|  |
| --- |
| UBA, Facultad de Ingeniería |
| Trabajo práctico Nº1 Protocolo SPI/I2C |
| Módulo ZS-042, Uso del RTC DS3231 |

**Protocolos de Comunicación de Sistemas Embebidos**

**Carrera de Especialización en Sistemas Embebidos**

|  |
| --- |
| **Docente:**  Ing. Pavelek Israel  **Autor:** Ing. Durante Matias.  24-9-2025 |

**Tabla de contenido**

Tabla de contenido

[1. Descripción del RTC DS3231 y hoja de datos. 4](#_Toc209787356)

[2. Diagramas de tiempos de la comunicación 6](#_Toc209787358)

[3. Descripción de la trama I2C para Escritura/Lectura 7](#_Toc209787359)

[Escritura de un registro 7](#_Toc209787360)

[Lectura de un registro 8](#_Toc209787361)

[4. Funciones de la HAL utilizadas. 9](#_Toc209787362)

[Funciones propiamente de la hal de STM 9](#_Toc209787363)

[ANEXO 13](#_Toc209787364)

[1. Datasheet del Dispositivo 13](#_Toc209787365)

[2. Repositorio del Proyecto en GitHub 13](#_Toc209787366)

[3. Documentación Doxygen de la API 13](#_Toc209787367)

**Registro de cambios**

|  |  |  |
| --- | --- | --- |
| **Revisión** | **Cambios realizados** | **Fecha** |
| 1.0 | Creación del documento | 24/09/2025 |

**Protocolo I2C/SPI**

**Actividades a realizar**

1. Seleccione un circuito integrado de interés, el cual puede ser cualquier tipo de sensor, expansor de IO, memoria EEPROM, controlador de LCD, entre otros. Una vez seleccionado, descargue su hoja de datos e incluya la primera página en este trabajo práctico (una captura de imagen es aceptable). De una breve explicación de las funcionalidades del CI seleccionado.  
   *NOTA:* S*e propone que en el caso de que el estudiante ya tenga definido su trabajo final de carrera y el mismo incluye un CI que se comunica por I2C o SPI, esta actividad sea basada en ese dispositivo.*
2. Dentro de la hoja de datos, identifique un diagrama de tiempos de la comunicación (usualmente se muestra bajo la leyenda bus timing) y construya una tabla identificando:
   1. Frecuencia máxima de operación
   2. Tiempo mínimo de pulso de clock en alto.
   3. Tiempo mínimo de pulso de clock en bajo.

Si eligió I2C:

* 1. Tiempo máximo de flanco ascendente (rise time).
  2. Tiempo máximo de flanco descendente (fall time).
  3. ¿Puede determinar una relación entre el punto a) y los puntos b) a e)?

Si eligió SPI:

d. Tiempo mínimo de seteo para CS (setup time - inicio de trama).

e. Tiempo mínimo de mantenimiento para CS (hold time - fin de trama).

f. Tiempo mínimo para CS válido (tiempo mínimo entre tramas).

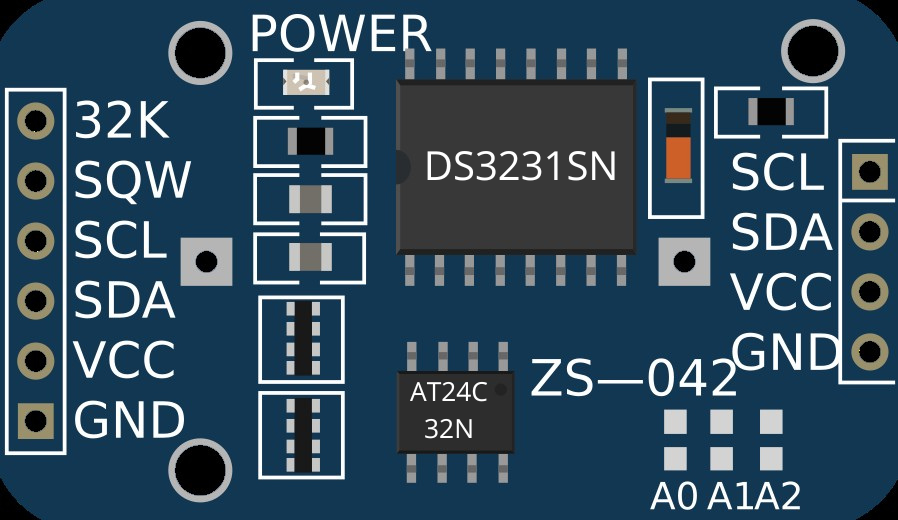
g. Tiempo máximo de mantenimiento de datos (input hold time)

h. ¿Puede determinar una relación entre el punto a) y los puntos b) a g)?

1. Incluya diagramas lógicos de una operación de escritura y una de lectura, explicando brevemente los significados de los distintos bits que componen la trama (i.e.: bits de dirección, de lectura/escritura, de dirección lógica, de datos, etc).
2. Dentro de la documentación del BSP para STM32, identifique el grupo de funciones correspondientes a las operaciones de comunicación con I2C o SPI según corresponda. Agregue los prototipos y explique brevemente el funcionamiento de las rutinas que facilita la HAL.

(\*)https://www.st.com/resource/en/user\_manual/dm00105879-description-of-stm32f4-hal-and-lowlayer-drivers-stmicroelectronics.pdf

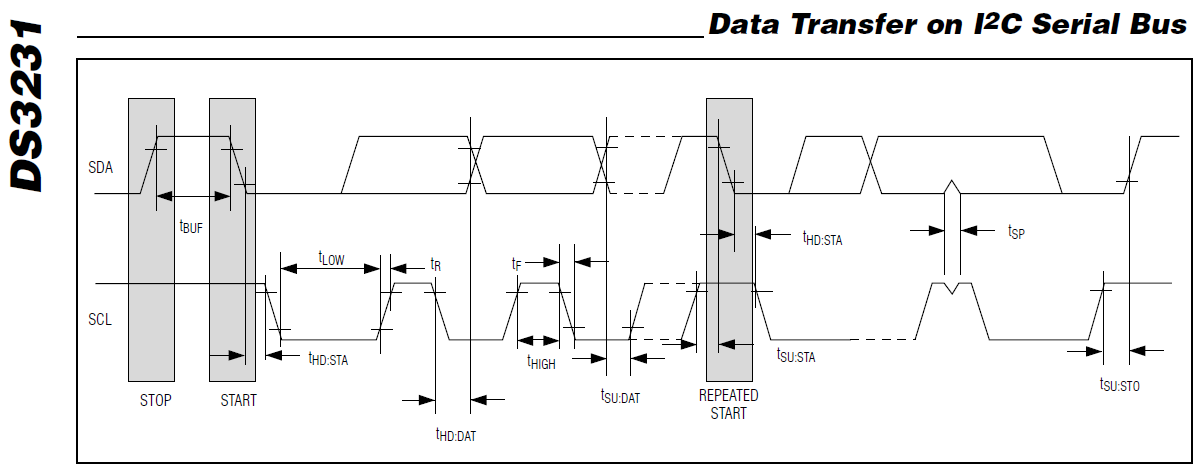
## 1. Descripción del RTC DS3231 y hoja de datos.

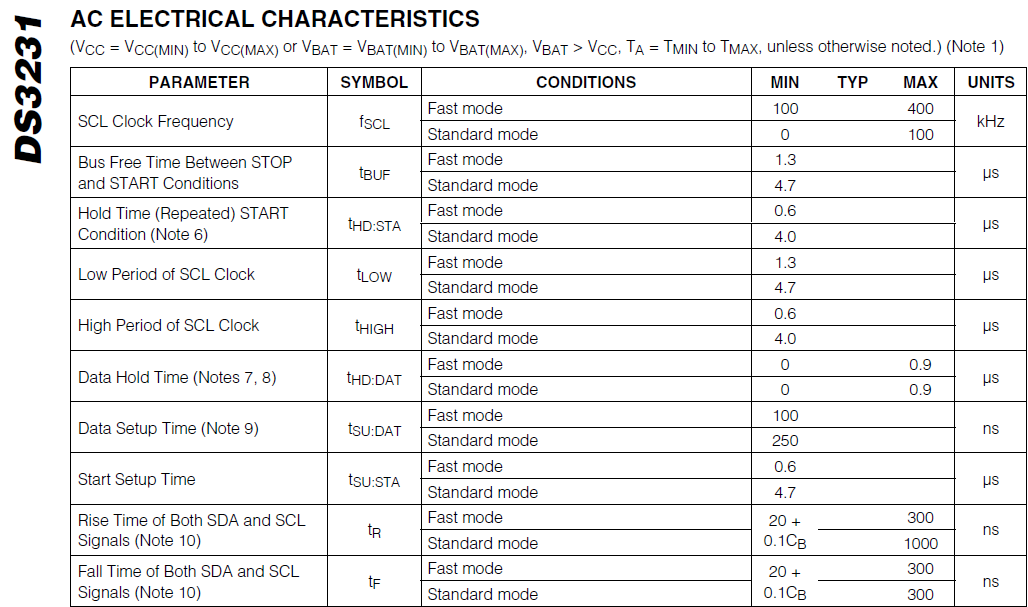


El módulo RTC (Real Time Clock) DS3231 es un circuito integrado que se comunica mediante el protocolo **I2C (funciona hasta 400 kHz)**, el cual permite mantener un registro preciso de la hora y fecha, incluso aun cuando el dispositivo principal (STM32) se encuentra apagado. Esto es debido a que posee una batería externa que se encarga de mantenerlo alimentado.   
Este RTC posee una gran precisión, permitiendo contar **segundos, minutos, horas, día de la semana, fecha, mes y año**, ajustando automáticamente los meses con menos de 31 días y los años bisiestos.

## 

## 2.Diagramas de tiempos de la comunicación

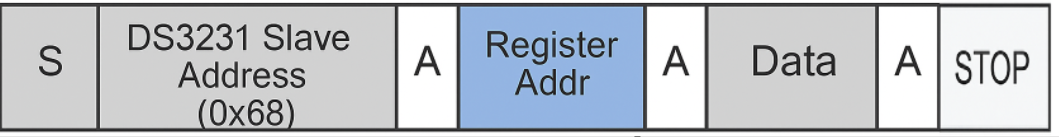


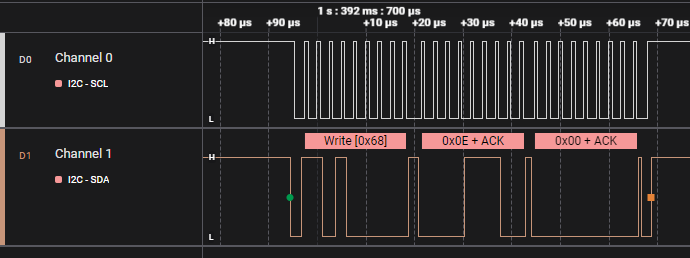


|  |  |  |
| --- | --- | --- |
| Parámetro | Fast-mode | Standard-mode |
| a) fSCL máx. | 400 kHz | 100 kHz |
| b) tHIGH mín. | 0.6 µs | 4.0 µs |
| c) tLOW mín. | 1.3 µs | 4.7 µs |
| d) tR máx. | 300 ns | 1000 ns |
| e) tF máx. | 300 ns | 300 ns |

## 3. Descripción de la trama I2C para Escritura/Lectura

### Escritura de un registro



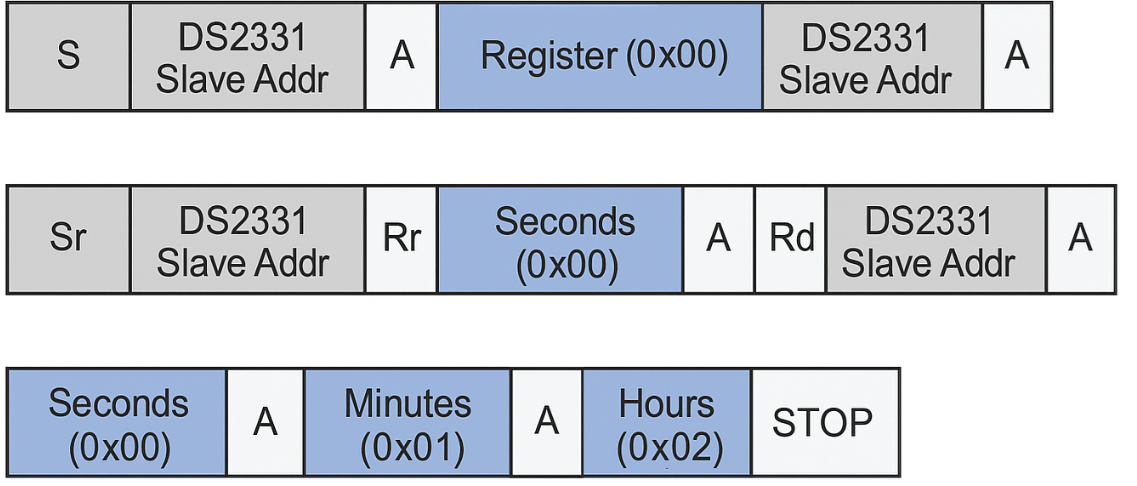


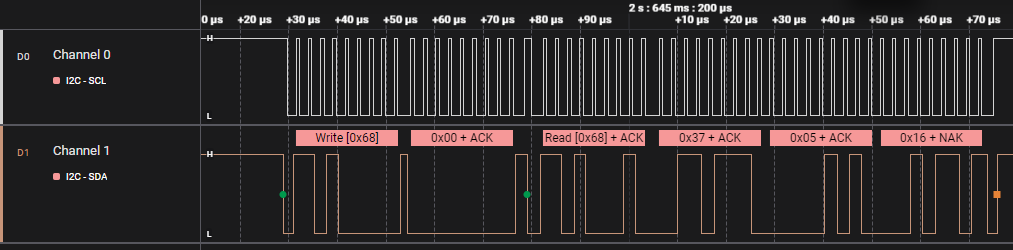
Img.1 - Ejemplo de escritura del registro de Control

#### **Descripción de cada campo**

* **S:** Condición de start generada por el Master (STM32).
* **DS3231 Slave Address:** Dirección fija → 0x68 (7 bits). + **Wr:** Bit R/W = 0 → Operación de escritura.
* **Register Addr:** Dirección registro a escribir. (Por ejemplo, 00h → para escribir los segundos).
* **Data:** Byte con los datos a escribir en el registro.
* **ACK / NACK:**
  + El Slave envía ACK tras luego del byte escrito.
* **STOP:** Señal de finalización de la comunicación.

### Lectura de un registro





Img.2 - Ejemplo de lectura de la hora

#### **Descripción de cada campo**

* **S:** Condición de start generada por el Master (STM32).
* **DS3231 Slave Address:** Dirección fija → 0x68 (7 bits).
* **Wr:** Bit R/W = 0 → Operación de escritura.
* **Register:** Dirección registro a leer (0x00 → segundos).
* **Sr:** Repeated Start.
* **Rd:** Bit R/W = 1 → Operación de lectura.
* **Seconds / Minutes / Hours:** Bytes enviados por el DS3231.
* **ACK / NACK:**
  + Master envía ACK tras cada byte leído para continuar.
  + En el último byte, el Master envía NACK para indicar fin de lectura.
* **STOP:** Señal de finalización de la comunicación.

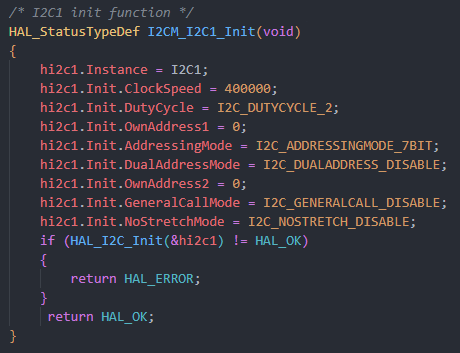
## 4. Funciones de la HAL utilizadas.

### Funciones propiamente de la hal de STM

|  |  |
| --- | --- |
| Tipo | Funciones |
| Inicialización/Apagado | * **HAL\_StatusTypeDef** HAL\_I2C\_Init(I2C\_HandleTypeDef \*hi2c); * **HAL\_StatusTypeDef** HAL\_I2C\_DeInit(I2C\_HandleTypeDef \*hi2c); |
| Transferencia de Datos | * **HAL\_StatusTypeDef** HAL\_I2C\_Master\_Transmit(I2C\_HandleTypeDef \*hi2c, uint16\_t DevAddr, uint8\_t \*pData, uint16\_t Size, uint32\_t Timeout); * **HAL\_StatusTypeDef** HAL\_I2C\_Master\_Receive(I2C\_HandleTypeDef \*hi2c, uint16\_t DevAddr, uint8\_t \*pData, uint16\_t Size, uint32\_t Timeout); |
| Acceso a Registros | * **HAL\_StatusTypeDef** HAL\_I2C\_Mem\_Read(I2C\_HandleTypeDef \*hi2c, uint16\_t DevAddr, uint16\_t MemAddress, uint16\_t MemAddSize, uint8\_t \*pData, uint16\_t Size, uint32\_t Timeout); |
| Utilidad | * **HAL\_StatusTypeDef** HAL\_I2C\_IsDeviceReady(I2C\_HandleTypeDef \*hi2c, uint16\_t DevAddr, uint32\_t Trials, uint32\_t Timeout); * **uint32\_t** HAL\_I2C\_GetError(I2C\_HandleTypeDef \*hi2c); |

#### FUNCIONES IMPLEMENTADAS EN EL DRIVER DE I2C MASTER

#### - I2CM\_I2CM1\_Init(void)

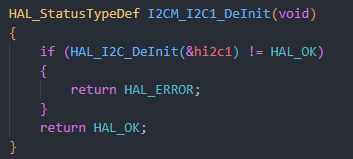


#### Descripción

Configura el I2C1 a 400 kHz (Fast Mode), 7-bit, sin dual-address, sin general call, sin clock stretching y llama luego a HAL\_I2C\_Init ().

Esta función es autogenerada por el ***STMCubeMX*** al momento de habilitar el periférico I2C1.

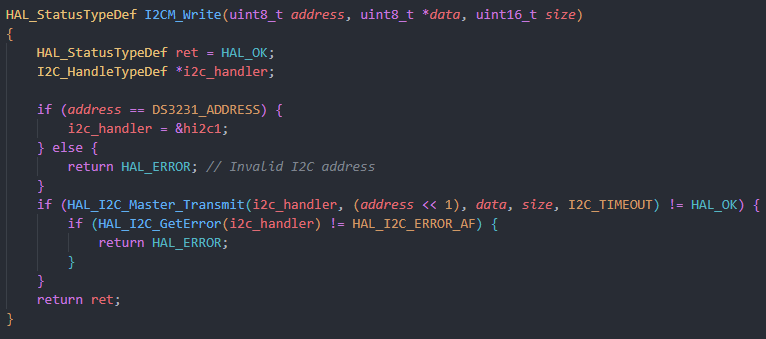
#### - I2CM\_I2CM1\_DEInit(void)



#### Descripción

Básicamente, se encarga de deshabilitar el periférico I2C1 y poner los GPIOs en estado de reset.

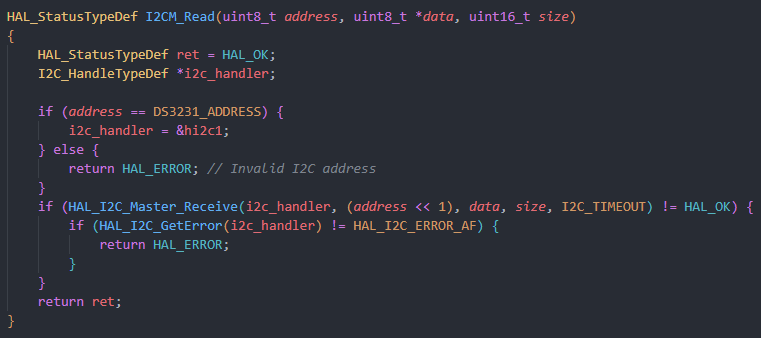
#### - I2CM\_Write(address, \*data, size)



#### Descripción

Se encarga de realizar la escritura maestro a esclavo (llamando a HAL\_I2C\_Master\_Transmit). El parámetro address es **0X68 para el DS3231 y se desplaza << 1** al llamar a la HAL.

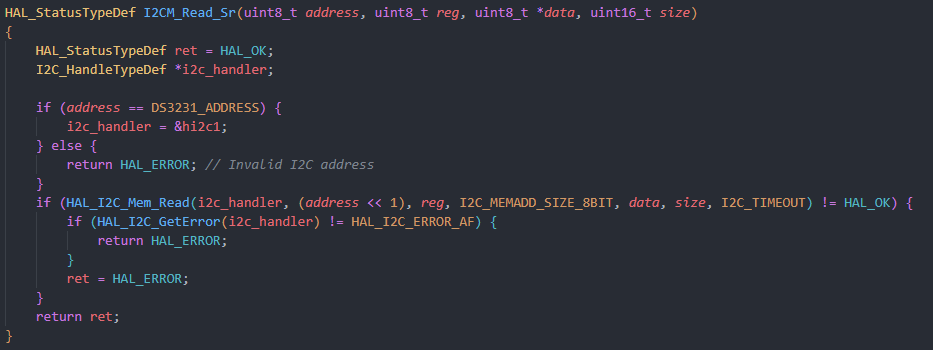
#### - I2CM\_Read(address, \*data, size)



#### Descripción

Se encarga de realizar la lectura desde el maestro al esclavo, utilizando HAL\_I2C\_Master\_Receive.

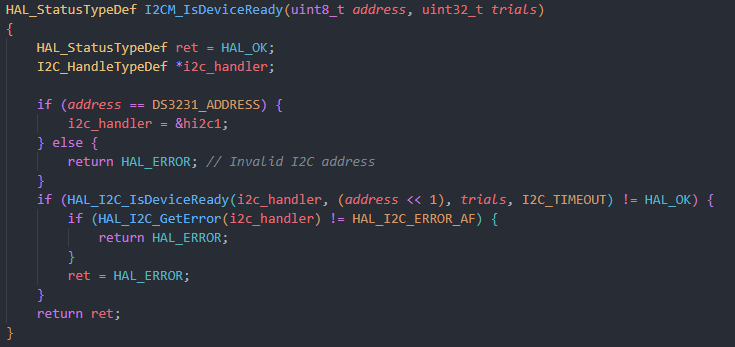
#### - I2CM\_Read\_Sr(address, reg, \*data, size)



#### Descripción

Se encarga de realizar la lectura de registro con *repeated start* desde el maestro al esclavo, utilizando HAL\_I2C\_Mem\_Read. Se utiliza para este dispositivo ya que primero se debe escribir la dirección del registro que queremos leer y, sin liberar el bus, obtenemos la información del registro desde el esclavo.

#### - I2CM\_ IsDeviceReady(address, trials)



#### Descripción

Se encarga de escribir a una dirección del bus de I2C para verificar presencia del dispositivo.

# ANEXO

### **1. Datasheet del Dispositivo**

**Enlace de acceso:**  
[DS3231 – Reloj de Tiempo Real (RTC)](https://www.alldatasheet.com/datasheet-pdf/pdf/112132/DALLAS/DS3231.html)

### **2. Repositorio del Proyecto en GitHUB**

En el siguiente enlace se encuentra mi repositorio de GitHub, donde aparecen disponibles los archivos y la documentación utilizada para el desarrollo de este proyecto.

**Enlace de acceso:**  
[Repositorio GitHub](https://github.com/matuuted/RTC_DS3231)

### **3. Documentación Doxygen de la API**

La documentación de la API para el driver del DS3231 se generó automáticamente con Doxygen y se encuentra disponible en el siguiente enlace:

**Enlace de acceso:**  
[Doxygen – API DS3231](https://matuuted.github.io/RTC_DS3231/)