

TÍTULO	LIBRERIA PARA MANEJO DE DS1302 USANDO SPI EN MODO HALF DUPLEX PARA STM32
AMBITO	Ingengería Electrónica
AUTOR	García Camoira Cristóbal
FECHA	Febrero de 2022

Índice

Listado de figuras	3
Listado de tablas	3
Listado de códigos de programación	3
Introducción	4
1 Configuración	4
1.1 STM CUBE MX	5
1.2 Programación del dispositivo	10

Listado de figuras

1	Vista de las pistas de la cara delantera de la PCB tras el proceso de insolado	4
2	Configuración de la frecuencia del sistema	5
3	Configuración de la frecuencia del sistema	5
4	Configuración de timer3 cada 500ms	6
5	Configuración de timer3 cada 500ms	6
6	Configuración del JTAG para el uso del STLINK-V2	6
7	Configuración del periférico USART para la visualización de los datos a través de un monitor serie	7
8	Configuración del periférico USART para la visualización de los datos a través de un monitor serie	7
9	Configuración del periférico USART para la visualización de los datos a través de un monitor serie	8
10	Añadiendo prioridades a las interrupciones	8
11	Configuración del SPI en modo Half-Duplex	9
12	Configuración del SPI en modo Half-Duplex	9
13	Configuración del project manager	10
14	Visualización del pinout	10

Listado de tablas

Listado de códigos de programación

Introducción

El presente proyecto tiene como objetivo la realización de una librería de funciones para el manejo del chip DS1302 , un reloj en tiempo real, usando una característica que poseen los microcontroladores de de STM microelectronics de 32 bits, concretamente la familia de STM32F103, mediante la cual el SPI puede usar solo 3pines para la transferencia de información, modo half duplex, donde tendremos las siguientes salidas:

- MOSI: Señal de datos, también llamada I/O, durante la escritura, la el pin del microcontrolador estará configurado como salida, y durante la lectura se comportara como entrada, este pin es bidireccional.
- CLK: Señal de reloj
- CE: Chip enable

Conexión

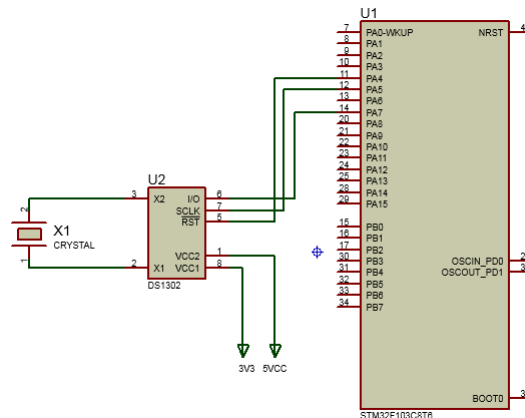
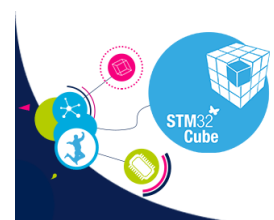
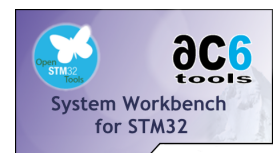


Figura 1: Pistas PCB cara delantera

1 Configuración

Para la configuración del software del dispositivo se han usado dos entornos de desarrollo, el primero es in IDE llamado System Workbench, este IDE, basado en eclipse integra todo lo necesario para poder programar microcontroladores de la familia STM32, incluido el compilador y sin ningún tipo de limite, es gratuito. Este se puede descargar del siguiente enlace aunque para ello primero debemos registrarnos en su pagina web <https://www.openstm32.org/HomePage>.

La segunda tool que necesitaremos es el STM CUBE MX, una tool que nos ofrece de forma gratuita el fabricante ST y que nos permite descargar las capas de abstracción de software especificas para nuestro dispositivo, lo que se denomina como HAL, en ingles, hardware abstraction layer, que, dependiendo del microcontrolador que vayamos a usar necesitaremos descargar una u otra. Esta tool nos permite la confuguracion del dispositivo para luego generar el código y crear el proyecto de eclipse en System workbench. Al igual que



para la obtención del IDE de programación va a ser necesario registrarse para obtener este software <https://www.st.com/en/development-tools/stm32cubemx.html#overview>

También necesitaremos los drivers para poder usar el programador y debugger stlinkV2, en la web esta disponible para mac, windows y linux <https://www.st.com/en/development-tools/st-link-v2.html#tools-software>

1.1 STM CUBE MX

Normalmente en la mayoría de proyectos solemos intentar que nuestro sistema vaya a la mayor velocidad posible, por ello trataremos de que nuestro sistema sea capaz de ejecutar las instrucciones lo mas rápido posible, para ello configuraremos la frecuencia del sistema a 72Mhz, que es la máxima que se puede alcanzar para este caso en concreto, dado que estamos hablando del STM32F103C8T6. Para ello lo único que debemos hacer es aplicar las siguientes opciones el modo de configuración del RCC:

1. Habilitar el HSE con la opción Crystal/ceramic resonator. Para ello es necesario disponer de un cristal de cuarzo de 8MHz conectado de forma externa, que , de serie ya incluye la placa de evaluación denominada «Blue pill»
2. Habilitar el LSE con la opción Crystal/ceramic resonator. Esta opción es recomendable aunque no necesaria, ésta habilita el cristal externo de 32khz para usar el RTC interno que dispone el microcontrolador. Si se requiere usar el RTC interno esta opciones indispensable.

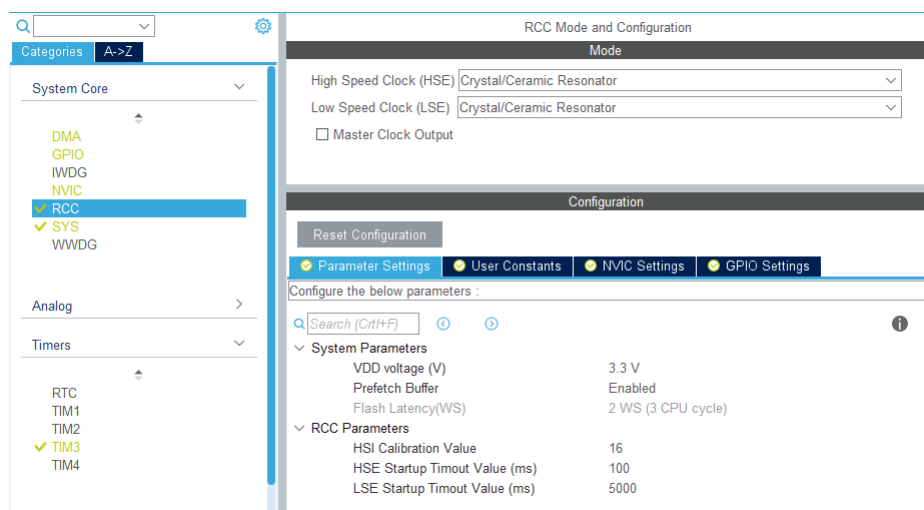


Figura 2

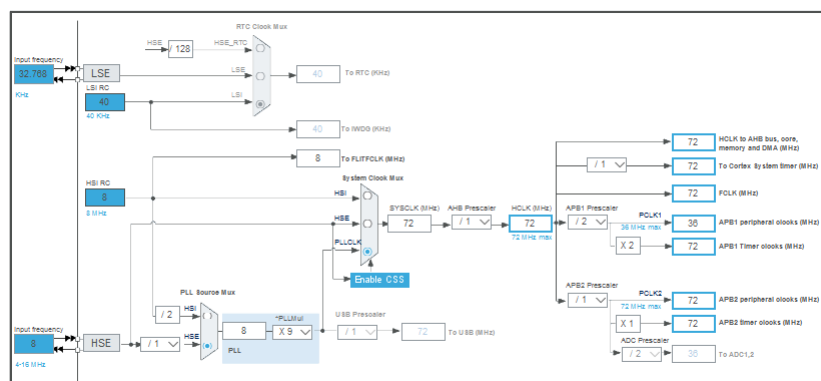


Figura 3

Otra cosa muy importante es configurar un timer cada medio segundo, de esta forma nos aseguraremos de no perder ningún cambio en la variable de los segundos, una vez que salte el timer, se consultara al driver la fecha y la hora y se actualizaran las variables internas.

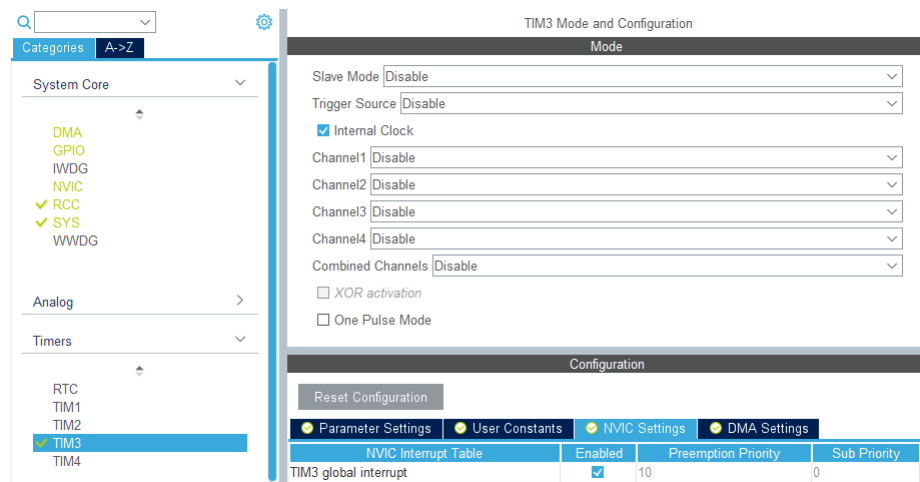


Figura 4: Configuración de timer3 cada 500ms

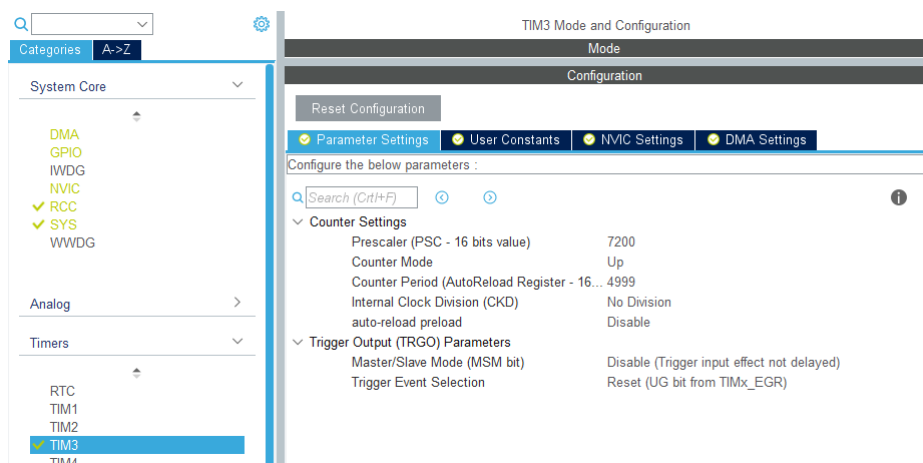


Figura 5: Configuración de timer3 cada 500ms

Por supuesto, para flashear nuestro código en la placa necesitaremos configurar en nuestro sistema el modo JTAG 4 pin , de esta forma permitiremos el uso del STLINK V2 para el flashing /debug de nuestro código.

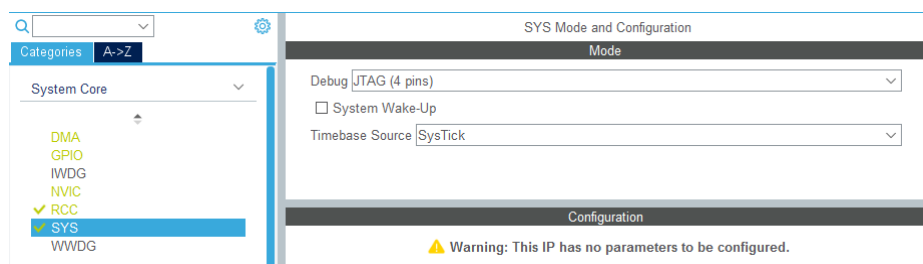


Figura 6: Configuración del JTAG para el uso del STLINK-V2

El método mas sencillo para conocer que nuestro dispositivo esta funcionando correctamente es poder visualizar los valores que el dispositivo DS1302 le envía a nuestro microcontrolador, por ello se ha decidido

usar un UART.

USART1 Mode and Configuration

Mode

Mode

Asynchronous

Hardware Flow Control (RS232)

Disable

Configuration

Reset Configuration

Parameter Settings User Constants NVIC Settings DMA Settings GPIO Settings

Configure the below parameters :

Search (Ctrl+F)

Basic Parameters

Baud Rate115200 Bits/s

Word Length8 Bits (including Parity)

ParityNone

Stop Bits1

Advanced Parameters

Data DirectionReceive and Transmit

Over Sampling16 Samples

Figura 7: Configuración del periférico USART para la visualización de los datos a través de un monitor serie

USART1 Mode and Configuration

Mode

Mode

Asynchronous

Hardware Flow Control (RS232)

Disable

Configuration

Reset Configuration

Parameter Settings User Constants NVIC Settings DMA Settings GPIO Settings

NVIC Interrupt Table	Enabled	Preemption Priority	Sub Priority
DMA1 channel4 global interrupt	<input checked="" type="checkbox"/>	4	0
DMA1 channel5 global interrupt	<input checked="" type="checkbox"/>	5	0
USART1 global interrupt	<input checked="" type="checkbox"/>	8	0

Figura 8: Configuración del periférico USART para la visualización de los datos a través de un monitor serie

USART1 Mode and Configuration

Mode: Asynchronous

Hardware Flow Control (RS232): Disable

Configuration

Reset Configuration

Parameter Settings | User Constants | NVIC Settings | **DMA Settings** | GPIO Settings

DMA Request	Channel	Direction	Priority
USART1_RX	DMA1 Channel 5	Peripheral To Memory	Low
USART1_TX	DMA1 Channel 4	Memory To Peripheral	Low

Add Delete

Figura 9: Configuración del periférico USART para la visualización de los datos a través de un monitor serie

Aunque no es muy recomendable esta practica, dentro de la interrupción del timer usaremos la función serie para imprimir la hora que nos va devolviendo el periférico Ds1302 cada medio segundo, para ello es necesario reconfigurar las interrupciones y darle mayor prioridad a la del timer3, de esta forma nunca habrá conflicto sobre cual interrupción se ejecutara antes en caso de recibir peticiones simultaneas.

NVIC Mode and Configuration

Configuration

NVIC | Code generation

Priority Group: 4 bits for pre-emption priority 0 bits for subpriority

Sort by Preemption Priority and Sub Priority

Search: Search (Ctrl+F)

Show only enabled interrupts Force DMA channels interrupts

NVIC Interrupt Table	Enabled	Preemption Priority	Sub Priority
Non maskable interrupt	<input checked="" type="checkbox"/>	0	0
Hard fault interrupt	<input checked="" type="checkbox"/>	0	0
Memory management fault	<input checked="" type="checkbox"/>	0	0
Prefetch fault, memory access fault	<input checked="" type="checkbox"/>	0	0
Undefined instruction or illegal state	<input checked="" type="checkbox"/>	0	0
System service call via SWI instruction	<input checked="" type="checkbox"/>	0	0
Debug monitor	<input checked="" type="checkbox"/>	0	0
Pendable request for system service	<input checked="" type="checkbox"/>	0	0
Time base: System tick timer	<input checked="" type="checkbox"/>	0	0
DMA1 channel4 global interrupt	<input checked="" type="checkbox"/>	4	0
DMA1 channel5 global interrupt	<input checked="" type="checkbox"/>	5	0
USART1 global interrupt	<input checked="" type="checkbox"/>	8	0
TIM3 global interrupt	<input checked="" type="checkbox"/>	10	0

Enabled Preemption Priority Sub Priority

Figura 10: Añadiendo prioridades a las interrupciones

Para el correcto uso del DS1302, necesitaremos un periférico en nuestro microcontrolador que satisfaga las necesidades de envío y recepción de los mensajes necesarios para la configuración de este dispositivo. El DS1302 dispone de una entrada de habilitación activa a nivel bajo, una señal de reloj y un pin bidireccional que funciona como entrada/salida de datos. El único periférico existente en nuestro microcontrolador capaz de comportarse de esta forma es el SPI en modo HALF-duplex.

SPI1 Mode and Configuration

Mode

Mode: Half-Duplex Master

Hardware NSS Signal: Disable

Configuration

Reset Configuration

Parameter Settings | User Constants | NVIC Settings | DMA Settings | GPIO Settings

Configure the below parameters :

Search (Ctrl+F)

Basic Parameters

- Frame Format: Motorola
- Data Size: 8 Bits
- First Bit: LSB First

Clock Parameters

- Prescaler (for Baud Rate): 64
- Baud Rate: 1.125 Mbits/s
- Clock Polarity (CPOL): Low
- Clock Phase (CPHA): 1 Edge

Advanced Parameters

- CRC Calculation: Disabled
- NSS Signal Type: Software

Figura 11: Configuración del SPI en modo Half-Duplex

SPI1 Mode and Configuration

Mode

Mode: Half-Duplex Master

Hardware NSS Signal: Disable

Configuration

Reset Configuration

Parameter Settings | User Constants | NVIC Settings | DMA Settings | GPIO Settings

Search Signals

Search (Ctrl+F)

☐ Show only Modified Pins

Pin Name	Signal on Pin	GPIO output level	GPIO mode	GPIO Pull-up/Pull-d...	Maximum output sp...	User Label	Modified
PA5	SPI1_SCK	n/a	Alternate Function ...	n/a	High		<input type="checkbox"/>
PA7	SPI1_MOSI	n/a	Alternate Function ...	n/a	High		<input type="checkbox"/>

Figura 12: Configuración del SPI en modo Half-Duplex

Por ultimo, es necesario generar la configuración valida para el IDE que dispongamos, en nuestro caso System Workbench, para ello es necesario seleccionar la opción SW4STM32 y la versión del hal que previamente hayamos descargado.

Home > STM32F103C8Tx > Test_SPI_HalfDuplex.lpc - Project Manager > GENERATE CODE

Pinout & Configuration	Clock Configuration	Project Manager	Tools
Project	Project Settings Project Name: <input type="text" value="Test_SPI_HalfDuplex"/> Project Location: <input type="text" value="C:\Users\Constobal\STM32Cube"/> Application Structure: <input type="text" value="Basic"/> <input type="checkbox"/> Do not generate the ma... Toolchain Folder Location: <input type="text" value="C:\Users\Constobal\STM32Cube\Test_SPI_HalfDuplex\"/> Toolchain / IDE: <input type="text" value="SW4STM32"/> <input checked="" type="checkbox"/> Generate Under ...		
Code Generator			
Advanced Settings	Linker Settings Minimum Heap Size: <input type="text" value="0x200"/> Minimum Stack Size: <input type="text" value="0x400"/> Mcu and Firmware Package Mcu Reference: <input type="text" value="STM32F103C8Tx"/> Firmware Package Name and Version: <input type="text" value="STM32Cube_FW_F1_V1.8.3"/> <input type="checkbox"/> Use latest available version <input checked="" type="checkbox"/> Use Default Firmware Location		

Figura 13: Configuración del project manager

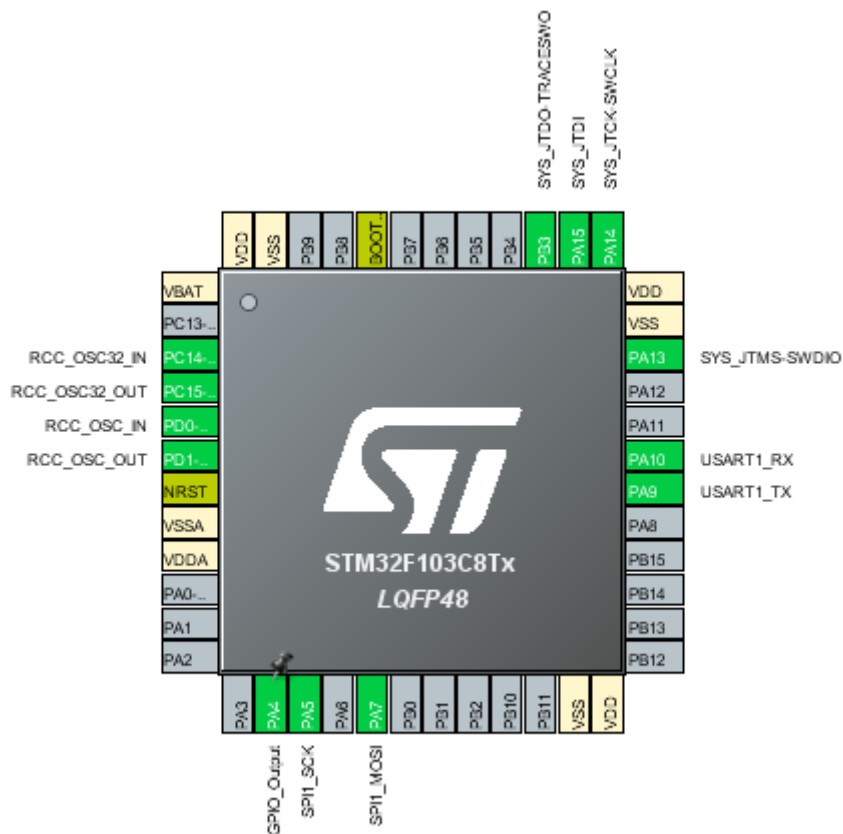


Figura 14: Visualización del pinout

1.2 Programación del dispositivo