MSS50/MSS52/MSS54/MSS54HP DIY DME Modification

The purpose of this document is to guide its reader through a successful modification of the MSS50 (E36 M3), MSS52 (E39 M5), MSS54 (E46 M3) or MSS54HP (E46 M3) in a comprehensive fashion without requiring the readers to overly burden themselves with a full understanding of the technical details of the processes involved.

Disclaimer

It's a terrible disservice to the enthusiast community that disclaimers are required, and while the threat of litigation and liability is ever increasing, it should not be used as a deterrent to those providing information au gratis. Thus, herein lies a relatively non-comprehensive disclaimer.

PERFORMANCE WARRANTY DISCLAIMER - By installing or attempting, by any means, any information, software, services, suggestions, tune, or tuning idea contained within this document, the reader both understands and agrees, unconditionally, that any such altered performance products, information, software, services, suggestions, tunes, or ideas may not perform as intended, or at all. Individual results may vary depending on make, model, year, and fitment for any particular use is not guaranteed. In no way shall the contributors of this document, or site administration, site owners, or sponsors be held responsible and/or liable for any results based on information claimed or unclaimed, that may or may not come from the direct or indirect use of information, software, services, suggestions, tunes or tuning ideas contained within. *YOU ARE ON YOUR OWN*. Any information, software, services, suggestions, tune or tuning ideas can and most will inhibit the ability of the car to perform as intended by its original manufacturer. It is also entirely likely that the information, software, services, suggestions, tune or tuning ideas contained within will interfere with time-tested OEM functionality, diagnostics and local emissions compliance. *USE AT YOUR OWN RISK*.

LIABILITY DISCLAIMER - In no way shall any contributor hereby referred to as an administrator, document owner, forum member or forum sponsor be liable for the information, software, services, suggestions, tune, or tuning idea or other such nomenclature contained within this document, regardless of proper or improper utilization. Those who actively choose to follow any advice, or seek the counsel of any other person in accordance with the information voluntarily submitted within this document accept the consequences of their actions or inactions without recourse. The user of this information, software, services, suggestions, tune or tuning idea assumes all liability and agrees to waive any and all remedies and liabilities, claims for remuneration, expressed or implied arising by law or otherwise, including without obligation by those volunteering any such information, software, services, suggestions, tune or tuning idea with respect to fitness or consequential damage, either by negligence or direct consequence. The contributors to this document also waive any such liability for personal injury or damages that may arise for any reason, not withstanding injury or damage to other persons. The user acknowledges that he or she is not relying on the contributors to this document to have any such skill or judgment to furnish correct or proper information, software, services, suggestions, tune, or tuning idea suitable for any particular purpose and that no liabilities exist whatsoever. By agreeing to these terms, either actively, passively or not at all, the user of the information, software, services, suggestions, tune, or tuning idea contained within agrees to indemnify the contributors to this document and to hold any such contributors harmless from any claim(s) related to the aforementioned information, software, services, suggestions, tune, or tuning idea. Under NO circumstance will any participant of this document be held liable for any damages or expenses for any reason, especially by use of the information, software, services, suggestions, tune, or tuning idea. Contributors to this document assume zero liability regarding the improper installation or misapplication of the information, software, services, suggestions, tune or tuning ideas contained herein. It is the user's responsibility to make proper judgment decisions when making changes to their own vehicle.

Nomenclature

- **Binary** – using or denoting a system of numerical notation that utilizes base 2 rather than base 10 or base 16
- **Bit** – a unit of information expressed as either a 0 or 1 in binary notation
- **Byte** – a group of binary digits or bits (usually 8) operated on as a unit
- **Decimal (dec)** – relating to or denoting a system of numbers and arithmetic that utilizes base 10, tenth parts, or powers of 10
- **DME** – Digital Motor Electronics, controls all key aspects of the engine's operation, ensuring optimum reliability, maximum performance and the lowest possible fuel consumption and emissions
- **Dump** – a computer file which contains a copy of the data contents from a read-only memory chip
- **EPROM** – Erasable Programmable Read Only Memory - a read-only memory whose contents are stored regardless of power and can be erased and reprogrammed
- **Flash** – electronic non-volatile computer storage device that can be electrically erased and reprogrammed
- **Hexadecimal (hex)** – using or denoting a system of numerical notation that utilizes base 16 rather than base 2 or base 10; often prefixed by '0x' or '0h', or alternately, an 'h' suffix
- **MSS50** – variant of the DME found in the E36 ///M3
- **MSS52** – variant of the DME found in the E39 ///M5 and E52 Z8
- **MSS54** – earliest variants of the DME found in the E46 ///M3 and E36 Z3M
- **MSS54HP** – later variants of the DME found in the E46 ///M3 with twice as much EPROM storage space as the MSS54
- **Offset** – an integer indicating the distance from the beginning of the object up until a given element or point, presumably within the same object
- **Word** – a fixed sized group of bits that are handled as a unit by the instruction set and/or hardware of the processor (16 bits for the Motorola 68k found in the DME)

Necessary Tools

Modified FTDI-based (FT232RL) VAG Com OBDII to USB cable

- The cable required for this procedure is a standard VAG Com cable modified for use with EDIABAS/INPA/NCSExpert/WinKFP/Progman/SSS/DIS. The procedure to modify a cable is not covered by this document; it is presumed that the cable works reliably with EDIABAS-based software (INPA, NCSExpert, WinKFP) prior to the initial flashing procedures.
- To date, some problems have arisen from FT232BM chipsets. Only FT232RL chipsets are supported at the time of this writing. The version of chipset can be determined using either of FTDI's utilities, FT_Prog or MProg, both of which are available online for free download.

10A Battery Charger/Tender

- The DME consumes considerable power; as such, leaving the ignition in a ready state will drain the battery quickly.
- Successful write operations depend heavily on a healthy power supply.
- Typical partial DME reads and writes can be performed in less than 2.5 minutes; however, full binary reads may take as long as 40 minutes.
- As battery chargers are relatively inexpensive, but the MSS54 DME isn't, nor is the replacement procedure simple, it's best not to take a chance.

Computer

- Desktop or Laptop with at least 1GB RAM and 1.5 GHZ CPU, available USB port
- 32-bit Windows XP, Vista or 7 for EDIABAS-based software: INPA/NCSExpert/WinKFP/Progman/SSS/DIS/TIS
- Windows or Mac OS X for BMWFlash

Necessary Software

BMWFlash (http://www.bimmersoftware.com/bmwflash)

- This is the cornerstone software of the project and is used to obtain both full and partial binary reads of the data contained within the DME.
- Both Read (aka dump) and Write (aka flash) operations are absolutely free.
- Available for both Windows and Apple platforms.

BMWLogger (http://www.bimmersoftware.com/bmwlogger)

- This datalogging software is used to augment BMWFlash and, in its free form, is a useful tool to check and clear DME DTCs and clear adaptations.
- Datalogging functionality currently costs $59 USD (current as of this writing) and is locked to the VIN of the DME being datalogged.
- Available for both Windows and Apple platforms; the licensing key is not platform-specific.

TESTO (http://phenoboy.kapsi.fi/testo)

- Testo is a free datalogger provided by user Pheno on bimmerforums.com. It utilizes the same cable required for BMWFlash and BMWLogger. **EDIABAS/INPA** (32-bit Windows XP/Vista/7 *only*)
- INPA is used as a fast means to read the flash counter and to reset the DME adaptations. It can also be used to extract DME program version information.
- This is commonly available online so acquisition and installation will not be covered within this document.

Hex Editing Software

- For a comparison of hex editors, please reference this Wikipedia page (http://en.wikipedia.org/wiki/Comparison_of_hex_editors) Due to a varied level of user familiarity with different programs, no further software recommendations will be made with respect to which hex editor is most optimal for this task.

FTDI Utilities

- FT_Prog (http://www.ftdichip.com/Support/Utilities.htm#FT_Prog) is used to identify and change the behavior of the FT232RL chipset-based VAG Com cable and is the latest EPROM editing software provided by FTDI, but does require .NET to be installed.
- MProg (http://www.ftdichip.com/Support/Utilities.htm#MProg) is used to identify and change the operation of the FT232RL chipset-based VAG Com cable, but is the deprecated EPROM editing software provided by FTDI, does not require .NET installation, and has been superseded by FT_Prog.

TeamViewer (http://www.teamviewer.com)

- TeamViewer is a free (for private use) remote desktop sharing suite that allows remote users to share and control the local user's screen, and also features integrated messaging and file transfer functionality.

Caveats

Embedded write-limit pseudocounter – this is a method employed by Siemens to artificially limit the number of times the DME can be flashed.

The DME can be flashed exactly 60 times before it refuses additional software flashing requests, at which point alternate flashing methods must be employed to reset the pseudocounter (which is unfortunately beyond the scope of this document). In theory, the AM29F200 and AM29F400 chips are capable of >100000 write cycles.

The first and subsequent flashes from BMW factory installation and software updates may consume several write slots. If a previous owner has performed any software updates, these will detract from the number of available writes.

The fastest way to determine exactly how many writes remain is to read the first 100 bytes at location 0x800 using the Read RAM feature in INPA. This will show a string such as '12 68 A0 00 00 00 00' followed by some deterministic number of '00's or 'FF's thereafter. One '00 00' grouping represents one write slot that has been used. A 'FF FF' represents an available write slot.

To know exactly how many flashes remain requires counting the number of '00's, dividing that number by two, and subtracting from 60.

available writes = 60 – ({n}/2) (*equation 1*)

Thus, if you count 20 ‘00’s in *figure 2*:

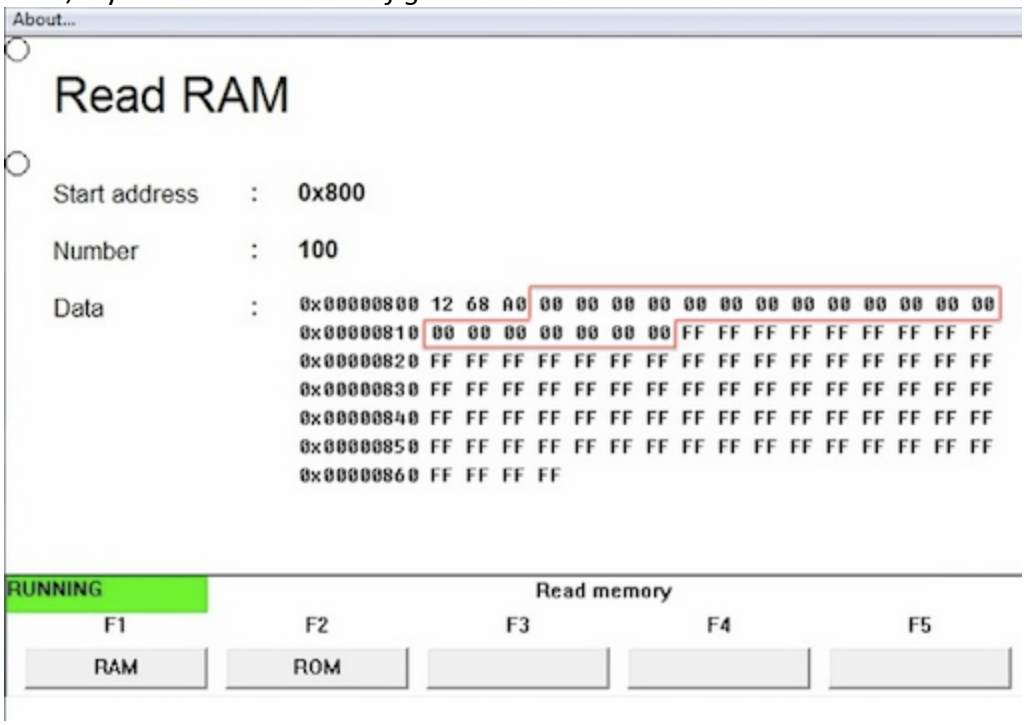

Figure 2: *INPA Read RAM screen shot used to determine the flash pseudocounter*

And consequently substitute 20 for {n} in *equation 1* as follows:

60 – (20/2) = **50 available flashes remaining** (*equation 1: sample calculation*)

The pseudocounter can also be examined by extracting a full DME read using BMWFlash. The location for the bytes occupied by the pseudocounter is at 0x4800. A full DME read can take as long as 40 minutes, which is most certain to require an external power source. (See battery charger information above)

Reading (dumping) the Parameter Space

Connect the battery charger to the car, then connect the OBD II cable between the car and computer, turn the ignition to position 2 (but do not start the car), and start the BMWFlash software.

Click on **File → Read DME**

A prompt will then ask '**Do you want to download the entire ROM contents?**'

Click '**No**' to take a partial read, consisting only of the parameter space, or click '**Yes**' to take a full read, which will provide a full dump of both EPROMs.

Do not remove the battery charger, cycle the ignition or unplug any cables during this procedure – it will take approximately 2.5 minutes to extract the parameter space.

Once the procedure has finished, the software will ask to save the file. Enter the desired file name and location. *It is highly recommended to use some kind of file name versioning system to differentiate between the original parameter space binary and the modified binary.*

For example:

`ayrton.senna_v0.1.0.bin` = *original parameter space dump*
`ayrton.senna_v0.2.0.bin` = *first major modification*
`ayrton.senna_v0.2.1.bin` = *first minor modification after major*
`ayrton.senna_v1.0.0.bin` = *first re-calibrated operational adjustment*
`ayrton.senna_v1.0.1.bin` = *first minor modification after recalibration*

Identifying the DME's Hardware Variant and Program Version

There are over a dozen different program versions across both DME variants. The tuning parameter space is matched to its corresponding program version so the parameter space from one DME *may not be (and very likely isn't) compatible with another*, unless the program versions are an *exact* match. The locations of maps, curves and values are different between program versions, so the DME's program version must be determined before proceeding to make modifications to the DME parameter space!

While the MSS50, MSS52, MSS54 and MSS54HP DMEs have two EPROM chips onboard, they are not identical in size or configuration. The MSS50 has two 256 KB chips (AM28F200), MSS52 and MSS54 have two 256 KB chips (AM29F200) and the MSS54HP has two 512 KB chips (AM29F400). The extracted sizes of both parameter space, program space, and a full DME read are provided in *Table 1*.

The first step is to identify the variant of the DME by looking at the partial read (aka parameter space or calibration). The fastest way to identify the DME variant is by examining the resultant file size of the parameter space dump. If it is a 32 KB file, it is the MSS50, MSS52 or MSS54 (non-HP) variant. If it is a 64 KB file, it is the MSS54HP variant. The E46 ///M3 CSL utilizes a variation of the MSS54HP with a minor hardware augmentation, and is also 64 KB.

To date, the *most accurate* way to determine both the DME variant and program version is by the 16-byte version string located at 0x7FB8 on the MSS50, MSS52 and MSS54 and 0xBFB8 on the MSS54HP. This will contain a series of characters such as '2113**2300****2701**J469'. Bytes 5-8 (**2300**) will denote the DME variant, as illustrated in *Table 1*. The segment located in bytes 9-12 (**2701**) will denote the program version. Bytes 13-16 (**J469**) will denote a difference in calibration, but with equivalent parameter offsets to other calibrations within the same program version group.

Table 1: *DME variant version string identification information*

DME Variant (platform)	Version ID	Parameter Space	Program Space	Full DME Read
MSS50 (E36 M3)	1800	32768 B (32 KB)	128 KB	512 KB
	1801	32768 B (32 KB)	128 KB	512 KB
MSS52 (E39 M5)	2100	32768 B (32 KB)	192 KB	512 KB
MSS54 (E46 M3)	2200	32768 B (32 KB)	192 KB	512 KB
MSS54HP (E46 M3)	2300	65536 B (64 KB)	512 KB	1024 KB
MSS54HP (E46 M3 CSL)	2500	65536 B (64 KB)	512 KB	1024 KB

The version can also be somewhat determined by opening the partial read using a hex editor and looking at the two bytes at 0x2 representing the program number, though there is some minor variance and overlap for program numbers 4.26, 4.27, 5.19, 5.22, 1.08 and 1.12. *The 16-byte version string will serve as **the best** method for accurately determining the hardware number.*

Table 2 will illustrate the cross reference to program version with respect to hardware number, version string, program number, DME variant and finally, file size.

Table 2: *Hardware number correlation and identification*

Hardware Number	Version String	Program Number	DME Variant	File Size
7831122	2113 2100 0502	4.14 (01 9E)	MSS52	32 KB
7832339	2113 2100 0901	4.20 (01 A4)	MSS52	32 KB
7833965	2113 2100 1301	4.26 (01 AA)	MSS52	32 KB
7835278	2113 2100 1401	4.26 (01 AA)	MSS52	32 KB
7835620	2113 2100 1501	4.27 (01 AB)	MSS52	32 KB
7837965	2113 2100 1601	4.27 (01 AB)	MSS52	32 KB
7843317	2113 2100 1701	4.28 (01 AC)	MSS52	32 KB
7831815	2113 2200 0701	5.03 (01 F7)	MSS54	32 KB
7832594	2113 2200 0901	5.08 (01 FC)	MSS54	32 KB
7832868	2113 2200 1101	5.10 (01 FE)	MSS54	32 KB
7833145	2113 2200 1301	5.13 (02 01)	MSS54	32 KB
7833284	2113 2200 1401	5.?? (02 ??)	MSS54	32 KB
7833892	2113 2200 1501	5.18 (02 06)	MSS54	32 KB
7835584	2113 2200 1701	5.19 (02 07)	MSS54	32 KB
7837941	2113 2200 2101	5.19 (02 07)	MSS54	32 KB
7842353	2113 2200 2401	5.21 (02 09)	MSS54	32 KB
7842559	2113 2200 2601	5.22 (02 0A)	MSS54	32 KB
7842975	2113 2200 2801	5.22 (02 0A)	MSS54	32 KB
7834910	2113 2300 1601	1.04 (00 68)	MSS54HP	64 KB
7835564	2113 2300 1801	1.07 (00 6B)	MSS54HP	64 KB
7836454	2113 2300 1901	1.08 (00 6C)	MSS54HP	64 KB
7837805	2113 2300 2001	1.08 (00 6C)	MSS54HP	64 KB
7842355	2113 2300 2301	1.11 (00 6F)	MSS54HP	64 KB
7842569	2113 2300 2501	1.12 (00 70)	MSS54HP	64 KB
7842981	2113 2300 2701	1.12 (00 70)	MSS54HP	64 KB
7834981	2113 2500 0301	2.10 (00 D2)	MSS54HP (csl)	64 KB
7837340	2113 2500 0401	2.11 (00 D3)	MSS54HP (csl)	64 KB

Modifying the Parameter Space

Several methods exist to modify the binary parameter space dump extracted from the DME using BMWFlash. The least sophisticated method is to use a Hex Editor, and will be the initial focus of this section. Alternately, the free software TunerPro can also be used with an appropriately crafted .xdf template file, or WinOLS (non-free) also with an appropriately crafted damos (.a2l) template file. Bear in mind that each program version will require a different .xdf or .a2l, which are not ready for public utilization at the time of this writing.

It is crucial to have a fundamental understanding of the three data types, as well as the three types of characteristics as they are organized within the parameter space, before making changes within the parameter space binary.

Data Types:

1. *byte* – range of 0 – 255 if unsigned, and -128 to 127 if signed
2. *word* – range of 0 – 65535 if unsigned, and -32768 to 32767 if signed
3. *long* – range of 0 – 4294967295 if unsigned, and -2147483648 to 2147483647 if signed

Characteristics:

1. **Value** – consists of a single *byte*, a *word* (two bytes in the case of the Motorola 68k CPU used on the DME), or a *long* (four bytes in the case of the Motorola 68k..) and may be *signed* or *unsigned*.
2. **Curve** – consists of an index and a single X-Axis comprised of *byte*, *word* or *long* values.
3. **Map** – consists of an index and both X- and Y-Axes comprised of *byte*, *word* or *long* values.

The smallest characteristics, and consequently easiest to modify, will be those defined as *values*. The largest characteristics, and most difficult to modify, will be those defined as a *maps*.

Both curves and maps have a small *word* preceding them, which will aid in the identification of the curve or map. For example:

`0A 00:` 10 (0x0A) points on the X-axis, 0 (0x00) points on the Y-Axis = *curve*
`0E 17:` 14 (0x0E) points on the X-axis, 23 (0x17) points on the Y-Axis = *map*

The location of each *value*, *curve*, or *map* within the parameter space is given by an *offset*. These offsets can vary heavily based on the program version and variant of the

DME. The set of tables below will assist in locating some of the more common/popular characteristics and are organized by DME variant, then further by program number.

Another consideration to bear in mind is that not all bytes, words, or longs will reflect their common units. Most will have a computation applied. This is usually defined in an accompanying DAMOS (.A2L) file, though these files are not known to exist for every version, or not freely available. Most of what has been discovered and freely distributed is due to the diligence and hard work by forum members trawling throw an old DAMOS file to match values from a matching program version to other versions lacking DAMOS files. Unfortunately, at the time of this writing, an explanation of computation methods is beyond the scope of this document.

Table 3: *Commonly-modified MSS54 program version functional parameter locations (to be updated in upcoming revisions to reflect updated platform knowledge)*

	0701	1501	1701	2101	2601	2801
6MT Shift Lights	0x28CB	0x1A7D	0x1AB3	0x1AB3	0x1AB3	0x1AB3
Sport Throttle Sensitivity	0x4188	0x4174	0x4174	0x4174	0x4174	0x4174
Comfort Throttle Sensitivity	0x41B2	0x419E	0x419E	0x419E	0x419E	0x419E
Engine RPM limits	0x03B4	0x03AE	0x03AE	0x03AE	0x03AE	0x03AE
Speed Limits/gear	0x6E70	0x6F00	0x6ED0	0x6ED0	0x6ED0	0x6ED0
Sport Mode Functionality	0x4024	0x4024	0x4024	0x4024	0x4024	0x4024
Oil Temp Warm-Up Lights	0xAC74	0x5454	0x5370	0x5370	0x5370	0x5370

Table 4: *Commonly-modified MSS54HP program version functional parameter locations (to be updated in upcoming revisions to reflect updated platform knowledge)*

	1601	1801	0401	2001	2501	2701
6MT Shift Lights	0x28C9	0x29CB	0x28CB	0x29CB	0x29CB	0x29CB
Sport Throttle Sensitivity	0x8228	0x8228	0x8228	0x8228	0x8228	0x8228
Comfort Throttle Sensitivity	0x8252	0x8252	0x8252	0x8252	0x8252	0x8252
Engine RPM limits	0x03E2	0x040A	0x03E2	0x040A	0x040A	0x040A
Speed Limits/gear	0x8F92	0x8F92	0x92AE	0x8F92	0x8F92	0x8F92
Sport Mode Functionality	0x8026	0x8026	0x8026	0x8026	0x8026	0x8026
Oil Temp Warm-Up Lights	0xB076	0xB076	0xAC74	0xB076	0xB076	0xB076

The sample below will illustrate how to enable shift lights on a E46 M3 with a manual transmission and a previously-determined program version of 2300 2001, indicating it is a MSS54HP (2300, see *Table 1*).

First, reference the appropriate offset in either *Table 3* or *Table 4* based on program version. For version 2001, the offset for shift light value is 0x29CB.

Within the parameter space binary, using a hex editor, note the location and observe the bytes at that location as illustrated within *Figure 3*.

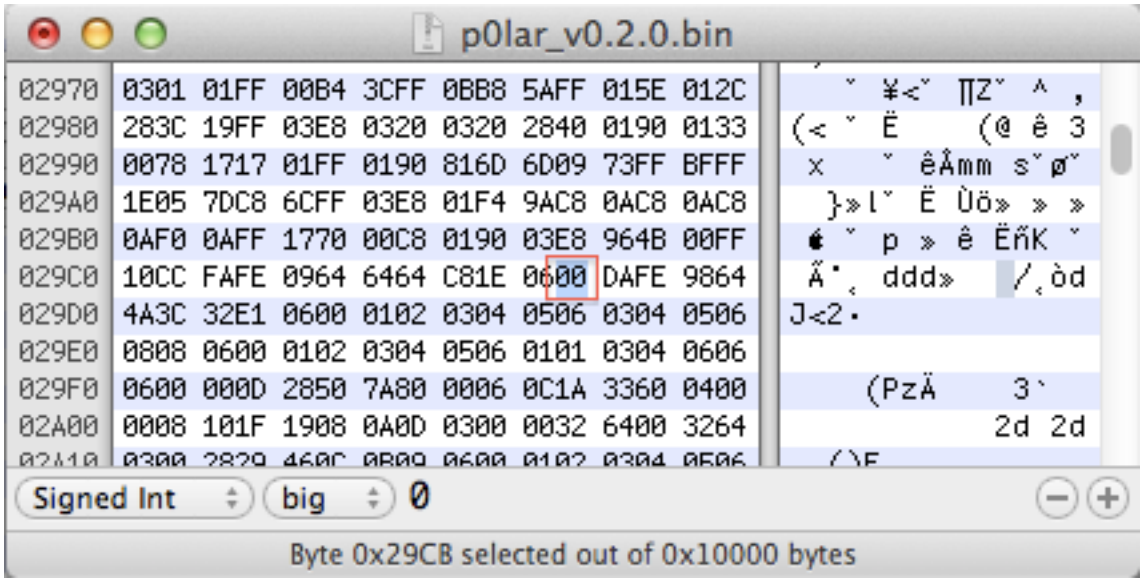

Figure 3: *Program version 2001, shift lights location in hex editor*

In the case of shift lights, a value of '00' means disabled, '01' means enabled always, and '02' means enabled with sport mode only. Change this value to '01' to enable shift lights regardless of sport/comfort mode, save the file, and proceed to the next modification.

Once all modifications have been performed to the parameter space binary, proceed to the next step, *Writing (flashing) the Modified Parameter Space.*

Writing (flashing) the Modified Parameter Space

After modifications to the parameter space have been saved, verify that the file is still the correct size according to the DME type within *Table 1*, and then connect a battery charger to the vehicle.

Connect the OBD II cable between the vehicle's OBD II port and computer, turn the ignition to position 2 (but do not start the car), and start the BMWFlash software. If the cable is properly detected, the screen will show a green square in the lower left corner as indicated in **Figure 4**.

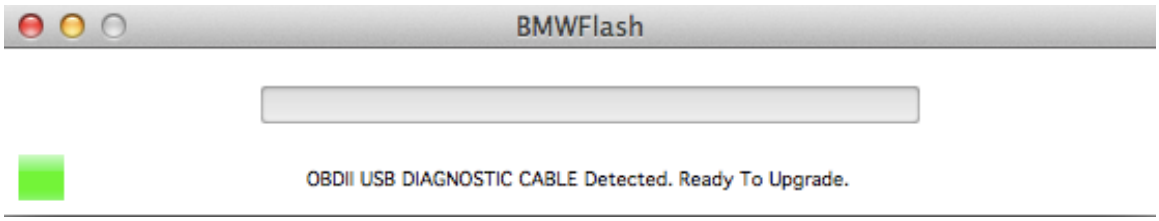

Figure 4: *BMWFlash: ready to upgrade*

Click on **File → Reflash DME**.

A prompt will then ask 'Are you sure you want to reflash your DME? If not powered properly during the reflash problems can occur.'

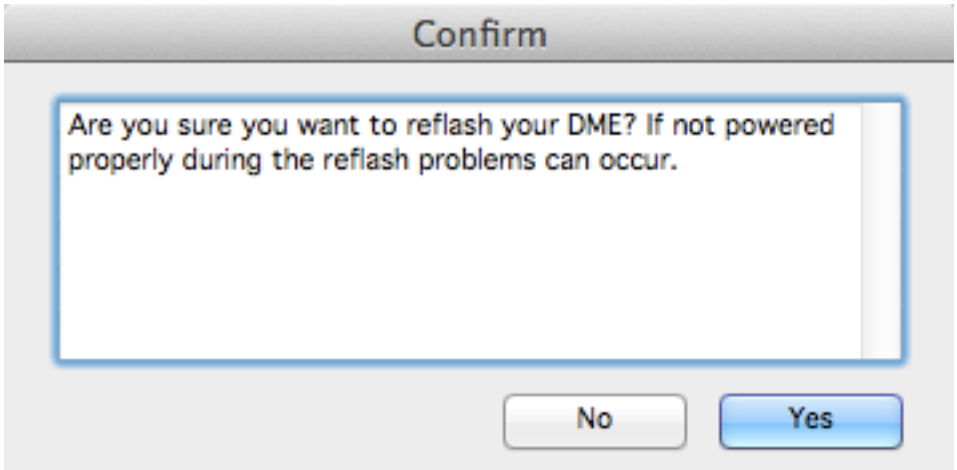

Figure 5: *BMWFlash pre-upgrade warning*

Upon clicking '**Yes**', a dialog will be presented asking for the location of the modified parameter space binary file. Locate the file and press '**Open**' and the flashing procedure will commence.

Do not remove the battery charger, cycle the ignition or unplug any cables during this procedure – it will take approximately 2.5 minutes.

Finalizing the Procedure

To finalize the modified parameter space write to the DME, the ignition must be turned off for approximately 15-20 seconds, then turned on again. If any operational changes to various engine parameters have been modified, it may be necessary to clear the DME adaptations using INPA or BimmerSoftware's free BMWLogger utility.

Frequently Asked Questions

Q: I have a Galletto 1260 and the accompanying software; can I use this tool/software rather than a modified VAG Com cable and BMWFlash?
A: Absolutely; however, be advised that the Galletto software flashing routines will not correct both CRC16 checksums located within the parameter space. These will need to be manually corrected prior to flashing or the DME will refuse to start.

Q: Do I have to correct the checksums before flashing?
A: Only when using the Galletto 1260 to write the modified parameter space. BMWFlash will correct the checksums inline on behalf of the user.

Q: Where do I purchase a proper VAG Com cable, and what modifications must be performed to make it compatible?
A: These can generally be purchased via eBay, but confirm with the seller that it indeed has the FTDI FT232RL chipset and *not* the FT232BM chipset or problems may ensue. Pins 7 & 8 must be soldered together inside the OBD II 16-pin head of the cable, then invert the RI and DSR signals and set C0 = RXLED#, C1 = TXLED#, C2 = PWRON#, C3 = PWRON#, C4 = SLEEP# using FT_Prog or MProg, then save the template and flash the cable. More detailed instructions are available for this online. Bimmersoftware may also have these available as a package purchase with a BMWLogger application.

Q: I'm still confused; can someone please help me from remote?
A: Likely! Download and install TeamViewer (http://www.teamviewer.com/en/index.aspx) and post a request in any of the various support threads located in the forum. It is very likely that someone will be willing to assist for free, or for a very nominal fee at most.

Q: My car has been tuned by {xyz} tuner. Will this affect the tune?
A: No. The methods presented in this document will allow you to alter the parameter space where the tune is located, but unless you overwrite the maps/curves/values used to specifically tune the vehicle, nothing will change. At the very worst, as long as the original binary was appropriately saved, you will have a free backup of your tune to revert to should something behave unexpectedly.

Q: Do I really need a battery charger?
A: It's not recommended to write to the DME without one, but the write procedure is less than 3 minutes, so the likelihood of draining the vehicle's battery is very low unless it is already in poor condition.

Q: BMWFlash can't find my cable, what's wrong?

A: Check the type of chipset utilized within the cable using the FTDI utilities FT_Prog or MProg to ensure it is definitely the FT232RL-based cable.

Q: How long does it take to register BMWFlash?

A: BMWFlash is now free, registration is no longer required.

Q: What hex editor do you recommend I use?

A: Almost any hex editor will work, the reason no specific software is recommended is because the interfaces vary to a great degree. Some people will find particular interfaces more comfortable than others so it's impossible to tell what someone will feel most comfortable using, which is very important when making fine-grained changes that affect the operation of the engine.

Q: Is this procedure dangerous? Can I 'brick' my DME?

A: Nothing concerning the MSS50, MSS52, MSS54 and MSS54HP DME is trivial; however, the routines required to flash the DME are not contained within the parameter space that governs functional and operational engine characteristics. As such, the likelihood of 'bricking' the DME is extremely slim since the boot sector and program space remain untouched. At worst, the vehicle simply won't start and a reflash will be required. Mind the flash pseudocounter flash and battery voltage level and most problems will be avoided.

Q: Will the flashing software work on Mac?

A: Yes, and no. BMWFlash is compiled for both Windows and Apple's OS X; however, EDIABAS-based software, including INPA, NCSExpert and WinKFP, as well as Progman/SSS and DIS/TIS are exclusive to the 32-bit Windows platform.

Q: The version information can't be found, help!

A: Using your hex editor, search for a text string starting with '2113' and note the 16-byte wide string, specifically bytes 5 – 8 and 9 – 12.

Q: How do I know if my VAG Com cable has the correct chipset?

A: Refer to the MProg or FT_Prog software utilities provided by FTDI to make the proper determination.

Q: My car was previously tuned by tuner {xyz}, and BMWFlash cannot read my DME, but BMWLogger works, what's wrong?

A: There is a distinct possibility that your DME has been involuntarily vendor-locked by your previous tuner. This is a tired, cumbersome practice and is usually improperly implemented to defeat exceptionally weak intellectual property theft techniques. Please contact tuner {xyz} for a solution.

Works Cited|References

- **Wikipedia**: for certain explanations and dictionary references
- **Google**: dictionary references, general information and basic search criteria
- **m3forum.net**: where the purity of this movement really took form and most of whose administration have gladly hosted our efforts without impeding progress. The bulk of the information shared on the MSS50, MSS52, MSS54 and MSS54HP DME can be located as a sticky at the top of the E46 M3 Coding and Tuning section of this forum.

Credits

- These are too numerous to specify, as there have been so many contributors. Eventually, this section will be populated; for now, please be patient as those who have donated their time and energy to this project will not go unnamed and are very well aware of who they are. To each and every one of them – *Thank you.*

Revision History

- v0.0.1 (p0lar) – initial draft release to proof/edit/test
- v0.0.2 (p0lar) – revision with minor corrections, grammatical modifications, more clarifications
- v0.0.3 (p0lar) – revision with more minor corrections, addition of more data to Table 1 (including MSS50 and MSS52 version information) as well as the beginnings of MSS50 and MSS52 inclusion in the process since their architectures are extremely similar.
- v0.0.4 (p0lar) – minor revision with error corrections
- v0.0.5 (p0lar) – minor revision to format, addition of Testo and BMWLogger software(s), preparation for DIY performance tuning strategies, update and inclusion of MSS50 and MSS52 program information as well as updates to MSS54 information.
- v0.0.6 (p0lar) – minor revisions to format, BMWLogger no longer requires registration, added two versions to the MSS54 version list, verbiage.
- v0.07 (p0lar) – minor revisions to syntax, BMWFlash -> BMWLogger, additional info about MSS50 chips/location of version strings.