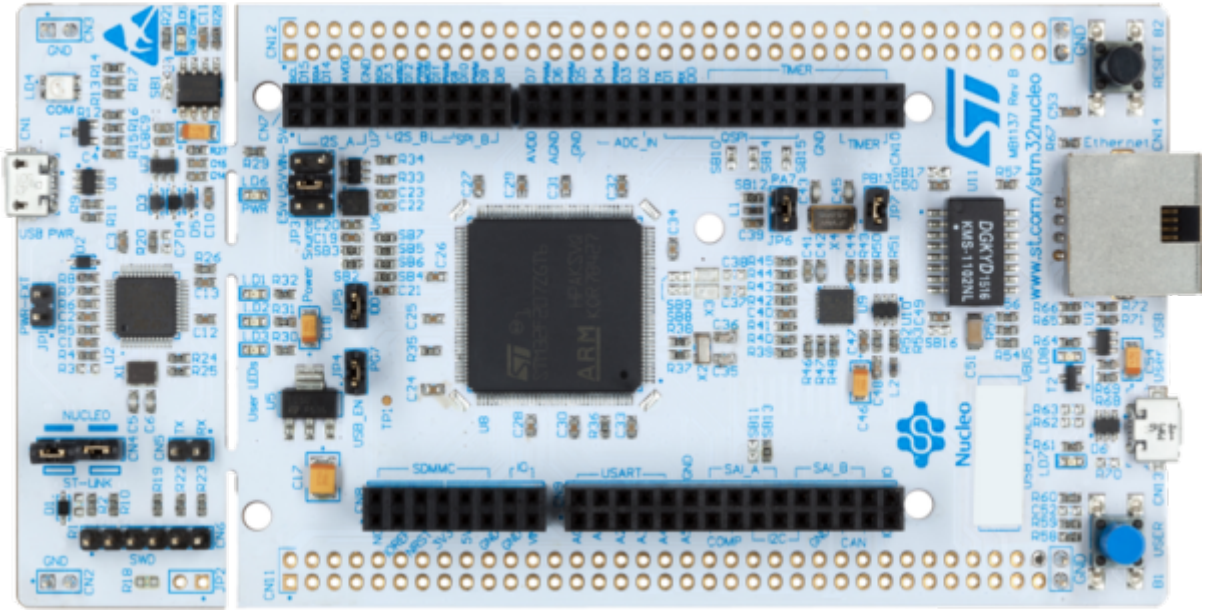
STM32

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 ??).

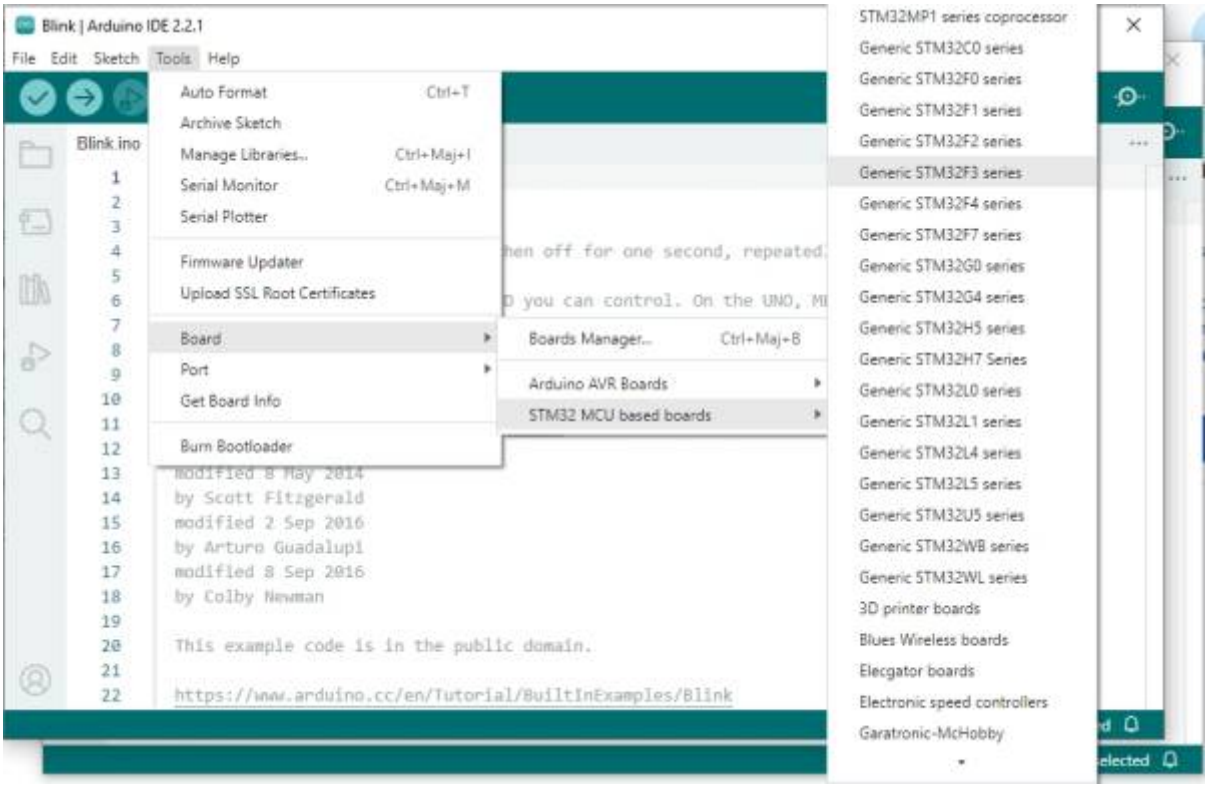
Pour télécharger STM32CubeProg.

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

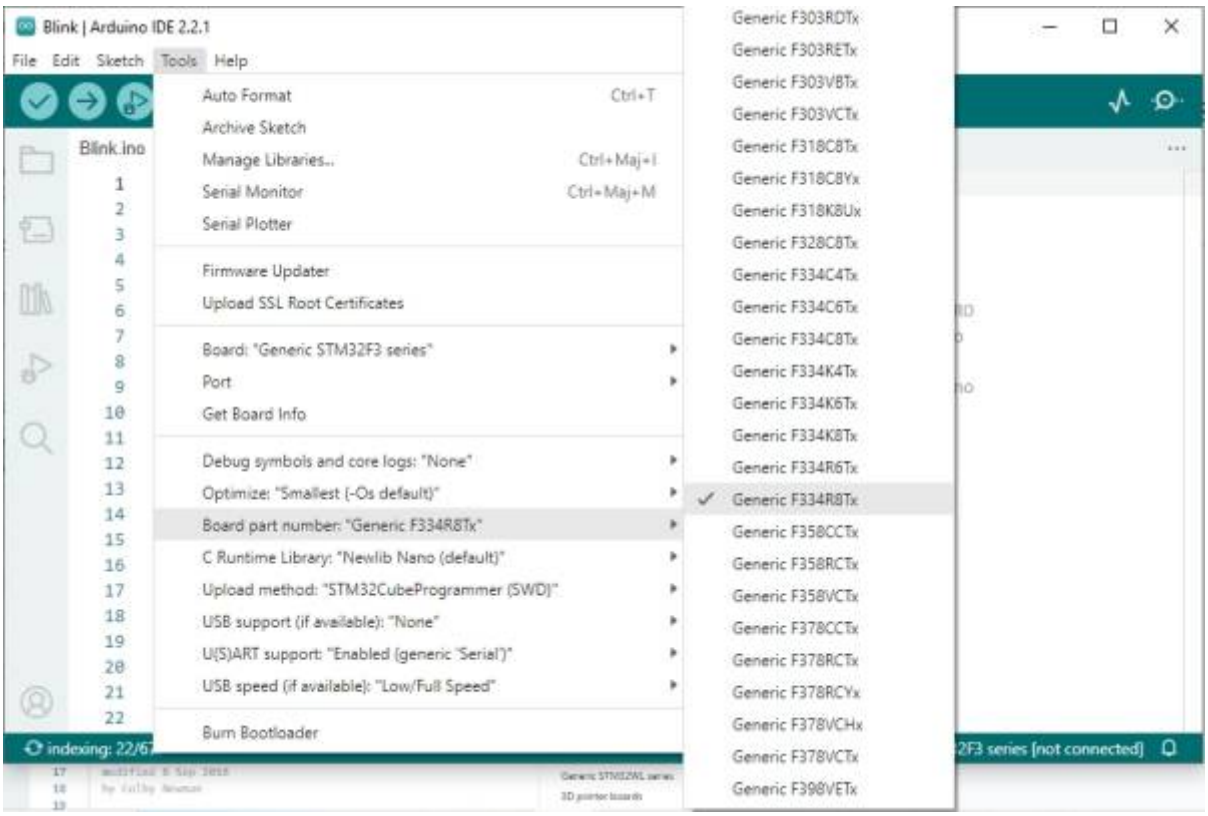
Configuration Arduino IDE 2 pour programmer F334R8

Dans Arduino IDE2, choisir la carte et vérifier les méthodes d’UpLoad :

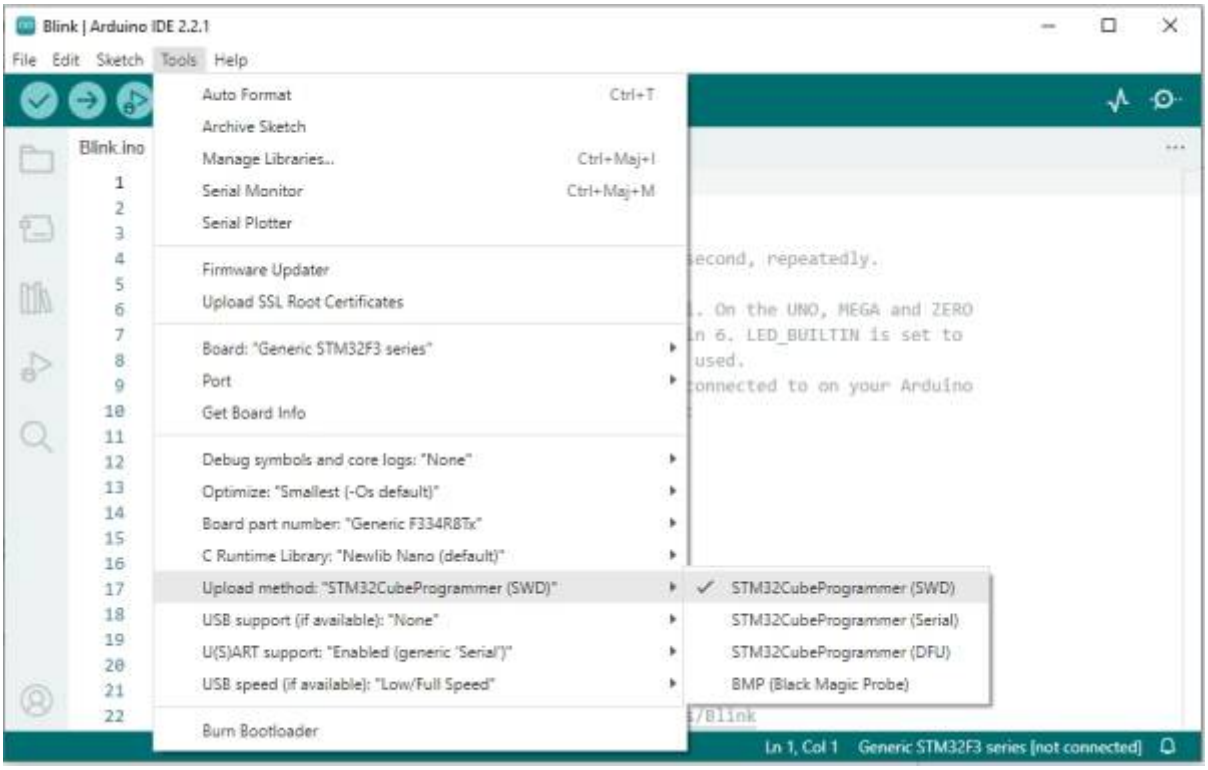
Choisir Generic STM32F3Series dans Board :



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

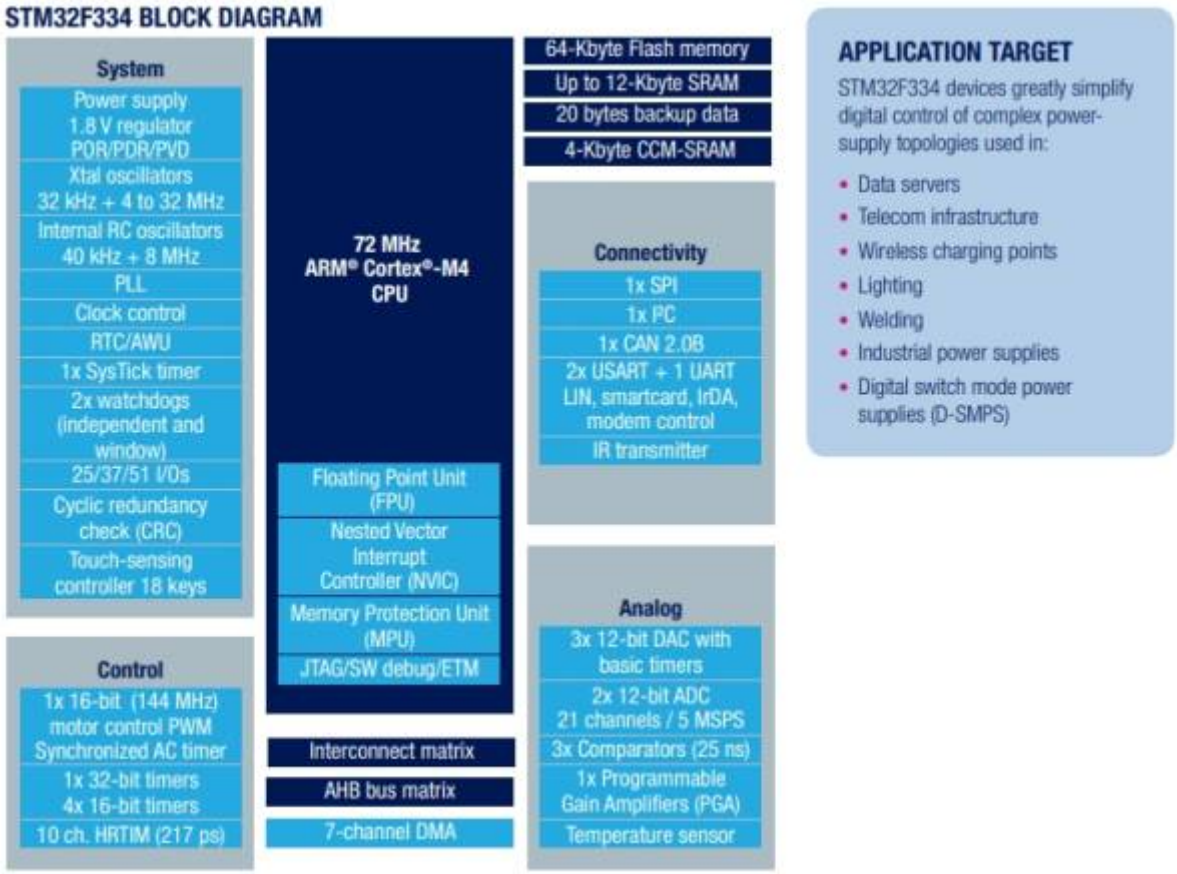


Dans Upload method : “STM32CubeProgrammer (SWD)”

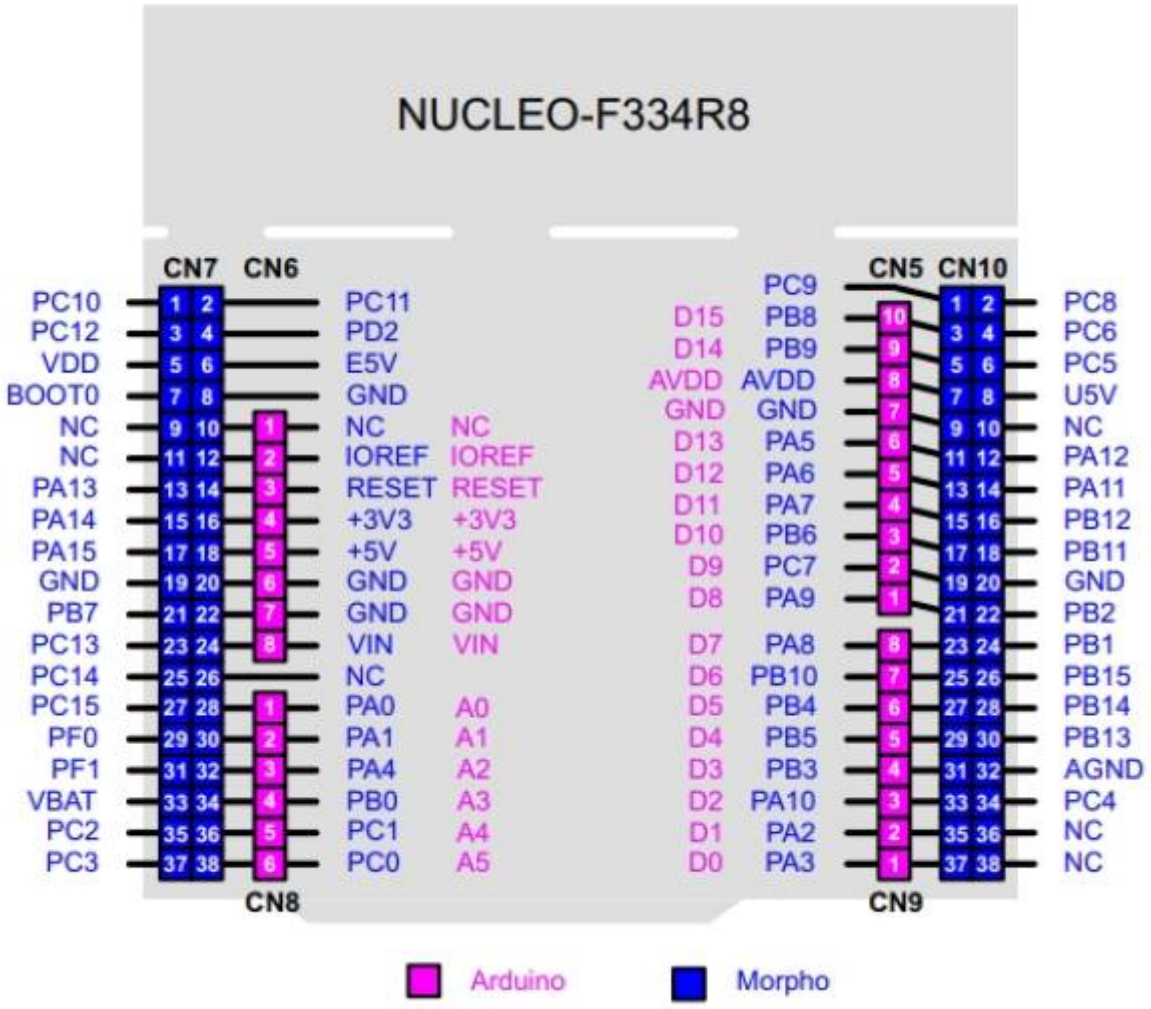


Caracteristiques de la carte F334R8

Vue d'ensemble



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

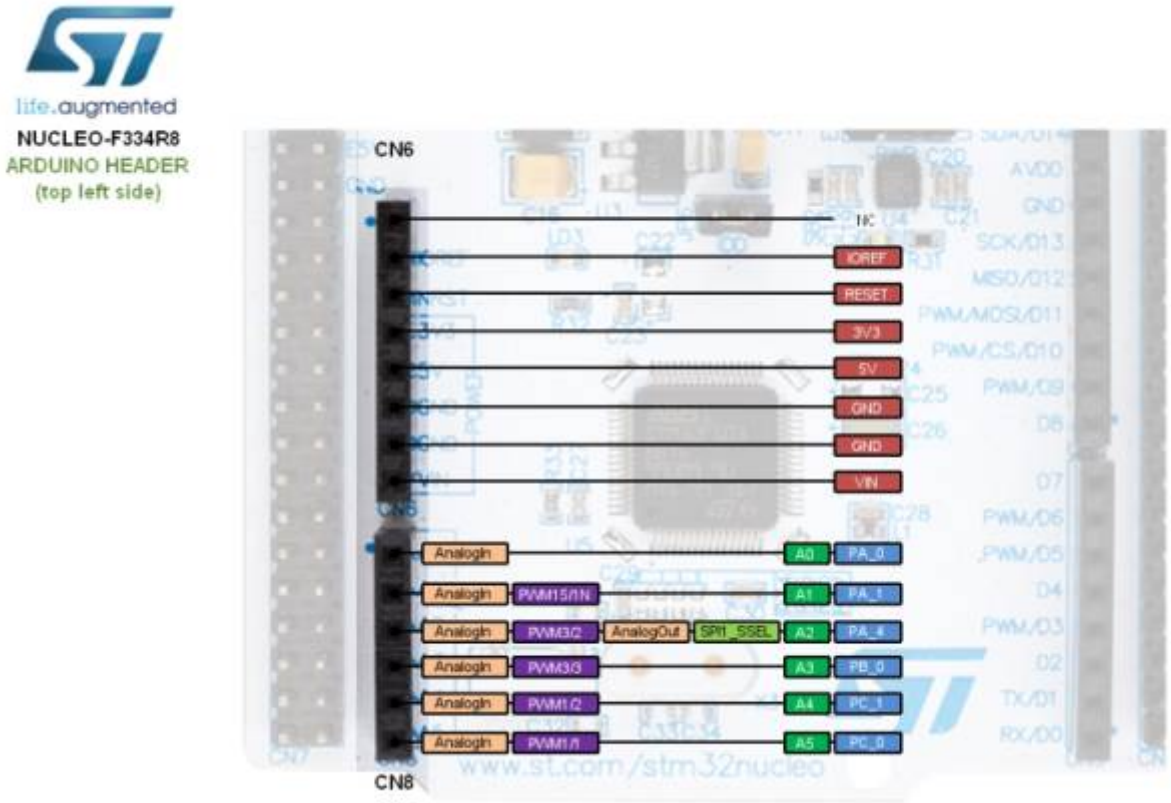
1. Refer to *Table 10: Solder bridges* for details.

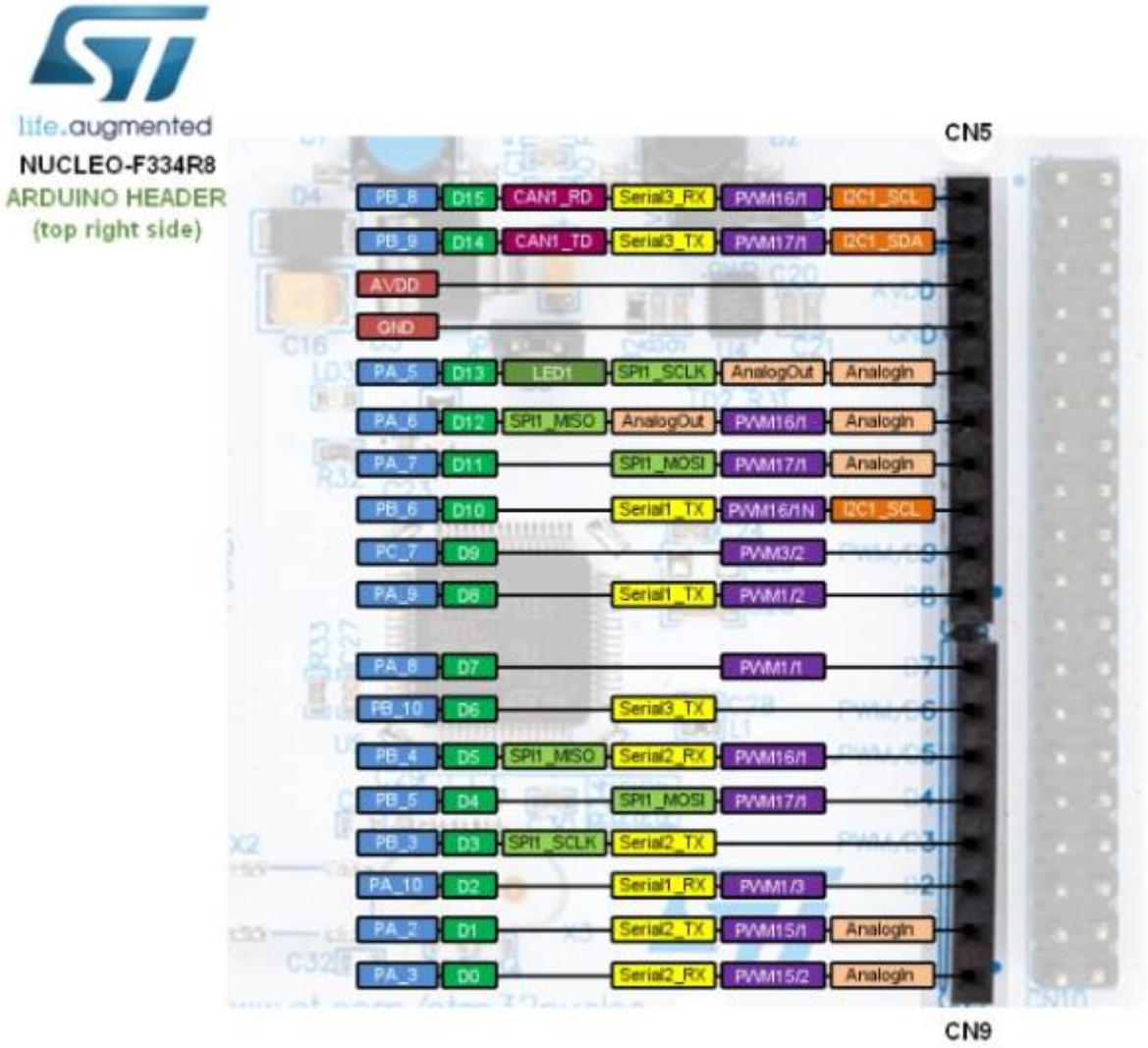
Table 26. ST morpho connector on NUCLEO-F072RB, NUCLEO-F091RC, NUCLEO-F303RE, NUCLEO-F334R8

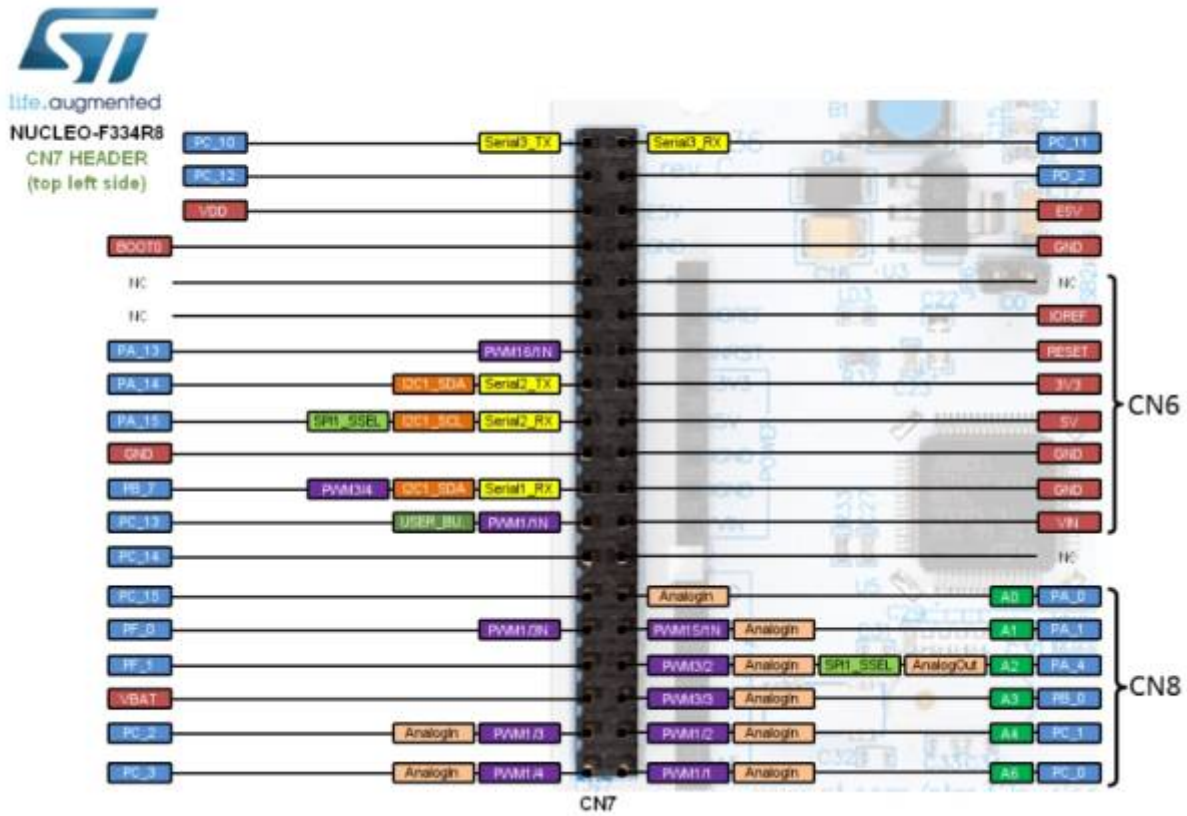
CN7 odd pins		CN7 even pins		CN10 odd pins		CN10 even pins	
Pin	Name	Name	Pin	Pin	Name	Name	Pin
1	PC10	PC11	2	1	PC9	PC8	2
3	PC12	PD2	4	3	PB8	PC6	4
5	VDD	E5V	6	5	PB9	PC5	6
7	BOOT0[(1)(2)]	GND	8	7	AVDD	U5V[(3)]	8
9	-	-	10	9	GND	-	10
11	-	IOREF	12	11	PA5	PA12	12
13	PA13[(4)]	RESET	14	13	PA6	PA11	14
15	PA14[(4)]	+3.3V	16	15	PA7	PB12	16
17	PA15	+5V	18	17	PB6	PB11	18
19	GND	GND	20	19	PC7	GND	20
21	PB7	GND	22	21	PA9	PB2	22
23	PC13	VIN	24	23	PA8	PB1	24
25	PC14	-	26	25	PB10	PB15	26
27	PC15	PA0	28	27	PB4	PB14	28
29	PF0	PA1	30	29	PB5	PB13	30
31	PF1	PA4	32	31	PB3	AGND	32
33	VBAT	PB0	34	33	PA10	PC4	34
35	PC2	PC1 or PB9[(5)]	36	35	PA2	-	36
37	PC3	PC0 or PB8[(5)]	38	37	PA3	-	38

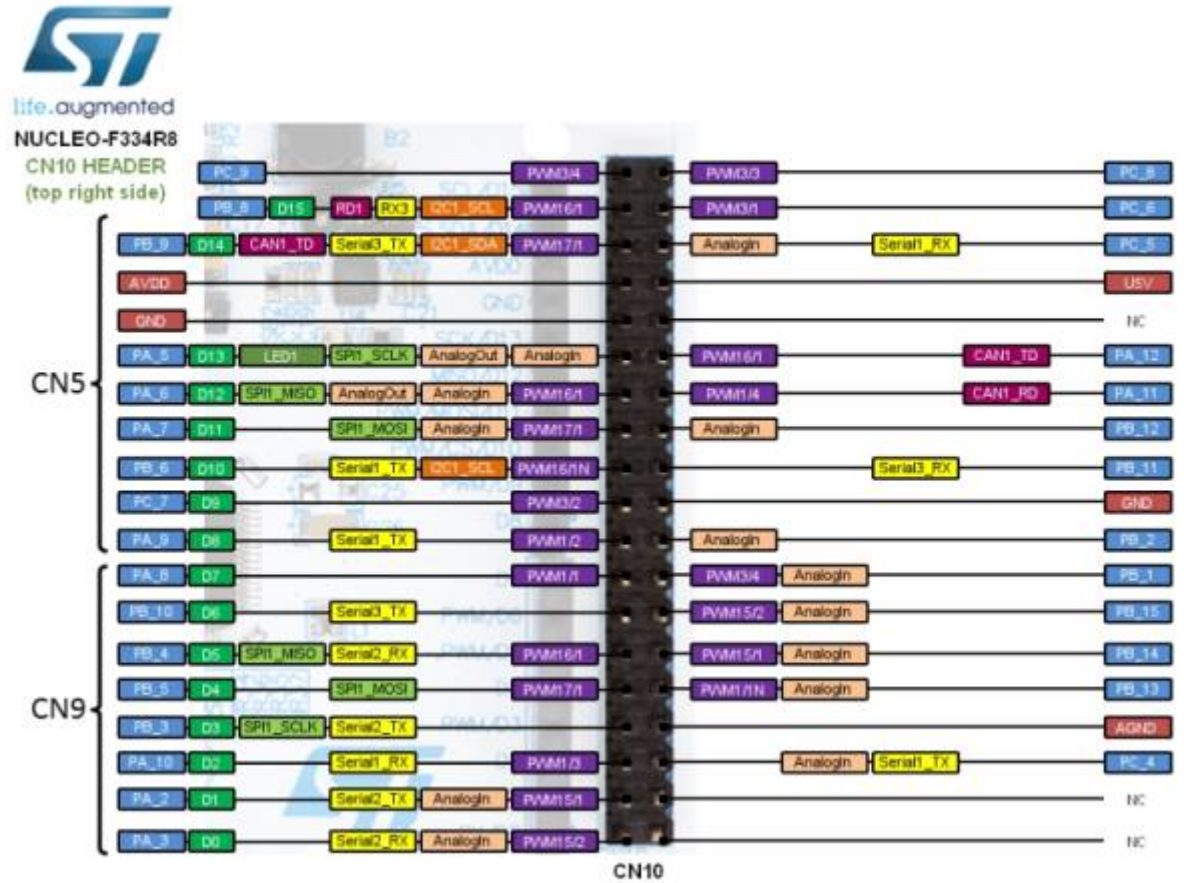
1. The default state of BOOT0 is LOW. It can be set to HIGH when a jumper is on pin5-7 of CN7. Two unused jumpers are available on CN11 and CN12 (bottom side of the board).
2. CN7 pin 7 (BOOT0) can be configured by engineering byte as PF11 on NUCLEO-F091RC.
3. U5V is 5 V power from ST-LINK/V2-1 USB connector and it rises before +5V.
4. PA13 and PA14 share with SWD signals connected to ST-LINK/V2-1, it is not recommended to use them as IO pins if the ST-LINK part is not cut.
5. Refer to *Table 10: Solder bridges* for details.

Numérotation des broches et fonctionnalités


NUCLEO-F334R8
ARDUINO HEADER
(top left side)
CN6
IOREF
RESET
3V3
5V
GND
GND
VIN
AnalogIn
A0
PA_0
AnalogIn
PWM15/1N
A1
PA_1
AnalogIn
PWM3/2
AnalogOut
SPI1_SSEL
A2
PA_4
AnalogIn
PWM3/3
A3
PB_0
AnalogIn
PWM1/2
A4
PC_1
AnalogIn
PWM1/1
A5
PC_0
CN8
www.st.com/stm32nucleo


NUCLEO-F334R8
ARDUINO HEADER
(top right side)
CN5
PB_8 D15 CAN1_RD Serial3_RX PWM16/1 I2C1_SCL
PB_9 D14 CAN1_TD Serial3_TX PWM17/1 I2C1_SDA
AVDD
GND
PA_5 D13 LED1 SPI1_SCLK AnalogOut AnalogIn
PA_6 D12 SPI1_MISO AnalogOut PWM16/1 AnalogIn
PA_7 D11 SPI1_MOSI PWM17/1 AnalogIn
PB_6 D10 Serial1_TX PWM16/1N I2C1_SCL
PC_7 D9 PWM3/2
PA_9 D8 Serial1_TX PWM1/2
PA_8 D7 PWM1/1
PB_10 D6 Serial3_TX
PB_4 D5 SPI1_MISO Serial2_RX PWM16/1
PB_5 D4 SPI1_MOSI PWM17/1
PB_3 D3 SPI1_SCLK Serial2_TX
PA_10 D2 Serial1_RX PWM1/3
PA_2 D1 Serial2_TX PWM15/1 AnalogIn
PA_3 D0 Serial2_RX PWM15/2 AnalogIn
CN9





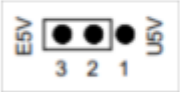
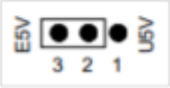
/!\ Dans le programme il faut spécifier uniquement les noms de broches inscrites en Bleu/blanc dans les figures ci-dessus.

Alimentation de la carte

L'alimentation électrique est fournie soit par le PC hôte via le câble USB, soit par une source externe.

source externe : broches d'alimentation VIN (de 7 V à 12 V), E5V (5 V) ou +3,3V sur CN6 ou CN7. Dans le cas où VIN, E5V ou +3.3V est utilisé pour alimenter la carte STM32 Nucleo, l'utilisation d'un bloc d'alimentation externe ou d'un équipement auxiliaire est nécessaire.

d'alimentation externe ou d'un équipement auxiliaire, cette source d'alimentation doit être conforme à la norme EN-60950-1 : 2006+A11/2009, et doit être de type Safety Extra Low Voltage (SELV) avec une capacité de puissance limitée.

Alimentation	Tension alim	JP5	Pin
via port USB			
externe	7 à 12V		CN6 pin 8 = CN7 pin 24 Et GND
externe	5V		CN7 pin 6 Et GND

Exemple programme “faire clignoter la led intégrée”

exempleLed.ino

```
/*
  Blink without Delay

  Turns on and off a light emitting diode (LED) connected to a digital
pin,
  without using the delay() function. This means that other code can
run at the
  same time without being interrupted by the LED code.

  The circuit:
  - Use the onboard LED.
  - Note: Most Arduinos have an on-board LED you can control. On the
UNO, MEGA
    and ZERO it is attached to digital pin 13, on MKR1000 on pin 6.
LED_BUILTIN
    is set to the correct LED pin independent of which board is used.
    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
```

```
  created 2005
  by David A. Mellis
  modified 8 Feb 2010
  by Paul Stoffregen
  modified 11 Nov 2013
  by Scott Fitzgerald
  modified 9 Jan 2017
  by Arturo Guadalupi

  This example code is in the public domain.

  https://www.arduino.cc/en/Tutorial/BuiltInExamples/BlinkWithoutDelay
*/

// constants won't change. Used here to set a pin number:
const int ledPin = PA5;  // the number of the LED pin OR PA_5

// Variables will change:
int ledState = LOW;  // ledState used to set the LED

// Generally, you should use "unsigned long" for variables that hold
time
// The value will quickly become too large for an int to store
unsigned long previousMillis = 0;  // will store last time LED was
updated

// constants won't change:
const long interval = 1000;  // interval at which to blink
(milliseconds)

void setup() {
  // set the digital pin as output:
  pinMode(ledPin, OUTPUT);
}

void loop() {
  // here is where you'd put code that needs to be running all the
time.

  // check to see if it's time to blink the LED; that is, if the
difference
  // between the current time and last time you blinked the LED is
bigger than
  // the interval at which you want to blink the LED.
  unsigned long currentMillis = millis();

  if (currentMillis - previousMillis >= interval) {
    // save the last time you blinked the LED
    previousMillis = currentMillis;

    // if the LED is off turn it on and vice-versa:
```

```
    if (ledState == LOW) {
      ledState = HIGH;
    } else {
      ledState = LOW;
    }

    // set the LED with the ledState of the variable:
    digitalWrite(ledPin, ledState);
  }
}
```

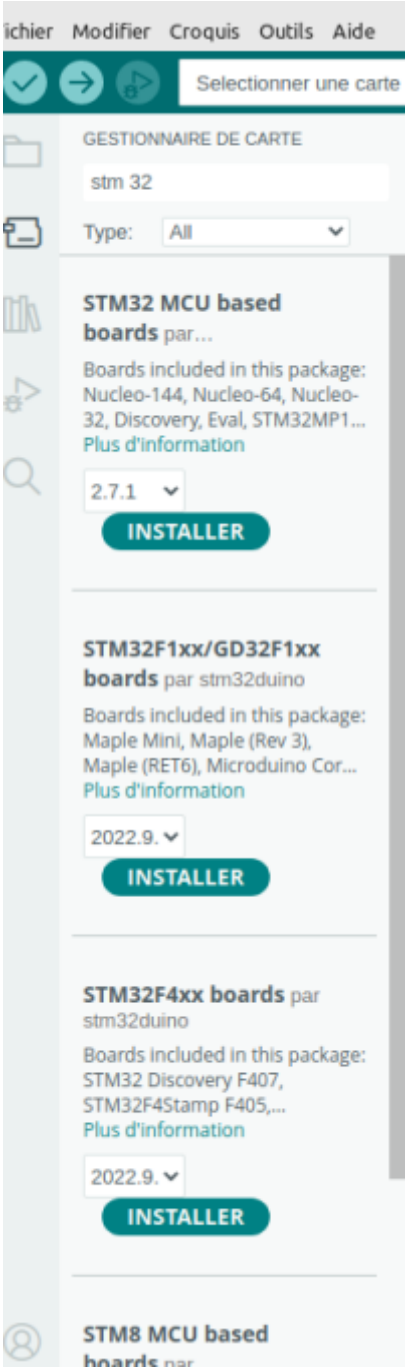
A SUIVRE

Debuter avec le STM32 144 NUCLEO-F429ZI

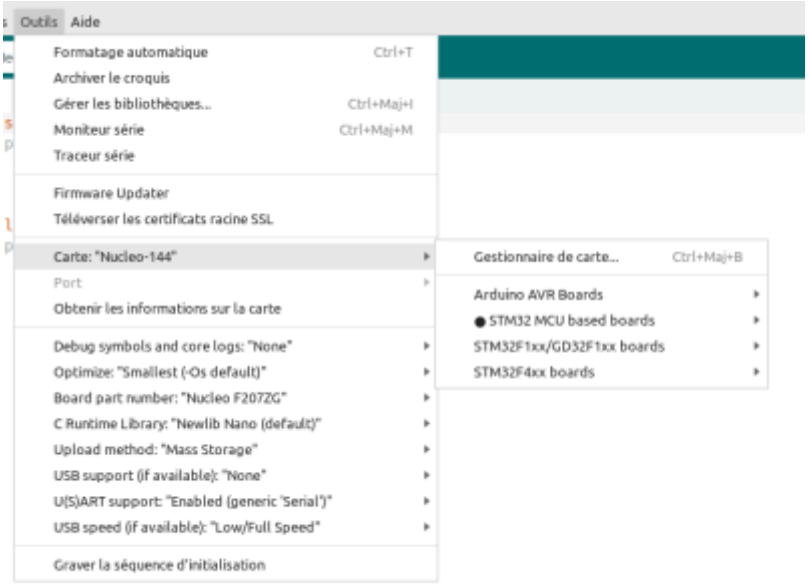
-1-Dans l IDE arduino version 2.x.x, inserer les deux lignes suivantes :

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

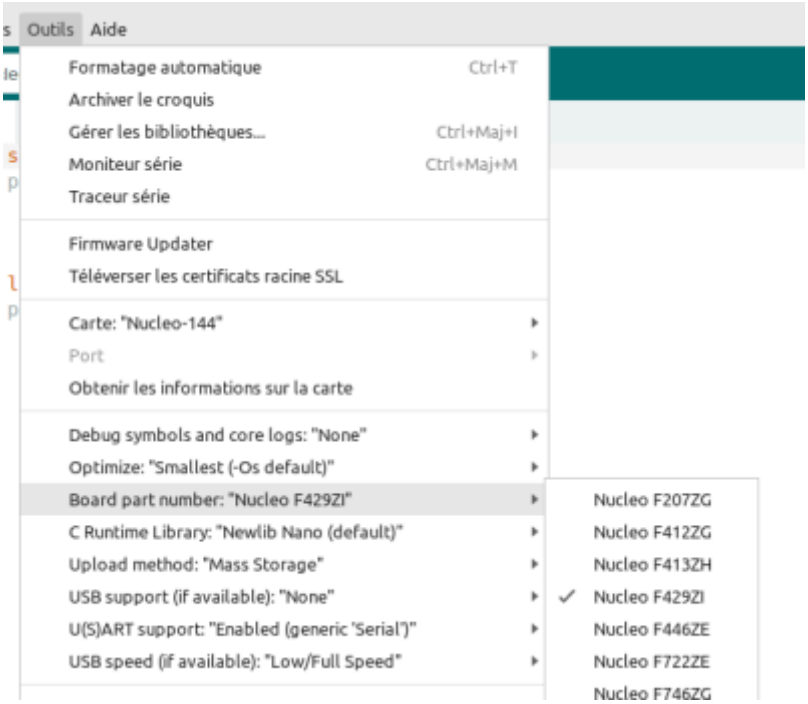
-2- Dans le gestionaire de carte installer toutes les cartes STM32



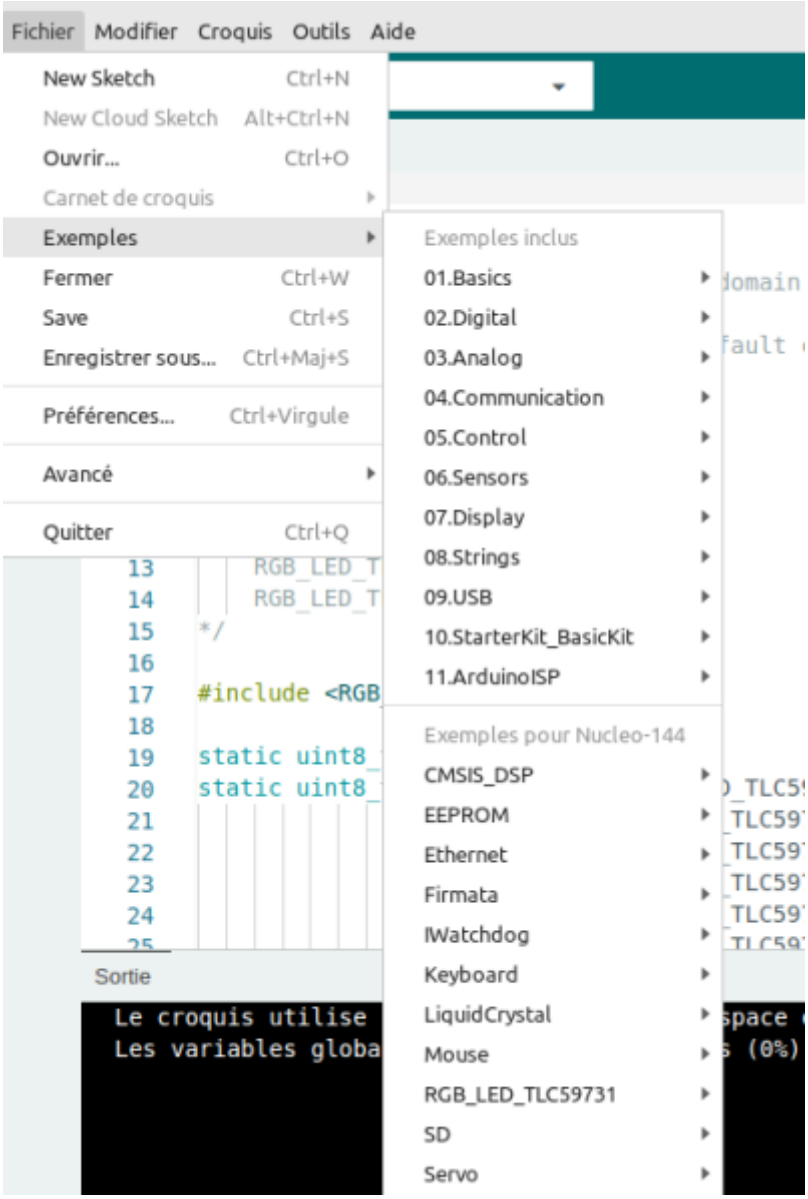
-3- Dans “Outils” –> “Carte” choisir “STM32 MCU Based Board” et **“Nucleo 144”**



-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”



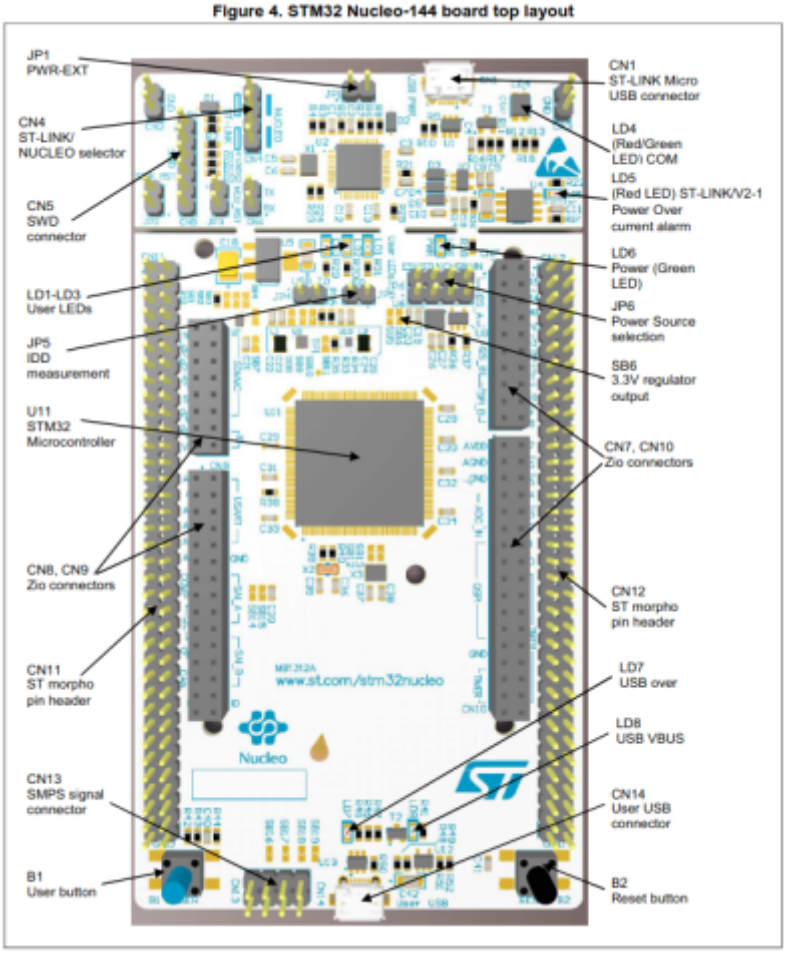
comment_programmer_et_deboguer_le_stm32_a_l_aide_de_l_ardui._led.pdf

Debuter avec Un STM32 144 NUCLEO-L4P5ZG

Carte Pour l'instant … non comparible 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).

Figure 4. STM32 Nucleo-144 board top layout



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

Last update: **2024/01/23 10:49**