

# Renault CAN/K-line ECU Tool V1.03

## User manual



Revision D

May 2012

## 1. Introduction

**Renault CAN/K-line ECU Tool** is an OBDII tool with various engine control unit internal memory modification functions, such as read/write EEPROM and FLASH, clearing immobilizer code. There is no need to open control unit, de-solder out memory IC or connect any wires directly to it. Clear Immobilizer Code function is fully automated – just press button and it is cleared. ECU is ready to accept new immobilizer code after previously programmed code is cleared – no future programming is required. It learns new code from operational immobilizer system or **Renault Immobilizer Emulator** (not included) by itself after first ignition on.

**NOTE: From supported in this version only SAGEM SAFIR and IAW 6R.30 engine control units can operate without immobilizer code stored.**

This is tool that makes your job easier and faster - it takes no more than 40 seconds to clear SIRIUS32 immobilizer data or about 2 minutes to write fully its AM29F200 FLASH memory. Besides, you can also check immobilizer status of any supported ECU - is immobilizer code programmed or not, is protection active or inactive.

## 2. PC requirements

**Renault CAN/K-line ECU Tool** connects to desktop or laptop PC using any free USB port. It is self-powered from USB and OBDII. Software runs on Windows 98, Windows XP, Windows Vista, Windows7. Software package includes USB drivers.

## 3. Supported engine and other control systems

Software version V1.03 supports:

Manufacturer	System	Read/Write FLASH	Read/Write EEPROM	Engine runs with immobilizer code cleared
BOSCH	EDC15C3	YES	YES	NO
SIEMENS	SID 301	YES	YES	NO
	SIRIUS 32	YES	YES	NO
	SIRIUS 34	YES	YES	NO
	EMS 3132	YES	YES	NO
SAGEM	SAFIR/SAFIR2	NO	ONLY READ	YES
MAGNETI MARELLI	IAW 6R	NO	ONLY READ	YES
VALEO	Espace-III BII	NO	YES	-
SIEMENS	UCH ('98-'02)	NO	YES	-
SAGEM	UCH ('01-'05)	NO	NO	-

\* - Correct immobilizer signal is necessary. New code to be stored from **Renault Immobilizer Emulator** (not supplied) or fully operational immobilizer system (valid key, etc.)

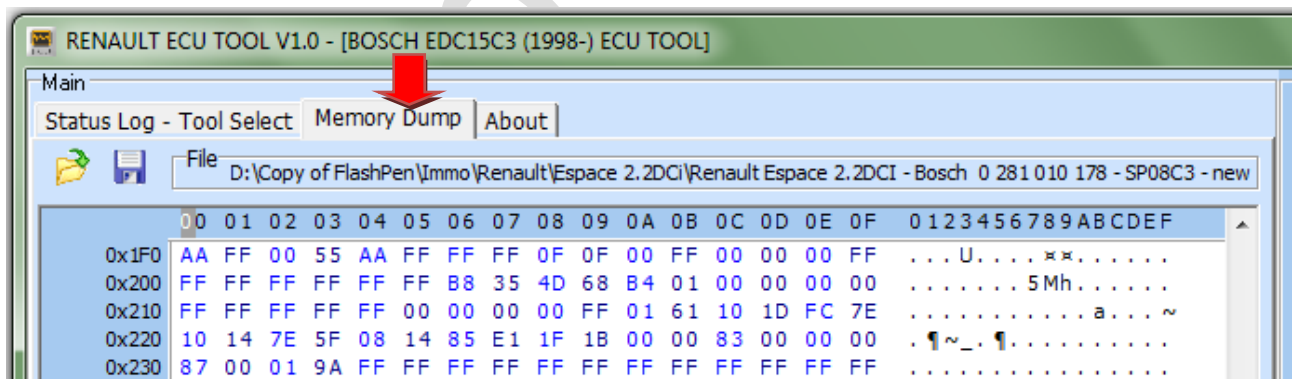
## 4. Software hints

Basicly, this is not a tool for TOTAL immobilizer disabling - it only clears previously memorised immobilizer code to give ability of using second-hand Engine Control Unit (ECU) in particular car or make a simple data cloning from one ECU to another. It is usefull, when immobilizer system is working properly, but original ECU has a failure and needs to be replaced. If red immobilizer indicator LED is blinking rapidly with ignition on (transponder key is not recognised), code clearing is not a way to make engine running (exceptions are **SAFIR/SAFIR2** and **IAW 6R**). Anyway, ECU needs to be learned with new immobilizer code and **Renault Immobilizer Emulator** (not included, sold on the internet by other sellers) can be a solution of this problem.

Therefore this tool has functional ability to write full FLASH memory by OBDII connector with binary memory dumps that had been previously read using ordinary FLASH programmer (like ELNEC, etc.) So you can somewhere find file for particular ECU with TOTALLY disabled immobilizer function and write it without desoldering of FLASH memory by OBDII. Always check for software and hardware version compatibility before programming. Do it on your own risk as we do not carry any responsibility for files, patched by somebody else!

**Open** binary file to write to FLASH or EEPROM memory from **Memory Dump** tab selector.

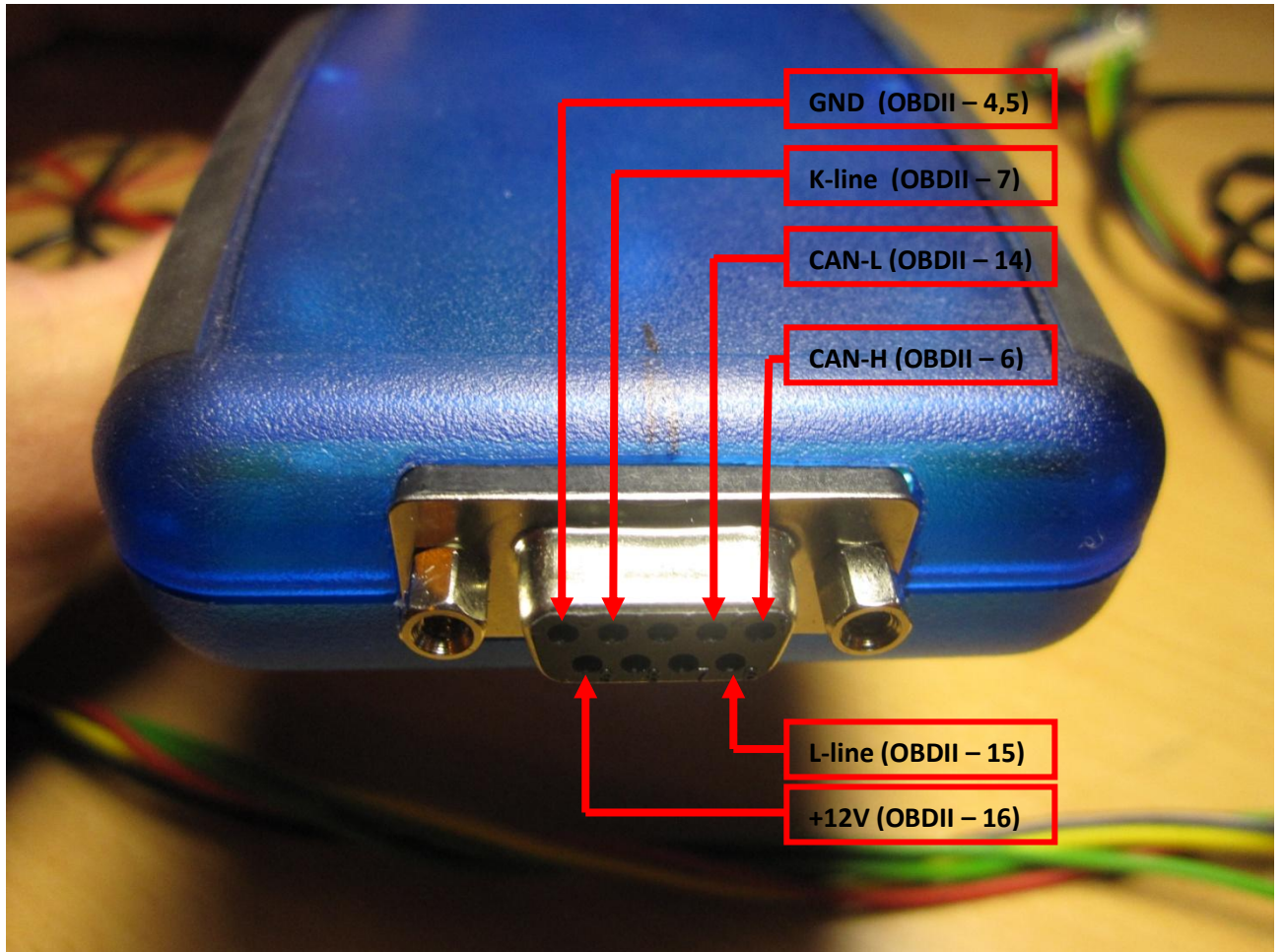
From the same tab you can also **save** FLASH or EEPROM data readout to file.



### Description of ECU statuses

- **Immo code programmed:**
  - YES** – immobilizer code is programmed. Used ECU.
  - NO** – immobilizer code is cleared. ECU is in autoprogramming mode for new immobilizer code to store.
- **Immobilizer status:**
  - ACTIVE** - ECU is locked by immobilizer. Starting impossible.
  - INACTIVE** - ECU is not locked. Engine starting is enabled.

## 5. Hardware



Immo

## **BOSCH EDC15C3**

### Hardware

- Microcontroller** – INFINEON SAK-C167CS-LM
- External FLASH memory** – AM29F400BT (Memory size – 512 kilobytes)
- External EEPROM memory** – 95P08 (Memory size – 1024 bytes)

### Basic functions

- **Read Identification**

Press this button to read EDC15C3 Software/Hardware version and status of immobilization.

### Advanced functions

- **Read EEPROM 95P08**

Press this button to read 95P08 EEPROM memory. Memory content can be modified and saved to disk with desired filename. Output file size – 1024 bytes.

- **Write EEPROM 95P08**

Open file, containing data to be programmed and press this button to write 95P08 EEPROM memory. Writes data from “Memory Dump” window to the 95P08 EEPROM. Content can be modified prior to writing procedure. Input file size – 1024 bytes.

- **Read FLASH AM29F400**

Press this button to read **full** AM29F400BT FLASH memory image. Memory content can be modified and saved to disk with desired filename. Output file size – 512 kilobytes.

- **Write FLASH AM29F400**

Open file, containing data to be programmed and press this button to **fully** rewrite AM29F400BT FLASH memory. Writes data from “Memory Dump” window to the AM29F400BT FLASH memory. Content can be modified prior to writing procedure. Input file size – 512 kilobytes.

- **Clear IMMO Code**

Press to clear immobilizer code stored in the 95P08 EEPROM memory. Software version automatic detection is implemented. ECU is ready to accept new code after clearing. In case of original immobilizer system malfunction, **Renault Immobilizer Emulator** (not included) can be used on EDC15C3 with non-CAN immobilizer system (1998-2000). **Engine will not start if no immobilizer code is stored at all.**

## **SIEMENS SID301**

### Hardware

- Microcontroller** – FREESCALE MPC561
- External FLASH memory** – AM29BDD160GB (Memory size – 2 Megabytes)
- External EEPROM memory** – 95320 (Memory size – 4 Kilobytes)

### Basic functions

- **Read Identification**

Press this button to read SID301 Software/Hardware versions, VIN number, status of immobilizer code programming.

### Advanced functions

- **EEPROM 95320 CRC Check**

Analyzes CRC of EEPROM blocks during **Read EEPROM 95320** and corrects them before **Write EEPROM 95320** if this Check Box is selected. V1.03 supports 4 known for us SID301 software versions. **May not work on some SW versions!** In that case, please, send EEPROM file to [support@immo-tools.it](mailto:support@immo-tools.it) and it will be included.

- **Read EEPROM 95320**

Press this button to read 95320 EEPROM memory. Memory content can be modified and saved to disk with desired filename. Output file size – 4 Kilobytes.

- **Write EEPROM 95320**

Open file, containing data to be programmed and press this button to write 95320 EEPROM memory. Writes data from “Memory Dump” window to the 95320 EEPROM. Content can be modified prior to writing procedure. Input file size – 4 Kilobytes.

- **Read FLASH AM29BDD160GB**

Press this button to read **full** AM29BDD160GB FLASH memory image. Memory content can be modified and saved to disk with desired filename. Output file size – 2 Megabytes.

- **Write FLASH AM29BDD160GB**

Open file, containing data to be programmed and press this button to **fully** rewrite AM29BDD160GB FLASH memory. Writes data from “Memory Dump”

window to the AM29BDD160GB FLASH memory. Content can be modified prior to writing procedure. Input file size – 2 Megabytes.

- **Clear IMPACT DETECTED**

Fuel injection is disabled and status **IMPACT DETECTED** is recorded after car suffers an accident. Press button **Clear IMPACT DETECTED** to re-enable fuel injection. This also can be done by clearing faults with CAN-CLIP, therefore immobilizer must be inactive. So this function is useful when you have SID301 ECU connected “on the bench” without a car.

- **Clear IMMO Code**

Press to clear immobilizer code. SID301 is ready to accept new code after clearing. No further learning with CLIP is necessary. **Engine will not start if no immobilizer code is stored at all.**

- **Read KM**

Press to read several mileages, stored inside the memory of SID301 ECU. Current version V1.03 does not have a function to change them, but you can do this manually: convert each mileage value to HEX (using Windows Calculator) and find these values in the EEPROM memory. Make a changes and write back EEPROM file with **EEPROM 95320 CRC Check** selected.

## SIEMENS SIRIUS 32 / SIRIUS 32N with Immo-by-Wire

### Hardware

<b>Microcontroller</b>	– INFINEON SAK-C167SR-LM or CR-LM
<b>External FLASH memory</b>	– AM29F200BB (Memory size – 256 kilobytes)
<b>EEPROM memory</b>	– In-FLASH emulated (Memory size – 16 kilobytes)

### Basic functions

- **Read Identification**

Press this button to read SIRIUS32 Software/Hardware version and status of immobilization.

### Advanced functions

- **Read EEPROM**

Press this button to read emulated EEPROM area (address range 4000-7FFF of FLASH memory). Memory content can be modified and saved to disk with desired filename. Output file size – 16 kilobytes.

- **Write EEPROM**

Press this button to write emulated EEPROM area (address range 4000-7FFF of FLASH memory). Programs EEPROM area with data from "Memory Dump" window. Content can be modified prior to writing procedure. Input file size – 16 kilobytes. **NOTE: NEVER program SIRIUS32 with EEPROM data from SIRIUS32N and vice versa! They have different memory block structure and check sum addresses! In the case when you need to clone FLASH memory of SIRIUS32 to hardware of SIRIUS32N (and vice versa), write FULL FLASH memory image using Write FLASH AM29F200!**

- **Read FLASH AM29F200**

Press this button to read full 29F200 FLASH memory image. Memory content can be modified and saved to disk with desired filename. Output file size – 256 kilobytes.

- **Write FLASH AM29F200**

Open file, containing data to be programmed and press this button to rewrite fully 29F200 FLASH memory. Writes data from "Memory Dump" window to the 29F200 FLASH memory. Content can be modified prior to writing procedure. Input file size – 256 kilobytes.

- **Clear IMMO Code**

Press to clear immobilizer code stored in the EEPROM memory. Software version automatic detection is implemented. ECU is ready to accept new code after clearing. **Renault Immobilizer Emulator** (not included) can be used in case of original immobilizer system malfunction. **Engine will not start if no immobilizer code is stored at all.**

## **SIEMENS SIRIUS 34**

### Hardware

<b>Microcontroller</b>	– INFINEON SAK-C167CS-LM
<b>External FLASH memory</b>	– AM29F400BB (Memory size – 512 kilobytes)
<b>EEPROM memory</b>	– In-FLASH emulated (Memory size – 16 kilobytes)

### Basic functions

- **Read Identification**

Press this button to read SIRIUS34 Software/Hardware version and status of immobilization.

### Advanced functions

- **Read EEPROM**

Press this button to read emulated EEPROM area (address range 4000-7FFF of FLASH memory). Memory content can be modified and saved to disk with desired filename. Output file size – 16 kilobytes.

- **Write EEPROM**

Open file, containing data to be programmed and press this button to write emulated EEPROM area (address range 4000-7FFF of FLASH memory). Programs EEPROM area with data from "Memory Dump" window. Content can be modified prior to writing procedure. Input file size – 16 kilobytes.

- **Read FLASH AM29F400**

Press this button to read **full** 29F400 FLASH memory image. Memory content can be modified and saved to disk with desired filename. Output file size – 512 kilobytes.

- **Write FLASH AM29F400**

Open file, containing data to be programmed and press this button to rewrite **fully** 29F400 FLASH memory. Writes data from "Memory Dump" window to the 29F400 FLASH memory. Content can be modified prior to writing procedure. Input file size – 512 kilobytes.

- **Clear IMMO Code**

Press to clear immobilizer code stored in the emulated EEPROM memory. Software version automatic detection is implemented. ECU is ready to accept new code by CAN-BUS after clearing. **Engine will not start if no immobilizer code is stored at all.**

## SIEMENS EMS3132 and SIRIUS32N with Immo-by-CAN

### Hardware

<b>Microcontroller</b>	–	INFINEON SAK-C167CS-LM
<b>External FLASH memory</b>	–	AM29F200BB (Memory size – 256 kilobytes)
<b>EEPROM memory</b>	–	In-FLASH emulated (Memory size – 16 kilobytes)

### Basic functions

- **Read Identification**

Press to read EMS3132 Software/Hardware version and status of immobilization.

### Advanced functions

- **Read EEPROM**

Press this button to read emulated EEPROM area (address range 4000-7FFF of FLASH memory). Memory content can be modified and saved to disk with desired filename. Output file size – 16 kilobytes.

- **Write EEPROM**

Press this button to write emulated EEPROM area (address range 4000-7FFF of FLASH memory). Programs EEPROM with data from "Memory Dump" window. Content can be modified prior to writing procedure. Input file size – 16 kilobytes.

- **Read FLASH AM29F200**

Press this button to read **full** 29F200 FLASH memory image. Memory content can be modified and saved to disk with desired filename. Output file size – 256 kilobytes.

- **Write FLASH AM29F200**

Open file, containing data to be programmed and press this button to rewrite **fully** 29F200 FLASH memory. Writes data from "Memory Dump" window to the 29F200 FLASH memory. Content can be modified prior to writing procedure. Input file size – 256 kilobytes.

- **Clear IMMO Code**

Press to clear immobilizer code stored in the EEPROM memory. Software version automatic detection is implemented. ECU is ready to accept new code after clearing. **Engine will not start if no immobilizer code is stored at all.**

## **SAGEM SAFIR/SAFIR2 , MAGNETI MARELLI SFR200**

### Hardware

<b>Microcontroller</b>	– TMS374
<b>External ROM memory</b>	– M27C512 (Memory size – 64 kilobytes)
<b>EEPROM memory</b>	– Internal MCU's EEPROM (Memory size – 256 bytes)

### Basic functions

- **Read Identification**

Press to read SAFIR, SAFIR2 or SFR200 Software/Hardware version and state of immobilization.

### Advanced functions

- **Read EEPROM**

Press this button to read internal microcontroller EEPROM. Memory content can be modified and saved to disk with desired filename. Output file size – 256 bytes.

- **Clear IMMO Code**

Press to clear immobilizer code stored in the EEPROM memory. ECU is ready to accept new code after clearing. Engine starts, even with code cleared – **NECESSARY to cut (disconnect) the immobilizer wire (pin 37 for SAFIR/SAFIR2 with 55 pin connector or pin 30 for SAFIR/SAFIR2/SFR200 with 35 pin connector).**

## MAGNETI MARELLI IAW6R.XX

### Hardware

- Microcontroller** – TMS370
- EEPROM memory** – Internal MCU's EEPROM (Memory size – 256 bytes)

### Basic functions

- **Read Identification**  
Press to read control unit identification and status of immobilization. ECU hardware and software versions are displayed.

### Advanced functions

- **Read EEPROM**  
Press this button to read internal microcontroller EEPROM. Memory content can be modified and saved to disk with desired filename. Output file size – 256 bytes.
- **Clear IMMO Code**  
Press to clear immobilizer code stored in the EEPROM memory. ECU is ready to accept new code after clearing. **Engine starts, even with code cleared – NECESSARY to cut (disconnect) the immobilizer wire (pin 25) in the 35 pin connector.**

## VALEO BII Interconnection Unit for ESPACE-III ('98-'02)

### Hardware

- Microcontroller** – MOTOROLA MC68HC(9)08AZ60
- EEPROM memory** – Internal MCU's EEPROM (Memory size – 256 bytes)

### Basic functions

- **Read Identification**  
Press to read control unit identification and a state of key programming:  
**YES** – programmed control unit. Needs to be cleared to use in the other car;  
**NO** – brand new or already cleared control unit. It is ready to accept new transponder key data on the next ignition-on cycle.  
BII Interconnection Unit Hardware/Software versions are also displayed.

### Advanced functions

- **Read MC68HC908AZ60 EEPROM**  
Press this button to read microcontroller's EEPROM data. Memory content can be modified and saved to disk with desired filename.  
Output file size – 256 bytes.
- **Write MC68HC908AZ60 EEPROM**  
Press this button to write data from "Memory Dump" window to the microcontroller's internal EEPROM. Content can be modified prior to writing.  
Input file size – 256 bytes.
- **Read KM**  
Press this button to read current mileage, stored in the BII.
- **Write KM**  
Press this button to change mileage, stored in the BII. Enter desired mileage into the input box, next to a button. Valid input data range is 0 - 999999km.  
**NOTE: mileage will be changed in the BII only. No changes to mileage stored in the instrument cluster will be done.**
- **Clear for Replacement**  
Press to clear key programming data and set mileage, stored in the memory, to **000000** km. On the next ignition-on cycle, control unit will accept new key programming data from transponder key that is currently in ignition lock. Mileage is to be automatically obtained from instrument cluster.

**NOTE: Disconnect and reconnect car battery (or pull out and connect back yellow connector SS1 to the BII) after Write MC68HC908AZ60 EEPROM , Write KM and Clear for Replacement operations. Do it right before turning ignition to ON!**

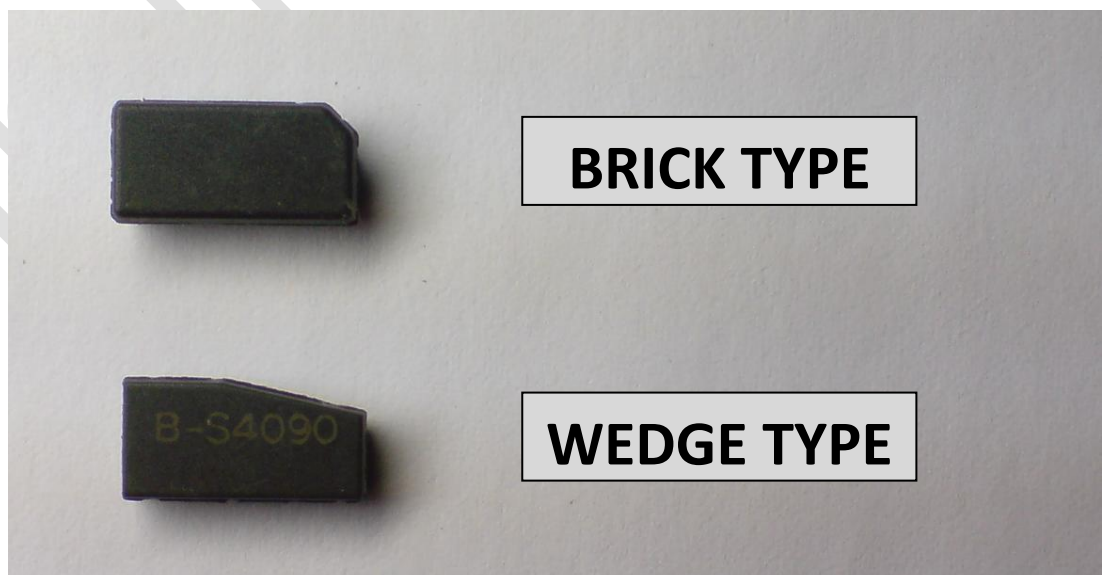
## SIEMENS UCH for MEGANE/SCENIC ('1998-'2002)

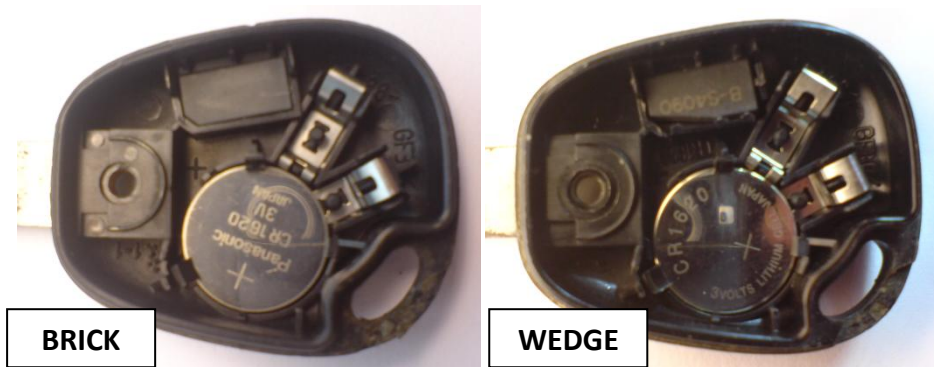
### Hardware

- Microcontroller** – MOTOROLA MC68HC705B32 (2D59J or 3G96A)
- EEPROM memory** – Internal MCU's EEPROM (Memory size – 256 bytes)

### Basic functions

- **Read Identification**  
Press to read control unit identification, configuration and a state of key programming.
- **Change UCH configuration**  
Press to customize existing UCH configuration. You can change 3 configurable parameters:
  1. **Immobilizer function enabled – YES / NO**  
**ENABLE/DISABLE** immobilizer function in the UCH. This configuration change will not make car running, but can be used to disable red immobilizer LED blinking in the instrument cluster. It is useful when immobilizer function in the Engine Control Unit is cleared, car runs, but immobilizer LED is still blinking and needs to be extinguished.
  2. **Transponder type – WEDGE / BRICK**  
SIEMENS UCH for MEGANE/SCENIC can be configured to accept two types of key transponders. Correct transponder type configuration can be recognized by its shape:





### 3. Coded Solenoid Valve – WITH / WITHOUT

Configure **WITH** for 1.9D and 1.9dT DIESEL only. Must be configured as **WITHOUT** for other diesel injection systems (MSA15.5/1.9dTi, EDC15C3/1.9dCi) and for all petrol injection systems.

- **PIN from File**

Press to calculate 8 digits PIN Code from binary file, loaded into **Memory Dump** tab. File size must be 256 bytes.

- **Start Key programming**

Press to start key programming procedure. PIN Code is necessary. It can be calculated from EEPROM dump or entered manually if it is already known. When programming additional key, already working key must be reprogrammed again.

**NOTE.** Use NEW original (never programmed) key, **BLANK** transponder TP19 or key, that has been already allocated to this car.

Up to 2 keys can be programmed.

**Procedure steps:**

- If you already have a PIN Code - enter it manually to the yellow Pin Code field;  
If you have previously read file from this UCH – load it to the Memory Dump tab and press **PIN from File** button;  
If **Read EEPROM of MC68HC705B32** was performed before, then yellow field already contains correct PIN Code.
- Press **Start Key Programming** button then Press **YES** button to confirm Key Programming. Red immobilizer LED in the instrument cluster goes OFF.
- From now you have 3 minutes to turn ON ignition with the key you want to program. **Correct key code will be indicated with series of short blinks of immobilizer LED in the instrument cluster.** If LED is doesn't blink at ignition ON, check transponder type configuration according to the pictures above. If you found out, that transponder type configuration is wrong, close key

programming procedure by pressing **NO** (exit without storing) button, then change it to correct type and restart key programming procedure. Problem can be that you are trying to program key, that has been already programmed to another car – **it is impossible to program key, somewhere used in another car.**

**When programming additional key always remember, that both transponders must have the same shape.**

- d) When one or both key codes were accepted, press **YES** to store Key Programming data.
- e) Resynchronize RF remotes if necessary. To do this:
  - Turn ignition to OFF
  - Press button on the remote control and hold it depressed – LED will flash quickly. Hold button until LED will illuminate steady.
  - Release button and press it 3 times in 2 seconds. Door will lock and unlock on successful resynchronization. Repeat for second remote control if necessary.

**NOTE.** If you would change transponder to brand new in the key with RF remote, it will be impossible to resynchronize remote control after key programming. Car would run, but RF remote control would stop working!

## BOOT Mode functions

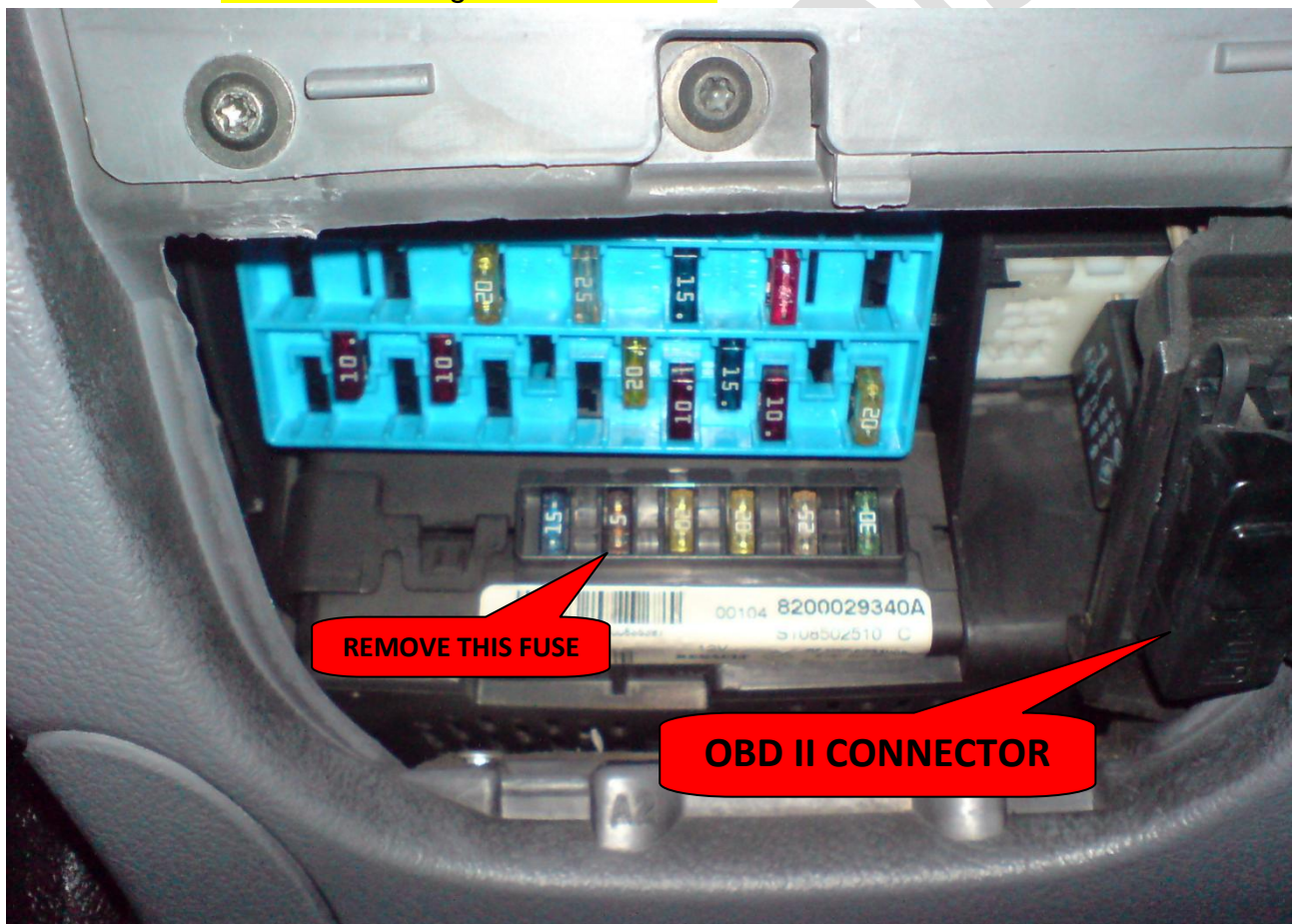
- **Enter BOOT Mode**

Press this button to force UCH to the BOOT mode, where MCU internal EEPROM Read/Write functions are available.

**Procedure steps:**

- f) Turn OFF ignition and press **Enter BOOT Mode** button,
- g) Remove 5 AMPS **ALIM UCH** fuse (FUSE-B in UCH),
- h) Press YES button to confirm entering to the **BOOT** Mode,
- i) Now you have UNLIMITED time to insert **ALIM UCH** fuse to finally enter the **BOOT** Mode. Procedure could be stopped at this step by pressing **Stop Sync** button.

Red immobilizer LED will illuminate permanently or will extinguish at all on successful change to **BOOT** Mode.



- **Exit BOOT Mode**

Pressing this button returns UCH to normal operating mode. If application was closed when UCH is in **BOOT** mode, return to **NORMAL** mode can be done by removing and reinserting **ALIM UCH** fuse or disconnecting and reconnecting UCH **PP5** connector or car battery for several seconds.

- **Read MC68HC705B32 EEPROM**

Press this button to read microcontroller's EEPROM data. Immobilizer PIN Code (8 digit) is calculated automatically after EEPROM reading is done. Memory content can be modified and saved to disk with desired filename. Output file size – 256 bytes.

- **Write MC68HC705B32 EEPROM**

Press this button to write data from "Memory Dump" window to the microcontroller's internal EEPROM. Content can be modified prior to writing. Input file size – 256 bytes.

**NOTE.** When cloning one UCH file to another, never write file from UCH with part number **82XXXXXXXX** to the UCH with part number **77XXXXXXXX** and vice versa. They have an incompatible file structure!

## Quick SIEMENS UCH repair

SIEMENS UCH for MEGANE/SCENIC ('1998-'2002) is very sensitive to dropouts of the battery voltage that can cause a loss of Key Programming or self changes to configuration. This leads to vehicle immobilization problems. Tool includes full function set to solve any software problem and to make immobilizer system operational again.

There are several most common problems and ways to repair them:

### **I. Car suddenly stopped to recognize original Key.**

Red immobilizer LED in the instrument cluster is blinking slowly with ignition OFF and starts to blink rapidly on ignition ON. Car was driven with this Key before a problem appeared.

**Solution:**

- Press Read Identification button
- If Number of programmed keys field value is 0 – perform Key programming procedure as described above. If its value is 1 or 2 – check if Transponder type field matches actual transponder shape (see picture in the Change UCH configuration section). Change it to correct type if necessary by pressing Change UCH Configuration button.

### **II. Red immobilizer LED in the instrument cluster doesn't blink at all.**

**Solution:**

Check if front driver side window, wipers and turn signals are operational with ignition ON. If they are not – check ALIM UCH fuse (FUSE-B in the UCH, 5A). If these functions are OK, configuration and Key Data is lost. Then:

- Press Read Identification button

- If **Key programming done** field value is **NO** – perform **Key programming procedure** as described above. Besides, also check if **Immobilizer function enabled** field holds value **YES**. Change it to **YES** if necessary by pressing **Change UCH Configuration button**.

**III. Red LED in the instrument cluster is blinking same way (slowly) with ignition OFF or ON.**

**Solution:**

It means, that UCH has no +12V\_After\_Ignition power supply. Check for blown FUSE (basically a 15 AMPS fuse for the rear window wiper).

Immo-Tools.it