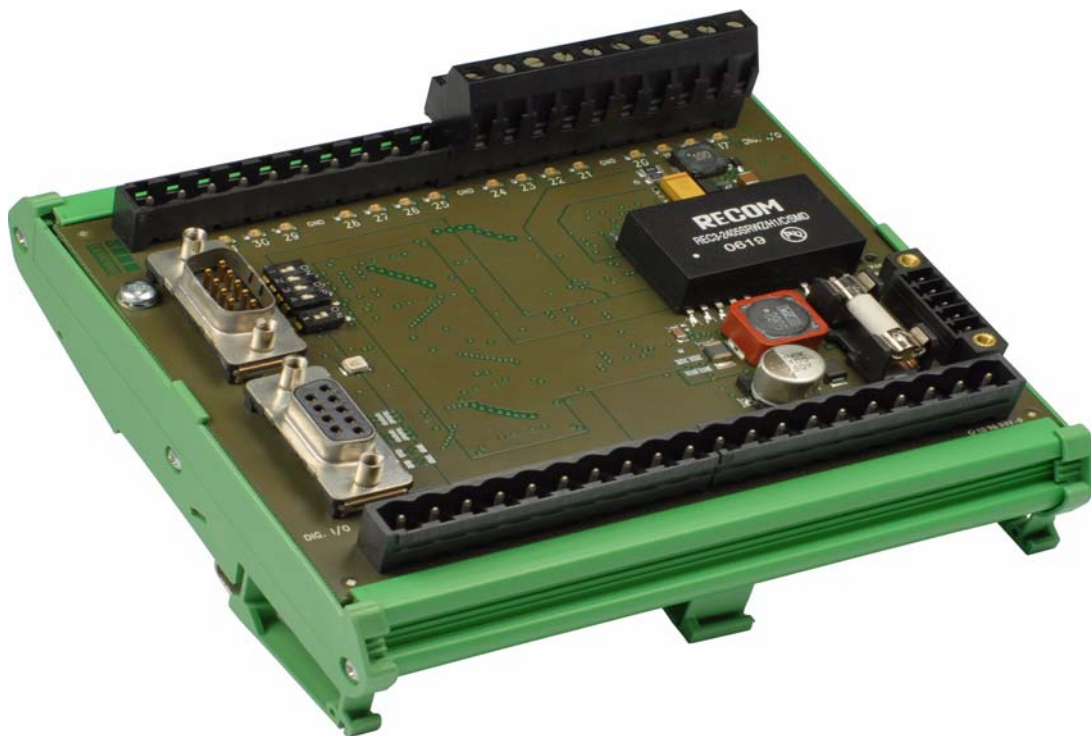


# ***ADwin HSM-24V***

**Modul für LS-Bus**

**Handbuch**



**Hier finden Sie immer einen Ansprechpartner für Ihre Fragen:**

Hotline: (0 62 51) 9 63 20  
Fax: (0 62 51) 5 68 19  
E-Mail: [info@ADwin.de](mailto:info@ADwin.de)  
Internet: [www.ADwin.de](http://www.ADwin.de)

 **JÄGER**  
Computergesteuerte  
Messtechnik GmbH  
Jäger Computergesteuerte  
Messtechnik GmbH  
Rheinstraße 2-4  
D-64653 Lorsch

## Inhaltsverzeichnis

Typografische Konventionen .....	IV
1 Zu diesem Handbuch .....	1
2 Der LS-Bus .....	2
3 HSM-24V .....	4
3.1 Hardware .....	4
3.2 Software .....	5
3.2.1 LS-Bus + <i>ADwin-light-16</i> .....	6
3.2.2 LS-Bus + Pro I. ....	14

## Typografische Konventionen



<C:\ADwin\ ...>

**Programmtext**

Var\_1

Das „Achtung“-Zeichen steht bei Informationen, die auf Folgeschäden durch Fehlbedienung an der Hard- oder Software, am Messaufbau oder an Personen hinweisen.

Einen „Hinweis“ finden Sie bei

- Informationen, die für einen fehlerfreien Betrieb unbedingt beachtet werden müssen.
- Tipps und Ratschlägen für einen effizienten Betrieb.

Das Zeichen „Information“ verweist auf weiterführende Informationen in dieser Dokumentation oder andere Quellen wie Handbücher, Datenblätter, Literatur etc.

Dateinamen und -verzeichnisse sind in spitzen Klammern und im Schrifttyp Courier New angegeben.

Programmanweisungen und Benutzer-Eingaben sind durch den Schrifttyp Courier New gekennzeichnet.

Elemente eines Quelltextes wie **BEFEHLE**, Variablen, Kommentar und sonstiger Text werden im Schrifttyp Courier New und farbig dargestellt (wie im Editor der Entwicklungsumgebung *ADbasic*).

In einem Datenwort (hier: 16 Bit) werden die Bits wie folgt nummeriert:

Bit-Nr.	15	14	13	...	1	0
Wert des Bits	$2^{15}$	$2^{14}$	$2^{13}$	...	$2^1=2$	$2^0=1$
Bezeichnung	MSB	-	-	-	-	LSB

## 1 Zu diesem Handbuch

Dieses Handbuch beschreibt das Modul HSM-24V, das am LS-Bus betrieben wird. Es wird ergänzt durch

- die Beschreibung der LS-Bus-Schnittstelle im Hardware-Handbuch für *ADwin-light-16*, *ADwin-Gold* oder *ADwin-Pro*.
- das Handbuch *ADbasic*, das grundlegende Befehle für den gleichnamigen Compiler enthält sowie das Funktionsprinzip von *ADwin*-Systemen näher erläutert.

Die Online-Hilfe hat den gleichen Inhalt wie das Handbuch *ADbasic* und enthält zusätzlich die Hardware-bezogenen Befehle.



### Bitte beachten Sie folgende Hinweise

Damit Ihr **ADwin**-System sicher arbeitet, halten Sie sich an die Informationen dieser und weiterführender Dokumentationen, auf die hier verwiesen wird.

Der Hersteller des in dieser Dokumentation beschriebenen Systems geht davon aus, dass an dem Gerät nur qualifiziertes Personal arbeitet.

*Qualifiziertes Personal sind Personen, die aufgrund ihrer Ausbildung, Erfahrung und Unterweisung sowie ihrer Kenntnisse über einschlägige Normen, Bestimmungen, Unfallverhütungsvorschriften und Betriebsverhältnisse von dem für die Sicherheit der Anlage Verantwortlichen berechtigt worden sind, die jeweils erforderlichen Tätigkeiten auszuführen und die dabei mögliche Gefahren erkennen und vermeiden können.  
(Definition für Fachkräfte nach VDE 105 und ICE 60364).*

Diese Produktdokumentation und Unterlagen, auf die verwiesen wird, müssen stets verfügbar sein und konsequent beachtet werden. Für Schäden, die durch Missachtung der Informationen in dieser bzw. der weiterführenden Dokumentation entstehen, übernimmt die Firma **Jäger Computergesteuerte Messtechnik GmbH**, Lorsch, keine Haftung.

Diese Dokumentation ist einschließlich aller Abbildungen urheberrechtlich geschützt. Reproduktion, Übersetzung sowie elektronische und fotografische Archivierung und Veränderung bedürfen der schriftlichen Genehmigung der Firma **Jäger Computergesteuerte Messtechnik GmbH**, Lorsch.

Fremdprodukte werden ohne Vermerk auf mögliche Patentrechte genannt, deren Existenz nicht auszuschließen ist.

Änderungen vorbehalten.

Hotline-Adresse siehe vordere Umschlagseite, innen.

### Einschränkung der Anwendergruppe

### Verfügbarkeit der Unterlagen

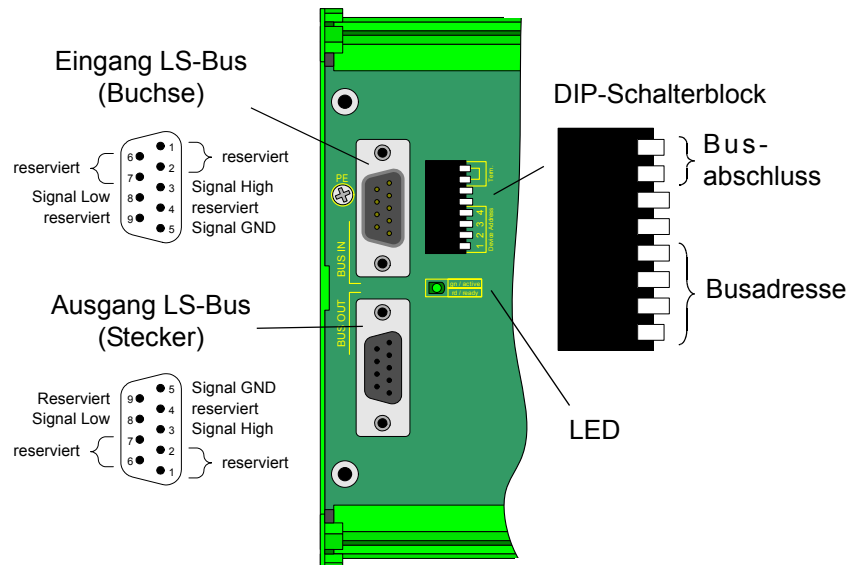


### Rechtliche Grundlagen

## 2 Der LS-Bus

Der LS-Bus ist ein serieller, bidirektionaler Bus mit einer Taktrate von 5MHz. Der Bus verbindet ein ADwin-System über dessen LS-Bus-Schnittstelle mit bis zu 15 LS-Bus-Modulen.

Abbildung 1 zeigt die Standard-Anschlüsse eines LS-Bus-Moduls: Bus-Ein-/Ausgang mit LED, Schutzleiter PE, DIP-Schalter für Busabschluss und Busadresse.



Der Bus ist als Linienverbindung aufgebaut, d.h. die Schnittstelle und die LS-Bus-Module sind jeweils über Zweipunktverbindungen miteinander verbunden. An jedem Modul sind eine Sub-D-Buchse (9-polig) als Bus-Eingang und ein Sub-D-Stecker (9-polig) als Bus-Ausgang vorhanden. Die maximale Buslänge beträgt 5m.

Die LED neben den Busein- und -ausgang zeigt den Datenverkehr auf dem LS-Bus an:

- LED leuchtet grün: Das Modul empfängt oder sendet Daten.
- LED leuchtet rot: Auf dem Bus werden Daten für andere Module gesendet.
- LED ist aus: Kein Datenverkehr.

### Busabschluss

Am letzten Modul des LS-Bus muss der Busabschluss mit den DIP-Schaltern *Term.*, aktiviert sein, an allen anderen Modulen deaktiviert. Zum Aktivieren des Busabschlusses werden die beiden DIP-Schalter nach unten gestellt.

### Busadresse

Jedes Modul am LS-Bus wird über seine Busadresse angesprochen; die Adresse muss daher für jedes Modul eindeutig sein.

Die Busadresse wird manuell an einem DIP-Schalterblock auf der Platine neben den Bussteckern eingestellt. Mit den 4 DIP-Schaltern *Device Address* sind die Adressen 1...15 einstellbar (siehe Abb. 1). Es gilt: 1 = DIP-Schalter unten, 0 = DIP-Schalter oben.

Die Adresse 0 deaktiviert das Modul. Auch wenn ein Modul deaktiviert ist, erhalten nachfolgende Module am LS-Bus alle Busnachrichten.

Modul- adresse	Einstellung der DIP-Schalter				
	1	2	3	4	
0	0	0	0	0	Modul deaktiviert
1	1	0	0	0	
2	0	1	0	0	
3	1	1	0	0	
4	0	0	1	0	
5	1	0	1	0	
...	...				
15	1	1	1	1	

Abb. 1 – Moduladressierung mit DIP-Schaltern

Der GND-Pegel des Moduls ist mit der Hutschiene verbunden, die als Schutzleiter (PE) dient. Der Schutzleiter ist mit der Schraube **PE** auf der Oberseite des Moduls verbunden.

### Schutzleiter

### 3 HSM-24V

Das Modul HSM-24V hat 32 digitale Kanäle, die 24V-Signale verarbeiten, und wird am LS-Bus betrieben.

#### 3.1 Hardware

Die 32 Kanäle können in Gruppen von jeweils 8 als Eingänge oder Ausgänge geschaltet werden. Nach dem Einschalten sind alle Kanäle als Eingänge geschaltet.

Das Modul verarbeitet eine Betriebsspannung von 19V... 29V. Der steckbare Versorgungsanschluss liegt rechts unten (siehe Abb. 2) auf dem Modul und hat je 2 Pins für  $V_{CC}$  und GND. Die Versorgungsspannung wird über eine Sicherung (5A, träge) geführt. Beide Masseleitungen sind mit dem Schutzleiter PE verbunden.

Das Modul hat einen integrierten Verpolungsschutz.

Die Kanäle verarbeiten typischerweise 24V-Signale und sind kurzschlussfest. Ein Signal über 82% der Betriebsspannung wird als High-Level verarbeitet, ein Signal unter 66% als Low-Level. Für jeden Kanal zeigt eine LED den Ist-Zustand an: LED ein entspricht High-Level.

Die Kanäle haben einen zulässigen Betriebsstrom von 0mA...150mA. Wenn der Strom 500mA an einem Kanal übersteigt, wird der Kanal automatisch abgeschaltet. Ein Überstrom im Bereich von 150mA...500mA kann den Überhitzungsschutz des Treibers auslösen, d.h. der Treiber wird abgeschaltet, mit samt den zugehörigen 16 Kanälen.

Jeweils 4 Kanäle haben eine gemeinsame GND-Leitung. Die Anschlussleisten sind steckbar und hat Schraubanschlüsse, über welche die Ein- und Ausgänge beschaltet werden.

Die Eingänge haben einen Filter mit ca. 12µs Verzögerung.

Das Modul ist in eine Montagewanne integriert, über die eine Schnapp-Befestigung auf einer DIN-Hutschiene möglich ist. Das Modul darf nur im Schaltschrank betrieben werden (Industrieeinsatz).

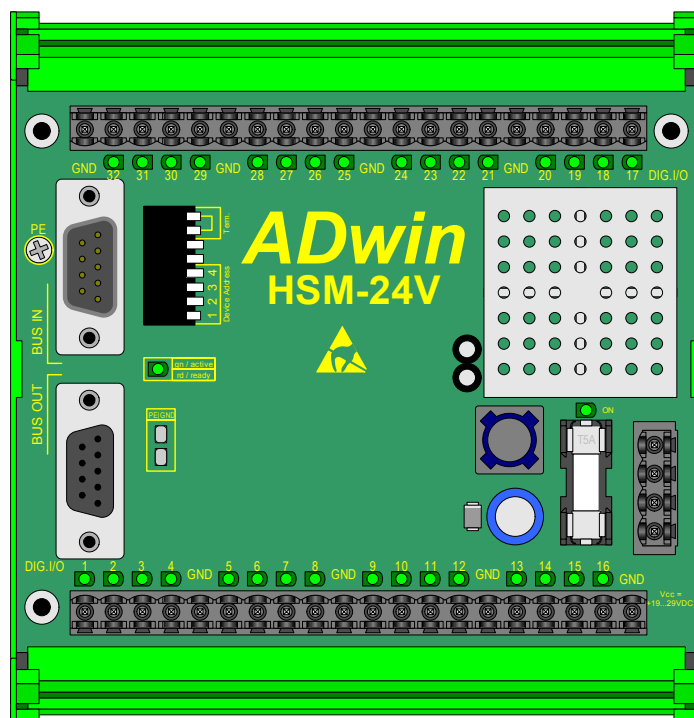


Abb. 2 – Platine HSM-24V





Abb. 3 – Pinbelegung LS-Bus [HSM-24V](#)

### 3.2 Software

Die Funktionen des Moduls [HSM-24V](#) werden mit *ADbasic*-Befehlen komfortabel programmiert. Beachten Sie bitte, dass die Befehle für die verschiedenen *ADwin*-Systeme unterschiedlich sind.

Die Befehle haben folgende Funktion:

Befehl	Funktion
LS_DIO_INIT	HSM-Modul initialisieren.
LS_DIGPROG	Kanäle als Ein- oder Ausgänge programmieren.
LS_WATCHDOG_INIT	Watchdog-Zähler deaktivieren oder aktivieren und die Watchdog-Zeit setzen.
LS_DIG_IO	Digital-Ausgänge setzen und die aktuellen Pegel zurückgeben. Der Befehl gilt nur für 1 Modul.

Abb. 4 – Befehle für [HSM-24V](#), Kurzübersicht

Die Befehle sind jeweils in einer Include-Datei enthalten; binden Sie die zum *ADwin*-System passende Datei am Beginn des Programms ein. Die folgende Liste für die *ADwin*-Systeme die passende Programmzeile und die Seite in dieser Dokumentation, auf der die Befehlsbeschreibung beginnt.

*ADwin-light-16:*      `#INCLUDE ADWL16.INC`      [Seite 6](#)  
*ADwin-Pro:*      `#INCLUDE ADwinPRO_ALL.INC`      [Seite 14](#)

## LS\_DIO\_INIT

**LS\_DIO\_INIT** initialisiert ein Modul vom Typ HSM-24V am LS-Bus und gibt den Fehlerstatus zurück.

## Syntax

```
#INCLUDE ADWL16.inc  
  
ret_val = LS_DIO_INIT(ls-module)
```

## Parameter

ls-module	Eingestellte Moduladresse am LS-Bus (1...15).	LONG
ret_val	Bitmuster, das den Fehlerstatus angibt. Bit = 0: kein Fehler. Bit = 1: Fehler aufgetreten.	LONG

Bit-Nr.	31...8	7	6	5...4	3	2	1	0
Status	–	Temp2	Temp1	–	WD	Time	Ovr	Par

- : don't care (mit 0CFh ausmaskieren)

Par: Parity-Fehler bei der Datenübertragung auf dem LS-Bus.

Ovr: Overrun-Fehler bei der Datenübertragung auf dem LS-Bus.

Time: Timeout-Fehler bei der Datenübertragung auf dem LS-Bus.

WD: Watchdog hat ausgelöst. Die Kanaltreiber sind deaktiviert.

Temp1: Übertemperatur am Treiber für Kanäle 1...16. Treiber ist deaktiviert.

Temp2: Übertemperatur am Treiber für Kanäle 17...32. Treiber ist deaktiviert.

## Bemerkungen

Die Anweisung soll nur im Abschnitt **INIT**: verwendet werden, weil sie eine lange Ausführungszeit hat.

Die Initialisierung setzt folgende Einstellungen:

- Alle DIO-Kanäle werden als Eingang programmiert.  
Andere Einstellungen siehe **LS\_DIGPROG**.
- Der Status für Überstrom (> ca. 500mA) wird zurückgesetzt.
- Der Fehlerstatus für Übertemperatur wird zurückgesetzt.
- Der Fehlerstatus für Timeout auf dem LS-Bus wird zurückgesetzt.

Der Fehler „Übertemperatur“ eines Treibers kann nur auftreten, wenn auf mehreren Kanälen gleichzeitig ein Überstrom im Bereich von 150...500mA anliegt. Unabhängig davon wird bei einem Überstrom über 500mA der betroffene Kanal automatisch abgeschaltet.

Die Kanäle des Moduls HSM-24V dürfen nur im Bereich von 0...150mA betrieben werden.



## Gültig für Module

HSM-24V + L16

## Siehe auch

[LS\\_DIGPROG](#), [LS\\_WATCHDOG\\_INIT](#), [LS\\_DIG\\_IO](#)

## Beispiel

REM Example prozess for one module HSM-24V and ADwin-L16

#INCLUDE ADWL16.inc

### INIT:

```
PROCESSDELAY = 4000000    '10Hz HP
PAR_1 = LS_DIO_INIT(1)
PAR_2 = LS_DIGPROG(1, 0Fh) 'channels 1...32 as output
PAR_3 = LS_WATCHDOG_INIT(1, 1, 1100) 'watchdog time 1,1 sec
```

### EVENT:

```
REM set one channel to high, rotating from 1 to 32
INC PAR_10
IF (PAR_10>=32) THEN PAR_10=0
PAR_11 = SHIFT_LEFT(1,PAR_10)
REM set channels and read back real state
PAR_12 = LS_DIG_IO(PAR_11)
```

## LS\_DIGPROG

**LS\_DIGPROG** programmiert die digitalen Kanäle 1...32 eines Moduls vom Typ HSM-24V am LS-Bus in Gruppen zu 8 als Ein- oder Ausgang.

## Syntax

```
#INCLUDE ADWL16.inc  
  
ret_val = LS_DIGPROG(ls-module, pattern)
```

## Parameter

**ls-module**    Eingestellte Moduladresse (1...15) am LS-Bus. LONG

**pattern**       Bitmuster, nach dem die Kanäle als Ein- oder Ausgang gesetzt werden:  
Bit = 0: Kanal als Eingang setzen.  
Bit = 1: Kanal als Ausgang setzen.

Bitnr.	31...4	3	2	1	0
Kanalnr.	–	32:25	24:17	16:9	8:1

**ret\_val**       Bitmuster, das den Fehlerstatus angibt. LONG  
Bit = 0: kein Fehler.  
Bit = 1: Fehler aufgetreten.

Bit-Nr.	31...8	7	6	5...4	3	2	1	0
Status	–	Temp2	Temp1	–	WD	Time	Ovr	Par

- :don't care (mit 0CFh ausmaskieren)

Par: Parity-Fehler bei der Datenübertragung auf dem LS-Bus.

Ovr: Overrun-Fehler bei der Datenübertragung auf dem LS-Bus.

Time: Timeout-Fehler bei der Datenübertragung auf dem LS-Bus.

WD: Watchdog hat ausgelöst. Die Kanaltreiber sind deaktiviert.

Temp1: Übertemperatur am Treiber für Kanäle 1...16. Treiber ist deaktiviert.

Temp2: Übertemperatur am Treiber für Kanäle 17...32. Treiber ist deaktiviert.

## Bemerkungen

Die Anweisung soll nur im Abschnitt **INIT** verwendet werden, weil sie eine lange Ausführungszeit hat.

Nach der Initialisierung mit **LS\_DIO\_INIT** sind alle Kanäle als Eingänge konfiguriert.

Die Kanäle können nur in Gruppen zu je 8 als Ein- oder Ausgang gesetzt werden (nur 4 relevante Bits, die anderen Bits werden ignoriert).

## Gültig für Module

HSM-24V + L16

## Siehe auch

[LS\\_DIO\\_INIT](#), [LS\\_WATCHDOG\\_INIT](#), [LS\\_DIG\\_IO](#)

## Beispiel

REM Example prozess for one module HSM-24V and ADwin-L16

#INCLUDE ADWL16.inc

### INIT:

```
PROCESSDELAY = 4000000    '10Hz HP
PAR_1 = LS_DIO_INIT(1)
PAR_2 = LS_DIGPROG(1, 0Fh) 'channels 1...32 as output
PAR_3 = LS_WATCHDOG_INIT(1, 1, 1100) 'watchdog time 1,1 sec
```

### EVENT:

```
REM set one channel to high, rotating from 1 to 32
INC PAR_10
IF (PAR_10>=32) THEN PAR_10=0
PAR_11 = SHIFT_LEFT(1,PAR_10)
REM set channels and read back real state
PAR_12 = LS_DIG_IO(PAR_11)
```

## LS\_WATCHDOG\_INIT

**LS\_WATCHDOG\_INIT** aktiviert oder deaktiviert den Watchdog-Zähler eines Moduls am LS-Bus. Beim Aktivieren erhält der Zähler seinen Startwert und wird gestartet.

### Syntax

```
#INCLUDE ADWL16.inc  
  
ret_val = LS_WATCHDOG_INIT(ls-module, enable, time)
```

### Parameter

ls-module	Eingestellte Moduladresse am LS-Bus (1...15).	LONG
enable	Status des Watchdog-Zählers einstellen: 0 : Watchdog-Zähler deaktivieren. 1 : Watchdog-Zähler aktivieren.	LONG
time	Auslösezeit (0...107374) des Zählers in Milli- sekunden.	LONG
ret_val	Bitmuster, das den Fehlerstatus angibt. Bit = 0: kein Fehler. Bit = 1: Fehler aufgetreten.	LONG

Bit-Nr.	31...8	7	6	5...4	3	2	1	0
Status	–	Temp2	Temp1	–	WD	Time	Ovr	Par
- : don't care ( mit 0CFh ausmaskieren)								
Par: Parity-Fehler bei der Datenübertragung auf dem LS-Bus.								
Ovr: Overrun-Fehler bei der Datenübertragung auf dem LS-Bus.								
Time: Timeout-Fehler bei der Datenübertragung auf dem LS-Bus.								
WD: Watchdog hat ausgelöst. Die Kanaltreiber sind deaktiviert.								
Temp1: Übertemperatur am Treiber für Kanäle 1...16. Treiber ist deaktiviert.								
Temp2: Übertemperatur am Treiber für Kanäle 17...32. Treiber ist deaktiviert.								

### Bemerkungen

Die Anweisung soll nur im Abschnitt **INIT** verwendet werden, weil sie eine lange Ausführungszeit hat.

Solange der Watchdog-Zähler aktiv ist, dekrementiert er den Zählerstand kontinuierlich. Nach der eingestellten Auslösezeit erreicht der Zählerstand 0 (Null). Das Modul nimmt nun eine Fehlfunktion an und wird gestoppt; dadurch werden alle Ausgangssignale zurückgesetzt.

Nach dem Einschalten des Moduls ist der Zähler auf den Startwert 10ms eingestellt und der Watchdog ist aktiv.

Setzen Sie den aktiven Watchdog-Zähler während seines Zählvorgangs mindestens einmal zurück, um die Funktion des Moduls zu gewährleisten. Zum Zurücksetzen können modulspezifische Befehle verwendet werden.

Die Watchdog-Funktion dient zur Verbindungsüberwachung zwischen ADwin-System und LS-Bus-Modul.

### Gültig für Module

HSM-24V + L16

### Siehe auch

[LS\\_DIO\\_INIT](#), [LS\\_DIGPROG](#), [LS\\_DIG\\_IO](#)



## Beispiel

REM Example prozess for one module HSM-24V and ADwin-L16

#INCLUDE ADWL16.inc

### INIT:

```
PROCESSDELAY = 4000000    '10Hz HP
PAR_1 = LS_DIO_INIT(1)
PAR_2 = LS_DIGPROG(1, 0Fh) 'channels 1...32 as output
PAR_3 = LS_WATCHDOG_INIT(1, 1, 1100) 'watchdog time 1,1 sec
```

### EVENT:

```
REM set one channel to high, rotating from 1 to 32
INC PAR_10
IF (PAR_10>=32) THEN PAR_10=0
PAR_11 = SHIFT_LEFT(1,PAR_10)
REM set channels and read back real state
PAR_12 = LS_DIG_IO(PAR_11)
```

## LS\_DIG\_IO

**LS\_DIG\_IO** setzt alle Digital-Ausgänge des Moduls HSM-24V am LS-Bus auf den Pegel High oder Low und gibt den Zustand aller Kanäle als Bitmuster zurück.

## Syntax

```
#INCLUDE ADWL16.inc  
  
ret_val = LS_DIG_IO(pattern)
```

## Parameter

**pattern** Bitmuster, mit dem die digitalen Ausgänge gesetzt werden (siehe Tabelle). LONG  
Bit = 0: Ausgang auf Pegel Low setzen.  
Bit = 1: Ausgang auf Pegel High setzen.

**ret\_val** Bitmuster mit dem Ist-Zustand aller digitalen Kanäle (siehe Tabelle). LONG  
Bit = 0: Pegel Low liegt an.  
Bit = 1: Pegel High liegt an.

Bitnr.	31	30	29	...	2	1	0
Eingang	32	31	30	...	3	2	1

## Bemerkungen

**LS\_DIG\_IO** arbeitet nur korrekt, wenn folgende Voraussetzungen erfüllt sind:

- Am LS-Bus ist nur ein Modul angeschlossen.
- Das Modul ist vom Typ HSM-24V.
- Am Modul ist die Moduladresse 1 eingestellt.

Die Kanäle werden mit **LS\_DIGPROG** als Ein- oder Ausgänge programmiert.

Das Bitmuster **pattern** wird nur für die Kanäle angewendet, die als Ausgang programmiert sind. Bits für Eingänge werden ignoriert.

Der Rückgabewert enthält den tatsächlichen Schaltzustand sowohl von Eingängen wie von Ausgängen. Beachten Sie: Die Eingänge haben einen Filter mit ca. 12µs Verzögerung.

**LS\_DIG\_IO** setzt den Watchdog-Zähler des Moduls auf den Startwert zurück. Der Zähler bleibt aktiv. Der Startwert wird mit **LS\_WATCHDOG\_INIT** eingestellt.

Setzen Sie den aktiven Watchdog-Zähler während seines Zählvorgangs mindestens einmal zurück, um die Funktion des Moduls zu gewährleisten.

## Gültig für Module

HSM-24V + L16

## Siehe auch

[LS\\_DIO\\_INIT](#), [LS\\_DIGPROG](#), [LS\\_WATCHDOG\\_INIT](#),



## Beispiel

REM Example prozess for one module HSM-24V and ADwin-L16

#INCLUDE ADWL16.inc

### INIT:

PROCESSDELAY = 4000000 '10Hz HP

PAR\_1 = LS\_DIO\_INIT(1)

PAR\_2 = LS\_DIGPROG(1, 0Fh) 'channels 1...32 as output

PAR\_3 = LS\_WATCHDOG\_INIT(1, 1, 1100) 'watchdog time 1,1 sec

### EVENT:

REM set one channel to high, rotating from 1 to 32

INC PAR\_10

IF (PAR\_10>=32) THEN PAR\_10=0

PAR\_11 = SHIFT\_LEFT(1,PAR\_10)

REM set channels and read back real state

PAR\_12 = LS\_DIG\_IO(PAR\_11)

## LS\_DIO\_INIT

**LS\_DIO\_INIT** initialisiert ein Modul vom Typ HSM-24V am LS-Bus über eine Schnittstelle des Pro-Moduls und gibt den Fehlerstatus zurück.

### Syntax

```
#INCLUDE ADwinPRO_ALL.inc

ret_val = LS_DIO_INIT(module, channel, ls-module)
```

### Parameter

<code>module</code>	Eingestellte Adresse des Pro-Moduls (1...255).	LONG
<code>channel</code>	Nummer (1, 2) der LS-Bus-Schnittstelle auf dem Pro-Modul.	LONG
<code>ls-module</code>	Eingestellte Moduladresse am LS-Bus (1...15).	LONG
<code>ret_val</code>	Bitmuster, das den Fehlerstatus angibt. Bit = 0: kein Fehler. Bit = 1: Fehler aufgetreten.	LONG

Bit-Nr.	31...8	7	6	5...4	3	2	1	0
Status	–	Temp2	Temp1	–	WD	Time	Ovr	Par

- : don't care (mit 0CFh ausmaskieren)

Par: Parity-Fehler bei der Datenübertragung auf dem LS-Bus.

Ovr: Overrun-Fehler bei der Datenübertragung auf dem LS-Bus.

Time: Timeout-Fehler bei der Datenübertragung auf dem LS-Bus.

WD: Watchdog hat ausgelöst. Die Kanaltreiber sind deaktiviert.

Temp1: Übertemperatur am Treiber für Kanäle 1...16. Treiber ist deaktiviert.

Temp2: Übertemperatur am Treiber für Kanäle 17...32. Treiber ist deaktiviert.

### Bemerkungen

Die Anweisung soll nur im Abschnitt **INIT**: verwendet werden, weil sie eine lange Ausführungszeit hat.

Die Initialisierung setzt folgende Einstellungen:

- Alle DIO-Kanäle werden als Eingang programmiert.  
Andere Einstellungen siehe **LS\_DIGPROG**.
- Der Status für Überstrom (> ca. 500mA) wird zurückgesetzt.
- Der Fehlerstatus für Übertemperatur wird zurückgesetzt.
- Der Fehlerstatus für Timeout auf dem LS-Bus wird zurückgesetzt.

Der Fehler „Übertemperatur“ eines Treibers kann nur auftreten, wenn auf mehreren Kanälen gleichzeitig ein Überstrom im Bereich von 150...500mA anliegt. Unabhängig davon wird bei einem Überstrom über 500mA der betroffene Kanal automatisch abgeschaltet.

Die Kanäle des Moduls HSM-24V dürfen nur im Bereich von 0...150mA betrieben werden.

### Gültig für Module

LS-2 Rev. A

### Siehe auch

[LS\\_DIGPROG](#), [LS\\_WATCHDOG\\_INIT](#), [LS\\_DIG\\_IO](#)



## Beispiel

```
REM Example prozess for one module HSM-24V and ADwin-Pro-LS2
#include ADwinPRO_ALL.inc

INIT:
    PROCESSDELAY = 4000000    '10Hz HP
    PAR_1 = LS_DIO_INIT(1,2,1)
    PAR_2 = LS_DIGPROG(1,2,1,0Fh) 'channels 1...32 as output
    PAR_3 = LS_WATCHDOG_INIT(1,2,1,1,1100) 'watchdog time 1.1 sec

EVENT:
    REM set one channel to high, rotating from 1 to 32
    INC PAR_10
    IF (PAR_10>=32) THEN PAR_10=0
    PAR_11 = SHIFT_LEFT(1,PAR_10)
    REM set channels and read back real state
    PAR_12 = LS_DIG_IO(1,2,PAR_11)
```

## LS\_DIGPROG

**LS\_DIGPROG** programmiert die digitalen Kanäle 1...32 eines Moduls vom Typ HSM-24V am LS-Bus über eine Schnittstelle des Pro-Moduls in Gruppen zu 8 als Ein- oder Ausgang.

## Syntax

```
#INCLUDE ADwinPRO_ALL.inc  
  
ret_val = LS_DIGPROG(module, channel, ls-module,  
                    pattern)
```

## Parameter

module	Eingestellte Adresse des Pro-Moduls (1...255).	LONG
channel	Nummer (1, 2) der LS-Bus-Schnittstelle auf dem Pro-Modul.	LONG
ls-module	Eingestellte Moduladresse (1...15) am LS-Bus.	LONG
pattern	Bitmuster, nach dem die Kanäle als Ein- oder Ausgang gesetzt werden: Bit = 0: Kanal als Eingang setzen. Bit = 1: Kanal als Ausgang setzen.	LONG

Bitnr.	31...4	3	2	1	0
Kanalnr.	–	32:25	24:17	16:9	8:1

ret_val	Bitmuster, das den Fehlerstatus angibt. Bit = 0: kein Fehler. Bit = 1: Fehler aufgetreten.	LONG
---------	--	------

Bit-Nr.	31...8	7	6	5...4	3	2	1	0
Status	–	Temp2	Temp1	–	WD	Time	Ovr	Par

- :don't care (mit 0CFh ausmaskieren)

Par:Parity-Fehler bei der Datenübertragung auf dem LS-Bus.

Ovr: Overrun-Fehler bei der Datenübertragung auf dem LS-Bus.

Time: Timeout-Fehler bei der Datenübertragung auf dem LS-Bus.

WD: Watchdog hat ausgelöst. Die Kanaltreiber sind deaktiviert.

Temp1: Übertemperatur am Treiber für Kanäle 1...16. Treiber ist deaktiviert.

Temp2: Übertemperatur am Treiber für Kanäle 17...32. Treiber ist deaktiviert.

## Bemerkungen

Die Anweisung soll nur im Abschnitt **INIT**: verwendet werden, weil sie eine lange Ausführungszeit hat.

Nach der Initialisierung mit **LS\_DIO\_INIT** sind alle Kanäle als Eingänge konfiguriert.

Die Kanäle können nur in Gruppen zu je 8 als Ein- oder Ausgang gesetzt werden (nur 4 relevante Bits, die anderen Bits werden ignoriert).

## Gültig für Module

LS-2 Rev. A

## Siehe auch

[LS\\_DIO\\_INIT](#), [LS\\_WATCHDOG\\_INIT](#), [LS\\_DIG\\_IO](#)

## Beispiel

```
REM Example prozess for one module HSM-24V and ADwin-Pro-LS2
#include ADwinPRO_ALL.inc

INIT:
    PROCESSDELAY = 4000000    '10Hz HP
    PAR_1 = LS_DIO_INIT(1,2,1)
    PAR_2 = LS_DIGPROG(1,2,1,0Fh) 'channels 1...32 as output
    PAR_3 = LS_WATCHDOG_INIT(1,2,1,1,1100) 'watchdog time 1.1 sec

EVENT:
    REM set one channel to high, rotating from 1 to 32
    INC PAR_10
    IF (PAR_10>=32) THEN PAR_10=0
    PAR_11 = SHIFT_LEFT(1,PAR_10)
    REM set channels and read back real state
    PAR_12 = LS_DIG_IO(1,2,PAR_11)
```

## LS\_WATCHDOG\_INIT

**LS\_WATCHDOG\_INIT** aktiviert oder deaktiviert den Watchdog-Zähler eines Moduls am LS-Bus über eine Schnittstelle des Pro-Moduls.

Beim Aktivieren erhält der Zähler seinen Startwert und wird gestartet.

### Syntax

```
#INCLUDE ADwinPRO_ALL.inc

ret_val = LS_WATCHDOG_INIT(module, channel,
                             ls-module, enable, time)
```

### Parameter

<code>module</code>	Eingestellte Adresse des Pro-Moduls (1...255).	LONG
<code>channel</code>	Nummer (1, 2) der LS-Bus-Schnittstelle auf dem Pro-Modul.	LONG
<code>ls-module</code>	Eingestellte Moduladresse am LS-Bus (1...15).	LONG
<code>enable</code>	Status des Watchdog-Zählers einstellen: 0 : Watchdog-Zähler deaktivieren. 1 : Watchdog-Zähler aktivieren.	LONG
<code>time</code>	Auslösezeit (0...107374) des Zählers in Millisekunden.	LONG
<code>ret_val</code>	Bitmuster, das den Fehlerstatus angibt. Bit = 0: kein Fehler. Bit = 1: Fehler aufgetreten.	LONG

Bit-Nr.	31...8	7	6	5...4	3	2	1	0
Status	–	Temp2	Temp1	–	WD	Time	Ovr	Par

- : don't care ( mit 0CFh ausmaskieren)  
 Par: Parity-Fehler bei der Datenübertragung auf dem LS-Bus.  
 Ovr: Overrun-Fehler bei der Datenübertragung auf dem LS-Bus.  
 Time: Timeout-Fehler bei der Datenübertragung auf dem LS-Bus.  
 WD: Watchdog hat ausgelöst. Die Kanaltreiber sind deaktiviert.  
 Temp1: Übertemperatur am Treiber für Kanäle 1...16. Treiber ist deaktiviert.  
 Temp2: Übertemperatur am Treiber für Kanäle 17...32. Treiber ist deaktiviert.

### Bemerkungen

Die Anweisung soll nur im Abschnitt **INIT**: verwendet werden, weil sie eine lange Ausführungszeit hat.

Solange der Watchdog-Zähler aktiv ist, dekrementiert er den Zählerstand kontinuierlich. Nach der eingestellten Auslösezeit erreicht der Zählerstand 0 (Null). Das Modul nimmt nun eine Fehlfunktion an und wird gestoppt; dadurch werden alle Ausgangssignale zurückgesetzt.

Nach dem Einschalten des Moduls ist der Zähler auf den Startwert 10ms eingestellt und der Watchdog ist aktiv.

Setzen Sie den aktiven Watchdog-Zähler während seines Zählvorgangs mindestens einmal zurück, um die Funktion des Moduls zu gewährleisten. Zum Zurücksetzen können modulspezifische Befehle verwendet werden.

Die Watchdog-Funktion dient zur Verbindungsüberwachung zwischen ADwin-System und LS-Bus-Modul.

### Gültig für Module

LS-2 Rev. A



## Siehe auch

[LS\\_DIO\\_INIT](#), [LS\\_DIGPROG](#), [LS\\_DIG\\_IO](#)

## Beispiel

REM Example prozess for one module HSM-24V and ADwin-Pro-LS2

**#INCLUDE** ADwinPRO\_ALL.inc

### INIT:

**PROCESSDELAY** = 4000000    '10Hz HP

**PAR\_1** = **LS\_DIO\_INIT**(1,2,1)

**PAR\_2** = **LS\_DIGPROG**(1,2,1,0Fh) 'channels 1...32 as output

**PAR\_3** = **LS\_WATCHDOG\_INIT**(1,2,1,1,1100) 'watchdog time 1.1 sec

### EVENT:

REM set one channel to high, rotating from 1 to 32

**INC** **PAR\_10**

**IF** (**PAR\_10**>=32) **THEN** **PAR\_10**=0

**PAR\_11** = **SHIFT\_LEFT**(1,**PAR\_10**)

REM set channels and read back real state

**PAR\_12** = **LS\_DIG\_IO**(1,2,**PAR\_11**)

## LS\_DIG\_IO

**LS\_DIG\_IO** setzt alle Digital-Ausgänge des Moduls HSM-24V am LS-Bus über eine Schnittstelle des Pro-Moduls auf den Pegel High oder Low und gibt den Zustand aller Kanäle als Bitmuster zurück.

### Syntax

```
#INCLUDE ADwinPRO_ALL.inc

ret_val = LS_DIG_IO(module, channel, pattern)
```

### Parameter

<code>module</code>	Eingestellte Adresse des Pro-Moduls (1...255).	LONG
<code>channel</code>	Nummer (1, 2) der LS-Bus-Schnittstelle auf dem Pro-Modul.	LONG
<code>pattern</code>	Bitmuster, mit dem die digitalen Ausgänge gesetzt werden (siehe Tabelle). Bit = 0: Ausgang auf Pegel Low setzen. Bit = 1: Ausgang auf Pegel High setzen.	LONG
<code>ret_val</code>	Bitmuster mit dem Ist-Zustand aller digitalen Kanäle (siehe Tabelle). Bit = 0: Pegel Low liegt an. Bit = 1: Pegel High liegt an.	LONG

Bitnr.	31	30	29	...	2	1	0
Eingang	32	31	30	...	3	2	1

### Bemerkungen

**LS\_DIG\_IO** arbeitet nur korrekt, wenn folgende Voraussetzungen erfüllt sind:

- Am LS-Bus ist nur ein Modul angeschlossen.
- Das Modul ist vom Typ HSM-24V.
- Am Modul ist die Moduladresse 1 eingestellt.

Die Kanäle werden mit **LS\_DIGPROG** als Ein- oder Ausgänge programmiert.

Das Bitmuster `pattern` wird nur für die Kanäle angewendet, die als Ausgang programmiert sind. Bits für Eingänge werden ignoriert.

Der Rückgabewert enthält den tatsächlichen Schaltzustand sowohl von Eingängen wie von Ausgängen. Beachten Sie: Die Eingänge haben einen Filter mit ca. 12µs Verzögerung.

**LS\_DIG\_IO** setzt den Watchdog-Zähler des Moduls auf den Startwert zurück. Der Zähler bleibt aktiv. Der Startwert wird mit **LS\_WATCHDOG\_INIT** eingestellt.

Setzen Sie den aktiven Watchdog-Zähler während seines Zählvorgangs mindestens einmal zurück, um die Funktion des Moduls zu gewährleisten.

### Gültig für Module

LS-2 Rev. A

### Siehe auch

[LS\\_DIO\\_INIT](#), [LS\\_DIGPROG](#), [LS\\_WATCHDOG\\_INIT](#),





## Beispiel

```
REM Example prozess for one module HSM-24V and ADwin-Pro-LS2
#include ADwinPRO_ALL.inc

INIT:
    PROCESSDELAY = 4000000    '10Hz HP
    PAR_1 = LS_DIO_INIT(1,2,1)
    PAR_2 = LS_DIGPROG(1,2,1,0Fh) 'channels 1...32 as output
    PAR_3 = LS_WATCHDOG_INIT(1,2,1,1,1100) 'watchdog time 1.1 sec

EVENT:
    REM set one channel to high, rotating from 1 to 32
    INC PAR_10
    IF (PAR_10>=32) THEN PAR_10=0
    PAR_11 = SHIFT_LEFT(1,PAR_10)
    REM set channels and read back real state
    PAR_12 = LS_DIG_IO(1,2,PAR_11)
```