



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

Sistema de Control de Acceso RFID

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

Autor:

Martino Ignacio Gastón

2018

Acreditación:

Fecha:

Comité Evaluador

Presidente: MSc. Ing. Pedro Danizio

1° Vocal: Esp. Ing. Marcelo Cejas

2° Vocal: Ing. Fabián Sensini

Dedicatorias

A mis padres, mi hermana, y la mujer que hoy es mi fiel compañera, quienes son el combustible inacabable que mueve la maquinaria de mi ser y que siempre han estado a mi lado para apoyarme y guiarme en este camino.

A mis seres queridos.

A mis compañeros con los cuales transite este apasionante camino.

Agradecimientos

Llegado este momento tan especial en mi vida quiero comenzar agradeciendo a mi madre Eva y a mi padre Javier por haberme apoyado, en todos los sentidos, en cada actividad que decidí realizar en estos años de estudios.

A mi hermana Lucia, por haber estado siempre a mi lado.

A Romina, mi fiel compañera, que siempre me ha acompañado y apoyado firmemente a lo largo de este trayecto.

A todos los profesores, quienes fueron mis mentores a lo largo de la carrera.

A mis compañeros, de los cuales recibí consejos, ideas y sugerencias.

Por último, al ingeniero Fabián Sensini, quien dio una respuesta a cada una de mis inquietudes.

Memoria Descriptiva

El presente proyecto consta de un dispositivo capaz de proporcionar un control de acceso y registro histórico de los mismos, contando con una base de datos en la cual se encuentran registrados aquellos usuarios que tienen permitido el acceso a un área o recinto de un determinado lugar.

Un lector/grabador RFID se encarga de extraer la información grabada en tarjetas, llaveros o tags RFID, para luego comparar dicha información con la que se encuentra almacenada en la base de datos del sistema (almacenamiento masivo, compuesto por una tarjeta microSD).

Luego se lleva a cabo la validación de los datos. Si el usuario se encuentra en el sistema, se procede a dar acceso al mismo dejando asentado nombre, hora y fecha en un registro. Si no hay coincidencia se restringe el acceso. De manera opcional se incluye un sistema de accionamiento magnético, el cual mantiene bloqueada una cerradura o la abre en función de si hay o no coincidencias, siempre y cuando el dispositivo desee emplearse para tal fin.

Mediante un LCD montado en panel frontal del dispositivo, se pueden visualizar diferentes mensajes en función de las operaciones que se lleven a cabo.

Índice

| | |
|--|-----------|
| Dedicatorias | 3 |
| Agradecimientos..... | 4 |
| Memoria Descriptiva..... | 5 |
| Índice | 6 |
| 1. Introducción | 8 |
| 2. Objetivos | 10 |
| 3. Adquisición de los datos | 11 |
| 3.1. Lectura de las tarjetas RFID | 11 |
| 3.1.1. Acerca del módulo de radiofrecuencia..... | 11 |
| 3.1.2. Acerca de las tarjetas RFID empleadas..... | 16 |
| 3.1.3. Modo de operación del lector/grabador..... | 18 |
| 3.2. Recepción de los datos en el microcontrolador | 20 |
| 3.2.1. Acerca del periférico MSSP (modo SPI)..... | 21 |
| 4. Base de datos..... | 24 |
| 4.1. Acerca de la Tarjeta microSD..... | 24 |
| 4.1.1. Pinout y conexiónado con el microcontrolador | 25 |
| 4.1.2. Modo de operación de la Tarjeta..... | 26 |
| 4.2. Acerca de la conformación de la Base de Datos..... | 28 |
| 5. Real Time Clock (RTC)..... | 30 |
| 5.1. Acerca del módulo RTC | 30 |
| 5.1.1. Acerca del DS1307..... | 31 |
| 5.2. Conexiónado y comunicación con el microcontrolador | 32 |
| 5.2.1. Acerca del periférico MSSP en modo I²C..... | 33 |
| 6. Sistema De Accionamiento | 37 |
| 6.1. Acerca del circuito de accionamiento | 37 |
| 7. Microcontrolador..... | 39 |
| 7.1. Comparación de la información..... | 39 |
| 7.1.1. Acerca del oscilador del sistema del microcontrolador..... | 40 |
| 7.1.2. Acerca de la programación | 41 |
| 8. Gestión de las Tarjetas RFID | 45 |
| 8.1. Acerca del módulo conversor | 45 |
| 8.1.1. Conexiónado y comunicación con el microcontrolador..... | 46 |
| 8.1.2. Acerca del módulo EUSART | 47 |

| | | |
|--------------|--|----|
| 8.2. | Interfaz gráfica de usuario (GUI)..... | 49 |
| 8.2.1 | Acerca de la interfaz..... | 49 |
| 9. | Alimentación | 51 |
| 9.1. | Acerca del regulador de 5 voltios..... | 51 |
| 9.2. | Acerca del regulador de 3,3 voltios..... | 52 |
| 10. | Integración..... | 54 |
| 11. | Conclusiones | 57 |
| 12. | Bibliografía..... | 58 |
| 13. | Anexos | 59 |
| 13.1. | Anexo Alfa | 59 |
| 13.2. | Anexo Bravo..... | 78 |
| 13.3. | Anexo Charly | 91 |
| 13.4. | Anexo Delta..... | 99 |

1. Introducción

En la actualidad, ya sea por cuestiones de seguridad u organización, las empresas, establecimientos y/o comercios buscan tener siempre su personal controlado de manera sencilla y organizada, tanto en los ingresos como los egresos de este al establecimiento, junto con sus marcas de tiempo correspondientes. Existen varios dispositivos para llevar a cabo esta tarea, como por ejemplo aquellos por control biométrico (uso de lectores de retina, faciales, dactilares, etc.) o mediante el ingreso de un PIN por teclado, entre otros. El primero de ellos resulta ser en ocasiones demasiado costoso y el segundo implica la introducción de una clave de manera manual. Mediante este proyecto se propone una solución inalámbrica para este tipo de tarea.

Un Sistema de Control de Acceso RFID (**R**adio **F**requency **I**Dentification, en español **I**Dentificación por **R**adio **F**recuencia) es un sistema de seguridad utilizado para el control de acceso mediante tarjetas de proximidad o compatibles (pulsera, llavero, tags). Este sistema mejora la gestión de llaves y usuarios en la entrada y salida de una oficina, departamento, complejos, etc.

En la Fig. 1 puede observarse un diagrama en bloques del dispositivo.

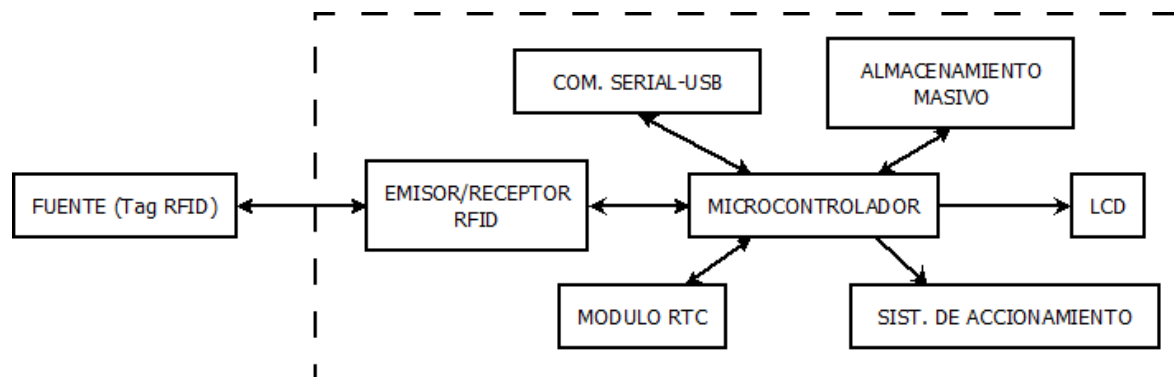


Fig. 1. Diagrama en bloques del sistema.

La “**Fuente**” hace referencia al tag RFID, el cual es un dispositivo pasivo que contiene la información relacionada a cada usuario, siendo así único e intransferible.

El “**Lector**” es la parte del sistema encargada de interactuar con los tags de manera inalámbrica, para de esta manera acceder a la información en ellos almacenada. Una vez hecho esto, dicha información es transferida al microcontrolador.

El “**Microcontrolador**” procesa la información proveniente del lector y procede a buscar coincidencias en la base de datos del sistema. De producirse un acierto, concede el acceso al usuario, al mismo tiempo que extrae del módulo RTC la fecha y hora exacta, para luego registrar el ingreso o egreso en la base datos del sistema.

El “**Almacenamiento Masivo**” es un dispositivo que alberga la información de cada usuario registrado en el sistema, así como también los logueos de cada uno de estos. Constituye la base de datos.

El “**Módulo RTC**” es quien proporciona al sistema de manera exacta la hora y fecha en la cual se producen cada uno de los logueos en el mismo.

El “**Sistema de Accionamiento**” es el encargado de mantener cerrada o abrir la cerradura de la puerta al lugar o contenido que se desea acceder.

La “**Comunicación Serial-USB**” permite establecer una comunicación con un ordenador, para de esta manera gestionar el dispositivo y llevar a cabo diferentes tareas con el mismo.

El “**LCD**” es una pequeña pantalla de cristal líquido, la cual funciona a modo de visor de eventos durante el proceso de validación y acceso al sistema.

2. Objetivos

Los objetivos de este proyecto son:

- Implementar un sistema de seguridad y control económico y eficiente.
- Mejorar la gestión de llaves y usuarios a la entrada de una oficina, departamento o complejo.
- Eliminar la necesidad de contacto físico a la hora del acceso.
- Crear una interfaz gráfica de usuario para la gestión de las tarjetas y el dispositivo.
- Disponer de una base de datos en la cual queden asentados todos los ingresos/egresos realizados, con nombre de usuario, hora y fecha.

En la Fig. 3 se observa como es físicamente el módulo.

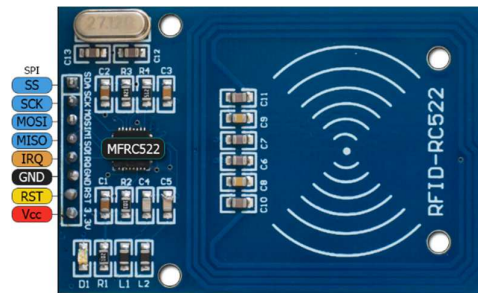


Fig. 3. Fotografía del módulo.

El componente principal del módulo es el integrado MFRC522, el cual es un lector/grabador altamente integrado para la comunicación inalámbrica, que opera a una frecuencia de 13,56 MHz. Dicho integrado soporta la norma ISO/IEC 14443 A/MIFARE y NTAG [13], el cual es un estándar internacional relacionado con las tarjetas de identificación electrónicas, en especial las tarjetas de proximidad.

El MFRC522 posee un transmisor interno, el cual es capaz de manejar una antena para comunicarse con tarjetas del tipo MIFARE según la norma ISO/IEC 14443 y otros transpondedores sin la necesidad de emplear circuitos activos. Presenta un módulo receptor, el cual proporciona una implementación robusta y eficiente para la demodulación y decodificación de señales de tarjetas y transpondedores compatibles que soporten dicha norma y un módulo digital que además de gestionar de manera completa la norma ISO/IEC 14443, proporciona una función de detección de errores (por paridad y CRC).

Cuenta con varios protocolos de comunicación (UART, I²C, SPI) para la comunicación con el host y esta empaquetado en un encapsulado de 32-pines HVQFN.

En la Fig. 4 se puede observar un diagrama en bloques simplificado del integrado.

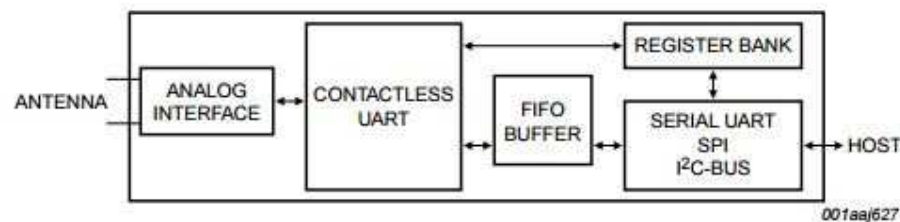


Fig. 4. Diagrama en bloques simplificado del MFRC522 [1].

La interfaz analógica se encarga de la modulación y demodulación de las señales analógicas cuando se lleva a cabo la comunicación con una tarjeta.

El bloque UART gestiona la comunicación entre las tarjetas RFID a través de la antena y la lógica interna del integrado, a su vez que se encarga también de la comunicación con el dispositivo

host. La tarea del buffer FIFO es que los datos sean transferidos de manera rápida y conveniente desde el host al bloque UART y viceversa.

En la Fig. 5 se puede apreciar un diagrama en bloques detallado del integrado.

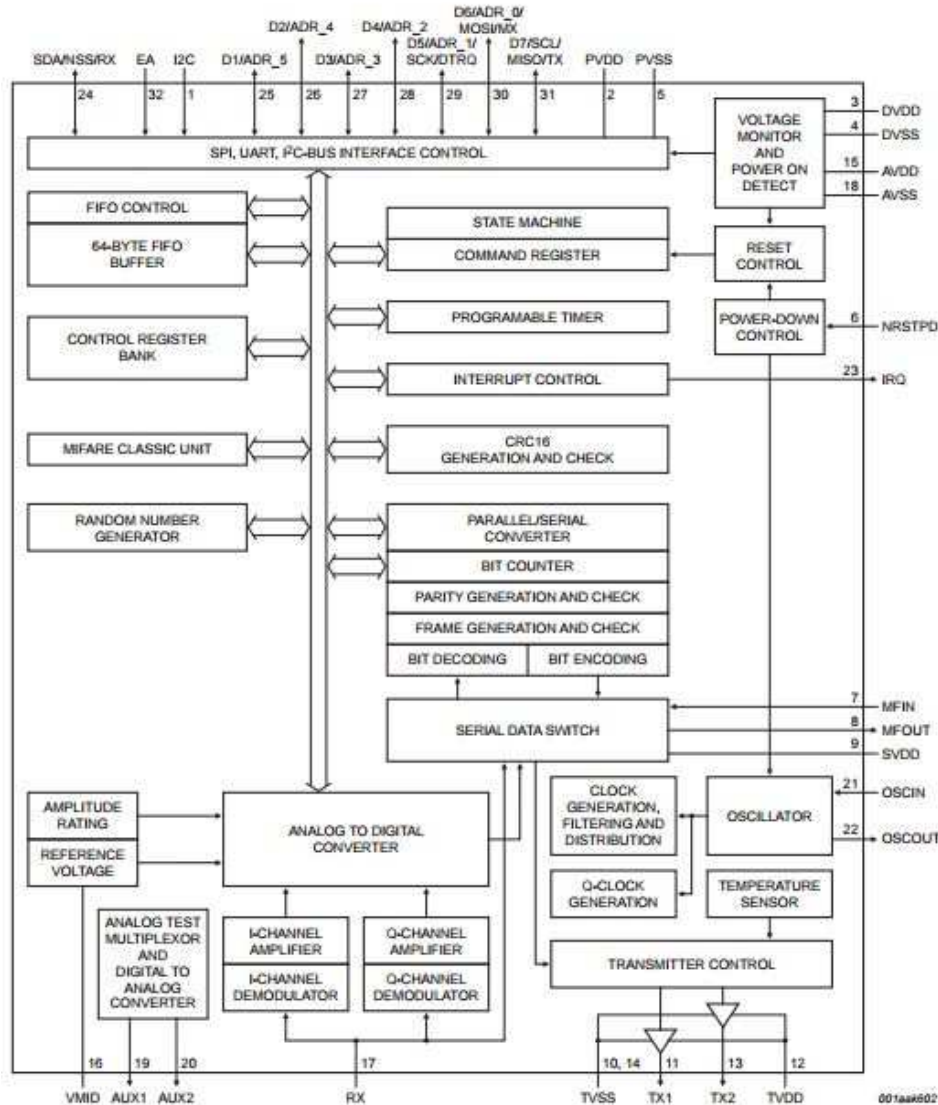


Fig. 5. Diagrama en bloques detallado MFRC522 [1].

Los parámetros de funcionamiento de este integrado, como por ejemplo, la ganancia de la antena, el tipo de modulación, el método CRC, etc., se configuran a través de una serie de registros internos (CONTROL REGISTER BANK) haciendo uso de un conjunto de comandos, los cuales son proporcionados por el host e ingresan al integrado a través de alguna de las interfaces arriba mencionadas. Para ver los comandos y las direcciones de los registros internos del MFRC522, se recomienda observar su hoja de especificaciones [1].

Si bien el integrado presenta varias interfaces de comunicación con el host, solo se trabajará con la interfaz SPI (Serial Peripheral Interface), ya que el módulo empleado en el presente proyecto viene diseñado para trabajar con dicho protocolo de comunicación. Esta interfaz permite

manejar datos a velocidades de hasta 10 Mbit/s. Al comunicarse con un host, el MFRC522 actúa como esclavo recibiendo datos para los registros de configuración, para posteriormente enviar y recibir información relevante mediante la interfaz RF.

En la Fig. 6 puede observarse un diagrama de ondas para un proceso de escritura y lectura del protocolo SPI y en la Fig. 7 las líneas de conexión del MFRC522 con el host (en este caso el microcontrolador).

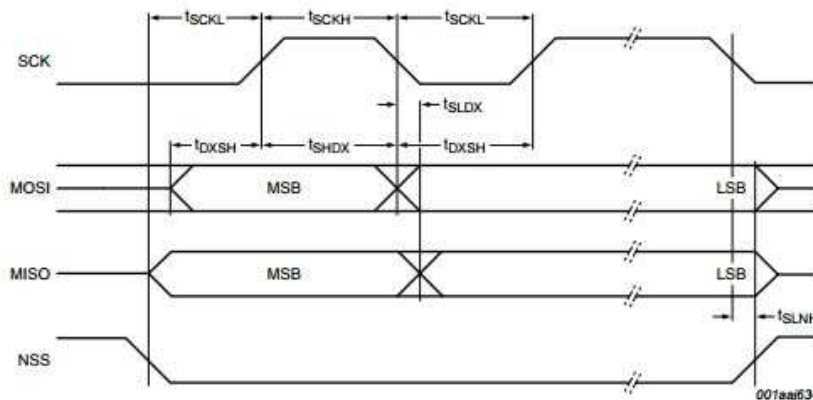


Fig. 6. Diagrama de ondas para un proceso de escritura y/o lectura [1].

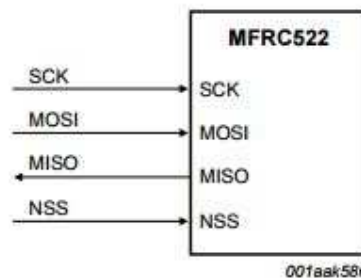


Fig. 7. Líneas de conexión de la interfaz SPI al MFRC522 [1]

Como puede verse en la Fig. 6, para iniciar un proceso de escritura o lectura de algunos de los registros del MFRC522, es necesario proporcionar un flanco descendente en el pin NSS (pin de selección de chip o CS) del módulo (de esto se encarga el micro). Para finalizar dicho proceso debe proporcionarse un flanco ascendente en dicho pin. Independientemente de que sea lectura o escritura, debe escribirse el comando (tipo de operación y dirección) en la línea MOSI (inmediatamente después del flanco descendente de NSS). Luego dependerá si es una escritura o lectura se continuará escribiendo en la línea MOSI o se seguirá leyendo en la línea MISO respectivamente. Como puede verse estos tres pines cumplen con la estructura de SPI (una línea de reloj y dos líneas de datos unidireccionales).

Tanto en la línea MOSI como MISO, los bytes de datos se envían siempre con el MSB primero. Los datos presentes en ambas líneas son estables en el flanco ascendente del Clock y cambian con el flanco descendente. El MFRC522 proporciona los datos en el flanco descendente de reloj y estos son estables durante el flanco ascendente.

En la Tabla 1 se muestra la estructura de los datos para llevar a cabo un proceso de lectura.

| 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.

Tabla 1. Orden de los bytes de las líneas MOSI y MISO durante la lectura [1].

En la Tabla 2 se muestra la estructura de los datos para llevar a cabo un proceso de escritura.

| 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.

Tabla 2. Orden de los bytes de las líneas MOSI y MISO durante la escritura [1].

Es posible leer y/o escribir hasta n bytes de datos. En ambos casos, el primer byte que se envía define el modo de operación y la dirección en donde se va a trabajar.

En la Tabla 3 se muestra el formato que debe tener el byte de dirección.

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

Tabla 3. Formato del byte de dirección [1].

Para leer datos desde el integrado MFRC522, el MSB se setea al valor lógico 1. Para escribir datos en el MFRC522, el MSB debe estar seteado al valor lógico 0. Los bits del 1 al 6 definen la dirección y el LSB se establece siempre en valor 0.

Tanto en la lectura como la escritura de datos, el MFRC522 emplea un buffer FIFO de 64 posiciones, cada una de estas con 8 bits de longitud. Esto hace que sea posible la gestión de flujo de datos de hasta 64 bytes entre el host y la máquina de estados interna del integrado sin la necesidad de tener en cuenta restricciones de tiempo.

Para acceder a este buffer se hace uso de un registro interno, denominado FIFODataReg. Al escribir en este registro, inmediatamente se almacena un byte de datos en un bloque del buffer FIFO, incrementándose en una unidad el puntero de escritura interno del buffer. Si se lee dicho registro, se muestra el contenido de una posición del buffer FIFO apuntado por el puntero de lectura interno y se decrementa en una unidad el valor de dicho puntero. La distancia que existe entre ambos punteros (lectura y escritura), se puede obtener mediante la lectura del registro FIFOLevelReg.

Los punteros del buffer FIFO se pueden reestablecer, fijando el valor del bit más significativo del registro FIFOLevelReg (denominado FlushBuffer) en el valor lógico 1. De esta manera, los bits de 0 a 6 del registro FIFOLevel (que apuntan a una posición del Buffer) son seteados a un 0 lógico, haciendo que los bytes almacenados en el buffer ya no sean accesibles, permitiendo así sobrescribirlo con nuevos datos.

En la Tabla 4 puede observarse la estructura del registro FIFOLevelReg.

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--------|-------------|----------------|---|---|---|---|---|---|
| Symbol | FlushBuffer | FIFOLevel[6:0] | | | | | | |
| Access | W | R | | | | | | |

Tabla 4. Estructura del registro FIFOLevelReg [1].

3.1.2. Acerca de las tarjetas RFID empleadas

En el presente proyecto se han empleado tarjetas MIFARE, la cual es una tecnología de tarjetas inteligentes sin contacto (TISC) de las más ampliamente utilizadas en el mundo. La distancia de lectura/escritura depende de la potencia del módulo encargado de llevar a cabo dichas tareas, siendo de unos cuantos centímetros.

MIFARE es propiedad de NXP Semiconductors. El modelo en concreto empleado es la “MIFARE Classic de 1k de memoria”. La misma está dividida en sectores, bloques y mecanismos simples de seguridad para el control de acceso, como puede observarse en la Fig. 8.

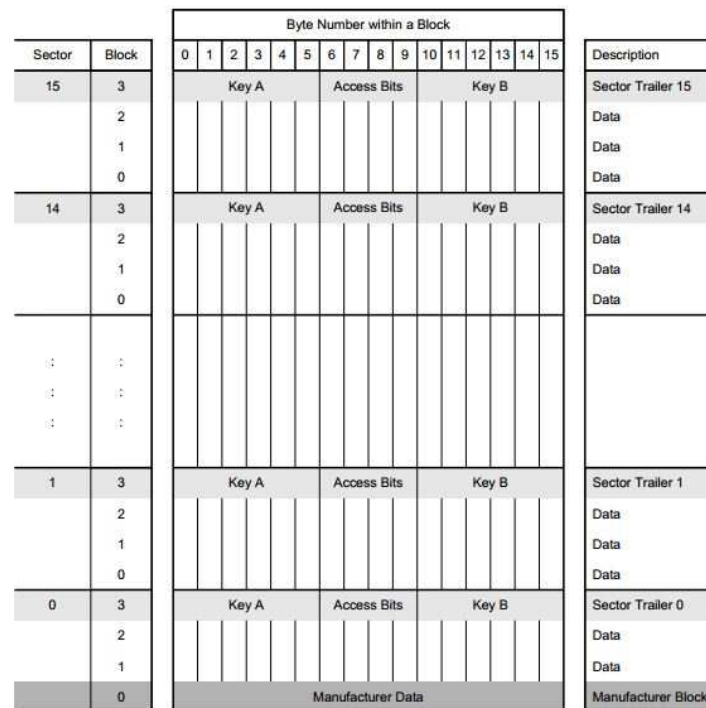


Fig. 8. Organización de la memoria [2].

Cada sector se divide en cuatro bloques, de los cuales tres pueden contener información del usuario. La información es de formato libre y se puede modificar con comandos simples de lectura y escritura. Los sectores utilizan dos claves de acceso llamadas key A y key B. Estas claves se almacenan en el cuarto bloque de cada sector junto con los permisos de acceso a cada uno de los tres bloques restantes. Estos permisos pueden ser: lectura, escritura, decremento o incremento (para bloques de valor).

Una vez que se acerca la tarjeta al módulo RFID, ésta se activa e inicia un proceso de intercambio con el mismo para establecer una comunicación cifrada. Este proceso es igual con todas las tarjetas y está diseñado para proveer protección contra “escucha” del canal y no para autenticar la tarjeta o el lector. Después de establecer un canal cifrado, la tarjeta envía un código de identificación de conexión, que usualmente es su número de serie. Con este número, el lector está en capacidad de realizar cualquier operación en la tarjeta, previa presentación de las claves de acceso a los respectivos sectores.

Estas tarjetas poseen un sistema anticolidión inteligente, el cual permite al módulo RFID operar con más de una dentro del campo de lectura de manera simultánea. Mediante dicho algoritmo, se selecciona cada tarjeta de manera individual y se asegura que la ejecución de una transacción con una tarjeta seleccionada se realiza correctamente sin la interferencia de otra en el campo operativo del módulo.

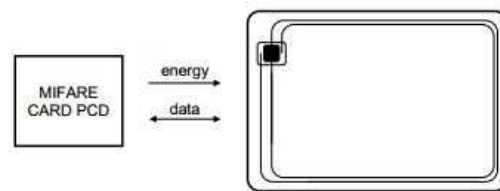


Fig. 9. Funcionamiento sistema MIFARE [2].

Los beneficios y características más destacables son:

- Transmisión de datos y energía sin contacto (no requiere una fuente interna)
- Integridad de datos con CRC de 16 bits, paridad, codificación de bits, conteo de bits
- Tiempo típico de emisión de datos menor a 100 ms
- Velocidad de transferencia de hasta 106 kbits/s
- Sistema anticolidión inteligente

En lo que respecta a la memoria EEPROM:

- Tamaño de 1 kilobyte dividida en 16 sectores de 4 bloques, donde el primer bloque del sector 0 contiene información del fabricante, el cuarto bloque de cada sector las claves de acceso para trabajar sobre los tres restantes, quedando de esta manera 47 sectores disponibles para el usuario.
- Capacidad de retención de los datos de al menos 10 años
- Soporta aproximadamente 200000 ciclos de escritura
- Las condiciones de acceso a cada bloque pueden ser establecidas por el usuario

En la Fig. 10 se puede observar un diagrama en bloques del chip empleado por el sistema de tarjetas MIFARE, el MF1S50yyX/V1 [2].

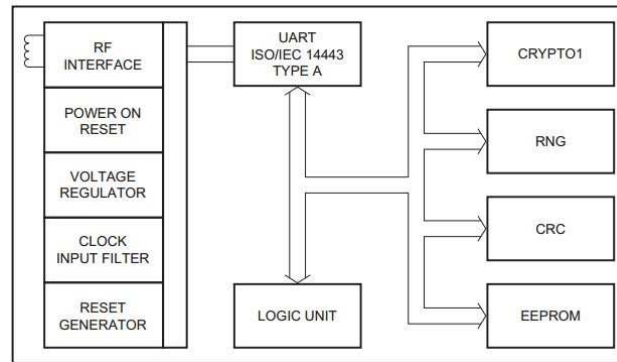


Fig. 10. Diagrama en bloques del MF1S50yyX/V1 [2].

Dicho chip, contiene la memoria EEPROM de 1k, una interfaz RF y la Unidad de Control Digital. Como se puede apreciar en la Fig. 10, la energía y los datos se transfieren mediante una antena, siendo esta una pequeña bobina con un reducido número de vueltas conectada al MF1S50yyX/V1.

3.1.3. Modo de operación del lector/grabador

Los comandos son iniciados por el lector/grabador y controlados por la Unidad de Control Digital del MF1S50yyX/V1. La respuesta al comando depende del estado del integrado de la tarjeta y las operaciones sobre la memoria de condiciones de acceso válidas para el sector correspondiente en el que se desea trabajar.

En la Fig. 11 se observa un diagrama de flujo de los comandos empleados para el uso de tarjetas MIFARE junto con sus tiempos aproximados de ejecución.

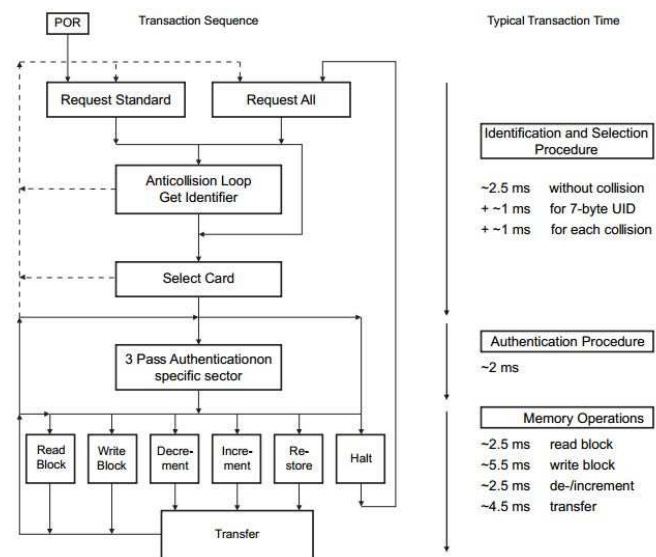


Fig. 11. Diagrama de flujo de comandos [2].

Después del POR (Power On Reset), la tarjeta responde a un comando de petición o de encendido. Luego se procede a leer el identificador (llamado UID) de la tarjeta. Si hay varias tarjetas en el campo operativo del lector/grabador, estas se pueden distinguir por su identificador, seleccionando la tarjeta necesaria para llevar a cabo las operaciones pertinentes. Las tarjetas no seleccionadas vuelven al estado inactivo y esperan un nuevo comando de solicitud.

Elegida la tarjeta, se especifica la dirección del bloque o bloques de memoria sobre los que se va a trabajar y se proporciona la clave correspondiente para un procedimiento de autenticación en tres pasos. Después de una autenticación exitosa, todos los comandos y las respuestas se cifran, pudiéndose llevar a cabo las diferentes operaciones sobre la memoria: escritura, lectura, incremento y decremento, restauración y transferencia de los valores de un bloque.

Para llevar a cabo la comunicación con la tarjeta y las distintas operaciones sobre la memoria de la misma, se emplean un conjunto de comandos, los cuales son siempre iniciados por el lector. Para ver los comandos y las direcciones de los registros internos del MF1S50yyX/V1, se recomienda ver su hoja de especificaciones [2].

En las siguientes figuras se pueden observar los tiempos y el formato de los paquetes de datos a la hora de llevar a cabo operaciones de lectura y escritura sobre la memoria de la tarjeta.

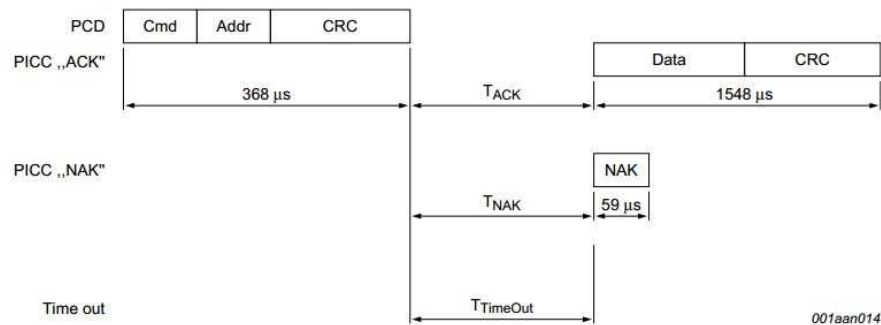


Fig. 12. Lectura de un bloque de memoria [2].

PCD hace referencia al lector/grabador y PICC a la tarjeta.

Según la Fig. 12, para iniciar la lectura de un bloque el módulo RFID envía un paquete de datos compuesto por un comando (en este caso el de lectura), la dirección del bloque a leer y el CRC para la verificación de errores. Si todo va bien, después de un tiempo T_{ACK} , la tarjeta transfiere los datos del bloque solicitado más un CRC para la verificación de errores. Después de haber realizado la transacción se espera un pequeño lapso de tiempo, en el cual, si no es solicitada la lectura de otro bloque, la tarjeta responde con un NAK. Si después de esta respuesta no hay solicitudes por parte del lector la tarjeta pasa a modo inactivo, completando la transacción.

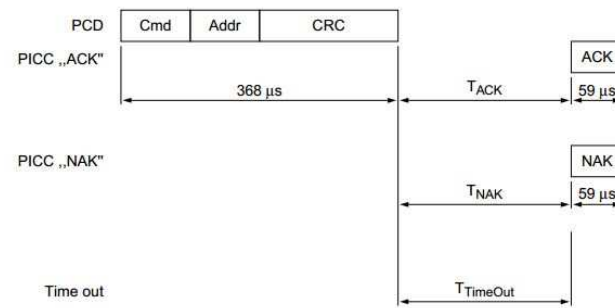


Fig. 13. Escritura de un bloque de memoria, parte 1 [2].

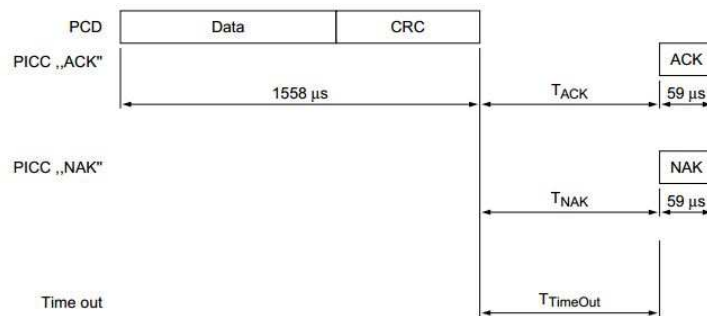


Fig. 14. Escritura de un bloque de memoria, parte 2 [2].

La Fig. 13 muestra la primera parte en un proceso de escritura sobre un bloque de memoria de una tarjeta. El primer paquete de datos enviados por el módulo, está compuesto por un comando (el de escritura), la dirección del bloque de memoria a escribir y un CRC para el control de errores. Si todo va bien, la tarjeta responde con ACK (recibido) y pasa a esperar por la llegada de los datos a almacenar en el bloque designado. En la Fig. 14 se puede observar la segunda parte del proceso de escritura. Una vez asignado el bloque a escribir, el lector envía un paquete de datos compuesto por la información más un CRC. Si todo está bien, la tarjeta responde con un ACK y pasa a esperar el siguiente comando. La ausencia de comando o peticiones por parte del módulo, genera una respuesta del tipo NAK por parte de la tarjeta y si después de esta no se registra ningún pedido más, la misma pasa a modo inactivo.

En el sistema desarrollado, el MFRC522 hace de mediador entre el microcontrolador y las tarjetas. Las operaciones de lectura/escritura son siempre iniciadas por el integrado MFRC522, el cual recibe los comandos por parte del micro a través de la interfaz SPI.

3.2. Recepción de los datos en el microcontrolador

Una vez que el módulo obtiene la información de una tarjeta determinada, la misma es enviada al microcontrolador mediante la interfaz que los vincula. En esta sección se describe dicho proceso. Para ver más detalles del microcontrolador, consulte Sección 7.

Según la sección 3.1, el protocolo de comunicación entre el microcontrolador y el MFRC522 es el SPI. El microcontrolador elegido cuenta con un periférico capaz de soportar dicho protocolo, el cual se denomina MSSP (Master Synchronous Serial Port).

En la Fig. 15, puede observarse un diagrama de conexionado entre el periférico MSSP (modo SPI) del microcontrolador y el módulo de RF.

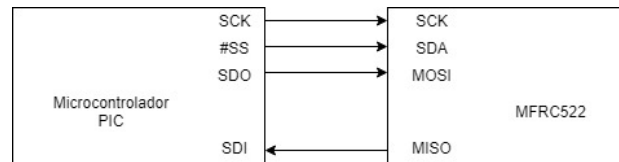


Fig.15. Diagrama de conexionado para comunicación SPI

3.2.1. Acerca del periférico MSSP (modo SPI)

El periférico MSSP proporciona una interfaz serie útil para comunicar al microcontrolador con otros dispositivos. Dicho periférico presenta dos modos de funcionamiento: modo SPI y modo I²C. En este apartado se hará hincapié en el modo SPI.

El protocolo SPI consta de tres señales, que viéndolas desde el punto de vista del microcontrolador son: SCK (señal de reloj, proporcionada por el micro), SDO (salida de datos) y SDI (entradas de datos). En la Fig. 16 se observa un diagrama en bloques del MSSP operando en modo SPI.

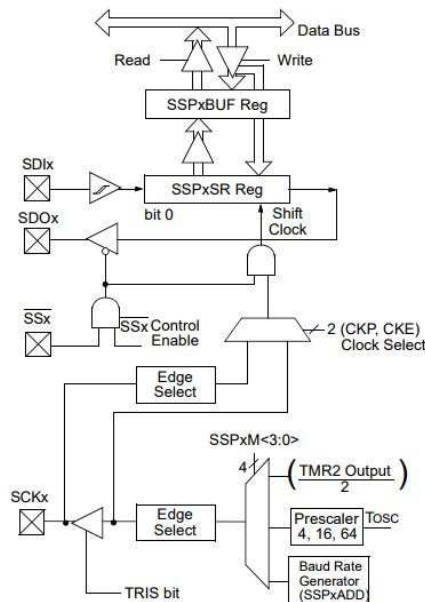


Fig. 16. Diagrama en bloques del periférico SPI [5].

Este periférico puede ser configurado como maestro o como esclavo. El periférico SPI del módulo MFRC522 sólo puede funcionar en modo esclavo, por ende, el microcontrolador toma el papel de maestro en la comunicación, siendo el encargado de proporcionar la señal del reloj y dar las órdenes. El periférico permite trabajar con datos de 8 o 16 bits. Por razones de simplicidad, el tamaño se ha fijado en 8 bits.

Los datos que se quieran enviar por SPI deben ser escritos en primera instancia en el registro SSPxBUF para luego pasar al registro SSPxSR (registro de desplazamiento), el cual los enviara al destino a través del pin SDOx del microcontrolador.

De manera similar, cuando se desea recibir datos, estos ingresaran a través del pin SDIx cargándose en el registro SSPxSR, para luego ser volcados en el registro SSPxBUF e ingresando al bus de datos del sistema.

En la Fig. 17a, se ilustra como es el proceso de escritura y lectura en el periférico SPI.

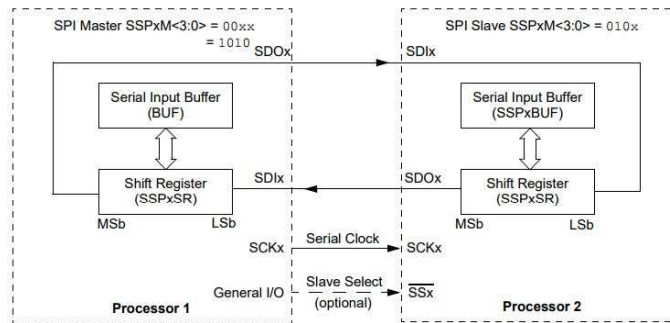


Fig. 17a. Proceso de escritura y lectura en el periférico SPI [5].

La transacción comienza cuando un dato es escrito en el registro SSPxBUF, siendo automáticamente trasladado al registro de desplazamiento SSPxSR donde es desplazado y enviado al otro dispositivo conectado a través de SDOx. Al mismo tiempo que se envía un dato, se recibe otro a través de la línea SDIx proveniente desde el otro dispositivo y luego de pasar por el registro de desplazamiento puede leerse en SSPxBUF. La señal de reloj no es una señal que corre libremente ya que sólo está disponible en los momentos que se envía/recibe un dato, siendo siempre controlada por el Master.

En el protocolo SPI no hay realmente transmisiones ni recepciones, solo intercambios de datos entre dos dispositivos, ya que por cada dato enviado debe haber siempre uno recibido, es decir, que no puedo enviar algo sin recibir nada a cambio. Dependiendo de las circunstancias, una transferencia SPI:

- Será una transmisión si el primer dispositivo tiene por objetivo enviar un dato al segundo dispositivo, mientras que el dato recibido es irrelevante (pero no puede "evitarse" recibirlo).
- Será una recepción si el dato enviado es irrelevante y solo se envía para obtener a cambio la información necesaria (no hay forma de recibir algo si no se envía nada "a cambio").
- Puede ser una transmisión/recepción simultánea si tanto el dato que se envía como el que se recibe son significativos para la comunicación.
- Finalmente, hay situaciones donde los datos intercambiados no le interesan a ningún dispositivo. (Por ejemplo, cuando se inicializa una tarjeta SD).

La línea SSx (selección de chip, CS), se emplea para habilitar y deshabilitar el MFRC522 al inicio y finalización de cada transacción.

Como se puede observar en la Fig.16, el pin SCKx está ligado a tres bloques divisores de frecuencia, los cuales son los encargados de generar la señal de reloj (Clock Rate o Bit Rate).

Según se mencionó en la sección 3.1.1, el puerto SPI del módulo de radiofrecuencia puede manejar datos a velocidades de hasta 10 Mbits/s.

El software de programación empleado permite llevar a cabo de manera conveniente, la configuración de la comunicación SPI en el microcontrolador. En la Fig. 17b, se observa la línea de programación en la cual se configura el puerto de la manera más óptima posible, en función de los parámetros del módulo RFID.

```
#use spi(MASTER, SPI2, BAUD=10000000, SAMPLE_RISE, MSB_FIRST, BITS=8, STREAM=MFRC522, FORCE_HW)
```

Fig. 17b. Configuración comunicación SPI.

El microcontrolador empleado posee dos periféricos MSSP, en donde uno de ellos está destinado a la comunicación con el MFRC522 (el MSSP2). Como puede verse en la configuración se establece el SPI2 para la comunicación entre ambos dispositivos, el microcontrolador es el Master, con una velocidad de 10Mbits/s, siendo la muestra estable en el flanco ascendente del reloj, enviando siempre el byte más significativo primero y con un tamaño del paquete de 8 bits.

4. Base de datos

La base de datos, es el lugar en donde se almacena toda la información relacionada con los usuarios y los logueos de estos. Esta se encuentra dentro de una Tarjeta microSD, la cual se comunica con el microcontrolador haciendo uso de una interfaz SPI.

4.1. Acerca de la Tarjeta microSD

Secure Digital (SD) es un dispositivo en formato de tarjeta de memoria, un medio de almacenamiento masivo con memoria del tipo no volátil que permite depositar en ella cualquier tipo de información digital. En el presente proyecto, la misma se emplea para almacenar la base de datos del dispositivo.

Uno podría preguntarse: ¿porque teniendo el microcontrolador con una memoria no volátil incorporada se recurre a una memoria externa? La respuesta es simple y se traduce a las limitaciones de tamaño que presenta, cosa que no sucede con una tarjeta de este tipo, las cuales ofrecen grandes capacidades de almacenamiento y portabilidad, una interfaz física/eléctrica sencilla y gran disponibilidad en el mercado.

En la Fig. 18 puede observarse un diagrama en bloques de la tarjeta SD y en la Fig. 19 la arquitectura de la misma.

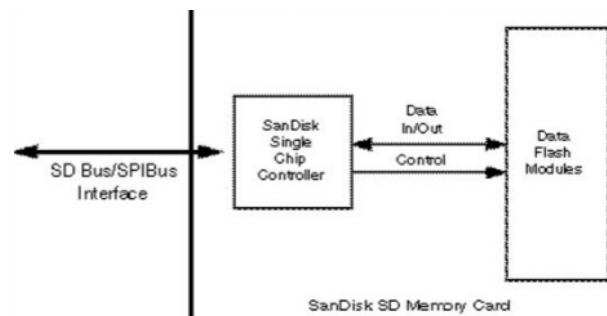


Fig. 18. Diagrama en bloques tarjeta SD [3].

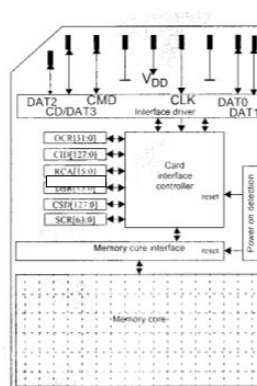


Fig. 19. Arquitectura tarjeta SD [3].

Este tipo de tarjetas pueden comunicarse con otros dispositivos mediante dos protocolos distintos: el protocolo SD y el protocolo SPI.

El primero de ellos es el más potente, ya que es el protocolo nativo con el que trabajan estas tarjetas y permite llevar a cabo operaciones de mayor complejidad y con un ancho de banda mayor (un Nibble de ancho de banda).

El segundo, podría considerarse una versión reducida del primero. Es más versátil y fácil de utilizar, ya que cualquier sistema embebido que posea un puerto SPI (como un microcontrolador) podrá comunicarse con la tarjeta, siendo suficiente para la mayoría de las aplicaciones.

4.1.1. Pinout y conexionado con el microcontrolador

En la Fig. 20, puede observarse la tarjeta microSD con sus pines de conexionado junto a una tabla en la cual se describe la función de cada uno de ellos, tanto en el modo SD como en el modo SPI.

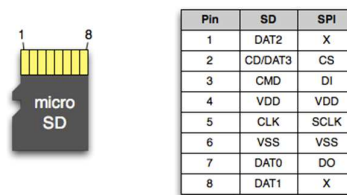


Fig. 20. Pinout de la tarjeta micro SD.

Como puede observarse, no todos los pines son empleados en el modo SPI. De los ocho, solamente seis son los utilizados. Para llevar a cabo la conexión entre el micro y la tarjeta, se adquirió un módulo adaptador con un socket para insertar la misma, el cual se muestra en la Fig. 21.

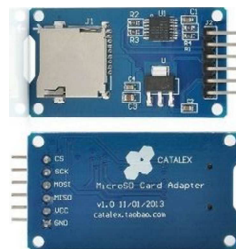


Fig. 21. Módulo adaptador para tarjeta micro SD.

La Tabla 5 muestra la relación entre los pines de la tarjeta microSD y del módulo.

| Pines Módulo | Pines microSD (modo SPI) | Tipo de Señal |
|--------------|---------------------------|------------------------------------|
| SCK | SCKL (PIN_5) | Reloj, proporcionada por el Master |
| MISO | DO (PIN_7) | Dato de salida de la tarjeta |
| MOSI | DI (PIN_3) | Dato de entrada a la tarjeta |
| CS | CS (PIN_2) | Selección de chip |
| VCC y GND | VDD (PIN_4) y GND (PIN_6) | Relacionados con la alimentación |

Tabla 5. Relación entre los pines de la tarjeta microSD y del módulo adaptador.

Estas tarjetas poseen una tensión de operación en el rango de los 2,7V a los 3,3V y en lo que respecta a la frecuencia de reloj, la misma ronda en los 25MHz para tarjetas normales y en los 50MHz para tarjetas de alta velocidad.

El módulo adaptador cuenta con un regulador de 3,3V y un adaptador de niveles de tensión. Esto es debido a que como se explicará en la Sección 9, el microcontrolador opera a una tensión de 5V, por lo tanto, sus salidas están todas a ese nivel. Entonces mediante el regulador y el adaptador todas las señales que llegan a la memoria no superan los 3,3V, asegurando un correcto funcionamiento de la misma y evitando cualquier daño permanente en esta.

En la Fig. 22, puede observarse un diagrama de conexionado entre el periférico MSSP del microcontrolador y el módulo adaptador.

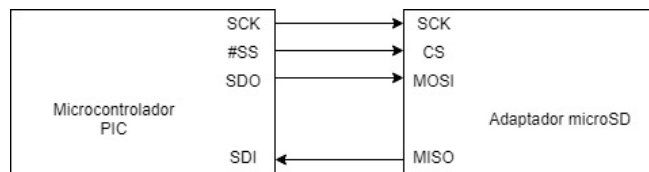


Fig. 22. Conexionado del módulo adaptador con el microcontrolador

El funcionamiento del periférico MSSP en modo SPI ya se explicó en la sección 3.2.1, por lo tanto, aquí se pasará por alto. El Master en esta comunicación es el microcontrolador, siendo el que inicia y finaliza las transacciones con la tarjeta y proporcionando la señal de reloj.

El software de programación empleado permite llevar a cabo de manera conveniente la configuración de la comunicación SPI en el microcontrolador. En la Fig. 23 se observa la línea de programación en la cual se configura el puerto de la manera más óptima posible, en función de los parámetros de la tarjeta microSD.

```
#use spi(MASTER, SPI1, BAUD=25000000, SAMPLE_RISE, MSB_FIRST, BITS=8,STREAM=SDCard, FORCE_HW)
```

Fig. 23. Configuración comunicación SPI.

El microcontrolador empleado posee dos puertos MSSP, en donde uno de ellos (MSSP2) se destina a la comunicación con el MFRC522, mientras que el otro (MSSP1) se destina para comunicación con la tarjeta microSD. Como puede verse, en la configuración se establece el SPI1 para la comunicación entre ambos dispositivos, el microcontrolador es el Master, con una velocidad de 25Mbits/s, siendo la muestra estable en el flanco ascendente del reloj, enviando siempre el byte más significativo primero y con un tamaño del paquete de 8 bits.

4.1.2. Modo de operación de la Tarjeta

La comunicación entre el PIC y tarjeta SD se basa en el envío de comandos desde el PIC y la recepción de respuestas de la tarjeta. La secuencia estándar es:

- Comando de PIC a tarjeta SD (total de 6 bytes)
- Respuesta de tarjeta SD a PIC (normalmente 1 o 2 bytes, casi siempre 1 y a veces más dependiendo del comando usado). La tarjeta siempre responde, aunque pueden llegar a necesitar hasta 8 peticiones de bytes hasta que llegue la respuesta.

- Opcionalmente un bloque de datos (en uno u otro sentido) si el comando era de escritura/lectura de datos.

El comando enviado es un código de un byte, seguido de un argumento de cuatro bytes y de un CRC de un byte, haciendo un total de seis bytes. El uso de un CRC correcto es opcional en el protocolo SPI y generalmente se ignora su valor (pero siempre hay que enviarlo). La única excepción, es cuando se desea pasar la tarjeta del modo SD al modo SPI, que es cuando la misma se reinicia. Como la tarjeta no está todavía en modo SPI, es necesario enviar un CRC válido, caso contrario no se saldrá del modo SD.

Existen varios tipos de respuesta por parte de la tarjeta, conocidas como R1, R2, R3 y R7, en donde cada una de estas depende del comando que se ha enviado. Para ver más información sobre los comandos y los tipos de respuesta se recomienda ver la hoja de especificaciones de la tarjeta [3].

En la Fig. 24 se muestra un diagrama de tiempo en el cual se le envía un comando a la tarjeta y se espera la respuesta.

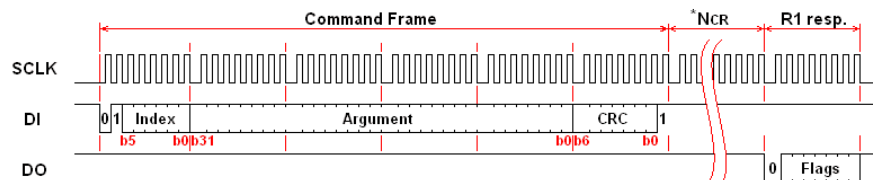


Fig. 24. Diagrama temporal de envío y respuesta

Cuando la tarjeta está lista para recibir un comando, la línea DO se pone a nivel alto. Después de enviar un comando, se devuelve una respuesta (R1, R2, R3 o R7) por parte de la tarjeta. Como el que controla la transferencia es el micro, el mismo debe mantener la señal de reloj activa y continuar leyendo la salida DO de la tarjeta hasta que se detecte una respuesta válida, al mismo tiempo que la señal DI debe mantenerse a un nivel alto durante el proceso de lectura. La respuesta por parte de la tarjeta se devuelve dentro del tiempo NCR. La señal CS debe ser conmutada de alto a bajo antes de enviar un comando y mantenida baja durante la transacción (comando, respuesta y transferencia de datos si existe).

Una vez inicializada la tarjeta y preparada para operar, es necesario poder emplear algún sistema de archivos que permita crear, modificar, leer y escribir la base de datos que se quiere implementar. Esta facultad la aporta una librería de programación, la cual permite trabajar con el sistema de archivos FAT16, por lo que previamente la tarjeta debe estar formateada en dicho sistema.

Dicha librería permite inicializar FAT16 en la tarjeta, crear directorios, archivos dentro de ellos y luego mediante determinadas funciones ubicarlos, abrirlos, leerlos y modificarlos. La misma contiene los comandos más relevantes para el uso de la memoria SD y las funciones necesarias para trabajar con archivos dentro de ella.

4.2. Acerca de la conformación de la Base de Datos

La base datos alojada dentro de la tarjeta microSD está compuesta por múltiples directorios. Uno de ellos se denomina “Registro” el cual contiene un archivo con las credenciales de acceso de los usuarios registrados, en este caso su DNI. El resto de los directorios se denominan “Historial XXXX”, donde las equis representan el año en curso. A su vez, estos poseen sub-directorios con los nombres de los meses del año, en los cuales residen dos archivos llamados “Ingresos” y “Egresos”, que clasifican como sus nombres lo dicen, los ingresos y egresos de los usuarios registrados en el corriente mes.

El microcontrolador es quien se encarga de crear automáticamente los directorios con los archivos correspondientes y de ir modificándolos en la medida que se agreguen nuevos usuarios y se produzcan los logueos.

En la Fig. 25 puede observarse los directorios creados por el micro dentro de la tarjeta SD.

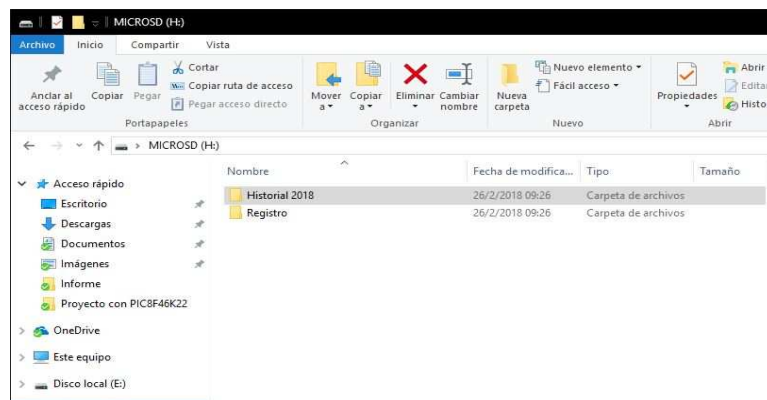


Fig. 25. Directorios de la base de datos.

Dentro del directorio “Registro”, se encuentra un archivo de texto que contiene las credenciales de accesos de todos los usuarios registrados. La credencial de cada usuario es su DNI. En la Fig. 26 puede observarse el archivo creado dentro del directorio y su contenido.

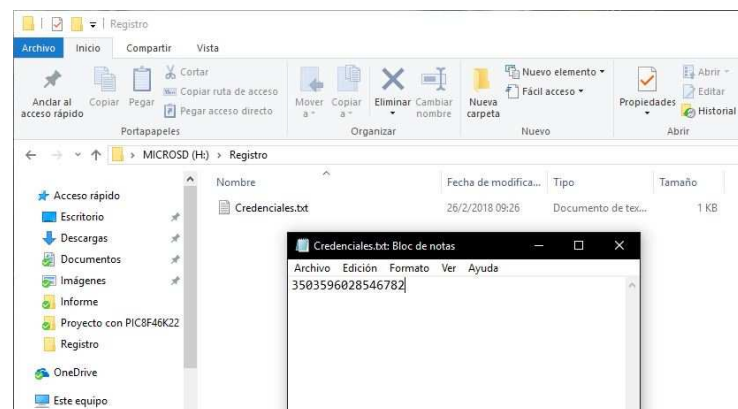


Fig. 26. Archivo de texto dentro del directorio Registro, con las credenciales de acceso.

En las Fig. 27a y Fig. 27b se muestra el contenido del directorio “Historial XXXX” y el contenido de sus sub-directorios los cuales almacenan los archivos con los ingresos y egresos correspondientes al corriente mes.

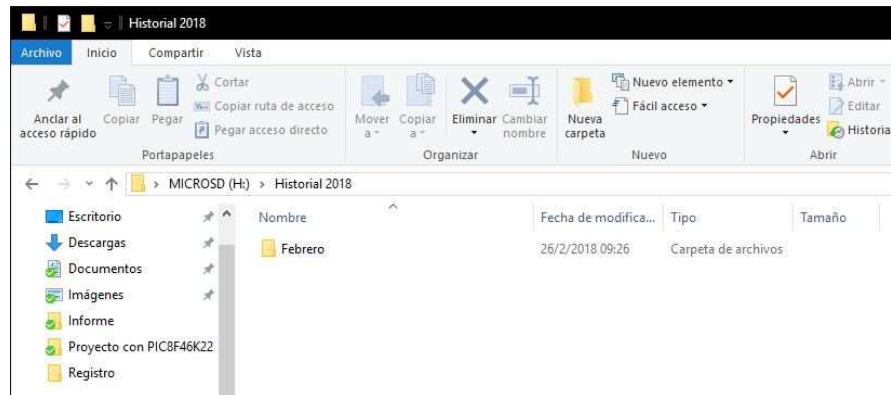


Fig. 27a. Directorio Historial y su contenido

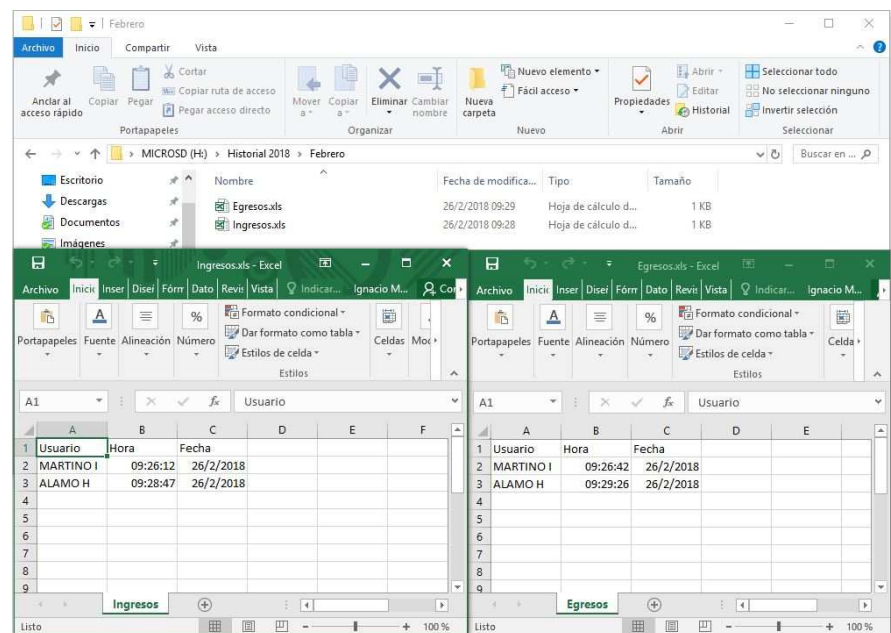


Fig. 27b. Contenido de uno de los sub-directorios del directorio Historial.

Como puede verse, en las planillas figura el nombre del usuario, la fecha y la hora a la cual se produce el logueo. El nombre de usuario está directamente vinculado con el DNI almacenado en el archivo de texto “Credenciales”.

Dentro de las tarjetas RFID se almacenan el DNI y el nombre de usuario. El DNI es comparado con la información que se tiene dentro de la base de datos y si el mismo se encuentra registrado, se toma su nombre y se carga el acceso correspondiente en la planilla de Excel. En la Sección 7 se explicará mejor el proceso de comparación.

Entre los aspectos más destacables tenemos el corazón del módulo, el DS1307 que es quien proporciona la información necesaria mediante una interfaz I²C, un cristal de 32.768KHz que funciona como oscilador del integrado, cuyo valor viene dado por el hecho de que $2^{15} = 32.768$ siendo dicha frecuencia divisible de forma exacta binariamente para generar 1000, nuestro segundo perfecto.

Cuenta también con una memoria EEPROM AT24C32 compartiendo el bus I²C con el DS1307, la cual permite almacenar información en ella, un espacio para agregar de manera opcional un sensor de temperatura DS18B20 y un zócalo para instalar una batería de respaldo, la cual se encarga de alimentar el oscilador del sistema cuando este se desconecta de la alimentación para así mantener la fecha y la hora en el DS1307 y que la misma no se pierda.

5.1.1. Acerca del DS1307

El circuito integrado DS1307 serial Real-Time Clock (RTC), es un pequeño reloj y calendario decimal codificado en binario (BCD) de bajo consumo con más 56 bytes de NVRAM, que posee una interfaz serial de comunicación bidireccional I²C. Proporciona información de segundos, minutos, horas, día, fecha, mes y año, presentando un ajuste automático para meses con menos de 31 días e incluyendo correcciones para años bisiestos. El reloj puede funcionar en formato de 24 horas o 12 horas con indicador AM/PM. Posee además un circuito de detección de energía, el cual en ausencia o fallos de alimentación cambia automáticamente a la fuente de reserva para mantener le hora. El mismo se presenta en un encapsulado 8-pines SO, como se observa en la Fig. 30.

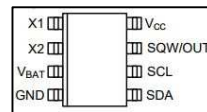


Fig. 30. Pinout DS1307 [6].

En la Fig. 31 se muestra un diagrama en bloques del DS1307.

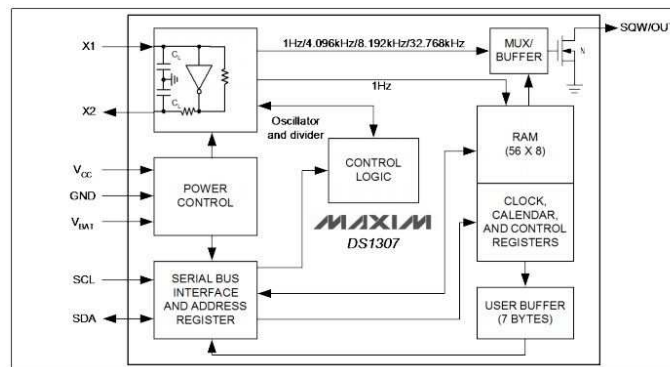


Fig. 31. Diagrama en bloques DS1307 [6].

En la Fig. 32 se muestra una tabla, la cual refleja la estructura de la NVRAM donde se mezclan los registros de configuración, aquellos utilizados para salvaguardar la fecha y hora del dispositivo y bancos de RAM para uso libre del usuario.

| ADDRESS | BIT 7 | BIT 6 | BIT 5 | BIT 4 | BIT 3 | BIT 2 | BIT 1 | BIT 0 | FUNCTION | RANGE |
|---------|---------|------------|------------|-------------|---------|-------|-------|---------|-------------------------|---------|
| 00h | CH | 10 Seconds | | | Seconds | | | Seconds | 00-59 | |
| 01h | 0 | 10 Minutes | | | Minutes | | | Minutes | 00-59 | |
| 02h | 0 | 12 | 10 Hour | 10 Hour | Hours | | | Hours | 1-12 +AM/PM 00-23 | |
| | | 24 | PM/ AM | | | | | | | |
| 03h | 0 | 0 | 0 | 0 | DAY | | | Day | 01-07 | |
| 04h | 0 | 0 | 10 Date | | Date | | | Date | 01-31 | |
| 05h | 0 | 0 | 0 | 10 Month | Month | | | Month | 01-12 | |
| 06h | 10 Year | | | | Year | | | Year | 00-99 | |
| 07h | OUT | 0 | 0 | SQWE | 0 | 0 | RS1 | RS0 | Control | — |
| 08h-3Fh | | | | | | | | | RAM 56 x 8 | 00h-FFh |

0 = Always reads back as 0.

Fig. 32. Estructura de la NVRAM del DS1307 [6].

El bit 7 del registro 00h es quien determina si el CI está habilitado o no. Para que el DS1307 funcione, este bit debe estar a valor 0.

El registro 07h es el registro de control del DS1307, el cual permite configurar la función del pin de salida (SQW/OUT). El bit 4 (SQWE) habilita o deshabilita la función de salida externa de dicho pin. El bit 7 (OUT) establece el estado del pin de salida cuando SQWE está deshabilitado. Si OUT es 1 y SQWE es 0 entonces el pin de salida está en alto indefinidamente, si OUT es 0 y SQWE es 0 entonces el pin de salida está por el contrario en bajo indefinidamente. Los bits 0 (RS0) y 1 (RS1) sirven para seleccionar la frecuencia de la señal cuadrada de salida cuando SQWE está en alto según la tabla expuesta en la Fig. 33.

| RS1 | RS0 | SQW/OUT OUTPUT | SQWE | OUT |
|-----|-----|----------------|------|-----|
| 0 | 0 | 1Hz | 1 | X |
| 0 | 1 | 4.096kHz | 1 | X |
| 1 | 0 | 8.192kHz | 1 | X |
| 1 | 1 | 32.768kHz | 1 | X |
| X | X | 0 | 0 | 0 |
| X | X | 1 | 0 | 1 |

Fig. 33. Función del pin SQW/OUT según los bits del registro de control [6].

Configurar correctamente el valor del pin de salida del DS1307 es importante, ya que cuando el mismo se queda sin suministro eléctrico el consumo de la batería de respaldo es mucho mayor, debido a que se requiere de una energía adicional para seguir entregando dicha salida y por consiguiente llevando a un agotamiento más rápido de la vida útil de la batería. De no ser necesaria, la salida se establece a nivel bajo.

5.2. Conexión y comunicación con el microcontrolador

En la Fig. 34 se muestra un diagrama de conexión entre el microcontrolador y el módulo RTC.

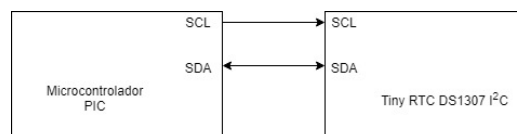


Fig. 34. Conexión entre microcontrolador y módulo.

Como se viene mencionando, la comunicación entre ambos se lleva a cabo mediante el protocolo I²C (Inter-Integrated Circuit) el cual es un protocolo serial simple, de carácter bidireccional que hace uso de solo dos hilos para la comunicación, uno destinado el reloj el cual es controlado por el master que inicia la comunicación y el otro para el intercambio de datos entre los dispositivos conectados.

El periférico capaz de soportar este protocolo de comunicación por parte del micro ya se mencionó en la Sección 3.2.1, dicho periférico es el MSSP. Ahora bien, el microcontrolador seleccionado solo dispone de dos de ellos, los cuales fueron empleados para establecer las comunicaciones con el MFRC522 y la tarjeta microSD, no quedando ninguno disponible para llevar a cabo la comunicación con el DS1307.

Para solventar este problema, el software de programación empleado permite implementar el protocolo I²C mediante software, es decir, le da la facultad al microcontrolador mediante una serie de rutinas de simular el comportamiento del periférico MSSP en modo I²C, utilizando dos pines de manera arbitraria como reloj y salida/entrada de datos para de esta manera entablar la comunicación con el módulo RTC.

5.2.1. Acerca del periférico MSSP en modo I²C

En la Fig. 35 se muestra un diagrama en bloques del periférico MSSP configurado en modo I²C.

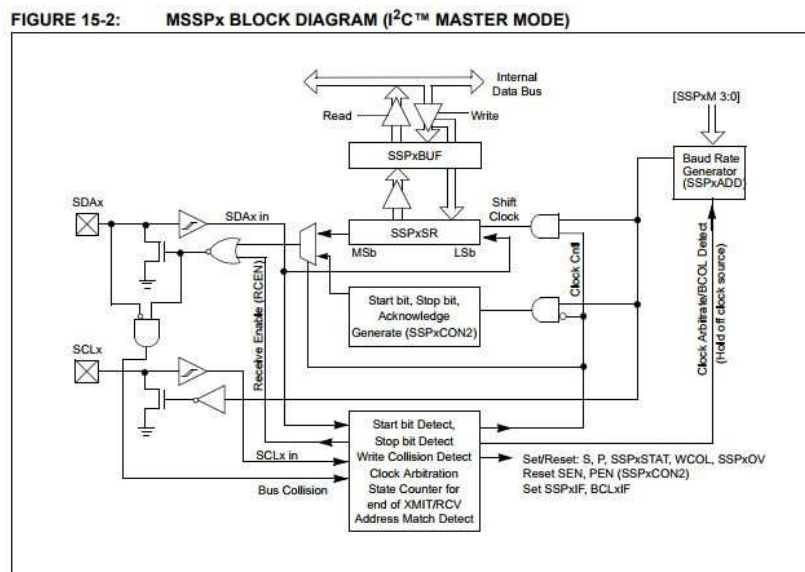


Fig. 35. Periférico MSSP en modo I²C [5].

Es un protocolo de comunicación Multi-Master de solo dos líneas, el cual permite la comunicación entre varios dispositivos maestros/esclavos, siendo iniciada siempre la transacción por un maestro. Los dispositivos esclavos son controlados mediante la asignación de una dirección y solo puede haber comunicación entre un maestro y esclavo a la vez.

La metodología de comunicación de datos del bus I²C es en serie y sincrónica. Una de las señales del bus marca el tiempo (pulsos de reloj-SCL) y la otra se utiliza para intercambiar datos (SDA).

En la Fig. 36 se muestra una conexión típica entre una maestro y esclavo.

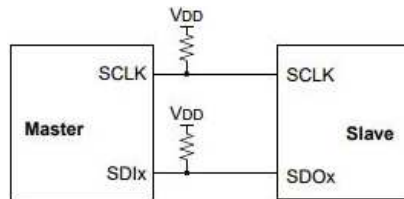


Fig. 36. Conexión entre maestro y esclavo [5].

Las líneas SDA y SCL son del tipo drenaje abierto, debiendo ser polarizadas a nivel alto mediante resistencias pull-up. De esta manera cuando el bus está inactivo las dos líneas se encuentran siempre a nivel alto, condición conocida como “Bus Libre”. En este estado, cualquier Master puede ocupar el bus estableciendo la condición de “Inicio”. Dicha condición se da cuando el master solicitante pone a nivel bajo la línea SDA dejando alta la línea SCL, como se ve en la Fig. 37.

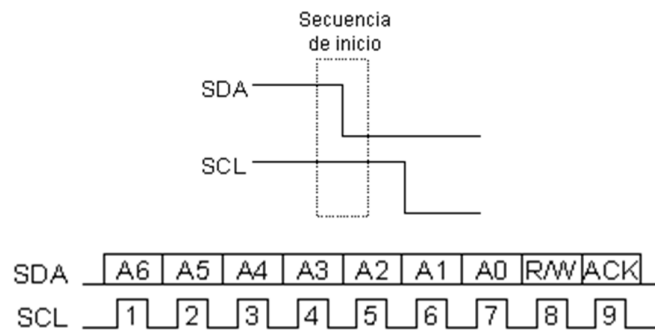


Fig. 37. Secuencia de inicio.

El primer byte que se transmite luego de la condición de inicio contiene siete bits que componen la dirección del dispositivo con el que se desea comunicar y un octavo bit que corresponde a la operación que se quiere realizar con él (lectura o escritura).

Si el dispositivo cuya dirección corresponde a la que se indica en los siete bits (A0-A6) está presente en el bus éste contesta con un bit en bajo, ubicado inmediatamente después del octavo bit que ha enviado el maestro. Este bit de reconocimiento (ACK) en bajo le indica al dispositivo maestro que el esclavo reconoce la solicitud y está en condiciones de comunicarse y comenzar con el intercambio de información.

Si el bit de R/W (lectura/escritura) fue puesto a nivel bajo (escritura), el dispositivo maestro envía datos al dispositivo esclavo. Esto se mantiene mientras continúe recibiendo señales de reconocimiento y finaliza cuando se hayan transmitido todos los datos.

Si el bit R/W fue puesto a nivel alto (lectura), el dispositivo maestro genera pulsos de reloj para que el dispositivo esclavo pueda enviar los datos. Luego de cada byte recibido el dispositivo maestro genera un pulso de “Reconocimiento”.

El maestro puede dejar libre el bus generando una condición de “Parada”. Si se desea seguir transmitiendo, el maestro puede generar otra condición de inicio. Esta nueva condición de inicio se denomina “Inicio Reiterado” y se puede emplear para direccionar un dispositivo esclavo diferente o para alterar el estado del bit de R/W.

Los datos se transmiten en secuencias de ocho bits. Estas se cargan en el registro de desplazamiento SSPxSR, siempre con el bit más significativo primero para luego ser enviadas mediante la línea SDA al dispositivo destino. Cargado un bit, la línea SCL se pone a nivel alto. Por cada byte enviado hay nueve pulsos de reloj, uno por cada bit de datos y el noveno corresponde al bit de reconocimiento enviado por el esclavo. Si este último es un nivel alto, todo está bien, pero si es un nivel bajo le indica al master que no puede seguir recibiendo datos y que debe finalizar la transacción mediante una condición de parada.

Las secuencias de inicio y la de parada son especiales porque son los dos únicos casos en que se permite que la línea SDA cambie cuando la línea de SCL está en nivel alto. Cuando se están transmitiendo datos, la línea SDA debe permanecer siempre estable mientras la línea SCL está alta.

En el presente proyecto, la comunicación es solo entre dos dispositivos: el microcontrolador y el módulo RTC, por lo tanto hay un solo maestro y un solo esclavo. Para controlar el módulo, se hace uso de una librería diseñada para dicho propósito.

En las Fig. 38 y Fig. 39 se muestra un diagrama temporal de escritura y lectura por parte del master, en este caso el microcontrolador, sobre el esclavo.

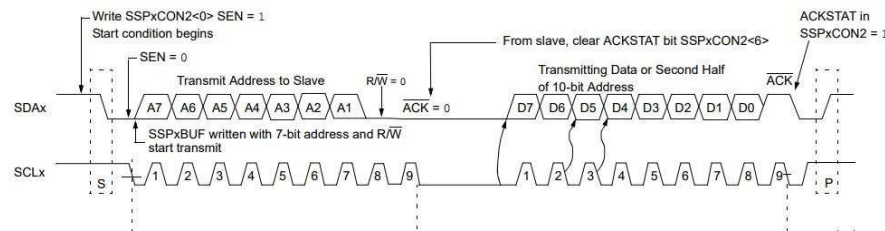


Fig. 38. Diagrama temporal de escritura [5].

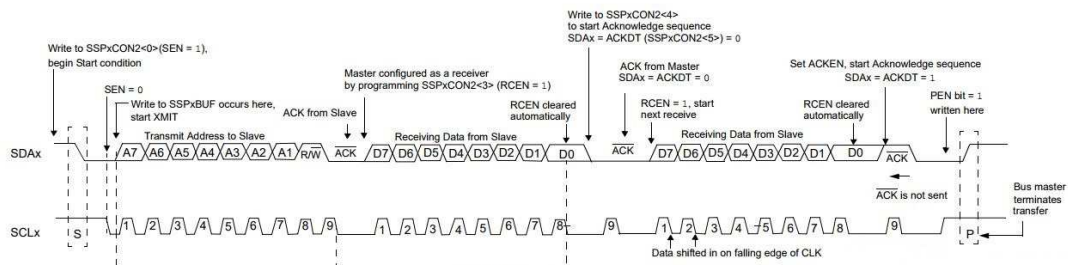


Fig. 39. Diagrama temporal de lectura [5].

Según la hoja de especificaciones del DS1307, la frecuencia optima de reloj para la comunicación con el master debe ser menor o igual a 100KHz. Mediante el software de programación, se configuran los parámetros necesarios para establecer una comunicación optima

entre ambos dispositivos. En la Fig. 40 se muestran las líneas de programa en la cual se configura la comunicación I²C en el microcontrolador.

```
#ifndef RTC_SDA
#define RTC_SDA PIN_C7
#define RTC_SCL PIN_C6
#endif

#use I2C(MASTER, SDA=RTC_SDA, SCL=RTC_SCL, FAST=100000, FORCE_SW)
```

Fig. 40. Configuración comunicación I²C.

Se definen los pines C6 y C7 del microcontrolador como las señales SCL (reloj) y SDA (datos) respectivamente, el micro es el master en la comunicación, la sentencia FAST=100000 determina una velocidad de reloj máxima de 100KHz de acuerdo con la hoja de datos del DS1307 para lograr la mayor velocidad y estabilidad posible y la sentencia FORCE_SW determina que la comunicación sea implementada vía software por las razones explicadas en la sección 5.2.

6. Sistema De Accionamiento

El sistema de accionamiento está compuesto por un transistor trabajando en corte-saturación el cual controla un relé que se encarga de mantener cerrada una cerradura electromagnética o abrirla en caso de que sea necesario. Esta etapa está implementada parcialmente, es decir, el dispositivo no cuenta con una cerradura electromagnética integrada en él por cuestiones de costos.

Quedando de esta manera implementada para un uso futuro, con los conectores a la vista para poder conectar la cerradura y hacer uso de ella. La etapa posee la posibilidad de ser deshabilitada por hardware, no consumiendo energía adicional en caso de que no se utilice.

6.1. Acerca del circuito de accionamiento

Como se dijo al principio de la sección, está compuesta por un relé comandado por un transistor trabajando en corte y saturación. El transistor empleado es del tipo NPN, siendo este un el BC548 al cual trabaja en su región de corte y saturación haciendo que el mismo funcione como una llave electrónica conmutando el estado del relé de normal cerrado a normal abierto cuando sea necesario.

El relé utilizado es de la empresa Songle, más concretamente el SRD-12VDC-SL-C, el cual posee una tensión de accionamiento de 12V y un consumo de 30mA en su bobina. La bobina del mismo se conecta directamente a la fuente de alimentación de 12V que posee el sistema (ver Sección 9: Alimentación) y su terminal normal cerrado también, entregando siempre 12V para energizar la cerradura electromagnética cuando esta permanece cerrada. En la Fig. 41 se muestra el circuito de la sección de accionamiento.

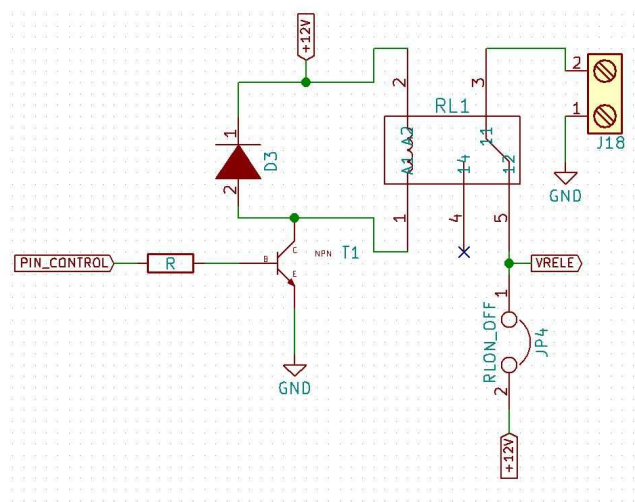


Fig. 41. Esquema del control de la cerradura.

En caso de que se produzca algún acierto al pasar una tarjeta por el lector RFID, el microcontrolador excita la base del transistor que comanda el relé, saturándolo y cerrando el camino de la bobina, haciendo que este conmute de normal cerrado a normal abierto dejando de esta manera de alimentar la cerradura electromagnética y abriéndola por un pequeño lapso de tiempo.

El diodo que está conectado inversamente polarizado en paralelo con la bobina del relé es de protección y se encargar de absorber los picos de tensión cuando se interrumpe el paso de la corriente por la bobina al pasar a estado de corte el transistor, ya que cuando esto sucede el campo magnético presente en el inductor genera por un breve periodo de tiempo una tensión opuesta de alto valor conocida como tensión extra de apertura, la cual puede dañar el transistor de manera permanente.

La resistencia en la base del transistor es para limitar la corriente que el mismo consume, ya que esta debe ser de un valor pequeño y adecuado para que opere en la zona de corte y saturación. Dicha resistencia se conecta a un pin del microcontrolador, que es quien proporciona la señal de accionamiento.

Y el jumper conectado entre los 12V y el terminal normal cerrado del relé, se emplea para habilitar o deshabilitar la tensión de salida que este entrega cuando el circuito de control de la cerradura no se emplee.

7. Microcontrolador

Hasta el momento se ha hablado del microcontrolador, pero no se ha especificado el mismo. En la búsqueda del microcontrolador que mejor se adapte a los requerimientos impuestos por el presente proyecto, se ha tenido en cuenta que el mismo posea al menos dos periféricos MSSP, ya que debe comunicarse con varios dispositivos al mismo tiempo, que trabaje a frecuencias altas de manera tal que se ajuste a las necesidades de los periféricos, que cuente con un elevado número de pines para el conexionado de los dispositivos necesarios, etc. Se eligió el microcontrolador PIC18F46K22 de la empresa Microchip Technology Inc. [5], el cual es un micro de 8 bits, posee una memoria de programa de 64 KB, 3896 Bytes de RAM, cuenta con los periféricos necesarios para el proyecto y viene en un encapsulado 40-pin PDIP.

De manera resumida, podría decirse que la tarea del microcontrolador es la de recibir los datos leídos de las tarjetas provenientes del módulo RFID a través de un periférico SPI, almacenarlos en RAM y luego compararlos con la información que se encuentra en la base de datos del sistema, la cual se almacena en la tarjeta microSD que se comunica con el microcontrolador mediante otro periférico SPI para de esta manera determinar si corresponde o no el acceso, dejando plasmado en la misma base de datos un registro histórico de los mismos, obteniendo la fecha y hora de un circuito integrado (el DS1307) con el que se comunica mediante el protocolo I²C y además conmutar el estado de la cerradura electrónica en caso de que la misma se encuentre implementada.

También mediante una interfaz de usuario bajo PC, se encarga de iniciar los procesos de grabación de nuevas tarjetas para de esta manera agregar nuevos usuarios al sistema, como así también el borrado de las mismas y la completa gestión del dispositivo.

Durante todas las operaciones el micro muestra mediante un LCD los sucesos de mayor relevancia, diferentes mensajes, etc.

7.1. Comparación de la información

Como se vio en la sección 3.1.1, el módulo RFID dispone de un buffer FIFO, en el cual almacena de manera temporal tanto en lecturas de tarjetas como escritura de las mismas, los datos leídos o que se van a escribir. El tamaño del buffer permite almacenar los datos de un sector completo de la memoria de una tarjeta (un total de 64 bytes).

De esta manera, cuando se lleva a cabo una lectura para obtener la información deseada, los datos obtenidos de los bloques leídos se almacenan en estas memorias “intermedias” hasta que se haya completado la lectura o en su defecto, hasta que se llenen si el proceso involucra a más de un sector completo.

Suponiendo que la información requerida no sobrepasa la capacidad del Buffer, al finalizar la lectura toda la información es enviada al microcontrolador sin ningún tipo de restricción de tiempo a través del puerto SPI.

En el presente proyecto únicamente se emplean dos bloques de un sector determinado (en los cuales se almacenan los datos del usuario: su nombre y DNI), es decir que la cantidad de información a extraer de las tarjetas no supera los 32 bytes, no desbordando en ningún caso el Buffer.

Una vez extraída la información por parte del módulo RFID, este la envía al microcontrolador el cual la almacena en dos vectores de 16 bytes de tamaño cada uno, para luego

proceder a llevar a cabo la comparación con la información que hay en la base de datos, para así determinar si el usuario se encuentra registrado o no en el sistema.

7.1.1. Acerca del oscilador del sistema del microcontrolador

En este apartado se verá la configuración del oscilador interno del microcontrolador, el cual presente una frecuencia de 64MHz.

En la Fig. 42 se observa un diagrama del oscilador interno del PIC18F46K22.

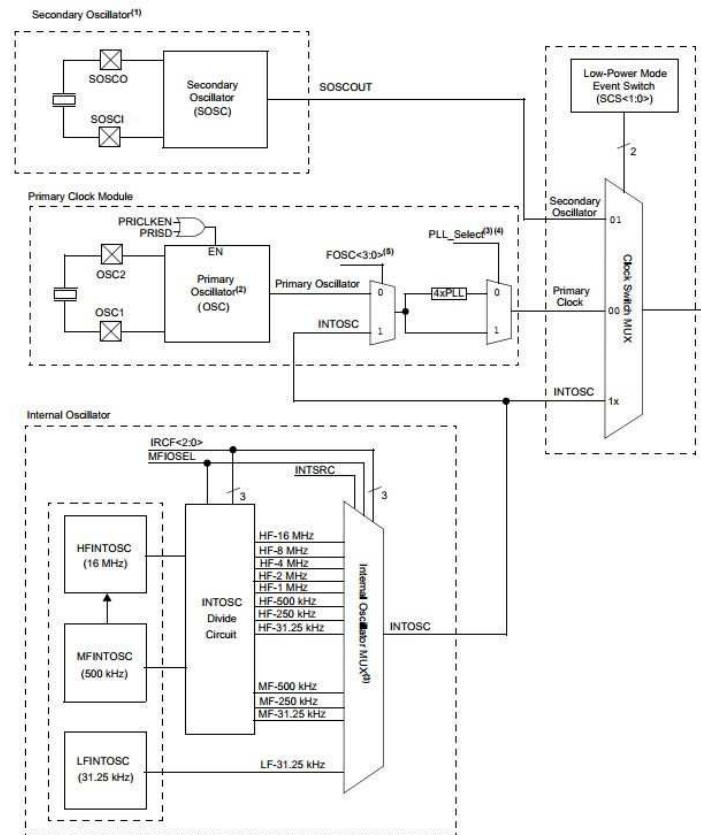


Fig. 42. Diagrama del oscilador interno del PIC18F46K22 [5].

De todas las fuentes que pueden seleccionarse para generar la frecuencia de Clock, se eligió el oscilador interno del microcontrolador, ya que mediante el mismo se puede lograr la frecuencia deseada prescindiendo de componentes externos como cristales, por ejemplo.

En la Fig. 43 se observa el valor de los bits de configuración del oscilador interno.

```

IRCF<2:0>: Internal RC Oscillator Frequency Select bits(2)
111 = HFINTOSC – (16 MHz)
110 = HFINTOSC/2 – (8 MHz)
101 = HFINTOSC/4 – (4 MHz)
100 = HFINTOSC/8 – (2 MHz)
011 = HFINTOSC/16 – (1 MHz)(3)

If INTSRC = 0 and MFIOSEL = 0:
010 = HFINTOSC/32 – (500 kHz)
001 = HFINTOSC/64 – (250 kHz)
000 = LFINTOSC – (31.25 kHz)

If INTSRC = 1 and MFIOSEL = 0:
010 = HFINTOSC/32 – (500 kHz)
001 = HFINTOSC/64 – (250 kHz)
000 = HFINTOSC/512 – (31.25 kHz)

If INTSRC = 0 and MFIOSEL = 1:
010 = MFINTOSC – (500 kHz)
001 = MFINTOSC/2 – (250 kHz)
000 = LFINTOSC – (31.25 kHz)

If INTSRC = 1 and MFIOSEL = 1:
010 = MFINTOSC – (500 kHz)
001 = MFINTOSC/2 – (250 kHz)
000 = MFINTOSC/16 – (31.25 kHz)

```

Fig. 43. Bits de configuración del oscilador interno del PIC18F46K22 [5].

El oscilador interno, proporciona tres frecuencias base, que son:

- Una de 16MHz, high frequency internal oscillator (HFINTOSC).
- Una de 500kHz, medium frequency internal oscillator (MFINTOSC).
- Y una de 31,25kHz, low frequency internal oscillator (LFINTOSC).

Para lograr los 64MHz, se selecciona la fuente más grande mediante los bits IRCF. Dicha fuente ingresa al bloque divisor, donde haciendo uso de los bits MFIOSEL e INTSRC se pueden obtener hasta en ocho valores distintos, logrando de esta manera ocho frecuencias base diferentes, acorde a las necesidades del usuario.

Ahora bien, el divisor se fija en uno ya que no nos interesa obtener frecuencias menores, de manera tal que a la salida del bloque del oscilador interno tendremos 16MHz.

Luego, mediante la habilitación del bloque PLL la frecuencia base es de llevada a 64MHz, debido a que el PLL multiplica x4 la frecuencia anteriormente generada. Finalmente, la salida del PLL es quien entrega la frecuencia de operación del microcontrolador.

La Ecu. 1, permite calcular el valor de la frecuencia final obtenida.

$$Clock_{System} = \frac{HFINTOSC}{1} * PLL = 16MHz * 4 = \mathbf{64MHz} \quad \text{Ecu. 1}$$

7.1.2. Acerca de la programación

En la Fig. 44 se observa un diagrama de flujo del programa principal que corre el microcontrolador. Debajo de dicha figura se encuentra una breve explicación de lo que hace cada bloque.



Fig. 44. Diagrama de flujo Programa Principal.

El bloque **CONFIG. MICROCONTROLADOR** hace referencia a la configuración inicial que se hace al propio micro para setear características propias tales como: palabra de configuración, declaración de librerías y variables, configuración del oscilador del sistema y configuración de puertos.

Una vez configurado el micro se procede a configurar mediante el bloque **CONFIG. COM SERIAL** una comunicación serial, la cual permite establecer una conexión con un ordenador y tiene prioridad para interrumpir el curso normal del programa cuando sea necesario. Mediante la misma y una interfaz gráfica de usuario creada, se llevan a cabo la carga de nuevos usuarios, borrado de tarjetas, seteo de fecha y hora del dispositivo, etc.

Mediante el bloque **INI. LCD** se inicializa el LCD en el cual se mostrarán distintos mensajes en función de los estados en los que se encuentre el sistema.

En el bloque **CONFIG SPI E I2C** se configuran los periféricos MSSP para establecer comunicaciones mediante los protocolos SPI e I²C. Luego se inicializa la tarjeta SD y el módulo RTC mediante el bloque **INI. SD Y DS1307** y se lleva a cabo la escritura de los registros internos del módulo RFID a través del bloque **CONFIG. MFRC522** para de esta manera configurarlo e inicializarlo.

Configurado e inicializado el MFRC522, el sistema queda a la espera de una tarjeta. La tarea del bloque de pregunta **HAY ALGUNA TARJETA?**, es la de detectar la presencia de algún TAG RFID en las proximidades del lector. En caso afirmativo se lee la tarjeta y se extrae la información en ella almacenada y se pasa al bloque **ESTA EL USUARIO REGISTRADO?**, en donde lo que se hace es buscar y comparar con la base de datos del sistema la info obtenida para de esta manera determinar si el usuario está registrado.

De estarlo, se extrae del módulo RTC la fecha y hora en que se produce el acceso y luego se registra el hecho en la base de datos, concediendo acceso al usuario.

De no estarlo, se deniega el acceso y el sistema queda a la espera de una nueva tarjeta, repitiendo el ciclo.

El programa ingresa a la rutina de interrupción cuando se recibe alguna petición por parte de la comunicación serial. Una vez dentro de esta, dependiendo del comando que se reciba, se llevara a cabo una tarea u otra. En la Fig. 45 se observa un diagrama de flujo de la sub-rutina de interrupción.

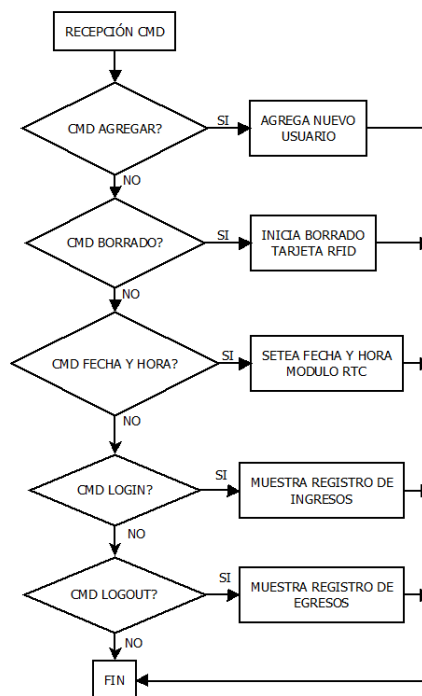


Fig. 45. Diagrama de flujo Subrutina.

En el bloque **RECEPCIÓN CMD** se lee el comando recibido, el cual es enviado por el operador mediante la interfaz gráfica. Luego mediante los bloques de pregunta se procede a determinar cuál ha sido el comando. El bloque **CMD AGREGAR?**, pregunta si el comando

recibido es para cargar un nuevo usuario, tanto en la tarjeta como en el sistema. En caso afirmativo, le solicita al operador que ingrese las credenciales del nuevo usuario y luego pase la tarjeta por el lector/grabador para de esta manera escribirla y cargar los datos en la base del sistema, finalizando la rutina de interrupción.

En caso negativo pasa al bloque **CMD BORRADO?**, que pregunta si el comando recibido es para llevar a cabo el borrado de una tarjeta. En caso afirmativo, solicita sostener la tarjeta sobre el lector/grabador dando comienzo al blanqueo de la misma y finalizando la rutina de interrupción.

En caso negativo pasa al bloque **CMD FECHA Y HORA?**, el cual pregunta si el comando recibido es para setear la fecha y la hora del módulo RTC. De ser así, se fija la hora y fecha proporcionada por el operador finalizando la rutina de interrupción.

En caso negativo pasa al bloque **CMD LOGIN?**, que pregunta si el comando recibido es para extraer el contenido del archivo con los registros de los ingresos del corriente mes. De ser así, se transfiere dicha información y se la carga en una planilla de Excel que la interfaz abre de manera automática, finalizando la rutina de interrupción.

En caso negativo pasa al bloque **CMD LOGOUT?**, el cual pregunta si el comando recibido es para extraer el contenido del archivo con los registros de los egresos del corriente mes. De ser así, se transfiere dicha información y se la carga en una planilla de Excel que la interfaz abre de manera automática, finalizando la rutina de interrupción.

Finalizada la subrutina, el microcontrolador retoma el punto del programa en el cual lo abandono antes de la interrupción, siguiendo con el curso normal del mismo.

8. Gestión de las Tarjetas RFID

Tanto como para cargar nuevos usuarios en las tarjetas y la base de datos del dispositivo como para borrar el contenido de las tarjetas, se diseñó una interfaz gráfica de usuario simple bajo PC. Esta interfaz se comunica con el microcontrolador haciendo uso del estándar de comunicación RS-232.

Para poder conectar el microcontrolador con la PC se emplea un módulo conversor “UART TTL a USB” el cual contiene un pequeño integrado, el CP2102 de la empresa Silicon Laboratories [7], que como dice su hoja de especificaciones es un puente entre el puerto USB de una PC y el puerto UART del microcontrolador. Con esto se logra que tanto la PC como el microcontrolador “hablen” el mismo idioma. Se emplea este puente para la comunicación, debido que el microcontrolador utilizado no cuenta con un módulo USB nativo para conectarlo directamente con la PC e implementar la comunicación.

8.1. Acerca del módulo conversor

En la Fig. 46 se muestra diagrama eléctrico del módulo conversor.

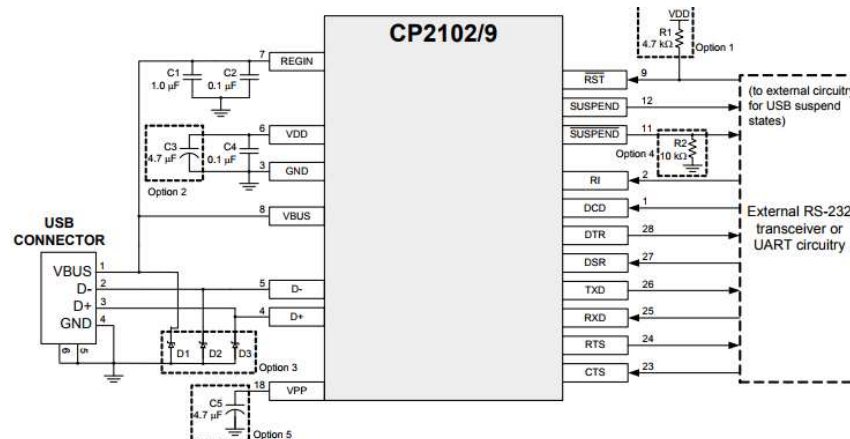


Fig. 46. Diagrama eléctrico del módulo conversor [7].

En la Fig. 47 se muestra una fotografía del módulo.



Fig. 47. Fotografía del módulo conversor.

El corazón del módulo es el integrado CP2102, que como se menciono es un puente USB-UART que permite actualizar diseños RS-232 a USB utilizando un mínimo de componentes y espacio. El CP2102 incluye un controlador USB 2.0, un transeptor USB, un oscilador, una

memoria EEPROM y un bus de datos serie asíncrono (UART). En la Fig. 48 se muestra un diagrama en bloques del integrado en cuestión.

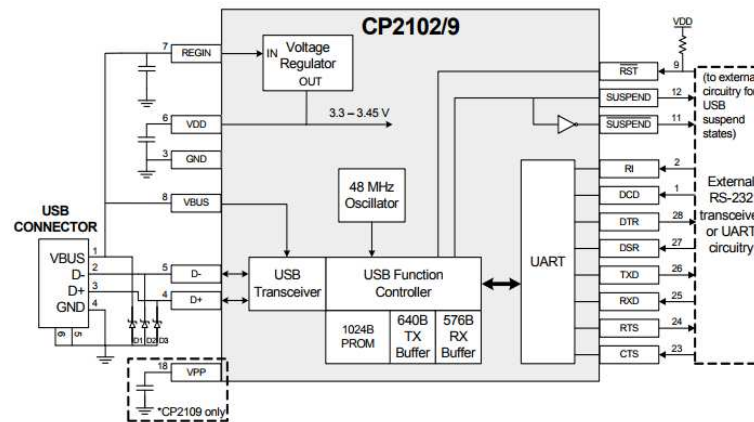


Fig. 48. Diagrama en bloques del CP2102 [7].

El fabricante del integrado proporciona un controlador que una vez instalado en la PC, genera un puerto COM virtual el cual es asignado al puerto USB en el cual se encuentra conectado el módulo. De esta manera configurando ciertos parámetros tanto en el microcontrolador como en la PC, se puede establecer una comunicación serial simple entre ambos dispositivos haciendo uso del estándar RS-232.

8.1.1. Conexión y comunicación con el microcontrolador

En la Fig. 49 se muestra un diagrama de conexión entre el microcontrolador y el módulo conversor, en donde los nombres de los pines son nombres que aparecen en las hojas de especificaciones de los componentes. Como puede observarse, se emplean únicamente dos líneas para la comunicación, una de transmisión y otra de recepción.

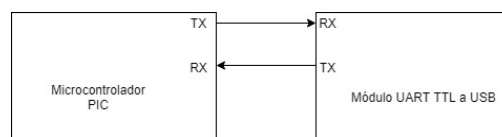


Fig. 49. Conexión entre microcontrolador y módulo.

Como se mencionó, la comunicación se lleva a cabo haciendo uso del estándar RS-232, el cual es una interfaz que designa una norma para el intercambio de información binaria entre dos dispositivos. En general, las cadenas de datos son caracteres ASCII los cuales incluyen los códigos de letras, números y signos de puntuación, además de caracteres especiales. Se trata de un estándar orientado a la transmisión de texto.

El microcontrolador cuenta con un módulo capaz de soportar dicho estándar, conocido con el nombre de EUSART (Enhanced Universal Synchronous Asynchronous Receiver Transmitter) o Transmisor-Receptor Síncrono-Asíncrono Universal Mejorado, el cual es un periférico de comunicación serial de E/S.

8.1.2. Acerca del módulo EUSART

En las Fig. 50 y 51, se muestran los diagramas en bloques del módulo EUSART tanto para la transmisión como para la recepción de datos.

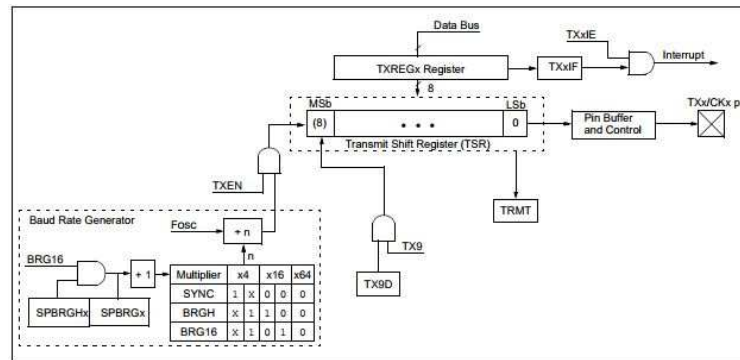


Fig. 50. Diagrama en bloques EUSART para la transmisión [5].

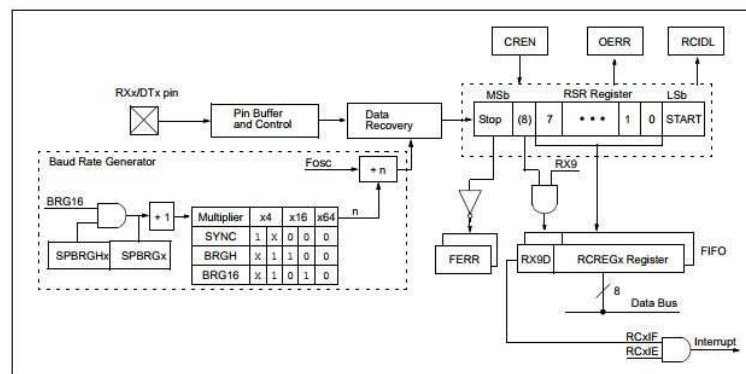


Fig. 51. Diagrama en bloques EUSART para la recepción [5].

El módulo contiene un generador de reloj, registros de desplazamiento y memorias intermedias de datos necesarias para llevar a cabo una transferencia seriada de datos, tanto de entrada como de salida, independientemente de la ejecución del programa del dispositivo. Puede configurarse como un sistema asíncrono full-duplex o un sistema síncrono semi-duplex.

El modo asíncrono full-duplex es útil para comunicaciones con sistemas periféricos, como computadoras. Se caracteriza porque la base de tiempo del emisor y receptor no es la misma, empleándose un reloj para la generación de datos en la transmisión y otro distinto para la recepción y es el que se implementó en el presente proyecto.

Como se ve en la Fig. 49 solo se necesitan de dos líneas para la comunicación entre los dispositivos, una de RX y otra de TX, en donde las líneas de cada dispositivo se conectan con la opuesta del otro con el cual se desea comunicar. El conector utilizado en el estándar RS-232 posee más líneas las cuales están orientadas al control de la comunicación, pero en nuestro caso no son necesarias y es suficiente con las líneas de transmisión y recepción de datos.

La comunicación entre los dispositivos se configura asíncrona full-duplex, de esta manera un dispositivo puede enviar y recibir datos al mismo tiempo. La información se transmite por

palabras, bytes o conjunto de bits, estando precedidos estos bits por un bit de arranque y finalizando con al menos un bit de parada, en donde este conjunto de bits compone un carácter. En este tipo de comunicación, el receptor sincroniza su reloj con el transmisor usando el bit de arranque que llega con cada carácter.

En la Fig. 52. se muestra un ejemplo para esta comunicación.

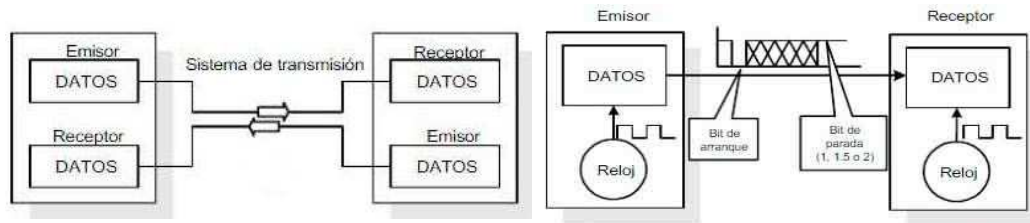


Fig. 52. A la izquierda configuración full-duplex. A la derecha comunicación asíncrona.

Para que lo anterior funciones correctamente, deben configurarse ciertos parámetros de igual manera en ambos dispositivos, como:

- La velocidad a la que operaran ambos módulos, en baudios
- El número de bits que se transmiten/reciben en un determinado momento
- La paridad del conjunto de bits que se envía y recibe
- El número de bits de parada
- Y el control de flujo de los datos

Hecho lo anterior, están dadas las condiciones para llevar a cabo la comunicación. En las Fig. 53 y 54 se observa la configuración del estándar RS-232, en la PC y el microcontrolador respectivamente.

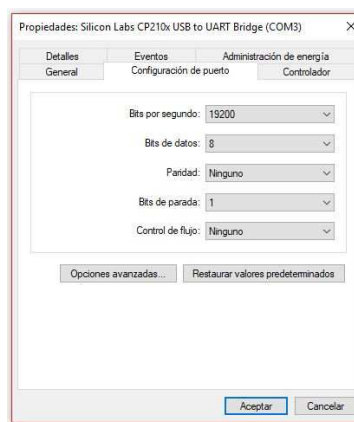


Fig. 53. Configuración RS-232 PC.

```
#USE RS232(UART2, BAUD=19200, DISABLE_INTS, BITS=8, PARITY=N, STOP=1)
```

Fig. 54. Configuración RS-232 microcontrolador.

La velocidad para la comunicación se fija en 19200 baudios, el número de bits de datos es de 8 sin paridad, con 1 bit de parada y control de flujo deshabilitado.

8.2. Interfaz gráfica de usuario (GUI)

Al comienzo de la presente sección se hizo mención de la interfaz de usuario y las principales funciones de la misma. En este apartado se la describirá en con más detalles.

8.2.1 Acerca de la interfaz

La interfaz gráfica de usuario (o GUI) fue desarrollada haciendo uso del entorno de desarrollo integrado (IDE) libre conocido como SharpDevelop, basado en el lenguaje de programación C Sharp. En la programación de la interfaz se configuran los mismos parámetros que en el microcontrolador para la comunicación RS-232, para que de esta manera no se tenga que hacer manualmente cada vez que se conecta el dispositivo a la PC.

En la Fig. 55 se muestra una imagen de la GUI desarrollada.

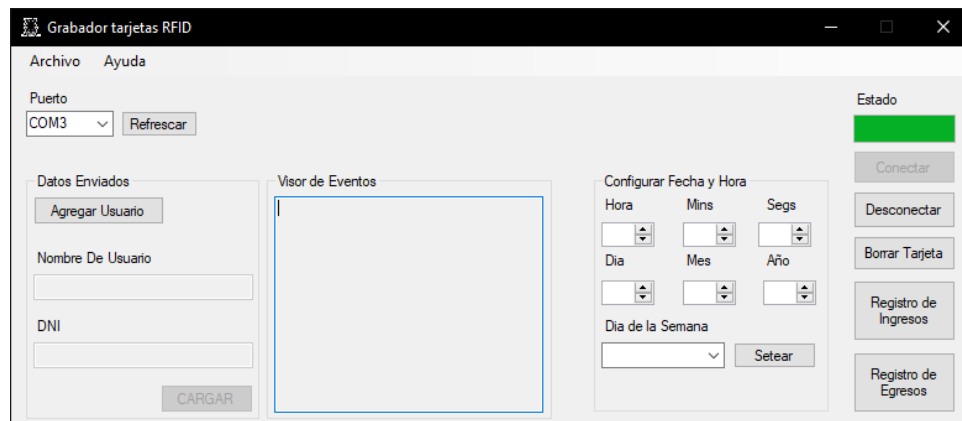


Fig. 55. Interfaz gráfica de usuario para la gestión del Sistema de Control de Acceso RFID.

Es una interfaz sencilla y fácil de utilizar. La misma permite seleccionar el puerto COM al cual se encuentra conectado el dispositivo y llevar a cabo:

- Carga de usuarios en el sistema y en las tarjetas
- Borrar el contenido de las tarjetas para blanquear las mismas
- Configurar la fecha y la hora del dispositivo en caso de que esta sea incorrecta
- Extraer el contenido de los archivos de los registros de los Ingresos y Egresos del mes en curso

La lista desplegable “Puerto” proporciona todos los puertos seriales presentes en la PC. De la misma debe seleccionarse aquel en el cual está conectado el dispositivo.

Una vez seleccionado el COM virtual generado por el CP2102 se procede a presionar el botón Conectar, para de esta manera entablar la comunicación con el dispositivo. Hecho esto el indicador de estado se llena, habiendo establecido la comunicación correctamente.

Para agregar un nuevo usuario se presiona sobre el botón “Agregar Usuario”, donde la interfaz pide que ingrese los campos necesarios y presione el botón “CARGAR”, para de esta manera grabar los datos en la tarjeta y cargarlos en la base de datos, como se ve en la Fig. 56.

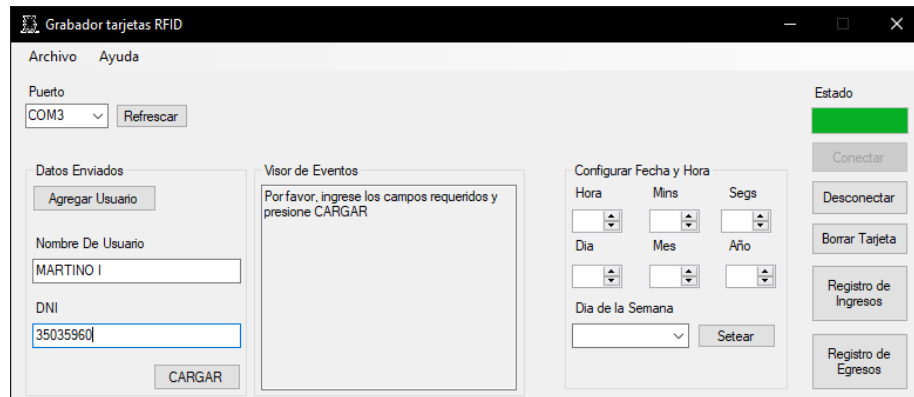


Fig. 56. Interfaz lista para cargar los datos.

Presionado el botón CARGAR, la interfaz le pedirá que se posicione la tarjeta sobre el lector/grabador para de esta manera transferir la información y cargarla también en la base de datos, cosa de la cual se encargara el microcontrolador.

Para borrar una tarjeta, se presiona el botón “Borrar Tarjeta”, y se siguen las instrucciones mostradas en el Visor de Eventos, como se ve en la Fig. 57.

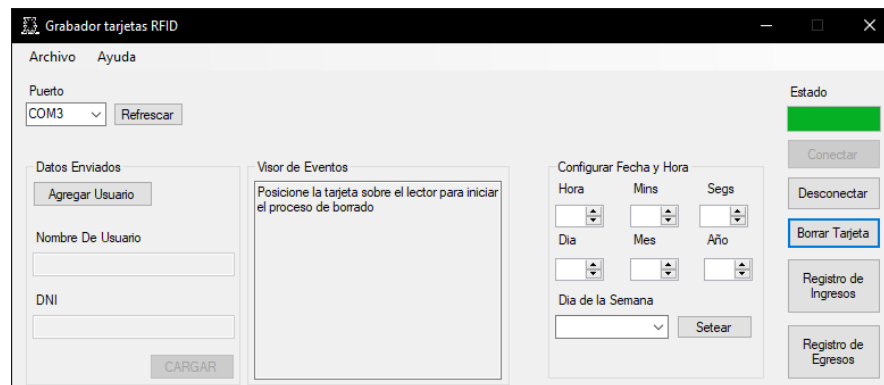


Fig. 57. Borrado de una tarjeta RFID.

Una vez puesta la tarjeta sobre el lector este la detecta y el microcontrolador se encarga de ejecutar una rutina que borra todos los sectores de la misma, llevando a cabo el blanqueo.

Para setear la fecha y la hora, se cargan los valores en las casillas correspondientes, se elige el día de la semana y se presiona el botón “Setear”.

Y finalmente para la extracción de los registros, se presiona los botones “Registro de Ingresos” si se desea obtener los registros de los ingresos del corriente mes o el botón “Registro de Egresos” si se desea obtener los registros de los egresos del corriente mes. Este proceso extrae la información de los archivos requeridos y la copia en uno nuevo que la interfaz gráfica abre automáticamente, como se vio en la Sección 4.2 Fig. 27b.

9. Alimentación

La fuente principal de alimentación del sistema es una fuente switching de 12 Voltios y 3 Amperios, la cual funciona como entrada de dos reguladores de tensión que se encargan de proporcionar los niveles de tensión adecuados a los distintos componentes del sistema.

El módulo para la tarjeta microSD, el microcontrolador, el módulo RTC y el LCD se alimentan con 5V, mientras que el módulo RFID lo hace con 3,3V.

Para lograr dichas tensiones a partir de los 12V que proporciona la fuente, los reguladores de tensión empleados fueron:

- Un LM7805 de la empresa ST, para obtener 5V.
- Y un LM317, también de la empresa ST para obtener los 3,3V.

9.1. Acerca del regulador de 5 voltios

En la Fig. 58. se muestra la arquitectura interna del regulador LM7805.

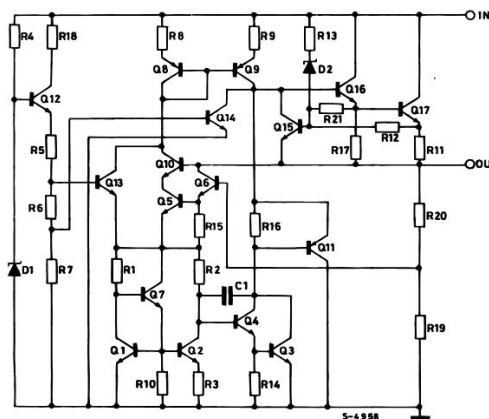


Fig. 58. Arquitectura interna LM7805 [8].

Este regulador consta de 3 pines y es conectado según la Fig. 59.

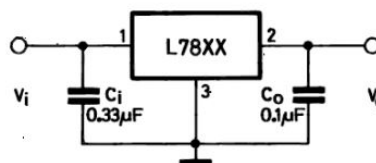


Fig. 59. Diagrama de conexionado exterior 7805 [8].

El pin V_i es conectado con el pin positivo de la fuente y el pin de GND con el negativo de la misma. El pin V_o es la salida regulada y es conectada a la entrada de alimentación de los

componentes mencionados al inicio de la sección 9, los cuales requieren de 5V para funcionar de manera adecuada. Como se observa en la figura el capacitor de entrada es de 0,33uF y el de salida es de 0,1uF y se emplean para mejorar la estabilidad de la tensión.

9.2. Acerca del regulador de 3,3 voltios

En la Fig. 60 se muestra la arquitectura interna del regulador LM317.

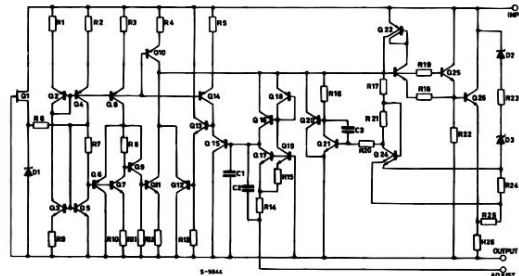


Fig. 60. Arquitectura interna LM317 [9].

Como se aprecia en la figura, el regulador consta de 3 pines y el mismo se conecta de la forma sugerida según la Fig. 61.

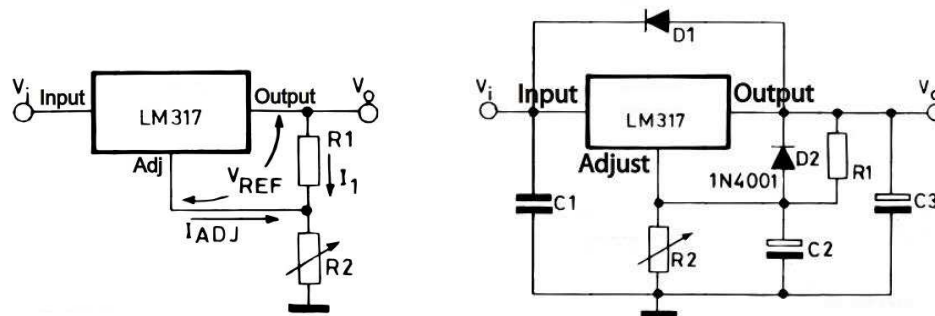


Fig. 61. Diagramas de conexionado exterior LM317 [9]. A la derecha conexionado básico y a la izquierda conexionado recomendado.

El LM317 es un regulador de voltaje que puede proporcionar una salida de tensión comprendida entre los 1,2V y los 37V. Para lograr el valor de tensión deseado es necesario emplear resistencias de valores adecuados y conectarlas como se muestra en la Fig. 61. El terminal V_i se conecta al terminal positivo de la fuente de alimentación. El terminal de ajuste, se conecta entre el terminal de salida V_o y GND mediante dos resistencias que forman un divisor resistivo, siendo sus valores los que determinan el valor de la tensión de salida. En este caso, las resistencias empleadas son de 1k2 para R1 y 1k8 para R2. Con estos valores, se obtiene en V_o una tensión regulada de 3,3V, la cual se emplea para alimentar el módulo RFID.

La Ecu. 2 nos permite calcular el valor de la tensión de salida V_o del regulador.

$$V_o = V_{REF} * \left(1 + \frac{R_2}{R_1}\right) + I_{ADJ} * R_2 \cong 1,25V * \left(1 + \frac{1k8\Omega}{1k2\Omega}\right) \cong 3,3V \quad \text{Ecu. 2.}$$

En donde el valor de I_{ADJ} es depreciable para la mayoría de las aplicaciones y el valor de V_{REF} es siempre el constante, de 1,25V.

El valor del capacitor de entrada es de 0.1uF, mientras que el capacitor de salida es de 1uF. Estos se emplean para mejorar la estabilidad de la tensión, tanto a la entrada como a la salida. El capacitor de ajuste C_{ADJ} se recomienda para mejorar el rechazo de ondulación de la tensión de salida, el mismo no siendo empleado en el presente proyecto. El diodo D2 solo es necesario si C_{ADJ} se utiliza. El diodo de protección D1 se emplea para proporcionar un camino de descarga de baja impedancia para evitar que C_o se descargue a la salida del regulador.

10. Integración

Si bien en el diagrama de flujo del programa principal de microcontrolador (Sección 7.1.2) se observa la interacción entre todas las partes del sistema, en esta sección se repasará como fue la integración parcial de cada una de ellas hasta conseguir el funcionamiento deseado.

La primera configuración realizada fue la del microcontrolador. Una vez funcionando de manera correcta, se conecta y configura un LCD. Se testea el mismo mostrando distintos mensajes en él. Hecho esto, se configuro el periférico MSSP en el cual se conectó el lector/grabador RFID. Luego se empezó a interactuar con las tarjetas. Primero se intentó detectarlas. Una vez detectadas, se obtuvo su número de serie y luego se procedió a autenticar los distintos bloques que componen la memoria de la misma, para después leer o escribir datos sobre ellos. Se llenaba la tarjeta con información conocida y luego se la extraía y se la mostraba en la pantalla LCD y en pantalla de la PC.

Funcionando el módulo RFID correctamente, se configuro otro periférico MSSP en el cual se conectó una tarjeta microSD. Primero se trató de inicializar la tarjeta de manera correcta y de escribir y leer algunos sectores. Logrado esto se inicializo el sistema FAT16 y se procedió a crear carpetas y archivos de texto de manera arbitraria dentro de la misma. Una vez que la tarjeta funcionaba a la perfección, se implementa la base datos en ella en donde se cargaron algunos usuarios. Dichos usuarios se encontraban cargados en algunas tarjetas RFID también.

El paso siguiente fue hacer que el lector leyera una tarjeta determinada, extrajera su información y se la pasara al microcontrolador para que este la comparara con los datos almacenados en la tarjeta microSD. Las tarjetas que contenían la misma información que la base de datos producían un acierto, mientras que las que estaban vacías o con información incorrecta no lo hacían.

Funcionando la base de datos y el lector, se adjuntó el módulo RTC mediante el protocolo de comunicación I²C. Establecida la comunicación, se configura y setea la fecha y la hora en el mismo. Hecho esto, cada vez que se produce un acceso el microcontrolador toma del módulo RTC la hora y fecha exacta en la que estos se generan, dejándolos plasmados en la base de datos del sistema.

Teniendo el sistema principal (por así decirlo) funcionando, se establece una comunicación RS-232 con la PC mediante la correcta configuración del módulo EUSART del microcontrolador. Primero se testea la misma enviando y recibiendo cadenas de caracteres simples y mostrándolas en la pantalla de la PC. Luego se prueba enviando una cadena de caracteres al microcontrolador y almacenándola en una tarjeta RFID para luego leerla y guardar dicho contenido en la tarjeta microSD. Funcionando adecuadamente la comunicación, se desarrolla la interfaz de usuario mediante la cual se pueden llevar a cabo las tareas mencionadas en la Sección 8.2.1.

En la siguiente fotografía se observar el interior del dispositivo. Puede visualizarse la placa principal, la fuente de alimentación y el puente para la comunicación con la PC, entre otras cosas.

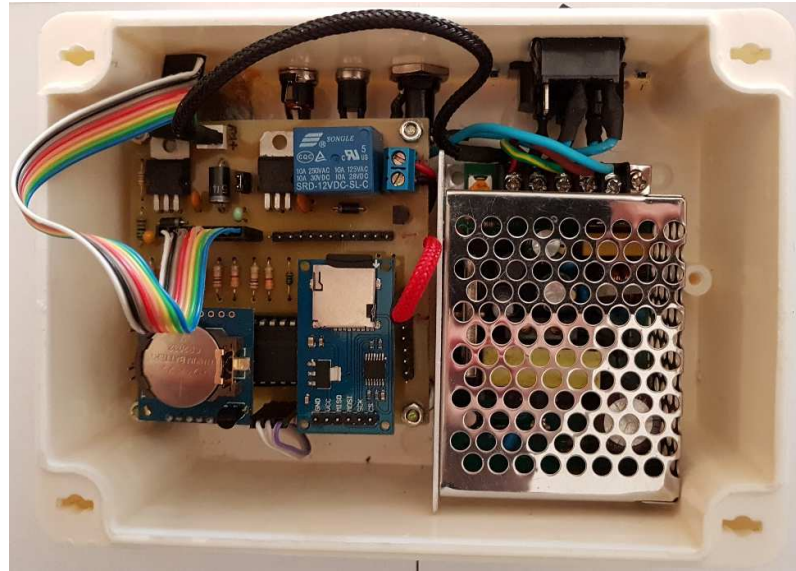


Fig. 62. Interior del dispositivo de control de acceso RFID.

La siguiente fotografía muestra la tapa del dispositivo, vista desde adentro, en donde se encuentra el LCD y el módulo RFID.

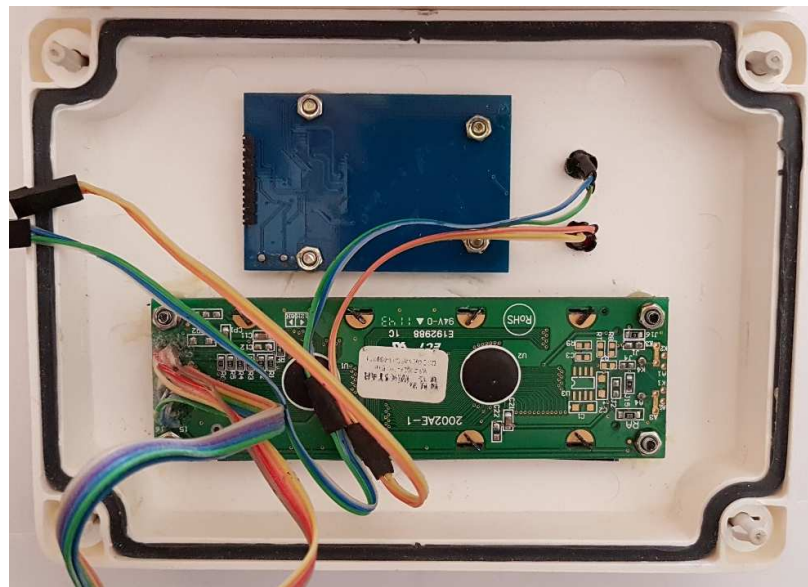


Fig. 63. Reverso de la tapa del dispositivo del control de acceso.

Y las siguientes fotografías muestran el exterior del dispositivo, tanto la parte trasera como la superior del mismo.



Fig. 64. Parte trasera del dispositivo.



Fig. 65. Vista superior del dispositivo ensamblado.

11. Conclusiones

La motivación de este proyecto como se planteó en la introducción, fue la creación de un dispositivo de control de acceso inalámbrico, económico y eficiente para de esta manera poder mejorar la gestión de llaves y usuarios a la entrada de un edificio, complejo, departamento, etc.

La realización del proyecto dividido en bloques facilitó el trabajo ya que se fueron desarrollando las soluciones para los diferentes problemas que se presentaban de manera aislada. A medida que estas partes se desarrollaban correctamente, las mismas se iban acoplando al sistema hasta que este quedó formado en su totalidad.

Se puede concluir que el proyecto desarrollado cumple de manera satisfactoria los objetivos planteados, ya que el mismo aborda la necesidad planteada. Además, cumple con los objetivos de la Sección 2, a saber:

- Se logró implementar un sistema de seguridad y control económico y eficiente, haciendo uso de componentes de bajo coste y excelente rendimiento frente a otras opciones muchas más costosas presentes en el mercado.
- Se mejora la gestión de llaves y usuarios a la entrada de una oficina, departamento o complejo, en donde cada usuario posee su propia credencial.
- Se elimina la necesidad de contacto físico a la hora del acceso con el dispositivo, ya que la comunicación entre las tarjetas y el mismo es inalámbrica.
- Se creó una interfaz gráfica de usuario que permite la gestión de las tarjetas RFID (grabación y borrado) sin la necesidad de tener que adquirir un grabador de tarjetas de manera extra y que además permite comunicar el dispositivo con una PC, pudiendo de esta manera obtener los archivos de registro y configurar la fecha y hora del mismo, entre otras tareas.
- Se implementó una base de datos en la cual queden asentados todos los ingresos/egresos realizados, con nombre de usuario, hora y fecha de manera organizada y separada en archivos diferentes.

Para concluir cabe mencionar que el funcionamiento del dispositivo se encuentra a la altura de las expectativas planteadas.


12. Bibliografía

- [1] NXP Semiconductors, MFRC522, datasheet; (Rev. 3.9 - 27 April 2016 112139)
- [2] NXP Semiconductors, MF1S50yyX/V1, datasheet; (Rev. 3.0 - 3 March 2014 279230)
- [3] SanDisk Corporation, Secure Digital Card; Product Manual Version 1.9 Document No. 80-13-00169 December 2003; SanDisk microSD OEM Product Manual Revision 2.0 Document No. 80-36-03335 March 2010
- [4] Microchip Technology Inc., PIC18F46K22 Silicon Errata and Sheet Clarification; (DS80000514J, 2016)
- [5] Microchip Technology Inc., PIC18F46K22, datasheet; (DS40001412G, 2016)
- [6] Maxim Integrated Products, DS1307 64 x 8, Serial, I2 C Real-Time Clock, datasheet; (REV: 3/15)
- [7] Silicon Laboratories Inc., CP2102, datasheet; (Rev. 1.8 1/17)
- [8] ST, L78 Positive Voltage Regulator ICs, 2016
- [9] ST, LM317 1.2 V to 37 V Adjustable Voltage Regulators, 2014
- [10] http://picmania.garcia-cuervo.net/proyectos_aux_rtc.php última consulta 07/05/2017
- [11] <http://www.todopic.com.ar/foros/index.php?topic=27786.0> última consulta 10/04/2017
- [12] [http://www.ucontrol.com.ar/forosmf/programacion-en-c/sd-card-libreria-fat16-libreria-a-nivel-hardware-\(ccs-c18-c30-ect-\)/msg58683/#msg58683](http://www.ucontrol.com.ar/forosmf/programacion-en-c/sd-card-libreria-fat16-libreria-a-nivel-hardware-(ccs-c18-c30-ect-)/msg58683/#msg58683) ultima consulta 10/04/2017
- [13] Norma ISO 14443 https://es.wikipedia.org/wiki/ISO_14443

13. Anexos

13.1. Anexo Alfa

Hojas de especificaciones PIC 18F46K22


MICROCHIP **PIC18(L)F2X/4XK22**
28/40/44-Pin, Low-Power, High-Performance
Microcontrollers with XLP Technology

High-Performance RISC CPU:

- C Compiler Optimized Architecture:
 - Optional extended instruction set designed to optimize re-entrant code
- Up to 1024 Bytes Data EEPROM
- Up to 64 Kbytes Linear Program Memory Addressing
- Up to 3896 Bytes Linear Data Memory Addressing
- Up to 16 MIPS Operation
- 16-bit Wide Instructions, 8-bit Wide Data Path
- Priority Levels for Interrupts
- 31-Level, Software Accessible Hardware Stack
- 8 x 8 Single-Cycle Hardware Multiplier

Flexible Oscillator Structure:

- Precision 16 MHz Internal Oscillator Block:
 - Factory calibrated to $\pm 1\%$
 - Selectable frequencies, 31 kHz to 16 MHz
 - 64 MHz performance available using PLL – no external components required
- Four Crystal modes up to 64 MHz
- Two External Clock modes up to 64 MHz
- 4X Phase Lock Loop (PLL)
- Secondary Oscillator using Timer1 @ 32 kHz
- Fail-Safe Clock Monitor:
 - Allows for safe shutdown if peripheral clock stops
 - Two-Speed Oscillator Start-up

Analog Features:

- Analog-to-Digital Converter (ADC) module:
 - 10-bit resolution, up to 30 external channels
 - Auto-acquisition capability
 - Conversion available during Sleep
 - Fixed Voltage Reference (FVR) channel
 - Independent input multiplexing
- Analog Comparator module:
 - Two rail-to-rail analog comparators
 - Independent input multiplexing
- Digital-to-Analog Converter (DAC) module:
 - Fixed Voltage Reference (FVR) with 1.024V, 2.048V and 4.096V output levels
 - 5-bit rail-to-rail resistive DAC with positive and negative reference selection
- Charge Time Measurement Unit (CTMU) module:
 - Supports capacitive touch sensing for touch screens and capacitive switches

eXtreme Low-Power Features (XLP) (PIC18(L)F2X/4XK22):

- Sleep mode: 20 nA, typical
- Watchdog Timer: 300 nA, typical
- Timer1 Oscillator: 800 nA @ 32 kHz
- Peripheral Module Disable

Special Microcontroller Features:

- 2.3V to 5.5V Operation – PIC18FXXK22 devices
- 1.8V to 3.6V Operation – PIC18LFXXK22 devices
- Self-Programmable under Software Control
- High/Low-Voltage Detection (HLVD) module:
 - Programmable 16-Level
 - Interrupt on High/Low-Voltage Detection
- Programmable Brown-out Reset (BOR):
 - With software enable option
 - Configurable shutdown in Sleep
- Extended Watchdog Timer (WDT):
 - Programmable period from 4 ms to 131s
- In-Circuit Serial Programming™ (ICSP™):
 - Single-Supply 3V
- In-Circuit Debug (ICD)

Peripheral Highlights:

- Up to 35 I/O Pins plus 1 Input-Only Pin:
 - High-Current Sink/Source 25 mA/25 mA
 - Three programmable external interrupts
 - Four programmable interrupt-on-change
 - Nine programmable weak pull-ups
 - Programmable slew rate
- SR Latch:
 - Multiple Set/Reset input options
- Two Capture/Compare/PWM (CCP) modules
- Three Enhanced CCP (ECCP) modules:
 - One, two or four PWM outputs
 - Selectable polarity
 - Programmable dead time
 - Auto-Shutdown and Auto-Restart
 - PWM steering
- Two Master Synchronous Serial Port (MSSP) modules:
 - 3-wire SPI (supports all 4 modes)
 - I²C Master and Slave modes with address mask

PIC18(L)F2X/4XK22

FIGURE 3: 40-PIN PDIP DIAGRAM

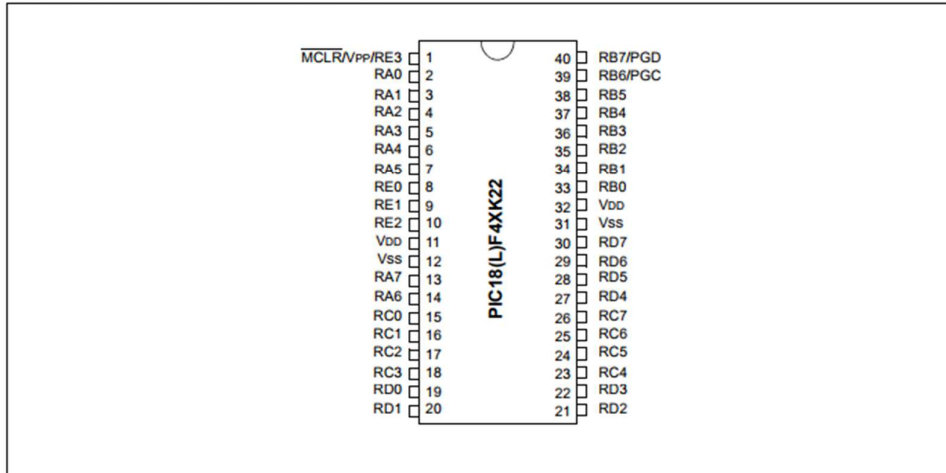
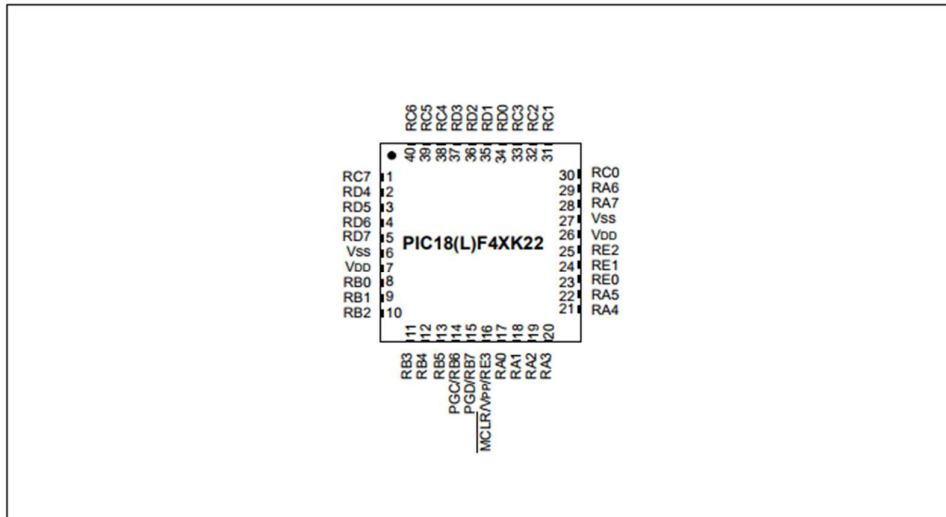


FIGURE 4: 40-PIN UQFN DIAGRAM



PIC18(L)F2X/4XK22

TABLE 3: PIC18(L)F4XK22 PIN SUMMARY

| 40-PDIP | 40-QJFN | 44-TQFP | 44-QFN | IO | Analog | Comparator | CTMU | SR Latch | Reference | (E)CCP | EUSART | MSSP | Timers | Interrupts | Pull-up | Basic |
|---------|---------|---------|--------|-----|--------|------------|-------|----------|-------------|-------------------------|---------|-----------|--------------------------------------|------------|---------|-----------|
| 2 | 17 | 19 | 19 | RA0 | AN0 | C12IN0- | | | | | | | | | | |
| 3 | 18 | 20 | 20 | RA1 | AN1 | C12IN1- | | | | | | | | | | |
| 4 | 19 | 21 | 21 | RA2 | AN2 | C2IN+ | | | VREF-DACOUT | | | | | | | |
| 5 | 20 | 22 | 22 | RA3 | AN3 | C1IN+ | | | VREF+ | | | | | | | |
| 6 | 21 | 23 | 23 | RA4 | | C1OUT | | SRQ | | | | | T0CKI | | | |
| 7 | 22 | 24 | 24 | RA5 | AN4 | C2OUT | | SRNQ | HLVDIN | | | SS1 | | | | |
| 14 | 29 | 31 | 33 | RA6 | | | | | | | | | | | | OSC2 CLK0 |
| 13 | 28 | 30 | 32 | RA7 | | | | | | | | | | | | OSC1 CLK1 |
| 33 | 8 | 8 | 9 | RB0 | AN12 | | | SRI | | FLT0 | | | | INT0 | Y | |
| 34 | 9 | 9 | 10 | RB1 | AN10 | C12IN3- | | | | | | | | INT1 | Y | |
| 35 | 10 | 10 | 11 | RB2 | AN8 | | CTED1 | | | | | | | INT2 | Y | |
| 36 | 11 | 11 | 12 | RB3 | AN9 | C12IN2- | CTED2 | | | CCP2 P2A ⁽¹⁾ | | | | | | Y |
| 37 | 12 | 14 | 14 | RB4 | AN11 | | | | | | | | T5G | IOC | Y | |
| 38 | 13 | 15 | 15 | RB5 | AN13 | | | | | CCP3 P3A ⁽³⁾ | | | T1G T3CKI ⁽²⁾ | IOC | Y | |
| 39 | 14 | 16 | 16 | RB6 | | | | | | | | | | IOC | Y | PGC |
| 40 | 15 | 17 | 17 | RB7 | | | | | | | | | | IOC | Y | PGD |
| 15 | 30 | 32 | 34 | RC0 | | | | | | P2B ⁽⁴⁾ | | | SOSCO T1CKI T3CKI ⁽²⁾ T3G | | | |
| 16 | 31 | 35 | 35 | RC1 | | | | | | CCP2 ⁽¹⁾ P2A | | | SOSCI | | | |
| 17 | 32 | 36 | 36 | RC2 | AN14 | | CTPLS | | | CCP1 P1A | | | T5CKI | | | |
| 18 | 33 | 37 | 37 | RC3 | AN15 | | | | | | | SCK1 SCL1 | | | | |
| 23 | 38 | 42 | 42 | RC4 | AN16 | | | | | | | SDI1 SDA1 | | | | |
| 24 | 39 | 43 | 43 | RC5 | AN17 | | | | | | | SDO1 | | | | |
| 25 | 40 | 44 | 44 | RC6 | AN18 | | | | | | TX1 CK1 | | | | | |
| 26 | 1 | 1 | 1 | RC7 | AN19 | | | | | | RX1 DT1 | | | | | |
| 19 | 34 | 38 | 38 | RD0 | AN20 | | | | | | | SCK2 SCL2 | | | | |
| 20 | 35 | 39 | 39 | RD1 | AN21 | | | | | CCP4 | | SDI2 SDA2 | | | | |
| 21 | 36 | 40 | 40 | RD2 | AN22 | | | | | P2B ⁽⁴⁾ | | | | | | |
| 22 | 37 | 41 | 41 | RD3 | AN23 | | | | | P2C | | SS2 | | | | |
| 27 | 2 | 2 | 2 | RD4 | AN24 | | | | | P2D | | SD02 | | | | |
| 28 | 3 | 3 | 3 | RD5 | AN25 | | | | | P1B | | | | | | |
| 29 | 4 | 4 | 4 | RD6 | AN26 | | | | | P1C | TX2 CK2 | | | | | |
| 30 | 5 | 5 | 5 | RD7 | AN27 | | | | | P1D | RX2 DT2 | | | | | |
| 8 | 23 | 25 | 25 | RE0 | AN5 | | | | | CCP3 P3A ⁽³⁾ | | | | | | |

Note 1: CCP2 multiplexed in fuses.
 2: T3CKI multiplexed in fuses.
 3: CCP3/P3A multiplexed in fuses.
 4: P2B multiplexed in fuses.



PIC18(L)F2X/4XK22

TABLE 3: PIC18(L)F4XK22 PIN SUMMARY (CONTINUED)

| 40-PDIP | 40-JOIFN | 44-TOFP | 44-QFN | IO | Analog | Comparator | CTMU | SR Latch | Reference | (E)CCP | EUSART | MSSP | Timers | Interrupts | Pull-up | Basic |
|-----------|----------|------------------|----------------|-----|--------|------------|------|----------|-----------|--------|--------|------|--------|------------|---------|-------------|
| 9 | 24 | 26 | 26 | RE1 | AN6 | | | | | P3B | | | | | | |
| 10 | 25 | 27 | 27 | RE2 | AN7 | | | | | CCP5 | | | | | | |
| 1 | 16 | 18 | 18 | RE3 | | | | | | | | | | | Y | MCLR VPP |
| 11, 32 | 7, 26 | 7, 28 | 7, 8 28, 29 | VDD | | | | | | | | | | | | VDD |
| 12, 31 | 6, 27 | 6, 29 | 6, 30, 31 | VSS | | | | | | | | | | | | VSS |
| — | — | 12, 13 33, 34 | 13 | NC | | | | | | | | | | | | |

Note 1: CCP2 multiplexed in fuses.
 2: T3CKI multiplexed in fuses.
 3: CCP3/P3A multiplexed in fuses.
 4: P2B multiplexed in fuses.

PIC18(L)F2X/4XK22

1.0 DEVICE OVERVIEW

This document contains device specific information for the following devices:

- PIC18F23K22
- PIC18F24K22
- PIC18F25K22
- PIC18F26K22
- PIC18F43K22
- PIC18F44K22
- PIC18F45K22
- PIC18F46K22
- PIC18LF23K22
- PIC18LF24K22
- PIC18LF25K22
- PIC18LF26K22
- PIC18LF43K22
- PIC18LF44K22
- PIC18LF45K22
- PIC18LF46K22

This family offers the advantages of all PIC18 microcontrollers – namely, high computational performance at an economical price – with the addition of high-endurance, Flash program memory. On top of these features, the PIC18(L)F2X/4XK22 family introduces design enhancements that make these microcontrollers a logical choice for many high-performance, power sensitive applications.

1.1 New Core Features

1.1.1 XLP TECHNOLOGY

All of the devices in the PIC18(L)F2X/4XK22 family incorporate a range of features that can significantly reduce power consumption during operation. Key items include:

- **Alternate Run Modes:** By clocking the controller from the Timer1 source or the internal oscillator block, power consumption during code execution can be reduced by as much as 90%.
- **Multiple Idle Modes:** The controller can also run with its CPU core disabled but the peripherals still active. In these states, power consumption can be reduced even further, to as little as 4% of normal operation requirements.
- **On-the-fly Mode Switching:** The power-managed modes are invoked by user code during operation, allowing the user to incorporate power-saving ideas into their application's software design.
- **Low Consumption in Key Modules:** The power requirements for both Timer1 and the Watchdog Timer are minimized. See [Section 27.0 "Electrical Specifications"](#) for values.

1.1.2 MULTIPLE OSCILLATOR OPTIONS AND FEATURES

All of the devices in the PIC18(L)F2X/4XK22 family offer ten different oscillator options, allowing users a wide range of choices in developing application hardware. These include:

- Four Crystal modes, using crystals or ceramic resonators
- Two External Clock modes, offering the option of using two pins (oscillator input and a divide-by-4 clock output) or one pin (oscillator input, with the second pin reassigned as general I/O)
- Two External RC Oscillator modes with the same pin options as the External Clock modes
- An internal oscillator block which contains a 16 MHz HFINTOSC oscillator and a 31 kHz LFINTOSC oscillator, which together provide eight user selectable clock frequencies, from 31 kHz to 16 MHz. This option frees the two oscillator pins for use as additional general purpose I/O.
- A Phase Lock Loop (PLL) frequency multiplier, available to both external and internal oscillator modes, which allows clock speeds of up to 64 MHz. Used with the internal oscillator, the PLL gives users a complete selection of clock speeds, from 31 kHz to 64 MHz – all without using an external crystal or clock circuit.

Besides its availability as a clock source, the internal oscillator block provides a stable reference source that gives the family additional features for robust operation:

- **Fail-Safe Clock Monitor:** This option constantly monitors the main clock source against a reference signal provided by the LFINTOSC. If a clock failure occurs, the controller is switched to the internal oscillator block, allowing for continued operation or a safe application shutdown.
- **Two-Speed Start-up:** This option allows the internal oscillator to serve as the clock source from Power-on Reset, or Wake-up from Sleep mode, until the primary clock source is available.

PIC18(L)F2X/4XK22

FIGURE 1-1: PIC18(L)F2X/4XK22 FAMILY BLOCK DIAGRAM

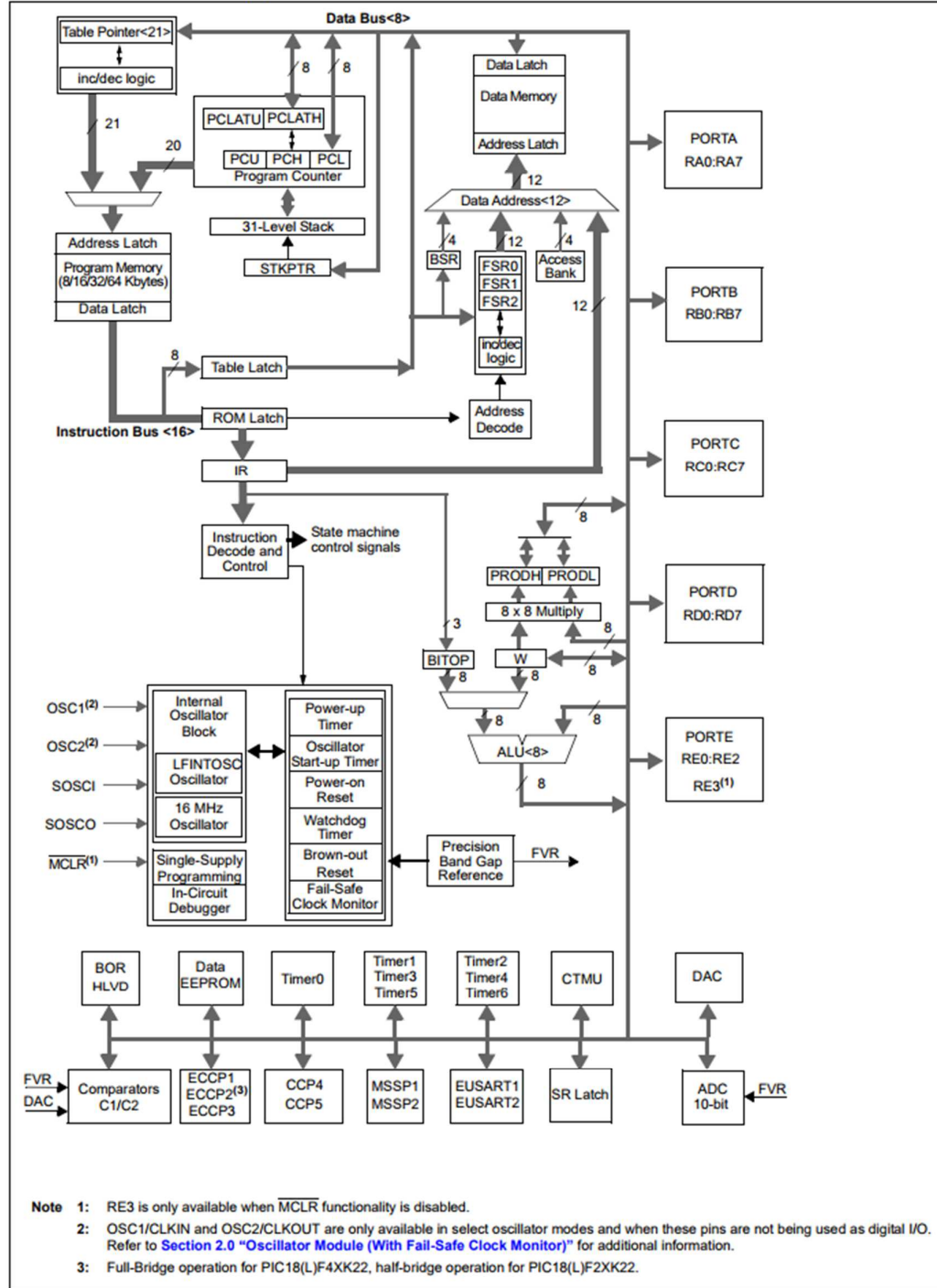


TABLE 1-3: PIC18(L)F4XK22 PINOUT I/O DESCRIPTIONS

| Pin Number | | | | Pin Name | Pin Type | Buffer Type | Description |
|------------|------|-----|------|----------------------------|----------|------------------|---|
| PDIP | TQFP | QFN | UQFN | | | | |
| 2 | 19 | 19 | 17 | RA0/C12IN0-/AN0 | | | |
| | | | | RA0 | I/O | TTL | Digital I/O. |
| | | | | C12IN0- AN0 | I I | Analog Analog | Comparators C1 and C2 inverting input. Analog input 0. |
| 3 | 20 | 20 | 18 | RA1/C12IN1-/AN1 | | | |
| | | | | RA1 | I/O | TTL | Digital I/O. |
| | | | | C12IN1- AN1 | I I | Analog Analog | Comparators C1 and C2 inverting input. Analog input 1. |
| 4 | 21 | 21 | 19 | RA2/C2IN+/AN2/DACOUT/VREF- | | | |
| | | | | RA2 | I/O | TTL | Digital I/O. |
| | | | | C2IN+ | I | Analog | Comparator C2 non-inverting input. |
| | | | | AN2 | I | Analog | Analog input 2. |
| | | | | DACOUT | O | Analog | DAC Reference output. |
| | | | | VREF- | I | Analog | A/D reference voltage (low) input. |
| 5 | 22 | 22 | 20 | RA3/C1IN+/AN3/VREF+ | | | |
| | | | | RA3 | I/O | TTL | Digital I/O. |
| | | | | C1IN+ | I | Analog | Comparator C1 non-inverting input. |
| | | | | AN3 | I | Analog | Analog input 3. |
| | | | | VREF+ | I | Analog | A/D reference voltage (high) input. |
| 6 | 23 | 23 | 21 | RA4/C1OUT/SRQ/T0CKI | | | |
| | | | | RA4 | I/O | ST | Digital I/O. |
| | | | | C1OUT | O | CMOS | Comparator C1 output. |
| | | | | SRQ | O | TTL | SR latch Q output. |
| | | | | T0CKI | I | ST | Timer0 external clock input. |

Legend: TTL = TTL compatible input CMOS = CMOS compatible input or output; ST = Schmitt Trigger input with CMOS levels; I = Input; O = Output; P = Power.

- Note**
- 1: Default pin assignment for P2B, T3CKI, CCP3/P3A and CCP2/P2A when Configuration bits PB2MX, T3CMX, CCP3MX and CCP2MX are set.
 - 2: Alternate pin assignment for P2B, T3CKI, CCP3/P3A and CCP2/P2A when Configuration bits PB2MX, T3CMX, CCP3MX and CCP2MX are clear.

PIC18(L)F2X/4XK22

TABLE 1-3: PIC18(L)F4XK22 PINOUT I/O DESCRIPTIONS (CONTINUED)

| Pin Number | | | | Pin Name | Pin Type | Buffer Type | Description |
|------------|------|--------|-----------------|--------------------------------|----------|-------------|---|
| PDIP | TQFP | QFN | UQFN | | | | |
| 7 | 24 | 24 | 22 | RA5/C2OUT/SRNQ/SS1/HLVDIN/AN4 | | | |
| | | | | RA5 | I/O | TTL | Digital I/O. |
| | | | | C2OUT | O | CMOS | Comparator C2 output. |
| | | | | SRNQ | O | TTL | SR latch \bar{Q} output. |
| | | | | SS1 | I | TTL | SPI slave select input (MSSP1). |
| | | | | HLVDIN | I | Analog | High/Low-Voltage Detect input. |
| 14 | 31 | 33 | 29 | RA6/CLKO/OSC2 | | | |
| | | | | RA6 | I/O | TTL | Digital I/O. |
| | | | | CLKO | O | — | In RC mode, OSC2 pin outputs CLKOUT which has 1/4 the frequency of OSC1 and denotes the instruction cycle rate. |
| 13 | 30 | 32 | 28 | RA7/CLKI/OSC1 | | | |
| | | | | RA7 | I/O | TTL | Digital I/O. |
| | | | | CLKI | I | CMOS | External clock source input. Always associated with pin function OSC1. |
| 33 | 8 | 9 | 8 | RB0/INT0/FLT0/SRI/AN12 | | | |
| | | | | RB0 | I/O | TTL | Digital I/O. |
| | | | | INT0 | I | ST | External interrupt 0. |
| | | | | FLT0 | I | ST | PWM Fault input for ECCP Auto-Shutdown. |
| 34 | 9 | 10 | 9 | RB1/INT1/C12IN3-/AN10 | | | |
| | | | | RB1 | I/O | TTL | Digital I/O. |
| | | | | INT1 | I | ST | External interrupt 1. |
| | | | | C12IN3- | I | Analog | Comparators C1 and C2 inverting input. |
| | | | | AN10 | I | Analog | Analog input 10. |
| 35 | 10 | 11 | 10 | RB2/INT2/CTED1/AN8 | | | |
| | | | | RB2 | I/O | TTL | Digital I/O. |
| | | | | INT2 | I | ST | External interrupt 2. |
| | | | | CTED1 | I | ST | CTMU Edge 1 input. |
| 36 | 11 | 12 | 11 | RB3/CTED2/P2A/CCP2/C12IN2-/AN9 | | | |
| | | | | RB3 | I/O | TTL | Digital I/O. |
| | | | | CTED2 | I | ST | CTMU Edge 2 input. |
| | | | | P2A ⁽²⁾ | O | CMOS | Enhanced CCP2 PWM output. |
| | | | | CCP2 ⁽²⁾ | I/O | ST | Capture 2 input/Compare 2 output/PWM 2 output. |
| | | | | C12IN2- | I | Analog | Comparators C1 and C2 inverting input. |
| AN9 | I | Analog | Analog input 9. | | | | |

Legend: TTL = TTL compatible input CMOS = CMOS compatible input or output; ST = Schmitt Trigger input with CMOS levels; I = Input; O = Output; P = Power.

Note 1: Default pin assignment for P2B, T3CKI, CCP3/P3A and CCP2/P2A when Configuration bits PB2MX, T3CMX, CCP3MX and CCP2MX are set.

Note 2: Alternate pin assignment for P2B, T3CKI, CCP3/P3A and CCP2/P2A when Configuration bits PB2MX, T3CMX, CCP3MX and CCP2MX are clear.

PIC18(L)F2X/4XK22

TABLE 1-3: PIC18(L)F4XK22 PINOUT I/O DESCRIPTIONS (CONTINUED)

| Pin Number | | | | Pin Name | Pin Type | Buffer Type | Description |
|------------|------|-----|------|-------------------------|----------|-------------|---|
| PDIP | TQFP | QFN | UQFN | | | | |
| 18 | 37 | 37 | 33 | RC3/SCK1/SCL1/AN15 | | | |
| | | | | RC3 | I/O | ST | Digital I/O. |
| | | | | SCK1 | I/O | ST | Synchronous serial clock input/output for SPI mode (MSSP). |
| | | | | SCL1 | I/O | ST | Synchronous serial clock input/output for I ² C mode (MSSP). |
| | | | | AN15 | I | Analog | Analog input 15. |
| 23 | 42 | 42 | 38 | RC4/SDI1/SDA1/AN16 | | | |
| | | | | RC4 | I/O | ST | Digital I/O. |
| | | | | SDI1 | I | ST | SPI data in (MSSP). |
| | | | | SDA1 | I/O | ST | I ² C data I/O (MSSP). |
| | | | | AN16 | I | Analog | Analog input 16. |
| 24 | 43 | 43 | 39 | RC5/SDO1/AN17 | | | |
| | | | | RC5 | I/O | ST | Digital I/O. |
| | | | | SDO1 | O | — | SPI data out (MSSP). |
| | | | | AN17 | I | Analog | Analog input 17. |
| 25 | 44 | 44 | 40 | RC6/TX1/CK1/AN18 | | | |
| | | | | RC6 | I/O | ST | Digital I/O. |
| | | | | TX1 | O | — | EUSART asynchronous transmit. |
| | | | | CK1 | I/O | ST | EUSART synchronous clock (see related RXx/DTx). |
| | | | | AN18 | I | Analog | Analog input 18. |
| 26 | 1 | 1 | 1 | RC7/RX1/DT1/AN19 | | | |
| | | | | RC7 | I/O | ST | Digital I/O. |
| | | | | RX1 | I | ST | EUSART asynchronous receive. |
| | | | | DT1 | I/O | ST | EUSART synchronous data (see related TXx/CKx). |
| | | | | AN19 | I | Analog | Analog input 19. |
| 19 | 38 | 38 | 34 | RD0/SCK2/SCL2/AN20 | | | |
| | | | | RD0 | I/O | ST | Digital I/O. |
| | | | | SCK2 | I/O | ST | Synchronous serial clock input/output for SPI mode (MSSP). |
| | | | | SCL2 | I/O | ST | Synchronous serial clock input/output for I ² C mode (MSSP). |
| | | | | AN20 | I | Analog | Analog input 20. |
| 20 | 39 | 39 | 35 | RD1/CCP4/SDI2/SDA2/AN21 | | | |
| | | | | RD1 | I/O | ST | Digital I/O. |
| | | | | CCP4 | I/O | ST | Capture 4 input/Compare 4 output/PWM 4 output. |
| | | | | SDI2 | I | ST | SPI data in (MSSP). |
| | | | | SDA2 | I/O | ST | I ² C data I/O (MSSP). |
| | | | | AN21 | I | Analog | Analog input 21. |

Legend: TTL = TTL compatible input CMOS = CMOS compatible input or output; ST = Schmitt Trigger input with CMOS levels; I = Input; O = Output; P = Power.

- Note** 1: Default pin assignment for P2B, T3CK1, CCP3/P3A and CCP2/P2A when Configuration bits PB2MX, T3CMX, CCP3MX and CCP2MX are set.
- 2: Alternate pin assignment for P2B, T3CK1, CCP3/P3A and CCP2/P2A when Configuration bits PB2MX, T3CMX, CCP3MX and CCP2MX are clear.

PIC18(L)F2X/4XK22

TABLE 1-3: PIC18(L)F4XK22 PINOUT I/O DESCRIPTIONS (CONTINUED)

| Pin Number | | | | Pin Name | Pin Type | Buffer Type | Description |
|------------|------|-----|------|----------------------|----------|-------------|---|
| PDIP | TQFP | QFN | UQFN | | | | |
| 21 | 40 | 40 | 36 | RD2/P2B/AN22 | | | |
| | | | | RD2 | I/O | ST | Digital I/O. |
| | | | | P2B ⁽¹⁾ | O | CMOS | Enhanced CCP2 PWM output. |
| | | | | AN22 | I | Analog | Analog input 22. |
| 22 | 41 | 41 | 37 | RD3/P2C/SS2/AN23 | | | |
| | | | | RD3 | I/O | ST | Digital I/O. |
| | | | | P2C | O | CMOS | Enhanced CCP2 PWM output. |
| | | | | SS2 | I | TTL | SPI slave select input (MSSP). |
| 27 | 2 | 2 | 2 | RD4/P2D/SDO2/AN24 | | | |
| | | | | RD4 | I/O | ST | Digital I/O. |
| | | | | P2D | O | CMOS | Enhanced CCP2 PWM output. |
| | | | | SDO2 | O | — | SPI data out (MSSP). |
| 28 | 3 | 3 | 3 | RD5/P1B/AN25 | | | |
| | | | | RD5 | I/O | ST | Digital I/O. |
| | | | | P1B | O | CMOS | Enhanced CCP1 PWM output. |
| | | | | AN25 | I | Analog | Analog input 25. |
| 29 | 4 | 4 | 4 | RD6/P1C/TX2/CK2/AN26 | | | |
| | | | | RD6 | I/O | ST | Digital I/O. |
| | | | | P1C | O | CMOS | Enhanced CCP1 PWM output. |
| | | | | TX2 | O | — | EUSART asynchronous transmit. |
| | | | | CK2 | I/O | ST | EUSART synchronous clock (see related RXx/DTx). |
| 30 | 5 | 5 | 5 | RD7/P1D/RX2/DT2/AN27 | | | |
| | | | | RD7 | I/O | ST | Digital I/O. |
| | | | | P1D | O | CMOS | Enhanced CCP1 PWM output. |
| | | | | RX2 | I | ST | EUSART asynchronous receive. |
| | | | | DT2 | I/O | ST | EUSART synchronous data (see related TXx/CKx). |
| 8 | 25 | 25 | 23 | RE0/P3A/CCP3/AN5 | | | |
| | | | | RE0 | I/O | ST | Digital I/O. |
| | | | | P3A ⁽²⁾ | O | CMOS | Enhanced CCP3 PWM output. |
| | | | | CCP3 ⁽²⁾ | I/O | ST | Capture 3 input/Compare 3 output/PWM 3 output. |
| 9 | 26 | 26 | 24 | RE1/P3B/AN6 | | | |
| | | | | RE1 | I/O | ST | Digital I/O. |
| | | | | P3B | O | CMOS | Enhanced CCP3 PWM output. |
| | | | | AN6 | I | Analog | Analog input 6. |

Legend: TTL = TTL compatible input CMOS = CMOS compatible input or output; ST = Schmitt Trigger input with CMOS levels; I = Input; O = Output; P = Power.

Note 1: Default pin assignment for P2B, T3CKI, CCP3/P3A and CCP2/P2A when Configuration bits PB2MX, T3CMX, CCP3MX and CCP2MX are set.

Note 2: Alternate pin assignment for P2B, T3CKI, CCP3/P3A and CCP2/P2A when Configuration bits PB2MX, T3CMX, CCP3MX and CCP2MX are clear.

PIC18(L)F2X/4XK22

TABLE 1-3: PIC18(L)F4XK22 PINOUT I/O DESCRIPTIONS (CONTINUED)

| Pin Number | | | | Pin Name | Pin Type | Buffer Type | Description |
|------------|--------------|--------------|-------|--------------|----------|-------------|---|
| PDIP | TQFP | QFN | UQFN | | | | |
| 10 | 27 | 27 | 25 | RE2/CCP5/AN7 | | | |
| | | | | RE2 | I/O | ST | Digital I/O. |
| | | | | CCP5 | I/O | ST | Capture 5 input/Compare 5 output/PWM 5 output |
| | | | | AN7 | I | Analog | Analog input 7. |
| 1 | 18 | 18 | 16 | RE3/VPP/MCLR | | | |
| | | | | RE3 | I | ST | Digital input. |
| | | | | VPP | P | | Programming voltage input. |
| | | | | MCLR | I | ST | Active-low Master Clear (device Reset) input. |
| 11,32 | 7, 28 | 7, 8, 28, 29 | 7, 26 | VDD | P | — | Positive supply for logic and I/O pins. |
| 12,31 | 6, 29 | 6,30, 31 | 6, 27 | VSS | P | — | Ground reference for logic and I/O pins. |
| | 12,13, 33,34 | 13 | | NC | | | |

Legend: TTL = TTL compatible input CMOS = CMOS compatible input or output; ST = Schmitt Trigger input with CMOS levels; I = Input; O = Output; P = Power.

- Note**
- 1: Default pin assignment for P2B, T3CKI, CCP3/P3A and CCP2/P2A when Configuration bits PB2MX, T3CMX, CCP3MX and CCP2MX are set.
 - 2: Alternate pin assignment for P2B, T3CKI, CCP3/P3A and CCP2/P2A when Configuration bits PB2MX, T3CMX, CCP3MX and CCP2MX are clear.

PIC18(L)F2X/4XK22

2.0 OSCILLATOR MODULE (WITH FAIL-SAFE CLOCK MONITOR)

2.1 Overview

The oscillator module has a wide variety of clock sources and selection features that allow it to be used in a wide range of applications while maximizing performance and minimizing power consumption. Figure 2-1 illustrates a block diagram of the oscillator module.

Clock sources can be configured from external oscillators, quartz crystal resonators, ceramic resonators and Resistor-Capacitor (RC) circuits. In addition, the system clock source can be configured from one of three internal oscillators, with a choice of speeds selectable via software. Additional clock features include:

- Selectable system clock source between external or internal sources via software.
- Two-Speed Start-up mode, which minimizes latency between external oscillator start-up and code execution.
- Fail-Safe Clock Monitor (FSCM) designed to detect a failure of the external clock source (LP, XT, HS, EC or RC modes) and switch automatically to the internal oscillator.
- Oscillator Start-up Timer (OST) ensures stability of crystal oscillator sources.

The primary clock module can be configured to provide one of six clock sources as the primary clock.

1. RC External Resistor/Capacitor
2. LP Low-Power Crystal
3. XT Crystal/Resonator
4. INTOSC Internal Oscillator
5. HS High-Speed Crystal/Resonator
6. EC External Clock

The HS and EC oscillator circuits can be optimized for power consumption and oscillator speed using settings in FOSC<3:0>. Additional FOSC<3:0> selections enable RA6 to be used as I/O or CLK0 (Fosc/4) for RC, EC and INTOSC Oscillator modes.

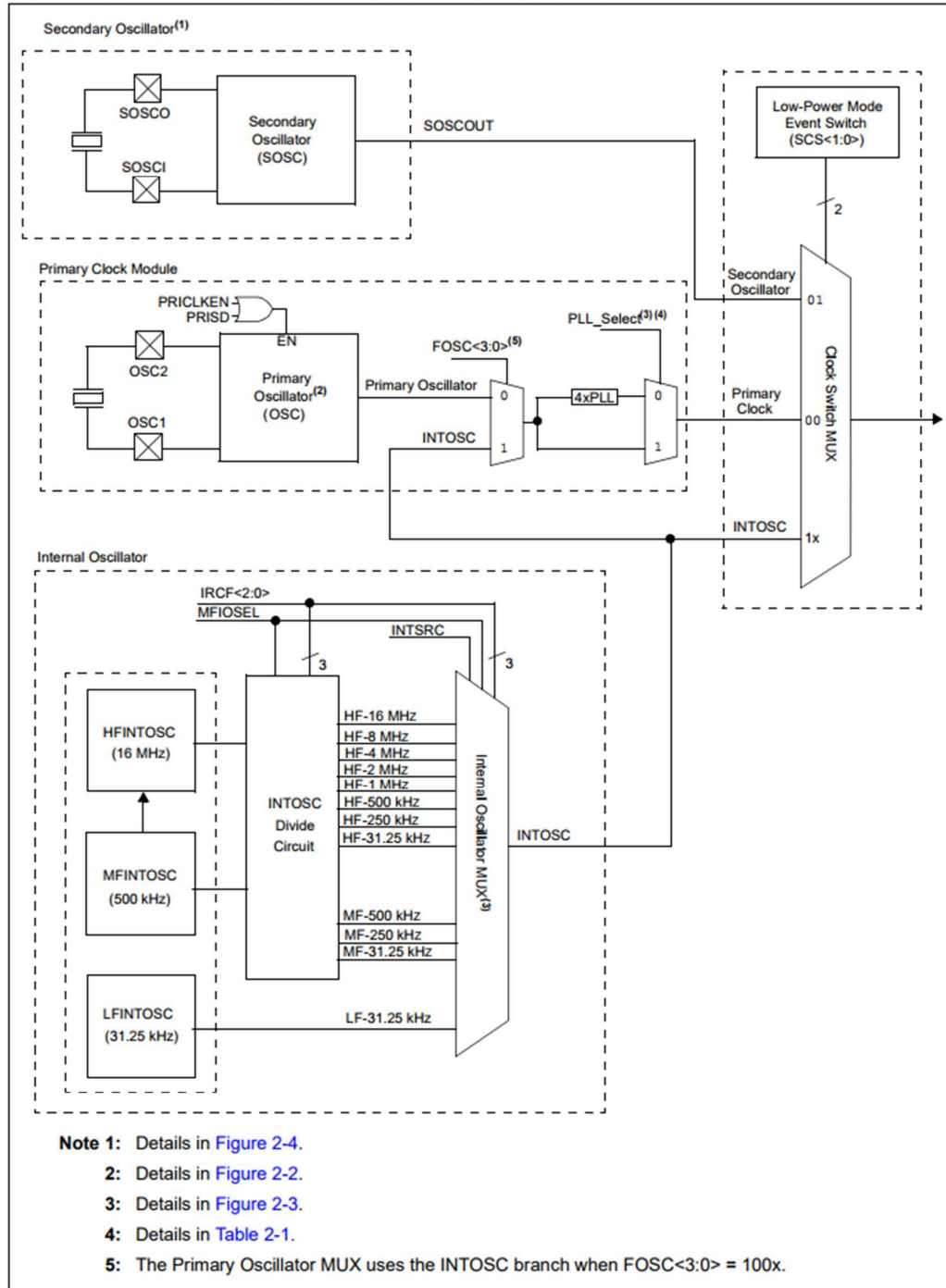
Primary Clock modes are selectable by the FOSC<3:0> bits of the CONFIG1H Configuration register. The primary clock operation is further defined by these Configuration and register bits:

1. PR1CLKEN (CONFIG1H<5>)
2. PRISD (OSCCON2<2>)
3. PLLCFG (CONFIG1H<4>)
4. PLEN (OSCTUNE<6>)
5. HFOFST (CONFIG3H<3>)
6. IRCF<2:0> (OSCCON<6:4>)
7. MFIOSEL (OSCCON2<4>)
8. INTSRC (OSCTUNE<7>)

The HFINTOSC, MFINTOSC and LFINTOSC are factory calibrated high, medium and low-frequency oscillators, respectively, which are used as the internal clock sources.

PIC18(L)F2X/4XK22

FIGURE 2-1: SIMPLIFIED OSCILLATOR SYSTEM BLOCK DIAGRAM



PIC18(L)F2X/4XK22

2.2 Oscillator Control

The OSCCON, OSCCON2 and OSCTUNE registers (Register 2-1 to Register 2-3) control several aspects of the device clock's operation, both in full-power operation and in power-managed modes.

- Main System Clock Selection (SCS)
- Primary Oscillator Circuit Shutdown (PRISD)
- Secondary Oscillator Enable (SOSCGO)
- Primary Clock Frequency 4x multiplier (PLEN)
- Internal Frequency selection bits (IRCF, INTSRC)
- Clock Status bits (OSTS, HFIOFS, MFIOFS, LFIOFS, SOSCRUN, PLLRDY)
- Power management selection (IDLEN)

2.2.1 MAIN SYSTEM CLOCK SELECTION

The System Clock Select bits, SCS<1:0>, select the main clock source. The available clock sources are

- Primary clock defined by the FOSC<3:0> bits of CONFIG1H. The primary clock can be the primary oscillator, an external clock, or the internal oscillator block.
- Secondary clock (secondary oscillator)
- Internal oscillator block (HFINTOSC, MFINTOSC and LFINTOSC).

The clock source changes immediately after one or more of the bits is written to, following a brief clock transition interval. The SCS bits are cleared to select the primary clock on all forms of Reset.

2.2.2 INTERNAL FREQUENCY SELECTION

The Internal Oscillator Frequency Select bits (IRCF<2:0>) select the frequency output of the internal oscillator block. The choices are the LFINTOSC source (31.25 kHz), the MFINTOSC source (31.25 kHz, 250 kHz or 500 kHz) and the HFINTOSC source (16 MHz) or one of the frequencies derived from the HFINTOSC postscaler (31.25 kHz to 8 MHz). If the internal oscillator block is supplying the main clock, changing the states of these bits will have an immediate change on the internal oscillator's output. On device Resets, the output frequency of the internal oscillator is set to the default frequency of 1 MHz.

2.2.3 LOW FREQUENCY SELECTION

When a nominal output frequency of 31.25 kHz is selected (IRCF<2:0> = 000), users may choose which internal oscillator acts as the source. This is done with the INTSRC bit of the OSCTUNE register and MFIOSEL bit of the OSCCON2 register. See Figure 2-2 and Register 2-1 for specific 31.25 kHz selection. This option allows users to select a 31.25 kHz clock (MFINTOSC or HFINTOSC) that can be tuned using the TUN<5:0> bits in OSCTUNE register, while maintaining power savings with a very low clock speed. LFINTOSC always remains the clock source for features such as the Watchdog Timer and the Fail-Safe Clock Monitor, regardless of the setting of INTSRC and MFIOSEL bits

This option allows users to select the tunable and more precise HFINTOSC as a clock source, while maintaining power savings with a very low clock speed.

2.2.4 POWER MANAGEMENT

The IDLEN bit of the OSCCON register determines whether the device goes into Sleep mode or one of the Idle modes when the SLEEP instruction is executed.

PIC18(L)F2X/4XK22

15.0 MASTER SYNCHRONOUS SERIAL PORT (MSSP1 AND MSSP2) MODULE

15.1 Master SSPx (MSSPx) Module Overview

The Master Synchronous Serial Port (MSSPx) 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 MSSPx module can operate in one of two modes:

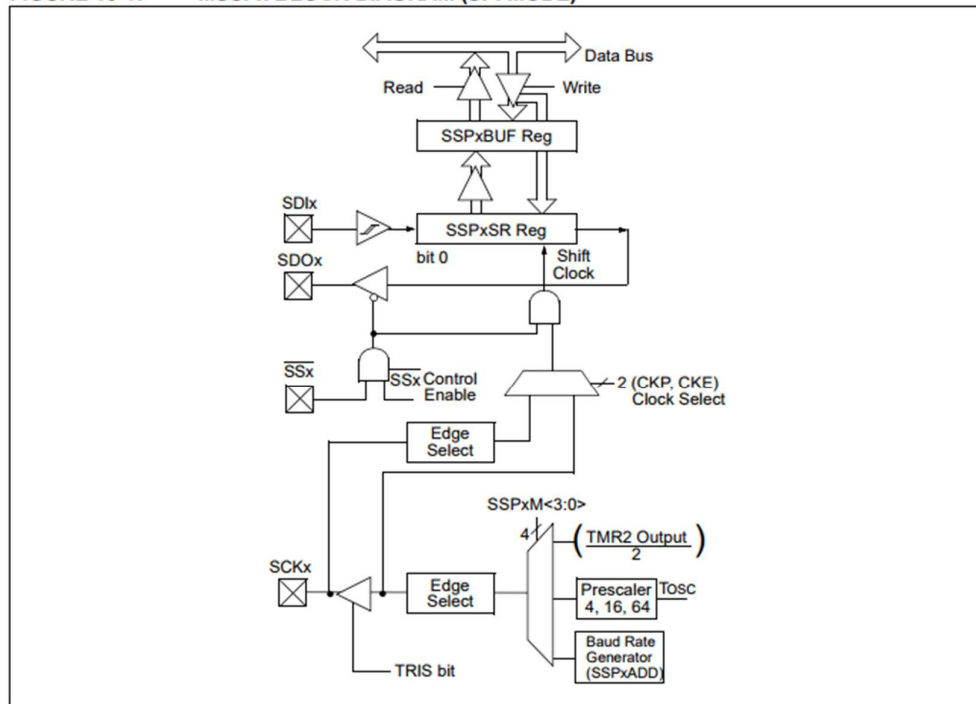
- Serial Peripheral Interface (SPI)
- Inter-Integrated Circuit (I²C)

The SPI interface supports the following modes and features:

- Master mode
- Slave mode
- Clock Parity
- Slave Select Synchronization (Slave mode only)
- Daisy chain connection of slave devices

Figure 15-1 is a block diagram of the SPI interface module.

FIGURE 15-1: MSSPx BLOCK DIAGRAM (SPI MODE)



PIC18(L)F2X/4XK22

The I²C interface supports the following modes and features:

- Master mode
- Slave mode
- Byte NACKing (Slave mode)
- Limited Multi-master support
- 7-bit and 10-bit addressing
- Start and Stop interrupts
- Interrupt masking
- Clock stretching
- Bus collision detection
- General call address matching
- Address masking
- Address Hold and Data Hold modes
- Selectable SDAx hold times

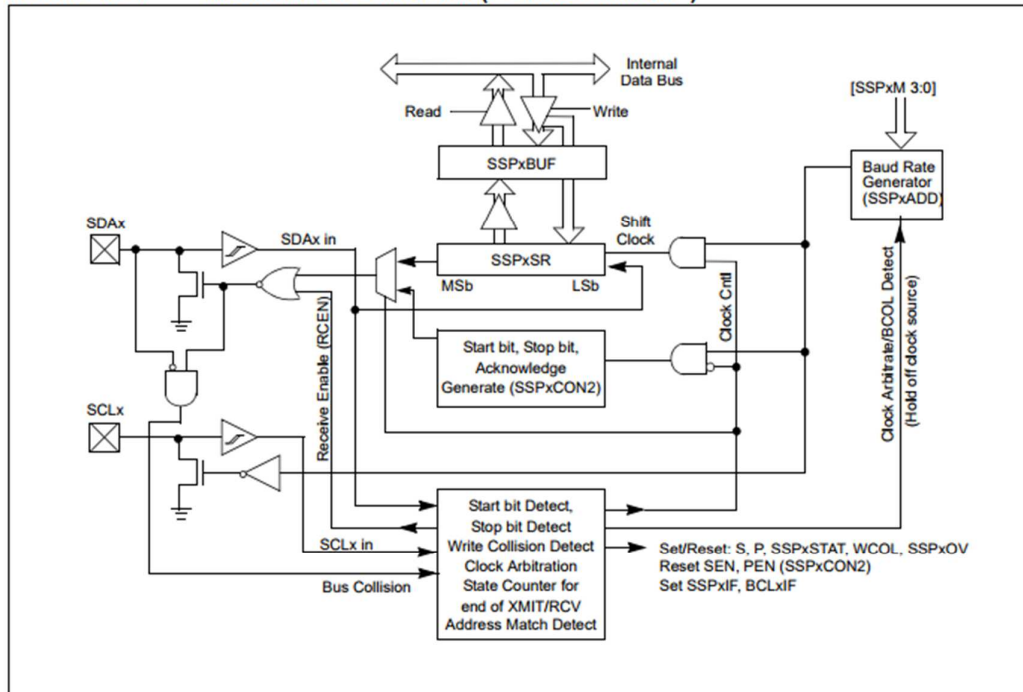
Figure 15-2 is a block diagram of the I²C interface module in Master mode. Figure 15-3 is a diagram of the I²C interface module in Slave mode.

The PIC18(L)F2X/4XK22 has two MSSP modules, MSSP1 and MSSP2, each module operating independently from the other.

Note 1: In devices with more than one MSSP module, it is very important to pay close attention to SSPxCONx register names. SSP1CON1 and SSP1CON2 registers control different operational aspects of the same module, while SSP1CON1 and SSP2CON1 control the same features for two different modules.

2: Throughout this section, generic references to an MSSP module in any of its operating modes may be interpreted as being equally applicable to MSSP1 or MSSP2. Register names, module I/O signals, and bit names may use the generic designator 'x' to indicate the use of a numeral to distinguish a particular module when required.

FIGURE 15-2: MSSPx BLOCK DIAGRAM (I²C MASTER MODE)



PIC18(L)F2X/4XK22

16.0 ENHANCED UNIVERSAL SYNCHRONOUS ASYNCHRONOUS RECEIVER TRANSMITTER (EUSART)

The Enhanced Universal Synchronous Asynchronous Receiver Transmitter (EUSART) module is a serial I/O communications peripheral. It contains all the clock generators, shift registers and data buffers necessary to perform an input or output serial data transfer independent of device program execution. The EUSART, also known as a Serial Communications Interface (SCI), can be configured as a full-duplex asynchronous system or half-duplex synchronous system. Full-Duplex mode is useful for communications with peripheral systems, such as CRT terminals and personal computers. Half-Duplex Synchronous mode is intended for communications with peripheral devices, such as A/D or D/A integrated circuits, serial EEPROMs or other microcontrollers. These devices typically do not have internal clocks for baud rate generation and require the external clock signal provided by a master synchronous device.

The EUSART module includes the following capabilities:

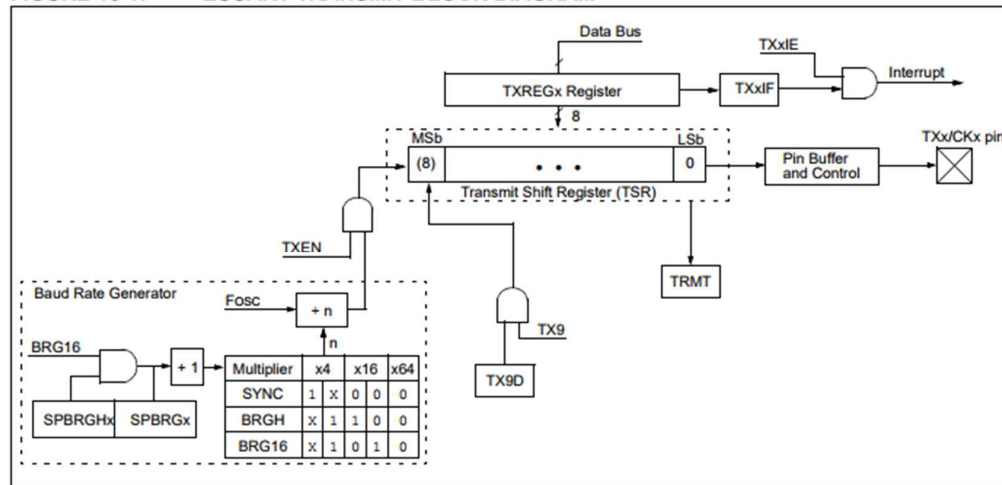
- Full-duplex asynchronous transmit and receive
- Two-character input buffer
- One-character output buffer
- Programmable 8-bit or 9-bit character length
- Address detection in 9-bit mode
- Input buffer overrun error detection
- Received character framing error detection
- Half-duplex synchronous master
- Half-duplex synchronous slave
- Programmable clock and data polarity

The EUSART module implements the following additional features, making it ideally suited for use in Local Interconnect Network (LIN) bus systems:

- Automatic detection and calibration of the baud rate
- Wake-up on Break reception
- 13-bit Break character transmit

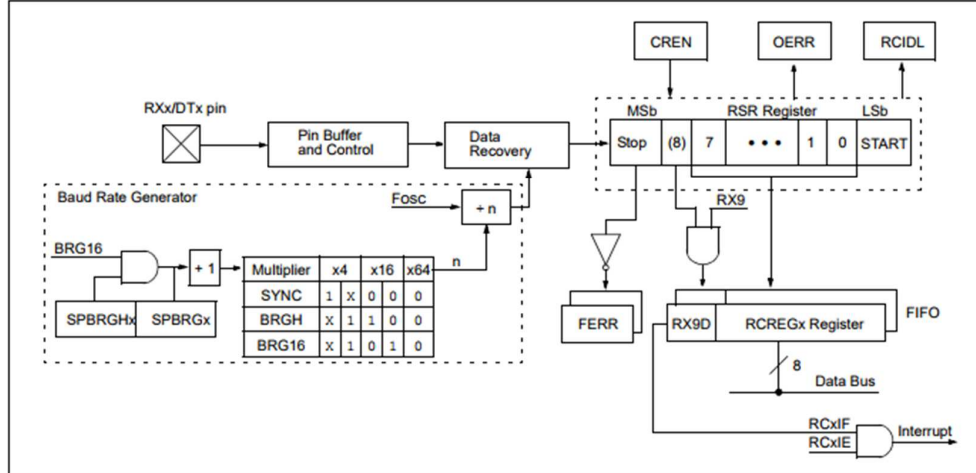
Block diagrams of the EUSART transmitter and receiver are shown in [Figure 16-1](#) and [Figure 16-2](#).

FIGURE 16-1: EUSART TRANSMIT BLOCK DIAGRAM



PIC18(L)F2X/4XK22

FIGURE 16-2: EUSART RECEIVE BLOCK DIAGRAM



The operation of the EUSART module is controlled through three registers:

- Transmit Status and Control (TXSTAx)
- Receive Status and Control (RCSTAx)
- Baud Rate Control (BAUDCONx)

These registers are detailed in [Register 16-1](#), [Register 16-2](#) and [Register 16-3](#), respectively.

For all modes of EUSART operation, the TRIS control bits corresponding to the RXx/DTx and TXx/CKx pins should be set to '1'. The EUSART control will automatically reconfigure the pin from input to output, as needed.

When the receiver or transmitter section is not enabled then the corresponding RXx/DTx or TXx/CKx pin may be used for general purpose input and output.

PIC18(L)F2X/4XK22

27.0 ELECTRICAL SPECIFICATIONS

Absolute Maximum Ratings ^(†)

| | |
|--|-----------------------------------|
| Ambient temperature under bias | -40°C to +125°C |
| Storage temperature | -65°C to +150°C |
| Voltage on any pin with respect to V _{SS} (except V _{DD} , and $\overline{\text{MCLR}}$)..... | -0.3V to (V _{DD} + 0.3V) |
| Voltage on V _{DD} with respect to V _{SS} | |
| PIC18LF24K22 | -0.3V to +4.5V |
| PIC18(L)F26K22 | -0.3V to +6.5V |
| Voltage on $\overline{\text{MCLR}}$ with respect to V _{SS} (Note 2) | 0V to +11.0V |
| Total power dissipation (Note 1)..... | 1.0W |
| Maximum current out of V _{SS} pin (-40°C to +85°C)..... | 300 mA |
| Maximum current out of V _{SS} pin (+85°C to +125°C)..... | 125 mA |
| Maximum current into V _{DD} pin (-40°C to +85°C)..... | 200 mA |
| Maximum current into V _{DD} pin (+85°C to +125°C) | 85 mA |
| Input clamp current, I _{IK} (V _I < 0 or V _I > V _{DD})..... | ±20 mA |
| Output clamp current, I _{OK} (V _O < 0 or V _O > V _{DD}) | ±20 mA |
| Maximum output current sunk by any I/O pin..... | 25 mA |
| Maximum output current sourced by any I/O pin..... | 25 mA |
| Maximum current sunk by all ports (-40°C to +85°C)..... | 200 mA |
| Maximum current sunk by all ports (+85°C to +125°C)..... | 110 mA |
| Maximum current sourced by all ports (-40°C to +85°C)..... | 185 mA |
| Maximum current sourced by all ports (+85°C to +125°C)..... | 70 mA |

Note 1: Power dissipation is calculated as follows:

$$P_{dis} = V_{DD} \times \{I_{DD} - \sum I_{OH}\} + \sum \{(V_{DD} - V_{OH}) \times I_{OH}\} + \sum (V_{OL} \times I_{OL})$$

- 2:** Voltage spikes below V_{SS} at the $\overline{\text{MCLR}}$ /V_{PP}/RE3 pin, inducing currents greater than 80 mA, may cause latch-up. Thus, a series resistor of 50-100Ω should be used when applying a "low" level to the $\overline{\text{MCLR}}$ /V_{PP}/RE3 pin, rather than pulling this pin directly to V_{SS}.

† **NOTICE:** Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at those or any other conditions above those indicated in the operation listings of this specification is not implied. Exposure to maximum rating conditions for extended periods may affect device reliability.

13.2. Anexo Bravo

Hoja de especificaciones MFRC522



MFRC522

Standard performance MIFARE and NTAG frontend

Rev. 3.9 — 27 April 2016
112139

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.

1.1 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.

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 and NTAG.

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)
- I²C-bus interface

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 and NTAG
- 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
 - ◆ I²C-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)} ≤ V _{DDA} = V _{DDD} = V _{DD(TVDD)} ; | [1][2] 2.5 | 3.3 | 3.6 | V |
| V _{DDD} | digital supply voltage | V _{SSA} = V _{SSD} = V _{SS(PVSS)} = V _{SS(TVSS)} = 0 V | 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 |

Table 1. Quick reference data ...continued

| Symbol | Parameter | Conditions | Min | Typ | Max | Unit | |
|----------------|------------------------|--|-----------|-----|-----|--------------------|----|
| I_{pd} | power-down current | $V_{DDA} = V_{DDD} = V_{DD(TVDD)} = V_{DD(PVDD)} = 3\text{ V}$ | | | | | |
| | | hard power-down; pin NRSTPD set LOW | [4] | - | 5 | μA | |
| | | soft power-down; RF level detector on | [4] | - | 10 | μA | |
| I_{DDD} | digital supply current | pin DVDD; $V_{DDD} = 3\text{ V}$ | - | 6.5 | 9 | mA | |
| I_{DDA} | analog supply current | pin AVDD; $V_{DDA} = 3\text{ V}$, CommandReg register's RcvOff bit = 0 | - | 7 | 10 | mA | |
| | | pin AVDD; receiver switched off; $V_{DDA} = 3\text{ V}$, CommandReg register's RcvOff bit = 1 | - | 3 | 5 | mA | |
| $I_{DD(PVDD)}$ | PVDD supply current | pin PVDD | [5] | - | 40 | mA | |
| $I_{DD(TVDD)}$ | TVDD supply current | pin TVDD; continuous wave | [6][7][8] | - | 60 | 100 | mA |
| T_{amb} | ambient temperature | HVQFN32 | -25 | - | +85 | $^{\circ}\text{C}$ | |

- [1] Supply voltages below 3 V reduce the performance in, for example, 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} .
- [4] I_{pd} is the total current for all supplies.
- [5] $I_{DD(PVDD)}$ depends on the overall load at the digital pins.
- [6] $I_{DD(TVDD)}$ depends on $V_{DD(TVDD)}$ and the external circuit connected to pins TX1 and TX2.
- [7] During typical circuit operation, the overall current is below 100 mA.
- [8] Typical value using a complementary driver configuration and an antenna matched to $40\ \Omega$ between pins TX1 and TX2 at 13.56 MHz.

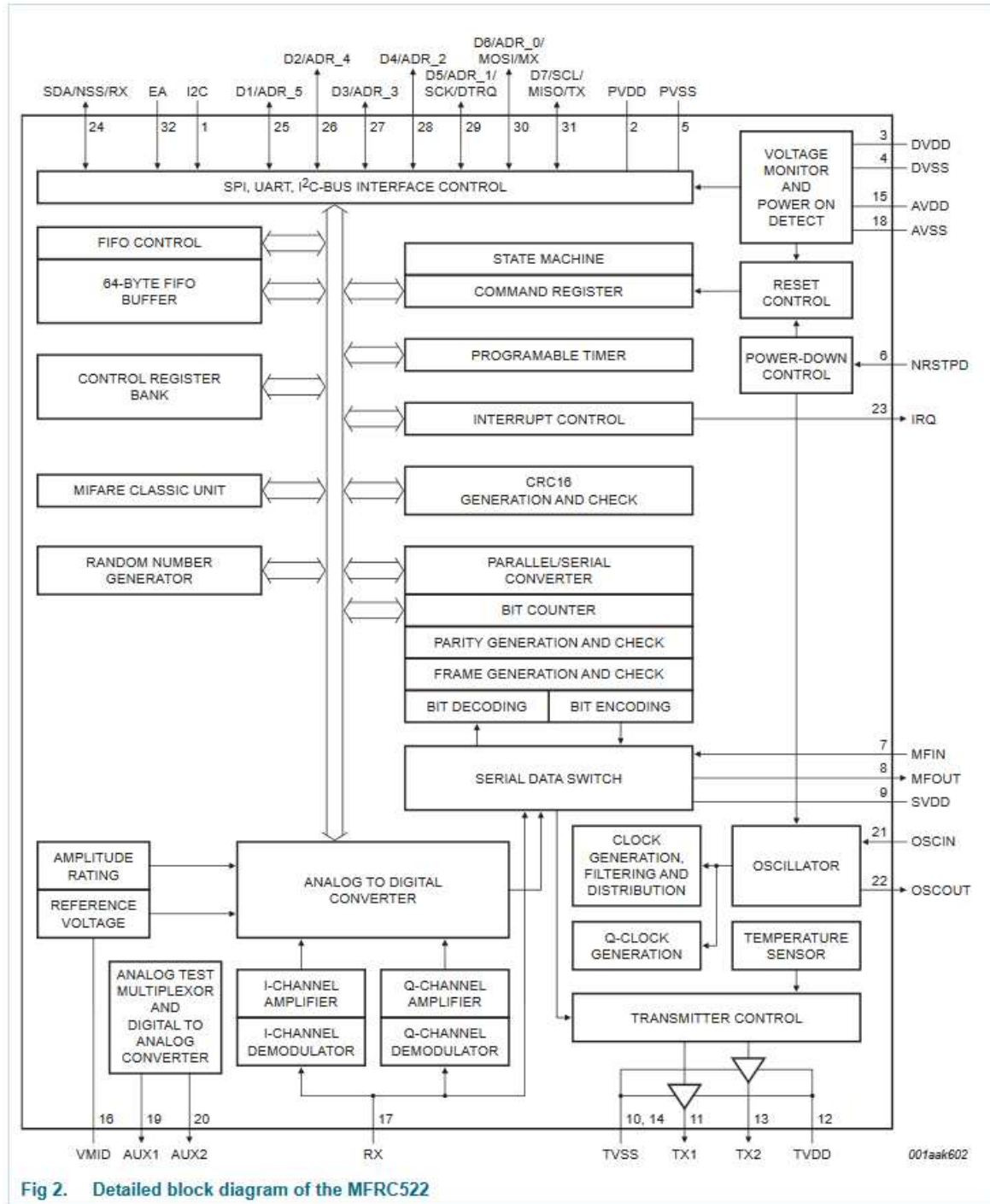


Fig 2. Detailed block diagram of the MFRC522

7. Pinning information

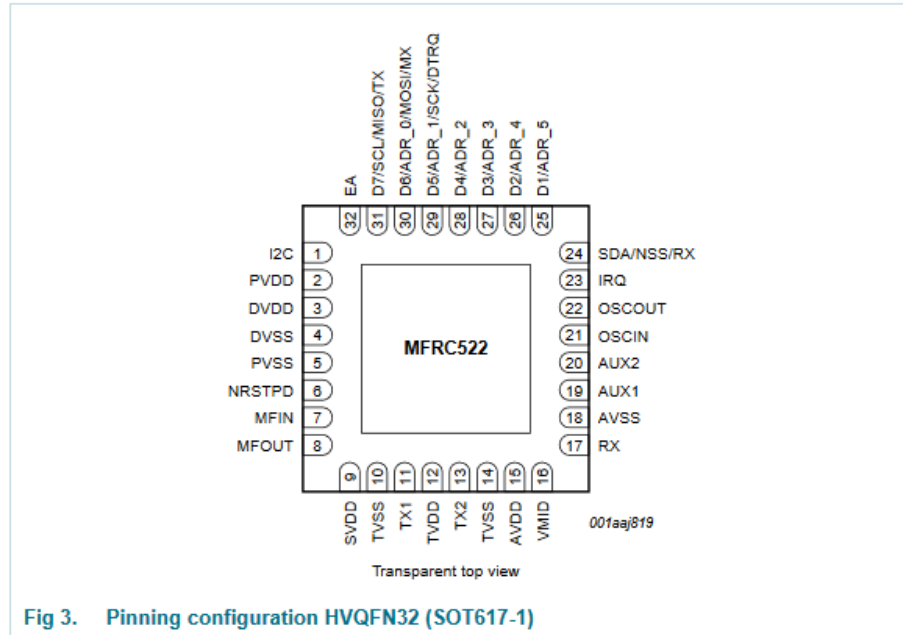


Fig 3. Pinning configuration HVQFN32 (SOT617-1)

7.1 Pin description

Table 3. Pin description

| Pin | Symbol | Type ^[1] | Description |
|-----|--------|---------------------|---|
| 1 | I2C | I | I ² C-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 | OSCOU | O | crystal oscillator inverting amplifier output |
| 23 | IRQ | O | interrupt request output: indicates an interrupt event |
| 24 | SDA | I/O | I ² C-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 | I ² C-bus address 5 input ^[2] |
| 26 | D2 | I/O | test port |
| | ADR_4 | I | I ² C-bus address 4 input ^[2] |
| 27 | D3 | I/O | test port |
| | ADR_3 | I | I ² C-bus address 3 input ^[2] |
| 28 | D4 | I/O | test port |
| | ADR_2 | I | I ² C-bus address 2 input ^[2] |
| 29 | D5 | I/O | test port |
| | ADR_1 | I | I ² C-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 | I ² C-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 | I ² C-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 I ² C-bus address ^[2] |

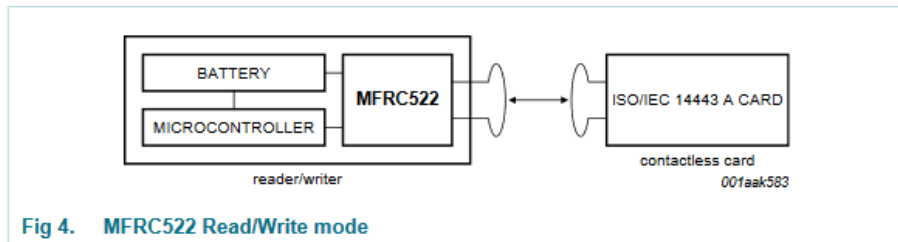
[1] Pin types: I = Input, O = Output, I/O = Input/Output, P = Power and G = Ground.

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

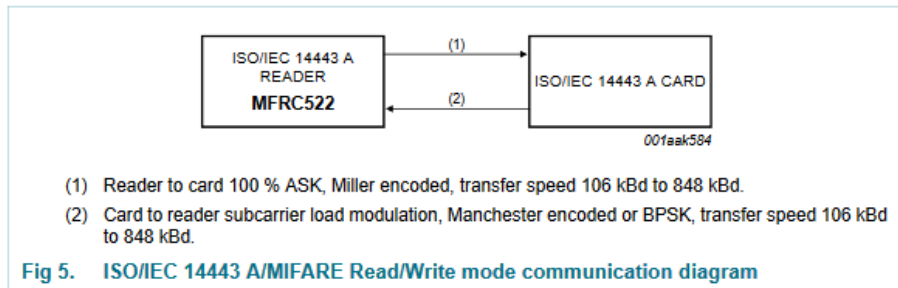
[3] Connection of heatsink pad on package bottom side is not necessary. Optional connection to pin DVSS is possible.

8. Functional description

The MFRC522 transmission module supports the Read/Write mode for ISO/IEC 14443 A/MIFARE using various transfer speeds and modulation protocols.



The physical level communication is shown in [Figure 5](#).



The physical parameters are described in [Table 4](#).

Table 4. Communication overview for ISO/IEC 14443 A/MIFARE reader/writer

| Communication direction | Signal type | Transfer speed | | | |
|---|------------------------|----------------------------|----------------------------|----------------------------|----------------------------|
| | | 106 kBd | 212 kBd | 424 kBd | 848 kBd |
| Reader to card (send data from the MFRC522 to a card) | reader side modulation | 100 % ASK | 100 % ASK | 100 % ASK | 100 % ASK |
| | bit encoding | modified Miller encoding | modified Miller encoding | modified Miller encoding | modified Miller encoding |
| | bit length | 128 (13.56 μs) | 64 (13.56 μs) | 32 (13.56 μs) | 16 (13.56 μs) |
| Card to reader (MFRC522 receives data from a card) | card side modulation | subcarrier load modulation | subcarrier load modulation | subcarrier load modulation | subcarrier load modulation |
| | subcarrier frequency | 13.56 MHz / 16 | 13.56 MHz / 16 | 13.56 MHz / 16 | 13.56 MHz / 16 |
| | bit encoding | Manchester encoding | BPSK | BPSK | BPSK |

The MFRC522's contactless UART and dedicated external host must manage the complete ISO/IEC 14443 A/MIFARE protocol. [Figure 6](#) shows the data coding and framing according to ISO/IEC 14443 A/MIFARE.

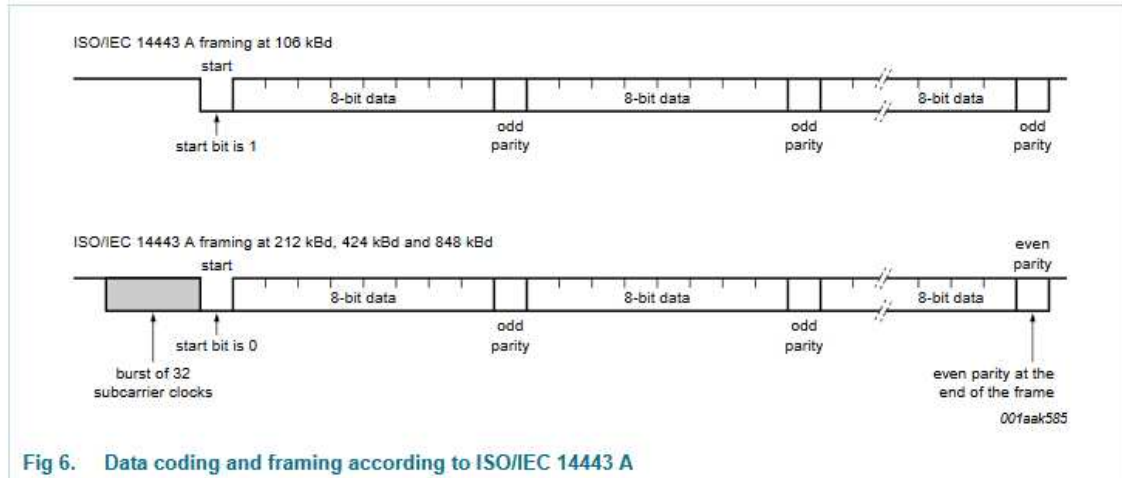


Fig 6. Data coding and framing according to ISO/IEC 14443 A

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) | I ² C-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](#).

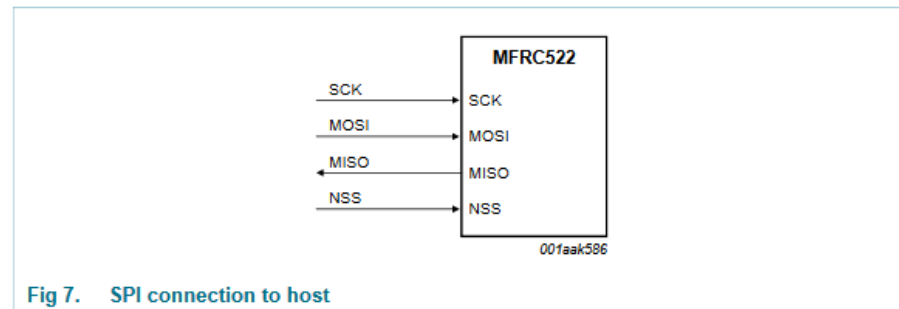


Fig 7. SPI connection to host

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 |

8.3 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:

- ComlEnReg register's LoAlertEn bit is set to logic 1. It activates pin IRQ when Status1Reg register's LoAlert bit changes to logic 1.
- ComlEnReg register's HiAlertEn 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](#):

$$HiAlert = (64 - FIFOLength) \leq WaterLevel \quad (3)$$

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](#):

$$LoAlert = FIFOLength \leq WaterLevel \quad (4)$$

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)} ≤ 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)} ≤ 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)} ≤ 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)} ≤ 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}.

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](#).

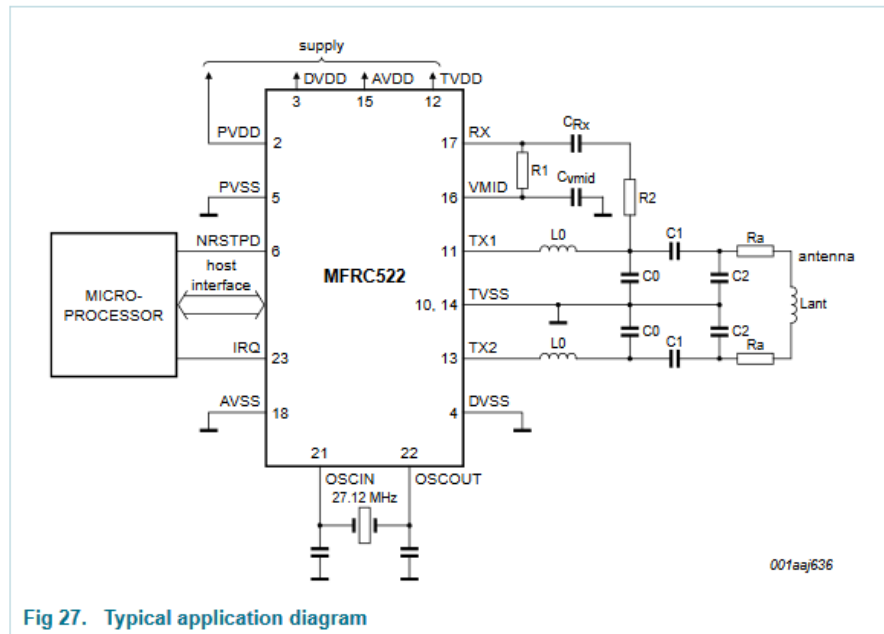


Fig 27. Typical application diagram

13.3. Anexo Charly

Hoja de especificaciones DS1307

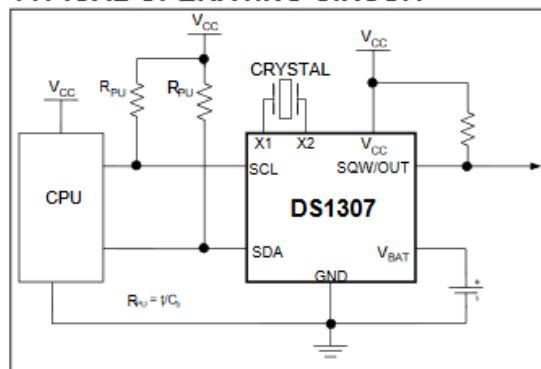


DS1307 64 x 8, Serial, I²C Real-Time Clock

GENERAL DESCRIPTION

The DS1307 serial real-time clock (RTC) is a low-power, full binary-coded decimal (BCD) clock/calendar plus 56 bytes of NV SRAM. Address and data are transferred serially through an I²C, bidirectional bus. The clock/calendar provides seconds, minutes, hours, day, date, month, and year information. The end of the month date 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 AM/PM indicator. The DS1307 has a built-in power-sense circuit that detects power failures and automatically switches to the backup supply. Timekeeping operation continues while the part operates from the backup supply.

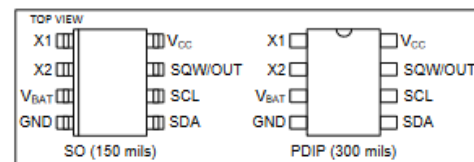
TYPICAL OPERATING CIRCUIT



BENEFITS AND FEATURES

- 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
 - 56-Byte, Battery-Backed, General-Purpose RAM with Unlimited Writes
 - Programmable Square-Wave Output Signal
- Simple Serial Port Interfaces to Most Microcontrollers
 - I²C Serial Interface
- Low Power Operation Extends Battery Backup Run Time
 - Consumes Less than 500nA in Battery-Backup Mode with Oscillator Running
 - Automatic Power-Fail Detect and Switch Circuitry
- 8-Pin DIP and 8-Pin SO Minimizes Required Space
- Optional Industrial Temperature Range: -40°C to +85°C Supports Operation in a Wide Range of Applications
- Underwriters Laboratories® (UL) Recognized

PIN CONFIGURATIONS



ORDERING INFORMATION

| PART | TEMP RANGE | VOLTAGE (V) | PIN-PACKAGE | TOP MARK* |
|--------------|----------------|-------------|-------------------------------|-----------|
| DS1307+ | 0°C to +70°C | 5.0 | 8 PDIP (300 mils) | DS1307 |
| DS1307N+ | -40°C to +85°C | 5.0 | 8 PDIP (300 mils) | DS1307N |
| DS1307Z+ | 0°C to +70°C | 5.0 | 8 SO (150 mils) | DS1307 |
| DS1307ZN+ | -40°C to +85°C | 5.0 | 8 SO (150 mils) | DS1307N |
| DS1307Z+T&R | 0°C to +70°C | 5.0 | 8 SO (150 mils) Tape and Reel | DS1307 |
| DS1307ZN+T&R | -40°C to +85°C | 5.0 | 8 SO (150 mils) Tape and Reel | DS1307N |

+Denotes a lead-free/RoHS-compliant package.

*A "+" anywhere on the top mark indicates a lead-free package. An "N" anywhere on the top mark indicates an industrial temperature range device. Underwriters Laboratories, Inc. is a registered certification mark of Underwriters Laboratories, Inc.

ABSOLUTE MAXIMUM RATINGS

| | |
|---|---|
| Voltage Range on Any Pin Relative to Ground | -0.5V to +7.0V |
| Operating Temperature Range (Noncondensing) | |
| Commercial..... | 0°C to +70°C |
| Industrial | -40°C to +85°C |
| Storage Temperature Range | -55°C to +125°C |
| Soldering Temperature (DIP, leads)..... | +260°C for 10 seconds |
| Soldering Temperature (surface mount)..... | Refer to the JPC/JEDEC J-STD-020 Specification. |

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 the absolute maximum rating conditions for extended periods may affect device reliability.

RECOMMENDED DC OPERATING CONDITIONS

(T_A = 0°C to +70°C, T_A = -40°C to +85°C.) (Notes 1, 2)

| PARAMETER | SYMBOL | CONDITIONS | MIN | TYP | MAX | UNITS |
|----------------------------------|------------------|------------|------|-----|-----------------------|-------|
| Supply Voltage | V _{CC} | | 4.5 | 5.0 | 5.5 | V |
| Logic 1 Input | V _{IH} | | 2.2 | | V _{CC} + 0.3 | V |
| Logic 0 Input | V _{IL} | | -0.3 | | +0.8 | V |
| V _{BAT} Battery Voltage | V _{BAT} | | 2.0 | 3 | 3.5 | V |

DC ELECTRICAL CHARACTERISTICS

(V_{CC} = 4.5V to 5.5V; T_A = 0°C to +70°C, T_A = -40°C to +85°C.) (Notes 1, 2)

| PARAMETER | SYMBOL | CONDITIONS | MIN | TYP | MAX | UNITS |
|---|---------------------|------------|--------------------------|-------------------------|--------------------------|-------|
| Input Leakage (SCL) | I _{LI} | | -1 | | 1 | μA |
| I/O Leakage (SDA, SQW/OUT) | I _{LO} | | -1 | | 1 | μA |
| Logic 0 Output (I _{OL} = 5mA) | V _{OL} | | | | 0.4 | V |
| Active Supply Current (f _{SCL} = 100kHz) | I _{CCA} | | | | 1.5 | mA |
| Standby Current | I _{CCS} | (Note 3) | | | 200 | μA |
| V _{BAT} Leakage Current | I _{BATLKG} | | | 5 | 50 | nA |
| Power-Fail Voltage (V _{BAT} = 3.0V) | V _{PF} | | 1.216 x V _{BAT} | 1.25 x V _{BAT} | 1.284 x V _{BAT} | V |

DC ELECTRICAL CHARACTERISTICS

(V_{CC} = 0V, V_{BAT} = 3.0V; T_A = 0°C to +70°C, T_A = -40°C to +85°C.) (Notes 1, 2)

| PARAMETER | SYMBOL | CONDITIONS | MIN | TYP | MAX | UNITS |
|--|--------------------|------------|-----|-----|-----|-------|
| V _{BAT} Current (OSC ON); SQW/OUT OFF | I _{BAT1} | | | 300 | 500 | nA |
| V _{BAT} Current (OSC ON); SQW/OUT ON (32kHz) | I _{BAT2} | | | 480 | 800 | nA |
| V _{BAT} Data-Retention Current (Oscillator Off) | I _{BATDR} | | | 10 | 100 | nA |

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

TIMING DIAGRAM

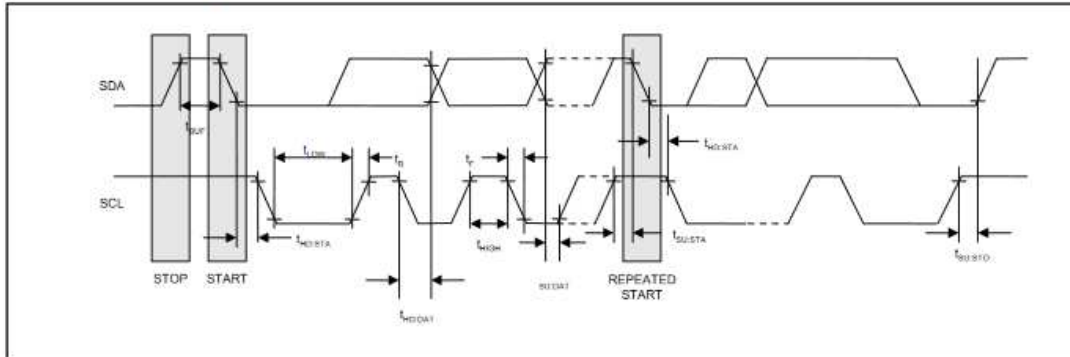
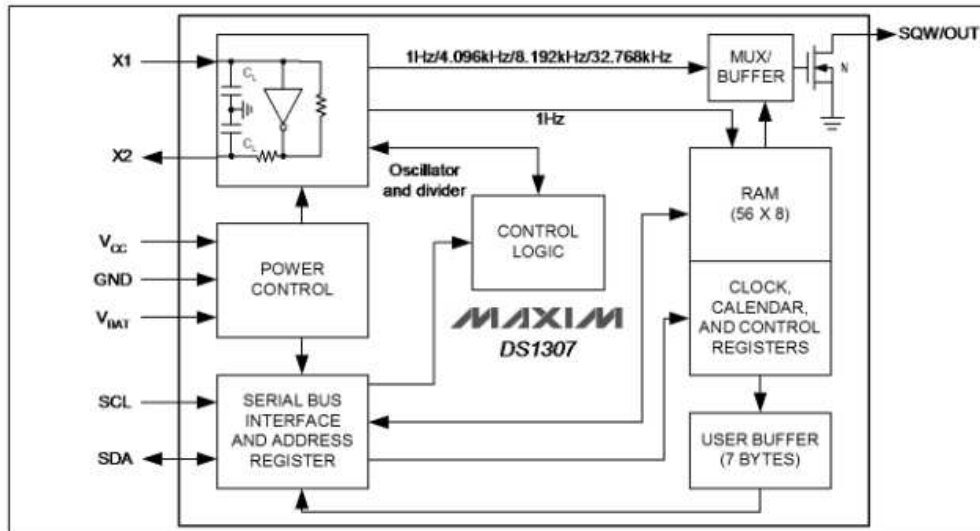


Figure 1. Block Diagram



PIN DESCRIPTION

| PIN | NAME | FUNCTION |
|-----|------------------|--|
| 1 | X1 | Connections for Standard 32.768kHz Quartz Crystal. The internal oscillator circuitry is designed for operation with a crystal having a specified load capacitance (C_L) of 12.5pF. X1 is the input to the oscillator and can optionally be connected to an external 32.768kHz oscillator. The output of the internal oscillator, X2, is floated if an external oscillator is connected to X1. Note: For more information on crystal selection and crystal layout considerations, refer to <i>Application Note 58: Crystal Considerations with Dallas Real-Time Clocks</i> . |
| 2 | X2 | |
| 3 | V _{BAT} | Backup Supply Input for Any Standard 3V Lithium Cell or Other Energy Source. Battery voltage must be held between the minimum and maximum limits for proper operation. Diodes in series between the battery and the V _{BAT} pin may prevent proper operation. If a backup supply is not required, V _{BAT} must be grounded. The nominal power-fail trip point (V _{PF}) voltage at which access to the RTC and user RAM is denied is set by the internal circuitry as 1.25 x V _{BAT} nominal. A lithium battery with 48mAh or greater will back up the DS1307 for more than 10 years in the absence of power at +25°C. UL recognized to ensure against reverse charging current when used with a lithium battery. Go to: www.maxim-ic.com/qa/info/ul/ . |
| 4 | GND | Ground |
| 5 | SDA | Serial Data Input/Output. SDA is the data input/output for the I ² C serial interface. The SDA pin is open drain and requires an external pullup resistor. The pullup voltage can be up to 5.5V regardless of the voltage on V _{CC} . |
| 6 | SCL | Serial Clock Input. SCL is the clock input for the I ² C interface and is used to synchronize data movement on the serial interface. The pullup voltage can be up to 5.5V regardless of the voltage on V _{CC} . |
| 7 | SQW/OUT | Square Wave/Output Driver. When enabled, the SQWE bit set to 1, the SQW/OUT pin outputs one of four square-wave frequencies (1Hz, 4kHz, 8kHz, 32kHz). The SQW/OUT pin is open drain and requires an external pullup resistor. SQW/OUT operates with either V _{CC} or V _{BAT} applied. The pullup voltage can be up to 5.5V regardless of the voltage on V _{CC} . If not used, this pin can be left floating. |
| 8 | V _{CC} | Primary Power Supply. When voltage is applied within normal limits, the device is fully accessible and data can be written and read. When a backup supply is connected to the device and V _{CC} is below V _{TP} , read and writes are inhibited. However, the timekeeping function continues unaffected by the lower input voltage. |

DETAILED DESCRIPTION

The DS1307 is a low-power clock/calendar with 56 bytes of battery-backed SRAM. 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 DS1307 operates as a slave device on the I²C bus. Access is obtained by implementing a START condition and providing a device identification code followed by a register address. Subsequent registers can be accessed sequentially until a STOP condition is executed. When V_{CC} falls below 1.25 x V_{BAT}, the device terminates an access in progress and resets the device address counter. Inputs to the device will not be recognized at this time to prevent erroneous data from being written to the device from an out-of-tolerance system. When V_{CC} falls below V_{BAT}, the device switches into a low-current battery-backup mode. Upon power-up, the device switches from battery to V_{CC} when V_{CC} is greater than V_{BAT} +0.2V and recognizes inputs when V_{CC} is greater than 1.25 x V_{BAT}. The block diagram in Figure 1 shows the main elements of the serial RTC.

OSCILLATOR CIRCUIT

The DS1307 uses an external 32.768kHz crystal. The oscillator circuit does not require any external resistors or capacitors to operate. Table 1 specifies several crystal parameters for the external crystal. Figure 1 shows a functional schematic of the oscillator circuit. If using a crystal with the specified characteristics, the startup time is usually less than one second.

CLOCK ACCURACY

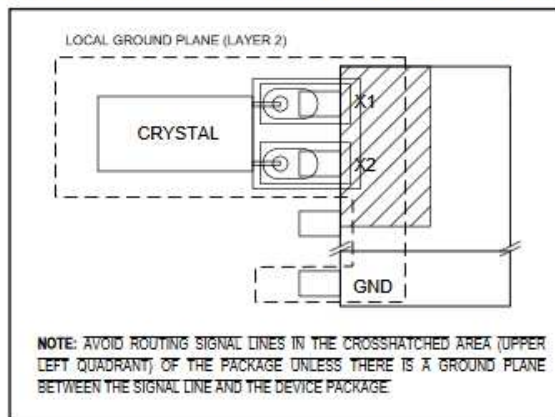
The accuracy of the clock is dependent upon the accuracy of the crystal and the accuracy of the match between the capacitive load of the oscillator circuit and the capacitive load for which the crystal was trimmed. Additional error will be added by crystal frequency drift caused by temperature shifts. External circuit noise coupled into the oscillator circuit may result in the clock running fast. Refer to Application Note 58: *Crystal Considerations with Dallas Real-Time Clocks* for detailed information.

Table 1. Crystal Specifications*

| PARAMETER | SYMBOL | MIN | TYP | MAX | UNITS |
|-------------------|--------|-----|--------|-----|------------|
| Nominal Frequency | f_0 | | 32.768 | | kHz |
| Series Resistance | ESR | | | 45 | k Ω |
| Load Capacitance | C_L | | 12.5 | | pF |

*The crystal, traces, and crystal input pins should be isolated from RF generating signals. Refer to Application Note 58: *Crystal Considerations for Dallas Real-Time Clocks* for additional specifications.

Figure 2. Recommended Layout for Crystal



RTC AND RAM ADDRESS MAP

Table 2 shows the address map for the DS1307 RTC and RAM registers. The RTC registers are located in address locations 00h to 07h. The RAM registers are located in address locations 08h to 3Fh. During a multibyte access, when the address pointer reaches 3Fh, the end of RAM space, it wraps around to location 00h, the beginning of the clock space.

CLOCK AND CALENDAR

The time and calendar information is obtained by reading the appropriate register bytes. Table 2 shows the RTC registers. The time and calendar are set or initialized by writing the appropriate register bytes. The contents of the time and calendar registers are in the BCD format. 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. Bit 7 of Register 0 is the clock halt (CH) bit. When this bit is set to 1, the oscillator is disabled. When cleared to 0, the oscillator is enabled. On first application of power to the device the time and date registers are typically reset to 01/01/00 01 00:00:00 (MM/DD/YY DOW HH:MM:SS). The CH bit in the seconds register will be set to a 1. The clock can be halted whenever the timekeeping functions are not required, which minimizes current (I_{BATDR}).

The DS1307 can be run in either 12-hour or 24-hour mode. Bit 6 of the hours register is defined as the 12-hour 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 second 10-hour bit (20 to 23 hours). The hours value must be re-entered whenever the 12/24-hour mode bit is changed.

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 I²C START. The time information is read from these secondary registers while the clock continues to run. This eliminates the need to re-read the registers in case the internal registers update during a read. The divider chain is reset whenever the seconds register is written. Write transfers occur on the I²C acknowledge from the DS1307. Once the divider chain is reset, to avoid rollover issues, the remaining time and date registers must be written within one second.

Table 2. Timekeeper Registers

| ADDRESS | BIT 7 | BIT 6 | BIT 5 | BIT 4 | BIT 3 | BIT 2 | BIT 1 | BIT 0 | FUNCTION | RANGE |
|---------|---------|------------|-----------|----------|---------|-------|-------|-------|---------------|-------------------------|
| 00h | CH | 10 Seconds | | | Seconds | | | | Seconds | 00–59 |
| 01h | 0 | 10 Minutes | | | Minutes | | | | Minutes | 00–59 |
| 02h | 0 | 12 | 10 Hour | 10 Hour | Hours | | | | Hours | 1–12 +AM/PM 00–23 |
| | | 24 | PM/ AM | | | | | | | |
| 03h | 0 | 0 | 0 | 0 | DAY | | | | Day | 01–07 |
| 04h | 0 | 0 | 10 Date | | Date | | | | Date | 01–31 |
| 05h | 0 | 0 | 0 | 10 Month | Month | | | | Month | 01–12 |
| 06h | 10 Year | | | Year | | | | Year | 00–99 | |
| 07h | OUT | 0 | 0 | SQWE | 0 | 0 | RS1 | RS0 | Control | — |
| 08h–3Fh | | | | | | | | | RAM 56 x 8 | 00h–FFh |

0 = Always reads back as 0.

CONTROL REGISTER

The DS1307 control register is used to control the operation of the SQW/OUT pin.

| BIT 7 | BIT 6 | BIT 5 | BIT 4 | BIT 3 | BIT 2 | BIT 1 | BIT 0 |
|-------|-------|-------|-------|-------|-------|-------|-------|
| OUT | 0 | 0 | SQWE | 0 | 0 | RS1 | RS0 |

Bit 7: Output Control (OUT). This bit controls the output level of the SQW/OUT pin when the square-wave output is disabled. If SQWE = 0, the logic level on the SQW/OUT pin is 1 if OUT = 1 and is 0 if OUT = 0. On initial application of power to the device, this bit is typically set to a 0.

Bit 4: Square-Wave Enable (SQWE). This bit, when set to logic 1, enables the oscillator output. The frequency of the square-wave output depends upon the value of the RS0 and RS1 bits. With the square-wave output set to 1Hz, the clock registers update on the falling edge of the square wave. On initial application of power to the device, this bit is typically set to a 0.

Bits 1 and 0: Rate Select (RS[1:0]). These bits control the frequency of the square-wave output when the square-wave output has been enabled. The following table lists the square-wave frequencies that can be selected with the RS bits. On initial application of power to the device, these bits are typically set to a 1.

| RS1 | RS0 | SQW/OUT OUTPUT | SQWE | OUT |
|-----|-----|----------------|------|-----|
| 0 | 0 | 1Hz | 1 | X |
| 0 | 1 | 4.096kHz | 1 | X |
| 1 | 0 | 8.192kHz | 1 | X |
| 1 | 1 | 32.768kHz | 1 | X |
| X | X | 0 | 0 | 0 |
| X | X | 1 | 0 | 1 |

I²C DATA BUS

The DS1307 supports the I²C protocol. A device that sends data onto the bus is defined as a transmitter and a device receiving data as a receiver. The device that controls the message is called a master. The devices that are controlled by the master are referred to as 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 DS1307 operates as a slave on the I²C bus.

Figures 3, 4, and 5 detail how data is transferred on the I²C bus.

- Data transfer can 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 will be 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 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 the 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 START and 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. Within the I²C bus specifications a standard mode (100kHz clock rate) and a fast mode (400kHz clock rate) are defined. The DS1307 operates in the standard mode (100kHz) only.

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 generating 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.

13.4. Anexo Delta

Hoja de especificaciones CP2102



CP2102/9

SINGLE-CHIP USB-TO-UART BRIDGE

For newer designs, the CP2102N devices offer compatible footprints and are recommended for use instead of the CP2102/9. See the Silicon Labs website (www.silabs.com/usbxpress) for more information.

Single-Chip USB to UART Data Transfer

- Integrated USB transceiver; no external resistors required
- Integrated clock; no external crystal required
- Internal 1024-byte programmable ROM for vendor ID, product ID, serial number, power descriptor, release number, and product description strings
 - EEPROM (CP2102)
 - EPROM (One-time programmable) (CP2109)
- On-chip power-on reset circuit
- On-chip voltage regulator
 - 3.3 V output (CP2102)
 - 3.45 V output (CP2109)
- 100% pin and software compatible with CP2101

USB Function Controller

- USB Specification 2.0 compliant; full-speed (12 Mbps)
- USB suspend states supported via SUSPEND pins

Asynchronous Serial Data BUS (UART)

- All handshaking and modem interface signals
- Data formats supported:
 - Data bits: 5, 6, 7, and 8
 - Stop bits: 1, 1.5, and 2
 - Parity: odd, even, mark, space, no parity
- Baud rates: 300 bps to 1 Mbps
- 576 Byte receive buffer; 640 byte transmit buffer
- Hardware or X-On/X-Off handshaking supported
- Event character support
- Line break transmission

Virtual COM Port Device Drivers

- Works with existing COM port PC Applications
- Royalty-free distribution license
- Windows 8/7/Vista/Server 2003/XP/2000
- Mac OS-X/OS-9
- Linux

USBXpress™ Direct Driver Support

- Royalty-Free Distribution License
- Windows 7/Vista/XP/Server 2003/2000
- Windows CE

Example Applications

- Upgrade of RS-232 legacy devices to USB
- Cellular phone USB interface cable
- USB interface cable
- USB to RS-232 serial adapter

Supply Voltage

- Self-powered: 3.0 to 3.6 V
- USB bus powered: 4.0 to 5.25 V

Package

- RoHS-compliant 28-pin QFN (5x5 mm)

Ordering Part Numbers

- CP2102-GM
- CP2109-A01-GM

Temperature Range: -40 to +85 °C

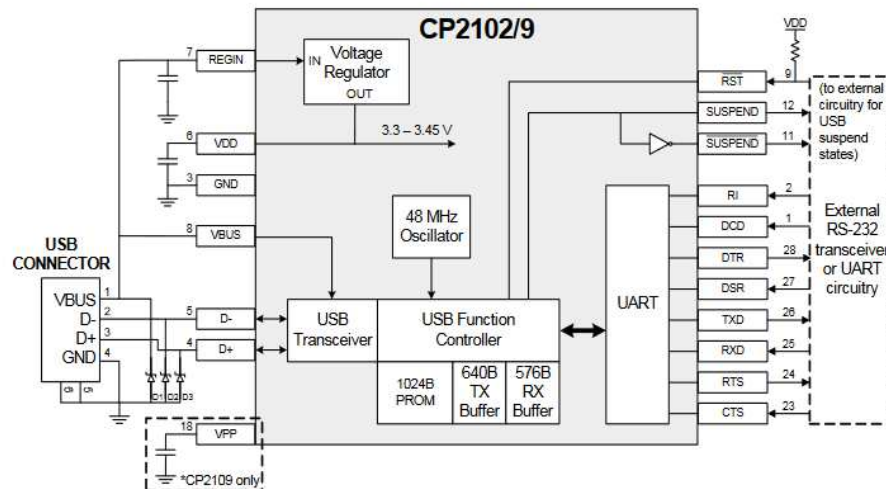


Figure 1. Example System Diagram

CP2102/9

1. System Overview

The CP2102/9 is a highly-integrated USB-to-UART Bridge Controller providing a simple solution for updating RS-232 designs to USB using a minimum of components and PCB space. The CP2102/9 includes a USB 2.0 full-speed function controller, USB transceiver, oscillator, EEPROM or EPROM, and asynchronous serial data bus (UART) with full modem control signals in a compact 5 x 5 mm QFN-28 package. No other external USB components are required.

The on-chip programmable ROM may be used to customize the USB Vendor ID, Product ID, Product Description String, Power Descriptor, Device Release Number, and Device Serial Number as desired for OEM applications. The programmable ROM is programmed on-board via the USB, allowing the programming step to be easily integrated into the product manufacturing and testing process.

Royalty-free Virtual COM Port (VCP) device drivers provided by Silicon Laboratories allow a CP2102/9-based product to appear as a COM port to PC applications. The CP2102/9 UART interface implements all RS-232 signals, including control and handshaking signals, so existing system firmware does not need to be modified. In many existing RS-232 designs, all that is required to update the design from RS-232 to USB is to replace the RS-232 level-translator with the CP2102/9. Direct access driver support is available through the Silicon Laboratories USBXpress driver set.

An evaluation kit for the CP2102 (Part Number: CP2102EK) is available. The kit includes a CP2102-based USB-to-UART/RS-232 evaluation board, a complete set of VCP device drivers, USB and RS-232 cables, and full documentation. Contact a Silicon Labs sales representative or go to www.silabs.com to order the CP2102 Evaluation Kit. The CP2102 Evaluation Kit serves as an evaluation kit for both the CP2102 and CP2109.

CP2102/9

3. Electrical Specifications

Table 2. Absolute Maximum Ratings

| Parameter | Symbol | Test Condition | Min | Typ | Max | Unit |
|--|------------|----------------------------|------|-----|----------------|------|
| Ambient Temperature under Bias | T_{BIAS} | | -55 | — | 125 | °C |
| Storage Temperature | T_{STG} | | -65 | — | 150 | °C |
| Voltage on V_{DD} with respect to GND | V_{DD} | | -0.3 | — | 4.2 | V |
| Maximum Total Current through V_{DD} and GND | | | — | — | 500 | mA |
| Maximum Output Current sunk by \overline{RST} or any I/O pin | | | — | — | 100 | mA |
| CP2102 | | | | | | |
| Voltage on any I/O Pin, V_{BUS} , or \overline{RST} with respect to GND | | | -0.3 | — | 5.8 | V |
| CP2109 | | | | | | |
| Voltage on any I/O Pin, V_{BUS} , or \overline{RST} with respect to GND | | $V_{DD} \geq 3.0\text{ V}$ | -0.3 | — | 5.8 | V |
| | | V_{DD} not powered | -0.3 | — | $V_{DD} + 3.6$ | |
| Note: Stresses above those listed may cause permanent device damage. This is a stress rating only, and functional operation of the devices at or exceeding the conditions in the operation listings of this specification is not implied. Exposure to maximum rating conditions for extended periods may affect device reliability. | | | | | | |

4. Pinout and Package Definitions

Table 9. CP2102/9 Pin Definitions

| Name | Pin # | Type | Description |
|--|-----------------|-----------|--|
| V _{DD} | 6 | Power In | 3.0–3.6 V Power Supply Voltage Input. |
| | | Power Out | 3.3 V Voltage Regulator Output. See "10. Voltage Regulator" on page 19. |
| GND | 3 | | Ground |
| RST | 9 | D I/O | Device Reset. Open-drain output of internal POR or V _{DD} monitor. An external source can initiate a system reset by driving this pin low for at least 15 μs. |
| REGIN | 7 | Power In | 5 V Regulator Input. This pin is the input to the on-chip voltage regulator. |
| VBUS | 8 | D In | VBUS Sense Input. This pin should be connected to the VBUS signal of a USB network. A 5 V signal on this pin indicates a USB network connection. |
| NC ¹ / V _{PP} ² | 18 | A Power | This pin should be left unconnected or tied to V _{DD} . This pin is unused on the CP2102 and may be connected to the V _{pp} programming capacitor to maintain board compatibility with the CP2109. V _{PP} Programming Supply Voltage |
| | | | |
| D+ | 4 | D I/O | USB D+ |
| D– | 5 | D I/O | USB D– |
| TXD | 26 | D Out | Asynchronous data output (UART Transmit) |
| RXD | 25 | D In | Asynchronous data input (UART Receive) |
| CTS | 23 ³ | D In | Clear To Send control input (active low) |
| RTS | 24 ³ | D Out | Ready to Send control output (active low) |
| DSR | 27 ³ | D in | Data Set Ready control input (active low) |
| DTR | 28 ³ | D Out | Data Terminal Ready control output (active low) |
| DCD | 1 ³ | D In | Data Carrier Detect control input (active low) |
| RI | 2 ³ | D In | Ring Indicator control input (active low) |
| SUSPEND | 12 ³ | D Out | This pin is driven high when the CP2102/9 enters the USB suspend state. |
| $\overline{\text{SUSPEND}}$ | 11 ³ | D Out | This pin is driven low when the CP2102/9 enters the USB suspend state. |
| NC | 10, 13–22 | | These pins should be left unconnected or tied to V _{DD} . |
| Notes: 1. For CP2102, pin is no connect (NC). 2. For CP2109, pin is V _{PP} . V _{PP} can be left unconnected when not used for in-application programming. 3. Pins can be left unconnected when not used. | | | |

CP2102/9

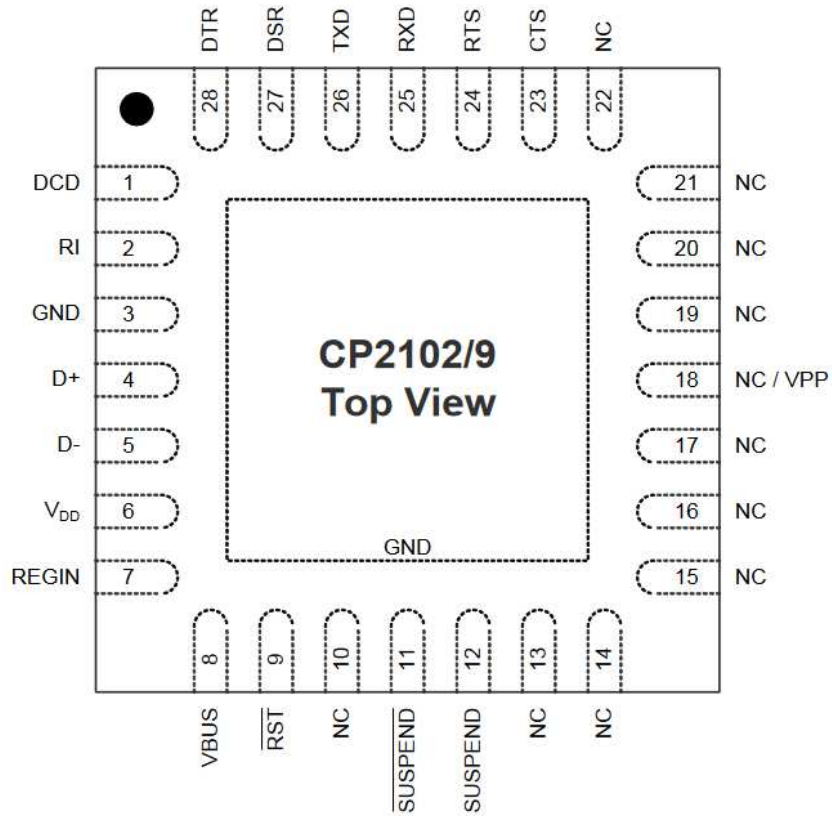


Figure 2. QFN-28 Pinout Diagram (Top View)

CP2102/9

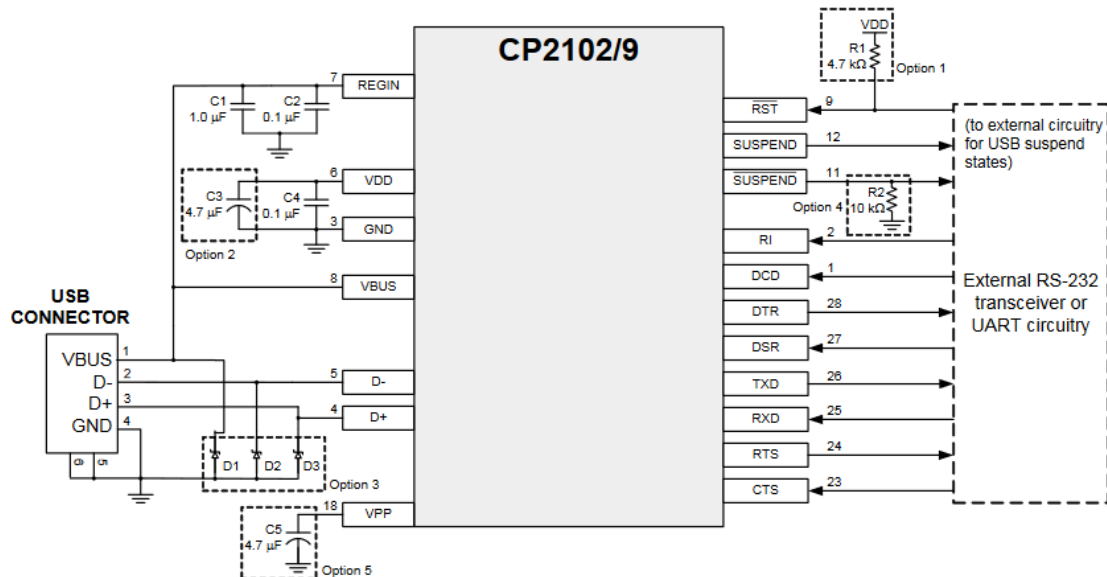
6. USB Function Controller and Transceiver

The Universal Serial Bus function controller in the CP2102/9 is a USB 2.0 compliant full-speed device with integrated transceiver and on-chip matching and pull-up resistors. The USB function controller manages all data transfers between the USB and the UART as well as command requests generated by the USB host controller and commands for controlling the function of the UART.

The USB Suspend and Resume signals are supported for power management of both the CP2102/9 device as well as external circuitry. The CP2102/9 will enter Suspend mode when Suspend signaling is detected on the bus. On entering Suspend mode, the CP2102/9 asserts the SUSPEND and SUSPEND signals. SUSPEND and SUSPEND are also asserted after a CP2102/9 reset until device configuration during USB Enumeration is complete.

The CP2102/9 exits Suspend mode when any of the following occur: (1) Resume signaling is detected or generated, (2) a USB Reset signal is detected, or (3) a device reset occurs. On exit of Suspend mode, the SUSPEND and SUSPEND signals are de-asserted.

Both SUSPEND and SUSPEND temporarily float high during a CP2102/9 reset. If this behavior is undesirable, a strong pull-down (10 kΩ) can be used to ensure SUSPEND remains low during reset. See Figure 5 for other recommended options.



- Option 1: A 4.7 kΩ pull-up resistor can be added to increase noise immunity.
- Option 2: A 4.7 μF capacitor can be added if powering other devices from the on-chip regulator.
- Option 3: Avalanche transient voltage suppression diodes should be added for ESD protection. Use Littelfuse p/n SP0503BAHT or equivalent.
- Option 4: 10 kΩ resistor to ground to hold SUSPEND low on initial power on or device reset.
- Option 5: A 4.7 μF capacitor can be added for in-system programming (CP2109 only).

Figure 5. Typical Connection Diagram

CP2102/9

7. Asynchronous Serial Data Bus (UART) Interface

The CP2102/9 UART interface consists of the TX (transmit) and RX (receive) data signals as well as the RTS, CTS, DSR, DTR, DCD, and RI control signals. The UART supports RTS/CTS, DSR/DTR, and X-On/X-Off handshaking.

The UART is programmable to support a variety of data formats and baud rates. If the Virtual COM Port drivers are used, the data format and baud rate are set during COM port configuration on the PC. If the USBXpress drivers are used, the CP2102/9 is configured through the USBXpress API. The data formats and baud rates available are listed in Table 12.

Table 12. Data Formats and Baud Rates

| | |
|---|---|
| Data Bits | 5, 6, 7, and 8 |
| Stop Bits | 1, 1.5 ¹ , and 2 |
| Parity Type | None, Even, Odd, Mark, Space |
| Baud Rates² | 300, 600, 1200, 1800, 2400, 4000, 4800, 7200, 9600, 14400, 16000, 19200, 28800, 38400, 51200, 56000, 57600, 64000, 76800, 115200, 128000, 153600, 230400, 250000, 256000, 460800, 500000, 576000, 921600 ³ |
| Notes: | |
| <ol style="list-style-type: none"> 1. 5-bit only. 2. Additional baud rates are supported. See "AN721: CP210x/CP211x Device Customization Guide". 3. 7 or 8 data bits only. | |