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

مقرر المتحكمات المصغرة

الجلسة الأولى

السنة الرابعة ميكاترونيك

2023/2022

محتويات الجلسة

1- تنصيب البرامج اللازمة STM32CubeIDE ، Proteus
2- تعريف عن المتحكم STM32G0 وأهم المزايا الخاصة به
3- تعريف عن البورد المستخدم
4- بناء تطبيق إضاءة ليد وإطفاؤه كل 100msecباستخدام بيئة المحاكاة Proteus
5- بناء تطبيق إضاءة ليد وإطفاؤه كل 100msecباستخدام البورد التطويري

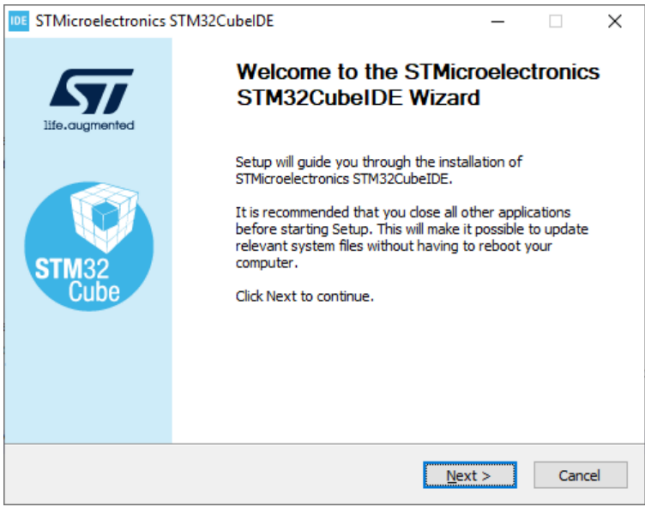
1- تنصيب البرامج اللازمة STM32CubeIDE ، Proteus

ملاحظة: متطلبات بيئة STM32CubeIDE :

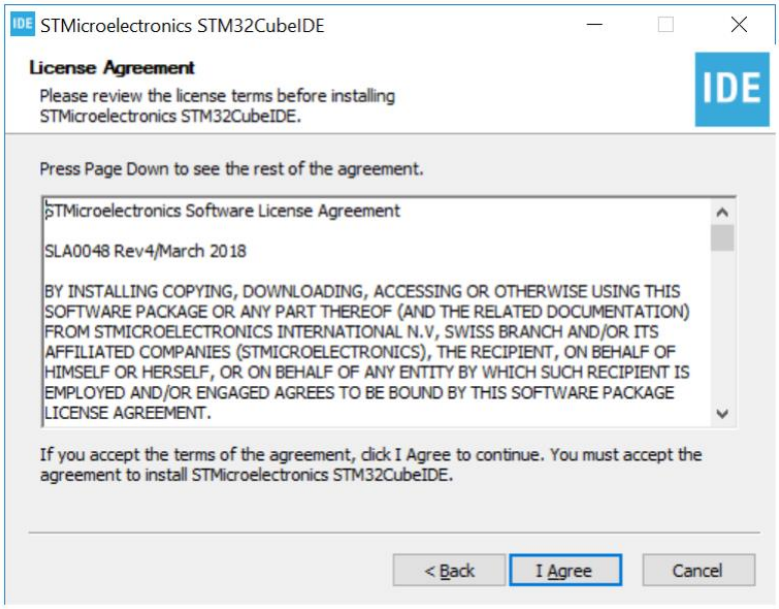
1. نظام تشغيل 64bit حصراً
2. يمكن تنصيبها على عدة أنظمة تشغيل Microsoft® Windows 10®، Linux®، macOS®
3. مواصفات الحاسب المطلوبة لتنصيب البيئة:
   - رامات بحجم 2GB على الأقل ويفضل أن تكون 4GB

في البداية نقوم بتنصيب بيئة STM32CubeIDE وفق الخطوات التالية:

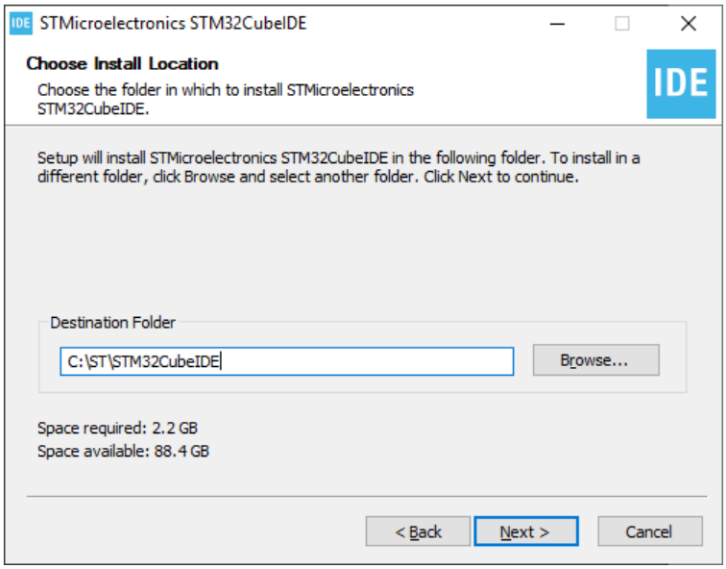
- نقوم بفتح الملف st-stm32cubeide_VERSION_ARCHITECHURE.exe فتظهر النافذة الترحيبية



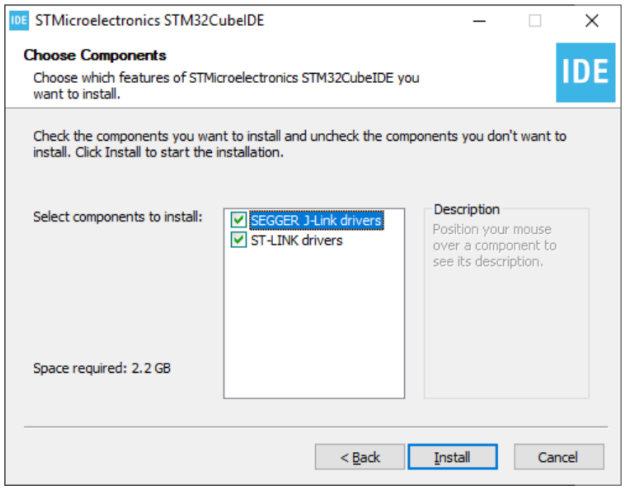
- نضغط Next



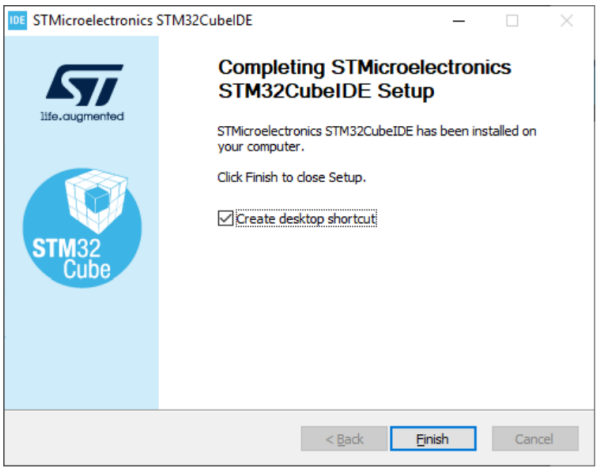
- نختر I agree



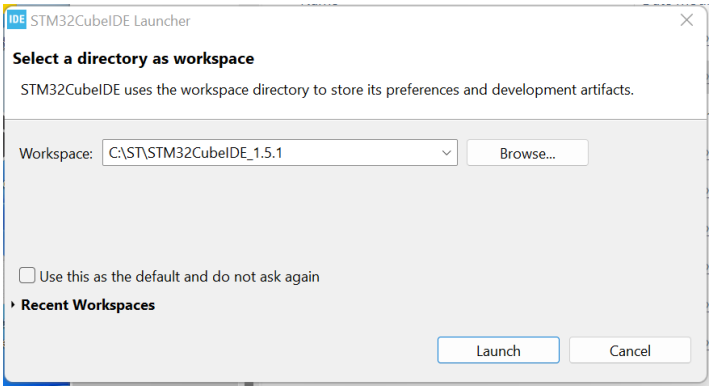
- Next



- نضغط على Install

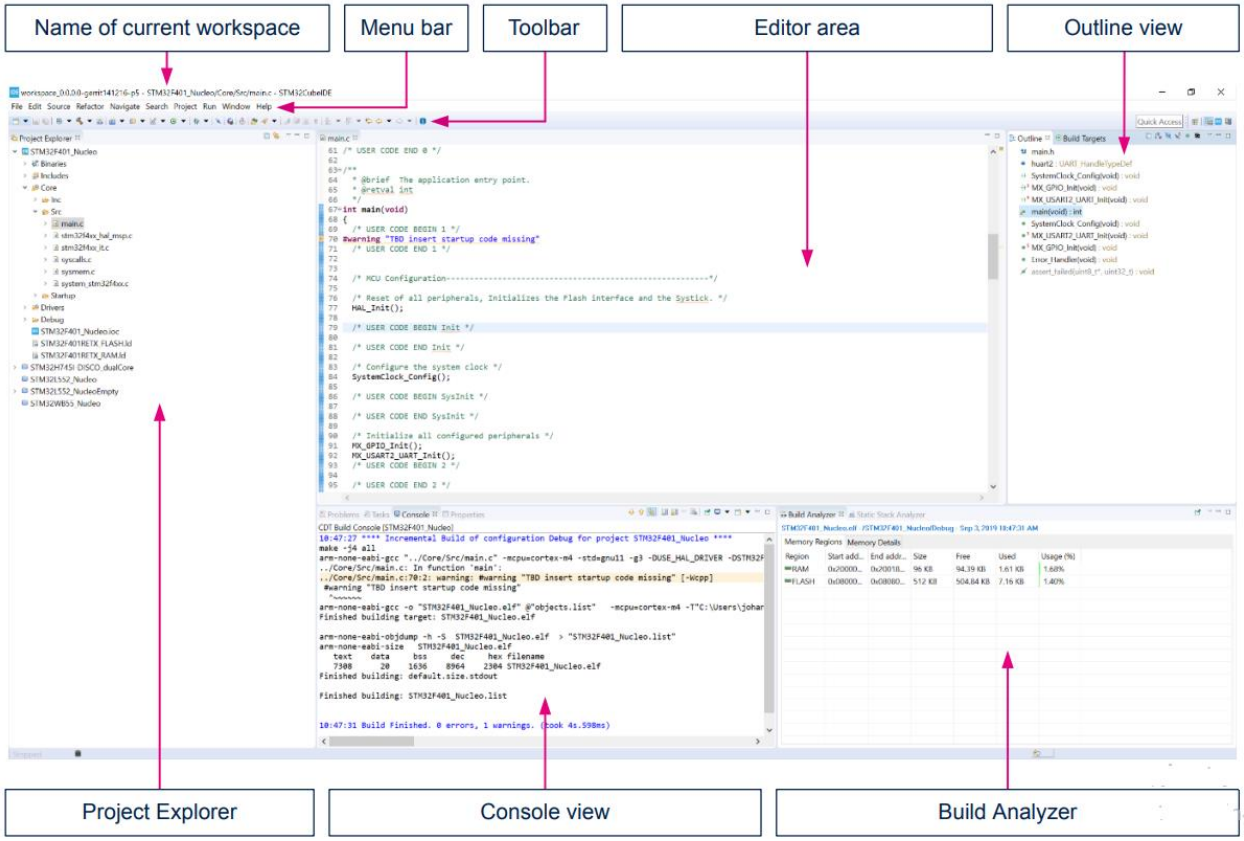


بذلك يكون تم تنصيب بيئة STM32CubeIDE، نقوم بفتح بيئة STM32CubeIDE فتظهر النافذة التالية:

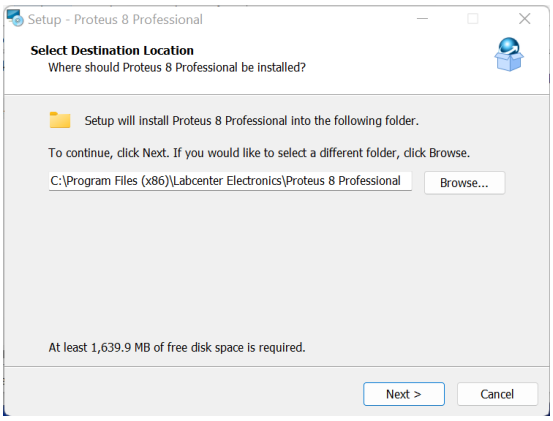


وهنا يطلب منك اختيار مساحة العمل الخاصة بالبيئة والتي ضمنها سيتم حفظ المشاريع والتطبيقات التي ستقوم بإنشائها لاحقاً ويفضل عدم تغيير المسار.

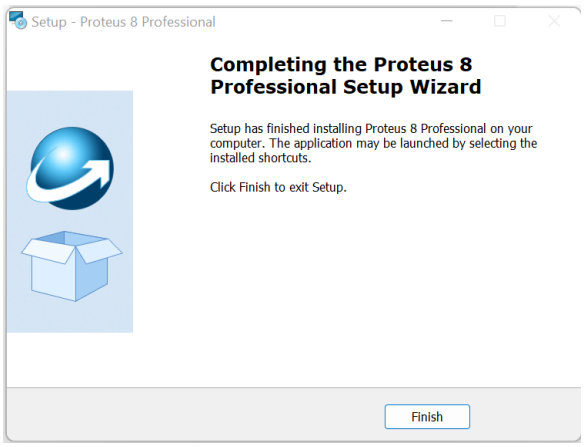
بعد الضغط على زر Launch فإذا قمنا باختيار أحد التطبيقات من الأمثلة الموجودة تظهر لنا النافذة التالية:

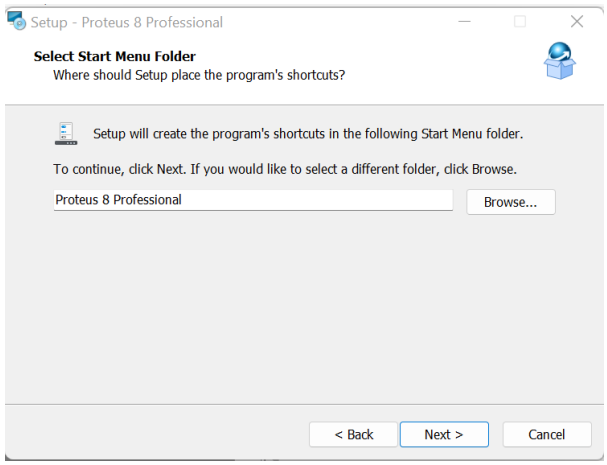


- لتنصيب برنامج PROTEUS نختار الملف proteus 8 professional.exe :

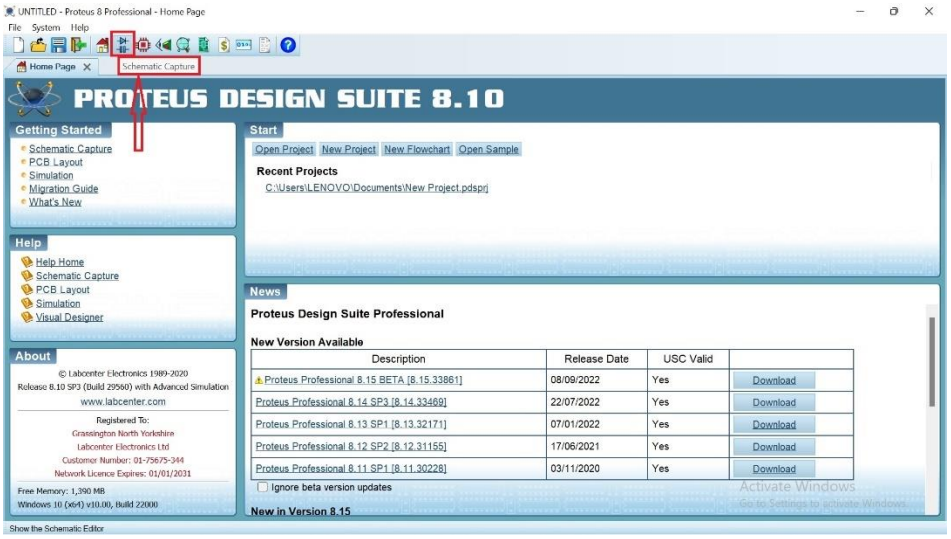


- اترك المسار كما هو واضغط على Next....Next... ثم Finish

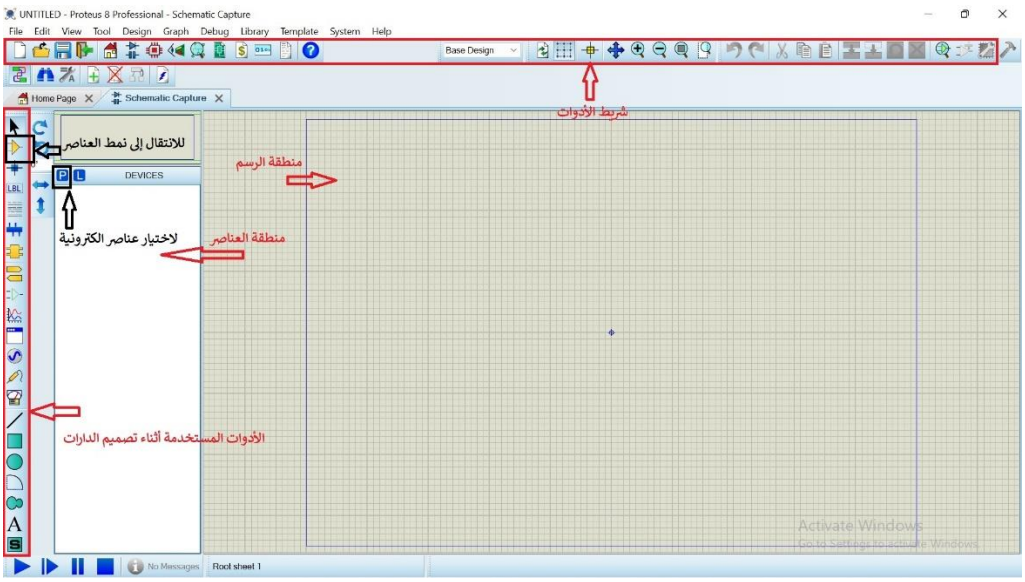




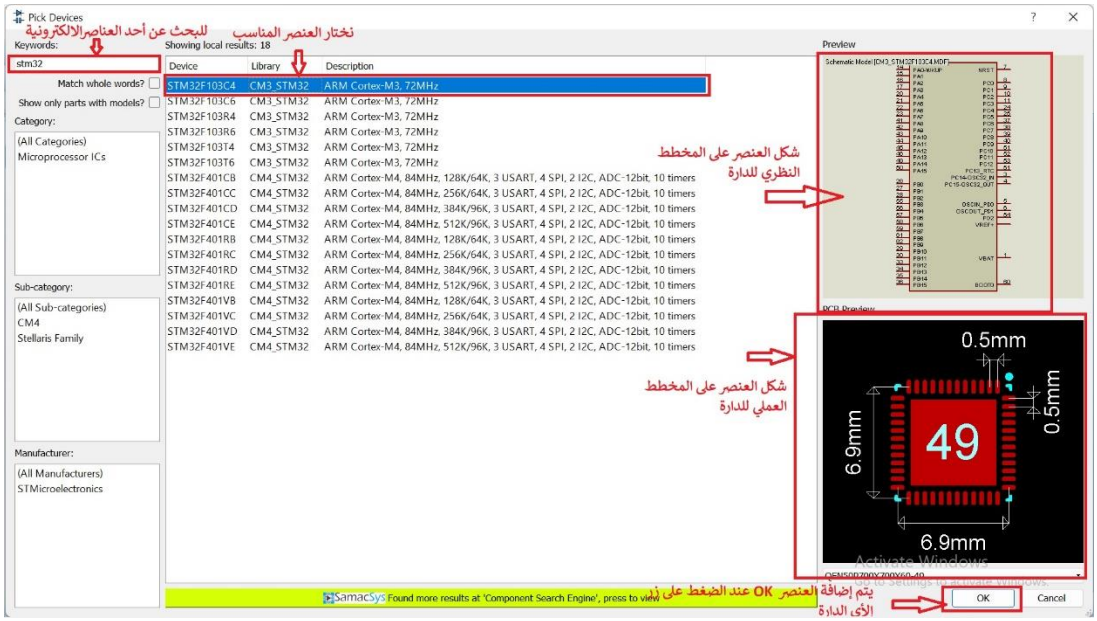
- عند تشغيل برنامج Proteus تظهر النافذة التالية:



- للانتقال إلى Schematic Capture نضغط على الأيقونة الموضحة بالشكل فتظهر النافذة التالية:

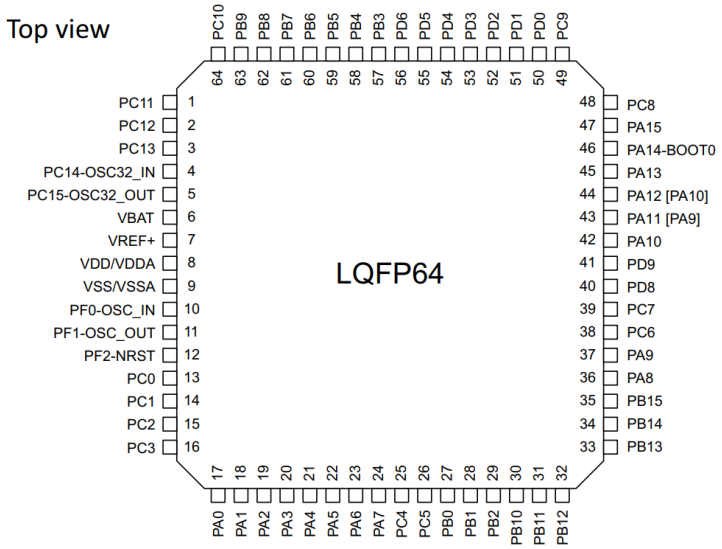


- لاختيار أحد العناصر وإضافتها للتصميم نضغط على الأيقونة P فتظهر النافذة التالية:



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,GPIOD,GPIOF وكل منها مؤلف من 16قطب، 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 128 Kbytes of Flash memory

– 36 Kbytes of SRAM

• CRC calculation unit

- 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 (±1 %)

– Internal 32 kHz RC oscillator (±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 µs 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

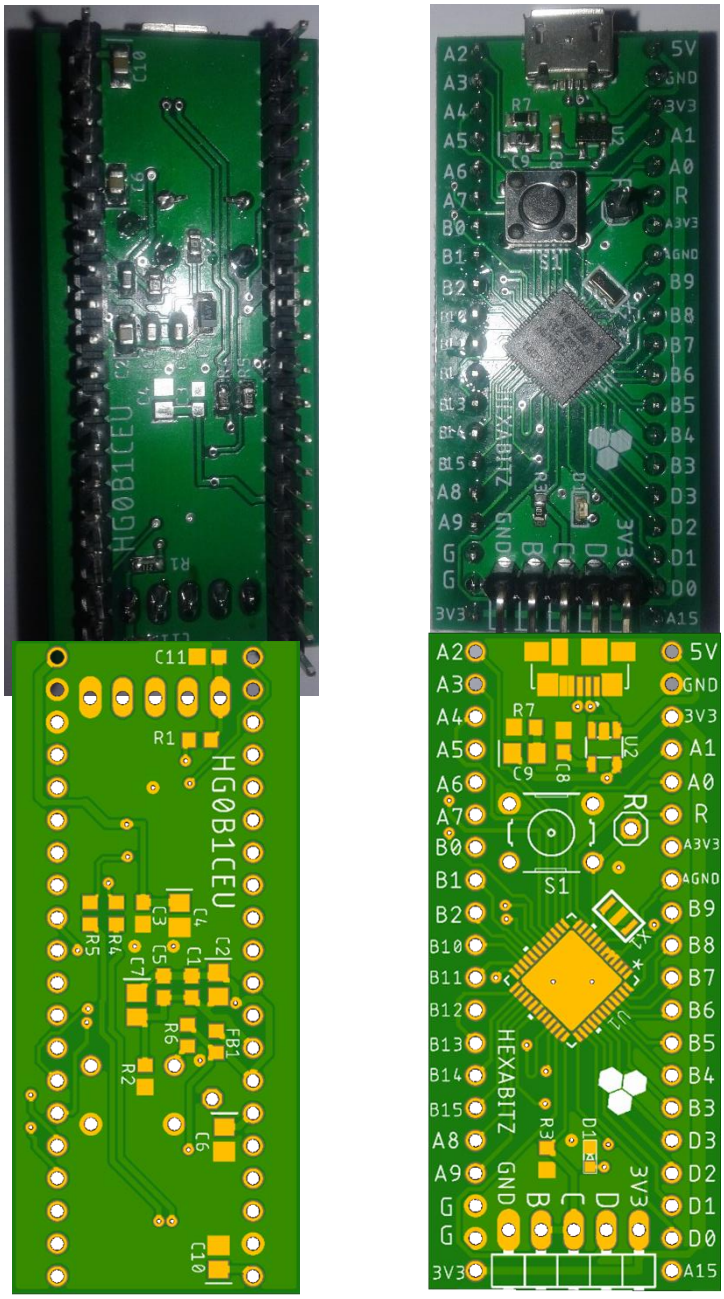
– Four USARTs with master/slave

– One low-power UART

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

3- **تعريف عن البورد المستخدم:**

بورد HG0B1CEU تم تصـميمها للتعامل مع معالج STM32G0B1CEU6N ، حيث تحتوي على المُعالج المُصغّر STM32G0B1CEU6N كمُكوّن أساسي وتم توصيل معظم أقطاب المعالج إلى pin-heads للتعامل مع طرفيات هذا المعالج ، حيث يمكن وضـــع هذه البورد على test-board لإجراء بعض التجارب كذلك يمكن وضعها على دارة مطبوعة بوضع البصمة المناسبة في ال Layout حيث أنّ البُعد ما بين صفّي ال Pin-heads يساوي البعد بين أرجل معالجatmega16 أو32 أو بورد Arduino-Nano :



تمتلك اللوحة التطويرية العديد من الميّزات الأخرى:

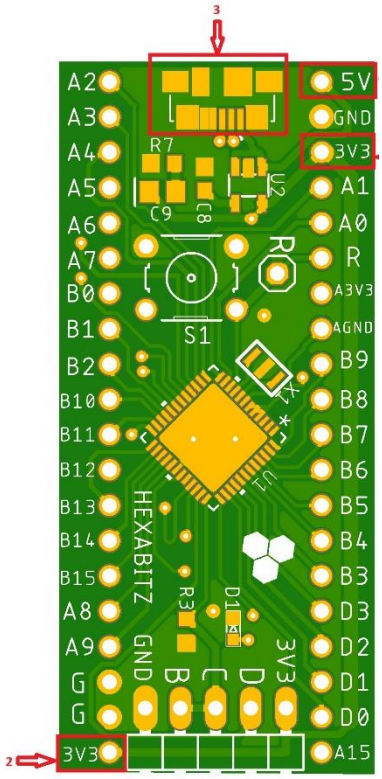
- التغذية:

يمكن تغذية البورد بعدة طرق:

1 - وصل 5 فولت إلى القطب 5 V

2 - وصل 3.3 فولت إلى القطب 3 V3الموجود على البورد

3 - وصل تغذية USB بـ 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 ال5 فولت أو الUSB-Micro

- تحتوي البورد على مُنظّم جهد 3.3 فولت ، وبالتالي عند توصيل 5 فولت إلى البورد يمكن الاستفادة من قطب V3 3 لتغذية دارات خارجيّة بـ 3.3 فولت ولكن التيار المُستجَر لا يجب أن يتجاوز ال100 مللي أمبير.

-تحتوي البورد على أقطاب تغذية A3V3 , AGND لتغذية الدارات التشابُهيّة بـ 3.3 فولت مُرشّحة بملف لكي لا تُؤثّر على عمل المُعالج.

-القطب GND هو نفسه القطب G

- البرمجة:

يمكن برمجة المعالج بعدة طرق :

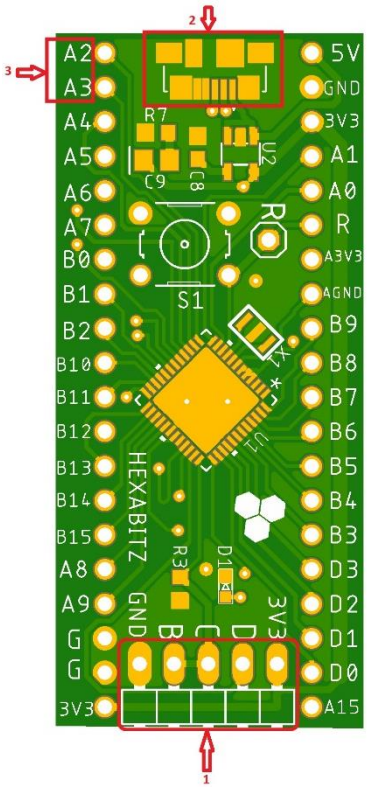
1 - عبر مُبرمجة SWD بعد وصل أقطابها إلى البورد (C - D - R) وفي هذه الحالة يمكن رفع الفيرموير إلى المُعالج وعمل Debug وتتبع تنفيذ الكود خطوة بخطوة من خلال بيئة STM32CubeIDE أو من خلال البرنامج STM32programmer

2 - عبر منفذ ال USB-Microحيث يدعم مُعالج ال G0الدخول إلى وضع الـــ Bootloader عبر ال USB بالتوصيل إلى الأقطاب المُخصّصة ولكن في هذه الحالة لا يمكن عمل Debug للكود.

3 -عبر نافذة ال USART2عبر الأقطاب PA2,PA3 حيث أنّ PA2 هو الـــ TXوال PA3 هو ال RX حيث ينم وصل القطب PA2 مع قطب الـــ RX لكبل الـــ FTDI ووصل قطب الـــ PA3 مع قطب الـــ TX لكبل الـ FTDI.

مُلاحظة :

الطريقتين رقم (2 و3) لا يمكن استخدامهما إلّا بعد رفع فيرميور مُسبقا على المعالج عبر الطريقة رقم (1)

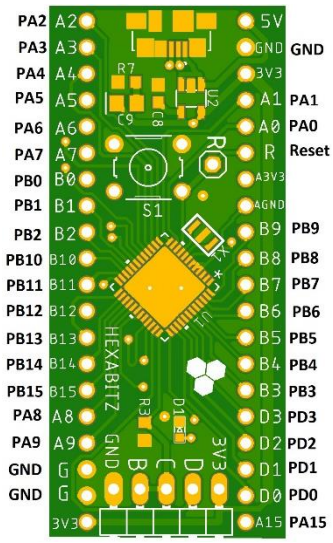


- **التعامل مع طرفيّات المعالج:**

1 - كما هو واضـح في الصـور فإن أسـماء أقطاب المعالج تم طباعتها على البورد بجانب كل Pin-head فمثلا (B3=PB3)وهكذا...

2 - يوجد زر S1 بجانب القطب R لعمل Reset للمعالج عند الرغبة.

3 - يمكن إدخال المُعالج بشكل قسري إلى وضع الـ Bootloader بوصل القطب B إلى ال3.3 فولت مباشرةً .

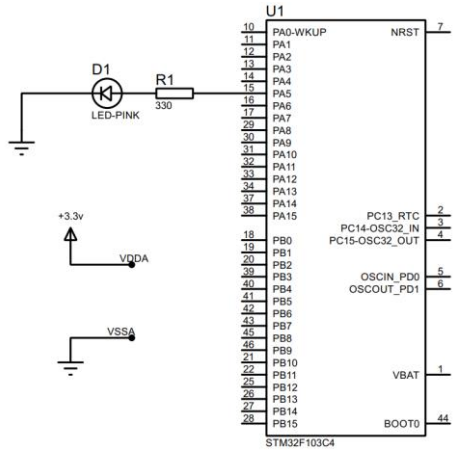


- بإمكانك الاطلاع على الملفات التصميمية للوحة التطويرية من خلال الرابط التالي:

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

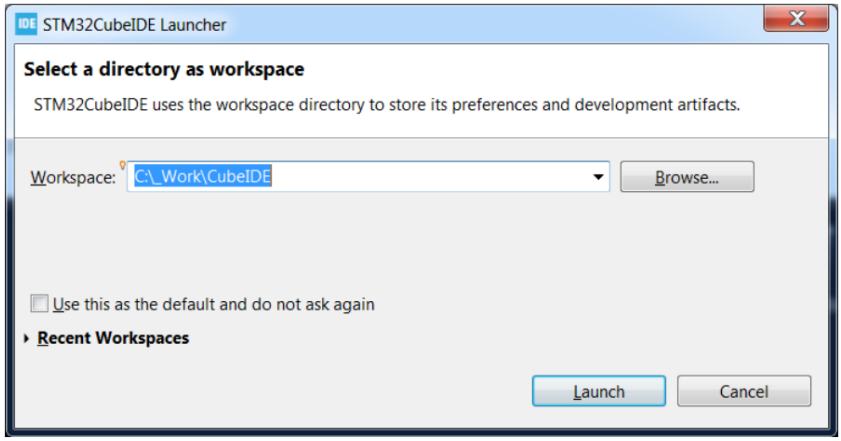
4- بناء تطبيق إضاءة ليد وإطفاؤه كل 100msecباستخدام بيئة المحاكاة Proteus

سنقوم بتصميم تطبيق يقوم بعمل toggle لليد المتصل بالقطب PA5:

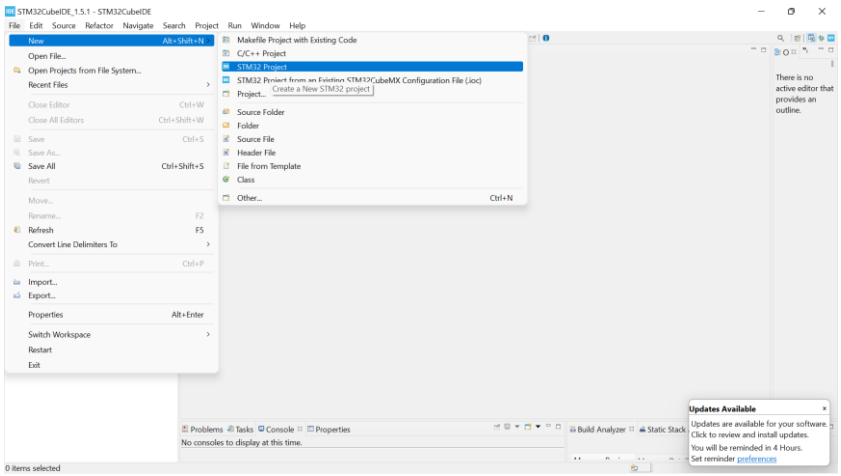


وذلك وفق الخطوات التالية:(استخدمنا هنا المتحكم **STM32F103C4** الموجود في برنامج proteus)

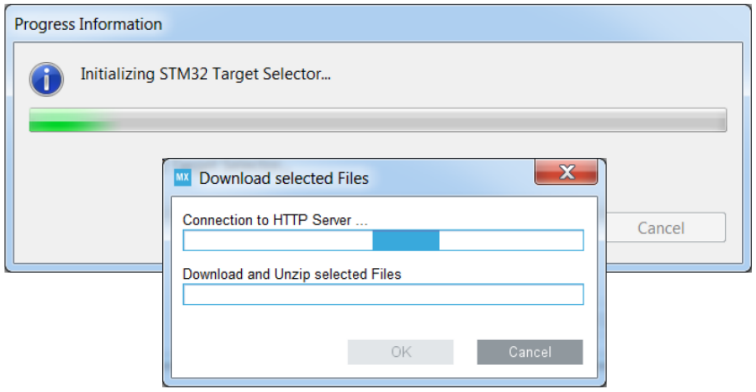
الخطوة الأولى: قم بفتح بيئة STM32cubeIDE



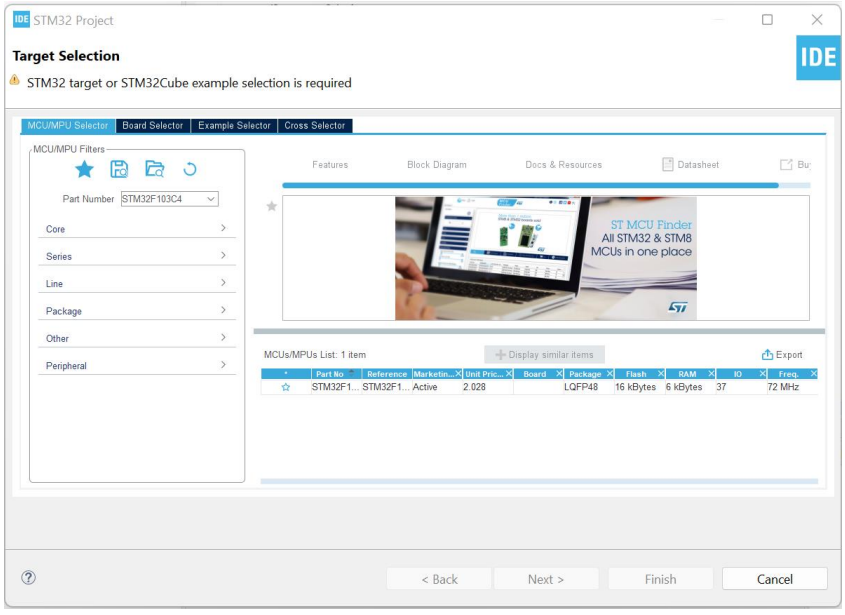
ثم نختار إنشاء مشروع جديد من file …new …Stm32project كما هو موضح بالشكل التالي:

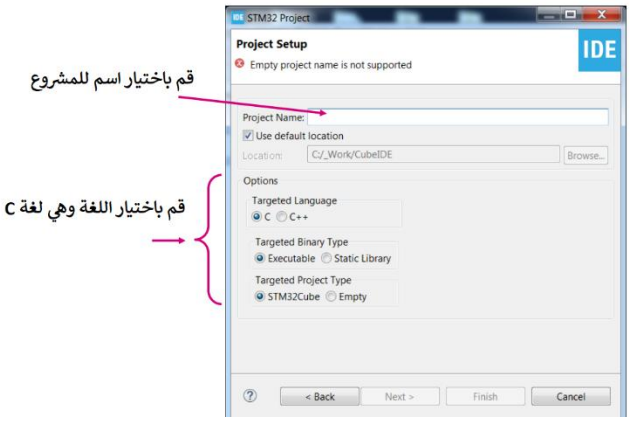


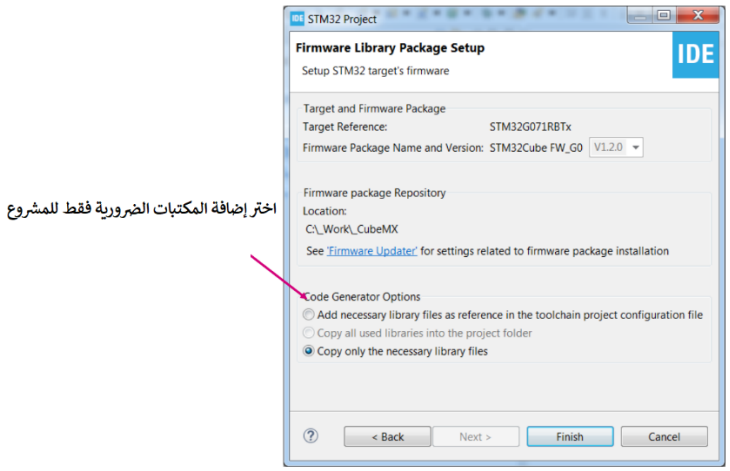
نلاحظ أن البرنامج يبحث عن تحديثات له عبر شبكة الانترنت في حال كان الحاسب متصل بالانترنت:



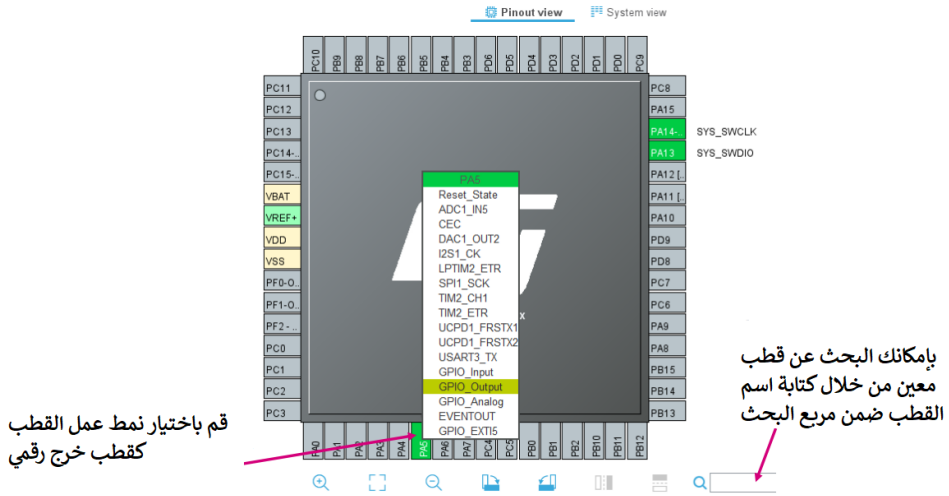
الخطوة الثانية : قم باختيار المتحكم من خلال كتابة اسم المتحكم وهو **STM32F103C4** أو كتابة اسم اللوحة التطويرية في حال استخدام لوحة تطويرية ضمن مربع البحث كما هو موضح بالشكل التالي:



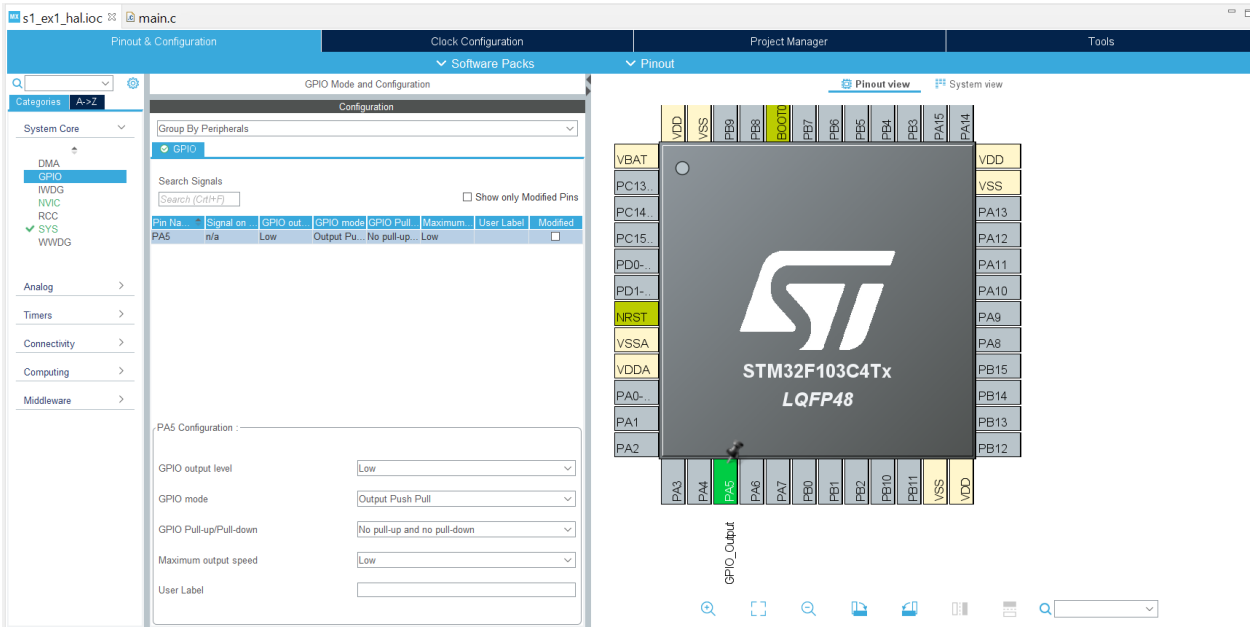




الخطوة الثالثة: قم بتحديد القطب PA5 كخرج كما هو موضح بالشكل التالي:

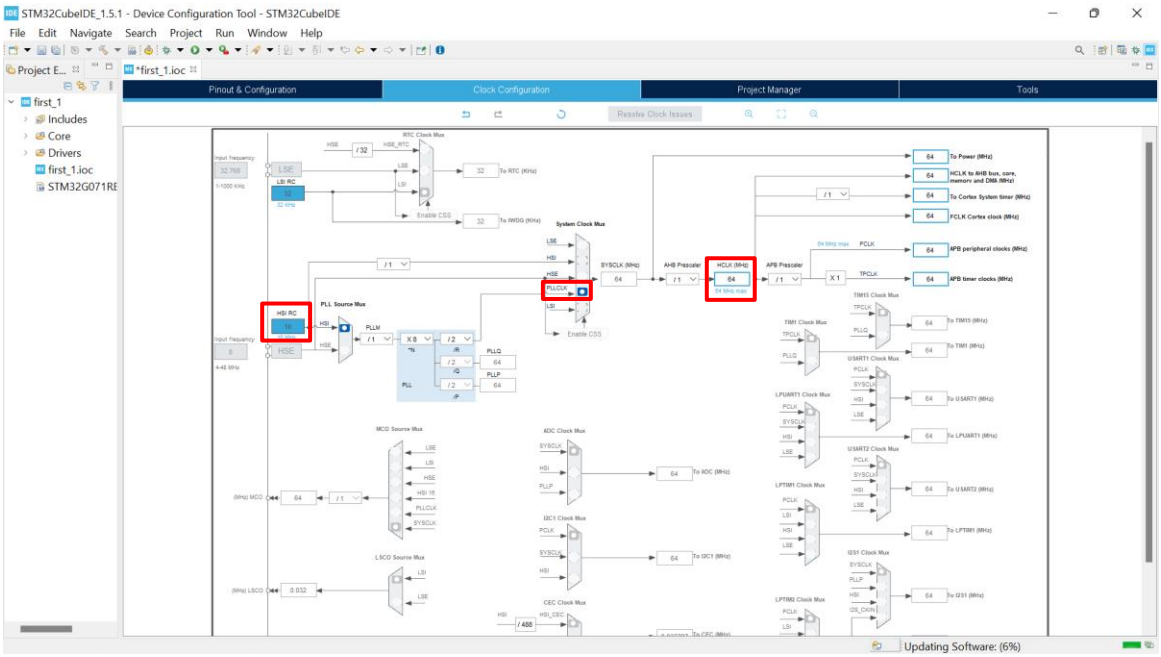


ثم قم بضبط الإعدادات الخاصة به:

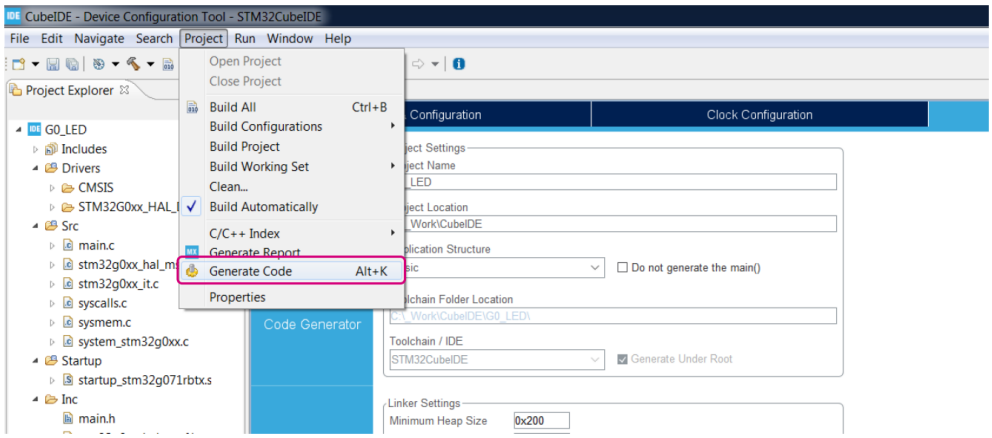


سنختار الحالة الافتراضية للقطب LOW أي يكون في حالة جهد منخفض افتراضياً، وسنختار نمط Output Push Pull وبدون استخدام مقاومات رفع أو خفض No Pull-up and Pull-down وسنختار سرعة الخرج High وسنختار اسم للقطب على سبيل المثال LED_GREEN.

الخطوة الرابعة: قم بضبط تردد الساعة للمتحكم واختر مصدر الساعة الداخلية HSI للمتحكم كما هو موضح بالشكل التالي:



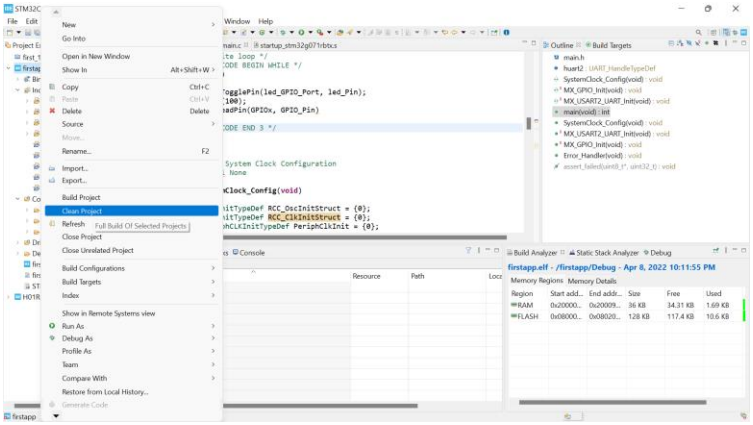
اضغط على Ctrl+s أو من Generate code...Project ، ليتم حفظ المشروع وتوليد الكود وإضافة المكتبات اللازمة، ثم قم بفتح main.c لتعديل الكود بما يناسب مشروعك



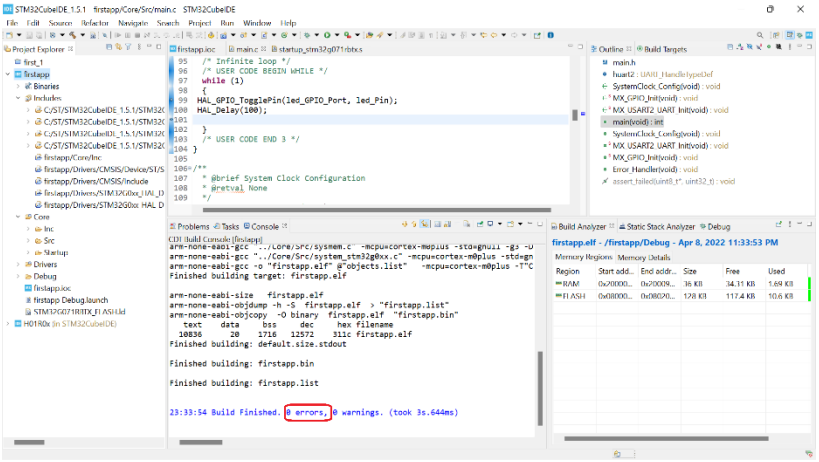
الخطوة الخامسة: نقوم بكتابة الكود المناسب كي يضيء الليد الموجود على القطب رقم 5 من المنفذ A لـ 100msec ويطفأ لـ 100msec ويعيد الكرة في كل مرة، يمكنك الاستعانة بـ Ctrl+Space لاستخدام ميزة الإكمال التلقائي للتعليمات ، حيث سنقوم بكتابة الكود المراد تكراره بشكل دوري ضمن حلقة While(1).

```
  /* 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 */
}
```

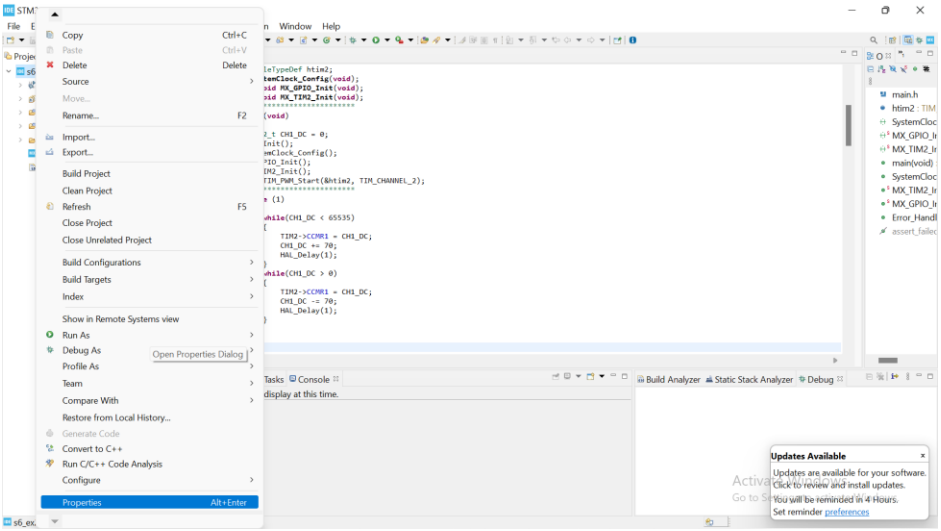
الخطوة السادسة: قم بالنقر بزر الفأرة الأيمن على اسم المشروع ثم اختر Clean project ثم Build project لتتم عملية ترجمة الكود للصيغة الثنائية والتأكد من خلو الكود من الأخطاء اللغوية Syntax errors



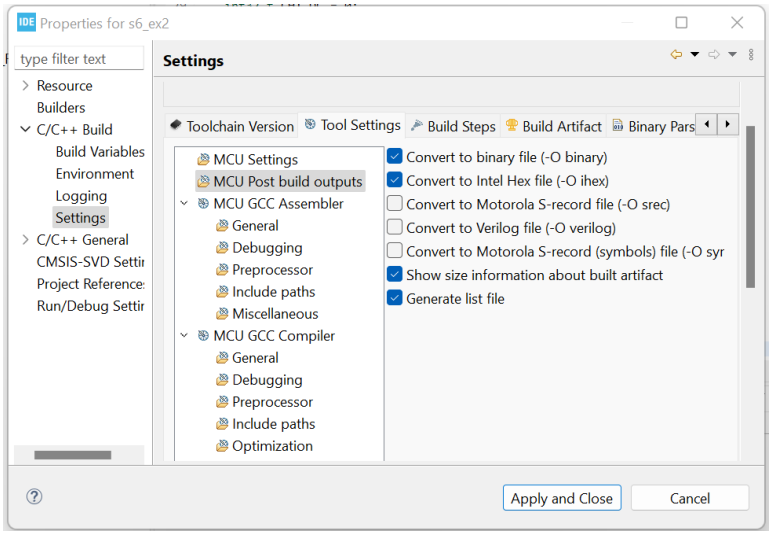
في حال خلو الكود من أخطاء الـ Syntax errors :



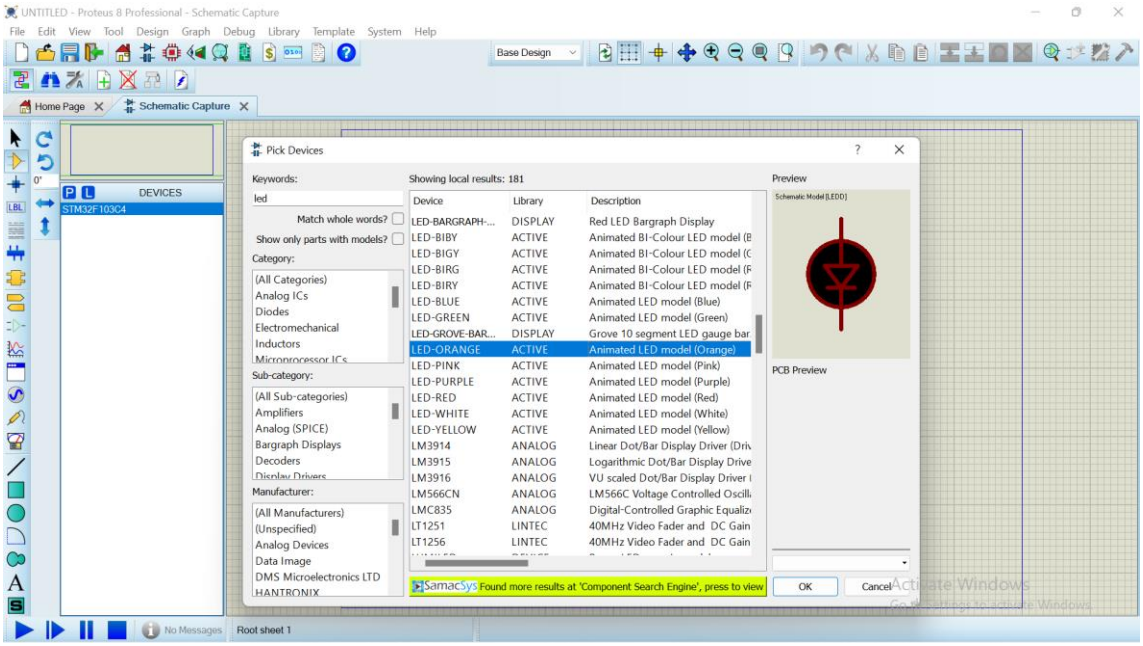
في حال لم يتم توليد ملف بامتداد hex. اذهب إلى خصائص المشروع



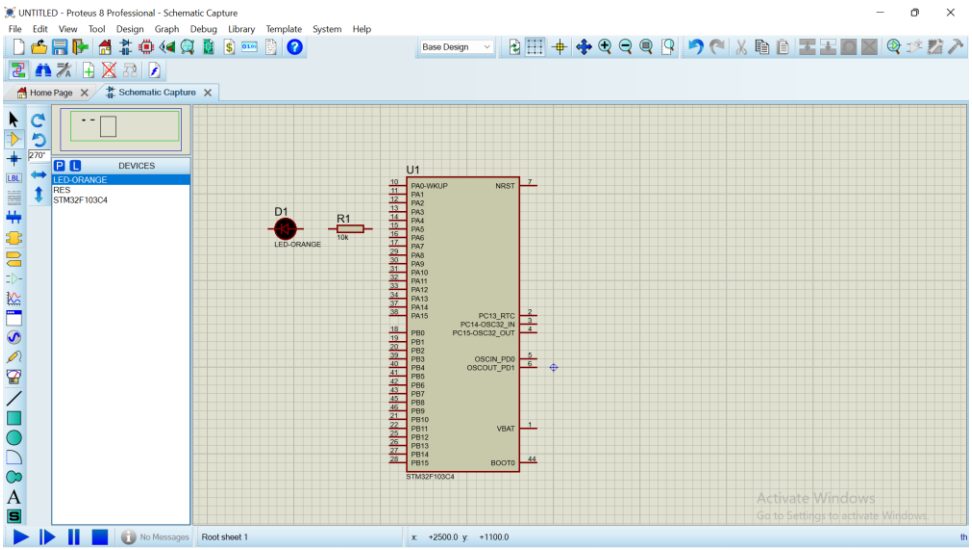
واختر مايلي:



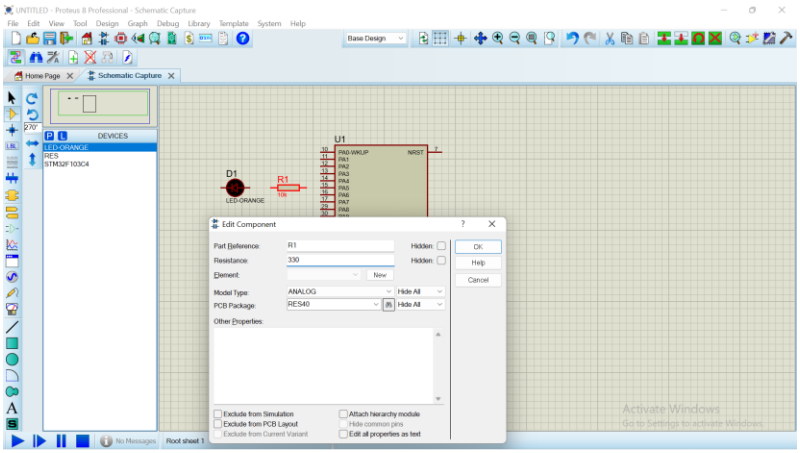
الخطوة السابعة: ننتقل لرسم الدارة على برنامج proteus : نفتح البرنامج ثم نبدأ بإضافة العناصر اللازمة للتطبيق الذي قمنا به وهي عبارة عن متحكم stm32f103c4 وليد ومقاومة



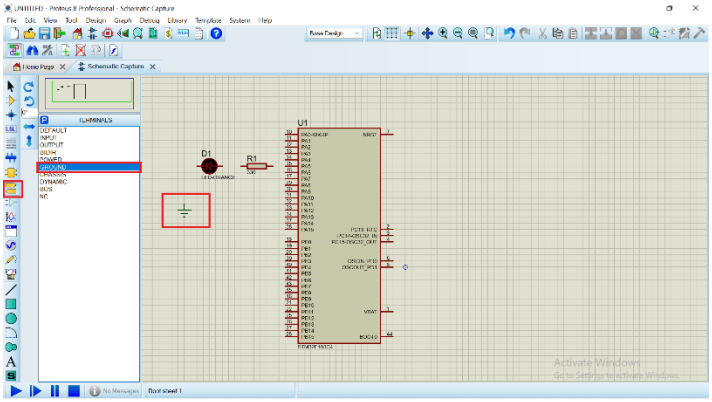
- نقوم بوضع العناصر على مساحة الرسم:



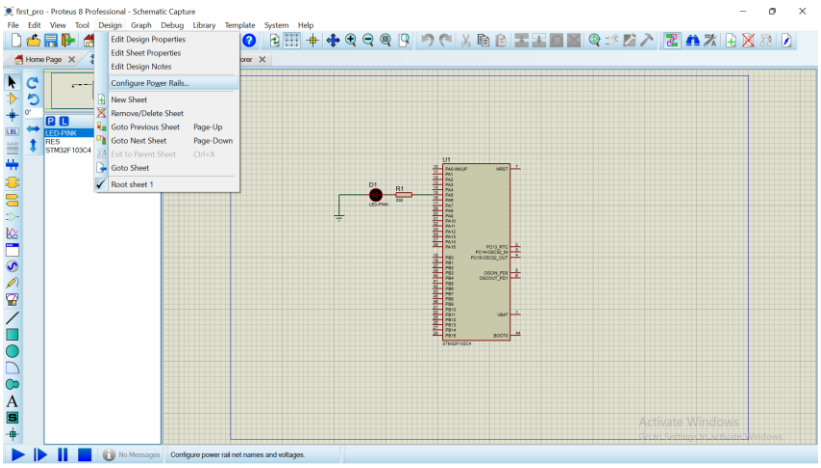
- نقوم بتغيير قيمة المقاومة إلى 330ohm من خلال الضغط بزر الفأرة الأيسر عليها :

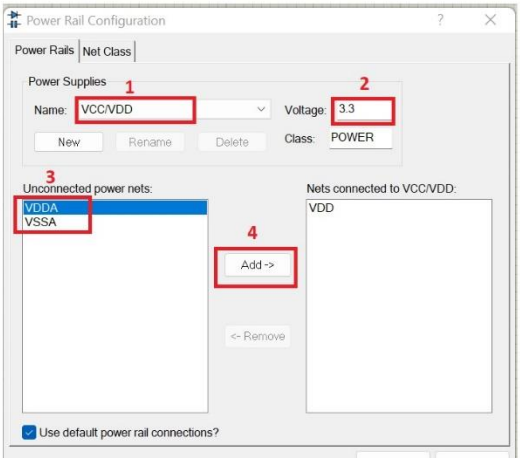


- نقوم بإضافة الأرضي GND والتغذية Power للدارة من خلال الأيقونة 

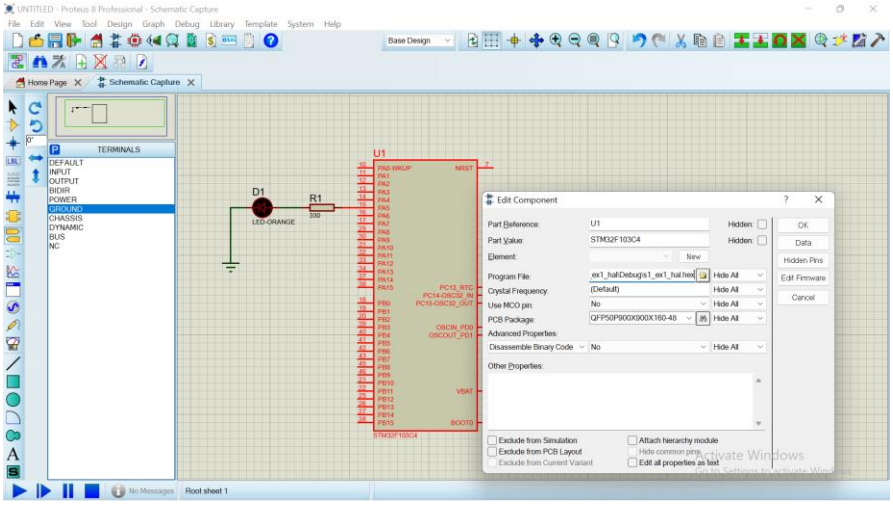


- نقوم بإضافة التغذية للمتحكم من خلال القائمة Design... configure power Rails

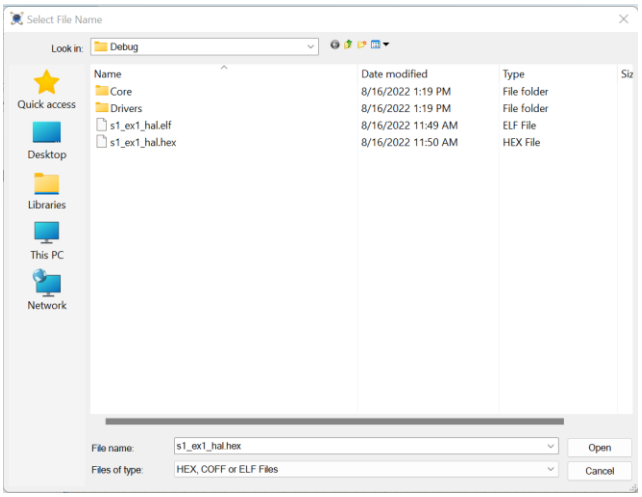




- نقوم بتوصيل العناصر ثم نقوم بالنقر المزدوج على المتحكم لحقنه بملف الكود



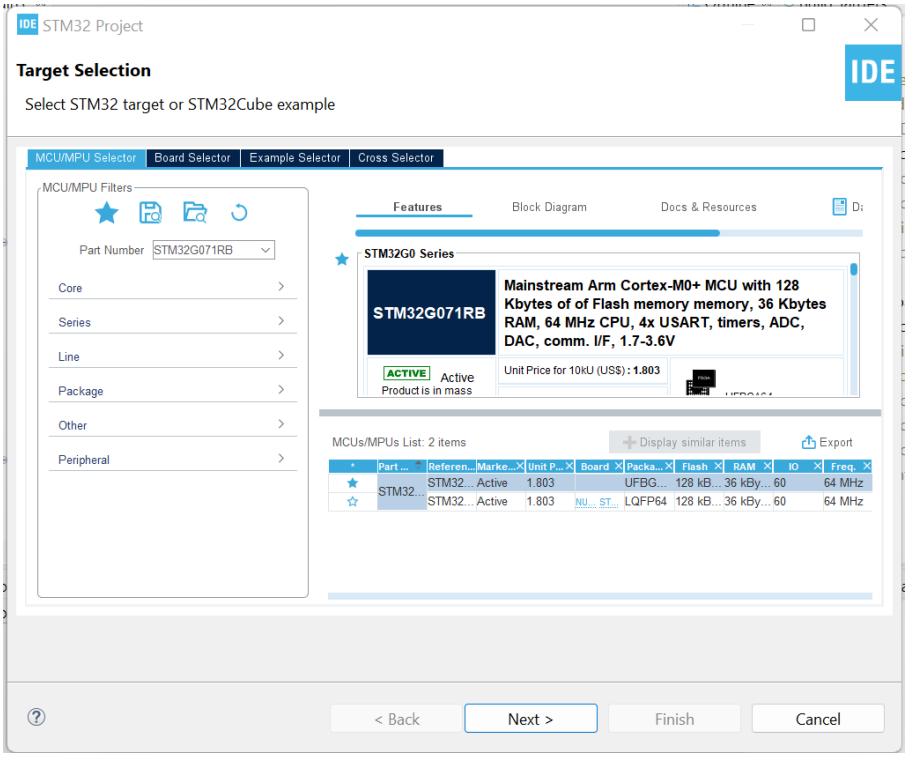
- نقوم باختيار الملف الثنائي الذي يكون بامتداد hex.



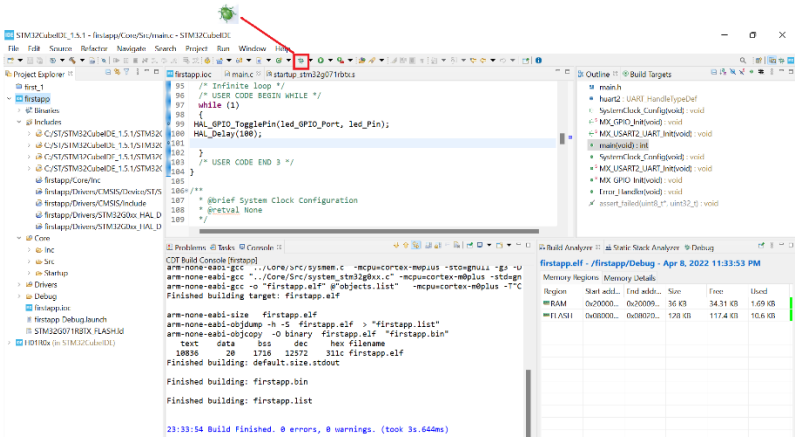
- أخيراً نقوم بالضغط على زر بدء المحاكاة  .

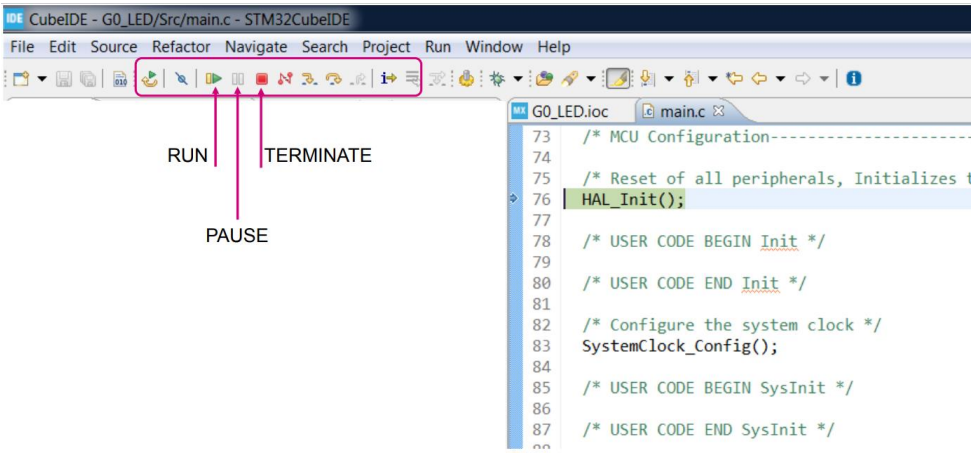
6- بناء تطبيق إضاءة ليد وإطفاؤه كل 100msec باستخدام البورد التطويري

نقوم بإعادة الخطوات من واحد إلى ستة التي قمنا بها في التطبيق السابق ، واستبدل في الخطوة الثانية **المتحكم** STM32F103C4 بالمتحكم **STM32G071RB** مربع البحث كما هو موضح بالشكل التالي:



الخطوة السابعة رفع الكود إلى المتحكم : 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.




CubeIDE - G0_LED/Src/main.c - STM32CubeIDE
File Edit Source Refactor Navigate Search Project Run Window Help
RUN
TERMINATE
PAUSE
G0_LED.ioc
main.c
/* MCU Configuration----------
/* Reset of all peripherals, Initializes t
HAL_Init();
/* USER CODE BEGIN Init */
/* USER CODE END Init */
/* Configure the system clock */
SystemClock_Config();
/* USER CODE BEGIN SysInit */
/* USER CODE END SysInit */