

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

مقرر المتحكمات المصغرة الجلسة الثانية

السنة الرابعة ميكاترونيك

2023/2022

الغاية من الجلسة

- 1- التعرف على التوابيع المستخدمة من مكتبة HAL للتحكم بالمخارج الرقمية في متحكم STM32
- 2- بناء تطبيق للتحكم بالمخارج الرقمية باستخدام متحكمات stm32 ومكتبة HAL
- 3- التحكم بالمخارج الرقمية للتحكم STM32 دون استخدام مكتبة HAL
- 4- إعادة التطبيق السابق دون استخدام مكتبة HAL
- 5- برمجة أقطاب الدخل في متحكمات stm32
- 6- التعريف بالدوال المستخدمة من مكتبة HAL للتحكم بالمداخل الرقمية في متحكم STM32
- 7- بناء تطبيق باستخدام متحكمات stm32 ومكتبة HAL
- 8- التعامل مع أقطاب الدخل في متحكمات stm32 بدون استخدام مكتبة HAL
- 9- إعادة التطبيق السابق دون استخدام مكتبة HAL
- 10- بناء عدة تطبيقات

- 1- التعرف على التوابيع 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 وسهلة للتعامل مع الوحدات المختلفة، دون الحاجة للتعامل مع المسجلات Registers المختلفة وللتسهيل على المبرمجين.
- إعطاء واحد منطقي set أو صفر منطقي reset لقطب محدد من أي منفذ من منافذ المتحكم نقوم باستخدام التابع التالي من مكتبة HAL:

```
void HAL_GPIO_WritePin(GPIO_TypeDef *GPIOx, uint16_t GPIO_Pin, GPIO_PinState PinState);
```



مثال 1: لكتابة واحد منطقي على القطب رقم 10 من المنفذ D نستخدم التابع التالي:

```
HAL_GPIO_WritePin(GPIOD, GPIO_PIN_10, 1);
```

مثال 2: لكتابة صفر منطقي على القطب رقم 5 من المنفذ A نستخدم التابع التالي:

```
HAL_GPIO_WritePin(GPIOA, GPIO_PIN_5, 0);
```

- لعكس الحالة المنطقية لأحد الأقطاب نستخدم التابع التالي:

```
HAL_GPIO_TogglePin(GPIO_TypeDef *GPIOx, uint16_t GPIO_Pin);
```

مثال: لعكس الحالة المنطقية للقطب رقم 5 من المنفذ A نستخدم التابع التالي:

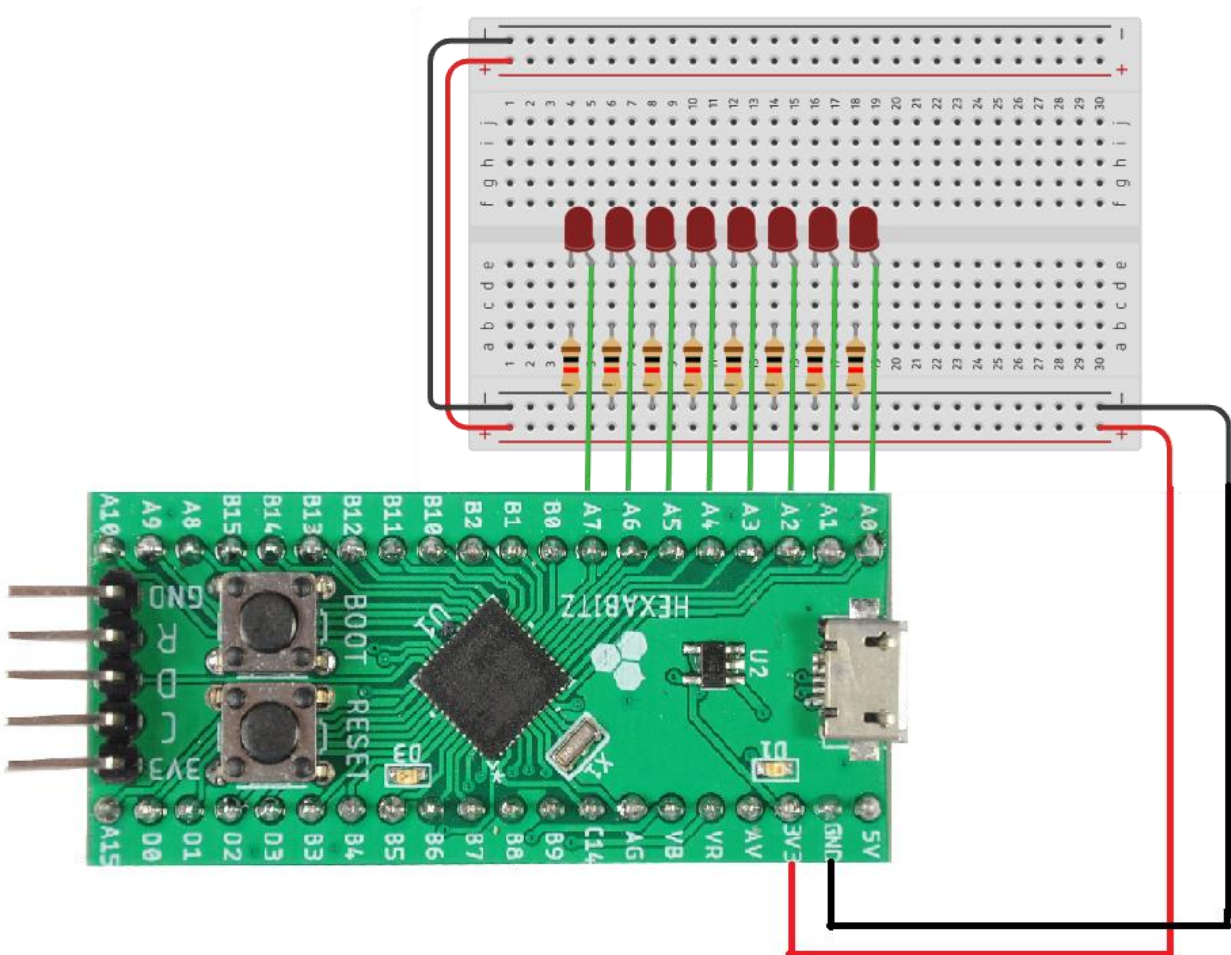
```
HAL_GPIO_TogglePin(GPIOA, GPIO_PIN_5);
```

- لإضافة تأخير زمني بالميللي ثانية نستخدم التابع التالي:

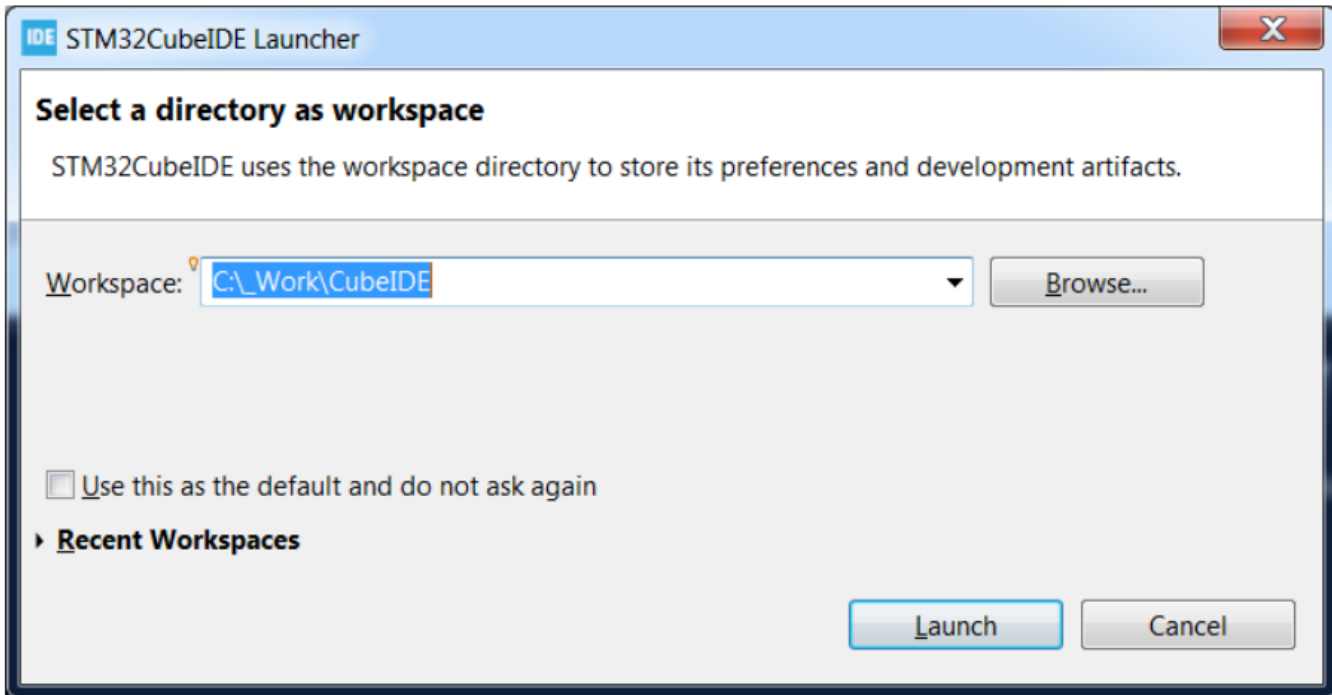
```
HAL_Delay(Milliseconds)
```

2- التطبيق الأول: تشغيل وإطفاء مجموعة لدات بشكل متسلسل حيث الزمن بين تشغيل الليد والذي يليه 250msec باستخدام البورد التطويري

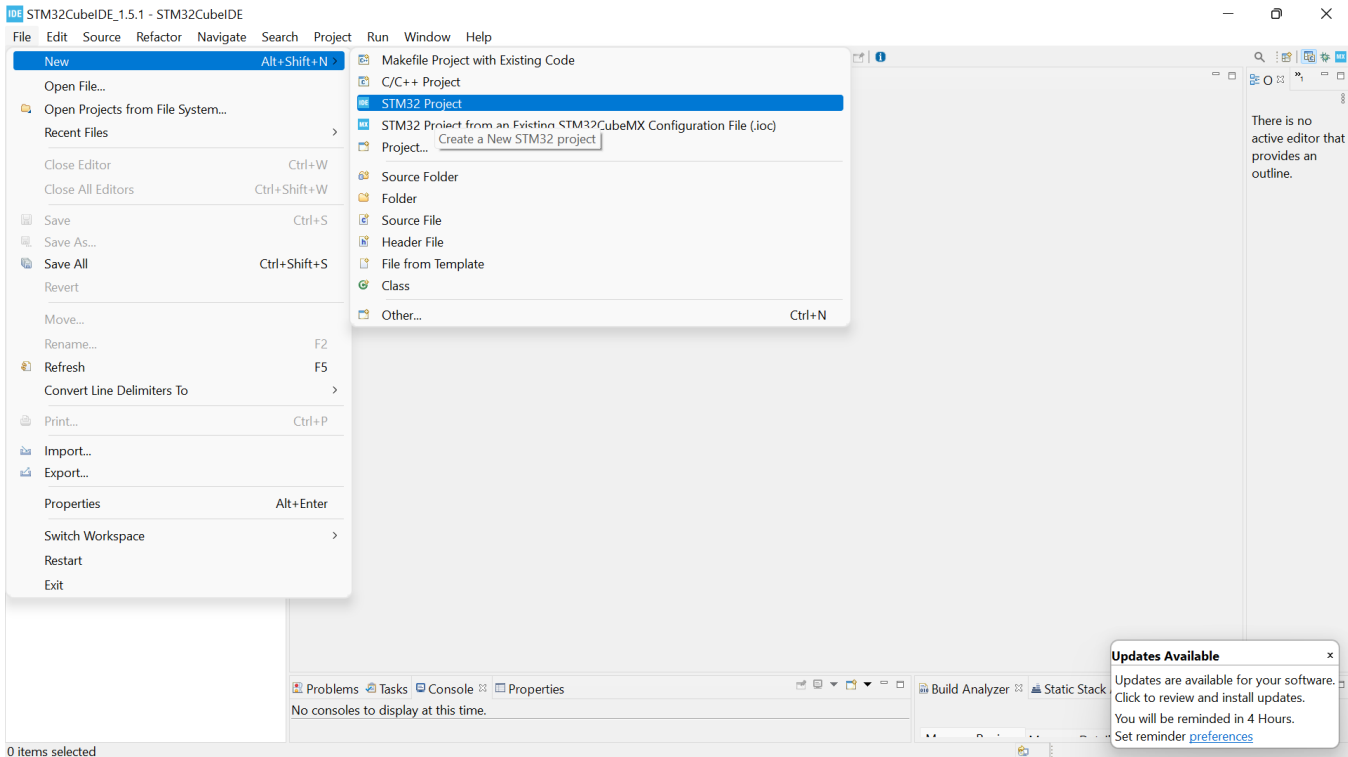
- رابط فيديو التطبيق على اليوتيوب: <https://youtu.be/pnm5KMzaAQQ>



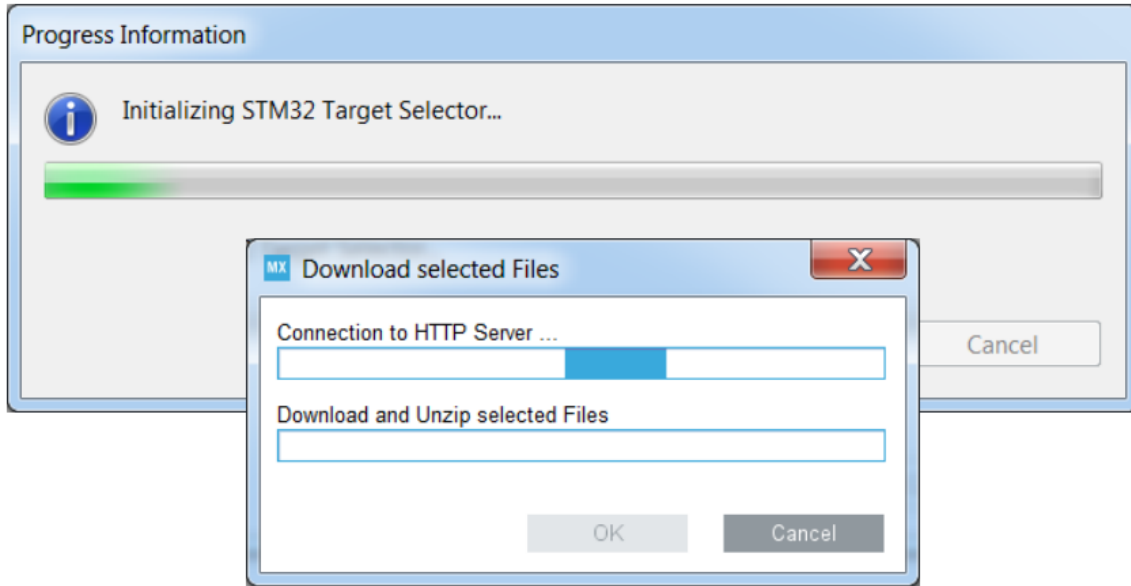
الخطوة الأولى: قم بفتح بيئة STM32cubeIDE



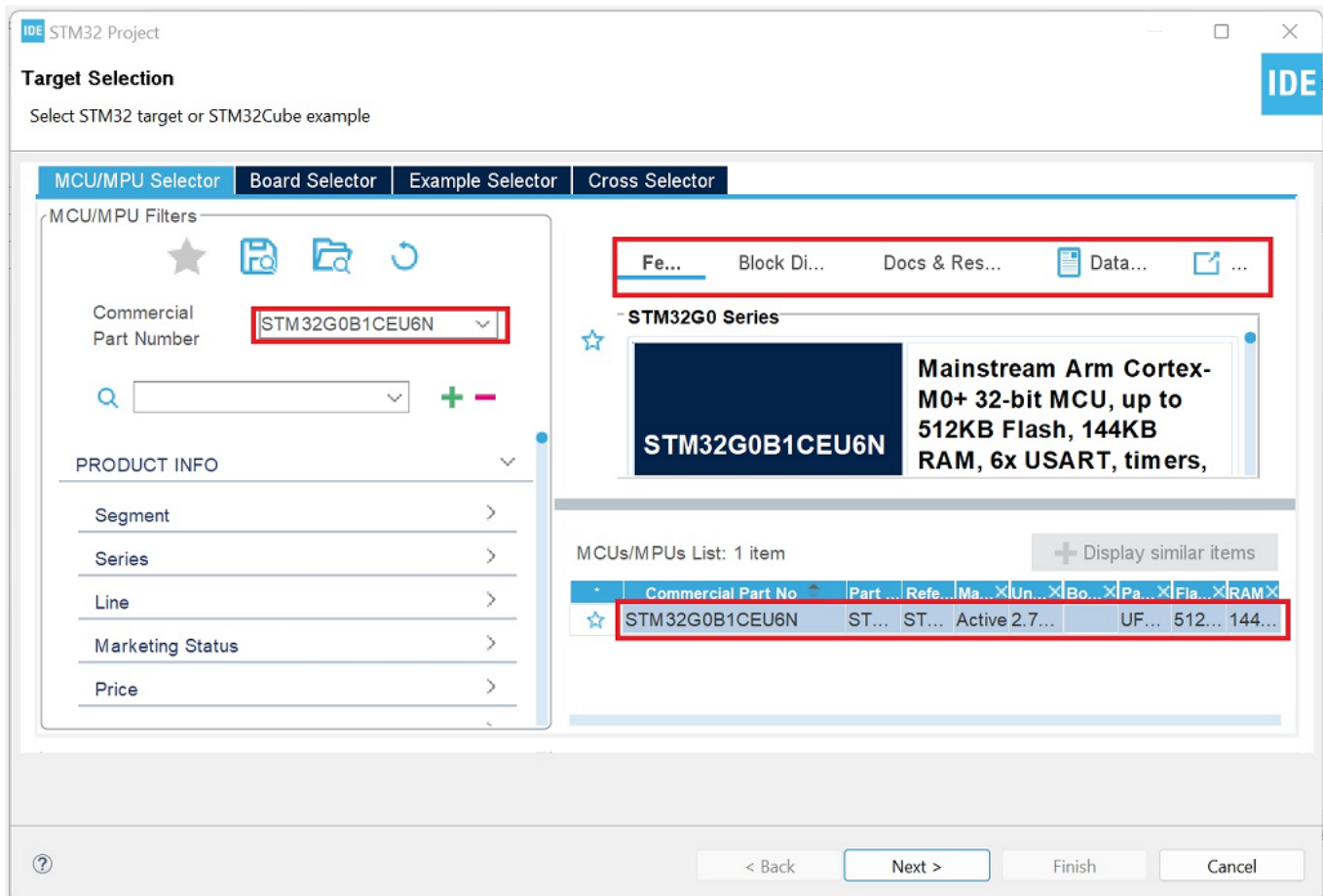
ثم نختار إنشاء مشروع جديد من file new... Stm32project.... كما هو موضح بالشكل التالي:

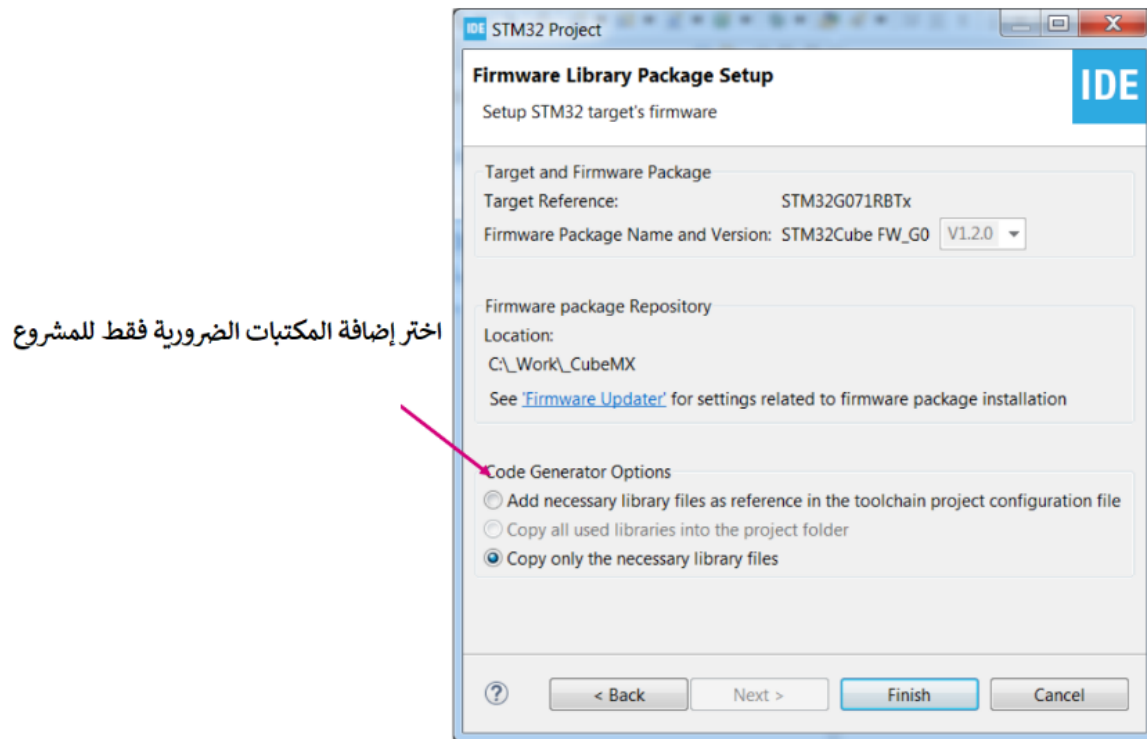
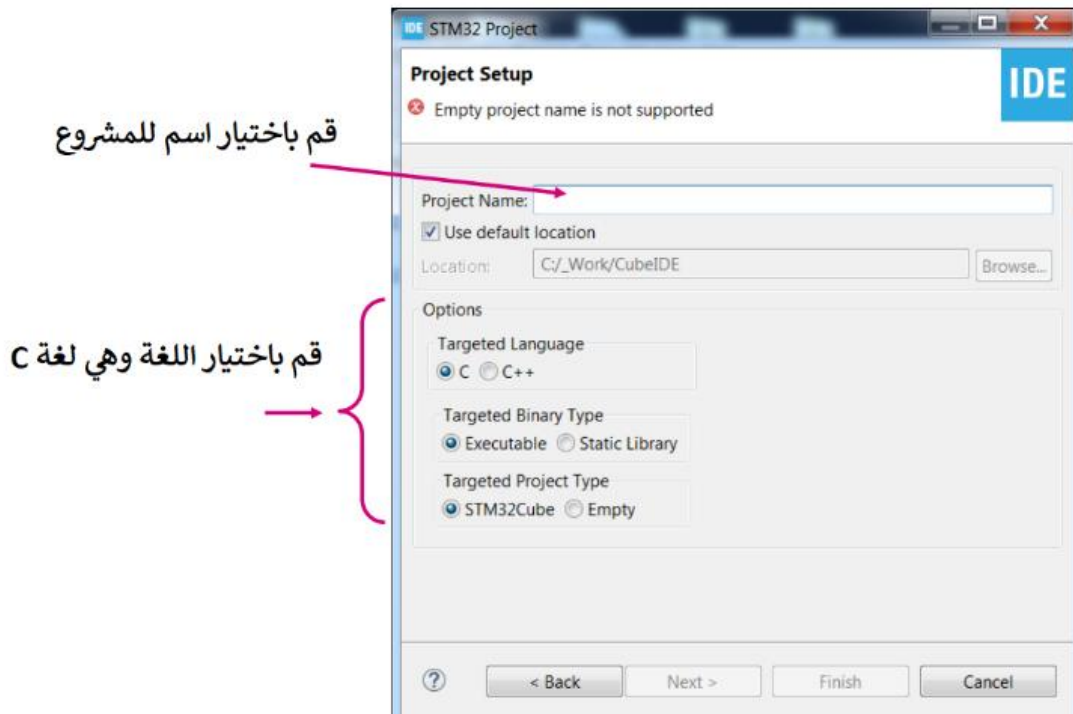


نلاحظ أن البرنامج يبحث عن تحديثات له عبر شبكة الانترنت في حال كان الحاسب متصل بالانترنت:

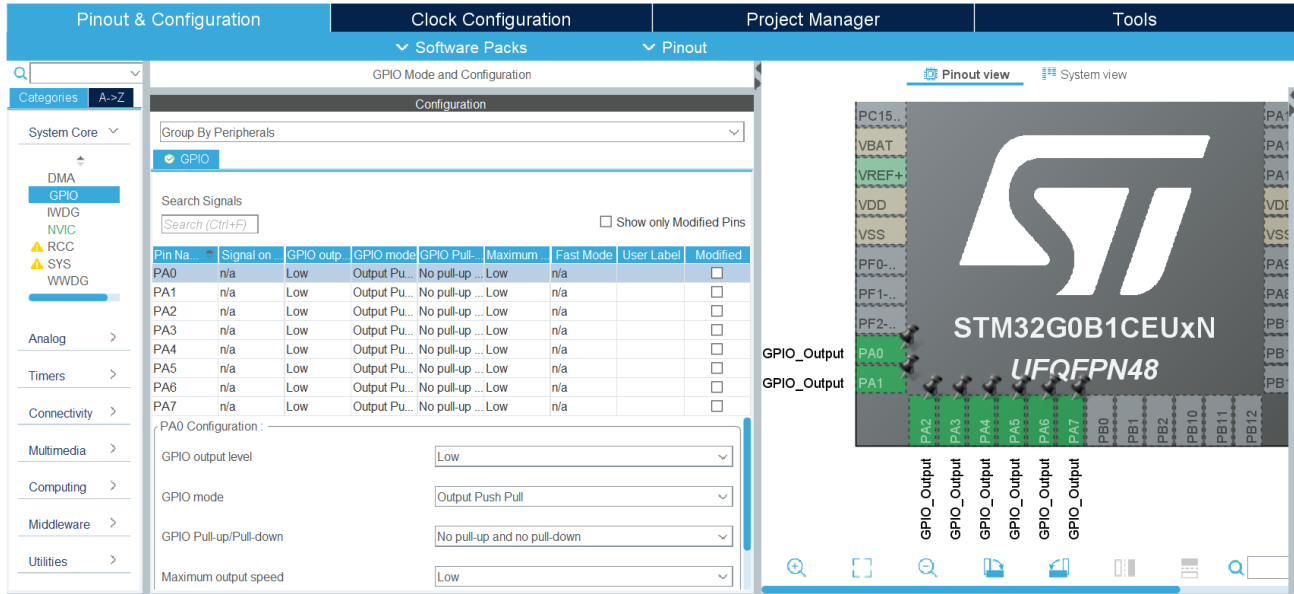


الخطوة الثانية : قم باختيار المتحكم من خلال كتابة اسم المتحكم STM32G0B1CEU6N ضمن مربع البحث كما هو موضح بالشكل التالي:

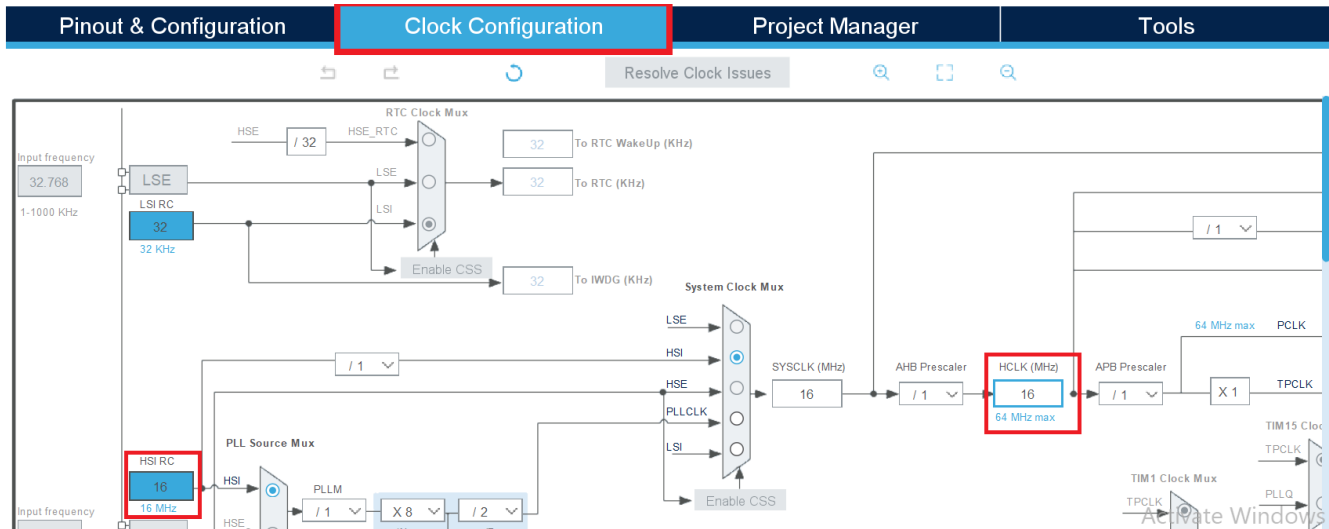




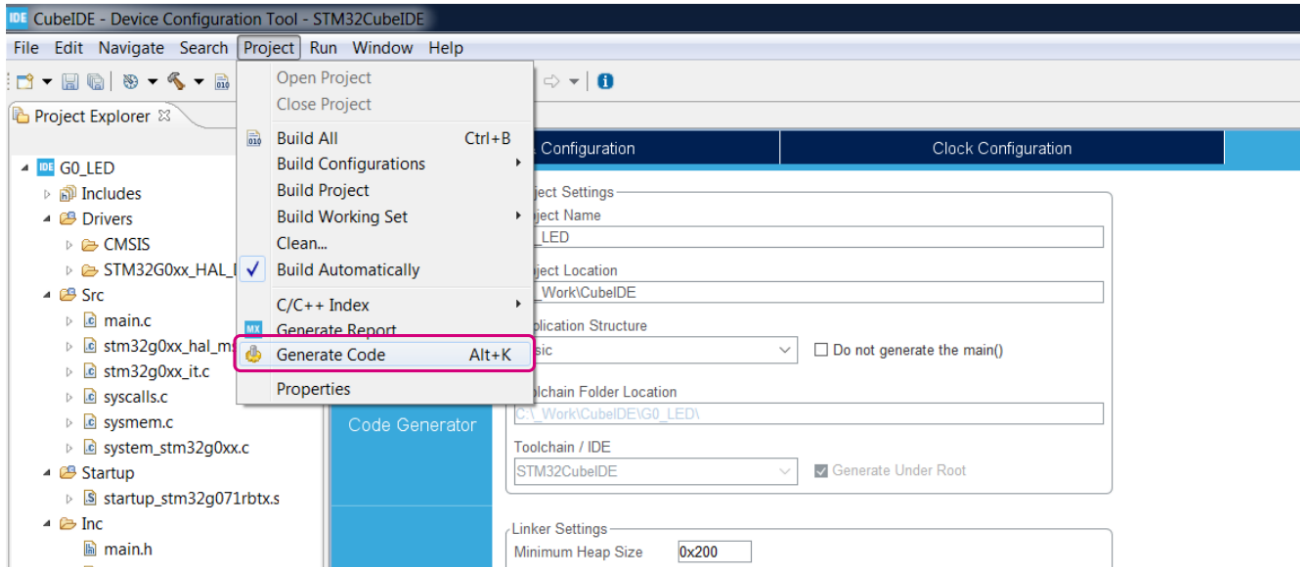
الخطوة الثالثة: قم بتحديد الأقطاب من PA0 حتى PA7 كخرج كما هو موضح بالشكل التالي:



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

```
#include "main.h"
void SystemClock_Config(void);
static void MX_GPIO_Init(void);
int main(void)
{
    HAL_Init();
    SystemClock_Config();
    MX_GPIO_Init();
    while (1)
    {
        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_0, 1);
        HAL_Delay(250);
        /**
        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_0, 0);
        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_1, 1);
        HAL_Delay(250);
        /**
        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_1, 0);
        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_2, 1);
        HAL_Delay(250);
        /**
        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_2, 0);
```

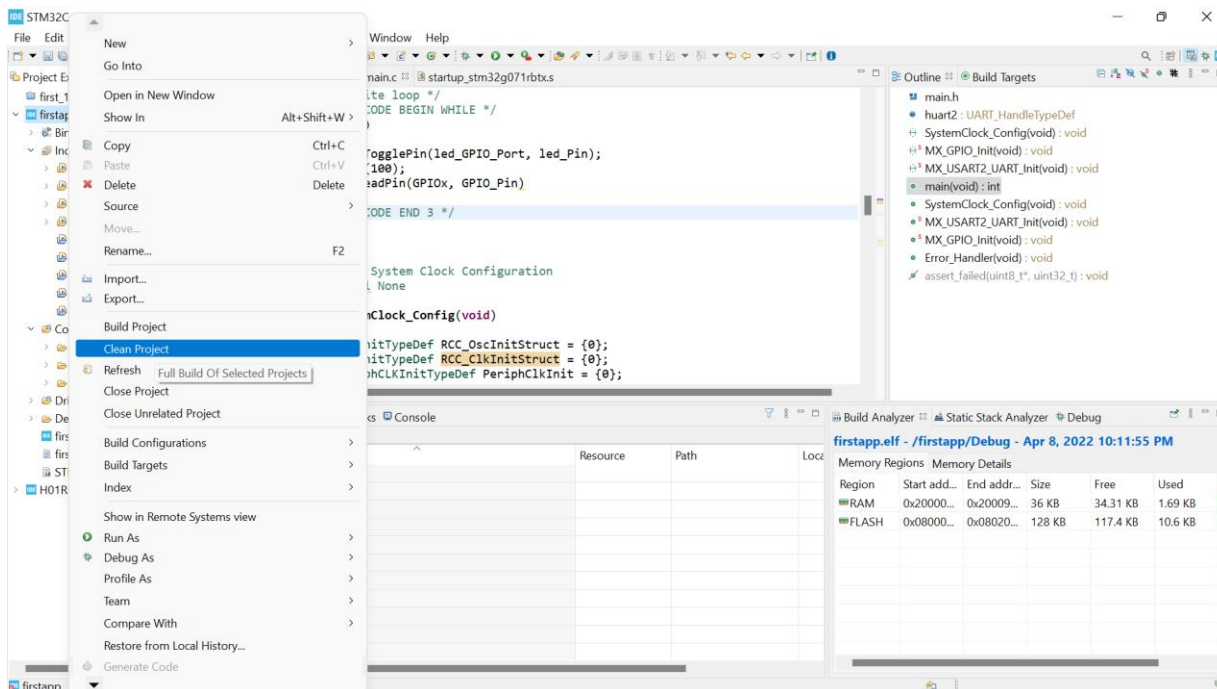


```

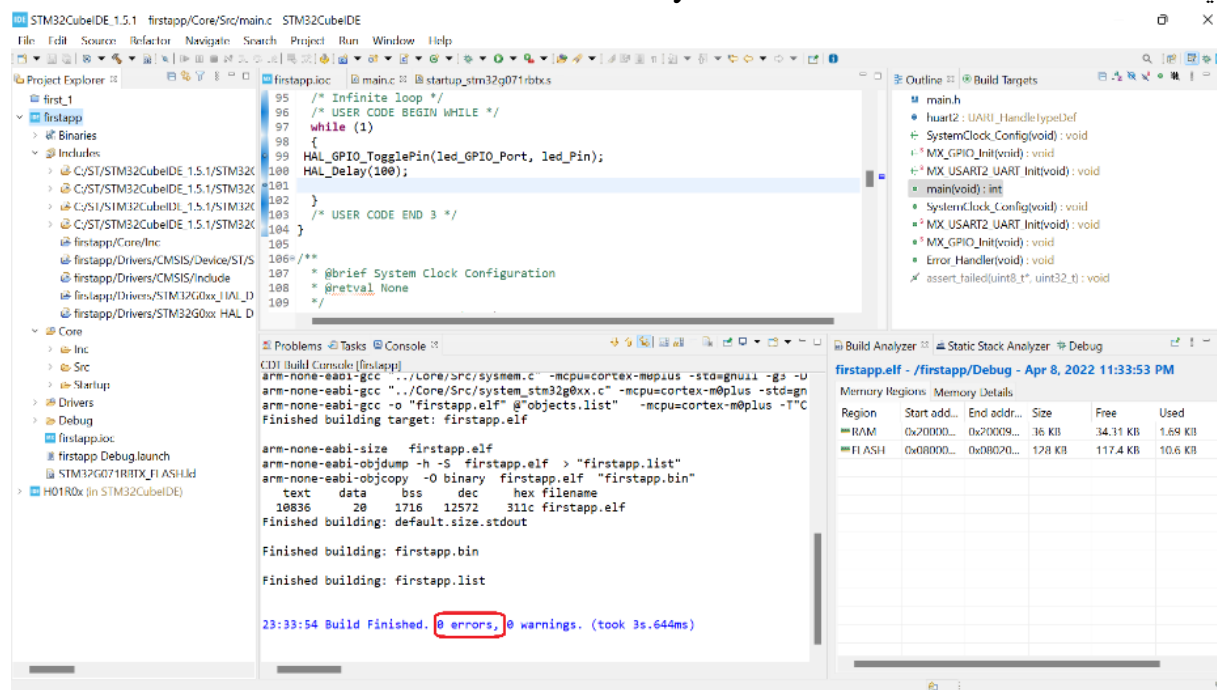
HAL_GPIO_WritePin(GPIOA, GPIO_PIN_3, 1);
HAL_Delay(250);
//*****
HAL_GPIO_WritePin(GPIOA, GPIO_PIN_3, 0);
HAL_GPIO_WritePin(GPIOA, GPIO_PIN_4, 1);
HAL_Delay(250);
//*****
HAL_GPIO_WritePin(GPIOA, GPIO_PIN_4, 0);
HAL_GPIO_WritePin(GPIOA, GPIO_PIN_5, 1);
HAL_Delay(250);
//*****
HAL_GPIO_WritePin(GPIOA, GPIO_PIN_5, 0);
HAL_GPIO_WritePin(GPIOA, GPIO_PIN_6, 1);
HAL_Delay(250);
//*****
HAL_GPIO_WritePin(GPIOA, GPIO_PIN_6, 0);
HAL_GPIO_WritePin(GPIOA, GPIO_PIN_7, 1);
HAL_Delay(250);
//*****
HAL_GPIO_WritePin(GPIOA, GPIO_PIN_7, 0);
HAL_Delay(250); } }

```

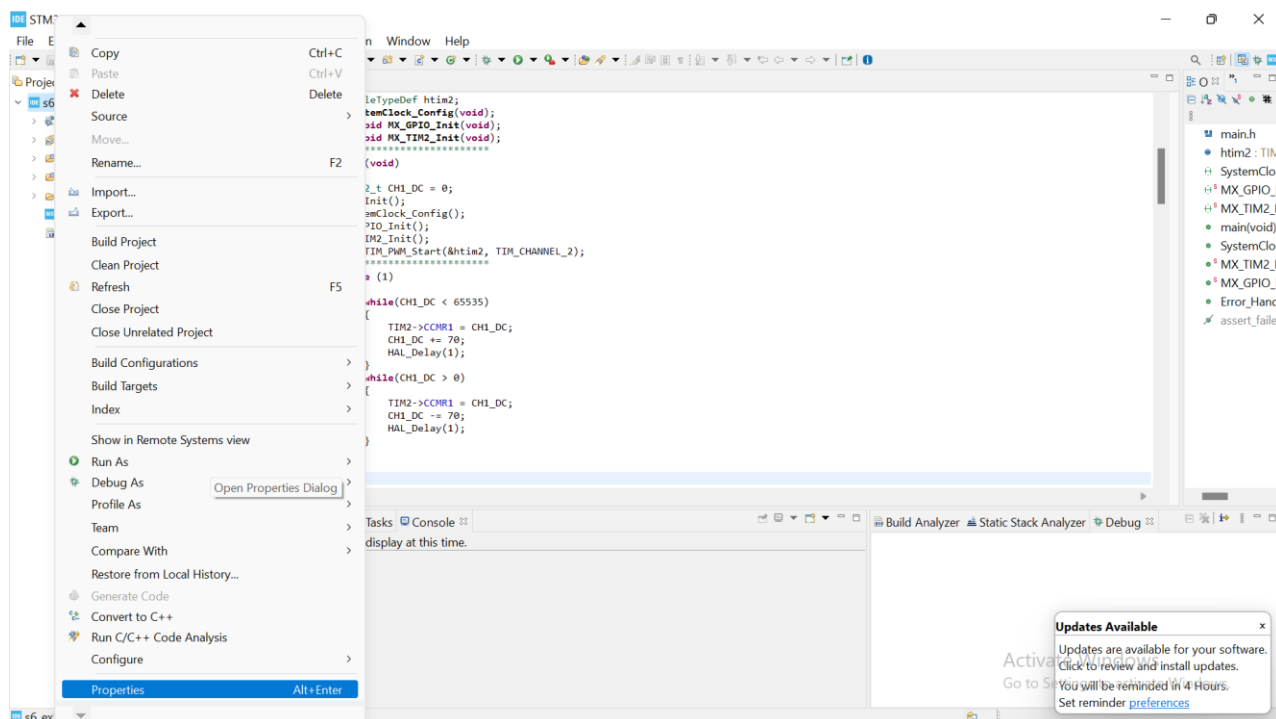
الخطوة السادسة: قم بالنقر بزر الفأرة الأيمن على اسم المشروع ثم اختر Clean project ثم Build project لتتم عملية ترجمة الكود للصيغة الثنائية والتأكد من خلو الكود من الأخطاء اللغوية Syntax errors



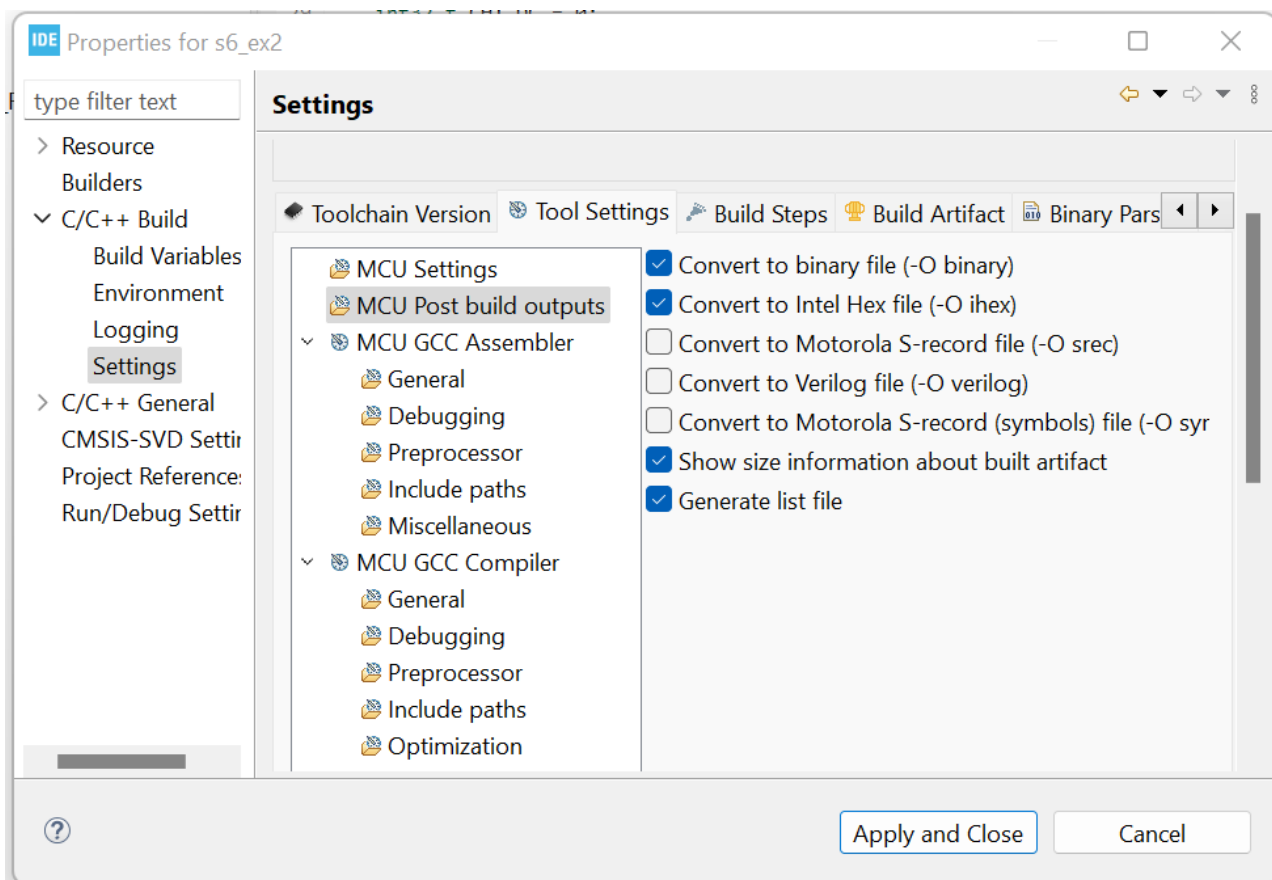
في حال خلو الكود من أخطاء الـ Syntax errors :



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



واختر مايلي:



الخطوة السابعة: قم باستخدام STM32CubeProgrammer برفع الكود للتحكم الموجود على البورد

3- التحكم بالمخارج الرقمية للتحكم STM32 دون استخدام مكتبة HAL:

هناك مجموعة من المسجلات تستخدم للتحكم بالمخارج الرقمية لمتحكم STM32 سنكتفي فقط بذكر المسجل المسؤول عن عمل set/reset للمنفذ أو لأحد الأقطاب الموجودة فيه (ولمزيد من المعلومات عن باقي المسجلات بإمكانك الرجوع إلى الـ datasheet الخاصة بالمتحكم).

7.4.6 GPIO port output data register (GPIOx_ODR) (x = A..H)

Address offset: 0x14

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ODR15	ODR14	ODR13	ODR12	ODR11	ODR10	ODR9	ODR8	ODR7	ODR6	ODR5	ODR4	ODR3	ODR2	ODR1	ODR0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits 31:16 Reserved, must be kept at reset value.

Bits 15:0 **ODRy**: Port output data (y = 0..15)

These bits can be read and written by software.

Note: For atomic bit set/reset, the ODR bits can be individually set and reset by writing to the GPIOx_BSRR register (x = A..H).

- نلاحظ أن المسجل GPIOx_ODR عبارة عن مسجل بطول 32bit ، لكن البتات 16:31 غير مستخدمة ويجب وضع صفر منطقي في كل منها، أما البتات من 0:15 فهي تعبر عن حالة أقطاب المنفذ المقابل لها، فعلى سبيل المثال لكتابة واحد منطقي على القطب رقم 5 من المنفذ A نكتب:

`GPIOA->ODR = 0x0020; // Set the Pin PA5`

- كما يمكن جعل المنفذ بالكامل بحالة set من خلال كتابة :

`GPIOA->ODR = 0xFFFF; // Set the PORTA HIGH`

- كما يمكن جعل المنفذ بالكامل بحالة reset من خلال كتابة :

`GPIOA->ODR = 0x0000; // Reset the PORTA`

4- إعادة التطبيق الأول دون استخدام مكتبة HAL

قم بإعادة الخطوات السابقة مع تعديل فقط الكود ليصبح بالشكل التالي:

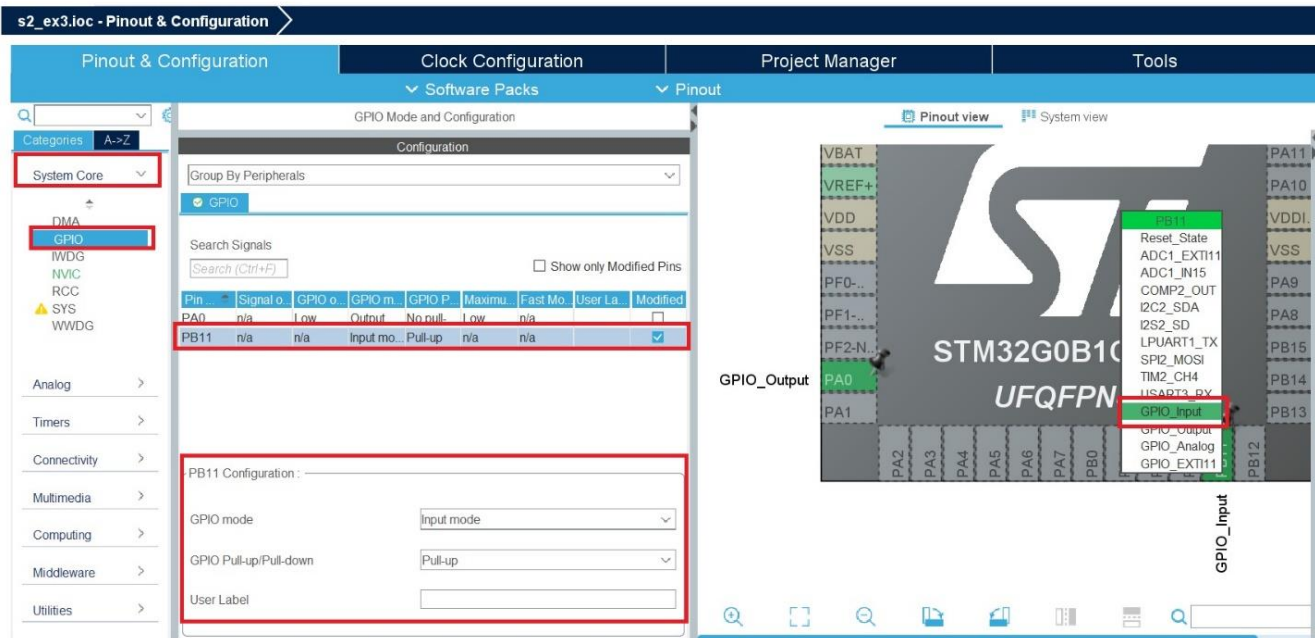
```

#include "main.h"
void SystemClock_Config(void);
static void MX_GPIO_Init(void);
int main(void)
{
    HAL_Init();
    SystemClock_Config();
    MX_GPIO_Init();
    while (1)
    {
        GPIOA->ODR = 0X0001;
        HAL_Delay(250);
        /**/
        GPIOA->ODR = 0X0002;
        HAL_Delay(250);
        /**/
        GPIOA->ODR = 0X0004;
        HAL_Delay(250);
        /**/
        GPIOA->ODR = 0X0008;
        HAL_Delay(250);
        /**/
        GPIOA->ODR = 0X0010;
        HAL_Delay(250);
        /**/
        GPIOA->ODR = 0X0020;
        HAL_Delay(250);
        /**/
        GPIOA->ODR = 0X0040;
        HAL_Delay(250);
        /**/
        GPIOA->ODR = 0X0080;
        HAL_Delay(250); } }

```

5- برمجة أقطاب الدخل في متحكمات stm32 :

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



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- الدوال المستخدمة من مكتبة HAL للتحكم بالمدخل 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);
```

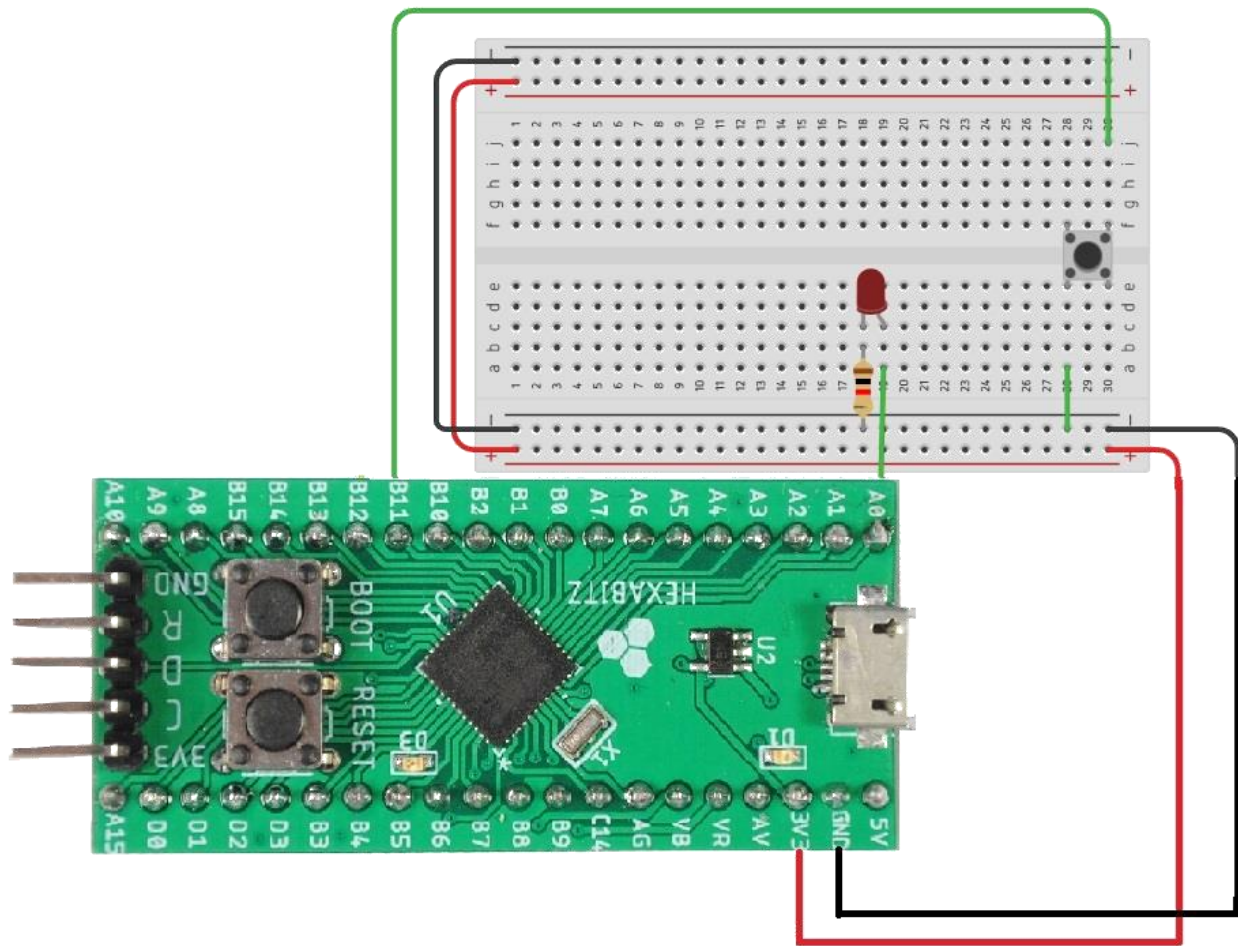
حيث يعيد هذا التابع 0 في حال كانت الحالة المنطقية للقطب في حالة جهد منخفض ويعيد 1 في حال كانت الحالة المنطقية للقطب في حالة جهد مرتفع.

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

```
HAL_GPIO_ReadPin(GPIOE, GPIO_PIN_13);
```

7- التطبيق الثاني: إضاءة ليد الموصول على القطب رقم 0 من المنفذ A من خلال مفتاح لحظي الموصول على القطب رقم 11 من المنفذ B باستخدام متحكمات stm32 ومكتبة HAL وباستخدام البورد التطويري

رابط فيديو التطبيق على اليوتيوب: <https://youtu.be/Th46hhtfAvM>

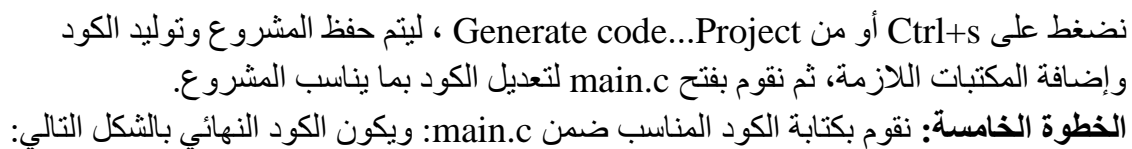
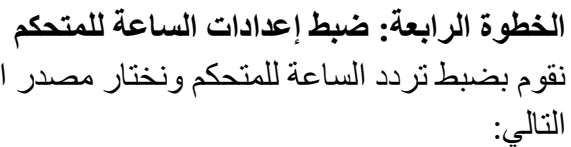


سنقوم باتتباع الخطوات التالية:

نقوم بإعادة الخطوات الأولى والثانية كما هي في التطبيق الأول

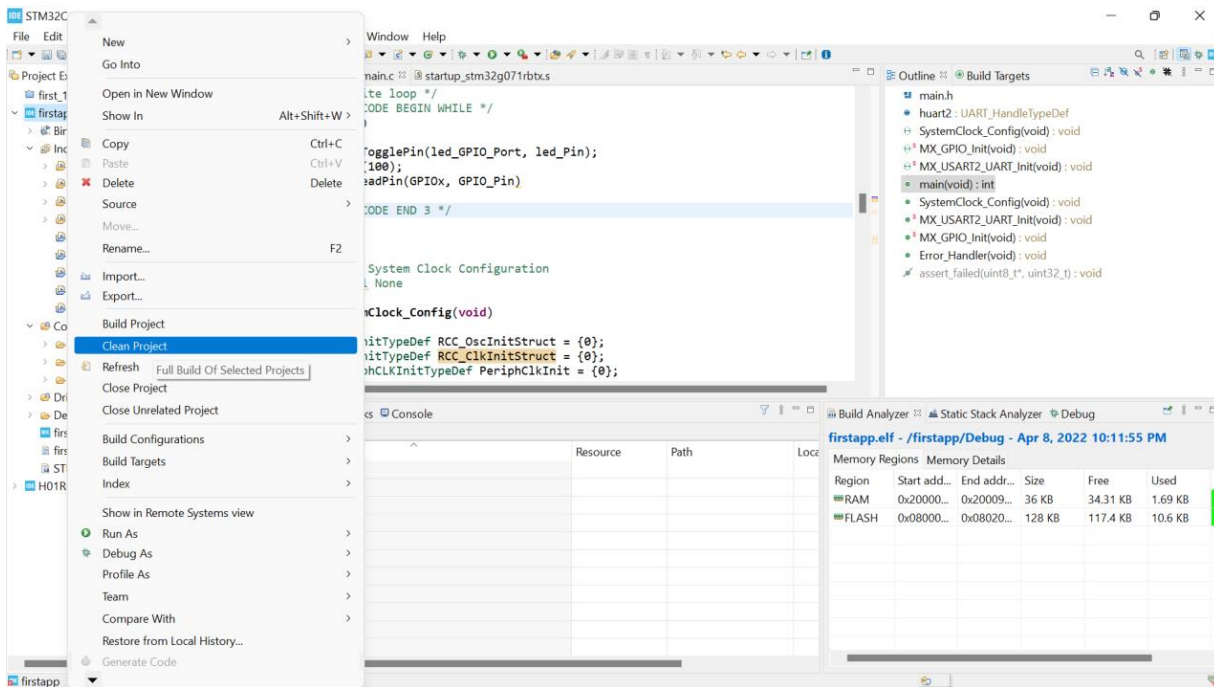
الخطوة الثالثة:

نقوم بضبط الإعدادات المناسبة للمicrocontroller حيث سنقوم بضبط القطب PA0 كقطب خرج من نوع Push-Pull وبدون استخدام مقاومات رفع أو خفض No pull-up no Pull-down و سنعطيه الاسم led ، أيضاً سنقوم بضبط القطب PB11 كقطب دخل و سنقوم بتفعيل مقاومة الرفع الداخلية Pull-up كما في الشكل التالي:

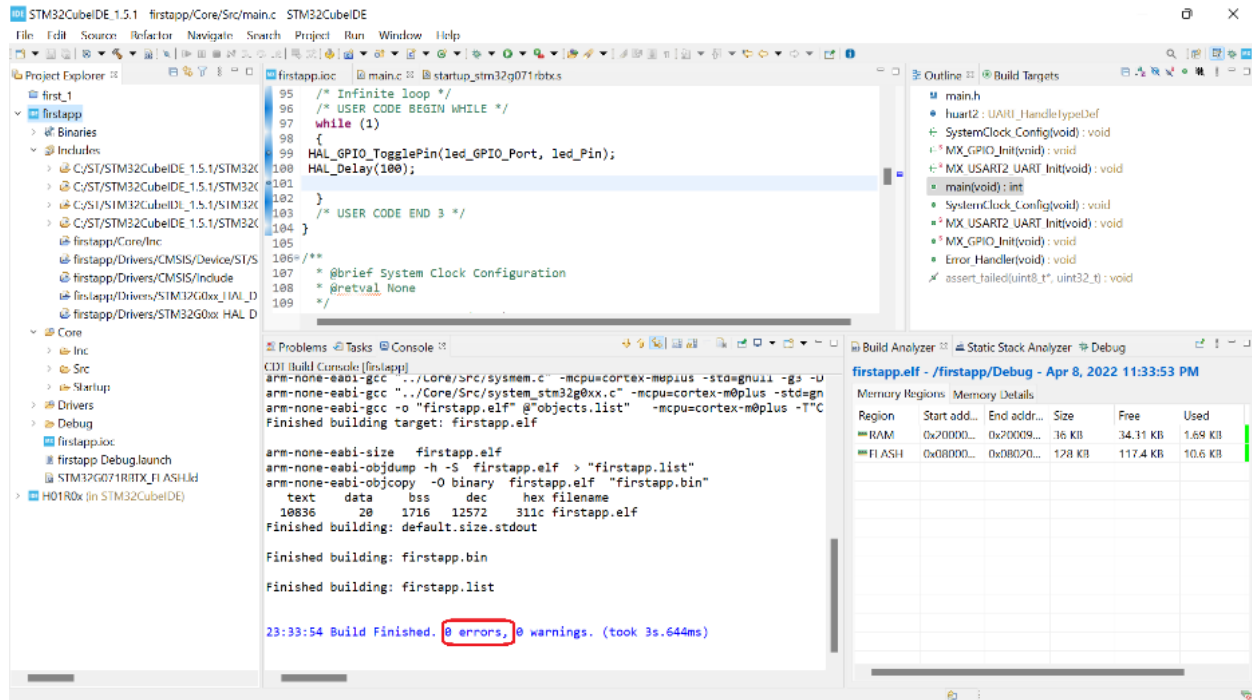



```
#include "main.h"
void SystemClock_Config(void);
static void MX_GPIO_Init(void);
int main(void)
{
    HAL_Init();
    SystemClock_Config();
    MX_GPIO_Init();
    while (1)
    {
        if(HAL_GPIO_ReadPin(GPIOB, GPIO_PIN_11)==0)
        {
            HAL_GPIO_WritePin(GPIOA, GPIO_PIN_0, 1);
        }
        else
            HAL_GPIO_WritePin(GPIOA, GPIO_PIN_0, 0);
    }
}
```

الخطوة السادسة: قم بالنقر بزر الفأرة الأيمن على اسم المشروع ثم اختر Clean project ثم Build project لتتم عملية ترجمة الكود للصيغة الثنائية



نتأكد من خلو الكود من الأخطاء اللغوية Syntax errors



الخطوة السابعة: قم باستخدام STM32CubeProgrammer برفع الكود للمتحكم الموجود على البورد.

8- للتحكم بالمداخل الرقمية للتحكم STM32 دون استخدام مكتبة HAL:

هناك مجموعة من الم سجلات ت تستخدم للتحكم بالمداخل الرقمية لمتحكم STM32 سنكتفي فقط بذكر الم سجل الم سؤال عن قراءة حالة المنفذ أو أحد الأقطاب الموجودة فيه (ولمزيد من المعلومات عن باقي المسجلات بإمكانك الرجوع إلى ال datasheet الخاصة بالمتحكم)، ويدعى Input data register (IDR) وله الشكل التالي:

7.4.5 GPIO port input data register (GPIOx_IDR) (x = A..H)

Address offset: 0x10

Reset value: 0x0000 XXXX (where X means undefined)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IDR15	IDR14	IDR13	IDR12	IDR11	IDR10	IDR9	IDR8	IDR7	IDR6	IDR5	IDR4	IDR3	IDR2	IDR1	IDR0
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

Bits 31:16 Reserved, must be kept at reset value.

Bits 15:0 IDRy: Port input data (y = 0..15)

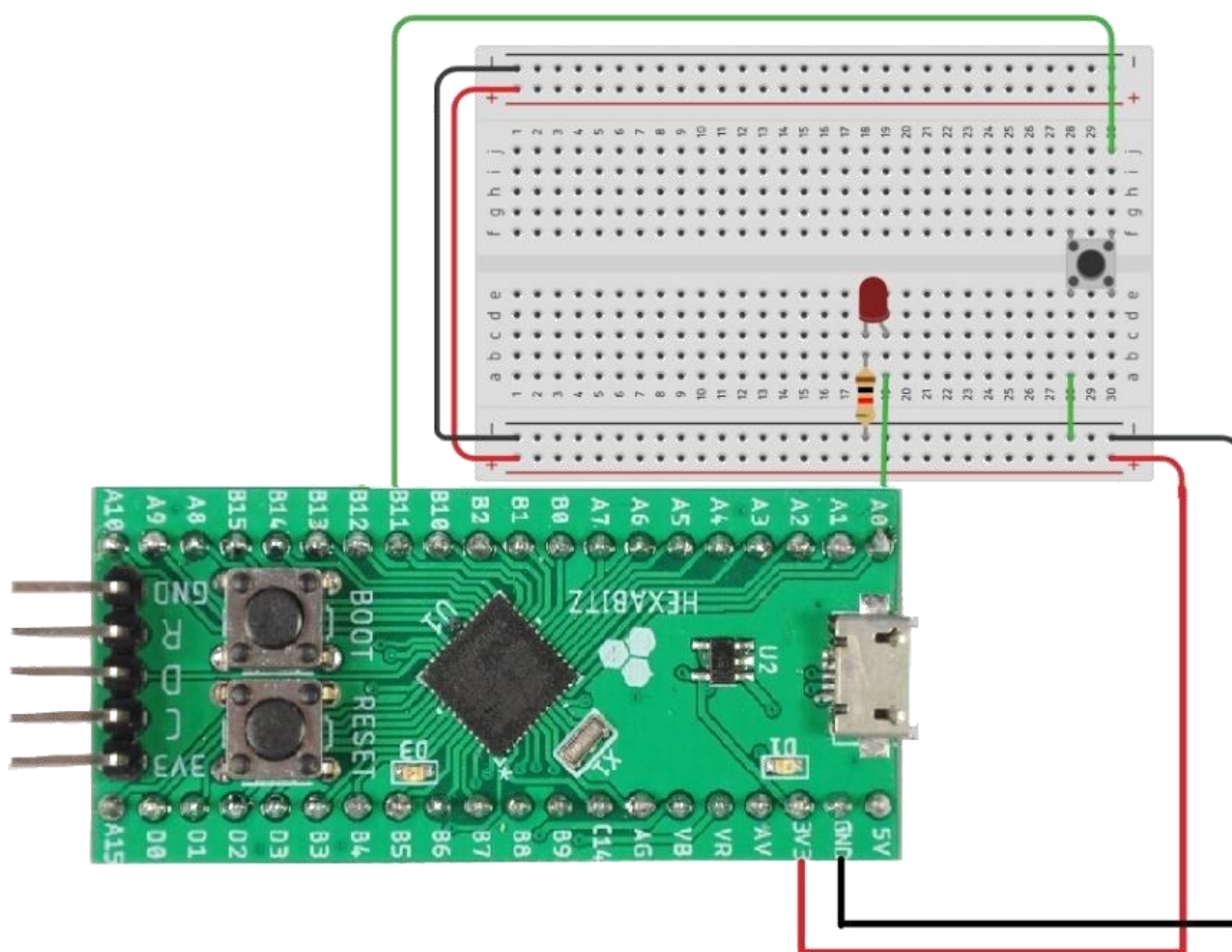
These bits are read-only and can be accessed in word mode only. They contain the input value of the corresponding I/O port.

والمسجل IDR هو مسجل للقراءة فقط ، البتات 16:31 غير مستخدمين، أما البتات 0:15 فيعبر كل بت عن حالة القطب المقابل له، ففي حال تفعيل مقاومة الرفع الداخلية للقطب عندها سيكون القطب في حالة HIGH بشكل دائم، وعند ضغط المفتاح الموصول معه سيصبح القطب في حالة LOW، لذا يجب مراقبة حالة القطب بشكل مستمر لحين يصبح القطب في حالة LOW عندها يكون المفتاح الموصول معه مضغوط.

فلنحس حالة القطب الأول من المنفذ A نستخدم جملة الشرط التالية:

```
if ( (GPIOx->IDR & GPIO_PIN_0) == 0 )
```

9- إعادة التطبيق الثاني دون استخدام مكتبة HAL



ويكون الشكل النهائي للكود:

```
#include "main.h"
```

```

void SystemClock_Config(void);
static void MX_GPIO_Init(void);
int main(void)
{
    HAL_Init();
    SystemClock_Config();
    MX_GPIO_Init();
    while (1)
    {
        if ( (GPIOB->IDR & GPIO_PIN_11) == 0 )

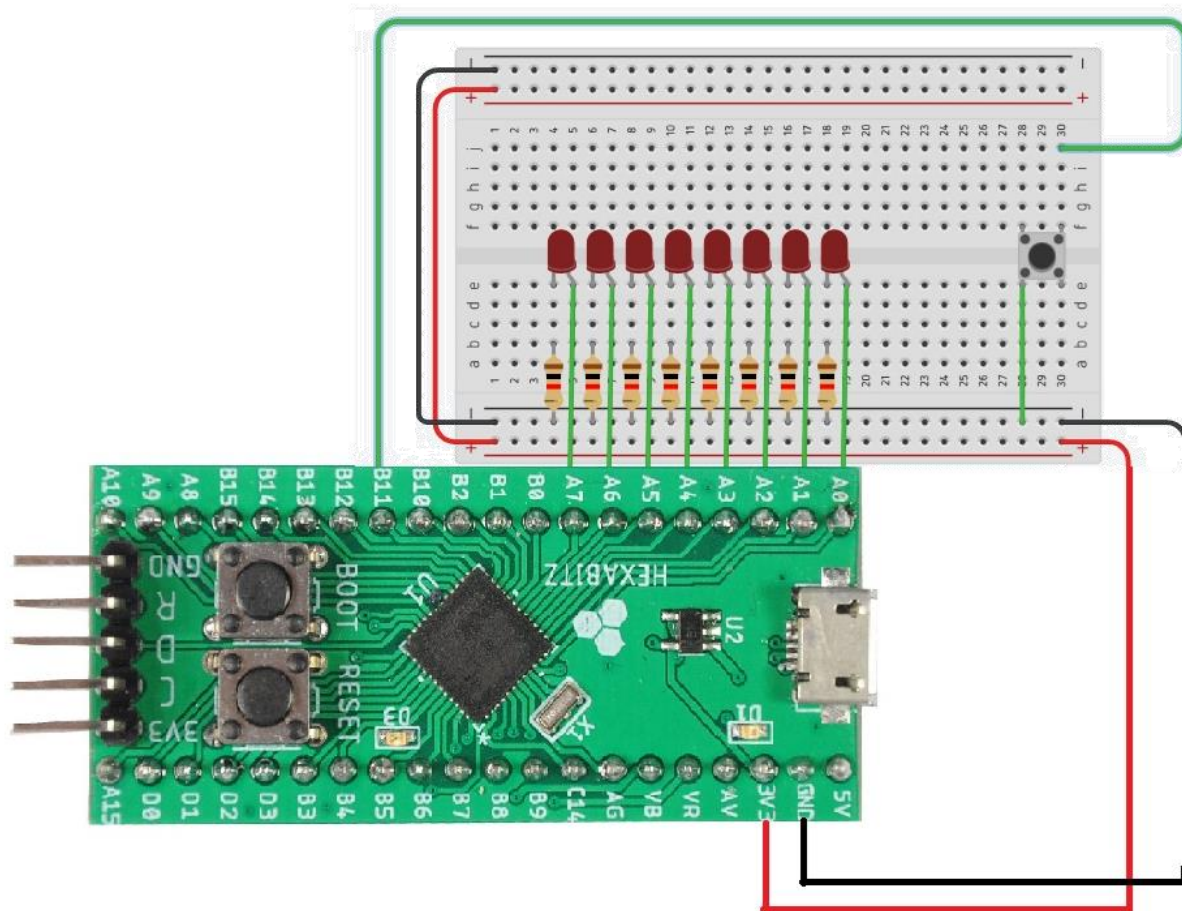
        {
            GPIOA->ODR = 0x0001;
        }

        else
            GPIOA->ODR = 0;

    }
}

```

10- التطبيق الثالث: إضاءة مجموعة من اللمبات من خلال زر start
 رابط فيديو التطبيق على اليوتيوب: <https://youtu.be/y-imBsRSXQA>



أعد الخطوات السابقة مع ضبط إعدادات الأقطاب بالشكل التالي:

s2_ex5.ioc - Pinout & Configuration

Pinout & Configuration | Clock Configuration | Project Manager | Tools

Software Packs | Pinout

GPIO Mode and Configuration

Configuration

Group By Peripherals

GPIO

Search Signals

Show only Modified Pins

Pin	Signal o	GPIO o	GPIO m	GPIO P	Maximu	Fast Mo	User La	Modified
PA1	n/a	Low	Output ...	No pull...	Low	n/a		<input type="checkbox"/>
PA2	n/a	Low	Output ...	No pull...	Low	n/a		<input type="checkbox"/>
PA3	n/a	Low	Output ...	No pull...	Low	n/a		<input type="checkbox"/>
PA4	n/a	Low	Output ...	No pull...	Low	n/a		<input type="checkbox"/>
PA5	n/a	Low	Output ...	No pull...	Low	n/a		<input type="checkbox"/>
PA6	n/a	Low	Output ...	No pull...	Low	n/a		<input type="checkbox"/>
PA7	n/a	Low	Output ...	No pull...	Low	n/a		<input type="checkbox"/>
PB11	n/a	n/a	Input mo...	Pull-up	n/a	n/a		<input checked="" type="checkbox"/>

GPIO mode: Input mode

GPIO Pull-up/Pull-down: Pull-up

User Label:

Pinout view | System view

STM32G0B1CEUxN

UFQFPN48

GPIO_Output

GPIO_Output

GPIO_Output

GPIO_Output

GPIO_Output

GPIO_Output

GPIO_Input

ويكون الشكل النهائي للكود :

```
#include "main.h"
void SystemClock_Config(void);
static void MX_GPIO_Init(void);
int main(void)
{
    HAL_Init();
    SystemClock_Config();
    MX_GPIO_Init();
    while (1)
    {
        if(HAL_GPIO_ReadPin(GPIOB, GPIO_PIN_11)==0)
        {
            HAL_GPIO_WritePin(GPIOA, GPIO_PIN_0, 1);
            HAL_Delay(250);
            //*****
            HAL_GPIO_WritePin(GPIOA, GPIO_PIN_1, 1);
            HAL_Delay(250);
            //*****
            HAL_GPIO_WritePin(GPIOA, GPIO_PIN_2, 1);
            HAL_Delay(250);
            //*****
            HAL_GPIO_WritePin(GPIOA, GPIO_PIN_3, 1);
            HAL_Delay(250);
            //*****
            HAL_GPIO_WritePin(GPIOA, GPIO_PIN_4, 1);
            HAL_Delay(250);
            //*****
            HAL_GPIO_WritePin(GPIOA, GPIO_PIN_5, 1);
            HAL_Delay(250);
            //*****
            HAL_GPIO_WritePin(GPIOA, GPIO_PIN_6, 1);
            HAL_Delay(250);
            //*****
            HAL_GPIO_WritePin(GPIOA, GPIO_PIN_7, 1);
            HAL_Delay(250);
        }
        else
```



```

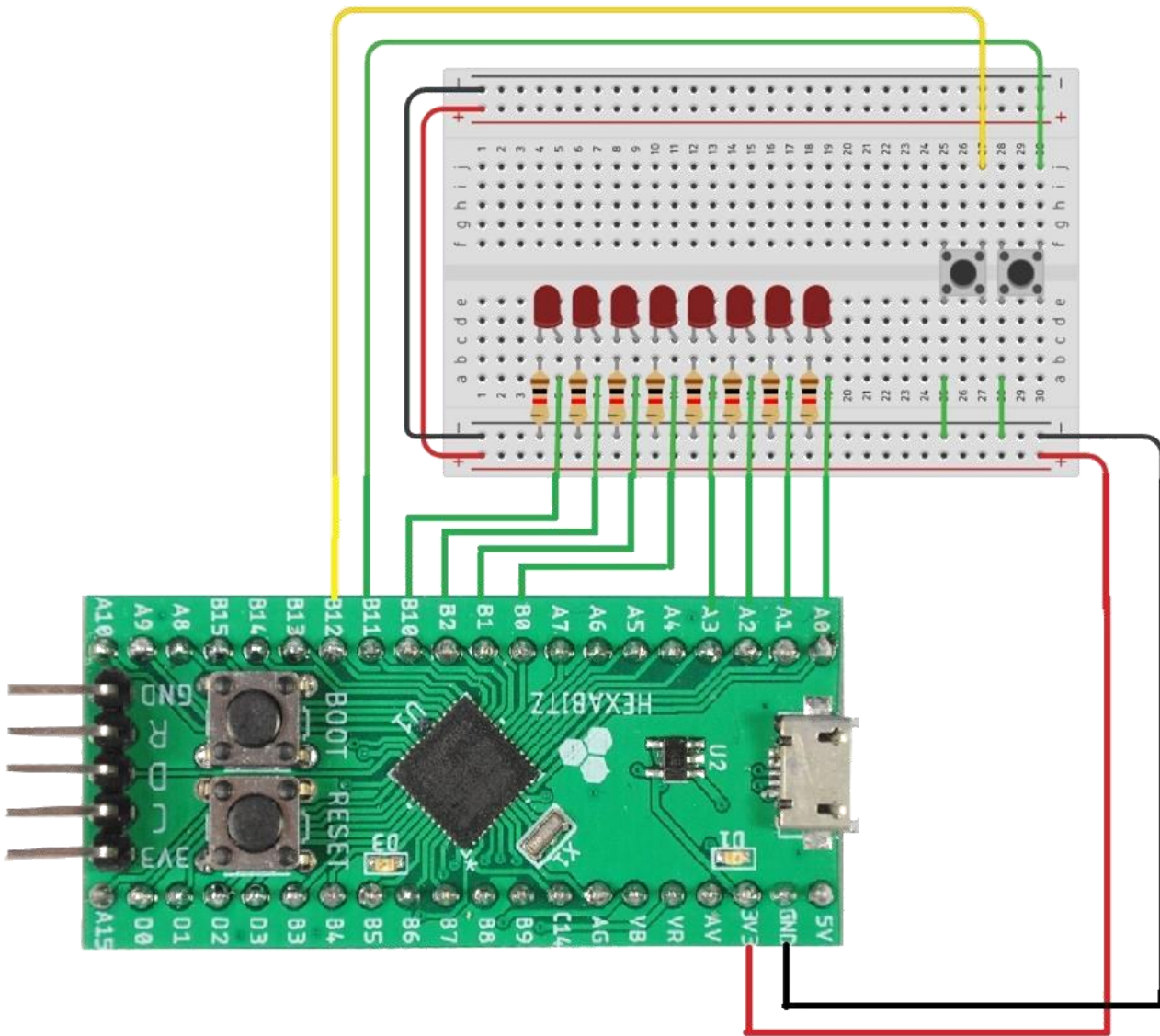
{
    GPIOA->ODR = 0;
}

}

```

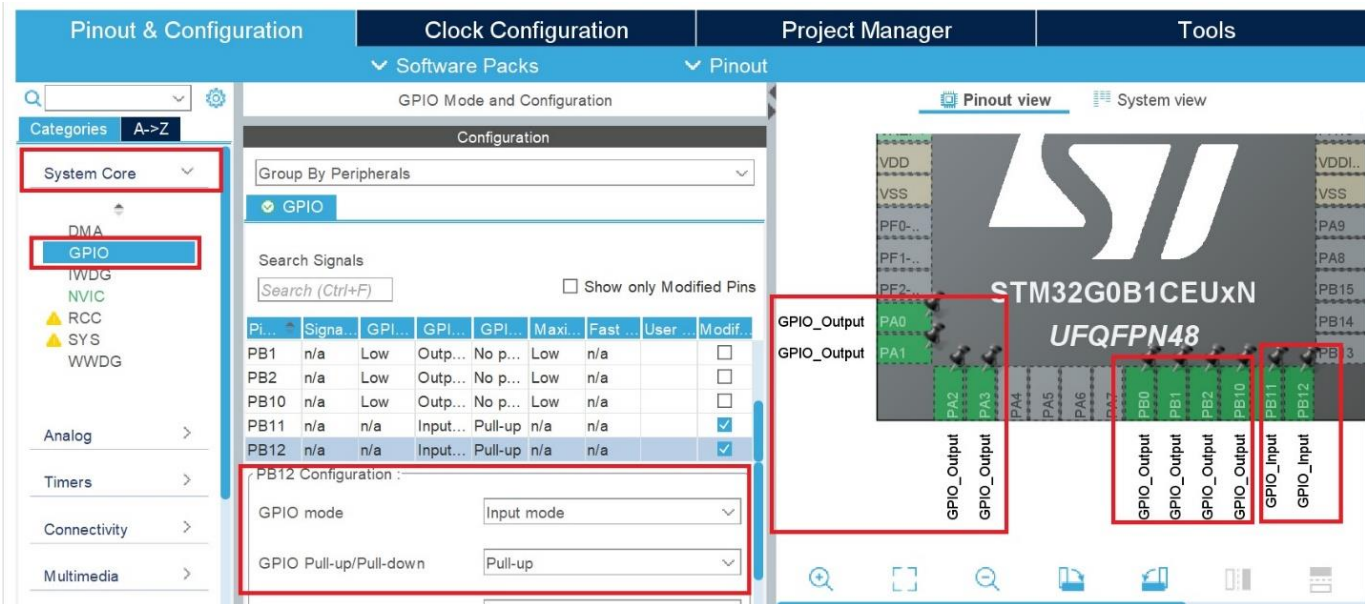
11- التطبيق الرابع: تشغيل مجموعتين من اللىدات بتسلسل معين عند الضغط على أزرار التشغيل

رابط فيديو التطبيق على اليوتيوب: <https://youtu.be/cRsP8Pt6ME8>



- عند الضغط على الزر الموصول مع الـ B11 يتم تشغيل اللدات الموصولة مع المنفذ A

- عند الضغط على الزر الموصول مع الـ B12 يتم تشغيل اللدات الموصولة مع المنفذ B
- أعد الخطوات السابقة مع ضبط إعدادات الأقطاب بالشكل التالي:



ويكون الشكل النهائي للبرنامج:

```
#include "main.h"
void SystemClock_Config(void);
static void MX_GPIO_Init(void);
int main(void)
{
    HAL_Init();
    SystemClock_Config();
    MX_GPIO_Init();
    while (1)
    {
        if(HAL_GPIO_ReadPin(GPIOB, GPIO_PIN_11)==0)
        {
            HAL_GPIO_WritePin(GPIOA, GPIO_PIN_0, 1);
            HAL_Delay(250);
            //*****
            HAL_GPIO_WritePin(GPIOA, GPIO_PIN_1, 1);
            HAL_Delay(250);
            //*****
            HAL_GPIO_WritePin(GPIOA, GPIO_PIN_2, 1);
            HAL_Delay(250);
        }
    }
}
```



```

    //*****
    HAL_GPIO_WritePin(GPIOA, GPIO_PIN_3, 1);
    HAL_Delay(250);
    //*****

}

else
{
    HAL_GPIO_WritePin(GPIOA, GPIO_PIN_0, 0);
    HAL_GPIO_WritePin(GPIOA, GPIO_PIN_1, 0);
    HAL_GPIO_WritePin(GPIOA, GPIO_PIN_2, 0);
    HAL_GPIO_WritePin(GPIOA, GPIO_PIN_3, 0);

}
//-----
if(HAL_GPIO_ReadPin(GPIOB, GPIO_PIN_12)==0)
{
    HAL_GPIO_WritePin(GPIOB, GPIO_PIN_0, 1);
    HAL_Delay(250);
    //*****
    HAL_GPIO_WritePin(GPIOB, GPIO_PIN_1, 1);
    HAL_Delay(250);
    //*****
    HAL_GPIO_WritePin(GPIOB, GPIO_PIN_2, 1);
    HAL_Delay(250);
    HAL_GPIO_WritePin(GPIOB, GPIO_PIN_10, 1);
    HAL_Delay(250);

}

else
{
    HAL_GPIO_WritePin(GPIOB, GPIO_PIN_0, 0);
    HAL_GPIO_WritePin(GPIOB, GPIO_PIN_1, 0);
    HAL_GPIO_WritePin(GPIOB, GPIO_PIN_2, 0);
    HAL_GPIO_WritePin(GPIOB, GPIO_PIN_10, 0);
}

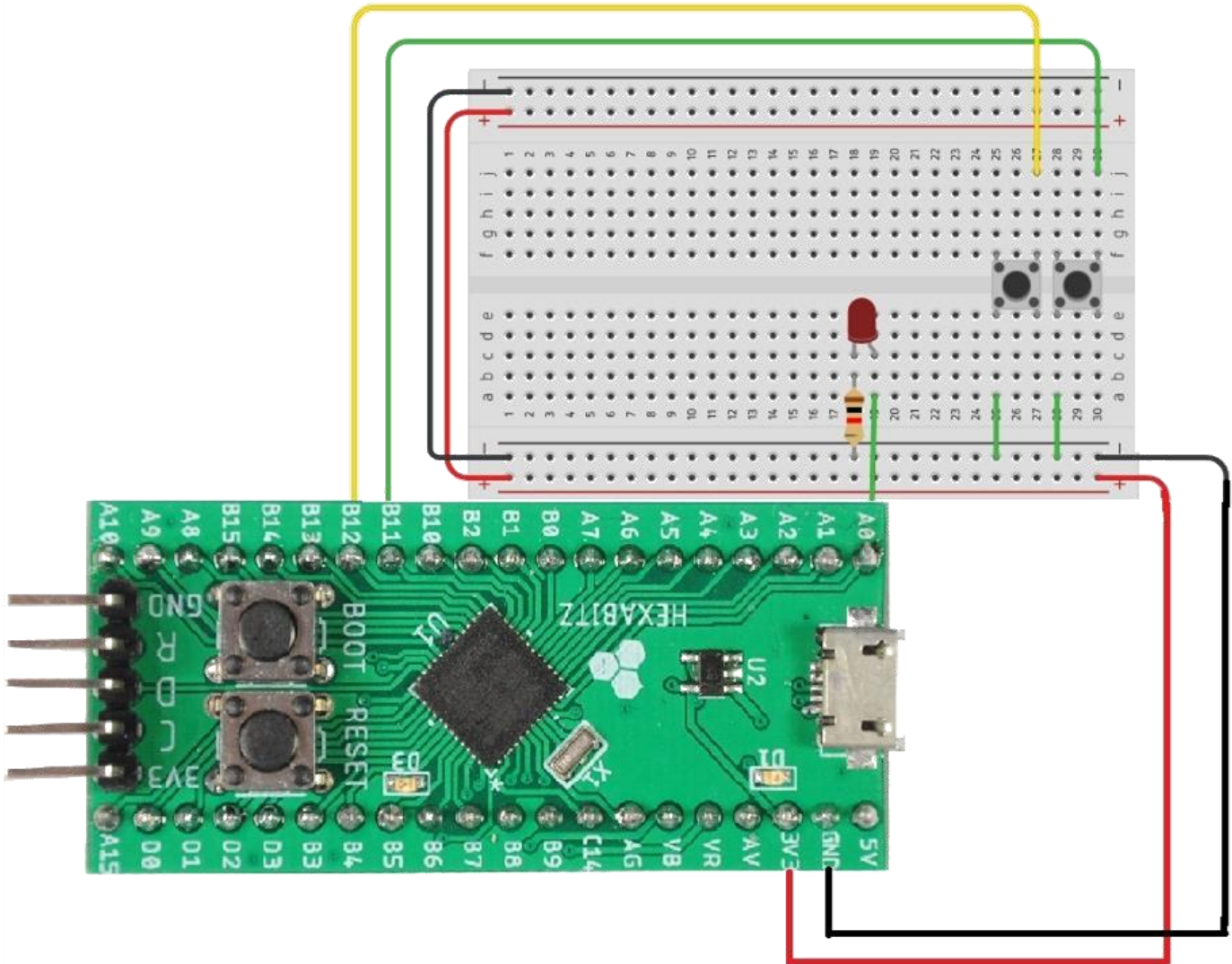
```

}}

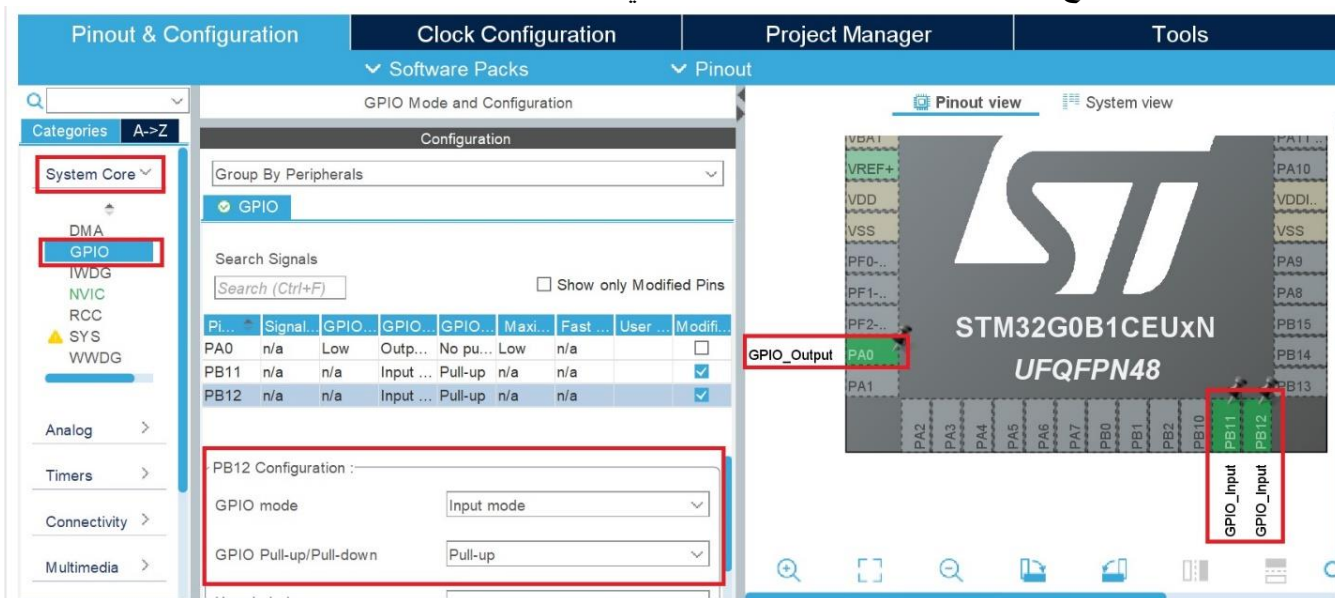
12- التطبيق الخامس: تشغيل الليد الموصل مع الـ A0 في حال كان الـ Switch و الـ Button

(الموصلين على الأقطاب PB11 و PB12 مضغوطين)

رابط فيديو التطبيق على اليوتيوب: <https://youtu.be/1kcyzlk63kw>



أعد الخطوات السابقة مع ضبط إعدادات الأقطاب بالشكل التالي:



ويكون الشكل النهائي للكود بالشكل التالي:

```
#include "main.h"
void SystemClock_Config(void);
static void MX_GPIO_Init(void);
int main(void)
{
    HAL_Init();
    SystemClock_Config();
    MX_GPIO_Init();
    while (1)
    {
        if((HAL_GPIO_ReadPin(GPIOB, GPIO_PIN_11)==0) &&
        (HAL_GPIO_ReadPin(GPIOB, GPIO_PIN_12)==0))
        {
            HAL_GPIO_WritePin(GPIOA, GPIO_PIN_0, 1);
        }
        else
            HAL_GPIO_WritePin(GPIOA, GPIO_PIN_0, 0);
    }
}
```

أيضاً بإمكانك محاكاة الأمثلة من خلال برنامج Proteus وبلاستعانة بالفيديوهات التالية:

- https://youtu.be/_mrJoJLcQzU -
- <https://youtu.be/X41r4gzhwxA> -
- <https://youtu.be/7eTOqtKbAHs> -
- <https://youtu.be/slekCuxG7oY> -
- <https://youtu.be/0X57y6m-3Og> -
- <https://youtu.be/XSg2jF3kY68> -