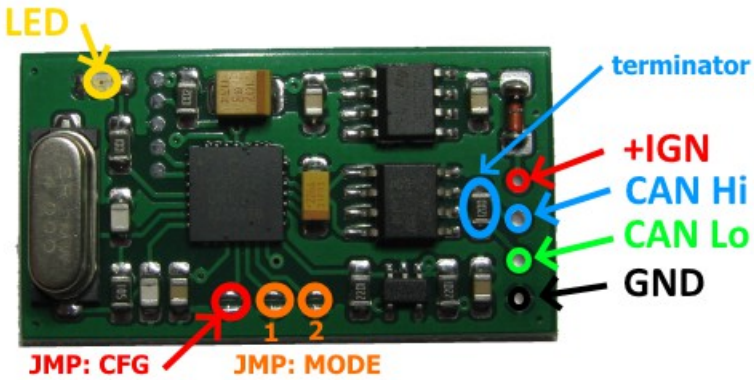


MB FBS3 emulator

for ECU, EGS, ISM/DSM authorization

CAN HS 500kB

Designed for MB FBS3 immobilizer system (based on hashes) to authorize ECU, EGS (VGS) or ISM/DSM unit. Basic principles are very similar to FBS3 IR KEY operation: must store initial hash, track number and distance into emulator. Compatible with later FBS3 systems (after 2006) where alternative frame ID's are used for authorization.



How to configure using CAN logger:

Solder joint **JMP:CFG** must be removed (open) to allow configuration mode. Must use any suitable CAN bus logger. Communication speed is 500 kb.

Abbreviations used:

-> transfer to emulator; <- response from emulator

step 1: must transfer **INITIAL HASH** - for example if INITIAL HASH is 14C2DAE130F81AC8:

```
-> 7F0 (8) 14 C2 DA E1 30 F8 1A C8
<- 7FF (8) 14 C2 DA E1 30 F8 1A C8
```

step 2: must transfer track number and distance -

example: track = **3**, distance = **2F1F7F** (full key track here, although it is possible to configure emulator with any valid distance value from 000001 up to 2F1F7F)

```
-> 7F1 (8) 03 2F 1F 7F 00 00 00 00      - 1st part of response must follow immediately:
<- 700 (4) 03 2F 1F 7F
```

If everything is okay and initial hash was received previously in current power-on cycle, emulator will go to prepare buffers now. Usually it takes about 20 seconds to complete task. When finished, emulator responds with calculated final hash for distance 2F1F7F (always for a full track length, no matter what distance was actually requested):

2nd part of response (calculated FINAL HASH: buffers already built, emulator is ready)

```
<- 7FF (8) D8 A5 57 F3 F8 FD 87 36
```

LED must go permanently ON, emulator is configured and ready for operation now. Must place solder joint to short **JMP:CFG** to enter normal operation mode.

Normal operational mode:

JMP:CFG is shorted, emulator is connected to CAN. Take attention on CAN terminator, possibly you may need to remove it from emulator board if CAN lines on car are already terminated. Set **JMP:MODE** (two solder joints) to select which control unit to authorize:

- **ECU** : **JMP:MODE 1** and **2** both open
- **EGS (VGS)** : **JMP:MODE 1** short, **JMP:MODE 2** open
- **ISM (DSM)** : **JMP:MODE 1** open, **JMP:MODE 2** short

Note that it is possible to authorize only one unit, although you can use two or more emulators to authorize entire system.

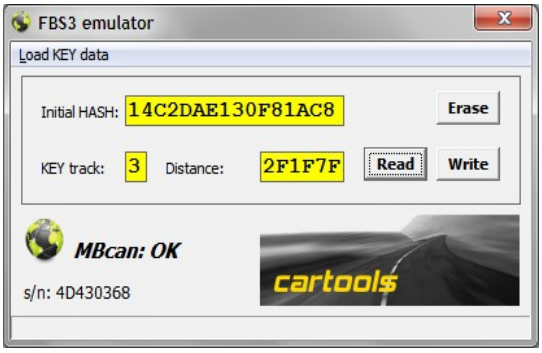
After power-up **LED** must go on, emulator is ready to communicate with ECU/EGS/ISM. If **LED** is blinking, emulator is not ready or there is an error state – it is recalculating buffers (busy), HASH is not valid, invalid data is stored into eeprom, remaining distance is very low etc.

You can check if control unit is authorized or not using diagnostic tool or MBcan.

How to configure using MBcan:

Attach **MBcan**, launch software, enter what's requested – INITIAL HASH, distance, track number. Then push button WRITE, wait until finished (up to 20 seconds). Note that **JMP:CFG** must be open while configuring emulator.

In addition it is possible to load all necessary data directly from generated KEY "dump" and read what is actually stored into emulator.



==========

Additional commands for enthusiasts (configuration mode only – **JMP:CFG** is open):

Erase ALL:

-> 7F1 (8) 55 00 00 00 00 00 00 00
Success is confirmed with message:
<- 7FF (6) 45 52 41 53 45 44 (ascii: ERASED)

Current HASH request (decrease life by 1):

-> 7F1 (8) AA 00 00 00 00 00 00 00
response, 2 frames:
<- 700 (4) KK DD DD DD, where **KK** = track number, **DDDDDD** = current distance
<- 7FF (8) 8xDATA , where **DATA** is current hash

Request data blocks (EEPROM):

-> 7F1 (8) FF xx 00 00 00 00 00 00, where **xx** = data block number, 00...FF
response:
<- 7xx (8) 8xDATA

==========

(c) RobinDAB 2015 / 2019 / 2024

cartools