

MASB

Protocolo de comunicación

MASB-COMM-S

#stm32

#c

#biomedical

#microcontroladores

#programacion

Albert Álvarez Carulla

hello@thealbert.dev

<https://thealbert.dev/>

7 de mayo de 2020



"MASB: Protocolo de comunicación MASB-COMM-S" © 2020
por Albert Álvarez Carulla se distribuye bajo
una Licencia Creative Commons
Atribución-NoComercial-SinDerivadas 4.0
Internacional

Índice

1	Protocolo de comunicación MASB-COMM-S	2
1.1	Introducción	2
1.2	Codificación de datos	2
1.3	<i>Endianness</i>	2
1.4	Paquete de comando	2
1.4.1	Comando	2
1.4.2	Parámetros	3
1.5	Paquete de datos	3
1.6	Detalles de los comandos	4
1.6.1	START_CV_MEAS	4
1.6.2	START_CA_MEAS	5

1. Protocolo de comunicación MASB-COMM-S

1.1. Introducción

A continuación, se detalla el funcionamiento del protocolo de comunicación MASB-COMM-S entre el *master* (ordenador) y el *slave* (dispositivo).

El protocolo de comunicación es la variante S (*Simplified*) del protocolo de comunicación MASB-COMM. En esta variante simplificada no se implementa:

- La gestión de errores (esto implica, por ejemplo, el no uso del campo *CRC16 (Cyclic Redundancy Check)*).
- Hoja de especificaciones electrónica incrustada.
- Juego de instrucciones estándar del protocolo (*write, read, ...*).
- Escritura de múltiples espacios de memoria con una única instrucción.

1.2. Codificación de datos

La (de)codificación de los mensajes o paquetes de bytes se realizará mediante el algoritmo de codificación *COBS (Consistent Overhead Byte Stuffing)*. El carácter utilizado como *term char*, que indicará el fin de un mensaje y que la codificación COBS eliminará del cuerpo del mismo, será el `0x00`.

1.3. Endianness

Los bytes de los valores enviados se envían en notación *little endian* (del byte menos significativo al más significativo).

1.4. Paquete de comando

El paquete de comando es el mensaje que enviará el *master* (ordenador) al *slave* (dispositivo). Los bytes que componen el paquete de comando son:

Comando	Parámetro 1	...	Parámetro N
---------	-------------	-----	-------------

1.4.1. Comando

Byte con el comando enviado al dispositivo.

Valor	Instrucción	Descripción
0x01	START_CV_MEAS	Pasa los parámetros de una voltimetría cíclica y la inicia.
0x02	START_CA_MEAS	Pasa los parámetros de una cronoamperometría y la inicia.
0x03	STOP_MEAS	Detiene la medición en curso.

1.4.2. Parámetros

Bytes auxiliares para el campo comando. Su contenido varía en función de dicho campo. Se puede enviar de 0 a 255 parámetros en un solo paquete de comando.

1.5. Paquete de datos

El paquete de datos es el paquete que el *slave* envía al *master* cada vez que realiza la medición de un punto durante una medición electroquímica. Los bytes que componen el paquete de datos son:

Parámetro 1	Parámetro 2	Parámetro 3	Parámetro 4
point_0	point_1	point_2	point_3

Parámetro 5	Parámetro 6	Parámetro 7	Parámetro 8
timeMs_0	timeMs_1	timeMs_2	timeMs_3

Parámetro 9	Parámetro 10	Parámetro 11	Parámetro 12
voltage_0	voltage_1	voltage_2	voltage_3

Parámetro 13	Parámetro 14	Parámetro 15	Parámetro 16
voltage_4	voltage_5	voltage_6	voltage_7

Parámetro 17	Parámetro 18	Parámetro 19	Parámetro 20
current_0	current_1	current_2	current_3

Parámetro 21	Parámetro 22	Parámetro 23	Parámetro 24
current_4	current_5	current_6	current_7

- `point uint32_t`: número identificativo del punto. Es secuencial y se inicia la cuenta en 1.
- `timeMs uint32_t`: milisegundos transcurridos desde el inicio de la medición.
- `voltage double`: tensión entre el electrodo de trabajo y el electrodo de referencia en voltios.
- `current double`: corriente a través de la celda electroquímica en amperios.

El índice indicado en la variable de cada parámetro de la tabla indica la significancia del byte dentro del valor, siendo 0 el byte menos significativo.

1.5.0.1. Ejemplo

- `point 1`
- `timeMs 100 ms`
- `voltage 0.23 V`
- `current 12.3 µA`

Paquete de datos (hex)

0100000064000000713D0AD7A370CD3F7050B12083CBE93E

Paquete codificado en COBS (hex)

020101010264010111713D0AD7A370CD3F7050B12083CBE93E00

1.6. Detalles de los comandos

1.6.1. START_CV_MEAS

1.6.1.1. Descripción El *master* envía al *slave* los parámetros de una voltametría cíclica y la inicia. Una vez iniciada la medición, el *slave* responde con un paquete de datos, con los datos correspondientes, cada vez que realiza la medición de un punto.

1.6.1.2. Parámetros

- **eBegin `double`**: potencial de celda en el que se inicia la voltametría cíclica. El potencial de la celda electroquímica se lleva del potencial de inicio al vértice de potencial 1. También indica el potencial en el que finalizará la voltametría cíclica.
- **eVertex1 `double`**: potencial al que se dirige la celda electroquímica des de el potencial de inicio. Una vez se llega a este potencial, se lleva el potencial de la celda electroquímica al vértice de potencial 2.
- **eVertex2 `double`**: potencial al que se dirige la celda electroquímica des de el vértice de potencial 1. Una vez se llega a este potencial, si no se han finalizado los ciclos programados, se lleva el potencial de la celda electroquímica al vértice de potencial 1. Si no quedan más ciclos, se lleva el potencial de la celda hasta el potencial de inicio y finaliza la medición.
- **cycles `uint8_t`**: número de ciclos de la voltametría cíclica.
- **scanRate `double`**: variación de la tensión de la celda electroquímica en el tiempo en voltios por segundo.
- **eStep `double`**: incremento/decremento de la tensión entre dos puntos consecutivos. Indicado en voltios.

1.6.1.3. Ejemplo

- eBegin 0.25 V
- eVertex1 0.5 V
- eVertex2 -0.5 V
- cycles 2
- scanRate 0.01 V/s
- eStep 0.005 V

Paquete de comando (hex)

```
01000000000000D03F000000000000E03F000000000000E0BF027B14AE47E17A843F7B14AE47E17A743F
```

Paquete codificado en COBS (hex)

```
0201010101010103D03F010101010103E03F010101010114E0BF027B14AE47E17A843F7B14AE47E17A743F00
```

1.6.2. START_CA_MEAS

1.6.2.1. Descripción El *master* envía al *slave* los parámetros de una cronoamperometría y la inicia. Una vez iniciada la medición, el *slave* responde con un paquete de datos, con los datos correspondientes, cada vez que realiza la medición de un punto.

1.6.2.2. Parámetros

- eDC **double**: potencial constante de la celda electroquímica durante la cronoamperometría en voltios.
- samplingPeriodMs **uint32_t**: tiempo en milisegundos entre muestra y muestra.
- measurementTime **uint32_t**: duración en segundos de la cronoamperometría.

1.6.2.3. Ejemplo

- eDC 0.3 V
- samplingPeriodMs 10 ms
- measurementTime 120 s

Paquete de comando (hex)

02333333333333D33F0A00000078000000

Paquete codificado en COBS (hex)

0B02333333333333D33F0A0101027801010100