

Freiprogrammierbare Solar- und Heizungsregelung UVR 1611

CANopen Kommunikation (Entwurf)

1 Einleitung

Die Regelung UVR 1611 besitzt 16 Hardwareeingänge die sowohl als Analog- als auch als Digitaleingänge konfiguriert werden können. Weiters besitzt sie bis zu 15 Ausgänge.

Die Gesamtfunktion wird durch die Verknüpfung einzelner Funktionsmodule bestimmt. Diese Funktionsmodule (**Tabelle 1**) werden in eine Funktionsliste eingetragen und können über Eingangs- und Ausgangsvariable miteinander verbunden werden. Jedes dieser Funktionsmodule besitzt einen Parametersatz, der die vom Anwender festgelegten Betriebsbedingungen und die Zuordnung der Eingangs- und Ausgangsvariablen enthält. Zusammen ergeben diese Parametersätze die Funktionsdaten.

Über CANopen sollen folgende Dienste durchgeführt werden können:

- Austausch von Variablen mit anderen Netzwerkknoten
- Zugriff auf Variablen und Parameter für Anzeige- und Einstellfunktionen
- Lesen und Schreiben von Funktionsdaten
- Softwareupdate

Im Folgenden sollen der Einfachheit halber nur die wesentlichen Punkte: Netzwerkmanagement, Netzwerkvariablen, und herstellerspezifische Objekte beschrieben werden.

Funktionsmodul
Heizkreisregelung
Heizungsanforderung
Warmwasseranforderung
Ladepumpe
Zirkulation
Solarregelung
Solarvorrang
PID-Regler
Vergleich
Logikfunktion
Analogfunktion
Wärmemengenzähler
Wartungsfunktion
Timer
Schaltuhr
Solarstarthilfsfunktion
Zähler
Solarrückkühlfunktion
Mischerregelung
Kontrollfunktion
Synchronisation
Profilfunktion
Kesselkaskade

Tabelle 1 Funktionsmodule

2 Netzwerkmanagement

2.1 Netzwerkstruktur

Da der verwendete Treiberbaustein maximal 64 CAN-Knoten treiben kann wird das Netzwerk auf 63 Teilnehmer (Knoten Nr. 1...63) und einen reservierten Teilnehmer (Knoten Nr. 0) für etwaige Master- bzw. Analysefunktionen begrenzt.

2.2 Bootup und Heartbeat

Die Regelung UVR 1611 sendet nach einem Reset eine Bootup-Nachricht und geht nach 10 Sekunden selbstständig in den Operational-Modus wenn es das Objekt 1F80 erlaubt. Die Heartbeatzeit ist mit 10 Sekunden fix eingestellt.

Für die geplanten Anzeige-, Erweiterungs- und Umsetzermodule ist ein ähnliches Verhalten vorgesehen.

Begründung: Netzwerk kann automatisch starten. Es ist kein Master notwendig. Ein etwaig vorhandener Master hat ausreichend Zeit die Kontrolle zu übernehmen. Die Heartbeatzeit ist mit 10 Sekunden für die geplanten Anwendungen ausreichend und hält die Buslast gering.

2.3 SDO Adressierung

2.3.1 SDO Struktur

Jeder der 63 Knoten besitzt einen Server-SDO (SSDO 1) der für eine Verbindung zum entsprechenden Client-SDO des Knoten 0 (CSDO 1...63) vorgesehen ist (Predefined-Master/Slave-Connection-Set).

Jeder der 63 Knoten besitzt einen Client-SDO (CSDO 1) mit einer Verbindung zum entsprechenden Server-SDO (SSDO 2...64) der anderen Knoten.

Jeder der 63 Knoten besitzt 62 zusätzliche Server-SDOs (SSDO 2...64) mit einer Verbindung zum Client-SDO (CSDO 1) des entsprechenden Knotens.

Es ist somit der CSDO 1 des Knoten N mit dem SSDO (N+1) der übrigen Knoten verbunden. Es obliegt dem Client die Verbindung mit dem gewünschten Knoten zu erlauben und alle anderen zu deaktivieren (siehe **2.3.2 Adressierung mittels MPDO**). Nach einem Reset sind alle Verbindungen deaktiviert.

SDO	Index	Subindex	Name	Wert
CSDO 1	1280	1	COB-ID Client-Server	640 + NC
		2	COB-ID Server-Client	5C0 + NC
		3	Node-ID Server	NS
SSDO 1	1200	1	COB-ID Client-Server	600 + NS
		2	COB-ID Server-Client	580 + NS
		3	Node-ID Client	0
SSDO 2...64	1201...123F	1	COB-ID Client-Server	640 + NC
		2	COB-ID Server-Client	5C0 + NC
		3	Node-ID Client	NC

Tabelle 2 SDO Zuordnung

NC: Knotennummer Client
NS: Knotennummer Server
Anmerkung: NC = NS wird ignoriert

Begründung: Jeder der 63 Knoten ist über Server SDO 1 vom Knoten 0 erreichbar. Weiters besteht die Möglichkeit, dass jeder Knoten auf jeden anderen zugreifen kann. Diese Möglichkeit ist besonders bei einem Netzwerk mit mehreren Reglern, Anzeige- und Umsetzermodule wichtig.

2.3.1.1 Geräteinterne SDO Struktur

Geräteintern sind tatsächlich nur 4 Server-SDOs vorhanden. Es können also nur 4 Clients gleichzeitig auf einen Knoten zugreifen. Einer dieser internen SSDOs ist für SSDO 1 reserviert. Die restlichen 3 werden bei *init upload/download* mit dem entsprechenden SSDO belegt. Ist bei *init upload/download* keiner dieser internen SSDOs mehr frei erfolgt eine Abbruchmeldung.

Eine weitere Einschränkung gilt für String- bzw. Anzeigeseitenobjekte. Da die entsprechende Ausgabefunktion aus Speicherplatzgründen nur einmal angelegt ist kann immer nur auf eines dieser Objekte zugegriffen werden. Ein Zugriff auf ein weiteres Objekt führt zu einer Abbruchmeldung.

2.3.2 Adressierung mittels MPDO

Um die Server-SDO Adressierung zu ermöglichen werden Multiplexor-PDOs (**Tabelle 3**) angelegt.

MPDO	COB - ID
TPDO 6	0400 + KnotenNr.
RPDO 33...95	0401...043F

Die Client- und Server-SDOs sind nach einem Reset deaktiviert. Will ein Client eine SDO-Verbindung zu einem Server aufnehmen sendet er per MPDO den Befehl *Request SDO* (**Bild 1**). Dieser Befehl kann entweder einen bestimmten Knoten oder alle Netzwerknoten (Broadcastadressierung) ansprechen. Der angesprochene Knoten vergleicht die im Befehl angegebene Server-Adresse mit seiner Knotennummer. Stimmen sie nicht überein deaktiviert er eine eventuell bestehende SDO-Verbindung zum entsprechenden Client. Stimmen sie überein wird der Server-SDO aktiviert und per MPDO die Freigabe für den Client-SDO gesendet (**Bild 2**).

Tabelle 3 Multiplexor- PDOs

63		48 47		40 39		32 31		24 23		8		7		6		0	
Index CSDO		Client		Server		Subindex		Index		F		Addr.		Adressierung			
1280		NC		NS		0		1F00		1		NS		Knoten			
1280		NC		NS		0		1F00		1		0		Broadcast			

Bild 1 Request SDO

NC: Knotennummer Client
NS: Knotennummer Server

63		62 43 42		32 31		24 23		8		7		6		0	
Freigabe		xxx		COB-ID C > S		Subindex		Index		F		Addr.			
0		0		640 + NC		1		1280		1		NC			

Bild 2 Freigabe Client

Die Deaktivierung der Verbindung erfolgt über den Befehl *Release SDO* (**Bild 3**). Empfängt ein Knoten diesen Befehl deaktiviert er die zugehörige SDO-Verbindung und löscht per MPDO die Freigabe für den Client-SDO (**Bild 4**).

Alternativ kann die Deaktivierung auch direkt über die SDO-Verbindung erfolgen. Während des Bestehens der Verbindung überwachen sich Client und Server über die Heartbeatfunktion. Im Fehlerfall wird die Verbindung deaktiviert.

63	48	47	40	39	32	31	24	23	8	7	6	0
Index CSDO	Client	Server	Subindex	Index	F	Addr.	Adressierung					
1280	NS	NS	0	1F01	1	NS	Knoten					

Bild 3 Release SDO

63	62	43	42	32	31	24	23	8	7	6	0
Freigabe	xxx	COB-ID C > S			Subindex	Index	F	Addr.			
1	0	640 + NC			1	1280	1	NC			

Bild 4 Löschen Freigabe Client

2.3.3 Zusätzliche Kontrollobjekte

Um einem eventuell vorhandenen Master die Möglichkeit zu bieten diesen Mechanismus zu blockieren und selbst die Kontrolle über die SDO-Adressierung zu übernehmen wird das **Objekt 1F02** (**Bild 5**) angelegt. Das **Objekt 1F03** (**Bild 6**) gibt Auskunft über eine bestehende SDO-Verbindung.

		31	30	29	11	10	0
Index	Subindex	V	U	Xxx	COB-ID		
1F02	1	0/1	0/1	0	640 + NC		
	2	0/1	0/1	0	5C0 + NC		

V: Freigabe COB-ID 0 = Knoten darf COB-ID verwenden
 1 = Knoten darf COB-ID nicht verwenden (SDO blockiert)
 U: COB-ID verwendet 0 = COB-ID momentan nicht verwendet
 1 = COB-ID momentan verwendet (SDO aktiv)

Bild 5 Object 1F02

		31	24	23	16	15	8	7	0
Index	Subindex	Client Offset	Client	Server Offset	Server				
1F03	1	0	NC	NC	NS				

Client-SDO Parameter Objekt = 1280 + Client Offset
 Server-SDO Parameter Objekt = 1200 + Server Offset

Bild 6 Objekt 1F03

3 Netzwerkvariablen

Der Einfachheit halber wird die Anwendung in 2 Komponenten zerlegt. Einerseits eine Funktion mit Eingangs- und Ausgangsvariablen, andererseits ein einfaches I/O-Modul.

Begründung: Die konkrete Funktion des Gerätes wird erst durch die sogenannten Funktionsdaten festgelegt. Diese legen auch fest, welche Eingangsvariablen die Funktion benötigt und welche Ausgangsvariablen sie bereitstellt. Da die Variablenzuordnung in der Funktion erfolgt kann das I/O-Modul sehr einfach sein. Dadurch ist relativ einfach auch eine primitive Kommunikation mit Fremdgeräten möglich (z.B. mit anderen I/O-Modulen).

3.1 Netzwerkausgänge

Das I/O-Modul besitzt 16 digitale und 16 analoge Eingänge. Damit kann es Ausgangsvariable der Funktion übernehmen und per PDO ans Netzwerk weiterleiten. Der Übertragungstyp ist herstellerspezifisch asynchron (d.h. die Funktion initiiert den Übertragungsvorgang). **Tabelle 4** zeigt die zugehörigen Objekte und PDO Parameter.

TPDO	Verwendung	COB - ID	TPDO mapping		Datentype
			Index	Subindex	
1	Digital NW Ausgang 1...16	0180 + NT	6100	1	UNSIGNED16
2	Analog NW Ausgang 1...4	0200 + NT	6401	1...4	INTEGER16
3	Analog NW Ausgang 5...8	0280 + NT	6401	5...8	INTEGER16
4	Analog NW Ausgang 9...12	0300 + NT	6401	9...12	INTEGER16
5	Analog NW Ausgang 13...16	0380 + NT	6401	13...16	INTEGER16

Tabelle 4 Netzwerkausgänge NT: Knotennummer

3.2 Netzwerkeingänge

Das I/O-Modul besitzt 16 digitale und 16 analoge Ausgänge. Es kann Netzwerkvariable übernehmen und an die Funktion als Eingangsvariable weitergeben. Die Funktion legt fest welche Netzwerkvariable benötigt werden. Die Adressierung erfolgt über die Knotennummer und die NW Ausgangsnummer des Senders. Pro RPDO kann nur ein Ausgang gemapped werden, der Rest sind Dummy Einträge. **Tabelle 5** zeigt die zugehörigen Objekte und PDO Parameter.

RPDO	Verwendung	COB - ID	RPDO mapping		Datentype
			Index	Subindex	
1...16	Digital NW Eingang 1...16	0180 + NT	6220	1...16	BOOLEAN
17...20	Analog NW Eingang 1...4	0200 + NA	6411	1...4	INTEGER16
21...24	Analog NW Eingang 5...8	0200 + NA	6411	5...8	INTEGER16
25...28	Analog NW Eingang 9...12	0200 + NA	6411	9...12	INTEGER16
29...32	Analog NW Eingang 13...16	0200 + NA	6411	13...16	INTEGER16

Tabelle 5 Netzwerkeingänge NT: Knotennummer des Senders
NA: NT + 80(hex) * int((Ausgangsnummer-1) / 4)

4 Herstellerspezifische Objekte

4.1 Objektverzeichnis

Anhang 1 zeigt eine Übersicht des herstellerspezifischen Bereichs im Objektverzeichnis. Von besonderem Interesse ist der Bereich von 5300 bis 57FF wo die funktionsbezogenen Objekte angelegt sind. Derzeit können in der Regelung UVR1611 maximal 40 Funktionen in die Funktionsliste eingetragen werden. Deshalb wird jedes funktionsbezogene Objekt als ARRAY mit höchstem Subindex 40 angelegt. Der jeweilige Subindex ist nur belegt wenn an der entsprechenden Stelle in der Funktionsliste auch die zugehörige Funktion eingetragen ist. Bei einem Zugriffsversuch auf einen nicht belegten Subindex wird eine Abbruchmeldung ausgegeben. Als Beispiel siehe **Tabelle 6**.

Funktion		Objekt 3283	
Nr.		Subindex	
		0	40
1	Heizkreisregelung	1	unbenutzt
2	Heizkreisregelung	2	unbenutzt
3	Solarregelung	3	maximalwert Kollektortemperatur
4	Schaltuhr	4	unbenutzt
...

Tabelle 6 Objekt 3283

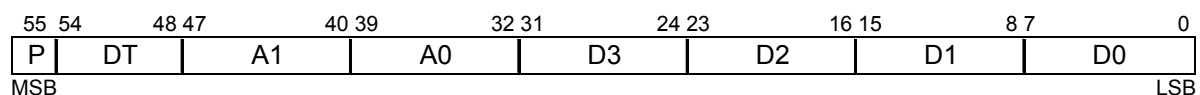
Anhang 2 zeigt als Beispiel das Objektverzeichnis der Funktion *Solarregelung*.

4.2 Datentypen

Im wesentlichen wird unterschieden zwischen Parametern, Variablen, Strings, Anzeigeseiten und Funktionsdaten.

4.2.1 Parameter und Variablen

Für Parameter und Variable wurde generell die Datentype UNSIGNED56 gewählt. Die Datenstruktur ist in **Bild 7** dargestellt



P : Variable/Parameter
DT : strukturinterne Datentype
A0, A1 : Information zur Objektdarstellung
D0...D3 : Daten

Bild 7 Datenstruktur Parameter und Variablen

Begründung: Die Daten sind geräteintern in einer ähnlichen Struktur abgelegt, dadurch kann der Softwareaufwand gering gehalten werden. 7 Byte enthalten einerseits alle wesentlichen Informationen andererseits können sie innerhalb eines SDO Rahmens übertragen werden.

4.2.1.1 Variablen

Variablen sind Ergebnisse interner Berechnungen und können nur gelesen werden. Je nach Datentype können D1, D2, D3 unbenutzt sein. (**Bild 8**)

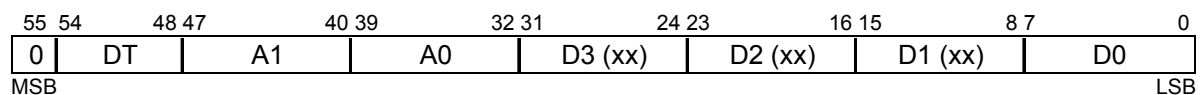


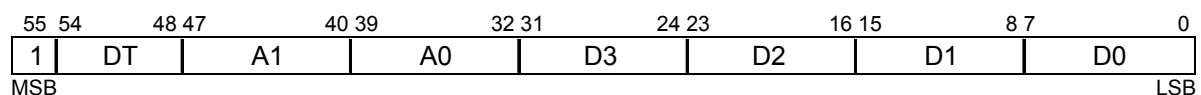
Bild 8 Datenstruktur Variablen

4.2.1.2 Parameter

Parameter sind Werte, die vom Anwender festgelegt werden. Sowohl lesen als auch schreiben ist möglich. Da die Datenbreite maximal 8 Bit beträgt sind strukturintern die Datentypen INTEGER und LONG INTEGER nicht möglich.

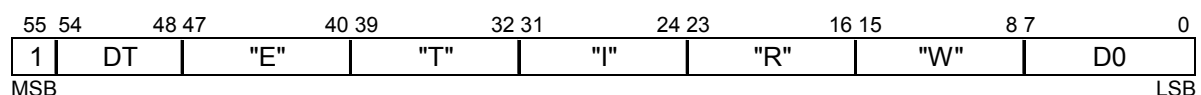
Begründung: Dadurch können auch der Wertebereich und der Defaultwert übertragen werden. Weiters musste geräteintern auf möglichst geringen Speicherverbrauch geachtet werden.

Datenstruktur Lesen siehe **Bild 9**. Datenstruktur Schreiben siehe **Bild 10**.



D0: Parameterwert
D1: maximaler Parameterwert
D2: minimaler Parameterwert
D3: Defaultwert

Bild 9 Datenstruktur Parameter lesen



D0: Parameterwert
W R I T E (ASCII codiert)

Bild 10 Datenstruktur Parameter schreiben

Begründung: Als zusätzlicher Schreibschutz muss der ASCII Code WRITE und die korrekte Datentype übertragen werden.

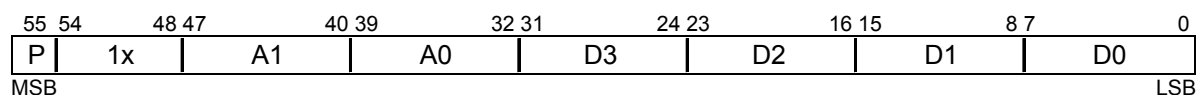
4.2.1.3 Strukturinterne Datentypen (siehe Tabelle 7)

DT (Hex)	Datentype	Datenbreite	Bemerkung
1x	Zeiger auf String	8 Bit	
2x	Bitfeld	8 Bit	
3x	Character	8 Bit	
4x	Integer	16 Bit	nicht für Parameter
5x	long Integer	32 Bit	nicht für Parameter

Tabelle 7 strukturinterne Datentypen

4.2.1.3.1 Zeiger auf String

Das Datenbyte D0 zeigt auf einen String (Subindex) in einer Stringtabelle (Index) im Objektverzeichnis.



A1: Index MSB
A0: Index LSB
D0: Subindex
x: nur geräteintern verwendet

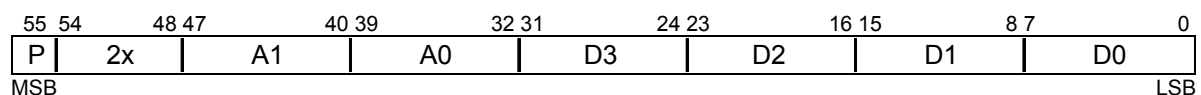
Bild 11 Datenstruktur Zeiger auf String

4.2.1.3.2 Bitfeld

Bitfelder sind logisch zusammenhängende 1 Bit Daten (z.B.: Wochentagszuordnung im Zeitprogramm) mit einer möglichen Datenbreite von 1 bis 8 Bit.

Zur Anzeige ist jedem Bit ein String zugeordnet, der je nach Status entsprechend markiert wird (z.B. : 0 = normale Anzeige, 1 = inverse Anzeige).

Der String (Subindex = Bitnummer + 1) steht in einer Stringtabelle (Index) im Objektverzeichnis.

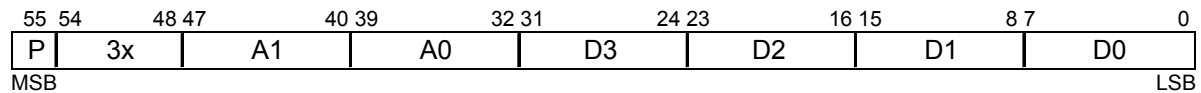


A1: Stringtabelle Index MSB
A0: Index LSB
D0: Daten Bitfeld b0... b(x-1)
x: Datenbreite

Bild 12 Datenstruktur Bitfeld

4.2.1.3.3 Character

Unter der Datentype Character sind 8 Bit Daten zusammengefasst, die aufgrund des LSD von DT (x) unterschiedlich interpretiert werden. Eine Übersicht zeigt **Tabelle 8**. Zusätzlich werden im Datenrahmen noch Informationen für die Anzeige (angezeigte Stellen, physikalische Einheit) übertragen. Die physikalische Einheit steht in einer Stringtabelle (Index = 5002).



A1: (physikalische) Einheit (Subindex zu Objekt 5002)
A0 MSD: Anzahl der angezeigten Stellen (inklusive unterdrückter Nullen)
A0 LSD: Anzahl der Stellen hinter dem Komma
D0: Datenbyte
x: Interpretation

Bild 13 Datenstruktur Character

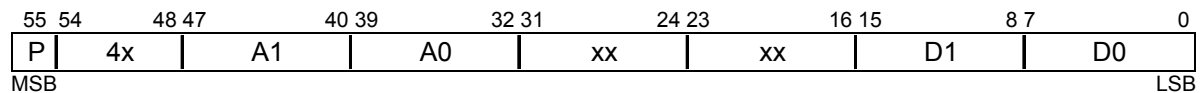
DT (hex)	Bereich	Interpretation	Bemerkung
30	0...255	D0	interne Stellenwertber.
31	0...255	D0	
32	0 1...255	"AUS" D0	
33	0...255	D0	keine Nullunterdrückung
34	0...25	D0	Anz. als ASCII Zeichen D0 + 41(hex)
35	0...255	D0 * 5	
36	0... 90 91...107 108...157 158...251	D0 (D0 - 87) * 30 (D0 - 97) * 60 (D0 - 155) * 1800	nichtlineare Darstellung Zeit
37	0...100 101...115	D0 (D0 - 90) * 10	nichtlineare Darstellung Schaltdifferenz
38	0...143	D0	Uhrzeit in 10 Minuten
39	0...255	D0	Datum + D0 Tage
3A	0...255	D0 * 10	
3B	0...100 101...115 116...127 128...140 141...155 156...255	D0 (D0 - 90) * 10 (D0 - 110) * 50 (D0 - 146) * 50 (D0 - 166) * 10 D0 - 256	nichtlineare Darstellung +/- Schaltdifferenz
3C	0...200 201...255	D0 D0 - 256	nichtlineare Darstellung +/- Sollwert
3D	0...255	D0 * 100	
3E	0...255	D0 * 50	

Tabelle 8 Unterteilung Datentype Character

Manche Interpretationen sind durch Aufteilung in Bereiche nichtlinear (siehe **Tabelle 8**).

Begründung: Einerseits müssen 8 Bit Datenbreite ausreichen, andererseits erleichtert die Nichtlinearität auch dem Anwender das Einstellen der Parameter.

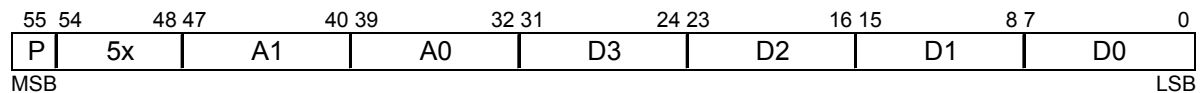
4.2.1.3.4 Integer



A1: (physikalische) Einheit (Subindex zu Objekt 5002)
A0 MSD: Anzahl der angezeigten Stellen (inklusive unterdrückter Nullen)
A0 LSD: Anzahl der Stellen hinter dem Komma
xx: unbenutzt
D0, D1: Daten
x: geräteintern verwendet

Bild 14 Datenstruktur Integer

4.2.1.3.5 long Integer



A1: (physikalische) Einheit (Subindex zu Objekt 5002)
A0 MSD: Anzahl der angezeigten Stellen (inklusive unterdrückter Nullen)
A0 LSD: Anzahl der Stellen hinter dem Komma
xx: unbenutzt
D0...D3: Daten
x: geräteintern verwendet

Bild 15 Datenstruktur long Integer

4.2.2 Strings

Strings sind in Stringtabellen (ab Index 5000) angelegt. Der Index adressiert die Stringtabelle, der Subindex den entsprechenden String. Damit werden Texte für Anzeigemodule zur Verfügung gestellt. Strings können beliebige Länge haben. Das Stringende ist durch 00 (hex) markiert. Strings sind vorerst nur lesbar. Mögliche Datentypen `VISIBLE_STRING`, `OCTET_STRING`, `DOMAIN` (???)

4.2.3 Anzeigeseiten

Damit soll Anzeigemodulen die Möglichkeit gegeben werden möglichst unabhängig von Software- und Geräteversionen auf Anzeigedaten zuzugreifen. Anzeigeseiten bestehen aus Strings und Befehlen. Die Darstellung der Befehle ist derzeit noch in Überlegung. Die Möglichkeiten gehen von der Übernahme des geräteinternen Befehlformats bis zu HTML. Anzeigeseiten sind vorerst nur lesbar. Die Datentype ist DOMAIN.

4.2.4 Funktionsdaten

Die Funktionsdaten sind eine Datei, welche die Funktion des Gerätes bestimmt. Funktionsdaten sind les- und beschreibbar. Die Datentype ist DOMAIN.

ANHANG 1 Objektverzeichnis Übersicht herstellerspezifischer Bereich

von	bis	Objectgruppe	verwendet	reserve
2000	20F0	Benutzer	0	16
2010	201F	Zeit	8	8
2020	205F	Zeitprogramm	50	14
2060	207F	Reserve		32
2080	209F	Eingang	12	20
20A0	20BF	Ausgang	12	20
20C0	20CF	Analogausgang	10	6
20D0	20FF	Reserve		48
2100	217F	Netzwerk	4	124
2180	21FF	Netzwerkeingang	15	113
2200	227F	Netzwerkausgang	6	122
2280	22FF	Opferanode	13	115
2300	237F	Antiblockierschutz	5	123
2380	23FF	Reserve		128
2400	24FF	Meldung	38	90
2500	2EFF	Reserve		2560
2F00	2FFF	Funktion allgemein	3	253
3000	307F	Heizkreisregelung	76	52
3080	30FF	Heizungsanforderung	33	95
3100	317F	Warmwasseranforderung	30	98
3180	31FF	Ladepumpe	31	97
3200	327F	Zirkulation	37	91
3280	32FF	Solarregelung	30	98
3300	337F	Solarvorrang	46	82
3380	33FF	PID-Regler	49	79
3400	347F	Vergleich	24	104
3480	34FF	Logikfunktion	45	83
3500	357F	Analogfunktion	39	89
3580	35FF	Wärmemengenzähler	32	96
3600	367F	Wartungsfunktion	20	108
3680	36FF	Timer	27	101
3700	377F	Schaltuhr	23	105
3780	37FF	Solarstarthilfsfunktion	38	90
3800	387F	Zähler	46	82
3880	38FF	Solarrückkühlfunktion	23	105
3900	397F	Mischerregelung	19	109
3980	39FF	Kontrollfunktion	31	97
3A00	3A7F	Synchronisation	33	95
3A80	3AFF	Profilfunktion	84	44
3B00	3B7F	Kesselkaskade	108	20
3B80	4FFF	Reserve		5284

von	bis	Objectgruppe	verwendet	reserve
5000	500F	Strings (Global)	9	7
5010	501F	Strings (Zeit)	3	13
5020	502F	Strings (Zeitprogramm)	2	14
5030	507F	Reserve		80
5080	509F	Strings (Eingang)	19	13
50A0	50BF	Strings (Ausgang)	12	20
50C0	50FF	Reserve		64
5100	5127	Strings (Netzwerk)	1	39
5128	512F	Strings (Opferanode)	1	7
5130	5137	Strings (Antiblockierschutz)		8
5138	513F	Reserve		8
5140	514F	Strings(Meldung)	3	13
5150	52FF	Reserve		432
5300	530F	Heizkreisregelung	8	8
5310	531F	Heizungsanforderung	3	13
5320	532F	Warmwasseranforderung	3	13
5330	533F	Ladepumpe	3	13
5340	534F	Zirkulation	4	12
5350	535F	Solarregelung	3	13
5360	536F	Solarvorrang	3	13
5370	537F	PID-Regler	5	11
5380	538F	Vergleich	4	12
5390	539F	Logikfunktion	4	12
53A0	53AF	Analogfunktion	4	12
53B0	53BF	Wärmemengenzähler	4	12
53C0	53CF	Wartungsfunktion	3	13
53D0	53DF	Timer	5	11
53E0	53EF	Schaltuhr	3	13
53F0	53FF	Solarstarthilfsfunktion	3	13
5400	540F	Zähler	4	12
5410	541F	Solarrückkühlfunktion	3	13
5420	542F	Mischerregelung	3	13
5430	543F	Kontrollfunktion	5	11
5440	544F	Synchronisation	4	12
5450	545F	Profilfunktion	3	13
5460	546F	Kesselkaskade	3	13
5470	57FF	Reserve		912
5800	5FFF	Anzeigeseiten		2048

ANHANG 2 Objektverzeichnis Funktion Solarregelung

INDEX (hex)	Subind	Object	Name	Type	Wert (hex)	Acces	Bemerkung
3281	1...40 (dez)	ARRAY	Einschaltdifferenz	UNSIGNED56	BB 07 31 50 81 7F PP	rw	-85...85 K (8,0)
3282		ARRAY	Ausschaltdifferenz	UNSIGNED56	BB 07 31 UP 80 UP PP	rw	-90...[3281]-h ([3281]-h) h: 0,1...5 K
3283		ARRAY	Maximalwert Kollektortemperatur	UNSIGNED56	B0 01 30 83 5A C8 PP	rw	90...200 °C (130)
3284		ARRAY	Hysterese max. Koll.temp.	UNSIGNED56	B7 07 21 50 05 73 PP	rw	0,5...25,0 K (8,0)
3285		ARRAY	Maximalbegrenzung 1	UNSIGNED56	B0 01 20 5A 00 63 PP	rw	0...99 °C (90)
3286		ARRAY	Hysterese Maximalbegrenzung 1	UNSIGNED56	B7 07 21 50 05 73 PP	rw	0,5...25,0 K (8,0)
3287		ARRAY	Maximalbegrenzung 2	UNSIGNED56	B0 01 20 5A 00 63 PP	rw	0...99 °C (90)
3288		ARRAY	Hysterese Maximalbegrenzung 2	UNSIGNED56	B7 07 21 50 05 73 PP	rw	0,5...25,0 K (8,0)
3289		ARRAY	Ausgangszuordnung	UNSIGNED56	A8 ST OB 00 00 01 PP	rw	Bitfeld (8) Ausgangszuordnung 1...8
328A		ARRAY	Ausgangszuordnung	UNSIGNED56	A6 ST OB 00 00 01 PP	rw	Bitfeld (6) Ausgangszuordnung 9...14
328B		ARRAY	Ausgangszuordnung	UNSIGNED56	A1 ST OB 00 00 01 PP	rw	Bitfeld (1) Ausgangszuordnung 15
328C		ARRAY	Modus Freigabe Solarkreis	UNSIGNED56	90 50 08 01 01 02 PP	rw	normal; invers (normal)
328D		ARRAY	Quelle Freigabe Solarkreis	UNSIGNED56	90 50 07 01 01 64 PP	rw	Interpretation = Quelle
328E		ARRAY	Adresse Freigabe Solarkreis	UNSIGNED56	B0 00 20 DP LP UP PP	rw	Interpretation = Adresse
328F		ARRAY	Bezeichnung Freigabe Solarkreis	VIS-STRING		ro	
3290		ARRAY	Status Freigabe Solarkreis	UNSIGNED56	90 50 A3 01 01 02 PP	rw	aus; ein (ein) write nur wenn [328D] = Benutzer
3291		ARRAY	Quelle Kollektortemperatur	UNSIGNED56	90 50 07 02 02 64 PP	rw	Interpretation = Quelle
3292		ARRAY	Adresse Kollektortemperatur	UNSIGNED56	B0 00 20 DP LP UP PP	rw	Interpretation = Adresse
3293		ARRAY	Bezeichnung Kollektortemperatur	VIS-STRING		ro	
3294		ARRAY	Wert Kollektortemperatur	UNSIGNED56	40 01 41 xx xx VV VV	ro	in 0,1 °C
3295		ARRAY	Quelle Referenztemperatur	UNSIGNED56	90 50 07 02 02 64 PP	rw	Interpretation = Quelle
3296		ARRAY	Adresse Referenztemperatur	UNSIGNED56	B0 00 20 DP LP UP PP	rw	Interpretation = Adresse
3297		ARRAY	Bezeichnung Referenztemperatur	VIS-STRING		ro	
3298		ARRAY	Wert Referenztemperatur	UNSIGNED56	40 01 41 xx xx VV VV	ro	in 0,1 °C
3299		ARRAY	Quelle Begrenzungstemperatur	UNSIGNED56	90 50 07 02 01 64 PP	rw	Interpretation = Quelle
329A		ARRAY	Adresse Begrenzungstemperatur	UNSIGNED56	B0 00 20 DP LP UP PP	rw	Interpretation = Adresse
329B		ARRAY	Bezeichnung Begrenzungstemperatur	VIS-STRING		ro	
329C		ARRAY	Wert Begrenzungstemperatur	UNSIGNED56	40 01 41 xx xx VV VV	ro	in 0,1°C
329D		ARRAY	Status Solarkreis	UNSIGNED56	10 50 A3 xx xx VV VV	ro	aus; ein
329E		ARRAY	Status Maximalbegrenzungen	UNSIGNED56	10 50 A3 xx xx VV VV	ro	aus; ein