



## Protocolo MF32 - Servidor



## Contenido

1. Introducción .....	2
2. Conexión.....	3
2.1. Conexión de coordenadas.....	3
2.2. Conexión de estado del taxímetro .....	4
2.3. Conexión Master/Slave .....	5
2.3.1. Peticiones del Host al MF32 .....	5
2.3.2. Respuestas del MF32 al Host .....	6
3. Comandos del protocolo.....	7
3.1. Lectura de totalizadores.....	7
3.2. Intervalos de refresco .....	7
3.3. Intervalos de refresco de KeepAlive.....	7
3.4. Actualización de firmware del MF32.....	8
3.5. Mostrar pantalla.....	8
3.6. Impresión de tique .....	9
3.7. Logout.....	9
3.8. Sonido.....	10
4. Eventos.....	11
4.1. Touch.....	11
4.2. Luminoso .....	11
4.3. Tarifa .....	11
4.4. Estado.....	12
4.5. Suplementos.....	12
4.6. Fin de carrera .....	12
4.7. Login .....	13
4.8. Login KO .....	13
4.9. Versión lista claves .....	13

# 1. Introducción

---

En este documento se va a explicar el protocolo de comunicación que sigue el módulo de flota MF32. Este permite la interacción con los taxímetros AM32 a través de un servidor cualquiera.

A lo largo de este documento se tiene que tener en cuenta que todas las fechas y horas especificadas en este están en formato UNIX time. Además, es importante notar que todas las líneas de datos mantendrán los campos separados por comas.

En los valores alfanuméricos, en caso de que el valor contenga el carácter de doble comilla o la contrabarra, irá escapado con un carácter “\”. Las “\” deben escaparse con “\” también.

Todos los importes monetarios se envían como un entero que representa céntimos de Euro. Por ejemplo: 3,20 € se envía como 320.

Utilizando los inicios y finales de cadenas se puede elegir entre enviar un único dato de información o enviarlos todos en un solo mensaje, teniendo en cuenta que siempre deben ir por parejas: EVENTO[x]DATO. Deberán comenzar con la palabra clave EV.

Los datos vacíos pueden indicarse como vacíos, pero también pueden eliminarse. El Fin de cadena nos permite enviar un número no prefijado de campos.

## 2. Conexión

---

Al encenderse el MF32, este se conecta en primer lugar con el taxímetro por bluetooth. Después, este avisa al servidor

```
2019-09-17 07:45:39,689 MainThread server INFO SERVER_NAME.Server.__init__()
2019-09-17 07:45:39,693 MainThread server INFO SERVER_NAME.Server.__init__() 1
2019-09-17 07:45:39,695 MainThread server INFO SERVER_NAME.Server.__init__() 2
2019-09-17 07:45:39,706 Server protocol INFO Starting listener...
2019-09-17 07:45:39,707 MainThread server INFO SERVER_NAME.Server.__init__() 3
2019-09-17 07:45:39,708 MainThread server INFO SERVER_NAME.Server.__init__() 4
```

Al conectarse el módulo de flota MF32 al servidor, el dispatcher de este tiene que contestar con información sobre tres puertos donde el MF32 pueda conectarse. Ej:

```
2019-09-17 07:45:39,709 MainThread server INFO Connecting
2019-09-17 07:45:39,709 MainThread server INFO Getting servers from
<domain_server:listener_port>
2019-09-17 07:45:49,743 MainThread server INFO <<<
"domain_server:port_1,domain_server:port_2,domain_server:port_3"
```

El nombre del host va separado del puerto por ":" y las diversas conexiones se separan entre ellas por ",".

Cuando la conexión ha quedado establecida, se denota de la siguiente forma:

```
2019-09-17 07:45:49,929 MainThread exa.mf32.ser INFO Server started
```

Los tres datos aportados por el dispatcher se usarán, en orden, para:

- Recepción de coordenadas
- Recepción de estado del taxímetro
- Comunicación Master (Host) / Slave (MF32) y transmisión de eventos especiales.

### 2.1. Conexión de coordenadas

El MF32 envía líneas cada cierto tiempo hacia el host. Este tiempo es programable. El MF32 no espera ningún tipo de respuesta del host al realizar esta tarea.

El formato de las líneas es el siguiente:

BOX\_ID,TIMESTAMP,LAT,LONG,VELOCIDAD,RUMBO,PRECISIÓN,ODÓMETRO

donde:

- BOX\_ID: Es el identificador único de la caja. Formato aún por determinar.
- TIMESTAMP: Momento en que se ha generado la información.
- LAT: Latitud en número decimal, positivo para N y negativo para S.
- LONG: Longitud en número decimal, positivo para E y negativo para W
- VELOCIDAD: Decim<sup>TM</sup>al en km/h
- RUMBO: En grados decimales
- PRECISIÓN: Estimación de la precisión (margen de error) en metros.
- ODÓMETRO: Entero que identifica la distancia recorrida en metros

Ej:

1519135548,41.490047,2.1397612000000663,0,0,200,1000

En caso de localización no se haya podido calcular, los campos estarán vacíos.

Ej:

3060637632,1568699167,0,0,0,0,0

## 2.2. Conexión de estado del taxímetro

A través de esta conexión el taxímetro manda una línea con el estado actual del taxímetro.

Esta línea presenta la siguiente forma:

BOX\_ID,TIMESTAMP,ESTADO,TARIFA,IMP\_C,IMP\_S,IMP\_O

donde:

- BOX\_ID Identificador único de la caja. Formato aún por determinar.
- TIMESTAMP: Momento en que se ha generado la información
- ESTADO: Estado del taxímetro, puede ser uno de:
  - L: Libre
  - O: Ocupado
  - P: A Pagar
  - S: Sin Servicio, el taxímetro no está en servicio
- TARIFA: Tarifa actual
- IMP\_C: Importe debido a la carrera
- IMP\_S: Importe debido a los suplementos
- IMP\_O: Importe debido a otros

Ej. 1

```
1519135550,1562141889,"L",0,0,0,0
```

Ej. 2

```
1519135550,1562141889,"O",1,5,0,0
```

Ej. 3

```
1519135550,1562141889,"S",0,0,0,0
```

## 2.3. Conexión Master/Slave

En esta conexión se pasarán tres tipos de información:

- Peticiones del Host al MF32.
- Respuestas del MF32 al Host
- Eventos especiales

Las comunicaciones se darán por líneas terminadas en CRLF con el siguiente patrón:

```
[*]campo1[x]campo2[x]etc...[*]+CRLF
```

Los valores irán separados por los tres caracteres: [x]. La cadena de información tendrá estos 3 caracteres como indicador de inicio y de final: [\*]

Hay dos tipos de comunicación, las peticiones o comandos y las respuestas

### 2.3.1. Peticiones del Host al MF32

Los comandos del Host tienen el siguiente formato:

```
[*]NONCE-S[x]ACCION_SOLICITADA[x]ARGUMENTOS[*]
```

Donde:

- NONCE-S: Identificador único distinto de cero

Ej.

```
[*]42[x]UPDATEFIRMWARE[x]http://pulse.eoden.es:3456/fw-update/482/[*]
```

```
[*]43[x]ECHO[x]hola[*]
```

```
[*]44[x]SHOWSCREEN[x]12345LINE 1111111111111111LINE 2222222222222222LINE  
3333333333333333LINE 4444444444444444LINE 5555555555555555LINE  
6666666666666666LINE 7777777777777777[*]
```

Nota: El icono que no se quiera mostrar se pone con un espacio en blanco.

## 2.3.2. Respuestas del MF32 al Host

Las respuestas del MF32 al Host tienen el siguiente formato:

```
[*]BOX_ID[x]TIMESTAMP[x]NONCE-C[x]DATOS_DE_RESPUESTA[*]
```

Donde:

- NONCE-C: Identificador único de la respuesta o evento
- BOX\\_ID: Número de serie único del MF32

Ej. (respecto al apartado anterior)

```
[*]123456[x]1568700105.924[x]42[x]ERROR[x]Connection failed[*]
```

```
[*]123456[x]1568700105.924[x]43[x]hola[*]
```

```
[*]123456[x]1568700105.924[x]44[x]OK[*]
```

A cada respuesta que el MF32 envíe al servidor, este esperará recibir un mensaje de confirmación de la recepción del comando del tipo:

```
[*]NONCE-C[x] ACK [*]
```

Cuando el MF32 no reciba la confirmación en un tiempo razonable, volverá a mandar la respuesta. Esto significa que es posible que, ocasionalmente, el servidor reciba dos o más veces la misma respuesta. En este caso, debe, simplemente, confirmar de nuevo la recepción e ignorarla.

Además, el MF32 mandará líneas de “Keep Alive” al servidor con el siguiente formato:

```
‘#’ + CRLF
```

El servidor ignorará estas líneas.

# 3. Comandos del protocolo

---

## 3.1. Lectura de totalizadores

- COMANDO: `TOTALIZERS`
- ARGUMENTOS: Ninguno
- CAMPOS RESPUESTA:
  - SLDT\_MT: Totalizador de distancia, en metros
  - SLDT\_MO: Totalizador de distancia ocupado, en metros
  - SLDT\_NS: Número de servicios realizados
  - SLDT\_IS: Totalizador de importe de suplementos, en céntimos de Euro
  - SLDT\_IT: Totalizador de importe, en céntimos de Euro

Ej.

```
[*]EV[x]TOTALIZERS[*]
```

## 3.2. Intervalos de refresco

Este comando indica al MF32 cada cuanto tiene que informar al servidor de su estado.

- COMANDO: `REFRESHTIME`
- ARGUMENTOS: Número entero de segundos entre refrescos

Ej.

```
[*]EV[x]REFRESHTIME[x]30[*]
```

## 3.3. Intervalos de refresco de KeepAlive

Este comando indica al MF32 cada cuanto tiene que enviar un evento de KeepAlive.

- COMANDO: `KEEPALIVE`
- ARGUMENTOS: Número entero de segundos entre refrescos

Ej.

```
[*]EV[x]KEEPALIVE[x]35[*]
```



## 3.4. Actualización de firmware del MF32

- COMANDO: `UPDATEFIRMWARE`
- ARGUMENTOS: URL\_FIRMWARE

Ej.

```
[*]EV[x]UPDATEFIRMWARE[x]URL_FIRMWARE[*]
```

## 3.5. Mostrar pantalla

- COMANDO: `SHOWSCREEN`
- ARGUMENTOS: "MY\_SCREEN\_STRING"

Ej.

```
[*]EV[x]SHOWSCREEN[x]MY_SCREEN_STRING[*]
```

El formato del volcado de la pantalla es una string con  $5 + 7 * 19$  caracteres.

Los 6 primeros son los 6 iconos que pueden mostrarse, en este orden:

- izquierda arriba
- izquierda centro
- izquierda abajo
- derecha arriba
- derecha centro
- derecha abajo

El primero de estos 6 dígitos indicará si la pantalla es modal o no. Si este primer dígito es 0, la pantalla no será modal, el taxímetro pondrá la tipografía de la Casa y se podrá salir de esa pantalla pulsando este icono.

Los iconos mostrados corresponden a la fuente "T32 pic.ttf" que está en la carpeta compartida de Dropbox. De todas maneras, no todos los iconos son válidos. Antes de usar un icono determinado conviene probarlo.

Los otros  $7 * 19$  caracteres corresponden a las 7 líneas de la pantalla de arriba hacia abajo. Las líneas siempre deben transmitirse completas, si hay menos de 19 caracteres deben rellenarse con espacios en blanco.

Cualquier volcado de pantalla con más o menos de  $5 + 7 * 19$  caracteres será considerado como un error.

La respuesta a esta solicitud no llegará hasta que la pantalla sea mostrada. Puede pasar un tiempo indeterminado hasta que ello suceda, ej. Minutos si está haciendo una carrera u horas si no está en servicio.

En caso de solicitar mostrar una pantalla antes de haber mostrado la anterior la petición de la anterior quedará anulada, mostrándose solo la última.

No todas las letras son válidas, no existen letras acentuadas ni otras especiales: en particular, la Ñ y la Ç. En caso de recibir una pantalla con caracteres no disponibles se rechazará la pantalla entera con una respuesta de ERROR.

Ej.

```
[*]44[x]SHOWSCREEN[x]12345LINE 1111111111111111LINE 2222222222222222LINE  
3333333333333333LINE 4444444444444444LINE 5555555555555555LINE  
6666666666666666LINE 7777777777777777[*]
```

## 3.6. Impresión de tique

Un tique se puede imprimir de dos maneras, a través de una URL o enviando el propio tique en formato string.

- COMANDO: `PRINTURL`
- ARGUMENTOS: "URL\_TIQUET"

Ej.

```
[*]EV[x]PRINTURL[x]URL_TIQUET[*]
```

- COMANDO: `PRINT`
- ARGUMENTOS: "TIQUET"

Ej.

```
[*]EV[x]PRINT[x]TIQUET[*]
```

## 3.7. Logout

La recepción de este comando provocará que se solicite de manera automática el LOGIN del conductor.

- COMANDO: `LOGOUT`
- ARGUMENTOS: ""

Ej.

```
[*]67[x]LOGOUT[*]
```

## 3.8. Sonido

La recepción de este comando provocará que se reproduzca el fichero con nombre indicado en NUM por la salida estándar. Estos ficheros están situados en /var/cache/mf32/.

- COMANDO: 'SONIDO'
- ARGUMENTOS: 'NUM'

Ej.

```
[*]EV[x]SONIDO[x]1[*]
```

# 4. Eventos

---

A continuación se explican los eventos enviados por el MF32 al servidor. Estos no esperan ninguna respuesta.

## 4.1. Touch

Campos del evento: 'TOUCH'

Ej.

```
[*]EV[x]TOUCH[x]a[*]
```

Donde el parámetro indica la parte de la pantalla pulsada. Consultar el manual XXX-##-ProtocoloDeComunicaciones para más información.

## 4.2. Luminoso

Campos del evento: 'LUMINOSO'

Ej.

```
[*]EV[x]LUMINOSO[x]001[*]
```

Evento que se genera cuando el estado del luminoso cambia. Contiene un parámetro con 3 dígitos numéricos. El significado de estos dígitos es, por orden:

- Booleano. Se refiere a la luz de libre. 0 apagada y 1 encendida.
- Booleano. Se refiere a la palabra TAXI. 0 apagada y 1 encendida
- Natural. Indica la el número de tarifa presente en el luminoso.

## 4.3. Tarifa

Campos del evento: "TARIFA"

Evento que se genera al cambiar la tarifa seleccionada en el taxímetro. Indica el nombre interno de la tarifa y el que se muestra en el taxímetro de la siguiente forma 'interno:externo'.

Ej.

```
[*]EV[x]TARIFA[x]108:1[*]
```

## 4.4. Estado

Campos del evento: 'ESTADO'

Ej.

```
[*]42[x]ESTADO[x]L[*]
```

Contiene el nuevo estado del AM32. Los valores posibles del estado son:

- L: Libre
- O: Ocupado
- P: A Pagar
- S: Fuera de servicio

## 4.5. Suplementos

Campos del evento: 'IDSUPL NSUPL SUPL '

Donde:

- IDSUPL: Identificador interno del suplemento
- NSUPL: Cantidad de suplementos añadidos
- SUPL: Nombre mostrado del suplemento

Ej.

```
[*]EV[x]IDSUPL[x]1[x]NSUPL[x]108[x]SUPL[x]maleta[*]
```

## 4.6. Fin de carrera

Campos del evento: 'INICIO FIN DURACION DISTANCIA IMPORTE SUPL BASE PCT\_IVA IMP\_IVA TOTAL NUM\_SERV TARIFAS' \*

Donde:

- INICIO: Fecha/Hora de inicio de la carrera
- FIN: Fecha/Hora de fin de la carrera
- DURACION: FIN - INICIO
- DISTANCIA: Distancia recorrida
- IMPORTE: Importe de la carrera
- SUPL: Importe de suplementos
- OTROS: Importe otros conceptos
- BASE: Importe base imponible
- PCT\_IVA: Porcentaje de IVA
- IMP\_IVA: Importe IVA

- TOTAL: Importe total: carrera + suplementos + otros
- NUM\_SERV: Numero de servicio
- TARIFAS: Lista de tarifas utilizadas, separadas por , (coma)

Ej.

```
[*]2424726315[x]1536752705.422[x]34[x]EV[x]SUPL:AER/PUE[x]310[x]OTROS[x]0[x]
NUM_SERV[x]143[x]SUPL:PASAJ[x]310[x]SUPL[x]620[x]TOTAL[x]845[x]DISTANCIA[
x]0[x]FIN[x]1536752778[x]DURACION[x]28[x]INICIO[x]1536752750[x]IMPORTE[x]225[
x]PCT_IVA[x]10.0[x]TARIFAS[x]1,6,P[x]BASE[x]768[x]IMP_IVA[x]77[*]
```

*\*Consultar apéndice 5.1.Equivalencia con campos solicitados por SERVER\_NAME*

## 4.7. Login

Cuando el conductor hace un login correcto con la tabla de conductores que tiene el MF32, este envía un evento al servidor.

- COMANDO: 'LOGIN'
- ARGUMENTOS:'id\_conductor versión\_tabla\_conductores'

Ej.

```
[*]EV[x]LOGIN[x]id_conductor[x]version_tabla_conductores[*]
```

Al enviar este evento, se espera que el MF32 reciba una pantalla que indica que el conductor se ha logado correctamente.

## 4.8. Login KO

Cuando el conductor hace un login incorrecto respecto la tabla de conductores que tiene el MF32, se envía este evento.

- EVENTO: 'LOGIN\_KO'
- ARGUMENTOS:'id\_conductor versión\_tabla\_conductores'

Ej.

```
[*]3060436925[x]1561640258.814[x]9666[x]EV[x]LOGIN_KO[x]123456[x]VERSION[x]2
[*]
```

## 4.9. Versión lista claves

Cuando el MF32 recibe y guarda una nueva tabla de conductores, envía este evento.

- EVENTO: 'VERSIONLISTACLAVES'
- ARGUMENTOS: 'versión\_tabla\_conductores'

Ej.

[\*]3060436925[x]1561640258.814[x]9666[x]EV[x]VERSIONLISTACLAVES[x]2[\*]