

برمجة أقطاب الدخل والخرج في متحكمات STM32

محتويات الجلسة:

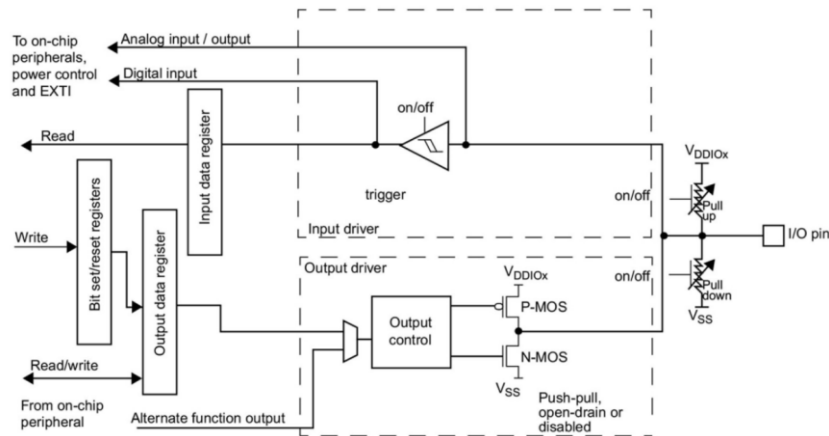
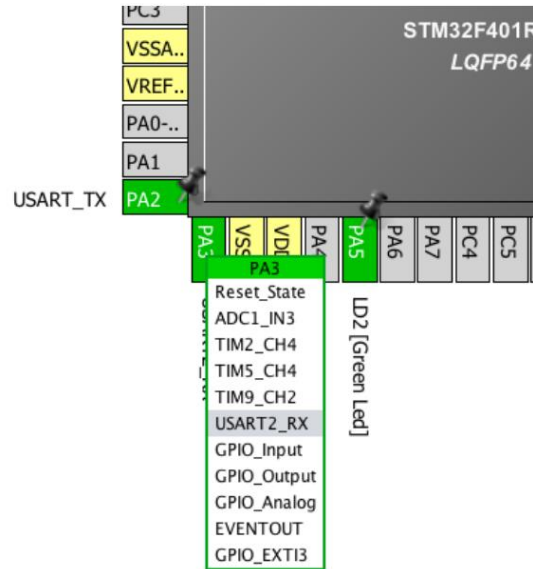
- 1- أنماط عمل أقطاب المتحكم GPIO Mode
- 2- برمجة أقطاب الخرج في متحكمات stm32
- 3- التوابع المستخدمة من مكتبة HAL للتحكم بالمخارج الرقمية في متحكم STM32
- 4- بناء أول تطبيق لإضاءة ليد باستخدام متحكمات stm32 ومكتبة HAL
- 5- برمجة أقطاب الدخل في متحكمات stm32
- 6- التوابع المستخدمة من مكتبة HAL للتحكم بالمداخل الرقمية في متحكم STM32
- 7- بناء تطبيق لإضاءة ليد من خلال مفتاح لحظي باستخدام متحكمات stm32 ومكتبة HAL

الأدوات اللازمة للجلسة:

- لوحة Nucleo-64bit
- كبل Type-A to Mini-B

1. أنماط عمل أقطاب المتحكم GPIO Mode:

- كل متحكم من متحكمات STM32 له عدد محدد من الأقطاب حيث يعتمد هذا العدد على:
- نوعية Package للمتحكم (أي على حسب شكل المتحكم وعدد أطرافه)
 - عائلة المتحكم (F0, F1, F2... إلخ).
 - استخدام مذبذبات خارجية External Crystal سواء HSE أو LSE.
- فألا - GPIO هي الوسيلة التي يتم من خلالها عمل اتصال بين المتحكم والعالم الخارجي، ولكل متحكم عدد محدد من الـ GPIO يمكن أن تعمل كـ:
- 1- أقطاب خرج : يمكنها التحكم بالأجهزة المختلفة والتي قد تكون عبارة عن ليد ، ريليه، ترانزستور وغيرها....
 - 2- أقطاب دخل: لقراءة قيم الحساسات الرقمية والمفاتيح المختلفة
 - 3- وحدات طرفية: أغلب أقطاب المتحكم يمكن أن يكون لها عدة وظائف، بإمكانها مثلاً أن تعمل كأقطاب دخل/خرج أو كوظائف بديلة كوسائل الاتصال التسلسلي المختلفة (UART, I2C, SPI وغيرها)، أو كأقطاب دخل تـ شابهية ADC أو كأقطاب خرج تـ شابهية ، مقاطعات ، وغيرها من الوحدات الطرفية المختلفة، ولكن يتم ضبط أي قطب من أقطاب المتحكم بوظيفة واحدة فقط ضمن الكود (دخول/خرج/وحدة طرفية).



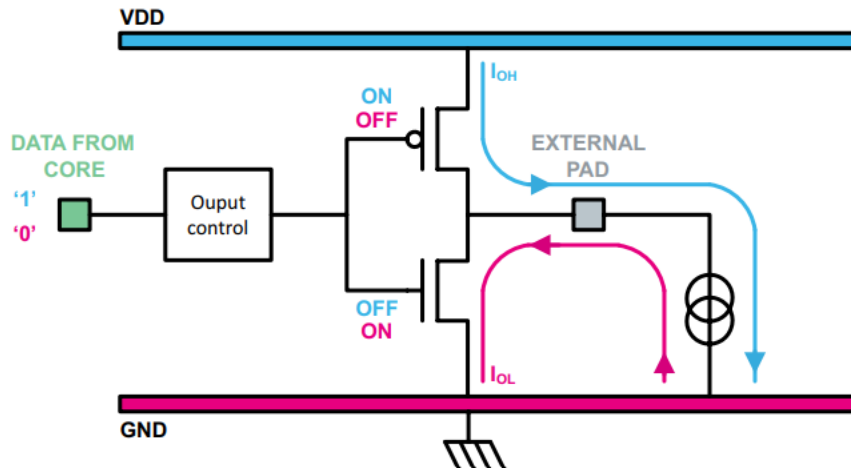
الشكل (1): أنماط عمل أقطاب المتحكم والبنية العادية لـ I/O لمتحكم STM32

2. برمجة أقطاب الخرج في متحكمات stm32 :

نميز نمطي عمل لكل قطب خرج من أقطاب المتحكم GPIO هما نمط العمل Push Pull ونمط العمل open drain(OD):

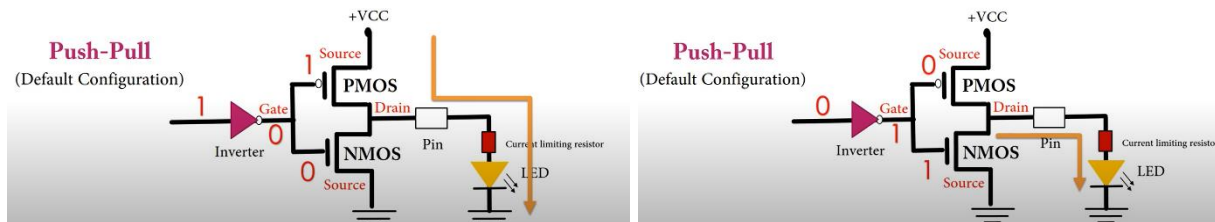
1. نمط **PUSH-PULL**: هو الحالة الافتراضية لأقطاب الخرج للمتحكم والأكثر استخداماً، حيث

يمكن في هذا النمط أن يعمل المتحكم كمنبع للتيار source أو كمصرف للتيار sink ، فيتم قيادة قطب الخرج من خلال ترانزستورين أحدهما من نوع PMOS والثاني من نوع NMOS كما هو موضح في الشكل التالي:



الشكل (1): نمط PUSH-PULL

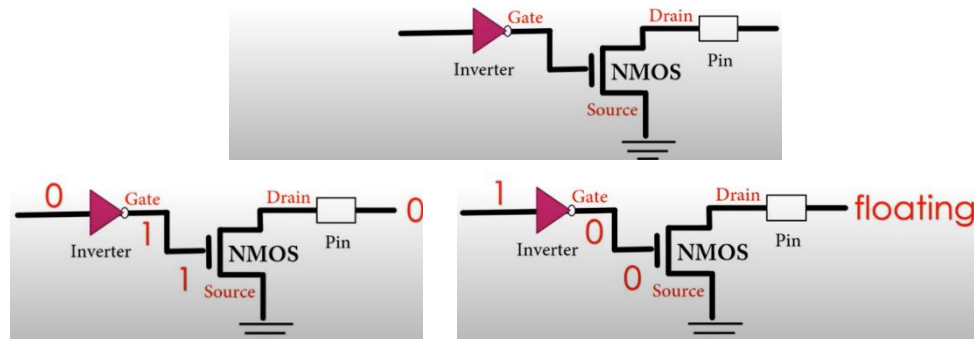
فبفرض تم وصل مـ صعد ليد مع قطب الإخراج للمتحكم المـ صغر، فعند تطبيق واحد منطقي على القطب يـ صبح الترانزستور PMOS بحالة تشغيل on وبالتالي يتم تطبيق جهد الـ - vcc على مصعد الليد ويضيء الليد، أما عند تطبيق صفر منطقي على القطب يـ صبح الترانزستور NMOS بحالة تشغيل on وبالتالي يتم تطبيق جهد الأرضي GND على مصعد الليد ويطفأ الليد كما هو موضح بالشكل التالي:



الشكل (2): نمط PUSH-PULL عند تطبيق الحالتين المنطقيتين على قطب الخرج

2. نمط **Open-Drain**: في هذا النمط يمكن للمتحكم أن يعمل كمـ صرف للتيار sink فقط، فيتم قيادة قطب الخرج من خلال ترانزستور واحد من نوع NMOS كما هو موضح في الأشكال التالية:

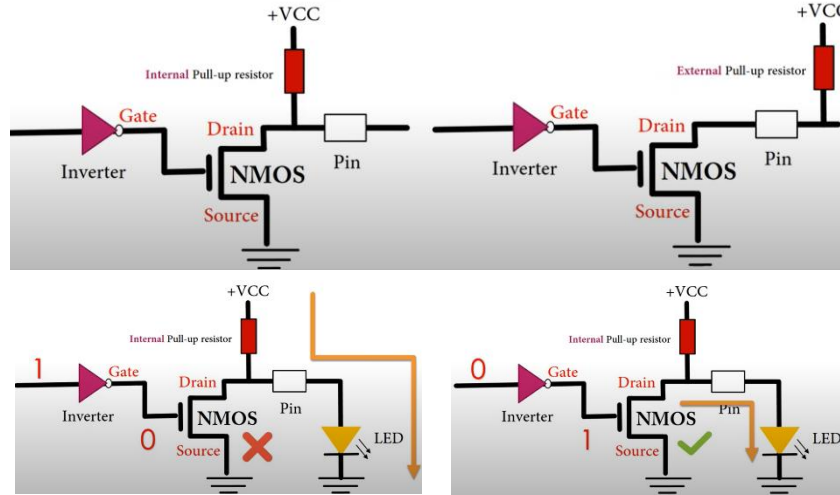
Open Drain



الشكل (3): نمط Open-Drain

نلاحظ من الأشكال السابقة أنه عند تطبيق صفر منطقي على قطب المتحكم يصبح الترانزستور بحالة توصيل on وبالتالي يتم توصيل الحمل مع الأرض GND ، أما في حالة تطبيق واحد منطقي على قطب المتحكم يصبح الترانزستور بحالة فصل off وبالتالي يصبح الجهد المطبق على الحمل عائم غير محدد floating لذا ولحل هذه المشكلة لابد من توصيل مقاومة رفع مع قطب المتحكم ، يمكن توصيل هذه المقاومة خارجياً أو يمكن تفعيل مقاومة الرفع الداخلية الموجودة ضمن المتحكم كما في الشكل التالي:

Open Drain with Pull-up Resistor -



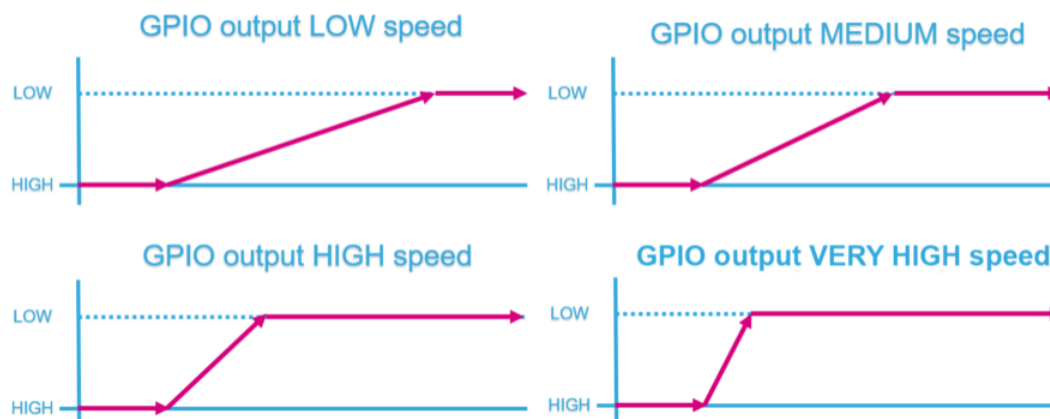
الشكل (4): نمط Open-Drain عند توصيل مقاومة رفع داخلية

تستخدم معظم التطبيقات التي تستخدم نمط Open-Drain مقاومات رفع خارجية، حيث تكون قيم مقاومات الرفع الداخلية غير كافية لها.

يعتبر نمط العمل open-drain مفيد عند ربط عدة بوابات أو عدة أقطاب معاً كما هو الحال عند استخدام I2C Bus ، حيث أثناء قيادة المتحكم المصغر للـ Bus يتم تطبيق جهد منخفض GND على قطب الخرج ، أما في حال عدم استخدام المتحكم المصغر للـ Bus يكون القطب في حالة ممانعة عالية high-impedance mode ويتم تطبيق جهد مرتفع Vcc على القطب، أيضاً إحدى الاستخدامات الهامة لهذا النمط من العمل هو عند قيادة عدة أجهزة من خلال قطب مقاطعة وحيد من المتحكم المصغر.

في حالتنا سنستخدم نمط الـ Push-Pull وبالتالي لن نحتاج لأي مقاومة رفع لذا سنختار الخيار No Pull-Up and Pull-Down.

أما عن خيار Maximum output speed فيعني سرعة انتقال الإشارة من الحالة المنخفضة low إلى الحالة المرتفعة HIGH وبالعكس وهو ما يسمى بزمن الـ rise time وزمن الهبوط fall time كما هو موضح بالشكل التالي:



الشكل (5): output speed

لكن يجب الانتباه أنه عند زيادة سرعة القطب output speed (أي تقليل زمن الصعود rise time) عندها سيزيد استرجار التيار وأيضاً سيزيد الضجيج الناتج، لذا يفضل اختيار السرعة المنخفضة LOW بشكل افتراضي ماعدا الحالات والتطبيقات التي تتطلب رفع السرعة .
في حالتنا قمنا باختيار السرعة المنخفضة LOW.
آخر خيار هو **user label** وهو اختيار اسم للقطب يعبر عن الجهاز الموصول معه، مثلاً سنسميه في حالتنا green_led أو led بإمكانك اختيار أي اسم تريد.
يمكن استرجار تيار من أي قطب من أقطاب المتحكم بحدود 25mA على ألا يتجاوز الاسترجار الكلي للتيار من جميع أقطاب المتحكم الـ 150mA كما هو موضح بالجدول التالي:

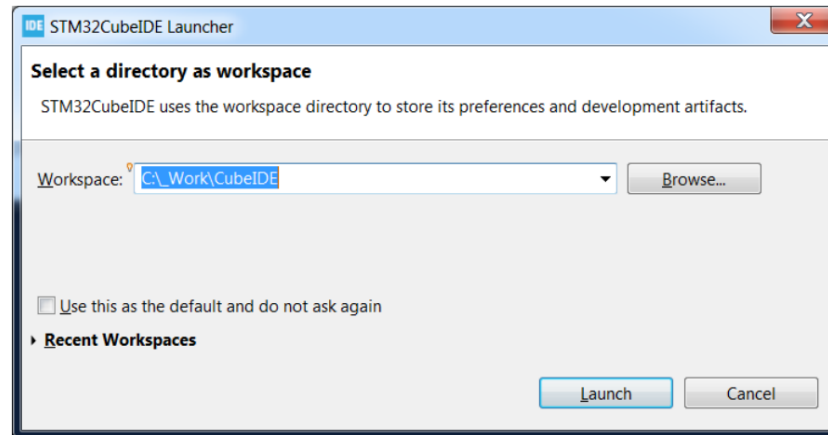
Symbol	Ratings	Max.	Unit
I_{VDD}	Total current into V_{DD}/V_{DDA} power lines (source) ⁽¹⁾	150	mA
I_{VSS}	Total current out of V_{SS} ground lines (sink) ⁽¹⁾	150	
I_{IO}	Output current sunk by any I/O and control pin	25	
	Output current source by any I/Os and control pin	-25	

الجدول (1):

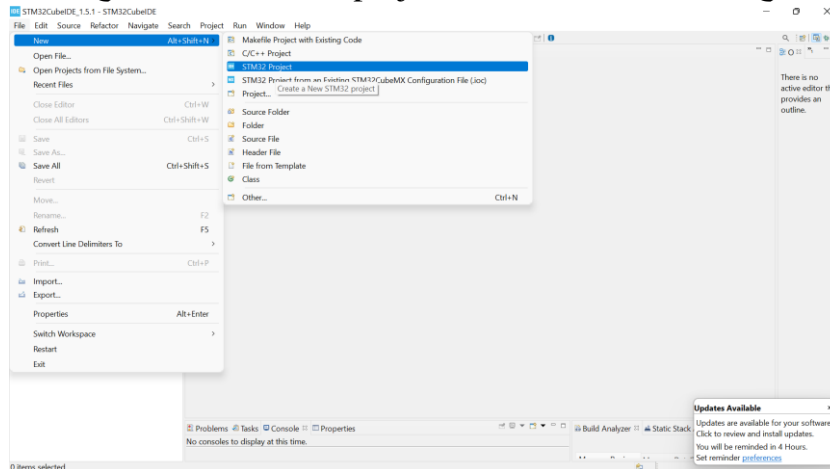
3. التوابع المستخدمة من مكتبة HAL للتحكم بالمخارج الرقمية في متحكم STM32:
- تم تصميم مكتبة HAL كي تستخلص عناوين الوحدات الطرفية المختلفة من الذاكرة، وإعطاء طريقة ثابتة و سهلة للتعامل مع الوحدات المختلفة، دون الحاجة للتعامل مع الم سجلات Registers المختلفة وللتسهيل على المبرمجين.
- لإعطاء واحد منطقي set أو صفر منطقي reset لقطب محدد من أي منفذ من منافذ المتحكم نقوم باستخدام التابع التالي من مكتبة HAL:

اسم المنفذ المراد التحكم بأحد أقطابه على سبيل المثال	رقم القطب المراد التحكم به على سبيل المثال	إعطاء واحد منطقي نكتب
GPIOA	GPIO PIN 5	GPIO_PIN_SET ولإعطاء صفر منطقي نكتب GPIO_PIN_RESET

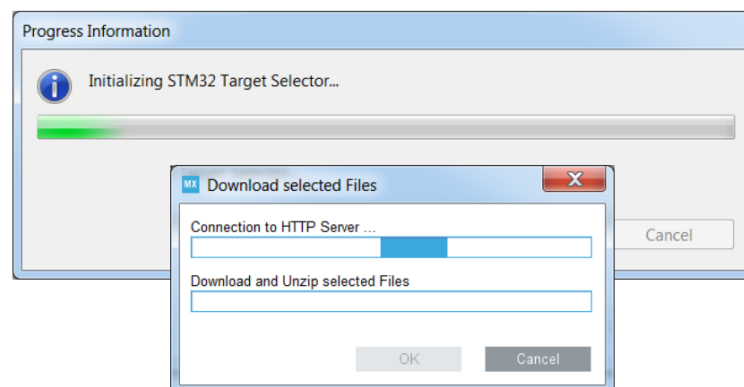
6



الشكل(7): فتح بيئة STM32cubeIDE
ثم نختار إنشاء مشروع جديد من file... new... STM32project.... كما هو موضح بالشكل التالي:

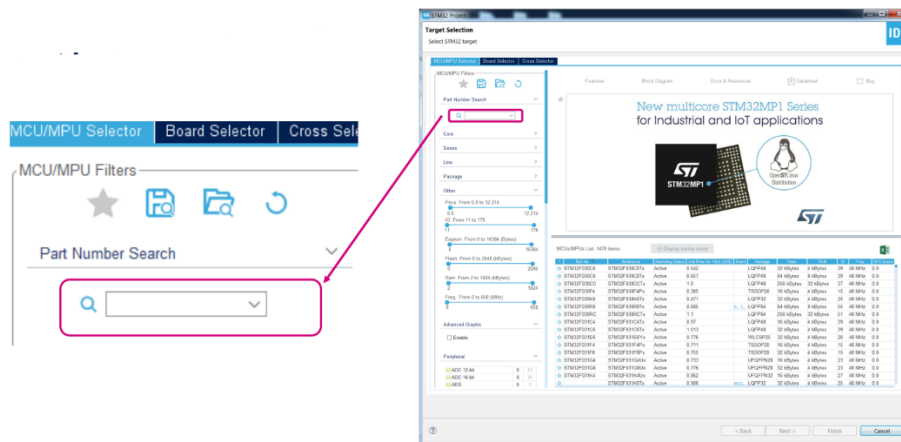


الشكل(8): إنشاء مشروع جديد
نلاحظ أن البرنامج يبحث عن تحديثات له عبر شبكة الانترنت في حال كان الحاسب متصل بالانترنت:



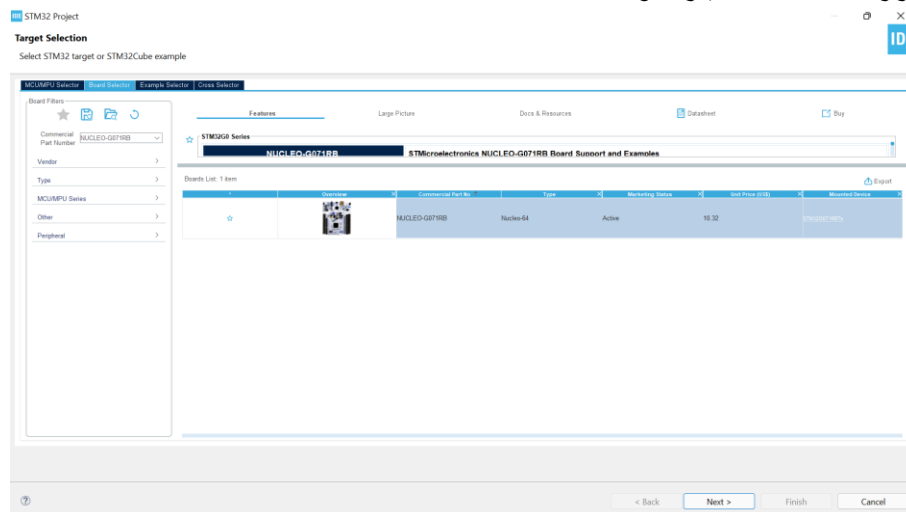
الشكل(9): البحث عن تحديثات للبرنامج

الخطوة الثانية : قم باختيار لوحة NUCLEO-G071RB من خلال كتابة اسم المتحكم الموجود في اللوحة وهو stm32G071RB أو كتابة اسم اللوحة Nucleo ضمن مربع البحث كما هو موضح بالشكل التالي:

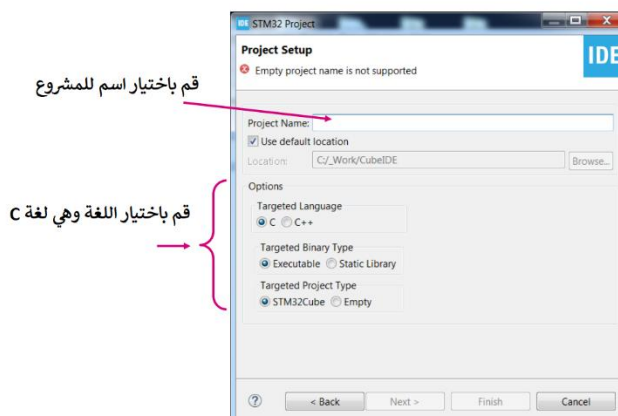


الشكل(10):اختيار المتحكم أو اللوحة التطويرية المناسبة

اضغط على الزر Next بعد اختيار اللوحة المناسبة:

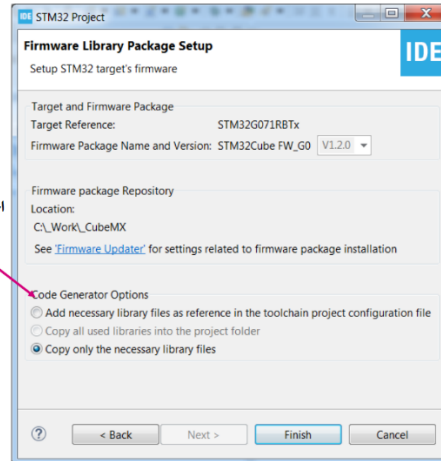


الشكل(11):اختيار لوحة Nucleo

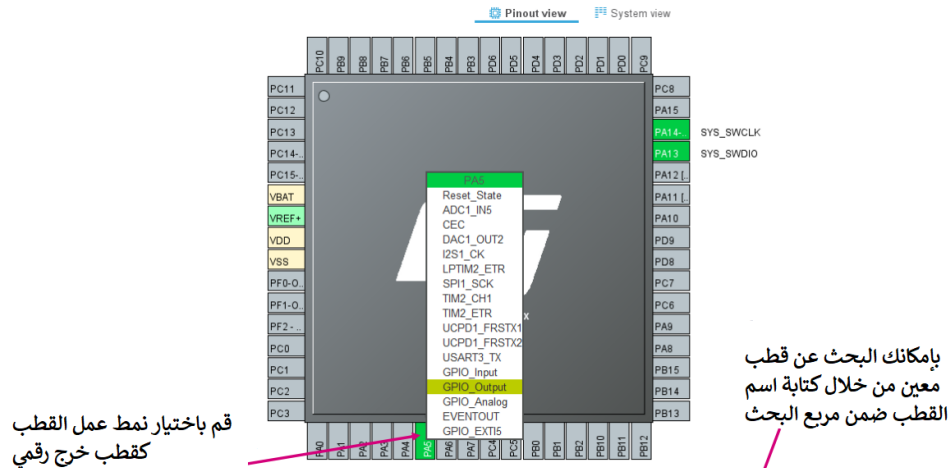


الشكل(12):ضبط إعدادات المشروع

اختر إضافة المكتبات الضرورية فقط للمشروع



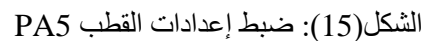
الشكل (13): إضافة المكتبات اللازمة فقط للمشروع
الخطوة الثالثة: قم بتحديد القطب PA5 كخرج كما هو موضح بالشكل التالي:



قم باختيار نمط عمل القطب كقطب خرج رقمي

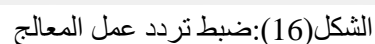
الشكل (14): ضبط نمط عمل القطب PA5

ثم قم بضبط الإعدادات الخاصة به:

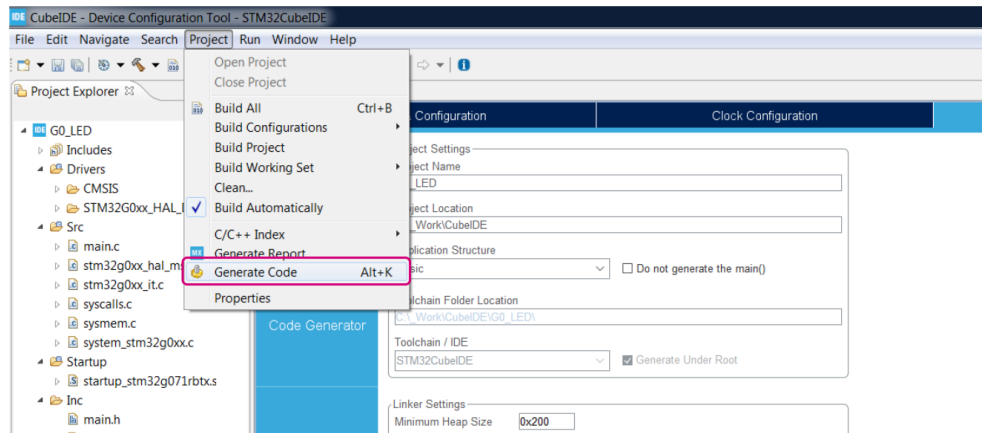


سنختار الحالة الافتراضية للقطب LOW أي يكون في حالة جهد منخفض افتراضياً، و سنختار نمط Output Push Pull وبدون استخدام مقومات رفع أو خفض No Pull-up and Pull-down و سنختار سرعة الخرج High وسنختار اسم للقطب على سبيل المثال LED_GREEN.

الخطوة الرابعة: قم بضبط تردد الساعة للتحكم واختر مصدر الساعة الداخلية HSI (لأنه لا يوجد كريستال X3 خارجية موصولة على اللوحة لكن بإمكانك وضع واحدة أنت) كما هو موضح بالشكل التالي:



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



الشكل (17): توليد الكود بشكل تلقائي

الخطوة الخامسة: نقوم بكتابة الكود المنا سب كي ي ضيء 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 (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 */
}
```

أو يمكن كتابة الكود باستخدام تعليمة Toggle بالشكل التالي:

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

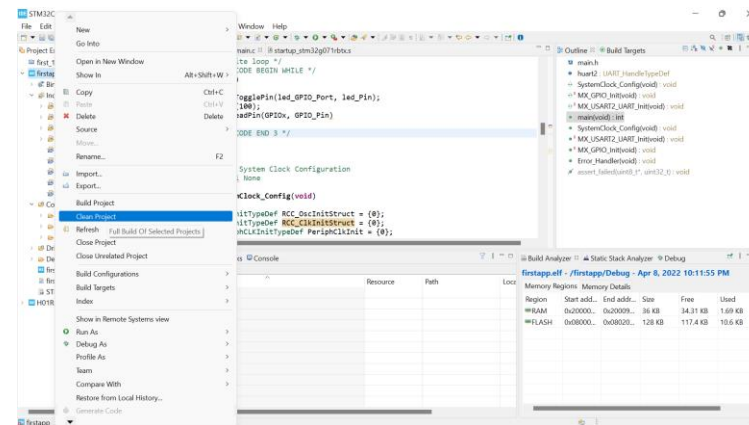
أيضاً يمكننا كتابة الكود باستخدام الاسم الذي اخترناه للقطب أثناء التصميم كما في الشكل التالي:

```

89  /* Initialize all configured peripherals */
90  MX_GPIO_Init();
91  /* USER CODE BEGIN 2 */
92
93  /* USER CODE END 2 */
94
95  /* Infinite loop */
96  /* USER CODE BEGIN WHILE */
97  while (1)
98  {
99      HAL_GPIO_TogglePin(LED_GREEN_GPIO_Port, LED_GREEN_Pin);
100     HAL_Delay(500);
101     /* USER CODE END WHILE */
102
103     /* USER CODE BEGIN 3 */
104 }
105 /* USER CODE END 3 */
106 }

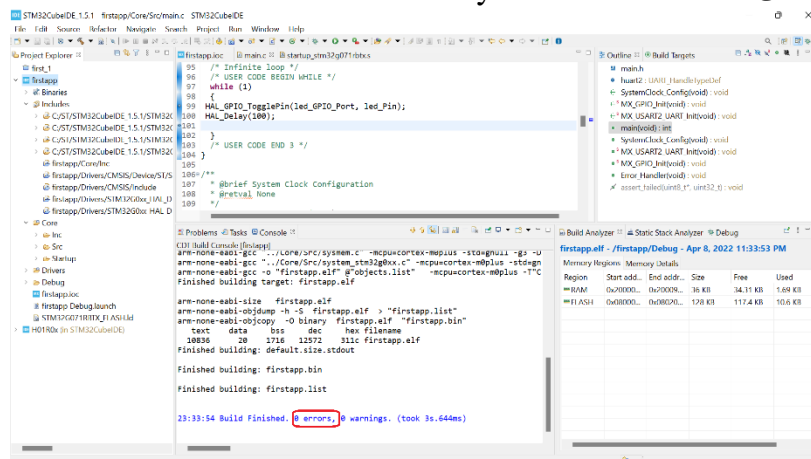
```

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



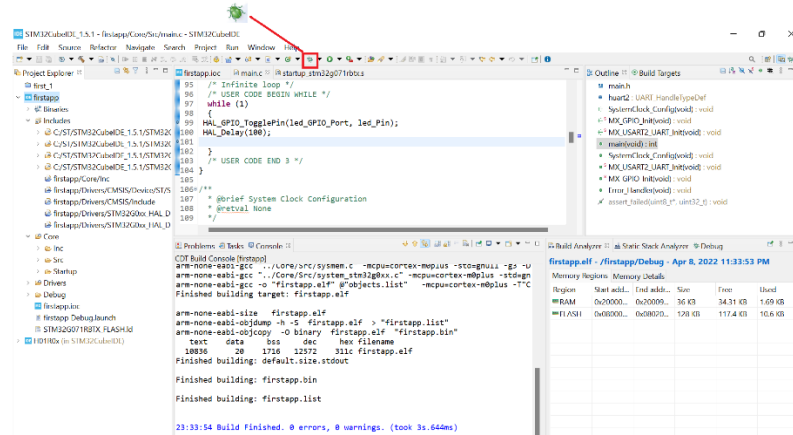
الشكل (18): تنظيف الكود ثم ترجمته

في حال خلو الكود من أخطاء الـ Syntax errors :

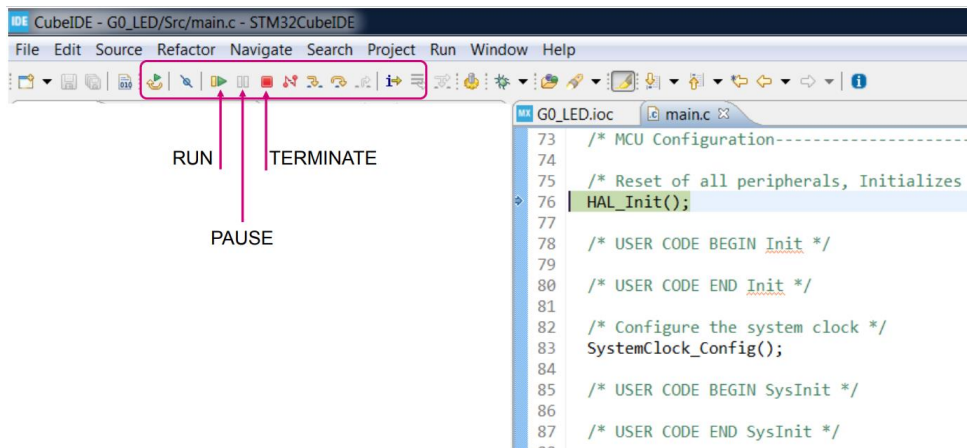


الشكل (19): التأكد من خلو الكود من الأخطاء

قم بالضغط على أيقونة الـ debug لتبدأ دارة ST-Link برفع الكود إلى المتحكم ثم البدء بـ debug لمراقبة الكود خطوة بخطوة وتصحيح الأخطاء البرمجية.



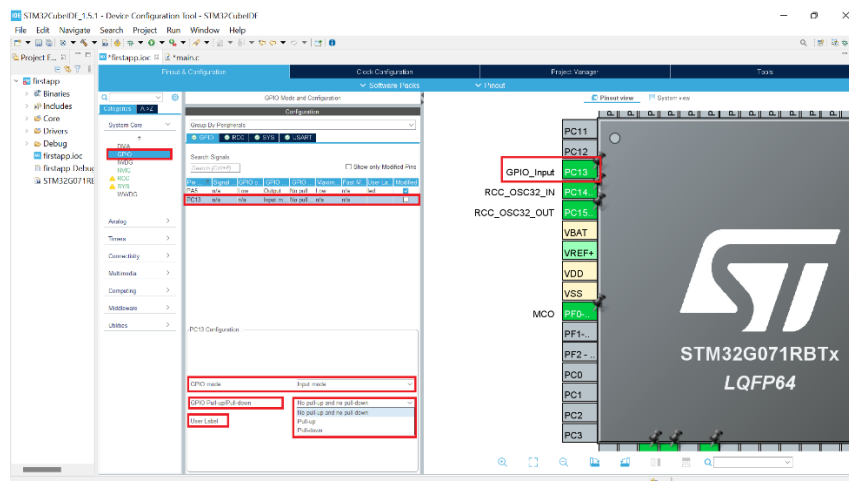
الشكل (20):فتح جلسة debug



الشكل (21):نافذة الـ debug

5. برمجة أقطاب الدخل في متحكمات stm32 :

لجعل أحد الأقطاب قطب دخل رقمي، نقوم بضبط الإعدادات المناسبة من خلال الواجهة الرسومية بالشكل التالي:



الشكل(22): ضبط الإعدادات المناسبة للدخل الرقمي

حيث نلاحظ وجود ثلاث خيارات للدخل الرقمي وهي:

- 1- عدم استخدام أي مقاومات للرفع أو الخفض No pull-up and no Pull-down
- 2- استخدام مقاومة رفع Pull-up
- 3- استخدام مقاومة خفض Pull-down

وأيضاً لدينا خيار بإعطاء اسم اختياري للقطب.

6. التوابع المستخدمة من مكتبة HAL للتحكم بالمداخل الرقمية في متحكم STM32:

نستخدم التابع التالي لمعرفة حالة الدخل الرقمي على أحد أقطاب المتحكم:

HAL_GPIO_ReadPin(GPIO_TypeDef *GPIOx, uint16_t GPIO_Pin);
حيث يعيد هذا التابع **GPIO_PIN_RESET** في حال كانت الحالة المنطقية للقطب في حالة جهد منخفض ، ويعيد **GPIO_PIN_SET** في حال كانت الحالة المنطقية للقطب في حالة جهد مرتفع.

مثال: لقراءة حالة الدخل الرقمي على القطب رقم 13 من المنفذ E نستخدم التابع التالي:

HAL_GPIO_ReadPin(GPIOE, GPIO_PIN_13);

7. بناء تطبيق لإضاءة ليد من خلال مفتاح لحظي باستخدام متحكمات stm32 ومكتبة HAL

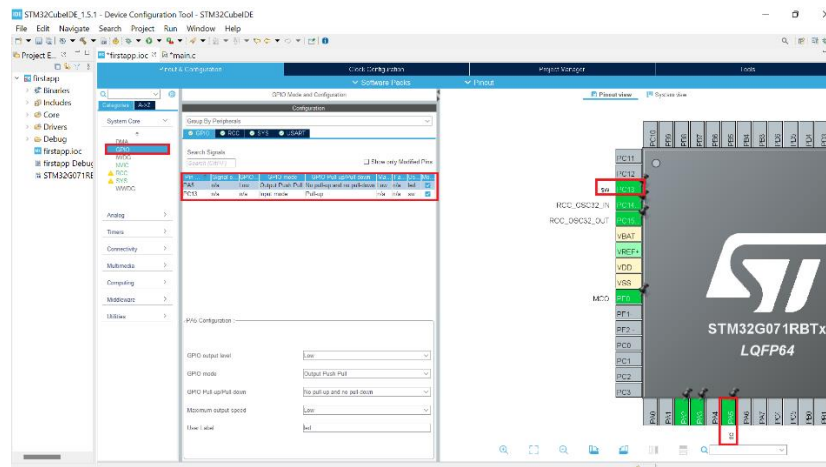
مطلوب كتابة الكود المنا سب بحيث يضيء الليد LD4 المو صول على القطب رقم 5 من المنفذ A عند الضغط على المفتاح B1 الموصول على القطب رقم 13 من المنفذ C من لوحة Nucleo.

الحل:

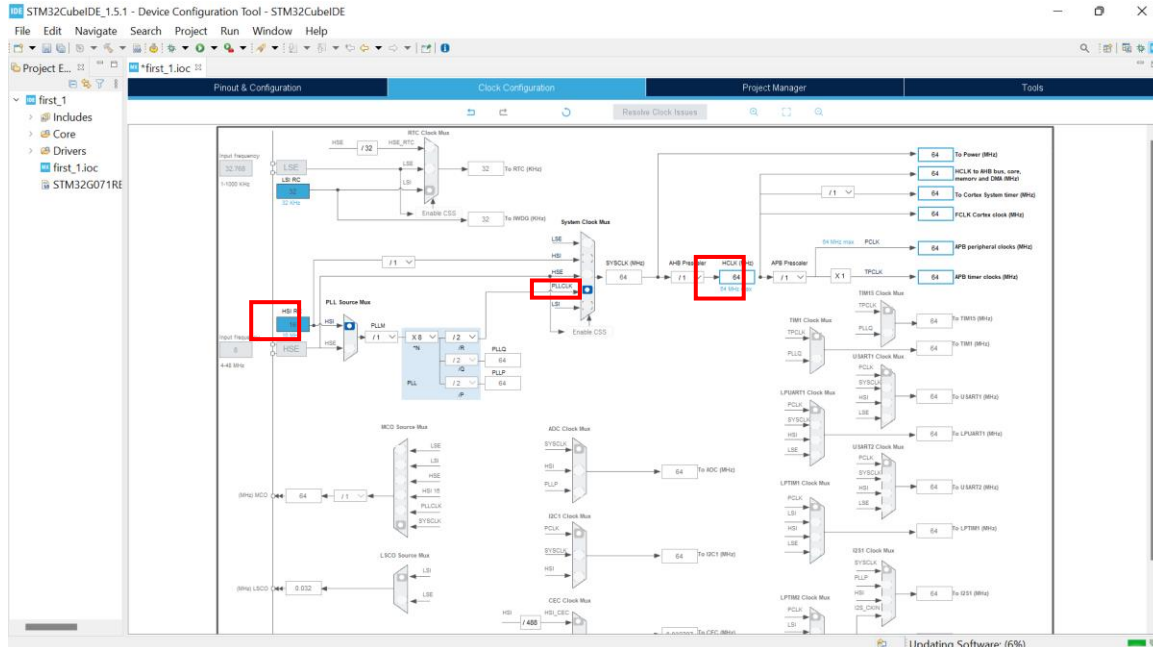
سنقوم باتتباع الخطوات التالية:

الخطوة الأولى:

نقوم ب ضبط الإعدادات المناسبة للم شروع حيث سنقوم ب ضبط القطب PA5 كقطب خرج من نوع Push-Pull وبدون استخدام مقاومات رفع أو خفض No pull-up no Pull-down وسنعطيه الاسم led ، أيضاً سنقوم ب ضبط القطب PC13 كقطب دخل و سنقوم بتفعيل مقاومة الرفع الداخلية Pull-up كما في الشكل التالي:



الشكل(23): ضبط الإعدادات المناسبة للمشروع
نقوم ب ضبط تردد ال ساعة للمتحكم ونختار م صدر ال ساعة الداخلية HSI (لأنه لا يوجد كريد ستالة X3 خارجية موصولة على اللوحة لكن بإمكانك وضع واحدة انت) كما هو موضح بالشكل التالي:



الشكل(24): ضبط تردد عمل المعالج
ن ضغط على Ctrl+s أو من Generate code...Project ، ليتم حفظ الم شروع وتوليد الكود وإضافة المكتبات اللازمة، ثم نقوم بفتح main.c لتعديل الكود بما يناسب المشروع.

الخطوة الثانية: نقوم بكتابة الكود المناسب ضمن main.c بالشكل التالي:

```

firstapp.ioc  main.c
93  /* USER CODE END 2 */
94
95  /* Infinite loop */
96  /* USER CODE BEGIN WHILE */
97  while (1)
98  {
99      /* USER CODE END WHILE */
100  if(HAL_GPIO_ReadPin(GPIOC, GPIO_PIN_13)==GPIO_PIN_RESET)
101  {
102      HAL_GPIO_WritePin(GPIOA, GPIO_PIN_5, GPIO_PIN_SET);
103  }
104  else
105      HAL_GPIO_WritePin(GPIOA, GPIO_PIN_5, GPIO_PIN_RESET);
106
107      /* USER CODE BEGIN 3 */
108  }
109  /* USER CODE END 3 */

```

الشكل(25):كتابة الكود

يمكننا أيضاً استخدام الأسماء التي اخترناها للأقطاب بدلاً من استخدام الأسماء الحقيقية لها كما في الشكل التالي:

الشكل (26): كتابة الكود باستخدام أسماء الأقطاب التي تم تحديدها

The screenshot shows the STM32CubeIDE 1.5.1 IDE interface. The top toolbar includes icons for file operations (File, Edit, Source, Refactor, Navigate, Search, Project, Run, Window, Help) and development tools (Compiler, Debugger, Memory, etc.). The left sidebar contains the Project Explorer, showing the project structure with folders like 'Inc', 'Src', and 'Debug'. The main editor area displays the 'main.c' file, which contains code for initializing the system clock and GPIO pins. The right sidebar shows the Outline view, listing the functions defined in the file. The bottom status bar indicates '23:33:54 Build Finished. 0 errors, 0 warnings. (took 3s.644ms)'.

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.