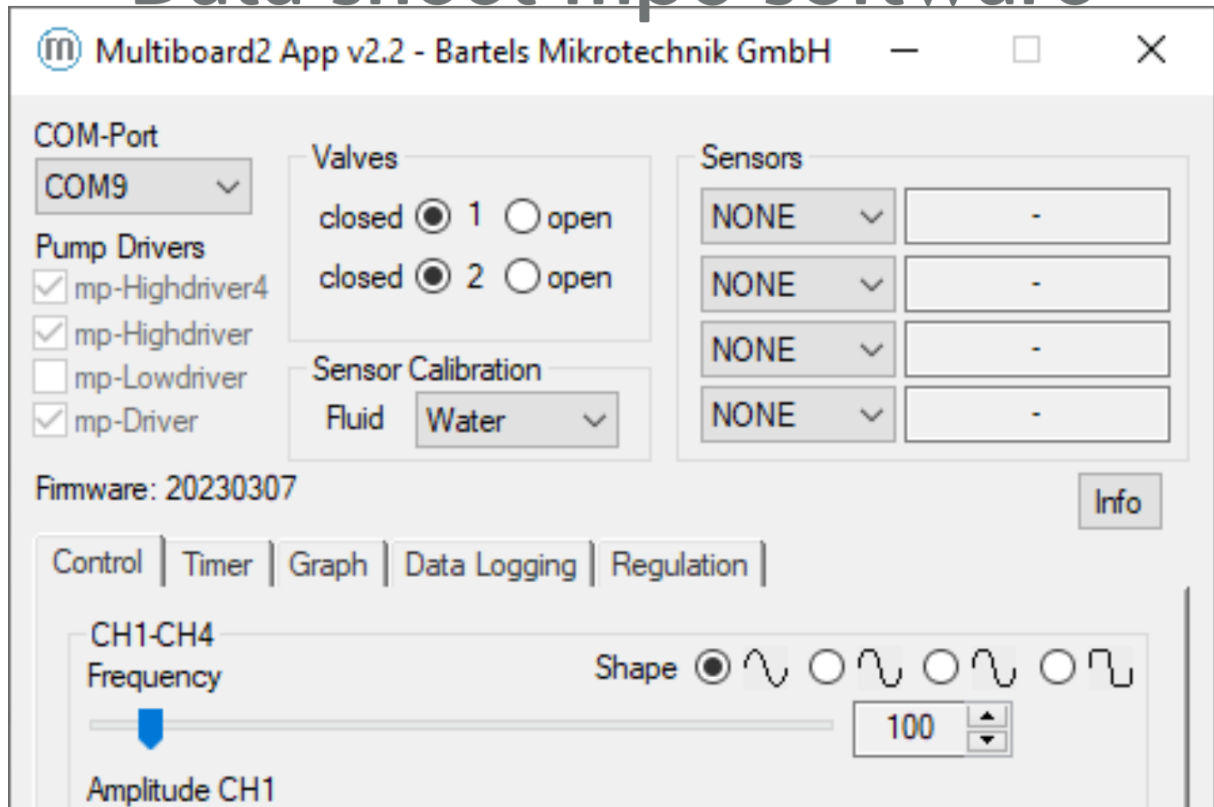


Data sheet mp6 software



Summary

This document contains detailed information on all software by Bartels Mikrotechnik for the mp6 micropump series as well as User Interface descriptions and USB driver installation guides.

Bartels software comprises of laboratory solutions with access to the full range of driving parameters down to smaller solutions for smart micro controller applications.

The evaluation board allows quick and flexible setups with our four integrable pumpdriver options and various flow and pressure sensors. Furthermore, we make an easy and user-friendly access possible with user-interfaces and Open-Source Codes for easier tinkering and prototyping for our customers.

Contents

1	Revision History.....	3
2	Software documentation for System- and Subsystem solutions.....	3
2.1	mp-Multiboard2.....	3
2.1.1	Setup instructions.....	3
2.1.2	USB-driver (Windows).....	3
2.1.3	Multiboard2 App.....	5
2.1.4	Updating the mp-Multiboard2 Firmware.....	10
2.1.5	Requesting new Features.....	12
2.1.6	USB/Serial Communication Protocol.....	14
2.1.7	Example Codes for.....	15
2.2	mp-Labtronix controller.....	19
2.2.1	Installation of the drivers "USB Micropump Control" and "USB Serial Port".....	19
2.2.2	Operation via USB port (after installation of the drivers).....	21
2.2.3	Possible commands (followed by the enter key).....	21
3	Discontinued Products.....	22
3.1	Evaluation Board: mp-Multiboard.....	22
3.1.1	Auxiliary connectors.....	22
3.1.2	USB-driver.....	22
3.1.3	Multiboard App.....	22
3.1.4	USB/Serial Communication Protocol.....	25
4	Important Notices.....	26
4.1	Warranty.....	26
4.2	Warning, Personal Injury.....	27
4.3	Declaration of conformity.....	28
5	Company information.....	28

1 Revision History

Original (Nov. 2022)

- Initial Release version

Changes from Original (Jan. 2022) to Revision 1.1

- New Flow Control types added in the Apps Regulation tab (Indirect Flow & Volume Dispensing)
- New Chapter: 2.1.6 Example Codes

2 Software documentation for System- and Subsystem solutions

2.1 mp-Multiboard2



2.1.1 Setup instructions

Make sure pump driver, pump cable harness and pumps are plugged in correctly; install the esp32 USB driver; connect the Multiboard via USB cable to the computer; start the Multiboard2 App

Do not disconnect pumps or driver while Multiboard is powered (via USB or otherwise)!

2.1.2 USB-driver (Windows)

In case that the ESP32 is not automatically recognised by your PC the USB driver must be installed manually. You can find the drivers on the bartels websites download section. Unzip the mp-Multiboard2_USB_Driver.zip and right-click the "silabser.inf" setup information file. Then from the menu select "install" and it should automatically install them on your device

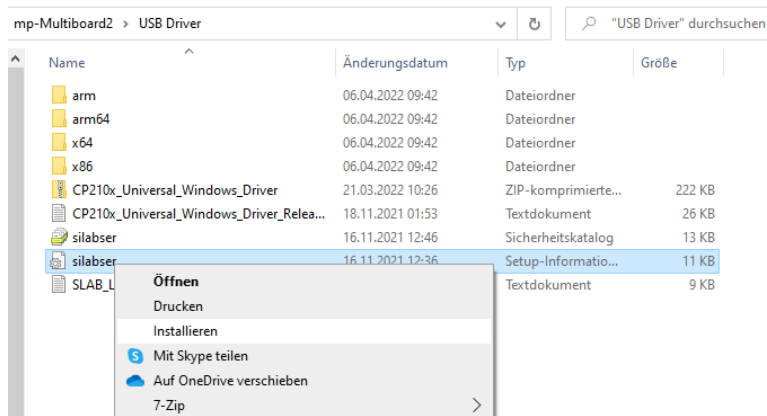


Figure 1 USB driver installation

Multiboard2 App

The Multiboard2 App is a Software tool that enables you to control various features of the Multiboard remotely via USB. It interfaces with the Esp32 via the USB-Serial interface and calls the protocol commands of the Firmware. A description of the protocol and list of commands can be found in 2.1.5 USB/Serial Communication Protocol, allowing to use the Multiboard2 using other software like LabVIEW, Matlab and python.

After executing the app, the following screen opens up:

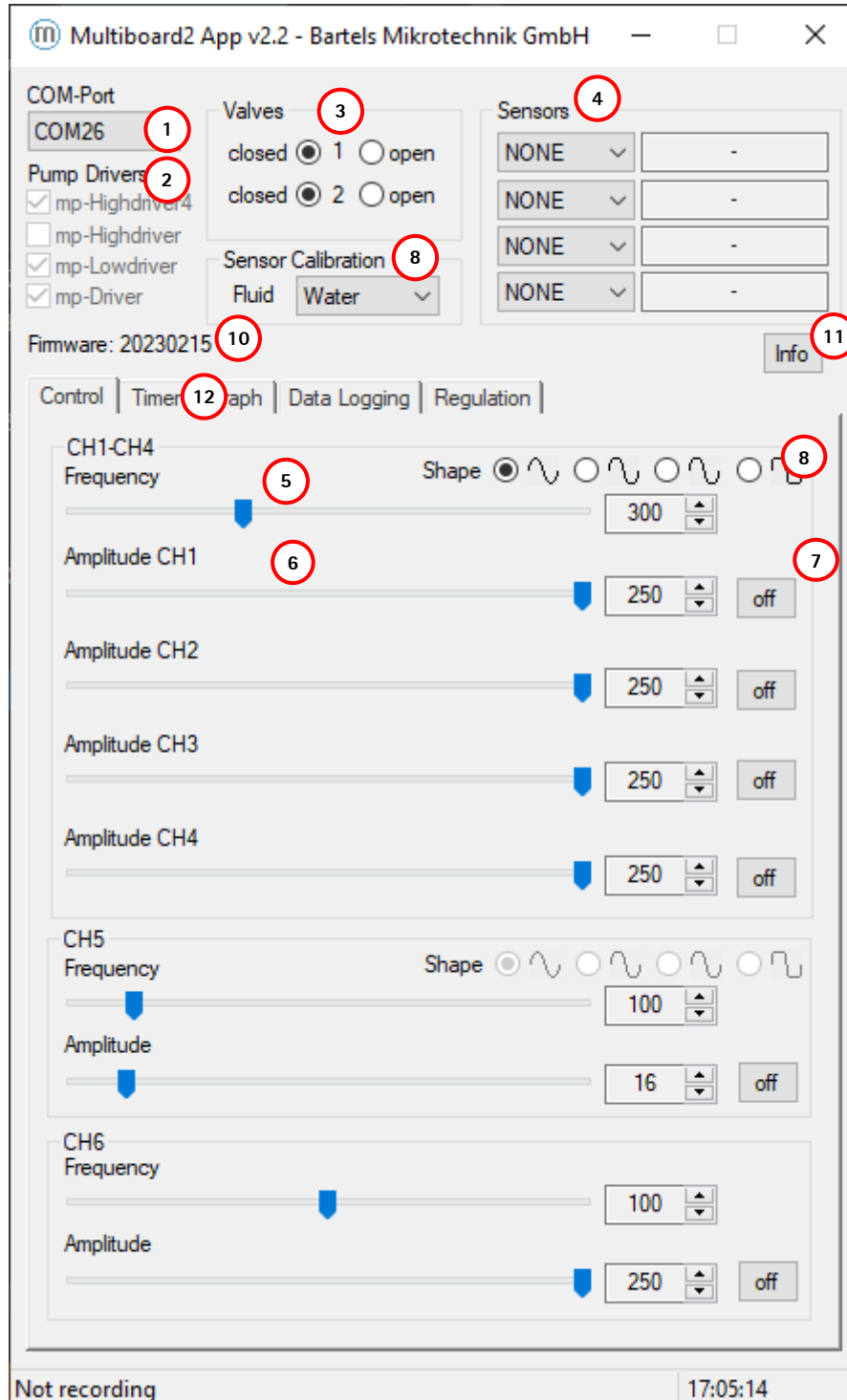


Figure 2 Multiboard App GUI

Select the COM-Port of your Multiboard2 in the list-box (1). Any plugged in pump driver will be automatically detected and will be checked underneath "Pump Drivers" (2). If a Highdriver4 is detected, CH1-CH4 will be enabled and sliders (6) can be used to adjust each pump's amplitude separately, switch every pump on and off (7) and change the frequency for all Highdriver4 pumps (CH1-CH4) globally (5). If a mp-Highdriver or mp-Lowdriver is detected,

then there will be CH5, or with a mp-Driver, CH6 will be enabled as well. There are four radio buttons (8) for the Highdriver platform to change the drivers signal shape for the pump. The selectable signal shapes are (left to right) full-sinusoid, 2:1 sinusoid:rectangular, 1:2 sinusoid:rectangular & full rectangular. For more details on the Highdrivers signal shape, refer to the electronics datasheet. The radio-buttons in the Valves-group (3) can be used to open and close active valves connected to the Multiboard, when a valve driver is plugged onto the Multiboard. The list-boxes in the Sensors-group (4) can be used to enable & disable sensor readout when sensors are plugged into the Multiboard. To change the calibration of the liquid sensor a fluid can be selected in the listbox (9). The Label (10) shows the current Firmware version on the Multiboard. The Info (11) will direct to overview with the companys information, an section to update the boards firmware and the unlocked software license.

By selecting the Timer-tab (12) you switch over to the Timer-mode user interface, showing settings to setup a timed on/off cycle of each pump:

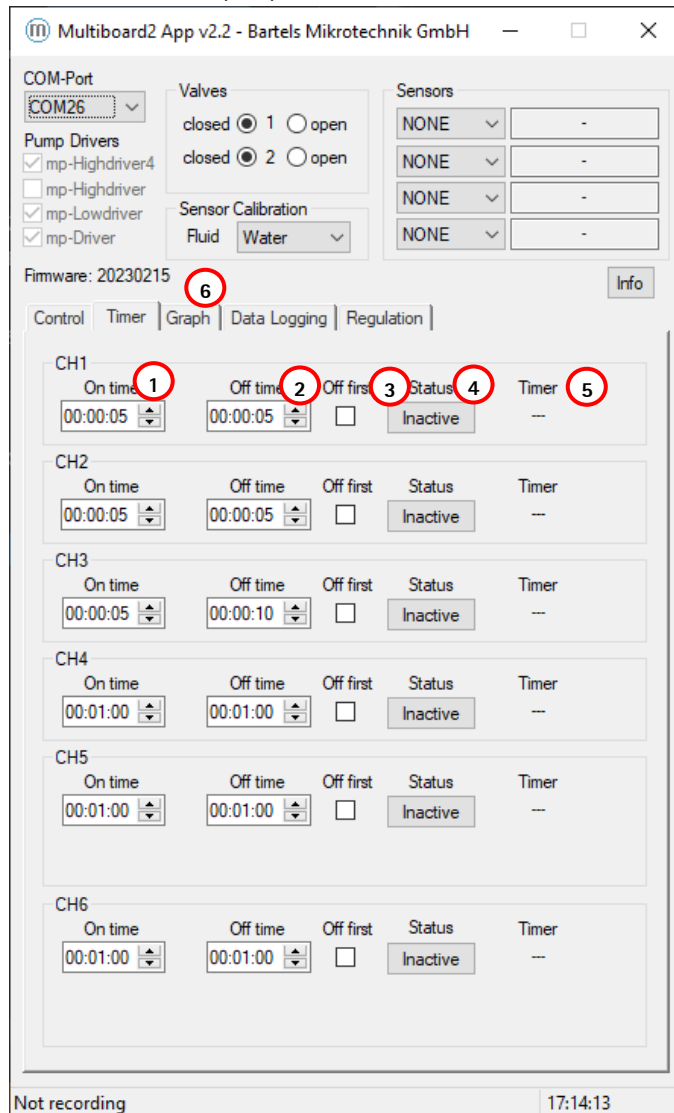


Figure 3 Multiboard App GUI in "Timer Mode"

The "On time" (1) defines how long the pump is to stay activated (pumping), while the "Off time" (2) defines how long the pump is to stay deactivated before starting a new cycle. When "Off first" (3) is ticked, the cycle starts with the pump deactivated. By pressing the "Status" button (4) this periodic pumping can be activated/deactivated for each pump. The indicator "Timer" (5) shows the current state of the pump (on/off) and the time remaining for this part of the cycle.

The on/off buttons on the Control-tab is disabled and greyed while timer mode is active. The frequency and amplitude controls are not disabled since they still take effect on the pumps in timer mode.

By selecting the Graph-tab (6) you switch over to a Graph showing the current flow & pressure sensor values of the attached (and enabled) sensors:

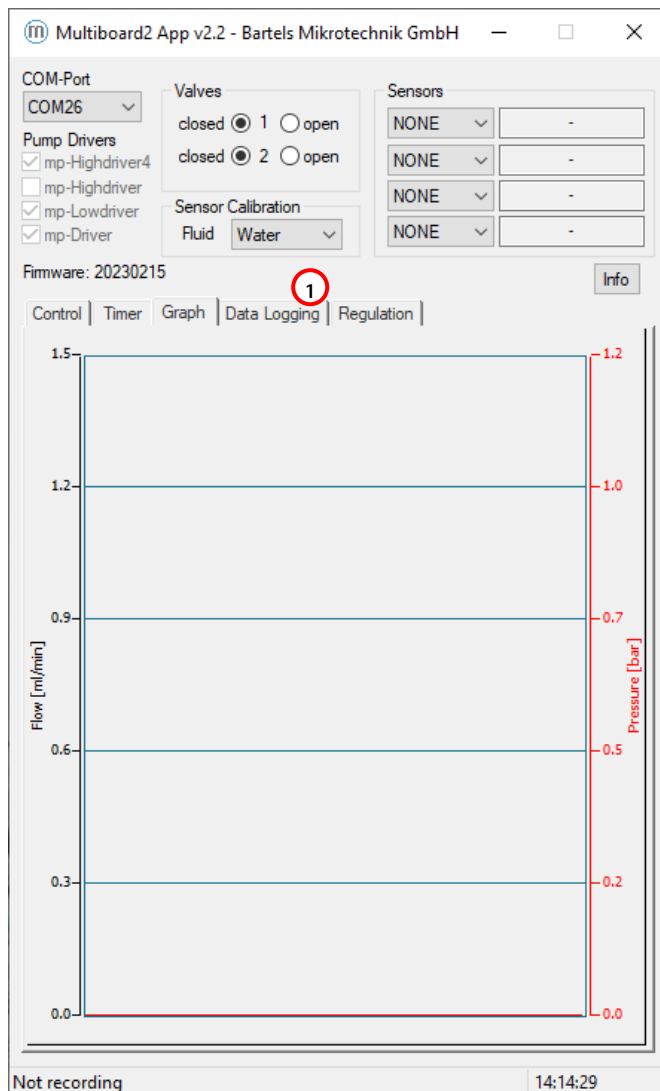


Figure 4 mp-Multiboard2 sensor time plot

The Data Logging-tab (1) allows the recording of your selected sensors.

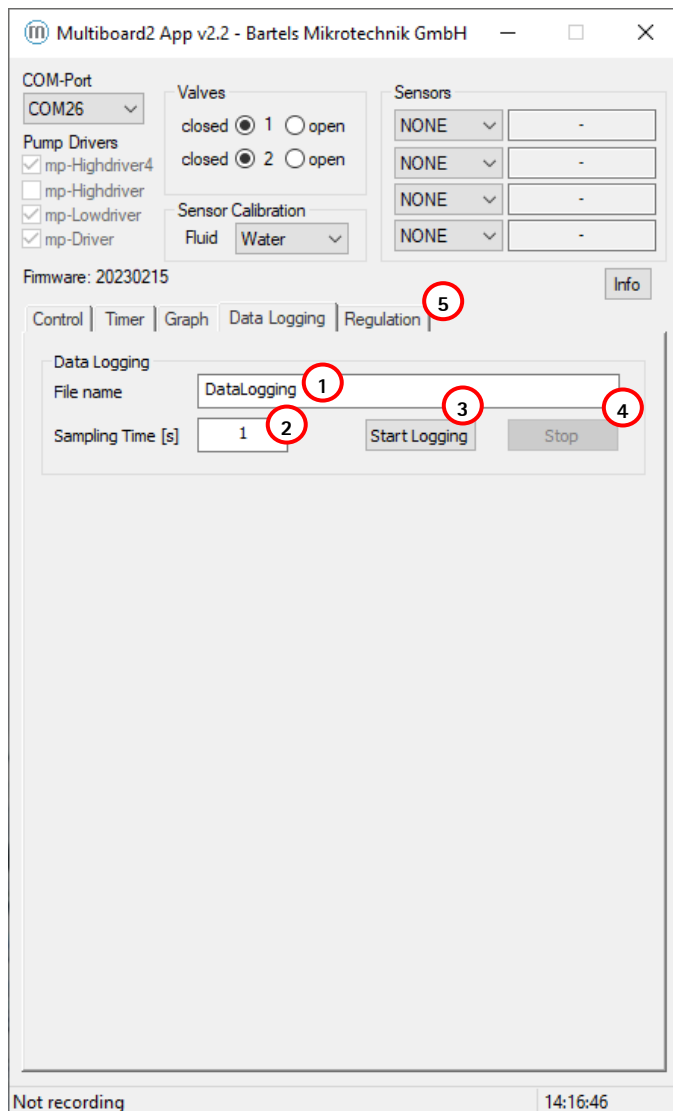


Figure 5 mp-Multiboard2 data logging tab

A file name can be set inside the dialog field (1) and the sampling time can be changed in in dialog field (2). The "Start Logging"-button begins the logging of the sensor data and the "Stop"-button ends the measurement. The file format will be a .txt that can be imported to excel if needed. After 32.000 lines of logged data the App start a new File with the same file name including an upcounting index number for file sequence. All .txt file are stored on the same data path the Multiboard2 App was executed from. On the Regulation tab (5) all control elements for flowrate related and regulated applications can be found.

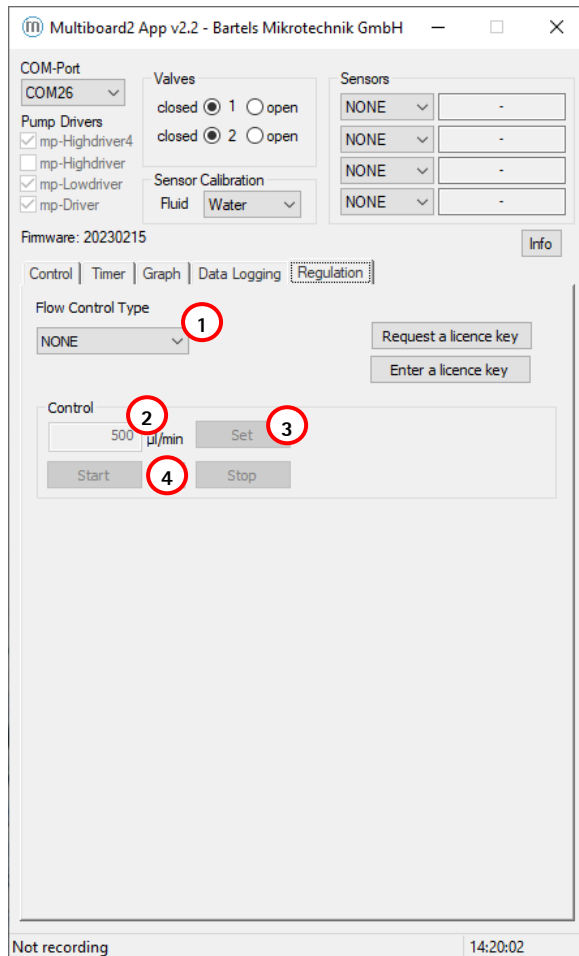


Figure 6 mp-Multiboard2 Regulation tab

A variety of different applications are listed inside the Flow Control Type drop down menu (1). The buttons in the Control section are enabled upon selecting a control type above. A target value can be entered (2) and confirmed by hitting the Enter key or by clicking the "Set" button (3). The pump can be manually started and stopped with the corresponding buttons (4).

2.1.3 Updating the mp-Multiboard2 Firmware

The Multiboard Firmware can be updated manually via the Multiboard App. The newest Firmware will be available on the Bartels website in the software download section:

<https://www.bartels-mikrotechnik.de/downloads/?eeFolder=EN/Software>

There will be a listed file named "mp-Multiboard2_Firmware_(FW-num).zip". After downloading the file needs to be unzipped. The unzipped folder looks as follows:

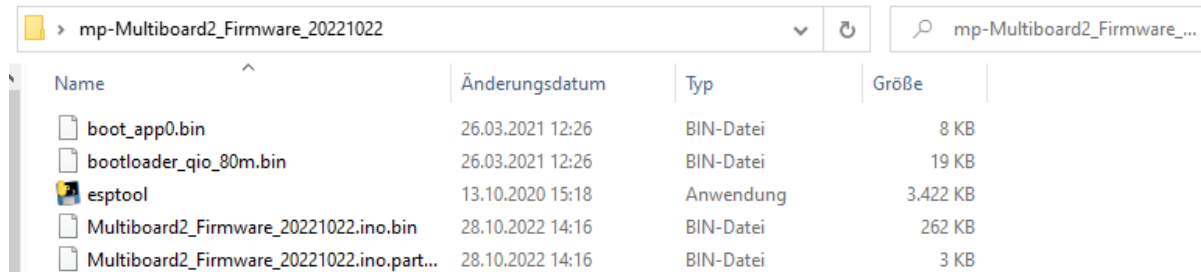


Figure 7 unzipped mp-Multiboard2 firmware folder

Now connect your mp-Multiboard2 to your PC via the provided USB cable and select the right COM Port. Click the "Info" button (1) and a new window will pop up. Upon clicking the "Update" button, an explorer window will open up asking you to provide the location of the previously unzipped folder. Select the Folder and confirm (3).

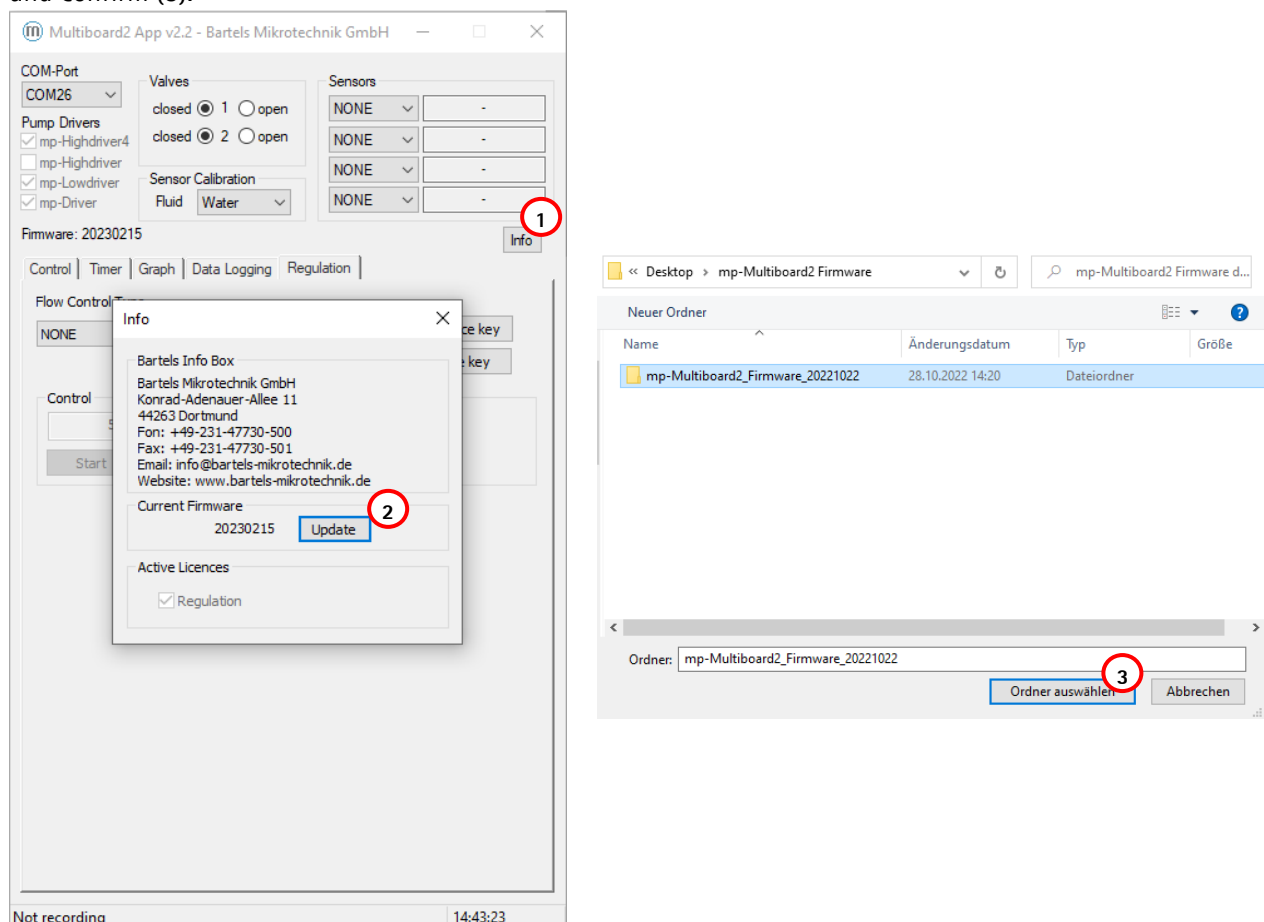


Figure 8 mp-Multiboard2 Info window

Afterwards a window will pop up showing the found file and asks for confirmation for a Firmware update.

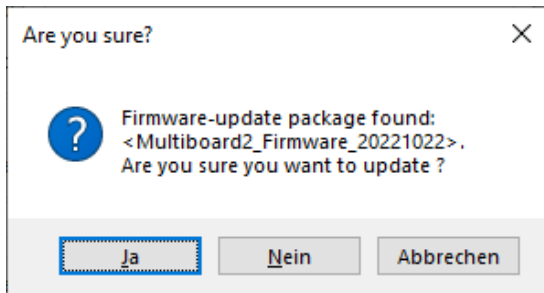


Figure 9 Firmware confirmation

A new window will open up for the upload of the Firmware. It will connect to the previously connected COM port will look like this:



Figure 10 Firmware upload window

To upload the Firmware a button (1) near the USB connector needs to be pushed and held until the upload is done. The window will close automatically and the mp-Multiboard2 is updated. The board will need to power down by unplugging the USB cable and any external power supply if in use and then reconnected.

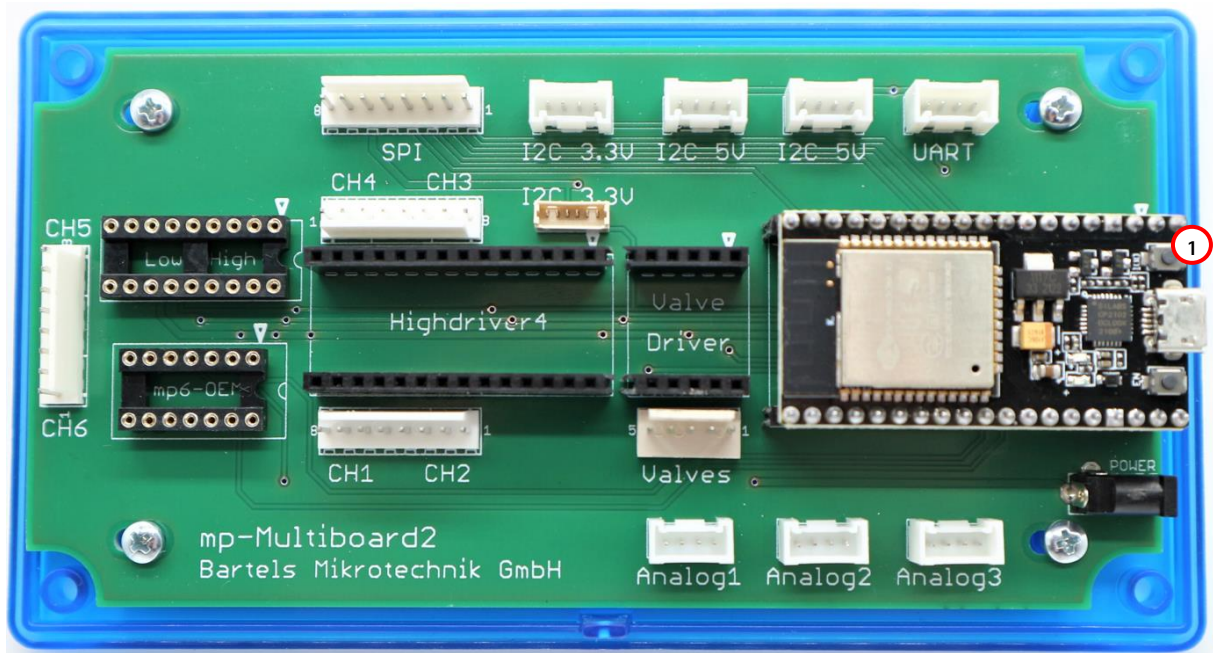


Figure 11 mp-Multiboard2 program button

2.1.4 Requesting new Features

The mp-Multiboard2 come with the option to unlock additional functions for this evaluation platform. The Regulation section is one of those and adds the possibility to have a mp6 micropump be regulated by an flow sensor.

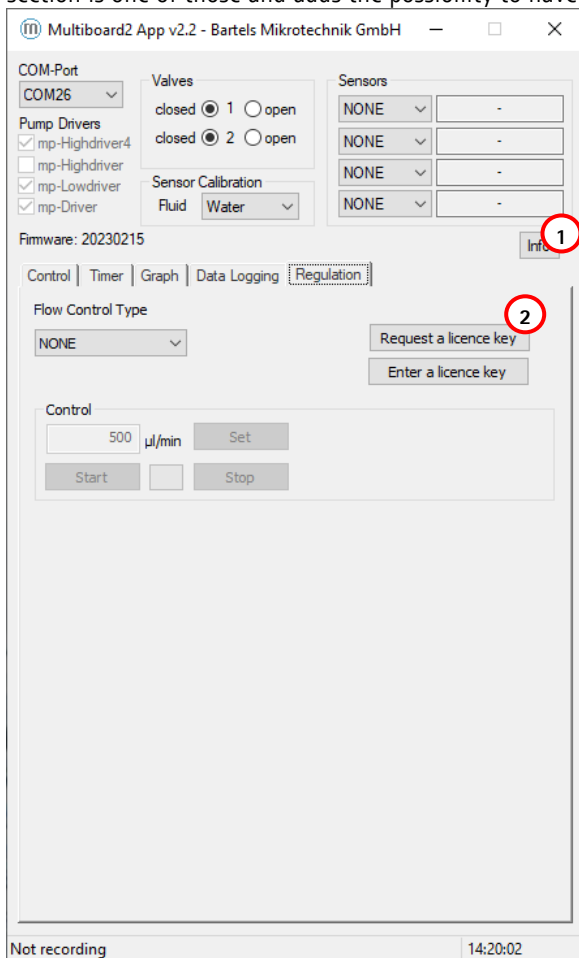


Figure 12 mp-Multiboard2 App Info Button

After starting the Multiboard2 App and have connected your board to your PC, select your boards COM port. When the board is recognized, select the Regulation tab (1) and click on the request licence key button.

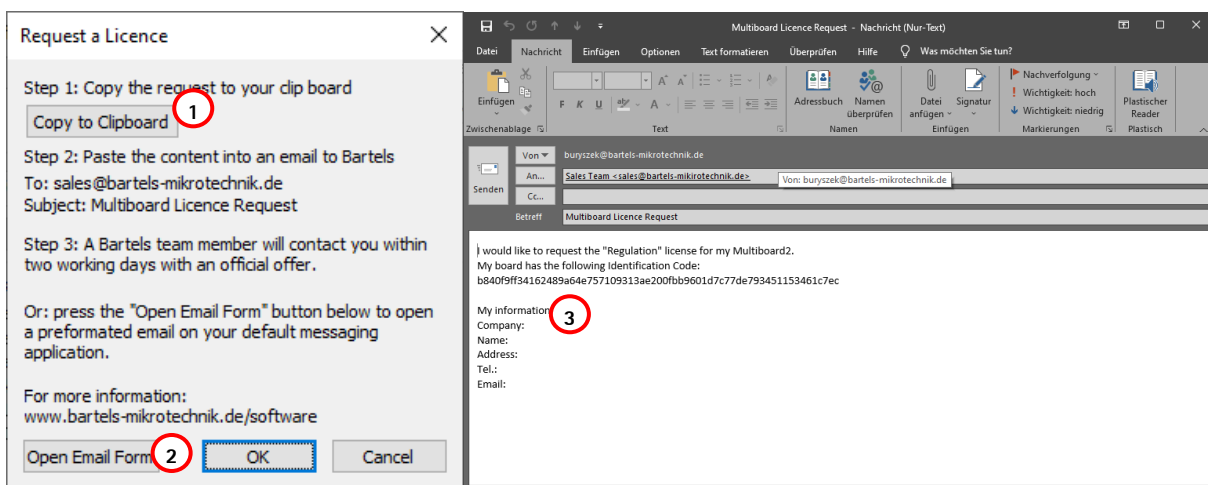


Figure 13 Requesting Email Form

A new window will open up with instruction on how to request a licence key. Either copy an automatically generated Email form to your clipboard (1) or directly open an Email Form (2) and send it to sales@bartels-mikrotechnik.de. For a faster processing of your request please fill out your information below (3).

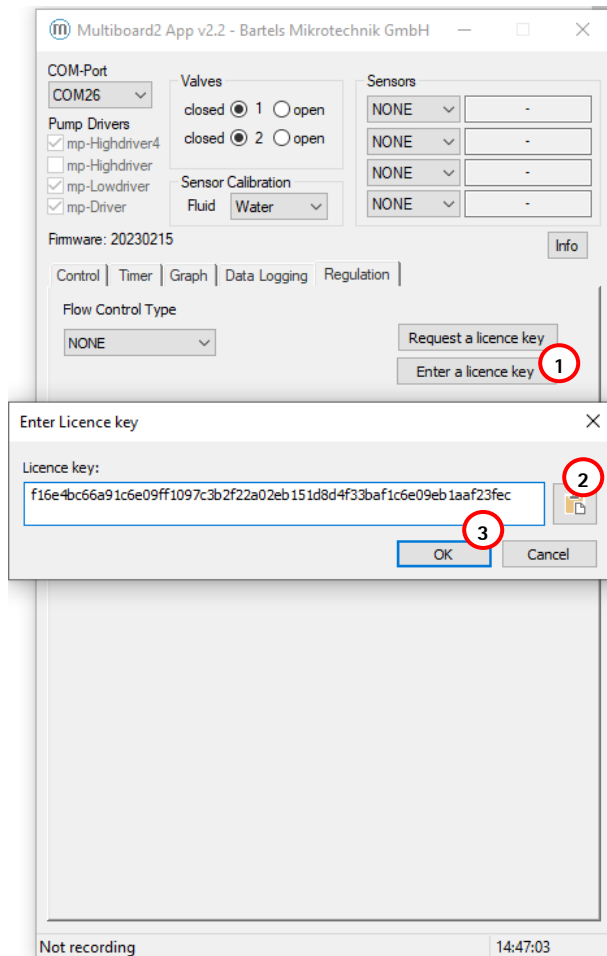


Figure 14 Entering licence key

After you received your licence key go back to the Regulation tab and click "Enter a licence key" (1). Another window will open. Paste your licence key inside the text box (2) and confirm your input by clicking the "Ok"-Button (3). The App will verify it and tell you, if it accepts it. Now it will show your active software packages inside the Info window and you will have access to your new features.

2.1.5 USB/Serial Communication Protocol

The Multiboard is connected to the computer via the USB connector of the ESP32. It acts as a USB-Serial interface so the Multiboard appears on the computer as a COM-Port inside the device manager. To communicate with the Multiboard any software capable of sending and receiving data to/from a serial port can be used. Programs like Hyperterminal and PuTTY can be used to send commands directly, while programs like LabVIEW, Matlab and Python can be used to write software that interacts with the user and the Multiboard allowing for more automation and ease of use. A list of commands used to control the Multiboard can be found in the table below:

PON	turns all pumps on
POFF	turns all pumps off
P<p>ON P1ON	Turns the selected pump <p> on Example: Turns pump #1 on
P<p>OFF P1OFF	Turns the selected pump <p> off Example: Turns pump #1 on
P<p>V<a> P1V250	Sets the amplitude for the selected pump <p> to the chosen value <a> Example: Pump #1 set to 250 Vpp
P<p>V? P1V?	Gets the amplitude for the selected pump <p> Example: Returns "250" meaning 250 Vpp for pump #1
F<d>=<f> F0=100	Sets the frequency for driver <d> (=0 for P1-4, =1 for P5 & =2 for OEMdriver) to the chosen value <f> Example: Sets the frequency for the mp-Highdriver4 to 100 Hz.
V1ON V2ON	Turns valve 1/2 on
V1OFF V2OFF	Turn valve 1/2 off
DPON DPOFF	Enables/disables pressure readout
DFON DFOFF	Enables/disables liquid flow readout
DGON DGOFF	Enables/disables gas flow readout
DA<a>ON DA<a>OFF	Enables/disables analog sensor readout at Analog<a> connector on board
L<l> L0	Change flowrate sensor calibration to <l> (0=water, 1=IPA) Example: Change to water calibration
V	Displays the current firmware version
(enter key)	displays current settings of the Multiboard

2.1.6 Example Codes for

This section will give you extracts from the mp-Multiboard2 Firmware for the individual components. These examples is based on Arduino covers the read out of sensors and valve drivers. For the pump drivers refer to the electronic datasheet or to the demo codes in the software download section on our website.

2.1.6.1 Pressure Sensor

```
#define I2C_ABP_ADRESS (0x78)    //Defaultadress for Honeywell ABP Pressure  
Sensor ABPMANV015PG7A5 0x78
```

```
bool startPressureMeasurement() {  
    uint8_t StatusByte;  
    Wire.requestFrom(I2C_ABP_ADRESS,1,true);    //Requests one databyte from  
sensor  
    StatusByte = Wire.read()>>6;  
    return (StatusByte == 0);  
}
```

```
void readPressure() {  
    uint16_t Pdata;  
    float scaled_pressure = 0;  
    uint8_t RawSensByte1 = 0;  
    uint8_t RawSensByte2 = 0;  
    uint8_t StatusByte;  
  
    Wire.requestFrom(I2C_ABP_ADRESS,2,true);    //Requests two databytes from  
sensor  
    RawSensByte1 = Wire.read();  
    RawSensByte2 = Wire.read();  
    StatusByte = RawSensByte1 >> 6;  
    Pdata = ( ((unsigned int)RawSensByte1) << 8 ) | RawSensByte2;  
    Pdata &= 0x3FFF;  
    scaled_pressure = ((float)Pdata) * 1.1443e-3f - 1.8744f; //conversion to  
psi  
    scaled_pressure *= 0.0689476f;    //conversion in bar  
}
```

2.1.6.2 Liquid Flow Sensor

```
#define I2C_SLF3S_ADRESS (0x08)    //Defaultadress for Sensirion SLF3S Flow-  
Sensor  
#define SCALE_FACTOR_1300 500.0f  
#define SCALE_FACTOR_0600 10000.0f  
#define PN_1300 0x07030200  
#define PN_0600 0x07030300
```

```
float SCALE_FACTOR_FLOW = 1.0;    //depending on connected sensor either  
=SCALE_FACTOR_1300 or SCALE_FACTOR_1300  
byte calibration_cmdByte = 0x08;    //default cmdByte for water measurement  
float scaled_flow_value;
```

```
bool startFlowMeasurement(void) {  
    uint32_t ProductNumber;  
    int ret;  
    if ((sensors & 0x02)>0) {    //Check if sensor isn't already running  
        //Stop continuous measurement  
        Wire.beginTransaction(I2C_SLF3S_ADRESS);  
        Wire.write(0x3F);  
        Wire.write(0xF9);  
        ret = Wire.endTransmission();  
    }  
    delay(100);
```

```
//Read Product Identifier and Serial number
Wire.beginTransaction(I2C_SLF3S_ADRESS);
Wire.write(0x36);
Wire.write(0x7C);
ret = Wire.endTransmission();

Wire.beginTransaction(I2C_SLF3S_ADRESS);
Wire.write(0xE1);
Wire.write(0x02);
ret = Wire.endTransmission();

Wire.requestFrom(I2C_SLF3S_ADRESS, 6);
delay(50);
ProductNumber = Wire.read();
ProductNumber = ProductNumber<<8 | Wire.read();
Wire.read(); //CRC
ProductNumber = ProductNumber<<8 | Wire.read();
ProductNumber = ProductNumber<<8 | Wire.read();
Wire.read(); //CRC
ret = Wire.endTransmission();

if ((ProductNumber & 0xFFFFF00) == PN_1300) {
    SCALE_FACTOR_FLOW = SCALE_FACTOR_1300;
} else if ((ProductNumber & 0xFFFFF00) == PN_0600) {
    SCALE_FACTOR_FLOW = SCALE_FACTOR_0600;
} else {
    SCALE_FACTOR_FLOW = 1;
    return false;
}
// Soft reset the sensor
Wire.beginTransaction(I2C_SLF3S_ADRESS);
Wire.write(0x06);
ret = Wire.endTransmission();
delay(100); // wait long enough for chip reset to complete

//Start continuous measurement
Wire.beginTransaction(I2C_SLF3S_ADRESS); //starting measurement
Wire.write(0x36); //Start continuous measurement
Wire.write(calibration_cmdByte); //Water calibration (0x15 for IPA)
ret = Wire.endTransmission();
return true;
}

void stopFlowMeasurement() {
    int ret;
    if ((sensors & 0x02)>0 || (sensors & 0x04)>0) {
        //Stop continuous measurement
        Wire.beginTransaction(I2C_SLF3S_ADRESS);
        Wire.write(0x3F);
        Wire.write(0xF9);
        ret = Wire.endTransmission();
        delay(100);
    }
}

bool startFlowMeasurement_simple(void) {
    // Soft reset the sensor
    Wire.beginTransaction(I2C_SLF3S_ADRESS);
    Wire.write(0x06);
    Wire.endTransmission();
    delay(50); // wait long enough for chip reset to complete

    //Start continuous measurement
```



```
Wire.beginTransaction(I2C_SLF3S_ADRESS); //starting measurement
Wire.write(0x36); //Start continuous measurement
Wire.write(calibration_cmdByte); //Water calibration (0x15 for IPA)
return (Wire.endTransmission()!=2); //0=success; 3 is already running; 2
is no response
}

void readFlow(void) {
    uint16_t sensor_flow_value;
    int16_t signed_flow_value;
    float scaled_flow_value;

    Wire.requestFrom(I2C_SLF3S_ADRESS, 3);
    sensor_flow_value = Wire.read() << 8; // read the MSB from the sensor
    sensor_flow_value |= Wire.read(); // read the LSB from the sensor
    Wire.read(); //CRC

    signed_flow_value = (int16_t) sensor_flow_value;
    scaled_flow_value = ((float) signed_flow_value) / SCALE_FACTOR_FLOW;
}
```

2.1.6.3 Thermal Conductivity Sensor

// Expansion of Sensirion Liquid Flow Sensor

```
void readConductivity(void) {
    uint16_t sensor_tc_value;
    uint8_t CRC1;
    uint16_t sensor_temp_value;
    uint8_t CRC2;
    uint16_t sensor_dt_value;
    uint8_t CRC3;
    int16_t signed_temp_value;
    float scaled_temp_value;
    int16_t signed_dt_value;
    float scaled_dt_value;

    Wire.requestFrom(I2C_SLF3S_ADRESS, 9);
    sensor_tc_value = Wire.read() << 8; // read the MSB from the sensor
    sensor_tc_value |= Wire.read(); // read the LSB from the sensor
    CRC1 = Wire.read();
    sensor_temp_value = Wire.read() << 8; // read the MSB from the sensor
    sensor_temp_value |= Wire.read(); // read the LSB from the sensor
    CRC2 = Wire.read();
    sensor_dt_value = Wire.read() << 8; // read the MSB from the sensor
    sensor_dt_value |= Wire.read(); // read the LSB from the sensor
    CRC3 = Wire.read();
    Wire.endTransmission();

    signed_temp_value = (int16_t) sensor_temp_value;
    scaled_temp_value = ((float) signed_temp_value) / 200.0f;
    signed_dt_value = (int16_t) sensor_dt_value;
    scaled_dt_value = ((float) signed_dt_value) / 1000.0f;
}

void Liquid_select(uint8_t _liquid) { //Only Sensirion Sensors for now
    switch(_liquid)
    {
        case 0:
            // set cmd byte for the SLF3S to water calibration
            calibration_cmdByte = 0x08;
            if ((sensors & 0x02) > 0){
```

```
        startFlowMeasurement();
        Serial_println("Switched to Water");
    }
    break;
case 1:
    // set cmd byte for the SLF3S to IPA calibration
    calibration_cmdByte = 0x15;
    if ((sensors & 0x02) > 0){
        startFlowMeasurement();
        Serial_println("Switched to IPA");
    }
    break;
default:
    break;
}
}
```

2.1.6.4 Analog Sensor

```
#define ANALOG_PIN1 39
#define ANALOG_PIN2 34
#define ANALOG_PIN3 35

int analog_sens_Value1 = 0;
int analog_sens_Value2 = 0;
int analog_sens_Value3 = 0;

void readAnalogSensor1(){
    analog_sens_Value1 = analogRead(ANALOG_PIN1);
}

void readAnalogSensor2(){
    analog_sens_Value2 = analogRead(ANALOG_PIN2);
}

void readAnalogSensor3(){
    analog_sens_Value3 = analogRead(ANALOG_PIN3);
}
```

2.1.6.5 Valvedriver

```
#define VALVEPIN1      33
#define VALVEPIN2      32

void Valves_init() {
    pinMode(VALVEPIN1, OUTPUT);
    pinMode(VALVEPIN2, OUTPUT);
    digitalWrite(VALVEPIN1, LOW);
    digitalWrite(VALVEPIN2, LOW);
}

void Valve_open(uint8_t _VALVEPIN) {
    pinMode(_VALVEPIN, OUTPUT);
    digitalWrite(_VALVEPIN, LOW);
}

void Valve_close(uint8_t _VALVEPIN) {
    pinMode(_VALVEPIN, INPUT);
}
```

2.2 mp-Labtronix controller



Access to the full range of driving parameters. A system for the professional evaluation of the micropumps.

2.2.1 Installation of the drivers "USB Micropump Control" and "USB Serial Port"

In order to control the mp-Labtronix with a computer, a driver needs to be installed. The installation is done in two steps. First, the USB driver is installed. Second, the driver for the mp-Labtronix itself is installed which can be used as a serial interface from different software packages.

For "Windows XP" and former Windows versions

- Step 1: Connect the control unit with the USB port and turn it on. A message appears that new hardware was found and the hardware assistant starts automatically. Please click "Next" to continue.
- Step 2: Choose "Find a suitable driver for the device" and click "Next" to continue.
- Step 3: Place the CD into the disc drive.
- Step 4: Choose "Search for the best driver in these locations" and tag "Include this location in the search". Then browse and select the CD.
- Step 5: When the software has found the driver named "USB Micropump Control" select "Next". The installation will start now. If the message appears that the Windows Logo Test was not successful, please select "Continue Anyway" as this has no relevance in this case. Click "Finish" to complete the installation.
- Step 6: After that installation, the hardware assistant pops up again for the serial port. Repeat the steps 1-5.

For "Windows 7"

Dependent on the individual system settings, different steps for installation might be necessary. The procedure applies in general also for Windows Vista.

- Step 1: Log in with administrator rights.
- Step 2: Connect the control unit with the USB port and turn it on. A message appears that new hardware was found.
- Step 3: Open the device manager and double click the item "USB Micropump control" filed under "additional hardware" (or similar). In the pop up window click on "update driver" and select the directory including the driver files (e.g. the cd rom drive). It needs to be confirmed that the driver should really be installed as it has no digital signature.
- Step 4: Repeat the steps 2 and 3 for the serial port. Starting with step 2 again, a device named "USB serial port" will be listed. The drivers need to be installed as described above.
- Step 5: The device manager will show the number of the serial port (e.g. COM 4). Note this for further use of the device with application programs.

For "Windows 8"

- Step 1: Log in with administrator rights.
- Step 2: From the Metro Start Screen, open Settings (move your mouse to the bottom-right-corner of the screen and wait for the pop-out bar to appear, then click the Gear icon).

Step 3: Click "More PC Settings".

Step 4: Click "General".

Step 5: Scroll down, and click 'Restart now' under "Advanced startup".

Step 6: Click "Troubleshoot".

Step 7: Click "Advanced Options".

Step 8: Click "Windows Startup Settings".

Step 9: Click "Restart".

Step 10: On the Startup Settings screen, tap the F7 or 7 key on your keyboard to select the "Disable driver signature enforcement" option. Wait until the system has restarted.


Step 11: Go to the mp-Labtronix driver folder "XU7 USB Driver 2.08.30" and click on "ftdibus Bami.inf", then click "Install". Wait until installation is finished.

Step 12: Go again to the mp-Labtronix driver folder "XU7 USB Driver 2.08.30". Click now "ftdiport Bami.inf", then "Install". Wait until installation is finished.

Step 13: Now you can connect the mp-Labtronix with any USB-port. If the mp-Labtronix was already connected, disconnect it and connect it again.

For "Windows 10"

Step 1: Log in with administrator rights.

Step 2: On the Start Menu, click in the Startup/Shutdown-Symbol () and while holding down the shift-key (arrow up) click on "Restart".

Step 3: Click "Troubleshoot".

Step 4: Click "Advanced Options".

Step 5: Click "Windows Startup Settings".

Step 6: Click "Restart".

Step 7: On the Startup Settings screen, tap the F7 or 7 key on your keyboard to select the "Disable driver signature enforcement" option. Wait until the system has restarted.

Step 8: Go to the mp-Labtronix driver folder "XU7 USB Driver 2.08.30" and click on "ftdibus Bami.inf", then click "Install". Wait until installation is finished.

Step 9: Go again to the mp-Labtronix driver folder "XU7 USB Driver 2.08.30". Click now "ftdiport Bami.inf", then "Install". Wait until installation is finished.

Step 10: Now you can connect the mp-Labtronix with any USB-port. If the mp-Labtronix was already connected, disconnect it and connect it again.

2.2.2 Operation via USB port (after installation of the drivers)

The driving parameters can be set via your PC. For this purpose you can use software (or programming language) of your choice that is capable of sending commands to serial COM-Ports. In the example below the Windows software Hyperterminal is used. As the Hyperterminal is not available in Windows 7 anymore, we recommend freeware terminal software like PuTTY (<http://www.putty.org>).

Please note that the mp-Labtronix requires a time of approx. 100 ms to process a command. Therefore real-time flow control is not possible. Regardless of the time for processing a command, every change (amplitude, frequency or signal form) results in a jump in flow.

Hyperterminal example:

Step 1: Connect the control unit to your computer and turn it on.

Step 2: Start Windows Hyperterminal. Every new session has to be titled.

Step 3: Choose the COM-port specified in the device manager.

Step 4: The connection-settings have to be:

Bits per second: 9600; Data bits: 8; Parity: none; Stop bits: 1; Flow control: none

2.2.3 Possible commands (followed by the enter key)

bon	turns the micropump on
boff	turns the micropump off
F(1-300) F100	sets the required frequency between 1 and 300 Hz here as an example 100 Hz.
A(1-250) A100	sets the required frequency between 1 and 250 Vpp here as an example 100 Vpp
MS	sets signal form modus (S)ine
MR	sets signal form modus (R)ectangular
MC	sets signal form modus SRS
(enter key)	displays present settings of the control unit

The control unit can also be used via LabView, Matlab or other programs. Using LabView, please ensure that the "NI-Serial" package is installed. This will be installed in regular cases together with LabView, but sometimes this option is skipped during install.

It can be downloaded on the National Instruments website as the "NI-Serial" package.

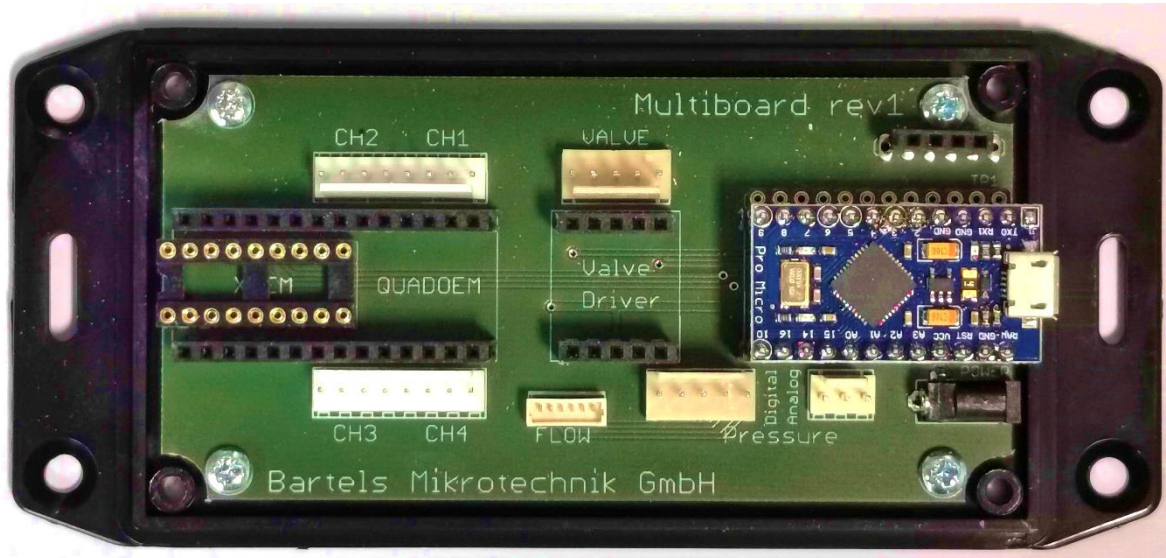
Afterwards it is possible to select the right COM-port for your mp-Labtronix in the VISA resource name.
On request, we can send a package of LabView-Routines as an example of implementation.

We also offer to develop customer specific software for the application of the controller units.

3 Discontinued Products

3.1 Evaluation Board: mp-Multiboard

3.1.1 Auxiliary connectors



The Multiboard has multiple auxiliary connectors available. Some of these are not in use at the moment and will be documented in this manual once fully implemented. Nevertheless, here is a list of the connectors and their (future) use:

CH2 / CH1, CH3 / CH4	Two 8-pin connectors for pump cable harness. Each cable harness has two FFC connectors on the other end to connect to the micropump flex cable
Valve	5-pin connector for two active valves
TP1	12-pin header for custom purposes
Flow	6-pin connector for an I ² C flow-sensor
Pressure digital	5-pin connector for an I ² C pressure-sensor
Pressure analog	3-pin connector for an analog pressure-sensor
Power	Barrel power supply connector

3.1.2 USB-driver

The most current USB-driver is commonly installed with the Arduino IDE. If you don't want to install the whole IDE and just need the driver, you can download the IDE including the driver as ZIP archive from the Arduino download site (<https://www.arduino.cc/en/software>) and install just the USB driver.

A driver-package is also included in the Multiboard App download.

3.1.3 Multiboard App

The Multiboard App is a Software tool that enables you to control various features of the Multiboard remotely via USB. It interfaces with the Arduino via the USB-Serial interface and calls the protocol commands of the Firmware. A description of the protocol and list of commands will be available in the future, allowing to use the Multiboard using other software like LabVIEW, Matlab and python.

After executing the app, the following screen opens up:

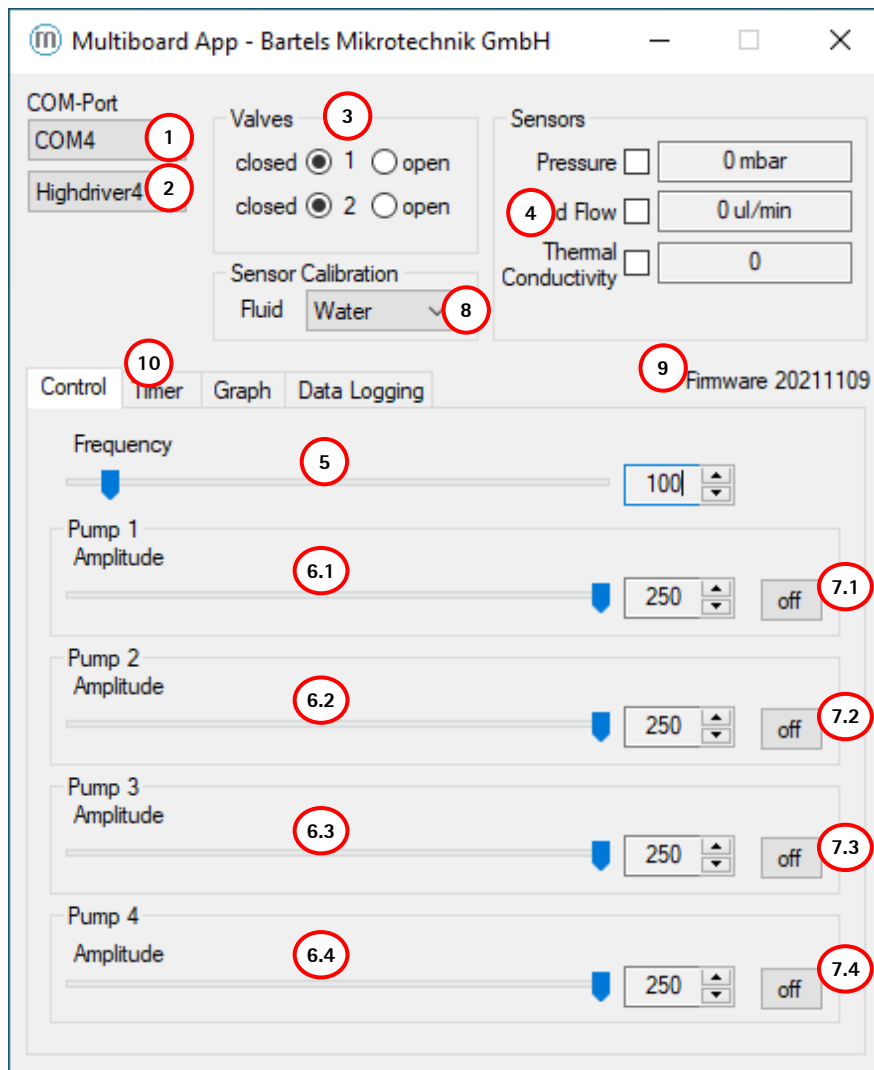


Figure 15: Multiboard App GUI

Select the COM-Port of your Multiboard in the list-box (1), followed by the pump driver currently inserted on the Multiboard in the list-box below (2).

Now you are able to adjust each pump's amplitude separately (6.1 - 6.4), switch every pump on and off (7.1 - 7.4) and change the frequency for all pumps globally (5).

The radio-buttons in the Valves-group (3) can be used to open and close active valves connected to the Multiboard, when a valve driver is plugged onto the Multiboard.

The checkboxes in the Sensors-group (4) can be used to enable & disable sensor readout when sensors are plugged into the Multiboard.

To change the calibration of the liquid sensor a fluid can be selected in the listbox (8).

The Label (9) shows the current Firmware version on the Multiboard. Always check if newer Software is available on our Website to get the latest features of the Multiboard unlocked. The App and Firmware should be both up to date to make sure, there are no compatibility issues.

By selecting the Timer-tab (10) you switch over to the Timer-mode user interface, showing settings to setup a timed on/off cycle of each pump:

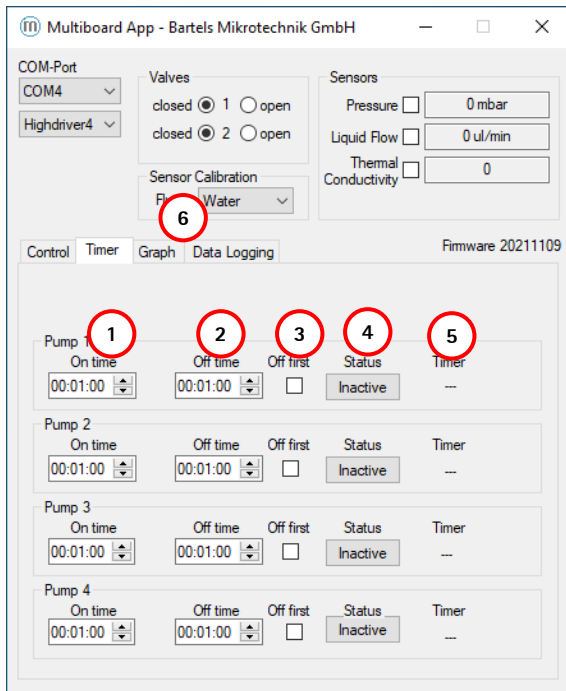
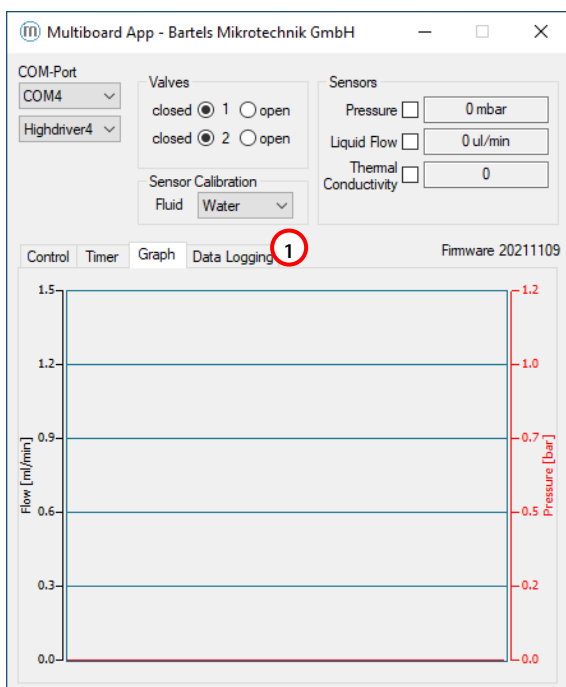


Figure 16: Multiboard App GUI in "Timer mode"

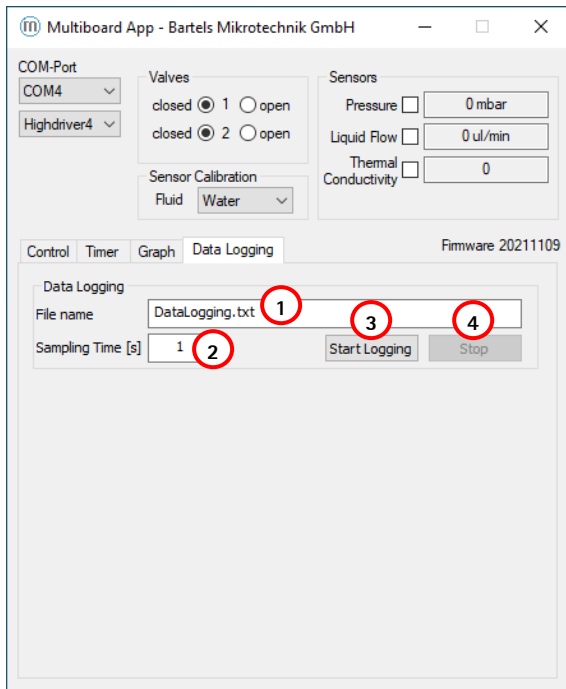
The "On time" (1) defines how long the pump is to stay activated (pumping), while the "Off time" (2) defines how long the pump is to stay deactivated before starting a new cycle. When "Off first" (3) is ticked, the cycle starts with the pump deactivated. By pressing the "Status" button (4) this periodic pumping can be activated/deactivated for each pump. The indicator "Timer" (5) shows the current state of the pump (on/off) and the time remaining for this part of the cycle.

The on/off button on the Control-tab is disabled and greyed while timer mode is active. The frequency and amplitude controls are not disabled since they still take effect on the pumps in timer mode.

By selecting the Graph-tab (6) you switch over to a Graph showing the current sensor values of the attached (and enabled sensors):



The Data Logging-tab (1) allows the recording of your selected sensors.



A file name can be set inside the dialog field (1) and the sampling time can be changed in in dialog field (2). The "Start Logging"-button begins the logging of the sensor data and the "Stop"-button ends the measurement. The file format will be a .txt that can be easily imported to excel if needed.

3.1.4 USB/Serial Communication Protocol

The Multiboard is connected to the computer via the USB connector of the Arduino pro micro. It acts as a USB-Serial interface so the Multiboard appears on the computer as a COM-Port inside the device manager. To communicate with the Multiboard any software capable of sending and receiving data to/from a serial port can be used. Programs like Hyperterminal and PuTTY can be used to send commands directly, while programs like LabVIEW, Matlab and Python can be used to write software that interacts with the user and the Multiboard allowing for more automation and ease of use. A list of commands used to control the Multiboard can be found in the table below:

SELECTLOWDRIVER SELECTHIGHDRIVER SELECTQUADDRIVER SELECTNONE	Tells the Multiboard which driver is currently on the board and initializes the driver
PON	turns all pumps on
POFF	turns all pumps off
P<p>ON P1ON	Turns the selected pump <p> on Example: Turns pump #1 on
P<p>OFF P1OFF	Turns the selected pump <p> off Example: Turns pump #1 on
P<p>V<a> P1V250	Sets the amplitude for the selected pump <p> to the chosen value <a> Example: Pump #1 set to 250 Vpp
P<p>V? P1V?	Gets the amplitude for the selected pump <p> Example: Returns "250" meaning 250 Vpp for pump #1
F<f> F100	Sets the frequency (for all pumps) to the chosen value <f> Example: Sets the frequency to 100 Hz.
V1ON V2ON	Turns valve 1/2 on
V1OFF V2OFF	Turn valve 1/2 off

DPON DPOFF	Enables/disables pressure readout
DFON DFOFF	Enables/disables flow readout
MC	sets signal form modus SRS
V	Displays the current firmware version
(enter key)	displays current settings of the Multiboard

4 Important Notices

This operating manual contains necessary instructions for the installation, commissioning, operation and maintenance of the mp6-series. The manual is intended to help you achieving optimal results in a short time and shall also assist avoiding possible sources of errors. The operating manual of the controllers and the accessories are available separately.

The products have been designed with state-of-the-art technology and in accordance with all relevant safety regulations. However, a risk of damage to the units, other property, the operator and/or other persons cannot be fully excluded.

Always ensure that specialized and trained personnel will comply with the following general instructions. Therefore, please keep this manual and hand out copies as required.

All values are approximate and no guarantee of specific technical properties.
Changes in the course of technical progress are possible without notice.

4.1 Warranty

The mp6 micropumps have been developed for the transport of gases or liquids. The controllers have been developed for operating the mp6 micropumps. Bartels Mikrotechnik can assume no liability for damages resulting from the pump media. This applies especially for hazardous fluids.

The mp6 micropumps must be operated with Bartels Mikrotechnik electronics. Bartels Mikrotechnik GmbH cannot guarantee the proper work of the units with customer specific electronics. If other controllers than the ones from Bartels Mikrotechnik are used, Bartels Mikrotechnik disclaims any warranty.

Bartels Mikrotechnik assumes no liability for abnormal handling, improper or negligent use of the mp6 micropumps and the controller that is not conform to the specified purpose of the system. This applies especially for micropump controllers, components and systems of other manufacturers, which have not been certified by Bartels Mikrotechnik.

We guarantee that the mp6 micropumps comply with the actual state of scientific and technical knowledge and due to this, the operational risks are limited to a minimum.

Bartels Mikrotechnik GmbH warrants solely to the original purchaser of this product for a period of 12 months (one year) from the date of delivery that this product shall be of the quality, material and workmanship defined in Bartels Mikrotechnik GmbH published specifications of the product. Within such period, if proven to be defective, Bartels Mikrotechnik GmbH shall repair and/or replace this product, in Bartels Mikrotechnik GmbH's discretion, free of charge to the Buyer, provided that:

- notice in writing describing the defects shall be given to Bartels Mikrotechnik GmbH within fourteen (14) days after their appearance

- such defects shall be found, to Bartels Mikrotechnik GmbH's reasonable satisfaction, to have arisen from Bartels Mikrotechnik GmbH's faulty design, material or workmanship
- the defective product shall be returned to Bartels Mikrotechnik's factory at the Buyer's expense
- the warranty period for any repaired or replaced product shall be limited to the unexpired portion of the original period.

This warranty does not apply to any equipment which has not been installed and used within the specifications recommended by Bartels Mikrotechnik GmbH for the intended and proper use of the equipment.

EXCEPT FOR THE WARRANTIES EXPRESSLY SET FORTH HEREIN, Bartels Mikrotechnik GmbH MAKES NO WARRANTIES, EITHER EXPRESS OR IMPLIED, WITH RESPECT TO THE PRODUCT. ANY AND ALL WARRANTIES, INCLUDING WITHOUT LIMITATION, WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE, ARE EXPRESSLY EXCLUDED AND DECLINED. Bartels Mikrotechnik GmbH is only liable for defects of this product arising under the conditions of operation provided for in the data sheet and proper use of the goods. Bartels Mikrotechnik GmbH explicitly disclaims all warranties, express or implied, for any period during which the goods are operated or stored not in accordance with the technical specifications. Bartels Mikrotechnik GmbH does not assume any liability arising out of any application or use of any product or circuit and specifically disclaims any and all liability, including without limitation consequential or incidental damages. All operating parameters, including without limitation recommended parameters, must be validated for each customer's applications by customer's technical experts. Recommended parameters can and do vary in different applications. Bartels Mikrotechnik GmbH reserves the right, without further notice, (i) to change the product specifications and/or the information in this document and (ii) to improve reliability, functions and design of this product.

4.2 Warning, Personal Injury

Bartels Mikrotechnik GmbH rejects any responsibility for damages to persons or property resulting from non-compliance with the instructions in this manual. In this case all warranties shall be void.

Do not use this product as safety or emergency stop devices or in any other application where failure of the product could result in personal injury. Do not use this product for applications other than its intended and authorized use. Before installing, handling, using or servicing this product, please consult the data sheet and application notes. Failure to comply with these instructions could result in death or serious injury. If the Buyer shall purchase or use Bartels Mikrotechnik GmbH products for any unintended or unauthorized application, Buyer shall defend, indemnify and hold harmless Bartels Mikrotechnik GmbH and its officers, employees, subsidiaries, affiliates and distributors against all claims, costs, damages and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if Bartels Mikrotechnik GmbH shall be allegedly negligent with respect to the design or the manufacture of the product.

Moreover, please note that components of the controller and pump are operating with high voltage. Therefore, persons wearing pacemakers are recommended to avoid the operating system. Do not open the housing of the micropump and the controllers.

The use of liquids, which may alone or in combination create explosive or otherwise health-endangering conditions (including vapors) is not permitted.

All work in connection with the installation, assembly, commissioning/decommissioning, disassembly, operation, servicing, cleaning and repairing of the pump and the controller must be carried out by qualified, suitably trained and instructed personnel. Work on electrical components and assemblies must be carried out by personnel with the necessary qualifications and skills.

We guarantee that the micropumps comply with the actual state of scientific and technical knowledge and due to this, the operational risks are limited to a minimum.

ESD Precautions: The inherent design of the active electronic components, i.e. mp-Lowdriver, -Highdriver, -Highdriver4, Valvedriver and -Labtronix causes it to be sensitive to electrostatic discharge (ESD). To prevent ESD-induced damage and/or degradation, take customary and statutory ESD precautions when handling this product.

4.3 Declaration of conformity

Bartels Mikrotechnik GmbH declares that the products are compliant to the RoHS directive 2011/65/EU. The controllers comply with the requirements of EMV 2014/30/EU and CE markings have been affixed to the devices. Additionally, the controllers are also compliant to the EU Low Voltage Directive 2014/35/EU.

5 Company information



Bartels Mikrotechnik is a globally active manufacturer and development service provider in the field of microfluidics. In the microEngineering division, the company supports industrial customers in the modification, adaptation and new development of high-performance and market-oriented product solutions through the innovative means of microsystems technology. The second division, microComponents, produces and distributes microfluidic products and systems, especially for miniaturized and portable applications. Our key products are micropumps that convey smallest quantities of gases or liquids and are used in a variety of ways in biotechnology, pharmaceuticals, medical technology and numerous other applications.

Bartels Mikrotechnik with passion for microfluidics!

Contact us:

Bartels Mikrotechnik GmbH
Konrad-Adenauer-Allee 11
44263 Dortmund Germany

www.bartels-mikrotechnik.de
info@bartels-mikrotechnik.de
Tel: +49-231-47730-500

Follow Bartels:

