

# WeTimer Installation logicielle et Programmation de la carte.

Pour être utilisée en tant que minuterie pour planeur de vol libre, la carte microcontrôleur Wemos D1 mini doit être chargée avec le programme WeTimer depuis un ordinateur (Windows, macOS ou Linux) à l'aide du logiciel Arduino IDE.



(Les copies d'écran de cette notice ont été réalisées sous Linux).

#### Installation du driver Windows:

**AVANT DE LA CONNECTER A VOTRE ORDINATEUR**, sous Windows, il est nécessaire d'installer un driver pour la prise en compte de la carte Wemos D1 mini. Téléchargez le driver ici : https://github.com/wemos/ch340\_driver/raw/master/CH341SER\_WIN\_3.5.ZIP Pour installer, décompressez le fichier ZIP et lancez le programme setup.exe. Sous Linux, la carte est reconnue directement et aucun driver supplémentaire n'est nécessaire.

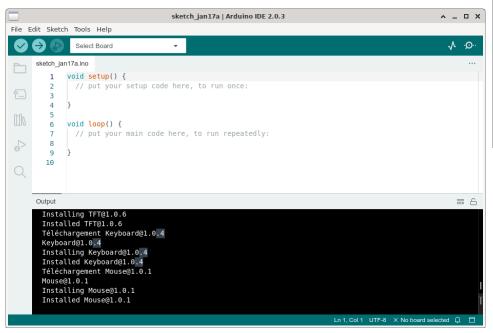
#### **Installation du logiciel Arduino IDE:**

Le logiciel Arduino IDE est un logiciel open source pour la programmation de microcontrôleurs en langage C et C++. C'est ce logiciel qui est utilisé pour le développement du programme de la minuterie WeTimer.

Pour commencer, si vous ne l'avez pas déjà, téléchargez gratuitement la dernière version d'Arduino IDE ici : https://www.arduino.cc/en/software

Installez le sur votre ordinateur comme n'importe quel autre logiciel. Au premier lancement du logiciel, vous devez accepter les termes de service en cliquant sur « Agree ».

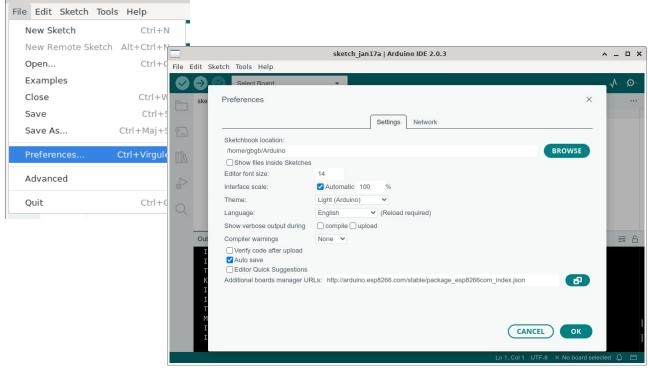
Vous obtiendrez une fenêtre ressemblant à ceci :



Arduino IDE ^ 🗆 X Terms of Service The Arduino software is provided to you 'as is" and we make no express or implied warranties whatsoever with respect to its functionality, operability, or use, including, without limitation, any implied warranties of merchantability, fitness for a particular purpose, or infringement. We expressly disclaim any liability whatsoever for any direct, indirect, consequential, incidental or special damages, including, without limitation, lost revenues, lost profits, losses resulting from business interruption or loss of data, regardless of the form of action or legal theory under which the liability may be asserted, even if advised of the Disagree

### Prise en charge des cartes ESP8266 dans l'IDE Arduino

Cliquez sur le menu « File » puis « Preferences » :



Dans la zone de saisie « Additional boards manager URLs : » ajoutez l'URL suivante :

http://arduino.esp8266.com/stable/package\_esp8266com\_index.json

Note : On peut entrer plusieurs URLs de cartes en les séparants par des virgules.

Cliquez sur le menu « Tools », « Boards » puis « Boards manager

8

10

esp8266 by ESP8266

Generic ESP8266 Module, Generic ESP8285 Module,

Lifely Agrumino Lemon v4, ESPDuino (ESP-13 Module), Adafruit Feather

HUZZAH ESP8266, WiFi Kit 8, Invent One, XinaBox CW01, ESPresso Lite 1.0,

Community Boards included in this

package:

Tools Help Auto Format Ctrl+T Archive Sketch Manage Libraries... Ctrl+Maj+I Serial Monitor Ctrl+Maj+M Serial Plotter WiFi101 / WiFiNINA Firmware Updater Upload SSL Root Certificates Boards Manager.. Board File Edit Sketch Tools Help Arduino AVR Boards Get Board Info Select Board Burn Bootloader BOARDS MANAGER sketch\_j ESP8266 딥 Entrez « ESP8266 » dans la zone de recherche puis cliquez sur le

ESP8266 Community ».

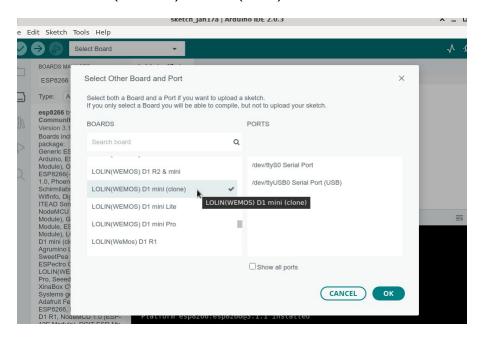
bouton « Install » au bas de la zone de texte « esp8266 by

Module), LOLIN(WEMOS) D1 mini (clone), Lifely Agrumino Lemon v4, SweetPea ESP-

3.1.0 V INSTALL

More info

Après avoir cliqué sur « Select Board », faire défiler la liste des cartes qui s'affiche, puis sélectionnez la carte « LOLIN(WEMOS) D1 mini (clone) » :



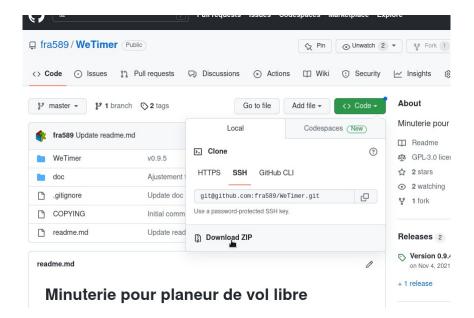
Sélectionnez également le port sur lequel vous avez connecté la carte, puis, cliquez sur « OK ».

Dans le menu « Tools », cliquez sur « E<rase Flash », puis sélectionnez « All Flash Content » pour une réinitialisation correcte de la carte lors de la programmation de WeTimer.

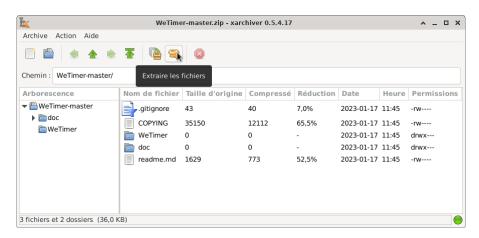


## **Programme WeTimer:**

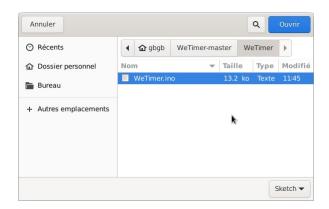
Téléchargez le programme WeTimer depuis Github ici : https://github.com/fra589/WeTimer Cliquez sur le bouton vert « <> Code ▼ », puis sur « Download ZIP ».



Décompressez le fichier téléchargé à l'endroit de votre choix sur votre ordinateur en conservant l'organisation et le nom des dossiers à l'intérieur du ZIP.



Dans l'IDE Arduino, cliquez sur le menu « File » puis « Open » et sélectionnez le fichier « Wetimer.ino » dans le dossier décompressé « WeTimer-Master/WeTimer/ ».



Une nouvelle fenêtre contenant les sources du programme de la minuterie s'ouvre :

```
WeTimer | Arduino IDE 2.0.3
                                                                                                                                       ^ _ D X
File Edit Sketch Tools Help
       WeTimer.ino WeTimer.h handleConfig.cpp handleConfig.h handleHttp.cpp handleHttp.h handleRoot.cpp handleRoot.h handleSetParams · · ·
           /* Copyright (C) 2021 Gauthier Brière (gauthier.briere "at" gmail.com)
                /* This file: WeTimer.ino is part of WeTimer
                /* WeTimer is free software: you can redistribute it and/or modify it
                /* under the terms of the GNU General Public License as published by
/* the Free Software Foundation, either version 3 of the License, or
÷>
                /* (at your option) any later version.
                /* WeTimer is distributed in the hope that it will be useful, but
                /* WITHOUT ANY WARRANTY; without even the implied warranty of /* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
                /* GNU General Public License for more details
          16
                /* You should have received a copy of the GNU General Public License
/* along with this program. If not, see <a href="http://www.gnu.org/licenses/">http://www.gnu.org/licenses/>.</a>
          18
          21
               #include "WeTimer.h"
          23
          24
25
                // Variables globales
                char cli_ssid[32] = DEFAULT_CLI_SSID;
char cli_pwd[63] = DEFAULT_CLI_PWD;
```

Cliquez sur le bouton « → » (Upload) pour compiler et envoyer le programme dans la carte.



A la fin du traitement, lorsque vous voyez le message :

```
Output

Writing at 0x0002c000... (80 %)
Writing at 0x00030000... (86 %)
Writing at 0x00034000... (93 %)
Writing at 0x00038000... (100 %)
Wrote 335968 bytes (238399 compressed) at 0x000000000 in 5.4 seconds (effective 502.1 kbit/s)...
Hash of data verified.

Leaving...
Hard resetting via RTS pin...
```

Cela indique que le téléchargement est termine sur la carte microcontrôleur et que celle ci à été redémarrée.

#### La minuterie est maintenant fonctionnelle!