FRVM de la Universidad Tecnológica Nacional
Departamento de Electrónica
Cátedra Trabajo Final de Grado

Control de acceso mediante RFID

Trabajo Final de Grado para obtener el título de Ingeniero en Electrónica

Autor/es:

Martini Luciano

AÑO 2019

Acreditación:

Fecha:

Comité evaluador:

Presidente:

1° vocal:

2° vocal:

Dedicatorias:

A mis padres, Marcela y Mario por darme la oportunidad de estudiar y brindarme todo su apoyo incondicional, a ellos le debo todo lo que soy.

A mi hermana, Sofía, abuelos, tíos y primos por acompañarme a lo largo de la carrera.

Especialmente a mi novia, Lucía, que caminó a mi lado todos estos años dándome su cariño y fuerza para seguir adelante.

Agradecimientos:

A mi familia, por el apoyo y afecto durante tantos años.

A docentes, quienes día a día nos brindaron sus conocimientos y experiencias.

Al departamento de Electrónica el cual siempre estuvo con buena predisposición.

A compañeros y amigos de la carrera quienes compartieron tantos años.

Memoria Descriptiva:

En el marco del Trabajo Final de Grado de la carrea Ingeniería Electrónica, se realizó un sistema de registro y control de acceso. El dispositivo fue pensado para ser implementado en nuestra facultad como medida de seguridad utilizada tanto por alumnos, docentes y personal no docente de nuestra institución.

El proyecto se realizó mediante la utilización de tarjetas electromagnéticas del tipo RFID (Radio Frequency Identification), mediante las que se permite o no el acceso a aulas, laboratorios y otras habitaciones de la facultad.

Se utilizaron también módulos lectores/grabadores de dichas tarjetas. Las tarjetas (tags) son utilizadas para que un determinado usuario pueda o no acceder a una habitación determinada. Los tags son utilizados como identificadores únicos que cuentan con un código que determinará el nivel de acceso y privilegios que tendrá cada usuario, además estos tags contienen los legajos correspondientes a los alumnos o personal que deseen ingresar a una determinada habitación quedando registrado el día y horario de ingreso en una memoria externa de fácil acceso.

Para realizar la grabación de las tarjetas se utilizó un teclado matricial 4x4, de esta manera se ingresa el legajo de los usuarios quedando grabado en la tarjeta para su posterior uso. Este es un proyecto secundario, ya que las tarjetas al ser grabadas y entregadas no será necesario volver a utilizarlo nuevamente.

Índice

1. Introducción.

Debido al avance tecnológico se observa un notorio crecimiento en los sistemas de acceso, los hay con diferentes sensores o módulos, como son los de huella dactilar, código de seguridad, cámara de video, tarjetas electromagnéticas RFID, voz, etc. Haciendo un balance costo/beneficio y considerando lo práctico y rápido del uso se decidió utilizar tarjetas electromagnéticas junto al módulo RFID.

Todo usuario tendrá es su poder una tarjeta electromagnética, la cual tendrá en su interior dos códigos grabados, el primero será utilizado para la comparación (entre microcontrolador y tag) y posterior concesión, o no, de acceso a las determinadas habitaciones. El segundo código será el que identifique al usuario correspondiente de dicha tarjeta, este será su propio número de legajo.

La implementación se realizará con el módulo RFID MFRC522 que trabaja en alta frecuencia (13,56Mhz), este será utilizado para la lectura y escritura de las tarjetas electromagnéticas. Para realizar la escritura de las mismas, se implementará un teclado matricial 4x4, con el cual podemos fácilmente ingresar el número de legajo a grabar en la tarjeta RFID.

El tipo de tarjeta utilizada es del tipo "MIFARE Classic" que cuentan con una memoria de 1Kb o 1024 bytes divididos en 16 sectores de 64 bytes cada uno.

La recopilación de datos se hará a través de una memoria externa micro SD, en ella se creará un archivo de extensión .xlsx (Excel) de fácil lectura, donde se generará un registro de las personas que ingresaron al lugar junto con la hora y fecha de acceso.

Para la fecha y hora, se utilizará un reloj en tiempo real, este posee una alimentación auxiliar para que siga funcionando en caso de cortes del suministro de energía, manteniendo siempre actualizada su información.

1.1 Diagrama en bloque

A continuación, se mostrará un diagrama en bloque simplificado del funcionamiento del control de acceso y se presentará un resumen de las funciones principales de cada uno. El proyecto se divide en dos partes (lector – grabador) siendo el lector el principal ya que con él se realizarán más interacciones entre el usuario y el control de acceso. La parte secundaria será el grabador ya que él será utilizado solo una vez para realizar la grabación de las tarjetas.

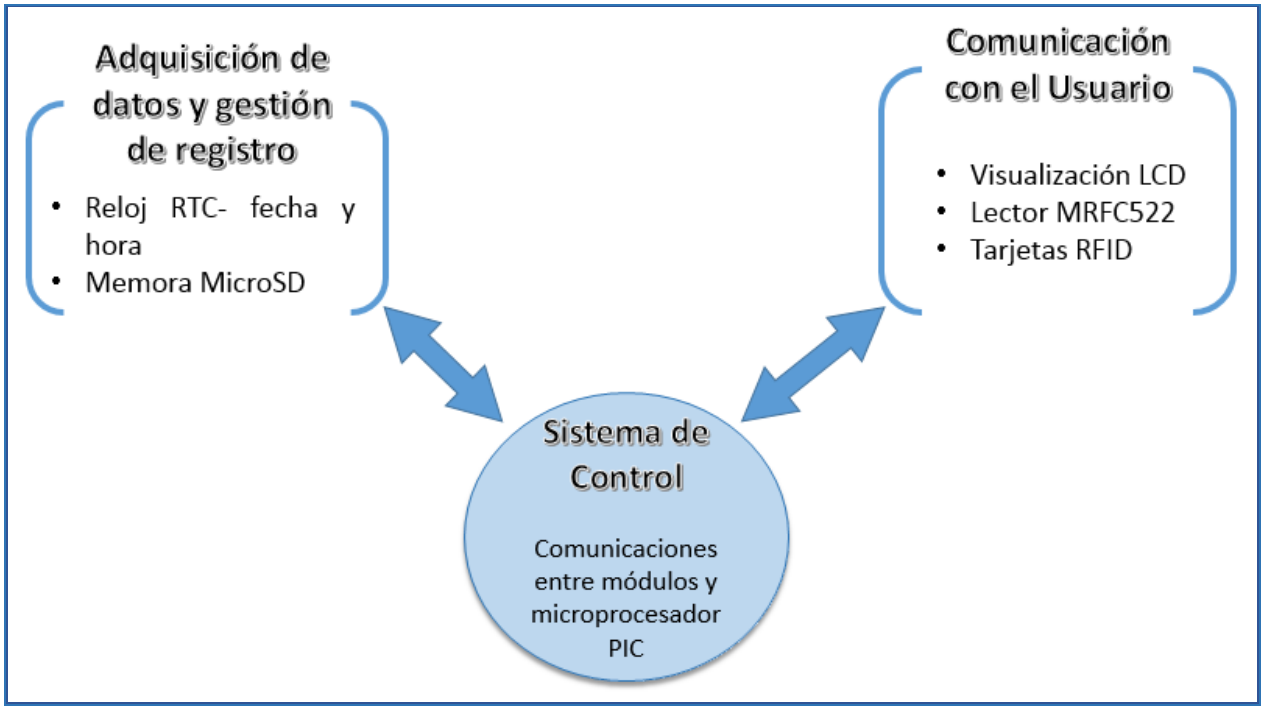


Diagrama.1 Diagrama en bloque principal

Para entender y desarrollar el control de acceso se divide el análisis en tres grandes grupos: Sistema de control, Comunicación con el usuario y Adquisición de datos y gestión de registro.

- *Sistema de control:*

El componente principal es el microcontrolador PIC, éste se encarga de gestionar y controlar todos los periféricos que serán utilizados. El PIC es el encargado de recibir y trasmitir los datos para su posterior uso.

- *Comunicación con el usuario:*

En esta sección se llevan a cabo todas las tareas que interactúan con el usuario, ya sea la utilización de las tarjetas electromagnéticas, el lector – grabador o el display LCD. También en este apartado entra le teclado matricial 4x4, pero este solo será necesario en la instancia de grabación.

- *Adquisición de datos y gestión de registro:*

Es aquí donde una vez obtenidos los datos del usuario se adquiere del RTC la fecha y hora para luego crear un registro de los usuarios que acceden a las determinadas habitaciones.

A continuación, se detallarán los componentes y módulos utilizados en los dos sistemas, lector de tarjetas electromagnéticas y grabador de tarjetas electromagnéticas.

Lector:

Para este sistema se utilizará la siguiente lista de componentes, donde el microcontrolador utilizado será el PIC18F4550 este es de mayor capacidad y prestaciones que el utilizado en la parte de grabación ya que aquí se utilizan módulos con mayor frecuencia de comunicación y trabajando todos a la vez.

- *Tarjeta Electromagnética (tags): Esta tendrá grabado dos códigos, uno será el que indique donde podrá, o no, ingresar y el otro será el que indique a que usuario pertenece la tarjeta.*
- *MifareRC522: Es el lector – grabador RFID encargado de la modulación/demodulación de la información y los datos que se trasmiten.*
- *Microcontrolador: Es el corazón del sistema, es el encargado de la interfaz con el usuario, además de ser el que graba y lee la información.*
- *RTC: Es un reloj en tiempo real, el cual dispone de una batería extra que será la encargada de mantener la información actualizada.*
- *Memoria SD: Se utilizará una memoria micro SD que será la encargada de guardar todos los datos correspondientes a los accesos.*
- *Visualización: Se trata de un LCD 16x2.*

.

Grabador:

El sistema grabador de tarjetas utiliza los mismos componentes y el principio de funcionamiento del módulo lector, pero se debe añadir un teclado matricial 4x4 para ingresar los datos. Para esta parte del proyecto se utilizará como sistema de control un PIC diferente al circuito del lector, ya que las exigencias del mismo son menores. En este sistema no será necesario utilizar el RTC y el módulo micro SD.

- *Teclado: Teclado del tipo matricial 4x4 con el cual se ingresará los datos a grabar en la tarjeta electromagnética.*

Dentro de los posibles sensores y módulos existentes en el mercado se utilizará el módulo RFID MFRC522 que es un lector – grabador de tarjetas electromagnéticas. Este es un módulo económico y fiable.

El desarrollo se llevará a cabo en un microcontrolador PIC 18F4550 que este será utilizado para el diseño del lector RFID ya para este desarrollo las características del PIC de la serie 18 son las que mejor se adecuan. El desarrollo del grabador se hará en un PIC de la gama 16 que el 16F887, el cual tiene menores prestaciones que el PIC18, pero cumple con los requisitos del sistema.

2. Objetivos

El desarrollo del proyecto cuenta con una serie de objetivos, ellos son:

- *Desarrollar e implementar un sistema de acceso que sea fiable, fácil de utilizar, económico y que cumpla con cierto nivel de seguridad.*
- *Generar un sistema de adquisición de datos a través de una tarjeta MicroSD*
- *Crear un grabador de tarjetas de fácil implementación para la utilización de los posibles usuarios.*

3. Comunicación con el usuario

3.1 RFID y Tarjetas electromagnéticas

La identificación por radiofrecuencia o RFID (Radio Frequency IDentification) en la actualidad fue creciendo y tomando la delantera en los sistemas de identificación, su fácil implementación y bajo costo lo convierten en la tecnología más utilizada dentro del mercado. Su uso abarca desde sistemas de seguridad, acceso de personal, identificación de productos, entre otras aplicaciones.

El RFID es un conjunto de tecnología inalámbrica diseñada para obtener información almacenada en un dispositivo denominado "tag". El principio de funcionamiento consiste en un módulo RFID y un TAG. El TAG se pasa cerca del módulo y este tiene la capacidad de enviar la información guardada en su memoria al módulo.

El TAG es un dispositivo transpondedor, vienen en diferentes modelos, los más comunes son tarjetas y llaveros, pero también vienen como etiquetas adhesivas e incluso ya incorporados a algunos productos. Los TAGs tienen internamente una antena y un microchip, encargados de generar todos los procesos de comunicación, la energía necesaria para el traspaso de información se obtiene de la señal de radiofrecuencia, esta energía es pequeña pero suficiente para operar el CI (circuito integrado) CMOS. Estos son comúnmente llamados TAGs pasivos, ya que no necesitan de una batería para funcionar por eso es necesario que el TAG se acerque a una distancia generalmente menor a 10cm del módulo RFID. Cabe destacar que existen TAGs activos, estos traen una batería incorporada proporcionando un alcance de varios metros de distancia (10m – 100m).

Otra manera de clasificar a las tarjetas magnéticas es por su memoria, cada TAG cuenta con una memoria y se clasifican en tres grupos.

- *Solo lectura: Esto es, que no podremos modificar la información que lleve la tarjeta magnética, el código se escribió durante la fabricación y no podremos modificarlo*
- *Lectura – escritura: Donde en este caso se puede modificar la información que llevara guardada la memoria.*
- *Anticolisión: Se trata de tarjetas especiales que permiten ser identificadas varias al mismo tiempo.*

El módulo RFID está compuesto por un transceptor, una antena y un decodificador. En primer lugar, el módulo emite una señal para establecer la comunicación, esta señal es recibida por los TAGs que estén dentro del rango de comunicación, la cual responden trasmitiendo la información que tienen almacenada, finalmente es captada y decodificada por el receptor.

El RFID opera en cuatro bandas de frecuencias, siendo la frecuencia más común 13.56Mhz

- *Baja frecuencia: 125 KHz – 134.2KHz. Para control de animales, llaves de automóviles.*
- *Alta frecuencia: 13.56MHz. Control de acceso, control de artículos en tiendas.*
- *Ultra alta frecuencia (UHF): 868MHz – 959MHz.*
- *Microondas: 2.45GHz.*

3.1.1 MIFARE RC522

Mifare es una tecnología de tarjetas inalámbricas propiedad de NXP Semiconductores. Es uno de los estándares más utilizados como tarjetas inteligentes sin contacto (TSIC) cumpliendo con los estándares ISO/ IEC 14443 y/o ISO/ IEC 15693.

El lecto grabador Mifare RC522 es un módulo RFID que incorpora la comunicación con el bus SPI, bus I2C y UART, por lo que es de fácil implantación con cualquier microcontrolador, soporta tarjetas del tipo Mifare Classic, Mifare S50, S70, Mifare UltraLigth, Mifare Pro y Mifare Desfire.

El módulo funcia a una frecuencia de 13.56MHz y tiene una distancia de lectura aproximada de 5cm, dependiendo de la antena.

A continuación, se muestra una figura del modelo eléctrico del lector y grabador RFID. De la figura se pueden obtener los siguientes parámetros.

Tensión de entrada digital (DVDD), tensión de entrada analógica (AVDD), entrada de tensión (PVDD) y entrada de tensión para los trasmisores (TVDD).

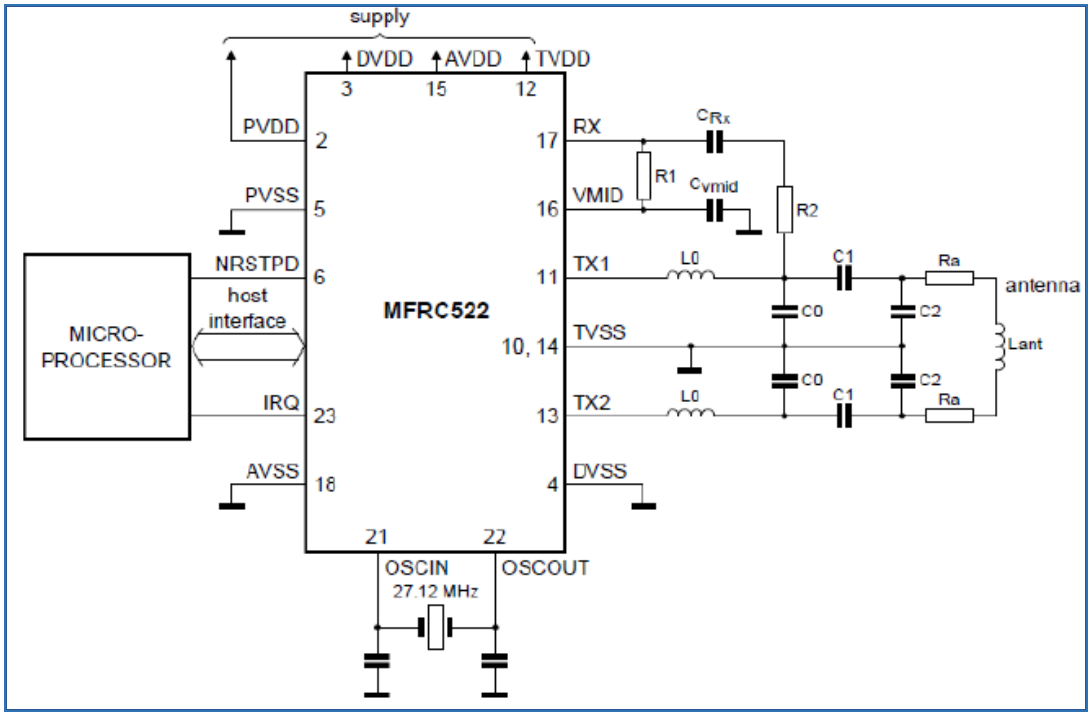


***Fig. 1** Circuito Electrónico MFRC555*

La tensión especificada DVDD puede estar en el rango de los 2,5V_{cc} y 3,6V_{cc} esta es la misma tensión recomendada para AVDD y TVDD.

El Circuito Integrado (CI) cuenta con una entrada de alimentación y su valor recomendado es de 3,3V_{cc}. El valor de tensión para PVDD, se encuentra en un intervalo de 1,6V_{cc} a 3,6V_{cc} y su valor recomendado es de 1,8V_{cc}.

Entre los pines OSCOUT y OSCIN se encuentra el circuito oscilador encargado de generar una frecuencia de reloj de 27,12Mhz con un ciclo de trabajo del 50%. Para el correcto funcionamiento y optima trasferencia de datos, en los pines TX1 y TX2 es necesario la implementación de un circuito de acondicionamiento (filtrado y adaptación) de la señal proveniente de la antena, dicho circuito está compuesto por un conjunto de componentes pasivos, la frecuencia de la señal modulante entregada por los pines mencionados es de 13,56Mhz, esta es derivada de la frecuencia del reloj (27,12Mhz). El pin RX es el encargado de recibir la señal de RF de respuesta generada por la etiqueta "tag".

El pin de interrupción IRQ no se utiliza debido a que el MFRC55 trabaja como esclavo y el microcontrolador es el encargado de decidir sobre el flujo de datos enviados y recibidos.

3.1.1.1 Diagrama en bloque simplificado

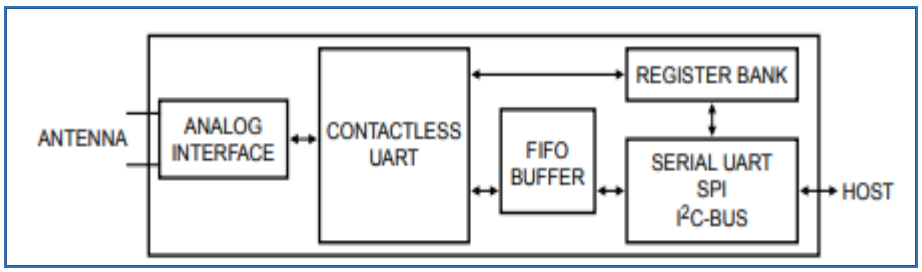


Diagrama. 2 *diagrama en bloque simplificado*

- *La interfaz analógica maneja la modulación y demodulación de las señales analógicas.*
- *El UART gestiona los requisitos del protocolo para la comunicación además de realizar la detección de errores mediante la comprobación de redundancia cíclica (CRC) en función de la paridad y la velocidad de transferencia.*
- *El buffer FIFO garantiza una transferencia de datos rápida entre el microcontrolador y la etiqueta debido a que el mismo puede gestionar flujos de datos de hasta 64 bytes de longitud sin la necesidad de tener en cuenta limitaciones de tiempo*
- *Buses de comunicación del tipo SPI, I2C y UART.*
- *Banco de registro.*
- *Se implementan varias interfaces de host para satisfacer todas las necesidades de los potenciales clientes.*

3.1.1.2 El bus SPI RFID

El MFRC522 soporta la comunicación serie SPI (Serial Peripheral Interface), que es la utilizada para la comunicación con el microcontrolador PIC 18f4550. La interfaz puede manejar velocidades de hasta 10Mbit/s. Cuando se comunica con el HOST, el MFRC522 actúa como esclavo, recibiendo datos del HOST para ajustar los registros, establecer un reloj de sincronismo y enviar y recibir la información relevante durante la comunicación.

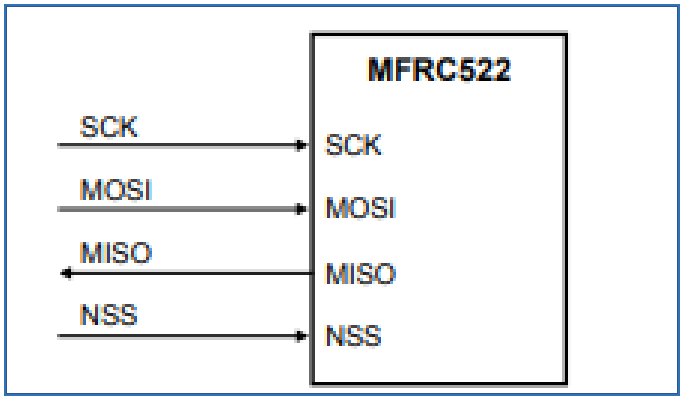


Fig. 2 *Líneas de comunicación SPI*

La señal SCK (reloj de sincronismo) debe ser generada por el maestro, en cada pulso del reloj se envían o reciben datos. La comunicación de datos desde el maestro hacia al esclavo se da por el pin MOSI (Master Output Salve Input). La línea MISO (Master Input Salve Output) se utiliza para enviar datos desde el MFRC522 al maestro. La línea NSS es la línea de activación del esclavo.

Los datos de las líneas MISO y MOSI deben ser establecidos en los flancos ascendentes del SCK y es posible cambiar la configuración para establecer la comunicación en los flancos descendentes del reloj.

Los datos son proporcionados por el MFRC522 en el flanco de reloj descendente y son estables durante el flaco de reloj ascendente. Los bytes de datos en ambas líneas MOSI y MISO se envían con el MSB primero.

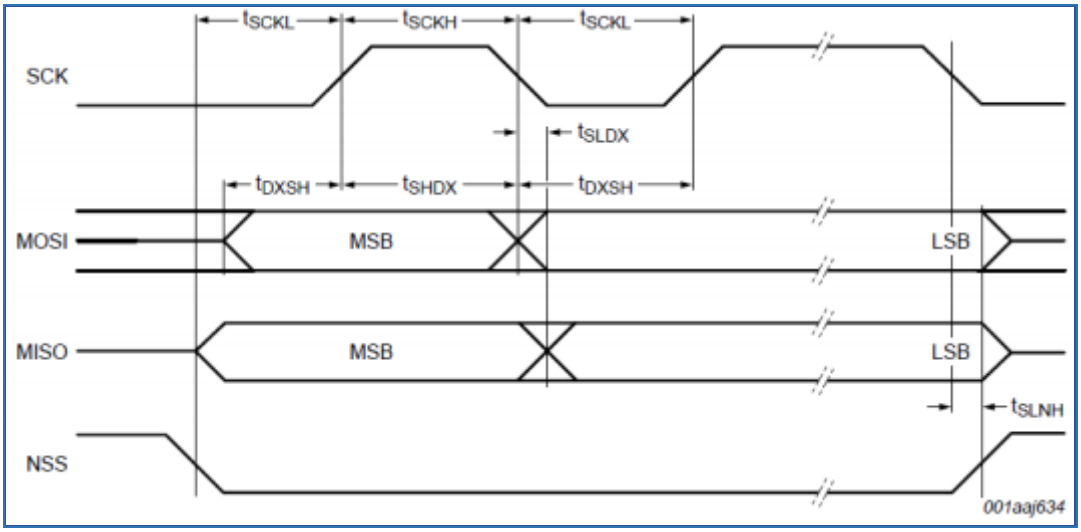


Fig. 3 *Esquema temporal de la interfaz SPI*

- *SPI Lectura de datos*

Línea	Byte 0	Byte 1	Byte 2	...	Byte n	Byte n+1
MOSI	Dirección 0	Dirección 1	Dirección 2	...	Dirección n	00
MISO	-	Dato 0	Dato 1	…	Dato n-1	Dato n

Tabla 1 *Orden de bytes de lectura MOSI y MISO*

La lectura de datos utilizando la comunicación SPI requiere que se utilice el orden mostrado en la tabla.

- *SPI Escritura de datos*

Línea	Byte 0	Byte 1	Byte 2	…	Byte n	Byte n+1
MOSI	Dirección 0	Dato 1	Dato 2	…	Dato n-1	Dato n
MISO	-	-	-	…	-	-

Tabla 2 *Orden de bytes de escritura MOSI y MISO*

Para garantizar la escritura de datos se requiere el orden de bytes que se muestran en la tabla, es posible escribir hasta n bytes de datos enviando solo un byte de dirección. El primer byte es el que define tanto el modo como la dirección de la información.

- *Direcciones*

7 MSB	6	5	4	3	2	1	0 LSB
1=Lectura 0=Escritura	Direcciones						0

Tabla 3 *Bytes de dirección*

El byte de dirección debe cumplir con el síguete formato, el bit más significativo (MSB) del primer byte define el modo utilizado. Para leer los datos del MFRC522, el MSB debe estar a un nivel lógico alto (1). Para escribir los datos en el MFRC522, el MSB debe estar en un nivel lógico bajo (0). Los bits del 6 – 1 definen la dirección y el bit menos significativo (LSB) se establece en un nivel lógico bajo (0).

3.1.1.3 Características

- *Tensión de alimentación: 3.3V.*
- *Corriente de operación: 13 – 26mA*
- *Corriente de stand by: 10 – 13mA*
- *Corriente de sleep-mode: <80uA*
- *Corriente máxima: 30mA*
- *Frecuencia de opresión: 13.56 MHz*
- *Distancia de lectura: 0 – 5 cm*
- *Velocidad de datos máxima: 10Mbit/s*

3.1.2 Tarjetas Electromagnéticas (TAGs)

La tarjeta electromagnética "tag" utilizada es del tipo pasiva, cuenta con un microchip y una antena encargados de generar la transmisión y recepción de datos. Estas tarjetas al ser de tipo pasivas no cuentan con una alimentación propia, sino que utilizan energía del campo electromagnético generado por el MFRC555.

El chip utilizado es el *MF1S503* fabricado por NXP SEMICONDUCTORES. A continuación, se muestra un diagrama en bloques del chip mencionado.

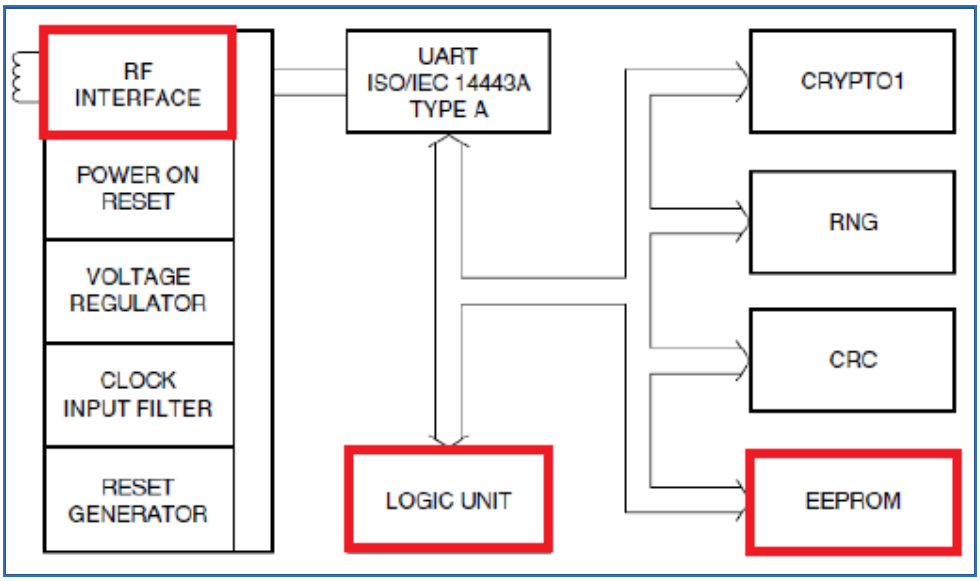


Diagrama. 3 *diagrama en bloques simplificado MF1S503*

Dentro del diagrama simplificado del chip podemos destacar tres bloques. La INTERFACE RF, LOGIC UNIT y EEPROM. La interfaz RF es la encargada de comunicar al lector con la tarjeta, para esta comunicación es necesario que la tarjeta este siempre dentro del campo electromagnético, ya que este es utilizado para su alimentación. Gracias a la unidad lógica se puede llevar a cabo acciones de incremento o decremento de los datos previamente almacenados.

El MFRC522 suele suministrase con tarjetas o llaveros del tipo MIFARE Classic que contiene una memoria de 1Kbytes. Esta memoria está dividida en bloques, con mecanismos simples de acceso a la información.

La memoria está dividida en 16 sectores de 64 bytes, protegidos con dos claves llamadas A y B. Cada una puede ser programada individualmente para permitir o bloquear operaciones de lectura – escritura.

Cada sector reserva cierta cantidad de memoria para las claves A y B, por lo que este espacio dedicado no puede ser utilizado para otro tipo de operación. De esta manera la memoria útil de la tarjeta o llavero se ve reducida, quedando a disposición del usuario 752 bytes.

La memoria EEPROM de las tarjetas MIFARE Classic soportan más de 100.000 ciclos de lectura – escritura, y pueden mantener el dato en la memoria más de 10 años.

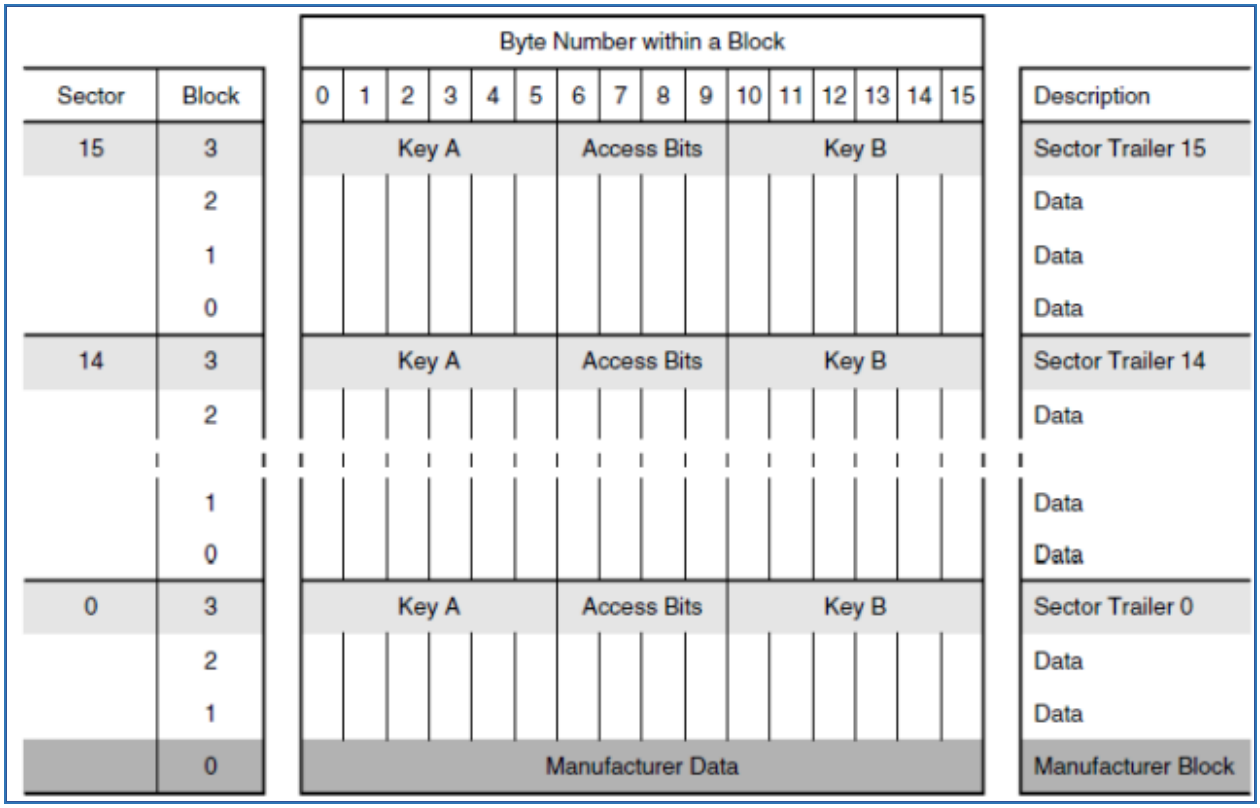


Fig. 4 *Mapeo de la memoria de una tarjeta MIFARE Classic*

Analizando la memoria del TAG podemos observar que el primer bloque de memoria (0) viene grabado de fábrica, en el podemos encontrar un UID único de cada tarjeta o llavero que está contenido en los primeros 4 bytes (0 – 3), el resto de los bytes grabados van a ser datos proporcionados por el fabricante.

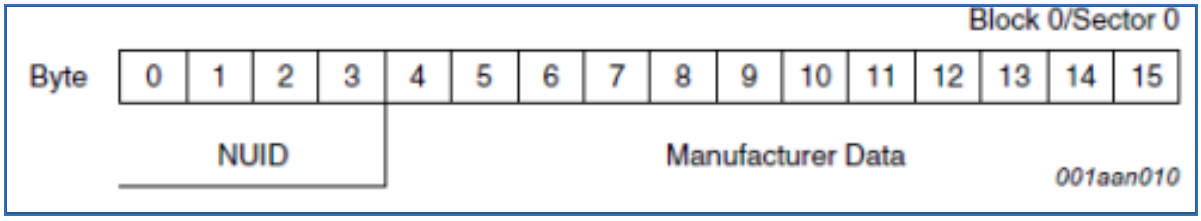


Fig. 5 *Bloque 0 de memoria*

En este caso al ser una tarjeta del tipo MIFARE Classic de 1K vamos a tener un "*Sector Trailer*" (Fig. 7) separados por 3 bloques de memoria útiles. En cada *Sector Trailer* encontramos las claves llamadas ***key A*** y ***key B***, donde para poder acceder a grabar o leer los "*Data Block*" antes tenemos que autentificar las claves A y B.

Las claves de acceso A y B son:

- **Key A= 00 00 00 00 00 00 en hexadecimal.**
- **Key B= FF FF FF FF FF en hexadecimal.**

Los llamados "Access Bits" (Bit de acceso), están formados por tres bits que son llamados C_1, C_2 y C_3, los cuales controlan los permisos de acceso a memoria para cada uno de los 4 bloques del sector. La identificación de los bits correspondientes a cada sector se hace manejando subíndices, en total se utilizan 12 bits para el manejo de los 4 bloques. En la siguiente tabla se muestra un ejemplo.

Bits de acceso	Bloque	Descripción
$C_{13}\, C_{23}\, C_{33}$	3	Sector trailer
$C_{12}\, C_{22}\, C_{32}$	2	Data
$C_{11}\, C_{21}\, C_{31}$	1	Data
$C_{10}\, C_{20}\, C_{30}$	0	Data

Tabla 4 *Condiciones de acceso*

En la fig.9 se muestra el módulo comercial RFID con sus correspondientes tarjetas electromagnéticas.

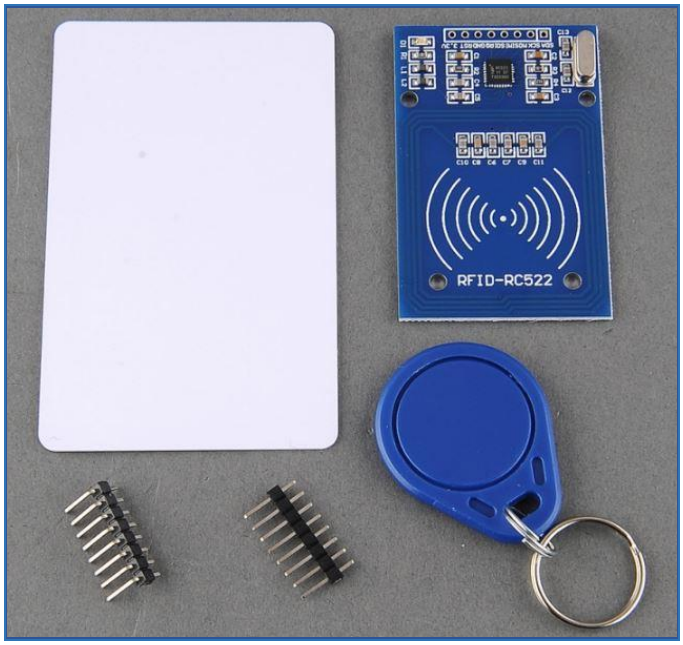


Fig.6 *Kit comercial MRC522*

3.2. Visualización

El sistema de comunicación con el usuario cuenta con una pantalla LCD de 16x2 caracteres, donde el usuario podrá observar el estado de la cerradura y ver si se concede o no el acceso. El dispositivo cuenta con 16 pines, de los cuales el pin 1 y el pin 2 son empleados en la fuente de alimentación (VSS y VCC) respectivamente, el pin 3 es para controlar el contraste de las letras en pantalla (VO) mediante la regulación de un potenciómetro, del pin 4 al 6 son pines de control, (RW, RS, E) del 7 al 14 se emplea el bus de datos (D0-D7), de los cuales los primeros 4 no son utilizados debido a que el mismo funciona solo con 4 bits y los últimos dos pines 15 y 16 se emplean en el control de la intensidad de la pantalla. En la siguiente figura se puede observar el circuito implementado.

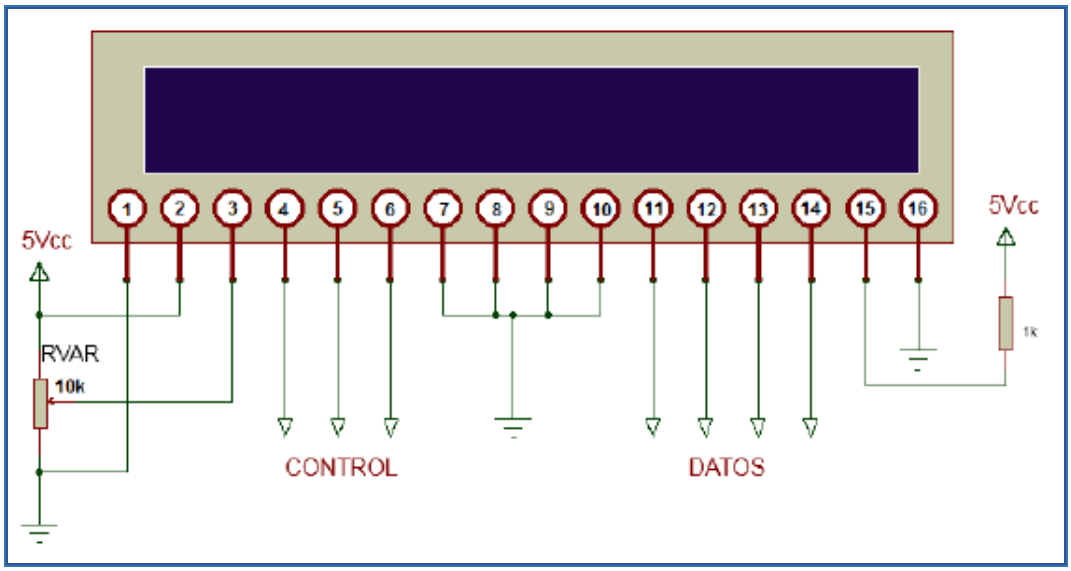


Fig.7 *Pines LCD 16x2*

3.3 Teclado Matricial 4x4.

El sistema de comunicación con el usuario cuenta además con un teclado matricial 4x4, Estos dispositivos agrupan los pulsadores en filas y columnas formando una matriz, disposición que da lugar a su nombre. Es frecuente una disposición rectangular pura de NxM columnas, esta disposición permite emplear un número menor de conductores para determinar la pulsación de las teclas.

A continuación, se mostrará el funcionamiento de un teclado matricial 4x4, pero es igualmente aplicable a otro tipo de configuración.

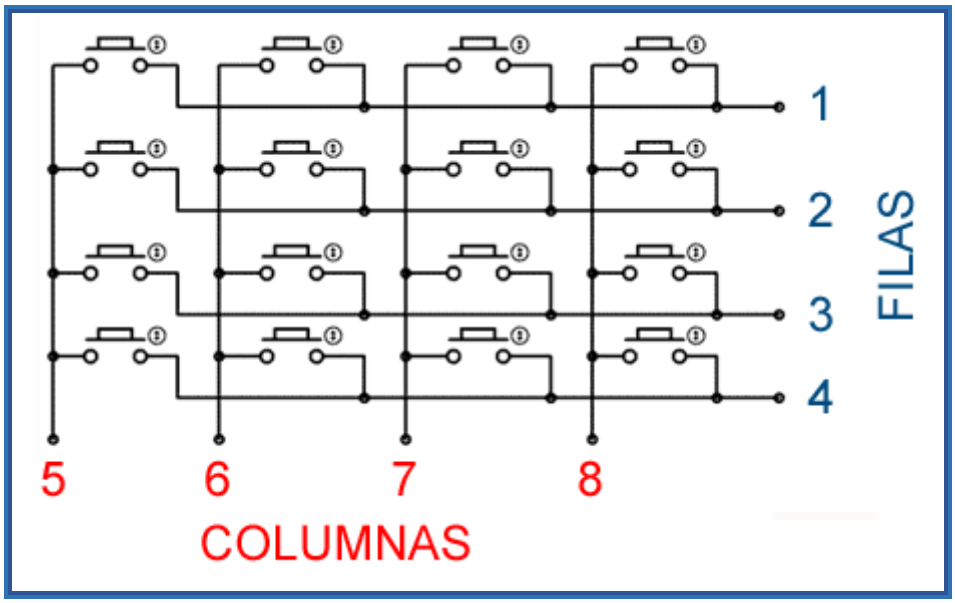


Fig.8 *Teclado matricial 4x4*

Al detectar la pulsación de la columna X y fila Y sabemos que se ha pulsado la tecla (X, Y). Internamente la disposición de los pulsadores es similar a la mostrada en la figura anterior. Para realizar una lectura tendremos que hacer un barrido por fila, en primer lugar, pondremos todas las filas a $5V_{cc}$ y determinaremos la entrada con una resistencia pull-up.

Progresivamente ponemos una fila a $0V_{cc}$ y leemos la entrada de la columna. Una vez realizada la lectura volvemos a poner a $5V_{cc}$, pasamos a la siguiente fila, y volvemos a realizar el proceso hasta recorrer todas las filas.

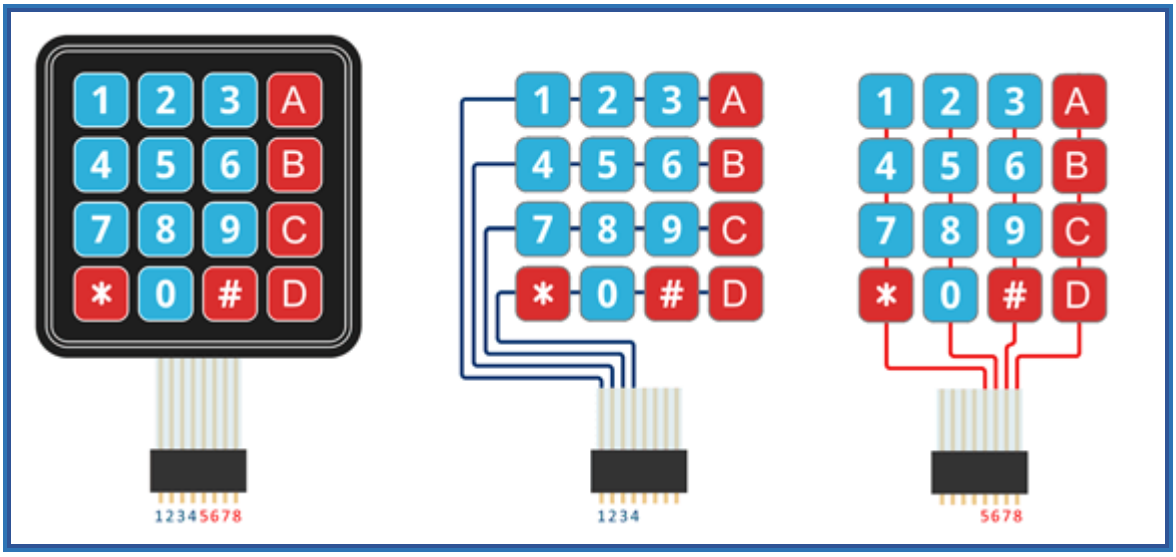


Fig. 9 Conexionado fila y columna

4 Adquisición de datos y gestión de registro

4.1 Memoria micro SD

Las tarjetas micro SD se han convertido en un estándar a la hora de hablar de almacenamiento masivo de datos en dispositivos electrónicos, desplazando a otros medios de almacenamiento, debido a su gran capacidad y pequeño tamaño. Por estos motivos han sido incorporadas en una gran cantidad de dispositivos, siendo en la actualidad un componente casi indispensable en smartphones, Tablet y otros elementos.

Dentro del mercado es posible encontrar diferentes tipos de lectores para las tarjetas de memoria, estos son de bajo costo y altamente confiables, gracias a su protocolo de comunicación SPI el cual es muy robusto y fiable ante la transición de datos. En el mercado existen dos tipos de lectores, los hay para memorias SD Estándar y Micro SD.

En ambos tipos de lectores, la lectura puede realizarse a través del bus SPI. Aunque pueden disponer de otros tipos de interfaces como el bus I2C o UART, normalmente es preferible y recomendable utilizar el bus SPI, ya que este tiene una mayor tasa de trasferencia de datos y es más confiable que los demás buses.

4.1.1 Características

- *Voltaje de trabajo: 4.5V ~ 5.5V.*
- *Corriente 0.2 ~ 200mA.*
- *Nivel eléctrico de la interfaz: 3.3V / 5V.*
- *Soporta micro SD hasta 2Gb.*
- *Soporta SDHC de hasta 32Gb.*
- *Formato de SD: Fat 16.*
- *Tarjetas SD o SDSC (Standard Capacity) – SDHC (High Capacity).*
- *No compatible con SDXC (Extended Capacity).*

4.1.2 Comandos usados con el bus SPI

Estos comandos son los utilizados para escribir y leer bloques de memoria. La estructura de los comandos costa de 6 Bytes.

Byte 1			Byte 2 – 5	Byte 6	
7	6	5 0	31 0	7	0
0	1	N° de comando	Valores de comando	CRC	1

Tabla 5 *Bytes bus SPI*

Los comandos utilizados en este trabajo son los siguientes:

- ***CMD0:*** *Coloca la tarjeta en inactividad.*
- ***CMD1:*** *Coloca la tarjeta en actividad.*
- ***CMD17:*** *Permite leer un sector de la tarjeta.*
- ***CMD24:*** *Indica el sector de inicio de escritura.*

Cuando hacemos el envío de un comando, la tarjeta responde a este enviando un byte de confirmación que nosotros debemos leer para saber si se concretó la operación.

El Byte a leer es el Byte 1 y esta es su estructura:

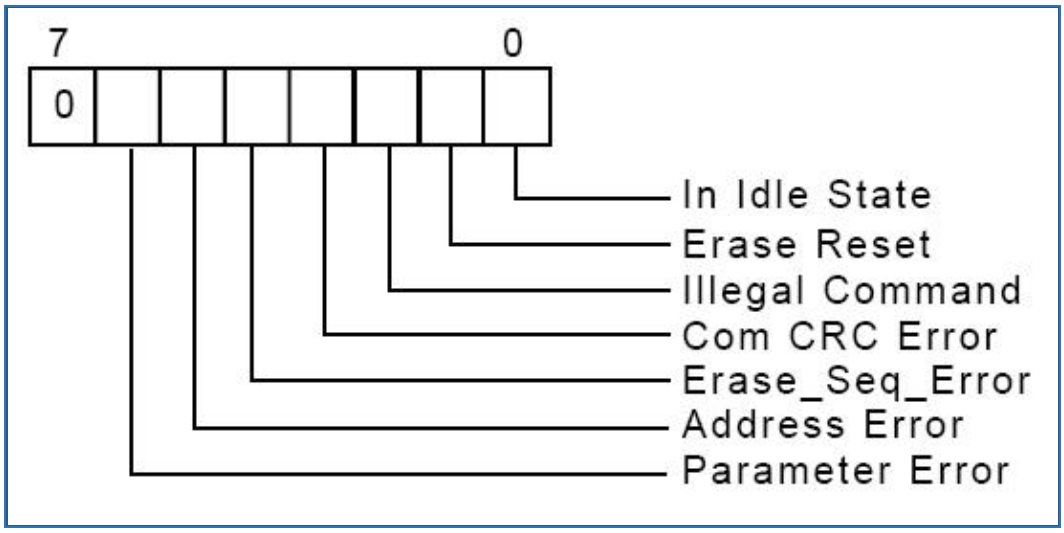


Fig. 10 *Byte de respuesta*

Respuestas según los comandos enviados a la tarjeta SD.

Comando	Respuesta	Descripción
CMD0	0x01	En espera
CMD1	0x00	Activo
CMD17	0xFE + (bytes leídos)	Lee un sector de la tarjeta
CMD24	0x00	Sigue con la escritura
0xFE + bloque a escribir	0xE5	Recepción correcta
Al valor 0xFE se lo denomina Token		

Tabla 6 *Respuesta módulo micro SD*

Valores del Token:

7					0
X	X	X	0	Status	1

Tabla 7 *Valores bite Token*

El status puede ser:

- *010: Datos aceptados.*
- *101: Datos rechazados por error de CRC.*
- *110: Datos rechazados por error de escritura.*

4.1.3 Modos de trasferencia

- ***Modo nativo de 1 bit***
- ***Modo nativo de 4 bit***
- ***Modo SPI***

Los modos nativos requieren de una interfaz host especial, usan pines diferentes para enviar – recibir datos. El modo nativo de 1 bit utiliza un solo pin de datos, el modo nativo de 4 bit, utiliza cuatro pines de datos.

El modo SPI utiliza una interfaz SPI estándar en modo 0 con un ancho de palabra de 8 bit, en aplicaciones de bajo costo con microcontroladores se suele usar el modo SPI.

Fig. 11 *Esquema de conexiones Micro SD*

Los pines 2, 3,4 y 7 son los pines encargados de la comunicación SPI, este es un modo alternativo de comunicación con menos funcionalidad que el modo nativo, pero más simple de implementar.

El set de comandos es reducido, pero permite realizar las funciones básicas de lectura – escritura, suficientes para almacenar datos. En la comunicación SPI la SD funciona como esclavo en modo 0, la frecuencia máxima de clock es de 25MHz, aunque durante el arranque o inicialización la frecuencia de operación debe ser entre 100KHz a 400KHz.

Fig. 12 *Módulo Micro SD comercial*

4.2. Reloj RTC.

El reloj en tiempo real (RTC) es un dispositivo electrónico que permite obtener mediciones de tiempo en las unidades temporales que son empleadas a diario. El termino RTC se creó para poder diferenciar a este tipo de reloj electrónicos de los usados anteriormente, donde solo contabilizaban pulsos de una señal, sin tener relación directa con las unidades temporales.

Los relojes tipos RTC, funcionan con segundos, minutos, horas, días y años, de esta manera se brinda una información más precisa y fiable.

Su construcción es en base a un resonador de cristal integrado con la electrónica necesaria para contabilizar de forma correcta el paso del tiempo. Dicha electrónica tiene en cuenta nuestra forma de medir, como pueden ser los meses con diferentes días o los años bisiestos.

El reloj RTC tiene la ventaja de liberar al microcontrolador de realizar la cuenta de señales y reducir el consumo, además los RTC tienen incorporado una batería, para que, de esta manera poder seguir teniendo energía suficiente para seguir funcionando y brindar información siempre actualizada.

4.2.1 RTC DS3231.

Comercialmente existen dos tipos de RTC, que son el DS1307 y DS3231 ambos fabricados por MAXIM (anteriormente Dallas Semiconductores). El DS3231 tiene una precisión muy superior al DS1307 y se puede considerar el sustituto.

Una de las desventajas por las cuales el DS1307 quedo fuera del mercado es su deficiencia ante los cambios de temperatura, estos cambios afectan la medición del tiempo en los cristales y esto se traduce en un error de desfasaje acumulado, llegando a ser un error considerable, de 1 o 2 minutos al día.

Para solucionarlo el DS3231 incorporó la medición y compensación de la temperatura, garantizando una precisión de 2ppm, lo que significa un desfasaje de 172ms/día, o 1s cada 6 días. Este error se puede seguir disminuyendo mediante consideraciones vía software, llegando a disminuir el desfasaje de 1 – 2 segundos al mes.

La comunicación entre el microcontrolador y el chip DS1307 o DS3231 se lleva a cabo mediante el uso del bus I2C, por lo que es sencillo obtener los datos.

4.2.2 Conexiones y diagrama en bloque

El circuito típico propuesto por el fabricante para el uso del chip es el que se muestra a continuación.

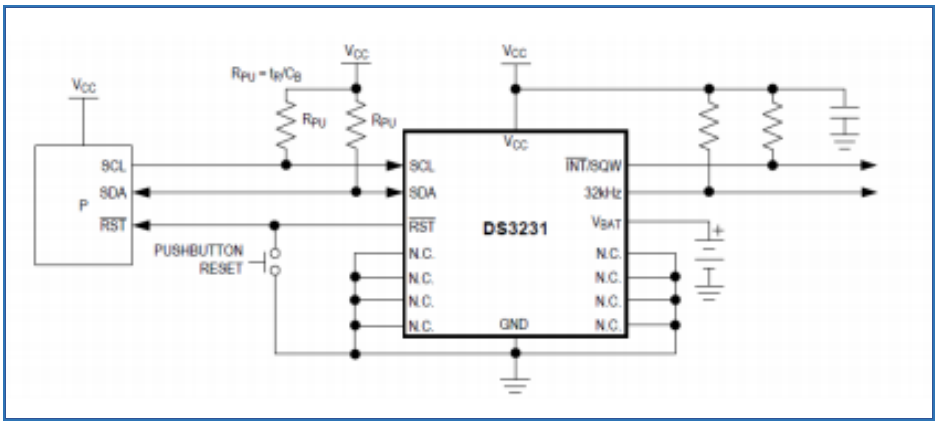


Fig. 13 *Circuito DS3231*

El módulo utilizado es el ZS-042 donde el diagrama en bloques simplificado es el siguiente.

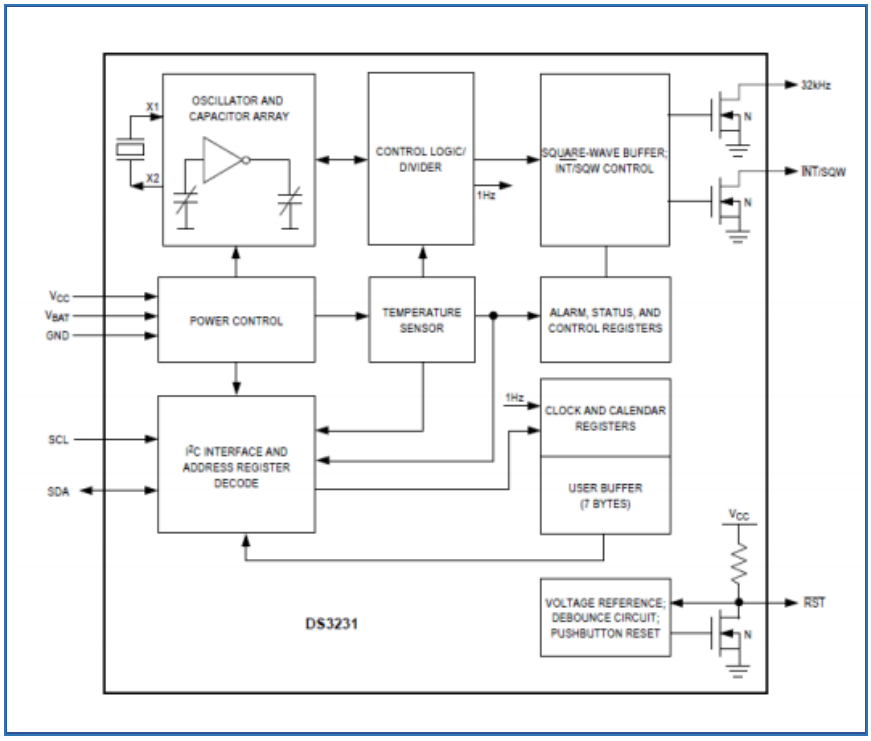


Diagrama. 4 *diagrama en bloques simplificados ZS-042*

Del diagrama en bloques podemos subdividirlo en 4 grupos funcionales: cristal oscilador con compensación de temperatura, control de alimentación, reinicio del dispositivo y reloj en tiempo real.

El primer grupo engloba al sensor de temperatura, oscilador y control lógico.

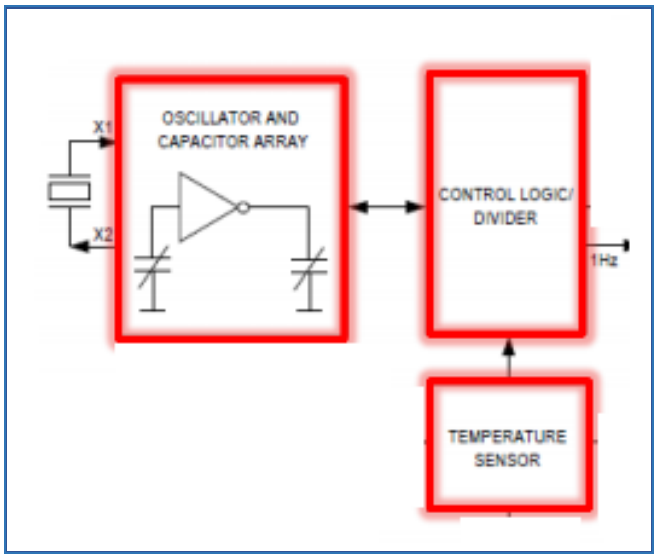


Fig. 14 *Grupo 1: Oscilador con compensación de temperatura.*

Con el sensor de temperatura se recopilan datos y estos se comparan con datos tabulados para determinar la capacitancia requerida y se añade la corrección en el registro. Todo esto ocurre solo cuando se detecta un cambio en la temperatura o cuando se completa una conversión de temperatura. Una conversión de temperatura se lleva a cabo cuando se alimenta el dispositivo o una vez cada un minuto y cuatro segundos.

El segundo grupo está constituido por una tensión de referencia compensada en función de la temperatura y un circuito comparador que controla el valor de V_{cc}.

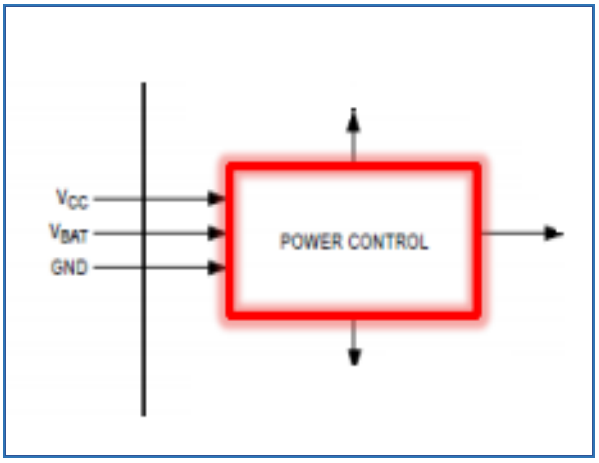


Fig. 15 *Grupo 2: Control de alimentación*

El control de alimentación se basa en un comparador, donde se compara V_{cc} y V_{PF} (Power – Faill Voltage ≈ 2.575V), si V_{cc} es mayor que V_{PF} el circuito se alimenta mediante V_{cc}. Lo mismo sucede si V_{cc} es menor que V_{PF}, pero mayor que V_{BAT} la siguiente tabla muestra, según la comparación, la alimentación del circuito.

Comparación	Suministro de tensión
$V_{cc}<V_{PF} - V_{cc}<V_{bat}$	V_{bat}
$V_{cc}<V_{PF} - V_{cc}>V_{bat}$	V_{cc}
$V_{cc}>V_{PF} - V_{cc}<V_{bat}$	V_{cc}
$V_{cc}>V_{PF} - V_{cc}>V_{bat}$	V_{cc}

Tabla 8 *Tensión de comparación DS3231*

Para que la betería tenga más vida útil, el DS3231 no empezara a funcionar si no se supera V_{PF} o se envíe una dirección válida por el bus I2C. Con cualquiera de las dos condiciones, el oscilador empieza a funcionar, proporcionando por defecto la fecha y hora: 01/01/2000 00:00:00.

En el tercer grupo tenemos el bloque de reinicio.

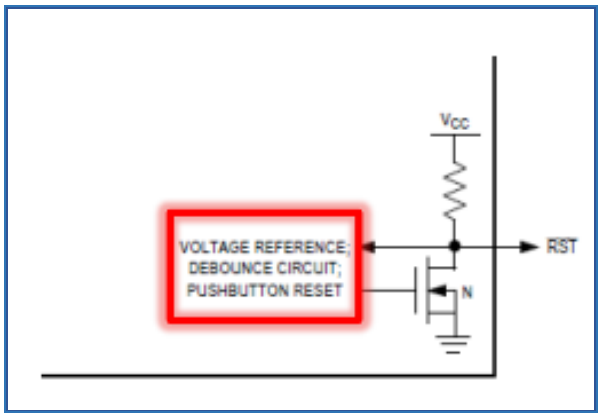


Fig. 16 *Grupo 3: Reinicio de dispositivo.*

El DS3231 incorpora un RST para llevar a cabo un reseteo del chip, el RST se activa con nivel lógico bajo (0V). El RTC está siempre esperando un flanco de bajada del RST, si se detecta una transición de este tipo, el DS3231 activa el transistor colocando el pin en un estado bajo. Después que el temporizador interno haya terminado (PB_{DB}=250ms), el DS3231 sigue verificando el RST, si este continua en un nivel bajo, se esperará un flanco de subida.

También el RST es utilizado para indicar fallos en $V_{cc,}$ cuando $V_{cc} < V_{PF}$ se fuerza el pin RST a un nivel bajo. Cuando la tensión V_{cc} regresa a su nivel el pin RST se mantendrá a 0V durante un tiempo t_{rec} = 250ms para permitir estabilizar la señal de alimentación.

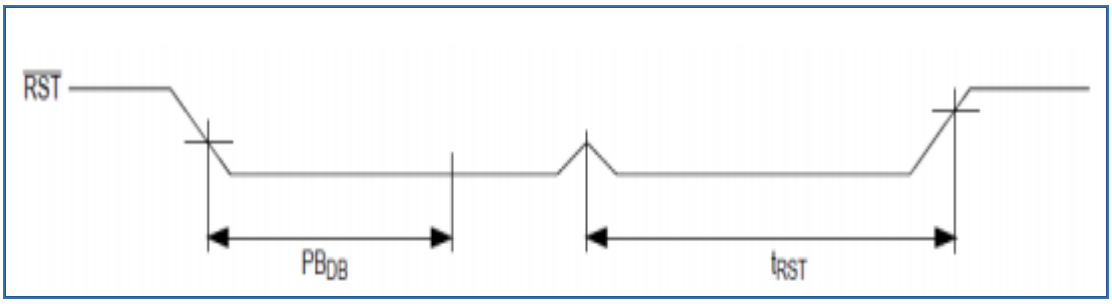


Fig. 17 *Fallo de tensión Vcc.*

El cuarto grupo es el reloj en tiempo real.

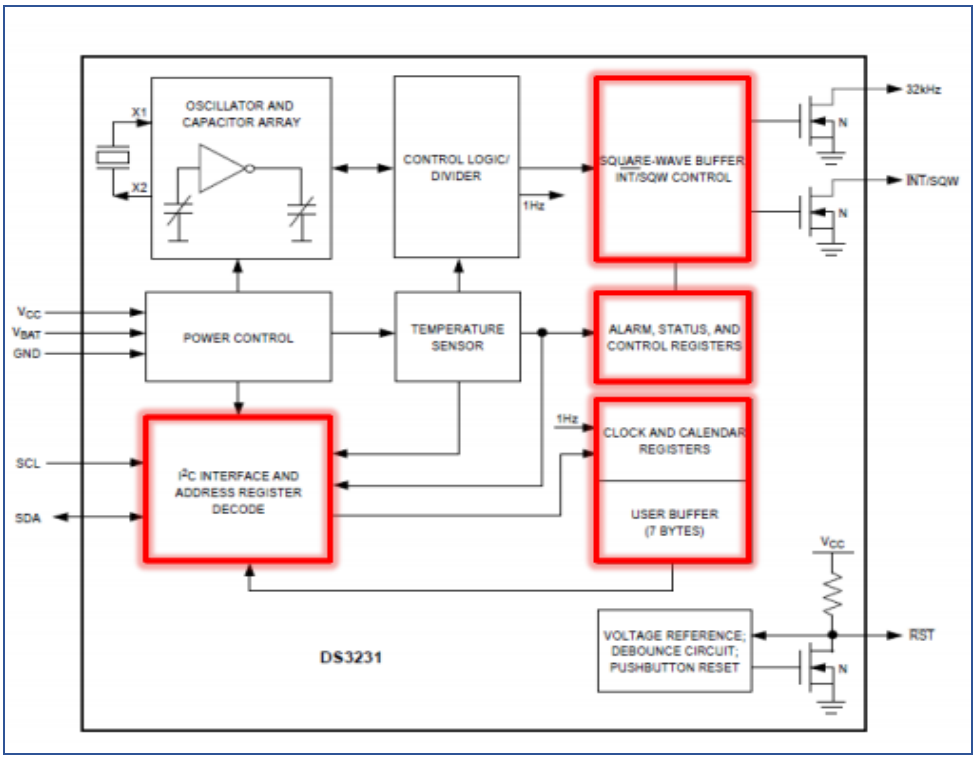


Fig. 18 *Grupo 4: Reloj de tiempo real.*

Este grupo nos proporciona información actualizada sobre los segundos, minutos, horas, días y años. Los días se ajustan automáticamente a los meses de 30 y 31 días, así también como febrero de 28 días y los años bisiestos. Tiene un formato de 12 o 24 horas con un bit indicador de AM/PM.

4.2.3 Comunicación I2C.

La comunicación entre el microcontrolador y el DS3231 se logra a través del bus I2C (Inter – Integrated Circuit). Este protocolo de comunicación reduce el número de cables a dos, uno SDA encargado de la trasferencia de datos y el otro SCL que es el reloj del sistema. El bus trabaja con lógica positiva, donde un nivel alto es $0.7V_{cc}$ y un nivel bajo es $0.3V_{cc}$.

El bus I2C está diseñado para trabajar como maestro – esclavo, donde el maestro inicia la comunicación y el esclavo reacciona a las peticiones. El maestro es el encargado de controlar al bus I2C, es el encargado de generar el reloj serial (SCL), controla el acceso de buses y es el encargado de generar las condiciones de START y STOP.

El DS3231 funciona como esclavo en el bus I2C, una configuración típica se muestra a continuación.

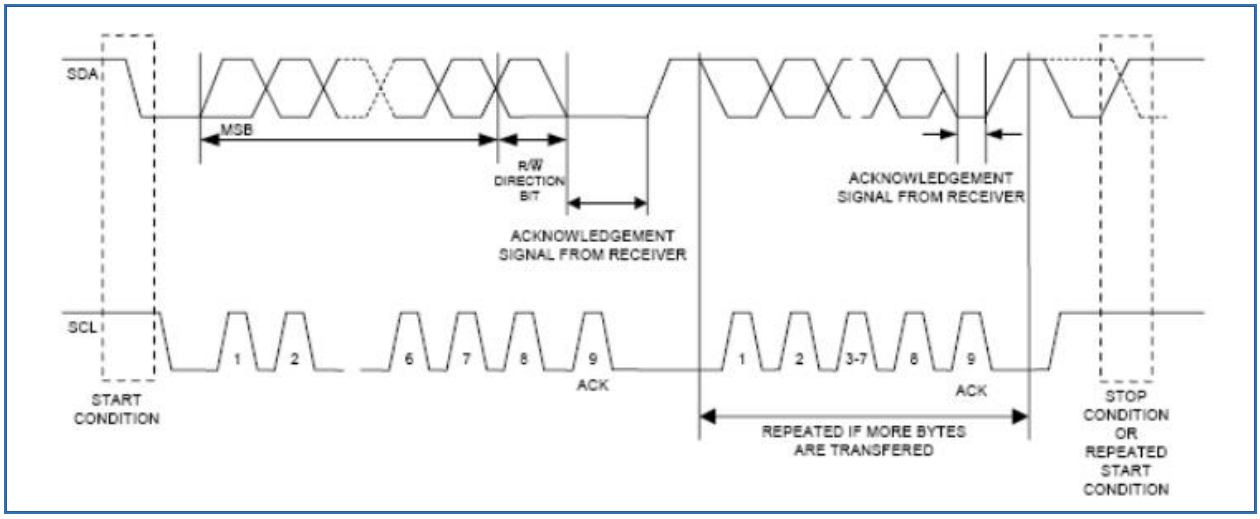


Fig. 19 Esquema de t*rasferencia de datos en el bus I2C.*

Durante la trasferencia de datos, la línea de datos debe permanecer estable cuando la línea de reloj es alta. Los cambios de línea de datos, mientras la línea de reloj es alta, se interpretarán como señales de control.

4.2.4 Escritura de datos

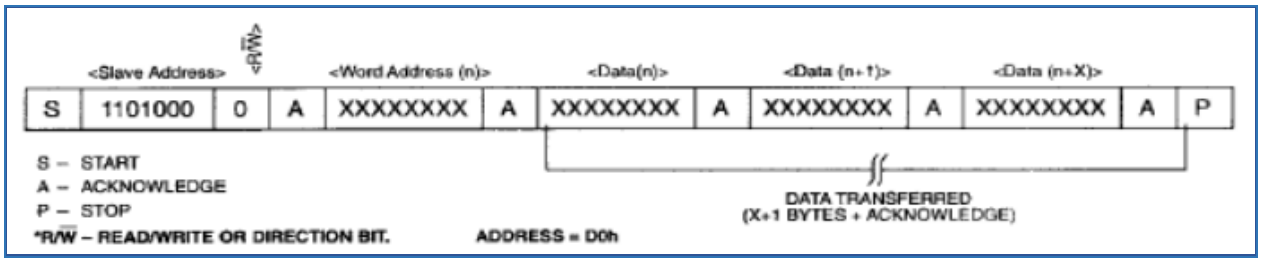


Fig. 20 *Escritura de datos en modo esclavo.*

Después de cada byte recibido, es trasmitido un bit de reconocimiento. Las condiciones START y STOP son reconocidas como el comienzo y el final de la trasmisión serie.

El byte de dirección es el primer byte recibido después de la condición START que es generada por el maestro. El byte de dirección esclavo contiene 7 bit seguida por el bit R/W, que para la escritura es un 0 y para la lectura un 1.

Después de decodificar el byte de direcciones, el esclavo enviará un bit de reconocido por la línea SDA. Después que el esclavo reconoce la dirección más el bit de escritura, el maestro trasmite una palabra de dirección al DS3231, esto establecerá el puntero del registro. Con el esclavo reconociendo cada byte recibido, el maestro termina la trasferencia de datos con la condición de STOP.

4.2.5 Lectura de datos

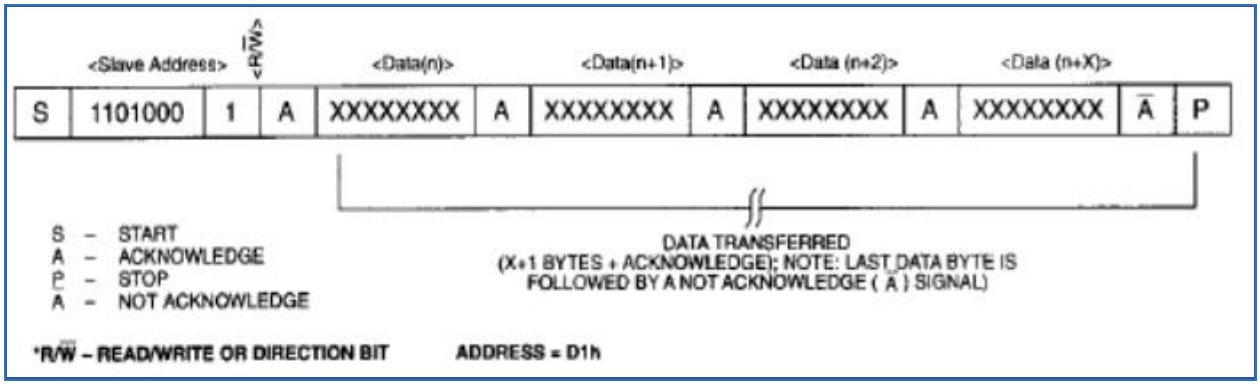


Fig. 21 *Lectura de datos en modo esclavo.*

El primer byte se recibe y maneja como en el modo escritura. Sin embargo, en este modo, el bit de dirección indicará que la dirección de trasferencia es inversa. Datos en serie se trasmiten del DS3231 al maestro.

Las condiciones de STAR y STOP indican lo mismo que en modo escritura. El primer byte es el de dirección que consta de 7 bit de dirección más un bit de R/W.

Después de recibir y decodificar el byte de dirección el esclavo responde con un bit de reconocido en la línea SDA. El DS3231 comienza entonces a trasmitir los datos que tiene en la dirección enviada anteriormente por el maestro. El esclavo pone fin a la lectura tras recibir un "no reconocido" por el maestro.

4.2.6 Características ZS-042.

- *Tensión de funcionamiento: 3,3v ~ 5v.*
- *Chip de reloj: DS3231.*
- *Precisión del reloj: 2ppm entre 0°C – 40°C.*
- *Salida de onda cuadrada programable.*
- *Sensor de temperatura incluido con una precisión ±3°C.*
- *Interfaz con el bus I2C.*
- *Batería integrada CR2032.*
- *Corriente de operación con 5v: 300µA.*
- *Corriente de stand-by con 5v: 170µA.*

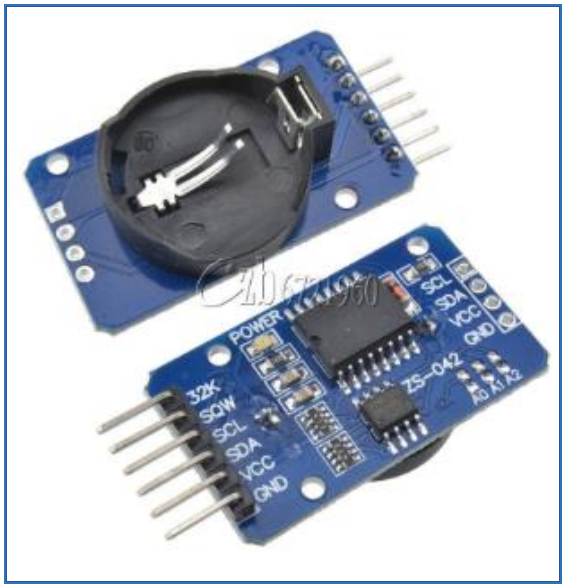


Fig. 22 *Módulo ZS-042 comercial.*

5. Sistema de control

El microcontrolador es el encargado de controlar y coordinar todas las funciones del sistema. Para la elección del microcontrolador PIC se tuvieron en cuenta las características con las que debía contar a la hora de controlar todas las funciones del sistema, es decir, que disponga de la cantidad de pines necesarios para poder conectar y controlar todos los módulos, contar con bus SPI e I2C necesarios para establecer la comunicación maestro – esclavo con los módulos usados y memoria suficiente.

Dicho todo esto se optó por el microcontrolador PIC 18F4550 de la empresa Microchip Technology Inc de 40 Pin PDIP.

5.1 Diagrama esquemático

En la fig.27 se muestra el diagrama y conexionado de los módulos junto al microcontrolador PIC18f4550.

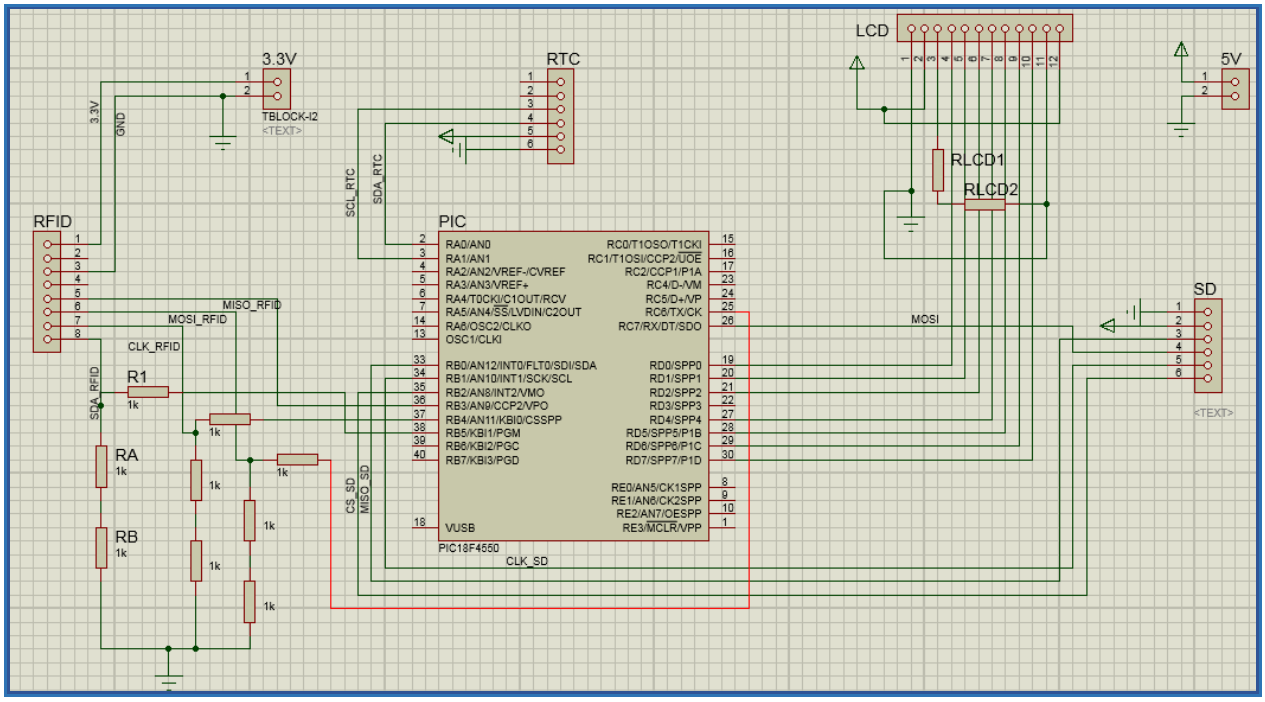


Fig. 23 *Representación del conexionado de los módulos*

Se van a utilizar dos tensiones diferentes para la alimentación, una fuente de 5v la cual va a alimentar al microcontrolador, modulo micro SD, LCD 16x2 y reloj RTC. La segunda tensión es de 3.3v la cual va a alimentar al módulo RFID el cual no soporta 5v.

Se utilizarán tres divisores resistivos para bajar la tensión de 5v (entregadas por el microcontrolador) a 3.3v (necesarias para no dañar al módulo RFID).

Cálculo del divisor de tensión:

$$V_{out} = V_1\left(\frac{R_2}{R_1 + R_2}\right)$$

$$R_2 = R_A + R_B$$

$$\frac{V_{out}}{V_1} = \left(\frac{R_2}{R_1 + R_2}\right)$$

$$\frac{V_{out}}{V_1}(R_1 + R_2) = R_2$$

$$\frac{V_{out}}{V_1}(R_1) + \frac{V_{out}}{V_1}(R_2) = R_2$$

$$\frac{V_{out}}{V_1}(R_1) = R_2 - \frac{V_{out}}{V_1}(R_2)$$

$$\frac{V_{out}}{V_1}(R_1) = R_2\left(1 - \frac{V_{out}}{V_1}\right)$$

$$R_2 = \frac{\frac{V_{out}}{V_1}(R_1)}{1 - \frac{V_{out}}{V_1}}$$

- $V_{out} = 3.3v \; y \; V_1 = 5v$
- *Suponemos una resistencia de* $R_1 = 1k\Omega$

$$R_2 = \frac{\frac{3.3v}{5v}(1k\Omega)}{1 - \frac{3.3v}{5v}}$$

$$R_2 = 1941{,}176\Omega \approx 2k\Omega$$

$$R_2 = 2k\Omega = R_A + R_B \rightarrow \boldsymbol{R_A = 1k\Omega \; y \; R_B = 1k\Omega}$$

El cálculo de la resistencia se aplica a los tres divisores ya que todos tienen las mismas características.

En la siguiente imagen se presenta una simulación para garantizar que la tensión entregada sea de 3.3v. El software que se utilizó para hacer la simulación es "Quite Universal Circuit Simulator".

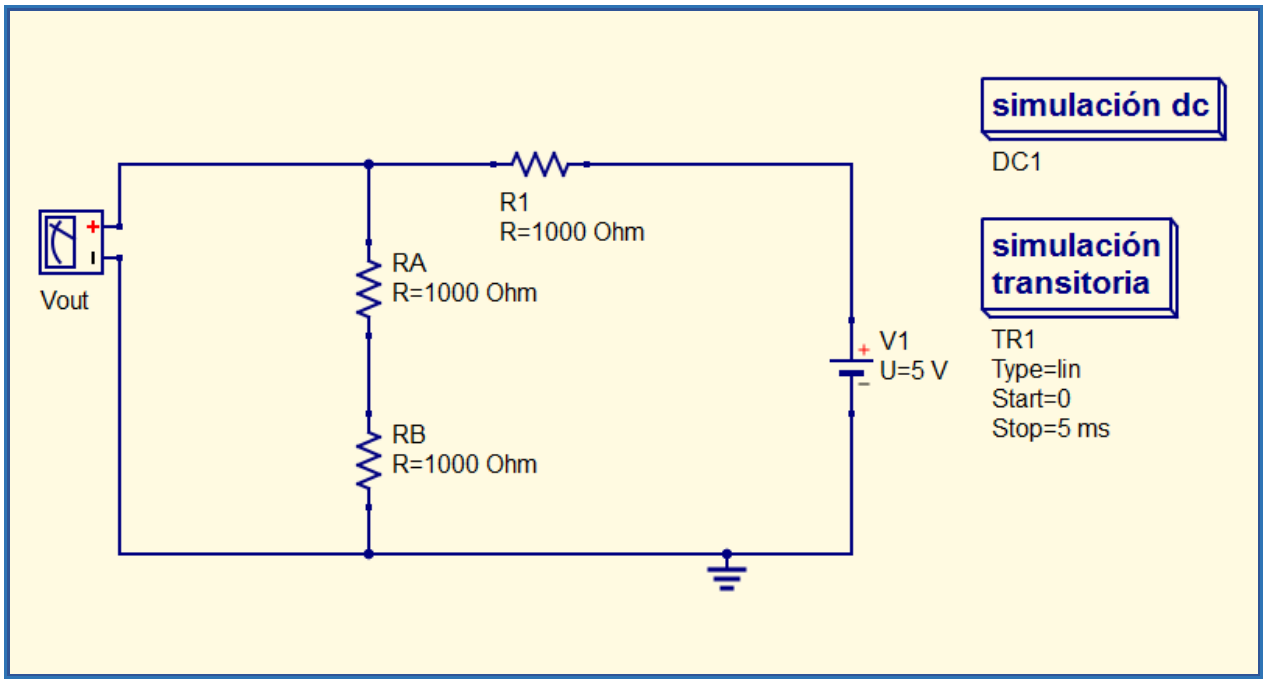


Fig.24 *Simulación circuito.*

number	Vout.V
1	3.33

Fig. 25 *Valores de salida Vout*

5.2 Bus SPI.

En la sección 2.1.2 y 3.2 se habló sobre el bus SPI usado en modo esclavo, ahora se desarrollará el bus usado como maestro, es decir, la configuración respecto al microcontrolador.

El protocolo de comunicación SPI consta de tres líneas MOSI o SDI (salida de datos del maestro y entrada de datos al esclavo), MISO o SDO (salida de datos del esclavo y entrada de datos al maestro) y la señal SCK (señal de reloj de sincronismo). Los dispositivos esclavos conectados no utilizan direcciones, sino que se utiliza una línea de control para la habilitación de cada esclavo la señal es SS o Select (selección del esclavo para la comunicación).

En la siguiente figura se muestra el diagrama en bloque del bus SPI.

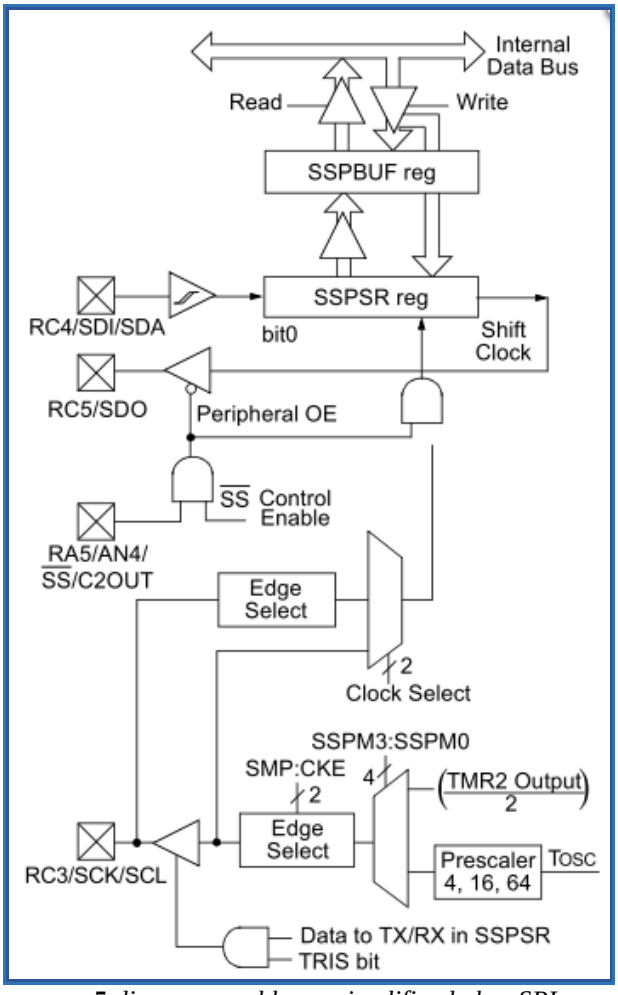


Diagrama. 5 *diagrama en bloque simplificado bus SPI*

Dado que los módulos vistos anteriormente trabajan como esclavos, el microcontrolador trabajara como maestro siendo este el encargado de proporcionar la señal de reloj necesaria para comandar las acciones del sistema.

En este tipo de trasmisión sincrónica se permite enviar y recibir datos de 8 bit a través del registro de desplazamiento SSPSR y un registro intermedio, SSPBUF.

La recepción o trasmisión de un dato se realiza ni bien se completa el registro SSPSR. En un proceso de recepción por el pin SDI, el registro de desplazamiento SSPSR trasfiere su información al registro SSPBUF y queda liberado para realizar la siguiente acción. Los datos presentes en SSPBUF se mantienen hasta que sean leídos, pudiendo aceptar nuevos datos el registro SSPSR, de esta manera el bus SPI trabaja con doble registro. Los datos que quieran ser enviados a través de la línea SDO deben ser escritos en el registro SSPSR, desde el registro SSPBUF.

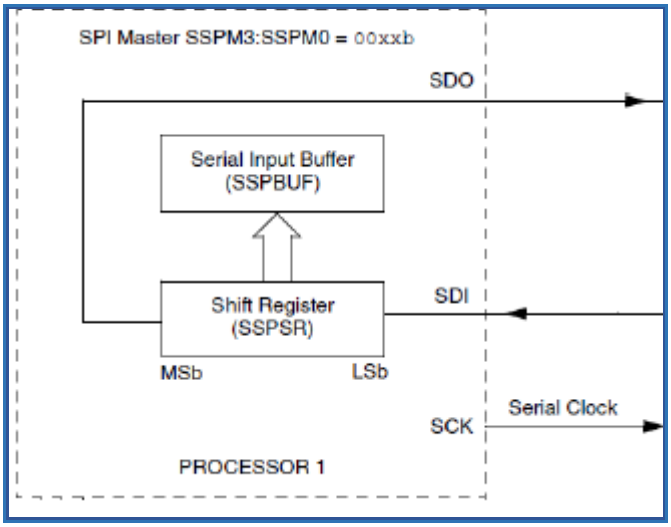


Fig. 36 *SPI configurado como maestro.*

El dispositivo maestro es el encargado de comandar los tiempos de operación, debido a que es él el que controla el reloj del sistema a través del pin SCK. Dicha señal está presente solo en el momento de trasferencia de datos entre el dispositivo maestro y esclavo.

5.2.1 Modo de funcionamiento bus SPI

Por defecto el maestro mantiene su línea SS en alto, cuando el maestro quiere iniciar la comunicación pone en un nivel lógico bajo la línea correspondiente al esclavo con el cual quiere iniciar la comunicación.

En cada pulso de la señal del reloj, normalmente en un flaco de subida, se da la trasferencia de datos entre maestro y esclavo. La trama (datos) enviados y recibidos no sigue ninguna regla, es decir podemos enviar cualquier secuencia arbitraria de bits.

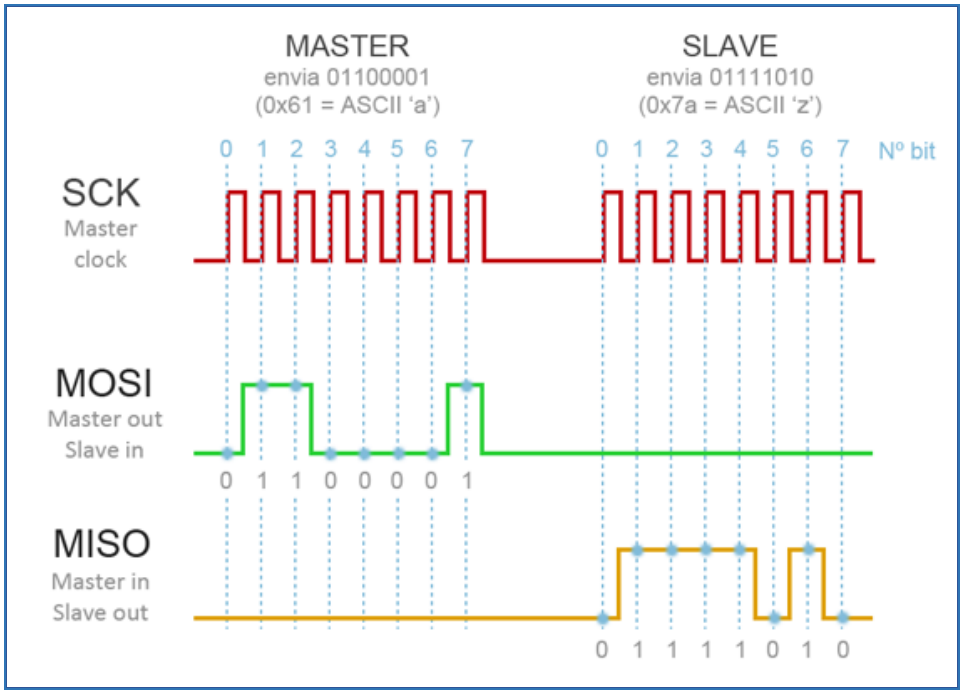


Fig. 27 *Ejemplo de comunicación SPI*

5.3 Bus I2C

El bus I2C (Inter-Integrated Circuit) requiere únicamente de dos cables para su funcionamiento, uno para la señal de reloj (CLK) y otro para el envío y recibimiento de datos (SDA). Esto significa una gran ventaja, ya que con solo dos hilos se establece la comunicación.

En este tipo de comunicación cada dispositivo dispone de una dirección, que se empela para poder acceder al dispositivo en cuestión de forma individual, esta dirección se puede emplear por software o hardware, donde en este último se utilizan jumpers para modificar los últimos 3 bit.

El bus I2C tiene su arquitectura del tipo maestro-esclavo. Esto es que el maestro es el encargado de indicar el inicio y fin de la comunicación, el esclavo solo de limita a enviar o recibir datos.

El maestro es el encargado de generar una señal de reloj (CLK) que mantiene sincronizado a todos los dispositivos conectados al bus, por eso decimos que el bus I2C es del tipo sincrónico.

5.3.1 Funcionamiento bus I2C

Para poder realizar la comunicación del maestro con el esclavo con un solo cable de datos (SDA), el bus I2C utiliza una trama amplia.

La comunicación consta de:

- *7 bits que serán la dirección del dispositivo esclavo con el cual queremos comunicarnos.*
- *Un bit indica si queremos enviar o recibir datos.*
- *Un bit de validación.*
- *Uno o más bytes de datos.*

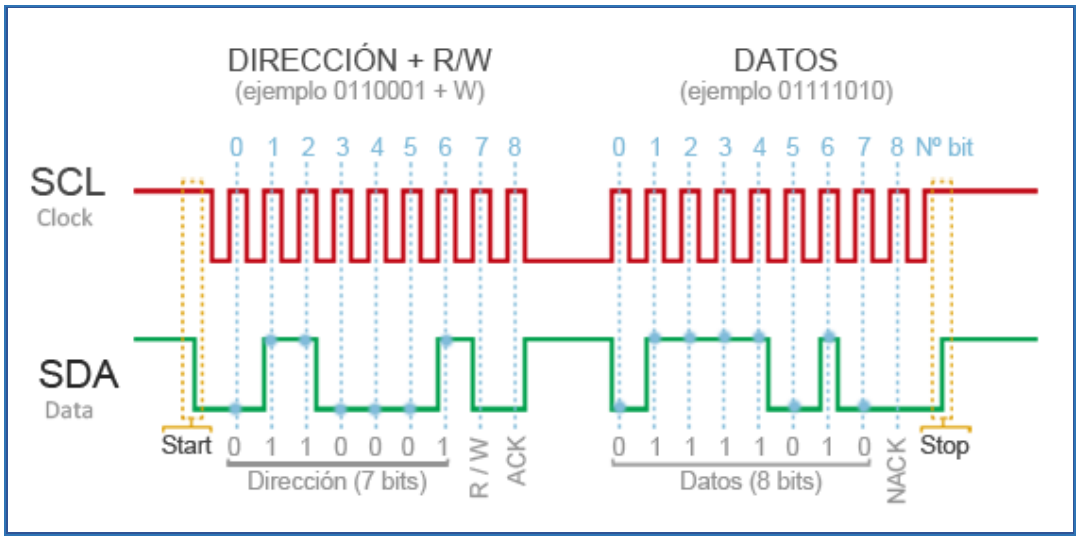


Fig. 28 *Ejemplo comunicación I2C*

Con 7 bit de dirección es posible conectar y comunicar 112 dispositivos a un mismo maestro.

La condición de inicio de la comunicación se da cuando el bus está disponible y el estado lógico de la línea SDA pasa de un nivel alto a un nivel bajo estando el CLK en un nivel lógico alto.

Para finalizar la trasferencia de datos hay un flanco de subida en la línea SDA mientras hay un nivel alto en el estado del CLK. Tanto la señal de inicio como parada solo la puede generar el MAESTRO.

La confirmación de recepción (ACK) se da cuando tenemos la señal CLK en alto y SDA en bajo, esta señal la puede generar tanto el MESTRO como el ESCLAVO.

5.4. Programación

En este apartado se hacer referencia a la programación que lleva el microcontrolador PIC, se realizará un diagrama de flujo simplificado de las funciones llevadas a cabo.

5.4.1 Diagrama de flujo del menú principal

La siguiente figura muestra el diagrama de flujo del programa principal, donde como primera acción se lleva a cabo una configuración de los parámetros usados por el microcontrolador, como puede ser el uso de las palabras de configuración, declaración de librerías, configuración del oscilador interno, uso de puertos SPI e I2C y uso de variables globales.

La programación está dividida en subconjuntos llamados funciones, de esta manera se logra una mejor organización de las tares y trabajos realizados por el PIC y en caso de tener algún error o falla es más fácil y sencillo encontrar y resolver el problema.

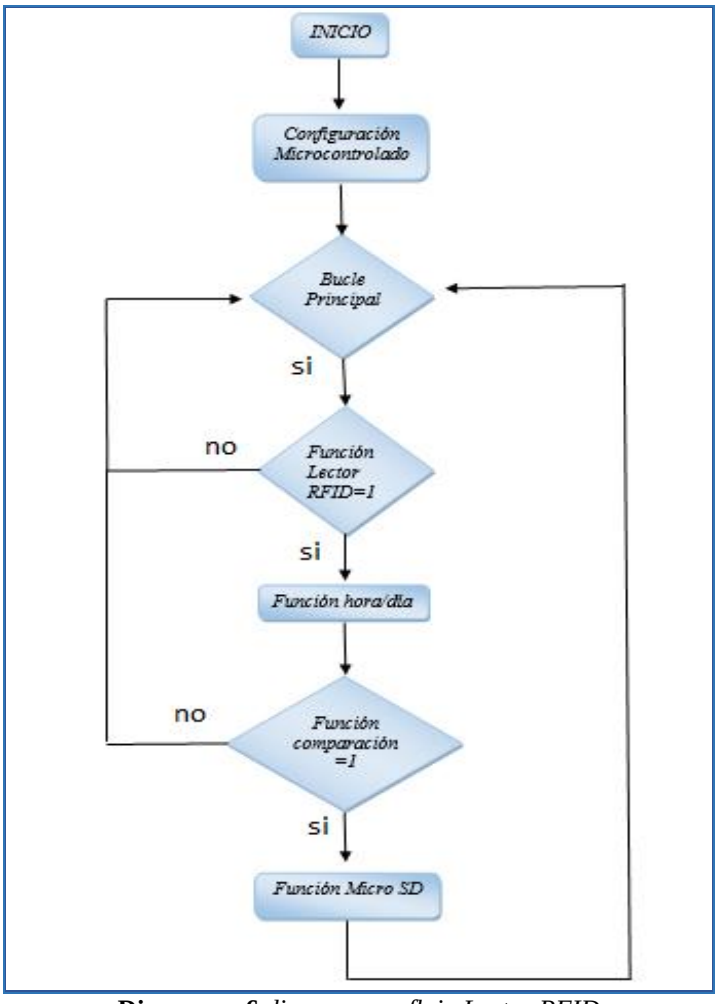


Diagrama. 6 *diagrama en flujo Lector RFID*

Este es el diagrama de flujo del Lector RFID. Se hará una breve descripción de cada función.

La primera función después del bucle principal es la Funcion Lector RFID en esta función se realiza la lectura del legajo y autentificación entre el lector MRFC555 y el tag electromagnético, toda esta función esta echa en base a una librearía antes cargada en la configuración del microcontrolador. Si la lectura del tag se realiza con éxito, en ese momento se llama a la Función hora/día la cual es la encargada de extraer del reloj RTC la fecha y hora exacta que se realizó la lectura.

Una vez que se tienen los datos de la tarjeta junto con los datos de la fecha y hora se procede a llamar a la Función Comprobación esta se encarga de realizar la comparación de los distintos niveles de privilegio, es decir, comprobar si la tarjeta que fue leída puede ingresar al área en cuestión.

Si el tag leído tiene el nivel de privilegio requerido, el último paso es de generar un registro y para ello se llama a la Función Micro SD donde ella se encarga de crear un archivo Excel donde quedara registrado todos los datos antes manipulados (hora, fecha, legajo).

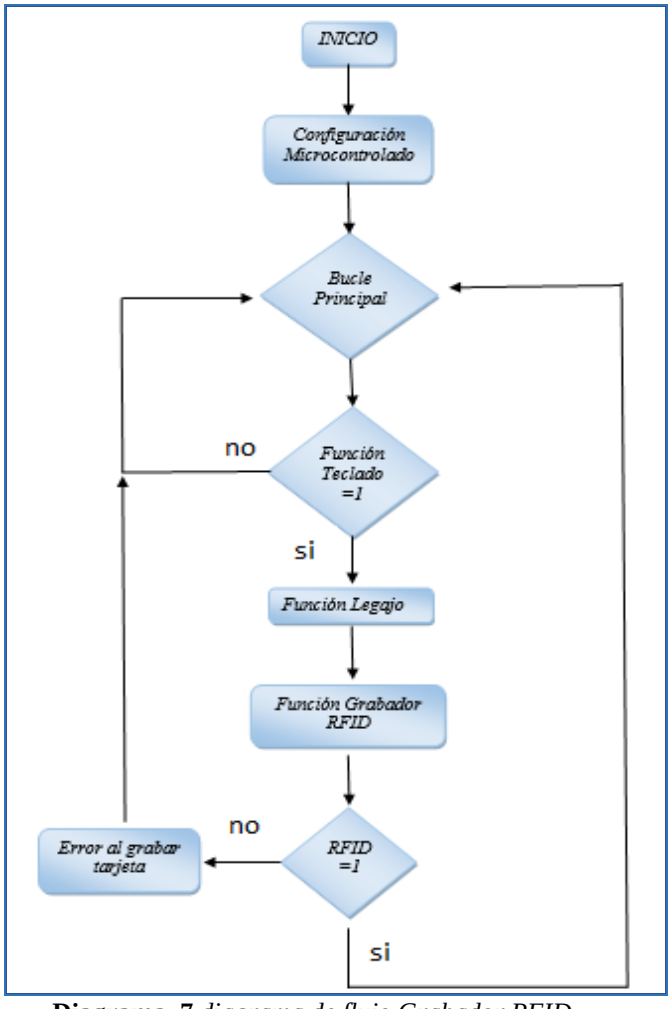


Diagrama. 7 *diagrama de flujo Grabador RFID*

El diagrama del Grabador RFID es similar al Lector, con la diferencia que en este caso se usara un teclado matricial 4x4, y la librería de dicho teclado se incluye en la parte de la Configuración del Microcontrolador.

Como primera medida en el grabador se llama a la Función Teclado donde en ella se digitará el nivel de privilegio de la tarjeta a grabar, disponemos de cuatro niveles de privilegio A, B, C y D. Después se llama a la Función Legajo donde nos pedirá ingresar el legajo correspondiente a la persona que se le entregará la tarjeta.

Después de ingresar todos los datos, el siguiente paso es llamar a la Función Grabador RFID donde ella es la que se encarga de grabar todos los datos en la tarjeta electromagnética. Una vez grabada la tarjeta se llevará a cabo una verificación de la tarjeta grabada, donde si hubo algún tipo de error en la grabación nos saldrá un menaje en la pantalla LCD y volveremos al bucle principal.

6. Alimentación

Para la alimentación de ambos módulos (lectura – escritura) se utilizarán dos tensiones diferentes una de 5v y la otra es de 3.3v.

Es necesario esta disposición ya que el RC522 funciona con una tensión 3.3v, y el resto de los componentes (PIC, micro SD, LCD, etc.) funcionan con 5v. De esta manera si todos los componentes que forman el circuito electrónico serian alimentados con la misma tensión, ya sea 3,3v o 5v el circuito no funcionaria.

Para la implementación de esta placa se utilizaron dos circuitos integrados reguladores de tensión. Uno es el LM7805 que es un regulador de 5v, dicha disposición y conexionado se puede ver en la siguiente figura.

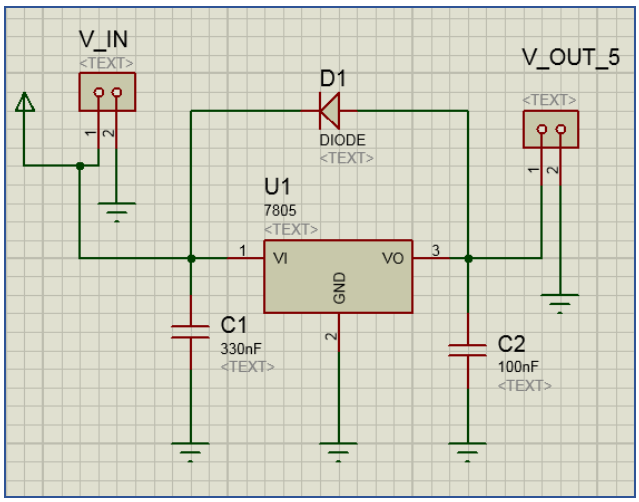


Fig.29 *LM7805*

El integrado utilizado para obtener una tensión 3.3v es el LD1117/A-3.3.

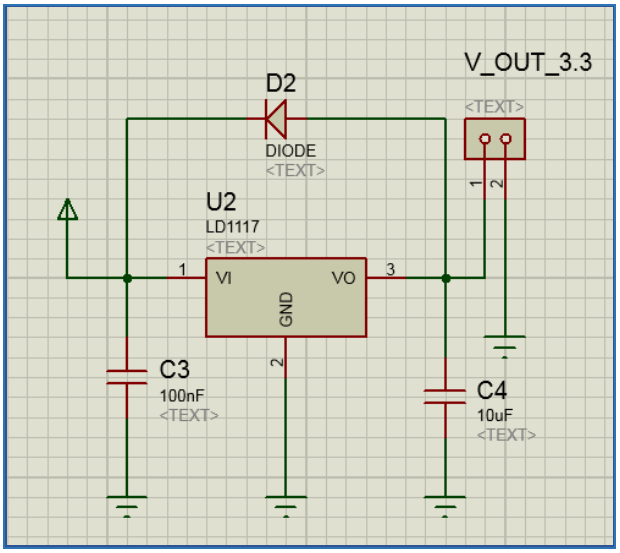


Fig. 30 *LD1117*

Ambos CI son lo suficientemente confiables y brindan una tensión de salida estable.

En el desarrollo final del PCB (placa de circuito impreso) la alimentación va a estar separada del circuito de control, ya que de esta manera si tuviéramos algún tipo de inconveniente con la alimentación esto se podría reemplazar y que el circuito de control no fuera afectado.

7. Diseño e implementación de PCB.

Para el desarrollo final del circuito se utilizó el software Proteus 8. A continuación, en las figuras 38 y 39, se presenta las imágenes tomadas del software.

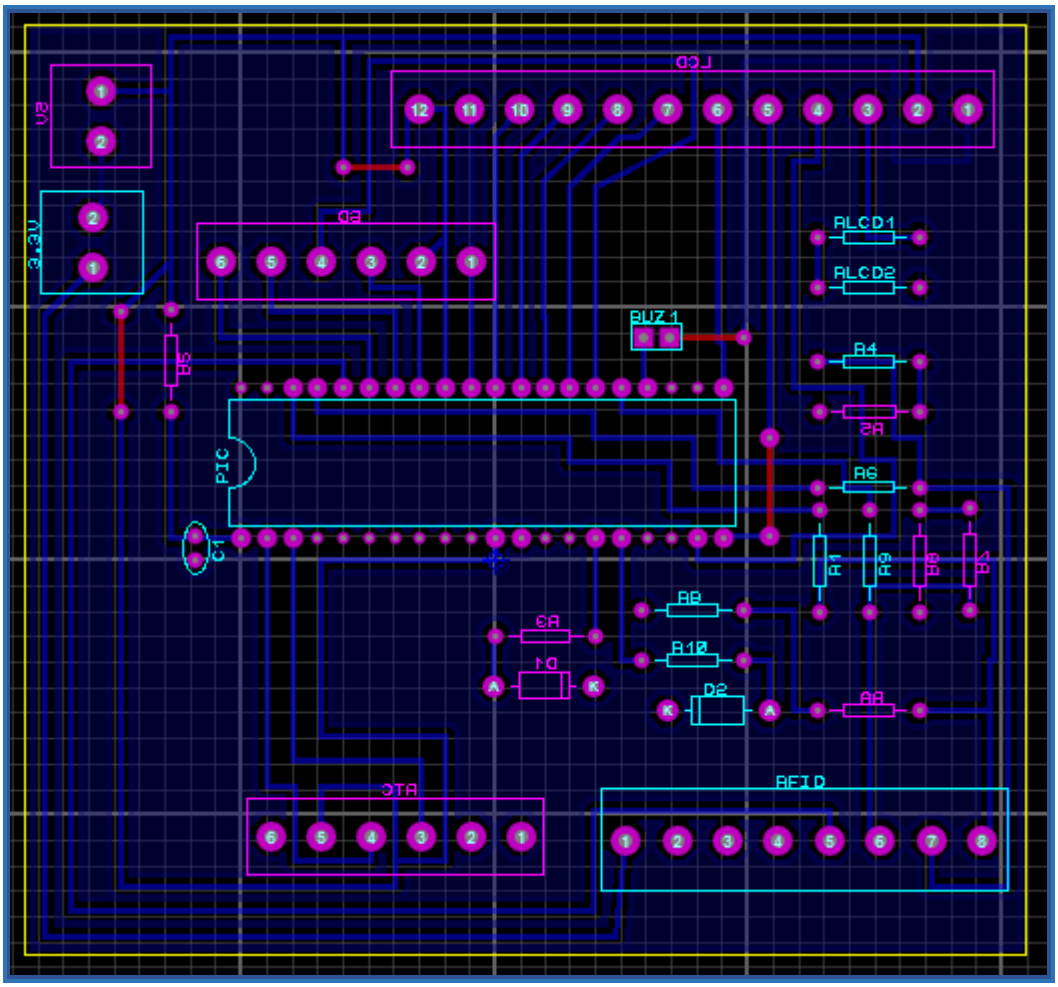


Fig. 31 *Esquema en ARES Lector*

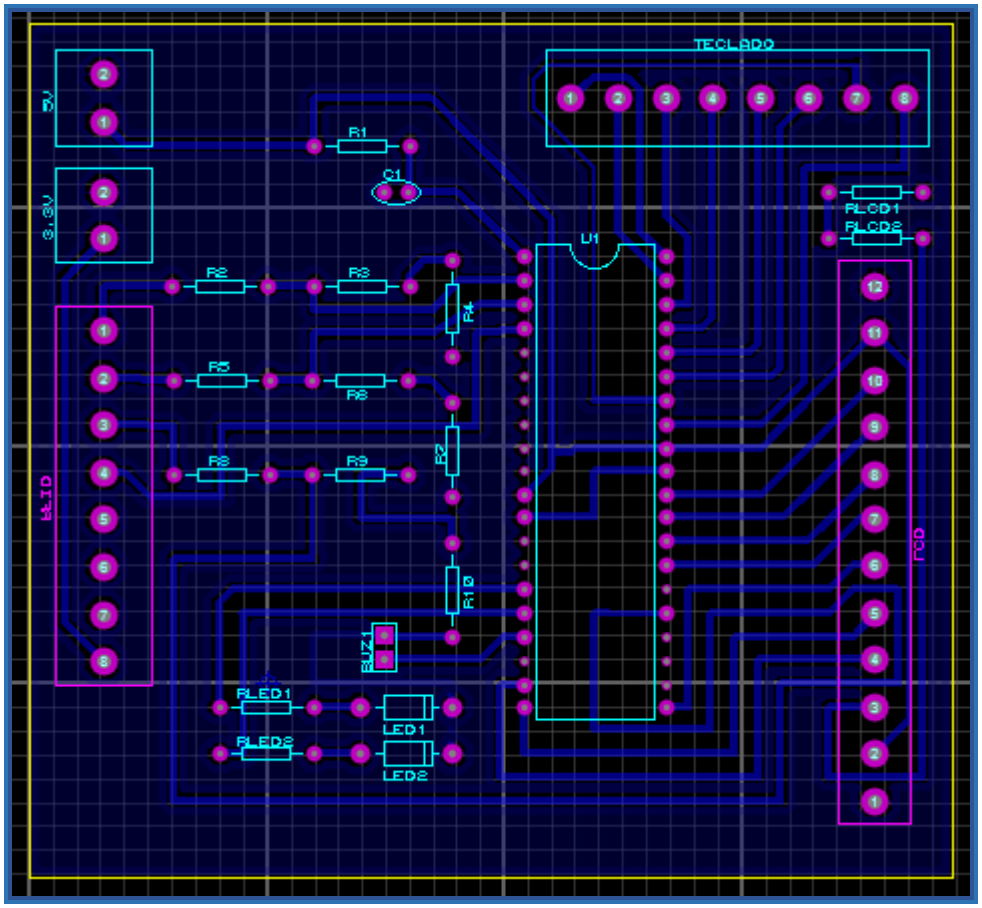


Fig. 32 *Esquema en ARES Grabador*

En la figura 40 se muestra el diagrama en ARES del circuito de alimentación.

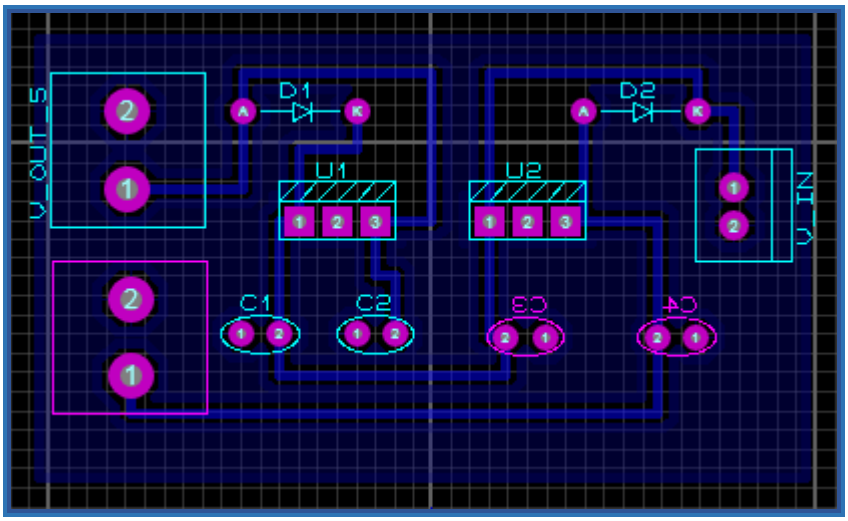


Fig. 33 *Esquema en ARES Alimentación 3,3v – 5v*

A continuación, se muestra el proyecto ya terminado.

Fig.34 *Grabador RFID.*

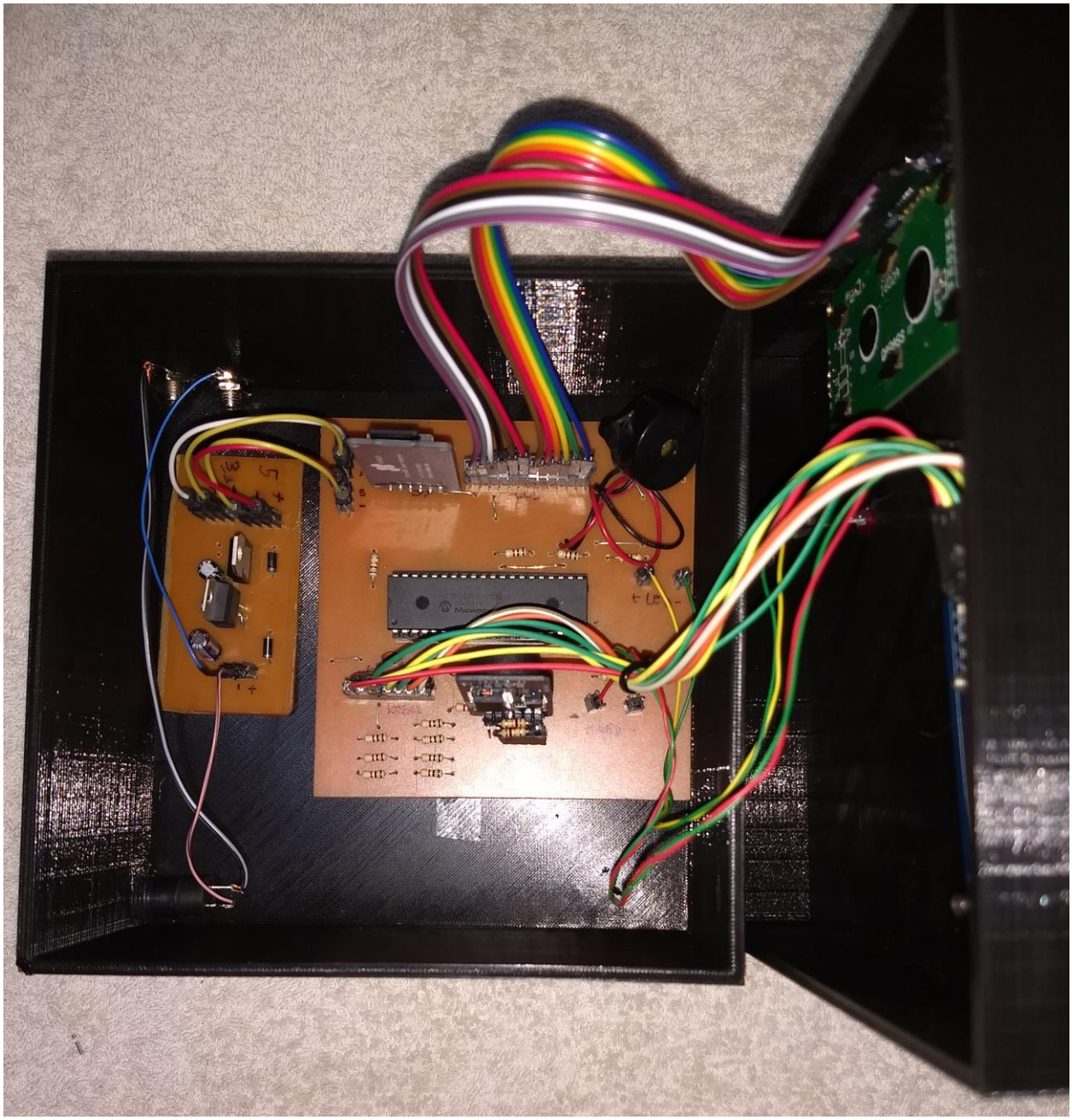


Fig.35 *Placa de lógica y placa de alimentación*

Fig.36 *Lector y grabador RFID.*

8. Conclusión.

Al finalizar el trabajo se obtuvo como resultado un sistema de control de acceso de fácil utilización tanto para quien se encarga de cargar la información en las tarjetas como para quienes utilizan el sistema para acceder a las distintas habitaciones. Cumple correctamente con la concesión o restricción de acceso en función de los niveles de privilegio de cada usuario

El sistema de adquisición de datos genera un registro fiable de las personas que accedieron a una determinada habitación con su respectiva fecha y hora, el cual se almacena correctamente en la tarjeta microSD.

Para el grabado de tarjetas magnéticas (tags) se creó un dispositivo de fácil implementación para que de esta manera se puedan crear nuevos usuarios de manera práctica y confiable.

Gracias al desarrollo del proyecto pude adquirir y reforzar conocimientos en programación, desarrollo y simulación, además de la implementación de nuevos protocolos de comunicación entre el microcontrolador PIC y los módulos utilizados.

9. Bibliografía.

[1] Compilador C CCS y simulador PROTEUS para microcontroladores PIC – Eduardo García Breijo.

[2] Manual de usuario del compilador PCW de CCS. C compiler for Microchip PICmicro^{R}MCUs – Escrito por ANDRES CANOVAS LOPEZ.

[3] Aprendiendo a programar Microcontroladores PIC en Lenguaje C con CCS. Entrega N°8 – Andrés Bruno Saravia.

[4] Microchip Technology Inc.; PIC18F4550, data sheet.

[5] Microchip Technology Inc.; PIC16F887, data sheet.

[6] NXP Semiconductores; MFRC522, data sheet.

[7] DALLAS Semiconductores; DS3231, data sheet.

[8]https://www.vishnumaiea.in/projects/hardware/interfacing-catalex-micro-sd-card-module última consulta 07/02/2019

[9]http://picfernalia.blogspot.com/2013/04/tarjetas-sd-lecturaescritura-de-datos.html última consulta 11/02/2019

[10] https://hispavila.com/wp-content/uploads/2015/08/ds1307_esp.pdf última consulta 18/02/2019

[11] https://slideplayer.es/slide/1758785/ última consulta 18/02/2019

10. Anexos.

Hoja de datos Microchip PIC18F4550

MICROCHIP PIC18F2455/2550/4455/4550

28/40/44-Pin, High-Performance, Enhanced Flash, USB Microcontrollers with nanoWatt Technology

Universal Serial Bus Features:

- USB V2.0 Compliant
- Low Speed (1.5 Mb/s) and Full Speed (12 Mb/s)
- Supports Control, Interrupt, Isochronous and Bulk Transfers
- Supports up to 32 Endpoints (16 bidirectional)
- 1-Kbyte Dual Access RAM for USB
- On-Chip USB Transceiver with On-Chip Voltage Regulator
- Interface for Off-Chip USB Transceiver
- Streaming Parallel Port (SPP) for USB streaming transfers (40/44-pin devices only)

Power-Managed Modes:

- Run: CPU on, peripherals on
- Idle: CPU off, peripherals on
- Sleep: CPU off, peripherals off
- Idle mode currents down to 5.8 µA typical
- Sleep mode currents down to 0.1 µA typical
- Timer1 Oscillator: 1.1 µA typical, 32 kHz, 2V
- Watchdog Timer: 2.1 µA typical
- Two-Speed Oscillator Start-up

Flexible Oscillator Structure:

- Four Crystal modes, including High Precision PLL for USB
- Two External Clock modes, up to 48 MHz
- Internal Oscillator Block:
  - 8 user-selectable frequencies, from 31 kHz to 8 MHz
  - User-tunable to compensate for frequency drift
- Secondary Oscillator using Timer1 @ 32 kHz
- Dual Oscillator options allow microcontroller and USB module to run at different clock speeds
- Fail-Safe Clock Monitor:
  - Allows for safe shutdown if any clock stops

Peripheral Highlights:

- High-Current Sink/Source: 25 mA/25 mA
- Three External Interrupts
- Four Timer modules (Timer0 to Timer3)
- Up to 2 Capture/Compare/PWM (CCP) modules:
  - Capture is 16-bit, max. resolution 5.2 ns (TCY/16)
  - Compare is 16-bit, max. resolution 83.3 ns (TCY)
  - PWM output: PWM resolution is 1 to 10-bit
- Enhanced Capture/Compare/PWM (ECCP) module:
  - Multiple output modes
  - Selectable polarity
  - Programmable dead time
  - Auto-shutdown and auto-restart
- Enhanced USART module:
  - LIN bus support
- Master Synchronous Serial Port (MSSP) module supporting 3-wire SPI (all 4 modes) and I²C™ Master and Slave modes
- 10-bit, up to 13-channel Analog-to-Digital Converter module (A/D) with Programmable Acquisition Time
- Dual Analog Comparators with Input Multiplexing

Special Microcontroller Features:

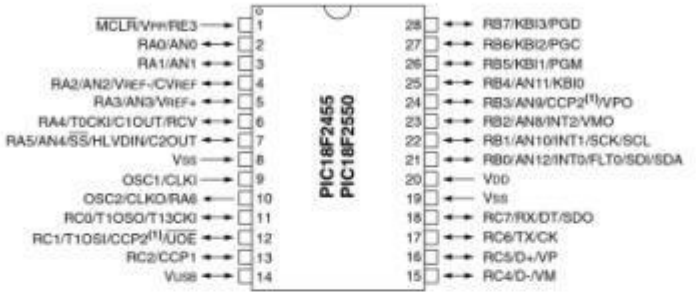
- C Compiler Optimized Architecture with optional Extended Instruction Set
- 100,000 Erase/Write Cycle Enhanced Flash Program Memory typical
- 1,000,000 Erase/Write Cycle Data EEPROM Memory typical
- Flash/Data EEPROM Retention: > 40 years
- Self-Programmable under Software Control
- Priority Levels for Interrupts
- 8 x 8 Single-Cycle Hardware Multiplier
- Extended Watchdog Timer (WDT):
  - Programmable period from 41 ms to 131s
- Programmable Code Protection
- Single-Supply 5V In-Circuit Serial Programming™ (ICSP™) via two pins
- In-Circuit Debug (ICD) via two pins
- Optional dedicated ICD/ICSP port (44-pin devices only)
- Wide Operating Voltage Range (2.0V to 5.5V)

Device	Program Memory		Data Memory		I/O	10-Bit A/D (ch)	CCP/ECCP (PWM)	SPP	MSSP		EAUSART	Comparators	Timers 8/16-Bit
	Flash (bytes)	# Single-Word Instructions	SRAM (bytes)	EEPROM (bytes)					SPI	Master I²C™			
PIC18F2455	24K	12288	2048	256	24	10	2/0	No	Y	Y	1	2	1/3
PIC18F2550	32K	16384	2048	256	24	10	2/0	No	Y	Y	1	2	1/3
PIC18F4455	24K	12288	2048	256	35	13	1/1	Yes	Y	Y	1	2	1/3
PIC18F4550	32K	16384	2048	256	35	13	1/1	Yes	Y	Y	1	2	1/3

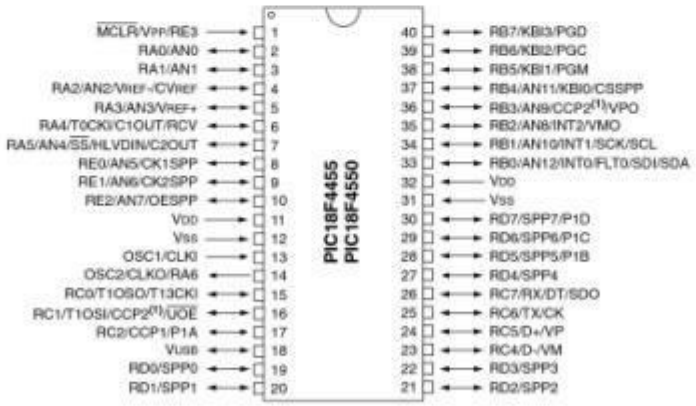
PIC18F2455/2550/4455/4550

Pin Diagrams

28-Pin PDIP, SOIC



40-Pin PDIP



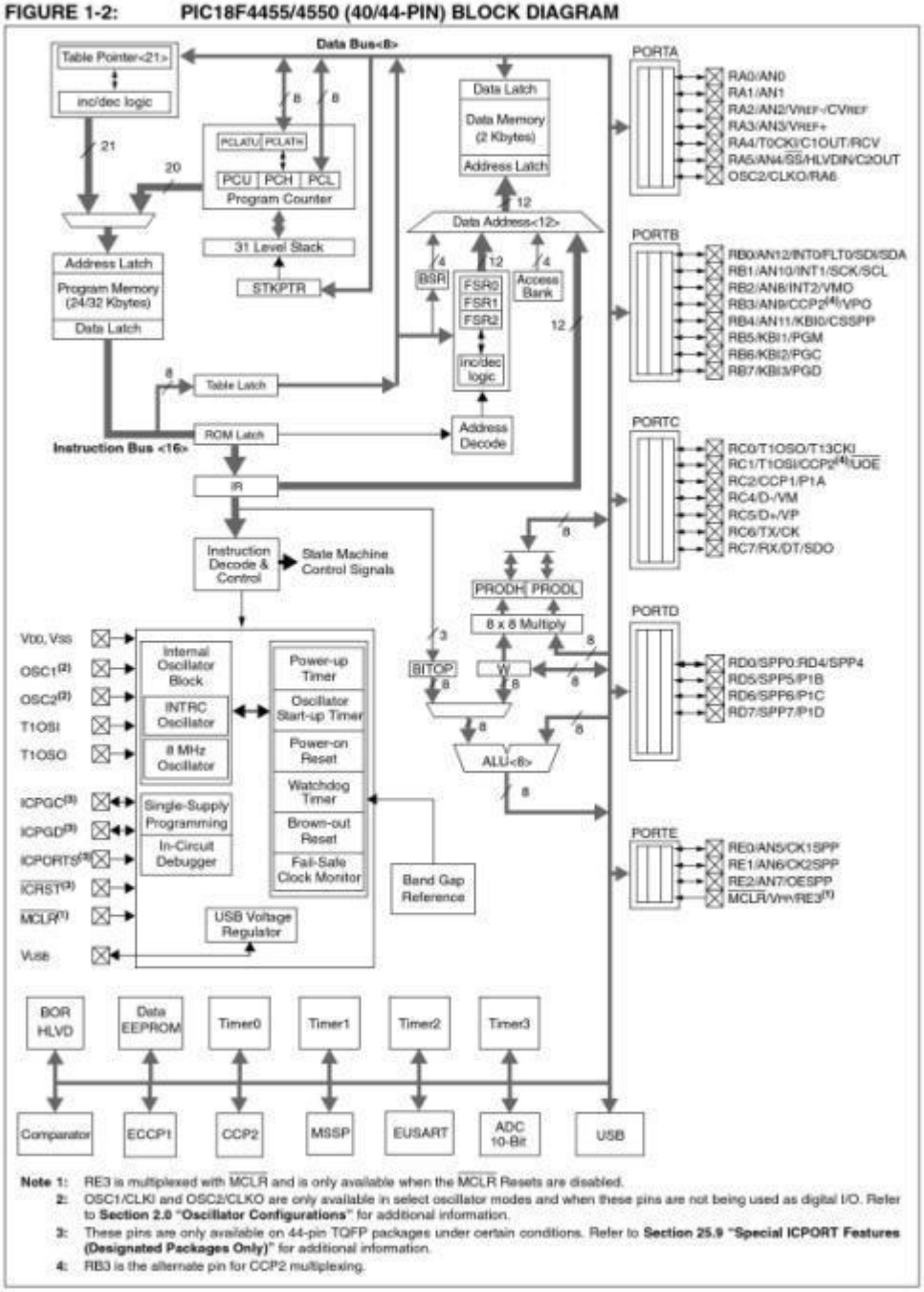
Note 1: RB3 is the alternate pin for CCP2 multiplexing.

PIC18F2455/2550/4455/4550

TABLE 1-1: DEVICE FEATURES

Features	PIC18F2455	PIC18F2550	PIC18F4455	PIC18F4550
Operating Frequency	DC – 48 MHz	DC – 48 MHz	DC – 48 MHz	DC – 48 MHz
Program Memory (Bytes)	24576	32768	24576	32768
Program Memory (Instructions)	12288	16384	12288	16384
Data Memory (Bytes)	2048	2048	2048	2048
Data EEPROM Memory (Bytes)	256	256	256	256
Interrupt Sources	19	19	20	20
I/O Ports	Ports A, B, C, (E)	Ports A, B, C, (E)	Ports A, B, C, D, E	Ports A, B, C, D, E
Timers	4	4	4	4
Capture/Compare/PWM Modules	2	2	1	1
Enhanced Capture/ Compare/PWM Modules	0	0	1	1
Serial Communications	MSSP, Enhanced USART	MSSP, Enhanced USART	MSSP, Enhanced USART	MSSP, Enhanced USART
Universal Serial Bus (USB) Module	1	1	1	1
Streaming Parallel Port (SPP)	No	No	Yes	Yes
10-Bit Analog-to-Digital Module	10 Input Channels	10 Input Channels	13 Input Channels	13 Input Channels
Comparators	2	2	2	2
Resets (and Delays)	POR, BOR, RESET Instruction, Stack Full, Stack Underflow (PWRT, OST), MCLR (optional), WDT	POR, BOR, RESET Instruction, Stack Full, Stack Underflow (PWRT, OST), MCLR (optional), WDT	POR, BOR, RESET Instruction, Stack Full, Stack Underflow (PWRT, OST), MCLR (optional), WDT	POR, BOR, RESET Instruction, Stack Full, Stack Underflow (PWRT, OST), MCLR (optional), WDT
Programmable Low-Voltage Detect	Yes	Yes	Yes	Yes
Programmable Brown-out Reset	Yes	Yes	Yes	Yes
Instruction Set	75 Instructions; 83 with Extended Instruction Set enabled	75 Instructions; 83 with Extended Instruction Set enabled	75 Instructions; 83 with Extended Instruction Set enabled	75 Instructions; 83 with Extended Instruction Set enabled
Packages	28-pin PDIP 28-pin SOIC	28-pin PDIP 28-pin SOIC	40-pin PDIP 44-pin QFN 44-pin TQFP	40-pin PDIP 44-pin QFN 44-pin TQFP

FIGURE 1-2: PIC18F4455/4550 (40/44-PIN) BLOCK DIAGRAM



Note 1: RE3 is multiplexed with MCLR and is only available when the MCLR Resets are disabled.
2: OSC1/CLKI and OSC2/CLKO are only available in select oscillator modes and when these pins are not being used as digital I/O. Refer to **Section 2.0 "Oscillator Configurations"** for additional information.
3: These pins are only available on 44-pin TQFP packages under certain conditions. Refer to **Section 25.9 "Special ICPORT Features (Designated Packages Only)"** for additional information.
4: RB3 is the alternate pin for CCP2 multiplexing.

2.0 OSCILLATOR CONFIGURATIONS

2.1 Overview

Devices in the PIC18F2455/2550/4455/4550 family incorporate a different oscillator and microcontroller clock system than previous PIC18F devices. The addition of the USB module, with its unique requirements for a stable clock source, make it necessary to provide a separate clock source that is compliant with both USB low-speed and full-speed specifications.

To accommodate these requirements, PIC18F2455/2550/4455/4550 devices include a new clock branch to provide a 48 MHz clock for full-speed USB operation. Since it is driven from the primary clock source, an additional system of prescalers and postscalers has been added to accommodate a wide range of oscillator frequencies. An overview of the oscillator structure is shown in Figure 2-1.

Other oscillator features used in PIC18 enhanced microcontrollers, such as the internal oscillator block and clock switching, remain the same. They are discussed later in this chapter.

2.1.1 OSCILLATOR CONTROL

The operation of the oscillator in PIC18F2455/2550/4455/4550 devices is controlled through two Configuration registers and two control registers. Configuration registers, CONFIG1L and CONFIG1H, select the oscillator mode and USB prescaler/postscaler options. As Configuration bits, these are set when the device is programmed and left in that configuration until the device is reprogrammed.

The OSCCON register (Register 2-2) selects the Active Clock mode; it is primarily used in controlling clock switching in power-managed modes. Its use is discussed in **Section 2.4.1 "Oscillator Control Register"**.

The OSCTUNE register (Register 2-1) is used to trim the INTRC frequency source, as well as select the low-frequency clock source that drives several special features. Its use is described in **Section 2.2.5.2 "OSCTUNE Register"**.

2.2 Oscillator Types

PIC18F2455/2550/4455/4550 devices can be operated in twelve distinct oscillator modes. In contrast with previous PIC18 enhanced microcontrollers, four of these modes involve the use of two oscillator types at once. Users can program the FOSC3:FOSC0 Configuration bits to select one of these modes:

1. XT Crystal/Resonator
2. XTPLL Crystal/Resonator with PLL enabled
3. HS High-Speed Crystal/Resonator
4. HSPLL High-Speed Crystal/Resonator with PLL enabled
5. EC External Clock with FOSC/4 output
6. ECIO External Clock with I/O on RA6
7. ECPLL External Clock with PLL enabled and FOSC/4 output on RA6
8. ECPIO External Clock with PLL enabled, I/O on RA6
9. INTHS Internal Oscillator used as microcontroller clock source, HS Oscillator used as USB clock source
10. INTXT Internal Oscillator used as microcontroller clock source, XT Oscillator used as USB clock source
11. INTIO Internal Oscillator used as microcontroller clock source, EC Oscillator used as USB clock source, digital I/O on RA6
12. INTCKO Internal Oscillator used as microcontroller clock source, EC Oscillator used as USB clock source, FOSC/4 output on RA6

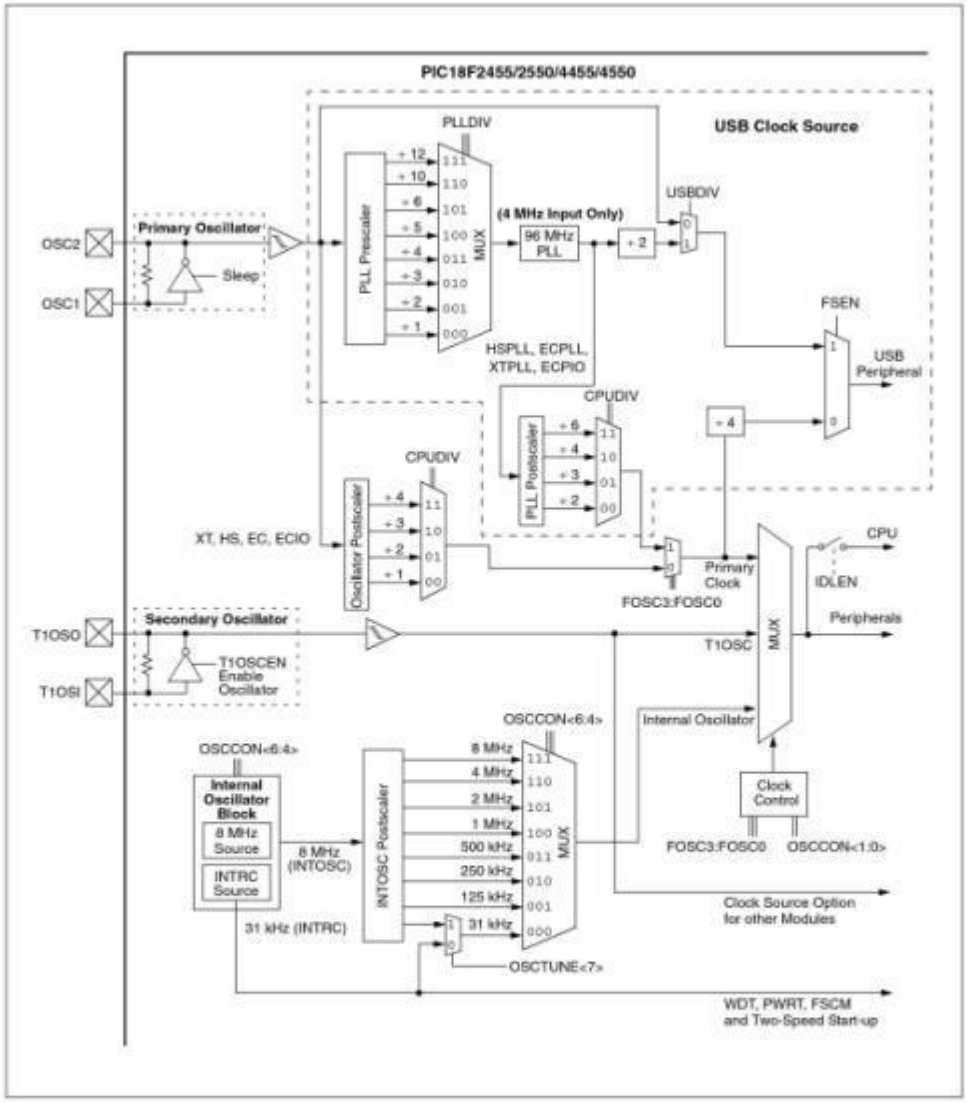
2.2.1 OSCILLATOR MODES AND USB OPERATION

Because of the unique requirements of the USB module, a different approach to clock operation is necessary. In previous PICmicro® devices, all core and peripheral clocks were driven by a single oscillator source; the usual sources were primary, secondary or the internal oscillator. With PIC18F2455/2550/4455/4550 devices, the primary oscillator becomes part of the USB module and cannot be associated to any other clock source. Thus, the USB module must be clocked from the primary clock source; however, the microcontroller core and other peripherals can be separately clocked from the secondary or internal oscillators as before.

Because of the timing requirements imposed by USB, an internal clock of either 6 MHz or 48 MHz is required while the USB module is enabled. Fortunately, the microcontroller and other peripherals are not required to run at this clock speed when using the primary oscillator. There are numerous options to achieve the USB module clock requirement and still provide flexibility for clocking the rest of the device from the primary oscillator source. These are detailed in **Section 2.3 "Oscillator Settings for USB"**.

PIC18F2455/2550/4455/4550

FIGURE 2-1: PIC18F2455/2550/4455/4550 CLOCK DIAGRAM



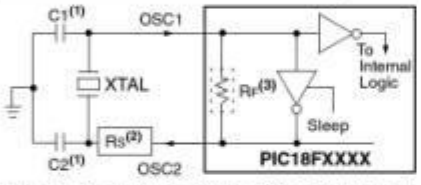
2.2.2 CRYSTAL OSCILLATOR/CERAMIC RESONATORS

In HS, HSPLL, XT and XTPLL Oscillator modes, a crystal or ceramic resonator is connected to the OSC1 and OSC2 pins to establish oscillation. Figure 2-2 shows the pin connections.

The oscillator design requires the use of a parallel cut crystal.

Note: Use of a series cut crystal may give a frequency out of the crystal manufacturer's specifications.

FIGURE 2-2: CRYSTAL/CERAMIC RESONATOR OPERATION (XT, HS OR HSPLL CONFIGURATION)



Note 1: See Table 2-1 and Table 2-2 for initial values of C1 and C2.
2: A series resistor (RS) may be required for AT strip cut crystals.
3: RF varies with the oscillator mode chosen.

TABLE 2-1: CAPACITOR SELECTION FOR CERAMIC RESONATORS

Typical Capacitor Values Used:			
Mode	**Freq**	**OSC1**	**OSC2**
XT	4.0 MHz	33 pF	33 pF
HS	8.0 MHz	27 pF	27 pF
	16.0 MHz	22 pF	22 pF

Capacitor values are for design guidance only.

These capacitors were tested with the resonators listed below for basic start-up and operation. **These values are not optimized**.

Different capacitor values may be required to produce acceptable oscillator operation. The user should test the performance of the oscillator over the expected VDD and temperature range for the application.

See the notes following Table 2-2 for additional information.

Resonators Used:
4.0 MHz
8.0 MHz
16.0 MHz

TABLE 2-2: CAPACITOR SELECTION FOR CRYSTAL OSCILLATOR

Osc Type	Crystal Freq	Typical Capacitor Values Tested:	
		C1	**C2**
XT	4 MHz	27 pF	27 pF
HS	4 MHz	27 pF	27 pF
	8 MHz	22 pF	22 pF
	20 MHz	15 pF	15 pF

Capacitor values are for design guidance only.

These capacitors were tested with the crystals listed below for basic start-up and operation. **These values are not optimized.**

Different capacitor values may be required to produce acceptable oscillator operation. The user should test the performance of the oscillator over the expected VDD and temperature range for the application.

See the notes following this table for additional information.

Crystals Used:
4 MHz
8 MHz
20 MHz

Note 1: Higher capacitance increases the stability of oscillator but also increases the start-up time.
2: When operating below 3V VDD, or when using certain ceramic resonators at any voltage, it may be necessary to use the HS mode or switch to a crystal oscillator.
3: Since each resonator/crystal has its own characteristics, the user should consult the resonator/crystal manufacturer for appropriate values of external components.
4: Rs may be required to avoid overdriving crystals with low drive level specification.
5: Always verify oscillator performance over the VDD and temperature range that is expected for the application.

An internal postscaler allows users to select a clock frequency other than that of the crystal or resonator. Frequency division is determined by the CPUDIV Configuration bits. Users may select a clock frequency of the oscillator frequency, or 1/2, 1/3 or 1/4 of the frequency.

An external clock may also be used when the microcontroller is in HS Oscillator mode. In this case, the OSC2/CLKO pin is left open (Figure 2-3).

4.5 Device Reset Timers

PIC18F2455/2550/4455/4550 devices incorporate three separate on-chip timers that help regulate the Power-on Reset process. Their main function is to ensure that the device clock is stable before code is executed. These timers are:

- Power-up Timer (PWRT)
- Oscillator Start-up Timer (OST)
- PLL Lock Time-out

4.5.1 POWER-UP TIMER (PWRT)

The Power-up Timer (PWRT) of the PIC18F2455/2550/4455/4550 devices is an 11-bit counter which uses the INTRC source as the clock input. This yields an approximate time interval of 2048 x 32 µs = 65.6 ms. While the PWRT is counting, the device is held in Reset.

The power-up time delay depends on the INTRC clock and will vary from chip to chip due to temperature and process variation. See DC parameter 33 (Table 28-12) for details.

The PWRT is enabled by clearing the $\overline{\text{PWRTEN}}$ Configuration bit.

4.5.2 OSCILLATOR START-UP TIMER (OST)

The Oscillator Start-up Timer (OST) provides a 1024 oscillator cycle (from OSC1 input) delay after the PWRT delay is over (parameter 33, Table 28-12). This ensures that the crystal oscillator or resonator has started and stabilized.

The OST time-out is invoked only for XT, HS and HSPLL modes and only on Power-on Reset or on exit from most power-managed modes.

4.5.3 PLL LOCK TIME-OUT

With the PLL enabled in its PLL mode, the time-out sequence following a Power-on Reset is slightly different from other oscillator modes. A separate timer is used to provide a fixed time-out that is sufficient for the PLL to lock to the main oscillator frequency. This PLL lock time-out (T_{PLL}) is typically 2 ms and follows the oscillator start-up time-out.

4.5.4 TIME-OUT SEQUENCE

On power-up, the time-out sequence is as follows:

1. After the POR condition has cleared, PWRT time-out is invoked (if enabled).
2. Then, the OST is activated.

The total time-out will vary based on oscillator configuration and the status of the PWRT. Figure 4-3, Figure 4-4, Figure 4-5, Figure 4-6 and Figure 4-7 all depict time-out sequences on power-up, with the Power-up Timer enabled and the device operating in HS Oscillator mode. Figures 4-3 through 4-6 also apply to devices operating in XT mode. For devices in RC mode and with the PWRT disabled, on the other hand, there will be no time-out at all.

Since the time-outs occur from the POR pulse, if $\overline{\text{MCLR}}$ is kept low long enough, all time-outs will expire. Bringing $\overline{\text{MCLR}}$ high will begin execution immediately (Figure 4-5). This is useful for testing purposes or to synchronize more than one PIC18FXXXX device operating in parallel.

TABLE 4-2: TIME-OUT IN VARIOUS SITUATIONS

Oscillator Configuration	Power-up[2] and Brown-out		Exit from Power-Managed Mode
	$\overline{\text{PWRTEN}}$ = 0	$\overline{\text{PWRTEN}}$ = 1	
HS, XT	66 ms[1] + 1024 TOSC	1024 TOSC	1024 TOSC
HSPLL, XTPLL	66 ms[1] + 1024 TOSC + 2 ms[2]	1024 TOSC + 2 ms[2]	1024 TOSC + 2 ms[2]
EC, ECIO	66 ms[1]	—	—
ECPLL, ECPIO	66 ms[1] + 2 ms[2]	2 ms[2]	2 ms[2]
INTIO, INTCKO	66 ms[1]	—	—
INTHS, INTXT	66 ms[1] + 1024 TOSC	1024 TOSC	1024 TOSC

Note 1: 66 ms (65.5 ms) is the nominal Power-up Timer (PWRT) delay.
2: 2 ms is the nominal time required for the PLL to lock.

12.0 TIMER1 MODULE

The Timer1 timer/counter module incorporates these features:

- Software selectable operation as a 16-bit timer or counter
- Readable and writable 8-bit registers (TMR1H and TMR1L)
- Selectable clock source (internal or external) with device clock or Timer1 oscillator internal options
- Interrupt on overflow
- Module Reset on CCP Special Event Trigger
- Device clock status flag (T1RUN)

A simplified block diagram of the Timer1 module is shown in Figure 12-1. A block diagram of the module's operation in Read/Write mode is shown in Figure 12-2.

The module incorporates its own low-power oscillator to provide an additional clocking option. The Timer1 oscillator can also be used as a low-power clock source for the microcontroller in power-managed operation.

Timer1 can also be used to provide Real-Time Clock (RTC) functionality to applications with only a minimal addition of external components and code overhead.

Timer1 is controlled through the T1CON Control register (Register 12-1). It also contains the Timer1 Oscillator Enable bit (T1OSCEN). Timer1 can be enabled or disabled by setting or clearing control bit, TMR1ON (T1CON<0>).

REGISTER 12-1: T1CON: TIMER1 CONTROL REGISTER

R/W-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
RD16	T1RUN	T1CKPS1	T1CKPS0	T1OSCEN	$\overline{\text{T1SYNC}}$	TMR1CS	TMR1ON
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 7 **RD16:** 16-Bit Read/Write Mode Enable bit
1 = Enables register read/write of Timer1 in one 16-bit operation
0 = Enables register read/write of Timer1 in two 8-bit operations

bit 6 **T1RUN:** Timer1 System Clock Status bit
1 = Device clock is derived from Timer1 oscillator
0 = Device clock is derived from another source

bit 5-4 **T1CKPS1:T1CKPS0:** Timer1 Input Clock Prescale Select bits
11 = 1:8 Prescale value
10 = 1:4 Prescale value
01 = 1:2 Prescale value
00 = 1:1 Prescale value

bit 3 **T1OSCEN:** Timer1 Oscillator Enable bit
1 = Timer1 oscillator is enabled
0 = Timer1 oscillator is shut off
The oscillator inverter and feedback resistor are turned off to eliminate power drain.

bit 2 **$\overline{\text{T1SYNC}}$:** Timer1 External Clock Input Synchronization Select bit
When TMR1CS = 1:
1 = Do not synchronize external clock input
0 = Synchronize external clock input
When TMR1CS = 0:
This bit is ignored. Timer1 uses the internal clock when TMR1CS = 0.

bit 1 **TMR1CS:** Timer1 Clock Source Select bit
1 = External clock from RC0/T1OSO/T13CKI pin (on the rising edge)
0 = Internal clock (FOSC/4)

bit 0 **TMR1ON:** Timer1 On bit
1 = Enables Timer1
0 = Stops Timer1

PIC18F2455/2550/4455/4550

12.1 Timer1 Operation

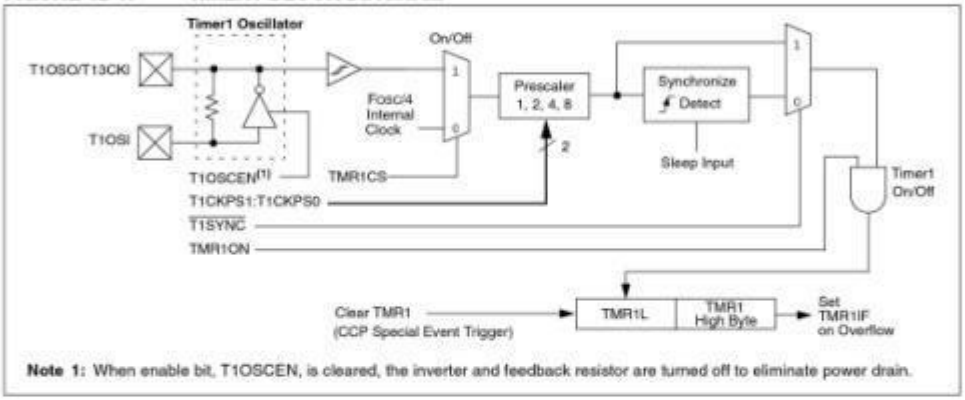
Timer1 can operate in one of these modes:

- Timer
- Synchronous Counter
- Asynchronous Counter

The operating mode is determined by the clock select bit, TMR1CS (T1CON<1>). When TMR1CS is cleared (= 0), Timer1 increments on every internal instruction cycle (FOSC/4). When the bit is set, Timer1 increments on every rising edge of the Timer1 external clock input or the Timer1 oscillator, if enabled.

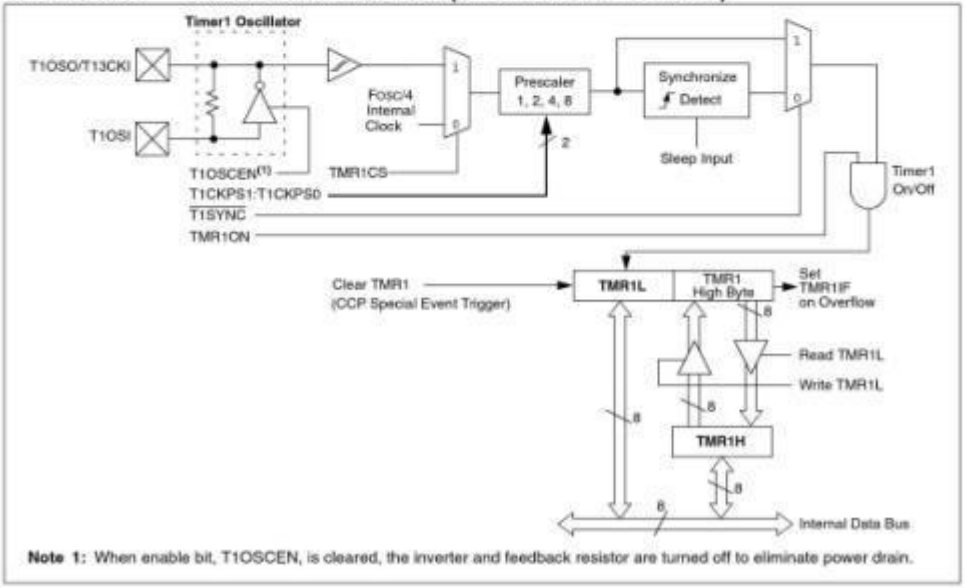
When Timer1 is enabled, the RC1/T1OSI/$\overline{UOE}$ and RC0/T1OSO/T13CKI pins become inputs. This means the values of TRISC<1:0> are ignored and the pins are read as '0'.

FIGURE 12-1: TIMER1 BLOCK DIAGRAM



Note 1: When enable bit, T1OSCEN, is cleared, the inverter and feedback resistor are turned off to eliminate power drain.

FIGURE 12-2: TIMER1 BLOCK DIAGRAM (16-BIT READ/WRITE MODE)



Note 1: When enable bit, T1OSCEN, is cleared, the inverter and feedback resistor are turned off to eliminate power drain.

12.2 Timer1 16-Bit Read/Write Mode

Timer1 can be configured for 16-bit reads and writes (see Figure 12-2). When the RD16 control bit (T1CON<7>) is set, the address for TMR1H is mapped to a buffer register for the high byte of Timer1. A read from TMR1L will load the contents of the high byte of Timer1 into the Timer1 high byte buffer. This provides the user with the ability to accurately read all 16 bits of Timer1 without having to determine whether a read of the high byte, followed by a read of the low byte, has become invalid due to a rollover between reads.

A write to the high byte of Timer1 must also take place through the TMR1H Buffer register. The Timer1 high byte is updated with the contents of TMR1H when a write occurs to TMR1L. This allows a user to write all 16 bits to both the high and low bytes of Timer1 at once.

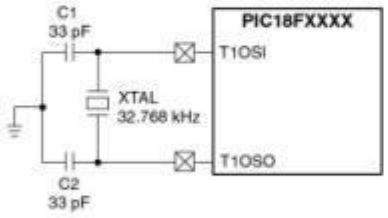
The high byte of Timer1 is not directly readable or writable in this mode. All reads and writes must take place through the Timer1 High Byte Buffer register. Writes to TMR1H do not clear the Timer1 prescaler. The prescaler is only cleared on writes to TMR1L.

12.3 Timer1 Oscillator

An on-chip crystal oscillator circuit is incorporated between pins T1OSI (input) and T1OSO (amplifier output). It is enabled by setting the Timer1 Oscillator Enable bit, T1OSCEN (T1CON<3>). The oscillator is a low-power circuit rated for 32 kHz crystals. It will continue to run during all power-managed modes. The circuit for a typical LP oscillator is shown in Figure 12-3. Table 12-1 shows the capacitor selection for the Timer1 oscillator.

The user must provide a software time delay to ensure proper start-up of the Timer1 oscillator.

FIGURE 12-3: EXTERNAL COMPONENTS FOR THE TIMER1 LP OSCILLATOR



Note: See the Notes with Table 12-1 for additional information about capacitor selection.

TABLE 12-1: CAPACITOR SELECTION FOR THE TIMER OSCILLATOR(2,3,4)

Osc Type	Freq	C1	C2
LP	32 kHz	27 pF(1)	27 pF(1)

Note 1: Microchip suggests these values as a starting point in validating the oscillator circuit.

2: Higher capacitance increases the stability of the oscillator but also increases the start-up time.

3: Since each resonator/crystal has its own characteristics, the user should consult the resonator/crystal manufacturer for appropriate values of external components.

4: Capacitor values are for design guidance only.

12.3.1 USING TIMER1 AS A CLOCK SOURCE

The Timer1 oscillator is also available as a clock source in power-managed modes. By setting the clock select bits, SCS1:SCS0 (OSCCON<1:0>), to '01', the device switches to SEC_RUN mode. Both the CPU and peripherals are clocked from the Timer1 oscillator. If the IDLEN bit (OSCCON<7>) is cleared and a SLEEP instruction is executed, the device enters SEC_IDLE mode. Additional details are available in **Section 3.0 "Power-Managed Modes"**.

Whenever the Timer1 oscillator is providing the clock source, the Timer1 system clock status flag, T1RUN (T1CON<6>), is set. This can be used to determine the controller's current clocking mode. It can also indicate the clock source being currently used by the Fail-Safe Clock Monitor. If the Clock Monitor is enabled and the Timer1 oscillator fails while providing the clock, polling the T1RUN bit will indicate whether the clock is being provided by the Timer1 oscillator or another source.

12.3.2 LOW-POWER TIMER1 OPTION

The Timer1 oscillator can operate at two distinct levels of power consumption based on device configuration. When the LPT1OSC Configuration bit is set, the Timer1 oscillator operates in a low-power mode. When LPT1OSC is not set, Timer1 operates at a higher power level. Power consumption for a particular mode is relatively constant, regardless of the device's operating mode. The default Timer1 configuration is the higher power mode.

As the low-power Timer1 mode tends to be more sensitive to interference, high noise environments may cause some oscillator instability. The low-power option is, therefore, best suited for low noise applications where power conservation is an important design consideration.

19.0 MASTER SYNCHRONOUS SERIAL PORT (MSSP) MODULE

19.1 Master SSP (MSSP) Module Overview

The Master Synchronous Serial Port (MSSP) module is a serial interface, useful for communicating with other peripheral or microcontroller devices. These peripheral devices may be serial EEPROMs, shift registers, display drivers, A/D converters, etc. The MSSP module can operate in one of two modes:

- Serial Peripheral Interface (SPI)
- Inter-Integrated Circuit (I^2C™)
  - Full Master mode
  - Slave mode (with general address call)

The I^2C interface supports the following modes in hardware:

- Master mode
- Multi-Master mode
- Slave mode

19.2 Control Registers

The MSSP module has three associated control registers. These include a status register (SSPSTAT) and two control registers (SSPCON1 and SSPCON2). The use of these registers and their individual Configuration bits differ significantly depending on whether the MSSP module is operated in SPI or I^2C mode.

Additional details are provided under the individual sections.

19.3 SPI Mode

The SPI mode allows 8 bits of data to be synchronously transmitted and received simultaneously. All four modes of the SPI are supported. To accomplish communication, typically three pins are used:

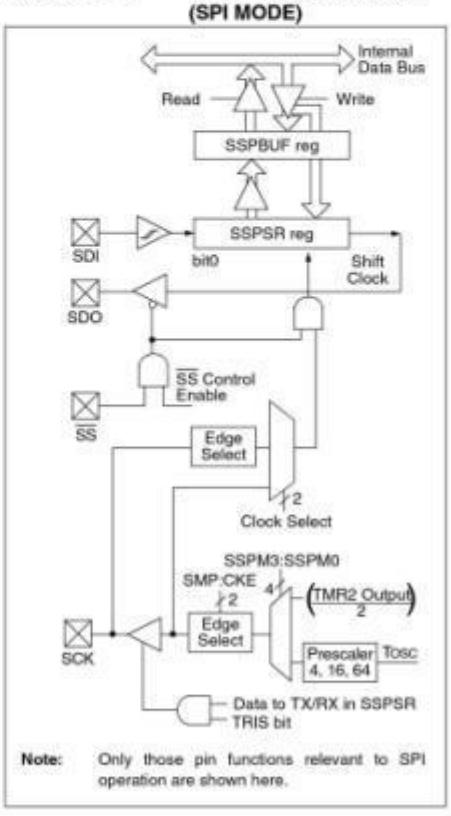
- Serial Data Out (SDO) – RC7/RX/DT/SDO
- Serial Data In (SDI) – RB0/AN12/INT0/FLT0/SDI/SDA
- Serial Clock (SCK) – RB1/AN10/INT1/SCK/SCL

Additionally, a fourth pin may be used when in a Slave mode of operation:

- Slave Select ($\overline{SS}$) – RA5/AN4/$\overline{SS}$/HLVDIN/C2OUT

Figure 19-1 shows the block diagram of the MSSP module when operating in SPI mode.

FIGURE 19-1: MSSP BLOCK DIAGRAM (SPI MODE)



Note: Only those pin functions relevant to SPI operation are shown here.

PIC18F2455/2550/4455/4550

19.3.2 OPERATION

When initializing the SPI, several options need to be specified. This is done by programming the appropriate control bits (SSPCON1<5:0> and SSPSTAT<7:6>). These control bits allow the following to be specified:

- Master mode (SCK is the clock output)
- Slave mode (SCK is the clock input)
- Clock Polarity (Idle state of SCK)
- Data Input Sample Phase (middle or end of data output time)
- Clock Edge (output data on rising/falling edge of SCK)
- Clock Rate (Master mode only)
- Slave Select mode (Slave mode only)

The MSSP module consists of a transmit/receive shift register (SSPSR) and a buffer register (SSPBUF). The SSPSR shifts the data in and out of the device, MSb first. The SSPBUF holds the data that was written to the SSPSR until the received data is ready. Once the eight bits of data have been received, that byte is moved to the SSPBUF register. Then, the Buffer Full detect bit, BF (SSPSTAT<0>) and the interrupt flag bit, SSPIF, are set. This double-buffering of the received data (SSPBUF) allows the next byte to start reception before reading the data that was just received. Any write to the SSPBUF register during transmission/reception of data will be ignored and the Write Collision detect bit, WCOL (SSPCON1<7>), will be set. User software must clear the WCOL bit so that it can be determined if the following write(s) to the SSPBUF register completed successfully.

When the application software is expecting to receive valid data, the SSPBUF should be read before the next byte of data to transfer is written to the SSPBUF. The Buffer Full bit, BF (SSPSTAT<0>), indicates when SSPBUF has been loaded with the received data (transmission is complete). When the SSPBUF is read, the BF bit is cleared. This data may be irrelevant if the SPI is only a transmitter. Generally, the MSSP interrupt is used to determine when the transmission/reception has completed. If the interrupt method is not going to be used, then software polling can be done to ensure that a write collision does not occur. Example 19-1 shows the loading of the SSPBUF (SSPSR) for data transmission.

The SSPSR is not directly readable or writable and can only be accessed by addressing the SSPBUF register. Additionally, the MSSP Status register (SSPSTAT) indicates the various status conditions.

EXAMPLE 19-1: LOADING THE SSPBUF (SSPSR) REGISTER

```
LOOP    BTFSS   SSPSTAT, BF     ;Has data been received (transmit complete)?
        BRA     LOOP            ;No
        MOVF    SSPBUF, W       ;WREG reg = contents of SSPBUF
        MOVWF   RXDATA          ;Save in user RAM, if data is meaningful
        MOVF    TXDATA, W       ;W reg = contents of TXDATA
        MOVWF   SSPBUF          ;New data to xmit
```

19.3.3 ENABLING SPI I/O

To enable the serial port, MSSP Enable bit, SSPEN (SSPCON1<5>), must be set. To reset or reconfigure SPI mode, clear the SSPEN bit, reinitialize the SSPCON registers and then set the SSPEN bit. This configures the SDI, SDO, SCK and $\overline{SS}$ pins as serial port pins. For the pins to behave as the serial port function, some must have their data direction bits (in the TRIS register) appropriately programmed as follows:

- SDI is automatically controlled by the SPI module
- SDO must have TRISC<7> bit cleared
- SCK (Master mode) must have TRISB<1> bit cleared
- SCK (Slave mode) must have TRISB<1> bit set
- $\overline{SS}$ must have TRISA<5> bit set

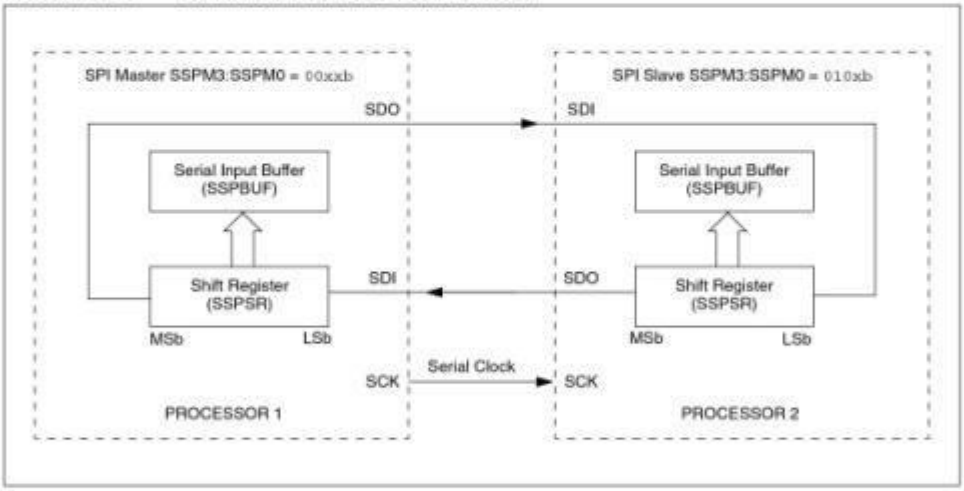
Any serial port function that is not desired may be overridden by programming the corresponding data direction (TRIS) register to the opposite value.

19.3.4 TYPICAL CONNECTION

Figure 19-2 shows a typical connection between two microcontrollers. The master controller (Processor 1) initiates the data transfer by sending the SCK signal. Data is shifted out of both shift registers on their programmed clock edge and latched on the opposite edge of the clock. Both processors should be programmed to the same Clock Polarity (CKP), then both controllers would send and receive data at the same time. Whether the data is meaningful (or dummy data) depends on the application software. This leads to three scenarios for data transmission:

- Master sends data – Slave sends dummy data
- Master sends data – Slave sends data
- Master sends dummy data – Slave sends data

FIGURE 19-2: SPI MASTER/SLAVE CONNECTION



21.0 10-BIT ANALOG-TO-DIGITAL CONVERTER (A/D) MODULE

The Analog-to-Digital (A/D) converter module has 10 inputs for the 28-pin devices and 13 for the 40/44-pin devices. This module allows conversion of an analog input signal to a corresponding 10-bit digital number.

The module has five registers:

- A/D Result High Register (ADRESH)
- A/D Result Low Register (ADRESL)
- A/D Control Register 0 (ADCON0)
- A/D Control Register 1 (ADCON1)
- A/D Control Register 2 (ADCON2)

The ADCON0 register, shown in Register 21-1, controls the operation of the A/D module. The ADCON1 register, shown in Register 21-2, configures the functions of the port pins. The ADCON2 register, shown in Register 21-3, configures the A/D clock source, programmed acquisition time and justification.

REGISTER 21-1: ADCON0: A/D CONTROL REGISTER 0

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	CHS3	CHS2	CHS1	CHS0	GO/$\overline{\text{DONE}}$	ADON
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 7-6 **Unimplemented:** Read as '0'

bit 5-2 **CHS3:CHS0:** Analog Channel Select bits
0000 = Channel 0 (AN0)
0001 = Channel 1 (AN1)
0010 = Channel 2 (AN2)
0011 = Channel 3 (AN3)
0100 = Channel 4 (AN4)
0101 = Channel 5 (AN5)(1,2)
0110 = Channel 6 (AN6)(1,2)
0111 = Channel 7 (AN7)(1,2)
1000 = Channel 8 (AN8)
1001 = Channel 9 (AN9)
1010 = Channel 10 (AN10)
1011 = Channel 11 (AN11)
1100 = Channel 12 (AN12)
1101 = Unimplemented(2)
1110 = Unimplemented(2)
1111 = Unimplemented(2)

bit 1 **GO/$\overline{\text{DONE}}$:** A/D Conversion Status bit
When ADON = 1:
1 = A/D conversion in progress
0 = A/D Idle

bit 0 **ADON:** A/D On bit
1 = A/D converter module is enabled
0 = A/D converter module is disabled

Note 1: These channels are not implemented on 28-pin devices.
2: Performing a conversion on unimplemented channels will return a floating input measurement.

Hoja de datos MFRC522

MFRC522

Standard 3V MIFARE reader solution

Rev. 3.8 — 17 September 2014
112138

Product data sheet
COMPANY PUBLIC

1. Introduction

This document describes the functionality and electrical specifications of the contactless reader/writer MFRC522.

Remark: The MFRC522 supports all variants of the MIFARE Mini, MIFARE 1K, MIFARE 4K, MIFARE Ultralight, MIFARE DESFire EV1 and MIFARE Plus RF identification protocols. To aid readability throughout this data sheet, the MIFARE Mini, MIFARE 1K, MIFARE 4K, MIFARE Ultralight, MIFARE DESFire EV1 and MIFARE Plus products and protocols have the generic name MIFARE.

2. General description

The MFRC522 is a highly integrated reader/writer IC for contactless communication at 13.56 MHz. The MFRC522 reader supports ISO/IEC 14443 A/MIFARE mode.

The MFRC522's internal transmitter is able to drive a reader/writer antenna designed to communicate with ISO/IEC 14443 A/MIFARE cards and transponders without additional active circuitry. The receiver module provides a robust and efficient implementation for demodulating and decoding signals from ISO/IEC 14443 A/MIFARE compatible cards and transponders. The digital module manages the complete ISO/IEC 14443 A framing and error detection (parity and CRC) functionality.

The MFRC522 supports MF1xxS20, MF1xxS70 and MF1xxS50 products. The MFRC522 supports contactless communication and uses MIFARE higher transfer speeds up to 848 kBd in both directions.

The following host interfaces are provided:

- Serial Peripheral Interface (SPI)
- Serial UART (similar to RS232 with voltage levels dependant on pin voltage supply)
- I^2C-bus interface

2.1 Differences between version 1.0 and 2.0

The MFRC522 is available in two versions:

- MFRC52201HN1, hereafter referred to version 1.0 and
- MFRC52202HN1, hereafter referred to version 2.0.

The MFRC522 version 2.0 is fully compatible to version 1.0 and offers in addition the following features and improvements:

- Increased stability of the reader IC in rough conditions
- An additional timer prescaler, see Section 8.5.
- A corrected CRC handling when RX Multiple is set to 1

This data sheet version covers both versions of the MFRC522 and describes the differences between the versions if applicable.

3. Features and benefits

- Highly integrated analog circuitry to demodulate and decode responses
- Buffered output drivers for connecting an antenna with the minimum number of external components
- Supports ISO/IEC 14443 A/MIFARE
- Typical operating distance in Read/Write mode up to 50 mm depending on the antenna size and tuning
- Supports MF1xxS20, MF1xxS70 and MF1xxS50 encryption in Read/Write mode
- Supports ISO/IEC 14443 A higher transfer speed communication up to 848 kBd
- Supports MFIN/MFOUT
- Additional internal power supply to the smart card IC connected via MFIN/MFOUT
- Supported host interfaces
  - SPI up to 10 Mbit/s
  - I^2C-bus interface up to 400 kBd in Fast mode, up to 3400 kBd in High-speed mode
  - RS232 Serial UART up to 1228.8 kBd, with voltage levels dependant on pin voltage supply
- FIFO buffer handles 64 byte send and receive
- Flexible interrupt modes
- Hard reset with low power function
- Power-down by software mode
- Programmable timer
- Internal oscillator for connection to 27.12 MHz quartz crystal
- 2.5 V to 3.3 V power supply
- CRC coprocessor
- Programmable I/O pins
- Internal self-test

4. Quick reference data

Table 1. Quick reference data

Symbol	Parameter	Conditions		Min	Typ	Max	Unit
V_{DDA}	analog supply voltage	$V_{DD(PVDD)} \leq V_{DDA} = V_{DDD} = V_{DD(TVDD)}$; $V_{SSA} = V_{SSD} = V_{SS(PVSS)} = V_{SS(TVSS)} = 0$ V	[1][2]	2.5	3.3	3.6	V
V_{DDD}	digital supply voltage			2.5	3.3	3.6	V
$V_{DD(TVDD)}$	TVDD supply voltage			2.5	3.3	3.6	V
$V_{DD(PVDD)}$	PVDD supply voltage		[3]	1.6	1.8	3.6	V
$V_{DD(SVDD)}$	SVDD supply voltage	$V_{SSA} = V_{SSD} = V_{SS(PVSS)} = V_{SS(TVSS)} = 0$ V		1.6	-	3.6	V

6. Block diagram

The analog interface handles the modulation and demodulation of the analog signals.

The contactless UART manages the protocol requirements for the communication protocols in cooperation with the host. The FIFO buffer ensures fast and convenient data transfer to and from the host and the contactless UART and vice versa.

Various host interfaces are implemented to meet different customer requirements.

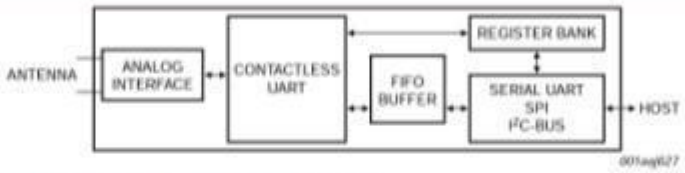


Fig 1. Simplified block diagram of the MFRC522

7. Pinning information

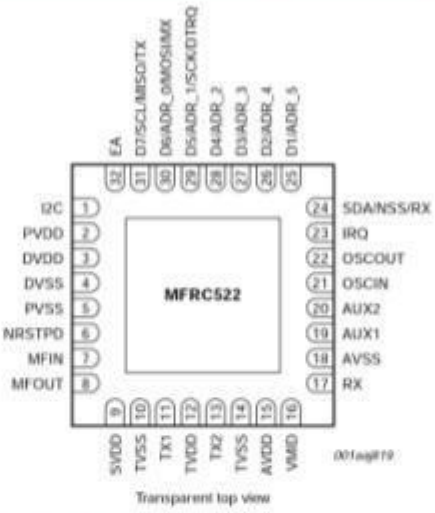


Fig 3. Pinning configuration HVQFN32 (SOT617-1)

7.1 Pin description

Table 3. Pin description

Pin	Symbol	Type[1]	Description
1	I2C	I	I²C-bus enable input[2]
2	PVDD	P	pin power supply
3	DVDD	P	digital power supply
4	DVSS	G	digital ground[3]
5	PVSS	G	pin power supply ground
6	NRSTPD	I	reset and power-down input: power-down: enabled when LOW; internal current sinks are switched off, the oscillator is inhibited and the input pins are disconnected from the outside world reset: enabled by a positive edge
7	MFIN	I	MIFARE signal input
8	MFOUT	O	MIFARE signal output
9	SVDD	P	MFIN and MFOUT pin power supply
10	TVSS	G	transmitter output stage 1 ground
11	TX1	O	transmitter 1 modulated 13.56 MHz energy carrier output
12	TVDD	P	transmitter power supply: supplies the output stage of transmitters 1 and 2
13	TX2	O	transmitter 2 modulated 13.56 MHz energy carrier output
14	TVSS	G	transmitter output stage 2 ground
15	AVDD	P	analog power supply

Table 3. Pin description ...continued

Pin	Symbol	Type[1]	Description
16	VMID	P	internal reference voltage
17	RX	I	RF signal input
18	AVSS	G	analog ground
19	AUX1	O	auxiliary outputs for test purposes
20	AUX2	O	auxiliary outputs for test purposes
21	OSCIN	I	crystal oscillator inverting amplifier input; also the input for an externally generated clock (f_{clk} = 27.12 MHz)
22	OSCOUT	O	crystal oscillator inverting amplifier output
23	IRQ	O	interrupt request output: indicates an interrupt event
24	SDA	I/O	I²C-bus serial data line input/output[2]
	NSS	I	SPI signal input[2]
	RX	I	UART address input[2]
25	D1	I/O	test port[2]
	ADR_5	I/O	I²C-bus address 5 input[2]
26	D2	I/O	test port
	ADR_4	I	I²C-bus address 4 input[2]
27	D3	I/O	test port
	ADR_3	I	I²C-bus address 3 input[2]
28	D4	I/O	test port
	ADR_2	I	I²C-bus address 2 input[2]
29	D5	I/O	test port
	ADR_1	I	I²C-bus address 1 input[2]
	SCK	I	SPI serial clock input[2]
	DTRQ	O	UART request to send output to microcontroller[2]
30	D6	I/O	test port
	ADR_0	I	I²C-bus address 0 input[2]
	MOSI	I/O	SPI master out, slave in[2]
	MX	O	UART output to microcontroller[2]
31	D7	I/O	test port
	SCL	I/O	I²C-bus clock input/output[2]
	MISO	I/O	SPI master in, slave out[2]
	TX	O	UART data output to microcontroller[2]
32	EA	I	external address input for coding I²C-bus address[2]

[1] Pin types: I = Input, O = Output, I/O = Input/Output, P = Power and G = Ground.

[2] The pin functionality of these pins is explained in Section 8.1 "Digital interfaces".

[3] Connection of heatsink pad on package bottom side is not necessary. Optional connection to pin DVSS is possible.

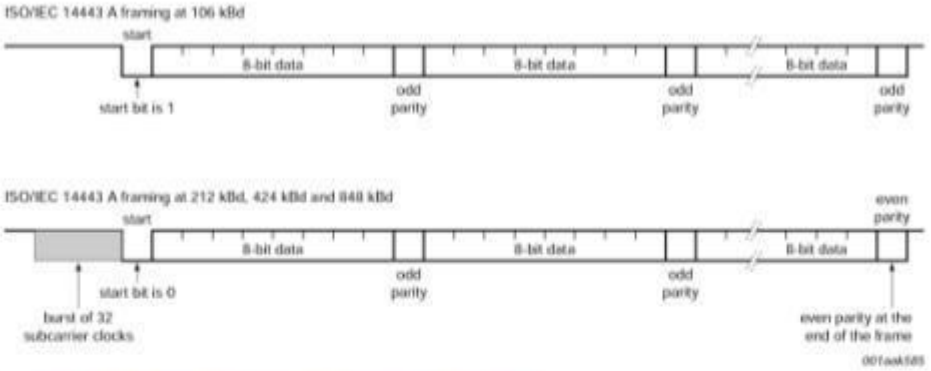


Fig 6. Data coding and framing according to ISO/IEC 14443 A

The internal CRC coprocessor calculates the CRC value based on ISO/IEC 14443 A part 3 and handles parity generation internally according to the transfer speed. Automatic parity generation can be switched off using the MfRxReg register's ParityDisable bit.

8.1 Digital interfaces

8.1.1 Automatic microcontroller interface detection

The MFRC522 supports direct interfacing of hosts using SPI, I²C-bus or serial UART interfaces. The MFRC522 resets its interface and checks the current host interface type automatically after performing a power-on or hard reset. The MFRC522 identifies the host interface by sensing the logic levels on the control pins after the reset phase. This is done using a combination of fixed pin connections. Table 5 shows the different connection configurations.

Table 5. Connection protocol for detecting different interface types

Pin	Interface type		
	UART (input)	SPI (output)	I²C-bus (I/O)
SDA	RX	NSS	SDA
I2C	0	0	1
EA	0	1	EA
D7	TX	MISO	SCL
D6	MX	MOSI	ADR_0
D5	DTRQ	SCK	ADR_1
D4	-	-	ADR_2
D3	-	-	ADR_3
D2	-	-	ADR_4
D1	-	-	ADR_5

8.1.2 Serial Peripheral Interface

A serial peripheral interface (SPI compatible) is supported to enable high-speed communication to the host. The interface can handle data speeds up to 10 Mbit/s. When communicating with a host, the MFRC522 acts as a slave, receiving data from the external host for register settings, sending and receiving data relevant for RF interface communication.

An interface compatible with SPI enables high-speed serial communication between the MFRC522 and a microcontroller. The implemented interface is in accordance with the SPI standard.

The timing specification is given in Section 14.1 on page 78.

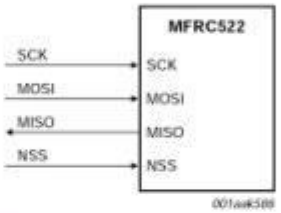


Fig 7. SPI connection to host

The MFRC522 acts as a slave during SPI communication. The SPI clock signal SCK must be generated by the master. Data communication from the master to the slave uses the MOSI line. The MISO line is used to send data from the MFRC522 to the master.

Data bytes on both MOSI and MISO lines are sent with the MSB first. Data on both MOSI and MISO lines must be stable on the rising edge of the clock and can be changed on the falling edge. Data is provided by the MFRC522 on the falling clock edge and is stable during the rising clock edge.

8.1.2.1 SPI read data

Reading data using SPI requires the byte order shown in Table 6 to be used. It is possible to read out up to n-data bytes.

The first byte sent defines both the mode and the address.

Table 6. MOSI and MISO byte order

Line	Byte 0	Byte 1	Byte 2	To	Byte n	Byte n + 1
MOSI	address 0	address 1	address 2	...	address n	00
MISO	X[1]	data 0	data 1	...	data n − 1	data n

[1] X = Do not care.

Remark: The MSB must be sent first.

8.1.2.2 SPI write data

To write data to the MFRC522 using SPI requires the byte order shown in Table 7. It is possible to write up to n data bytes by only sending one address byte.

The first send byte defines both the mode and the address byte.

Table 7. MOSI and MISO byte order

Line	Byte 0	Byte 1	Byte 2	To	Byte n	Byte n + 1
MOSI	address 0	data 0	data 1	...	data n − 1	data n
MISO	X[1]	X[1]	X[1]	...	X[1]	X[1]

[1] X = Do not care.

Remark: The MSB must be sent first.

8.1.2.3 SPI address byte

The address byte must meet the following format.

The MSB of the first byte defines the mode used. To read data from the MFRC522 the MSB is set to logic 1. To write data to the MFRC522 the MSB must be set to logic 0. Bits 6 to 1 define the address and the LSB is set to logic 0.

Table 8. Address byte 0 register; address MOSI

7 (MSB)	6	5	4	3	2	1	0 (LSB)
1 = read 0 = write	address						0

8.1.3 UART interface

8.1.3.1 Connection to a host

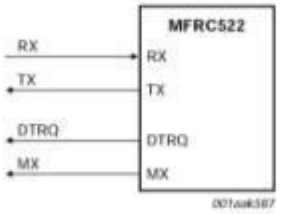


Fig 8. UART connection to microcontrollers

Remark: Signals DTRQ and MX can be disabled by clearing TestPinEnReg register's RS232LineEn bit.

8.2.5 CRC coprocessor

The following CRC coprocessor parameters can be configured:

- The CRC preset value can be either 0000h, 6363h, A671h or FFFFh depending on the ModeReg register's CRCPreset[1:0] bits setting
- The CRC polynomial for the 16-bit CRC is fixed to $x^{16} + x^{12} + x^5 + 1$
- The CRCResultReg register indicates the result of the CRC calculation. This register is split into two 8-bit registers representing the higher and lower bytes.
- The ModeReg register's MSBFirst bit indicates that data will be loaded with the MSB first.

Table 17. CRC coprocessor parameters

Parameter	Value
CRC register length	16-bit CRC
CRC algorithm	algorithm according to ISO/IEC 14443 A and ITU-T
CRC preset value	0000h, 6363h, A671h or FFFFh depending on the setting of the ModeReg register's CRCPreset[1:0] bits

8.3 FIFO buffer

An 8 × 64 bit FIFO buffer is used in the MFRC522. It buffers the input and output data stream between the host and the MFRC522's internal state machine. This makes it possible to manage data streams up to 64 bytes long without the need to take timing constraints into account.

8.3.1 Accessing the FIFO buffer

The FIFO buffer input and output data bus is connected to the FIFODataReg register. Writing to this register stores one byte in the FIFO buffer and increments the internal FIFO buffer write pointer. Reading from this register shows the FIFO buffer contents stored in the FIFO buffer read pointer and decrements the FIFO buffer read pointer. The distance between the write and read pointer can be obtained by reading the FIFOLevelReg register.

When the microcontroller starts a command, the MFRC522 can, while the command is in progress, access the FIFO buffer according to that command. Only one FIFO buffer has been implemented which can be used for input and output. The microcontroller must ensure that there are not any unintentional FIFO buffer accesses.

8.3.2 Controlling the FIFO buffer

The FIFO buffer pointers can be reset by setting FIFOLevelReg register's FlushBuffer bit to logic 1. Consequently, the FIFOLevel[6:0] bits are all set to logic 0 and the ErrorReg register's BufferOvfl bit is cleared. The bytes stored in the FIFO buffer are no longer accessible allowing the FIFO buffer to be filled with another 64 bytes.

8.3.3 FIFO buffer status information

The host can get the following FIFO buffer status information:

- Number of bytes stored in the FIFO buffer: FIFOLevelReg register's FIFOLevel[6:0]
- FIFO buffer almost full warning: Status1Reg register's HiAlert bit

- FIFO buffer almost empty warning: Status1Reg register's LoAlert bit
- FIFO buffer overflow warning: ErrorReg register's BufferOvfl bit. The BufferOvfl bit can only be cleared by setting the FIFOLevelReg register's FlushBuffer bit.

The MFRC522 can generate an interrupt signal when:

- ComIEnReg register's LoAlertIEn bit is set to logic 1. It activates pin IRQ when Status1Reg register's LoAlert bit changes to logic 1.
- ComIEnReg register's HiAlertIEn bit is set to logic 1. It activates pin IRQ when Status1Reg register's HiAlert bit changes to logic 1.

If the maximum number of WaterLevel bytes (as set in the WaterLevelReg register) or less are stored in the FIFO buffer, the HiAlert bit is set to logic 1. It is generated according to Equation 3:

$$HiAlert \quad (64 - FIFOLength) \leq WaterLevel \tag{3}$$

If the number of WaterLevel bytes (as set in the WaterLevelReg register) or less are stored in the FIFO buffer, the LoAlert bit is set to logic 1. It is generated according to Equation 4:

$$LoAlert \quad FIFOLength \leq WaterLevel \tag{4}$$

8.4 Interrupt request system

The MFRC522 indicates certain events by setting the Status1Reg register's IRq bit and, if activated, by pin IRQ. The signal on pin IRQ can be used to interrupt the host using its interrupt handling capabilities. This allows the implementation of efficient host software.

8.4.1 Interrupt sources overview

Table 18 shows the available interrupt bits, the corresponding source and the condition for its activation. The ComIrqReg register's TimerIRq interrupt bit indicates an interrupt set by the timer unit which is set when the timer decrements from 1 to 0.

The ComIrqReg register's TxIRq bit indicates that the transmitter has finished. If the state changes from sending data to transmitting the end of the frame pattern, the transmitter unit automatically sets the interrupt bit. The CRC coprocessor sets the DivIrqReg register's CRCIRq bit after processing all the FIFO buffer data which is indicated by CRCReady bit = 1.

The ComIrqReg register's RxIRq bit indicates an interrupt when the end of the received data is detected. The ComIrqReg register's IdleIRq bit is set if a command finishes and the Command[3:0] value in the CommandReg register changes to idle (see Table 149 on page 70).

The ComIrqReg register's HiAlertIRq bit is set to logic 1 when the Status1Reg register's HiAlert bit is set to logic 1 which means that the FIFO buffer has reached the level indicated by the WaterLevel[5:0] bits.

The ComIrqReg register's LoAlertIRq bit is set to logic 1 when the Status1Reg register's LoAlert bit is set to logic 1 which means that the FIFO buffer has reached the level indicated by the WaterLevel[5:0] bits.

The ComIrqReg register's ErrIRq bit indicates an error detected by the contactless UART during send or receive. This is indicated when any bit is set to logic 1 in register ErrorReg.

Table 18. Interrupt sources

Interrupt flag	Interrupt source	Trigger action
IRq	timer unit	the timer counts from 1 to 0
TxIRq	transmitter	a transmitted data stream ends
CRCIRq	CRC coprocessor	all data from the FIFO buffer has been processed
RxIRq	receiver	a received data stream ends
IdleIRq	ComIrqReg register	command execution finishes
HiAlertIRq	FIFO buffer	the FIFO buffer is almost full
LoAlertIRq	FIFO buffer	the FIFO buffer is almost empty
ErrIRq	contactless UART	an error is detected

8.5 Timer unit

The MFRC522A has a timer unit which the external host can use to manage timing tasks. The timer unit can be used in one of the following timer/counter configurations:

- Timeout counter
- Watchdog counter
- Stop watch
- Programmable one shot
- Periodical trigger

The timer unit can be used to measure the time interval between two events or to indicate that a specific event occurred after a specific time. The timer can be triggered by events explained in the paragraphs below. The timer does not influence any internal events, for example, a time-out during data reception does not automatically influence the reception process. Furthermore, several timer-related bits can be used to generate an interrupt.

The timer has an input clock of 13.56 MHz derived from the 27.12 MHz quartz crystal oscillator. The timer consists of two stages: prescaler and counter.

The prescaler (TPrescaler) is a 12-bit counter. The reload values (TReloadVal_Hi[7:0] and TReloadVal_Lo[7:0]) for TPrescaler can be set between 0 and 4095 in the TModeReg register's TPrescaler_Hi[3:0] bits and TPrescalerReg register's TPrescaler_Lo[7:0] bits.

The reload value for the counter is defined by 16 bits between 0 and 65535 in the TReloadReg register.

The current value of the timer is indicated in the TCounterValReg register.

When the counter reaches 0, an interrupt is automatically generated, indicated by the ComIrqReg register's TimerIRq bit setting. If enabled, this event can be indicated on pin IRQ. The TimerIRq bit can be set and reset by the host. Depending on the configuration, the timer will stop at 0 or restart with the value set in the TReloadReg register.

The timer status is indicated by the Status1Reg register's TRunning bit.

The timer can be started manually using the ControlReg register's TStartNow bit and stopped using the ControlReg register's TStopNow bit.

The timer can also be activated automatically to meet any dedicated protocol requirements by setting the TModeReg register's TAuto bit to logic 1.

The delay time of a timer stage is set by the reload value + 1. The total delay time (t_{d1}) is calculated using Equation 5:

$$t_{d1} = \frac{(TPrescaler \times 2 + 1) \times (TReloadVal + 1)}{13.56\ MHz} \quad (5)$$

An example of calculating total delay time (t_d) is shown in Equation 6, where the TPrescaler value = 4095 and TReloadVal = 65535:

$$39.59\ s = \frac{(4095 \times 2 + 1) \times (65535 + 1)}{13.56\ MHz} \quad (6)$$

Example: To give a delay time of 25 μs requires 339 clock cycles to be counted and a TPrescaler value of 169. This configures the timer to count up to 65535 time-slots for every 25 μs period.

The MFRC522 version 2.0 offers in addition a second prescaler timer. Due to the fact that the prescaler counts down to 0 the prescaler period always count an odd number of clocks (1, 3, 5, ..). This may lead to inaccuracy. The second available prescaler timer implements the possibility to change the prescaler reload value to odd numbers, which results in an even prescaler period. This new prescaler can be enabled only in version 2.0 using the register bit DemodeReg, see Table 72. Within this option, the total delay time (t_{d2}) is calculated using Equation 5:

$$t_{d2} = \frac{(TPrescaler \times 2 + 2) \times (TReloadVal + 1)}{13.56\ MHz} \quad (7)$$

8.6 Power reduction modes

8.6.1 Hard power-down

Hard power-down is enabled when pin NRSTPD is LOW. This turns off all internal current sinks including the oscillator. All digital input buffers are separated from the input pins and clamped internally (except pin NRSTPD). The output pins are frozen at either a HIGH or LOW level.

8.6.2 Soft power-down mode

Soft Power-down mode is entered immediately after the CommandReg register's PowerDown bit is set to logic 1. All internal current sinks are switched off, including the oscillator buffer. However, the digital input buffers are not separated from the input pins and keep their functionality. The digital output pins do not change their state.

During soft power-down, all register values, the FIFO buffer content and the configuration keep their current contents.

After setting the PowerDown bit to logic 0, it takes 1024 clocks until the Soft power-down mode is exited indicated by the PowerDown bit. Setting it to logic 0 does not immediately clear it. It is cleared automatically by the MFRC522 when Soft power-down mode is exited.

Remark: If the internal oscillator is used, you must take into account that it is supplied by pin AVDD and it will take a certain time (t_{osc}) until the oscillator is stable and the clock cycles can be detected by the internal logic. It is recommended for the serial UART, to first send the value 55h to the MFRC522. The oscillator must be stable for further access to the registers. To ensure this, perform a read access to address 0 until the MFRC522 answers to the last read command with the register content of address 0. This indicates that the MFRC522 is ready.

8.6.3 Transmitter power-down mode

The Transmitter Power-down mode switches off the internal antenna drivers thereby, turning off the RF field. Transmitter power-down mode is entered by setting either the TxControlReg register's Tx1RFEn bit or Tx2RFEn bit to logic 0.

8.7 Oscillator circuit

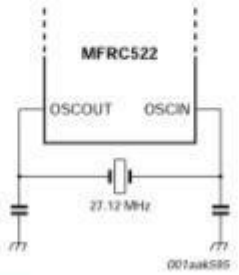


Fig 22. Quartz crystal connection

The clock applied to the MFRC522 provides a time basis for the synchronous system's encoder and decoder. The stability of the clock frequency, therefore, is an important factor for correct operation. To obtain optimum performance, clock jitter must be reduced as much as possible. This is best achieved using the internal oscillator buffer with the recommended circuitry.

If an external clock source is used, the clock signal must be applied to pin OSCIN. In this case, special care must be taken with the clock duty cycle and clock jitter and the clock quality must be verified.

8.8 Reset and oscillator start-up time

8.8.1 Reset timing requirements

The reset signal is filtered by a hysteresis circuit and a spike filter before it enters the digital circuit. The spike filter rejects signals shorter than 10 ns. In order to perform a reset, the signal must be LOW for at least 100 ns.

8.8.2 Oscillator start-up time

If the MFRC522 has been set to a Power-down mode or is powered by a V_{DDX} supply, the start-up time for the MFRC522 depends on the oscillator used and is shown in Figure 23.

The time ($t_{startup}$) is the start-up time of the crystal oscillator circuit. The crystal oscillator start-up time is defined by the crystal.

The time (t_d) is the internal delay time of the MFRC522 when the clock signal is stable before the MFRC522 can be addressed.

The delay time is calculated by:

$$t_d \quad \frac{1024}{27\ \mu} \quad 37.74\ \mu \tag{8}$$

The time (t_{osc}) is the sum of t_d and $t_{startup}$.

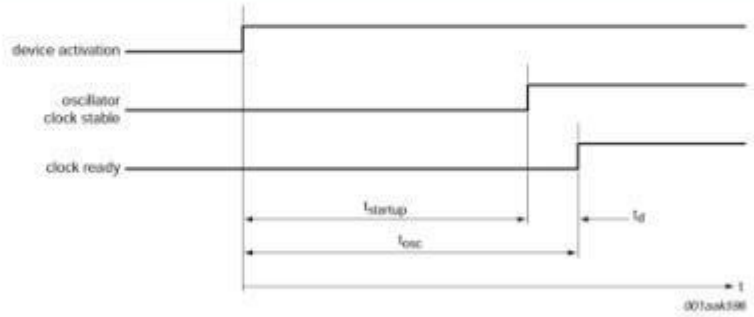


Fig 23. Oscillator start-up time

9. MFRC522 registers

9.1 Register bit behavior

Depending on the functionality of a register, the access conditions to the register can vary. In principle, bits with same behavior are grouped in common registers. The access conditions are described in Table 19.

Table 19. Behavior of register bits and their designation

Abbreviation	Behavior	Description
R/W	read and write	These bits can be written and read by the microcontroller. Since they are used only for control purposes, their content is not influenced by internal state machines, for example the ComIEnReg register can be written and read by the microcontroller. It will also be read by internal state machines but never changed by them.
D	dynamic	These bits can be written and read by the microcontroller. Nevertheless, they can also be written automatically by internal state machines, for example the CommandReg register changes its value automatically after the execution of the command.
R	read only	These register bits hold values which are determined by internal states only, for example the CRCReady bit cannot be written externally but shows internal states.
W	write only	Reading these register bits always returns zero.
reserved	-	These registers are reserved for future use and must not be changed. In case of a write access, it is recommended to always write the value "0".
RFT	-	These register bits are reserved for future use or are for production tests and must not be changed.

9.2 Register overview

Table 20. MFRC522 register overview

Address (hex)	Register name	Function	Refer to
Page 0: Command and status			
00h	Reserved	reserved for future use	Table 21 on page 38
01h	CommandReg	starts and stops command execution	Table 23 on page 38
02h	ComIEnReg	enable and disable interrupt request control bits	Table 25 on page 38
03h	DivIEnReg	enable and disable interrupt request control bits	Table 27 on page 39
04h	ComIrqReg	interrupt request bits	Table 29 on page 39
05h	DivIrqReg	interrupt request bits	Table 31 on page 40
06h	ErrorReg	error bits showing the error status of the last command executed	Table 33 on page 41
07h	Status1Reg	communication status bits	Table 35 on page 42
08h	Status2Reg	receiver and transmitter status bits	Table 37 on page 43
09h	FIFODataReg	input and output of 64 byte FIFO buffer	Table 39 on page 44
0Ah	FIFOLevelReg	number of bytes stored in the FIFO buffer	Table 41 on page 44
0Bh	WaterLevelReg	level for FIFO underflow and overflow warning	Table 43 on page 44
0Ch	ControlReg	miscellaneous control registers	Table 45 on page 45
0Dh	BitFramingReg	adjustments for bit-oriented frames	Table 47 on page 46
0Eh	CollReg	bit position of the first bit-collision detected on the RF interface	Table 49 on page 46
0Fh	Reserved	reserved for future use	Table 51 on page 47
Page 1: Command			
10h	Reserved	reserved for future use	Table 53 on page 47
11h	ModeReg	defines general modes for transmitting and receiving	Table 55 on page 48
12h	TxModeReg	defines transmission data rate and framing	Table 57 on page 48
13h	RxModeReg	defines reception data rate and framing	Table 59 on page 49
14h	TxControlReg	controls the logical behavior of the antenna driver pins TX1 and TX2	Table 61 on page 50
15h	TxASKReg	controls the setting of the transmission modulation	Table 63 on page 51
16h	TxSelReg	selects the internal sources for the antenna driver	Table 65 on page 51
17h	RxSelReg	selects internal receiver settings	Table 67 on page 52
18h	RxThresholdReg	selects thresholds for the bit decoder	Table 69 on page 53
19h	DemodReg	defines demodulator settings	Table 71 on page 53
1Ah	Reserved	reserved for future use	Table 73 on page 54
1Bh	Reserved	reserved for future use	Table 75 on page 54
1Ch	MfTxReg	controls some MIFARE communication transmit parameters	Table 77 on page 55
1Dh	MfRxReg	controls some MIFARE communication receive parameters	Table 79 on page 55
1Eh	Reserved	reserved for future use	Table 81 on page 55
1Fh	SerialSpeedReg	selects the speed of the serial UART interface	Table 83 on page 55
Page 2: Configuration			
20h	Reserved	reserved for future use	Table 85 on page 57

11. Limiting values

Table 150. Limiting values
In accordance with the Absolute Maximum Rating System (IEC 60134).

Symbol	Parameter	Conditions	Min	Max	Unit
V_{DDA}	analog supply voltage		−0.5	+4.0	V
V_{DDD}	digital supply voltage		−0.5	+4.0	V
$V_{DD(PVDD)}$	PVDD supply voltage		−0.5	+4.0	V
$V_{DD(TVDD)}$	TVDD supply voltage		−0.5	+4.0	V
$V_{DD(SVDD)}$	SVDD supply voltage		−0.5	+4.0	V
V_I	input voltage	all input pins except pins MFIN and RX	$V_{SS(PVSS)}$ − 0.5	$V_{DD(PVDD)}$ + 0.5	V
		pin MFIN	$V_{SS(PVSS)}$ − 0.5	$V_{DD(SVDD)}$ + 0.5	V
P_{tot}	total power dissipation	per package; and V_{DDD} in shortcut mode	-	200	mW
T_j	junction temperature		-	100	°C
V_{ESD}	electrostatic discharge voltage	HBM; 1500 Ω, 100 pF; JESD22-A114-B	-	2000	V
		MM; 0.75 μH, 200 pF; JESD22-A114-A	-	200	V
		Charged device model; JESD22-C101-A			
		on all pins	-	200	V
		on all pins except SVDD in TFBGA64 package	-	500	V

12. Recommended operating conditions

Table 151. Operating conditions

Symbol	Parameter	Conditions		Min	Typ	Max	Unit
V_{DDA}	analog supply voltage	$V_{DD(PVDD)} \le V_{DDA} = V_{DDD} = V_{DD(TVDD)}$; $V_{SSA} = V_{SSD} = V_{SS(PVSS)} = V_{SS(TVSS)} = 0$ V	[1][2]	2.5	3.3	3.6	V
V_{DDD}	digital supply voltage	$V_{DD(PVDD)} \le V_{DDA} = V_{DDD} = V_{DD(TVDD)}$; $V_{SSA} = V_{SSD} = V_{SS(PVSS)} = V_{SS(TVSS)} = 0$ V	[1][2]	2.5	3.3	3.6	V
$V_{DD(TVDD)}$	TVDD supply voltage	$V_{DD(PVDD)} \le V_{DDA} = V_{DDD} = V_{DD(TVDD)}$; $V_{SSA} = V_{SSD} = V_{SS(PVSS)} = V_{SS(TVSS)} = 0$ V	[1][2]	2.5	3.3	3.6	V
$V_{DD(PVDD)}$	PVDD supply voltage	$V_{DD(PVDD)} \le V_{DDA} = V_{DDD} = V_{DD(TVDD)}$; $V_{SSA} = V_{SSD} = V_{SS(PVSS)} = V_{SS(TVSS)} = 0$ V	[3]	1.6	1.8	3.6	V
$V_{DD(SVDD)}$	SVDD supply voltage	$V_{SSA} = V_{SSD} = V_{SS(PVSS)} = V_{SS(TVSS)} = 0$ V		1.6	-	3.6	V
T_{amb}	ambient temperature	HVQFN32		−25	-	+85	°C

[1] Supply voltages below 3 V reduce the performance (the achievable operating distance).

[2] V_{DDA}, V_{DDD} and $V_{DD(TVDD)}$ must always be the same voltage.

[3] $V_{DD(PVDD)}$ must always be the same or lower voltage than V_{DDD}.

13. Thermal characteristics

Table 152. Thermal characteristics

Symbol	Parameter	Conditions	Package	Typ	Unit
$R_{th(j-a)}$	thermal resistance from junction to ambient	in still air with exposed pin soldered on a 4 layer JEDEC PCB	HVQFN32	40	K/W

14. Characteristics

Table 153. Characteristics

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
Input characteristics						
Pins EA, I2C and NRSTPD						
I_{LI}	input leakage current		−1	-	+1	µA
V_{IH}	HIGH-level input voltage		$0.7V_{DD(PVDD)}$	-	-	V
V_{IL}	LOW-level input voltage		-	-	$0.3V_{DD(PVDD)}$	V
Pin MFIN						
I_{LI}	input leakage current		−1	-	+1	µA
V_{IH}	HIGH-level input voltage		$0.7V_{DD(SVDD)}$	-	-	V
V_{IL}	LOW-level input voltage		-	-	$0.3V_{DD(SVDD)}$	V
Pin SDA						
I_{LI}	input leakage current		−1	-	+1	µA
V_{IH}	HIGH-level input voltage		$0.7V_{DD(PVDD)}$	-	-	V
V_{IL}	LOW-level input voltage		-	-	$0.3V_{DD(PVDD)}$	V
Pin RX[1]						
V_i	input voltage		−1	-	V_{DDA} +1	V
C_i	input capacitance	V_{DDA} = 3 V; receiver active; $V_{RX(p-p)}$ = 1 V; 1.5 V (DC) offset	-	10	-	pF
R_i	input resistance	V_{DDA} = 3 V; receiver active; $V_{RX(p-p)}$ = 1 V; 1.5 V (DC) offset	-	350	-	Ω
Input voltage range; see Figure 24						
$V_{i(p-p)(min)}$	minimum peak-to-peak input voltage	Manchester encoded; V_{DDA} = 3 V	-	100	-	mV
$V_{i(p-p)(max)}$	maximum peak-to-peak input voltage	Manchester encoded; V_{DDA} = 3 V	-	4	-	V
Input sensitivity; see Figure 24						
V_{mod}	modulation voltage	minimum Manchester encoded; V_{DDA} = 3 V; RxGain[2:0] = 111b (48 dB)	-	5	-	mV
Pin OSCIN						
I_{LI}	input leakage current		−1	-	+1	µA
V_{IH}	HIGH-level input voltage		$0.7V_{DDA}$	-	-	V
V_{IL}	LOW-level input voltage		-	-	$0.3V_{DDA}$	V

Table 153. Characteristics ...continued

Symbol	Parameter	Conditions		Min	Typ	Max	Unit
C_I	input capacitance	V_{DDA} = 2.8 V; DC = 0.65 V; AC = 1 V (p-p)		-	2	-	pF
Input/output characteristics							
pins D1, D2, D3, D4, D5, D6 and D7							
I_{LI}	input leakage current			−1	-	+1	µA
V_{IH}	HIGH-level input voltage			$0.7V_{DD(PVDD)}$	-	-	V
V_{IL}	LOW-level input voltage			-	-	$0.3V_{DD(PVDD)}$	V
V_{OH}	HIGH-level output voltage	$V_{DD(PVDD)}$ = 3 V; I_O = 4 mA		$V_{DD(PVDD)}$ − 0.4	-	$V_{DD(PVDD)}$	V
V_{OL}	LOW-level output voltage	$V_{DD(PVDD)}$ = 3 V; I_O = 4 mA		$V_{SS(PVSS)}$	-	$V_{SS(PVSS)}$ + 0.4	V
I_{OH}	HIGH-level output current	$V_{DD(PVDD)}$ = 3 V		-	-	4	mA
I_{OL}	LOW-level output current	$V_{DD(PVDD)}$ = 3 V		-	-	4	mA
Output characteristics							
Pin MFOUT							
V_{OH}	HIGH-level output voltage	$V_{DD(SVDD)}$ = 3 V; I_O = 4 mA		$V_{DD(SVDD)}$ − 0.4	-	$V_{DD(SVDD)}$	V
V_{OL}	LOW-level output voltage	$V_{DD(SVDD)}$ = 3 V; I_O = 4 mA		$V_{SS(PVSS)}$	-	$V_{SS(PVSS)}$ + 0.4	V
I_{OL}	LOW-level output current	$V_{DD(SVDD)}$ = 3 V		-	-	4	mA
I_{OH}	HIGH-level output current	$V_{DD(SVDD)}$ = 3 V		-	-	4	mA
Pin IRQ							
V_{OH}	HIGH-level output voltage	$V_{DD(PVDD)}$ = 3 V; I_O = 4 mA		$V_{DD(PVDD)}$ − 0.4	-	$V_{DD(PVDD)}$	V
V_{OL}	LOW-level output voltage	$V_{DD(PVDD)}$ = 3 V; I_O = 4 mA		$V_{SS(PVSS)}$	-	$V_{SS(PVSS)}$ + 0.4	V
I_{OL}	LOW-level output current	$V_{DD(PVDD)}$ = 3 V		-	-	4	mA
I_{OH}	HIGH-level output current	$V_{DD(PVDD)}$ = 3 V		-	-	4	mA
Pins AUX1 and AUX2							
V_{OH}	HIGH-level output voltage	V_{DDD} = 3 V; I_O = 4 mA		V_{DDD} − 0.4	-	V_{DDD}	V
V_{OL}	LOW-level output voltage	V_{DDD} = 3 V; I_O = 4 mA		$V_{SS(PVSS)}$	-	$V_{SS(PVSS)}$ + 0.4	V
I_{OL}	LOW-level output current	V_{DDD} = 3 V		-	-	4	mA
I_{OH}	HIGH-level output current	V_{DDD} = 3 V		-	-	4	mA
Pins TX1 and TX2							

Table 153. Characteristics ...continued

Symbol	Parameter	Conditions		Min	Typ	Max	Unit
V_{OH}	HIGH-level output voltage	$V_{DD(TVDD)}$ = 3 V; $I_{DD(TVDD)}$ = 32 mA; CWGsP[5:0] = 3Fh		$V_{DD(TVDD)}$ − 0.15	-	-	V
		$V_{DD(TVDD)}$ = 3 V; $I_{DD(TVDD)}$ = 80 mA; CWGsP[5:0] = 3Fh		$V_{DD(TVDD)}$ − 0.4	-	-	V
		$V_{DD(TVDD)}$ = 2.5 V; $I_{DD(TVDD)}$ = 32 mA; CWGsP[5:0] = 3Fh		$V_{DD(TVDD)}$ − 0.24	-	-	V
		$V_{DD(TVDD)}$ = 2.5 V; $I_{DD(TVDD)}$ = 80 mA; CWGsP[5:0] = 3Fh		$V_{DD(TVDD)}$ − 0.64	-	-	V
V_{OL}	LOW-level output voltage	$V_{DD(TVDD)}$ = 3 V; $I_{DD(TVDD)}$ = 32 mA; CWGsP[5:0] = 0Fh		-	-	0.15	V
		$V_{DD(TVDD)}$ = 3 V; $I_{DD(TVDD)}$ = 80 mA; CWGsP[5:0] = 0Fh		-	-	0.4	V
		$V_{DD(TVDD)}$ = 2.5 V; $I_{DD(TVDD)}$ = 32 mA; CWGsP[5:0] = 0Fh		-	-	0.24	V
		$V_{DD(TVDD)}$ = 2.5 V; $I_{DD(TVDD)}$ = 80 mA; CWGsP[5:0] = 0Fh		-	-	0.64	V
Current consumption							
I_{pd}	power-down current	V_{DDA} = V_{DDD} = $V_{DD(TVDD)}$ = $V_{DD(PVDD)}$ = 3 V					
		hard power-down; pin NRSTPD set LOW	[2]	-	-	5	µA
		soft power-down; RF level detector on	[2]	-	-	10	µA
I_{DDD}	digital supply current	pin DVDD; V_{DDD} = 3 V		-	6.5	9	mA
I_{DDA}	analog supply current	pin AVDD; V_{DDA} = 3 V; CommandReg register's bit RcvOff = 0		-	7	10	mA
		pin AVDD; receiver switched off; V_{DDA} = 3 V; CommandReg register's bit RcvOff = 1		-	3	5	mA
$I_{DD(PVDD)}$	PVDD supply current	pin PVDD	[3]	-	-	40	mA
$I_{DD(TVDD)}$	TVDD supply current	pin TVDD; continuous wave	[4][5][6]	-	60	100	mA
$I_{DD(SVDD)}$	SVDD supply current	pin SVDD	[7]	-	-	4	mA
Clock frequency							
f_{clk}	clock frequency			-	27.12	-	MHz
δ_{clk}	clock duty cycle			40	50	60	%
t_{jit}	jitter time	RMS		-	-	10	ps
Crystal oscillator							

Table 153. Characteristics ...continued

Symbol	Parameter	Conditions		Min	Typ	Max	Unit
V_{OH}	HIGH-level output voltage	pin OSCOUT		-	1.1	-	V
V_{OL}	LOW-level output voltage	pin OSCOUT		-	0.2	-	V
C_i	input capacitance	pin OSCOUT		-	2	-	pF
		pin OSCIN		-	2	-	pF
Typical input requirements							
f_{xtal}	crystal frequency			-	27.12	-	MHz
ESR	equivalent series resistance			-	-	100	Ω
C_L	load capacitance			-	10	-	pF
P_{xtal}	crystal power dissipation			-	50	100	mW

[1] The voltage on pin RX is clamped by internal diodes to pins AVSS and AVDD.

[2] I_{pd} is the total current for all supplies.

[3] $I_{DD(PVDD)}$ depends on the overall load at the digital pins.

[4] $I_{DD(TVDD)}$ depends on $V_{DD(TVDD)}$ and the external circuit connected to pins TX1 and TX2.

[5] During typical circuit operation, the overall current is below 100 mA.

[6] Typical value using a complementary driver configuration and an antenna matched to 40 Ω between pins TX1 and TX2 at 13.56 MHz.

[7] $I_{DD(SVDD)}$ depends on the load at pin MFOUT.

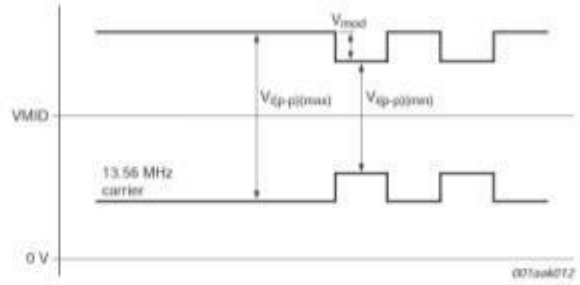


Fig 24. Pin RX input voltage range

14.1 Timing characteristics

Table 154. SPI timing characteristics

Symbol	Parameter	Conditions		Min	Typ	Max	Unit
t_{WL}	pulse width LOW	line SCK		50	-	-	ns
t_{WH}	pulse width HIGH	line SCK		50	-	-	ns
$t_{h(SCKH-D)}$	SCK HIGH to data input hold time	SCK to changing MOSI		25	-	-	ns

Table 154. SPI timing characteristics ...continued

Symbol	Parameter	Conditions		Min	Typ	Max	Unit
$t_{su(D-SCKH)}$	data input to SCK HIGH set-up time	changing MOSI to SCK		25	-	-	ns
$t_{h(SCKL-O)}$	SCK LOW to data output hold time	SCK to changing MISO		-	-	25	ns
$t_{(SCKL-NSSH)}$	SCK LOW to NSS HIGH time			0	-	-	ns
t_{NHNL}	NSS high before communication			50	-	-	ns

Table 155. I²C-bus timing in Fast mode

Symbol	Parameter	Conditions	Fast mode		High-speed mode		Unit
			Min	Max	Min	Max	
f_{SCL}	SCL clock frequency		0	400	0	3400	kHz
$t_{HD;STA}$	hold time (repeated) START condition	after this period, the first clock pulse is generated	600	-	160	-	ns
$t_{SU;STA}$	set-up time for a repeated START condition		600	-	160	-	ns
$t_{SU;STO}$	set-up time for STOP condition		600	-	160	-	ns
t_{LOW}	LOW period of the SCL clock		1300	-	160	-	ns
t_{HIGH}	HIGH period of the SCL clock		600	-	60	-	ns
$t_{HD;DAT}$	data hold time		0	900	0	70	ns
$t_{SU;DAT}$	data set-up time		100	-	10	-	ns
t_r	rise time	SCL signal	20	300	10	40	ns
t_f	fall time	SCL signal	20	300	10	40	ns
t_r	rise time	SDA and SCL signals	20	300	10	80	ns
t_f	fall time	SDA and SCL signals	20	300	10	80	ns
t_{BUF}	bus free time between a STOP and START condition		1.3	-	1.3	-	µs

15. Application information

A typical application diagram using a complementary antenna connection to the MFRC522 is shown in Figure 27.

The antenna tuning and RF part matching is described in the application note Ref. 1 and Ref. 2.

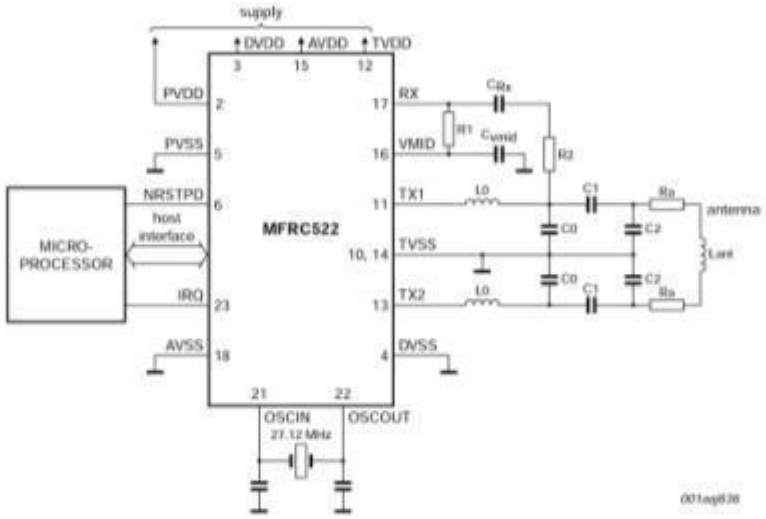
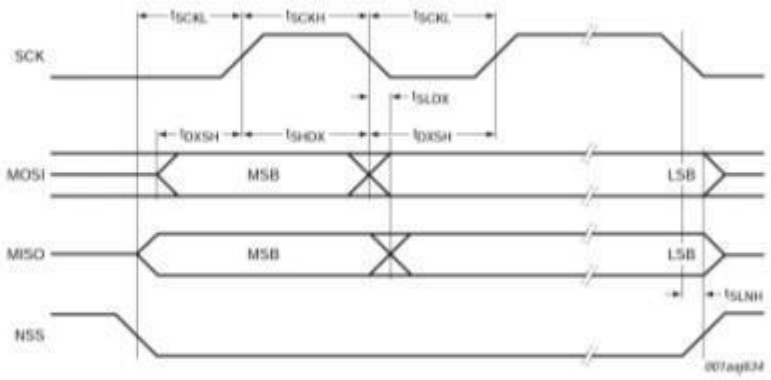


Fig 27. Typical application diagram



Remark: The signal NSS must be LOW to be able to send several bytes in one data stream. To send more than one data stream NSS must be set HIGH between the data streams.

Fig 25. Timing diagram for SPI

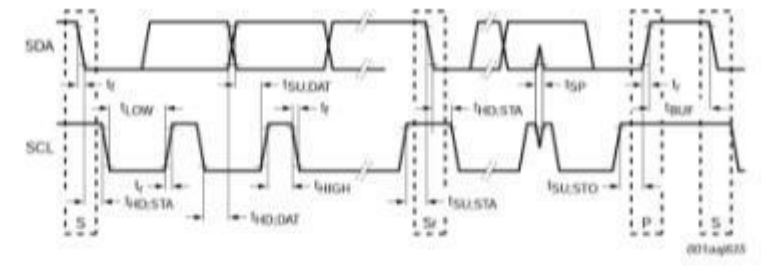


Fig 26. Timing for Fast and Standard mode devices on the I²C-bus

Hoja de datos DS3231

DS3231

Extremely Accurate I²C-Integrated RTC/TCXO/Crystal

General Description

The DS3231 is a low-cost, extremely accurate I²C real-time clock (RTC) with an integrated temperature-compensated crystal oscillator (TCXO) and crystal. The device incorporates a battery input, and maintains accurate timekeeping when main power to the device is interrupted. The integration of the crystal resonator enhances the long-term accuracy of the device as well as reduces the piece-part count in a manufacturing line. The DS3231 is available in commercial and industrial temperature ranges, and is offered in a 16-pin, 300-mil SO package.

The RTC maintains seconds, minutes, hours, day, date, month, and year information. The date at the end of the month is automatically adjusted for months with fewer than 31 days, including corrections for leap year. The clock operates in either the 24-hour or 12-hour format with an AM/PM indicator. Two programmable time-of-day alarms and a programmable square-wave output are provided. Address and data are transferred serially through an I²C bidirectional bus.

A precision temperature-compensated voltage reference and comparator circuit monitors the status of V_{CC} to detect power failures, to provide a reset output, and to automatically switch to the backup supply when necessary. Additionally, the $\overline{RST}$ pin is monitored as a pushbutton input for generating a µP reset.

Benefits and Features

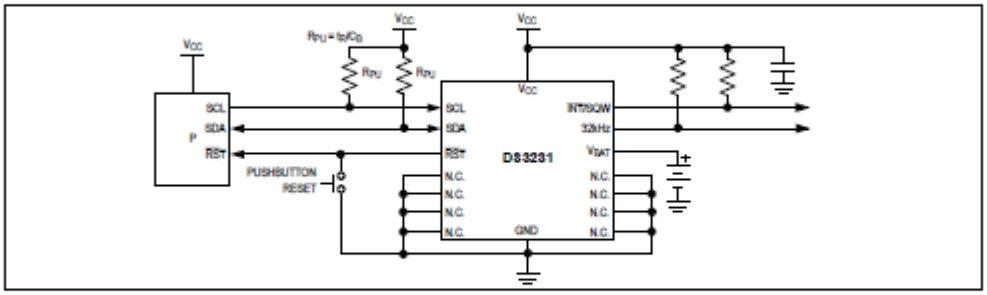
- Highly Accurate RTC Completely Manages All Timekeeping Functions
  - Real-Time Clock Counts Seconds, Minutes, Hours, Date of the Month, Month, Day of the Week, and Year, with Leap-Year Compensation Valid Up to 2100
  - Accuracy ±2ppm from 0°C to +40°C
  - Accuracy ±3.5ppm from -40°C to +85°C
  - Digital Temp Sensor Output: ±3°C Accuracy
  - Register for Aging Trim
  - $\overline{RST}$ Output/Pushbutton Reset Debounce Input
  - Two Time-of-Day Alarms
  - Programmable Square-Wave Output Signal
- Simple Serial Interface Connects to Most Microcontrollers
  - Fast (400kHz) I²C Interface
- Battery-Backup Input for Continuous Timekeeping
  - Low Power Operation Extends Battery-Backup Run Time
  - 3.3V Operation
- Operating Temperature Ranges: Commercial (0°C to +70°C) and Industrial (-40°C to +85°C)
- Underwriters Laboratories® (UL) Recognized

Applications

- Servers
- Telematics
- Utility Power Meters
- GPS

Ordering Information and Pin Configuration appear at end of data sheet.

Typical Operating Circuit



Underwriters Laboratories is a registered certification mark of Underwriters Laboratories Inc.

maxim integrated

19-6170; Rev 10; 3/16

Absolute Maximum Ratings

Voltage Range on Any Pin Relative to Ground.....-0.3V to +6.0V
Junction-to-Ambient Thermal Resistance (θ_{JA}) (Note 1)73°C/W
Junction-to-Case Thermal Resistance (θ_{JC}) (Note 1).....23°C/W
Operating Temperature Range
DS3231S .....0°C to +70°C
DS3231SN..... -40°C to +85°C
Junction Temperature.....+125°C
Storage Temperature Range ..... -40°C to +85°C
Lead Temperature (soldering, 10s) .....+260°C
Soldering Temperature (reflow, 2 times max) .....+260°C
(see the *Handling, PCB Layout, and Assembly* section)

Note 1: Package thermal resistances were obtained using the method described in JEDEC specification JESD51-7, using a four-layer board. For detailed information on package thermal considerations, refer to www.maximintegrated.com/thermal-tutorial.

Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated in the operational sections of the specifications is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

Recommended Operating Conditions

(T_A = T_{MIN} to T_{MAX}, unless otherwise noted.) (Notes 2, 3)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
Supply Voltage	V_{CC}		2.3	3.3	5.5	V
	V_{BAT}		2.3	3.0	5.5	V
Logic 1 Input SDA, SCL	V_{IH}		0.7 x V_{CC}		V_{CC} + 0.3	V
Logic 0 Input SDA, SCL	V_{IL}		-0.3		0.3 x V_{CC}	V

Electrical Characteristics

(V_{CC} = 2.3V to 5.5V, V_{CC} = Active Supply (see Table 1), T_A = T_{MIN} to T_{MAX}, unless otherwise noted.) (Typical values are at V_{CC} = 3.3V, V_{BAT} = 3.0V, and T_A = +25°C, unless otherwise noted.) (Notes 2, 3)

PARAMETER	SYMBOL	CONDITIONS		MIN	TYP	MAX	UNITS
Active Supply Current	I_{CCA}	(Notes 4, 5)	V_{CC} = 3.63V			200	µA
			V_{CC} = 5.5V			300	
Standby Supply Current	I_{CCS}	I²C bus inactive, 32kHz output on, SQW output off (Note 5)	V_{CC} = 3.63V			110	µA
			V_{CC} = 5.5V			170	
Temperature Conversion Current	$I_{CCSCONV}$	I²C bus inactive, 32kHz output on, SQW output off	V_{CC} = 3.63V			575	µA
			V_{CC} = 5.5V			650	
Power-Fail Voltage	V_{PF}			2.45	2.575	2.70	V
Logic 0 Output, 32kHz, $\overline{INT}$/SQW, SDA	V_{OL}	I_{OL} = 3mA				0.4	V
Logic 0 Output, $\overline{RST}$	V_{OL}	I_{OL} = 1mA				0.4	V
Output Leakage Current 32kHz, $\overline{INT}$/SQW, SDA	I_{LO}	Output high impedance		-1	0	+1	µA
Input Leakage SCL	I_{LI}			-1		+1	µA
$\overline{RST}$ Pin I/O Leakage	I_{OL}	$\overline{RST}$ high impedance (Note 6)		-200		+10	µA
V_{BAT} Leakage Current (V_{CC} Active)	I_{BATLKG}				25	100	nA

Electrical Characteristics (continued)

(V_{CC} = 2.3V to 5.5V, V_{CC} = Active Supply (see Table 1), T_A = T_{MIN} to T_{MAX}, unless otherwise noted.) (Typical values are at V_{CC} = 3.3V, V_{BAT} = 3.0V, and T_A = +25°C, unless otherwise noted.) (Notes 2, 3)

PARAMETER	SYMBOL	CONDITIONS		MIN	TYP	MAX	UNITS
Output Frequency	f_{OUT}	V_{CC} = 3.3V or V_{BAT} = 3.3V			32.768		kHz
Frequency Stability vs. Temperature (Commercial)	$\Delta f/f_{OUT}$	V_{CC} = 3.3V or V_{BAT} = 3.3V, aging offset = 00h	0°C to +40°C			±2	ppm
			>40°C to +70°C			±3.5	
Frequency Stability vs. Temperature (Industrial)	$\Delta f/f_{OUT}$	V_{CC} = 3.3V or V_{BAT} = 3.3V, aging offset = 00h	-40°C to <0°C			±3.5	ppm
			0°C to +40°C			±2	
			>40°C to +85°C			±3.5	
Frequency Stability vs. Voltage	$\Delta f/V$				1		ppm/V
Trim Register Frequency Sensitivity per LSB	Δf/LSB	Specified at:	-40°C		0.7		ppm
			+25°C		0.1		
			+70°C		0.4		
			+85°C		0.8		
Temperature Accuracy	Temp	V_{CC} = 3.3V or V_{BAT} = 3.3V		-3		+3	°C
Crystal Aging	$\Delta f/f_O$	After reflow, not production tested	First year		±1.0		ppm
			0–10 years		±5.0		

Electrical Characteristics

(V_{CC} = 0V, V_{BAT} = 2.3V to 5.5V, T_A = T_{MIN} to T_{MAX}, unless otherwise noted.) (Note 2)

PARAMETER	SYMBOL	CONDITIONS		MIN	TYP	MAX	UNITS
Active Battery Current	I_{BATA}	$\overline{EOSC}$ = 0, BBSQW = 0, SCL = 400kHz (Note 5)	V_{BAT} = 3.63V			70	µA
			V_{BAT} = 5.5V			150	
Timekeeping Battery Current	I_{BATT}	$\overline{EOSC}$ = 0, BBSQW = 0, EN32kHz = 1, SCL = SDA = 0V or SCL = SDA = V_{BAT} (Note 5)	V_{BAT} = 3.63V		0.84	3.0	µA
			V_{BAT} = 5.5V		1.0	3.5	
Temperature Conversion Current	I_{BATTC}	$\overline{EOSC}$ = 0, BBSQW = 0, SCL = SDA = 0V or SCL = SDA = V_{BAT}	V_{BAT} = 3.63V			575	µA
			V_{BAT} = 5.5V			650	
Data-Retention Current	I_{BATTDR}	$\overline{EOSC}$ = 1, SCL = SDA = 0V, +25°C				100	nA

AC Electrical Characteristics

(V_{CC} = $V_{CC(MIN)}$ to $V_{CC(MAX)}$ or V_{BAT} = $V_{BAT(MIN)}$ to $V_{BAT(MAX)}$, V_{BAT} > V_{CC}, T_A = T_{MIN} to T_{MAX}, unless otherwise noted.) (Note 2)

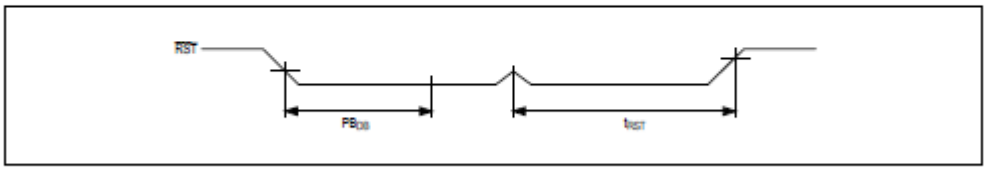
PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
SCL Clock Frequency	f_{SCL}	Fast mode	100		400	kHz
		Standard mode	0		100	
Bus Free Time Between STOP and START Conditions	t_{BUF}	Fast mode	1.3			µs
		Standard mode	4.7			
Hold Time (Repeated) START Condition (Note 7)	$t_{HD:STA}$	Fast mode	0.6			µs
		Standard mode	4.0			
Low Period of SCL Clock	t_{LOW}	Fast mode	1.3			µs
		Standard mode	4.7			
High Period of SCL Clock	t_{HIGH}	Fast mode	0.6			µs
		Standard mode	4.0			
Data Hold Time (Notes 8, 9)	$t_{HD:DAT}$	Fast mode	0		0.9	µs
		Standard mode	0		0.9	
Data Setup Time (Note 10)	$t_{SU:DAT}$	Fast mode	100			ns
		Standard mode	250			
START Setup Time	$t_{SU:STA}$	Fast mode	0.6			µs
		Standard mode	4.7			
Rise Time of Both SDA and SCL Signals (Note 11)	t_R	Fast mode	20 + 0.1C_B		300	ns
		Standard mode			1000	
Fall Time of Both SDA and SCL Signals (Note 11)	t_F	Fast mode	20 + 0.1C_B		300	ns
		Standard mode			300	
Setup Time for STOP Condition	$t_{SU:STO}$	Fast mode	0.6			µs
		Standard mode	4.7			
Capacitive Load for Each Bus Line	C_B	(Note 11)			400	pF
Capacitance for SDA, SCL	$C_{I/O}$			10		pF
Pulse Width of Spikes That Must Be Suppressed by the Input Filter	t_{SP}			30		ns
Pushbutton Debounce	PB_{DB}			250		ms
Reset Active Time	t_{RST}			250		ms
Oscillator Stop Flag (OSF) Delay	t_{OSF}	(Note 12)		100		ms
Temperature Conversion Time	t_{CONV}			125	200	ms

Power-Switch Characteristics

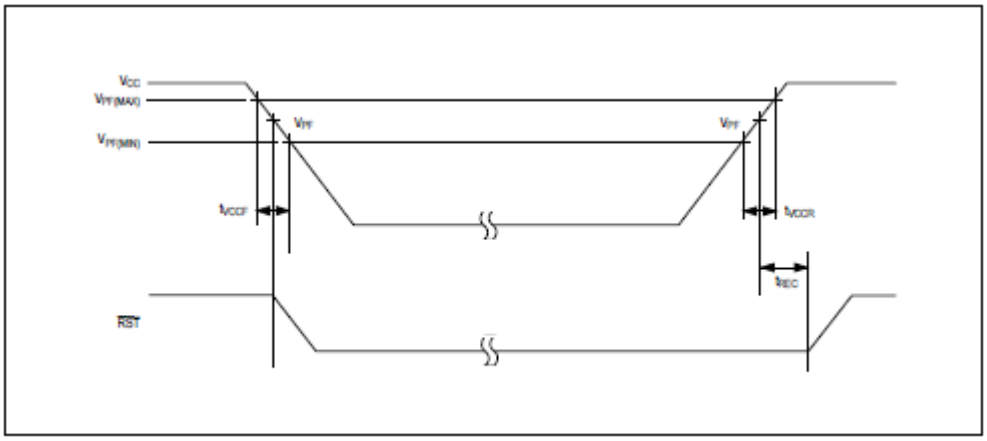
(T_A = T_{MIN} to T_{MAX})

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
V_{CC} Fall Time; $V_{PF(MAX)}$ to $V_{PF(MIN)}$	t_{VCCF}		300			µs
V_{CC} Rise Time; $V_{PF(MIN)}$ to $V_{PF(MAX)}$	t_{VCCR}		0			µs
Recovery at Power-Up	t_{REC}	(Note 13)		250	300	ms

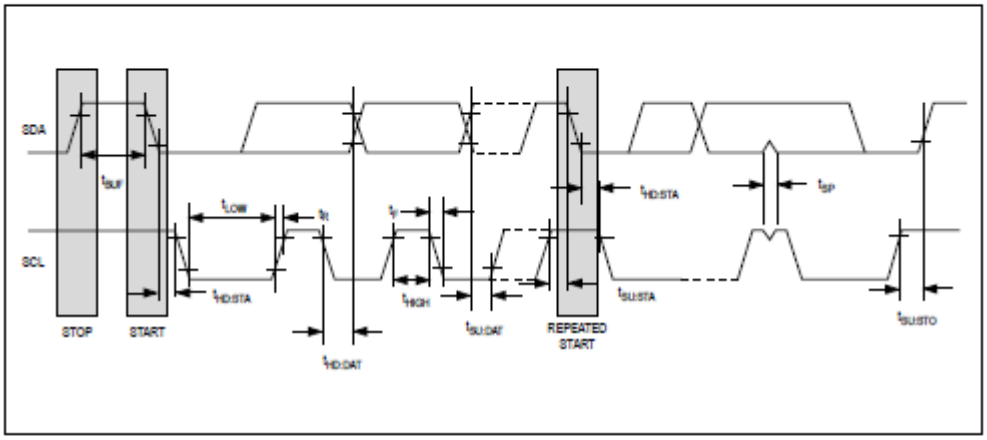
Pushbutton Reset Timing



Power-Switch Timing



Data Transfer on I²C Serial Bus



WARNING: Negative undershoots below -0.3V while the part is in battery-backed mode may cause loss of data.

Note 2: Limits at -40°C are guaranteed by design and not production tested.

Note 3: All voltages are referenced to ground.

Note 4: I_{CCA}—SCL clocking at max frequency = 400kHz.

Note 5: Current is the averaged input current, which includes the temperature conversion current.

Note 6: The $\overline{RST}$ pin has an internal 50kΩ (nominal) pullup resistor to V_{CC}.

Note 7: After this period, the first clock pulse is generated.

Note 8: A device must internally provide a hold time of at least 300ns for the SDA signal (referred to the $V_{IH(MIN)}$ of the SCL signal) to bridge the undefined region of the falling edge of SCL.

Note 9: The maximum $t_{HD:DAT}$ needs only to be met if the device does not stretch the low period (t_{LOW}) of the SCL signal.

Note 10: A fast-mode device can be used in a standard-mode system, but the requirement $t_{SU:DAT} \geq 250ns$ must then be met. This is automatically the case if the device does not stretch the low period of the SCL signal. If such a device does stretch the low period of the SCL signal, it must output the next data bit to the SDA line $t_{R(MAX)} + t_{SU:DAT} = 1000 + 250 = 1250ns$ before the SCL line is released.

Note 11: C_B—total capacitance of one bus line in pF.

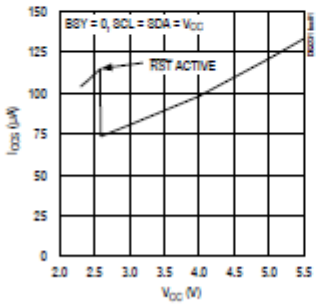
Note 12: The parameter t_{OSF} is the period of time the oscillator must be stopped for the OSF flag to be set over the voltage range of $0.0V \leq V_{CC} \leq V_{CC(MAX)}$ and $2.3V \leq V_{BAT} \leq 3.4V$.

Note 13: This delay applies only if the oscillator is enabled and running. If the $\overline{EOSC}$ bit is a 1, t_{REC} is bypassed and $\overline{RST}$ immediately goes high. The state of $\overline{RST}$ does not affect the I²C interface, RTC, or TCXO.

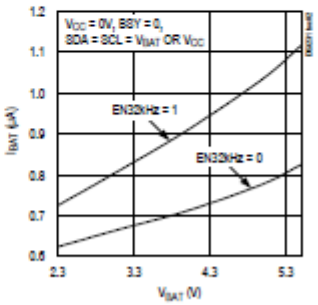
Typical Operating Characteristics

(V_{CC} = +3.3V, T_A = +25°C, unless otherwise noted.)

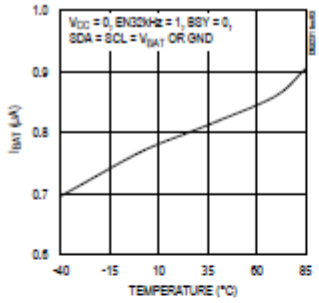
STANDBY SUPPLY CURRENT vs. SUPPLY VOLTAGE



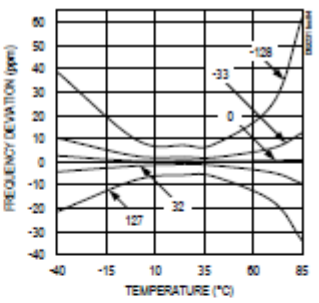
SUPPLY CURRENT vs. SUPPLY VOLTAGE



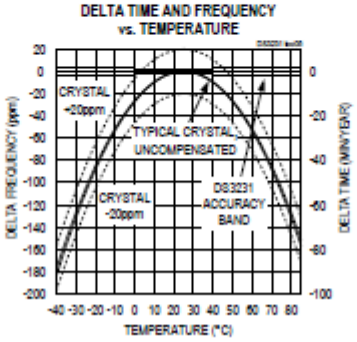
SUPPLY CURRENT vs. TEMPERATURE



FREQUENCY DEVIATION vs. TEMPERATURE vs. AGING VALUE

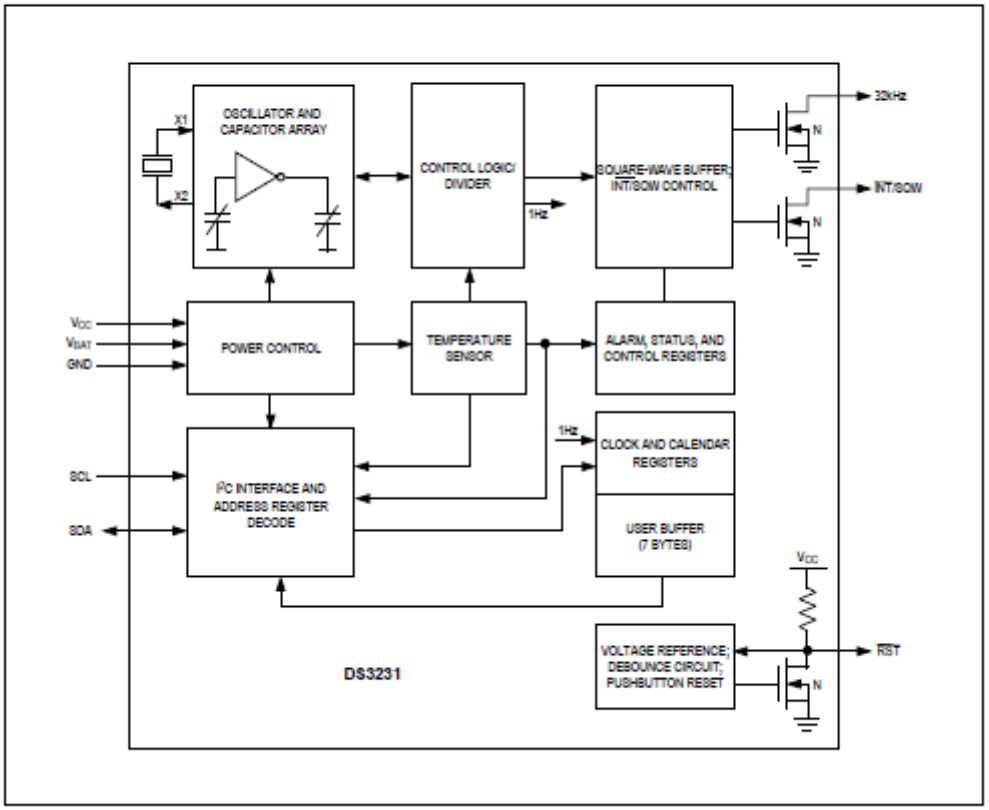


DELTA TIME AND FREQUENCY vs. TEMPERATURE



DS3231 Extremely Accurate I²C-Integrated RTC/TCXO/Crystal

Block Diagram



Pin Description

PIN	NAME	FUNCTION
1	32kHz	32kHz Output. This open-drain pin requires an external pullup resistor. When enabled, the output operates on either power supply. It may be left open if not used.
2	V_{CC}	DC Power Pin for Primary Power Supply. This pin should be decoupled using a 0.1µF to 1.0µF capacitor. If not used, connect to ground.
3	INT/SQW	Active-Low Interrupt or Square-Wave Output. This open-drain pin requires an external pullup resistor connected to a supply at 5.5V or less. This multifunction pin is determined by the state of the INTCN bit in the Control Register (0Eh). When INTCN is set to logic 0, this pin outputs a square wave and its frequency is determined by RS2 and RS1 bits. When INTCN is set to logic 1, then a match between the timekeeping registers and either of the alarm registers activates the INT/SQW pin (if the alarm is enabled). Because the INTCN bit is set to logic 1 when power is first applied, the pin defaults to an interrupt output with alarms disabled. The pullup voltage can be up to 5.5V, regardless of the voltage on V_{CC}. If not used, this pin can be left unconnected.
4	RST	Active-Low Reset. This pin is an open-drain input/output. It indicates the status of V_{CC} relative to the V_{PF} specification. As V_{CC} falls below V_{PF}, the RST pin is driven low. When V_{CC} exceeds V_{PF}, for t_{RST}, the RST pin is pulled high by the internal pullup resistor. The active-low, open-drain output is combined with a debounced pushbutton input function. This pin can be activated by a pushbutton reset request. It has an internal 50kΩ nominal value pullup resistor to V_{CC}. No external pullup resistors should be connected. If the oscillator is disabled, t_{REC} is bypassed and RST immediately goes high.
5–12	N.C.	No Connection. Must be connected to ground.
13	GND	Ground
14	V_{BAT}	Backup Power-Supply Input. When using the device with the V_{BAT} input as the primary power source, this pin should be decoupled using a 0.1µF to 1.0µF low-leakage capacitor. When using the device with the V_{BAT} input as the backup power source, the capacitor is not required. If V_{BAT} is not used, connect to ground. The device is UL recognized to ensure against reverse charging when used with a primary lithium battery. Go to www.maximintegrated.com/qa/info/ul.
15	SDA	Serial Data Input/Output. This pin is the data input/output for the I²C serial interface. This open-drain pin requires an external pullup resistor. The pullup voltage can be up to 5.5V, regardless of the voltage on V_{CC}.
16	SCL	Serial Clock Input. This pin is the clock input for the I²C serial interface and is used to synchronize data movement on the serial interface. Up to 5.5V can be used for this pin, regardless of the voltage on V_{CC}.

Detailed Description

The DS3231 is a serial RTC driven by a temperature-compensated 32kHz crystal oscillator. The TCXO provides a stable and accurate reference clock, and maintains the RTC to within ±2 minutes per year accuracy from -40°C to +85°C. The TCXO frequency output is available at the 32kHz pin. The RTC is a low-power clock/calendar with two programmable time-of-day alarms and a programmable square-wave output. The INT/SQW provides either an interrupt signal due to alarm conditions or a square-wave output. The clock/calendar provides seconds, minutes, hours, day, date, month, and year information. The date at the end of the month is automatically adjusted for months with fewer than 31 days, including corrections for leap year. The clock operates in either the 24-hour or 12-hour format with an AM/PM indicator. The internal registers are accessible though an I²C bus interface.

A temperature-compensated voltage reference and comparator circuit monitors the level of V_{CC} to detect power failures and to automatically switch to the backup supply when necessary. The RST pin provides an external pushbutton function and acts as an indicator of a power-fail event.

Operation

The block diagram shows the main elements of the DS3231. The eight blocks can be grouped into four functional groups: TCXO, power control, pushbutton function, and RTC. Their operations are described separately in the following sections.

32kHz TCXO

The temperature sensor, oscillator, and control logic form the TCXO. The controller reads the output of the on-chip temperature sensor and uses a lookup table to determine the capacitance required, adds the aging correction in AGE register, and then sets the capacitance selection registers. New values, including changes to the AGE register, are loaded only when a change in the temperature value occurs, or when a user-initiated temperature conversion is completed. Temperature conversion occurs on initial application of V_{CC} and once every 64 seconds afterwards.

Power Control

This function is provided by a temperature-compensated voltage reference and a comparator circuit that monitors the V_{CC} level. When V_{CC} is greater than V_{PF}, the part is powered by V_{CC}. When V_{CC} is less than V_{PF} but greater than V_{BAT}, the DS3231 is powered by V_{CC}. If V_{CC} is less than V_{PF} and is less than V_{BAT}, the device is powered by V_{BAT}. See Table 1.

Table 1. Power Control

SUPPLY CONDITION	ACTIVE SUPPLY
$V_{CC} < V_{PF}$, $V_{CC} < V_{BAT}$	V_{BAT}
$V_{CC} < V_{PF}$, $V_{CC} > V_{BAT}$	V_{CC}
$V_{CC} > V_{PF}$, $V_{CC} < V_{BAT}$	V_{CC}
$V_{CC} > V_{PF}$, $V_{CC} > V_{BAT}$	V_{CC}

To preserve the battery, the first time V_{BAT} is applied to the device, the oscillator will not start up until V_{CC} exceeds V_{PF}, or until a valid I²C address is written to the part. Typical oscillator startup time is less than one second. Approximately 2 seconds after V_{CC} is applied, or a valid I²C address is written, the device makes a temperature measurement and applies the calculated correction to the oscillator. Once the oscillator is running, it continues to run as long as a valid power source is available (V_{CC} or V_{BAT}), and the device continues to measure the temperature and correct the oscillator frequency every 64 seconds.

On the first application of power (V_{CC}) or when a valid I²C address is written to the part (V_{BAT}), the time and date registers are reset to 01/01/00 01 00:00:00 (DD/MM/YY DOW HH:MM:SS).

V_{BAT} Operation

There are several modes of operation that affect the amount of V_{BAT} current that is drawn. While the device is powered by V_{BAT} and the serial interface is active, active battery current, I_{BATA}, is drawn. When the serial interface is inactive, timekeeping current (I_{BATT}), which includes the averaged temperature conversion current, I_{BATTC}, is used (refer to Application Note 3644: *Power Considerations for Accurate Real-Time Clocks* for details). Temperature conversion current, I_{BATTC}, is specified since the system must be able to support the periodic higher current pulse and still maintain a valid voltage level. Data retention current, I_{BATTDR}, is the current drawn by the part when the oscillator is stopped ($\overline{EOSC}$ = 1). This mode can be used to minimize battery requirements for times when maintaining time and date information is not necessary, e.g., while the end system is waiting to be shipped to a customer.

Pushbutton Reset Function

The DS3231 provides for a pushbutton switch to be connected to the $\overline{RST}$ output pin. When the DS3231 is not in a reset cycle, it continuously monitors the $\overline{RST}$ signal for a low going edge. If an edge transition is detected, the DS3231 debounces the switch by pulling the $\overline{RST}$ low. After the internal timer has expired (PB_{DB}), the DS3231 continues to monitor the $\overline{RST}$ line. If the line is still low, the DS3231 continuously monitors the line looking for a rising edge. Upon detecting release, the DS3231 forces the $\overline{RST}$ pin low and holds it low for t_{RST}.

$\overline{RST}$ is also used to indicate a power-fail condition. When V_{CC} is lower than V_{PF}, an internal power-fail signal is generated, which forces the $\overline{RST}$ pin low. When V_{CC} returns to a level above V_{PF}, the $\overline{RST}$ pin is held low for approximately 250ms (t_{REC}) to allow the power supply to stabilize. If the oscillator is not running (see the *Power Control* section) when V_{CC} is applied, t_{REC} is bypassed and $\overline{RST}$ immediately goes high. Assertion of the $\overline{RST}$ output, whether by pushbutton or power-fail detection, does not affect the internal operation of the DS3231.

Real-Time Clock

With the clock source from the TCXO, the RTC provides seconds, minutes, hours, day, date, month, and year information. The date at the end of the month is automatically adjusted for months with fewer than 31 days, including corrections for leap year. The clock operates in either the 24-hour or 12-hour format with an AM/PM indicator.

The clock provides two programmable time-of-day alarms and a programmable square-wave output. The $\overline{INT}$/SQW pin either generates an interrupt due to alarm condition or outputs a square-wave signal and the selection is controlled by the bit INTCN.

ADDRESS	BIT 7 MSB	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0 LSB	FUNCTION	RANGE
00h	0	10 Seconds			Seconds				Seconds	00–59
01h	0	10 Minutes			Minutes				Minutes	00–59
02h	0	12/24	AM/PM / 20 Hour	10 Hour	Hour				Hours	1–12 + AM/PM / 00–23
03h	0	0	0	0	0	Day			Day	1–7
04h	0	0	10 Date		Date				Date	01–31
05h	Century	0	0	10 Month	Month				Month/ Century	01–12 + Century
06h	10 Year				Year				Year	00–99
07h	A1M1	10 Seconds			Seconds				Alarm 1 Seconds	00–59
08h	A1M2	10 Minutes			Minutes				Alarm 1 Minutes	00–59
09h	A1M3	12/24	AM/PM / 20 Hour	10 Hour	Hour				Alarm 1 Hours	1–12 + AM/PM / 00–23
0Ah	A1M4	DY/DT	10 Date		Day				Alarm 1 Day	1–7
					Date				Alarm 1 Date	1–31
0Bh	A2M2	10 Minutes			Minutes				Alarm 2 Minutes	00–59
0Ch	A2M3	12/24	AM/PM / 20 Hour	10 Hour	Hour				Alarm 2 Hours	1–12 + AM/PM / 00–23
0Dh	A2M4	DY/DT	10 Date		Day				Alarm 2 Day	1–7
					Date				Alarm 2 Date	1–31
0Eh	EOSC	BBSQW	CONV	RS2	RS1	INTCN	A2IE	A1IE	Control	—
0Fh	OSF	0	0	0	EN32kHz	BSY	A2F	A1F	Control/Status	—
10h	SIGN	DATA	DATA	DATA	DATA	DATA	DATA	DATA	Aging Offset	—
11h	SIGN	DATA	DATA	DATA	DATA	DATA	DATA	DATA	MSB of Temp	—
12h	DATA	DATA	0	0	0	0	0	0	LSB of Temp	—

Figure 1. Timekeeping Registers

***Note:** Unless otherwise specified, the registers' state is not defined when power is first applied.*

Address Map

Figure 1 shows the address map for the DS3231 timekeeping registers. During a multibyte access, when the address pointer reaches the end of the register space (12h), it wraps around to location 00h. On an I²C START or address pointer incrementing to location 00h, the current time is transferred to a second set of registers. The time information is read from these secondary registers, while the clock may continue to run. This eliminates the need to reread the registers in case the main registers update during a read.

I²C Interface

The I²C interface is accessible whenever either V_{CC} or V_{BAT} is at a valid level. If a microcontroller connected to the DS3231 resets because of a loss of V_{CC} or other event, it is possible that the microcontroller and DS3231 I²C communications could become unsynchronized, e.g., the microcontroller resets while reading data from the DS3231. When the microcontroller resets, the DS3231 I²C interface may be placed into a known state by toggling SCL until SDA is observed to be at a high level. At that point the microcontroller should pull SDA low while SCL is high, generating a START condition.

Clock and Calendar

The time and calendar information is obtained by reading the appropriate register bytes. Figure 1 illustrates the RTC registers. The time and calendar data are set or initialized by writing the appropriate register bytes. The contents of the time and calendar registers are in the binary-coded

decimal (BCD) format. The DS3231 can be run in either 12-hour or 24-hour mode. Bit 6 of the hours register is defined as the 12- or 24-hour mode select bit. When high, the 12-hour mode is selected. In the 12-hour mode, bit 5 is the AM/PM bit with logic-high being PM. In the 24-hour mode, bit 5 is the 20-hour bit (20–23 hours). The century bit (bit 7 of the month register) is toggled when the years register overflows from 99 to 00.

The day-of-week register increments at midnight. Values that correspond to the day of week are user-defined but must be sequential (i.e., if 1 equals Sunday, then 2 equals Monday, and so on). Illogical time and date entries result in undefined operation.

When reading or writing the time and date registers, secondary (user) buffers are used to prevent errors when the internal registers update. When reading the time and date registers, the user buffers are synchronized to the internal registers on any START and when the register pointer rolls over to zero. The time information is read from these secondary registers, while the clock continues to run. This eliminates the need to reread the registers in case the main registers update during a read.

The countdown chain is reset whenever the seconds register is written. Write transfers occur on the acknowledge from the DS3231. Once the countdown chain is reset, to avoid rollover issues the remaining time and date registers must be written within 1 second. The 1Hz square-wave output, if enabled, transitions high 500ms after the seconds data transfer, provided the oscillator is already running.

Alarms

The DS3231 contains two time-of-day/date alarms. Alarm 1 can be set by writing to registers 07h to 0Ah. Alarm 2 can be set by writing to registers 0Bh to 0Dh. The alarms can be programmed (by the alarm enable and INTCN bits of the control register) to activate the INT/SQW output on an alarm match condition. Bit 7 of each of the time-of-day/date alarm registers are mask bits (Table 2). When all the mask bits for each alarm are logic 0, an alarm only occurs when the values in the timekeeping registers match the corresponding values stored in the time-of-day/date alarm registers. The alarms can also be programmed to repeat every second, minute, hour, day, or date. Table 2 shows the possible settings. Configurations not listed in the table will result in illogical operation.

The DY/DT bits (bit 6 of the alarm day/date registers) control whether the alarm value stored in bits 0 to 5 of that register reflects the day of the week or the date of the month. If DY/DT is written to logic 0, the alarm will be the result of a match with date of the month. If DY/DT is written to logic 1, the alarm will be the result of a match with day of the week.

When the RTC register values match alarm register settings, the corresponding Alarm Flag 'A1F' or 'A2F' bit is set to logic 1. If the corresponding Alarm Interrupt Enable 'A1IE' or 'A2IE' is also set to logic 1 and the INTCN bit is set to logic 1, the alarm condition will activate the INT/SQW signal. The match is tested on the once-per-second update of the time and date registers.

Table 2. Alarm Mask Bits

DY/DT	ALARM 1 REGISTER MASK BITS (BIT 7)				ALARM RATE
	A1M4	A1M3	A1M2	A1M1	
X	1	1	1	1	Alarm once per second
X	1	1	1	0	Alarm when seconds match
X	1	1	0	0	Alarm when minutes and seconds match
X	1	0	0	0	Alarm when hours, minutes, and seconds match
0	0	0	0	0	Alarm when date, hours, minutes, and seconds match
1	0	0	0	0	Alarm when day, hours, minutes, and seconds match

DY/DT	ALARM 2 REGISTER MASK BITS (BIT 7)			ALARM RATE
	A2M4	A2M3	A2M2	
X	1	1	1	Alarm once per minute (00 seconds of every minute)
X	1	1	0	Alarm when minutes match
X	1	0	0	Alarm when hours and minutes match
0	0	0	0	Alarm when date, hours, and minutes match
1	0	0	0	Alarm when day, hours, and minutes match

Control Register (0Eh)

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
NAME:	EOSC	BBSQW	CONV	RS2	RS1	INTCN	A2IE	A1IE
POR:	0	0	0	1	1	1	0	0

Special-Purpose Registers

The DS3231 has two additional registers (control and status) that control the real-time clock, alarms, and square-wave output.

Control Register (0Eh)

Bit 7: Enable Oscillator ($\overline{EOSC}$). When set to logic 0, the oscillator is started. When set to logic 1, the oscillator is stopped when the DS3231 switches to V_{BAT}. This bit is clear (logic 0) when power is first applied. When the DS3231 is powered by V_{CC}, the oscillator is always on regardless of the status of the $\overline{EOSC}$ bit. When $\overline{EOSC}$ is disabled, all register data is static.

Bit 6: Battery-Backed Square-Wave Enable (BBSQW). When set to logic 1 with INTCN = 0 and $V_{CC} < V_{PF}$, this bit enables the square wave. When BBSQW is logic 0, the $\overline{INT}$/SQW pin goes high impedance when $V_{CC} < V_{PF}$. This bit is disabled (logic 0) when power is first applied.

Bit 5: Convert Temperature (CONV). Setting this bit to 1 forces the temperature sensor to convert the temperature into digital code and execute the TCXO algorithm to update the capacitance array to the oscillator. This can only happen when a conversion is not already in progress. The user should check the status bit BSY before forcing the controller to start a new TCXO execution. A user-initiated temperature conversion does not affect the internal 64-second update cycle.

A user-initiated temperature conversion does not affect the BSY bit for approximately 2ms. The CONV bit remains at a 1 from the time it is written until the conversion is finished, at which time both CONV and BSY go to 0. The CONV bit should be used when monitoring the status of a user-initiated conversion.

Bits 4 and 3: Rate Select (RS2 and RS1). These bits control the frequency of the square-wave output when the square wave has been enabled. The following table shows the square-wave frequencies that can be selected with the RS bits. These bits are both set to logic 1 (8.192kHz) when power is first applied.

SQUARE-WAVE OUTPUT FREQUENCY

RS2	RS1	SQUARE-WAVE OUTPUT FREQUENCY
0	0	1Hz
0	1	1.024kHz
1	0	4.096kHz
1	1	8.192kHz

Bit 2: Interrupt Control (INTCN). This bit controls the $\overline{INT}$/SQW signal. When the INTCN bit is set to logic 0, a square wave is output on the $\overline{INT}$/SQW pin. When the INTCN bit is set to logic 1, then a match between the timekeeping registers and either of the alarm registers activates the $\overline{INT}$/SQW output (if the alarm is also enabled). The corresponding alarm flag is always set regardless of the state of the INTCN bit. The INTCN bit is set to logic 1 when power is first applied.

Bit 1: Alarm 2 Interrupt Enable (A2IE). When set to logic 1, this bit permits the alarm 2 flag (A2F) bit in the status register to assert $\overline{INT}$/SQW (when INTCN = 1). When the A2IE bit is set to logic 0 or INTCN is set to logic 0, the A2F bit does not initiate an interrupt signal. The A2IE bit is disabled (logic 0) when power is first applied.

Bit 0: Alarm 1 Interrupt Enable (A1IE). When set to logic 1, this bit permits the alarm 1 flag (A1F) bit in the status register to assert $\overline{INT}$/SQW (when INTCN = 1). When the A1IE bit is set to logic 0 or INTCN is set to logic 0, the A1F bit does not initiate the $\overline{INT}$/SQW signal. The A1IE bit is disabled (logic 0) when power is first applied.

Status Register (0Fh)

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
NAME:	OSF	0	0	0	EN32kHz	BSY	A2F	A1F
POR:	1	0	0	0	1	X	X	X

Status Register (0Fh)

Bit 7: Oscillator Stop Flag (OSF). A logic 1 in this bit indicates that the oscillator either is stopped or was stopped for some period and may be used to judge the validity of the timekeeping data. This bit is set to logic 1 any time that the oscillator stops. The following are examples of conditions that can cause the OSF bit to be set:

1) The first time power is applied.
2) The voltages present on both V_{CC} and V_{BAT} are insufficient to support oscillation.
3) The $\overline{EOSC}$ bit is turned off in battery-backed mode.
4) External influences on the crystal (i.e., noise, leakage, etc.).

This bit remains at logic 1 until written to logic 0.

Bit 3: Enable 32kHz Output (EN32kHz). This bit controls the status of the 32kHz pin. When set to logic 1, the 32kHz pin is enabled and outputs a 32.768kHz square-wave signal. When set to logic 0, the 32kHz pin goes to a high-impedance state. The initial power-up state of this bit is logic 1, and a 32.768kHz square-wave signal appears at the 32kHz pin after a power source is applied to the DS3231 (if the oscillator is running).

Bit 2: Busy (BSY). This bit indicates the device is busy executing TCXO functions. It goes to logic 1 when the conversion signal to the temperature sensor is asserted and then is cleared when the device is in the 1-minute idle state.

Bit 1: Alarm 2 Flag (A2F). A logic 1 in the alarm 2 flag bit indicates that the time matched the alarm 2 registers. If the A2IE bit is logic 1 and the INTCN bit is set to logic 1, the $\overline{INT}$/SQW pin is also asserted. A2F is cleared when written to logic 0. This bit can only be written to logic 0. Attempting to write to logic 1 leaves the value unchanged.

Bit 0: Alarm 1 Flag (A1F). A logic 1 in the alarm 1 flag bit indicates that the time matched the alarm 1 registers. If the A1IE bit is logic 1 and the INTCN bit is set to logic 1, the $\overline{INT}$/SQW pin is also asserted. A1F is cleared when written to logic 0. This bit can only be written to logic 0. Attempting to write to logic 1 leaves the value unchanged.

Aging Offset

The aging offset register takes a user-provided value to add to or subtract from the codes in the capacitance array registers. The code is encoded in two's complement, with bit 7 representing the sign bit. One LSB represents one small capacitor to be switched in or out of the capacitance array at the crystal pins. The aging offset register capacitance value is added or subtracted from the capacitance value that the device calculates for each temperature compensation. The offset register is added to the capacitance array during a normal temperature conversion, if the temperature changes from the previous conversion, or during a manual user conversion (setting the CONV bit). To see the effects of the aging register on the 32kHz output frequency immediately, a manual conversion should be started after each aging register change.

Positive aging values add capacitance to the array, slowing the oscillator frequency. Negative values remove capacitance from the array, increasing the oscillator frequency.

The change in ppm per LSB is different at different temperatures. The frequency vs. temperature curve is shifted by the values used in this register. At +25°C, one LSB typically provides about 0.1ppm change in frequency.

Use of the aging register is not needed to achieve the accuracy as defined in the EC tables, but could be used to help compensate for aging at a given temperature. See the *Typical Operating Characteristics* section for a graph showing the effect of the register on accuracy over temperature.

Aging Offset (10h)

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
NAME:	Sign	Data	Data	Data	Data	Data	Data	Data
POR:	0	0	0	0	0	0	0	0

Temperature Register (Upper Byte) (11h)

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
NAME:	Sign	Data	Data	Data	Data	Data	Data	Data
POR:	0	0	0	0	0	0	0	0

Temperature Register (Lower Byte) (12h)

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
NAME:	Data	Data	0	0	0	0	0	0
POR:	0	0	0	0	0	0	0	0

Temperature Registers (11h–12h)

Temperature is represented as a 10-bit code with a resolution of 0.25°C and is accessible at location 11h and 12h. The temperature is encoded in two's complement format. The upper 8 bits, the integer portion, are at location 11h and the lower 2 bits, the fractional portion, are in the upper nibble at location 12h. For example, 00011001 01b = +25.25°C. Upon power reset, the registers are set to a default temperature of 0°C and the controller starts a temperature conversion. The temperature is read on initial application of V_{CC} or I²C access on V_{BAT} and once every 64 seconds afterwards. The temperature registers are updated after each user-initiated conversion and on every 64-second conversion. The temperature registers are read-only.

I²C Serial Data Bus

The DS3231 supports a bidirectional I²C bus and data transmission protocol. A device that sends data onto the bus is defined as a transmitter and a device receiving data is defined as a receiver. The device that controls the message is called a master. The devices that are controlled by the master are slaves. The bus must be controlled by a master device that generates the serial clock (SCL), controls the bus access, and generates the START and STOP conditions. The DS3231 operates as a slave on the I²C bus. Connections to the bus are made through the SCL input and open-drain SDA I/O lines. Within the bus specifications, a standard mode (100kHz maximum clock rate) and a fast mode (400kHz maximum clock rate) are defined. The DS3231 works in both modes.

The following bus protocol has been defined (Figure 2):

- Data transfer may be initiated only when the bus is not busy.
- During data transfer, the data line must remain stable whenever the clock line is high. Changes in the data line while the clock line is high are interpreted as control signals.

Accordingly, the following bus conditions have been defined:

Bus not busy: Both data and clock lines remain high.

START data transfer: A change in the state of the data line from high to low, while the clock line is high, defines a START condition.

STOP data transfer: A change in the state of the data line from low to high, while the clock line is high, defines a STOP condition.

Data valid: The state of the data line represents valid data when, after a START condition, the data line is stable for the duration of the high period of the clock signal. The data on the line must be changed during the low period of the clock signal. There is one clock pulse per bit of data.

Each data transfer is initiated with a START condition and terminated with a STOP condition. The number of data bytes transferred between the START and the STOP conditions is not limited, and is determined by the master device. The information is transferred byte-wise and each receiver acknowledges with a ninth bit.

Acknowledge: Each receiving device, when addressed, is obliged to generate an acknowledge after the reception of each byte. The master device must generate an extra clock pulse, which is associated with this acknowledge bit.

A device that acknowledges must pull down the SDA line during the acknowledge clock pulse in such a way that the SDA line is stable low during the high period of the acknowledge-related clock pulse. Of course, setup and hold times must be taken into account. A master must signal an end of data to the slave by not generat-

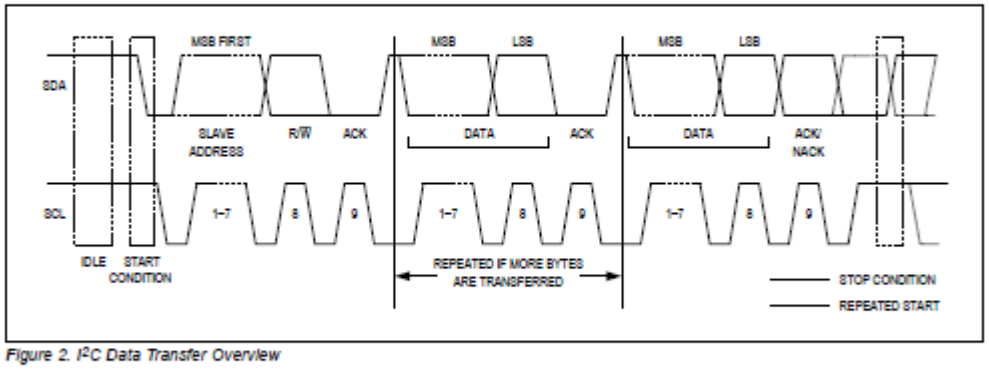


Figure 2. I²C Data Transfer Overview

ing an acknowledge bit on the last byte that has been clocked out of the slave. In this case, the slave must leave the data line high to enable the master to generate the STOP condition.

Figures 3 and 4 detail how data transfer is accomplished on the I²C bus. Depending upon the state of the R/W bit, two types of data transfer are possible:

Data transfer from a master transmitter to a slave receiver. The first byte transmitted by the master is the slave address. Next follows a number of data bytes. The slave returns an acknowledge bit after each received byte. Data is transferred with the most significant bit (MSB) first.

Data transfer from a slave transmitter to a master receiver. The first byte (the slave address) is transmitted by the master. The slave then returns an acknowledge bit. Next follows a number of data bytes transmitted by the slave to the master. The master returns an acknowledge bit after all received bytes other than the

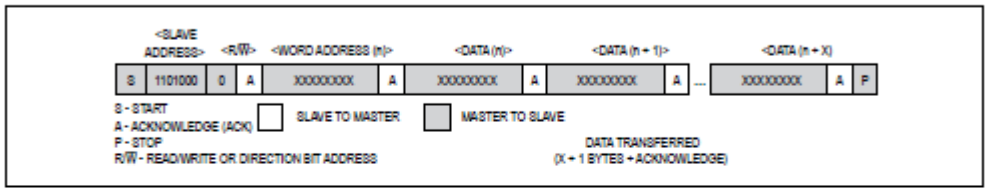


Figure 3. Data Write—Slave Receiver Mode

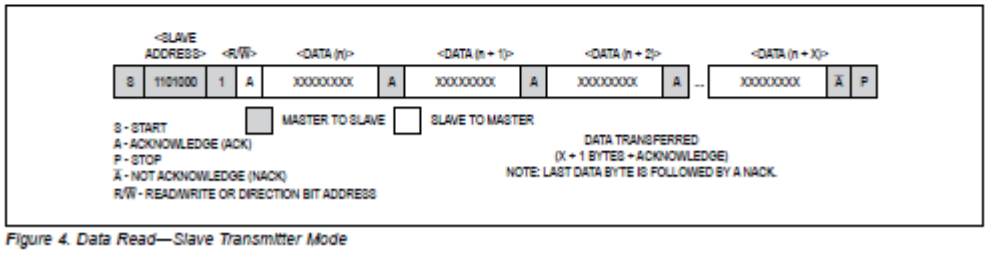


Figure 4. Data Read—Slave Transmitter Mode

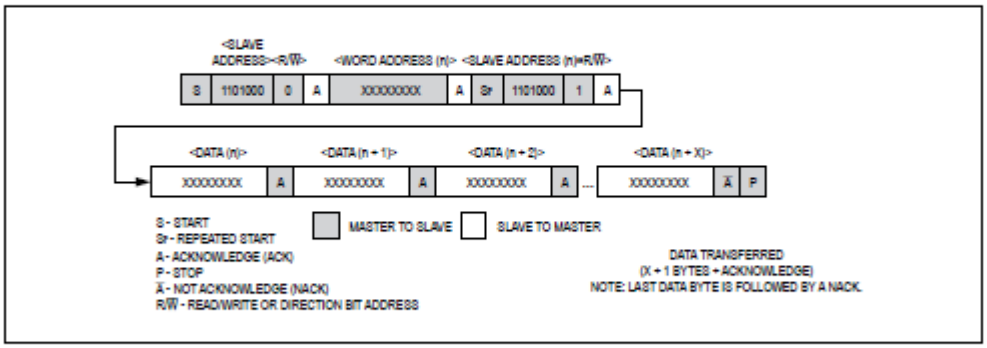


Figure 5. Data Write/Read (Write Pointer, Then Read)—Slave Receive and Transmit

last byte. At the end of the last received byte, a not acknowledge is returned.

The master device generates all the serial clock pulses and the START and STOP conditions. A transfer is ended with a STOP condition or with a repeated START condition. Since a repeated START condition is also the beginning of the next serial transfer, the bus will not be released. Data is transferred with the most significant bit (MSB) first.

The DS3231 can operate in the following two modes:

Slave receiver mode (DS3231 write mode): Serial data and clock are received through SDA and SCL. After each byte is received, an acknowledge bit is transmitted. START and STOP conditions are recognized as the beginning and end of a serial transfer. Address recognition is performed by hardware after reception of the slave address and direction bit. The slave address byte is the first byte received after the master generates the START condition. The slave address byte contains the 7-bit DS3231 address, which is 1101000, followed by the direction bit (R/$\overline{W}$), which is 0 for a write. After receiving and decoding the slave address byte, the DS3231 outputs an acknowledge on SDA. After the DS3231 acknowledges the slave address + write bit, the master transmits a word address to the DS3231. This sets the register pointer on the DS3231, with the DS3231 acknowledging the transfer. The master may then transmit zero or more bytes of data, with the DS3231 acknowledging each byte received. The register pointer increments after each data byte is transferred. The master generates a STOP condition to terminate the data write.

Slave transmitter mode (DS3231 read mode): The first byte is received and handled as in the slave receiver mode. However, in this mode, the direction bit indicates that the transfer direction is reversed. Serial data is transmitted on SDA by the DS3231 while the serial clock is input on SCL. START and STOP conditions are recognized as the beginning and end of a serial transfer. Address recognition is performed by hardware after reception of the slave address and direction bit. The slave address byte is the first byte received after the master generates a START condition. The slave address byte contains the 7-bit DS3231 address, which is 1101000, followed by the direction bit (R/$\overline{W}$), which is 1 for a read. After receiving and decoding the slave address byte, the DS3231 outputs an acknowledge on SDA. The DS3231 then begins to transmit data starting with the register address pointed to by the register pointer. If the register pointer is not written to before the initiation of a read mode, the first address that is read is the last one stored in the register pointer. The DS3231 must receive a not acknowledge to end a read.

DS3231

Extremely Accurate I²C-Integrated RTC/TCXO/Crystal

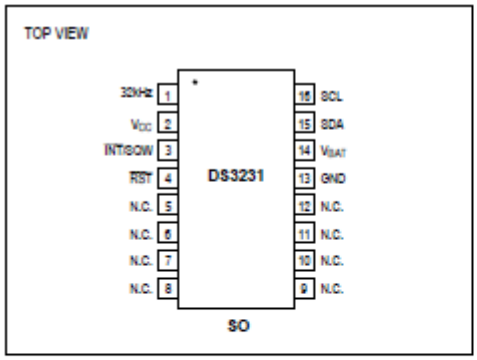
Handling, PCB Layout, and Assembly

The DS3231 package contains a quartz tuning-fork crystal. Pick-and-place equipment can be used, but precautions should be taken to ensure that excessive shocks are avoided. Ultrasonic cleaning should be avoided to prevent damage to the crystal.

Avoid running signal traces under the package, unless a ground plane is placed between the package and the signal line. All N.C. (no connect) pins must be connected to ground.

Moisture-sensitive packages are shipped from the factory dry packed. Handling instructions listed on the package label must be followed to prevent damage during reflow. Refer to the IPC/JEDEC J-STD-020 standard for moisture-sensitive device (MSD) classifications and reflow profiles. Exposure to reflow is limited to 2 times maximum.

Pin Configuration



Chip Information

SUBSTRATE CONNECTED TO GROUND

PROCESS: CMOS

Ordering Information

PART	TEMP RANGE	PIN-PACKAGE
DS3231S#	0°C to +70°C	16 SO
DS3231SN#	-40°C to +85°C	16 SO

#Denotes an RoHS-compliant device that may include lead (Pb) that is exempt under RoHS requirements. The lead finish is JESD97 category e3, and is compatible with both lead-based and lead-free soldering processes. A "#" anywhere on the top mark denotes an RoHS-compliant device.

Package Information

For the latest package outline information and land patterns (footprints), go to www.maximintegrated.com/packages. Note that a "+", "#", or "-" in the package code indicates RoHS status only. Package drawings may show a different suffix character, but the drawing pertains to the package regardless of RoHS status.

PACKAGE TYPE	PACKAGE CODE	OUTLINE NO.	LAND PATTERN NO.
16 SO	W16#H2	21-0042	90-0107

Revision History

REVISION NUMBER	REVISION DATE	DESCRIPTION	PAGES CHANGED
0	1/05	Initial release.	—
1	2/05	Changed Digital Temp Sensor Output from ±2°C to ±3°C.	1, 3
		Updated *Typical Operating Circuit*.	1
		Changed T_A = -40°C to +85°C to T_A = T_{MIN} to T_{MAX}.	2, 3, 4
		Updated *Block Diagram*.	8
2	6/05	Added "UL Recognized" to Features; added lead-free packages and removed S from top mark info in *Ordering Information* table; added ground connections to the N.C. pin in the Typical Operating Circuit.	1
		Added "noncondensing" to operating temperature range; changed V_{PF} MIN from 2.35V to 2.45V.	2
		Added aging offset specification.	3
		Relabeled TOC4.	7
		Added arrow showing input on X1 in the *Block Diagram*.	8
		Updated pin descriptions for V_{CC} and V_{BAT}.	9
		Added the I²C interface section.	10
		Figure 1: Added sign bit to aging and temperature registers; added MSB and LSB.	11
		Corrected title for rate select bits frequency table.	13
		Added note that frequency stability over temperature spec is with aging offset register = 00h; changed bit 7 from Data to Sign (Crystal Aging Offset Register).	14
		Changed bit 7 from Data to Sign (Temperature Register); correct pin definitions in *I²C Serial Data Bus* section.	15
		Modified the *Handing, PC Board Layout,* and *Assembly* section to refer to J-STD-020 for reflow profiles for lead-free and leaded packages.	17
3	11/05	Changed lead-free packages to RoHS-compliant packages.	1
4	10/06	Changed $\overline{RST}$ and UL bullets in *Features*.	1
		Changed EC condition "V_{CC} > V_{BAT}" to "V_{CC} = Active Supply (see Table 1)."	2, 3
		Modified Note 12 to correct t_{REC} operation.	6
		Added various conditions text to TOCs 1, 2, and 3.	7
		Added text to pin descriptions for 32kHz, V_{CC}, and $\overline{RST}$.	9
		Table 1: Changed column heading "Powered By" to "Active Supply"; changed "applied" to "exceeds V_{PF}" in the *Power Control* section.	10
		Indicated BBSQW applies to both SQW and interrupts; simplified temp convert description (bit 5); added "output" to $\overline{INT}$/SQW (bit 2).	13
		Changed the *Crystal Aging* section to the *Aging Offset* section; changed "this bit indicates" to "this bit controls" for the enable 32kHz output bit.	14
5	4/08	Added Warning note to EC table notes; updated Note 12.	6
		Updated the *Typical Operating Characteristics* graphs.	7
		In the *Power Control* section, added information about the POR state of the time and date registers; in the *Real-Time Clock* section, added to the description of the $\overline{RST}$ function.	10
		In Figure 1, corrected the months date range for 04h from 00–31 to 01–31.	11

Revision History (continued)

REVISION NUMBER	REVISION DATE	DESCRIPTION	PAGES CHANGED
6	10/08	Updated the *Typical Operating Circuit*.	1
		Removed the V_{PU} parameter from the *Recommended DC Operating Conditions* table and added verbiage about the pullup to the *Pin Description* table for INT/SQW, SDA, and SCL.	2, 9
		Added the Delta Time and Frequency vs. Temperature graph in the *Typical Operating Characteristics* section.	7
		Updated the *Block Diagram*.	8
		Added the V_{BAT} *Operation* section, improved some sections of text for the *32kHz TCXO* and *Pushbutton Reset Function* sections.	10
		Added the register bit POR values to the register tables.	13, 14, 15
		Updated the *Aging Offset* and *Temperature Registers (11h–12h)* sections.	14, 15
		Updated the I2C timing diagrams (Figures 3, 4, and 5).	16, 17
7	3/10	Removed the "S" from the top mark in the *Ordering Information* table and the *Pin Configuration* to match the packaging engineering marking specification.	1, 18
8	7/10	Updated the *Typical Operating Circuit*; removed the "Top Mark" column from the *Ordering Information*; in the *Absolute Maximum Ratings* section, added the theta-JA and theta-JC thermal resistances and Note 1, and changed the soldering temperature to +260°C (lead(Pb)-free) and +240°C (leaded); updated the functional description of the V_{BAT} pin in the *Pin Description*; changed the timekeeping registers 02h, 09h, and 0Ch to "20 Hour" in Bit 5 of Figure 1; updated the BBSQW bit description in the *Control Register (0Eh)* section; added the land pattern no. to the *Package Information* table.	1, 2, 3, 4, 6, 9, 11, 12, 13, 18
9	1/13	Updated *Absolute Maximum Ratings*, and last paragraph in *Power Control* section	2, 10
10	3/15	Revised *Benefits and Features* section.	1