

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

مقرر المتحكمات المصغرة الجلسة الثالثة

السنة الرابعة ميكاترونيك

2023/2022

الغاية من الجلسة

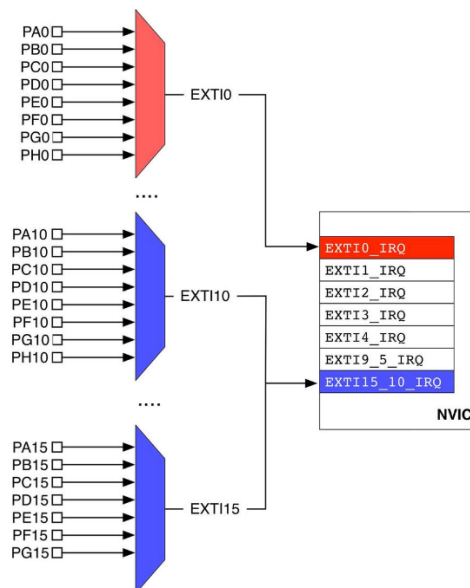
- 1- التعرف على المقاطعات الخارجية في متحكمات STM32 بحيث يكون قادراً على تهيئة هذه المقاطعات في المتحكم لاستثمارها حسب الحاجة.
- 2- التطبيق 1: بناء لنظام يحتوي على زر طوارئ لتفعيل برنامج للطوارئ حيث يتم تنفيذه بغض النظر عن عمل النظام باستخدام البورد التطويري.
- 3- التعرف إلى شاشة 7-seg
- 4- التطبيق 2: بناء لنظام يحتوي على شاشة 7-seg لتعمل كعداد من 0 حتى 9 وباستخدام البورد التطويري.
- 5- التطبيق 3: تشغيل وإطفاء جهاز من خلال زر لحظي واحد باستخدام المقاطعة إظهار حالة الجهاز على شاشة الإظهار باستخدام البورد التطويري.
- 6- التطبيق 4: عداد زوار وإظهار العدد على شاشة 7-seg بخانتين ,حيث ان الكباسين موصولين على اقطاب المقاطعة الخارجية (INT0,INT1) ويتم تفعيل المقاطعتين عند الجبهة الهابطة باستخدام البورد التطويري.

1- تعريف المقاطعة:

هي الاستجابة لحدث ما أثناء تنفيذ البرنامج الرئيسي void main وذلك عن طريق تنفيذ برنامج فرعي يسمى برنامج خدمة المقاطعة. وغالباً ما نصادف المقاطعات في الحياة اليومية وكمثال عليها يمثل رنين الهاتف مقاطعة تعبر عن وجود مكالمات هاتفية تنتظر الرد عليها ويمثل برنامج خدمة المقاطعة في المتحكم الرد على هذه المكالمات.

المقاطعات الخارجية INTx:

يحتوي متحكم STM32 على 16 خط للمقاطعات الخارجية هي من الخط 0 حتى الخط 15 حيث تشير أرقام الخطوط إلى أرقام أطراف الـ GPIOs يتم تجميع المقاطعات الخارجية ضمن خطوط متصلة مع أطراف المتحكم المصغر GPIO، وبما أن للمعالج عدة أطراف GPIOs لذا فإن كل خط من خطوط المقاطعات الخارجية EXTI هو عبارة عن خط مشترك بين عدة أطراف Pins.



في كل خط من خطوط المقاطعات الخارجية (كل مجموعة) يحق لـ pin واحد أن يقوم بتوليد المقاطعة وعلى البرنامج أن يكتشف أي pin قام بتوليد المقاطعة حيث يمكن قذح المقاطعة عند الجبهة الـ صاعدة rising edge أو الهابطة falling edge أو عند كليهما ، هذا يعني أن الطرف PA0 متصل بالخط LINE0 والطرف PA13 متصل بالطرف LINE13 ، أيضاً PB0 و PC0 متصلين بالخط LINE0 ، بمعنى أن جميع الأطراف ذات الأرقام Px0 متصلة بالخط LINE0 (حيث يعبر x عن اسم المنفذ) ، أيضاً جميع الأطراف ذات الأرقام Px3 متصلة بالخط LINE3 وهكذا...

GPIO_MODE_IT_RISING



GPIO_MODE_IT_RISING_FALLING



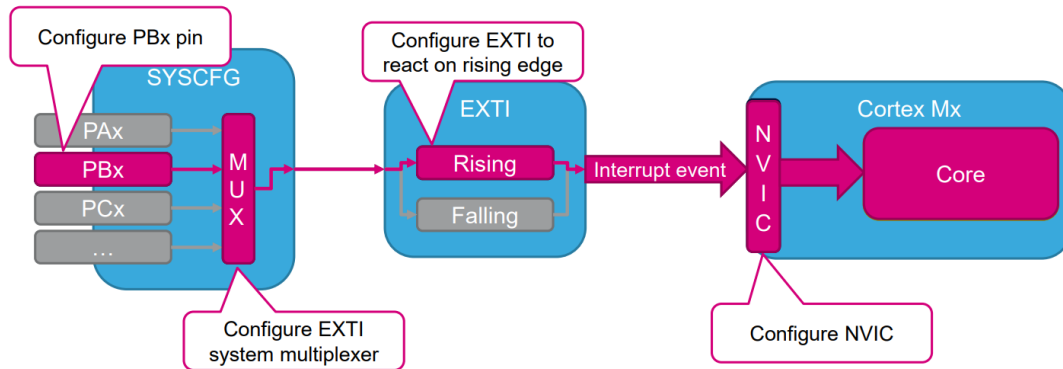
GPIO_MODE_EVT_FALLING



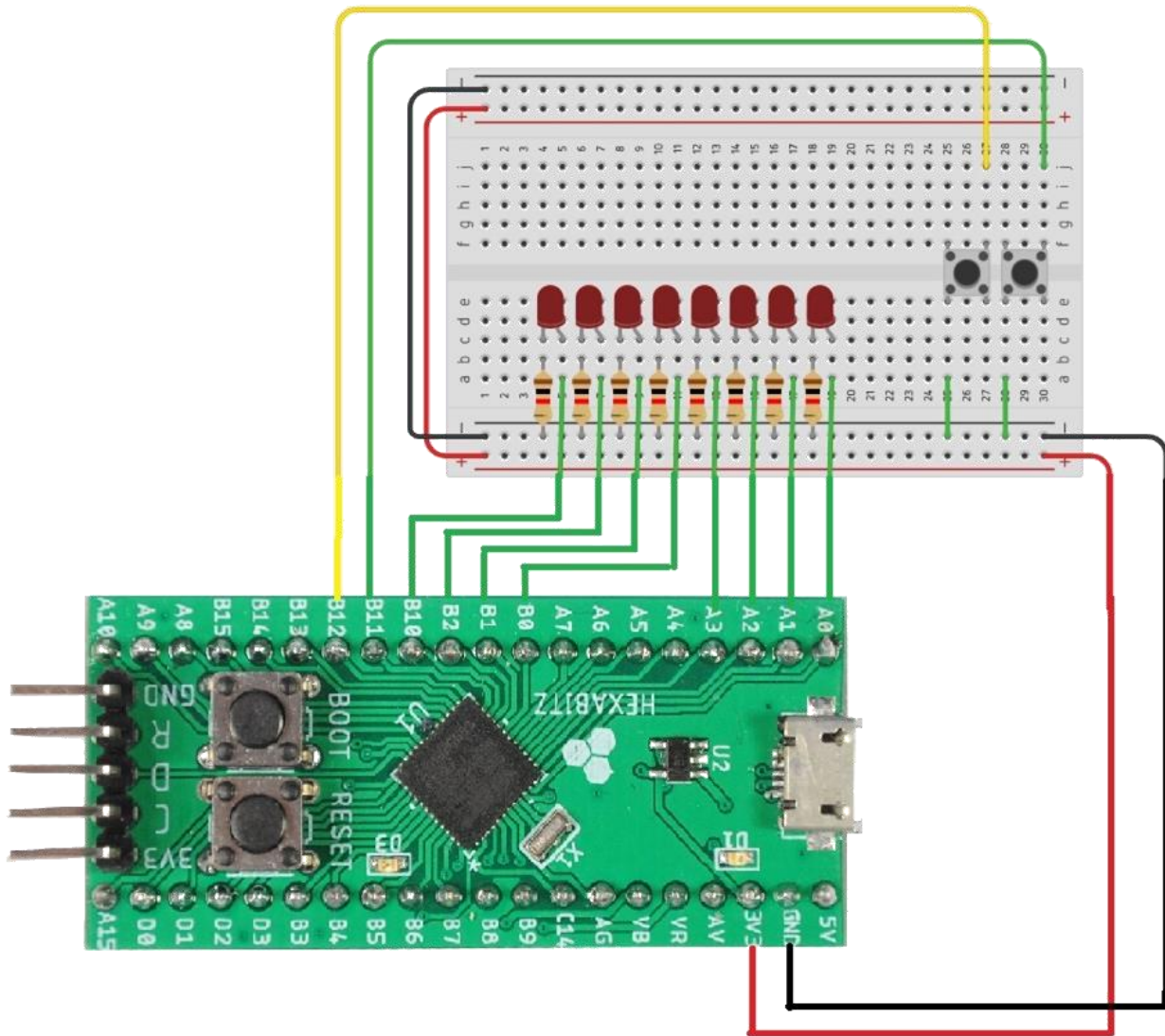
لكن يجب الانتباه للملاحظات التالية:

1. الأطراف PA0، PB0، PC0... جميعها متصلة بالخط LINE0 لذا في اللحظة الواحدة بإمكانك توليد مقاطعة على طرف واحد فقط من هذه الأطراف.
2. الأطراف PA0، PA5 متصلين على خطين مختلفين من خطوط المقاطعة لذا يمكنك استخدامها في نفس اللحظة لتوليد المقاطعة.

في حال حدوث المقاطعة فإن المعالج يتوقف عن تنفيذ الكود الحالي ويقوم بمعالجة المقاطعة من خلال تنفيذ برنامج خدمة المقاطعة (ISR) (Interrupt Service Routines) والذي يتم تحديد عنوانه في الذاكرة من خلال جدول أشعة المقاطعة المعروف مسبقاً (Vector Interrupt Table (VIC)).



- 2- **التطبيق العملي 1:** زر طوارئ لتفعيل برنامج للطوارئ حيث يتم تنفيذه بغض النظر عن عمل النظام (إنارة اللدات) و بعد الانتهاء من برنامج خدمة المقاطعة (الطوارئ) يعود للبرنامج الرئيسي ليتابع عمله وذلك باستخدام البورد التطويري:

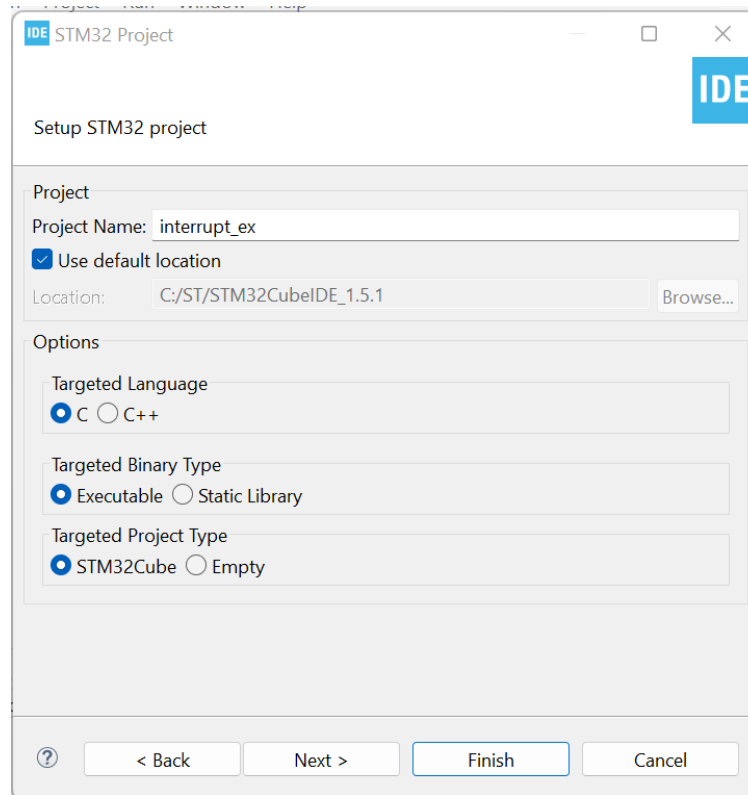


1. ضبط إعدادات المشروع:

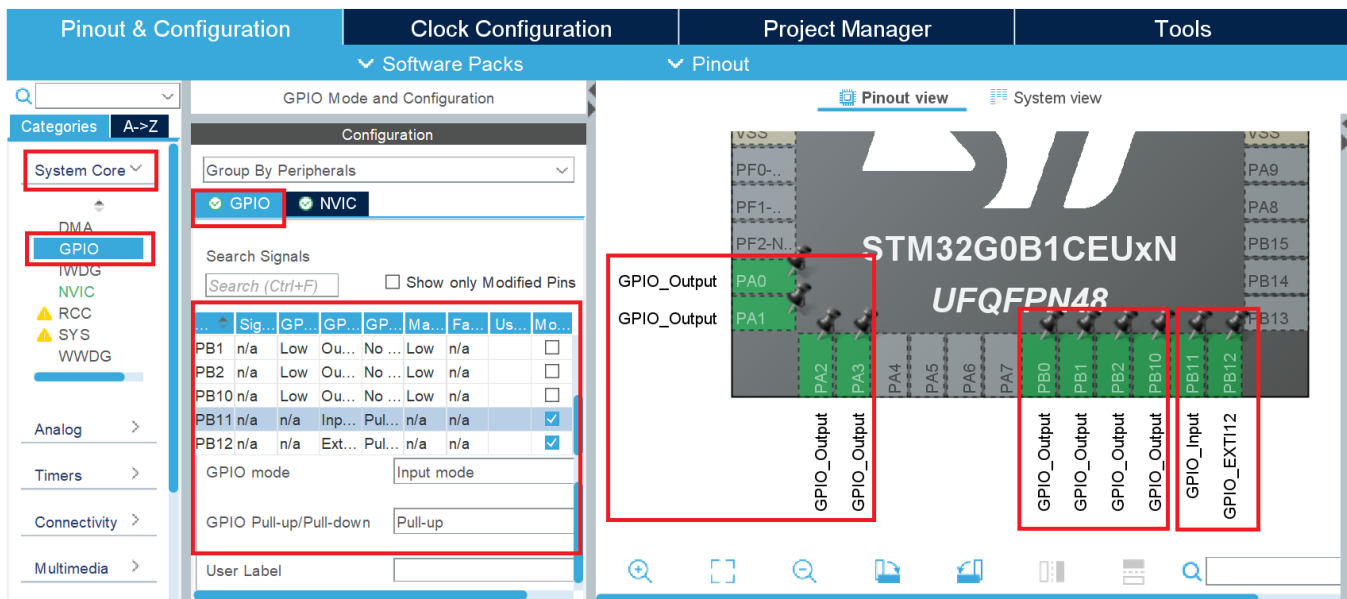
الخطوة الأولى: فتح بيئة STM32CubeIDE وإن شاء م شروع جديد ثم قم باختيار المتحكم

STM32G0B1CEU6N

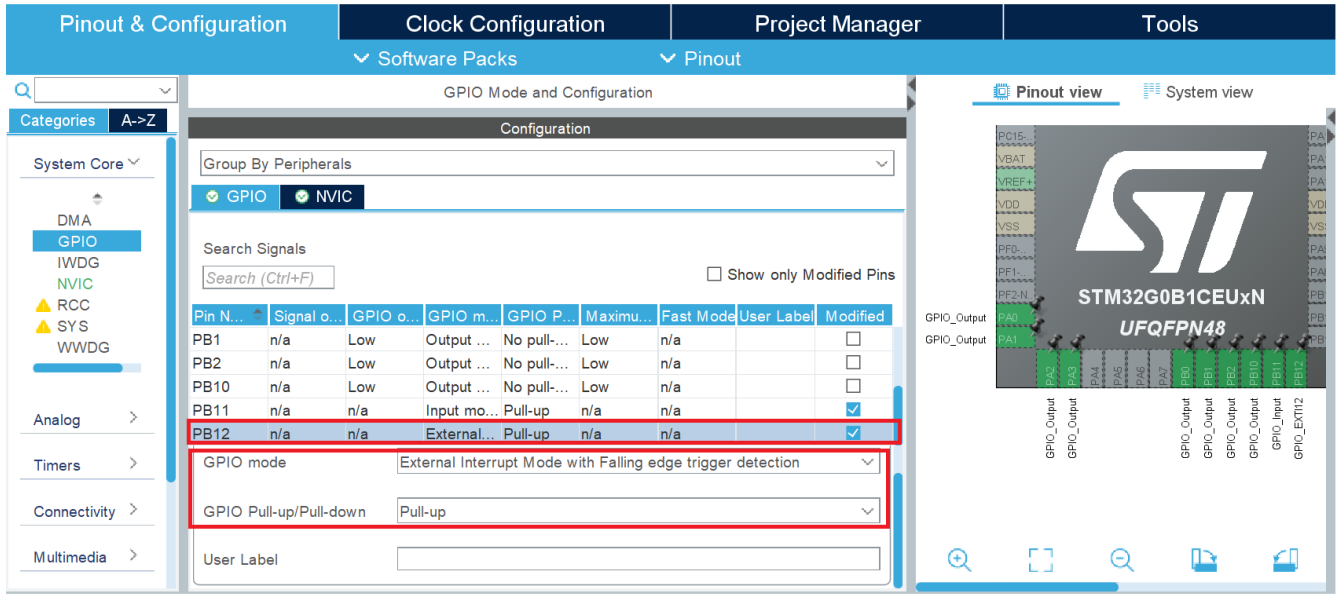
الخطوة الثانية: اختيار اسم للمشروع



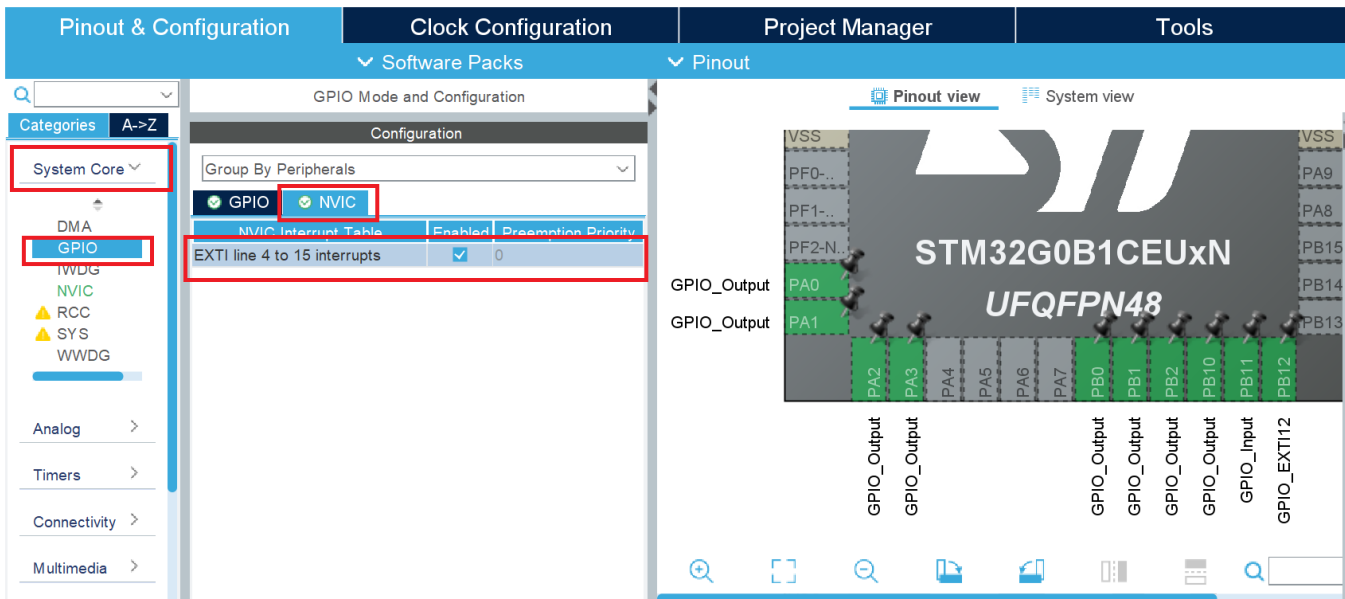
الخطوة الثالثة: اختر الأقطاب PA0:PA3، والأقطاب PB0:PB1: PB2: PB10، وضبطها كأقطاب خرج ، والقطب PB11 كقطب دخل مع تفعيل مقاومة الرفع الداخلية، والقطب PB12 كقطب مقاطعة خارجية:



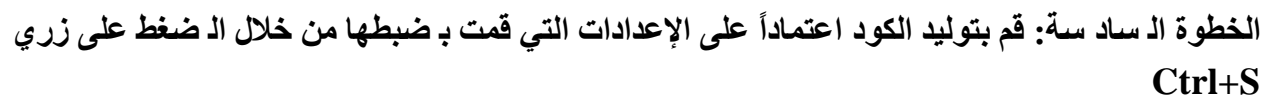
بما أننا قمنا بتفعيل مقاومة الرفع الداخلية لقطب المقاطعة لذا سنقوم بتفعيل المقاطعة عند الجبهة الهابطة.



الخطوة الرابعة: قم بفتح NVIC Tab ثم قم بتفعيل المقاطعات الخارجية:



الخطوة الخامسة: قم بضبط ساعة النظام على الساعة الداخلية (HSI) واختر التردد 16MHZ

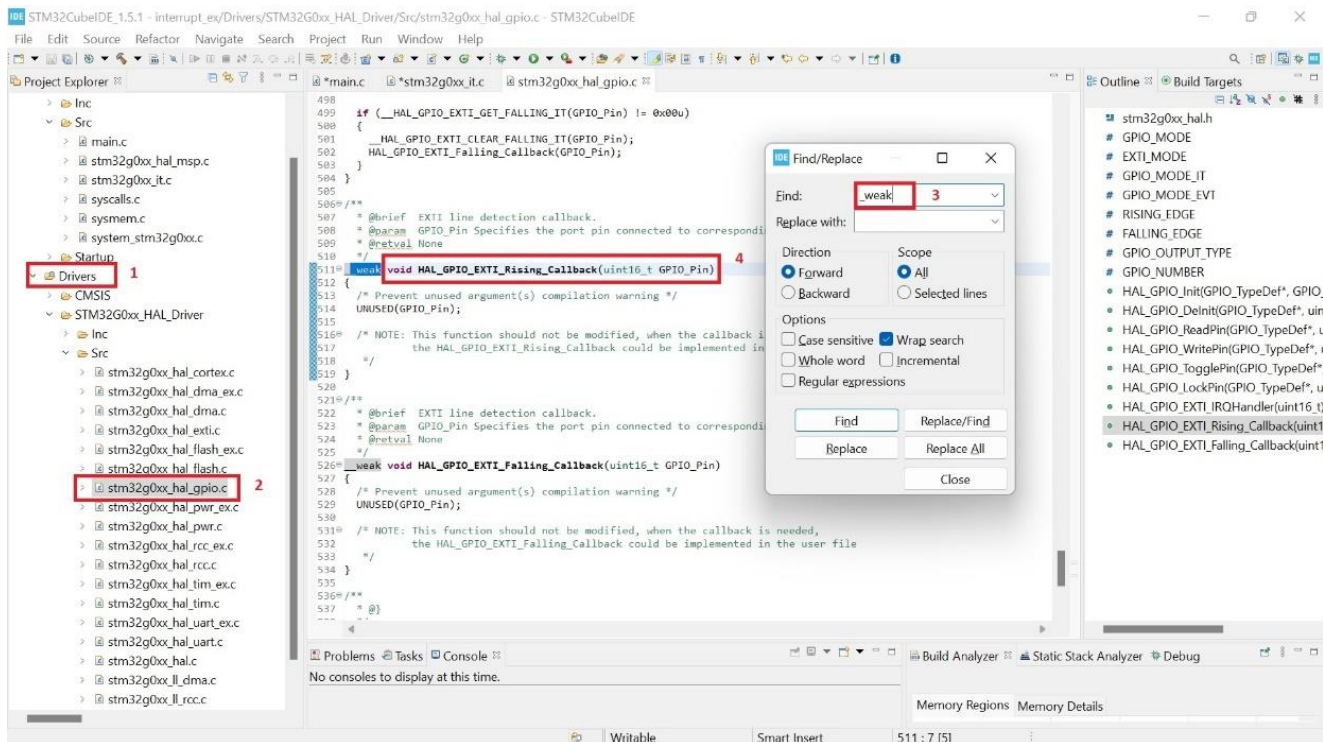


نلاحظ أن ملف الـ `main.c` يحتوي على تهيئة وتضمين مكتبة HAL وضبط ساعة النظام بالإضافة إلى تهيئة المداخل والمخارج والطرفيات كما في الشكل التالي:



الخطوة السابعة :

- 1- نختار المجلد Drivers
- 2- نقوم بفتح الملف stm32g0xx_hal_gpios.c
- 3- نبحث عن الـ function الذي يبدأ بكلمة `_weak` ونحدد اسم الـ function وننسخه



تعني كلمة `_weak` أي سيتم استدعاؤه في حال لم يكتب المستخدم برنامج خدمة مقاطعة يحمل نفس الاسم.

4- نقوم بلصق الـ function ضمن ملف الـ `main.c` إما قبل أو بعد الـ `int main()` مع إزالة كلمة `_weak` ودفع فيه الأوامر التي نريد تنفيذها عند طلب المقاطعة مثلاً هنا قمنا بفحص القطب الذي حدثت عنده المقاطعة ثم قمنا باستدعاء تابع HAL الذي يقوم بعكس الحالة المنطقية لليد عند حدوث المقاطعة.

الكود بالكامل:

```
#include "main.h"
void SystemClock_Config(void);
static void MX_GPIO_Init(void);

void HAL_GPIO_EXTI_Falling_Callback(uint16_t GPIO_Pin)
{
    if(GPIO_Pin == GPIO_PIN_12)
```

```

    {
        HAL_GPIO_TogglePin(GPIOB, GPIO_PIN_0);
        HAL_GPIO_TogglePin(GPIOB, GPIO_PIN_1);
        HAL_GPIO_TogglePin(GPIOB, GPIO_PIN_2);
        HAL_GPIO_TogglePin(GPIOB, GPIO_PIN_10);
    }
}

int main(void)
{
    HAL_Init();
    SystemClock_Config();
    MX_GPIO_Init();
    while (1)
    {
        if (HAL_GPIO_ReadPin(GPIOB, GPIO_PIN_11) == 0)
        {
            GPIOA->ODR = 0X0001;
            HAL_Delay(50);
            //*****
            GPIOA->ODR = 0X0002;
            HAL_Delay(50);
            //*****
            GPIOA->ODR = 0X0004;
            HAL_Delay(50);
            //*****
            GPIOA->ODR = 0X0008;
            HAL_Delay(50);
            //*****
            GPIOA->ODR = 0X0010;
            HAL_Delay(50);
            //*****
            GPIOA->ODR = 0X0020;
            HAL_Delay(50);
            //*****
            GPIOA->ODR = 0X0040;
            HAL_Delay(50);
            //*****
            GPIOA->ODR = 0X0080;

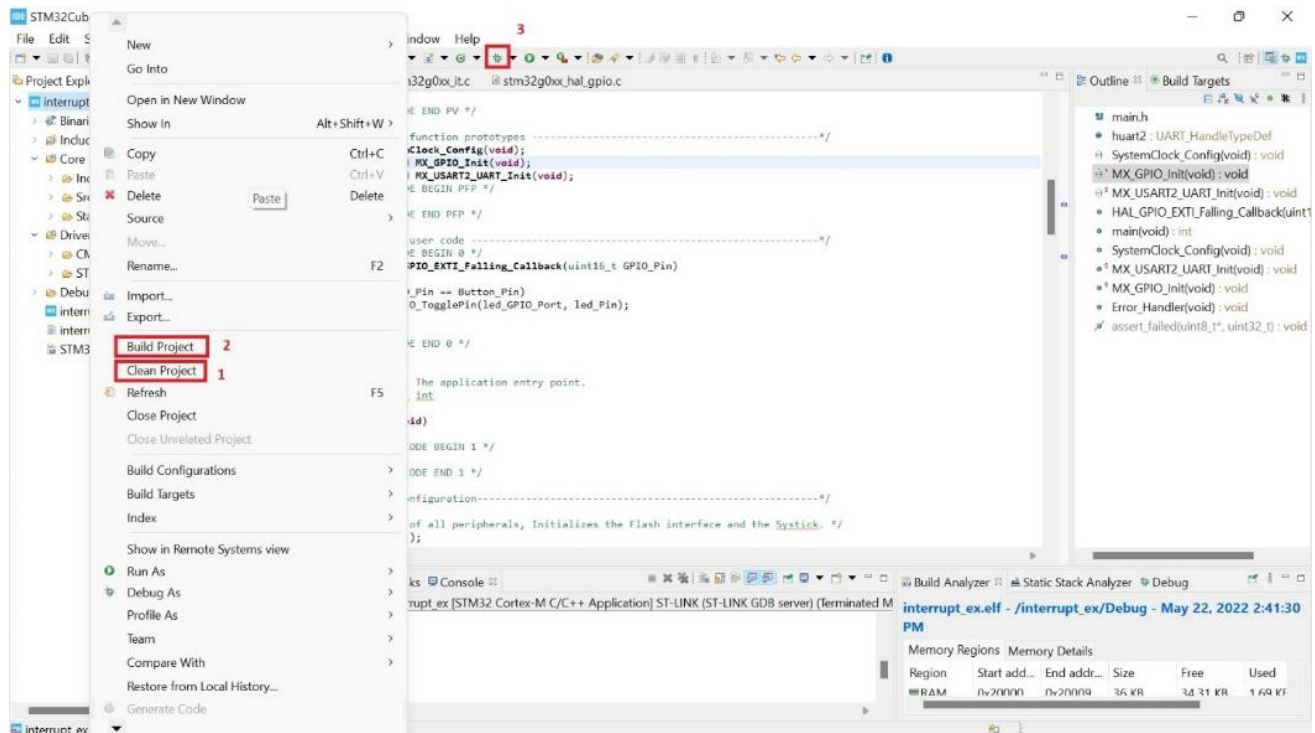
```

```

        HAL_Delay(100);
    }
    else
    {
        GPIOA->ODR = 0X0000;
    }
}
}

```

الخطوة الثامنة: التأكد من خلو الكود من أي أخطاء برمجية من خلال تنظيف الكود clean project ثم ترجمة الكود من خلال Build Project من أجل توليد ملف .hex.



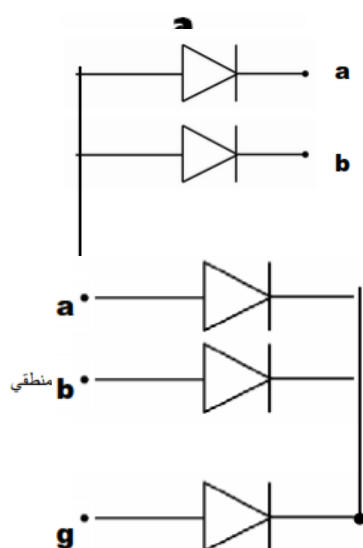
الخطوة التاسعة: قم باستخدام STM32CubeProgrammer برفع الكود للمتحكم الموجود على البورد.

املاحظة:

لزيادة سرعة الاستجابة للمقاطعات عليك الاستغناء عن مكتبة HAL واستخدام المسجلات بشكل مباشر. أيضاً لا يمكن استخدام التأخير الزمني من مكتبة HAL ضمن برنامج خدمة المقاطعة.

3- التعرف إلى شاشة 7-seg :

تستخدم شاشات الإظهار الرقمية ذات الـ سبع قطع 7-Segment Led لإظهار الأعداد ومثال عليها استخدامهما لإظهار عدد الزوار في معرض أو إظهار التوقيت كما في الـ ساعة الرقمية أو إظهار درجة الحرارة أو قيمة الفولت والتيار في ساعات القيم الكهربائية وغيرها الكثير، وهي عبارة عن سبعة لدات موزعة بترتيب يمكن من تشكيل الأرقام من 0 حتى 9 وتسمى اللدات بالأحرف a,b,c,...g كما في الشكل وتقسم هذه الشاشات إلى نوعين:



الشاشات ذات المصاعد المشتركة: بهذه الحالة تكون جميع مصاعد اللدات موصولة إلى قطب مشترك يسمى المصعد المشترك ويكون مهبط كل لد موصول إلى قطب خارجي خاص مسؤول عن إضاءة هذا اللد.

الشاشات ذات المهابط المشتركة:

بهذه الحالة تكون جميع مهابط اللدات موصولة إلى قطب مشترك يسمى المهبط المشترك ويكون مصعد كل لد موصول إلى قطب خارجي خاص مسؤول عن إضاءة هذا اللد.

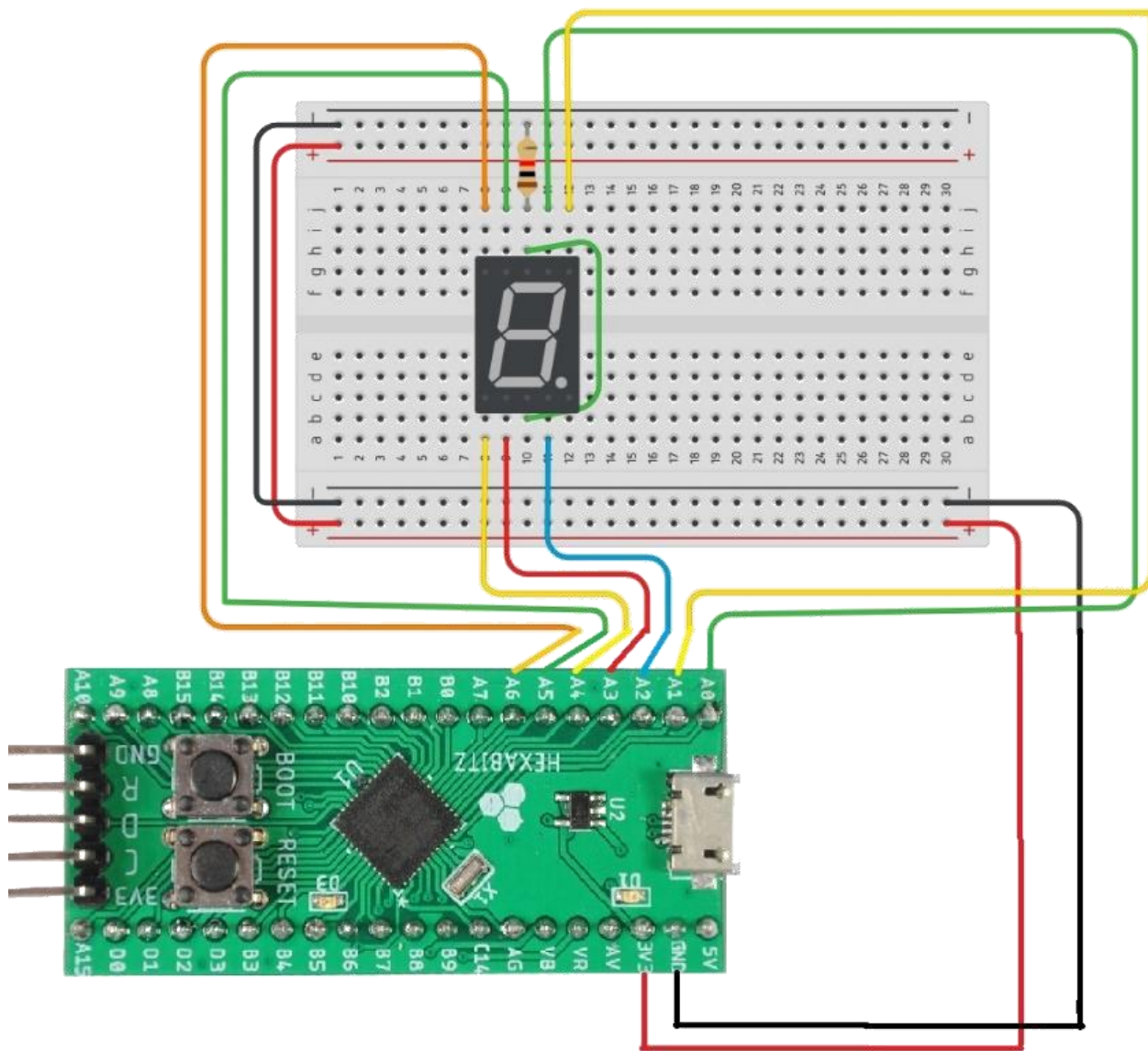
التوصيل:

من أجل الشاشات ذات المهابط المشتركة نقوم بتوصيل القطب المشترك إلى 0v ونوصل الأقطاب a,b,c,...g إلى منفذ الخرج الرقمي وليكن مثلا المنفذ A وبحسب القيمة على هذا المنفذ أي حسب القيمة الخارجة من المنفذ الرقمي ستضاء اللدات الموصولة معها ويوضح الجدول التالي توصيل شاشة ذات مهابط مشتركة.

الرقم على الشاشة	PA0 a	PA1 b	PA2 c	PA3 d	PA4 e	PA5 f	PA6 g	PA7 dot	القيمة ست عشريا على المنفذ A
0	1	1	1	1	1	1	0	0	0x3F
1	0	1	1	0	0	0	0	0	0x06
2	1	1	0	1	1	0	1	0	0x5B
3	1	1	1	1	0	0	1	0	0x4F
4	0	1	1	0	0	1	1	0	0x66
5	1	0	1	1	0	1	1	0	0x6D
6	1	0	1	1	1	1	1	0	0x7D

0x07	0	0	0	0	0	1	1	1	7
0x7F	0	1	1	1	1	1	1	1	8
0x6F	0	1	1	0	1	1	1	1	9

4- التطبيق العملي 2: برمجة شاشة واحدة ذات مصاعد مشتركة لتعمل كعداد يعد من 0 حتى 9 كل زمن معين وتكرر العملية بشكل دائم وذلك باستخدام البورد التطويري:

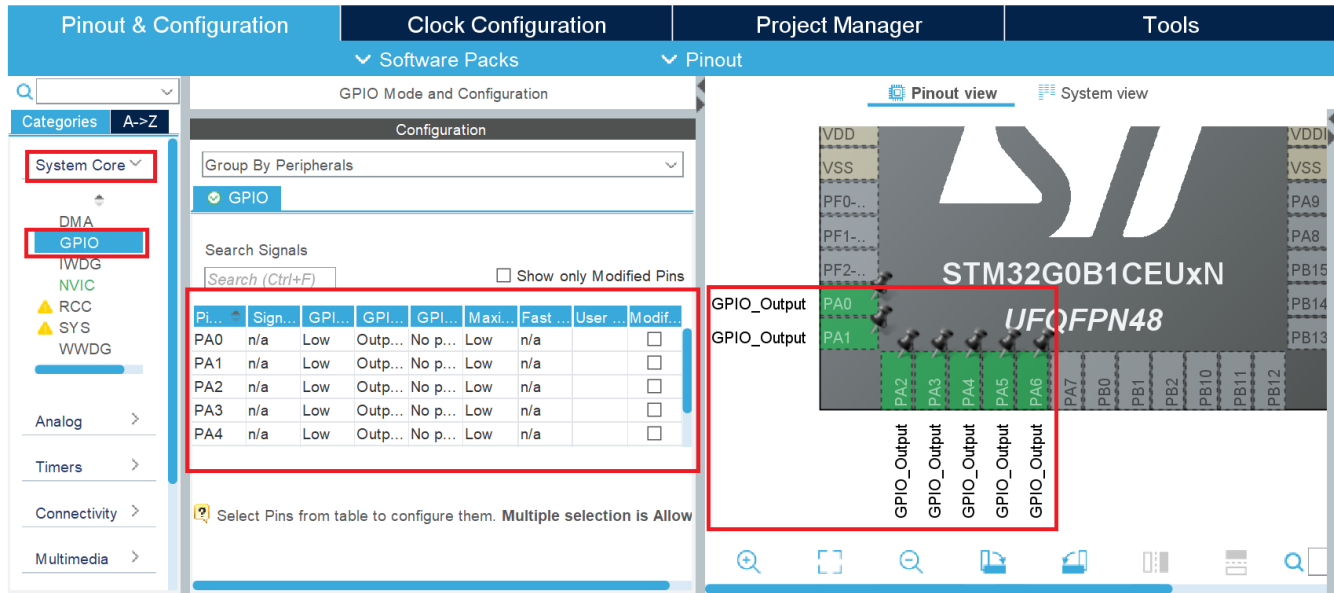


ضبط إعدادات المشروع:

الخطوة الأولى: فتح بيئة STM32CubeIDE وإنشاء مشروع جديد ثم قم باختيار المتحكم STM32G0B1CEU6N

الخطوة الثانية: اختيار اسم للمشروع

الخطوة الثالثة: اختر الأقطاب PA0:PA6 لضبطها كأقطاب خرج



الخطوة الرابعة: قم بضبط ساعة النظام على الساعة الداخلية (HSI) واختر التردد 8MHZ

الخطوة الخامسة: قم بتوليد الكود اعتماداً على الإعدادات التي قمت ب ضبطها من خلال الضغط على زر

Ctrl+S

يصبح الكود النهائي:

```
#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 = 0x003F; //Displaying 0
        HAL_Delay(1000);      //One second delay
    }
}
```

```

GPIOA ->ODR = 0x0006; //Displaying 1
HAL_Delay(1000);      //One second delay
GPIOA ->ODR = 0x005B; //Displaying 2
HAL_Delay(1000);      //One second delay
GPIOA ->ODR = 0x004F; //Displaying 3
HAL_Delay(1000);      //One second delay
GPIOA ->ODR = 0x0066; //Displaying 4
HAL_Delay(1000);      //One second delay
GPIOA ->ODR = 0x006D; //Displaying 5
HAL_Delay(1000);      //One second delay
GPIOA ->ODR = 0x007D; //Displaying 6
HAL_Delay(1000);      //One second delay
GPIOA ->ODR = 0x0007; //Displaying 7
HAL_Delay(1000);      //One second delay
GPIOA ->ODR = 0x007F; //Displaying 8
HAL_Delay(1000);      //One second delay
GPIOA ->ODR = 0x0067; //Displaying 9
HAL_Delay(1000);      //One second delay
    }
}

```

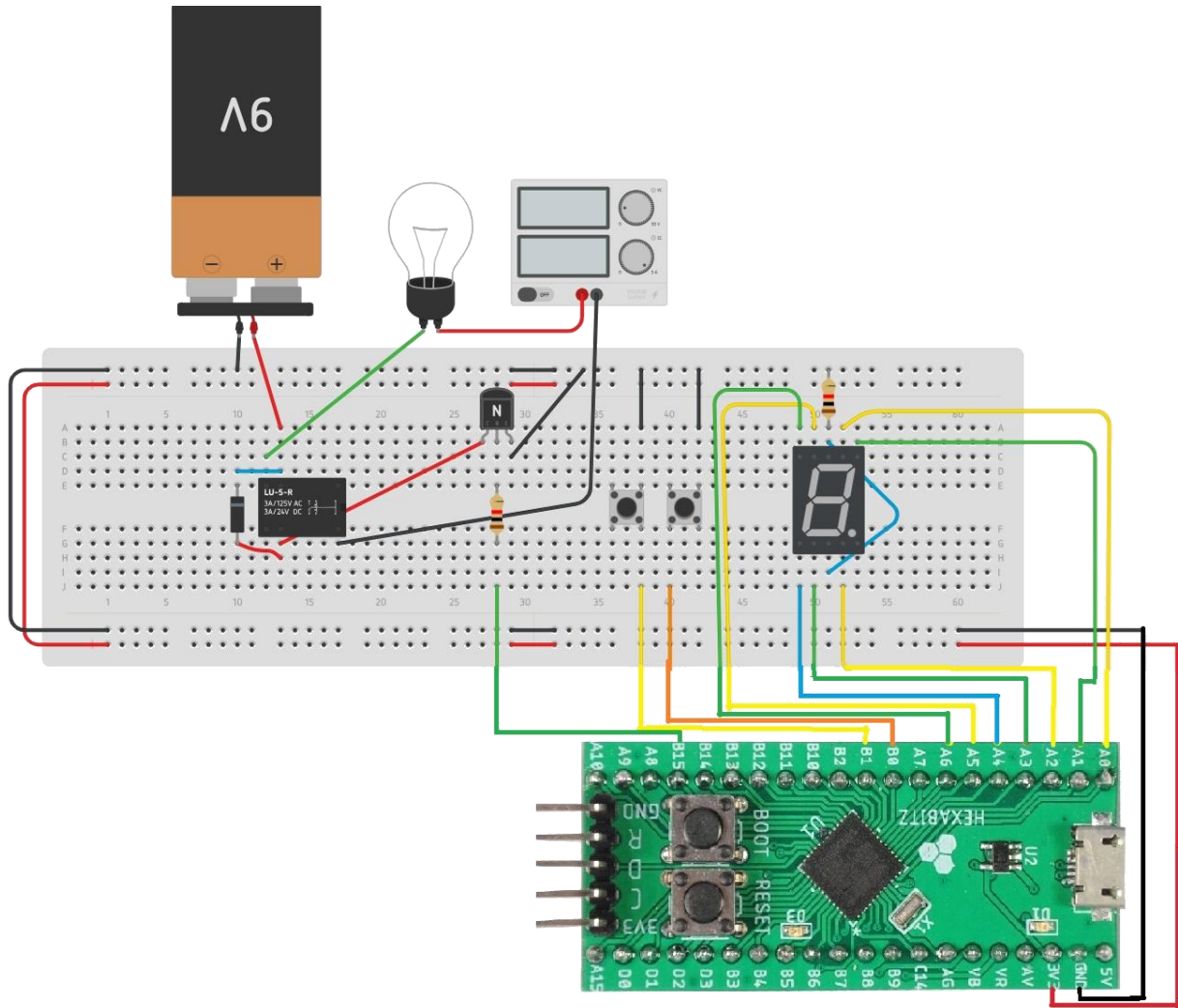
الخطوة السادسة:

التأكد من خلو الكود من أي أخطاء برمجية من خلال تنظيف الكود clean project ثم ترجمة الكود من خلال Build Project من أجل توليد ملف .hex

الخطوة السابعة:

قم باستخدام STM32CubeProgrammer برفع الكود للمتحكم الموجود على البورد.

التطبيق العملي 3: تشغيل جهاز موصول مع ترانزستور وريليه على القطب PB15 من خلال زر لحظي موصول على القطب PB0 باستخدام المقاطعة، وإطفائه من خلال زر لحظي موصول على القطب PB1 أيضاً باستخدام المقاطعة، وإظهار حالة الجهاز على شاشة الإظهار بحيث يتم كتابة الرقم (1) في حال كان الخرج مفعل (On) وإظهار الرقم (0) في حال كان الخرج مطفئ (Off)، باستخدام البورد التطويري



ضبط إعدادات المشروع:

الخطوة الأولى: فتح بيئة STM32CubeIDE وإن شاء م شروع جديد ثم اختيار المتحكم

STM32G0B1CEU6N

الخطوة الثانية: اختيار اسم للمشروع

الخطوة الثالثة: اختر الأقطاب PA0:PA6 لضبطها كأقطاب خرج، و ضبط القطب PB15 كقطب

خرج، والقطب PB0 والقطب PB1 كأقطاب مقاطعة خارجية.

Pinout & Configuration | Clock Configuration | Project Manager | Tools

Software Packs | Pinout

GPIO Mode and Configuration

Configuration

Group By Peripherals

GPIO | NVIC

Search Signals

Search (Ctrl+F) ☐ Show only Modified Pins

Pin	Signal	GPIO	GPIO	GPIO	Maxim	Fast	User	Modifi
PA5	n/a	Low	Output...	No pu...	Low	n/a		<input type="checkbox"/>
PA6	n/a	Low	Output...	No pu...	Low	n/a		<input type="checkbox"/>
PB0	n/a	n/a	Exter...	Pull-up	n/a	n/a		<input checked="" type="checkbox"/>
PB1	n/a	n/a	Exter...	Pull-up	n/a	n/a		<input checked="" type="checkbox"/>
PB15	n/a	Low	Output...	No pu...	Low	n/a		<input type="checkbox"/>

PB0 Configuration :

GPIO mode: External Interrupt Mode with Falling edge trigger

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

Pinout view | System view

STM32G0B1CEUxN

UFQFPN48

GPIO_Output

GPIO_Output

GPIO_Output

GPIO_Output

GPIO_Output

GPIO_Output

GPIO_EXTI0

GPIO_EXTI1

قم بتفعيل المقاطعات على الأقطاب:

Pinout & Configuration | Clock Configuration | Project Manager | Tools

Software Packs | Pinout

GPIO Mode and Configuration

Configuration

Group By Peripherals

GPIO | NVIC

NVIC Interrupt Table

	Enabled	Preemption Priority
EXTI line 0 and line 1 interrupts	<input checked="" type="checkbox"/>	0

Pinout view | System view

STM32G0B1CEUxN

UFQFPN48

GPIO_Output

GPIO_Output

GPIO_Output

GPIO_Output

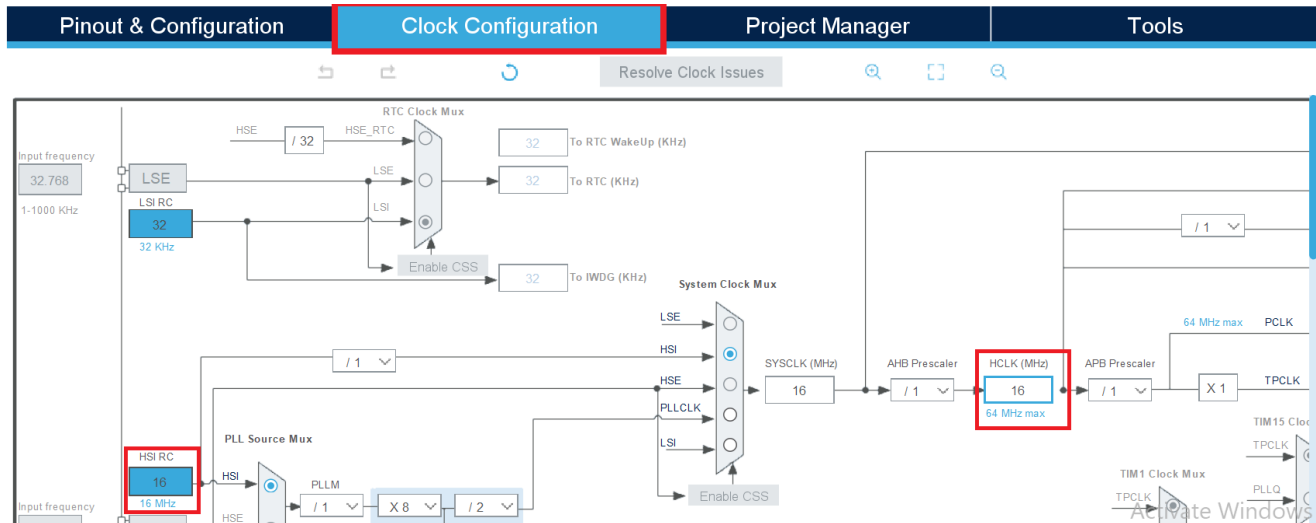
GPIO_Output

GPIO_Output

GPIO_EXTI0

GPIO_EXTI1

الخطوة الرابعة: قم بضبط ساعة النظام على الساعة الداخلية (HSI) واختر التردد 16MHZ



الخطوة الخامسة: قم بتوليد الكود اعتماداً على الإعدادات التي قمت بـ ضبطها من خلال الضغط على زر
Ctrl+S

يصبح الكود النهائي:

```
#include "main.h"
void SystemClock_Config(void);
static void MX_GPIO_Init(void);

void HAL_GPIO_EXTI_Falling_Callback(uint16_t GPIO_Pin)
{
    if(GPIO_Pin == GPIO_PIN_0)
    {
        HAL_GPIO_WritePin(GPIOB, GPIO_PIN_15, 0);
        GPIOA->ODR = 0x003f; // 0x0040; //Displaying 0
    }
    if(GPIO_Pin == GPIO_PIN_1)
    {
        HAL_GPIO_WritePin(GPIOB, GPIO_PIN_15, 1);
        GPIOA->ODR = 0x0006; // 0x0079; //Displaying 1
    }
}
```

```

int main(void)
{
    HAL_Init();
    SystemClock_Config();
    MX_GPIO_Init();
    GPIOA->ODR = 0x003f; //0x0040; //Displaying 0
    while (1)
    {

    }
}

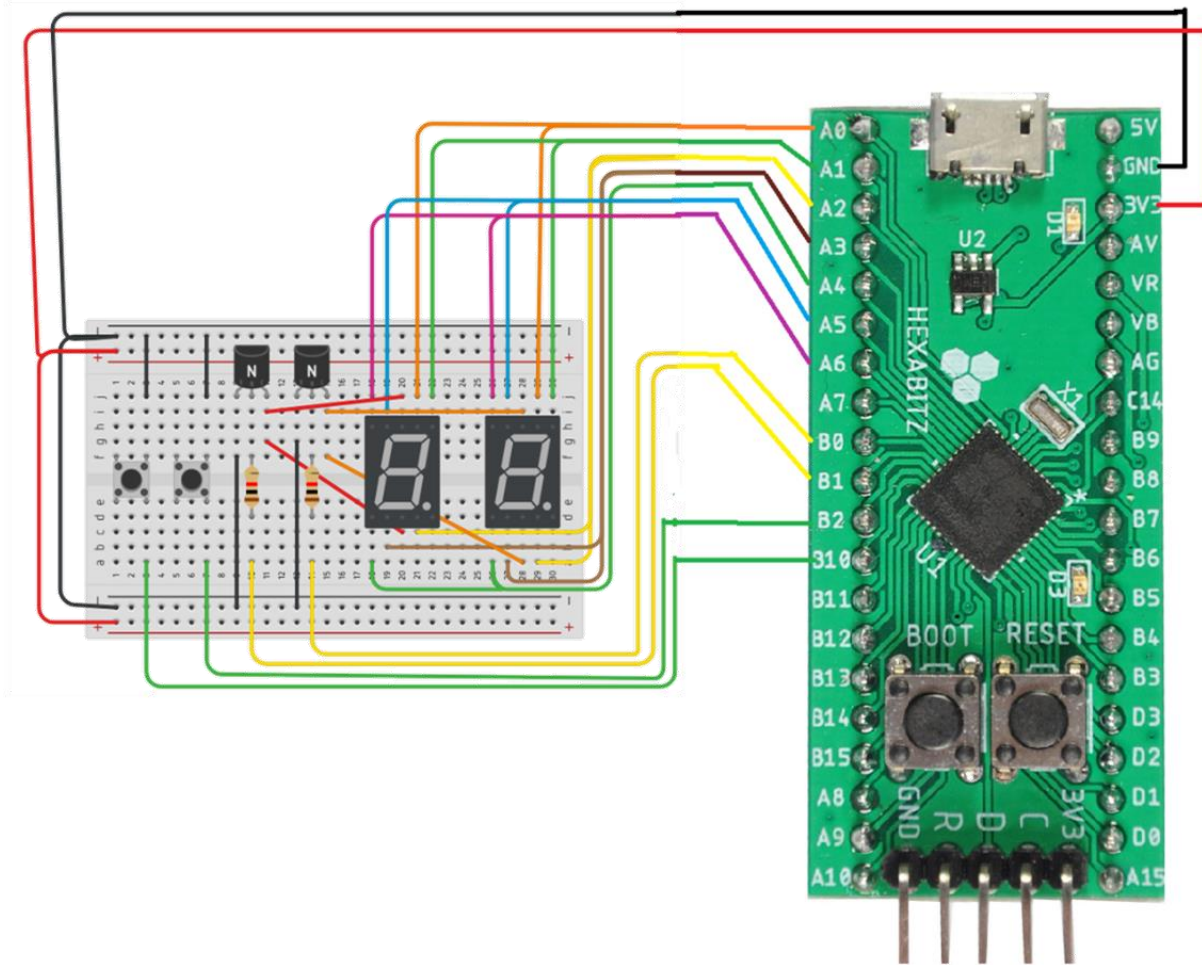
```

الخطوة السادسة: التأكد من خلو الكود من أي أخطاء برمجية من خلال تنظيف الكود clean project ثم ترجمة الكود من خلال Build Project من أجل توليد ملف .hex

الخطوة السابعة:

قم باستخدام STM32CubeProgrammer برفع الكود للمتحكم الموجود على البورد.

التطبيق العملي 4: عداد زوار وإظهار العدد على شاشة 7-seg بخانتين ,حيث ان الكباسين موصولين على اقطاب المقاطعة الخارجية (INT0,INT1) ويتم تفعيل المقاطعتين عند الجبهة الهابطة باستخدام البورد التطويري



ضبط إعدادات المشروع:

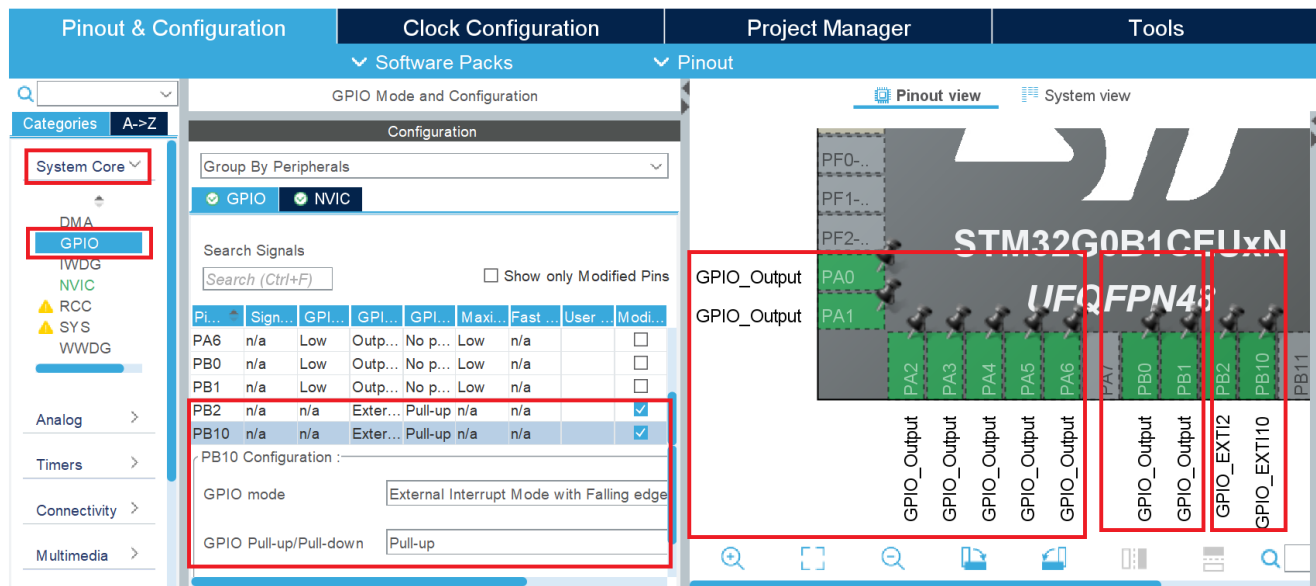
الخطوة الأولى: فتح بيئة STM32CubeIDE وإن شاء م شروع جديد ثم اختيار المتحكم

STM32G0B1CEU6N

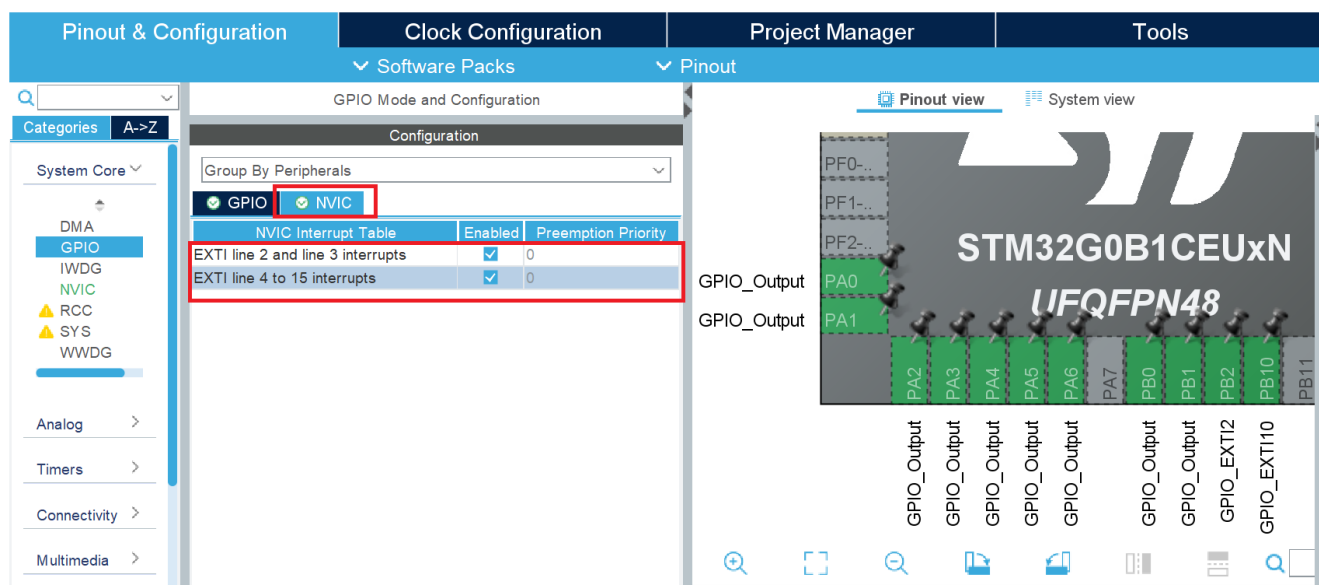
الخطوة الثانية: اختيار اسم للمشروع

الخطوة الثالثة: اختر الأقطاب PA0:PA6 ، والأقطاب PB0,PB1 ل ضبطها كأقطاب خرج،

والأقطاب PB2,PB10 كأقطاب مقاطعة خارجية



وقم بتفعيل المقاطعة:



الخطوة الرابعة: قم بضبط ساعة النظام على الساعة الداخلية (HSI) واختر التردد 16MHZ

الخطوة الخامسة: قم بتوليد الكود اعتماداً على الإعدادات التي قمت ب ضبطها من خلال الضغط على زر Ctrl+S

يصبح الكود النهائي:

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

```

void SystemClock_Config(void);
static void MX_GPIO_Init(void);
int8_t i=0;
void HAL_GPIO_EXTI_Falling_Callback(uint16_t GPIO_Pin)
{
    if(GPIO_Pin == GPIO_PIN_2)
    {
        i++;
        if(i==100)
            i=99;
    }
    else if(GPIO_Pin ==GPIO_PIN_10)
    {
        if(i>0)
            i--;
    }
}

int main(void)
{
    HAL_Init();
    SystemClock_Config();
    MX_GPIO_Init();
    /* USER CODE BEGIN WHILE */
    unsigned char
decode[10]={0x003f,0x0006,0x005B,0x004f,0x0066,0x006d,0x007d
,0x0007, 0x007f,0x006f};
    unsigned char a,b;
    while (1)
    {
        a=i%10;
        HAL_GPIO_WritePin(GPIOB, GPIO_PIN_0, 1);
        HAL_GPIO_WritePin(GPIOB, GPIO_PIN_1, 0);
        GPIOA->ODR = decode[a]; //Displaying ones
        HAL_Delay(5);
        b=i/10;
        HAL_GPIO_WritePin(GPIOB, GPIO_PIN_0, 0);
        HAL_GPIO_WritePin(GPIOB, GPIO_PIN_1, 1);
        GPIOA->ODR = decode[b]; //Displaying tens
    }
}

```

```
        HAL_Delay(5);  
    }  
  
    /* USER CODE END WHILE */  
}
```

الخطوة السادسة: التأكد من خلو الكود من أي أخطاء برمجية من خلال تنظيف الكود clean project ثم ترجمة الكود من خلال Build Project من أجل توليد ملف .hex
الخطوة السابعة:
قم باستخدام STM32CubeProgrammer برفع الكود للمتحكم الموجود على البورد.

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

- <https://youtu.be/Sj3JCy9NBVs>
- <https://youtu.be/vlwIFrN1Ze0>
- <https://youtu.be/A12RjL907vw>
- <https://youtu.be/JO3GivrCFos>