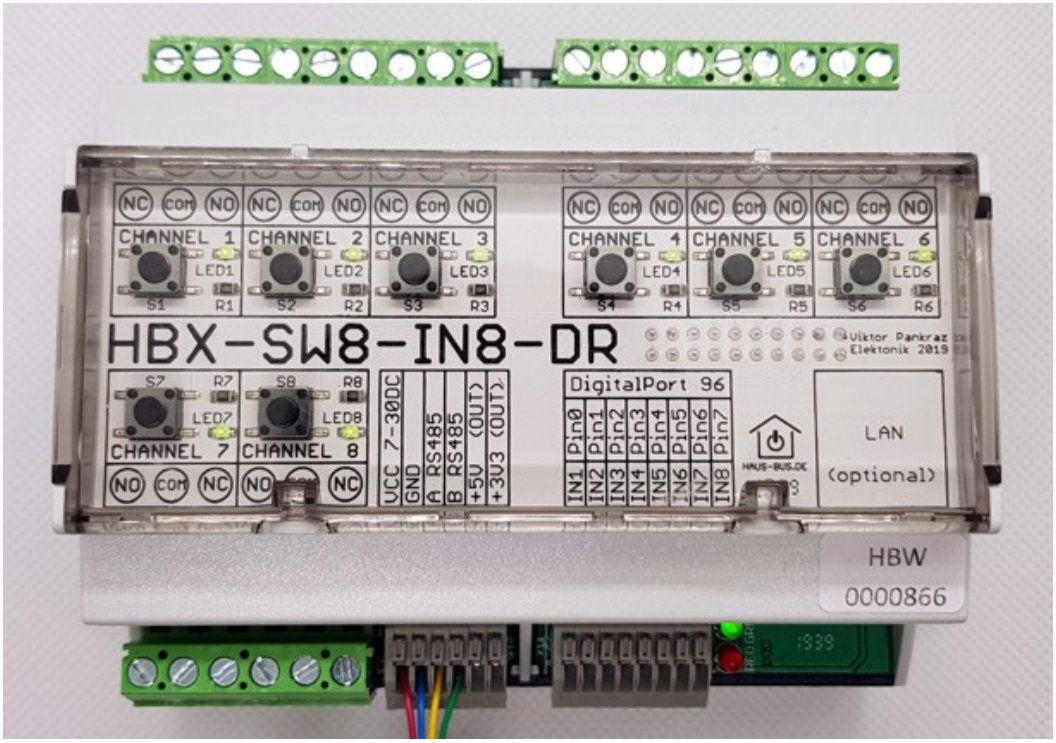
Produktbeschreibung

HBX-SW8-IN8-DR



Projekt	Haus Bus Controls
Edition	V0.1 – Arbeitsstand (25.05.2020)
Author	Dipl.-Ing. Pankraz Viktor

Copyright © Viktor Pankraz 2020
All rights reserved

Sie haben...

... Fragen und Anregungen zu dieser Produktbeschreibung?

Wenden Sie sich bitte unter Angabe der Quelle und Version dieser Beschreibung an:

Anschrift:

Viktor Pankraz Elektronik
Lippstädter Weg 94
D - 33758 Schloß-Holte Stukenbrock

E-Mail: manuals@pankraz.org

... technische Fragen oder Probleme?

Wenden Sie sich bitte an info@pankraz.org

1 Einleitung

Diese Produktbeschreibung beschreibt ausschließlich das Modul HBX-SW8-IN8-DR und dessen Einsatzmöglichkeiten in den verschiedenen Hausautomatisierungssystemen.
Sie liefert Ihnen Informationen über die Hardware und ggf. nötige Software-Voraussetzungen sowie die Handhabung der Komponenten, die Sie für die Anwendung des Moduls HBX-SW8-IN8-DR in einem Hausautomatisierungssystem benötigen.

Nach dem Studium dieser Produktbeschreibung sind Sie in der Lage:

- das Modul HBX-SW8-IN8-DR für die Anwendung im Hausautomatisierungs-system zu konfigurieren und in Betrieb zu nehmen,
- Störungen und Defekte zu identifizieren,
- und das Modul HBX-SW8-IN8-DR vorschriftsmäßig zu betreiben

1.1 Darstellungsmittel

- Texte, die dieser Markierung folgen, sind Aufzählungen.

„“ Texte in Anführungszeichen sind Verweise auf andere Kapitel oder Abschnitte.

- Texte, die dieser Markierung folgen, beschreiben Tätigkeiten, die Sie in der vorgegebenen Reihenfolge ausführen sollen.

Texte, die dieser Markierung folgen, müssen Sie besonders beachten, um Gefährdungen und Verletzungen zu vermeiden.

Texte, die dieser Markierung folgen, sind allgemeine Hinweise, die zur Erleichterung von Arbeiten oder Vermeidung von Fehlern beitragen.

1.2 Installationshinweise

Das Gerät kann für feste Installation in trockenen Innenräumen, zum Einbau in Starkstromverteiler oder Kleingehäuse auf Hutschienen EN 60715-TH35-7,5 verwendet werden. Installation und Verdrahtung sind entsprechend VDE 0100 (VDE 0100-410, VDE 0100-510 usw.) durchzuführen. Es sind die Vorschriften der Technischen Anschlussbestimmungen (TAB) des Energieversorgers zu berücksichtigen.

Das Modul ist nur für den Einsatz in Wohnbereichen, Geschäfts- und Gewerbebereichen oder in Kleinbetrieben bestimmt.

Bei Einsatz in einer Sicherheitsanwendung ist das Modul bzw. System in Verbindung mit einer USV (unterbrechungsfreie Stromversorgung) zu betreiben, um einen möglichen Netzausfall zu überbrücken.

Jeder andere Einsatz, als der in dieser Bedienungsanleitung beschriebene, ist nicht bestimmungsgemäß und führt zu Gewährleistungs- und Haftungsausschluss.

Aufgrund von Rundsteuersignalen kann es zu kurzzeitigem Flackern der Leuchtmittel kommen. Dies ist kein Defekt des Moduls.

Betreiben Sie das Gerät nur in trockener und staubfreier Umgebung und setzen Sie es keinem Einfluss von Feuchtigkeit, Vibrationen, ständiger Sonnen- oder anderer Wärmeeinstrahlung, übermäßiger Kälte und keinen mechanischen Belastungen aus.

Das Modul darf nicht verwendet werden, wenn es von außen erkennbare Schäden z. B. am Gehäuse, an Bedienelementen oder an den Anschlussbuchsen hat oder eine Funktionsstörung aufweist. Lassen Sie das Modul im Zweifelsfall von einer Fachkraft prüfen.

1.3 Wichtige Sicherheitshinweise

Bei Sach- oder Personenschäden, die durch unsachgemäße Handhabung oder Nichtbeachten der Sicherheitshinweise verursacht werden, übernehmen wir keine Haftung. In solchen Fällen erlischt jeder Gewährleistungsanspruch! Für Folgeschäden übernehmen wir keine Haftung!

Gefahr durch elektrischen Schlag. Das Modul HBX-SW8-IN8-DR ist nicht zum Freischalten geeignet. Auch bei ausgeschaltetem Ausgang ist die Last nicht galvanisch vom Netz getrennt.

Gefahr durch elektrischen Schlag. Vor Arbeiten am Modul HBX-SW8-IN8-DR oder vor Auswechseln von Leuchtmitteln Netzspannung freischalten und Sicherungsautomaten abschalten.

Das Modul HBX-SW8-IN8-DR darf nur von einer zugelassenen Elektrofachkraft installiert und in Betrieb genommen werden

Das Modul HBX-SW8-IN8-DR darf nicht geöffnet oder anderweitig modifiziert werden, es befinden sich keine durch den Anwender zu wartenden Teile darin

Die geltenden Sicherheits- und Unfallverhütungsvorschriften sind zu beachten

Bei der Planung und Errichtung von elektrischen Anlagen sind die einschlägigen Richtlinien, Vorschriften und Bestimmungen des jeweiligen Landes zu beachten. Der Betrieb des Moduls ist am 50 Hz-Wechselspannungsnetz mit maximal 230V zulässig. Arbeiten am 230-V-Netz dürfen nur von einer Elektrofachkraft (nach VDE 0100) erfolgen. Zur Vermeidung eines elektrischen Schlages am Modul, schalten Sie bitte die Netzspannung frei (Sicherungsautomat abschalten). Durch Nichtbeachtung der Installationshinweise können Brand oder andere Gefahren verursacht werden.

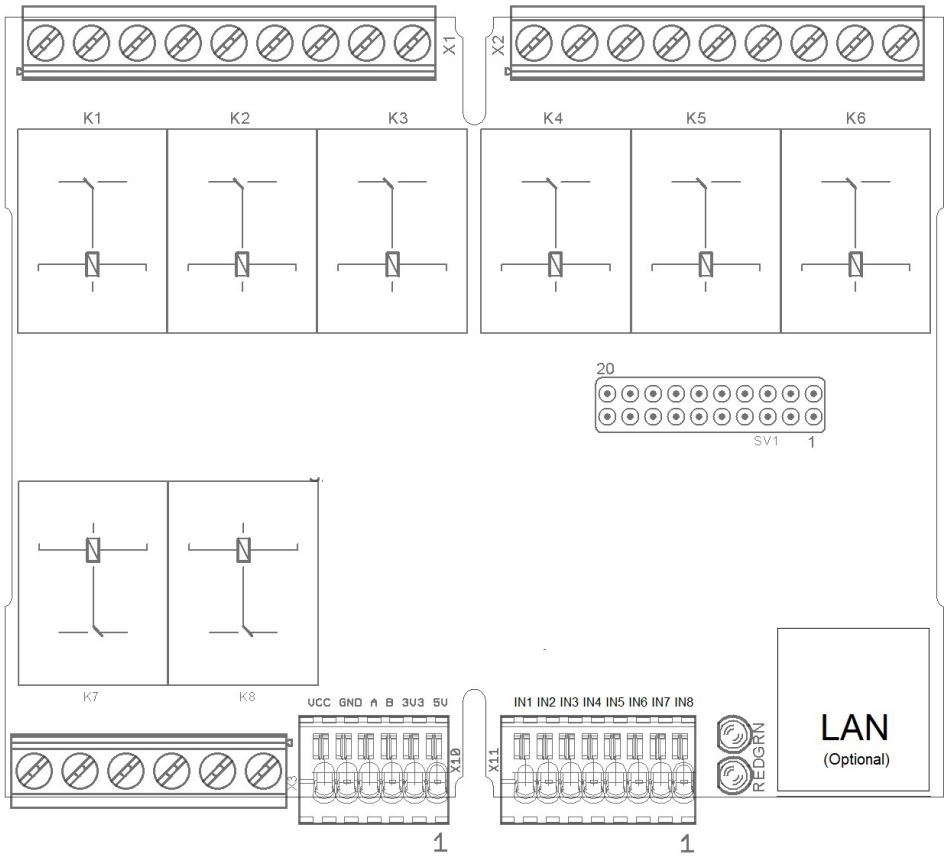
Die angeschlossenen Verbraucher müssen über eine ausreichende Isolierung verfügen.

Eine Überlastung kann zur Zerstörung des Geräts, zu einem Brand oder zu einem elektrischen Schlag führen.

Beachten Sie vor Anschluss eines Verbrauchers die technischen Daten, insbesondere die maximal zulässige Schaltleistung der Lastkreise und Art des anzuschließenden Verbrauchers. Belasten Sie den Aktor nur bis zur angegebenen Leistungsgrenze.

Setzen Sie beim Betrieb mit elektronischen Vorschaltgeräten nur Transformatoren ein, die den Anforderungen nach DIN EN 61347-1 sowie DIN EN 61047 entsprechen

2 Technische Daten



Ansicht von Oben: Stecker und Klemmenposition

Gehäuse:	ABS-Hutschienengehäuse,
	107mm x 90mm x 62mm (B x T x H)
Spannungsversorgung:	9-30V DC
Stromaufnahme:	100mA(Ethernet), max. 1000mA
Eingangsstecker(X11):	für 8x IN (max. 30V @1mA)
Ausgangsklemmleiste (X1/X2/X3):	
Schaltspannung:	277 VAC / 30 VDC
Schaltleistung:	max. 3 kW pro Ausgang
	max. 24 kW für gesamtes Modul
Optische Statusanzeige:	Status-LEDs (1 x rot, 1 x grün)
RS485-BUS:	Modul kann um 127 weitere erweitert werden

3 Funktion

Das Modul HBX-SW8-IN8-DR stellt 8 potentialfreie Schaltkontakte mit jeweils einem Öffner und Schließer-Kontakt zu Verfügung. Diese können alle voneinander unabhängig über die Verfügbaren Schnittstellen RS485 oder LAN (optional) angesteuert werden. Nach einem erfolgreichen Schaltvorgang sendet das Gerät eine entsprechende Quittung zurück. Des Weiteren werden an 8 digitale Eingänge die Zustände überwacht und bei Veränderungen entsprechend gemeldet.

Je nach gewählter Integration in ein Automatisierungssystem kann das Modul HBX-SW8-IN8-DR vom Benutzer definiertes Verhalten (z.B. Ablaufsteuerung, Überwachung, Regelung usw.), welches in Form eines Regelwerkes in das Modul geladen werden kann, selbstständig und ohne die Ansteuerung durch z.B. externe Server oder Steuersoftware durchzuführen.

Die Kommunikation mit dem Modul HBX-SW8-IN8-DR kann über Ethernet oder RS485 stattfinden. Bei bestehender Verbindung werden alle Ereignisse innerhalb des Moduls auch über alle Schnittstellen gemeldet.

3.1 Eingänge

In der Standardausführung des Moduls HBX-SW8-IN8-DR sind die Eingänge auf den Stecker X11 angeschlossen. Die zu verarbeitenden Signale sollten nur im Bereich 0-30VDC liegen. Spannungen > 2V werden als Zustand ‚1' erkannt. Um den Zustand ‚0' zu erreichen, muss die Spannung am Eingang unter 1 Volt betragen.

3.1.1 Aktionen

Bezeichnung	Parameter	Beschreibung
Aktivieren	-	Mit dieser Aktion werden alle konfigurierten Ereignisse bei ihrem künftigen Auftreten gemeldet.
Deaktivieren	Dauer der Deaktivierung	Mit dieser Aktion werden für die Dauer der Deaktivierung keine Ereignisse gemeldet.

3.1.2 Ereignisse

Jeder Eingang kann folgende Ereignisse melden:

Bezeichnung	Parameter	Beschreibung
Betätigt	-	Zustand ‚0'
Nicht Betätigt	-	Zustand ‚1'
Geklickt	-	Zustand ‚1' – ‚0' – ‚1'
Gehalten	-	Zustand ‚0' für eine bestimmte Zeit, konfigurierbar
Losgelassen	-	Zustand ‚1' nur nach dem Ereignis gehalten
Mehrfach Geklickt	Anzahl Klicks	Zustand ‚1' – ‚0' – ‚1'– ‚0' – ‚1' - … Alle Klicks werden summiert solange eine definierte Pausenzeit nicht überschritten wird.

3.1.3 Konfiguration

Die Zustandswechsel zwischen den Zuständen ‚0' und ‚1' sind mit 40ms entprellt.

Bezeichnung	Wertebereich	Beschreibung
Haltezeit	0,1 – 2,5s	Mindestzeit im Zustand ‚0' um das Ereignis **Gehalten** zu senden
Pausenzeit	0,1 – 2,5s	Maximale Pausenzeit um Mehrfachklicks zu zählen
Aktive Ereignisse	Betätigt	Aktiviert / Deaktiviert das Ereignis
	Nicht Betätigt	Aktiviert / Deaktiviert das Ereignis
	Geklickt	Aktiviert / Deaktiviert das Ereignis
	Gehalten	Aktiviert / Deaktiviert das Ereignis
	Losgelassen	Aktiviert / Deaktiviert das Ereignis
	Mehrfach Geklickt	Aktiviert / Deaktiviert das Ereignis

3.1.4 Software IDs

Die Eingänge haben in der Konfigurationssoftware folgende IDs und Bezeichnungen:

Stecker Bezeichnung	Digital Port-ID	Digital Pin- ID	Software Objekt-ID	Software Bezeichnung
X11– IN1	96	0	XXXX1061h	Taster 97
X11– IN2	96	1	XXXX1062h	Taster 98
X11– IN3	96	2	XXXX1063h	Taster 99
X11– IN4	96	3	XXXX1064h	Taster 100
X11– IN5	96	4	XXXX1065h	Taster 101
X11– IN6	96	5	XXXX1066h	Taster 102
X11– IN7	96	6	XXXX1067h	Taster 103
X11– IN8	96	7	XXXX1068h	Taster 104

Die Bezeichnung in der Software lässt sich aber jederzeit durch den Benutzer verändern!

3.2 Ausgänge

Die Relaiskontakte des Moduls HBW-LC-SW16-IN8-DR sind auf die WAGO Klemmen X1 bis X3 rausgeführt. Die einzelnen Klemmstellen sind folgendermaßen anzuschließen:

	X1.1	**X1.2**	**X1.3**	**X1.4**	**X1.5**	**X1.6**	**X1.7**	**X1.8**	**X1.9**
Relais	NC 1	L 1	NO 1	NC 2	L 2	NO 2	NC 3	L 3	NO 3

	X2.1	**X2.2**	**X2.3**	**X2.4**	**X2.5**	**X2.6**	**X2.7**	**X2.8**	**X2.9**
Relais	NC 4	L 4	NO 4	NC 5	L 5	NO 5	NC 6	L 6	NO 6

	X3.1	**X3.2**	**X3.3**	**X3.4**	**X3.5**	**X3.6**
Relais	NO 7	L 7	NC 7	NO 8	L 8	NC 8

Es ist außerdem möglich, die auf dem Stecker X11 befindliche Eingänge als Ausgänge zu konfigurieren und zu verwenden. Weitere Informationen dazu finden sich im Kapitel X.X (ToDo).

3.2.1 Aktionen

Jeder Ausgang kann folgende Aktionen durchführen, die aus einem Kommando oder Reaktion auf eine Regel entstehen können:

Bezeichnung	Parameter	Beschreibung
Ausschalten	-	Der Ausgang wird deaktiviert.
Einschalten	Dauer	Der Ausgang wird aktiviert. Wird eine Dauer vorgegeben, so deaktiviert sich der Ausgang nach Ablauf dieser Zeit automatisch.
Umschalten	Anzahl Dauer ‚AUS' Dauer ‚AN'	Anzahl der Umschaltvorgänge Verweildauer im Zustand ‚AUS' Verweildauer im Zustand ‚AN'

3.2.2 Ereignisse

Jeder Ausgang kann folgende Ereignisse melden:

Bezeichnung	Parameter	Beschreibung
Ausgeschaltet	-	Zustand ‚AUS' wurde erreicht
Eingeschaltet	-	Zustand ‚AN' wurde erreicht
Umgeschaltet	-	Eine Umschaltung hat begonnen. Nach Abschluss der Umschaltvorgänge wird der Zustand über ein weiteres Ereignis gemeldet.

3.2.3 Konfiguration

Bezeichnung	Wertebereich	Beschreibung
Max. Einschaltdauer	0 1 - 255	Keine maximale Einschaltdauer vorgegeben Wert x Zeitbasis ergibt die Zeitdauer, nach der der Ausgang automatisch deaktiviert wird.
Ausschaltverzögerung	0 - 255	Wert x Zeitbasis ergibt die Zeitdauer, um die der Ausschaltvorgang verzögert wird.
Zeitbasis	0,01 – 65s	Diese Zeit wird als Basis für alle Parameter der Aktionen und Konfigurationen verwendet.

3.2.4 Software IDs (Relais)

Die Ausgänge haben in der Konfigurationssoftware folgende IDs und Bezeichnungen:

Stecker Bezeichnung	Slot	Software Objekt-ID	Software Bezeichnung
X1 – NO/NC 1	1	XXXX1301h	Schalter 1
X1 – NO/NC 2	2	XXXX1302h	Schalter 2
X1 – NO/NC 3	3	XXXX1303h	Schalter 3
X2 – NO/NC 4	4	XXXX1304h	Schalter 4
X2 – NO/NC 5	5	XXXX1305h	Schalter 5
X2 – NO/NC 6	6	XXXX1306h	Schalter 6
X3 – NO/NC 7	7	XXXX1307h	Schalter 7
X3 – NO/NC 8	8	XXXX1308h	Schalter 8
LED (rot)	-	XXXX13D2h	Schalter 210

Die Bezeichnung in der Software lässt sich aber jederzeit durch den Benutzer verändern!

4 Kommunikationsprotokoll

Das Modul HBX-SW8-IN8-DR unterstützt folgende Kommunikationsprotokolle:

- proprietäres Hausbusprotokoll (UDP Port 9)
- Modbus TCP/IP
- Loxone RS485 / LAN
- Homematic RS485

4.1 Hausbusprotokoll UDP Port 9

Zur Kommunikation zwischen den einzelnen Teilnehmern wird ein Objekt orientiertes Protokoll verwendet, das wie folgt definiert ist.

4.1.1 Allgemeine Form der Nachrichten

Aufbau einer Nachricht

Position	Länge	Bedeutung	Kommentar
- L_H	L_H	Header	Der Header hat eine variable Länge, die abhängig von der Bus-Hardware ist. Er wird von den jeweiligen Gateways vor dem versenden der Nachricht an andere Geräte erzeugt bzw. bei interner Kommunikation der Objekte entfernt.
0	12	ControlFrame	Dieser Block enthält Informationen über Sender und Empfänger der Nachricht und die Länge der Daten
12	L_D	Data	Formatfreie Daten die vom Empfänger interpretiert werden müssen

UDP Port 9 - Header

Position	Länge	Attribut	Kommentar
0	2	*magicNumber*	Erkennungszahl 0xEFEF

RS485-Header

Position	Länge	Attribut	Kommentar
0	1	*StartZeichen*	0xFD

ControlFrame

Position	Länge	Attribut	Kommentar
0	1	Kontroll-Byte	Enthält Priorität, Nachrichtentyp und Wiederholungsbit
1	1	Nachrichtenzähler	Wird mit jeder versendeten Nachricht inkrementiert
2	4	Sender-ID	32-Bit Objekt-ID des Senders
6	4	Empfänger-ID	32-Bit Objekt-ID des Empfängers
10	2	Datenlänge	Länge des folgenden Datenpakets

Object-ID

Position	Länge	Attribut	Kommentar
0	2	*deviceId*	16-Bit Controller ID
2	1	*classId*	8-Bit Klassen ID
3	1	*instanceId*	8-Bit Instanz ID

Broadcast-Nachrichten

Alle Nachrichten, die in der Empfänger-ID das Attribut *deviceId* oder *instanceId* auf den Wert 0 setzten, sind Broadcast-Nachrichten. D.h. für diese Nachrichten gibt es möglicherweise mehrere Empfänger. Da jede Klasse einen unabhängigen Satz an Kommandos und Events besitzt, kann eine Nachricht niemals an alle unterschiedlichen Klassen gesendet werden. Deshalb wird über die Klassen-ID der Typ des Objektes immer ausgewählt (siehe nächstes Kapitel).

Durch Setzen der Controller-ID auf den Wert 0 wird erreicht, dass die Nachricht an alle sich im Netzwerk befindlichen Controller verschickt wird. Ist auch die Instanz-ID gleich Null, so werden alle Instanzen der ausgewählten Klasse innerhalb des Controllers diese Nachricht erhalten.

Es ist somit möglich alle Instanzen einer Klasse auf allen Controllern, alle Instanzen einer Klasse auf einem bestimmten Controller oder eine bestimmte Instanz auf allen Controllern anzusprechen.

4.1.2 Objekttypen (Klassen)

Alle verfügbaren Objekte in einem Modul werden eindeutig über ihre *classId* und *instanceId* identifiziert. Folgende Klassen sind im Modul HBX-SW8-IN8-DR verfügbar:

ID	Name	Funktion
0	Controller	Der Controller ist die Wurzel des Objektbaumes und in jedem Modul nur einmal vorhanden. Seine primäre Funktion ist es, das Modul grundsätzlich zu konfigurieren.
14	ModbusSlave	Der ModbusSlave bietet die Funktionalität des Moduls über die MODBUS-Schnittstelle auf dem Netzwerk mittels TCP/IP auf dem Port 502 an. Diese Klasse muss im Ethernet erst aktiviert werden.
15	DigitalPort	Der DigitalPort bietet die Möglichkeit alle zu Verfügung stehenden Pins zu konfigurieren. Pro Port sind maximal 8 Pins verfügbar.
16	Taster	Der Taster wertet Signale auf einem Portpin digital aus und erzeugt entsprechende Events.
19	Schalter	Der Schalter steuert einen PortPin als Ausgang an.
162	Ethernet	Ethernet bietet grundsätzliche Netzwerkfunktionen wie z.B. Konfiguration (IP, DHCP, MODBUS, SNMP) an
163	SnmpAgent	Der SnmpAgent bietet die Funktionalität des Moduls über die SNMP-Schnittstelle auf dem Netzwerk mittels UDP auf dem Port 161 an. Diese Klasse muss im Ethernet erst aktiviert werden.
176	Gateway	Das Gateway ist für die Verteilung der Nachrichten zuständig, jede Hardware Schnittstelle (USB, RS485, Ethernet) realisiert solch ein Gateway um Nachrichten, die über eine Schnittstelle eintreffen an die anderen weiter zu leiten.

4.1.3 Controller

Kommandos

GENERATE_RANDOM_DEVICE_ID (0)

Das Gerät soll sich eine Zufällige ID vergeben.

Parameter: keine

Antwort: keine

RESET (1)

Das Modul wird neugestartet.

Parameter: keine

Antwort: keine

GET_MODULE_ID (2)

Die durch den im Parameter festgelegte Modul-Kennung soll gemeldet werden.

Parameter:

Byte 1: index (0=aktives Modul, 1=inaktives Modul)

Antwort: MODULE_ID (128)

GET_REMOTE_OBJECTS (3)

Es soll eine Liste aller Objekte zurückgegeben werden, die Nachrichten erhalten und darauf reagieren können.

Parameter: keine

Antwort: REMOTE_OBJECTS (129)

GET_UNUSED_MEMORY (4)

Die Größe des bisher ungenutzten RAM-Speichers soll zurückgegeben werden.

Parameter: keine

Antwort: UNUSED_MEMORY (130)

GET_CONFIGURATION (5)

Die gesamte Konfiguration soll gemeldet werden.

Parameter: keine

Antwort: CONFIGURATION (131)

SET_CONFIGURATION (6)

Die gesamte Konfiguration des Objektes soll gesetzt werden.

Parameter:

Byte 1: startupDelay (0-255 * 10ms)

Byte 2: logicalButtonMask

Byte 3: deviceIdByte0

Byte 4: deviceIdByte1

Byte 5: slotType0
Byte 6: slotType1
Byte 7: slotType2
Byte 8: slotType3
Byte 9: slotType4
Byte 10: slotType5
Byte 11: slotType6
Byte 12: slotType7

Antwort: keine

READ_MEMORY (7)

Der Speicherinhalt ab einer bestimmten Adresse und Länge soll gelesen werden.*

Parameter:

Byte 1: addressByte0
Byte 2: addressByte1
Byte 3: addressByte2
Byte 4: addressByte3
Byte 5: lengthByte0
Byte 6: lengthByte1

Antwort: MEMORY_DATA

WRITE_MEMORY (8)

Der Speicherinhalt ab einer bestimmten Adresse und Länge soll geschrieben werden.*

Parameter:

Byte 1: addressByte0
Byte 2: addressByte1
Byte 3: addressByte2
Byte 4: addressByte3
Byte n: data

Antwort: MEMORY_STATUS

* Aufgrund der Prozessor-Architektur liegen Flash und Daten im gleichen Adressraum. Zur Unterscheidung wird für die Adressierung der Daten(EEPROM, SRAM, …) das Bit 30 der

Adresse gesetzt. Somit adressiert man mit der Adresse 0x00000000 – 0x3FFFFFFF den Flashbereich. Auf der Adresse 0x40000000 – 0x4000FFFF liegen die internen IO-Register, das Schreiben dieser Register wird von der Firmware abgelehnt. Im Adressbereich 0x40010000 – 0x4001FFFF kann das EEPROM gelesen bzw. beschrieben werden. Ab Adresse 0x40020000 befindet sich das interne SRAM.

WRITE_RULES (9)

Es soll ein Teil der Regeln in das EEPROM geschrieben werden. Die erste Übertragung deaktiviert die Regel-Engine. Um diese wieder in Betrieb zu nehmen, wird ein Reset benötigt.

Parameter:

Byte 1: offsetByte0

Byte 2: offsetByte1

Byte n: data

Antwort: MEMORY_STATUS

PING (127)

Der Controller wird „angepingt“.

Parameter: keine

Antwort: PONG

Antworten

MODULE_ID (128)

Die Modul-Kennung des Gerätes wird gemeldet.

Parameter:

Byte1-16: name (NULL terminierter String)

Byte 17: sizeByte0 (Modulgröße in Bytes)

Byte 18: sizeByte1

Byte 19: sizeByte2

Byte 20: sizeByte3

Byte 21: majorRelease (Release-Kennung im Format *major.minor*)

Byte 22: minorRelease

Byte 23: reserved

Byte 24: checksum (Checksumme über die gesamte Firmware)

REMOTE_OBJECTS (129)

Eine Liste der Verfügbaren Objekte im Geräte wird zurückgegeben.

Parameter:

Byte(2*n): classId

Byte(2*n-1): instanceId

UNUSED_MEMORY (130)

Der seit Programmstart ungenutzte RAM-Speicher wird zurückgegeben.

Parameter:

Byte1: freeStack Byte0

Byte2: freeStack Byte1

Byte3: freeHeep Byte0

Byte4: freeHeep Byte1

CONFIGURATION (131)

Die gesamte Konfiguration des Objektes wird gemeldet.

Parameter:

Byte 1: startupDelay (0-255 * 10ms)

Byte 2: logicalButtonMask

Byte 3: deviceIdByte0

Byte 4: deviceIdByte1

Byte 5: slotType0

Byte 6: slotType1

Byte 7: slotType2

Byte 8: slotType3

Byte 9: slotType4

Byte 10: slotType5

Byte 11: slotType6

Byte 12: slotType7

Byte 13: timeCorrection

Byte 14: reserved

Byte 15: reserved

Byte 16: dataBlockSizeByte0 (maximale Größe des Datenblocks in einer Nachricht)

Byte 17: dataBlockSizeByte1

Byte 18: FCKE (Bauzustand der Elektronik)

MEMORY_DATA (132)

Der Inhalt eines Speichers wird gemeldet.

Parameter:

Byte 1: addressByte0

Byte 2: addressByte1

Byte 3: addressByte2

Byte 4: addressByte3

Byte n: data

MEMORY_STATUS (133)

Der Speicherstatus wird gemeldet.

Parameter:

Byte 1: status (0=alles OK, 255=Firmware nicht vorhanden oder korrupt, sonstiger Fehlercode)

PONG (199)

Der Controller antwortet auf ein PING.

Parameter: keine

4.1.4 ModbusSlave

Kommandos

GET_CONFIGURATION (0)

Die gesamte Konfiguration soll gemeldet werden.

Parameter: keine

Antwort: CONFIGURATION (128)

SET_CONFIGURATION (1)

Die gesamte Konfiguration soll gesetzt werden.

Parameter: keine (noch nicht verwendet)

Antwort: keine

READ_COIL_STATUS (2)

Die digitalen Ausgänge sollen gelesen werden.

Parameter:

Byte 1: regIdByte0 (Startregister ab dem gelesen werden soll)

Byte 2: regIdByte1

Byte 2: length (Anzahl der digitalen Ausgänge, die gelesen werden sollen)

Antwort: COIL_STATUS (129)

READ_INPUT_STATUS (3)

Die digitalen Eingänge sollen gelesen werden.

Parameter:

Byte 1: regIdByte0 (Startregister ab dem gelesen werden soll)

Byte 2: regIdByte1

Byte 2: length (Anzahl der digitalen Eingänge, die gelesen werden sollen)

Antwort: INPUT_STATUS (130)

READ_HOLDING_REGISTERS (4)

Die analogen Eingänge sollen gelesen werden.

Parameter:

Byte 1: regIdByte0 (Startregister ab dem gelesen werden soll)

Byte 2: regIdByte1

Byte 2: length (Anzahl der analogen Eingänge, die gelesen werden sollen)

Antwort: HOLDING_REGISTERS (131)

Antworten

CONFIGURATION (128)

Parameter: keine (aktuell nicht verwendet)

COIL_STATUS (129)

Parameter:

Byte 1: coilState (Status der ersten 8 digitalen Ausgänge ab regID)

Byte n: coilState (Status der nächsten 8 digitalen Ausgänge)

INPUT_STATUS (130)

Parameter:

Byte 1: inputState (Status der ersten 8 digitalen Eingänge ab regID)

Byte n: inputState (Status der nächsten 8 digitalen Eingänge)

HOLDING_REGISTERS (131)

Parameter:

Byte 2*n-1: holdingRegByte0 (analoger 16Bit-Wert ab regID)

Byte 2*n: holdingRegByte1

4.1.5 DigitalPort

In der Klasse DigitalPort werden bis zu 8 Pins zu einem Port zusammengefasst. Diese liegen natürlich auch alle auf **einem** physikalischen Port des Prozessors. Sobald ein Pin durch die Konfiguration einer Klasse zugeordnet wird, können die Objekte nach einem Neustart verwendet werden.

Kommandos

GET_CONFIGURATION (0)

Die gesamte Konfiguration soll gemeldet werden.

Parameter: keine

Antwort: CONFIGURATION (128)

SET_CONFIGURATION (1)

Die gesamte Konfiguration des Objektes soll gesetzt werden.

Parameter:

Byte 1: pin0 (classId der gewünschten Funtion für den Pin0)

| (classId der gewünschten Funtion für den PinX)

Byte 8: pin7 (classId der gewünschten Funtion für den Pin7)

Antwort: keine

Antworten

CONFIGURATION (128)

Die gesamte Konfiguration des Objektes wird gemeldet.

Parameter:

Byte 1: pin0 (classId für den Pin0)

| (classId für den PinX)

Byte 8: pin7 (classId für den Pin7)

4.1.6 Taster

Das Verhalten des Tasters ist folgendermaßen definiert:

In den jeweiligen notify…() wird immer abhängig von der Konfiguration die Nachricht gesendet.
HOLD_TIME ist die Zeit, die der Taster mindestens gedrückt bleiben soll, um nach HOLD wechseln zu können.
CLICK_TIME ist die Zeit, die gewartet werden muss bevor man die Nachricht „geklickt“ senden kann. (nur wenn Doppelklick aktiv ist)

Kommandos

GET_CONFIGURATION (0)

Die gesamte Konfiguration soll gemeldet werden.

Parameter: keine

Antwort: CONFIGURATION (128)

SET_CONFIGURATION (1)

Die gesamte Konfiguration des Objektes soll gesetzt werden.

Parameter:

Byte 1: holdTimeout(0-255 a 10ms)

Byte 2: waitForDoubleClickTimeout(0-255 a 10ms)

Byte 3: eventMask

Bit	Event	Bedeutung
0	notifyOnCovered	**0 = keine Nachricht wenn gedrückt** **1 = Nachricht *evCovered* senden**
1	notifyOnClicked	**0 = keine Nachricht wenn geklickt** **1 = Nachricht *evClicked* senden**
2	notifyOnDoubleClicked	**0 = keine Nachricht wenn doppelt geklicked** **1 = Nachricht *evDoubleClicked* senden**
3	notifyOnHoldStart	**0 = keine Nachricht wenn gehalten** **1 = Nachricht *evHoldStart* senden**
4	notifyOnHoldEnd	**0 = keine Nachricht wenn losgelassen nach gehalten** **1 = Nachricht *evHoldEnd* senden**
5	notifyOnFree	**0 = keine Nachricht wenn losgelassen** **1 = Nachricht *evFree* senden**

Antworten

CONFIGURATION (128)

Die gesamte Konfiguration des Objektes wird gemeldet.

Parameter:

Byte 1: holdTimeout(0-255 a 10ms)

Byte 2: waitForDoubleClickTimeout (0-255 a 10ms)

Byte 3: eventMask

EVENT_COVERED (200)

Ein Taster wurde gedrückt.

Parameter: keine

EVENT_CLICKED (201)

Ein Taster wurde geklickt.

Parameter: keine

EVENT_DOUBLE_ CLICKED (202)

Ein Taster wurde doppelt geklickt.

Parameter: keine

EVENT_ HOLD_START (203)

Ein Taster wird gehalten.

Parameter: keine

EVENT_HOLD_END (204)

Ein Taster wurde gehalten und jetzt losgelassen.

Parameter: keine

EVENT_FREE (205)

Ein Taster wurde losgelassen.

Parameter: keine

4.1.7 Schalter

Kommandos

GET_CONFIGURATION (0)

Die gesamte Konfiguration soll gemeldet werden.

Parameter: keine

Antwort: CONFIGURATION (128)

SET_CONFIGURATION (1)

Die gesamte Konfiguration des Objektes soll gesetzt werden.

Parameter:

Byte 1: dutyOffset (nicht relevant)

Byte 2: minDuty (nicht relevant)

Byte 3: timeBaseByte0 (Zeitbasis für Schaltbefehle [ms])

Byte 4: timeBaseByte1 (Zeitbasis für Schaltbefehle [ms])

Byte 5: options (invert, driveOnState, driveOffState)

OFF (2)

Es soll ausgeschaltet werden.

Parameter:

ON (3)

Es soll eingeschaltet werden.

Parameter:

Byte 1: durationByte0 (Einschaltdauer)

Byte 2: durationByte1

TOGGLE (2)

Es soll der jeweilige andere Zustand eingenommen werden.

Parameter:

Byte 1: offTimeByte0 (Ausschaltdauer)

Byte 2: offTimeByte1

Byte 3: onTimeByte0 (Einschaltdauer)

Byte 4: onTimeByte1

Byte 5: quantity (Anzahl der Zustandswechsel)

GET_STATUS (3)

Es soll der Status des Schalters gemeldet werden.

Parameter:

Antwort: STATUS (129)

Antworten

CONFIGURATION (128)

Die gesamte Konfiguration des Objektes wird gemeldet.

Parameter:

Byte 1: dutyOffset (nicht relevant)

Byte 2: minDuty (nicht relevant)

Byte 3: timeBaseByte0 (Zeitbasis für Schaltbefehle [ms])

Byte 4: timeBaseByte1 (Zeitbasis für Schaltbefehle [ms])

Byte 5: options (invert, driveOnState, driveOffState)

STATUS(129)

Status des Schalters wird gemeldet.

Parameter:

Byte 1: status (0=ausgeschaltet, 1=eingeschaltet)

EVENT_OFF (200)

Der Schalter wurde ausgeschaltet.

Parameter: keine

EVENT_ON (201)

Der Schalter wurde eingeschaltet.

Parameter: keine

EVENT_TOGGLE (201)

Der Schalter wechselt seinen Zustand.

Parameter: keine

4.2 Modbus TCP

Das Modbus-Protokoll ist ein offenes Kommunikationsprotokoll und ist in der Version Modbus TCP Teil der Norm IEC 61158 und wird in IEC 61784-2 als CPF 15/1 referenziert. Das Modbus-Protokoll wurde ursprünglich von der Firma Modicon für den Datenverkehr mit ihren Controllern entwickelt. Daten wurden in Form von 16-Bit-Registern (Integer-Format) oder als Status-Informationen in Form von Datenbytes übertragen. Im Laufe der Zeit wurde das Protokoll erweitert und auch von anderen Herstellern für ihre Geräte übernommen. Neue Datentypen wurden hinzugefügt, insbesondere um mehr Auflösung für die übertragenen Werte zu erhalten. Der grundsätzliche Aufbau des Datenbereichs und die Adressierungs-Mechanismen wurden dabei aber aus Kompatibilitätsgründen immer beibehalten. Das Modbus-Protokoll ist ein Single-Master Protokoll. Dieser Master steuert die gesamte Übertragung und überwacht eventuell auftretende Timeouts (keine Antwort vom adressierten Gerät). Die angeschlossenen Geräte dürfen nur nach Anforderung durch den Master Telegramme versenden.

4.2.1 Register IDs Eingänge

Die Eingänge können über folgende Register über das Modbus-Protokoll abgefragt werden:

Stecker Bezeichnung	Register ID
X11 – IN1	0x0030
X11 – IN2	0x0031
X11 – IN3	0x0032
X11 – IN4	0x0033
X11 – IN5	0x0034
X11 – IN6	0x0035
X11 – IN7	0x0036
X11 – IN8	0x0037

4.2.2 Register IDs Ausgänge

Die Ausgänge können über folgende Register über das Modbus-Protokoll geschaltet werden:

Stecker Bezeichnung	Register ID
X1 – NO/NC 1	0x0000
X1 – NO/NC 2	0x0001
X1 – NO/NC 3	0x0002
X1 – NO/NC 4	0x0003
X1 – NO/NC 5	0x0004
X1 – NO/NC 6	0x0005
X2 – NO/NC 7	0x0006
X2 – NO/NC 8	0x0007
LED (rot)	0x0038

Regeln

4.1 Layout

Position	Länge	Attribut	Kommentar
0	1	numOfRules	Gesamtanzahl der in diesem Block enthaltenen Regeln
1	n	groupRule	1. Gruppenregel
1+n	m	groupRule	x-te Gruppenregel
1+n+m	1	endOfGroupRule	Terminierung des Regelblocks (ist immer 0)

Gruppenregel

Position	Länge	Attribut	Kommentar
0	n	rule	Regelelement
n	m	rule	x-tes Regelelement
n+m	1	endOfRule	Terminierung des Regelblocks (ist immer 0)

Regelelement

Position	Länge	Attribut	Kommentar
0	1	numOfConditions	Anzahl der Triggerbedingungen
1	1	numOfActions	Anzahl der Aktionen
2	1	triggerState	Benötigter Regelzustand für die Triggerbedingungen
3	1	nextState	Folgezustand der Regel bei erfüllter Bedingung
4	10	condition	1. Bedingung
14	10x	condition	x-te Bedingung
14+10x	10	action	1. Aktion
24+10x	10y	action	y-te Aktion

Bedingung

Position	Länge	Attribut	Kommentar
0	4	senderId	32-Bit Objekt-ID des Senders
4	6	data	Event-ID, Parameter1, …, Parameter5

Aktion

Position	Länge	Attribut	Kommentar
0	4	receiverId	32-Bit Objekt-ID des Empfängers
4	1	length	Tatsächliche Datenlänge
5	5	data	Funktions-ID, Parameter1, …, Parameter4

5 Firmware update

Um ein Firmware update durchführen zu können, muss der Bootloader aktiv sein.

Der Bootloader erbt von der Klasse Controller und unterstützt somit auch alle dort aufgeführten Befehle. Die ObjectId unterscheidet sich auch nur in der InstanceId (Firmware=1, Bootloader=2) von der der ObjectId der Firmware. Im Gegensatz zur Firmware, kann der Bootloader auch den Flashbereich beschreiben.

Der Bootloader ist der Teil der Firmware, der nach PowerOn zuerst gestartet wird. Er kann über die Schnittstellen I2C, UART, oder UDP erreicht werden. Wenn innerhalb von 3 Sekunden die Funktion PING über eine der Schnittstellen aufgerufen wird oder keine gültige Firmware geladen ist, bleibt der Bootloader aktiv. Läuft die Zeit ab ohne dass eine Funktion aufgerufen wird, startet die Firmware.

Wenn die Firmware bereits läuft, kann über folgende Sequenz in den Bootloader geschaltet werden:

1. Funktion RESET aufrufen
2. 1s warten
3. Funktion PING aufrufen, der Bootloader antwortet mit PONG. An der ObjectId der Antwort kann man nun erkennen, ob es sich um den Bootloader handelt. (0xXXXX0002)
4. Funktion GET_CONFIGURATION aufrufen um die Blockgröße der Daten zu ermitteln

Der Funktionsaufruf in Punkt 4 muss direkt auf dem Bootloader-Objekt erfolgen damit der Bootloader aktiv bleibt.

Die Firmware muss in einem Binär-Format vorliegen, dann kann der Inhalt mit dem Funktionsaufruf WRITE_MEMORY die Datei blockweise in das Flash ab Adresse 0x00000000 übertragen. Nach jedem Aufruf von WRITE_MEMORY muss der

MEMORY_STATUS abgewartet werden. Erst wenn dieser OK ist kann mit dem nächsten Block fortgesetzt werden. Im Fehlerfall kann versucht werden den letzten Block erneut zu übertragen. Nach erfolgreicher Übertragung kann der Controller mit dem Aufruf von RESET neu gestartet werden. Wenn der Bootloader dann nicht mehr angesprochen wird, sollte die neue Firmware starten.

5.1 Downgrade Schutz / Sperre

Um ungültige oder manipulierte Programme detektieren und abweisen zu können, sind im Bootloader folgende Mechanismen aktiviert:

5.1.1 Modul-Kennung

Jede FW enthält in den ersten Blöcken eine Modul-Kennung die folgende Informationen beinhaltet: Modul-Name, Größe des Moduls, Release-Kennung (Major.Minor), Firmware-ID. Diese Modulkennung ist über eine Checksumme geschützt, und wird vom Bootloader auf Echtheit überprüft. Kann die Echtheit nicht festgestellt werden, wird der Download-Prozess mit der Fehlermeldung `ABORTED` abgebrochen und die vorherige Firmware sofort wieder gestartet.

5.1.2 Regeln für erlaubte Downloads

Es kann generell nur eine Firmware nachgeladen werden, deren Modul-Kennung exakt dieselbe Firmware-ID und dieselbe Major-Release-Kennung verfügt. Die Minor-Release-Kennung muss größer sein, als die der bereits geladenen Firmware. Ist keine Firmware geladen, muss die Minor-Release-Kennung größer sein als die des Bootloaders.

5.2 Manipulationsschutz

Der Bootloader prüft bei jedem Start des Gerätes, ob die darin enthaltene Firmware noch gültig ist und nicht auf anderen Wegen manipuliert wurde. Erst wenn diese Prüfung erfolgreich ist, wird die FW gestartet.

6 Anschluss an weitere Automatisierungssysteme

ToDo

7 Konformitätserklärung

Der Hersteller,

Viktor Pankraz Elektronik
Lippstädter Weg 94
33758 Stukenbrock

erklärt in alleiniger Verantwortung, dass das Produkt,

Bezeichnung: LAN / RS485 Schaltaktor für Hutschienenmontage
Model: HBX-SW8-IN8-DR
Baujahr: 2020

allen einschlägigen Bestimmungen der Richtlinie 2014/35/EU - Niederspannungsrichtlinie entspricht.

Das Produkt entspricht weiterhin allen Bestimmungen der folgenden Richtlinien:

- Richtlinie 2014/30/EU über die elektromagnetische Verträglichkeit
- Richtlinie 2011/65/EU RoHS

Folgende harmonisierte Normen wurden angewandt:

Norm	**Titel**
EN 60669-1	Schalter für Haushalt und ähnliche ortsfeste elektrische Installationen Teil 1
EN 60669-2	Schalter für Haushalt und ähnliche ortsfeste elektrische Installationen Teil 2
EN 55014-1	Anforderungen an Elektrogeräte – Teil 1: Störaussendung
EN 61000-3-2	Emissionsgrenzwerte für Geräte bis 16A Nennstrom
EN 61000-3-3	Elektromagnetische Verträglichkeit (EMV) – Anforderungen – Teil 3

Folgende sonstige technische Normen und Spezifikationen wurden angewandt:

- ..Keine

Für die Zusammenstellung der technischen Unterlagen ist beauftragt:

Herr Viktor Pankraz
Lippstädter Weg 94
33758 Stukenbrock

Unterzeichner und Angaben zum Unterzeichner: Herr Viktor Pankraz (Geschäftsführer)

Ort, Datum: Stukenbrock 25.05.2020 Unterschrift: V. Pankraz