



COMPANION POUR MSUN'PV

Le Companion pour le routeur solaire Msun'PV permet un affichage externe des valeurs envoyées par le routeur. Son format permet de le positionner où vous le souhaitez pour une consultation aisée.



Le module d'affichage choisi est un Lilygo T-Display S3 à base de processeur ESP2 autorisant une communication WiFi aisée et évitant le brochage avec un écran.

Vous aurez ainsi les informations concernant :

- La production des panneaux solaires installés
- La consommation instantanée de votre domicile
- La consommation utilisée vers le cumulus et sa température de chauffe actuelle
- Un bouton assistant vous indiquant la possibilité de lancer une machine énergivore
- La date et l'heure actuelle (avec gestion des heures été/hiver)
- La météo et la température extérieure actuelle
- Divers voyants indiquant la mise en route d'un chauffage électrique ou le point de givre

Les programmes fournis vous permettent de l'adapter à votre configuration et de modifier la position des icônes comme vous le souhaitez. L'installation est détaillée ici et ne nécessite aucune connaissance dans la programmation. Vous pourrez vous en construire un par vous même, et le forum Ard-tech vous donnera d'autres informations et la résolution de problèmes rencontrés.

Dans ce document , vous pouvez cliquer sur les liens qui émaillent les pages, et vous dirigeront automatiquement vers le bon site internet. Bonne lecture et bon bricolage !

JJHontebeyrie (alias Baroudeur)

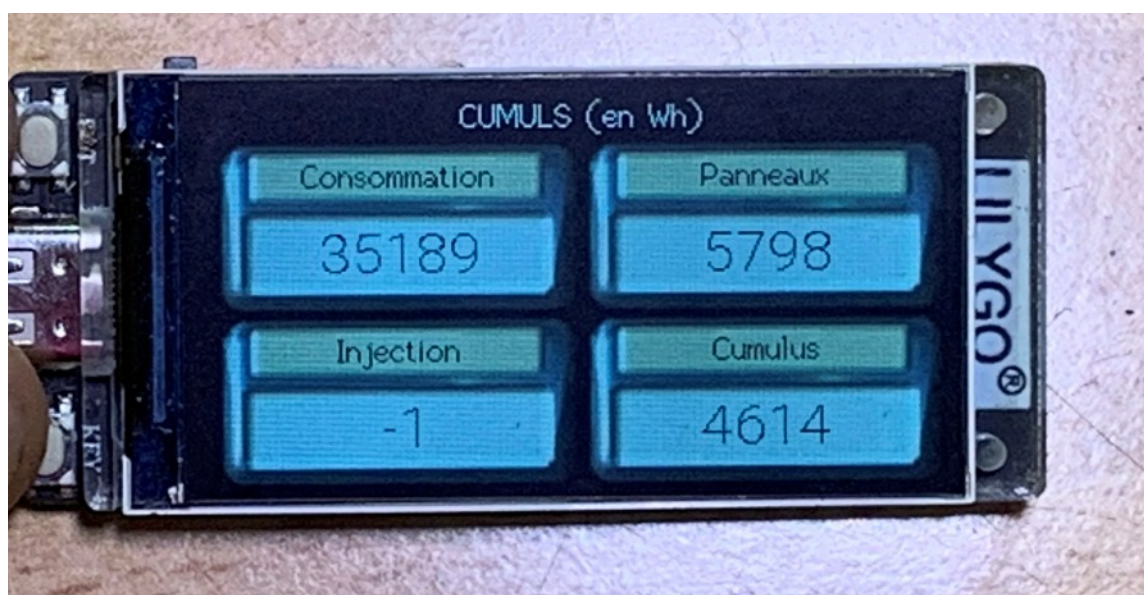
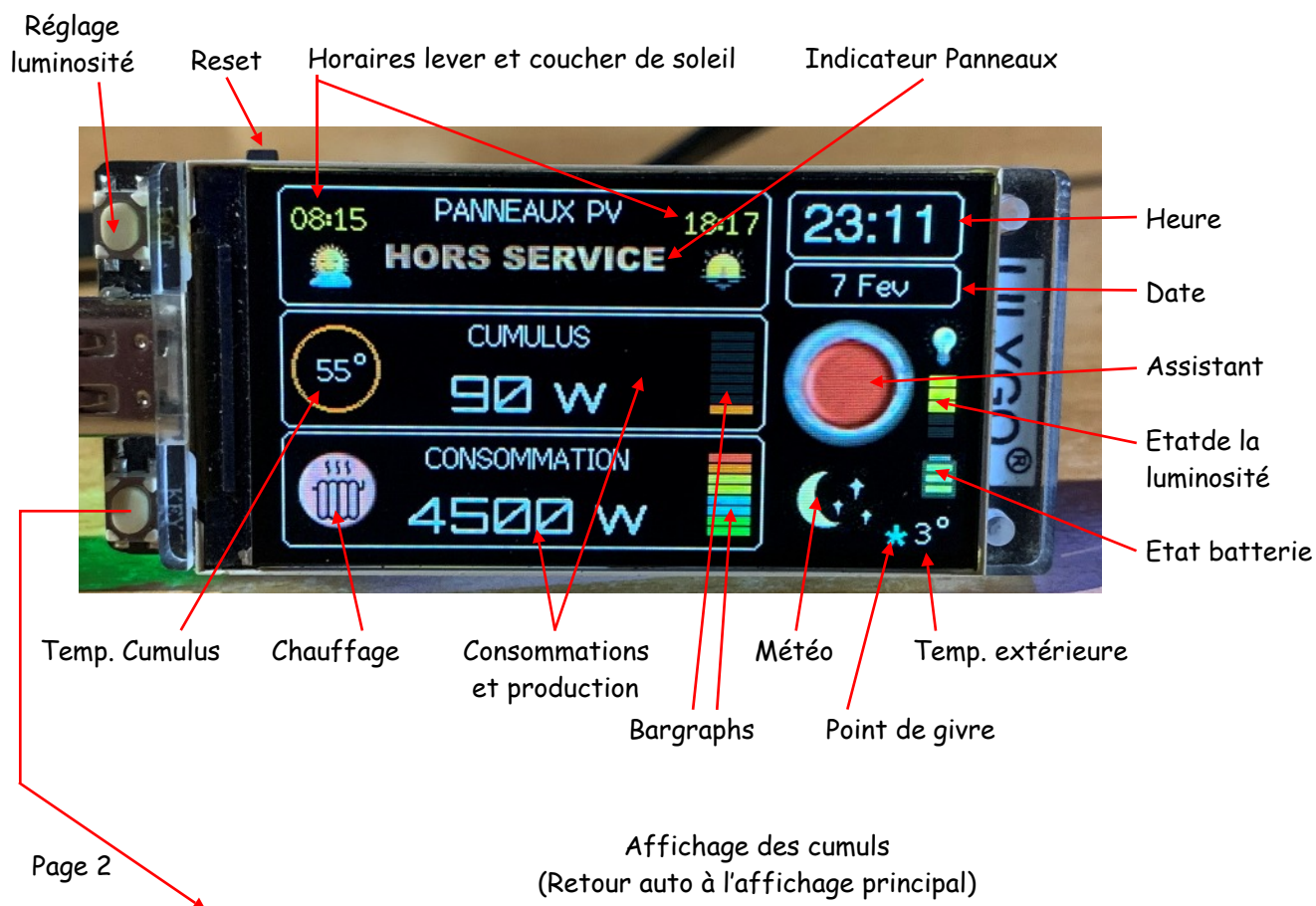
Liens

- Site et Forum Ard-Tek <https://ard-tek.com>
- Achat du T-Display https://s.click.aliexpress.com/e/_DEdEzkF
- Github du programme <https://github.com/JJHontebeyrie/Companion>



Les affichages du Companion

Voici les données affichées sur le companion (simulation, ce ne sont pas des données réelles). Il est constitué de 2 pages écran, un bouton permet d'afficher la page des cumuls, le retour à la page principale se faisant automatiquement après quelques secondes.





Préparation à l'installation

La programmation du module est réalisée sous environnement Arduino que vous obtiendrez sur le site de l'organisation <https://www.arduino.cc/>

Allez sur l'onglet Software et télécharger l'installateur correspondant à votre ordinateur.

Downloads



Arduino IDE 2.0.3

The new major release of the Arduino IDE is faster and even more powerful! In addition to a more modern editor and a more responsive interface it features autocompletion, code navigation, and even a live debugger.

For more details, please refer to the [Arduino IDE 2.0 documentation](#).

Nightly builds with the latest bugfixes are available through the section below.

SOURCE CODE

The Arduino IDE 2.0 is open source and its source code is hosted on [GitHub](#).

DOWNLOAD OPTIONS

Windows Win 10 and newer, 64 bits
Windows MSI installer
Windows ZIP file

Linux AppImage 64 bits (X86-64)
Linux ZIP file 64 bits (X86-64)

macOS Intel, 10.14: "Mojave" or newer, 64 bits
macOS Apple Silicon, 11: "Big Sur" or newer, 64 bits

[Release Notes](#)

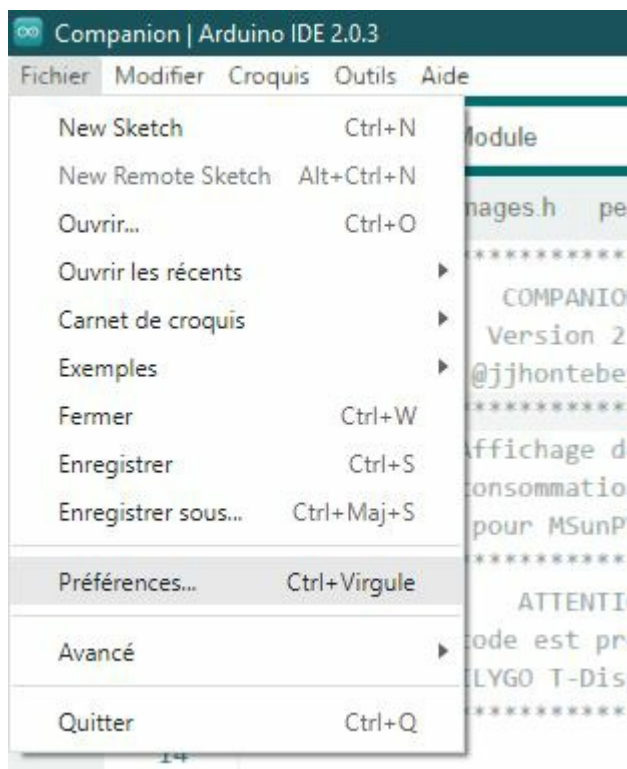
Lorsque vous lancez Arduino IDE, la barre de menu latérale vous permet d'ajouter des cartes de développement comme le T-Display S3. C'est une carte récente et elle ne figure pas dans la liste des cartes listées avec le programme de base. Il va donc être nécessaire d'installer les drivers pour cette carte manuellement.

Onglet Cartes ->



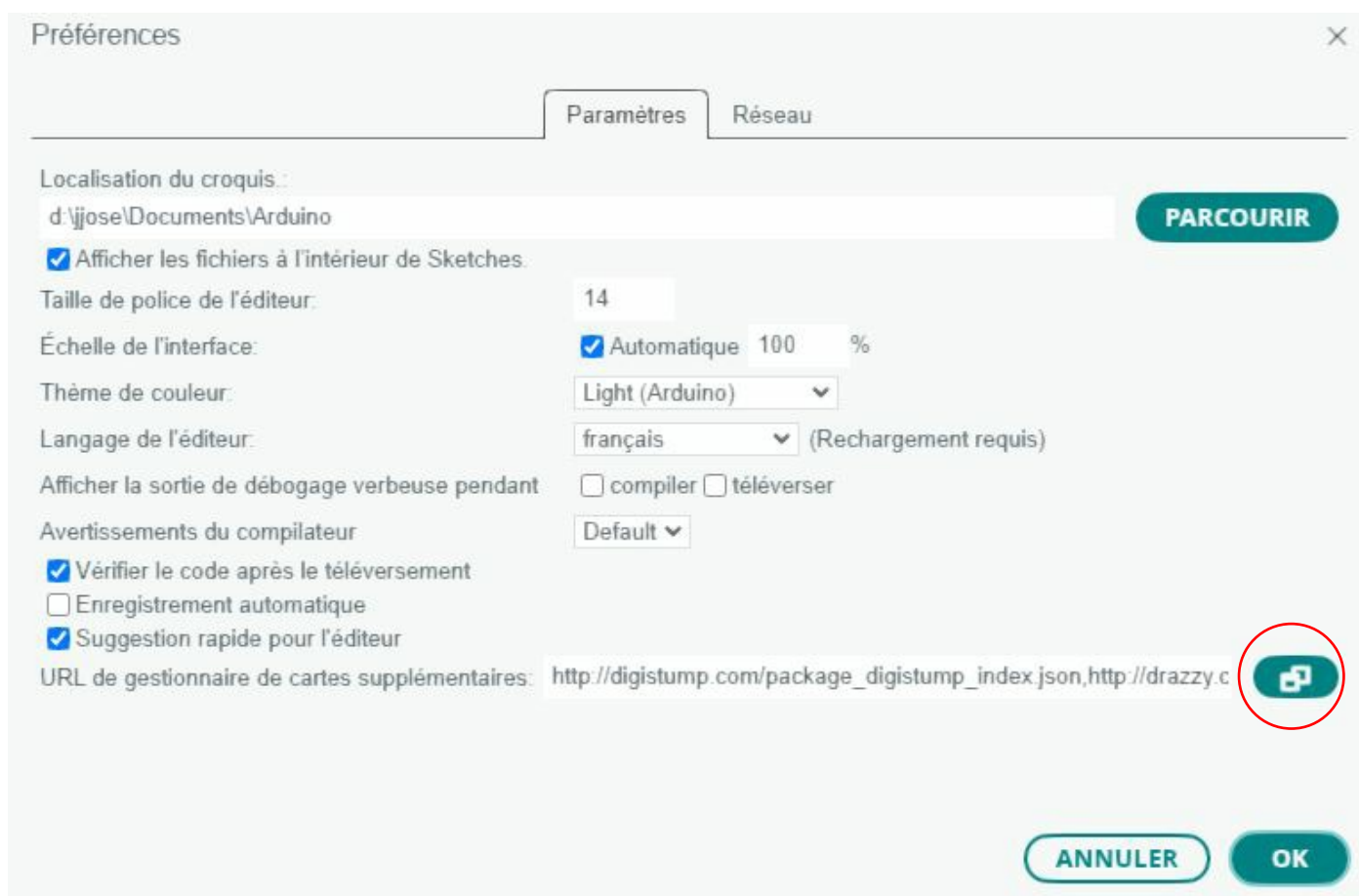


Installation des paramètres de la carte S3



Allez sur le sous menu 'Préférences' dans le menu 'Fichier' pour afficher les réglages de base Arduino.

Dans la fenêtre qui apparaît, cliquez sur le bouton de URL de gestionnaire de cartes supplémentaires

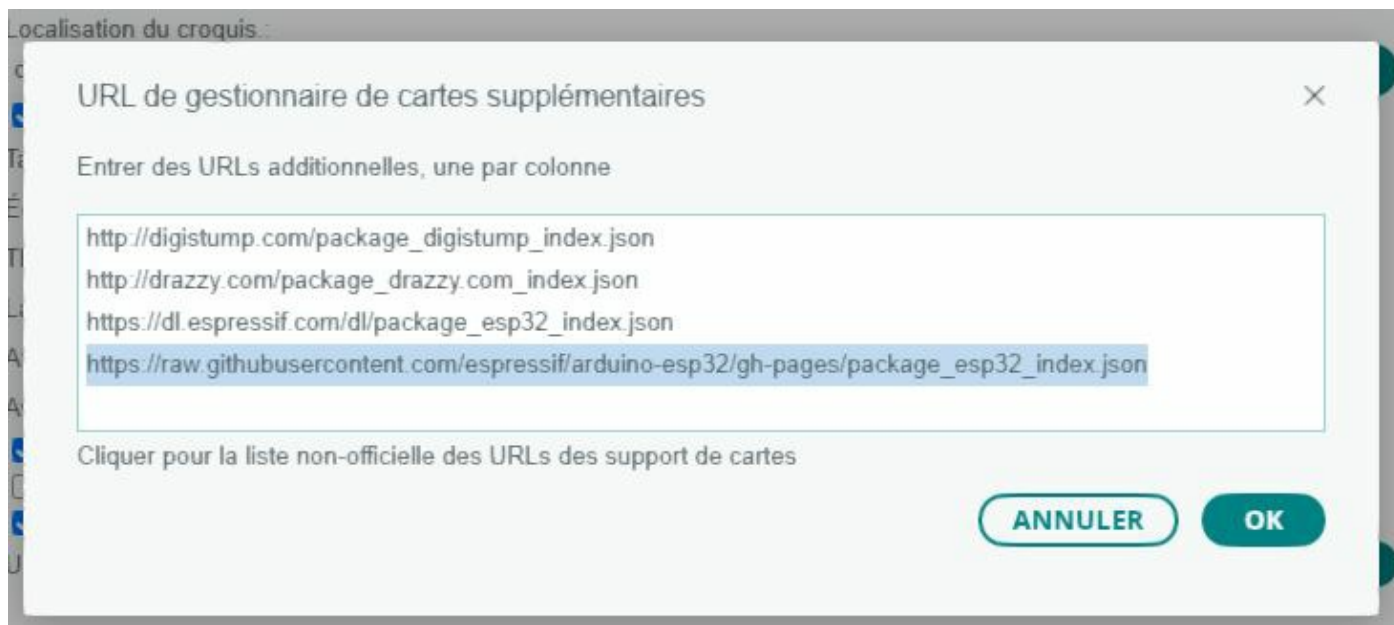




La ligne permettant d'ajouter les caractéristiques de cette carte est :

https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_index.json

Vous la copiez et la collez dans la fenêtre



Puis vous validez, c'est tout ce qu'il y a à faire pour que votre carte soit reconnue.

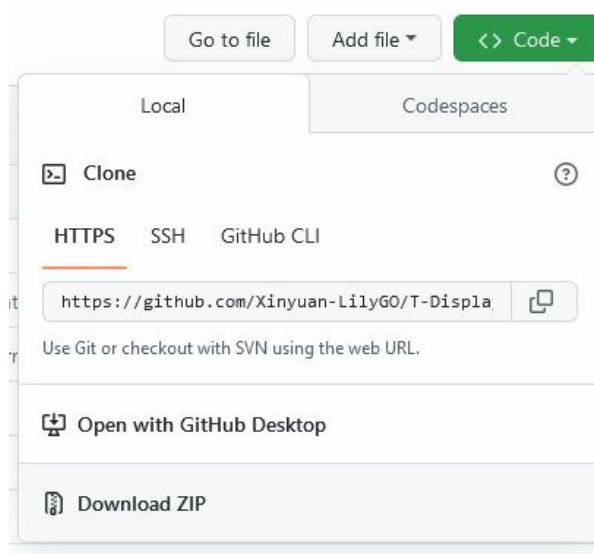
Installation des bibliothèques pour la carte S3

La Lilygo T-Display S3 étant très récente, la bibliothèque pour l'utiliser n'est pas encore incluse dans Arduino IDE. L'installation va devoir être faite à la main. Rendez-vous dans le repository Github

<https://github.com/Xinyuan-LilyGO/T-Display-S3>

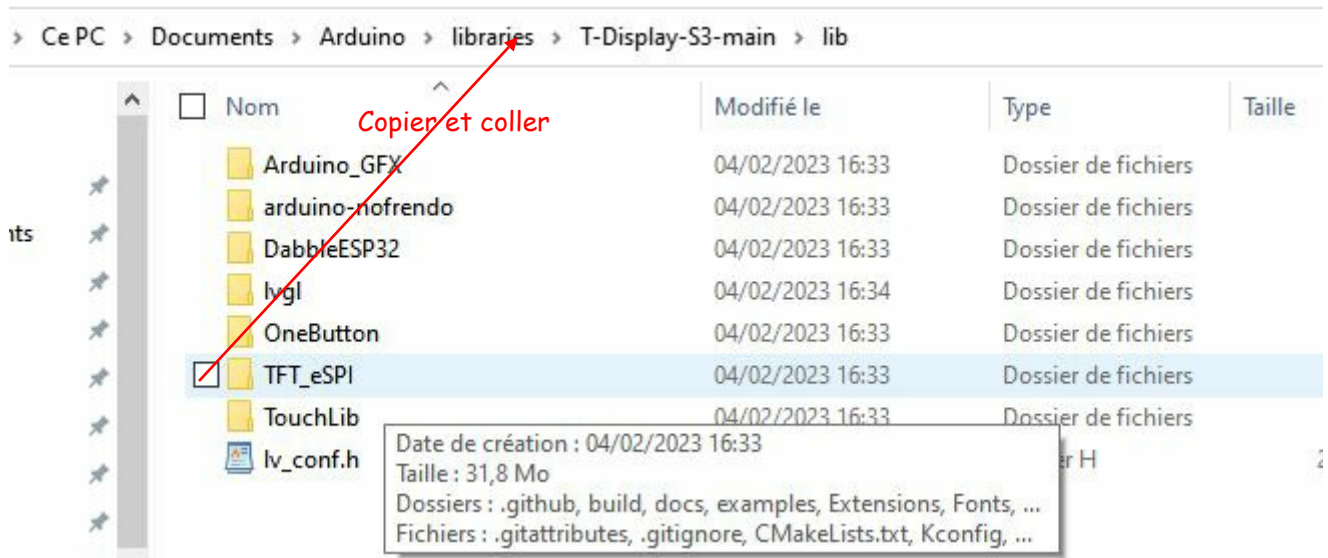
Cliquez sur le bouton vert « Code » et downloadez le zip. Vous le décompactez dans votre ordinateur.

Quand Arduino a été installé, vous avez un dossier 'Arduino' dans 'Mes Documents'. Copiez le dossier décompacté T-Display-S3-Main dans Arduino/libraries





Ensuite, une petite manip est à faire. Vous avez copié le dossier T-Display-S3-Main dans libraries. Il contient un sous dossier 'lib', ouvrez le et copiez le dossier TFT_eSPI que vous venez coller dans libraries



Une fois toutes ces manipulations réalisées, lorsque vous lancez Arduino, résistez à la tentation d'appuyer sur 'Install All' dans la fenêtre d'informations qui va apparaître, faites 'Plus Tard'.

En effet comme cette bibliothèque TFT_eSPI est nouvelle, il voudra la remplacer, mais il le fera par une version antérieure qui est non compatible avec le S3.





Installation du programme Companion

Comme vous avez chargé le gihub pour la S3, vous allez maintenant charger le programme gérant le companion pour qu'il fonctionne avec le Msun'PV. Allez sur l'adresse :

<https://github.com/JJHontebeyrie/Companion>

Et téléchargez le zip que vous décompactez sur votre ordinateur. Il faudra le copier / coller dans le dossier Documents/Arduino.

Si vous ouvrez ce dossier, vous devriez avoir tous ces fichiers. Pour lancer la programmation, il suffira de double cliquer sur Companion.ino

> Ce PC > Documents > Arduino > Companion				
<input type="checkbox"/> Nom	Modifié le	Type	Taille	
Companion.ino	08/02/2023 12:59	Fichier INO	30 Ko	
images.h	08/02/2023 12:59	Fichier H	335 Ko	
logo.h	08/02/2023 12:59	Fichier H	453 Ko	
meteo.h	08/02/2023 12:59	Fichier H	406 Ko	
NTP_Time.h	08/02/2023 12:59	Fichier H	11 Ko	
perso.h	08/02/2023 12:59	Fichier H	3 Ko	

Installation des bibliothèques pour Companion

Comme vous avez chargé le gihub pour la S3, vous allez maintenant charger le programme gérant le companion. Mais avant de faire ceci, il va falloir installer certaines bibliothèques spécifiques pour que le programme Companion fonctionne. Téléchargez ces 4 bibliothèques :

<https://github.com/PaulStoffregen/Time>

<https://github.com/JChristensen/Timezone>

https://github.com/Bodmer/JSON_Decoder

<https://github.com/Bodmer/OpenWeather>

Vous les installez comme vous avez procédé avec la bibliothèque du S3, les dossiers décompactés sont collés dans Documents/Arduino/libraries

Bon, maintenant allez prendre un café avec un cognac ou un doliprane suivant votre état ;-)

On va enfin pouvoir installer le programme, mais il faut vérifier certains paramètres auparavant.

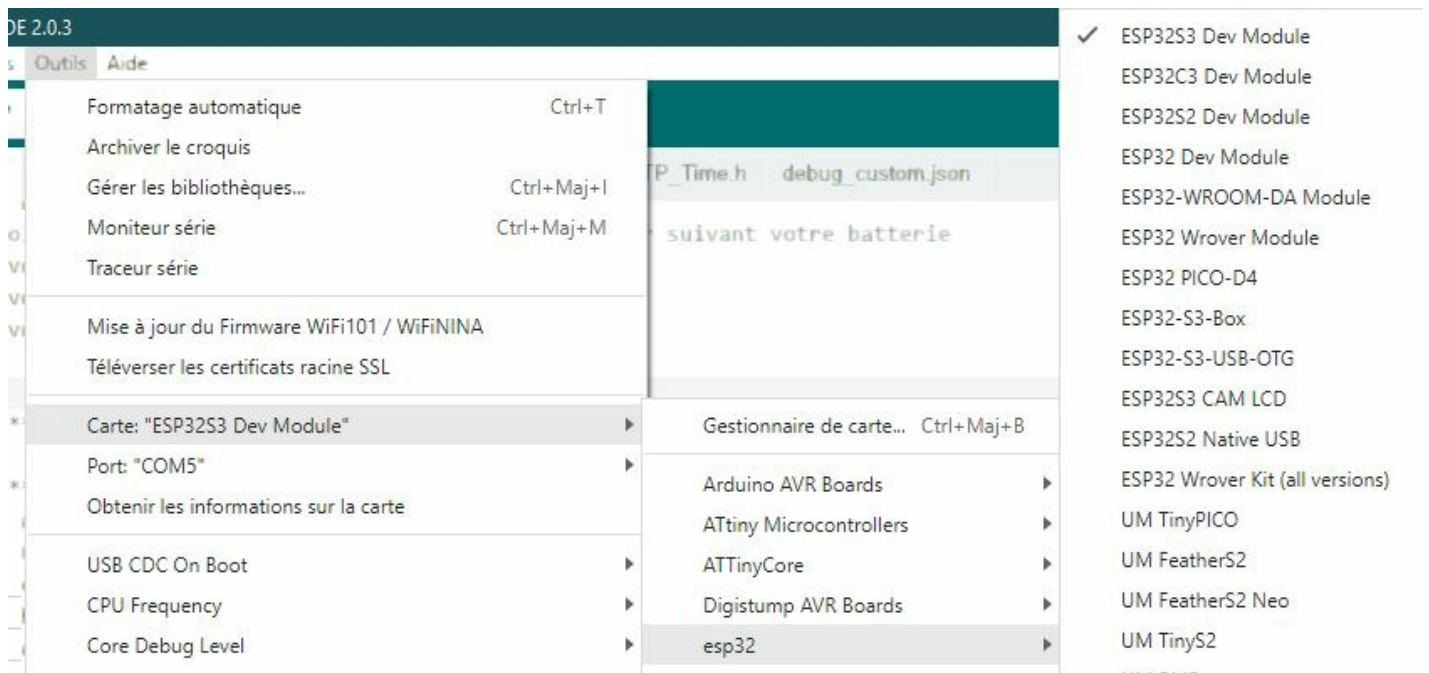


Reconnaissance par Arduino du T-Display S3

Ces réglages sont **hyper** importants pour que tout téléversement depuis Arduino se fasse sans problème sur ce module.

Vous branchez sur un port USB votre S3, attendez sa reconnaissance par l'ordinateur et allez dans l'onglet 'Outils', et choisissez Carte/esp32/ESP32S3 Dev Module

Sous la ligne Carte, choisissez son port de communication qui sera dans la liste.



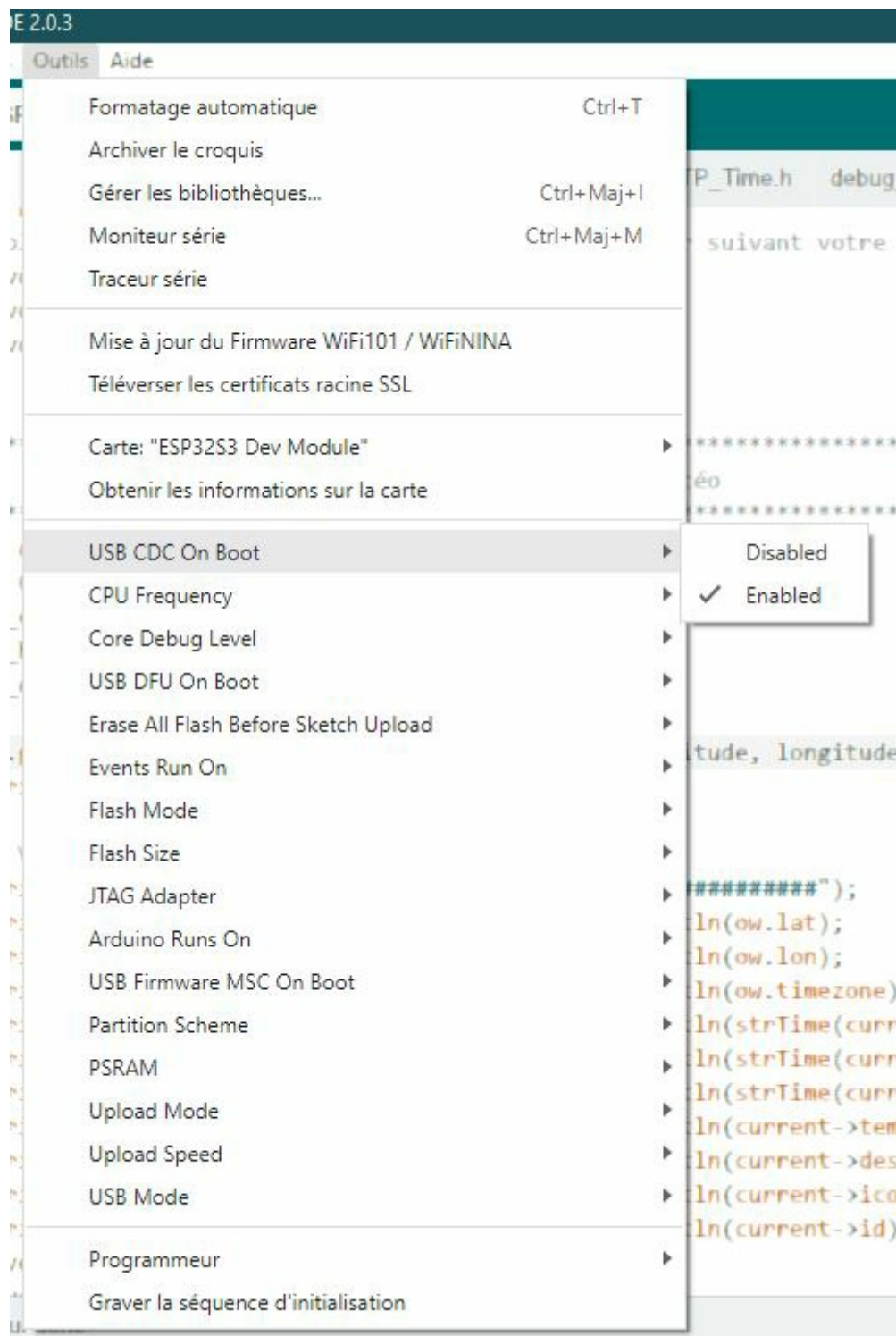
Cette manipulation n'est à faire qu'une seule fois, par la suite lorsque vous brancherez le module, il sera automatiquement reconnu, il faudra s'assurer que le port de communication est bien sélectionné.

Enfin, il faut procéder aux paramétrages de cette carte qu'il ne faudra faire aussi qu'une seule fois. Je vous explique cela dans la page suivante.



Réglages nécessaires pour le T-Display S3

Les réglages sont accessibles sous l'onglet 'Outils' et sont listés sous les choix de la carte faits précédemment. Voyez le tableau page suivante pour mettre les bonnes valeurs.





Réglages nécessaires pour le T-Display S3

Voici donc les valeurs à choisir :

- USB CDC on Boot : Enabled
- CPU Frequency : 240 MHz (wifi)
- Core Debug Level : Info
- USB DFU on Boot : Disabled
- Erase All Flash Before Sketch Upload : Disabled
- Events Run On : Core 1
- Flash Mode : QIO 80 MHz
- Flash Size : 16 MB (128 Mb)
- JTAG Adapter : Integrated USB JTAG
- Arduino Run On : Core 1
- USB Firmware MSC On Boot : Disabled
- Partition Scheme : Huge App (3 MB no OTA / 1 MB SPIFFS)
- PSRAM : OPI PSRAM
- Upload Mode : UART0 / Hardware CDC
- Upload Speed : 921600
- USB Mode : Hardware CDC and JTAG
- Programmeur : Esptool

Voilà, c'est fini. Il était bon le café ?

Bon, maintenant vous pouvez double cliquer sur Companion.ino pour le voir sous Arduino. Le fichier ino sera installé ainsi que tous les modules l'accompagnant, vous avez la liste en haut de l'écran. Cliquez sur 'perso.h' qui contient des valeurs personnelles pour votre installation à la maison.





Personnalisation de votre Companion

Certaines valeurs sont à saisir dans ce fichier pour un fonctionnement optimal depuis chez vous, en particulier, le nom de votre réseau WiFi et sa clef de décryptage, vous remplacez les * par vos valeurs. Il faudra aussi mettre l'adresse dans votre réseau du Msunpv, vous la trouverez sur son interface web.

```
Companion.ino  logo.h  images.h  perso.h  perso2.h  meteo.h  NTP_Time.h
1  //*****
2  //***** Données personnelles *****
3  //*****
4  // Mettez ici toutes vos valeurs personnelles pour le fonctionnement ***
5  // correct de l'afficheur externe Companion ***
6  //*****
7
8  // Vos codes accès au wifi et adresse de votre MSunPV dans votre réseau
9  // Remplacez les * par vos valeurs
10 const char* ssid = "*****";
11 const char* password = "*****";
12
13 // Adresse IP du serveur local (ici MSunPV). Il se peut suivant votre réseau
14 // que vous ayez besoin de remplacer toute la chaîne, respectez : entre les chiffres
15 char server[] = "192.168.1.**";
16
17 // Boitier horizontal prise à gauche, pour prise à droite mettez rotation = 1
18 int rotation = 3;
19
```

Il faudra également renseigner la latitude et la longitude de votre ville, vous pourrez trouver ces valeurs en faisant une recherche internet.

Et enfin une clé api pour OpenWeather <https://openweathermap.org/> où vous créez un compte gratuit qui vous permettra de générer une clé api que vous mettrez ici.

```
// localisation de votre ville et décalage horaire pour lever/coucher soleil
String latitude = "44.8378"; // 90.0000 to -90.0000 negative for Southern hemisphere
String longitude = "-0.594"; // 180.000 to -180.000 negative for West
#define TIME_OFFSET 1UL * 3600UL // UTC + 0 hour (2UL * 3600L pour UTC + 1, etc...)

// Données pour openweathermap.org ( API key et ville)
String api_key = "*****";
String units = "metric"; // ou "imperial"
String language = "fr";
```

Les autres paramètres à modifier dans ce fichier sont en rapport avec votre installation photovoltaïque et sur la façon dont vous voulez avoir les informations sur le Companion.

Sous Arduino, dans l'onglet Outils, cochez la ligne 'Moniteur Série', ceci vous permettra d'avoir des informations sur les valeurs récupérées par le Companion et de voir si les bonnes sondes sont sélectionnées.

Dans le fichier Companion.ino, ces valeurs sont indiquées par défaut, mais il est possible de les modifier pour une adaptation aux éléments ue vous avez connectés sur votre MsunPV.



Modification des valeurs des compteurs

Sur le moniteur série, le programme fait apparaître des données qui vont vous permettre de peaufiner vos réglages. Vous avez tout d'abord les données météo de votre ville qui sont mises à jour toutes les 15 minutes, puis la ligne compteurs vous montre les valeurs des 16 compteurs qui équipent le Msonpv et ensuite les valeurs des compteurs cumulés.

Repérez l'emplacement des valeurs compteurs et les positions des cumuls pour voir les valeurs que vous souhaitez afficher.

```
##### Données météo #####
Latitude      : 44.84
Longitude     : -0.59
Timezone      : Europe/Paris
Heure actuelle : 22:58
Lever soleil  : 08:13
Coucher soleil : 18:19
température   : 1.21
description   : ciel dégagé
icone         : 01n
ID            : 800

Initialisation de la connexion au serveur...
Connecté au serveur
Attente de la réponse serveur...
requete validée

Compteurs >> 5593,3;-9,8;3,7; 0;216,5;0,0;0,0;0,0; 0; 0; 0; 0; 0; 0; 0; 0;
Cumul 0 >> 52849,5
Cumul 1 >> -1846,0
Cumul 2 >> 10854,3
Cumul 3 >> 5585,0
Cumul 4 >> 0,0
Cumul 5 >> 0,0
Cumul 6 >> 0,0
Cumul 7 >> 0,0
```

Position 0 (valeur 5593.3)

Position 1 (valeur -9.8)

Position 2 (valeur 3.7)

Au niveau du programme dans le fichier Companion.ino, vous pourrez retrouver ces valeurs à partir de la ligne 419 et qui sont par exemple ici en position 0 (MsgSplit[0]) pour la consommation du domicile, en position 1 pour la production des panneaux et en position 2 pour la consommation du cumulus. Vous pouvez modifier le choix des positions pour qu'ils s'accordent à ceux vus dans le moniteur série. Si par exemple votre cumulus est en position 4 (attention on commence par position 0) vous aurez dans le programme CU = MsgSplit[4];

```
419 // Récupération des données à afficher
420 CO = MsgSplit[0]; // Consommation
421 PV = MsgSplit[1]; // Panneaux PV
422 PV = String(abs(PV.toInt())); // (avec prod en + ou en -)
423 CU = MsgSplit[2]; // Cumulus
424 TEMPCU = MsgSplit[5]; // Sonde température cumulus
425 TEMPCU = TEMPCU.toInt();
```




Modification des valeurs des cumuls

En ce qui concerne les cumuls, ceux-ci sont traités au niveau de la ligne 419 et suivantes. Et vous suivez le même modus operandi.

```
Compteurs >> 5593,3;-9,8;3,7; 0;216,5;0,0;0,0;0,0; 0; 0; 0; 0; 0; 0; 0; 0;
Cumul 0 >> 52849,5
Cumul 1 >> -1846,0
Cumul 2 >> 10854,3
Cumul 3 >> 5585,0
Cumul 4 >> 0,0
Cumul 5 >> 0,0
Cumul 6 >> 0,0
Cumul 7 >> 0,0
```

Position 0 (valeur 52849.5)

Position 2 (valeur 10854.3)

```
456 //*****
457 // Suivant les modifications que vous avez apporté au MSunPV
458 // il est possible que les cumuls souhaités ne soient pas
459 // ceux affichés. Vérifiez sur le moniteur série le bon index
460 CUMCO = MsgSplit2[0]; // Cumul Conso
461 CUMINJ = MsgSplit2[1]; // Cumul Injection
462 CUMPV = MsgSplit2[2]; // Cumul Panneaux
463 CUMBAL = MsgSplit2[3]; // Cumul Ballon cumulus
464 //*****
```

Une fois toutes ces modifications réalisées, sauvez le fichier, et en cliquant sur l'icône avec la flèche, le programme se chargera sur votre module normalement sans difficulté, sinon, on en reparle sur le forum Ard-Tek.



Rendre le module autonome

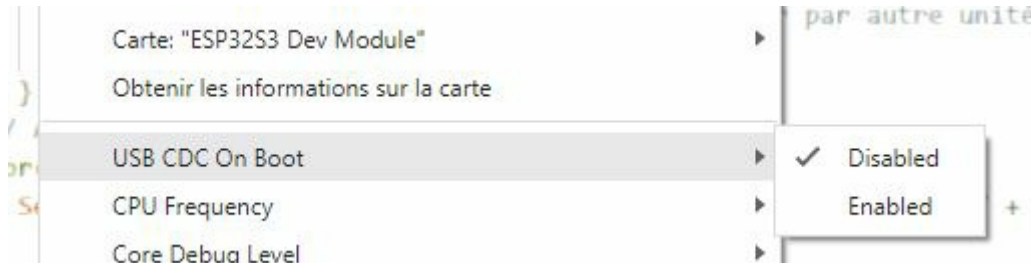
Le problème, c'est que votre module ne peut fonctionner que connecté à votre ordinateur avec Arduino lancé. Comme ce n'est pas très pratique de mettre un ordinateur dans la cuisine par exemple, avec un bout de fil et un module d'affichage qui pendouille, il va falloir le rendre autonome.

Il y a bien sur une astuce que je vous montre dans la prochaine page.



One more Thing

Dans les réglages de la carte de développement S3, le réglage USB CDC on Boot Enabled permet de charger le programme sur le module mais aussi de voir la fenêtre Moniteur Série pour le débogage. Si vous modifiez ce réglage en le mettant Disabled :



Il suffira de compiler et reverser encore une fois le programme dans le module pour que celui-ci fonctionne lorsqu'il est connecté sur l'ordinateur, mais également de manière complètement autonome. A noter que dans ce cas, le Moniteur Série ne sera plus accessible, mais il vous suffit de le remettre sur Enabled lors de vérification du programme, pensez à le remettre Disabled pour un fonctionnement autonome.

Conclusion

Je crois vous avoir tout dit pour réaliser cet afficheur, à l'utilisation, il s'avère très pratique et discret. Le programme fourni est extrêmement documenté et vous pouvez bien sur y apporter toutes les modifications que vous souhaitez pour le rendre plus pratique dans votre utilisation personnelle.

Et si vous voulez, on en reparle sur le forum Ard-Tek.

Développer des accessoires de ce type me demande pas mal d'efforts et de temps, et si vous souhaitez m'offrir un café en remerciement, je prend !

Cliquez sur la tasse



Et merci de m'avoir lu dans tous les cas !

Vous pouvez aussi vous rendre sur ma chaîne par curiosité, et pourquoi pas vous abonner et liker les épisodes ! Cliquez donc sur l'image !

