

OWOS
Dokumentation

Stand: 01/2013

Version 8.3

1.	Rechtsgrundlagen	1
1.1	Haftungsausschluss.....	1
1.2	Eingetragene Markenzeichen	1
2.	Einleitung.....	1
2.1	Ziele	1
2.2	Globale Konstanten	2
3.	Datentypen der OWOS Bibliothek	3
3.1	owp (one-wire-Parameter)	3
3.2	i_Button_key_V8.....	4
3.3	DS1820_V8	4
3.4	DS2438_V8	4
3.5	DS2450_V8	4
3.6	one_wire_binaer_V8.....	5
3.7	DS2408_V8	5
3.8	DS2423_V8	6
3.9	LCD_Text_163.....	6
3.10	LCD_Text_4_20.....	6
4.	Bausteine	7
4.1	one_wire	7
4.2	ow_temperatur.....	9
4.3	i_Button_suchen	10
4.4	ow_temperatur_i_button	11
5.	Anwendungshinweise.....	12
5.1	Daten eintragen	12
5.2	Zugriff auf Daten	13
5.3	ID Suchen	14
5.4	Bausteineinstellungen.....	16
5.4.1	Aufrufintervall „T_a“	16
5.5	Verwendung von mehreren Schnittstellen	17
6.	LCD	18
6.1	Ansteuerung	18
6.1.1	LCD_Tool_ow.....	18
6.1.2	LCD_Tool_ow_2.....	19
6.1.3	LCD_tool_163	19
6.1.4	LCD_tool_163_2	20
6.1.5	LCD an Digitalausgängen (LCD_tool_DO).....	20
6.2	LCD Texte.....	22
6.2.1	Texte_20_10_1_1	22

Inhalt

6.2.2	Texte_20_10_4_1	23
6.2.3	Texte_16_10_3_1	24
6.2.4	Texte_16_10_3_1_nummer	25
6.2.5	LCD_Texte_10_nummer	25
7.	Tools.....	26
7.1	Luftfeuchte Luftdruck Luftgüte (ow_feuchte_druck)	26
7.2	3-Fach Analog Input (ow_a_i).....	27
7.3	Messwerte vom 1-Wire Hub (ow_hub).....	28
8.	Hardware.....	29
8.1	Anschluss an die interne Serviceschnittstelle	29
8.2	Verwendung einer seriellen Klemme 750-650	30
8.2.1	Einstellung der seriellen Klemme 750-650/003-000	31

1. Rechtsgrundlagen

1.1 Haftungsausschluss

Die in der OWOS Bibliothek enthaltenen Softwarebausteine werden als Entwicklungsvorlage angeboten. Es wird keinerlei Gewährleistung für die Richtigkeit und Fehlerfreiheit übernommen. Der dargestellte Programmcode ist nicht für Produktivzwecke vorgesehen und dient nur als Anwendungsbeispiel. Der Nutzer ist selbst verantwortlich dafür, eventuell in seiner Anwendung verwendete Module mit geeigneten Verfahren zu testen und die notwendige Fehlerfreiheit, Qualität und Funktionalität zu verifizieren.

Eine Funktionsgarantie wird nicht übernommen und wird explizit ausgeschlossen.

Da die in der Bibliothek enthaltenen Softwaremodule ohne jegliche Kosten bereitgestellt werden, besteht keinerlei Gewährleistung.

1.2 Eingetragene Markenzeichen

Alle aufgeführten Bezeichnungen, Logos, Namen und Warenzeichen (auch solche, die nicht explizit gekennzeichnet sind) sind Warenzeichen, eingetragene Warenzeichen oder sonstige urheberrechtlich oder Marken- bzw. Titelrechtlich geschützte Bezeichnungen ihrer jeweiligen Eigentümer und werden von uns als solche ausdrücklich anerkannt. Die Nennung dieser Bezeichnungen, Logos, Namen und Warenzeichen geschieht lediglich zu Identifikationszwecken und stellt keinen irgendwie gearteten Anspruch von E-Service-Online, Andreas Geisler an bzw. auf diese Bezeichnungen, Logos, Namen und Warenzeichen dar. Zudem kann aus dem Erscheinen auf diesen WWW-Seiten nicht darauf geschlossen werden, dass Bezeichnungen, Logos oder Namen frei von gewerblichen Schutzrechten sind.

2. Einleitung

2.1 Ziele

OWOS steht für „One Wire Open Source“.

OWOS erstellt eine Open Source Bibliothek zur Verfügung, mit welcher eine Kommunikation zwischen einem WAGO-Controller (getestet mit 750-841 und 750-881) und dem 1-Wire-Bus der Firma DALLAS MAXIM aufgebaut werden kann.

Die Kommunikation erfolgt über eine serielle Schnittstelle, welche

1. über die Serviceschnittstelle des Controllers (COM1, TTL-Pegel),
2. über eine serielle Klemme 750-650 (COM >1, RS232-Pegel)

aufgebaut werden kann, wobei als Gateway der DS2480B verwendet wird.

Folgende 1-Wire Bausteine werden unterstützt:

Bezeichnung	Funktion	Sensor / Bausteingruppe	Datentyp
DS18B20 (DS18S20)	Temperatursensor	DS1820	DS1820_V8
DS2438	AD-Wandler	DS2438	DS2438_V8
DS2450	AD-Wandler	DS2450	DS2450_V8
DS2405	1 Port PIO	one_wire_binaer	one_wire_binaer_V8
DS2406	1 (2) Port PIO		
DS2413	2 Port PIO		
DS2408	8 Port PIO	DS2408	DS2408_V8
DS2423	2 Kanal Zähler	DS2423	DS2423_V8
iButton			i_button_key_V8

Weiterhin werden folgende Bausteine bereitgestellt:

- `anwesenheit_iButton` hiermit können iButtons am 1-Wire Netz abgefragt werden
- `LCD` ermöglicht die Ansteuerung eines LCD-Displays am 1-wire Bus unter Verwendung des DS2408 oder an SPS-Digitalausgängen
- `Tools` Bausteine für die Messung von Spannung / Strom / Luftfeuchte / Luftdruck / Luftgüte mit Modulen von <http://www.eservice-online.de>

Um eine ordnungsgemäße Funktion zu gewährleisten werden folgende Bibliotheken benötigt:

- `serial_interface_1.lib` (wird von der Fa. Wago bereitgestellt)
- `SysLibRtc.lib` (wird von der Fa. Wago bereitgestellt)
- `oscat_basic.lib` Version 3.33 (<http://oscat.de>)

2.2 Globale Konstanten

Die OWOS Bibliothek benötigt einige globale Konstanten um die Bibliothek an die Anwendungen anzupassen.

`mem_sendezeichen`:

`mem_sendezeichen` definiert die Größe des Sendezeichenpuffers (COM)

`mem_empfang`:

`mem_empfang` definiert die Größe des Empfangsspeichers (COM)

`mem_suchen`:

`mem_suchen` definiert die maximale Anzahl an one-wire Bausteinen, welche gesucht werden können [default=100]

`Anzahl_Fehlermeldungen`:

`Anzahl_Fehlermeldungen` definiert die Anzahl Fehlermeldungen, welche unter „owp.Fehlermeldung“ abgelegt werden können.

`ow`:

`ow` definiert verwendete Anweisungen und Parameter für die one-wire-Kommunikation welche im **TYP `ow_Anweisungen_Parameter`** definiert sind.

3. Datentypen der OWOS Bibliothek

Die OWOS Bibliothek definiert neben den Standard Datentypen weitere Datentypen. Diese werden innerhalb der Bibliothek verwendet und können jederzeit von Anwender für eigene Deklarationen verwendet werden.

3.1 owp (one-wire-Parameter)

Diese Parameter dienen zur Steuerung und Überwachung des one-wire Bausteins. Hier können diverse Einstellungen vorgenommen werden.

Parameter		Beschreibung
one_wire_Reset		TRUE-Signal Initialisierung, Signal wird automatisch abgeschaltet
Statusmeldung		Textmeldung über Status bzw. Störungen
Info		hier wird die Restzeit ("T_Sensor" - abgelaufene Zeit), bis zur nächsten Messung, eingetragen [in Minuten]
sonstige_werte	Stoerungen_Reset	Bausteinstörungen zurücksetzen
	DS1820_neu	neue Messwerte können gelesen werden
	DS2438_neu	
	DS2450_neu	
	werte_ow_io	
ID_suchen	neue_ID_suchen	Startsignal zur ID-Suche, wird automatisch abgeschaltet
	was_suchen	Typen auswählen, welche gesucht und eingetragen werden sollen
	anzahl_gefunden	hier wird automatisch die gefundene Anzahl je Typ eingetragen
gefundener_typ		Hier wird eingetragen, welche Bausteine gefunden wurden
Fehlerbehandlung	Sensoren_Fehleranzahl	one-wire-Baustein wird bei Störungen >Fehleranzahl als fehlerhaft gekennzeichnet
	Binaerbaustein_Fehleranzahl	
Fehlermeldung	Anzahl_Fehler	Anzahl der aufgetretenen Fehlermeldungen
	Fehler_loeschen	TRUE-Signal löscht alle Fehlermeldungen, Signal wird abgeschaltet
	Fehlermeldungen	Fehlermeldungen mit Zeitstempel (Array [1..Anzahl_Fehlermeldungen])

DS1820_neu, DS2438_neu, DS2450_neu

Mit diesem Signal kann eine ereignisgesteuerte Task zum Lesen der Messwerte gesteuert werden.

Statusmeldung

Der Text ist selbsterklärend und beschreibt, was gerade passiert bzw. was gerade erkannt wurde. Ziel der Statusmeldungen ist es Störungs- und Fehlerursachen zu erkennen.

Fehlerbehandlung

Bei der Datenübertragung wird, sofern der Sensor / Baustein dies unterstützt, eine CRC-Prüfung durchgeführt. Die Anzahl der aufgetretenen Fehler wird in den Daten des jeweiligen Bausteins abgelegt.

Mit dem Parameter Fehlerbehandlung kann festgelegt werden, ab welcher Anzahl Störungen der Baustein als fehlerhaft gekennzeichnet wird. (Fehler=TRUE)

Fehlermeldungen

Aufgetretene Fehler werden mit einem Zeitstempel versehen in ein Array nach dem FIFO-Prinzip folgendermaßen abgelegt.

.Zeit	Zeitstempel
.Ursache	Fehlerbeschreibung

Die Größe des Fehlerspeichers wird mit der Globalen Variable [Anzahl Fehlermeldungen](#) eingestellt.

3.2 i_Button_key_V8

Der Datentyp „i_Button_key_V8“ definiert und speichert Vorgaben und Status für i_Button.

*.Name: STRING	Name des keys (optional)
*.ID: STRING(16)	8-stellige ID des Sensors
*.Aktiv:BOOL	iButton angeschlossen (TRUE-aktiv)

3.3 DS1820_V8

Der Datentyp „DS1820_V8“ definiert und speichert Vorgaben Werte und Status für die Temperatursensoren 18S20 und 18B20.

*.Name: STRING	Name des Sensors (optional)
*.ID: STRING(16)	8-stellige ID des Sensors
*.Temperatur: REAL	Messergebnis
*.Stoerungen:INT	Anzahl der Sensorstörungen
*.Fehler:BOOL	es sind mehr als unter Sensoren Fehleranzahl definierte Störungen aufgetreten
*.aktiv :BOOL	Sensor wird bearbeitet (TRUE-aktiv)

3.4 DS2438_V8

Der Datentyp „DS2438_V8“ definiert und speichert Vorgaben Werte und Status für den AD-Wandler DS2438.

*.Name: STRING	Name des Sensors (optional)
*.ID: STRING(16)	8-stellige ID des Sensors
*.Temperatur: REAL	Messergebnis
*.VAD:REAL	Analogeingangsspannung
*.VDD:REAL	Busspannung [normal 5V]
*.XSSENS:REAL	Spannungssignal 0-250mV vom Stromeingang
*.Stoerungen:INT	Anzahl der Sensorstörungen
*.Fehler:BOOL	es sind mehr als unter Sensoren Fehleranzahl definierte Störungen aufgetreten
*.aktiv:BOOL	Sensor wird bearbeitet (TRUE-aktiv)

3.5 DS2450_V8

Der Datentyp „DS2450_V8“ definiert und speichert Vorgaben Werte und Status für den AD-Wandler DS2450.

*.Name: STRING	Name des Sensors (optional)
*.ID: STRING(16)	8-stellige ID des Sensors
*.AI1:REAL	Messergebnis ADC A
*.AI2:REAL	Messergebnis ADC B
*.AI3:REAL	Messergebnis ADC C
*.AI4:REAL	Messergebnis ADC D
*.E.	Aufloesung_A:INT Auflösung Kanal A
	Aufloesung_B:INT Auflösung Kanal B
	Aufloesung_C:INT Auflösung Kanal C
	Aufloesung_D:INT Auflösung Kanal D
	u_ref_A:BOOL 0=2,56V/1=5,12V
	u_ref_B:BOOL 0=2,56V/1=5,12V

	u_ref_C:BOOL	0=2,56V/1=5,12V
	u_ref_D:BOOL	0=2,56V/1=5,12V
	AI_DO:BYTE	2#Freigabe,Belegung linke Tetrade Freigabe DCBA, 1-messen/0 -nix rechte Tetrade Belegung DCBA, 1-Binärausgang/0-AI oder frei
	Anzahl_DO:INT	wird automatisch ermittelt und eingetragen
*.Ausgangsdaten.	DO_A:BOOL	Signalzustand Kanal A, wenn als Ausgang verwendet
	DO_B:BOOL	Signalzustand Kanal B, wenn als Ausgang verwendet
	DO_C:BOOL	Signalzustand Kanal C, wenn als Ausgang verwendet
	DO_D:BOOL	Signalzustand Kanal D, wenn als Ausgang verwendet
*.Status	DO_A:BOOL	Status Kanal A, wenn als Ausgang verwendet, interne Verwendung
	DO_B:BOOL	Status Kanal B, wenn als Ausgang verwendet, interne Verwendung
	DO_C:BOOL	Status Kanal C, wenn als Ausgang verwendet, interne Verwendung
	DO_D:BOOL	Status Kanal D, wenn als Ausgang verwendet, interne Verwendung
*.Stoerungen:INT	Anzahl der Sensorstörungen	
*.Fehler:BOOL	es sind mehr als unter Sensoren Fehleranzahl definierte Störungen aufgetreten	
*.aktiv :BOOL	Sensor wird bearbeitet (TRUE-aktiv)	

3.6 one_wire_binaer_V8

Der Datentyp „one_wire_binaer_V8“ definiert und speichert Vorgaben Werte und Status für die Binärbausteine DS2405, DS2406 und DS2413.

*.Name:STRING	Name des Sensors (optional)		
*.ID. STRING(16)	8-stellige ID des Sensors		
*.Status_Byte:BYTE	Status der Ports	nur	Bit 0 bei DS2405 und Bit 0 und 1 bei DS2413 verwendet
			Bit 2 ⁶ und 2 ⁷ werden als Hilfsbit bei DS2405 verwendet
*.Aenderung_Byte:BYTE	Port, an welchem sich der Zustand geändert hat		
*.Ausgangs_Byte:BYTE	Wert, welcher geschrieben werden soll		
*.DO_verwendet:BYTE	interne Verwendung		
*.DS2405_DI_06_single:BYTE	TRUE, wenn ein DS2405 als DI verwendet oder bei DS2406 nur Port A verwendet wird, sonst bedeutungslos		
*.Stoerungen:INT	Anzahl der Sensorstörungen		
*.Fehler:BOOL	es sind mehr als unter Binaerbaustein Fehleranzahl definierte Störungen aufgetreten		
*.aktiv:BOOL	Sensor wird bearbeitet (TRUE-aktiv)		

3.7 DS2408_V8

Der Datentyp „DS2408_V8“ definiert und speichert Vorgaben Werte und Status für den Binärbaustein DS2408.

Da der DS2408 zur LCD-Anzeige verwendet wird beinhaltet dieser Datentyp den Datentyp LCD in welcher LCD-spezifische Daten abgelegt werden.

*.Name: STRING	Name des Sensors (optional)
*.ID:STRING(16)	8-stellige ID des Sensors
*.Status_Byte:BYTE	Status der Ports
*.Aenderung_Byte: BYTE	Port, an welchem sich der Zustand geändert hat
*.Ausgangs_Byte: BYTE	Wert, welcher geschrieben werden soll
*.DO_verwendet: BYTE	interne Verwendung
*.LCD: STRUCT:LCD	wird zur LCD Ansteuerung verwendet
*.Stoerungen:INT	Anzahl der Sensorstörungen
*.Fehler:BOOL	es sind mehr als unter Binaerbaustein_Fehleranzahl definierte Störungen aufgetreten
*.aktiv:BOOL	Sensor wird bearbeitet (TRUE-aktiv)

3.8 DS2423_V8

Der Datentyp „DS2423_V8“ definiert und speichert Vorgaben Werte und Status für den Zähler DS2423.

*.Name:STRING	Name des Sensors (optional)
*.ID:STRING(16)	8-stellige ID des Sensors
*.zaehler1:DINT	Zählerstand Kanal A
*.zaehler2:DINT	Zählerstand Kanal B
*.Stoerungen:INT	Anzahl der Störungen
*.Fehler:BOOL	es sind mehr als unter Binaerbaustein_Fehleranzahl definierte Störungen aufgetreten
*.aktiv:BOOL	Baustein wird bearbeitet (TRUE-aktiv)

3.9 LCD_Text_163

Der Datentyp „LCD_Text_163“ definiert die Zeilentexte und die Hintergrundbeleuchtung für LCD_163.

*.Zeile1: STRING(16)	(Text Zeile 1)
*.Zeile2: STRING(16)	(Text Zeile 2)
*.Zeile3: STRING(16)	(Text Zeile 3)
*.Beleuchtung: BYTE	(Backlight 0-Aus 1-50% 2-100% 3- Blinken 1Hz)

3.10 LCD_Text_4_20

Der Datentyp „LCD_Text_4_20“ definiert die Zeilentexte und die Hintergrundbeleuchtung für ein 4-Zeiliges LCD Display mit 20 Zeichen pro Zeile.

*.Zeile1: STRING(20)	(Text Zeile 1)
*.Zeile2: STRING(20)	(Text Zeile 2)
*.Zeile3: STRING(20)	(Text Zeile 3)
*.Zeile4: STRING(20)	(Text Zeile 4)
*.Beleuchtung: BOOL	(Backlight-Status, TRUE-EIN)

4. Bausteine

4.1 one_wire

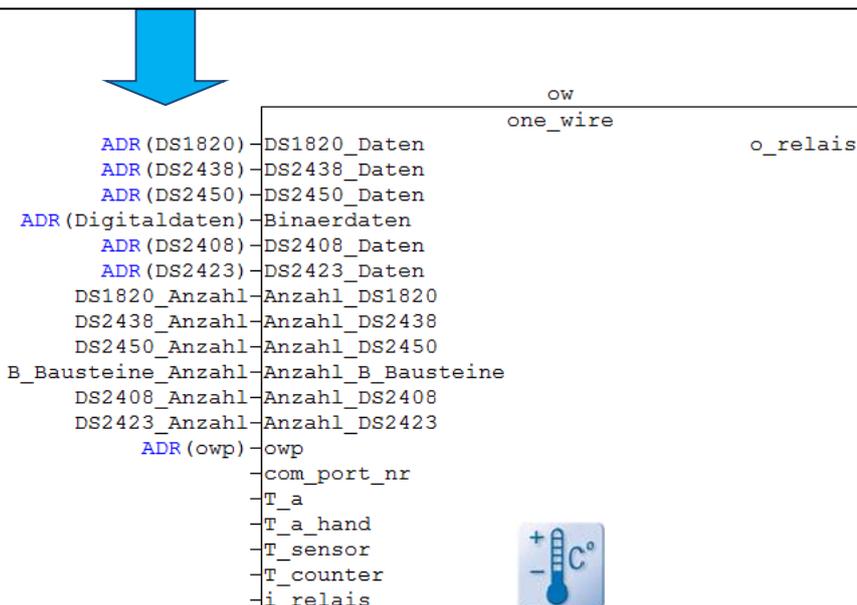
Funktionsbaustein

Input			
DS1820_Daten	POINTER TO ARRAY		Zeiger auf das Array der DS1820
DS2438_Daten	POINTER TO ARRAY		Zeiger auf das Array der DS2438
DS2450_Daten	POINTER TO ARRAY		Zeiger auf das Array der DS450
Binaerdaten	POINTER TO ARRAY		Zeiger auf das Array der Binärsensoren
DS2408_Daten	POINTER TO ARRAY		Zeiger auf das Array der DS2480
DS2423_Daten	POINTER TO ARRAY		Zeiger auf das Array der DS2423
Anzahl_DS1820	INT		Anzahl DS1820
Anzahl_DS2438	INT		Anzahl DS2438
Anzahl_DS2450	INT		Anzahl DS2450
Anzahl_B_Bausteine	INT		Anzahl Binärsensoren
Anzahl_DS2408	INT		Anzahl DS2408
Anzahl_DS2423	INT		Anzahl DS2423
owp	POINTER TO owp		Zeiger auf das Array ow_parameter

INPUT CONSTANT			
com_port_nr	BYTE	Nummer des COM-Ports	[Default 16#1]
Ta	TIME	Aufrufintervall, Intervall der Bausteinaufrufe	[Default T#60ms]
T_a_hand	BOOL	TRUE -Aufrufintervall wird manuell eingestellt FALSE -Aufrufintervall wird automatisch eingestellt	
T_sensor	TIME	Abtastintervall der Sensoren	[Default T#4m]
T_counter	TIME	Abtastintervall DS2423	[Default T#10m]
i_relais	BOOL	TRUE-Signal, aktiviert Spannungsunterbrechung ow-Master durch Relais am Ausgang o_relais im Fehlerfall	[Default FALSE]

OUTPUT		
o_relais	BOOL	Relais zur Spannungsunterbrechung

Dem Funktionsbaustein **one_wire** werden die Daten bzw. Adressen der Daten, welche in den Globalen Variablen definiert wurden, übergeben.



Mit dem Funktionsbaustein **one_wire** ist ein Zugriff auf die one-wire Bausteine DS2405, DS2406, DS2408, DS2413, DS18(B)20, DS2423, DS2438 und DS2450 möglich.

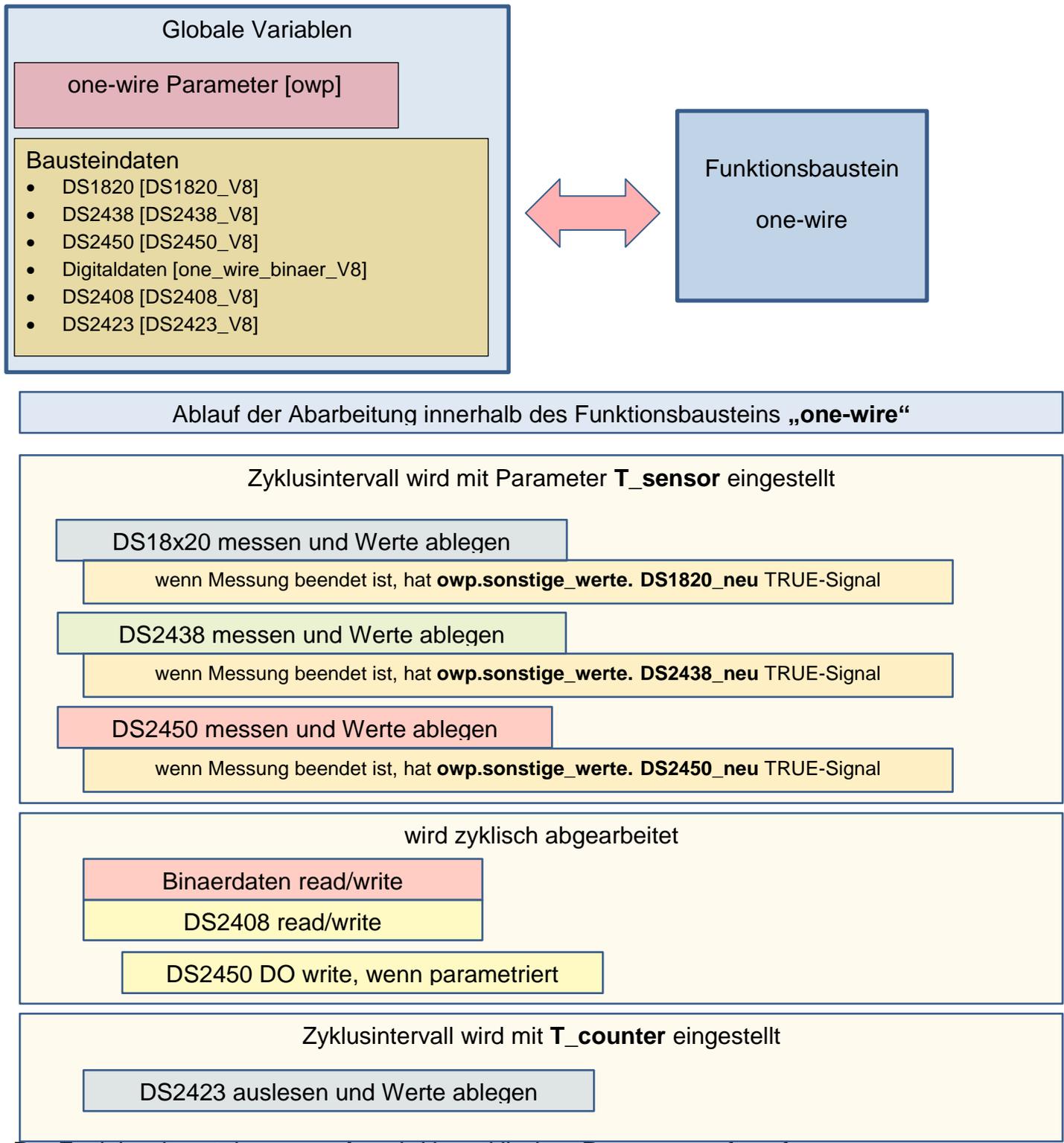
Ein parasitärer Betrieb ist **nicht** möglich!

Die wichtigsten Einstellungen werden direkt am Baustein vorgenommen.

Die Daten der Sensoren werden aus den globalen Variablen gelesen und dort abgelegt.

Hinweis:

wenn z.B. „DS2450“ nicht verwendet wird, dann bei „DS2450_Daten“ und „Anzahl_DS2450“ eine 0 eintragen.



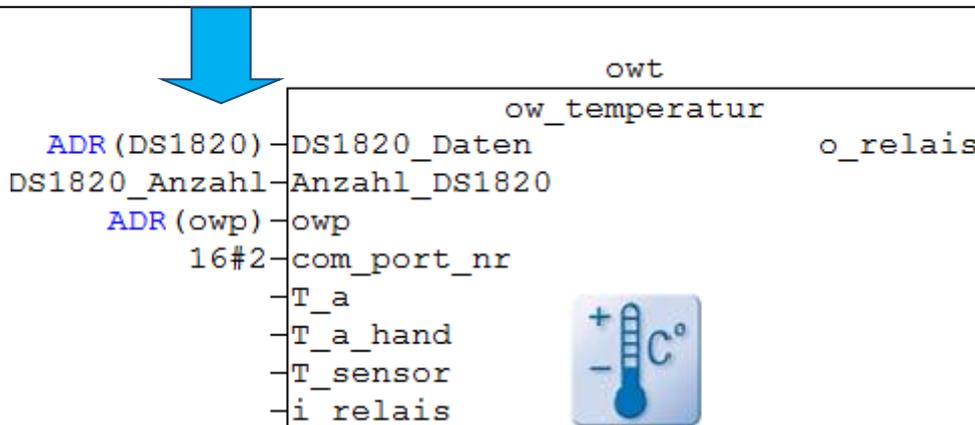
Der Funktionsbausteins **one_wire** wird im zyklischen Programm aufgerufen (mit 4 – 90ms Zykluszeit getestet).

4.2 ow_temperatur

Funktionsbaustein

Input	DS1820_Daten	POINTER TO ARRAY	Zeiger auf das Array der DS1820
	Anzahl_DS1820	INT	Anzahl DS1820
	owp	POINTER TO owp	Zeiger auf das Array ow_parameter
INPUT	com_port_nr	BYTE	Nummer des COM-Ports
CONSTANT	<u>Ta</u>	TIME	Aufrufintervall, Intervall der Bausteinaufrufe
	T_a_hand	BOOL	TRUE -Aufrufintervall wird manuell eingestellt FALSE-Aufrufintervall wird automatisch eingestellt
	T_Sensor	TIME	Abtastintervall der Sensoren
	i_relais	BOOL	TRUE, aktiviert Spannungsunterbrechung ow-Master durch Relais am Ausgang o_relais im Fehlerfall
OUTPUT	o_relais	BOOL	Relais zur Spannungsunterbrechung

Dem Funktionsbaustein **ow_temperatur** werden die Daten bzw. Adressen der Daten, welche in den Globalen Variablen definiert wurden, übergeben.

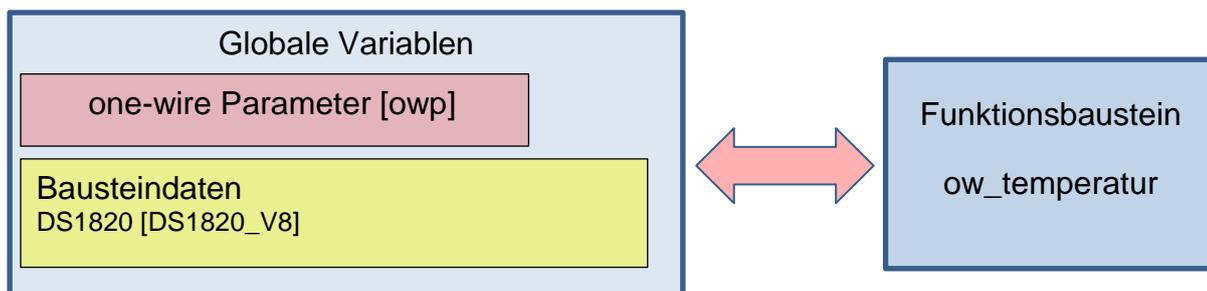


Mit dem Funktionsbaustein **ow_temperatur** ist eine Temperaturmessung mit dem one-wire Baustein DS18x20 möglich.

Ein parasitärer Betrieb ist **nicht** möglich!

Die wichtigsten Einstellungen werden direkt am Baustein vorgenommen.

Die Daten der Sensoren werden aus den globalen Variablen gelesen und dort abgelegt.

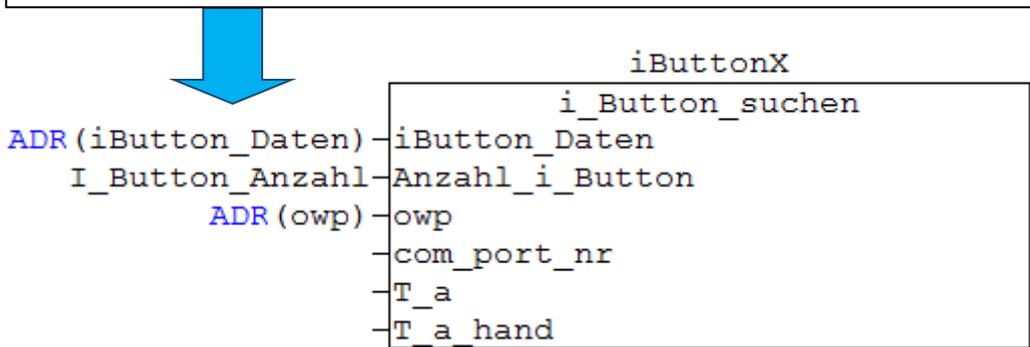


4.3 i_Button_suchen

Funktionsbaustein

Input	iButton_Daten	POINTER TO ARRAY	Zeiger auf das Array der iButton-Daten
	Anzahl_i_Button	INT	Anzahl iButton
	owp	POINTER TO owp	Zeiger auf das Array ow_parameter
INPUT	com_port_nr	BYTE	Nummer des COM-Ports
CONSTANT	T_a	TIME	Aufrufintervall, Intervall der Bausteinaufrufe
	T_a_hand	BOOL	TRUE -Aufrufintervall wird manuell eingestellt FALSE-Aufrufintervall wird automatisch eingestellt

Dem Funktionsbaustein **i_Button_suchen** werden die Daten bzw. Adressen der Daten, welche in den Globalen Variablen definiert wurden, übergeben.



Der Funktionsbaustein **i_Button_suchen** ist die Abfrage der Anwesenheit von iButton am 1-Wire Netz möglich.

- Der Funktionsbausteins ‚i_Button_suchen‘ wird im zyklischen Programm aufgerufen (mit 10 – 50ms Zykluszeit getestet).
- Jede gefundene iButton-ID wird mit den hinterlegten ID’s verglichen und bei Übereinstimmung wird **aktiv = TRUE**.

Globale Variablen: one_wire_Daten

```

Anzahl i Button = 16#0003
├─ owp
├─ iButton
│   └─ iButton[1]
│       ├── .Name = 'nummer 1'
│       ├── .ID
│       └── .aktiv = FALSE
│   └─ iButton[2]
│       ├── .Name = 'nummer 2'
│       ├── .ID
│       └── .aktiv = FALSE
│   └─ iButton[3]
│       ├── .Name = 'nummer 3'
│       ├── .ID
│       └── .aktiv = FALSE

```

TRUE-Signal, wenn ID mit gefundener ID übereinstimmt

4.4 ow_temperatur_i_button

Funktionsbaustein

Input	DS1820_Daten	POINTER TO ARRAY	Zeiger auf das Array der DS1820
	iButton_Daten	POINTER TO ARRAY	Zeiger auf das Array der iButton-Daten
	Anzahl_DS1820	INT	Anzahl DS1820
	Anzahl_i_Button	INT	Anzahl iButton
	owp	POINTER TO owp	Zeiger auf das Array ow_parameter
INPUT	com_port_nr	BYTE	Nummer des COM-Ports
CONSTANT	<u>Ta</u>	TIME	Aufrufintervall, Intervall der Bausteinaufrufe
	T_a_hand	BOOL	TRUE -Aufrufintervall wird manuell eingestellt FALSE -Aufrufintervall wird automatisch eingestellt
	T_Sensor	TIME	Abtastintervall der Sensoren
	i_relais	BOOL	TRUE, aktiviert Spannungsunterbrechung ow-Master durch Relais am Ausgang o_relais im Fehlerfall
OUTPUT	o_relais	BOOL	Relais zur Spannungsunterbrechung

Der Baustein „ow_temperatur_i_button“ ist eine Kombination aus „ow_temperatur“ und „i_Button_suchen“.

5. Anwendungshinweise

5.1 Daten eintragen

Für eine ordnungsgemäße Funktion müssen unter „Globale Variablen“ folgende Daten eingetragen werden:

1. Die Anzahl der verwendeten Sensoren / Bausteine
2. die one-wire-Parameter (Struktur **owp**)
3. für jede verwendete Sensor- Bausteingruppe ein Array mit einer Größe wie unter Punkt 1 (Anzahl Sensoren / Bausteine) definiert wurde mit der Struktur der jeweiligen Sensor- Bausteingruppe.

Beispiel:

1.	<pre> VAR_GLOBAL CONSTANT DS1820_Anzahl : INT := 3; DS2438_Anzahl : INT := 3; DS2450_Anzahl : INT := 1; Anzahl_B_Bausteine: INT := 2; (*Anzahl an binären 1-wire Bausteinen *) DS2408_Anzahl : INT := 2; DS2423_Anzahl : INT := 2; Anzahl_I_Button: INT := 2; END_VAR </pre> <p>Wenn ein Typ nicht verwendet wird kann er weggelassen werden!</p>
2.	<pre> owp: owp; (*Parameter für one_wire Baustein*) </pre>
3.	<pre> DS1820 DS1820 : ARRAY [1.. DS1820_Anzahl] OF DS1820_V8:= (Name:='Sensor Box',ID:='68000002EF81B528',aktiv:= TRUE), (Name:='LCD Box', ID:=' 4E000002EFB12B28',aktiv:= TRUE); DS2438 DS2438 : ARRAY [1.. DS2438_Anzahl] OF DS2438_V8:= (Name:='Sensor Box',ID:='AD000000CB049A26',aktiv:=TRUE), (Name:='gelbe Box', ID:='8A000000CFDD1826',aktiv:=TRUE); DS2450 DS2450 : ARRAY [1.. DS2450_Anzahl] OF DS2450_V8:= (name:='gelbe Box',ID:= ,68000002EF81B520',aktiv:=TRUE); Binärsensoren: Binaersensoren: ARRAY[1..A_Anzahl_B_Bausteine] OF one_wire_binaer_V8:= (Name:='DI/DO Box 2405', ID:='68000002EF81B505', DS2405_DI:=FALSE, aktiv:= TRUE), (Name:='Sensor Box 2413', ID:='68000002EF81B53A', aktiv:= TRUE); DS2408: DS2408: ARRAY[1..Anzahl_2408] OF DS2408_V8:= (Name:='LCD 1', ID:='68000002EF81B529',aktiv:= TRUE), (Name:='DI/DO Box 2408', ID:='68000002EF81B529',aktiv:= TRUE); </pre>

5.2 Zugriff auf Daten

Der Zugriff auf die Daten im Programm erfolgt folgendermaßen:

DS18x20: T_Sensordaten [**Nummer des Sensors**]. Temperatur

Beispiel: DS1820[1].Temperatur

DS2438:

DS2438 [**Nummer des Sensors**]. Temperatur

DS2438 [**Nummer des Sensors**]. VAD

Spannungssignal am AD Eingang

DS2438 [**Nummer des Sensors**]. VDD

Busspannung

DS2438 [**Nummer des Sensors**]. XSENS

Spannungssignal / Stromeingang

Beispiel: DS2438 [2].VAD

DS2438 [2].VDD

DS2438 [2].XSENS

DS2450:

DS2450 [**Nummer des Sensors**]. AI Kanalnummer

Beispiel: DS2450 [2]. AI1

one_wire_binaer DS2413:

Binaersensoren [**Nummer des Sensors**]. Status_Byte.0

Statusabfrage von Bit 2⁰

Binaersensoren [**Nummer des Sensors**]. Aenderungs_Byte.1

Änderungsabfrage von Bit 2¹

Binaersensoren [**Nummer des Sensors**]. Ausgangs_Byte.1

Zugriff auf Ausgangsbit 2¹

Beispiel: Binaersensoren [1].Status_Byte.0

Binaersensoren [1].Aenderungs_Byte.1

Binaersensoren [1].Ausgangs_Byte.1

DS2408:

DS2408 [**Nummer des Sensors**]. Status_Byte.4

Statusabfrage von Bit 2⁴

DS2408 [**Nummer des Sensors**]. Aenderungs_Byte.5

Änderungsabfrage von Bit 2⁵

DS2408 [**Nummer des Sensors**]. Ausgangs_Byte.6

Zugriff auf Ausgangsbit 2⁶

Beispiel: DS2408 [2].Status_Byte.4

DS2408 [2].Aenderungs_Byte.5

DS2408 [2].Ausgangs_Byte.6

5.3 ID Suchen

Es ist möglich die ID's der angeschlossenen Sensoren zu suchen.

Vorgehensweise: **Onlinemodus CoDeSys**

einstellen was gesucht werden soll: z.B. DS18x20

Globale Variablen →one_wire_Daten→

owp.ID_suchen.was_suchen.DS1820 TRUE (Strg+F7 [Werte schreiben])

Suche starten:

Globale Variablen →one_wire_Daten→

owp.ID_suchen.neue_ID_suchen TRUE (Strg+F7 [Werte schreiben])

Nun wird im 1-Wire Netz nach ID's gesucht. Gefundenen ID's werden mit den schon eingetragenen verglichen und wenn eine noch nicht eingetragene ID gefunden wird, wird sie an der Position im Array eingetragen, an welcher die letzten beiden Ziffern (ID [String]) eine ,00' ist oder die ID leer ist.

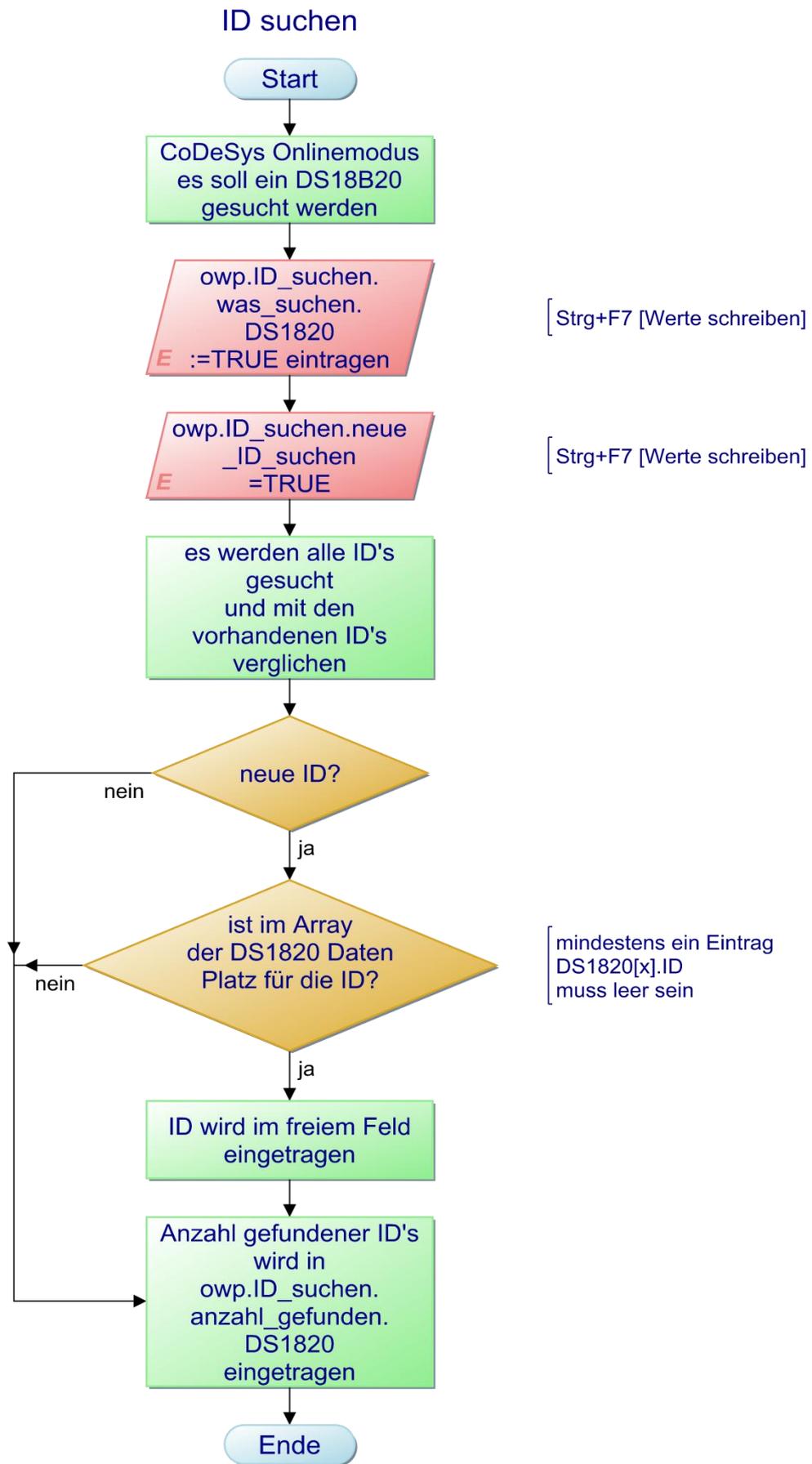
Am Ende der ID-suche wird unter

Globale Variablen →one_wire_Daten→**owp.ID_suchen.anzahl_gefunden.DS1820** die Anzahl der neu gefundenen, *nicht der vorhandenen*, ID's eingetragen und

owp.ID_suchen.neue_ID_suchen wird auf **FALSE** zurückgesetzt.

Ein manuelles eintragen der gefundenen ID's ist aber empfehlenswert da

1. Die gefundene ID nach einem Neustart des Controllers nicht mehr vorhanden ist.
2. Die Zuordnung der ID's zufällig erfolgt. D.h. wenn mehrere ID's gesucht werden sollen ist die Zuordnung schwer.



5.4 Bausteineinstellungen

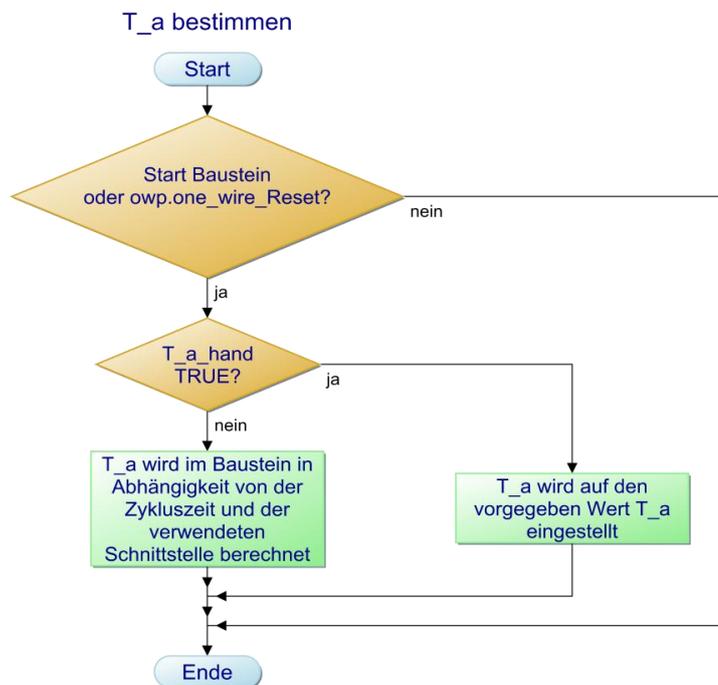
5.4.1 Aufrufintervall „T_a“

Der Parameter **T_a** bestimmt die Geschwindigkeit der Datenübertragung, mit einem Wert von T#30 - 100ms, von der verwendeten Schnittstelle abhängig, sollte auch ein größeres Netzwerk stabil laufen.

Wenn der Bausteineingang **T_a** nicht belegt ist, wird die Zeit automatisch in Abhängigkeit von Zykluszeit und verwendeter Schnittstelle berechnet.

Wenn **T_a** manuell eingestellt werden soll ($T_a_hand = TRUE$) ist folgendes zu beachten:

- Bei Verwendung der internen Schnittstelle COM1 muss **T_a** mindestens **1,4-mal Zykluszeit** sein.
- Bei Verwendung einer seriellen Klemme ($COM > 1$) muss für **T_a** mindestens 60ms eingetragen werden.
- Der Parameter **T_a** wird nur bei Reset übernommen siehe PAP.
- Wenn viele Sensorstörungen auftreten, muss der Wert vergrößert werden (in 10ms Schritten).



5.5 Verwendung von mehreren Schnittstellen

Es ist unter Berücksichtigung folgender Punkte der Einsatz mehrerer serieller Schnittstellen möglich:

- Es müssen one-wire-Parameter (Struktur **owp**) für jede Schnittstelle angelegt werden

```
owp_1: owp; (*Parameter für COM 1*)
owp_2: owp; (*Parameter für COM2 *)
```

- Die Anzahl der verwendeten Sensoren / Bausteine muss für jede Schnittstelle getrennt angelegt werden.

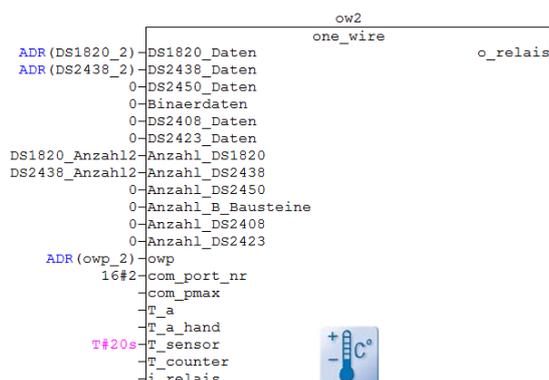
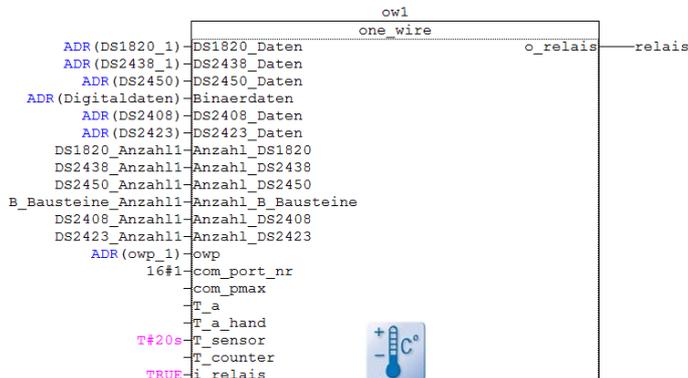
```
DS1820_Anzahl1: INT := 6;(*Anzahl für COM 1*)
DS1820_Anzahl2: INT := 2;(*Anzahl für COM 2*)
```

- Die Daten-Arrays der Sensor- Bausteingruppen müssen für jede Schnittstelle getrennt angelegt werden.

```
DS1820_1: ARRAY[1..DS1820_Anzahl1] OF DS1820_V8:=
(Name:='1',ID:='68000002ef81b528',aktiv:= TRUE),
(Name:='2',ID:='4E000002EFB12B28',aktiv:= TRUE),
(Name:='3',ID:='4000000384094528',aktiv:= TRUE),
(Name:='3',ID:='E50000035530E528',aktiv:= TRUE);
```

```
DS1820_2: ARRAY[1..DS1820_Anzahl2] OF DS1820_V8:=
(Name:='1',ID:='68000002ef81b528',aktiv:= TRUE),
(Name:='2',ID:='4E000002EFB12B28',aktiv:= TRUE);
```

- Es muss für jede Schnittstelle ein Baustein mit eigener Instanz aufgerufen werden



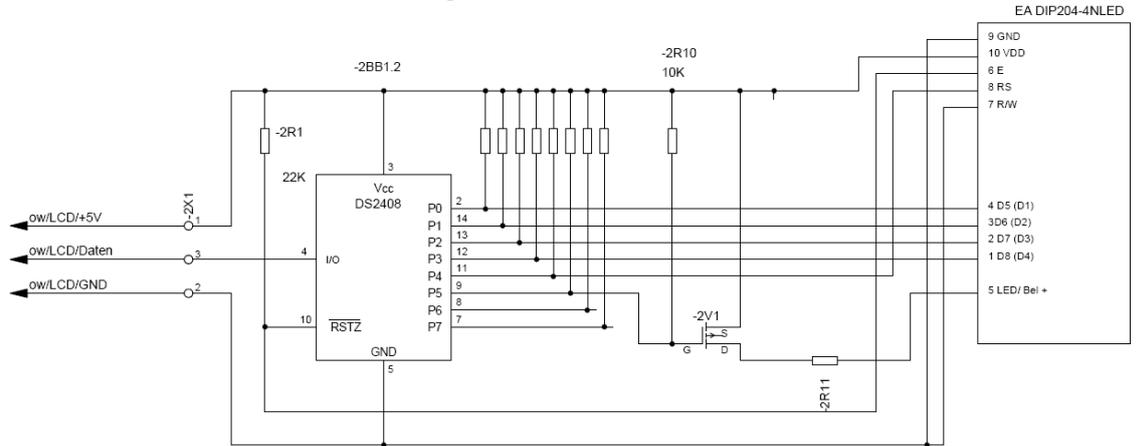
6. LCD

6.1 Ansteuerung

6.1.1 LCD_Tool_ow

Der Funktionsbaustein **LCD_tool_ow** ermöglicht die Verwendung einer LCD Anzeige, welche mit einem DS2408 folgendermaßen am one-wire Bus angeschlossen ist.

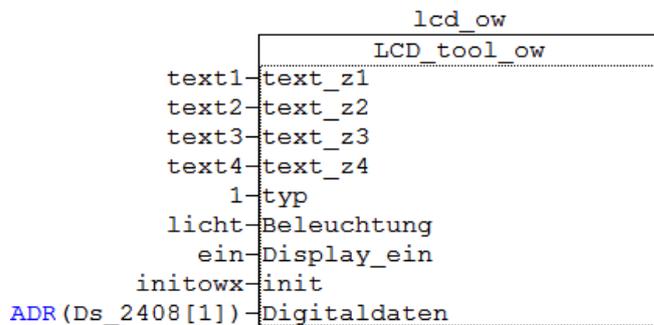
DS2408	LCD
P0	D4
P1	D5
P2	D6
P3	D7
P4	RS
P5	LED
RSTZ	E



Der LCD-Controller kann am Eingang Typ folgendermaßen eingestellt werden:

- 0-standard 4-Zeilen
- 1-KS0073-4 Zeilen
- 2-16x2
- 3-ST7036 3 Zeilen

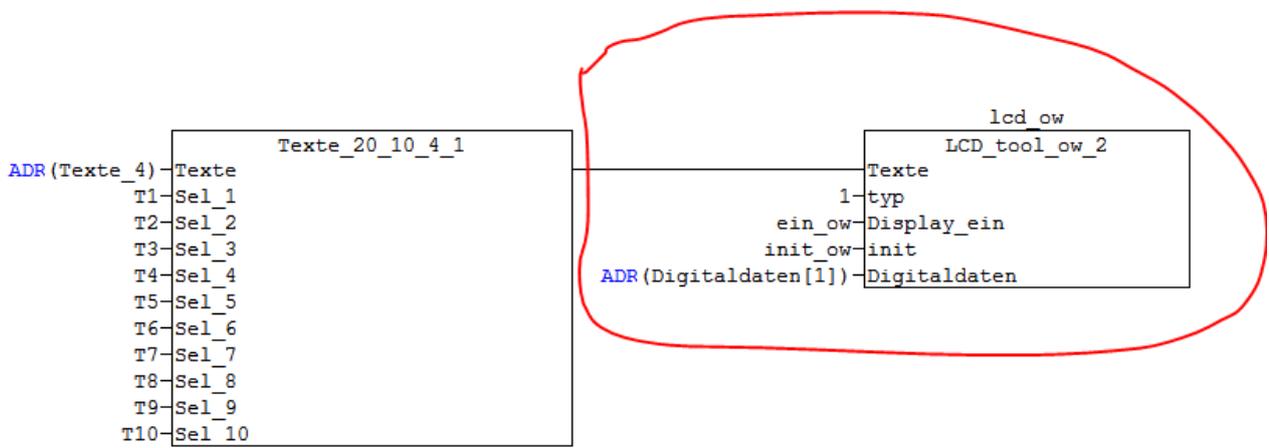
Input	text_z1:	STRING(20)	(Text Zeile 1)
	text_z2:	STRING(20)	(Text Zeile 2)
	text_z3:	STRING(20)	(Text Zeile 3)
	text_z4:	STRING(20)	(Text Zeile 4)
	typ:	INT	(0-standard 4-Zeilen 1-KS0073-4 Zeilen 2-16x2 3-ST7036 3 Zeilen)
	Beleuchtung:	BOOL	(Hintergrundbeleuchtung steuern)
	Display_ein:	BOOL	(Display Ein- Ausschalten)
	init:	BOOL	(bei Signalwechsel wird Baustein initialisiert)
	Digitaldaten:	POINTER TO DS2408_V8	(Zeiger auf DS2408 an welchem die LCD Anzeige angeschlossen ist)



6.1.2 LCD_Tool_ow_2

Dieser Baustein ist optimiert für die Funktion [Texte 20 10 4 x](#) mit welcher eine optimierte Textausgabe möglich ist.

Input	Texte:	LCD_Text_4_20	(Texte 20 Zeichen 4 Zeilen + Hintergrundbeleuchtung)
	Typ:	INT	(0-standard 4-Zeilen 1-KS0073-4 Zeilen 2-16x2 3-ST7036 3 Zeilen)
	Display_ein:	BOOL	(Display Ein- Ausschalten)
	init:	BOOL	(bei Signalwechsel wird Baustein initialisiert)
	Digitaldaten:	POINTER TO DS2408_V8	(Zeiger auf DS2408 an welchem die LCD Anzeige angeschlossen ist)

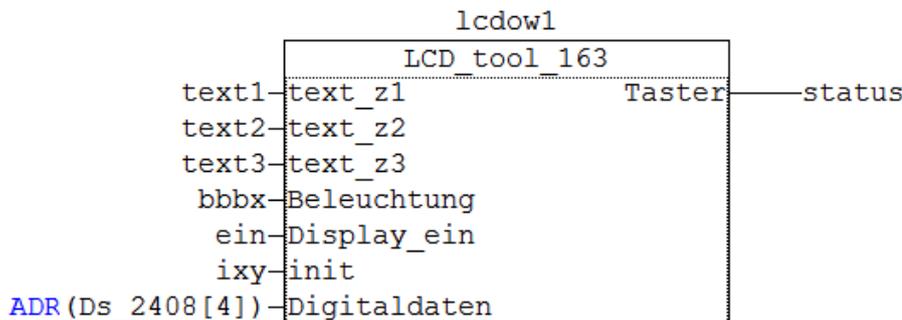


6.1.3 LCD_tool_163

Der Funktionsbaustein LCD_tool_163 ist für das „LCD-Display 163“ von <http://www.eservice-online.de> optimiert.

Daten und Anschlussbelegung werden vom Hersteller dort bereitgestellt.

INPUT	text_z1:STRING(16);	(Text Zeile 1)
	text_z2:STRING(16);	(Text Zeile 2)
	text_z3:STRING(16);	(Text Zeile 3)
	Beleuchtung: BYTE;	(Hintergrundbeleuchtung steuern s. Datenblatt)
	Display_ein: BOOL;	(Display Ein- Ausschalten)
	init: BOOL;	(bei Signalwechsel wird Baustein initialisiert)
	Digitaldaten: POINTER TO DS2408_V8;	(DS2408 an welchem die LCD Anzeige angeschlossen ist)
Output	Taster:BOOL;	(Status des Tasters am Display)



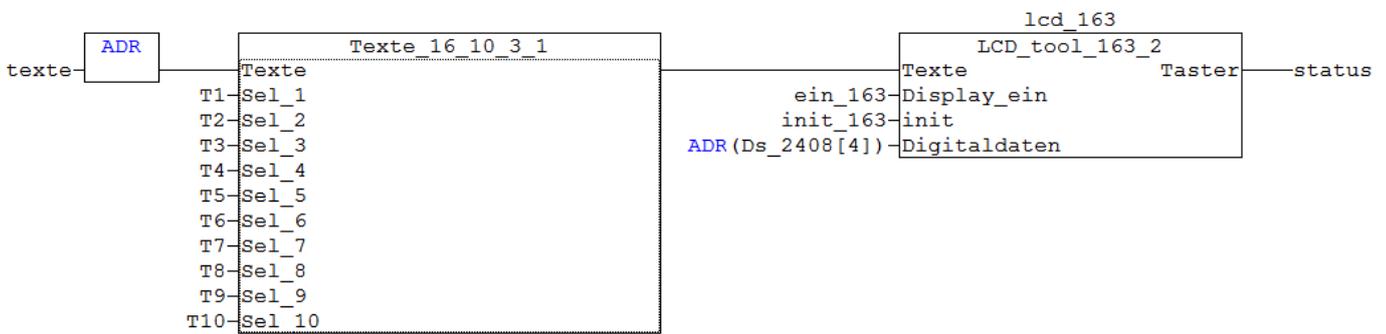
6.1.4 LCD_tool_163_2

Der Funktionsbaustein LCD_tool_163 ist für das „LCD-Display 163“ von <http://www.eservice-online.de> optimiert.

Daten und Anschlussbelegung werden vom Hersteller dort bereitgestellt.

Dieser Baustein ist optimiert für die Funktion [Texte 16 10 3 x](#) mit welcher eine optimierte Textausgabe möglich ist.

Input	Texte	LCD_Text_163	(Texte 16 Zeichen 3 Zeilen + Hintergrundbeleuchtung)
	Display_ein:	BOOL	(Display Ein- Ausschalten)
	init:	BOOL	(bei Signalwechsel wird Baustein initialisiert)
	Digitaldaten:	POINTER TO DS2408_V8	(DS2408 an welchem die LCD Anzeige angeschlossen ist)
OUTPUT	Taster:	BOOL	(Status des Tasters am Display)

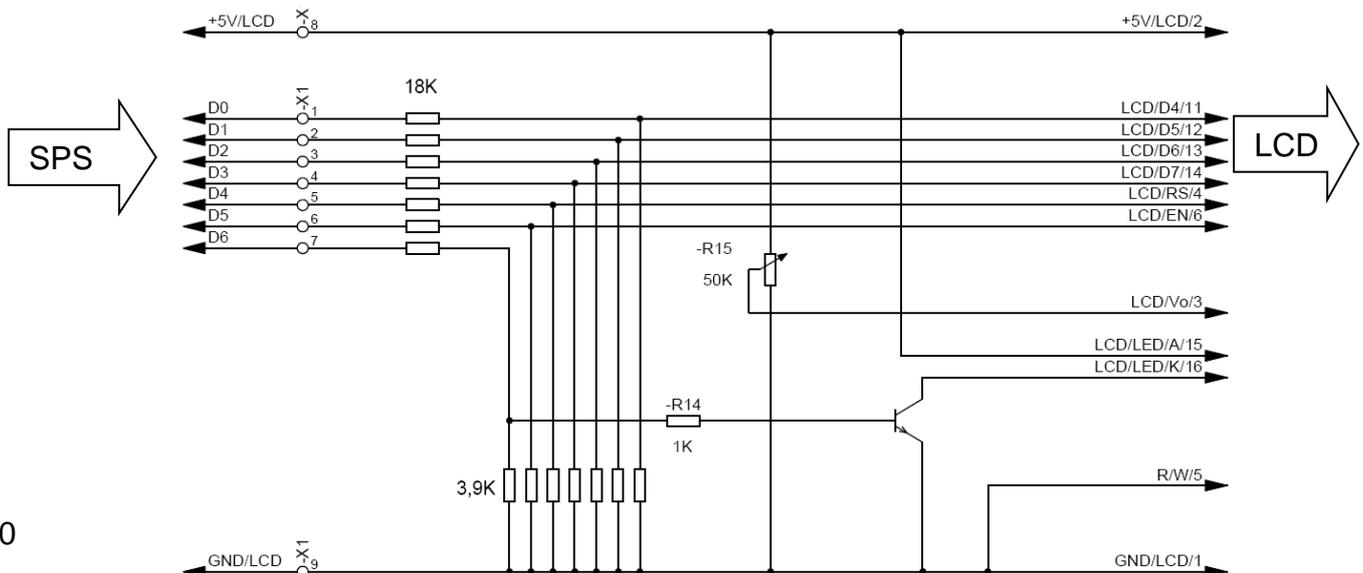


6.1.5 LCD an Digitalausgängen (LCD_tool_DO)

Der Funktionsbaustein **LCD_tool_DO** ermöglicht die Verwendung einer LCD Anzeige welche an binären Ausgängen folgendermaßen angeschlossen ist. **(nicht am 1-wire Bus!!!)**

LCD Steuerung mit SPS Binärausgängen	SPS	LCD
	Bit 0/2 ⁰	D4
	Bit 1/2 ¹	D5
	Bit 2/2 ²	D6
	Bit 3/2 ³	D7
	Bit 4/2 ⁴	RS
	Bit 5/2 ⁵	EN
	Bit 6/2 ⁶	LED

Anschlussbeispiel (YM-2004A)

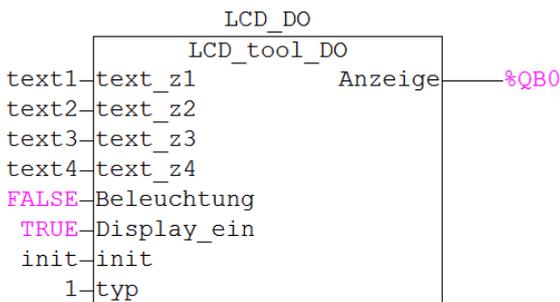


Input	text_z1:	STRING	(Text Zeile 1)
	text_z2:	STRING	(Text Zeile 2)
	text_z3:	STRING	(Text Zeile 3)
	text_z4:	STRING	(Text Zeile 4)
	Beleuchtung:	BOOL	Steuerung der Hintergrundbeleuchtung
	Display_ein:	BOOL	Display Ein-Ausschalten
	init:	BOOL	(bei TRUE-Signal wird Baustein initialisiert)
	typ:	INT	Displaytyp einstellen
Output	Anzeige:	BYTE	(Ausgangsbyte an welchem LCD Anzeige angeschlossen ist)

Hinweis zur Adressierung:

Byte-Adresse: Byte = 2 x Wortadresse

Typ	Display
0	Standard 4-Zeilen mit 20 Zeichen / Zeile
1	KS0073 4-Zeilen mit 20 Zeichen / Zeile
2	2-Zeilen mit 16 Zeichen / Zeile
3	ST7036 3 Zeilen



6.2 LCD Texte

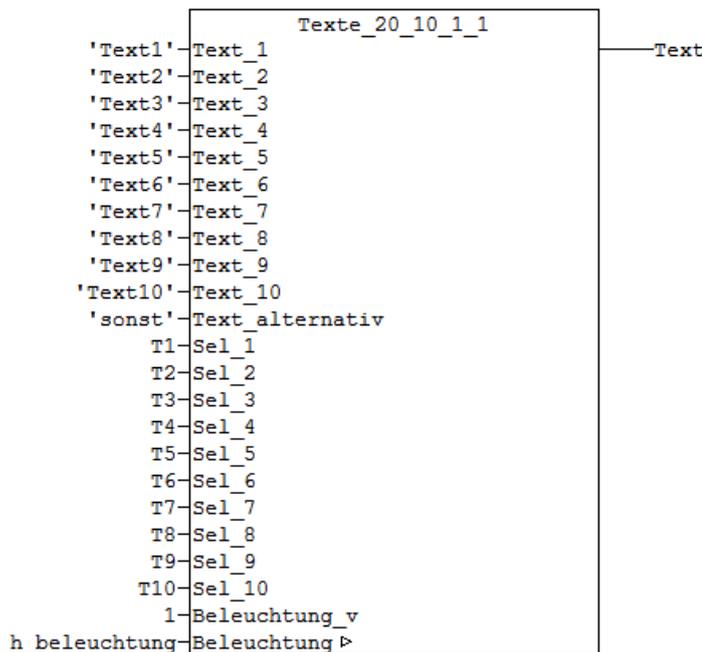
Bezeichnung der Bausteine: **Texte**_{Zeichenlänge_Anzahl der Texte_Zeilen_x}

x=1 Text kann nur von einem Baustein erstellt werden

x=2 Text kann von mehreren Bausteinen erstellt werden, damit sind mehr als 10 Texte möglich

6.2.1 Texte_20_10_1_1

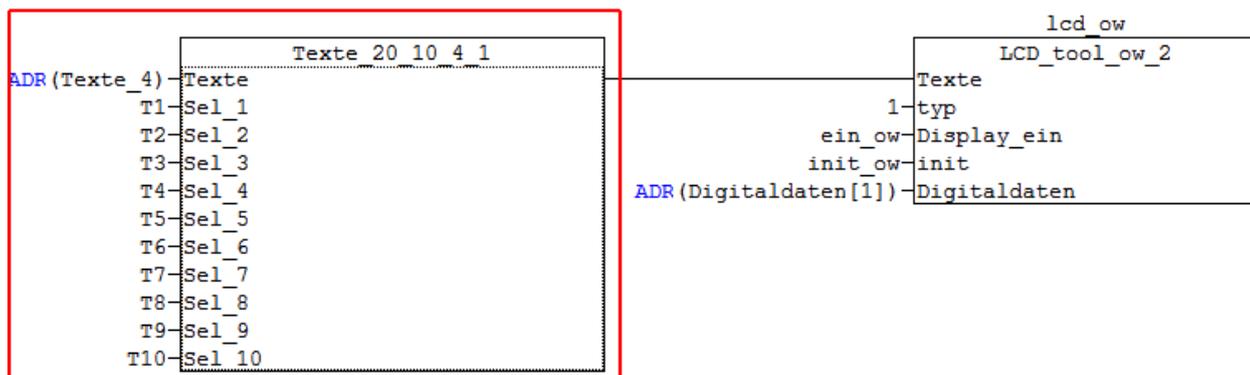
Type	Funktion String (20)		
Input	Text_1:	STRING (20)	(Text bei Sel_1)
	.		
	Text_10:	STRING (20)	(Text bei Sel_10)
	Sel_1:	BOOL	(Text_1 auswählen)
	.		
	Sel_10	BOOL	(Text_10 auswählen)
	Text_alternativ:	STRING (20)	(Text wenn SEL1 bis SEL10 aus ist)
Beleuchtung_v	BYTE	(Auswahl der Hintergrundbeleuchtung bei Text_1 bist Text_10)	
Input_Output	Beleuchtung:	BYTE	(Aktuelle Hintergrundbeleuchtung LCD-Display 163)
OUTPUT		STRING (20)	(Ausgangswert)



Texte_20_10_1_1 wählt abhängig von Sel_1 bis Sel_10 Text_1 bis Text_10 aus. Ist kein Text ausgewählt, wird Text_alternativ ausgegeben. Wenn Text_1 bis Text_10 gewählt wurde wird an Beleuchtung Beleuchtung_v übergeben, sonst hat Beleuchtung den Wert 0.

6.2.2 Texte_20_10_4_1

Type	Funktion	LCD_Text_4_20	
Input	Texte	POINTER TO ARRAY [0..10] OF LCD_Text_4_20	
	Sel_1:	BOOL	(Text_1 auswählen)
	.		
	Sel_10:	BOOL	(Text_10 auswählen)
OUTPUT		LCD_Text_4_20	(Ausgangswert)



Die Funktion **Texte_20_10_4_1** übergibt am Ausgang einen von 11 Texten, welche in einem Array mit der Struktur LCD_Text_4_20 abgelegt sind. Dem Baustein wird der Zeiger auf das Array übergeben.

In einem externen Array können nach folgendem Muster bis zu 11 Texte definiert werden. Wobei für jeden Text **Zeile1**, **Zeile2**, **Zeile3**, **Zeile4** und **Beleuchtung** (Hintergrundbeleuchtung) definiert werden kann.

Text [0] ist der Alternativtext, wenn Text [1] bis Text [10] nicht ausgewählt sind.

Texte_4: ARRAY[0..10] OF LCD_Text_4_20:=

(Zeile1:='sonst Zeile 1',Zeile2:='sonst Zeile 2',Zeile3:='sonst Zeile 3',Zeile4:='sonst Zeile 4',Beleuchtung:=FALSE),

(Zeile1:='Text1 Zeile 1',Zeile2:='Text1 Zeile 2',Zeile3:='Text1 Zeile 3',Zeile4:='Text1 Zeile 4',Beleuchtung:=TRUE),

(Zeile1:='Text2 Zeile 1',Zeile2:='Text2 Zeile 2',Zeile3:='Text2 Zeile 3',Zeile4:='Text2 Zeile 4',Beleuchtung:=TRUE),

(Zeile1:='Text3 Zeile 1',Zeile2:='Text3 Zeile 2',Zeile3:='Text3 Zeile 3',Zeile4:='Text3 Zeile 4',Beleuchtung:=TRUE),

(Zeile1:='Text4 Zeile 1',Zeile2:='Text4 Zeile 2',Zeile3:='Text4 Zeile 3',Zeile4:='Text4 Zeile 4',Beleuchtung:=TRUE),

(Zeile1:='Text5 Zeile 1',Zeile2:='Text5 Zeile 2',Zeile3:='Text5 Zeile 3',Zeile4:='Text5 Zeile 4',Beleuchtung:=TRUE),

(Zeile1:='Text6 Zeile 1',Zeile2:='Text6 Zeile 2',Zeile3:='Text6 Zeile 3',Zeile4:='Text6 Zeile 4',Beleuchtung:=TRUE),

(Zeile1:='Text7 Zeile 1',Zeile2:='Text7 Zeile 2',Zeile3:='Text7 Zeile 3',Zeile4:='Text7 Zeile 4',Beleuchtung:=TRUE),

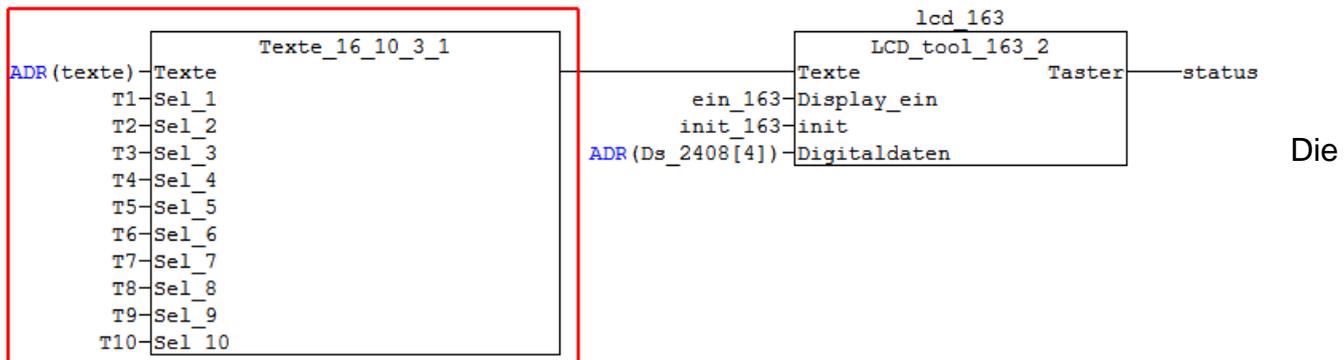
(Zeile1:='Text8 Zeile 1',Zeile2:='Text8 Zeile 2',Zeile3:='Text8 Zeile 3',Zeile4:='Text8 Zeile 4',Beleuchtung:=TRUE),

(Zeile1:='Text9 Zeile 1',Zeile2:='Text9 Zeile 2',Zeile3:='Text9 Zeile 3',Zeile4:='Text9 Zeile 4',Beleuchtung:=TRUE),

(Zeile1:='Text1 Zeile 10',Zeile2:='Text10 Zeile 2',Zeile3:='Text10 Zeile 3',Zeile4:='Text10 Zeile 4',Beleuchtung:=TRUE);

6.2.3 Texte_16_10_3_1

Type	Funktion	LCD_Text_163	
Input	Texte	POINTER TO ARRAY [0..10] OF LCD_Text_163	
	Sel_1:	BOOL	(Text_1 auswählen)
	.		
	Sel_10:	BOOL	(Text_10 auswählen)
OUTPUT		LCD_Text_163	(Ausgangswert)



Funktion **Texte_16_10_3_1** übergibt am Ausgang einen von 11 Texten, welche in einem Array mit der Struktur LCD_Text_163 abgelegt sind. Dem Baustein wird der Zeiger auf das Array übergeben.

In einem externen Array können nach folgendem Muster bis zu 11 Texte definiert werden. Wobei für jeden Text **Zeile1**, **Zeile2**, **Zeile3** und **Beleuchtung** (Hintergrundbeleuchtung) definiert werden kann. Text [0] ist der Alternativtext, wenn Text [1] bis Text [10] nicht ausgewählt sind.

```

Texte: ARRAY[0..10] OF LCD_Text_163:=
    (Zeile1:='sonst Zeile 1',Zeile2:='sonst Zeile 2',Zeile3:='sonst Zeile 3',Beleuchtung:=16#0),
    (Zeile1:='text1 Zeile 1',Zeile2:='text1 Zeile 2',Zeile3:='text1 Zeile 3',Beleuchtung:=16#1),
    (Zeile1:='text2 Zeile 1',Zeile2:='text2 Zeile 2',Zeile3:='text2 Zeile 3',Beleuchtung:=16#1),
    (Zeile1:='text3 Zeile 1',Zeile2:='text3 Zeile 2',Zeile3:='text3 Zeile 3',Beleuchtung:=16#1),
    (Zeile1:='text4 Zeile 1',Zeile2:='text4 Zeile 2',Zeile3:='text4 Zeile 3',Beleuchtung:=16#1),
    (Zeile1:='text5 Zeile 1',Zeile2:='text5 Zeile 2',Zeile3:='text5 Zeile 3',Beleuchtung:=16#1),
    (Zeile1:='text6 Zeile 1',Zeile2:='text6 Zeile 2',Zeile3:='text6 Zeile 3',Beleuchtung:=16#1),
    (Zeile1:='text7 Zeile 1',Zeile2:='text7 Zeile 2',Zeile3:='text7 Zeile 3',Beleuchtung:=16#1),
    (Zeile1:='text8 Zeile 1',Zeile2:='text8 Zeile 2',Zeile3:='text8 Zeile 3',Beleuchtung:=16#1),
    (Zeile1:='text9 Zeile 1',Zeile2:='text9 Zeile 2',Zeile3:='text9 Zeile 3',Beleuchtung:=16#1),
    (Zeile1:='text10 Zeile 1',Zeile2:='text10 Zeile 2',Zeile3:='text10 Zeile 3',Beleuchtung:=16#1);
    
```

6.2.4 Texte_16_10_3_1_nummer

Type	Funktion	LCD_Text_163	
Input	Texte	POINTER TO ARRAY [0..10] OF LCD_Text_163	
	Sel_1:	BOOL	(Text_1 auswählen)
	.		
	Sel_10:	BOOL	(Text_10 auswählen)
	T_nummer_anfang:	INT	(Beginn der Nummerierung)
IN_OUT	T_nummer	INT	(Text-Nummer Text1= T_nummer_anfang Text2= T_nummer_anfang+1 ... kein Text=0)
OUTPUT		LCD_Text_163	(Ausgangswert)

Funktion wie **Texte_16_10_3_1** mit der Erweiterung dass in einer Variablen die Nummer des Textes abgelegt wird.

6.2.5 LCD_Texte_10_nummer

Mit dem Funktionsbaustein **LCD_Texte_10_nummer** kann zwischen 10 Texten umgeschaltet werden.

Weiterhin wird eine Nummer von ‚nummer_anfang‘ bis ‚nummer_anfang‘+10 vergeben.

Wenn kein Text gewählt ist, wird Nummer 0 vergeben.

Input	Text_1	STRING	Text bei Sel_1
	.		
	Text_10	STRING	Text bei Sel_10
	Text_alternativ	STRING	Text wenn SEL1 bis SEL10 ausist
	SEL_1	BOOL	Auswahl Text_1
	.		
	SEL10	BOOL	Auswahl Text_10
	Beleuchtung_v	BYTE	Auswahl der Hintergrundbeleuchtung LCD-Display, wenn Text 1 bis Text 10 aktiv ist (s. Datenblatt LCD-Display 163)
OUTPUT	nummer_anfang	INT	Beginn der Nummerierung
	Text	STRING	Ausgewählter Text
	Beleuchtung	BYTE	Aktuelle Hintergrundbeleuchtung LCD-Display 163
Var_IN_OUT	t_nummer	INT	Textnummer

Beispiel der Textnummerierung:

Aufruf 1	nummer_anfang=1	t_nummer wird von 1 bis 10 vergeben, wenn kein Text, dann 0
Aufruf 2	nummer_anfang=11	t_nummer wird von 11 bis 20 vergeben, wenn kein Text, dann 0

7. Tools

7.1 Luftfeuchte Luftdruck Luftgüte (ow_feuchte_druck)

Der Funktionsbaustein **ow_feuchte_druck** berechnet

- **Luftfeuchtigkeit und den Taupunkt** (Sensor=1 oder 3)
- **Luftdruck** (Sensor=2)
- **Luftgüte** (Sensor=3)

bei Einsatz des Temperatur und Luftfeuchte Wohnraumsensors von

<http://www.eservice-online.de>.

Mit Parameter Sensor wird eingestellt was berechnet werden soll.

Sensor 1 -Luftfeuchte
 2 -Luftdruck
 3 -Luftfeuchte +Luftgüte

ow_feuchte_druck

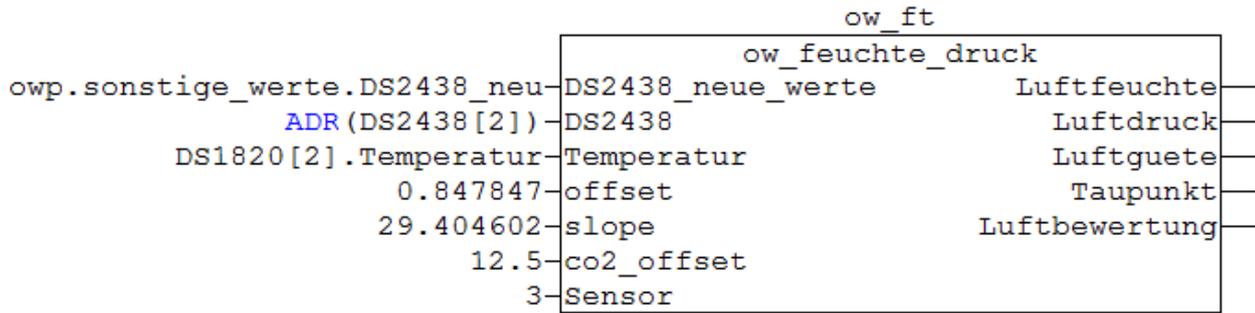
INPUT	DS2438_neue_werte	BOOL	neue Messwerte am DS2438
	DS2438	POINTER TO DS2438_V8	
	Temperatur	REAL	vom DS18B20
	offset	REAL	hier Daten aus dem Kalibrierdatenblatt eintragen [V]
	slope	REAL	hier Daten aus dem Kalibrierdatenblatt eintragen [mV/%RH]
	co2_offset:	REAL	Korrekturwerte Luftgütesensor
	Sensor:	INT	Einstellung was gemessen werden soll
OUTPUT	Luftfeuchte:	REAL	
	Luftdruck:	REAL	
	Luftgüte:	REAL	
	Taupunkt:	REAL	
	Luftbewertung:	STRING	Text der Luftbewertung

Am Eingang „DS2438_neue_werte“ wird signalisiert dass neue Messwerte des DS2438 vorhanden sind. Wenn die Variable „owp.sonstige_werte.DS2438_neu“ dort angeschlossen ist, wird die Berechnung nur durchgeführt wenn neue Werte mit dem DS2438 gemessen wurden. Dieser Eingang ist flankengesteuert, d.h. nur bei einer positiven Flanke wird der Baustein bearbeitet.

Der Ausgang „Luftbewertung“ gibt in Textform die Luftbewertung nachfolgend aus:

Luftgüte	Luftbewertung
<= 850	gut
>850 und <1200	ausreichend
>=1200 und <1600	schlecht
>1600	Lüften!

Aufruf des Bausteins: Messung [Luftfeuchte +Luftgüte](#)



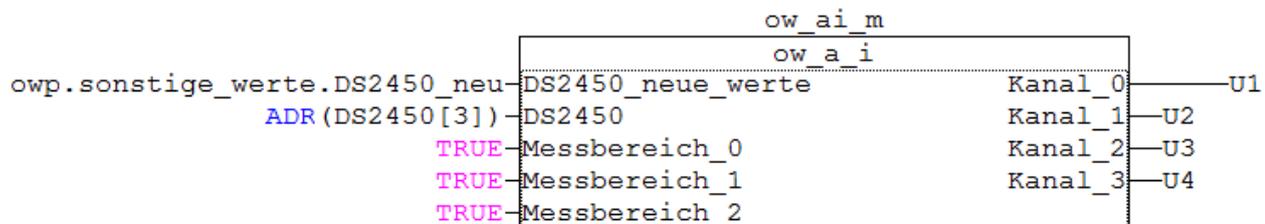
7.2 3-Fach Analog Input (ow_a_i)

Der Baustein ermöglicht das Messen von Spannungen bei Einsatz des 3-Fach Analog Inputmoduls von <http://www.eservice-online.de> mit einer Auflösung von 12Bit.

ow_a_i

INPUT	DS2450_neue_werte:	BOOL	(neue Messwerte am D2-Sensor)
	DS2450:	POINTER TO	DS2450_V8
	Messbereich_0	BOOL	Messbereich Kanal 0, False 0-5V/ True 0-10V
	Messbereich_1	BOOL	Messbereich Kanal 1, False 0-5V/ True 0-10V
	Messbereich_2	BOOL	Messbereich Kanal 2, False 0-5V/ True 0-10V
OUTPUT	Kanal 0	REAL	Messwert Kanal 0
	Kanal 1	REAL	Messwert Kanal 1
	Kanal 2	REAL	Messwert Kanal 2
	Kanal 3	REAL	Versorgungsspannung

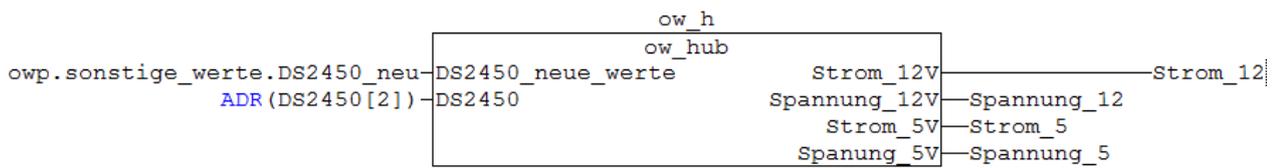
Die Messbereiche werden nur bei Start des Bausteins eingestellt. Wenn der Messbereich verändert wird, muss die Variable „init“ = FALSE sein, damit die geänderten Messbereichswerte übernommen werden.



7.3 Messwerte vom 1-Wire Hub (ow_hub)

Der Baustein ermöglicht das Messen von Spannung- und Strom bei Einsatz des 1-Wire Hub von <http://www.eservice-online.de>.

INPUT	DS2450_neue_werte	BOOL	neue Messwerte am D2-Sensor
	DS2450	POINTER TO	one_wire_D2_Sensor_V8
OUTPUT	Strom_12V	REAL	Strom der 12V Versorgungsspannung [mA]
	Spannung_12V	REAL	
	Strom_5V	REAL	Strom der 5V Versorgungsspannung [mA]
	Spannung_5V	REAL	



8. Hardware

Es gibt 2 Varianten für die Kopplung der SPS mit dem seriellen Buskoppler (DS2480B)

8.1 Anschluss an die interne Serviceschnittstelle

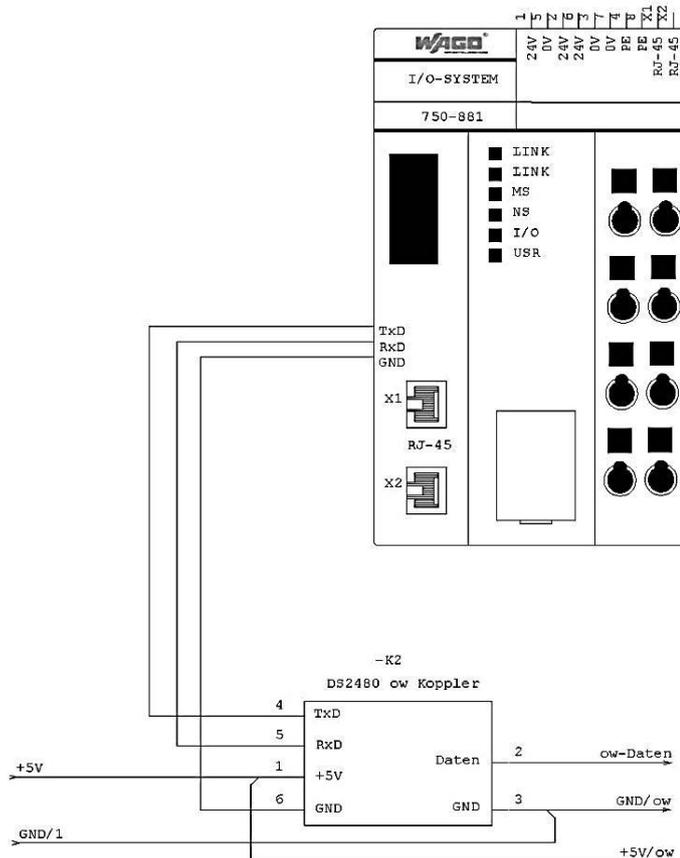
Der DS2480B (Serial to 1-Wire Line Driver) wird an die serielle Schnittstelle des Controllers angeschlossen, diese Schnittstelle arbeitet mit TTL-Signal (+5V).

Hierbei muss ein Buskoppler verwendet werden, welcher mit TTL-Pegel arbeitet z.B.: der **1-Wire Buskoppler SPS** von eservice-online.de oder ein vergleichbares Modell.

Die Anschlussbelegung der seriellen Schnittstelle (siehe Bild unten) ist folgendermaßen:

TxD
RxD
+5V
GND (Masse)

Weiterhin benötigt der DS2480 eine Versorgungsspannung von 5V.



Da diese Variante die interne Serviceschnittstelle verwendet und ein falscher Anschluss zur Zerstörung des Controllers führen kann ist Vorsicht geboten.

8.2 Verwendung einer seriellen Klemme 750-650

Der DS2480B (Serial to 1-Wire Line Driver) wird an eine serielle Klemme angeschlossen, diese Schnittstelle arbeitet mit RS232-Signal.

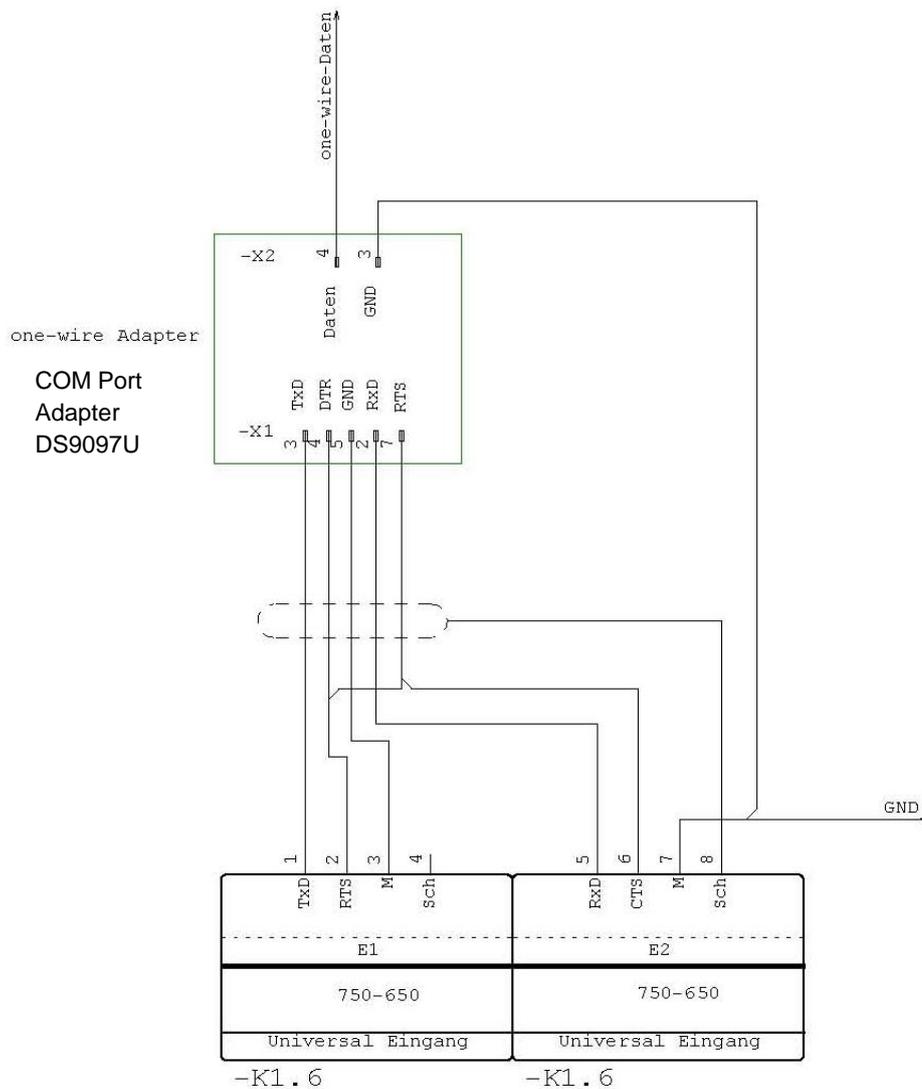
Hierbei muss ein Buskoppler verwendet werden, welcher mit RS232-Signal arbeitet!

z.B.: der **1-Wire Buskoppler mit RS232 (V24)** von eservice-online.de,
der **Universal 1-Wire COM Port Adapter DS9097U** (im Bild verwendet)
oder ein vergleichbares Modell

Die serielle Klemme muss folgende Eigenschaften besitzen:

- Übertragungsrate: 9600 Baud
- Parität: None
- Datenbits: 8
- Stopbits: 1

z.B.: 750/650, 750-650/000-001, 750-650/003-000



8.2.1 Einstellung der seriellen Klemme 750-650/003-000

Einstellung der seriellen Klemme 750-650/003-000 mit Wago WAGO-IO-Check

