BECKHOFF New Automation Technology

Documentation | EN

IPxxxx-B520

Fieldbus Box modules for DeviceNet

2021-06-21 | Version: 1.6

Table of contents

1 Foreword

1.1 Notes on the documentation

Intended audience

This description is only intended for the use of trained specialists in control and automation engineering who are familiar with the applicable national standards.
It is essential that the documentation and the following notes and explanations are followed when installing and commissioning these components.
It is the duty of the technical personnel to use the documentation published at the respective time of each installation and commissioning.

The responsible staff must ensure that the application or use of the products described satisfy all the requirements for safety, including all the relevant laws, regulations, guidelines and standards.

Disclaimer

The documentation has been prepared with care. The products described are, however, constantly under development.

We reserve the right to revise and change the documentation at any time and without prior announcement.

No claims for the modification of products that have already been supplied may be made on the basis of the data, diagrams and descriptions in this documentation.

Trademarks

Beckhoff®, TwinCAT®, TwinCAT/BSD®, TC/BSD®, EtherCAT®, EtherCAT G®, EtherCAT G10®, EtherCAT P®, Safety over EtherCAT®, TwinSAFE®, XFC®, XTS® and XPlanar® are registered trademarks of and licensed by Beckhoff Automation GmbH. Other designations used in this publication may be trademarks whose use by third parties for their own purposes could violate the rights of the owners.

Patent Pending

The EtherCAT Technology is covered, including but not limited to the following patent applications and patents: EP1590927, EP1789857, EP1456722, EP2137893, DE102015105702 with corresponding applications or registrations in various other countries.

Copyright

1.2 Safety instructions

Safety regulations

Please note the following safety instructions and explanations!
Product-specific safety instructions can be found on following pages or in the areas mounting, wiring, commissioning etc.

Exclusion of liability

All the components are supplied in particular hardware and software configurations appropriate for the application. Modifications to hardware or software configurations other than those described in the documentation are not permitted, and nullify the liability of Beckhoff Automation GmbH & Co. KG.

Personnel qualification

This description is only intended for trained specialists in control, automation and drive engineering who are familiar with the applicable national standards.

Description of instructions

In this documentation the following instructions are used.
These instructions must be read carefully and followed without fail!

⚠ DANGER

Serious risk of injury!

Failure to follow this safety instruction directly endangers the life and health of persons.

⚠ WARNING

Risk of injury!

Failure to follow this safety instruction endangers the life and health of persons.

⚠ CAUTION

Personal injuries!

Failure to follow this safety instruction can lead to injuries to persons.

NOTE

Damage to environment/equipment or data loss

Failure to follow this instruction can lead to environmental damage, equipment damage or data loss.

Tip or pointer

This symbol indicates information that contributes to better understanding.

1.3 Documentation Issue Status

Version	Comment
1.6	• Structure update • Chapter “DeviceNet Connector” added
1.0…1.5	Previous versions

Firmware and hardware versions

This documentation refers to the firmware and hardware version that was applicable at the time the documentation was written.

The module features are continuously improved and developed further. Modules having earlier production statuses cannot have the same properties as modules with the latest status. However, existing properties are retained and are not changed, so that older modules can always be replaced with new ones.

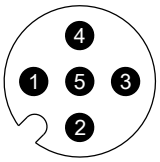
2 Introduction

The purpose of this document is to provide DeviceNet Specific Information needed to run Beckhoff IPxxxx-B520 fieldbus boxes in a DeviceNet Network.

DeviceNet Characteristics

Characteristic	Description
DeviceNet Functionality	Group Two Only Slave for DeviceNet Master / Scanner
IO – Modes	Polling, Bit Strobe, Change of State / Cyclic
Configuration	Switches (Node Address) , Configuration Objects, Electronic Data Sheet (EDS)
LEDs	Module / Network Status LED, Vendor Specific IO LEDs
Electronic Data Sheet	Electronic Data Sheet for each type of IPxxxx-bB520 (www.beckhoff.com)
Connector	Sealed Micro Style Connector
Baud Rates	125 Kbaud, 250 Kbaud, 500 Kbaud

3 DeviceNet Connector

M12 plug	Pin	Symbol
	1	Shield
	2	V+
	3	V-
	4	CANH
	5	CANL

4 IO Data Mapping

The IO Data Mapping describes the contents of the IO Data of the IPxxx in Receive and Transmit direction.

The description is done by DeviceNet Assembly Objects.

Each of the IPxxxx supports one Assembly Object in each data direction. The input data/status or output data is mapped to a Byte Stream exchanged with the DeviceNet Master / Scanner by IO-Data Transfer.

For all possible IO-Modes (Poll, Bit-Strobe, Change of State / Cyclic) the same IO Assembly is used.

4.1 Assembly Objects

The Assembly Object binds attributes of multiple objects, which allows data to or from each object to be sent or received over a single connection (IO or Explicit). Assembly objects are used to bind input data and output data.

Class Code: 4 (04_{hex})

Class Attributes

Attribute ID	Access Rule	Name	Data Type	Description	Semantics of Value
1	Get	Revision	UINT	Revision of implementation	2

Instance Attributes

Attribute ID	Access Rule	Name	Data Type	Description	Semantics of Value
3 (03_{hex})	Get/set	Value	Array of Byte	Input or Output Data of the IPxxxx.	

Common Services

Service Code	Service Name	Description
14 ($0E_{hex}$)	Get_Attribute_Single	Returns the contents of the specified attribute
16 (10_{hex})	Set_Attribute_Single	Modifies an attribute Value

IO Assembly Instances

Number	Type	Name	IPxxxx
24	Input	8-Point Input with multiple Status Bits	IP10xx-B520
44	Input	8-Point Output Status Bits	IP20xx-B520
63	Input	4-Point Input with multiple Status Bits and multiple Output Status Bits	IP23xx-B520
64	Input	8-Point Input with multiple Status Bits and multiple Output Status Bits	IP10xx-B520
150	Input	4-Point Input with multiple Status Bits	IP3xxx-B520
151	Input	4-Point Output Status Bits	IP4xxx-B520
171	Input	Serial Interface	IP60xx-B520
172	Input	Up/Down Counter	IP1502-B520
33	Output	4-Point Output (discrete)	IP23xx-B520
34	Output	8-Point Output (discrete)	IP20xx-B520 IP24xx-B520
160	Output	4-Point Output (analog)	IP4xxx-B520
180	Output	Serial Interface	IP60xx-B520
181	Output	Up/Down Counter	IP1502-B520

4.2 IP10xx-B520

Input Data (2 Byte)

IO Assembly Instances

Number	Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
24	0	Discrete Input8	Discrete Input7	Discrete Input6	Discrete Input5	Discrete Input4	Discrete Input3	Discrete Input2	Discrete Input1
	1	Status Discrete Input8	Status Discrete Input7	Status Discrete Input6	Status Discrete Input5	Status Discrete Input4	Status Discrete Input3	Status Discrete Input2	Status Discrete Input1

No Output Data (0 Bytes)

Mapping IO Assembly Data to DeviceNet Objects

Data Component Name	Class Name	Class Number	Instance	Attribute Name	Attribute Number
Discrete Input n	Discrete Input Point	8	1 – 8	Value	3
Status Discrete Input n	Discrete Input Point	8	1 – 8	Status	4

4.3 IP20xx-B520

Input Data (1 Byte)

IO Assembly Instances

Number	Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
44	0	Status Discrete Output8	Status Discrete Output7	Status Discrete Output6	Status Discrete Output5	Status Discrete Output4	Status Discrete Output3	Status Discrete Output2	Status Discrete Output1

Output Data (1 Byte)

IO Assembly Instances

Number	Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
34	0	Discrete Output8	Discrete Output7	Discrete Output6	Discrete Output5	Discrete Output4	Discrete Output3	Discrete Output2	Discrete Output1

Mapping IO Assembly Data to DeviceNet Objects

Data Component Name	Class Name	Class Number	Instance	Attribute Name	Attribute Number
Discrete Output n	Discrete Output Point	9	1 – 8	Value	3
Status Discrete Output n	Discrete Output Point	9	1 – 8	Status	4

4.4 IP23xx-B520

Input Data (2 Bytes)

IO Assembly Instances

Number	Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
63	0	Status Discrete Input4	Status Discrete Input3	Status Discrete Input2	Status Discrete Input1	Discrete Input4	Discrete Input3	Discrete Input2	Discrete Input1
	1	Reserved	Reserved	Reserved	Reserved	Status Discrete Output4	Status Discrete Output3	Status Discrete Output2	Status Discrete Output1

Output Data (1 Byte)

IO Assembly Instances

Number	Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
33	0	Reserved	Reserved	Reserved	Reserved	Discrete Output4	Discrete Output3	Discrete Output2	Discrete Output1

Mapping IO Assembly Data to DeviceNet Objects

Data Component Name	Class Name	Class Number	Instance	Attribute Name	Attribute Number
Discrete Input n	Discrete Input Point	8	1 – 4	Value	3
Status Discrete Input n	Discrete Input Point	8	1 – 4	Status	4
Discrete Output n	Discrete Output Point	9	1 – 4	Value	3
Status Discrete Output n	Discrete Output Point	9	1 – 4	Status	4

4.5 IP24xx-B520

Input Data (3 Byte)

IO Assembly Instances

Number	Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
64	0	Discrete Input8	Discrete Input7	Discrete Input6	Discrete Input5	Discrete Input4	Discrete Input3	Discrete Input2	Discrete Input1
	1	Status Discrete Input8	Status Discrete Input7	Status Discrete Input6	Status Discrete Input5	Status Discrete Input4	Status Discrete Input3	Status Discrete Input2	Status Discrete Input1
	2	Status Discrete Output8	Status Discrete Output7	Status Discrete Output6	Status Discrete Output5	Status Discrete Output4	Status Discrete Output3	Status Discrete Output2	Status Discrete Output1

Output Data (1 Byte)

IO Assembly Instances

Number	Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
34	0	Discrete Output8	Discrete Output7	Discrete Output6	Discrete Output5	Discrete Output4	Discrete Output3	Discrete Output2	Discrete Output1

Mapping IO Assembly Data to DeviceNet Objects

Data Component Name	Class Name	Class Number	Instance	Attribute Name	Attribute Number
Discrete Input n	Discrete Input Point	8	1 – 8	Value	3
Status Discrete Input n	Discrete Input Point	8	1 – 8	Status	4
Discrete Output n	Discrete Output Point	9	1 – 8	Value	3
Status Discrete Output n	Discrete Output Point	9	1 – 8	Status	4

4.6 IP3xxx-B520

Input Data (9 Bytes)

IO Assembly Instances

Number	Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
150	0	Low Byte Analog Input 1							
	1	High Byte Analog Input 1							
	2	Low Byte Analog Input 2							
	3	High Byte Analog Input 2							
	4	Low Byte Analog Input 3							
	5	High Byte Analog Input 3							
	6	Low Byte Analog Input 4							
	7	High Byte Analog Input 4							
	8	Reserved	Reserved	Reserved	Reserved	Status Analog Input4	Status Analog Input3	Status Analog Input2	Status Analog Input1

No Output Data (0 Bytes)

Mapping IO Assembly Data to DeviceNet Objects

Data Component Name	Class Name	Class Number	Instance	Attribute Name	Attribute Number
Analog Input n	Analog Input Point	10	1 – 4	Value	3
Status Analog Input n	Analog Input Point	10	1 – 4	Status	4

4.7 IP4xxx-B520

Input Data (1 Byte)

IO Assembly Instances

Number	Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
151	0	Reserved	Reserved	Reserved	Reserved	Status Analog Output4	Status Analog Output3	Status Analog Output2	Status Analog Output1

Output Data (8 Byte)

IO Assembly Instances

Number	Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
160	0	Low Byte Analog Output 1							
	1	High Byte Analog Output 1							
	2	Low Byte Analog Output 2							
	3	High Byte Analog Output 2							
	4	Low Byte Analog Output 3							
	5	High Byte Analog Output 3							
	6	Low Byte Analog Output 4							
	7	High Byte Analog Output 4							

Mapping IO Assembly Data to DeviceNet Objects

Data Component Name	Class Name	Class Number	Instance	Attribute Name	Attribute Number
Analog Output n	Analog Output Point	11	1 – 4	Value	3
Status Analog Output n	Analog Output Point	11	1 – 4	Status	4

4.8 IP1502-B520

Input Data (11 Byte)

IO Assembly Instances

Number	Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
172	0	Status Channel 1							
	1	DataIn[0] / Low Byte, Channel 1							
	2	DataIn[0] / High Byte, Channel 1							
	3	DataIn[1] / Low Byte, Channel 1							
	4	DataIn[1] / High Byte, Channel 1							
	5	Status Channel 2							
	6	DataIn[0] / Low Byte, Channel 2							
	7	DataIn[0] / High Byte, Channel 2							
	8	DataIn[1] / Low Byte, Channel 2							
	9	DataIn[1] / High Byte, Channel 2							
	10	FB_Error	res.	res.	res.	Diag	res.	IP_Cfg	IP_Error

Output Data (10 Byte)

IO Assembly Instances

Number	Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
181	0	Control Channel 1							
	1	DataOut[0] / Low Byte, Channel 1							
	2	DataOut [0] / High Byte, Channel 1							
	3	DataOut [1] / Low Byte, Channel 1							
	4	DataOut [1] / High Byte, Channel 1							
	5	Control Channel 2							
	6	DataOut0] / Low Byte, Channel 2							
	7	DataOut0] / High Byte, Channel 2							
	8	DataOut [1] / Low Byte, Channel 2							
	9	DataOut [1] / High Byte, Channel 2							

Mapping IO Assembly Data to DeviceNet Objects

Data Component Name	Class Name	Class Number	Instance	Attribute Name	Attribute Number
State + Data In n	Analog Input Channel	103	1	Value	3
Control + Data Out n	Analog Output Channel	104	1	Value	3
IP Status n	IP Config Object	100	1	IP-Status	5

4.9 IP60xx-B520

Input Data (7 Byte)

IO Assembly Instances

Number	Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
171	0	State							
	1	Data In 0							
	2	Data In 1							
	3	Data In 2							
	4	Data In 3							
	5	Data In 4							
	6	FB_Error	res.	res.	res.	Diag	res.	IP_Cfg	IP_Error

Output Data (6 Byte)

IO Assembly Instances

Number	Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
180	0	Control							
	1	Data Out 0							
	2	Data Out 1							
	3	Data Out 2							
	4	Data Out 3							
	5	Data Out 4							

Mapping IO Assembly Data to DeviceNet Objects

Data Component Name	Class Name	Class Number	Instance	Attribute Name	Attribute Number
Ser. State + Data In n	Analog Input Channel	103	1	Value	3
Ser. Control + Data Out n	Analog Output Channel	104	1	Value	3
IP Status n	IP Config Object	100	1	IP-Status	5

5 DeviceNet Objects

5.1 IP Config Object

The coupler provides the vendor specific object class to access its status, diagnostic and configuration data. Within the IP Config Object the full range of Registers and Status-Information of the IPxxxx and the channels is accessible.

Class Code: 100 (64_{hex})

Class Attributes

Attribute ID	Access Rule	Name	Data Type	Description	Semantics of Value
1	Get	Revision	UINT	Revision of implementation	1
2	Get	Max. Instance	UINT	Max. number of instances	1

Instance Attributes

Attribute ID	Access Rule	Name	Data Type	Description	Semantics of Value
1	Get/Set	Terminal Number	USINT	Number of terminal	0: Coupler >0: Channels
2	Get/Set	Table Number	USINT	Number of table	See IPxxxx / Channel Description
3	Get/Set	Register Number	USINT	Number of Register	See IPxxxx / Channel Description
4	Get/Set	Register Data	DWORD	Register Value	See semantics
5	Get	IP-Status	BYTE	Status of the IPxxxx	See semantics
6	Get	Channel Status	WORD	Status of the Ipxxxx Channel	See semantics
10	Get/Set	IO Error Action	BYTE	Action to be performed if a fieldbus error occurs	see semantics
15	Get	Input Size Poll Mode	BYTE	Number of Bytes produced via the Poll mode	
16	Get	Input Size Bit Strobe Mode	BYTE	Number of Bytes produced via the Bit Strobe mode	
17	Get	Input Size COS / Cyclic Mode	BYTE	Number of Bytes produced via the Change of State / Cyclic mode	
18	Get	Output Size Poll / COS/ Cyclic Mode	BYTE	Number of Bytes consumed via the Poll / Change of State /Cyclic mode	
25	Get/Set	Bus Off Behavior	BYTE	Behavior of IPxxxx-B520 after detection of a Bus-Off event	see semantics

5.1.1 Semantics

5.1.1.1 Register Data

Within the response of a Get_Attribute_Single Service to the „Register Data“ Attribute the status of the internal reading and the registers data is returned by the IPxxxx. The meaning of the registers data is described in the IPxxxx manual.

Within the request of a Set_Attribute_Single Service to the „Register Data“ Attribute the Low-Word of the attribute „RegisterData“ is used to send the Register data to the IPxxxx. The meaning of the registers data is described in the IPxxxx manual.

Response Data of Get_Attribute_Single

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	Status (Low Byte)							
1	Status (High Byte) : 0 = OK, >0 = Error							
2	Register data (Low Byte)							
3	Register data (High Byte)							

Request Data of Set_Attribute_Single

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	Register data (Low Byte)							
1	Register data (High Byte)							
2	Not used							
3	Not used							

Response Data of Set_Attribute_Single

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	Status (Low Byte)							
1	Status (High Byte) : 0 = OK, >0 = Error							

5.1.1.2 IP Status

The „IP Status“ attribute shows the actual status of the IPxxxx.

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Byte	FB_Error	res.	res.	res.	Diag	res.	IP_Cfg	IP_Error

IP_Error : IO Error, internal Data exchange IPxxxx has failed

IP_Cfg : IPxxxx Configuration Error

Diag : Diagnosis of analog Channel, cleared by reading Channel Status Attribute

FB_Error : Fieldbus Error / Idle Mode

5.1.1.3 Channel Status

The Channel Status attribute describes which channel of the Ipxxxx-B520 has encountered a diagnosis event. **After reading the diagnosis data, the attribute is cleared until the next diagnosis appears. Reading of the attribut also clears the Diag-Bit within the attribute „IP Status“.**

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	Terminal Number							
1	Status	Error Code					Channel Number	

Terminal Number : Number of faulty terminal

Channnel Number : Number of faulty channel of the terminal

Error Code : Terminal specific Error Code

Status : 0 = Error is reseted

1 = Error occured

5.1.1.4 IO Error Action

Action to be performed if a fieldbus error occurred.

Note: To activate a new IO error Action setting a device reset has to be performed either by executing a power cycle to the ILxxxx-B520 or by executing a Reset Service (Service Code 5) to the ILxxxx-B520 Identity Object (Class Id 1, Instance 1).

Value	Description
0	Leave local IO Cycle
1	Leave local IO Cycle and reset outputs (default)
2	freeze outputs

5.1.1.5 Bus Off Behavior

Action to be performed if a fieldbus error occurred.

Note: To activate a new Bus Off Behaviour setting a device reset has to be performed either by executing a power cycle to the ILxxxx-B520 or by executing a Reset Service (Service Code 5) to the ILxxxx-B520 Identity Object (Class Id 1, Instance 1).

Value	Description
0	Hold CAN in Bus Off (default)
1	Reset Communication / Device

5.2 Discrete Input Point Object

The IPxxxx provides Discrete Input Point Objects to access the discrete input channels data of the IPxxxx. For each discrete input point exits one instance of the Discrete Input Point Object Class.

For a detailed description of the Discrete Input Point Object see DeviceNet Specification Volume 2

Class Code: 8 (08_{hex})

Class Attributes

Attribute ID	Access Rule	Name	Data Type	Description	Semantics of Value
1	Get	Revision	UINT	Revision of implementation	1
2	Get	Max. Instance	UINT	Max. number of instances	

Instance Attributes

Attribute ID	Access Rule	Name	Data Type	Description	Semantics of Value
3 (03_{hex})	Get	Value	BOOL	Input Point Value	0: OFF 1: ON
4 (04_{hex})	Get	Status	BOOL	Input Point Status	0: OK 1: alarm / error, idle mode, data not valid

Common Services

Service Code	Service Name	Description
14 ($0E_{hex}$)	Get_Attribute_Single	Returns the contents of the specified attribute

5.3 Discrete Output Point Object

The IPxxxx provides Discrete Output Point Objects to access the discrete output channels data of the IPxxxx. For each discrete output point exits one instance of the Discrete Output Point Object Class.

For a detailed description of the Discrete Output Point Object see DeviceNet Specification Volume 2.

Class Code: 9 (09_{hex})

Class Attributes

Attribute ID	Access Rule	Name	Data Type	Description	Semantics of Value
1	get	Revision	UINT	Revision of implementation	1
2	get	Max. Instance	UINT	Max. number of instances	

Instance Attributes

Attribute ID	Access Rule	Name	Data Type	Description	Semantics of Value
3 (03_{hex})	Get/Set	Value	BOOL	Output Point Value	0: OFF 1: ON
4 (04_{hex})	Get	Status	BOOL	Output Point Status	0: OK 1: failure or alarm, idle mode data not valid
5 (01_{hex})	Get	Fault Action	BOOL	Action taken on Outputs value in Recoverable Fault State	see semantics
7 (01_{hex})	Get	Idle Action	BOOL	Action taken on Outputs value in Recoverable Fault State	see semantics

5.3.1 Semantics

5.3.1.1 Fault Action / Idle Action

The attribute values are according to the adjustment of the “IO Error Action Attribute” of the IP Config Object. (see chapter IP Config Object)

Value	Description
0	reset output, Fault Value Attribute default (“0”)
1	hold last state

Common Services

Service Code	Service Name	Description
14 ($0E_{hex}$)	Get_Attribute_Single	Returns the contents of the specified attribute
16 (10_{hex})	Set_Attribute_Single	Modifies an attribute Value

5.4 Analog Input Point Object

The Analog Input Point Class allows the access to the IO-Data and the Register Data of each analog channel of the IPxxxx. At Boot Up the IPxxxx determines the number of analog input channels and creates one instance of the object class for each channel.

Class Code: 10 ($0A_{hex}$)

Class Attributes

Attribute ID	Access Rule	Name	Data Type	Description	Semantics of Value
1	get	Revision	UINT	Revision of implementation	1
2	get	Max. Instance	UINT	Max. number of instances	

Instance Attributes

Attribute ID	Access Rule	Name	Data Type	Description	Semantics of Value
3 (03_{hex})	get	Value	INT or based on Attribute 8	Value of analog input	Number of Bytes differs between analog and special signal boxes
4 (04_{hex})	get	Status	BOOL	Indicates fault or alarm conditions	0 = OK, 1 = alarm or fault see semantics
7 (07_{hex})	get	Input Range	USINT	Input Range	0 = -10 to 10V, 2 = 0V to 10V 3 = 4mA to 20mA 8 = 0mA to 20mA 100 = Vendor Spec. 101 = Resistor 102 = Thermo 103 = SSI 104 = Increment 105 = Serial
8 (08_{hex})	get	Value Data Type	USINT	Datatype of Value	0 = INT 6 = UINT 100 = Vendor specific
100 (64_{hex})	get/set [1)]	Register 0	DWORD	Value of Register 0 of the analog input channel	See semantics
…	…	…	…	…	…
106 ($6A_{hex}$)	get	Register 6	DWORD	Diagnosis Register	see semantics
…	…	…	…	…	…
163 ($A3_{hex}$)	get/set [1)]	Register 63	DWORD	Value of Register 63 of the analog input channel	see semantics
170 (AA_{hex})	get	Value Length	USINT	Value length in Bytes	

[1)] Before writing the registers the write protection of the registers has to be disabled.

5.4.1 Semantics

5.4.1.1 Status

The Status of an Analog Input Point indicates the following conditions:

Value	Description
0	OK, no failure
1	internal IO Error Over range Under range Diagnostics (depends on IPxxxx, see manual) Idle Mode, no Output data via DeviceNet

5.4.1.2 Register 0…63

Within the response of a Get_Attribute_Single Service to the „Register Data“ Attribute the status of the internal reading and the registers data is returned by the IPxxxx. The meaning of the registers data is described in the IPxxxx manual.

Within the response of a Set_Attribute_Single Service to the „Register Data“ Attribute the status of the internal reading is returned by the coupler.

Get_Attribute Response

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	Status (Low Byte)							
1	Status (High Byte) : 0 = OK, >0 = Error							
2	Register data (Low Byte)							
3	Register data (High Byte)							

Set_Attribute Request

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	Register data (Low Byte)							
1	Register data (High Byte)							
2	Not used							
3	Not used							

Set_Attribute Response

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	Status (Low Byte)							
1	Status (High Byte) : 0 = OK, >0 = Error							

Common Services

Service Code	Service Name	Description
14 ($0E_{hex}$)	Get_Attribute_Single	Returns the contents of the specified attribute
16 (10_{hex})	Set_Attribute_Single	Modifies an attribute Value

5.5 Analog Output Point Object

The Analog Output Point Class allows the access to the IO-Data and the Register Data of each analog output channel of the IPxxxx. At Boot Up the IPxxxx determines the number of analog output channels and creates one instance of the object class for each channel.

Class Code: 11 ($0B_{hex}$)

Class Attributes

Attribute ID	Access Rule	Name	Data Type	Description	Semantics of Value
1	get	Revision	UINT	Revision of implementation	1
2	get	Max. Instance	UINT	Max. number of instances	

Instance Attributes

Attribute ID	Access Rule	Name	Data Type	Description	Semantics of Value
3 (03_{hex})	Get/set	Value	INT or based on Attribute 8	Value of analog output	Number of Bytes differs between analog and special signal boxes
4 (04_{hex})	Get	Status	BOOL	Indicates fault or alarm conditions	0 = OK, 1 = alarm or fault see semantics
7 (07_{hex})	Get	Output Range	USINT	Output Range	0 = 4mA to 20mA 1 = 0V to 10V 2 = 0mA to 20mA 3 = -100V to 10V 100 = Vendor Spec. 101 = Resistor 102 = Thermo 103 = SSI 104 = Increment 105 = Serial
8 (08_{hex})	Get	Value Data Type	USINT	Data type of Value	0 = INT 6 = UINT 100 = Vendor specific
9 (09_{hex})	Get	Fault State	BOOL	Output Value if failure occurs	See semantics
10 ($0A_{hex}$)	Get	Idle State	BOOL	Output Value if idle mode	See semantics

Instance Attributes

100 (64_{hex})	get/set [1)]	Register 0	DWORD	Value of Register 0 of the analog output channel	See semantics
:	:	:	:	:	:
106 ($6A_{hex}$)	get	Register 6	DWORD	Diagnosis Register	See semantics
:	:	:	:	:	:
163 ($A3_{hex}$)	get/set [1)]	Register 63	DWORD	Value of Register 63 of the analog output channel	See semantics
170 (AA_{hex})	get	Value Length	USINT	Value length in Bytes	

[1)] before writing the registers the write protection of the registers has to be disabled.

5.5.1 Semantics

5.5.1.1 Status

The Status of an Analog Output Point indicates the following conditions:

Value	Description
0	OK, no failure
1	internal IO Error Idle Mode, no Output data via DeviceNet Diagnostics (depends on IPxxxx, see manual) dle Mode, no Output data via DeviceNet

5.5.1.2 Fault State / Idle State

The attribute values are according to the adjustment of the “IO Error Action Attribute” of the IP Config Object. (see chapter IP Config Object)

Value	Description
0	hold last state
1	Low Limit, reset output

5.5.1.3 Register 0…63

Within the response of a Get_Attribute_Single Service to the „Register Data“ Attribute the status of the internal reading and the registers data is returned by the IPxxxx. The meaning of the registers data is described in the IPxxxx manual.

Within the response of a Set_Attribute_Single Service to the „Register Data“ Attribute the status of the internal reading is returned by the coupler.

Get_Attribute Response

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	Status (Low Byte)							
1	Status (High Byte) : 0 = OK, >0 = Error							
2	Register data (Low Byte)							
3	Register data (High Byte)							

Set_Attribute Request

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	Register data (Low Byte)							
1	Register data (High Byte)							
2	Not used							
3	Not used							

Set_Attribute Response

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	Status (Low Byte)							
1	Status (High Byte) : 0 = OK, >0 = Error							

Common Services

Service Code	Service Name	Description
14 ($0E_{hex}$)	Get_Attribute_Single	Returns the contents of the specified attribute
16 (10_{hex})	Set_Attribute_Single	Modifies an attribute Value

5.6 Analog Input Channel Object

The Analog Input Channel Class allows the access to the IO-Data and the Register Data of analog channels of the IPxxxx. At Boot Up the IPxxxx determines the number of analog input channels and creates one instance of the object class for each channel.

Class Code: 103 (67hex)

Class Attributes

Attribute ID	Access Rule	Name	Data Type	Description	Semantics of Value
1	get	Revision	UINT	Revision of implementation	1
2	get	Max. Instance	UINT	Max. number of instances	

Instance Attributes

Attribute ID	Access Rule	Name	Data Type	Description	Semantics of Value
3 (03_{hex})	get	Value	INT or based on Attribute 8	Value of analog input	Number of Bytes differs between analog and special signal boxes
4 (04_{hex})	get	Status	BOOL	Indicates fault or alarm conditions	0 = OK, 1 = alarm or fault see semantics
7 (07_{hex})	get	Input Range	USINT	Input Range	100 = Vendor Spec. 101 = Resistor 103 = SSI 104 = Increment 105 = Serial
8 (08_{hex})	get	Value Data Type	USINT	Datatype of Value	100 = Vendor specific
100 (64_{hex})	get/set [1)]	Register 0	DWORD	Value of Register 0 of the analog input channel	See semantics
:	:	:	:	:	:
106 ($6A_{hex}$)	get	Register 6	DWORD	Diagnosis Register	see semantics
:	:	:	:	:	:
163 ($A3_{hex}$)	get/set [1)]	Register 63	DWORD	Value of Register 63 of the analog input channel	see semantics
170 (AA_{hex})	get	Value Length	USINT	Value length in Bytes	

[1)] before writing the registers the write protection of the registers has to be disabled.

5.6.1 Semantics

5.6.1.1 Status

The Status of an Analog Input Channel indicates the following conditions:

Value	Description
0	OK, no failure
1	internal IO Error Over range Under range Diagnostics (depends on IPxxxx, see manual) Idle Mode

5.6.1.2 Register 0…63

Within the response of a Get_Attribute_Single Service to the „Register Data“ Attribute the status of the internal reading and the registers data is returned by the IPxxxx. The meaning of the registers data is described in the IPxxxx manual.

Within the response of a Set_Attribute_Single Service to the „Register Data“ Attribute the status of the internal reading is returned by the coupler.

Get_Attribute Response

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	Status (Low Byte)							
1	Status (High Byte) : 0 = OK, >0 = Error							
2	Register data (Low Byte)							
3	Register data (High Byte)							

Set_Attribute Request

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	Register data (Low Byte)							
1	Register data (High Byte)							
2	Not used							
3	Not used							

Set_Attribute Response

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	Status (Low Byte)							
1	Status (High Byte) : 0 = OK, >0 = Error							

Common Services

Service Code	Service Name	Description
14 ($0E_{hex}$)	Get_Attribute_Single	Returns the contents of the specified attribute
16 (10_{hex})	Set_Attribute_Single	Modifies an attribute Value

5.7 Analog Output Channel Object

The Analog Output Point Class allows the access to the IO-Data and the Register Data of each analog output channel of the IPxxxx. At Boot Up the Ipxxxx determines the number of analog output channels and creates one instance of the object class for each channel.

Class Code: 104 (68_{hex})

Class Attributes

Attribute ID	Access Rule	Name	Data Type	Description	Semantics of Value
1	get	Revision	UINT	Revision of implementation	1
2	get	Max. Instance	UINT	Max. number of instances	

Instance Attributes

Attribute ID	Access Rule	Name	Data Type	Description	Semantics of Value
3 (03_{hex})	Get/set	Value	INT or based on Attribute 8	Value of analog output	Number of Bytes differs between analog and special signal boxes
4 (04_{hex})	Get	Status	BOOL	Indicates fault or alarm conditions	0 = OK, 1 = alarm or fault see semantics
7 (07_{hex})	Get	Output Range	USINT	Output Range	100 = Vendor Spec. 101 = Resistor 103 = SSI 104 = Increment 105 = Serial
8 (08_{hex})	Get	Value Data Type	USINT	Data type of Value	100 = Vendor specific
9 (09_{hex})	Get	Fault State	BOOL	Output Value if failure occurs	0: Hold Last Value 1: Low Limit See semantics
10 ($0A_{hex}$)	Get	Idle State	BOOL	Output Value if idle mode	0: Hold Last Value 1: Low Limit See semantics

Instance Attributes

100 (64hex)	get/set 1)	Register 0	DWORD	Value of Register 0 of the analog output channel	See semantics
:	:	:	:	:	:
106 ($6A_{hex}$)	get	Register 6	DWORD	Diagnosis Register	See semantics
:	:	:	:	:	:
163 ($A3_{hex}$)	get/set [1)]	Register 63	DWORD	Value of Register 63 of the analog output channel	See semantics
170 (AA_{hex})	get	Value Length	USINT	Value length in Bytes	

[1)] Before writing the registers the write protection of the registers has to be disabled.

5.7.1 Semantics

5.7.1.1 Status

The Status of an Analog Output Point indicates the following conditions:

Value	Description
0	OK, no failure
1	internal IO Error Idle Mode, no Output data via DeviceNet Diagnostics (depends on IPxxxx, see manual) dle Mode, no Output data via DeviceNet

5.7.1.2 Fault State / Idle State

The attribute values are according to the adjustment of the “IO Error Action Attribute” of the IP Config Object. (see chapter IP Config Object)

Value	Description
0	hold last state
1	Low Limit, reset output

5.7.1.3 Register 0…63

Within the response of a Get_Attribute_Single Service to the „Register Data“ Attribute the status of the internal reading and the registers data is returned by the IPxxxx. The meaning of the registers data is described in the IPxxxx manual.

Within the response of a Set_Attribute_Single Service to the „Register Data“ Attribute the status of the internal reading is returned by the coupler.

Get_Attribute Response

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	Status (Low Byte)							
1	Status (High Byte) : 0 = OK, >0 = Error							
2	Register data (Low Byte)							
3	Register data (High Byte)							

Set_Attribute Request

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	Register data (Low Byte)							
1	Register data (High Byte)							
2	Not used							
3	Not used							

Set_Attribute Responset

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	Status (Low Byte)							
1	Status (High Byte) : 0 = OK, >0 = Error							

Common Services

Service Code	Service Name	Description
14 ($0E_{hex}$)	Get_Attribute_Single	Returns the contents of the specified attribute
16 (10_{hex})	Set_Attribute_Single	Modifies an attribute Value

6 Indicators and Switches

6.1 Start-up procedure and Diagnostics LEDs

Start-up procedure and Diagnostic

After switching on, the IPxxxx-B520 immediately checks the configuration. Error-free start-up is signalled by the red "I/O ERR“ LED being extinguished. If the “I/O ERR” LED blinks, an error is indicated. The error code can be determined from the frequency and number of blinks. This permits rapid rectification of the error. There is a detailed description in the section on "The diagnostic LEDs".

The diagnostic LEDs

The IPxxxx-B520 has two groups of LEDs for the display of status. The upper group with two LEDs indicates the status of the respective fieldbus. The significance of the “fieldbus status“ LED is explained in the relevant sections of this manual - it conforms to conventional fieldbus displays.

On the bottom of the IPxxxx-B520 are two more green LEDs that indicate the supply voltage. The left hand LED indicates the presence of the 24 V supply for the IPxxxx-B520. The right hand LED indicates the presence of the supply to the power contacts.

Local errors

Two LEDs, the “I/O LEDs”, in the area below the field bus status LEDs referred to above, serve to indicate the operating status of the IPxxxx-B520. The green LED lights up in order to indicate fault-free operation. The red LED blinks with two different frequencies in order to indicate an error. The error is encoded in the blinks as follows:

Blink code	Description
Fast blinking	Start of the error code
First slow sequence	Error code
Second slow sequence	Error code argument

Error code	Error code argument	Description
1 pulse	0	EEPROM checksum error
	1	Inline code buffer overflow
	2	Unknown data type
2 pulses	0	Programmed configuration
		Incorrect table entry / IPxxxx-B520
3 pulses	0	IP command error
4 pulses	0	IP data error
5 pulses	0	IP error in register communication
6 pulses	0	Special fieldbus error
	n (n > 0)	

The number of pulses in the first sequence indicates the error type, while the second sequence indicates the position of the the fault.

In the case of some errors, rectification does not cause the IPxxxx-B520 to leave the blink sequence. The IPxxxx-B520 stays in the "Stop" state. The IPxxxx-B520 can only be re-started either by switching the power supply off and on again, or by a scanner reset.

6.2 Fieldbus / DeviceNet LEDs

DeviceNet Status LED

The red/green LED pair provides information about the device and communication status of the IPxxxx-B520. The LEDs acts as the bi-color combined Module/Network Status LED defined in the DeviceNet Specification.

The LED Pair is located next to the configuration interface for adjustment of the DeviceNet address (MacId)

States of Module / Status LED

LED State	Description
Green Flashing	Boot Up OK, Device has executed Duplicate MacId Check and is ON-Line. The IPxxxx-B520 is not allocated by a Master / Scanner, no Data Exchange with a Master / Scanner
Green ON	No Error, IPxxxx-B520 is allocated by a Master / Scanner, Data Exchange (Explicit or IO) with Master / Scanner is OK
Green OFF	• Bus Sense Error (24V DeviceNet Voltage in not available) (all LEDs off, including IO-Run, IO-Error LEDs) • No BaudRate, IPxxxx-B520 is not able to detect BaudRate (IO-Run, IO-Error LEDs On)
Red Flashing	Time Out, IO-Connection has timed out
Red ON	• Duplicate MacId Fault, check for same Address in Network • Bus-Off, check cabling, check bus termination, check bus length • Receive/Transmit Overrun, reduce IO-Cycle Time / Interscan delay at Master / Scanner
Red OFF	• Bus Sense Error (24V DeviceNet Voltage in not available) (all LEDs off, including IO-Run, IO-Error LEDs) • No BaudRate, IPxxxx-B520 is not able to detect BaudRate (IO-Run, IO-Error LEDs On)

6.3 DeviceNet Node Address Switches

The Node Address Switches consist of two, ten position rotary switches within the Configuration Interface of the IPxxxx-B520.

Node Address	Description
0 - 63	Node Address from Switches is valid, not programmable
> 63	Node Address is programmable by Master / Scanner

7 Appendix

7.1 Support and Service

Beckhoff and their partners around the world offer comprehensive support and service, making available fast and competent assistance with all questions related to Beckhoff products and system solutions.

Beckhoff's branch offices and representatives

Please contact your Beckhoff branch office or representative for local support and service on Beckhoff products!

The addresses of Beckhoff's branch offices and representatives round the world can be found on her internet pages: https://www.beckhoff.com

You will also find further documentation for Beckhoff components there.

Beckhoff Support

Support offers you comprehensive technical assistance, helping you not only with the application of individual Beckhoff products, but also with other, wide-ranging services:

- support
- design, programming and commissioning of complex automation systems
- and extensive training program for Beckhoff system components

Hotline:	+49 5246 963 157
Fax:	+49 5246 963 9157
e-mail:	support@beckhoff.com

Beckhoff Service

The Beckhoff Service Center supports you in all matters of after-sales service:

- on-site service
- repair service
- spare parts service
- hotline service

Hotline:	+49 5246 963 460
Fax:	+49 5246 963 479
e-mail:	service@beckhoff.com

Beckhoff Headquarters

Beckhoff Automation GmbH & Co. KG

Huelshorstweg 20
33415 Verl
Germany

Phone:	+49 5246 963 0
Fax:	+49 5246 963 198
e-mail:	info@beckhoff.com
web:	https://www.beckhoff.com

Beckhoff Automation GmbH & Co. KG
Hülshorstweg 20
33415 Verl
Germany
Phone: +49 5246 9630
info@beckhoff.com
www.beckhoff.com