

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

مقرر المتحكمات المصغرة الجلسة الرابعة

السنة الرابعة ميكاترونيك

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.

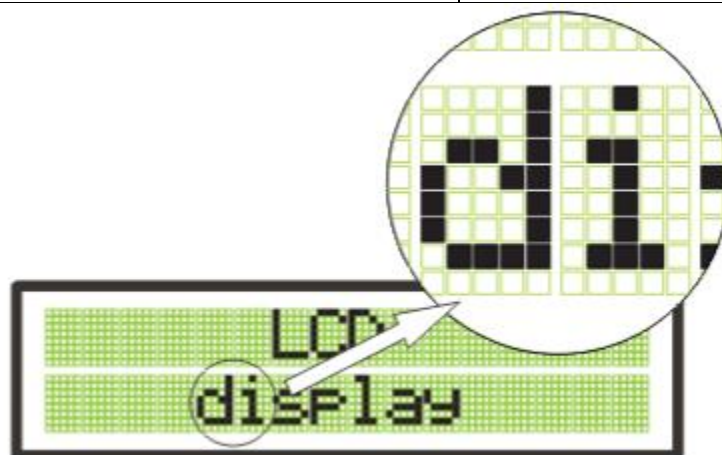
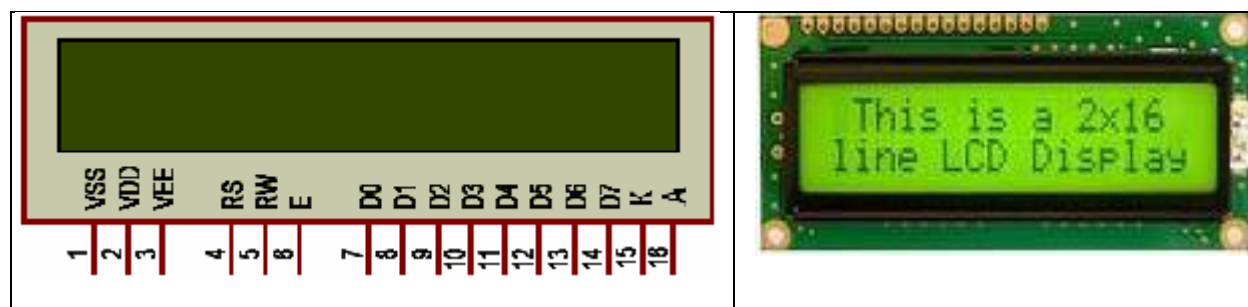
وأكثر الشاشات شيوعاً هي الشاشات ذات القياسات التالية : 40x1 20x4 20x2 16x4 16x2

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

وتتميز هذه الشاشات بزواوية رؤية كبيرة واستهلاك منخفض للطاقة.

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

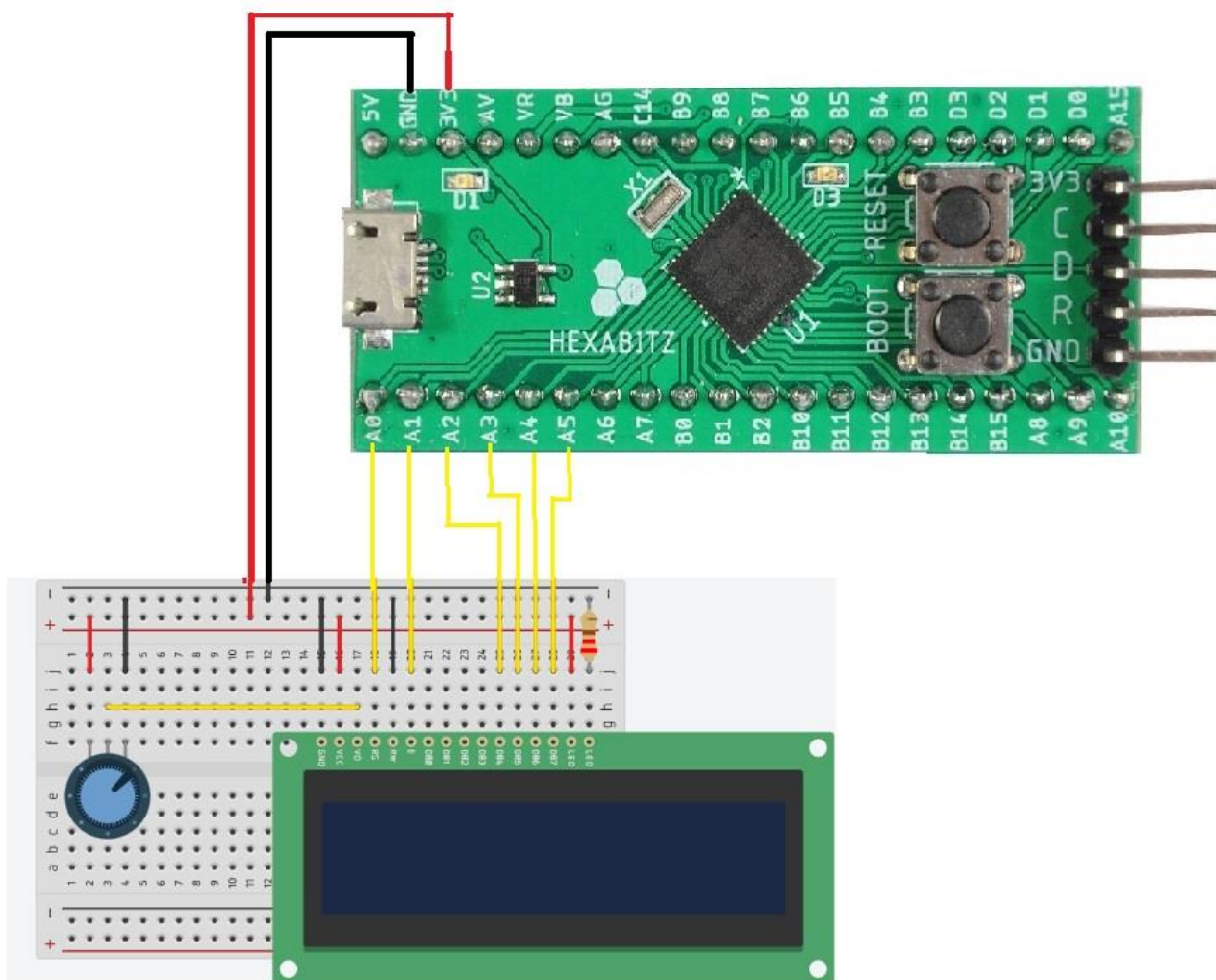
ومن الممكن إظهار جميع رموز الأسكي عليها وهي (189) رمز مختلف.



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

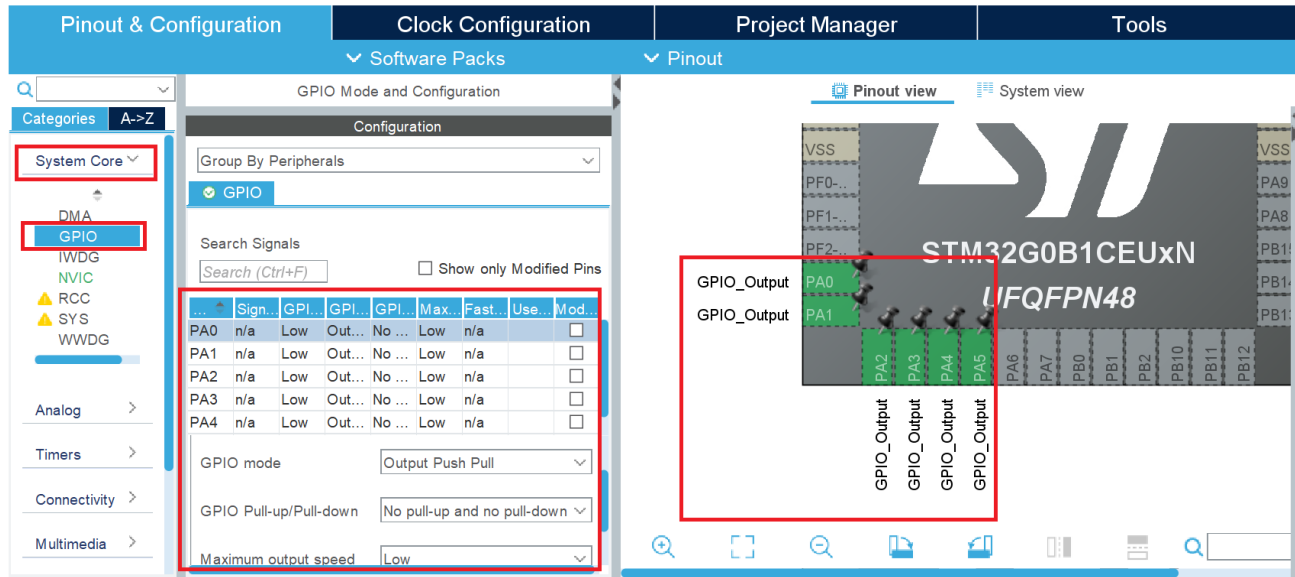


ضبط إعدادات المشروع:

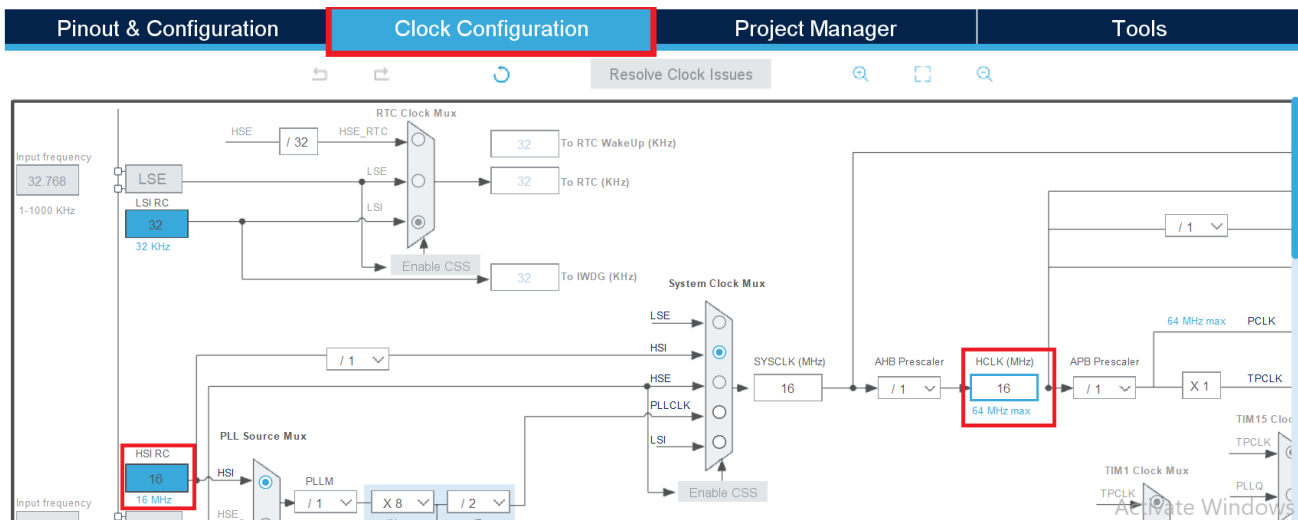
الخطوة الأولى: فتح بيئة STM32CubeIDE وإنشاء مشروع جديد ثم اختيار المتحكم STM32G0B1CEU6N عند التطبيق العملي

الخطوة الثانية: اختيار اسم للمشروع

الخطوة الثالثة: اختر الأقطاب PA0:PA5 كأقطاب خرج



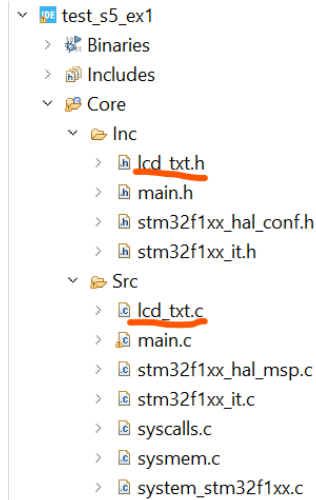
الخطوة الرابعة: قم بضبط ساعة النظام على الساعة الداخلية (HSI) واختر التردد 16MHZ



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

```

s4_ex1.ioc  main.c  *lcd_txt.h x
3
4 #include "stm32G0xx_hal.h"
5 /*----- Define LCD Use -----
6 /*Note: Comment which not use */
7 #define LCD16xN //For lcd16x2 or lcd16x4
8 // #define LCD20xN //For lcd20x4
9 /*----- Define For Connection -----
10 #define RS_PORT      GPIOA
11 #define RS_PIN       GPIO_PIN_0
12 #define EN_PORT      GPIOA
13 #define EN_PIN       GPIO_PIN_1
14 #define D7_PORT      GPIOA
15 #define D7_PIN       GPIO_PIN_5
16 #define D6_PORT      GPIOA
17 #define D6_PIN       GPIO_PIN_4
18 #define D5_PORT      GPIOA
19 #define D5_PIN       GPIO_PIN_3
20 #define D4_PORT      GPIOA
21 #define D4_PIN       GPIO_PIN_2
22

```

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);

```

الكود بالكامل:

```

#include "main.h"
#include "lcd_txt.h"
void SystemClock_Config(void);
static void MX_GPIO_Init(void);
int main(void)
{
    int8_t* str = (int8_t*)"Hello STM32";

    HAL_Init();
    SystemClock_Config();
    MX_GPIO_Init();
    /* USER CODE BEGIN 2 */
    lcd_init();
    lcd_puts(0,0,str);

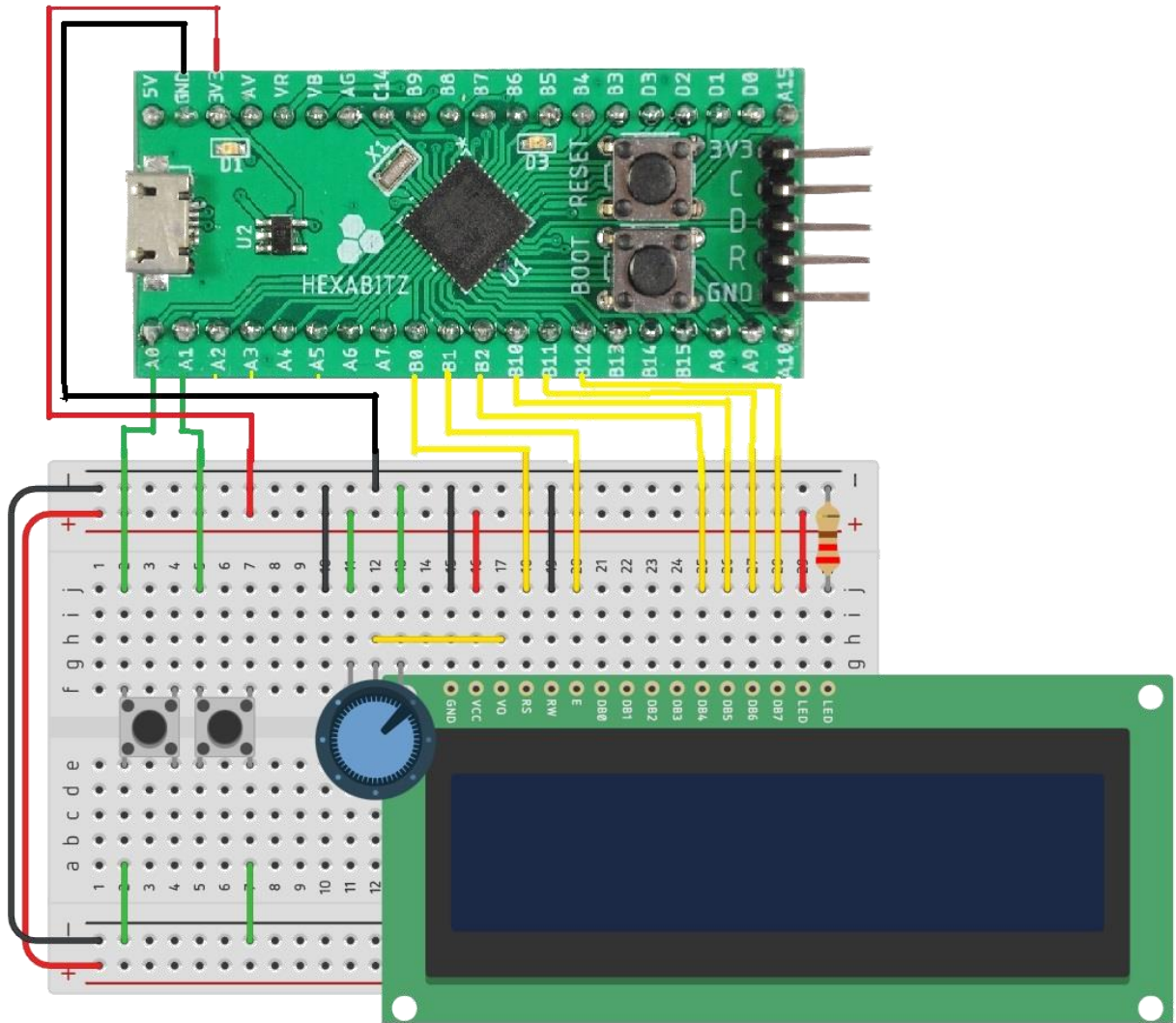
    while (1)
    {

    }
    /* USER CODE END 2 */
}

```

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

التطبيق العملي 2: إظهار عداد زوار و إظهار العدد على شاشة LCD حيث الحساس الأول يعبر عن زيادة العدد ويتم وصله على المدخل PB0 و الحساس الثاني يعبر عن إنقاص العدد ويتم وصله على المدخل PB1 ، استخدم المقاطعات الخارجية



ضبط إعدادات المشروع:

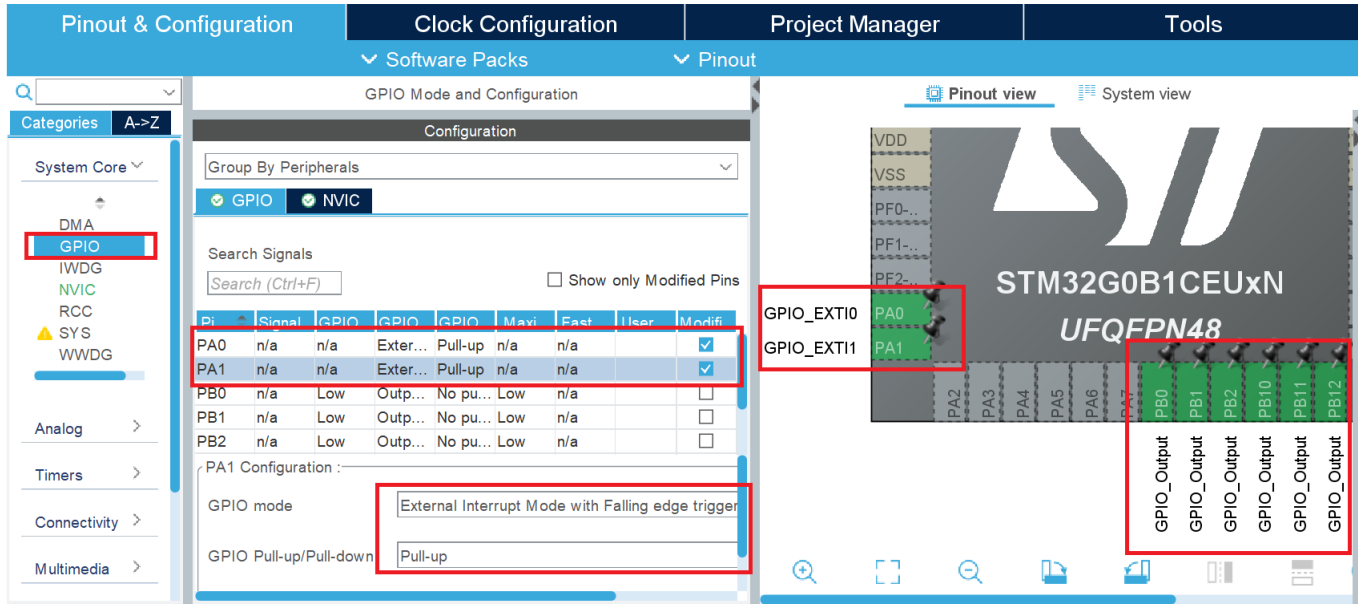
الخطوة الأولى: فتح بيئة STM32CubeIDE وإنشاء مشروع جديد ثم اختيار المتحكم

STM32G0B1CEU6N

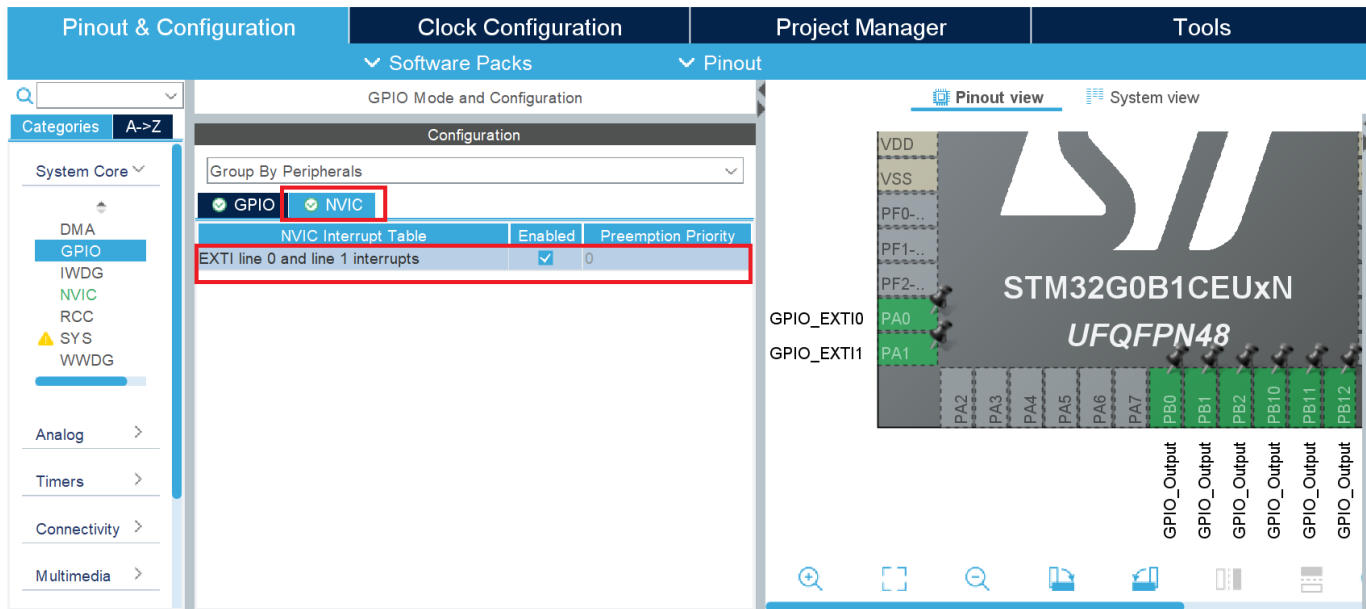
الخطوة الثانية: اختيار اسم للمشروع

الخطوة الثالثة: اختر الأقطاب PB10:PB12 والأقطاب PB0:PB2 لضبطها كأقطاب خرج، والأقطاب

PA0:PA1 كأقطاب مقاطعة خارجية



الخطوة الرابعة: تفعيل المقاطعة الخارجية على الأقطاب PA0:PA1



الخطوة الخامسة: ضبط تردد الساعة للمتحكم على التردد 16MHZ

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

```

s4_ex2.ioc  main.c  *lcd_txt.h ×
1 #ifndef      LCDTXT_H
2 #define      __LCDTXT_H
3
4 #include "stm32G0xx_hal.h"
5 /*----- Define LCD Use -----
6 /*Note: Comment which not use */
7 #define LCD16xN //For lcd16x2 or lcd16x4
8 // #define LCD20xN //For lcd20x4
9 /*----- Define For Connection -----
10 #define RS_PORT      GPIOB
11 #define RS_PIN        GPIO_PIN_0
12 #define EN_PORT      GPIOB
13 #define EN_PIN        GPIO_PIN_1
14 #define D7_PORT      GPIOB
15 #define D7_PIN        GPIO_PIN_12
16 #define D6_PORT      GPIOB
17 #define D6_PIN        GPIO_PIN_11
18 #define D5_PORT      GPIOB
19 #define D5_PIN        GPIO_PIN_10
20 #define D4_PORT      GPIOB
21 #define D4_PIN        GPIO_PIN_2

```

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

الكود النهائي:

```

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

int8_t i=0;
char buffer[16] ;
int8_t* str ;

void HAL_GPIO_EXTI_Falling_Callback(uint16_t GPIO_Pin)
{
    if (GPIO_Pin ==GPIO_PIN_0)

```

```

        {
            i++;
        }
    else if(GPIO_Pin ==GPIO_PIN_1)
    {
        if(i>0)
            i--;
    }
}

int main(void)
{
    HAL_Init();
    SystemClock_Config();
    MX_GPIO_Init();
    lcd_init();
    lcd_clear();

    while (1)
    {
        sprintf(buffer,"people= %d", i);
        str = (int8_t*)buffer;
        lcd_puts(0,0,str);
        HAL_Delay(100);
        lcd_clear();
    }
}

```

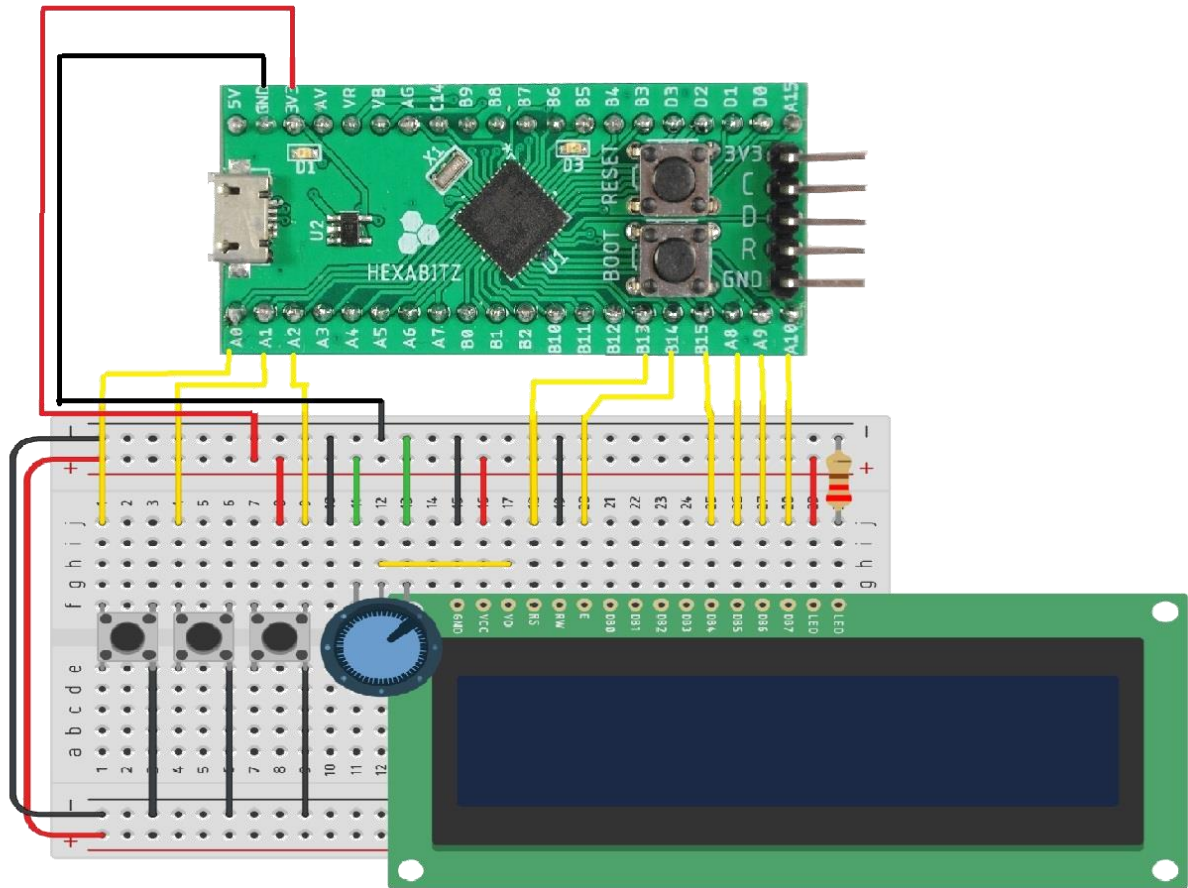
ملاحظة:

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 و وفق الشكل التالي.



ضبط إعدادات المشروع:

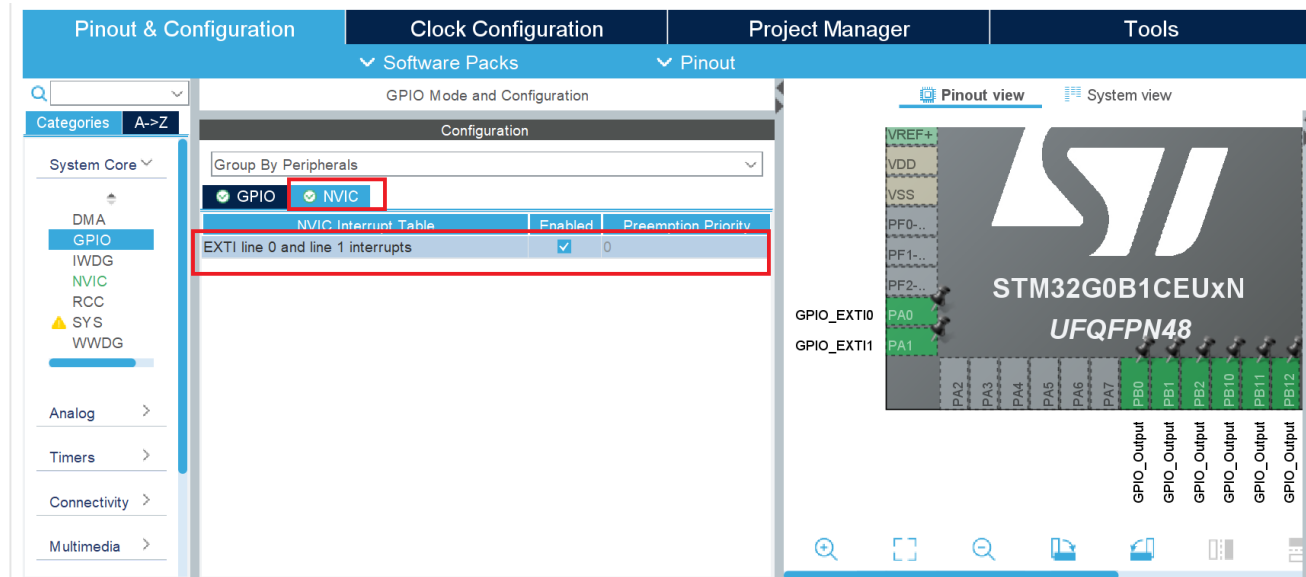
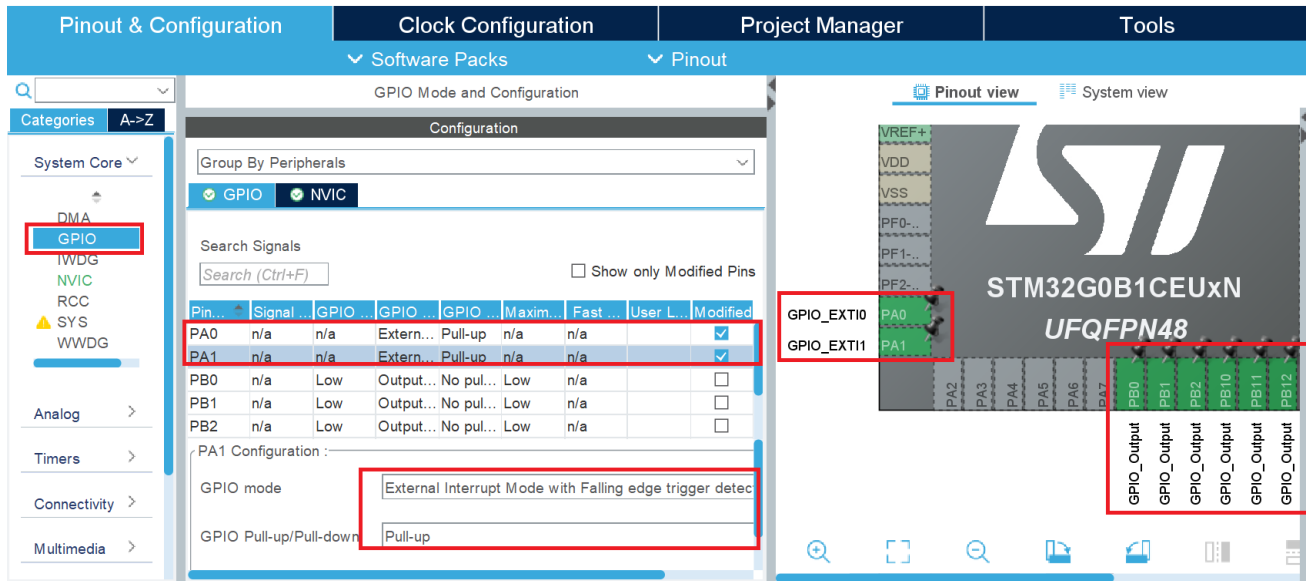
الخطوة الأولى: فتح بيئة STM32CubeIDE وإنشاء مشروع جديد ثم اختيار المتحكم

STM32G0B1CEU6N

الخطوة الثانية: اختيار اسم للمشروع

الخطوة الثالثة: اختر الأقطاب PA8:PA10، PB13:PB15 لضبطها كأقطاب خرج، والأقطاب

PA0:PA2 لضبطها كأقطاب دخل مع تفعيل مقاومة الرفع الداخلية



الخطوة الرابعة: ضبط تردد الساعة للمتحكم على التردد 16MHZ

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);
int8_t* str1 = (int8_t*)"The level is 0%";
int8_t* str2 = (int8_t*)"The level is 30%";
int8_t* str3 = (int8_t*)"The level is 60%";
int8_t* str4 = (int8_t*)"The level is 100%";
int8_t* str5 = (int8_t*)"Error";
int main(void)
{
    HAL_Init();
    SystemClock_Config();
    MX_GPIO_Init();
    /* USER CODE BEGIN 2 */
    lcd_init();

    while (1)
    {

        if      ( (HAL_GPIO_ReadPin(GPIOA, GPIO_PIN_0)==1) &&
                  (HAL_GPIO_ReadPin(GPIOA, GPIO_PIN_1)==1) &&
                  (HAL_GPIO_ReadPin(GPIOA, GPIO_PIN_2)==1)
        )
        {
            lcd_puts(0,0,str1);//Displaying on Lcd
        }
        else if( (HAL_GPIO_ReadPin(GPIOA, GPIO_PIN_0)==0) &&
                  (HAL_GPIO_ReadPin(GPIOA, GPIO_PIN_1)==1) &&
                  (HAL_GPIO_ReadPin(GPIOA, GPIO_PIN_2)==1)
        )
        {
            lcd_puts(0,0,str2);//Displaying on Lcd
        }
        else if( (HAL_GPIO_ReadPin(GPIOA, GPIO_PIN_0)==0) &&
                  (HAL_GPIO_ReadPin(GPIOA, GPIO_PIN_1)==0) &&
                  (HAL_GPIO_ReadPin(GPIOA, GPIO_PIN_2)==1)
        )
        {
            lcd_puts(0,0,str3);//Displaying on Lcd
        }
        else if( (HAL_GPIO_ReadPin(GPIOA, GPIO_PIN_0)==0) &&
                  (HAL_GPIO_ReadPin(GPIOA, GPIO_PIN_1)==0) &&

```

```
(HAL_GPIO_ReadPin(GPIOA, GPIO_PIN_2)==0)
)
{
    lcd_puts(0,0,str4); //Displaying on Lcd
}
else
    lcd_puts(0,0,str5); //Displaying on Lcd

    HAL_Delay(50);
    lcd_clear();
}
}
```