

Sistemas Basados en Microprocesador

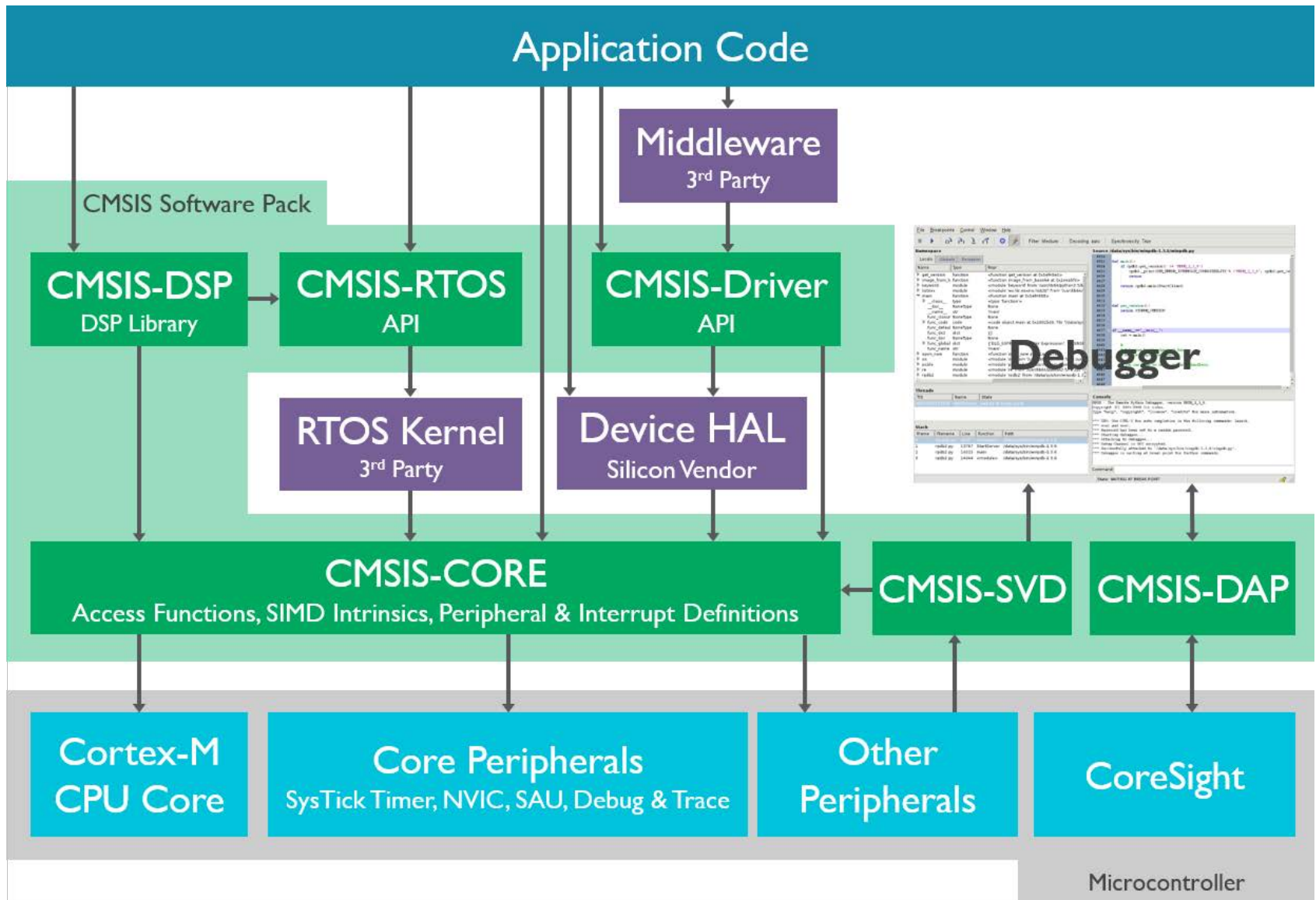
B2 CMSIS DRIVER

CMSIS Driver

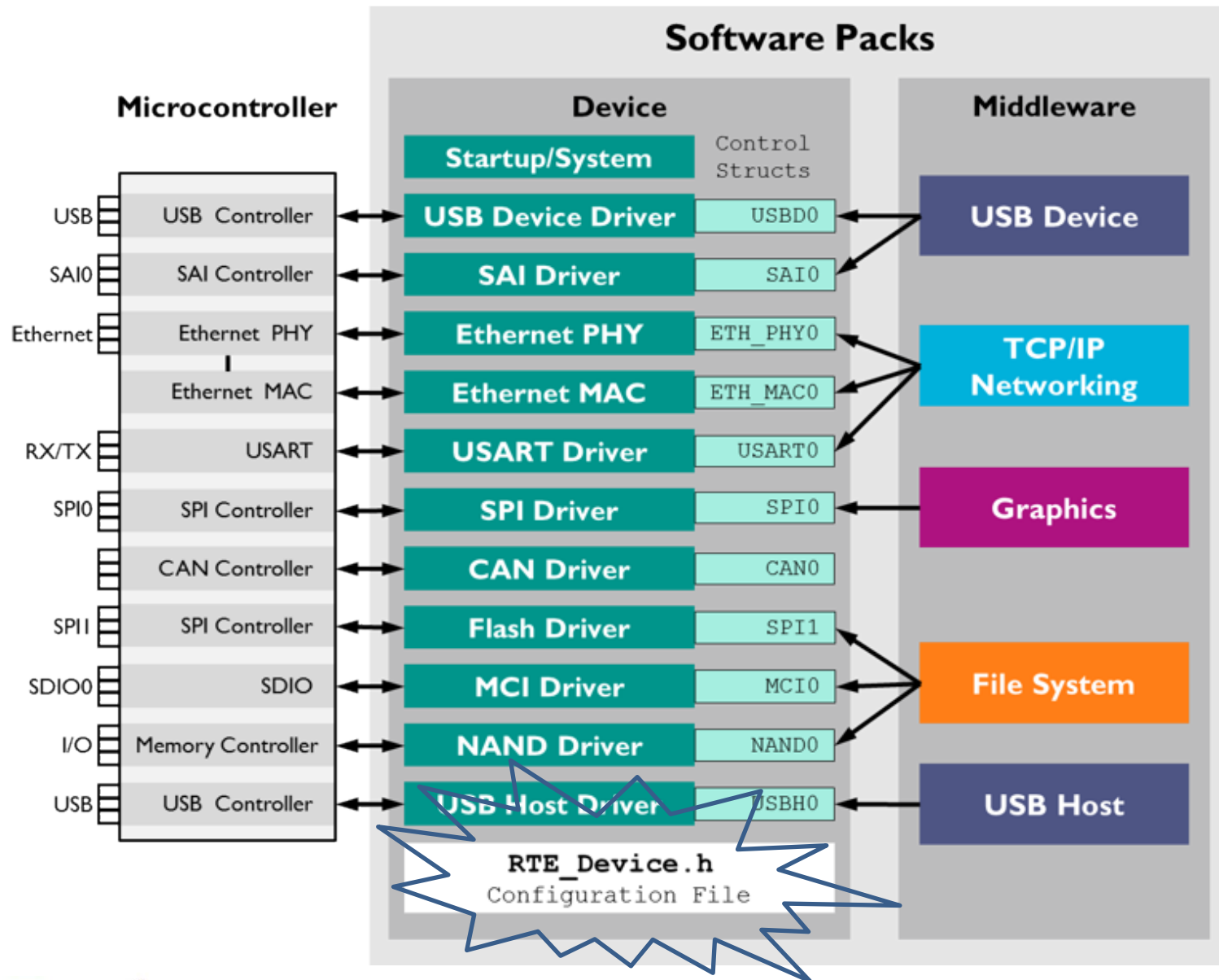
- ¿Qué es un Driver?
 - Elemento software que se encarga de gestionar las funciones realizadas por un hardware
- ¿Qué es una API?
 - API: Application Program Interface
 - Especificación de las funciones que se incluyen en una librería (o en un código) para realizar un conjunto de operaciones
- CMSIS-Driver: Estandarización en el uso y manejo de los periféricos del microcontrolador con una API
- Todos los driver de dispositivos tienen las mismas funciones con objeto de homogeneizar su uso

<http://www.keil.com/pack/doc/CMSIS/Driver/html/index.html>

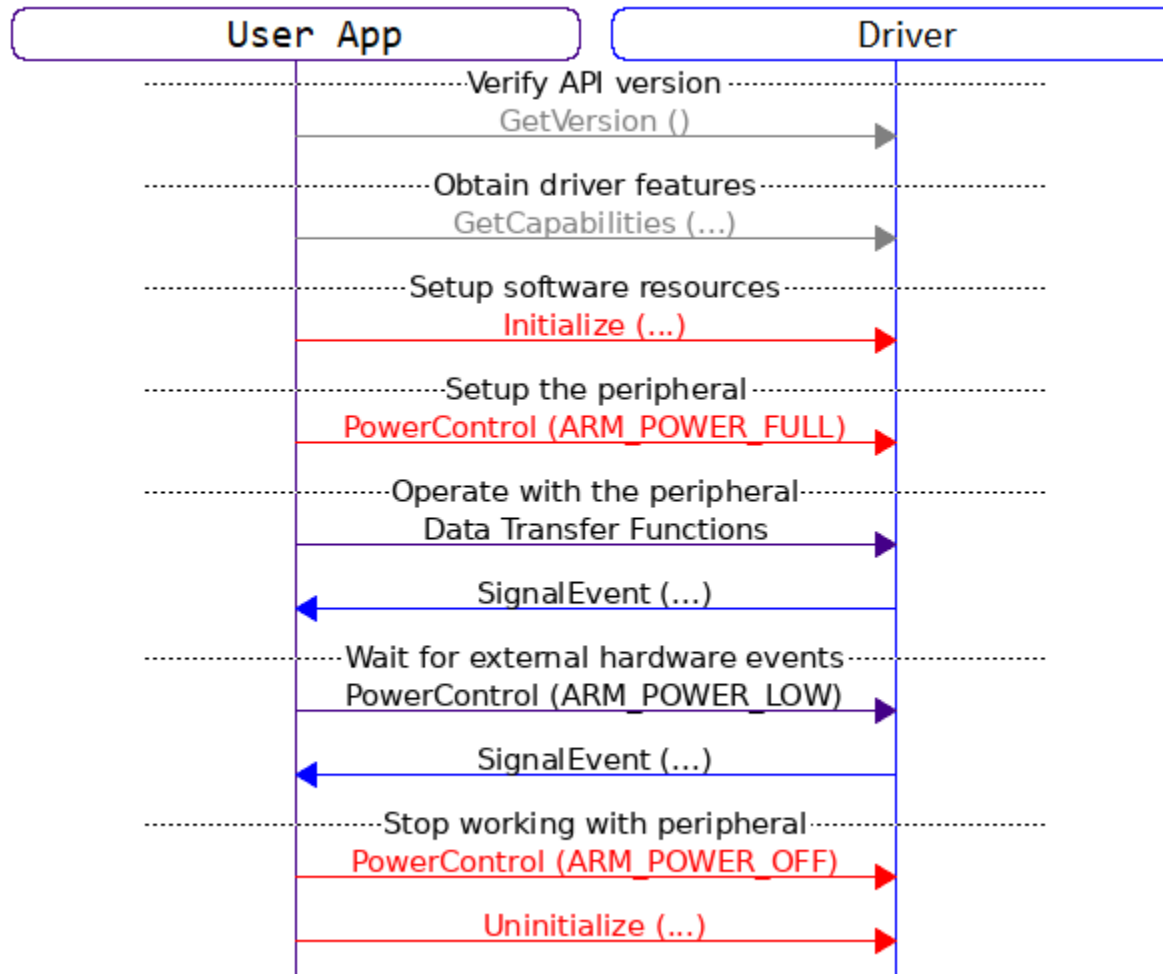
CMSIS



CMSIS-Driver: Periféricos soportados



Secuencia de uso de un driver



Ejemplo de secuencia de operaciones

1. SPIdrv->Initialize
2. SPIdrv->PowerControl (ON)
3. SPIdrv->Control (Parámetros)
4. SPIdrv->Send
5. SPIdrv->Receive (opcional)
6. SPIdrv->PowerControl (OFF)
7. SPIdrv->Uninitialize

API estandarizada por CMSIS-Driver

Función	Significado
GetVersion:	Devuelve la información de la versión del driver
GetCapabilities	Devuelve información acerca de las opciones soportadas por el driver
Initialize	<p>Función que configura los pines a utilizar por el periférico y su CLK, si es necesario. Los pines se configuran usando PULL-UP.</p> <p>Esta función debe ser la primera en ejecutarse (a excepción de las dos anteriores)</p> <p>A esta función se le puede pasar como parámetro el nombre de una función que se ejecutará cuando se produzca un evento asociado con el periférico (callback)</p>
GetStatus	Devuelve información acerca del estado del periférico. Se suele consultar tras una operación de transferencia de información.

API estandarizada por CMSIS-Driver

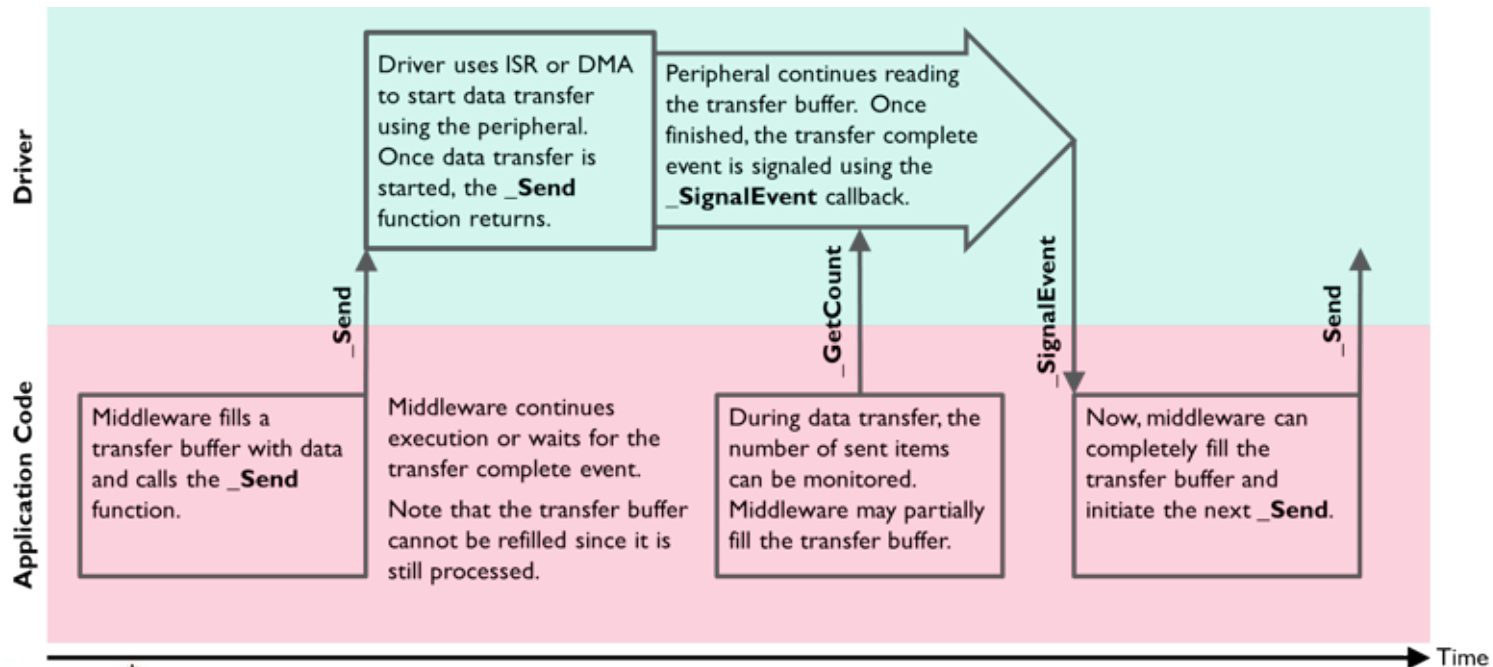
Función	Significado
PowerControl	<p>Función que configura el funcionamiento en términos de consumo del periférico. Hay tres posibles valores para el parámetro de entrada:</p> <ul style="list-style-type: none">•ARM_POWER_FULL: Peripheral is turned on and fully operational. The driver initializes the peripheral registers, interrupts, and (optionally) DMA.•ARM_POWER_LOW: (optional) Peripheral is in low power mode and partially operational; usually, it can detect external events and wake-up.•ARM_POWER_OFF: Peripheral is turned off and not operational (pending operations are terminated). This is the state after device reset
Uninitialize	Última función a utilizar que libera los recursos utilizados
Control	Función que permite configurar el funcionamiento del periférico con valores (<i>define's</i>) que conforman la palabra de configuración

Transferencia en CMSIS-Driver

Función	Significado
Send	Envía datos al periférico
Receive	Recibe datos del periférico
Transfer	Operación combinada de escritura/lectura

Transferencia de información no bloqueante

Utilización del callback/eventos asociados a la instancia del driver



Transferencia en CMSIS-Driver

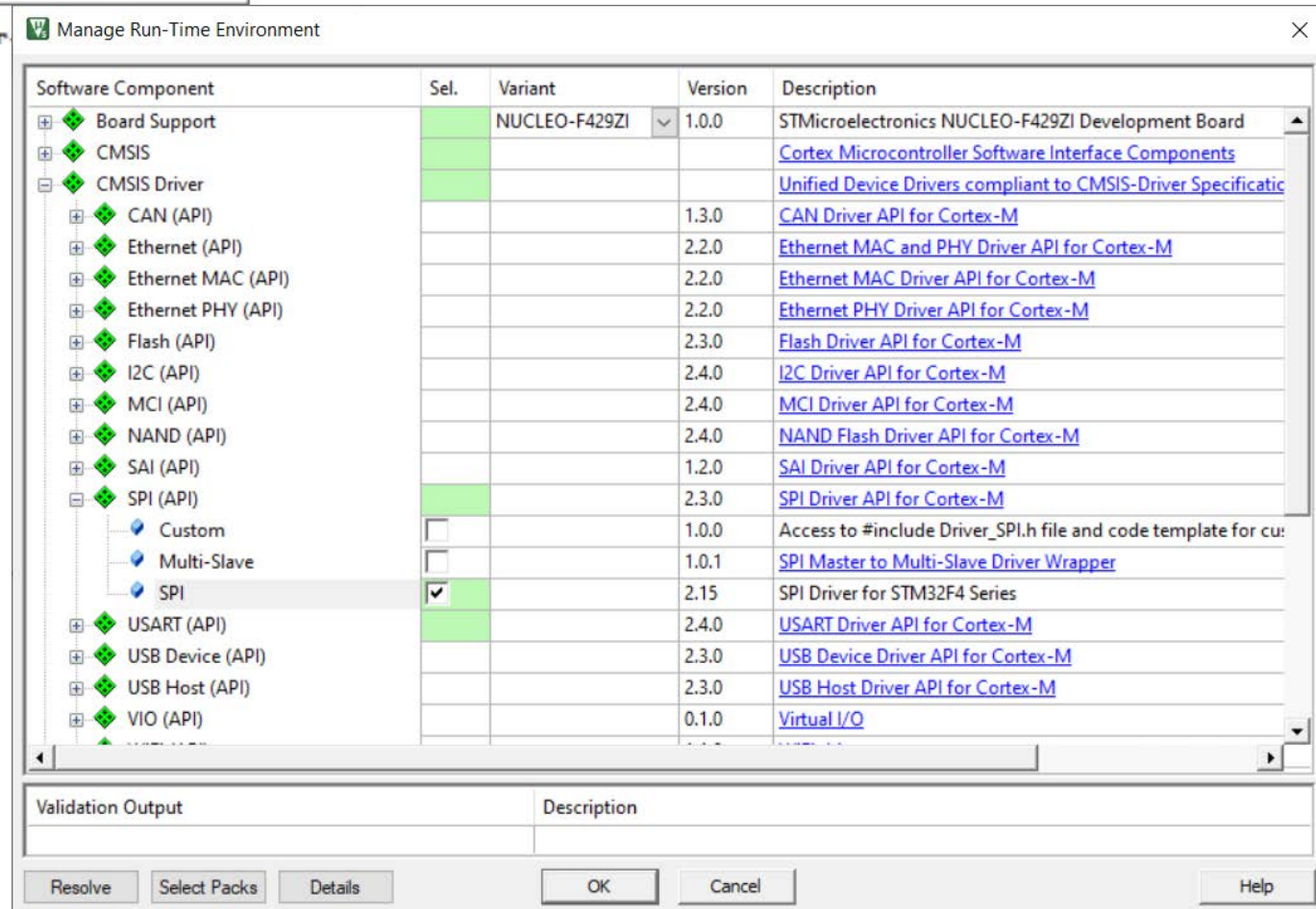
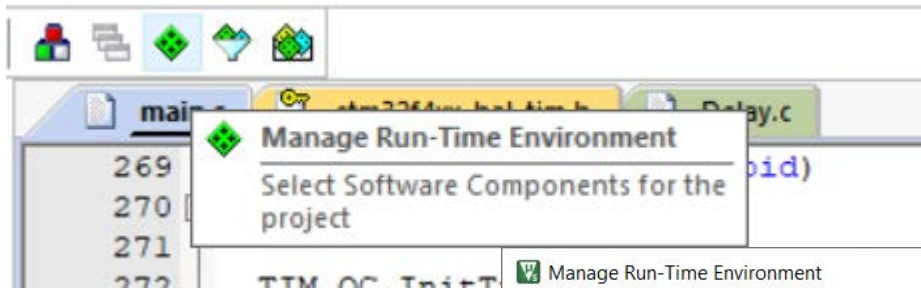
Función	Significado
Send	Envía datos al periférico
Receive	Recibe datos del periférico
Transfer	Operación combinada de escritura/lectura

Transferencia de información bloqueante:

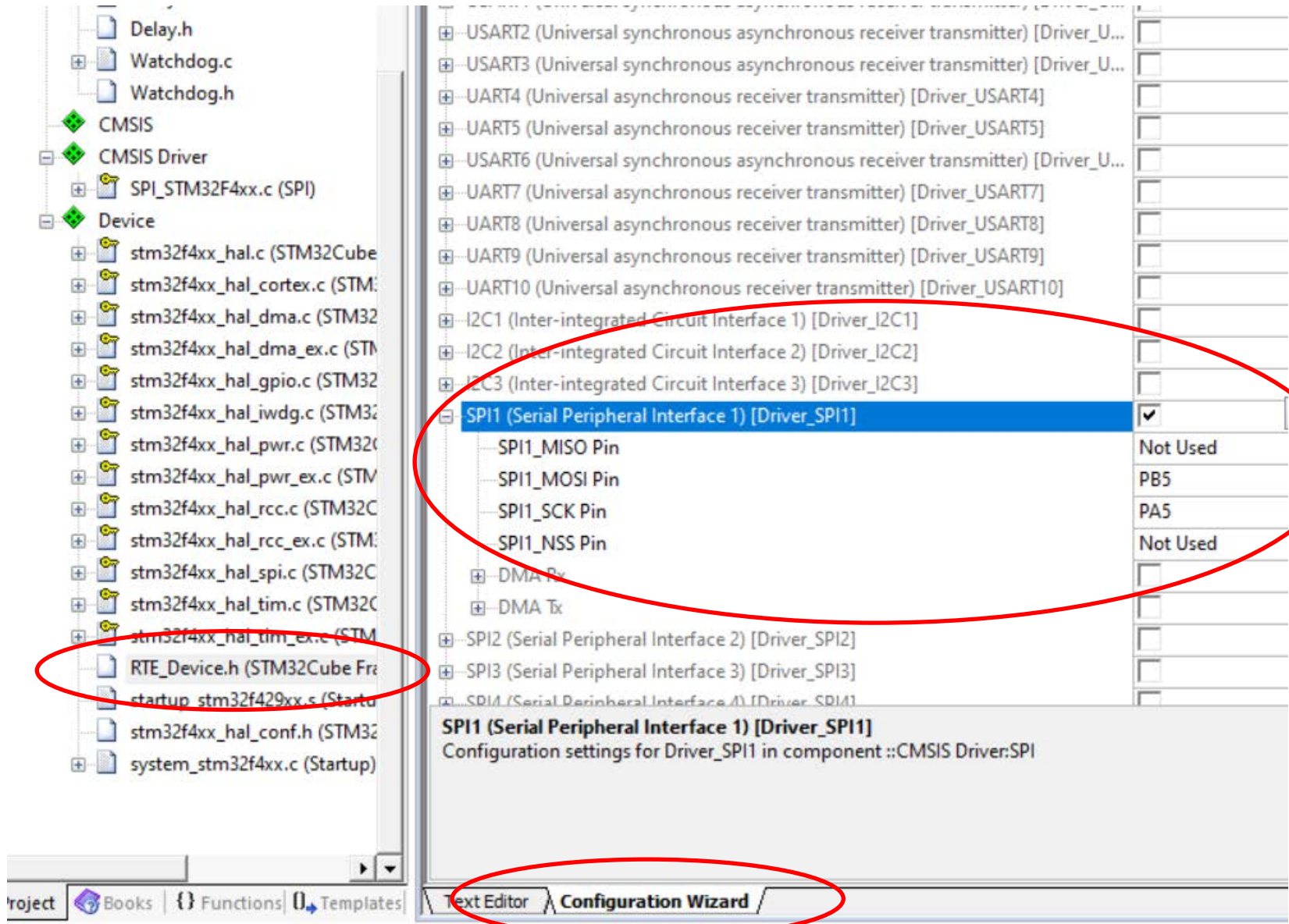
```
stat = SPIDrv->Send(&buf, sizeof(buf));  
  
do  
{  
    stat = SPIDrv->GetStatus();  
}  
while (stat.busy);
```



Añadiendo Drivers CMSIS (RTE)



Configuración del sistema con RTE_Device.h



¿Cómo se maneja el driver desde una aplicación en C?

- CMSIS define esta estructura de datos. Para acceder a ella se deben añadir estas sentencias en el programa:
 - extern [ARM_DRIVER_SPI](#) Driver_SPI<n>;
 - [ARM_DRIVER_SPI](#)* SPIDrv = &Driver_SPI<n>;

```
typedef struct _ARM_DRIVER_SPI {  
    ARM\_DRIVER\_VERSION (*GetVersion) (void);  
    ARM\_SPI\_CAPABILITIES (*GetCapabilities) (void);  
    int32_t (*Initialize) (ARM\_SPI\_SignalEvent\_t cb_event);  
    int32_t (*Uninitialize) (void);  
    int32_t (*PowerControl) (ARM\_POWER\_STATE state);  
    int32_t (*Send) (const void *data, uint32_t num);  
    int32_t (*Receive) (void *data, uint32_t num);  
    int32_t (*Transfer) (const void *data_out, void  
        *data_in, uint32_t num);  
    uint32_t (*GetDataCount) (void);  
    int32_t (*Control) (uint32_t control, uint32_t arg);  
    ARM\_SPI\_STATUS (*GetStatus) (void);  
} const ARM\_DRIVER\_SPI;
```

**n debe ser el número del
interface seleccionado en
el fichero de configuración
RTE_device.h**

Sistemas Basados en Microprocesador

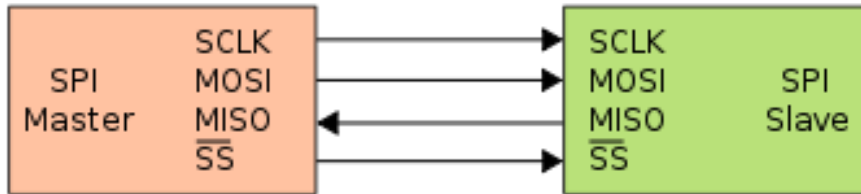
B2 SPI (Serial Peripheral Interface Bus)

SPI Características

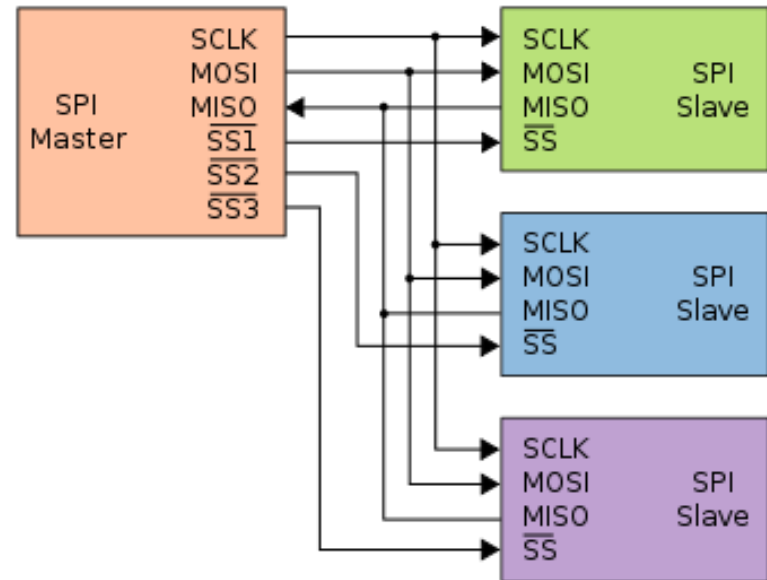
- Bus síncrono, serie, con comunicación bidireccional simultanea (Full Duplex)
- SPI maestro/esclavo
- Se envía un dato en cada pulso de reloj
- De 8 a 16 bits por transferencia

SPI – Ejemplo Arquitectura

Bus SPI: un maestro y un esclavo.



SPI bus: un maestro y tres esclavos.



Señales en el interfaz SPI

- MISO: Master In / Slave Out data. This pin can be used to transmit data in slave mode and receive data in master mode.
- MOSI: Master Out / Slave In data. This pin can be used to transmit data in master mode and receive data in slave mode.
- SCK: Serial Clock output for SPI masters and input for SPI slaves.
- NSS: Slave select. This is an optional pin to select a slave device. This pin acts as a 'chip select' to let the SPI master communicate with slaves individually and to avoid contention on the data lines. Slave NSS inputs can be driven by standard IO ports on the master device. The NSS pin may also be used as an output if enabled (SSOE bit) and driven low if the SPI is in master configuration. In this manner, all NSS pins from devices connected to the Master NSS pin see a low level and become slaves when they are configured in NSS hardware mode. When configured in master mode with NSS configured as an input (MSTR=1 and SSOE=0) and if NSS is pulled low, the SPI enters the master mode fault state: the MSTR bit is automatically cleared and the device is configured in slave mode (refer to [Section 28.3.10](#)).

Señalización

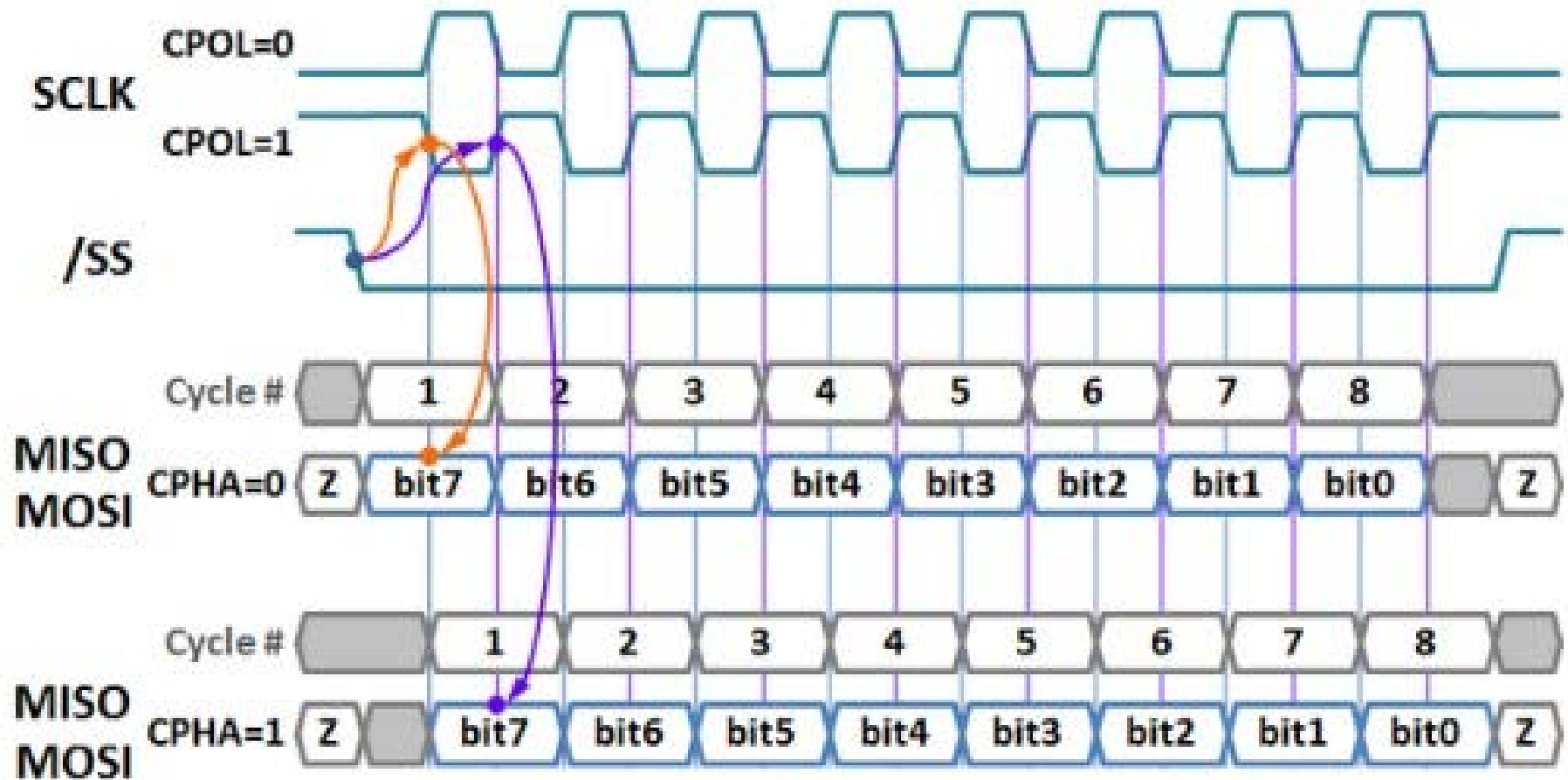
- La polaridad del reloj (flanco de subida o bajada) es configurable con el bit CPOL en el registro de control.
- Se puede programar cuándo comienza la transmisión del primer bit

Table 360. SPI Data To Clock Phase Relationship

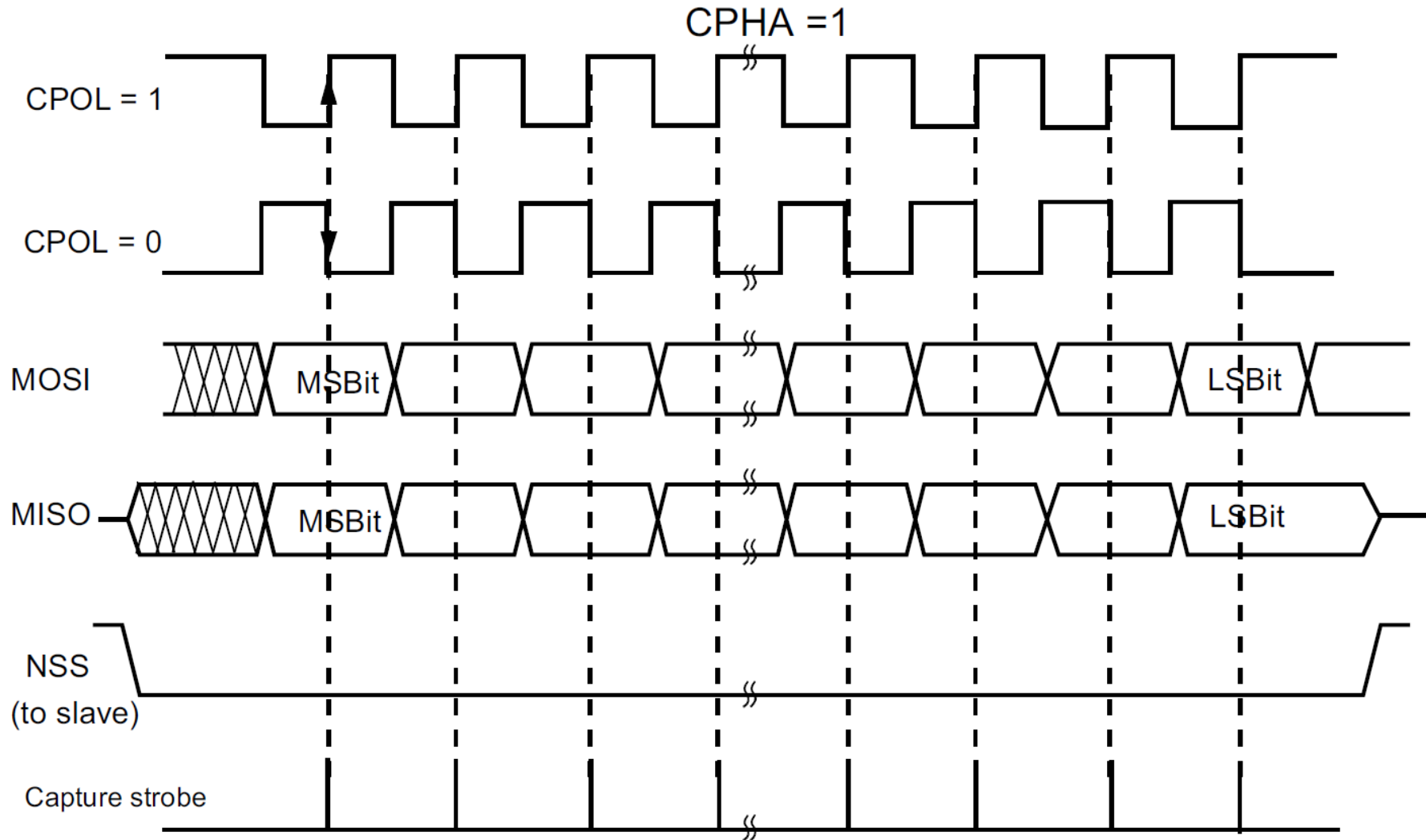
CPOL and CPHA settings	When the first data bit is driven	When all other data bits are driven	When data is sampled
CPOL = 0, CPHA = 0	Prior to first SCK rising edge	SCK falling edge	SCK rising edge
CPOL = 0, CPHA = 1	First SCK rising edge	SCK rising edge	SCK falling edge
CPOL = 1, CPHA = 0	Prior to first SCK falling edge	SCK rising edge	SCK falling edge
CPOL = 1, CPHA = 1	First SCK falling edge	SCK falling edge	SCK rising edge

Cronogramas

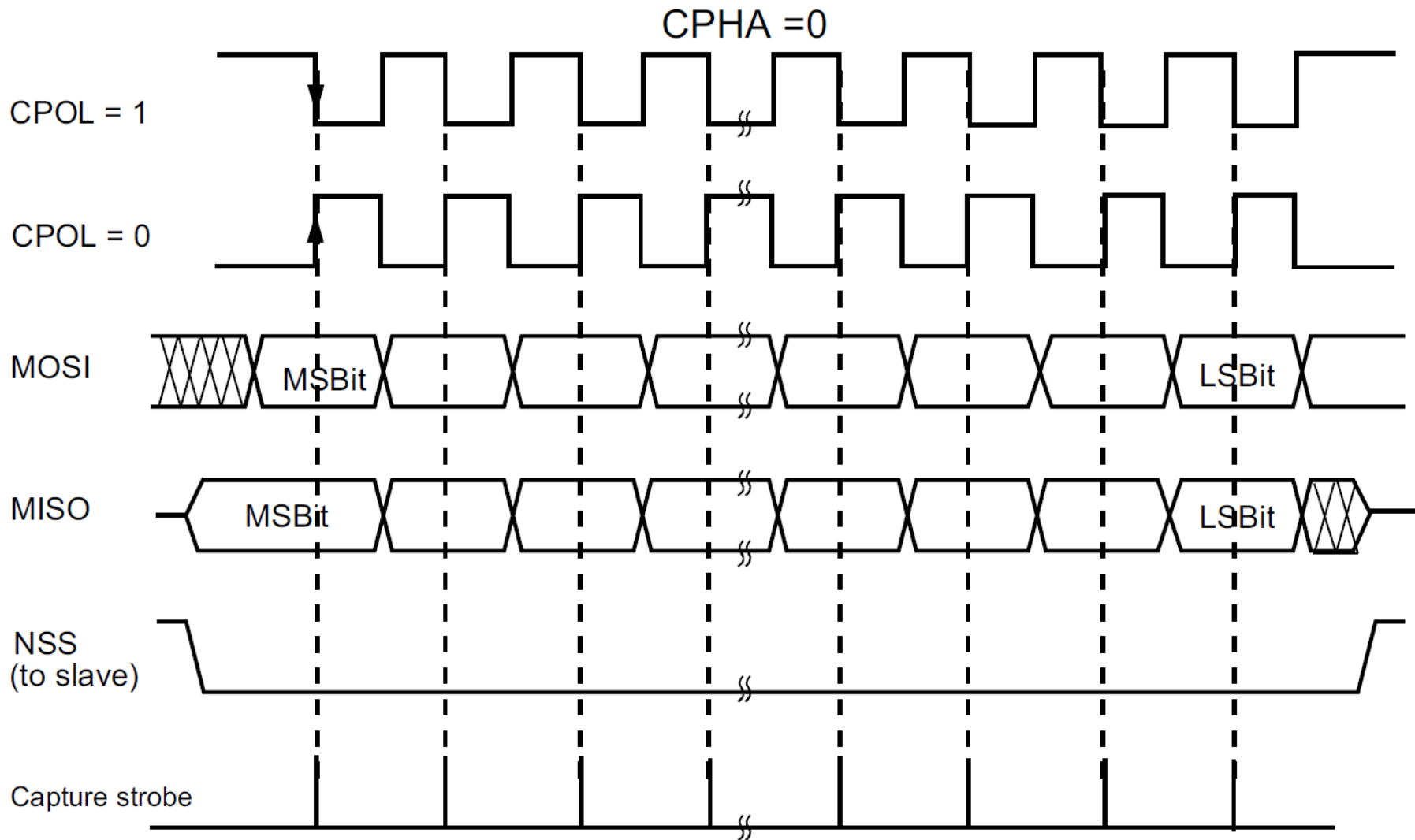
SPI Timing Chart



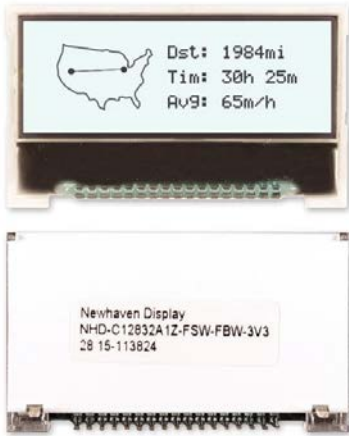
Cronogramas



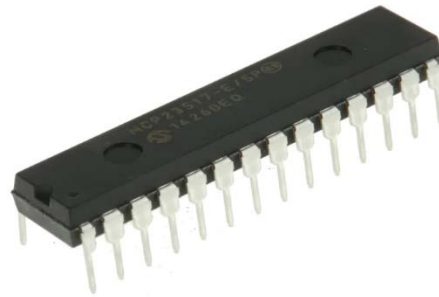
Cronogramas



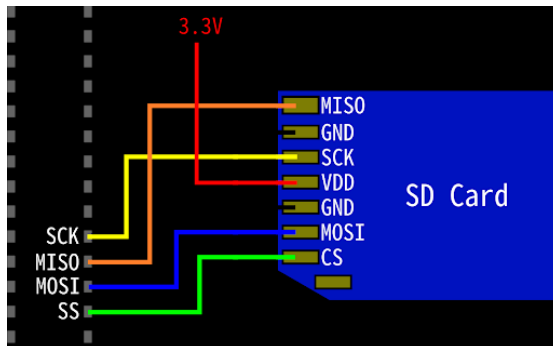
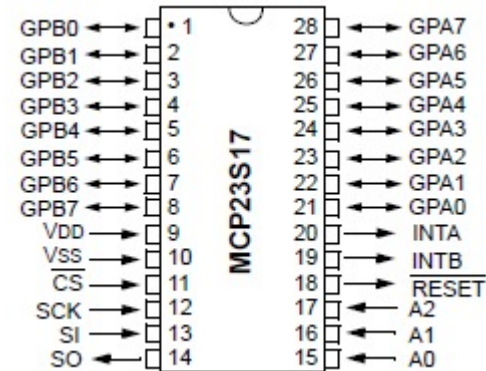
Periféricos SPI



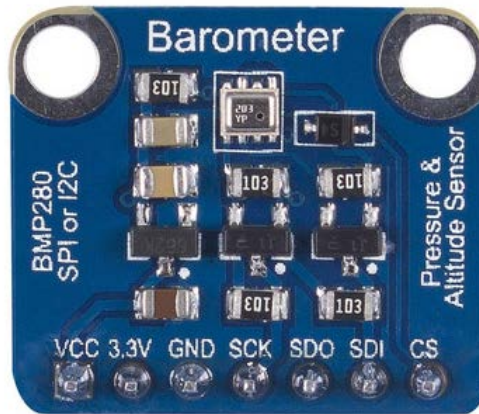
LCD Gráfico



Expansor GPIOs



Lector SD

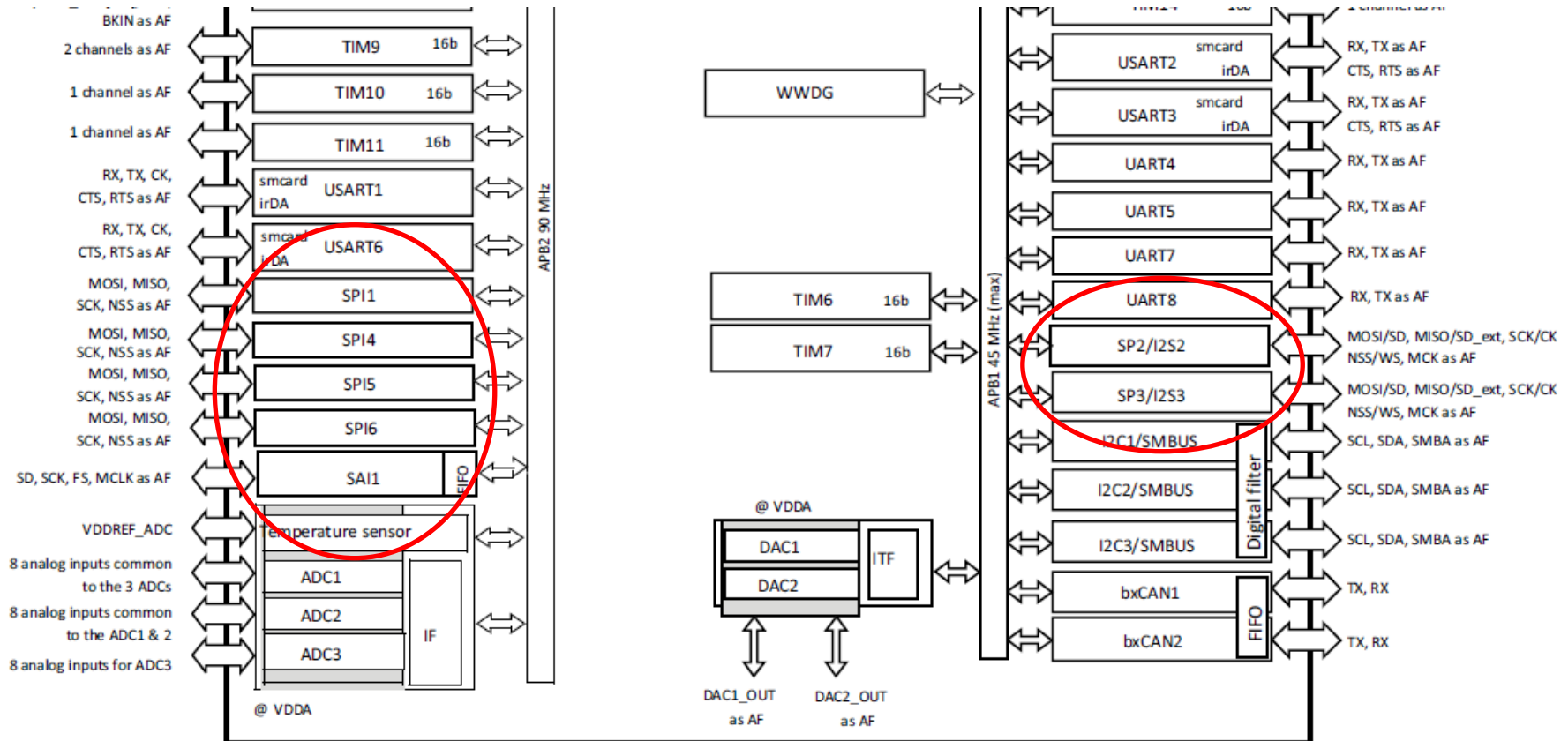


Sensor Tª/Presión



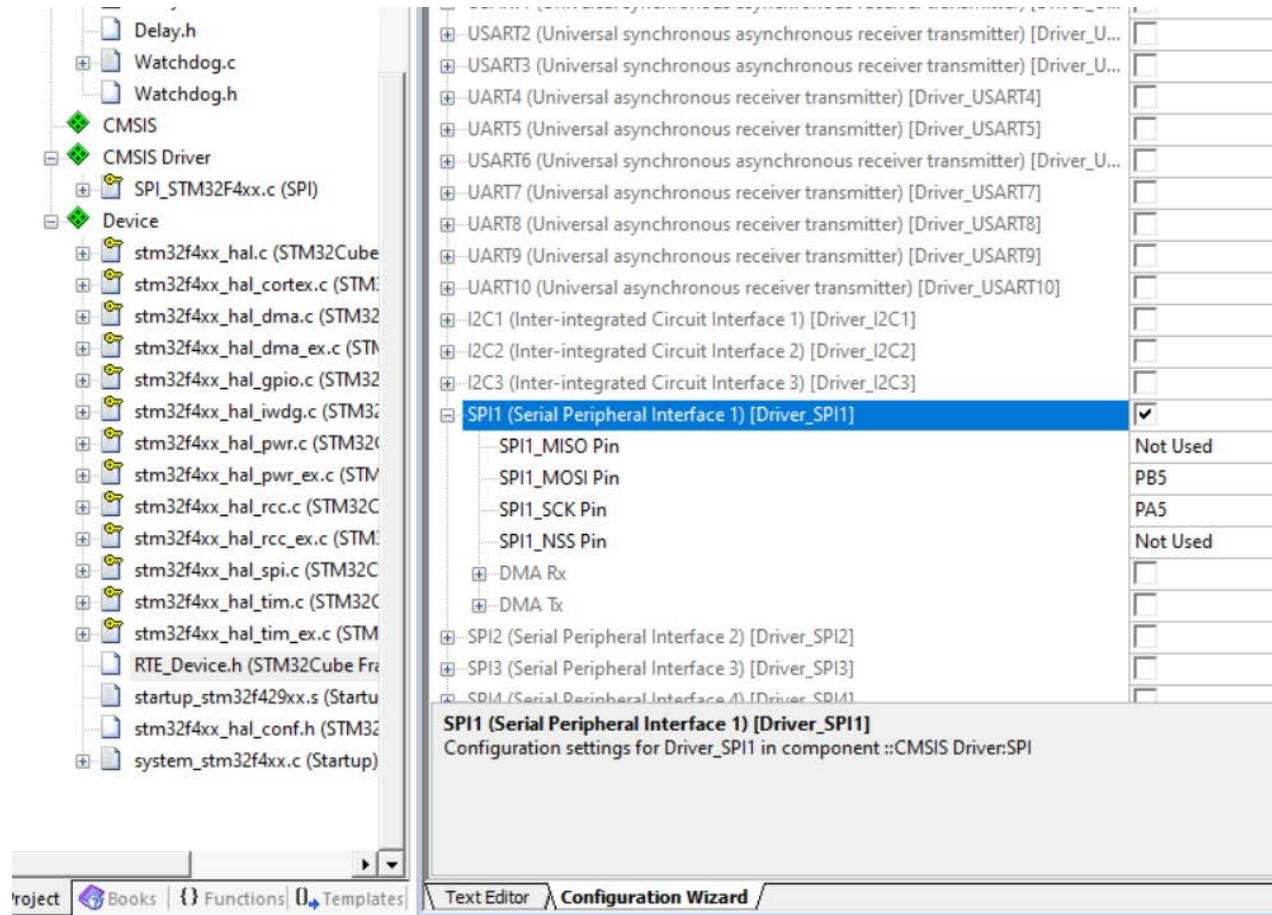
Emisor RF-433

SPI en el F429ZI



SPI en el F429ZI

Hasta 6 interfaces series síncronos (SP1-SPI6) que se pueden utilizar para dispositivos SPI y otros dispositivos serie síncronos.



Registros

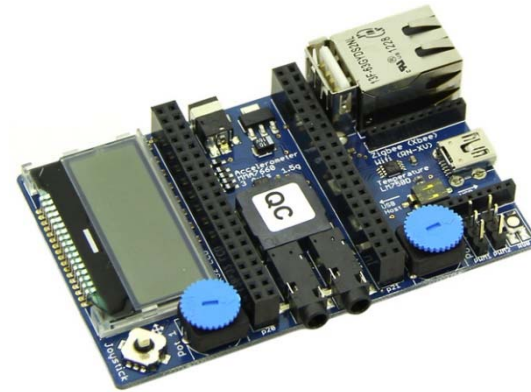
Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0						
0x00	SPI_CR1	Reserved																BIDIMODE	BIDIOE	CRCEN	CRCNEXT	DFE	RXONLY	SSM	SSI	LSBFIRST	SPE	BR [2:0]			MSTR	CPOL	CPHA						
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
0x04	SPI_CR2	Reserved																								TXEIE	RXNEIE	ERRIE	FRF	Reserved	SSOE	TXDMAEN	RXDMAEN						
	Reset value																									0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x08	SPI_SR	Reserved																							FRE	BSY	OVR	MODF	CRCERR	UDR	CHSIDE	TXE	RXNE						
	Reset value																								0	0	0	0	0	0	0	0	1	0					
0x0C	SPI_DR	Reserved																DR[15:0]																					
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0					
0x10	SPI_CRCPR	Reserved																CRCPOLY[15:0]																					
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1					
0x14	SPI_RXCR	Reserved																RxCRC[15:0]																					
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
0x18	SPI_TXCR	Reserved																TxCRC[15:0]																					
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0					
0x1C	SPI_I2SCFGR	Reserved																				I2SMOD	I2SE	I2SCFG	PCMSSYNC	Reserved	I2SSTD	CKPOL	DATLEN	CHLEN									
	Reset value																					0	0	0	0	0	0	0	0	0	0	0	0	0	0				
0x20	SPI_I2SPR	Reserved																						MCKOE	ODD	I2SDIV													
	Reset value																							0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	

Registros

Denominación	Función
SPI_CR1/SPI_CR2	SPI Control Registers
SPI_DR	SPI Data Register
SPI_SR	SPI Status Register

Página 916 (STM32F429-REFERENCE-MANUAL.pdf)

Sistemas Basados en Microprocesador



B2 LCD mbed APPBoard

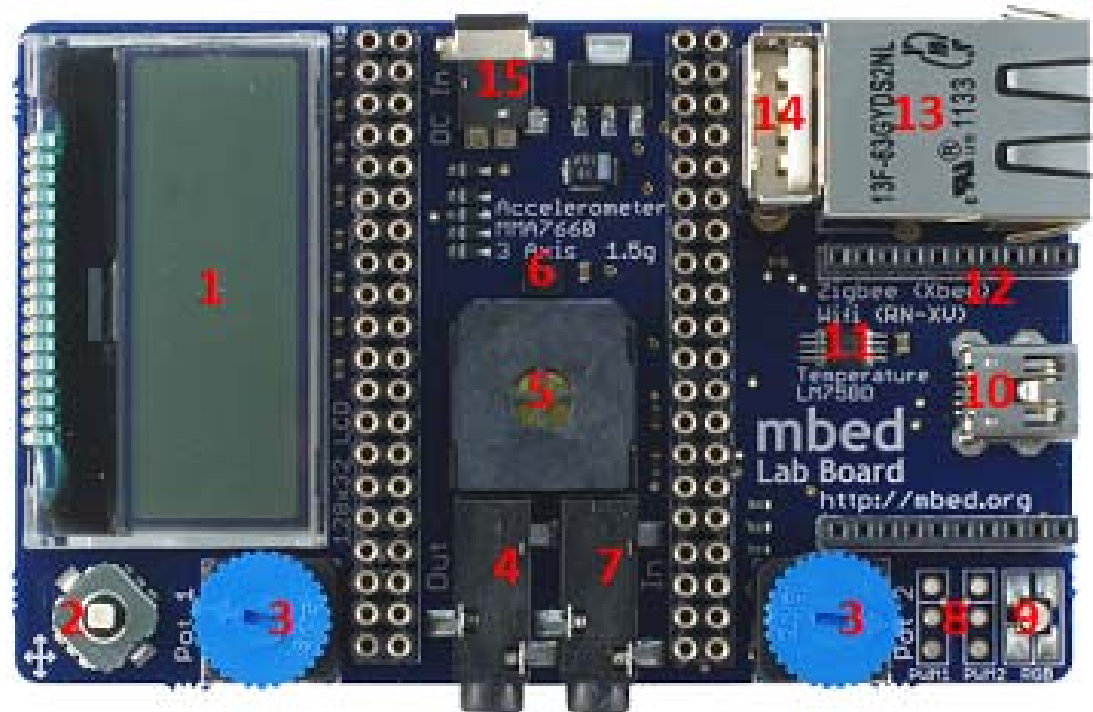
LCD

- Dispositivo con interfaz SPI. Enviando información en serie al LCD se puede configurar y representar información en él
- El display que se va a utilizar está montado en la placa de aplicaciones *mbed App Board*
- Es un display gráfico de 128x32 pixels. Eso quiere decir que debemos gestionar si un punto de la pantalla se enciende o no
- El display tiene una memoria interna de 65x132 donde se almacenan los datos que se representan

Display SPI 128x32 Gráfico

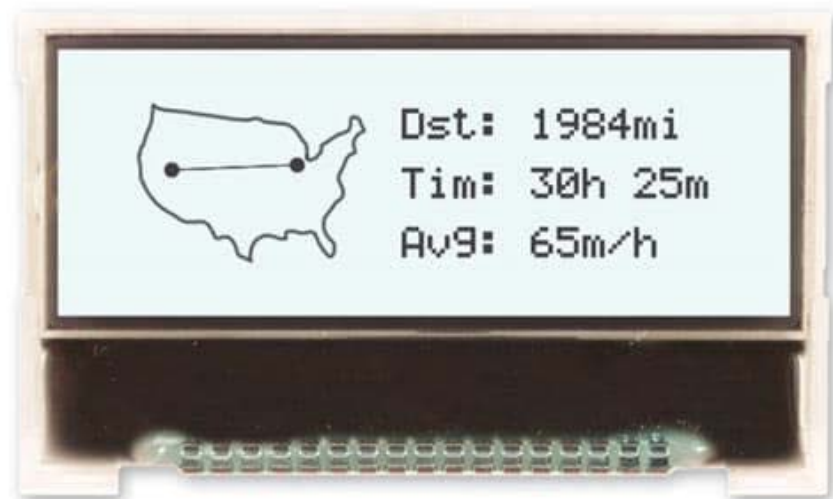
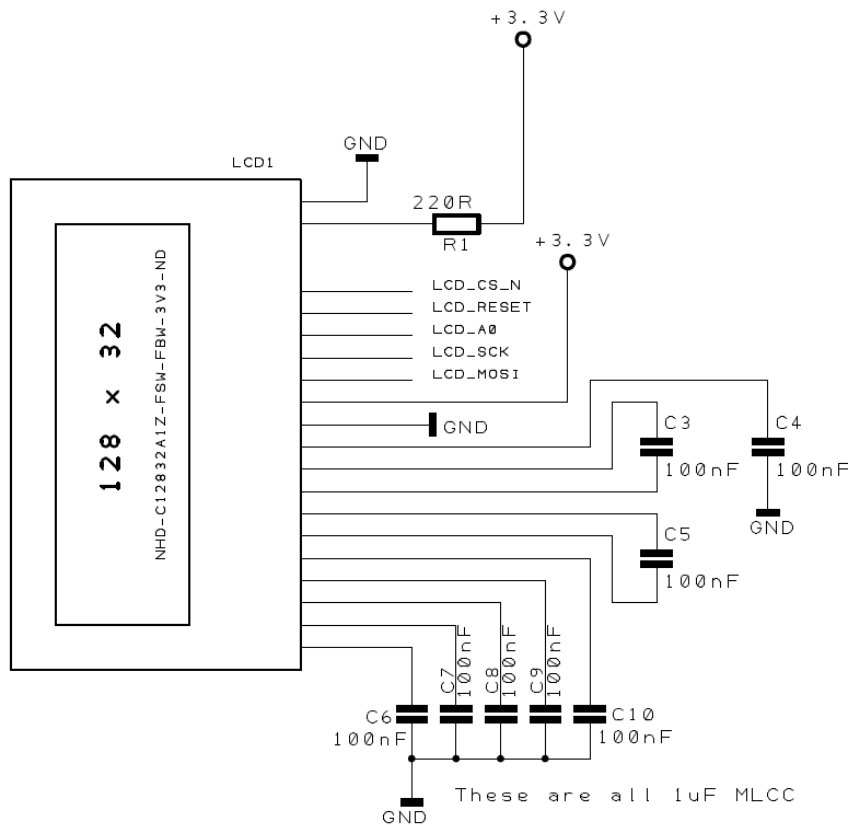
■ mbed Application Board

1. 128x32 Graphics LCD
2. 5 way joystick
3. 2 x Potentiometers
4. 3.5mm Audio jack (Analog Out)
5. Speaker, PWM Connected
6. 3 Axis +/- 1.5g Accelerometer
7. 3.5mm Audio jack (Analog In)
8. 2 x Servo motor headers
9. RGB LED, PWM connected
10. USB-mini-B Connector
11. Temperature sensor
12. Socket for for Xbee (Zigbee) or RN-XV (Wifi)
13. RJ45 Ethernet connector
14. USB-A Connector
15. 1.3mm DC Jack input



<http://mbed.org/cookbook/mbed-application-board>

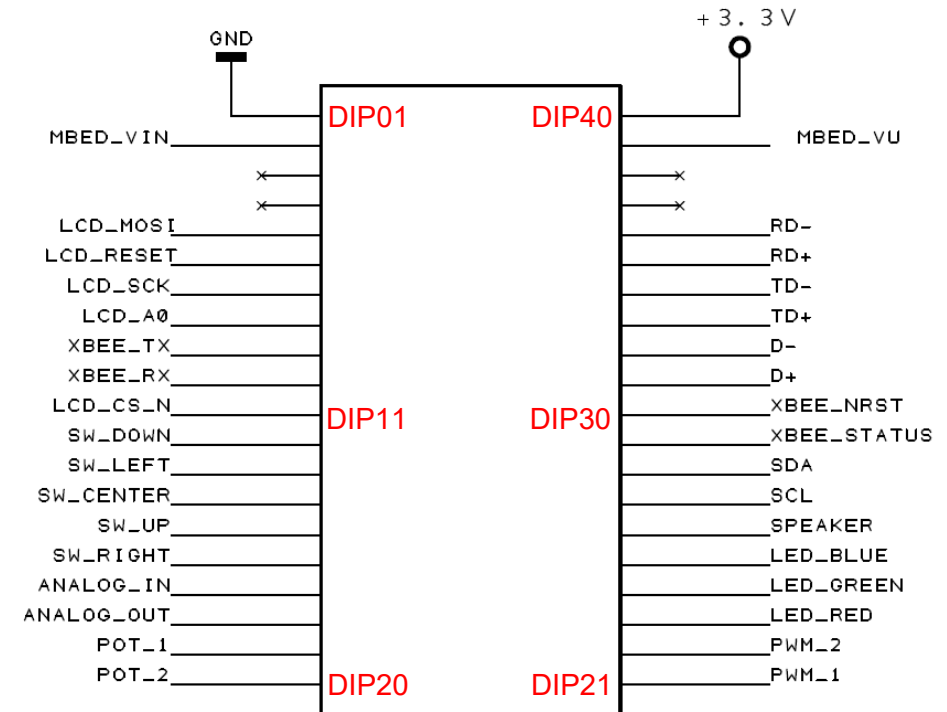
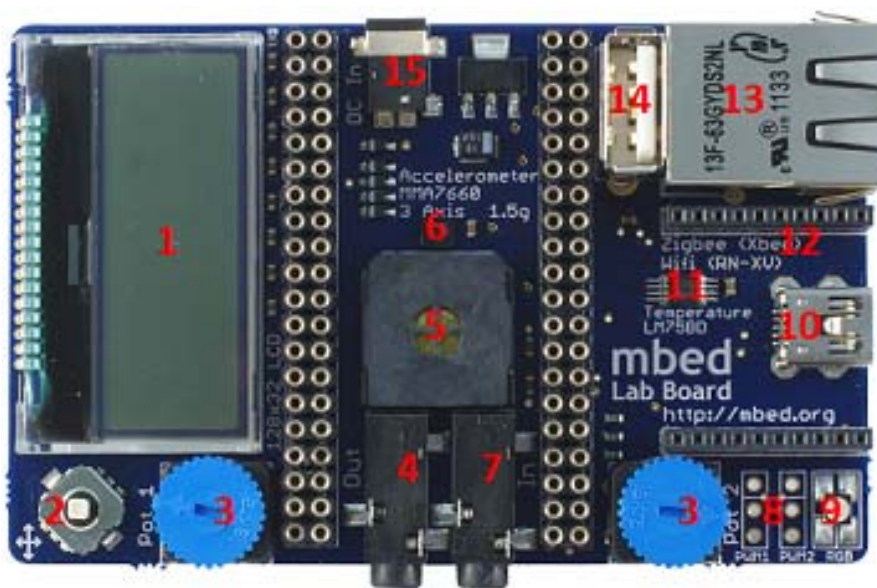
Display SPI 128x32 Gráfico



https://os.mbed.com/media/uploads/chris/mbed-014.1_b.pdf

<https://os.mbed.com/components/mbed-Application-Board/>

Display SPI 128x32 Gráfico



Conexiones STM-F429ZI - mbedAPPBoard

mbedAPPBoard		STM-F429ZI	
GND	DIP01	GND(11-CN8))	
+3.3v	DIP40	+3V3(7-CN8)	
LCD_MOSI	DIP05	D11	PA7
LCD_RESET	DIP06	D12	PA6
LCD_SCK	DIP07	D13	PA5
LCD_A0	DIP08	D7	PF13
LCD_CS_N	DIP11	D10	PD14

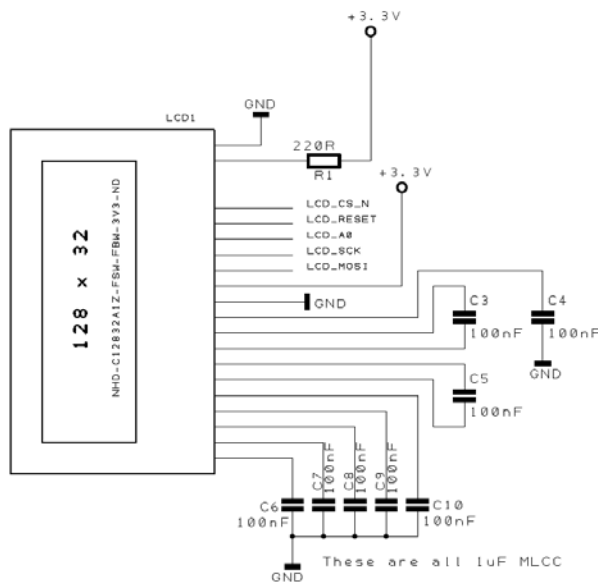
Las señales **RESET, CS y A0** del display se tienen que manejar directamente con el GPIO.

Display SPI 128x32 Gráfico

<https://www.newhavendisplay.com/specs/NHD-C12832A1Z-FSW-FBW-3V3.pdf>

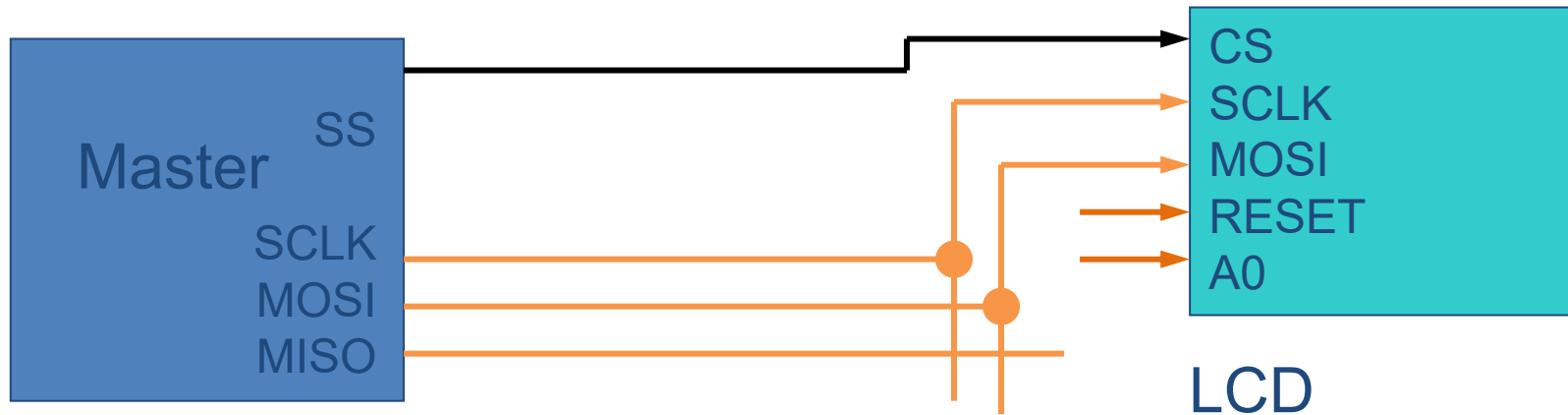
<http://www lcd-module.de/eng/pdf/zubehoer/st7565r.pdf>

Pin Description and Wiring Diagram



Pin No.	Symbol	External Connection	Function Description
1	V ₀	Power Supply	0.1μF – 1μF Capacitor to V _{SS}
2	V ₁	Power Supply	0.1μF – 1μF Capacitor to V _{SS}
3	V ₂	Power Supply	0.1μF – 1μF Capacitor to V _{SS}
4	V ₃	Power Supply	0.1μF – 1μF Capacitor to V _{SS}
5	V ₄	Power Supply	0.1μF – 1μF Capacitor to V _{SS}
6	C2-	Power Supply	Connect 1μF – 2.2μF Capacitor to C2+ (pin 7)
7	C2+	Power Supply	Connect 1μF – 2.2μF Capacitor to C2- (pin 6)
8	C1+	Power Supply	Connect 1μF – 2.2μF Capacitor to C1- (pin 9)
9	C1-	Power Supply	Connect 1μF – 2.2μF Capacitor to C1+ (pin 8)
10	V _{OUT}	Power Supply	Connect 1μF – 2.2μF Capacitor to V _{SS} (pin 11)
11	V _{SS}	Power Supply	Ground
12	V _{DD}	Power Supply	Supply Voltage for LCD and Logic (+3V)
13	SI	MPU	Serial Data
14	SCL	MPU	Serial Clock
15	A0	MPU	Register Select. A0=0: Instruction, A0=1: Data
16	/RST	MPU	Active LOW Reset signal
17	/CS1	MPU	Active LOW Chip Select signal
A	LED+	Power Supply	Backlight Anode(+3V)
K	LED-	Power Supply	Backlight Cathode (Ground)

SPI – Conexión entre micro y el display



F429ZI

¿Qué pines del micro se deben configurar para conectar el controlador SPI?

¿Cuántos controladores SPI hay en el F429ZI?

¿Para que sirven las señales RESET y A0 del Display?

Gestión del display

- La información se envía al display usando la línea serie síncrona del SPI (MOSI) y la señal de reloj (SCLK)
- La información es interpretada de dos maneras:
 - Si **A0=0** la información recibida se interpreta como **comandos** en el display. Es decir realizan operaciones de *configuración*.
 - Si **A0=1** la información recibida se guardan en la **memoria del display** porque esta es la *información gráfica* que se quiere representar.
- El display sólo puede recibir **operaciones de escritura** (MOSI) y no de lectura (MISO).

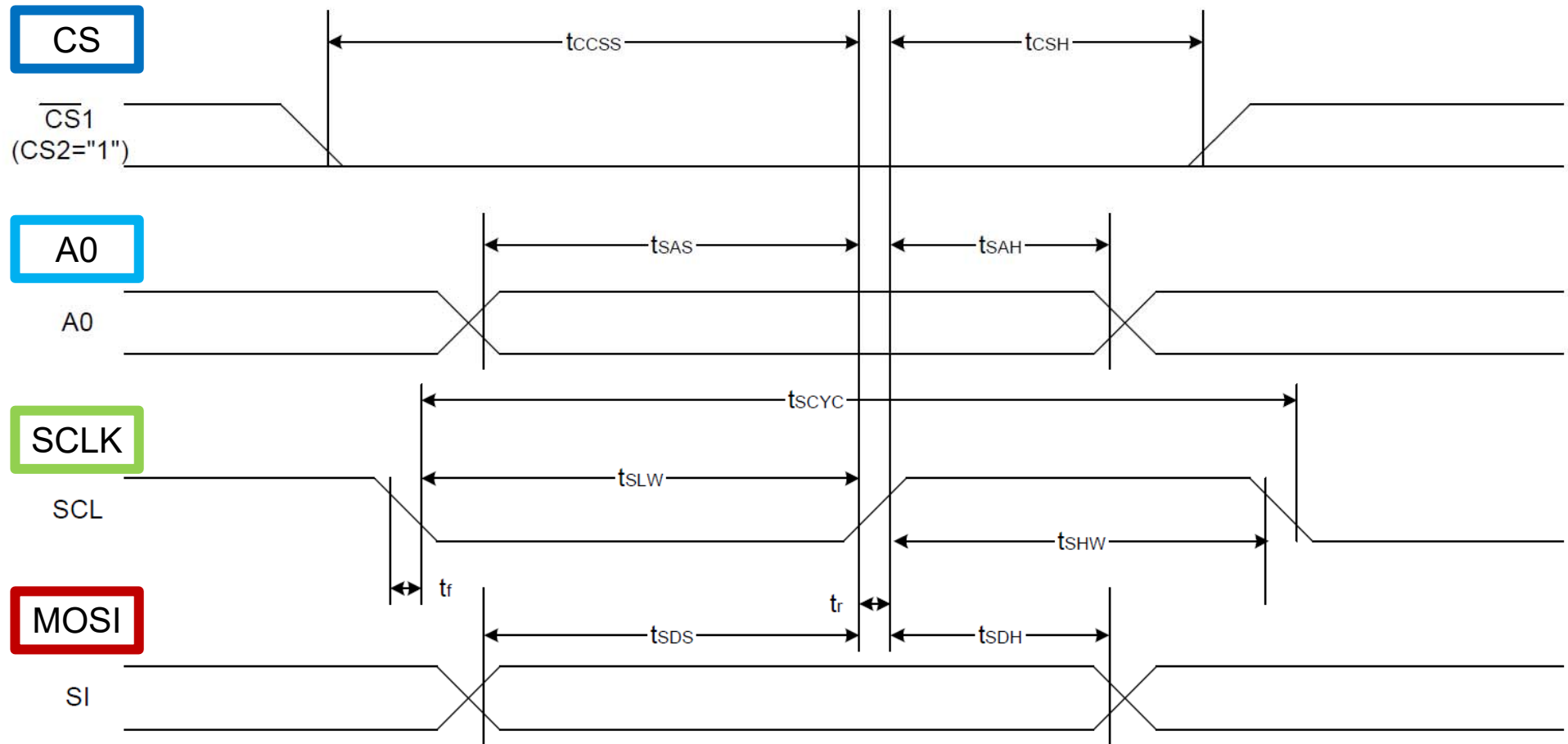
Comandos/datos en el display

Command	Command Code										Function	
	A0	/RD	/WR	D7	D6	D5	D4	D3	D2	D1		D0
(1) Display ON/OFF	0	1	0	1	0	1	0	1	1	1	0 1	LCD display ON/OFF 0: OFF, 1: ON
(2) Display start line set	0	1	0	0	1	Display start address					Sets the display RAM display start line address	
(3) Page address set	0	1	0	1	0	1	1	Page address			Sets the display RAM page address	
(4) Column address set upper bit Column address set lower bit	0	1	0	0	0	0	1	Most significant column address			Sets the most significant 4 bits of the display RAM column address.	
				0	0	0	0	Least significant column address			Sets the least significant 4 bits of the display RAM column address.	
(5) Status read	0	0	1	Status			0	0	0	0	Reads the status data	
(6) Display data write	1	1	0	Write data					Writes to the display RAM			
(7) Display data read	1	0	1	Read data					Reads from the display RAM			
(8) ADC select	0	1	0	1	0	1	0	0	0	0	0 1	Sets the display RAM address SEG output correspondence 0: normal, 1: reverse
(9) Display normal/reverse	0	1	0	1	0	1	0	0	1	1	0 1	Sets the LCD display normal/ reverse 0: normal, 1: reverse
(10) Display all points ON/OFF	0	1	0	1	0	1	0	0	1	0	0 1	Display all points 0: normal display 1: all points ON
(11) LCD bias set	0	1	0	1	0	1	0	0	0	1	0 1	Sets the LCD drive voltage bias ratio 0: 1/9 bias, 1: 1/7 bias (ST7565R)
(12) Read-modify-write	0	1	0	1	1	1	0	0	0	0	0	Column address increment At write: +1 At read: 0

Comandos/datos en el display

(13) End	0	1	0	1	1	1	0	1	1	1	0	Clear read/modify/write
(14) Reset	0	1	0	1	1	1	0	0	0	1	0	Internal reset
(15) Common output mode select	0	1	0	1	1	0	0	0	*	*	*	Select COM output scan direction 0: normal direction 1: reverse direction
(16) Power control set	0	1	0	0	0	1	0	1	Operating mode			Select internal power supply operating mode
(17) V_0 voltage regulator internal resistor ratio set	0	1	0	0	0	1	0	0	Resistor ratio			Select internal resistor ratio(R_b/R_a) mode
(18) Electronic volume mode set Electronic volume register set	0	1	0	1	0	0	0	0	0	0	1	Set the V_0 output voltage electronic volume register
(19) Sleep mode set	0	1	0	1	0	1	0	1	1	0	0	0: Sleep mode, 1: Normal mode
(20) Booster ratio set	0	1	0	1	1	1	1	1	0	0	0	select booster ratio 00: 2x,3x,4x 01: 5x 11: 6x
(21) NOP	0	1	0	1	1	1	0	0	0	1	1	Command for non-operation
(22) Test	0	1	0	1	1	1	1	*	*	*	*	Command for IC test. Do not use this command

Temporización SPI del LCD



Temporización del reset

Reset Timing

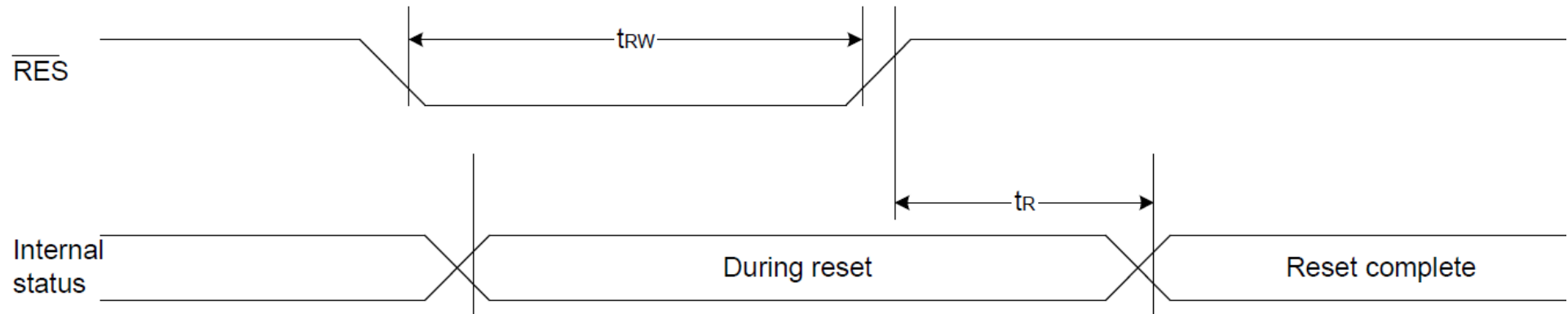


Figure 41

Table 30

(V_{DD} = 3.3V, T_a = -30 to 85°C)

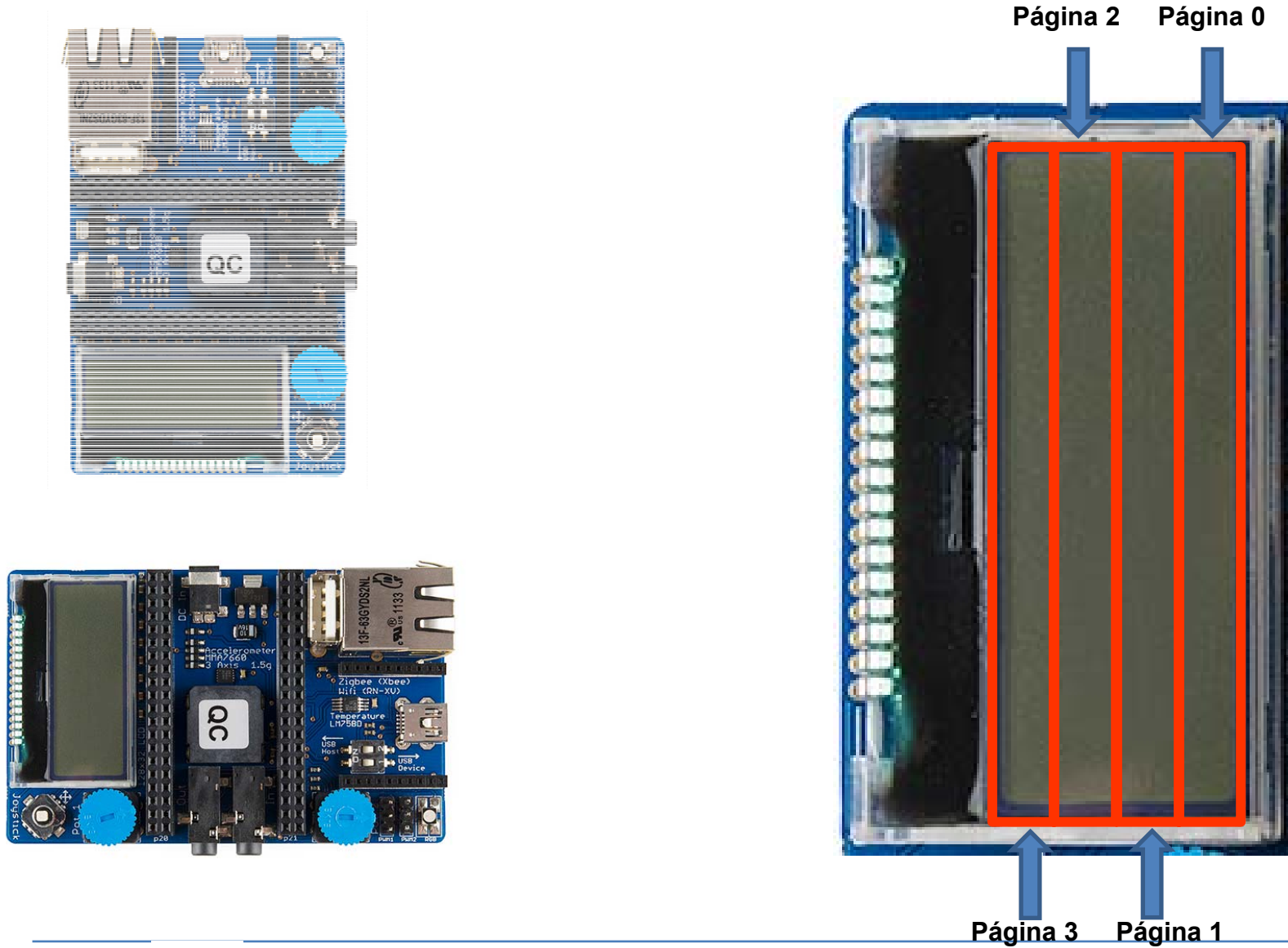
Item	Signal	Symbol	Condition	Rating			Units
				Min.	Typ.	Max.	
Reset time		t _R		—	—	1.0	us
Reset “L” pulse width	/RES	t _{RW}		1.0	—	—	us

Table 31

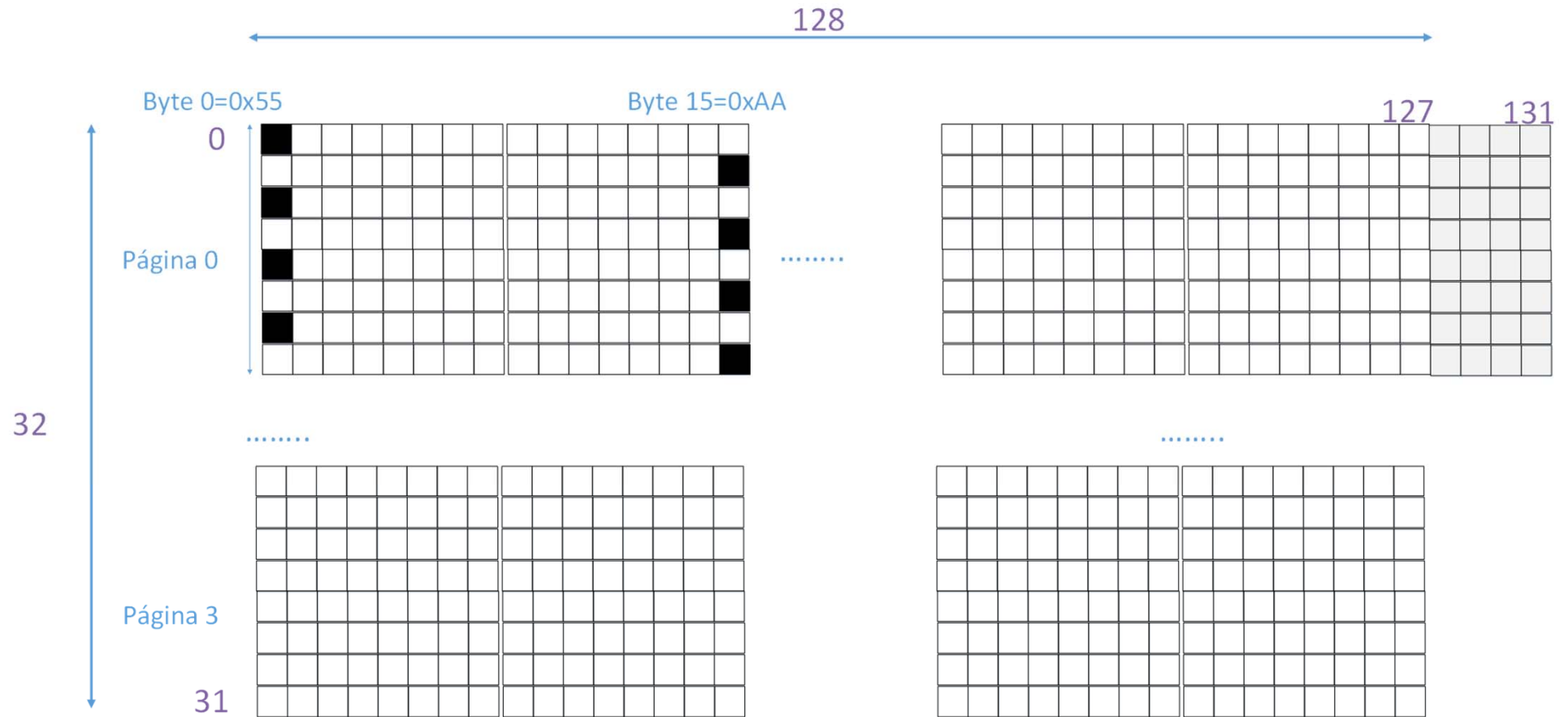
(V_{DD} = 2.7V, T_a = -30 to 85°C)

Item	Signal	Symbol	Condition	Rating			Units
				Min.	Typ.	Max.	
Reset time		t _R		—	—	2.0	us
Reset “L” pulse width	/RES	t _{RW}		2.0	—	—	us

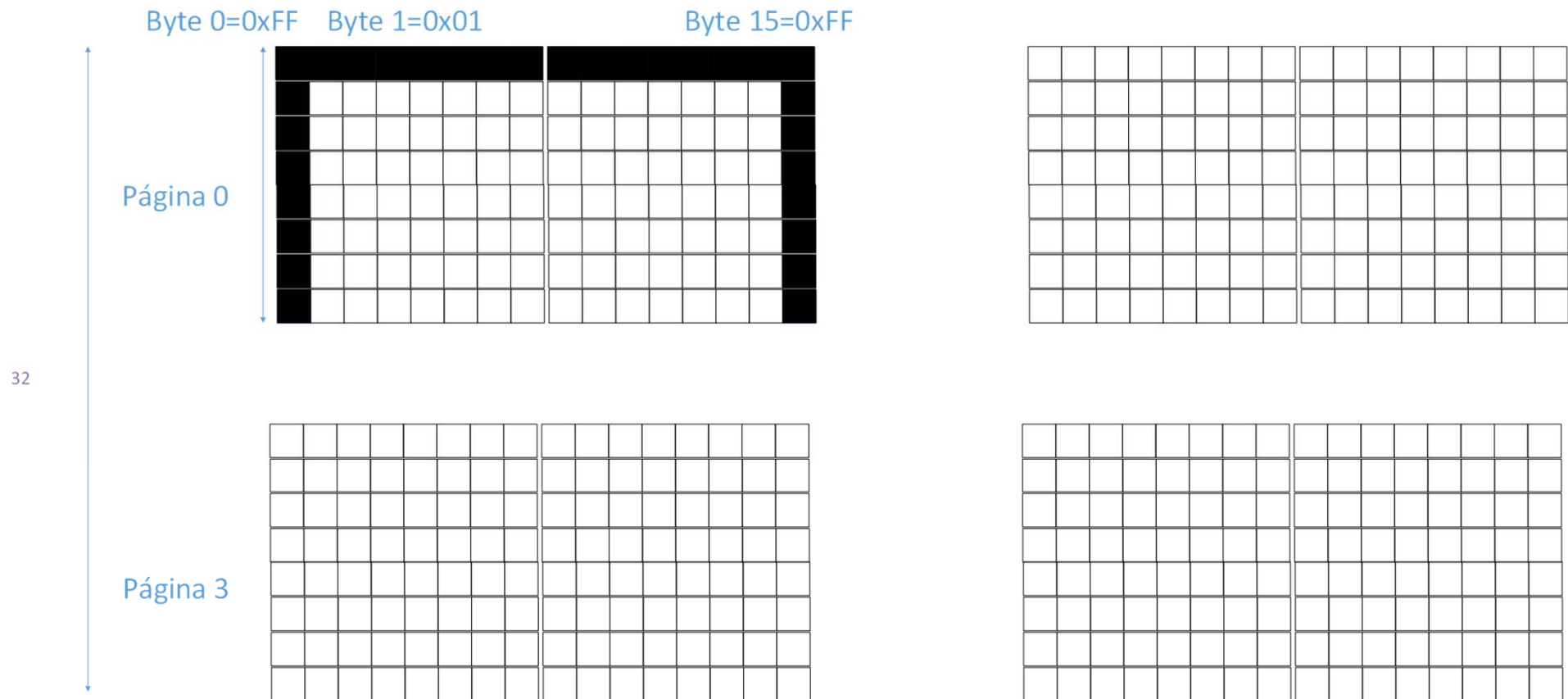
Organización de la información en la pantalla: ejemplo



Organización de la información en la pantalla: ejemplo



Organización de la información en la pantalla: ejemplo 2



Demo

