

LE PROJET

Pourquoi?

J'ai dit oui au linky après qu'ils m'aient assuré que ce serait transparent pour l'EJP et pour la prod photovoltaïque..

Pour la prod, c'est OK.

Pour l'EJP, il n'y a plus d'asservissement, ni de préavis disponible. Les relais asservis sur un signal analogique spécifique ne fonctionnent plus. Plus de loupiotte orange en cas d'EJP.

Pas content, il fallait faire quelque chose...

Comment?

Le linky offre une interface série spécifique qui se monitore assez facilement via une raspberry.

C'est très bien documenté là: <https://blog.bigm.fr/suivre-sa-consommation-electrique/>

Le linky offre aussi un relais connecté en cas d'EJP.

Il faut donc:

- Tirer un nouveau câble entre le compteur et la maison (au moins 2 brins)
- Monter une Rpi de récup pour monitorer tout ça
- Monter un circuit simple pour alimenter la LED de la cuisine qui nous disait quand on était EJP (prix électricité exorbitant !)

Câbler le Linky

On a 2 bornes à raccorder: I1/I2 (liaison série côté Linky)

On dédie la paire blanc/marron à la liaison série: **I1=blanc, I2=marron**

Intégrer la RPi

Emplacement

On va l'installer sur la hotte de la cuisine, à l'endroit occupé par notre défunte LED orange de EJP.

On a donc filé un câble 4 paires qui part du tableau électrique et arrive en dessus de hotte de cuisine

Alimentation RPi

La prise électrique du tableau électrique a été recablée. Elle est maintenant branchée en continu au 220.

On va y installer l'alimentation de la Rpi dont on a coupé le câble qu'on a relié à notre 4 paires vers la cuisine:

- Brins rouge+jaune pour le +
- Brins bleu+gris pour le -

Liaison série Linky / Rpi (I1/I2 <-> blanc-marron)

La paire marron/blanc est prolongé vers la cuisine.

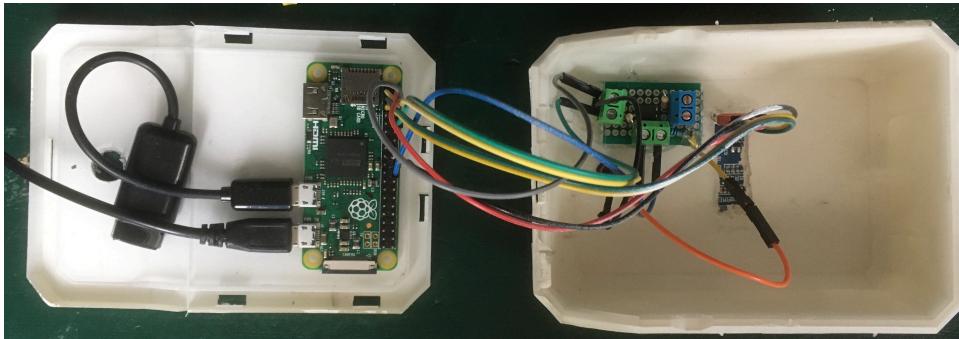
Intégration dans un boitier de récup

Une vieille boite de clous plastique repeinte comme neuve

On y place avec un peu de colle:

- La Rpi-zéro et son dongle WiFi
- La carte d'acquisition
- La led de warning EJP
- L'écran OLED (non visible sur la photo)

Et voilà le travail !!



LE HARDWARE d'ACQUISITION

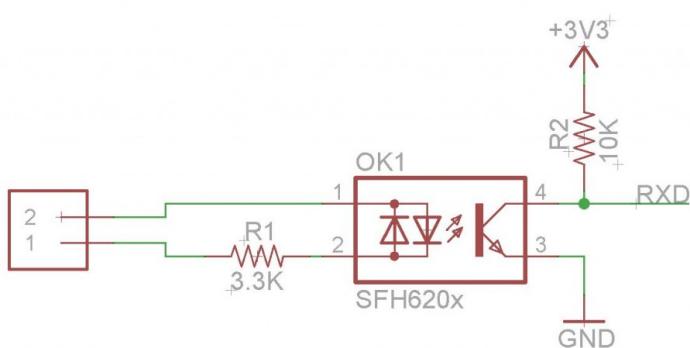
La carte d'acquisition du linky

Beaucoup de schémas sur le net.

Charles Hallard, qui semble avoir une super expertise vend une carte sur [tindy](#).

Sur la page suivante (<http://hallard.me/demystifier-la-teleinfo/>), il explique l'historique de la carte et toutes les étapes par lesquelles il est passé.

De cet article, on retient le schéma suivant avec **R1=1.2K, R2=10K**: (valeur figure fausse)



LE CABLAGE ACQUISITION <-> RPI

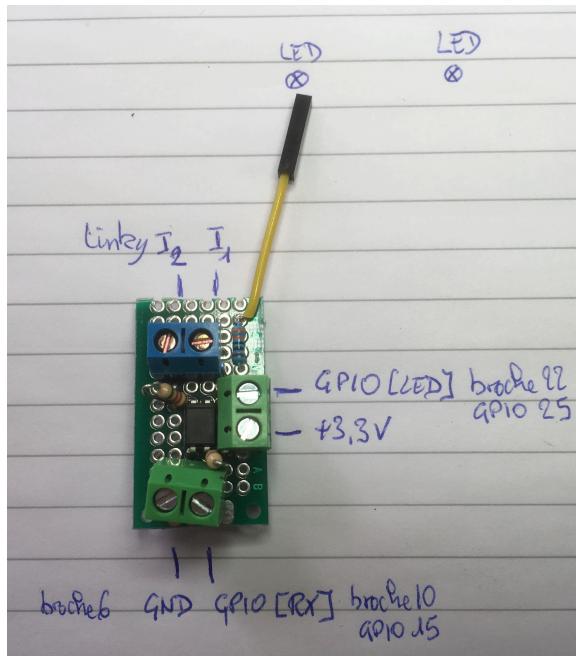
Les entrées / sorties de la carte

L'optocoupleur est alimenté / relié à 3 broches du GPIO:

- La masse (en bas à gauche), relié par exemple à la broche n°6
- Le RX du port série (en bas à droite), relié à la broche n° 10 commandée par le GPIO-15
- L'alim 3.3V (à droite en bas), relié à la broche n° 1
- La commande de LED (optionnel, à droite en haut) relié à la broche n° 22 commandée par le GPIO-25

Le linky est connecté au bornier bleu

- I1 en haut à droite
- I2 en haut à gauche

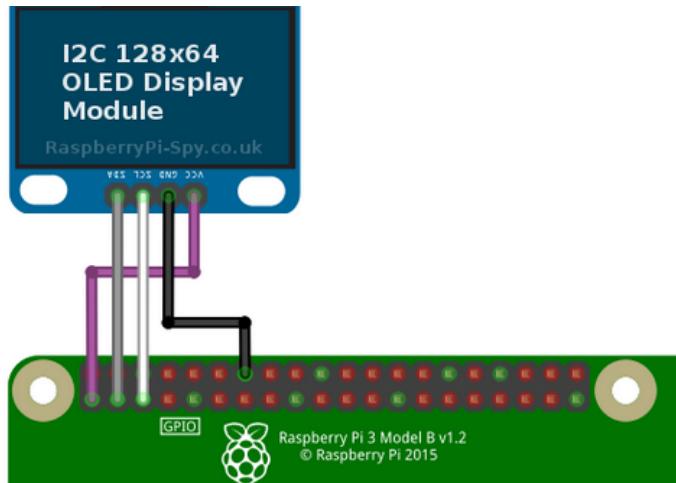


L'écran OLED

On prend la configuration I2C qui va utiliser les broches GPIO 3 et 5 pour la transmission des data.

L'alim sera en 3.3V sur la broche 1

BROCHE OLED	BROCHE GPIO PI	REMARQUES
Vcc	1 *	3.3V
Terre	14 **	Terre
SCL	5	I2C SCL
ADD	3	I2C SCA



PREPARATION OS RASPBERRY

Installation OS

On part sur une carte utilisée précédemment (debian buster nov-2021)
User sudo par défaut: pi / raspberry.

Configuration du système:

** L'image Linux:

Ca boote assez vite (30s quand même...)
On archivera tout ça sur le Mac en fin de parcours
==> L'archive est dispo sur Documents/Perso/Etudes Energie/LinkyMonitor/IMGs

** L'extension de partition:

Si la carte a une capacité supérieure à l'image Linux, on l'étend avec raspi-config

** Clavier français:

Raspi-config est ton ami.

** Nom de la machine

Raspi-config : "System Options"

Ou

Dans le fichier **/etc/hostname**, mettre le nom de la machine:

LinkyMonitor

** Réseau et IP statique:

Édition **/etc/wpa_supplicant/wpa_supplicant.conf** file (ne pas oublier la ligne id_str qui sera référencée dans

/etc/network/interfaces)

```
ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev
update_config=1
country=FR
```

```
network={
    ssid="MainjeotteBouygues"
```

```

psk="La_Mainjeotte"
id_str="MainjeotteBouygues"
}

Edition /etc/network/interfaces
# interfaces(5) file used by ifup(8) and ifdown(8)
# Include files from /etc/network/interfaces.d:
source /etc/network/interfaces.d/*

auto lo
iface lo inet loopback
iface eth0 inet manual

auto wlan0
allow-hotplug wlan0
iface wlan0 inet manual
wpa-roam /etc/wpa_supplicant/wpa_supplicant.conf
iface MainjeotteBouygues inet static
address 192.168.1.6
netmask 255.255.255.0
gateway 192.168.1.1

```

**** Accès SSH**

Sous **raspi-config**, activer les accès distants via SSH

Pour se connecter sans mot de passe:

Dans le fichier **/etc/ssh/sshd_config**, vérifier les lignes suivantes:

```

PubkeyAuthentication yes
AuthorizedKeysFile .ssh/authorized_keys

```

Recopier le fichier **~/.ssh/id_rsa.pub** du client (ici, le mac) vers le fichier **.ssh/authorized_keys** du user 'pi'

Installation du module perl d'affichage OLED: HiPi:

**** Dépendances:**

```

sudo apt-get update
sudo apt-get install gdebi-core
wget http://hipi.znix.com/download/libhipi-perl\_0.86-1\_armhf.deb
sudo gdebi libhipi-perl_0.86-1_armhf.deb

```

L'installation de lighttpd

NB: si l'OS de la carte SD est déjà vieux. Il faut forcer apt-get update à accepter les vieux trucs:

```
sudo apt-get update --allow-releaseinfo-change
```

Puis installer lighttpd et php:

```

sudo apt-get install lighttpd
sudo apt install php7.3 php7.3-fpm php7.3-cgi
sudo lighttpd-enable-mod fastcgi-php

```

Par défaut, le répertoire des pages HTML est :

```
/var/www/html
```

La configuration du port série

Après quelques essais sur board, ça marche moyen: plein de caractères foireux sur le port série.
La solution vient encore du site de Mr Hallard:

- Il faut d'abord **dévalider les connections ssh** à travers les prts série (raspi-config)
set **No** to "Would you like a login shell to be accessible over serial?"

- Il faut ensuite **activer le protocole UART dans le noyau**: fichier /boot/config.txt
`enable_uart=1`

Et là, c'est parfaitement fluide après **avoir configuré le port série** comme il faut:

```
stty -F /dev/ttyAMA0 1200 sane evenp parenb cs7 -crtscs
```

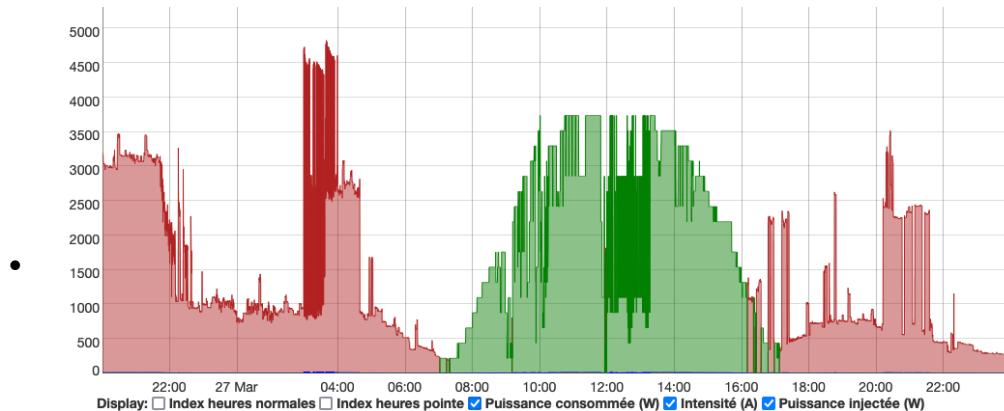
DEVELOPPEMENT LOGICIEL

L'architecture

Même architecture que GyroPlastic et Solarmonitor:

- Une partie acquisition indépendante
 - Une partie web pour servir des pages par période
- Ici: un jour de data:

Linky Monitor: 2022_03_27



Période:	Heures Normales	
Conso EJP sur période:	0 kWh	0,00 €
Conso HN sur période:	25.296 kWh	3,25 €
Conso totale sur période:	25.296 kWh	3,25 €
Injecté sur période:	20.479 kWh	3,28 €

TODO: Voir si on peut faire:

- Des graphes dynamiques updatés via lecture des logs (tail -f ?)
- Des concaténations de logs pour visualiser des périodes plus longues que 1j

Le codage de l'acquisition : LinkyAcq.pl

Le script LinkyAcq.pl est le **process chargé des acquisitions** :

- Aquierte les lignes linky sur /dev/ttyAMA0
 Acquisition avec timeout en cas de pb
 Acquisition de chaque ligne avec vérification du checksum
- Lit les variables
HCHC: index heures creuses
HCHP: index heures pointe

PTEC: type de période ("HN.." pour les heures normales)

IINST: intensité vue par le compteur

PAPP: puissance consommé

- Rafraîchit l'écran OLED avec la valeur EJP (OUI|NON) et les valeurs PAPP et PINJ (calculé: (PAPP=0)?IINST*0.22:0)
- Attends une série de N lignes PAPP (par défaut 8 occurrences configurable dans fichier de conf)
- Les transforme en une ligne de logs au format habituel
2021/02/20 19:10:22 DATA Periode=HN.., IndexHN=000688322, IndexHPM=000080072, PuissConsommee=00000, IConsommee=013

Le codage de la partie graphique : LinkyGraphCreate.pl

Le script LinkyGraphCreate.pl est le **process chargé des visualisations HTML:**

- Va lire les lignes DATA d'un fichier de log journalier
- Calcule la puissance injectée en se basant sur IINST et sur une PAPP=0
- Calcule les totaux consommés et les coûts associés

Les interactions sont prises en charge côté client par le module javascript DyGraph

En tout 680 lignes de perl/php/JavaScript bien commentées (475 lignes instructions).

Ca va...

EXPLOITATION

L'installation

Le répertoire choisi est **/var/www/html/LinkyMonitor/**

L'installation consiste à

- Créer l'arbo:
 - **/var/www/html/LinkyMonitor/bin**
 - **/var/www/html/LinkyMonitor/conf**
 - **/var/www/html/LinkyMonitor/logs**
- Copier les fichiers LinkyAcq.pl et LinkyGraphCreate.php et LinkyGraphCreate.pl dans bin
- Copier le fichier LinkyAcq.conf dans conf

La configuration

Le script LinkyAcq.pl est configuré par le fichier **conf/LinkyAqc.conf**.

Ce fichier est rechargé dynamiquement à chaque trame (série de N PAPP)

Les valeurs par défaut des différentes variables sont les suivantes:

```
# Port série de lecture linky (Raspberry Pi-2B sur Jessie : dev=/dev/ttyAMA0)
#INPUT_DEV /dev/ttyAMA0
```

```

# Temps d'attente maxi en lecture
#READ_TIMEOUT 5

# Delai entre essais de lecture en cas d'echec
#WAIT_BETWEEN_READ_FAILURE 10
# Delai d'attente entre les lectures Linky frames
#WAIT_BETWEEN_FRAMES 10
# Temps max en execution (3 years)
#MAX_TIME 99999999
# Linky mode (H=historique ou S=standard)
#LINKY_MODE H
# OLED Display type 128x32 contrôlé via I2C
#OLED_DISPLAY_TYPE SSD1306_128_X_32_I2C
# Default OLED address
#OLED_ADDRESS 0x3C
# Default OLED Flip
#OLED_FLIPPED 0
# Default GPIO command for LED initialization
#LED_INIT_CMD "raspi-gpio set 25 op"
# Default GPIO comand for LED switch ON
#LED_ON_CMD "raspi-gpio set 25 dh"
# Default GPIO command for LED switch OFF
#LED_OFF_CMD "raspi-gpio set 25 dl"

```

L'utilisation

LinkyAcq.pl:

Les différentes options sont renseignées dans l'usage:

```

Usage: $0 [-m linky_mode] [-t max_sec]
-h      help / usage
-m      Linky mode 'historique' or 'standard'
-t      max exec time (sec)
-l      Affichage OLED des informations lues + allumage LED en EJP

```

Note Avr 2022: le mode standard n'est pas encore supporté

LinkyGraphCreate.pl:

Les différentes options sont renseignées dans l'usage:

```

Usage: $0 [-h] [-s date] [-e date] [-h]
-s      Start date for plot (format YYYY_MM_DD)
-e      End date for plot (format YYYY_MM_DD)
-a      activate auto_update (every 30sec)
-h      Help: display this message

```

Le lancement au boot

Utilisation du script /etc/rc.local qui est lancé en fin de boot:

```
/var/www/html/LinkyMonitor/bin/LinkyAcp.pl -l &
```

A noter l'option -l qui permet d'activer la gestion de l'écran OLED et de la LED d'alarme EJP
Sans cette option, le script ne fera que la création des logs, ce qui marche également si on a choisi de ne consulter que la partie Web de création des graphes.