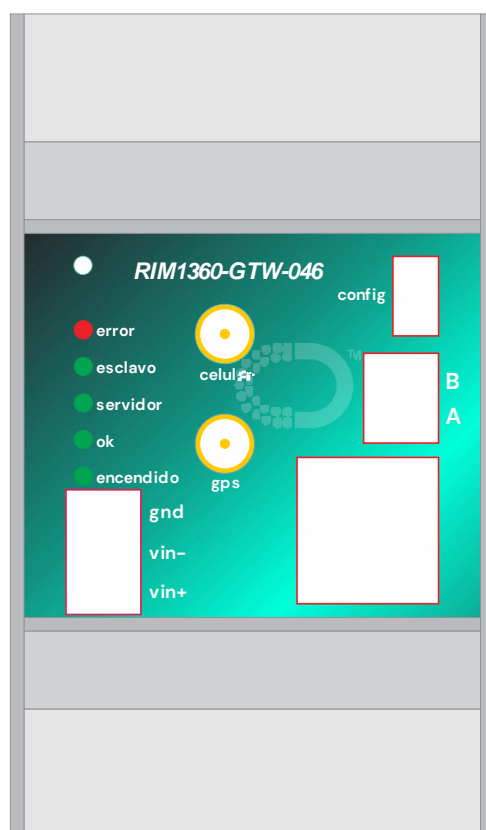


Manual de servicio

RIM1360-XXX

Configuración via JSON



Contenido

01 VERSIONES	2
02 CONTACTO	3
OFICINAS TESACOM	3
FABRICANTE	3
03 INTRODUCCIÓN	4
04 FORMATO DEL JSON	5
FUNCTION CODE	5
IDENTIFICATION	5
VERSION	6
Version 1	6

01

Versiones

<i>Fecha</i>	<i>Modificaciones</i>	<i>Versión</i>
11/08/2022	Implementación primera versión	1.0

02

Contacto

Twin Dimension[®] es una marca registrada por el Grupo Tesacom. El grupo Tesacom posee oficinas en Argentina, Perú, Paraguay, Chile

Oficinas Tesacom

- Perú: Calle Simón Bolívar Nro. 472 Dpto. 405, Miraflores, Lima.
- Argentina: MAZA 2140-CABA -BS.AS, Argentina.
- Paraguay: República de Siria 407, Asunción.
- Chile: Av. del Valle Sur 576, Oficina 405, Huechuraba, Santiago de Chile (Región Metropolitana).

Contacto Telefónico:



Llámenos

 0810-345-6728

 800-801456

 (+59521) 214-444

 (+511) 421-5534

Página web:

<http://www.tesacom.net/>

Soporte clientes vía E-mail:

clientes@tesacom.net

info@tesacom.net

Fabricante

Tesam Argentina S.A.

**Maza 2140 Ciudad Autónoma de Buenos Aires
Argentina**

03

Introducción

El presente documento establece los alcances del módulo de debug y configuración encapsulada en Json, estos mensajes pueden enviarse por puerto serie, vinculo bluetooth, socket TCP o encapsulado en TSD.

O4

Formato del JSON

Cada Json que se envíe al y desde el equipo debe contener toda la información necesaria para procesar la consulta, la información no puede estar encadenada de información de Jsons anteriores.

Los paquetes JSON mantendrán el siguiente formato y utilizarán los siguientes campos

```
{
  "func": "Function Code",
  "id": "identification",
  "ver": "Version de estructura Json",
  "varA": "Valor de la variable A",
  "varB": "Valor de la variable B",
  "varZ": "Valor de la variable Z"
}
```

Function Code

El primer parámetro a enviar en el Json es el código de función que indica la funcionalidad del paquete.

- read : paquete de lectura de configuración
- write : paquete de escritura de configuración
- error : error en el paquete recibido
- ok : confirmación de paquete recibido
- debug : paquete de debug
- resp : paquete de respuesta a una lectura

Identification

Este campo es la identificación del dispositivo, este id es un código que combina el tipo de equipo y el código único de equipo

- 4 bytes código de tipo de equipo
- 6 bytes número de serie del equipo

Version

Este campo indica la versión de Json utilizada para enviar los campos de datos

Version 1

Lectura

En los paquetes de lectura, las consultas se enviarán con el function code ***“read”*** y teniendo en cuenta las siguientes reglas

- Si el elemento es único se enviará como un par tag:índice donde el índice debe ser 0, ejemplo :

```
{
  "func": "read",
  "id": "0001000001",
  "ver": "1",
  "tsdId": "0"
}
```

- Si el elemento es un array se enviará como un array, el índice representa el índice del array que se consulta, puede ser único o varios, ejemplo

```
{
  "func": "read",
  "id": "0001000001",
  "ver": "1",
  "simAlp": [{"index": "0"}, {"index": "1"}]
}
```

- Si se desean todos los elementos del array el índice debe ser -1,

```
{
  "func": "read",
  "id": "0001000001",
  "ver": "1",
  "simAlp": [{"index": "-1"}]
}
```

Escritura

En los paquetes de escritura los datos a escribir se enviarán con el function code **“write”** y teniendo en cuenta las siguientes reglas

- Si el elemento es único se envía el par tag:valor

```
{
  "func": "write",
  "id": "0001000001",
  "ver": "1",
  "tsdID": "2195"
}
```

- Si el elemento forma parte de un array se debe enviar un JSONArray con un solo elemento indicando el índice

```
{
  "func": "write",
  "id": "0001000001",
  "ver": "1",
  "simAlp": [{"index": "0", "valor": " 190.2.12.148"}]
}
```

- Se pueden enviar más de un elemento en forma simultánea como un JSONArray

```
{
  "func": "write",
  "id": "0001000001",
  "ver": "1",
  "simAlp": [{"index": "0", "valor": " 190.2.12.148"} ,
  {"index": "1", "value": "190.2.12.148"},
  {"index": "2", "value": "190.2.12.148"}]
}
```


Respuesta

Los paquetes de respuesta a las lecturas enviarán con el function code **"resp"** y teniendo en cuenta las mismas reglas que los paquetes de escritura.

Confirmación

Los paquetes de confirmación de una escritura se enviarán con el function code **"ok"** y teniendo el siguiente formato.

```
{
  "func": "ok",
  "id": "0001000001",
  "ver": "1",
}
```

Error

En caso de recibir un paquete de escritura con errores se responderá con function code **"error"**

```
{
  "func": "error",
  "id": "0001000001",
  "ver": "1",
  "data": "descripción del error"
}
```

