

Personnaliser un firmware Marlin 2.1.2.1 en Français, avec ou sans BL Touch

Télécharger les fichiers source ici: <https://marlinfw.org/meta/download/>

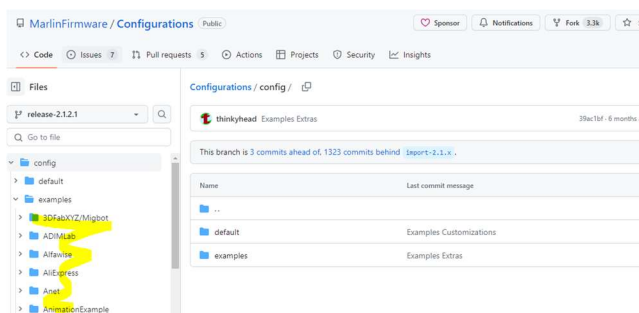
Description	Version	Download	Configurations
Current Marlin Release <i>Supports AVR and ARM Arduino and PlatformIO</i>	2.1.2.1	2.1.2.1.zip	View / Download

La base de la configuration adaptée à votre imprimante, à télécharger , et à décompresser dans la racine de votre PC

Ce PC > Disque local (C:)			
	Nom	Modifié le	Type
e	Marlin-2.1.x	29/12/2023 15:03	Dossier de fichiers

Puis:

Description	Version	Download	Configurations
Current Marlin Release <i>Supports AVR and ARM Arduino and PlatformIO</i>	2.1.2.1	2.1.2.1.zip	View / Download

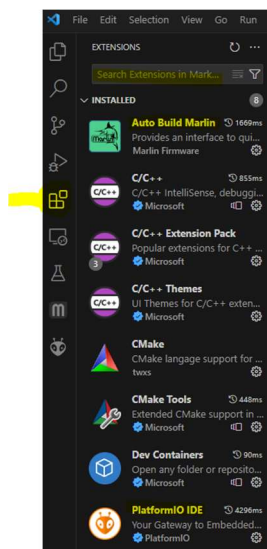


Téléchargez les fichiers nécessaires à votre imprimante : sélectionnez la marque, le modèle, puis le type de carte mère dont vous disposez

Ce PC > Disque local (C:) > Marlin-2.1.x > Marlin

Nom	Modifié le
lib	29/12/2023 15:02
src	29/12/2023 15:02
_Bootscreen	29/12/2023 15:01
_Statusscreen	29/12/2023 15:01
config	29/12/2023 15:02
Configuration	29/12/2023 15:55
Configuration_adv	29/12/2023 15:38
Makefile	29/12/2023 15:02
Marlin.ino	29/12/2023 15:02
Version	29/12/2023 15:02

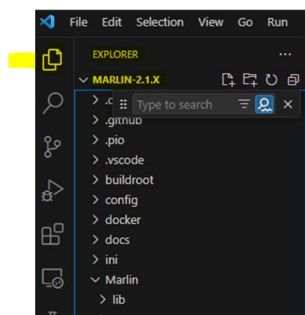
Téléchargez les fichiers, et copier (écraser) les ici, dans
Marlin-2.1.x/Marlin/



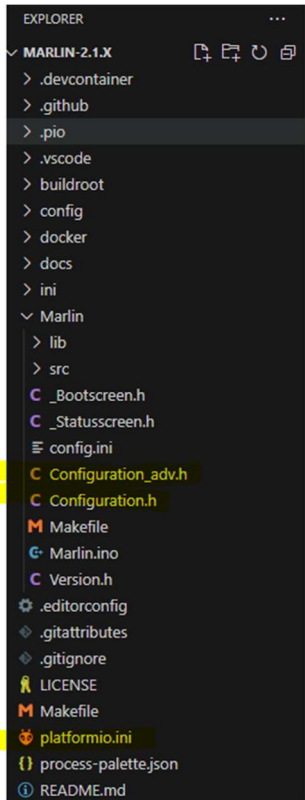
Vous devez disposer de VSCode
<https://code.visualstudio.com/>

Lorsque le logiciel est installé,

sur le menu de gauche cliquez sur “Extension” et recherchez et installez, les extensions “Plateformio IDE” et “AUTO Build Marlin”



VS Code installé, sélectionnez votre répertoire “MARLIN-2.1.X”
présent dans la racine (pas le dossier Marlin qui est dedans...mais
bien celui qui est dans la racine)



Les 3 fichiers que vous devrez modifier sont :

Configuration.h

Configuration_adv.h

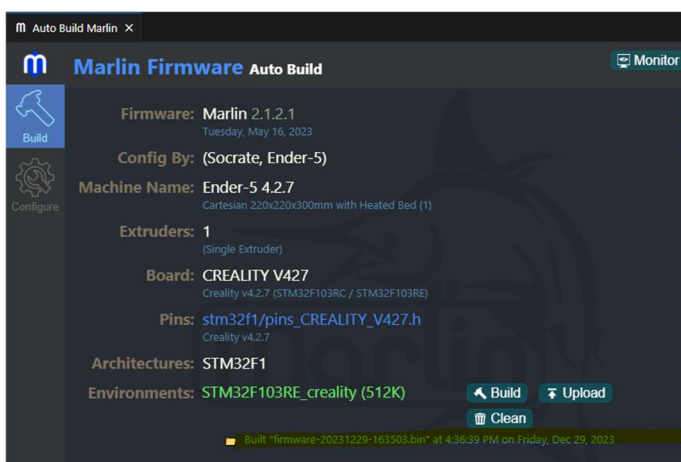
platformio.ini

(facultatif, si vous utilisez l'auto build)

Dans les pages suivantes, en fonction des fichiers, vous recherchez la première ligne, pour la remplacer la seconde.



Après avoir modifié les 3 fichiers, vous sauvegardez, et cliquez sur Build



Enfin, pour retrouver votre fichier, cliquez sur "Built 'firmware...' at..." vous trouverez le fichier .bin à mettre dans la racine de votre carte SD. Insérez la carte dans votre imprimante, et le firmware se met à jour.

 firmware-20231229-163503.bin

La configuration testée ici est une Ender 5 ou 3, avec carte mère 4.2.7, 4.2.2 ou 1.1.x

le Firmware Marlin 2.1.2.1

Fichier platformio.ini

4.2.7 ou 4.2.2 :

```
default_envs = STM32F103RE_creality
```

(Pour les cartes 1.1.x, `default_envs melzi_optiboot_optimized`)

Fichier Configuration.h

--

Pour dire qui apporte les modifications au firmware :

Remplacez :

```
#define STRING_CONFIG_H_AUTHOR "(Dust, Ender-5)"
```

Par :

```
#define STRING_CONFIG_H_AUTHOR "(Votre pseudo, Ender-5)"
```

--

Pour personnaliser le nom de votre imprimante qui s'affiche sur l'écran

Remplacez :

```
#define CUSTOM_MACHINE_NAME "Ender-5 4.2.7"
```

Par :

```
#define CUSTOM_MACHINE_NAME "Le nom que vous souhaitez"
```

--

Pour basculer la langue en Français (**A noter** : Vous pouvez configurer une seconde langue, dans la partie Configuration_adv.h) :

Remplacez :

```
#define LCD_LANGUAGE en
```

Par :

```
#define LCD_LANGUAGE fr
```

--

Pour régler les Drivers, remplacez :

```
//#define X_DRIVER_TYPE  A4988  
//#define Y_DRIVER_TYPE  A4988  
//#define Z_DRIVER_TYPE  A4988  
//#define X2_DRIVER_TYPE A4988  
//#define Y2_DRIVER_TYPE A4988  
//#define Z2_DRIVER_TYPE A4988  
//#define Z3_DRIVER_TYPE A4988  
//#define Z4_DRIVER_TYPE A4988  
//#define I_DRIVER_TYPE  A4988  
//#define J_DRIVER_TYPE  A4988  
//#define K_DRIVER_TYPE  A4988  
//#define U_DRIVER_TYPE  A4988  
//#define V_DRIVER_TYPE  A4988  
//#define W_DRIVER_TYPE  A4988  
//#define E_DRIVER_TYPE  A4988
```

Pour une carte mere 4.2.7, par :

```
#define X_DRIVER_TYPE  TMC2209_STANDALONE  
#define Y_DRIVER_TYPE  TMC2209_STANDALONE  
#define Z_DRIVER_TYPE  TMC2209_STANDALONE  
//#define X2_DRIVER_TYPE A4988  
//#define Y2_DRIVER_TYPE A4988  
//#define Z2_DRIVER_TYPE A4988  
//#define Z3_DRIVER_TYPE A4988  
//#define Z4_DRIVER_TYPE A4988  
//#define I_DRIVER_TYPE  A4988  
//#define J_DRIVER_TYPE  A4988  
//#define K_DRIVER_TYPE  A4988  
//#define U_DRIVER_TYPE  A4988  
//#define V_DRIVER_TYPE  A4988  
//#define W_DRIVER_TYPE  A4988  
#define E0_DRIVER_TYPE TMC2209_STANDALONE
```

Pour une carte mère 4.2.2 (ou 1.1.4 silent driver et 1.1.15) :

```
#define X_DRIVER_TYPE  TMC2208_STANDALONE  
#define Y_DRIVER_TYPE  TMC2208_STANDALONE  
#define Z_DRIVER_TYPE  TMC2208_STANDALONE  
//#define X2_DRIVER_TYPE A4988  
//#define Y2_DRIVER_TYPE A4988  
//#define Z2_DRIVER_TYPE A4988  
//#define Z3_DRIVER_TYPE A4988  
//#define Z4_DRIVER_TYPE A4988  
//#define I_DRIVER_TYPE  A4988  
//#define J_DRIVER_TYPE  A4988  
//#define K_DRIVER_TYPE  A4988  
//#define U_DRIVER_TYPE  A4988  
//#define V_DRIVER_TYPE  A4988
```

```
//#define W_DRIVER_TYPE A4988
#define E0_DRIVER_TYPE TMC2208_STANDALONE
```

--

Pour régler la taille du plateau, par exemple si vous souhaitez limiter la zone d'impression :

Si vous utilisez une Ender 3 (laissez 235 pour en Ender 5) , remplacez:

```
// The size of the printable area
#define X_BED_SIZE 235
#define Y_BED_SIZE 235
```

Par

```
#define X_BED_SIZE 220
#define Y_BED_SIZE 220
```

Et si vous utilisez des pinces sur votre Ender 3 ou 5, prévoir des marges...

Remplacez :

```
#define PROBING_MARGIN 10
```

par :

```
#define PROBING_MARGIN 15
```

--

Paramétrez la vitesse des moteurs.

Les valeurs sont Ok pour une Ender 3, modifiez si vous avez une Ender 5

Remplacez :

```
#define DEFAULT_AXIS_STEPS_PER_UNIT { 80, 80, 400, 93 }
```

Par :

```
#define DEFAULT_AXIS_STEPS_PER_UNIT { 80, 80, 800, 93 }
```

A noter, si vous connaissez les valeurs propres à votre imprimante, vous pouvez les intégrer en mettant la config de votre imprimante.

par exemple:

```
#define DEFAULT_AXIS_STEPS_PER_UNIT { 81.4, 81.2, 405.3, 97.5 }
```

81.4 correspond à la valeur X pas/mm (ou steps/mm)

81.2 correspond à la valeur Y pas/mm (ou steps/mm)

405.3 correspond à la valeur Z pas/mm (ou steps/mm)

97.5 correspond à la valeur E (extruder) pas/mm (ou steps/mm)

--

(On s'arrête ici, si vous n'avez pas de BL Touch / CR Touch)

Activation du BL Touch :

```
//#define BLTOUCH
```

```
#define BLTOUCH
```

```
--
```

Pour ajuster le positionnement de votre BL / CR Touch vis-à-vis de votre tête d'impression, remplacez:

```
#define NOZZLE_TO_PROBE_OFFSET { 10, 10, 0 }
```

Par:

```
#define NOZZLE_TO_PROBE_OFFSET { -42, -7, 0 }
```

Avec dans l' exemple: -42 = distance de 4.2cm entre la tête d'impression et la pointe du BL Touch

Et -7 = le BL touch est 0.7 cm plus avancé que la tête d'impression

```
--
```

Activation du nivellement du lit, si le BL Tous est sélectionné

```
//#define AUTO_BED_LEVELING_BILINEAR
```

```
#define AUTO_BED_LEVELING_BILINEAR
```

```
--
```

```
//#define Z_SAFE_HOMING
```

```
#define Z_SAFE_HOMING
```

```
--
```

Ajustement du nombre de points de test du niveau du lit : 3 = 9 points testés, suffisant pour les petits plateaux, 5 = 25 points testés nécessaire pour les grands plateaux, ou si vous souhaitez plus de précision (un peu plus long tout de même)

```
#define GRID_MAX_POINTS_X 3
```

```
#define GRID_MAX_POINTS_X 5
```

```
--
```

Pour définir si le BL Touch s'occupe d'arrêter l'axe Z, ou si on laisse faire l'interrupteur. Ici, j'ai pris le parti de dire que si le BL Touch était activé, c'est lui qui fera le job, s'il ne l'est pas, cela sera l'interrupteur.

Remplacez:

```
#define MIN_SOFTWARE_ENDSTOP_Z
```

par

```
// #define MIN_SOFTWARE_ENDSTOP_Z
```

```
--
```

Pour les cartes mère 4.2.x qui utilisent un port BL Touch dédié (Ne rien toucher si vous avez une ancienne carte mère 1.1.x), remplacez :

```
//#define USE_PROBE_FOR_Z_HOMING
```

par:

```
#define USE_PROBE_FOR_Z_HOMING
```

et remplacez:

```
#define Z_MIN_PROBE_USES_Z_MIN_ENDSTOP_PIN
```

par :

```
// #define Z_MIN_PROBE_USES_Z_MIN_ENDSTOP_PIN
```

--

enfin, activer le bed Sensor , en remplaçant:

```
// #define BD_SENSOR
```

Par:

```
#define BD_SENSOR
```

Fichier Configuration_adv.h

Mise en place dans le menu de l'imprimante de la fonction de nivellement automatique

```
//#define PROBE_OFFSET_WIZARD // Add a Probe Z Offset calibration  
option to the LCD menu  
#define PROBE_OFFSET_WIZARD // Add a Probe Z Offset calibration option  
to the LCD menu
```

N'oubliez pas de définir vos marges, si vous avez des pinces...

```
#if PROBE_SELECTED && !IS_KINEMATIC  
#define PROBING_MARGIN_LEFT 5  
#define PROBING_MARGIN_RIGHT 5  
#define PROBING_MARGIN_FRONT 20  
#define PROBING_MARGIN_BACK 20
```

—

Installer une seconde langue dans votre imprimante, par exemple l'anglais, remplacez:

```
#if ANY(DOGLCD, TFT_COLOR_UI, TOUCH_UI_FTDI_EVE, IS_DWIN_MARLINUI)  
//#define LCD_LANGUAGE_2 fr  
//#define LCD_LANGUAGE_3 de  
//#define LCD_LANGUAGE_4 es  
//#define LCD_LANGUAGE_5 it  
#ifndef LCD_LANGUAGE_2  
//#define LCD_LANGUAGE_AUTO_SAVE // Automatically save language to EEPROM  
on change  
#endif  
#endif
```


par :

```
#if ANY(DOGLCD, TFT_COLOR_UI, TOUCH_UI_FTDI_EVE, IS_DWIN_MARLINUI)
  #define LCD_LANGUAGE_2 en
  //#define LCD_LANGUAGE_3 de
  //#define LCD_LANGUAGE_4 es
  //#define LCD_LANGUAGE_5 it
  #ifdef LCD_LANGUAGE_2
    #define LCD_LANGUAGE_AUTO_SAVE // Automatically save language to EEPROM on
change
  #endif
#endif
```

--

Activez le bed leveling avant chaque impression

(Vous pouvez aussi ajouter dans votre slicer type Cura, dans le paramétrage de votre imprimante, une ligne avec « G29 », juste après celle « G28 »)

Remplacez :

```
//#define ENABLE_LEVELING_AFTER_G28
```

Par :

```
#define ENABLE_LEVELING_AFTER_G28
```

--

Pour préchauffer avant le bed leveling

Remplacez :

```
//#define PREHEAT_BEFORE_LEVELING
#if ENABLED(PREHEAT_BEFORE_LEVELING)
  #define LEVELING_NOZZLE_TEMP 120 // (°C) Only applies to E0 at this time
  #define LEVELING_BED_TEMP 50
#endif
```

Par :

```
#define PREHEAT_BEFORE_LEVELING
#if ENABLED(PREHEAT_BEFORE_LEVELING)
  #define LEVELING_NOZZLE_TEMP 210 // (°C) Only applies to E0 at this time
  #define LEVELING_BED_TEMP 60
#endif
```