

Programming a Bypass

There are two different ways to program a bypass in the HHL-C central unit. This guide describes these options and their differences. The bypass options are known as address unit bypass and user-based bypass.

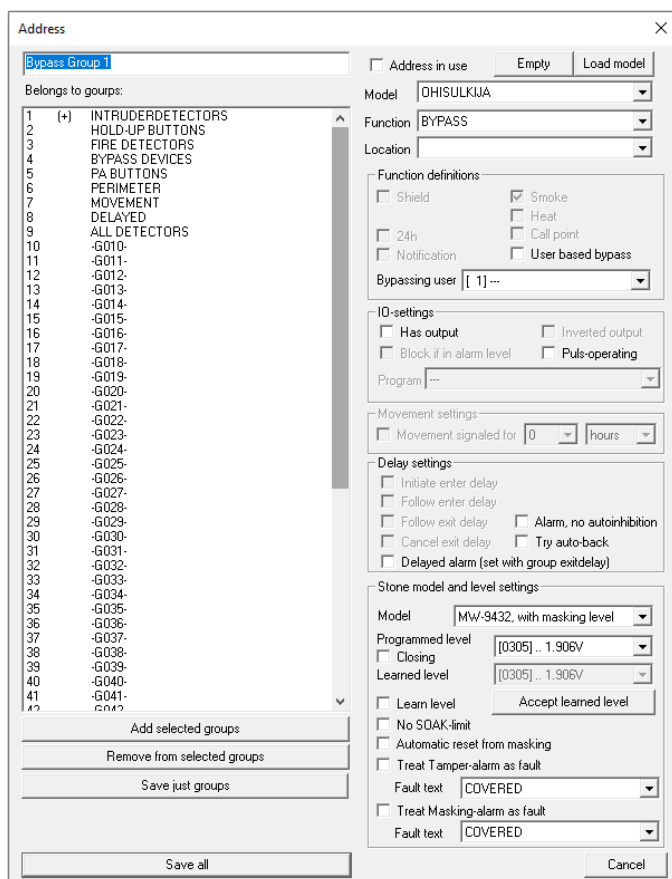


Note! When programming/modifying a bypass, the address unit must be in rest state.

Address Unit Bypass

With an address unit bypass, the bypass is instantaneous and the group's exit delay is ignored.

1. Select the address that performs the bypass.
2. Type a name for the address and select the group(s) you want to bypass.
3. Select **Address in use**.
4. Select **Model: Bypass device** and click **Load model** to retrieve the **Bypass** function for the address. Alternatively, select **Function: Bypass Device** manually.
5. Program the settings for the address unit.
6. Save the address data and send it to the central unit.



Technical Note

HHL-C Alarm System

TN-HHL-C-Programming-Bypass-EN | Version 1.0 | 10.1.2024



User-based Bypass

A user-based bypass can use the exit delay that is set for the group. The bypass function is tied to a stack user.

1. Create a new stack user and define the group(s) to bypass.
2. Set the **Level** and **Code** for the user. This user does not require access rights to a user panel.
3. Save the user data and send it to the central unit.

Technical Note

HHL-C Alarm System

TN-HHL-C-Programming-Bypass-EN | Version 1.0 | 10.1.2024



4. After creating the user, select the address that performs the bypass.
5. Type a name for the address and select the group(s) you want to bypass.
6. Select **Address in use**.
7. Select **Model: Bypass device** and click **Load model** to retrieve the **Bypass** function for the address. Alternatively, select **Function: Bypass Device** manually.
8. Under **Function definitions**, select **User based bypass** and select the stack user you created from the **Bypassing user** dropdown list.
9. Program the settings for the address unit.
10. Save the address data and send it to the central unit.