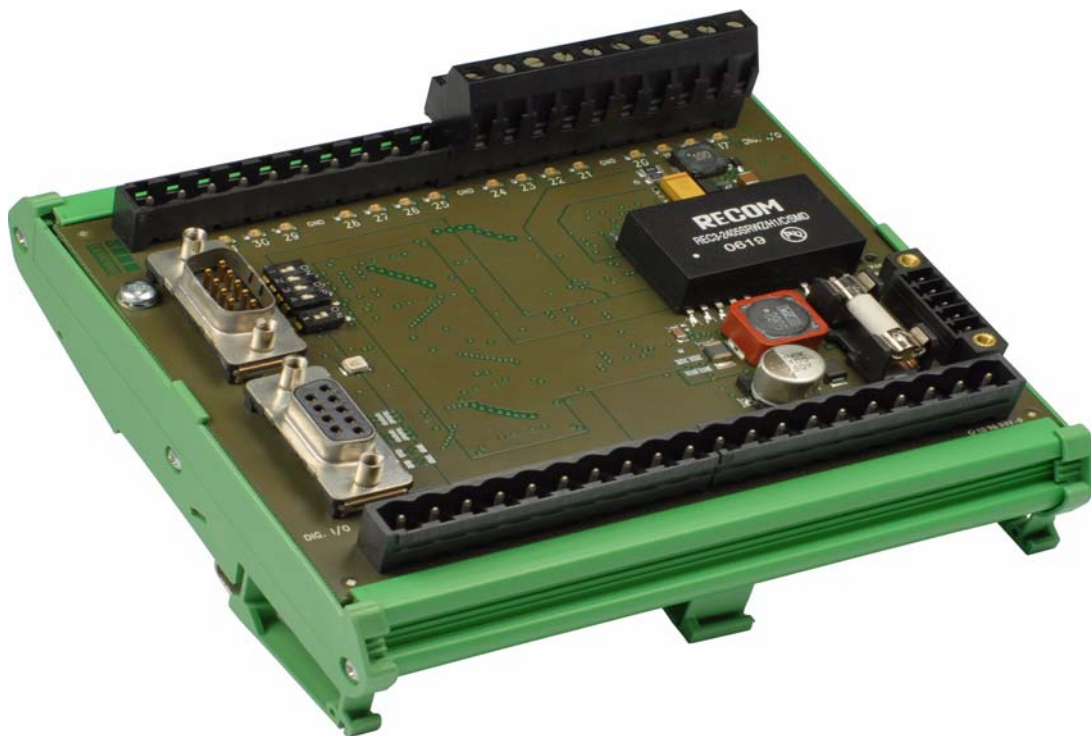


# ***ADwin HSM-24V***

**Module for LS Bus**

**Manual**



**For any questions, please don't hesitate to contact us:**

Hotline: +49 6251 96320  
Fax: +49 6251 56819  
E-Mail: [info@ADwin.de](mailto:info@ADwin.de)  
Internet: [www.ADwin.de](http://www.ADwin.de)



Jäger Computergesteuerte  
Messtechnik GmbH  
Rheinstraße 2-4  
D-64653 Lorsch  
Germany

## Table of Contents

Typographical Conventions .....	IV
1 Information about this Manual .....	1
2 The LS bus .....	2
3 HSM-24V .....	4
3.1 Hardware .....	4
3.2 Software .....	5
3.2.1 LS Bus + ADwin-light-16 .....	6
3.2.2 LS-Bus + Pro I. ....	14

## Typographical Conventions



<C:\ADwin\ ...>

Programmtext

"Warning" stands for information, which indicate damages of hardware or software, test setup or injury to persons caused by incorrect handling.

You find "note" next to

- information, which have absolutely to be considered in order to guarantee an operation without any errors.
- advice for efficient operation.

"Information" refers to further information in this documentation or to other sources such as manuals, data sheets, literature, etc.

File names and paths are placed in angle brackets and characterized in the font Courier New.

Program instructions and user inputs are characterized in the font Courier New

*ADbasic* source code elements such as **INSTRUCTIONS**, *variables*, *comment* and *other text* are characterized by colours and the font Courier New (as with the editor of the development environment *ADbasic*).

Bits in data (here: 16 bit) are referred to as follows:

Bit No.	15	14	13	...	01	00
Bit value	$2^{15}$	$2^{14}$	$2^{13}$	...	$2^1=2$	$2^0=1$
Synonym	MSB	-	-	-	-	LSB

### 1 Information about this Manual

This manual describes the LS bus and the modules being operated on the LS bus. Additional information are available in

- the description of the LS bus interface in the hardware manual for *ADwin-light-16*, *ADwin-Gold* or *ADwin-Pro*.
- the manual *ADbasic*, which describes the basic instructions for the compiler of same denominator as well explains the function principle of *ADwin* systems.

The online help has the same content as the manual and additionally contains the hardware related instructions, LS bus too.

#### Please note:

To have your ADwin systems work properly, keep strictly to the information given in this documentation and in other mentioned manuals.

Programming, start-up and operation, as well as the modification of program parameters must be performed only by appropriately qualified personnel.

Qualified personnel are persons who, due to their education, experience and training as well as their knowledge of applicable technical standards, guidelines, accident prevention regulations and operating conditions, have been authorized by a quality assurance representative at the site to perform the necessary activities, while recognizing and avoiding any possible dangers. (Definition of qualified personnel as per VDE 105 and ICE 60364).

This product documentation and all documents referred to, have always to be available and to be observed. For damages caused by disregarding the information in this documentation or in all other additional documentations, no liability is assumed by the company Jäger Computergesteuerte Messtechnik GmbH, Lorsch, Germany.

This documentation, including all pictures is protected by copyright. Reproduction, translation as well as electronical and photographic archiving and modification require a written permission by the company Jäger Computergesteuerte Messtechnik GmbH, Lorsch, Germany.

OEM products are mentioned without referring to possible patent rights, whose existence is not to be excluded.

Subject to change.

Hotline address: see inner side of cover page.



#### Qualified personnel

#### Availability of Documents

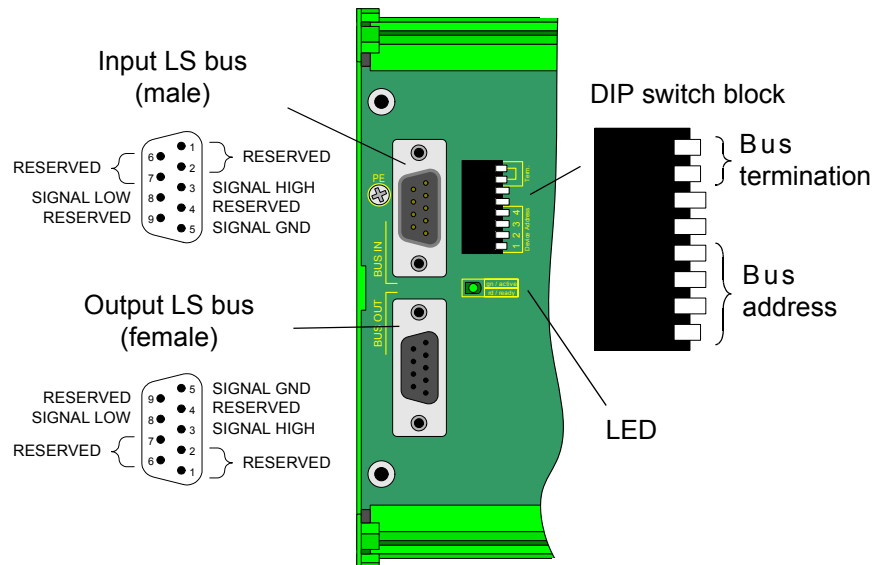


#### Legal instructions

## 2 The LS bus

The LS bus is a bi-directional serial bus with 5MHz clock rate. The bus connects an ADwin system via its LS bus interface with up to 15 LS bus modules.

Figure 1 shows the standard connections of an LS bus module: Bus input and output with LED, protection earth PE, DIP switch for bus termination and bus address.



The bus is set up as line connection, i.e. the interface and the LS bus modules are connected to each other via two-way links. Each module has a female DSub connector (9-pole) as bus input and a male DSub connector (9-pole) as bus output. The maximum bus length is 5m.

The LED besides bus Bus input/output indicates data traffic on the LS bus:

- Green LED: The module receives or sends data.
- Red LED: Data for other modules is sent on the bus.
- LED off: No data traffic.

### Bus Termination

The bus termination on the last module of the LS bus must be activated with the DIP switches  $T_{erm.}$ , deactivated on all other modules. To activate the bus termination both DIP switches are set down.

### Bus address

Each module on the LS bus is addresses via its bus address; therefore the address must be unique for each module.

The bus address is set manually with the DIP switch block on the PCB besides the bus connectors (see [fig. 1](#)). Using the 4 DIP switches  $Device\ Address$  the address may be set to 1...15. The setting is: 1 = DIP switch down, 0 = DIP switch up.

Address 0 disables the module. Even if a module is disabled, the following modules on the LS bus receive all bus data.

Module address	Setting of DIP switches				Module disabled
	1	2	3	4	
0	0	0	0	0	
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	

Fig. 1 – Module addressing with DIP switches

The GND level of the module is connected to the top hat rail, which serves as protection earth (PE). Protection earth is connected to the screw **PE** on the module's top.

**Protection Earth**

### 3 HSM-24V

The LS bus module HSM-24V provides 32 digital channels which process 24V signals.

#### 3.1 Hardware

The 32 channels can be set to inputs or outputs in groups of 8. After power-up all channels are set as inputs.

The module processes a supply voltage of 19V... 29V. The supply connector plug is placed bottom right (see [fig. 2](#)) on the module and has each 2 pins for  $V_{CC}$  and GND. The supply voltage is led to a fuse (5A, delay-action). Both ground pins are connected to protection earth PE.

The supply connection is reverse polarity safe.

The channels are designed to process 24V signals typically and are short-circuit-proof. A signal above 82% of supply voltage is processed as High level, a signal below 66% as Low level. For each channel there is a LED indicating the status: LED on relates to High level.

The channels have a permissible operation current of 0mA...150mA. If the current exceeds 500mA at a channel, the channel is automatically switched off. An over-current in the range of 150mA...500mA may activate the super-heating protection of the driver, i.e. the driver is switched off, including the corresponding 16 channels.

Each 4 channels have a common GND. The input / output wires are connected to plug-in blocks of binding posts.

The inputs have a filter causing about 12µs signal delay.

The module is easily snapped onto the DIN top hat rail. The module may be processed inside a control cabinet only (industrial use).

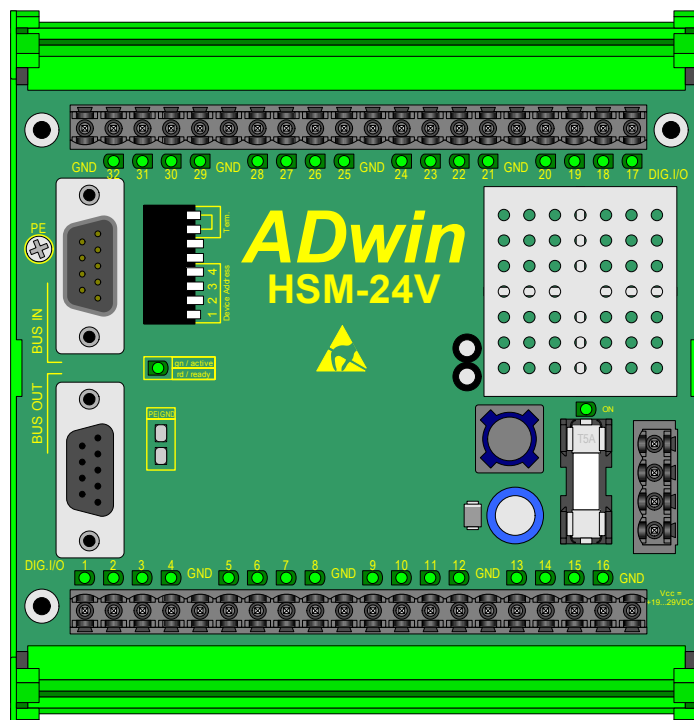


Fig. 2 – Board HSM-24V



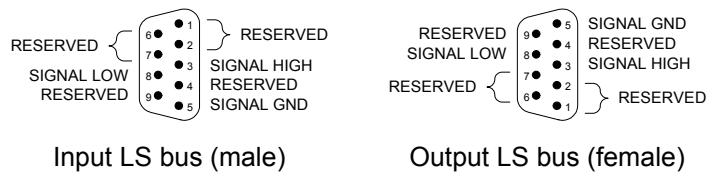


Fig. 3 – Pin assignment LS bus [HSM-24V](#)

### 3.2 Software

The functions of the module [HSM-24V](#) are easily programmed with *ADbasic* instructions. Please note, that instructions for the *ADwin* systems are different.

The instructions have the following function:

Instruction	Function
LS_DIO_INIT	Initialize modul HSM-24V.
LS_DIGPROG	Set channels as inputs or outputs.
LS_WATCHDOG_INIT	Disable watchdog counter or enable and set watchdog time.
LS_DIG_IO	Set level of digital outputs and return current levels. This instruction is valid for 1 module only.

Fig. 4 – Instructions for [HSM-24V](#), overview

Each set of instructions is contained in an include file; include the file appropriate to the *ADwin* system at the top of the program. The following list shows the required program lines for the *ADwin* systems and the page number where the description starts in this manual.

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

## LS\_DIO\_INIT

**LS\_DIO\_INIT** initializes the specified module of type HSM-24V on the LS bus and returns the error status.

## Syntax

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

## Parameters

**ls-module** Specified module address on the LS bus (1...15). LONG

**ret\_val** Bit pattern representing the error status. LONG  
Bit = 0: No error.  
Bit = 1: Error occurred.

Bit no.	31...8	7	6	5...4	3	2	1	0
Status	–	Temp2	Temp1	–	WD	Time	Ovr	Par

- :don't care (mask with 0CFh).

Par:Parity error during data transfer on the LS bus.

Ovr:Overrun error during data transfer on the LS bus.

Time:Timeout error during data transfer on the LS bus.

WD:Watchdog was released. The channel drivers are deactivated.

Temp1:Superheating on driver for channels 1...16. Driver is deactivated.

Temp2:Superheating on driver for channels 17...32. Driver is deactivated.

## Notes

The instruction only be used in section **INIT** :, since it takes long processing time.

The initialization does the following settings:

- All DIO channels are set as inputs.  
Other settings see **LS\_DIGPROG**.
- The over-current status (> ca. 500mA) is reset.
- The error status for superheating is reset.
- The error status for timeout on the LS bus is reset.

The error "superheating" of a driver may only occur, if over-current in the range of 150...500mA is present on several channels at the same time. Irrespective of this, an over-current of mor than 500mA automatically switches off the concerned channel.

The channels of the module HSM-24V may only be operated in the range of 0...150mA.



## To be used for the modules

HSM-24V + L16

## See also

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

## Example

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** sets the digital channels 1...32 of the specified module of type HSM-24V on the LS bus as inputs or outputs in groups of 8.

### Syntax

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

### Parameters

**ls-module** Specified module address on the LS bus (1...15). LONG

**pattern** Bit pattern, setting the channels as inputs or outputs:  
Bit = 0: Set channels as inputs.  
Bit = 1: Set channels as outputs.

Bit No.	31...4	3	2	1	0
Channel no.	–	32:25	24:17	16:9	8:1

**ret\_val** Bit pattern representing the error status. LONG  
Bit = 0: No error.  
Bit = 1: Error occurred.

Bit no.	31...8	7	6	5...4	3	2	1	0
Status	–	Temp2	Temp1	–	WD	Time	Ovr	Par

- :don't care (mask with 0CFh).

Par:Parity error during data transfer on the LS bus.

Ovr:Overflow error during data transfer on the LS bus.

Time:Timeout error during data transfer on the LS bus.

WD:Watchdog was released. The channel drivers are deactivated.

Temp1:Superheating on driver for channels 1...16. Driver is deactivated.

Temp2:Superheating on driver for channels 17...32. Driver is deactivated.

### Notes

The instruction only be used in section **INIT** : , since it takes long processing time.

After initialization with **LS\_DIO\_INIT** all channels are set as inputs.

The channels may be set as inputs or outputs in groups of 8 only (4 relevant bits only, other bits are ignored).

### To be used for the modules

HSM-24V + L16

### See also

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

## Example

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** sets all digital outputs of the specified module HSM-24V on the LS bus to the level High oder Low and returns the status of all channels as bit pattern.

## Syntax

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

## Parameters

**pattern** Bit pattern, setting the digital outputs (see table). LONG  
Bit = 0: Set outputs to level Low.  
Bit = 1: Set outputs to level High.

**ret\_val** Bit pattern representing the real state of all digital channels (see table). LONG  
Bit = 0: Channel has level Low.  
Bit = 1: Channel has level High.

Bit No.	31	30	29	...	2	1	0
Input no.	32	31	30	...	3	2	1

## Notes

**LS\_DIG\_IO** only runs correctly, if the following conditions are given:

- There is only one module on the LS bus.
- The module is of type HSM-24V.
- The module's address is set to 1.

The channels are set as inputs or outputs using **LS\_DIGPROG**.

The **pattern** is applied to those channels only, which are set as outputs. Bits for input channels are ignored.

The return value contains the real state of both inputs and outputs. The inputs have a filter causing about 12µs signal delay.

**LS\_DIG\_IO** resets the watchdog counter of the module to the start value. The counter remains enabled. The start value is set using **LS\_WATCHDOG\_INIT**.

Reset the active watchdog timer at least once to the start value within the counting interval, in order to keep the module working.

## To be used for the modules

HSM-24V + L16

## See also

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

## Example

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** enables or disables the watchdog counter of a specified module on the LS bus. If enabled, the counter is set to the start value and is started.

### Syntax

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

### Parameters

ls-module	Specified module address on the LS bus (1...15).	LONG
enable	Set status of watchdog counter: 0 : Disable watchdog counter. 1 : Enable watchdog counter.	LONG
time	Release time (0...107374) of the counter in milliseconds.	LONG
ret_val	Bit pattern representing the error status. Bit = 0: No error. Bit = 1: Error occurred.	LONG

Bit no.	31...8	7	6	5...4	3	2	1	0
Status	–	Temp2	Temp1	–	WD	Time	Ovr	Par

- : don't care (mask with 0CFh)

Par: Parity error during data transfer on the LS bus.

Ovr: Overrun error during data transfer on the LS bus.

Time: Timeout error during data transfer on the LS bus.

WD: Watchdog was released. The channel drivers are deactivated.

Temp1: Superheating on driver for channels 1...16. Driver is deactivated.

Temp2: Superheating on driver for channels 17...32. Driver is deactivated.

### Notes

The instruction only be used in section **INIT** : , since it takes long processing time.

As long as the watchdog counter is enabled, it decrements the counter value continuously. After the set release time the counter value reaches 0 (zero). If so, the module assumes a malfunction and stops; thus, all output signals are reset.

After power-up of the module the counter is set to the start value 10ms and the watchdog counter is enabled.

Reset the active watchdog timer at least once to the start value within the counting interval, in order to keep the module working. To reset the module use any module specific instruction or **LS\_WATCHDOG\_RESET**.

The watchdog function is used as to monitor the connection between ADwin system and LS bus module.

### To be used for the modules

HSM-24V + L16

### See also

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





## Example

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** initializes the specified module of type HSM-24V on the LS bus via an interface of the Pro module and returns the error status

## Syntax

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

## Parameters

module	Specified module address (1...255).	LONG
channel	number (1, 2) of the LS bus interface on the Pro module.	LONG
ls-module	Specified module address on the LS bus (1...15).	LONG
ret_val	Bit pattern representing the error status. Bit = 0: No error. Bit = 1: Error occurred.	LONG

Bit no.	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.

## Notes

The instruction only be used in section **INIT** : , since it takes long processing time.

The initialization does the following settings:

- All DIO channels are set as inputs.  
Other settings see **LS\_DIGPROG**.
- The over-current status (> ca. 500mA) is reset.
- The error status for superheating is reset.
- The error status for timeout on the LS bus is reset.

The error "superheating" of a driver may only occur, if over-current in the range of 150...500mA is present on several channels at the same time. Irrespective of this, an over-current of mor than 500mA automatically switches off the concerned channel.

The channels of the module HSM-24V may only be operated in the range of 0...150mA.



## To be used for the modules

LS-2 Rev. A

## See also

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

## Example

```
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** sets the digital channels 1...32 of the specified module of type HSM-24V on the LS bus as inputs or outputs in groups of 8 via an interface of the Pro module.

### Syntax

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

### Parameters

module	Specified module address (1...255).	LONG
channel	number (1, 2) of the LS bus interface on the Pro module.	LONG
ls-module	Specified module address on the LS bus (1...15).	LONG
pattern	Bit pattern, setting the channels as inputs or outputs: Bit = 0: Set channels as inputs. Bit = 1: Set channels as outputs.	LONG

Bit No.	31...4	3	2	1	0
Channel no.	–	32:25	24:17	16:9	8:1

ret_val	Bit pattern representing the error status. Bit = 0: No error. Bit = 1: Error occurred.	LONG
---------	--	------

Bit no.	31...8	7	6	5...4	3	2	1	0
Status	–	Temp2	Temp1	–	WD	Time	Ovr	Par

- :don't care (mask with 0CFh).

Par:Parity error during data transfer on the LS bus.

Ovr:Overflow error during data transfer on the LS bus.

Time:Timeout error during data transfer on the LS bus.

WD:Watchdog was released. The channel drivers are deactivated.

Temp1:Superheating on driver for channels 1...16. Driver is deactivated.

Temp2:Superheating on driver for channels 17...32. Driver is deactivated.

### Notes

The instruction only be used in section **INIT** , since it takes long processing time.

After initialization with **LS\_DIO\_INIT** all channels are set as inputs.

The channels may be set as inputs or outputs in groups of 8 only (4 relevant bits only, other bits are ignored).

### To be used for the modules

LS-2 Rev. A

### See also

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

## Example

```
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** sets all digital outputs of the specified module HSM-24V on the LS bus to the level High oder Low and returns the status of all channels as bit pattern.

### Syntax

```
#INCLUDE ADwinPRO_ALL.inc

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

### Parameters

<code>module</code>	Specified module address (1...255).	LONG
<code>channel</code>	number (1, 2) of the LS bus interface on the Pro module.	LONG
<code>pattern</code>	Bit pattern, setting the digital outputs (see table). Bit = 0: Set outputs to level Low. Bit = 1: Set outputs to level High.	LONG
<code>ret_val</code>	Bit pattern representing the real state of all digital channels (see table). Bit = 0: Channel has level Low. Bit = 1: Channel has level High.	LONG

Bit No.	31	30	29	...	2	1	0
Input no.	32	31	30	...	3	2	1

### Notes

**LS\_DIG\_IO** only runs correctly, if the following conditions are given:

- There is only one module on the LS bus.
- The module is of type HSM-24V.
- The module's address is set to 1.

The channels are set as inputs or outputs using **LS\_DIGPROG**.

The `pattern` is applied to those channels only, which are set as outputs. Bits for input channels are ignored.

The return value contains the real state of both inputs and outputs. The inputs have a filter causing about 12µs signal delay.

**LS\_DIG\_IO** resets the watchdog counter of the module to the start value. The counter remains enabled. The start value is set using **LS\_WATCHDOG\_INIT**.

Reset the active watchdog timer at least once to the start value within the counting interval, in order to keep the module working.

### To be used for the modules

LS-2 Rev. A

### See also

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

## Example

```
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** enables or disables the watchdog counter of a specified module on the LS bus via an interface of the Pro module.

If enabled, the counter is set to the start value and is started.

### Syntax

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

### Parameters

module	Specified module address (1...255).	LONG
channel	number (1, 2) of the LS bus interface on the Pro module.	LONG
ls-module	Specified module address on the LS bus (1...15).	LONG
enable	Set status of watchdog counter: 0 : Disable watchdog counter. 1 : Enable watchdog counter.	LONG
time	Release time (0...107374) of the counter in milliseconds.	LONG
ret_val	Bit pattern representing the error status. Bit = 0: No error. Bit = 1: Error occurred.	LONG

Bit no.	31...8	7	6	5...4	3	2	1	0
Status	–	Temp2	Temp1	–	WD	Time	Ovr	Par
- : don't care (mask with 0CFh)								
Par: Parity error during data transfer on the LS bus.								
Ovr: Overrun error during data transfer on the LS bus.								
Time: Timeout error during data transfer on the LS bus.								
WD: Watchdog was released. The channel drivers are deactivated.								
Temp1: Superheating on driver for channels 1...16. Driver is deactivated.								
Temp2: Superheating on driver for channels 17...32. Driver is deactivated.								

### Notes

The instruction only be used in section **INIT** : , since it takes long processing time.

As long as the watchdog counter is enabled, it decrements the counter value continuously. After the set release time the counter value reaches 0 (zero). If so, the module assumes a malfunction and stops; thus, all output signals are reset.

After power-up of the module the counter is set to the start value 10ms and the watchdog counter is enabled.

Reset the active watchdog timer at least once to the start value within the counting interval, in order to keep the module working. To reset the module use any module specific instruction or **LS\_WATCHDOG\_RESET**.

The watchdog function is used as to monitor the connection between ADwin system and LS bus module.

### To be used for the modules

LS-2 Rev. A





## See also

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

## Example

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**)