BMW E60 LCI Dynamic Cruise Control Retrofit S544A



Cruise Control Stalk: 61 31 6 951 352 Steering Column Lower trim: 61316947773 Steering Column Front trim: 61316947778 Coding Cable: D-Can/K-Line Cable or ICOM

INTRODUCTION

Some BMW E6x LCI were factory-lacking a standard option, which is Cruise Control, even some 530i and 630i models. For E6x Pre-LCI, the cruise control retrofit was straightforward (Changing the Steering column Unit with another unit that has the Cruise control switch, VO-coding CAS and LM modules and coding the car to default). Starting from LCI (March 2007 onwards), officially in most cases, the retrofit requires replacing the whole instrument cluster with a version that supports Dynamic Cruise Control. Yes, the standard Cruise Control (Option S540A) is no longer possible, instead option S544A was implemented as standard.

However, for a E6x LCI that came without cruise control, retrofit of DCC (Dynamic Cruise Control) without replacing the instrument cluster is still possible by modifying the hard-coded data in the instrument cluster factory-settings (Settings that cannot be modified by standard coding tools).

One needs to check whether the current instrument cluster has a high-variant (Cruise control led pointer around speedometer ring is present) or a low-variant (led pointer around speedometer ring is absent). A KI-Test (Instrument Self-test, through the hidden menu or through INPA, ISTA+, Tool32 ...) can reveal which variant it is.

However, some parts are still needed. Below is the practical procedure for this retrofit.

THE CAR'S HARDWARE

The subject car came with a low-variant instrument cluster and a high-variant steering column unit:

	Labeled BMW Part	BMW Part Number	ZUSB
SZL	?	9204505	9204506
KOMBI	9153759	9196108	9194893

ADD VEHICLE ORDER S544A

The tool needed is NCSExpert.

- Open NCSExpert and load an Expert profile (Manipulation must be disabled)



SW/PSW	×
Read FSW/PSW:	
C FSW/PSW processing inactive	
from CVT	
C From file	_
FSW-/PSW-Trace	
FSW-/PSW-Manipulate with FSW_PSW.MAN	

- Step 1: Read the car using CAS ECU.
- Step 2: Choose Job '*Enter FA*' and enter the option *\$544* in the Attribute field and click 'Add'.
- Step 3: Once done, click OK and click 'back' and change Job to 'FA_write', hence execute the job. Repeat this process (Step 3) for LMA ECU, by changing ECU from CAS to LMA and executing 'FA_write' job.

Up to this stage, the Option S544A was written to both ECUs CAS and LMA. INFO: CAS ECU contains VO information. (Vehicle Order) which is redundant onto LMA Ecu.



Enter FA Attribute:		Chasis: E60_	×	
Add				
#0307 \$205 \$25B \$346 \$354 \$423 \$428		•		
\$431 \$439 \$441 \$442 \$473 \$548	•			
ОК				

- Warning: Do not close NCSExpert or change the profile.

You can code the car with Ignition off. But pay attention, the car can go on Sleep mode. Also, make sure the battery has at least 12.2 Volts.

CODE THE CAR TO DEFAULT

 Delete errors, using INPA or ISTA+ or any OBD scanner compatible with BMW.

Before continuing, make sure the option S544 was added to your VO. You can use NCSExpert or ISTA+ (Vehicle Information).

- Step 1: In NCSExpert change the Job to 'sg_codieren'.

- Step 2: Click on 'Process ECU' and choose SZL Ecu.
- Step 3: Execute the job.
- Repeat Step 2 and 3 for the following ECUs: KOMBI (KMBI), DSC (EHB3)..

Up to now, if you plug the car to INPA or ISTA and check for errors, you should find a new generated error that says:

DSC: DCC: Error in interface to Instrument cluster.

Which is pretty logical, since the DSC is trying to communicate with the DCC interface in the instrument cluster but this latter is still deactivated.

HACK INSTRUMENT CLUSTER

- Delete errors.
- We use BMW Tool32 and load SGBD
 'KOMB60' (In C: \EDIABAS\ECU).
- We select the job '**c_c_lesen**'.
- In 'Arguments' field, check the DATA checkbox and paste the following hexadecimal byte code: 01 01 01
 01 00 00 00 00 00 00 00 00 00 00
 17 00 00 00 00 01 31 00 FF FF
 FF FF FF FF FF FF FF FF FF FF FF
 FF FF FF FF FF FF FF FF FF FF FF
 FF FF FF FF FF FF FF FF FF FF FF
- Double-click on the job 'c_c_lesen'.
- In the Results page, copy the CODIER_DATEN byte code. In our
 case, the result was: 01 01 01 01 00
 00 00 00 00 00 00 00 00 17 00 00
 00 01 31 00 8D 40 01 00 A0 00
 18 01 90 01 08 02 26 01 36 14 98
 22 FB 30 5D 3F F8 03
- The result below may differ from car to another.

 Copy and save the result below on something safe (Google Drive, USB Key, Email ...etc). In case things go wrong, the original CODIER_DATEN byte coded would serve to restore default settings.

As you can see, the FF buffer is an output buffer that was replaced by the coding data of our instrument cluster and the byte highlighted in red (F8) is the cruise control pointer.

Just for information the instrument cluster coding data is addressed as follows:

- 0x3100 Speedometer
- 0x3101 Cruise Control Marker
- 0x3102 Fuel gauge
- 0x3103 RPM gauge
- 0x3104 Redline Marker
- 0x3105 Oil Temp / Instant MPG

So what is that F8?

It is simply the byte value that decides whether DCC is active or not; specifically the pointer ACC_ZEIGER_VERBAUT (Cruise control pointer installed?).

	Hex	Decimal	Binary
DCC disabled	F8	248 (even)	11111000
DCC enabled	F9	249 (odd)	1111100 1

So as illustrated in the table below, we're going to change F8 to F9 in order to activate cruise control, thus, we modify the CODIER_DATEN block with the modified value and write it using Tool32:

01 01 01 01 00 00 00 00 00 00 00 00 00 00 17 00 00 00 00 01 31 00 8D 40 01 00 A0 00 18 01 90 01 08 02 26 01 36 14 98 22 FB 30 5D 3F F9 03

But before we write the new code, we need to calculate its checksum. Otherwise we will get errors later.

- In Tool32, we select the job 'c_checksumme'.
- We paste the modified code below and check the DATA checkbox and double-click on 'c_checksumme'
- In the results window, we should get a new CODIER_DATEN code. Copy it.

In our case, here is a comparison between he modified code with the old and new checksum:

Modified code before checksum calculation:

After Checksum calculation:

The code in green is the code we are going to write in our instrument cluster:

 Still in Tool32, we select the job 'c_c_schrieben'

4/5 - October 2019 <<u>https://bmw.opentika.ovh</u>> kyoshuu.madani@gmail.com

- We paste the code in green, in the Arguments field, check DATA checkbox and double-click on 'c_c_schrieben'.
- The Status execution of the job should return OKAY.
- Execute again the job 'c_c_lesen' as described before in order to make sure the new data has been written successfully.

Note that the main purpose is to modify the last part of the binary value from 0 to 1.

RESET INSTRUMENT CLUSTER

Up to now, DCC is activated but we still need to reset the instrument cluster.

- In Tool32, we select the job 'steuergeraete_reset'. Nothing on arguments and DATA checkbox NOT checked.
- We double-click on 'steuergeraete_reset'.

The job status must say OKAY and time is also reset.

SET TIME AND DATE

Use the IDrive (Settings) to set time and date or Tool32 job 'sg_reset_ohne_uhr_datum'.

RESULT



NOTES

This procedure worked fine for the subject car which is a BMW E60 523iA LCI of 2007 and many others.

By following this guide you take full responsibility. The author of this guide is not responsible for any consequent damage.