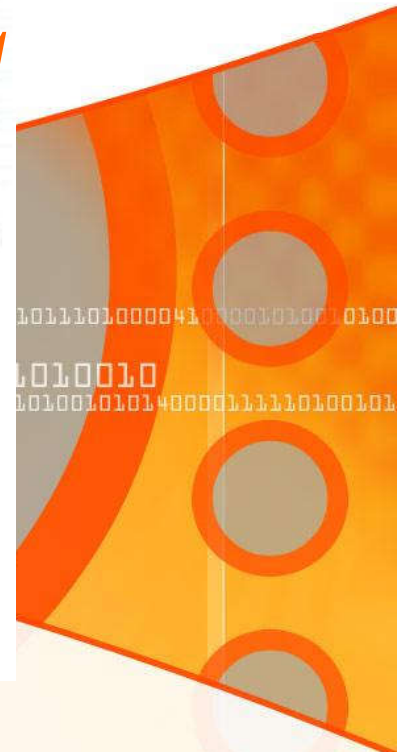


متحکّمات

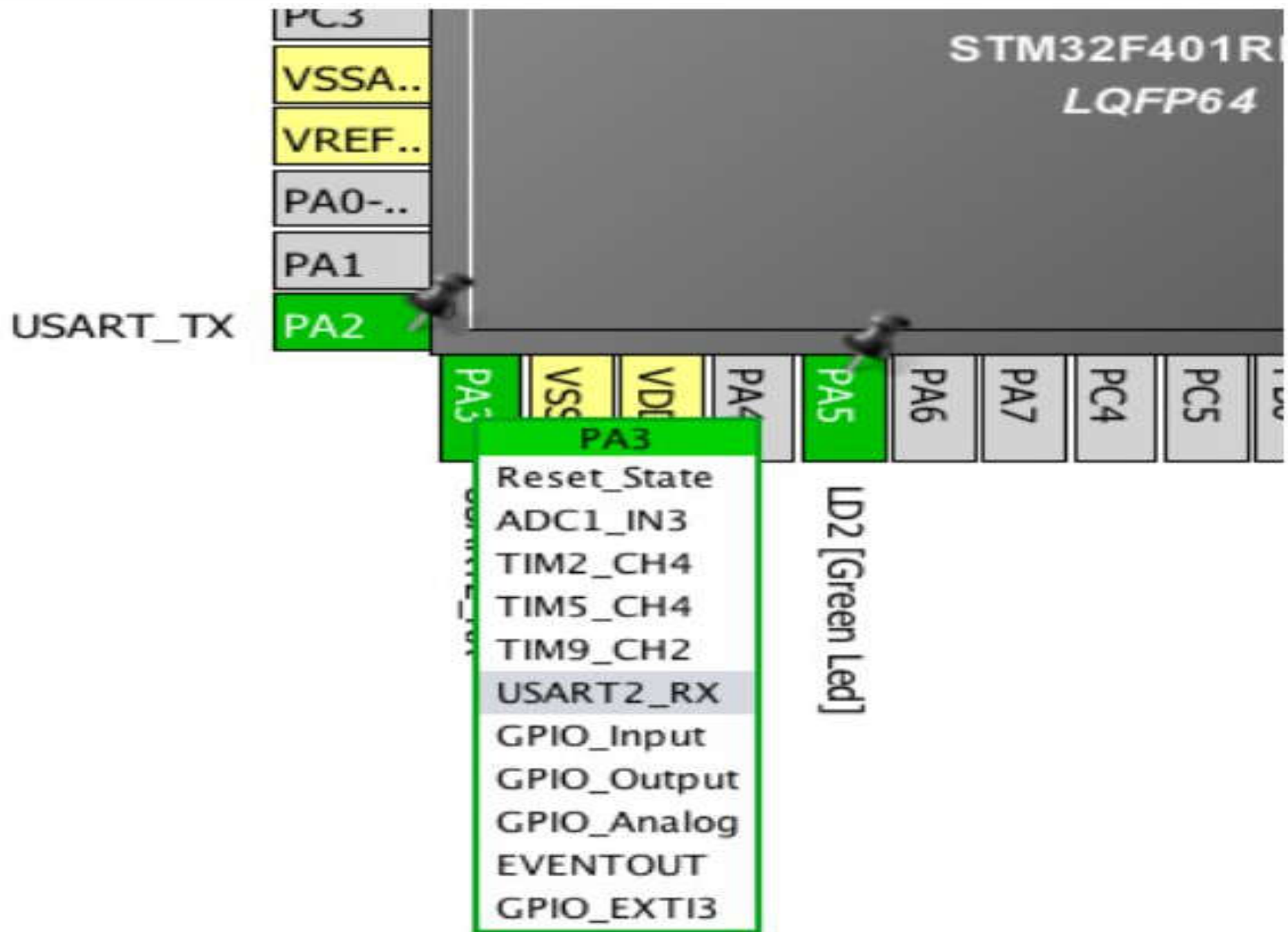
STM32

2



- ☐ أنماط عمل أقطاب المتحكم GPIO Mode
- ☐ برمجة أقطاب الخرج في متحكمات stm32
- ☐ التوابع المستخدمة من مكتبة HAL للتحكم بالمخارج الرقمية في متحكم STM32.
- ☐ بناء أول تطبيق لإضاءة ليد باستخدام متحكمات stm32 ومكتبة HAL
- ☐ برمجة أقطاب الدخل في متحكمات stm32
- ☐ التوابع المستخدمة من مكتبة HAL للتحكم بالمداخل الرقمية في متحكم STM32.
- ☐ بناء تطبيق لإضاءة ليد من خلال مفتاح لحظي باستخدام متحكمات stm32 ومكتبة HAL

1. أنماط عمل أقطاب المتحكم GPIO Mode



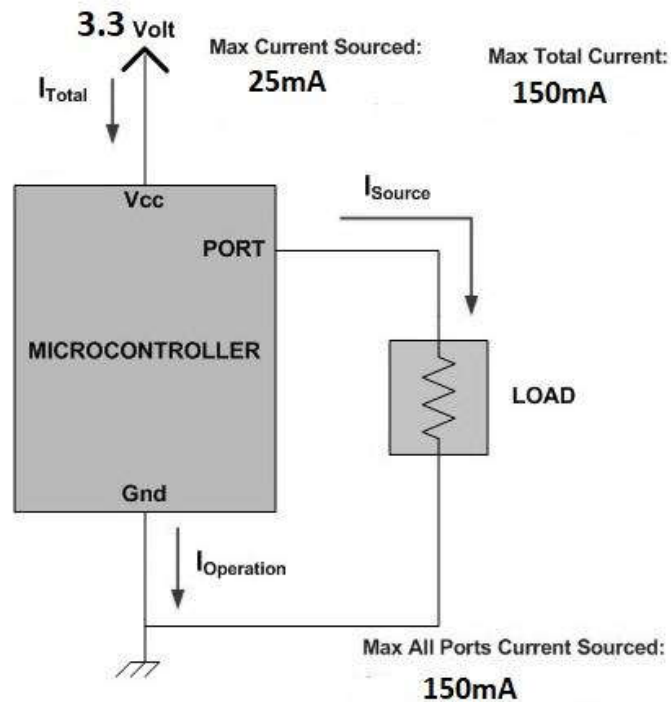
A

ربط الأحمال مع مخارج المتحكم

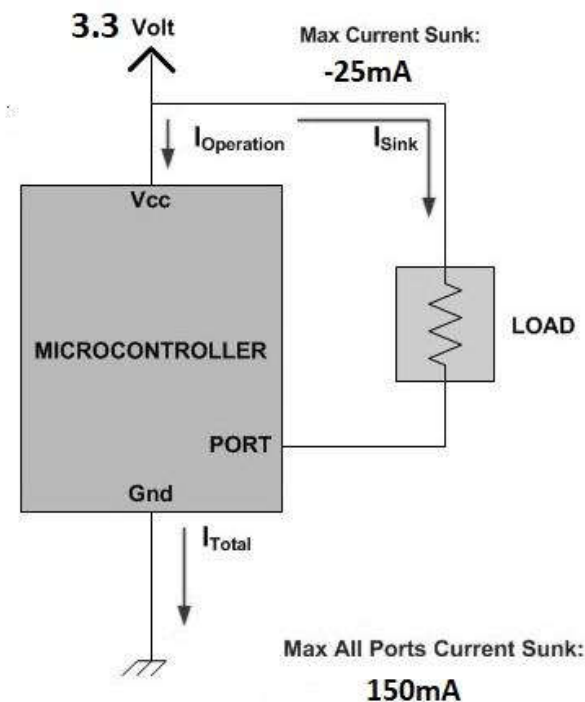
إن وصل الأحمال مع أقطاب المتحكم يكون بطريقتين:

A. يعمل القطب كمنبع لتيار تشغيل الحمل (Source).

B. يعمل القطب كمصرف لتيار تشغيل الحمل (Sink).

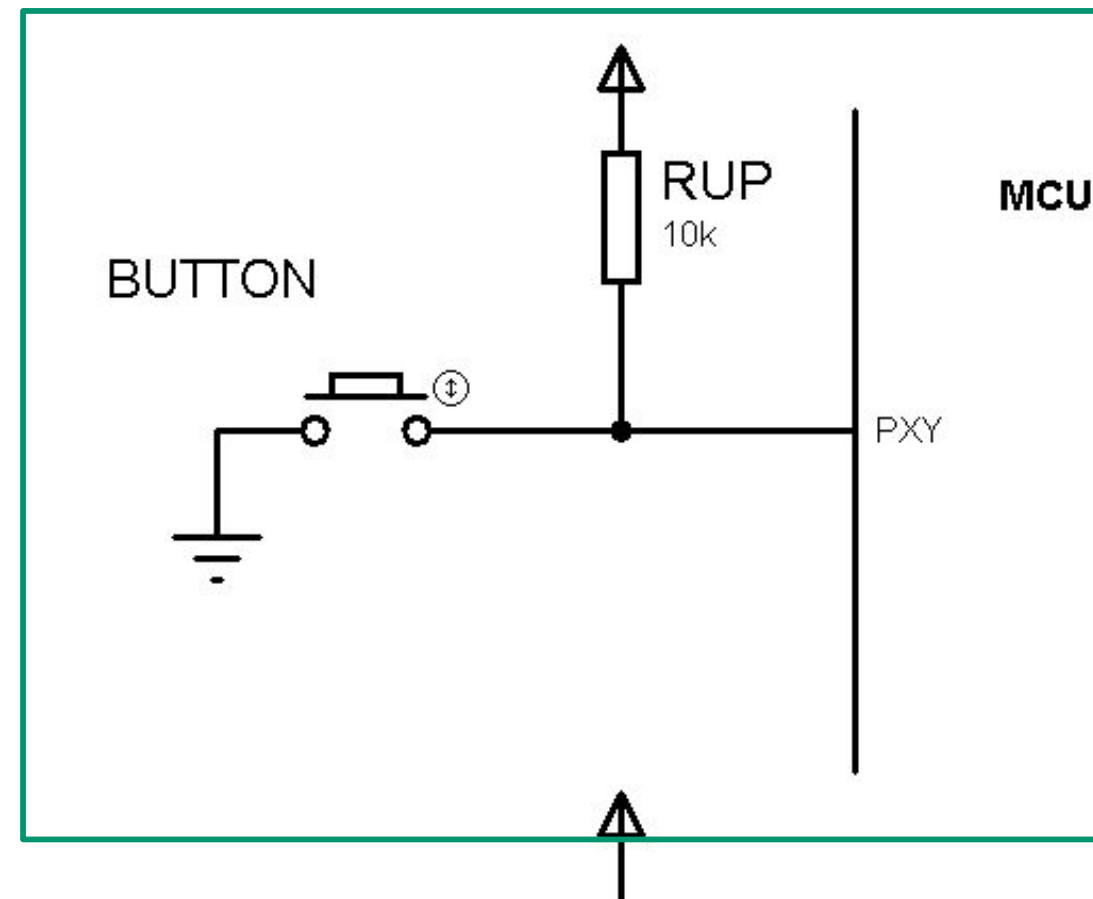
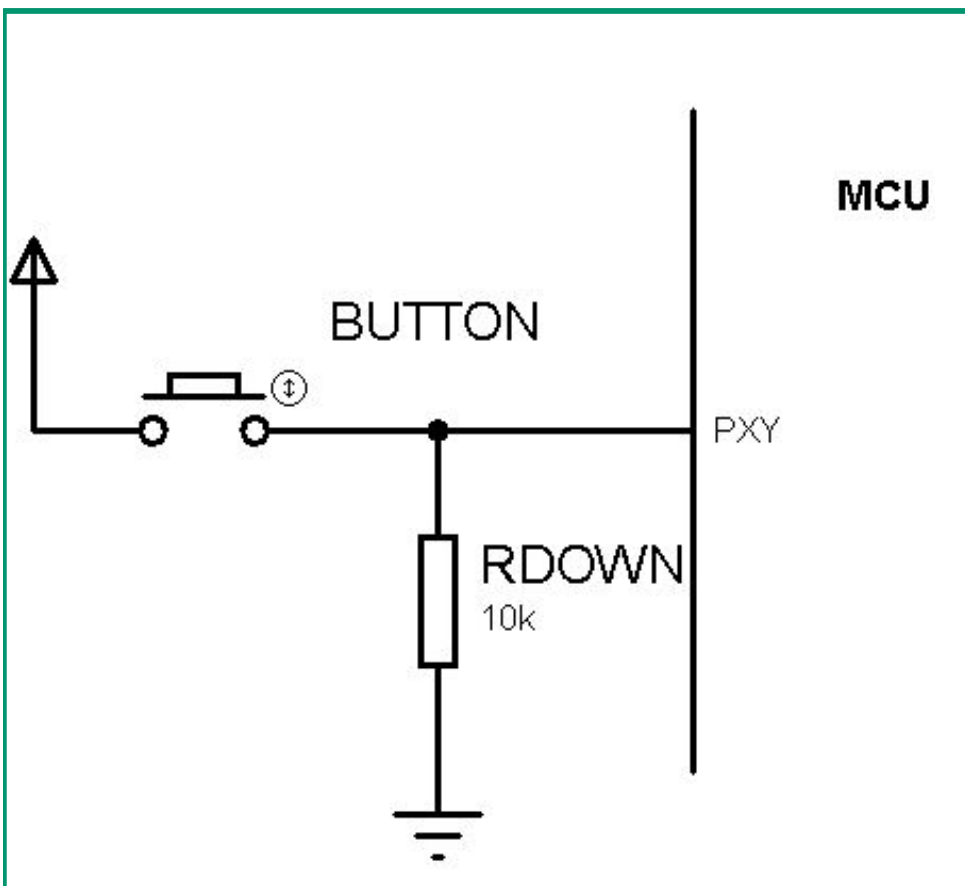


A. Microcontroller's Port is used as a Current Source

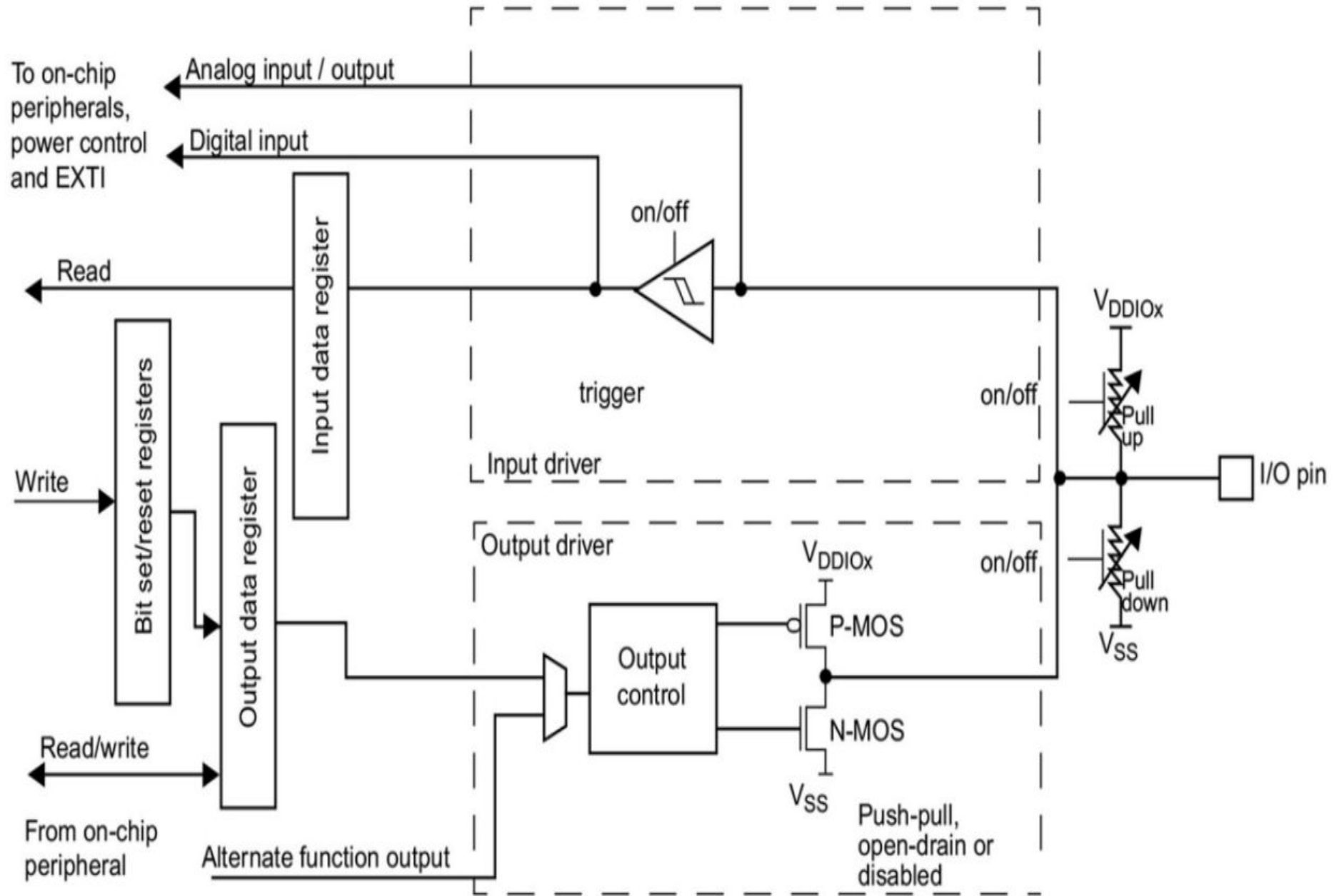


B. Microcontroller's Port is used to Sink Current

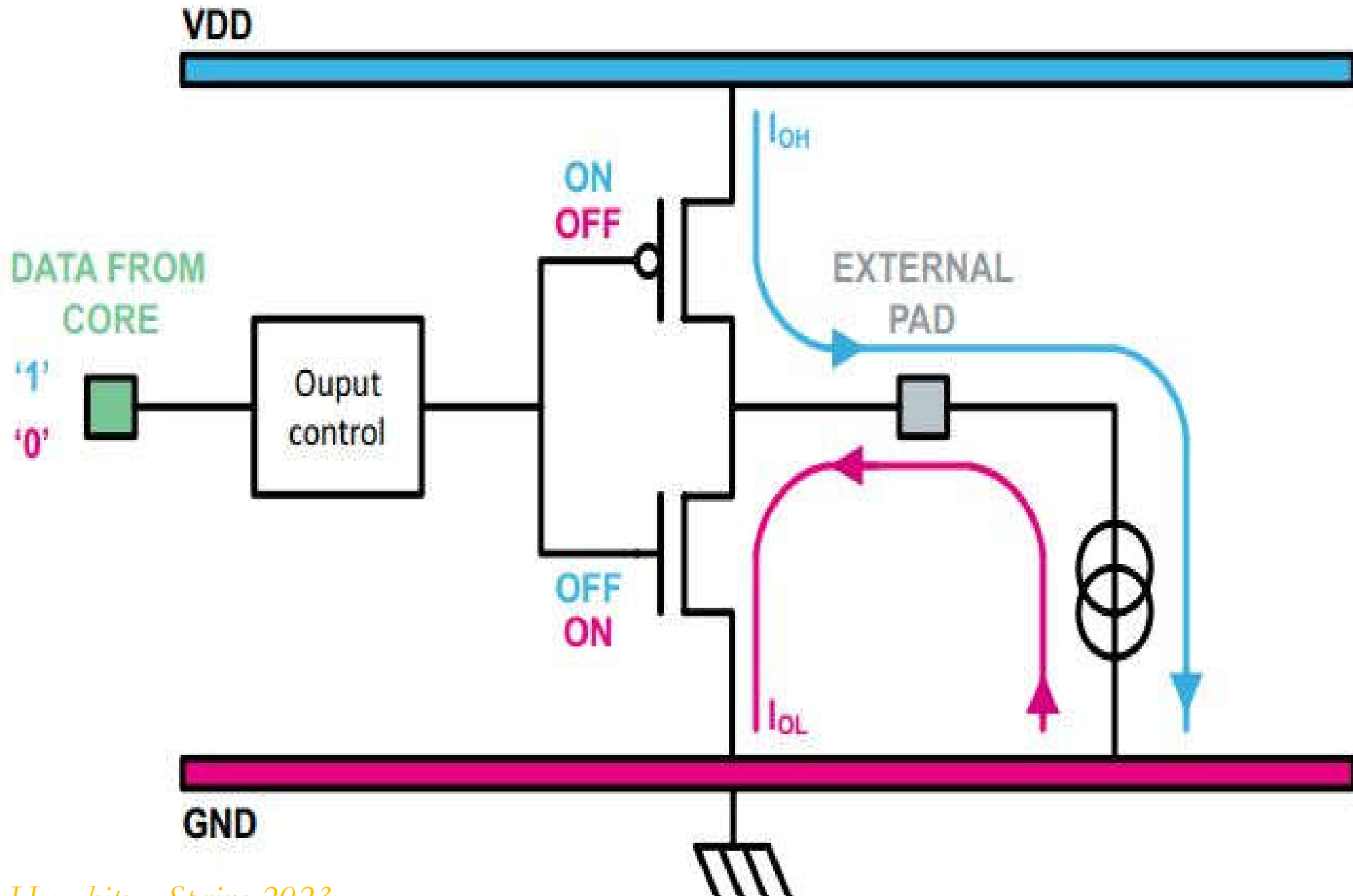
مفهوم مقاومة الرفع PULL_UP RES ومقاومة الخفض PULL_DOWN_RES



1. أنماط عمل أقطاب المتحكم GPIO Mode



2. برمجة أقطاب الخرج في متحكمات stm32



2. برمجة أقطاب الخرج في متحكمات stm32

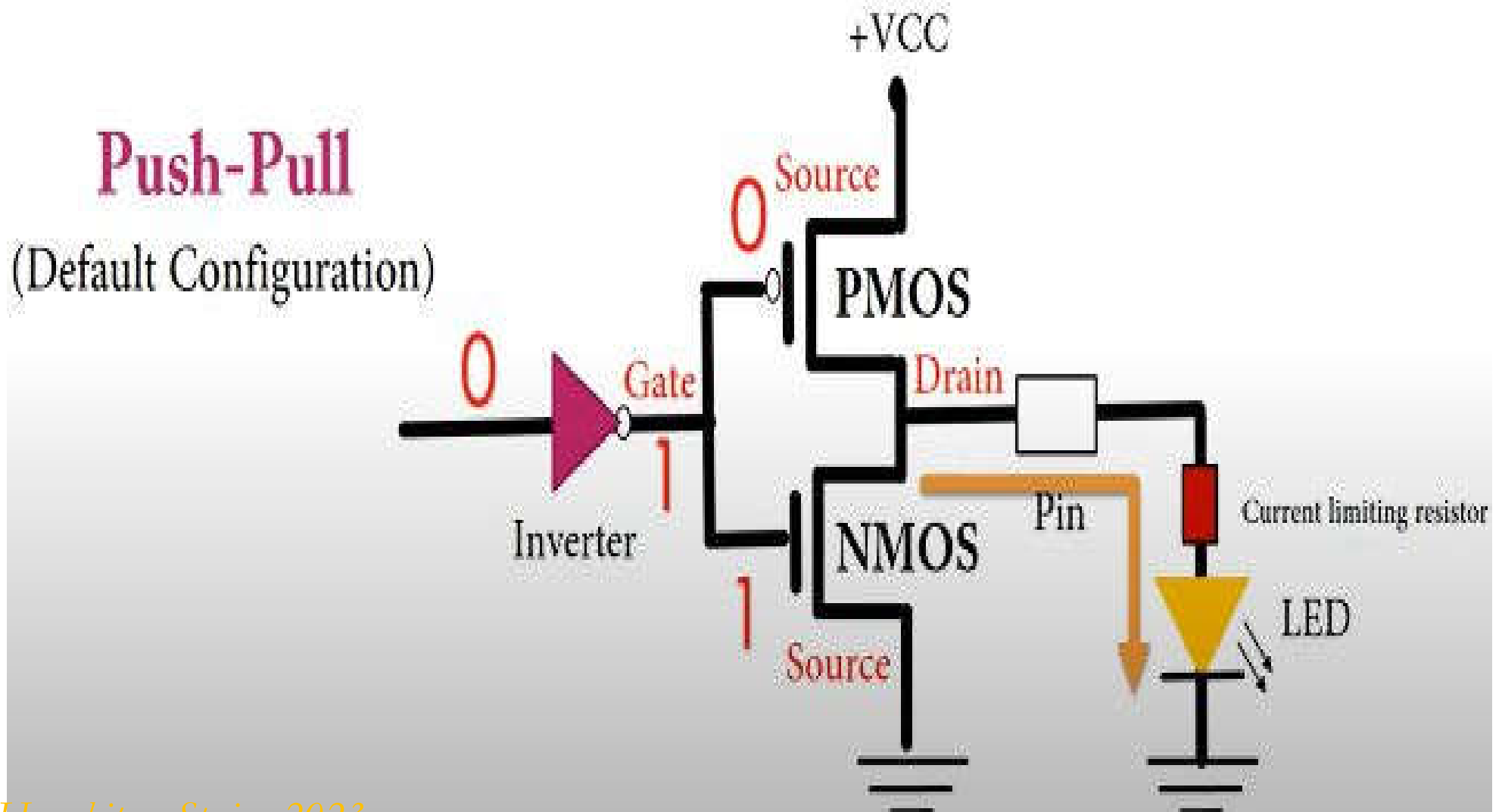
نميز نمطي عمل لكل قطب خرج من أقطاب المتحكم GPIO:

□ **نمط PUSH-PULL:** يفترض تم وصل مصعد ليد مع قطب الإخراج للمتحكم المصغر، فعند تطبيق واحد منطقي على القطب يصبح الترانزستور PMOS بحالة تشغيل on وبالتالي يتم تطبيق جهد الـ VCC على مصعد الليد ويضيء الليد، أما عند تطبيق صفر منطقي على القطب يصبح الترانزستور NMOS بحالة تشغيل on وبالتالي يتم تطبيق جهد الأرضي GND على مصعد الليد ويطفأ الليد كما هو موضح بالشكل التالي:

2. برمجة أقطاب الخرج في متحكمات stm32

نميز نمطي عمل لكل قطب خرج من أقطاب المتحكم GPIO:

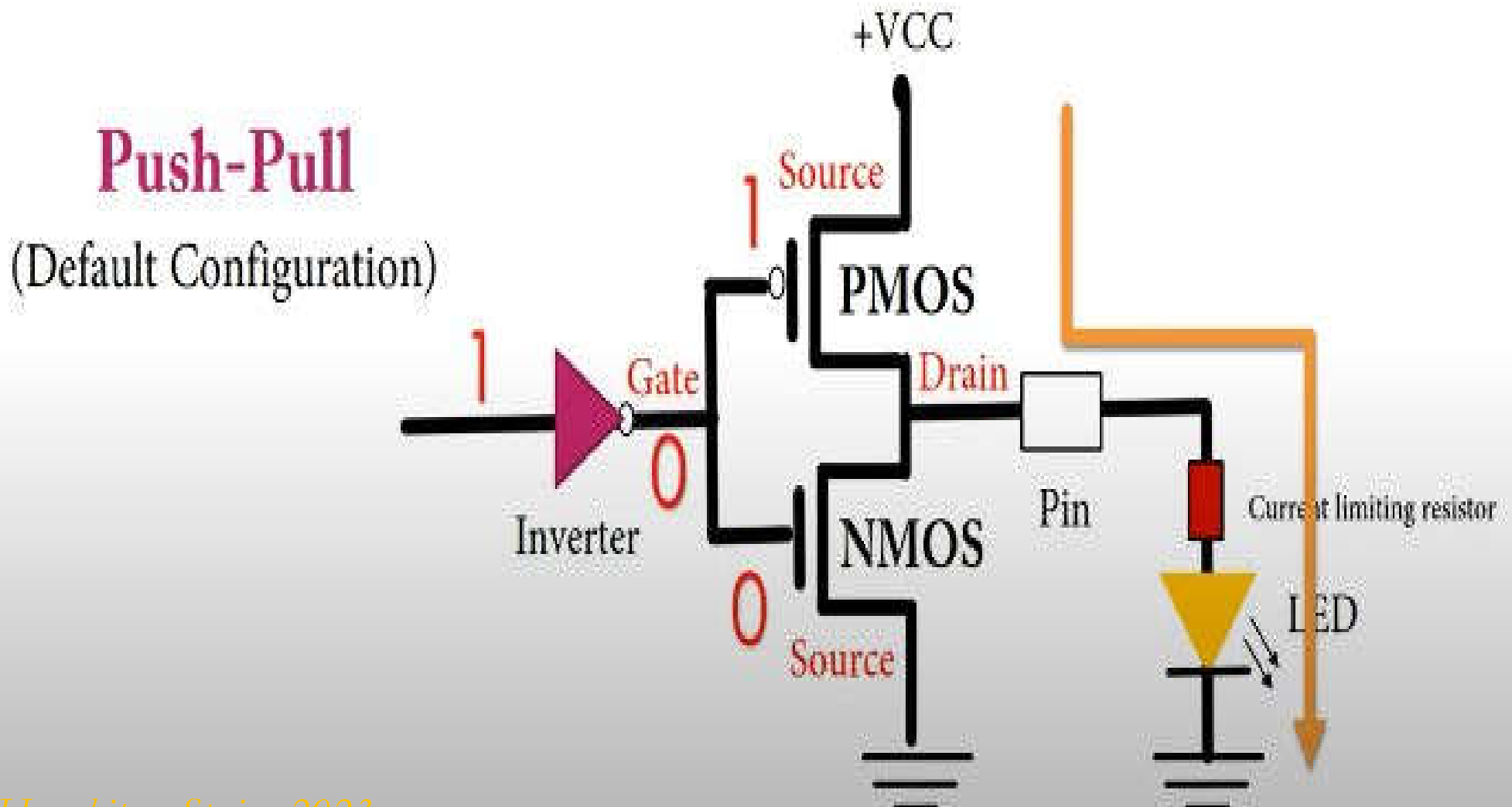
نمط **PUSH-PULL** ☐



2. برمجة أقطاب الخرج في متحكمات stm32

نميز نمطي عمل لكل قطب خرج من أقطاب المتحكم GPIO:

نمط **PUSH-PULL** □

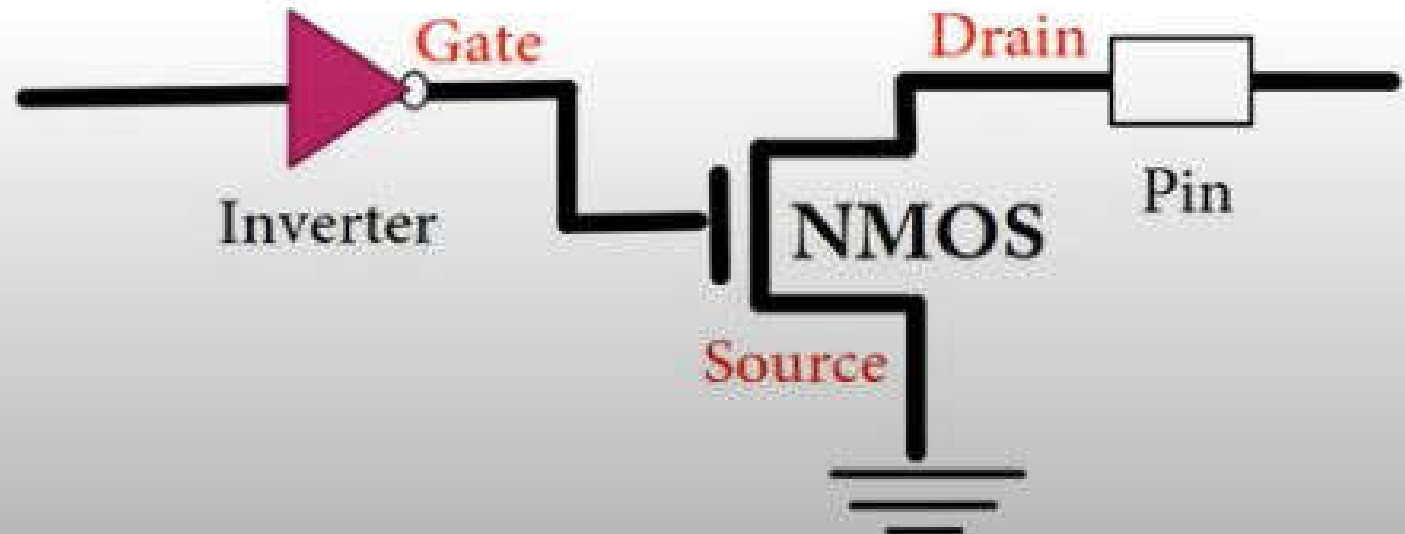


2. برمجة أقطاب الخرج في متحكمات stm32

نميز نمطي عمل لكل قطب خرج من أقطاب المتحكم GPIO:

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

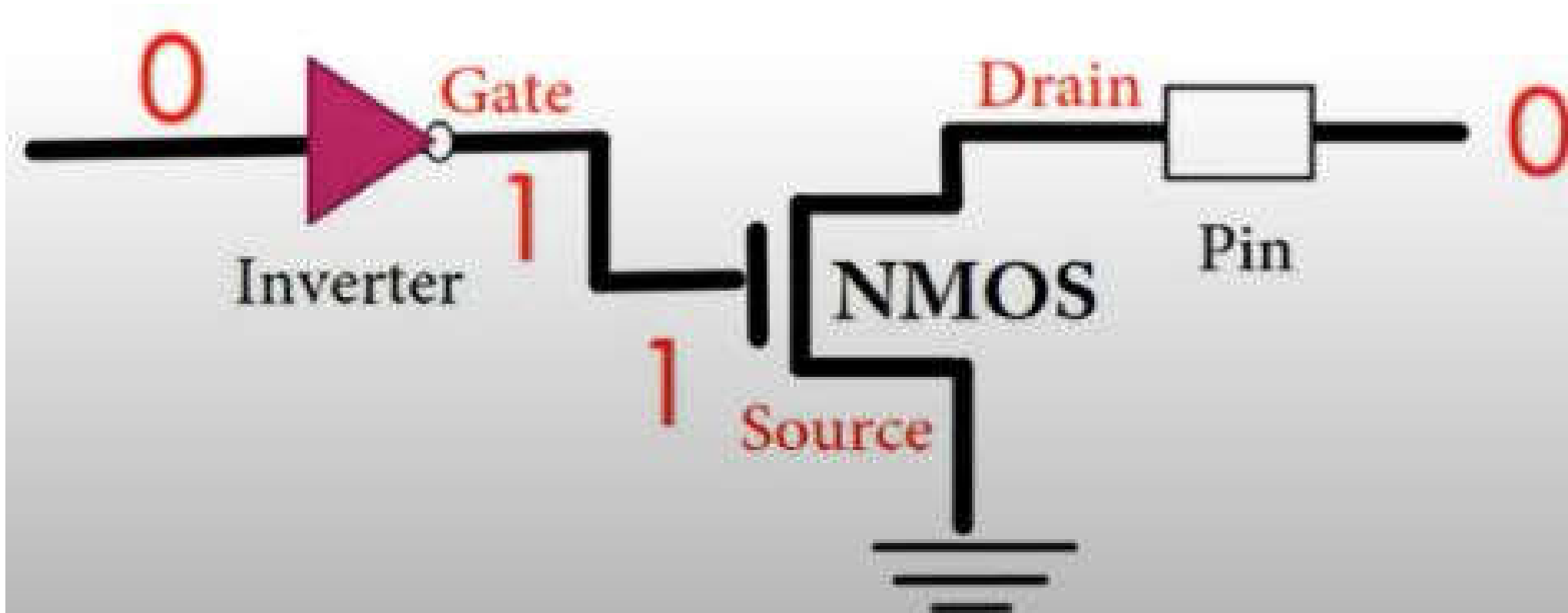
Open Drain



2. برمجة أقطاب الخرج في متحكمات stm32

نميز نمطي عمل لكل قطب خرج من أقطاب المتحكم GPIO:

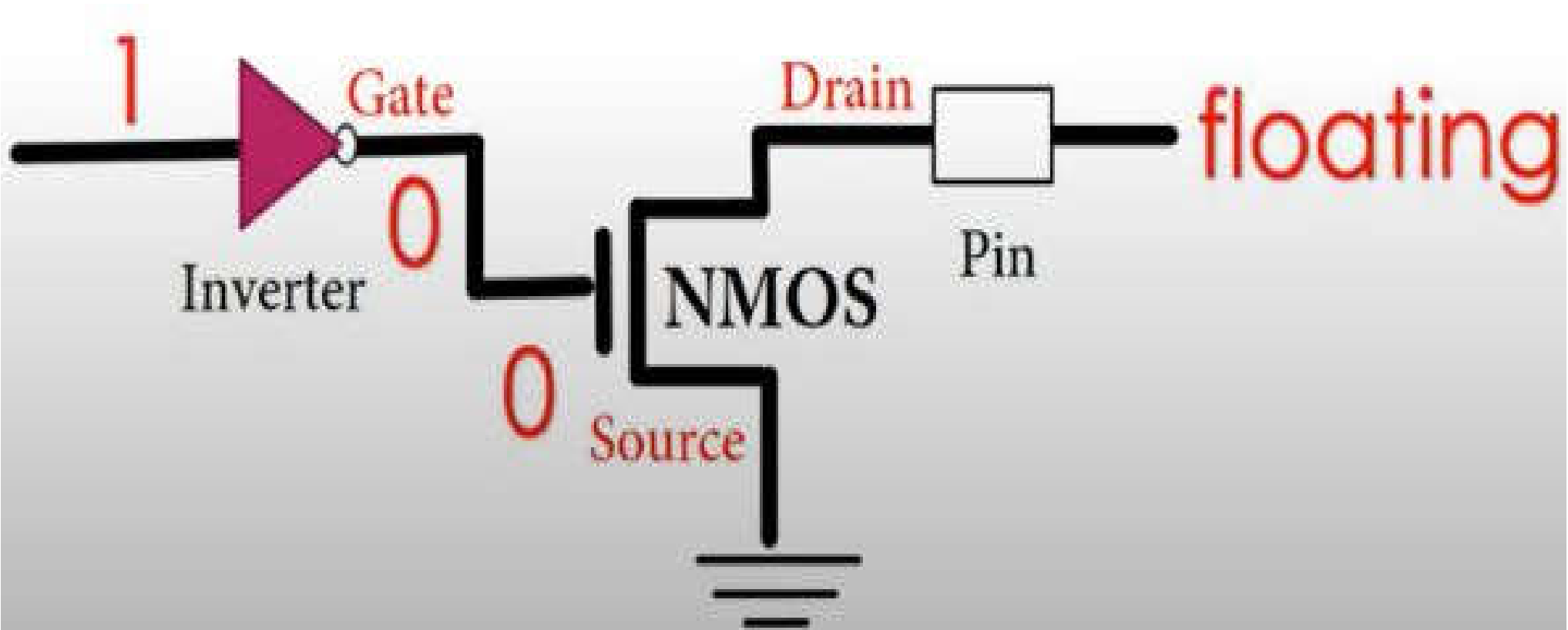
□ **نمط Open-Drain** في هذا النمط يمكن للمتحكم أن يعمل كمصرف للتيار sink فقط، فيتم قيادة قطب الخرج من خلال ترانزستور واحد من نوع NMOS كما هو موضح في الأشكال التالية:



2. برمجة أقطاب الخرج في متحكمات stm32

نميز نمطي عمل لكل قطب خرج من أقطاب المتحكم GPIO:

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



2. برمجة أقطاب الخرج في متحكمات stm32

نميز نمطي عمل لكل قطب خرج من أقطاب المتحكم GPIO:

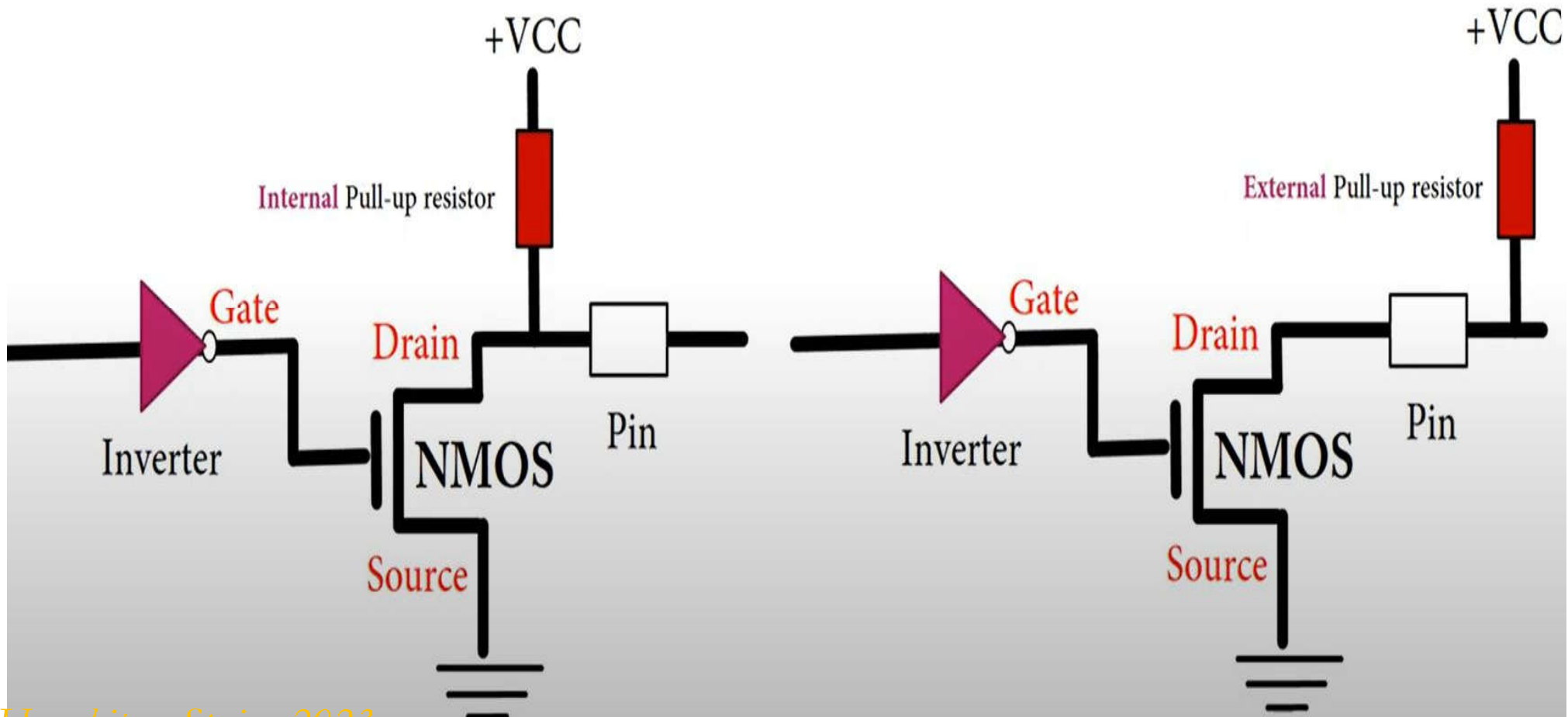
□ **نمط Open-Drain** نلاحظ من الأشكال السابقة أنه عند تطبيق صفر منطقي على قطب المتحكم يصبح الترانزستور بحالة توصيل on وبالتالي يتم توصيل الحمل مع الأرضي GND ، أما في حالة تطبيق واحد منطقي على قطب المتحكم يصبح الترانزستور بحالة فصل off وبالتالي يصبح الجهد المطبق على الحمل عائم غير محدد floating لذا ولحل 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:

2. برمجة أقطاب الخرج في متحكمات stm32

نميز نمطي عمل لكل قطب خرج من أقطاب المتحكم GPIO:

نمط Open-Drain ☐

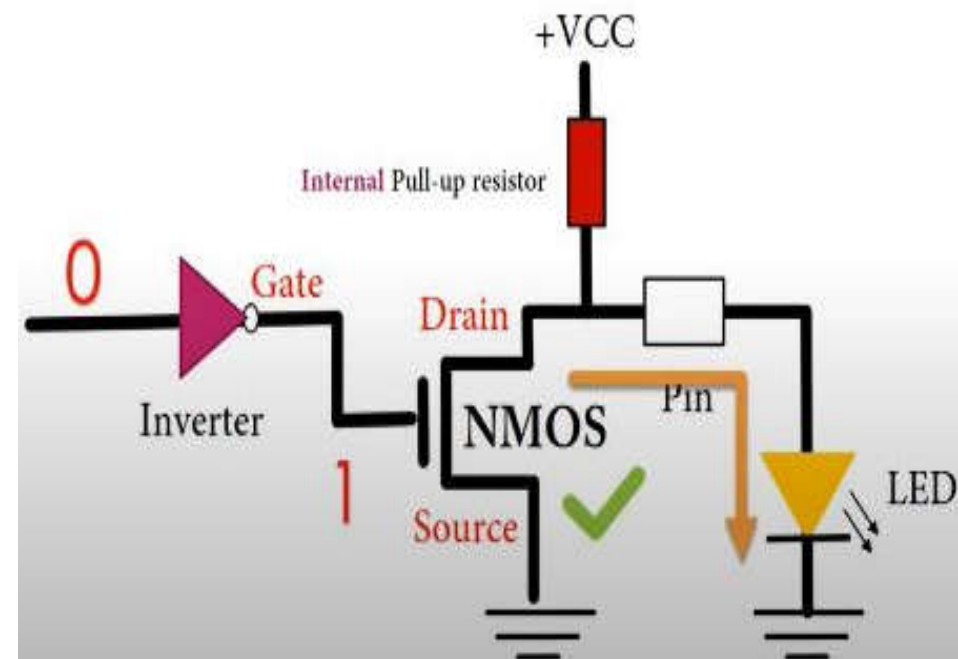
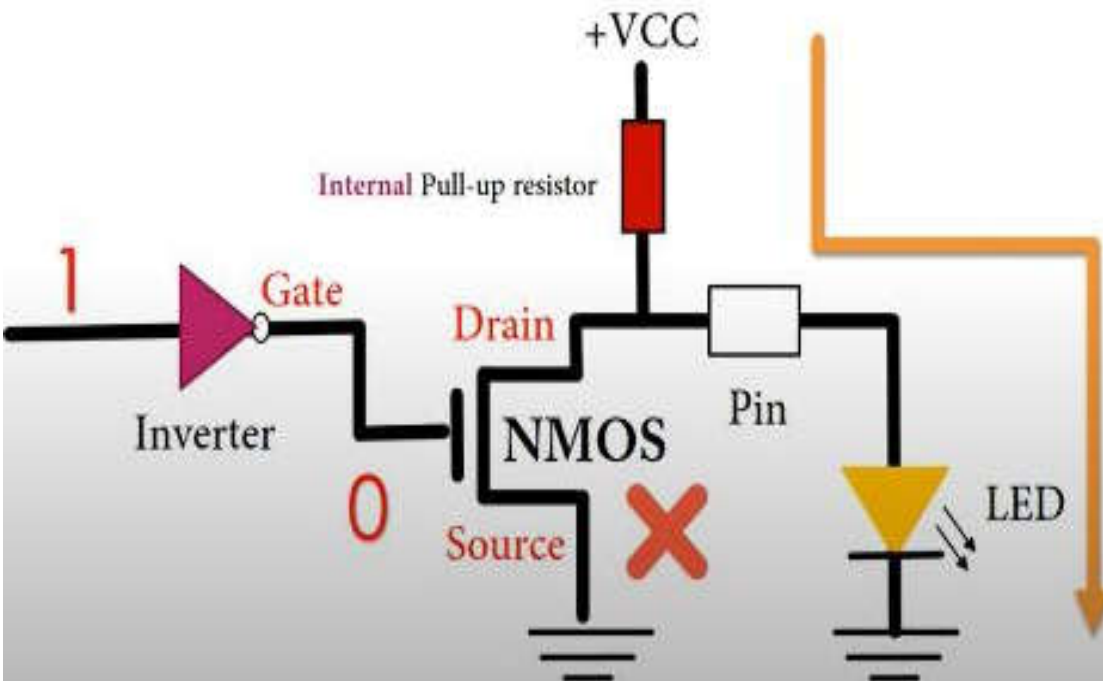
Open Drain with Pull-up Resistor -



2. برمجة أقطاب الخرج في متحكمات stm32

نميز نمطي عمل لكل قطب خرج من أقطاب المتحكم GPIO:

نمط Open-Drain ☐



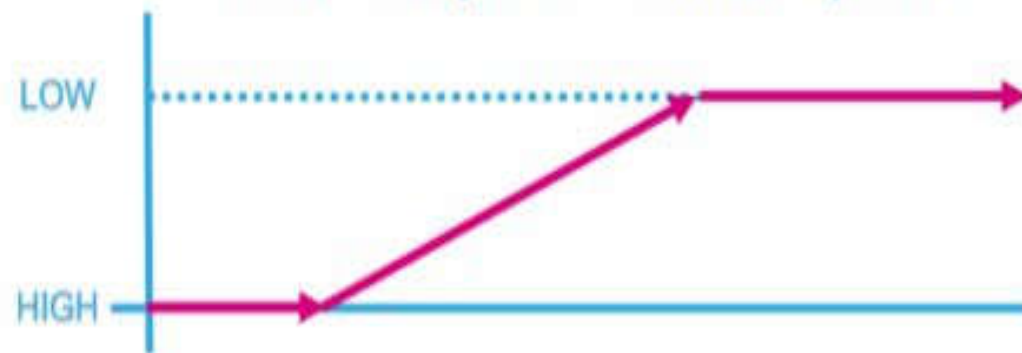
برمجة أقطاب الخرج في متحكمات stm32

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

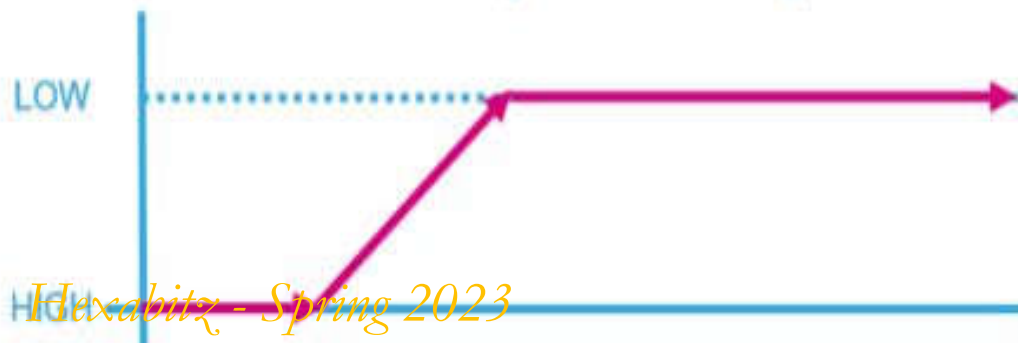
GPIO output LOW speed



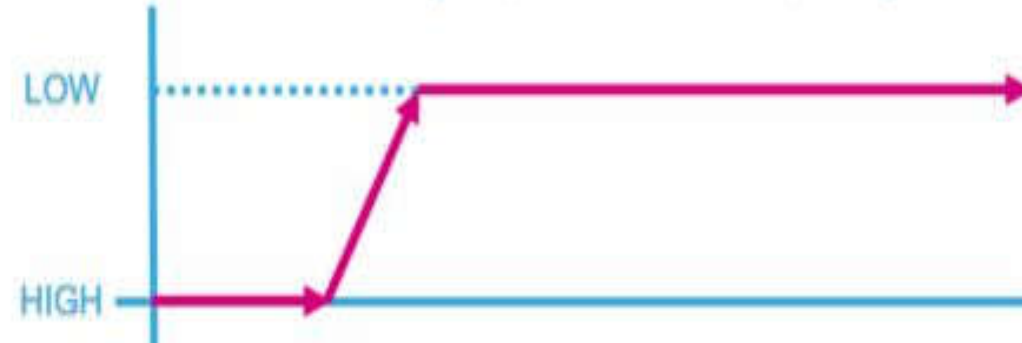
GPIO output MEDIUM speed



GPIO output HIGH speed



GPIO output VERY HIGH speed



ربط الأحمال مع مخرج المتحكم

أهم الاعتبارات التي يجب أن تؤخذ بعين الاعتبار عن ربط أقطاب المتحكم إلى الأحمال هو:

- التيار الأعظمي المستهلك من قطب المتحكم (V_{cc} to Gnd) الذي يمكن سحبه أو تصريفه عن طريق المتحكم بشكل كلي هو 150mA وفق المواصفات الكهربائية لعائلة متحكمات STM32.

- وقيمة التيار التي يمكن سحبها أو تصريفها لقطب خرج من أقطاب المتحكم تتراوح عادة من 25mA حسب المواصفات الكهربائية للمتحكم المصغر STM32.

ربط الأحمال مع مخارج المتحكم

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	

التوابع المستخدمة من مكتبة HAL للتحكم بالمخارج الرقمية في متحكم STM32:

□ لإعطاء واحد منطقي set أو صفر منطقي reset لقطب محدد من
أي منفذ من منافذ المتحكم نقوم باستخدام التابع التالي من مكتبة
HAL:

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

اسم المنفذ المراد التحكم بأحد
أقطابه على سبيل المثال
GPIOA

رقم القطب المراد التحكم به
على سبيل المثال
GPIO PIN 5

لإعطاء واحد منطقي نكتب
GPIO_PIN_SET
ولإعطاء صفر منطقي
نكتب
GPIO_PIN_RESET

التوابع المستخدمة من مكتبة HAL للتحكم بالمخارج الرقمية في متحكم STM32:

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

□ **HAL_GPIO_WritePin(GPIOD, GPIO_PIN_10, GPIO_PIN_SET);**

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

□ **HAL_GPIO_WritePin(GPIOA, GPIO_PIN_5, GPIO_PIN_RESET);**

التوابع المستخدمة من مكتبة HAL للتحكم بالمخارج الرقمية في متحكم STM32:

□ 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:

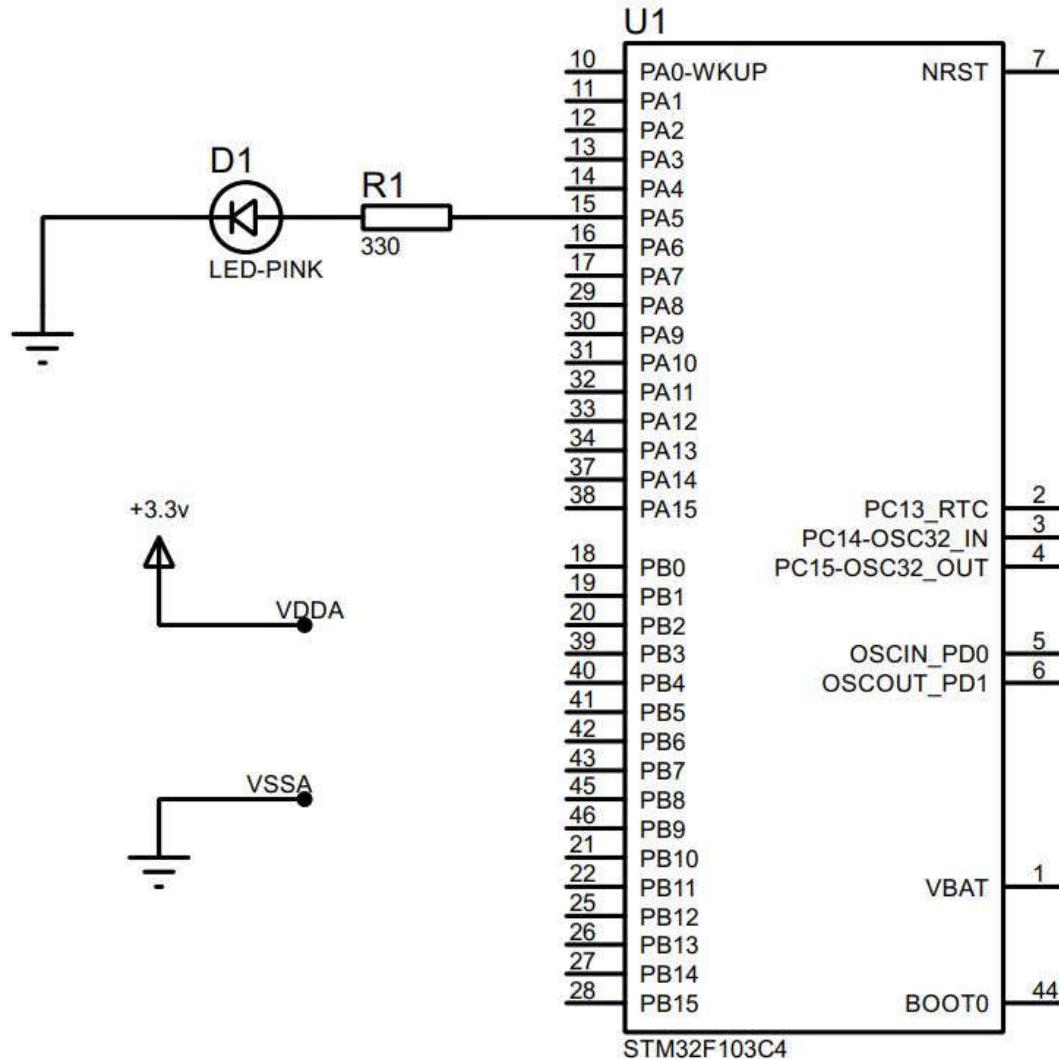
```
HAL_GPIO_TogglePin(GPIOA,GPIO_PIN_5);
```

□ لإضافة تأخير زمني بالميلي ثانية نستخدم التابع التالي:

```
HAL_Delay(Milliseconds)
```

STM32 ومكتبة HAL

سنقوم بتصميم تطبيق يقوم بعمل toggle لليد المتصل بالقطب PA5



STM32 ومكتبة HAL

نقوم بضبط إعدادات القطب PA5 كقطب خرج



The screenshot displays the STM32CubeIDE interface with the 'Pinout & Configuration' tab selected. The 'GPIO Mode and Configuration' window is open, showing the configuration for pin PA5. The pin is configured as an output with push-pull mode and no pull-up or pull-down. The maximum output speed is set to Low. The pin is labeled 'GPIO_Output' in the pinout view.

GPIO Mode and Configuration

Group By Peripherals: GPIO

Search Signals: Search (Ctrl+F) ☐ Show only Modified Pins

Pin No...	Signal on ...	GPIO out...	GPIO mode	GPIO Pull...	Maximum...	User Label	Modified
PA5	n/a	Low	Output Pu...	No pull-up...	Low		<input type="checkbox"/>

PA5 Configuration :

GPIO output level: Low

GPIO mode: Output Push Pull

GPIO Pull-up/Pull-down: No pull-up and no pull-down

Maximum output speed: Low

User Label:

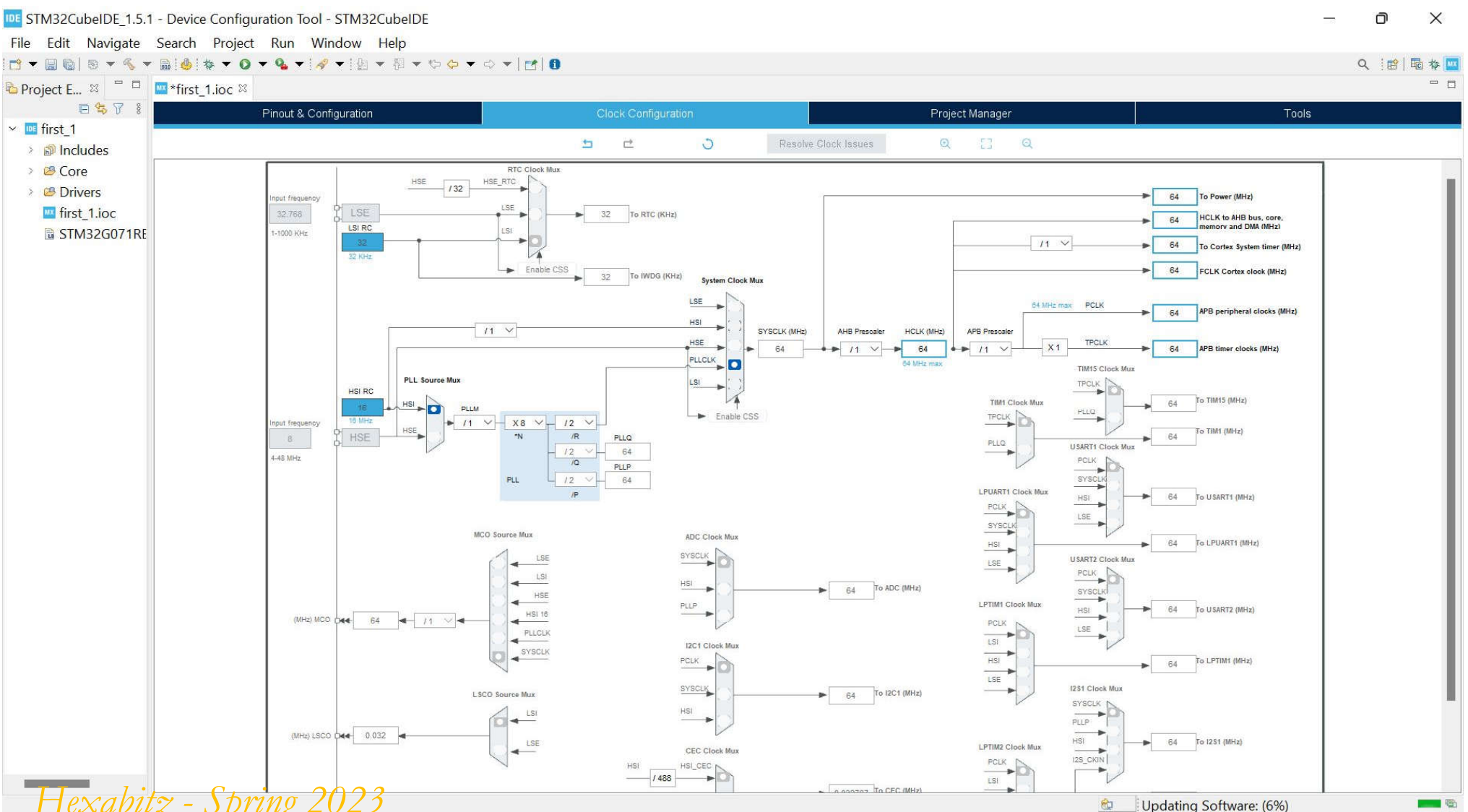
Pinout view


STM32F103C4Tx LQFP48

GPIO_Output

التطبيق الأول: إضاءة ليد وإطفاءه كل 0.5sec باستخدام متحكمات STM32 ومكتبة HAL

نقوم بضبط تردد الساعة للمتحكم



نقوم بالضغط على Ctrl+s أو من Project...Generate code  ، ليتم حفظ المشروع وتوليد الكود ثم نقوم بإضافة الجزء 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)
{
    HAL_GPIO_TogglePin(GPIOA, GPIO_PIN_5);
    HAL_Delay(500);
}

}
```

التحكم بالمخارج الرقمية للـ 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 □

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
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.

Hexabitz - Spring 2023 **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).

التحكم بالمخارج الرقمية للـ STM32 دون استخدام مكتبة HAL:

□ لكتابة واحد منطقي على القطب رقم 5 من المنفذ A نكتب:

□ `GPIOA->ODR |= 1<<5; // Set the Pin PA5`

□ و لكتابة صفر منطقي عليه نكتب:

□ `GPIOA->ODR &= ~(1<<5); // Reset 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 = 0x00; // Reset the PORTA`

التطبيق الثاني: إضاءة ليد وإطفاءه كل 0.5 sec دون استخدام مكتبة HAL

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

```
void SystemClock_Config(void);
```

```
static void MX_GPIO_Init(void);
```

```
int main(void)
```

```
{
```

```
    HAL_Init();
```

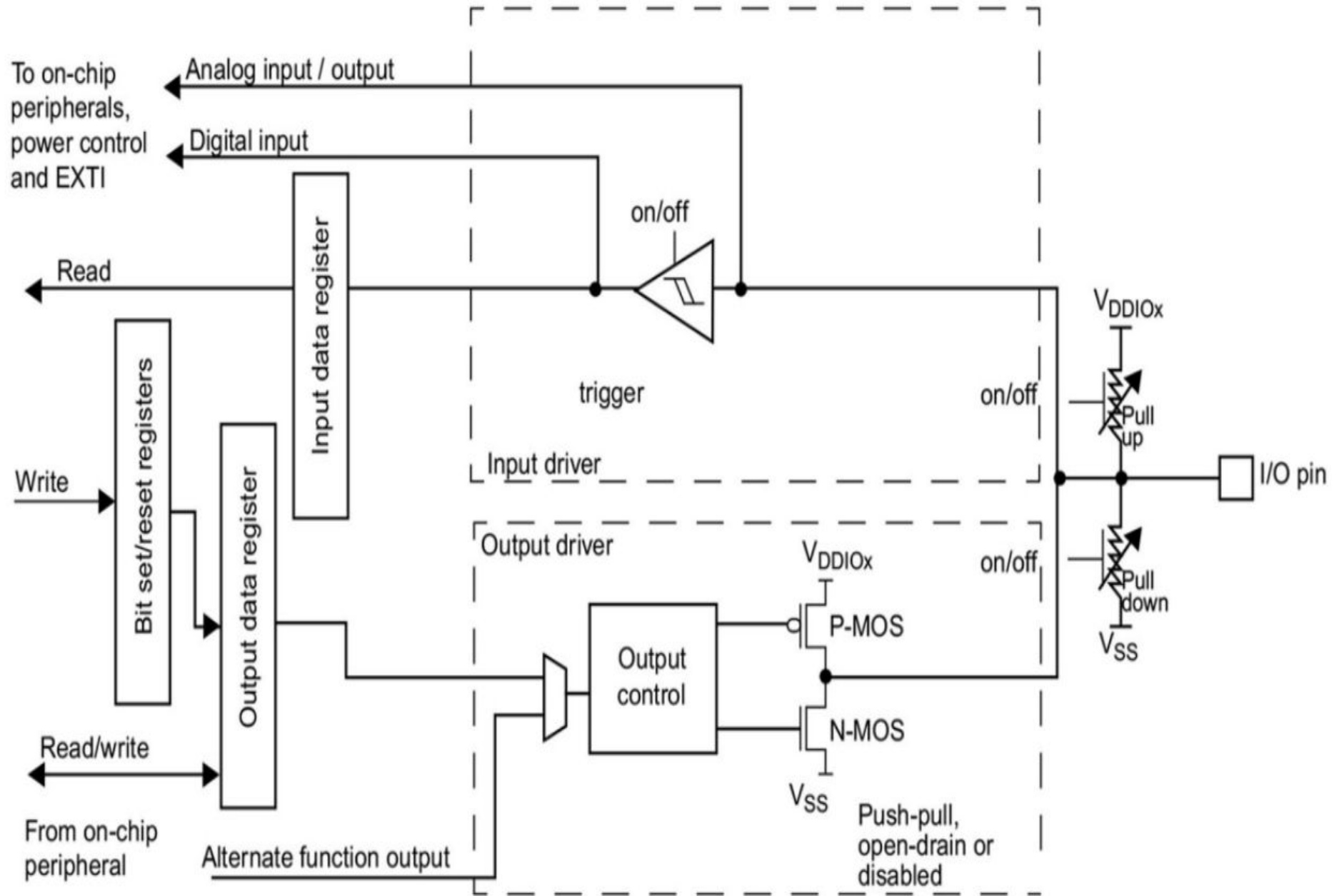
```
    SystemClock_Config();
```

```
    MX_GPIO_Init();
```

التطبيق الثاني: إضاءة ليد وإطفاءه كل 0.5 sec دون استخدام مكتبة HAL

```
while (1)
{
    GPIOA->ODR = 0x0020; // Set the Pin PA5
    HAL_Delay(500);
    GPIOA->ODR = 0x0000; // Reset the PORTA
    HAL_Delay(500);
}
}
```

برمجة أقطاب الدخل في متحكمات stm32



برمجة أقطاب الدخل في متحكمات stm32

عند استخدام قطب المتحكم كقطب دخل رقمي يتم إلغاء تفعيل Output buffer وتفعيل ال- Input buffer والذي يحتوي على قاذح schmitt الذي يحول إشارة الدخل إلى نمط TTL

أيضا يمكن تفعيل مقاومات الرفع أو الخفض لتجنب حالة Floating لقيمة الجهد على القطب

الأنماط المختلفة لقطب الدخل الرقمي

نميز ثلاث أنماط مختلفة لقطب الدخل الرقمي للمتحكم :

High impedance of Floating نمط ☐

PULL-Up نمط الـ ☐

Pull-Down نمط الـ ☐

التوابع المستخدمة من مكتبة HAL للتحكم بالمخارج 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);
```

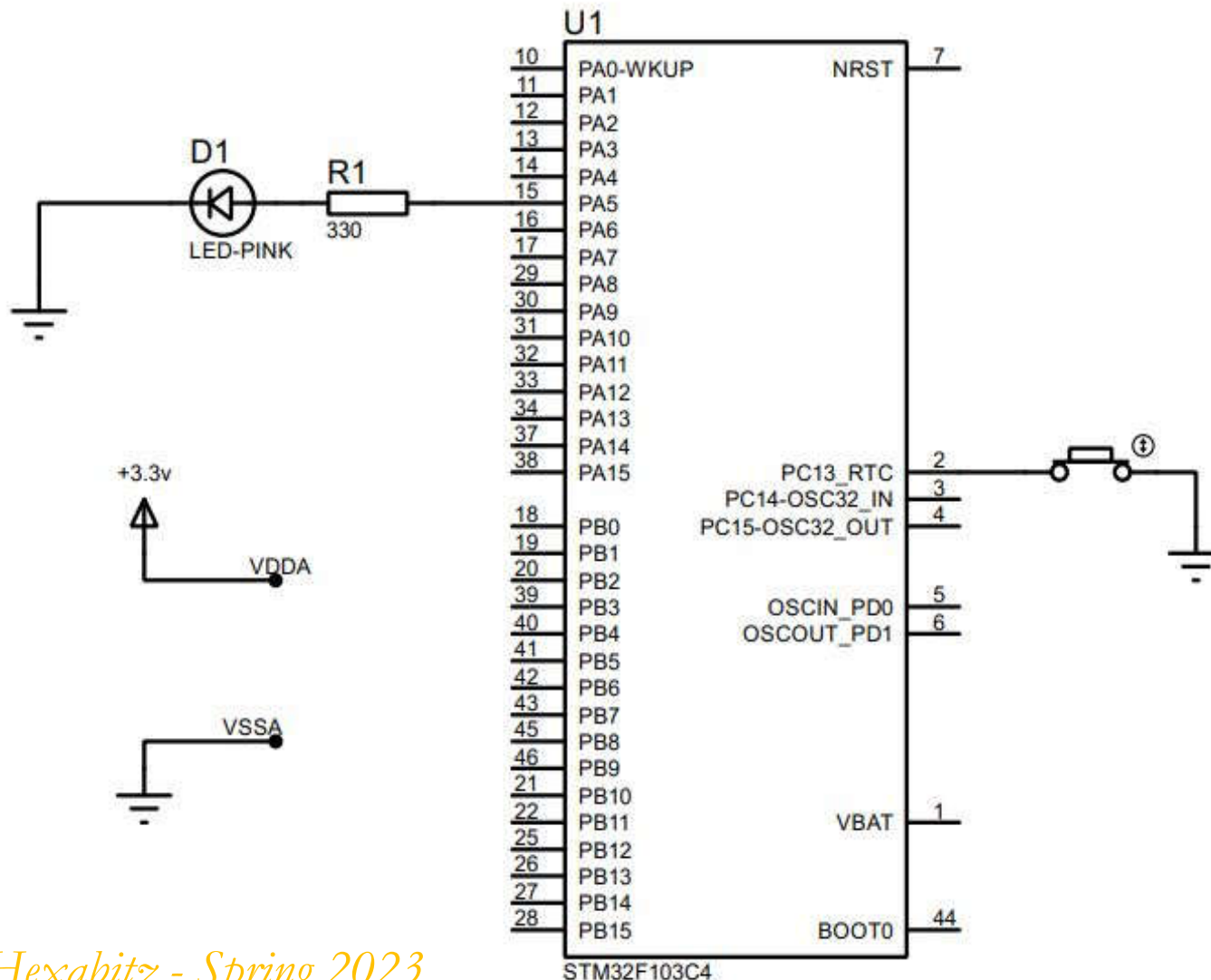
حيث يعيد هذا التابع 0 في حال كانت الحالة المنطقية للقطب في حالة جهد منخفض ، ويعيد 1 في حال كانت الحالة المنطقية للقطب في حالة جهد مرتفع.

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

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

التطبيق الأول: إضاءة ليد من خلال مفتاح لحظي باستخدام متحكمات STM32 ومكتبة HAL

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



التطبيق الأول: إضاءة ليد وإطفائه كل 0.5sec باستخدام متحكمات STM32 ومكتبة HAL

نقوم بضبط إعدادات القطب PA5 كقطب خرج والقطب PC13 كقطب دخل

The screenshot shows the STM32CubeMX Pinout & Configuration window. The left sidebar lists various system components, with GPIO selected. The main area displays the GPIO Mode and Configuration for PA5 and PC13-TAMPER-RTC. PA5 is configured as an output pin (Output Push mode, No pull-up or pull-down, Low maximum output). PC13-TAMPER-RTC is configured as an input pin (Input mode, Pull-up, n/a maximum output). The bottom section shows the configuration for PC13-TAMPER-RTC, with GPIO mode set to Input mode, GPIO Pull-up/Pull-down set to Pull-up, and User Label set to PC13-TAMPER-RTC.

Pin Name	Signal on Pin	GPIO output I...	GPIO mode	GPIO Pull-up/...	Maximum out...	User Label	Modified
PA5	n/a	Low	Output Push ...	No pull-up an...	Low		<input type="checkbox"/>
PC13-TAMPER...	n/a	n/a	Input mode	Pull-up	n/a		<input checked="" type="checkbox"/>

PC13-TAMPER-RTC Configuration :

GPIO mode: Input mode

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

User Label: PC13-TAMPER-RTC

نقوم بضبط تردد الساعة للمتحكم



التطبيق الأول: إضاءة ليد وإطفائه كل 0.5sec باستخدام متحكمات STM32 ومكتبة HAL

نقوم بالضغط على Ctrl+s أو من Project...Generate code  ، ليتم حفظ المشروع وتوليد الكود ثم نقوم بإضافة الجزء التالي:

```
#include "main.h"

void SystemClock_Config(void);
static void MX_GPIO_Init(void);
int main(void)
{
    HAL_Init();
    SystemClock_Config();
    MX_GPIO_Init();
```

التطبيق الأول: إضاءة ليد وإطفائه كل 0.5sec باستخدام متحكمات STM32 ومكتبة HAL

```
while (1){  
if(HAL_GPIO_ReadPin(GPIOC,GPIO_PIN_13)==0)  
{  
    HAL_GPIO_WritePin(GPIOA, GPIO_PIN_5,1);  
}  
else  
    HAL_GPIO_WritePin(GPIOA, GPIO_PIN_5,0);  
}}
```


للتحكم بالمداخل الرقمية للـ STM32 دون استخدام مكتبة HAL:

هناك مجموعة من المسجلات تستخدم للتحكم بالمداخل الرقمية لـ STM32 سنكتفي فقط بذكر المسجل المسؤول عن قراءة حالة المنفذ أو أحد الأقطاب الموجودة فيه 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 IDRx: 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.

التحكم بالمخارج الرقمية للـ STM32 دون استخدام مكتبة HAL:

□ 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 الموصل معه سيصبح القطب في حالة LOW، لذا يجب مراقبة حالة القطب بشكل مستمر لحين يصبح القطب في حالة LOW عندها يكون المفتاح الموصل معه مضغوط.

□ فلفحص حالة القطب الأول من المنفذ A نستخدم جملة الشرط التالية:

□ `if (!(GPIOA->IDR &(1<<1)))`

```
#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 (!(GPIOC->IDR &(1<<13)))
    {
        GPIOA->ODR = 1<<5;
    }
    else
        GPIOA->ODR &= ~(1<<5);

}}
```

Thank you for listening