

Öltemperatur statt KVA im E60 Kombi codieren

Erstellt von intershopper, mit Hilfe von acolt und hates.

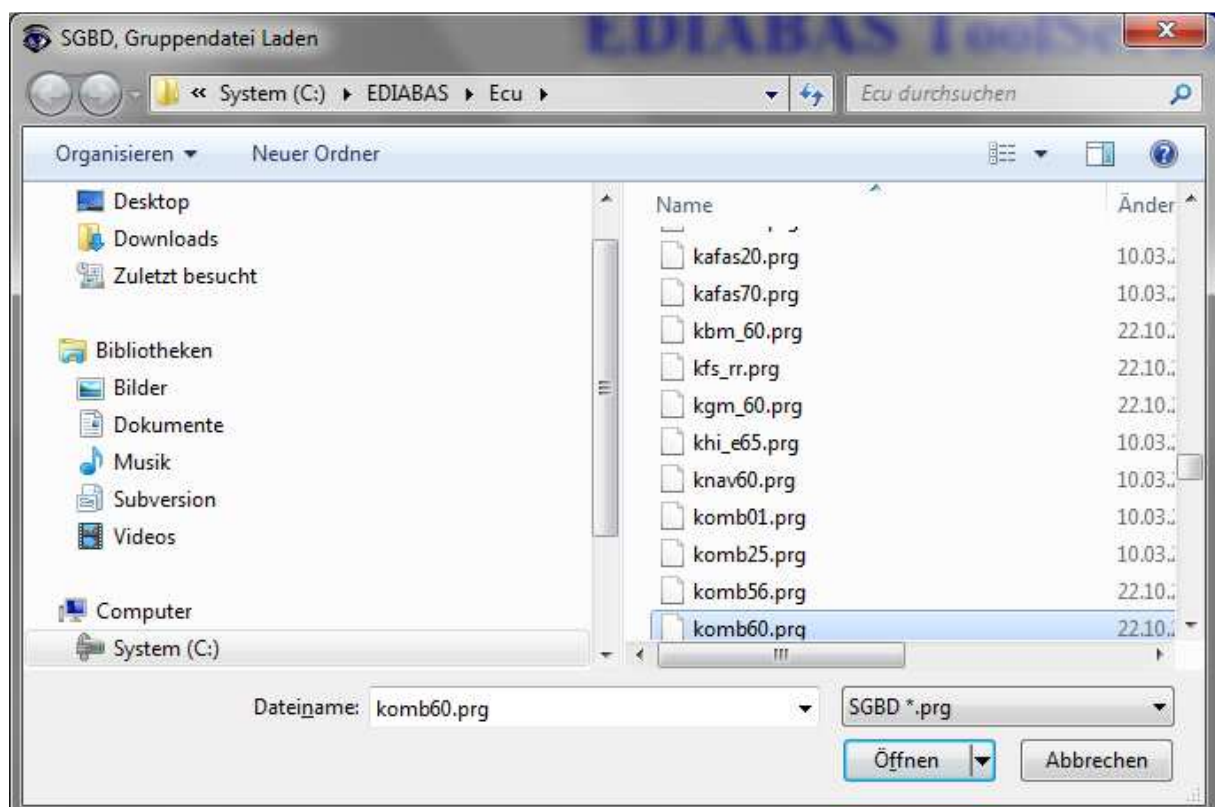
1 Vorbedingung

ODB 2 Kabel ist einsatzbereit und mit dem Auto verbunden. Ich verwende das MaxDia Diag 2+ - Diagnose Interface für BMW-Fahrzeuge(Bj. 2007 - 2016)

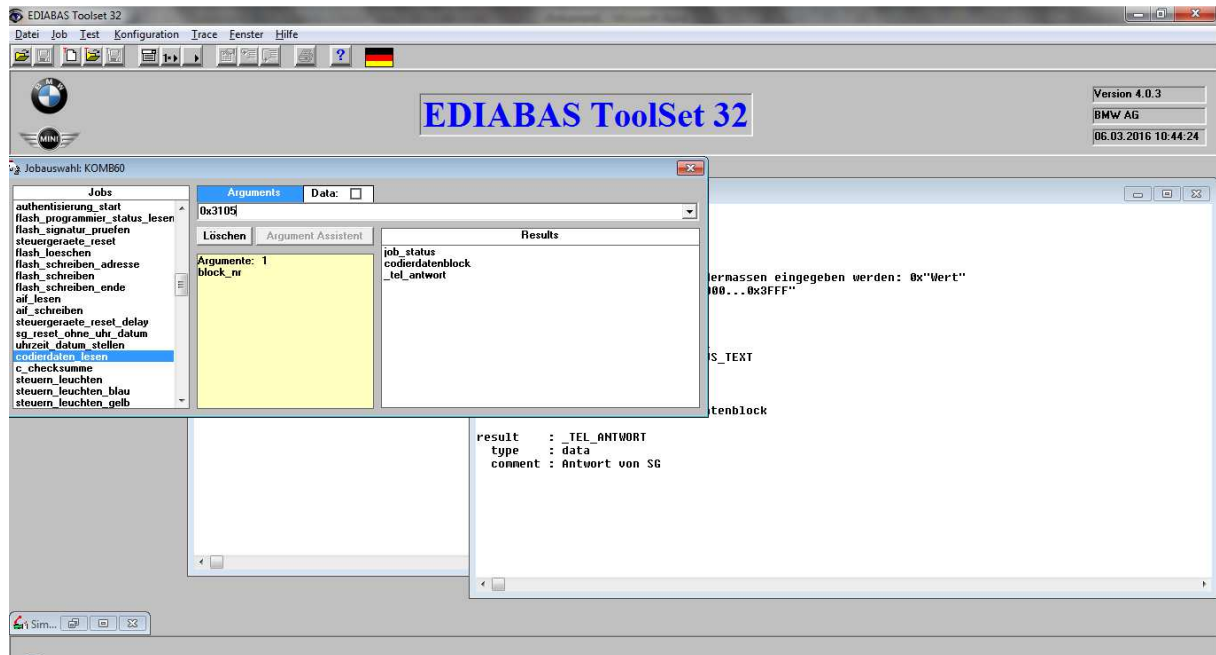
<https://www.obdexpert.de/prestashop/home/54-9-maxdia-diag-2-bmw-diagnose-interface-bj-032007-2013html.html>

Vorkenntnisse Tool32 bedienen und Umgang mit Hexadezimalen Zahlen.

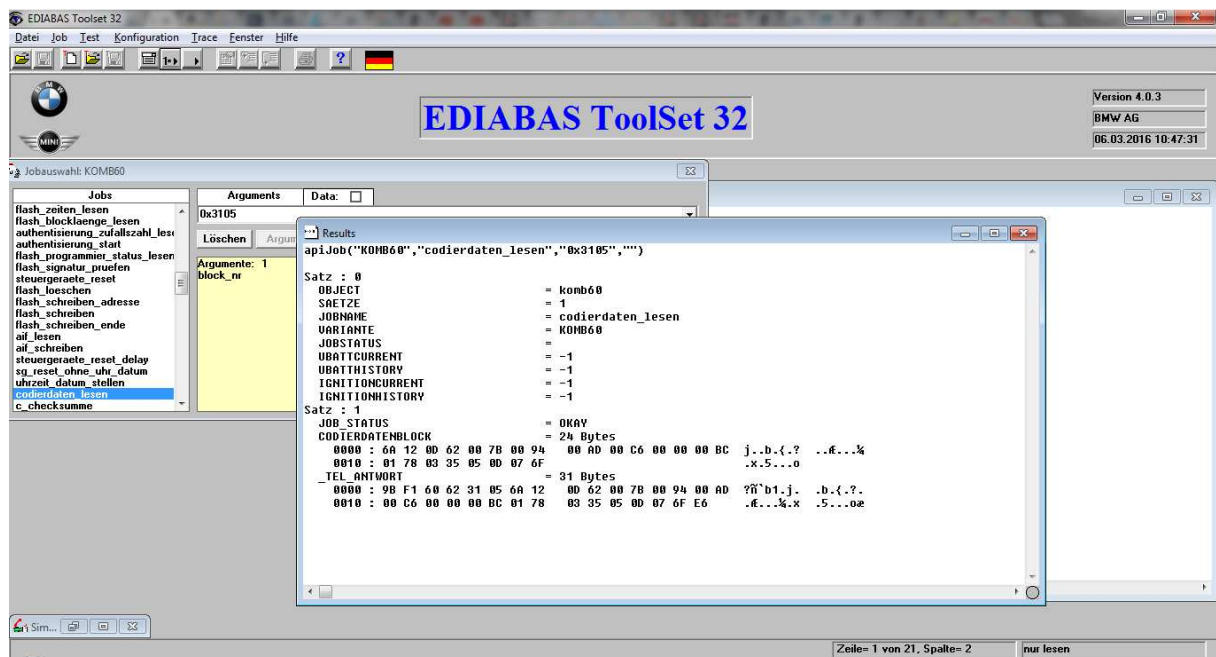
Tool32 starten und Komb60.prg laden. Datei -> SGBD, Gruppdatei laden



Job codierdaten_lesen, dort einfach als Argument 0x3105 eingeben. Und Job einmal ausführen.



In Results sollten dann die Codierdaten erscheinen

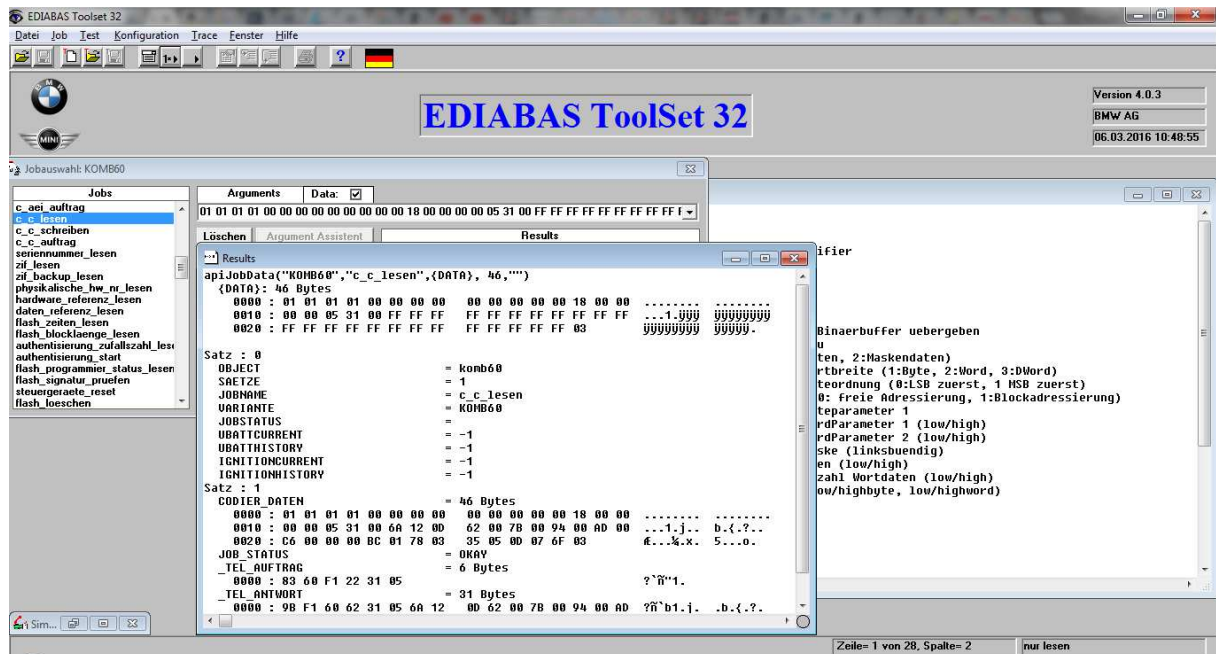


Das Ergebnis ist das gleiche, was du mit c_c_Lesen erhältst.

Beim Job c_c_lesen musst du den folgen Buffer als Argument angeben und DATA anhängen

c_c_lesen Buffer:

```
01 01 01 01 00 00 00 00 00 00 00 00 00 18 00 00 00 00 05 31 00
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF 03
```



Als Ergebnis erhältst du:

```
apiJobData("KOMBI60", "c_c_lesen", {DATA}, 46, "")
{DATA}: 46 Bytes
0000 : 01 01 01 01 00 00 00 00 00 00 00 00 00 18 00 00 00 00 05 31 00
0010 : 00 00 05 31 00 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
0020 : FF FF FF FF FF FF FF FF FF FF FF FF FF FF 03

Satz : 0
OBJECT = kombi60
SAETZE = 1
JOBNAME = c_c_lesen
VARIANTE = KOMBI60
JOBSTATUS =
UBATTCURRENT = -1
UBATTHISTORY = -1
IGNITIONCURRENT = -1
IGNITIONHISTORY = -1

Satz : 1
CODIER_DATEN = 46 Bytes
0000 : 01 01 01 01 00 00 00 00 00 00 00 00 00 18 00 00 00 00 05 31 00
0010 : 00 00 05 31 00 73 12 0D 00 00 36 00 67 00 98 00 00 00 00 00 00 00
0020 : CA 00 00 00 BC 01 78 03 35 05 F1 06 6E 03

JOB_STATUS = OKAY
_TEL_ANTWORT = 31 Bytes
0000 : 9B F1 60 62 31 05 73 12 0D 00 00 36 00 67 00 98
0010 : 00 CA 00 00 00 BC 01 78 03 35 05 F1 06 6E EC
```

Die roten Bytes sind die Codierdaten. Wie du siehst wurden die FF Platzhalter in dem übergebenen Buffer ersetzt.

```
01 01 01 01 00 00 00 00 00 00 00 00 00 18 00 00 00 00 05 31 00
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF 03
```

```
01 01 01 01 00 00 00 00 00 00 00 00 00 18 00 00 00 00 05 31 00
73 12 0D 00 00 36 00 67 00 98 00 CA 00 00 00 BC 01 78 03 35 05
F1 06 6E 03
```

Um nun Öltemp zu codieren musst du die blauen abändern:

```
73 12 0D 00 00 36 00 67 00 98 00 CA 00 00 00 BC 01 78 03 35 05
F1 06 6E
```

Dabei stehen die Werte 36 67 98 CA für die Skaleneinteilung. Der Wert 6E ist für KVA.

Für die Öltemperaturanzeige müssen wir die Skaleneinteilung abändern. Dafür müssen wir auf die gewünschte Temperaturanzeige einen Korrekturwert (46-48) addieren.

Um nun 50-150°C zu programmieren teilen wir die Skala wie folgt ein.
50°C, 75°C, 100°C, 125°C, 150°C

Öltemperatur °C	Operand	Korrekturwert		Dezimal	Hexwert
50°C	+	46	=	96	60
75°C				121	79
100°C				146	92
125°C				171	AB
150°C				196	C4

Temperatur + Korrekturwert = dezimaler Wert = in Hexwert umrechnen

Der Wert 6F schaltet Öltemperaturanzeige ein.

Das result der Berechnungen sieht wie folgt aus

```
73 12 0D 60 00 79 00 92 00 AB 00 C4 00 00 00 BC 01 78 03 35 05
0D 07 6F
```

0D und 07 werden einfach so übernommen. Derzeit ist dessen Funktion unbekannt

Dann die geänderten Werte in Buffer kopieren und den Job c_checksumme in T32 aufrufen und den Buffer mit den neuen Werten als Argument angeben. (Data anhaken nicht vergessen)

Job c_checksumme Argument:

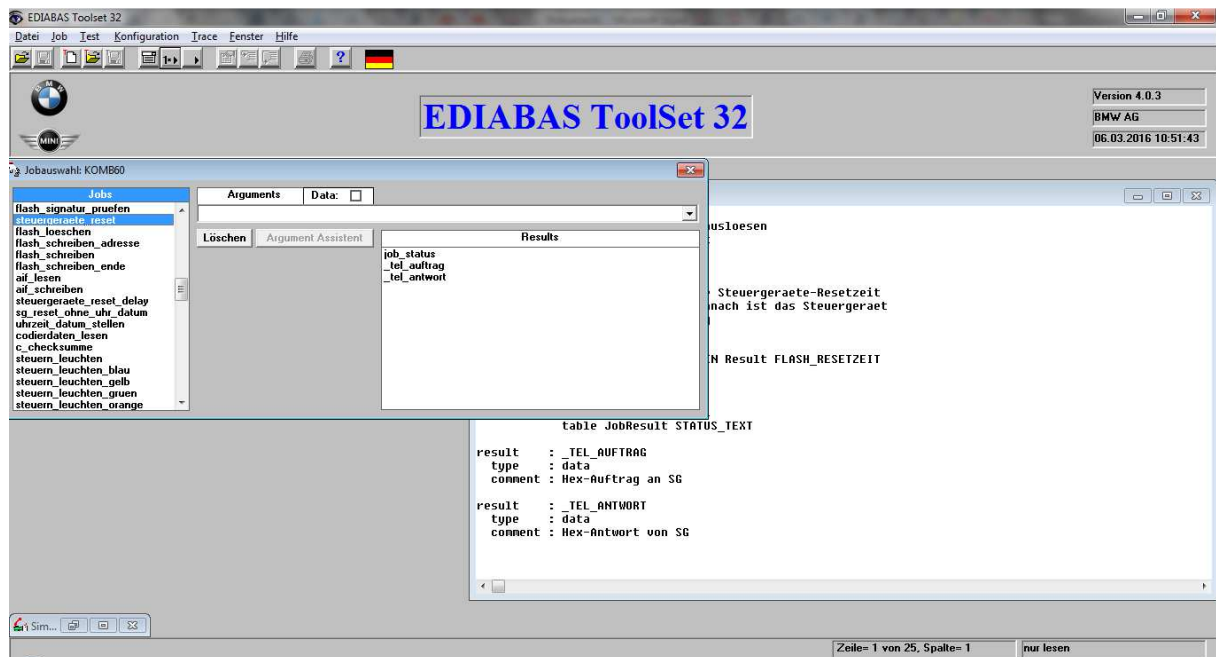
```
01 01 01 01 00 00 00 00 00 00 00 00 00 18 00 00 00 00 05 31 00
73 12 0D 60 00 79 00 92 00 AB 00 C4 00 00 00 BC 01 78 03 35 05
0D 07 6F 03
```

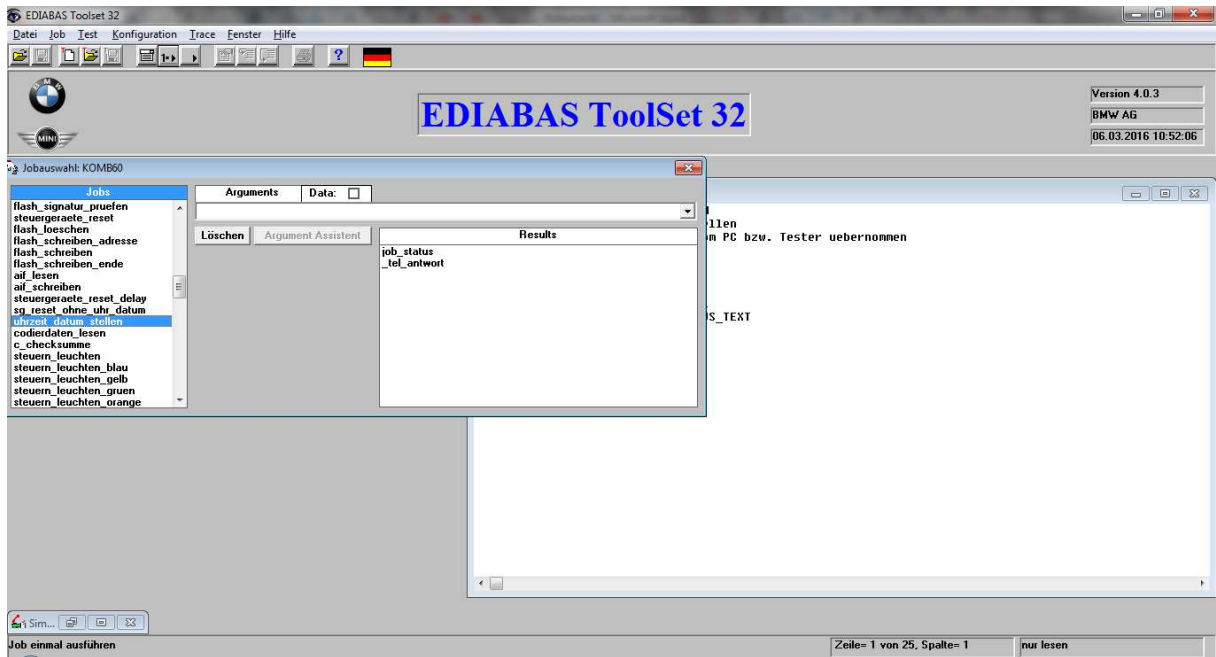
Als Ergebnis erhält du eine neue Checksumme, in meinem Fall 68

```
01 01 01 01 00 00 00 00 00 00 00 00 00 18 00 00 00 00 05 31 00
68 12 0D 60 00 79 00 92 00 AB 00 C4 00 00 00 BC 01 78 03 35 05
0D 07 6F 03
```

Dann wieder den gesamten Buffer als Argument kopieren und mittels Job c_c_Schreiben codieren, DATA anhaken nicht vergessen.

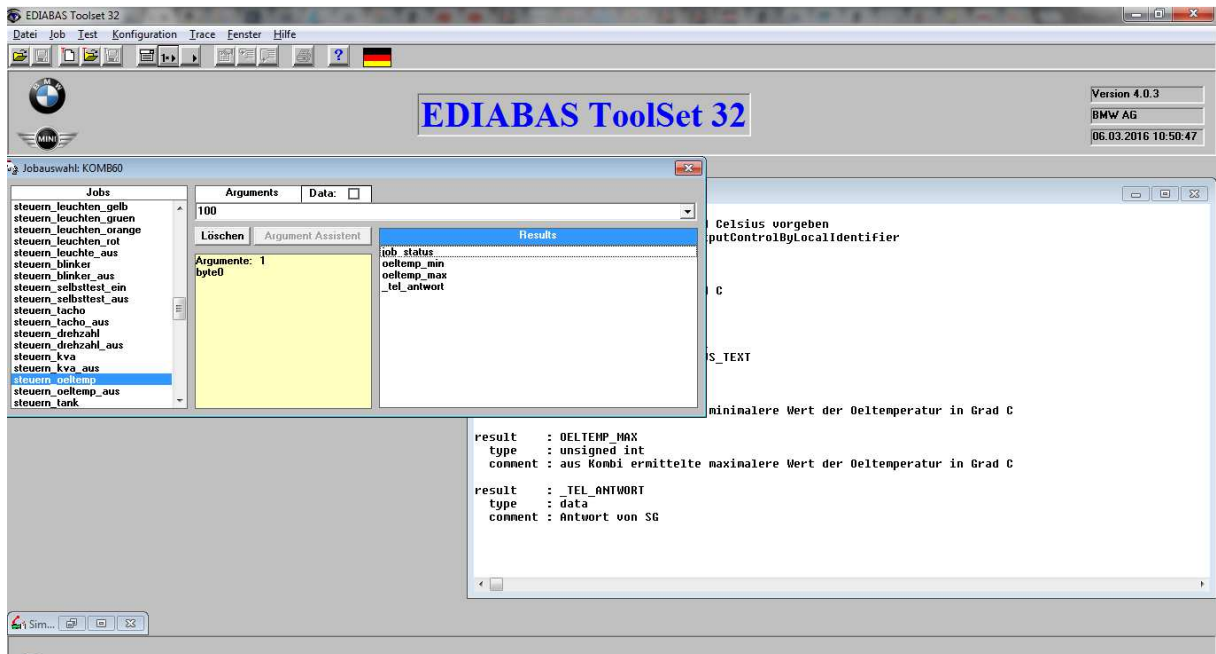
Dann Job steuergeraete_reset ausführen.





Bei Bedarf noch Datum und uhrzeit mittels Job uhrzeit_datum_stellen ausführen. Dadurch wird die Zeit vom Laptop übernommen.

Nun kann noch mittels Job steuern_oeltemperatur getestet werden, ob die Öltemperaturanzeige funktioniert



Als Argument die gewünschte Öltemp in °C angeben. Z.B. 100 und Job einmal ausführen.

Falls die Anzeige nicht genau stimmt, 100°C sollte bei 10L liegen, muss der Korrekturwert angepasst werden. In meinem Fall bin ich mit Korrekturwert 48 besser gefahren. Die Hexwerte sind dann anzupassen und die Checksumme muss erneut

ermittelt werden. Danach den neuen Buffer wieder mit c_c_schreiben und steuergeraete_reset durchführen.

Wenn alles korrekt. Job steuern_oeltemperatur_aus.

Danksagung

Besonderen Dank möchte ich a acolt und Hates richten (beide bwm-syndikat.de). Erst durch die Intensive Zusammenarbeit mit diesem beiden Forenmitgliedern, war es möglich das Geheimnis zu lüfenten

Offene Fragen

Für die beiden blauen Bytes fehlen noch die Erklärungen.

```
01 01 01 01 00 00 00 00 00 00 00 00 00 18 00 00 00 00 05 31 00  
73 12 0D 60 00 79 00 92 00 AB 00 C4 00 00 00 BC 01 78 03 35 05  
0D 07 6F 03
```

Vermutung 07 ist nur für KVA und bedeutet, dass die KVA erst ab 7 km/h anspringt.

0D ist noch völlig unklar. Ich vermute, dass es sich hierbei um die Dämpfung der Anzeige handelt.

EDIABAS Toolset 32

Datei Job Test Konfiguration Trace Fenster Hilfe

EDIABAS ToolSet 32

Version 4.0.3
BMW AG
06.03.2016 10:51:09

Jobauswahl: KOMB60

Jobs

- steuern_leuchten_gelb
- steuern_leuchten_gruen
- steuern_leuchten_orange
- steuern_leuchten_rot
- steuern_leuchte_aus
- steuern_blinker
- steuern_blinker_aus
- steuern_selbsttest_ein
- steuern_selbsttest_aus
- steuern_tacho
- steuern_tacho_aus
- steuern_drehzahl
- steuern_drehzahl_aus
- steuern_kva
- steuern_kva_aus
- steuern_oeltemp
- steuern_oeltemp_aus
- steuern_tank

Arguments Data:

Löschen Argument Assistant

Results

job_status
_tel_antwort

atur-Vorgabenodus wieder aus

S_TEXT

Zeile= 1 von 25, Spalte= 1 nur lesen