



DRIVING EMBEDDED EXCELLENCE

ETAS INCA-MIP V7.4

Benutzerhandbuch

Copyright

Die Angaben in diesem Schriftstück dürfen nicht ohne gesonderte Mitteilung der ETAS GmbH geändert werden. Des Weiteren geht die ETAS GmbH mit diesem Schriftstück keine weiteren Verpflichtungen ein. Die darin dargestellte Software wird auf Basis eines allgemeinen Lizenzvertrages oder einer Einzellizenz geliefert. Benutzung und Vervielfältigung ist nur in Übereinstimmung mit den vertraglichen Abmachungen gestattet.

Unter keinen Umständen darf ein Teil dieser Veröffentlichung in irgendeiner Form ohne schriftliche Genehmigung der ETAS GmbH kopiert, vervielfältigt, in einem Retrievalsystem gespeichert oder in eine andere Sprache übersetzt werden.

© **Copyright 2022** ETAS GmbH, Stuttgart

Die verwendeten Bezeichnungen und Namen sind Warenzeichen oder Handelsnamen ihrer entsprechenden Eigentümer.

MATLAB und Simulink sind eingetragene Warenzeichen von The MathWorks, Inc. Die Website [mathworks.com/trademarks](https://www.mathworks.com/trademarks) enthält weitere Warenzeichen.

INCA-MIP V7.4 - Benutzerhandbuch R01 DE - 03.2022

Inhaltsverzeichnis

1	Einleitung	6
1.1	Bestimmungsgemäße Verwendung	6
1.2	Zielgruppe	6
1.3	Klassifizierung von Warnhinweisen	6
1.4	Sicherheitshinweise	7
1.5	Datenschutzhinweise	8
2	Über INCA-MIP	9
2.1	INCA Definitionen	10
3	Installation	12
3.1	Systemvoraussetzungen	12
3.2	INCA-MIP Installieren	12
3.3	Cache für MATLAB® Toolboxverzeichnisse aktualisieren	15
3.4	Cache für MATLAB®-Toolbox-Verzeichnisse deaktivieren	15
3.5	Lizenzierung der Software	15
4	API-Funktionen	17
4.1	Kennenlernen der INCA-MIP API anhand von Beispieldateien	21
4.2	Allgemeine Funktionen	22
4.2.1	Message IDs über das INCA-MIP API zurückgeben	22
4.2.2	Nachrichten während der Skriptausführung anzeigen	25
4.2.3	Vorhandensein einer gültigen INCA-MIP-Lizenz anzeigen (INCA-MIP Erweitert)	25
4.2.4	Informationen zu allen installierten INCA-Versionen lesen	25
4.2.5	Informationen zu allen installierten Produkt-Add-ons lesen	26
4.2.6	INCA-Version auslesen	27
4.2.7	Eigenschaften von INCA lesen (INCA-MIP Erweitert)	27
4.3	Initialisierung	27
4.3.1	INCA öffnen	28
4.3.2	INCA schließen (INCA-MIP Erweitert)	28
4.3.3	Datenbank öffnen	29
4.3.4	Datenbank importieren (INCA-MIP Erweitert)	29
4.3.5	Datenbankelemente lesen (INCA-MIP Erweitert)	30
4.3.6	Projekt und Datensatz einem Gerät zuordnen (INCA-MIP Erweitert)	31
4.3.7	Experiment öffnen	32

4.3.8	Experiment zurücksetzen	33
4.3.9	Geräte lesen (INCA-MIP Erweitert)	33
4.3.10	Eigenschaften von Geräten lesen (INCA-MIP Erweitert)	33
4.4	Messen und Aufzeichnen	34
4.4.1	Messgrößen auflisten (INCA-MIP Erweitert)	35
4.4.2	Messraster lesen (INCA-MIP Erweitert)	35
4.4.3	Messgröße zu Experiment hinzufügen	36
4.4.4	Messung starten	37
4.4.5	Messung beenden	38
4.4.6	Eigenschaften der Aufzeichnung lesen (INCA-MIP Erweitert)	38
4.4.7	Eigenschaften der Aufzeichnung ändern (INCA-MIP Erweitert)	40
4.4.8	Aufzeichnungsmodus lesen (INCA-MIP Erweitert)	42
4.4.9	Aufzeichnungsmodus festlegen (INCA-MIP Erweitert)	43
4.4.10	Aufzeichnung starten	44
4.4.11	Aufzeichnung beenden	45
4.4.12	Datenlesemodus einstellen (Online/Offline-Daten)	45
4.4.13	Messdaten lesen	46
4.4.14	Ringpuffer zurücksetzen	49
4.4.15	Hardware-Status lesen (INCA-MIP Erweitert)	49
4.4.16	Trigger definieren (INCA-MIP Erweitert)	50
4.4.17	Manuellen Trigger auslösen (INCA-MIP Erweitert)	52
4.4.18	Aufzeichnungsstatus lesen (INCA-MIP Erweitert)	52
4.4.19	Liste von Messgrößen lesen (INCA-MIP Erweitert)	52
4.5	Verstellen	53
4.5.1	Verstellgrößen auflisten (INCA-MIP Erweitert)	53
4.5.2	Verstellgröße hinzufügen	54
4.5.3	Verstellgröße lesen	55
4.5.4	Verstellgröße ändern	57
4.5.5	Datensatz einem Gerät zuordnen (INCA-MIP Erweitert)	60
4.5.6	Datensätze eines Gerätes auflisten (INCA-MIP Erweitert)	60
4.5.7	Verstellmodus einstellen (INCA-MIP Erweitert)	61
4.5.8	Geräte gruppieren (INCA-MIP Erweitert)	62
4.5.9	DCM-Datei schreiben (INCA-MIP Erweitert)	62
4.6	Speicherseitenverwaltung	62
4.6.1	Speicherseite aktivieren	63
4.6.2	Aktive Speicherseite lesen (INCA-MIP Erweitert)	63
4.6.3	Schreibschutz prüfen	63

4.6.4	Speicherseite hinunterladen	64
4.6.5	Speicherseite kopieren	64
4.6.6	Unterschiede hinunterladen	65
4.6.7	Speicherseiten hochladen (INCA-MIP Erweitert)	65
4.7	Anwendungsbeispiele	66
5	Erstellung und Verteilung von ausführbaren Dateien mit dem MATLAB® Compiler	68
5.1	m-Dateien kompilieren	68
5.2	Ausführbare Dateien verteilen	69
6	Kontaktinformationen	70
Index	71

1 Einleitung

1.1 Bestimmungsgemäße Verwendung

INCA und INCA Add-ons werden für Automobilanwendungen und für die in der Anwenderdokumentation für INCA und INCA Add-Ons beschriebenen Vorgehensweisen entwickelt und freigegeben.

Das INCA-MIP Add-On (INCA MATLAB® Integration Package) stellt eine Programmierschnittstelle dar, über die die Funktionalität von INCA aus MATLAB® heraus angesteuert werden kann.

INCA und die INCA Add-ons sind für den Einsatz in Industrielaboren und in Testfahrzeugen vorgesehen.

Die ETAS GmbH kann nicht für Schäden haftbar gemacht werden, die durch falschen Gebrauch und Missachtung der Safety Information verursacht werden.

1.2 Zielgruppe

Dieses Softwareprodukt und dieses Benutzerhandbuch richten sich an qualifiziertes Personal, das in den Bereichen Entwicklung und Applikation von Kfz-Steuergeräten arbeitet, sowie an Systemadministratoren und Benutzer mit Administratorrechten, die Software installieren, warten oder deinstallieren. Spezielle Kenntnisse in den Bereichen Messtechnik und Steuergerätetechnik sind erforderlich.

Sie sollten sich mit INCA und INCA-MIP gleichermaßen gut auskennen, wenn Sie das MATLAB® API nutzen wollen. Darüber hinaus sollten Sie auch mit der Verwendung von Skripten in MATLAB® vertraut sein.

1.3 Klassifizierung von Warnhinweisen

Die Warnhinweise warnen vor Gefahren, die zu Verletzungen oder Sachschäden führen können.



GEFAHR

GEFAHR kennzeichnet eine gefährliche Situation, die zum Tod oder zu schweren Verletzungen führt, wenn sie nicht vermieden wird.



WARNUNG

WARNUNG kennzeichnet eine gefährliche Situation, die zum Tod oder zu schweren Verletzungen führen kann, wenn sie nicht vermieden wird.

**VORSICHT**

VORSICHT kennzeichnet eine gefährliche Situation, die zu leichten oder mittelschweren Verletzungen führen kann, wenn sie nicht vermieden wird.

HINWEIS

HINWEIS kennzeichnet eine Situation, die zu Sachschäden führen kann, wenn sie nicht vermieden wird.

1.4 Sicherheitshinweise

Beachten Sie die folgenden Sicherheitshinweise bei der Arbeit mit INCA und INCA Add-ons:

**WARNUNG****Gefahr von unerwartetem Fahrzeugverhalten**

Applikationstätigkeiten beeinflussen das Verhalten des Steuergeräts und der mit dem Steuergerät verbundenen Systeme.

Dies kann zu einem unerwarteten Verhalten des Fahrzeugs führen, wie zum Beispiel Abschalten des Motors sowie Bremsen, Beschleunigen oder Ausweichen des Fahrzeugs.

Führen Sie Applikationstätigkeiten nur durch, wenn Sie im Umgang mit dem Produkt geschult sind und die möglichen Reaktionen der damit verbundenen Systeme einschätzen können.



WARNUNG

Gefahr von unerwartetem Fahrzeugverhalten

Das Senden von Nachrichten über Bussysteme wie zum Beispiel CAN, FlexRay, LIN oder Ethernet beeinflusst das Verhalten der daran angeschlossenen Systeme.

Dies kann zu einem unerwarteten Verhalten des Fahrzeugs führen, wie zum Beispiel Abschalten des Motors sowie Bremsen, Beschleunigen oder Ausweichen des Fahrzeugs.

Senden Sie Nachrichten über Bussysteme nur dann, wenn Sie ausreichende Kenntnisse im Umgang mit dem jeweiligen Bussystem haben und das Verhalten der daran angeschlossenen Systeme einschätzen können.

Befolgen Sie die Anweisungen im ETAS Sicherheitshinweis und die Sicherheitsinformationen in der Online-Hilfe und den Benutzerhandbüchern. Öffnen Sie den ETAS Sicherheitshinweis im Hilfemenü von INCA unter **? > Sicherheitshinweis**.

1.5 Datenschutzhinweise

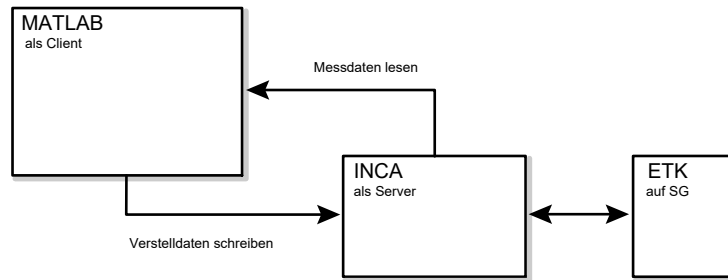
Beachten Sie, dass bei Nutzung von INCA-MIP personenbezogene Daten verarbeitet werden. Der Erwerber ist als Verantwortlicher für die Rechtskonformität der Verarbeitungen verpflichtet (gemäß Art. 4 Nr. 7 DSGVO/EU). Die ETAS GmbH haftet als Hersteller nicht für etwaiges Fehlverhalten im Umgang mit diesen Daten.

Weitere Informationen finden Sie in der INCA Online-Hilfe.

2 Über INCA-MIP

Das INCA-MIP Add-On (INCA MATLAB® Integration Package) stellt eine Programmierschnittstelle dar, über die die Funktionalität von INCA aus MATLAB® heraus angesteuert werden kann. Dabei verhält sich INCA als Client, der auf die Ressourcen von MATLAB®, in dem Fall den Server, zugreift.

Das folgende Diagramm zeigt einen typischen Anwendungsfall für das INCA-MIP API, in dem INCA zur Adressierung eines ETK verwendet wird.



Die folgende Übersicht fasst die INCA-Funktionalität zusammen, die aus MATLAB® heraus angesprochen werden kann.

Speicherseitenverwaltung

Es wird die Umschaltung zwischen Speicherseiten und das Herunterladen von Speicherseiten ins Steuergerät unterstützt.

Verstellen

Alle Verstellgrößen eines INCA Experiments können geändert werden. Für jedes Element sowie ggf. auch für die zugehörigen Stützstellenverteilungen können die Werte ausgelesen und neu geschrieben werden.

Messen

Alle Messgrößen eines INCA Experiments können ausgelesen werden. Darüber hinaus können Messungen aus MATLAB® heraus gestartet und angehalten werden. Auf alle Messdaten, die in INCA zur Verfügung stehen, kann auch in MATLAB® zugegriffen werden. Der Messdatendurchsatz an der INCA-MIP-Schnittstelle wurde optimiert.

Da INCA Mess- und Verstelldaten im Format Double übergibt, werden in MATLAB® keine Umrechnungsformeln zur Aufbereitung benötigt.

Die in diesem Dokument beschriebenen INCA API-Funktionen werden aus - MATLAB® Skripten, den sogenannten M-Dateien, heraus aufgerufen, in denen die gesamte Steuerung der INCA Experimente definiert werden kann.

In den folgenden Abschnitten wird die zugrundeliegende Architektur der INCA-MIP API, die vorhandenen API-Funktionen sowie die Installation beschrieben. Auf die Bedienung von INCA oder MATLAB® wird nicht weiter eingegangen.

2.1 INCA Definitionen

In der API-Beschreibung werden bestimmte Fachbegriffe verwendet, mit denen erfahrene Anwender von INCA vertraut sein sollten. Diese Begriffe sind im folgenden kurz definiert.

Verstellgröße

Eine Verstellgröße ist ein Element, das sowohl gelesen als auch geändert werden kann. Als Verstellgrößen können Skalare, Vektoren, Matrizen, Kennlinien und Kennfelder verwendet werden. Die zugehörigen Stützstellenverteilungen können ebenfalls ausgelesen und abgeändert werden.

Messdatensatz

Ein Messdatensatz besteht aus einem Zeitstempel und allen Messwerten, die zu dem entsprechenden Zeitpunkt innerhalb einer Signalgruppe erfasst wurden. Die Messdaten einer Signalgruppe bestehen wiederum aus allen Messdatensätzen, die über die Dauer eines Messvorgangs erzeugt werden.

Gerät

Ein Messgerät, mit dem in einem bestimmten Messraster eine Messgrößenerfassung erfolgen kann. Einige Messgeräte unterstützen darüber hinaus auch das Verstellen entsprechender Größen. So können beispielsweise SMB-Geräte nur zum Messen verwendet werden, während etwa der ETK sowohl zum Messen als auch zum Verstellen geeignet ist.

Messdaten

Alle Daten, die im Rahmen einer Messung für die einzelnen Messraster erfasst werden.

Messraster

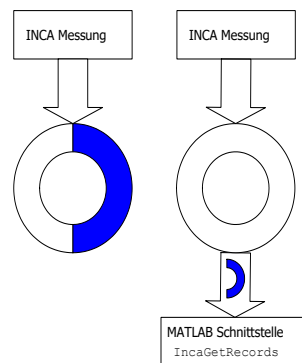
Abtastrate (Messfrequenz), mit der die Messung eines oder mehrerer Signale in einer Signalgruppe erfolgt.

Es ist möglich, zwei oder mehr Raster in einem so genannten Mehrfachraster zu kombinieren. Zu diesem Zweck werden die Rasternamen durch ein '+'-Zeichen verbunden, z. B. '10ms+100ms'. Bei Verwendung eines solchen Mehrfachrasters wird ein neues virtuelles Raster erstellt. Jedes Signal kann nur in genau einem Raster oder Mehrfachraster gemessen werden.

Ringpuffer

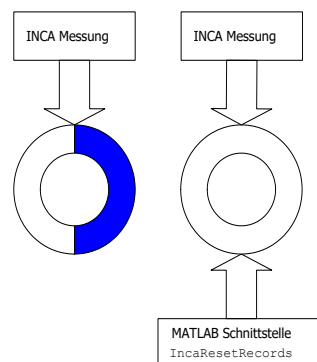
Um die zuverlässige Übertragung der Messdaten von INCA nach MATLAB® sicherzustellen, wird für jedes Messraster (Signalgruppe) ein Ringpuffer verwendet. Bei einer Online-Anzeige in INCA oder während des Aufzeichnens von Messdaten werden diese automatisch im Ringpuffer zwischengespeichert.

Mit Hilfe des Befehls `IncaGetRecords` können die Zeitstempel und Messdaten aus dem Ringpuffer in MATLAB® eingelesen werden:



Der Ringspeicher ist auf 1 MByte pro Signal begrenzt und kann je nach Messrate bis zu ca. 8 Sekunden Messdaten speichern. Nach dieser Zeit werden alte Messdaten überschrieben. Um zu vermeiden, dass dadurch Daten verloren gehen, muss der Befehl `IncaGetRecords` periodisch ausgeführt werden. Typischerweise wird dies ungefähr einmal je Sekunde vorgenommen.

Mit dem Befehl `IncaResetRecords` können die im Ringpuffer gehaltenen Zeitstempel und Messdaten aller Messraster entfernt werden. Alle im Ringpuffer gehaltenen Daten werden dadurch gelöscht.



Signal

Ein Signal ist ein Element, dessen Wert in INCA gemessen wird. Jedes Signal wird charakterisiert durch Datentyp (Boolesche, Integer, Float), Länge (1, 2, 4 oder 8 Byte) und die Umrechnungsformel. Die Umsetzung des physikalischen Messwerts auf der Implementierungsebene ist in der Umrechnungsformel spezifiziert.

Signalgruppe

Eine Signalgruppe besteht aus mehreren Einzelsignalen. Sie wird durch das Messraster charakterisiert, das für alle Einzelsignale gleich ist. Jede Signalgruppe hat einen eindeutigen Namen.

3 Installation

INCA-MIP ist eine Funktionserweiterung zu INCA.

In MATLAB® wird die Kommunikation mit anderen Programmen über dynamisch eingebundene Funktionsaufrufe bewerkstelligt, die sogenannten MEX-Dateien. Das INCA-MIP API besteht aus einer Sammlung solcher MEX-Dateien, die bei der Installation in entsprechende Unterverzeichnisse Ihres MATLAB® Programmverzeichnisses kopiert werden.

INCA-MIP liegt in zwei Varianten vor. Mit dem Befehlssatz aus dem INCA-MIP Base Paket kann direkt nach der Installation gearbeitet werden. Um den erweiterten Befehlssatz aus dem Paket INCA-MIP Extended zu nutzen, benötigen Sie einen Lizenzschlüssel. Eine Liste der API-Befehle sowie Erläuterungen finden Sie im Kapitel ["API-Funktionen"](#) auf Seite 17.

3.1 Systemvoraussetzungen

Damit Sie das INCA-MIP Add-On nutzen können, muss INCA auf Ihrem Computer installiert sein. Weitere Informationen zu den INCA Systemvoraussetzungen finden Sie im INCA Installationshandbuch.

Wenn Sie INCA Skripte für den Zugriff auf MATLAB® selbst entwickeln möchten, benötigen Sie außerdem eine MATLAB® Lizenz.

INCA-MIP für INCA V7.4 benötigt die folgenden Programmversionen:

- INCA V7.4 SPx



Info

Für die Installation dieser INCA-MIP Version ist INCA V7.4 erforderlich. Stellen Sie sicher, dass die INCA Releasenummer der INCA Installation mit der Versionsnummer des Add-on-Pakets INCA-MIP kompatibel ist.

Nach der Installation können Sie diese INCA-MIP Version verwenden, um mit jeder INCA V7.x Version zu arbeiten. ["INCA öffnen"](#) auf Seite [28](#).

- MATLAB® 64-Bit-Version 2016a oder höher (für MATLAB® integrierte Installation)

Weitere Informationen zu unterstützten INCA-Versionen erhalten Sie bei Ihrem MATLAB® Support.

3.2 INCA-MIP Installieren

Wenn Sie das Add-On installieren, müssen Sie zunächst entscheiden, welche Variante Sie einsetzen möchten. Die folgenden Varianten stehen zur Verfügung:

- **MATLAB® integrated installation**

Wählen Sie diese Option, wenn Sie genau eine MATLAB® Version für die Entwicklung von MATLAB® Skripten einsetzen.

- **Installation into ETASData**

Wählen Sie diese Option, wenn Sie lediglich kompilierte INCA-MIP Skripte aufrufen oder wenn Sie auf Ihrem Computer MATLAB® mit verschiedenen MATLAB® Versionen einsetzen möchten. Eine genauere Beschreibung finden Sie unten.

INCA-MIP installieren

Stellen Sie sicher, dass eine Vollversion von INCA auf Ihrem Computer installiert ist und dass die INCA-Version kompatibel mit der Version des INCA-MIP Add-Ons ist.

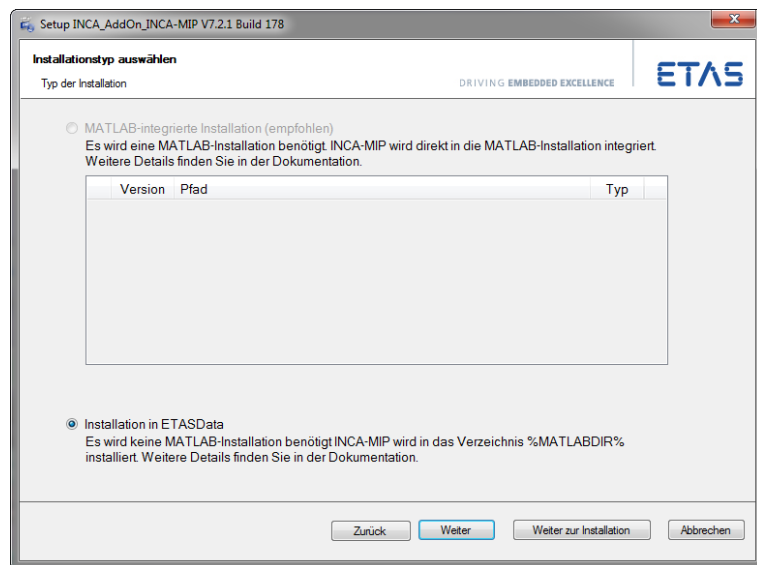
Wenn Sie Ihre eigenen MATLAB® Skripte für den Zugriff auf INCA entwickeln möchten, stellen Sie sicher, dass MATLAB® auf Ihrem Computer installiert ist und dass die Versionsnummer der MATLAB® Installation mit der Versionsnummer des INCA-MIP Zusatzpakets kompatibel ist.

1. Schließen Sie alle laufenden Programme.
2. Je nach Ihren unternehmensspezifischen Bestimmungen werden die Installationsdateien auf einem Netzlaufwerk oder auf einer DVD zur Verfügung gestellt.

Bei Verwendung der DVD startet die Installationsroutine automatisch. Wenn dies nicht der Fall ist, führen Sie die Datei `Auto-start.exe` auf der DVD manuell aus, klicken Sie auf Installation und wählen Sie die Installation INCA-MIP.

Wenn Sie das Programm von einem Netzlaufwerk installieren, führen Sie die Datei `setup.exe` aus.

3. Folgen Sie den Instruktionen des Installationsprogramms, um INCA-MIP auf Ihrem Computer zu installieren.
4. In der Installationsroutine werden Sie nach dem gewünschten Installationstyp gefragt:



5. Wenn Sie MATLAB[®] Skripte mit genau einer auf Ihrem PC installierten MATLAB[®] Version entwickeln möchten, wählen Sie die Option **MATLAB[®]** integrierte Installation.

oder

Wählen Sie die Option **Installation in ETASData**, wenn einer der folgenden Fälle zutrifft:

- Sie möchten INCA-MIP mit verschiedenen MATLAB[®] Versionen einsetzen.
In diesem Fall müssen Sie bei jeder einzelnen INCA-MIP Installation auf Ihrem PC das INCA-MIP Unterverzeichnis in das MATLAB[®] Toolbox-Verzeichnis kopieren, bevor Sie -Befehle nutzen können. Lesen Sie in der Benutzerdokumentation von MATLAB[®] nach, wie Sie Verzeichnisse zum MATLAB[®] Pfad hinzufügen.
- Sie möchten nur leicht verfügbare MATLAB[®] eigenständige ausführbare Dateien ausführen, die mit MATLAB[®] erstellt wurden.

Info

Die Installation in ETASData ist erforderlich, wenn Sie Programmdateien einsetzen, die INCA Kommandos zur Steuerung von MATLAB[®] enthalten. In diesem Fall benötigen Sie keine MATLAB[®] Lizenz. Die ausführbaren Dateien müssen von Entwicklern mit einer "Erstellung und Verteilung von ausführbaren Dateien mit dem MATLAB[®] Compiler" auf Seite 68 Installation bereitgestellt werden (siehe MATLAB[®]).

6. Setzen Sie die Installation fort.

INCA-MIP lizenzieren

Wenn Sie mit dem erweiterten API-Befehlssatz arbeiten möchten, benötigen Sie einen Software-Lizenzschlüssel.

Weitere Informationen zur Lizenzierung finden Sie in "Lizenzierung der Software" unten.

3.3 **Cache für MATLAB® Toolboxverzeichnisse aktualisieren**

Nach der Installation der INCA-MIP API sollten Sie zunächst den Cache für die MATLAB® Toolbox-Verzeichnisse aktualisieren, falls dieser Cache bei Ihrer MATLAB® Installation aktiviert wurde. Wenn Sie die Standardeinstellungen verwenden, ist das ab MATLAB® V6 der Fall, bei früheren Versionen ist der Cache deaktiviert. Das Aktualisieren des Cache ist erforderlich, damit die für die INCA-MIP API verwendeten Dateien in MATLAB® registriert werden.

Wie Sie vorgehen müssen, um den Cache für die MATLAB® Toolbox-Verzeichnisse zu aktualisieren, ist in Ihrer MATLAB® Benutzerdokumentation beschrieben.

3.4 **Cache für MATLAB®-Toolbox-Verzeichnisse deaktivieren**

Für die Arbeit mit INCA-MIP wird empfohlen, die Zwischenspeicherung der MATLAB®-Toolbox-Verzeichnisse zu deaktivieren. Andernfalls kann es zu Fehlfunktionen kommen, weil entweder das INCA-MIP Add-On oder einzelne, neu hinzugefügte Skriptdateien möglicherweise nicht gefunden werden.

Als Alternative zum Deaktivieren der Zwischenspeicherung können Sie bei aktivem Cache jeweils eine Aktualisierung des Cache wie oben beschrieben erzwingen. Zur Vermeidung von Fehlbedienungen wird jedoch empfohlen, den Cache während der Arbeit mit INCA-MIP zu deaktivieren.

Wie Sie vorgehen müssen, um den Cache für die MATLAB®-Toolbox-Verzeichnisse zu aktivieren oder zu deaktivieren, ist in Ihrer -Benutzerdokumentation beschrieben. MATLAB®

3.5 **Lizenzierung der Software**

Für die Benutzung von INCA benötigen Sie eine gültige Lizenz. Die für die Lizenzierung benötigte Lizenzdatei erhalten Sie entweder von Ihrem Tool-Koordinator oder über ein Self-Service-Portal auf der ETAS Internet-Seite unter <http://www.etas.com/support/licensing>. Für die Anforderung der Lizenzdatei müssen Sie die Aktivierungsnummer eingeben, die Sie von ETAS bei Ihrer Bestellung erhalten haben.

Wählen Sie im Windows Start Menü

E > ETAS > ETAS-Lizenzmanager.

Folgen Sie den Anweisungen des Dialogs. Für weitere Informationen zu z.B. ETAS Lizenzmodelle und Ausleihen einer Lizenz drücken Sie **F1** im ETAS License Manager.

4 API-Funktionen

INCA-MIP stellt eine Reihe von API-Funktionen für die Automatisierung von INCA-Prozessen zur Verfügung. Einige Funktionen sind im Paket INCA-MIP Base enthalten, andere stehen nur zur Verfügung, wenn Sie das Paket INCA-MIP Extended erworben haben.

Info

API-Funktionen, die nur im Paket INCA-MIP Extended enthalten sind, sind durch einen Software-Lizenzschlüssel geschützt. Sollten Sie Funktionen aus dem INCA-MIP Extended-Paket ohne gültigen Lizenzschlüssel verwenden, so wird beim Ausführen des entsprechenden MATLAB[®] Skriptes eine Exception geworfen.

Wir empfehlen Ihnen folgende Entwicklungsrichtlinie: Vor dem Ausführen von Kommandos aus dem Paket INCA-MIP Extended sollte mit Hilfe des Befehls `IncaIsLicenseValid` geprüft werden, ob eine gültige Lizenz vorliegt.

In der folgenden Tabelle sind alle INCA-MIP API-Funktionen aufgeführt, die im Add-on für INCA V7.4 zur Verfügung stehen. Es wird angegeben,

- ob die Funktion auch im Paket INCA-MIP Base oder nur in INCA-MIP Extended enthalten ist;
- ob die Funktion für Initialisierung, Messen, Verstellen, Speicherseitenverwaltung oder allgemeinere Aufgaben handelt;
- wo Sie im vorliegenden Dokument weitere Informationen zu der entsprechenden Funktion finden.

Funktion	Base ^a	Ext. ^b	Kategorie	Siehe
<code>IncaAddCalibrationElement</code>	x	x	Verstellen	auf Seite 54
<code>IncaAddMeasureElement</code>	x	x	Messen	auf Seite 36
<code>IncaBrowseCalibrationElements</code>		x	Verstellen	auf Seite 53
<code>IncaBrowseItemsInFolder</code>		x	Initialisieren	auf Seite 30
<code>IncaBrowseMeasureElements</code>		x	Messen	auf Seite 35

Funktion	Base ^a	Ext. ^b	Kategorie	Siehe
IncaClose		x	Initialisieren	auf Seite 28
IncaCopyPageFromTo	x	x	Speicherseitenverwaltung	auf Seite 64
IncaDatabaseImport		x	Initialisieren	auf Seite 29
IncaDownloadDifferences	x	x	Speicherseitenverwaltung	auf Seite 65
IncaDownloadPage	x	x	Speicherseitenverwaltung	auf Seite 64
IncaExecuteManualTrigger		x	Messen	auf Seite 52
IncaGetCalibrationValue	x	x	Verstellen	auf Seite 55
IncaGetCurrentPage		x	Speicherseitenverwaltung	auf Seite 63
IncaGetDatasetsForDevice		x	Verstellen	auf Seite 60
IncaGetDeviceProperties		x	Initialisieren	auf Seite 33
IncaGetDevices		x	Initialisieren	auf Seite 33
IncaGetHardwareStatus		x	Messen	auf Seite 49
IncaGetInstalledAddOnInfo	x	x	Allgemein	auf Seite 26
IncaGetInstalledProductInfo	x	x	Allgemein	auf Seite 25
IncaGetMeasureRatesForDevice		x	Messen	auf Seite 35
IncaGetProperties		x	Allgemein	auf Seite 27
IncaGetRecordingMode		x	Messen	auf Seite 42
IncaGetRecordingProperties		x	Messen	auf Seite 38

Funktion	Base ^a	Ext. ^b	Kategorie	Siehe
IncaGetRecordingState		x	Messen	auf Seite 52
IncaGetRecords	x	x	Messen	auf Seite 46
IncaGetRecordStruct		x	Messen	auf Seite 52
IncaGetVersion	x	x	Allgemein	auf Seite 27
IncaGroupDevices		x	Verstellen	auf Seite 62
IncaIsLicenseValid		x	Allgemein	auf Seite 25
IncaMessageIds	x	x	Allgemein	auf Seite 22
IncaIsPageWriteProtected	x	x	Memory-PageManager	auf Seite 63
IncaOpen	x	x	Initialisieren	auf Seite 28
IncaOpenDatabase	x	x	Initialisieren	auf Seite 29
IncaOpenExperiment	x	x	Initialisieren	auf Seite 32
IncaResetExperiment	x	x	Initialisieren	auf Seite 33
IncaResetRecords	x	x	Messen	auf Seite 49
IncaSetCalibrationMode		x	Verstellen	auf Seite 61
IncaSetCalibrationValue	x	x	Verstellen	auf Seite 57
IncaSetDatasetInDevice		x	Verstellen	auf Seite 60
IncaSetMeasureReadMode	x	x	Messen	auf Seite 45
IncaSetProjectAndDatasetInDevice		x	Initialisieren	auf Seite 31

Funktion	Base ^a	Ext. ^b	Kategorie	Siehe
IncaSetRecordingMode		x	Messen	auf Seite 43
IncaSetRecordingProperties		x	Messen	auf Seite 40
IncaSetTrigger		x	Messen	auf Seite 50
IncaShowMessages	x	x	Allgemein	auf Seite 25
IncaStartMeasurement	x	x	Messen	auf Seite 37
IncaStartRecording	x	x	Messen	auf Seite 44
IncaStopMeasurement	x	x	Messen	auf Seite 38
IncaStopRecording	x	x	Messen	auf Seite 45
IncaSwitchPage	x	x	Speicherseitenverwaltung	auf Seite 63
IncaUploadPages		x	Speicherseitenverwaltung	auf Seite 65
IncaWriteToFile		x	Verstellen	auf Seite 62

a Funktion ist im Paket INCA-MIP Base enthalten

b Funktion ist im Paket INCA-MIP Extended enthalten

Im Folgenden sind die Funktionsbeschreibungen entsprechend ihres Einsatzbereiches geordnet:

- ["Allgemeine Funktionen" auf Seite 22](#)
- ["Initialisierung" auf Seite 27](#)
- ["Messen und Aufzeichnen" auf Seite 34](#)
- ["Verstellen" auf Seite 53](#)
- ["Speicherseitenverwaltung" auf Seite 62](#)

Darüber hinaus werden eine Reihe von Beispieldateien zur Verfügung gestellt. Diese befinden sich in ["Kennenlernen der INCA-MIP API anhand von Beispieldateien"](#) auf der nächsten Seite.

Anwendungsbeispiele erhalten Sie unter ["Anwendungsbeispiele"](#) auf Seite 66.

 **Info**

Das INCA-MIP Add-On operiert grundsätzlich auf den globalen Einstellungen der INCA Benutzeroptionen. Weitere Informationen zu den Benutzeroptionen in INCA finden Sie in der INCA Benutzerdokumentation.

4.1 Kennenlernen der INCA-MIP API anhand von Beispieldateien

Zum Lieferumfang von INCA-MIP gehört auch eine Beispielsammlung. Die Beispieldateien werden zusätzlich zu den MEX-Dateien automatisch auf Ihrem Rechner installiert. Mit den Beispieldateien wird die Verwendung der INCA-MIP-Funktionen exemplarisch vorgeführt.

Die Beispielsammlung umfasst einige M-Dateien, die auf das INCA-MIP API zugreifen, und eine INCA-Datenbank, in der die in den Beispielskripten verwendeten Elemente bereits angelegt sind.

Die Beispieldateien werden bei der Installation in folgende Verzeichnisse kopiert (siehe ["INCA-MIP Installieren"](#) auf Seite 12):

- Bei der Installation im MATLAB®-Pfad:
M-Dateien: %MatlabVerz%\toolbox\matlab\demos
- Bei der Installation in ETASData:
M-Dateien: %EtasDataDir%\INCA-MIPx64
INCA Demo-Datenbank: %EtasDataVerz%\Database\db_matlabtest

Zur Arbeit mit den Beispieldateien müssen Sie zunächst INCA starten und dann die Beispieldatenbank darin öffnen. Es wird keine Hardware benötigt.

Die Funktion der M-Dateien ist im Folgenden beschrieben.

- `tOpen.m` – Stellt eine Verbindung zwischen INCA und MATLAB® her. Diese Funktion muss zu Beginn einer INCA-MIP-Sitzung verwendet werden, bevor das MATLAB® API verwendet werden kann.
- `tDummy.m` – Öffnet ein leeres INCA-Experiment, das eine Hardwarekonfiguration mit dem VADI-Testgerät benutzt. Das Skript legt im INCA-Experiment verschiedene Messgrößen an.
- `tEtKDummy.m` – Öffnet ein leeres INCA-Experiment, das eine Hardwarekonfiguration mit dem ETK-Testgerät benutzt. Das Skript legt im INCA-Experiment verschiedene Mess- und Verstellgrößen an. Darüber hinaus werden Arbeits- und Referenzseite heruntergeladen, die Mess- und Verstellgrößen ausgelesen und die Werte einzelner Verstellgrößen geändert.

- `tGetRecords(aGroupName).m` – Sammelt 20 s lang die Messdaten für die Signalgruppe `aGroupName` und stellt die Daten danach in MATLAB[®] dar. Diese Funktion kann sowohl im Zusammenhang mit dem VADI- als auch mit dem ETK-Beispiel verwendet werden (zu „Signalgruppen“ vgl. ["INCA Definitionen" auf Seite 10](#)).
- `tPrintDB({aFolder{, aFileId}}).m` – Gibt den gesamten Datenbankinhalt ab dem Datenbankverzeichnis `aFolder` auf die Datei `aFileId` aus. Wird die Funktion ohne Parameter aufgerufen, so wird die gesamte Datenbankhierarchie auf die Standardausgabe ausgegeben.
- `tHWStatus.m` – Beispiel für die Anwendung der API-Funktion `IncaGetHardwareStatus`. MATLAB[®] verbindet sich mit einem bereits geöffneten Experiment und wählt die erste gefundene Messgröße im ersten gefundenen Messgerät aus. Anschließend wird für 5 Minuten eine Messung durchgeführt. Tritt während der Messung eine Warnung oder ein Fehler auf, so wird der Messzyklus abgebrochen und nach einer Wartezeit von 5 Sekunden von Neuem gestartet.

4.2 Allgemeine Funktionen

Die folgenden allgemeinen API-Funktionen stehen zur Verfügung:

4.2.1 Message IDs über das INCA-MIP API zurückgeben

Beim Ausführen von Befehlen des INCA-MIP API kann es vorkommen, dass Fehlermeldungen ausgegeben werden.

Bei Verwendung von try/catch-Blöcken können detaillierte Informationen über die aufgetretenen Fehler zurückgegeben werden.

Beispiel

```
try,
    <command_1>
    ...
    <command_n>
catch,
    [msgstr,msgid] = lasterr
    ...
end
```

wobei:

<code>msgstr</code>	Beschreibung als Zeichenkette
<code>msgid</code>	<p>Message ID. Die folgenden Message IDs stehen zur Verfügung:</p> <ul style="list-style-type: none"> • <code>INCA:ParameterError</code> • <code>INCA:ReturnParameterError</code> • <code>INCA:WrongParameterValue</code> • <code>INCA:WrongParameterType</code> • <code>INCA:NaN</code> • <code>INCA:ExecutionError</code> • <code>INCA:ResourceError</code> • <code>INCA:RasterFull</code> • <code>INCA:ObjectIsWriteProtected</code> • <code>INCA:CallSequenceError</code> • <code>INCA:LicenseError</code> • <code>INCA:RecordingInProgress</code> • <code>INCA:NotInstalled</code> • <code>INCA:WrongVersion</code>

Hinweise für die Reaktion auf Fehler-IDs:

<code>INCA:ParameterError</code>	Es wird eine falsche Anzahl an Argumenten (Übergabeparametern) verwendet. (Parameter auf der rechten Seite)
<code>INCA:ReturnParameterError</code>	Es liegt eine falsche Anzahl an Rückgabeparametern vor. (Parameter auf der linken Seite)
<code>INCA:WrongParameterValue</code>	Eines der Argumente befindet sich außerhalb des gültigen Bereichs oder der Spezifikation.
<code>INCA:WrongParameterType</code>	Eines der Argumente besitzt den falschen Datentyp.
<code>INCA:NaN</code>	Einer der Parameter enthält einen 'not a number'-Wert.
<code>INCA:ExecutionError</code>	Während der Ausführung des Befehls ist ein Fehler aufgetreten. Wenn Sie die Funktion über die INCA-Benutzerschnittstelle ausführen, erhalten Sie möglicherweise mehr Informationen über die Ursache des Fehlers. Eventuell kann das Problem durch einen Neustart von INCA oder des Systems behoben werden.

<code>INCA:ResourceError</code>	Es ist nicht möglich, Betriebssystem-Ressourcen zu erhalten. Eventuell kann das Problem durch einen Neustart von INCA oder des Systems behoben werden.
<code>INCA:RasterFull</code>	Beim Versuch, eine Messung hinzuzufügen, ist die Messliste des entsprechenden Rasters bereits voll.
<code>INCA:ObjectIsWriteProtected</code>	Objekt kann nicht verstellt werden, da es schreibgeschützt ist.
<code>INCA:CallSequenceError</code>	Die Ausführung des Befehls setzt die Ausführung anderer Befehle voraus. Z.B. ist <code>IncaOpenExperiment</code> erforderlich, bevor <code>IncaAddMeasureElement</code> ausgeführt werden kann.
<code>INCA:LicenseError</code>	Für die Ausführung des Befehls mit den verwendeten Parametern wird eine Lizenz benötigt.
<code>INCA:RecordingInProgress</code>	Das angeforderte Kommando (z. B. Signale für Aufzeichnung mit <code>IncaSetRecordingMode</code> aktivieren oder deaktivieren) kann nicht ausgeführt werden, weil momentan eine Aufzeichnung läuft.
<code>INCA:NotInstalled</code>	Die angegebene INCA-Version kann nicht mit dem Kommando <code>IncaOpen</code> geöffnet werden, weil die entsprechende INCA-Version nicht installiert ist.
<code>INCA:WrongVersion</code>	Die angegebene INCA-Version kann aus einem der folgenden Gründe nicht mit dem Kommando <code>IncaOpen</code> geöffnet werden: <ul style="list-style-type: none">• INCA ist bereits gestartet und es wird ein <code>IncaOpen</code>-Kommando mit einem 'Version'-Parameter ausgeführt, der sich von der bereits geöffneten INCA-Version unterscheidet.• Es wird ein <code>IncaOpen</code>-Kommando aus INCA-MIP für INCA Vx.y mit einem 'Version'-Parameter mit Hauptversion != x ausgeführt.

4.2.2 Nachrichten während der Skriptausführung anzeigen

Name	<code>IncaShowMessages</code>	
Beschreibung	Aktiviert/deaktiviert die Informationsanzeige im MATLAB® Fenster während der Skriptausführung	
Syntax	<code>IncaShowMessages(trueOrFalse)</code>	
Rückgabewerte	keine	
Argumente	<code>trueOrFalse</code>	Nummerischer Parameter, dessen Wert Null oder ungleich Null ist. Bei Null wird die Anzeige von Informationen ausgeschaltet, andernfalls wird die Anzeige eingeschaltet (Vorgabe).
Beispiele	<pre>IncaShowMessages(0); IncaShowMessages;</pre>	

4.2.3 Vorhandensein einer gültigen INCA-MIP-Lizenz anzeigen (INCA-MIP Erweitert)

Name	<code>IncaIsLicenseValid</code>	
Beschreibung	Gibt einen Status zurück, der anzeigt, ob eine gültige INCA-MIP-Lizenz vorliegt oder nicht	
Syntax	<code>s = IncaIsLicenseValid</code>	
Rückgabewerte	<code>s</code>	Lizenzstatus: <ul style="list-style-type: none"> • 0: keine gültige Lizenz vorhanden • 1: gültige Lizenz vorhanden
Argumente	keine	
Beispiele	<code>status = IncaIsLicenseValid</code>	

4.2.4 Informationen zu allen installierten INCA-Versionen lesen

Name	<code>IncaGetInstalledProductInfo</code>	
Beschreibung	Stellt Informationen zu allen installierten INCA-Versionen bereit. Dieses Kommando kann vor <code>IncaOpen</code> ausgeführt werden.	
Syntax	<code>info = IncaGetInstalledProductInfo</code>	
Rückgabewerte	<code>info</code>	Informationen zu installierten INCA Versionen als MATLAB® Struktur für jede Installation, bestehend aus folgenden Einträgen:
	<code>info.name</code>	Produktname

<code>info.version</code>	Produktversionszeichenfolge
<code>info.hotfixVersion</code>	Installierter Hotfix als Zeichenfolge oder eine leere Zeichenfolge, wenn kein Hotfix installiert ist

Argumente keine

Beispiele `i = IncaGetInstalledProductInfo;`

4.2.5 Informationen zu allen installierten Produkt-Add-ons lesen

Name `IncaGetInstalledAddOnInfo`

Beschreibung Stellt Informationen zu allen installierten Add-ons für ein angegebenes Produkt bereit. Dieses Kommando kann vor `IncaOpen` ausgeführt werden.

Syntax `info = IncaGetInstalledAddOnInfo (productName, productVersion)`

Rückgabewerte `info` Informationen zu installierten Add-ons als MATLAB® Struktur für jede Installation, bestehend aus folgenden Einträgen:

<code>info.name</code>	Name des installierten Add-ons
<code>info.version</code>	Versionszeichenfolge des installierten Add-ons

Argumente

<code>productName</code>	Name des Produkts
<code>productVersion</code>	Die Produktversion als Zeichenfolge. Die gesamte Versionszeichenfolge ist relevant.

Beispiele

```
i = IncaGetInstalledAddOnInfo('INCA',
'V7.4.0');
i = IncaGetInstalledAddOnInfo('INCA',
'V7.4.3 Build 100');
```



Info

Stellen Sie sicher, dass Sie für die Eingabeargumente `productName` und `productVersion` genau den Namen und die Version eines Produkts verwenden, die von `IncaGetInstalledProductInfo` zurückgegeben werden.

4.2.6 INCA-Version auslesen

Name	<code>IncaGetVersion</code>	
Beschreibung	Gibt die Version von INCA zurück	
Syntax	<code>IncaGetVersion</code>	
Rückgabewerte	<code>version</code>	die INCA-Version als String
Argumente	keine	
Beispiele	<code>IncaGetVersion;</code>	

4.2.7 Eigenschaften von INCA lesen (INCA-MIP Erweitert)

Name	<code>IncaGetProperties</code>	
Beschreibung	Liest die Eigenschaften von INCA	
Syntax	<code>p = IncaGetProperties</code>	
Rückgabewerte	<code>p</code>	<p>Eigenschaften von INCA als MATLAB[®] struct, bestehend aus folgenden Einträgen:</p> <ul style="list-style-type: none"> • <code>p.databasePath</code> - Pfadname der geöffneten INCA Datenbank. Ist keine INCA Datenbank geöffnet, wird ein Leerstring zurückgegeben. • <code>p.dataPath</code> - Pfadname des INCA Datenverzeichnisses. • <code>p.installationPath</code> - Pfadname des INCA Installationsverzeichnisses. • <code>p.tempPath</code> - Pfadname des Verzeichnisses, das ETAS-Anwendungen für temporäre Dateien verwenden.
Argumente	keine	
Beispiele	<code>p = INCAGetProperties;</code>	

4.3 Initialisierung

Alle Mess- und Verstelloperationen in INCA werden im Kontext eines Experiments vorgenommen. Vor dem Öffnen eines Experiments muss zunächst eine Arbeitsumgebung mit gültiger Hardwarekonfiguration erstellt und zugewiesen werden.

Für die Arbeit mit dem INCA-MIP API muss ein leeres Experiment in der INCA Datenbank vorhanden sein, für das eine gültige Arbeitsumgebung und Hardwarekonfiguration vorliegt. Das Experiment kann von MATLAB aus geöffnet werden.

Zum Initialisieren stehen die nachfolgend beschriebenen API-Funktionen zur Verfügung.

4.3.1 INCA öffnen

Name	IncaOpen	
Beschreibung	Öffnet INCA und initialisiert die Verbindung zwischen INCA und MATLAB®	
Syntax	IncaOpen IncaOpen(version)	
Rückgabewerte	keine	
Argumente	version	INCA-Version, die geöffnet werden soll (optional). Syntax: <MajorVersion>.<MinorVersion>. INCA-MIP für INCA x.y kann nur Verbindungen zu INCA-Installationen mit derselben Hauptversion x herstellen.
Beispiele	IncaOpen; IncaOpen('7.3');	

4.3.2 INCA schließen (INCA-MIP Erweitert)

Name	IncaClose	
Beschreibung	Schließt INCA, nachdem mit IncaOpen erfolgreich eine Verbindung zu INCA hergestellt wurde.	
Syntax	IncaClose IncaClose(isDisconnectOnly)	
Rückgabewerte	keine	
Argumente	isDisconnectOnly	Legt fest, ob nur die Verbindung zwischen INCA und INCA aufgehoben wird oder ob MATLAB® außerdem geschlossen wird (optional). Mögliche Werte: <ul style="list-style-type: none"> • 0: hebt Verbindung zu INCA auf und schließt es (Default) • 1: hebt Verbindung zu INCA auf und lässt es geöffnet.
Beispiele	INCAclose; INCAclose(1);	

4.3.3 Datenbank öffnen

Name	IncaOpenDatabase	
Beschreibung	Öffnet die Datenbank im angegebenen Verzeichnis	
Syntax	IncaOpenDatabase ({pathName})	
Rückgabewerte	keine	
Argumente	pathName	Verzeichnis, in dem die zu öffnende Datenbank gespeichert ist. Wenn Sie kein Verzeichnis angeben, wird die aktuelle Datenbank geöffnet.
Beispiele	<pre>IncaOpenDatabase; % open current database IncaOpenDatabase ('c:\etasdata\mydatabase');</pre>	

4.3.4 Datenbank importieren (INCA-MIP Erweiterung)

Name	IncaDatabaseImport	
Beschreibung	Importiert eine Datenbank-Exportdatei (*.exp) in INCA. Bestehende gleichnamige Datenbankelemente werden dabei grundsätzlich überschrieben.	
Syntax	<pre>IncaDatabaseImport (path) name = IncaDatabaseImport (path) [name, type] = IncaDatabaseImport (path)</pre>	
Rückgabewerte	name	Array der vollständigen Pfadangaben aller importierter Datenbankelemente
		Verwenden Sie <code>deblank()</code> zum Zugriff auf einzelne Elemente innerhalb des Arrays: <pre>name2 = deblank(name(2,:))</pre>

	<code>type</code>	<p>Array aller Arten der importierten Datenbankelemente;</p> <p>Mögliche Werte:</p> <ul style="list-style-type: none"> • <code>Folder</code>: Verzeichnis in der Datenbank • <code>Experiment</code>: Experimentierumgebung • <code>Workspace</code>: Arbeitsumgebung • <code>Asap2Project</code>: SG-Projekt • <code>MeasurementCatalog</code>: Messgrößenkatalog • <code>CanDB</code>: ASAP2 CAN-DB
		<p>Verwenden Sie <code>deblank()</code> zum Zugriff auf einzelne Elemente innerhalb des Arrays:</p> <pre>type2 = deblank(type(2,:))</pre>
Argumente	<code>path</code>	Vollständiger Pfad der zu importierenden *.exp-Datei
Beispiele	<pre>names = IncaDatabaseImport('D:\ ETASData\[[[Undefined variable FM_import t.INCA_Version_Code]]]\export\Project0815.exp')</pre>	

4.3.5 Datenbankelemente lesen (INCA-MIP Erweitert)

Name	<code>IncaBrowseItemsInFolder</code>	
Beschreibung	Liest Datenbankelemente im angegebenen Datenbankverzeichnis mit Suchmuster	
Syntax	<code>[name, type] = IncaBrowseItemsInFolder(pattern, folderName)</code>	
Rückgabewerte	<code>name</code>	Liste der Namen der Datenbankelemente
	<code>type</code>	<p>Liste der Typen der Datenbankelemente:</p> <ul style="list-style-type: none"> • <code>Folder</code>: Datenbankverzeichnis • <code>Experiment</code>: Experiment • <code>Workspace</code>: Arbeitsumgebung • <code>Asap2Project</code>: SG-Projekt

Argumente	<code>pattern</code>	Suchmuster für die zu lesenden Datenbankinträge. Ein '*' steht für kein oder eine beliebige Anzahl beliebiger Zeichen. Ein '#' steht für genau ein beliebiges Zeichen. Alle anderen Zeichen müssen mit dem Datenbankeintrag übereinstimmen. Es wird nicht zwischen Groß- und Kleinschreibung unterschieden.
	<code>folderName</code>	Datenbankverzeichnis, in dem die Datenbankeinträge gelesen werden. Verzeichnishierarchien werden durch '\' getrennt. Ein Leerstring steht für die oberste Hierarchieebene.
Beispiele	<pre>[n, t]=IncaBrowseItemsInFolder ('*', 'DEFAULT\MyProject'); [name, type]=IncaBrowseItemsInFolder ('Prj*_ ##', '');</pre>	

4.3.6 Projekt und Datensatz einem Gerät zuordnen (INCA-MIP Erweitert)

Name	<code>IncaSetProjectAndDatasetInDevice</code>	
Beschreibung	Weist einem Gerät in einer bestehenden Arbeitsumgebung ein Projekt und einen Datensatz zu. Dies ist nur möglich, wenn zur Zeit kein Experiment geöffnet ist.	
Syntax	<code>IncaSetProjectAndDatasetInDevice(workspace, device, project, dataset)</code>	
Rückgabewerte	keine	
Argumente	<code>workspace</code>	Datenbankpfad der Arbeitsumgebung
	<code>device</code>	Name des Gerätes
	<code>project</code>	Datenbankpfad des Projektes
	<code>dataset</code>	Datenbankpfad des Datensatzes
Beispiele	<pre>IncaSetProjectAndDatasetInDevice('DEFAULT\ workspace', 'ETK:1', 'DEFAULT\Prj0815', 'Ds4711- \Ds4711_3')</pre>	

4.3.7 Experiment öffnen

Name	<code>IncaOpenExperiment</code>	
Beschreibung	Öffnet das angegebene Experiment in der Experimentierumgebung. Nach dem Öffnen eines Experiments können Sie über das INCA-MIP API nach Belieben die erforderlichen Mess- und Verstellgrößen hinzufügen.	
Syntax	<code>IncaOpenExperiment ({closeAllViewsFlag})</code> oder <code>IncaOpenExperiment (expFolderName, experimentName, workspaceFolderName, workspaceName {, closeAllViewsFlag})</code>	
Rückgabewerte	keine	
Argumente	<code>expFolderName</code>	Verzeichnis, in dem das Experiment abgelegt ist
	<code>experimentName</code>	Name des Experiments
	<code>workspaceFolderName</code>	Verzeichnis, in dem die Arbeitsumgebung abgelegt ist
	<code>workspaceName</code>	Name der Arbeitsumgebung
	<code>closeAllViewsFlag</code>	Schließt alle Mess- und Verstellfenster im ausgewählten Experiment. Zulässige Einstellungen: <ul style="list-style-type: none"> • 1: schließt alle Fenster (Vorgabe) • 0: lässt die Fenster unverändert
Beispiele	<pre>IncaOpenExperiment ('ExpFolder', 'MyExperiment', 'WorkspaceFolder', 'MyWorkspace');</pre>	

Info

Ist das Experiment beim Aufruf von `IncaOpenExperiment` bereits geöffnet, so sind die Argumente zur Spezifizierung der Umgebung optional.

Falls das Experiment noch nicht geöffnet ist, muss zunächst der Befehl `IncaOpenDatabase` aufgerufen werden, bevor `IncaOpenExperiment` ausgeführt werden kann.

4.3.8 Experiment zurücksetzen

Name	<code>IncaResetExperiment</code>
Beschreibung	Setzt das aktuelle Experiment zurück und schließt es. Sie können diese Funktion verwenden, um alle Variablen aus einem Experiment zu entfernen. Das Entfernen einzelner Größen wird zum gegenwärtigen Zeitpunkt nicht unterstützt.
Syntax	<code>IncaResetExperiment</code>
Rückgabewerte	keine
Argumente	keine
Beispiele	<code>IncaResetExperiment</code>

Info

Wurde das Experiment manuell und nicht über den Befehl MATLAB[®] geöffnet, setzt `IncaResetExperiment` das Experiment zurück, schließt aber nicht das Fenster. Sie müssen den Befehl `IncaOpenExperiment` ausführen, bevor Sie erneut auf das Experiment zugreifen können.

4.3.9 Geräte lesen (INCA-MIP Erweitert)

Name	<code>IncaGetDevices</code>				
Beschreibung	Liest alle im Experiment verfügbaren Geräte				
Syntax	<code>[name, type] = IncaGetDevices</code>				
Rückgabewerte	<table> <tr> <td><code>name</code></td> <td>Liste der Namen der Geräte</td> </tr> <tr> <td><code>type</code></td> <td>Liste der Gerätetypen: <ul style="list-style-type: none"> • <code>WorkbaseDevice</code>: Gerät mit Datensätzen • <code>MeasurementDevice</code>: Messgerät </td> </tr> </table>	<code>name</code>	Liste der Namen der Geräte	<code>type</code>	Liste der Gerätetypen: <ul style="list-style-type: none"> • <code>WorkbaseDevice</code>: Gerät mit Datensätzen • <code>MeasurementDevice</code>: Messgerät
<code>name</code>	Liste der Namen der Geräte				
<code>type</code>	Liste der Gerätetypen: <ul style="list-style-type: none"> • <code>WorkbaseDevice</code>: Gerät mit Datensätzen • <code>MeasurementDevice</code>: Messgerät 				
Argumente	keine				
Beispiele	<code>[name,type]=IncaGetDevices;</code>				

4.3.10 Eigenschaften von Geräten lesen (INCA-MIP Erweitert)

Name	<code>IncaGetDeviceProperties</code>
Beschreibung	Liest die Eigenschaften eines Gerätes
Syntax	<code>p = IncaGetDeviceProperties(deviceName)</code>
Rückgabewerte	<code>p</code> Eigenschaften des Geräts als MATLAB [®] struct, bestehend aus folgenden Einträgen:

<code>p.name</code>	Gerätename
<code>p.descriptionFile</code>	Pfadname der Beschreibungsdatei für das dem Gerät zugeordnete Projekt. Falls dem Gerät kein Projekt zugeordnet ist, ein Leerstring.
<code>p.binaryFile</code>	Pfadname der Binärdatei für das dem Gerät zugeordnete Projekt. Falls dem Gerät kein Projekt zugeordnet ist, ein Leerstring.
<code>p.projectDBPath</code>	Pfadname innerhalb der INCA-Datenbank des dem Gerät zugeordneten Projekts. Falls dem Gerät kein Projekt zugeordnet ist erscheint ein Leerstring.
<code>p.isWriteProtected</code>	<ul style="list-style-type: none"> • 0: Gerät hat keine Speicherseiten oder aktuelle Seite ist nicht schreibgeschützt • 1: Aktuelle Speicherseite ist schreibgeschützt
<code>p.isActive</code>	<ul style="list-style-type: none"> • 0: Gerät ist nicht verbunden oder nicht aktiv • 1: Gerät ist verbunden und aktiv
<code>p.isWorkbaseDevice</code>	<ul style="list-style-type: none"> • 0: Gerät hat keine Datensätze • 1: Gerät hat Datensätze

Argumente `deviceName` Name des Geräts

Beispiele `p = IncaGetDeviceProperties('Device');`

4.4 Messen und Aufzeichnen

Ein Signal oder eine Messgröße wird immer in einem Messraster für das jeweilige Messgerät erfasst. Jede Messgröße darf nur in einem Messraster enthalten sein. Zur Konfiguration eines Experiments werden zunächst die Messgrößen den einzelnen Messrastern zugeordnet.

Info

Namen von Elementen, Geräten, Signalen und Messrastern unterscheiden sich durch Groß- und Kleinschreibung.

4.4.1 Messgrößen auflisten (INCA-MIP Erweitert)

Name	<code>IncaBrowseMeasureElements</code>	
Beschreibung	Liest eine Liste der Messgrößen des Experiments mit Suchmuster und optional dem Gerätenamen	
Syntax	<pre>[name, type] = IncaBrowseMeasureElements (pattern, {deviceName}) [name] = IncaBrowseMeasureElements(pat- tern, {deviceName})</pre>	
Rückgabewerte	<code>name</code>	Liste der Namen der Messgrößen
	<code>type</code>	Liste der Typen der Messgrößen: <ul style="list-style-type: none"> • <code>Scalar</code>: Skalar • <code>Array</code>: Vektor • <code>Matrix</code>: Matrix
Argumente	<code>pattern</code>	Suchmuster für die zu lesenden Messgrößen. Ein '*' steht für kein oder eine beliebige Anzahl beliebiger Zeichen. Ein '#' steht für genau ein beliebiges Zeichen. Alle anderen Zeichen müssen mit dem Messgrößennamen übereinstimmen. Es wird nicht zwischen Groß- und Kleinschreibung unterschieden.
	<code>deviceName</code>	Name des Geräts
Beispiele	<pre>[n,t]=IncaBrowseMeasureElements('ign*', 'Device'); [name,type]=IncaBrowseMeasureElements ('*');</pre>	

4.4.2 Messraster lesen (INCA-MIP Erweitert)

Name	<code>IncaGetMeasureRatesForDevice</code>	
Beschreibung	Liest alle Messraster eines Gerätes	
Syntax	<pre>[name] = IncaGetMeasureRatesForDevice (deviceName)</pre>	
Rückgabewerte	<code>name</code>	Liste der Namen der Messraster
Argumente	<code>deviceName</code>	Name des Geräts
Beispiele	<pre>n=IncaGetMeasureRatesForDevice('Device'); name=IncaGetMeasureRatesForDevice('Dev');</pre>	

4.4.3 Messgröße zu Experiment hinzufügen

Name	IncaAddMeasureElement	
Beschreibung	Fügt eine Messgröße mit oder ohne angegebenes Messraster zu einem Experiment hinzu.	
Syntax	<pre>IncaAddMeasureElement(deviceName, groupName, signalName {, displayMode}) groupName = IncaAddMeasureElement(deviceName, [], signalName{, displayMode})</pre>	
Rückgabewerte	keine	
Argumente	deviceName	Name des Geräts
	groupName	Name des Messrasters. Es können mehrere Raster verwendet werden, indem Rasternamen durch ein '+'-Zeichen verbunden werden, z. B. '10ms+100ms'. Bei Verwendung eines solchen Mehrfachrasters wird ein neues virtuelles Raster erstellt. Jedes Signal kann nur in genau einem Raster oder Mehrfachraster gemessen werden. Als Gruppenname kann [] angegeben werden (siehe Hinweis unten).
	signalName	Name des Messsignals. Für Skalare genügt der Name; für Vektoren und Arrays muss der Index im Format [n] oder [n,m] an den Namen angehängt werden. Das erste Element hat den Index "null".
	displayMode	Anzeigemodus für das Element: <ul style="list-style-type: none"> • 1: Messgröße wird angezeigt (Standard) • 0: keine Anzeige
Beispiele	<pre>IncaAddMeasureElement('MyDevice', '10ms', 'Channel01', 0); IncaAddMeasureElement('ETK:1', '1.0ms', 'Matrix[2,1]'); group = IncaAddMeasureElement('CalcDev', [], 'MyCalcSig1');</pre>	

Info

Wenn das Messraster voll ist, wird die Messgröße nicht zum Raster hinzugefügt.

 **Info**

Wenn das Eingangsargument `groupName []` (d. h. leer) ist, wird die Signalgruppe auf folgende Weise ermittelt:

- Ist das Signal bereits Teil des Experiments, wird dessen vorhandener Signalgruppenname verwendet.
- Ist das Signal nicht Teil des Experiments, wird willkürlich irgendeine verfügbare Signalgruppe verwendet. Im Falle des berechneten Geräts (CalcDev, das für berechnete Signale genutzt wird) oder von CAN-Monitoring wird die für das betreffende Signal definierte Signalgruppe verwendet.

Da der Name der Signalgruppe für `IncaGetRecords`, `IncaGetRecordStruct` oder `IncaGetRecordCount` benötigt wird, wird er als optionaler linksseitiger Parameter zurückgegeben.

Beispiele:

```
groupName = IncaAddMeasureElement( 'CalcDev', [],
'MyCalcSig')
groupName = IncaAddMeasureElement( 'CAN-Monitoring:1',
[], 'nmot', 1)
```

 **Info**

Die maximale Anzahl der möglichen Signale, welche verwendet werden können, ist sowohl geräte- als auch protokollspezifisch. Die Anzahl der Signale wird zusätzlich durch die Größe des Pufferspeichers des Target Server Prozess bestimmt. Die maximale Größe des Pufferspeichers ist abhängig von der Abtastrate.

Beispiel:

Ein Signal mit 0,1ms Abtastrate benötigt >3 Megabyte an Datenvolumen. Daher liegt die maximale Anzahl an Signalen, die verwendet werden können, zwischen 400 und 600 Signalen. Langsamere Abtastraten erlauben das Hinzufügen von weiteren Signalen.

4.4.4 Messung starten

Name	<code>IncaStartMeasurement</code>
Beschreibung	Startet eine Messung in INCA
Syntax	<code>IncaStartMeasurement</code>

Rückgabewerte	keine
Argumente	keine
Beispiele	<code>IncaStartMeasurement;</code>

4.4.5 Messung beenden

Name	<code>IncaStopMeasurement</code>		
Beschreibung	Beendet die laufende Messung sowie ggf. die laufende Aufzeichnung in INCA		
Syntax	<code>IncaStopMeasurement{ (mdfFileName) }</code>		
Rückgabewerte	keine		
Argumente	<table> <tr> <td><code>mdfFileName</code></td> <td>Name der MDF-Datei, in der die Aufzeichnungsdaten gespeichert werden, wenn zugleich mit der Messung auch eine laufende Aufzeichnung beendet wird. Geben Sie immer den vollständigen Zugriffspfad für die Datei an (also z.B.: <code>'c:\mydata\store1.dat'</code>).</td> </tr> </table>	<code>mdfFileName</code>	Name der MDF-Datei, in der die Aufzeichnungsdaten gespeichert werden, wenn zugleich mit der Messung auch eine laufende Aufzeichnung beendet wird. Geben Sie immer den vollständigen Zugriffspfad für die Datei an (also z.B.: <code>'c:\mydata\store1.dat'</code>).
<code>mdfFileName</code>	Name der MDF-Datei, in der die Aufzeichnungsdaten gespeichert werden, wenn zugleich mit der Messung auch eine laufende Aufzeichnung beendet wird. Geben Sie immer den vollständigen Zugriffspfad für die Datei an (also z.B.: <code>'c:\mydata\store1.dat'</code>).		
Beispiele	<code>IncaStopMeasurement ('c:\mydata\store1.dat');</code>		

Info

Beenden Sie bei hohem Datenaufkommen eine laufende Aufzeichnung immer mit dem Befehl `IncaStopMeasurement (mdfFileName)`, um Datenverluste durch eine weiterlaufende Messung zu vermeiden. Verwenden Sie im Anschluss daran nochmals den Befehl `IncaGetRecords`, um die restlichen Messdaten nach MATLAB® zu übertragen.

4.4.6 Eigenschaften der Aufzeichnung lesen (INCA-MIP Erweitert)

Name	<code>IncaGetRecordingProperties</code>
Beschreibung	Liest die Eigenschaften der primären Ausgabedatei des Standardrekorders und die Dateierweiterung für das ausgewählte primäre Aufzeichnungsformat.
Syntax	<code>properties = IncaGetRecordingProperties</code>

Rückgabewerte	<code>properties</code>	Eigenschaften der Messdatenaufzeichnung als MATLAB® struct, bestehend aus folgenden Einträgen:
	<code>properties.fileName</code>	Dateiname der Aufzeichnung einschließlich des primären Ausgabeformats
	<code>properties.directory</code>	Verzeichnis für die Messdatei
	<code>properties.fileFormat</code>	Dateiformat der Messdatei; die folgenden Formatangaben sind möglich: <ul style="list-style-type: none"> • ETASBinary • #DiademATF • ETASAscii • ETASGroupAscii • ETASMATLABMFILE • ETASGroupMatlabM • FamosRecord • ETASMDF • ETASMDF4
	<code>properties.autoIncrement</code>	Angabe, ob der Name der Aufzeichnungsdatei automatisch inkrementiert werden soll: <ul style="list-style-type: none"> • 0: nicht automatisch inkrementieren • 1: automatisch inkrementieren
	<code>properties.comment</code>	Der Kommentar der Aufzeichnungsdatei. Er darf 1024 Zeichen minus Anzahl der Zeichen für den defaultComment nicht überschreiten.
	<code>properties.defaultComment</code>	Der von INCA generierte Vorgabe-Kommentar

<code>properties.company</code>	Firmenname im Kopf der Aufzeichnungsdatei
<code>properties.project</code>	Projektname im Kopf der Aufzeichnungsdatei
<code>properties.user</code>	Anwendername im Kopf der Aufzeichnungsdatei
<code>properties.vehicle</code>	Fahrzeugname im Kopf der Aufzeichnungsdatei

Argumente keine

Beispiele `properties = IncaGetRecordingProperties;`

4.4.7 Eigenschaften der Aufzeichnung ändern (INCA-MIP Erweitert)

Name `IncaSetRecordingProperties`

Beschreibung Ändert eine oder mehrere Eigenschaften der nächsten Messdatenaufzeichnung

Syntax `IncaSetRecordingProperties(properties)`

Rückgabewerte keine

Argumente `properties` Eigenschaften der Messdatenaufzeichnung als MATLAB struct, besteht aus einer beliebigen Kombination der folgenden Einträge:

`properties.fileName` Dateiname der Aufzeichnung

`properties.directory` Verzeichnis für die Messdatei

<code>properties.fileFormat</code>	Dateiformat der Messdatei; die folgenden Formatangaben sind möglich: <ul style="list-style-type: none"> • ETASBinary • #DiademATF • ETASAscii • ETASGroupAscii • ETASMATLABMFILE • ETASGroupMatlabM • FamosRecord • ETAS MDF • ETAS MDF4
<code>properties.autoIncrement</code>	Angabe, ob der Name der Aufzeichnungsdatei automatisch inkrementiert werden soll: <ul style="list-style-type: none"> • 0: nicht automatisch inkrementieren • 1: automatisch inkrementieren
<code>properties.comment</code>	Der Kommentar der Aufzeichnungsdatei darf 1024 Zeichen minus Anzahl der Zeichen für den <code>defaultComment</code> nicht überschreiten.
<code>properties.company</code>	Firmenname im Kopf der Aufzeichnungsdatei
<code>properties.project</code>	Projektname im Kopf der Aufzeichnungsdatei
<code>properties.user</code>	Anwendername im Kopf der Aufzeichnungsdatei
<code>properties.vehicle</code>	Fahrzeugname im Kopf der Aufzeichnungsdatei

Beispiele

```
properties.user = 'Michael';
properties.project = 'K70';
IncaSetRecordingProperties(properties);
```

Info

Wenn Sie die Aufzeichnung mit `IncaSetRecordingProperties` einstellen, sollten Sie eine Aufzeichnung nicht mit `IncaStopRecording` beenden. Stattdessen können Sie `IncaSetTrigger` verwenden. Zum Beenden der Aufzeichnung können Sie jede beliebige Triggerbedingung verwenden.

Beispiel:

Beenden einer Aufzeichnung nach einer festen Aufzeichnungsdauer:

```
TIMEDURATION_SECONDS = 25;
IncaSetTrigger('none', 'none', 'none', 'none',
TIMEDURATION_SECONDS);
IncaStartRecording;
% Recording automatically stops after TIMEDURATION_
SECONDS seconds
```

Beenden einer Aufzeichnung nach einem manuellen Trigger:

```
IncaSetTrigger('none', 'manual');
IncaStartRecording;
% Do anything until the stop trigger condition is met
...
IncaExecuteManualTrigger('stop');
```

4.4.8 Aufzeichnungsmodus lesen (INCA-MIP Erweitert)

Name	<code>IncaGetRecordingMode</code>
Beschreibung	Zeigt an, ob ein Signal im Standardrekorder aufgezeichnet wird oder nicht.
Syntax	<code>IncaGetRecordingMode(deviceName, signalName)</code>
Rückgabewerte	<p>Aufzeichnungsmodus für den Standardrekorder:</p> <ul style="list-style-type: none"> • 0: Signal wird nicht im Standardrekorder aufgezeichnet • 1: Signal wird im Standardrekorder aufgezeichnet

Argumente	deviceName	Name des Geräts
	signalName	Name des Messsignals. Für Skalare genügt der Name; für Vektoren und Arrays muss der Index im Format [n] oder [n,m] an den Namen angehängt werden. Das erste Element hat den Index "null".
Beispiele	<pre>m = IncaGetRecordingMode('ETK:1', 'hfm'); mode = IncaGetRecordingMode('CalcDev', 'MyCalcSig1');</pre>	

Info

Vor Verwendung von `IncaSetRecordingMode` muss das Signal mit `IncaAddMeasureElement` hinzugefügt werden.

4.4.9 Aufzeichnungsmodus festlegen (INCA-MIP Erweitert)

Name	<code>IncaSetRecordingMode</code>	
Beschreibung	<p>Aktiviert oder deaktiviert die Aufzeichnung eines Signals im Standardrekorder. Die Aufzeichnung kann nur für Signale deaktiviert werden, die im INCA-Experiment angezeigt werden.</p> <p>Vor Ausführung dieses Kommandos muss das Signal mit <code>IncaAddMeasureSignal</code> zum Experiment hinzugefügt werden.</p>	
Syntax	<pre>IncaSetRecordingMode(deviceName, signalName, recordingMode)</pre>	
Rückgabewerte	keine	
Argumente	deviceName	Name des Geräts

signalName	Name des Messsignals. Für Skalare genügt der Name; für Vektoren und Arrays muss der Index im Format [n] oder [n,m] an den Namen angehängt werden. Das erste Element hat den Index "null".
recordingMode	Aufzeichnungsmodus für den Standardrekorder: <ul style="list-style-type: none"> • 0: Messgröße wird aus dem Standardrekorder entfernt • 1: Messgröße wird zum Standardrekorder hinzugefügt

Beispiele

```
IncaSetRecordingMode('ETK:1', 'hfm', 1);
IncaSetRecordingMode('CalcDev', 'MyCalcSig1', 0);
```

 **Info**

Vor Verwendung von `IncaSetRecordingMode` muss das Signal mit `IncaAddMeasureElement` hinzugefügt werden.

4.4.10 Aufzeichnung starten

Name	<code>IncaStartRecording</code>
Beschreibung	Startet eine Aufzeichnung in INCA. Diese Funktion kann entweder nach oder anstelle von <code>IncaStartMeasurement</code> verwendet werden. Nach dem Start einer Messung oder Aufzeichnung stehen die Messdaten auch in MATLAB® zur Verfügung.
Syntax	<code>IncaStartRecording</code>
Rückgabewerte	keine
Argumente	keine
Beispiele	<code>IncaStartRecording;</code>

4.4.11 Aufzeichnung beenden

Name	<code>IncaStopRecording</code>	
Beschreibung	Beendet die laufende Aufzeichnung in INCA. Die Messung läuft weiter und muss ausdrücklich über <code>IncaStopMeasurement</code> beendet werden. Es ist möglich, während fortlaufender Messung die Aufnahme mehrfach ein- und auszuschalten.	
Syntax	<code>IncaStopRecording (mdfFileName)</code>	
Rückgabewerte	keine	
Argumente	<code>mdfFileName</code>	Name der MDF-Datei, in der die Aufzeichnungsdaten gespeichert werden. Geben Sie immer den vollständigen Zugriffspfad für die Datei an (also z.B.: <code>'c:\mydata\store1.dat'</code>).
Beispiele	<code>IncaStopRecording ('c:\mydata\store1.dat');</code>	

4.4.12 Datenlesemodus einstellen (Online/Offline-Daten)

Name	<code>IncaSetMeasureReadMode</code>	
Beschreibung	<p>Legt fest, von welcher Quelle die Messdaten an MATLAB[®] übertragen werden. Die Daten werden entweder zunächst in INCA aufbereitet und dann an MATLAB[®] übertragen (Offline-Daten) oder unmittelbar aus dem Gerätepuffer ausgelesen (Online-Daten).</p> <p>Bei der Messdatenanzeige stehen für einige Geräte, wie die ES1303-Karte und die Geräte der ES6xx-Reihe, keine Offline-Daten zur Verfügung. Es wird empfohlen, bei der Messdatenanzeige stets mit Online-Daten zu arbeiten.</p> <p>Bei der Messdatenaufzeichnung kann sowohl mit Online- als auch mit Offline-Daten gearbeitet werden. In beiden Fällen ergibt sich aus der Optimierung des Übertragungsverhaltens für die Darstellung jeweils eine besondere Charakteristik der Ergebnisse: Online-Daten können bei hoher Auslastung unvollständig sein, während Offline-Daten bei der Messdatenaufzeichnung immer vollständig sind. Offline-Daten können aber bei hoher Auslastung nur zeitversetzt übertragen werden. Es wird empfohlen, bei der Messdatenaufzeichnung mit Offline-Daten zu arbeiten.</p>	
Syntax	<code>IncaSetMeasureReadMode (measureReadMode)</code>	

Rückgabewerte	keine
Argumente	<p><code>measureReadMode</code> Numerischer Parameter, dessen Wert die Datenquelle angibt. Zulässige Einstellungen:</p> <ul style="list-style-type: none"> • 1: Offline-Daten • 0: Online-Daten (Vorgabe)
Beispiele	<code>IncaSetMeasureReadMode (0) ;</code>

4.4.13 Messdaten lesen

Name	<code>IncaGetRecords</code>
Beschreibung	<p>Überträgt Messdaten an MATLAB®. Die Messdaten jeder Signalgruppe werden in einem dedizierten Ringpuffer gespeichert, der Daten für eine Messzeit von bis zu 30 Sekunden aufnehmen kann. Die Messdaten werden in Gruppen aus MATLAB® abgerufen. Deshalb sollten Sie die Ausführung Ihres Skripts in MATLAB® stoppen, nachdem Sie die Messdaten abgerufen haben. Je größer die Datenmenge, die bei jedem Vorgang übertragen wird, desto effizienter ist die Datenübertragung.</p> <p>Diese Funktion überträgt eine vorgegebene Anzahl von Datensätzen für das angegebene Messraster.</p> <p>Für weitere Informationen zum Ringpuffer sei auf den entsprechenden Eintrag in <u>"INCA Definitionen"</u> auf Seite 10 verwiesen.</p>
Syntax	<pre>[time, data {,state}] = IncaGetRecords (deviceName, groupName, maxRecords {{,la- test{, exact}}})</pre>
Rückgabewerte	<p><code>time</code> Ein Vektor mit den Zeitmarken der übertragenen Datensätze. Diese Variable enthält eine maximale Anzahl von <code>m</code>-Werten, wobei $m \leq \text{maxRecords}$.</p> <p><code>data</code> Ein zweidimensionales Array mit den Datenwerten für jede Messgröße in der Reihenfolge, in der sie mit <code>IncaAddMeasureElement</code> zum Experiment hinzugefügt wurde. In diesem Array gibt die Dimension <code>m</code> die Anzahl übertragener Datensätze wieder, während <code>n</code> die Anzahl der Messraster angibt.</p>

	state	<p>Optionaler Rückgabeparameter:</p> <ul style="list-style-type: none"> • 0: Erfolg. Datensätze empfangen • 1: Erfassung wird nicht ausgeführt. Keine Datensätze empfangen • 2: Nicht genug Datensätze. Keine Datensätze empfangen. Diese Rückgabe ist nur bei exakt = 1 möglich.
Argumente	deviceName	Name des Geräts
	groupName	<p>Name des Messrasters.</p> <p>Es können mehrere Raster verwendet werden, indem Rasternamen durch ein '+'-Zeichen verbunden werden, z. B. '10ms+100ms'. Bei Verwendung eines solchen Mehrfachrasters wird ein neues virtuelles Raster erstellt.</p> <p>Jedes Signal kann nur in genau einem Raster oder Mehrfachraster gemessen werden.</p>
	maxRecords	(Maximale) Anzahl der zu empfangenden Datensätze. Siehe auch Parameter <i>exact</i> . Die Anzahl, die Sie hier eingeben, ist die Dimension <i>m</i> für die oben genannte time- oder data-Variable. Wenn diese Dimension den Wert von <i>maxRecords</i> erreicht, werden nicht alle vorhandenen Datensätze gelesen, sodass der Ringpuffer überlaufen kann.
	latest	Gibt an, ob die ältesten oder letzten <i>n</i> Datensätze empfangen werden.
	exact	Gibt an, dass auch dann Datensätze empfangen werden, wenn <i>n</i> < maxDatensätze im Ringpuffer verfügbar sind oder der Ringpuffer unverändert bleiben sollte.
Beispiele	<pre>[t, d] = IncaGetRecords('ETK:1', '100ms', 500); data = [data; d]; time = [time; t]; [t, d, s] = IncaGetRecords('ETK:1', '100ms', 25, 1, 1);</pre> <p>Für die Verwendung dieses Code-Fragments in einem größeren Kontext sei auf "Beispiel 2" auf Seite 66 verwiesen.</p>	

 **Info**

Das in `IncaGetRecords` verwendete Raster korrespondiert direkt mit dem in `IncaAddMeasureElement` verwendeten Raster, d. h. Sie müssen das gleiche Raster oder Multi-Raster verwenden.

Beispiel:

```
IncaAddMeasureElement('ETK test device:1','RASTER_
A+RASTER_B', 'N')
IncaAddMeasureElement('ETK test device:1','RASTER_
A+RASTER_B', 'n')
[t,d]= IncaGetRecords('ETK test device:1','RASTER_
A+RASTER_B',15)
```

Zum Überprüfen der Rasterzuordnung von Signalen können Sie das Kommando `IncaGetRecordStruct` verwenden.

Beispiel:

```
l=IncaGetRecordStruct('ETK test device:1', 'RASTER_
A+RASTER_B')
```

 **Info**

Die folgenden Parameterverknüpfungen werden wie folgt ausgeführt:

- `latest = 0, exact = 0`: (Standard)
Gibt die ältesten bis zu `maxRecords` Datensätze aus dem Ringpuffer zurück. Alle neueren Datensätze bleiben unverändert.
- `latest = 1, exact = 0`:
Gibt die letzten bis zu `maxRecords` Datensätze aus dem Ringpuffer zurück. Alle älteren Datensätze werden absichtlich zurückgewiesen.
- `latest = 0, exact = 1`:
Gibt die ältesten `maxRecords` Datensätze aus dem Ringpuffer zurück. Alle neueren Datensätze bleiben unverändert. Wenn nur `n < maxRecords` Datensätze im Ringpuffer verfügbar sind, wird nichts empfangen.
- `latest = 1, exact = 1`:
Gibt die letzten `maxRecords` Datensätze aus dem Ringpuffer zurück. Alle älteren Datensätze werden absichtlich zurückgewiesen. Wenn nur `n < maxRecords` Datensätze im Ringpuffer verfügbar sind, wird nichts empfangen.

4.4.14 Ringpuffer zurücksetzen

Name	IncaResetRecords
Beschreibung	Setzt den Ringpuffer für alle Messraster zurück. Diese Funktion kann auch während laufender Messung zum Zurücksetzen aller Ringpuffer verwendet werden. Beim Start einer Messung oder Aufzeichnung werden sie automatisch zurückgesetzt, ein expliziter Aufruf ist nicht erforderlich. Für weitere Informationen zum Ringpuffer sei auf den entsprechenden Eintrag in <u>"INCA Definitionen"</u> auf Seite 10 verwiesen.
Syntax	IncaResetRecords
Rückgabewerte	keine
Argumente	keine
Beispiele	IncaResetRecords;

4.4.15 Hardware-Status lesen (INCA-MIP Erweitert)

Name	IncaGetHardwareStatus				
Beschreibung	Liest den aktuellen Hardware-Status während der Messung oder Aufzeichnung				
Syntax	[status, message] = IncaGetHardwareStatus				
Rückgabewerte	<table> <tr> <td>status</td> <td>aktueller Hardware-Status: <ul style="list-style-type: none"> • 0: Status ok • 1: Warnung • 2: Fehler </td> </tr> <tr> <td>message</td> <td>Falls status den Wert 1 oder 2 liefert, gibt message eine Beschreibung der Warnung oder des Fehlers zurück.</td> </tr> </table> <p>Wenn status den Wert 1 oder 2 liefert, muss die Messung oder Messdatenaufzeichnung beendet werden, bevor erneut IncaGetHardwareStatus aufgerufen werden kann. Siehe auch Beispielskript <code>thwstatus.m</code> unter <u>"Kennenlernen der INCA-MIP API anhand von Beispieldateien"</u> auf Seite 21.</p>	status	aktueller Hardware-Status: <ul style="list-style-type: none"> • 0: Status ok • 1: Warnung • 2: Fehler 	message	Falls status den Wert 1 oder 2 liefert, gibt message eine Beschreibung der Warnung oder des Fehlers zurück.
status	aktueller Hardware-Status: <ul style="list-style-type: none"> • 0: Status ok • 1: Warnung • 2: Fehler 				
message	Falls status den Wert 1 oder 2 liefert, gibt message eine Beschreibung der Warnung oder des Fehlers zurück.				
Argumente	keine				
Beispiele	[s, m] = IncaGetHardwareStatus;				

4.4.16 Trigger definieren (INCA-MIP Erweitert)

Name	IncaSetTrigger	
Beschreibung	Stellt die Triggerbedingung ein, die beim Start einer Messung oder Aufzeichnung mit <code>IncaStartMeasurement</code> oder <code>IncaStartRecording</code> gelten soll.	
Syntax	<code>IncaSetTrigger(startTrigger{, stopTrigger{, preTriggerTime{, postTriggerTime{, duration}}})</code>	
Rückgabewerte	keine	
Argumente	<code>startTrigger</code>	Bedingung für den Start-Trigger: <ul style="list-style-type: none"> • <code>manual</code> für einen manuellen Start-Trigger • <code>none</code> falls kein Start-Trigger verwendet werden soll
	<code>stopTrigger</code>	Bedingung für den Stopp-Trigger: <ul style="list-style-type: none"> • <code>manual</code> für einen manuellen Stop-Trigger • <code>none</code> falls kein Stop-Trigger verwendet werden soll (Vorgabe)
	<code>preTriggerTime</code>	Triggervorlaufzeit in Sekunden <ul style="list-style-type: none"> • <code>none</code> falls diese nicht spezifiziert werden soll (Vorgabe)
	<code>postTriggerTime</code>	Triggernachlaufzeit in Sekunden <ul style="list-style-type: none"> • <code>none</code> falls diese nicht spezifiziert werden soll (Vorgabe)
	<code>duration</code>	Dauer der Messung oder Aufzeichnung in Sekunden <ul style="list-style-type: none"> • <code>none</code> falls diese nicht spezifiziert werden soll (Vorgabe); Die Aufzeichnungsdauer ist in diesem Fall endlos.
Beispiele	<pre>IncaSetTrigger('nmot\ETK:1 > 2000', 'none', 2.0, 3.0) IncaSetTrigger('none', 'none', 'none', 'none', 360)</pre>	

In der folgenden Tabelle sind alle unterstützten Kombinationen von Eingangsparametern aufgeführt. Andere Kombinationen führen zu einer Ausnahme.

Kombinationen von Eingangsparametern

Trigger-Funktionalität	startTrigger	stopTrigger	preTriggerTime	postTriggerTime	duration
Aufzeichnung mit fester Aufzeichnungsdauer	'none '	'none '	'none '	'none '	value
Aufzeichnung mit manuellem Start-Trigger, Trigger-Vorlaufzeit und manuellem Stopp-Trigger	'manual '	'manual '	value	'none '	'none '
Aufzeichnung mit manuellem Start-Trigger, Trigger-Vorlaufzeit und Trigger-Nachlaufzeit	'manual '	'none '	value	value	'none '
Aufzeichnung mit manuellem Start-Trigger, Trigger-Vorlaufzeit und Stopp-Trigger-Bedingung	'manual '	value	value	'none '	'none '
Aufzeichnung mit manuellem Stop-Trigger	'none '	'manual '	'none '	'none '	'none '
Aufzeichnung mit Start-Trigger-Bedingung und fester Aufzeichnungsdauer	value	'none '	'none '	'none '	value
Aufzeichnung mit Start-Trigger-Bedingung und manuellem Stopp-Trigger	value	'manual '	'none '	'none '	'none '
Aufzeichnung mit Start-Trigger-Bedingung, Trigger-Vorlaufzeit und manuellem Stopp-Trigger	value	'manual '	value	'none '	'none '
Aufzeichnung mit Start-Trigger-Bedingung, Trigger-Vorlaufzeit und Trigger-Nachlaufzeit	value	'none '	value	value	'none '
Aufzeichnung mit Start-Trigger-Bedingung, Trigger-Vorlaufzeit und Stopp-Trigger-Bedingung	value	value	value	'none '	'none '

4.4.17 Manuellen Trigger auslösen (INCA-MIP Erweitert)

Name	<code>IncaExecuteManualTrigger</code>	
Beschreibung	Löst einen manuellen Start- oder Stopp-Trigger aus. Dieser Befehl wird nur wirksam, wenn zuvor der Befehl <code>IncaSetTrigger</code> ausgeführt wurde, wobei einer der Parameter <code>startTrigger</code> oder <code>stopTrigger</code> auf den Wert <code>manual</code> gesetzt wurde.	
Syntax	<code>IncaExecuteManualTrigger (type)</code>	
Rückgabewerte	keine	
Argumente	<code>type</code>	Triggertyp: <ul style="list-style-type: none"> • <code>start</code> zum Auslösen eines manuellen Start-Triggers • <code>stop</code> zum Auslösen eines manuellen Stopp-Triggers
Beispiele	<code>IncaExecuteManualTrigger ('start')</code>	

4.4.18 Aufzeichnungsstatus lesen (INCA-MIP Erweitert)

Name	<code>IncaGetRecordingState</code>	
Beschreibung	Liest den aktuellen Aufzeichnungsstatus	
Syntax	<code>result = IncaGetRecordingState</code>	
Rückgabewerte	<code>result</code>	Aufzeichnungsstatus: <ul style="list-style-type: none"> • 0: Aufzeichnung findet nicht statt. • 1: Aufzeichnung findet statt oder Warten auf Trigger
Argumente	keine	
Beispiele	<code>s = IncaGetRecordingState</code>	

4.4.19 Liste von Messgrößen lesen (INCA-MIP Erweitert)

Name	<code>IncaGetRecordStruct</code>	
Beschreibung	Liest eine Liste von Messgrößen, die zum Messen oder Aufzeichnen ausgewählt wurden. Die Liste gibt die Namen der Messgrößen in der Reihenfolge zurück, in der sie mit <code>IncaAddMeasureElement</code> zugewiesen wurden.	
Syntax	<code>list = IncaGetRecordStruct (device, groupName)</code>	
Rückgabewerte	keine	

Argumente	<code>device</code>	Name des Gerätes
	<code>groupName</code>	Name des Messrasters Es ist möglich, mehrere Raster durch Verkettung mit einem Pluszeichen anzugeben, z.B. '10ms+100ms'.
Beispiele	<pre>l = IncaGetRecordStruct('ETK:1', '10ms'); list = IncaGetRecordStruct('device1', 'Syn- cro');</pre>	

4.5 Verstellen

Verstelloperationen können mit Skalaren, Kennlinien und -feldern sowie den zugehörigen Stützstellenverteilungen durchgeführt werden. In jedem Experiment kann eine beliebige Anzahl von Verstellgrößen definiert werden.

Info

Beachten Sie, dass die Namen aller Verstellgrößen sich auch durch Groß- und Kleinschreibung unterscheiden.

4.5.1 Verstellgrößen auflisten (INCA-MIP Erweitert)

Name	<code>IncaBrowseCalibrationElements</code>	
Beschreibung	Liest eine Liste der Verstellgrößen des Experiments mit Suchmuster und optional dem Gerätenamen	
Syntax	<pre>[name, type] = IncaBrowseCalibrationElements (pattern, {deviceName}) name = IncaBrowseCalibrationElements (pattern, {deviceName})</pre>	
Rückgabewerte	<code>name</code>	Liste der Namen der Verstellgrößen
	<code>type</code>	Liste der Typen der Verstellgrößen: <ul style="list-style-type: none"> • <code>Distribution</code>: Stützstellenverteilung • <code>OneDTable</code>: Kennlinie • <code>TwoDTable</code>: Kennfeld • <code>Scalar</code>: Skalar • <code>Array</code>: Vektor • <code>Matrix</code>: Matrix

Argumente	<code>pattern</code>	Suchmuster für die zu lesenden Verstellgrößen. Ein '*' steht für kein oder eine beliebige Anzahl beliebiger Zeichen. Ein '#' steht für genau ein beliebiges Zeichen. Alle anderen Zeichen müssen mit dem Verstellgrößennamen übereinstimmen. Es wird nicht zwischen Groß- und Kleinschreibung unterschieden.
	<code>deviceName</code>	Name des Geräts
Beispiele	<pre>[n,t]=IncaBrowseCalibrationElements ('MAP*', 'Device'); [name,type]= IncaBrowseCalibrationElements ('*');</pre>	

4.5.2 Verstellgröße hinzufügen

Name	<code>IncaAddCalibrationElement</code>	
Beschreibung	Fügt eine Verstellgröße zum aktiven Experiment hinzu. Verstellaktionen können mit Skalaren, Kennlinien und Kennfeldern, einschließlich der zugeordneten Stützstellenverteilungen, durchgeführt werden. In jedem Experiment kann eine beliebige Anzahl von Verstellgrößen definiert werden. Auch Stützstellenverteilungen und Gruppenstützstellenverteilungen werden mit diesem Kommando unterstützt.	
Syntax	<code>IncaAddCalibrationElement(deviceName, calibrationName {, displayMode})</code>	
Rückgabewerte	keine	
Argumente	<code>deviceName</code>	Name des Geräts

<code>calibrationName</code>	Name der Verstellgröße
<code>displayMode</code>	Darstellungsart der Verstellgröße: <ul style="list-style-type: none"> • 2: Verstellgröße wird angezeigt und ständig aktualisiert (Vorgabe) • 1: Verstellgröße wird angezeigt aber nicht aktualisiert • 0: Verstellgröße wird nicht angezeigt

Die Auswahl 1 (nur Anzeige) kann bei hohem Datenaufkommen die Übertragungsleistung erheblich verbessern.

Beispiele

```
IncaAddCalibrationElement('anEtk', 'Scalar');
IncaAddCalibrationElement('anEtk', 'Curve');
IncaAddCalibrationElement('anEtk', 'Map');
```

Info

Bei Verstellgrößen vom Typ 'group axis' wird keine Interpolation der davon abhängigen Kennlinien und Kennfeldern vorgenommen.

4.5.3 Verstellgröße lesen

Name	<code>IncaGetCalibrationValue</code>
Beschreibung	Liest den aktuellen Wert einer Verstellgröße oder der zugehörigen Stützstellenverteilung
Syntax	<code>value = IncaGetCalibrationValue(deviceName, calibrationName {, start, size} {, valueType})</code>
Rückgabewerte	<code>value</code> <p>Aktueller Wert der Verstellgröße; die Übereinstimmung der Datentypen ist zu beachten:</p> <ul style="list-style-type: none"> • Skalare: eine (1,1)-Matrix • Kennlinien: eine (x,1)-Matrix • Kennfelder: eine (x,y)-Matrix • Stützstellenverteilungen: eine (x,1)-Matrix

Argumente	deviceName	Name des Geräts
	calibrationName	Name der Verstellgröße
	start	Startindex. Zulässige Datentypen: <ul style="list-style-type: none"> • Für Kennlinien und Stützstellenverteilungen ist hier der Startindex x anzugeben. $x \geq 1$ • Für Kennfelder ist hier der Startindex $[x, y]$ anzugeben. $x, y \geq 1$
	size	Anzahl der zu lesenden Werte. Zulässige Datentypen: <ul style="list-style-type: none"> • Für Kennlinien und Stützstellenverteilungen ist hier die Anzahl x anzugeben. $x \geq 1$ • Für Kennfelder ist hier die Anzahl $[x, y]$ anzugeben. $x, y \geq 1$
	valueType	Auswahl des Rückgabewertes (als Zeichenkette). Zurückgegeben wird entweder der Wert der Verstellgröße (Vorgabe) oder die x- und y-Stützstellenverteilung. Zulässige Einstellungen: <ul style="list-style-type: none"> • v: Wert • x: x-Stützstelle (Kennlinien und -felder) • y: y-Stützstelle (Kennfelder)

Beispiele

```

aValue = IncaGetCalibrationValue('anEtk', 'Scalar');
aCurve = IncaGetCalibrationValue('anEtk', 'Curve');
aMap = IncaGetCalibrationValue('anEtk', 'Map');
xMap = IncaGetCalibrationValue('anEtk', 'Map', 'x');
yMap = IncaGetCalibrationValue('anEtk', 'Map', 'y');
aCurveRange = IncaGetCalibrationValue('anEtk', 'Curve', 2, 3);
aMapRange = IncaGetCalibrationValue('anEtk', 'Map', [2,3], [3,4]);
xMapRange = IncaGetCalibrationValue('anEtk', 'Map', 2, 3, 'x');

```


4.5.4 Verstellgröße ändern

Name	IncaSetCalibrationValue
Beschreibung	Weist einer Verstellgröße oder der zugehörigen Stützstelle einen Wert zu
Syntax	<pre>IncaSetCalibrationValue(deviceName, cali- brationName, value) IncaSetCalibrationValue(deviceName, cali- brationName, value, valueType) IncaSetCalibrationValue(deviceName, cali- brationName, value, start) IncaSetCalibrationValue(deviceName, cali- brationName, value, start, valueType) result = IncaSetCalibrationValue(devi- ceName, calibrationName, value) result = IncaSetCalibrationValue(devi- ceName, calibrationName, value, valueType) result = IncaSetCalibrationValue(devi- ceName, calibrationName, value, start) result = IncaSetCalibrationValue(devi- ceName, calibrationName, value, start, valueType)</pre>

Rückgabewerte `result`

Ergebnis der Verstellaktion (optional, nur bei Fehlern)

Wenn kein Ergebnisbit gesetzt ist, wurde der Verstellgrößenwert erfolgreich geändert. Dies ist auch der Fall, wenn eines der Bits 5 bis 8 gesetzt ist, die lediglich zusätzliche Informationen bereitstellen.

Ist jedoch eines der Bits 0 bis 4 gesetzt, ist die Verstellaktion fehlgeschlagen.

- Bit 0 gesetzt: keine Verstellung durchgeführt
- Bit 1 gesetzt: untere weiche Grenzen verletzt
- Bit 2 gesetzt: obere weiche Grenzen verletzt
- Bit 3 gesetzt: untere harte Grenzen verletzt
- Bit 4 gesetzt: obere harte Grenzen verletzt
- Bit 5 gesetzt: auf untere weiche Grenzen begrenzt
- Bit 6 gesetzt: auf obere weiche Grenzen begrenzt
- Bit 7 gesetzt: auf untere harte Grenzen begrenzt
- Bit 8 gesetzt: auf obere harte Grenzen begrenzt

Es können verschiedene Ursachen vorliegen, wenn eine Verstellaktion nicht ausgeführt wird. Z.B. kann es sein, dass bei dem aktiven Verstellmodus eine der Grenzen verletzt werden würde. In diesem Fall werden mit den Bits 1 bis 4 genauere Informationen zurückgegeben. Ein anderer Grund könnte darin bestehen, dass die Verstellgröße oder die aktive Speicherseite schreibgeschützt ist oder dass eine x- oder y-Stützenverteilung die Monotonie verletzen würde. In diesen Fällen würde nur Bit 0 gesetzt werden.

Argumente	deviceName	Name des Geräts
	calibrationName	Name der Verstellgröße
	value	Der Wert der Verstellgröße. Zulässige Datentypen: <ul style="list-style-type: none"> • Skalare: eine (1,1)-Matrix • Kennlinien: eine (x,1)-Matrix • Kennfelder: eine (x,y)-Matrix • x- und y-Stützstellenverteilungen: eine (x,1)-Matrix
	start	Startindex. Zulässige Datentypen: <ul style="list-style-type: none"> • Für Kennlinien und Stützstellenverteilungen ist hier der Startindex x anzugeben. $x \geq 1$ • Für Kennfelder ist hier der Startindex $[x, y]$ anzugeben. $x, y \geq 1$
	valueType	Auswahl des Rückgabewertes (als Zeichenkette). Zurückgegeben wird entweder der Wert der Verstellgröße (Vorgabe) oder die x- und y-Stützstellenverteilung. Zulässige Einstellungen: <ul style="list-style-type: none"> • v: Wert (Vorgabe) • x: x-Stützstelle (Kennlinien und -felder) • y: y-Stützstelle (Kennfelder)

Beispiele

```

IncaSetCalibrationValue('anEtk', 'Scalar',
aValue);
IncaSetCalibrationValue('anEtk', 'Curve',
aCurve);
IncaSetCalibrationValue('anEtk', 'Map',
aMap);
IncaSetCalibrationValue('anEtk', 'Map',
xMap, 'x');
IncaSetCalibrationValue('anEtk', 'Map',
yMap, 'y');
IncaSetCalibrationValue('anEtk', 'Curve',
aCurveRange, 2);
IncaSetCalibrationValue('anEtk', 'Map',
aMapRange, [2, 3]);
IncaSetCalibrationValue('anEtk', 'Map',
xMapRange, 2, 'x');

```

4.5.5 Datensatz einem Gerät zuordnen (INCA-MIP Erweitert)

Name	<code>IncaSetDatasetInDevice</code>	
Beschreibung	Ordnet in einem geöffneten Experiment einem Gerät einen Datensatz zu	
Syntax	<code>IncaSetDatasetInDevice(device, dataset)</code>	
Rückgabewerte	keine	
Argumente	<code>device</code>	Name des Gerätes
	<code>dataset</code>	Datenbankpfad des Datensatzes
Beispiele	<code>IncaSetDatasetInDevice('ETK:1', 'Ds4711\Ds4711_3')</code>	

4.5.6 Datensätze eines Gerätes auflisten (INCA-MIP Erweitert)

Name	<code>IncaGetDatasetsForDevice</code>	
Beschreibung	Listet die Namen der Datensätze eines Gerätes auf	
Syntax	<code>name = IncaGetDatasetsForDevice(device)</code> <code>[name, properties] = IncaGetDatasetsForDevice(device)</code>	
Rückgabewerte	<code>name</code>	Liste von Zeichenketten mit dem vollständigen Pfad aller gefundenen Datensätze
	<code>properties</code>	Liste von Zeichenketten mit den Datensatz-Eigenschaften: Mögliche Werte: <ul style="list-style-type: none"> • " (leere Zeichenkette): Datensatz mit Schreib-Lese-Zugriff • r : Datensatz mit Nur-Lese-Zugriff • m : Master-Datensatz mit Schreib-Lese-Zugriff • mr: Master-Datensatz mit Nur-Lese-Zugriff
Argumente	<code>device</code>	Name des Gerätes
Beispiele	<code>l = IncaGetDatasetsForDevice('ETK:1')</code>	

4.5.7 Verstellmodus einstellen (INCA-MIP Erweitert)

Name	<code>IncaSetCalibrationMode</code>	
Beschreibung	Stellt den globalen Verstellmodus ein, der für alle anschließenden Verstellaktionen gilt, die mit dem Befehl <code>IncaSetCalibrationValue</code> ausgeführt werden. Der eingestellte Modus gilt auch nach dem Schließen und erneuten Öffnen eines Experimentes. Beim Start der Schnittstelle MATLAB® ist der Standardmodus für untere und obere Grenzwerte <code>rejectWeakBoundViolation</code> .	
Syntax	<code>IncaSetCalibrationMode(lowerLimitMode, upperLimitMode)</code>	
Rückgabewerte	keine	
Argumente	<code>lowerLimitMode</code>	Neuer Verstellmodus für untere Grenzen
	<code>upperLimitMode</code>	Neuer Verstellmodus für obere Grenzen: <ul style="list-style-type: none"> • <code>rejectWeakBoundViolation</code>: Weist die gesamte Verstellung zurück, wenn eine weiche Grenze mindestens einmal verletzt werden würde (Vorgabe). • <code>limitToWeakBound</code>: Falls die untere oder obere weiche Grenze verletzt werden würde, setzt es den Verstellwert stattdessen auf die untere oder obere weiche Grenze • <code>rejectHardBoundViolation</code>: Ignoriert die weichen Grenzen. Weist die gesamte Verstellung zurück, wenn eine harte Grenze mindestens einmal verletzt werden würde. • <code>limitToHardBound</code>: Ignoriert die weichen Grenzen. Falls die untere oder obere harte Grenze mindestens einmal verletzt werden würde, wird der Verstellwert stattdessen auf die untere oder obere harte Grenze gesetzt.
Beispiele	<code>IncaSetCalibrationMode('rejectHardBoundViolation', 'limitToHardBound')</code>	

4.5.8 Geräte gruppieren (INCA-MIP Erweitert)

Name	<code>IncaGroupDevices</code>	
Beschreibung	Aktiviert oder deaktiviert Steuergerätegruppierungen	
Syntax	<code>IncaGroupDevices (onOff)</code>	
Rückgabewerte	keine	
Argumente	<code>onOff</code>	<ul style="list-style-type: none"> • 0: Deaktiviert die Gruppierung von Steuergeräten • 1: Aktiviert die Gruppierung von Steuergeräten
Beispiele	<code>IncaGroupDevices (1)</code>	

4.5.9 DCM-Datei schreiben (INCA-MIP Erweitert)

Name	<code>IncaWriteToFile</code>	
Beschreibung	Schreibt eine DCM-Datei zum geöffneten Experiment	
Syntax	<code>IncaWriteToFile (format, file, device, calibs {, options})</code>	
Rückgabewerte	keine	
Argumente	<code>format</code>	Identifikation des Dateiformats: <ul style="list-style-type: none"> • 'DCM': DCM-Format
	<code>file</code>	Vollständiger Pfad der zu schreibenden Datei
	<code>device</code>	Gerät dessen Verstellgrößen aufgelistet werden sollen
	<code>calibs</code>	Array der aufzulistenden Verstellgrößen
	<code>options</code>	Optionen für das Herausschreiben im spezifizierten Format
Beispiele	<pre>calibs = {'A0_KW', 'BRABEVI_KL', 'KFZW_GKF'}; IncaWriteToFile('DCM', 'C:\DCMOut1.dcm', 'device1', calibs); IncaWriteToFile('DCM', 'C:\DCMOut2.dcm', 'ETK:1', 'A0_KW');</pre>	

4.6 Speicherseitenverwaltung

Alle bisher beschriebenen API-Funktionen wirken sich immer auf die jeweils aktive Seite eines Geräts aus. Der Verstellzugriff ist prinzipiell nur auf der

Arbeitsseite möglich. Es kann jedoch vorkommen, dass der Schreibzugriff auf die Arbeitsseite des ETKs blockiert wird, weil die Prüfsummen der Arbeitsseiten in der INCA-Datenbank und im ETK nicht übereinstimmen.

Für die Speicherseitenverwaltung sind die nachfolgend beschriebenen API-Funktionen verfügbar.

4.6.1 Speicherseite aktivieren

Name	IncaSwitchPage	
Beschreibung	Setzt die angegebene Speicherseite aktiv	
Syntax	IncaSwitchPage(deviceName, pageName)	
Rückgabewerte	keine	
Argumente	deviceName	Name des Geräts
	pageName	Name der Seite: <ul style="list-style-type: none"> • wp: Arbeitsseite • rp: Referenzseite
Beispiele	<pre>IncaSwitchPage('MyDevice', 'wp'); IncaSwitchPage('Dev', 'rp');</pre>	

4.6.2 Aktive Speicherseite lesen (INCA-MIP Erweitert)

Name	IncaGetCurrentPage	
Beschreibung	Gibt die gerade aktive Speicherseite zurück	
Syntax	pageName = IncaGetCurrentPage(deviceName)	
Rückgabewerte	pageName	Name der aktiven Seite: <ul style="list-style-type: none"> • wp: Arbeitsseite • rp: Referenzseite
Argumente	deviceName	Name des Geräts
Beispiele	<pre>p = IncaGetCurrentPage('MyDevice');</pre>	

4.6.3 Schreibschutz prüfen

Name	IncaIsPageWriteProtected	
Beschreibung	Prüft, ob für die angegebene Speicherseite Schreibschutz besteht	
Syntax	isRW = IncaIsPageWriteProtected(deviceName, pageName)	
Rückgabewerte	isRW	<ul style="list-style-type: none"> • 0: Seite kann beschrieben werden • not 0: Seite ist schreibgeschützt

Argumente	deviceName	Name des Geräts
	pageName	Name der Seite: <ul style="list-style-type: none"> • wp: Arbeitsseite • rp: Referenzseite
Beispiele	<pre>isETK1RW = IncaIsPageWriteProtected('ETK:1', 'wp'); isETK2RW = IncaIsPageWriteProtected('ETK:2', 'rp');</pre>	

4.6.4 Speicherseite hinunterladen

Name	IncaDownloadPage	
Beschreibung	Lädt die angegebene Speicherseite ins Steuergerät	
Syntax	IncaDownloadPage(deviceName, pageName)	
Rückgabewerte	keine	
Argumente	deviceName	Name des Geräts
	pageName	Name der herunterzuladenden Seite: <ul style="list-style-type: none"> • wp: Arbeitsseite • rp: Referenzseite
Beispiele	<pre>IncaDownloadPage('ETK:1', 'wp'); IncaDownloadPage('ETK:1', 'rp');</pre>	

4.6.5 Speicherseite kopieren

Name	IncaCopyPageFromTo	
Beschreibung	Kopiert die angegebene Speicherseite. Zur Zeit ist das Kopieren nur von der Referenz- auf die Arbeitsseite möglich, andere Kombinationen aus Quelle und Ziel werden nicht unterstützt.	
Syntax	IncaCopyPageFromTo(deviceName, sourcePageName, destinationPageName)	
Rückgabewerte	keine	
Argumente	deviceName	Name des Geräts

<code>sourcePageName</code>	Name der zu kopierenden Seite: <ul style="list-style-type: none"> • <code>wp</code>: Arbeitsseite • <code>rp</code>: Referenzseite
<code>destinationPageName</code>	Name der Seite, auf die kopiert wird: <ul style="list-style-type: none"> • <code>wp</code>: Arbeitsseite • <code>rp</code>: Referenzseite

Beispiele `IncaCopyPageFromTo('ETK:1', 'rp', 'wp');`

4.6.6 Unterschiede hinunterladen

Name	<code>IncaDownloadDifferences</code>
Beschreibung	Lädt die Unterschiede zwischen Arbeits- und Referenzseite ins Steuergerät. Wie beim entsprechenden Menübefehl, erfolgt auch hier ein Abgleich nur dann, wenn die Arbeits- und Referenzseite im Zielgerät mit der Referenzseite in INCA übereinstimmen.
Syntax	<code>IncaDownloadDifferences(deviceName)</code>
Rückgabewerte	keine
Argumente	<code>deviceName</code> Name des Geräts
Beispiele	<code>IncaDownloadDifferences('ETK:1');</code>

4.6.7 Speicherseiten hochladen (INCA-MIP Erweitert)

Name	<code>IncaUploadPages</code>
Beschreibung	Lädt die Referenz- und Arbeitsseite in neu erzeugte Datensätze hoch. Die neuen Datensätze werden automatisch dem Gerät zugeordnet.
Syntax	<code>IncaUploadPages(device{,referencePage, workingPage})</code>
Rückgabewerte	keine
Argumente	<code>device</code> Name des Gerätes

referencePage	Datensatzname für die hochgeladene Referenzseite. Falls kein Name spezifiziert wird, verwendet INCA einen Standardnamen.
workingPage	Datensatzname für die hochgeladene Arbeitsseite. Falls kein Name spezifiziert wird, verwendet INCA einen Standardnamen.

Beispiele

```
IncaUploadPages('ETK:1');
IncaUploadPages('ETK:1', 'ref_1', 'work_1');
```

4.7 Anwendungsbeispiele

Beispiel 1

```
% Check if working page is write-protected and
% download the page if it is write-protected
if(IncaIsPageWriteProtected('anEtk', 'wp'))
    IncaDownloadPage('anEtk', 'wp');
end
% Switch to the working page
IncaSwitchPage('anEtk', 'wp');
```

Beispiel 2

Im folgenden Beispiel werden die oben beschriebenen Funktionen zum Auslesen von Messwerten aus dem Gerät `MyDevice` und Messraster 10ms verwendet. Wenn Sie dieses Beispiel ausführen wollen, müssen Sie zunächst in INCA ein Experiment öffnen, in dem ein Gerät namens `MyDevice` zugeordnet ist.

```
% Measure the following signals
IncaAddMeasureElement('MyDevice', '10ms', 'Chan1');
IncaAddMeasureElement('MyDevice', '10ms', 'Chan2');
IncaAddMeasureElement('MyDevice', '10ms', 'Chan3');
IncaAddMeasureElement('MyDevice', '10ms', 'Chan4');

% Now measure
data = [];
time = [];
IncaShowMessages(0);
IncaSetMeasureReadMode(0);
IncaStartMeasurement;
```

```
deltaT = 0;
% Measure for 20 seconds
while( deltaT < 20 )
    % Pause for 0.1 seconds to have more than
one
    % record -- saves processor time.
    pause(0.1)
    % Get up to 500 records for group 10ms
    [ t, d ]=IncaGetRecords( 'MyDevice', '10ms',
500 );
    % Append t and d to time and data
    data = [data; d];
    time = [time; t];
    if( length(time) )
        % Calculate time measured
        deltaT = time( length(time)) - time(1);
    end
end
IncaStopMeasurement;
IncaShowMessages(1);
% Plot the results
plot(time, data);
```

In diesem Beispiel wird nur ein einziges Messraster verwendet. Sie können jedoch mehrere Messraster verwenden und die Daten für die einzelnen Raster unabhängig voneinander von MATLAB[®] aus abrufen.

5 Erstellung und Verteilung von ausführbaren Dateien mit dem MATLAB® Compiler

Mit INCA-MIP können Sie m-Dateien mit INCA-MIP-API-Funktionen, einschließlich MATLAB®-Funktionen, erstellen und kompilieren. Die daraus resultierenden ausführbaren Dateien können auch in Umgebungen ohne MATLAB®-Installation ausgeführt werden.

Für die Erstellung von ausführbaren Dateien wird eine MATLAB®-Installation benötigt. Die dabei entstehende ausführbare Datei kann auf dem Zielsystem genutzt werden, ohne dass eine MATLAB®-Installation oder Kopien zusätzlicher - und ETAS-DLLs vorhanden sein müssen. MATLAB®

Darüber hinaus können MATLAB®-Laufzeitbibliotheken für die Ausführung kompilierter ausführbarer Dateien auf dem Zielsystem erforderlich sein.

Informationen zur Installation und Verwendung der MATLAB-Laufzeitbibliotheken finden Sie in Ihrer MATLAB-Benutzerdokumentation unter Verteilen von Stand-Alone-Anwendungen.

5.1 m-Dateien kompilieren

m-Dateien mit dem MATLAB®-Compiler kompilieren:

1. Kopieren Sie alle `Inca*.dll` und `Inca.*.mexw64` Dateien in das aktuelle Arbeitsverzeichnis.

Stellen Sie sicher, dass der INCA-MIP-Compiler und die MATLAB®-Pfade zur Windows-Environmental-Variable hinzugefügt werden.

Wie Sie die Environmental-Variable einstellen, finden Sie in der Windows-Benutzerdokumentation zu den erweiterten Systemeinstellungen.

Führen Sie den folgenden Befehl aus:

```
mcc -m <m-file-script> -a incaRci2Matlab.dll
```

Beispiel:

Mit dem folgenden Befehl wird aus der Datei `testCase2.m` eine eigenständige ausführbare Datei erzeugt:

```
mcc -m testCase2 -a incaRci2x64Matlab.dll
```

Ergebnis ist die Datei `testCase2.exe`.

Der MATLAB® Compiler erzeugt einen Container mit allen MEX Funktions-DLLs und abhängigen DLLs, die zur Ausführung des kompilierten Skripts benötigt werden. MATLAB® Alle `Inca*.dll`-Dateien, die vom Skript verwendet werden, sowie die `incaRci2x64Matlab.dll` müssen in diesem Container enthalten sein. INCA MEX Functions DLLs haben die Erweiterung `*.mexw64`.

Wenn das kompilierte Skript ausgeführt wird, müssen die DLLs nicht auf dem System vorhanden sein.

Die benötigten Versionen und die entsprechenden Einstellungen des MATLAB®-Compilers finden Sie in Ihrer -Benutzerdokumentation unter dem Stichwort *MATLAB®-Compiler* oder *mcc.MATLAB®*

Info

INCA kann zur selben Zeit nur von einer MATLAB®-Session aus gesteuert werden. Der Versuch, INCA gleichzeitig von mehreren MATLAB®-Instanzen oder ausführbaren Dateien aus zu steuern, wird mit einer Fehlermeldung abgebrochen.

Info

Ab MATLAB® R2016a haben die INCA MEX Funktions-DLLs die Endung
* .mexw64.

5.2 Ausführbare Dateien verteilen

Zum Ausführen von eigenständigen ausführbaren Dateien, die mit dem MATLAB® Compiler kompiliert wurden, ist nur die ausführbare Datei selbst erforderlich. Eine Installation von MATLAB® oder Kopien von MATLAB® Bibliotheken sind nicht erforderlich.

Ausführbare Dateien verteilen, die mit dem MATLAB® R13 Compiler übersetzt wurden

- Kopieren Sie die ausführbaren Dateien auf das Zielsystem.

Anschließend können Sie sie einfach ausführen; weitere Schritte sind nicht erforderlich.

6 **Kontaktinformationen**

ETAS Headquarters

ETAS GmbH

Borsigstraße 24

70469 Stuttgart

Deutschland

Telefon: +49 711 3423-0

Fax: +49 711 3423-2106

WWW: www.etas.com

ETAS Regionalgesellschaften und Technischer Support

Informationen zu Ihrem lokalen Vertrieb und zu Ihrem lokalen Technischen Support bzw. den Produkt-Hotlines finden Sie im Internet:

ETAS Regionalgesellschaften WWW: www.etas.com/de/contact.php

ETAS technical support WWW: www.etas.com/de/hotlines.php

Index

B	
Beispieldateien	21
C	
contact information	70
E	
eigenständige ausführbare Dateien	68
ETAS	
contact information	70
G	
Gerät	10
I	
IncaAddCalibrationElement	54
IncaAddMeasureElement	36
IncaBrowseCalibrationElements	53
IncaBrowseItemsInFolder	30
IncaBrowseMeasureElements	35
IncaClose	28
IncaCopyPageFromTo	64
IncaDatabaseImport	29
IncaDownloadDifferences	65
IncaDownloadPage	64
IncaExecuteManualTrigger	52
IncaGetCalibrationValue	55
IncaGetCurrentPage	63
IncaGetDatasetsForDevice	60
IncaGetDeviceProperties	33
IncaGetDevices	33
IncaGetHardwareStatus	49
IncaGetInstalledAddOnInfo	26
IncaGetInstalledProductInfo	25
IncaGetMeasureRatesForDevice	35
IncaGetProperties	27
IncaGetRecordingMode	42
IncaGetRecordingProperties	38
IncaGetRecordingState	52
IncaGetRecords	46
IncaGetRecordStruct	52
IncaGetVersion	27
IncaGroupDevices	62
IncaLicenseValid	25
IncaPageWriteProtected	63
IncaOpen	28
IncaOpenDatabase	29
IncaOpenExperiment	32
IncaResetExperiment	33
IncaResetRecords	49
IncaSetCalibrationMode	61
IncaSetCalibrationValue	57
IncaSetDatasetInDevice	60
IncaSetMeasureReadMode	45
IncaSetProjectAndDatasetInDevice	31
IncaSetRecordingMode	43
IncaSetRecordingProperties	40
IncaSetTrigger	50
IncaShowMessages	25
IncaStartMeasurement	37
IncaStartRecording	44
IncaStopMeasurement	38
IncaStopRecording	45
IncaSwitchPage	63
IncaUploadPages	65
IncaWriteToFile	62
L	
Lizenzierung	15
M	
M-Dateien	9,21
MATLAB-Skripte	9
mcc	68
Messdaten	10
Messdatensatz	10
Messen	9
Messraster	10
MEX-Dateien	12
P	
Privacy Notice	8
R	
Ringpuffer	10
S	
Signal	11
Signalgruppe	11
Speicherseitenverwaltung	9

V

Verstellen	9
Verstellgröße	10