

Einführung in ASAP2

Worum geht es ?

- Was ist ASAP2 ?
- Wie ist eine ASAP2-Beschreibung aufgebaut ?
- Was ist wichtig für INCA ?
Was macht INCA mit den ASAP2-Informationen ?
- Welche „Fallen“ gibt es ?
Was passiert, wenn ich etwas falsch beschreibe ?
- Was sind die Unterschiede zwischen ASAP2 V1.21 und ASAP2 V1.3 ?
- Wo kann ich nachschlagen, wenn ich mehr wissen will ?

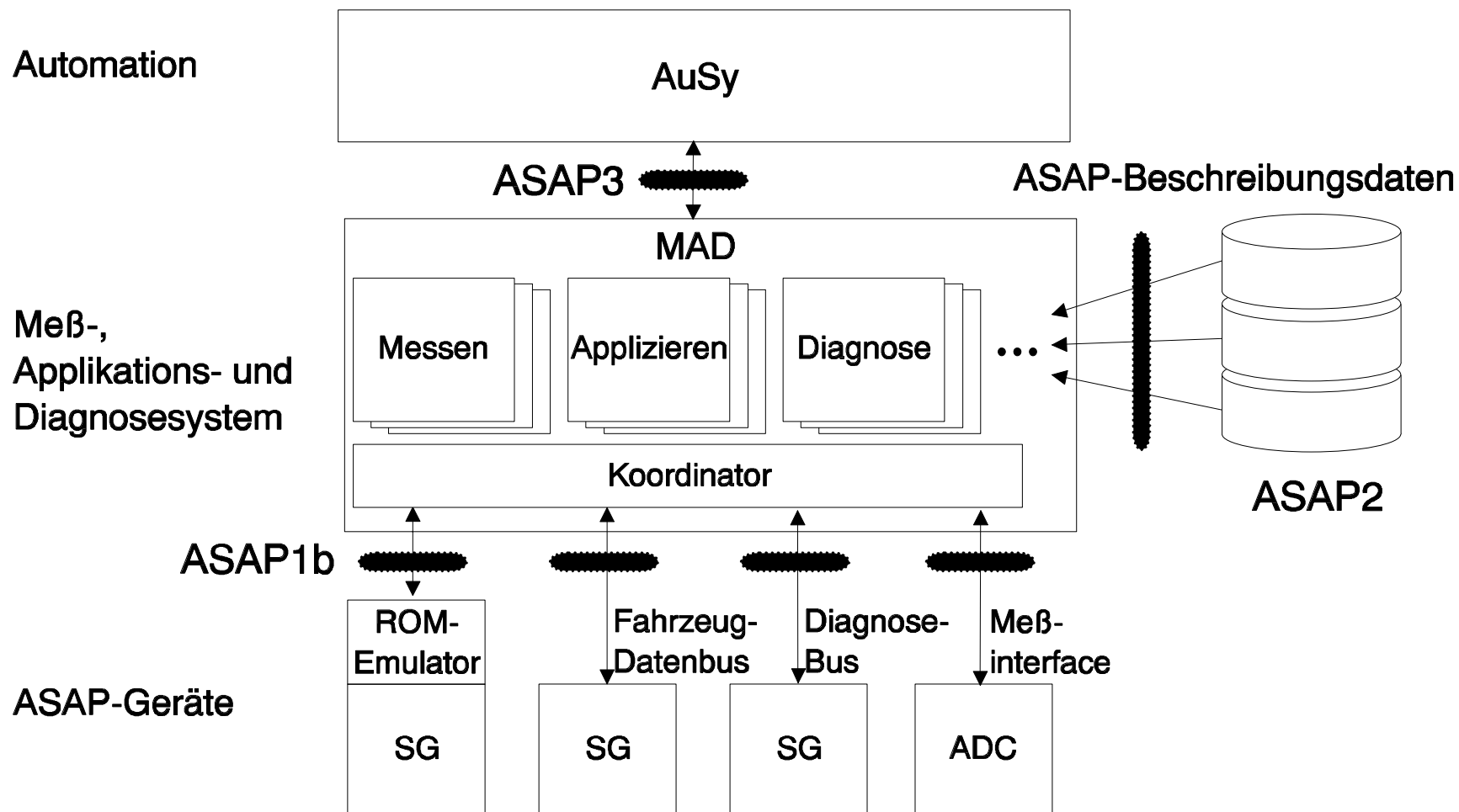
Einführung in ASAP2

Was ist ASAP(2) ?

- ASAP = **A**rbeitskreis zur **S**tandardisierung von **A**pplikationssystemen
(engl. Applications Systems Standardization Working Group)
- Gründung im Herbst 91 auf Initiative großer deutscher Automobilhersteller
- Ziel ist Kostenreduktion durch Vereinheitlichung von Schnittstellen zu Applikationssystemen
- Mitglieder sind Vertreter von Automobilherstellern (z.B. Audi, BMW, DaimlerChrysler, Porsche, VW, Volvo) und Toolherstellern
- Unterarbeitskreis von ASAM (= **A**rbeitskreis zur **S**tandisierung von **A**utomatisierungs- und **M**eißsystemen)

Einführung in ASAP2

Überblick ASAP-Schnittstellen



Einführung in ASAP2

Überblick ASAP2

- Eine ASAP2-Datei beschreibt
 - allg. projektspezifische Daten des SG
 - Speicheraufbau des Steuergeräts
 - Zugriffsinformationen über bestimmte Applikationsschnittstellen
 - Meß- und Verstellgrößen im SG auf physikalischer Ebene; Umrechnungsformeln; Zugriffs- und Strukturinformationen
 - Zuordnung von Meß- und Verstellgrößen zu Funktionen; Zugriffsrechte
 - spezielle Applikationsverfahren
 - Variantencodierung

Einführung in ASAP2

Entwicklung von ASAP2

- unterschiedliche ASAP2-Versionen, z.B.
 - 3/94: Version 1.0
 - 9/97: Version 1.21 (neue Schlüsselworte, alte Schlüsselworte teilweise durch /begin .. /end gekapselt)
 - 11/98: Version 1.3 (neue Schlüsselworte)
- Erweiterungen werden durch neue Anforderungen der SG-Hersteller getrieben und im Rahmen der Arbeitskreissitzungen abgestimmt und verabschiedet
- Aufteilung in ASAP2-D (Diagnose), ASAP2-MC (Messen + Verstellen)
- geplant:
 - Zusammenführung von ASAP2-D und ASAP2-MC
 - SGML-Format statt ASCII-Format (ASAP2 MSR)
 - Überarbeitung von Schlüsselworten

Einführung in ASAP2

Grobstruktur einer ASAP2-Datei

```
/begin PROJECT
  /begin HEADER ...
  /end HEADER

  /begin MODULE
    /begin MOD_PAR ...
      /begin MEMORY_LAYOUT ...
      /end MEMORY_LAYOUT
      ...
    /end MOD_PAR

    /begin MOD_COMMON ...
    /end MOD_COMMON

    /begin IF_DATA ...
    /end IF_DATA
    ...

    /begin CHARACTERISTIC ...
    /end CHARACTERISTIC
    ...

    /begin MEASUREMENT ...
    /end MEASUREMENT
    ...

    /begin COMPU_METHOD ...
    /end COMPU_METHOD
    ...

    /begin RECORD_LAYOUT ...
    /end RECORD_LAYOUT ...
    ...

  /end MODULE
  ...
/end PROJECT
```

allgemeine Projektinformationen

Beschreibung eines Steuergeräts (eine Sektion pro Steuergerät)
allg. Daten dieses Steuergeräts (einmal pro MODULE)
Beschreibung eines Speicherbereichs im Steuergerät (eine Sektion pro Speicherbereich)

Standardwerte (Bytereihenfolge, Ablageschema, ..), die für gesamte MODULE-Sektion gelten (einmal pro MODULE)

Zugriffsinformation für eine bestimmte Applikationsschnittstelle bzw. ein bestimmtes Protokoll (ETK, McMess / KWP2000 über K-Leitung, CAN-CCP, ...)
(eine Sektion pro Applikationsschnittstelle/Protokoll)

Beschreibung einer Verstellgröße (eine Sektion pro Verstellgröße)

Beschreibung einer Meßgröße (eine Sektion pro Meßgröße)

Beschreibung einer Umrechnungsformel (eine Sektion pro Umrechnungsformel)

Beschreibung eines Verstellgrößen-Ablageschemas (eine Sektion pro Ablageschema)

(Reihenfolge der Sektionen ist beliebig)

Einführung in ASAP2

IF_DATA-Sektion

- Allgemein: IF_DATA-Sektionen beschreiben applikationsinterface-spezifische bzw. protokollspezifische Daten
- IF_DATA-Sektionen an verschiedenen Stellen möglich, z.B. bei MODULE, MEMORY_LAYOUT / MEMORY_SEGMENT, CHARACTERISTIC, MEASUREMENT
- Inhalt nicht standardisiert, sondern treiberspezifisch; unterschiedliche Versionen der IF_DATA-Sektionen für ein Applikationsinterface möglich
- Festlegung durch Treiberhersteller, formale Beschreibung durch AML-Datei
- Inhalt wird (oft) als ‚Binary large object‘ (‚Blob‘) an den Treiber übergeben

Einführung in ASAP2

IF_DATA; Beispiel ETK

Uralt:

```
/begin IF_DATA DIM
    VECTAB_ADDRESS    0xB1EC
    TRGSEG_ADDRESS    0xF000
    DAT_MOD            1
    TRG_MOD            39
    DISPLAY_TAB        12
    DISPLAY_PAR        0xDF56 8 0xF000 0x0 "Syncro" 0
    DISPLAY_PAR        0xDF68 20 0xF100 0xF038 "TimeB" 10
    DISPLAY_PAR        0xDF92 20 0xF200 0xF03A "TimeC" 100
/end IF_DATA
```

Ganz alt:

```
/begin IF_DATA DIM_X
    TRGSEG_ADDRESS    0x383200
    DAT_MOD            1
    TRG_MOD            39
    DISPLAY_TAB        12
    ETK_CFG            0xf 0xf0 0xff 0x3 0xff 0xee 0xff 0x1
    MEMORY_LAYOUT PRG_EXRAM 0x380000 0x8000 -1 -1 -1 -1 -1
    DISPLAY_PAR        0x817F68 10 0x3819C8 0x383236 "Synchro" 0
    DISPLAY_PAR        0x817F7E 26 0x3819D2 0x383238 "10ms" 10
    DISPLAY_PAR        0x817FB4 36 0x3819EC 0x38323A "100ms" 100
/end IF_DATA
```

Einführung in ASAP2

IF_DATA; Beispiel ETK

Alt:

```
/begin IF_DATA ASAP1B_ETK
  /begin SOURCE
    "Synchro" 103 3
    QP_BLOB 1 10 0x814CBE 0x380C2C 0x381FF6 0 0 0
  /end SOURCE
  /begin SOURCE
    "10ms" 3 10
    QP_BLOB 2 26 0x814CD4 0x380C36 0x381FF8 10 0 0
  /end SOURCE
  /begin SOURCE
    "100ms" 3 100
    QP_BLOB 3 36 0x814D0A 0x380C50 0x381FFA 100 0 0
  /end SOURCE

  /begin TP_BLOB
    0x381FC0 1 39 12 2 0x0 0x0 0x0 0x0 0x0 0
    ETK_CFG F F0 FF 3 FF EE FF 1
    EXRAM 0x380000 0x387FFF -1 -1 -1 -1 -1
  /end TP_BLOB
/end IF_DATA
```

Neu:

```
/begin IF_DATA ETK
  /begin SOURCE "Synchro" 103 1
    QP_BLOB 1 0x3C 0x1000BC 0x12008C 0x120080 0 0 0
  /end SOURCE
  /begin SOURCE "10ms" 4 1
    QP_BLOB 2 0x58 0x1001B4 0x1200C8 0x120084 10 0 0
  /end SOURCE
  /begin SOURCE "100ms" 5 1
    QP_BLOB 3 0x58 0x10031C 0x120120 0x120088 100 0 0
  /end SOURCE

  /begin TP_BLOB
    0x1000001
    0x0
    0xFFFFFFF07
  /begin DISTAB_CFG
    0xD 0x1 0x1 0x120000 0x1020100
    TRG_MOD 0x1 0x27 0x0 0x0 0x0
  /end DISTAB_CFG
  CODE_CHK 0x0 0x0 0x0 0x0
  ETK_CFG 0xF 0xF0 0xFF 0x3 0xFF 0xEE 0xFF 0x1
  RESERVED 0x1000BC 0x104 EXTERN -1 -1 -1 -1 -1
  ETK_MAILBOX 0x55555555 DATA 0x600000 0x8000
    EXRAM 0x620000 0x8000
  /end TP_BLOB
/end IF_DATA
```

Versionsnummer !

Falsche Daten in der IF_DATA-Sektion führen dazu, daß der Devicetreiber nicht korrekt arbeitet (z.B. funktioniert Messen nicht) !

Einführung in ASAP2

MEMORY_LAYOUT

Beschreibung der Speicheraufteilung im Steuergerät,
z.B. ME71:

```
/begin MEMORY_LAYOUT
    PRG_CODE 0x800000 0x10000 -1 -1 -1 -1
    /begin IF_DATA ASAP1B_ETK
        DP_BLOB 0x800000 0x10000
    /end IF_DATA
/end MEMORY_LAYOUT
/begin MEMORY_LAYOUT
    PRG_CODE 0x820000 0x60000 -1 -1 -1 -1
    /begin IF_DATA ASAP1B_ETK
        DP_BLOB 0x820000 0x60000
    /end IF_DATA
/end MEMORY_LAYOUT
/begin MEMORY_LAYOUT
    PRG_DATA 0x810000 0x10000 -1 -1 -1 -1
    /begin IF_DATA ASAP1B_ETK
        DP_BLOB 0x810000 0x10000
    /end IF_DATA
/end MEMORY_LAYOUT
```

Codebereich
0x800000..0x80FFFF
im SG-Flash

Codebereich
0x820000..0x87FFFF
im SG-Flash

Datenbereich
0x810000..0x81FFFF
im SG-Flash

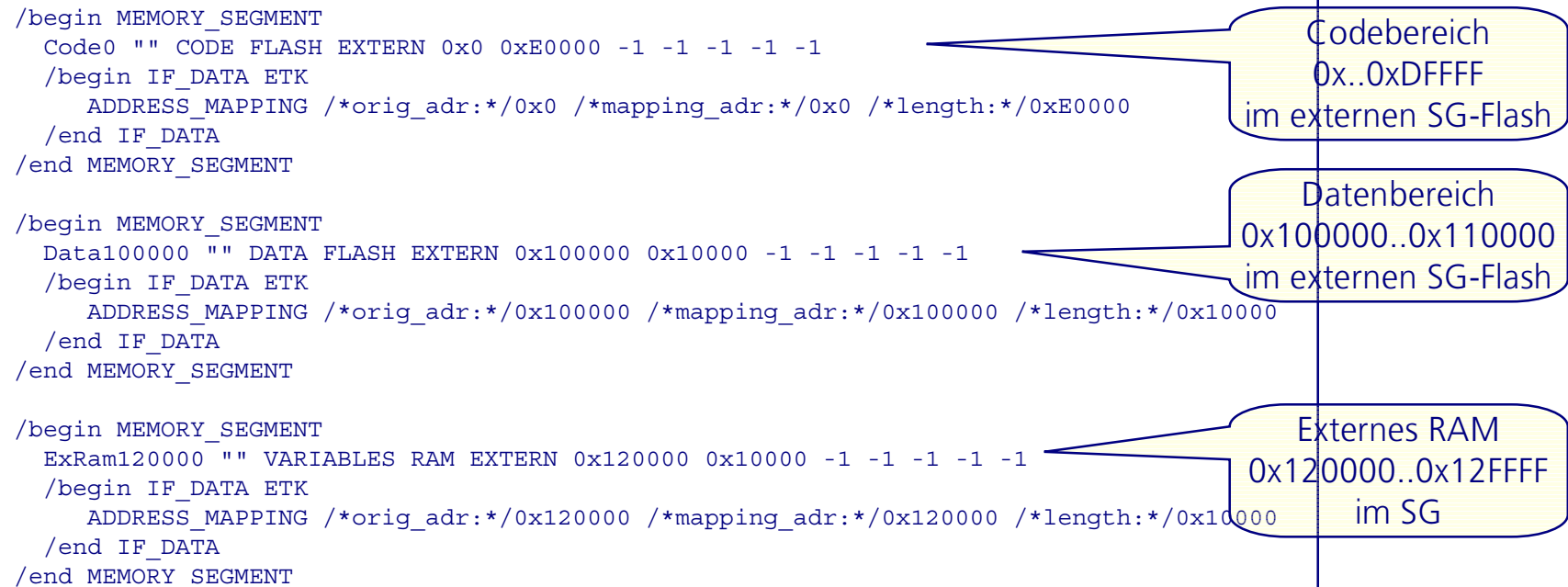
```
/begin IF_DATA ASAP1B_ETK
    /begin TP_BLOB
        ...
        EXRAM 0x380000 0x387FFF -1 -1 -1 -1
    /end TP_BLOB
    ...
/end IF_DATA
```

Externes RAM
0x380000..0x387FFF
auf SG

Einführung in ASAP2

MEMORY_SEGMENT

Beispiel ME92:



Eine falsche Beschreibung der Speicheraufteilung im Steuergerät führt dazu, daß der Devicetreiber nicht korrekt arbeitet !
MEMORY_SEGMENT ist gleichwertig zu MEMORY_LAYOUT, enthält aber mehr Informationen.

Einführung in ASAP2

MEMORY_SEGMENT

Weitere spezielle Typen von Speicherbereichen:

```
/begin MEMORY_SEGMENT
  Code "" CODE FLASH EXTERN 0x0 0xE0000 -1 -1 -1 -1 -1
/end MEMORY_SEGMENT

/begin MEMORY_SEGMENT
  Reserved "" RESERVED FLASH EXTERN 0xE0000 0x10000 -1 -1 -1 -1 -1
/end MEMORY_SEGMENT

/begin MEMORY_SEGMENT
  Data "" DATA FLASH EXTERN 0x100000 0x10000 -1 -1 -1 -1 -1
/end MEMORY_SEGMENT

/begin MEMORY_SEGMENT
  OfflineData "" OFFLINE_DATA FLASH EXTERN 0x110000 0x10000 -1 -1 -1 -1 -1
/end MEMORY_SEGMENT

/begin MEMORY_SEGMENT
  ExRam "" VARIABLES RAM EXTERN 0x120000 0x10000 -1 -1 -1 -1 -1
/end MEMORY_SEGMENT
```

Bereich mit
„reservierten“ Daten

Bereich mit nur offline
applizierbaren Daten

- Reservierte Bereiche und Offline-Datenbereiche werden in der Arbeitsbasis abgelegt, aber nicht dem Devicetreiber bekanntgemacht
- Kenngrößen in Offline-Datenbereichen können nur offline verstellt werden
- beide Bereiche können geflasht werden

Einführung in ASAP2

MEASUREMENT

Beschreibung von Meßgrößen

In ASAP2 1.21:

```
/begin MEASUREMENT
nmot
"Motordrehzahl"
UBYTE
nmot_ub_q40
0
0
0.000000
10200.000000
BIT_MASK 0xFF
ARRAY_SIZE 5

/begin IF_DATA ASAP1B_ETK
KP_BLOB 0x380C74 1 1
/end IF_DATA
/end MEASUREMENT
```

Name der Meßgröße (muß eindeutig sein)
„Langname“ der Meßgröße (Kurzkommentar)
Datentyp der Meßgröße
Name der zugehörigen Umrechnungsformel (Verweis auf COMPU_METHOD)
Quantisierung (kleinste mögliche Differenz) (wird von INCA ignoriert)
Genauigkeit (mögliche Abweichung vom exakten Wert in %) (wird von INCA ignoriert)
Untere phys. Plausibilitätsgrenze der Meßwerte
Obere phys. Plausibilitätsgrenze der Meßwerte
Optional: Bitmaske (hier überflüssig)
Optional: Beschreibung eines Arrays von n Meßgrößen mit gleichen Eigenschaften, die direkt hintereinander abgelegt sind
optional: spezifische Zugriffsinformation für diese Meßgröße
hier für ETK: Startadresse, Bytelänge und Adreßort (internes/externes RAM)

In ASAP2 1.3:

```
/begin MEASUREMENT
nmot
"Motordrehzahl"
UBYTE
nmot_ub_q40
0
0
0.000000
10200.000000
ECU_ADDRESS 0x380C74
BIT_MASK 0xFF
ARRAY_SIZE 5
/end MEASUREMENT
```

ECU_ADDRESS beschreibt Adresse im Steuergerät und macht dadurch IF_DATA-Sektionen überflüssig

Einführung in ASAP2

COMPU_METHOD

Beschreibung von Umrechnungsformeln

Beispiel: Lineare Umrechnungsformel

```
/begin COMPU_METHOD
  nmot_ub_q40
  ""
  RAT_FUNC
  "%6.1"
  "Upm"
  COEFFS 0 1.000000 0.000000 0 0.000000 40.000000
/end COMPU_METHOD
```

Name der Umrechnungsformel
„Langname“ der Umrechnungsformel (Kurzkommentar)
Formeltyp (RAT_FUNC, TAB_INTP, TAB_NOINTP, TAB_VERB, FORM)
Ausgabeformat (%[Gesamtlänge].[Nachkommastellen])
phys. Einheit
optional: Koeffizienten der rationalen Umrechnungsformel
 $\text{hex} = (\text{ax}^2 + \text{bx} + \text{c}) / (\text{dx}^2 + \text{ex} + \text{f})$ mit $x = \text{phys}$
 $\Rightarrow$ hier: $\text{hex} = 1/40 * \text{phys}$ bzw. $\text{phys} = 40 * \text{hex}$

Beispiel: Verbale Umrechnungsformel

```
/begin COMPU_METHOD
  B_TRUE
  "boolsche Zustände, positive Logik"
  TAB_VERB
  "%6.3"
  ""
  COMPU_TAB_REF B_TRUE
/end COMPU_METHOD

/begin COMPU_VTAB
  B_TRUE
  ""
  TAB_VERB
  2
  0 "----"
  1 "TRUE"
/end COMPU_VTAB
```

Formeltyp „Verbale tabellarische Umrechnungsformel“

Name der zugehörigen String-Tabelle

Beschreibung der String-Tabelle
Name der String-Tabelle
Langname
Tabellentyp (redundant; Wiederholung des Typs aus COMPU_METHOD)
Anzahl der Wertepaare
Wertepaare (Hexwert -> String)

Einführung in ASAP2

CHARACTERISTIC

Beschreibung von Verstellgrößen

Beispiel Kennlinie:

```
/begin CHARACTERISTIC
BRABEVI_KL
  "Beschleunigung bei Rampe Beschleunigen"
CURVE
0x814E00
DAMOS_KL111
0
bfzg_uw_qp1085
0.000000
7110.550000
EXTENDED_LIMITS 0.0 8000.0

/begin AXIS_DESCR
STD_AXIS
vfzg_w

vfzg_uw_b512
8
0.000000
511.992000
DEPOSIT ABSOLUT
/end AXIS_DESCR

/begin IF_DATA ASAP1B_ETK
DP_BLOB 0x814E00 0x23
/end IF_DATA
/end CHARACTERISTIC
```

Name der Verstellgröße (muß eindeutig sein)
„Langname“ der Verstellgröße (Kurzkommentar)
Typ der Verstellgröße (VALUE, CURVE, MAP, CUBOID, VAL_BLK, ASCII)
Startadresse der Verstellgröße im Datenbereich des Steuergerät
Ablageschema der Verstellgröße im Steuergerät (Verweis auf RECORD_LAYOUT)
max. Differenz zwischen zwei benachbarten Werten (von INCA ignoriert)
Name der Umrechnungsformel der Werte (Verweis auf COMPU_METHOD)
Untergrenze des plausiblen Wertebereichs der Werte („weiche Grenzen“)
Obergrenze des plausiblen Wertebereichs der Werte („weiche Grenzen“)
optional: Unter-/Obergrenze des zulässigen Wertebereichs der Werte („harte Grenzen“)
Beschreibung der (x-)Stützstellen
Typ der Stützstellen (STD_AXIS, FIX_AXIS, COM_AXIS, RES_AXIS)
Name der zugehörigen Meßgröße (Verweis auf MEASUREMENT)
oder NO_INPUT_QUANTITY
Name der Umrechnungsformel der Stützstellen (Verweis auf COMPU_METHOD)
Maximale Stützstellenanzahl
Untergrenze des plausiblen Wertebereichs der Stützstellen („weiche Grenzen“)
Obergrenze des plausiblen Wertebereichs der Stützstellen („weiche Grenzen“)
optional: Ablageart der Stützstellen (ABSOLUTE, DIFFERENCE)

optional: spezifische Zugriffsinformation für diese Verstellgröße
hier für ETK: Startadresse und Bytelänge; wird von INCA ignoriert, da redundant !

Einführung in ASAP2

CHARACTERISTIC

Beispiel Gruppenkennfeld:

```
/begin CHARACTERISTIC
  KFZW_GKF "Zündwinkelkennfeld"
  MAP 0x812C80 DAMOS_GKF2 0 zw_sb_q0p75 -96.000000 95.250000
  /begin AXIS_DESCR
    COM_AXIS nmot nmot_ub_q40 16 0.000000 10200.000000
    AXIS_PTS_REF SNM16ZUUB_GST
  /end AXIS_DESCR
  /begin AXIS_DESCR
    COM_AXIS rl rel_ub_q0p75 12 0.000000 191.250000
    AXIS_PTS_REF SRL12ZUUB_GST
  /end AXIS_DESCR
/end CHARACTERISTIC
```

Verweis auf zugehörige
x-Gruppenstützstellenverteilung

Verweis auf zugehörige
y-Gruppenstützstellenverteilung

```
/begin AXIS_PTS
  SNM16ZUUB_GST "Stützstellenverteilung Drehzahl, 16 Sst."
  0x810718 nmot DAMOS_SST97 0 nmot_ub_q40 16 0.000000 10200.000000
  DEPOSIT ABSOLUT
/end AXIS_PTS
```

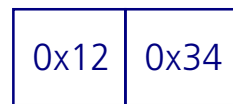
Separate Beschreibung der
Stützstellenverteilungen als
eigene Kenngrößen

```
/begin AXIS_PTS
  SRL12ZUUB_GST ""
  0x810752 rl DAMOS_SST97 0 rel_ub_q0p75 12 0.000000 191.250000
  DEPOSIT ABSOLUT
/end AXIS_PTS
```

Einführung in ASAP2

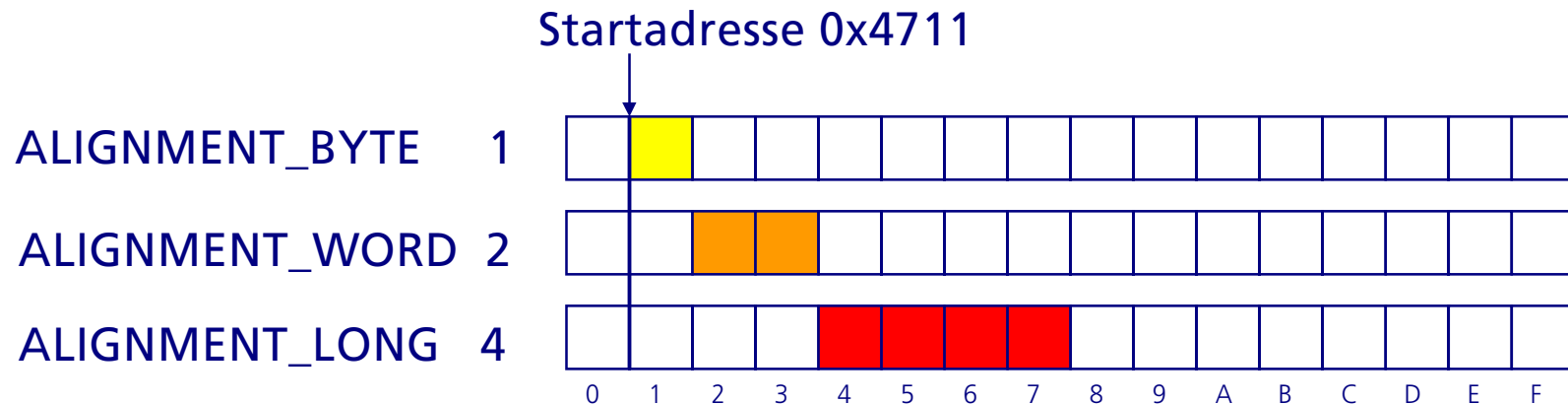
BYTE_ORDER und ALIGNMENT

```
/begin MOD_COMMON ""  
    BYTE_ORDER MSB_LAST  
    ALIGNMENT_BYTE 1  
    ALIGNMENT_WORD 2  
    ALIGNMENT_LONG 4  
/end MOD_COMMON
```



BYTE_ORDER MSB_FIRST => 0x1234

BYTE_ORDER MSB_LAST => 0x3412



Bei einem falschen Alignment stimmt die „Startadresse“ der Verstellgröße nicht, bei einer falschen Byteorder werden die Werte der Verstellgröße falsch dargestellt !

Einführung in ASAP2 RECORD_LAYOUT

```
/begin RECORD_LAYOUT DAMOS_KL111
NO_AXIS_PTS_X 1 UBYTE
AXIS_PTS_X 2 UWORD INDEX_INCR DIRECT
FNC_VALUES 3 UWORD COLUMN_DIR DIRECT
/end RECORD_LAYOUT
```

Interpretation aus
RECORD_LAYOUT

Hex-Editor

File Options Help

Address Refresh time [s]

Address	Page	0	1	2	3	4	5	6	7
814E00h	WP (DS)	08h	00h	00h	0Fh	00h	1Eh	00h	2Dh
814E08h	WP (DS)	00h	3Ch	00h	4Bh	00h	5Ah	00h	69h
814E10h	WP (DS)	00h	7Dh	09h	00h	09h	00h	09h	00h
814E18h	WP (DS)	09h	00h	07h	00h	07h	00h	07h	00h
814E20h	WP (DS)	07h	00h	08h	00h	00h	0Fh	00h	1Eh

Daten aus SG

Table Editor [1]

[hex] ☒ vfzg_w [hex]

x	0F00h	1E00h	2D00h	3C00h	4B00h	5A00h	6900h	7D00h
z	09h	09h	09h	09h	07h	07h	07h	07h

Bei einem falschen RECORD_LAYOUT werden die Verstellgrößen falsch dargestellt !

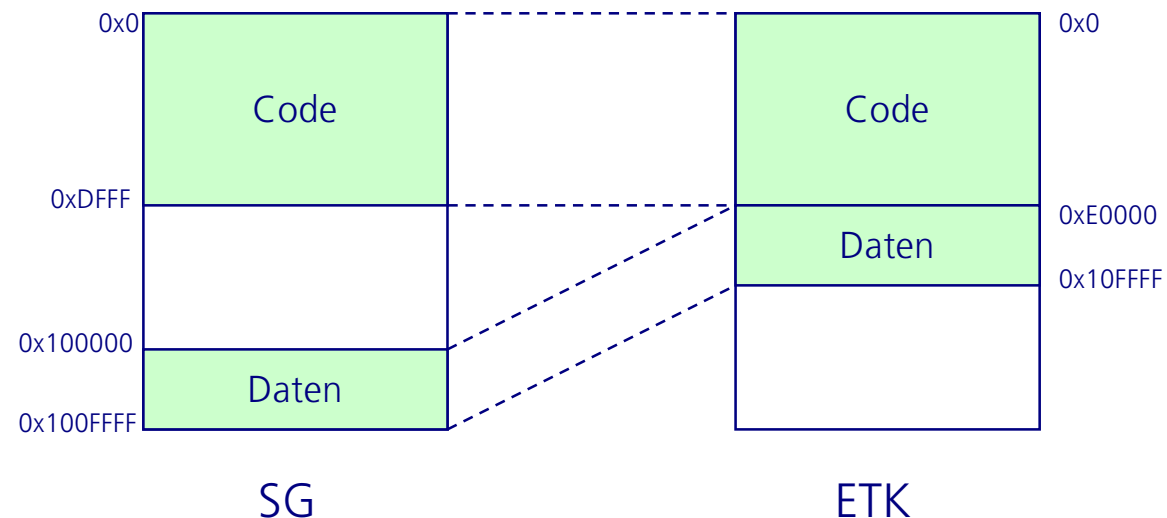
ASAP2 für Spezialisten

ADDRESS_MAPPING

Beschreibt eine devicespezifische Abbildung der SG-Adressen auf Device-
(z.B. ETK-) Adressen;
wirkt sich auf die Übergabe aller Adressen an den Devicetreiber aus

```
/begin MEMORY_SEGMENT
  Code0 "" CODE FLASH EXTERN 0x0 0xE0000 -1 -1 -1 -1 -1
/end MEMORY_SEGMENT

/begin MEMORY_SEGMENT
  Data100000 "" DATA FLASH EXTERN 0x100000 0x10000 -1 -1 -1 -1 -1
  /begin IF_DATA ETK
    ADDRESS_MAPPING /*orig_adr:*/0x100000 /*mapping_adr:*/0xE0000 /*length:*/0x10000
  /end IF_DATA
/end MEMORY_SEGMENT
```



ASAP2 für Spezialisten

CALIBRATION_METHOD

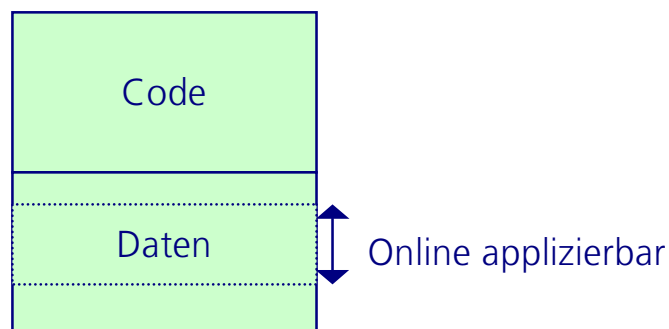
CALIBRATION_METHOD gibt deviceübergreifende Informationen zu bestimmtem Applikationsverfahren;

Beispiel serielles Applikationsverfahren D-SERAP

```
/begin MEMORY_LAYOUT PRG_CODE 0x8000 0x20000 -1 -1 -1 -1 -1
/end MEMORY_LAYOUT
/begin MEMORY_LAYOUT PRG_DATA 0x28000 0x8000 -1 -1 -1 -1 -1
/end MEMORY_LAYOUT

/begin CALIBRATION_METHOD
  "DSERAP"          /* Name of Method */
  1                 /* Version No. */
  /begin CALIBRATION_HANDLE
    0x28500          /* Begin of Calibration-Data */
    0x2000           /* Length of Calibration-Data */
  /end CALIBRATION_HANDLE
/end CALIBRATION_METHOD
```

Einschränkung des
applizierbaren
Datenbereichs



SG-Flash

Einführung in ASAP2

Unterschiede zwischen ASAP2 V1.21 und V1.3

- Neu in ASAP2 V1.3:
 - MEMORY_SEGMENT statt MEMORY_LAYOUT; erlaubt Spezifikation zusätzlicher Attribute
 - ADDRESS_MAPPING (bei MEMORY_SEGMENT)
 - ECU_ADDRESS (bei MEASUREMENT; erlaubt Weglassen der IF_DATA-Sektion)
 - CALIBRATION_METHOD
 - ECU_CALIBRATION_OFFSET
 - VIRTUAL_CHARACTERISTIC
 - Beschreibung des Recordlayouts von Rescale-Kenngrößen (derzeit Chrysler-spezifisch)

Einführung in ASAP2

Literatur

- \\Fe13606\VS:\Doku\Asap\Asap2-MC
Spezifikationen und Arbeitskreisprotokolle zu ASAP2 und den anderen ASAP-Schnittstellen;
z.B. Spezifikation für ASAP2 V1.3:
\\Fe13606\VS:\Doku\Asap\Asap2-MC\Spec\ASAP2_V130\Asap2.doc
- \\Fe13606\VS:\Doku\Asap2\Aml
AML-Dateien (Beschreibung des Aufbaus der IF_DATA-Sektionen) für alle von INCA unterstützten Devices
- \\Fe13606\VS:\Doku\Asap2-MC\INCA-Support
Liste(n) der von INCA unterstützten ASAP2-Schlüsselworte
- \\Fe13606\IND:\SG-Test; \\Fe13606\IND:\SG-Freigabe
Massenhaft ASAP2-Dateien zu verschiedensten Steuergeräten