



---

## HYDRONIC\_Notifications

<b>1.</b>	Overview .....	2
<b>2</b>	System notifications .....	2
2.1	IOs addressing messages.....	2
2.2	Firmware update messages.....	2
2.3	Runtime execution messages .....	2
2.3.1.	Runtime python code execution error messages .....	3
2.3.2.	Runtime communication error messages.....	3
2.3.3.	Runtime IO devices messages .....	4
2.3.4.	Runtime filesystem messages .....	4
2.4	System Reset messages .....	4
2.5	System core error messages .....	5
2.6	File integrity messages .....	5
<b>3</b>	UI notifications .....	6
3.1	Project status messages.....	6
3.2	UI update messages.....	6
3.3	Compile and activate messages .....	6



This document describes all the possible notification messages logged in the HYDRONIC system due to either specific system behavior or important user actions.

The user may see these notifications in the UI either at the login page under the section “uC Messages” or as a programmer user in “HYDRONIC uController Configuration” modal under section “uController Messages”.

Generally, the notifications are divided into two major categories, the system notifications and the UI notifications.

## 2 System notifications

These notifications inform the user about important system behavior and can be further subdivided into the following categories:

### 2.1 IOs addressing messages

The notifications related to the IO addressing procedure may be the following:

- “There have been detected changes in IO devices ordering”. This message will be displayed upon reordering of devices, which were previously already recognized during successful IO addressing.
- “At position <position>, the IO has been replaced by another device of the same type <IO type>, but with different SN”. This message will be displayed if an IO device is replaced with another IO of the same type.

The following messages will be displayed upon unsuccessful IO addressing and inform the user about all the misconfigured IO devices detected.

- “misconfiguredIOs:[<list of misconfigured IOs>]”,
- “At position <position>, <IO type> is missing” and
- “At position <position> there should be <IO type correct>, and not <IO type wrong>”.

### 2.2 Firmware update messages

The notifications related to the Firmware update of the CON64A may be the following:

- “Firmware update to version <version> was successful!”
- “Firmware update failed, please restart the system and try again”.
- “Firmware update unspecified error, please restart the system and try again”.

### 2.3 Runtime execution messages

The following messages will be displayed when an active project is running on the master device:

### 2.3.1. Runtime python code execution error messages

These messages are standard python errors. They concern the executable python code and will be displayed upon the python code execution, in case of unexpected code behavior. When these errors occur, the master device shall reset automatically itself, reactivate the project and execute again the micropython code. If the runtime code error still persists, then the master device shall deactivate the project and reset again itself automatically for a second and final time. The error will be presented with the following two messages:

- “Python code execution error. Waiting for traceback...” – Upon any type of runtime error, this message shall be displayed first, notifying the user that a runtime error occurred.
- “line : <number>, message : <error type with description>” – The second message will show detailed information about the error (line and specific error type with description). For example, the error type can be one of the following:
  - “NameError”
  - “MemoryError”
  - “AttributeError”
  - “TypeError”
  - “ZeroDivisionError”
  - “ImportError”
  - “IndexError”
  - “KeyError”

For a full list of the possible error types for Python in general, please refer to:  
<https://docs.python.org/3/library/exceptions.html>

### 2.3.2. Runtime communication error messages

The following messages concern the communication between the master device and the IOs devices during an active project:

These messages will be displayed when an active project is running and the master-IOs devices communication becomes corrupted or unexpectedly interrupted:

- “Errors detected in the IOs communication with the master”,
- “Execute IO Addressing again to see more details about the missing IOs. Trying to reconnect...” –

In case that the communication is successfully restored automatically after having been interrupted:

- “The IOs communication errors were automatically resolved successfully...”

### 2.3.3. Runtime IO devices messages

The following messages concern the IO devices behavior during an active project:

- “Please connect a sensor for <IO type>, channel <channel>” – This message is a warning and informs the user that an AIN5A channel has been configured as sensor-type, but no sensor has been detected to be properly connected to that channel. This message will be logged only during project activation by the programmer user.
- “IOs misconfiguration was detected upon reset with active project”. – This message will be displayed if a project has been activated and upon system reset the master device detects that the IO devices do not match the IOs configuration. In that case, the project becomes deactivated automatically.

### 2.3.4. Runtime filesystem messages

The following message concerns the filesystem in general during runtime:

- “Filesystem was reinitialized.” – During runtime, the filesystem constantly executes various filesystem operations (open, close, write, read, traverse directory, etc). Sometimes due to heavy workload, some of these operations (close file, traverse directory) may fail. In order to overcome this, the system checks periodically the filesystem and resolves such issues. In case these errors persist, the system reinitializes the filesystem, logs this message and continues operating normally.

## 2.4 System Reset messages

The following messages inform the user about the root cause of a system reset:

- “Brownout reset” and “Low power reset” – Reset caused when the supply voltage goes below a certain voltage level (specified by manufacturer) and the device stays into a reset state to ensure proper startup when power returns.
- “Window watchdog reset” and “Independent watchdog reset” - The watchdog timers are used to detect and resolve malfunctions due to software failures. It triggers a reset sequence when it is not refreshed within the expected time-window.
- “Software reset” – System reset from the software using high priority interrupt.
- “Power-on/power-off reset” – Reset caused by system power off - power on (power supply switched off and on)
- “External reset pin reset” – Reset is caused by an external source.

**Note:** Usually, only “software reset” and “power-on/power-off reset” messages shall be displayed during normal operation as reset reason. For more information, please refer to the

manufacturer documentation:

<https://www.st.com/en/microcontrollers-microprocessors/stm32f437vi.html>

## 2.5 System core error messages

These errors indicate that the microcontroller reached an illegal program state caused by an unknown code behavior and such errors normally must never occur. Nevertheless, in case that such an error occurs, the master device automatically resets itself with a software reset and afterwards the system continues to operate normally.

These errors in the HYDRONIC system that are currently logged are the following:

- “System core MemManage error” – exception for memory access violations to regions that are defined in the Memory Management Unit (MPU); for example, code execution from a memory region with read/write access only.
- “System core BusFault error” – exception for memory access violation on instruction fetch, data read/write, interrupt vector fetch, and register stacking (save/restore) on interrupt (entry/exit).
- “System core UsageFault error” – exception for execution of undefined instructions (divide-by-zero) and unaligned memory access for load/store multiple.
- “System core HardFault error” – the ultimate exception that can be triggered because of an error during other exception processing, or because an exception cannot be managed by any other exception mechanism.
- “System core HAL error” – indicates error at application level caused by system peripherals and internal clock initialization.
- “System core RTOS configASSERT error” – indicates error in the tasks of the real time operating system.
- “System core NLR jump fail error” – indicates micropython execution error during exception handling.
- “System core general assert error (filesystem, micropython or other system assert)” – indicates filesystem low level error or general micropython low level error.
- “System core undefined error” – indicates any other unknown exception.

**Note:** For more information, please refer to the supplier documentation “Using Cortex-M3/M4/M7 Fault Exceptions”: <http://www.keil.com/appnotes/files/apnt209.pdf>

## 2.6 File integrity messages

These messages are related to JSON files, in case either the file size is too large to parse or data corruption is detected upon read/write operation. Data corruption means that the file does not contain valid JSON formatted data and in this case, only for a few critical files, the respective backup file shall be used. If the backup file is corrupted as well, then

both files are set to their default data. In normal system operation, and only for the critical files, the system keeps backup files (denoted with the suffix “\_b”) to use in such cases of corrupted data.

The possible file integrity messages are the following:

- “Data corruption detected inside <filename.json> file upon read/write operation.”, in case any file is found to be corrupted.
- “Backup file for <filename.json> will be used instead.”, in case a critical file is found to be corrupted and its respective backup file contains valid JSON data and is not corrupted.
- “<filename.json> file is too large to parse upon JSON read operation.”, in case a file contains too much JSON data to parse and there is not sufficient memory to load its contents.
- “<filename.json> and <filename\_b.json> files are written with their default data.”, in case a critical file is found to be corrupted and its respective backup file is corrupted as well. In this case, both files shall be reset to their default data (same for both files).

### 3 UI notifications

These notifications inform the user about important user actions in the UI and are the following:

#### 3.1 Project status messages

The following messages inform the user about project activation or deactivation:

- “An activation has been performed by the <user>”.
- “A deactivation has been performed by the <user>”.

#### 3.2 UI update messages

The following messages inform the user about the UI update:

- “UI update to version <version> has been performed successfully”.
- “UI update was unsuccessful. Please try again”.

#### 3.3 Compile and activate messages

During the "Compile" and "Compile and Activate" actions, the python code is analyzed for syntax errors. In case any errors are found, these are displayed in the "Info" section which



## HYDRONIC\_Notifications

---

can be found under the "Routines" section of the selected project in the programming page.

The line of the error along with the error description is logged. This line refers to the combined code of the routines which is displayed as a pop-up window after the procedure is complete.