

Instruction manual

Climate chamber



The photo above is for reference only and may show options not included in the standard equipment. The real appearance, colour and structure of the material may differ from the ones presented in the photo.

This document serves as a guide for communication between the HMI and the controllers in the standard climate chamber project. It will describe how the HMI is put together, how it communicates with the controllers and what signals are or can be exchanged.

Reference will also be made to controller documentation. It will indicate how it should be interpreted so that there is no ambiguity about how it is addressed.

If the document no longer matches the HMI in the climate chamber, please contact Apollo Service Handelsonderneming B.V.

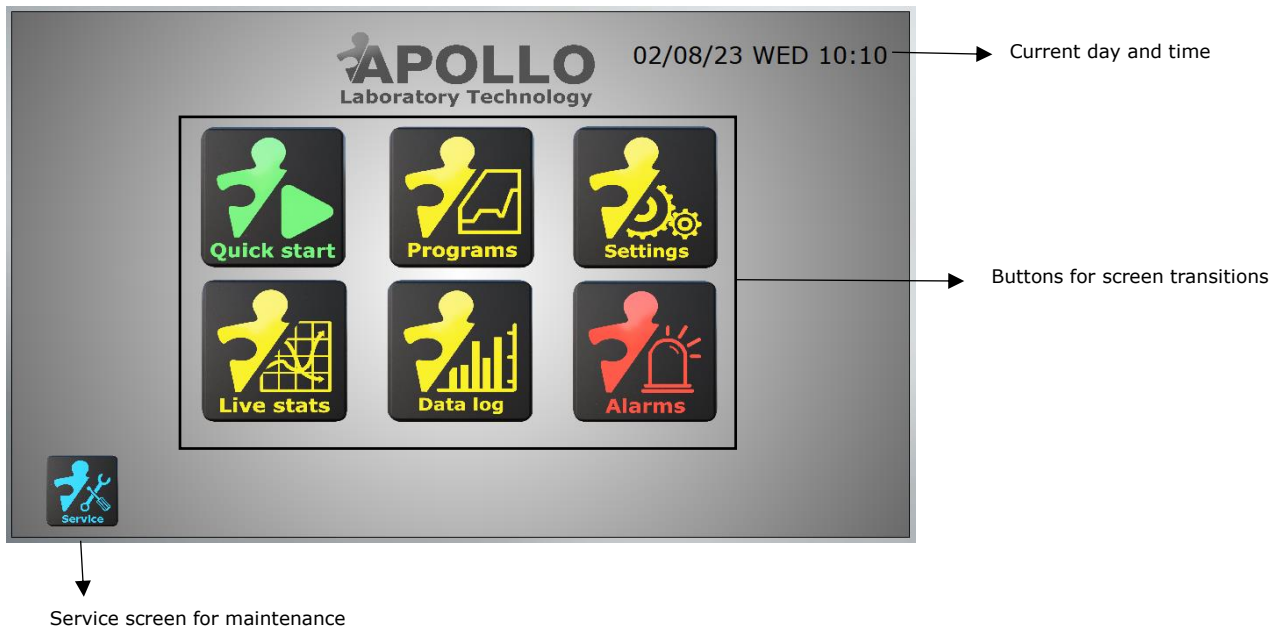
TABLE OF CONTENTS

1. HUMAN MACHINE INTERFACE – (HMI)	4
2. CONNECTION BETWEEN THE HMI AND CONTROLLERS	8
3. SENDING DATA TO THE CONTROLLER	9
4. COMMUNICATION BETWEEN PC AND CONTROLLER	11
5. IMPORTANT ADDRESSES IN THE CONTROLLER	12

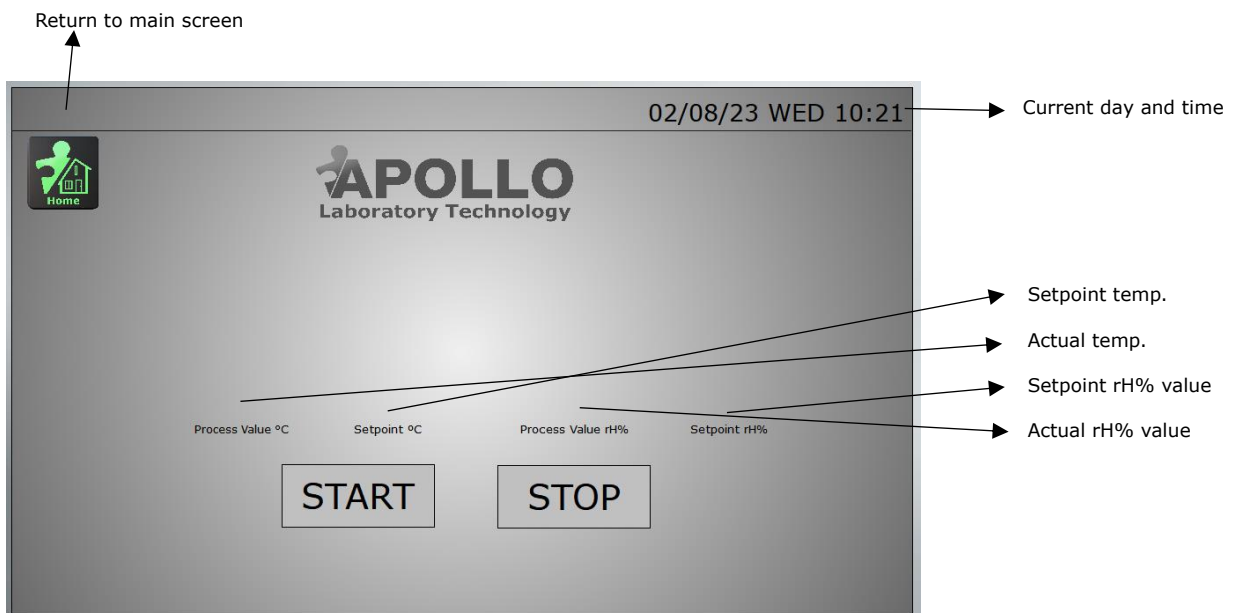
1. HUMAN MACHINE INTERFACE – (HMI)

For each screen, this section describes how the HMI is put together visually. The illustrations below show what each part does. If this no longer matches the current version of the HMI, please contact Apollo Service Handelsonderneming B.V.

1.1 Main screen

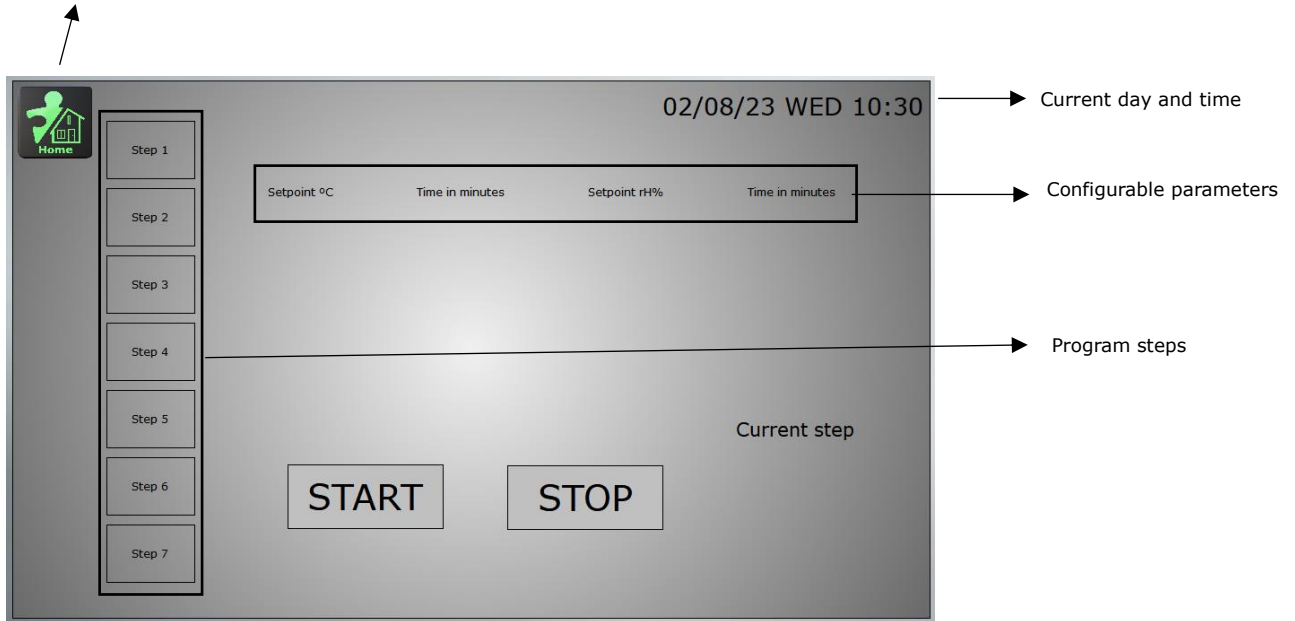


1.2 Quick start



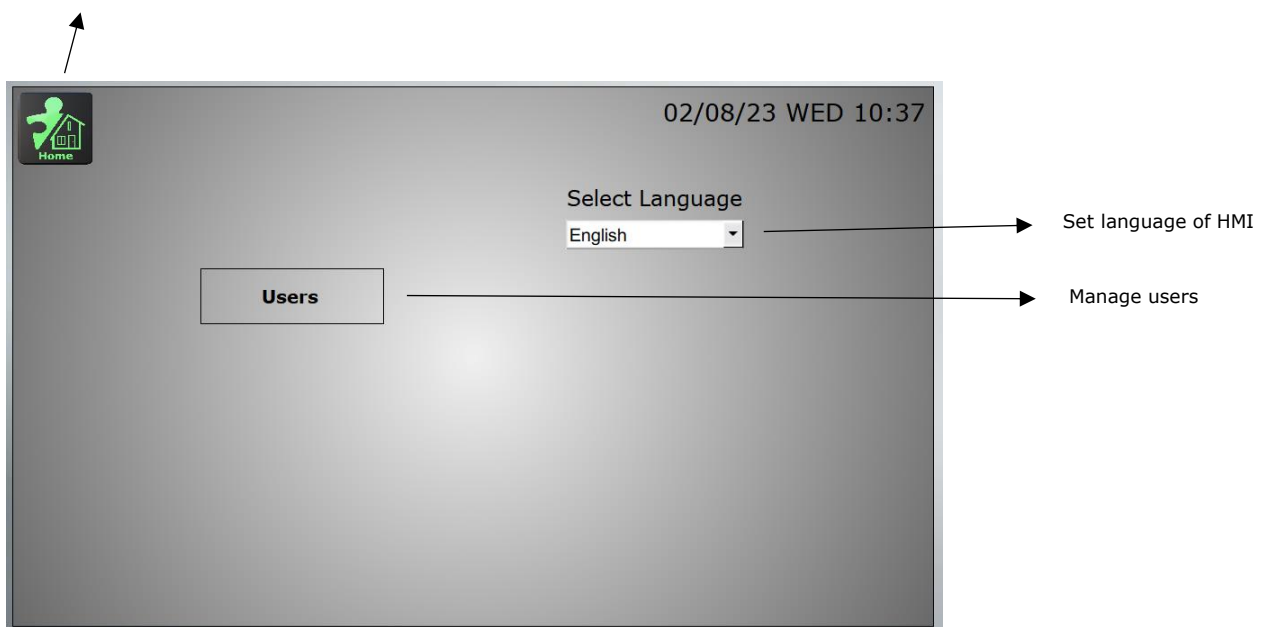
1.3 Programs

Return to main screen



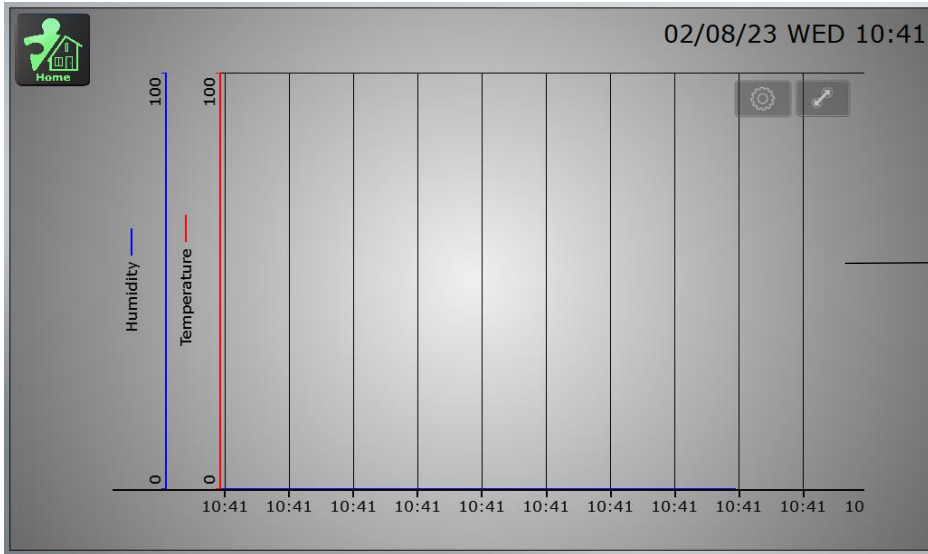
1.4 Settings

Return to main screen



1.5 Live statistics

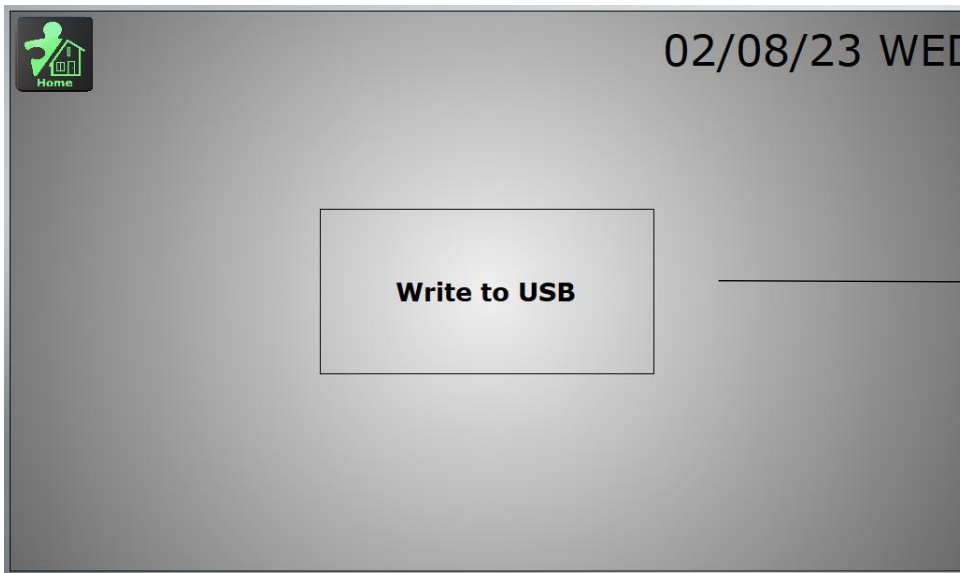
Return to main screen



Trend chart current values

1.6 Data log

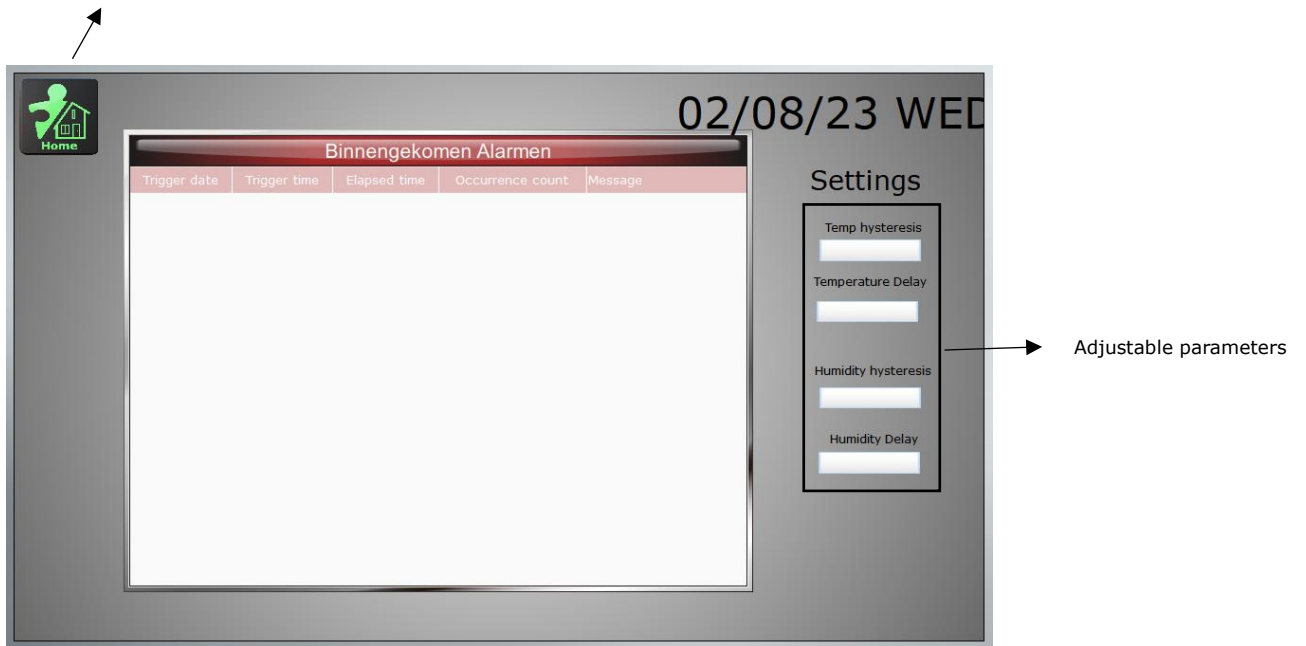
Return to main screen



Write data to USB

1.7 Alarms

Return to main screen



2. CONNECTION BETWEEN THE HMI AND CONTROLLERS

In the system parameters, the connection protocol between the HMI and the controller can be found and adjusted if necessary. If it has been adjusted and no longer works, the following settings can be used to restore the connection. For connection between the cMT-FHDX-820 HMI and the RKC controllers, the MODBUS protocol is used, specifically the MODBUS RTU (Zero-based Addressing) protocol.

The controllers each have their own device address. It is important to create a separate connection for both controllers. Leave the default station no. unchanged.

In the settings, choose "I/F:" for the RS-485 2W option. This means communication over Modbus 485 2 wire.

For the COM port, choose COM 3. Here, set the baud rate to 19200, no parity, 8 data bits and 1 stop bit.

A complete overview of the settings is shown in the chart below:

Settings	Option
I/F	RS-485 2W
COM poort	COM 3
Baudrate	19200
Data bit	8
Stop bit	1

Chart 1: Settings connection

3. SENDING DATA TO THE CONTROLLER

To send data to the controller, an object must be linked between the HMI and the controller. When the object is created, the attributes can be used to specify which address to link to in the controller. There is always read or write on the 4x registers in the controllers.

The address is structured as follows:

[Device address controller in Modbus] # [Byte number in decimal digits] : [bit number in two decimal digits].

Below are some examples to show how this works. The examples below are done with random numbers and may not be executable on the controller.

Example 1:

User would like to write a WORD to the TIO-A module.

This module is located at device address 1.

The byte number to write the WORD to is 112 in decimal digits.

Since a WORD is being written, no bit number is specified.

The object is then given the following as its read/write address:

[4x] 1#112

This writes what is being entered into the object in register 4 of device 1 at byte 112.

Example 2:

User would like to write a BIT to the TIO-A module.

This module is located at device address 1.

The byte number the BIT should be written to is 81 in decimal digits.

The bit number the BIT should be written to is 2.

The object is then given the following as its read/write address:

[4x] 1#81:02

This writes register 4 of device 1 to byte 81 bit 2 what is entered into the object. Here it is important to indicate that the bit number is written as '02', not '2'. The bit number should always be written in two decimal digits.

Example 3:

User would like to write a BIT to the DIO module.

This module is located at device address 17. (The station address is not the same as the device address)

The byte number the BIT should be written to is 5 in decimal digits.

The bit number the BIT should be written to is 4.

The object is then given the following as its read/write address:

[4x] 17#5:04

This writes to register 4 of device 17 on byte 5 bit 4 what is entered into the object.

The reason the DIO module is at device address 17 is because this was a choice of RKC. The address can be set on the TIO-A and DIO modules. Here, for the TIO-A module for the actual device 1 must be added, while for the DIO modules 17 must be added. If both controllers are set to "device address 0" on the physical controllers, it means that they are in reality address 1 for the TIO-A and 17 for the DIO module.

IMPORTANT:

While the controller is set to MODBUS protocol, it can still be communicated with via the RKC protocol via the loader port. This can be done regardless of which way the controller is set. The PROTEM2 program can be used to read and control the controller at any time. Actual control via this program is shown later in the document.

4. COMMUNICATION BETWEEN PC AND CONTROLLER

When the controller needs to be addressed directly for writing Engineer Settings or other reasons, it can be done through the Loader port on the front on the controller. An RKC adapter can be used to connect from a USB port on the laptop/PC.

This can also be done while the HMI is communicating with the controller over MODBUS. While the controller is communicating over MODBUS, it can be addressed via the loader port.

The program first displays a menu for the type of controller you want to address. Here, choose the BASE TOOL.

When you have chosen the BASE TOOL, a new project can be created where you choose the one you are connected to. When a controller is already created you can choose it and press [OK].

After this the settings in the program can be modified and the controller can be controlled/read. The settings and memory areas can be updated live when the connection to the controller is active.

Important:

When the engineer settings need to be adjusted, the controller must be in STOP (or HOLD). When the controller is in RUN, the engineer settings cannot be adjusted.

To put the controller in RUN/STOP, under the [Operations] heading, a 0 can be written to the [RUN/STOP Transfer] option to put the controller in STOP, or a 1 can be written to the [RUN/STOP Transfer] option to put the controller in RUN.

When using the memory areas, pay close attention to which areas are being written or read and where the controller is actually running. Typically (and in the current configuration of the HMI), memory area 8 is used for the quick start function and areas 1 through 7 are available for the stepper program. A quick overview of this is shown below:

Memory area	Function in the HMI
1	Programs: Step 1 parameters
2	Programs: Step 2 parameters
3	Programs: Step 3 parameters
4	Programs: Step 4 parameters
5	Programs: Step 5 parameters
6	Programs: Step 6 parameters
7	Programs: Step 7 parameters
8	Quick start parameters

Tabel 2: Memory Area in the HMI

5. IMPORTANT ADDRESSES IN THE CONTROLLER

This section describes the addresses of interest when communicating between the controller and a laptop or between the controller and the HMI.

The table below describes which values are interesting to read/write via the HMI and/or PC to/from the controller.

If there are addresses where these are defined as [A ... B], this means that they are separately configurable for channels 1 to 4 respectively.

A complete overview of all I/O of the controller can be found in the manual provided for the TIO-A and DIO controller or via the datasheet.

Z-TIO-A Module			
Name	MODBUS address	Bit	Function
Proces Value	0 ... 3	-	Input addresses for sensors
Operations state ⁽¹⁾	9 ... 11	0 ... 3	Bit 0: STOP mode Bit 1: RUN mode Bit 2: Manual mode Bit 3: Remote mode (not applicable at present)
Autotune function	97 ... 100	-	0: Autotune off 1: Autotune on (When the autotune is finished it turns itself off)
Auto/manual mode	101 ... 104	-	0: Automatic mode 1: Manual mode
Memory Area	110 ... 113	-	1 t/m 8, depending on which area you want to assign to a channel.
Setpoint	142 ... 145	-	Setting a set point per channel.
⁽²⁾ Control Response Parameter	158 ... 161	-	0: Slow 1: Medium 2: Fast
⁽³⁾ Overlap	174 ... 177	-	Input voltage from -100% to 100%
Area soak time	190 ... 193	-	Time per memory area/step in programs in minutes (or seconds depending on settings)
Operations mode	322 ... 325	-	1: Monitor 2: Monitor + Event 3: Control
Input type (sensor)	374 ... 377	-	12: PT100 15: 4 to 20 mA DC
⁽⁴⁾ Control Action	562 ... 565	-	0: Brilliant PID direct action 1: Brilliant PID reverse action 2: Heat/Cool control (water-cooling) 3: Heat/Cool control (air-cooling) 4: Heat/Cool control (linear cool gain) 5: Position proportioning control
Z-DIO Module			
Name	MODBUS adres	Bit	Function

Digital Input	0	0 ... 7	A high or low signal can be read in per bit
Digital Output	1	0 ... 7	Reading out which outputs are high or low
RUN/STOP	70	-	0: STOP 1: RUN

(1)With this address, the input must be given in decimal digits and you do not directly address a bit. The input is a decimal number from 0 to 15.

Examples:

So for run mode you enter 2. For Stop mode you enter 1.

(2)Works only when the I action is active. There is an overshoot at Medium and Fast response.

(3)Use only the negative values with this option; the (-) signals the controller that the overlap function should become active. This is useful for using a heat/cool control to output multiple outputs simultaneously.

(4)Channels 1 and 3 can receive all control settings. Channels 2 and 4 can only be set to direct or reverse control.