

## OmsiDeluxe

*Der etwas andere Omsi*



KOMSI 2.3-

Die Kommunikationsschnittstelle für OMSI

---



## Inhaltsverzeichnis

### Kapitel I – Allgemeines

Einleitung	1
Was ist KOMSI?	2
Rechtliches	3

### Kapitel II – Installation

Installation	4
--------------	---

### Kapitel III - KOMSI Server

Startbildschirm	5
Einstellungen	6
Hinweise zur Kommunikation zwischen KOMSI Server und Arduino	7

### Kapitel IV - KOMSI Client

Allgemein	9
Startbildschirm	10
Experimentiermodus	
Leuchtmelder	12
Messinstrumente (Tacho, Drehzahl etc.)	14
Überwachungsmonitor	15
Aufbau der OPL Datei	16

### Kapitel V - Arduino und Co.

Allgemeine Informationen	17
--------------------------	----

### Kapitel VI - Schnellstart / Inbetriebnahme

KOMSI ist geladen, was muss ich nun tun?	18
--	----

### Kapitel VII – sonstiges

nützliche Tools	19
Allgemeines und Danksagungen	28



## Einleitung

Nach einiger Zeit, Arbeit, Schweiß und durchgemachter Nächte ist es nun soweit. KOMSI - Die Kommunikationsschnittstelle für OMSI ist fertig!

Mein Ziel war es, eine Schnittstelle zu entwerfen die von allen genutzt werden kann und erweiterbar für eigene Ideen ist.

Was KOMSI ist und macht werde ich in den nächsten Kapiteln genauer erläutern.

Jedenfalls wünsche ich euch noch viel Spaß beim „Basteln“ und viel Erfolg für die Umsetzung eures Fahrerarbeitsplatzprojektes. Ich hoffe, dass die „Fan“-Gemeinde weiter wächst und wir aus den Erfahrungen von anderen Usern lernen können.

Euer  
Lars Jobst (Pulle) – für omsideluxe.de

P.S. Bitte bedenkt, dass ich nicht hauptberuflich Programmierer bin und mir alles Notwendige für das Plugin selbst angeeignet habe. Sollte etwas nicht auf Anhieb funktionieren, bitte ich um Nachsicht und Geduld. Ich werde schnellstmöglich versuchen, euch bei auftretenden Problemen zu helfen.

Meine Familie und mein Beruf gehen hier selbstverständlich vor!



## Was ist KOMSI?

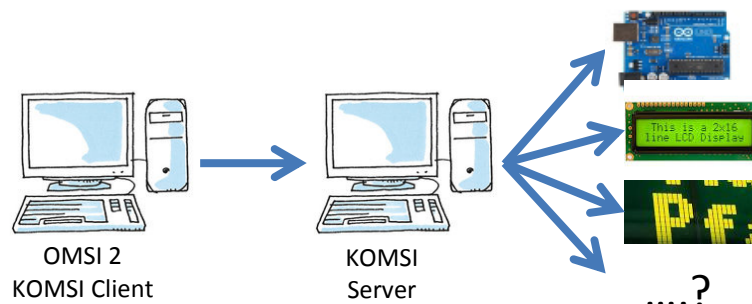
KOMSI besteht aus zwei Teilen. Zum einen der „Sendeteil“, dieser wird auf dem Spielerechner, auf dem auch OMSI 2 installiert ist (im späteren Verlauf KOMSI Client genannt) installiert. Zum anderen aus einem „Empfangsteil“ (im späteren Verlauf KOMSI Server genannt), dass auf einem weiteren, beliebigen Rechner installiert wird (siehe Abbildung 1). Steht ein zweiter Rechner nicht zur Verfügung, so kann selbstverständlich der KOMSI Server auch auf dem Spielerechner installiert und ausgeführt werden.

Die Datenübertragung via LAN/WLAN hat den Vorteil, dass schwächere Spielerechner entlastet werden und die eigentliche „Arbeit“ auf dem Empfangsrechner stattfindet. Von hier werden die Leuchtmelder und Co. gesteuert.

Unterstützt werden bis zu 32 Leuchtmelder, 8 Anzeigen (Tacho, Drehzahl etc.), ein Display für das IBIS-Gerät, ein Display für den Fahrscheindrucker und die Innenanzeige (nur möglich wenn sie über eine externes Programm (z.B. ibisserial.exe (Homepage: <http://cybox.ib-luehning.de/ibis/>) gesteuert werden können).

Im Installationspaket nicht vollständig enthalten, sind die notwendigen Arduinoskripte zum Betreiben der einzelnen zusätzlichen Komponenten (IBIS, Drucker etc.). Da diese ständig erweitert werden und in Bewegung sind, werde(n) ich/wir diese immer aktuell unter dem im Forum: [www.omsideluxe.de](http://www.omsideluxe.de) zum Download bereitstellen. Gleiches gilt natürlich auch für die eigentliche Software, das Plugin KOMSI und alle weiteren notwendigen Dateien bzw. Skripte.

Abbildung 1:





## Rechtliches

Das KOMSI Plugin und alle dazu gehörigen (im Installationspaket enthaltenen) Komponenten (Handbuch, Arduinoskripte, KTT Koms Test Tool, OFK OPL Creator for KOMSI etc..) sind mein, Lars Jobst, geistiges und körperliches Eigentum und werden im folgenden Text unter Software zusammengefasst.

Die Software darf nicht verändert oder über andere Plattformen als <http://www.omsideluxe.de> (z.B. andere Foren, Downloadportale, Weitergabe durch User) verteilt werden.

Für etwaige Schäden die durch die Nutzung meiner Software entstehen, werde ich keine Haftung übernehmen. Der Nutzer dieser Software hat keinen Anspruch auf Schadenersatz.

Die Software ist kostenlos und darf nicht kommerziell verwendet bzw. angeboten werden.

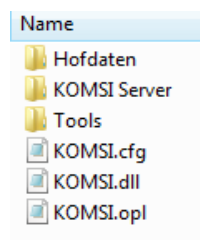
© 2014 Lars Jobst



## Installation und Erläuterung der einzelnen Komponenten

Die Installation von KOMSI ist denkbar einfach und sollte mit 2. Arbeitsschritten erledigt sein.

Im ZIP Archiv (KOMSI.ZIP) findet ihre folgende Dateien und Unterverzeichnisse die für den Betrieb von KOMSI notwendig sind.



### Schritt 1:

Kopiert die Dateien aus dem Verzeichnis Plugins in euer Spieleverzeichnis von OMSI 2. Steam\..Common\OMSI\Plugins\.. Ihr solltet nun die Verzeichnisse Hofdaten, KOMSI Server und Tools sowie die Dateien KOMSI.dll, KOMSI.cfg und KOMSI.opl sehen.

Wofür die einzelnen Dateien genau sind und welche Funktion sie haben beschreibe ich im späteren Verlauf dieses Handbuchs.

### Schritt 2:

#### *2 Rechner Lösung (KOMSI Server läuft über einen zweiten Rechner)*

Solltet ihr einen zweiten Rechner für den Empfangsteil nutzen, so kopiert das Verzeichnis KOMSI Server auf diesen. Wohin ist in dem Fall egal, da es sich um eine eigenständig ausführbare Datei handelt.

#### *1 Rechner Lösung (KOMSI Server läuft auf dem gleichen Rechner wie OMSI)*

Kopiert das Verzeichnis KOMSI Server Verzeichnis auf den Spielerechner. Speicherort ist auch hier egal (oder belasst es im Plugin Verzeichnis).

KOMSI ist nun betriebsbereit. Das heißt aber nicht, dass jetzt alle Melder usw. funktionieren. Hier ist natürlich noch ein wenig Arbeit notwendig. Werden eines oder mehrere Arduinos benutzt, so kann man den Skriptcode aus dem Arduinoverzeichnis auf die Karte(n) hochladen. Wichtig ist hierbei: Überprüft die vergebenen COM-Ports sehr genau, damit nicht irgendein Schaden durch falsch angeschlossene Hardware etc. entsteht.



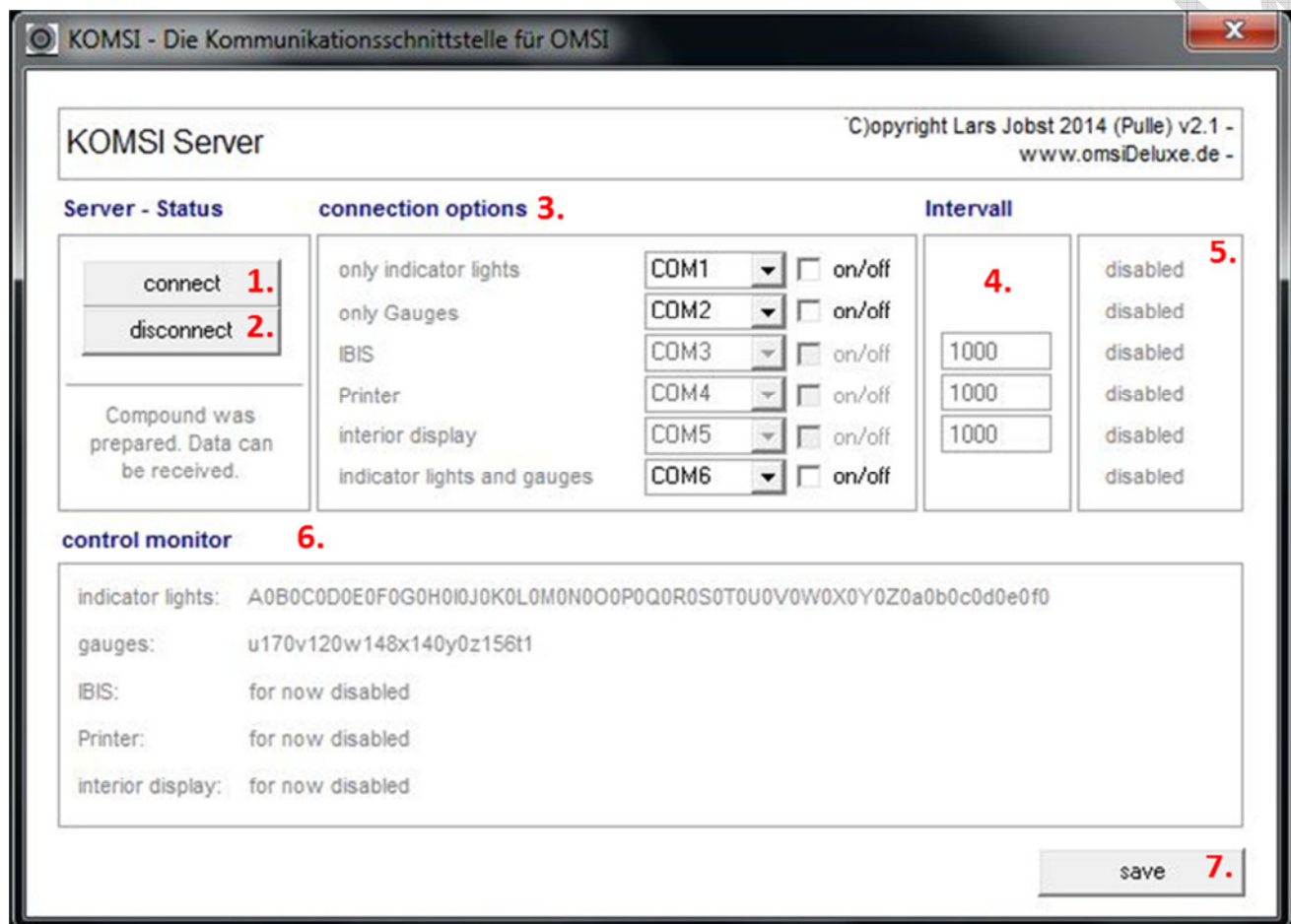
Mehr dazu im **Kapitel IV KOMSI Client**.



## Startbildschirm

Durch einen Doppelklick auf die KOMSI Server.exe erhält ihr folgenden Startbildschirm.

Abbildung 2



Nach der Ersteinrichtung (alle Einstellungen vorgenommen und gespeichert) reicht es aus den KOMSI Server zu starten und einen entsprechenden COMPORT (Anschluss Arduino) auszuwählen.

### NEU mit Update 2.1:

Es ist nicht mehr erforderlich die Schaltfläche (1) nach dem Start von KOMSI Server.exe zu bedienen. Dies erfolgt nun automatisch.



**Einstellungen** (siehe Abbildung 2; Seite 5)

Für die Ersteinrichtung erkläre ich hier die einzelnen Auswahlpunkte:

1. Verbindung wird hergestellt. Daten vom KOMSI Client können empfangen werden.
2. Verbindung wird getrennt. Es werden keine Daten mehr empfangen.
3. Hier wählt man die zu benutzenden Ports aus. Mit einem Klick auf die Schaltfläche – save – (7) werden die Daten für den nächsten Start gespeichert. Eine erneute Auswahl ist nicht mehr notwendig.

Warum kommen hier Leuchtmelder und Messinstrumente doppelt vor?

Folgende Varianten gibt es zu den Punkten Leuchtmelder und Messinstrumente:

Möglichkeit 1 - die Leuchtmelder und Messinstrumente jeweils über ein Arduinoboard anzusteuern oder

Möglichkeit 2 - über den letzten Punkt Leuchtmelder und Messinstrumente mit einem Arduinoboard zu steuern.

Ein Klick auf die Checkbox (Haken) öffnet eine Verbindung zu Arduino und Co.

4. Je nach Hardware (Display und Innenanzeige) ist es ratsam, eine Art Zeitintervall einzubauen. Eine Innenanzeige kann unterschiedlich auf gesendete Daten reagieren. Sendet man die Daten zu schnell, hat die Innenanzeige keine Zeit zum Initialisieren und verschluckt die empfangen Daten. Der Intervall wird in Millisekunden angegeben. 1.000 ms = 1 Sekunde.



5. Ist ein Statusfenster, in dem angezeigt wird, ob eine Verbindung zum Arduino und Co. besteht.
6. Im Überwachungsmonitor sieht man die empfangenen Daten vom KOMSI Client. Werden im Betrieb keine Daten angezeigt, besteht keine Verbindung zum KOMSI Client.

Die hier gebildeten Strings werden an die angeschlossenen Arduino und Co. gesendet. Die Funktion des Überwachungsmonitors ist hilfreich, wenn z.B. ein Skript für ein Arduino programmiert werden soll.

7. Alle Einstellungen werden gespeichert (COMPort, Intervall).



### Hinweise zur Kommunikation zwischen KOMSI Server und Arduino

Dem Installationspaket liegen drei Arduinoskripte bei.

Zum Betrieb der Anzeigen wie Tacho, Drehzahlmesser usw. mit der Tone bzw. Hz Methode, nutzt bitte den Code im Verzeichnis Leuchtmelder und Messinstrumente Tone. Dieser funktioniert ausschließlich ab Arduino Mega oder Klone aufwärts. Ein Uno reicht hier nicht mehr aus.

Zum Betrieb der Anzeigen mit Modellbauservos, nutzt bitte den Code im Verzeichnis Leuchtmelder und Messinstrumente Servo. Dieser funktioniert ausschließlich ab Arduino Mega oder Klone aufwärts.

Solltet ihr die Leuchtmelder und Anzeigen getrennt voneinander nutzen wollen, ist ein Eingriff in den Arduinocode notwendig. Sollte dies verlangt werden, so kann ich einen modifizierten Code im Forum bereitstellen. Um diese Option nutzen zu können, müssen die beiden ersten Punkte im KOMSI Server ausgewählt werden. (siehe Abbildung 2; Seite 5).

### WICHTIG SICHERHEITSHINWEISE AUF DER NÄCHSTEN SEITE BEACHTEN!



#### NEU mit Update 2.2:

Mit dieser Version biete ich nun einen Beispielsketch zum Betrieb eines Fahrscheindruckers an. Der „Printer“-sketch ist für ein Arduino DUE konzipiert da dieser die serielle Kommunikation als auch die Emulation einer Tastatur unterstützt.

Betrachtet diesen Sketch als eine Art Grundlagenvorlage. Bestandteile sind:

- Matrixtastatur
- LCD
- serielle Kommunikation (String Variante)
- Tastatur Emulation

Mit Bestandteilen dieser Vorlage lässt sich auch leicht das IBIS umsetzen.



**WICHTIG! ACHTUNG!**

Bevor irgendein Code auf einem Mega mit KOMSI getestet werden soll, vergewissert euch dass:

- das Arduino oder Klone mit dem richtigen Comport angemeldet sind!?
- die externe Stromversorgung angeschlossen und richtig mit + und – verbunden ist!?
- externe Stromversorgung auch tatsächlich für den Betrieb mit Arduino, Relais usw. geeignet ist und oder keine direkte Verbindung mit dem Arduino besteht(Spannung)!?



Es empfiehlt sich immer erst eine „trockenübung“ mit dem im Installationspaket enthaltenen KTT - KOMSI Test Tool zu machen, **bevor** KOMSI tatsächlich komplett gestartet wird. Mehr zu den mitgelieferten Tools findet ihr in **Kapitel VII – Sonstiges – nützliche Tools.**

Bei Nichtbeachtung der durch Arduino und Co. vorgegebenen Sicherheitshinweise und Richtlinien ist eine Zerstörung möglich!

**FÜR EVENTUELL ENTSTEHENDE SCHÄDEN BEIM BETRIEB VON KOMSI AM PC, ARDUINO USW. ÜBERNEHME ICH KEINE HAFTUNG. ALLES AUF EIGENE GEFAHR. BEACHTET DAHER BITTE IMMER DIE SICHERHEITSHINWEISE DER JEWELIGEN HERSTELLER!**

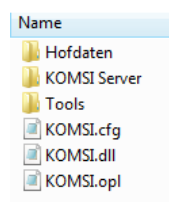




## Allgemein

Nach dem das Pluginverzeichnis entsprechend meiner Installationsbeschreibung (siehe Kapitel II) vorbereitet wurde, ist der KOMSI Client betriebsbereit. Mit jedem Start von OMSI 2 wird nun auch der KOMSI Client gestartet.

Bevor wir uns dem Startbildschirm widmen, möchte ich kurz noch auf die Dateien im Plugin Verzeichnis von OMSI 2 eingehen. Im Idealfall sieht das Verzeichnis nun so aus:



### Verzeichnis Hofdaten:

Das Verzeichnis Hofdaten beinhaltet Textdateien, die für die Darstellung von Zeichen auf einem ...x16 Zeichen Display (z.B. IBIS MAS1) dargestellt werden. Hier können die Haltestellennamen manuell hinterlegt werden.

Warum? Als Beispiel: In OMSI 2 fahre ich einen Bus mit einem IBIS MAS3. Dies kann 24 Zeichen in einer Zeile darstellen. Meine originale Hardware aber nur 16 Zeichen. Würde ich nun die neue Funktion aus OMSI nutzen und den angezeigten String auf meinem Display anzeigen, wird dieser nicht ganz dargestellt. Die Funktion - use original Hofdatei - kann manuell über den Startbildschirm gesteuert werden. (siehe Abbildung 3; ab Seite 10; Punkt 3)

### KOMSI.cfg:

In dieser Datei werden alle Einstellungen, die im KOMSI Client getätigt werden, abgespeichert. Bitte nicht manuell in der Datei Veränderungen vornehmen, da sonst eine reibungslose Funktion nicht mehr gewährleistet ist.

### KOMSI.dll:

Ist das eigentliche Herz von KOMSI und wird automatisch mit OMSI gestartet.

### KOMSI.opl:

Hier werden die Variablen benannt bzw. hinterlegt die ausgelesen und durch KOMSI weiterverarbeitet werden sollen. Funktion und Aufbau erläutere ich im

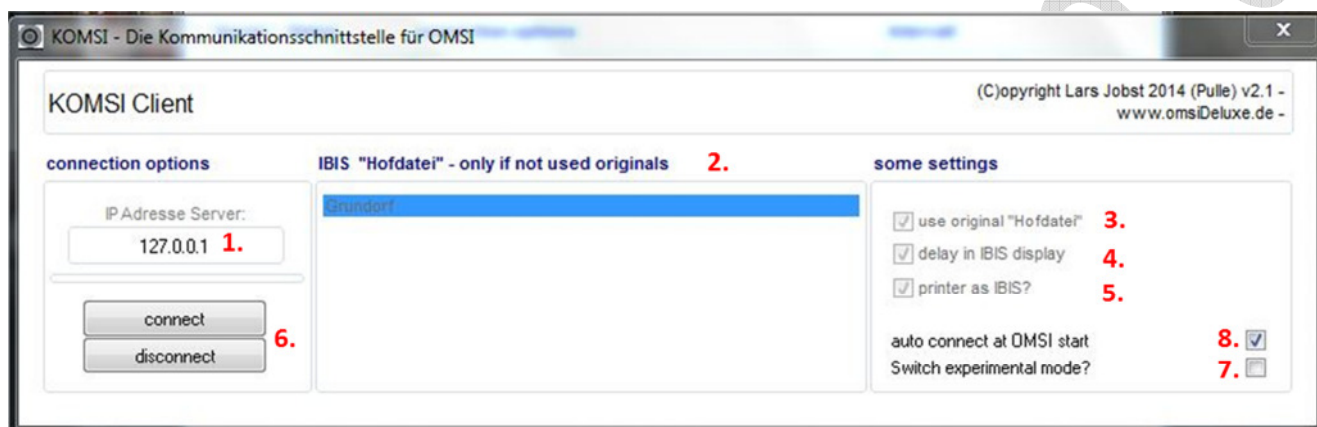
**Kapitel VII – sonstiges.**



## Startbildschirm

Mit dem Start von OMSI 2 und des KOMSI-Servers, wird KOMSI automatisch geladen. Nach der Ersteinrichtung (alle Einstellungen vorgenommen und gespeichert) muss vor dem „spielen“ nur noch die Verbindung hergestellt (6) werden.

Abbildung 3:



1. IP Adresse des Rechners auf dem der KOMSI Server installiert wurde. Ist der Spielrechner auch Empfänger, (KOMSI Server auf Spielerechner installiert) dann ist die IP Adresse 127.0.0.1.
2. Mit der IBIS Vorwahl können eigene Textdateien zur Ausgabe der Haltestellennamen genutzt werden. (Siehe Anmerkung auf Seite 9)
3. Wird hier der Haken entfernt, wird die in Punkt 2 genannte Textdatei zur Ausgabe der Haltestellennamen genutzt. Ist der Haken gesetzt, übermittelt KOMSI Client den aktuellen (in OMSI auf IBIS Gerät abgebildeten) Namen.
4. Auch hier ist es abhängig davon, welchen Displaytyp man verwendet. Bei einem IBIS MASI wird in der Regel keine Verspätungsminute angezeigt. Daher Haken weg. Die Variable wird trotzdem an den KOMSI Server gesendet. Passt man also den Arduinocode nach den eigenen Wünschen an kann auch eine Verspätung auf dem IBIS MASI angezeigt werden.
5. Drucker als IBIS: Diese Funktion habe ich eingebaut, um verschiedene Druckertypen bedienen zu können. In einigen Regionen wurde der Drucker auch als „IBIS“ genutzt. Besitzt man so wie ich einen Almexdrucker mit 2x40 Zeichen, lasse ich den Haken an. Der KOMSI Server empfängt neben den Druckerdaten (Ticket und Preis) auch alle IBIS Daten. Verwendet man z.B. „nur“ ein 1X40 Zeichen Display, dann deaktiviert man einfach diese Funktion. Es werden dann nur noch Ticket und Preis übergeben.
6. Hier wird die Verbindung zwischen KOMSI Client und KOMSI Server aktiviert.



### Startbildschirm

7. Der wohl wichtigste Punkt. Der Experimentalmodus. Aktiviert man diesen Haken erhält man Zugriff auf ein Untermenü in dem die Einstellungen für Leuchtmelder, Messinstrumente, IP Adresse und weitere Themen vorgenommen werden können.

Hierzu mehr ab **Seite 12 – Experimentiermodus**.

### NEU mit Update 2.1:

8. Wird dieser Haken aktiviert, startet die Kommunikation zwischen KOMSI Client und KOMSI Server automatisch. Es ist nun nicht mehr erforderlich nach dem Start von OMSI die Connect Schaltfläche zu betätigen.

**WICHTIG:** Der KOMSI Sever muss zwingend vor dem Aufruf von OMSI gestartet werden, da sonst keine Verbindung zustande kommen kann.

Ebenfalls neu sind nun folgende Funktionen:

- Wird OMSI beendet schaltet der KOMSI Server automatisch alle Anzeigen und Leuchtmelder aus. Die Anzeigen fallen auf die vorher definierten Nullwerte und die Leuchtmelder gehen aus.
- Wird die Elektrik im Fahrzeug ausgeschaltet, werden wie oben bereits beschrieben ebenfalls die Leuchtmelder ausgeschaltet und die Anzeigen auf den Nullwert geschaltet.

### NEU mit Update 2.2:

Es werden nun endlich auch die notwendigen Daten für IBIS, Drucker und Co. übergeben. Ein Bespielsketch findet ihr im Installationsverzeichnis.



## Leuchtmelder

Nach dem der Experimentiermodus eingeschaltet wurde, erhält man folgenden Bildschirm:

Abbildung 4:

### 1. Leuchtmelder:

Hier können allgemeine Einstellungen vorgenommen werden. Der Wert in Klammern entspricht dem Port am Arduino. Die Bezeichnungsfelder dienen als Erinnerungshilfe und haben keinen Einfluss auf die OPL Datei.

Status kann nur 0 und 1 sein. 0 = Leuchtmelder aus; 1 = Leuchtmelder an. Über die Hakenfunktion kann ein Leuchtmelder kurzfristig ausgeschaltet werden. Durch einen Klick auf die Schaltfläche Speichern werden diese Daten übernommen und bei einem nächsten Start von OMSI stehen die vorher gespeicherte Auswahl wieder zur Verfügung.

Warum sind die Felder mit der Portbezeichnung 49,50,51 und 53 geschützt? Da nicht in jedem Bus ein Freigabemelder (wie z.B. im O405) existiert und trotzdem diese Funktion genutzt werden soll, war es notwendig eine feste Programmierung für diesen Fall einzubauen.



### **Leuchtmelder**

Der Status dieses Melders ist der einzige, der einen Wert  $> 1$  haben kann. Daher ist es zwingend notwendig, die in OPL Datei hinterlegten Werte für Tür1 ff. an dieser Position zu belassen.

Mehr dazu auf Seite 16. Dort werde ich den Aufbau der OPL Datei näher beschreiben.



## Messinstrumente

Durch einen Klick auf den Karteireiter Messinstrumente erhält man folgenden Bildschirm:

Abbildung 5:

The screenshot shows the 'gauges' tab selected. It contains a table with sensor data and a checkbox at the bottom.

Port ()	Bezeichnung	Servostellwerte			maximaler Anzeigewert
		aktuell	minimal	maximal	
(01)	Druckanzeige 1	65	170	10	10
(02)	Druckanzeige 2	65	170	10	10
(03)	Motortemperatur	105	120	10	135
(04)	Öldruck	112	148	18	5
(05)	Tank	42	140	24	100
(06)	Geschwindigkeit	0	0	179	95
(07)	Kühlwasser	143	156	50	150
(08)	Drehzahl	39	1	179	2400

☒ Values with factor convert (min / max)?

### 2. Messinstrumente

Ursprünglich war hier eine Unterscheidung der Ansteuerungsmethode in Servo und Hz./Tone vorgesehen.

Nach einigen Tests hat sich herausgestellt das die „Servostellwert-Berechnung“ nicht nur für Servos sondern auch für die andere Methode nutzbar ist. Der Einfachheit halber erläutere ich die Felder wie sie für die Servotechnik vorgesehen war.

**Servostellwert minimal:** Hier wird der kleinstmögliche Stellbereich des Servos eingetragen.

**Servostellwert maximal:** Hier wird der größtmögliche Stellbereich des Servos eingetragen.

**Maximaler Anzeigewert:** Damit ist der maximal Wert auf der Anzeige gemeint. z.B. Öldruck = 5, Druckanzeige = 10 (bar)

Mit diesen Angaben ermittelt KOMSI nun automatisch den aktuellen Stellwert.

Wird der Haken entfernt, werden die ausgelesenen OMSI Werte 1:1 an den KOMSI Server übergeben.



## Überwachungsmonitor

Abbildung 6:



3. Der Überwachungsmonitor gibt Auskunft über die ausgewerteten Daten und zeigt an, welche Daten an den KOMSI Server übergeben werden. Der Überwachungsmonitor eignet sich hervorragend für diejenigen, die eigene Arduinosketches entwickeln wollen.



### Aufbau der OPL Datei

Was ist eine OPL Datei und wozu benötige ich diese?

Eine OPL Datei veranlasst das geschriebene Plugin zur Laufzeit von OMSI geladen und ausgeführt zu werden. Desweiteren werden in der OPL Datei Variablen benannt, die zur Laufzeit von OMSI ständig mit Inhalten gefüllt werden. Nehmen wir als Beispiel die Variable „Haltewunsch“. Wird im Fahrzeug der Haltestellenwunsch betätigt, wird die Variable „Haltewunsch“ mit der Zahl 1 gefüllt. Erlischt die Anzeige nachdem die Fahrgäste ausgestiegen sind, bekommt die Variable den Wert 0.

Variablen können frei vergeben werden, solange sie in irgendeiner Form in den Scriptdateien eines Fahrzeuges vorkommen. Um das Handling dieser Datei zu erleichtern, habe ich den OFK - OPL Creator for KOMSI „erfunden“. Mehr hierzu findet ihr im **Kapitel VII – sonstiges – nützliche Tools**.

### WICHTIG:

Die OPL Datei ist sehr empfindlich. Einige Variablen musste ich fest definieren bzw. festlegen. Daher bitte ich euch unbedingt den OPL Creator zu nutzen, um mögliche Fehlerquellen schon im Vorfeld auszuschließen.



## Allgemeine Informationen

Was ist eigentlich ein Arduino?

Ein Arduino ist vom Grundsatz her ein Mikrocontrollerboard, dass es in verschiedenen Ausführungen und Variationen gibt. Wie bei fast jedem elektronischen Produkt, gibt es davon auch Klone, die über bekannte Einkaufsseiten bezogen werden können. Der Einfachheit halber werde ich mich aber immer auf ein Arduino beziehen.

Ein Arduino UNO hat Beispielsweise 14 digitale Ein-/Ausgänge (6 davon können als PWM Kanäle genutzt werden), 6 analoge Eingänge, ein 16 MHz Quarz, USB Schnittstelle, einen 6 poligen ISP Anschluss und einen Rückstellknopf.

Ein Arduino MEGA hat Beispielsweise 52 digitale I/O-Pins (von denen 14 als PWM-Ausgänge verwendet werden können), 16 analoge Eingänge, 4 UARTs (serielle Hardware-Schnittstellen), einen 16 MHz Kristalloszillator, eine USB-Schnittstelle, einen Stromanschluss, einen ICSP-Header und einen Rückstellknopf.

Je nach Bauvorhaben eures FAP Arbeitsplatzes (z.B. für das Betreiben mit KOMSI) reicht ein UNO eventuell nicht aus. Überlegt euch daher im Vorfeld eures Projektes was ihr alles haben und umsetzen wollt, damit direkt die richtige Hardware geordert werden kann. Nichts ist ärgerlicher als wenn man sein Projekt unterbrechen muss um neue Hardware zu bestellen.

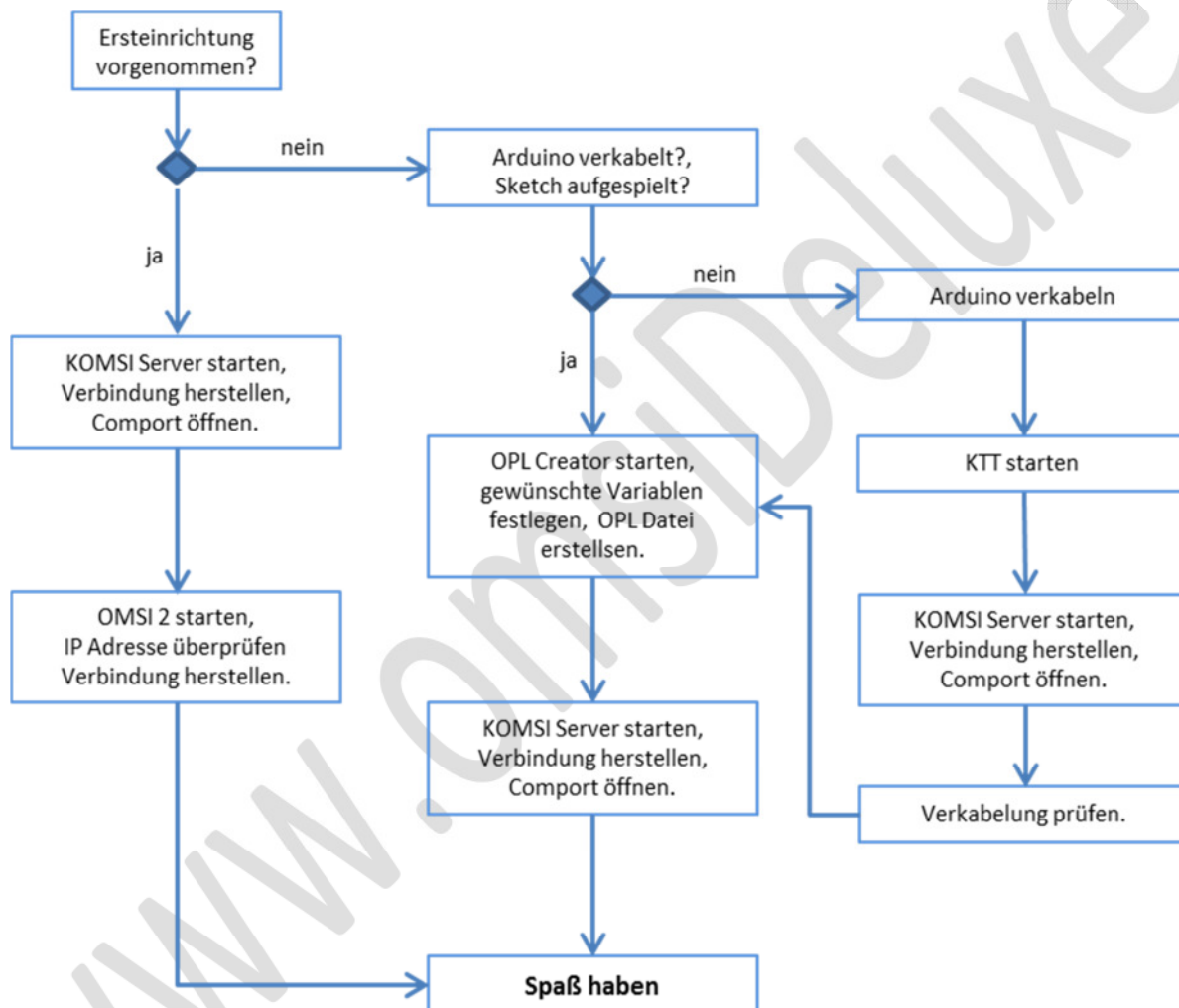
Wie ihr den Arduinocode (auch Sketch genannt) verarbeiten, bearbeiten und modifizieren könnt, könnt ihr auf der Homepage [www.Arduino.cc](http://www.Arduino.cc) nachlesen. Die ganze Prozedur ist recht einfach und wird mit einer „Arduinosoftware“ vollzogen.

Ihr könnt auch gerne eigene Sketche entwerfen und der Community auf [www.omsideluxe.de](http://www.omsideluxe.de) zur Verfügung stellen. Wir werden dafür einen eigenen Bereich einrichten, in dem ihr sie zur Verfügung einstellen könnt. Wir würden uns freuen.



### KOMSI ist geladen was muss ich nun tun?

Auf den vorherigen Seiten habe ich ja schon beschrieben, was alles mit KOMSI gemacht werden kann. Wahrscheinlich auch in vielen Dingen sehr umfangreich und eventuell zu komplex. Daher habe ich folgendes Schaubild als eine Art Kurzzusammenfassung entworfen.





## nützliche Tools

Dem Installationspaket liegen einige nützliche Tools bei:

- KTT - KOMSI Test Tool und ein
- OPL Creator for KOMSI.

### KTT - KOMSI Test Tool

Das KOMSI Test Tool ist eine „standalone“ Variante des KOMSI Client. Mit diesem werden OMSI Signale simuliert ohne OMSI ständig starten bzw. laufen lassen zu müssen.

Ein Beispiel:

Der Spielerechner mit OMSI 2 steht im Wohnzimmer, das Dashboard (FAP) aber im Keller auf der Werkbank. Wie also die Verkabelung testen ohne den Spielerechner in den Keller zu schleppen oder das Dashboard in das Wohnzimmer?

Eine Lösung wäre, dass KTT – KOMSI Test Tool sowie den KOMSI Server auf einen Laptop zu kopieren, diesen auf die Werkbank zustellen und direkt Vorort die Verkabelung zu testen.

Wie das KTT - KOMSI Test Tool aufgebaut ist und was es kann wird auf **Seite 20** beschrieben.

### OFK - OPL Creator for KOMSI

Mit dem OPL Creator werden OPL Dateien für KOMSI erstellt. Eine OPL Datei liest vorher definierte Variablen in den Speicher und füllt diese nachdem Start von OMSI mit Melder Daten usw. auf.

Da es für jeden Bus unterschiedliche Variablen geben kann, bietet der OPL Creator für KOMSI eine Möglichkeit verschiedene Profile zu speichern und nach Bedarf zu aktivieren. Die Profile können auch in die KOMSI.cfg Datei geschrieben werden. Als Ergebnis dieser Aktion sieht man nach dem Start von OMSI und KOMSI Client, in den Bezeichnungsfeldern im Klartext die vorher ausgewählten Leuchtmelder. Mehr hierzu auf **Seite 22**.



## nützliche Tools - KTT - KOMSI Test Tool

KTT - KOMSI Test Tool

(C)opyright Lars Jobst 2014 (Pulle)  
v1.4 - www.omsideluxe.de -

**Indicator Lights 1.**

(22) A	<input type="checkbox"/>	(30) I	<input type="checkbox"/>	(38) Q	<input type="checkbox"/>	(46) Y	<input type="checkbox"/>
(23) B	<input type="checkbox"/>	(31) J	<input type="checkbox"/>	(39) R	<input type="checkbox"/>	(47) Z	<input type="checkbox"/>
(24) C	<input type="checkbox"/>	(32) K	<input type="checkbox"/>	(40) S	<input type="checkbox"/>	(48) a	<input type="checkbox"/>
(25) D	<input type="checkbox"/>	(33) L	<input type="checkbox"/>	(41) T	<input type="checkbox"/>	(49) b	<input type="checkbox"/>
(26) E	<input type="checkbox"/>	(34) M	<input type="checkbox"/>	(42) U	<input type="checkbox"/>	(50) c	<input type="checkbox"/>
(27) F	<input type="checkbox"/>	(35) N	<input type="checkbox"/>	(43) V	<input type="checkbox"/>	(51) d	<input type="checkbox"/>
(28) G	<input type="checkbox"/>	(36) O	<input type="checkbox"/>	(44) W	<input type="checkbox"/>	(52) e	<input type="checkbox"/>
(29) H	<input type="checkbox"/>	(37) P	<input type="checkbox"/>	(45) X	<input type="checkbox"/>	(53) f	<input type="checkbox"/>

Order from left to right (Port, Value Trigger, Hook = on) **6.** all / none

**Gauges 2.**

Trigger Value	Port	Place Value	min. Value	max. Value
t	free	0	100	120
u	(3)	0	50	55
v	(6)	0	50	55
w	(10)	0	50	55
x	(11)	0	50	55
y	(9)	0	50	55
z	(5)	0	50	55

Intervall (ms): 100 **7.** Start / Stop

IP Address -> KOMSI Server **3.**  
127.0.0.1

Connect **4.**  
Disconnect  
Save settings **5.**

Local  
127.0.0.1

## 1. indicator lights:

Über die Checkboxfelder (Haken) kann man einzelne Ports des Arduinos steuern. Grundvoraussetzung für alle hier aufgeführten Möglichkeiten ist, dass der KOMSI Server zuvor gestartet und die Verbindung hergestellt wird.

## 2. Gauges:

Hier besteht die Möglichkeit die angeschlossenen Messinstrumente zu testen. Im Feld min. value ist die Kleinstmögliche und im Feld max. value die Größtmögliche Stellgröße einzutragen. Nach einem Klick auf die Schaltfläche (7) Start / Stop wird der definierte Bereich durchlaufen. Diese Werte werden im Feld Place value übernommen und automatisch an den KOMSI Server gesendet. **BITTE SICHERHEISHINWEIS NÄCHSTE SEITE BEACHTEN!**

## 3. IP Adresse des Rechners auf dem der KOMSI Server installiert wurde. Ist es der gleiche Rechner dann 127.0.01.

## 4. Verbindung herstellen bzw. trennen.

## 5. Voreinstellungen werden gespeichert.

## 6. Alle Haken Felder gleichzeitig aktivieren bzw. deaktivieren.

## 7. Der Intervall legt fest, mit welcher Geschwindigkeit die in Punkt 2 genannten Werte durchlaufen werden. 1.000 ms = 1 Sekunde.





## nützliche Tools

### **SICHERHEITSHINWEIS beim Betrieb von KTT – KOMSI Test Tool**

Bevor die Schaltfläche Start / Stop (wie auf Seite 20; Punkt 2 beschrieben) geklickt wird, vergewissert euch, dass:

1. Der minimale und maximale Stellbereich richtig angegeben ist.
2. Um die möglichen Stellwerte herauszufinden, fangt mit einem Wert von 50 an (min. und max.). Klickt dann erst die Schaltfläche.
3. Erhöht die Werte schrittweise bis ihr den möglichen Stellbereich ermittelt habt.



Warum?

Soll eine Anzeige einen nicht möglichen Stellbereich anzeigen, kann es zur Zerstörung der Instrumente kommen. Das ist bei der Benutzung von Servos extrem wichtig, da sonst die auf dem Servo fixierte Nadel den Stellbereich nicht ausführen kann und den Servo mit dieser mechanischen Begrenzung zerstören kann.



**nützliche Tools – OFK – OPL Creator for KOMSI**

Wofür benötige ich OFK?

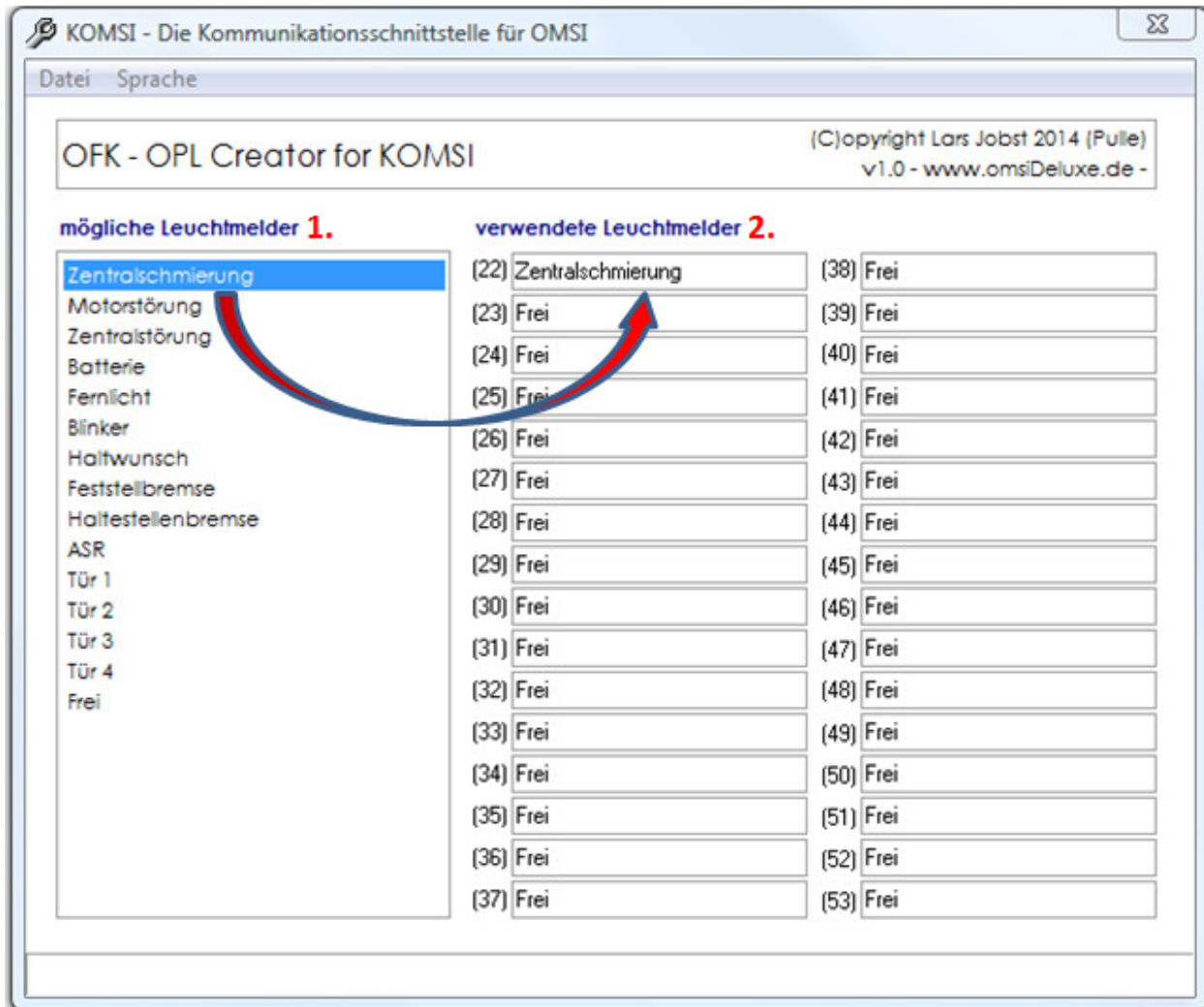
Mit dem OFK können per Drag&Drop-Verfahren im Handumdrehen eigene OPL Dateien erstellt werden, die sofort mit KOMSI einsatzbereit sind. Vorher erstellte Profile können jederzeit wieder geladen und in eine OPL Datei umgewandelt werden. Dies hat den Vorteil, dass für unterschiedliche Fahrzeuge unterschiedliche Profile erstellt werden können.

Was eine OPL Datei ist und macht habe ich auf Seite 22 bereits erklärt. Dies sei nur noch mal für die „Schnellleser“ gesagt, die diesen Abschnitt eventuell übersprungen haben. :D

Auf den folgenden Seiten erkläre ich kurz die Funktions- und Vorgehensweise anhand einzelner Screenshots.



## nützliche Tools – OFK – OPL Creator for KOMSI



Per Drag&Drop Methode können mögliche Variablen, den gewünschten Ports zugewiesen werden. Die Liste (links auf dem Screenshot) ist erweiterbar. Im Unterverzeichnis dieses Tools befindet sich eine OPL Content.db die mit einem Texteditor einfach erweiterbar ist. (Ports in Klammern)

Bevor ich euch die einzelnen Schritte zur Erstellung einer funktionierenden OPL Datei erläutere, möchte ich kurz auf das Menüband eingehen.



## nützliche Tools – OFK – OPL Creator for KOMSI

Menüband des OFK



1. Wurde ein Profil bereits angelegt, kann dieses jederzeit wieder geladen werden.
2. Wurden Leuchtmelder -> Ports zugewiesen, wird eine OPL Datei erstellt. Im folgenden Untermenü kann zusätzlich die Option gewählt werden, ob die definierten Namen auch für KOMSI Client zur Verfügung stehen sollen. (Hier geht es nur um die Namen der Leuchtmelder und nicht um die Variablenbezeichnung)
3. OFK Beenden
4. Mögliche Sprachen Deutsch und Englisch.

Auf den folgenden Seite werde ich Schritt für Schritt erklären wie mit dem OFK gearbeitet werden kann.



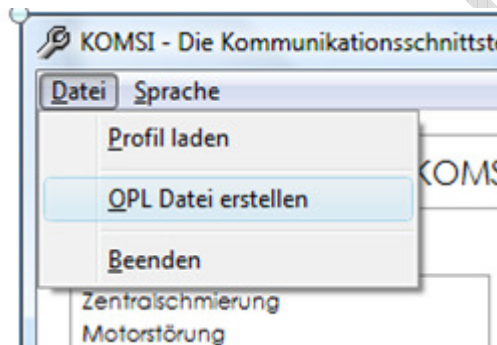
## nützliche Tools – OFK – OPL Creator for KOMSI

Schritt für Schrittanleitung

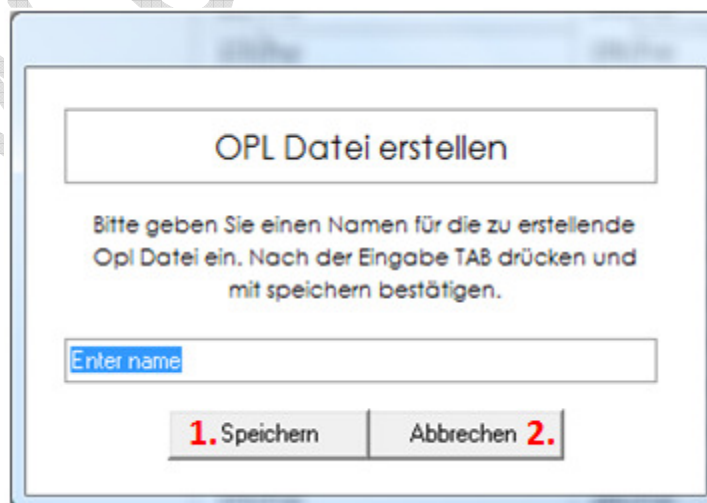
1. Per Drag&Drop Leuchtmelder -> Ports zuweisen.



2. Im Menüband Datei -> OPL Datei erstellen wählen



3. Ein neues Untermenü erscheint in dem ein aussagefähiger Name hinterlegt werden kann. Über diesen Namen ist ein bereits angelegtes Profil jederzeit wieder aufrufbar.



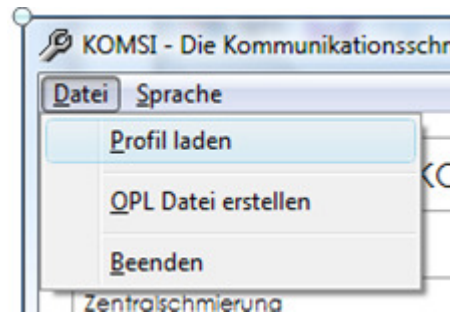
Schaltfläche1: Profil speichern. Eine OPL Datei wird erstellt und im Unterverzeichnis von OFK abgelegt.

Schaltfläche2: Vorgang wieder abgebrochen.

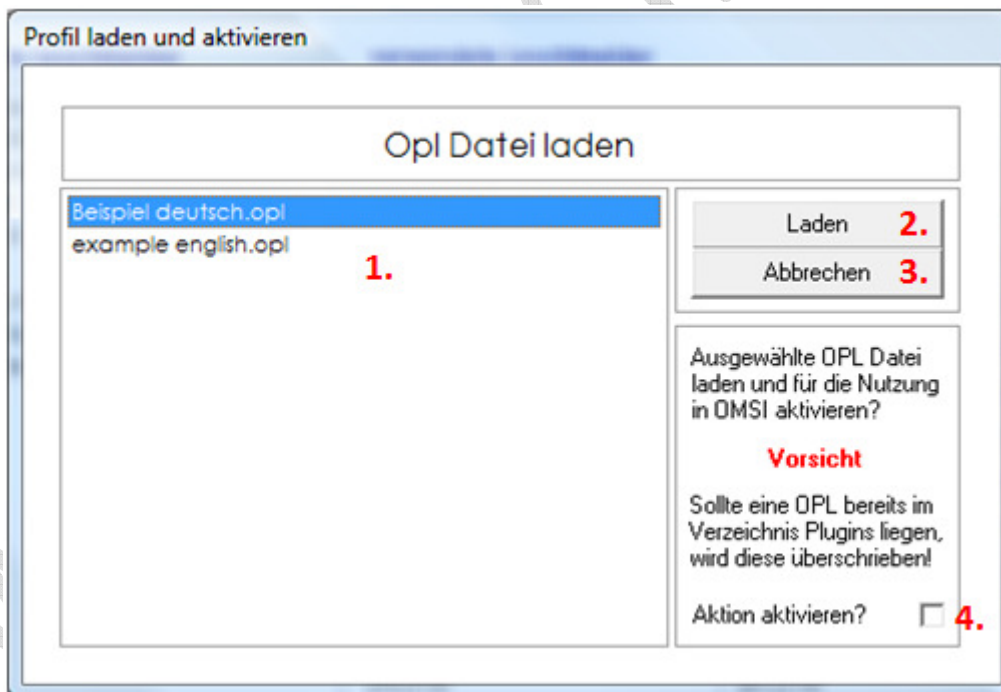


## nützliche Tools – OFK – OPL Creator for KOMSI

4. Eine OPL Datei ist nun erstellt aber für KOMSI noch nicht nutzbar. Hierzu sind folgende Schritte noch notwendig:
- Im Menüband Datei -> Profil laden wählen.



- Im folgenden Menü unter 1. das vorher erstellte Profil auswählen. Bei Punkt 4. den Haken setzen und mit Laden (2) bestätigen. Danach ist das Profil im Verzeichnis Plugins unter KOMSI.opl und KOMSI.cfg angelegt.



Soll nur das Profil geladen werden, um es evtl. weiter zu bearbeiten, einfach den Haken weglassen und nur Laden(2) klicken. Profil wird in den Portfeldern angezeigt, aber keine OPL Datei etc. erstellt.

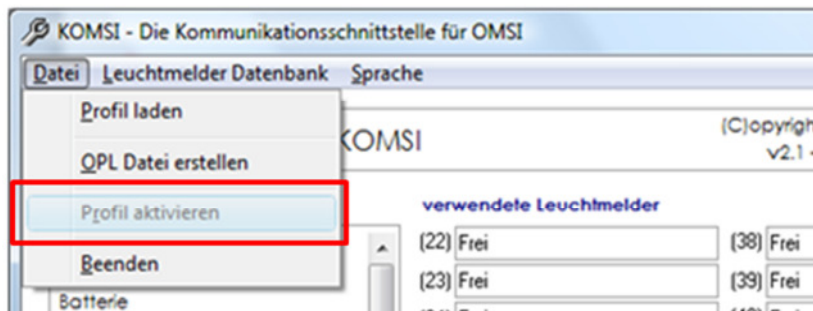


## nützliche Tools – OFK – OPL Creator for KOMSI

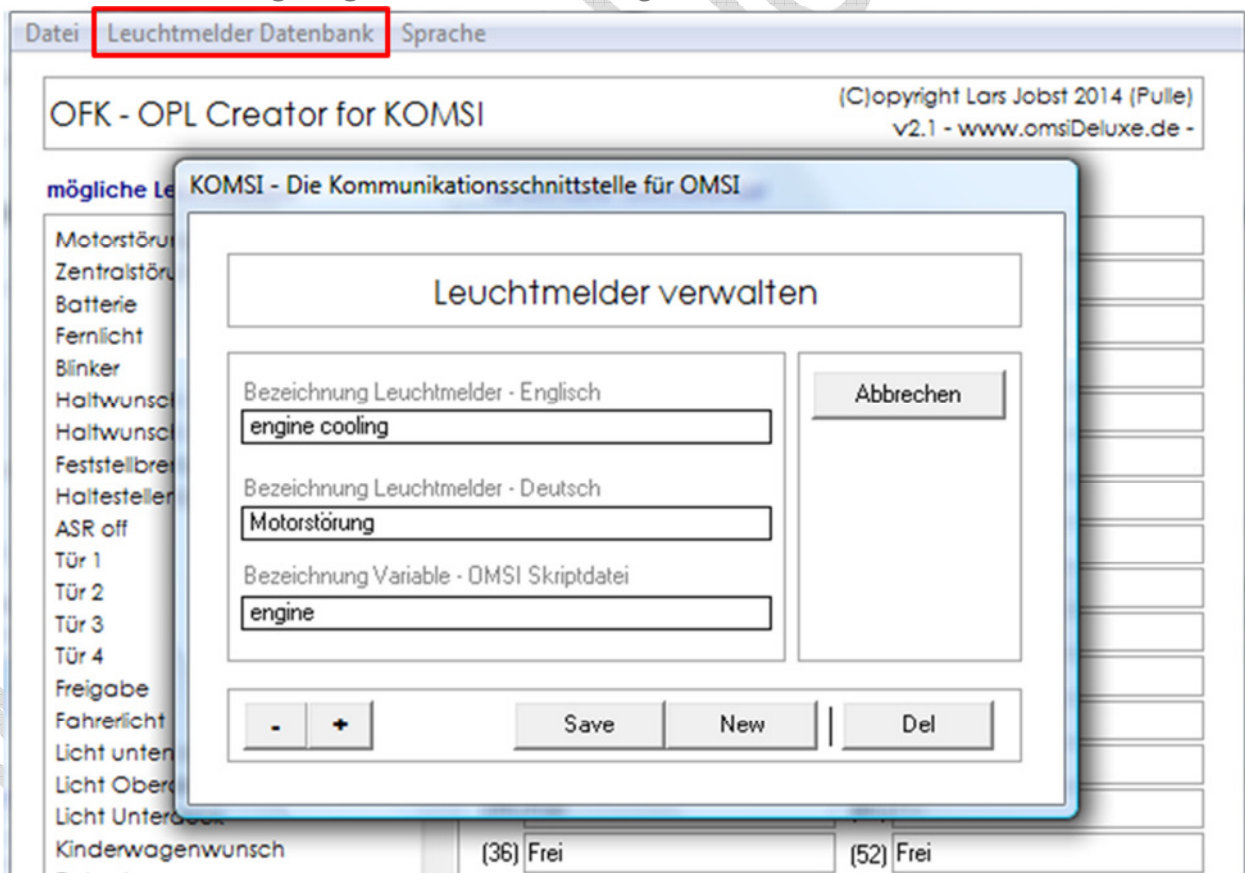
**NEU mit Update 2.1:**

Mit diesem Update habe ich dem OFK noch ein paar Funktionen hinzugefügt:

1. „Hotbutton“ um eine OPL Datei direkt zu aktivieren. Es ist nicht mehr notwendig eine OPL Datei zu laden und dann zu aktivieren.



2. „Leuchtmelder Datenbank“ Hier können nun direkt Datensätze der Datenbank hinzugefügt, bearbeitet und gelöscht werden.





### Allgemeines und Danksagungen

In KOMSI und den weiteren dazugehörigen Komponenten steckt bereits schon jetzt eine Menge Arbeit.

Daher bedanke ich mich erst einmal bei mir selbst :-D und bei meiner Familie die sehr, sehr, sehr viel Geduld und Verständnis für mich und mein Hobby aufgebracht hat und auch noch aufbringen muss :D

Spaß beiseite...

Für das eifrige Betatesten, Ideen liefern und Korrekturlesen des Handbuches möchte ich mich ganz besonders bei GAZZ und Frank (und dessen Frau) bedanken.

Bedanken möchte ich mich auch bei Tom der das alles hier erst in die „richtigen“ Wege gelenkt, ermöglicht und mich ermutigt hat. Als wir vor ca. 1 ½ Jahren das erstemal Kontakt hatten, wäre ich nie im Traum darauf gekommen, dass ein solches Tool entwickelt und vor allem das es schon so viele „verrückte“ FAP Bauer geben wird.

Ansonsten hoffe ich, dass ihr mit KOMSI und Tools schnell zur recht kommt. Hilfe und Support findet ihr natürlich bei uns im Forum.

Wenn ihr Screens, Videos etc. im Internet bereitstellt und verbreitet, ist es echt nett, wenn ihr auf unsere Seite verlinkt, Bezug nehmt oder sonst auch die „Werbetrommel“ rühren könntet. Vielen Dank...

Also bis dahin  
Euer  
Lars