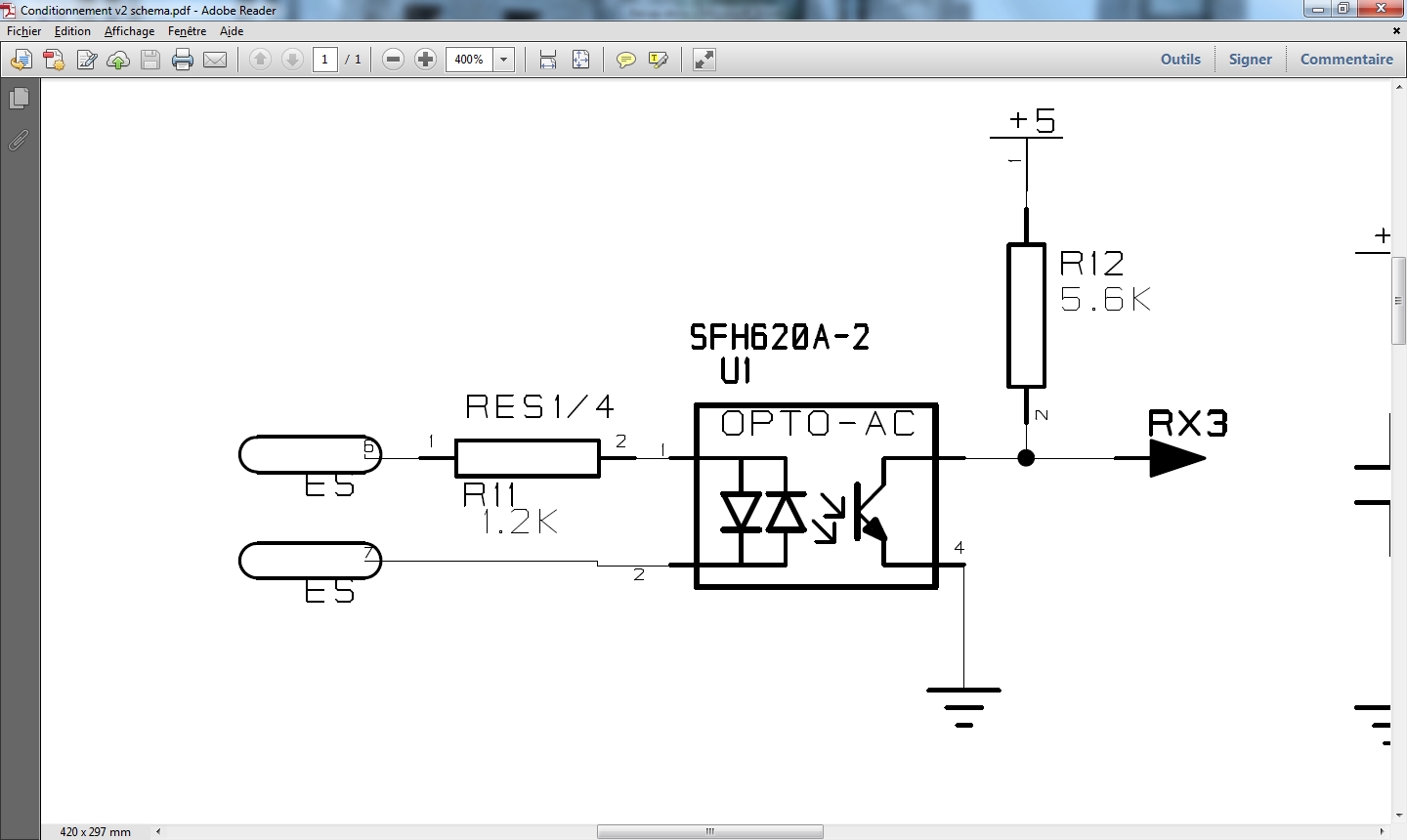
-> 1 bit de parité, parité paire (even)

- un bit de start avant chaque caractère <=> "0" logique

- un bit de stop après chaque caractère <=> "1" logique

1. Schéma de démodulation

**P3V3**



**P3V3**

**GND**

**4.7µF**

<http://fr.farnell.com/vishay/sfh620a-2/optocoupleur-entree-ac/dp/1045384>

L’opto supporte 60mA en entrée et 50mA en sortie. Tenue en tension de 5kV.

Pas d’information sur le courant de sortie, utilisation d’une résistance 1.2kΩ et vérifier de la chute de tension.

Pour une sortie sous 3,3V ; R12 = 3,3V / 5mA = 660Ω ; normalisé à **680Ω**.

1. Composition d’une trame

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Format d'un groupe contenant une donnée non horodatée** | | | | | | |
| <LF> (0x0A) | Etiquette | <HT> (0x09) | Donnée | <HT> (0x09) | Checksum | <CR> (0x0D) |
|  | Zone contrôlée par la checksum | | | |  |  |
|  |  |  |  |  |  |  |
| HT = Horizontal tab <=> espace | | |  |  |  |  |

# Cheksum

Le calcul de la *checksum* met en œuvre le même algorithme de calcul décrit dans le document de référence [1] ; mais l’ensemble des caractères pris en compte est différent. Elle est calculée sur l'ensemble des caractères allant du début du champ *Etiquette* à la fin du champ *Donnée*, caractères <HT> inclus.

1. On fait tout d'abord la somme des codes ASCII de tous ces caractères.
2. Pour éviter d'introduire des fonctions ASCII (0x00 à 0x1F), on ne conserve que les six bits de poids faible du résultat obtenu (cette opération se traduit par un ET logique entre la somme précédemment calculée et 0x3F).
3. Enfin, on ajoute 0x20.
4. Le résultat sera donc toujours un caractère ASCII imprimable (signe, chiffre, lettre majuscule) allant de 0x20 à 0x5F.

# Etiquette d’une trame

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Désignation** | **Etiquette** | **Nombre de caractères** | **Unité** | **Compteur Linky d’expérimentation monophasé** | |
| **Contrat historique** | **Contrat non historique** |
| Adresse du compteur | ADCO | 12 |  | ADS | |
| Option tarifaire choisie | OPTARIF | 4 |  | Selon contrat | "BASE" |
| Intensité souscrite | ISOUSC | 2 | A | PREF (en VA) / 200 V | |
| Index option Base | BASE | 9 | Wh | Index fournisseur 1 | Index Totalisateur |
| Index option Heures Creuses | | | | | |
| Heures Creuses | HCHC | 9 | Wh | Index fournisseur 1 | NON TRANSMIS |
| Heures Pleines | HCHP | 9 | Wh | Index fournisseur 2 |
| Index option EJP | | | | | |
| Heures Normales | EJPHN | 9 | Wh | Index fournisseur 1 | NON TRANSMIS |
| Heures de Pointe Mobile | EJPHPM | 9 | Wh | Index fournisseur 2 |
| Index option Tempo | | | | | |
| Heures Creuses Jours Bleus | BBRHCJB | 9 | Wh | Index fournisseur 1 | NON TRANSMIS |
| Heures Pleines Jours Bleus | BBRHPJB | 9 | Wh | Index fournisseur 2 |
| Heures Creuses Jours Blancs | BBRHCJW | 9 | Wh | Index fournisseur 3 |
| Heures Pleines Jours Blancs | BBRHPJW | 9 | Wh | Index fournisseur 4 |
| Heures Creuses Jours Rouges | BBRHCJR | 9 | Wh | Index fournisseur 5 |
| Heures Pleines Jours Rouges | BBRHPJR | 9 | Wh | Index fournisseur 6 |
| Préavis Début EJP (30 min) | PEJP | 2 | min | 30, en préavis EJP | 30, en préavis de pointe mobile |
| Période Tarifaire en cours | PTEC | 4 |  | Selon contrat et tarif | TH.. |
| Couleur du lendemain | DEMAIN | 4 |  | Selon annonce, en Tempo | NON TRANSMIS |
| Intensité Instantanée | IINST | 3 | A | Courant efficace (en A) | |
| Avertissement de Dépassement De Puissance Souscrite | ADPS | 3 | A | Courant efficace, si Ilnst > IR | |
| Intensité maximale appelée\* | IMAX | 3 | A | Imax (en A) | |
| Puissance apparente | PAPP | 5 | VA | S (en VA), arrondi à la dizaine la plus proche | |
| Horaire Heures Pleines Heures Creuses | HHPHC | 1 |  | "A" | |
| Mot d'état du compteur | MOTDETAT | 6 |  | "000000" | |
|  |  |  |  |  |  |
| **\*L'intensité maximale est remise à 0 chaque jour au moment de l'arrêté d'index journalier** | | | | |  |

1. Connexion à la Raspery Pi

Utilisation des pins 3V3, GND et GPIO15 (RXD).



1. Algorithme

Réception d’un caractère

Attente du STX

Sauvegarde de toute la trame dans une variable

1. Installations sur R-Pi

# Utilisation de Microcom comme terminal rs232 sur R-Pi

<http://packages.debian.org/fr/wheezy/s390/microcom>

## Utiliser le terminal minicom

You can use minicom using the next line command

**minicom** -b 1200 -o -D /dev/**ttyAMA0**

Faire CTRL + A puis z pour afficher les menus, y compris pour quitter

# Utilisation du compilateur Gcc

GNU GCC est déjà installé

<http://www710.univ-lyon1.fr/~jciehl/Public/educ/gcc.html> Bon tuto

Utilisation de MinGW pour compiler sous Windows

# Utilisation de Bcm2835 comme bibliothèque hardware

C library for Broadcom BCM 2835 as used in Raspberry Pi

<http://www.airspayce.com/mikem/bcm2835/index.html>

L’installer sur la R-PI en suivant les informations contenues dans le fichier INSTALL

# Configuration de l’UART

## Retirer l’utilisation du périphérique au noyau pour pouvoir s’en servir :

<https://sites.google.com/site/semilleroadt/raspberry-pi-tutorials/gpio>

In order to use the dedicated UART pins on the raspberry pi, first they have to be removed from their default application which is debugging.

To do this edit "/boot/cmdline.txt" and "/etc/inittab".

You can backup this files if you want to return to the default configuration:

cp /boot/cmdline.txt /boot/cmdline.bak

cp /etc/inittab /etc/inittab.bak

Remove "console=ttyAMA0,115200" and "kgdboc=ttyAMA0,115200" configuration parameters from the "/boot/cmdline.txt" configuration file using nano editor.

nano /boot/cmdline.txt

Comment the last line on the "/etc/inittab" file. Put a '#' before "T0:23:respawn:/sbin/getty -L ttyAMA0 115200 vt100.

nano /etc/inittab

Now the RXD (GPIO15) and TXD (GPIO14) pins are available for general UART use.

## Programme (en PY ?) pour interagir avec l’UART

<http://www.blaess.fr/christophe/2012/11/26/les-gpio-du-raspberry-pi/>

## Différents codes pour accéder aux GPIO

<http://elinux.org/RPi_Low-level_peripherals>

### Advanced Serial Port Control

<https://projects.drogon.net/raspberry-pi/wiringpi/serial-library/>

The ***wiringSerial*** library is intended to provide simplified control – suitable for most applications, however if you need advanced control – e.g. parity control, modem control lines (via a USB adapter, there are none on the Pi’s on-board UART!) and so on, then you need to do some of this the “old fashioned” way.

For example – To set the serial line into **7 bit mode plus even parity**, you need to do this…

In your program:

#include <termios.h>

and in a function:

  struct termios options ;

  tcgetattr (fd, &options) ; // Read current options

cfsetispeed (&options, B1200); // 1200 Bauds

  options.c\_cflag &= ~CSIZE ; // Mask out size

  options.c\_cflag |= CS7 ; // Or in 7-bits

options.c\_cflag |= PARENB ; // Enable Parity - even by default

tcsetattr (fd, TCSANOW | TCSAFLUSH, &options) ; // Set new options

// ATTENTION, verifier ces option TCSANOW | TCSAFLUSH car fonctionnement incorrect

 The ‘fd’ variable above is the file descriptor that serialOpen() returns.

Please see the man page for tcgetattr for all the options that you can set.

man tcgetattr

1. Installation d’une bibliothèque pour la liaison série

<http://wiringpi.com/download-and-install/>

1. Connection à la R-Pi via VNC

Sous VNC viewer, saisir **192.168.1.11:1** puis mdp

Tuto : <http://the-raspberry.com/vnc>

1. Dongle Wifi TP-Link TL-WN725N

# Installer le driver sous R-Pi

<http://www.raspberrypi.org/forums/viewtopic.php?p=462982#p462982>

Remplacer le XX par la distribution obtenue, équivalence depuis la page ci-dessus.

uname -a

wget https://dl.dropboxusercontent.com/u/80256631/8188eu-201**XX**.tar.gz

tar -zxvf 8188eu-201**XX**.tar.gz

sudo install -p -m 644 8188eu.ko /lib/modules/$(uname -r)/kernel/drivers/net/wireless

sudo insmod /lib/modules/$(uname -r)/kernel/drivers/net/wireless/8188eu.ko

sudo depmod –a

sudo modprobe 8188eu

Vérifier que le pilote a bien été installé, le pilote 8188eu doit apparaître.

sudo lsmod

# Configurer le dongle en point d’accès Wifi

Getting WN725N V2 working as an access point

Comment faire:

<http://raspberry-at-home.com/hotspot-wifi-access-point>

En cas de problème :

<http://www.raspberrypi.org/forums/viewtopic.php?f=26&t=49355>