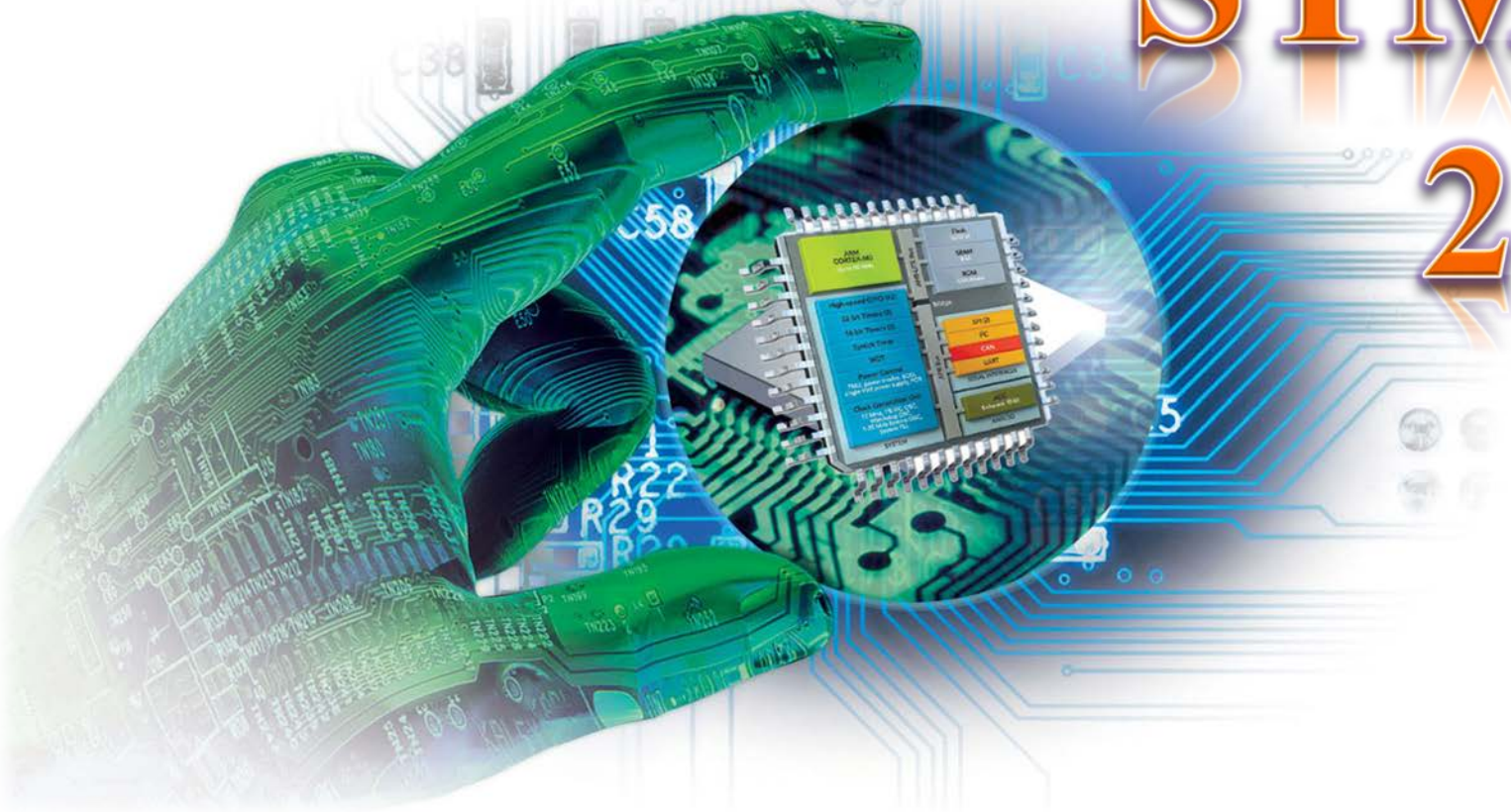
متحکمات STM32

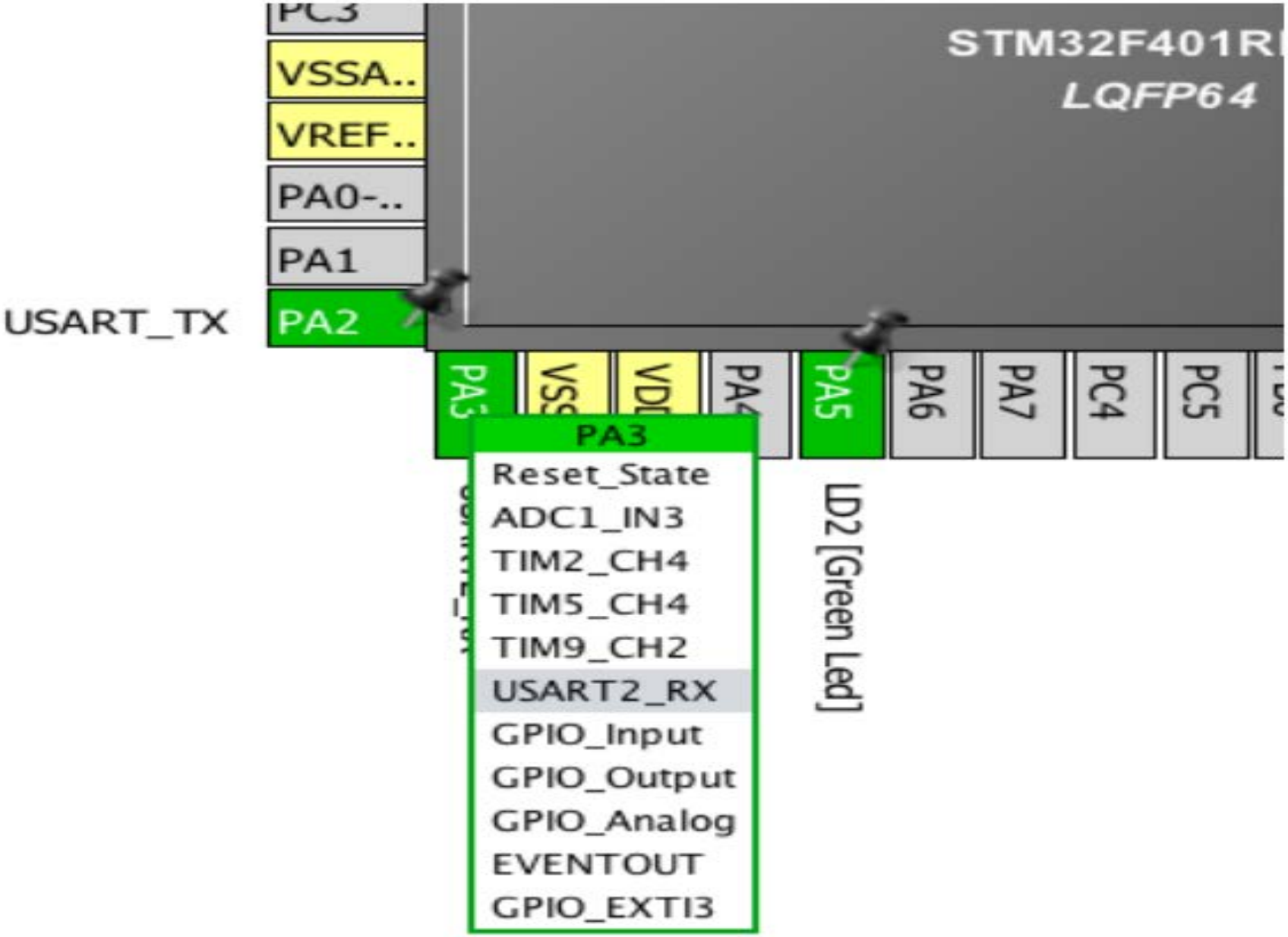
2



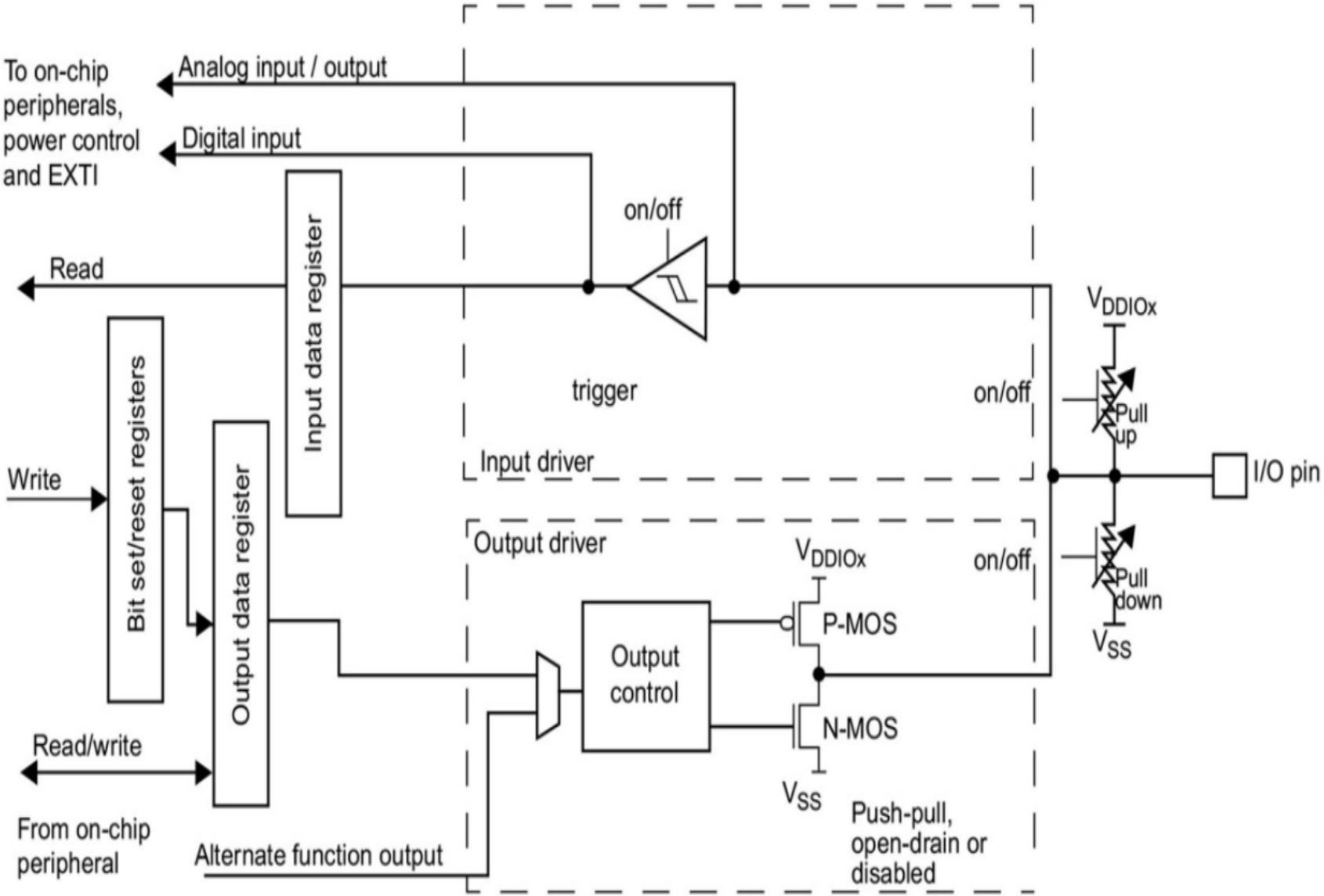
موضوعات المحاضرة:

- 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 stm32ومكتبة HAL
- برمجة أقطاب الدخل في متحكمات stm32
- التوابع المستخدمة من مكتبة HALللتحكم بالمداخل الرقمية في متحكم STM32
- بناء تطبيق لإضاءة ليد من خلال مفتاح لحظي باستخدام متحكمات stm32ومكتبة HAL

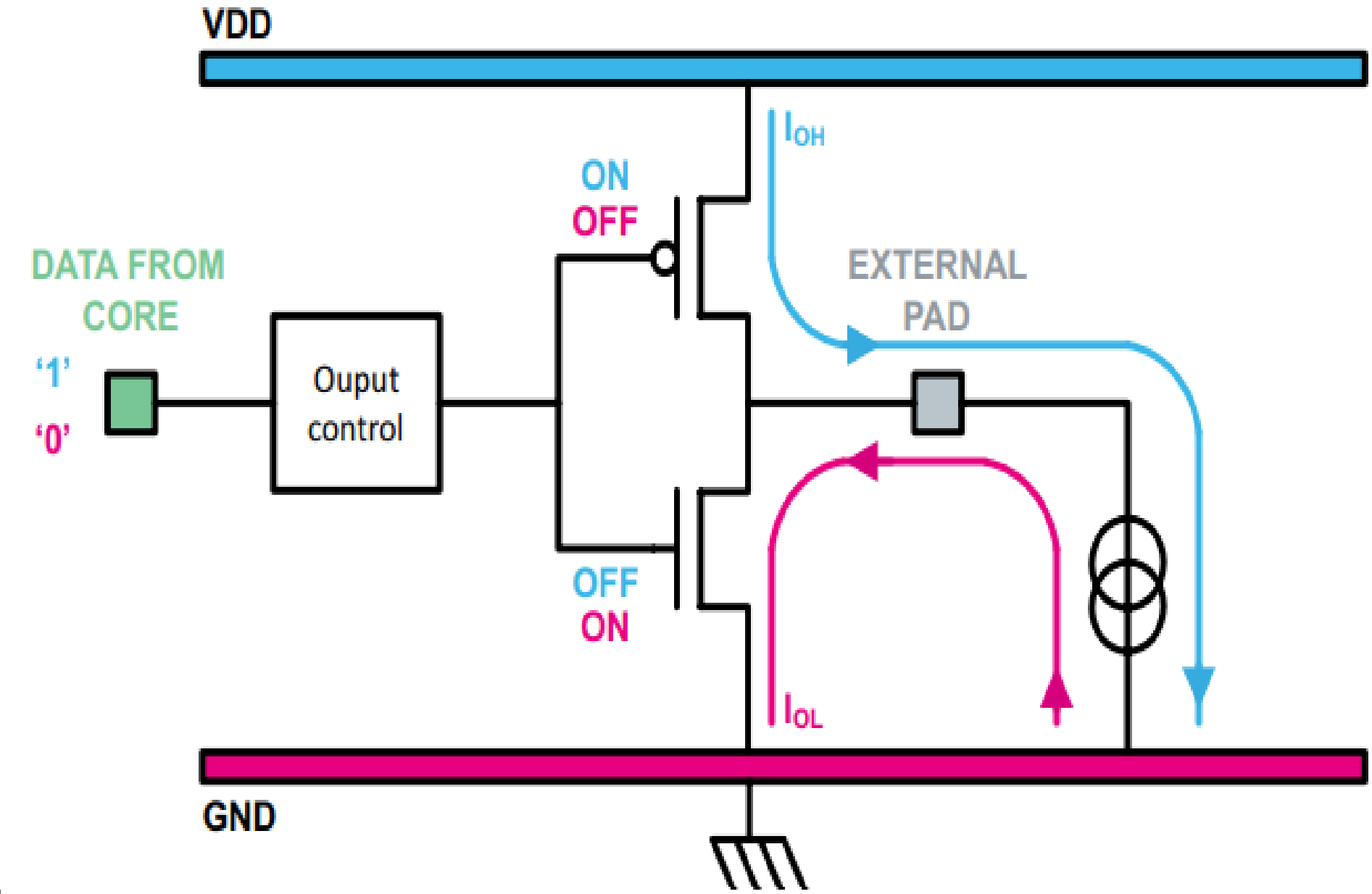
1. أنماط عمل أقطاب المتحكم GPIO Mode



1. أنماط عمل أقطاب المتحكم GPIO Mode



2. برمجة أقطاب الخرج في متحكمات stm32

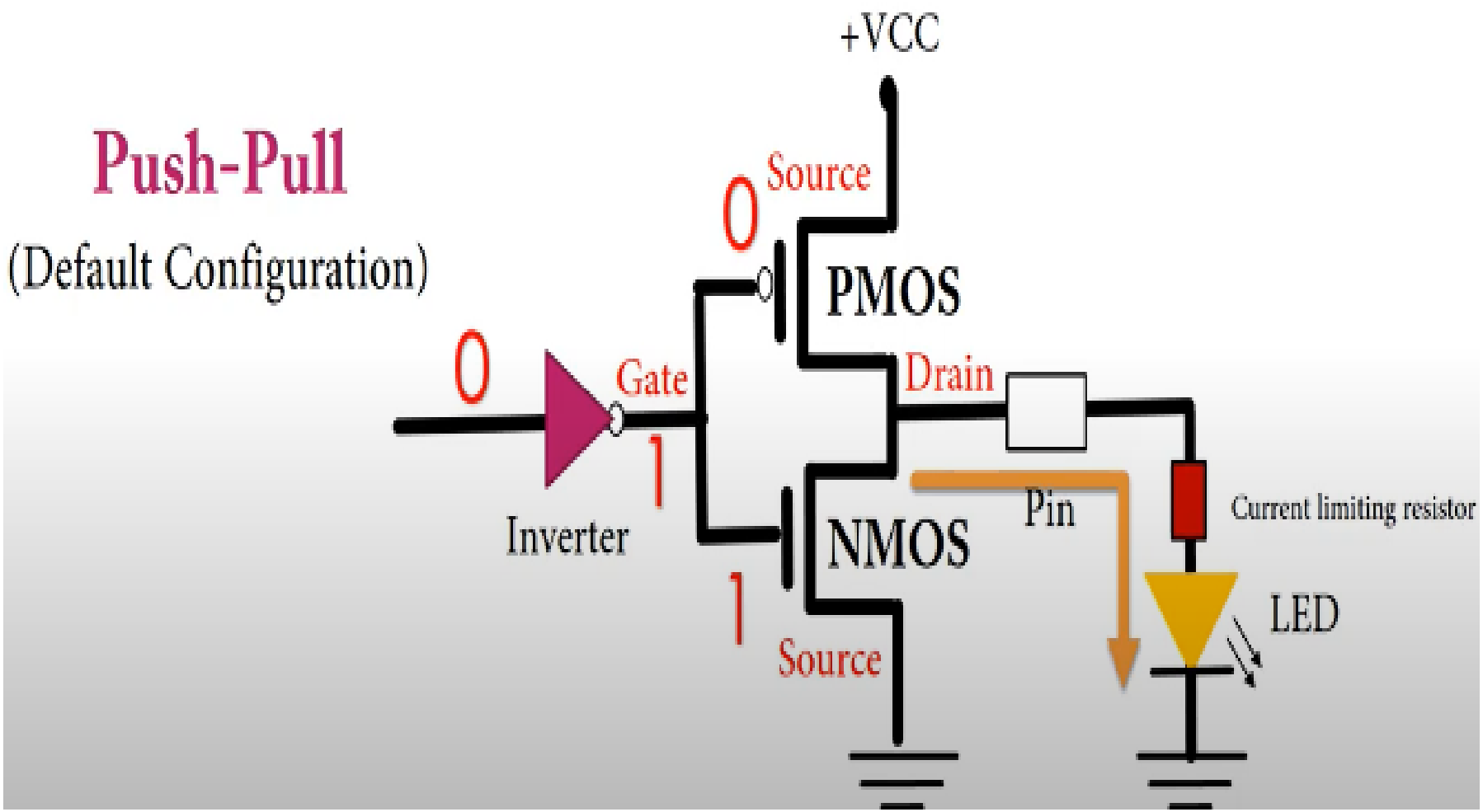


نميز نمطي عمل لكل قطب خرج من أقطاب المتحكم GPIO:

- **نمط PUSH-PULL:** فبفرض تم وصل مصعد ليد مع قطب الإخراج للمتحكم المصغر، فعند تطبيق واحد منطقي على القطب يصبح الترانزستور PMOSبحالة تشغيل onوبالتالي يتم تطبيق جهد الـ vccعلى مصعد الليد ويضيء الليد، أما عند تطبيق صفر منطقي على القطب يصبح الترانزستور NMOSبحالة تشغيل on وبالتالي يتم تطبيق جهد الأرضي GNDعلى مصعد الليد ويطفأ الليد كما هو موضح بالشكل التالي:

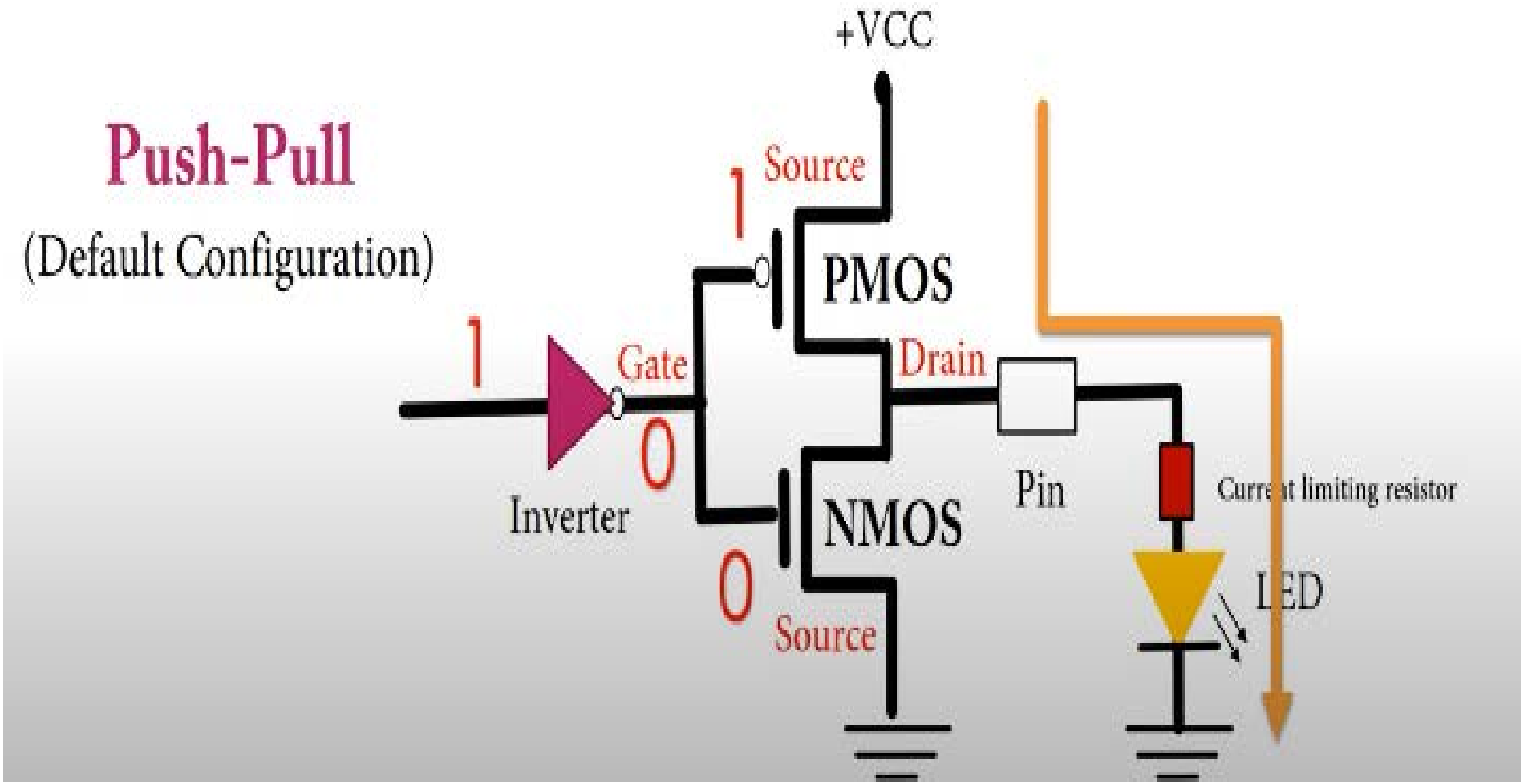
نميز نمطي عمل لكل قطب خرج من أقطاب المتحكم GPIO:

- **نمط PUSH-PULL:**



نميز نمطي عمل لكل قطب خرج من أقطاب المتحكم GPIO:

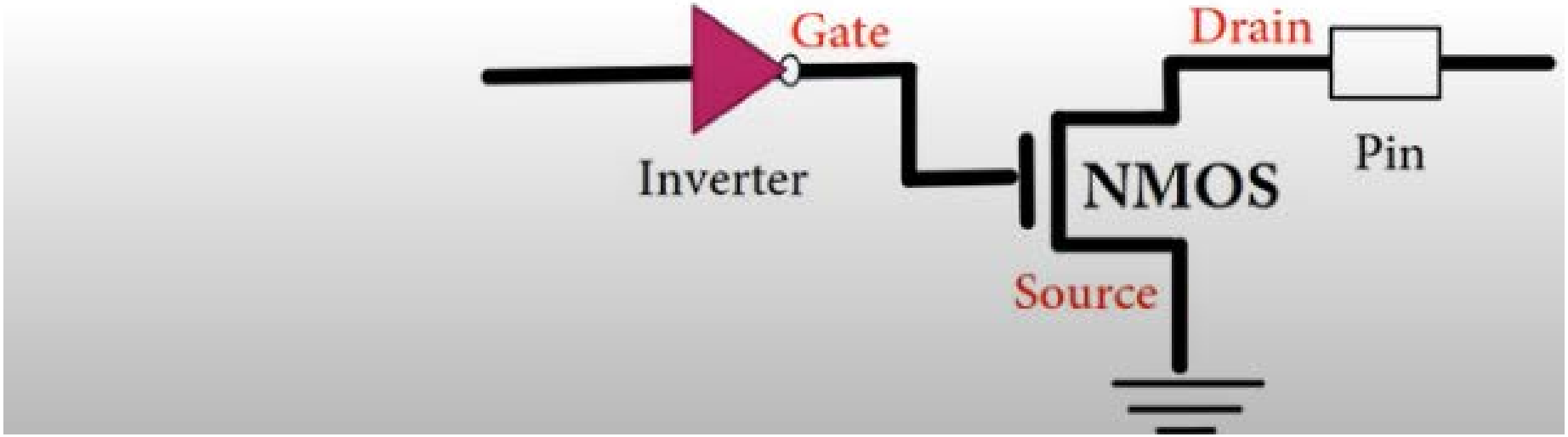
- **نمط PUSH-PULL:**



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 ترانزستورواحد من نوع NMOS كما هو موضح في الأشكال التالية:

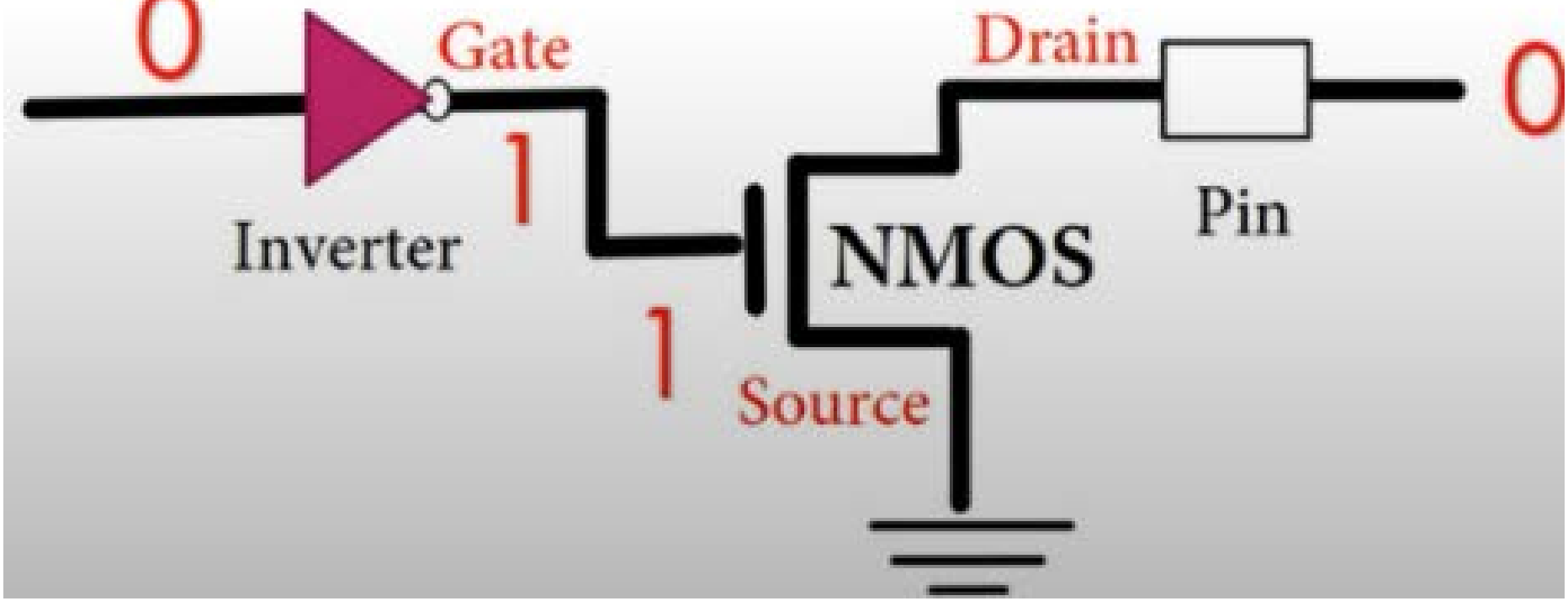
Open Drain



2. برمجة أقطاب الخرج في متحكمات stm32

نميز نمطي عمل لكل قطب خرج من أقطاب المتحكم GPIO:

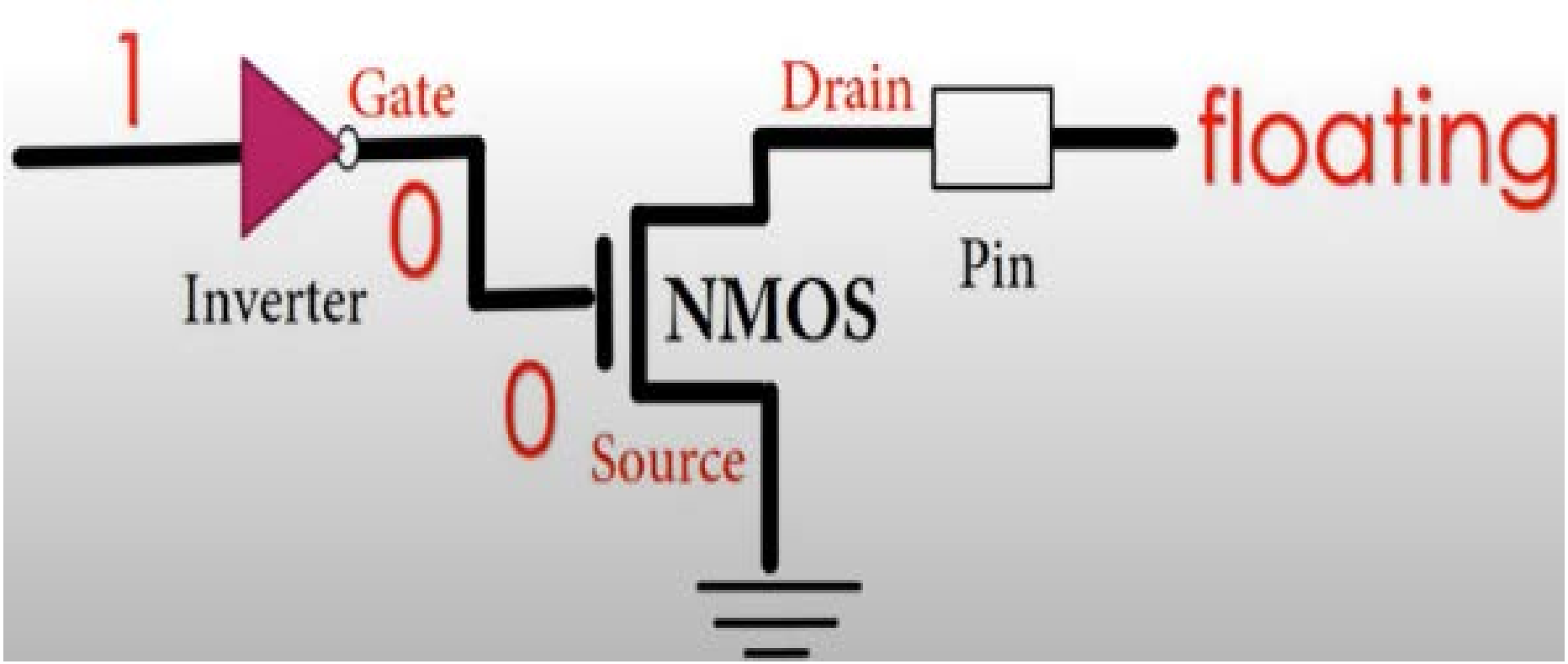
- **نمط Open-Drain** في هذا النمط يمكن للمتحكم أن يعمل كمصرف للتيار sink فقط، فيتم قيادة قطب الخرج من خلال ترانزستورواحد من نوع NMOS كما هو موضح في الأشكال التالية:



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 ترانزستورواحد من نوع NMOS كما هو موضح في الأشكال التالية:

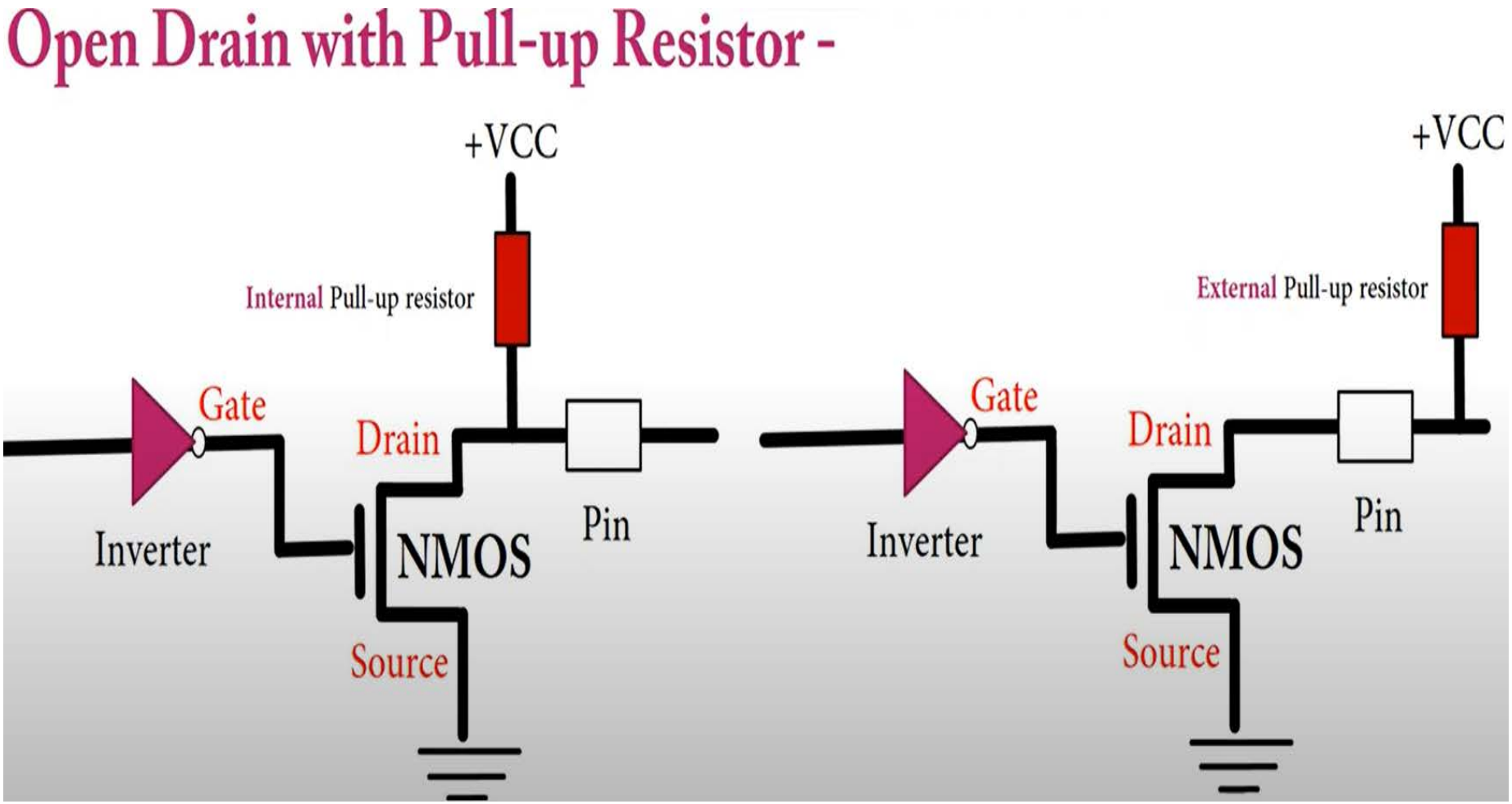


نميز نمطي عمل لكل قطب خرج من أقطاب المتحكم GPIO:

- **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:

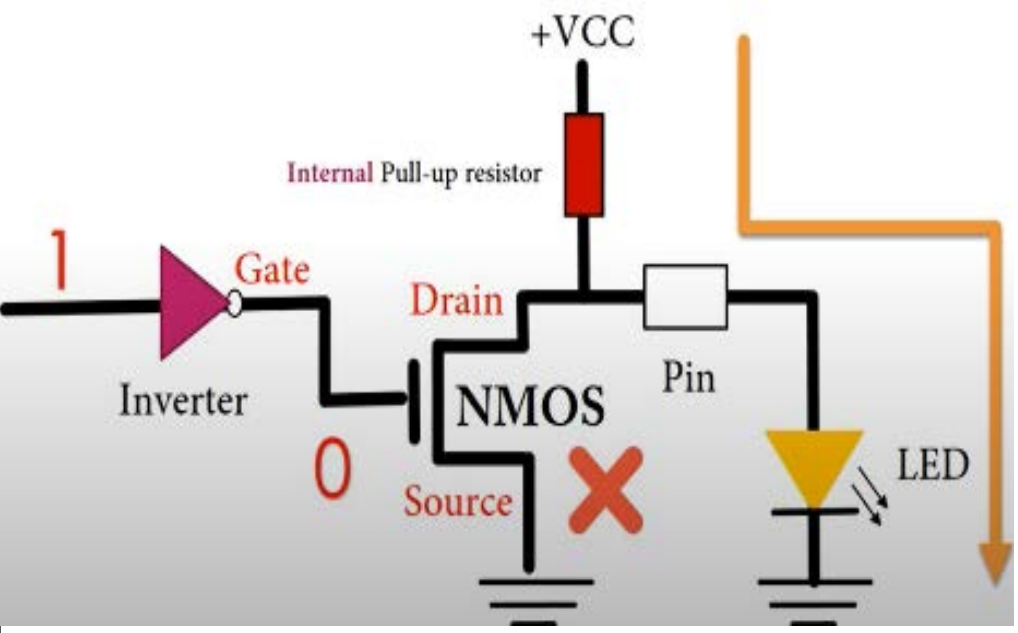
نميز نمطي عمل لكل قطب خرج من أقطاب المتحكم GPIO:

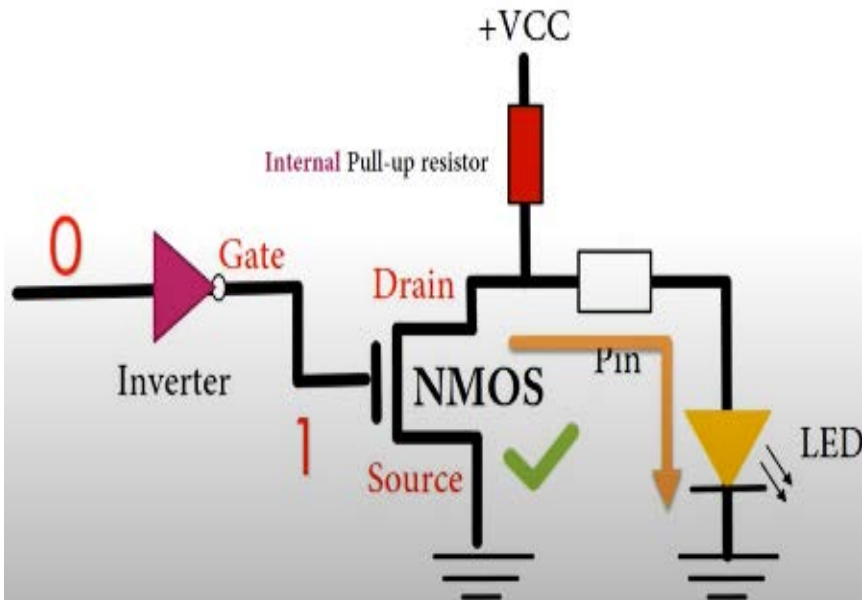
- نمط **Open-Drain**



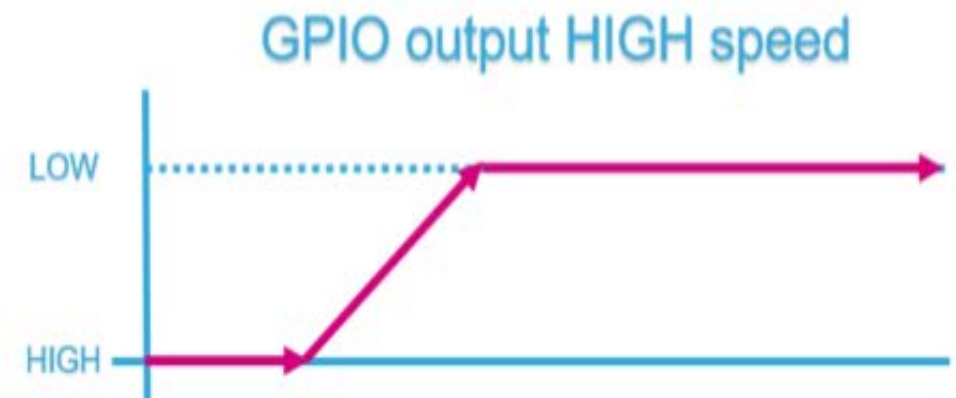
نميز نمطي عمل لكل قطب خرج من أقطاب المتحكم GPIO:

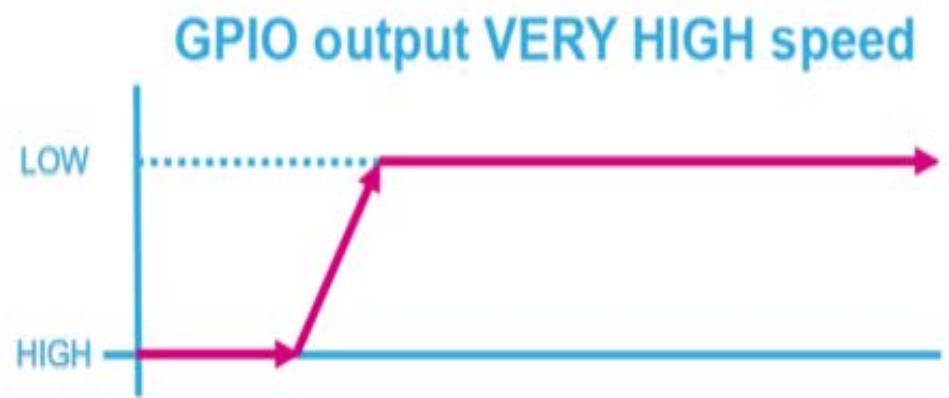
- نمط **Open-Drain**





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:





2. برمجة أقطاب الخرج في متحكمات stm32

يمكن **استجرار تيار** من أي قطب من أقطاب المتحكم بحدود 25mA على ألا يتجاوز الاستجرار الكلي للتيار من جميع أقطاب المتحكم الـ 150mA كما هو موضح بالجدول التالى:

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:

```
void 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:

- مثال1: لكتابة واحد منطقي على القطب رقم10 من المنفذ Dنستخدم التابع التالي:

- **HAL_GPIO_WritePin(GPIOD, GPIO_PIN_10, GPIO_PIN_SET);**
- مثال2: لكتابة صفر منطقي على القطب رقم5 من المنفذ Aنستخدم 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 Aنستخدم التابع التالي:

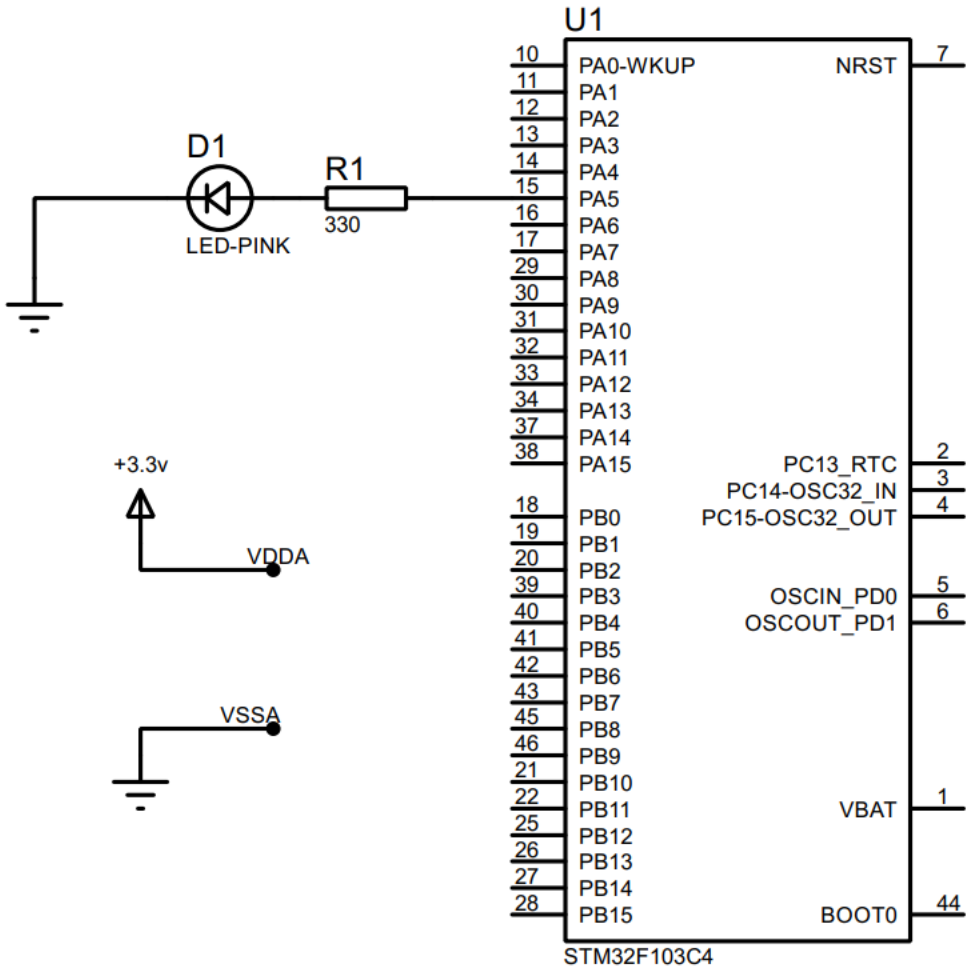
HAL_GPIO_TogglePin(GPIOA,GPIO_PIN_5);

- **لإضافة تأخير زمني بالميللي ثانية نستخدم التابع التالي:**

HAL_Delay(Milliseconds)

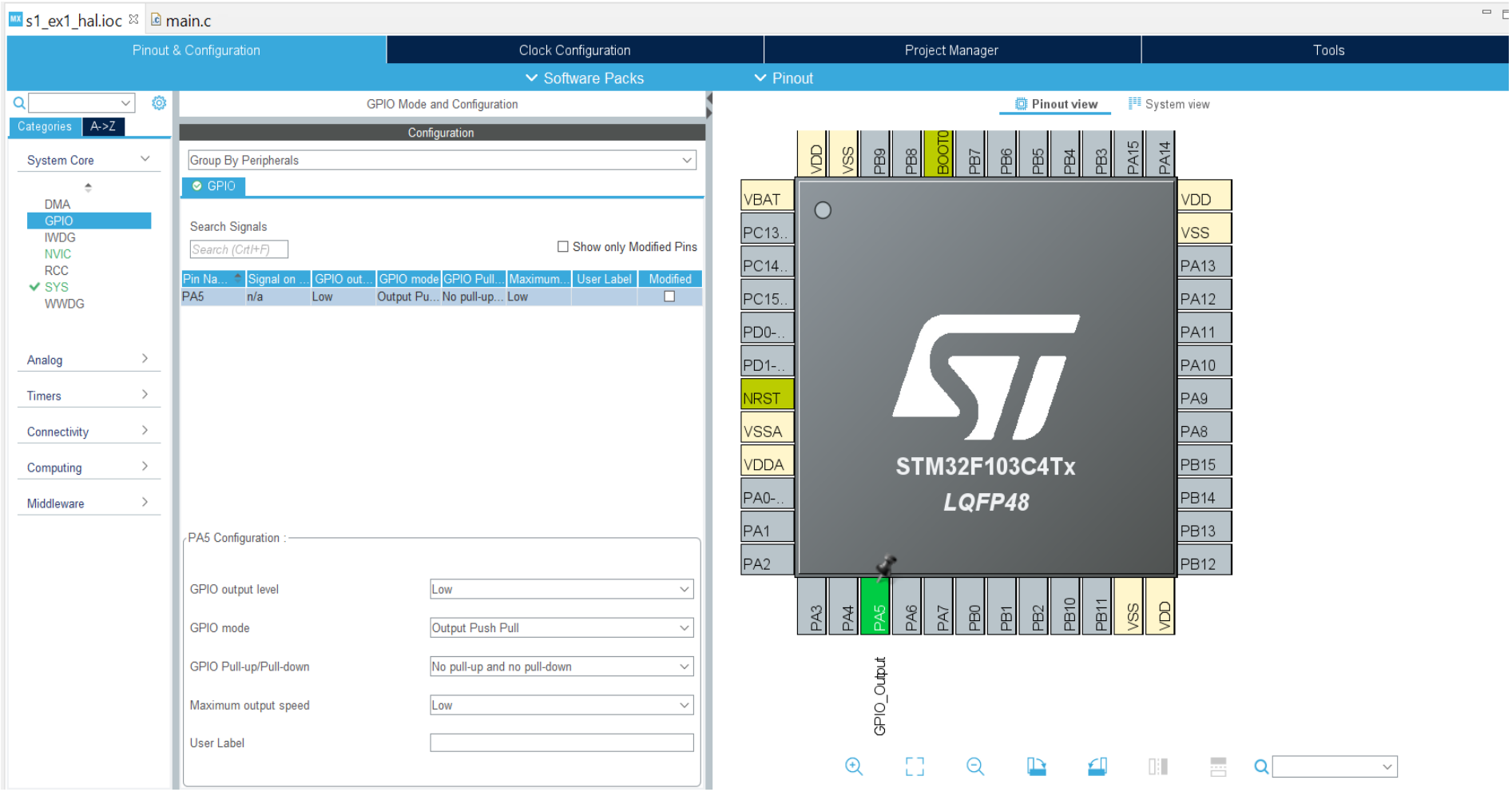
التطبيق الأول: إضاءة وإطفاء كل 0.5sec باستخدام المكتبات STM32 ومكتبة HAL

- سنقوم بتصميم تطبيق يقوم بعمل toggle لليد المتصل بالقطب PA5



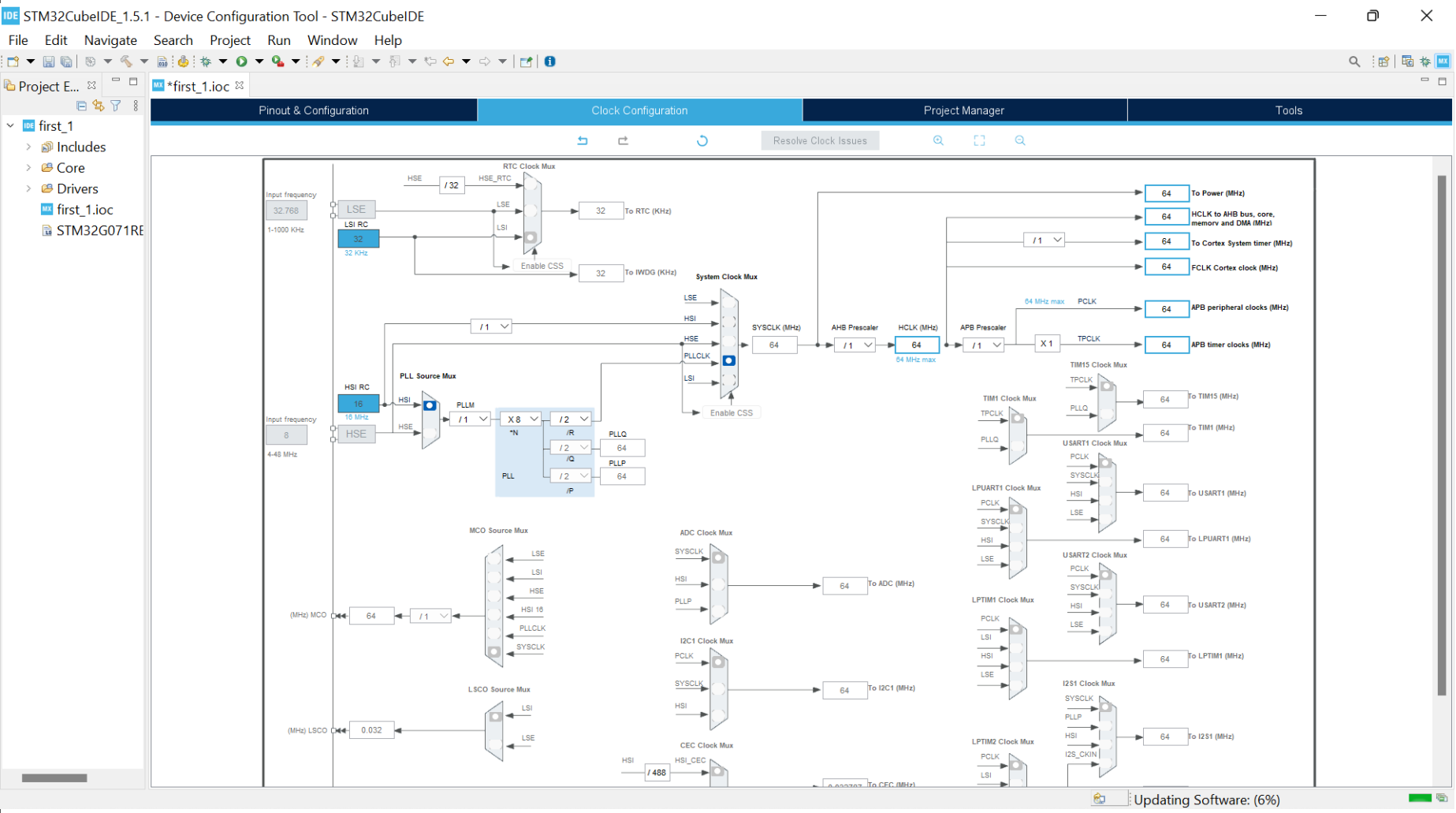
التطبيق الأول: إضاءة وإطفاء LED كل 0.5sec باستخدام المقاطعات ومكتبة HAL وSTM32

- نقوم بضبط إعدادات القطب PA5 كقطب خرج



التطبيق الأول: إضاءة وإطفاء الـ Led كل 0.5sec باستخدام المتحكم STM32 ومكتبة HAL

- نقوم بضبط تردد الساعة للمتحكم



التطبيق الأول: إضاءة وإطفاء كل 0.5sec باستخدام مكتبة HAL ومكتبة STM32

- 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 وإطفاء كل 0.5sec باستخدام مكتبات STM32 ومكتبة HAL

```
while (1)
{
    HAL_GPIO_TogglePin(GPIOA, GPIO_PIN_5);
    HAL_Delay(500);
  }

}
```

التحكم بالمخارج الرقمية للمتحكم 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.

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 بالمعالج الرئيسي للمتحكم STM32 دون استخدام مكتبة HAL:

- لكتابة واحد منطقي على القطب رقم 5 من المنفذ Aنكتب:
- 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 setمن خلال كتابة :
- GPIOA->ODR = 0xFFFF; // Set the PORTA HIGH
- كما يمكن جعل المنفذ بالكامل بحالة resetمن خلال كتابة :
- GPIOA->ODR = 0x00; // Reset the PORTA

التطبيق التالي: إضاءة ليد وإطفاؤه 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 التالي: إضاءة ليد وإطفاؤه كل 0.5 sec دون استخدام مكتبة HAL

```
while (1)
 {
  GPIOA->ODR = 1<<5;
      HAL_Delay(500);
      GPIOA->ODR &= ~(1<<5);
      HAL_Delay(500);
 }

}
```

نهاية المحاضرة

Thank you for listening