

SUIVI DE LA TÉLÉINFORMATION ERDF AVEC ARDUINO

REV 2019-08-11

1 Fonctionnement de la téléinformation EDF

La téléinformation s'obtient sur les compteurs possédant les bornes I1 et I2, appelés compteur bleu (modèle pour les particuliers) ou compteur jaune (modèle pour les professionnels).

Il s'agit d'un port série à 1200 bps, les données voyageant sur un courant porteur (j'ai mesuré 6V environ, mais la documentation erdf/enedis indique que le système doit pouvoir assumer exceptionnellement 220V). On peut obtenir:

- le(s) compteur(s) (en Wh, plus précis que les kWh affichés sur l'écran);
- l'intensité fournie instantanée (en ampères, par phase si triphasé);
- la période tarifaire ou la couleur des jours si les contrats prévoient cela;
- et d'autres informations, décrites dans la documentation.

Pour l'instant, je n'ai pas encore de Linky, donc je ne peux rien dire sur les possibilités avec ce dernier. Néanmoins, il paraît que c'est la même chose.

2 Partie arduino

2.1 Hardware

Composants nécessaires:

- une carte arduino (Uno, mini, ou nano);
- un shield ethernet (j'utilise le ENC28J60);
- un photocoupleur: par exemple 4N25;
- une diode: par exemple 1N4001;
- deux résistances: 1 k Ω (résistance de tirage) et 220 Ω (pour la DEL);
- une DEL.

La DEL permet d'avoir un retour de l'activité: elle clignote quand il y a un transfert d'information. Cette partie du montage est optionnelle.

La diode permet de supprimer une alternance du signal, le photocoupleur permet d'isoler le signal du circuit électronique.

D'après Planète-domotique:

- Le signal sur les bornes I1/I2 est un signal modulé à 50kHz.
- La présence de modulation correspond à un 0 logique, et l'absence à un 1 logique.
- Ces informations sont émises cycliquement sous forme de messages composés d'une étiquette d'identification suivie généralement d'une valeur.
- Il est donc nécessaire de démoduler ce signal. Après quoi on obtient une suite de caractères ASCII émise à 1200 bits/s, 7 bits/caractères, parité paire, 1 bit de stop.



Figure 1: Compteur « bleu »

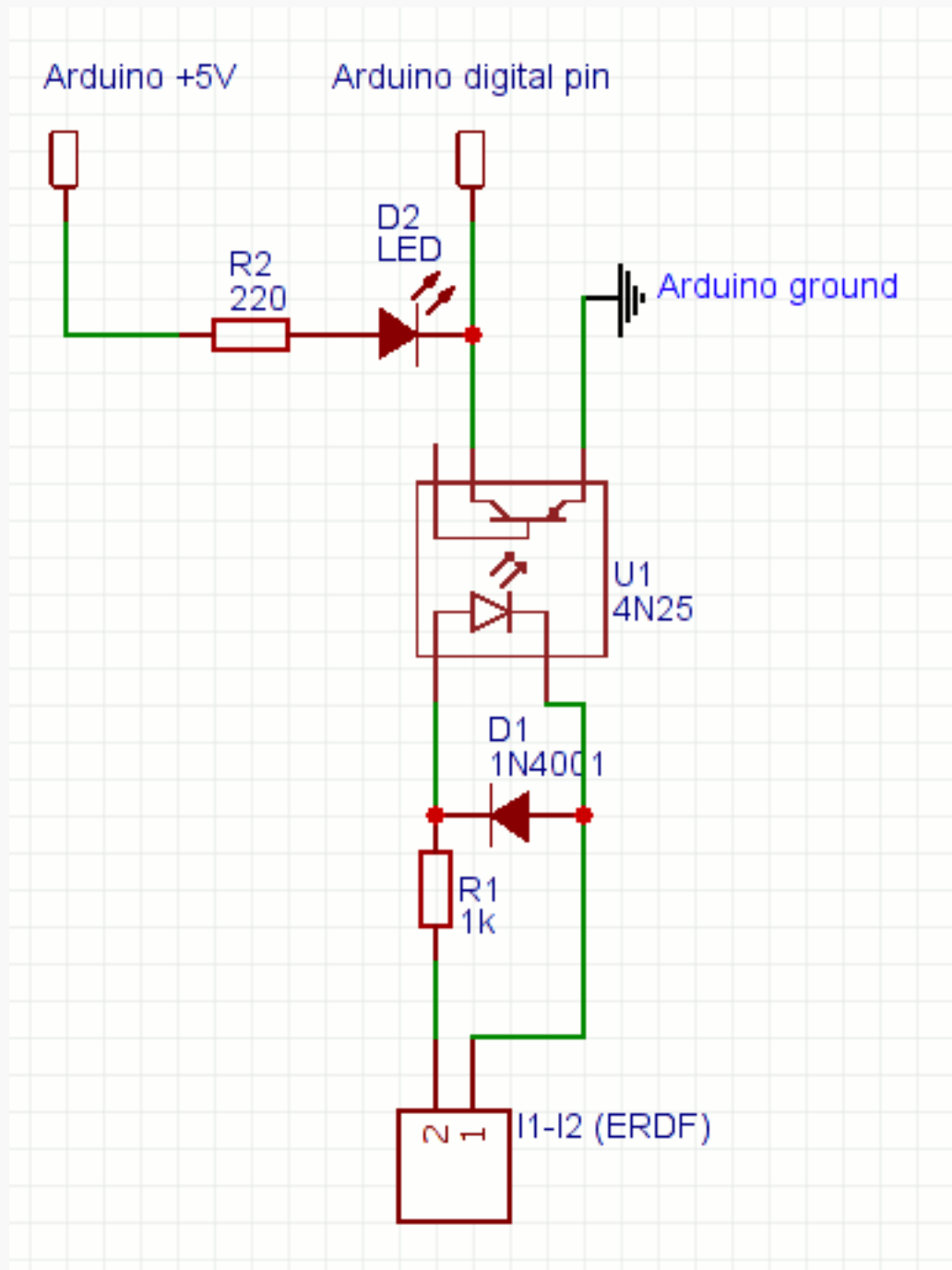


Figure 2: Schéma électronique

2.2 Software

[!info] Il y a un *checksum* pour vérifier la ligne envoyé par le Linky : voir la page [[Checksum des trames de téléinformation enedis]]

```
#include <UIPEthernet.h>
byte mac[] = { 0x00, 0x11, 0x22, 0x33, 0x44, 0x25 };
byte ip[] = { 192, 168, 1, 25 };
EthernetServer server(80);

#include <SoftwareSerial.h>
SoftwareSerial cptSerial(2, 3);
#define startFrame 0x02
#define endFrame 0x03
#define startLine 0x0A
#define endLine 0x0D

void setup()
{
    cptSerial.begin(1200);
    Ethernet.begin(mac, ip);
    server.begin();
}

String GetTeleInfo()
{
    String TeleInfo = "";
    char charIn = 0;
    while (charIn != startLine)
    {
        charIn = cptSerial.read() & 0x7F;
    }
    while (charIn != endLine)
    {
        if (cptSerial.available() > 0)
        {
            charIn = cptSerial.read() & 0x7F;
            TeleInfo += charIn;
        }
    }
    return TeleInfo;
}

String ShowTeleInfo(String keyword, String unit, int length)
{
    int essai = 0;
    // Nombre d'étiquettes maximum, cf documentation ERDF
    int max_essais = 33;
    String data = "";
    String msg = "";
```

```

while(data.substring(0,keyword.length()) != keyword && essai != max_essais)
{
    data = GetTeleInfo();
    essai++;
}
msg = "\t<";
msg += keyword;
msg += " unit=\"";
msg += unit;
msg += "\">";
if (essai != max_essais)
{
    msg += data.substring((keyword.length() + 1),(length + (keyword.length() + 1)));
}
else
{
    msg += "NA";
}
msg += "</";
msg += keyword;
msg += ">";
return msg;
}

void loop()
{
    EthernetClient client = server.available();
    if (client)
    {
        boolean current_line_is_blank = true;
        while (client.connected())
        {
            if (client.available())
            {
                char c = client.read();
                if (c == '\n' && current_line_is_blank)
                {
                    client.println("HTTP/1.1 200 OK");
                    client.println("Content-Type: text/xml");
                    client.println();
                    client.println("<?xml version=\"1.0\"?>");
                    client.println("<teleinformation>");
                    client.println(ShowTeleInfo("ADCO", "", 12));
                    client.println(ShowTeleInfo("OPTARIF", "", 4));
                    client.println(ShowTeleInfo("ISOUSC", "A", 2));
                    client.println(ShowTeleInfo("BASE", "Wh", 9));
                    client.println(ShowTeleInfo("HCHC", "Wh", 9));
                    client.println(ShowTeleInfo("HCHP", "Wh", 9));
                    client.println(ShowTeleInfo("EJPHN", "Wh", 9));
                }
            }
        }
    }
}

```

```

        client.println(ShowTeleInfo("EJPHPM", "Wh", 9));
        client.println(ShowTeleInfo("BBRHCJB", "Wh", 9));
        client.println(ShowTeleInfo("BBRHPJB", "Wh", 9));
        client.println(ShowTeleInfo("BBRHCJW", "Wh", 9));
        client.println(ShowTeleInfo("BBRHPJW", "Wh", 9));
        client.println(ShowTeleInfo("BBRHCJR", "Wh", 9));
        client.println(ShowTeleInfo("BBRHPJR", "Wh", 9));
        client.println(ShowTeleInfo("PEJP", "min", 2));
        client.println(ShowTeleInfo("PTEC", "", 4));
        client.println(ShowTeleInfo("DEMAIN", "", 4));
        client.println(ShowTeleInfo("IINST", "A", 3));
        client.println(ShowTeleInfo("IINST1", "A", 3));
        client.println(ShowTeleInfo("IINST2", "A", 3));
        client.println(ShowTeleInfo("IINST3", "A", 3));
        client.println(ShowTeleInfo("IMAX", "A", 3));
        client.println(ShowTeleInfo("IMAX1", "A", 3));
        client.println(ShowTeleInfo("IMAX2", "A", 3));
        client.println(ShowTeleInfo("IMAX3", "A", 3));
        client.println(ShowTeleInfo("PMAX", "W", 5));
        client.println(ShowTeleInfo("PAPP", "VA", 5));
        client.println(ShowTeleInfo("HHPHC", "", 1));
        client.println(ShowTeleInfo("MOTDETAT", "", 6));
        client.println(ShowTeleInfo("PPOT", "", 2));
        // On ne retiens pas ADIR(1,2,3) ou ADPS (peuvent être calculés)
        client.println("</teleinformation>");
        break;
    }
    if (c == '\n')
    {
        current_line_is_blank = true;
    }
    else if (c != '\r')
    {
        current_line_is_blank = false;
    }
}
}
delay(200);
client.stop();
}
}

```

```

<?xml version="1.0"?>
<teleinformation>
  <ADCO unit="">041430244526</ADCO>
  <OPTARIF unit="">HC..</OPTARIF>

```

```

<ISOUSC unit="A">20</ISOUSC>
<BASE unit="Wh">NA</BASE>
<HCHC unit="Wh">001202472</HCHC>
<HCHP unit="Wh">001965554</HCHP>
<EJPHN unit="Wh">NA</EJPHN>
<EJPHPM unit="Wh">NA</EJPHPM>
<BBRHCJB unit="Wh">NA</BBRHCJB>
<BBRHPJB unit="Wh">NA</BBRHPJB>
<BBRHCJW unit="Wh">NA</BBRHCJW>
<BBRHPJW unit="Wh">NA</BBRHPJW>
<BBRHCJR unit="Wh">NA</BBRHCJR>
<BBRHPJR unit="Wh">NA</BBRHPJR>
<PEJP unit="min">NA</PEJP>
<PTEC unit="">HP..</PTEC>
<DEMAIN unit="">NA</DEMAIN>
<IINST unit="A"> 00</IINST>
<IINST1 unit="A">001</IINST1>
<IINST2 unit="A">004</IINST2>
<IINST3 unit="A">010</IINST3>
<IMAX unit="A"> 01</IMAX>
<IMAX1 unit="A">012</IMAX1>
<IMAX2 unit="A">019</IMAX2>
<IMAX3 unit="A">025</IMAX3>
<PMAX unit="W">08230</PMAX>
<PAPP unit="VA">03340</PAPP>
<HHPHC unit="">C</HHPHC>
<MOTDETAT unit="">000000</MOTDETAT>
<PPOT unit="">00</PPOT>
</teleinformation>

```

Exemple de sortie XML Il est bien sûr tout à fait possible de faire la même chose avec une sortie json.

3 Plugin munin

Munin est un outil de surveillance des ressources, simple à utiliser et à programmer.

```

#!/bin/python3
from lxml import etree
import sys

ARDUINO_URL = '192.168.1.25'

if "config" in sys.argv:
    print("graph_title Compteur EDF")
    print("graph_period hour")
    print("graph_args --base 1000")
    print("graph_vlabel Consommation (Wh)")

```

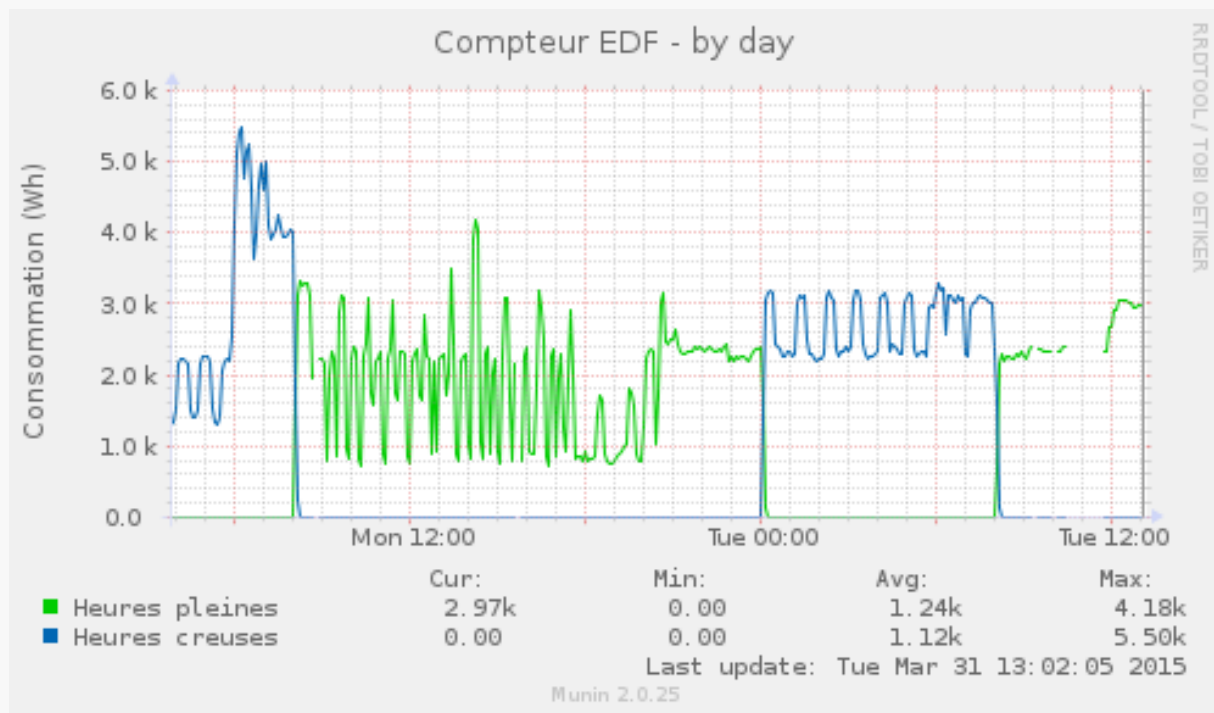


Figure 3: Exemple de graphique produit par Munin

```
print("graph_category Domotique")
print("graph_info Consommation electrique relevee par le compteur")
print("hp.label Heures pleines")
print("hc.label Heures creuses")
print("hp.info Heures pleines")
print("hc.info Heures creuses : 0h-8h")
print("hp.type COUNTER")
print("hc.type COUNTER")
else:
    root = etree.parse('http://{}/'.format(ARDUINO_URL))
    infos = root.xpath('/teleinformation')
    HP = infos[0].find('HCHP').text
    HC = infos[0].find('HCHC').text
    print("hc.value ", HC)
    print("hp.value ", HP)
```

4 Exemples d'applications

- Déclencher certains appareils uniquement lors des heures creuses, les jours « EJP », les jours « bleus »;
- Afficher en direct la consommation, ou la puissance donnée par le compteurs;

- Créer un système de délestage, afin d'éviter une coupure en cas de surconsommation.