

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

مقرر \_\_\_\_\_

الجلسة الرابعة

السنة \_\_\_\_\_

2023/2202

## الغاية من الجلسة:

- التعرف بشاشات الـ LCD
- عرض تطبيقات مختلفة مع عرض خطوات تنزيل المكتبة المناسبة لشاشات الـ LCD

## شاشات الـ LCD:

وهي عبارة عن شاشة مؤلفة من سطر أو أكثر يحتوي كل سطر على عدد من الخانات, والخانة هي عبارة عن مربع صغير أبعاده ( 5\*8 ) بكسل (Pixel)

وكل خانة تستطيع إظهار محرف واحد فقط.

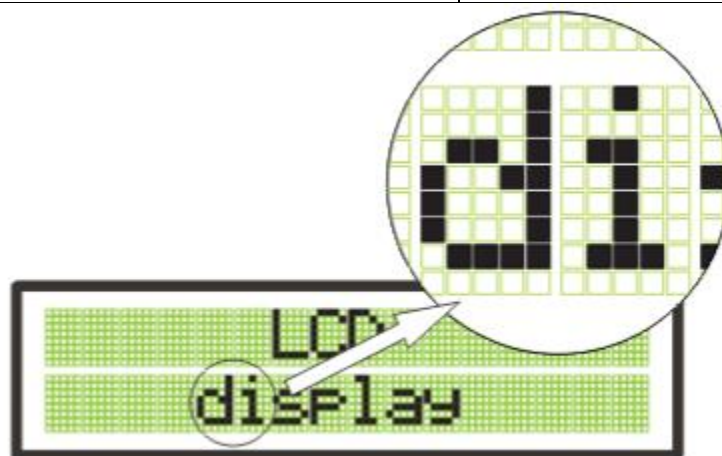
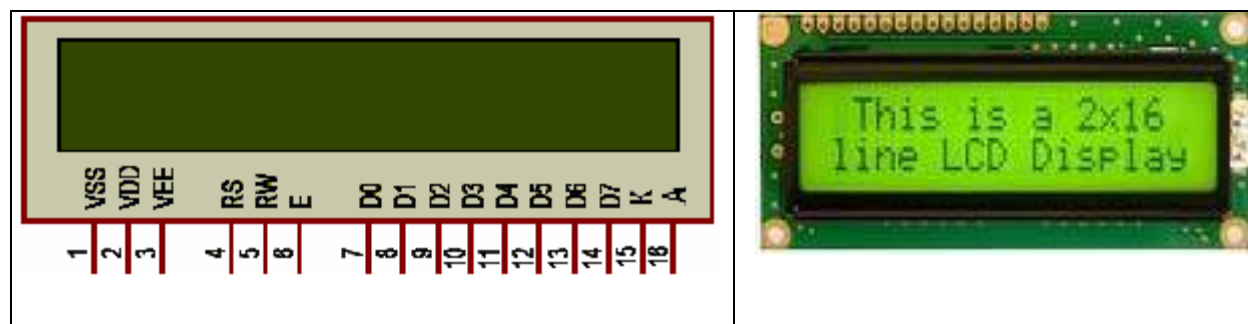
وأكثر الشاشات شيوعاً هي الشاشات ذات القياسات التالية : 2x16 4x20 2x20 4x20 1x40

تتم تغذية هذه الشاشة بجهد +5V ويمكن ملائمتها بواسطة 4 أقطاب أو 8 أقطاب.

وتتميز هذه الشاشات بزوايا رؤية كبيرة واستهلاك منخفض للطاقة.

تملك شاشة الإظهار الكريستالية معالج إظهار خاص و تزود شاشة الـ LCD بذاكرة داخلية

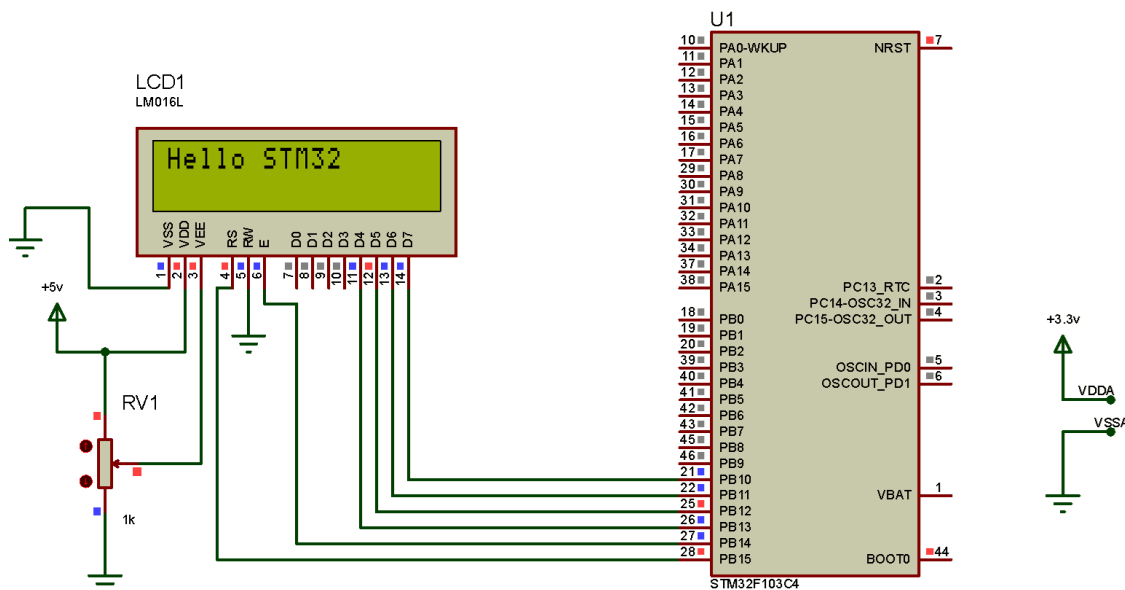
ومن الممكن إظهار جميع رموز الأسكي عليها وهي (189) رمز مختلف.



شرح أقطاب الـ LCD :

- القطب VSS : هو قطب التغذية لشاشة الـ LCD , وهو جهد الأرضي (0) منطقي
- القطب VDD : هو أيضاً قطب التغذية لشاشة الـ LCD , ولكن ذو القيمة ( +5V )
- القطب VEE : هو قطب جهد التباين , ويقصد بالتباين هو حدة ظهور الرمز على الشاشة وأقل تباين أن لا نرى شيئاً على الشاشة يكون عند تطبيق ( +5V ) على هذا القطب , وأعلى تباين للشاشة يكون عند تطبيق ( 0 V ) .
- القطب RS : وهو مسجل اختيار الدخول لشاشة الـ LCD وذلك في حال طبق عليه (0) منطقي عندها نريد إرسال كلمة تحكم , بينما في حال طبق (1) منطقي فعندها نريد إرسال معطيات .
- القطب R/W : وهو للقراءة أو الكتابة إلى الشاشة , نطبق (0) منطقي على هذا القطب عندما نريد كتابة ( إرسال ) المعلومات إلى شاشة الـ LCD , ونطبق (1) منطقي عندما نريد قراءة ( استقبال ) المعلومات من شاشة الـ LCD .
- القطب E : وهو قطب تمكين شاشة الـ LCD , فكل معلومة يتم كتابتها أو قراءتها من شاشة الـ LCD يجب إرفاقها بنبضة تمكين على هذا القطب .
- الأقطاب D0----D7 : هي أقطاب المعطيات ( DATA ) , حيث يتم كتابة المعطيات أو كلمات التحكم عبر هذه الأقطاب إلى شاشة الـ LCD وكذلك قراءة المعطيات ويتم عملياً استخدام الأقطاب D4--D7 فقط لعملية القراءة و الكتابة .
- القطبين K و A تزود بعض الشاشات بهذين القطبين , وهما على الترتيب قطبي المهبط و المصعد للد الإضاءة الخلفية لشاشة الـ LCD .

### التطبيق العملي 1 : إظهار جملة Hello STM32 على شاشة الـ LCD

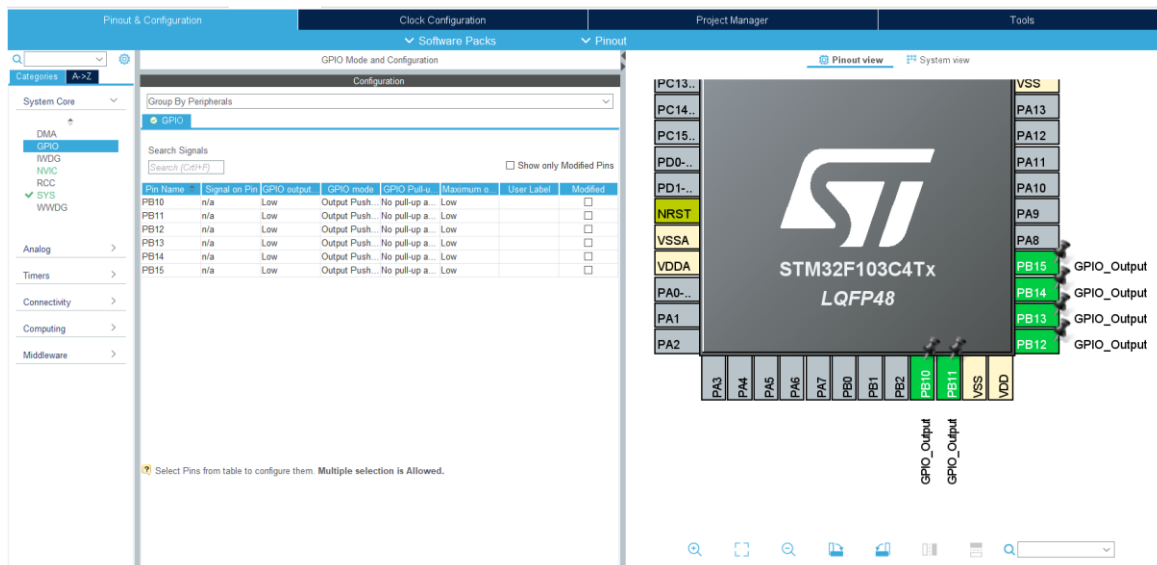


ضبط إعدادات المشروع:

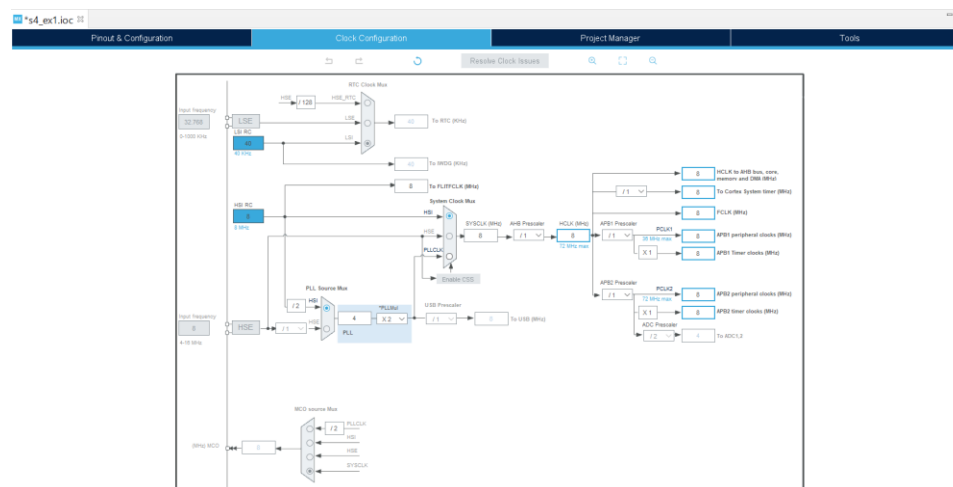
**الخطوة الأولى:** فتح بيئة STM32CubeIDE وإن شاء م شروع جديد ثم اختيار المتحكم المناسب سب (STM32F103C4) عند استخدام المحاكاة ، و المتحكم STM32G0 عند التطبيق العملي

**الخطوة الثانية:** اختيار اسم للمشروع

**الخطوة الثالثة:** اختر الأقطاب PB10:PB15 كأقطاب خرج



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

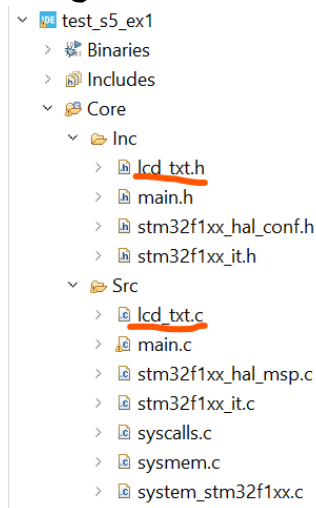


**الخطوة السادسة:** قم بتوليد الكود اعتماداً على الإعدادات التي قمت بضبطها من خلال الضغط على زر Ctrl+S

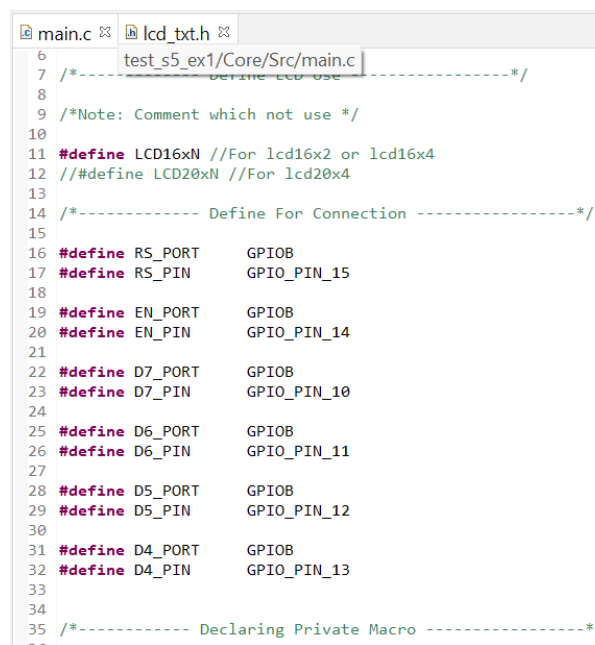
**الخطوة السابعة :** إضافة مكتبة للـ LCD لتسهيل التعامل معها من خلال الخطوات التالية:

1- نقوم بنسخ الملف lcd\_txt.h ثم لصقه ضمن المجلد inc للمشروع

2- نقوم بنسخ الملف lcd\_txt.c ثم لصقه ضمن المجلد src للمشروع



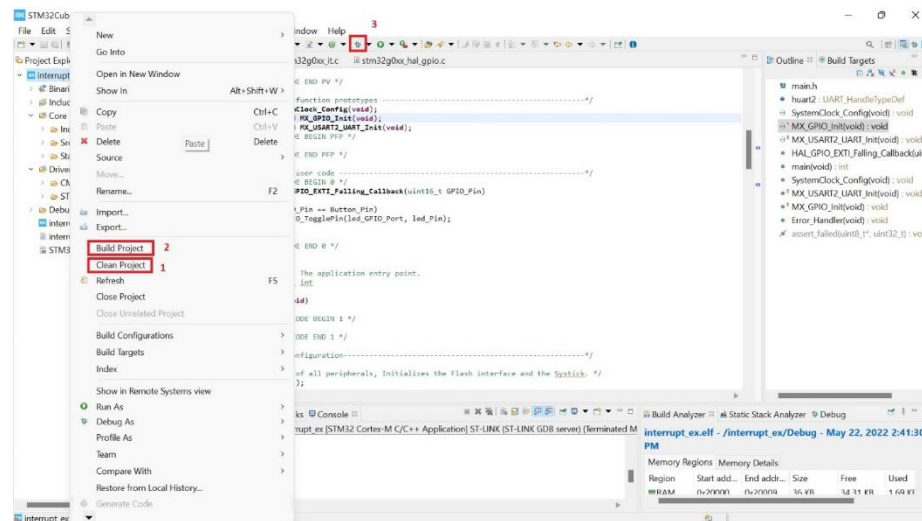
3- نقوم بتعديل أسماء الأقطاب الموصولة مع الشاشة حسب التوصيل الخاص بمشروعنا



4- نستخدم الدوال التي تؤمنها المكتبة للتعامل مع الشاشة وهي:

```
/*----- Declaring Function Prototype -----*/
void lcd_init(void);
void lcd_write(uint8_t type, uint8_t data);
void lcd_puts(uint8_t x, uint8_t y, int8_t *string);
void lcd_clear(void);
```

**الخطوة الثامنة:** نقوم بكتابة الكود المنا سب ثم نقوم بالتأكد من خلو الكود من أي أخطاء برمجية من خلال تنظيف الكود clean project ثم ترجمة الكود من خلال Build Project ثم رفع الكود للمتحكم (متحكم لوحة Nucleo) ومراقبته عبر Debug.



**الكود بالكامل:**

```
#include "main.h"
#include "lcd_txt.h"
void SystemClock_Config(void);
static void MX_GPIO_Init(void);
int main(void)
{
    HAL_Init();
    SystemClock_Config();
    MX_GPIO_Init();
    /* USER CODE BEGIN 2 */
    lcd_init();
    lcd_puts(0,0,"Hello STM32");

    while (1)
    {

    }
    /* USER CODE END 2 */
}
```

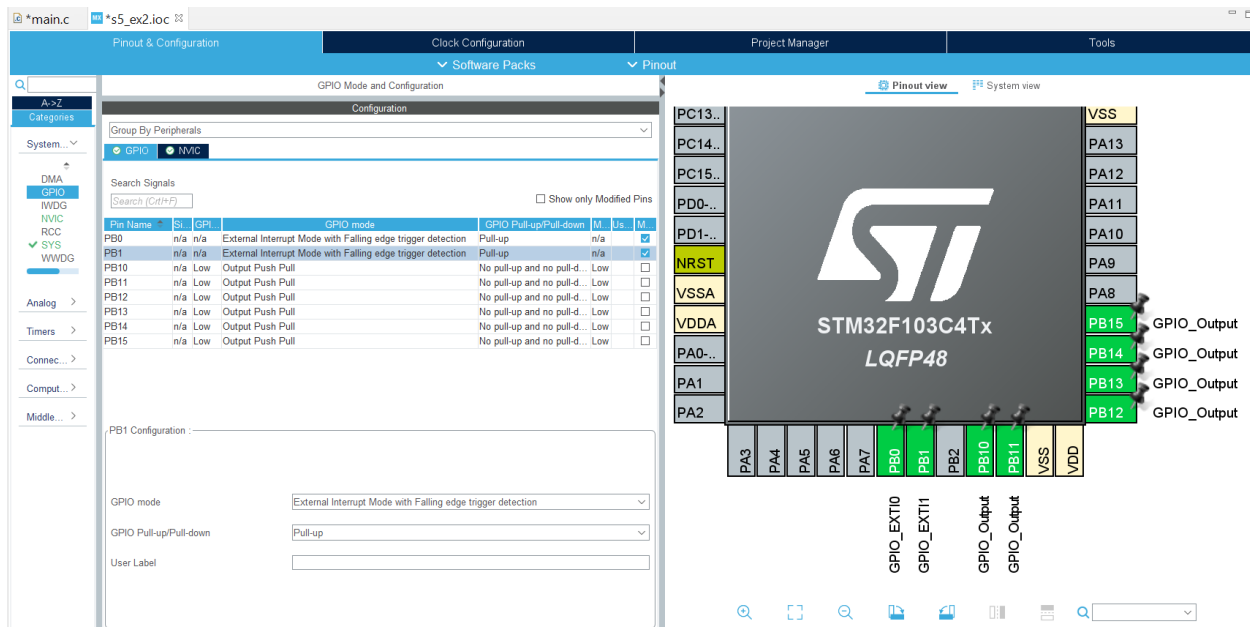
**التطبيق العملي 2:** إظهار عداد زوار و إظهار العدد على شاشة LCD حيث الد ساس الأول يعبر عن زيادة العدد ويتم وصله على المدخل PB0 و الد ساس الثاني يعبر عن إنقاص العدد ويتم وصله على المدخل PB1 ، ا ستخدم المقاطعات الخارجية

**ضبط إعدادات المشروع:**

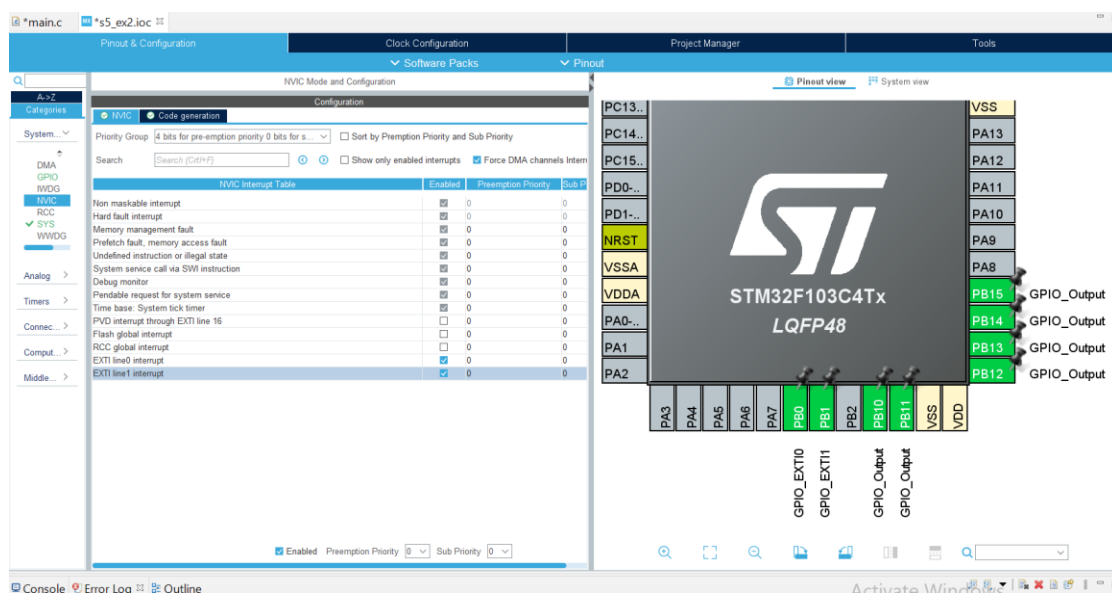
**الخطوة الأولى:** فتح بيئة STM32CubeIDE وإن شاء م شروع جديد ثم اختيار المتحكم المنا سب (STM32F103C4) عند استخدام المحاكاة ، و المتحكم STM32G0 عند التطبيق العملي

**الخطوة الثانية:** اختيار اسم للمشروع

**الخطوة الثالثة:** اختر الأقطاب PB10:PB15 لضبطها كأقطاب خرج، والأقطاب PB0:PB1 كأقطاب مقاطعة خارجية



**الخطوة الرابعة:** تفعيل المقاطعة الخارجية على الأقطاب PB0:PB1



**الخطوة الخامسة:** ضبط تردد الساعة للمتحكم على التردد 8MHZ أثناء الـ simulation

**الخطوة السادسة:** إضافة مكتبة الـ LCD كما فعلنا في التطبيق السابق

**الخطوة السابعة:** توليد الكود من خلال الضغط على ctrl+s

## الكود النهائي:

```

#include "main.h"
#include "lcd_txt.h"
void SystemClock_Config(void);
static void MX_GPIO_Init(void);
int8_t i=0;
void HAL_GPIO_EXTI_Callback(uint16_t GPIO_Pin)
{
    if(GPIO_Pin ==GPIO_PIN_0)
    {
        i++;
        if(i==10)
            i=9;
    }
    else if(GPIO_Pin ==GPIO_PIN_1)
    {
        if(i>0)
            i--;
    }
}

int main(void)
{
    unsigned char buffer ;
    HAL_Init();
    SystemClock_Config();
    MX_GPIO_Init();
    /* USER CODE BEGIN 2 */
    lcd_init();
    lcd_clear();

    while (1)
    {
        snprintf(buffer, 10, "%d", i);
        lcd_puts(0,0,"people=");
        lcd_puts(0,10,buffer);
        HAL_Delay(50);
    }
    /* USER CODE END 2 */
}

```

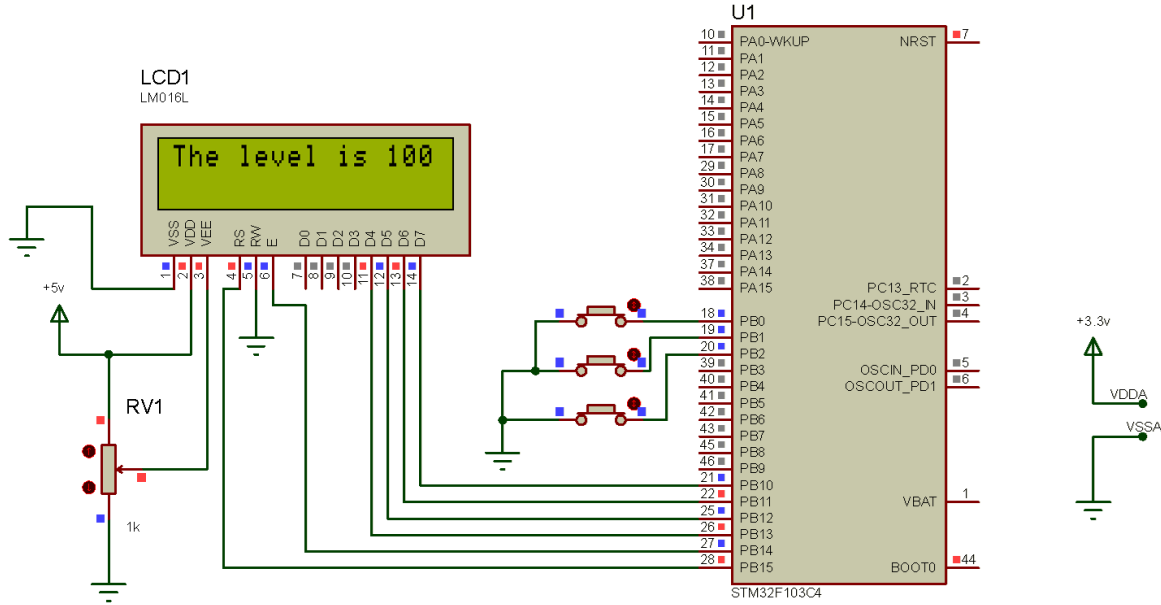
## ملاحظة:

الخطوات التي تم اتباعها لإضافة المكتبة الخاصة بشاشة الـ LCD هي نفسها لإضافة أي مكتبة على سبيل المثال الـ Driver الخاص بالـ :

- Servo Motor
- Stepper Motor
- I2C\_LCD
- KeyPAD
- CapTouch
- 7Segments Display
- MPU6050
- OLED
- RTC
- ...



**التطبيق العملي 3:** قياس مستوى سائل باستخدام حساسات رقمية و إظهار المستوى على شاشة الإظهار الكريستالية : علماً أن المداخل تم تفعيل مقاومات الرفع لها و التوصيل وفق الشكل التالي.

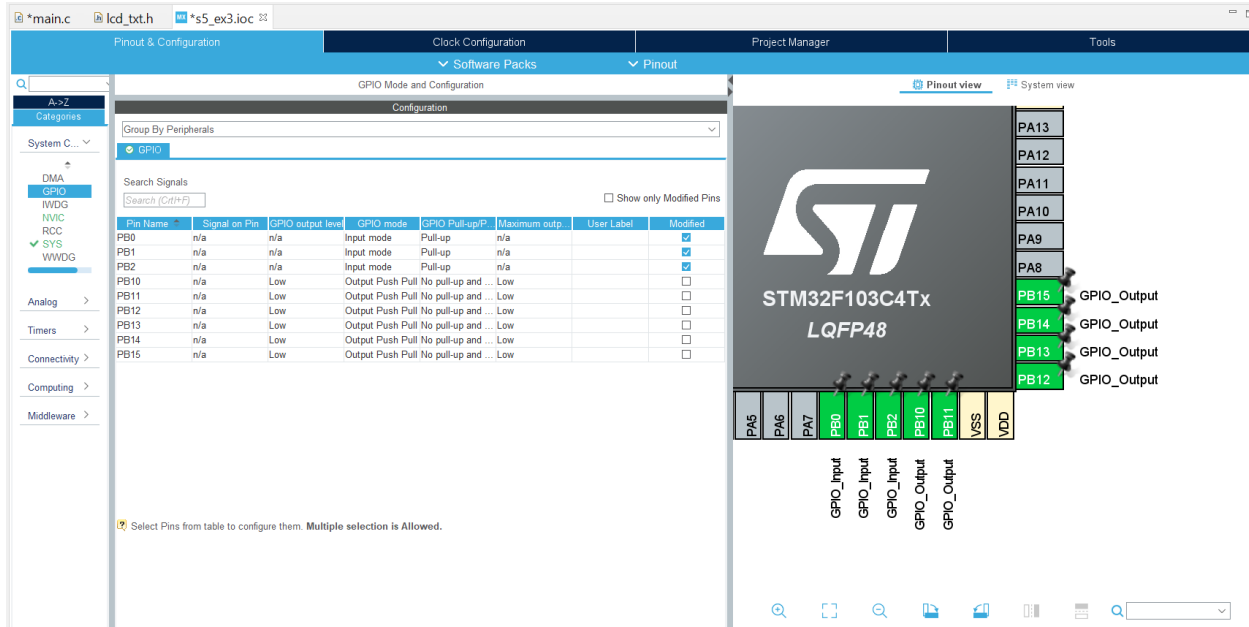


### ضبط إعدادات المشروع:

**الخطوة الأولى:** فتح بيئة STM32CubeIDE وإن شاء م شروع جديد ثم اختيار المتحكم المنا سب (STM32F103C4) عند استخدام المحاكاة ، و المتحكم STM32G0 عند التطبيق العملي

**الخطوة الثانية:** اختيار اسم للمشروع

**الخطوة الثالثة:** اختر الأقطاب PB10:PB15 ل ضبطها كأقطاب خرج، والأقطاب PB0:PB1 ل ضبطها كأقطاب دخل مع تفعيل مقاومة الرفع الداخلية



الخطوة الرابعة: ضبط تردد الساعة للمتحكم على التردد 8MHZ أثناء الـ simulation

الخطوة الخامسة: إضافة مكتبة الـ LCD كما فعلنا في التطبيق السابق

الخطوة السادسة: توليد الكود من خلال الضغط على ctrl+s

يصبح الكود النهائي:

```
#include "main.h"
#include "lcd_txt.h"
void SystemClock_Config(void);
static void MX_GPIO_Init(void);
int main(void)
{
    HAL_Init();
    SystemClock_Config();
    MX_GPIO_Init();
    /* USER CODE BEGIN 2 */
    lcd_init();

    while (1)
    {
        lcd_clear();
        if((HAL_GPIO_ReadPin(GPIOB, GPIO_PIN_0)==GPIO_PIN_SET) && (HAL_GPIO_ReadPin(GPIOB,
GPIO_PIN_1)==GPIO_PIN_SET)&& (HAL_GPIO_ReadPin(GPIOB, GPIO_PIN_2)==GPIO_PIN_SET))
        {
            lcd_puts(0,0,"The level is 0%");//Displaying on Lcd
        }
        else if((HAL_GPIO_ReadPin(GPIOB, GPIO_PIN_0)==GPIO_PIN_RESET) && (HAL_GPIO_ReadPin(GPIOB,
GPIO_PIN_1)==GPIO_PIN_SET)&& (HAL_GPIO_ReadPin(GPIOB, GPIO_PIN_2)==GPIO_PIN_SET))
        {
            lcd_puts(0,0,"The level is 30%");//Displaying on Lcd
        }
        else if((HAL_GPIO_ReadPin(GPIOB, GPIO_PIN_0)==GPIO_PIN_RESET) && (HAL_GPIO_ReadPin(GPIOB,
GPIO_PIN_1)==GPIO_PIN_RESET)&& (HAL_GPIO_ReadPin(GPIOB, GPIO_PIN_2)==GPIO_PIN_SET))
        {
            lcd_puts(0,0,"The level is 60%");//Displaying on Lcd
        }
    }
}
```

```
else if((HAL_GPIO_ReadPin(GPIOB, GPIO_PIN_0)==GPIO_PIN_RESET) && (HAL_GPIO_ReadPin(GPIOB,
GPIO_PIN_1)==GPIO_PIN_RESET)&& (HAL_GPIO_ReadPin(GPIOB, GPIO_PIN_2)==GPIO_PIN_RESET))
{
    lcd_puts(0,0,"The level is 100%");//Displaying on Lcd
}
else
    lcd_puts(0,0,"Error");//Displaying on Lcd

    HAL_Delay(50);
}
/* USER CODE END 2 */
}
```