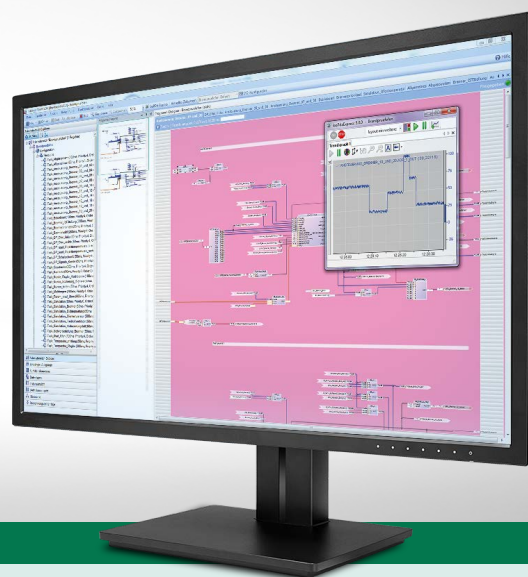




# Messtechnik und Automatisierung in einem System

ibaLogic



**ibaLogic**

Signalverarbeitung und Automatisierung

## Einsatzgebiete

- › Signalmanager in der Messtechnik
- › Komplexe Signalvorverarbeitung (z. B. Schwingungsüberwachung und Messsysteme)
- › Systemkopplungen bei der Modernisierung von Anlagen
- › Automatisierung von Anlagen
- › Prozesssimulation
- › Energiemanagement und Gebäudetechnik
- › Realisierung von Prozessmodellen zur Optimierung

# Prozesse mit nur einem System steuern und gleichzeitig messtechnisch erfassen



Vor allem in der Mess- und Regelungstechnik mit ihren schnellen und dynamischen Prozessen wird ibaLogic als System zur Signalverarbeitung, Steuerung und Simulation sowie als Kommunikationsgateway eingesetzt.

## Auf einen Blick

- › PC-basiertes System zur Signalverarbeitung, Steuerung und Simulation
- › Grafische Programmierung nach IEC 61131-3
- › Client-/Server Prinzip, PAC-Architektur
- › Verschiedene Laufzeitplattformen
- › Einfache Programmierung und intuitive Bedienung
- › Anwenderspezifische Funktionsbausteine und Datentypen
- › Einbindung vorhandener C(++) Algorithmen
- › Eingebundene Datenaufzeichnung
- › Integriertes ibaPDA-Express zur Messwertanzeige
- › Ereignis-Tasks
- › Unterbrechung von Tasks möglich (je nach Priorität)

## Ein System für viele Anwendungen

ibaLogic ist ein System zur Signalverarbeitung und Automatisierung. Die fünfte Generation des Systems ibaLogic, das bereits in vielen industriellen Anwendungen zuverlässig arbeitet, wurde programmtechnisch neu konzipiert und mit einer zeitgemäßen Oberfläche versehen.

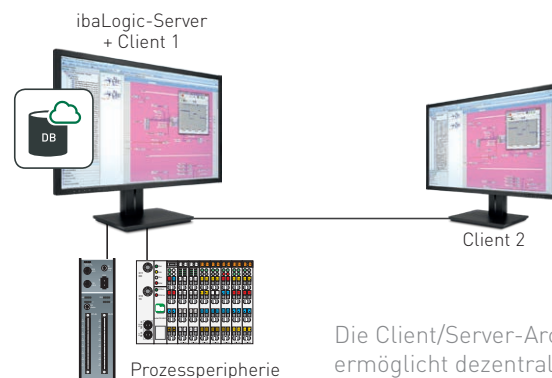
Basierend auf der Architektur eines Programmable Automation Controllers (PAC) nutzt ibaLogic sowohl leistungsstarke PCs als auch spezielle Laufzeitsysteme, wie z. B. ibaPADU-S-IT-2x16, um Aufgaben einer klassischen SPS zu lösen. ibaLogic wird vor allem in der Mess- und Regelungstechnik für schnelle und dynamische

Prozesse eingesetzt. Kurze Programmzykluszeiten bis zu 1 ms und ein deterministisches Zeitverhalten machen dies möglich.

## Das Programmkonzept

Um den stetig wachsenden Ansprüchen an die Automatisierung und Messtechnik gerecht zu werden, wurde ein Programmkonzept entwickelt, das auch für zukünftige Entwicklungen die erforderliche Flexibilität gewährleistet und sich u. a. durch folgende Eigenschaften auszeichnet:

- › Client/Server-Architektur
- › Datenbankbasiert (Microsoft SQL Server)
- › Programmierer ist mit dem Laufzeitsystem über TCP/IP-Kommunikation verbunden



Die Client/Server-Architektur ermöglicht dezentrales Arbeiten.



Programm-Designer und Navigationsbereich in ibaLogic

Diese Eigenschaften ermöglichen, dass die Anwendung auf einem autarken System (PC oder ibaPADU-S-IT) abläuft, und über einen Entwicklungs-PC oder Notebook programmiert und beobachtet werden kann. Die Projektdaten (z. B. Programme und Konfigurationen) werden zentral in einer Datenbank abgelegt.

### PAC-Architektur

ibaLogic wurde nach der Programmable Automation Controller (PAC)-Architektur programmiert. Ein Kennzeichen der PAC-Architektur ist die Verbindung von PC-typischen Eigenschaften und IT-Standards mit den Automatisierungsfunktionen der SPS-Technik.

So lassen sich Funktionen der Prozess- und Anlagensteuerung mit der Verwendung von Standards wie z. B. Windows XP/CE, HTML, SMTP, FTP und SQL sowie Ethernet, TCP/IP, USB, CAN-Bus, usw. kombinieren.

Im Falle von ibaLogic wurde die PAC-Architektur außerdem um die Funktionen der Messung und der Datenaufzeichnung erweitert. iba AG bezeichnet

deshalb das System als „Programmable Measurement and Automation Controller“ (PMAC).

### Einfache Programmerstellung und intuitive Bedienung

Die einfache grafische Programmierung und ein benutzerfreundliches Bedienkonzept machen es Neueinsteigern und auch erfahrenen Programmierern leicht, sich schnell in ibaLogic einzuarbeiten. Die Steuerung, Konfiguration und Bedienung der Programmelemente ist durch die übersichtliche Anordnung der Fenster intuitiv und einfach vorzunehmen. Alle wichtigen Funktionen sind auf einen Blick sichtbar oder können über Kontextmenüs erreicht werden. Einstellungen und Konfigurationen werden über Dialoge vorgenommen.

Die Bedienoberfläche ist in zwei Hauptbereiche gegliedert.

### Programm-Designer

Der Programm-Designer ist das Hauptfenster der Programmieroberfläche. In diesem wird die eigentliche „Programmierarbeit“ vorgenommen, d. h. hier werden Funktionsbausteine platziert und miteinander verknüpft. Anhand

der Hintergrundfarbe lässt sich sofort erkennen, ob das Programm im Editiermodus arbeitet oder „scharf geschaltet“ ist. Bei laufender Berechnung werden die Istwerte aller Baustein-konnektoren alphanumerisch angezeigt. Die Zustände False/True bei binären Signalen werden mit den Farben Blau/Rot der Verbindungslinien angezeigt.

### Navigationsbereich

Im Navigationsbereich befinden sich alle für die Konfiguration und Programmierung erforderlichen Elemente und Informationen. Programmelemente können aus der entsprechenden Rubrik entweder per Drag & Drop in den Programm-Designer gezogen oder über Kontextmenüs konfiguriert werden.

Dazu gehören:

- › Projekt- und Programmstruktur
- › Ein- und Ausgangsressourcen
- › Funktionsbausteinbibliothek
- › Datentypen

Darüber hinaus bietet der Navigationsbereich weitere Ansichten mit den verwendeten Programmelementen bezüg-

lich ihrer Instanzen, Hierarchie und Berechnungsreihenfolge.

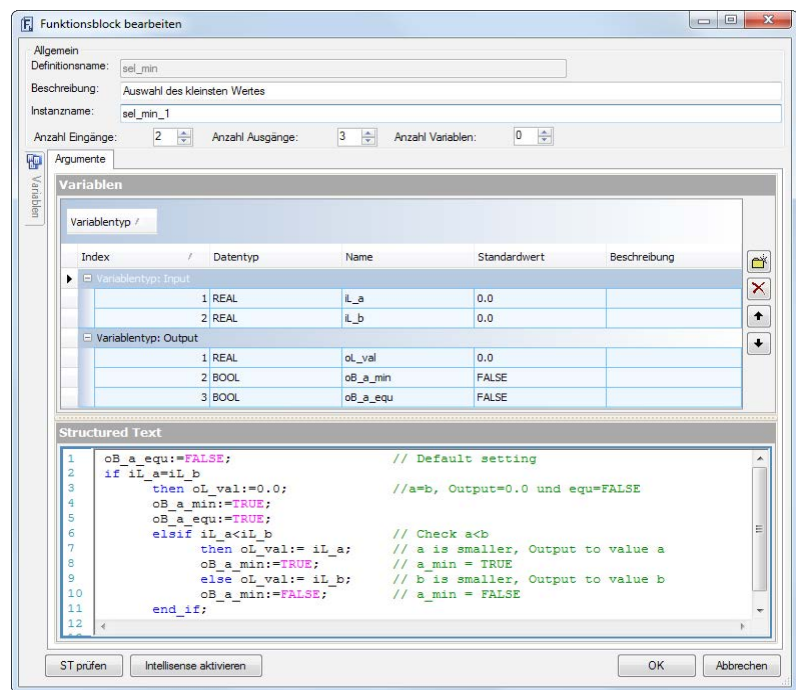
Diese Ansichten sind bei der Fehlerbehebung (Debugging) eines Programms hilfreich. Beispielsweise können Bausteine in der Hierarchie-Ansicht einfach per Doppelklick gefunden und angezeigt werden. Sowohl Projektierung als auch Fehlersuche können damit sehr effizient erfolgen.

### Erweiterbare Funktionsbaustein-Bibliothek

Die Standardbibliothek verfügt über alle in der IEC-Norm vorgesehenen Standard-Funktionsbausteine (FBs) sowie über hilfreiche Ergänzungen (Mittelwertbilder, PID-Regler, FFT-Bausteine, usw.). Selbsterstellte Funktionsbausteine und/oder Makros können mit einem Mausklick im FB-Eigenschaften-Menü der Bibliothek hinzugefügt werden. Zusätzlich können Funktionsbausteine und Programme in andere ibaLogic-Anwendungen ex- und importiert werden. Für die Programmierung werden die Bausteine einfach per Drag & Drop aus dem Navigationsbereich in den Programm-Designer platziert.

### Benutzerdefinierte Funktionsbausteine

Die Erzeugung eigener Funktionsbausteine mit der textuellen Eingabemethode auf Basis von IEEE Strukturierter Text (ST) ist integraler Bestandteil von ibaLogic. Dazu stellt ibaLogic einen komfortablen Funktionsblock-Editor zur Verfügung. Im Funktionsblock-Editor können Eingänge, Ausgänge und interne Variablen sowie die Verarbeitungslogik des Bausteins definiert werden. Nachdem ein Funktionsblock erzeugt wurde, steht dieser in der Bausteinbibliothek zur Verfügung.



Komfortable Bearbeitung der Funktionsbausteine im Funktionsblock-Editor

### Integration existierender C(++) Algorithmen

Die Nutzung vorhandener Hochsprachenprogramme ist eine wertvolle Eigenschaft von ibaLogic. Technologisches Know-how kann so vor ungewolltem Zugriff geschützt werden und die Kontinuität bewährter technischer Lösungen ist auch bei Modernisierungen gewährleistet. ibaLogic stellt dafür einen Programmierrahmen zur Verfügung, in den vorhandener Code eingebaut werden kann. Die so generierten Bausteine haben die gleichen Eigenschaften wie Funktionsbausteine und werden in der grafischen Oberfläche mit allen Konnektoren und den aktuellen Signalwerten angezeigt.

Mit dem Entwicklungs-Kit ibaLogic-V5-DLL-SDK\* können Anwender eigene Funktionen erstellen und anwenderspezifische Verarbeitungen und Berechnungen im ibaLogic-Laufzeitsystem vornehmen. Die Funktionen werden von ibaLogic in Echtzeit ausgeführt.

Die ibaLogic-V5-DLL- Runtime\*-Lizenz ist erforderlich, um anwenderspezifische DLLs freizuschalten.

### Hierarchisches Design und Makrotechnik

Die Übersichtlichkeit des Gesamtprogramms lässt sich erhöhen, indem Programmteile, die mit diskreten Funktionsbausteinen erstellt wurden, zu übergeordneten Funktionsbausteinen, den sogenannten Makrobausteinen, zusammengefasst werden. Darüber hinaus besteht die Möglichkeit, diese Makrobausteine in anderen ibaLogic-Projekten zu verwenden. Makrobausteine können von Grund auf programmiert oder mittels Mehrfachauswahl aus vorhandenen Funktionsbausteinen gebildet werden. Die Anzahl der Hierarchien ist nicht limitiert.

### Benutzerdefinierte Datentypen

Dem Anwender stehen neben den Standard-Datentypen gemäß IEC-Norm auch verschiedene benutzerdefinierte Datentypen zur Verfügung. Diese Datentypen können mithilfe einfacher Dialoge frei definiert bzw. aus Standard- oder anderen Datentypen abgeleitet werden. Zu den typischen benutzerdefinierten Datentypen gehören insbesondere Aufzählungen (Enum), Strukturen (Strukt) und Felder (Array). Die Datentypen sind im Navigationsbereich dargestellt und können sofort im Programm verwendet werden.

### Programmänderungen während des Betriebes

Änderungen an einem Programm können während des Betriebes vorgenommen werden, da keine separaten Compilierungsläufe und Ladevorgänge erforderlich sind.

Alle Änderungen können sofort automatisch übernommen werden. Die Programme, die mit einem ibaLogic-Client erstellt und gestartet wurden, laufen in kompilierter Form als sogenannte Laufzeit auf dem Zielsystem (z. B. einem ibaLogic-Rechner). Diese Laufzeit kann immer dann automatisch und stoßfrei aktualisiert (neu kompiliert) werden, wenn im ibaLogic-System Änderungen vorgenommen werden.

### Verschiedene Laufzeitumgebungen (Plattformen)

Die Laufzeit kann unabhängig vom ibaLogic-Programmiersystem (Server) arbeiten, sofern eine geeignete Umgebung vorhanden ist. ibaLogic-Server und -Client werden nur während der Programmerstellung oder -änderung benötigt.

Für ibaLogic gibt es zurzeit zwei Plattformen:

- ▶ Windows-PC (keine Echtzeiterweiterung nötig)
- ▶ ibaPADU-S-IT2x16

### Prozess-Schnittstellen

ibaLogic bietet das gesamte Spektrum der iba-Connectivity, um Ein- und Ausgabesignale zu verarbeiten.

Über die ibaNet-Lichtwellenleiter können alle Geräte aus der ibaPADU-Familie, Buskopppler und Systemanschlüsse angeschlossen werden. Zusätzlich stehen Kopplungen über Feldbus, Ethernet und Sonderprotokolle zur Verfügung.

Aufgrund dieser Eigenschaften eignet sich ibaLogic besonders für die Einbindung in vorhandene, heterogene Automatisierungsstrukturen für die Modernisierung.

### Performance und Systemdynamik

Die kleinste Zykluszeit von ibaLogic beträgt 1 ms. Bei einem Einsatz der entsprechenden iba-Peripheriebaugruppen ist eine schnelle Datenerfassung bis 40 kHz möglich, sodass z. B. Vibrations- und Schwingungsanalysen problemlos möglich sind. Die Anzahl möglicher Tasks ist nicht begrenzt. Jedem Task kann eine individuelle Zykluszeit und eine Priorität zugeordnet werden.

### Diagnose der Task-Berechnungen

Mit dem Zusatz-Tool ibaLogic Timing Diagnostics kann das Zeitverhalten des Systems durch Darstellung der aktuellen Task-Zustände µs-genau als Messdatei (\*.dat) aufgezeichnet und in ibaAnalyzer analysiert werden.

### Messwertanzeige und Datenaufzeichnung in ibaLogic

Die Messdatenaufzeichnung ist integraler Bestandteil von ibaLogic. Mithilfe eines speziellen Funktionsbausteins können Messdateien (\*.dat) wie mit einem ibaPDA-System erzeugt und anschließend mit ibaAnalyzer ausgewertet werden.

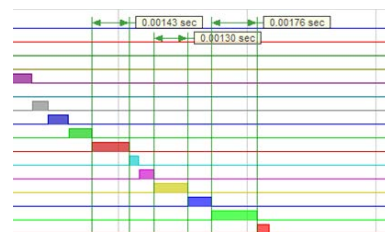
Für Test und Inbetriebnahme steht das integrierte ibaPDA-Express zur Verfügung, mit dem Istwerte aus dem Programm live angezeigt werden können.

### ibaLogic als Simulationssystem

ibaLogic ist auch für die Simulation von Prozessen geeignet. Durch die OPC-Konnektivität ist es möglich, Prozessabläufe mit ibaLogic zu simulieren und die Simulationsergebnisse (z. B. Bewegungen von Maschinenelementen) mit der 3D-Visualisierung ibaAnimation-3D anschaulich darzustellen. Selbstverständlich können dafür auch andere Visualisierungssysteme verwendet werden.

### Datenbank-Verbindung

Mit dem ODBC\_ACCESS Baustein kann der Datenaustausch mit Datenbanken erfolgen.



Beispiel für Anzeige der Timing-Diagnose in ibaAnalyzer

# Technische Daten

	Laufzeitsystem mit Projektierungsumgebung	Nur Projektierungsumgebung (Laufzeit auf getrennter Plattform)
<b>Systemvoraussetzungen</b>	Doppel-Pentium 2 GHz Dual-Core >2 GByte RAM 500 MByte freier Festplatten-Speicher Windows 2008 Server (32 Bit), Windows 7 (32 Bit /64 Bit), Windows 2008 Server R2 (64 Bit), Windows 8 (32 Bit/64 Bit), Windows 2012 Server (64 Bit), Windows 8.1 (32 Bit/64 Bit), 2012 Server R2 (64 Bit) und Windows 10 (32 Bit/64 Bit)	Pentium 2 GHz Single-Core 500 MByte freier Festplatten-Speicher Microsoft Windows 7
<b>Allgemeine Daten</b>	Programmierbares Mess- und Automatisierungssystem/ Programmable Measurement and Automation Controller (PMAC) Grafische Programmierung nach IEC1131-3 (IEEE 61131-3) Meta-Sprache Strukturierter Text (ST) Unbegrenzte Anzahl von unterbrechbaren, zyklischen und ereignisgesteuerten Tasks mit individuellen Zyklen und Prioritäten Unbegrenzte Anzahl von Hierarchien in einem Projekt Signalverarbeitung mit Zyklen bis zu 1 ms Export des Applikationsprogramms in lesbare ASCII-Dateien	
<b>Betriebsarten</b>	Steuerungsmodus: Verwendet werden stets die jüngsten Mess-/Eingangswerte Gepufferter Modus: Für schnelle Mess-/Eingangswerte (bis 40 kHz/Kanal)	
<b>Standarddatentypen</b>	BOOL, BITSTRING, INT, DINT, UDINT, REAL, LREAL, ARRAY (4-dim), STRING, TIME	
<b>Benutzerdatentypen</b>	Arrays, Enums, Structs und weitere abgeleitete Datentypen	
<b>Funktionsbausteine, Makros und dynamische Bibliotheken (DLL)</b>	IEEE-Funktionsblöcke (vollständiger Satz) Erstellung anwenderspezifischer Funktionsblöcke mit Strukturierter Text (ST) Erstellung und Erweiterung von Makroblöcken mit hierarchischer Programmstruktur Applikationsspezifische Bibliotheken (auf Anfrage bei iba AG)	
<b>I/O-System</b>	Komplettes iba I/O-Spektrum (ibaNet-Komponenten für diskrete Signale, z. B. ibaPADU, ibaNet750) Feldbus-Anbindungen (z. B. Profibus Master/Slave) Ethernet-Anbindungen (z. B. ibaPDA, EtherNet/IP) OPC (z. B. für die HMI-Kommunikation) Sonderprotokolle (z. B. Reflective Memory) Bidirektionale Verbindungen zu den meisten SPS- und Regelsystemen (auf Anfrage bei iba AG)	
<b>Diagnose</b>	Istwertanzeige aller Signale und Variablen an den Bausteinkonnektoren Bei binären Signalen: Zustandsabhängige Farbgebung der Verbindungslinien ibaPDA-Express integriert, zur Messwertanzeige beliebiger Signale und Variablen Zusatztool ibaLogic Timing Diagnose zur Analyse des Task-Verhaltens mit ibaAnalyzer	

# Lizenzkostenfreie Einstiegsversion: ibaLogic Lite

- › Mit der kostenlosen Einstiegsversion realisieren Sie einfache Automatisierungslösungen.
- › Download auf [www.iba.ag.com](http://www.iba.ag.com)

ibaLogic im Vergleich	ibaLogic-V5-Lite	ibaLogic-V5	Embedded ibaPADU-S-IT <sup>1)</sup>
Unbegrenzte Anzahl IO	✓	✓	✓
Zykluszeit	≥ 50 ms	≥ 1 ms	≥ 1 ms
ibaPDA Interface <sup>2)</sup>	✓	✓	✓
Unterstützte Hardware ibaNet AVR-NET-IO	✓ ✓	✓ ✓	✓ ✓
Playback von Messdateien	✓	✓	✓
TCP/IP	4 Verbindungen	✓	✓
Datenspeicherung im DatFileFormat DFW (Signale)	-	64 <sup>3)</sup> /256 <sup>3)</sup> /1024 <sup>3)</sup>	256
OPC DA Server oder OPC UA Server	-	unbegrenzt	128
Andere Schnittstellen (DLL/RFM/SST/Profibus/ ... )	-	Schnittstellen auf Anfrage <sup>3)</sup>	
Bestellnr.	32.500002	32.500001	in Hardware enthalten

1) im Hardware-Preis enthalten 2) Hierfür ist ein kostenpflichtiges Interface auf der PDA-Seite erforderlich 3) kostenpflichtiges Add-on

## Bestellinformationen ibaLogic Add-ons

Bestellnr.	Bezeichnung	Beschreibung
32.500005	ibaLogic-V5 Timing Diagnostics	Extended Diagnostics
32.500009	ibaLogic-V5 upgrade with 64-DatFileWrite	Datenaufzeichnung mit DatFileWrite-Baustein, 64 Signale pro Laufzeitsystem
32.500010	ibaLogic-V5 upgrade 64 to 256-DatFileWrite	Datenaufzeichnung mit DatFileWrite-Baustein, Lizenzenerweiterung auf 256 Signale pro Laufzeitsystem
32.500011	ibaLogic-V5 upgrade 256 to 1024-DatFileWrite	Datenaufzeichnung mit DatFileWrite-Baustein, Lizenzenerweiterung auf 1024 Signale pro Laufzeitsystem
32.500020	ibaLogic-V5-Interface-RFM-Access	Schnittstelle für den Datenaustausch über Reflective Memory
32.500021	ibaLogic-V5-Interface-Profibus-DP-Master	Schnittstelle für Profibus DP-Masteranschlutung
32.500030	ibaLogic-V5-DB ACCESS	Funktionsbaustein für Datenbankzugriff
32.500035	ibaLogic-V5-DLL-SDK	OEM Vereinbarung erforderlich, inkl. 3 Tage Schulung bei iba oder per Teamviewer
32.500036	ibaLogic-V5-DLL-Runtime	DLL Runtime Freigabelizenz



## iba AG

### Hausanschrift

Königswarterstr. 44  
90762 Fürth

Telefon: +49 (911) 97282-0

Telefax: +49 (911) 97282-33

### Postanschrift

Postfach 1828  
90708 Fürth

[www.iba-ag.com](http://www.iba-ag.com)

[iba@iba-ag.com](mailto:iba@iba-ag.com)

Durch Tochterunternehmen und Vertriebspartner ist die iba AG weltweit vertreten. Technische Änderungen und Irrtümer vorbehalten.