# CLASE 3

## GPIO

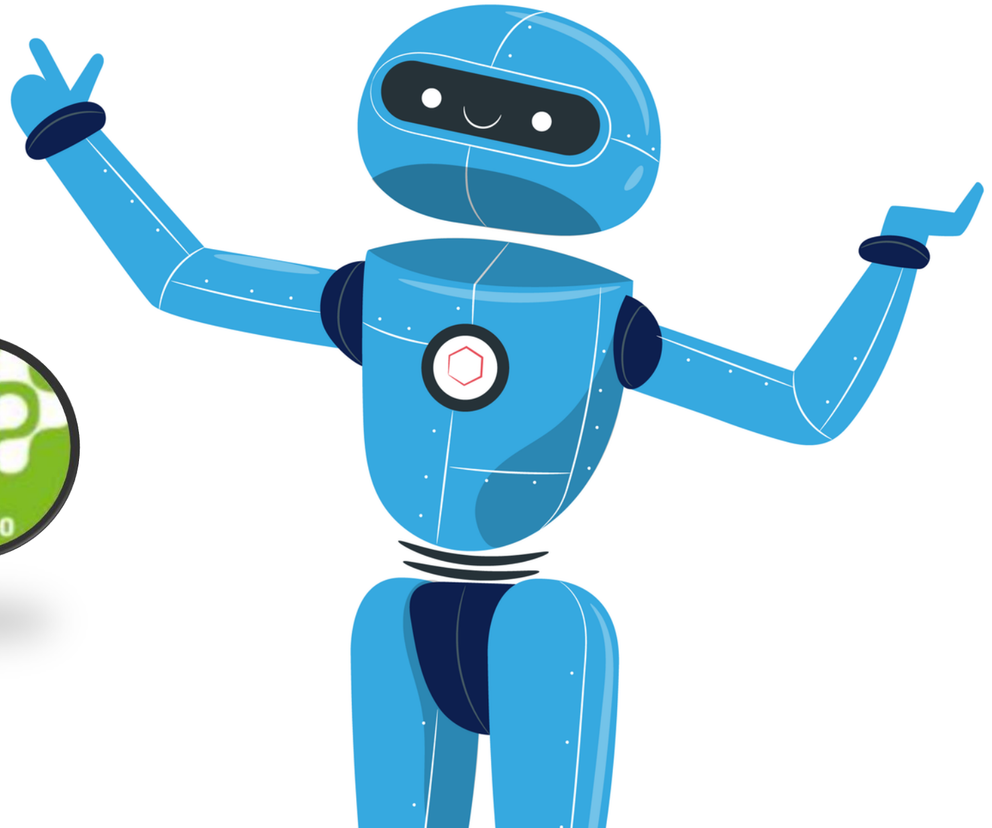
## MICROCONTROLADORES

ARM

life.augmented

STM32 L4

STM32 Nucleo



TEXAS INSTRUMENTS


UMAKER | CENTRO DE CAPACITACIÓN DE DESARROLLO TECNOLÓGICO

# DESCRIPCION EL GPIO

arm

MICRO-
CONTRO-
LADORES
ARM

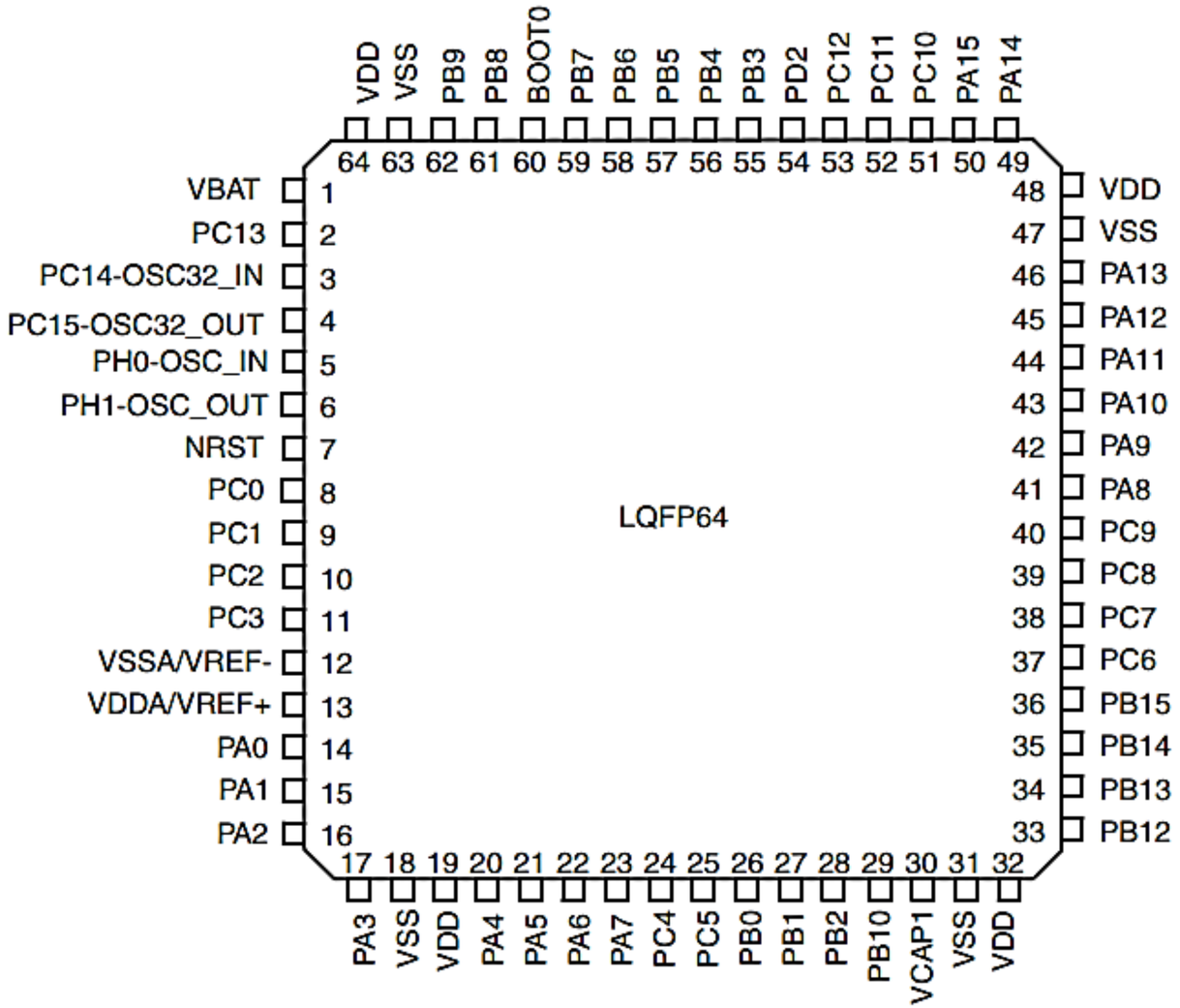
UMAKER | CENTRO DE CAPACITACIÓN DE DESARROLLO TECNOLÓGICO

# General-purpose I/Os (GPIO)

## (Entrada/Salida de propósito general)

### CARACTERISTICAS

- Cada puerto tiene hasta 16 pines.
- Estados de salida: push-pull/open drain + pull-up/pull-down.
- Selección de velocidad para cada pin E/S.
- Estados de entrada: Floating, pull-up/pull-down y analógico.
- Registros de entrada/salida de función alternativa(10 Afs por pin).



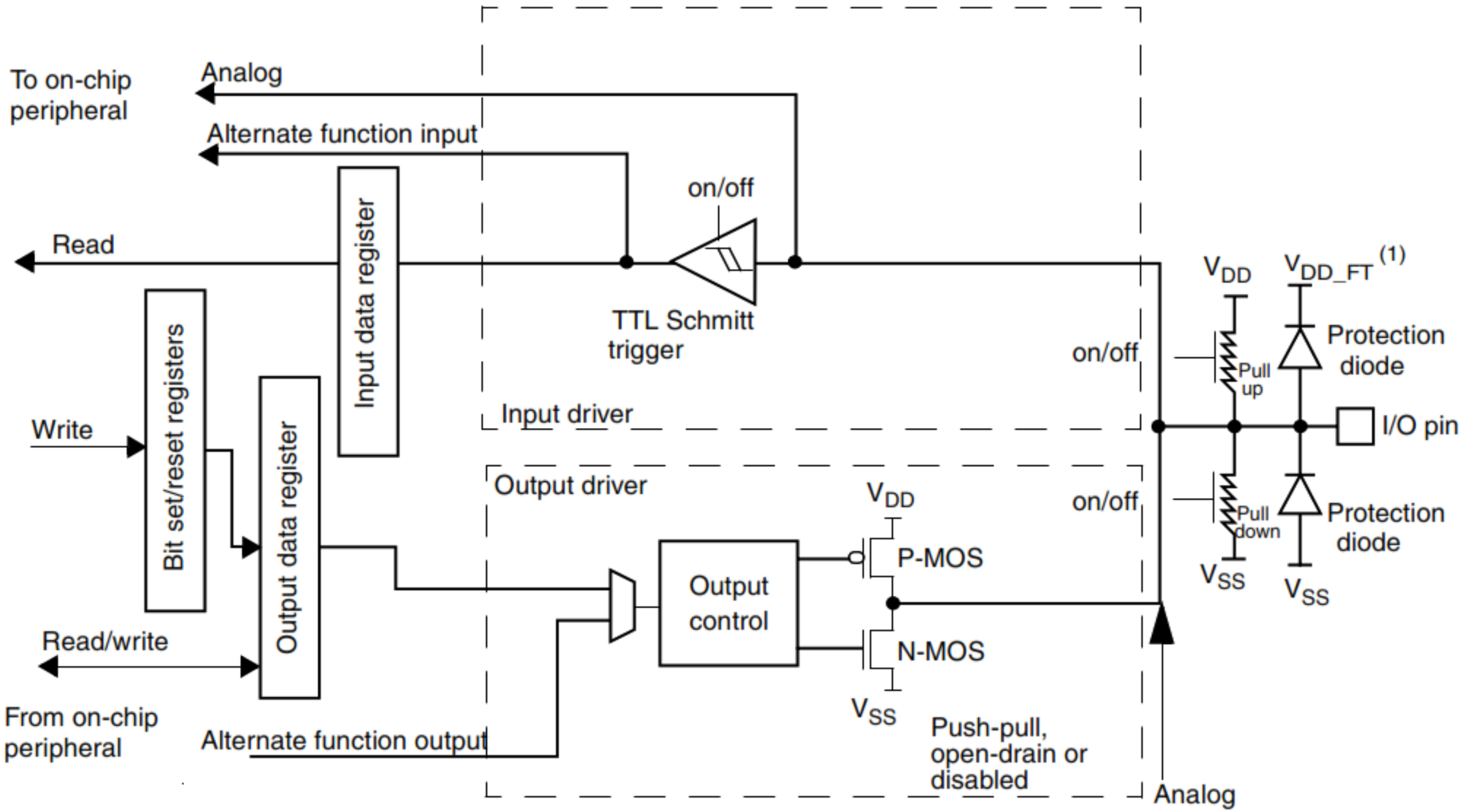

arm

MICRO-CONTRO-LADORES ARM

UMAKER | CENTRO DE CAPACITACIÓN DE DESARROLLO TECNOLÓGICO

# General-purpose I/Os (GPIO)

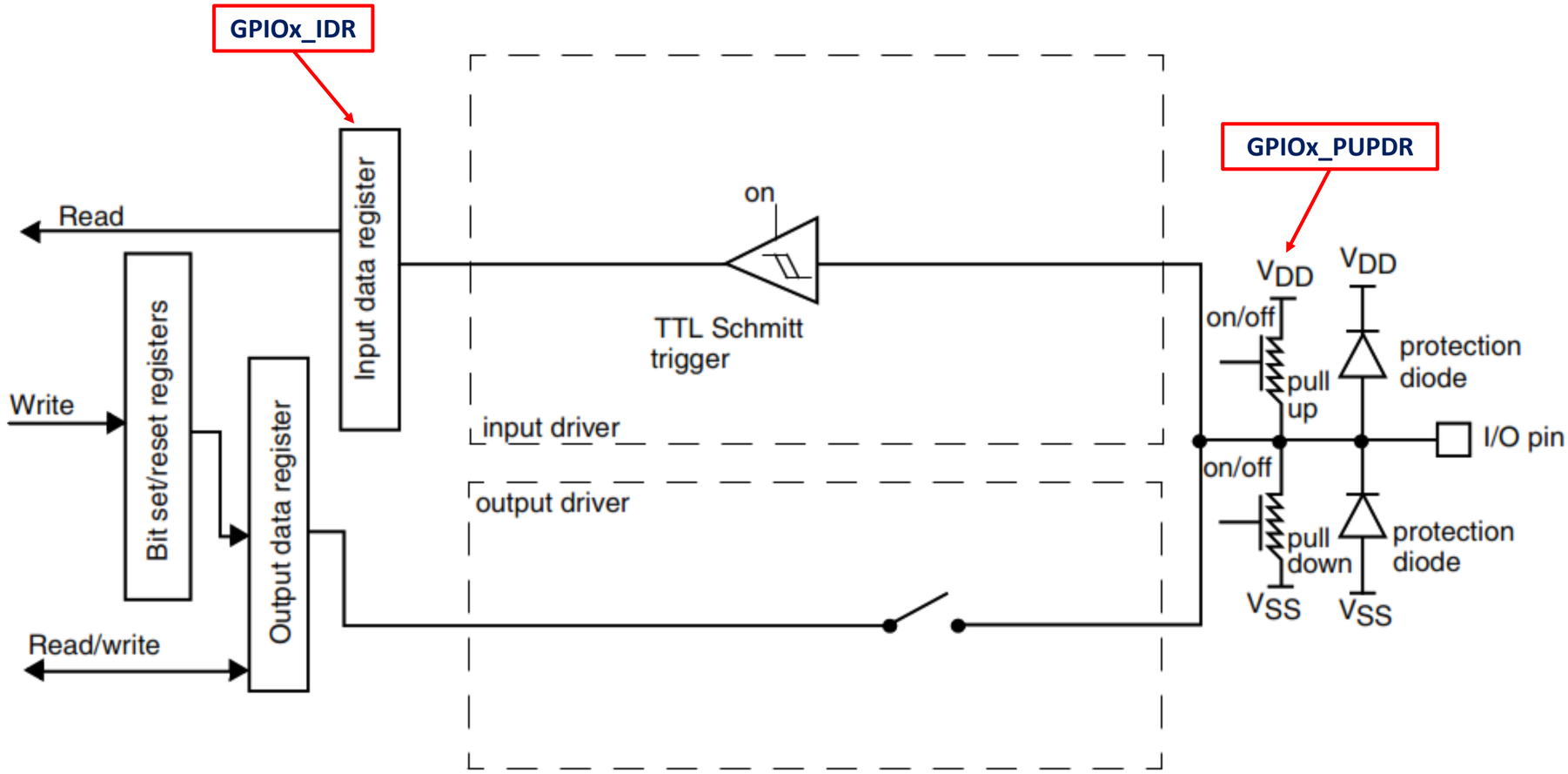
## (Entrada/Salida de propósito general)



# General-purpose I/Os (GPIO)

## (Entrada/Salida de propósito general)

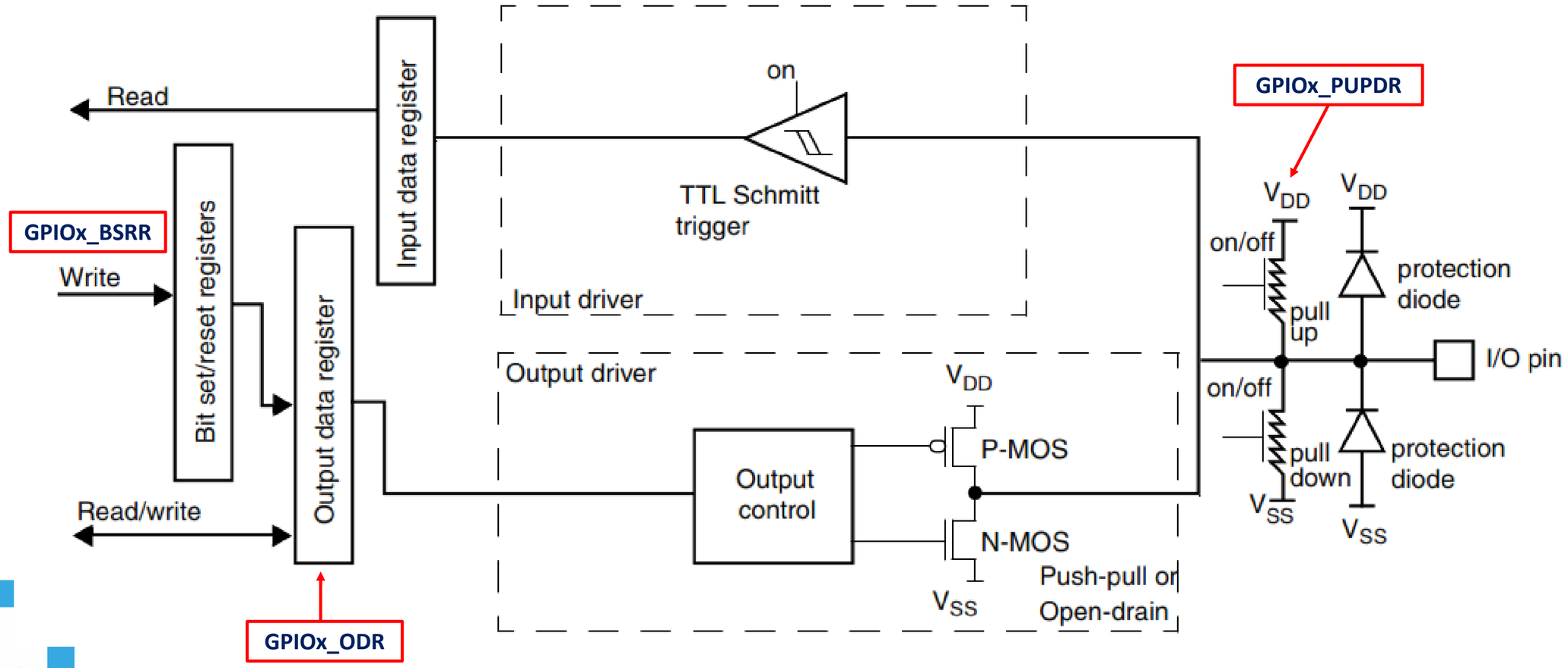
**CONFIGURACION DE ENTRADA → GPIOx_MODER**



# General-purpose I/Os (GPIO)

## (Entrada/Salida de propósito general)

**CONFIGURACION DE SALIDA → GPIOx_MODER**




MICRO-CONTRO-LADORES ARM

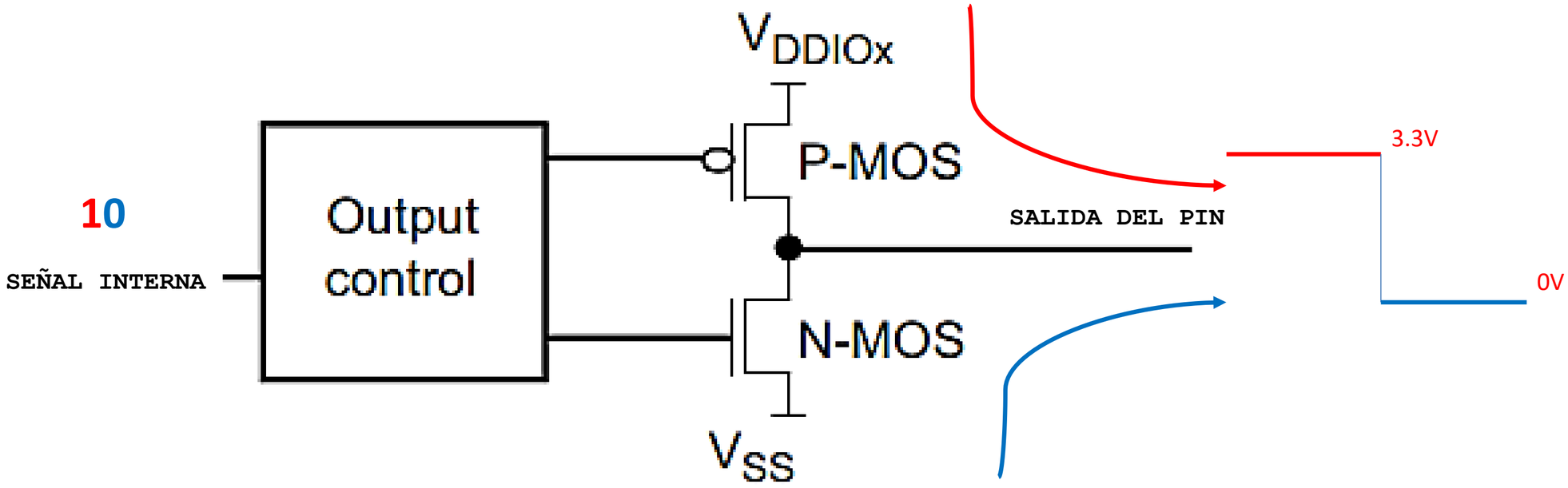
UMAKER | CENTRO DE CAPACITACIÓN DE DESARROLLO TECNOLÓGICO

# General-purpose I/Os (GPIO)

(Entrada/Salida de propósito general)

CONFIGURACION DE SALIDA → GPIOx_MODER

PUSH-PULL → GPIOx_OTYPER

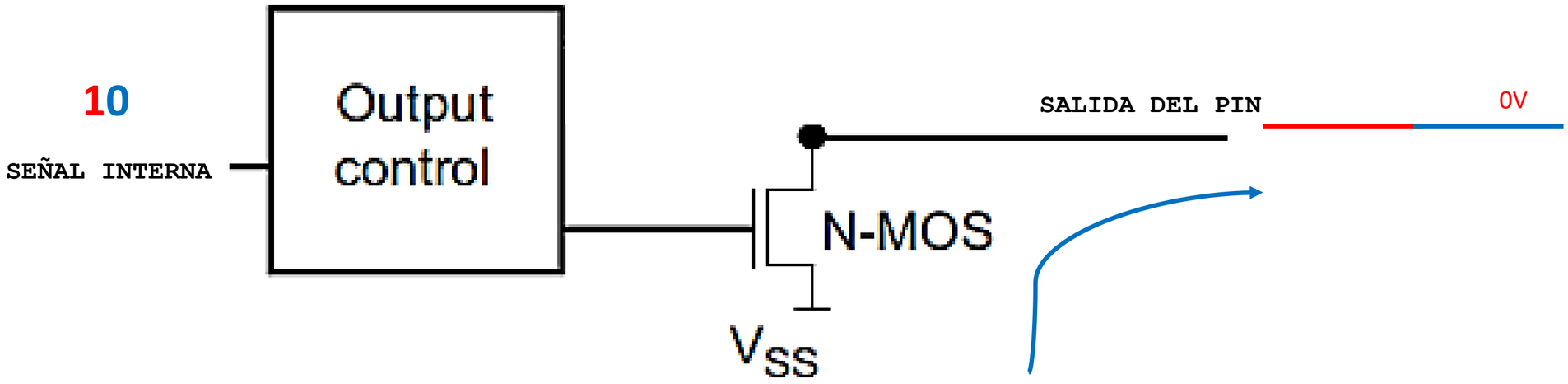


# General-purpose I/Os (GPIO)

(Entrada/Salida de propósito general)

CONFIGURACION DE SALIDA → GPIOx_MODER

OPEN-DRAIN → GPIOx_OTYPER



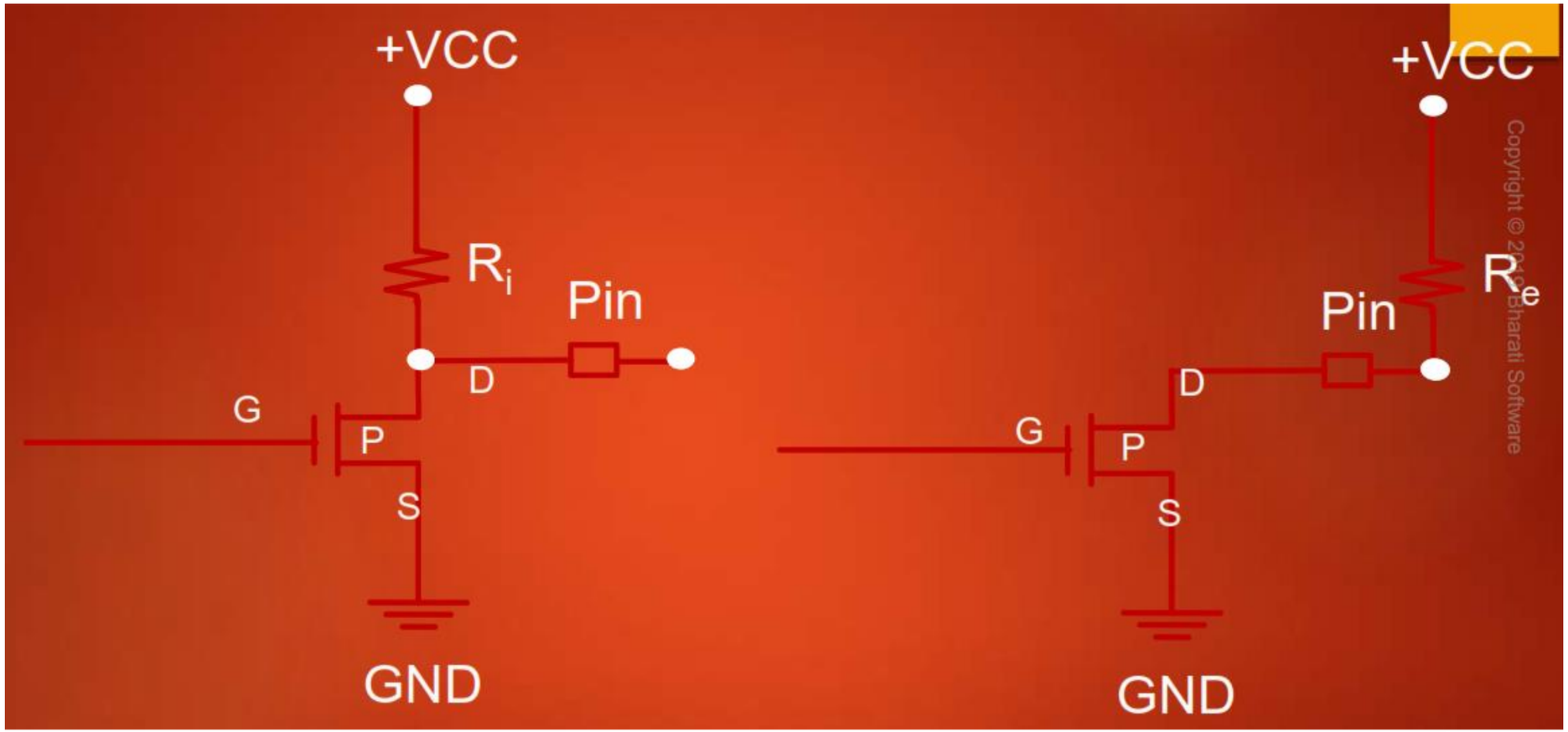
# General-purpose I/Os (GPIO)

(Entrada/Salida de propósito general)

CONFIGURACION DE SALIDA → GPIOx_MODER

OPEN-DRAIN

PULL-UP INTERNO/EXTERNO



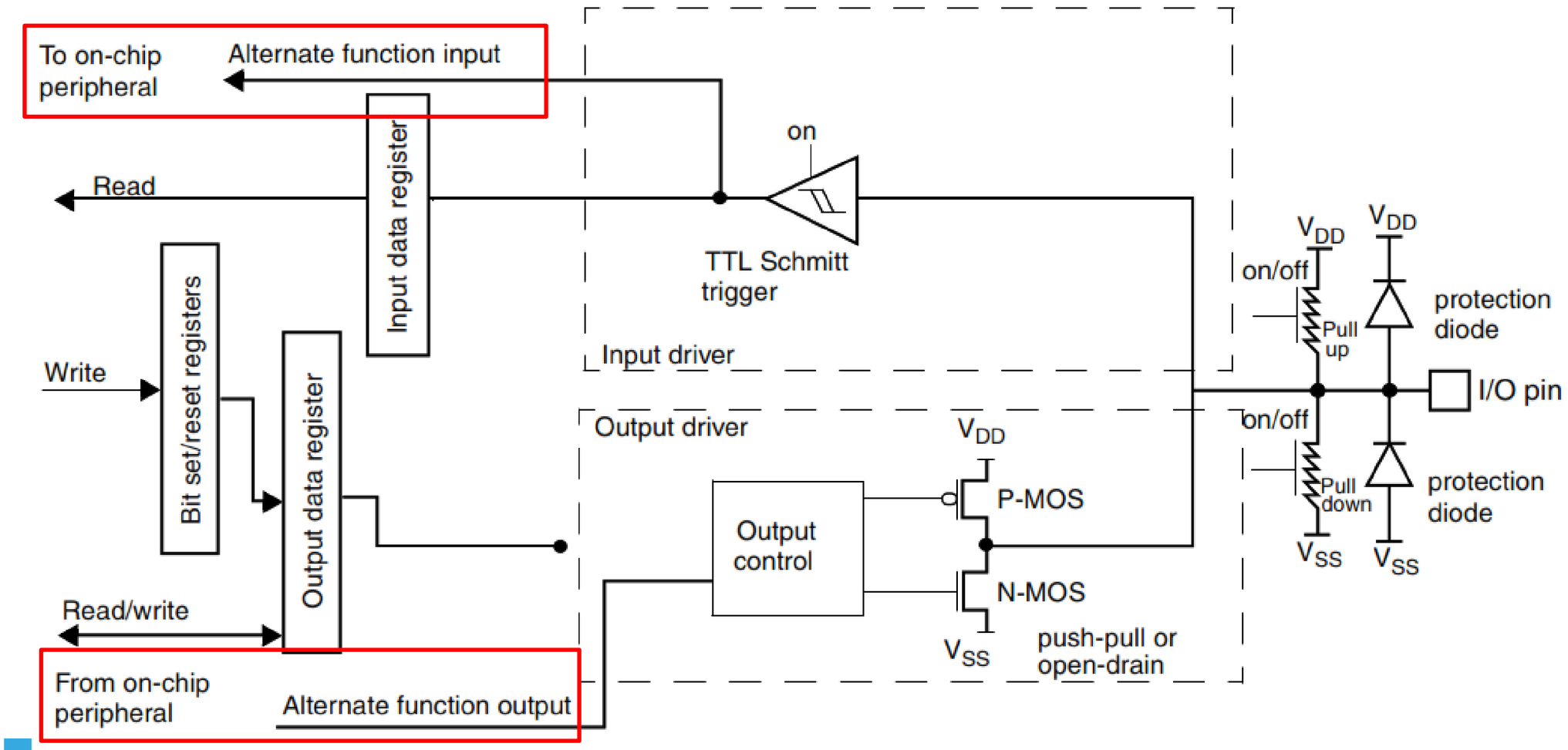

MICRO-CONTRO-LADORES ARM

UMAKER | CENTRO DE CAPACITACIÓN DE DESARROLLO TECNOLÓGICO

# General-purpose I/Os (GPIO)

## (Entrada/Salida de propósito general)

**CONFIGURACION DE FUNCION ALTERNATIVA → GPIOx_MODER**



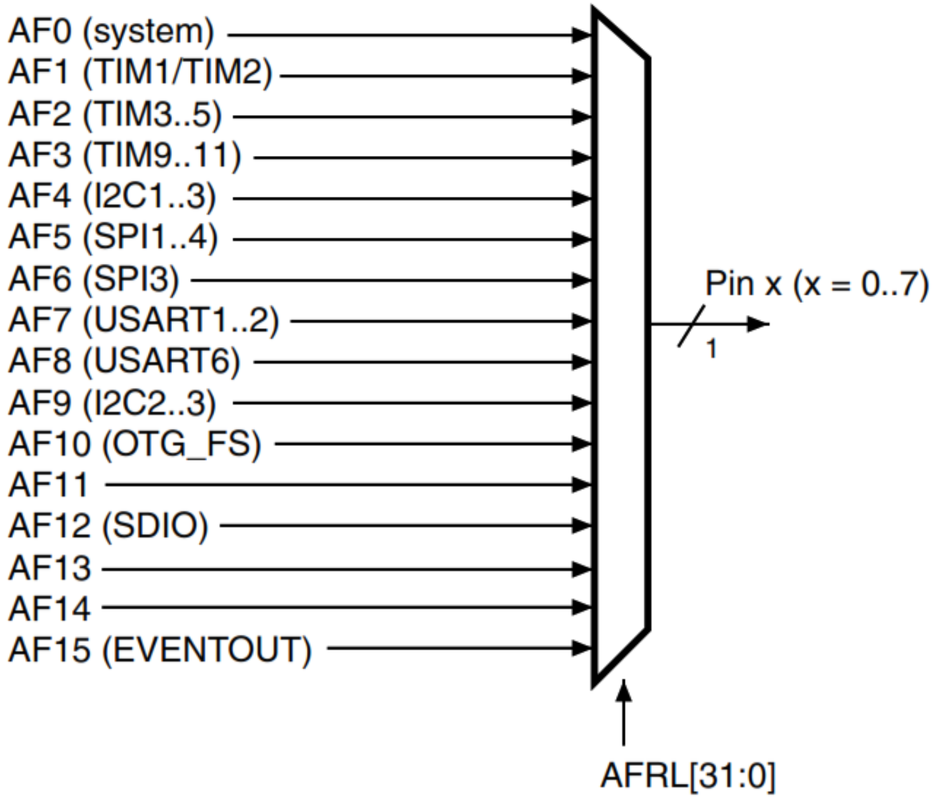

arm
MICRO-CONTRO-LADORES ARM

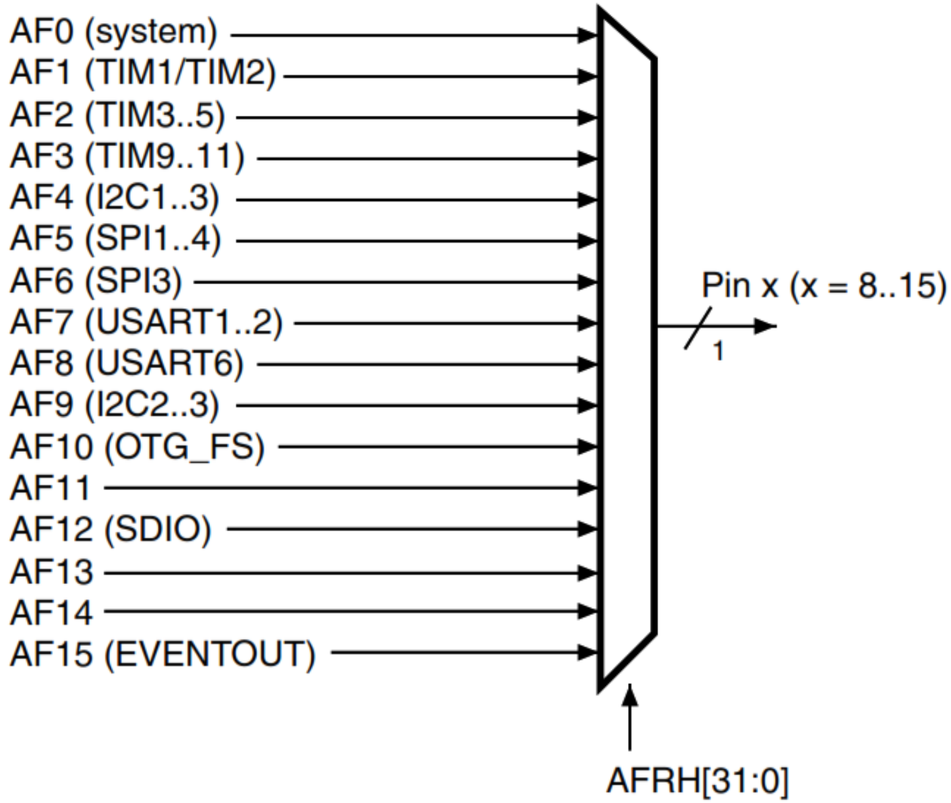
UMAKER | CENTRO DE CAPACITACIÓN DE DESARROLLO TECNOLÓGICO

# General-purpose I/Os (GPIO)

## (Entrada/Salida de propósito general)

### CONFIGURACION DE FUNCION ALTERNATIVA → GPIOx_AFLR/GPIOx_AFHR



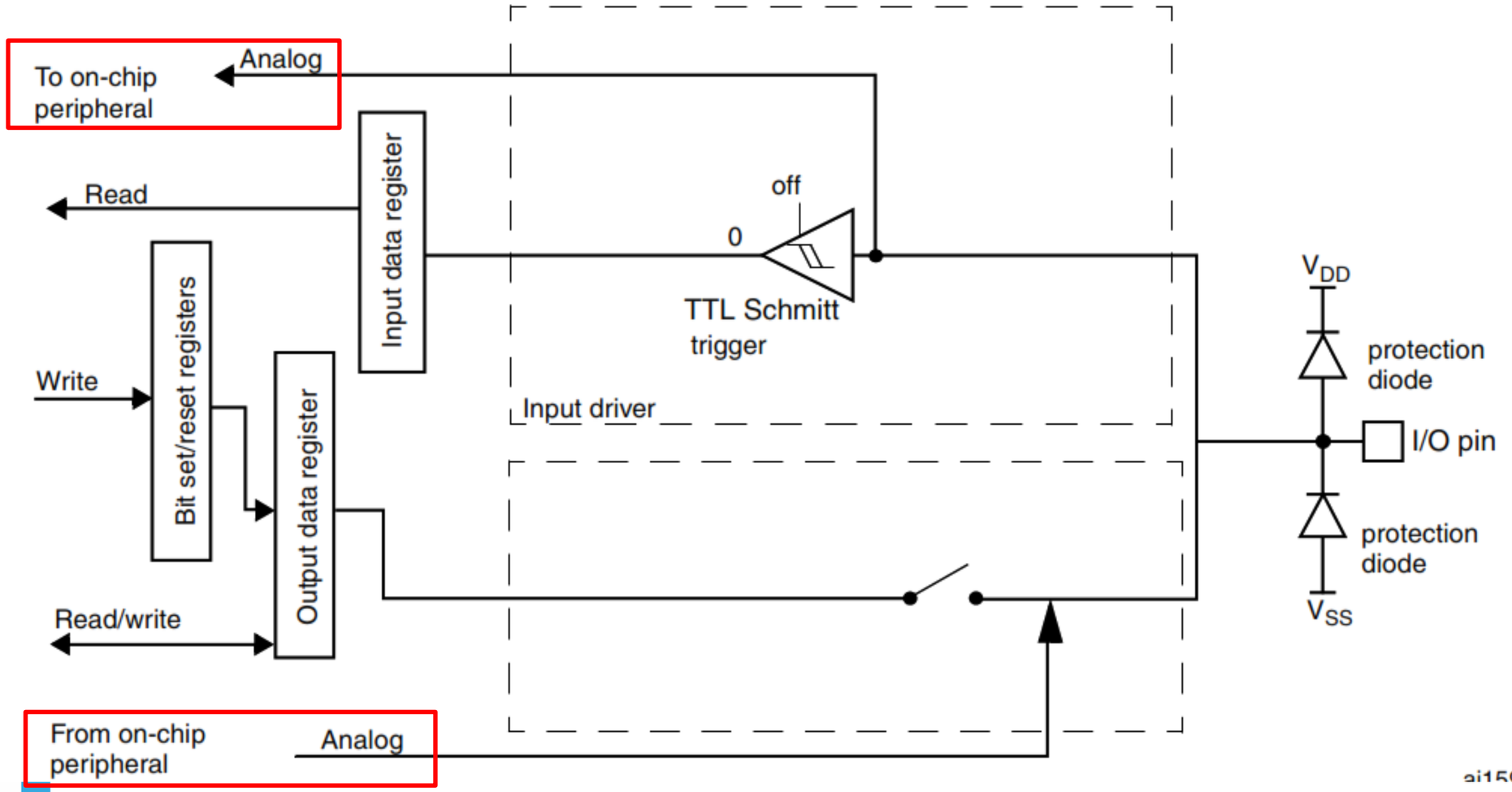

MICRO-CONTRO-LADORES ARM

UMAKER | CENTRO DE CAPACITACIÓN DE DESARROLLO TECNOLÓGICO

# General-purpose I/Os (GPIO)

## (Entrada/Salida de propósito general)

**ENTRADA ANALOGICA → GPIOx_MODER**



# REGISTROS DEL GPIO

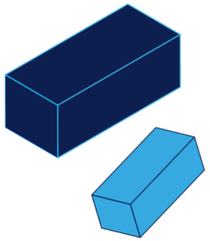
arm

MICRO-
CONTRO-
LADORES
ARM

UMAKER | CENTRO DE CAPACITACIÓN
DE DESARROLLO TECNOLÓGICO

# General-purpose I/Os (GPIO)

**(Entrada/Salida de propósito general)**



**REGISTROS DE CONTROL**

Cada puerto GPIOx cuenta con registros de control de 32-bits **(GPIOx_MODER, GPIOx_OTYPER, GPIOx_OSPEEDR, GPIOx_PUPDR)** para configurar los pines de manera individual.

**REGISTROS DE DATOS**

Cada puerto GPIOx cuenta con dos registros de datos de 16-bits **(GPIOx_IDR and GPIOx_ODR).**

**REGISTRO DE MANEJO DE DATOS BIT A BIT**

Cada puerto cuenta con un registro que permite restablecer y establecer cada bit de manera individual.

**REGISTROS DE BLOQUEO**

Registro de bloqueo que bloquea los registros**(GPIOx_MODER, GPIOx_OTYPER, GPIOx_OSPEEDR, GPIOx_PUPDR, GPIOx_AFRL and GPIOx_AFRH)**

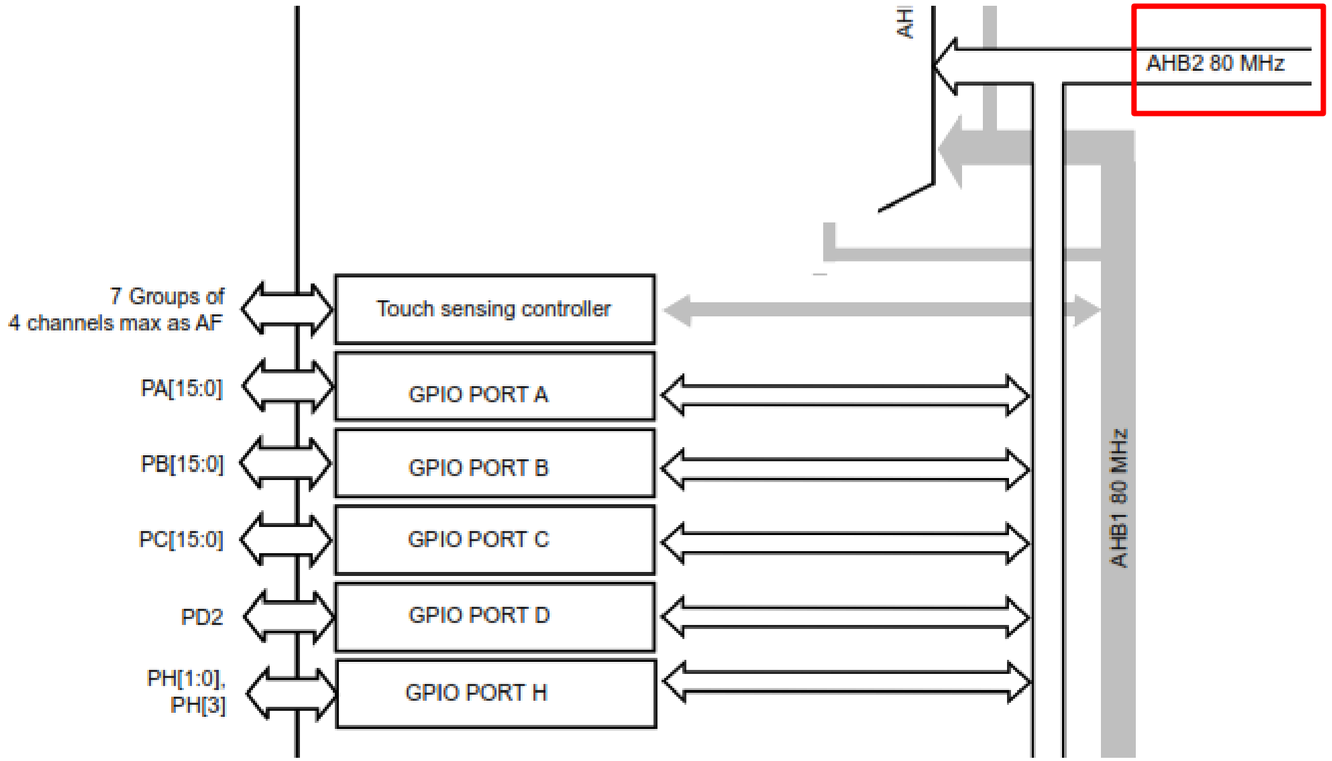
arm

MICRO-CONTRO-LADORES ARM

UMAKER | CENTRO DE CAPACITACIÓN DE DESARROLLO TECNOLÓGICO

# General-purpose I/Os (GPIO)

## (Entrada/Salida de propósito general)




### AHB2

| | | | |
|---|---|---|---|
| AHB2 | 0x4800 2000 - 0x4FFF FFFF | ~127 MB | Reserved |
| | 0x4800 1C00 - 0x4800 1FFF | 1 KB | GPIOH |
| | 0x4800 1400 - 0x4800 1BFF | 2 KB | Reserved |
| | 0x4800 1000 - 0x4800 13FF | 1 KB | GPIOE(2) (3) |
| | 0x4800 0C00 - 0x4800 0FFF | 1 KB | GPIOD(2) |
| | 0x4800 0800 - 0x4800 0BFF | 1 KB | GPIOC |
| | 0x4800 0400 - 0x4800 07FF | 1 KB | GPIOB |
| | 0x4800 0000 - 0x4800 03FF | 1 KB | GPIOA |
| | 0x4002 4400 - 0x47FF FFFF | ~127 MB | Reserved |

# General-purpose I/Os (GPIO)

## (Entrada/Salida de propósito general)

### GPIO port mode register (GPIOx_MODER) (x = A..E and H)

Address offset:0x00

Reset value:

- 0xABFF FFFF (for port A)
- 0xFFFF FEBF (for port B)
- 0xFFFF FFFF for ports C..E
- 0x0000 000F (for port H)

PIN 15

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| MODER15[1:0] | | MODER14[1:0] | | MODER13[1:0] | | MODER12[1:0] | | MODER11[1:0] | | MODER10[1:0] | | MODER9[1:0] | | MODER8[1:0] | |
| rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| MODER7[1:0] | | MODER6[1:0] | | MODER5[1:0] | | MODER4[1:0] | | MODER3[1:0] | | MODER2[1:0] | | MODER1[1:0] | | MODER0[1:0] | |
| rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |

PIN 0

Bits 2y:2y+1 **MODERy[1:0]:** Port x configuration bits (y = 0..15)

These bits are written by software to configure the I/O direction mode.

00: Input (reset state)

01: General purpose output mode

10: Alternate function mode

11: Analog mode

PB3 → SALIDA DIGITAL

PA0 → ENTRADA DIGITAL

```
GPIOB->MODER &=~ GPIO_MODER_MODE3;
GPIOB->MODER |= 0x1U<<2*3;
GPIOA->MODER &=~ GPIO_MODER_MODE0;
```

# General-purpose I/Os (GPIO)

## (Entrada/Salida de propósito general)

### GPIO port output type register (GPIOx_OTYPER)

Address offset: 0x04

Reset value: 0x0000 0000

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| OT15 | OT14 | OT13 | OT12 | OT11 | OT10 | OT9 | OT8 | OT7 | OT6 | OT5 | OT4 | OT3 | OT2 | OT1 | OT0 |
| rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |

PIN 0

Bits 31:16 Reserved, must be kept at reset value.

Bits 15:0 **OTy**: Port x configuration bits (y = 0..15)

These bits are written by software to configure the output type of the I/O port.

0: Output push-pull (reset state)

1: Output open-drain

**PB3 → PUSH PULL**

```
/*tipo de salida*/
GPIOB->OTYPER &=~ 0x1U<<3;
```

arm MICRO-CONTRO-LADORES ARM

UMAKER | CENTRO DE CAPACITACIÓN DE DESARROLLO TECNOLÓGICO

# General-purpose I/Os (GPIO)

## (Entrada/Salida de propósito general)

### GPIO port output speed register (GPIOx_OSPEEDR)

Address offset: 0x08
Reset value: 0x0C00 0000 (for port A)
Reset value: 0x0000 0000 (for the other ports)

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| OSPEEDR15 [1:0] | | OSPEEDR14 [1:0] | | OSPEEDR13 [1:0] | | OSPEEDR12 [1:0] | | OSPEEDR11 [1:0] | | OSPEEDR10 [1:0] | | OSPEEDR9 [1:0] | | OSPEEDR8 [1:0] | |
| rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| OSPEEDR7[1:0] | | OSPEEDR6[1:0] | | OSPEEDR5[1:0] | | OSPEEDR4[1:0] | | OSPEEDR3[1:0] | | OSPEEDR2[1:0] | | OSPEEDR1 [1:0] | | OSPEEDR0 1:0] | |
| rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |

PIN 0

Bits 2y:2y+1 **OSPEEDRy[1:0]:** Port x configuration bits (y = 0..15)

These bits are written by software to configure the I/O output speed.

00: Low speed
01: Medium speed
10: High speed
11: Very high speed

*Note: Refer to the product datasheets for the values of OSPEEDRy bits versus $V_{DD}$ range and external load.*

**PB3 → VERY HIGH SPEED**

```
/*Velocidad de conmutacion*/
GPIOB->OSPEEDR |= 0x3U<<2*3;
```

# General-purpose I/Os (GPIO)

## (Entrada/Salida de propósito general)

### GPIO port output speed register (GPIOx_OSPEEDR)

LOW


HIGH


MEDIUM


VERY HIGH


arm

MICRO-CONTRO-LADORES ARM

UMAKER | CENTRO DE CAPACITACIÓN DE DESARROLLO TECNOLÓGICO

# General-purpose I/Os (GPIO)

## (Entrada/Salida de propósito general)

### GPIO port pull-up/pull-down register (GPIOx_PUPDR)

Address offset: 0x0C

Reset values:

- 0x6400 0000 for port A
- 0x0000 0100 for port B
- 0x0000 0000 for other ports

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| PUPDR15[1:0] | | PUPDR14[1:0] | | PUPDR13[1:0] | | PUPDR12[1:0] | | PUPDR11[1:0] | | PUPDR10[1:0] | | PUPDR9[1:0] | | PUPDR8[1:0] | |
| rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| PUPDR7[1:0] | | PUPDR6[1:0] | | PUPDR5[1:0] | | PUPDR4[1:0] | | PUPDR3[1:0] | | PUPDR2[1:0] | | PUPDR1[1:0] | | PUPDR0[1:0] | |
| rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |

PIN 0

Bits 2y:2y+1 **PUPDRy[1:0]:** Port x configuration bits (y = 0..15)

These bits are written by software to configure the I/O pull-up or pull-down

00: No pull-up, pull-down

01: Pull-up

10: Pull-down

11: Reserved

**PB3 → NO PULL/DOWN**

**PA0 → PULL-DOWN**

```
/*resistencia pull up/down*/
GPIOB->PUPDR &=~ GPIO_MODER_MODE3;
GPIOA->PUPDR &=~ GPIO_PUPDR_PUPD0;
GPIOA->PUPDR |= 0x2U<<2*0;
```

arm MICRO-CONTRO-LADORES ARM

UMAKER | CENTRO DE CAPACITACIÓN DE DESARROLLO TECNOLÓGICO

# General-purpose I/Os (GPIO)

## (Entrada/Salida de propósito general)

### GPIO port output data register (GPIOx_ODR) (x = A..E and H)

Address offset: 0x14

Reset value: 0x0000 0000

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |
| 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 |

PIN 0

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..E and H).*

**PB3 → SET/CLEAR**

```
/*PB3->SET*/
GPIOB->ODR |= 1U<<3;
/*PB3->CLEAR*/
GPIOB->ODR &=~1U<<3;
```

MICRO-CONTRO-LADORES ARM

UMAKER | CENTRO DE CAPACITACIÓN DE DESARROLLO TECNOLÓGICO

# General-purpose I/Os (GPIO)

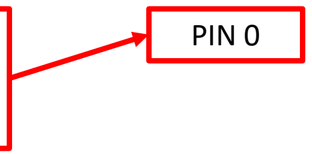
## (Entrada/Salida de propósito general)

### GPIO port input data register (GPIOx_IDR) (x = A..E and 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |
| 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 **IDRy**: 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.

**PA0 → LECTURA**

```
uint8_t PA0_Value;
PA0_Value = (GPIOA->IDR>>0 & 0x1);
```

MICRO-CONTRO-LADORES ARM

UMAKER | CENTRO DE CAPACITACIÓN DE DESARROLLO TECNOLÓGICO

# General-purpose I/Os (GPIO)

## (Entrada/Salida de propósito general)

### GPIO port bit set/reset register (GPIOx_BSRR) (x = A..E and H)

Address offset: 0x18

Reset value: 0x0000 0000

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| BR15 | BR14 | BR13 | BR12 | BR11 | BR10 | BR9 | BR8 | BR7 | BR6 | BR5 | BR4 | BR3 | BR2 | BR1 | BR0 |
| w | w | w | w | w | w | w | w | w | w | w | w | w | w | w | w |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| BS15 | BS14 | BS13 | BS12 | BS11 | BS10 | BS9 | BS8 | BS7 | BS6 | BS5 | BS4 | BS3 | BS2 | BS1 | BS0 |
| w | w | w | w | w | w | w | w | w | w | w | w | w | w | w | w |

Bits 31:16 **BRy:** Port x reset bit y (y = 0..15)

These bits are write-only and can be accessed in word, half-word or byte mode. A read to these bits returns the value 0x0000.

0: No action on the corresponding ODRx bit

1: Resets the corresponding ODRx bit

*Note: If both BSx and BRx are set, BSx has priority.*

Bits 15:0 **BSy:** Port x set bit y (y= 0..15)

These bits are write-only and can be accessed in word, half-word or byte mode. A read to these bits returns the value 0x0000.

0: No action on the corresponding ODRx bit

1: Sets the corresponding ODRx bit

**PB3 → SET/CLEAR**

```
/*PB3-> SET*/
GPIOB->BSRR |= 1<<3;
/*PB3-> CLEAR*/
GPIOB->BSRR |= 1<<(16 + 3);
```

# PROGRAMANDO DESDE REGISTROS

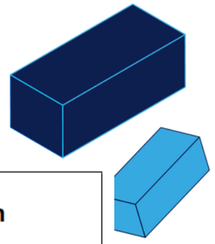
arm

MICRO-
CONTRO-
LADORES
ARM

UMAKER | CENTRO DE CAPACITACIÓN DE DESARROLLO TECNOLÓGICO

# General-purpose I/Os (GPIO)

## (Entrada/Salida de propósito general)



| MODER(i) [1:0] | OTYPER(i) | OSPEEDR(i) [B:A] | PUPDR(i) [1:0] | | I/O configuration | |
|---|---|---|---|---|---|---|
| 01 | 0 | SPEED [B:A] | 0 | 0 | GP output | PP |
| | 0 | | 0 | 1 | GP output | PP + PU |
| | 0 | | 1 | 0 | GP output | PP + PD |
| | 0 | | 1 | 1 | Reserved | |
| | 1 | | 0 | 0 | GP output | OD |
| | 1 | | 0 | 1 | GP output | OD + PU |
| | 1 | | 1 | 0 | GP output | OD + PD |
| | 1 | | 1 | 1 | Reserved (GP output OD) | |

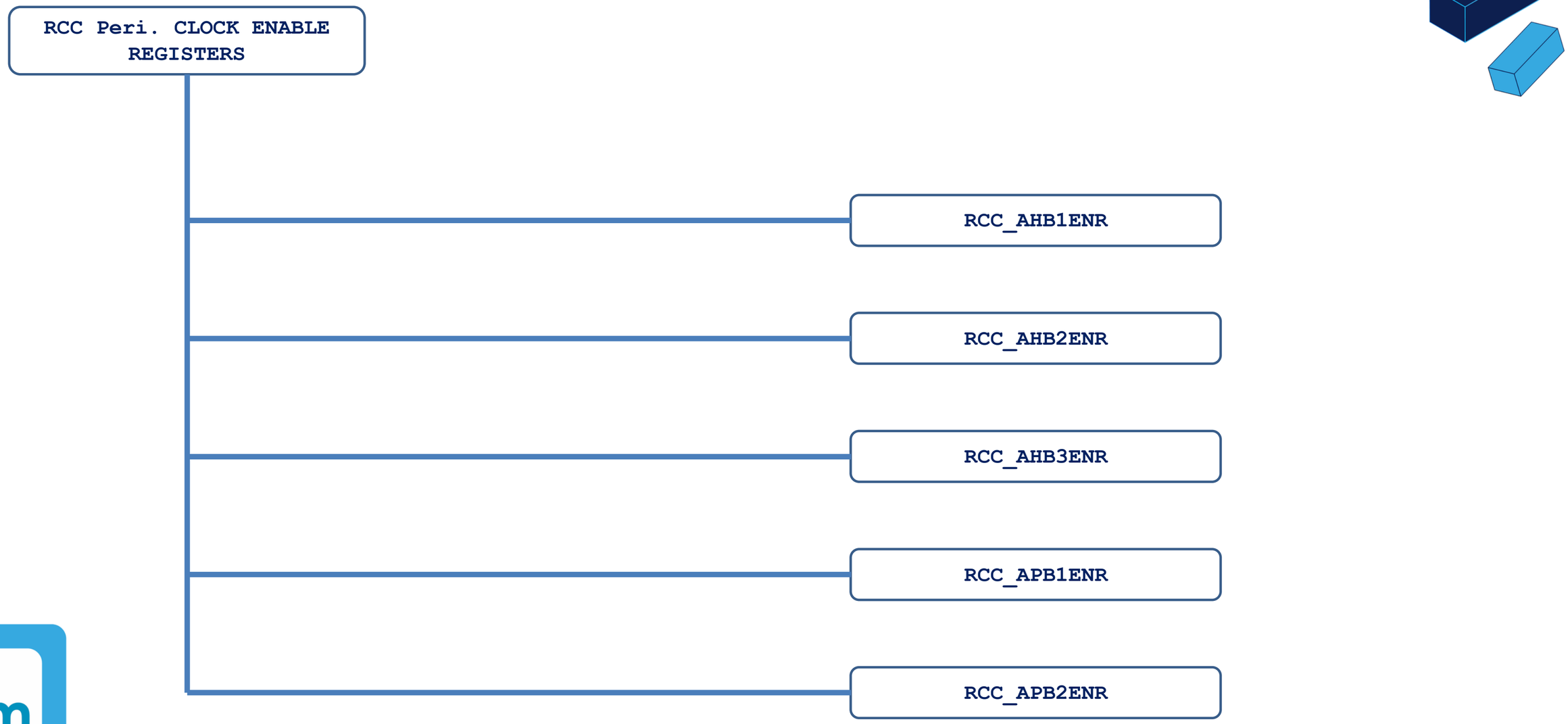
| MODER(i) [1:0] | OTYPER(i) | OSPEEDR(i) [B:A] | | PUPDR(i) [1:0] | | I/O configuration | |
|---|---|---|---|---|---|---|---|
| 10 | 0 | SPEED [B:A] | | 0 | 0 | AF | PP |
| | 0 | | | 0 | 1 | AF | PP + PU |
| | 0 | | | 1 | 0 | AF | PP + PD |
| | 0 | | | 1 | 1 | Reserved | |
| | 1 | | | 0 | 0 | AF | OD |
| | 1 | | | 0 | 1 | AF | OD + PU |
| | 1 | | | 1 | 0 | AF | OD + PD |
| | 1 | | | 1 | 1 | Reserved | |
| 00 | x | x | x | 0 | 0 | Input | Floating |
| | x | x | x | 0 | 1 | Input | PU |
| | x | x | x | 1 | 0 | Input | PD |
| | x | x | x | 1 | 1 | Reserved (input floating) | |
| 11 | x | x | x | 0 | 0 | Input/output | Analog |
| | x | x | x | 0 | 1 | Reserved | |
| | x | x | x | 1 | 0 | | |
| | x | x | x | 1 | 1 | | |

arm

MICRO-CONTRO-LADORES ARM

UMAKER | CENTRO DE CAPACITACIÓN DE DESARROLLO TECNOLÓGICO

# RCC PERI. CLOCK ENABLE REGISTER



# RCC PERI. CLOCK ENABLE REGISTER

## AHB2 peripheral clock enable register (RCC_AHB2ENR)

Address offset: 0x4C
Reset value: 0x0000 0000
Access: no wait state, word, half-word and byte access
Note: When the peripheral clock is not active, the peripheral registers read or write access is not Supported

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | RNG EN | res. | AESEN (1) |
| | | | | | | | | | | | | | rw | | rw |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Res. | res. | ADCEN | res. | Res. | Res. | Res. | res. | GPIOH EN | res. | res. | GPIOE EN | GPIOD EN | GPIOC EN | GPIOB EN | GPIOA EN |
| | | rw | | | | | | rw | | | rw | rw | rw | rw | rw |

1. Available on STM32L42xxx, STM32L44xxx and STM32L46xxx devices only.

**Bit 0 GPIOAEN: IO port A clock enable**
Set and cleared by software.
0: IO port A clock disabled
1: IO port A clock enabled

```
RCC->AHB2ENR |= RCC_AHB2ENR_GPIOAEN;
```

MICRO-CONTRO-LADORES ARM

UMAKER | CENTRO DE CAPACITACIÓN DE DESARROLLO TECNOLÓGICO

# General-purpose I/Os (GPIO)

## (Entrada/Salida de propósito general)

```c
#include "stm32f4xx.h"                  // Device header

#define GPIOA_Base       0x40020000U
#define GPIOA_MODER      *((volatile uint32_t*)(GPIOA_Base + 0x00))
#define GPIOA_OTYPER     *((volatile uint32_t*)(GPIOA_Base + 0x04))
#define GPIOA_OSPEEDR    *((volatile uint32_t*)(GPIOA_Base + 0x08))
#define GPIOA_PUPDR      *((volatile uint32_t*)(GPIOA_Base + 0x0C))
#define GPIOA_ODR        *((volatile uint32_t*)(GPIOA_Base + 0x14))
#define GPIOA_BSRR       *((volatile uint32_t*)(GPIOA_Base + 0x18))

void delayMs(uint32_t delay);

int main(void){

  /*HABILITAR EL RELOJ*/
  RCC->AHB1ENR |= RCC_AHB1ENR_GPIOAEN;
  /*CONFIGURAR EL PA5 COMO SALIDA*/
  GPIOA_MODER &=~(1U<<(11) | 1U<<10);
  GPIOA_MODER |= 1U<<10;            //salida de proposito general
  GPIOA_OTYPER &=~(1U<<5);          //PUSH-PULL
  GPIOA_OSPEEDR |= 1U<<10;          //MEDIUM SPEED
  GPIOA_PUPDR &=~(1U<<(11) | 1U<<10);//no pull/down


  while(1){
    GPIOA_ODR |=1<<5;
    delayMs(100);
    GPIOA_ODR &=~(1U<<5);
    delayMs(200);
    GPIOA_BSRR |= 1<<5;
    delayMs(200);
    GPIOA_BSRR |= 1<<(16 + 5);
    delayMs(200);
  }
}


void delayMs(uint32_t delay){
  uint32_t i;
  SysTick->LOAD = SystemCoreClock/1000;
  SysTick->VAL = 0;
  SysTick->CTRL |= 1<<2 | 1<<0;
  for(i = 0;i<delay;i++){
    while(!(SysTick->CTRL & 1<<16));
  }
}
```

**USANDO POR COMPLETO LA CMSIS**

```c
#include "stm32f4xx.h"                  // Device header

void delayMs(uint32_t delay);

int main(void){
  /*HALITTAR EL RELOJ*/
  RCC->AHB1ENR |= RCC_AHB1ENR_GPIOAEN;
  /*CONFIGURACION PA5*/
  GPIOA->MODER &=~(1U<<(11) | 1U<<10);
  GPIOA->MODER |= 1U<<10;            //salida de proposito general
  GPIOA->OTYPER &=~(1U<<5);          //PUSH-PULL
  GPIOA->OSPEEDR |= 1U<<10;          //MEDIUM SPEED
  GPIOA->PUPDR &=~(1U<<(11) | 1U<<10);//no pull/down
  while(1){
    GPIOA->ODR |= 1<<5;
    delayMs(100);
    GPIOA->ODR &=~(1U<<5);
    delayMs(100);
  }
}

void delayMs(uint32_t delay){
  uint32_t i;
  SysTick->LOAD = SystemCoreClock/1000;
  SysTick->VAL = 0;
  SysTick->CTRL |= 1<<2 | 1<<0;
  for(i = 0;i<delay;i++){
    while(!(SysTick->CTRL & 1<<16));
  }
}
```

arm
MICRO-CONTRO-LADORES ARM

UMAKER | CENTRO DE CAPACITACIÓN DE DESARROLLO TECNOLÓGICO

# General-purpose I/Os (GPIO)

## (Entrada/Salida de propósito general)

USANDO POR COMPLETO LA CMSIS

```c
#include <stdint.h>
#include "stm32l4xx.h"

int main(void)
{
    uint8_t PA0_Value;
    RCC->AHB2ENR |= RCC_AHB2ENR_GPIOAEN;
    /*configuracion de modo*/
    GPIOB->MODER &=~ GPIO_MODER_MODE3;
    GPIOB->MODER |= 0x1U<<2*3;
    GPIOA->MODER &=~ GPIO_MODER_MODE0;
    /*tipo de salida*/
    GPIOB->OTYPER &=~ 0x1U<<3;
    /*Velocidad de conmutacion*/
    GPIOB->OSPEEDR |= 0x3U<<2*3;
    /*resistencia pull up/down*/
    GPIOB->PUPDR &=~ GPIO_MODER_MODE3;
    GPIOA->PUPDR &=~ GPIO_PUPDR_PUPD0;
    GPIOA->PUPDR |= 0x2U<<2*0;
    /*PB3->SET*/
    GPIOB->ODR |= 1U<<3;
    /*PB3->CLEAR*/
    GPIOB->ODR &=~1U<<3;
    /*READ VALUE PA0*/
    PA0_Value = (GPIOA->IDR>>0 & 0x1);
    /*PB3-> SET*/
    GPIOB->BSRR |= 1<<3;
    /*PB3-> CLEAR*/
    GPIOB->BSRR |= 1<<(16 + 3);

    /* Loop forever */
    for(;;);
}
```

arm

MICRO-CONTRO-LADORES ARM

UMAKER | CENTRO DE CAPACITACIÓN DE DESARROLLO TECNOLÓGICO

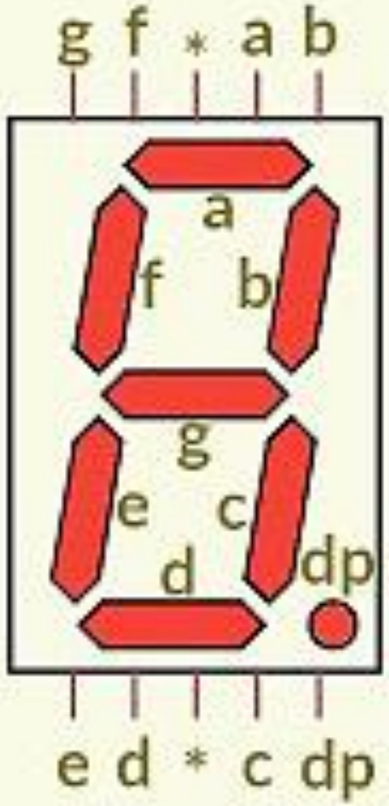
# MANEJO DE DISPLAY DE 7 SEGMENTOS

arm

MICRO-
CONTRO-
LADORES
ARM

UMAKER | CENTRO DE CAPACITACIÓN DE DESARROLLO TECNOLÓGICO

# DISPLAY 7 SEGMENTOS CATADO COMUN



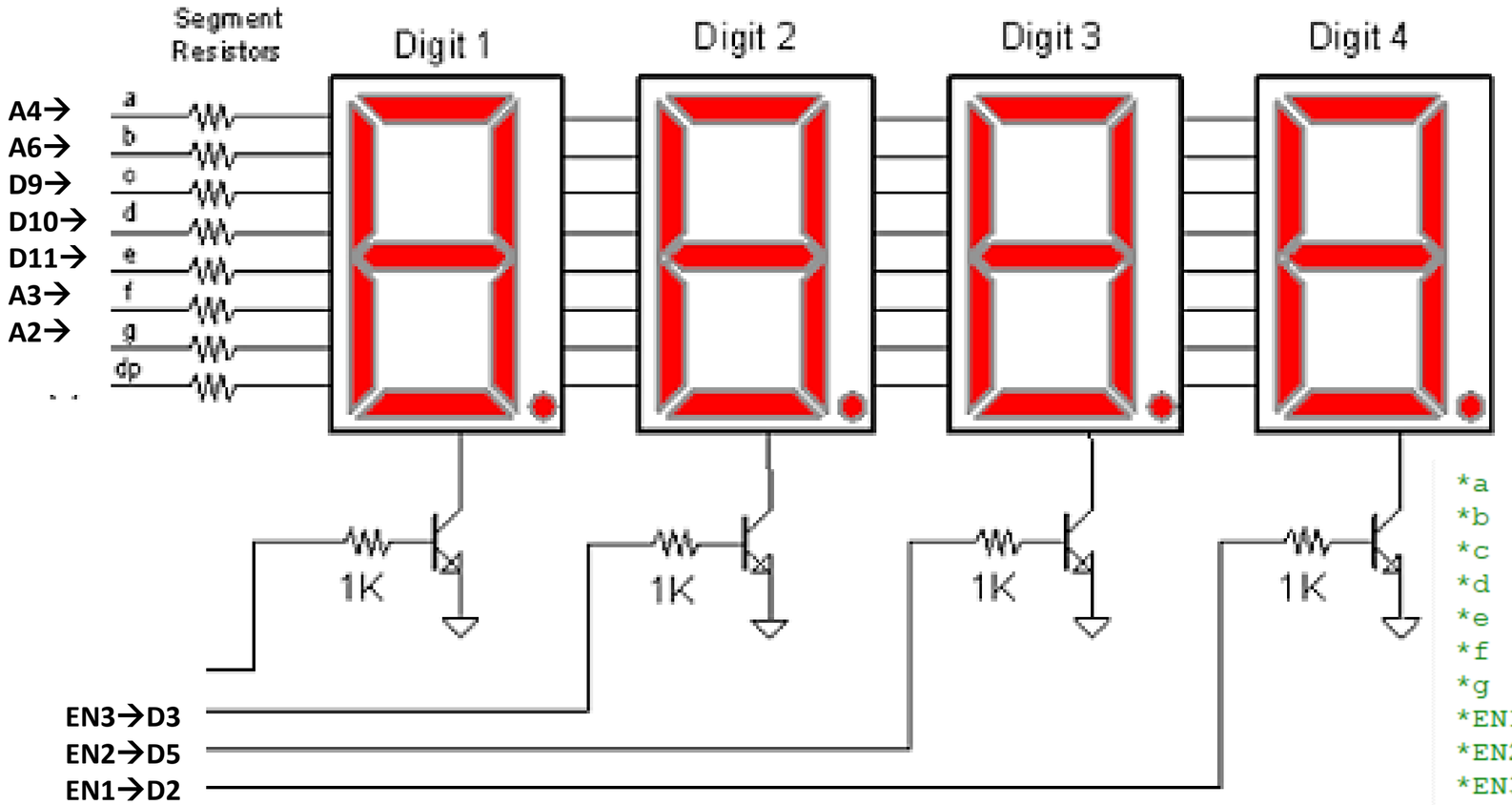

| Cátodo Cómun | | | | | | | | | HEX |
|---|---|---|---|---|---|---|---|---|---|
| Común* | Número | g | f | e | d | c | b | a | |
| GND | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0x3F |
| GND | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0x06 |
| GND | 2 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 0x5B |
| GND | 3 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 0x4F |
| GND | 4 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0x66 |
| GND | 5 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 0x6D |
| GND | 6 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 0x7D |
| GND | 7 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0x07 |
| GND | 8 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0x7F |
| GND | 9 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 0x67 |

arm
MICRO-CONTRO-LADORES ARM

UMAKER | CENTRO DE CAPACITACIÓN DE DESARROLLO TECNOLÓGICO

# DISPLAY 7 SEGMENTOS CATADO COMUN

## MULTIPLEXACION




```
*a    ->A4(PA5)
*b    ->A6(PA7)
*c    ->D9(PA8)
*d    ->D10(PA11)
*e    ->D11(PB5)
*f    ->A3(PA4)
*g    -> A2(PA3)
*EN1  ->D2(PA12)
*EN2  ->D5(PB6)
*EN3  ->D3(PB0)
*Push button-> D6 (PB1)
```

MICRO-CONTRO-LADORES ARM

UMAKER | CENTRO DE CAPACITACIÓN DE DESARROLLO TECNOLÓGICO

UMAKER | CENTRO DE CAPACITACIÓN DE DESARROLLO TECNOLÓGICO