HardwareX 20 (2024) e00581

Contents lists available at ScienceDirect

HardwareX

journal homepage: www.elsevier.com/locate/ohx

Hardware Article

MULA, an affordable framework for multifunctional liquid automation in natural- and life sciences with a focus on hardware design, setup, modularity and validation

Leon F. Richter [1], Wolfgang R.E. Büchele [1], Alexander Imhof, Fritz E. Kühn *

Technical University of Munich, TUM School of Natural Sciences, Department of Chemistry and Catalysis Research Centre, Molecular Catalysis, Lichtenbergstr. 4, Garching bei München, Germany

A R T I C L E I N F O

Keywords:
3D printing
Liquid handling
Micro-syringe
Chemistry
Biology
Pipetting

A B S T R A C T

The implementation of automation has already had a considerable impact on chemical and pharmaceutical industrial laboratories. However, academic laboratories have often been more reluctant to adopt such technology due to the high cost of commercial liquid handling systems, although, in many instances, there would be a huge potential to automate repetitive tasks, resulting in elevated productivity. We present here a detailed description of the setup, validation, and utilization of a multifunctional liquid automation (MULA) system that can be used to automate various chemical and biological tasks. Considering that such a setup must be highly customizable, we also designed MULA with respect to modularity, providing detailed insight as far as possible. Including all 3D-printed parts and the used Hamilton gastight micro syringe, the total construction cost is approximately 700 €. This allows us to achieve a highly reliable and accurate system that exceeds the precision of a classical air displacement pipette while still retaining the ability to use closed vial (septa) setups. To encourage other groups to adopt this setup, detailed instructions and tips for every step of the process are provided, along with the complete CAD design of MULA and control code, which are freely available for download under the CC BY NC 3.0 license.

Specifications table.

Hardware name	MULA (Multifunctional liquid automation)
Subject area	• Engineering and materials science • Chemistry and biochemistry • Medical (e.g., pharmaceutical science) • Biological sciences (e.g., microbiology and biochemistry)
Hardware type	• Biological sample handling and preparation • Syringe manipulation

(*continued on next page*)

* Corresponding author.
E-mail address: fritz.kuehn@ch.tum.de (F.E. Kühn).
[1] Equally contributing authors.

https://doi.org/10.1016/j.ohx.2024.e00581
Received 2 July 2024; Received in revised form 25 August 2024; Accepted 30 August 2024
Available online 1 September 2024

(*continued*)

Hardware name	MULA (Multifunctional liquid automation)
	• Liquid and gas handling
Closest commercial analog	No commercial analog is available.
Open source license	CC BY NC 3.0
Cost of hardware	700 €
Source file repository	https://doi.org/10.17632/3m3t4f9ft3

1. Hardware in context

In the past decade, the development of automated systems has become increasingly important in daily laboratory operations, particularly for highly repetitive experiments like high throughput liquid handling work, which are prone to errors resulting especially from human influence.[1–6] A significant challenge faced by chemists and life scientists is achieving reproducible results.[7,8] A *meta*-analysis published in 2015 for the years 2011 to 2014 revealed that less than half of the data produced is reproducible, leading to substantial costs and time-consuming experiments, especially in drug research.[9] In the United States alone, over 28 billion[8,9] dollars are spent on non-reproducible preclinical research. However, even small enhancements in reproducibility or consistency could yield significant returns on investments in terms of cost reductions and time spent on drug development [8].

One potential solution is the deployment of liquid-handling robots, which offer consistent performance and operate 24/7. These robots boost throughput, document each step, lower labor costs, ensure a safer lab operation, and provide high precision and accuracy.[1,4] While companies like *Hamilton*, *Chemspeed Technologies*, *Mettler Toledo*, *Tecan* and *AmigoChem* already offer such robots, they require a significant budget and space for implementation and are optimized for rather specialized tasks.[4,10–12] If the assignment changes often, those robots are rarely appropriate and lack flexibility for easy modification or variation. Additionally, personnel must be trained for each company's specific system, as competing systems usually have no compatibility. In addition, technical support tends to prioritize larger industries more often than smaller university labs. When these robots require maintenance, they are usually not easy to fix, and the costs can be substantial, creating a significant burden for smaller groups with limited budgets. Furthermore, the customers can be very frustrated about the large downtimes of machines that are crucial for experiments. Simultaneously, there has been a rise in the use of G-code devices, including 3D printers, computer numerical control (CNC) routers, and laser engravers, which can achieve linear movements in a simple manner. Thanks to the RepRap project, the cost of these devices has significantly decreased in the last 15 years, making them accessible even to the general consumer.[13] Their potential for scientific purposes, including chemical synthesis[14], simplifying chromatographic processes[15], liquid handling[16], and fabricating functional materials[17] is a testament to their versatility and cost-effectiveness. On the other hand, commercial systems for liquid handling are often too expensive for the average academic research laboratory and lack the individual customization options of do-it-yourself (DIY) systems. Alternatives for specific experiments based on open-source 3D printing technology[18–20] have emerged, which are pretty similar to commercial ones; however, due to the open-source basis, the costs and makes are reduced, and the possibility for specially tailored robots for specific experiments is possible. By investing in versatile and cost-effective alternatives, laboratories can foster innovation while minimizing financial strain. Ultimately, embracing a more adaptable approach to automation will enable researchers to focus on discovery and innovation rather than logistical constraints and repetitive work, e.g., pipetting of defined volumes. Complementing already existing DIY liquid handling systems such as EvoBot[21], FINDUS[22], BioCloneBot [23] and OSMAR[24], we started developing MULA (Table 1). MULA represents a step forward in laboratory automation, offering a mixture of affordability, flexibility, highly customizable, and user-friendliness. It is designed for the needs of smaller labs and academic institutions in mind. The high customizability of MULA allows researchers to make easy modifications for new procedures or experiments without the need for costly overhauls or specialized training. In this manuscript, we present a detailed documentation of MULA and its setup, validation, and utilization as a multifunctional liquid automation robot with the intent of helping researchers focus on discovery and advancement without the constraints of high costs and inflexible equipment.

2. Hardware description

As the name implies, MULA (**mu**ltifunctional **l**iquid **a**utomation) is a highly modular and customizable framework that can be built

Table 1
List of DIY systems already published and compared to this work.

Name, reference	**OSMAR**[24]	**EVO-BOT**[21]	**FINDUS**[22]	**BioCloneBot**[23]	**MULA (this work)**
Structural parts made *via*	Laser-cutting	Laser-cutting	3D-printing	3D-printing	3D-printing
Mainboard, Firmware	MKS Gen-L, custom Marlin	MEGA 2560 R3, custom Marlin	ESP8266 12F, custom	MKS Gen-L, custom	BTT Octopus, custom Marlin
Control *via*	AutoIt, Hype!terminal	Python script on Raspberry Pi	Python script	Custom C# frontend	Python script, Pronterface
Fluid dispensing *via*	Microsyringe & Needle	Plastic-syringe & Needle	Pipette & Tip	Microsyringe & Tip	Microsyringe & Needle
Proposed cost	$700AU	$600	$400 + pipette	<$2000CAD	~ 700€

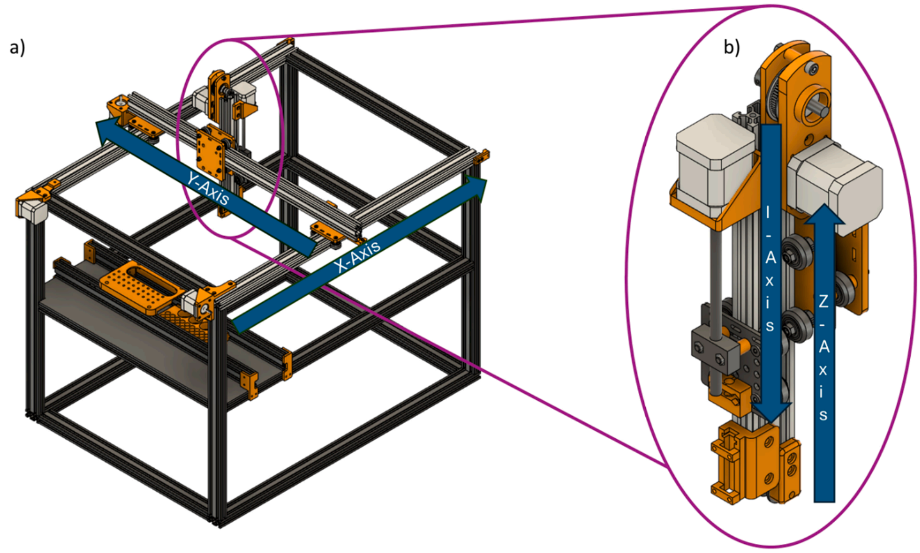
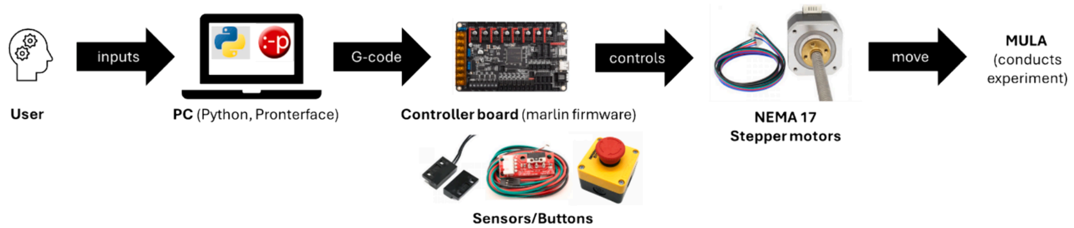


Fig. 1. CAD model of MULA: a) assembled frame with 3D-printed parts (orange) and 2040 V-slot profiles (white); b) Sampling head of MULA. For visibility reasons, some parts, like the electronics board case, are not depicted in this illustration. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)



Scheme 1. Command-chain overview of MULA: The user inputs parameters to the PC, which are transformed into G-code commands, which the software then sends to the control board. The firmware on the board uses these instructions to control the stepper motors, which make MULA move accordingly and conduct the experiment.

in various sizes and for different applications. The robot is built with a cartesian motion system, which is the most common for DIY and commercial liquid handling systems, although there are some interesting exceptions [25]. The system has four movable axes. X- and Y-axes are both horizontal and use a timing belt mechanism, with the X-axis being duplicated. The two remaining axes are both vertical, using a lead screw (I-axis) and a geared timing belt (Z-axis) mechanism. All axes use 2040 V-slot profiles and V-slot wheels to support linear motion. We have decided to use a timing-belt mechanism with a 3:1 gear ratio for the Z-axis to combine the Z- and I-axis on one 2040 V-slot profile, making the head very compact. Fig. 1 depicts this elegant approach, which makes this system easy to assemble and more compact than other DIY approaches.[24] The X-, Y-, and Z-axes control the movement of the syringe, while the I-axis controls the movement of the syringe plunger. Since we have constructed MULA like a 3D printer, we wońt discuss the accuracy and reliability of the timing belt mechanisms of the X-, Y- and Z-axis but instead focus on the liquid handling ability. MULA contains a *Hamilton* micro syringe with a removable needle instead of an air displacement pipette since we plan to use this system with closed vials (septa) and notice during testing that air displacement pipettes are not ideal when using non-aqueous liquids. As a bonus, when using a gastight syringe, this system is also able to handle gases. We configured MULA with a sampling area measuring 41 cm on the Y-axis, 56 cm on the X-axis, and 10 cm on the Z-axis, although this can be easily adapted as described later. To adapt to different experimental conditions, we have designed a modular rack system that slides into regular aluminum profiles. The template of this rack as well as the rack-top part is supplied as a.step file and can easily be adjusted by common CAD tools. We provide a template for a 30-vial rack for GC vials. *NEMA* 17 stepper motors are employed on all axes due to their good availability and documentation. The BTT Octopus board with custom marlin firmware is used via a computer for motion control to run the stepper motors. Combining this specific board with the used *Trinamic* 2209 drivers allows to use a sensorless homing procedure for all axes except the Z-axis, reducing the number of limit switches and the associated wiring.[2] Sensorless homing is also achieved for the syringe plunger (I-axis), which improves reproducibility and user experience and reduces the need to manually recalibrate the position of the syringe plunger when an accident happens. In general, it can be assumed that the ability to home the machine is a significant safety and convenience feature, which we considered

[2] For more info visit https://marlinfw.org/docs/hardware/tmc_drivers.html (2024, June 26).

as missing on other DIY systems.[24] Furthermore, the board can control up to 8 individual stepper motors, allowing for further upgrades (as depicted in the outlook) since we currently only use 5 Motors (X_1, X_2, Y, Z, I). By using G-code instructions, the machine we designed is controlled using open-source and freely available software (Marlin, Pronterface), just like other machines such as 3D printers, CNC routers, and laser cutters. The complete process of communicating with MULA is depicted in Scheme 1.

While prototyping this machine, we developed several improvements over comparable DIY systems such as OSMAR and EVO-bot. Hamilton gastight syringes (although relatively costly) with removable needles make the system described in this work very versatile and durable, especially when dealing with liquids other than water (organic solvents like dichloromethane corrupt plastic syringes very quickly). Moreover, we want to highlight the high accessibility of this machine since all custom parts can be 3D-printed using a regular desktop 3D printer. There is no need for soldering or laser-cutting parts, which require harsh safety precautions and might not be accessible to smaller universities. Furthermore, our setup is compact, modular, user-friendly and simple to maintain and assemble.

In Summary:

- Accessible, simple to construct, easy to maintain and control
- Highly customizable
- Potential for further upgrades (see outlook)
- Cheap and modular system for automatic liquid handling

3. Design files

This section encompasses the 3D-printed components and firmware utilized in the construction and operation of MULA. As mentioned above, one advantage of our approach is that it does not require the use of sophisticated machines such as CNC routers and laser engravers. Instead, a basic desktop 3D printer is sufficient to create all custom parts.

3.1. 3D-printed parts

The files were printed with *Prusament* PETG on a *Prusa* I3 MK3S+ printer with a 0.6 mm nozzle. To increase the mechanical strength of all parts, the perimeter count was increased to 4 in *PrusaSlicer.* We printed the X- and Y-gantry and all stepper motor mounts with 80 % infill and 0.3 mm layer height. All other parts were printed with 40 % infill and 0.3 mm layer height. Besides all parts' .stl and .step files, we also provide the .3mf files (ready to print) with our settings in the repository. The 3D-printed parts make up a large part of the machine we developed. We have designed all 3D-printed parts to limit the number of different screw sizes where possible. Therefore, M4x10 and M3x10 screws are used to mount compounds to the aluminum profiles and attach stepper motors, respectively. There are three principal categories of parts:

- Mount: Those parts are mounted to the frame to support relevant mechanical or electronic structures
- Gantry: Those parts are incorporated in timing belts and are needed for the movement of the axes
- Spacer: Those parts are needed to fill gaps between other parts

Gantry_X contains several holes to mount the V-slot wheels and the timing belt and connect the Y-axis profile. Gantry_Y is a slightly more complex part to attach the V-slot wheels for the Y-axis, the timing belt of the Y-axis, but also the V-slot wheels for the Z-axis and the timing belt of the Z-axis. Gantry_Y_mount incorporates a mount for a cable drag chain. Furthermore, there are four mounts for the stepper motors (X_left, X_right, Y, I), accounting for different axial mounting conditions. Although they have a similar design, they are clearly distinguishable from each other.

For the Z-axis, a gearbox-like mount is utilized for the stepper motor, which is built from the two Z_Gearbox parts (A and B). Next, since the timing belt must be connected to an idler pulley at the other end of the axis, the Idler_mount part is needed for each one of the four axes employing a timing belt mechanism. The mounting of the micro syringe the frame is conducted using three parts: Syringe_mount is a parametric part that can be adapted for different syringe sizes and keeps the main body of the syringe connected to the I-, Z-axis profile; Plunger_mount, on the other hand, is another parametric part responsible for securing the plunger of the syringe to the I-axis gantry (there are also two versions for plungers with or without threads); Lastly, the Syringe_bracket parts are used to secure the syringe body to the Syringe_mount part. Then, there are 4 different parts needed for the pipetting area: 30Vial_rack and 96Well_rack, which contain the sample and solvent vials. They slide into the pipetting area profiles and can easily be customized. 30Vial_top is mounted on top of the pipetting area profiles and is needed to keep the vials in place when working with septa. The two parts named Rack_mount (A and B) are needed to connect the pipetting area profiles to the main frame and thus ensure a consistent location of all vials during the experiments. The small part labelled Endstop_mount is employed on the Z-Axis to mount the endstop switch to the Z-Axis profile. Lastly, the spacer parts (M5_Spacer_6, M5_Spacer_7, M4_Spacer_12 and M5_Spacer 14) are needed for encapsulating screws in different parts of the machine.

3.2. Software

3.2.1. Calculation of important parameters

Several parameters must be specified in the original marlin firmware when building a custom variant of MULA. While a large part of

Table 2
Software and hardware parameters and resulting calculated steps/mm for all axes of MULA.

Axis	Micro-stepping	Steps/Revolution (1.8° resolution)	Slope (lead screw)	Motor pulley teeth (T_m)	Distance teeth GT2 belt (b)	Gearbox ratio (r)	Calculated steps/mm
X,Y	16	200	–	20	2 mm	1	80
Z	16	200	–	20	2 mm	3	240
I	16	200	8 mm	–	–	1	400

Table 3
Custom modifications to the marlin firmware.

File	Line	Modification	Purpose
Config.h	174–188	Change the relevant (X, Y, Z, X2, I, E0) driver types to: **TMC2209**	Enables sensorless homing
	216	Comment (add "//" before): #define AXIS4_rotates	Use linear motion for the syringe plunger
	886, 894	Uncomment (remove "//" before): USE_IMIN_PLUG USE_ZMAX_PLUG	Enables endstops on I- and Z-axis
	964, 971	Change to: Z_MAX_ENDSTOP_INVERTING **true** Z_MIN_PROBE_ENDSTOP_INVERTING **true**	Invert logic of end-stop and probe to align with hardware setup
	1019, 1026, 1039, 1285	Change to: DEFAULT_AXIS_STEPS_PER_UNIT { **80, 80, 240, 400, 400**} DEFAULT_MAX_FEEDRATE **{ 5000, 5000, 2000, 200, 200}** DEFAULT_MAX_ACCELERATION **{ 500, 500, 300, 100, 100}** NOZZLE_TO_PROBE_OFFSET **{ 10, 10, 0, 0}**	To accommodate for the additional axis, enter the previously calculated steps/mm and set speed and acceleration
	1125	Comment (add "//" before): #define Z_MIN_PROBE_USES_Z_MIN_ENDSTOP_PIN	Enables homing on Z-axis
	1165	Uncomment (remove "//" before): FIX_MOUNTED_PROBE	Enables homing on Z-axis
	1412, 1424	Uncomment (remove "//" before): I_ENABLE_ON 0DISABLE_I false	Enables I-axis
	1444, 1485	Change to: INVERT_Z_DIR **true** Z_HOME_DIR **1**	Ensure that the Z-axis moves in the right direction
	1445, 1486	Uncomment (remove "//" before) and/or change to: INVERT_I_DIR **true** I_HOME_DIR −1	Enables correct homing for the syringe plunger
	1496, 1497, 1505	Change to: X_BED_SIZE **500** Y_BED_SIZE **350** Z_MAX_POS **75**	Sets the software limits for the sampling area; Can be disabled afterwards by the M221 G-code command (not recommended!)
	1506, 1507	Uncomment (remove "//" before) and change to: I_MIN_POS 0 I_MAX_POS **100**	Sets a software limit of 100 mm for the syringe plunger
	1887	Change to: HOMING_FEEDRATE_MM_M **{ (50*60), (50*60), (10*60), (10*60)}**	Sets the speed for the homing procedure and includes the plunger axis
Config_adv.h	500, 502, 506, 507, 513	Uncomment (remove "//" before) and/or change to: USE_CONTROLLER_FAN CONTROLLER_FAN_PIN **FAN2_PIN** CONTROLLERFAN_SPEED_ACTIVE **170** CONTROLLERFAN_SPEED_IDLE **30** CONTROLLER_FAN_EDITABLE	Configures the Fan that cools the stepper drivers
	773	Uncomment (remove "//" before): #define INVERT_X2_VS_X_DIR	To make the second X-axis motor turn in the right direction
	837, 838	Change to: HOMING_BUMP_MM **{ 0, 0, 3, 0}** HOMING_BUMP_DIVISOR **{ 2, 2, 4, 2 }**	Settings for sensorless homing; Only the non-sensorless Z-axis backoffs during homing
	844	Uncomment (remove "//" before): #define HOME_Z_FIRST	Ensures that the z-axis is up before homing other axes
	1014	Change to: AXIS_RELATIVE_MODES **{false, false, false, false, false}**	
	1039	Change to: DISABLE_INACTIVE_Z **false**	Ensures that the Z-axis wońt lower when inactive
	1264	Change to: MANUAL_FEEDRATE **{ 50*60, 50*60, 4*60, 4*60, 4*60}**	
	3078	Change to: CHOPPER_TIMING CHOPPER_DEFAULT_**24V**	
	3113, 3178	Uncomment (remove "//" before): MONITOR_DRIVER_STATUSSENSORLESS_HOMING	Enables sensorless homing
	3182, 3184, 3190, 3222	Uncomment (remove "//" before) and/or change to: X_STALL_SENSITIVITY **100** Y_STALL_SENSITIVITY **100** I_STALL_SENSITIVITY **100** TMC_DEBUG	Sets the sensitivity for sensorless homing; this can be changed afterwards using the M914 G-code command
Pins.h	54,	Change to: **I**_DIAG_PIN PG11	Enables the I-axis

(*continued on next page*)

Table 3 (*continued*)

File	Line	Modification	Purpose

	140 ff.	`140 #ifdef I_STALL_SENSITIVITY`
		`141   #define I_STOP_PIN          I_DIAG_PIN`
		`142   #if I_HOME_TO_MIN`
		`143     #define I_MAX_PIN        E3_DIAG_PIN  // PWRDET`
		`144   #else`
		`145     #define I_MIN_PIN        E3_DIAG_PIN  // PWRDET`
		`146   #endif`
		`147 #elif NEEDS_I_MINMAX`
		`148   #ifndef I_MIN_PIN`
		`149     #define I_MIN_PIN        I_DIAG_PIN  // Z-STOP`
		`150   #endif`
		`151   #ifndef I_MAX_PIN`
		`152     #define I_MAX_PIN        E3_DIAG_PIN  // PWRDET`
		`153   #endif`
		`154 #else`
		`155   #define I_STOP_PIN         I_DIAG_PIN  // Z-STOP`
		`156 #endif`
		`157`
		`158 #undef NEEDS_X_MINMAX`
		`159 #undef NEEDS_Y_MINMAX`
		`160 #undef NEEDS_Z_MINMAX`
		`161 #undef NEEDS_I_MINMAX`
			Enables the I-axis

	209 ff.	`209  #define I_STEP_PIN          PG4  // MOTOR 3`
		`210  #define I_DIR_PIN           PC1`
		`211 #ifndef I_ENABLE_PIN`
		`212   #define I_ENABLE_PIN       PA0`
		`213 #endif`
		`214 #ifndef I_CS_PIN`
		`215   #define I_CS_PIN           PC7`
		`216 #endif`
			Enables the I-axis

the settings will remain unchanged, there are some parameters that likely deviate. This includes the size of the sample area, the choice of stepper drivers (Note again that only TMC2209 and 2226 drivers support sensorless homing!), the sensitivity for sensorless homing and the default axis-steps per mm. Besides the dimension of the sample area (which is calculated in Chapter 5), another important parameter to specify in the firmware is the correct correlation between motor steps and mm. We want to demonstrate how we calculated this parameter in the following to make it as easy as possible to adapt the system to other possible configurations:

For the belt movements (X, Y, Z) we used the formula: $\frac{steps}{mm} = \frac{Steps/Revolution \times Microstepping}{T_m \times b} \times r$

For the movements of the lead screw (I), we instead used: $\frac{steps}{mm} = \frac{Steps/Revolution \times Microstepping}{Slope}$

Table 2 includes the used parameters and calculated steps/mm for our build. With the correct correlation between motor steps and mm, the movement of the X-, Y-, and Z-axis can be controlled precisely. Besides the different correlation of steps/mm, for the two different transmissions from rotational to linear motion (timing belt and lead screw), there are other factors to consider, such as longevity and accuracy. [26] The sensitivity for homeless probing (STALL_SENSITIVITY) was determined empirically. We found a value of 100 suitable for all axes, but this might differ in other builds.[3]

3.2.2. Modifications to the marlin firmware

In Table 3, we briefly comment on the parameters that we have changed from the original marlin configuration from the board manufacturer.

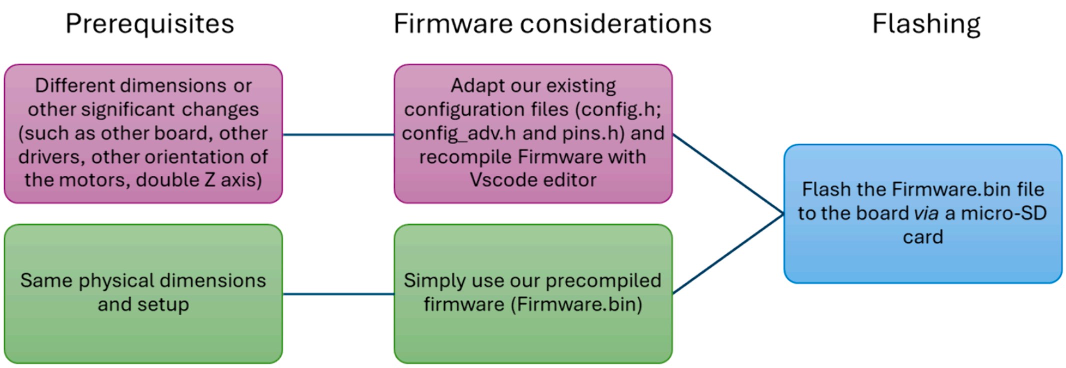
3.2.3. Flashing/Compiling of the firmware

MULA is controlled by a *BTT Octopus* 1.1 board, which is otherwise used to control DIY 3D printers. In order to send commands, the board requires a USB cable connection to a PC and the correct firmware. We recommend using a micro-SD card containing the Firmware.bin file according to the board's manual to flash the correct firmware to the Octopus board. The MULA marlin firmware build can be found in the repository. To reproduce our setup without significant changes, our Firmware.bin can be directly used (note that you can change many of the settings, such as the axis steps-per-unit *via* G-code commands in Pronterface: see https://marlinfw.org/meta/gcode/). However, if it is necessary to adjust the machine's dimensions or change the control board or stepper driver type, we advise recompiling the firmware from scratch using the VSCode editor and the platformIO IDE extension. Scheme 2 illustrates the decision tree for firmware flashing/compiling. A brief guide for setting up VSCode editor accordingly is available in the assembly manual, although this process is generally very well documented. Note that we have changed several parameters from the stock marlin configuration so that the user can adapt the firmware by changing the provided config.h and config_adv.h and the pins_BTT_OCTOPUS_COMMON.h files.

3.2.4. Prerequisites for Machine control on windows

When the correct firmware is flashed and the electronics are connected properly, the board is ready to receive G-code commands (like a 3D printer) from the computer *via* USB. In 3D printing, the G-code file is generated in the slicer, where a complex algorithm

[3] For more information visit https://marlinfw.org/docs/gcode/M914.html (2024, June 26).



Scheme 2. Decision tree for the steps to flash the firmware to the control board.

generates the G-code file layer by layer based on the object's geometry. In our case, we had to approach the G-code generation differently. Therefore, we created a primitive slicer equivalent with an intuitive GUI that allows for straightforward input of all relevant parameters and then generates the respective G-code file. We are providing those programs as executables (.exe) files to eliminate the need to establish a working Python environment, which might overcharge inexperienced users. However, we still provide the original Python scripts in the repository for experienced users. The free software Pronterface should also be installed (from https://github.com/kliment/Printrun/releases) for initial calibration, manual control and sending of the G-code files.

3.3. Design files summary

Design file name	File type	Open source license	Location of the file
MULA_complete	STEP / F3D (Fusion 360)	CC BY NC 3.0	Mendeley Data repository: https://doi.org/10.17632/3m3t4f9ft3
MULA_Parametric_syringe	STEP / F3D (Fusion 360)	CC BY NC 3.0	Mendeley Data repository
Gantry_X	STL	CC BY NC 3.0	Mendeley Data repository
Gantry_Y	STL	CC BY NC 3.0	Mendeley Data repository
Gantry_Y_mount	STL	CC BY NC 3.0	Mendeley Data repository
Stepper_mount_X_left	STL	CC BY NC 3.0	Mendeley Data repository
Stepper_mount_X_right	STL	CC BY NC 3.0	Mendeley Data repository
Stepper_mount_Y	STL	CC BY NC 3.0	Mendeley Data repository
Stepper_mount_I	STL	CC BY NC 3.0	Mendeley Data repository
Z_Gearbox_A	STL	CC BY NC 3.0	Mendeley Data repository
Z_Gearbox_B	STL	CC BY NC 3.0	Mendeley Data repository
Idler_mount	STL	CC BY NC 3.0	Mendeley Data repository
GT2_Belt_clip	STL	CC BY NC 3.0	Mendeley Data repository
Syringe_mount_100μL	STL	CC BY NC 3.0	Mendeley Data repository
Plunger_mount_100μL	STL	CC BY NC 3.0	Mendeley Data repository
Syringe_bracket_100μL	STL	CC BY NC 3.0	Mendeley Data repository
Syringe_mount_250μL	STL	CC BY NC 3.0	Mendeley Data repository
Plunger_mount_250μL	STL	CC BY NC 3.0	Mendeley Data repository
Syringe_bracket_250μL	STL	CC BY NC 3.0	Mendeley Data repository
Syringe_mount_1000μL	STL	CC BY NC 3.0	Mendeley Data repository
Plunger_mount_1000μL	STL	CC BY NC 3.0	Mendeley Data repository
Syringe_bracket_1000μL	STL	CC BY NC 3.0	Mendeley Data repository
Syringe_mount_2500μL	STL	CC BY NC 3.0	Mendeley Data repository
Plunger_mount_2500μL	STL	CC BY NC 3.0	Mendeley Data repository
Syringe_bracket_2500μL	STL	CC BY NC 3.0	Mendeley Data repository
30Vial_rack	STL	CC BY NC 3.0	Mendeley Data repository
96Well_rack	STL	CC BY NC 3.0	Mendeley Data repository
30Vial_top	STL	CC BY NC 3.0	Mendeley Data repository
Rack_mount_A	STL	CC BY NC 3.0	Mendeley Data repository
Rack_mount_B	STL	CC BY NC 3.0	Mendeley Data repository
Rack_mount_5mm_acrylic_A	STL	CC BY NC 3.0	Mendeley Data repository
Rack_mount_5mm_acrylic_B	STL	CC BY NC 3.0	Mendeley Data repository
Endstop_mount	STL	CC BY NC 3.0	Mendeley Data repository
M5_Spacer_6	STL	CC BY NC 3.0	Mendeley Data repository
M5_Spacer_7	STL	CC BY NC 3.0	Mendeley Data repository
M4_Spacer_12	STL	CC BY NC 3.0	Mendeley Data repository
M5_Spacer_14	STL	CC BY NC 3.0	Mendeley Data repository
Liquid_handling.exe	executable	CC BY NC 3.0	Mendeley Data repository
Volume_calibration.exe	executable	CC BY NC 3.0	Mendeley Data repository
Liquid_handling.py	Python script	CC BY NC 3.0	Mendeley Data repository

(continued on next page)

(*continued*)

Design file name	File type	Open source license	Location of the file
Volume_calibration.py	Python script	CC BY NC 3.0	Mendeley Data repository
config.ini	Configuration file	CC BY NC 3.0	Mendeley Data repository
MULA_Marlin firmware	Software package	CC BY NC 3.0	Mendeley Data repository

4. Bill of materials summary

The complete bill of materials can be found in the supplementary information. The total cost of all parts is 676.83 € in Germany, including the *Hamilton* gastight syringe and needle, as well as 1 kg of PETG filament. The cost of the tools and a 3D printer, as well as a PC for control, is not included.

5. Assembly instructions

Different angles of the CAD model are depicted in Fig. 2. To make the assembly of our system as straightforward as possible, we have eliminated the need for special tools and techniques like soldering. Since almost all necessary parts are 3D-printed, basically all

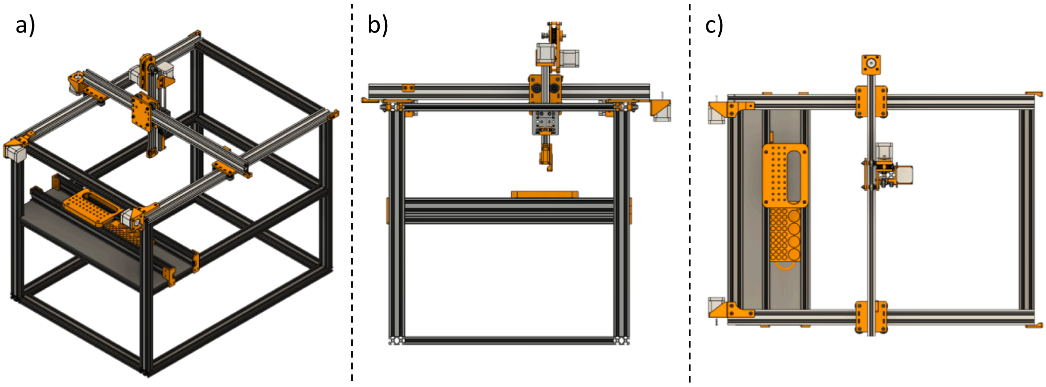


Fig. 2. CAD model of MULA in different orientations: a) Diagonal view b) side-view c) top-view.

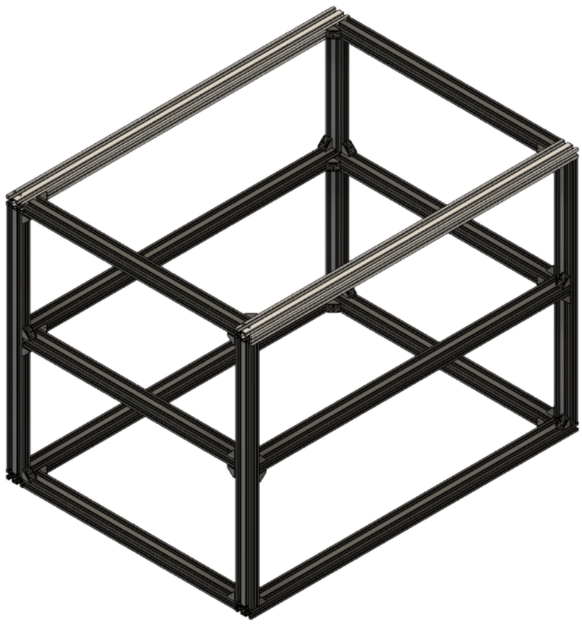


Fig. 3. Assembled structural frame of our implementation of MULA: A double cuboid made from 2040 Nut 6 aluminium profiles with dimensions of 800 x 600 x 600 mm. 2040 V-slot profiles are illustrated in white.

that is needed to build MULA is a set of hex keys, some pliers, and a wrench. If not available in the right length, the aluminum profiles and the hardened steel rod can be cut to the correct length in your mechanic's department or local hardware store.

Timing Belts: To cut the correct length of the timing belt, we recommend using the formula $L[mm] = 2(n + 150)$, where n is the length of the profile in mm. This accommodates the space needed for the idler- and stepper pulleys. It might be easier to connect the timing belt to the gantry plate with the provided clips or cable ties before assembly; however, connecting the belt to the already assembled and mounted gantry plate is also possible (and necessary later for the z-axis).

Hammer nuts: To ensure proper mounting to the frame, make sure the hammer nuts are aligned correctly. The flat side of the nut should be in contact with the aluminum profiles.

Sampling area: If MULA needs to be designed with respect to the effective sampling area X_s x Y_S x Z_S, the following formulae can be used to estimate the actual dimensions X x Y x Z of the frame, from which the actual lengths of the aluminum profiles are calculated as described in the next chapter:

$$X = X_s + 240\ mm | Y = Y_s + 220\ mm | Z = Z_s + 100\ mm$$

Assembly manual: Since we wanted to make the assembly as easy as possible, we have created a detailed illustrated assembly manual, which we provide with the article in the supplementary information.

Aluminium profiles and frame assembly (assembly manual pages 1–3).

Most of the frame is assembled from standard 2040 aluminum extrusions as depicted in Fig. 3. However, the profiles where the two X-axis gantries are mounted should have a 2040 V-slot type. The profiles of the main frame are connected to each other with 90° corner brackets (in our case, made from metal but could also be 3D-printed), M4x10 screws and M4 hammer nuts. The main build has a frame

Table 4
Formulas to calculate the correct lengths and parts for the frame.

Category	Our Build (800 x 600 x 600 mm)	Parametric double cuboid (**X** x **Y** x **Z**)	Parametric cuboid (**X** x **Y** x **Z**)
Motion	2 x 800 mm 2040 V-Slot	2 x **X** mm 2040 V-Slot	2 x **X** mm 2040 V-Slot
	1 x 700 mm 2040 V-slot	1 x (**Y**+100 mm) 2040 V-Slot	1 x (**Y**+100 mm) 2040 V-Slot
	1 x 200 mm 2040 V-Slot	1 x (**Z**/2 – 100 mm) 2040 V-Slot	1 x (**Z** – 100 mm) 2040 V-Slot
Frame	2 x 800 mm 2040	2 x **X** mm 2040	2 x **X** mm 2040
	2 x 760 mm 2040	2 x (**X** − 40 mm) 2040	4 x (**Y** − 80 mm) 2040
	6 x 520 mm 2040	6 x (**Y** − 80 mm) 2040	4 x (**Z** − 40 mm) 2040
	4 x 560 mm 2040	4 x (**Z** − 40 mm) 2040	24 x corner brackets
	44 x corner brackets	44 x corner brackets	48 x M4 x 10 screw
	88 x M4 x 10 screw	88 x M4 x 10 screw	48 M4 hammer nut
	88 M4 hammer nut	88 M4 hammer nut	
Rack	2 x 600 mm 2040	2 x **Y** mm 2040	2 x **Y** mm 2040

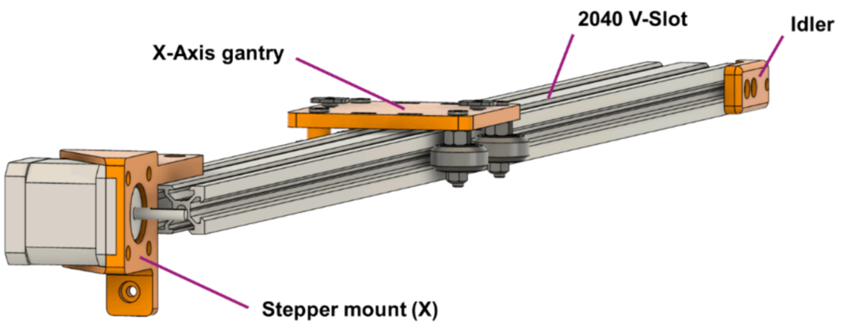


Fig. 4. CAD model of the assembled X-axis. Note that several parts were omitted, including screws, pulleys and belts.

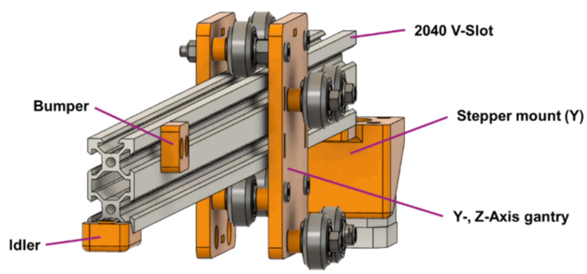


Fig. 5. CAD model of the assembled Y-axis. Note that several parts were omitted, including screws, pulleys and belts.

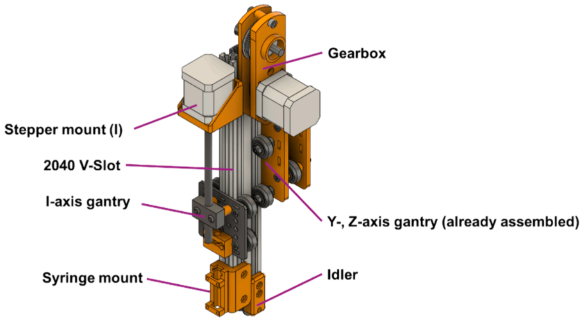


Fig. 6. CAD model of the assembled I-, Z-axis. Note that several parts were omitted, including screws, pulleys and belts.

size of 800 x 600 x 600 mm. However, if adjustment of the dimensions is planned to build MULA with the dimensions X x Y x Z (note that this is the size of the frame, not the sampling area), it is necessary to calculate the length of the 2040 profiles according to Table 4. V-Slot profiles are used to support the moving parts. We recommend cutting the Y-axis profile to Y+100 mm and the Z-axis profile to Z/2 – 100 mm for the double cuboid and Z − 100 mm for the cuboid (if the Z-axis is too long here, the syringe might drop to the bottom every time the motors are disabled).

For example, our implementation was designed to have a sample area of 380 mm in the Y-direction, leading to a total frame size of 600 (380 + 220) mm. The 2040 Y-axis V-slot profile then has a total length of 700 (600 + 100) mm.

X-Axis assembly (assembly manual pages 6–13, Fig. 4***)***.

To assemble the X-axis, the first step is to assemble and mount the two X-axis gantries to the frame and continue by mounting the stepper motors and idler pulleys. Then, feed the belt over the pulleys and secure it to the gantries using zip-ties or 3D-printed clips. Lastly, attach two M4x10 screws and M4 hammer nuts on each gantry, where the y-axis profile will be connected later.

Y-Axis assembly (assembly manual pages 15–23, Fig. 5***)***.

The Y-axis assembly is very similar to the X-axis; one Y-axis plate is assembled as described below. After this, connect the second Y-axis plate and attach the Y-axis gantry to the Y-axis profile. Then, attach the assembled Y-axis stepper mount and idler mount to the Y-axis profile. Proceed with the timing belt and secure it as described in the X-axis assembly. Lastly, attach the assembled Y-axis to the two X-axis gantries.

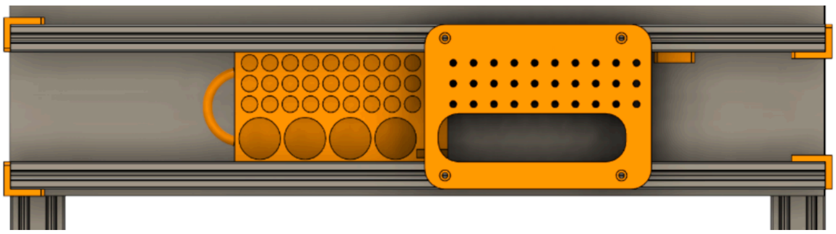


Fig. 7. Moveable rack assembly.

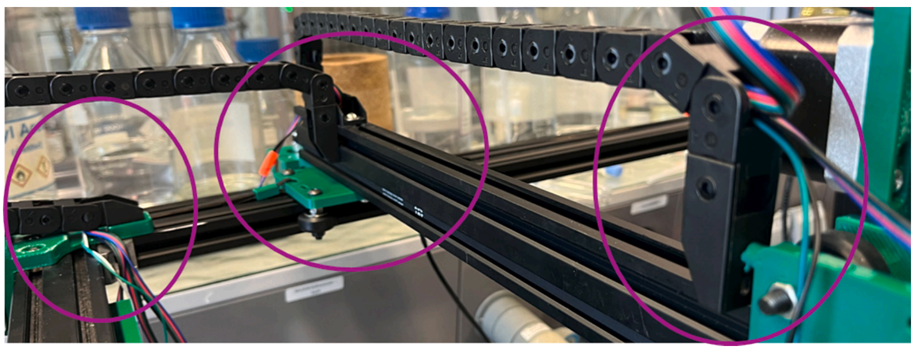


Fig. 8. Drag chain setup of MULA.

I-, Z-Axis assembly (assembly manual pages 24–41, Fig. 6***)***.

The I-, Z-axis assembly begins with the Z-axis stepper gearbox assembly and attachment of the gearbox to the I-, Z-axis profile.

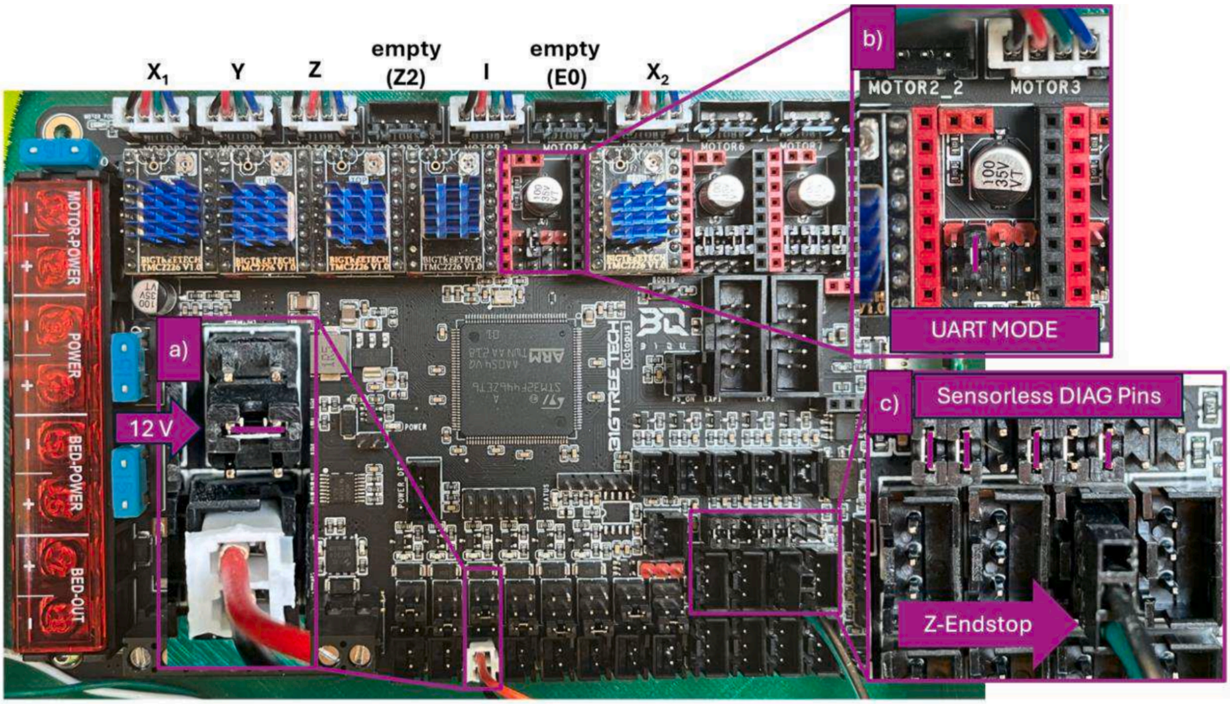


Fig. 9. Octopus control board; with a) showing the fan connector and fan power jumper (12 V). b) Show the driver's settings for UART mode. c) showing the sensorless DIAG Pings and the connected cable for the Z-endstop. The connected stepper motors to the X_1-,Y-, Z-, I- and X_2-axes are shown on the top left (more information at https://3dwork.io/en/btt-octopus/).

Then, attach the assembled I-axis stepper mount and I-axis gantry to the Z-axis profile. Next, mount the assembled idler mount to the lower end of the Z-axis profile and prepare the timing belt. Using a 200 mm long V-slot profile, we use around 700 mm of timing belt according to the aforementioned formula. Note that we had problems in sourcing the Acme nut block for the lead screw due to it only being available as part of an *Openbuilds* C-beam assembly. However, we found a 3D-printable version of the part that can be used as a replacement when printed from PETG.[4] We have created different mounts for *Hamilton* gastight syringes (100, 250, 1000 and 2500 μL). Note that the two larger volume syringe plungers have a thread in the plunger, and the smaller ones do not. We designed two variants of the Plunger_mount to accommodate this. Due to simpler detachment, we recommend using the threaded plunger when possible. Insert and mount the syringe into the Syringe_mount and secure it with the Syringe_bracket parts. Lastly, attach the assembly to the I-, and Z-axis profile and verify that the plunger can be attached to the Plunger_mount using a M3x8 screw.

Rack assembly (assembly manual pages 42–48, Fig. 7***).***

Attach the 30Vial_top part to one profile. Then, slide the 30Vial_rack into the aluminum profile (sometimes, the remains of support structures must be removed first; slide the rack back and forth on the profile until it moves smoothly). To ensure that the moveable rack does not move too far, align the rack with the holes of the Rack_top part and then secure it with a Bumper_30mm part. We have designed two variants of the Rack_mount (A/B) parts; one accommodates a 5 mm acrylic plate beneath the profiles, and the other one does not. Slide the second profile into both the top part and the moveable rack. Add two Rack_mountA/B parts and mount the assembly to the main frame. Make sure that the rack still can be removed. Then, attach the two remaining Rack_mountA/B parts on the other side of the rack to finish the rack assembly.

Cable and board management (assembly manual pages 50, 51, Fig. 8***).***

Two drag chains are used for proper cable management. One is mounted to the head of MULA on the Y-gantry_mount part with an M3x10 screw and on the Y-axis profile, while the other is also mounted to the Y-axis profile and to the Stepper_mount_right with an M4x16 screw.

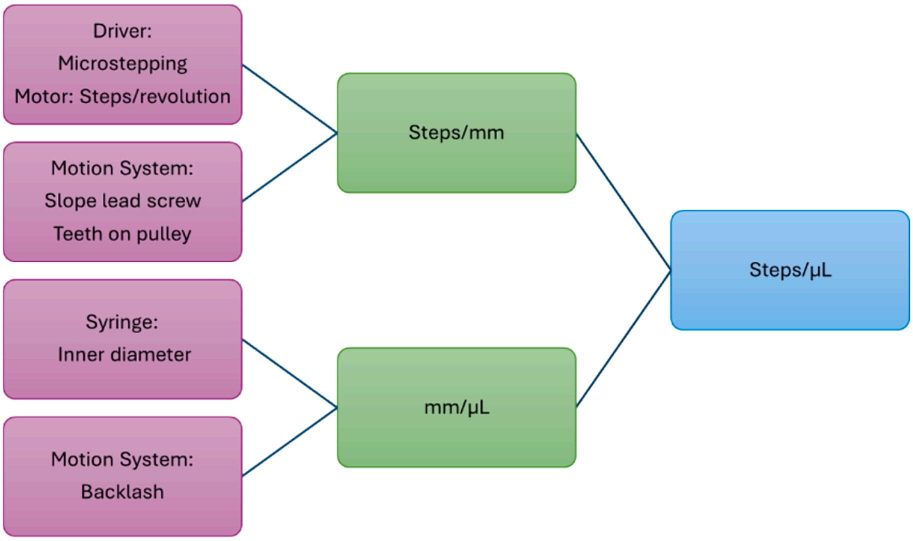
Furthermore, we have incorporated the octopus control board in a 3D-printed case, which is then mounted to the frame to make the setup cleaner and better manage the cables from the board. We used an available case from Thingiverse[5] that suited our needs well.

Electronic assembly and validation (assembly manual pages 55–56, Fig. 9***).***

To ensure a safe and reliable operation, the assembly of the electronic part of MULA must be conducted carefully and with caution. Five or more *Trinamic* 2226 (or similar) drivers must be installed on the controller board after ensuring that the configuration jumpers

[4] For more information visit https://www.thingiverse.com/thing:2607994 (2024, June 30).

[5] For more information https://www.thingiverse.com/thing:5463756 (2024, June 30).



Scheme 3. Overview of the correlation of steps, mm and µL.

under each driver are set for UART mode, as depicted in Fig. 9b). Note that there is a free driver slot between the I- and X_2-axis drivers; this is where the driver for the extruder can be mounted. Then, the plugs from the stepper motors are connected to the board, as depicted at the top of Fig. 9. Since we are currently not using a double Z-axis, the 4th motor connector from the left side must be left empty. Furthermore, the power connectors and an emergency button are connected to the board. Then, the stallguard diagnostic jumpers and Z-endstop connector must be configured according to Fig. 9c. Lastly, the fan connector and fan power jumper must be configured according to Fig. 9a. After connecting all the electronics, the following steps are recommended to eliminate the possibility of false connections or other issues with the setup:

Firmware Flashing and Check: Prepare the firmware as previously described and flash it to the board by inserting the micro-SD card and powering it. Further information can be found in the control board manual (https://github.com/bigtreetech/BIGTREETECH-OCTOPUS-V1.0;page 20/21). Then, check if the flash was successful by removing the micro-SD card from the board and reinserting it into your PC; if the flash was successful, there should now be a file named FIRMWARE.CUR. It is unnecessary to place the micro-SD card back into the octopus board unless you want to flash a new firmware (in that case, just copy the new FIRMWARE.bin file to it). You can now connect the board to the PC using the provided USB-C cable. In the device manager, you should find the assigned COM port of the board, which must be selected in Pronterface in the next step.

Endstop Check: send the M119 command *via* Pronterface[6]; with our configuration, you should see the parameter z_max change from open to TRIGGERED when pressing the Z-axis end-stop switch and resending the M119 command.

Driver/Stepper Check: send the M122 command *via* Pronterface[7]; this should display some information for all configured stepper motors; when the driver and motor of one axis are installed correctly, all axes should be listed here.

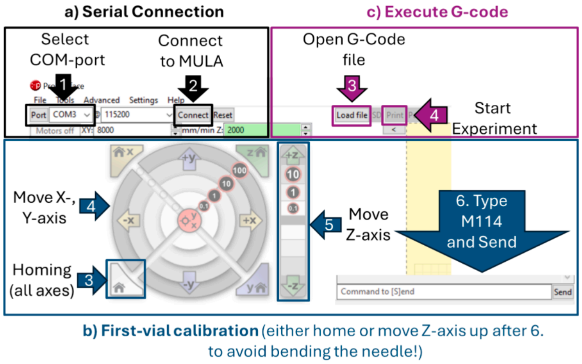
Homing Check: The machine is ready to conduct a first homing sequence using Pronterface when all previous checks give good results. Make sure that nothing is obstructing the movement of the machine! Then, pressing the homing symbol will home all axes (Note: The Z-axis should always home first!). When the machine is not moving as expected, you can always press the emergency power cutoff and either adapt the firmware (invert homing directions) or check the wiring of the stepper motors before trying again.

Volume correlation

For the movement of the syringe plunger, the unit mm is unsuitable since the user wants to specify a volume instead of a length. Therefore, a correlation between the linear motion of the plunger I-axis (in mm) and the actual dispensed volume (in µL) is crucial. Scheme 3 gives an overview of the parameters and other considerations to obtain a correlation between motor steps and dispensed µL. We are introducing this correlation not in the firmware. Instead, the Python script calculates the distance of the plunger according to the volume input and generates the appropriate G-code commands. In the following paragraph, the basic theory behind this approach is demonstrated:

To calculate the volume of the syringe displaced when moving the plunger 1 mm, the inner diameter (d) of the used syringe must be obtained from the manufacturer. Then, with the formula: $\frac{mm}{\mu L} = \frac{1}{\pi \times \left(\frac{d}{2}\right)^2}$ for d = 4.61 mm as in our case, we get a theoretical

[6] More information on https://marlinfw.org/docs/gcode/M119.html (2024, June 30).
[7] More information on https://marlinfw.org/docs/gcode/M122.html (2024, June 30).



Scheme 4. Pronterface tutorial: a) When opening Pronterface, the first step is always to establish a serial connection. In our case, COM-Port 3 is assigned to MULA. However, this can differ between setups. b) To conduct a first-vial calibration, start by homing all axes and then manually move the X- and Y-axis to the right position above the vial. Lower the Z-axis to verify and then request the position of the head by sending the M114 command via the terminal. c) To execute a G-code file, simply load the file and click the button labeled "Print" to start the experiment.

Table 5
Explanation of parameters in the config.ini file.

Parameter in the config.ini file	Purpose
min/max_volume	Min/max volume that should be handled with the installed syringe
theoretical_factor, backlash_correction	See the previous paragraph about volume calibration
vial1_x/y, solvent1_x/y, waste_x/y	Position of 1st sample vial, 1st solvent vial and waste
Number_of_solvents dx_s/dy_s, increment_y	Number of solvent positions in the used rackThe relative distance of vials and solvents in the rack, used by the script to calculate the positions of all other vials and solvents
vials_per_row, columns	Number of vials in the rack in Y (vials_per_row) and X (columns) direction
Z_min, Z_max Z_slow	Absolute minimum and maximum coordinates of the Z-axis Coordinates of the Z-axis for layering (not in contact with liquid)
Fz, Fxy	Speed for Z- and XY-movements
Fa_push, Fa_pull, Fa_slow	Plunger feed rate (speed) during pull- and push movements

correlation factor of around $0.06 \frac{mm}{\mu L}$, meaning that the I-axis has to move 0.06 mm (or $0.06\ mm \times 400 \frac{steps}{mm} = 24\ steps$) [see chapter 3.2.2] to dispense 1 μL. Therefore, with a 1000 μL syringe, we can achieve a theoretical volume resolution of $\frac{1\mu L}{24 steps} \approx 0.0417\ \mu L$. If that is not enough, one could always further increase the micro-stepping of the I-axis (or better change to a smaller syringe). Keep in mind that a micro syringe will quickly lose accuracy with volumes smaller than 10 % of the maximum volume. However, during testing with only the theoretical correlation factor, we found that there was an absolute systematic error of around 25 μL for all tested volumes. A possible explanation for this observation might be the backlash of the plunger axis. For a detailed explanation of backlash in the field of liquid handling, please consider this article. [27] To tackle this error, a backlash correction is conducted with $absolute\ systematic\ error\ [\mu L] \times theoretical\_factor \frac{mm}{\mu L} = backlash\_correction\ [mm]$. This procedure is detailed in the paragraph for volume calibration.

6. Operation instructions

6.1. Software

In Section 3.2, the requisite configuration of the control board has been delineated, encompassing driver installation and firmware upload. Once the aforementioned procedures have been completed, it is possible to utilize any software that facilitates serial communication in order to control MULA. Pronterface is our software of choice for accepting and executing G-code files. The software is used in two different ways. The first step is establishing a serial connection with the marlin control board. This is achieved by selecting the correct COM-port (when connecting MULA with a USB cable, the assigned COM-port appears in the port selection in the Pronterface) and clicking the *Connect* button (Scheme 4a). Then, the user can either load the G-code file and start an experiment by clicking the *Print* button or manually home the machine and conduct the first-vial calibration by moving the axes and sending the M114 command when at the correct positions to request the coordinates. The process of executing G-code files or conducting first-vial calibration is depicted in Scheme 4b) and c).

To generate a custom G-code file with the provided executables, the following things must be considered:

The config.ini file: This file stores many relevant parameters for vials, machine settings, and syringe size and is read by the executable every time it is launched. Ensure that the executable and the configuration file are in the same folder and that the correct configuration file is loaded. Due to changes in experiments, regular modification of this file is inevitable. In Table 5, the relevant

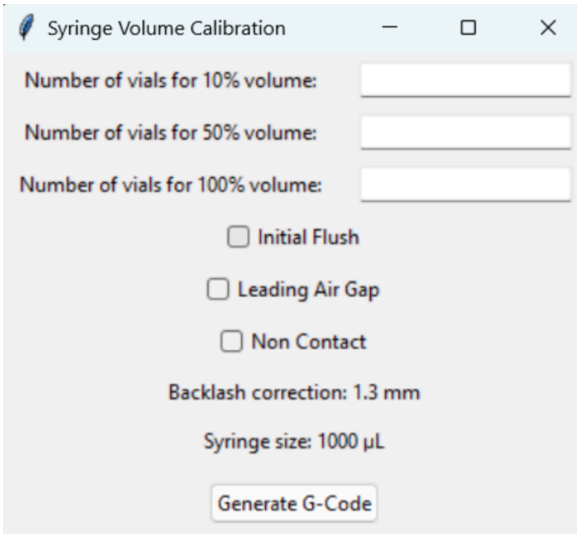


Fig. 10. The Syringe Volume calibration.exe file. The software, as depicted, gives the possibility, depending on which syringe is installed, to take 10%, 50%, or 100% of the maximum volume. In addition, the user can choose between different pulling modes to calibrate the syringe. Furthermore, the backlash correction can be calibrated as well by changing the config file.

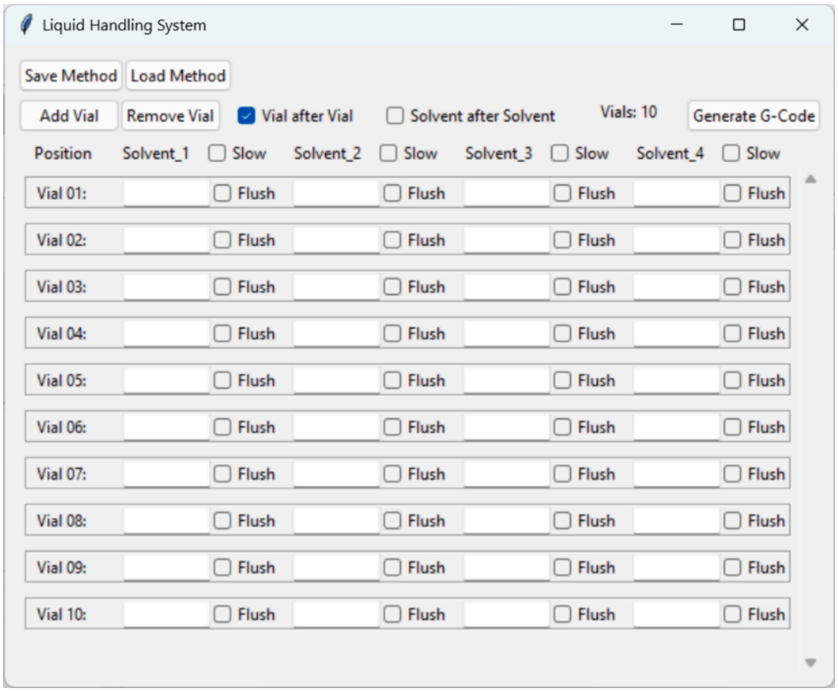


Fig. 11. GUI of the program for general liquid handling.

parameters in the config file are briefly explained.

The Volume calibration.exe file: For general calibration of syringes, we have created a GUI to generate G-code instructions for MULA (Fig. 10). The user can pull 10 %, 50 %, and 100 % of the maximum volume of the installed syringe as defined in the config file. By default, the user can fill a maximum of 30 vials in total with the installed rack. Furthermore, the user can choose between different modes, such as initial flush, leading air gap, and non-contact dispensing, while calibrating the syringe. It is also possible to use all the different pull methods at once. In our testing, using the Initial flush setting gives higher reproducibility; therefore, for volume calibration, we always use the initial flush mode at the beginning. MULA will then flush the installed syringe by default three times with the chosen fluid. In addition, the user can adjust the backlash correction in the config file and fine-tune the calibration of the syringe.

The Liquid handling.exe file: For general liquid handling tasks, we have created an intuitive GUI to generate G-code instructions

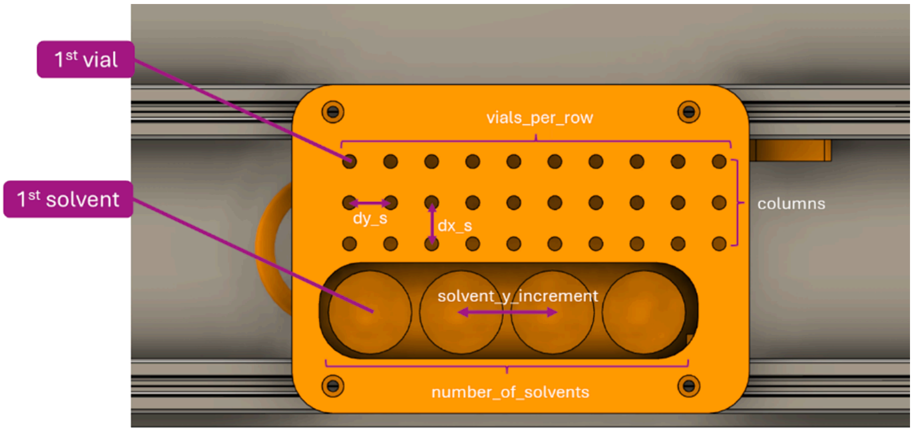


Fig. 12. Visual explanation of several parameters from the Rack section in the configuration file. For the depicted 30Vial rack, the following values are given: vials_per_row = 10; columns = 3; dx_s, dy_s = 15 [mm], solvent_y_increment = 35 [mm], number_of_solvents = 4 (when the waste is located somewhere else; otherwise set to 3 and calibrate the position of the most right solvent as waste coordinates.

for MULA (Fig. 11). Users can save or load a method with the respective buttons, as in commercial implementations. By default, there are 10 vials in the GUI, and the number of vials can be adapted by interacting with the two buttons to add or remove vials. Then, there are two modes of how MULA can handle the task: Vial after Vial and Solvent after Solvent. In the first mode, MULA fills one vial after another with all selected solvents, whereas in the second mode, it is ensured that all vials are filled with one solvent type before changing to the next solvent. This largely eliminates the need for flushing operations. Checkboxes for flushing operations are on the right side of each volume input window (white square; input in μL). If those are selected, MULA will flush once before executing the respective pipetting task. Lastly, by ticking the slow checkbox for a solvent type, all dispensing operations for that solvent will happen at a reduced speed and with contact-free dispensing (needle not in contact with liquid). This is especially useful when layering miscible solvents for crystal growth experiments.

6.2. Hardware

Safety considerations: The described machine deals with micro syringes with sharp needles that can be hazardous. The risk is even higher if the micro syringe samples are corrosive or poisonous liquids or gases. As the user must assemble the syringe driver in the presented setup, it is crucial to exercise care and attention to ensure proper operation and avoid exposing people to unnecessary risks. Furthermore, injuries can happen from the moving parts. Immediately press the emergency cutoff in the case of an emergency.

Positioning of the machine: The machine should be located on a flat surface, and ideally, dampeners should be mounted to the bottom of the frame to reduce vibrations. Verify that nothing obstructs the movement of the axes by carefully moving the head in all corners.

Dry run: Before attaching the needle to the syringe and connecting the plunger to the I-gantry, we suggest trying a dry run to validate all axes' correct function and homing and ensure no cable is stuck.

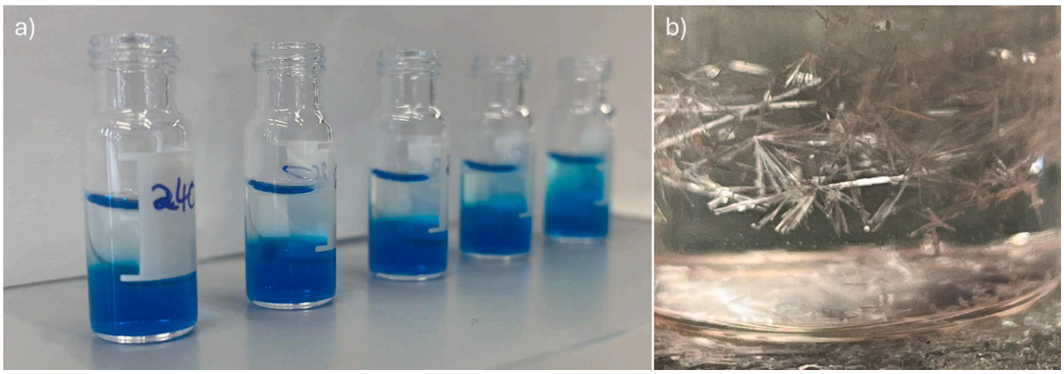


Fig. 13. Solvent layering experiments with MULA: a) optimization of the best dispensing velocity. Interestingly, going very slow did not yield the best results because the liquid then simply dropped down from the needle, disturbing the lower layer. We have found that a feed rate of 240 works best with a type 5 needle tip to ensure the liquid is dispensed via the glass vial wall. b) Single crystals of D-glucose after several layering attempts of a saturated aqueous D-glucose solution with acetone.

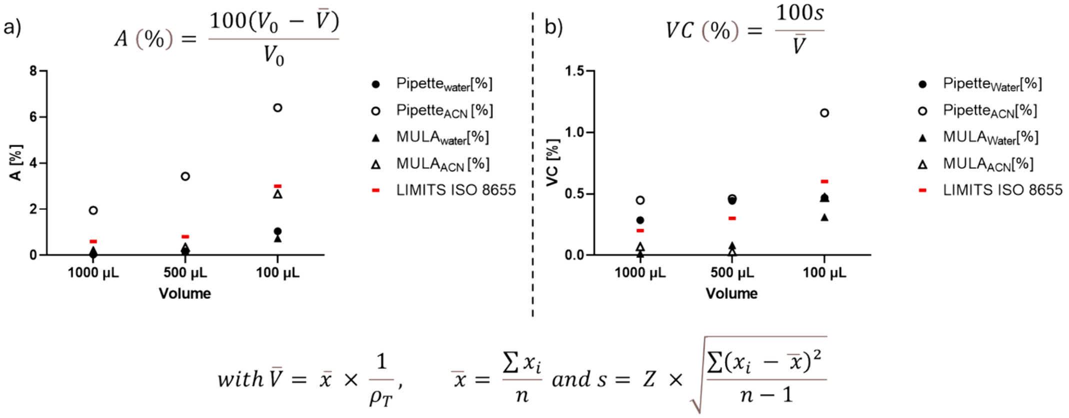


Fig. 14. Validation results of MULA in comparison to an air displacement pipette with water and acetonitrile (ACN) as solvents. a) Both MULA and the tested pipette show comparable accuracies within the ISO 8655 limits when water is used as a solvent. However, when an organic solvent like ACN is of interest, MULA delivers significantly better accuracy. b) For small volumes (100 µL), MULA delivers slightly better repeatability between measurements. For larger volumes (500, 1000 µL), interestingly, under the tested conditions, the pipette fails to fulfill the ISO 8655 limits, while MULA again yields significantly lower repeatability errors.

First-vial calibration: For a safe and reliable machine operation, it is critical to calibrate the absolute positions of the relevant 1st vial locations (sample, solvent and waste) in the removable rack when the needle is connected to the syringe (Fig. 12). Any misalignment can result in damage to the syringe needle. This can be accomplished after needle attachment by moving the syringe horizontally to the desired position in Pronterface and then slowly lowering the Z-axis (and readjusting the horizontal position) until it is in the middle of the vial opening. Command M114 can then be used to determine the absolute position of the vial. It is advisable to exercise caution during this procedure. Once the position of the first vial has been optimally adjusted, it must be stored in the config.ini file. Repeat this procedure also for the first solvent vial and the position of the waste. Don't forget to save the config.ini file afterwards. Despite the care taken during the initial positioning, it is virtually impossible to completely avoid accidents when dealing with micro syringe sampling because penetrating through septa can cause the needle to bend, necessitating either straightening or replacing with a new needle. Following the replacement of the needle, it is again essential to conduct a first-vial calibration.

Volume calibration: When absolute volume accuracy is critical for your work, volume calibration is recommended for each syringe-needle combination to accommodate for variations of the inner syringe diameter and backlash of the I-axis. This is done gravimetrically by weighing vials before and after the liquid handling task and adjusting the theoretical factor and/or backlash correction in the config.ini file:

1. If not done already, set the volume correlation factor in the config.ini file to the theoretical factor
2. Download the Excel template for volume calibration
3. Measure the weight of the empty test vials before the experiment
4. Conduct at least 3 samples each for 10 %, 50 % and 100 % of the total syringe capacity with the solvent of your choice
5. Measure the weight of the test vials after the liquid dispensing and calculate the mass of only the liquid
6. Calculate the volume of liquid using the correct density and compare with the target value
7. If the measured volume is too low, increase the value for backlash_correction according to the aforementioned formula and *vice versa*.

Syringe and needle configuration: Vials with thicker septa may present a challenge when attempting to puncture them using syringes with relatively thin needles (e.g. RN 26). Additionally, previously bent needles from accidents may prove more difficult to penetrate septa compared to new, straight needles. It is, therefore, necessary to conduct a trial-and-error process to identify the optimal combination of syringe and septum. Other than previously shown [24], *Hamilton* micro syringes with a type 2 needle tip were found to clog with silicone parts from the septum (soaking in toluene usually works to remove silicone remains from the needle) therefore, we investigated type 5 needles, which were highly reliable when used with 1.0 mm 55° shore A silicone/PTFE septa. Note that storing a separate config.ini file for each syringe size is a good practice when switching between several syringes.

7. Validation and characterization

There are several methods for growing single crystals for single-crystal X-ray diffractometry. Besides diffusion and cooling of a saturated solution, layering of two different solvents is a common technique to obtain the desired crystals. However, finding suitable solvents and solvent ratios can be very tiring and tricky. Using MULA, we conducted a high-throughput screening for crystal growth after optimizing the speed of adding the second layer. This is illustrated in Fig. 13a, where an aqueous solution of methylene blue is layered with acetone. We then used the slow checkbox for the liquid handling script with a type 5 needle (hole on the side; the liquid is dispensed slowly via the glass vial wall) to automatically layer different volumes of acetone on top of a saturated glucose solution. This gave us beautiful crystals that we then analyzed to obtain a suitable molecular model for D-glucose, as depicted in Fig. 13b). Note that the execution of this experiment by MULA is also demonstrated in a short, detailed video that can be found in the repository. When screening multiple different solvent mixtures, using an automated liquid handler like MULA thus can save the user several hours of manual pipetting.

Furthermore, we tested the liquid handling performance according to DIN EN ISO 8655. The tests were performed *via* contact dispensing (needle in contact with liquid) using a 1000 µL *Hamilton Gastight* syringe with a removable needle (RN) in size 22 with a type 2 tip. As recommended in the ISO norm, we have conducted 10 measurements for each of the three specified volumes: 100 µL (10 %), 500 µL (50 %) and 1000 µL (100 %). Distilled water and HPLC-grade acetonitrile were used for the testing to further demonstrate the versatility of this system. For comparison, we had a trainee conduct the same experiments with a calibrated 1000 µL air displacement pipette. We then calculated the accuracy (**A**) and coefficient of variation (**VC**) according to the formulas for statistical quality control of air displacement pipettes. Those experiments again highlighted the advantages in accuracy and repeatability of MULA over classical air-displacement pipettes, especially for volatile liquids like acetonitrile, as depicted in Fig. 14.

To conclude, we have found automatic liquid handling using MULA to be superior to manual pipetting with an air-displacement pipette in the experiments that were conducted. Due to its simplified nature and control, MULA has some limitations. The control *via* open-loop-control (no direct feedback from MULA) might result in problems for more elaborate experiments. Currently, only one rack is supported simultaneously during an experiment (although this is rather a software limitation than a hardware limitation and can be resolved in the future), making more complicated experiments not feasible at the moment. Another limitation is that only one syringe can be integrated so far. However, this is currently being investigated since the hardware and electronic configuration can include a second individual Z-, I-Axis. Such a dual-syringe setup is useful for parallelizing experiments or excelling in dilution experiments, where both very large and very small volumes are transferred. Also, as an outlook, the ability to connect three more stepper motors to the control board enables creative users to complement the herein-reported modes with further improvements, such as a stepper-controlled centrifuge or a stepper-controlled sliding mechanism for lifting or locking vials in racks. Finally, our group is currently not only investigating the ability to conduct time-specific experiments (kinetic experiments) for different homogenous catalyzed reactions but also enabling a close-loop-control (direct feedback) setup by using a custom Python script to control MULA directly *via* serial communication.

CRediT authorship contribution statement

Leon F. Richter: Writing – original draft, Visualization, Validation, Software, Project administration, Methodology, Investigation, Data curation, Conceptualization. **Wolfgang R.E. Büchele:** Writing – original draft, Validation, Project administration, Investigation, Data curation, Conceptualization. **Alexander Imhof:** Software, Conceptualization. **Fritz E. Kühn:** Writing – review & editing, Resources.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

The authors thank Michelle Coskovic for her experimental support. Furthermore, Nadja Sommer, Dr. Alexander Pöthig and Jürgen Kudermann are acknowledged for their valuable input.

This research received no specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Appendix A. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.ohx.2024.e00581.

References

[1] M.G.O. Lorenz, Liquid-Handling Robotic Workstations for Functional Genomics, SLAS Technology 9 (2004) 262–267, https://doi.org/10.1016/j.jala.2004.03.010.

[2] K. Potgieter, R. Meijboom, Robotics-assisted high-throughput catalytic investigation of PVP nanoparticles in the oxidation of morin, J. Chem. Technol. Biotechnol. 96 (2021) 2547–2557, https://doi.org/10.1002/jctb.6795.

[3] X. Yang, D. Acevedo, A. Mohammad, N. Pavurala, H. Wu, A.L. Brayton, R.A. Shaw, M.J. Goldman, F. He, S. Li, R.J. Fisher, T.F. O'Connor, C.N. Cruz, Risk Considerations on Developing a Continuous Crystallization System for Carbamazepine, Org. Process Res. Dev. 21 (2017) 1021–1033, https://doi.org/10.1021/acs.oprd.7b00130.

[4] L. Bessemans, V. Jully, C. de Raikem, M. Albanese, N. Moniotte, P. Silversmet, D. Lemoine, Automated Gravimetric Calibration to Optimize the Accuracy and Precision of TECAN Freedom EVO Liquid Handler, J. Lab. Autom. 21 (2016) 693–705, https://doi.org/10.1177/2211068216632349.

[5] M.A.H. Capelle, R. Gurny, T. Arvinte, High throughput screening of protein formulation stability: Practical considerations, Eur. J. Pharm. Biopharm. 65 (2007) 131–148, https://doi.org/10.1016/j.ejpb.2006.09.009.
[6] F. He, C.E. Woods, E. Trilisky, K.M. Bower, J.R. Litowski, B.A. Kerwin, G.W. Becker, L.O. Narhi, V.I. Razinkov, Screening of monoclonal antibody formulations based on high-throughput thermostability and viscosity measurements: design of experiment and statistical analysis, J. Pharm. Sci. 100 (2011) 1330–1340, https://doi.org/10.1002/jps.22384.
[7] C.G. Begley, J.P.A. Ioannidis, Reproducibility in Science, Circ. Res. 116 (2015) 116–126, https://doi.org/10.1161/CIRCRESAHA.114.303819.
[8] B. Miles, P.L. Lee, Achieving Reproducibility and Closed-Loop Automation in Biological Experimentation with an IoT-Enabled Lab of the Future, SLAS TECHNOLOGY: Translating Life Sciences Innovation 23 (2018) 432–439, https://doi.org/10.1177/2472630318784506.
[9] L.P. Freedman, I.M. Cockburn, T.S. Simcoe, The Economics of Reproducibility in Preclinical Research, PLoS Biol. 13 (2015) e1002165.
[10] M. Christensen, L.P.E. Yunker, P. Shiri, T. Zepel, P.L. Prieto, S. Grunert, F. Bork, J.E. Hein, Automation isn't automatic, Chem. Sci. 12 (2021) 15473–15490, https://doi.org/10.1039/D1SC04588A.
[11] M. E. DiLorenzo, C. F. Timoney, R. A. Felder, Technological advancements in liquid handling robotics, JALA: Journal of the Association for Laboratory Automation (2001), 6, 36-40. https://doi.org/10.1016/S1535-5535-04-00123-6.
[12] Q. Wei, B. Shi, F. Wang, S. Shao, L. Zhu, X. Zhao, Simple and Rapid Preparation of MIL-121 with Small Particles for Lithium Adsorption from Brine, Coatings 11 (2021) 854. https://www.mdpi.com/2079-6412/11/7/854.
[13] R. Arnott, The RepRap Project—Open Source meets 3D printing, (2008). https://hdl.handle.net/10523/1531.
[14] M. O'Brien, L. Konings, M. Martin, J. Heap, Harnessing open-source technology for low-cost automation in synthesis: Flow chemical deprotection of silyl ethers using a homemade autosampling system, Tetrahedron Lett. 58 (2017) 2409–2413, https://doi.org/10.1016/j.tetlet.2017.05.008.
[15] D. A. V. Medina, A. Lozada-Blanco, J. P. G. Rodríguez, F. M. Lanças, Á. J. Santos-Neto, An open-source smart fraction collector for isocratic preparative liquid chromatography, HardwareX (2023), 15. https://doi.org/10.1016/j.ohx.2023.e00462.
[16] A. Faiña, B. Nejati, K. Stoy, EvoBot: An Open-Source, Modular, Liquid Handling Robot for Scientific Experiments, Appl. Sci. 10 (2020) 814, https://doi.org/10.3390/app10030814.
[17] M. Politi, F. Baum, K. Vaddi, E. Antonio, J. Vasquez, B.P. Bishop, N. Peek, V.C. Holmberg, L.D. Pozzo, A high-throughput workflow for the synthesis of CdSe nanocrystals using a sonochemical materials acceleration platform, Digital Discovery 2 (2023) 1042–1057, https://doi.org/10.1039/D3DD00033H.
[18] J.M.P. Gutierrez, T. Hinkley, J.W. Taylor, K. Yanev, L. Cronin, Evolution of oil droplets in a chemorobotic platform, Nat. Commun. 5 (2014) 5571, https://doi.org/10.1038/ncomms6571.
[19] M.M. Hanczyc, J.M. Parrilla, A. Nicholson, K. Yanev, K. Stoy, Creating and Maintaining Chemical Artificial Life by Robotic Symbiosis, Artif. Life 21 (2015) 47–54, https://doi.org/10.1162/ARTL_a_00151.
[20] C. Zhang, B. Wijnen, J.M. Pearce, Open-source 3-D platform for low-cost scientific instrument ecosystem, J. Lab. Autom. 21 (2016) 517–525, https://doi.org/10.1177/2211068215624406.
[21] A. Faina, F. Nejatimoharrami, K. Stoy, EvoBot: An Open-Source, Modular, Liquid Handling Robot for Scientific Experiments, Appl. Sci. 10 (2020) 814, https://doi.org/10.3390/app10030814.
[22] F. Barthels, U. Barthels, M. Schwickert, T. Schirmeister, FINDUS: An Open-Source 3D Printable Liquid-Handling Workstation for Laboratory Automation in Life Sciences, SLAS Technology 25 (2020) 190–199, https://doi.org/10.1177/2472630319877374.
[23] K. K. C. D. H. Wells, N. Kharma, B. B. Jaunky, K. Nie, G. Aguiar-Tawil, D. Berry, BioCloneBot: A versatile, low-cost, and open-source automated liquid handler, HardwareX (2024), 18. https://doi.org/10.1016/j.ohx.2024.e00516.
[24] M.C. Carvalho, R.H. Murray, Osmar, the open-source microsyringe autosampler, HardwareX 3 (2018) 10–38, https://doi.org/10.1016/j.ohx.2018.01.001.
[25] G. Gome, J. Waksberg, A. Grishko, I. Y. Wald, O. Zuckerman, in Proceedings of the Thirteenth International Conference on Tangible, Embedded, and Embodied Interaction, Association for Computing Machinery: Tempe, Arizona, USA, 2019, pp. 55–64. https://doi.org/10.1145/3294109.3295619.
[26] D.C. Florian, M. Odziomek, C.L. Ock, H. Chen, S.A. Guelcher, Principles of computer-controlled linear motion applied to an open-source affordable liquid handler for automated micropipetting, Sci. Rep. 10 (2020) 13663, https://doi.org/10.1038/s41598-020-70465-5.
[27] A. Gervasi, P. Cardol, P.E. Meyer, Open-hardware wireless controller and 3D-printed pumps for efficient liquid manipulation, HardwareX (2021) 9, https://doi.org/10.1016/j.ohx.2021.e00199.