

Die DigiWEB

Scriptsprache und Registerbelegung



Digitronic Automationsanlagen GmbH

Auf der Langwies 1 · D - 65510 Hünstetten-Wallbach · Tel. +49 6126 9453-0 · Fax -42
Internet: <http://www.digitronic.com> · E-Mail: mail@digitronic.com

Zur Beachtung

Dieses Handbuch entspricht dem Stand der Software 08.2019, ab WEB-Server Software Version: 4.109 für DigiWEB, CamCon DC190, DigiENERGY und bluebox. Die Firma Digitronic Automationsanlagen GmbH behält sich vor, Änderungen, welche eine Verbesserung der Qualität oder der Funktionalität zur Folge haben, jederzeit ohne Vorankündigung durchzuführen. Die Bedienungsanleitung wurde mit größtmöglicher Sorgfalt erstellt, dennoch können Fehler nicht ausgeschlossen werden. Für Hinweise, die eventuelle Fehler in der Bedienungsanleitung betreffen, sind wir dankbar.

Update

Sie erhalten dieses Handbuch auch im Internet unter <http://www.digitronic.com> in der neuesten Version als PDF Datei.

Qualifiziertes Personal

Inbetriebnahme und Betrieb der Software dürfen nur von qualifiziertem Personal vorgenommen werden.

Haftung

(1) Der Verkäufer haftet für von ihm oder dem Rechtsinhaber zu vertretende Schäden bis zur Höhe des Verkaufspreises. Eine Haftung für entgangenen Gewinn, ausgebliebene Einsparungen, mittelbare Schäden und Folgeschäden ist ausgeschlossen.

(2) Die obigen Haftungsbeschränkungen gelten nicht für zugesicherte Eigenschaften und Schäden, die auf Vorsatz oder grober Fahrlässigkeit beruhen.

Schutz

Diese Software und dieses Handbuch sind urheberrechtlich geschützt. Alle Rechte sind vorbehalten. Weder die Software, noch dieses Dokument, dürfen in Teilen oder im Ganzen kopiert, fotokopiert, reproduziert, übersetzt oder übertragen werden auf irgendwelche elektronische Medien oder maschinenlesbare Formen, ohne vorherige schriftliche Genehmigung durch die Firma Digitronic Automationsanlagen GmbH.

Hinweis: CamCon, bluebox und DigiENERGY sind eingetragene Markenzeichen der Firma Digitronic Automationsanlagen GmbH.

Hinweis: Das Gerät erfüllt die Normen: DIN EN 61000-6-2, DIN EN 61000-4-2, DIN EN 61000-4-4, DIN EN 61000-4-5, DIN EN 61000-4-8 und DIN EN 55011/DIN EN sowie RoHS 3.



(c) Copyright 1992 - 2022 / Datei: DWSCRIPT.DOC

Digitronic Automationsanlagen GmbH
Auf der Langwies 1
D-65510 Hünstetten - Wallbach
Tel. (+49)6126/9453-0 Fax. (+49)6126/9453-42
Internet: <http://www.digitronic.com>
E-Mail: mail@digitronic.com

Inhaltsverzeichnis

1. Einleitung	5
2. Allgemeines zum Syntax und Editor	5
2.1. Ausdrücke bzw. Expression	5
2.1.1. Adressierung für Datenbausteine (absolut)	5
2.1.2. Adressierung für Datenbausteine mit Gerätenummer (absolut)	6
2.1.3. Symbolische Adressierung.....	6
2.1.4. Adressierung für Datenaustausch zwischen DigiWEB und DigiWEB via Netzwerk	7
2.1.4.1. Absolute Adressierung	7
2.1.4.2. Symbolische Adressierung.....	7
2.1.5. Adressierung im S7/S5 Format.....	8
2.1.5.1. Adressierung für Datenbausteine	8
2.1.5.2. Adressierung für Ein - Ausgänge und Merker.....	8
2.1.5.3. Adressierung für Zähler	8
2.1.5.4. Adressierung für Zeiten (Timer).....	8
2.1.6. Indirekte Adressierung.....	9
2.2. Operatoren.....	9
2.2.1. Arithmetische Operatoren.....	9
2.2.1.1. Berechnungen > 32 Bit	9
2.2.2. Boolesche Operatoren	9
2.2.3. Logische Operatoren	10
2.2.4. String - bzw. Zeichenkettenoperatoren	11
2.2.4.1. String - bzw. Zeichenkettenlänge ermitteln	12
2.3. Lokale Variablen	12
2.3.1. Lokale Variablen als Parameter übergeben.....	12
2.3.2. Variablenfunktion "##defined()".....	12
2.4. Ablaufsteuerung.....	13
2.4.1. Die "#if" Bedingung	13
2.4.2. Die "#while" Schleife	13
2.4.3. Die "#until" Schleife	14
2.4.4. Die "#include" Funktion.....	14
3. Die DigiWEB Funktionen	15
3.1. Die Funktionen der DigiWEB SQL Datenbank	17
3.2. Zusätzliche Funktionen ab der DigiWEB 2 Hardware und höher.....	18
3.2.1. Allgemeine Funktionen	18
3.2.2. Netzwerk Sendempfangs - Funktionen für TCPIP	19
3.2.3. Netzwerk Sendempfangs - Funktionen für UDP	19
3.2.4. Funktionen zur Steuerung der serielle Schnittstelle.....	20
3.2.5. IoBUS bzw. DBUS - Funktionen des DigiWEB 3 und 4	20
3.2.6. Die MODBUS Funktionen	21
3.2.6.1. Erweiterung Parameter 'Port' bei MODBUS TCP Port - Nummer	21
3.2.6.2. Erweiterung Parameter 'Port/Zieladresse' bei MODBUS RTU mit DigiWEB 4 und 4.5.....	22
3.2.6.3. Erweiterung Parameter 'Reg' bei MODBUS RTU/TCP.....	22
3.2.6.3.1. Anzahl der maximal gelesenen Register	22
3.2.7. Die Exchange Funktionen bzw. Variablen austausch.....	23
3.2.7.1. Die ##ExchangeGet() Funktion zum Variablen austausch	24
3.2.8. Die PLC (SPS) Funktionen	25
3.2.9. Die Debug Funktionen	26

4. Die Interne Datentabelle bzw. die Register/Variablen des DigiWEB	27
4.1. DigiWEB Applikation Variablen bzw. Symbole.....	27
4.1.1. DigiWEB App - Variables in der erweiterten Konfiguration	28
4.2. Die Netzwerkkonfiguration, Systemvariablen und Systempasswort	29
4.3. Status Flash - Daten - Speicher	30
4.4. Zeit/Uhrzeit und NTP Serverkonfiguration	31
4.5. E-Mail - Server und Event - Händlerkonfiguration	31
4.6. Die Merker des DigiWEB	31
4.7. Die DigiWEB Ein - Ausgangs Schnittstelle	32
4.7.1. Die Puls - Weiten - Modulation (PWM) Funktion der Ausgänge.....	33
4.7.2. Die Impulzzähler - und SSI - Encoder Eingänge - des DigiWEB.....	34
4.7.2.1. Einstellung der Geschwindigkeitsanzeige.....	36
4.8. Konfiguration der Seriellen - und der MPI bzw. Profibus - Schnittstelle.....	37
4.8.1. Status der ColorNet Schnittstelle	39
4.9. Auto Update Funktion	39
4.10. "Easy Access" Funktion	40
4.11. Die zusätzlichen Register des DigiWEB 3	41
4.11.1. DigiWEB 3 BUS Ausbaustufe	41
4.11.2. Die Register des DigiWEB 3 Modul: X - Die digitale Ein - Ausgangskarte	41
4.11.3. Die Register des DigiWEB 3 Modul: A + V - Die analogen Eingangskarten.....	42
4.11.3.1. Modul: A + V - Abgleichen.....	42
4.11.4. Die Register des DigiWEB 3 Modul: T - Die PT1000 Temperaturfühler Eingangskarte.....	43
4.11.4.1. Modul: T abgleichen bzw. kalibrieren.....	43
4.12. Die zusätzliche Funktionen und Register des DigiWEB 4	44
4.12.1. Funktionen	44
4.12.2. Die Register des DigiWEB 4	44
5. Beispiele für "REQ" - Scriptdateien.....	45
5.1. Beispiele zur Dateiverwaltung.....	48
5.2. Beispiel zur Datenbank	49
6. Fehlermeldungen bzw. Error Codes	50
7. Script - bzw. REQ - Dateien in der DigiVISU	52
7.1. Rückgabe der Script - Ausgaben an DigiVISU	52
7.2. Parameter beim Schreiben durch ein Script	52
7.3. Locale Variable der DigiVISU dem Script als Parameter übergeben.....	52
8. Die "TXH" - Scriptdatei.....	53
8.1. Backup und Restore durch die TXH - Scriptdatei	53
9. Die seriellen Protokolle für den Softwareentwickler	54
9.1. Das 3964(R) Protokoll.....	54
9.2. Das RK512 Protokoll.....	55
10. Stichwortverzeichnis	56

1. Einleitung

Um komplexe Funktionen bzw. Programme verarbeiten zu können, kann das DigiWEB Script - Programme ausführen (z.B. ein Datenlogger). Diese Scripte haben im Dateisystem die Erweiterung ".req" und können zyklisch über die Event - Steuerung oder auch durch die DigiVISU - WEB Oberfläche aufgerufen werden (Hinweis: Auch eine HTML oder XML Datei kann ein Script enthalten).

Zum Testen und Debuggen kann eine REQ - bzw. Scriptdatei auch durch Aufrufen der Datei im Browser ausgeführt und das Ergebnis zum Debuggen angezeigt werden (z.B. "<http://DigiWEB-IP/test.req>").

2. Allgemeines zum Syntax und Editor

Alle Eingaben in einer Scriptdatei, die nicht in einem HTML - Kommentar "<!-- -->" stehen und kein einleitendes '#' - Zeichen enthalten, werden bei der Script - Verarbeitung ignoriert bzw. werden als Kommentar betrachtet. Eine Anweisung enthält also immer ein Kommentar und min. ein '#' - Zeichen.

Zur Steuerung des Programmablaufs ist eine Anweisung mit einer '#' erforderlich (z.B. <!--#if...-->). Zum Lesen oder Schreiben einer Variable muß die Adresse bzw. der Variablenname immer mit den Zeichen "##" (Zahl) oder "#\$" (String) beginnen (z.B. <!--##000256:=5-->). Muß oder soll der Wert einer Variablen zum Debuggen ausgegeben werden, so muß direkt hinter dem HTML - Kommentar ein Platzhalter definiert werden das zur Laufzeit ersetzt wird. Eine '#' steht hierbei für ein Zeichen. Die Zeichen '#' stehen für beliebig viele Zeichen (z.B. <!--##000256-->#*).

Zum Bearbeiten der REQ - Scripte können Sie in der DigiWEB "**advanced**" Konfiguration "<http://DigiWEB-IP/config>" unter dem Menüpunkt: "**File Commander**" neue Scripte (New file) erzeugen und diese dann durch Doppelklick im Text - Editor öffnen, bearbeiten und testen. Der Editor kann auch in der Event - Steuerung aufgerufen werden, wenn im Eingabefeld "Command" ein REQ - Script - Dateiname steht und der Name der Datei doppelt angeklickt wird.

2.1. Ausdrücke bzw. Expression

Die Kombination von Adressen bzw. Datenworten, Operatoren und Bedingungen (logische Operatoren) wird auch als **Ausdruck** oder **Expression** bezeichnet:

<!--##000256:=5+(3*##000256)--> oder <!--##DB50.DBW10:=5+(3*##MW6)-->

<!--##000256:=(##000187b10 ? 1 : 2 --> (z.B. mit logischem Operator)

Achtung: Je nach eingestellter "**RK512 Cell Size**" liefert das DigiWEB bei einer Zahl ("##") 1, 2 oder 4 Byte je Variable bzw. Adresse. Sehen Sie hierzu die DigiWEB "**advanced**" Konfiguration "<http://DigiWEB-IP/config>" unter dem Menüpunkt: "**Serial/MPI/PB**".

Nachfolgend werden nun die einzelnen Adressierungsarten, Operatoren und Bedingungen vorgestellt:

2.1.1. Adressierung für Datenbausteine (absolut)

Adressierung: **##XXYYYY** Es werden 8, 16 oder 32 Bit mit Vorzeichen gelesen/geschrieben.
einer Zelle. z.B. ##000187 XXX = dezimale Nummer des Datenbausteins.
YYY = dezimale Nummer des Datenwortes.
##XXYYYYb Es werden 8, 16 o. 32 Bit ohne Vorzeichen gelesen/geschrieben.
z.B. ##000187b

Adressierung: **##XXYYYYbnn** Es wird ein Bit gelesen/geschrieben.
eines Bits. z.B. ##000187b11 nn = dezimale Bitnummer.

Adressierung: **##XXYYYYbnn_zz** Es werden mehrere Bits ohne Vorzeichen gelesen/geschrieben.
mehrere Bits. z.B. ##000187b1_2 nn = dezimale Bitnummer des ersten Bits.
zz = dezimale Bitnummer des letzten Bits

##XXYYYYsnn_zz Es werden mehrere Bits mit Vorzeichen gelesen/geschrieben.
z.B. ##000187s1_2 Hier wird das letzte Bit (2) als Vorzeichen verwendet.

Hinweis: Wird bei einem Bitzugriff der Bereich der eingestellten "**RK512 Cell Size**" verlassen, werden entsprechend der Anzahl und der Position der Bits mehrere Zellen gelesen bzw. geschrieben.

Adressierung: **#\$XXXXYYbZZZ** Es wird Text (Z-Zeichen) ab der Zelle gelesen oder geschrieben.
(String) z.B. **#\$000100b6** XXX= dezimale Nummer des Datenbausteins.
YYY = dezimale Nummer des Datenwortes.
ZZZ = Anzahl der Zeichen die gelesen oder geschrieben werden.

2.1.2. Adressierung für Datenbausteine mit Gerätenummer (absolut)

Mit Gerätenummer müssen z.B. Geräte im MPI BUB oder Nockenschaltwerk CamBUS angesprochen werden, wenn mehr als ein Gerät vorhanden ist.

Adressierung: **##XXXYYGZZ** Es werden 8, 16 oder 32 Bit gelesen/geschrieben.
eines Gerätes. z.B. **##201000G01** XXX = dezimale Nummer des Datenbausteins.
YYY = dezimale Nummer des Datenwortes bzw. Offset.
ZZ = Gerätenummer des CamCon oder MPI Adr. der S7.

Adressierung: **##XXXYYbnnGZZ** Es wird das Bit nn im Gerät ZZ gelesen/geschrieben.
eines Bits in z.B. **##201000b00G01**
einem Gerät.

Adressierung: **##XXXYYbnn_vvGZZ** Es werden mehrere Bits (nn bis vv) ohne Vorzeichen im
mehrere Bits z.B. **##201000b01_05G01** Gerät ZZ gelesen/geschrieben.
in einem Gerät.

Hinweis: Wird keine Gerätenummer angegeben, wird mit der in der Konfiguration angegebenen "Device Default Adr." kommuniziert. Bei DigiWEB mit MPI - Option stehen zusätzlich zu den 64 Gerätenummern an der seriellen Schnittstelle auch noch die Adressen 128 bis 254 zur Verfügung. Diese sind die Adressen der MPI - Schnittstelle. Soll eine S7 - CPU mit MPI - Adresse 2 angesprochen werden, so muß die Gerätenummer 130 verwendet werden.

2.1.3. Symbolische Adressierung

Im DigiWEB (ab V2) ist auch eine symbolische Adressierung möglich. Diese müssen in der DigiWEB Konfiguration zunächst definiert und können dann im Script verwendet werden.

Sehen Sie hierzu die DigiWEB "advanced" Konfiguration "<http://DigiWEB-IP/config/>" unter dem Menüpunkt: "Symbols". Hier kann z.B. ein Symbol mit dem Namen ##Eingang1 definiert werden, das dann z.B. auf das S7 Eingangsbit ##E0.0 zeigt.

Für jedes Symbol kann ein separater Lese - und Schreib - Ausdruck bzw. Expression (Adresse) definiert werden. Hierdurch ist es z.B. möglich, ein Datenwort mit einem Faktor zu versehen, wobei beim Schreiben des Symbols der zugewiesene Wert zunächst in der Variable ##INPUT steht und dann der absoluten Adresse mit der Verrechnung oder auch Bedingung zugewiesen wird.

Beispiel 1: Symbolname: "##Wert1"
Lesen: "##DB50.DBW10" Schreiben: "##DB50.DBW10:=##INPUT"

Beispiel 2: Symbolname: "##Wert2" mit Verrechnung.
Lesen: "##DB50.DBW10 / 2" Schreiben: "##DB50.DBW10:=##INPUT*2"

Beispiel 3: Symbolname: "##Wert3" mit logischem Operator.
Lesen: "##DB50.DBW10" Schreiben: "##DB50.DBW10:=(##INPUT>5:5:##INPUT)"

Beispiel 4: Symbolname: "##VBIT[##A]" mit Parameter. Symbole können auch mit Parametern aufgerufen werden um so z.B. einen Indizierung zu ermöglichen.
Lesen: "##PLCBITGET(##A-1,'V')"
Schreiben: "##PLCBITSET(##INPUT?1:0,(##A-1),'V')"

2.1.4. Adressierung für Datenaustausch zwischen DigiWEB und DigiWEB via Netzwerk

Es ist möglich, auf eine Variable eines zweiten DigiWEB (quasi remote) lesend oder schreibend zuzugreifen (r/w). Hierzu muß statt einer Gerätenummer nur die IP - Adresse des remote bzw. slave DigiWEB angegeben werden.

Achtung: Diese Adressierungsart darf nicht bidirektional erfolgen. Das heißt, nur ein DigiWEB (Master) darf den Datenaustausch beginnen. Das zweite DigiWEB (Slave) darf **keinesfalls** versuchen, seinerseits eine Verbindung zum Master aufzunehmen. Dies gilt sowohl für das Lesen (r) als auch für das Schreiben (w) von Variablen.

Das **nicht** beachten dieses Single - Master - Prinzips führt zu einer Verklemmung und zum Neustart beider DigiWEB Geräte.

Hinweis: Auf passwortgeschützte Variablen kann nur dann zugegriffen werden, wenn die IP - Adresse des Master DigiWEBs in der "User" - Datenbank des remote bzw. slave DigiWEBs eingetragen ist oder der Passwortschutz der Variablen entfernt wird.

Tipp: Ab Firmware V3.308 04/2013 sollten Sie die Funktion `##ExchangeInsert()` verwenden. Diese löst die beiden unter Achtung und Hinweis zuvor beschriebenen Probleme. Sehen Sie hierzu Kapitel "3.2.7. Die Exchange Funktionen bzw. Variablenaustausch" auf Seite 23.

2.1.4.1. Absolute Adressierung

Adressierung: `##XXXYYYPZ.Z.Z.Z` z.B. lese Register `##000000` vom DigiWEB
via IP-Adresse. z.B. `##000000IP192.168.2.42` mit der IP 192.168.2.42.

mit variabler `##XXXYYYP[Z].[Z].[Z].[Z]` Lesen mit indizierter IP Adresse, wobei in `##IP`
IP-Adr. z.B. `##000000IP192.168.2.[##IP]` z.B. nur die letzte Stelle der IP steht.

Hinweis: Die IP - Adresse kann auch als 32 Bit Zahl angegeben werden, wenn folgender Syntax verwendet wird: `##IP[0xC0A8022F]XXXYYY` bzw. mit Variable `##IP[##000512]XXXYYY`.

2.1.4.2. Symbolische Adressierung

Auch die symbolische Adressierung kann bei der Kommunikation zwischen DigiWEB <-> DigiWEB via Ethernet verwendet werden, wobei die Symbole des remote DigiWEB angegeben werden müssen.

Folgender **Ausdruck** wird hierzu beispielsweise verwendet: `##IP[0xC0A8022F]DWTimer`

2.1.5. Adressierung im S7/S5 Format

Zusätzlich zu den zuvor beschriebenen Adressierungsarten ist auch die Siemens S7 bzw. S5 Adressierung zulässig. Hierbei müssen jedoch immer zwei Doppelkreuze als Einleitung voran gestellt werden. Für die S5 und S7 wird der gleiche Syntax verwendet.

Achtung: Die "RK512 Cell Size" liegt bei dieser Adressierungsart für eine S7 immer fix auf 1 und für eine S5 immer fix auf 2, unabhängig vom eingestellten "RK515 Cell Size" Wert. Sehen Sie hierzu die DigiWEB "advanced" Konfiguration "<http://DigiWEB-IP/config>" unter dem Menüpunkt: "Serial/MPI/PB".

2.1.5.1. Adressierung für Datenbausteine

Bit:	##DBx.DBXy.z	x = Datenbausteinnummer / y = Datenbyteadresse / z = Datenbitnummer.
Byte:	##DBx.DBBy	x = Datenbausteinnummer / y = Datenbyteadresse.
Wort:	##DBx.DBWy	x = Datenbausteinnummer / y = Datenwortadresse.
Doppelwort:	##DBx.DBDy	x = Datenbausteinnummer / y = (DINT) Datendoppelwortadresse.
Timersollwert:	##DBx.DBTy	x = Datenbausteinnummer / y = Datenwortadresse für den Timer - Sollwert.
Zählersollwert:	##DBx.DBZy	x = Datenbausteinnummer / y = Datenwortadresse für den Zähler - Sollwert.
Flieskommazahl:	##DBx.DByFz (nur S7)	x = Datenbausteinnummer / y = Kommastelle / z = Datendoppelwortadresse.

2.1.5.2. Adressierung für Ein - Ausgänge und Merker

Bit:	##Ex.y	x = Eingangsbyteadresse / y = Bitnummer.
Byte:	##EBx	x = Eingangsbyteadresse.
Wort:	##EWx	x = Eingangswortadresse.
Doppelwort:	##EDx	x = Eingangsdoppelwortadresse.

Hinweis: Der Syntax für Ausgänge und Merker verhält sich analog zum Syntax der Eingänge "E". Es werden hierzu nur die Zeichen "A" für Ausgänge und "M" für Merker verwendet.

2.1.5.3. Adressierung für Zähler

Zähler: ##Zx x = Zählernummer. Der maximale Wert eines Zählers ist 999.

2.1.5.4. Adressierung für Zeiten (Timer)

Zähler: ##Tx x = Timernummer
Die Zeit - bzw. Timerwerte werden in einer Auflösung von 10 ms übertragen und müssen gegebenenfalls mit 10 multipliziert werden..

Hinweis: Zähler und Timer können durch das DigiWEB **nicht** beschrieben werden.

Beispiele:

"##DB50.DBX7.1"	= Datenbaustein 50 - Bit 7.1,	"##E5.1"	= Eingangsbit 5.1,
"##Z1"	= Zähler 1,	"##T20"	= Zeit / Timer 20
"##AB5"	= Ausgangsbyte 5	"##DB50.DB3F20"	= Flieskommazahl im DD20 mit 3 Nachkommastellen.
"##DB20.DBD10"	= Datenbaustein 20 - DInt 10		

2.1.6. Indirekte Adressierung

In allen Adressvarianten kann eine indirekte (indizierte) Adressierung (Index) angegeben werden (auch bei der IP - Adresse). Durch Einfügen von [] Klammern können Datenbaustein, Datenwort, Bitposition und Gerätenummer indirekt (indiziert) adressiert werden. Dies ist zur Erstellung von Schleifen notwendig.

Beispiel 1: `##000[##000256+1]b[##000257], ##[##Index]000` oder `##[##Index1][##Index2]`

als S7 Adresse: `##DB50.DBW[##Index], ##DB[##Datenbaustein].DBW[##Datenwort]` oder
 `##DB50.DB3F[##Index]`

Beispiel 2: `##DB50[##Index1].DBW[##Index2]G[128+##MPI_ADR]IP192.168.2.[42]`

2.2. Operatoren

Das DigiWEB kann eine von seinem Kommunikationspartner (CamCon, S7, S5 oder eine andere Steuerung) gelesene Variable durch "Operatoren" umrechnen bzw. durch "Bedingungen" verändern. Es stehen verschiedene "Operatoren" bzw. "Bedingungen" zur Verfügung, die in Ausdrücke zusammengefasst werden.

2.2.1. Arithmetische Operatoren

Es gelten die normalen "**Punkt vor Strich**" Rechenvorschriften. Alle Rechengänge werden in 32 Bit Werten mit Vorzeichen **ohne** Kommastelle durchgeführt. Müssen mehrere Rechengänge durchgeführt werden, so können diese durch ein Semikolon voneinander getrennt werden. Das Ergebnis des letzten Ausdrucks wird hierbei zur Anzeige gebacht (z.B. `##000256 := 5; ##000257 * ##000258 + ##000259`).

Addition:	<code>##000187 + ##000256</code>
Subtraktion:	<code>##000187 - ##000256</code>
Multiplikation:	<code>##000187 * ##000256</code>
Division:	<code>##000187 / ##000256</code>
Modulo (Teilungsrest):	<code>##000187 % ##000256</code>
Klammerebenen:	<code>##000187 * (5 + ##000256)</code> oder <code>(##000187 + 0x5) * ##000256</code>
Zuweisung:	<code>##000187 := 5</code>
Konstanten:	Konstanten können dezimal (5) oder hexadezimal (0x5) eingegeben werden.

2.2.1.1. Berechnungen > 32 Bit

Zur Berechnung von Werten, dessen Zwischenergebnis größer als 32 Bit wird, verwenden Sie die Funktion `##muldiv(x,y,z)`. Diese rechnet intern mit einem 64 Bit Wert.

Multiplikation / Division: `##muldiv(Wert1,Wert2,Wert3)`
 Wert 1 und Wert 2 werden miteinander multipliziert und dann durch den Wert 3 dividiert.

2.2.2. Boolesche Operatoren

UND bzw. AND	<code>##000187b1 & ##000187b2</code>	
ODER bzw. OR	<code>##000187b1 ##000187b2</code>	
Schiebe rechts bzw. Shift right	<code>##000187 >> 6</code>	(z.B. Schiebe Wert 6 Bit nach rechts)
Schiebe links bzw. Shift left	<code>##000187 << 2</code>	(z.B. Schiebe Wert 2 Bit nach links)
XODER bzw. XOR	<code>##000187 ^ ##000256</code>	
NICHT bzw. Invertierung	<code>##000050b5 && !##000050b1</code>	(z.B. Wenn Bit 5 und NICHT Bit 1)
Stürzen bzw. Bit invertieren	<code>~##000050b0_3</code>	(z.B. Aus 1001 wird 0110)

2.2.3. Logische Operatoren

Frage bzw. Bedingungen	<code>##000187b1 ? 5 : 6</code> Ist das Bit 1 gesetzt, so wird hier als Ergebnis 5, bei nicht gesetztem Bit 1 wird 6 geliefert.
Fragen auf gleich	<code>##000187 == ##000257 ? 5 : ##000258</code> Ist <code>##000187</code> gleich <code>##000257</code> , so wird hier als Ergebnis 5, bei ungleich wird der Wert aus <code>##000258</code> geliefert.
Fragen auf ungleich	<code>##000187 != ##000257</code>
Fragen auf größer	<code>##000187 > ##000257</code>
Fragen auf kleiner	<code>##000187 < ##000257</code>
Fragen auf größer gleich	<code>##000187 >= ##000257</code>
Fragen auf kleiner gleich	<code>##000187 <= ##000257</code>
UND, logisches	<code>##000187 == 5 && ##000257 == 6</code>
ODER, logisches	<code>##000187 == 5 ##000257 == 6</code>

Beispiel für eine logische Operation mit bedingter (Fragezeichen '?') Anweisung:

```
<!--##000256:= (##000187b5 ? 5 : 6)-->
```

Hinweis: Wenn Bit 5 im Register `##000187` gesetzt ist (true), wird eine 5 in das Register `##000256` geschrieben oder eine 6, wenn das Bit nicht gesetzt (false) ist.

Eine bedingte bzw. Fragezeichenanweisung besteht immer aus fünf Elementen:

1. Der Frage
2. Dem 1. Seperator bzw. dem Fragezeichen '?'.
3. Das Ergebniss, wenn die Frage wahr (true) ist.
4. Dem 2. Seperator bzw. dem Doppelpunkt ':'.
5. Das Ergebniss, wenn die Frage unwahr (false) ist.

Hinweis: Zur Übersichtlichkeit sollten die 5 Elemente immer von einer Klammer umschlossen werden, siehe oben. Es sind auch verschachtelte Fragezeichenanweisungen möglich.

Beispiel für eine logische Operation in einer "#if" Anweisung der Ablaufsteuerung:

```
<!--#if(##000187b5)-->  
    <!--##000256:=5-->  
<!--#else-->  
    <!--##000256:=6-->  
<!--#endif-->
```

Beispiel für eine logische Operation kleiner (<) in einer "#while" Schleife der Ablaufsteuerung:

```
<!--##Index:=1-->  
<!--#while( ##Index < 5 )-->  
    <!--##[##Index]001:=50-->  
    <!--##[##Index]002:=60-->  
    <!--##Index:=##Index+1-->  
<!--#endwhile-->
```

2.2.4. String - bzw. Zeichenkettenoperatoren

Frage bzw. Bedingungen `##000187b1 ? 'Text1' : 'oder Text2'`
Ist das Bit 1 gesetzt, so wird der Text "Text1" geliefert, im anderen Fall "oder Text2".

Addition (String anhängen): `##$Str1 + ##$Str2`
Bei der Addition von Strings wird der Inhalt eines Strings (`##$Str2`) an den Inhalt des anderen Strings (`##$Str1`) angehängt. Hierbei ist auch eine Kombination von Strings und Variablen möglich.

Beispiel: `##$Str2:='Offset';##Type:=178;##$Str1:='Text1: '+##Type+' mm '+##$Str2`
Ergibt in `##$Str1: 'Text1: 178 mm Offset'`.

Subtraktion: `##$Str1 - 10`
Es werden die ersten Zeichen (z.B. 10) des Strings entfernt.

Modulo (String verkürzen): `##$Str1 % 5`
Der String wird auf maximal 5 Zeichen begrenzt bzw. der String wird abgeschnitten, wenn er länger als 5 Zeichen ist.

Splitten: `'Str1|Str2|Str3' / 1`
Wenn ein String das Zeichen '|' (0x7c) enthält, kann durch die Division der String zerlegt werden. Als Ergebnis erhält man das durch den Divisor angegebene Element (hier z.B. 'Str2').

Zuweisung: `##$Str0 := 'Text1' + ' und ' + 'Text2'`

Fragen auf gleich `##$Str0 == ##$Str1`

Fragen auf ungleich `##$Str0 != ##$Str1`

Fragen auf größer `##$Str0 > ##$Str1`

Fragen auf kleiner `##$Str0 < ##$Str1`

Fragen auf größer gleich `##$Str0 >= ##$Str1`

Fragen auf kleiner gleich `##$Str0 <= ##$Str1`

Hinweis: Bei allen Vergleichsoperatoren gilt: Das Ergebnis wird wahr, wenn die Frage mit "ja" beantwortet werden kann. Hierzu muss man wissen, dass die Zeichen der Strings nacheinander verglichen werden. Das erste Zeichen in einem String, das unterschiedlich ist, entscheidet, wie der Vergleich ausfällt.

Beispiel 1: 'Text1' und 'Text2' - 'Text1' ist kleiner (<) als 'Text2' da '1' kleiner als '2' ist.

Beispiel 2: '1-Text' und '2-Text' - auch hier ist der '1-Text' kleiner (<) als '2-Text'.

Beispiel 3: 'Text1' und 'Text' - hier ist der 'Text1' grösser (>) als 'Text'.

Achtung: Strings bzw. Zeichenketten müssen immer zwischen zwei " " Zeichen (Ticks) gesetzt sein.

Beispiel 1:

```
<!--##$Str0:='S'--><!--##Typ:=7--><!--##ED0-->
<!--##$Status:=(##errno())?'keine ':' '+##$Str0+##Typ+' gefunden.'-->
```

Beispiel 2 mit dem Zeitstring:

```
<!--#if( ( ##$000172 - 16 ) % 3 == ':00' ) -->
    Sekunden ist Gleich :00 bzw. volle Minute ist erreicht
<!--#else-->
    Sekunden sind ungleich :00
<!--#endif-->
```

2.2.4.1. String - bzw. Zeichenkettenlänge ermitteln

Um die Länge einer Zeichenkette (String) zu ermitteln, verwenden Sie die Funktion `##strlen(##$var)`. Dieser übergeben Sie die Stringvariable und erhalten die Länge des Strings zurück.

Diese Funktion benötigen Sie zum Beispiel beim Schreiben eines Strings in eine S7, da hier die Anzahl der Zeichen eines Strings in ein Register eingetragen werden muß.

Beispiel zum Schreiben einer Zeichenkette bei einer S7 - SPS:

```
##DB1.DB16b[##DB1.DBB14]:=##$Text; ##DB1.DBB15:=##strlen(##$Text)
```

Erklärung: Die Definition der Zeichenkette beginnt in der S7 bei DB1.DB14. Die ersten Zeichen beginnen jedoch bei DBB16. `##DB1.DBB14` ist hier die Anzahl der Bytes die maximal gespeichert werden können. `##DB1.DBB15` enthält die effektive Länge der Zeichenkette.

2.3. Lokale Variablen

Lokale Variablen werden durch eine einfache Zuweisung (`:=`) automatisch angelegt. Der Variablenname darf nur aus alphanumerischen Zeichen und dem Unterstrichzeichen bestehen, wobei erst immer `##` (Zahl) oder `##$` (String) gefolgt von einem **nicht** numerischen Zeichen sein muß. Keinenfalls darf der gewählte Namen dem eines Symbolnamens entsprechen. Ihre Anwendung ist vielfältig wie z.B. die indizierte Adressierung in Schleifen.

Beispiel:

```
<!--##Index:=256--> Zuweisung der Index - Variablen.  
<!--##000[##Index]:=1--> Indizierte Adressierung im Bereich 000  
<!--##Index:=##Index+1--> Erhöhung der Index - Variablen um 1.  
<!--##000[##Index]:=2--> Neue Adresse.
```

Beispiel in einer Schleife durch die Ablaufsteuerung:

```
<!--##Index:=1-->  
<!--#while( ##Index < 6 )-->  
  <!--##[##Index]001:=50-->  
  <!--##[##Index]002:=60-->  
  <!--##Index:=##Index+2-->  
<!--#endwhile-->
```

2.3.1. Lokale Variablen als Parameter übergeben

Eine lokale Variable wird auch angelegt bzw. definiert, wenn beim Aufrufen der REQ - Datei ein Parameter aus der DigiVISU übergeben wird (z.B. `var.req?##Wert:=2`).

Hinweis: Eine Parameterübergabe in der Event - Steuerung ist zur Zeit nicht möglich.

Wird die REQ - Datei von mehreren Stellen aus mit oder auch ohne Parameter aufgerufen, so muß vor dem Verwenden der Parametervariablen im Script diese zunächst auf Vorhandensein geprüft oder neu definiert werden.

Hierzu dient die Variablenfunktion `##defined()` die "true" oder "false" zurückliefert.

Beispiel:

```
<!--#if(##defined(##Wert))-->  
  Das Script wurde mit einem Parameter ##Wert aufgerufen.  
  <!--##000256:=##Wert-->  
<!--#else-->  
  Das Script wurde ohne Parameter aufgerufen.  
  <!--##000256:=0-->  
<!--#endif-->
```

2.3.2. Variablenfunktion "##defined()"

Das DigiWEB stellt eine Funktion zur Verfügung, die feststellt, ob eine Variable bereits definiert wurde. Diese besteht aus dem Ausdruck `##defined(VarName)` mit dem Variablennamen in der Klammer. Ist die Variable schon definiert, so ist das Ergebnis "Wahr" und kann in im Script abgefragt werden. Sehen Sie das Beispiel im Kapitel zuvor.

2.4. Ablaufsteuerung

Durch die Ablaufsteuerung ist es möglich, Abschnitte eines Scripts bedingt (#if) oder mehrfach (#until, #while) aufzurufen. Es stehen 3 Typen der Steuerung zur Verfügung. Diese sind: Die "#if" - Bedingung mit einem "#else" -Zweig (entweder oder), die "#while" - Schleife (solange wie) und die "#until" - Schleife (solange bis).

Die Ablaufsteuerung kann maximal bis zu 9 Ebenen tief sein.

2.4.1. Die "#if" Bedingung

Die "#if" Anweisung besteht aus min. zwei oder mehreren HTML - Kommentaren. Der erste Kommentar muß immer die Zeichenkette "#if(Bedingung)" und die logische Bedingung in einer Klammer enthalten. Der zweite Kommentar enthält entweder die Zeichenkette "#endif" als Kennung für das Ende der Bedingung oder die Zeichenkette "#else" oder "#elif(Bedingung)". Die "#else" bzw. "#elif(Bedingung)" Bedingung wird ausgeführt, wenn die Bedingung in der "#if" Abfrage nicht wahr ist. Der letzte Kommentar einer "#if" Bedingung muß immer die "#endif" Anweisung sein. Das DigiWEB verarbeitet nun nur den Script - Code, der zwischen den Kommentaren in Abhängigkeit der Bedingung steht.

Beispiel:

```
<!--#if(##000187b5)-->
    Dieser Code wird bearbeitet wenn Bit 5 true ist.
    <!--##Wert1:=1-->
<!--#endif-->
```

Beispiel mit "#else" - Zweig:

```
<!--#if(##000187b5)-->
    Dieser Code wird bearbeitet wenn Bit 5 true ist.
    <!--##Wert1:=1-->
<!--#else-->
    Dieser Code wird bearbeitet wenn Bit 5 false ist.
    <!--##Wert1:=0-->
<!--#endif-->
```

Beispiel mit "#elif" und "#else" - Zweig:

```
<!--#if(##$Text=='T1')-->
    Text war 'T1'
<!--#elif(##$Text=='T2')-->
    Text war 'T2'
<!--#elif(##$Text=='T3')-->
    Text war 'T3'
<!--#else-->
    Text war weder 'T1' noch 'T2' oder 'T3'
<!--#endif-->
```

2.4.2. Die "#while" Schleife

Die "#while" Schleife besteht aus zwei HTML - Kommentaren. Der erste Kommentar muß immer die Zeichenkette "#while(Bedingung)" und die logische Bedingung in einer Klammer enthalten. Der zweite Kommentar ist die Endekennung der Schleife "#endwhile". Das DigiWEB wiederholt den Script - Code, der zwischen den Kommentaren steht, solange bis die Bedingung unwahr wird. Ist die Bedingung zu Beginn schon unwahr, erfolgt im Gegensatz zur "#until" - Schleife keine Einmal - Verarbeitung.

Beispiel:

```
<!--##Index:=1-->
<!--#while( ##Index < 6 )-->
    <!--##[##Index]001:=50-->
    <!--##[##Index]002:=60-->
    <!--##Index:=##Index+2-->
<!--#endwhile-->
```

Achtung: Wird die logische Bedingung der while - Schleife niemals "unwahr" bzw. "false", haben Sie eine Endlosschleife, die nach ca. 20 - 30 Sekunden zu einem Neustart des DigiWEBs führt (Watchdog).

2.4.3. Die "#until" Schleife

Die "#until" Schleife besteht aus zwei HTML - Kommentaren. Der erste Kommentar muß immer die Zeichenkette "#repeat" enthalten. Der zweite Kommentar enthält die Endekennung der Schleife "#until(Bedingung)" und die logische Bedingung in einer Klammer. Das DigiWEB wiederholt den Script - Code, der zwischen den Kommentaren steht solange, bis die Bedingung wahr wird. Im Gegensatz zur "#while" - Schleife wird bei der "#until" - Schleife die Bedingung erst am Ende überprüft, sodass der Script - Code mindestens einmal durchlaufen wird.

Beispiel:

```
<!--##Index:=5-->
<!--#repeat-->
    <!--##[##Index]001:=50-->
    <!--##[##Index]002:=60-->
    <!--##Index:=##Index-1-->
<!--#until(##Index<1)-->
```

Achtung: Wird die logische Bedingung der until - Schleife niemals "wahr" bzw. "true", haben Sie eine Endlosschleife, die nach ca. 20 - 30 Sekunden zu einem Neustart des DigiWEBs führt (Watchdog).

2.4.4. Die "#include" Funktion

Die "#include" Funktion kann dazu verwendet werden, um Scripte nachzuladen. Hierdurch ist eine **bedingte Art** Unterprogramm Funktion (Makro) möglich. Zusätzlich wird die Verarbeitung der aufrufenden Script - Datei schneller, da die z.B. bei einer "#if" Verzweigung nicht mehr den ganzen Scriptcode zu parsen braucht.

Beispiel 1:

```
<!--#if(##V>5)-->
    <!--#include virtual="script1.req"-->
<!--#else-->
    <!--#include virtual="script2.req"-->
<!--#endif-->
```

Beispiel 2:

Unterprogramm nachladen mit lokaler Variable ##Var als Parameter

```
<!--##Var:=512-->
<!--#include virtual="script.req"-->
<!--##Var:=612-->
<!--#include virtual="script.req"-->
```

Parameter verwenden in der Datei: script.req

```
<!--#if(##defined(##Var))--> zur Sicherheit Fragen ob ##Var da ist.
    <!--##000[##Var]:=1-->
    <!--##000[##Var+1]:=2-->
    <!--##000[##Var+2]:=3-->
    <!--##000[##Var+2]:=3-->
<!--#endif-->
```

3. Die DigiWEB Funktionen

Hier folgt eine Übersicht aller zur Zeit im DigiWEB verfügbarer Funktionen der Script - Sprache in alphabetischer Reihenfolge.

Funktion	Beschreibung
##ALLOWEDAEREAS() Rückgabe: : Long bzw. 32Bit Wert	Liefert die erlaubten Zugriffsbereiche als Bit - Maske während der aktuellen Anfrage.
##COMMENTOUT('Dateiname') Rückgabe: : keine Beispiel: <!--##COMMENTOUT('Dateiname')--> <!--##\$String:='Diesen Text schreiben'-->#*^ <!--##\$String:='Diesen Text nicht'-->#*	Ausgaben des DigiWEB, bei denen die Platzhalterzeichen '#' mit dem Zeichen '^' schließen in eine Datei schreiben. Hinweis: Diese Funktion kann nur in einer HTML - oder REQ - Script - Datei verwendet werden.
##CPYFILE('Quelldatei', 'Zieldatei') Rückgabe: == 0 : OK ! = 0 : Fehler	Kopiert eine Datei. Es sind Supervisor bzw. FTP - Rechte notwendig.
#\$DATETOSTR(Long/32Bit,'Format') Rückgabe: : Datums und Zeitstring Beispiel 1: #\$DATETOSTR(-827521292,'Y:M:D:h:m:s') Rückgabe: : 2009:11:17:11:26:44 Beispiel 2: #\$DATETOSTR(-827521208,'YY:M:D h-m-ss') Rückgabe: : 09:11:17 11-28-08	Wandelt eine Zahl (32Bit) in eine Zeichenkette bzw. einen String um. Die Umwandlung der 32Bit Zahl beginnt ab dem 1.1.1900. Ab Firmware Version 3.030 kann ein zusätzlicher Formatstring angegeben werden. Hierbei werden folgende Platzhalterzeichen verwendet bzw. ersetzt: Y = Jahr / M = Monat / D = Tag / N = Monat als englischer Text h = Stunde / m = Minute / s = Sekunde. Die Zeichen können auch mehrfach angegeben werden. Hierdurch wird die Ausgabe des Elements auf die Anzahl der Zeichen begrenzt bzw. mit führenden Nullen aufgefüllt.
##DEFINED(Variablenname) Rückgabe: == 0 : Variable nicht definiert ! = 0 : Variable definiert	Prüft, ob eine lokale Variable definiert bzw. angelegt ist oder z.B. als Parameter übergeben wurde. Dies ist notwendig, da der Zugriff auf eine nicht definierte Variable eine Fehlermeldung verursacht.
##DELFILE('Dateiname') Rückgabe: == 0 : OK ! = 0 : Fehler	Löscht eine Datei auf dem DigiWEB - Flash - Speicher. Hierfür sind Supervisor bzw. FTP - Rechte notwendig.
##DNS('URL-Namen/Host') Rückgabe: == 0 : Fehler ! = 0 : Host - IP - Adresse / 32Bit	Liefert die IP - Adresse (Long/32Bit) eines Host - Computers mit der angegebenen URL - bzw. Hostname. Hierzu muß ein DNS - Server in der Konfiguration definiert sein.
##ERRNO() Rückgabe: : Fehlercode	Liefert den aktuellen Fehlercode der HTML - Seite oder des letzten Befehls.
##EXISTS('Dateinamen') Rückgabe: < 0 : Datei nicht vorhanden >= 0 : Dateilänge	Prüft ob eine Datei im DigiWEB Flash - Speicher vorhanden ist bzw. wie groß diese ist.
##FLUSH() Rückgabe: : keine	Schreibt sofort alle Variablen aus dem Cache in die Gegenstelle (z.B. S7 oder CamCon). Diese Funktion kann nur in einer HTML - oder REQ - Script - Datei verwendet werden.
#\$FORMAT('FormatString',Wert) Rückgabe: : Formatierter Wert bzw.String Beispiel: #\$FORMAT('##.##mm',51) => 5.1mm #\$FORMAT('#B####',5) => 101 #\$FORMAT('#X##',77) => 4D	Formatiert eine Zahl/Wert durch einem Formatstring, z.B. zur Darstellung mit Kommastelle. Die Platzhalterzeichen '#' des Strings werden mit dem Wert befüllt. Das Ausgabeformat läßt sich in HEX und Binär umschalten, wenn die ersten beiden Zeichen des Formatstrings ein '#X' = HEX oder '#B' = BIN enthalten.
##GETDIRHANDLE('File-such-String') Rückgabe: == 0 : Keine Datei gefunden > 0 : Handlenummer Beispiel: ##H:=##GETDIRHANDLE('*.*csv') #\$GETDIRLINE(##H)	Diese Funktion erwartet einen Suchstring und dient zum Durchsuchen des DigiWEB - Flash - Speichers nach Dateien. Sie liefert einen Handle bzw. eine Zahl zurück, die dann der Funktion #\$GETDIRLINE(Handle) als Parameter übergeben werden kann und die Informationen der ersten gefundenen Datei zurückgibt.
#\$GETDIRLINE(Handle) Rückgabe: : String mit Dateinfos Beispiel: #\$GETDIRLINE(##H) => '-rwxrw-rw- 0 ftp ftp 110 Jan 25 2009 Strom.csv'	Liefert eine durch das Handle definierte gefundene Datei bzw. Dateiinformation zurück. Bei wiederholte Aufruf der Funktion wird die nächste gefundene Datei zurückgeliefert. Ein Leerstring zeigt an, das keine Datei mehr gefunden wurde.
#\$GETFILENAME() Rückgabe: : Dateiname Beispiel: : 'Index.htm'	Liefert den Namen der Datei, die gerade verarbeitet bzw. geladen wird. Diese Funktion kann nur in einer HTML - oder REQ - Datei verwendet werden und nicht in der DigiVISU.
##GETLANGUAGE(Magicnummer) Rückgabe: : 0..255	Liefert die in der Passwortdatenbank eingestellte Sprache des Users. Als Parameter muß die Magic - bzw. Sitzungsnummer übergeben werden.
##GETMG() Rückgabe: == 0 : nicht eingeloggt > 0 : 32 Bit Zahl = Magicnummer	Diese Funktion liefert die aktuelle Magic - bzw. Sitzungsnummer der Anfrage des eingeloggtten Users.

Funktion	Beschreibung
##LOGCOUNT() Rückgabe: : 0..16	Diese Funktion liefert ab DW2 die Anzahl der eingeloggten User. Maximal können sich 16 User einloggen, wobei jedoch die Anzahl der gleichzeitigen HTTP/1.1 Verbindungen begrenzt ist. Siehe Register ##000153b0_7 im Kapitel "4.2. Die Netzwerkkonfiguration, Systemvariablen und Systempasswort".
##LOGIN('Username','Passwort') Rückgabe: == 0 : Login nicht erfolgreich. > 0 : Magic - / Sitzungsnummer.	Diese Funktion dient zur Anmeldung eines Benutzers. Sie liefert eine neue Magic - bzw. Sitzungsnummer für den mit Username und Passwort angegebenen Benutzer.
##MINMAX(ZuprüfenderWert,Min,Max) Rückgabe: : geprüfter/beschnittener Wert	Funktion zur Min - / Max - bzw. zur Eingabebegrenzung. Der zurückgegebene Wert ist überprüft und ggf. beschnitten werden.
##MKDIR('Verzeichnis') Rückgabe: == 0 : OK != 0 : Fehler	Erzeugt ein Unterverzeichnis im DigiWEB - Flash - Speicher. Hierfür sind Supervisor bzw. FTP - Rechte notwendig.
##MULDIV(Wert,Multiplikator,Divisor) Rückgabe: : Long bzw. 32Bit Wert	Diese Funktion berechnet einen Ausdruck mit 64 Bit Zwischenergebnis. Multipliziert erst den 'Wert' mit dem 32Bit 'Multiplikator', anschließend wird durch den 32Bit 'Divisor' geteilt.
##RENAME('Oldname','Newname') Rückgabe: == 0 : OK != 0 : Fehler	Benennt eine Datei auf dem DigiWEB - Flash - Speicher um. Hierfür sind Supervisor bzw. FTP - Rechte notwendig.
##RMDIR('Verzeichnisname') Rückgabe: == 0 : OK != 0 : Fehler	Löscht ein leeres Unterverzeichnis im DigiWEB - Flash - Speicher. Hierfür sind Supervisor bzw. FTP - Rechte notwendig.
##STRLEN('String') Rückgabe: : 0..400	Liefert die Anzahl der Zeichen eines Strings. Zur Zeit kann ein String im DigiWEB maximal 400 Zeichen (inklusive Nullbyte) lang werden.
##STRTODATE('String') Rückgabe: : 0..400	Wandelt eine Zeichenkette in einen 32 Bit Datum und Zeitwert um. Der String muß folgenden Aufbau haben: '2005-07-20 15:40:02'. Das Datum der 32Bit Zahl beginnt ab dem 1.1.1900.
##\$URLENCODE('URL-String') Rückgabe: : HEX codierter URL - String.	Diese Funktion wandelt die Sonderzeichen einer URL in HEX - Codierung um (z.B. '#' = '%23+%23' oder 'H=?' = 'H%3D%3F').
##USERLEVELS() Rückgabe: : Long bzw. 32Bit Maske	Liefert den Userlevel des eingeloggten Users bzw. für die aktuelle Magic - / Sitzungsnummer als Bit - Maske zurück. Beispiel: Ist bei einem User der Bereich 1,2 + 3 freigegeben, so liefert die Funktion eine 14 als Ergebnis. Bit 0 zeigt den Superuser Zugang an

3.1. Die Funktionen der DigiWEB SQL Datenbank

Das DigiWEB verwendet zur Datenverwaltung wie z.B. E - Mail, Passwortschutz, Sprachumschaltung und Datenlogging eine integrierte SQL - Datenbank.

DigiWEB emuliert einen SQL - Server mit eingeschränktem Befehlssatz. Die Datenbank - Tabellen werden auf der Flash - Disk als Dateien im CSV - Format (Excel) gespeichert. Zur Darstellung und zum Bearbeiten einzelner Datensätze stehen folgende Funktionen zur Verfügung:

Datenbank (SQL) Funktion	Beschreibung
<pre>##SqlConnection('DB-Name','User','Passwort') Rückgabe: == 0 : OK != 0 : Fehler</pre>	Stellt die Verbindung zum DigiWEB SQL - Server her. Zur Zeit werden die Parameter DB-Name, User und Passwort nicht benötigt, sodass dies der notwendige Aufruf ist: <code>##SqlConnection(",",")</code> .
<pre>##SqlDisconnect() Rückgabe: == 0 : OK != 0 : Fehler</pre>	Beendet bzw. trennt die aktuelle Verbindung zum DigiWEB SQL - Server.
<pre>##SqlQuery('select * from table.csv') Rückgabe: == 0 : OK != 0 : Fehler ##SqlQuery('select count() from tables.csv') Rückgabe: Liefert nach dem ##SqlFetch() im ##SqlField(0) die Anzahl der Datensätze in der Datenbank.</pre>	Wählt alle Datensätze aus der Tabelle, die dann mit <code>##SqlFetch()</code> gelesen werden können. Durch hinzufügen einer Bedingung "where" werden nur bestimmte Datensätze ausgewählt. Beispiele mit "where" - Bedingung: <code>##SqlQuery('select * from table.csv where Dataset = 3')</code> <code>##SqlQuery('select * from table.csv where Dataset > 20')</code> <code>##SqlQuery('select * from table.csv where Text = \BSPTXT\')</code> .
<pre>##SqlFetch() Rückgabe: == 0 : OK bzw. Datensatz gefunden. != 0 : Fehler bzw. kein weiterer Datensatz verfügbar</pre>	Befehl zum Lesen des ersten bzw. nächsten Datensatzes aus den mit <code>##SqlQuery(..)</code> ausgewählten Datensätzen. Jeder weitere <code>##SqlFetch()</code> Aufruf erhöht den Datensatzzeiger, bis keine Datensätze mehr zur Verfügung bzw. vorhanden sind.
<pre>##SqlField('Feld-Name') Rückgabe: : Text bzw. String</pre>	Datenbankfunktion zum Lesen oder Schreiben von Text bzw. String - Variablen eines Datenbank Feldes. Beispiel zum Schreiben: <code>##SqlField('Feld-Name') := 'Text-String'</code>
<pre>##SqlField('Feld-Name') Rückgabe: : Wert. == 0 : kein Zahlenwert gefunden. Bsp.read: ##SqlConnection(",",") ##SqlQuery('select * from table.csv') ##SqlField('Feld-Name') := 123 ##SqlUpdate('table.csv','ID=1'):=1 ##SqlField('Feld-Name') #endif</pre>	Datenbankfunktion zum Lesen oder Schreiben von Werten bzw. von Variablen. Nachfolgend ein Beispiel zum Schreiben: <code>##SqlConnection(",",")</code> <code>##SqlField('Feld-Name') := 123</code> <code>##SqlUpdate('table.csv','ID=1'):=1</code>
<pre>##SqlUpdate('table.csv','Bedingung'):=1 Rückgabe: == 0 : OK != 0 : Fehler</pre>	Update - Befehl bzw. Datensatz ändern. Der Parameter 'Bedingung' legt hierbei fest, welche Datensätze geändert werden sollen. Beispiel zum Schreiben des aktuellen Datensatzes: <code>##SqlConnection(",",")</code> <code>##SqlField('Feld-Name') := 123</code> <code>##SqlUpdate('table.csv','ID'+##SqlField('ID')):=1</code>
<pre>##SqlDelete('table.csv','Bedingung'):=1 Rückgabe: == 0 : OK != 0 : Fehler</pre>	Delete - Befehl bzw. Datensatz löschen. Der Parameter 'Bedingung' legt fest, welche Datensätze gelöscht werden sollen. <code>##SqlDelete('table.csv','Dataset=' + ##SqlField('Dataset')):=1</code>
<pre>##SqlInsert('table.csv'):=1 Rückgabe: == 0 : OK != 0 : Fehler Beispiel: ##SqlConnection(",",") ##SqlField('Text'):=Eintrag' ##SqlInsert('table.csv'):=1</pre>	Befehl zum Erzeugen bzw. Anhängen eines Datensatzes in einer Tabelle ohne automatisches Erzeugen der Datensatznummer bzw. der ID des Datensatzes. Die Daten, Werte oder Strings müssen zuvor in die entsprechenden Datenbank - Felder eingetragen werden.
<pre>##SqlInsertInto('table.csv','SqlQuery'):=1 Rückgabe: == 0 : OK != 0 : Fehler Beispiel: ##SqlInsertInto('t2.csv','select * from t1.csv'):=1</pre>	Ab Firmware Version 3.062 kopiert diese Funktion Datensätze von einer Datenbank in eine Andere. Die zu kopierenden Datensätze werden durch die 'SqlQuery' ausgewählt.
<pre>##SqlInsertMax('table.csv','Feld-Name'):=1 Rückgabe: == 0 : OK != 0 : Fehler Beispiel: ##SqlConnection(",",") ##SqlField('Wert1'):=123 ##SqlField('Text'):=Text-Eintrag' ##SqlInsertMax('logger.csv','ID'):=1</pre>	Befehl zum Erzeugen eines Datensatzes mit einer automatisch Erzeugten bzw. erhöhten Datensatznummer bzw. ID in einem numerischen Feld. Das DigiWEB nimmt bei diesem Befehl die größte Datensatznummer aus dem entsprechendem Feld, erhöht diese um 1 und speichert den neuen Datensatz ab. Auch hier müssen die Daten zuvor in die entsprechenden Felder eingetragen werden.

Datenbank (SQL) Funktion	Beschreibung
<pre>##SqlInsertLog('table.csv',maxlines):=1 Rückgabe: == 0 : OK != 0 : Fehler Beispiel: ##SqlConnect(",","") ##SqlField('Date'):=##000171 ##SqlField('Log'):= 'Event 1' ##SqlInsertLog('writelog.csv',1000):=1</pre>	<p>Ab Firmware Version 3.094 ist es durch diese Funktion möglich, eine Log - bzw. Protokolldatei zu erzeugen, bei der die Datensätze nicht angehängt werden sondern am Anfang der Datei eingefügt werden.</p> <p>Durch den zweiten Parameter wird die Anzahl der Datensätze angegeben. Enthält die Datenbank mehr als mit "maxlines" angegebene Datensätze, werden die älteren Datensätze gelöscht.</p> <p>! Achtung: Wird sehr häufig geschrieben, verkürzt dies die Lebensdauer des Flash - Speichers.</p>

3.2. Zusätzliche Funktionen ab der DigiWEB 2 Hardware und höher

3.2.1. Allgemeine Funktionen

Funktion	Beschreibung
<pre>##CASTFLOAT(Wert,Mul,Div) Rückgabe: Integer Wert</pre>	<p>Diese Funktion wandelt einen durch die ##Modbus() Funktion gelesenen Float - Wert in einen Integerwert um.</p> <ol style="list-style-type: none"> 1. Parameter ist ein IEEE Formatierter 32 Bit Float-Wert, 2. Parameter ist der Multiplikator, 3. Parameter ist der Divisor.
<pre>##\$CHARTOHEX(Byte) Rückgabe: HEX - codierter - String</pre>	<p>Diese Funktion liefert einen 2 Zeichen langen String in HEX aus einer 8 Bit Zahl (z.B. ##\$CHARTOHEX(63) ergibt:'3F').</p>
<pre>##CLIENTIP() Rückgabe: Client IP als Long/32Bit</pre>	<p>Diese Funktion liefert die Internet - IP - Adresse des Browser Computers.</p>
<pre>##CRC16('Str',Len,Upper) Rückgabe: CRC16 Prüfsumme</pre>	<p>Ab Firmware V3.204 liefert diese Funktion die CRC16 Prüfsumme eines Strings (Str) mit der Länge (Len). Wird der Parameter Upper auf 1 gesetzt, werden klein geschriebene Buchstaben zuvor durch 'toUpper' ersetzt.</p>
<pre>##HEXTOLONG('HEXCodeString') Rückgabe: 32 Bit Wert</pre>	<p>Liefert einen 32 Bit Wert aus einem 8 Zeichen langen HEX codierten - String (z.B. ##HEXTOLONG('00000100') ergibt:256).</p>
<pre>##\$HTTPHeader('String') Rückgabe: Header-String == -34 : nicht gefunden Frage: ##\$HTTPHeader('User-Agent') Antwort: ": Mozilla/5.0 (Windows NT....."</pre>	<p>Diese Funktion sucht den als String übergebenen Parameter im HTTP - Header, den das DigiWEB vom Browser empfangen hat.</p> <p>Hiermit kann das DigiWEB erkennen welcher Browser eingesetzt wird.</p>
<pre>##\$LONGTOHEX(Long) Rückgabe: String</pre>	<p>Liefert einen 8 Zeichen langen HEX - codierten - String aus einer 32 Bit Zahl (z.B. ##\$LONGTOHEX(255) ergibt:'000000FF')</p>
<pre>##LOGOUT() Rückgabe: == 0 : OK != 0 : Fehler</pre>	<p>Mit dieser Funktion wird die aktuelle Magic - / Sitzungsnummer freigegeben und der Benutzer abgemeldet.</p>
<pre>##LOGCOUNT() Rückgabe: : 0..15</pre>	<p>Diese Funktion liefert die Anzahl der zur Zeit angemeldeten Benutzer bzw. aktiver Magic - / Sitzungsnummern.</p>
<pre>##SENDMAIL('Sub','Text/File','Adr','File-Att.') Rückgabe: : immer 0</pre> <p>Hinweis: Ob die E-Mail erfolgreich versandt wurde kann nur in der SMTPLOG.CSV Datenbank überprüft werden.</p>	<p>Diese Funktion steht ab der Version 3.074 zur Verfügung und sendet eine E-Mail direkt ohne ein Event der SMTP - Datenbank. Sie können 4 Parameter übergeben:</p> <ol style="list-style-type: none"> 1. Text der Subjekt - Zeile 2. Text der E-Mail oder ein Dateiname, der als Text gesendet wird. 3. E - Mail Adresse 4. (optional) Dateiname, der als Anhang gesendet wird.
<pre>##\$STRUPPER('STRING') Rückgabe: 'STRING'</pre>	<p>Diese Funktion wandelt einen String in Großbuchstaben um und gibt diesen zurück.</p>
<pre>##\$STRLOWER('String') Rückgabe: 'string'</pre>	<p>Diese Funktion wandelt einen String in Kleinbuchstaben um und gibt diesen zurück.</p>
<pre>##\$USERNAME() Rückgabe: : Name des Benutzers</pre>	<p>Diese Funktion liefert zur Aktuellen Magic - / Sitzungsnummer den Benutzernamen.</p>

3.2.2. Netzwerk Sende/Empfangs - Funktionen für TCPIP

Funktion	Beschreibung
##TCPCLOSE() Rückgabe: == 0 : OK != 0 : Fehler	Schließt eine mit ##TCPOPEN geöffnete TCPIP - Netzwerk - Verbindung
##TCPISCLOSE() Rückgabe: == 0 : Verbindung offen != 0 : Verbindung beendet	Diese Funktion liefert den Status einer mit ##TCPOPEN() geöffnete TCPIP - Netzwerk - Verbindung.
##TCPOPEN(IP , Port , Timeoutzeit) Rückgabe: == 0 : OK != 0 : Fehler	Öffnet eine TCPIP - Netzwerk - Verbindung zur Angegebenen IP - Adresse (32Bit) und Port - Nummer (16Bit). Die Verbindung wird beendet, wenn ##TCPCLOSE() aufgerufen wird oder die Timeoutzeit (in ms) abgelaufen ist.
#\$TCPRECEIVE() Rückgabe: String	Liefert eine ASCII String des Hosts bzw. der aktuellen Verbindung, der mit "\r\n" am Ende gesendet wurde. Der String enthält nicht die Zeichen "\r\n"
#\$TCPRECEIVEB(Timeoutzeit) Rückgabe: HEX - codierter - String	Liefert einen in HEX - codierten - String der vom Host während der Warte - bzw. Timeoutzeit (in ms) gesendeten Daten. Die Funktion wird vorzeitig beendet wenn die Länge des Strings die maximale DigiWEB Stringlänge übersteigt.
##TCPSEND('Text String') Rückgabe: == 0 : OK != 0 : Fehler	Sendet den String im ASCII Code an den Host und hängt automatisch die Zeichen "\r\n" an.
##TCPSENDB('HEXCodeString') Rückgabe: == 0 : OK != 0 : Fehler	Wandelt den HEX - codierten - String in Binärcode um und sendet diesen zum Host (z.B. ##TCPSENDB('001BFF00')).

3.2.3. Netzwerk Sende/Empfangs - Funktionen für UDP

Funktion	Beschreibung
##UDPSENDB(Ziel-IP,'PortDataString') Rückgabe: == 1 : OK != ? : Fehler Beispiel: ##UDPSENDB(0xC0A8022F, '7A7A' + '0051' + '010203040506')	Durch diese Funktion kann ein UDP Datenpaket an eine bestimmte IP - Adresse gesendet werden. Das Datenpaket muß als 2 Byte HEX - codierter - String übergeben werden. Dieser enthält: Byte 0 + 1 : Quell - Port (z.B. '7A7A' = 31354) Byte 2 + 3 : Ziel - Port (z.B. '0051' = 81) Byte 4 - 180 : bis zu 176 Datenbytes (z.B.'010203040506') Hinweis: Wird als Ziel - IP Adresse = 0xFFFFFFFF angegeben, so wird das Paket als Broadcast gesendet.

3.2.4. Funktionen zur Steuerung der serielle Schnittstelle

Ab der DigiWEB Firmware V3.059 kann die serielle Schnittstelle des DigiWEB durch die folgenden vier Funktionen angesteuert werden, wenn das RS Protokoll auf **"by DigiWEB Function"** eingestellt ist. Wird eine dieser Funktionen aufgerufen ohne das dies Protokoll eingestellt ist, so erhalten Sie einen Kommunikationsfehler "???".

Hinweis: Als Schnittstellenparameter werden die Einstellungen der **"Virtual Com"** (Register ##000130) verwendet. Sehen Sie Kapitel "4.8. Konfiguration der Seriellen - und der MPI bzw. Profibus - Schnittstelle".

Funktion	Beschreibung
##COMSEND (Mode , 'String') Rückgabe: == 0 oder 1 : OK == ??? : Protokollfehler Beispiel: ##COMSEND (0,'Wert\r\n') ##COMSEND (1,'300D0A')	Funktion zum Senden von Daten bzw. Strings. Im Mode = 0 wird ein ASCII - String mit Nullbyte Terminierung gesendet. Im Mode = 1 wird ein in 2 HEX - Zeichen codierter String gesendet. Dies dient zum Senden von Binärzeichen.
#\$COMRECEIVE (Mode , AnzBytes) Rückgabe: Text bzw. String Beispiel: #\$COMRECEIVE (0,5) #\$COMRECEIVE (1,5)	Diese Funktion liest eine bestimmte Anzahl Bytes aus dem Empfangspuffer in einem bestimmten Mode: Mode 0 = Bytes als ASCII String z.B. 'Wert' lesen. Mode 1 = Bytes als HEX - codierter - String z.B. '300D0A' lesen.
##COMRLEN ('HEXCodeCompString') Rückgabe: 0..255 Anzahl bzw. Position der Bytes im Puffer Beispiel: ##COMRLEN ('0D0A')	Diese Funktion liefert die Anzahl der Bytes im Empfangspuffer bzw. bei Übergabe eines in HEX codierten Vergleichsstring als Parameter die Anzahl der Bytes bis zum Vergleichsstring inklusive dessen Länge.
##COMSFREE () Rückgabe: 255..0 Anzahl freier Bytes im Puffer	Diese Funktion liefert die Anzahl der freien Bytes im Sendepuffer. Hiermit kann geprüft werden, ob noch Bytes zum Senden anstehen.

3.2.5. IoBUS bzw. DBUS - Funktionen des DigiWEB 3 und 4

Funktion	Beschreibung
#\$DBUS ('HEXSTRING') Rückgabe: : String Beispiel 1: #\$DBUS ('0024') Liest 2 Byte von Geräte 0 ab Datenadr. 8 Rückgabe: '02000102' = 25.8° bei "DEZ TEMP/R/BUS" Beispiel 2: #\$DBUS ('7E24') Rückgabe: '020000F1' = 24.1° bei "DEZ TEMP/A/BUS" Beispiel 3: #\$DBUS ('118001') Schreibt auf Datenadresse.0, 1 Byte mit Wert 1 an Geräte 17 z.B. "DEZ BUSIO" .	Diese Funktion dient zum Schreiben und Lesen auf dem DigiWEB "IoBUS" bzw. "D-BUS" für z.B. "DEZ TEMP/R/BUS" , "DEZ TEMP/A/BUS" , "DEZ CO2/BUS" , "DEZ DT1" oder "DEZ BUSIO" . Das 1.Byte des HEX - Strings ist die Geräteadresse, das 2.Byte ist das Kommando und dann die zu schreibenden Daten. Aufbau Kommandobyte: Bit 7 - 1 = schreiben / 0 = lesen Bit 5+6 - Datenlänge (0=1 / 1=2 / 2=4 / 3=8 Byte) Bit 0..4 - Adresse Die Datenadresse ist = (Adresse * Datenlänge) Als Rückgabe kommt als 1. Byte der Status (2 = OK, 3 = Error bzw. Geräteadresse doppelt oder 4 = Error Gerät nicht gefunden). Ab dem 3. Byte kommen die gelesenen Daten. Hinweis: Der "IoBUS" bzw. "D-BUS" muß hierzu im Register ##000106b0_7 aktiviert bzw. auf einen 24V Ein - Ausgang (Modul: X) gelegt werden. Sehen Sie hierzu auch Register #\$000226

3.2.6. Die MODBUS Funktionen

Funktion	Beschreibung
<p>##MODBUS(IP,Port,Reg,Time,Value,FC,RS) Rückgabe: Werte <= 16 Bit == erfolgreich gelesener Wert Rückgabe: 0x80000000 == Variable nicht bzw. noch nicht verbunden/gelesen Rückgabe: 0x80000001 == Speicherüberlauf</p> <p>Beispiel 1: ##000900:=##MODBUS(0,15,10,2000) Register 10 von Adr. 15 alle 2000ms lesen.</p> <p>Beispiel 2: ##MODBUS(0,15,7,5000,0,0x0501) Bit 7 von Bitadresse (Coil) alle 5000ms lesen.</p> <p>Beispiel 3: ##MODBUS(0,40,50,10000,0,0x0003,32839) Leistung eines Finder Energiezählers mit Modbus RTU, Adresse 40, Register 50, alle 10 Sekunden, als Read Holding Register auslesen. Baudrate ist auf 96/8n2 eingestellt.</p>	<p>Diese Funktion erzeugt einen Liste bzw. Tabelle mit Einträgen zum zyklischen Variablen austausch via MODBUS Protokoll.</p> <p>Die Funktion blockiert nicht beim Lesen oder Schreiben. Sie initiiert ein Hintergrundprogramm, das zyklisch liest oder schreibt.</p> <p>Parameter: (max. 7 mit 32 Bit) IP: IP bei Modbus TCP 0 = Modbus RTU mit RS485 Port: Zielport bei Modbus TCP Zieladresse 1-247 bei Modbus RTU (0=Broadcast). Reg: Nummer des Modbus Registers (0 od. 1 bis 65535) Time: bestimmt den Lesezyklus im ms. Mit Werten < 0 wird die Variable einmalig geschrieben, der negative Wert ist dann ebenfalls die Lesezykluszeit. Ist das LSB gesetzt, wird nur geschrieben, wenn der gelesene Wert und der zu schreibende Wert ungleich sind. 0 beendet das zyklische Lesen und die Variable wird aus der Transferliste entfernt. Value: Wert beim Schreiben des Registers FC: Function Codes 0x0 = Kompatibilitätsmode / Write Single Holdingregister / Read Holdingregisters 0x0501 = Write Single Coil / Read Coils (Bits) 0x0F01 = Write Multiple Coil / Read Coils (Bits) 0x0002 = Read Discrete Inputs 0x0603 = Write Single Holding Register / Read Holding Registers 0x1003 = Write Multiple Holding Register / Read Holding Registers 0x0004 = Read Input Registers RS: RS485 Parameter 0 = default 9600/8/n/1 <> 0 = siehe Register ##000130 auf Seite 37</p> <p>Beispiele für den Parameter RS bzw. die Baudrate:</p> <p>71 = Baudrate: 9.6k 8n1 8263 = Baudrate: 9.6k 8e1 16455 = Baudrate: 9.6k 8o1</p> <p>32839 = Baudrate: 9.6k 8n2 41031 = Baudrate: 9.6k 8e2 49223 = Baudrate: 9.6k 8o2</p> <p>35 = Baudrate: 19.2k 8n1 8227 = Baudrate: 19.2k 8e1 16419 = Baudrate: 19.2k 8o1</p>

3.2.6.1. Erweiterung Parameter 'Port' bei MODBUS TCP Port - Nummer

Bei MODBUS TCP wurde das Register für den TCP Port erweitert. Hierbei wird nun im Bit 0 -15 wie bisher der TCP Port angegeben.

Bei machen MODBUS TCP Slaves muss zusätzlich eine Geräte - ID bzw. ein Unit - Identifier (default Wert ist = 0) angegeben werden. Dieser wird in den Bits 16 - 23 gesetzt.

Syntax Beispiel: **'(25<<16) + 502'** Hier wäre die 25 die Unit - ID und die 502 der TCP Port.

3.2.7. Die Exchange Funktionen bzw. Variablen austausch

Funktion	Beschreibung
<p>##ExchangeInsert('V1', 'V2', Time, 'Host', Dir.) Rückgabe: == Werte 32 Bit Der zurückgegebene 32 Bit Wert enthält die Verbindungsnummer (Handle). Über diese lässt sich die Verbindung Überwachen oder Löschen.</p> <p>Tipp: Die Verbindungsnummer sollten Sie sich in einer nicht remanenten DigiWEB Register - Variablen merken (ab ##000512) wenn Sie die Verbinden später überwachen oder löschen wollen.</p> <p>Hinweis: Alle Verbindungen werden bei einem Neustart des DigiWEB gelöscht.</p>	<p>Diese Funktion erzeugt einen Liste bzw. Tabelle mit Einträgen zum zyklischen Variablen austausch via TCP/IP Protokoll zwischen den DigiWEB WEB Servern.</p> <p>Die Funktion blockiert nicht, Sie initiiert ein Hintergrund - Programm das zyklisch liest oder schreibt.</p> <p>Parameter: V1: Variable des DigiWEBs die das ##ExchangeInsert() ausgeführt. z.B. "##Value[0]" V2: Variable des Host bzw. entfernten DigiWEBs. z.B. "##Aussen.F" Time: Lese - oder Schreibzyklen im ms. Host: Hostname bzw. IP des entfernten DigiWEBs inklusive Usernamen und Passwort. z.B. "Username:Password@192.168.178.2" Dir: Auswahl Datenrichtung bzw. ob die Variable gelesen oder geschrieben werden soll. 0 = Lesen (V1 = V2) 1 = Schreiben (V2 = V1)</p> <p>Achtung: Es können als V1 und V2 Parameter nur Variablen oder Symbole angegeben werden, Funktionen oder Ausdrücke wie z.B. ##000512*2 oder ##PLCBITGET(0,'O') dürfen/können hier nicht angegeben werden.</p>
<p>##ExchangeError(Handle) Rückgabe: == Werte 32 Bit Statuscode Siehe Kapitel "6. Fehlermeldungen bzw. Error Codes" aus Seite 50.</p>	<p>Mit dieser Funktion kann der aktuelle Status des Datentransfers gelesen werden.</p> <p>Hierzu muss die Verbindungsnummer bzw. das Handle als Parameter übergeben werden, das durch die Funktion ##ExchangeInsert() zurück gegeben wurde.</p>
<p>##ExchangeDelete(Handle)</p>	<p>Mit dieser Funktion wird eine Verbindung gelöscht.</p> <p>Hierzu muss die Verbindungsnummer bzw. das Handle der zu löschenden Verbindung als Parameter übergeben werden.</p>

Hinweis: Als Timeout für diese Funktionen wird ab Firmware V4.047 (4/2016) der Wert aus Register "##000044" verwendet. Vor dieser Version wurde das Timeout durch das TCPIP Timeout aus Register "##000167B0_7" geteilt durch 4 errechnet (default 1.75 Sek.).

Achtung: Die Funktion ##ExchangeInsert() darf **keinesfalls** in der DigiWEB "advanced" Konfiguration "<http://DigiWEB-IP/config/>" unter dem Menüpunkt: "**Variables**" verwendet werden. Verwenden Sie hier nur die Funktion ##ExchangeGet() !

Tipp: Die Funktion ##ExchangeInsert() sollte in der Script - Datei "/autoexec.req" oder bei DigiENERGY bzw. Bluebox in der Script - Datei "/oem/autoexec.req" eingefügt werden, damit der Variablen austausch beim Starten des DigiWEBs automatisch aktiviert wird.

Beispiel:

Das folgende Script erzeugt einen Eintrag in der DigiWEB <-> DigiWEB Variablenübertragungstabelle.

Hier wird z.B. die Temperatur des 2. Raums, des 1. Heizkreises eines DigiENERGY gelesen und dann in das ##Value[0] Register der virtuellen Anschlüssen des eigenen DigiWEBs geschrieben.

Die Verbindungsnummer bzw. das Handle wird hierbei im Register ##000600 gespeichert, um z.B. bei einem weiteren Aufruf des Scripts den Variablen austausch wieder löschen zu können oder an anderer Stelle eine Statusüberprüfung durchführen zu können.

```
<!--#if(!##000600)--> Variablen austausch hinzugefügt
  <!--##000600:=##ExchangeInsert('##Value[0]', '##Raum[0].Einzel[1].F', 10000, 'ftp:ftp@10.10.1.1', 0)-->
<!--#else--> Variablen austausch gelöscht
  <!--##ExchangeDelete(##000600)-->
  <!--##000600:=0-->
<!--#endif-->
```

3.2.7.1. Die ##ExchangeGet() Funktion zum Variablen austausch

Funktion	Beschreibung
<p>##ExchangeGet('V1', Time, 'Host')</p> <p>Rückgabe: == 0 Verbindung noch nicht aktiv == Wert (32Bit)</p> <p>#\$ExchangeGet('V1', Time, 'Host')</p> <p>Rückgabe: == leer String Verbindung noch nicht aktiv == Sting</p> <p>Beispiel 1: Dieses Beispiel liest alle 500ms den Timer des DigiWEB mit der IP - Adresse: 192.168.178.36</p> <p>##ExchangeGet('##000187',500,'ftp:ftp@192.168.178.36')</p> <p>Beispiel 2: Lesen der Versionsnummer: #\$ExchangeGet('#\$000199',500,'ftp:ftp@192.168.178.36')</p> <p>Hinweis: Alle Verbindungen werden bei einem Neustart des DigiWEB gelöscht.</p>	<p>Diese Funktion erzeugt eine Liste bzw. Tabelle mit Einträgen zum zyklischen Variablen austausch via TCP/IP Protokoll zwischen den DigiWEB WEB Servern.</p> <p>Die Funktion blockiert nicht, sondern initiiert ein Hintergrund - Programm das zyklisch liest. Schreiben ist mit dieser Funktion nicht möglich.</p> <p>Als Returnwert erhalten Sie nach erfolgreichem Verbindungsaufbau den Variablenwert des entfernten DigiWEBs</p> <p>Parameter: V1: Variable des Host bzw. entfernten DigiWEBs. z.B. "##Aussen.F"</p> <p>Time: Lesezyklen in ms.</p> <p>Host: Hostname bzw. IP des entfernten DigiWEBs inklusive Usernamen und Passwort. z.B. "Username:Password@192.168.178.2"</p> <p>Achtung: Es können als V1 Parameter nur Variablen oder Symbole angegeben werden, Funktionen oder Ausdrücke wie z.B. ##000512*2 oder ##PLCBITGET(0,'O') dürfen/können hier nicht angegeben werden.</p>
<p>##ExchangeGetErr('V1', Time, 'Host')</p> <p>Rückgabe: == Werte 32 Bit Statuscode Siehe Kapitel "6. Fehlermeldungen bzw. Error Codes" aus Seite 50.</p>	<p>Mit dieser Funktion kann der aktuelle Status des Datentransfers gelesen werden.</p> <p>Hierzu müssen die gleichen Parameter wie beim Aufrufen der Funktion ##ExchangeGet() bzw. #\$ExchangeGet() angegeben werden.</p>
<p>##ExchangeGetDel('V1', Time, 'Host')</p> <p>Rückgabe: == 0 = Fehler 1 = Erfolgreich</p>	<p>Mit dieser Funktion wird eine Verbindung gelöscht.</p> <p>Hierzu müssen die gleichen Parameter wie beim Aufrufen der Funktion ##ExchangeGet() bzw. #\$ExchangeGet() angegeben werden.</p>

3.2.8. Die PLC (SPS) Funktionen

Die Funktion der integrierten SPS - Logik - Funktion des DigiWEB steht erst ab den Prozessoren bzw. den CPU Typen der 4. DigiWEB Generation zur Verfügung.

Hierzu zählen alle DigiWEBs ab DigiWEB 4 sowie DigiWEB 3 im Ausbau **„.R2“** bzw. mit CPU3.1.

Hinweis: Die hier beschriebenen Funktionen stellen nur die Funktionen dar, die zur Steuerung und Abfrage der SPS - Logik - Funktion durch ein Script sinnvoll zu verwenden sind.

Funktion	Beschreibung
<p>##PLCBitGet(Nummer, 'Typ') Rückgabe: == 0 / 1</p> <p>Tipp: Um den Zugriff beim Programmieren im Script zu vereinfachen, sollten Sie ein indiziertes Symbol mit der entsprechenden Expression verwenden.</p> <p>Beispiel: Symbolname: ##VBIT[##A] Read Exp. ##PLCBITGET((##A-1),'V')</p>	<p>Mit dieser Funktion wird der Status der Ein - bzw. Ausgangsbits sowie der Merker des SPS - Logik - Moduls gelesen.</p> <p>Zur Zeit können folgende Bit - Typen gelesen werden: 'O', 'P', 'M', 'X', 'N', 'V', 'S', 'I'</p> <p>Parameter: Nummer: Bitnummer im jeweiligen Bereich bzw. Typ. (0 = 1. E/A Signal) Typ: Typenkennung z.B. 'I' = Eingang</p>
<p>##PLCBitSet(Value,Nummer,'Typ') Rückgabe: == immer 0</p> <p>Tipp: Verwenden Sie auch hier ein indiziertes Symbol für den Zugriff.</p> <p>Beispiel: Symbolname: ##VBIT[##A] Read Exp. ##PLCBITGET((##A-1),'V') Write Expression: ##PLCBITSET(##INPUT?1:0,(##A-1),'V')</p>	<p>Mit dieser Funktion wird ein Bit im SPS - Logik - Modul geschrieben.</p> <p>Zur Zeit können nur die Bits vom Typ 'V' geschrieben werden:</p> <p>Parameter: Value: 0 oder 1 0 = Bit wird aus - bzw. 1 = Bit wird eingeschaltet.</p> <p>Nummer: Bitnummer im 'V' - Bereich. (0 = V-Eingang 1)</p> <p>Typ: z.Z. immer nur 'V' möglich.</p>
<p>#\$PlcVisual('HEXSTRING') Rückgabe: == HEX String</p> <p>Beispiel: ##HEXTOLONG((#\$PlcVisual('0800')-16)%8)</p> <p>Hierdurch wird der Zählerstand des Zählers in Merker 1 gelesen.</p>	<p>Mit dieser Funktion wird der Status eines Netzwerks gelesen.</p> <p>Hierzu zählen Istwerte von Zählern und Zeiten inklusive aller aktiver Strompfadsymbole.</p> <p>Parameter: Der HEXSTRING enthält die Bereichskennung und die Bitnummer des gewünschten Netzwerks in HEX (je 8 Bit).</p> <p>'00XX' Istwert aus dem O-Bereich, XX = Bitnummer '04XX' Istwert aus dem P-Bereich, XX = Bitnummer '08XX' Istwert aus dem M-Bereich, XX = Bitnummer '0CXX' Istwert aus dem X-Bereich, XX = Bitnummer</p>

Hinweis: Die Verknüpfungen bzw. die Netzwerke, Kommentare und Symbolnamen werden in der Datei /plc.csv im Root - Verzeichnis des DigiWEB gespeichert.

3.2.9. Die Debug Funktionen

Funktion	Beschreibung
<pre>#\$PEEK(ADR, LEN) Rückgabe: : HEX String Beispiel: <!--##ADR:=0--><!--##Step:=32--> <!--#while(##ADR<150000)--> <!--#\$PEEK(##ADR,##Step)-->#* <!--##ADR:=##ADR+##Step--> <!--#endwhile--></pre>	<p>Diese Funktion dient zum Auslesen beliebiger Speicherzellen (ADR) im RAM und Flash des DigiWEB. Je Anfrage können max. 196 Byte (LEN) gelesen werden. Es sind Supervisor bzw. FTP - Rechte notwendig.</p>
<pre>##DEBUG(Mode) Rückgabe: : 32 Bit Wert</pre>	<p>Mit dieser Funktion können Debug Variablen des DigiWEB gelesen werden. Der Parameter Mode wählt die gewünschte Variable aus.</p> <ul style="list-style-type: none"> 0 = Anzahl der angeforderten Speicherblöcke (Malloc counter). 1 = Anzahl Bytes der angeforderten Speicherblöcke (used). 2 = Freier Platz für Speicherblöcke . 3 = Frei count 4 = Max malloc size 5 = Free malloc size 10 = Stackspeicher (Size) maximal. 11 = Bisher maximal verbrauchter Stackspeicher. 30 = Verbrauch des Ticker im Durchschnitt in Promille 31 = Maximum Ticker in den letzten 4 Sek. in Promille 32 = Verlorene Ticker 33 = Verzögerung beim Ticker (Jitter) 40 = TCPIP MSS size. 41 = Status "Easy access Server" (1 = online / 0 = offline) Ab hier nur DW4 55 = Anzahl der Kommunikationsfehler auf dem DW4 IR BUS 60 = DBUS Daten auf serielle Schnittstelle ausgeben OFF 61 = DBUS Daten auf serielle Schnittstelle ausgeben ON

4. Die Interne Datentabelle bzw. die Register/Variablen des DigiWEB

In der Internen Datentabelle des DigiWEB (Datenbereich 0 der RK512 Tabelle) werden die Konfigurations - und Statusdaten des DigiWEB gespeichert bzw. gelesen und geschrieben.

Die Datentabelle ist in 1000 Register unterteilt (ab DigiWEB 2 bis zu 15999 Register, je nach Speicher und Software), wobei die Konfigurationsregister und die Register von 256 bis 511 remanent gespeichert werden. Ein Register kann mit einem Wert (32 Bit) oder mit einem String bzw. einer Zeichenkette belegt sein.

Soll das Register als String gelesen werden, so muß das zweite Doppelkreuz in der Adresse durch ein Dollarzeichen ersetzt und die zu lesende Länge in Byte mit angegeben werden.

Beispiel: ##000256 = Wert oder #\$000256b10 = String mit 10 Zeichen
Der String belegt nicht nur Register #\$000256, sondern auch 000257 und die Hälfte (2 Byte) von Register 000258.

Hinweis: Bei den Registern mit Nummern kleiner als 256 (DigiWEB Konfigurationsregister) und dem \$ (= String) vor der Registernummer ist keine Längenangabe notwendig bzw. möglich, da dies spezial Register des DigiWEB sind.

4.1. DigiWEB Applikation Variablen bzw. Symbole

Ab der DigiWEB OS Version 4.105 ist es möglich, die Variablen - bzw. Symbolnamen der auf dem DigiWEB installierten Applikation wie z.B. DigiENERGY oder bluebox durch den Aufruf der "**app.json**" Datei auszulesen. Diese erfordert Administrator - Berechtigung durch Username und Passwort.

<http://192.168.2.17/app.json?symbolinfo&user=Service&password=Service>

oder

<http://192.168.178.42/app.json?symbolinfo&user=ftp&password=ftp>

oder via EMS

<https://ems.digitronic.com/0050c21e2be9/app.json?symbolinfo&user=ftp&password=ftp>

Als Antwort erhalten Sie eine im JSON Format kodierte Datei mit allen Symbolen bzw. Variablen - Namen der installierten Applikation (siehe Bild rechts).

Hierbei sind die Variablen in "children" mit einer Element - Nummer und den entsprechenden Eigenschaften aufgelistet:

symbolName: Der eigentliche Symbol - bzw. Variablen - Name.

comment: Kommentar bzw. Beschreibung.

security: Notwendige Berechtigung zum Lesen bzw. Schreiben.

type: Symbol ist eine Zahl (#) bzw. String (\$) - Variable.

arraysize: Variable hat einen Index [] von 0 bis +n.

children: Weitere zusätzliche Member - bzw. Sub - Variable des Symbols z.B. **##Brenner[1].Aktiv**

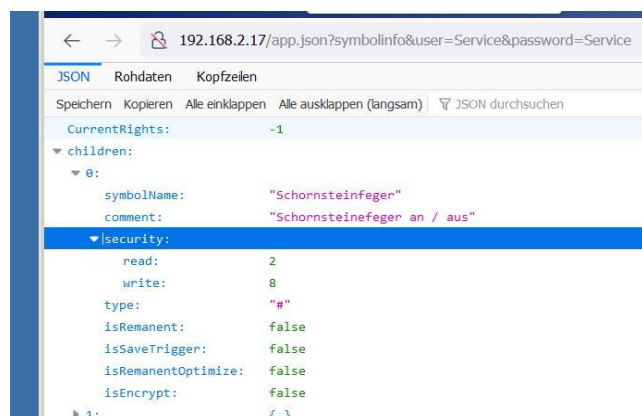
isRemanent: Variable wird remanent bzw. netzspannungsausfallsicher im Flash - Speicher gespeichert, wenn dies aus "true" steht.

! Achtung: Remanente Variablen dürfen nicht ständig geschrieben werden, da hierdurch der Flash - Datenspeicher überlastet und zerstört wird!

isEncrypt: Variable wird verschlüsselt gespeichert, z.B. Passwörter, und können nicht gelesen sondern nur geschrieben werden.

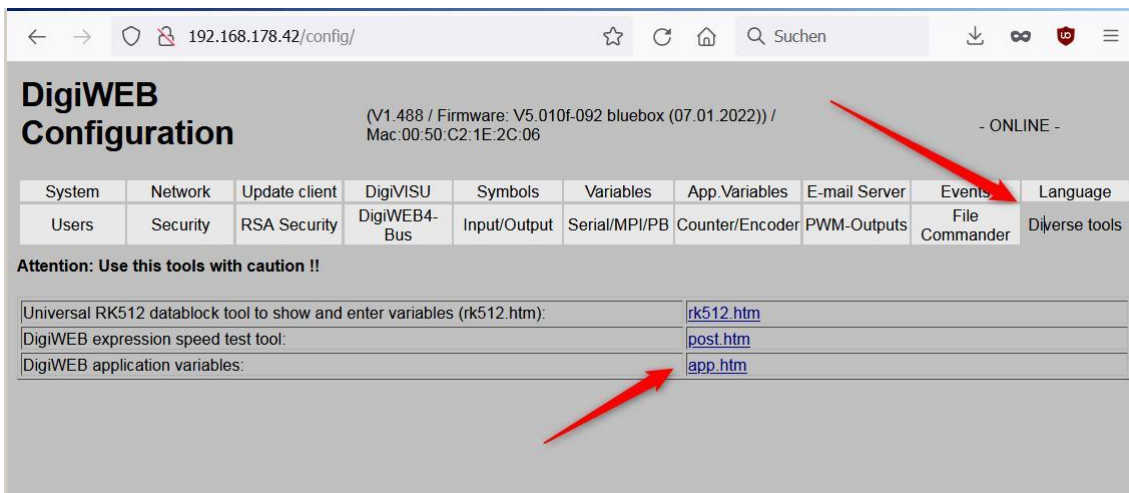
Hinweis: Die Symbol - Eigenschaften: **isSaveTrigger** und **isRemanentOptimize** sind bei externem Zugriff nicht von belang.

TIP: In der erweiterten DigiWEB Konfiguration ist ab 2/2022 ein Menüpunkt für die Applikation Variables hinzu gekommen, der die "app.json" darstellt, siehe nächste Seite.



4.1.1. DigiWEB App - Variables in der erweiterten Konfiguration

In der erweiterten DigiWEB Konfiguration ist ab 2/2022 unter dem Menüpunkt für "Diverse tools" die Seite "app.htm" hinzugekommen.



Hier wird die mit dem "app.json?symbolinfo" Aufruf gelieferte JSON Datei dekodiert und alphabetisch sortiert dargestellt.

Variable name / Number of variables: 762	Value	Read	Comment
##Alarmanlage Minuten	1440	<input type="button" value="Read"/>	60 min Alarm Nachlaufzeit
##Alarmanlage OR[0 .. 31]	1	<input type="button" value="Read"/>	Lichtkreise die bei Alarm eingeschaltet werden //standardwert war 0
##Alarmanlage.Timer	0	<input type="button" value="Read"/>	Alarmtimer, solange laeuft der Alarm noch (in sec)
##Alarmanlage.Disable		<input type="button" value="Read"/>	Alarmanlage disable (zum wegnehmen der Alarmbeleuchtung trotz Konfiguration in Matrix oder via Kundenzaehler) ;default 0 Alarmanlage betriebsbereit
#\$Alarmanlage.ArmedRemZaehler.Parameters		<input type="button" value="Read"/>	remanent counter parameter. 4-Hex-Byte little endian index, n-Hex-Byte offset value
##Alarmanlage.Delay		<input type="button" value="Read"/>	Sekunden um den Alarm entschaerfen zu koennen ;default 0 -> Licht geht sofort an
##Alarmanlage.Modebits		<input type="button" value="Read"/>	Modebits für den Alarm, 0x01=Alarm Kundenzaehler auch als Alarmeingang, 0x02=Audiosystem gibt den Einbruchalarm aus
##Alarmanlage.Armed		<input type="button" value="Read"/>	Alarmanlage ist scharf geschaltet
##Alarmanlage.ArmingTimer		<input type="button" value="Read"/>	Restzeit in Sekunden bis die Alarmanlage scharf ist
##Alarmanlage.TriggerTimer		<input type="button" value="Read"/>	Restzeit in Sekunden bis der Alarm ausgeloeset wird
##Alarmanlage.Is		<input type="button" value="Read"/>	Alarmanlage mit Verzoeigerung ist konfiguriert
##Anwesend_VisuColor		<input type="button" value="Read"/>	Farbe Anwesend
##Anwesenheit.ROR[0 .. 31]		<input type="button" value="Read"/>	Rotierende Lichtkreise //standardwert war 0
##Anwesenheit.OFF[0 .. 31]		<input type="button" value="Read"/>	Lichtkreise die bei Abwesenheit abgeschaltet werden //standardwert war 0
##Anwesenheit.Szene		<input type="button" value="Read"/>	Szene bei langem Druicken des Anwesenheitstaster (0= Disable, 1-10 = Szene[0]-Szene[9])
##Anwesenheit.Extra		<input type="button" value="Read"/>	Sind wir im Anwesenheit Extra Mode?
##Anwesenheit.SparenSzene		<input type="button" value="Read"/>	SparenSzene wird aktiviert wenn Schaltuhr auf ExtraSzene steht und Abwesend aktiv ist (0= Disable, 1-10 = Szene[0]-Szene[9])
##Anwesenheit.Sparen		<input type="button" value="Read"/>	Sind wir im SparenMode? (Abwesend während in der Schaltuhr die ExtraSzene eingestellt ist)
#\$Anwesenheit.SzeneSelectFeldOption		<input type="button" value="Read"/>	SelectFeldOption fuer Szenen
#\$Anwesenheit.TasterRemZaehler.Parameters		<input type="button" value="Read"/>	remanent counter parameter. 4-Hex-Byte little endian index, n-Hex-Byte offset value
#\$Audio.Mac		<input type="button" value="Read"/>	Mac des AudioSystems oder URL
#\$Audio.User		<input type="button" value="Read"/>	User des AudioSystems
#\$Audio.Password		<input type="button" value="Read"/>	Password des AudioSystems
##Audio.IP		<input type="button" value="Read"/>	aus der MAC ermittelte IP
##Audio.ComState		<input type="button" value="Read"/>	Status der Kommunikation
##Audio.Geraet		<input type="button" value="Read"/>	Geraetetyp -1=keines 0=Echion

Zusätzlich kann jede Variable durch Klicken auf den Knopf "Read" einmalig gelesen werden und wird in der Spalte "Value" dargestellt.

Ist die Variable ein Array, kann zusätzlich der entsprechende Index dazu eingegeben werden und durch erneutes Klicken auf "Read" gelesen werden. Liegt der Index ausserhalb des gültigen Bereichs erhalten Sie eine Fehlermeldung.

4.2. Die Netzwerkkonfiguration, Systemvariablen und Systempasswort

Bereich 0 / Register	Bytes / Bits	DigiWEB Netzwerk Konfiguration und Statusanzeigen lesen(r) / schreiben(w) möglich	
43	4	Bei PC DigiWEB bzw. Simulation Sleeptime	(r/w)
44	4	Timeout beim HTTP - Variablentransfer z.B. durch die Funktion ##ExchangeInsert() und ##ExchangeGet() (ab V4.074 Firmware 4/2016)	(r/w)
103	Bits 0..15	ab DW2 = Anzahl der DigiWEB Neustarts bzw. Power UP oder Resets	(r)
149	Bit 0..15 Bits 16..31	Portnummer des DigiWEB Servers (default = 80) (1..65534) 2. Portnummer des Servers ab DW2 (default = 0 = off)	(r/w) >DW1
150	4	DigiWEB IP - Adresse (4 x 8 Bit)	(r/w)
\$151	17	MAC Adresse (formatierter String "00:AA:FF:11:22:33")	(r/w)
152	4	Gateway IP - Adresse (4 x 8 Bit)	(r/w)
153	Bit 0..7 Bit 16..23 Bit 24..31 Bit 24 Bit 25 Bit 26	Anzahl der freien TCPIP Sockets des Servers (je nach Type max. 12) (max. 5 * DigiVISU Projekt und 1 * Config Seite können angezeigt werden) Ethernet Boot-UP Status: 0 = Ethernet nicht angeschlossen 1 = Ethernet angeschlossen, noch keine IP gesetzt 2 = DHCP an, Local Link Adresse gesetzt (z.B. 169.254.x.x/16) 3 = IP gefunden, via DHCP oder fest eingestellt 4 = TCPIP Gateway gefunden Ethernet Schnittstellen Status (Bit codiert) Angeschlossen bzw. LINK (0 = nicht OK / 1 = OK) Duplex Mode (0 = Halb Duplex / 1 = Voll Duplex) Baudrate (0 = 10MBit / 1 = 100MBit)	(r) (r) (r) (r) (r)
154	4	Subnet Mask (4 x 8 Bit)	(r/w)
\$160	10	DigiWEB FTP Login Name (default = 'ftp') (String)	(r/w)
\$161	10	DigiWEB FTP Passwort (default = 'ftp') (String)	(r/w)
164	4	TCP Paketgröße (MTU/MSS) (default 1500)	(r/w)
\$166	100	Hostname für passives FTP bei fester Datenportnummer 33100 im Router	(r/w)
167	Bits 0..7 Bits 8..15	TCPIP Timeout (default = 7 Sekunden...60 Sekunden) Cache - Zeit (Lifetime) der DigiWEB Bilder im Browser (0..240 Minuten)	(r/w) (r/w)
169	4	DNS Server IP - Adresse (4 x 8 Bit)	(r/w)
177	1	Nummer der Standardsprache (0..250)	(r/w)
186	Bit 0 Bit 1 " Bit 2 Bits 3..4 Bit 7 Bit 8 Bit 12 " Bit 13 " Bit 14	Status seriellen Schnittstelle (0 = Fehler / 1 = OK) Status der I/O Schnittstelle (0 = OK / 1 = Ausgangs Fehler) Ausgangsfehler quittieren (1 = Quittieren) DigiWEB Neustart auslösen (mit Shutdown) (1 = DigiWEB Neustarten) DHCP Server Konfiguration (0 = DHCP aus) (1 = DHCP ein) (2 = Nur beim Booten mit bootp Protokoll) DHCP Server (0 = nicht gefunden / 1 = DHCP Server erkannt) Status der MPI Schnittstelle (0 = Fehler / 1 = OK) DigiWEB 4 Hardware - Modulwarnung (0 = OK / 1 = Warnung) (0 = Modulwarnung quittieren) DigiWEB 3/4 Hardware - Modulfehler (0 = OK / 1 = Soll - <- Istausbau) (! 0 = Quittieren -> Istausbau in Sollausbau übernehmen !) SMTP - Server DNS Adresse gefunden (0 = nicht gefunden / 1 = OK)	(r) (r) (w) (w) (r/w) (r) (r) (r) (w) (r) (w) (r)
186	Bit 15 Bit 16 Bit 17 Bits 24..31	DigiWEB Neustart auslösen (1 = DigiWEB Neustarten) (Hinweis: Ohne aufrufen der shutdown.req Datei) - Status der "Easy access" - Server Verbindung (ab Firmware 8/2016) (0 = nicht verbunden / 1 = verbunden) Server Hardware Version: 0 = DW1 1 = DW2 2 = DW3 / CPU Type 2 3 = DigiWEB PC Simulation 4 = DC190 5 = DC190 Bootloader 6 = DW3.1 / CPU Type 4 7 = DW3.1 CPU Bootloader / CPU Type 4 8 = DW4 / CPU Type 4 9 = DW4 Bootloader 10 = DW CT Zeitschaltuhr oder Comptrol 4 WEB V1 / CPU Type 5 11 = Comptrol 4 WEB V2 / CPU Type 6 12 = DW4.5 / CPU Type 7	(w) (r) (r)
187	4	DigiWEB Timer 32Bit in ms Einheiten	(r)

\$199	20	Software Version der DigiWEB Firmware	(String)	(r)
210	4	ab DW2 = Software Version des Bootloaders	(1000 = V1.000)	(r)

Sehen Sie hierzu auch die DigiWEB "**advanced**" Konfiguration "<http://DigiWEB-IP/config>" unter dem Menüpunkt: "**Network**".

4.3. Status Flash - Daten - Speicher

Bereich 0 / Register	Bytes / Bits	DigiWEB Flash Status lesen(r) / schreiben(w) möglich	
148	4	Anzahl defekter Cluster des Flash Speichers	(r)
163	4	Ab Firmware V3.259 09/2012 steht hier die Anzahl der Flash - Speicher - Schreibzyklen.	(r)
198	4	Freier Flash - Speicher	(r)

Hinweis: Im Flash - Daten - Speicher des DigiWEB 1, 2 und 3 können zur Zeit maximal 1000 Dateien gespeichert werden.

4.4. Zeit/Uhrzeit und NTP Serverkonfiguration

Bereich 0 / Register	Bytes / Bits	DigiWEB Uhrzeit - bzw. NTP Server Konfiguration und Status lesen(r) / schreiben(w) möglich	
170	4	IP - Adresse des NTP Zeit - Servers (4 x 8 Bit) oder ab DW2 kann zusätzlich eine Liste von Servern in die "/NTP.CSV" Datenbank eingetragen werden.	(r/w) >DW1
171	4	Zeit in Sekunden seit 1.1.1900 (32 Bit)	(r/w)
\$172	23	Aktuelle Zeit im Format "JJJJ-MM-TT SS:MM:SS" (String)	(r)
173	Bit 0	NTP Time Status (0 = asynchron / 1 = Zeit gesetzt bzw. synchron)	(r)
	Bit 1	1 = Zeit muß nachsynchronisiert werden.	(r)
	Bit 2	1 = Zeitanfrage zum NTP -Server läuft (lesen). 1 = Zeitanfrage zum NTP -Server neu starten bzw. tiggern (schreiben).	(r/w)
	Bit 3	1 = RTC nicht gefunden (lesen) 1 = RTC deaktivieren (schreiben)	(r/w)
	Bit4	1 = RTC nicht initialisiert bzw .deaktiviert	(r)
174	Bit 0	Automatische Sommer/Winterzeit Umschaltung (0 = aus / 1 = ein)	(r/w)
	Bits 0..31	Zeitdifferenz in Sekunden zur GMT/UTC Zeit. (nur gerade Zahlen)	(r/w)
179	4	Eingangsbite an dem der DCF 77 Empfänger angeschlossen ist, um das Funkuhrsignal der PTB zur Uhrzeiteinstellung des DigiWEB zu verwenden. (-1 = off, 0..63)	(r/w) >DW1

Sehen Sie hierzu auch die DigiWEB "advanced" Konfiguration "<http://DigiWEB-IP/config/>" unter dem Menüpunkt: "**Network**".

4.5. E-Mail - Server und Event - Händlerkonfiguration

Hier wird der SMTP Server eingestellt und die Event - Steuerung eingeschaltet. Zusätzlich muß die Datei "smtpfile.csv" im Root - Verzeichniss des DigiWEB vorhanden sein, Welche die Event - Regeln enthält.

Bereich 0 / Register	Bytes / Bits	DigiWEB SMTP Server und Event Händler Konfiguration lesen(r) / schreiben(w) möglich	
\$155	50	SMTP User Name für Auth. SMTP (String)	(r/w)
156	4	SMTP Server IP Adresse (siehe auch Register 168) (4 x 8 Bit)	(r/w)
\$156	100	Ab Firmware V3.273 11/2012 geht auch ein Hostname SMTP Server Host (z.B. smtp.gmail.com) Achtung: Die beiden Methoden dürfen nicht gemischt verwendet werden, da ein Schreibzugriff via ##000156 den Hostnamen zerstört !	(r/w)
\$157	10	SMTP Passwort für Auth. SMTP (String)	(r/w)
\$158	50	SMTP Absender Name (String)	(r/w)
\$159	50	SMTP Absender E-mail Adresse (String)	(r/w)
162	Bits 0..29 Bits 30..31	Eventhändler Pausezeit (in ms) von Check zu Check bzw. zu Eintrag. Dient zur Aktivierung der DigiWEB Event - Funktion (/smtpfile.csv) 0 = Eventhändler ist aktiv, wenn die SMTP Server IP <> 0 ist (default). 1 = Eventhändler ist aktiv, auch wenn keine SMTP Server IP eingeben ist. 2 = Eventhändler ist immer inaktiv/aus, auch wenn Reg.156 <> 0	(r/w) >DW1
168	Bits 0..15	Portnummer des SMTP Servers (siehe Register 156)	(r/w)
178	Bit 0	Ist dieses Bit gesetzt, wird die Datei "/LogEvent.req" aufgerufen, wenn ein SMTP - E-Mail - Event auftritt. Dies dient zur Protokollierung des E-Mail Versands.	(r/w) >DW1

Sehen Sie hierzu auch die DigiWEB "advanced" Konfiguration "<http://DigiWEB-IP/config/>" unter dem Menüpunkt: "**E-Mail Server**" und "**Events**".

4.6. Die Merker des DigiWEB

In diesen Registern können Werte bzw. Hilfsvariablen oder Merker abgelegt werden, die frei verwendet werden können.

Es stehen insgesamt 744 (ab DW2 15744) Merkerregister zu je 32 Bit zur Verfügung. Die Register 256 bis 511 werden hier von remanent bzw. netzspannungsausfallsicher in einem EE - Prom gespeichert.

Bereich 0 / Register	Bytes	DigiWEB Merker - Register lesen(r) / schreiben(w) möglich	
256..511	4	DigiWEB Remanentmerker Bereich	(r/w)
512..999	4	DigiWEB Merkerbereich	(r/w)
1000.. 15999	4	erweiterter Merkerbereich ab DigiWEB 2 nur mit der Standard Firmware. Kann nur indiziert angesprochen werden (z.B. ##000[5100]).	(r/w)

4.7. Die DigiWEB Ein - Ausgangs Schnittstelle

Die Bit - Breite der Register 0..7 und 50..57, über welche die Ein - / Ausgänge gesteuert werden, läßt sich über die Register 40 + 90 auf 8,16,24 und 32 Bit einstellen. Hierdurch ist es z.B. möglich, I/O Karten oder Module, die mehr als 8 Hardware E/A haben, auf ein Register festzulegen (z.B. DC16 I/O, AWA 2x12B oder die DigiWEB 3 AC Power Karte).

Bereich 0 / Register	Bytes / Bits	DigiWEB I/O Konfiguration und I/O Statusanzeige lesen(r) / schreiben(w) möglich	
0	1..4	Ausgangsbereich/Register 1 der I/O Schnittstelle	(r/w)
1	1..4	Ausgangsbereich/Register 2 der I/O Schnittstelle	(r/w)
2	1..4	Ausgangsbereich/Register 3 der I/O Schnittstelle	(r/w)
3	1..4	Ausgangsbereich/Register 4 der I/O Schnittstelle	(r/w)
4	1..4	Ausgangsbereich/Register 5 der I/O Schnittstelle	(r/w)
5	1..4	Ausgangsbereich/Register 6 der I/O Schnittstelle	(r/w)
6	1..4	Ausgangsbereich/Register 7 der I/O Schnittstelle	(r/w)
7	1..4	Ausgangsbereich/Register 8 der I/O Schnittstelle	(r/w)
40	Bits 0..1 Bits 2..3 Bits 4..5 Bits 6..7 Bits 8..9 Bits 10..11 Bits 12..13 Bits 14..15	Ausgangsbereich Konfiguration (Anzahl der Ausgangsbytes bzw. Bitbreite) Bitbreite im Ausgangsbereich 1 (0 = 8 / 1 = 16 / 2 = 24 / 3 = 32 Bits) Bitbreite im Ausgangsbereich 2 (0 = 8 / 1 = 16 / 2 = 24 / 3 = 32 Bits) Bitbreite im Ausgangsbereich 3 (0 = 8 / 1 = 16 / 2 = 24 / 3 = 32 Bits) Bitbreite im Ausgangsbereich 4 (0 = 8 / 1 = 16 / 2 = 24 / 3 = 32 Bits) Bitbreite im Ausgangsbereich 5 (0 = 8 / 1 = 16 / 2 = 24 / 3 = 32 Bits) Bitbreite im Ausgangsbereich 6 (0 = 8 / 1 = 16 / 2 = 24 / 3 = 32 Bits) Bitbreite im Ausgangsbereich 7 (0 = 8 / 1 = 16 / 2 = 24 / 3 = 32 Bits) Bitbreite im Ausgangsbereich 8 (0 = 8 / 1 = 16 / 2 = 24 / 3 = 32 Bits)	(r/w)
41..49	-	Reserve	
50	1..4	Eingangsbereich/Register 1 der I/O Schnittstelle	(r)
51	1..4	Eingangsbereich/Register 2 der I/O Schnittstelle	(r)
52	1..4	Eingangsbereich/Register 3 der I/O Schnittstelle	(r)
53	1..4	Eingangsbereich/Register 4 der I/O Schnittstelle	(r)
54	1..4	Eingangsbereich/Register 5 der I/O Schnittstelle	(r)
55	1..4	Eingangsbereich/Register 6 der I/O Schnittstelle	(r)
56	1..4	Eingangsbereich/Register 7 der I/O Schnittstelle	(r)
57	1..4	Eingangsbereich/Register 8 der I/O Schnittstelle	(r)
90	Bits 0..1 Bits 2..3 Bits 4..5 Bits 6..7 Bits 8..9 Bits 10..11 Bits 12..13 Bits 14..15	Eingangsbereich Konfiguration (Anzahl der Eingangsbytes bzw. Bitbreite) Bitbreite im Eingangsbereich 1 (0 = 8 / 1 = 16 / 2 = 24 / 3 = 32 Bits) Bitbreite im Eingangsbereich 2 (0 = 8 / 1 = 16 / 2 = 24 / 3 = 32 Bits) Bitbreite im Eingangsbereich 3 (0 = 8 / 1 = 16 / 2 = 24 / 3 = 32 Bits) Bitbreite im Eingangsbereich 4 (0 = 8 / 1 = 16 / 2 = 24 / 3 = 32 Bits) Bitbreite im Eingangsbereich 5 (0 = 8 / 1 = 16 / 2 = 24 / 3 = 32 Bits) Bitbreite im Eingangsbereich 6 (0 = 8 / 1 = 16 / 2 = 24 / 3 = 32 Bits) Bitbreite im Eingangsbereich 7 (0 = 8 / 1 = 16 / 2 = 24 / 3 = 32 Bits) Bitbreite im Eingangsbereich 8 (0 = 8 / 1 = 16 / 2 = 24 / 3 = 32 Bits)	(r/w)
186	Bit 1 Bit 1	Status der I/O Schnittstelle (0 = OK / 1 = Ausgangs Fehler) Durch Schreiben einer 1 wird versucht die Fehlermeldung zu quittieren	(r) (w)

Die Ein - und Ausgangsbits der I/O - Karten liegen, je nach Anzahl der Hardware Ein - Ausgänge der Karte selbst, nach Bit - Breiten - Einstellung (Register 40 + 90), nach Position der I/O Karte im BUS und nach Anzahl weiterer Module in anderen Registern der Ein - / Ausgangsbereiche.

Beispiel 1: DigiWEB 2 mit zwei DC16 I/O und ein AWA 2x12b Modul, Register 40 + 90 = 0:

Hardware Ausgänge	Register	Hardware Eingänge	Register	AWA 2x12B Bits	Register
1. DC16 I/O - 1..8	0 - Bit 0..7	1. DC16 I/O - 1..8	50 - Bit 0..7	Kanal 1 - MSB	54 - Bit 0..7
1. DC16 I/O - 9..11	1 - Bit 0..7	1. DC16 I/O - 9..11	51 - Bit 0..7	Kanal 1 - LSB	55 - Bit 0..7
2. DC16 I/O - 1..8	2 - Bit 0..7	2. DC16 I/O - 1..8	52 - Bit 0..7	Kanal 2 - MSB	56 - Bit 0..7
2. DC16 I/O - 9..11	3 - Bit 0..7	2. DC16 I/O - 9..11	53 - Bit 0..7	Kanal 2 - LSB	57 - Bit 0..7

Beispiel 2: DigiWEB 3 mit zwei 24 Volt I/O - Module und einer AC Karte (Best.Nr.: "DW3 XXP").
Register 40 = 5 + Register 90 = 0:

Hardware Ausgänge	Register	Hardware Eingänge	Register	Hardware Ausgänge	Register
1. Module (X) 1..8	0 - Bit 0..7	1. Module (X) 1..8	50 - Bit 0..7	3. Modul (P) 1..8	1 - Bit 0..7
2. Module (X) 1..8	0 - Bit 8..15	2. Module (X) 1..8	51 - Bit 0..7	3. Modul (P) 9..16	1 - Bit 8..15

Sehen Sie hierzu auch die DigiWEB "advanced" Konfiguration "<http://DigiWEB-IP/config/>" unter dem Menüpunkt: "Input/Output".

4.7.1. Die Puls - Weiten - Modulation (PWM) Funktion der Ausgänge

Die PWM Funktion der Ausgänge steht ab DigiWEB 2 für bis zu 8 Ausgänge zur Verfügung, wobei die Ausgangsnummer frei festgelegt bzw. definiert werden kann.

Bereich 0 / Register	Bytes / Bits	DigiWEB Puls - Weiten - Steuerung der Ausgänge lesen(r) / schreiben(w) möglich	
200	Bits 0..7	Ausgangsbit des 1. PWM Registers (Bit 0..63 = Ausgang 1..64)	(r/w)
	Bits 8..15	Ausgangsbit des 2. PWM Registers (Bit 0..63 = Ausgang 1..64)	(r/w)
	Bits 16..23	Ausgangsbit des 3. PWM Registers (Bit 0..63 = Ausgang 1..64)	(r/w)
	Bits 24..31	Ausgangsbit des 4. PWM Registers (Bit 0..63 = Ausgang 1..64)	(r/w)
201	Bits 0..7	Ausgangsbit des 5. PWM Registers (Bit 0..63 = Ausgang 1..64)	(r/w)
	Bits 8..15	Ausgangsbit des 6. PWM Registers (Bit 0..63 = Ausgang 1..64)	(r/w)
	Bits 16..23	Ausgangsbit des 7. PWM Registers (Bit 0..63 = Ausgang 1..64)	(r/w)
	Bits 24..31	Ausgangsbit des 8. PWM Registers (Bit 0..63 = Ausgang 1..64)	(r/w)
202 (Wird nicht remanent gespeichert und kann nicht gelesen werden bzw. enthält teilweise nicht die geschriebenen Werte)	Bits 0..15	Einstellung für 1. PWM Register	
		Länge des low Impulses in ms Sekunden bzw. wenn der Wert > 60000 ist, dann in Sekunden (z.B. 60001 = 1 Sekunde)	(w)
	Bits 16..31	Länge des high Impulses in ms Sekunden bzw. wenn der Wert > 60000 ist, dann in Sekunden (z.B. 60015 = 15 Sekunde)	(w)
		0x000000 = Deaktivieren (letzter Impuls wird noch abgearbeitet)	(w)
	Bits 0..31	0xffffffe = Reset (um neuen Zeitwert direkt zu übernehmen)	(w)
	Bits 0..31	Ausgang statisch setzen (0xffff0000 = ein / 0x000ffff = aus)	(w)
	Bits 0..31	0xffffXXX Ausgangsimpuls high erzeugen (XXXX = Zeitwert)	(w)
Bits 0..31	0XXXXffff Ausgangsimpuls low erzeugen (XXXX = Zeitwert)	(w)	
203..209	wie Reg.202	Einstellung für 2. bis 8. PWM Register wie Reg.202	(w)

Sehen Sie hierzu auch die DigiWEB "advanced" Konfiguration "<http://DigiWEB-IP/config>" unter dem Menüpunkt: "PWM-Outputs".

Achtung: Es wird nur die Ausgangsnummer remanent gespeichert, die Puls - Weiten - Parameter müssen nach einem Neustart des DigiWEB wieder neu gesetzt werden.

Beispiel 1: Pumpenansteuerung PulsPaket von 10% - 90% mit einem Mindestimpuls von 70ms.

Definieren Sie zunächst ein Symbol mit zwei Parametern: z.B.:##PWM[##R][##O]. Der Parameter ##R ist hierbei die Nummer des PWM Registers (0-7) und ##O die Ausgangsnummer (1-64). Geben Sie anschließend folgende Expressions ein:

Read Expression:

```
(##000[202+##R]b16_31*1000/(##000[202+##R]b16_31+##000[202+##R]b0_15+1)+5)/10
```

Write Expression:

```
##000[200+##R/4]b[0+##R%4*8]_[7+##R%4*8]:=##O-1;##000[202+##R]:=  
##INPUT<1?0x0000ffff:##INPUT>90?0xffff0000:(##INPUT>50?70+(##INPUT*70/(100-  
##INPUT)<<16):4587520+(100-##INPUT)*70/(##INPUT)
```

Durch die Zuweisung ##PWM[0][1]=50 wird nun der 1. Ausgang des DigiWEB auf 50% PWM Modulation gesetzt. Hierdurch schaltet der Ausgang 70ms ein und 70ms aus. Bei einem Wert von 25% beträgt die Einschaltzeit 70ms und die Ausschaltzeit 210ms, bei 75% beträgt die Einschaltzeit 210ms und die Ausschaltzeit 70ms

Beispiel 2: Analoger Ausgang via externen DA - Wandler 0 - 10V Type: "D PWM-ANALOG".

Definieren Sie zunächst wieder ein Symbol: z.B.:##A[##R][##O]. Parameter wie unter Beispiel 1 und geben Sie anschließend folgende Expressions ein:

Read Expression:

```
(##000[202+##R]b16_31*1000/(##000[202+##R]b16_31+##000[202+##R]b0_15+1)+5)/10
```

Write Expression:

```
##000[200+##R/4]b[0+##R%4*8]_[7+##R%4*8]:=##O-1;##000[202+##R]:=  
##INPUT<1?0x0000ffff:##INPUT>99?0xffff0000:(##INPUT<<16)+(100-##INPUT)
```

Durch die Zuweisung ##A[0][8]=50 wird der 8. Ausgang auf 50ms Ein - und Ausschaltzeit gesetzt, wodurch am PWM - Analogmodul 5,0 Volt ausgegeben werden. Bei einem Wert von 25 wird eine Spannung von 2,5 Volt ausgegeben. Die Genauigkeit beträgt +/- 0.1 Volt bzw. 1%.

4.7.2. Die Impulszähler - und SSI - Encoder Eingänge - des DigiWEB

Bereich 0 / Register	Bytes / Bits	Impulszähler Eingangs Funktion und der SSI - Encoder Eingang lesen(r) / schreiben(w) möglich	
91	Bits 0..7 Bits 8..15 Bits 16..23 Bits 24..31	1. Zähler: Typ / Mode 0 = SSI - Encoder 1 = Mono - Impuls - Zähler (nur steigende Flanke). 2 = wie 1 mit remanent Speicherung (nicht mehr unterstützt) 3 = Zähler für Eltako Stromzähler 4 = wie 3 mit remanent Speicherung (nicht mehr unterstützt) 5 = High - Speed - Zähler bei DigiWEB4 (bis 5kHz) (steigende und fallende Flanke wird gezählt). 2. Zähler: " " 3. Zähler: " " 4. Zähler: " " Hinweis: Bei DW1 = 50Hz, ab DW2 = 500Hz maximale Zählfrequenz, ab DW4 mit Unterkarten V5 bis max. 5kHz.	(r/w) >DW1 >DW3
92	Bits 0..7 Bits 8..15 Bits 16..23 Bits 24..31	1. Zähler: Typ 0 = Kappung bei gekaptem (SSI) Graycode 2. Zähler: " " 3. Zähler: " " 4. Zähler: " "	(r/w) >DW1 >DW1 >DW1
93	Bits 0..7 Bits 8..15 Bits 16..23 Bits 24..31	1. Zähler: Typ 0 = die Position des ersten SSI Bits (MSB) im I - Bereich. Typ 1/2 = das Eingangsbit im I - bzw. Eingangs Bereich. 2. Zähler: " " 3. Zähler: " " 4. Zähler: " "	(r/w) >DW1 >DW1 >DW1
94	Bits 0..7 Bits 8..15 Bits 16..23 Bits 24..31	1. Zähler: Typ 0 = Anzahl der SSI Bits die ausgewertet werden = 0..28 Typ 1/2 = Freigabe des Zählers wenn Wert <> 0 2. Zähler: " " 3. Zähler: " " 4. Zähler: " "	(r/w) >DW1 >DW1 >DW1
95	Bits 0..7 Bits 8..15 Bits 16..23 Bits 24..31	1. Zähler: Typ 0 = Position des SSI - Error bzw. Stop - Bits 2. Zähler: " " 3. Zähler: " " 4. Zähler: " "	(r/w)
96	Bits 0..29 Bit 30 Bit 31 Bits 0..31	1. Zähler: Typ 0 = aktueller Istwert 1. Zähler: Typ 0 = SSI - Startbit nicht erkannt. 1. Zähler: Typ 0 = SSI - Errorbit (Stoppbit) nicht erkannt. 1. Zähler: aktueller Zählerstand (kann durch Schreiben gesetzt werden)	(r) (r/w)
97	4	1. Zähler: aktuelle Geschwindigkeit. Dieser Wert wird im DigiWEB durch folgende Formel ermittelt: $\text{Wert} = \text{Impuls}(\text{pro } 10\text{ms}) * 2^{\text{Exponent}} * 2^{\text{Dämpfung}} * 2$ Hinweis: Beachten Sie hierzu das folgende Kapitel.	(r)
98	Bits 0..7 Bits 8..15 Bits 16..23 Bits 24..31	1. Zähler: Dämpfung der Geschwindigkeitsermittlung 2. Zähler: " " 3. Zähler: " " 4. Zähler: " " Hinweis: Beachten Sie hierzu das nächste Kapitel.	(r/w) >DW1 >DW1 >DW1
99	Bits 0..7 Bits 8..15 Bits 16..23 Bits 24..31	1. Zähler: Exponent der Geschwindigkeitsanzeige 2. Zähler: " " 3. Zähler: " " 4. Zähler: " " Hinweis: Beachten Sie hierzu das nächste Kapitel.	(r/w) >DW1 >DW1 >DW1

Achtung: Die I/O Schnittstelle des DW1 und DW2 ist ein RS422 Schnittstelle für die Digitronic IO Module DC16 IO, AWA 2x12b, DAC16 I/U, DC91 IO, DC92 I oder einen SSI - Encoder bzw. Winkelcodierer oder Wegmeßsystem.

Sie arbeitet mit einer Taktfrequenz von ca. 700kHz. Das Anschlußkabel für das SSI - Wegmeßsystem darf keinesfalls länger als 30 Meter sein. Das Anschlußkabel besteht aus einem Adapter mit der Bestellnummer: "**KK EXT/OUT-0.05**" und dem SSI Winkelcodiererkabel: "**KK EXT/S-XX**" (Kabellänge = XX in Meter).

Dieser Hinweis gilt **nicht** für DigiWEB3, da das DW3 ein Modulsystem ist und keine RS422 Schnittstelle hat. Somit können auch kein SSI - Encoder oder DC16 IO Modul angeschlossen werden.

Bereich 0 / Register	Bytes / Bits	Impuls Zähler / Counter lesen(r) / schreiben(w) möglich	all >DW1
100	4	2. Zähler: aktueller Zählerstand (kann durch Schreiben gesetzt werden)	(r/w)
101	4	2. Zähler: aktuelle Geschwindigkeit	(r)
102	4	ab DW2 = Entprellung in 0 - 255ms für die Impuls - Zähler 1 - 8, ab Firmware V3.091 bis 3600000ms,	(r/w)
104	4	3. Zähler: aktueller Zählerstand (kann durch Schreiben gesetzt werden)	(r/w)
105	4	3. Zähler: aktuelle Geschwindigkeit	(r)
106	Bit 0..7	Ausgangsnummer für DigiWEB "ioBUS" Schnittstelle 0 = off / 1..64 = auf Ausgang 1..64 werden die "ioBus" Geräte angeschlossen (z.B. DEZ DT1 oder DEZ BUSIO). Sehen Sie hierzu auch Register \$226 und die #DBUS() Funktion.	(r/w)
107	-	reserve	
108	4	4. Zähler: aktueller Zählerstand (kann durch Schreiben gesetzt werden)	(r/w)
109	4	4. Zähler: aktuelle Geschwindigkeit	(r)
211	Bits 0..7 Bits 8..15 Bits 16..23 Bits 24..31	5. Zähler: Typ (siehe Register ##000091) 6. Zähler: " " 7. Zähler: " " 8. Zähler: " "	(r/w) (r/w) (r/w) (r/w)
212	Bits 0..7 Bits 8..15 Bits 16..23 Bits 24..31	5. Zähler: Typ 0 = Kappung bei gekapptem (SSI) Graycode 6. Zähler: " " 7. Zähler: " " 8. Zähler: " "	(r/w) (r/w) (r/w) (r/w)
213	Bits 0..7 Bits 8..15 Bits 16..23 Bits 24..31	5. Zähler: Typ 1/2 = Eingangsbit im I - bzw. Eingangs Bereich. 6. Zähler: " " 7. Zähler: " " 8. Zähler: " "	(r/w) (r/w) (r/w) (r/w)
214	Bits 0..7 Bits 8..15 Bits 16..23 Bits 24..31	5. Zähler: Typ 1/2 = Freigabe des Zähler wenn Wert <> 0 6. Zähler: " " 7. Zähler: " " 8. Zähler: " "	(r/w) (r/w) (r/w) (r/w)
215	Bits 0..7 Bits 8..15 Bits 16..23 Bits 24..31	5. Zähler: Typ 0 = Position des SSI - Error bzw. Stop Bits 6. Zähler: " " 7. Zähler: " " 8. Zähler: " "	(r/w) (r/w) (r/w) (r/w)
216	4	5. Zähler: aktueller Zählerstand (kann durch Schreiben gesetzt werden)	(r/w)
217	4	5. Zähler: aktuelle Geschwindigkeit	(r)
218	Bits 0..7 Bits 8..15 Bits 16..23 Bits 24..31	5. Zähler: Dämpfung der Geschwindigkeitsermittlung 6. Zähler: " " 7. Zähler: " " 8. Zähler: " " Hinweis: Beachten Sie hierzu das nächste Kapitel.	(r/w) (r/w) (r/w) (r/w)
219	Bits 0..7 Bits 8..15 Bits 16..23 Bits 24..31	5. Zähler: Exponent der Geschwindigkeitsanzeige 6. Zähler: " " 7. Zähler: " " 8. Zähler: " " Hinweis: Beachten Sie hierzu das nächste Kapitel.	(r/w) (r/w) (r/w) (r/w)
220	4	6. Zähler: aktueller Zählerstand (kann durch Schreiben gesetzt werden)	(r/w)
221	4	6. Zähler: aktuelle Geschwindigkeit	(r)
222	-		
223	-		
224	4	7. Zähler: aktueller Zählerstand (kann durch Schreiben gesetzt werden)	(r/w)
225	4	7. Zähler: aktuelle Geschwindigkeit	(r)
\$226	String	IO-Bus bzw. DBUS Adressenzuweisung. In dieser Variable sind die Seriennummer der einzelnen Buskomponenten und die Funktionsadressen des IO- bzw. D-BUS Systems hinterlegt.	(r/w)
227	4	siehe hierzu Kapitel 4.10. "Easy Access" Funktion auf Seite 40.	
228	4	8. Zähler: aktueller Zählerstand (kann durch Schreiben gesetzt werden)	(r/w)
229	4	8. Zähler: aktuelle Geschwindigkeit	(r)

Sehen Sie hierzu auch die DigiWEB "advanced" Konfiguration "<http://DigiWEB-IP/config/>" unter dem Menüpunkt: "**Counter/Encoder**".

4.7.2.1. Einstellung der Geschwindigkeitsanzeige

Um einen möglichst ruhigen und exakten Geschwindigkeitswert zu erhalten, wird dieser durch einen Tiefpass - Filter ermittelt. Der Filter benötigt zwei Werte: 1. Die Dämpfung und 2. Den Exponent.

Zunächst wird die Dämpfung festgelegt. Die benötigte Dämpfung ist abhängig von dem maximal zulässigen Anzeigefehler (in %) und der maximalen Geschwindigkeit (in Impulse pro Sekunde).

$$\text{Dämpfung} = \text{Log}_2 \left(\frac{10000}{\text{Anzeigefehler (in \%)} * \text{Max.Impuls (pro Sek.)}} \right)$$

Achtung: Die Dämpfung muß immer aufgerundet werden und darf nicht kleiner 1 und größer 15 sein.

Es ergibt sich eine Gesamtmeß - bzw. Aktualisierungszeit von:

$$\text{Gesamtmeßzeit} = 0.01\text{Sek.} * 2^{(\text{Dämpfung})}$$

Hinweis: Dauert die Ermittlung der Geschwindigkeit zu lang, so kann diese auf Kosten der Genauigkeit reduziert werden, in dem die Dämpfung kleiner gewählt wird.

Die Genauigkeit zum Meßbereichsendwert errechnet sich nach folgender Formel:

$$\text{Anzeigefehler (in \%)} = \frac{10000}{\text{Max.Impulse(pro Sek.)} * 2^{(\text{Dämpfung} * 2)}}$$

Der Exponent ist nun so einzustellen, dass der durch das DigiWEB ermittelte Geschwindigkeitswert bei maximaler Geschwindigkeit nicht überläuft (<31 Bit). Der Maximalwert rechnet sich nach folgender Formel:

$$\text{Max. Geschwindigkeitswert} = \text{Max.Impuls(pro Sek.)} * 0.01 * 2^{\text{Exponent} * 2^{(\text{Dämpfung} * 2)}}$$

Der Exponent ist jedoch so groß wie möglich einzustellen um eine optimale Rechengenauigkeit zu erreichen. Den größt möglichen Exponenten errechnen Sie nach folgender Formel:

$$\text{Exponent} = \text{Log}_2 \left(\frac{2^{31} * 100}{2^{(\text{Dämpfung} * 2)} * \text{Max.Impuls(pro Sek.)}} \right)$$

Achtung: Der Exponent muß immer abgerundet werden und darf nicht kleiner 0 und größer 31 sein.

Ist der Exponent eingegeben und wird die Geschwindigkeit im Register ##000097 errechnet, muß nun, um den gewünschten Anzeigewert zu erhalten, die Dämpfung und der Exponent herausgerechnet und auf die Anzeige skaliert (z.B. U/Min.) werden. Dies geschieht mit folgender Formel:

$$\text{Geschwindigkeit} = \text{##000097} * 100 / 2^{(\text{Exponent} + \text{Dämpfung} * 2)} * \text{skal.Faktor}$$

Beispiel:

Ein SSI - Encoder mit 360 Impulsen pro Umdrehung läuft mit maximal 550 U/Min. = (3300Imp./Sek.). Der Anzeigefehler soll 1% nicht übersteigen.

$$\text{Dämpfung} = \text{Log}_2 (10000 / (1 * 3300)) = \text{Log}_2 (3.03) = \sim 2$$

$$\text{Exponent} = \text{Log}_2 (214748364800 / (16 * 3300)) = \text{Log}_2 (4067203) = \sim 21$$

Um nun den Wert in U/Min. zu erhalten, muß der Multiplikator zunächst wieder herausgerechnet werden, die Zeiteinheit auf Minuten sowie die Impulse auf Umdrehungen umgerechnet werden.

$$\text{Speed (in Imp. pro Sek.)} = \text{##000097} * 100 / 2^{21+2*2} = \text{##MULDIV}(\text{##000097},100,33554432)$$

$$\text{Speed (in U/Min)} = \text{##MULDIV}(\text{##000097},100,33554432) / \text{Auflösung} * 60 \text{ Sek.}$$

oder

Faktor "/ Auflösung * 60 Sek." wird direkt in die Umrechnung mit eingebaut. Jedoch muß der Faktor gekürzt werden, damit kein Parameter der ##MULDIV Funktion größer 2³¹ wird:

$$\text{Speed (in U/Min)} = \text{##MULDIV}(\text{##000097},100,201326592)$$

Hinweis: Der Logarithmus von 2 errechnet sich: $\text{Log}_2 = \frac{\text{Log}(x)}{\text{Log}(2)}$

Bereich 0 / Register	Bytes / Bits	Konfiguration und Status der Seriellen - und der MPI - Schnittstelle lesen(r) / schreiben(w) möglich	
183	1	RK512 Wortgröße 1 = 1 Byte für S7 MPI / 2 = 2 Byte für S5 (Standard) 4 = 4 Byte (Digitronic CamCon Extended)	(1, 2, 4) (r/w)
184	Bit 0..31	An den RS485 BUS angeschlossene Geräte Bit Nr.: 0..31 gesetzt = Gerätenummer 0..31	(Bit codiert) (r)
185	Bit 0..31	An den RS485 BUS angeschlossene Geräte Bit Nr.: 0..31 gesetzt = Gerätenummer 32..63	(Bit codiert) (r)
186	Bit 0 Bit 8	Status der Seriellen Schnittstelle Status der MPI Schnittstelle	(0 = Fehler / 1 = OK) (r) (0 = Fehler / 1 = OK) (r)

Sehen Sie hierzu auch die DigiWEB "**advanced**" Konfiguration "<http://DigiWEB-IP/config/>" unter dem Menüpunkt: "**Serial/MPI/PB**".

4.8.1. Status der ColorNet Schnittstelle

Hier wird ab DigiWEB 3 (bei Model: D) der Status der ColorNet Schnittstelle bzw. die belegten ColorNet Adressen angezeigt.

Bereich 0 / Register	Bytes / Bits	Statusanzeigen ColorNet lesen(r) / schreiben(w) möglich	
140	Bits 0..31	Adresse 0..31 (Bit vorhanden = Adresse vorhanden)	(r)
141	Bits 0..31	Adresse 32..63 (Bit vorhanden = Adresse vorhanden)	(r)
142	Bits 0..31	Adresse 64..95 (Bit vorhanden = Adresse vorhanden)	(r)
143	Bits 0..31	Adresse 96..127 (Bit vorhanden = Adresse vorhanden)	(r)
144	Bits 0..31	Adresse 128..159 (Bit vorhanden = Adresse vorhanden)	(r)
145	Bits 0..31	Adresse 160..191 (Bit vorhanden = Adresse vorhanden)	(r)
146	Bits 0..31	Adresse 192..223 (Bit vorhanden = Adresse vorhanden)	(r)
147	Bits 0..23 Bits 24..31	Adresse 224..246 (Bit vorhanden = Adresse vorhanden) ColorNet Addresszähler (1..239)	(r) (r)

4.9. Auto Update Funktion

Mit diesen Registern wird der automatische Firmware Update des DigiWEB gesteuert. Diese Funktion ist notwendig bzw. sinnvoll, wenn Sie eine größere Anzahl von DigiWEB Geräten betreiben (z.B. DigiENERGY und bluebox) und den Software Update automatisch durchführen lassen wollen bzw. müssen.

Bereich 0 / Register	Bytes / Bits	DigiWEB Auto Update Funktion Konfiguration und Status lesen(r) / schreiben(w) möglich	
110	Bits 0..1 Bit 2 Bit 3..5 Bit 6 Bit 7 Bits 8..15 (signed)	Mode 0 = off / 1 = Automatisches Update / 2 = Check nach Update 1 = Update available Zeit Intervall der Prüfung auf Updates 1 = 1h / 2 = 4h / 3 = 8h / 4 = 12h / 5 = 1 Tag / 6 = 1 Woche / 7 = 2 Wochen Startet den Updatevorgang wenn eine 1 geschrieben wird Check auf Updates wenn eine 1 geschrieben wird Aktueller Status -1 = "Undefined error" -2 = "Undefined expression in control file" -5 = "Error on execute update" -7 = "Error on ns-lookup" -8 = "Error on getting http header" -9 = "Error on getting file content" -10 = "Filesystem error" -11 = "Undefined http header" -12 = "Error on send request" -13 = "Error no contentlength in http header" -14 = "Error on allocate memory for expression decoder" -15 = "URL changed while updating" -16 = "File not found" -17 = "Update off" -18 = "No URL" 1 = "ns-lookup" 2 = "Downloading controlfile" 3 = "Downloading update data" 4 = "Waiting for install time" 5 = "Installing update" 6 = "Update successful" 7 = "No update available" 8 = "Update available" 9 = "Wait for reset"	(r/w) (r) (r/w) (w) (w) (r)
	Bits 16..30 Bit 31	Zeit bis zum nächsten Check auf Updates in Sekunden oder Minuten Zeitbasis für Bits 16..30 in: 1 = Minuten 0 = Sekunden	(r) (r)
\$111	200	URL des Update Servers (mit http und File - Extension) z.B: http://update.digienergy.info/digiwebOnlineUpdater/updList.php	(r/w)

Sehen Sie hierzu die DigiWEB "advanced" Konfiguration "<http://DigiWEB-IP/config>" unter dem Menüpunkt: "Update client".

4.10. "Easy Access" Funktion

Durch die ab der Firmware V3.119 eingeführt "Easy Access" Funktion ist es möglich, ein DigiWEB, ein DigiENERGY, eine bluebox oder ein Lan - Gateway einfach über das Internet zu erreichen, ohne das im DSL Router (z.B. Fritzbox) eine Port - Weiterleitung konfiguriert werden muß.

Zusätzlich ist es auch möglich, über eine Standard mobile Datenanbindung (GPRS, EDGE oder UMTS - Router) den WEB - Server vom Internet aus zu erreichen, ohne einen komplizierte VPN - Verbindung herstellen zu müssen.

Hierzu muß in das Register #000119 nur die Adresse, die Portnummer und die Reconnect - Zeit des "Easy Access" Servers, mit Doppelpunkt getrennt, eingetragen werden.

Bereich 0 / Register	Bytes / Bits	DigiWEB Easy Access Funktion Konfigurieren lesen(r) / schreiben(w) möglich	
41	4	Interne EMS Verbindungsnummer	(r/w)
\$119	200	Domain Name des "Easy Access" - Servers (Reverse Proxy) Beispiel: "digitronic-portal.de:443:30000" "portal.digienergy.info:443:30000"	(r/w)
227	4	Easy Access Watchdog - Ausgangsnummer. Dieser Ausgang wird, wenn keine Verbindung zum Easy-Access-Server besteht, all 60 Minuten für 20 Sekunden eingeschaltet. Ein angeschlossener GSM Router kann hierdurch neu gestartet werden. Hinweis: Wird nicht remanend gespeichert.	(r/w)
186	Bit 17	Status der "Easy access" - Server Verbindung (ab Firmware 8/2016) (0 = nicht verbunden / 1 = verbunden) Hinweis: auch über ##DEBUG(41) möglich	(r)

Der WEB - Server ist bei erfolgreicher Kontaktaufnahme unter der "Easy Access" - Server Adresse und der eigenen MAC - Adresse bzw. Seriennummer des DigiWEB zu erreichen.

Beispiel: "http://portal.digienergy.info/0050C21E226B/"

Sehen Sie hierzu die DigiWEB "advanced" Konfiguration "<http://DigiWEB-IP/config>" unter dem Menüpunkt: "Update client".

4.11. Die zusätzlichen Register des DigiWEB 3

Hier folgt nun die Beschreibung der zusätzlich Register, die für das DigiWEB 3 BUS - bzw. Modulsystem notwendig sind.

4.11.1. DigiWEB 3 BUS Ausbaustufe

Das DigiWEB 3 besteht aus einer CPU und bis zu maximal 8 Modulen. Jedes Modul hat einen eigenen Buchstaben bzw. eine Bestelloption, die auf dem Typenschild (linke Gehäuseseite) des DigiWEB 3 aufgedruckt ist.

Somit ergibt sich eine Bestellnummer, die mit dem Gerätetyp ("DW3" = "DigiWEB 3" oder "DE" = "DigiENERGY") bzw. Grund - oder CPU - Modul beginnt und zusätzlich mit bis zu 8 Buchstaben für die verschiedenen Module codiert ist (z.B. DW3 XAP oder DE XXTAP). Die Ausbaustufe kann in den beiden Registern 38 und 39 gelesen werden.

Bereich 0 / Register	Bytes / Bits	Ausbaustufe des DigiWEB 3 BUS - bzw. Modulsystem lesen lesen(r) / schreiben(w) möglich		
38	Bits 0..7	Typkennung des Moduls mit Codeschalter Stellung/Nummer 0.	(r)	
	Bits 8..15	Typkennung des Moduls mit Codeschalter Stellung/Nummer 1.	(r)	
	Bits 16..23	Typkennung des Moduls mit Codeschalter Stellung/Nummer 2.	(r)	
	Bits 24..31	Typkennung des Moduls mit Codeschalter Stellung/Nummer 3.	(r)	
	Mögliche Module bzw. Typenkennungen:			
		0x01 = "Serial Interface"	Best.Code: 2+4	
		0x09 = "Modem"	Best.Code: M	
		0x80 = "MPI Interface"	Best.Code: S	
		0x10 = "Input / Output Modul"	Best.Code: X	
		0x40 = "Analog/Digital converter"	Best.Code: A+V	
		0x41 = "PT1000 Sensor"	Best.Code: T	
		0x42 = "Analog I Input Modul"	Best.Code: U	
39	Bits 0..7	Typkennung des Moduls mit Codeschalter Stellung/Nummer 4.	(r)	
	Bits 8..15	Typkennung des Moduls mit Codeschalter Stellung/Nummer 5.	(r)	
	Bits 16..23	Typkennung des Moduls mit Codeschalter Stellung/Nummer 6.	(r)	
	Bits 24..31	Typkennung des Moduls mit Codeschalter Stellung/Nummer 7.	(r)	
		0x20 = "AC 220V Modul"	Best.Code: P	
		0x51 = "Digital/Analog converter 0-20mA"	Best.Code: B	
		0x52 = "Digital/Analog converter 0-10V"	Best.Code: C	
	0x02 = "ColorNet"	Best.Code: D		

Sehen Sie hierzu die DigiWEB "advanced" Konfiguration "<http://DigiWEB-IP/config/>" unter dem Menüpunkt: "**DigiWEB3-Bus**".

Hinweis: Der Anwender kann das DigiWEB 3 BUS - System nicht selbst erweitern.

4.11.2. Die Register des DigiWEB 3 Modul: X - Die digitale Ein - Ausgangskarte

Sehen Sie hierzu Kapitel "4.7. Die DigiWEB Ein - Ausgangs Schnittstelle" in diesem Handbuch und in der DigiWEB "advanced" Konfiguration "<http://DigiWEB-IP/config/>" den Menüpunkt: "**Input/Output**".

4.11.3. Die Register des DigiWEB 3 Modul: A + V - Die analogen Eingangskarten

Bereich 0 / Register	Bytes / Bits	Register der Analogeingangskarte Modul: A oder Modul: V lesen(r) / schreiben(w) möglich	
58	Bits 0..27	1. A oder V - Modul im DigiWEB 3 BUS / 1. Eingang Aktueller Messwert mit Vorzeichen (lesen mit z.B. ##000058s0_27)	(r)
	Bit 28	Offset (0 = abgeglichen / 1 = nicht abgeglichen)	(r)
	Bit 29	Skalierung (+) (0 = abgeglichen / 1 = nicht abgeglichen)	(r)
	Bit 30	Skalierung (-) (0 = abgeglichen / 1 = nicht abgeglichen)	(r)
	Bit 31	Eingangstyp abgeglichen auf: (0 = Spannung für Modul: V) (1 = Strom für Modul: A)	(r)
	Bits 0..31	Zum Schreiben bzw. Abgleichen sehen Sie das nächste Kapitel.	(w)
59 .. 61	wie Reg. 58	1. A / V - Modul im DigiWEB 3 BUS / Eingang 2 bis 4 (wie Reg. 58)	(r/w)
62 .. 65	wie Reg. 58	2. A / V - Modul im DigiWEB 3 BUS / Eingang 1 bis 4 (wie Reg. 58)	(r/w)
66 .. 69	wie Reg. 58	3. A / V - Modul im DigiWEB 3 BUS / Eingang 1 bis 4 (wie Reg. 58)	(r/w)
70 .. 73	wie Reg. 58	4. A / V - Modul im DigiWEB 3 BUS / Eingang 1 bis 4 (wie Reg. 58)	(r/w)
74 .. 77	wie Reg. 58	5. A / V - Modul im DigiWEB 3 BUS / Eingang 1 bis 4 (wie Reg. 58)	(r/w)
78 .. 81	wie Reg. 58	6. A / V - Modul im DigiWEB 3 BUS / Eingang 1 bis 4 (wie Reg. 58)	(r/w)
82 .. 85	wie Reg. 58	7. A / V - Modul im DigiWEB 3 BUS / Eingang 1 bis 4 (wie Reg. 58)	(r/w)
86 .. 89	wie Reg. 58	8. A / V - Modul im DigiWEB 3 BUS / Eingang 1 bis 4 (wie Reg. 58)	(r/w)

4.11.3.1. Modul: A + V - Abgleichen

Abgleichen können Sie die analogen Eingangskarte direkt in der DigiWEB 3 "advanced" Konfiguration "<http://DigiWEB-IP/config>" unter dem Menüpunkt: "A/D Conv. (U/I)" oder durch Ihre eigene Applikation durch das Schreiben in das Register des jeweiligen Eingangs:

1. Zum Starten des Abgleichvorgangs muß zunächst eine 10 (1010) oder 11 (1011) in die Registerbits 28..31 geschrieben werden. Hierdurch wird der Eingangstyp eingestellt 10 = Spannung (Modul: A) / 11 = Strom (Modul: V) und die alten Abgleichwerte zurückgesetzt.

Hinweis: Die anschließend im aktuellen Meßwert gelieferten Werte sind nun die reellen Bits des A/D - Wandlers ohne die Skalierung. Wollen Sie die Werte des A/D - Wandlers direkt verarbeiten, müssen Sie jetzt den Abgleich abrechnen.

2. Schließen Sie nun den Nullpunkt - Meßwert an (z.B. 4mA = 0° oder 0V = 0°) oder schließen Sie den Eingang einfach kurz. Zum Schreiben des Offsetwerts wird nun in die Registerbits 28..31 eine 1 und gleichzeitig "ODER - Verknüpft" in die Bits 0..27 der gewünschte Offsetwert (Nullwert) geschrieben, hier (z.B. 0°).
3. Der positive Wert wird nun angeschlossen (z.B. bei 25° = ca. 6.77mA oder ca. 2,5Volt). Die angelegte Spannung oder der Strom ist dann der Skalierungswert, der geschrieben werden muß. Wollen Sie die Anzeige mit Nachkommastellen haben, so geben Sie den Wert einfach mit einem 10, 100 oder 1000 Faktor ein (z.B. 25000 = 25.000°). Hierzu wird in die Registerbits 28..31 eine 2 und gleichzeitig "ODER - Verknüpft" in die Bits 0..27 der gewünschte positive Wert (z.B. 25000) geschrieben.
4. Negative Skalierung schreiben. Hierzu wird auch wieder der zu messende negative Wert angelegt und das gewünschte Ergebnis eingegeben bzw. geschrieben. Zum Schreiben wird in die Registerbits 28..31 eine 4 und gleichzeitig "ODER - Verknüpft" in die Bits 0..27 der gewünschte minimal Wert mit Vorzeichen (z.B. ##000058s0_27:=-10000) geschrieben.

Hinweis: Die Abgleichwerte werden nullspannungsfest in einem EE - Prom auf der Eingangskarte gespeichert.

Tipp: Um eine möglichst hohe Genauigkeit zu erreichen, sollte der Strom - oder Spannungswert beim Abgleich im oberen Viertel des Meßbereichs liegen (>15mA) oder (>7,5Volt).

4.11.4. Die Register des DigiWEB 3 Modul: T - Die PT1000 Temperaturfühler Eingangskarte

Bereich 0 / Register	Bytes / Bits	Register der Temperaturfühler Eingangskarte Modul: T lesen(r) / schreiben(w) möglich	
8	Bits 0..31	1. T - Modul im DigiWEB 3 BUS / 1. Eingang Aktueller Meßwert in tausendstel Grad (0.001°C) mit Vorzeichen. Hinweis: Entspricht der gelesene Wert 0x80000000 oder 0x80000001, so ist die PT1000 Eingangskarte nicht vorhanden oder defekt. Ein Wert kleiner -100000 zeigt einen Kurzschluß der Meßleitung an. Ein Wert größer +179000 zeigt einen Kabelbruch an.	(r)
	Bits 0..31	Zum Schreiben bzw. Abgleichen sehen Sie das nächste Kapitel	(w)
9..16	wie Reg. 8	1. T - Modul im DigiWEB 3 BUS / Eingang 2 bis 9 (wie Reg. 8)	(r/w)
17..25	wie Reg. 8	2. T - Modul im DigiWEB 3 BUS / Eingang 1 bis 9 (wie Reg. 8)	(r/w)
26..34	wie Reg. 8	3. T - Modul im DigiWEB 3 BUS / Eingang 1 bis 9 (wie Reg. 8)	(r/w)
131..139	wie Reg. 8	4. T - Modul im DigiWEB 3 BUS / Eingang 1 bis 9 (wie Reg. 8) Hinweis: 4. T - Modul erst ab Firmware V3.100 möglich.	(r/w)

4.11.4.1. Modul: T abgleichen bzw. kalibrieren

Abgleichen können Sie die PT1000 Temperaturfühler Eingangskarte direkt in der DigiWEB 3 "advanced" Konfiguration "<http://DigiWEB-IP/config/>" unter dem Menüpunkt: "Temp. PT1000" oder durch Ihre eigene Applikation durch das Schreiben in das Register des jeweiligen Eingangs:

1. Schließen Sie hierzu den gewünschten PT1000 Meßfühler mit dem zum Schluß verlegten Kabel und der entsprechenden Kabellänge an den jeweiligen Eingang an.
2. Messen Sie mit einem präzisen Temperaturmeßgerät die aktuelle Temperatur am Meßfühler und schreiben Sie den gemessenen Wert in tausendstel Grad (0.001°C) in das Register für den jeweiligen Eingang.

Hinweis: Die Abgleichwerte werden nullspannungsfest in der DigiWEB Konfigurationsdatei "config.csv" gespeichert. Wird die PT1000 Temperaturfühler Eingangskarte gewechselt oder die Datei "config.csv" gelöscht bzw. überschrieben, so ist ein erneuter Abgleich notwendig.

TIP: Alternativ können Sie zum Abgleich auch einen Präzisionswiderstand mit 1000 Ohm und 0.1% Toleranz verwenden. Schließen Sie diesen auch mit dem später verlegten Kabel und der entsprechenden Kabellänge an den Eingang an und schreiben Sie dann den Wert 0 in das entsprechende Register.

4.12. Die zusätzliche Funktionen und Register des DigiWEB 4

Hier folgt nun die Beschreibung der zusätzlichen Register und Funktionen für den DigiWEB 4 BUS - bzw. das DigiWEB 4 Modulsystem.

4.12.1. Funktionen

Funktion	Beschreibung
##IOANALOGGET(Slot, Pin) Rückgabe: : Long	Mit dieser Funktion werden die Analogwerte der A/D - Eingänge gelesen.
##IOCONFIG(Slot, Pin, Command, Value) Rückgabe: : Long Beispiel 1: ##IOCONFIG(0,0,0x807E,0) Rückgabe: 3 = Software V3	Mit dieser Funktion werden die Multifunktions - Ein - und Ausgänge der DigiWEB 4 Module konfiguriert oder der Modulstatus gelesen.

4.12.2. Die Register des DigiWEB 4

Bereich 0 / Register	Bytes / Bits	Register lesen(r) / schreiben(w) möglich	
128	Bit 0	DigiWEB 4 Ethernetslave (0 = inaktiv / 1 = aktiv)	r/w

5. Beispiele für "REQ" - Scriptdateien

Beispiel 1:

Datei: Logger.req

Wird diese Datei zyklisch (z.B. 60 Minuten) aufgerufen so werden in der CSV - Datenbank "logger.csv" zwei Werte z.B. ##000187 in Value1 und ##201000 in Value2 + dem aktuellen Datum und der Zeit abgelegt. Die Datenbank kann dann später per FTP herunter geladen und weiterverarbeitet werden.

```
Demo für Datenbank Logger im DigiWEB
<!--##SqlConnection('','')-->           Datenbank öffnen
<!--#if(##000173b0)-->                   Zeit eintragen wenn synchron.
  <!--##$SqlField('Date') := ##$000172 % 19 --> Datum und Zeit eintragen
<!--#else-->
  <!--##$SqlField('Date') := 'kein NTP-Server' --> Kein Datum zum Eintragen vorhanden.
<!--#endif-->
<!--##$SqlField('Value1') := ##000187--> Wert eintragen z.B. DigiWEB Timer.
<!--##$SqlField('Value2') := ##201000--> Wert eintragen oder z.B. CamCon Istwert.
<!--##$SqlInsertMax('logger.csv','Dataset'):=1--> Datensatz anhängen
<!--##$SqlDisconnect()-->             Datenbank schliessen
```

Beispiel 1.1:

Wenn die Datenbankdatei "logger.csv" für das Beispiel zuvor noch nicht angelegt ist, so kann diese durch das folgenden Script angelegt werden. Dieses Script kann zur Sicherheit natürlich auch zyklisch vor dem Logger oder im Logger - Script selbst ausgeführt werden.

```
<!--#if(##EXISTS('logger.csv')==1)-->
<!--##COMMENTOUT('logger.csv') --> Eine neue leere Datenbank angelegt.
  <!--##$Dummy:='Dataset;Date;Value1;Value2;\r\n'-->##^
<!--##COMMENTOUT('')-->
<!--#endif-->
```

Beispiel 2:

Datei: TimSetup.req

Diese Datei wird durch die Event - Steuerung des DigiWEB aufgerufen und synchronisiert die Uhrzeit des DigiWEB mit der Echtzeituhr einer angeschlossenen S7.

Durch den Aufruf dieser Seite wird die DigiWEB Zeit mit der S7 Zeit synchronisiert wenn im DB5 der S7 SPS die aktuelle Uhrzeit hinterlegt ist.

Hinweis:

Um diese Seite aufzurufen fügen Sie in der SMTP Datenbank eine Anweisung ein, die ausgeführt wird, wenn das Bit 1 im Register 173 gesetzt wird. Dieses Bit wird jede Stunde einmal auf 1 gesetzt um die Zeit zu synchronisieren. z.B. "10;##000173:=1?0:##000187/10000;;timsetup.req;Uhrzeit all Stunde.synchronisieren" Solange das Synchronisieren nicht geklappt hat wird es alle 10 Sekunden wiederholt.

Hier wird die aktuelle Zeit aus der S7 gelesen.

```
Jahr: <!--##Jahr:= ##DB5.DBB8b0_3 + (##DB5.DBB8b4_7 * 10) + 2000-->####
Monat: <!--##Monat:= ##DB5.DBB9b0_3 + (##DB5.DBB9b4_7 * 10) -->##
Tag: <!--##Tag:= ##DB5.DBB10b0_3 + (##DB5.DBB10b4_7 * 10) -->##
Stunde: <!--##Stunde:= ##DB5.DBB11b0_3 + (##DB5.DBB11b4_7 * 10) -->##
Minute: <!--##Minute:= ##DB5.DBB12b0_3 + (##DB5.DBB12b4_7 * 10) -->##
Sekunde: <!--##Sekunde:= ##DB5.DBB13b0_3 + (##DB5.DBB13b4_7 * 10) -->##
```

Wenn das Lesen der Zeit erfolgreich war wird die Zeit eingestellt.

```
<!--#if(!##errno)-->
Sekunden seit 1900:<!--##000171:=(367*##Jahr+(##Tag-7*(##Jahr+(##Monat+9)/12)/4-3*((##Jahr+
  (##Monat-9)/7)/100+1)/4+275*##Monat/9)-693992)*86400 +
  (##Stunde*3600+##Minute*60+##Sekunde) -->#####
Offset:<!--##000174:=7200-->##### ( 7200 = Zeit Offset für Sommerzeit / 3600 = Zeit
  Offset für Winterzeit )
NTP Server IP:<!--##000170:=0--># = keine NTP SERVER IP
!!! Zeit wurde eingestellt.
<!--#else-->
!!! Zeit wurde NICHT eingestellt, da keine Infos von der S7.
<!--#endif-->
```

Beispiel 3:

Datei: getsr232.req

Dies ist ein Beispiel wie die Serielle Schnittstelle des DigiWEB für eine einfache Datenübertragung verwendet werden kann.

Für dieses Beispiel muß das "RS Interface/Protokoll" des DigiWEB auf "by DigiWEB Function" eingestellt werden und in der Event - Steuerung im Eingabefeld "Expression:" = "##COMLEN('0D0A')*##000187" und im Eingabefeld "Command:" = "/getsr232.req" eingetragen werden.

Wird im Schnittstellen - Puffer die Endekennung '0D0A' erkannt, so wird die Script - Datei solange aufgerufen, bis keine Endekennung '0D0A' mehr gefunden wird.

Das Script selbst holt dann die Zeichen bis zur Endekennung aus dem Schnittstellen Puffer und dekodiert diese und sendet daraufhin eine Antwort.

```
<!--##Len:##COMLEN('0D0A')-->
<!--if(##Len)--> Endekennung erkannt
  <!--$Text:##COMRECEIVE(0,##Len-2)--> nur den String holen
  <!--$Dummy:##COMRECEIVE(0,2)--> Endekennung aus Puffer entfernen
  <!--if($Text=='Frage1')-->
    <!--$Send:='Wert1'-->
  <!--elif($Text=='Frage2')-->
    <!--$Send:='Wert2:'+##000187-->
  <!--elif($Text=='Frage3')-->
    <!--$Send:='Wert3'-->
  <!--else-->
    <!--$Send:='Error:+'+$Text-->
  <!--endif-->
  <!--$Send:=$Send+'\r\n'-->
  <!--COMSEND(0,$Send)-->
<!--endif-->
```

Beispiel 4:

Datei: sendtcp.req

In diesem Beispiel wird über die MPI Schnittstelle aus eine S7 mit der MPI - Adresse 2 ein String mit max. 10 Byte Länge gelesen und via TCP/IP an den Port 8000 des PC mit der IP 192.168.2.41 gesendet. Die IP wird hierbei durch die Funktion "##dns('192.168.2.41')" in die notwendige 32 Bit Zahl umgerechnet, die der Funktion "##tcpopen" als Adresse übergeben werden muß.

Die Antwort des PC wird dann wieder zurück in die S7 - SPS geschrieben, wenn der Antwortstring Zeichen enthält (strlen > 0). Auf die Antwort des PC wird hierbei maximal 100ms gewartet (Timeout).

```
<!--$String:##DB1.DBB16B10G130-->* Wert aus S7 SPS
<!--##tcpopen(##dns('192.168.2.41'),8000,100)-->
<!--##tcpsend($String)-->
<!--$Antwort:##tcpreceive()-->
<!--##tcpclose()-->
<!--if(strlen($Antwort)-->
  <!--##DB1.DB16b[##DB1.DBB14G137]G130:=$Antwort-->* Wert von PC zurück in S7 SPS
  <!--##DB1.DBB15G130:##strlen($Antwort)-->
<!--endif-->
```

Hinweis: Die Funktion "##tcpreceive()" erwartet einen Zeichenumbuch "\r\n". Dieser muß in der Antwort des PC enthalten sein, da sonst der String nicht gelesen bzw. verarbeitet wird und der Antwortstring nach dem 100ms Timeout leer ist.

Beispiel 5:

Datei: wol.req

Hiermit wird ein Wake on LAN (WOL) UDP broadcast Paket=IP:255.255.255.255 Port:9 an die MAC Adresse einer PC Netzwerkkarte gesendet.

```
Hiermit wird ein Wake on LAN (WOL) UDP broadcast Paket=IP:255.255.255.255 Port:9
an die MAC Adresse des NIC gesendet.
<!--$MAC:='000B6A837BBD'-->
<!--##UDPSENDB(0xFFFFFFFF,'7A7A'+ '0009'+ 'FFFFFFFFFFFFFF'+$MAC+$MAC+$MAC+$MAC+$MAC-->
```

Beispiel 6:

Diese Expression liefert den Wochentag des aktuellen Datums.

```
<!--##DayofWeek:=(((##000171-0xa9491c00)/(24*60*60))%7)-->  
6 = Sonntag / 0 = Montag
```

5.1. Beispiele zur Dateiverwaltung

Das DigiWEB stellt verschiedene Funktionen zur Verwaltung von Dateien im Flashspeicher zur Verfügung. Diese sind:

##EXISTS('Name.csv') Diese Funktion liefert als Ergebnis eine -1, wenn die Datei nicht existiert. Existiert die Datei, dann liefert die Funktion die Länge der Datei als Ergebnis.

Beispiel:

```
<!--#if(##EXISTS('logger.cs')!=-1) -->  
<!--#if(##EXISTS('logger.csv') > 8192 )-->
```

##DELFILE('Name.csv') Löscht eine Datei im DigiWEB.
Als Ergebnis liefert sie:
0 = Fehlerfrei
-1 = Fehler bei der Ausführung.

##RENAME('Name.alt', 'Name.neu') Die Datei mit dem Namen 'Name.alt' wird umbenannt in die Datei 'Name.neu'. Existiert eine Datei mit dem neuen Namen schon, so wird diese überschrieben.
Als Ergebnis liefert sie:
0 = Fehlerfrei
-1 = Fehler bei der Ausführung.

##CPYFILE('Quelle.csv', 'Ziel.csv') Die Datei mit dem Namen 'Quelle.csv' wird kopiert in die Datei 'Ziel.csv'. Existiert eine Datei mit dem Zielnamen schon, so wird diese überschrieben.
Als Ergebnis liefert sie:
0 = Fehlerfrei
-1 = Fehler bei der Ausführung.

Achtung: Bei den Funktionen DELFILE, RENAME und CPYFILE darf das Funktionsergebnis **nicht** direkt in einer "#if" Anweisung verwendet werden. Es muß immer in einer lokalen Variablen gespeichert und anschließend verarbeitet bzw. geprüft werden.

Beispiel:

```
<!--##Ergeb:= ##[DEL][REN][CPY]FILE( 'Name.....' )-->  
<!--#if(##Ergeb=0)-->  
OK  
<!--#endif-->
```

##COMMENTOUT('logger.csv') Mit diesem Befehl wird eine neue Datei erstellt.
Hiernach werden alle DigiWEB spezifischen Ausgaben, die am Ende des Ausgabeformats "#" das Zeichen "^" enthalten, in die neu erzeugte Datei geschrieben. Hiermit lassen sich zum Beispiel Datenbanken oder auch Steuerdateien neu erzeugen.

##COMMENTOUT("") Mit dem gleichen Befehl ohne einen Dateinamen wird die geöffnete Datei gespeichert und geschlossen.

Beispiel:

```
<!--##COMMENTOUT('logger.csv') --> Eine neue leere Datenbank angelegt.  
<!--#$Dummy:='Dataset;Date;DB;Wert0;Wert1\r\n'-->##^  
<!--#$Dummy:='Dieser Text wird nicht in der Datei gespeichert.\r\n'-->##  
<!--##COMMENTOUT('')-->
```

Hinweis: Beachten Sie auch den freien Speicherplatz.
Der noch zur Verfügung stehende Flashspeicher kann in Kilobyte aus dem DigiWEB Register ##000198 gelesen werden. Sehen Sie Kapitel "4.3. Status Flash - Daten - Speicher".

5.2. Beispiel zur Datenbank

Wenn Sie das folgende Beispiel in einer "REQ" - Datei speichern und diese durch die Event - Steuerung z.B. alle 10 Minuten aufrufen lassen, so wird hier z.B. das aktuelle Datum mit Uhrzeit, der DigiWEB - Timer und der Istwert eines CamCon Nockenschaltwerks in der Datenbank "Logger.csv" gespeichert.

Datei: Logger.req

```
<!--##SqlConnection('', '', '')--> Datenbank öffnen
<!--if(##000173b0)--> Nur wenn die Uhrzeit synchron ist, wird die Zeit eingetragen.
  <!--##SqlField('Date') := ##000172 % 19 --> Datum eintragen.
<!--else-->
  <!--##SqlField('Date') := 'kein NTP-Server' -->Kein Datum zum Eintragen vorhanden.
<!--endif-->
<!--##SqlField('Value1') := ##000187--> DigiWEB Timer Register: 000187 eintragen.
<!--##SqlField('Value2') := ##201000--> CamCon Istwert Register: 201000 eintragen.
<!--##SqlInsertMax('logger.csv', 'Dataset'):=1--> Datensatz anhängen
<!--##SqlDisconnect()--> Datenbank schliessen
```

Datei: Logger.csv leer

Dataset;	Date;	Value1;	Value2
----------	-------	---------	--------

Datei: Logger.csv mit Daten

Dataset;	Date;	Value1;	Value2
1;	2004-02-24 16:40:39;	5191470;	253
2;	2004-02-24 16:40:42;	5194010;	62
3;	2004-02-24 16:40:45;	5196840;	251
usw....			

Um die notwendigen Daten in die Event - Steuerdatenbank "smtpfile.csv" einzufügen, kann natürlich auch ein SQL - Zugriff verwendet werden.

Beispiel Script - Code:

```
<!--##SqlConnection('', '', '')-->
<!--##SqlField('Expression') := '##000187/1000/600'-->
<!--##SqlField('Text') := 'logger.req'-->
<!--##SqlField('Subject') := 'Starte alle 10 Minuten den Logger' -->
<!--##SqlInsertMax('smtpfile.csv', 'Id'):=1-->"
```

Wird diese Code ausgeführt, so wird in die Datenbank "smtpfile.csv" die Logger - Anweisung eingefügt (Achtung: nur einmal!) und bei eingeschalteter Event - Steuerung alle 10 Minuten aufgerufen.

Hinweis: Je größer die Anzahl der Datensätze einer Datenbank wird, desto länger benötigt das DigiWEB zum Einfügen von neuen Datensätzen.

Achtung F. Jeder Schreibzugriff auf eine Datenbankdatei belastet den Flashspeicher. Der Flashspeicher des DigiWEB kann maximal 100000 mal gelöscht und geschrieben werden. Der Flashspeicher führt jedoch einen Selbsttest durch um zu Prüfen, ob der Speicher noch OK ist. Eine defekte Speicherzelle wird hierdurch erkannt und intern markiert. Dies führt zunächst nur zur Verringerung des Speicherplatzes im Flash. Sehen Sie hierzu das Register ##000198.

6. Fehlermeldungen bzw. Error Codes

"???"	Die Datenübertragung zum Kommunikationspartner ist nicht möglich = Timeout.
****	Der Wert ist zu groß und kann nicht dargestellt werden.
---	Es kann kein Wert angezeigt werden = keine Variable.
\$\$\$	Es kann kein Wert angezeigt werden = Puffer zu voll.
^^^	Die Datenübertragung zu einem anderen DigiWEB via Ethernet ist nicht möglich.
\$\$\$	Die Variable, die gelesen bzw. geschrieben werden soll, ist geschützt.
"~~~"	Die serielle Schnittstelle ist momentan belegt (z.B. durch die virtuelle COM - Schnittstelle oder durch das Digisoft Programm).
0	OK, kein Error.
1	OK2, kein Error.
2	OK3, kein Error.
-1	Unbekannter Error bzw. es ist kein Code zugeordnet.
-10	Klammernanzahl stimmt nicht, Klammer zu ")" fehlt.
-11	Variable fängt nicht mit # an.
-12	Adresse ##xxxxyy xxx oder yyy nicht korrekt.
-13	Parameter hatte keine Klammer zu ")" an der passende Stelle.
-14	Indexfehler [x], x darf keine Zeichenkette (String) sein.
-15	Indexfehler [x], Klammer zu "]" fehlt.
-16	Konstante Zahl oder "[" erwartet.
-17	'.' in einer IP-Adresse erwartet.
-18	Fehler in S7 Syntax, startet nicht mit "DB".
-19	Fehler in S7 Syntax, geht nicht mit ".DB" weiter.
-20	Fehler in S7 Syntax, Syntax nicht bekannt.
-21	#Input steht links von ":=".
-22	Kein Inputstring vorhanden.
-23	Klammer zu ")" in Funktionsaufruf erwartet.
-24	Variable beginnt nicht mit einem Buchstaben.
-25	Lokale Variable nicht definiert oder anlegbar.
-26	Funktion nicht definiert bzw. unbekannt.
-27	Klammer zu ")" im Ausdruck erwartet.
-28	Konstanter String zu groß.
-29	Vorzeichen kann nicht mit Stringvariablen arbeiten.
-30	Operand für Stringverarbeitung nicht zulässig.
-31	Operand für Wertverarbeitung nicht zulässig.
-32	Stringaddition würde zu lang werden.
-33	Doppelpunkt ":" nach Fragezeichen Operator "?" erwartet.
-34	Fehler in der Stringverarbeitung, möglicherweise nicht genügend Speicherplatz.
-35	Unerwartetes Zeichen, Semikolon erwartet ";".
-36	Fehler in der Expression bzw. Speicherüberlauf bei der Auswertung.
-37	Klammer zu "]" in der Expression erwartet.
-38	Fehler in der Expression, Zeichen ":-" nicht gefunden.
-39	String würde bei URL - Decodierung zu lang werden.
-40	String würde beim Formatieren zu lang werden.
-41	Längenangabe bei der Stringverarbeitung fehlt (#\$DB50.DBW0 b5 = 5Byte).
-42	Division durch NULL.
-43	Fehler beim Zugriff auf die DigiWEB SQL - Datenbank.
-44	z.Z. nicht definiert
-45	HTTP IP Transfer Get nicht durchgeführt
-46	HTTP IP Transfer Put nicht durchgeführt
-47	Symbol Array Fehler
-48	Symbolzugriff nicht korrekt, Arrayzugriff nicht korrekt.
-49	Expression kann nicht decodiert werden, Ausdruck scheint keine Expression zu sein.

- 50 Zeichen "=" im Script nicht gefunden.
- 51 Fehler bei der Ausführung, Speicherüberlauf bei der Parameterverarbeitung.
- 52 Timeout bei der Parameterübergabe zum Java Applet.
- 53 Speicherüberlauf bei der Parameterübergabe.
- 54 Parameter nicht decodierbar.
- 55 Diagramm Parameter X nicht gefunden.
- 56 Diagramm Parameter Y nicht gefunden.
- 57 Script Puffer voll.
- 58 Zugriff beim Datei schreiben verweigert.
- 59 Diagrammschleifenzähler zu groß.

- 60 Fehler beim E - Mail - Versand, nicht genügend Speicher.
- 61 Fehler beim E - Mail - Versand, z.B. SMTP Server nicht erreichbar.

- 62..71 z.Z. nicht definiert

- 72 Datei nicht gefunden bzw. Fehler.
- 73 Expression kann nicht decodiert werden.
- 74 Expression kann nicht decodiert werden, Ende fehlt.
- 75 Fehler bei der Ausführung, Speicherüberlauf bei MEM CMD.
- 76 Fehler bei der Ausführung, Speicherüberlauf bei STR CMD.
- 77 Fehler bei der Ausführung, Speicherüberlauf bei FILL CMD.
- 78 Anführungszeichen " fehlt.
- 79 Anführungszeichen im Attribut fehlt, zB. <input attr=">.

- 80 String Variable an dieser Stelle nicht zulässig.
- 81 Fehler bei der Ausführung, Speicherüberlauf.
- 82 Verschachtelung zu tief. IF kann nicht tiefer geschachtelt werden.
- 83 Schleifenfehler, UNTIL ohne REPEAT.
- 84 Klammer auf erwartet "(".
- 85 Klammer zu erwartet ")".
- 86 Fehler bei der Ausführung, Speicherüberlauf bei Fill CMD.
- 87 Fehler bei der Ausführung, Speicherüberlauf bei der Stringverarbeitung.
- 88 Fehler bei der Ausführung, Speicherüberlauf bei der Stringverarbeitung.
- 89 Fehler bei der Stringverarbeitung, keine Zahl.

- 90 Unerwartetes Dateiende (EOF).
- 91 Fehler im GIF Dateinamen.
- 92 Fehler bei der Ausführung, Speicherüberlauf bei Print CMD.
- 93 Fehler bei der Ausführung, Internal Loop Error.
- 94 Verschachtelung für Include Datei zu tief.
- 95 Include Datei nicht gefunden.

- 100 Fehler bei der Kommunikation, der Datentransfer - Speicher (Cache) ist voll.
- 101 Fehler bei der Kommunikation, Variable in der Gegenstelle (z.B. S7, CamCon) nicht vorhanden.
- 102 Fehler bei der seriellen Schnittstellen Kommunikation, keine Antwort durch die Gegenstelle (z.B. falsche MPI Adresse, das falsche Protokoll, falsche Gerätenummer eingestellt oder Kabel nicht angeschlossen).
- 103 Bei TCP/IP - Transfer kein Kontakt.
- 104 Variable ist lesegeschützt.
- 105 Variable ist schreibgeschützt.
- 106 Serielle Schnittstelle belegt.
- 107 Passwortsperrung durch "Brute Force". Es wurde 3 oder mehrmalig ein falscher "Username" oder ein falsches "Passwort" eingegeben. Die 1. Sperre ist 30 Sekunden lang, jeder weitere nicht erfolgreiche Einlogversuch erhöht diese Zeit auf bis zu 11 Stunden !

- 110 Fehler beim Schreiben einer Datei.
- 120 HTTP Port belegt.

Hinweis: Die Fehlermeldungen von -1 bis -95 führen zum Abbruch der Scriptverarbeitung bzw. der Seitenübertragung bei interpretierten *.htm, *.xml, *.req, *.txh oder *.js Seiten. Fehlermeldungen ab -100.. müssen bei diesen Seiten mit der Funktion "##errno()" abgefragt werden.

7. Script - bzw. REQ - Dateien in der DigiVISU

Auch in einem DigiVISU Projekt kann eine Script - Datei zum Lesen und Schreiben von Werten bzw. Variablen verwendet bzw. aufgerufen werden.

Hierzu muß im Animationsattribut **"TXT"** bzw. **"OnWrite"** der Dateiname (test.req) der Script - Datei eingegeben werden. Dies ist z.B. bei einem Textfeld, Eingabefeld oder einem Button (Knopf) möglich.

Stell DigiVISU fest, dass das Animationsattribut **"TXT"** angezeigt werden soll (update) oder der Wert verändert wurde (Benutzereingabe) und geschrieben **"OnWrite"** werden muß, ruft DigiVISU die Script - Datei mit den entsprechenden Parametern bzw. Wert auf.

Die durch das Script gelieferten Daten werden anschließend in der DigiVISU ausgewertet und im entsprechenden Eingabefeld oder z.B. auch Textfeld angezeigt.

Achtung: Die Aktualisierungsrate (update rate) für ein Animationsattribut mit Script - bzw. REQ - Datei darf niemals kleiner als **"All sec"** (1 Sekunde) eingestellt werden.

7.1. Rückgabe der Script - Ausgaben an DigiVISU

Zur Rückgabe bzw. Animation des DigiVISU Objekts durch das Script muß der Wert mit zwei führenden Doppelpunkten quasi markiert zurück gegeben werden.

Beispiel 1: `::<!--##000187-->#*`

Es können auch mehrere DigiVISU Objekte durch den Aufruf eines einzigen Scripts animiert werden. Hierzu muß jedoch der absolute DigiVISU Objektname und die Animations - Typennummer des Objekts bekannt sein.

Der Objektname, die Typennummer und der eigentliche Wert müssen mit je zwei Doppelpunkten begrenzt zurück gegeben werden.

Beispiel 1: Script Rückgabe der Textanimation (0) für Element "MENU15\Testfeld2":

```
::MENU15\Testfeld2:0:0:<!--##000187-->#*
```

Beispiel 2: Script Rückgabe für zwei Elemente gleichzeitig:

```
::MENU15\Testfeld2:0:0:<!--##000187-->#*  
::MENU15\Input1:0:0:<!--##000512-->#*
```

7.2. Parameter beim Schreiben durch ein Script

Beim Schreiben eines Wertes übergibt die DigiVISU dem Script die Variable `#$VALUE` den in einem Eingabefeld (INPUT) eingegebenen Wert als Parameter, sodass im Script dieser geschrieben werden kann. Zum Vermeiden einer Fehlermeldung muß jedoch hierbei geprüft werden, ob die Variable `#$VALUE` überhaupt definiert ist, da diese z.B. beim Lesen natürlich nicht übergeben wird.

```
::MENU15\Testfeld2:0:0:<!--##000187-->#*  
<!--#if(##defined(#$VALUE))-->  
::MENU15\Input1:0:0:<!--##000512:=0+#$VALUE-->#*  
<!--#endif-->  
::MENU15\Input1:0:0:<!--##000512:=##000512+1-->#*  
<!--#endif-->
```

7.3. Locale Variable der DigiVISU dem Script als Parameter übergeben

Hierzu muß hinter dem Scriptnamen, mit ? getrennt, die locale DigiVISU - Variable (`@#Var`) einer lokalen DigiWEB - Variablen (`##Var`) zugewiesen werden. Hierdurch wird diese dem Script als Parameter übergeben. Im Script kann dann die locale DigiWEB - Variable (`##Var`) verwendet werden.

Beispiel 1: (mit Parameterübergabe) **"script.req?##Var:=@#Var"**

```
<!--#if(##defined(##Var))-->  
  <!--##000512:=##Var-->  
<!--#endif-->
```

Beispiel 2: (mit zwei Parametern) **"script.req.req?##Var:=@#Var&&##Var1:=@#DWTimer"**

```
<!--#if(##defined(##Var) && ##defined(##Var1))-->  
  <!--##000512:=##Var-->  
  <!--##000513:=##Var1-->  
<!--#endif-->
```

Hinweis: Hierbei müssen die Zuweisungen durch drei '&' - Zeichen voneinander getrennt werden.

8. Die "TXH" - Scriptdatei

Die DigiWEB "TXH" - Scriptdatei wird verwendet, um z.B. E-Mails als reine Text E-Mails (ohne HTML Teil) zu versenden oder auch zur Steuerung eines Backup und Restore - Systems.

Wie in einer REQ - Scriptdatei können auch hier die meisten DigiWEB spezifischen Befehle verwendet werden. Es sind jedoch **keine** Funktionen möglich, die eine Eingabe (z.B. "<input ..>") erfordern und es ist keine graphische Darstellung (z.B. "<img...>") möglich.

Der Vorteil der TXH - Datei gegenüber einer HTML - Seite ist, dass der Code schmaler ist und schneller ausgeführt werden kann. Zusätzlich kann durch eine TXH - Datei eine CVS - Datenbankdatei automatisch erstellt oder deren Datensätze gelesen bzw. geschrieben werden, wenn eine CVS - Datei mit dem gleichen Namen wie die einer TXH - Datei via FTP übertragen wird. Sehen Sie hierzu das nächste Kapitel.

8.1. Backup und Restore durch die TXH - Scriptdatei

Für jede TXH - Datei, die im DigiWEB gespeichert ist, wird automatisch eine temporäre CSV - Datei (Datenbank) mit gleichem Namen erzeugt. Diese CSV - Datei wird mit den Ausgaben der gleichnamigen TXH - Datei gefüllt, so dass automatisch eine Datenbank bzw. ein Backup von frei definierten Daten erstellt werden kann. Wird die CSV - Datei mittels FTP - Transfer aus dem DigiWEB gelesen, so wird zunächst die Datei temporär neu erzeugt (Backup der aktuellen Daten) und anschließend übertragen.

Wird beim Erstellen der CSV - Datenbank, ein DigiWEB konformer Syntax eingehalten, so werden beim Schreiben einer CSV - Datei via FTP, für die eine gleichnamige TXH - Datei existiert, die Ausdrücke bzw. Expressions aus der Datenbank ausgeführt (Restore).

Beispiel:

Datei: Backup.txh

```
Register;Value;Info
<!--'##DB1.DBT0;' + ##DB1.DBT0 + ';Timer Sollwert 1'-->##*
<!--'##DB1.DBT2;' + ##DB1.DBT2 + ';Timer Sollwert 2'-->##*
<!--'##DB1.DB3F4;' + #FORMAT('#####',##DB1.DB3F4) + ';Real Wert'-->##*
<!--'##DB1.DBZ8;' + ##DB1.DBZ8 + ';Zähler Sollwert 1'-->##*
<!--'##DB1.DBZ10;' + ##DB1.DBZ10 + ';Zähler Sollwert 2'-->##*
<!--'##DB1.DBB16b10;' + #DB1.DBB16b10 + ';String 1'-->##*
<!--'##DB1.DBB15;##STRLEN(#DB1.DBB16b10);Berechne String Len für String 1'-->##*
<!--'##DB1.DBB28b10;' + #DB1.DBB28b10 + ';String 2'-->##*
<!--'##DB1.DBB27;##STRLEN(#DB1.DBB28b10);Berechne String Len für String 2'-->##*
<!--'##DB1.DBX38.0;' + ##DB1.DBX38.0 + ';Bit 1'-->##*
<!--'##DB1.DBX38.1;' + ##DB1.DBX38.1 + ';Bit 2'-->##*
<!--'##DB1.DBX38.2;' + ##DB1.DBX38.2 + ';Bit 3'-->##*
<!--'##DB1.DBX38.3;' + ##DB1.DBX38.3 + ';Bit 4'-->##*
<!--'##DB1.DBX38.4;' + ##DB1.DBX38.4 + ';Bit 5'-->##*
```

Diese Datei erzeugt durch ihre Ausgabe folgende temporäre CSV - Datei während der FTP Datenübertragung mit den aktuellen Daten aus DB1 einer S7.

Datei: Backup.csv

```
Register;Value;Info
##DB1.DBT0;1000;Timer Sollwert 1
##DB1.DBT2;1000;Timer Sollwert 2
##DB1.DB3F4;12,345;Real Wert
##DB1.DBZ8;50;Zähler Sollwert 1
##DB1.DBZ10;50;Zähler Sollwert 2
#$DB1.DBB16b10;Martin;String 1
##DB1.DBB15;##STRLEN(#DB1.DBB16b10);Calc String Len String 1
#$DB1.DBB28b10;Schultz;String 2
##DB1.DBB27;##STRLEN(#DB1.DBB28b10);Calc String Len String 2
##DB1.DBX38.0;1;Bit 1
##DB1.DBX38.1;1;Bit 2
##DB1.DBX38.2;1;Bit 3
##DB1.DBX38.3;1;Bit 4
##DB1.DBX38.4;1;Bit 5
```

Wird diese CSV - Datei via FTP zurück in das DigiWEB übertragen, so wird überprüft, ob im 1. Feld der Datensätze ein DigiWEB Ausdruck bzw. Expression vorhanden ist. Ist dies der Fall, so wird aus dem jeweils 2. Feld der zu schreibende Wert gelesen und geschoben (restore).

In dem Beispiel wird z.B. in den DB1.DBW0 ein Timersollwert von 1000ms geschrieben.

9. Die seriellen Protokolle für den Softwareentwickler

9.1. Das 3964(R) Protokoll

Das 3964(R) Protokoll wird zur Punkt zu Punkt Kommunikation zwischen SPS Steuerungen und PCs eingesetzt. Für die meisten SPS Steuerungen und IPC stehen diese als Kommunikationsprozessor oder Windows DLL Treiber zur Verfügung.

Müssen Sie das 3964(R) Protokoll in eine eigene Steuerung integrieren, so folgt hier nun eine kurze Beschreibung:

Allgemeiner Telegrammaufbau in HEX:

PC, SPS oder DigiWEB sendet	CamCon oder Fremdsystem antwortet
0x02	
	0x10
Daten (RK512 Telegrammanfrage) Achtung: Enthalten die Daten das Zeichen 0x10 so muß dies jeweils doppelt gesendet werden !	
0x10, 0x03, BCC	
	0x10

PC, SPS oder DigiWEB antwortet:	CamCon oder Fremdsystem sendet:
	0x02
0x10	
	Daten (RK512 Telegrammquittierung) Achtung: Enthalten die Daten das Zeichen 0x10, so wird dies jeweils doppelt gesendet !
	0x10, 0x03, BCC
0x10	

BCC: Die Prüfsumme BCC ist die Exklusiv - Oder - Summe der Daten und der Zeichen 0x10 und 0x03. Ein doppeltes 0x10 Zeichen in den Daten wird zur Prüfsumme hinzugezählt.

Fehler: Wird ein Fehler im Protokoll festgestellt, so antwortet die jeweilige Station mit dem Zeichen 0x15 für eine negative Quittierung.

Timeout: Das Timeout der Kommunikation beträgt 220ms.

9.2. Das RK512 Protokoll

Ein RK512 Telegramm wird immer durch das Senden zweier NULL Bytes begonnen. Wird das Telegramm verstanden, so werden die entsprechenden Daten durch ein Antworttelegramm zurückgesendet.

Hinweis: Für die CamCon Nockensteuerung steht z.Zeit nur der Befehl Datenbereiche bzw. Bausteine schreiben oder lesen mit einer Blocklänge von max. 128 Byte = 32 Worte je 32Bit (DINT) zur Verfügung.

Beispiel einer Anfrage:

Der PC, DigiWEB oder die SPS sendet:

1.Byte:	0h	Kennung des Telegramms		
2.Byte:	0h	" "		
3.Byte:		Kennung der Auftragsart	Wert:	"A" oder "E"
3.Byte	45h	"E" = Daten aus CamCon lesen		
oder				
3.Byte	41h	"A" = Daten in CamCon schreiben		
4.Byte:		Kennung des Datenbereichs	Wert:	"D" = Datenbaust.
5.Byte:		Ziel - Datenbereich (Baustein)	Werte:	1 - 255
6.Byte:		Anfangsadresse (Offset)	Werte:	0 - 255
7.Byte:		Anzahl der Datenworte	High Wert:	immer 0h
8.Byte:		" "	Low Wert:	2 - 64 (Anz. 16Bit-Worte)
9.Byte:		Koordinierungsmerker-Byte	Wert:	immer FFh für aus
10.Byte:		Koordinierungsmerker-Bit	Wert:	immer FFh für aus
n.Byte:		Daten	Anzahl:	maximal 128 Byte

Steuerung antwortet:

1.Byte:	0h	Kennung des Antworttelegramms		
2.Byte:	0h	" "		
3.Byte:	0h	" "		
4.Byte:		Fehlernummer	Werte:	0 = kein Fehler
			Wert:	16 = Befehl unbekannt (nur Datenbausteine "D")
			Wert:	20 = außerhalb des DB
			Wert:	22 = Befehl unbekannt (nur E oder A als Art)
			Wert:	52 = zu viele Daten
			Wert:	54 = allgem. Fehler
n.Byte:		Daten	Anzahl:	maximal 128 Byte

Achtung: Um einen breiteren Adressraum zu erhalten, wird bei den CamCon Nockenschaltwerken eine auf 32 Bit (4 Byte) erweiterte "extended" Datenwortstruktur verwendet. Wird ein Standard RK512 Treiber für S5 Steuerungen eingesetzt, so muß beim Lesen oder Schreiben berücksichtigt werden, daß sich die Daten Doppelworte (DD) nicht wie bei einer S5 üblich überlappen. Bei einer S5 belegt z.B. das DD1 einen Teil von DD0 oder das DD3 einen Teil von DD4. Für das CamCon jedoch ist jedes DD ein eigenes 32 Bit breites Register.

10. Stichwortverzeichnis

##CASTFLOAT	22
##CPYFILE	48
##defined()	12
##DELFILE	48
##ExchangeGet	23; 24
##ExchangeInsert	23
##EXISTS	48
##MODBUS	21
##muldiv()	9
##PLC	25
##RENFILE	48
#elif	13
#else	13
#endif	13
#if	13
#include	14
#repeat	14
#until	14
Ablaufsteuerung	13
Adressierung, DigiWEB <-> DigiWEB	7
Adressierung, indirekt	9
Adressierung, S7/S5 Format	8
Adressierung, Symbolische	6
App.json	27
Applikation, Symbole	27
Applikation, Variablen	27
Ausdrücke	5
Ausgabestand	2
Auto Update Funktion	39
Backup via FTP	53
Bedingungen	9
Cache, Bowser Bilder Cache	29
CE - Zeichen, EMV - Verträglichkeit	2
ColorNet, Status	39
Datei, grösse	48
Datei, kopieren	48
Datei, löschen	48
Datei, neu erzeugen	48
Datei, umbenennen	48
Dateiverwaltung	48
Datenaufzeichnung	49
Datenaustausch, DigiWEB <-> DigiWEB	7
Datenbank, SQL	17
DCF77	31
DigiVISU und Scriptdateien	52
DigiWEB 3, BUS/Modul Ausbau	41
DigiWEB 3, Modul	
A	42
T	43
X	41
DigiWEB 3, Register	41
DigiWEB 4, Funktionen	44
DigiWEB 4, Register	44

E-Mail	31
Easy Access Funktion.....	40
Ein - / Ausgänge.....	32
Ein - / Ausgänge, Fehlermeldung quittieren	32
Einleitung	5
endwhile, #endwhile.....	13
Event Händler, aktivieren	31
Flash	30
Flash, free	30
Flash, Schreibzyklen	30
Flashspeicher, defekt.....	30
Flashspeicher, frei.....	30
Funktionen, allgemein ab DW2.....	18
Funktionen, Debug.....	26
Funktionen, Exchange	23
Funktionen, IoBUS bzw. DBUS.....	20
Funktionen, MODBUS	21
Funktionen, PLC - Logik - Modul	25
Funktionen, Serielle Schnittstelle	20
Funktionen, TCP/IP manuelle Verbindung.....	19
Funktionen, Übersicht.....	15
Funktionen, UDP Paket senden.....	19
Gerätenummer, Adressierung	6
I/O Schnittstelle, Software.....	32
I/O Schnittstelle, SSI Wegmeßsystem.....	34
Impulszählung.....	34
Inhaltsverzeichnis.....	3
IP - Adresse	29
Konfiguration	27
Logging, Daten.....	49
Markenzeichen.....	2
Merker	31
Merker, remanent.....	31
Modbus - Registeranzahl	22
MPI - Adresse, CPU adressieren.....	6
MPI, Einstellung	37
Netzwerk	29
NTP, Einstellung	31
Operatoren, arithmetische	9
Operatoren, boolsche	9
Operatoren, logische.....	10
Parameter, übergeben	12
Passwort, System	29
PDF - Datei	2
Profibus, Einstellung	37
Protokoll, 3964(R)	54
Protokoll, RK512	55
Protokolle für den Softwareentwickler.....	54
Puls - Weiten - Steuerung.....	33
PWM	33

Rechnen.....	9
REQ - Scriptdatei, Beispiele.....	45
Restore via FTP	53
RK512 Tabelle 0	27
RoHS.....	2
Serielle Schnittstelle, Einstellung	37
SMTP, Einstellung.....	31
Sommer/Winterzeit	31
SPS - Logik - Modul, Zugriff.....	25
SQL	17
Strings	11
Strings, #strlen().....	12
Symbole	6
Syntax	5
System Register.....	29
TCPIP, Parameter.....	29
TXH - Scriptdatei.....	53
Uhrzeit.....	31
Unterprogramme.....	14
Update, Handbuch im Internet	2
Variablen, lokal.....	12
Variablenfunktionen	12
Variablentransport, DigiWEB to DigiWEB.....	23
while, #while	13
Wochentag.....	47