



Application Manual

 **Cumulocity**
Cloud Fieldbus



PSsystec GmbH • Heinbergstr. 27
86 356 Neusäß • www.pssystec-gmbh.de

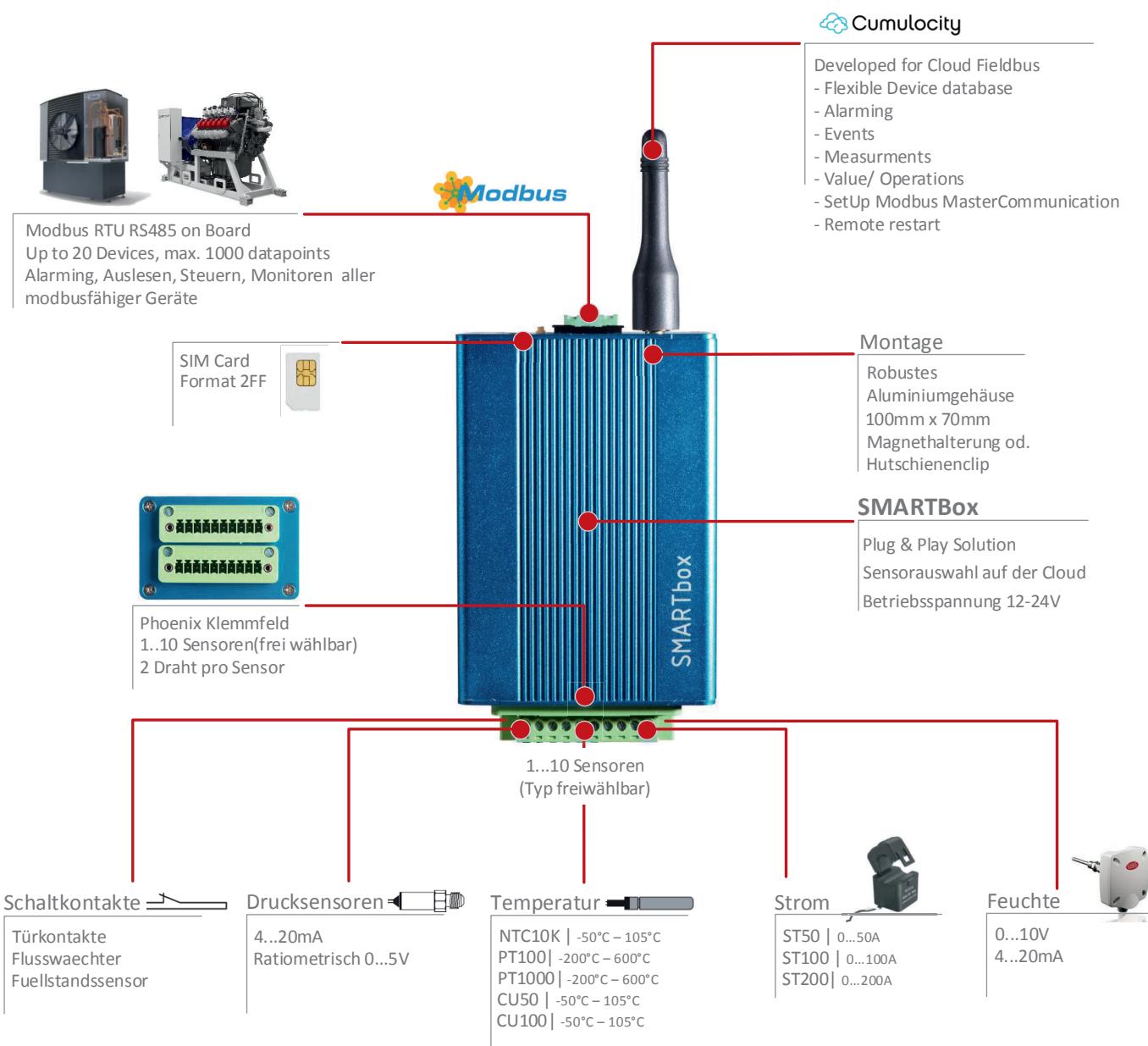
PSsystec SMARTbox Agent User's Guide

Overview

Smartbox Modbus, based on the Telit Chipset HE910 is a ready to use solution for connecting Sensors and in parallel Modbus devices to the Cumulocity. It provides a Master Slave Communication on RS485 for connecting up to 20 devices. Easy configure the configuration of building automation fielddevices like pumps, chillers, E-meters, Airhandling units in Cumulocity – the Smartbox Modbus will take care of it!

By automatically picking up the coils and registers and sending alarms, measurements and status back to Cumulocity.

Pssystec GmbH is specialized in building automation. With SMARTbox Modbus we developed a slim Line M2M device, fully integrated in the HVAC market.

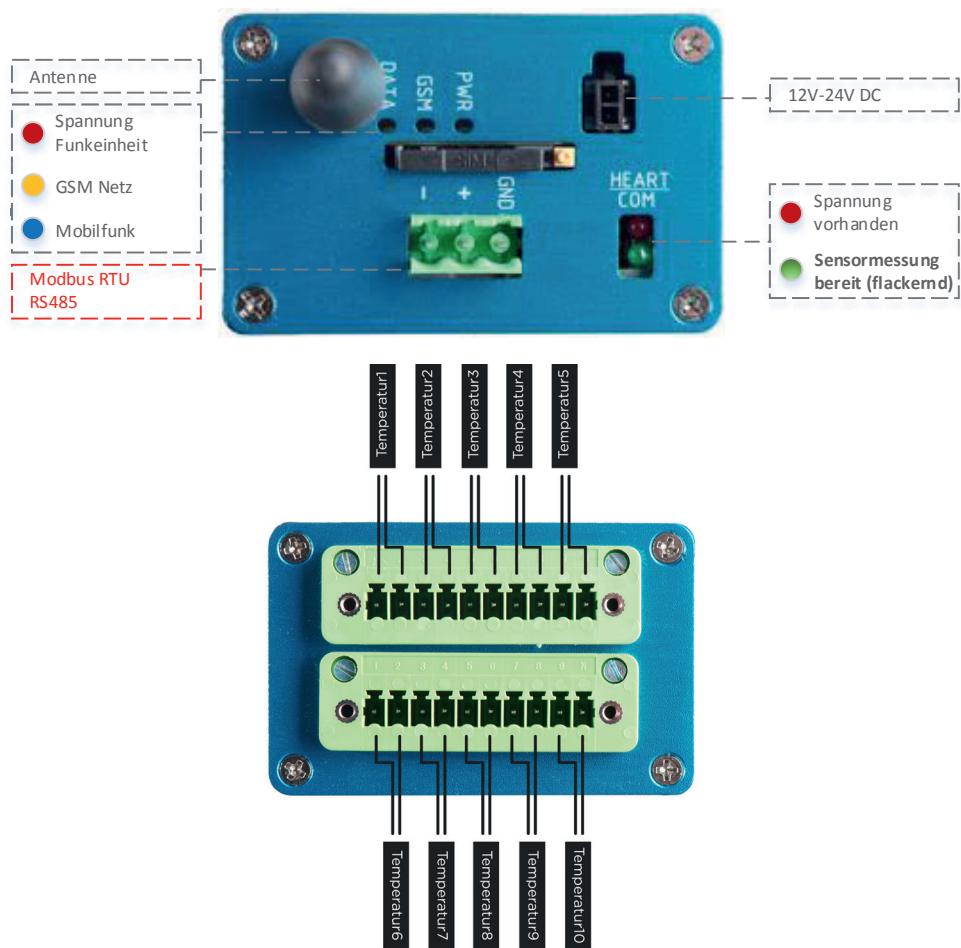


The Smartbox comes up with different hardware configurations. If you don't find your configuration, get in contact with us. We have a flexible IO system which allows to change the IO configuration to your needs.

	Smartbox	HW Typ A	HW Typ B	HW Typ C	HW Typ D	HW Typ E	HW Typ F
	Max. No Sensors	10	4	10	10	10	10
Sensors	NTC	10	-	10	-	-	-
	ON/OFF (voltage free)	10	-	10	-	8	-
	PT100	-	4	10	-	-	-
	PT1000	-	4	10	-	-	-
	Cu50	-	4	-	-	-	-
	Cu100	-	4	-	-	-	-
	4..20mA	-	-	-	10	-	-
	0..10V	-	-	6	-	-	10
Fieldbus	0..5V	10	-	6	-	-	-
Out	Relais Output	-	-	1	-	-	-
Fieldbus	Modbus RS485	Yes	Yes	Yes	Yes	Yes	Yes

Wire your Sensors and Modbus RTU RS485 Network

The Hardware interface is as follows:



Configure the terminal

By default the terminal supports Cloud Fieldbus from cumulocity. To do so, you need to:

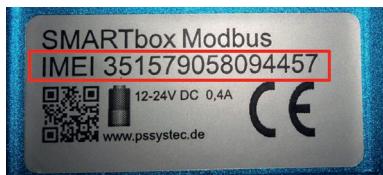
- Subscribe your account to the Cloud Fieldbus app by contacting support@pssystec-gmbh.de or support@cumulocity.com
- To configure your APN put your SIM CARD to the Smartbox. After this, power On the Smartbox Modbus , wait > 10 seconds – and send an SMS to the phone number, given by the provider from your SIM CARD.

Change APN	Send SMS with APN, > 10s after Power on GPRS=public4.m2minternet.com
------------	---

From this point, the Smartbox stores your set APN.

Connecting the terminal

After setting APN, register your SMARTbox Modbus to Cumulocity. „Connecting devices“: Type the IMEI of the terminal in the registration tab of Cloud Fieldbus and press register. The IMEI is printed on the back side of the terminal as shown in the screenshot below.



After clicking the „accept“ button, navigate to „All devices“, the terminal should appear after ≈ 30 seconds after registration.

The screenshot shows the 'FIELDDBUS3' software interface. The left sidebar has a dark theme with white text. The 'Alle Geräte' (All Devices) tab is highlighted with a blue arrow. The main area lists several devices with their names and status indicators. One device, 'Terminal 351579058095389', is highlighted with a green border.

Device	Status
Terminal 351579058094580	1 ⓘ
Terminal 359785020094887	1 ⓘ
Terminal 359785020093962	1 ⓘ
Datenlogger von Klama & more GmbH &Co.KG	1 ⓘ
Terminal 351579058094333	1 ⓘ
Terminal 351579058095389	1 ⓘ
Terminal 351579058095264	1 ⓘ
Weidner Holding GmbH & Co.KG	1 ⓘ
Terminal 359785029009738	1 ⓘ
Terminal 351579058094341	1 ⓘ
Terminal 351579058094184	1 ⓘ

After this step:

Configure the sensors

Depending on the Typ (A-F) you have to do following steps:

1. Switch to the tab “Device database” in fieldbus3. And put the data to the device:
Add the device Smartbox A:

The screenshot shows the 'FIELDDBUS3' software interface. The left sidebar has a dark theme with white text. The 'Device Database' tab is highlighted with a green border. The main area lists several devices. A green box highlights the 'Add device type' button in the top right corner.

Device
ModDevtest
Bigdevice3
Bigdevice1
BigDeviceModbus
Powergen
Datenlogger
tec301
Bigdevice
Bigdevice2

2. Add a register and put the data from list above to it. E.g. for Sensor 1 and repeat this for all sensors.

For TypA the list is as follows:

Smartbox HW A

Communication parameter	Baudrate	19200
	Datenbits	8
	Parity	None
	Stopbits	1
	Adress	1

Device Database (Register)	Description	Decimal Places	Measurement
101	Actual value Sensor 1	1	yes
102	Actual value Sensor 2	1	yes
103	Actual value Sensor 3	1	yes
104	Actual value Sensor 4	1	yes
105	Actual value Sensor 5	1	yes
106	Actual value Sensor 6	1	yes
107	Actual value Sensor 7	1	yes
108	Actual value Sensor 8	1	yes
109	Actual value Sensor 9	1	yes
110	Actual value Sensor 10	1	yes

Device Database (Register)	Description	Enumeration Type	Update status
114	Configuration Sensor 1	6 - DIN, 7 = NTC	yes
115	Configuration Sensor 2	6 - DIN, 7 = NTC	yes
116	Configuration Sensor 3	6 - DIN, 7 = NTC	yes
117	Configuration Sensor 4	6 - DIN, 7 = NTC	yes
118	Configuration Sensor 5	6 - DIN, 7 = NTC	yes
119	Configuration Sensor 6	6 - DIN, 7 = NTC	yes
120	Configuration Sensor 7	6 - DIN, 7 = NTC	yes
121	Configuration Sensor 8	6 - DIN, 7 = NTC	yes
122	Configuration Sensor 9	6 - DIN, 7 = NTC	yes
123	Configuration Sensor 10	6 - DIN, 7 = NTC	yes

In future you will get an procedure for automatic download the configuration to your device database in your tenant. Up to now you have to do it manually..but only once a time, then you can use this for all smartboxes TypA in your tenant.

Example for setup Sensor1 – Measurement data

The image displays two side-by-side screenshots of the Smartbox TypA configuration software interface, specifically for setting up a sensor.

Left Screenshot (Sensor Configuration):

- NAME:** ConfigurationSensor1
- DISPLAY CATEGORY:** Address
- NUMBER:** Address (114), Start Bit (0), Number of Bits (16)
- MULTIPLIER:** 1
- DIVISOR:** 1
- DECIMAL PLACES:** 0
- UNIT:** (empty)
- MINIMUM:** 6
- MAXIMUM:** Decimal Places 0
- OPTIONS:**
 - Signed
 - Unsigned
- ENUMERATION VALUES:**
 - 6 (Off_On)
 - 7 (NTC)
- FUNCTIONALITIES:**
 - Show status (read-only access)
 - Update status (read/write access)
 - Send measurement
 - Raise alarm
 - Send event

Buttons at the bottom: OK, CLOSE

Right Screenshot (Measurement Data Setup):

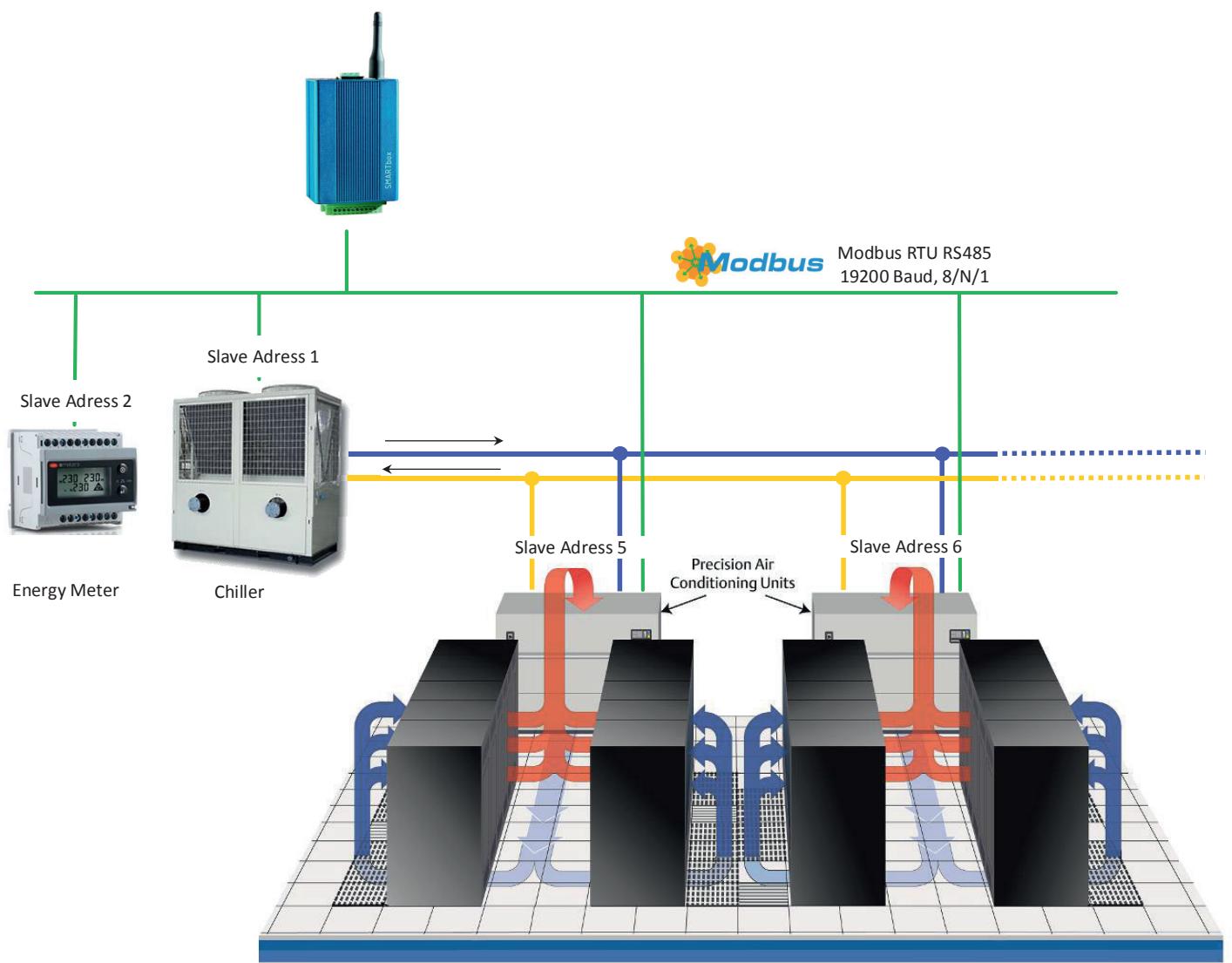
- NAME:** Sensor1
- DISPLAY CATEGORY:** Address
- NUMBER:** Address (101), Start Bit (0), Number of Bits (16)
- MULTIPLIER:** 1
- DIVISOR:** 1
- DECIMAL PLACES:** 1
- UNIT:** Type the unit e.g. °C
- OPTIONS:**
 - Signed
 - Unsigned
 - Enumeration type
- FUNCTIONALITIES:**
 - Show status (read-only access)
 - Update status (read/write access)
 - Send measurement → Set Send Measurement
 - Raise alarm
 - Send event
- MEASUREMENT TYPE:** Sensor1
- MEASUREMENT SERIES:** Sensor1

Buttons at the bottom: OK, CLOSE

Now you made the Setup for the Sensors in Smartbox TypA. In the next chapter you setup the Modbus network. You need to add this device to the Smartbox you are using.

Configure the Modbus network (optional)

Lets make an example: Assume you have a data center application: A chiller provides constant cold water of 7°C at the outlet. For each Serverrack line, a precision airconditioner is installed which maintain the racktemperature to 20°C by blowing cool Air through the floor grid to the racks. The warm air at the outlet of each rack will be again cooled down by a heat exchanger, installed in the air conditioning units, feeded from cool water coming from the chiller. Your company servicing the cooling system for your customer.



To connect this datacenter application to Cumulocity follow these steps. Be aware, that due to the sensors following is fixed:

Adress1 is reserved for the sensors

Communication frame: 19200/8N1

1. The Smartbox, acts as a Modbus Master. Connect all slaves together in one line and put different Slave addresses in the fielddevices, as well as a common Baudrate and Communication frame (e.g. 8/N/1). Normally all fielddevices provides such setting at a local display.
2. For each different Fielddevice (Chiller, AirConditioning units, Energy meter) create the Device database in Cloud Fieldbus in Cumulocity (see the Appendix “Create” Device Database).
3. SetUp in Modbus configuration Tab
 - A. Set Field Communication Parameter
 - B. Set Communication parameter between Smartbox and Server. Note Events, Alarms, Values (See device database) are transmitted to server if the read out value from Modbus is changed. So only the transmittate for measurements is defined.
 - C. BuildUp the network by adding slave devices defined in the device database. In Cumulocity, each Fielddevice come up with a single childdevice. So in this cases we would have 3 childdevices. After adding the slave devices, the terminal will discover all childdevices and its Modbus items and begin feeding the platform with measurements, alarms, events, values. Note: Without a setup network, the terminal will not send any data.

ModbusBaudrate Select

- 3200
- 9600
- 19200
- 38400* (coming soon)

Modbus Communication Frame:

- Databits: 8 or 7
- Parity: None, Even, Odd
- Stopbits: 1 or 2

TERMINAL 351579058055433

SERIELLE KOMMUNIKATION

BAUDRATE: 19200

DATENBITS: 8

PARITÄT: N

STOP-BITS: 1

MODBUS-KOMMUNIKATION

ÜBERTRAGUNGSRATE: 20 Sekunden

POLLING-INTERVALL: Not used but set 1 or any value

PROTOKOLL: RTU

Transmissionrate Terminal to Cumulocity Cloud
Measurements: define the cycle time here
Events: if value read out changes
Alarms: if value read out changes
Value: if value read out changes

Pressing this button will save the values on the terminal, you can do this during runtime

Anderungen speichern

Add a slave device

Neues Gerät hinzufügen

Name	Gerätetyp	Adresse
EnergyMeter	Powergen	2
AirConditioning	Datenlogger	5
Chiller	tec301	1

Custom name can be defined

Device from device database

Slave address

The default name of the terminal is the IMEI. Click on the terminal to view the detailed information. You can change the terminal's name on the „Info“ tab, which also displays basic information such as serial number of the router and SIM card data. After changing the name, remember to click „save changes“ button at the bottom of the „Info“ page. All data coming from the fielddevices are available under the section childdevices.

Manage the Application

- General: Managing the sensors or Modbus items form any field devices is at least the same. Only in case of sensors you will not have events or alarms, you will need only the measurement tab in the child device to browse the values and fieldbus- widget to set the type of sensor according to the device model from Type A for example.

Manage: In order to monitor the Measurements, events, alarms from the fielddevices, go to the child device tab. Inside each childdevice tab all relevant data are available according to the specific device and the set configuration in the device database.

To see and manage the values which are defined in Device database as ReadStatus for Read values or Update Status for Read/Write Values add a tab including the fieldbus widget (which comes up with the fieldbus3 application). Here you can set and read out and write these defined values.

For the sensors connected directly to the Smartbox, you will need:

- a) The measurement tab for browsing the sensor values
- b) The fieldbus widget to set the configuration of the sensors

2. Sending Alarm, Events, Measurements, Values: The Send cycle is as follows:

Send cycle	Values (Readstatus)	On Change	Add Fieldbus-widget
	Values (Updatestatus)	On Change	Add Fieldbus-widget
	Alarms	On Change	Found in Alarntab Childevice
	Events	On Change	Found in Events Childevice
	Measurements	Defined in Modbus tab	Found in Measurements Childevice
	Signal strength	Is sent every 10 Min as an measurement	Found in Measurements

3. Monitor the success of operations: All operations send from server to the samrtbox can be monitored if they are successful processed. Operations are e.g. Changing the transmit rate, changing Modbus communication frame, adding new slave devices, remote restart, changing values on the Modbus (Updatestatus)

- Restart the terminal: The terminal can be restarted either by power On/off or Remote Restart by Cumulocity Device Control, on terminal level
- Location: The terminal features cell Location and is available in Location tab on terminal level.

The screenshot shows the Cumulocity interface for a device named 'FIELDBUS3'. The left sidebar has a 'Startseite' tab selected. Under 'MESSWERTE' (Measurements), the 'Standort' (Location) tab is selected. The main content area shows a map of the Augsburg region with a blue marker indicating the device's location. To the right of the map are input fields for 'ADRESSE', 'BREITENGRAD' (Latitude) set to 48.356011, 'LÄNGENGRAD' (Longitude) set to 10.957442, and 'HÖHE' (Height). A 'Speichern' (Save) button is at the bottom. Below the map, there is a note about dragging the map to zoom and a link to Leaflet and OpenStreetMap contributors.

- Signal strength: The signal strength of the terminal is sent each 10minutes and is displayed as a measurement on terminal level.

The screenshot shows the Cumulocity interface for the same device 'FIELDBUS3'. The left sidebar has a 'MESSWERTE' (Measurements) tab selected. The main content area shows a line graph titled 'SIGNAL STRENGTH' with the Y-axis labeled 'Signal Strength >= null [dBm]' ranging from 0 to -100. The X-axis shows dates from Sat 23 to Sun 24. The graph shows a green line with data points, generally staying between -100 and -90 dBm. There are dropdown menus for 'Keine Aggregation' (No aggregation), 'Letzter Tag' (Last day), and 'Automatische Aktualisierung' (Automatic update).

- Indication and Software Version: Go to Indication and Software tab to identify the terminals IMEI, Version and FW installed.

The screenshot shows the Cumulocity interface for the device 'FIELDBUS3'. The left sidebar has a 'SOFTWARE' tab selected. The main content area shows two sections: 'FIRMWARE' and 'SOFTWARE'. Under 'FIRMWARE', it says 'Aktuelle Firmware HE910 - 12.00.006'. Under 'SOFTWARE', there is a table with columns 'Name' and 'Version'. One entry is 'appZone' with 'Version' 0.9.5.

The screenshot shows a device registration interface. On the left, a sidebar menu includes 'Startseite', 'GERÄTE' (selected), 'ÜBERSICHTEN', and 'Alarme'. The main area displays a single device entry for 'c8y_IMEI' with ID '351579058094739'. A toolbar at the top right includes search and filter icons.

8. Real time Clock: The terminal has an Real Time clock installed. Each time a measurement, event or alarm is sent to Cumulocity, the time is retrieved from the internal real time clock and gets the GMT time from a NIST server. This means, you don't need to use a SIM Card with the NITZ feature.
9. Buffering data: In case of bad connection, measurements, events, alarms are permanently stored in an internal buffer. About 24h can be stored in case of bad communication. Getting online again the terminal sends the data to Cumulocity.

Appendix: “Create Device database”

1. Select the tab Device database and add device type

The screenshot shows a device database interface. The sidebar menu includes 'Startseite', 'GERÄTE' (selected), 'ÜBERSICHTEN', 'Alarme', 'Kommandos', 'Ereignisse', and 'DEVICE TYPES' (selected). The main area lists device types: ModDevtest, Bigdevice3, Bigdevice1, BigDeviceModbus, Powergen, Datenlogger, tec301, Bigdevice, and Bigdevice2. A green box highlights the 'Add device type' button in the top right corner.

General: The SmartboxModbus supports following Function Codes:

- FC1 Read Coils (Address Cumulocity == AddressDevice +1)
- FC2 Read Input Status (Address Cumulocity == AddressDevice +10001)
- FC3 Read Register (Address Cumulocity == AddressDevice +1)
- FC4 Read Input Register (Address Cumulocity == AddressDevice +10001)
- FC5 Write Coils (Address Cumulocity == AddressDevice +1)
- FC6 Write Register (Address Cumulocity == AddressDevice +1)

Taking the Modbus Datapoint list from your device you want to connect to the smartbox, you can now define all registers and coils in the Device database by adding register or adding coils. Note: The limitation is 100 datapoints if you have 10 slave devices or 1000 datapoints if you have only 1 device you want to connect.

2. Add Registers

NEW REGISTER										
Name of datapoint displayed in CC Datapoint Tab (used in Fieldbus Widget)	<input type="text" value="Tset"/>									
Address of Register in the Slave (+1 or +10001, according to FC3 or FC4)	<input type="text" value="234"/>									
Scaling of Value: Value in CC == Value * Multiplier / Divisor * 10 ^{-DecimalPlaces}	<input type="text" value="1"/> <input type="text" value="1"/> <input type="text" value="2"/>									
Unit displayed in CC Datapoint Tab (used in Fieldbus Widget)	<input type="text" value="°C"/>									
Signed Value = the value has an + or -	<input checked="" type="checkbox"/> Signed									
Needed, if Value has Enum Values like On, Off, Defrost (used in Fieldbus Widget)	<input type="checkbox"/> Enumeration type									
Definition of enum Values (used in Fieldbus Widget)	<table border="1"> <tr> <td><input type="text" value="0"/> 0</td> <td><input type="text" value="On"/></td> <td><input type="button" value="Remove value"/></td> </tr> <tr> <td><input type="text" value="1"/> 1</td> <td><input type="text" value="Off"/></td> <td><input type="button" value="Remove value"/></td> </tr> <tr> <td colspan="3">+ Add value</td> </tr> </table>	<input type="text" value="0"/> 0	<input type="text" value="On"/>	<input type="button" value="Remove value"/>	<input type="text" value="1"/> 1	<input type="text" value="Off"/>	<input type="button" value="Remove value"/>	+ Add value		
<input type="text" value="0"/> 0	<input type="text" value="On"/>	<input type="button" value="Remove value"/>								
<input type="text" value="1"/> 1	<input type="text" value="Off"/>	<input type="button" value="Remove value"/>								
+ Add value										
Readout the Value without Timestamp, (Display in FieldbusWidget)	<input checked="" type="checkbox"/> Show status (read-only access)									
Readout the Value withTimestamp , Display Graph in Measurements of Childdevice (ModbusSlave)	<input checked="" type="checkbox"/> Update status (read/write access)									
Event with Timestamp, (Display the Measured Value in Events of Childdevice)	<input checked="" type="checkbox"/> Send measurement <input type="checkbox"/> Raise alarm <input checked="" type="checkbox"/> Send event									
Defining Events: Event Type: Possibility to group, But you can take any name e.g. same as „Name“ Event Text: Displayed text in Event tab folder	<table border="1"> <tr> <td><input type="text" value="Measurement Type"/></td> <td><input type="text" value="Measurement Series"/></td> </tr> <tr> <td><input type="text" value="Alarm Type"/></td> <td><input type="text" value="Alarm Text"/></td> </tr> <tr> <td><input type="text" value="Alarm Severity"/></td> <td><input checked="" type="checkbox"/></td> </tr> <tr> <td><input type="text" value="Event Type"/></td> <td><input type="text" value="Event Text"/></td> </tr> </table>	<input type="text" value="Measurement Type"/>	<input type="text" value="Measurement Series"/>	<input type="text" value="Alarm Type"/>	<input type="text" value="Alarm Text"/>	<input type="text" value="Alarm Severity"/>	<input checked="" type="checkbox"/>	<input type="text" value="Event Type"/>	<input type="text" value="Event Text"/>	
<input type="text" value="Measurement Type"/>	<input type="text" value="Measurement Series"/>									
<input type="text" value="Alarm Type"/>	<input type="text" value="Alarm Text"/>									
<input type="text" value="Alarm Severity"/>	<input checked="" type="checkbox"/>									
<input type="text" value="Event Type"/>	<input type="text" value="Event Text"/>									
DisplayCategory, Displayed in FieldbusWidget Defines which bits of the 16bit register assigned to this datapoint Startbit = Starting Bit Number of Bits = Number of bits used of the 16Bit Register										
Read/Write Value without Timestamp, (Display in FieldbusWidget) Alarm with Timestamp, (Display in Alarms of Childdevice) 0 = no Alarm, 1 = Alarm Note: This Alarm can only be published from the device to CC. it must be acknowledged in CC. The device send no „clear“ to the Alarm in CC, even if the alarm is not present anymore on the device										
Defining Measurements: Series: Name of Measured Value e.g. same as under „Name“ (Display in Tab Measurements) Type: Possibility to group measurements. But you can take any name e.g. same as „Name“										
Defining Alarms: Alarm Type: Possibility to group, But you can take any name e.g. same as „Name“ Alarm Text: Displayed text in Alarm tab folder Alarm Severity: Displayed Severity in Alarm tab folder										

3. Add Coil

NEW COIL

NAME	DISPLAY CATEGORY
MotorStatus	Statusinformation DisplayCategory, Displayed in FieldbusWidget
NUMBER	
2	Read/Write Value without Timestamp, (Display in FieldbusWidget)
FUNCTIONALITIES	Alarm with Timestamp, (Display in Alarms of Childdevice) 0 = no Alarm, 1 = Alarm Note: This Alarm can only be published from the device to CC. it must be acknowledged in CC. The device send no „clear“ to the Alarm in CC, even if the alarm is not present anymore on the device
<input checked="" type="checkbox"/> Show status [read-only access] <input checked="" type="checkbox"/> Update status [read/write access] <input checked="" type="checkbox"/> Raise alarm <input checked="" type="checkbox"/> Send event	
TEXT TO SHOW IF COIL IS ZERO	TEXT TO SHOW IF COIL IS ONE
Off	On
ALARM TYPE	ALARM TEXT
ALARM SEVERITY	
	Defining Alarms: Alarm Type: Possibility to group, But you can take any name e.g. same as „Name“ Alarm Text: Displayed text in Alarm tab folder Alarm Severity: Displayed Severity in Alarm tab folder
EVENT TYPE	EVENT TEXT
Defining Events: Event Type: Possibility to group, But you can take any name e.g. same as „Name“ Event Text: Displayed text in Event tab folder	
OK CLOSE	



PSsystec GmbH • Heinbergstr. 27
86 356 Neusäß • www.pssystec-gmbh.de