

 Advanced
Microfluidics
AMF Control Center

Instrument Control and Product Automation Software

User Guide V1.1

Applies to AMF Control Center V1.1.1 – 06 October 2026

Contents

1. Introduction.....	4
1.1 Application Overview	4
1.2 Global Application Workflow	4
1.3 Prerequisites	4
1.4 Installation, activation and uninstallation	4
1.5 Trial and full version	5
1.6 Network and security	6
2. General Concepts	7
2.1 Communication Overview	7
2.2 Broadcast Behavior	7
2.3 Sequential vs Non-Sequential Communication	7
2.4 Send, Delay, Clear, General GO and STOP	8
2.5 Values, units and rounding	8
3. Instrument View	10
3.1 Purpose of the Instrument View	10
3.2 COM Port Selection and Detection	11
3.3 Documentation Access	12
3.4 Logs	12
3.5 Error Messages	13
3.6 Product Card	16
3.7 Instrument Settings	17
3.8 Firmware Update.....	19
3.9 Valve Visual	19
3.10 Syringe Control	20
3.11 Syringe Visual	21
3.12 Setting a position by dragging the plunger	21
3.13 On-the-Fly Behavior.....	21
3.14 Custom Commands	23
3.15 Presets.....	23
4. Sequence View	25
4.1 Purpose of the Sequence View	25
4.2 Mini Product Card Visualization	25
4.3 Default Unit selection	26
4.4 Sequence Line Presentation	26
4.5 Action Types.....	28
4.6 Pull and Wait	29
4.7 Adding, Skipping, Duplicating, and Deleting Sequence Lines	29

4.8 Starting and Stopping a Sequence	30
4.9 Sequence Options	31
4.10 Build: checking a sequence before running it	31
4.11 Following the running line	33
4.12 Log panel of the Sequence view	34
4.13 Example sequences	34
4.14 Export to Python.....	34
4.15 Keyboard Shortcuts	36
5. Troubleshooting	37
5.1 COM Port Not Detected	37
5.2 Instrument Does Not Respond	37
5.3 Broadcast Does Not Behave as Expected	37
5.4 Firmware Update Fails or Is Refused	37
5.5 Commands Are Not Sent or not Delayed	38
5.6 Sequence Does Not Start	38
5.7 Sequence Stops Unexpectedly	38
5.8 Pump card greyed out	39
5.9 Application freezes or a port cannot be closed	39
5.10 Logs and Support Diagnosis	39
6. FAQ	40
7. Glossary	42
Appendix A. Example sequences	43
A.1 Where the examples are	43
A.2 Before you run an example	43
A.3 Example 1: Synchronised start	44
A.4 Example 2: Continuous perfusion.....	45
A.5 Example 3: Reference sequence	47
A.6 Writing your own sequence with an AI assistant	49
A.7 Rules to remember	50

1. Introduction

1.1 Application Overview

AMF Control Center is a desktop instrument control and sequencing application for AMF instruments. The application is built around two primary views:

- **Instrument view**, for detecting connected instruments, getting a products overview, setting instrument parameters, sending commands, updating firmware, and reviewing logs.
- **Sequence view**, for building, checking and running step-based instrument workflows.

The current software supports AMF pump products (LSPone, SPM) and AMF rotary valve products (RVMLP, RVMFS, RVMmini). It enables quick and smooth operations without integration knowledge.

1.2 Global Application Workflow

A typical workflow is:

1. Open AMF Control Center.
2. Select one or more COM ports.
3. Click **DETECT & INIT**.
4. Check the instrument configuration in each product **Settings** panel, or load an existing preset.
5. Use the Instrument view for manual valve and pump control.
6. Open the Sequence view to automate a workflow for valves and pumps: write or load the sequence, click **Build** to check it, then **Play**.

1.3 Prerequisites

Area	Requirement or Status
Operating system	Windows 10 or 11, 64-bit (x64). The setup refuses to run on a 32-bit Windows. Windows on ARM is not supported (not tested).
Installation rights	None for the application itself: it installs per user, see section 1.4. The FTDI USB driver needs administrator rights: on a computer where it is missing, Windows asks for an administrator confirmation (UAC prompt) during the setup.
Serial access	The user must have permission to access the selected COM ports.
Drivers	The FTDI USB driver is included in the setup and installed when it is not found on the computer (Windows asks for an administrator confirmation; nothing is shown on a computer that already has it). If the confirmation is refused, the application is installed anyway and the driver can be installed later: on most computers Windows Update installs it automatically when the first instrument is plugged in, otherwise download it from ftdichip.com (section 5.1).
Instrument setup	Instruments must be connected, powered, and visible as COM ports. The application drives up to six instruments at a time (slots 1 to 6), on one or several COM ports; if more products are detected, only the first six are connected and a warning is written in the log.
RS485 setup	RS485 detection scans product addresses 1 to 9 and A to E. Instruments on the same RS485 bus must have distinct, valid addresses.
Firmware requirements	Keep the firmware up to date (available on the AMF website) to access all functionalities. Pumps require firmware L1.1.1 or newer: with an older firmware the product card is locked except the Settings panel, from which the firmware can be updated (section 3.7).
Display	The window adapts to the screen: minimum 1280 × 720 logical pixels. On laptops with Windows scaling at 125 % or 150 % the interface is scaled down automatically.

1.4 Installation, activation and uninstallation

Installation. Run the setup file `AMF_Control1_Center_Setup_<version>.exe`. The installer:

- installs the application for the current user only, in %LOCALAPPDATA%\Programs\AMF\AMF_Control_Center. No administrator rights are needed and the application is not in Program Files;
- installs the FTDI USB driver when it is not found on the computer: Windows then shows an **administrator confirmation (UAC)** for the driver installer, accept it. Nothing is shown when the driver is already there. If the confirmation is refused or no administrator account is available, a message says so and the installation of the application completes normally (driver: section 5.1);
- creates a Start menu shortcut and, if selected, a desktop shortcut;
- asks for the **activation key** on its second page: enter it and click **Next**, or leave the field empty and click **Next** to install the **trial version** (section 1.5); the key can be entered later from the application.

First start. At the very first launch the application unpacks its runtime: the window may take a few seconds to appear (a “Preparing AMF Control Center for its first launch...” screen is shown if it takes longer). The following launches are immediate.

First start before any instrument was plugged in. Windows completes the FTDI driver installation the first time an AMF instrument is connected by USB. Until then the log terminal shows, in orange, “[FTDI] FTDI D2XX driver not found (ftd2xx.dll): product serial numbers cannot be read until an AMF instrument has been plugged in once on this computer, then restart the application.” The application works normally (detection, commands, sequences); only the product serial numbers are unavailable. Plug an instrument in, wait for Windows to finish installing the device, then restart the application: the message disappears.

User data. Everything the user produces lives in %APPDATA%\AMF\AMF_Control_Center: logs\ (log files), Preset\ (instrument presets), Sequences\ (sequence files, including the three example sequences copied there at the first start) and license.txt (activation key). This folder survives updates and uninstallation.

Update. Install the new version over the previous one: settings, presets, sequences, logs and the activation key are kept. If the activation key page is left empty during an update, the existing key is preserved.

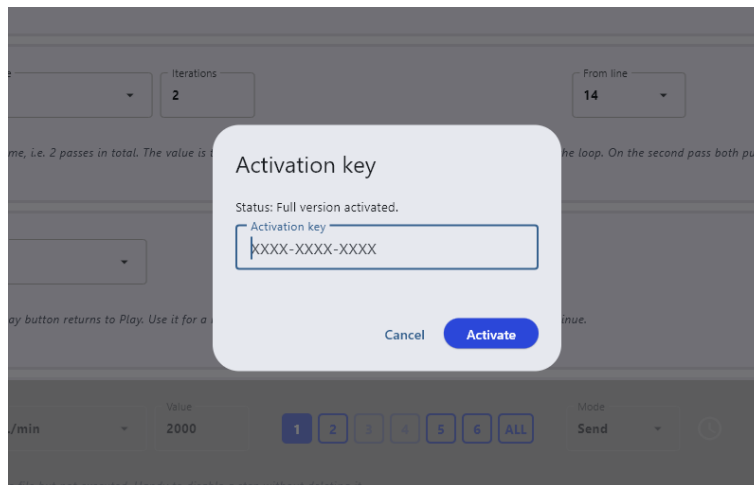
Uninstallation. Use Windows Settings > Apps > AMF Control Center, or the Start menu entry. The user data folder %APPDATA%\AMF\AMF_Control_Center is **not** removed: delete it by hand if presets, sequences and logs are no longer needed.

1.5 Trial and full version

Without an activation key the application runs as a **trial version**:

- only **one** instrument can be connected: selecting a second COM port shows “Trial Version: Limited to 1 instrument connection.” in the log terminal;
- the Sequence view is locked: its tab reads **Sequence (🔒 Trial)**.

To activate the full version, open the ? menu (top right) and choose **Activation key**. The dialog shows the current status, accepts the key (case and spaces are ignored) and activates the full version immediately: the Sequence tab unlocks without restarting the application. A wrong key shows “Invalid activation key.” and nothing is written. The key is stored in license.txt in the user data folder.



The Activation key dialog (? menu > Activation key).

1.6 Network and security

AMF Control Center does not open any outgoing network connection: no telemetry, no update check, no licence server. Communication with the instruments goes through the serial (COM) ports only. The application uses a local socket between its interface and its Python runtime on the computer itself; a personal firewall may ask to allow it once. The links of the ? menu (User Guide, Documentations & Downloads, AMF website) open the default web browser only when clicked.

2. General Concepts

2.1 Communication Overview

The application exposes COM port selection to the user. It does not provide a separate user-facing protocol dropdown. Communication mode is determined during detection by the AMF communication layer.

Communication Mode	Typical Use	Multi-Instrument Support	Broadcast Behavior	Notes
USB	Direct serial-over-USB connection to an instrument.	Supported by selecting multiple COM ports when each instrument appears on its own port.	All-target operations are sent individually (sequentially) when multiple COM ports are used.	USB and RS232 are detected together as USB/RS232. Firmware updates require this connection.
RS232	Serial communication through an RS232 or serial adapter.	Supported by selecting multiple COM ports when each instrument appears on its own port.	All-target operations are sent individually (sequentially) when multiple COM ports are used.	The adapter serial number may be shown instead of the instrument serial number.
RS485	Multi-drop serial communication on a shared bus (requires a custom cable).	Supported. Each instrument on the bus must have a unique address 1 to 9 or A to E.	If multiple instruments are found on one selected COM port, the app creates an RS485 broadcast target using address _.	RS485 broadcast mode does not return normal product answers. The adapter serial number may be shown instead of the instrument serial number.

2.2 Broadcast Behavior

Broadcast behavior depends on how instruments are connected and detected.

Detected Setup	Internal Behavior	User Consequence
Multiple instruments on one COM port	The app creates a broadcast product target using RS485 broadcast address _.	All-target commands can be sent once to the shared bus. Instruments start from the same broadcast command, with no delay between them.
Instruments across multiple COM ports	The app sends the command sequentially to each connected product.	All-target commands are queued for each connected product individually.
Sequence All-target command using μ L units	The app sends the command sequentially to connected pumps so each pump can use its own syringe size.	This avoids using one global conversion for pumps with different syringe sizes.

Warning: RS485 broadcast mode does not return normal product answers. The application still monitors individual products after commands are sent, but broadcast responses themselves should not be expected.

2.3 Sequential vs Non-Sequential Communication

- Commands are queued sequentially by the application.
- Each command is sent with non-blocking behavior, so an instrument can continue moving while the application sends or monitors other commands.
- On one RS485 bus, a broadcast command can address multiple instruments at once and consequently make them start their motion simultaneously.

- Across multiple COM ports, an all-target operation sends one command per connected product. These commands are issued one after another by the queue, but instruments can run concurrently after receiving their commands.
- Sequence execution usually advances after a command is sent, unless the line uses Pull and Wait or a wait-style action.

Practical consequence: when a precise simultaneous start is important, prefer the RS485 broadcast setup where multiple instruments are detected on the same COM port, or use **Delay** followed by **General GO**. Across multiple COM ports, commands are sent individually and may start with small timing differences (around 100 ms).

2.4 Send, Delay, Clear, General GO and STOP

The application uses several related controls:

Control	Meaning
Send	Send the configured command immediately.
Delay	Load the command in the product without executing it. The product status turns yellow ("command pending") and the Delay button becomes Clear . Each product holds one pending command at most : a second Delay replaces the first, and a Send cancels it. A General GO is required to start the execution of a delayed command.
Clear	Clear the pending command of that product (sends cØR).
General GO	Execute all pending commands at once (sends R to all products, one broadcast frame on an RS485 bus).
STOP	Hard stop one instrument (τ). A pending command is cancelled.
General STOP	Hard stop all instruments and pause a running sequence.

Delay + General GO is the way to start several pumps at the same instant, whatever the wiring: load a move in each pump with Delay, then release them all with General GO.

2.5 Values, units and rounding

The pump only understands integers: a speed in **u** (firmware speed unit) and a position in **motor steps** (3000 steps per stroke in Standard resolution, 24000 in High resolution). When a value is entered in µL/min, mL/min or µL, the application converts it into the pump unit, **rounds it to the nearest integer** and sends that integer.

- **The field always keeps what you typed.** The application never rewrites your value.
- The value really applied is written in the log terminal when it differs from the entry, for example "Pump 1: speed 500 µL/min -> u3355 = 499.93 µL/min", and always in the log file. The "actual: ..." labels of the product card show the value read back from the pump.
- Decimals: two decimals are accepted in µL/min and µL, four in mL/min; V, u and steps values are rounded to the nearest integer. Only digits, dot and comma can be typed.
- Min/max bounds are unchanged: they are shown in the field tooltip and the value turns red when out of range.

Formulas (reducer = 1 on the standard series, 13.5 on the HD variant, read from the pump):

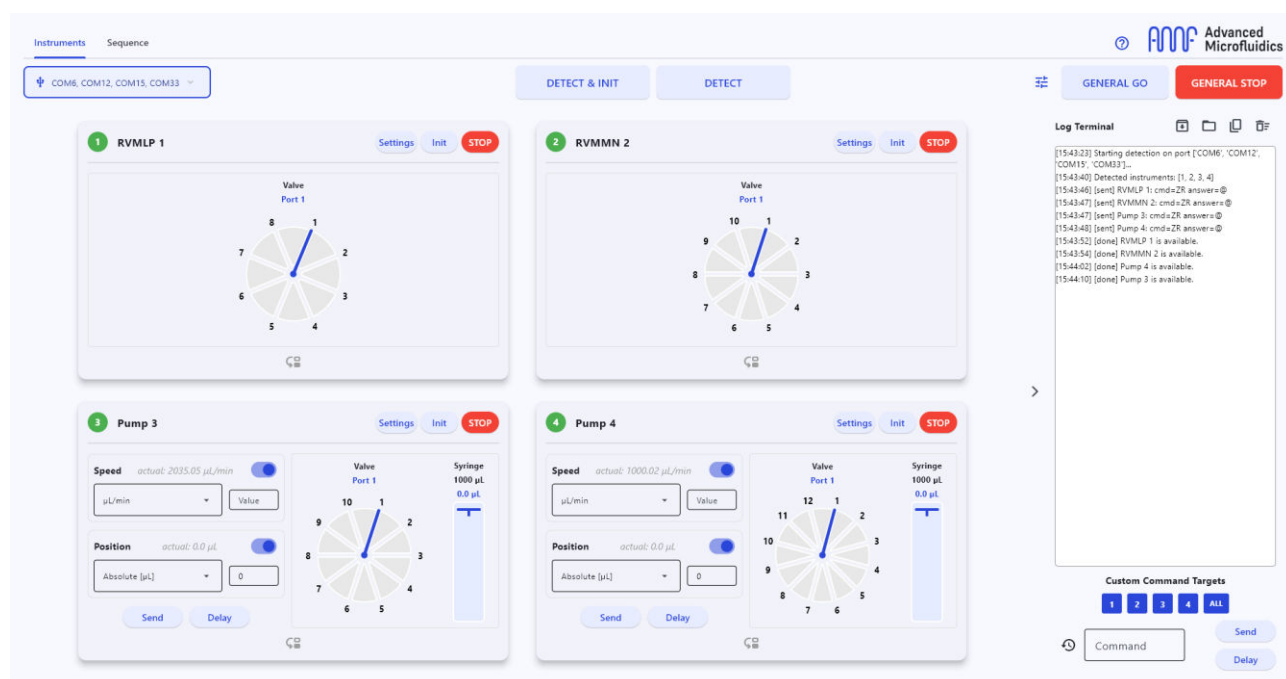
- Flow rate: $u = \text{round}(\mu\text{L}/\text{min} \times 3000 \times 134.217728 \times \text{reducer} / (60 \times \text{syringe } \mu\text{L}))$, i.e. $1 u = \text{syringe} / (6710.89 \times \text{reducer}) \mu\text{L}/\text{min}$.
- Although the speed limits are respectively V1600 and V500 for standard and HD pumps, we recommend using maximum speeds of V1500 and V400.
- Volume: $\text{steps} = \text{round}(\mu\text{L} \times \text{resolution} / \text{syringe } \mu\text{L})$, i.e. $1 \text{ step} = \text{syringe} / \text{resolution } \mu\text{L}$.

Smallest representable increment, by syringe size and pump model:

Syringe (µL)	1 u, standard (µL/min)	Range, standard (µL/min)	1 u, HD (µL/min)	Range, HD (µL/min)	1 step, 3000 (µL)	1 step, 24000 (µL)
25	0.0037	0.0037 – 800	0.00028	0.0036 – 250	0.0083	0.0010
50	0.0075	0.0075 – 1600	0.00055	0.0072 – 500	0.0167	0.0021
100	0.0149	0.0149 – 3200	0.0011	0.0143 – 1000	0.0333	0.0042
250	0.0373	0.0373 – 8000	0.0028	0.0359 – 2500	0.0833	0.0104
500	0.0745	0.0745 – 16000	0.0055	0.0717 – 5000	0.1667	0.0208
1000	0.149	0.149 – 32000	0.011	0.143 – 10000	0.3333	0.0417
2500	0.373	0.373 – 80000	0.0276	0.359 – 25000	0.8333	0.1042
5000	0.745	0.745 – 160000	0.0552	0.717 – 50000	1.6667	0.2083

Examples: 1000 µL syringe, 500 µL/min → u3355 → 499.93 µL/min; 5000 µL/min → u33554 → 4999.94 µL/min. 50 µL syringe, 100 µL/min → u13422 → 100.00 µL/min. Volume: 333.33 µL on a 1000 µL syringe in Standard resolution → 1000 steps → 333.33 µL; 0.1 µL on a 50 µL syringe in High resolution → 48 steps → 0.1 µL. The difference is always smaller than half of the “1 u” or “1 step” value of the table: the larger the syringe, the more visible it is.

3. Instrument View



The Instrument view with two pumps, an RVM low-power valve and an RVMMini.

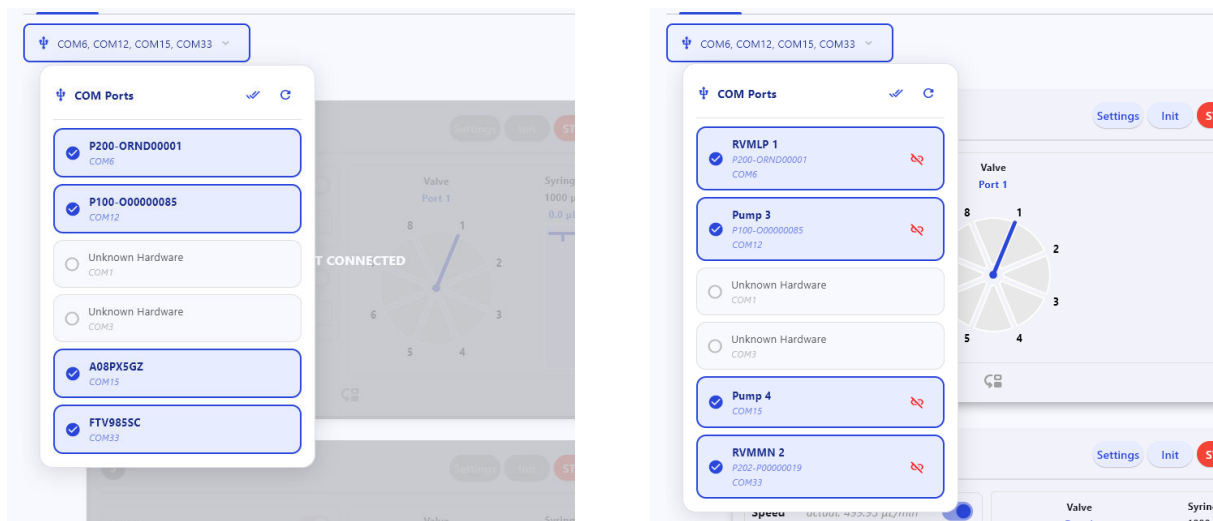
3.1 Purpose of the Instrument View

Use the Instrument view to detect and operate instruments directly. This view supports:

- COM port selection, refresh and per-port disconnection.
- Instrument detection and initialization.
- Product cards for connected instruments, with the product serial number.
- Manual pump speed and position commands, including setting a position by dragging the plunger.
- Valve visualization and port selection.
- Syringe position visualization.
- Product Settings panels.
- Card reordering by drag and drop.
- Firmware update.
- Product initialization and hard stop.
- General GO and General STOP.
- Custom command entry and command target selection.
- Log terminal, support report export, log copy and visible log clearing.
- Preset save, load, and latest preset loading.

3.2 COM Port Selection and Detection

The COM port selector appears in the top-left area of the Instrument view.



Port panel before detection: USB serial number and COM port.

After detection: product name, serial number and COM port, with the close-port button.

- The application automatically refreshes the COM port list when the software starts. The refresh icon of the panel reloads it at any time.
- The selector shows available COM ports as selectable port pills. Before connection a pill shows the USB serial number of the port (or “Unknown Hardware”) and the COM port name.
- The **double check** icon selects or deselects every listed port at once.
- If a single COM port is detected, it is automatically selected. If multiple COM ports are detected, the user must manually select targets.
- **DETECT & INIT** and **DETECT** are disabled until at least one COM port is selected.
- Clicking **DETECT** resets the existing product state, disconnects current products, clears target selections, and starts detection on the selected COM port list. Products are detected but no move command is sent.
- Clicking **DETECT & INIT** also homes all the products (ZR): valves and plungers move to their reference position. A product that is not initialized cannot execute a move command.
- Once products are connected, each pill shows the product name(s), the serial number(s) when they can be read, and the COM port. A red “**broken link**” button on the pill closes that port without closing the application: its products are disconnected (cards turn grey), the port is released for another program, and the port is unchecked. Detect again to reconnect them.



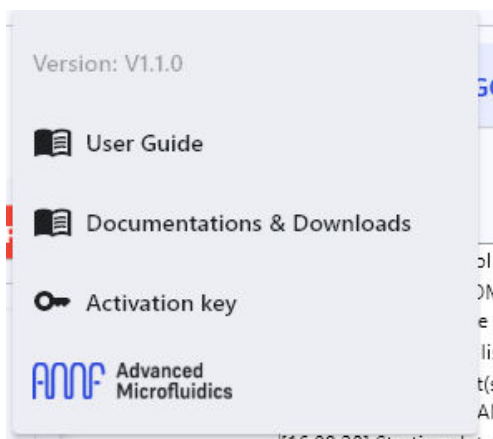
The **DETECT & INIT** and **DETECT** buttons, enabled once a port is selected.

Common connection failure causes include:

- The instrument is not powered.
- USB or serial cable is disconnected.
- The driver is missing.
- The COM port is already open in another application (including another instance of AMF Control Center).
- User permissions do not allow serial port access.
- The wrong adapter or communication mode is connected.

3.3 Documentation Access

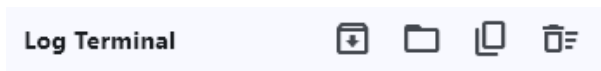
The ? button at the top right opens the help menu with:



- the software version;
- **User Guide**: this document (online);
- **Documentations & Downloads**: product manuals and firmware downloads on the AMF website;
- **Activation key**: enter or check the activation key (section 1.5);
- the AMF logo, which opens the AMF website.

3.4 Logs

The log terminal on the right of the Instrument view shows the recent application events (last 60 lines, with colors: orange for warnings, red for errors, green for a successful firmware update). Its header offers four actions:



Icon	Action
Export	Export a support report (.zip) : saves the current log file and the recent files in one archive to send to AMF support.
Folder	Open the log folder in Windows Explorer.
Copy	Copy the full session log to the clipboard (everything since the application started, not only the 60 visible lines).
Clear	Clear the terminal : the log file is kept.

- Visible log entries are timestamped with the local time.
- The arrow between the product cards and the terminal collapses or expands the log and command panel (keyboard: **Ctrl + Tab**).
- The application writes one log file per launch in %APPDATA%\AMF\AMF Control Center\logs, named AMFCC_<date>_<time>_proc<process>.log. The file contains more than the terminal: every serial command sent and its answer, every user action, the value conversions and full error traces. Up to 30 files of 10 MB are kept, the oldest are deleted.
- The Sequence view has its own log panel, closed by default, that opens automatically on an error (section 4.12).

3.5 Error Messages

Messages shown in the log terminal, with their meaning and the recommended action.

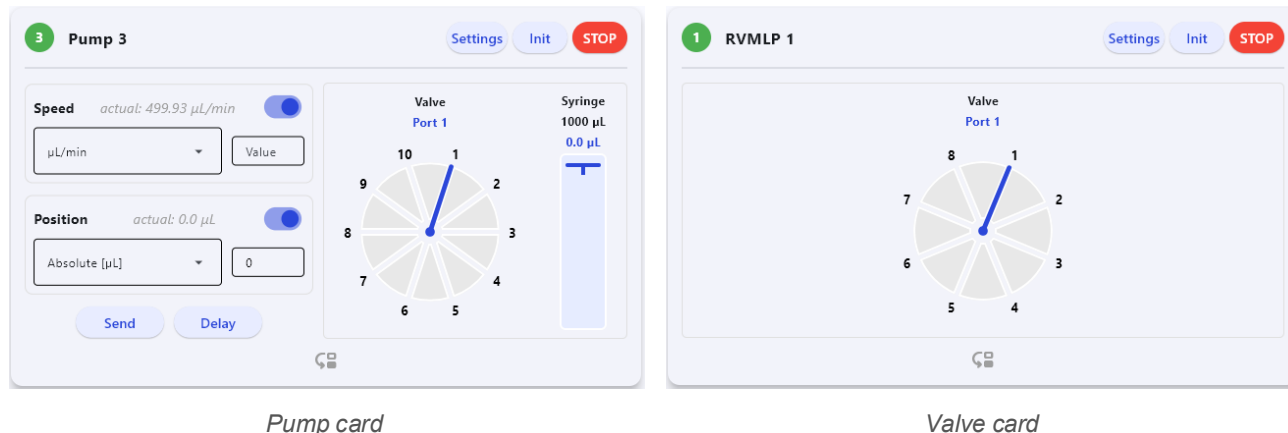
Error Message	Meaning	Likely Cause	Recommended Action
No COM ports detected. Please connect an instrument and refresh.	The app did not find serial ports.	Device unplugged, driver missing, adapter not recognized.	Check power, cable, driver, adapter, and click refresh.
No COM port specified. Please select a COM port and try again.	Detection was requested without a selected COM port.	No COM port selected.	Open the selector and select at least one port.
No instruments detected. Please check the connections and try again.	A COM port was scanned but no AMF instrument was found.	Wrong port, unpowered device, wrong communication mode, address issue, port held by another program.	Check cabling, power, RS485 address, close other programs using the port, and retry detection.
Error during instrument detection: ...	Detection failed with an exception.	Serial error, unavailable port, device communication issue.	Check the detailed log message and port availability.
Error during product initialization: ...	A detected product could not be initialized.	Port permission issue, stale connection, instrument not responding.	Close other serial tools, reconnect, and run detection again.
<name>: valve with N ports has no visual available. A 12-port wheel is shown; ports the valve does not have will answer "Invalid operand".	The valve reports a port count without a dedicated visual (3, 7...).	Non-standard valve.	Nothing to do: the valve is fully controllable. Only click ports that exist on the valve (section 3.9).
[FTDI] FTDI D2XX driver not found (ftd2xx.dll): product serial numbers cannot be read until an AMF instrument has been plugged in once on this computer, then restart the application.	Shown at start-up, in orange, when the FTDI driver library is not yet on the computer. Everything works except the reading of product serial numbers ("Not readable" in Settings and in the port panel).	No AMF instrument has been connected by USB to this computer yet, or the FTDI driver was not installed (administrator confirmation refused during the setup).	Plug an instrument in, let Windows finish the device installation, restart the application. If the message stays, install the FTDI driver with administrator rights (section 5.1).
[DETECTION WARNING] N products detected, only the first 6 are supported (slots 1 to 6).	More than six products answered on the selected ports; the extra ones are ignored.	Too many instruments on the selected COM ports.	Select fewer ports, or split the instruments between two sessions of the application.
Trial Version: Limited to 1 instrument connection.	A second COM port was selected in trial mode.	No activation key.	Enter the key in ? > Activation key (section 1.5).
Product not connected	A command targeted a missing or disconnected product.	Product was disconnected, its port was closed, or the internal target was not available.	Re-detect instruments and confirm target selection.
<name>: firmware Lx.y.z is older than Firmware-UFNG-L1.1.1. Update the pump firmware to use this software ...	The pump firmware is too old for this software.	Pump not updated.	The card is locked except Settings: update the firmware from the Settings panel (section 3.8).
<name>: syringe size not defined on the product (read as ...). Choose it in Settings ...	The pump does not store its syringe size (old firmware) or the read failed.	Firmware older than L1.0.76, or communication error.	Choose the syringe size in Settings: µL/min, mL/min and µL modes are disabled until then.
<name>: reduction ratio read as ..., using 1. µL/min and mL/min speeds may be wrong ...	The pump reducer could not be read; 1 is assumed.	Communication error during detection.	Check the connection and open Settings to read the product again, or re-detect.
Product is currently busy with a command. Please wait for it to finish before Flashing.	Firmware update was attempted while a product was busy.	Active command or motion in progress.	Wait for the product to finish or stop it safely before updating firmware.
[FIRMWARE UPDATE REFUSED] <name>: Firmware update is only possible through the product's own USB cable. Connect it directly by USB, run DETECT and retry.	The product is behind a serial adapter or cable (RS485, RS232) or shares its port with other products; nothing was sent to it.	Wiring.	Connect the product directly with its USB cable, detect it and retry (section 3.8). The log file names the port and the adapter.
Firmware update command refused. Use the firmware update button in the product Settings.	@FWUPDATER was typed as a custom command.	Manual bootloader command.	Use the update icon of the Settings panel, which checks the connection first.
No connected product selected.	A custom command was sent with no active target.	No command target selected or no connected products.	Select one or more target boxes.

Speed mode not selected.	Pump speed command is incomplete.	Speed switch enabled but no speed mode selected.	Select V, u, µL/min, or mL/min.
Speed ignored for <name> because speed value is empty.	The Speed switch is on but the value field is empty.	Only the position was meant to be sent.	Enter a speed or switch Speed off.
Write timeout on port ... / No answer from the product on port ... Check that the product is properly connected	The application could not write to, or get an answer from, the serial device.	Cable, port, driver, or device communication problem.	Check the physical connection, restart the device if needed, and re-detect.
[product DISCONNECTED] <name>: Disconnected	Repeated communication errors on that product.	Device stopped responding or was unplugged.	Check power and cable, then re-detect.
[ERROR] Command loop stalled for n s on: ... Check the USB cable or hub of that port.	A serial write is blocked by the driver.	Faulty USB cable or hub; port behind several hubs.	Unplug and replug the USB cable of that port; avoid chained USB hubs. "[recovered] Command loop resumed" follows when it recovers.
[port ERROR] COMxx: the driver did not release the port, unplug and replug the USB cable.	Closing the port hung in the driver.	Same causes as above.	Unplug and replug the USB cable, then detect again.
[PUMP ERROR] <name>: product error detected: ...	Product status reported an error.	Instrument-level fault.	Check the product state and the detailed error text.
Error: Invalid command	Product rejected the command.	Unsupported command or invalid custom command.	Verify the command against the operating manual.
Error: Invalid operand	Product rejected a command value.	Value out of range or malformed (e.g. valve port number above the port count).	Check speed, position, port number, or parameter format.
Error: Missing trailing [R]	Product expected command termination.	Custom command may be incomplete.	Add the required command terminator if applicable.
Error: Device not initialized	Product requires initialization/homing.	Instrument has not been initialized (also after a firmware update or a power cycle).	Click Init and retry.
Error: Command overflow	The product received a command while still executing one.	Device busy; a second move sent to a valve a few hundred milliseconds after the previous one.	Wait for readiness (use Pull and Wait in a sequence) and retry.
Valve error: [144, 'Not homed', ...]	Valve is not homed.	Initialization missing or incorrect configuration.	Click Init and verify valve port configuration.
Move out of range	Pump move is outside allowed limits.	Position or relative move exceeds syringe travel.	Verify current position, syringe size and command value.
Speed out of range	Pump speed is outside allowed limits.	Speed value too high or invalid.	Reduce speed and retry or consult the product documentation for the corresponding speed limit.
Firmware file too small or corrupted (less than 32 bytes).	Selected firmware file is invalid.	Wrong or corrupted .bin file.	Select the correct firmware file.
Firmware file corrupted or incorrect ...	Firmware file type does not match product type.	Wrong firmware for selected instrument.	Select firmware for the correct product family.
Unable to synchronize with the NXP bootloader ... / Bootloader did not acknowledge the synchronization byte ... The product may now be waiting in its bootloader: unplug and replug its power supply, then press DETECT ...	Firmware update could not enter or sync with the bootloader.	Not a direct USB connection, or the product firmware is not compatible with the update procedure.	Unplug and replug the product power supply (a product waiting in its bootloader is not detected until it restarts), press DETECT, connect it directly by USB and retry.
<name> did not come back after the firmware update. Unplug and replug its power supply, then press DETECT.	The product did not answer after flashing.	Product still restarting or blocked in bootloader.	Power-cycle the product and click DETECT.
<name>: <setting> changed by the firmware update (old -> new). Check the product settings before use.	A setting differs after the update.	The firmware reset that setting.	Open Settings and restore the value (syringe size, resolution, force, ports, address).
ERROR: Invalid parameters	A sequence line is incomplete.	Missing value or mode.	Complete the sequence line before running.
ERROR: Invalid port parameter	Valve port selection is invalid.	Value is blank or not numeric.	Enter a valid port number or use the port shortcut menu.

ERROR: Empty execution string	Custom command action has no value.	Empty Custom Command line.	Enter a command string.
ERROR: Non-numeric clock duration	Wait action duration is not numeric.	Invalid Wait value.	Enter a numeric duration.
No products detected.	Sequence was started with no detected products.	No successful detection in Instrument view.	Detect and initialize instruments first.
[BUILD] Line n: ...	The Build found an error or a warning on that line.	See section 4.10.	Fix the line, or read the warning and decide.
Sequence paused at action n: unexpected error (...). Press Play to retry this action.	An internal error occurred while executing a line.	Unexpected product answer or state.	Check the line, the product state and the log file, then press Play: the line is replayed.

3.6 Product Card

Each product card represents one detected or available instrument. Pumps have Speed, Position and syringe controls; valves only the valve wheel. The grid has **six cards** (slots 1 to 6): this is the maximum number of instruments the application drives at a time. Cards 5 and 6 are shown only when a product occupies slot 5 or 6.

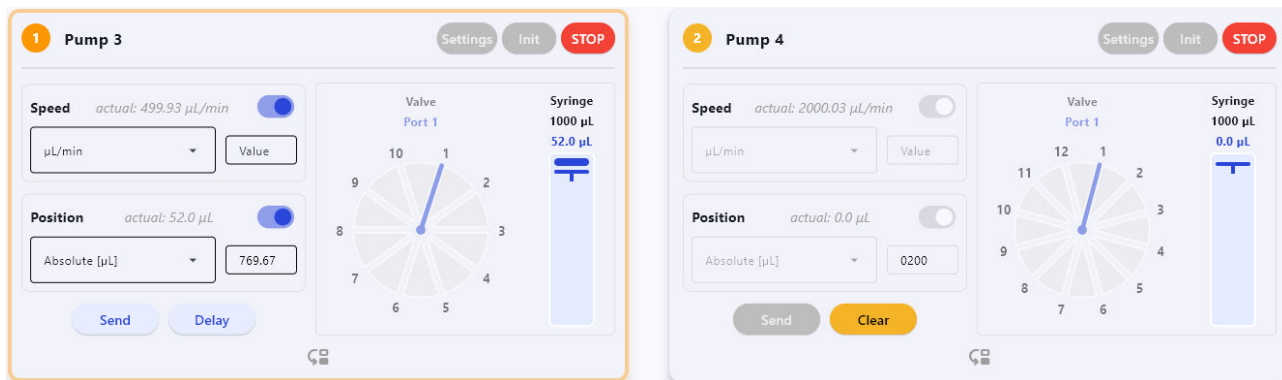


Card elements:

- Slot number and status indicator (colored circle).
- Editable product name (click on it).
- **Settings** button (section 3.7).
- **Init** button: homes the product (ZR).
- **STOP** button: hard stop of this product.
- Valve visual (section 3.9).
- Pump speed controls for pump-family products.
- Pump position controls for pump-family products.
- Syringe visual for pump-family products, with a draggable plunger (section 3.12).
- **Send**, **Delay** and **Clear** buttons for pump-family product cards.
- Drag and Drop handle at the bottom of the card.

Status colors:

Status Color		Tooltip / Meaning
Grey		Disconnected.
Green		Available.
Orange		Busy (moving or executing a command). Settings and Init are disabled meanwhile.
Red		Error.
Yellow		Command pending after Delay: press General GO to execute it, or Clear.
Blue		Firmware updating.

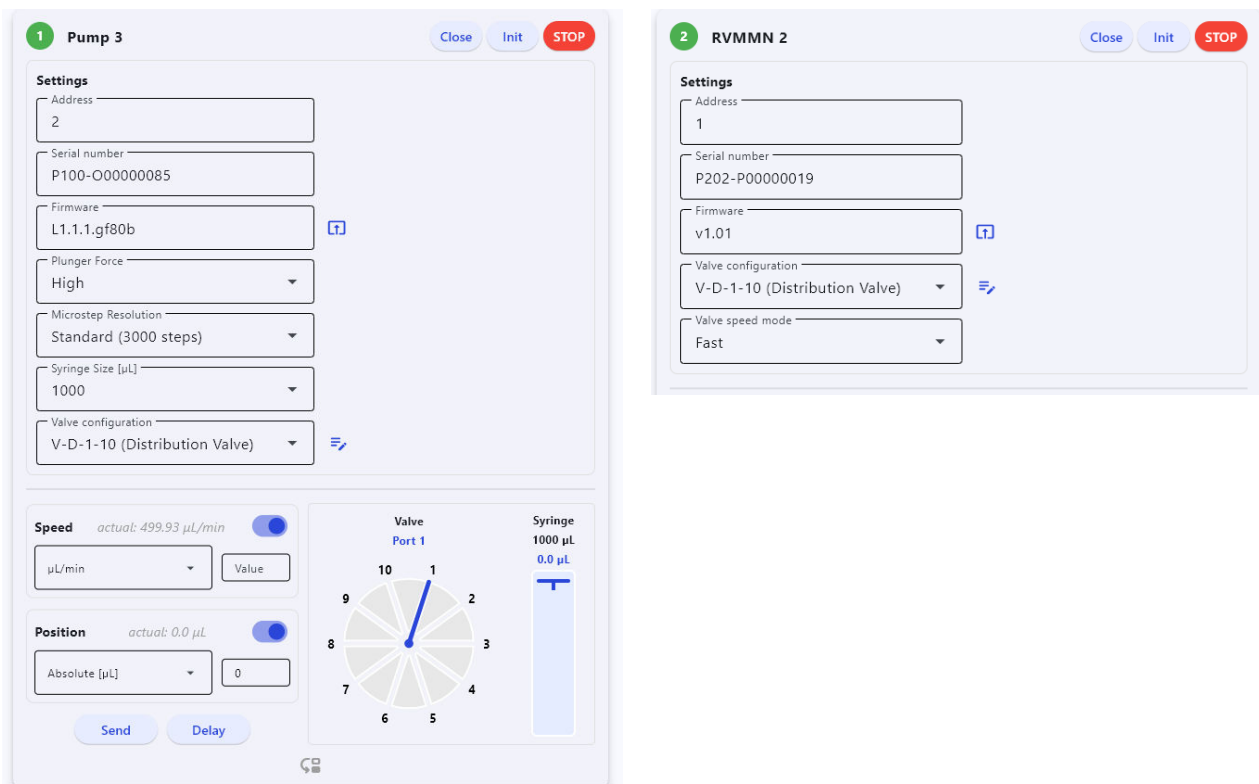


Left: pump busy (orange) while filling. Right: pump with a pending delayed command (yellow) and the Clear button.

Product cards can be dragged using the handle at the bottom of the card to swap visual slots. Slot changes are synchronized with the Sequence view mini product cards and target tooltips, and saved in presets.

3.7 Instrument Settings

Click **Settings** on a product card to open the settings panel (the button becomes **Close**).

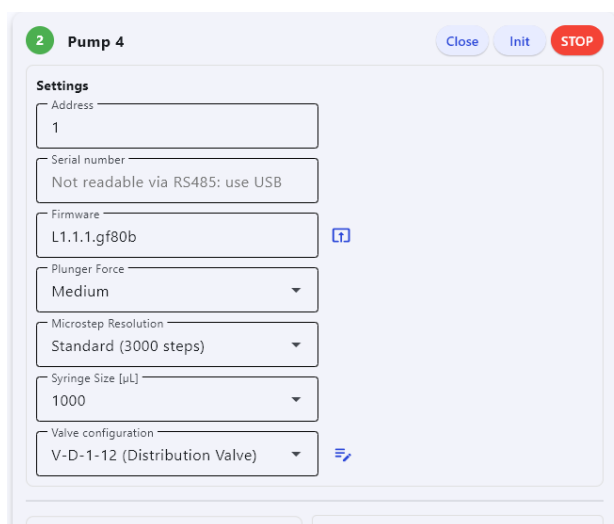


Pump settings

Valve settings (RVMMini, with the valve speed mode)

Setting	Product Type	Behavior
Address	All detected products	Change the product address (1–9, A–E).
Serial number	All detected products	Read-only. Read from the USB chip of the product (pumps, RVM, RVMFS) or from the firmware (RVMmini). Through an RS232 or RS485 adapter, or on a product whose number was never programmed, it reads “Not readable” (the tooltip explains why): connect the product directly by its USB cable to read it. The serial number is also shown on the port pill.
Firmware	All detected products	Shows the firmware version. The firmware update icon is next to it (section 3.8).
Plunger Force	Pump	Sends the selected force immediately. Options: Low, Medium, Normal, High (increasing force). Use a lower force with small syringes.
Microstep Resolution	Pump	Sends the resolution command immediately. Options: Standard (3000 steps), High (24000 steps).
Syringe Size [μL]	Pump	Sends the syringe size immediately and refreshes the syringe visual. Options: 25, 50, 100, 250, 500, 1000, 2500, 5000. Shows “Not defined” when the pump does not store it: choose it before using μL units.
Valve configuration	All product cards	Changes the valve visual and sends the matching port count to the connected instrument. A valve whose port count has no visual (3 or 7 ports, for example) is shown as “Custom (no visual)”: the 12-port wheel is displayed and every port stays clickable; a port the valve does not have answers “Invalid operand”. Custom is set by the detection only, it cannot be chosen by hand.
Valve port names	All product cards	The edit icon next to the valve configuration lets users customize the visible valve port names and tooltips. Saved in presets.
Valve speed mode	RVMFS and RVMmini	Sends the Slow or Fast valve speed command.

Warning: Changing the valve configuration sends a valve port count command and logs a message instructing the user to initialize the product. Initialize the product after changing the valve configuration.



Pump connected through an RS485 adapter: the serial number cannot be read.

Pump firmware too old. When a pump runs a firmware older than **L1.1.1**, its card is locked (greyed) except the Settings panel, and the terminal explains it in red. Update the firmware from the Settings panel: the card unlocks as soon as the new version is read.

3.8 Firmware Update

Firmware update is available from the product Settings panel through the update icon next to the Firmware field.

Confirmed firmware update behavior:

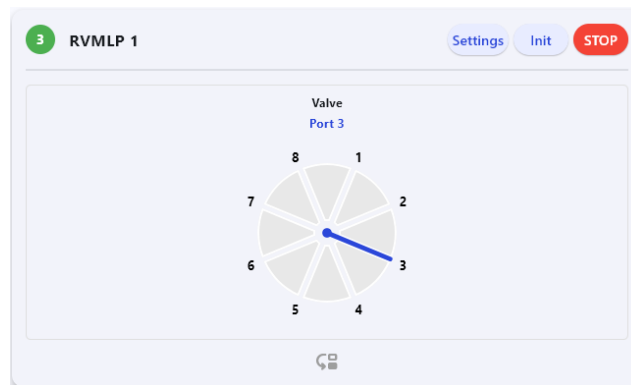
1. The product must be connected **directly by its USB cable**. The application checks the port before anything is sent: a product behind a serial adapter or cable (RS485, RS232, including an RVMmini on its RS232 port), or sharing its port with other products, is refused with the message “Firmware update is only possible through the product's own USB cable. Connect it directly by USB, run DETECT and retry.” and nothing happens.
2. The product must not be busy.
3. The user selects one .bin firmware file for the product family (firmware files are available from ? > Documentations & Downloads).
4. The application retrieves the product serial port, address and type, and takes a snapshot of its settings.
5. The product is disconnected from normal control before flashing.
6. A firmware update overlay is displayed with the instruction: “Do not disconnect the instrument”.
7. The update runs in the background (about 40 s for a valve, 75 s for a pump).
8. The app logs either “[FIRMWARE UPDATE OK]” or “[FIRMWARE UPDATE ERROR]”.
9. The app reconnects the product (“Reconnecting to ...”, up to about 13 s), reads its firmware information again and **compares its settings** with the snapshot: every difference (valve ports, syringe size, resolution, force, valve speed mode, address) is reported in orange. Nothing is restored automatically: check the Settings panel.
10. The product is then **homed automatically** (it restarts non-initialized after a flash) and is available again.

Warning: Firmware update is only possible via USB. Do not disconnect the instrument during the update. If the update fails with “Unable to synchronize with the NXP bootloader”, the product may be waiting in its bootloader: unplug and replug its power supply (a product waiting in its bootloader is not detected until it restarts), then press DETECT and retry.

3.9 Valve Visual

The valve visual shows the currently configured valve type and selected port.

Configuration	Type	Ports
V-O-1-2	ON/OFF Valve	4 internal ports with ON/OFF display behavior.
V-D-1-4	Distribution Valve	4
V-S-1-4	Switch Valve	4
V-D-1-6	Distribution Valve	6
V-S-1-6	Switch Valve	6
V-D-1-8	Distribution Valve	8
V-D-1-10	Distribution Valve	10
V-D-1-12	Distribution Valve	12
Custom (no visual)	Any other port count	Set by the detection when the valve reports a port count that has no visual (3, 7...). The 12-port wheel is shown; the Build and the Python export use the real port count.



Valve moved to port 3 by clicking on the wheel.

User behavior:

- Click a port on the valve wheel to send a valve port command.
- If the product is busy, the app logs that the product is busy instead of sending the valve move.
- The selected port label updates from the configured port name.
- ON/OFF valves display ON and OFF labels.
- Custom port names can be edited in Settings and are reflected in tooltips.
- If the device is not initialized or has lost its initialization, an orange dot appears in the center of the visual: click **Init**.
- A valve records its position a few milliseconds after a rotation and does not answer during that time: leave about one second, or use Pull and Wait in a sequence, before the next command.

3.10 Syringe Control

Pump product cards include Speed and Position controls.

Speed controls:

- Toggle to enable or disable speed command entry.
- Speed modes: V, u, $\mu\text{L}/\text{min}$, mL/min .
- Numeric value field (min/max in the tooltip; red when out of range).
- “actual: ...” display of the speed read from the pump.

Position controls:

- Toggle to enable or disable position command entry.
- Position modes in steps, depending on the microstep resolution (Standard -> 3000 steps, High -> 24000 steps): Pickup [steps], Dispense [steps], Absolute [steps].
- Position modes in μL : Pickup [μL], Dispense [μL], Absolute [μL].
- Numeric value field.
- “actual: ...” display of the position read from the pump.

Send / Delay behavior:

- **Send** sends the enabled command(s) immediately (speed, position, or both).
- **Delay** loads them in the pump; the card turns yellow and the button becomes **Clear**. Execute with **General GO**.
- Speed and volume values are converted to integer pump units and rounded; the field keeps the typed value and the real value is written in the log (section 2.5).
- When the syringe size is “Not defined”, the $\mu\text{L}/\text{min}$, mL/min and μL modes are disabled until a size is chosen in Settings.

Warning: Confirm the syringe size and microstep resolution before sending volume-based commands. Incorrect settings produce incorrect movement volumes.

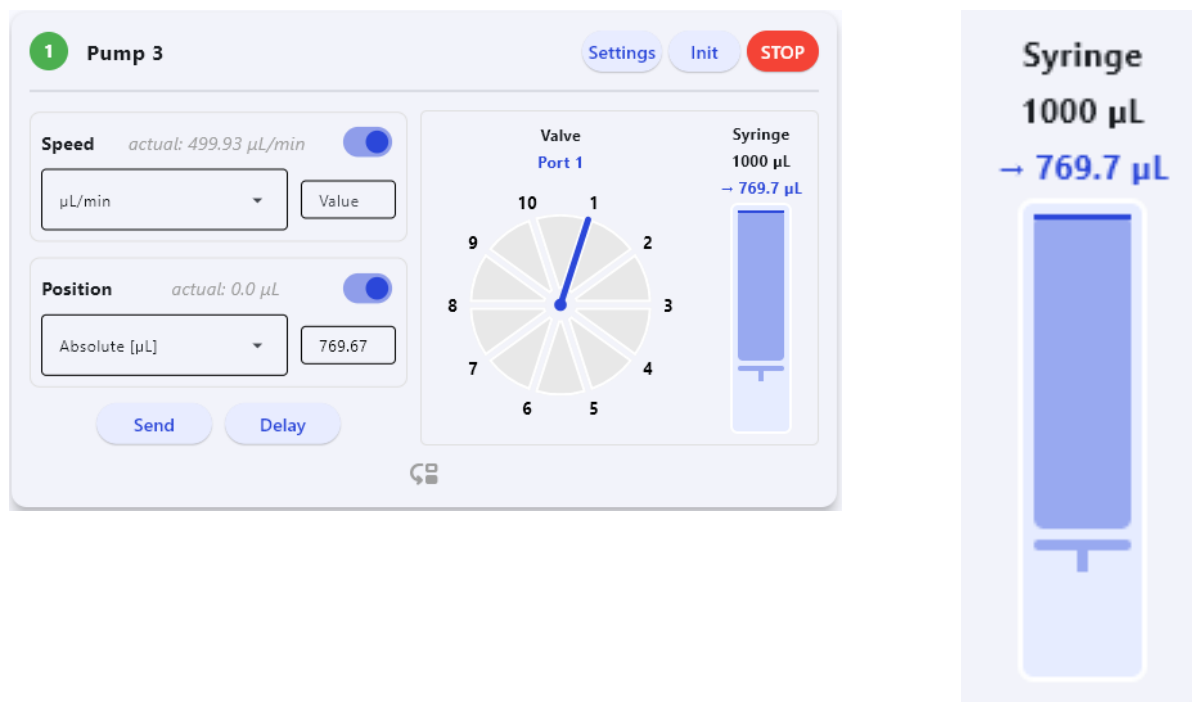
3.11 Syringe Visual

The syringe visual is shown on pump cards. It displays:

- Configured syringe capacity in μL .
- Current volume in μL , calculated from the plunger position.
- A fill level representing the plunger position.
- Initialization status (orange when not initialized).
- “not defined” when the syringe size is unknown.

3.12 Setting a position by dragging the plunger

The T-shaped plunger of the syringe visual can be grabbed with the mouse and moved vertically: down to fill, up to empty. This is a quick way to prime a syringe or to bring it to a given level.



The target volume is written in the Position field (Absolute mode)...

...and previewed on the syringe.

- While dragging, the fill level follows the pointer and the target is shown above the syringe (“→ 769.7 μL ”). Nothing is sent to the pump yet.
- On release, the Position mode is set to **Absolute** (in the current unit, μL by default), the value is written in the field and the Position switch is turned on. The syringe stays on the target in a light color, with a thin line at the real position, until the command is sent.
- Click **Send** to move now, or **Delay** then **General GO**. Typing a value, changing the mode or the syringe size cancels the preview.
- The gesture is refused when the pump is busy (red message), when the syringe size is not defined (orange), or when a delayed command is pending (orange: press Clear or General GO first). A non-initialized pump is accepted.

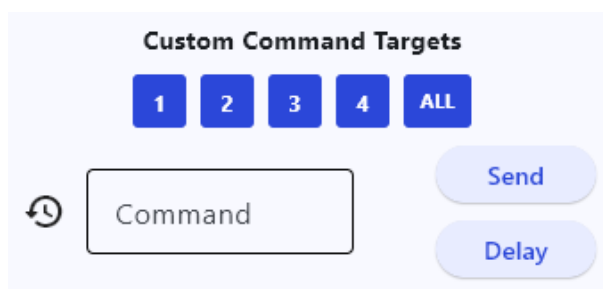
3.13 On-the-Fly Behavior

You can modify pump speed or position parameters while a motion is in progress. Enable the desired speed or position switches, configure the new values, and click **Send** (or **Delay** followed by **General GO**). The updates take effect immediately without interrupting the current operation: a new speed is applied on the fly, a move in the same direction continues from the current position.

Warning: On-the-fly changes are only supported by pump firmware L1.1.1 or newer; older firmware answers with an overflow error. A move in the opposite direction of the current motion also gives an overflow error: stop the pump first.

3.14 Custom Commands

The custom command panel is on the right side of the Instrument view, below the log terminal.



Controls:

- Command targets section with slot boxes for connected instruments and **ALL**.
- Custom command text field.
- Command history menu (clock icon) with the last six commands of the current session.
- **Send** button: sends the command as typed.
- **Delay** button: sends it as a delayed command (h... prefix for pumps, c0H... for valves), executed by General GO. With ALL on an RS485 bus, one broadcast frame is used.

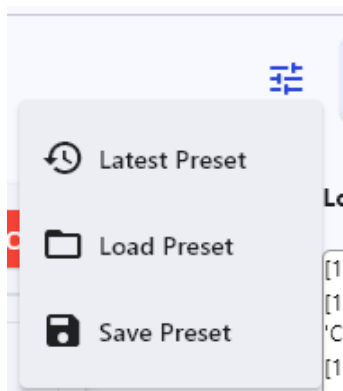
Behavior:

- Commands are sent only to connected selected targets.
- Use a simplified syntax that excludes the product address and the leading slash. For example: /1ZR → ZR, /1A234R → A234R, /1?9200 → ?9200.
- Multi-line answers are read completely and shown in the terminal.
- The app does not validate the custom command syntax beyond empty/non-empty and delay wrapping. Invalid commands produce instrument errors.
- @FWUPDATER is refused as a custom command: use the update icon of the Settings panel.

Warning: Custom commands can move instruments or change their configuration. Use only commands validated for the connected product (see the product operating manual).

3.15 Presets

Preset controls are available from the tuning icon in the Instrument view top bar.



Preset Action	Behavior
Latest Preset	Reloads the last preset loaded or saved in this session. Every load writes “Preset loaded: <file> (n of m products matched by unique ID)” in the log.
Load Preset	Opens a file picker for a .json preset file.
Save Preset	Saves a .json preset file.

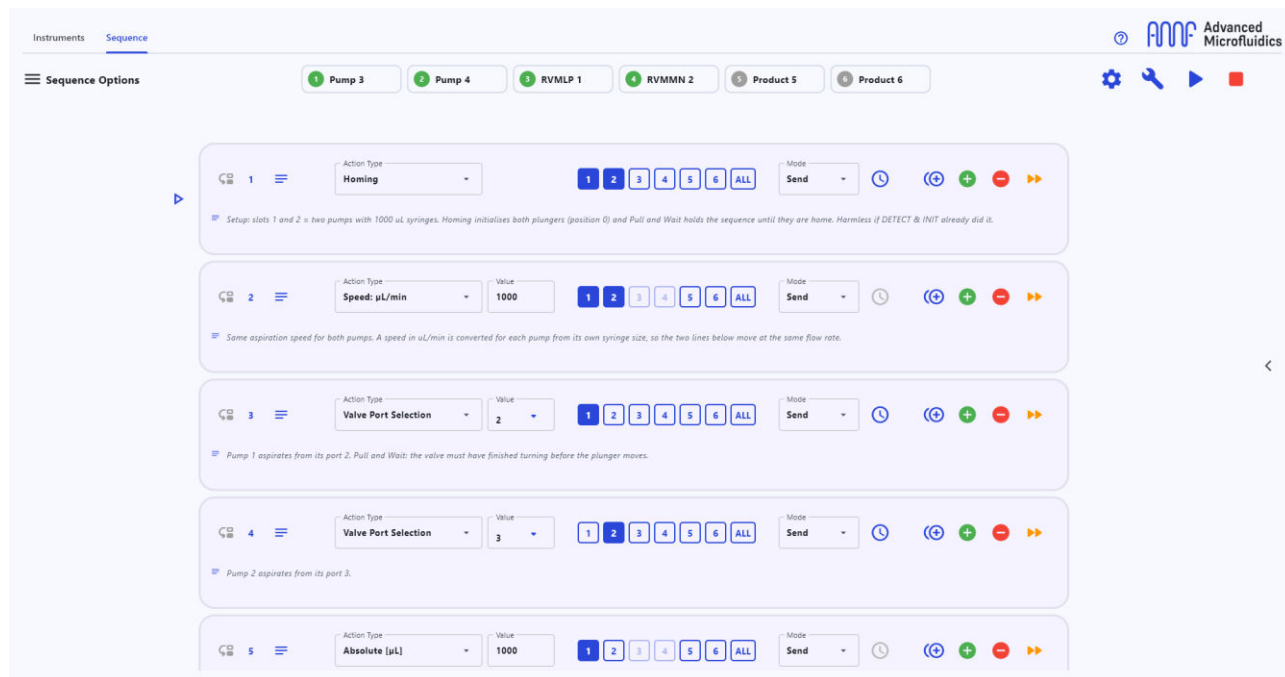
Preset contents:

- Connected product unique ID.
- Product name.
- Slot number.
- Valve configuration.
- Valve port names.

Presets are stored by default in %APPDATA%\AMF\AMF Control Center\Preset. No preset is delivered with the application: the folder stays empty until the first Save Preset (a preset.json file left there by a version older than V1.0.10 was an R&D test preset and can be deleted).

Shortcut	Action
Ctrl + S	Save Preset
Ctrl + O	Load Preset
Ctrl + Shift + O	Latest Preset
Ctrl + Tab	Retract or expand the log and commands panel

4. Sequence View



The Sequence view with the example sequence “1_synchronized_start” loaded.

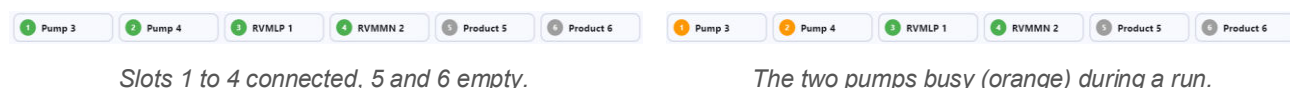
4.1 Purpose of the Sequence View

Use the Sequence view to create, check and run ordered instrument workflows. Capabilities:

- Visualize up to six instrument slots in the top bar.
- Swap slots by dragging mini product cards.
- Create new sequences, load and save sequence workflows.
- Select the default speed and position units.
- Add, duplicate, delete, skip, drag, and annotate sequence lines.
- Select target slots or ALL per sequence line.
- Select an action type, value, and mode per line.
- **Build**: simulate the sequence and report errors and warnings before running it.
- Run, pause, and stop a sequence; the list follows the running line.
- Use Wait, Pull and Wait, Loop, Homing and Pause actions.
- Show current and next-step indicators, warnings and sequence status messages.
- Read the log in a side panel that opens automatically on errors.

4.2 Mini Product Card Visualization

The top bar contains six mini product cards, one for each slot.



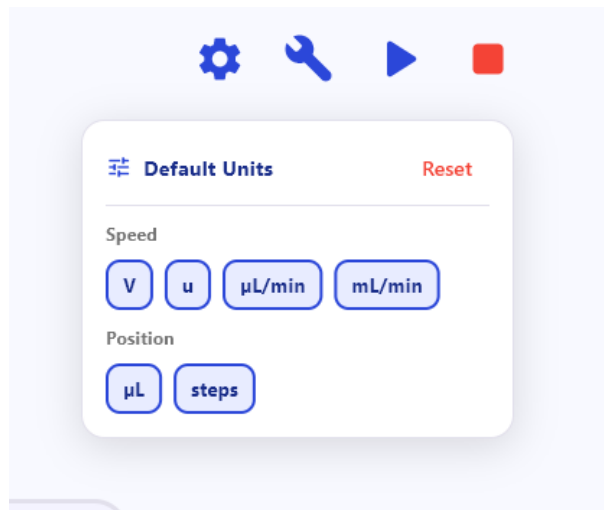
Slots 1 to 4 connected, 5 and 6 empty.

The two pumps busy (orange) during a run.

Each mini card shows the slot number, the product name and the status color matching the Instrument view card. Mini cards can be dragged to swap slots; slot swaps are synchronized with the Instrument view, and product names update when changed in the Instrument view.

4.3 Default Unit selection

Use the Default Units panel to filter the options available in the Action Type dropdown:



- Click the **Settings** (gear) icon in the Sequence view top bar to open the panel.
- Toggle specific Speed units (V, u, µL/min, mL/min) to show or hide them.
- Toggle specific Position units (µL, steps) to show or hide them.
- Disabled units are instantly removed from the action menus across all sequence lines.
- Click **Reset** to restore all default unit options.

4.4 Sequence Line Presentation



A new sequence line.

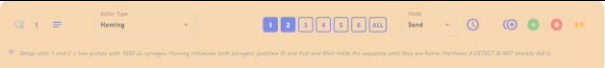


A line with its note open.

Each sequence line contains:

- Drag and Drop handle.
- Line number.
- Note button and optional note field.
- Action Type dropdown.
- Value field (or Iterations, Duration, depending on the action).
- Target boxes for slots 1 to 6 and ALL.
- Mode dropdown (Send / Delay; From line for a Loop; Unit for a Wait).
- Pull and Wait clock button.
- Duplicate, Add, Delete and Skip buttons.

Visual states:

State	Visual Behavior	
Next	Blue outlined play indicator on the left of the line.	
Current	Orange play indicator and orange line.	
Sent	Green line background.	
Skipped	Grey line background.	
Error	Red line background; the sequence pauses on it.	
Build error / warning	Red or orange border with an icon whose tooltip gives the message (section 4.10).	
Loop	A colored rail on the left shows the loop span.	

Target selection behavior:

- Users can select one slot, multiple slots, or ALL.
- Some actions force target selection to ALL (General GO, Wait).
- Pump-specific actions visually dim non-pump products.
- New lines start with no target selected; select a target before running.

4.5 Action Types

Action Type	Purpose	Required Inputs	Behavior	Notes
Speed: V	Set pump speed using V command units.	Target pump, numeric value, mode Send or Delay.	Sends a speed command.	Value is rounded to an integer.
Speed: u	Set pump speed using u command units.	Target pump, numeric value, mode Send or Delay.	Sends a speed command.	Value is rounded to an integer.
Speed: µL/min	Set pump flow rate in µL/min.	Target pump, numeric value, mode Send or Delay.	Converts the flow rate using syringe size and reducer, then sends a u speed command.	With ALL, the app sends the command individually to each connected pump.
Speed: mL/min	Set pump flow rate in mL/min.	Target pump, numeric value, mode Send or Delay.	Converts the flow rate using syringe size and reducer, then sends a u speed command.	With ALL, the app sends the command individually to each connected pump.
Absolute [µL]	Move the pump to an absolute volume position.	Target pump, numeric value, mode Send or Delay.	Converts µL to steps and sends an absolute position command.	Uses syringe size and microstep resolution.
Absolute [steps]	Move the pump to an absolute step position.	Target pump, numeric step value, mode Send or Delay.	Sends an absolute position command.	Value is rounded to an integer.
Pickup [µL]	Aspirate a relative volume.	Target pump, numeric value, mode Send or Delay.	Converts µL to steps and sends a pickup command.	Uses syringe size and microstep resolution.
Dispense [µL]	Dispense a relative volume.	Target pump, numeric value, mode Send or Delay.	Converts µL to steps and sends a dispense command.	Uses syringe size and microstep resolution.
Pickup [steps]	Aspirate a relative number of steps.	Target pump, numeric step value, mode Send or Delay.	Sends a pickup command.	Value is rounded to an integer.
Dispense [steps]	Dispense a relative number of steps.	Target pump, numeric step value, mode Send or Delay.	Sends a dispense command.	Value is rounded to an integer.
Valve Port Selection	Move the valve to a port.	Target instrument, numeric port value.	Sends a valve port move command.	The value field offers a shortcut menu based on the configured port names of the selected slot.
Homing	Initialize the product (valve and plunger).	Target instrument(s) or ALL, mode Send or Delay.	Sends ZR (c0HZR in Delay).	Use it with Pull and Wait at the start of a sequence; harmless if DETECT & INIT already did it.
Custom Command	Send a raw command string.	Target instrument, command text.	Sends the command as entered.	Invalid commands trigger instrument errors. The Build only simulates the basic commands.
General GO	Execute delayed commands.	No value required.	Sends R (one broadcast frame on an RS485 bus).	Target selection is forced to ALL. Optional Pull and Wait waits for the moves to finish.
Wait	Wait for a time duration.	Numeric duration, unit ms, s, or min.	Pauses sequence execution for the specified duration; nothing is sent.	Target selection is forced to ALL.
Pull and Wait	Synchronize sequence execution with instrument completion.	Target selection.	Waits until the selected targets are no longer busy, then continues.	See section 4.6.
Loop	Repeat a block of lines.	Iterations (total number of passes, 1 = once) and From line (first line of the block).	Jumps back to "From line" until the block has been executed Iterations times. During the run the label shows the pass in progress: "Iterations (1/3)" from the first line of the block, up to "(3/3)" during the last pass, then "Iterations" again.	"From line" must be before the Loop line. 0 or empty is an error.

Pause	Pause sequence execution.	No value required.	Stops sequence progression and returns the play button to Play.	Users must resume manually with Play.
-------	---------------------------	--------------------	---	---------------------------------------



Loop line: Iterations and From line.



Pause line.



Wait line, with the loop rail on the left.



General GO line, target forced to ALL.

4.6 Pull and Wait

Pull and Wait is implemented in two forms:

- A dedicated **Pull and Wait** action type.
- A clock icon toggle available for most action lines.

Behavior:

- When Pull and Wait is active, the sequence waits for the selected products to finish before continuing.
- For multi-select targets, the sequence waits until all selected products are no longer busy.
- For ALL, the sequence waits until all connected products are no longer busy.
- If a product error occurs during Pull and Wait, the line is marked as error and the sequence pauses.
- If Mode is Delay, the Pull and Wait icon is disabled for that line.
- A Pull and Wait on a pump that holds a delayed command never finishes (the pump stays busy until General GO): the Build reports it.

Use Pull and Wait when the next line must not start until the instrument motion or operation has completed: after a valve move before the plunger moves, after a Pickup or Dispense before the next command to the same pump. Without it, a second command sent to a moving product gives a “Command overflow” error or a move restarted from the current position.

4.7 Adding, Skipping, Duplicating, and Deleting Sequence Lines

Action	Icon	Result
Add		Inserts a new line after the current line.
Duplicate		Duplicates the line after the current line.
Delete		Deletes the line unless it is the only remaining line.
Skip		Marks the line as skipped. Skipped lines are kept in the file but not executed.
Drag and Drop		Reorders sequence lines and updates loop targets.
Note		Expands or collapses a one-line note field.
Pull and Wait		Waits for the targets of the line to finish before the next line.

Additional behavior:

- If a skipped line is the current running line, the sequence advances to the next line.
- If a Wait line is skipped while waiting, the wait is cancelled and the sequence advances.
- On the current running line the delete icon becomes a stop button: it sends a hard stop to the affected target and pauses the sequence.

4.8 Starting and Stopping a Sequence



Settings, Build, Play, Stop.



While running: the Play icon becomes Pause.

Use the play/pause icon to start and pause a sequence, the red square to stop it.

Start behavior:

- If no products are detected, the app shows “No products detected.”
- Starting a sequence disables some Instrument view controls.
- The app logs “Sequence execution started.”
- The sequence begins from the next available non-skipped line.
- The list scrolls so that the running line stays visible (section 4.11).

Pause behavior:

- Clicking the pause icon changes it back to play.
- Any active Wait timer is cancelled.
- The app logs “Sequence execution paused.”
- Pause does not hard stop a product command already sent.

Stop behavior:

- The stop icon stops sequence execution and resets the current step.
- Pull and Wait and active Wait states are cancelled.
- Lines return to normal or skipped styling; the Loop progress labels are reset.
- The app logs and displays “Sequence execution stopped.”
- The app sends a hard stop to all products (one broadcast frame on an RS485 bus).

Error behavior:

- Command errors and product errors mark the current line in red, the sequence pauses (Play icon) and the log panel of the Sequence view opens.
- Instrument controls are re-enabled.
- Press **Play** to retry the failed line, or fix it first. Pressing **STOP** on a product card or **GENERAL STOP** in the Instrument view also pauses the sequence (“... stopped — sequence paused, press Play to resume.”).

The screenshot displays the 'Sequence Options' interface. It features a sequence of six steps, each with a specific action type and value. Step 2 is 'Speed: $\mu\text{L}/\text{min}$ ' with a value of 1000. Step 3 is 'Valve Port Selection' with a value of 220. Step 4 is 'Valve Port Selection' with a value of 3. Step 5 is 'Absolute [μL]' with a value of 1000. Step 6 is 'Absolute [μL]' with a value of 1000. The Log Terminal on the right shows a list of commands and their responses, with a red error message at the bottom: '[18:36:57] [ERROR] Pump 3: cmd=8220R -> Error: Invalid parameter (error code: 0)'.

A line in error (red) after an invalid valve port: the previous line is green, the log panel opened by itself with the error message.

4.9 Sequence Options

The 'Sequence Options' menu is shown, with the following options: 'New Sequence', 'Load Sequence', 'Save Sequence', and 'Export to Python'. The 'Pump' button is visible in the top right corner.

Option	Behavior
New Sequence	Clears the current sequence and creates three new blank lines.
Load Sequence	Opens a .json file picker and loads a sequence workflow. Loading is ignored while a sequence is running.
Save Sequence	Opens a .json save dialog and saves the current sequence workflow.
Export to Python	Writes a standalone Python script that replays the sequence with the AMF Python SDK (section 4.14). Not available while a sequence is running.

Sequence files are stored by default in %APPDATA%\AMF\AMF Control Center\Sequences. Three commented **example sequences** are copied there at the first start (see section 4.13 and Appendix A). A sequence file is a plain text file that can be edited, e-mailed or given to an AI assistant.

4.10 Build: checking a sequence before running it

The **Build** button (wrench icon) replays the whole sequence in a simulator, without sending anything to the instruments, and reports what would go wrong. It reads the real state of the connected products first: position, initialization, syringe size, resolution, valve port count, firmware version.



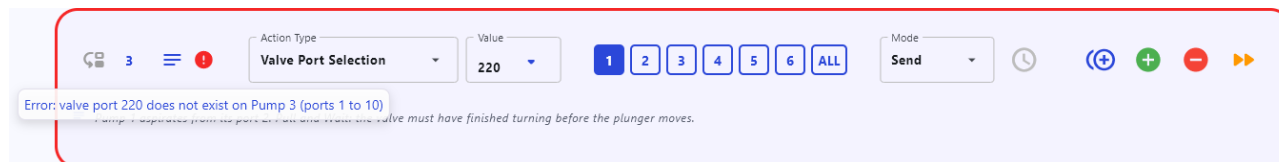
Build OK



One error

Result:

- A badge on the wrench icon: **OK** (green), the number of warnings (orange) or the number of errors (red).
- A banner under the top bar and “[BUILD] ...” lines in the log terminal, one per issue, with the line number (and the loop pass when relevant).
- On each faulty line, a red (error) or orange (warning) border and an icon: hover it to read the message.



Hover the icon to read the Build message.

```
[18:33:43] [BUILD] Checking 14 lines (14 steps simulated)...
```

```
[18:33:43] [BUILD] Line 3: valve port 220 does not exist on
```

```
Pump 3 (ports 1 to 10)
```

```
[18:33:43] [BUILD] Build failed: 1 error, 0 warnings - see
```

```
the log terminal.
```

The same messages in the log terminal.

What the Build checks:

- Every line is complete: action, value, mode, targets; loop targets and iterations; wait duration and unit.
- Targets are connected and initialized; pump actions are not sent to valves; ALL in steps does not mix resolutions (warning) and does not reach a valve (error).
- Values are in range: speeds and flow rates for the syringe size, valve port numbers for the valve.
- **Positions:** each pickup, dispense and absolute move is simulated from the real plunger position. A move that would exceed the stroke or go below zero is an error; a move that may exceed it depending on where the plunger really is (after a move without Pull and Wait) is a warning. A second pass simulates the sequence from an empty syringe and reports as warnings what would only break from position 0.
- **Delayed commands:** one pending command per pump; a Delay overwritten by a second Delay or deleted by a Send is an error (that line will never run); a Delay never released by a General GO is a warning; a Pull and Wait on a pump that holds a delayed command is an error (it would never finish).
- **Movement in progress:** a command sent to a product that may still be moving (no Pull and Wait since its last move) is a warning: firmware Overflow error possible.
- Loops are replayed entirely (up to 200 000 simulated steps). Custom commands other than the basic move, speed and valve commands are not simulated (warning).
- Firmware too old, syringe or reducer not read correctly: warning.

Build never blocks Play: the sequence runs as written even when errors are reported. Any edit of a line or any change on the products (command sent, disconnection, detection) clears the marks: build again.

Build OK: 9 steps simulated, no issue found.

Log Terminal:

```
[16:07:14] AMF Control Center V1.1.0 - Full version
[16:08:03] Multiple COM ports detected. Please manually
[16:08:03] select targets from the port selector panel.
[16:08:03] COM ports list refreshed.
[16:08:24] 2 COM ports selected. Skipped (no USB
[16:08:24] identification, not an AMF product): COM1, COM3.
[16:09:28] Starting detection on port ["COM12", "COM22"].
[16:09:28] (a few seconds per port, up to about 3 minute)
[16:09:29] Detected instruments: [1, 2]
[16:09:31] valve with 3 ports has no visual available. A 12-
[16:09:31] port wheel is shown: ports the valve does not have will
[16:09:31] answer "invalid opened".
[16:09:32] [send] Pump 1: cmd=ZR answer=@
[16:09:33] [send] RVMF5 2: cmd=ZR answer=@
[16:09:37] [done] RVMF5 2 is available.
[16:09:47] [done] Pump 1 is available.
[16:13:59] [BUILD] Checking 9 lines (9 steps simulated)...
[16:13:59] [BUILD] Build OK: 9 steps simulated, no issue
[16:13:59] found.
```

Build OK: badge on the wrench and “[BUILD] Build OK” in the log panel.

4.11 Following the running line

While a sequence runs, the list scrolls automatically so that the running line sits in the upper third of the view. The following is suspended as soon as you scroll with the mouse wheel, drag the list or edit a line, so that you can read or fix the rest of the sequence. When the running line is out of view, a button appears at the bottom of the list:



Click it to jump to the running line and resume following. Play (from a stop) and Stop re-arm the following.

Dispense speeds differ: pump 1 (continuous phase, e.g. oil) at 500 µL/min. Speed lines stay in Send mode: a pump only keeps ONE delayed command, so the speed is set now and the move is delayed below.

Pump 2 (dispersed phase, e.g. water) twice as fast: 1000 µL/min.

Delay = the command is loaded into the pump but NOT executed yet. The line turns yellow (command pending). Pull and Wait is not available on a delayed line.

Second pending command, in pump 2. Each pump now holds exactly one delayed move.

General GO releases every pending command at once: both plungers start together (a single broadcast frame on an RS485 bus). Pull and Wait: the sequence ends when both syringes are empty, pump 2 after 1 min, pump 1 after 2 min.

Line 7 running

The list scrolled down by hand while line 7 is running: the button brings you back to it.

4.12 Log panel of the Sequence view

The Sequence view has the same log terminal as the Instrument view, in a side panel closed by default. Open it with the arrow at the right edge of the view or with **Ctrl + Tab**. It shows the same lines (the terminal is shared) with the same header buttons: export a support report, open the log folder, copy the session log, clear the display of this panel.

The panel **opens automatically** when a red error is logged while the Sequence view is on screen (command error, product error, disconnection). An error that happens while the Instrument view is displayed does not change the panel state.

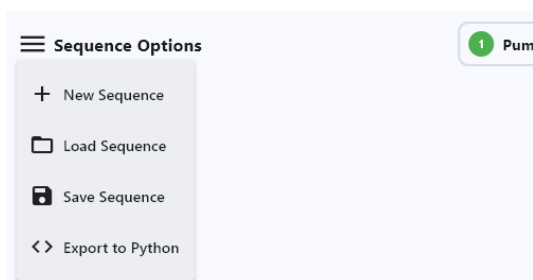
4.13 Example sequences

File	What it shows
1_synchronized_start.json	Two pumps load from different ports, then start dispensing at the same instant with Delay + General GO , at two different flow rates.
2_continuous_perfusion.json	Two pumps in relay: one delivers while the other refills and gets ready, in a Loop ; each delivery starts the instant the previous one ends.
3_reference_all_actions.json	Every action type used once, one rule per line: the reference to read and to give to an AI assistant.

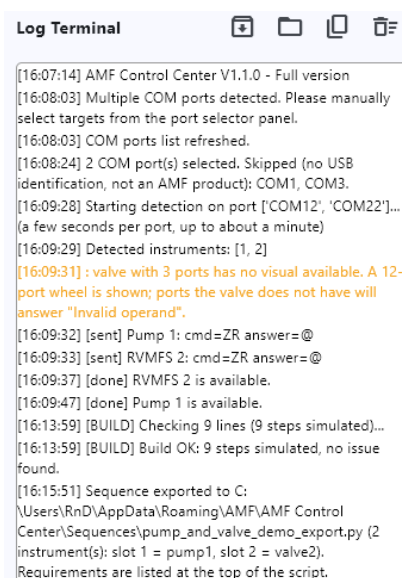
They are written for two pumps with 1000 µL syringes in slots 1 and 2, every line carries a note, and they build without any remark. Load them from Sequence Options > Load Sequence. Appendix A describes them line by line and gives a prompt to write your own sequence with an AI assistant.

4.14 Export to Python

Sequence Options > Export to Python writes the current sequence as a standalone Python script that replays it with the AMF Python SDK (amftools, the package AMF Control Center itself is built on). The script is meant for integration: each sequence line is one commented block, so you can insert your own code between two lines, for example to trigger a camera, read a sensor or log data, and run the whole experiment from your own program without AMF Control Center.



The fourth entry of Sequence Options



The log after an export: file written, instruments of the script

The proposed file name is the name of the sequence followed by `_export.py` (sequence_export.py for a sequence that was never saved), in `%APPDATA%\AMF\AMF Control Center\Sequences` by default. The log of the Sequence view confirms the export and lists the instruments found in the script. Any problem that prevents the export (a Loop with 0 iterations, two loops that overlap without being nested, a line without target, a non-numeric value) is written in red and nothing is written: fix the line and export again.

Requirements to run the script

Requirement	Details
Python	Python 3.10 or newer, from python.org. Any operating system supported by the SDK.
AMF Python SDK	<code>pip install amftools==0.1.11</code> (the version is written at the top of the script; a different installed version only gives a warning at start). The package installs <code>pyserial</code> and <code>ftd2xx</code> ; the USB driver is the one already used by AMF Control Center.
COM ports	Close AMF Control Center, or close its COM ports (section 3.2), before running the script: a COM port can be opened by one program at a time.
Instruments	Powered and connected as they were when the sequence was exported. The header of the script repeats these requirements and the safety notes.

How the instruments are identified

The script declares one instrument per slot used by the sequence, in a `connect_instruments()` function, exactly as AMF Control Center saw it at export time:

- an instrument alone on its USB cable is identified by the **serial number of its USB chip** (e.g. P100-00000085): the SDK finds the COM port from that number, so the script also works on another PC without any change;
- an instrument on an **RS485 bus** behind a USB adapter is identified by **COM port and address** (e.g. COM15, address 1), because the serial number visible on the USB side is the adapter's. The alternative that finds the port from the adapter serial number is given as a comment under each instrument.

A slot targeted by the sequence but empty at export time is written with the identifier `REPLACE_ME` and an orange message in the log: fill it in before running the script, which otherwise stops before sending anything. Pumps are declared with their syringe size: the SDK writes it to the pump at connection, exactly as the Syringe menu of the product card does.

What each line becomes

Sequence line	Python code (AMF SDK)
Speed $\mu\text{L}/\text{min}$, mL/min	<code>setFlowRate($\mu\text{L}/\text{min}$):</code> the SDK converts to the <code>u</code> unit with the syringe size and reduction ratio of the pump, like AMF Control Center.
Speed <code>V</code> , <code>u</code>	<code>setSpeed(V)</code> , <code>setSpeedUltraLowFlow(u)</code>
Absolute / Pickup / Dispense [μL]	<code>pumpVolume</code> , <code>pumpPickupVolume</code> , <code>pumpDispenseVolume</code> (μL , converted by the SDK with the resolution read on the pump)
Absolute / Pickup / Dispense [steps]	<code>pumpAbsolutePosition</code> , <code>pumpPickup</code> , <code>pumpDispense</code> (motor steps)
Valve Port Selection	<code>valveShortestPath(port, enforced=True):</code> the <code>B</code> command, a full turn if the valve is already on that port
Homing	<code>home()</code>
Custom Command	<code>sendBrute("..."):</code> the raw command string, unchanged
Delay mode	The frame AMF Control Center sends (<code>c0H...</code> for a move or a valve, <code>h...</code> for a speed) by <code>sendBrute</code> , with a comment; the SDK has no higher-level function for pending commands.
General GO	<code>resume()</code> on each targeted pump, one after the other (no broadcast frame in the script: the pumps start a few milliseconds apart).
Pull and Wait (icon or action)	<code>block=True</code> on the call for a single target; several targets are started together then waited for with <code>pullAndWait()</code> on each of them.
Wait	<code>time.sleep(seconds)</code>
Pause	A native dialog "Program paused, press OK to continue" (Tkinter; falls back to a prompt in the terminal), the script waits for the click.
Loop	A Python <code>for</code> loop around the block, with the total number of passes (section 4.5). Nested loops are nested <code>for</code> loops; two loops that overlap without being nested cannot be exported.
ALL target	<code>ALL</code> , the list of every instrument declared in the script; <code>PUMPS</code> for pump-only actions.
Skipped line	Kept as commented code.

Running the script

Open a terminal in the folder of the script and run `python 2_continuous_perfusion_export.py`. The script connects to the instruments, prints each sequence line as it starts (with the time of day), then Sequence completed. **Ctrl+C** stops it: every instrument receives a hard stop (`T`), like the STOP button. Any error reported by an instrument stops the script the same way, with the SDK message.

Warning: The script moves the instruments exactly as the sequence does, without the checks of the Build. Build the sequence in AMF Control Center before exporting it, and check tubing, reservoirs and syringe sizes before running the script. The values (volumes, flow rates, syringe sizes) are those of the sequence at export time: export the sequence again after changing it.

An exported example, `2_continuous_perfusion_export.py`, is copied next to the example sequences at the first start (section 4.13): it shows the structure of a script, the two ways of identifying an instrument and a loop. The functions used by the scripts are documented in the AMF Python SDK (`amftools`) documentation.

4.15 Keyboard Shortcuts

Shortcut	Action
Ctrl + S	Save Sequence
Ctrl + O	Load Sequence
Ctrl + Tab	Open or close the log panel

5. Troubleshooting

5.1 COM Port Not Detected

Symptoms:

- COM selector is empty.
- Log shows “No COM ports detected. Please connect an instrument and refresh.”

Actions:

1. Confirm the instrument is powered.
2. Check USB, RS232, or RS485 adapter cabling.
3. Confirm the FTDI driver is installed: the setup installs it with an administrator confirmation, and Windows Update installs it as well when an instrument is plugged in. If the port still does not appear, install the driver from <https://ftdichip.com/drivers/> with administrator rights.
4. Close other applications using the COM port.
5. Disconnect and reconnect the cable.
6. Click the COM selector refresh icon.
7. Restart the application if the port remains unavailable.

5.2 Instrument Does Not Respond

Symptoms:

- Log shows “No answer from the product...”, “Write timeout...”, or “[product DISCONNECTED]”.
- Product card turns red or grey.

Actions:

1. Check power and cabling.
2. Confirm the correct COM port is selected.
3. For RS485, confirm the product address is valid and unique.
4. Verify the device is not held by another serial application.
5. Re-run DETECT & INIT or DETECT.

5.3 Broadcast Does Not Behave as Expected

Possible causes:

- Instruments are on multiple COM ports, so the app sends commands individually.
- μ L-based ALL commands are intentionally sent individually to use each pump's syringe size.

Actions:

1. Confirm whether instruments are on one RS485 COM port or multiple COM ports.
2. Use specific slot targets for troubleshooting.
3. For a simultaneous start on several ports, use Delay on each product then General GO.
4. Check logs for “[sent] Broadcast” versus per-product command entries.

5.4 Firmware Update Fails or Is Refused

Symptoms:

- Log shows “[FIRMWARE UPDATE REFUSED]”: the product is not connected directly by its USB cable (serial adapter or cable, or shared port). Nothing was sent to the product.
- Log shows “[FIRMWARE UPDATE ERROR]”: corrupted firmware, bootloader synchronization, reconnect failure.
- After the update, orange lines report settings changed by the firmware.

Actions:

1. Connect the product directly with its own USB cable, detect it, and retry.
2. Confirm the selected file is a .bin file for the correct product family.
3. Confirm the product is connected and not busy before the update.
4. If the log says “Unable to synchronize with the NXP bootloader”, “Bootloader did not acknowledge the synchronization byte” or “did not come back”: unplug and replug the product power supply, press DETECT, then retry.
5. After a successful update, check the Settings panel (syringe size, resolution, force, valve ports) and restore any changed value; the product is homed automatically.
6. Export a support report and contact support if the synchronization keeps failing.

Warning: Firmware cannot be updated through RS485 or RS232: only through the USB cable of the product.

5.5 Commands Are Not Sent or not Delayed

Possible causes:

- No target selected.
- Product is busy.
- Sequence is waiting for Pull and Wait.
- Invalid custom command.
- Missing mode or value in a sequence line.
- Speed switch on with an empty value (“Speed ignored ... because speed value is empty”).

Actions:

1. Confirm selected target slots.
2. Check the product card status color.
3. If the card shows Clear, either click Clear or use General GO if the delayed command should proceed.
4. Use STOP only when immediate interruption is required.
5. Review log messages for invalid parameters or product errors.

5.6 Sequence Does Not Start

Symptoms:

- Warning shows “No products detected.”
- Sequence returns to play state.
- First line is marked incorrect or error.

Actions:

1. Detect and initialize instruments in the Instrument view.
2. Click **Build** and fix the reported errors.
3. Select target slots for each actionable line.
4. Confirm action type, value, and mode are complete.
5. Avoid selecting non-pump targets for pump actions.
6. Correct any red or invalid lines.

5.7 Sequence Stops Unexpectedly

Possible causes:

- Product error or command error (invalid port, overflow, move out of range).
- Wait action cancelled.

- Pull and Wait received a product error.
- User clicked Stop, General STOP, or a product STOP.
- Current line was deleted while running.
- A Loop line with Iterations 0 or empty.

Actions:

1. Check the line highlighted in red.
2. Read the log panel, which opens by itself on the first error message.
3. Check instrument status in the Instrument view.
4. Re-initialize instruments if errors mention not homed or device not initialized.
5. Press Play to retry the line, after resolving the instrument state. Build the sequence before the next run.

5.8 Pump card greyed out

- If only **Settings** stays active and the terminal shows “firmware ... is older than Firmware-UFNG-L1.1.1”: update the pump firmware from the Settings panel (section 3.8). The card unlocks once the new version is read.
- If the whole card is grey with “NOT CONNECTED”: the product was disconnected or its port was closed; detect again.

5.9 Application freezes or a port cannot be closed

Symptoms: the interface no longer responds, the log shows “[ERROR] Command loop stalled ...” or “[port ERROR] ...: the driver did not release the port”.

Cause: the USB driver is blocked on a write to the port, typically with a faulty cable or a product behind several USB hubs.

1. Unplug and replug the USB cable of the port named in the message: the loop resumes (“[recovered] ...”).
2. Connect the products directly to the computer, or through a single powered hub.
3. If the application does not recover, close it, unplug the cable, and start it again.
4. Export a support report (log header button) for AMF support.

5.10 Logs and Support Diagnosis

Before contacting support, collect:

- Application version shown in the ? menu.
- Exact instrument setup: USB, RS232, RS485, single port, or multiple ports.
- COM port names, product names, slots, addresses and serial numbers (port panel or Settings).
- Firmware versions shown in Settings.
- Sequence or manual action being performed (attach the sequence .json file).
- The **support report** (.zip) exported with the first button of the log terminal header, or the log files from %APPDATA%\AMF\AMF Control Center\logs.
- Firmware file name and product type if a firmware update failed.

6. FAQ

Which communication mode should I use?

Use the mode that matches your hardware. The app auto-detects USB/RS232 first, then RS485. For multiple instruments on one bus, RS485 is the multi-drop mode.

Why is my COM port not visible?

The instrument may be disconnected, unpowered, missing a driver, or already open in another program. Refresh the COM selector after checking the connection.

Which instrument is “Pump 1”?

Open the port panel: each pill shows the product name, its serial number and its COM port. The serial number is also in the Settings panel of the card. Through an RS485 adapter it cannot be read: connect the product by USB, or rename the product (click its name) and save a preset.

What is the difference between DETECT and DETECT & INIT?

Both detect the products. DETECT & INIT also homes them (valve and plunger to their reference position), which is required before any move. Use DETECT alone when the products must not move, then Init on each card.

Can I update firmware over RS485 or RS232?

No, only through the USB cable of the product. The application refuses the update when the product is behind an adapter, before sending anything.

What is the difference between Send, Delay, General GO and General STOP?

Send executes the configured command immediately. Delay loads it in the product without executing it and switches the card to Clear (yellow status). General GO executes all pending commands at once. General STOP sends a hard stop to all products and pauses a running sequence.

How do I start several pumps at exactly the same time?

Use Delay on each pump (or Delay lines in a sequence), then General GO: all pending commands start together. On an RS485 bus this is a single broadcast frame.

Why does broadcast behave differently depending on the communication setup?

When multiple instruments are detected on one COM port, the app creates an RS485 broadcast target. When multiple COM ports are used, the app sends commands to each product individually.

Why does the log say 499.93 µL/min when I typed 500?

The pump only accepts integer speed units; 500 µL/min is not exactly representable. The field keeps 500, the pump runs at the nearest value, and the log tells you which one (section 2.5).

Where can I find logs?

Recent logs are shown in the log terminal. One file per launch is written in %APPDATA%\AMF\AMF Control Center\logs (open it with the folder icon of the terminal header). The export icon builds a support report to send to AMF.

What happens when I skip a sequence line?

Skipped lines are kept in the file but not executed. If a currently running Wait line is skipped, the wait is cancelled and the sequence advances.

What does the Loop value mean?

Iterations is the total number of passes of the block from “From line” to the Loop line: 1 = once (no repetition), 3 = three times.

What does Build check, and do I have to use it?

Build simulates the sequence with the real state of the products and reports errors (a move that would overflow the syringe, a wrong port, a target not initialized, a delayed command that would never run) and warnings. It is optional: Play never checks anything, but a clean build avoids most run-time errors.

The sequence runs but I lost the running line in a long list. How do I get back to it?

Click the “Line n running” button at the bottom of the list. The list follows the running line by itself until you scroll or edit.

Can I change values while the instrument is running?

On-the-fly behavior is available for pumps with firmware version L1.1.1 or newer: a new speed is applied immediately, a move in the same direction continues from the current position.

How do presets work?

Presets save connected product unique IDs, names, slots, valve configuration, and valve port names. They do not store COM ports, sequences, current motion settings, or custom command history.

Why did my μ L command to ALL send individually?

The Sequence view intentionally sends ALL-target μ L commands individually to connected pumps so each pump can use its own syringe size.

What should I do after changing the valve configuration?

The application logs that the product should be initialized. Click Init after changing the valve configuration.

Does the application need an Internet connection?

No. It never connects to the network by itself; only the links of the ? menu open the web browser.

What is kept when I uninstall or update?

Presets, sequences, logs and the activation key live in %APPDATA%\AMF\AMF Control Center and are kept by updates and by the uninstaller.

7. Glossary

Term	Definition
COM port	A serial communication port used by the application to communicate with instruments.
USB	Direct serial-over-USB communication, detected together with RS232 as USB/RS232.
RS232	Serial communication through an RS232 or serial adapter, detected together with USB as USB/RS232.
RS485	Multi-drop serial communication that can support multiple addressed instruments on one bus.
Broadcast	Sending a command to an internal all-products target, used for multiple RS485 instruments on one COM port.
Sequential communication	Commands are queued and dispatched one at a time by the application.
Non-blocking command	A command is sent without waiting for the full instrument motion to finish.
Delay	Halt-encoded command loaded in the product, executed later by General GO. One per product.
Clear	Product card action that sends <code>c0R</code> and cancels the pending delayed command.
General GO	Global command that executes all pending delayed commands (<code>R</code>).
General STOP	Global hard stop (<code>T</code>) that also pauses sequence execution.
Homing / Init	Initialization of a product (<code>ZR</code>): valve and plunger move to their reference position. Required before any move, after a power cycle or a firmware update.
Firmware	Software running on the instrument, displayed in each product Settings panel.
Serial number	Identifier of the product (P100-..., P200-...), read from its USB chip or its firmware.
Preset	Saved instrument naming, slot, valve configuration, and port-name setup.
Sequence	An ordered workflow made of sequence lines, saved as a <code>.json</code> file.
Sequence line	One step in a sequence, including action type, value, target selection, mode, and optional note.
Build	Simulation of the whole sequence from the real product state, reporting errors and warnings before the run.
Slot	A numbered visual position from 1 to 6 used to identify instruments in both views.
Product card	The Instrument view panel representing one instrument or slot.
Mini product card	The Sequence view top-bar representation of an instrument slot.
Syringe position	The pump plunger position displayed as steps and calculated volume.
Valve position	The selected valve port shown in the valve visual.
Pull and Wait	Sequence behavior that waits until selected instruments are no longer busy before continuing.
Loop	Sequence action that repeats a block of lines a total number of times (Iterations).
u / V	Firmware speed units of the pump. Flow rates in $\mu\text{L}/\text{min}$ or mL/min are converted to <code>u</code> .
Steps	Motor steps of the plunger: 3000 per stroke in Standard resolution, 24000 in High resolution.
Support report	Zip archive of the log files, exported from the log terminal header for AMF support.

Appendix A. Example sequences

AMF Control Center is delivered with three example sequences. They are real, runnable sequences written for two AMF pumps, and every line carries a note that explains what it does and which rule it illustrates. Use them to discover the Sequence view, as templates for your own protocols, or as a reference to give to an AI assistant that will write a sequence for you. The tables below are read from the files installed with this version.

File	What it shows	Duration
1_synchronized_start.json	Two pumps load from different ports, then start dispensing at the same instant with Delay + General GO , at two different flow rates.	about 3 min
2_continuous_perfusion.json	Two pumps in relay: one delivers while the other refills and gets ready, in a Loop ; each delivery starts the instant the previous one ends.	about 7 min (3 passes)
3_reference_all_actions.json	Every action type used once, one rule per line: the reference to read and to give to an AI assistant.	about 2 min + one Pause

A.1 Where the examples are

- Open the Sequence view, click **Sequence Options** then **Load Sequence**: the file dialog opens in your sequences folder.
- The folder is %APPDATA%\AMF\AMF Control Center\Sequences. The three example files are copied there the first time the application starts, together with 2_continuous_perfusion_export.py, the Python export of the second example (section 4.14). They are never overwritten afterwards: you can edit them, and you can restore the originals by deleting them and restarting the application.
- A saved sequence is a plain .json text file: you can open it in any text editor, send it by e-mail, or paste it into an AI assistant (see A.6).

A.2 Before you run an example

1. Connect **two pumps** and detect them with **DETECT & INIT**. The examples assume **1000 µL syringes** in standard resolution; with another syringe, divide or multiply the volumes accordingly (the note on the first line of each example reminds you).
2. In the Sequence view, make sure the two pumps are in **slots 1 and 2** of the top bar. If not, drag the mini product cards to swap slots.
3. Connect the tubing as described in the first note of the example (which port is the reservoir, which port is the outlet). Ports are numbered clockwise on the valve drawing of the Instrument view.
4. Click **Build** (the wrench icon). Build replays the sequence step by step without moving anything and reports what would fail: a wrong port number, a syringe overflow, a pump that is not initialised. All three examples build without any remark on the hardware above.
5. Press **Play**. The running line is highlighted in orange and the list follows it; if you scroll away, a **Line n running** button appears at the bottom to bring you back.

Syringes are moved for real. Run the examples with empty syringes or with water first, and keep the **STOP** button in reach. Pull and Wait lines and General GO lines wait for the instruments: a full syringe at 500 µL/min takes 2 minutes.

A.3 Example 1: Synchronised start

Two pumps each aspirate a full syringe from their own port (pump 1 on port 2, pump 2 on port 3), switch to port 4 and empty their syringe at the same instant, pump 2 twice as fast as pump 1. This is the typical pattern of a droplet generator: a continuous phase and a dispersed phase that must start together but do not run at the same flow rate.

What the example teaches:

- **Pull and Wait** waits until the selected instruments are no longer busy, whatever made them busy: a valve move, a plunger move or a settings command. Use it whenever the next line depends on the previous one being finished: after a valve move before the plunger moves, after a Pickup or Dispense before the next command to the same pump.
- **Delay** loads a command in a pump without executing it; the line turns yellow. Each pump holds one pending command at most.
- **General GO** releases all pending commands with a single broadcast frame: the two plungers start together. With **Pull and Wait** on the General GO line, the sequence waits until both pumps have finished.
- A **speed** line is always sent immediately (Send mode): set the speed first, then delay the move.

#	Target	Action	Value	Mode	P&W	What the line does
1	1 + 2	Homing		Send	yes	Setup: slots 1 and 2 = two pumps with 1000 uL syringes. Homing initialises both plungers (position 0) and Pull and Wait holds the sequence until they are home. Harmless if DETECT & INIT already did it.
2	1 + 2	Speed: $\mu\text{L}/\text{min}$	1000	Send		Same aspiration speed for both pumps. A speed in $\mu\text{L}/\text{min}$ is converted for each pump from its own syringe size, so the two lines below move at the same flow rate.
3	1	Valve Port Selection	2	Send	yes	Pump 1 aspirates from its port 2. Pull and Wait (clock icon) waits until the selected instrument is no longer busy: use it after any valve move, plunger move or settings command when the next line depends on it. Here the valve must have finished turning before the plunger moves.
4	2	Valve Port Selection	3	Send	yes	Pump 2 aspirates from its port 3.
5	1	Absolute [μL]	1000	Send		Fill pump 1 completely (1000 uL syringe: adapt the value to yours). No Pull and Wait here: the next command targets the other pump, so both syringes fill at the same time.
6	2	Absolute [μL]	1000	Send		Fill pump 2, started a few milliseconds after pump 1.
7	1 + 2	Pull and Wait		Send		Pull and Wait as an action: the sequence waits until both selected pumps have stopped. Never move a valve while its plunger is still travelling.
8	1	Valve Port Selection	4	Send	yes	Both pumps dispense through their port 4 (for example the two inlets of a droplet junction).
9	2	Valve Port Selection	4	Send	yes	Same for pump 2.
10	1	Speed: $\mu\text{L}/\text{min}$	500	Send		Dispense speeds differ: pump 1 (continuous phase, e.g. oil) at 500 $\mu\text{L}/\text{min}$. Speed lines stay in Send mode: a pump only keeps ONE delayed command, so the speed is set now and the move is delayed below.
11	2	Speed: $\mu\text{L}/\text{min}$	1000	Send		Pump 2 (dispersed phase, e.g. water) twice as fast: 1000 $\mu\text{L}/\text{min}$.
12	1	Absolute [μL]	0	Delay		Delay = the command is loaded into the pump but NOT executed yet. The line turns yellow (command pending). Pull and Wait is not available on a delayed line.
13	2	Absolute [μL]	0	Delay		Second pending command, in pump 2. Each pump now holds exactly one delayed move.
14	ALL	General GO		Send	yes	General GO releases every pending command at once: both plungers start together (a single broadcast frame on an RS485 bus). Pull and Wait: the sequence ends when both syringes are empty, pump 2 after 1 min, pump 1 after 2 min.

Variations: change the two dispense speeds to your own flow rates; replace the two Absolute [μL] 0 lines by Dispense [μL] lines to deliver only part of the syringe; add a Loop line at the end that points back to line 3 to repeat the fill-and-dispense cycle.

A.4 Example 2: Continuous perfusion

Two pumps share the same reservoir (port 1) and the same outlet (port 2). While pump 1 delivers at the perfusion flow rate, pump 2 refills four times faster and moves to the outlet; the sequence then waits for the end of pump 1 only, and pump 2 starts delivering on the very next line. The dead time between two deliveries is a single command, without General GO.

What the example teaches:

- A **targeted Pull and Wait**: the Pull and Wait action on pump 1 alone lets the sequence wait for the delivering pump while the other one is being prepared.
- Lines **without** Pull and Wait let the sequence go on while a pump runs: that is how the refill of one pump happens during the delivery of the other.
- A **Loop** line repeats a block of lines. **Iterations** is the total number of passes: 3 means the block runs 3 times, 1 means once (no repeat). The **From line** field is the first line of the block, here line 11. During execution the label shows the pass in progress, Iterations (1/3) to (3/3). The state at the end of the block must be the state at its start: here pump 1 delivering, pump 2 full on the reservoir.

#	Target	Action	Value	Mode	P&W	What the line does
1	1 + 2	Homing		Send	yes	Setup: slots 1 and 2 = two pumps with 1000 uL syringes; on both valves port 1 = reservoir, port 2 = outlet. Homing puts both plungers at 0. Relay principle: one pump delivers while the other refills and gets ready on the outlet, so the next delivery starts the instant the previous one ends.
2	1 + 2	Speed: $\mu\text{L}/\text{min}$	4000	Send		Refill speed: 4000 $\mu\text{L}/\text{min}$, 15 s per syringe, four times the delivery flow rate.
3	1	Valve Port Selection	1	Send	yes	Priming: both pumps fill from the reservoir once. Pull and Wait after the valve move: the plunger must not move while the valve turns.
4	1	Absolute [μL]	1000	Send		Pump 1 fills. No Pull and Wait: pump 2 is prepared in the meantime.
5	2	Valve Port Selection	1	Send	yes	Pump 2 on the reservoir.
6	2	Absolute [μL]	1000	Send		Pump 2 fills at the same time as pump 1.
7	1 + 2	Pull and Wait		Send		Wait until both syringes are full. Pull and Wait waits until the selected instruments are no longer busy, whatever made them busy: valve move, plunger move or settings command.
8	1	Valve Port Selection	2	Send	yes	Pump 1 on the outlet, ready to deliver.
9	1	Speed: $\mu\text{L}/\text{min}$	1000	Send		Delivery flow rate: 1000 $\mu\text{L}/\text{min}$, 1 min per syringe. Change this value (and the same one for pump 2) to set your perfusion rate.
10	1	Absolute [μL]	0	Send		Pump 1 delivers. No Pull and Wait on this line: the sequence goes on while pump 1 runs, to get pump 2 ready. The loop below returns to the next line, with pump 1 delivering and pump 2 full on the reservoir port, exactly this state.
11	2	Valve Port Selection	2	Send	yes	LOOP START. Pump 2 (full) moves to the outlet while pump 1 delivers.
12	2	Speed: $\mu\text{L}/\text{min}$	1000	Send		Delivery speed for pump 2.
13	1	Pull and Wait		Send		Wait for the END OF PUMP 1 ONLY (target 1). Pump 2 is ready on the outlet: the dead time between the two deliveries is a single command.
14	2	Absolute [μL]	0	Send		Pump 2 delivers immediately. No Pull and Wait: pump 1 is refilled during this minute.
15	1	Valve Port Selection	1	Send	yes	Pump 1 back to the reservoir.
16	1	Speed: $\mu\text{L}/\text{min}$	4000	Send		Fast refill speed.
17	1	Absolute [μL]	1000	Send	yes	Pump 1 refills in 15 s. Pull and Wait: the plunger must be stopped before the valve moves again.
18	1	Valve Port Selection	2	Send	yes	Pump 1 back on the outlet, ready long before pump 2 is empty.
19	1	Speed: $\mu\text{L}/\text{min}$	1000	Send		Delivery speed again for pump 1.
20	2	Pull and Wait		Send		Wait for the end of pump 2 only.
21	1	Absolute [μL]	0	Send		Pump 1 delivers immediately, no Pull and Wait: pump 2 is refilled during this minute.
22	2	Valve Port Selection	1	Send	yes	Pump 2 back to the reservoir.
23	2	Speed: $\mu\text{L}/\text{min}$	4000	Send		Fast refill speed.
24	2	Absolute [μL]	1000	Send	yes	Pump 2 refills; when it is full we are back in the loop start state: pump 1 delivering, pump 2 full on the reservoir.
25	ALL	Loop	3 passes	From line 11		Loop: the block from line 11 ('From line') down to here runs 3 times in total (Iterations = 3), i.e. 6 deliveries plus the first one, about 7 min of continuous flow. 1 would mean once, no repeat. Set the value to the number of passes you need.
26	1	Pull and Wait		Send		End: wait for the last delivery of pump 1 before the sequence completes.

To set your own flow rate, change the 1000 $\mu\text{L}/\text{min}$ lines (lines 9, 12 and 19). To set the duration, change the Loop value: each extra pass adds two syringes, 2 minutes at 1000 $\mu\text{L}/\text{min}$. For an endless perfusion, use a large value and stop the sequence with the Stop button. The Python export of this example, ``2_continuous_perfusion_export.py``, is installed next to it (section 4.14).

A.5 Example 3: Reference sequence

This sequence is not a protocol. It uses every action type of the Sequence view exactly once, on two pumps, and its notes state the rule behind each line: units, relative and absolute moves, valve control, custom commands, Delay and General GO, Wait, Pull and Wait, Loop, Pause and skipped lines. It runs in about two minutes and stops once at a Pause line: press Play to finish it.

#	Target	Action	Value	Mode	P&W	What the line does
1	ALL	Homing		Send	yes	REFERENCE SEQUENCE: every action type once, with the rule it illustrates. Runs on two pumps (slots 1 and 2, 1000 uL syringes, standard 3000-step resolution). Target ALL = every connected product. Homing sends ZR and Pull and Wait waits for the end.
2	ALL	Speed: $\mu\text{L}/\text{min}$	1000	Send		Speed in $\mu\text{L}/\text{min}$ for ALL pumps: converted per pump from its syringe size and sent one pump after the other. The value you type is kept; the exact value the pump can do is written in the log.
3	1	Speed: V	100	Send		Speed in V: raw firmware unit (integer). With a 1000 uL syringe 1 V = 20 $\mu\text{L}/\text{min}$, so 100 V = 2000 $\mu\text{L}/\text{min}$.
4	2	Speed: u	13422	Send		Speed in u: finest firmware unit, 1 V = 134.217728 u. 13422 u = 100 V = 2000 $\mu\text{L}/\text{min}$ with a 1000 uL syringe.
5	1	Speed: mL/min	0.5	Send		Speed in mL/min (up to 4 decimals). 0.5 mL/min = 500 $\mu\text{L}/\text{min}$.
6	1	Valve Port Selection	1	Send	yes	Pump valve to port 1. Port numbers run clockwise on the valve drawing of the Instruments tab. Pull and Wait: never move the plunger while the valve turns. The same rule holds after a plunger move or a settings command: without Pull and Wait the next line reaches a busy instrument.
7	1	Pickup [μL]	200	Send	yes	Pickup = relative aspiration: +200 uL from the current position. Pull and Wait waits for the end of the move before the next line.
8	1	Valve Port Selection	2	Send	yes	Valve to port 2 before dispensing.
9	1	Dispense [μL]	100	Send	yes	Dispense = relative: -100 uL. The plunger is now at 100 uL. The Build checks that no move goes below 0 or above the syringe volume.
10	2	Pickup [steps]	600	Send	yes	Same actions in motor steps: 600 steps = 200 uL with a 1000 uL syringe in standard resolution (3000 steps per stroke; 24000 in high resolution).
11	2	Dispense [steps]	300	Send	yes	-300 steps = -100 uL.
12	1 + 2	Absolute [steps]	0	Send		Absolute = go to a position (here 0 = empty). Several targets on one line: the command is sent to slot 1 then slot 2, a few milliseconds apart. No Pull and Wait here, the next line does it.
13	1 + 2	Pull and Wait		Send		Pull and Wait as an action: waits until every selected product has stopped. Equivalent to the clock icon, but usable after several lines started together.
14	ALL	Wait	2 s			Wait: the sequence pauses for 2 s, nothing is sent to the instruments. Units: ms, s, min. Loop start for the loop below.
15	1	Custom Command	B1R	Send	yes	Custom Command: raw firmware command string, here B1R = valve to port 1 (see the firmware manual). Only for commands that have no action type; the Build simulates P, D, A, V, U, u, B, N, Z, W and @SYRINGE=, and warns for the others.
16	1	Absolute [μL]	500	Delay		Delay: the move is loaded in pump 1 but not executed (line turns yellow). One pending command per pump: a second Delay on the same pump replaces it, a Send on the same pump cancels it.
17	2	Absolute [μL]	500	Delay		Pending command in pump 2.
18	ALL	General GO		Send	yes	General GO executes all pending commands at the same time (one broadcast frame on an RS485 bus). Pull and Wait waits for both pumps. This is how to start several pumps together.
19	ALL	Loop	2 passes	From line 14		Loop: the block from line 14 (the 'From line' field) down to here runs 2 times in total (Iterations = 2; 1 = once, no repeat). During execution the label shows the progress, e.g. Iterations (1/2). On the second pass both pumps are already at 500 uL, the moves do nothing.
20	ALL	Pause		Send		Pause: the sequence stops here and the Play button returns to Play. Use it for a manual step (change a vial, read a value), then press Play to continue.
21	1	Speed: $\mu\text{L}/\text{min}$ (skipped)	2000	Send		Skipped line (grey, skip button): kept in the file but not executed. Handy to disable a step without deleting it.
22	1 + 2	Absolute [μL]	0	Send	yes	Back to empty syringes: end of the reference sequence.

A.6 Writing your own sequence with an AI assistant

Because a sequence is a small text file with self-explanatory notes, an AI assistant such as Claude can write one for you from a description of your protocol. The reference sequence gives it the format and the rules.

1. Open `3_reference_all_actions.json` in a text editor and copy its whole content.
2. Describe your protocol: which pump does what, ports, volumes, flow rates, timing, what must happen in parallel, how many repetitions.
3. Ask for a file in the same format, then save the answer as a `.json` file in your sequences folder.
4. Load it in AMF Control Center and click **Build**. Fix what Build reports (with the assistant if you like) until the build is clean, then run it with empty syringes once before using real liquids.

Prompt to start from:

Here is an example sequence file for AMF Control Center (JSON). Every line has a comment that explains the rule it illustrates. Write a new sequence in exactly the same format for the following protocol:

- Hardware: 2 AMF pumps in slots 1 and 2, 1000 μL syringes, standard resolution; pump 1 valve: port 1 = buffer, port 2 = sample, port 3 = outlet; pump 2 valve: ...
- Protocol: ... (volumes, flow rates, order, what runs in parallel, repetitions)

Rules: use Pull and Wait whenever the next line needs the previous move or setting to be finished (valve moves, plunger moves, settings); use Delay + General GO for moves that must start together; never exceed the syringe volume; one pending command per pump; a Loop value is the total number of passes (1 = once); keep "format": 2 at the root; comment every line. Return only the JSON.

Fields of a sequence line

Field	Values	Meaning
display_order	1, 2, 3 ...	Position of the line. Lines are played in this order.
selected_product	"All" or a list of slot numbers as strings: ["1"], ["1", "2"]	Target slots of the top bar. General GO, Wait, Loop and Pause always use "All".
action_type	Speed: V · Speed: u · Speed: $\mu\text{L}/\text{min}$ · Speed: mL/min · Absolute [μL] · Absolute [steps] · Pickup [μL] · Dispense [μL] · Pickup [steps] · Dispense [steps] · Valve Port Selection · Homing · Custom Command · General GO · Wait · Pull and Wait · Loop · Pause	Exactly one of these strings (μ is written <code>\u00b5</code> in the file).
action_value	number as a string, port number, command text, or ""	Speed, volume, position, port, raw command, Wait duration, or total number of passes for Loop (1 = once).
mode	"Send" or "Delay" · Wait: "ms", "s", "min" · Loop: the 'From line' number	Delay = load without executing, released by General GO (not for valves, Homing, Wait).
PullandWait	"True" or "False"	Wait for the targets to stop before the next line. Not available on a Delay line.
skip_flag	true or false	A skipped line is kept but not executed.
loop_number	integer, same as action_value for Loop lines, else 0	Total number of passes of the loop.
comment	free text	The note shown by the note button of the line. Comment every line.
format (file root)	2	Written next to "sequence" at the root of the file. Files without it use the previous loop convention (value = extra passes) and are converted when loaded: each Loop value is increased by 1 and the log says so.

A.7 Rules to remember

- **Initialise** the pumps (DETECT & INIT, or a Homing line) before any move: the firmware refuses to move a pump that is not initialised.
- **Pull and Wait after any action the next line depends on:** a valve move, a Pickup, a Dispense, an Absolute move, a settings command. It waits until the selected instruments are no longer busy. Without it, the next command reaches a pump that is still moving: a speed is applied on the fly, a move in the same direction restarts from the current position, anything else is an overflow. Leave it out on purpose only when you want the sequence to go on while the pump runs.
- **One pending command per pump.** A second Delay on the same pump replaces the first; a Send on the same pump cancels it. Set speeds with Send lines before the delayed move.
- **General GO + Pull and Wait** is the way to start several pumps together and to wait for all of them.
- **Loop, Iterations = N** means the block runs N times in total (1 = once); the block starts at the **From line** and ends at the Loop line.
- **Build first, run second.** Build sees your real syringe sizes and plunger positions; a clean build on your hardware is the best check before pressing Play.
- The **volumes** of the examples are written for 1000 µL syringes. µL/min speeds are converted for each pump from its own syringe size, so they need no change.