

ibaAnalyzer-E-Dat

Importschnittstelle für externe Messdateiformate

Handbuch
Ausgabe 2.2

Messsysteme für Industrie und Energie

www.iba-ag.com

Hersteller

iba AG
Königswarterstraße 44
90762 Fürth
Deutschland

Kontakte

Zentrale	+49 911 97282-0
Support	+49 911 97282-14
Technik	+49 911 97282-13
E-Mail	iba@iba-ag.com
Web	www.iba-ag.com

Weitergabe sowie Vervielfältigung dieser Unterlage, Verwertung und Mitteilung ihres Inhalts sind nicht gestattet, soweit nicht ausdrücklich zugestanden. Zuwiderhandlungen verpflichten zu Schadenersatz.

© iba AG 2024, alle Rechte vorbehalten.

Der Inhalt dieser Druckschrift wurde auf Übereinstimmung mit der beschriebenen Hard- und Software überprüft. Dennoch können Abweichungen nicht ausgeschlossen werden, so dass für die vollständige Übereinstimmung keine Garantie übernommen werden kann. Die Angaben in dieser Druckschrift werden jedoch regelmäßig aktualisiert. Notwendige Korrekturen sind in den nachfolgenden Auflagen enthalten oder können über das Internet heruntergeladen werden.

Die aktuelle Version liegt auf unserer Website www.iba-ag.com zum Download bereit.

Version	Datum	Revision	Autor	Version SW
2.2	03-2024	Erweiterung der unterstützten Formate	rm	8.0.0

Windows® ist eine Marke und eingetragenes Warenzeichen der Microsoft Corporation. Andere in diesem Handbuch erwähnte Produkt- und Firmennamen können Marken oder Handelsnamen der jeweiligen Eigentümer sein.

Inhalt

1	Zu dieser Dokumentation	5
1.1	Zielgruppe und Vorkenntnisse	5
1.2	Schreibweisen	5
1.3	Verwendete Symbole	6
2	ibaAnalyzer Import-Schnittstelle	7
2.1	Unterstützte Dateiformate	7
2.2	ASCII-Dateien	8
2.2.1	Erzeugen einer geeigneten ASCII-Datei	10
2.2.2	Prinzipieller Dateiaufbau	11
2.2.3	Formatierungshinweise	12
2.2.3.1	Signal-IDs und Kennung Zeitspalte (Zeile 0)	12
2.2.3.2	Signalnamen	14
2.2.3.3	Signaleinheiten	14
2.2.3.4	Infofelder	14
2.2.3.5	Datenzeilen	16
2.2.3.6	Zeitstempel	16
2.2.3.7	Hinweise zur Verwendung von Dezimaltrennzeichen	19
2.3	TDMS-Dateien	21
2.4	Apache Parquet	23
2.4.1	Von ibaAnalyzer erzeugte Dateien	23
2.4.2	Öffnen gewöhnlicher Parquet-Dateien	23
2.4.3	Metadaten für Interpretation durch ibaAnalyzer	25
2.4.3.1	Metadaten Dateiebene	25
2.4.3.2	Metadaten Modulebene	25
2.4.3.3	Metadaten Signalebene	26
2.5	Matlab	27
2.5.1	Datenstruktur	27
2.5.1.1	Struct "filename"	28
2.5.1.2	Struct "fileinfo"	28
2.5.1.3	Struct "module_name"	28
2.5.1.4	Struct "moduleinfo"	28
2.5.1.5	Struct "signal_name"	28

2.5.2	Metadaten für Interpretation durch ibaAnalyzer	29
2.5.2.1	Metadaten Dateiebene.....	29
2.5.2.2	Metadaten Modulebene	29
2.5.2.3	Metadaten Signalebene.....	30
3	Anhang: Beispiele für die Arbeit mit Tabellenprogrammen.....	31
3.1	Messwerttabelle mit Signalnamen und Einheiten ohne Zeit.....	32
3.2	Messwerttabelle mit Zeitstempel, Signalnamen und Einheiten	34
3.3	Messwerttabelle mit Zeitstempel, Signalnamen, Einheiten und Infofeldern	36
4	Support und Kontakt	39

1 Zu dieser Dokumentation

Diese Dokumentation beschreibt die Funktion und die Anwendung der Software *ibaAnalyzer-E-Dat*.

1.1 Zielgruppe und Vorkenntnisse

Diese Dokumentation wendet sich an ausgebildete Fachkräfte, die mit dem Umgang mit elektrischen und elektronischen Baugruppen sowie der Kommunikations- und Messtechnik vertraut sind. Als Fachkraft gilt, wer auf Grund der fachlichen Ausbildung, Kenntnisse und Erfahrungen sowie Kenntnis der einschlägigen Bestimmungen die übertragenen Arbeiten beurteilen und mögliche Gefahren erkennen kann.

1.2 Schreibweisen

In dieser Dokumentation werden folgende Schreibweisen verwendet:

Aktion	Schreibweise
Menübefehle	Menü <i>Funktionsplan</i>
Aufruf von Menübefehlen	<i>Schritt 1 – Schritt 2 – Schritt 3 – Schritt x</i> Beispiel: Wählen Sie Menü <i>Funktionsplan – Hinzufügen – Neuer Funktionsblock</i>
Tastaturtasten	<Tastename> Beispiel: <Alt>; <F1>
Tastaturtasten gleichzeitig drücken	<Tastename> + <Tastename> Beispiel: <Alt> + <Strg>
Grafische Tasten (Buttons)	<Tastename> Beispiel: <OK>; <Abbrechen>
Dateinamen, Pfade	<i>Dateiname, Pfad</i> Beispiel: <i>Test.docx</i>

1.3 Verwendete Symbole

Wenn in dieser Dokumentation Sicherheitshinweise oder andere Hinweise verwendet werden, dann bedeuten diese:

Gefahr!



Wenn Sie diesen Sicherheitshinweis nicht beachten, dann droht die unmittelbare Gefahr des Todes oder der schweren Körperverletzung!

- Beachten Sie die angegebenen Maßnahmen.

Warnung!



Wenn Sie diesen Sicherheitshinweis nicht beachten, dann droht die mögliche Gefahr des Todes oder schwerer Körperverletzung!

- Beachten Sie die angegebenen Maßnahmen.

Vorsicht!



Wenn Sie diesen Sicherheitshinweis nicht beachten, dann droht die mögliche Gefahr der Körperverletzung oder des Sachschadens!

- Beachten Sie die angegebenen Maßnahmen.

Hinweis



Hinweis, wenn es etwas Besonderes zu beachten gibt, wie z. B. Ausnahmen von der Regel usw.

Tipp



Tipp oder Beispiel als hilfreicher Hinweis oder Griff in die Trickkiste, um sich die Arbeit ein wenig zu erleichtern.

Andere Dokumentation



Verweis auf ergänzende Dokumentation oder weiterführende Literatur.

2 ibaAnalyzer Import-Schnittstelle

2.1 Unterstützte Dateiformate

Mit *ibaAnalyzer* lassen sich Messdateien im Normalfall nur dann öffnen und auswerten, wenn sie das iba-eigene Format und die Endung *.dat* haben.

Mit der Zusatzlizenz *ibaAnalyzer-E-Dat*, die im Dongle freigeschaltet werden muss, lassen sich darüber hinaus auch andere Dateiformate einlesen.

ibaAnalyzer unterstützt den Import und die Auswertung von folgenden Dateiformaten, wenn die *ibaAnalyzer-E-Dat*-Lizenz vorhanden ist:

- ASCII (.txt, .csv)
- COMTRADE CFF (*.cff)
- NI TDMS (.tdm, .tdms)¹
- Vista Control (*.varc)
- Apache Parquet
- FDA (*.das)
- Matlab (.mat)
- Wave (*.wav)
- PQDIF (*.pqd)
- Universal 58 (*.unv, *.bunf, *.uff)
- Vold files (*.tra, *.tor, *.spl)

¹ Für die Verarbeitung von TDMS-Dateien beachten Sie die gesonderten Hinweise im Kapitel [TDMS-Dateien](#), Seite 21.

Referenz



Ausführliche Informationen zu den allgemeinen Funktionen von *ibaAnalyzer* finden Sie im *ibaAnalyzer*-Handbuch.

Im "Messdatei öffnen"-Dialog werden nur Dateien mit den Endungen *.dat*, *.hdq*, *.pdc*, *.zip*, *.rar* usw. sowie *.txt*, *.csv*, *.tdm*, *.tdms*, *.das*, *.parquet* oder *.mat* angezeigt. Für die jeweiligen Endungen sind separate Filter verfügbar. Je nach Dateityp sind unterschiedliche Einstellungen zum Lesen der jeweiligen Daten vorhanden.

2.2 ASCII-Dateien

Diese Dateien können mit verschiedenen Werkzeugen erstellt werden, z. B. mit einem Text-Editor, mit MS Excel oder anderen Programmen, die Daten als Textdatei exportieren können.

Damit *ibaAnalyzer* die Dateien importieren und die enthaltenen Daten korrekt darstellen kann, sind einige Rahmenbedingungen für die Formatierung der ASCII-Dateien zu beachten. Verschiedene Optionen bei der Einstellung der Importschnittstelle bieten trotzdem eine gewisse Flexibilität.

Der Dialog zur Einstellung der Importschnittstelle öffnet sich automatisch, sowie Sie eine ASCII-Datei über den "Messdatei öffnen"-Dialog öffnen oder aus dem Windows Explorer per Drag & Drop in das Signalbaumfenster von *ibaAnalyzer* ziehen.

Tipp



Wenn Sie ASCII-Messdateien in *ibaAnalyzer* importieren wollen, versehen Sie sie am besten mit der Endung *.txt* oder *.csv*, damit die Dateien im Browser des "Messdatei öffnen"-Dialogs angezeigt werden.

Benennen Sie die Dateien nicht mit der Endung *.dat*, da *ibaAnalyzer* bei dieser Endung nur Original iba-Messdateien erwartet und akzeptiert.

Andere Dateiendungen, außer den oben genannten, werden nicht akzeptiert.

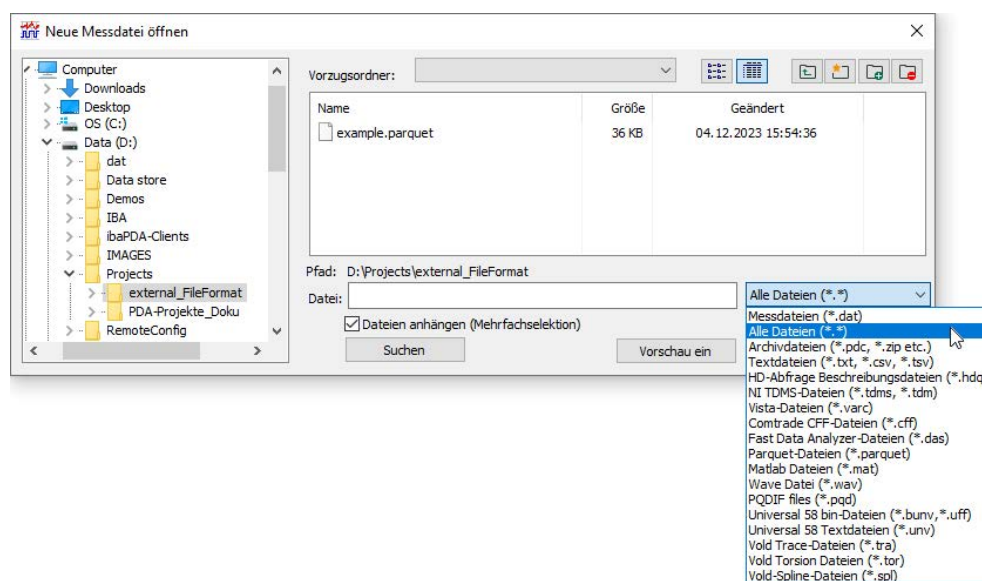


Abb. 1: Dialog zum Öffnen der Messdateien

Hinweis

Die erweiterte Dateivorschau im "Messdatei öffnen"-Dialog wird für die meisten externen Formate nicht unterstützt. Darauf weist Sie eine Mitteilung hin, wenn Sie eine Textdatei bei erweiterter Vorschau öffnen wollen.



Quittieren Sie die Meldung mit <OK>, und fahren Sie mit dem Öffnen der Datei fort. Sie können die Meldung vermeiden, wenn Sie vorher die Vorschau ausschalten.

Wenn Sie eine Datei mit der Endung .txt oder .csv auswählen und auf <OK> klicken, dann öffnet sich der Dialog zur Einstellung der Importschnittstelle:

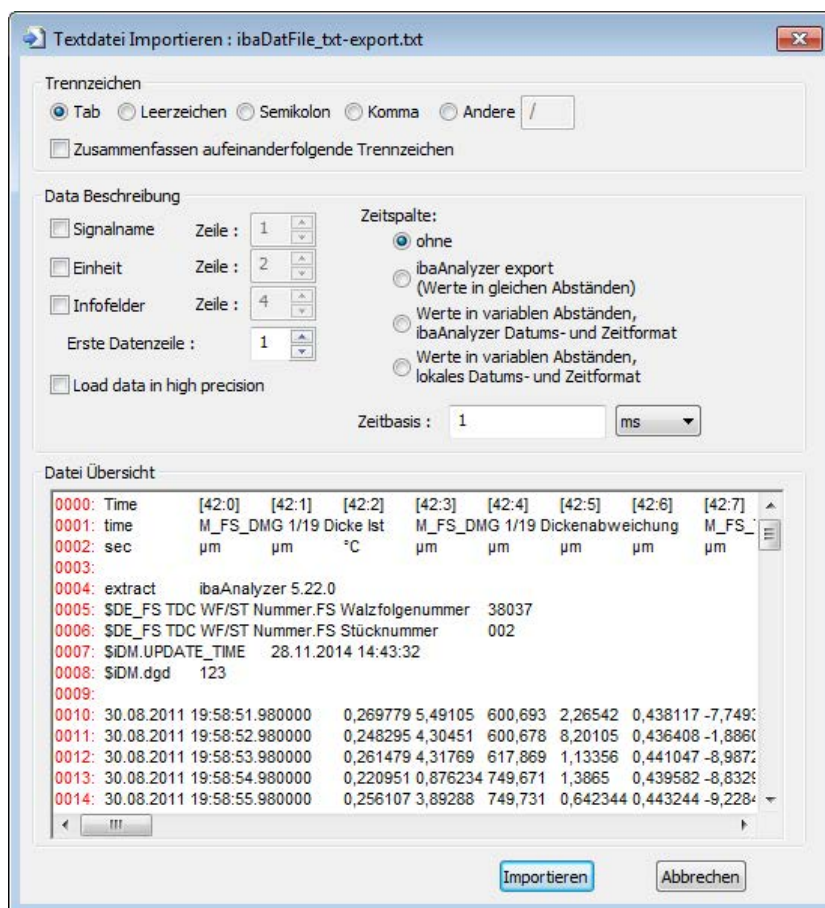


Abb. 2: Dialog zum Importieren von Messdateien im ASCII-Format

In diesem Dialog nehmen Sie die Einstellungen entsprechend dem Format der Quelldatei vor. Voreingestellt sind stets die Einstellungen des letzten erfolgreichen Imports der aktuellen oder einer vorhergehenden *ibaAnalyzer*-Sitzung.

Trennzeichen

Tab, Leerzeichen, Semikolon,...

Wählen Sie hier das Trennzeichen aus, mit dem in der Quelldatei die Parameter und Messwertreihen voneinander getrennt sind. *ibaAnalyzer* verwendet beim Export in eine ASCII-Datei als Standard das Tabulatorzeichen zur Trennung und erzeugt somit eher "TSV"-Dateien (tab separated values) als "CSV"-Dateien (comma separated values). Sowohl für den Export als auch für den Import können Sie aber auch echte "CSV"-Dateien mit Komma als Trennzeichen verwenden. Das funktioniert jedoch nur dann, wenn bei den (Analog-)Messwerten nicht das Komma als Dezimaltrennzeichen benutzt wird. Andere Trennzeichen wie Leerzeichen, Semikolon oder benutzerdefinierte Zeichen werden ebenfalls akzeptiert. Dass im Fall von Leerzeichen als Trennzeichen weder Signalnamen noch Infofelder oder Technostrings Leerzeichen enthalten dürfen, versteht sich von selbst.

Zusammenfassen aufeinanderfolgender Trennzeichen

Wenn diese Option deaktiviert ist (Default) und in der Quelldatei zwei oder mehr Trennzeichen direkt hintereinander stehen (z. B. "Wert1;Wert2;;Wert4;..."), dann werden an den betreffenden Stellen im Signalbaum Signale ohne Inhalt erzeugt.

Wenn Sie diese Option aktivieren, dann werden die hintereinander stehenden Trennzeichen wie ein einziges behandelt, so dass keine Lücken oder leere Signale entstehen.

Tipp



Die Aktivierung dieser Option kann dann sinnvoll sein, wenn die Daten aus einem Tabellenprogramm, z. B. MS Excel, in eine CSV-Datei exportiert werden, um sie mit *ibaAnalyzer* zu analysieren. Bei solchen Programmen ist es nicht unüblich, dass leere Spalten enthalten sind, z. B. aus gestalterischen Gründen. Leere Spalten erzeugen beim ASCII-Export im Allgemeinen aufeinanderfolgende Trennzeichen.

2.2.1 Erzeugen einer geeigneten ASCII-Datei

Im vorigen Kapitel sind die wesentlichen Parameter, die *ibaAnalyzer* für das Lesen von ASCII-Dateien benötigt, beschrieben worden.

Im Folgenden finden Sie detaillierte Hinweise für die Formatierung der Datei und der Inhalte.

Tipp



Wenn Sie über iba-Messdateien verfügen (z. B. erzeugt von *ibaPDA*), dann können Sie sehr einfach eine ASCII-Musterdatei erzeugen, indem Sie die iba-Messdatei mit *ibaAnalyzer* öffnen und anschließend die Export-Funktion (Menü *Datei – Export...*) nutzen. Wenn Sie Ihre eigenen ASCII-Dateien entsprechend diesem Schema formatieren können, dann sollte der Import problemlos funktionieren.

2.2.2 Prinzipieller Dateiaufbau

Zeile	Beschreibung
0	Kennung Zeitspalte (Time) und/oder Signal-IDs ([M:K])
1	Kennung 'time' und/oder Signalnamen
2	Kennung 'sec' und/oder Signaleinheiten
3	\
4	> Bereich für Infofelder/Technostrings
...	/
X	Erste Datenzeile (= erste Samples), ggf. mit Zeitstempel
...	Messwerte bis Dateiende (jede Zeile ein Sample pro Signal),
Y	ggf. mit Zeitstempel

Informationen wie Signal-IDs, Signalnamen, Signaleinheiten und Infofelder sind zwar optionale Angaben, jedoch sind sie für eine verständliche Darstellung der Messwerte in *ibaAnalyzer* sinnvoll.

Wenn die Informationen zwar in der Datei enthalten sind, Sie diese aber nicht nutzen wollen, dann müssen Sie nur die entsprechende Option im Import-Dialog deaktivieren.

Um die einzelnen Informationseinheiten innerhalb der Zeilen voneinander zu trennen, sollte ein einheitliches Trennzeichen verwendet werden. Bewährt hat sich das Tabulatorzeichen (<TAB> bzw. <→|>), da dieses Zeichen in der Regel nicht selbst Bestandteil der Informationen ist. Auch das Semikolon (;) ist als Trennzeichen weit verbreitet. Wenn Sie Komma oder Leerzeichen als Trennzeichen verwenden wollen, müssen Sie darauf achten, dass diese Zeichen nicht in den Daten selbst verwendet werden (z. B. Leerzeichen in Signalnamen). Entsprechendes gilt auch für benutzerdefinierte Trennzeichen.

Der Bereich zwischen der ersten Zeile und dem Beginn der Messwertzeilen ist nicht fest eingeteilt. Empfohlen wird aber ein Aufbau gemäß Tabelle 1. Leerzeilen sind zulässig, aber es dürfen keine Leerzeilen innerhalb der Datenblöcke mit Infefeldern oder Messwerten enthalten sein. Da Leerzeilen auch keine Trennzeichen enthalten dürfen, empfehlen wir, Leerzeilen zu vermeiden.

2.2.3 Formatierungshinweise

2.2.3.1 Signal-IDs und Kennung Zeitspalte (Zeile 0)

Die erste Zeile der Datei sollte der Kennung für eine Zeitspalte und den Signal-IDs vorbehalten sein.

Signal-IDs

Mit den Signal-IDs können Sie die Messwerte strukturieren, entsprechend der Modul-Kanal-Struktur von *ibaAnalyzer*. Sie haben damit die Möglichkeit einer strukturierten Darstellung der Messsignale im Signalbaum. Wenn Sie auf die Nutzung der Signal-ID verzichten, werden alle Signale im Signalbaum unter einem Modul "0" gelistet. Außerdem bietet nur die Verwendung der Signal-IDs die Möglichkeit, Analog- und Digitalsignale zu unterscheiden.

Die Signal-ID setzt sich wie folgt zusammen:

- [Modulnummer:Kanalnummer] für Analogsignale
- [Modulnummer.Kanalnummer] für Digitalsignale
- [Modulnummer:Kanalnummer]_text für Textsignale

Ein Beispiel, das den Effekt der Signal-IDs verdeutlicht, finden Sie am Ende des Kapitels.

In der Signal-ID werden die Module nur über ihre Nummer identifiziert. Modulnamen können zusätzlich als Infofelder vorgegeben werden.

Siehe dazu Kapitel [↗ Infofelder](#), Seite 14.

Kennung Zeitspalte

Wenn Sie einen Zeitstempel für die Messwerte in der Datei verwenden wollen, so muss die Zeitangabe in den Messwertzeilen stets die erste Information sein. Tragen Sie in diesem Fall daher in die erste Zeile zuerst den Begriff 'Time' ein und erst dahinter – getrennt durch Trennzeichen – die Signal-IDs. Dies ist erforderlich, damit die korrekte Zuordnung der Signal-IDs und Messwertreihen erhalten bleibt.

Entsprechend tragen Sie 'time' in die Zeile mit den Signalnamen und 'sec' in die Zeile mit den Einheiten als erste Information ein.

```

1 Time → [0:1] → [0:2] → [1:0] → [1.7]
2 time → Ana_Signal A → Ana_Signal B → Ana_Signal C → Dig_Signal D
3 sec → A → V → W
4
5 16.02.2015 17:51:01.430000 → 0 → 0 → 0 → 0
6 16.02.2015 17:51:02.430000 → 1 → 2 → 3 → 1
7 16.02.2015 17:51:03.430000 → 4 → 5 → 6 → 0
8 16.02.2015 17:51:04.430000 → 7 → 8 → 9 → 0
9

```

Abb. 3: ASCII-Datei mit Zeitspalte (gelb) und Signal-IDs (grün)

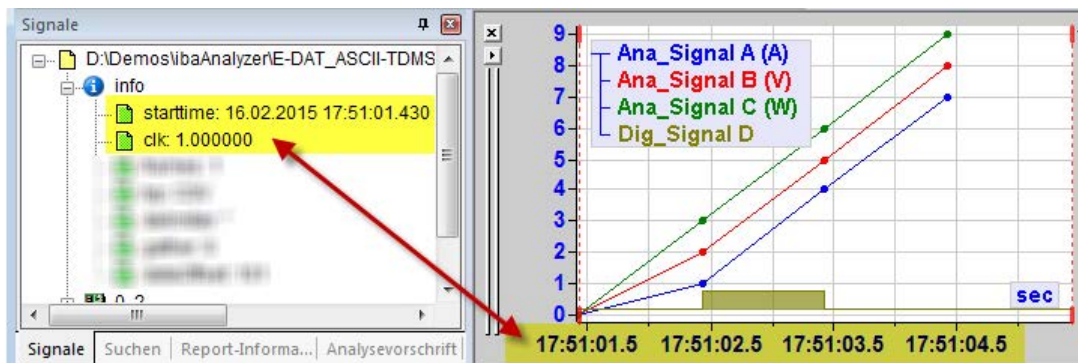
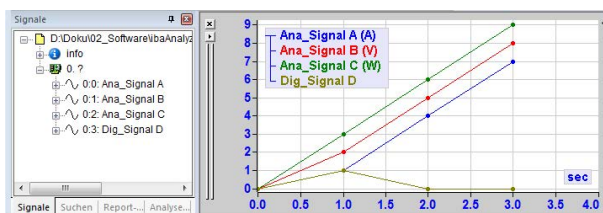


Abb. 4: Darstellung in ibaAnalyzer mit ausgelesener Startzeit und berechneter Zeitbasis

ASCII-Datei ohne Signal-IDs

1	Ana_Signal A	Ana_Signal B	Ana_Signal C	Dig_Signal D
2	A	V	W	
3	0	0	0	0
4	1	2	3	1
5	4	5	6	0
6	7	8	9	0
7				

Ergebnis nach dem Import:

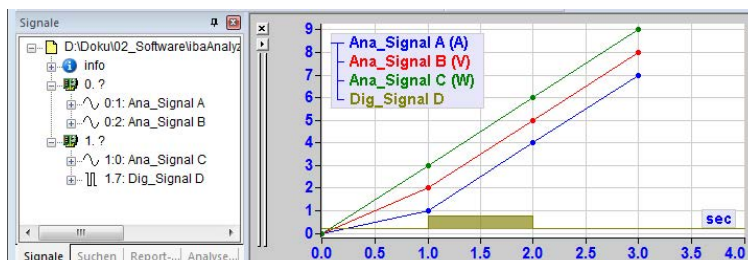


Signale stehen im Signalbaum auf einer Ebene. Digitalsignal "Dig_Signal D" wird wie ein Analogsignal angezeigt.

ASCII-Datei mit Signal-IDs

1	[0:1]	[0:2]	[1:0]	[1:7]
2	Ana_Signal A	Ana_Signal B	Ana_Signal C	Dig_Signal D
3	A	V	W	
4	0	0	0	0
5	1	2	3	1
6	4	5	6	0
7	7	8	9	0

Ergebnis nach dem Import:



Signale stehen strukturiert im Signalbaum. Digitalsignal "Dig_Signal D" wird wie ein Digitalsignal dargestellt.

2.2.3.2 Signalnamen

Wenn Sie Signalnamen an *ibaAnalyzer* übertragen wollen, dann müssen alle Signalnamen in einer Zeile stehen, vorzugsweise in der zweiten, auf jeden Fall jedoch in einer Zeile vor den Messwertzeilen. Wenn keine Zeitspalte vorhanden ist, dann beginnt die Zeile mit dem ersten Signalnamen. Wenn eine Zeitspalte vorhanden ist, dann beginnt die Zeile mit "time", gefolgt von einem Trennzeichen und den Signalnamen, ebenfalls durch Trennzeichen getrennt.

Für die Signalnamen können alle Buchstaben und Zahlen sowie Bindestrich, Unterstrich und Leerzeichen verwendet werden. Das gewählte Trennzeichen sollte nicht in Signalnamen verwendet werden.

Die Signalnamen werden in *ibaAnalyzer* im Signalbaum und in der Legende angezeigt. Sie dienen auch als alternative Referenz bei Analysefunktionen (anstelle der Signal-ID).

2.2.3.3 Signaleinheiten

Wenn Sie die physikalischen Signaleinheiten an *ibaAnalyzer* übertragen wollen, dann müssen alle Signaleinheiten in einer Zeile stehen, vorzugsweise in der dritten, auf jeden Fall jedoch in einer Zeile vor den Messwertzeilen.

Wenn keine Zeitspalte vorhanden ist, dann beginnt die Zeile mit dem ersten Einheitenzeichen. Wenn eine Zeitspalte vorhanden ist, dann beginnt die Zeile mit "sec", gefolgt von einem Trennzeichen und den Einheitenzeichen, ebenfalls durch Trennzeichen getrennt.

Die Signaleinheiten werden in *ibaAnalyzer* zur Beschriftung der Achsen und in der Legende verwendet. Digitalsignale haben keine Einheit, so dass die entsprechende Stelle in der Zeile leer bleibt. Die Anzahl der Trennzeichen muss trotzdem der Anzahl Spalten entsprechen.

```

1 Time → [0:1] → [0:2] → [1:0] → [1.7]
2 time → Ana_Signal A → Ana_Signal B → Ana_Signal C → Dig_Signal D
3 sec → A → V → W →
4
5 16.02.2015 17:51:01.430000 → 0 → 0 → 0 → 0
6 16.02.2015 17:51:02.430000 → 1 → 2 → 3 → 1
7 16.02.2015 17:51:03.430000 → 4 → 5 → 6 → 0
8 16.02.2015 17:51:04.430000 → 7 → 8 → 9 → 0

```

Abb. 5: Signalnamen (rot) und Signaleinheiten (blau)

2.2.3.4 Infofelder

Sie können zusätzliche Informationen, die nicht Bestandteil der Messreihen sind, an *ibaAnalyzer* übertragen. Diese Informationen können sowohl Zahlenwerte als auch Texte sein. Sie nutzen dazu die Funktionen der Infofeld-Zuweisungen oder Technostrings in *ibaAnalyzer*. Diese Zusatzinformationen werden dann im Signalbaum im Zweig "Info" angezeigt und können mit geeigneten Funktionen in den Analysen verwendet werden, z. B. als Textsignal.

Es können beliebig viele Infofelder eingetragen werden. Pro Zeile darf nur ein Infofeld definiert werden. Die Infofelder sollten als Block zusammenhängend nach den oben beschriebenen Zeilen und vor Beginn der Messwertzeilen stehen.

Als erste Information (erste Spalte) steht in der Zeile der Name des Infofeldes. Nach einem Trennzeichen steht in der zweiten Spalte der Inhalt des Infofeldes (Zahlen, Text oder alphanumerischer String).

```

1 Time → [0:1] → [0:2] → [1:0] → [1.7]
2 time → Ana_Signal A → Ana_Signal B → Ana_Signal C → Dig_Signal D
3 sec → A → V → W →
4
5 Infofeld_A → 10
6 Infofeld_B → 0.5
7 Technostring_1 → Textdatei mit baAnalyzer-Zeitformat
8
9 16.02.2015 17:51:01.430000 → 0,0 → 0,0 → 0,0 → 0
10 16.02.2015 17:51:02.430000 → 1,25 → 2,5 → 3,75 → 1
11 16.02.2015 17:51:03.430000 → 4,25 → 5,50 → 6,75 → 0
12 16.02.2015 17:51:04.430000 → 7,25 → 8,5 → 9,75 → 0

```

Abb. 6: Beispiel für 3 Infofelder

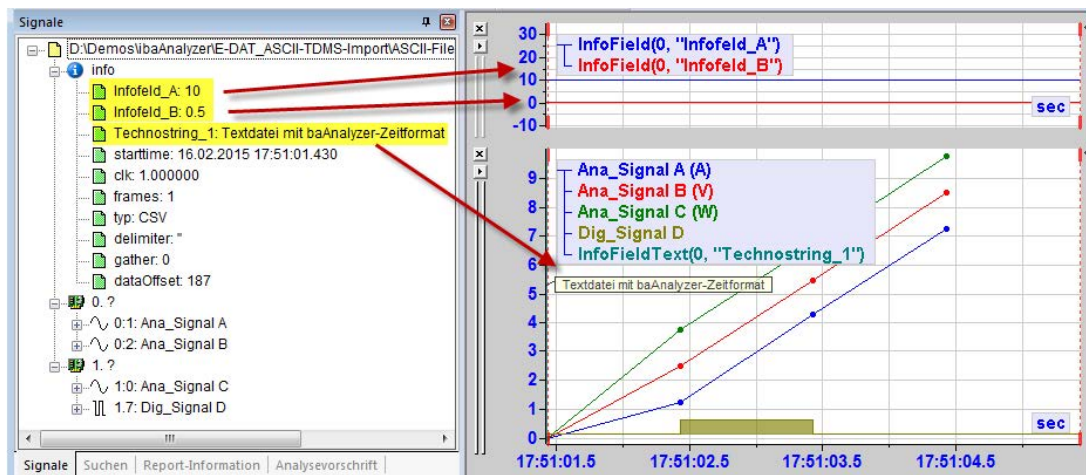


Abb. 7: Darstellung der Infofelder in ibaAnalyzer

Eine andere wichtige Funktion des Infofeldbereiches ist die Möglichkeit, den Modulen einen Namen zu geben.

Für diese Funktion ist die Syntax der Infofelder fest vorgeschrieben:

`Module_name_Index;Mein Modulname`

Index ist hierbei die Nummer des Moduls, wie sie in der Signal-ID vorgegeben wurde (z. B. 0, 1, ...). Nach einem Trennzeichen (z. B. TAB oder Semikolon) folgt der gewünschte Klartextname für das Modul.

Die folgende Abbildung zeigt als Beispiel eine Textdatei mit 2 Modulen (Nr. 0 und 1), dargestellt in *ibaAnalyzer*. Das Inset zeigt die Textdatei mit der Modulnamendefinition. Im Signalbaum werden die Modulnamen entsprechend angezeigt.

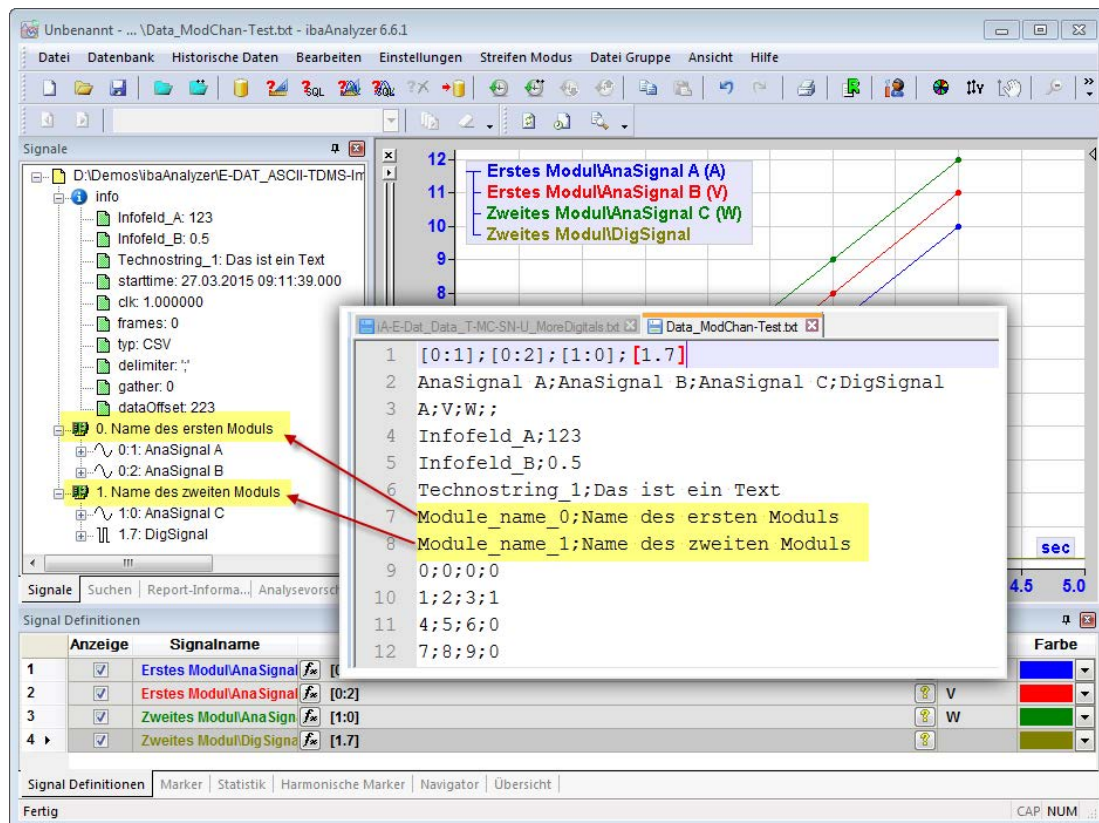


Abb. 8: Definition von Modulnamen in der Textdatei und Resultat im Signalbaum von ibaAnalyzer

2.2.3.5 Datenzeilen

Die Datenzeilen enthalten die Zeitstempel und Messwerte.

Wenn keine Zeitspalte vorhanden ist, dann beginnt die Zeile mit dem Messwert des ersten Signals. Wenn eine Zeitspalte vorhanden ist, dann beginnt die Zeile mit dem Zeitstempel (Datum Uhrzeit), gefolgt von einem Trennzeichen und den Messwerten aller Signale zu diesem Zeitpunkt, ebenfalls durch Trennzeichen getrennt.

Hinweis



Als Dezimalzeichen bei Gleitkommawerten ist das Zeichen zu verwenden, das in der Windows Systemsteuerung, *Region und Sprache – Formate – Weitere Einstellungen – Dezimaltrennzeichen*, eingestellt ist.

2.2.3.6 Zeitstempel

Wenn ein Zeitstempel für die Messwerte vorhanden ist, muss er am Anfang jeder Datenzeile stehen.

Bei der Formatierung der Zeitangabe ist darauf zu achten, dass sie entweder dem *ibaAnalyzer*-Zeitformat (dd.mm.yyyy hh:mm:ss.ms) oder den Einstellungen in der Windows-Systemsteuerung entspricht.

Wenn im Import-Dialog die Zeitspalten-Option "Werte in variablen Abständen (lokales Datums- und Zeitformat)" gewählt wird, dann muss die Zeitangabe entsprechend der Einstellungen in der Windows Systemsteuerung formatiert sein. Ist das nicht der Fall, dann kann *ibaAnalyzer* die Zeitangabe nicht interpretieren und die Messwerte werden nicht dargestellt.

Hinweis



Die 6-stellige Erweiterung für die Mikrosekunden kann der Zeitangabe nach einem Dezimaltrennzeichen hinzugefügt werden. Bei Wahl des *ibaAnalyzer*-Zeitformats muss es der Punkt sein, bei Wahl des lokalen Zeitformats kann es Punkt oder Komma sein.

Beispiel für Einstellung "Deutschland" und Verwendung des ibaAnalyzer-Zeitformats



Abb. 9: Windows Systemsteuerung, Region und Sprache, Format Deutschland

```

1 Time → [0:1] → [0:2] → [1:0] → [1.7]
2 time → Ana_Signal A → Ana_Signal B → Ana_Signal C → Dig_Signal D
3 sec → A → V → W →
4
5 Infofeld_A → 10
6 Infofeld_B → 0.5
7 Technostring_1 → Textdatei mit ibaAnalyzer-Zeitformat
8
9 16.02.2015 17:51:01.430000 → 0,0 → 0,0 → 0,0 → 0
10 16.02.2015 17:51:02.430000 → 1,25 → 2,5 → 3,75 → 1
11 16.02.2015 17:51:03.430000 → 4,25 → 5,50 → 6,75 → 0
12 16.02.2015 17:51:04.430000 → 7,25 → 8,5 → 9,75 → 0

```

Abb. 10: Muster-ASCII-Datei, deutsch

Beispiel für Einstellung "English (United States)" und Verwendung des lokalen (US-) Zeitformats

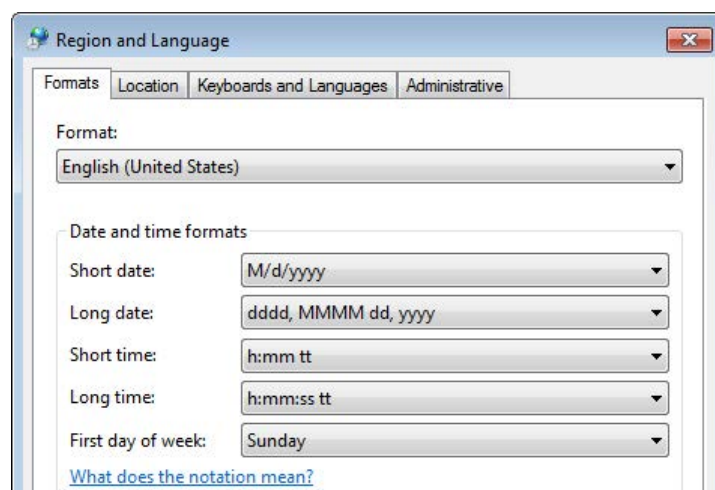


Abb. 11: Windows Systemsteuerung, Region und Sprache, Format English (United States)

```

1 Time → [0:1] → [0:2] → [1:0] → [1.7]
2 time → Ana_Signal A → Ana_Signal B → Ana_Signal C → Dig_Signal D
3 sec → A → V → W →
4
5 Infofeld_A → 10
6 Infofeld_B → 0.5
7 Technostring_1 → This is a file with US date-time format
8
9 2/16/2015 5:51:01.430000 pm → 0.0 → 0.0 → 0.0 → 0
10 2/16/2015 5:51:02.430000 pm → 1.5 → 2.5 → 3.75 → 1
11 2/16/2015 5:51:05.430000 pm → 4.5 → 5.50 → 6.75 → 0
12 2/16/2015 5:51:07.430000 pm → 7.5 → 8.5 → 9.75 → 0

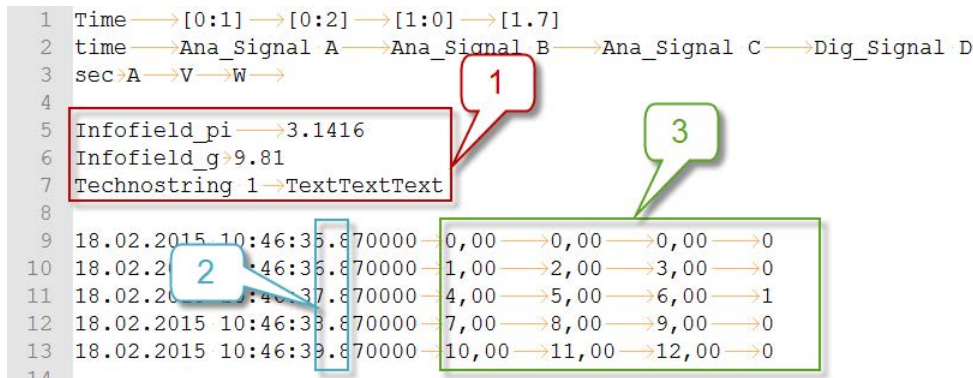
```

Abb. 12: Muster-ASCII-Datei, englisch

2.2.3.7 Hinweise zur Verwendung von Dezimaltrennzeichen

Die Verwendung der richtigen Dezimaltrennzeichen ist wichtig für den korrekten Import von numerischen Daten. Dabei ist die Interpretation der verschiedenen Daten von ihrer Art, von den Systemeinstellungen und teilweise von der Art des Imports abhängig.

Für den Import unterscheidet man drei unterschiedliche Datenarten oder Bereiche:



- 1 Infoceld-Zuweisungen und Technostrings (Technostrings nur bei Messdateien von *ibaPDA* <v7.0)
- 2 Zeitangabe mit Mikrosekunde
- 3 Messwerte (Analogwerte)

Je nach Bereich gibt es folgende Dinge bei der Verwendung von Dezimaltrennzeichen zu beachten.

Infoceld-Zuweisungen und Technostrings

Informationen in Infoceldern und Technostrings werden von *ibaAnalyzer* grundsätzlich als Text (String) interpretiert. Um den Inhalt dieser Größen in *ibaAnalyzer* darstellen zu können, werden spezielle Funktionen verwendet:

- InfoField: Erzeugt ein numerisches Signal
- InfoFieldText: Erzeugt ein Textsignal

Wenn einem Infoceld ein numerischer (Gleitkomma-)Wert zugewiesen werden soll, dann muss - **unabhängig von den Windows-Regionseinstellungen** - als Dezimaltrennzeichen stets ein Punkt verwendet werden.

Wird stattdessen ein Komma benutzt, dann wird der Inhalt des Infoceldes immer als Text interpretiert und dargestellt (Textkanal).

Sie können das leicht nachvollziehen, indem Sie in *ibaAnalyzer* ein Infoceld vom Signalbaum in den Kurvenbereich ziehen oder einen Doppelklick darauf machen. Steht in dem Infoceld eine Zahl mit Dezimalpunkt, dann wird automatisch die Funktion *InfoField* angewendet. Handelt es sich um eine Zahl mit Dezimalkomma wird automatisch die Funktion *InfoFieldText* verwendet.

Nutzen Sie gezielt die Funktion *InfoField*, dann wird nur die Zahl vor dem Komma als Wert interpretiert.

Zeitangabe mit Mikrosekunden

Wollen Sie einen hochgenauen Zeitstempel mit den Messwerten importieren, dann können Sie nach der üblichen Zeitangabe ...hh:mm:ss noch bis zu 6 Stellen (Mikrosekunden) nach einem Dezimaltrennzeichen hinzufügen.

Welches Dezimaltrennzeichen Sie verwenden müssen, hängt nur vom gewünschten Zeitformat bzw. von der Zeitspalten-Option ab. Die Windows-Regionseinstellungen sind irrelevant.

Zeitspalten-Option (Import-Dialog)	Dezimaltrennzeichen
ohne	kein
ibaAnalyzer export (Werte in gleichen Abständen)	Punkt
Werte in variablen Abständen, ibaAnalyzer Datums- und Zeitformat	Punkt
Werte in variablen Abständen, lokales Datums- und Zeitformat	Punkt oder Komma

Messwerte

Bei Analogwerten im Gleitkommaformat ist stets das Dezimaltrennzeichen zu verwenden, das in den Windows-Regionseinstellungen vorgewählt ist.

2.3 TDMS-Dateien

Um Dateien im TDMS-Format lesen zu können, müssen die Dateien die Endung *.tdm* oder *.tdms* haben.

Wichtiger Hinweis



Für die Nutzung der TDMS-Funktionalität müssen Sie auf dem Rechner, auf dem Sie die TDMS-Dateien mit *ibaAnalyzer* öffnen wollen, noch ein zusätzliches Programm installieren: *ibaTDMSBundleInstall.exe*.

Sie finden diese Datei als *ibaTDMSBundleInstall.zip*, auf dem Datenträger "iba Software & Manuals" im Verzeichnis "01_Software\ibaAnalyzer\TDMS-Extraction".

Die in diesem Dateityp enthaltenen Kanalgruppen werden in *ibaAnalyzer* als Module dargestellt, wobei die zugehörigen Kanäle unterhalb der Module angeordnet werden. Dateieigenschaften werden in *ibaAnalyzer* auf Infofelder übertragen und im Info-Zweig des Signalbaums angeordnet. Kanaleigenschaften werden auf Kanal-Infofelder übertragen und unterhalb des jeweiligen Kanals im Signalbaum dargestellt. Kanalgruppen-Eigenschaften werden auf Modul-Infofelder übertragen und können mit den Funktionen *ModuleInfoField* bzw. *ModuleInfoFieldText* ausgewertet werden. Die Bestimmung der Zeitstempel der Messwerte erfolgt anhand folgender Größen:

- Eigenschaft Startzeit der Datei (*datetime*)
- Kanaleigenschaften *wf_increment*, *wf_start_offset* und *wf_start_time*

Weitere Kanaleigenschaften werden wie folgt genutzt:

- *name*: Kanal- bzw. Signalname
- *unit_string*: Wenn vorhanden, dann als Einheit für die Messwerte
- *description*: Wenn vorhanden, dann als Kommentar 1 eines Kanals
- *wf_xname*: Wenn vorhanden, dann wird der Kanal in einer anderen physikalischen Domäne als der Zeit dargestellt. Ist dieser Parameter nicht vorhanden oder enthält er einen unzulässigen Wert, werden die Messwerte des Kanals zeitbasiert mit Basiseinheit Sekunde dargestellt.

Folgende Werte sind zulässig:

- *length*: Längenbasierte Signale, Basiseinheit ist die in *ibaAnalyzer* gewählte Standard-Längeneinheit
- *frequency*: Frequenzbasierte Signale, Basiseinheit ist Hertz
- *invlength*: Invers längenbasierte Signale, Basiseinheit ist 1/Standard-Längeneinheit

Da das TDMS-Messdateiformat keinen eindeutigen Indikator dafür liefert, ob ein Signal als Digitalsignal zu interpretieren ist oder nicht, nutzt *ibaAnalyzer* andere Wege, um ein Signal als Digitalsignal zu klassifizieren. Ein Signal gilt als Digitalsignal, wenn eine der folgenden Bedingungen erfüllt ist:

- Eine Kanaleigenschaft "*iba_digital*" vom Typ 8-Bit-Integer ist vorhanden und hat einen Wert ungleich 0.
- Eine Kanaleigenschaft "*unit_string*" ist vorhanden und hat den Wert "*bit*".
- Der Name des Kanals endet mit "*[bit]*"

2.4 Apache Parquet

Apache Parquet ist ein spaltenorientiertes, binäres Datenformat, das eine effiziente Datenkomprimierung und verschiedene Kodierungen bietet. Aufgrund seiner Spaltenstruktur und der Möglichkeit Metadaten hinzuzufügen ähnelt das Dateiformat dem iba Messdateiformat (*.dat). Auch wegen der vergleichbaren Speichergröße empfehlen wir dieses Dateiformat als Austauschformat mit externen Systemen.

Komprimierung

Das Apache Parquet-Format bietet unterschiedliche Kompressionsmethoden. *ibaAnalyzer* unterstützt flache Kodierung (unkomprimiert), Snappy, Gzip, Brotli, LZ4 und ZStandard.

2.4.1 Von ibaAnalyzer erzeugte Dateien

Mit dem Exportdialog von *ibaAnalyzer* oder mit dem Datenextraktor können Parquet-Dateien mit *ibaAnalyzer* erzeugt werden. Diese Dateien können natürlich auch wieder mit *ibaAnalyzer* geöffnet werden. Die Daten sind in der Ausgabedatei wie folgt strukturiert:

- Ein Kanal (oder ein Ausdruck) entspricht einer Parquet-Spalte
- Die Modulstruktur, wie sie in iba dat-Dateien üblich ist, hat keine Entsprechung im Parquet-Format und wird daher mit Metadaten abgebildet.
- Alle Infofelder sind als Parquet-Metadaten gespeichert.

Die korrekte Datenstruktur ermöglicht es *ibaAnalyzer*, die komplette Dateistruktur beim Öffnen der extrahierten Parquet-Datei wiederherzustellen. In diesem Fall sind alle Module vorhanden und alle (nicht leeren) Infofelder sind verfügbar.

Beachten Sie, dass die Infofelder nach dem Wiederimport alphabetisch sortiert werden. Die erforderlichen Metadaten zur Umsetzung solch einer Struktur in Ihren benutzerdefinierten Parquet-Dateien ist weiter unten beschrieben.

Andere Dokumentation



Weitere Informationen zum Datenextraktor in *ibaAnalyzer* finden Sie im Handbuch *ibaAnalyzer-File-Extract*.

2.4.2 Öffnen gewöhnlicher Parquet-Dateien

Mit *ibaAnalyzer-E-Dat* ist es möglich, gewöhnliche Parquet-Dateien zu öffnen.

Für den Fall, dass kein Metadatum mit dem Schlüssel "clk" vorhanden ist, ist die Eingabe einer Zeitbasis erforderlich. Diese Zeitbasis wird dann für den Import der Daten aus den einzelnen Spalten verwendet.

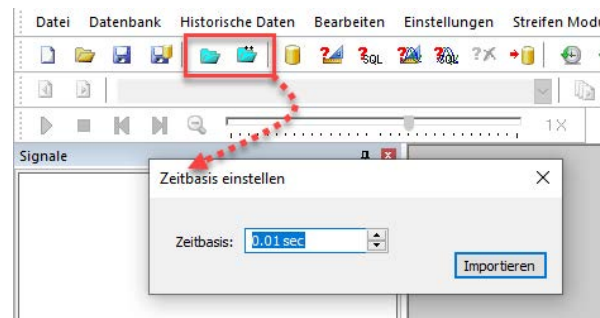


Abb. 13: Aufforderung zur Eingabe einer Zeitbasis beim Öffnen einer Datei ohne Zeitspalte

Beim Import gewöhnlicher Dateien werden alle Signale in ein Modul "0" gepackt und fortlaufend nummeriert. Die Spaltennamen werden als Signalnamen verwendet.

Datentypen

Im Parquet-Format sind viele Datentypen definiert, die nicht alle von *ibaAnalyzer* unterstützt werden. Im Folgenden finden Sie eine Liste mit den unterstützten Datentypen und den resultierenden Datentypen in *ibaAnalyzer*. Die Palette der unterstützten Datentypen kann bei Bedarf noch erweitert werden.

Parquet Datentyp	Interner Speicher in ibaAnalyzer	Bemerkung
Type:: STRING	string	
Type:: BOOL	bool	
Type:: FLOAT	float	
Type:: HALF_FLOAT	float	Kein Genauigkeitsverlust; etwas speicherintensiv zugunsten von Einfachheit
Type:: DOUBLE	double	
Type:: INT8	float	Kein Genauigkeitsverlust; etwas speicherintensiv zugunsten von Einfachheit
Type:: UINT8		
Type:: INT16		
Type:: UINT16		
Type:: INT32	double	
Type:: UINT32		
Type:: TIMESTAMP (based on int64)	double	<p>Möglichkeit des Genauigkeitsverlustes zugunsten von Einfachheit</p> <p>Beachten Sie bitte, dass es keinen geeigneten Weg gibt, Zeitwerte in <i>ibaAnalyzer</i> auf der Y-Achse abzubilden. Sie erscheinen nur als sehr große Zahlenwerte ohne ersichtlichen Sinn.</p>

Tab. 1: Parquet-Datentypen und entsprechende ibaAnalyzer-Datentypen

2.4.3 Metadaten für Interpretation durch ibaAnalyzer

Wenn eine Parquet-Datei mit *ibaAnalyzer* geöffnet wird, dann werden mehrere Metadaten in besonderer Weise interpretiert. Dieser Mechanismus dient dazu, die ursprüngliche Dateistruktur exportierter dat-Dateien wiederherzustellen. Indem Sie die korrekten Metadaten verwenden, werden Ihre Parquet-Dateien strukturiert in *ibaAnalyzer* geöffnet.

Zeitspalte

Die Spalte "time" (z. B. in Dateien, die mit der entsprechenden Einstellung von *ibaAnalyzer* exportiert wurden) wird ignoriert, wenn die Metadaten "clk" und "starttime" vorhanden sind. Diese Werte werden dazu verwendet, die Abtastrate und den Startzeitpunkt der Datei zu ermitteln.

Wenn diese Metadaten nicht vorhanden sind, dann wird die Zeitspalte wie eine normale Signalspalte behandelt, d. h. der Zeitwert steht in *ibaAnalyzer* als "Messwert" zur Verfügung.

Spaltennamen

Üblicherweise interpretiert *ibaAnalyzer* die Spaltennamen als Signalnamen. Der Anzeigename kann überschrieben werden, indem die korrekten Metadaten auf Signalebene (siehe unten) verwendet werden.

2.4.3.1 Metadaten Dateiebene

Alle einfachen Schlüssel-Wert-Paare werden als Infofelder auf Dateiebene interpretiert. Einige reservierte Schlüsselwörter werden von *ibaAnalyzer* wie folgt interpretiert:

- "clk"
Definiert die Abtastzeit des Signals in Sekunden.
- "starttime"
Setzt die Startzeit der Datei. Die Werte müssen in Unixzeit (epoch time) formatiert sein.
- "module_name_n"
Setzt den Anzeigenamen von Modul Nr. "n".

2.4.3.2 Metadaten Modulebene

Alle Schlüsselwörter, die mit "M[n]" beginnen, werden als Infofelder auf Modulebene für das Modul Nr. "n" interpretiert. Diese Daten werden nicht weiter interpretiert, sie werden nur unter dem entsprechenden Modul im Signalbaum sortiert.

2.4.3.3 Metadaten Signalebene

Schlüsselworte, die mit "[column_name]" beginnen, wobei "column_name" der Name der entsprechenden Spalte in der Parquet-Datei ist, werden als Info-Felder auf Signalebene interpretiert. Die folgenden Felder werden interpretiert:

- "[column_name]name"
Setzt den angezeigten Signalnamen in *ibaAnalyzer*
- "[column_name]unit"
Setzt die angezeigte Einheit in der Signaltabelle
- "[column_name]\$PDA_comment1"
Setzt den Kommentar 1, der in der Signaltabelle angezeigt wird; gilt entsprechend für Kommentar 2 (\$PDA_comment2).
- "[column_name]Lengthbased"
Vorhandensein dieses Feldes zeigt an, dass die Spalte längenbezogene Daten enthält
- "[column_name]LengthBase"
Im Fall von längenbezogenen Daten wird hiermit der Abtastabstand in Meter angegeben
- "[column_name]Nr"
Setzt Modul- und Signalnummer. Hierbei wird die *ibaAnalyzer*-Notation verwendet.

Beispiele:

- "3.5" zeigt ein Digitalsignal mit Signalnummer 5 in Modul Nr. 3.
- "4:2" zeigt ein Analogsignal mit Signalnummer 2 in Modul Nr. 4.
- "2:1:1" zeigt den ersten Subkanal von Signal Nr. 1 in Modul Nr. 2

2.5 Matlab

Die Software MATLAB® der Fa. MathWorks® nutzt ein eigenes (binäres) Datenformat mit der Dateierweiterung ".mat". Dateien des Typs .mat können mit *ibaAnalyzer* geöffnet werden, wenn sie eine bestimmte Struktur haben, die im Folgenden beschrieben ist.

Tipp



Daten mit der richtigen Struktur können auch mit dem Datenextraktor in *ibaAnalyzer* erzeugt werden. Weitere Informationen zum Datenextraktor in *ibaAnalyzer* finden Sie im Handbuch *ibaAnalyzer-File-Extract*.

2.5.1 Datenstruktur

Die Software Matlab unterstützt verschiedene Datentypen und Strukturen. *ibaAnalyzer* unterstützt .mat-Dateien, die sogenannte "struct"-Objekte enthalten. Ein "struct"-Objekt in Matlab kann jeden Datentyp enthalten sowie auch andere "struct"-Objekte.

Unterstützt werden eingebettete "struct"-Objekte in folgender Form:

Struct "filename"

 Struct "fileinfo"

 Struct "Module_name_1"

 Struct "moduleinfo"

 Struct "signalname_1"

 Struct "signalname_2"

 ...

 Struct "signalname_n"

 Struct "module_name_2"

 Struct "moduleinfo"

 Struct "signalname_1"

 ...

 ...

 Struct "module_name_n"

Indem Sie solche eingebetteten Strukturen verwenden, können Sie Messdateien ähnlich den iba DAT-Dateien erzeugen, die mit Modulen und Signalen gegliedert sind.

Die verschiedenen "struct"-Objekte können unterschiedliche Informationen enthalten, die in *ibaAnalyzer* entweder als Signale oder als Infofelder auf Datei-, Modul- bzw. Signalebene angezeigt werden.

2.5.1.1 Struct "filename"

Dies ist die Wurzelstruktur. Sie enthält nur andere Strukturen, die ihrerseits Infofelder oder Module enthalten. Alle anderen in dieser Struktur vorhandenen Daten werden ignoriert.

Der Name der Struktur wird als Dateiname im Signalbaum von *ibaAnalyzer* angezeigt.

2.5.1.2 Struct "fileinfo"

Diese Struktur benötigt den festen Namen "fileinfo", um richtig interpretiert zu werden.

Alle enthaltenen numerischen oder textlichen Schlüssel-Wert-Paare werden in *ibaAnalyzer* als Infofelder angezeigt.

Ein spezielles Feld "clk" gibt die Abtastzeit der Signalwerte (in s) an.

Des Weiteren wird ein Feld "starttime" benötigt, um den Zeitstempel für den ersten Messpunkt festzulegen.

2.5.1.3 Struct "module_name"

Der Name dieser Struktur wird als Modulname in *ibaAnalyzer* angezeigt. Wenn die Struktur "moduleinfo" ein Feld "name" enthält, wird stattdessen dessen Inhalt als Name verwendet.

Wie bei der Wurzelstruktur enthalten diese Strukturen nur weitere Strukturen mit Infefeldern auf Modulebene oder Signale. Andere Informationen werden ignoriert.

2.5.1.4 Struct "moduleinfo"

Diese Struktur benötigt den festen Namen "moduleinfo", um richtig interpretiert zu werden. Sie kann in allen Modulen vorhanden sein.

Wie auf Dateiebene werden alle enthaltenen numerischen oder textlichen Schlüssel-Wert-Paare in *ibaAnalyzer* als Infofelder auf Modulebene angezeigt.

2.5.1.5 Struct "signal_name"

Der Name dieser Struktur wird als Signalname in *ibaAnalyzer* angezeigt. Wenn die Struktur ein Feld "name" enthält, wird stattdessen dessen Inhalt als Name verwendet.

Andere numerische oder textliche Schlüssel-Wert-Paare werden in *ibaAnalyzer* als Infofelder auf Signalebene angezeigt.

Die Strukturen auf dieser Ebene enthalten außerdem die eigentlichen Signalwerte. Jede Signalstruktur muss ein Feld (Array) mit dem festen Namen "data" enthalten, das seinerseits die Messwerte enthält. Diese Werte werden dann in *ibaAnalyzer* als Signal angezeigt.

2.5.2 Metadaten für Interpretation durch ibaAnalyzer

Wie zuvor beschrieben, werden mehrere Metadaten auf den verschiedenen Ebenen in besonderer Weise von *ibaAnalyzer* interpretiert. Dieser Mechanismus dient dazu, die ursprüngliche Dateistruktur exportierter dat-Dateien wiederherzustellen.

Indem Sie die korrekten Metadaten verwenden, werden Ihre Matlab-Dateien strukturiert in *ibaAnalyzer* geöffnet.

2.5.2.1 Metadaten Dateiebene

Alle einfachen Schlüssel-Werte-Paare in der Struktur "fileinfo" werden als Infofelder auf Dateiebene interpretiert. Einige reservierte Schlüsselwörter werden von *ibaAnalyzer* wie folgt interpretiert:

- "clk"
Definiert die Abtastzeit des Signals in Sekunden.
- "starttime"
Setzt die Startzeit der Datei. Die Werte müssen als String wie folgt formatiert sein:
"dd.mm.yyyy hh:mm:ss.msmsms"

Das Feld "Time" (z. B. in Dateien, die von *ibaAnalyzer* mit der entsprechenden Option exportiert wurden) enthält einen Spaltenvektor mit den Zeitstempeln der Signale. Das Feld wird ignoriert, wenn die Metadaten "clk" und "starttime" vorhanden sind. Diese Werte werden dazu verwendet, Abtastrate und Startzeitpunkt der Datei zu ermitteln.

2.5.2.2 Metadaten Modulebene

Alle Schlüssel-Wert-Paare, die in einer "moduleinfo"-Struktur vorhanden sind, werden als Infofelder auf Modulebene für die entsprechende Modulnummer interpretiert. Die meisten Felder werden nicht weiter interpretiert, sie werden nur unter dem entsprechenden Modul im Signalbaum sortiert.

Die einzige Ausnahme ist das Feld "ModuleID", mit dem die Modulnummer für die Anzeige in *ibaAnalyzer* gespeichert wird.

2.5.2.3 Metadaten Signalebene

Schlüssel-Wert-Paare, die in den "signal"-Strukturen enthalten sind, werden als Info-Felder auf Signalebene interpretiert. Die einzige Ausnahme ist das Feld "data", das die Messwerte enthält.

Die folgenden Felder werden speziell interpretiert:

- "SignalID"
Setzt die Signalnummer innerhalb des Moduls
- "name"
Setzt den angezeigten Signalnamen in *ibaAnalyzer*
- "unit"
Setzt die angezeigte Einheit in der Signaltabelle
- "PDA_comment1"
Setzt den Kommentar 1, der in der Signaltabelle angezeigt wird; gilt entsprechend für Kommentar 2 (PDA_comment2).
- "PDA_TBase"
Setzt eine Abtastzeit abweichend von dem globalen "clk"-Wert
- "Lengthbased"
Vorhandensein dieses Feldes zeigt an, dass die Spalte längenbezogene Daten enthält
- "LengthBase"
Im Fall von längenbezogenen Daten wird hiermit der Abtastabstand in Meter angegeben

3 Anhang: Beispiele für die Arbeit mit Tabellenprogrammen

Die folgenden Beispiele zeigen verschiedene Möglichkeiten der Datenbereitstellung in Tabellenform, von der einfachen Messwerttabelle bis zur vollständigen Tabelle mit Zeitstempel und Infofeldern.

In jedem Beispiel finden Sie:

- Tabellenstruktur
- Tabelle mit Beispielwerten
- Exportierte Textdatei mit Tabulator als Trennzeichen und die dazu passenden *ibaAnalyzer* Importeinstellungen
- Exportierte CSV-Datei mit Semikolon als Trennzeichen und die dazu passenden *ibaAnalyzer* Importeinstellungen

In den Beispielen werden grundsätzlich die Signal-IDs genutzt.

Tipp



Vermeiden Sie Leerzeilen in den Tabellen, weil beim Export in Text- oder CSV-Dateien in die Leerzeilen trotzdem Trennzeichen für jede Spalte geschrieben werden.

Diese müssen Sie vor dem Einlesen in *ibaAnalyzer* mit einem Texteditor aus den Dateien entfernen. Gleiches gilt für leere Spalten im Bereich Infofelder.

```

1 Time;[0:1];[0:2];[1:0];[1.7]
2 time;Ana_Signal A;Ana_Signal B;Ana_Signal C;Dig_
3 sec;A;V;W;
4 ;;;;
5 Infofield_pi;3,1416;;;
6 Infofield_g;9,81;;;
7 Technostring 1;TextTextText;;;
8 ;;;;
9 18.02.2015 10:46:35.870000;0,00;0,00;0,00;0,00
10 18.02.2015 10:46:36.870000;1,00;2,00;3,00;0,00

```

Dateiexport mit überflüssigen Trennzeichen

Leerzeilen mit Trennzeichen führen dazu, dass die X-Achse in der Kurvenansicht verlängert wird, obwohl keine Werte enthalten sind.

Leerzeilen mit Trennzeichen im Infofeldbereich kann *ibaAnalyzer* nicht interpretieren.

3.1 Messwerttabelle mit Signalnamen und Einheiten ohne Zeit

Tabellenstruktur

	Spalte 1	Spalte 2	...	Spalte m
Zeile 0	[M:C]	[M:C]	[M:C]	[M:C]
Zeile 1	Messwert	Messwert	Messwert	Messwert
Zeile 2	Messwert	Messwert	Messwert	Messwert
Zeile 3	Messwert	Messwert	Messwert	Messwert
Zeile 4	Messwert	Messwert	Messwert	Messwert
Zeile 5	Messwert	Messwert	Messwert	Messwert
Zeile 6	Messwert	Messwert	Messwert	Messwert
Zeile 7	Messwert	Messwert	Messwert	Messwert
Zeile 8	Messwert	Messwert	Messwert	Messwert
Zeile 9	Messwert	Messwert	Messwert	Messwert
Zeile 10	Messwert	Messwert	Messwert	Messwert
Zeile 11	Messwert	Messwert	Messwert	Messwert
...
Zeile n	Messwert	Messwert	Messwert	Messwert

Tabelle mit Werten

[0:1]	[0:2]	[1:0]	[1.7]
Ana_Signal A	Ana_Signal B	Ana_Signal C	Dig_Signal D
A	V	W	
0,00	0,00	0,00	0
1,00	2,00	3,00	0
4,00	5,00	6,00	1
7,00	8,00	9,00	0
10,00	11,00	12,00	0

Excel-Export als Textdatei mit Tab und ibaAnalyzer-Importeinstellungen

```

IA-E_Dat_01_Excel_txt-Tab_de.txt
1  [0:1] → [0:2] → [1:0] → [1.7]
2  Ana_Signal A → Ana_Signal B → Ana_Signal C → Dig_Signal D
3  A → V → W →
4  0,00 → 0,00 → 0,00 → 0
5  1,00 → 2,00 → 3,00 → 0
6  4,00 → 5,00 → 6,00 → 1
7  7,00 → 8,00 → 9,00 → 0
8  10,00 → 11,00 → 12,00 → 0
9

```


Textdatei Importieren : iA-E_Dat_01_Excel_txt-Tab_de.txt

Trennzeichen
☒ Tab ☐ Leerzeichen ☐ Semikolon ☐ Komma ☐ Andere /

☐ Zusammenfassen aufeinanderfolgende Trennzeichen

Data Beschreibung

☒ Signalname Zeile : 1
☒ Einheit Zeile : 2
☐ Infofelder Zeile : 4
Erste Datenzeile : 3

Zeitspalte:
☒ ohne
☐ ibaAnalyzer export (Werte in gleichen Abständen)
☐ Werte in variablen Abständen, ibaAnalyzer Datums- und Zeitformat
☐ Werte in variablen Abständen, lokales Datums- und Zeitformat

☐ Load data in high precision

Zeitbasis : 1 s

Datei Übersicht

0000:	[0:1]	[0:2]	[1:0]	[1.7]
0001:	Ana_Signal A		Ana_Signal B	Ana_Signal C Dig_Signal D
0002:	A V	W		
0003:	0,00	0,00	0,00	0,00
0004:	1,00	2,00	3,00	0,00
0005:	4,00	5,00	6,00	1,00
0006:	7,00	8,00	9,00	0,00
0007:	10,00	11,00	12,00	0,00

Excel-Export als CSV-Datei mit Semikolon und ibaAnalyzer-Importeinstellungen

iA-E_Dat_01_Excel_csv_de.csv

```
1 [0:1];[0:2];[1:0];[1.7]
2 Ana_Signal A;Ana_Signal B;Ana_Signal C;Dig_Signal D
3 A;V;W;
4 0,00;0,00;0,00;0
5 1,00;2,00;3,00;0
6 4,00;5,00;6,00;1
7 7,00;8,00;9,00;0
8 10,00;11,00;12,00;0
```

Textdatei Importieren : iA-E_Dat_01_Excel_csv_de.csv

Trennzeichen
☐ Tab ☐ Leerzeichen ☒ Semikolon ☐ Komma ☐ Andere /

☐ Zusammenfassen aufeinanderfolgende Trennzeichen

Data Beschreibung

☒ Signalname Zeile : 1
☒ Einheit Zeile : 2
☐ Infofelder Zeile : 4
Erste Datenzeile : 3

Zeitspalte:
☒ ohne
☐ ibaAnalyzer export (Werte in gleichen Abständen)
☐ Werte in variablen Abständen, ibaAnalyzer Datums- und Zeitformat
☐ Werte in variablen Abständen, lokales Datums- und Zeitformat

☐ Load data in high precision

Zeitbasis : 1 s

Datei Übersicht

0000:	[0:1];[0:2];[1:0];[1.7]
0001:	Ana_Signal A;Ana_Signal B;Ana_Signal C;Dig_Signal D
0002:	A;V;W;
0003:	0,00;0,00;0,00;0,00
0004:	1,00;2,00;3,00;0,00
0005:	4,00;5,00;6,00;1,00
0006:	7,00;8,00;9,00;0,00
0007:	10,00;11,00;12,00;0,00

3.2 Messwerttabelle mit Zeitstempel, Signalnamen und Einheiten

Tabellenstruktur

	Spalte 1	Spalte 2	...	Spalte m
Zeile 0	Time	[M:C]	[M:C]	[M:C]
Zeile 1	time	Signalname 1	Signalname 2	Signalname n
Zeile 2	sec	Einheit	Einheit	Einheit
Zeile 3				
Zeile 4	Zeitstempel 0	Messwert	Messwert	Messwert
Zeile 5	Zeitstempel 1	Messwert	Messwert	Messwert
Zeile 6	Zeitstempel 2	Messwert	Messwert	Messwert
Zeile 7	Zeitstempel 3	Messwert	Messwert	Messwert
Zeile 8	Zeitstempel 4	Messwert	Messwert	Messwert
Zeile 9	Zeitstempel 5	Messwert	Messwert	Messwert
Zeile 10	Zeitstempel 6	Messwert	Messwert	Messwert
Zeile 11	Zeitstempel 7	Messwert	Messwert	Messwert
...
Zeile n	Zeitstempel n	Messwert	Messwert	Messwert

Tabelle mit Werten

Time	[0:1]	[0:2]	[1:0]	[1:7]
time	Ana_Signal A	Ana_Signal B	Ana_Signal C	Dig_Signal D
sec	A	V	W	
18.02.2015 10:46:35.870000	0,00	0,00	0,00	0
18.02.2015 10:46:36.870000	1,00	2,00	3,00	0
18.02.2015 10:46:37.870000	4,00	5,00	6,00	1
18.02.2015 10:46:38.870000	7,00	8,00	9,00	0
18.02.2015 10:46:39.870000	10,00	11,00	12,00	0

Leerzeilen vermeiden!

Excel-Export als Textdatei mit Tab und ibaAnalyzer-Importeinstellungen

```

1 Time → [0:1] → [0:2] → [1:0] → [1.7]
2 time → Ana_Signal A → Ana_Signal B → Ana_Signal C → Dig_Signal D
3 sec → A → V → W →
4 18.02.2015 10:46:35.870000 → 0,00 → 0,00 → 0,00 → 0
5 18.02.2015 10:46:36.870000 → 1,00 → 2,00 → 3,00 → 0
6 18.02.2015 10:46:37.870000 → 4,00 → 5,00 → 6,00 → 1
7 18.02.2015 10:46:38.870000 → 7,00 → 8,00 → 9,00 → 0
8 18.02.2015 10:46:39.870000 → 10,00 → 11,00 → 12,00 → 0
9

```

Textdatei Importieren : iA-E_Dat_02_Excel_bt-Tab_de.bt

Trennzeichen
☒ Tab ☐ Leerzeichen ☐ Semikolon ☐ Komma ☐ Andere /

☐ Zusammenfassen aufeinanderfolgende Trennzeichen

Data Beschreibung

☒ Signalname Zeile : 1
☒ Einheit Zeile : 2
☐ Infofelder Zeile : 4
 Erste Datenzeile : 3
☐ Load data in high precision

Zeitspalte:
☒ ibaAnalyzer export (Werte in gleichen Abständen)
☐ Werte in variablen Abständen, ibaAnalyzer Datums- und Zeitformat
☐ Werte in variablen Abständen, lokales Datums- und Zeitformat

Zeitbasis : 1 s

Datei Übersicht

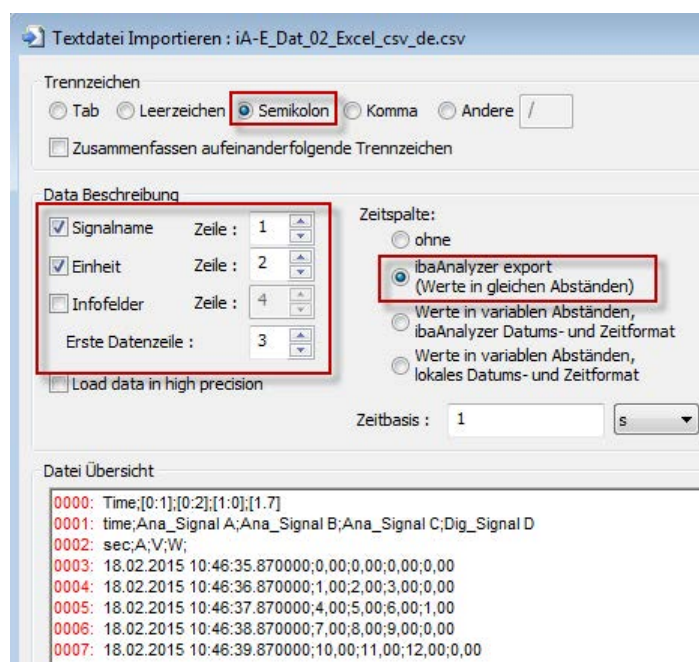
	Time	[0:1]	[0:2]	[1:0]	[1.7]		
0000:	Time						
0001:	time	Ana_Signal A	Ana_Signal B	Ana_Signal C	Dig_Signal D		
0002:	sec	A	V	W			
0003:	18.02.2015 10:46:35.870000	0,00	0,00	0,00	0,00		
0004:	18.02.2015 10:46:36.870000	1,00	2,00	3,00	0,00		
0005:	18.02.2015 10:46:37.870000	4,00	5,00	6,00	1,00		
0006:	18.02.2015 10:46:38.870000	7,00	8,00	9,00	0,00		
0007:	18.02.2015 10:46:39.870000	10,00	11,00	12,00	0,00		

Excel-Export als CSV-Datei mit Semikolon und ibaAnalyzer-Importeinstellungen

```

1 Time; [0:1]; [0:2]; [1:0]; [1.7]
2 time; Ana_Signal A; Ana_Signal B; Ana_Signal C; Dig_Signal D
3 sec; A; V; W;
4 18.02.2015 10:46:35.870000; 0,00; 0,00; 0,00; 0
5 18.02.2015 10:46:36.870000; 1,00; 2,00; 3,00; 0
6 18.02.2015 10:46:37.870000; 4,00; 5,00; 6,00; 1
7 18.02.2015 10:46:38.870000; 7,00; 8,00; 9,00; 0
8 18.02.2015 10:46:39.870000; 10,00; 11,00; 12,00; 0
9

```



3.3 Messwerttabelle mit Zeitstempel, Signalnamen, Einheiten und Infefeldern

Tabellenstruktur

	Spalte 1	Spalte 2	...	Spalte m
Zeile 0	Time	[M:C]	[M:C]	[M:C]
Zeile 1	time	Signalname 1	Signalname 2	Signalname n
Zeile 2	sec	Einheit	Einheit	Einheit
Zeile 3				
Zeile 4	Infofeld 1	Inhalt Info/String 1		
Zeile 5	Infofeld 2	Inhalt Info/String 2		
Zeile 6		
Zeile 7	Infofeld X	Inhalt Info/String X		
Zeile 8				
Zeile 9	Zeitstempel 0	Messwert	Messwert	Messwert
Zeile 10	Zeitstempel 1	Messwert	Messwert	Messwert
Zeile 11	Zeitstempel 2	Messwert	Messwert	Messwert
...
Zeile n	Zeitstempel n	Messwert	Messwert	Messwert

Tabelle mit Werten

Time	[0:1]	[0:2]	[1:0]	[1.7]
time	Ana_Signal A	Ana_Signal B	Ana_Signal C	Dig_Signal D
sec	A	V	W	
Infofield_pi	3.1416			
Infofield_g	9.81			
Technostring 1	TextTextText			
18.02.2015 10:46:35.870000	0,00	0,00	0,00	0
18.02.2015 10:46:36.870000	1,00	2,00	3,00	0
18.02.2015 10:46:37.870000	4,00	5,00	6,00	1
18.02.2015 10:46:38.870000	7,00	8,00	9,00	0
18.02.2015 10:46:39.870000	10,00	11,00	12,00	0

Leerzeilen vermeiden!

Excel-Export als Textdatei mit Tab und ibaAnalyzer-Importeinstellungen

```

1 Time → [0:1] → [0:2] → [1:0] → [1.7]
2 time → Ana_Signal A → Ana_Signal B → Ana_Signal C → Dig_Signal D
3 sec → A → V → W →
4 Infofield_pi → 3.1416
5 Infofield_g → 9.81
6 Technostring 1 → TextTextText
7 18.02.2015 10:46:35.870000 → 0,00 → 0,00 → 0,00 → 0
8 18.02.2015 10:46:36.870000 → 1,00 → 2,00 → 3,00 → 0
9 18.02.2015 10:46:37.870000 → 4,00 → 5,00 → 6,00 → 1
10 18.02.2015 10:46:38.870000 → 7,00 → 8,00 → 9,00 → 0
11 18.02.2015 10:46:39.870000 → 10,00 → 11,00 → 12,00 → 0
12

```

Textdatei Importieren : iA-E_Dat_03_Excel_txt-Tab1_de.txt

Trennzeichen
☒ Tab ☐ Leerzeichen ☐ Semikolon ☐ Komma ☐ Andere /

☐ Zusammenfassen aufeinanderfolgende Trennzeichen

Data Beschreibung

☒ Signalname Zeile : 1
☒ Einheit Zeile : 2
☒ Infofelder Zeile : 3
Erste Datenzeile : 6

☐ Daten mit hoher Genauigkeit

Zeitspalte:
☒ ohne
☒ ibaAnalyzer export (Werte in gleichen Abständen)
☐ Werte in variablen Abständen, ibaAnalyzer Datums- und Zeitformat
☐ Werte in variablen Abständen, lokales Datums- und Zeitformat

Zeitbasis : 1 s

Datei Übersicht

0000:	Time	[0:1]	[0:2]	[1:0]	[1.7]
0001:	time	Ana_Signal A	Ana_Signal B	Ana_Signal C	Dig_Signal D
0002:	sec	A	V	W	
0003:	Infofield_pi	3.1416			
0004:	Infofield_g	9.81			
0005:	Technostring 1	TextTextText			
0006:	18.02.2015 10:46:35.870000	0,00	0,00	0,00	0
0007:	18.02.2015 10:46:36.870000	1,00	2,00	3,00	0
0008:	18.02.2015 10:46:37.870000	4,00	5,00	6,00	1
0009:	18.02.2015 10:46:38.870000	7,00	8,00	9,00	0
0010:	18.02.2015 10:46:39.870000	10,00	11,00	12,00	0

Excel-Export als CSV-Datei mit Semikolon und ibaAnalyzer-Importeinstellungen

```

1 Time;[0:1];[0:2];[1:0];[1.7]
2 time;Ana_Signal A;Ana_Signal B;Ana_Signal C;Dig_Signal D
3 sec;A;V;W;
4 Infofield_pi;3.1416
5 Infofield_g;9.81
6 Technostring 1;TextTextText
7 18.02.2015 10:46:35.870000;0,00;0,00;0,00;0
8 18.02.2015 10:46:36.870000;1,00;2,00;3,00;0
9 18.02.2015 10:46:37.870000;4,00;5,00;6,00;1
10 18.02.2015 10:46:38.870000;7,00;8,00;9,00;0
11 18.02.2015 10:46:39.870000;10,00;11,00;12,00;0

```

Textdatei Importieren : iA-E_Dat_03_Excel_csv1_de.csv

Trennzeichen
☐ Tab ☐ Leerzeichen ☒ Semikolon ☐ Komma ☐ Andere /

☐ Zusammenfassen aufeinanderfolgende Trennzeichen

Data Beschreibung

<input checked="" type="checkbox"/> Signalname	Zeile : 1
<input checked="" type="checkbox"/> Einheit	Zeile : 2
<input checked="" type="checkbox"/> Infofelder	Zeile : 3

Erste Datenzeile : 6

☐ Daten mit hoher Genauigkeit

Zeitspalte:
☐ ohne
☒ ibaAnalyzer export (Werte in gleichen Abständen)
☐ Werte in variablen Abständen, ibaAnalyzer Datums- und Zeitformat
☐ Werte in variablen Abständen, lokales Datums- und Zeitformat

Zeitbasis : 1 s

Datei Übersicht

```

0000: Time;[0:1];[0:2];[1:0];[1.7]
0001: time;Ana_Signal A;Ana_Signal B;Ana_Signal C;Dig_Signal D
0002: sec;A;V;W;
0003: Infofield_pi;3.1416
0004: Infofield_g;9.81
0005: Technostring 1;TextTextText
0006: 18.02.2015 10:46:35.870000;0,00;0,00;0,00;0
0007: 18.02.2015 10:46:36.870000;1,00;2,00;3,00;0
0008: 18.02.2015 10:46:37.870000;4,00;5,00;6,00;1
0009: 18.02.2015 10:46:38.870000;7,00;8,00;9,00;0
0010: 18.02.2015 10:46:39.870000;10,00;11,00;12,00;0

```

4 Support und Kontakt

Support

Tel.: +49 911 97282-14

E-Mail: support@iba-ag.com

Hinweis



Wenn Sie Support benötigen, dann geben Sie bitte bei Softwareprodukten die Nummer des Lizenzcontainers an. Bei Hardwareprodukten halten Sie bitte ggf. die Seriennummer des Geräts bereit.

Kontakt

Hausanschrift

iba AG
Königswarterstraße 44
90762 Fürth
Deutschland

Tel.: +49 911 97282-0

E-Mail: iba@iba-ag.com

Postanschrift

iba AG
Postfach 1828
90708 Fürth

Warenanlieferung, Retouren

iba AG
Gebhardtstraße 10
90762 Fürth

Regional und weltweit

Weitere Kontaktadressen unserer regionalen Niederlassungen oder Vertretungen finden Sie auf unserer Webseite:

www.iba-ag.com