



PUCP

PROGRAMACIÓN DE MICROCONTROLADORES ARM

CETAM - PUCP



Sesión 4 - 11/07/2024:

- Microcontroladores ARM
- Punteros
- Puertos GPIO
- Ejemplos de aplicación

arm Developer

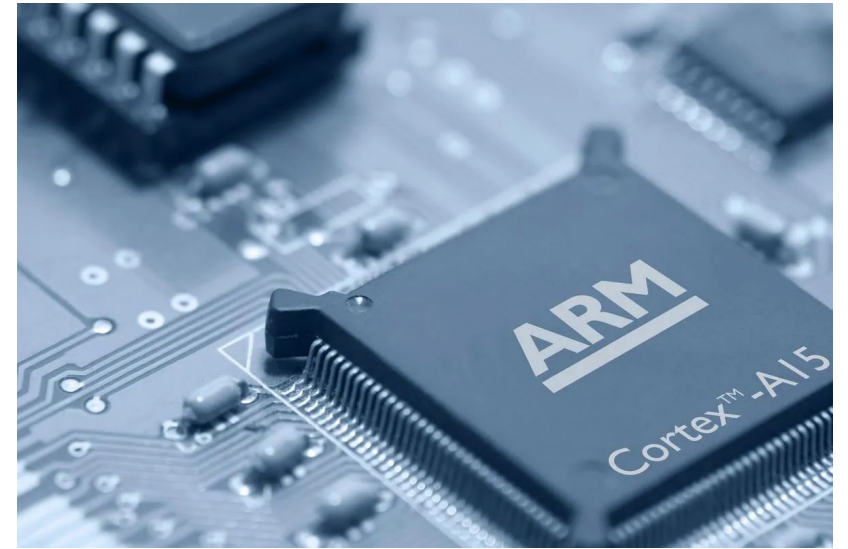
Microcontroladores ARM

Programación de microcontroladores ARM - Sesión 4



¿Qué es un procesador ARM?

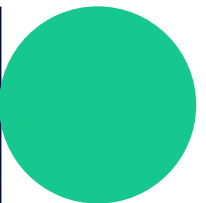
Un procesador ARM es un sistema digital programable que ejecuta instrucciones lógicas utilizando instrucciones RISC, a diferencia de procesadores INTEL o AMD que utilizan instrucciones CISC



Instrucciones CISC

Las instrucciones CISC tienen la siguiente estructura :

- Código de operación
- Operandos fuente (OP1, OP2)
- Operando destino o resultado (OPd)
- Dirección de la instrucción siguiente (IS)



Instrucciones RISC

Las instrucciones RISC tienen la siguiente estructura :

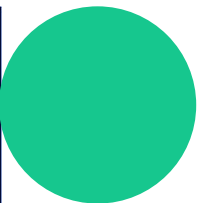
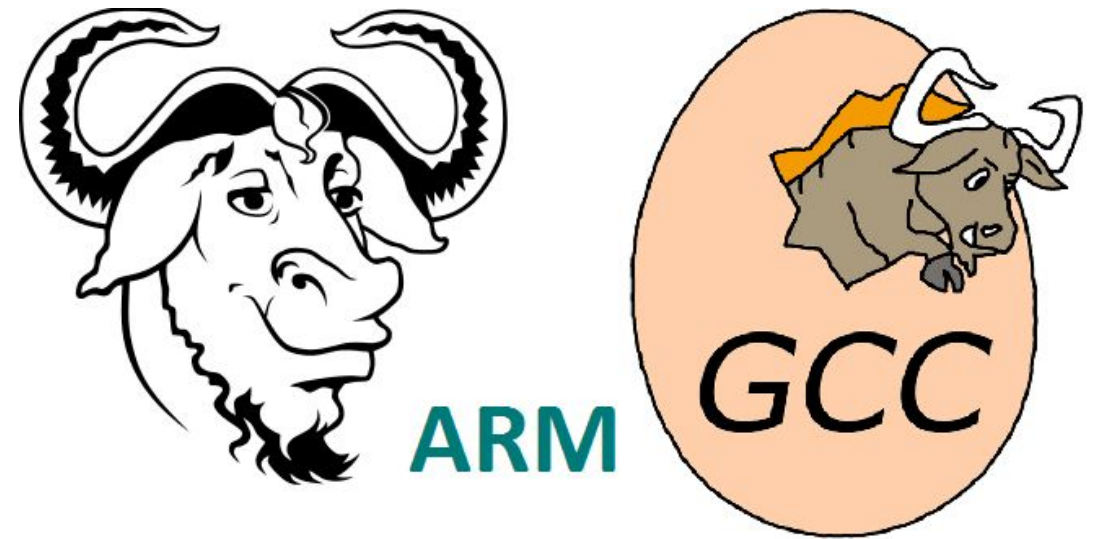
- Código de operación (CO)
- Operandos fuente (OP1, OP2)
- Operando destino o resultado (OPd)

CO	OP1	OP2	OPd
----	-----	-----	-----



GNU ARM y GNU GCC

- GNU - GCC .- Es la base del compilador para el lenguaje de programación C y traduce hacia el código ensamblador para ordenadores con arquitectura x86, x64, etc
- GNU - ARM .- Utiliza el set de instrucciones RISC y traduce el lenguaje C para dicha estructura



Cortex - A, Cortex - R, Cortex - M

Cortex - A

Highest performance

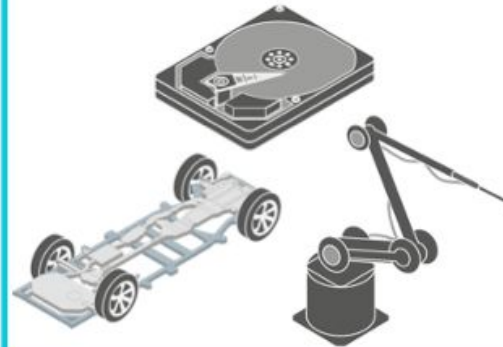
Optimised for
rich operating systems



Cortex - R

Fast response

Optimised for
high performance,
hard real-time applications



Cortex - M

Smallest/lowest power

Optimised for
discrete processing and
microcontrollers

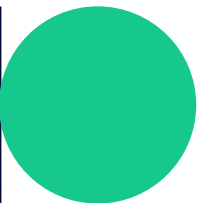


Diagrama de bloques de un microcontrolador

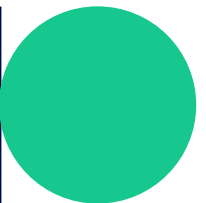
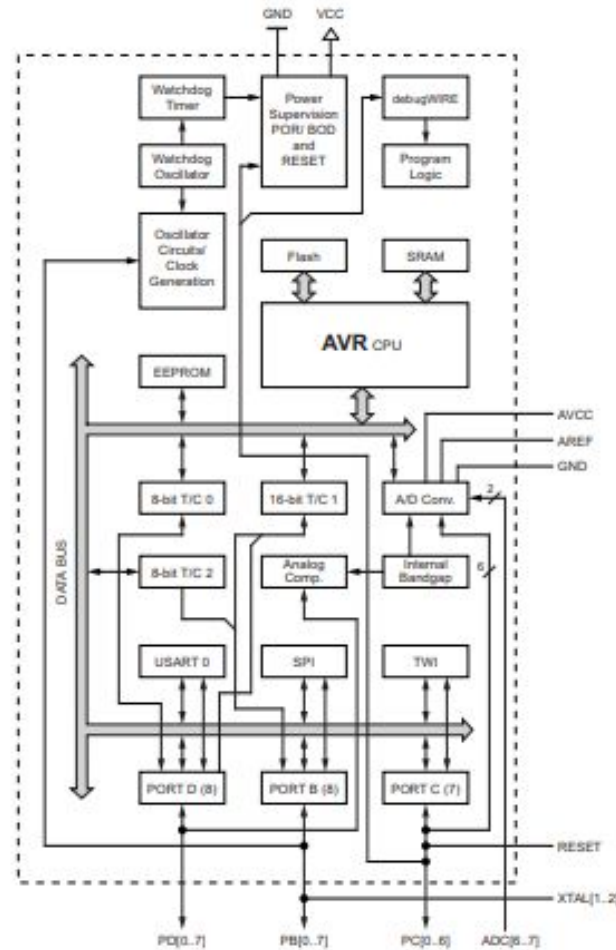


Diagrama de bloques de un microcontrolador

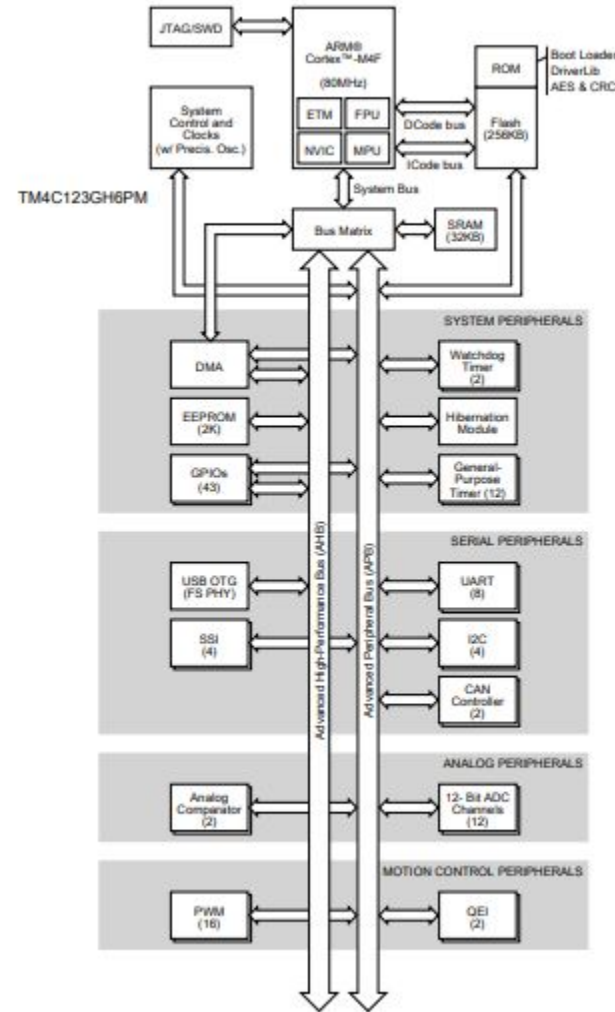
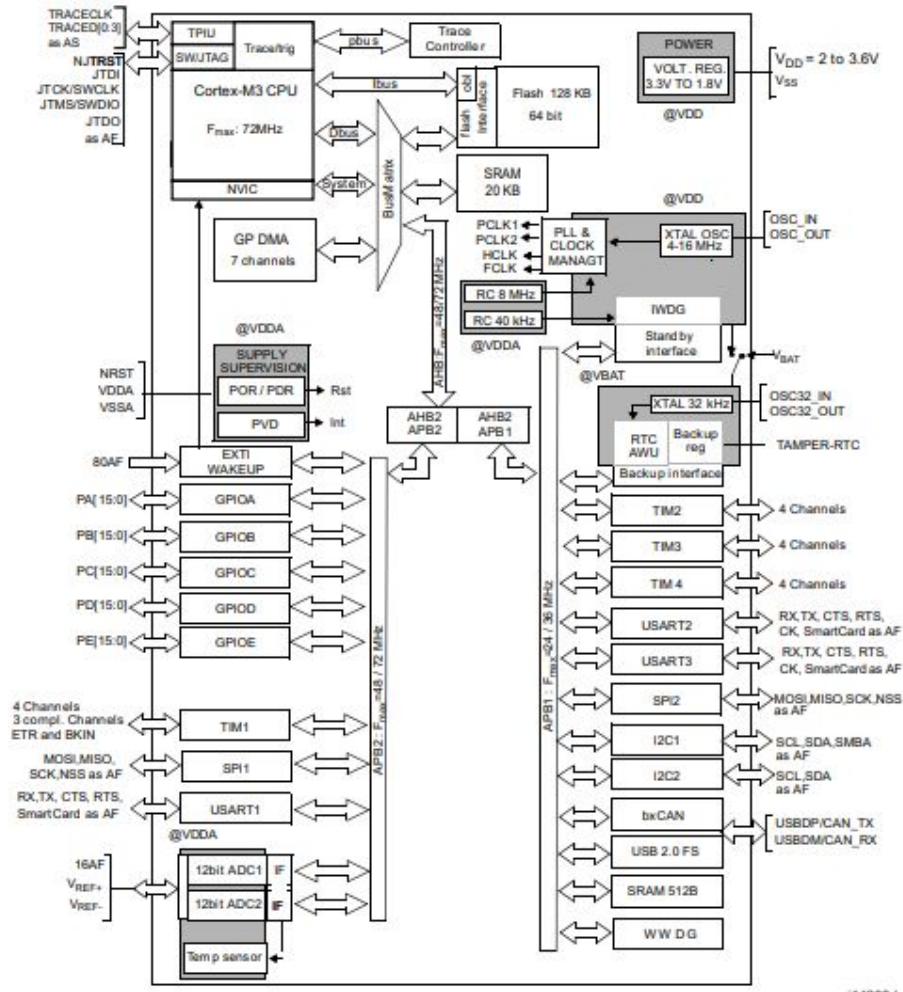
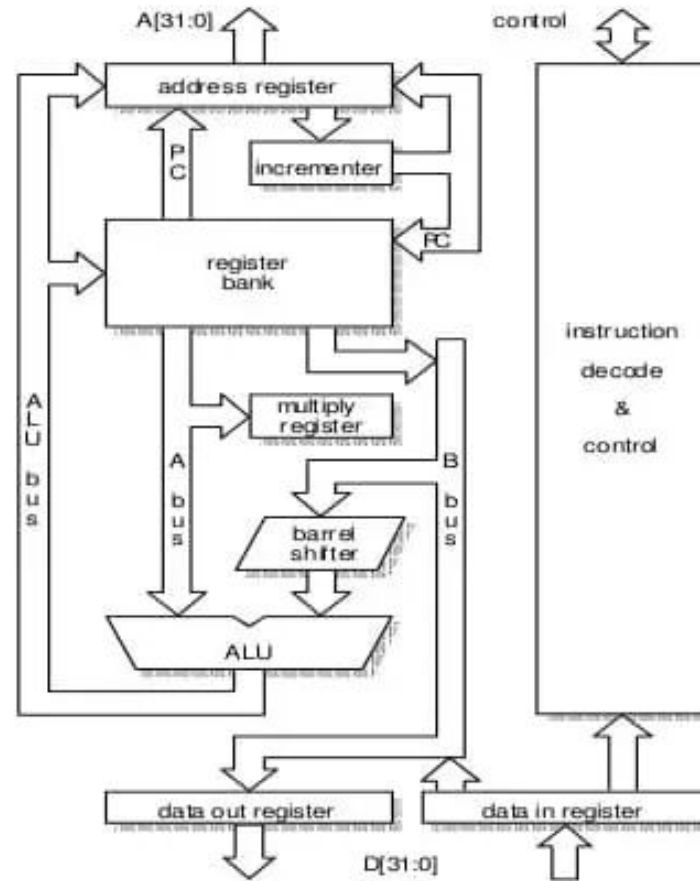


Diagrama de bloques de un microcontrolador



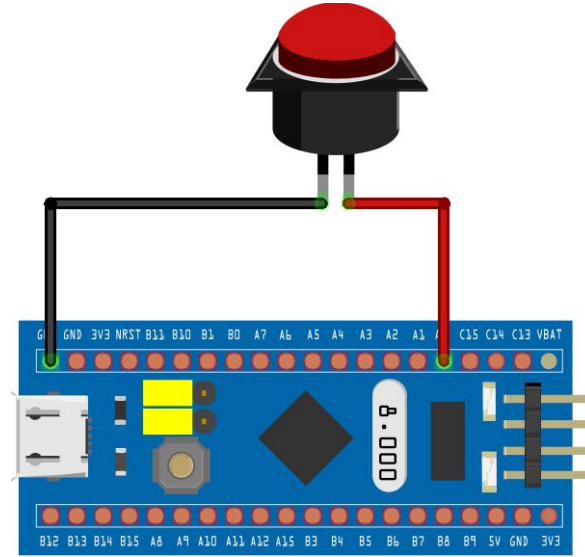
Arquitectura ARM



Enmascaramiento y desplazamiento de bits

- Colocar el valor de 1 en ciertos bits
 - $R0 = R0 \mid 0b000001100;$
 - $R0 = R0 \mid 0b000000110;$
- Colocar el valor de 0 en ciertos Bits
 - $R1 = R1 \& 0b11110011;$
 - $R1 = R1 \& 0b11111110;$
- Desplazamientos de bits hacia la derecha y hacia la izquierda
 - $R1 = 7 \ll 2;$
 - $R2 = 2 \ll 4;$
 - $R3 = 48 \gg 2$





Introducción a GPIO

Programación de microcontroladores ARM - Sesión 3



Programación de alto nivel en un microcontrolador

- Integración de instrucciones en funciones predefinidas
- Alto entendimiento del código
- Bajo performance
- Baja posibilidad de cometer errores, pero de encontrarse errores internos, estos serán muy difíciles de encontrar.

```
1 #include "mbed.h"
2
3 DigitalInOut led1(LED1);
4
5 int main() {
6     while (true) {
7         led1 = 1;
8         wait(0.5);
9
10        led1 = 0;
11        wait(0.5);
12    }
```



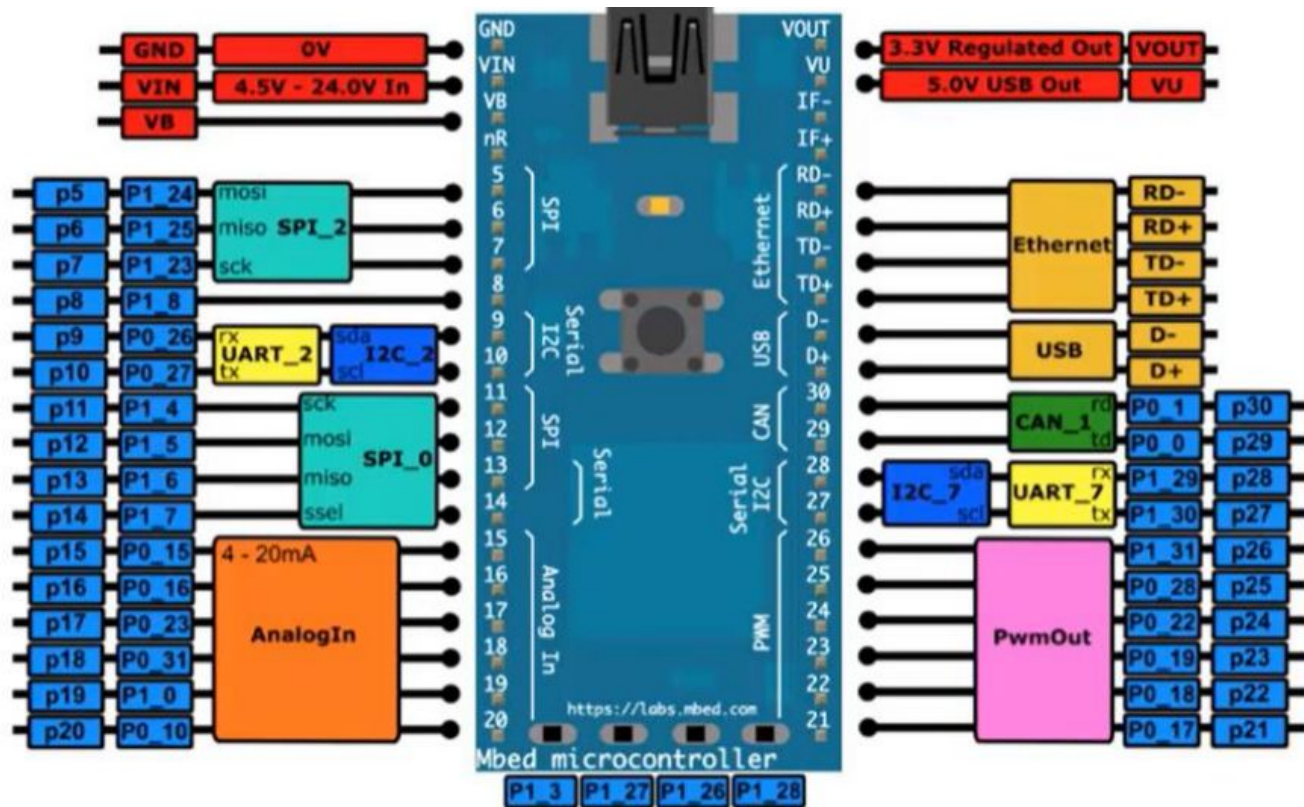
Voltajes y valores lógicos

Los valores lógicos en el código son traducidos a voltajes eléctricos que interactúan con otros dispositivos electrónicos. Es importante revisar que todos los dispositivos electrónicos integrados en una red puedan entenderse adecuadamente a través de los voltajes apropiados

Logic	Voltage	Boolean	Circuit	Switch
'1'	3.3V	True	Closed	On
'0'	0V	False	Open	Off

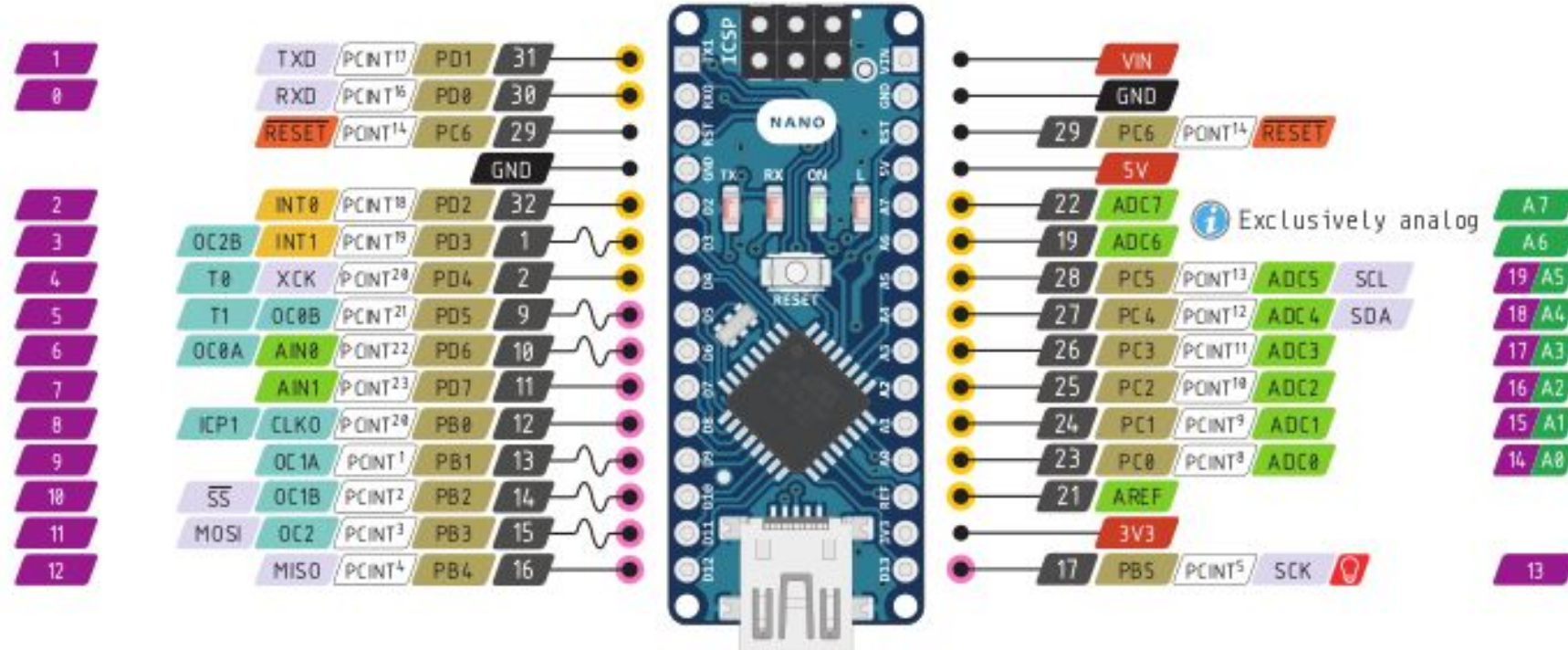


Puertos en un microcontrolador: Simulador MBED



Pin	Pin name in Mbed API
Switch 1	p5
Switch 2	p6
Switch 3	p7
Switch 4	p8
Red LED	p9
Blue LED	p10
Yellow LED	p11

Puertos en un microcontrolador: Arduino Nano



Puertos en un microcontrolador: Tiva Launchpad



LaunchPad with LM4F120H5QR
LaunchPad with TM4C123GH6PM
Revision 1

Flash	256	KB
SRAM	32	KB

Serial	hardware
ADC	12 bits

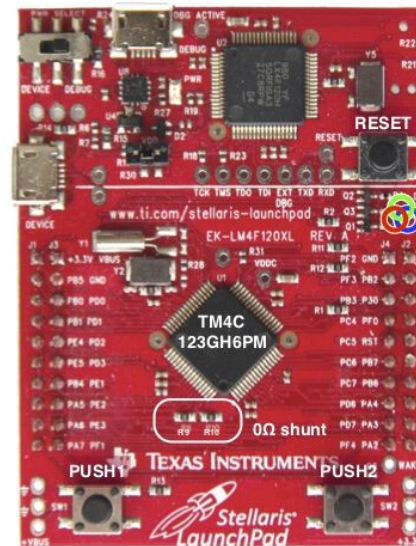
Use pins numbers only!

Quadrature Encoder · TTL level
only on TM4C123GH6PM

				J1	J3				
		+3.3V		1	21		+5V		
CS (2)		A11	PB 5	2	22		n		
RX (1)			PB 0	3	23	PD 0	A7	SCL (3)	SCK (3)
TX (1)			PB 1	4	24	PD 1	A6	SDA (3)	CS (3)
RX (5)	SCL (2)	A9	PE 4	5	25	PD 2	A5		MISO (3)
TX (5)	SDA (2)	A8	PE 5	6	26	PD 3	A4	Index (0)	MOSI (3)
SCK (2)		A10	PB 4	7	27	PE 1	A2		TX (7)
MOSI (0)			PA 5	8	28	PE 2	A1		
	SCL (1)		PA 6	9	29	PE 3	A0		
	SDA (1)		PA 7	10	30	PF_1	RED_LED	PhB (0)	MOSI (1)
						GROUND			
						GROUND			
						VBUS			

			0Ω shunt		
23	SCL (3)	PD_0	R9	PB_6	14
24	SDA (3)	PD_1	R10	PB_7	15

Remove shunts for compatibility



Hardware
Pin number
Other pin number

I²C

Serial UART

SPI

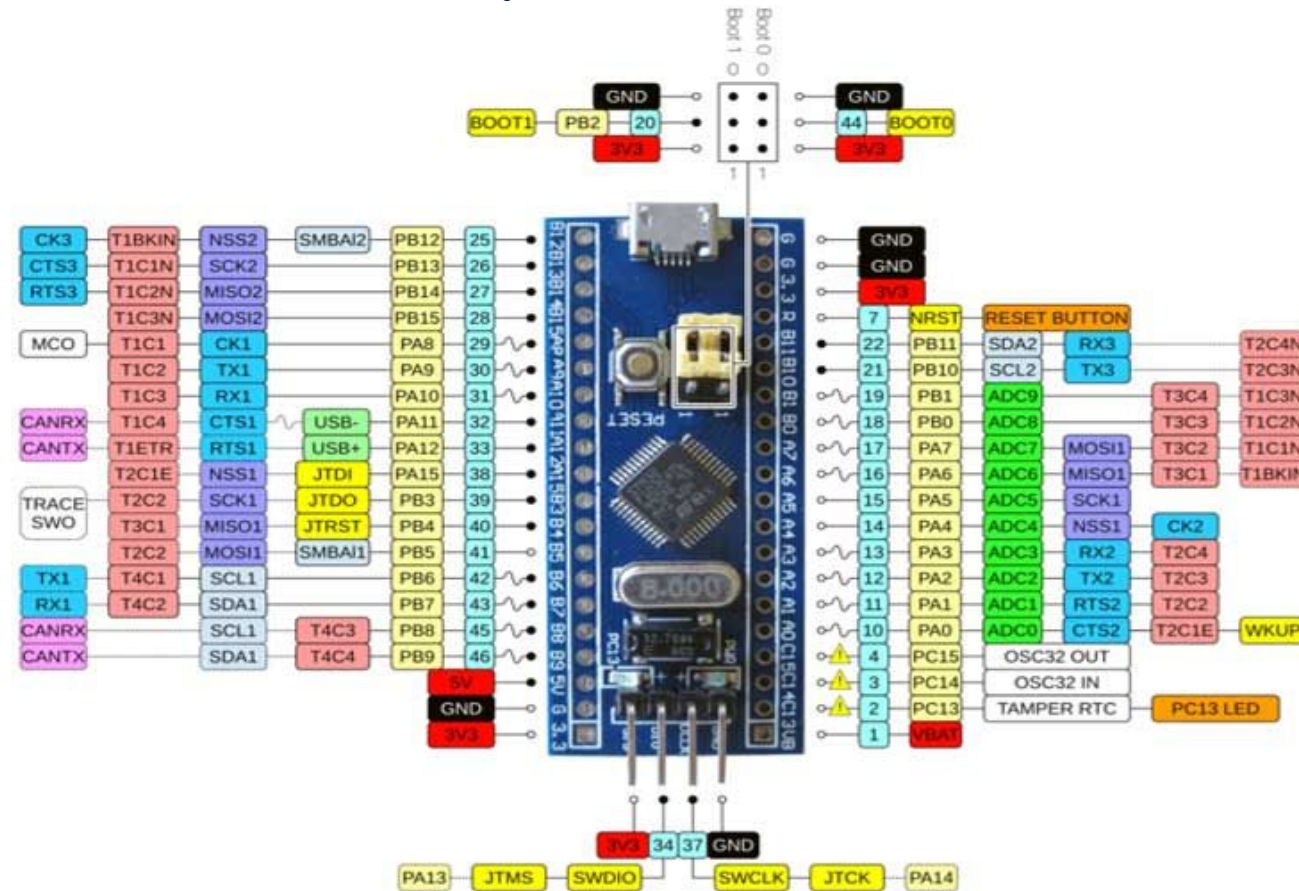
- analogRead()
- digitalRead() and digitalWrite()
- digitalRead(), digitalWrite() and analogWrite()

				J4	J2				
SCK (1)		BLUE_LED	PF_2	40	20	GROUND			
CS (1)		GREEN_LED	PF_3	39	19	PB_2		SCL (0)	
	SDA (0)		PB_3	38	18	PE_0	A3		RX (7)
RX (1)	Index (1)		PC_4	37	17	PF_0	PUSH2		MISO (1)
TX (1)	PhA (1)		PC_5	36	16		RESET		
RX (3)	PhB (1)		PC_6	35	15	PB_7		SDA (3)	MOSI (2)
TX (3)			PC_7	34	14	PB_6		SCL (3)	MISO (2)
RX (2)	PhA (0)		PD_6	33	13	PA_4			MISO (0)
TX (2)		Detection	PD_7	32	12	PA_3			CS (0)
		PUSH1	PF_4	31	11	PA_2			SCK (0)
		GROUND			PA_0				RX (0)
		GROUND			PA_1				TX (0)
		+3.3V			PD_4				RX (6)
					PD_5				TX (0)

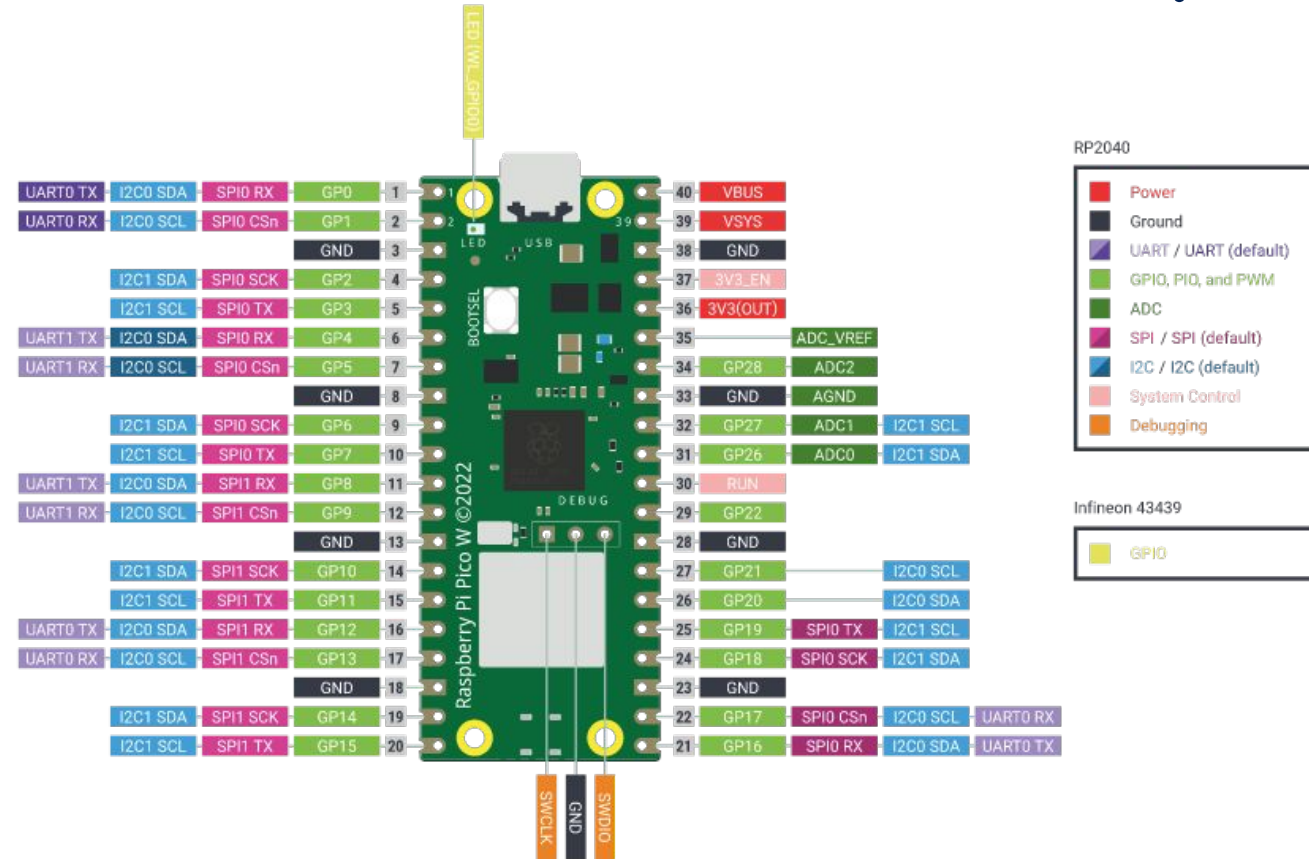
VBUS detection PD_7

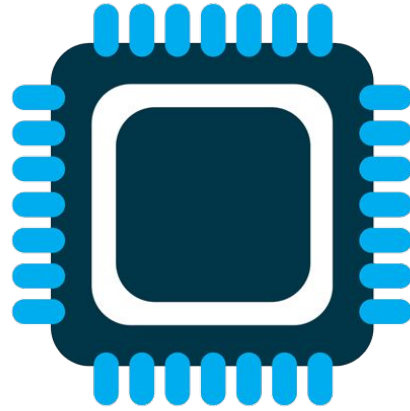
 Rei Vilo, 2012-2017
embeddedcomputing.weebly.com
version 2.1 2015-12-19

Puertos en un microcontrolador: Blue Pill



Puertos en un microcontrolador: Raspberry pi pico





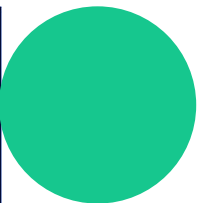
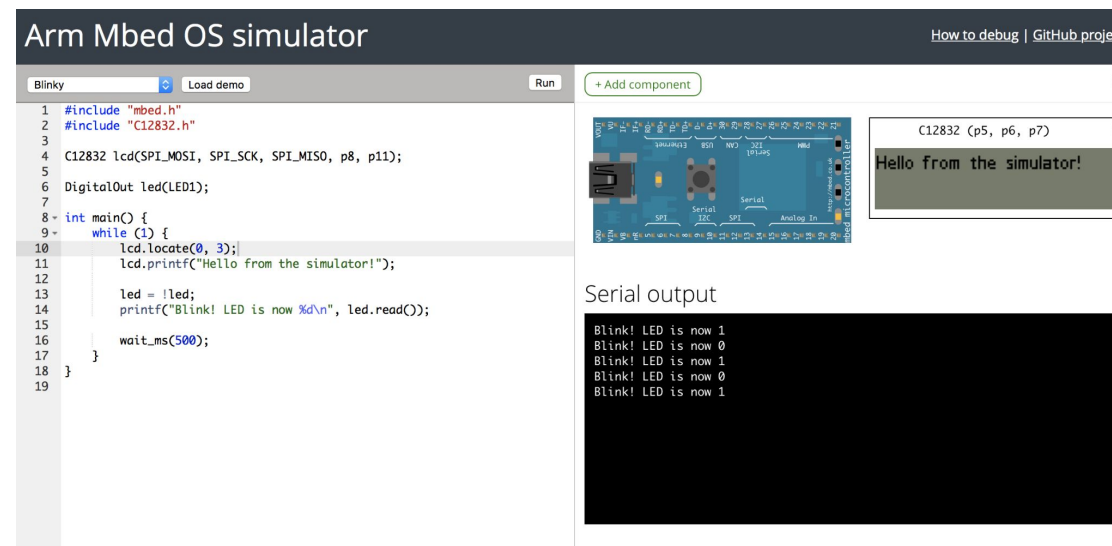
Ejemplos de aplicación

Programación de microcontroladores ARM - Sesión 4



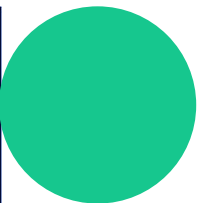
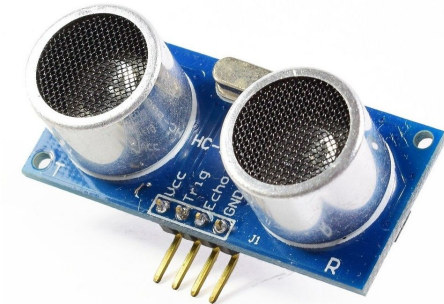
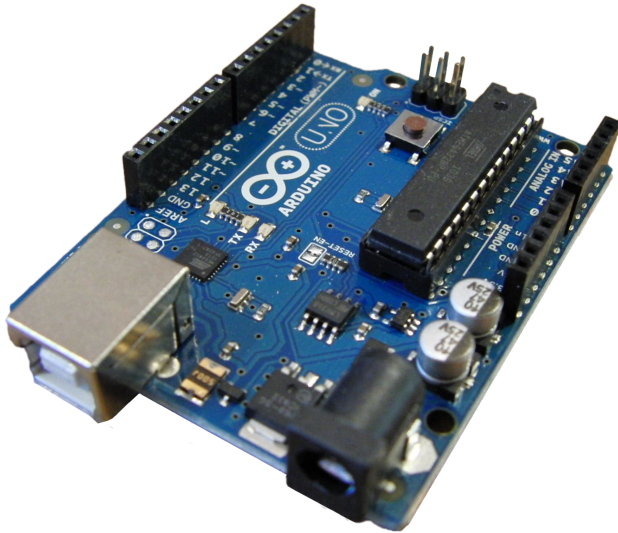
Programación de alto nivel

En MBED se tiene un simulador interesante para poder revisar los conceptos de uso de los periféricos E/S. En esta ocasión se revisarán los principales ejemplos.



Programación de alto nivel

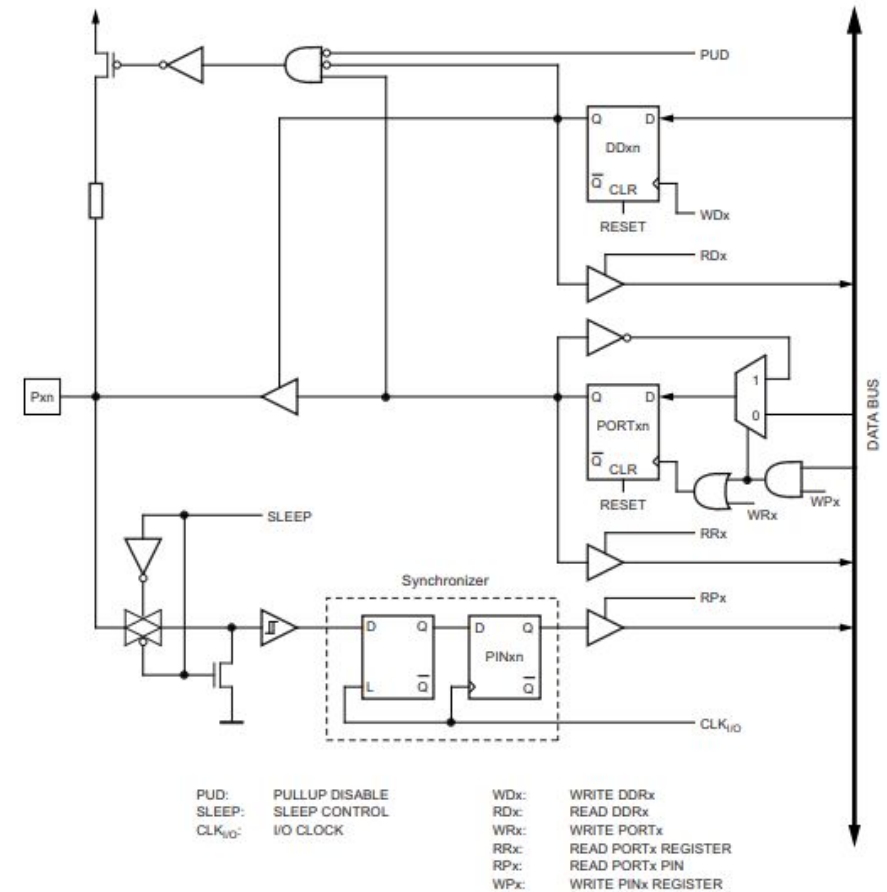
Desarrollar un programa para el microcontrolador ATMEGA 328P que recoja los datos de un sensor ultrasónico y que los muestre en la terminal serie. Utilice programación de alto nivel



Salidas GPIO - ATMEGA328P

Para el microcontrolador ATMEGA328P se tienen 3 registros en memoria que controlan el funcionamiento de los puertos I/O:

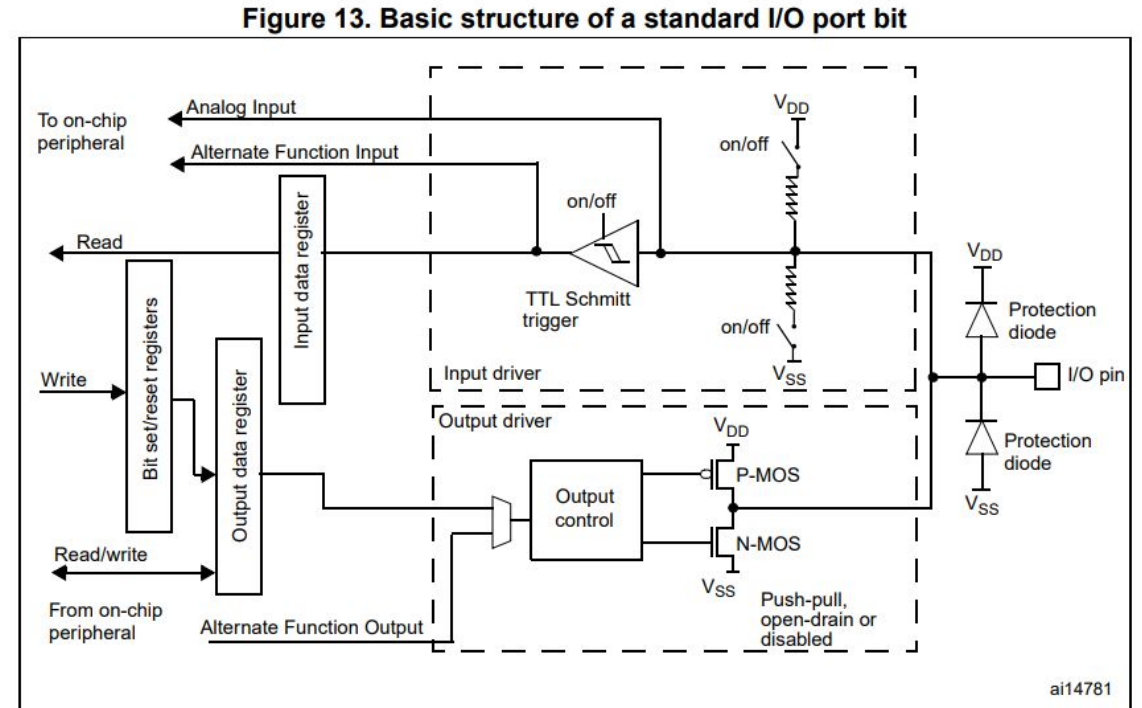
- DDRx
- PORTx
- PINx



Salidas GPIO - STM32F103C8T6

Para el microcontrolador STM32F103C8T6 se tienen 4 registros en memoria que controlan el funcionamiento de los puertos I/O:

- GPIO_CRH
- GPIO_CRL
- GPIO_ORD
- GPIO_IDR



Salidas GPIO - TM4C123GH6PM

Para el microcontrolador TM4C123GH6PM se tienen 4 registros en memoria que controlan el funcionamiento de los puertos I/O:

- GPIO_DIR
- GPIO_DEN
- GPIO_DR8D
- GPIO_DATA

