ASAP

Working Group on Standardisation of Application Systems

Interface Specification

ASAP3-MC

Version 2.1 – Released – 1998-08-17

ASAP3-MC

AUTOMATION / OPTIMISATION and

ECU CALIBRATION SYSTEM INTERFACE

Version 2.1 – Released – 1998-08-17

‘SERIAL’ VERSION

1 CONTENTS

Note: These specifications may be changed without prior notice.

2 REVISION HISTORY

Version	Modification	Reason	State
Version 1.0 (AVL – protocol) AVL			released Nov. 17th, 1993
Version 2.0 – German Mr. Bloder – AVL	Final version		released Feb. 7th, 1994
Version 2.0 – English Mr. Larson, Volvo	Translation to English	English version	October, 1995
Rev. 2.1 Draft 1 - 11/05/98 Mr. Weber - Siemens AG Mr. Kern - Siemens AG	Text changes for v 2.1 Cover, Figure 1, Figure 2, Glossary, Abbr., etc. Add. extended commands	New Request for ASAP 3 Inputs from Siemens	in hand May 14th, 1998
Rev. 2.1 Draft 1 Mr. Atz - ETAS GmbH Mr. Flattich – ETAS GmbH	Command rework, Additional commands	Inputs from ETAS	in hand
Rev. 2.1 – English Draft 1 Mr. Stadler – AVL	Collecting all inputs, creating final version		in hand
Rev. 2.1 – English Version Release Mr. Weber - Siemens AG	ASAP Logo updated Release notice Date: 1998-08-17	Document released	Released on August 1998

3 OVERVIEW

This interface description contains all the basic information about the integration of engine electronic systems in a higher-order test bed system (AuSy – **Au**tomation **Sy**stem) via an engine electronic calibration system (MC System – Measurement Calibration). The actual integration occurs using a standardised serial RS232 data connection and uses the ASAP3 protocol that allows straightforward command and data exchange between the AuSy and MC system.

Figure 1 shows the basic hardware structure of this integration. It clearly demonstrates that in any case the MC system is used as an intelligent relay station between the AuSy and the engine electronic device (ECU – **E**lectronic **C**ontrol **U**nit). The various ECUs can be operated via the MC system, which can also be used interactively as a stand-alone device. In the shown solution the AuSy only communicates with the MC system. Only dimensions in physical units (e.g. m³/h) with related logical name (e.g. AIRFLOW) are exchanged between the AuSY and MC system. All names or labels used in the AuSy and MC system are handeld as they are defined in an ASAP2 compliant ECU data description file. The software package used on the MC system is consequently responsible for the preparation and conversion into hexadecimal values suitable for the ECUs.

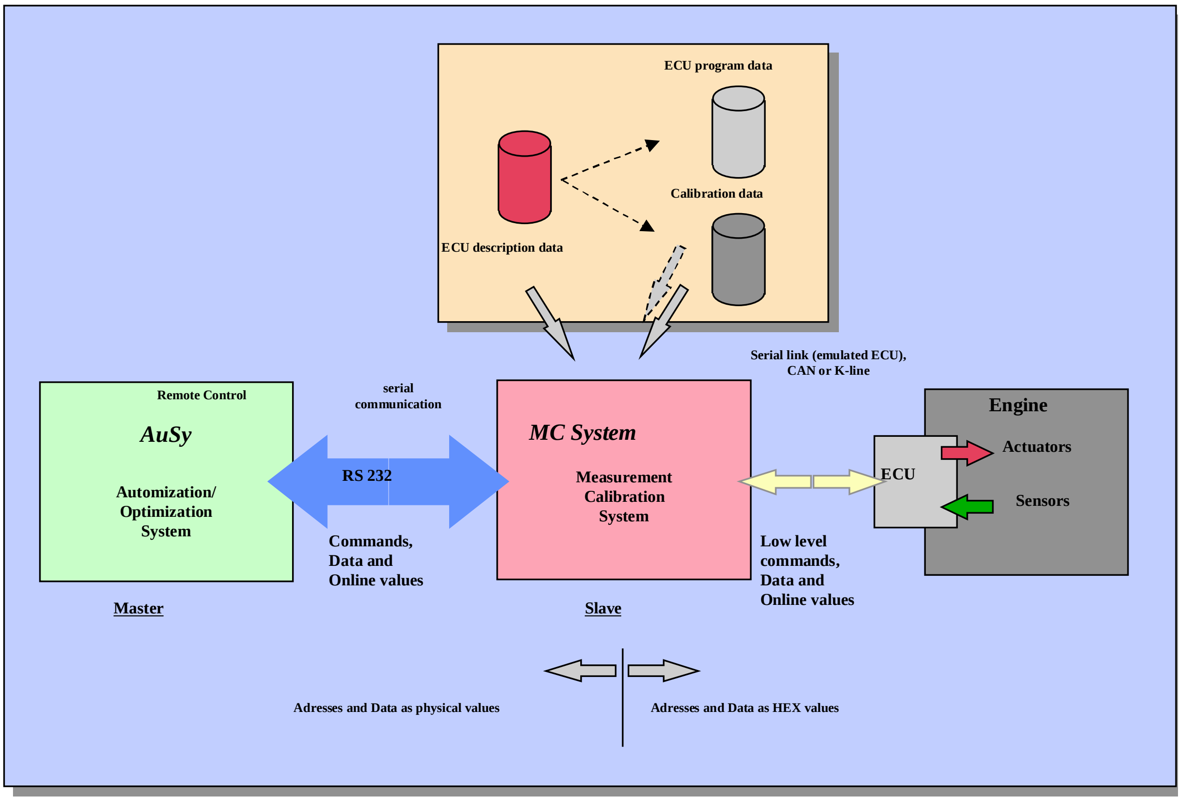

ECU program data
Calibration data
ECU description data
Serial link (emulated ECU), CAN or K-line
Remote Control
AuSy
Automization/ Optimization System
serial communication
RS 232
MC System
Measurement Calibration System
Engine
Actuators
ECU
Sensors
Commands, Data and Online values
Low level commands, Data and Online values
Master
Slave
Adresses and Data as physical values
Adresses and Data as HEX values

Figure 1

The connection with a standard RS232 line is based on the following basic settings:

9600 Baud (adjustable)
8 Bits
1 Stop Bit
no parity bit
no support of XON/XOFF
ALL COMMANDS ARE INITIATED BY THE AuSY
(STRICT MASTER-SLAVE PRINCIPLE)

The following pages detail all used data structures, data types and commands.

3.1 OPERATIONAL SAFETY OF THE ASAP3 INTERFACE

The experiences of the version 2.0 implementations have shown that there often problems occur in the communication between the AuSy and the MC System, if only small timeouts are used (see Communication Sequence, Figure 2). To get more operational safety of the interface, it is recommended to use command dependent timeouts in the AuSy. Additionally the MC System should confirm a command from the AuSy, after having received the command correctly.
Because there are different time periods for reaction within the MC Systems for operation, and therefore for acknowledgement and answering, which depend on the commands transferred from the AuSy to the MC System, it is recommended to adjust the timeouts in the AuSy corresponding to the commands which were transferred to the MC System. Additionally, it should be taken into account that the time for operation within the MC Systems are tool specific and differs depending on the tool chain being in use (MC System with memory emulation interface, CAN interface or K-Line interface). Therefore a switch to a higher baud rate will not necessarily lead to a more performed total system.

3.2 REQUEST FOR DATA ACQUISITION

The experience of working with version 2.0 using 9600 Baud as default has shown that the user requests concerning data rates are not sufficiently fulfilled.
The request from user side for data acquisition mainly lies in getting higher data rates for measurements (eg. approx. 50 channels with 10 Hz).
Because the function of data acquisition within the MC system is a tool specific solution and depends also on the interface to the ECU (memory emulation, CAN or K-Line), it is possible to get higher performance in some cases by switching to higher baud rates.

4 LIST OF CHANGES

4.1 CHANGES WITH RESPECT TO VERSION 2.0

The terms PR-S and AP-S have been substituted by AuSy and MC system (see glossary).

Note:

The new EXTENDED commands are not an official part of the standard version 2.1. They are rather supposed to be used for test implementations. The EXTENDED commands will be finally released in version 2.2. In this document, their special status is indicated by a shaded background.

4.1.1 Transmission

No changes.
Hint: To speed up the communication between the AuSy and the MC System, it is recommended to use higher baud rate of the serial link (e.g. 115 kBaud).

4.1.2 Data Formats

An additional data type of IEEE Real 8 (Double) which can be used with the 'EXTENDED' commands, which have been introduced in version 2.1

In former versions of the ASAP3 specification, the 4-Byte-Real data type was used. When min/max values of the values for the physical model are exceeding the range of the 4-Byte-Real data type or when the base type of the value is already a 4-Byte-Integer or 8-Byte-Real type, then the 8-Byte-Real type has to be used. The 8-Byte-Real data type could be used in general to simplify the implementation as long as communication speed is of less priority.
The data type STRING will be used when the value has a string conversion defined in the description (ASAP2).

Definition of the smallest negative IEEE value for invalid measurements.

4.1.3 Telegram Structure

Introduction of status $2344 for changed measuring data list.

4.1.4 Communication Sequence

No changes.

4.1.5 Extended existing commands

SET FORMAT	**Code**	**18**

The parameter 'Model' = 2 was introduced.

4.1.6 New Commands

Group **Initialisation, idendification, emergency**

EXIT	**Code**	**50**

Group **Configuration**

DEFINE DESCRIPTION FILE AND BINARY FILE	**Code**	**30**

Group **Map manipulation**

EXTENDED SELECT LOOK-UP TABLE	**Code**	**106**
EXTENDED PUT LOOK-UP TABLE	**Code**	**107**
EXTENDED GET LOOK-UP TABLE	**Code**	**108**
EXTENDED GET LOOK-UP TABLE VALUE	**Code**	**109**
EXTENDED INCREASE LOOK-UP TABLE	**Code**	**110**
EXTENDED SET LOOK-UP TABLE	**Code**	**111**

Group **Parameter manipulation**

EXTENDED GET PARAMETER	**Code**	**114**
EXTENDED SET PARAMETER	**Code**	**115**

Group **Measurement data recording**

GET USER DEFINED VALUE	**Code**	**21**
GET USER DEFINED VALUE LIST	**Code**	**22**
EXTENDED PARAMETER FOR VALUE ACQUISITION	**Code**	**112**
EXTENDED GET ONLINE VALUE	**Code**	**119**

Group **Recorder**

EXTENDED GET RECORDER RESULT DATA TYPES	**Code**	**149**
EXTENDED GET RECORDER RESULTS	**Code**	**146**

Group **Miscellaneous**

CASE SENSITIVE LABELS	**Code**	**61**

4.1.7 Command Sequences

Definition of default LUN = 0 in the case of unapplied commands ‘SELECT DESCRIPTION AND BINARY FILE’ or ‘DEFINE DESCRIPTION AND BINARY FILE’ (see command ‘DEFINE DESCRIPTION AND BINARY FILE’).

The AuSy can terminate an ASAP3 session with the command ‘EXIT’.

4.2 CHANGES WITH RESPECT TO VERSION 1.0

This version of the ASAP3 protocol differs from the **previous V1.0 to version 2.0** in the following points:

4.2.1 Version Mechanism

A version management mechanism is activated together with this version of the interface protocol. The following applies for the identification of protocol versions VX.Y:

X: Main protocol version number (1...255). A change of this dimension is necessary if the general data or telegram structure is modified, if the data frame of existing functions has changed, if the meaning of existing commands differs considerably from those of the previous version or if the command sequence to be respected has been modified. If a AuSy and an MC system with an interface implementation of the same main protocol version number meet, in principle it may be assumed that they can communicate with each other.

Y: Sub-version number (0...255). This dimension is changed if a new function was added to the interface definition. An AuSy and MC system with the same main protocol version number but different sub-version numbers can communicate with one another but the full range of functions may not be available.

4.2.2 Transmission

The BAUD rate indicated for data transmission must be adjustable on the AuSy as well as on the MC system. A BAUD rate of 9600 must be considered a minimum requirement.

4.2.3 Data Formats

1: Change in the length of the ‘NAME’ protocol data type from the current 8 ASCII characters to the ‘STRING’ data type with a variable length (max. length: ‘Implementation Dependent’).

2: Integration of an ‘Integer4’ data type to transmit large integral units.

4.2.4 Data Frames

1: To avoid ambiguities in the interpretation of error statistics in the answer telegram from the MC system to the AuSy, the MC system may be requested by the AuSy to re-transmit the last sent telegram. However, this does not initiate a more recent version of the basic AuSy telegram.

2: In addition to the answer data of the MC system defined for each AuSy command a general error answer data frame will be introduced. This data frame will always be sent to the AuSy by the MC system instead of the regular data frame whenever an error occurs during the processing of AuSy commands on the MC system, and contains in addition to an error code an error message in clear text.

4.2.5 Functions

1: ‘IDENTIFY’ telegram: After initialising the communication (‘INIT’ command) a mutual identification is initiated in which the name of the AuSy and MC system and of the relevant realised protocol version are exchanged. This information is necessary for compatibility checking and for the clear text output of user messages.

2: Introduction of a set of new telegrams making a possibly available recorder function of the application system usable for the test bed system.

3: An additional command to transmit a packet of online values has been introduced. In addition to the number of actual values contained in the packet it contains the global list of actual values. This modifies the answer telegram of the 'SWITCHING OFF LINE/ON LINE' command. No more data are transmitted with this command.

4: An additional telegram was introduced to activate the reset line of the command devices via the MC system from the AuSy.

5: A new command to influence the representation mode allows to transfer data from the MC system to the AuSy in the micro-controller model of the command devices.

6: In the data frame for the selection of the description and binary files ('SELECT DESCRIPTION-FILE AND BINARY-FILE') the destination (Emulator) is available as an additional parameter for which both file data must be valid.

7: In the data frame for the 'COPY BINARY FILE' command there is an additional source or target code: physical emulator board.

8: The interpretation of the 'PARAMETER FOR VALUE ACQUISITION' telegram was concretised. In future it will be possible to repeatedly use this telegram for a given ECU-(EP)PROM ('EMULATOR LUN'). The lists of dimensions to be recorded online contained in these telegrams are concatenated in their entry sequence whatever the indicated ('Emulator LUN') and thus produce a final list of on line values. A telegram with an 'empty' list ('Number of values'=0) will lead to a rejection of the global list built up to that moment whatever the indicated ('Emulator LUN')('CLEAR ON LINE VALUE LIST').

4.2.6 Status

$5656:

This MC system error status explicitly informs the AuSy of the non-availability of a selected function.

$FFFF:

An error has occurred during the execution of an AuSy command on the MC system. The answer telegram of the MC system to the AuSy contains information about the error cause/error location. (Error Code) as well as an error message in clear text.

5 DATA FORMATS

All data structures used for the exchange of data between the AuSy and MC system belong to one of 6 basic types:

Data type	Length	Comments
BYTE (0...255)	1 BYTE	---
WORD (0...65535)	2 BYTE	Motorola type
INTEGER 4 (signed)	4 BYTE	Motorola type
IEEE-REAL	4 BYTE	Motorola type
IEEE-REAL 8	8 BYTE	Motorola type
STRING	2n BYTE	see definition

The ‘STRING’ data type is a composite data type:

Length	Character 1	Character 2	...	Character x
1 WORD	1 BYTE	1 BYTE	...	1 BYTE

Length: Number of subsequent valid characters in Byte. The maximum allowed length is implementation dependent.
Character 1,2 ... Byte value in ASCII interpretation.
Character x: Filler; if the number of valid characters in the Byte (length) is odd, a filler byte will make the effective string-length L_{off} (in Byte) even again.

L_{off} = ‘Length’ + “Length’ MOD 2 + 2

The maximum length of names and labels used for communication depends on the latest valid definition of ASAP2.

Please note: PC platforms usually use INTEL data types.
The latter must be converted to the Motorola format (‘swap’).

In addition to these 6 real data types 3 logical data types may also be indicated. They are directly related to the conditions prevailing in modern engine electronics.

log. Data type	structure	Meaning
Map	field	Site oriented representation of a context z=f(x,y) or z=f(x); z(ij)=f[x(i),y(j)];z(i)=f[x(i)]; retrievable and modifiable via the MC system
Parameter	scalar	fixed value; retrievable and modifiable via the MC system
Actual value	scalar	Values cyclically measured or calculated by the ECU; retrievable but not modifiable via the MC system

The data type DATATYPE is an enumeration of type WORD and has the values:

DATATYPE (Word)	Data type
0	IEEE-Real
1	IEEE-Real 8
2	STRING

Definition of the IEEE Format (IEEE-REAL 32-Bit):

Sign	Biases Exponent					Significant						
s	e7	e6	...	e1	e0	b1	b2	b3	...	b21	b22	b23

31 23 0

Representation of real numbers: $(-1)^s * 2^E * b_{0\Delta} b_1 b_2 b_3 ... b_{23}$

s: 0 or 1
E: any integer between -127 and +128 (E = e - 127)
b_i: 0 or 1 (where b_0 = 1)

RealNumber $= (-1)^s \; * \; 2^{(-127) + \sum_{i=0}^{7} (e_i * 2^i)} \; * \; \sum_{i=0}^{23} \left(\frac{b_i}{2^i} \right)$ where b_0 = 1

Definition of the IEEE Format (IEEE-REAL 8, 64-Bit):

Sign	Biases Exponent					Significant						
s	e10	e9	...	e1	e0	b1	b2	b3	...	b50	b51	b52

63 52 0

Representation of real numbers: $(-1)^s * 2^E * b_{0\Delta} b_1 b_2 b_3 ... b_{52}$

s: 0 or 1
E: any integer between -1023 and +1024 (E = e - 1023)
b_i: 0 or 1 (where b_0 = 1)

RealNumber $= (-1)^s \; * \; 2^{(-1023) + \sum_{i=0}^{11} (e_i * 2^i)} \; * \; \sum_{i=0}^{52} \left(\frac{b_i}{2^i} \right)$ where b_0 = 1

Invalid Measurements

The smallest negative IEEE REAL value indicates invalid measurements to the AuSy.

A certain defined REAL value will be transmitted to the AuSy in the case of invalid measurements. Measured quantities are considered invalid if at the MC system the acquistion of measurements is not possible or the measurements are not available.

Definition of invalid measurements:

The related HEX values:

FF000000 h (IEEE-REAL, 32-Bit)

FFF0000000000000 h (IEEE-REAL 8, 64-Bit)

This corresponds to REAL value < -3.4028E+38 (32-Bit) and < -1.7E308 (64-Bit), respectively.

6 TELEGRAM STRUCTURE

The communication between the AuSy and MC system occurs via 2 defined telegrams with a fixed format:

AuSy to MC system
MC system to AuSy

General command structure for **'AuSy to MC system'** telegrams:

Length	Command code	Data	Check sum
1 WORD	1 WORD	n WORD	1 WORD

General command structure of the **'MC system to AuSy'** telegrams:

Length	Command code	Status	Data	Check sum
1 WORD	1 WORD	1 WORD	m WORD	1 WORD

Communication control and data exchange between the AuSy and MC system occur with these two telegrams.

For communication control there are 3 variants of these general telegrams, the command telegrams:

Repeat request to the MC system
Repeat request from the MC system
Acknowledgement message of the MC system

Repeat request to the MC system:

Length	Command code	Check sum
6	0	1 WORD

Repeat request from the MC system:

Length	Command code	Status	Check sum
8	0	$EEEE	1 WORD

Acknowledgement message from the MC system:

Length	Command code	Status	Check sum
8	1 WORD	$AAAA	1 WORD

The repeat request is intended to prompt the MC system or AuSy to re-transmit the last telegram sent ('AuSy to MC system' or 'MC system to AuSy') *('the last sent telegram was not correctly received or interpreted by the MC system/AuSy')*. A 'repeat request to the MC system' will not, however, result in a more recent version of the commands to the MC system in the basic 'AuSy to the MC system' telegram.
The acknowledgement message informs the AuSy of the correct reception of the 'AuSy to MC system' telegram ('*the last telegram was received and processing has started on the MC System').*

The individual telegram inputs have the following meaning:

Length: Length of the global telegram in Bytes (including length-WORDs, command code, data record and check sum; for the MC system answer ‘MC system to AuSy’ obviously the status is included). The length is always an even number. The minimum length for the ‘AuSy to MC system’ telegrams is of 6 Bytes, that of the ‘MC system to AuSy’ answer is 8 Bytes.

Command: Fixed number; the command code of the MC system answer ‘MC system to AuSy’ always corresponds with the original ‘AuSy to MC system’ telegram sent by the AuSy.

Status: This value serves to transfer error information in the ‘MC system to AuSy’ answer telegram.

Status	Meaning
$0000	Faultless execution of the last command
$1232	Faultless execution of the last command
$2343	Value $2343 indicates that the last sent telegram was not processed and that the AuSy must again set up communication, as interactively major changes have been made to the MC system configuration predefined by the AuSy. The successful execution of the ‘INIT’ command (Code 2) resets this status (see ‘User command “START”’ in the section on ‘some command sequences’).
$2344	Measuring data list has changed
$3454	A $3454 status informs the AuSy that the MC system is in simulation mode. The status is interpreted in the subsequent processing as an ‘0’ or ‘$1232’ status (faultless command execution).
$5656	This status informs the AuSy of the non-availability of the function of the last sent command.
$FFFF	An error occurred during the processing of the last AuSy command sent to the MC system. An error number and a detailed error message in clear text are sent by the MC system to the AuSy.
all others	reserved.
$AAAA	This status is considered as an acknowledgement of the last command by the AuSy. The final answer of the MC system occurs after the command has been processed (see telegram above).
$EEEE	In conjunction with command code 0 in the AuSy RETRY telegram, this status may be considered as a command repeat request (see telegram above).

Data: The length and structure of the data records - described further - depend on the actual command. They may differ for the ‘AuSy to MC system’ telegrams and for the ‘MC system to AuSy’ telegrams. A length 0 for the data record is allowed.

Checksum: Is obtained by adding all telegram WORD dimensions and contains the ‘Least Significant WORD’ of this sum.

7 COMMUNICATION SEQUENCE

A command sequence AuSy → MC system and MC system → AuSy is always introduced by the AuSy with a ‘AuSy to MC system’ telegram and is concluded by the MC system with the ‘MC system to AuSy’ telegram. Between them the command telegrams ‘Repeat request to MC system’, ‘Repeat request from the MC system’ or ‘Acknowledgement message of the MC system’ may be used. This results in the following diagram :

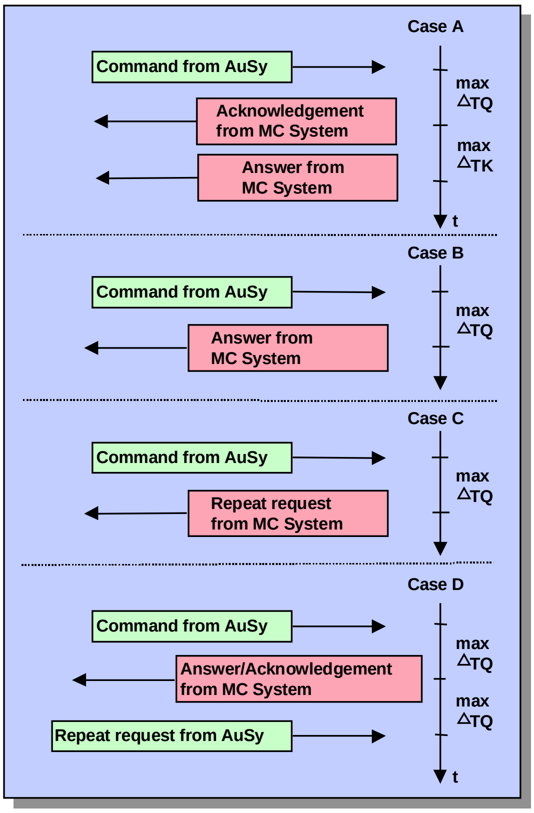


Figure 2

Following an AuSy command either an acknowledgement message, a repeat request or the final answer telegram must be sent by the MC system within a predetermined time range ΔTQ. If this time range is exceeded the line between AuSy and MC system will be considered as non-operational and corresponding error messages will be issued and reactions will occur on the AuSy.

If an AuSy command is answered by the acknowledgement message the final answer telegram must be sent within a command-dependent timeout ΔTK without a new prompt from the MC system to the AuSy.

8 DETAILED COMMANDS

Version V2.1 consists of the following command groups:

Note:

The new EXTENDED commands are not an official part of the standard version 2.1. They are rather supposed to be used for test implementations. The EXTENDED commands will be finally released in version 2.2. In this document, their special status is indicated by a shaded background.

Initialisation, Identification, Emergency	Code	Code/hex	page
EMERGENCY	1	0001 h	20
INIT	2	0002 h	20
IDENTIFY	20	0014 h	20
EXIT	50	0032 h	21

Configuration	Code	Code/hex	page
SELECT DESCRIPTION FILE AND BINARY FILE	3	0003 h	21
COPY BINARY FILE	4	0004 h	23
CHANGE BINARY FILE NAME	5	0005 h	24
DEFINE DESCRIPTION FILE AND BINARY FILE	30	001E h	22

Map Manipulation	Code	Code/hex	page
SELECT LOOK-UP TABLE	6	0006 h	24
PUT LOOK-UP TABLE	7	0007 h	25
GET LOOK-UP TABLE	8	0008 h	26
GET LOOK-UP TABLE VALUE	9	0009 h	26
INCREASE LOOK-UP TABLE	10	000A h	27
SET LOOK-UP TABLE	11	000B h	27

Extended Map Manipulation	Code	Code/hex	page
EXTENDED SELECT LOOK-UP TABLE	106	006A h	28
EXTENDED PUT LOOK-UP TABLE	107	006B h	28
EXTENDED GET LOOK-UP TABLE	108	006C h	29
EXTENDED GET LOOK-UP TABLE VALUE	109	006D h	29
EXTENDED INCREASE LOOK-UP TABLE	110	006E h	30
EXTENDED SET LOOK-UP TABLE	111	006F h	30

Measurement Data Recording	Code	Code/hex	page
PARAMETER FOR VALUE ACQUISITION	12	000C h	31
EXTENDED PARAMETER FOR VALUE ACQUISITION	112	0070 h	31
SWITCHING OFFLINE/ONLINE	13	000D h	32
GET ONLINE VALUE	19	0013 h	32
EXTENDED GET ONLINE VALUE	119	0077 h	34

GET USER DEFINED VALUE	21	0015 h	33
GET USER DEFINED VALUE LIST	22	0016 h	33

Parameter Manipulation	**Code**	**Code/hex**	**page**
GET PARAMETER	14	000E h	34
SET PARAMETER	15	000F h	35

Extended Parameter Manipulation	**Code**	**Code/hex**	**page**
EXTENDED GET PARAMETER	114	0072 h	35
EXTENDED SET PARAMETER	115	0073 h	35

Recorder	**Code**	**Code/hex**	**page**
DEFINE RECORDER PARAMETERS	41	0029 h	36
DEFINE TRIGGER CONDITION	42	002A h	37
ACTIVATE RECORDER	43	002B h	37
GET RECORDER STATUS	44	002C h	38
GET RECORDER RESULT HEADER	45	002D h	38
EXTENDED GET RECORDER RESULT DATA TYPES	149	0095 h	40
GET RECORDER RESULTS	46	002E h	39
EXTENDED GET RECORDER RESULTS	146	0092 h	40
SAVE RECORDER FILE	47	002F h	41
LOAD RECORDER FILE	48	0030 h	41

Miscellaneous	**Code**	**Code/hex**	**page**
SET GRAPHIC MODE	16	0010 h	42
RESET DEVICE	17	0011 h	42
SET FORMAT	18	0012 h	43
SET CASE SENSITIVE LABELS	61	003D h	44

8.1 GENERAL

The following data frames of the MC system describe answers to the individual commands that are related to the faultless execution on the MC system. If during the execution of the command on the MC system an error occurs, a fixed data frame will be transmitted to the AuSy for error analysis. It replaces the data frame for the faultless execution.

Data from the MC system:

Entry	**Data type**
Error code	WORD
Error text	STRING

8.2 INITIALISATION, IDENTIFICATION, EMERGENCY

EMERGENCY	**Code**	**1**

This command may be sent to the MC system during active communication in an emergency situation. ECU-specific emergency reactions may be initiated on the MC system. A definition of such a situation, the value allocated to the 'Event' and how it must be dealt with, depends on the actual realisation of the MC system.

Data to the MC system:

Entry	**Data type**
Event	WORD

Data from the MC system:

NO DATA

INIT	**Code**	**2**

This command is the first to be sent by the testing system and starts communication with the MC system. The MC system performs all the necessary initialisations. After this command the MC system is ready for REMOTE operation with the AuSy.

Data to the MC system:

NO DATA

Data from the MC system:

NO DATA

IDENTIFY	**Code**	**20**

After initialising the connection (Command 'INIT') a randomly attributable name and the current protocol version number of the communication partner must be exchanged with this command. If this command is acknowledged by an error message from the MC system, it may be assumed by the AuSy that a protocol version V1.x has been implemented on the MC system.
If, however, this command is not sent by the AuSy, then the MC system may assume that a protocol version V1.x has been implemented on the AuSy.

Data to the MC system:

Entry	**Data type**
Protocol version number	WORD
AuSy name	STRING

Data from the MC system:

Entry	**Data type**
Protocol version number	WORD
MC system name	STRING

For a protocol version VX.Y : ‘Protocol Version Number’ = 256*X+Y applies.

EXIT	**Code**	**50**

This command is the last one to be sent by the AuSy to release the ASAP3 interface of the MC system. This enables the MC system to free all allocated resources and to switch back to normal mode. After this command has been issued, the command INIT has to be sent again to start another ASAP3 session.

Data to the MC system:

NO DATA

Data from the MC system:

NO DATA

8.3 CONFIGURATION

SELECT DESCRIPTION FILE AND BINARY FILE	**Code**	**3**

In general for each ECU-(FLASH / EP)ROM there is a binary data file and a description file containing the structure information of the binary file and of the ECU-RAM structure. In the description file there are also links of map addresses, parameter addresses and actual value addresses with logical names and labels, e.g. the map SP_ADV (spark advance map) is linked to start address $06F0 or parameter MAX_N (maximum speed) is in address $780A or actual value AKT_N (actual engine speed) is in RAM address $0030, etc. As concerns the integration of a MC system in the AuSy it is recommended that these files - and hence the basic configuration - can be set by the higher-order AuSy.

This command may only be transmitted to the MC system after a successful execution of the ‘INIT’ command. A ‘Binary data file’ and a related ‘Description file’ are jointly linked to a logical Unit Number (EMULATOR LUN). Both files are local to the MC system. The path and file extension correspond with the default settings of the MC system.

With the indication of a destination number agreed by the AuSy and MC system the AuSy can define the valid location (multiple of a control device) for the allocation of the description file and binary file. This is required if a corresponding indication is not present in the description file. The destination number and the ‘Emulator LUN’ need not correspond.

Data for the MC system:

Entry	Data type
Description file name	STRING
Binary file name	STRING
Destination	WORD

Destination: 0...destination is automatically defined by the MC system; 1, 2, ... final allocation to destinations 1, 2,... by the AuSy.

Data from the MC system:

Entry	Data type
Emulator LUN	WORD

DEFINE DESCRIPTION FILE AND BINARY FILE	Code	30

This command can be used alternatively to the command ‘SELECT DESCRIPTION FILE AND BINARY FILE‘. With the parameter mode it is possible to proceed to control the MC system in an already existing configuration. For example with mode = 0 the long time for data download of to the ECU is no longer necessary.

Command ‘SELECT DESCRIPTION FILE AND BINARY FILE‘ and command ‘DEFINE DESCRIPTION FILE AND BINARY FILE‘ are optional. If none of these commands is used the default **LUN = 0** is taken.

Data to MC system:

Entry	Data type
Description file name	STRING
Program code (+Calibration data) file name	STRING
Calibration data file name	STRING
Destination	WORD
Mode	WORD

Description file name:

Name of description file, eg. name of ASAP2 file.

Program code (+Calibaration data) file name:

A blank string can be used here for Mode = 0 or Mode = 2. The Calibration data are only taken if the following ‘Calibration data file name‘ is a blank string.

Calibration data file name:

A blank string can be used here for Mode = 0. In the case of a blank string and a Mode > 0 the MC system takes the calibration data from the previously defined ‘Program code file name‘.

Destination:

0	destination is automatically defined by MC system
1	parallel EPROM emulation
2	CAN-Bus
3	K-Line

Note: MC system specific interfaces can be defined if the **Bit 15** of ‘Destination‘ is set to 1.

Mode:

0 No download to the memory of destination. Only a search for an already existing configuration of this type is done. The response from the MC system contains the data described below.

1 Download of program code and calibration data to the memory of destination. In general an engine stop is necessary which is not always of interest and therefore mode 2 was introduced.

2 Download of only calibration data to the memory of destination. This mode is important because modern ECUs do not have necessarily an emulation memory for the program code.

3 Down load of ECU program code only if the MC system regcognizes (eg CRC check) that the program code of the destination memory is different to the desired configuration.

Note: MC system specific modes can be defined if the **Bit 15** of ‘Mode‘ is set to 1.

Data from MC system:

Entry	Data type
Emulator LUN	WORD
Description file name	STRING
Program code (+Calibration data) file name	STRING
Calibration data file name	STRING

COPY BINARY FILE	**Code**	**4**

After an automatic move and modification of the ECU data via the AuSy – MC system link, it is recommended to save the modified binary data in a file on the hard disk.
This command must only be executed if the quantity ‘Emulator LUN’ has already been determined with the ‘SELECT DESCRIPTION FILE AND BINARY’ command. This command transfers the current binary file of the specified emulator from the specified source to the specified target, e.g. from the emulator board to a file on the hard disk.

Data for the MC system:

Entry	Data type
Target	WORD
Source	WORD
Emulator LUN	WORD

Data from the MC system:

NO DATA

Source, target: 1... EPROM, 2... file, 3... virtual emulator board, 4... physical emulator board.

The board from which the control device obtains its current data is considered as the physical emulator board . As opposed to this the virtual emulator board is a representation of the contents of the physical emulator board on the MC system without having to have the same contents as the latter at all times.
The minimum requirement is the realisation of the copy command with the 'virtual Emulator Board' as a source and 'File' as a target. This allows to store the binary file of the indicated emulator on the hard disk under its current name.

CHANGE BINARY FILE NAME	Code	5

This command is only sent by the AuSy after the successful definition of the 'Emulator LUN' by the 'SELECT DESCRIPTION FILE AND BINARY FILE' command. It is used to change the binary file name of the designated emulator in the prescribed manner. A subsequent 'COPY BINARY FILE' will store the binary file under this new name. This allows to save modified binary file contents without losing the original data.

Data for the MC system:

Entry	Data type
New Binary File Name	STRING
Emulator LUN	WORD

Data from the MC system:

NO DATA

8.4 MAP MANIPULATION

SELECT LOOK-UP TABLE	Code	6

This command is only sent by the AuSy after a successful definition of the 'Emulator LUN' by the 'SELECT DESCRIPTION FILE AND BINARY FILE' command. This command combines a specific map with the 'Emulator LUN' into an unambiguous 'map number'. Further basic information about the map dimensions (nx, ny) will be requested by the MC system. A function z=f(x,y) expands in the x direction as well as in the y direction, i.e. nx > 1 and ny >1. A function z=f(x), however, only expands in the x direction, i.e., nx > 1 and ny=1. Each map to be addressed by the AuSy must be notified to the MC system in this manner. Subsequently, each map shall only be addressed with the 'map number'. The maximum allowed dimensions for the maps depends on the implementation.

Data for the MC system:

Entry	Data type
Emulator LUN	WORD
Map name	STRING

Data from the MC system:

Entry	Data type
Map number	WORD

Y dimension (ny)	WORD
X dimension (nx)	WORD
Address (logging info)	WORD

PUT LOOK-UP TABLE	Code	7

This command is only sent after the ‘Map number’ has been defined with ‘SELECT LOOK-UP TABLE’. This command transmits the entire map content to the MC system. The map structure defined by means of the ‘SELECT LOOK-UP TABLE’ command remains unchanged. Depending on the MC system software this command may not only change the z body but also the axe definition (position of the sites).

Data for the MC system:

Entry	Data type
Map number	WORD
Map length	WORD
Y(1)	REAL
Y(2)	REAL
:	:
Y(ny)	REAL
X(1)	REAL
X(2)	REAL
:	:
X(nx)	REAL
Minimum Z(ij)	REAL
Maximum Z(ij)	REAL
Minimum increment	REAL
Z[X(1),Y(1)]	REAL
Z[X(2),Y(1)]	REAL
Z[X(3),Y(1)]	REAL
:	:
Z[X(2),Y(5)]	REAL
:	:

The same format is used for functions z=f(x). In this case the entry will be Y(1) followed by a random dummy value!

Data from the MC system:

NO DATA

The entry ‘Map length’ states the number of REAL values of the map (Length=ny + nx + (ny * nx) + 3; the maximum length is thus obtained for a maximum map definition of 20 x 20 by 20 + 20 +(20 * 20) + 3 =443 or for 32 x 32 maps by 32 + 32 + (32 * 32) + 3 = 1091) and must obviously correspond with the map dimensions of the answer telegram to command 6.
The entries ‘Minimum Z(ij)’ and ‘Maximum Z(ij)’ are global limits for the z body of the map, accordingly the ‘minimum increment’ entry is dependent upon the definition of the data type used in the ECU.
These 3 dimensions are defined by the MC system in the answer telegram to command 8 and only serve to inform the AuSy about these global map restrictions.

GET LOOK-UP TABLE	Code	8

This command is only sent after the ‘Map number’ has been defined with ‘SELECT LOOK-UP TABLE’ command. This command transmits the entire map content from the MC system to the AuSy. A copy of the map is thus available on the AuSy and can be archived or manipulated.

Data for the MC system:

Entry	Data type
Map number	WORD

Data from the MC system:

Entry	Data type
Map length	WORD
Y(1)	REAL
Y(2)	REAL
:	:
Y(ny)	REAL
X(1)	REAL
X(2)	REAL
:	:
X(nx)	REAL
Minimum Z(ij)	REAL
Maximum Z(ij)	REAL
Minimum increment	REAL
Z[X(1),Y(1)]	REAL
Z[X(2),Y(1)]	REAL
Z[X(3),Y(1)]	REAL
:	:
Z[X(2),Y(5)]	REAL
:	:

The same format is used for functions z=f(x). In this case the entry will be Y(1) followed by a random dummy value!

The entry ‘Map length’ states the number of REAL values of the map (Length=ny + nx + (ny * nx) + 3; the maximum length is thus obtained for a maximum map definition of 20 x 20 by 20 + 20 +(20 * 20) + 3 =443 or for 32 x 32 maps by 32 + 32 + (32 * 32) + 3 = 1091) and must obviously correspond with the map dimensions of the answer telegram to command 6. The entries ‘Minimum Z(ij)’ and ‘Maximum Z(ij)’ are global limits for the z body of the map, accordingly the ‘minimum increment’ entry is dependent upon the definition of the data type used in the ECU.

GET LOOK-UP TABLE VALUE	Code	9

This command is only sent after the ‘Map number’ has been defined with ‘SELECT LOOK-UP TABLE’ command. This command makes the MC system transmit a single map value defined by the indication of the site location to the AuSy.

Data for the MC system:

Entry	Data type
Map field number	WORD
Y index	WORD
X index	WORD

Data from the MC system:

Entry	Data type
Value	REAL

INCREASE LOOK-UP TABLE	Code	10

This command is only sent after the ‘Map number’ has been defined with ‘SELECT LOOK-UP TABLE’ command. This command makes the MC system add an offset to the specified map. The relative form of the map remains unchanged in all map areas meeting the restrictions of the z body. All other areas receive the corresponding limit value (Maximum Z(ij), Minimum Z(ij). The map area to be changed is indicated by the start index and the delta range.

Data for the MC system:

Entry	Data type
Map number	WORD
Y index	WORD
X index	WORD
Y Delta	WORD
X Delta	WORD
Offset	REAL

Data from the MC system:

NO DATA

SET LOOK-UP TABLE	Code	11

This command is only sent after the ‘Map number’ has been defined with ‘SELECT LOOK-UP TABLE’ command. This command makes the MC system set the contents of the selected map in the specific range to a constant value. In this range the map will be brought to a horizontal level.

Data for the MC system:

Entry	Data type
Map number	WORD
Y index	WORD
X index	WORD
Y Delta	WORD
X Delta	WORD
Value	REAL

Data from the MC system:

NO DATA

8.4.1 Extended Map Manipulation

EXTENDED SELECT LOOK-UP TABLE	Code	106

This command is like ‘SELECT LOOK-UP TABLE’ (Code 6), but it adds a coding scheme for the different parts of the lookup table. The fields 'y data type', 'x data type' and 'z data type' contain enumerations, which indicate the type of the data being transmitted in physical model (phys) or in controller model (controller).

Data for the MC system:

Entry	Data type
Emulator LUN	WORD
Map name	STRING

Data from the MC system:

Entry	Data type
Map number	WORD
y dimension (ny)	WORD
x dimension (nx)	WORD
y data type (phys)	DATATYPE
x data type (phys)	DATATYPE
z data type (phys)	DATATYPE
y data type (controller)	DATATYPE
x data type (controller)	DATATYPE
z data type (controller)	DATATYPE
Address (logging info)	INTEGER 4

EXTENDED PUT LOOK-UP TABLE	Code	107

This command is like ‘PUT LOOK-UP TABLE’ (Code 7). The data types for the different parts of the command have to be the same like returned by command 106 depending on the current setting for physical/controller model.

Data for the MC system:

Entry	Data type
Map number	WORD
Map length	WORD
Y(1)	y data type
Y(2)	y data type
:	:
Y(ny)	y data type
X(1)	x data type
X(2)	x data type
:	:
X(nx)	x data type

Minimum Z(ij)	z data type
Maximum Z(ij)	z data type
Minimum increment	z data type
Z[X(1),Y(1)]	z data type
Z[X(2),Y(1)]	z data type
Z[X(3),Y(1)]	z data type
:	:
Z[X(2),Y(5)]	z data type
:	:

Data from the MC system:

NO DATA

EXTENDED GET LOOK-UP TABLE	Code	108

This command is like ‘GET LOOK-UP TABLE’ (Code 8). It returns the data of the three parts of the answer using the data type returned by command 106 depending on the current setting for physical/controller model.

Data for the MC system:

Entry	Data type
Map number	WORD

Data from the MC system:

Entry	Data type
Map length	WORD
Y(1)	y data type
Y(2)	y data type
:	:
Y(ny)	y data type
X(1)	x data type
X(2)	x data type
:	:
X(nx)	x data type
Minimum Z(ij)	z data type
Maximum Z(ij)	z data type
Minimum increment	z data type
Z[X(1),Y(1)]	z data type
Z[X(2),Y(1)]	z data type
Z[X(3),Y(1)]	z data type
:	:
Z[X(2),Y(5)]	z data type
:	:

EXTENDED GET LOOK-UP TABLE VALUE	Code	109

This command is like ‘GET LOOK-UP TABLE VALUE’ (Code 9). The data type of the returned value is specified by the answer of command 106 ('z data type') depending on the current setting for physical/controller model.

Data for the MC system:

Entry	Data type
Map number	WORD
Y index	WORD
X index	WORD

Data from the MC system:

Entry	Data type
Value	z data type

EXTENDED INCREASE LOOK-UP TABLE	Code	110

This command is like ‘INCREASE LOOK-UP TABLE’ (Code 10). The data type of the value has to be the same like returned by comand 106 ('z data type') depending on the current setting for physical/controller model. For the data type STRING, this command cannot be executed!

Data for the MC system:

Entry	Data type
Map number	WORD
Y index	WORD
X index	WORD
Y Delta	WORD
X Delta	WORD
Offset	z data type

Data from the MC system:

NO DATA

EXTENDED SET LOOK-UP TABLE	Code	111

This command is like ‘SET LOOK-UP TABLE’ (Code 11). The data type of the value has to be the same like returned by comand 106 ('z data type') depending on the current setting for physical/controller model.

Data for the MC system:

Entry	Data type
Map number	WORD
Y index	WORD
X index	WORD
Y Delta	WORD
X Delta	WORD
Value	z data type

Data from the MC system:

NO DATA

8.5 MEASUREMENT DATA AQUISTION

PARAMETER FOR VALUE ACQUISITION	Code	12

This command is only sent by the AuSy after the 'Emulator LUN' has been defined by the 'SELECT DESCRIPTION FILE AND BINARY FILE'. This command informs the MC system of the actual values, their sequence and the acquisition rate at which they must be transmitted to the AuSy. Scanning periods (scanning frequencies) in the 10000 ms (0,1 Hz) to 500 ms (2 Hz) range are provided. This command can be repeated a number of times for each 'Emulator LUN'. In this case the transmitted lists are concatenated in their access sequence, whatever the indicated 'Emulator LUN'. If a number of emulators are available from the MC system or if the command was repeatedly sent, the entry of the scanning rate of the last telegram will apply.

If the AuSy does not require any actual values from the MC system an empty list with 'Number of values'=0 may also be sent to the MC system. An empty list furthermore entails a rejection of the actual value list compiled until that moment, whatever the indicated 'Emulator LUN' ('CLEAR ON LINE VALUE LIST').

If a value will be defined more than once, it has to be transmitted in respect to the definition order. An error message will be generated only, if a label name is wrong. Hint: If possible, avoid this option in respect of interface performance!

Data for the MC system:

Entry	Data type
Emulator LUN	WORD
Scanning time (ms)	WORD
Number of values	WORD
Name of the 1st value	STRING
Name of the 2nd value	STRING
:	:

Data from the MC system:

NO DATA

EXTENDED PARAMETER FOR VALUE ACQUISITION	Code	112

This command is like PARAMETER FOR VALUE ACQUISTION (Code 12). The answer from the MC system contains the data type coding of the according value.

Data for the MC system:

Entry	Data type
Emulator LUN	WORD
Scanning time (ms)	WORD
Number of values	WORD
Name of the 1st value	STRING
Name of the 2nd value	STRING
:	:

Data from the MC system:

Entry	Data type
Data type of the 1st actual value (physical)	DATATYPE
Data type of the 1st actual value (controller)	DATATYPE
Data type of the 2nd actual value (physical)	DATATYPE
Data type of the 2nd actual value (controller)	DATATYPE
:	;

SWITCHING OFFLINE/ONLINE	Code	13

If any actual values are to be collected this command will only be sent after the command ‘PARAMETER FOR VALUE ACQUISITION’ has been processed.
With the first execution of this command with Mode=1 the MC system will subsequently also transfer all current and future modifications of the map and parameters on the MC system to the motor ECU (transparent operation). Moreover, from then onwards the AuSy may request data packets with lists of the current actual values (command ‘GET ONLINE VALUE’). This transparent type of operation is again de-activated with Mode=0 and all subsequent map and parameter changes will again only occur locally on the MC system. At the same time, the MC system is informed of the setting of the cyclical actual value acquisition.

Data for the MC system:

Entry	Data type
Mode	WORD

Mode: O... OFF-LINE, 1...ON-LINE

Data from the MC system:

NO DATA

GET ONLINE VALUE	Code	19

In the event actual values are to be acquired this command will only be sent after the ‘SWITCHING OFFLINE/ONLINE’ command has been processed with Mode=1.
This command prompts the MC system to transfer to the AuSy a packet with the current actual values (in accordance with the list definition of the ‘PARAMETER FOR VALUE ACQUISITION’ command).

Data for the MC system:

NO DATA

Data from the MC system:

Entry	Data type
Number of actual values	WORLD
Value of the 1st actual value	REAL
Value of the 2nd actual value	REAL
:	:

GET USER DEFINED VALUE	**Code**	**21**

This command corresponds to the command ‘GET ONLINE VALUE‘. But in this case only those quantities are transferred to the AuSy which the user has selected by hand at the MC system (no additional quantities that have been generated by the MC system !).
This command can be only applied after previous successful executed commands ‘INIT‘ and ‘IDENTIFY‘ at the MC system.
A local change in the list of measurements at the MC system will result to a status $2344 in the answer telegram from the MC system to AuSy. This error message will be set back from the newly executed command ‘GET USER DEFINED VALUE LIST‘ and with following commands ‘GET USER DEFINED VALUES‘ the measurements of the changed list can be requested from the MC system.

Data for the MC system:

NO DATA

Data from the MC system:

Entry	Datentyp
Number of actual values	WORD
Value of the 1st actual value	REAL
Value of the 2nd actual value	REAL
:	:

GET USER DEFINED VALUE LIST	**Code**	**22**

This command transmits the names of the actual values which have been created by hand at the MC system. This command can be applied only after previous successfully executed commands ‘INIT‘ and ‘IDENTIFY‘ at the MC system.
In the case of not applied commands ‘SELECT DESCRIPTION FILE AND BINARY FILE‘ or ‘DEFINE DESCRIPTION FILE AND BINARY FILE‘ the MC system transmits **LUN = 0** to the AuSy. Furthermore, for MC system devices for which the LUNs cannot determined properly the **LUN = $FFFF** is send back to AuSy (eg. special measurement equipment or multi ECU).

Data for the MC system:

NO DATA

Data from the MC system:

Entry	Data type

Number of actual values	WORD
Emulator LUN of the 1st value	WORD
Name of the 1st value	STRING
Emulator LUN of the 2nd value	WORD
Name of the 2nd value	STRING
:	:

EXTENDED GET ONLINE VALUE	**Code**	**119**

This command is like ‘GET ON LINE VALUE’ (Code 19). Every value is returned with data data type returned by the command ‘EXTENDED PARAMETER FOR VALUE ACQUISTION’ (Code 112).

Data for the MC system:

NO DATA

Data from the MC system:

Entry	Data type
Number of actual values	WORD
Value of the 1st actual value	data type
Value of the 2nd actual value	data type
:	:

8.6 PARAMETER MANIPULATION

GET PARAMETER	**Code**	**14**

This command is only sent by the AuSy after the ‘Emulator LUN’ has been defined by the ‘SELECT DESCRIPTION FILE AND BINARY FILE COMMAND’. This command transmits the specified parameter from the MC system to the AuSy.

Data to the MC system:

Entry	Data type
Emulator LUN	WORD
Parameter name	STRING

Data from the MC system:

Entry	Data type
Value	REAL
Minimum value	REAL
Maximum value	REAL
Minimum increment	REAL

The entries ‘Minimum value’ and Maximum value’ are global limits for this parameter. The entry ‘Minimum increment’ is dependent on the resolution of the data type used for this parameter in the ECU.

SET PARAMETER	Code	15

This command is only sent by the AuSy after the ‘Emulator LUN’ has been defined by the ‘SELECT DESCRIPTION FILE AND BINARY FILE COMMAND’. This command transmits the specified parameter with the defined value from the AuSy to the MC system.

Data for the MC system:

Entry	Data type
Emulator LUN	WORD
Parameter name	STRING
Value	REAL

Data from the MC system:

NO DATA

8.6.1 Extended Parameter Manipulation

EXTENDED GET PARAMETER	Code	114

This command is like GET PARAMETER (Code 14). It additionally returns the data type to be used with this parameter and returns the values using the given data type.

Data to the MC system:

Entry	Data type
Emulator LUN	WORD
Parameter name	STRING

Data from the MC system:

Entry	Data type
Data type	DATATYPE
Value	data type
Minimum value	data type
Maximum value	data type
Minimum increment	data type

The entries 'Minimum value', 'Maximum value' and 'Minimum increment' are not defined when 'Data type' referes to STRING.

EXTENDED SET PARAMETER	Code	115

This command is like SET PARAMETER (Code 15). The AuSy additionally specifies the data type of the value.

Data for the MC system:

Entry	Data type
Emulator LUN	WORD
Parameter name	STRING
Data type	DATATYPE
Value	data type

Data from the MC system:

NO DATA

8.7 RECORDER

DEFINE RECORDER PARAMETERS	Code	41

This command is only sent by the AuSy after the ‘Emulator LUN’ has been defined by the ‘SELECT DESCRIPTION FILE AND BINARY FILE COMMAND’. This command defines the necessary parameters for the recorder function on the MC system.
This command can be repeated for each ‘Emulator LUN’. In this case the transmitted lists of dimensions to be included are concatenated in their access sequence, whatever the indicated ‘Emulator LUN’. If a number of emulators are available from the MC system or if the command was repeatedly sent, the entries for the ‘Recording type’, ‘Scanning time’ and ‘Divider factor’ of the last telegram will apply. An empty list furthermore entails a rejection of the list of dimensions to be recorded compiled until that moment, whatever the indicated ‘Emulator LUN’ (‘CLEAR RECORDER LIST’).

Data to the MC system:

Entry	Data type
Emulator LUN	WORD
Recording type	WORD
Scanning time (ms)	REAL
Divider factor	WORD
Number of values	WORD
Name of 1st value	STRING
Name of 2nd value	STRING
:	:

Recording type: 0... equidistant in time; 1... synchronous to the system interval time.

Data from the MC system:

NO DATA

In the event of time synchronous recording (Recording type=0) the ‘scanning time’ value will apply as time between two subsequent scannings. If the recording is synchronous to the system interval time of the control device that time interval between two scannings may change. Often the system interval time is linked to the position of the crankshaft and is thus crank angle synchronous. In this case the division factor indicates the number of scannings per system interval time.

DEFINE TRIGGER CONDITION	Code	42

This command is only sent by the AuSy after the Recorder parameter has been determined by the ‘DEFINE RECORDER PARAMETERS’. It includes as transition parameters the conditions for starting and stopping the recorder. These conditions are to be considered as logical functions capable of supplying the answer ‘TRUE’ or ‘FALSE’. The structure and type of the allowed nodes are implementation dependent. For repeated transmission the settings of the last sent telegram apply. If no trigger condition is defined for the Start or Stop (empty string), starting and stopping will occur manually, i.e. the maximum sample number will generate a stop of the recorder.

‘Start delay’ and ‘Stop delay’ generate a shift in the starting or stopping of the recording of the indicated samples. The ‘max. sample number’ is the measure for the max. global length of the recording. A negative value for the start delay is possible in view of the possibly available Post Mortem operating mode of the MC system recorder.

Data to the MC system:

Entry	Data type
Trigger START	STRING
Trigger STOP	STRING
Max. sample number	INTEGER4
Start delay	INTEGER4
Stop delay	INTEGER4

Data from the MC system:

NO DATA

ACTIVATE RECORDER	Code	43

This command is only sent by the AuSy after the trigger conditions have been defined in the ‘DEFINE TRIGGER CONDITION’ command. The sending of this command starts the recording, activates the defined conditions for starting the recorder or stops a possibly ongoing recording process.

Data to the MC system:

Entry	Data type
Mode	WORD

Mode = 0 ... recorder stop (independent of ‘Trigger STOP’); Mode = 1 ... activated recorder; Mode = 2 ... start recorder start (‘independent of ‘Trigger START’)

Data from the MC system:

NO DATA

GET RECORDER STATUS	Code	44

This command allows to query the current status of the ongoing recording and the number of already collected samples using the command 'ACTIVATE RECORDER' once the recorder has been activated. Moreover, information on stopping the recorder is given in a clear text message.

Data to the MC system:

NO DATA

Data from the MC system:

Entry	Data type
Recorder Status	WORD
Current Samples	INTEGER4
Stop Condition	WORD
Stop Information	STRING

Recorder status:

'Recorder on':	1
'Recorder off':	2
'Recorder ready':	3

Stop condition:

'Manual':	1
'Trigger Stop':	2
'End of Recording':	3
'Stop due to error':	256

GET RECORDER RESULT HEADER	Code	45

This command allows to transmit general information via the recording from the MC system to the AuSy upon completion of the current recording (the last valid recording following the 'INIT' command) or after the successful execution of the 'LOAD RECORDER FILE' command.

Data to the MC system:

NO DATA

Data from the MC system:

Entry	Data type
Start time (Trigger Start)[s]	STRING
Recording type	WORD
Scanning time [ms]	REAL
Divider Factor	WORD
Number of samples	INETGER4

Start Delay	INTEGER4
Stop Delay	INTEGER4
Lost Samples	INTEGER4
Max. Phase Errors	INTEGER4

The ‘Start Time’ dimension defines the time at which the trigger condition occurs for the start or the time of the manual start. It is given as the absolute time of day of the MC system and at least corresponds with the first 14 positions of the following format:

Year	Month	Day	Hour	Minute	Second	s/100	...
‘93’	‘11’	‘02’	‘18’	‘58’	‘45’	‘79’	...

The length of the string is thus of at least 14. More accurate resolutions are possible by adding pairs of additional characters (s/10000, s/1000000,...).

The ‘Recording Type’, the ‘Scanning Time’ and the ‘Divider Factor’ correspond with the values obtained with the ‘DEFINE RECORDER PARAMETERS’ command.

The ‘Number of Samples’ dimension gives the global number of samples including possibly lost samples. The ‘Start Delay’ and ‘Stop Delay’ values correspond with the values of the ‘DEFINE TRIGGER CONDITION’ command.

‘Lost Samples’ and ‘max. Phase Errors’ must be considered as additional information for the quality of the recording. A single sample contains the values for all the dimensions defined with the ‘DEFINE RECORDER PARAMETER’ telegram at the current scanning time. If the ‘Scanning Time’ or the ‘Divider Factor’ was unfavourably selected with the ‘DEFINE RECORDER PARAMETER’ telegram, it is possible that not all samples can be taken on time. They will then be lost or be undefined thus leaving blanks. The number of lost samples in the longest concatenated blank is the ‘max. phase error’.

GET RECORDER RESULTS	Code	46

This command allows to transfer the recording result by packets from the MC system to the AuSy at the end of the current recording (the last valid recording since the ‘INIT’ command) or after the successful execution of the ‘LOAD RECORDER FILE ‘ command. The data packets may be requested individually per ‘data packet number’.
Each data packet contains the sample defined with the ‘data packet number’. The sequence of the values in the data packet corresponds with the list compiled with the ‘DEFINE RECORDER PARAMETER’ telegram or indicated with the ‘LOAD RECORDER FILE’ command. The number of values in the data packet is transmitted with the dimension ‘Number values m’. If this value is indicated with ‘0’ the scanning on the MC system did not occur correctly, i.e. this is a lost sample and the values in the list have no significance.

Data to the MC system:

Entry	Data type
Data Packet Number n	INTEGER4

Data from the MC system:

Entry	Data type
Data Packet Number n	INTEGER4

Number of values m	WORD
Value 1 in sample n	REAL
Value 2 in sample n	REAL
Value 3 in sample n	REAL
:	:
Value m in sample n	REAL

EXTENDED GET RECORDER RESULT DATA TYPES	Code	149

This command is to be issued before the command EXTENDED GET RECORDER RESULTS (Code 146) to retrieve the data type coding of the variables.

Data to the MC system:

NO DATA

Data from the MC system:

Entry	Data type
Number of values m	WORD
Data type of value 1	DATATYPE
Data type of value 2	DATATYPE
Data type of value 3	DATATYPE
:	:
Data type of value m	DATATYPE

EXTENDED GET RECORDER RESULTS	Code	146

This command is like GET RECORDER RESULTS (Code 46). Every value is returned using the data type returned by the command EXTENDED GET RECORDER RESULT DATA TYPES (Code 149).
In addition to Command 46, this command enables the AuSy to request more than one sample (Data Packet) at a time. The MC returnes the actually returned number of samples, which is less or equal to the requested number of samples.

Data to the MC system:

Entry	Data type
Data packet number n	INTEGER4
Number of requested data packet	INTEGER4

Data from the MC system:

Entry	Data type
Data packet number n	INTEGER4
Number of actually returned data packets x	INTEGER4
Number of values m	WORD
Value 1 in sample n	data type
Value 2 in sample n	data type

Value 3 in sample n	data type
:	:
Value m in sample n	
:	:
Value 1 in sample n +x-1	data type
Value 2 in sample n +x-1	data type
Value 3 in sample n +x-1	data type
:	:
Value m in sample n+x-1	data type

SAVE RECORDER FILE	**Code**	**47**

This command allows to save the result of a recorder file locally on the MC system under an attributable file name. The ‘File Name ‘ dimension is based on a default directory and a default extension.

Data to the MC system:

Entry	Data type
File Name	STRING

Data from the MC system:

NO DATA

LOAD RECORDER FILE	**Code**	**48**

This command can only be transmitted by the AuSy to the MC system after initialisation by the ‘INIT’ telegram. This will result in a local record file stored on the MC system and specified by the ‘File Name’ entry being loaded by the MC system. The ‘File Name’ dimension is based on a Default Directory and a Default Extension. After this loading process the commands ‘GET RECORDER RESULT HEADER’ and ‘GET RECORDER RESULT’ will be executed as if the file process had been prompted on the MC system by the ‘DEFINE RECORDER PARAMETERS’, ‘DEFINE TRIGGER CONDITION’ and ‘ACTIVATE RECORDER’ commands from the AuSy.

The dimensions and sequence available for a transfer to the AuSy can be drawn from the list of ‘Number of Values’ length sent by the MC system to the AuSy.

Data to the MC system:

Entry	Data type
File Name	STRING

Data from the MC system:

Entry	Data type
Number of Values	WORD
Number of 1st value	STRING

Number of 2nd value	STRING
:	:

8.8 MISCELLANEOUS

SET GRAPHIC MODE	Code	16

This command can only be transmitted after the AuSY ↔ MC system connection has been initialised. It is thus also possible to make MC system resources available for more important system tasks (communication with the ECU,...) by deactivating a possible local visualisation.

Data to the MC system:

Entry	Data type
Mode	WORD

Mode = 0 ... visualisation is deactivated; Mode = 1... visualisation is activated.

Data from the MC system:

NO DATA

RESET DEVICE	Code	17

This command causes the MC system to activate the reset line of the control device belonging to the indicated ‘Emulator LUN’. It can only be used after the ‘Emulator LUN’ quantity has been defined.
After the execution of the command on the MC system the latter will be in the same condition as before the execution of this command with respect to the link to the AuSy and control device.

Data to the MC system:

Entry	Data type
Emulator LUN	WORD

Emulator LUN = 0; all the control devices belonging to the selected ‘Emulator LUNs’ are reset.

Data from the MC system:

NO DATA

SET FORMAT	Code	18

In addition to the physical representation of the control device dimensions (maps, parameters, actual values) a control device (controller) like representation of these data is desired on the AuSy. This representation can be realised by temporarily replacing a random conversion formula ‘Controller value x’ → ‘Physical value y’ (e.g. y_p = k0*(1-exp.(-k1*x))) MC system internally by the special representation y_p = x. An internal control device temperature value T =-234 is represented in the physical model as T_p = -17.25 °C and in the controller model as T_c = -234.0
The representation type is made individually switchable for the three logical data types ‘map’, ‘parameter’ and ‘actual value’. All the dimensions required for the representation of a map, parameter,.. are affected by this switch over.

(e.g. Y(1), Y(2), ... minimum Z(ij),... minimum increment, ... Z[X(2), Y(5)],... given a map) and also all the selected data type related commands (e.g. ‘SET LOOK-UP TABLE’, ‘GET LOOK-UP TABLE’,... for maps, ‘GET ON LINE VALUES’ for On line values etc.). The data frames allocated to the commands are not affected by this model switch-over (the relevant dimensions are still transmitted in IEEE-REAL format).

A more recent transfer of the command with a modified model switch allows to again change the representation type. The physical model is the default setting. For the log. data type ‘Actual value’ a switch over of the model is only possible in ‘Off line’ condition or with an inactive recorder.

An additional value for Model is defined. This value is only valid for the new commands introduced by version 2.1. The ‘EXTENDED’ commands can carry besides 4-bytes REAL values also 8-bytes real values or strings, depending on data type and conversion. The value 0 for model means to accept or transfer the data after performing the conversion defined in the description (ASAP2) file; the outcome can be a number or a string. The value 1 always forces to transfer the raw (controller) value. The value 2 means, that all values are transmitted in physical model except these variables/values where a string conversion is defined; these values will be transmitted in controller model.

Data to the MC system:

Entry	Data type
Log. data type	WORD
Model	WORD

log. data type: 0... all, 1... map, 2... parameter, 3...actual values

model: 0... mixed model (STRING types in controller model, all other types in physical model), 1... controller model, 2... physical model.

Data from the MC system:

NO DATA

If, for example, out of 6 maps to be processed the 4th is present in the controller model on the AuSy, the command ‘SET FORMAT’ allows to switch the representation type for the maps to ‘controller model’. The command ‘GET LOOK-UP TABLE’ transfers the 4th map to the AuSy and a more recent transmission of the ‘SET FORMAT’ command allows to switch back to the physical model to process the other maps.

SET CASE SENSITIVE LABELS	Code	61

This command can be send after the command ‘IDENTIFY‘ to give the MC system the message to process the label names in the following session as case sensitive.
Case insensitive processing is the default.

Data to the MC system:

NO DATA

Data from the MC system:

NO DATA

9 COMMAND SEQUENCES

All commands are executed using the software handshake (i.e. before a command can be executed the previous one must be completed) and be monitored with an individual timeout.

Please note:
All commands were initiated by the AuSy (AuSy = Master, MC system = Slave)

Basic Sequence

⇒ The MC system periodically checks the reception buffer of the serial line for an incoming AuSy command.

⇒ The AuSy sends a command and awaits an answer from the MC system. If this answer does not reach the AuSy within a command dependent timeout, it is assumed that the MC system is inactive.

⇒ The MC system reads the command, performs all the necessary actions and sends the answer containing the status information back to the AuSy.

⇒ The AuSy receives this answer and examines all incoming data.

9.1 EXAMPLES

Example 1: User command ‘START’ on the AuSy:

The AuSy sends the ‘INIT’ user command to the MC system

Entry	Value
Length	6
Code	2
Check sum	8

The MC system reads this message from the receive buffer and performs all the necessary actions to prepare the ‘Remote Control’ operation.

The answer of the MC system is transmitted to the AuSy.

Entry	Value
Length	8
Code	2
Status (ok)	0
Check sum	10

The AuSy reads this answer and recognises that the MC system is ready for further commands.

Subsequently, the communication participants and their implemented protocol versions identify themselves (version v2.1=256*2+1):

Entry	Value
Length	16

Code	31
Protocol version	513
Name AuSy length Name AuSy character	5 ‘AuSyx’
Check sum	xxxxx

The MC system name and its protocol version number are transmitted to the AuSy

Entry	**Value**
Length	20
Code	31
Status (ok)	0
Protocol version	512
Name MC system length	7
Name MC system character	‘MCD_xyz’
Check sum	xxxxx

The AuSy reads this answer and recognises that the MC system is based on protocol version V2.0. The basic communication capacity is thus ensured.

The command ‘SWITCHING OFFLINE/ONLINE’ with mode = 0 is transmitted to the MC system. Modifications of maps and parameters are henceforth only effective locally on the MC system.

Entry	**Value**
Length	8
Code	13
Mode	0
Check sum	21

The MC system answers will be:

Entry	**Value**
Length	8
Code	13
Status	0
Check sum	21

Subsequently, the ‘SELECT DESCRIPTION FILE AND BINARY FILE’ command is sent by the AuSy. This command can be repeatedly transmitted if a number of controllers in a engine electronic device can be operated with their own (EP)ROM and RAM contents. These units are kept separate because of their different ‘Emulator LUNs’.

Entry	**Value**
Length	26
Code	3
Description file name:Length	8
Description file name:Character	‘FORM_TST’
Binary file name:Length	8
Binary file name:Character	‘DATA_TST’
Destination	0

Check sum	xxxxx

The MC system returns the related ‘Emulator LUN’.

Entry	Value
Length	10
Code	3
Status	0
Emulator LUN	1
Check sum	14

The ‘PARAMETERS FOR VALUE ACQUISITION’ command is then transmitted. This command can be transmitted to the MC system for each emulator separately.

Entry	Value
Length	yyyyy
Code	12
Emulator LUN	1
Scanning rate [ms]	1000
Number of values	15
Name 1st value:Length	5
Name 1st value:Character	SPARK
Name 2nd value:Length	9
Name 2nd value:Character	‘ENGINE_SP’
:	:
Check sum	xxxxx

The MC system accepts these entries.

Entry	Value
Length	8
Code	12
Status	0
Check sum	20

For each defined parameter the ‘GET PARAMETER’ command is now processed. They are thus first available on the AuSy.

Entry	Value
Length	16
Code	14
Emulator LUN	1
Parameter name:Length	6
Parameter name:Character	‘P IDLE’
Check sum	xxxxx

The corresponding quantities of the requested parameter are transmitted by the MC sytem.

Entry	Value

Length	24
Code	14
Status	0
Value	1.23
Minimum value	0.00
Maximum value	2.55
Minimum increment	0.01
Check sum	xxxxx

The following ‘SELECT LOOKUP TABLE’ command identifies all maps to be processed. The ‘SELECT LOOKUP TABLE’ command is issued for each map. The map number allocated by the MC system to the map must be unambiguous.

Entry	**Value**
Length	18
Code	6
Emulator LUN	1
Map name:Length	7
Map name:Character	‘IT BASE’
Check sum	xxxxxx

The AuSy receives the basic data of the map as an answer from the MC system:

Entry	**Value**
Length	16
Code	6
Status	0
Map number	1
y dimension (ny)	3
x dimension (nx)	3
Address (logging info)	1234
Checksum	xxxxx

Subsequently, all selected maps are transferred to the AuSy with the ‘GET LOOKUP TABLE’ command, where they are stored as a backup copy.

Entry	**Value**
Length	8
Code	8
Map number	1
Check sum	17

For each requested map the global map record is transmitted.

Entry	**Value**
Length	82
Code	8
Status	0
Map length	18
Y(1)	0.00

:	:
Y(3)	5.00
X(1)	0.00
:	:
X(3)	2.00
:	:
Check sum	xxxxx

Example 2: 'ON LINE' user command on the AuSy:
The AuSy transmits the command 'SWITCHING OFFLINE/ONLINE' to the MC system with mode = 1. Subsequently, the 'GET ON LINE VALUE' command will be periodically executed with the defined scanning rate to transmit the actual values from the MC system to the AuSy. Obviously, other commands can be executed between two consecutive packet requests. However, it is possible that for a short while the scanning interval does not correspond with the defined scanning rate.
After this command all the modification interventions on the maps and parameters will also be effective in the engine electronic device.

Entry	**Value**
Length	8
Code	13
Mode	1
Check sum	22

After a positive MC system answer the AuSy will collect the first data packet from the MC system with the 'GET ON LINE VALUE' command.

Entry	**Value**
Length	6
Code	19
Status	0
Check sum	25

In the answer the MC system transmits a packet with actual values in accordance with the definition in the 'PARAMETERS FOR VALUE ACQUISITION' command.

Entry	**Value**
Length	68
Code	19
Status	0
Number of values	15
SPARK	20.9
ENGINE-SP	2509.0
:	:
Check sum	xxxxx

Example 3: ‘OFFLINE’ user command on the AuSy:

The AuSy command ‘SWITCHING OFFLINE/ONLINE’ is transmitted to the MC system using mode = 0. Subsequently, all further operations in the maps or parameters are only locally effective on the MC system and no cyclical querying of the current actual values occurs.

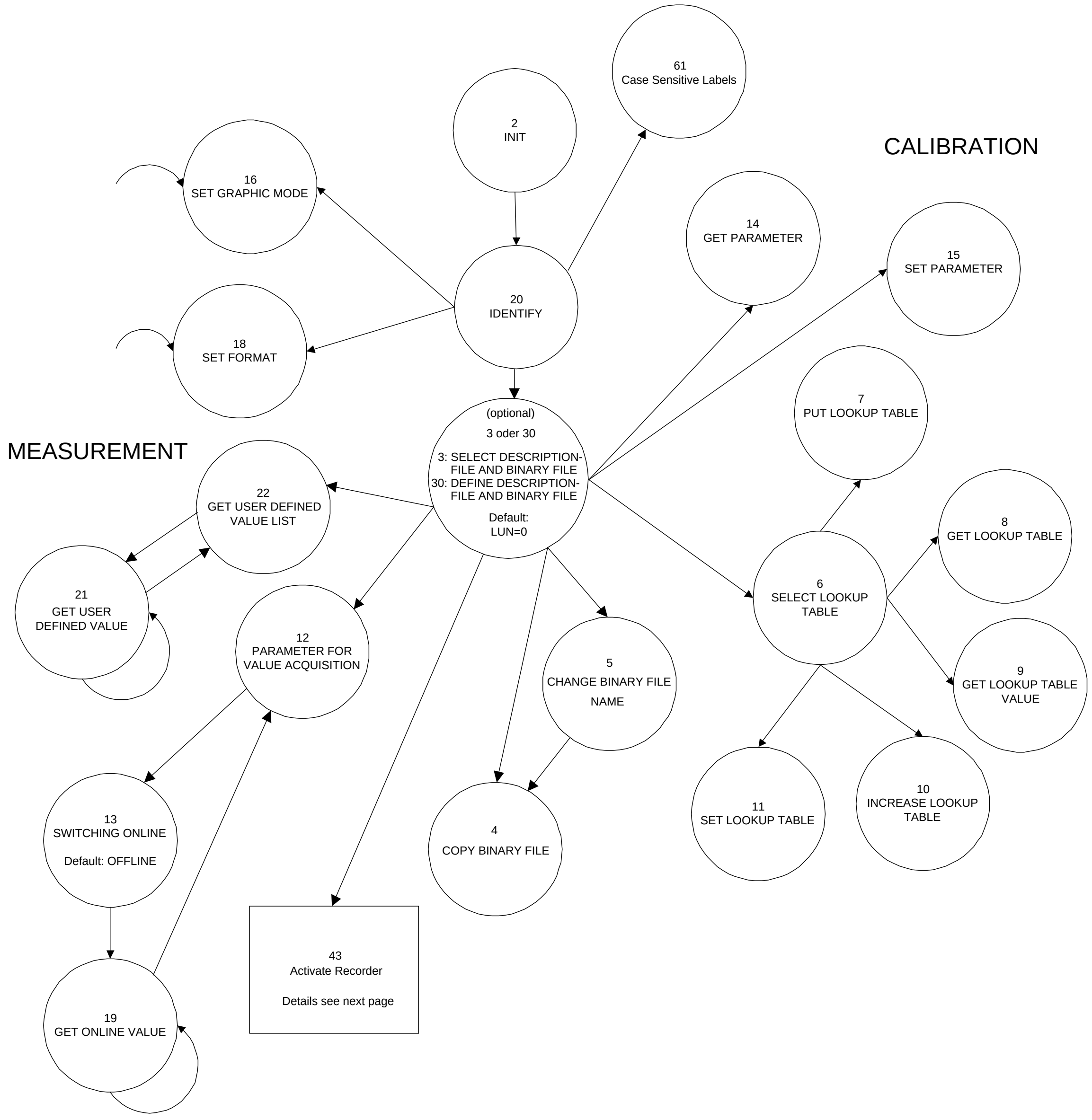
Entry	Value
Length	8
Code	13
Mode	0
Check sum	21

Answer transmitted by the MC system:

Entry	Value
Length	8
Code	13
Status	0
Check sum	21

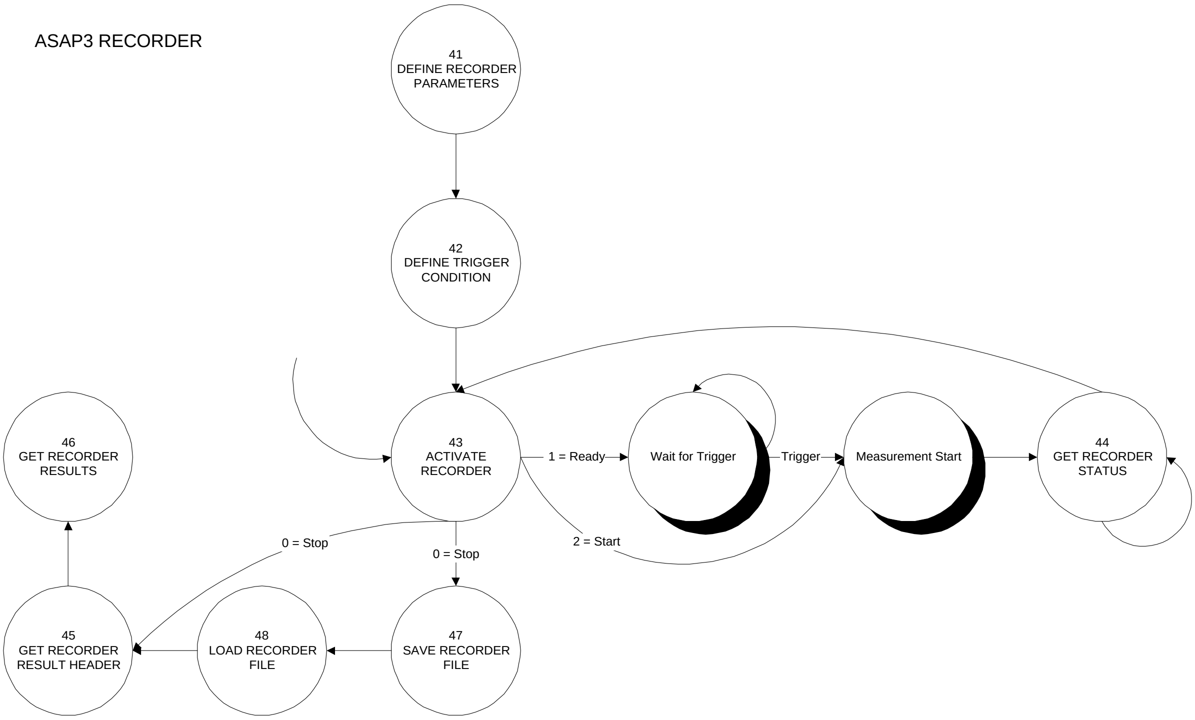
10 STATE DIAGRAM

10.1 STATE DIAGRAM WITHOUT RECORDER



]

10.2 STATE DIAGRAMM OF RECORDER



11 COMMAND OVERVIEW

Note:

The new EXTENDED commands are not an official part of the standard version 2.1. They are rather supposed to be used for test implementations. The EXTENDED commands will be finally released in version 2.2. In this document, their special status is indicated by a shaded background.

COMMAND	Code	Code/hex	page
ACTIVATE RECORDER	43	002B h	37
CHANGE BINARY FILE NAME	5	0005 h	24
COPY BINARY FILE	4	0004 h	23
DEFINE DESCRIPTION FILE AND BINARY FILE	30	001E h	22
DEFINE RECORDER PARAMETER	41	0029 h	36
DEFINE TRIGGER CONDITION	42	002A h	37
EMERGENCY	1	0001 h	20
EXIT	50	0032 h	21
EXTENDED GET LOOK-UP TABLE	108	006C h	29
EXTENDED GET LOOK-UP TABLE VALUE	109	006D h	29
EXTENDED GET ONLINE VALUE	119	0077 h	34
EXTENDED GET PARAMETER	114	0072 h	35
EXTENDED GET RECORDER RESULT	146	0092 h	40
EXTENDED GET RECORDER RESULT DATA TYPES	149	0095 h	40
EXTENDED INCREASE LOOK-UP TABLE	110	006E h	30
EXTENDED PARAMETER FOR VALUE ACQUISITION	112	0070 h	31
EXTENDED PUT LOOK-UP TABLE	107	006B h	28
EXTENDED SELECT LOOK-UP TABLE	106	006A h	28
EXTENDED SET LOOK-UP TABLE	111	006F h	30
EXTENDED SET PARAMETER	115	0073 h	35
GET LOOK-UP TABLE	8	0008 h	26
GET LOOK-UP TABLE VALUE	9	0009 h	26
GET ONLINE VALUE	19	0013 h	32
GET PARAMETER	14	000E h	34
GET RECORDER RESULT HEADER	45	002D h	38
GET RECORDER RESULTS	46	002E h	39
GET RECORDER STATUS	44	002C h	38
GET USER DEFINED VALUE	21	0015 h	33
GET USER DEFINED VALUE LIST	22	0016 h	33
IDENTIFY	20	0014 h	20
INCREASE LOOK-UP TABLE	10	000A h	27
INIT	2	0002 h	20
LOAD RECORDER FILE	48	0030 h	41
PARAMETER FOR VALUE ACQUISITION	12	000C h	31
PUT LOOK-UP TABLE	7	0007 h	25
RESET DEVICE	17	0011 h	42
SAVE RECORDER FILE	47	002F h	41
SELECT DESCRIPTION FILE AND BINARY FILE	3	0003 h	21
SELECT LOOK-UP TABLE	6	0006 h	24
SET CASE SENSITIVE LABELS	61	003D h	44
SET FORMAT	18	0012 h	43
SET GRAPHIC MODE	16	0010 h	42

SET LOOK-UP TABLE	11	000B h	27
SET PARAMETER	15	000F h	35
SWITCHING OFFLINE/ONLINE	13	000D h	32

12 GLOSSARY

ASAP2

ASAP2 specifies the data description for the MC system, see ASAP2 interface specification (http://www.asam.de).

ACS

Automatic **C**alibration **S**ystem - a computer based tool which enables a decision process to be defined, so that given environment the optimal parameters of an ECU system can be identified.

AuSy

Automation **Sy**stem - a test bench system (TBS) which is supported by an optimisation system for automatic map tuning (calibration work) via the ASAP3 interface.

Baud:

Baud rate refers to the rate at which the communication line transfers states(for example tones/notes used to represent the states ‘1’ or ‘0’). When each state represents a single bit, the rate in which states are transferred is the same as the rate at which bits are communicated, making the bit rate equal to the baud rate.

The measure of speed in serial communications which indicates the number of bits that can be transmitted in one second is bits per second (bps). This is often confused with baud rate, as their values are typically the same at many rates.

The term is named after J.M.E. Baudot, the inventor of the Baudot telegraph code.

Baud rate

see “Baud”.

Calibration

The activity of systematically tuning of characteristics and parameters of an ECU in real time.

Data description

The data description file describes the ECU internal data, e.g. conversion procedures, structure of calibration data, measurements or online values, which are used or generated by a running ECU and processed in the MC system.

The data description will be generated by a special data declaration tool and stored in the data description file. The ECU data description based on ASAP2 format is delivered from the ECU supplier to the carmaker. For further information, see ASAP2 interface specification.

Data description file

see “Data description”.

ECU data description

see “Data description”.

MC System

Measurement and **C**alibration System - is a tool used in the field of automotive development, manufacturing and after sale service. The currently designed MC part of the system, mostly used for calibration and measurement work of ECUs on the test benches or in the car, also supports the ASAP3 interface for automatic map tuning (calibration).

There are future plannings for a new ASAP3 interface (ACS interface) based on CORBA (see ASAM-C working group).

TBS

Test **B**ench **S**ystem, see AuSy.

RS-232C

A standard interface (RS stands for *Recommended Standard*) approved by the Electronic Industries Association (EIA) for connecting serial devices.

Most personal computers have an RS-232C port for connecting a modem or other device. The RS-232C standard supports two types of connectors - a 25-pin D-type connector (DB-25) and a 9-pin D-type connector (DB-9). The type of serial communications used by PCs requires only 9 pins so either type of connector will work equally well.

Although RS-232C is still the most common standard for serial communication, the EIA has recently defined successors to RS-232C called RS-422 and RS-423. The new standards are backward compatible so that RS-232 devices can connect to an RS-422 port.

In 1987, the EIA upgraded and changed the named of RS-232C. The new name is EIA-232D, but the standard is generally still referred to by its old name.

13 ABBREVIATIONS

AuSy	**Au**tomation **Sy**stem (incl. optimisation system)
ECU	**E**lectronic **C**ontrol **U**nit
HEX	**Hex**adecimal (0 ... F H for Intel or $0 ... $F for Motorola)
MC System	**M**easurement and **C**alibration **System**
ACS	**A**utomatic **C**alibration **S**ystem
ASAP	“**A**rbeitskreis zur **S**tandardisierung von **Ap**plikationssystemen”, the German synonym, which is also well-known as Working Group of Standardisation of Calibration and Diagnosis systems (MC systems). For further information see Internet page: http://www.asam.de
ASAP2	ASAP2 interface - see ASAP
ASAM	**A**ctivity Group for the **S**tandardisation of **A**utomation and **M**easuring System
TBS	**T**est **B**ench **S**ystem

14 INDEX