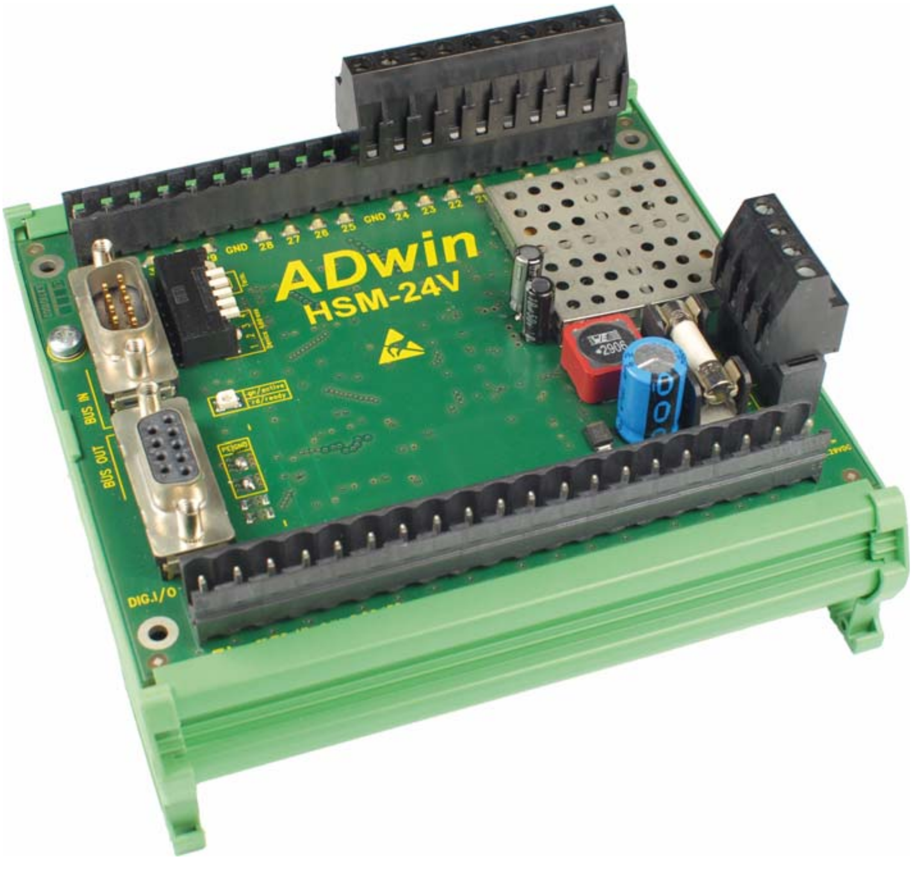
ADwin HSM-24V

Module for LS Bus

Manual



ADwin LS Bus, Manual version 1.5, March 2010

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

Hotline: +49 6251 96320
Fax: +49 6251 56819
E-Mail: info@ADwin.de
Internet www.ADwin.de

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

ADwin LS Bus, Manual version 1.5, March 2010

Table of Contents

Typographical Conventions

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

You find a "note" next to

- information, which absolutely have to be considered in order to guarantee an error free operation.
- advice for efficient operation.

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

`<C:\ADwin\ …>`

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

`Program text`

Program instructions and user inputs are characterized by the font `Courier New`.

`Var_1`

ADbasic source code elements such as instructions, variables, comments and other text are characterized by the font `Courier New` and are printed in color (see also the editor of the *ADbasic* development environment).

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:

For *ADwin* systems to function correctly, adhere strictly to the information provided in this documentation and in other mentioned manuals.

Qualified personnel

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 acivities, while recognizing and avoiding any possible dangers.*
> *(Definition of qualified personnel as per VDE 105 and ICE 364).*

Availability of the documents

This product documentation and all documents referred to, have always to be available and to be strictly 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.

Legal information

This documentation, including all pictures is protected by copyright. Reproduction, translation as well as electronical and photographical 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, the existence of which, may not be excluded.

Subject to change.

Hotline address: see inner side of cover page.

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.

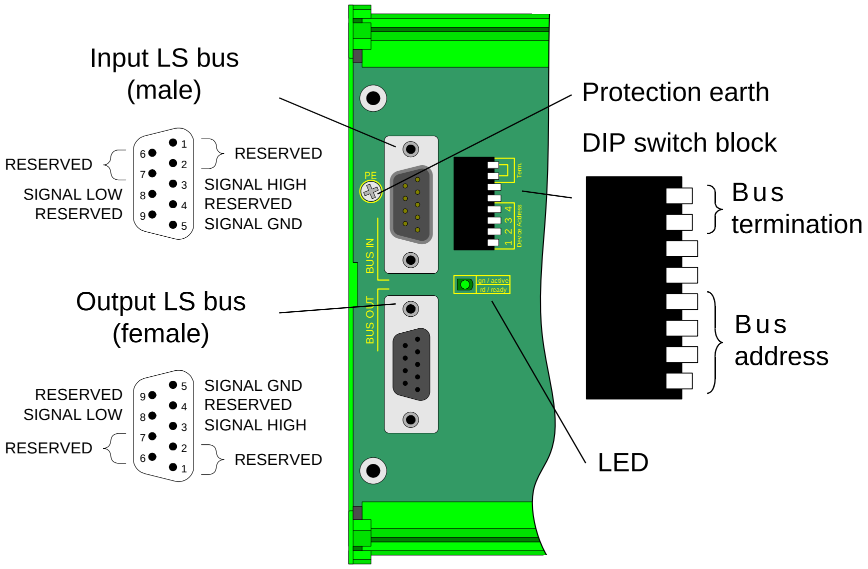


Fig. 1 – Standard connectors

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 `Term.`, 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. 2). 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 foloowing modules on the LS bus receive all bus data.

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

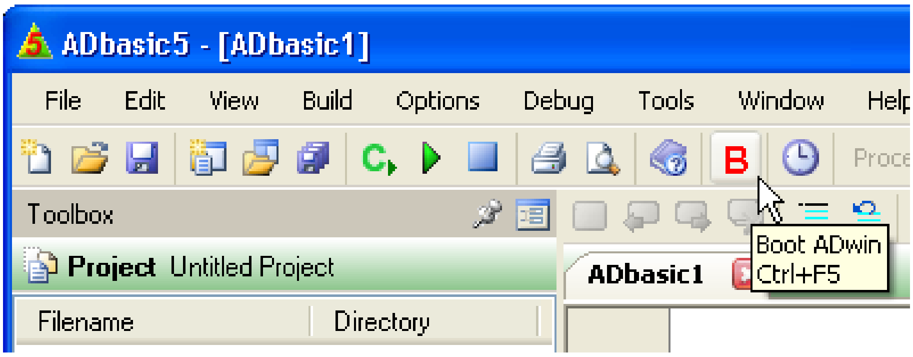
Fig. 2 – Module addressing with DIP switches

Protection Earth

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.

Booten

Starten Sie *ADbasic* und booten das *ADwin*-System durch Anklicken des Boot-Schaltfläche B.



3 HSM-24V

The module HSM-24V provides 32 digital channels which process 24V signals and is operated on the LS bus.

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 requires an external supply voltage of 19V… 29V. The supply connector plug is placed bottom right (see fig. 3) 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 superheating 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).

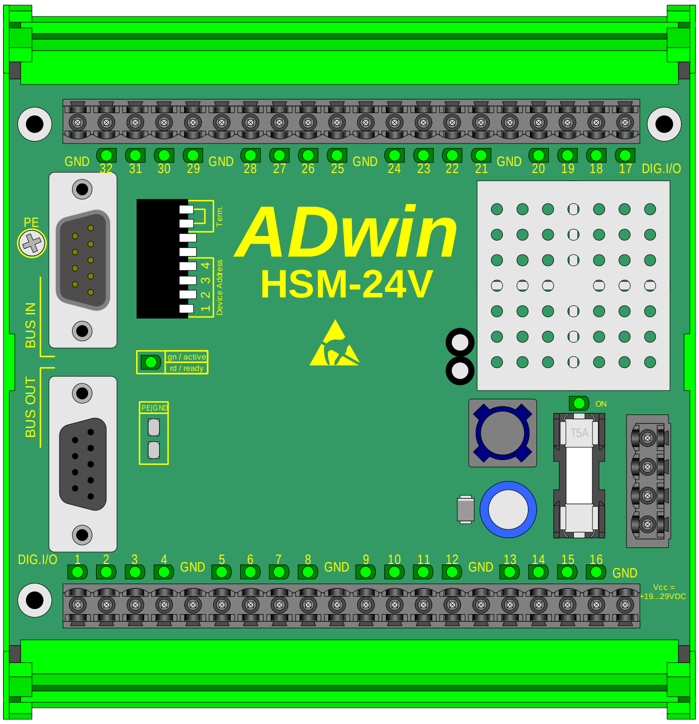
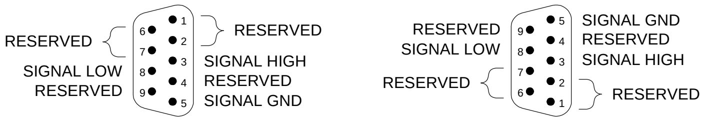


Fig. 3 – Board HSM-24V



Input LS bus (male) Output LS bus (female)

Fig. 4 – Pin assignment LS bus HSM-24V

The module is designed for operation in dry rooms with a room temperature of +5°C … +50°C and a relative humidity of 0 … 80% (no condensation).

3.2 RoHS Declaration of Conformity

The directive 2002/95/EG of the European Union on the restriction of the use of certain hazardous substances in electrical und electronic equipment (RoHS directive) has become operative as from 1st July, 2006.

The following substances are involved:

- Lead (Pb)
- Cadmium (Cd)
- Hexavalent chromium (Cr VI)
- Polybrominated biphenyls (PBB)
- Polybrominated diphenyl ethers (PBDE)
- Mercury (Hg)

The module HSM-24V complies with the requirements of the RoHS directive.

3.3 Software

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

An *ADwin* process which accesses the module HSM-24V should in any case run with low priority.
If otherweise the (relatively slow) access to the module runs with high priority, all other processes have to wait and their real-time qualities may be lost. Furthermore the data connection between PC and *ADwin* system may be cut.

The instructions have the following fuction:

Instruction	Function
`LS_DIO_INIT`	Initialize modul HSM-24V.
`LS_DIGPROG`	Set channels as inputs or outputs.
`LS_DIG_IO`	Set level of digital outputs and return current status. This instruction is valid for a single module only. No error handling.
`LS_DIGOUT_LONG`	Set level of digital outputs.
`LS_DIGIN_LONG`	Read current levels of digital inputs.
`LS_GET_OUTPUT_STATUS`	Read over-current status of digital outputs.
`LS_WATCHDOG_INIT`	Disable watchdog counter or enable and set watchdog time.
`LS_WATCHDOG_RESET`	Reset watchdog counters of all modules on LS bus to start values.

Fig. 5 – 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 7
ADwin-Gold II:	`#INCLUDE ADwinGoldII.INC`	page 21
ADwin-Pro:	`#INCLUDE ADwinPRO_ALL.INC`	page 37

With *ADwin-Gold II* all instructions may be used as well in *TiCoBasic* to access the HSM module. But, you have to use the include file `GoldIITiCo.inc` instead.

3.3.1 LS-Bus + ADwin-light-16

This section describes instructions which apply to LS bus modules connected to *ADwin-light-16*.

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.
	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.

The initialization does the following settings:

- All DIO channels are set as inputs.
  Other settings see **`LS_DIGPROG`**.
- The over-current status (> ca. 500 mA) 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-currrent in the range of 150…500 mA is present on several channels at the same time. Irrespective of this, an over-current of mor than 500 mA automatically switches off the concerned channel.

☞ The channels of the module HSM-24V may only be operated in the range of 0…150 mA. This ensures the module HSM-24V is working permanently without interruption even if all channels are used in parallel.

Valid for

HSM-24V + L16

See also

LS_DigProg, LS_Dig_IO, LS_Digout_Long, LS_Digin_Long, LS_Get_Output_Status, LS_Watchdog_Init, LS_Watchdog_Reset

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.	`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: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.

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).

Valid for

HSM-24V + L16

See also

LS_DIO_Init, LS_Dig_IO, LS_Digout_Long, LS_Digin_Long, LS_Get_Output_Status, LS_Watchdog_Init, LS_Watchdog_Reset

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). Bit = 0: Set outputs to level Low. Bit = 1: Set outputs to level High.	`LONG`

`ret_val`	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
Channel 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 12 µs 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.

Valid for

HSM-24V + L16

See also

LS_DIO_Init, LS_DigProg, LS_Dig_IO, LS_Digout_Long, LS_Digin_Long, LS_Get_Output_Status, LS_Watchdog_Init, LS_Watchdog_Reset

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_Digout_Long

`LS_DIGOUT_LONG` sets or clears all digital outputs of the specified module HSM-24V on the LS bus according to the transferred 32 bit value.

Syntax

```
#INCLUDE ADWL16.Inc
LS_DIGOUT_LONG(ls-module,pattern)
```

Parameters

`ls-module`	Specified module address on the LS bus (1…15).	`LONG`
`pattern`	Bit pattern, setting the digital outputs (see table). Bit = 0: Set outputs to level Low. Bit = 1: Set outputs to level High.	`LONG`

Bit No.	31	30	…	2	1	0
Channel no.	32	31	…	3	2	1

Notes

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.

Valid for

HSM-24V + L16

See also

LS_DIO_Init, LS_DigProg, LS_Dig_IO, LS_Digin_Long, LS_Get_Output_Status, LS_Watchdog_Init, LS_Watchdog_Reset

Example

```
REM Example process for ADwin-L16 and 2 modules HSM-24V
REM Set process to low priority!
#INCLUDE ADWL16.Inc

INIT:
 PROCESSDELAY = 4000000   '10Hz HP
 Par_1 = LS_DIO_INIT(1)   'LS module no. 1
 Par_2 = LS_DIGPROG(1, 01111b) 'channels 1…32 as output
 Par_3 = LS_WATCHDOG_INIT(1, 1, 1100) 'watchdog time 1.1 sec

 Par_11 = LS_DIO_INIT(3)  'LS module no. 3
 Par_12 = LS_DIGPROG(3, 0h) 'channels 1…32 as input
 Par_13 = LS_WATCHDOG_INIT(3, 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 of module 1
 LS_DIGOUT_LONG(1,Par_11)
 REM read channels of module 3
 Par_15 = LS_DIGIN_LONG(3)
 REM reset watchdog
 LS_WATCHDOG_RESET()
```

LS_Digin_Long

LS_DIGIN_LONG returns the status of all channels of the specified module HSM-24V on the LS bus as bit pattern.

Syntax

```
#INCLUDE ADWL16.Inc
ret_val = LS_DIGIN_LONG(ls-module)
```

Parameter

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

`ret_val`	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	…	2	1	0
Channel no.	32	31	…	3	2	1

Notes

We recommend to set the used channels as inputs with **LS_DIGPROG** before use.

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

Valid for

HSM-24V + L16

See also

LS_DIO_Init, LS_DigProg, LS_Dig_IO, LS_Digout_Long, LS_Get_Output_Status, LS_Watchdog_Init, LS_Watchdog_Reset

Example

```
REM Example process for ADwin-L16 and 2 modules HSM-24V
REM Set process to low priority!
#INCLUDE ADWL16.Inc

INIT:
 PROCESSDELAY = 4000000   '10Hz HP
 Par_1 = LS_DIO_INIT(1)   'LS module no. 1
 Par_2 = LS_DIGPROG(1, 01111b) 'channels 1…32 as output
 Par_3 = LS_WATCHDOG_INIT(1, 1, 1100) 'watchdog time 1.1 sec

 Par_11 = LS_DIO_INIT(3)  'LS module no. 3
 Par_12 = LS_DIGPROG(3, 0h) 'channels 1…32 as input
 Par_13 = LS_WATCHDOG_INIT(3, 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 of module 1
 LS_DIGOUT_LONG(1,Par_11)
 REM read channels of module 3
 Par_15 = LS_DIGIN_LONG(3)
 REM reset watchdog
 LS_WATCHDOG_RESET()
```

LS_Get_Output_Status

`LS_GET_OUTPUT_STATUS` returns the over-current status of outputs of the specified module HSM-24V on the LS bus as bit pattern.

Syntax

```
#INCLUDE ADWL16.Inc
ret_val = LS_GET_OUTPUT_STATUS(ls-module)
```

Parameters

`ls-module`	Specified module address on the LS bus (1…15).	`LONG`
`ret_val`	Bit pattern. Each bit (31…0) represents the over-current status of a digital output (see table). Bit = 0: Standard status. Bit = 1: Over-current occurred, output disabled.	`LONG`

Bit no.	31	30	…	2	1	0
Channel no.	32	31	…	3	2	1

Notes

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

After a "superheating" errorthe module is reset with **`LS_DIO_INIT`**.

☞ The channels of the module HSM-24V may only be operated in the range of 0…150 mA. This ensures the module HSM-24V is working permanently without interruption even if all channels are used in parallel.

Valid for

HSM-24V + L16

See also

LS_DIO_Init, LS_DigProg, LS_Dig_IO, LS_Digout_Long, LS_Digin_Long, LS_Watchdog_Init, LS_Watchdog_Reset

Example

```
REM Example process for ADwin-L16 and 2 modules HSM-24V
REM Set process to low priority!
#INCLUDE ADWL16.Inc

INIT:
 PROCESSDELAY = 4000000   '10Hz HP
 Par_1 = LS_DIO_INIT(1)   'LS module no. 1
 Par_2 = LS_DIGPROG(1, 01111b) 'channels 1…32 as output
 Par_3 = LS_WATCHDOG_INIT(1, 1, 1100) 'watchdog time 1.1 sec

 Par_11 = LS_DIO_INIT(3)  'LS module no. 3
 Par_12 = LS_DIGPROG(3, 0h) 'channels 1…32 as input
 Par_13 = LS_WATCHDOG_INIT(3, 1, 1100) 'watchdog time 1.1 sec

EVENT:
 REM check for over-current
 Par_5 = LS_GET_OUTPUT_STATUS(1) + LS_GET_OUTPUT_STATUS(3)
 IF (Par_5 > 0) THEN END  'over-current: exit program

 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 of module 1
 LS_DIGOUT_LONG(1,Par_11)
 REM read channels of module 3
 Par_15 = LS_DIGIN_LONG(3)
 REM reset watchdog
 LS_WATCHDOG_RESET()
```

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.

Valid for

HSM-24V + L16

See also

LS_DIO_Init, LS_DigProg, LS_Watchdog_Reset, LS_Dig_IO, LS_Digout_Long, LS_Digin_Long, LS_Get_Output_Status, LS_Watchdog_Reset

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_Reset

LS_WATCHDOG_RESET resets the watchdog counters of all modules on the LS bus to the appropriate start value. The counters remain enabled.

Syntax

```
#INCLUDE ADWL16.Inc
LS_WATCHDOG_RESET()
```

Parameters

- / -

Notes

As long as a 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.

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 you may also use any module specific instruction.

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

Valid for

HSM-24V + L16

See also

LS_DIO_Init, LS_DigProg, LS_Dig_IO, LS_Digout_Long, LS_Digin_Long, LS_Get_Output_Status, LS_Watchdog_Init

Example

```
REM Example process for ADwin-L16 and 2 modules HSM-24V
REM Set process to low priority!
#INCLUDE ADWL16.Inc

INIT:
 PROCESSDELAY = 4000000   '10Hz HP
 Par_1 = LS_DIO_INIT(1)   'LS module no. 1
 Par_2 = LS_DIGPROG(1, 01111b) 'channels 1…32 as output
 Par_3 = LS_WATCHDOG_INIT(1, 1, 1100) 'watchdog time 1.1 sec

 Par_11 = LS_DIO_INIT(3)  'LS module no. 3
 Par_12 = LS_DIGPROG(3, 0h) 'channels 1…32 as input
 Par_13 = LS_WATCHDOG_INIT(3, 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 of module 1
 LS_DIGOUT_LONG(1,Par_11)
 REM read channels of module 3
 Par_15 = LS_DIGIN_LONG(3)
 REM reset watchdog
 LS_WATCHDOG_RESET()
```

3.3.2 LS-Bus + *ADwin-Gold II*

This section describes instructions which apply to LS bus modules connected to *ADwin-Gold II*.

LS_DIO_Init

T11 TiCo

`LS_DIO_INIT` initializes the specified module of type HSM-24V on the LS bus and returns the error status.

Syntax

```
#INCLUDE ADwinGoldII.inc / GoldIITiCo.inc
ret_val = LS_DIO_INIT(channel,ls-module)
```

Parameters

`channel`	Number (1, 2) of the LS-Bus interface.	`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 (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. 500 mA) 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-currrent in the range of 150…500 mA is present on several channels at the same time. Irrespective of this, an over-current of mor than 500 mA automatically switches off the concerned channel.

The channels of the module HSM-24V may only be operated in the range of 0…150 mA. This ensures the module HSM-24V is working permanently without interruption even if all channels are used in parallel.

Valid for

HSM-24V + Gold II

See also

LS_DigProg, LS_Dig_IO, LS_Digout_Long, LS_Digin_Long, LS_Get_Output_Status, LS_Watchdog_Init, LS_Watchdog_Reset

Example

```
REM Example process for one module HSM-24V and ADwin-Gold II
Rem Please select the appropriate include for ADbasic / TiCoBasic
#INCLUDE ADwinGoldII.inc / GoldIITiCo.inc

INIT:
 PROCESSDELAY = 4000000   '10Hz HP
 Par_1 = LS_DIO_INIT(1,1) AND 0CFh
 Par_2 = LS_DIGPROG(1,1,0Fh) AND 0CFh 'channels 1…32 as output
 Par_3 = LS_WATCHDOG_INIT(1,1,1,1100) AND 0CFh 'watchdog 1.1 s

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,Par_11)
```

LS_DigProg

T11 TiCo

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 ADwinGoldII.inc / GoldIITiCo.inc
ret_val = LS_DIGPROG(channel,ls-module,pattern)
```

Parameters

`channel`	Number (1, 2) of the LS-Bus interface.	`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: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.

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).

Valid for

HSM-24V + Gold II

See also

LS_DIO_Init, LS_Dig_IO, LS_Digout_Long, LS_Digin_Long, LS_Get_Output_Status, LS_Watchdog_Init, LS_Watchdog_Reset

Example

```
REM Example process for one module HSM-24V and ADwin-Gold II
Rem Please select the appropriate include for ADbasic / TiCoBasic
#INCLUDE ADwinGoldII.inc / GoldIITiCo.inc

INIT:
 PROCESSDELAY = 4000000   '10Hz HP
 Par_1 = LS_DIO_INIT(1,1) AND 0CFh
 Par_2 = LS_DIGPROG(1,1,0Fh) AND 0CFh 'channels 1…32 as output
 Par_3 = LS_WATCHDOG_INIT(1,1,1,1100) AND 0CFh 'watchdog 1.1 s

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,Par_11)
```

LS_Dig_IO

T11 TiCo

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 ADwinGoldII.inc / GoldIITiCo.inc
ret_val = LS_DIG_IO(channel, pattern)
```

Parameters

`channel`	Number (1, 2) of the LS-Bus interface.	LONG
`pattern`	Bit pattern, setting the digital outputs (see table). Bit = 0: Set outputs to level Low. Bit = 1: Set outputs to level High.	LONG

`ret_val`	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
Channel 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 12 µs 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.

Valid for

HSM-24V + Gold II

See also

LS_DIO_Init, LS_DigProg, LS_Digout_Long, LS_Digin_Long, LS_Get_Output_Status, LS_Watchdog_Init, LS_Watchdog_Reset

Example

```
REM Example process for one module HSM-24V and ADwin-Gold II
Rem Please select the appropriate include for ADbasic / TiCoBasic
#INCLUDE ADwinGoldII.inc / GoldIITiCo.inc

INIT:
 PROCESSDELAY = 4000000   '10Hz HP
 Par_1 = LS_DIO_INIT(1,1) AND 0CFh
 Par_2 = LS_DIGPROG(1,1,0Fh) AND 0CFh 'channels 1…32 as output
 Par_3 = LS_WATCHDOG_INIT(1,1,1,1100) AND 0CFh 'watchdog 1.1 s

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,Par_11)
```

LS_Digout_Long

T11 TiCo

LS_DIGOUT_LONG sets or clears all digital outputs of the specified module HSM-24V on the LS bus according to the transferred 32 bit value.

Syntax

```
#INCLUDE ADwinGoldII.inc / GoldIITiCo.inc
LS_DIGOUT_LONG(channel, ls-module, pattern)
```

Parameters

`channel`	Number (1, 2) of the LS-Bus interface.	LONG
`ls-module`	Specified module address on the LS bus (1…15).	LONG
`pattern`	Bit pattern, setting the digital outputs (see table). Bit = 0: Set outputs to level Low. Bit = 1: Set outputs to level High.	LONG

Bit No.	31	30	…	2	1	0
Channel no.	32	31	…	3	2	1

Notes

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.

Valid for

HSM-24V + Gold II

See also

LS_DIO_Init, LS_DigProg, LS_Dig_IO, , LS_Digin_Long, LS_Get_Output_Status, LS_Watchdog_Init, LS_Watchdog_Reset

Example

```
REM Example process for ADwin-Gold II and 2 modules HSM-24V
REM Set process to low priority!
Rem Please select the appropriate include for ADbasic / TiCoBasic
#INCLUDE ADwinGoldII.inc / GoldIITiCo.inc

INIT:
 PROCESSDELAY = 4000000   '10Hz HP
 Par_1 = LS_DIO_INIT(1,1) AND 0CFh
 Par_2 = LS_DIGPROG(1,1,0Fh) AND 0CFh 'channels 1…32 as output
 Par_3 = LS_WATCHDOG_INIT(1,1,1,1100) AND 0CFh 'watchdog 1.1 s

 Par_11 = LS_DIO_INIT(1,3)'LS module no. 3
 Par_12 = LS_DIGPROG(1,3,0h) 'channels 1…32 as input
 Par_13 = LS_WATCHDOG_INIT(1,3,1,1100) '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 of module 1
 LS_DIGOUT_LONG(1,1,Par_11)
 REM read channels of module 3
 Par_15 = LS_DIGIN_LONG(1,3)
 REM reset watchdog
 LS_WATCHDOG_RESET(1)
```

LS_Digin_Long

T11 TiCo

`LS_DIGIN_LONG` returns the status of all channels of the specified module HSM-24V on the LS bus as bit pattern.

Syntax

```
#INCLUDE ADwinGoldII.inc / GoldIITiCo.inc
ret_val = LS_DIGIN_LONG(module, channel, ls-module)
```

Parameters

`channel`	Number (1, 2) of the LS-Bus interface.	`LONG`
`ls-module`	Specified module address on the LS bus (1…15).	`LONG`

`ret_val`	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	…	2	1	0
Channel no.	32	31	…	3	2	1

Notes

We recommend to set the used channels as inputs with **`LS_DIGPROG`** before use.

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

Valid for

HSM-24V + Gold II

See also

LS_DIO_Init, LS_DigProg, LS_Dig_IO, LS_Digout_Long, LS_Get_Output_Status, LS_Watchdog_Init, LS_Watchdog_Reset

Example

```
REM Example process for ADwin-Gold II and 2 modules HSM-24V
REM Set process to low priority!
Rem Please select the appropriate include for ADbasic / TiCoBasic
#INCLUDE ADwinGoldII.inc / GoldIITiCo.inc

INIT:
 PROCESSDELAY = 4000000   '10Hz HP
 Par_1 = LS_DIO_INIT(1,1) AND 0CFh 'LS module no. 1
 Par_2 = LS_DIGPROG(1,1,0Fh) AND 0CFh 'channels 1…32 as output
 Par_3 = LS_WATCHDOG_INIT(1,1,1,1100) AND 0CFh 'watchdog 1.1 s

 Par_11 = LS_DIO_INIT(1,3) AND 0CFh 'LS module no. 1
 Par_12 = LS_DIGPROG(1,3,0h) AND 0CFh 'channels 1…32 as input
 Par_13 = LS_WATCHDOG_INIT(1,3,1,1100) AND 0CFh 'watchdog 1.1 s

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 of module 1
 LS_DIGOUT_LONG(1,1,Par_11)
 REM read channels of module 3
 Par_15 = LS_DIGIN_LONG(1,3)
 REM reset watchdog
 LS_WATCHDOG_RESET(1)
```

LS_Get_Output_Status

T11 TiCo

LS_GET_OUTPUT_STATUS returns the over-current status of outputs of the specified module HSM-24V on the LS bus as bit pattern.

Syntax

```
#INCLUDE ADwinGoldII.inc / GoldIITiCo.inc
ret_val = LS_GET_OUTPUT_STATUS(channel, ls-module)
```

Parameters

`channel`	Number (1, 2) of the LS-Bus interface.	`LONG`
`ls-module`	Specified module address on the LS bus (1…15).	`LONG`
`ret_val`	Bit pattern. Each bit (31…0) represents the over-current status of a digital output (see table). Bit = 0: Standard status. Bit = 1: Over-current occurred, output disabled.	`LONG`

Bit no.	31	30	…	2	1	0
Channel no.	32	31	…	3	2	1

Notes

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

After a "superheating" errorthe module is reset with **LS_DIO_INIT**.

The channels of the module HSM-24V may only be operated in the range of 0…150 mA. This ensures the module HSM-24V is working permanently without interruption even if all channels are used in parallel.

Valid for

HSM-24V + Gold II

See also

LS_DIO_Init, LS_DigProg, LS_Dig_IO, LS_Digout_Long, LS_Digin_Long, LS_Watchdog_Init, LS_Watchdog_Reset

Example

```
REM Example process for ADwin-Gold II and 2 modules HSM-24V
REM Set process to low priority!
Rem Please select the appropriate include for ADbasic / TiCoBasic
#INCLUDE ADwinGoldII.inc / GoldIITiCo.inc

INIT:
 PROCESSDELAY = 4000000   '10Hz HP
 Par_1 = LS_DIO_INIT(1,1) AND 0CFh 'LS module no. 1
 Par_2 = LS_DIGPROG(1,1,0Fh) AND 0CFh 'channels 1…32 as output
 Par_3 = LS_WATCHDOG_INIT(1,1,1,1100) AND 0CFh 'watchdog 1.1 s

 Par_11 = LS_DIO_INIT(1,3) AND 0CFh 'LS module no. 1
 Par_12 = LS_DIGPROG(1,3,0h) AND 0CFh 'channels 1…32 as input
 Par_13 = LS_WATCHDOG_INIT(1,3,1,1100) AND 0CFh 'watchdog 1.1 s

EVENT:
 REM check for over-current
 Par_5 = LS_GET_OUTPUT_STATUS(1,1) + LS_GET_OUTPUT_STATUS(1,3)
 IF (Par_5>0) THEN END    'over-current: exit program

 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 of module 1
 LS_DIGOUT_LONG(1,1,Par_11)
 REM read channels of module 3
 Par_15 = LS_DIGIN_LONG(1,3)
 REM reset watchdog
 LS_WATCHDOG_RESET(1)
```

LS_Watchdog_Init

T11 TiCo

`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 ADwinGoldII.inc / GoldIITiCo.inc
ret_val = LS_WATCHDOG_INIT(channel,ls-module,enable,
    time)
```

Parameters

`channel`	Number (1, 2) of the LS-Bus interface.	`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.

Valid for

HSM-24V + Gold II

See also

LS_DIO_Init, LS_DigProg, LS_Dig_IO, LS_Digout_Long, LS_Digin_Long, LS_Get_Output_Status, LS_Watchdog_Reset

Example

```
REM Example process for one module HSM-24V and ADwin-Gold II
Rem Please select the appropriate include for ADbasic / TiCoBasic
#INCLUDE ADwinGoldII.inc / GoldIITiCo.inc

INIT:
 PROCESSDELAY = 4000000   '10Hz HP
 Par_1 = LS_DIO_INIT(1,1) AND 0CFh 'LS module no. 1
 Par_2 = LS_DIGPROG(1,1,0Fh) AND 0CFh 'channels 1…32 as output
 Par_3 = LS_WATCHDOG_INIT(1,1,1,1100) AND 0CFh 'watchdog 1.1 s

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,Par_11)
```

LS_Watchdog_Reset

T11 TiCo

`LS_WATCHDOG_RESET` resets the watchdog counters of all modules on the LS bus to the appropriate start value. The counters remain enabled.

Syntax

```
#INCLUDE ADwinGoldII.inc / GoldIITiCo.inc
LS_WATCHDOG_RESET(channel)
```

Parameters

`channel`	Number (1, 2) of the LS-Bus interface.	`LONG`

Notes

As long as a 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.

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 you may also use any module specific instruction.

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

Valid for

HSM-24V + Gold II

See also

LS_DIO_Init, LS_DigProg, LS_Dig_IO, LS_Digout_Long, LS_Digin_Long, LS_Get_Output_Status, LS_Watchdog_Init

Example

```
REM Example process for ADwin-Gold II and 2 modules HSM-24V
REM Set process to low priority!
Rem Please select the appropriate include for ADbasic / TiCoBasic
#INCLUDE ADwinGoldII.inc / GoldIITiCo.inc

INIT:
 PROCESSDELAY = 4000000   '10Hz HP
 Par_1 = LS_DIO_INIT(1,1) AND 0CFh 'LS module no. 1
 Par_2 = LS_DIGPROG(1,1,0Fh) AND 0CFh 'channels 1…32 as output
 Par_3 = LS_WATCHDOG_INIT(1,1,1,1100) AND 0CFh 'watchdog 1.1 s

 Par_11 = LS_DIO_INIT(1,3)'LS module no. 3
 Par_12 = LS_DIGPROG(1,1,0h) 'channels 1…32 as input
 Par_13 = LS_WATCHDOG_INIT(1,1,1,1100) '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 of module 1
 LS_DIGOUT_LONG(1,1,Par_11)
 REM read channels of module 3
 Par_15 = LS_DIGIN_LONG(1,3)
 REM reset watchdog
 LS_WATCHDOG_RESET(1)
```

3.3.3 LS-Bus + Pro I

This section describes instructions which apply to Pro I LS bus modules:

- LS_DIO_Init (page 38)
- LS_DigProg (page 40)
- LS_Dig_IO (page 42)
- LS_Digout_Long (page 44)
- LS_Digin_Long (page 46)
- LS_Get_Output_Status (page 48)
- LS_Watchdog_Init (page 50)
- LS_Watchdog_Reset (page 52)

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	–	Temp 2	Temp 1	–	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. 500 mA) 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-currrent in the range of 150…500 mA is present on several channels at the same time. Irrespective of this, an over-current of mor than 500 mA automatically switches off the concerned channel.

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

Valid for

LS-2 Rev. A

See also

LS_DigProg, LS_Dig_IO, LS_Digout_Long, LS_Digin_Long, LS_Get_Output_Status, LS_Watchdog_Init, LS_Watchdog_Reset

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	–	Temp 2	Temp 1	–	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.

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).

Valid for

LS-2 Rev. A

See also

LS_DIO_Init, LS_Dig_IO, LS_Digout_Long, LS_Digin_Long, LS_Get_Output_Status, LS_Watchdog_Init, LS_Watchdog_Reset

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

`module`	Specified module address (1…255).	LONG
`channel`	number (1, 2) of the LS bus interface on the Pro module.	LONG
`pattern`	Bit pattern, setting the digital outputs (see table). Bit = 0: Set outputs to level Low. Bit = 1: Set outputs to level High.	LONG

`ret_val`	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
Channel 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 12 µs 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.

Valid for

LS-2 Rev. A

See also

LS_DIO_Init, LS_DigProg, LS_Digout_Long, LS_Digin_Long, LS_Get_Output_Status, LS_Watchdog_Init, LS_Watchdog_Reset

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 check for over-current
 PAR_5 = LS_GET_OUTPUT_STATUS(3,1,2)
 PAR_5 = PAR_5 + LS_GET_OUTPUT_STATUS(3,1,4)
 IF (PAR_5>0) THEN END    'over-current: Exit program

 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 of LS module 2
 LS_DIGOUT_LONG(3,1,2,PAR_11)
 Rem read channels of LS module 4
 PAR_15 = LS_DIGIN_LONG(3,1,4)
 Rem reset watchdog
 LS_WATCHDOG_RESET(3,1)
```

LS_Digout_Long

`LS_DIGOUT_LONG` sets or clears all digital outputs of the specified module HSM-24V on the LS bus according to the transferred 32 bit value.

Syntax

```
#INCLUDE ADwinPro_All.Inc
LS_DIGOUT_LONG(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 digital outputs (see table). Bit = 0: Set outputs to level Low. Bit = 1: Set outputs to level High.	`LONG`

Bit No.	31	30	…	2	1	0
Channel no.	32	31	…	3	2	1

Notes

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.

Valid for

LS-2 Rev. A

See also

LS_DIO_Init, LS_DigProg, LS_Dig_IO, LS_Digin_Long, LS_Get_Output_Status, LS_Watchdog_Init, LS_Watchdog_Reset

Example

```
REM Example process for ADwin-Pro and 2 modules HSM-24V
REM Set process to low priority!
#INCLUDE ADwinPro_All.Inc

INIT:
 PROCESSDELAY = 4000000   '10Hz HP
 REM Settings for LS module 2 via Pro module 3, channel 1
 Par_1 = LS_DIO_INIT(3,1,2)
 Par_2 = LS_DIGPROG(3,1,2,01111b) 'channels 1…32 as output
 Par_3 = LS_WATCHDOG_INIT(3,1,2, 1, 1100) 'watchdog time 1.1 sec

 REM Settings for LS module 4 via Pro module 3, channel 1
 Par_11 = LS_DIO_INIT(3,1,4)
 Par_12 = LS_DIGPROG(3,1,4, 0h) 'channels 1…32 as input
 Par_13 = LS_WATCHDOG_INIT(3,1,4, 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 of LS module 2
 LS_DIGOUT_LONG(3,1,2,Par_11)
 REM read channels of LS module 4
 Par_15 = LS_DIGIN_LONG(3,1,4)
 REM reset watchdog
 LS_WATCHDOG_RESET(3,1)
```

LS_Digin_Long

`LS_DIGIN_LONG` returns the status of all channels of the specified module HSM-24V on the LS bus as bit pattern.

Syntax

```
#INCLUDE ADwinPro_All.Inc
ret_val = LS_DIGIN_LONG(module,channel,ls-module)
```

Parameter

`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 real state of all digital channels (see table).
	Bit = 0: Channel has level Low.
	Bit = 1: Channel has level High.
		`LONG`

Bitnr.	31	30	…	2	1	0
Channel no.	32	31	…	3	2	1

Notes

We recommend to set the used channels as inputs with **`LS_DIGPROG`** before use.

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

Valid for

LS-2 Rev. A

See also

LS_DIO_Init, LS_DigProg, LS_Dig_IO, LS_Digout_Long, LS_Get_Output_Status, LS_Watchdog_Init, LS_Watchdog_Reset

Example

```
REM Example process for ADwin-Pro and 2 modules HSM-24V
REM Set process to low priority!
#INCLUDE ADwinPro_All.Inc

INIT:
 PROCESSDELAY = 4000000   '10Hz HP
 REM Settings for LS module 2 via Pro module 3, channel 1
 Par_1 = LS_DIO_INIT(3,1,2)
 Par_2 = LS_DIGPROG(3,1,2,01111b) 'channels 1…32 as output
 Par_3 = LS_WATCHDOG_INIT(3,1,2, 1, 1100) 'watchdog time 1.1 sec

 REM Settings for LS module 4 via Pro module 3, channel 1
 Par_11 = LS_DIO_INIT(3,1,4)
 Par_12 = LS_DIGPROG(3,1,4, 0h) 'channels 1…32 as input
 Par_13 = LS_WATCHDOG_INIT(3,1,4, 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 of LS module 2
 LS_DIGOUT_LONG(3,1,2,Par_11)
 REM read channels of LS module 4
 Par_15 = LS_DIGIN_LONG(3,1,4)
 REM reset watchdog
 LS_WATCHDOG_RESET(3,1)
```

LS_Get_Output_Status

LS_GET_OUTPUT_STATUS returns the over-current status of outputs of the specified module HSM-24V on the LS bus as bit pattern.

Syntax

```
#INCLUDE ADwinPro_All.Inc
ret_val = LS_GET_OUTPUT_STATUS(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. Each bit (31…0) represents the over-current status of a digital output (see table). Bit = 0: Standard status. Bit = 1: Over-current occurred, output disabled.	`LONG`

Bit no.	31	30	…	2	1	0
Channel no.	32	31	…	3	2	1

Notes

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

After a "superheating" errorthe module is reset with **LS_DIO_INIT**.

The channels of the module HSM-24V may only be operated in the range of 0…150 mA. This ensures the module HSM-24V is working permanently without interruption even if all channels are used in parallel.

Valid for

LS-2 Rev. A

See also

LS_DIO_Init, LS_DigProg, LS_Dig_IO, LS_Digout_Long, LS_Digin_Long, LS_Watchdog_Init, LS_Watchdog_Reset

Example

```
REM Example process for ADwin-Pro and 2 modules HSM-24V
REM Set process to low priority!
#INCLUDE ADwinPro_All.Inc

INIT:
 PROCESSDELAY = 4000000   '10Hz HP
 REM Settings for LS module 2 via Pro module 3, channel 1
 Par_1 = LS_DIO_INIT(3,1,2)
 Par_2 = LS_DIGPROG(3,1,2,01111b) 'channels 1…32 as output
 Par_3 = LS_WATCHDOG_INIT(3,1,2, 1, 1100) 'watchdog time 1.1 sec

 REM Settings for LS module 4 via Pro module 3, channel 1
 Par_11 = LS_DIO_INIT(3,1,4)
 Par_12 = LS_DIGPROG(3,1,4, 0h) 'channels 1…32 as input
 Par_13 = LS_WATCHDOG_INIT(3,1,4, 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 of LS module 2
 LS_DIGOUT_LONG(3,1,2,Par_11)
 REM read channels of LS module 4
 Par_15 = LS_DIGIN_LONG(3,1,4)
 REM reset watchdog
 LS_WATCHDOG_RESET(3,1)
```

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	–	Temp 2	Temp 1	–	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.

Valid for

LS-2 Rev. A

See also

LS_DIO_Init, LS_DigProg, LS_Dig_IO, LS_Digout_Long, LS_Digin_Long, LS_Get_Output_Status, LS_Watchdog_Reset

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_Reset

LS_WATCHDOG_RESET resets the watchdog counters of all modules on the LS bus to the appropriate start value. The counters remain enabled.

Syntax

```
#INCLUDE ADwinPro_All.Inc
LS_WATCHDOG_RESET(module, channel)
```

Parameters

`module`	Specified module address (1…255).	LONG
`channel`	number (1, 2) of the LS bus interface on the Pro module.	LONG

Notes

As long as a 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.

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 you may also use any module specific instruction.

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

Valid for

LS-2 Rev. A

See also

LS_DIO_Init, LS_DigProg, LS_Dig_IO, LS_Digout_Long, LS_Digin_Long, LS_Get_Output_Status, LS_Watchdog_Init

Example

```
REM Example process for ADwin-Pro and 2 modules HSM-24V
REM Set process to low priority!
#INCLUDE ADwinPro_All.Inc

INIT:
 PROCESSDELAY = 4000000   '10Hz HP
 REM Settings for LS module 2 via Pro module 3, channel 1
 Par_1 = LS_DIO_INIT(3,1,2)
 Par_2 = LS_DIGPROG(3,1,2,01111b) 'channels 1…32 as output
 Par_3 = LS_WATCHDOG_INIT(3,1,2, 1, 1100) 'watchdog time 1.1 sec

 REM Settings for LS module 4 via Pro module 3, channel 1
 Par_11 = LS_DIO_INIT(3,1,4)
 Par_12 = LS_DIGPROG(3,1,4, 0h) 'channels 1…32 as input
 Par_13 = LS_WATCHDOG_INIT(3,1,4, 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 of LS module 2
 LS_DIGOUT_LONG(3,1,2,Par_11)
 REM read channels of LS module 4
 Par_15 = LS_DIGIN_LONG(3,1,4)
 REM reset watchdog
 LS_WATCHDOG_RESET(3,1)
```