

Publicité



Menu

fl

Premiers pas avec STM32 Nucleo dans Arduino IDE

Ceci est un guide de démarrage sur STM32 Nucleo avec [Arduino IDE](#) . Dans ce didacticiel, nous apprendrons à utiliser les cartes STM32 Nucleo avec le cadre de programmation Arduino. Dans un premier temps, nous verrons comment installer le framework STM32duino dans l'IDE Arduino. Deuxièmement, nous verrons un exemple de clignotement de LED pour utiliser les broches GPIO de la carte Nucleo comme broches de sortie numérique.

Présentation des cartes Nucleo STM32

ST Microelectronics a introduit des cartes de développement puissantes, économiques et abordables par rapport aux cartes de découverte STM32. Parce que les cartes découverte ont des prix plus élevés que les cartes Nucleo. Ces cartes vous permettent de développer et de prototyper votre application rapidement à moindre coût. Plus important encore, ils sont compatibles avec un grand nombre de modules complémentaires matériels disponibles pour Arduino Uno et Arduino Nano.

Trois sont trois principaux types de sangliers Nucleo qui sont classés en termes de taille de flash, de hautes performances, de puissance ultra-faible et de fonctions de capacité sans fil telles que Nucleo-32, Nucleo -64 et

00:39 / 03:20





PCBWay PCB Fabrication & Assembly

ONLY \$5 for 10 PCBs

✓ 24-hour Build Time ✓ Quality Guaranteed

✓ Most Soldermask Colors:

■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■

Order now

www.pcbway.com

Publicité

Il existe différents environnements de programmation disponibles pour programmer les cartes Nucleo telles que Arduino, STM32Cube, Mbed, Zephyr RTOS, etc. Mais nous utiliserons le framework Arduino dans cette série de tutoriels.



Pourquoi utiliser Arduino pour programmer les cartes Nucleo STM32 ?

La principale raison d'utiliser Arduino pour programmer les cartes STM32 est due à la simplicité et à la facilité d'utilisation de l'IDE Arduino pour programmer diverses cartes. Deuxièmement, si vous connaissez déjà Arduino et que vous l'avez utilisé pour programmer des cartes Arduino, vous pouvez facilement appliquer cette expérience pour programmer également des cartes STM32. Plus important encore, l'équipe officielle de STM32 Core a porté toutes les cartes Nucleo et autres cartes STM32 dans l'IDE Arduino. Cela signifie que nous pouvons utiliser les mêmes API Arduino pour utiliser les cartes STM.

Vous pouvez obtenir plus d'informations sur le noyau Arduino STM32 à partir de ce [référentiel](#).

Installation du noyau STM32 dans l'IDE Arduino

Après avoir brièvement présenté les cartes Nucleo STM32 et le noyau Arduino STM32, voyons comment installer la bibliothèque STM32 dans l'IDE Arduino.

rer que l'IDE Arduino est installé sur votre système. Téléchargez et installez exploitation tels que Windows, Linux et MAC,

00:39 / 03:20

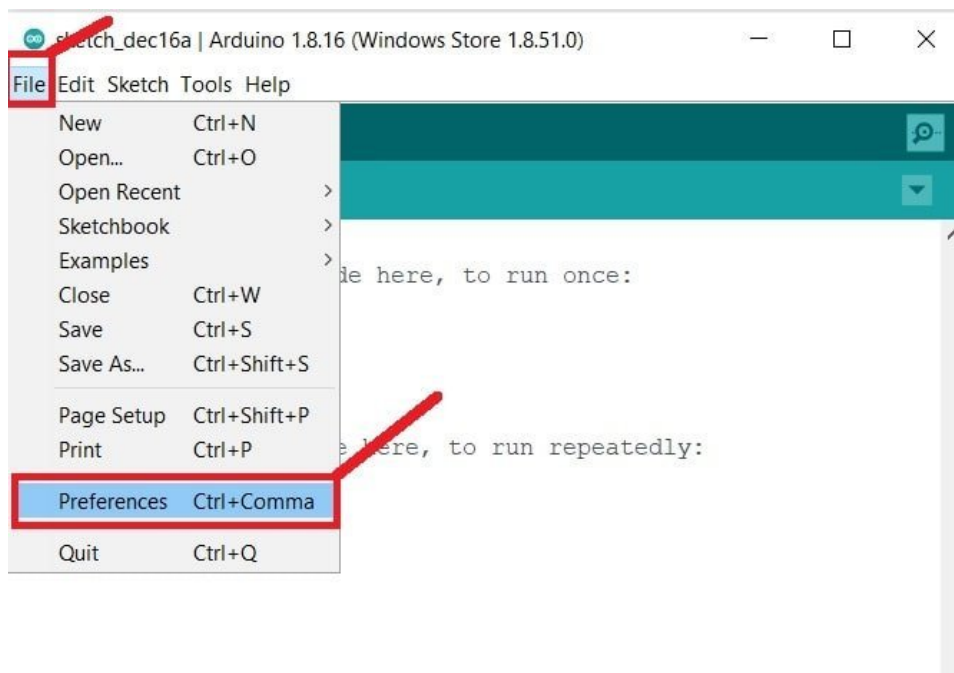


Publicité

Ajout du package de support de carte

Après avoir installé l'IDE Arduino, nous devons ajouter un package de support de carte du noyau Arduino STM32 dans l'IDE Arduino. Le gestionnaire de packages de support de carte vous permet d'ajouter des cartes supplémentaires dans l'IDE Arduino. Suivez ces étapes pour ajouter STM32 :

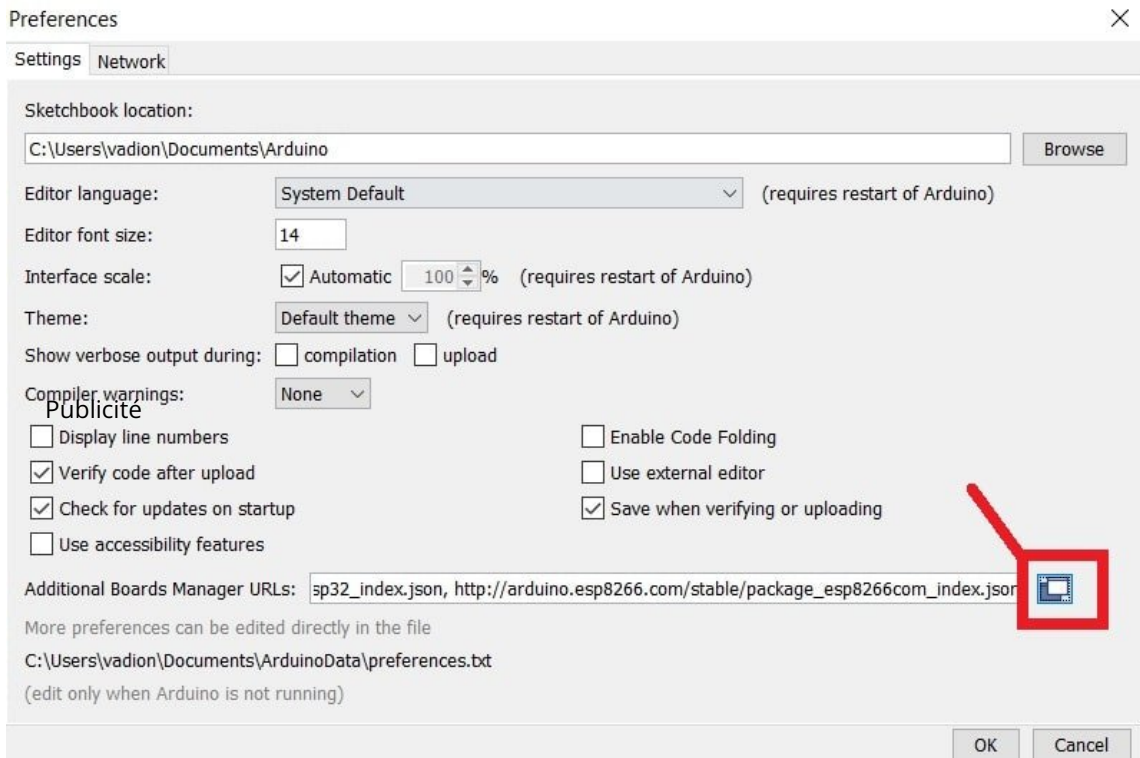
Ouvrez IDE et cliquez sur Fichiers -> Préférences :



- ⊗ Après cela, la fenêtre "Préférences" apparaîtra. Cliquez sur la case rectangulaire à côté de "URL supplémentaires du gestionnaire de tableaux".

00:39 / 03:20





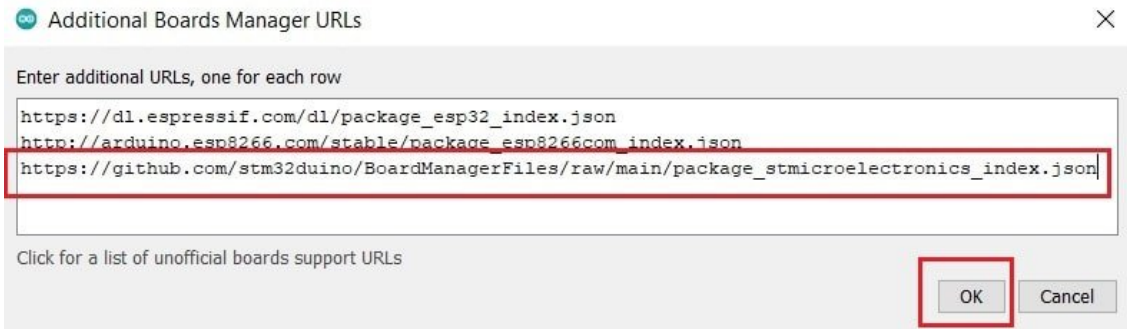
Ajoutez maintenant le lien suivant dans le champ « **URL supplémentaires des administrateurs de tableaux** » :



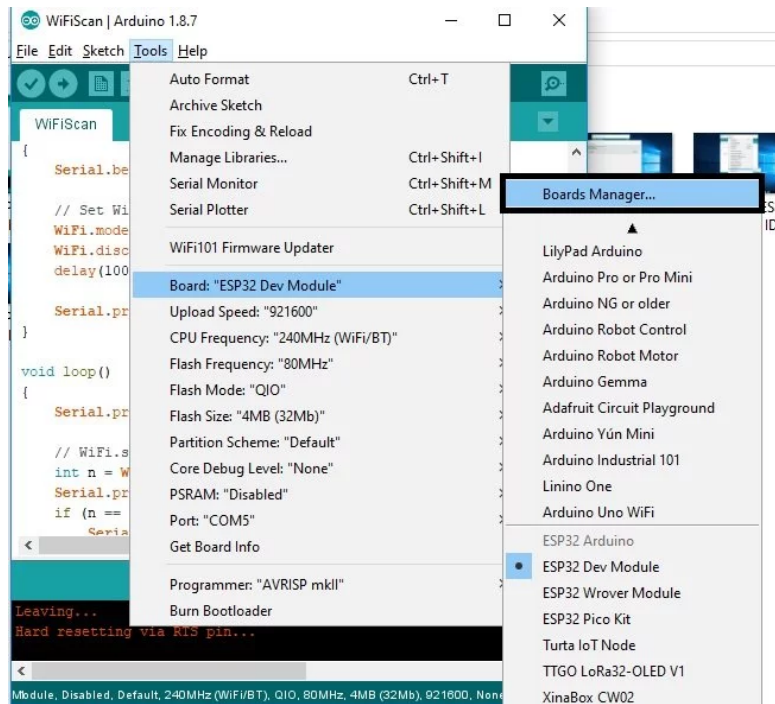
https://github.com/stm32duino/BoardManagerFiles/raw/main/package_stmicroelectronics_index.json



Selon les cartes supplémentaires que vous avez utilisées avec Arduino IDE, il peut déjà y avoir d'autres URL. Vous devez les ajouter à une nouvelle ligne. Maintenant, cliquez sur "OK" pour enregistrer les paramètres.



Allez maintenant dans outils > tableau > Gestionnaire de tableaux
Publicité

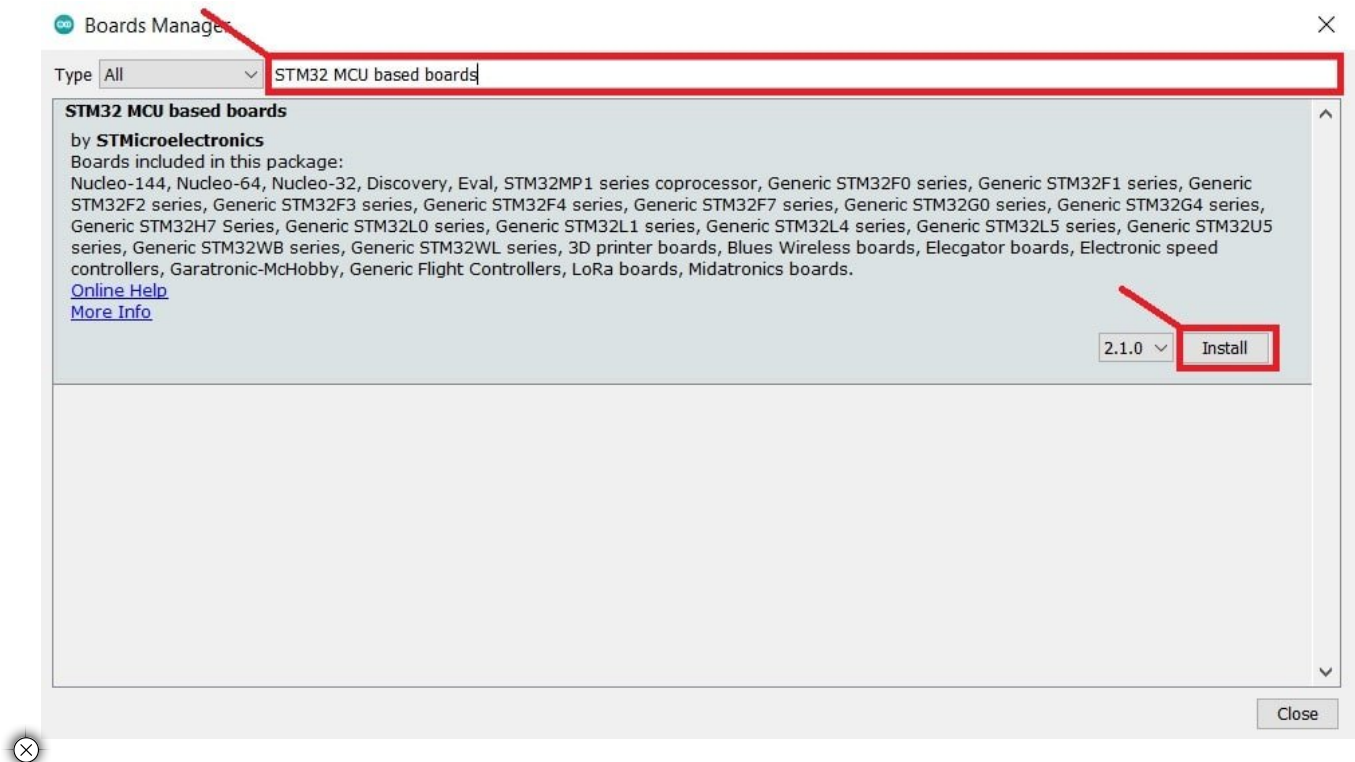


Après cela, une fenêtre de gestionnaire de carte s'ouvrira où vous pourrez voir une liste des cartes installées dans votre IDE Arduino.

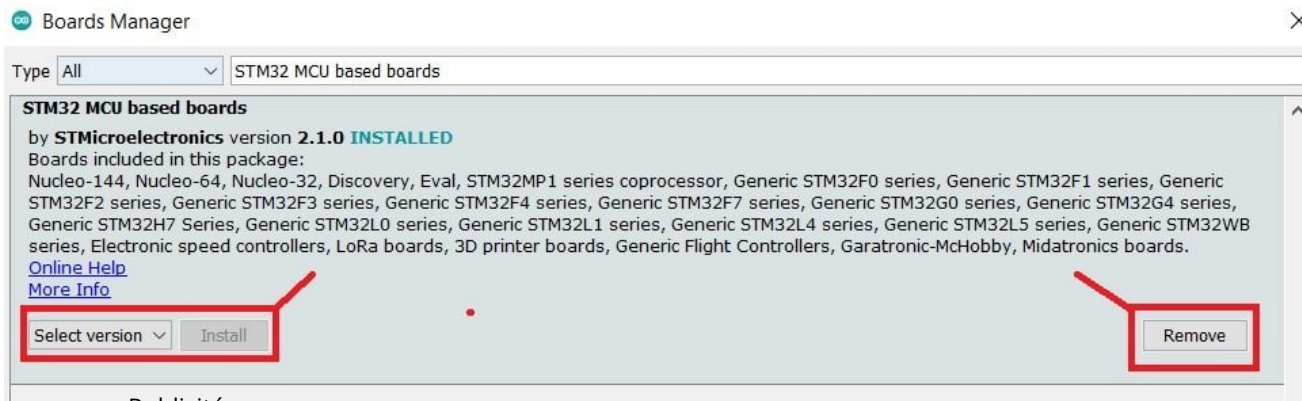




Tapez les « **cartes basées sur MCU STM32** » dans la barre de recherche et sélectionnez « **cartes basées sur MCU STM32** » et cliquez sur le bouton d'installation :

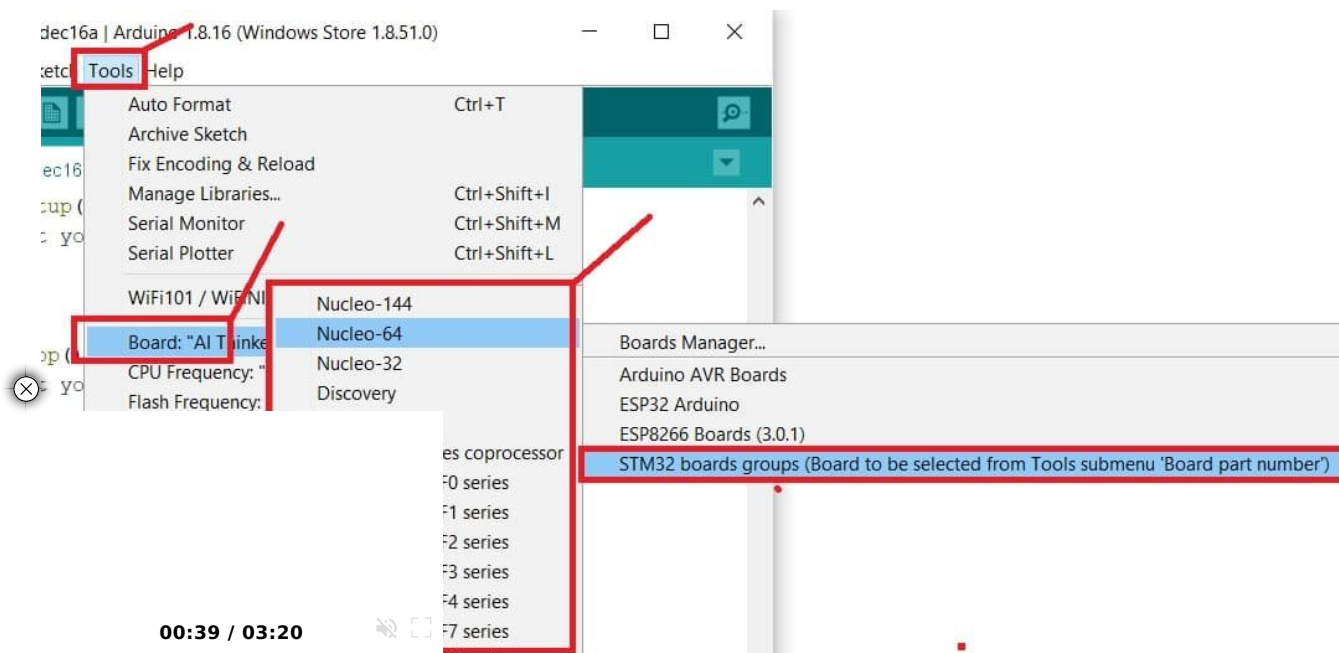


et API de base STM32 Arduino requises. Une fois l'installation terminée, un
s du nom du noyau STM32.



C'est ça. Vous pouvez installer le noyau Arduino STM32 dans l'IDE Arduino. Vous pouvez maintenant fermer la fenêtre du gestionnaire de carte.

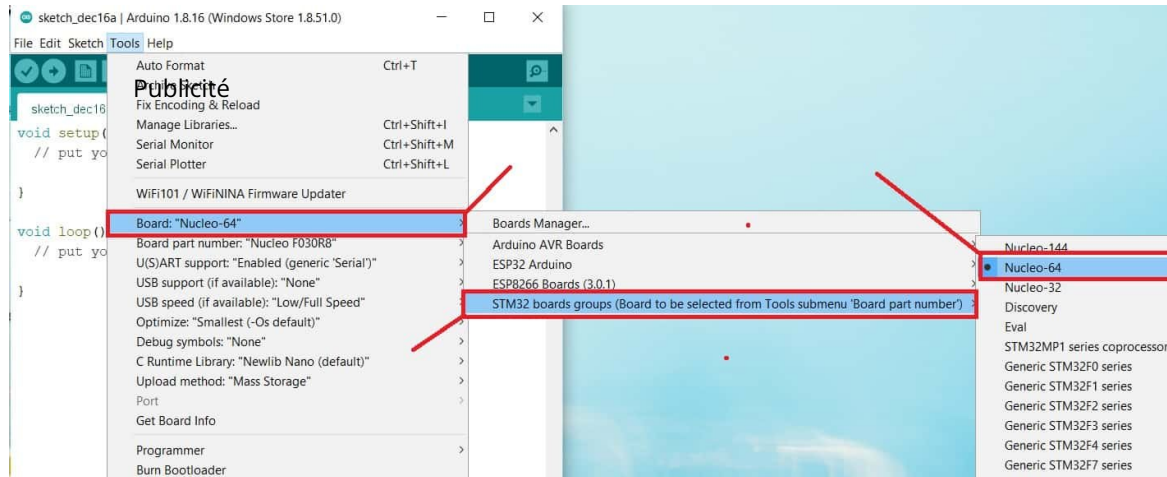
To check if the library is installed properly or not, you can go to tools > boards > stm32 boards groups. You can select the Nucleo boards such as Nucleo-64, Nucleo-144, etc.



write your first program and upload it to the STM32 Nucleo board.

In this tutorial, we will use the Nucleo-F103RB board. But you can use any other Nucleo board also and follow the same instructions to program your boards.

The Nucleo-F103RB board which we are using in this tutorial belongs to Nucleo-64 types. Therefore select, Nucleo-64 from STM32 board groups. You can select other Nucleo types according to the board which you are using such as Nucleo-32 and Nucleo-144.

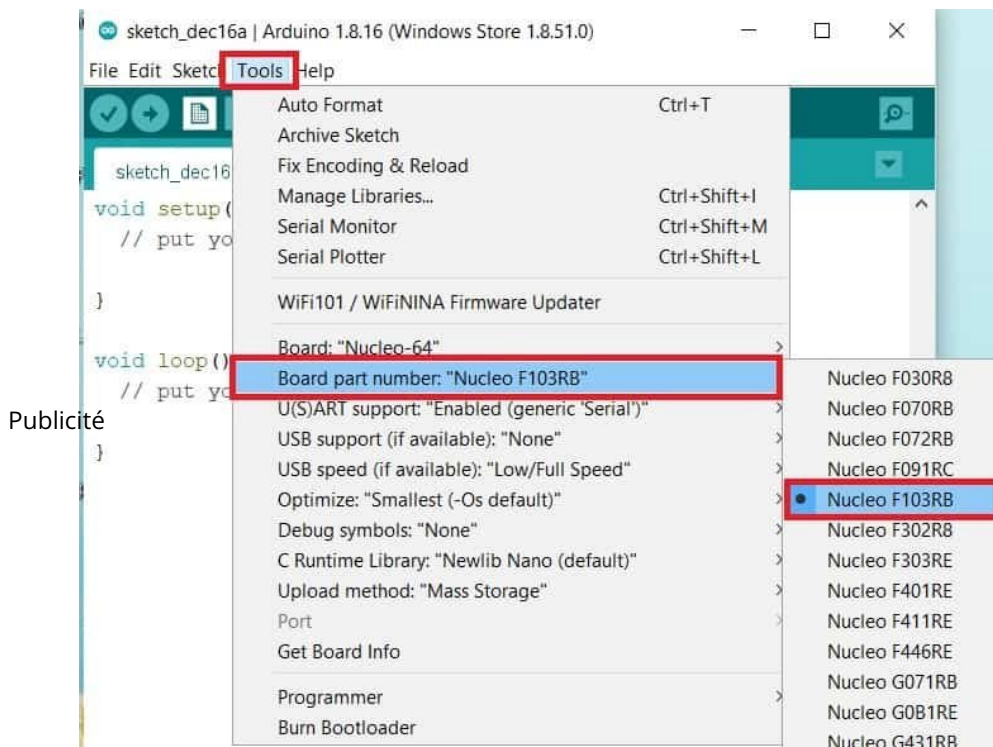


As mentioned earlier, there are many variants of Nucleo boards that come with different STM32 chips. The one which we are using has a part number of Nucleo-F103RB. Hence, select your board part number from Tools > Board Part Number:



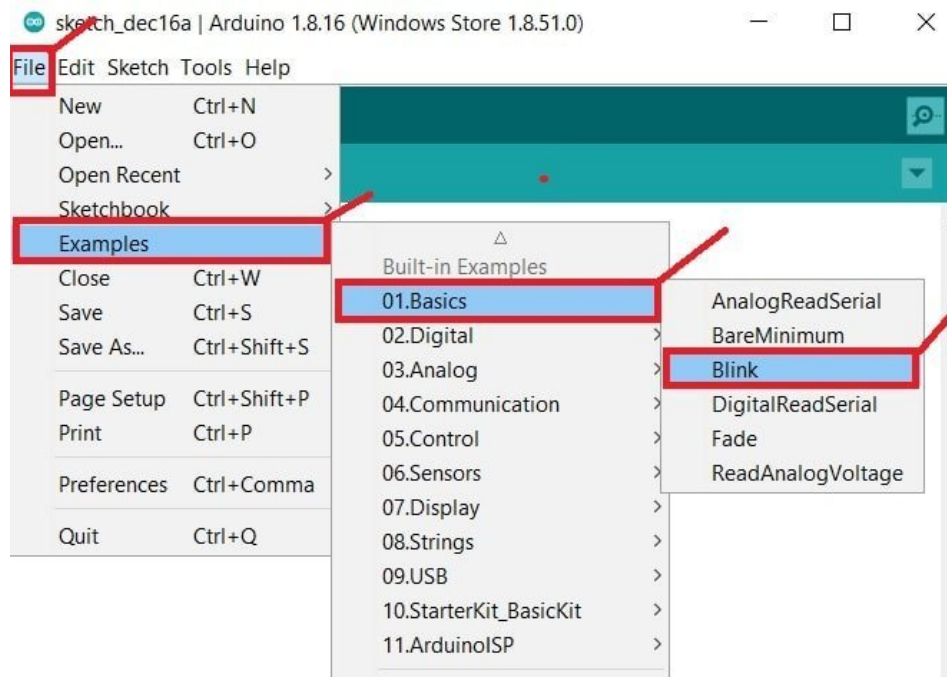
00:39 / 03:20





STM32 Nucleo LED Blinking Example

Now let's see a sample LED blinking example available in examples of Arduino IDE. To open the LED blinking example go to File > Examples > Basics > Blink and click on it. It will open a simple LED blinking example which will blink onboard LED of Nucleo-F103RB at the rate of 1 second.



00:39 / 03:20

second, then off for one second, repeatedly.

If you want to know what pin the on-board LED is connected to on your Arduino model, check the Technical Specs of your board at:

<https://www.arduino.cc/en/Main/Products>

modified 8 May 2014

by Scott Fitzgerald

modified 2 Sep 2016

by Arturo Guadalupi

modified 8 Sep 2016

by Colby Newman

This ~~example~~ ^{Publicité} code is in the public domain.

<https://www.arduino.cc/en/Tutorial/BuiltInExamples/Blink>

```
*/
```

```
// the setup function runs once when you press reset or power the board
```

```
void setup() {
```

```
  // initialize digital pin LED_BUILTIN as an output.
```

```
  pinMode(LED_BUILTIN, OUTPUT);
```

```
}
```

```
// the loop function runs over and over again forever
```

```
void loop() {
```

```
  digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
```

```
  delay(1000); // wait for a second
```

```
  digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW
```

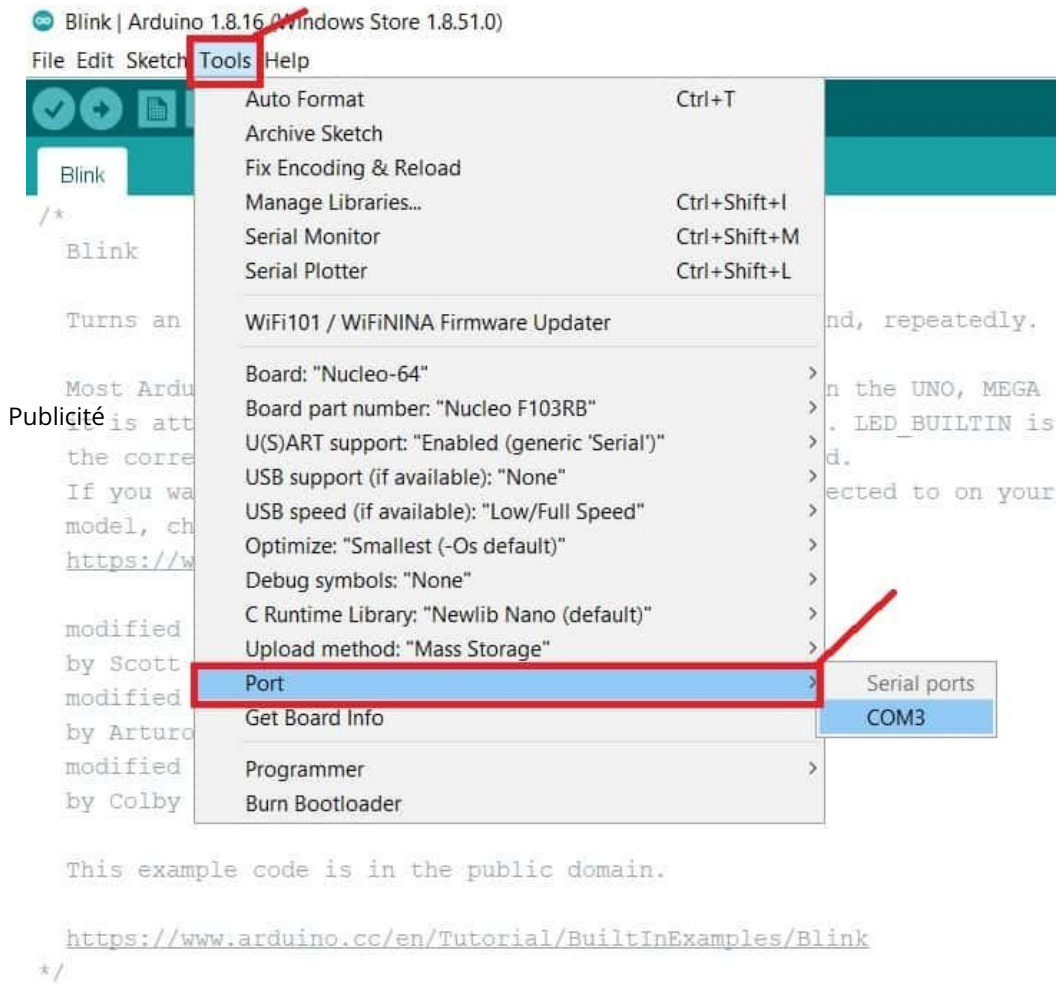
```
  delay(1000); // wait for a second
```

```
}
```

The above code blinks an onboard LED of Nucleo-F103RB after every one second. This is mentioned as LD2 on the Nucleo board and it is connected with the PA5 pin of F103RB.

Now select the COM port to which your Nucleo board is connected. Go to Tools > Ports:





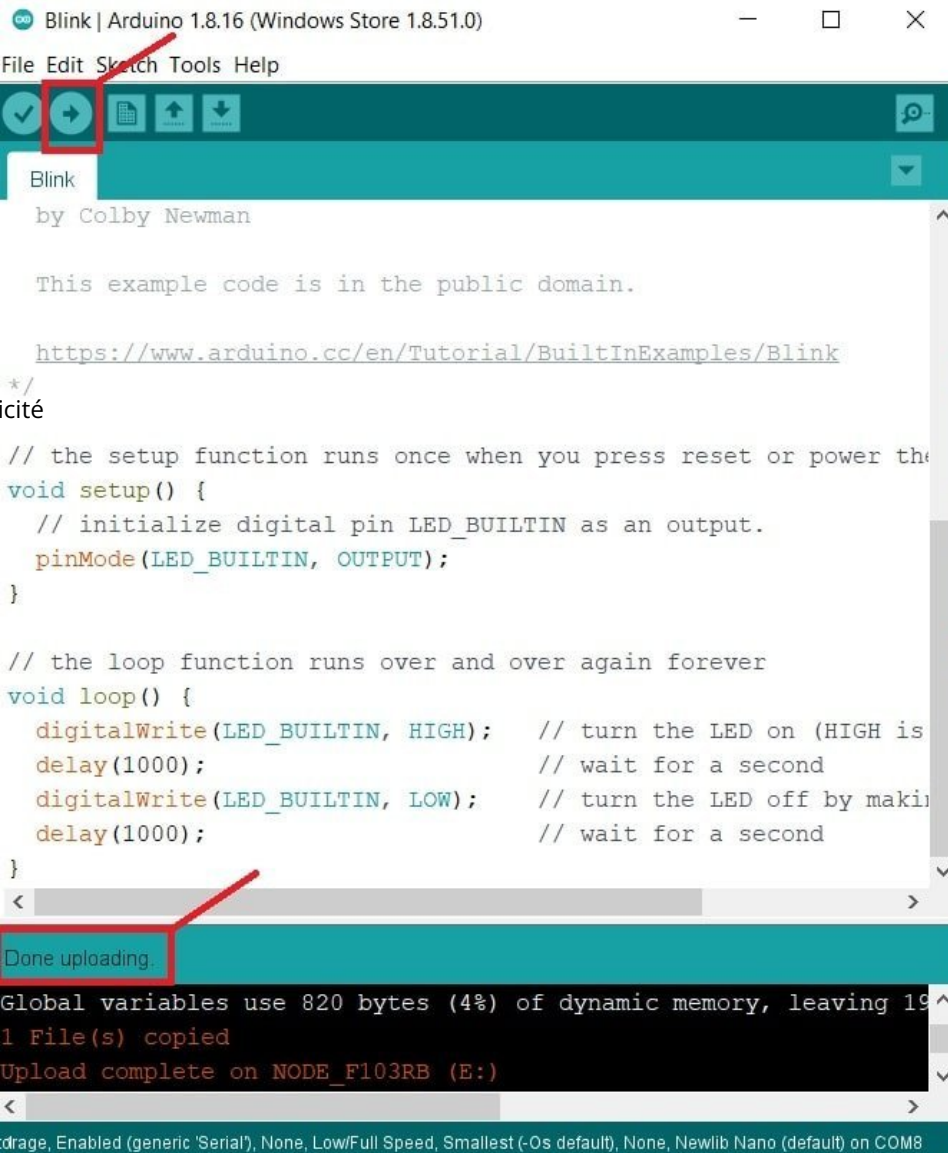
After that, compile the code and hit on upload button to upload code to the Nucleo board.

Ad



00:39 / 03:20





The screenshot shows the Arduino IDE window titled "Blink | Arduino 1.8.16 (Windows Store 1.8.51.0)". The menu bar includes "File", "Edit", "Sketch", "Tools", and "Help". The toolbar contains icons for "Verify", "Upload", "New", "Open", and "Save". The "Upload" icon is highlighted with a red box and a red arrow. Below the toolbar, the sketch name "Blink" is shown, followed by "by Colby Newman". The text "This example code is in the public domain." and the URL "https://www.arduino.cc/en/Tutorial/BuiltInExamples/Blink" are visible. The code is as follows:

```
*/
Publicité

// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the positive voltage)
  delay(1000); // wait for a second
  digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW
  delay(1000); // wait for a second
}
< >
```

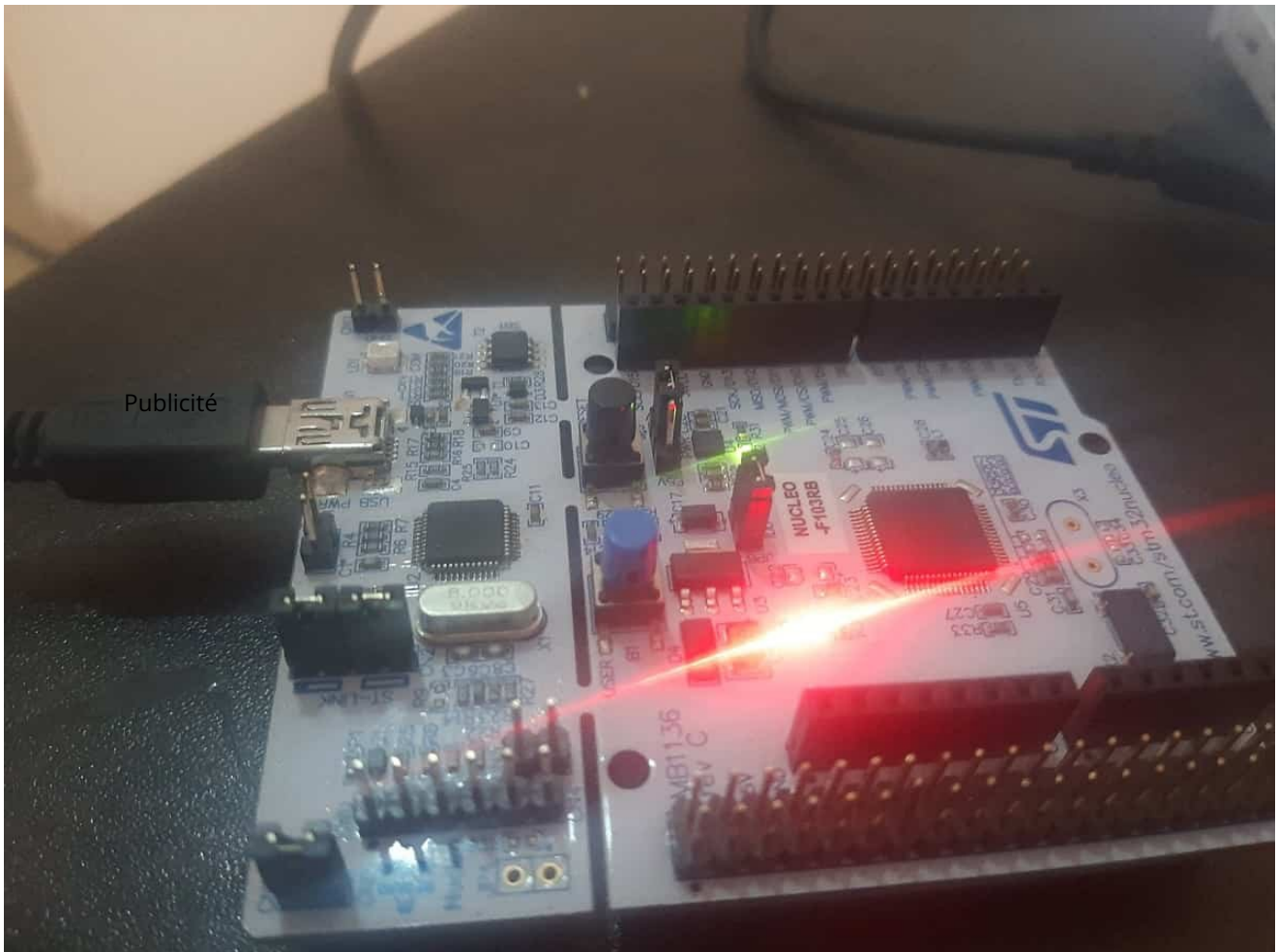
Below the code editor, the "Done uploading." message is shown in a teal box, also highlighted with a red box and a red arrow. The status bar at the bottom indicates "Global variables use 820 bytes (4%) of dynamic memory, leaving 19... 1 File(s) copied Upload complete on NODE_F103RB (E:)".

After uploading the code, you will notice that the onboard LED of the Nucleo board will start blinking as follows:



00:39 / 03:20





In summary, in this tutorial, we have learned to use STM32 Nucleo boards with Arduino. Firstly, we have seen an Introduction to Nucleo boards. Secondly, we have learned to install STM32 Arduino Core in Arduino IDE. In the end, we see how to program and upload code to the Nucleo board using Arduino IDE.

You may also like to read:

- [LED Blinking Tutorial STM32F4 Discovery Board – GPIO Pins with HAL Library](#)
- [Getting started with Keil uVision: Write your first Program for Tiva LaunchPad](#)
- [STM32 Blue Pill GPIO Pins with STM32Cube IDE: LED Blinking Tutorial](#)

STM32 Nucleo

Subscribe to Blog via Email

Entrez votre adresse e-mail pour vous abonner à ce blog et recevoir les notifications de nouveaux articles par e-mail.

00:39 / 03:20



Mohamed Babazadeh

17 décembre 2021 à 19h47

Avec mes meilleurs remerciements

[Répondre](#)

Publicité

Laissez un commentaire

Avertissez-moi des nouveaux commentaires par email.

Avertissez-moi des nouveaux articles par e-mail.

POSTER UN COMMENTAIRE



00:39 / 03:20



Publicité

Choisir une catégorie 



00:39 / 03:20



Publicité

Publicité

Publicité

Abonnez-vous au blog par e-mail

 S'ABONNER

00:39 / 03:20



Publicité

Publicité



Catégories

Messages récents

[Serveur Web de commutateur momentané ESP32/ESP8266 : contrôler les sorties GPIO](#)

[Reconnectez ESP32 au WiFi après une perte de connexion \(résolu\)](#)

[ESP32 avec accéléromètre, gyroscope et capteur de température MPU6050 \(IDE Arduino\)](#)

[ESP32 Enregistrer les données sur Flash en permanence à l'aide de la bibliothèque de préférences](#)

[ESP32 conserve le dernier état GPIO après une réinitialisation](#)

Publicité



©2013-2023 Microcontrollerslab.com Tous droits réservés

00:39 / 03:20

