IPexRFID

Manual lector BLE

Odin

Revisión 1

Septiembre 2022

Odin Solutions S.L.

Odin Solutions, S.L.
Calle Palma de Mallorca 2
30009 - Murcia
Tlf.: +34 902 570 121
E-mail: info@odins.es
Web: www.odins.es

IPexRFID

Manual del lector BLE

INDICE

FIGURAS

No se encontraron elementos de tabla de contenido.

FOTOS

TABLAS

1. Introducción

Este manual está diseñado para ayudar a entender el funcionamiento del lector BLE (Bluetooth Low Energy) y cómo utilizarlo adecuadamente.

El lector está desarrollado con tecnología Bluetooth Low Energy v5.0, para ser utilizado como medio alternativo de identificación en sistemas de control de accesos. Este sistema viene a reemplazar los antiguos sistemas de identificación mediante tarjetas chip o MiFare, dotando de mayor comodidad al usuario al poder utilizar su propio móvil un una APP para identificarse, ya sea de forma automatizada o por intervención del usuario (depende de la APP utilizada).

El sistema de autenticación se basa en la validación de un token, generado en el instante que el usuario necesita identificarse, contra un servidor de autenticación. Estos token son de validez reducida, para impedir que puedan replicarse por captura del mismo, lo que reduce el riesgo de uso fraudulento. Así mismo, el propio token contiene información extra (de forma segura), para poder autenticar el usuario en caso de estar caído el servidor de autenticación. Todo esto se verá más detalladamente en los siguientes apartados.

2. Especificaciones técnicas

- Tecnología bluetooth 5.0.
- Comunicación por uart 3.3v a 19200bps 8,n,1
- Interfaz de comandos AT por uart para configuración y lectura tokens.
- Alimentación a 3.3V/100mA.
- Dimensiones 50x30mm.
- Conexionado por conector JST de paso 1mm.

3. Conexionado

Como se ha mencionado anteriormente el conexionado del lector se realiza mediante un conector JST de paso 1mm, por lo que no hay lugar a error de conexionado, pero igualmente si es necesario conectarlo a un dispositivo que no es un IPexRFID a continuación se describe el pinout de dicho conector para poder realizar el conector adecuado a nuestras necesidades.

La referencia del conector JST para realizar el cable de conexión es **SHR-04V-S-B** y los contactos para los cables es **SSH-003T-P0.2-H.**

3.1. Pinout Lector

En la siguiente foto se puede ver el pinout del conector.

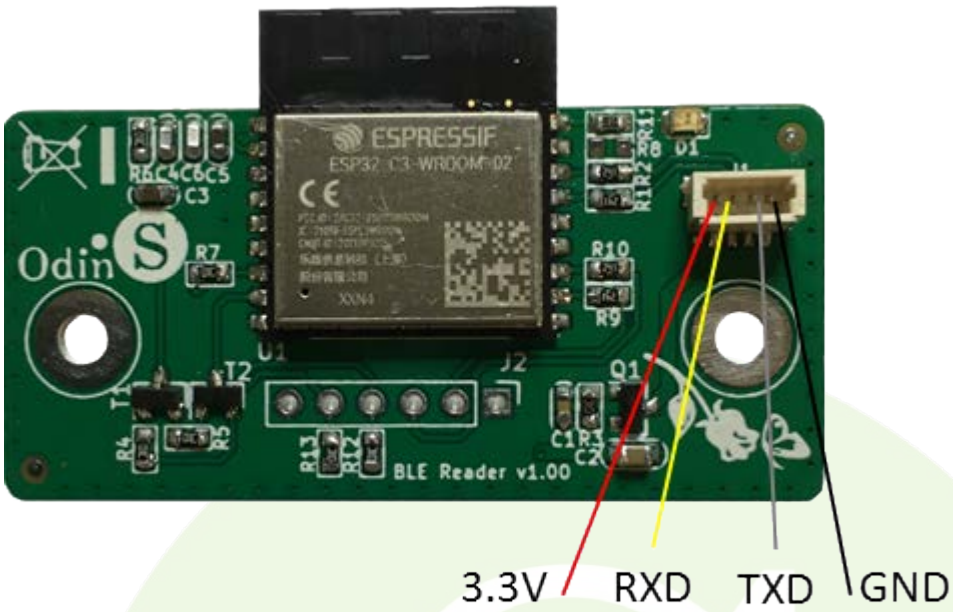

Foto 1. Pinout lector BLE

- 3.3V. Alimentación a 3.3VDC
- RXD. Recepción de la uart del lector (niveles TTL 3.3V).
- TXD. Transmisión de la uart del lector (niveles TTL 3.3V).
- GND. Retorno de alimentación (Masa/Ground).

***Nota**: Cualquier cableado incorrecto del lector puede dar como resultado en un daño del mismo y estos no están cubiertos por la garantía.*

4. Comunicación con el lector

Para comunicarse con el lector se utiliza la UART descrita en el apartado anterior. La configuración de esta es:

- 19200bps
- 8 bits de datos
- Sin paridad
- 1 bit de stop

La comunicación en sentido HOST a Lector se realiza mediante sencillos comandos AT (configuración), pero una vez configurado el

lector queda en modo transparente enviando por la uart lo que reciba por BLE (siempre que sean caracteres ascii imprimibles).

Cuando el lector se inicia envía por la uart la palabra ***"READY"***, para indicar al host que está preparado para su funcionamiento.

4.1. Conjunto de comandos

4.1.1. Sintaxis

Cada comando comienza con el prefijo AT y termina con alguno de los siguientes finales de línea <LF>, <CR> o <CR><LF>.

Las respuestas tiene el siguiente formato:
<Respuesta><CR><LF>

4.1.2. Lista de comandos

Comando	Descripción
AT	Solicitar atención
ATI	Información general del lector
AT+BLECFG	Configuración del lector
AT+AUTH	Establecimiento del resultado de autenticación

Tabla I. Lista de comandos AT

4.2. Descripción detallada de cada comando

4.2.1. AT Solicitar atención

AT Solicitar atención	
Ejecución del comando **AT**	Respuesta **OK**

Tabla II. Comando AT

4.2.2. ATI Mostrar información del lector

ATI Mostrar información del lector	
Ejecución del comando **ATI**	Respuesta Información del lector **Manufacturer: Odin Solutions** **Product: BLE Reader** **Firmware Version: <version>** **Serial Number: <serial>**
	Parámetros **<version>** Versión del firmware **<serial>** Número de serie del lector

Tabla III. Comando ATI

4.2.3. AT+BLECFG Configurar lector

AT+BLECFG Configurar lector	
Comando de ayuda **AT+BLECFG=?**	Respuesta Obtiene información de uso **+BLECFG:<size>,<rssi>**

	Parámetros **<size>** Tamaño máximo en caracteres que puede tener el token (1 a 512). **<rssi>** Nivel de rssi mínimo permitido de un cliente, para filtrar las conexiones entrantes, dependiente del alcance deseado (-127 a 0).
Comando de lectura **AT+BLECFG?**	Respuesta **+BLECFG:<size>,<rssi>**
	Parámetros Ver comando ayuda
Comando escritura **AT+BLECFG=<size>,<rssi>**	Respuesta Establece los parámetros de configuración **OK** Si hay algún error la respuesta es **ERROR**
	Parámetros Ver comando de ayuda

Tabla IV. Comando AT+BLECFG

4.2.4. AT+AUTH Establecer resultado operación autenticación

AT+AUTH Establecer resultado autenticación	
Comando de ayuda **AT+AUTH=?**	Respuesta Obtiene información de uso **+AUTH:<status>**
	Parámetros **<status>** Resultado de la operación de autenticación. Valor dependiente del equipo host, pero puede ser un valor entre 0 y 255.
Comando de escritura **AT+AUTH=<status>**	Respuesta Establece el resultado de la operación de autenticación **OK** Si hay algún error la respuesta es **ERROR**
	Parámetros Ver comando de ayuda

Tabla V. Comando AT+AUTH

4.3. Token

Un token no es más que una secuencia de caracteres ascii imprimibles. En este caso el lector puede manejar longitudes de tokens de 1 hasta 512 caracteres, que es el máximo permitido por BLE. El formato de este token puede ser definido por el usuario o aplicación host que maneja el lector, ya que este no realiza ningún tipo de procesamiento sobre él, salvo comprobar la longitud máxima de este

(que puede ser establecida por el host) y que sea una secuencia ascii válida (caracteres imprimibles).

Una vez que el lector reciba un token valido este será enviado inmediatamente por la uart al host para su procesamiento. El envío hacia el host se realiza en crudo, añadiendo solamente un fin del línea "\n", para que sea más fácil identificar el final del mismo.

En el caso concreto que nos ocupa de aplicación para IPexRFID, se utiliza un formato de token especifico que ha sido desarrollado para esta aplicación.

4.3.1. Formato token

El token para IPexRFID está compuesto de dos secciones, una que llamaremos online y otra offline. La sección online no es más que un identificador de usuario generado aleatoriamente que sirve para autenticar a este contra el servicio de validación. La sección offline, contiene información relativa al usuario para poderlo autenticar si el servicio de validación no se encuentra disponible en ese momento. Esta sección se encuentra a su vez cifrada para mantener la confidencialidad de los datos, así como impedir que pueda utilizarse para realizar accesos no autorizados.

<table>
<tr><th colspan="5">TOKEN</th></tr>
<tr><td>Sección online</td><td>"-"</td><td colspan="3">Sección offline</td></tr>
<tr><td rowspan="3">UUID Online

Longitud variable entre 1 y 99 caracteres</td><td colspan="4">UUID Offline</td></tr>
<tr><td>Vector</td><td colspan="3">Datos Cifrados</td></tr>
<tr><td>IV
12 Bytes.
Aleatorio</td><td>ID Usuario
10 Bytes</td><td>";"</td><td>Fecha Caducidad
10 Bytes</td></tr>
<tr><td>Ascii</td><td colspan="4">Ascii Hex</td></tr>
</table>

Tabla VI. Formato token

La sección online de la sección offline se encuentran separadas por el carácter **"-"** (guión medio), para que sea fácil de identificar donde empieza y acaba cada una. Como la sección online es una cadena de caracteres que puede ser "cualquier cosa", pueden aparecer por tanto más guiones en ella. Por tanto, para saber dónde acaba la sección online siempre se buscará el último guión más a la derecha del token.

4.3.1.1. UUID Online

Se trata de una cadena de caracteres de longitud variable entre 1 y 99 caracteres, que sirve para validar al usuario contra el servicio de validación y cuyo tiempo de validez es muy corto (definido por la aplicación de usuario). Este UUID puede estar formado por cualquier carácter imprimible desde el 0x20 al 0x7E, cualquier otro carácter fuera de ese rango hará que se descarte el token.

4.3.1.2. UUID Offline

Consta de una cadena de caracteres ascii hex (ascii hexadecimal), es decir, solo pueden aparecer caracteres ascii correspondientes del **"0"** al **"9"** y de la **"a"** a la **"f"** (también en mayúsculas). Esta cadena proviene del resultado de concatenar el vector, los datos cifrados, y a su vez convertirlos a ascii hex.

- **Vector o IV**. Se trata del vector de inicialización utilizado en el cifrado del campo datos y que es necesario para que el host pueda descifrar estos.
- **Datos**. Se trata de los datos identificativos del usuario y la fecha de caducidad de los mismos. Estos una vez descifrado contienen:
    - **ID usuario.** Identificador de usuario, que normalmente es un DNI o un identificador MiFare, pero puede ser cualquier cadena ascii de longitud exactamente 10 caracteres. Si es MiFare seguirá el formato

descrito en el manual del IPexRFID añadiendo dos "Z" al principio para completar los 10 caracteres y si es un DNI se añadirán "0" para completarlos. Por ejemplo: "ZZ1234ABCD" ó "0048555777".

- **";"**. Separador entre el ID usuario y la fecha de caducidad.
- **Fecha caducidad**. Se trata de una marca de tiempo en segundos desde 01/01/1970 00:00:00, con la fecha de expiración del identificador del token. De forma que host utilizará dicha marca para saber si el token es válido antes de verificar el UUID en la BBDD. **Esta marca de tiempo debe ser coherente con la hora del controlador, es decir, si éste trabaja con hora local esta marca deberá estar también en segundos locales y si trabaja en UTC, deberá estar en segundos UTC.**

Dependiendo de la librería que se utilice para el descifrado de los datos, es posible que sea necesario indicar donde se encuentra el TAG de validación de integridad de los mismos. En este caso se encuentra ubicado al final de los datos cifrados y se corresponde con los últimos 32 caracteres (16 bytes) de estos. Por tanto, si es necesario deberá separarse para proporcionárselos a la rutina de descifrado.

4.3.2. Ejemplos de tokens

A continuación se muestran un par de ejemplos de tokens válidos, siguiendo el formato expuesto en el apartado anterior. Se ha separado por colores las distintas partes que forman el token, azul UUID Online, verde Vector, negro Datos cifrados y rojo TAG validación datos.

```
7e6f500-a6b7-4337-94fc-a48803ddcc1a-
5a69893ff9e9ff48e458db54e02585d630da352542ad5b82ba4260c2c7065d69123ae1
c6b6506dcdc051206a463c35becd
```

El resultado de descifrar el token anterior da como resultado la cadena (sin comillas): "0012345678;1652265068"

```
fdcdb218-2936-4b68-b290-0b8c81b524fe-
a43489d3d8c237cc6bc82d78fc0ddbc0885baba0cbd033814fbcd5bcc91cb038c4dab1
de33ff4ec04a5de17c128436c253
```

Descifrando obtenemos la siguiente cadena (sin comillas): "ZZ56422702;1656630000"

4.3.3. Descifrado token

En los apartados anteriores se vio que la parte de datos de la sección Offline se cifra para mayor seguridad, así como garantizar la integridad y validez de los mismos. Para ello y en el caso concreto del IPexRFID se utiliza criptografía simétrica, por limitación de recursos que tiene dicho dispositivo. Como algoritmo de cifrado se usa **AES GCM con claves de 128 bits y vector de inicialización de 96 bits.**

En cualquier caso, esto no depende del lector BLE sino de host, por lo que se podría usar otro tipo de algoritmos, pero el descrito es el utilizado en los terminales IPexRFID hasta el momento de desarrollo de este documento (ver firmware 1.32).

4.4. Servicios del lector

El funcionamiento del lector BLE se basa en el funcionamiento estándar de un servidor GATT (Generic Attributes) de Bluetooth. En el cual se crea un servicio que tendrá 2 características sobre las que la aplicación móvil podrá interactuar para escribir el token o leer el resultado de la operación de autenticación.

- **Servicio**. No es nada más que una agrupación de datos, los cuales a su vez se dividen en características. Sirve para organizar los datos de forma jerárquica. Cada servicio tiene un UUID único que puede ser de 16 bits si está estandarizado por BLE o 128 bits si es un servicio personalizado.

- **Característica**. Son los datos que deseamos proveer en nuestro servicio. Igual que el servicio dispone de un UUID único que identifica la característica que ofrece el servicio. Cada característica a su vez tiene asociado un valor, que será la información que el cliente puede leer o escribir en ella. Para ello, a la hora de crearlas habrá que asignarle las propiedades correspondientes que pueden ser lectura, escritura y notificación o combinación de estas.

En el caso concreto del lector se proporciona los siguientes UUID para identificar los servicios y características que se detallarán más delante.

Servicio autenticación: **"4f509fad-7186-47f7-bd52-f0eeb5cc551c"**
Característica escritura: **"6c83b1b5-b41d-4c16-b47f-1387fa73eb3d"**
Característica estado autenticación: **"635a11b5-50e6-4815-9c6d-2c803108f82d"**

4.4.1. Servicio de autenticación

Es el servicio que se ha montado en el lector para poderlo identificar cuando un cliente intenta conectarse. El UUID de dicho servicio se envía en los anuncios del dispositivo para que sea más fácil la identificación del mismo.

4.4.2. Característica de escritura

En dicha característica, que además solamente tiene **solamente propiedad de escritura**, un cliente podrá escribir el token que desea utilizar para autenticarse, siguiendo el formato descrito anteriormente.

El formato soportado por esta característica es de array de bytes o cadena UTF-8, con una longitud máxima de 512 bytes.

Si transcurren 10 segundos sin escribir en la característica, el cliente conectado es expulsado (desconectado) para evitar que monopolice el lector.

4.4.3. Característica de estado de autenticación

Al contrario que la anterior, esta característica **solamente tiene la propiedad de lectura y notificación**, por lo que cualquier cambio que se produzca en esta le será notificado al cliente que esté conectado en ese momento y si este previamente se ha registrado para ser notificado. Pudiendo entonces leer para verificar la respuesta del host a la petición de autenticación. Si el cliente no leyese la característica será expulsado (desconectado) transcurridos 10 segundos, para evitar que monopolice el lector.

El formato soportado por esta característica es de 1 byte que puede mostrar uno de los siguientes valores:

Resultados del proceso de autenticación	
1	Permiso concedido
2	Permiso denegado
3	Token incorrecto (longitud, carácter ilegal, error descifrado, etc)
4	Token caducado
5	Timeout (no hubo respuesta del controlador)
6	Token caducado y anteriormente utilizado
7	Token ya utilizado
8	Token caducado con usuario en BBDD local
9	Token caducado sin usuario en BBDD local
10	Operación Incorrecta
11	Plazas parking reservadas
12	Parking lleno

Tabla VII. Resultados de autenticación

Los valores anteriores son los básicos soportados actualmente (ver firmware 1.32), pero en un futuro se podrían ir añadiendo si fuese necesario, por lo que la lista queda abierta a incluir nuevos códigos según necesidades. Los códigos 10, 11 y 12 no se utilizan en el ámbito de uso del IPexRFID, ya que están reservados para las aplicaciones de Parking.


OdinS