

جامعة حلب  
كلية الهندسة الكهربائية والإلكترونية  
قسم هندسة التحكم والأتمتة  
مخبر التحكم

## مقرر المتحكمات المصغرة الجلسة الأولى

السنة الرابعة ميكاترونيك

2023/2022

## محتويات الجلسة

- 1- تنصيب البرامج اللازمة STM32CubeProgrammer ، STM32CubeIDE
- 2- تعريف عن المتحكم STM32G0 وأهم المزايا الخاصة به
- 3- تعريف عن البورد المستخدم
- 4- بناء تطبيق إضاءة ليد وإطفائه كل 100msec باستخدام البورد التطويري

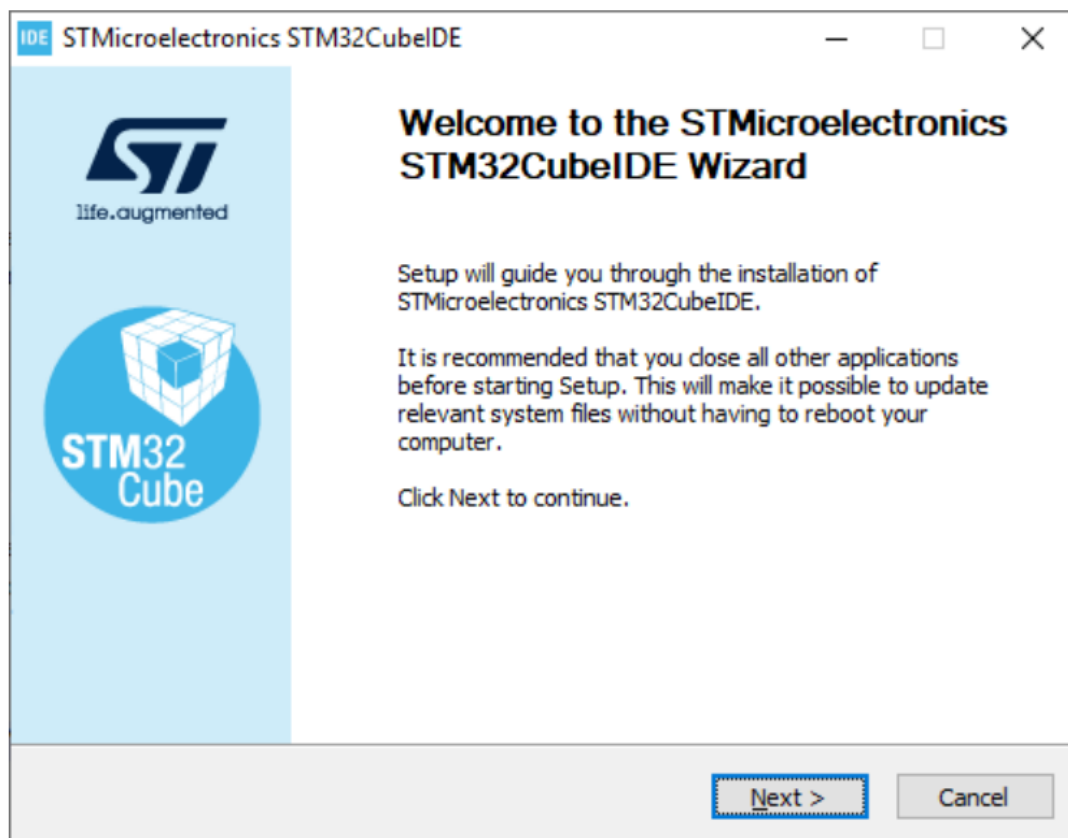
## 1- تنصيب البرامج اللازمة STM32CubeProgrammer ، STM32CubeIDE

ملاحظة: متطلبات بيئة STM32CubeIDE :

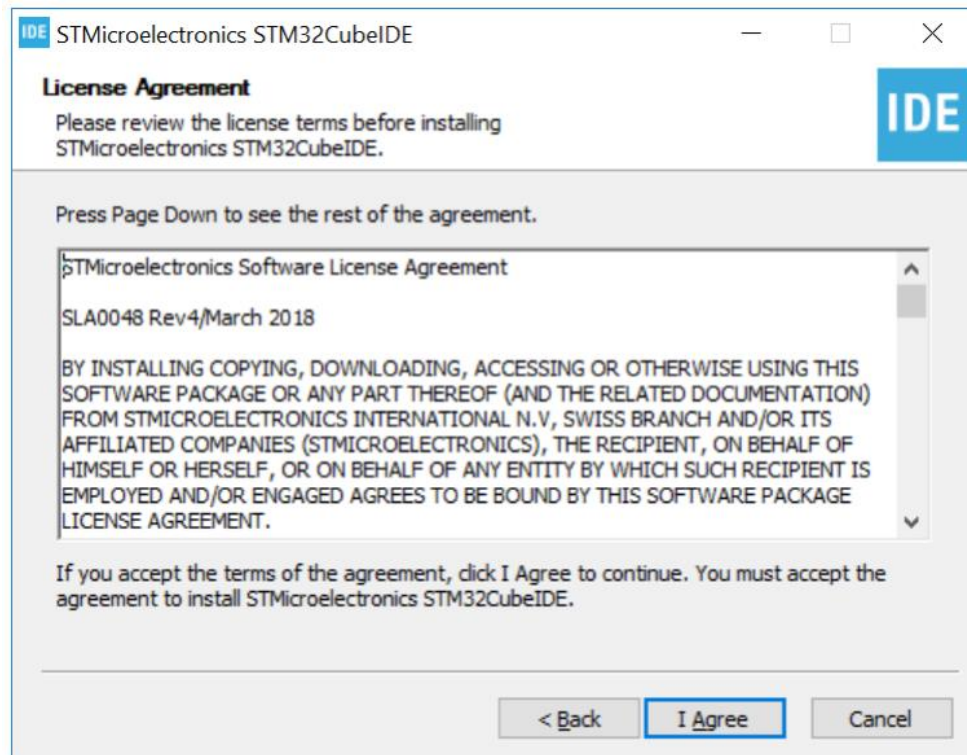
1. نظام تشغيل 64bit حصراً
  2. يمكن تنصيبها على عدة أنظمة تشغيل 10® Windows ، Linux® ، macOS®
  3. مواصفات الحاسب المطلوبة لتنصيب البيئة:
- رامات بحجم 2GB على الأقل ويفضل أن تكون 4GB

في البداية نقوم بتنصيب بيئة STM32CubeIDE وفق الخطوات التالية:

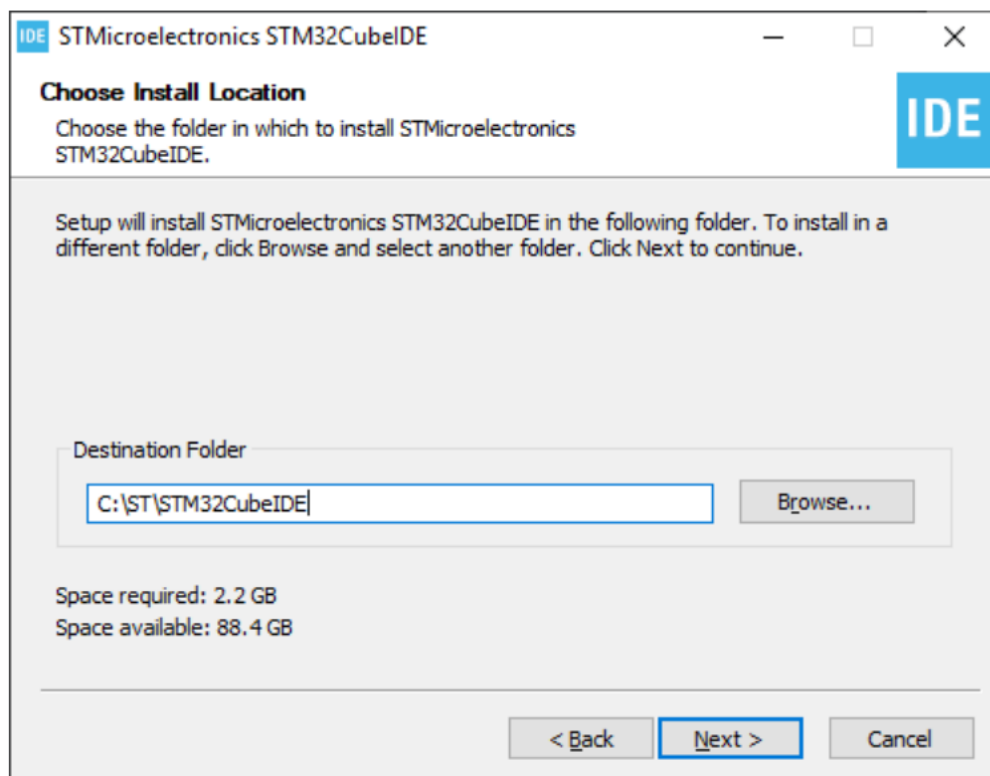
- نقوم بفتح الملف st-stm32cubeide\_VERSION\_ARCHITECTURE.exe فتظهر النافذة الترحيبية



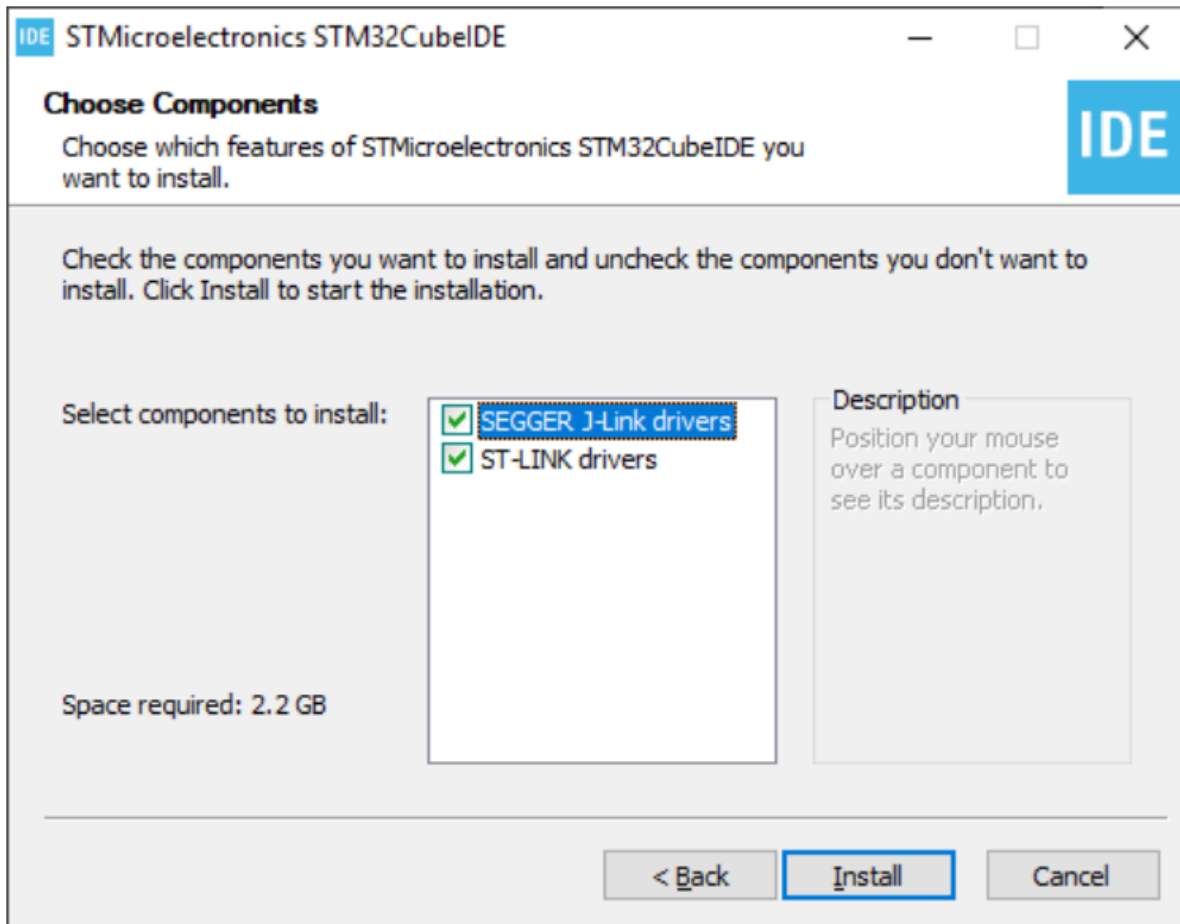
- اضغط Next



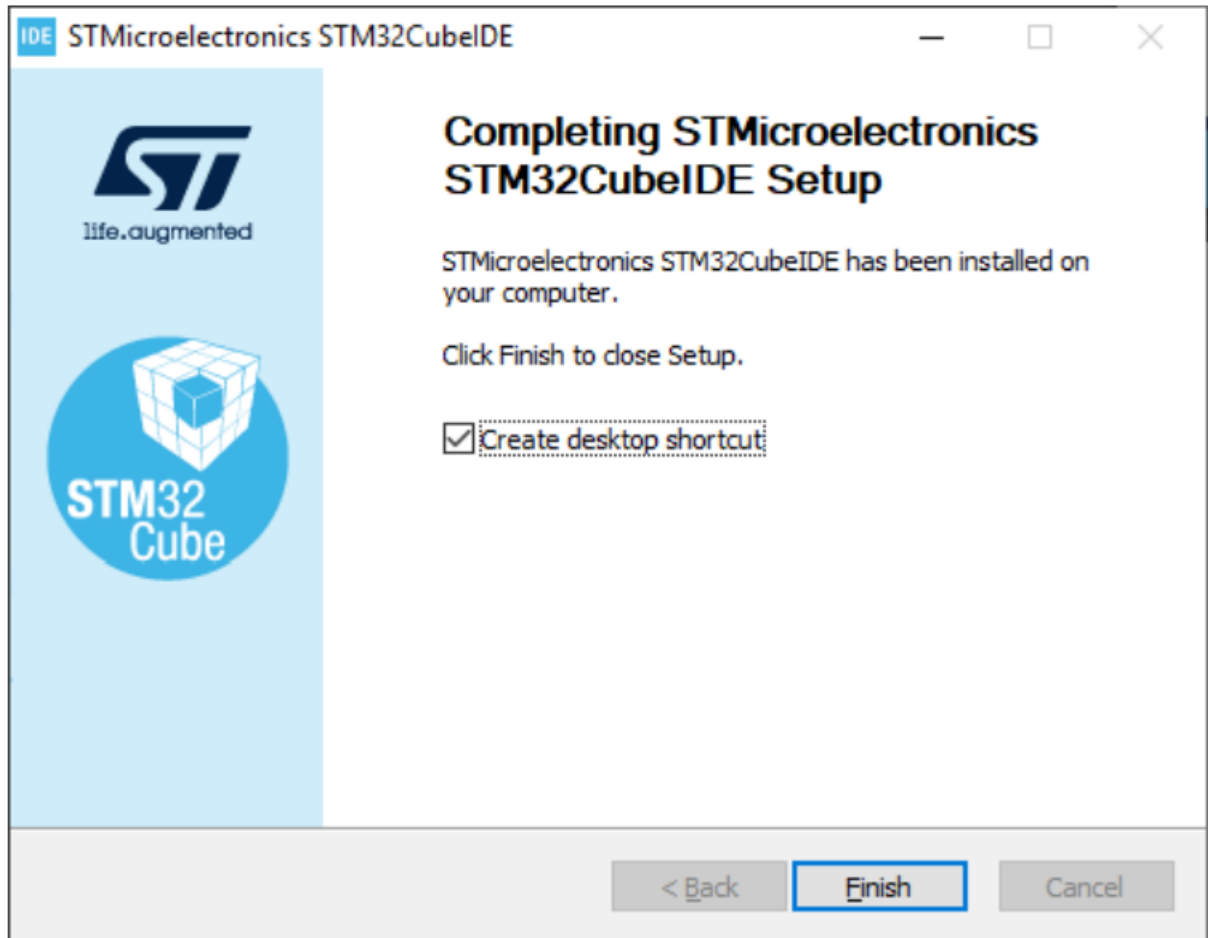
- اختر I agree



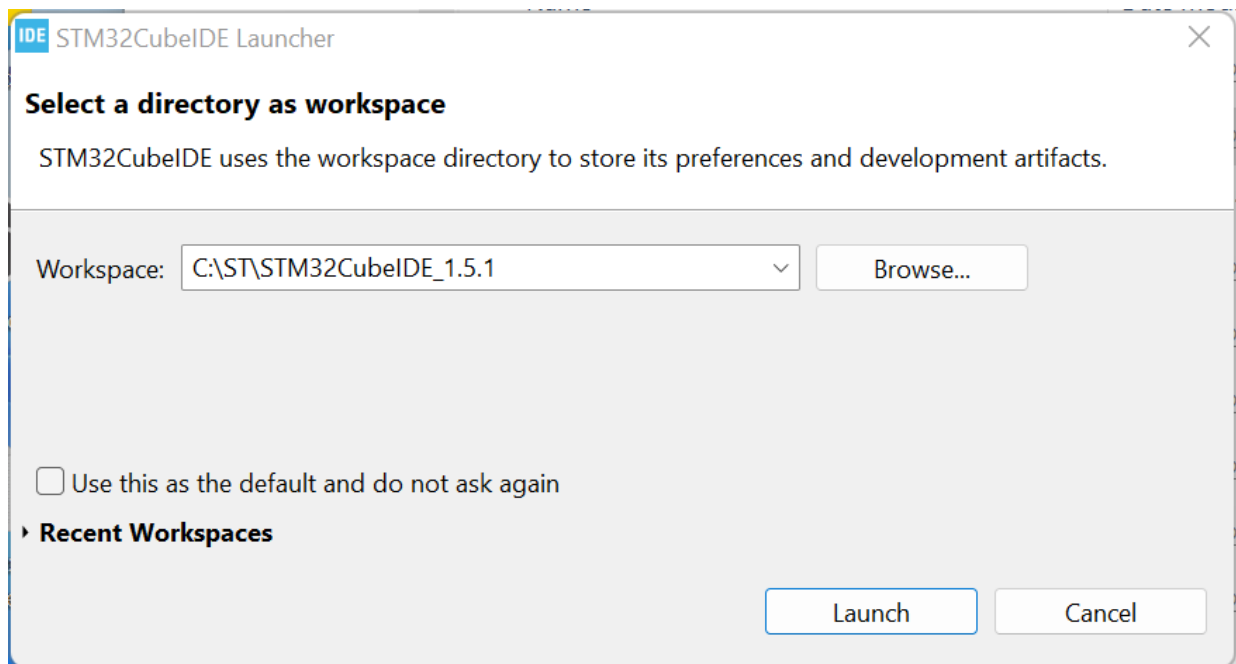
Next -



- نضغط على Install

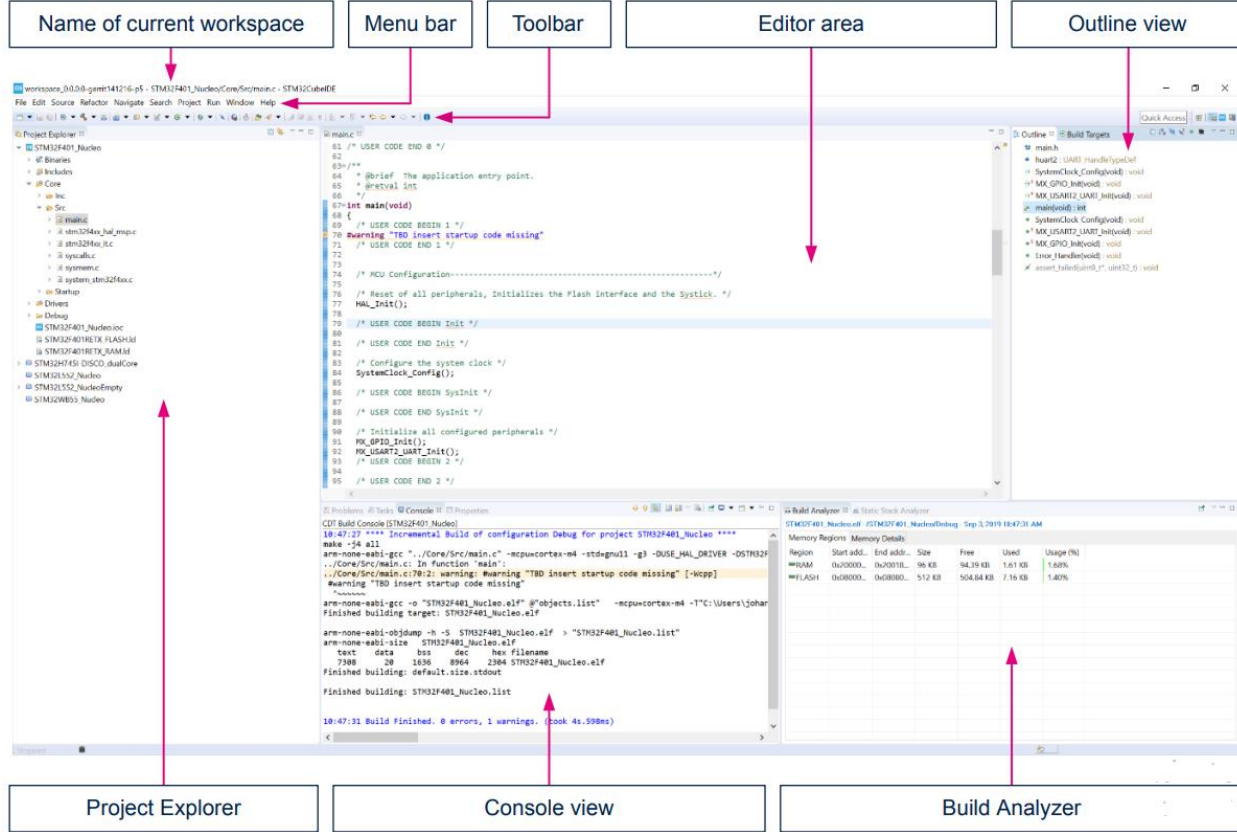


بذلك يكون تم تنصيب بيئة STM32CubeIDE، نقوم بفتح بيئة STM32CubeIDE فتظهر النافذة التالية:

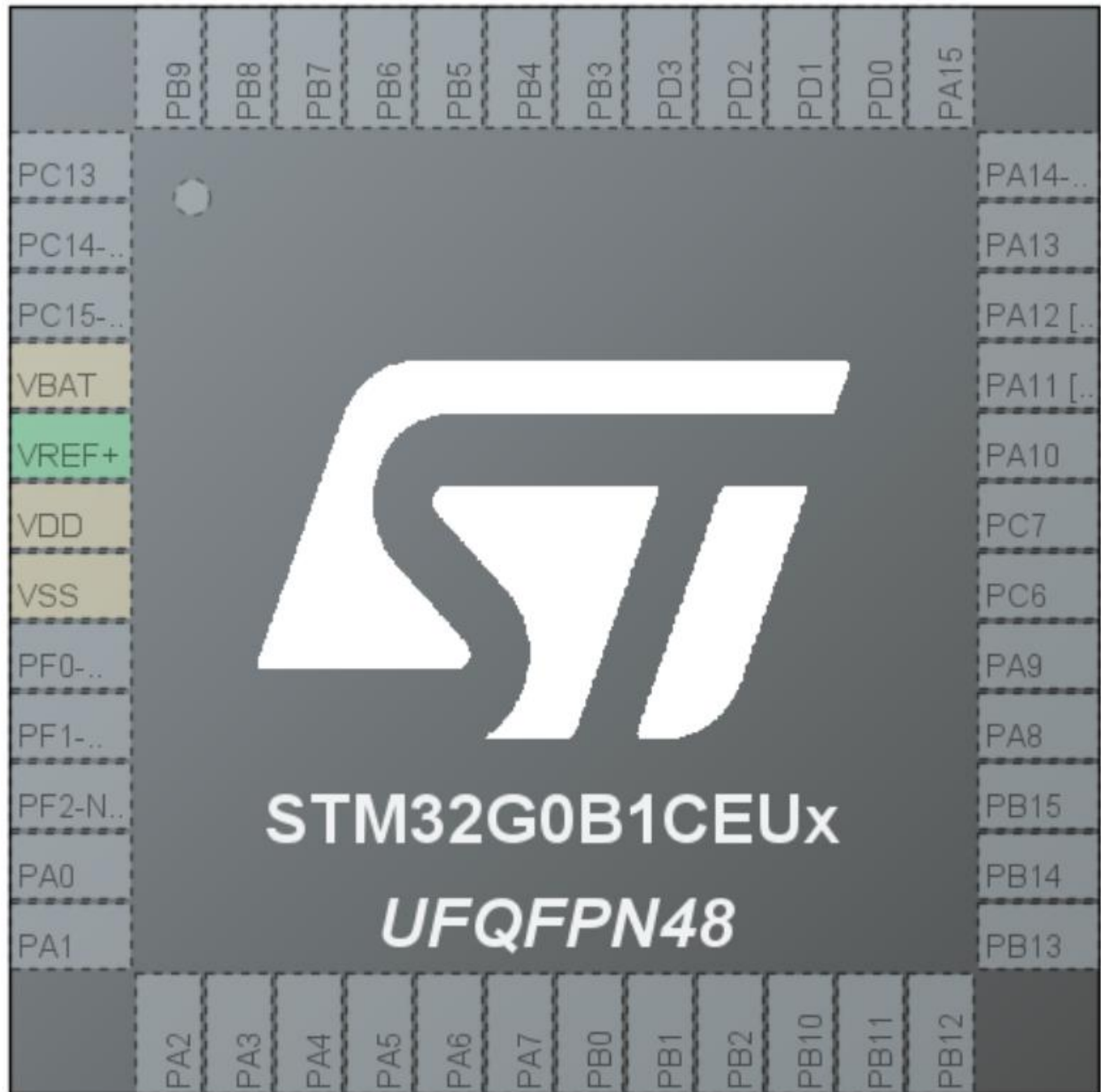


وهنا يطلب منك اختيار مساحة العمل الخاصة بالبيئة والتي ضمنها سيتم حفظ المشاريع والتطبيقات التي ستقوم بإنشائها لاحقاً ويفضل عدم تغيير المسار.

بعد الضغط على زر Launch فإذا قمنا باختيار أحد التطبيقات الموجودة تظهر لنا النافذة التالية:



2- تعريف عن المتحكم STM32G0 وأهم المزايا الخاصة به  
للمتحكم STM32G0 الشكل التالي:



يحتوي متحكم STM32G0 على خمسة منافذ دخل أو خرج رقمية تسمى GPIOA,GPIOB,GPIOC,PIOD,PIOF وكل منها مؤلف من 16 قطب، لكن ليست جميع الأقطاب مستخدمة.

**أهم مزايا المتحكم STM32G0 :**

- Core: Arm® 32-bit Cortex®-M0+ CPU, frequency up to 64 MHz
- -40°C to 85°C/105°C/125°C operating temperature
- Memories

- Up to 512 Kbytes of Flash memory
- 36 Kbytes of SRAM
- Reset and power management
- Voltage range: 1.7 V to 3.6 V
- Low-power modes:
  - Sleep, Stop, Standby, Shutdown
- Clock management
- 4 to 48 MHz crystal oscillator
- 32 kHz crystal oscillator with calibration
- Internal 16 MHz RC with PLL option ( $\pm 1\%$ )
- Internal 32 kHz RC oscillator ( $\pm 5\%$ )
- Up to 60 fast I/Os
- All mappable on external interrupt vectors
- Multiple 5 V-tolerant I/Os
- 7-channel DMA controller with flexible mapping
- 12-bit, 0.4  $\mu$ s ADC (up to 16 ext. channels)
- Two 12-bit DACs, low-power sample-and-hold
- Two fast low-power analog comparators
- 14 timers (two 128 MHz capable)

#### Communication interfaces

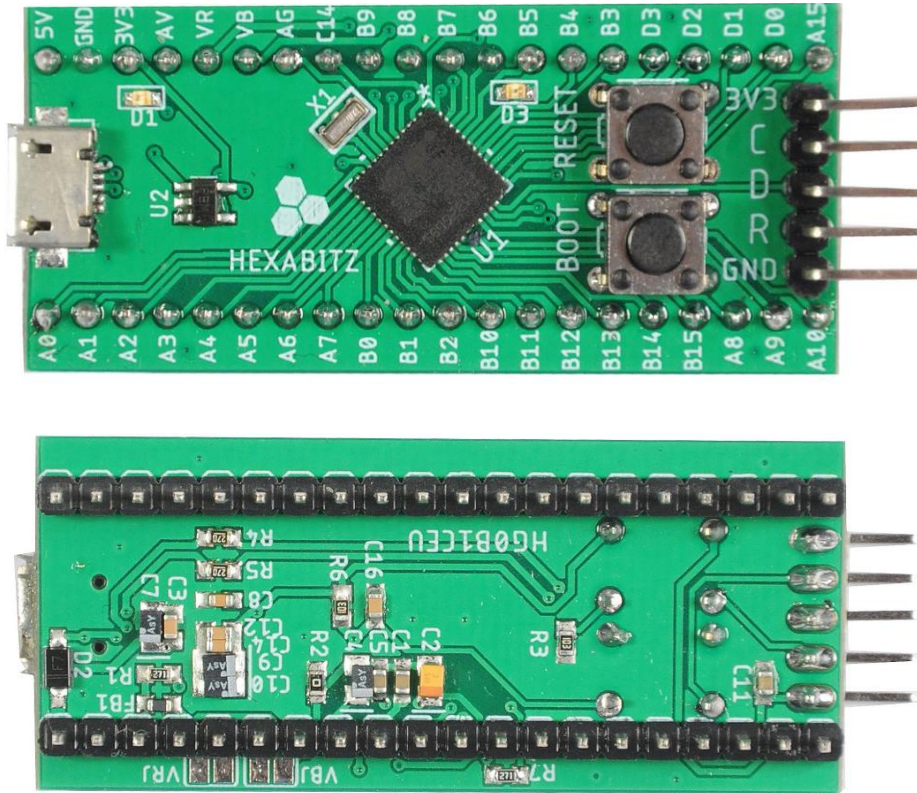
- Two I2C-bus
- Four USARTs with master/slave
- One low-power UART



– Two SPIs (32 Mbit/s) with 4- to 16-bit

### 3- تعريف عن البورد المستخدم:

بورد STM32G0B1CEU6N تم تصميمها للتعامل مع معالج STM32G0B1CEU6N حيث تحتوي على المعالج المُصغّر STM32G0B1CEU6N كمكوّن أساسي وتم توصيل معظم أقطاب المعالج إلى-pin heads للتعامل مع طرفيات هذا المعالج حيث يمكن وضع هذه البورد على test-board لإجراء بعض التجارب كذلك يمكن وضعها على دائرة مطبوعة بوضع البصمة المناسبة في ال Layout حيث أنّ البُعد مابين صقّي ال Pin-heads يساوي البعد بين أرجل معالج atmega16 أو 32 أو بورد-Arduino Nano .



#### أ - التغذية :

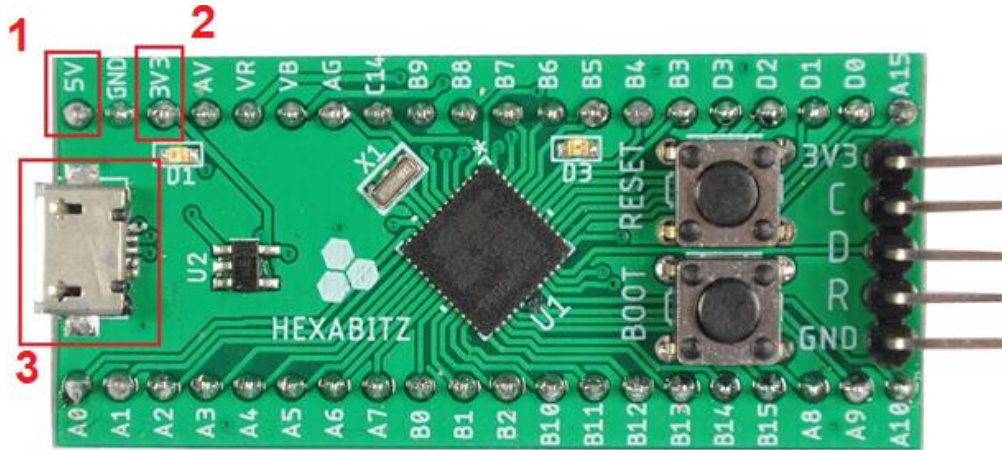
-يمكن تغذية البورد بعدة طرق :

- 1 - وصل 5 فولت إلى القطب V5
  - 2 - وصل 3.3 فولت إلى القطب V33 الموجود على البورد
  - 3 - وصل تغذية USB ب 5 فولت عبر منفذ ال USB-Micro الموجود على البورد
- ملاحظة : يجب استخدام طريقة واحدة فقط لتغذية البورد ، فمثلا يجب عدم توصيل تغذية 3.3 فولت بنفس الوقت مع تغذية قطب ال 5 فولت أو ال USB-Micro .

-تحتوي البورد على مُنظّم جهد 3.3 فولت ، وبالتالي عند توصيل 5 فولت إلى البورد يمكن الاستفادة من قطب V33 لتغذية دارات خارجيّة ب 3.3 فولت ولكن التيار المُستجَر لا يجب أن يتجاوز الـ 100 مللي أمبير .

- تحتوي البورد على أقطاب تغذية AGND(AG) , A3V3(AV) لتغذية الدارات التشابهيّة ب 3.3 فولت مُرشّحة بملف لكي لا تُؤثّر على عمل المُعالج .

- القطب GND هو نفسه القطب G



- تتيح البورد الوصول المباشر إلى الأقطاب VREF(VR) و VBAT(VB) ، كذلك يوجد SMD\_JUMPER لفصل أو وصل هذه الأقطاب عن مصدر التغذية الأساسي للمعالج .

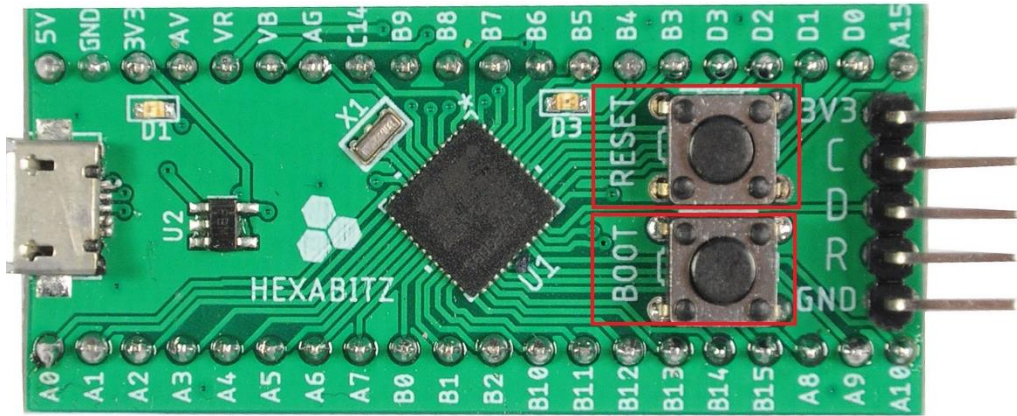
#### ب - البرمجة :

يمكن برمجة المعالج بعدة طرق :

- 1 - عبر مُبرمجة SWD بعد وصل أقطابها إلى البورد (C - D - R) وفي هذه الحالة يمكن رفع الفيرموير إلى المُعالج وعمل Debug .
- 2 - عبر منفذ الـ USB-Micro حيث يدعم مُعالج الـ G0 الدخول إلى وضع البوت عبر الـ USB بالتوصيل إلى الأقطاب المُخصّصة .
- 3- عبر نافذة الـ USART2 عبر الأقطاب PA2,PA3 حيث أنّ PA2 هو الـ TX والـ PA3 هو الـ RX



ملاحظة : الطريقتين رقم (2 و 3) لا يمكن استخدامهما بعد ضبط فيوز  $nBOOT\_SEL\_bit = 0$  ولا يمكن عمل debug باستخدامهما ويجب إدخال المعالج في وضع الـ boot وذلك بالاستمرار بالضغط على زر الـ BOOT أثناء الإقلاع (بفصل ووصل التغذية أو بالضغط على زر الـ RESET).



ج - التعامل مع طرفيات المعالج :

كما هو واضح في الصور فإن أسماء أقطاب المعالج تم طباعتها على البورد بجانب كل Pin-head فمثلا (B3=PB3) وهكذا ...

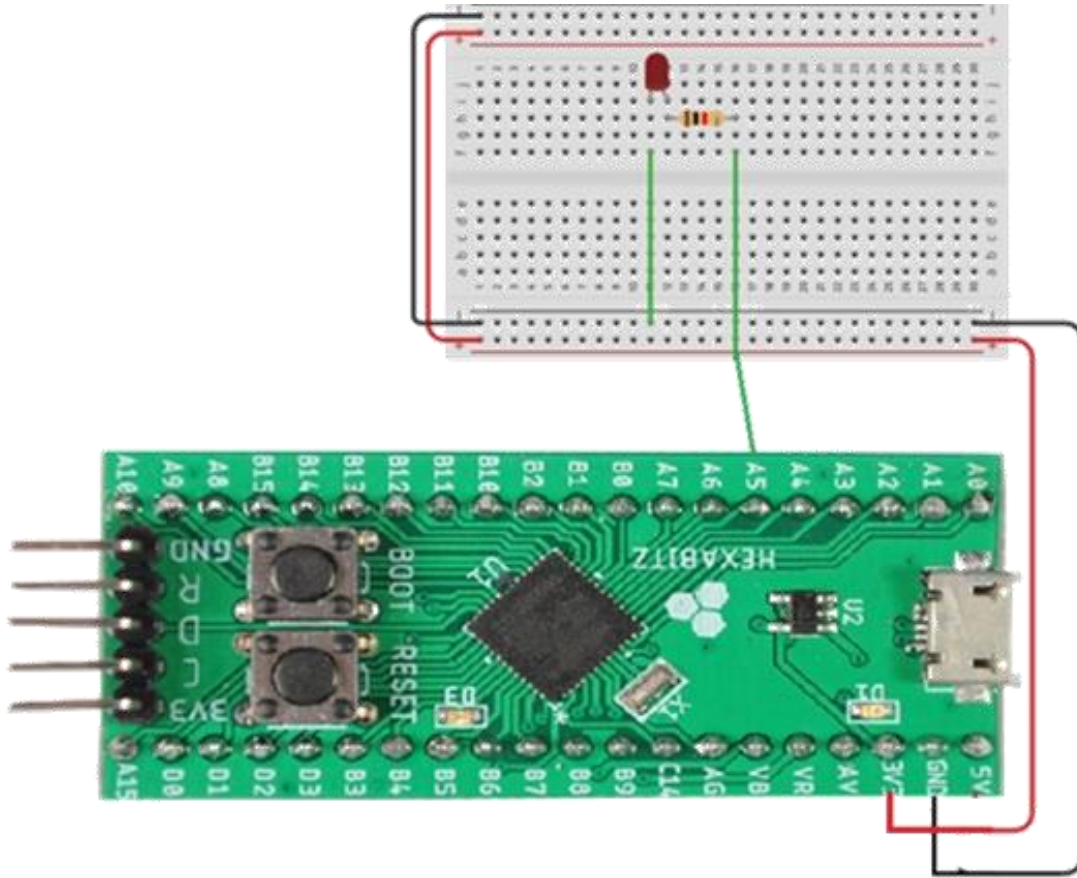
- يوجد USER\_LED على البورد موصول إلى القطب PC13 .

- بإمكانك الاطلاع على الملفات التصميمية للوحة التطويرية من خلال الرابط التالي:

<https://github.com/HexabitzPlatform/HG0B1CEU-Hardware/tree/HG0B1CE>

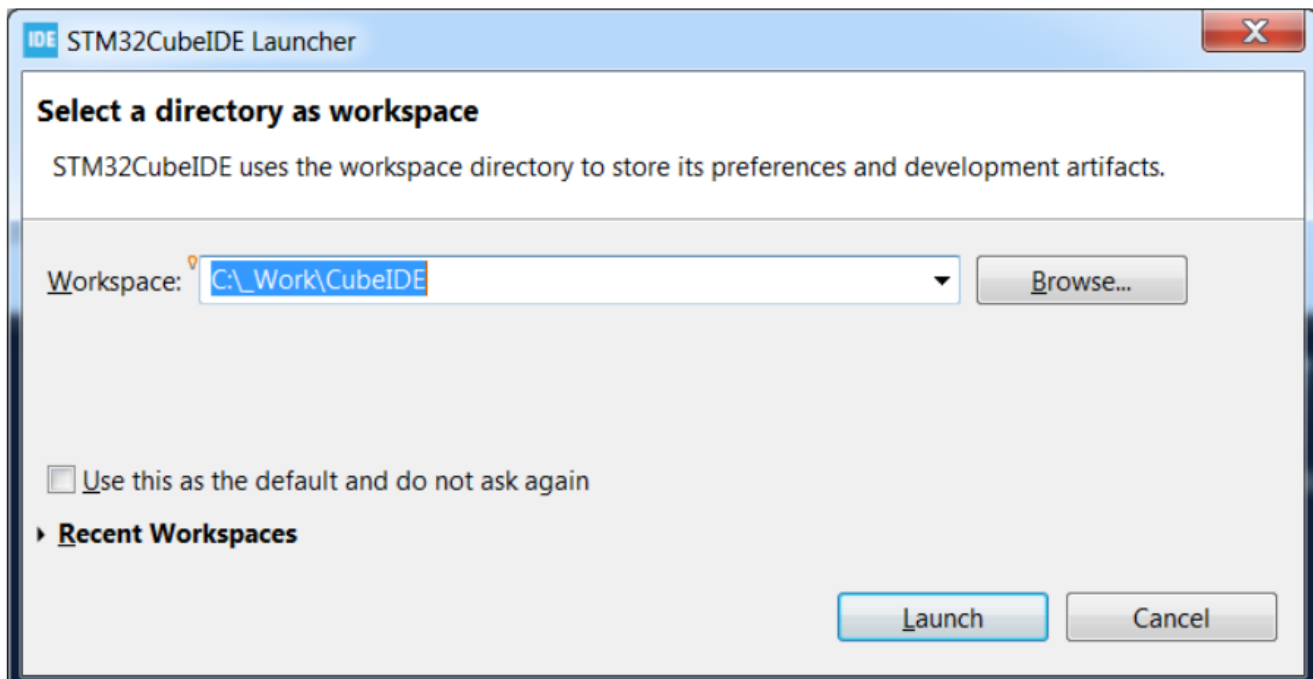
4- بناء تطبيق إضاءة ليد وإطفائه كل 100msec باستخدام البورد التطويري  
سنقوم بتصميم تطبيق يقوم بعمل toggle لليد المتصل بالقطب PA5:



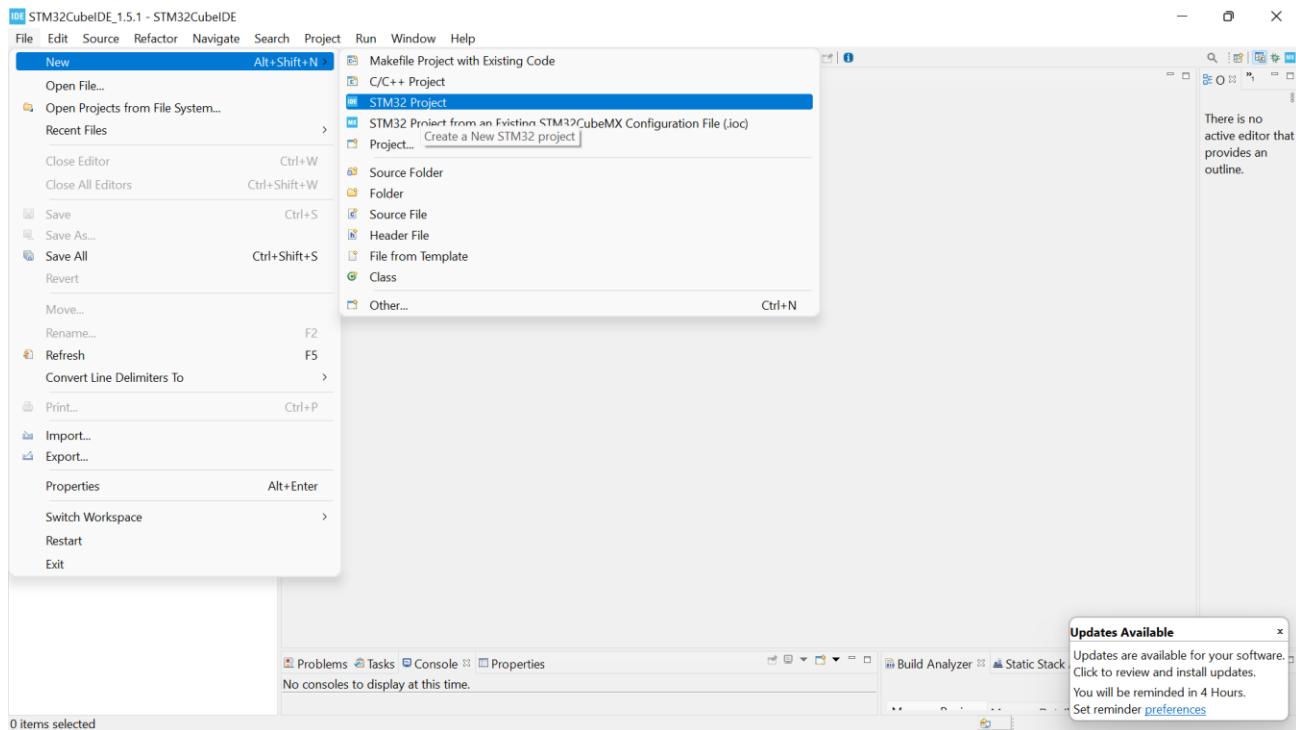


وذلك وفق الخطوات التالية:

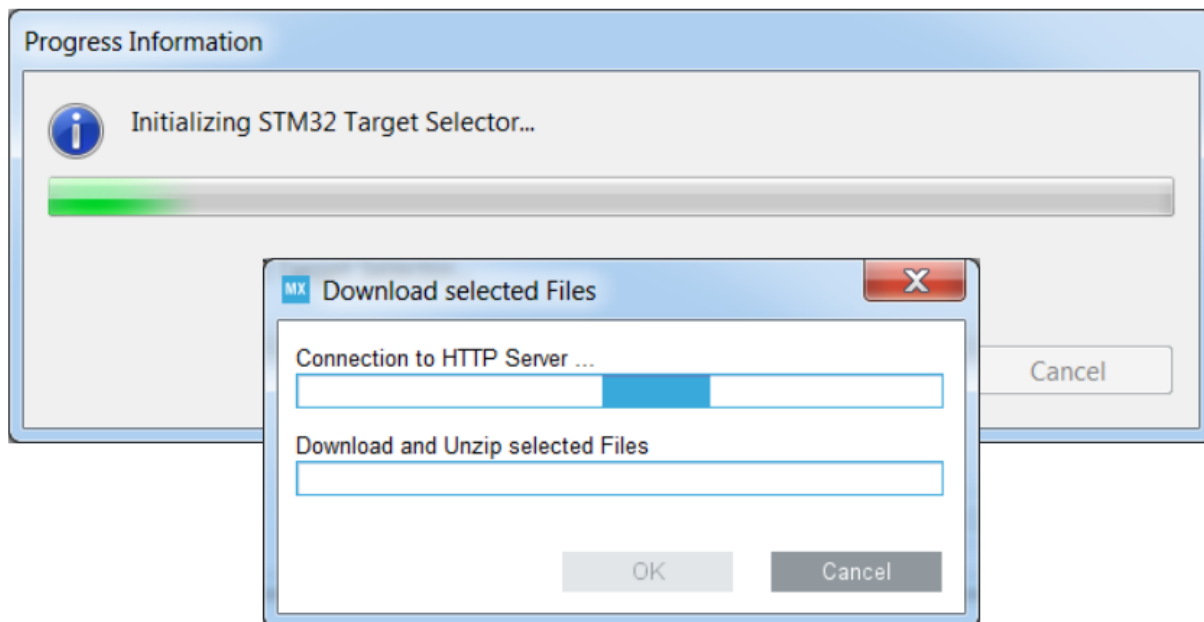
**الخطوة الأولى:** قم بفتح بيئة STM32cubeIDE



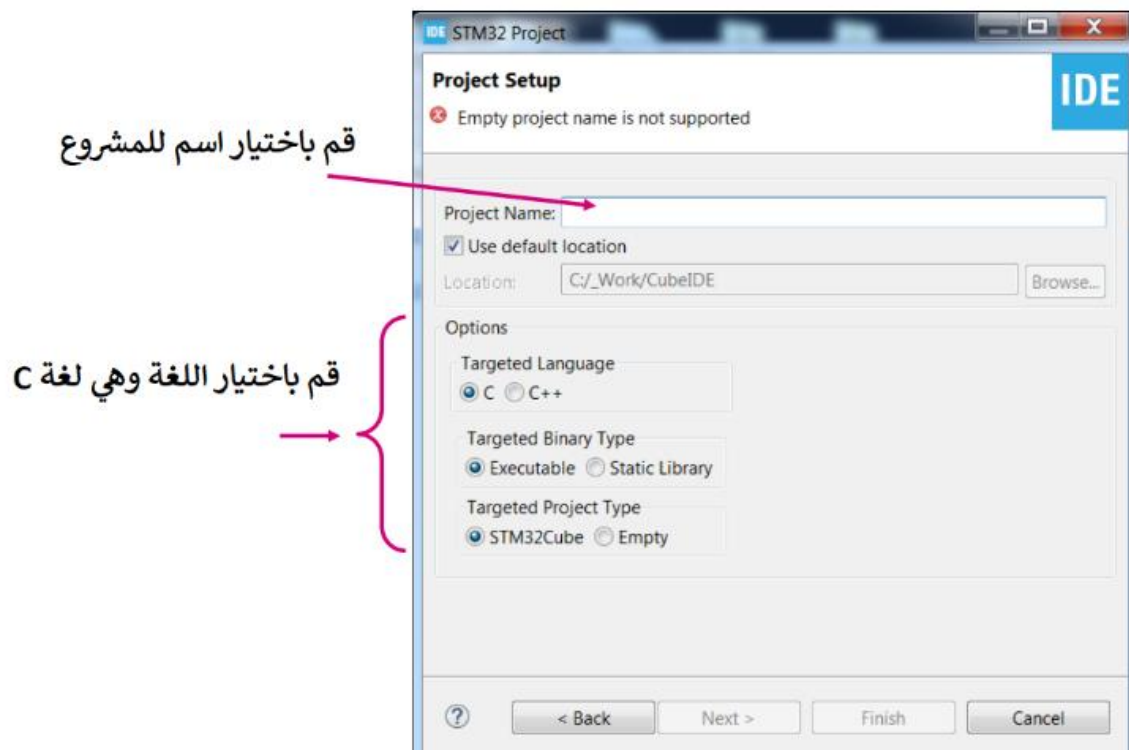
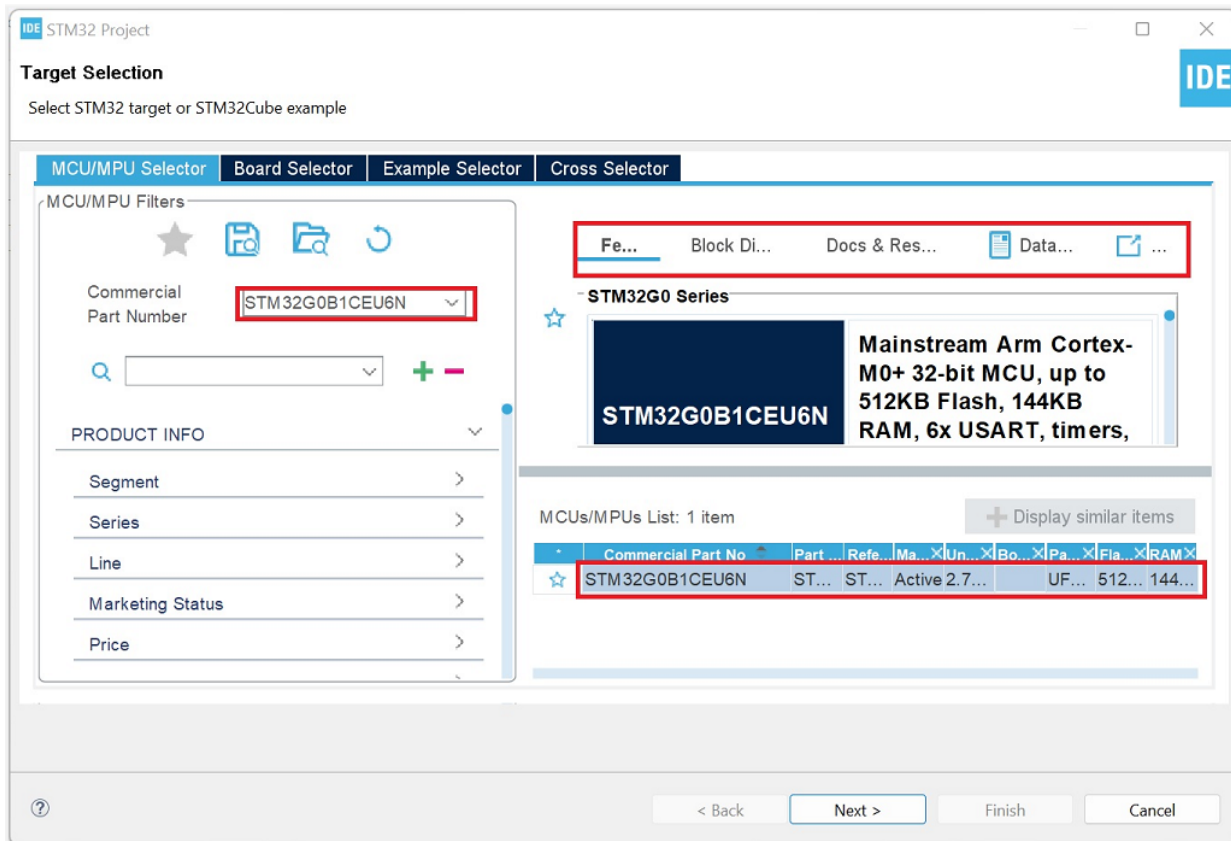
ثم نختار إنشاء مشروع جديد من file new... Stm32project.... كما هو موضح بالشكل التالي:



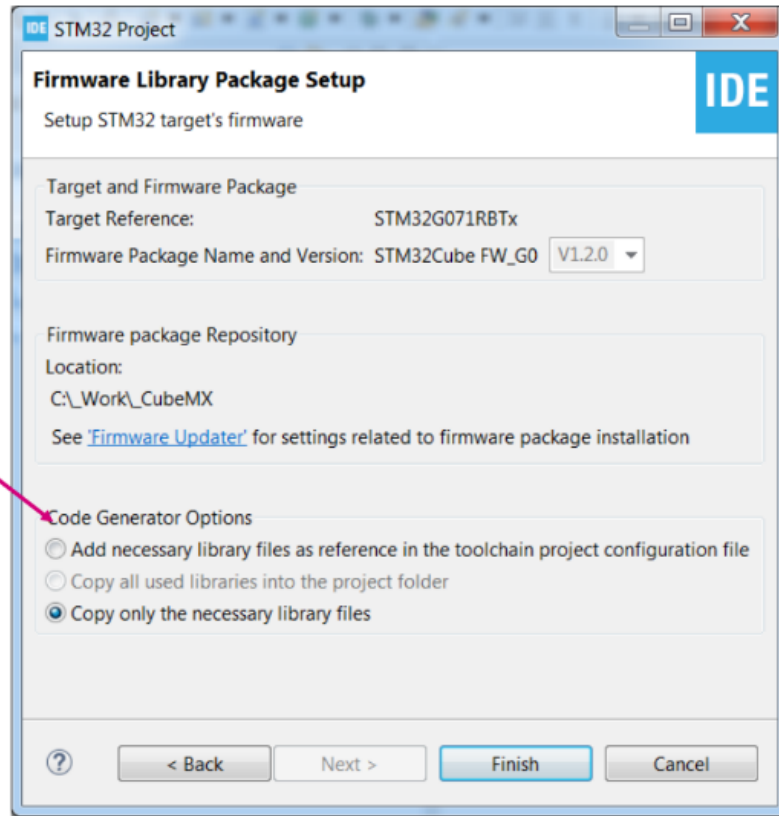
نلاحظ أن البرنامج يبحث عن تحديثات له عبر شبكة الانترنت في حال كان الحاسب متصل بالانترنت:



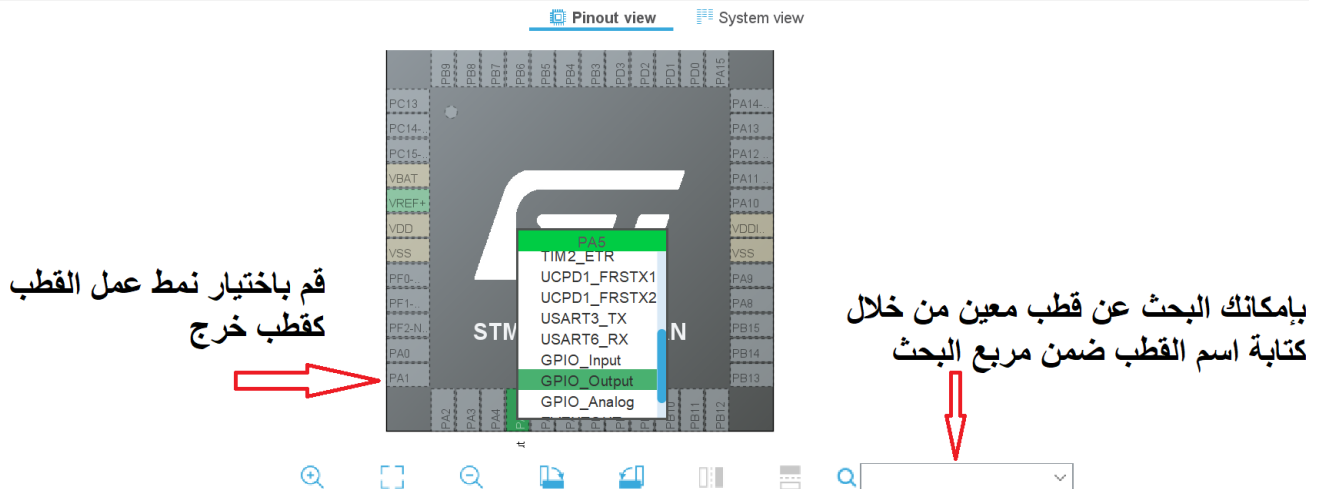
**الخطوة الثانية :** قم باختيار المتحكم من خلال كتابة اسم المتحكم وهو STM32G0B1CEU6N كما هو موضح بالشكل التالي:



اختر إضافة المكتبات الضرورية فقط للمشروع

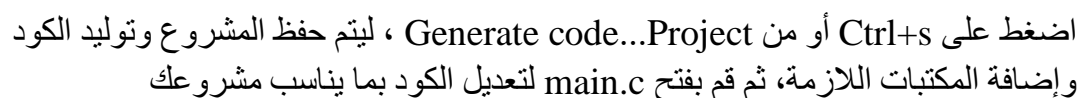
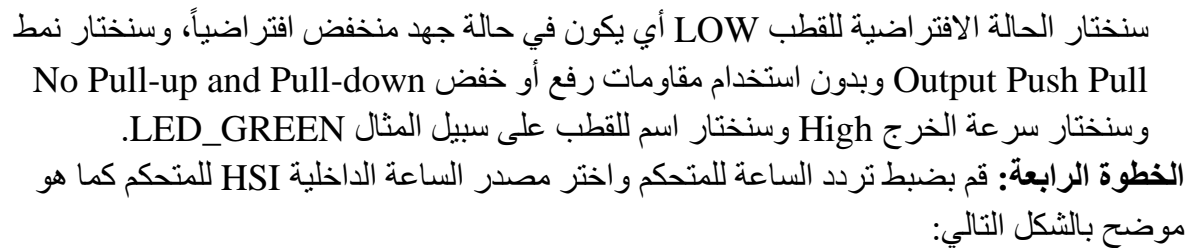


الخطوة الثالثة: قم بتحديد القطب PA5 كخرج كما هو موضح بالشكل التالي:

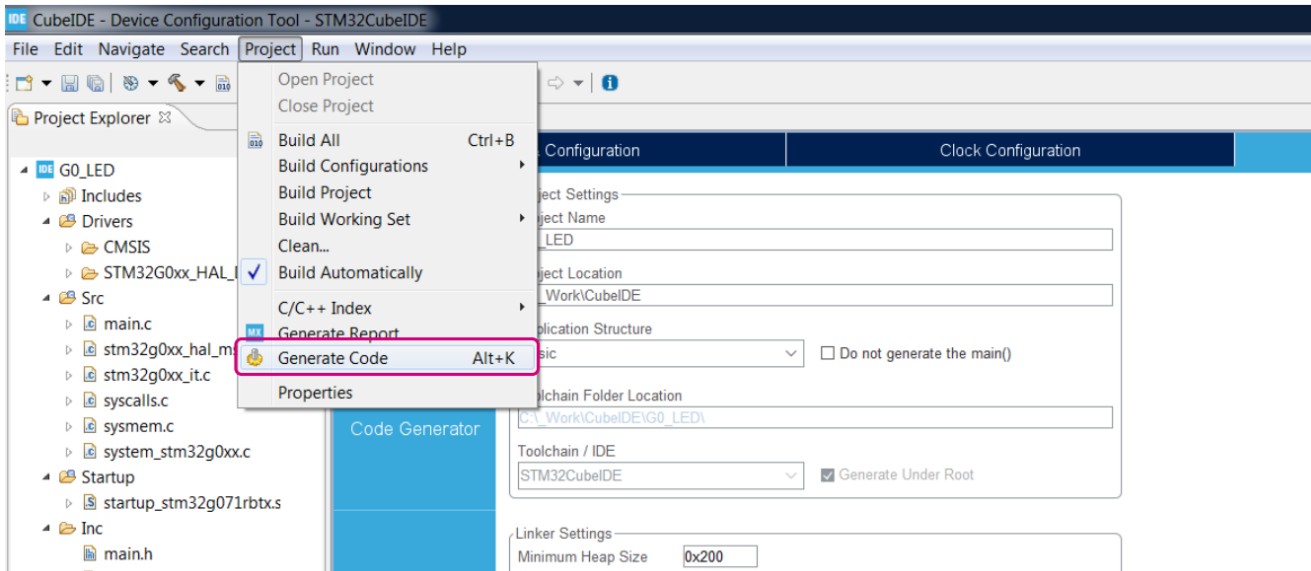


قم باختيار نمط عمل القطب كقطب خرج

بإمكانك البحث عن قطب معين من خلال كتابة اسم القطب ضمن مربع البحث

[illegible]





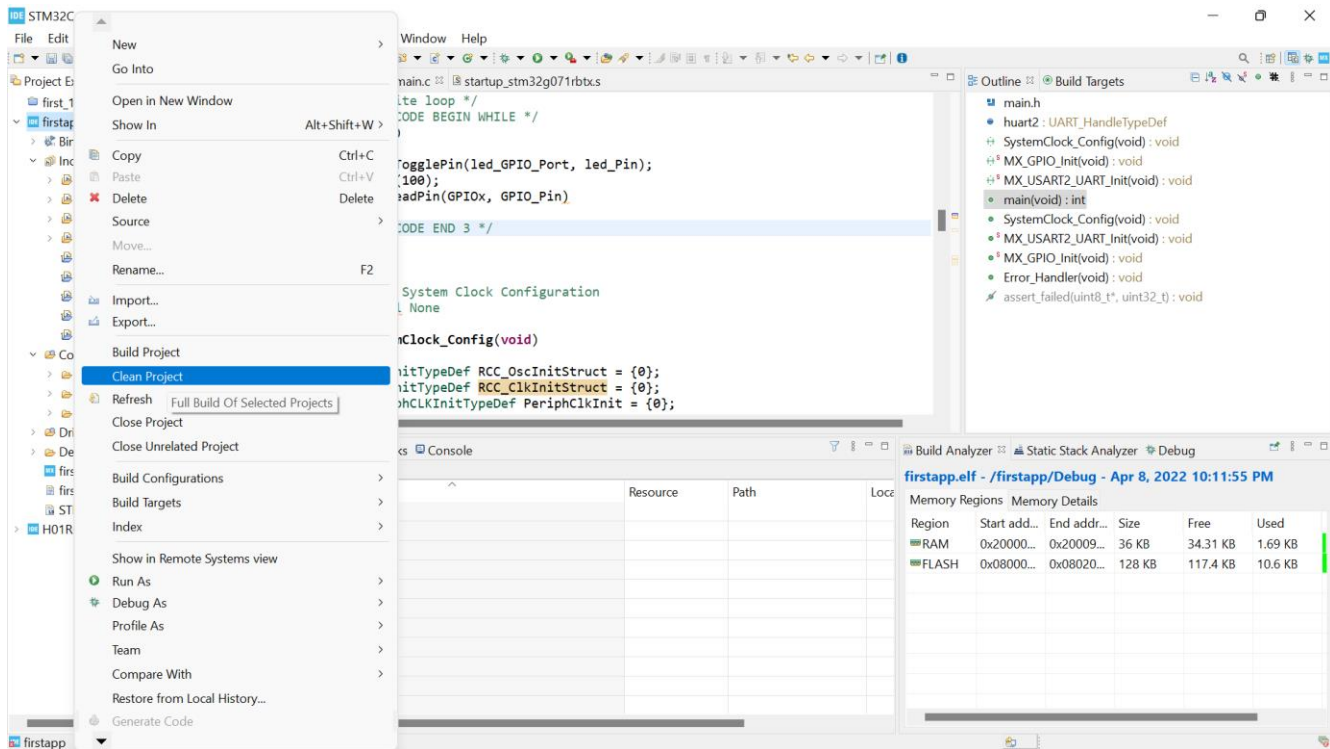
**الخطوة الخامسة:** نقوم بكتابة الكود المناسب كي يضيء الليد الموجود على القطب رقم 5 من المنفذ A  
 100msec 100msec ويطفأ 100msec ويعيد الكرة في كل مرة، يمكنك الاستعانة بـ Ctrl+Space  
 لاستخدام ميزة الإكمال التلقائي للتعليمات ، حيث سنقوم بكتابة الكود المراد تكراره بشكل دوري ضمن  
 حلقة (1).While.

```

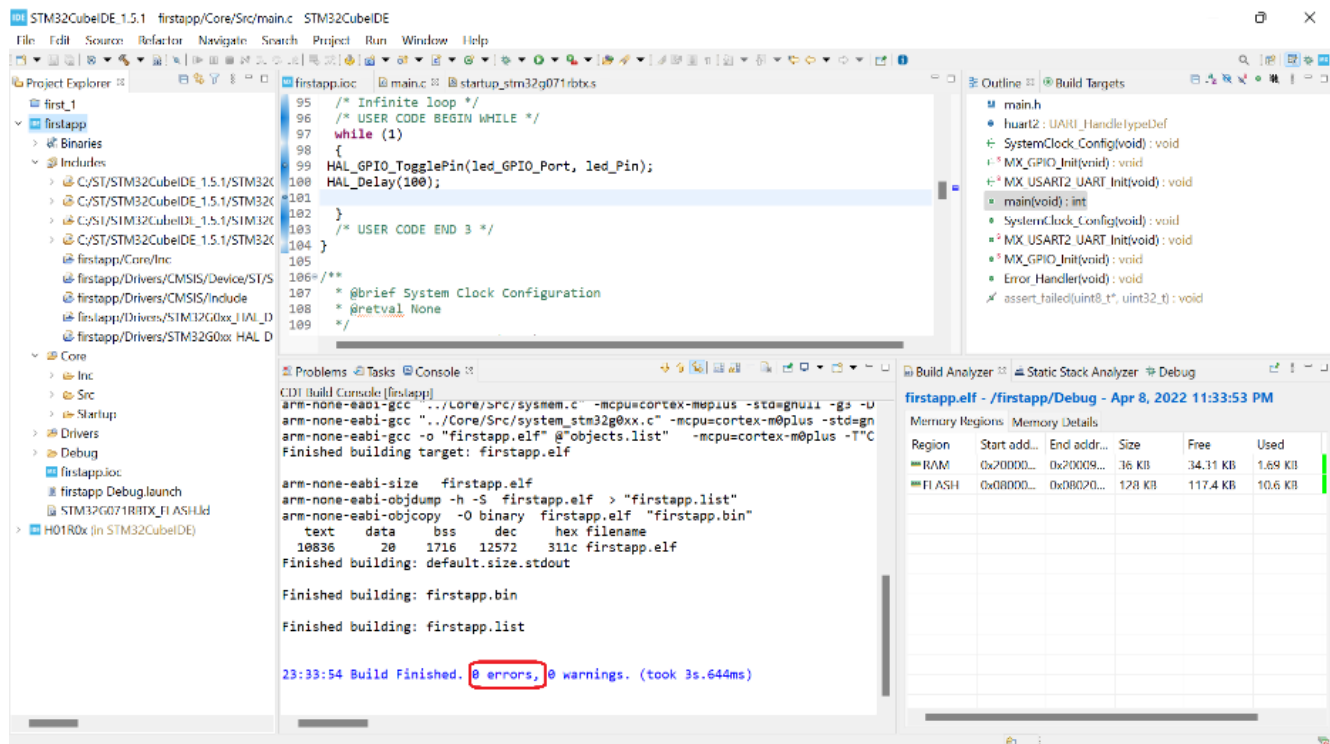
/* USER CODE BEGIN WHILE */
while (1)
{
  HAL_GPIO_WritePin(GPIOA, GPIO_PIN_5, GPIO_PIN_SET);
  HAL_Delay(100);
  HAL_GPIO_WritePin(GPIOA, GPIO_PIN_5, GPIO_PIN_RESET);
  HAL_Delay(100);
}
/* USER CODE END 3 */
}

```

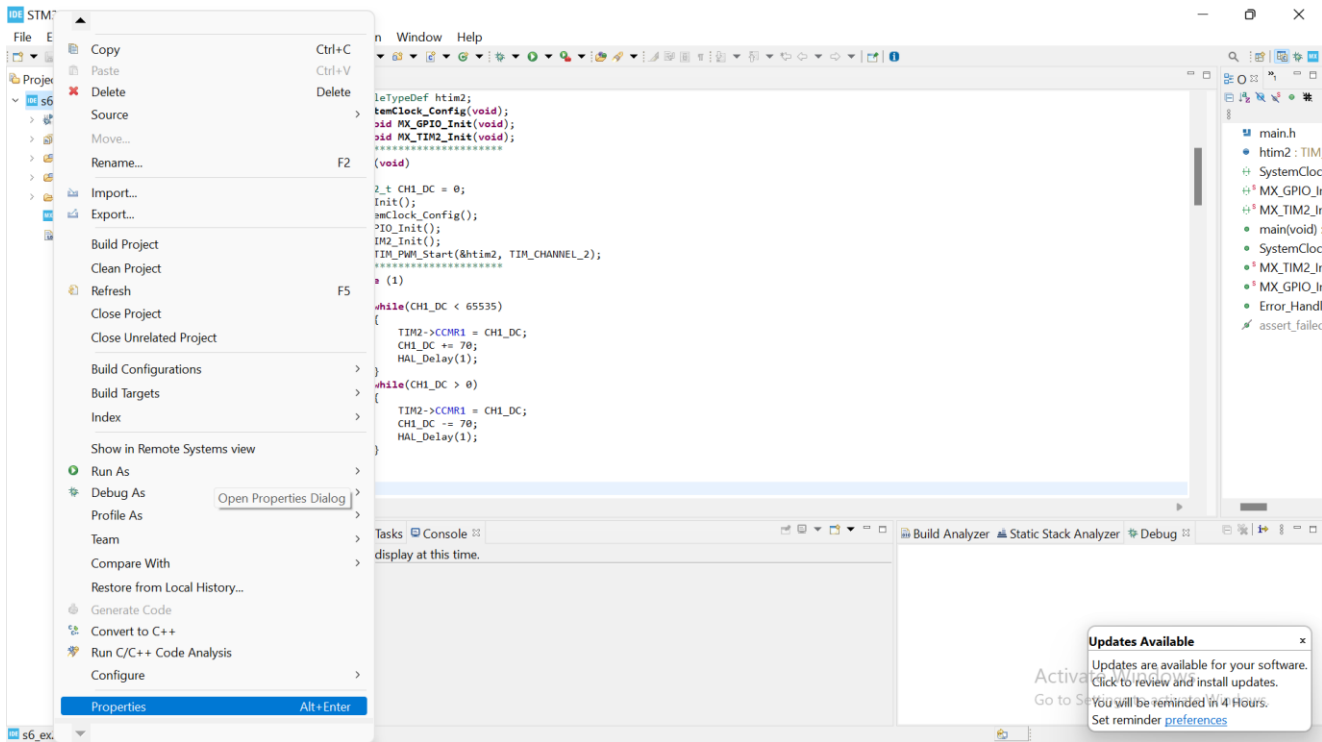
**الخطوة السادسة:** قم بالنقر بزر الفأرة الأيمن على اسم المشروع ثم اختر Clean project ثم Build  
 project لتتم عملية ترجمة الكود للصيغة الثنائية والتأكد من خلو الكود من الأخطاء اللغوية  
 Syntax errors



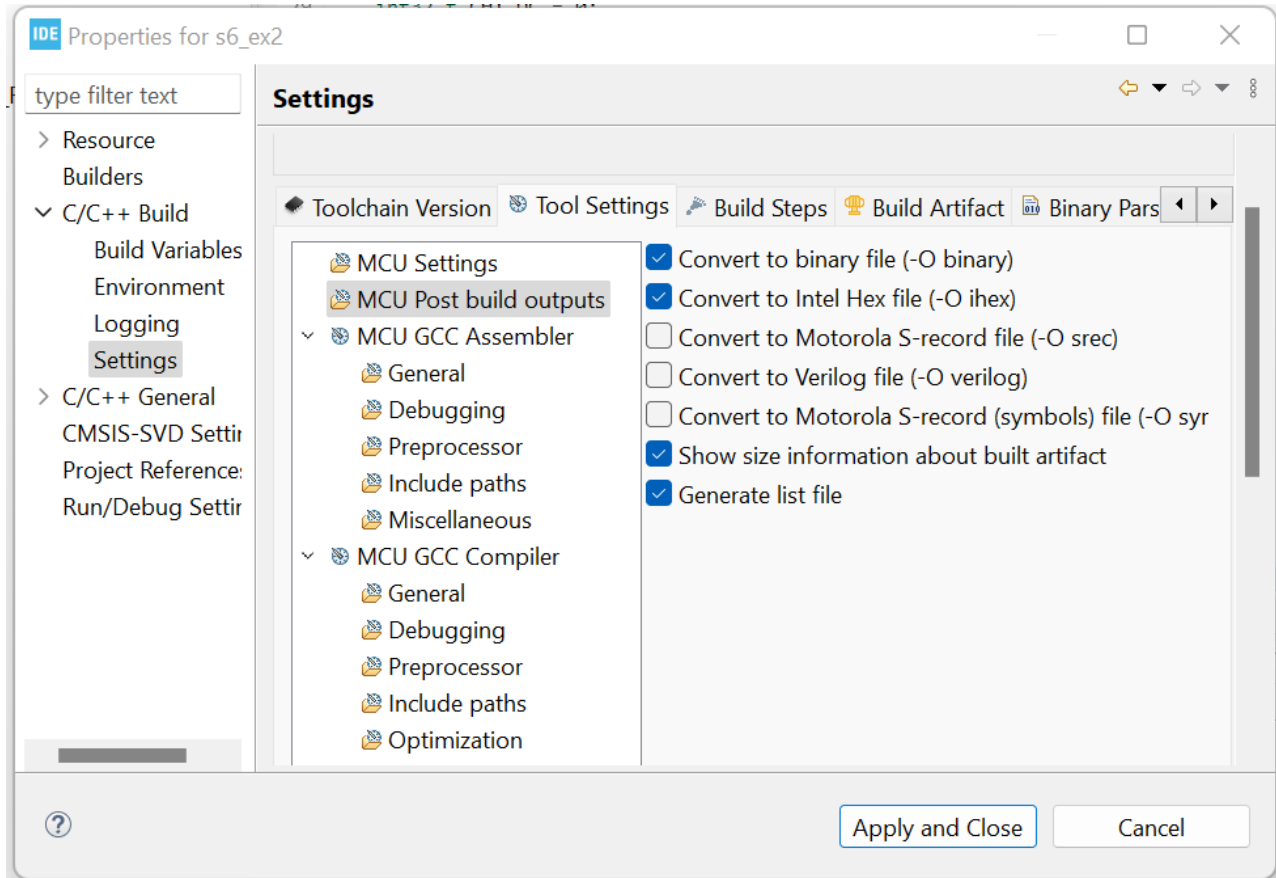
: Syntax errors في حال خلو الكود من أخطاء الـ



في حال لم يتم توليد ملف بامتداد hex. اذهب إلى خصائص المشروع

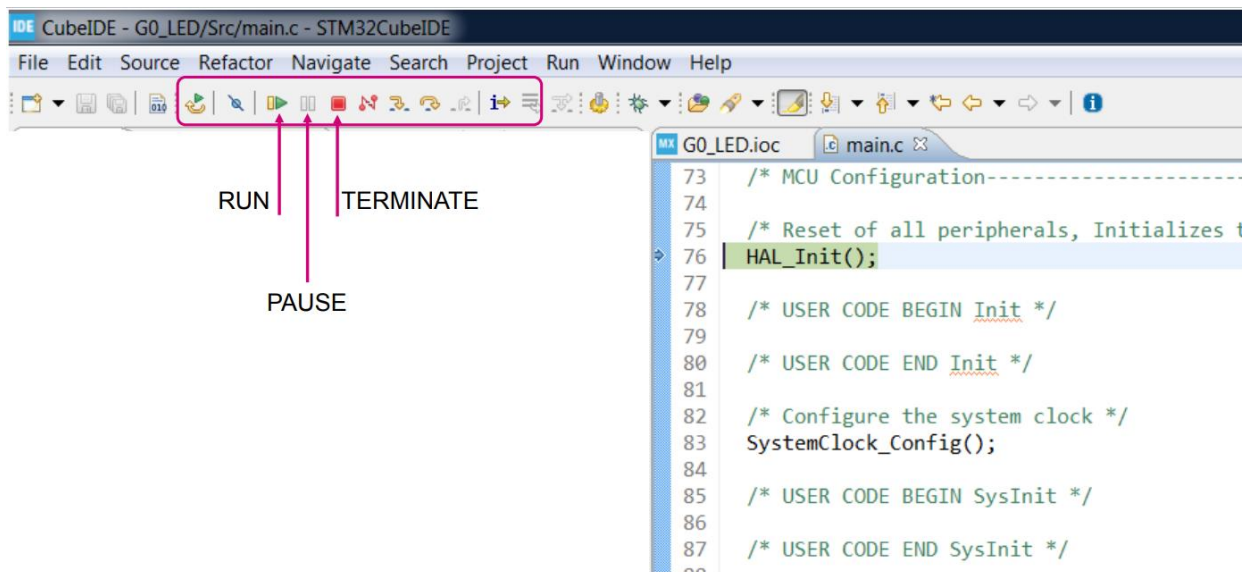


واختر مايلي:



- الخطوة السابعة رفع الكود إلى المتحكم : هناك ثلاث طرق لرفع الكود للمتحكم:
- عن طريق الـ **debug**: لكن في هذه الحالة يجب أن يكون لديك مبرمجة ST-LINK ، قم بالضغط على أيقونة الـ debug لتبدأ دارة ST-Link برفع الكود إلى المتحكم ثم البدء بجلسة debug لمراقبة الكود خطوة بخطوة وتصحيح الأخطاء البرمجية.  
رابط الفيديو على اليوتيوب:

- <https://youtu.be/bCDBXjFm7vc?list=PLX2KimaUhReFKS3CazJhtwHqpisQvEc>  
Td



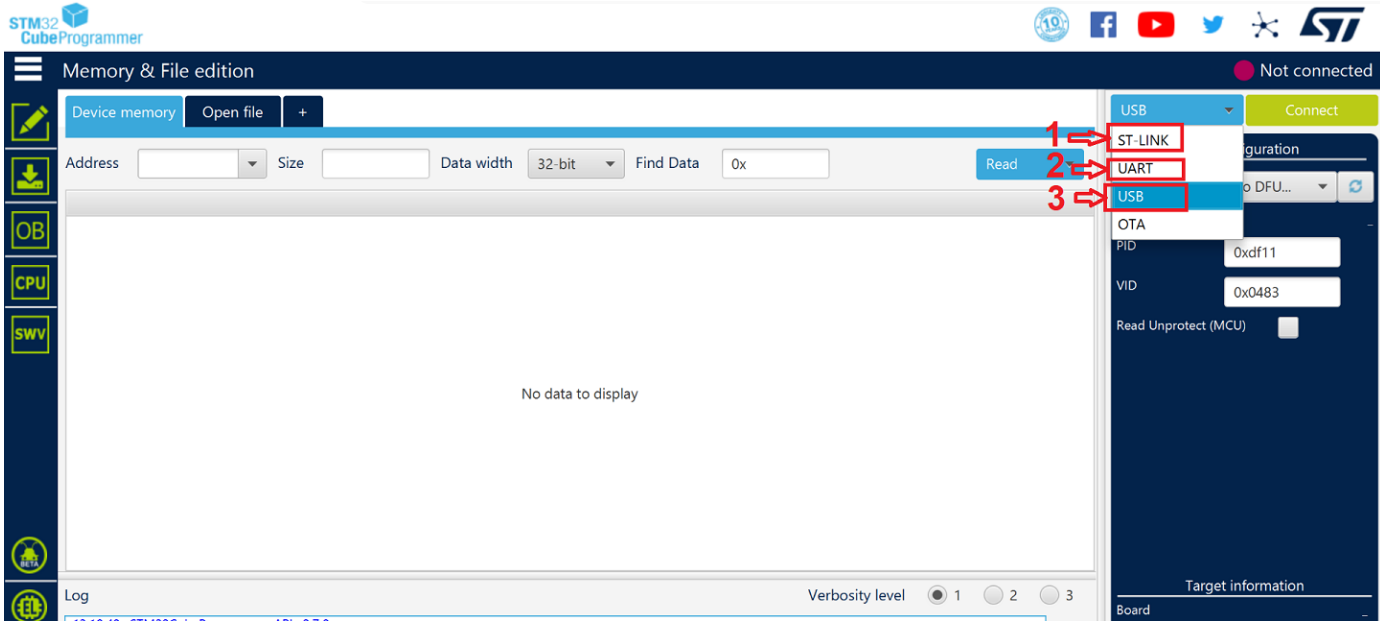
- 21**

رابط الفيديو على اليوتيوب:

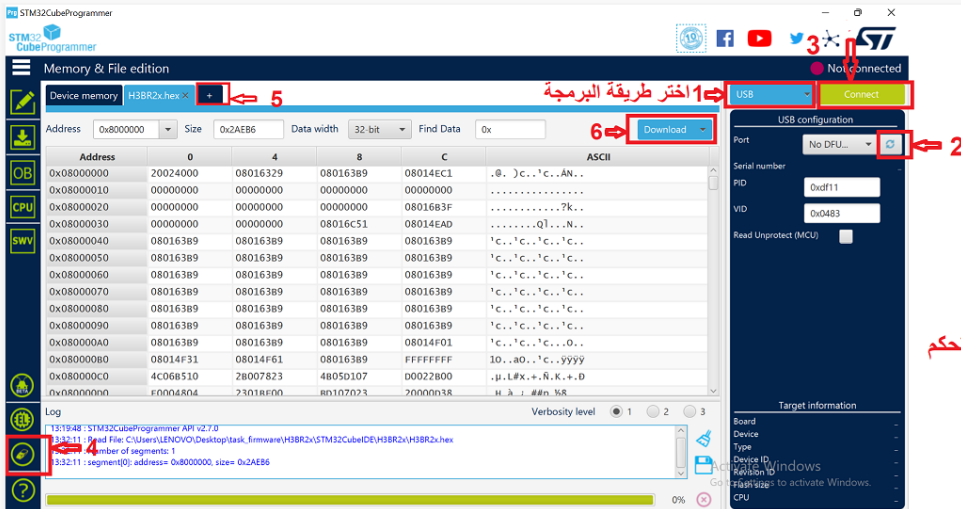
[https://youtu.be/JcTeVRRh-](https://youtu.be/JcTeVRRh-6A?list=PLX2KimaUhReFKS3CazJhtwHqpisQvEcTd)

[6A?list=PLX2KimaUhReFKS3CazJhtwHqpisQvEcTd](https://youtu.be/JcTeVRRh-6A?list=PLX2KimaUhReFKS3CazJhtwHqpisQvEcTd)

[https://youtu.be/FG6xmco4Zbc?list=PLX2KimaUhReFKS3CazJhtwHqpisQvEc](https://youtu.be/FG6xmco4Zbc?list=PLX2KimaUhReFKS3CazJhtwHqpisQvEcTd)  
[Td](https://youtu.be/FG6xmco4Zbc?list=PLX2KimaUhReFKS3CazJhtwHqpisQvEcTd)



سنختار البرمجة عن طريق الـ usb وذلك تبعاً للخطوات التالية:



- 1- قم باختيار طريقة البرمجة
- 2- قم بعمل تنشيط للمنفذ
- 3- قم بالاتصال بالمتحكم
- 4- قم بمسح الكود القديم من ذاكرة المتحكم
- 5- قم باختيار ملف الكود الذي تود تحميله للمتحكم
- 6- قم بتحميل الكود للمتحكم

روابط فيديوهات الجلسة على اليوتيوب:

- <https://youtu.be/JcTeVRRh-6A?list=PLX2KimaUhReFKS3CazJhtwHqpisQvEcTd>
- <https://youtu.be/FG6xmco4Zbc?list=PLX2KimaUhReFKS3CazJhtwHqpisQvEcTd>
- <https://youtu.be/bCDBXjFm7vc?list=PLX2KimaUhReFKS3CazJhtwHqpisQvEcTd>

أيضاً بإمكانك محاكاة المثال من خلال برنامج **Proteus** وبلاستعانة بالفيديو التالي:

- <https://youtu.be/fCKR9B23q20?list=PLX2KimaUhReFKS3CazJhtwHqpisQvEcTd>