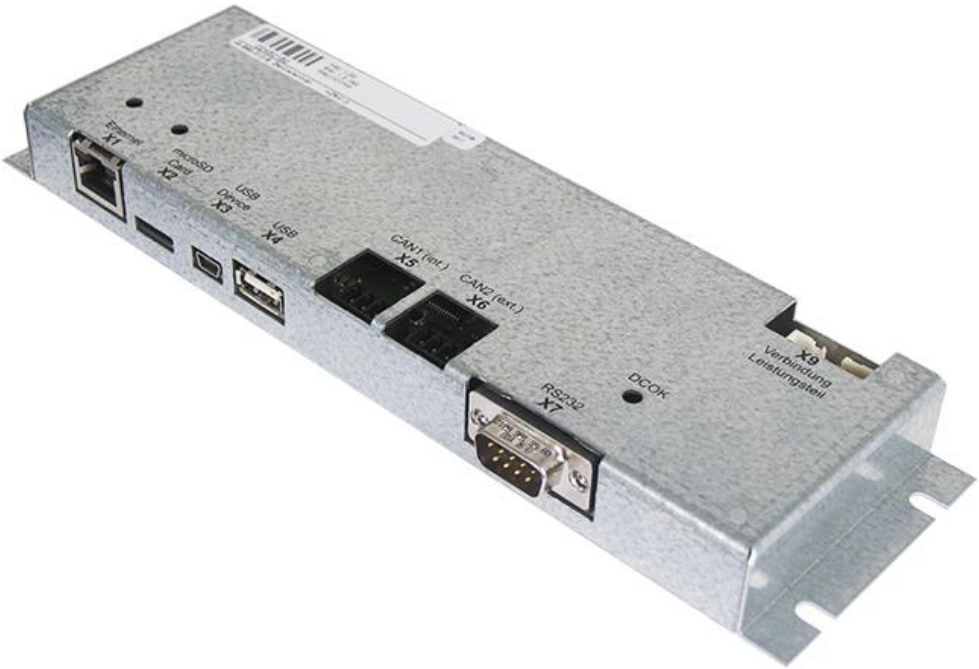
Central Unit with EDGE CPU HZS 512

The HZS 512 central unit runs the control program and is thereby an essential component of the heating control. This module is a simple component that is used to control automated processes in a heating system. The HZS 512 central unit communicates with the boiler function module over a CAN bus connection.

A micro SD card serves as the storage medium for the operating system, application and application data. For program updates, the integrated USB interface can be used.



Symbolfoto

Technical Data

Processor unit performance data

Processor	EDGE-Technology X86 compatible
Internal cache	32-kbyte L1 Cache 256-kbyte L2 Cache
Processor clock frequency	500 MHz
BIOS	AMI
Internal program and data memory (DDR2 RAM)	64 Mbytes
Internal remnant data memory	512 Kbytes (1)
Internal storage device (IDE)	micro SD card (1 GB)
Interfaces	1 x Ethernet (RJ45) 1 x USB 1.1, Type Mini B (online USB) 1 x USB 2.0, Type A (full speed 12 Mbit/s) 1 x CAN-Bus 1 (internal) 1 x CAN-Bus 2 (external) 1 x RS232 (online/Modem)
Data buffer	Yes
Signal generator	No
Real-time clock	Yes (10-day buffering via GoldCap)
Cooling	Passive (fanless)

(1) See chapter “Note on SRAM Behavior”

Electrical requirements

Power supply +24V	Minimum +18 V DC	Maximum +30 V DC
Current consumption of voltage supply (for internal electronics)	Typically 120 mA at + 24 V	Maximum 325 mA at 18 V Maximum 275 mA at 24 V DC Maximum 225 mA at 30 V DC
Current consumption of voltage supply +24 V (for external display units)	Maximum 1.4 A	

Mechanical

Material	1.0 mm galvanized sheet steel
Mechanical Dimensions	230 mm x 67 mm x 21.8 mm (L x W x H)

Environmental conditions

Storage temperature	-20 -+70 °C	
Operating temperature	0 -+60 °C	
Humidity	0 - 95 %, non-condensing	
EMC stability	according to EN 61000-6-2 (industrial area)	
EMC - noise generation	According to EN 61000-6-3 (living area)	
Shock resistance	EN 60068-2-27	150 m/s²

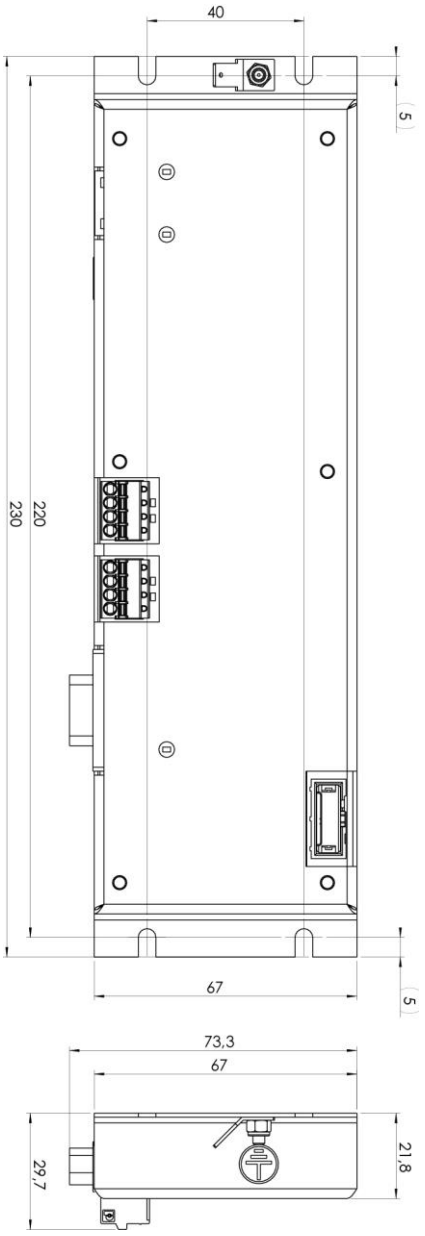
Miscellaneous

Article number	05-895-512
HW Version	1.x

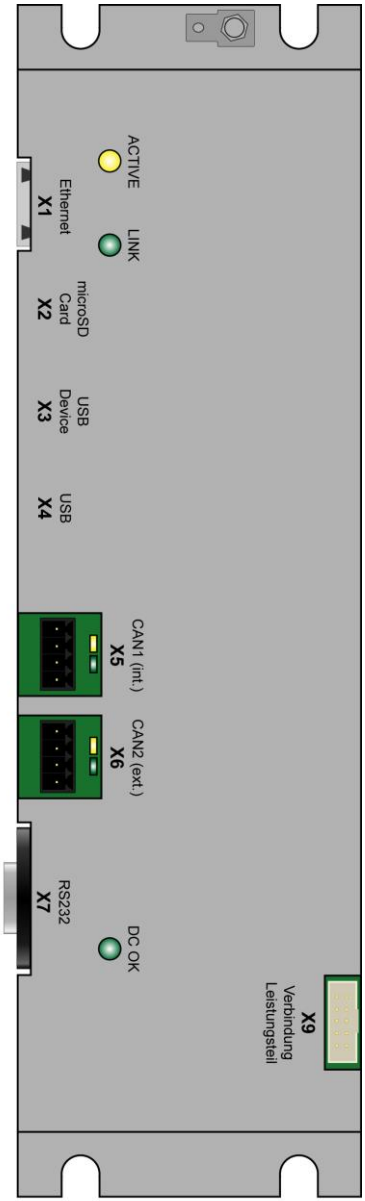
Terminal requirements

Connection technology	Connector terminals are not included in delivery! The following sprint terminals are required: 2 x 4-pin FK-MCP 1.5/4-ST-3.5 Phoenix Contact spring terminal connector RM 3.5

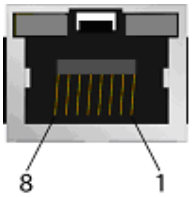
Mechanical Dimensions



Connector Layout



X1: Ethernet

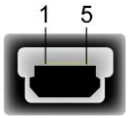


PIN	Function
1	TX +
2	TX -
3	RX +
4	-
5	-
6	RX -
7	-
8	-

Problems can arise if a control is connected to an IP network, which contains modules that are not running with a SIGMATEK operating system. With such devices, Ethernet packets could be sent to the control with such a high frequency (e.g. broadcasts), that the high interrupt load could cause a real-time runtime error or runtime error. By configuring the packet filter (Firewall or Router) accordingly however, it is possible to connect a network with SIGMATEK hardware to a third party network without triggering the error mentioned above.

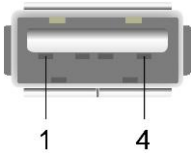
X2: micro SD Card

X3: USB 1.1 (Type Mini B)



PIN	Function
1	+5 V
2	D-
3	D+
4	-
5	GND

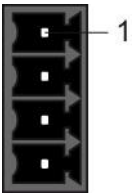
X4: USB 2.0 (Type A full speed 12 Mbit/s)



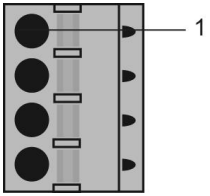
PIN	Function
1	+5 V_USB
2	D-
3	D+
4	GND

It should be noted that many of the USB devices on the market do not comply with USB specifications; this can lead to device malfunctions. It is also possible that these devices will not be detected at the USB port or function correctly. Therefore, it is recommended that every USB stick be tested before actual use.

X5: CAN bus 1 (internal)



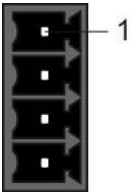
PIN	Function
1	+24 V
2	CAN1 A
3	CAN1 B
4	GND



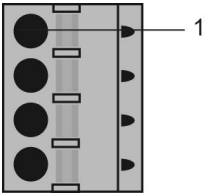
If the CAN bus must be terminated, a 120-Ohm resistor should be inserted into the X5 connector between pin 2 and pin 3.

CAUTION: The CAN bus cannot be connected while voltage is applied (X5, X6)!

X6: CAN bus 2 (external)

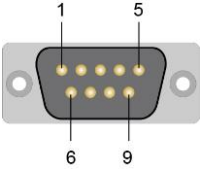


PIN	Function
1	+24 V
2	CAN2 A
3	CAN2 B
4	GND



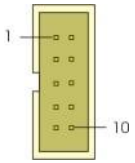
The CAN bus is terminated through the internal electronics!

X7: RS232 (front view)



PIN	Function
1	DCD
2	Rx
3	TX
4	DTR
5	GND
6	DSR
7	RTS
8	CTS
9	RI

X9: CPU board connection to function module 10-Pin blade terminal



PIN	Function
1	CAN A
2	CAN B
3 – 4	GND
5 – 7	+24 V
8	-
9 – 10	GND

Applicable connectors

TCP/IP: 8-pin RJ45
USB: 4-pin Type A (downstream connector)
CAN: 4-pin spring terminal connector FK-MCP1.5/4-ST-3.5
RS232: 9-pin Dsub socket
Function module connection: 10-Pin spring terminal

LEDs

LED	Color	Description
ACTIVE	Yellow	Blinks when data is sent or received.
Link	Green	Blinks when the connection between the two PHYs is established
CANx TX	Yellow	Lights when data is sent.
CANx RX	Green	Lights when data is received.
DC OK	Green	Lights when the power supply is OK

Real-time clock

The HZS 512 central unit has an integrated buffered real-time clock.

The time and date can be used for:

- Recording operation and idle time
- Recording disruption times
- Activation of time-dependant processes

The conversion between summer and wintertime is not automatic!

BIOS

The HZS 512 is started with BIOS developed by SIGMATEK that can boot LASAL and DOS. For this purpose, a bootable micro SD card need only be inserted into the HZS 512.

Wiring Guidelines

Earth Connection

The central unit must be connected to earth through the mounting on the furnace or over the earth connection provided. It is important to create a low-ohm earth connection, only then can error-free operation be guaranteed. The earth connection should have the maximum cross section and the largest electrical surface possible.

Shielding

With Ethernet, a CAT-5 cable with shielded RJ45 connectors is required. The shielding in the CAT-5 cable is connected to earth through the RJ45 connector. Noise therefore cannot reach the electronics and affect the function.

ESD Protection

Typically, USB devices (keyboard, mouse) are not equipped with shielded cables. These devices are disrupted by ESD and in some instances, no longer function. Before any device is connected to or disconnected from the central unit, the potential should be equalized with ground (by touching the control cabinet or earth terminal). Electrostatic loads (through clothing and shoes) can be thereby dissipated.

USB Interface Connections

The central unit has a USB interface that can be used in LASAL to connect various USB devices (keyboard, mouse, storage media, Hubs …). Several USB devices, which are fully functional in LASAL, can be connected using a hub.

CAN Bus Setup

This section explains how to correctly configure the CAN bus. The following parameters must first be set: Station number and data transfer rate.

CAN bus station number

Each CAN bus station is assigned its own station number. With this station number, data can be exchanged with other stations connected to the bus. Up to 31 stations can be installed in a CAN bus system. In a CAN bus system however, each station number can only be assigned once!

CAN bus data transfer rate

Various data transfer rates (baud rate) can be set on the CAN bus; however, the longer the length of the bus, the smaller the transfer rate that must be selected.

Value	Baud rate	Maximum length
0	615 kbits/s	60 m
1	500 kbits/s	80 m
2	250 kbits/s	160 m
3	125 kbits/s	320 m
4	100 kbits/s	400 m
5	50 kbits/s	800 m
6	20 kbits/s	1200 m
7	1 Mbit/s / s	30 m

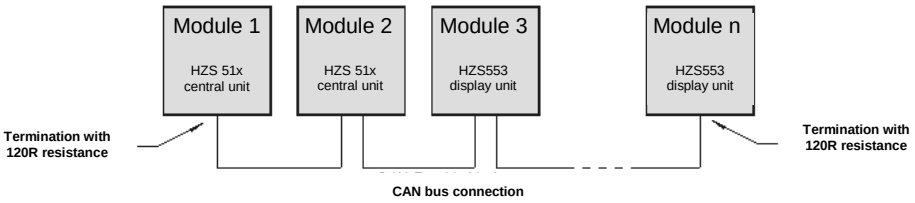
These values are valid for the following cable: 120 Ω, Twisted Pair.

Note: For the CAN bus protocol: 1 kBit/ s = 1 kBaud.

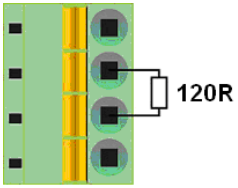
CAN Bus Termination

In a CAN bus system, both end modules must be terminated. This is necessary to avoid transmission errors caused by reflections in the line.

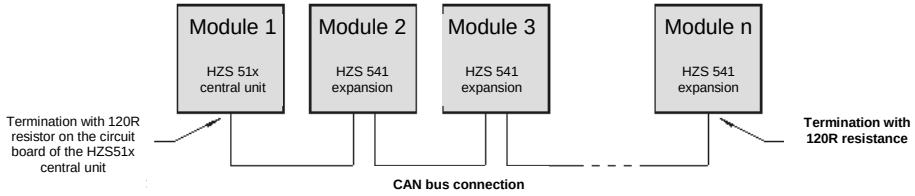
Example: CAN bus 1 HZS 512 central unit:



If the HZS 512 is an end module, it can be terminated by placing a 120-Ohm resistor between CAN-A (Low) and CAN-B (High).

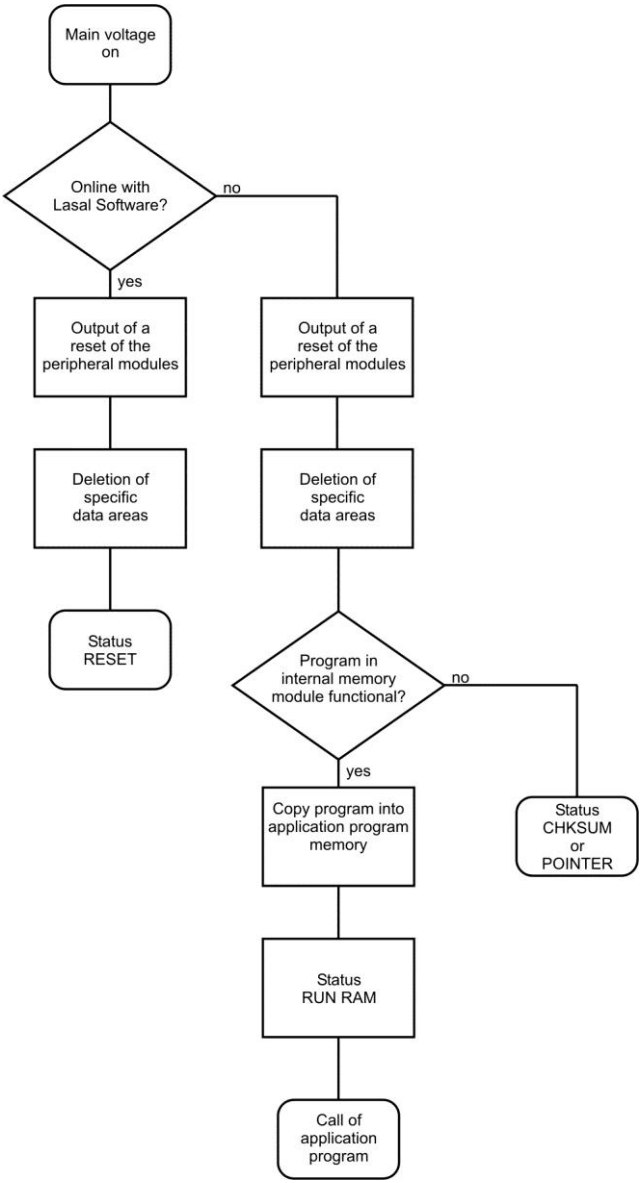


Example: CAN bus 2 HZS 512 central unit:



The central unit HZS 512 is an end unit on the CAN bus and has a 120-Ohm termination on the circuit board between CAN A (LOW) and CAN B (HIGH).

Process Diagram



Status an error Messages

Status of error messages is shown in the LASAL Class software status test. POINTER or CHKSUM messages can also be shown on the terminal screen.

Number	Message	Definition	Cause/solution
00	**RUN RAM**	The user program is currently running in RAM. The display is not affected.	
01	**RUN ROM**	The user program in the program memory module was loaded into the RAM and is currently being run. The display is not affected.	
02	**RUNTIME**	The total duration of all cyclic objects exceeds the maximum time; the time can be configured using 2 system variables: -Runtime: time remaining -SWRuntime: pre-selected value for the runtime counter	
03	**POINTER**	Incorrect program pointers were detected before running the user program	Possible Causes: - The program memory module is missing, not programmed or defect. - The program in the user program memory (RAM) is not executable. - The buffering battery has failed. - The user program is overwriting a software error Solution: - Reprogram the memory module, if the error reoccurs exchange the module. - Exchange the buffering battery - Correct programming error
04	**CHKSUM**	Before running the user program, a false checksum was detected.	Cause/solution: s. POINTER

05	**Watchdog**	The program was interrupted through the watchdog logic.	Possible Causes: - Interrupts the user program blocked of a long time period (STI instruction forgotten) - Programming error in a hardware interrupt. - INB, OUTB, INW, OUTW instructions used incorrectly. - The processor is defect. Solution: - Correct programming error. - Exchange CPU.
06	**GENERAL ERROR**	General error	
07	**PROM DEFECT**	An error has occurred while programming the memory module.	Cause: - The program memory module is defect. - The user program is too large. - The program memory module is missing. Solution: - Exchange the program memory module
08	**Reset**	The CPU has received the reset signal and is waiting for further instructions. The user program is not processed.	
09	**WD DEFEKT**	The hardware monitoring circuit (watchdog logic) is defect. After power-up, the CPU checks the watchdog logic function. If an error occurs during this test, the CPU deliberately enters an infinite loop from which no further instructions are accepted.	Solution: Exchange CPU.
10	**STOP**		
11	**PROG BUSYS**		
12	**PROGRAM LENGTH**		
13	**PROG END**	The memory module was successfully completed.	
14	**PROG MEMO**	The CPU is currently programming the memory module.	

15	**STOP BRKPT**	The CPU was stopped by a breakpoint in the program.	
16	**CPU STOP**	The CPU was stopped by the PG software (F6 HALT in status test).	
17	**INT ERROR**	The CPU has triggered a false interrupt and stopped the user program or has encountered an unknown instruction while running the program.	Cause: - A nonexistent operating system was used. - Stack error (uneven number of PUSH and POP instructions). - The user program was interrupted by a software error. Solution: - Correct programming error.
18	**SINGLE STEP**	The CPU is in single step mode and is waiting for further instructions.	
19	**READY**	A module or project has been sent to the CPU and it is ready to run the program.	
20	**LOAD**	The program has stopped and is receiving a module or project.	
21	**UNZUL. Modul**	The CPU has received a module, which does not belong to the project.	
22	**MEMORY FULL**	The operating system memory /heap) is too small. No more memory could be reserved, when an internal or interface function was called from the application.	
23	**NOT LINKED**	When starting the CPU, a missing module or a module that does not belong to the project was detected.	
24	**DIV BY 0**	A division error has occurred.	Possible Causes: - Division by 0 - The result of a division does not fit in the result register. Solution: - Correct programming error.

25	**DIAS ERROR**	An error has occurred while accessing a DIAS module.	Possible Causes: - An attempt is made to access a nonexistent DIAS module. - DIAS bus error. Solution: - Check the DIAS bus - Check the termination resistors.
26	**WAIT**	The CPU is busy.	
27	**OP PROG**	The operating system is currently being reprogrammed.	
28	**OP INSTALLED**	The operating system has been reinstalled.	
29	**OS TOO LONG**	The operating system cannot be loaded; too little memory.	
30	**NO OPERATING SYSTEM**	Boot loader message. No operating system found in RAM.	
31	**SEARCH FOR OS**	The boot loader is searching for the operating system in RAM.	
32	**NO DEVICE**		
33	**UNUSED CODE**		
34	**MEM ERROR**	The operating system loaded does not match the hardware configuration.	
35	**MAX IO**		
36	**MODULE LOAD ERROR**	The LASAL Module or project cannot be loaded.	
37	**GENERELLER BS-FEHLER**	A general error has occurred while loading the operating system.	
38	**APPLMEM ERROR**	An error has occurred in the application memory (user heap).	
39	**Offline**		
40	**APPL LOAD**		
41	**APPL SAVE**		

46	**APPL-LOAD-ERROR**	An error has occurred while loading the application.	
47	**APPL-SAVE-ERROR**	An error has occurred while attempting to save the application.	
50	**ACCESS-EXCEPTION-ERROR**	Read or write access of a restricted memory area. (e.g. writing to the NULL pointer).	
51	**BOUND EXCEEDED**	An exception error caused by exceeding the memory limits	
52	**PRIVILEDGED INSTRUCTION**	An unauthorized instruction for the current CPU level was given. For example, setting the segment register.	
53	**FLOATING POINT ERROR**	An error has occurred during a floating-point operation.	
60	**DIAS-RISC-ERROR**	Error from the Intelligent DIAS-Master.	
64	**INTERNAL ERROR**	An internal error has occurred, all applications are stopped.	Restart; report error to Sigmatek.
65	**FILE ERROR**	An error has occurred during a file operation.	
66	**DEBUG ASSERTION FAILED**	Internal error.	Restart; report error to Sigmatek.
67	**REALTIME RUNTIME**	The total duration of all real-time objects exceeds the maximum time; the time cannot be configured. 2 ms for 386 CPUs 1 ms for all other CPUs	Starting from Version 1.1.7
68	**BACKGROUND RUNTIME**	The total time for all background objects exceed the maximum time; the time can be configured using two system variables: -BTRuntime: time remaining -SWBTRuntime: pre-selected value for the runtime counter	
70	**C-DIAS ERROR**	An error occurred in connection with a C-DIAS module.	Cause: - The reason for this error is documented in the log file Solution: - Depends on the cause

75	**SRAM ERROR**	Only EDGE CPUs An error occurred while initializing, reading or writing SRAM data.	Possible causes: - - SRAM configured incorrectly - - SD card formatted incorrectly - - SD card removed Solution: - - evaluate log file (Event00.log) - - check configuration - - format SD card as EDGE medium with Lasal Class 2 - - check SD card
95	**USER DEFINED 0**	User-definable code.	
96	**USER DEFINED 1**	User-definable code.	
97	**USER DEFINED 2**	User-definable code.	
98	**USER DEFINED 3**	User-definable code.	
9	**USER DEFINED 4**	User-definable code.	
100	**C_INIT**	Initialization start; the configuration is run.	
101	**C_RUNRAM**	The LASAL project was successfully started from RAM.	
102	**C_RUNROM**	The LASAL project was successfully started from ROM.	
103	**C_RUNTIME**		
104	**C_READY**	The CPU is ready for operation.	
105	**C_OK**	The CPU is ready for operation.	
106	**C_UNKNOWN_CID**	An unknown class from a stand-along or embedded object: unknown base class.	
107	**C_UNKNOWN_CONSTR**	The operating system class cannot be created; the operating system is probably wrong.	
108	**C_UNKNOWN_OBJECT**	Reference to an unknown object in an interpreter program, creation of more than one DCC080 object.	
109	**C_UNKNOWN_CHNL**	The hardware module number is greater than 60.	

110	**C_WRONG_CONNECT**	No connection to the required channels.	
111	**C_WRONG_ATTR**	Wrong server attribute.	
112	**C_SYNTAX_ERROR**	No specific error, recompile all and reload project components.	
113	**C_NO_FILE_OPEN**	An attempt was made to open an unknown table.	
114	**C_OUTOF_NEAR**	Memory allocation error	
115	**C_OUT OF_FAR**	Memory allocation error	
116	**C_INCOMAPTIBLE**	An object with the same name exists but has another class.	
117	**C_COMPATIBLE**	An object with the same name and class exists but must be updated.	
224	**LINKING**	The application is currently linking.	
225	**LINKING ERROR**	An error has occurred while linking. An error messaged is generated in the LASAL status window.	
226	**LINKING DONE**	Linking is complete.	
230	**OP BURN**	The operating system is currently being burned into the Flash memory.	
231	**OP BURN FAIL**	An error has occurred while burning the operating system.	
232	**OP INSTALL**	The operating system is currently being installed.	
240	**USV-WAIT**	The power supply was disconnected; the UPS is active.	
241	**Reboot**	The operating system is restarted.	
242	**LSL SAVE**		
243	**LSL LOAD**		
252	**CONTINUE**		
253	**PRERUN**	The application is started.	
254	**PRERESET**	The application is ended.	
255	**CONNECTION BREAK**		

Application exceptions

SRAM and IRQ routines

Writing remnant data during interrupt routines is not allowed and leads to a system crash.

SRAM and consistency of changed data

If more than 32 different sectors are changed (512 bytes each) shortly before shutting down the voltage supply while the user program is writing to the Micro SD card can sometimes lead to partial loss of remnant data.

The file system does not support safe writing through SRAM

If files are stored, modified or written on the micro SD card from the user program, these files must always be stored with a fixed maximum size. Since changes in size and the simultaneous shutdown of the voltage supply can corrupt the file system, a later change in the file size is not allowed.

Data Breakpoint

This CPU does not support the data breakpoint feature.

Note on SRAM Behavior

Because the SRAM (remnant memory) is emulated via the microSD card, there are two different mechanisms for saving SRAM data to the microSD card:

1. Cyclic writing when data is changed (default)

2. Writing only in the event of PowerFail with a backup time buffered through the hardware (starting with version 01.02.195)

The advantage of cyclic writing is that in the event of a severe system crash, it's possible to reference an image of the SRAM data that with the standard settings, is a maximum of 1 minute older than the last change. With extensive use, the amount and frequency of SRAM data changes from the user program can have a massive effect on the microSD card lifespan.

Detailed information regarding the SRAM behavior and the corresponding settings can be found in the LASAL OS documentation, in the chapter “SRAM”.

In the LASAL CLASS project, seldom changed value settings in retentive servers as well as RamEx and StringRam objects, can be converted to file storage. Should existing objects be converted from SRAM to File, the loader version 02.02.140 or higher and the RamEx and StringRam classes of the Tools library version 01.02.033 or higher must be used.

If the user program runs cyclic writing processes in files, the tool “Flash Media Lifetime Calculation” included in LASAL CLASS can be used to determine the effects of the operations mentioned above on the flash media. This allows the lifespan of the media to be calculated for different, configurable writing scenarios.