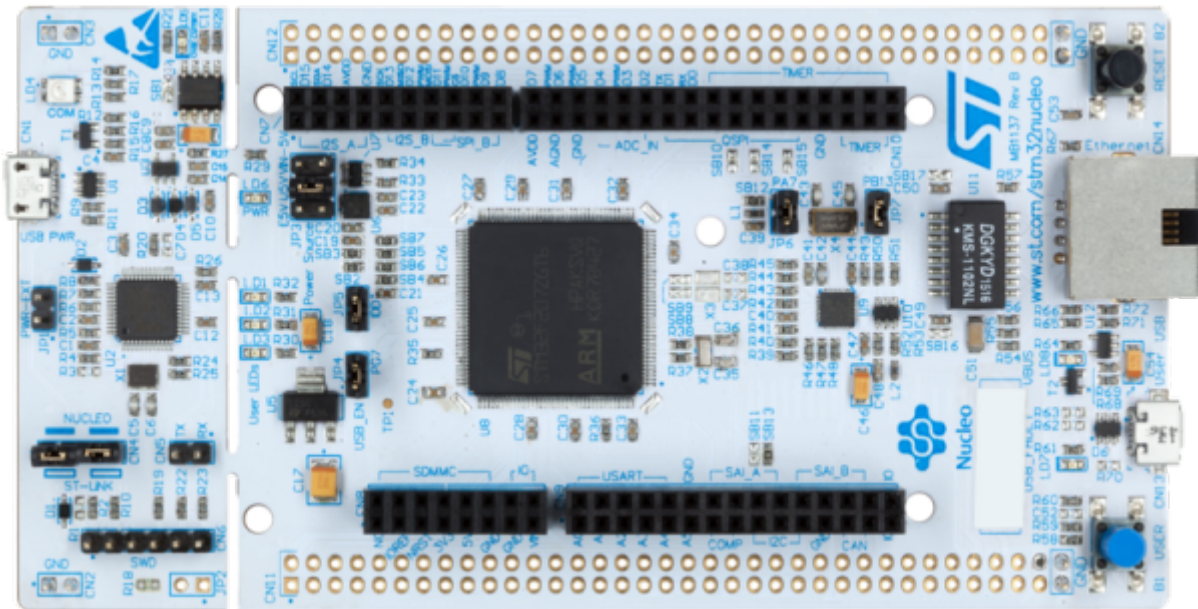


STM32

STM32 NUCLEO 144



[STM32 pour l'éducation](#)

[Achat Kits et STM32](#)

[Les Kits STM32](#)

[Introduction Debogueur Arduino](#)

[Premiers pas avec STM32 Nucleo dans Arduino IDE - LED clignotante](#)

[premiers_pas_avec_stm32_nucleo_dans_arduino_ide_-_led_clignotante.pdf](#)

Debuter avec la carte STM32 Nucleo F334R8

Connexion à la carte

Via un câble USB connecté sur CN1

Logiciel

Arduino IDE2

Ajouter la bibliothèque de cartes complémentaires dans "préférences" Ajouter ce lien dans "Additional Boards Managers URLs":

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

Télécharger et installer Cube32Programmer

Au 21/01/2024, note de JPD : Il existe peut être une autre méthode pour transférer un programme fait depuis Arduino IDE dans la carte STM32 nucléo, mais à aujourd'hui, je n'ai essayé qu'avec l'upload via STM32CubeProgrammer.

Cette application est très probablement utilisée en arrière plan pour compiler et/ou transférer le programme. C'est masqué, il suffit juste d'installer STM32CubeProg sur le PC où Arduino IDE est utilisé. Attention à bien l'installer à l'emplacement par défaut, sinon j'ai cru lire qu'il fallait ajouter manuellement le lien dans un fichier (lequel ??).

Pour télécharger STM32CubeProg.

Lien site STM32 : <https://www.st.com/en/development-tools/stm32cubeprog.html#get-software>

Lien Drive CastellLab (version pas forcément à jour, version 2.15 au moment de la création):
https://drive.google.com/file/d/1mh0SZxBggXplmLCxyDhg-fykKXA28RM0/view?usp=drive_link

Configuration Arduino IDE 2 pour programmer F334R8

Dans Arduino IDE2, choisir la carte et vérifier les méthodes d'Upload :

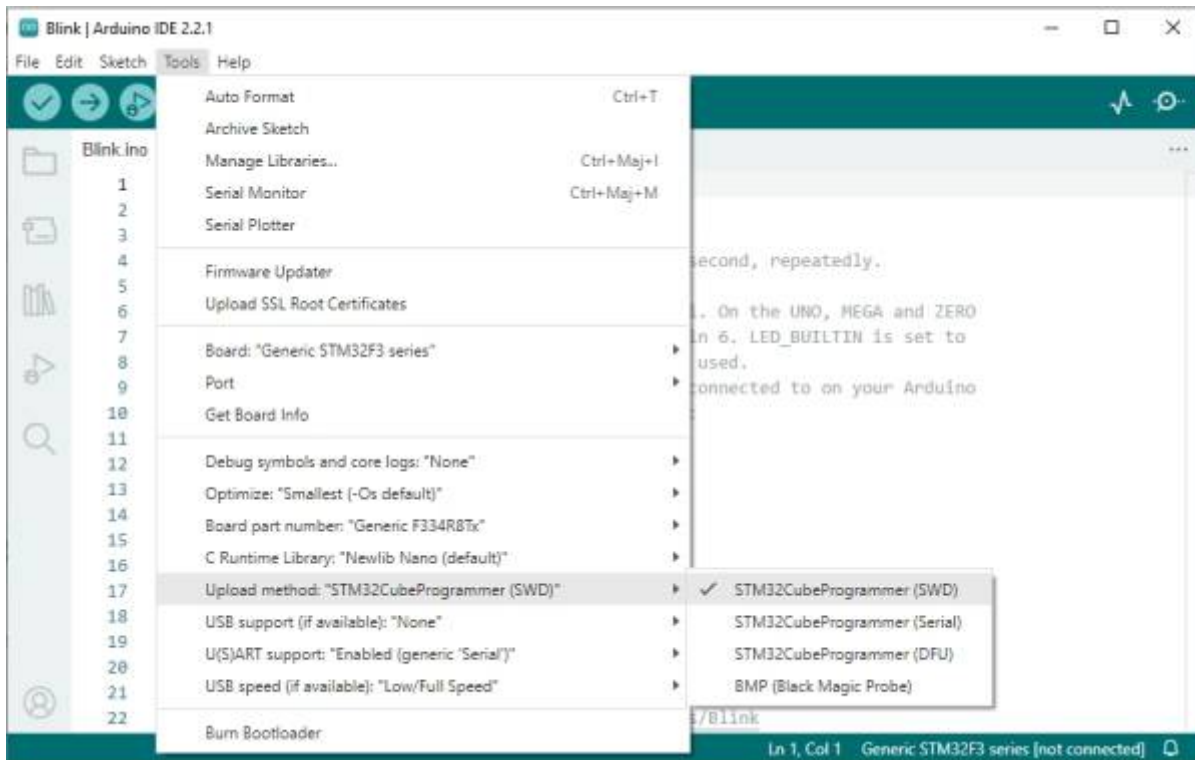
Choisir Generic STM32F3Series dans Board :



Dans les nouveaux sous-menu dans Tools, Choisir la carte F334R8Tx :



Dans Upload method : "STM32CubeProgrammer (SWD)"



Caracteristiques de la carte F334R8

Vue d'ensemble

STM32F334 BLOCK DIAGRAM



Broche GPIO Pinout



Pour piloter la led intégrée qui est connue comme "D13" sur Arduino, il faut sur la carte STM32334R8, piloter la sortie "PA5", ou "PA_5" comme illustré dans le tableau de correspondance ci-dessous :

Table 15. ARDUINO® connectors on NUCLEO-F334R8

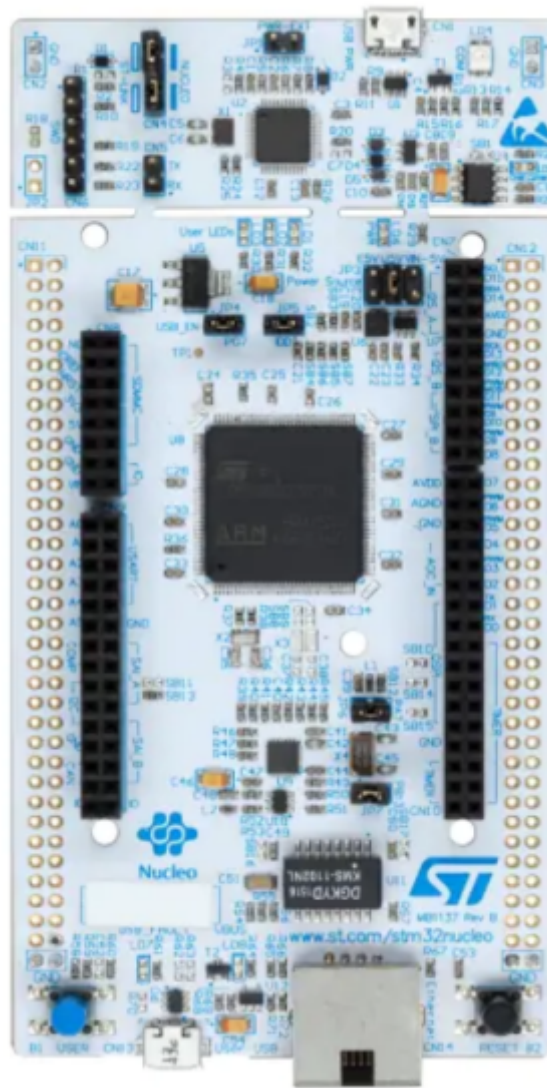
| Connector | Pin | Pin name | STM32 pin | Function |
|------------------|-----|----------|---------------------------|---------------------------------|
| Left connectors | | | | |
| CN6 power | 1 | NC | - | - |
| | 2 | IOREF | - | 3.3V Ref |
| | 3 | RESET | NRST | RESET |
| | 4 | +3.3V | - | 3.3V input/output |
| | 5 | +5V | - | 5V output |
| | 6 | GND | - | ground |
| | 7 | GND | - | ground |
| | 8 | VIN | - | Power input |
| CN8 analog | 1 | A0 | PA0 | ADC1_IN1 |
| | 2 | A1 | PA1 | ADC1_IN2 |
| | 3 | A2 | PA4 | ADC2_IN1 |
| | 4 | A3 | PB0 | ADC1_IN11 |
| | 5 | A4 | PC1 or PB9 ⁽¹⁾ | ADC_IN7 (PC1) or I2C1_SDA (PB9) |
| | 6 | A5 | PC0 or PB8 ⁽¹⁾ | ADC_IN6 (PC0) or I2C1_SCL (PB8) |
| Right connectors | | | | |
| CN5 digital | 10 | D15 | PB8 | I2C1_SCL |
| | 9 | D14 | PB9 | I2C1_SDA |
| | 8 | AREF | - | AVDD |
| | 7 | GND | - | ground |
| | 6 | D13 | PA5 | SPI1_SCK |
| | 5 | D12 | PA6 | SPI1_MISO |
| | 4 | D11 | PA7 | TIM17_CH1 or SPI1_MOSI |
| | 3 | D10 | PB6 | TIM16_CH1N or SPI1_CS |

Table 15. ARDUINO® connectors on NUCLEO-F334R8 (continued)

| Connector | Pin | Pin name | STM32 pin | Function |
|-------------|-----|----------|-----------|-----------|
| CN5 digital | 2 | D9 | PC7 | TIM3_CH2 |
| | 1 | D8 | PA9 | - |
| CN9 digital | 8 | D7 | PA8 | - |
| | 7 | D6 | PB10 | TIM2_CH3 |
| | 6 | D5 | PB4 | TIM3_CH1 |
| | 5 | D4 | PB5 | - |
| | 4 | D3 | PB3 | TIM2_CH2 |
| | 3 | D2 | PA10 | - |
| | 2 | D1 | PA2 | USART2_TX |
| | 1 | D0 | PA3 | USART2_RX |

1. Refer to [Table 10: Solder bridges](#) for details.

Debuter avec le STM32 144 NUCLEO-F429ZI

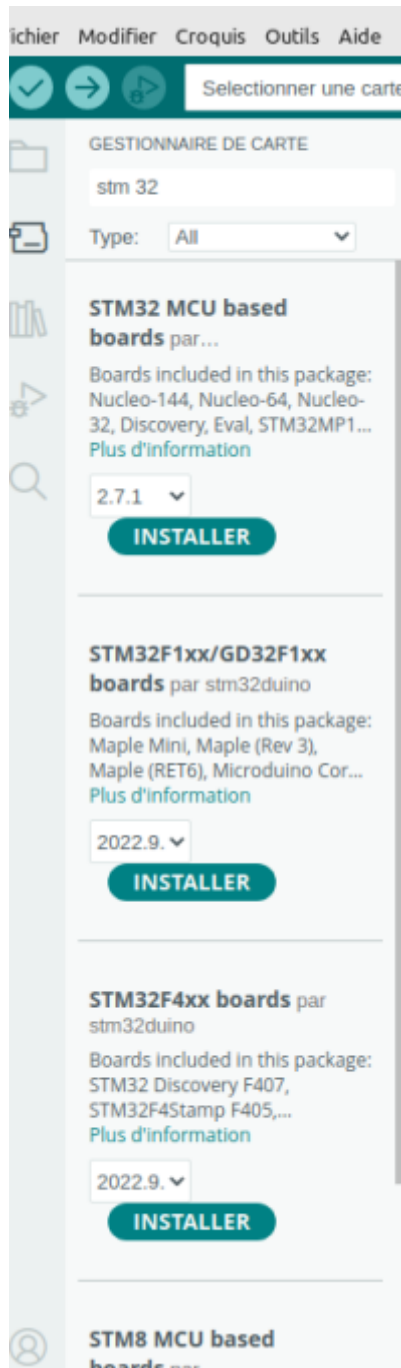


-1-Dans l IDE arduino version 2.x.x, inserer les deux lignes suivantes :

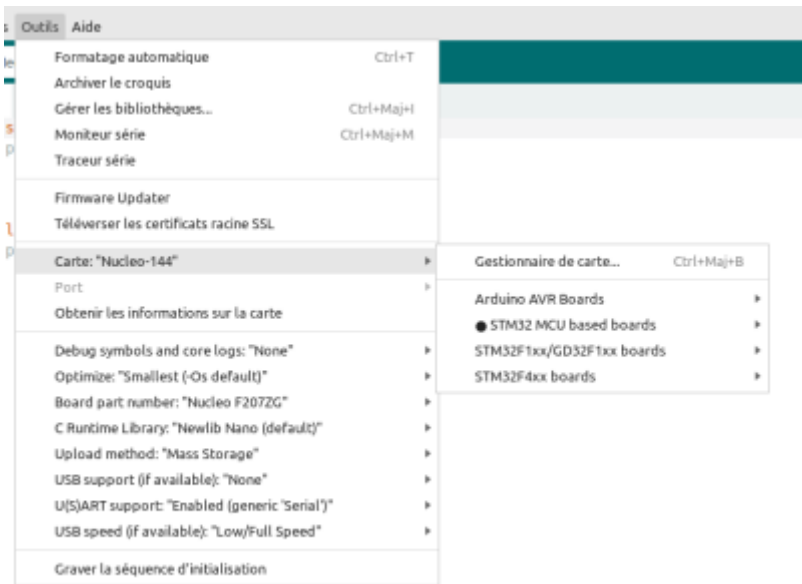
```
http://dan.drown.org/stm32duino/package_STM32duino_index.json  
https://github.com/stm32duino/BoardManagerFiles/raw/main/package_stmicroelec  
tronics_index.json
```



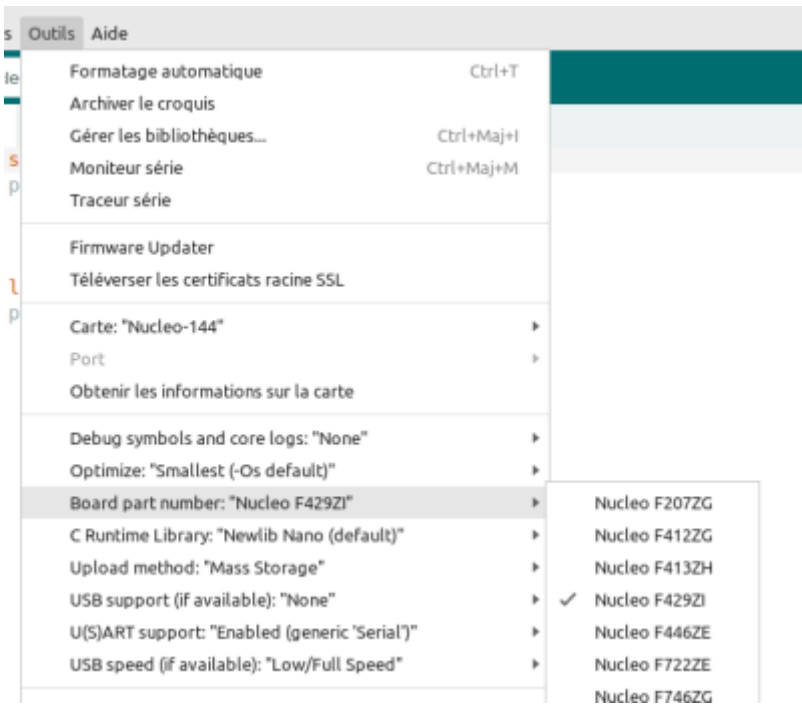
-2- Dans le gestionnaire de carte installer toutes les cartes STM32



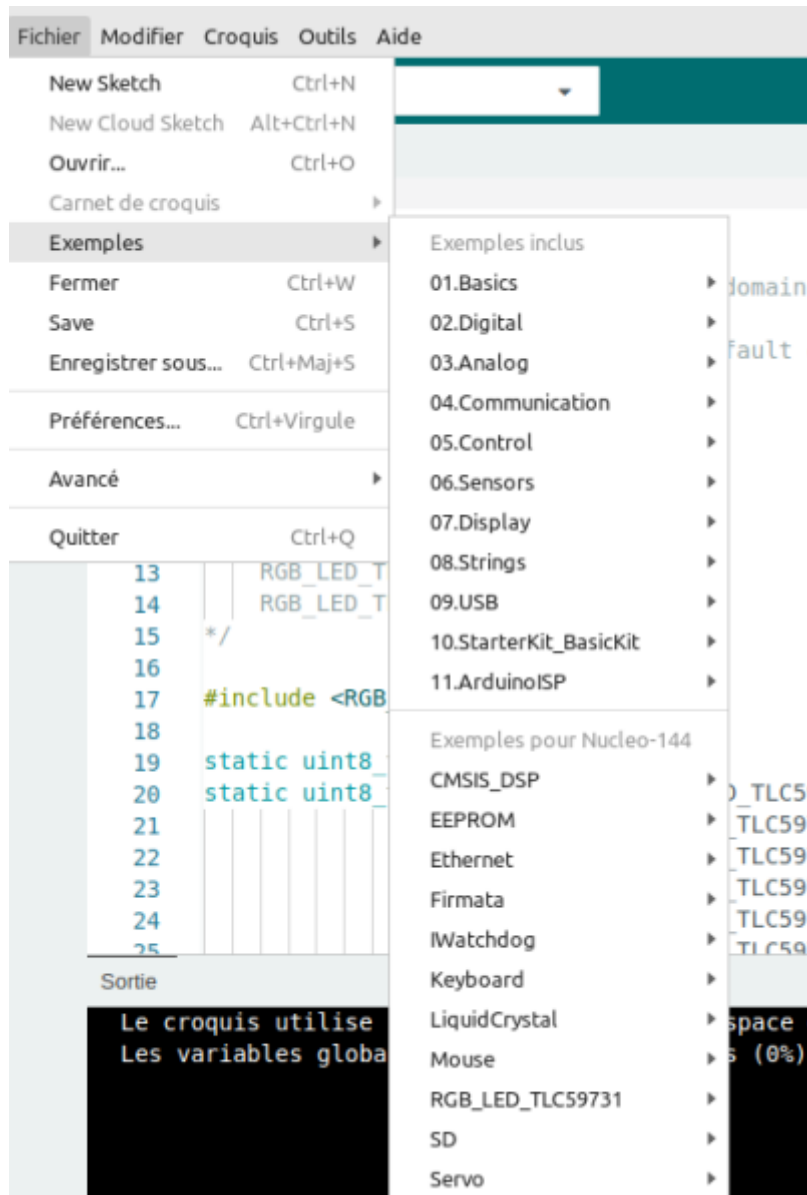
-3- Dans "Outils" -> "Carte" choisir "STM32 MCU Based Board" et "**Nucleo 144**"



-4- Dans l'option "Board part number" choisir ; "Nucleo F429ZI"



-5- On peut pour tester la carte , envoyer le programme exemple : "RGB_LED_TLC59731" ou le programme "Blink"



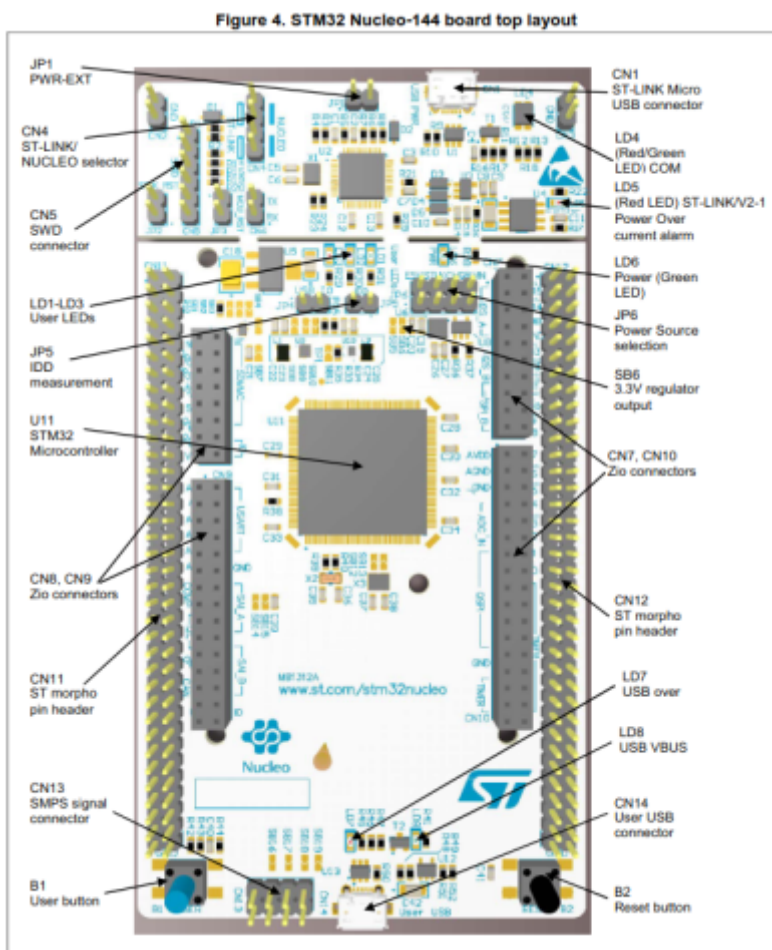
comment_programmer_et_deboguer_le_stm32_a_l_aide_de_l_arduino_led.pdf

Debuter avec Un STM32 144 NUCLEO-L4P5ZG



Carte Pour l'instant ... non comparable IDE arduino

Suivez la séquence ci-dessous pour configurer la carte Nucleo-144 et lancer la démonstration application (pour l'emplacement des composants, reportez-vous à la Figure 4 : Disposition du dessus de la carte STM32 Nucleo-144).



1. -. Vérifiez la position du cavalier sur la carte : JP1 (PWR-EXT) OFF (voir Section 5.5.1 : Entrée d'alimentation de ST-LINK/V2-1 USB connecteur pour plus de détails) JP6 (source d'alimentation) côté STLK (pour plus de détails, voir le tableau 7 : cavalier lié à l'alimentation) JP5 (IDD) ON (pour plus de détails, voir Section 5.8 : JP5 (IDD))CN4 ON sélectionné (pour plus de détails, voir Tableau 4 : états CN4 des cavaliers).
2. -. Pour l'identification correcte des interfaces de l'appareil à partir du PC hôte et avant connectant la carte, installez le pilote Nucleo USB disponible sur le Site Web www.st.com/stm32nucleo.
3. -. Pour alimenter la carte, connectez la carte STM32 Nucleo-144 à un PC avec un port USB de type A. au câble Micro-B' via le connecteur USB CN1 sur le ST-LINK. En conséquence, le les LED vertes LD6 (PWR) et LD4 (COM) s'allument et la LED rouge LD3 clignote.
4. -. Appuyez sur le bouton B1 (bouton gauche).
5. -. Observez que la fréquence de clignotement des trois LED LD1 à LD3 change, en cliquant sur sur le bouton B1.
6. -. La démonstration du logiciel et les nombreux exemples de logiciels, qui permettent à l'utilisateur de utiliser les fonctions Nucleo, sont disponibles sur la page Web www.st.com/stm32nucleo.
7. -. Développez une application en utilisant les exemples disponibles.

Flipper zero

[Flipper Zero](#)

From:

<https://magenealogie.chanterie37.fr/www/fablab37110/> - Castel'Lab le Fablab MJC de Château-Renault

Permanent link:

<https://magenealogie.chanterie37.fr/www/fablab37110/doku.php?id=start:stm32&rev=1705918525>

Last update: **2024/01/22 11:15**

