

The background features a dark blue gradient with a starry or particle effect. On the left side, there are several circular gauges and progress indicators. One large gauge has a scale from 140 to 260 with major ticks every 10 units and minor ticks every 2 units. Other gauges are smaller and some are partially visible. The overall aesthetic is technical and futuristic.

ECS VERSION 2020

MISE À JOUR DU MONITORING
HARDWARE V4 - SOFTWARE V6

THÈMES TRAITÉS

- Rappel du but du projet
- Nouvel hardware in situ
- Nouveau software embarqué
- Résultats et analyses

Mesures

ÉLÉMENT

- Circuit retour panneau
- Eau solaire (200l)
- Eau sanitaire (200l)
- Marche de la pompe
- Date / Temps
- Logging data
- Echantillonnage

TECHNIQUE

- Sonde t° 1
- Sonde t° 2
- Sonde t° 3
- Relais sur input I/O
- NTP Linux
- SD card, 1 fichier par mois
- 10 minutes (144 val/jour)

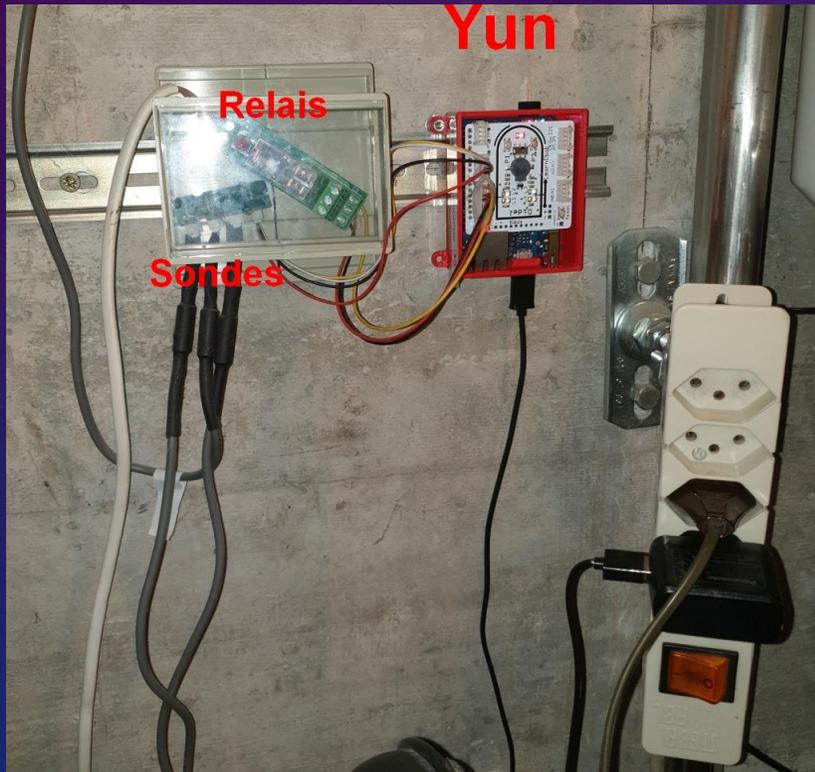
AMÉLIORATIONS 2014 .. 2019

Site WEB: ecs.yvesmasur.ch

- Datas en tableau HTML au lieu de texte
- Datas en tableau par PHP, sélection de la date
- Calcul de l'énergie de la pompe, en Python
- Datas en graphique par JS (BootStrap)
- Calcul de la constante de temps en Python
- 2014-08-09 YM Version with onewire + data logger
- 2014-08-31 YM: log file named with Month and year; add pump state input
- 2014-09-26 YM: compute proportion of unused time by pump
- 2016-05-05 YM: memorize last temp 1 with pump activity
- 2019-03-10 YM: add yield() to main loop

HARD

Avant



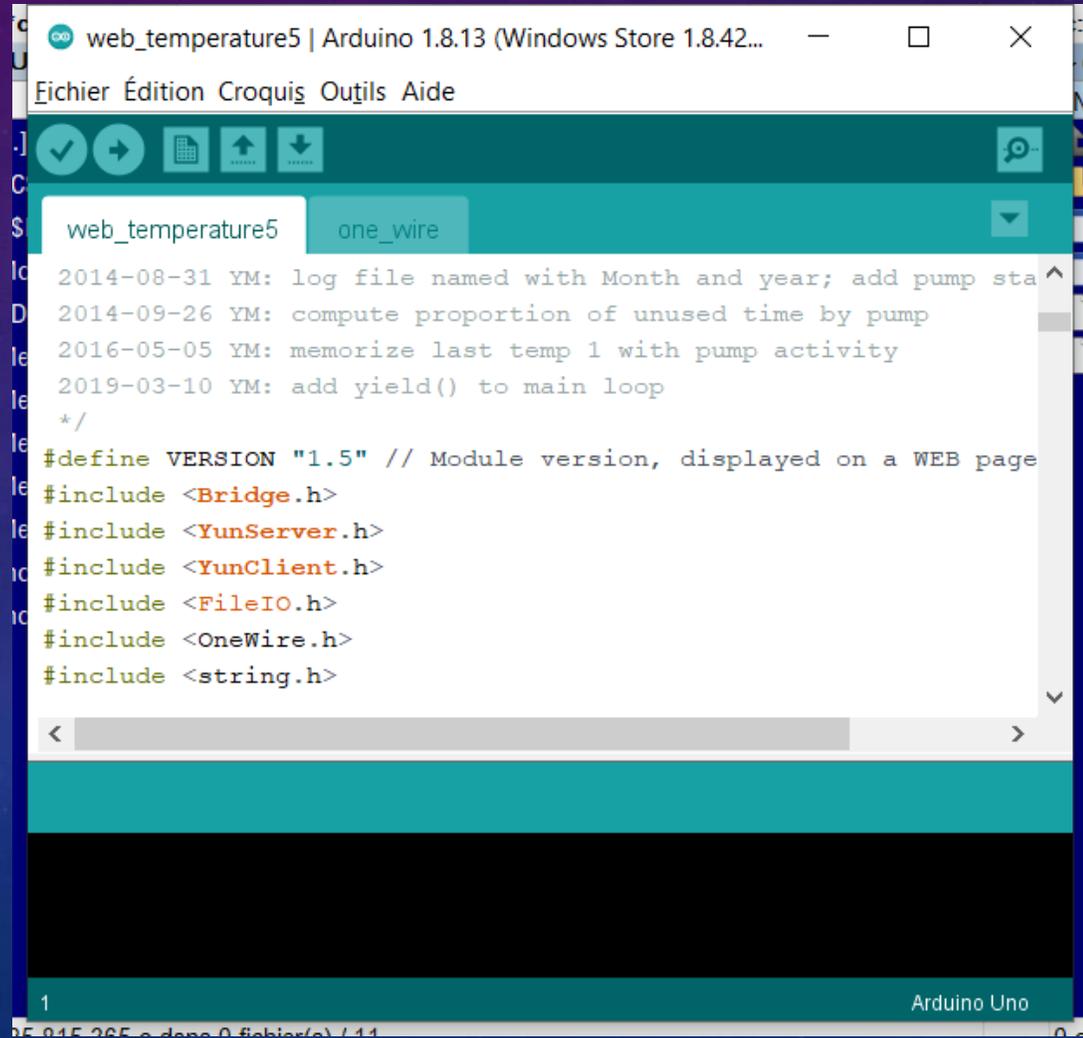
Un peu léger...

- Relais 230VAC / 5V= dans boîtier
- 3 sondes 1-W en parallèle, avec bus Grove
- Yun: boîtier ABS accroché au rail DIN
- Alim bloc secteur, patte USB-C longue

SOFTWARE

Avant

- Deux fichiers .ino
- Monitorer par serveur WEB local
- Temps, selon NTP Linux
- Lecture sondes tts les 10 minutes
- Sondes onewire en parallèle
- Etat pompe via relais électro-mécanique
- Stockage données sur carte SD
- Pousser les données par FTP



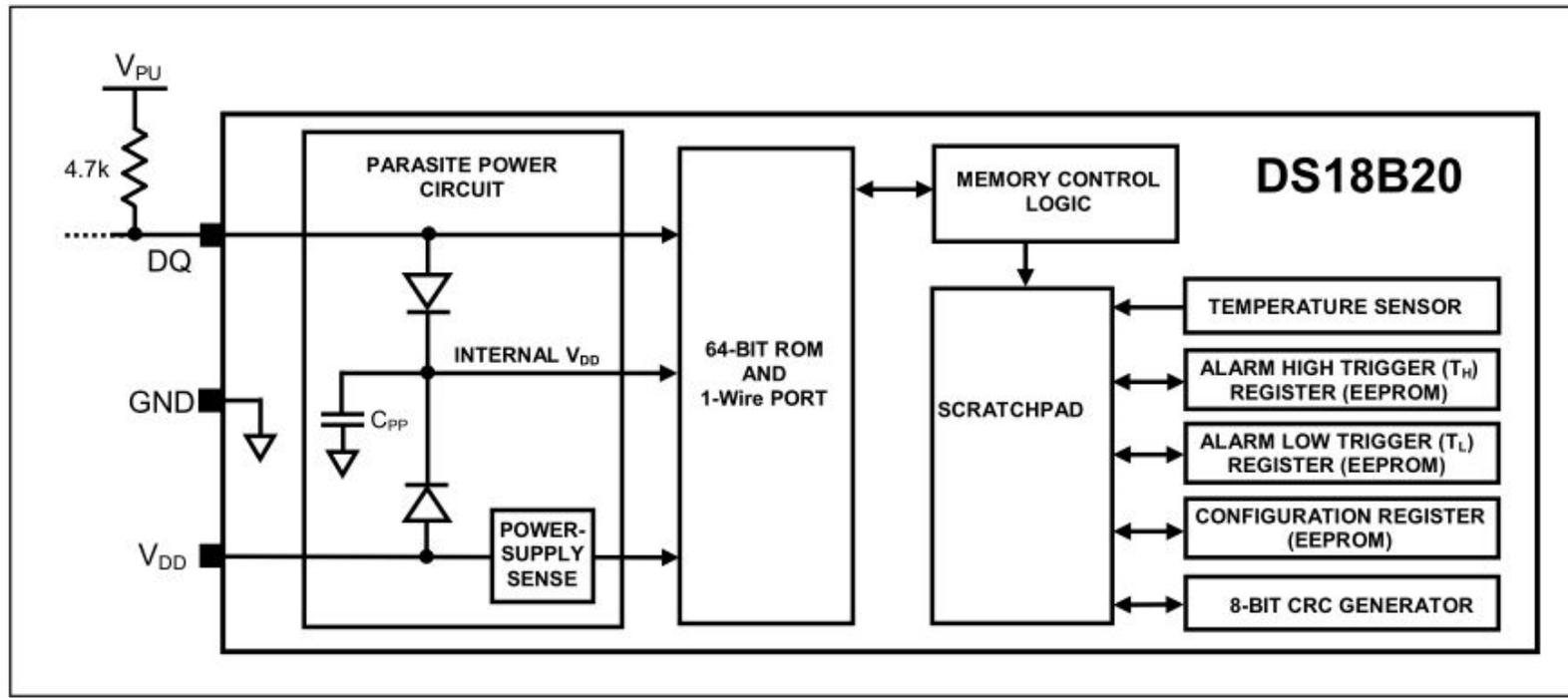
```
web_temperature5 | Arduino 1.8.13 (Windows Store 1.8.42...
Fichier Édition Croquis Outils Aide
web_temperature5 one_wire
2014-08-31 YM: log file named with Month and year; add pump sta
2014-09-26 YM: compute proportion of unused time by pump
2016-05-05 YM: memorize last temp 1 with pump activity
2019-03-10 YM: add yield() to main loop
*/
#define VERSION "1.5" // Module version, displayed on a WEB page
#include <Bridge.h>
#include <YunServer.h>
#include <YunClient.h>
#include <FileIO.h>
#include <OneWire.h>
#include <string.h>
```

DIGRESSION – ONEWIRE - DALLAS

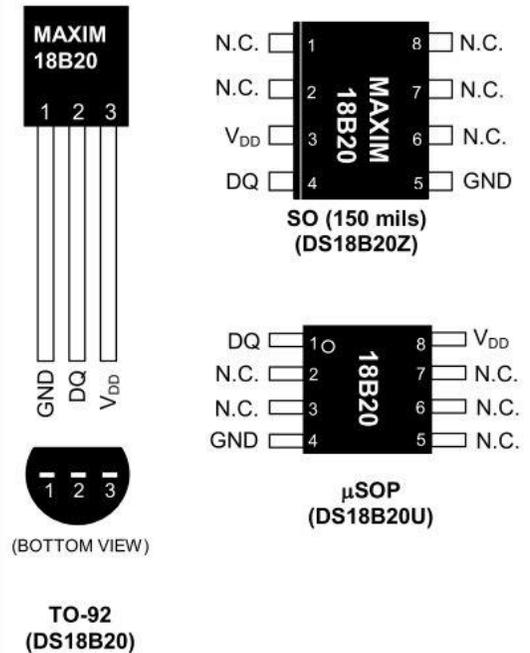
- DS18S20 – Thermomètre (sonde de température)
DS18B20 – Capteur de température avec résolution ajustable
- DS2404 – Horloge temps-réel, timer et NVRAM (4 KO)
- DS2405 – Interrupteur ou détecteur d'état (bascule)
- DS2406 – Double bascule (équivalent à 2x DS2405)
- DS2408 – Bascule 8 canaux
- DS2415 – Horloge temps réelle (RTC)
- DS2423 – Compteur d'impulsions / NVRAM 4KO
- DS2438 – Capteur de température / Convertisseur A/N
- DS2760 – Capteur de température, courant et convertisseur A/N
- DS2890 – Potentiomètre numérique

SONDE DE TEMPÉRATURE

Figure 1. DS18B20 Block Diagram

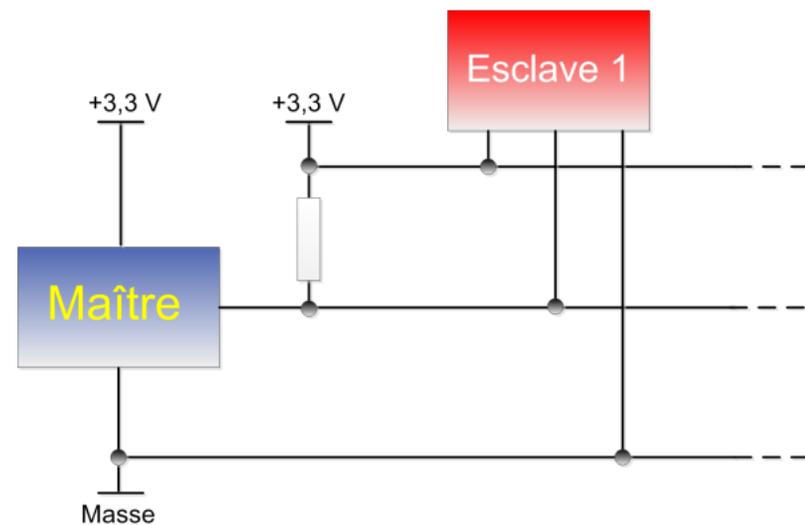
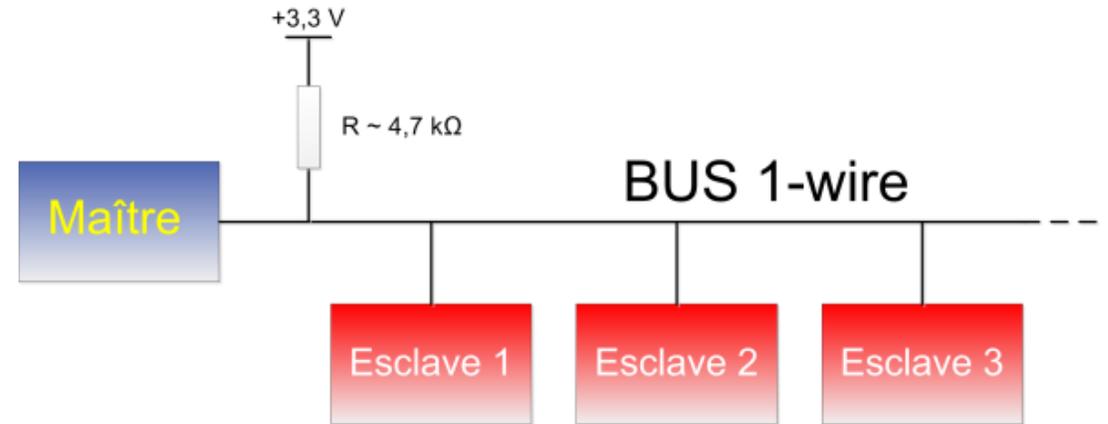


PIN CONFIGURATIONS



ONEWIRE - CÂBLAGE

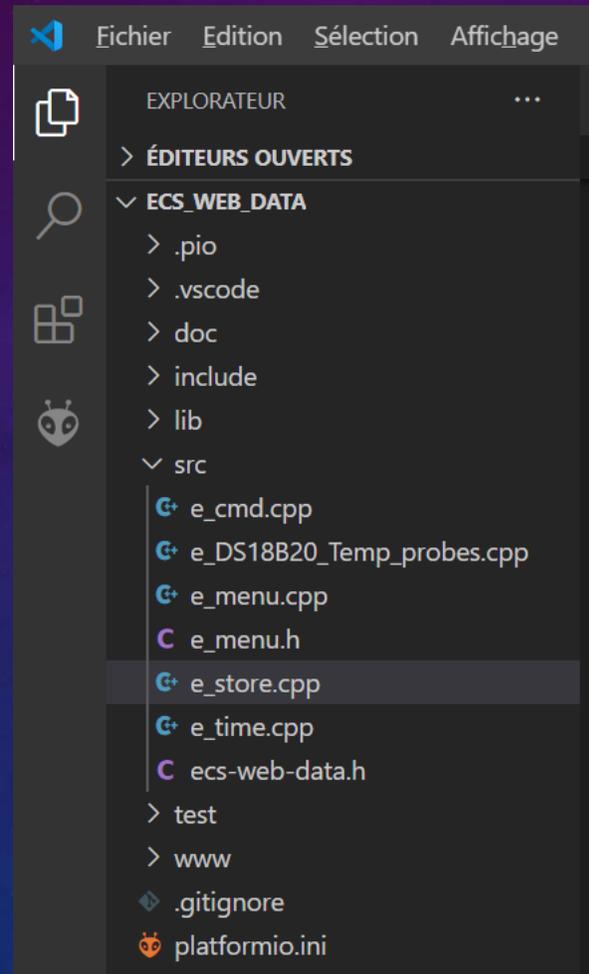
- Original One-Wire
 - Délai lecture ~ 750 ms
 - Un court-circuit casse tout
- Amélioration possible
 - Ajouter le +V sur un fil
 - Câbler en étoile



SOFTWARE

Avant

- Deux fichiers .ino
- Monitorer par serveur WEB local
- Temps, selon NTP Linux
- Lecture sondes à 10 minutes
- Sondes OneWire en parallèle
- Etat pompe via relais à contact
- Stockage données sur SD
- Pousser les données par FTP

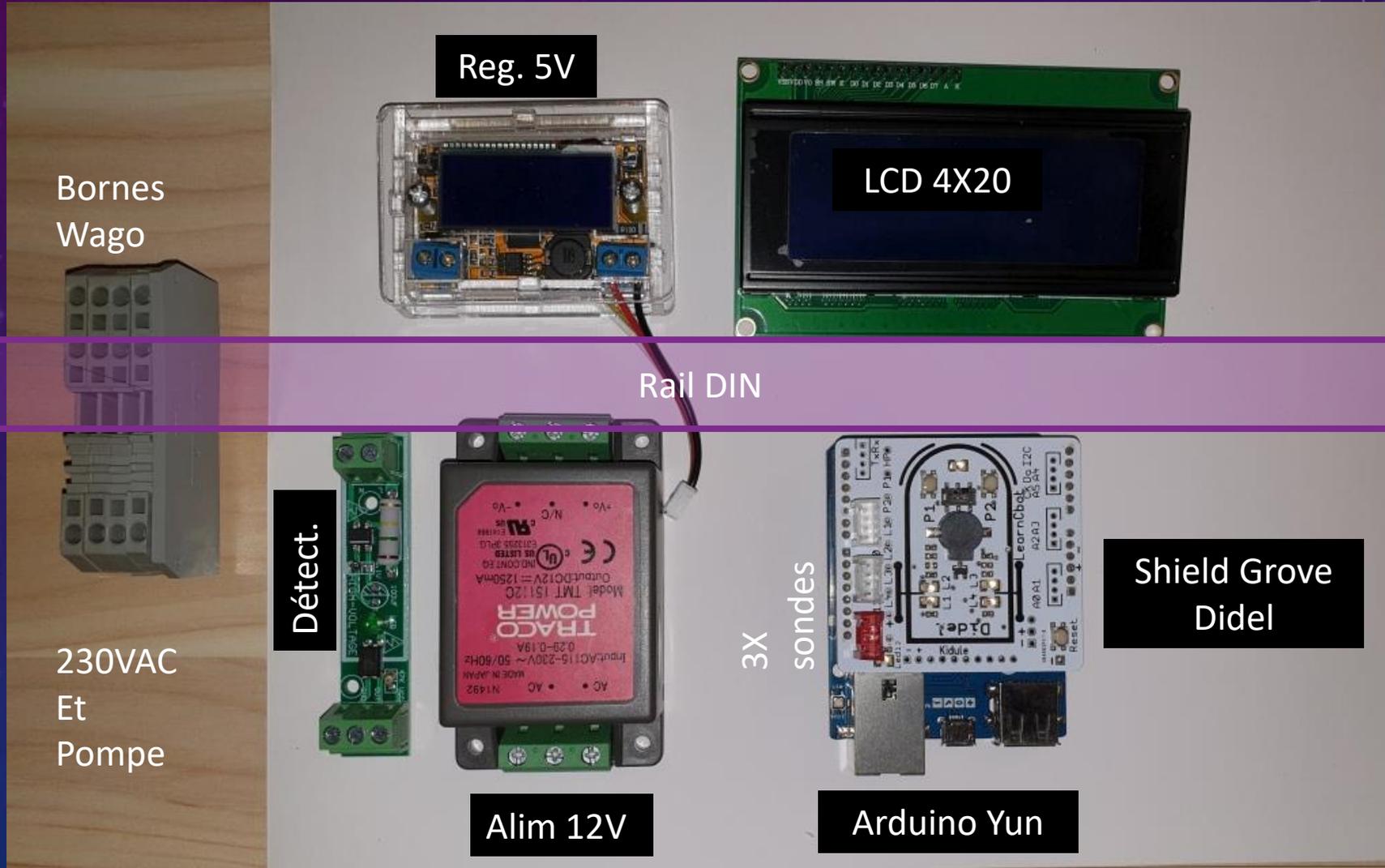


Après

- Modules CPP, headers séparés, VS-Code & PflO
- Monitoring par display LCD 4x20 chrs
- Temps : RTC + re-synchro par NTP
- Lecture sondes à la seconde
- Sondes en entrée assignées
- Etat pompe, via relais électronique
- Stockage données sur clé USB
- Pousser les données par FTP (idem)



POSER LE NOUVEL HARDWARE



RÉALISATION HARD

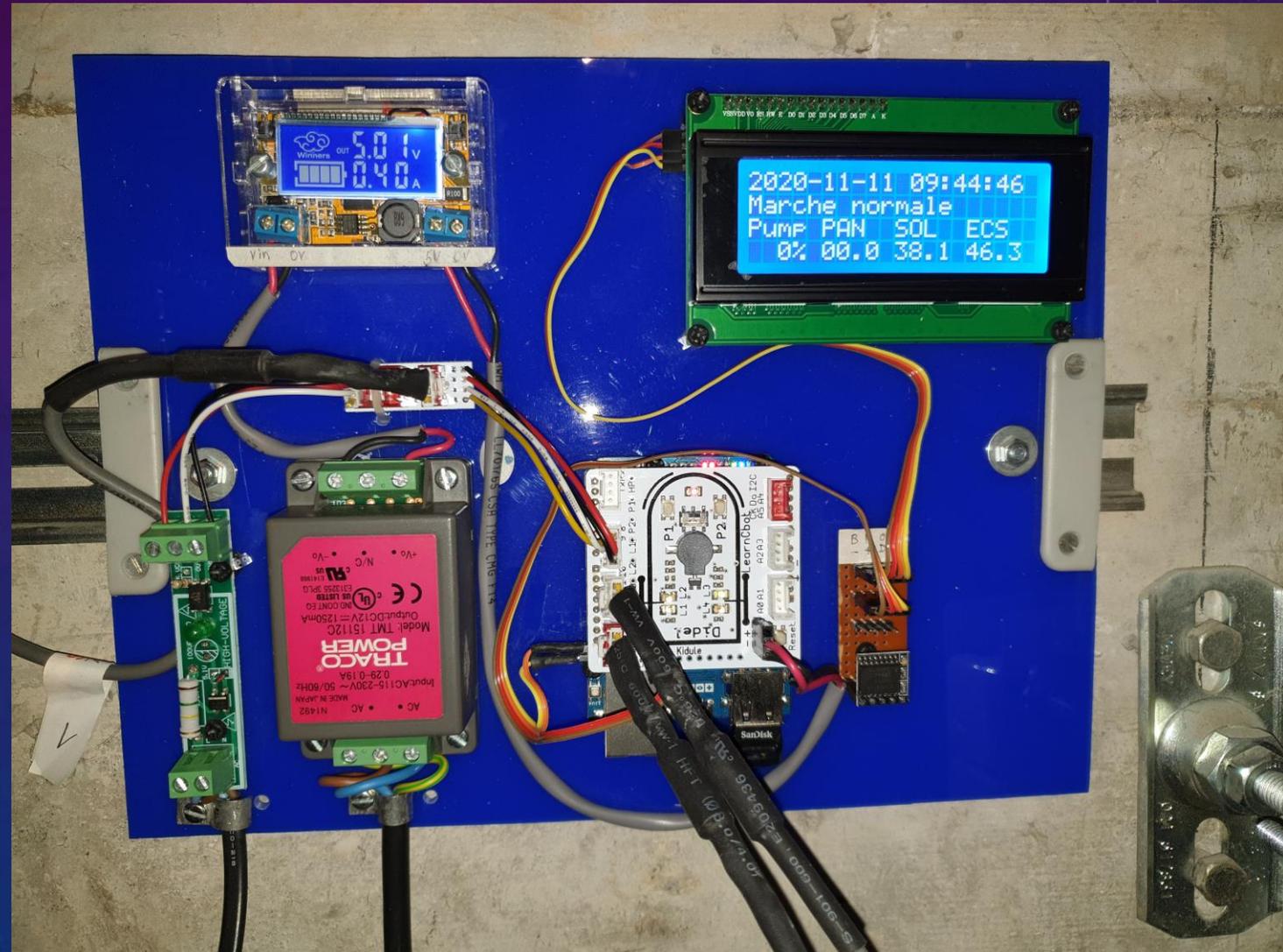
Alim, avec I/U contrôlés

Yun version 2, avec Didel
LearnCbot

Sondes sur inputs séparées

Petit bus Grove, petit bus I2C,
RTC

Relais électronique pour
pompe



SOFT VERSION 6

Complètement ré-écrit!

- Gestion temps réel avec **jm_Scheduler**
- RTC et synchro via Unix NTP
- Gestion de menus, avec 2 boutons
- Affichage de données temps réel sur LCD
- Log de fonctionnement enregistré
- Mesure % usage pompe revu



MODULES SOFT EMBARQUÉ

Github : https://github.com/ymasur/ECS_WEB_DATA

• e_cmd.cpp	8 763	11.11.2020 11:10	-a--
• e_DS18B20_Temp_probes.cpp	2 584	11.11.2020 10:36	-a--
• e_menu.cpp	6 448	11.11.2020 10:36	-a--
• e_menu.h	2 679	11.11.2020 11:13	-a--
• e_store.cpp	3 804	11.11.2020 12:10	-a--
• e_time.cpp	5 129	06.11.2020 09:34	-a--
• ecs-web-data.h	4 704	11.11.2020 11:11	-a--

POMPE ACTIVE, TAUX D'ACTIVITÉ

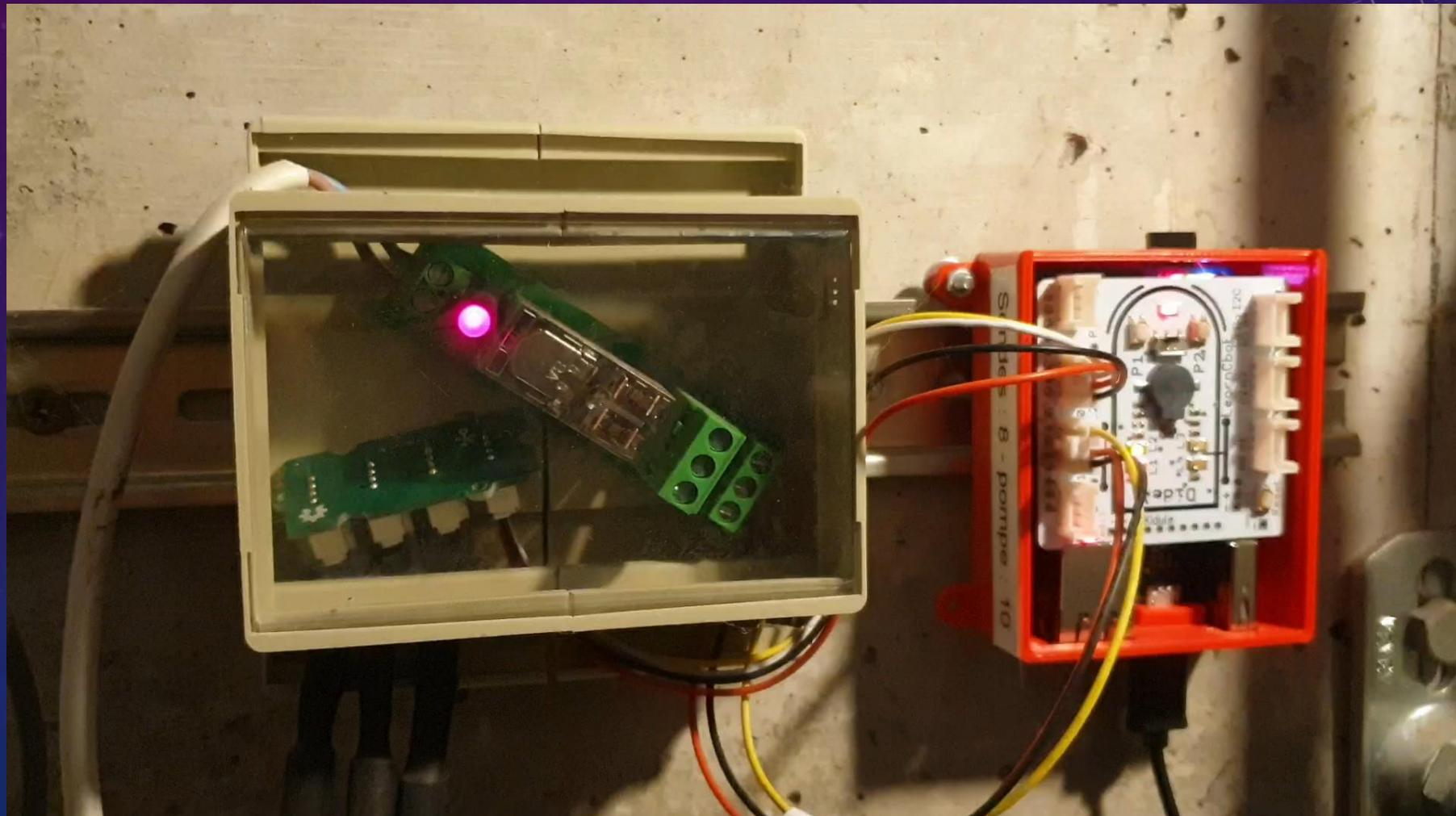
Avant

- Relais électromécanique
- PWM: douteux, durée de vie...
- T échantillon : variable
- Selon boucle en free run
- Sommateur en long 32 bits

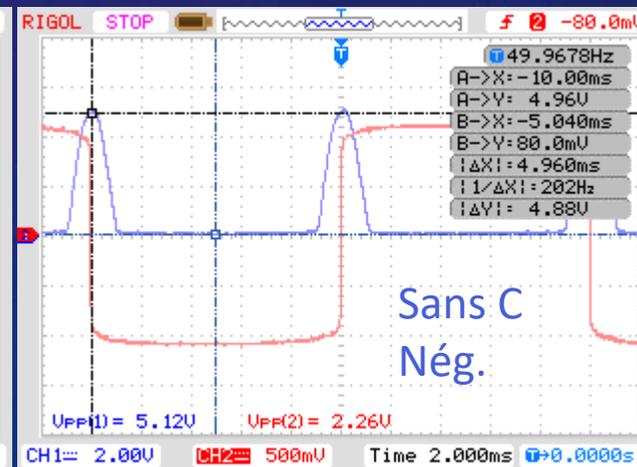
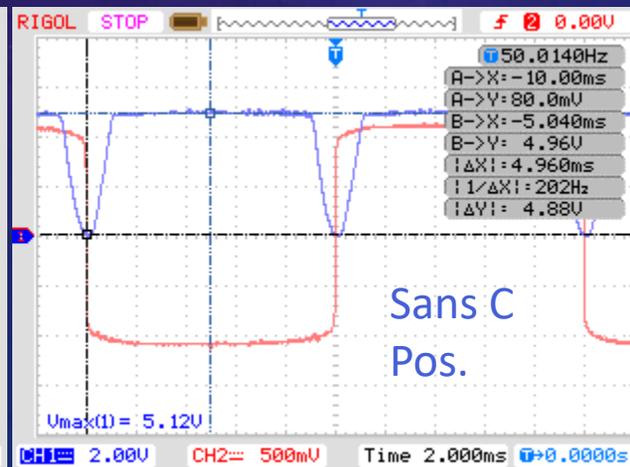
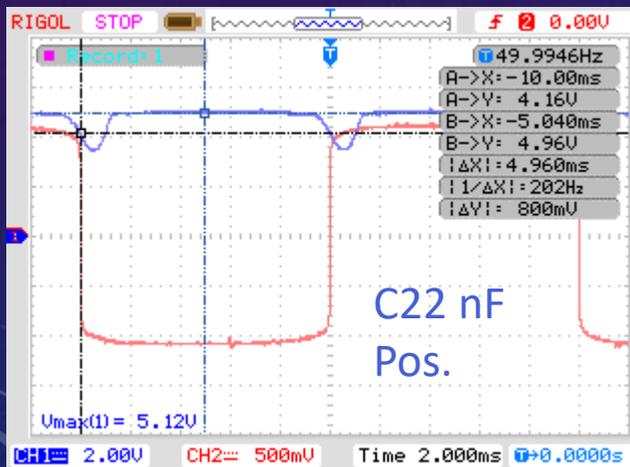
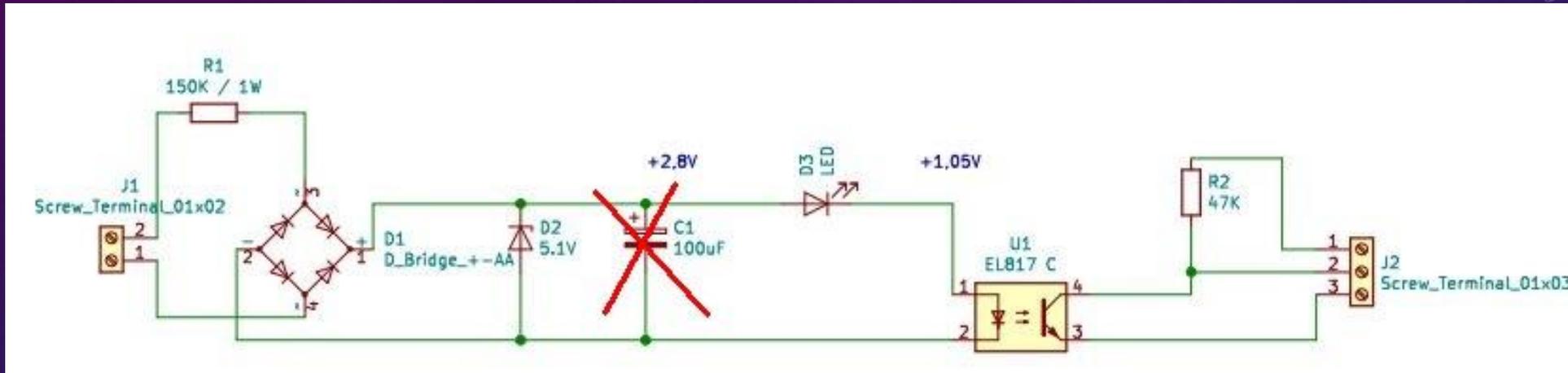
Après

- Relais électronique modifié
- Suivi PWM complet
- T échantillon = 23 ms
- Différent de f réseau 50 Hz
- 20 ms; 10 ms en passage à 0
- Calcul en float; sur 10 minutes
- Correction pour passage à 0 (env. 6%)

AVANT: VIDÉO PWM

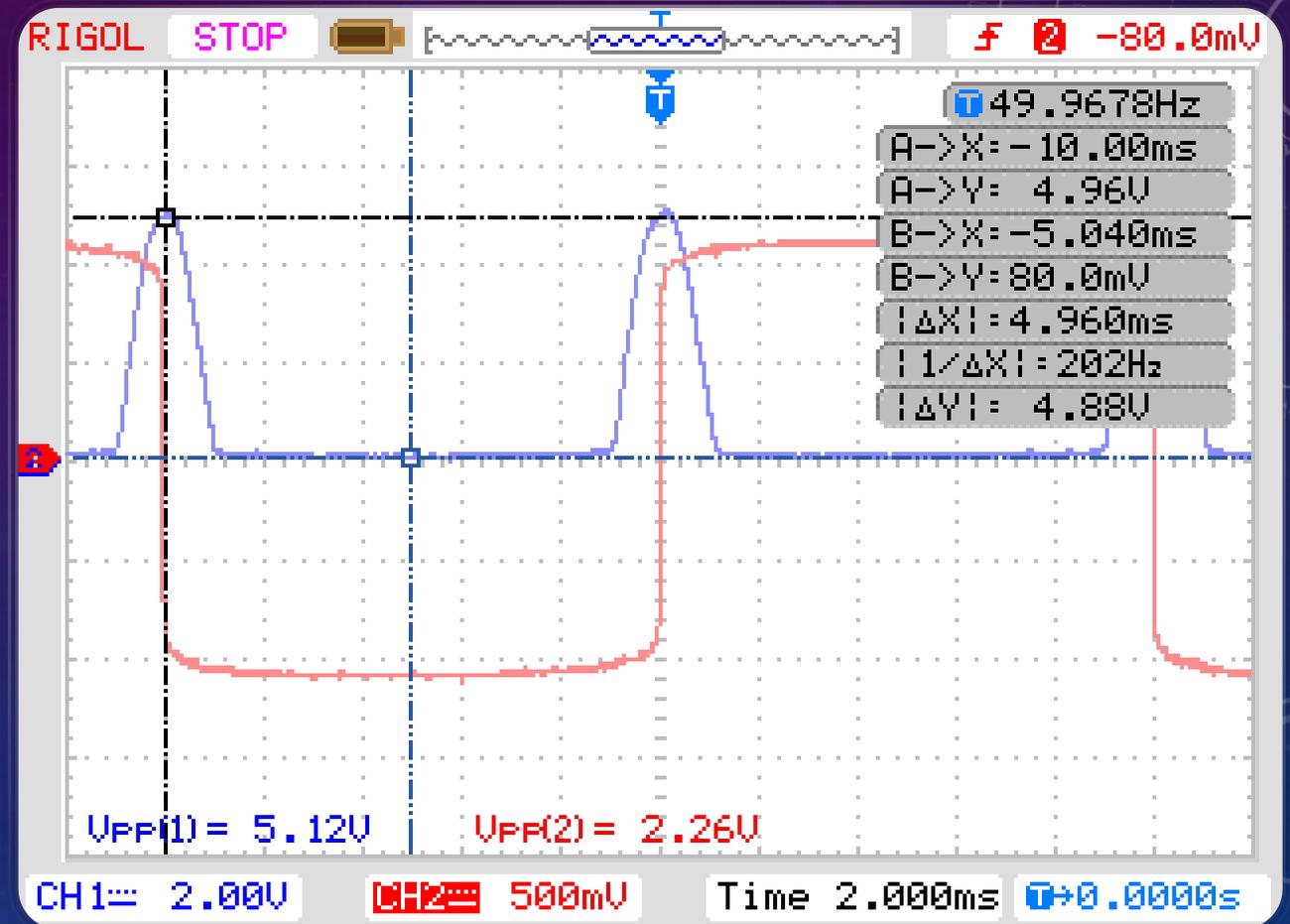


RELAIS - SCHÉMA



REAL TIME RAPIDE

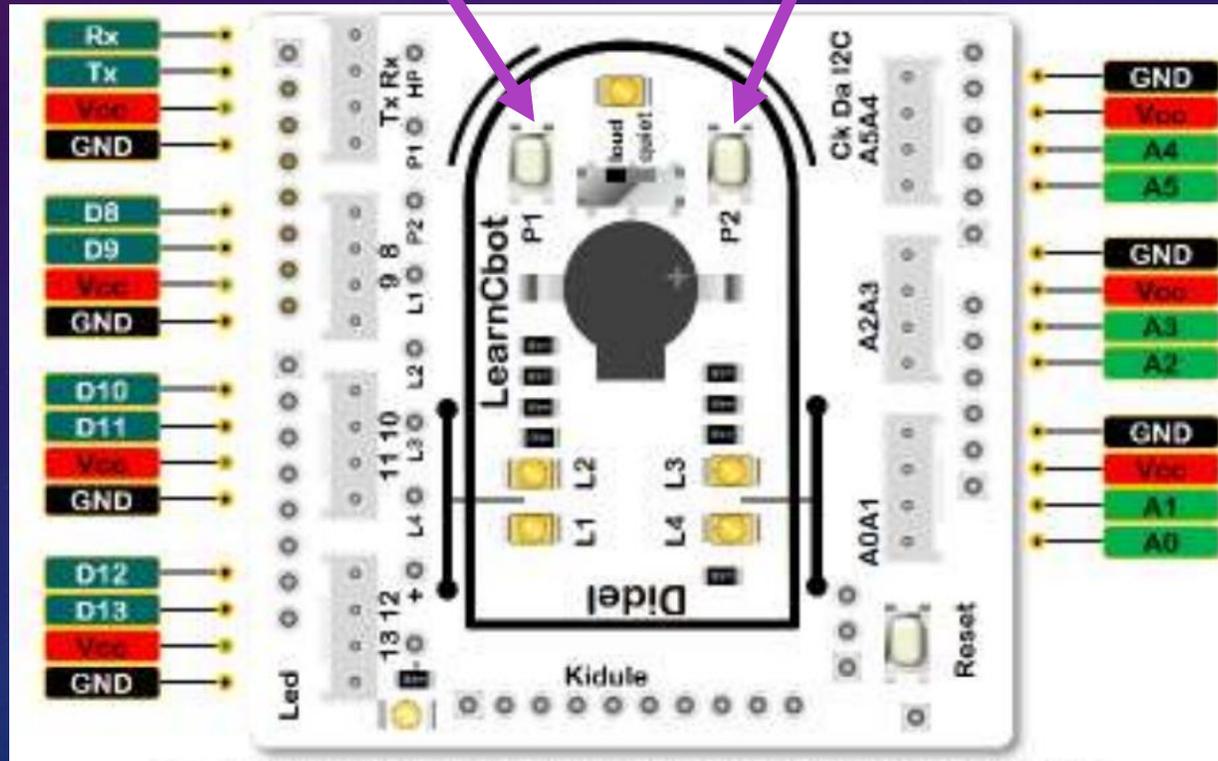
- La pompe est en PWM 50 Hz
- Pas de filtre RC du relais
- Ne pas échantillonner sur un multiple de 10 ms
- Choix: 23 ms



INTERFACE DIDEL LEARNTCBOT

Menu +

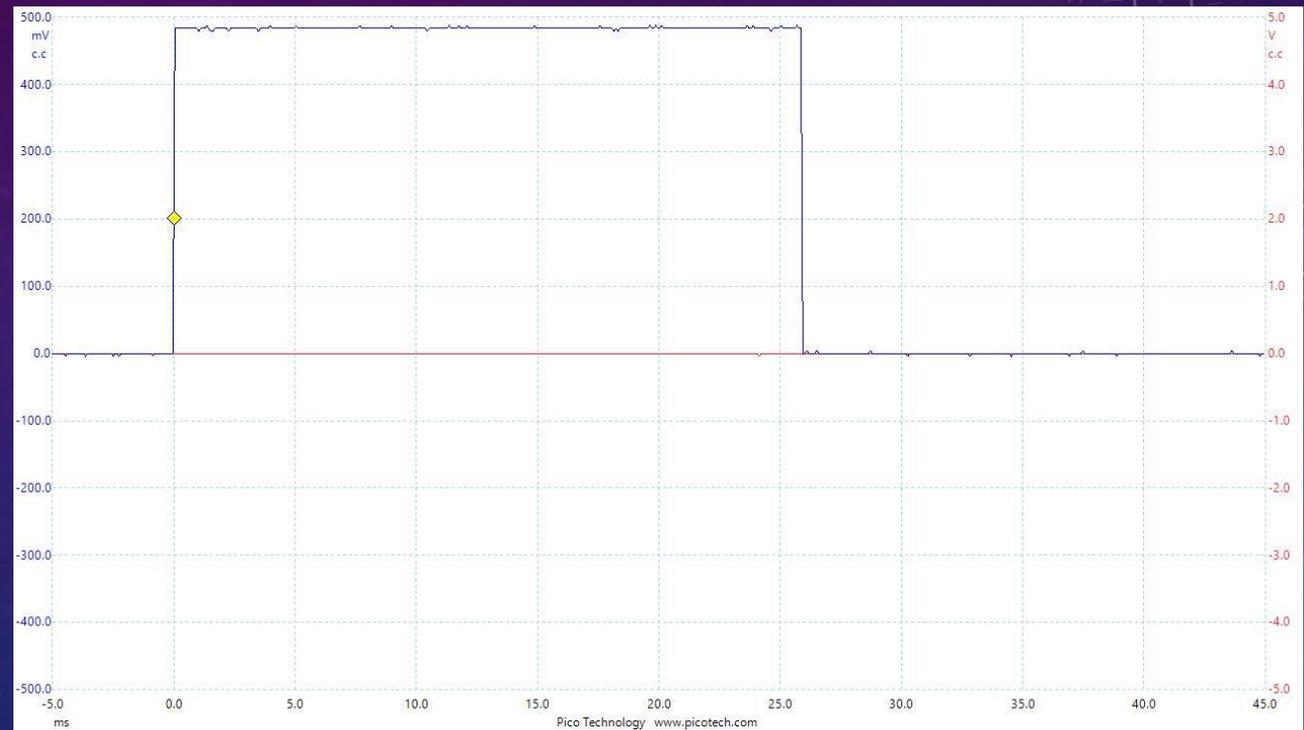
Menu -



MESURE SCAN SONDE

Env. 26 ms

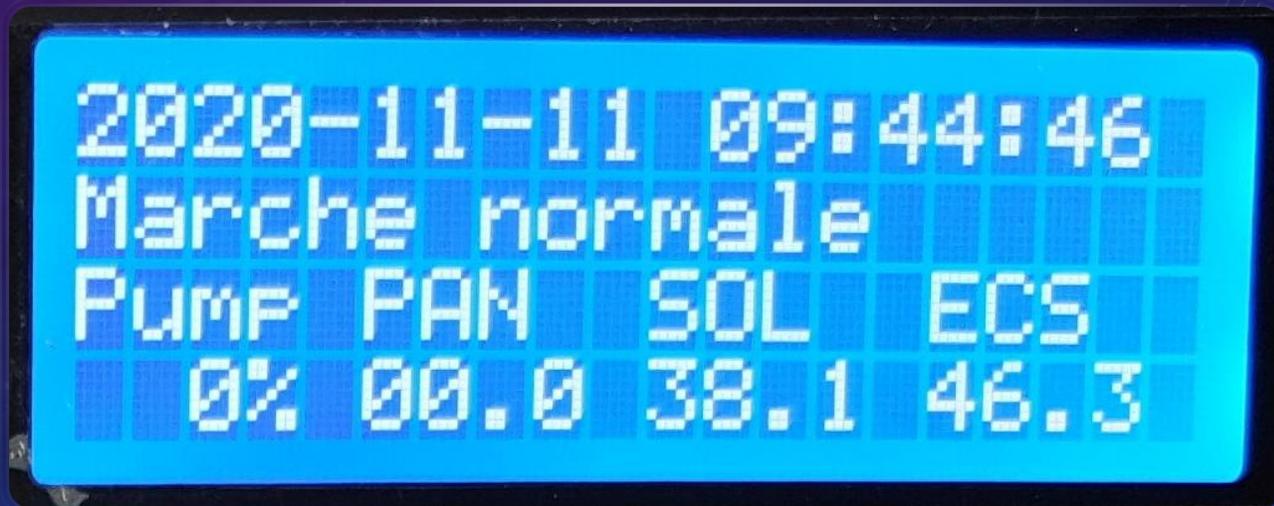
OK pour un scan à la seconde de 3 sondes



REALTIME PRINCIPAL, 1 SECONDE

Dans cet ordre:

- À 03h00: synchro de la RTC
- Scan des sondes
- Calcul ratio pompe en %
- Affiche la date/heure, valeurs
- À 10 min, enregistrement sur SD
- Led de vie, état inversé



LOG MESSAGES

Messages

- 2020-11-11 12:11:34 ECS-WEB-data 6.01
- 2020-11-11 12:14:13 Sonde 1 KO !!
- 2020-11-11 12:14:21 Sonde 1 OK !!
- 2020-11-11 12:14:28 Get NTP time...
- 2020-11-11 12:14:28 T Correction: -1
- 2020-11-11 12:14:28 Time NTP = RTC

LCD, ligne 1 (sur 0..3)

- ERREUR : le message reste affiché, jusqu'à la disparition de l'erreur
- Info, le message disparaît après 15 sec. Ou un changement de menu

MENU PAR SÉRIEL

Interpréteur à 1 caractère

- '+' : incrément index menu
- '-' : décrément index menu
- '0' à '4' : choix menu
- '?' : aide

MENU 0

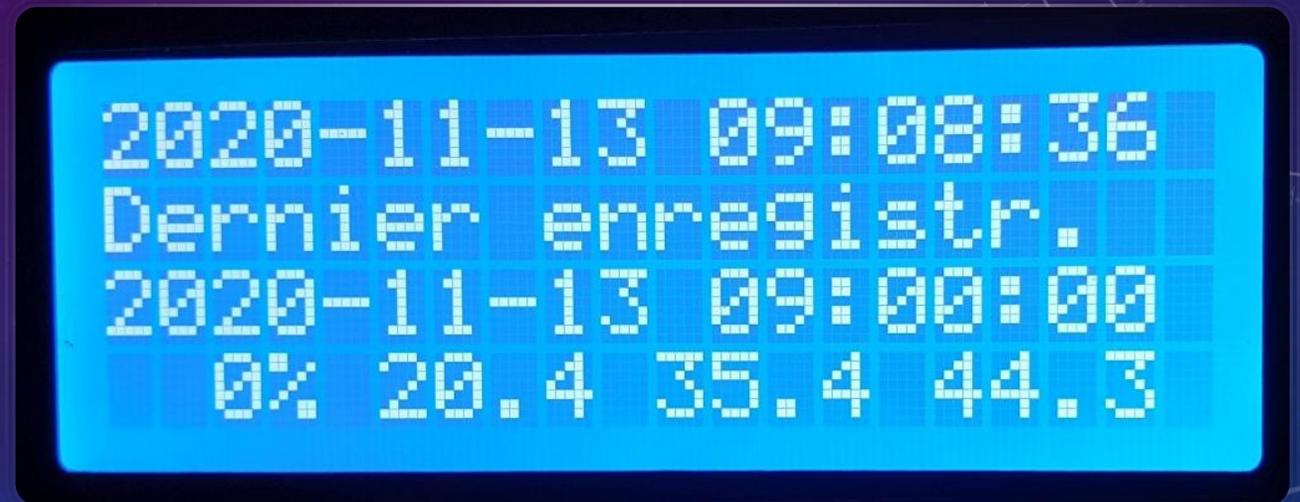
- Ln0: date/heure
- Ln1 : message / dernier log
- Ln2 : textes des champs
- Ln3 : mesures en cours



- Si le menu est 0 depuis plus de 5 minutes (300 sec.), afficher en Ln1 message « Marche normale »

MENU 1

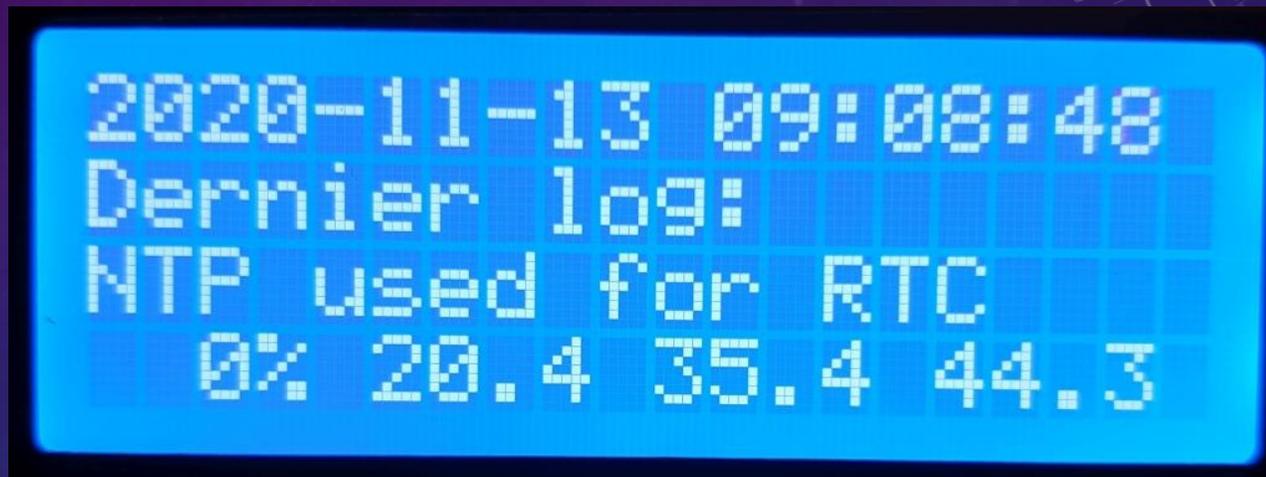
- Date/valeurs du dernier enregistrement



MENU 2

Message du dernier log

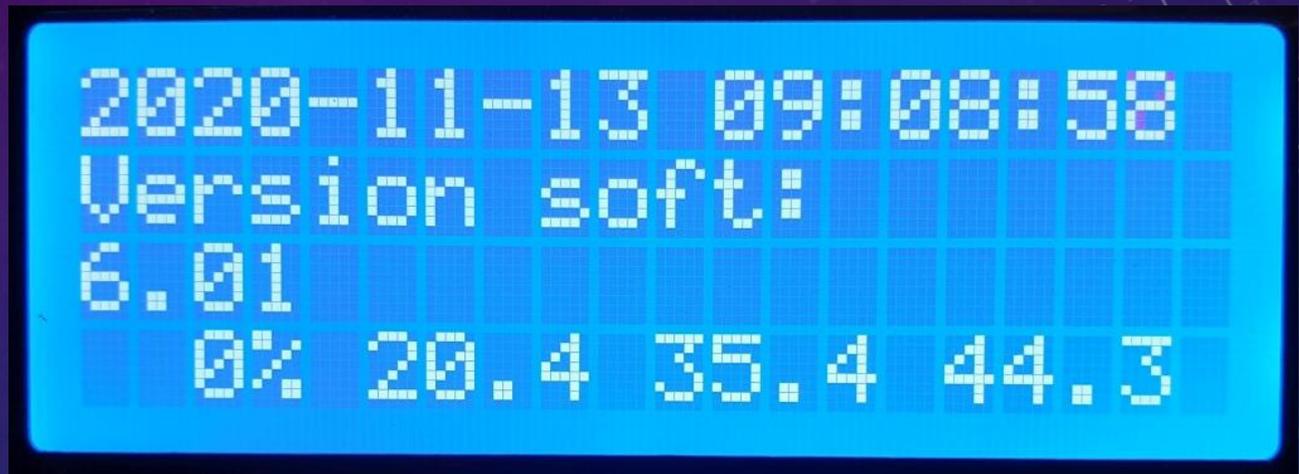
Ligne du bas: valeurs en temps réel



MENU 3

Version soft

Ligne du bas: valeurs en temps réel

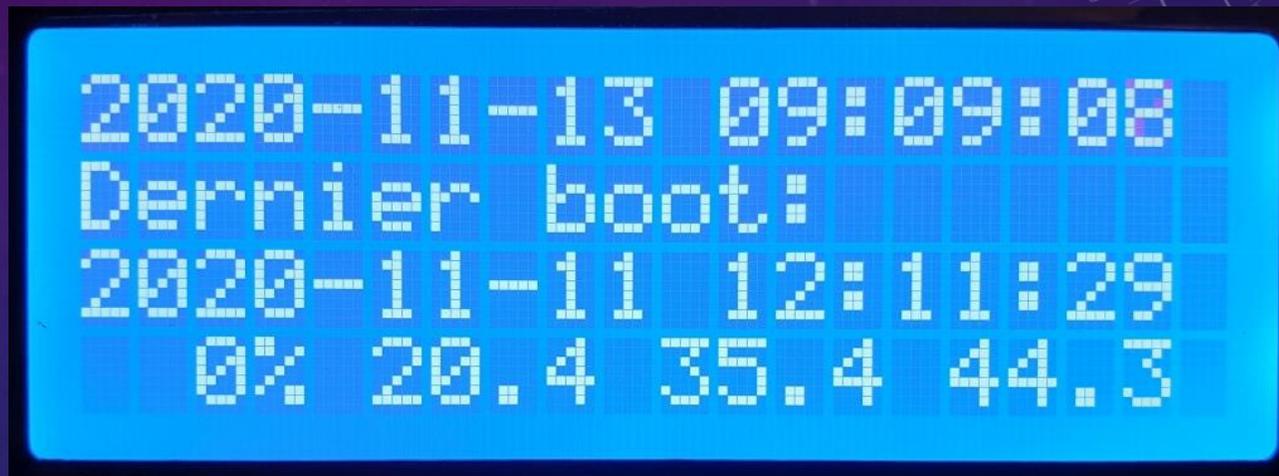


MENU 4

Ln 1: Dernier boot

Ln2 date/heure boot

Ligne du bas: valeurs en temps réel



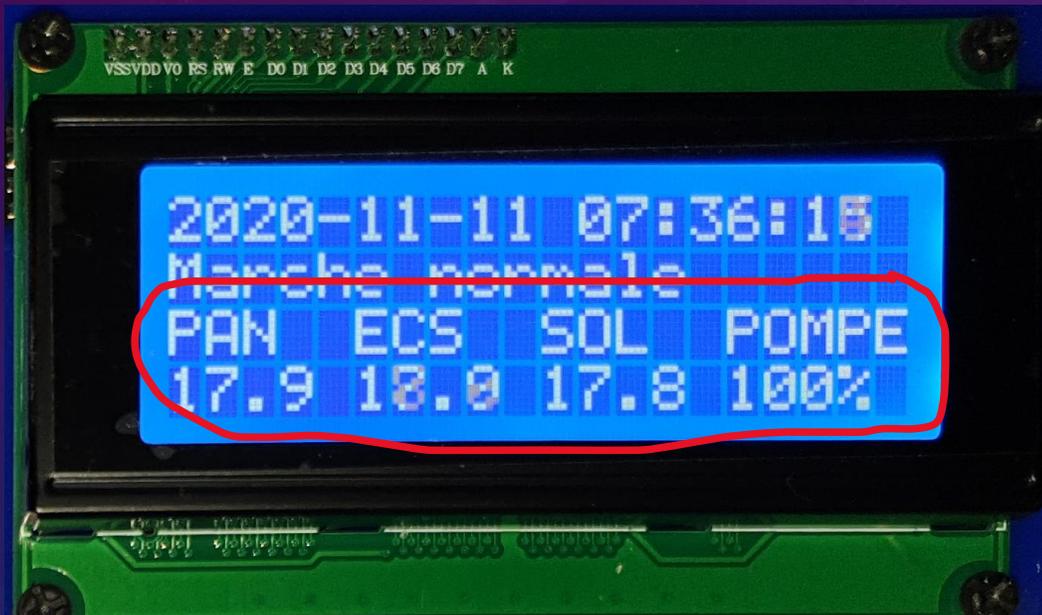
MISE EN SERVICE

RÉSULTATS



HUM... HUM...

Ordre des valeurs... !



Ordre des valeurs corrigé!



CORRECTIONS

- Ordre des données enregistrées (compatibilité sur site WEB!)
- Taux pompe corrigé de 6%, par facteur défini:

```
#define PUMP_CORRECTION (1.06) // due of phase dead time
```

RÉSULTATS, DERNIÈRES CORRECTIONS

- Usabilité: OK
- Compatibilité des données sur site WEB : OK
- Fiabilité: à améliorer? 2 reboot en 45 jours
- Horloge se décale de ~8 sec par jour?
 - Conflit de fichier avec le FTP possible
 - FTP décalé de 12 secondes

DÉRIVE HORLOGE?

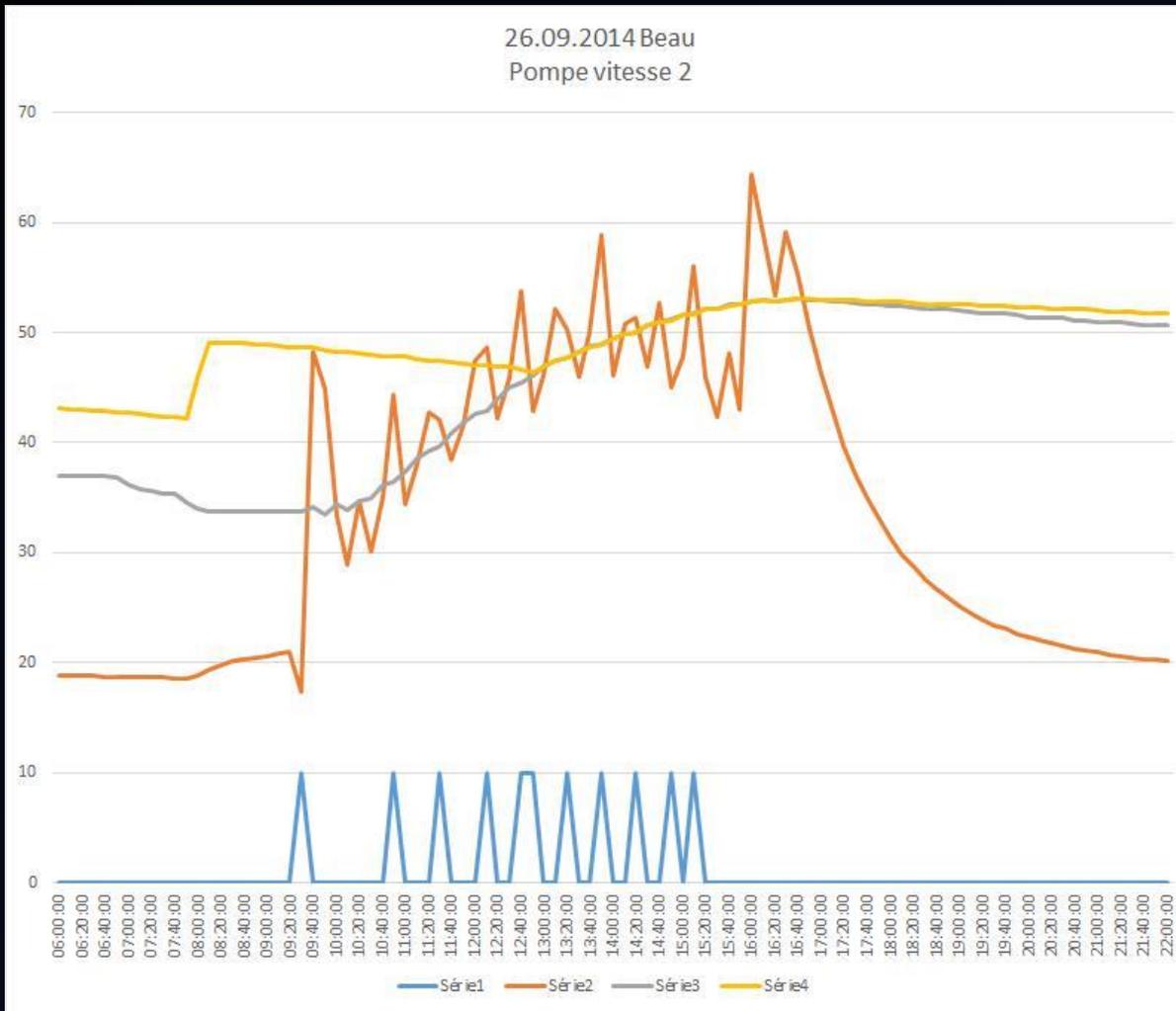
2021-01-25 03:02:59	Get NTP time...
2021-01-25 03:02:59	T Correction: -8
2021-01-25 03:02:59	Get NTP time...
2021-01-25 03:02:59	NTP used for RTC
2021-01-26 03:02:59	Get NTP time...
2021-01-26 03:02:59	T Correction: -9
2021-01-26 03:02:59	Get NTP time...
2021-01-26 03:02:59	NTP used for RTC
2021-01-27 03:02:59	Get NTP time...
2021-01-27 03:02:59	T Correction: -8
2021-01-27 03:02:59	Get NTP time...
2021-01-27 03:02:59	NTP used for RTC

```
time.sleep(15)    # be sure Arduino complete the file
ftp = ftplib.FTP(siteweb,username,passwd)
file = open(inpath + infile,'rb')
# file to send
ftp.storbinary('STOR ' + outpath + infile, file)  # send the file
file.close()
```

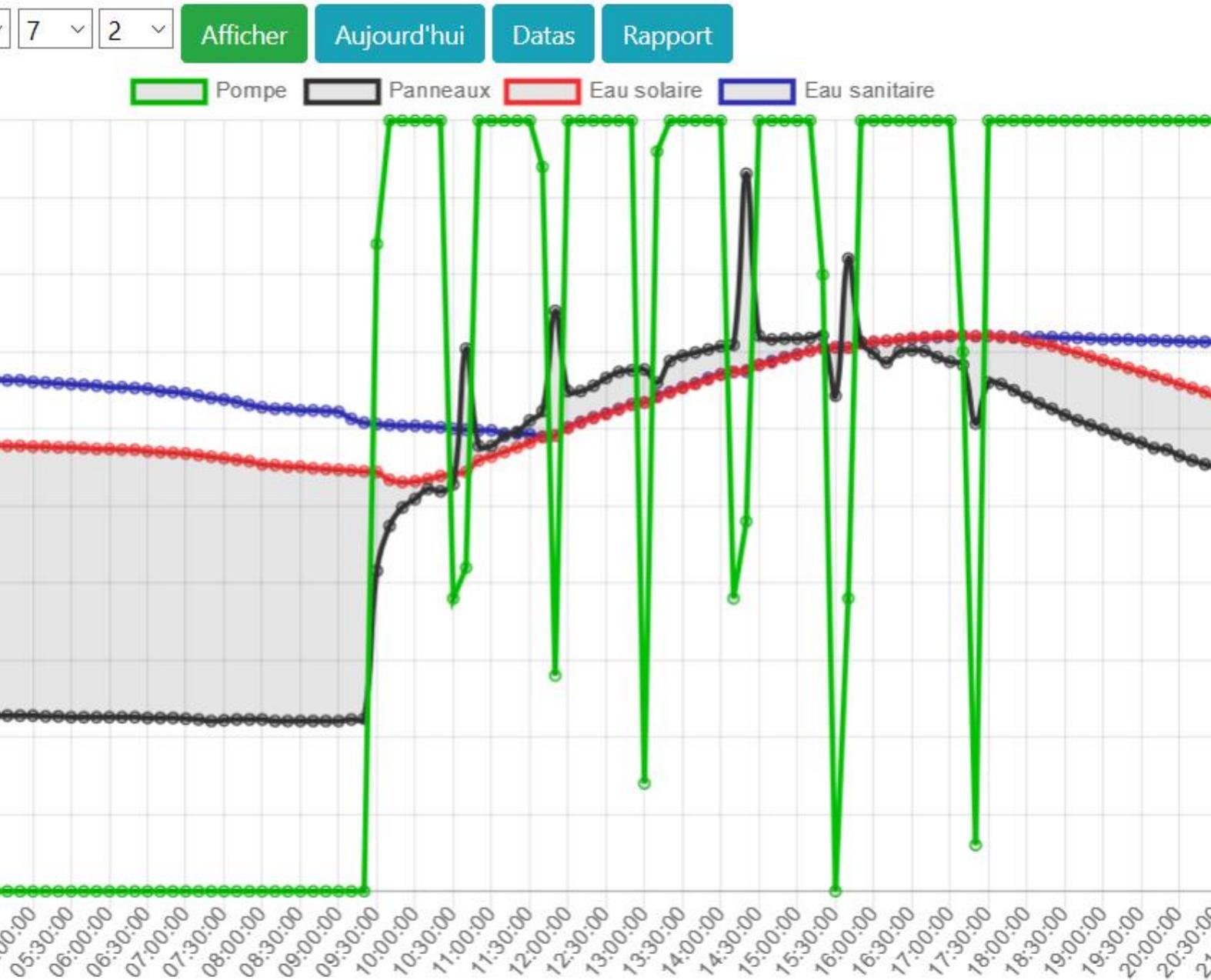
ECS MONITOR - UTILITÉ

- Réglage de la vitesse de la pompe
- On améliore pour indiquer en %

Analyse – vitesse pompe: 1, 2 ou 3?



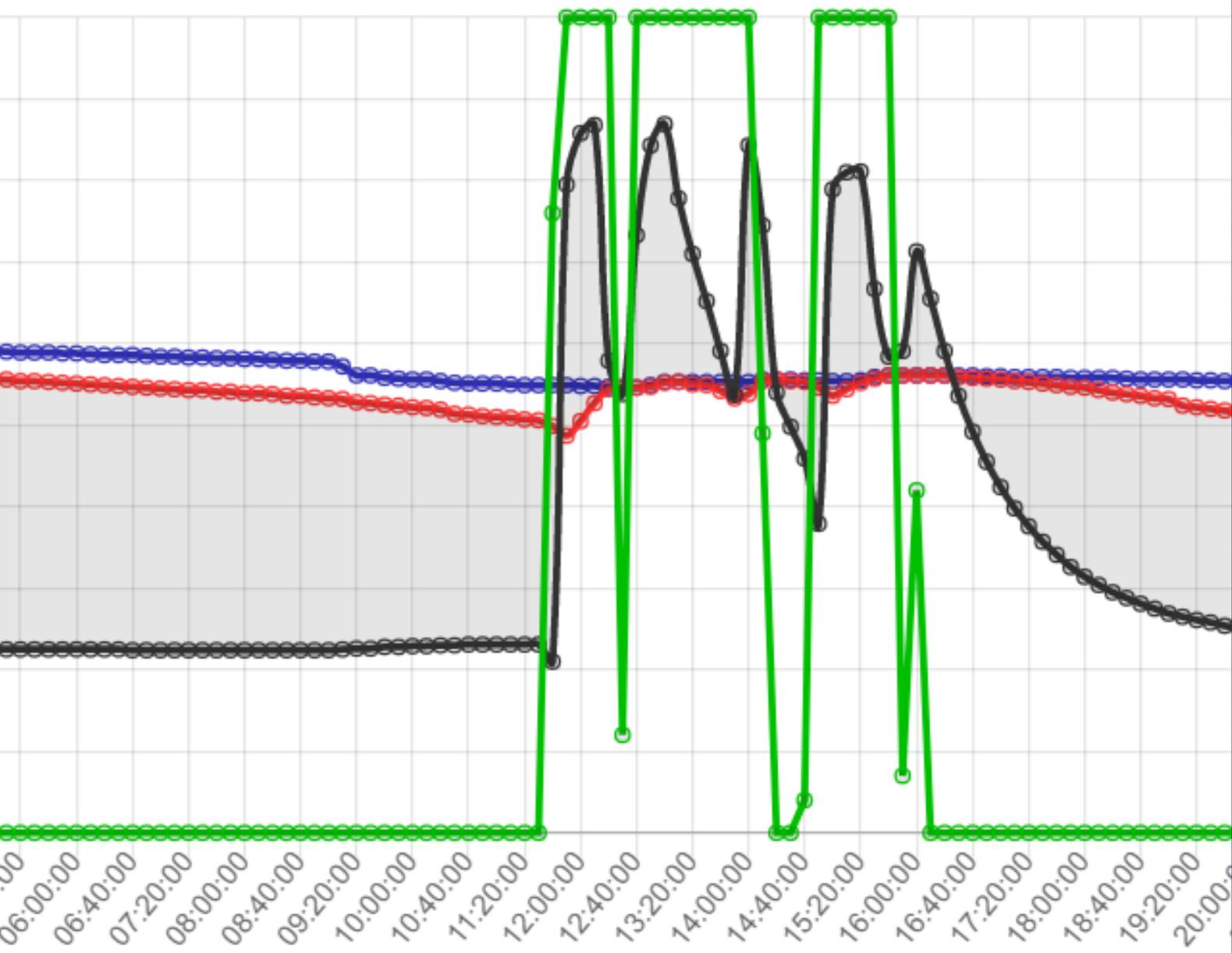
hique)



BOITIER DÉBLOQUE

- La pompe tourne la nuit!!
- Refroidit le volume solaire

 Pompe  Panneaux  Eau solaire  Eau sanitaire



MANQUE DE GLYCOL

- La pression insuffisante ne permet plus le transport de chaleur
- L'excès de t° déclenche la pompe et affiche une alarme

Data du:

2021 ▾

1 ▾

1 ▾

Afficher

Aujourd'



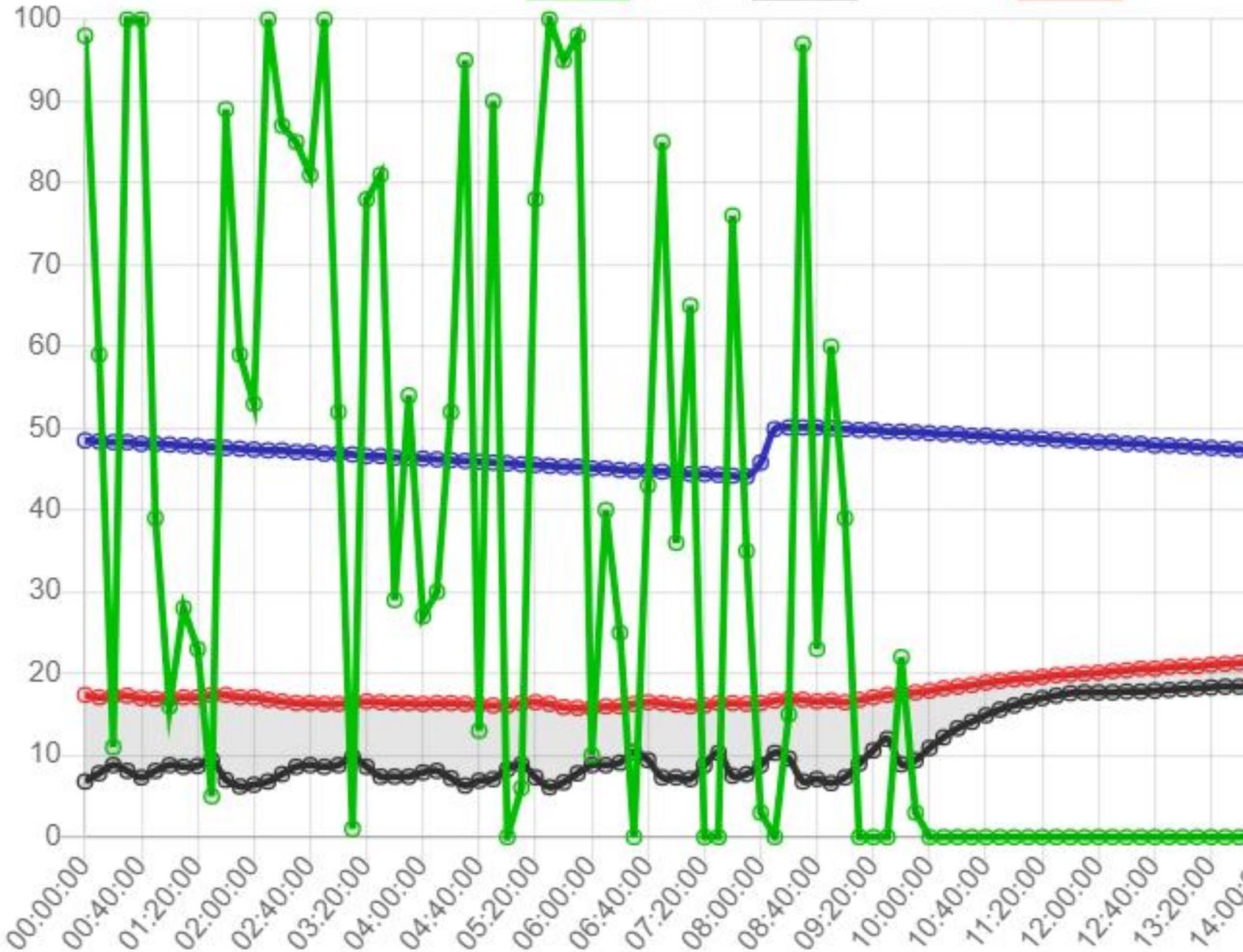
Pompe



Panneaux



Eau sola



SONDE DE PANNEAU DÉBLOQUE

- Se croit à -10°C !
- Swing de la pompe de nuit!
- Refroidit l'eau solaire...

Data ECS (graphique)

Data du: 2021 2 1

Afficher

Aujourd'hui

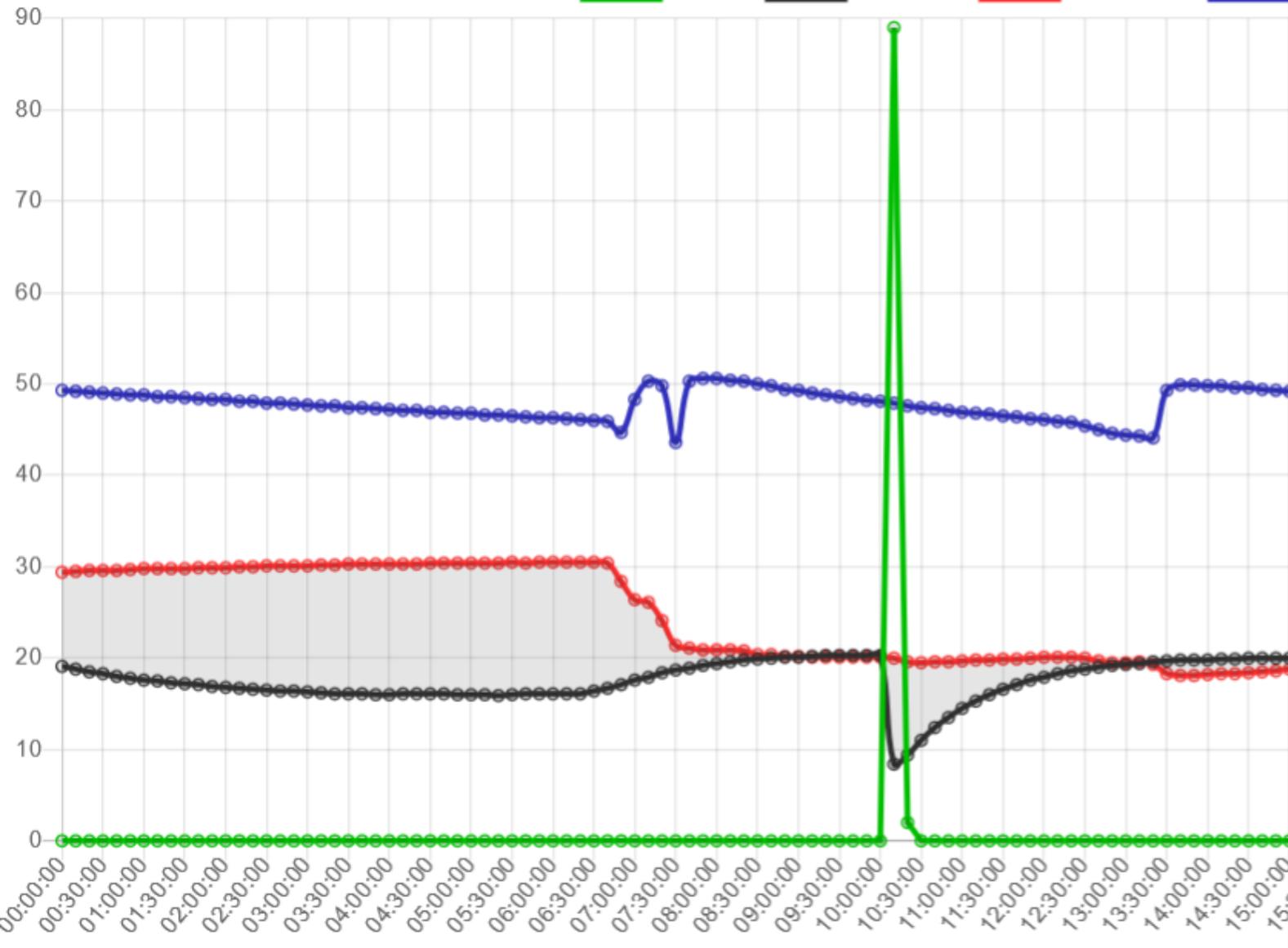
Datas

Rapport

Pompe

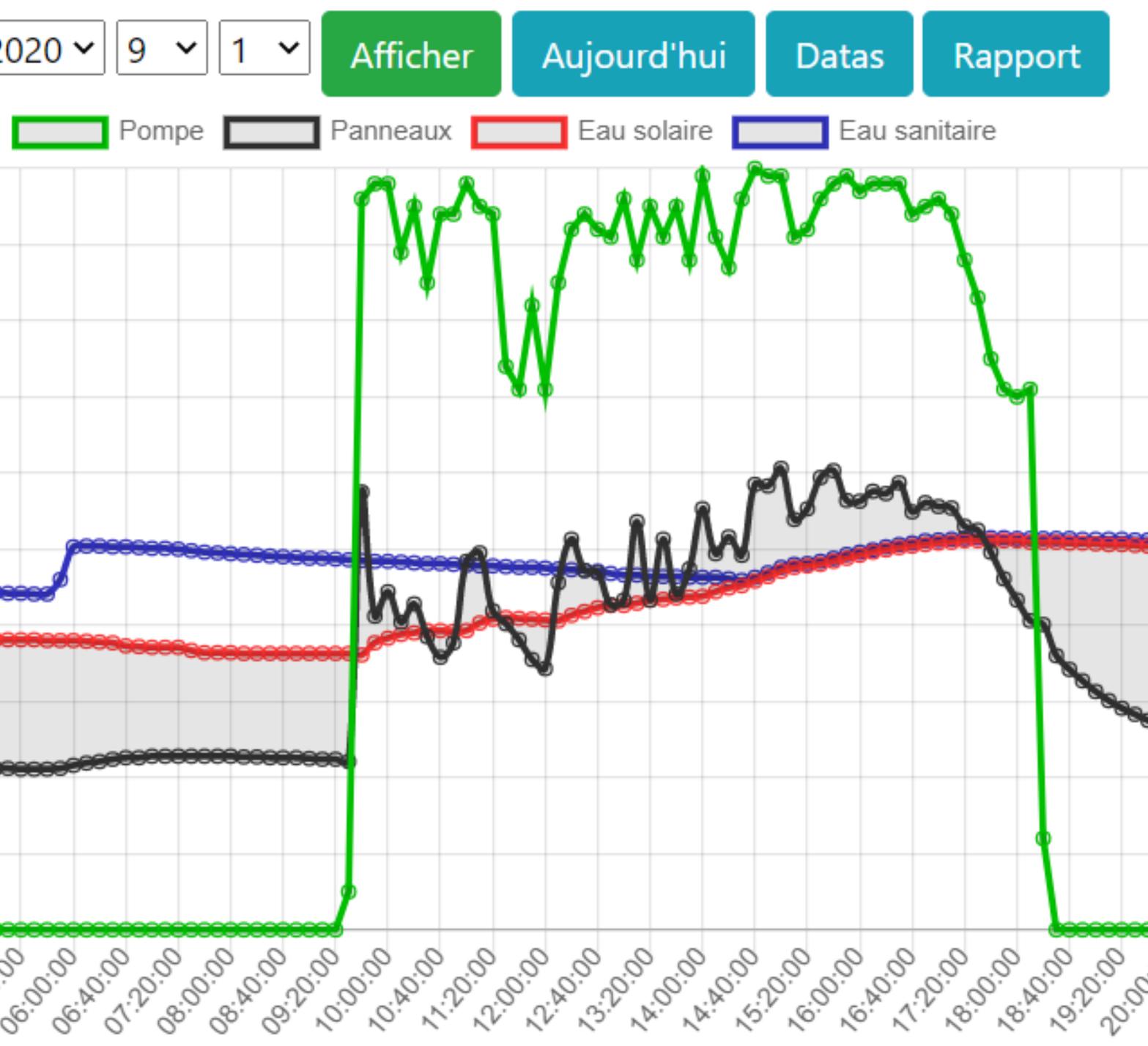
Panneaux

Eau solaire



FUTUR

- Calcul du gaz utilise pour l'ECS
 - Déterminer la montée de t°
 - Bilan annuel
- Alerte en cas de "déraillement"
 - Envoi d'un email



FONCTIONNEMENT CORRECT

- La vitesse de la pompe est selon la différence de température (zone grise)

DES QUESTIONS ?

Merci pour votre attention!

- Dépôt GIT du projet: https://github.com/yomasur/ECS_WEB_DATA
- Partie WEB: <https://github.com/yomasur/ecs-web-monitoring>
- Valeurs : <https://ecs.yvesmasur.ch>

