Mostly 3D printed chemical synthesis robot

Dhanalakshmi Vadivel a,b,*, Diego Savio Branciforti a, Othman Kerroumi a, Marco Dondi a, Daniele Dondi a,b

aDepartment of Chemistry, University of Pavia, Viale Taramelli 12, 27100 Pavia, Italy
bINFN, Sezione di Pavia, Via Agostino Bassi, 6 - 27100 Pavia, Italy

article info

Article history:
Received 3 January 2022
Received in revised form 14 April 2022
Accepted 17 April 2022

Keywords:
Open-source hardware
3D printing
Robot
Python
Chemistry
Raspberry Pi 3
Hotplate interface
pH
Electrode interface

abstract

Thanks to the current technology derived from the open-source world of 3D printers, it is conceivable to automate some laboratory activities remotely. In fact, simple operations, such as mixing liquids or solutions, stirring, heating and sampling to control the reaction course can be easily implemented. The idea of automating the chemical laboratory would have immediate advantages, for example in terms of safety. The operators will be able to remotely control the machines and in case of handling dangerous material or accidents, there would only be damage to the hardware components. Many of the process parameters can also be read with low-cost probes and devices that can be easily interfaced with microprocessors. We include for example, but not limited to, temperature, pH, redox potential, electrochemical measurements in general or the use of probes for specific analytes. In this work we wish to present our liquid sampling station able to control up to 6 reagents and a temperature controlled chemical reactor. The workstation can be used graphically with an intuitive interface written in Python. The control program is structured to have modularity and contains a built-in programming language to control the interfaces.

© 2022 Published by Elsevier Ltd. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

Specifications table

| Hardware name   | Syringebot |
|-----------------|------------|
| Subject area    |            |
|                 | • Chemistry and biochemistry |
|                 | • Medical (e.g., pharmaceutical science) |
|                 | • Biological sciences (e.g., microbiology and biochemistry) |
|                 | • Educational tools and open-source alternatives to existing infrastructure |
|                 | • Measuring physical properties and in-lab sensors |
|                 | • Biological sample handling and preparation |
|                 | • Field measurements and sensors |

(continued on next page)
Hardware name | Syringebot
---|---
- Synthetic chemistry
- Manipulation of hazardous (toxic, radioactive) materials

Open source license | Creative Commons (CC) BY 4.0
Cost of hardware | 1000-1200 euros in the most extended version (6 syringes). See the complete breakdown of costs in a later section (Bill of materials).
Source file repository | [https://osf.io/quz7d/files/osfstorage](https://osf.io/quz7d/files/osfstorage)

1. Hardware in context

Despite the high interest in lab automation, the solutions available on the market concern, for example, applications in the field of microbiology or in the field of radiopharmaceutical synthesis. Clearly, these technologies involve machines that have very high costs and only a few research centres can consider purchasing them. Furthermore, this technology is based on closed-source machines, where the intervention of a specialized operator is required to modify the final applications and to have a customization. The introduction instead of an open-source perspective as regards not only the software but also considering the possible realization of 3D printed parts, would greatly reduce the costs necessary for the realization, development, testing and customization. This perspective is feasible and easily achievable if the technology developed for the realization of 3D printers is used. This open revolution has allowed the reduction of the costs of the mechanical and electronic components present in the printers. Software, firmware and other computer programs related to the world of 3D printing have also seen such development that they are competitive with commercial products.

In fact, in the last few years, some open hardware appeared in the scientific literature. Application covers basic laboratory procedures like liquid manipulation [1], microextraction [2], auto sampling [3,4] and weighing [5] just to cite some. Also, more complex frameworks including modular workstations are present [6–8]. High-level research used robots to perform artificial evolutions and modelling protocols in vitro [9,10]. Some research groups are going to prepare for the next leap, where AI can control a synthesis robot [11].

More surprisingly, even if an organic chemistry synthesis automated lab is desirable [12,13] a low cost, open-source machine able to perform organic synthesis (like controlling heaters, stirrers and other basic instrumentation) is still missing. Under this perspective, we decided to assemble a liquid handling robot able to control a heater-stirrer and expanded it with modules for pH and redox electrodes. The unit is controlled by a Raspberry Pi 3 module and a Python program that renders the interfacing easier.

1.1. Hardware description

The hardware dedicated to control syringes will be referred to as Syringebot. This module handles the movement of disposable syringes for the dosage of reagents and the control of three-way valves to direct the flows of the reactants and solvents. The three-way valves were chosen from those available in the medical field and therefore replaceable at a minimum expense. Syringes are driven by stepper motors, while valves are moved by using RC servos available on the market (radio-controlled toy models). See Fig. 1 for a schematic of the robot.

These modules are controlled by a microcomputer (Raspberry Pi 3) and it is internally referred to as CPU. A control software in Python was created on the minicomputer to manage the syringe and the robot units. The Raspberry Pi 3 with a Linux distribution eases remote management (see Figs. 1 and 2).

The Syringebot unit can control up to six syringes and it is implemented by using a motherboard for 3D printers able to control 3 extruders (Megatronics v3.1). The endstop switches are exploited by using a board having 2 endstops (min and max) for each axis. Additional 2 pins are used for each syringe to control the two three-way valves for directing the liquid flow. The motherboard also has 2 inputs for a direct connection of thermocouples. The firmware used for the Arduino board is a modified version of Marlin. Obviously it is possible to adapt the software to different boards. We suggest reading the Marlin manual and/or its forum if interested in the modification of the firmware.

Open source parametric and procedural CAD (OpenSCAD) was used for the design of the components to be printed. This ensures quick readjustment to different hardware components (e.g. syringes with different volumes or different valves/RC servos) as the main design parameters are indicated with variables (see variables names in Fig. 4a). OpenSCAD can be downloaded for free and it is available for different operating systems [https://openscad.org/]. Evidently, skilled users can adapt our design to different components available on the market. In the current setup we designed a hotplate interface, a pH/electrode interface, and a load cell interface (Fig. 1).
The Syringebot is controlled via USB with a Raspberry Pi 3 and a control program was written in Python. Syringebot (Fig. 2) is composed by:

- A chassis (or frame)
- One or more syringe units
- A valve block for each syringe having 2 three-way valves controlled by servo motors
- Motherboard
- RC servos board
- Hotplate interface

CPU is composed by:

- Raspberry Pi 3
- Power supply (12 V for stepper motors and 5 V for RC servos)

Additional modules described here:

- pH/electrode interface
- Load cell interface (balance)

All the components of the robot (syringe unit, valve unit, hotplate interface, and pH interface and control software) will be discussed in the paper in the following sections.
Please note that the modules proposed are to show the possible modularity and showing possible extension to different types of sensors.

Since we are considering a robotic system, it is essential to monitor the system with cameras (e.g. IP cameras) and Wi-Fi controlled smart plugs in order to intervene in the case of safety issues. In our current system we also implemented a SCARA robotic arm (not discussed here) [https://www.thingiverse.com/thing:3207936] used for sampling.

**Fig. 2.** Schematic of the Syringebot in the full configuration with 6 syringes and 12 valves. Connecting tubes are shown only for the first syringe at the left for the sake of clarity.
2. Design files

| Design File Name          | File type  | Open source license | Location of the file |
|---------------------------|------------|---------------------|----------------------|
| Syringe support           | OpenSCAD   | GNU GPL v3.         | https://osf.io/2jxb8/|
| Valve block               | OpenSCAD   | GNU GPL v3.         | https://osf.io/ayb98/|
| Support for valve block   | OpenSCAD   | GNU GPL v3.         | https://osf.io/2y4hn/|
| Basement for CPU and power supply | OpenSCAD   | GNU GPL v3.         | https://osf.io/8hb4g/|
| Box for electrodes interface | OpenSCAD   | GNU GPL v3.         | https://osf.io/2t3pc/|
| Box for load cell interface | OpenSCAD   | GNU GPL v3.         | https://osf.io/4pyvm/|

All the components were designed using the software open source OpenSCAD. All electronic files are available at https://osf.io/jvg7n/?view_only=a91c3084738f4274a4cb7c598efdf7cfc and include STL models.

3. Bill of materials

In this work the metric units are used. All the dimensions of the STL files are in millimetres and standard parts purchased from Italian hardware stores are given using mm units.

The printable parts of the robot were printed by using less than 1 kg of PLA filament 1.75 mm with a commercial cost of 22.00 €/kg (bought from Futura Elettronica). Also cable ties are not listed in the bill of materials but they are needed for fixing cables.

3.1. Chassis

| Code       | Part               | Description                                           | Qty | Unit Cost (€) | Total Cost (€) | Suppliers       |
|------------|--------------------|-------------------------------------------------------|-----|---------------|----------------|-----------------|
| 8856-PROFALL275 | Aluminium Profile | Aluminium profile with square section 27.5 × 27.5 mm – 1 m | 5   | € 12.00       | € 60.00        | Futura Elettronica |
| 8856-ANGOL275 | Angular            | Angular for aluminium profile 27.5 mm                 | 10  | € 3.00        | € 30.00        | Futura Elettronica |
| 8856-DADOM5  | M5 square nut      | Box of 6                                              | 3   | € 1.20        | € 3.60         | Futura Elettronica |
| 8856-TAPP0275| End cap            | End cap for aluminium profile with square section 27.5 × 27.5 mm | 6   | € 1.00        | € 6.00         | Futura Elettronica |

3.2. Syringe unit (parts for each syringe unit)

| Code       | Part               | Description                                           | Qty | Unit Cost (€) | Total Cost (€) | Suppliers       |
|------------|--------------------|-------------------------------------------------------|-----|---------------|----------------|-----------------|
| 7300-STEPMOT3DRAG | Stepper Motor   | Stepper motor bipolar NEMA 17–2.5 A                   | 1   | € 46.00       | € 46.00        | Futura Elettronica |
| 2846-CAVOSTEPMOT03 | Cable for motor | Extension cable to connect motor – 1 m                | 1   | € 1.50        | € 1.50         | Futura Elettronica |
| 8220-BTIWF4  | Cable for motor   | Cable to connect motherboard to extension cable       | 1   | € 1.70        | € 1.70         | Futura Elettronica |
| 8220-STB24BK | Heat shrinkable sheath | Heat shrinkable sheath 2.5 mm diameter               | 1*  | Pack of 50    | € 9.50         | Futura Elettronica |
| 2828-BARRA8  | Rectified round bar | Rectified round bar 8 mm – 1 m                      | 1   | € 8.00        | € 8.00         | Futura Elettronica |

(continued on next page)
### Table 3 (continued)

| Code          | Part Description                          | Qty | Unit Cost (EU€) | Total Cost (EU€) | Suppliers              |
|---------------|-------------------------------------------|-----|-----------------|------------------|------------------------|
| 2846-ENDSTOP3D | Endstop Mechanical endstop with 70 cm cable | 1   | € 3.50          | € 3.50           | Futura Elettronica     |
| 2846-VITET8P2-500 | Threaded rod Galvanized threaded rod 8 mm – 1 m | 1   | € 8.00          | € 8.00           | Futura Elettronica     |
| 8220-LM8UU/SP  | Linear Bearing Linear Bearing 8 mm LM8UU | 2   | € 2.50          | € 5.00           | Futura Elettronica     |
| 8300-HUB5MM8   | Aluminium joint Aluminium joint to connect the threaded rod to the motor | 1   | € 2.50          | € 2.50           | Futura Elettronica     |
| 8856-DADOM5    | M5 square nut - box of 6 | 2   | € 1.20          | € 2.40           | Futura Elettronica     |
| –              | Seeger E 15 (UNI7435) for LM8UU | 4   | € 0.50          | € 2.00           | Local hardware store   |
| 8400-KSC1      | M3 X10 Screw Box of 330 (this set is sufficient to build 3 syringes) | 12  | € 3.50          | € 3.50           | Futura Elettronica     |
| 8400-KSC2      | M4 X10 Screw Box of 330* | 1   | € 0.01          | € 0.01           | Futura Elettronica     |
| 8400-KSC3      | M4 Hex Nut Box of 330* | 1   | € 0.01          | € 0.01           | Futura Elettronica     |
| 8400-KSC4      | M3 Hex Nut Box of 330* | 12  | € 0.01          | € 0.12           | Futura Elettronica     |
| –              | Syringe Luer Lock Syringe volume 60 ml/10 ml | 1   | € 1.00          | € 1.00           | Amazon                 |
| –              | Inlet/outlet tubes Tygon tubing with Luer Lock connector | 2   | € 0.10**        | € 0.20           | Amazon                 |

* (price refers to a pack of 50 pieces, more than needed to build 6 syringes).
** Estimated price from a bag of 50.

### 3.3. Valve block (parts for each syringe unit)

#### Table 4

**Bill of materials for a valve block composed of two three-way valves.**

| Code          | Part Description                          | Qty | Unit Cost (EU€) | Total Cost (EU€) | Suppliers              |
|---------------|-------------------------------------------|-----|-----------------|------------------|------------------------|
| –             | RC servo Ultra Torque Hitec HS-645MG       | 2   | € 25.00         | € 50.00          | Amazon                 |
| 8400-KSC1     | M3 X10 Screw Box of 330 pieces (this set is sufficient to build 2 valve blocks) | 6   | € 3.50          | € 3.50           | Futura Elettronica     |
| 8400-KSC2     | M4 X10 Screw M2 screw Box of 330* | 8   | € 0.01          | € 0.08           | Futura Elettronica     |
| 8400-KSC3     | M3 Hex Nut Box of 330* | 6   | € 0.01          | € 0.13           | Futura Elettronica     |
| 8400-KSC4     | M4 Hex Nut Box of 330* | 8   | € 0.01          | € 0.08           | Futura Elettronica     |
| 7300-SERVOEXT600 | Cable extension Cable extension for RC servo – length 600 mm | 2   | € 2.50          | € 5.00           | Futura Elettronica     |
| –             | Three-way valve Three way valve for medical use (disposable three way stopcock) | 4   | Price is for a box of 50 | € 15.00 | ebay                 |

### 3.4. Motherboard

#### Table 5

**Bill of materials for the motherboard.**

| Code          | Part Description                          | Qty | Unit Cost (EU€) | Total Cost (EU€) | Suppliers              |
|---------------|-------------------------------------------|-----|-----------------|------------------|------------------------|
| RR-MEGATRONICS3PRO | Motherboard Megatronics v 3.1 Driver based on DRV8825 chip to control stepper motor | 1   | € 108.90        | € 108.90         | Robot domestici        |
| 2846-DRIVERBOARD | Driver 6* | | € 7.90 | € 47.40 | Futura Elettronica |
| –             | Panel Acrylic sheet 200 x 300 x 5 mm | 1   | € 5.00          | € 5.00           | Amazon                 |
| 8220-PAC601B018 | USB cable A to B – 2.5 m | 1   | € 2.60          | € 2.60           | Futura Elettronica     |

*Depending on the number of stepper motors (= syringes) installed.*
### 3.5. RC servos board

| Code       | Part          | Description                                | Qty | Unit Cost (EU€) | Total Cost (EU€) | Suppliers               |
|------------|---------------|--------------------------------------------|-----|-----------------|-------------------|-------------------------|
| 6905-ECL12 | PCB           | Stripboard 100 × 80 mm                     | 1   | € 5.90          | € 5.90            | Futura Elettronica      |
| 4420-CT/40 | Strip Connector | 40 PIN 2.54 mm pitch male strip connector  | 1   | € 0.50          | € 0.50            | Futura Elettronica      |
| 2846-CAVOSTPMOT03 | Cable for connecting RC servo board to motherboard | Cable 4 pin – 1 m | 4*   | € 1.50          | € 6.00            | Futura Elettronica      |

*Useful for all 12 RC servos.

### 3.6. Hotplate interface

| Code       | Part                        | Description                                | Qty | Unit Cost (EU€) | Total Cost (EU€) | Suppliers               |
|------------|-----------------------------|--------------------------------------------|-----|-----------------|-------------------|-------------------------|
| –          | DIN plug                    | DIN connector plug, male, 5 pins           | 1   | € 1.00          | € 1.00            | ebay                    |
| 6900-P-970 | PCB                         | Matrix board                              | 1   | € 3.00          | € 3.00            | Futura Elettronica      |
| R1, R2     | Resistor                    | Resistor 10 k Ohm ¼ W                     | 2   | € 0.20          | € 0.40            | ebay                    |
| R3         | Resistor                    | Resistor 1 k Ohm ¼ W                      | 1   | € 0.10          | € 0.10            | ebay                    |
| R4         | Resistor                    | Resistor 200 Ohm ¼ W                      | 1   | € 0.10          | € 0.10            | ebay                    |
| –          | Thermocouple                | Thermocouple type K                       | 1   | € 10.00         | € 10.00           | ebay                    |
| Q1, Q2     | Transistor                  | Transistor type BC237B                    | 2   | € 0.20          | € 0.40            | ebay                    |

### 3.7. CPU

| Code       | Part                          | Description                                        | Qty | Unit Cost (EU€) | Total Cost (EU€) | Suppliers               |
|------------|-------------------------------|----------------------------------------------------|-----|-----------------|-------------------|-------------------------|
| 7310-RPI3BPLUS | Raspberry Pi 3 Type B+     | Raspberry Pi 3 Type B+ with Wi-Fi and Bluetooth   | 1   | € 89.00         | € 89.00           | Futura Elettronica      |
| 8220-PSS6E0520 | Power Supply                | Power Supply 5 V-2A                                | 1   | € 9.90          | € 9.90            | Futura Elettronica      |
| 4125-MW14512 | Power Supply                 | Power Supply 12 V-144 W                           | 1   | € 24.00         | € 24.00           | Futura Elettronica      |

### 3.8. Sensors (pH, ORP, load cell)

| Code       | Part                          | Description                                                                 | Qty | Unit Cost (EU€) | Total Cost (EU€) | Suppliers               |
|------------|-------------------------------|-----------------------------------------------------------------------------|-----|-----------------|-------------------|-------------------------|
| 8300-YB427 | pH sensor module              | Module for the pH probe or other combined electrodes also known as SEN0161 | 2   | € 29.50         | € 59.00           | Futura Elettronica      |
| 7300-ARDUINOUNOREV3 | Arduino Uno | Arduino Uno REV3 with Atmega328                                              | 1   | € 24.50         | € 24.50           | Futura Elettronica      |
| 2846-PHSENSOR | pH probe                  | Combined electrode for measuring pH                                        | 1   | € 13.50         | € 13.50           | Futura Elettronica      |
| B07VLFMK63 | REDOX Electrode              | Combined electrode for measuring the redox potential                         | 1   | € 62.50         | € 62.50           | Amazon                  |
4. Build instructions

The Syringebot is composed of mechanical parts (chassis, syringe supports, valve blocks) and interfaces (megatronics board, RC servos interface, hotplate interface). The CPU basement module is composed of a Raspberry Pi 3, which contains the control software for all the modules, and power supplies for 12 and 5 V DC. In addition, the sensor module is composed of an Arduino Uno and two interfaces for the electrode connections (in our case one pH and one oxidation–reduction potential (ORP) combined electrodes).

Safety concern:

- Though designed to be minimal, assembling the Syringebot requires some drilling and cutting operations involving hand tools.
- Wear appropriate personal protective equipment (safety goggles and thick gloves) while performing cutting/drilling operations.
- The tip of a soldering iron can reach hundreds of degrees Celsius and leave serious burns if in contact with skin.
- The heat coming from the heat gun could cause severe burns.
- The assembly of the power supplies requires the connection of 220 AC plugs. Caution while performing operations. Unplug the AC power supply also when you are working on low tension apparatus.
- Once finished, the use of the robot should be followed by professional chemists. This is essential for the handling of hazardous substances, exothermic chemical reactions, disposal of wastes etc. We suggest you insert the robot inside a fume hood.

A minimal set of tools is required for the assembly of the Syringebot (see Table 10a).

4.1. Syringebot

Syringebot is composed of:

- A chassis (or frame) discussed in Section 4.1.1
- One or more Syringe units discussed in Section 4.1.2
- A valve block for each syringe having 2 three-way valves and RC servos discussed in Section 4.1.3
- Motherboard discussed in Section 4.1.4
- RC servos board discussed in Section 4.1.5
- Hotplate interface discussed in Section 4.1.6

### Table 9b
Bill of materials for the load cell interface.

| Code       | Part                        | Description                                 | Qty | Unit Cost (€) | Total Cost (€) | Suppliers       |
|------------|-----------------------------|---------------------------------------------|-----|---------------|----------------|-----------------|
| 3085-HX711BOARD | Load cell interface         | Hx711 module for interfacing load cells     | 1   | € 5.00        | € 5.00         | Futura Elettronica |
| 7300-ARDUINOUNOREV3 | Arduino Uno                  | Arduino Uno REV3 with Atmega328             | 1   | € 24.50       | € 24.50        | Futura Elettronica |
| 193,139     | Load cell                   | Micro Load Cell (0–100 g) - CZL639HD        | 1   | € 10.60       | € 10.60        | Robot Italy     |

### Table 10a
Minimal tool set required for the assembly of the Syringebot.

| Tool                        | Use                                                       |
|-----------------------------|-----------------------------------------------------------|
| Soldering Iron and Soldering wire | for soldering wires (steppermotors) and electronic components (hotplate interface, RC servo interface) |
| Wire Stripers               | for connecting motor wires                                |
| Drill with drill bits       | to drill plastic panels, eventually enlarge holes in 3D printed parts |
| Heat Gun                    | for heat shrinkable tubes                                 |
| Plier for seeger rings      | for seeger rings (to fix LM8UU linear bearings)           |
| Allen key                   | for screws                                                |
| Safety glasses              | for safety while cutting or drilling                      |
| Vice or Vice Grips          | for cutting rods                                          |
| Hacksaw for metal           | for cutting rods                                          |
| Socket or Wrench            | for 5 mm nuts                                             |
| Screwdriver                 | for screws                                                |

8
4.1.1. Chassis

The hardware components listed in Table 2 are required to assemble the chassis Fig. 3 of the robot.

1. Initially the aluminium profiles must be cut in this way: 3 profiles cut to 75 cm for the horizontal parts and 1 profile cut in half to have two 50 cm parts for the vertical supports. Last one was divided into 2 parts of 35 cm. Obviously, the profiles for the horizontal part can be used with the desired length based on the number of syringe supports provided in the robot. Please consider at least 12 cm for each syringe installed.

2. Insert the square nuts that will be used to fix the syringe supports inside the slots of the aluminium profiles. For each support, 2 square nuts must be provided. As an example, in the configuration shown, 6 square nuts must be inserted in the top horizontal bar (for the upper support of the 60 ml syringes), 6 in the intermediate (for the upper support of the 10 ml syringes) and 12 in the lower bar (for the bottom support of all syringes).

3. The 2 aluminium profiles of 35 cm will be used as “feet” for the chassis.

4. Insert 2 screws in each angular piece by screwing it onto a square nut without tightening it.

5. Insert 2 angular pieces on the aluminium profile, one on each end.

6. Take the 50 cm long aluminium profiles, with the angular pieces inserted as explained in the previous point, fix each of them perpendicularly in the central part of the 35 cm aluminium profiles, forming an inverted “T”.

7. Place the two “T” upside down with a distance such as to be able to fix the 75 cm long profiles, as will be explained in the next point.

8. Starting from the top down, the second bar should be placed at about 10 cm. The third bar should be placed at a distance of 18 cm from the bar above. These distances could vary if different syringes were used.

9. Tighten the fixing screws of the angular pieces, making sure that the profiles are well aligned with each other and perfectly perpendicular.

10. Finally, fix the end caps to the feet and upper part of the chassis.

4.1.2. Syringes support

The robot is equipped with two types of Luer Lock syringes: 60 ml and 10 ml (Table 10b). Syringe supports can be easily adapted by modifying the OpenSCAD file. Variable names in the SCAD files are relative to Fig. 4a.

Table 10b

| Syringe | d_piston | d_plunger | d_syringeholder | d_syringe | h_syringe | groove |
|---------|----------|-----------|-----------------|-----------|-----------|--------|
| 60 ml volume | 31       | 19        | 48              | 32        | 4.1       |        |
| 10 ml volume  | 16.6     | 14        | 32.2            | 16.6      | 4.1       |        |

Fig. 3. Schematic structure of the supporting frame.
In this part the assembling and the installation of a single syringe support is shown. These instructions can be followed for both types of syringes.

A syringe support is composed of the 3D printable parts Table 1 (first row) and Table 10c (for a more detailed list) and mechanical parts (Table 3). After printing parts, check if the holes are big enough for the passage of screws etc. If it is not due to printing tolerances, drill holes with the proper drill bit.

**Table 10c**
Required printed parts for the assembly of one syringe unit.

| Component               | Number | Material type |
|-------------------------|--------|---------------|
| Stepper_support (1A)    | 1      | PLA           |
| Piston (2A)             | 1      | PLA           |
| Body_syringe_block (3A) | 1      | PLA           |
| Head_block (4A)         | 1      | PLA           |
| End_Stop_support (ES)   | 1      | PLA           |
| Rods_block (RB)         | 8      | PLA           |

Cut the threaded bar and rectified bars according to the length of your syringe, plus an additional 7–8 cm for the joints (see Fig. 4b).
For a detailed description of the building see section “S1. Construction of syringe holder” in the supplementary material of this paper.

4.1.3. Valves block

Each syringe installed on the robot has a valve block composed by 2 three-way motorized valves connected in series. A motorized valve is composed of the 3D printable parts Table 1 (second and third rows) and Table 11 (for a detailed list) and mechanical parts (Table 4).

For a detailed description of the building of this part consider section S2 (Construction of the valve support) in the supplementary material.

4.1.4. Motherboard connections and wiring

Components for the motherboard are listed in Table 5. Motherboard was placed on an acrylic panel of 20 × 30 cm ca. The panel is fixed to the vertical bar of the chassis with two M5 screws (with the corresponding square nuts inserted in the aluminium profile). Holes are drilled in the acrylic panel to fix the motherboard and RC servos board. Plastic spacers are needed to keep the motherboard distanced from the acrylic panel (here we used cut part of plastic tubes).

According to the number of stepper motors installed, insert the corresponding drivers DRV8825. Kind care about their orientation is considered; see Fig. 5 for the motherboard overview.

While plugging/unplugging devices on the motherboard already fixed on the acrylic panel, please take care not to bend the PCB. This could irreversibly damage the circuits.

---

Table 11

| Component           | Number | Material type |
|---------------------|--------|---------------|
| Servomotor_support_p1 | 2      | PLA           |
| Servomotor_support_p2 | 2      | PLA           |
| Valve_controller    | 2      | PLA           |

For a detailed description of the building of this part consider section S2 (Construction of the valve support) in the supplementary material.
Important: move the jumper of JP5V in the position indicated in Fig. 5, in between the pins 1 and 2. This is to take the +5 V power supply directly from the USB.

Stepper motors are connected to the upper connectors X, Y, Z, I, J, K respectively for the motors 1–6 (see Fig. 6). If you are using up to 6 motors, start to connect following the sequence indicated. If it is not, the control program is stuck.

Fix motor wires to the upper horizontal bar of the chassis by using cable ties.
Plug the endstops wires on the board following the connections in Table 12. Fix endstop wires to the middle horizontal bar of the chassis by using cable ties.

Insert the +12 V power supply wires to the motherboard RESPECTING THE POLARITY. Connect hotplate interface (polarity) and thermocouple as indicated in Fig. 6. Before using the steppers, each driver 8825 must be regulated according to the motor current. After switching on the 12 V power supply, by placing the two terminals of the multimeter as in Fig. 7 (one on GND and the other at the centre of the trimmer). The voltage read (Vref) is proportional to the maximum current according to formula:

\[
\text{Max current} = \frac{\text{Vref}}{C^3}
\]

Carefully moving the trimmer, the voltage can be adjusted.

Even if drivers could give currents up to 2A, we suggest staying on a safe maximum value of 1A (Vref = 0.5). In fact, for higher currents, dissipators should be fixed on drivers in order to avoid overheating.

Connect the motherboard to USB and, via Arduino IDE, flash the firmware (marlin.ino). Please check the correct CPU for the compiling (AT Mega 2560).
4.1.5. RC servo motor interface

RC servos can absorb high currents. For this reason, it is not safe to use the +5 V coming from the motherboard. An external power supply is needed for safe use. For our needs a stripboard was used as a connection. See Table 6 for a complete list of components. Since each RC servo has a cable having ground (black), +5 Vcc (red) and control (yellow), we decided to place them on the same 2.54 mm pitch male strip connector soldered on the stripboard. Connections on the stripboard were interrupted as shown in Fig. 8. As it is visible, the GND is also connected to the motherboard. Many wires are soldered to connect all the ground and the +5 V power supply lines.

![Diagram of stripboard connections](image)

**Fig. 7.** Positioning of multimeter terminals (red = positive, black = negative) to measure the voltage Vref (see text). Remember to set the multimeter to DC. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

**Fig. 8.** Scheme of the stripboard for the connection of RC servos. The coloured dots in the centre represent the male strip connector showing colours corresponding to RC servos pins (S0, S1 ...).

The upper part of the stripboard is connected to the motherboard according to Table 13. Connections are made by using the 4×1 pin header wires.
Table 13
Connections between the RC servos and the motherboard.

| Motherboard connector name | PIN number | RC servos interface | RC servo number |
|----------------------------|------------|---------------------|----------------|
| AUX3                       | 3. D49     |                     | 11             |
| AUX3                       | 4. D48     |                     | 9              |
| AUX3                       | 5. D47     |                     | 10             |
| AUX3                       | 6. D46     |                     | 8              |
| AUX3                       | 7. GND     | GND                 |                |
| AUX3                       | 8. GND     | GND                 |                |
| Keypad                     | 3. D45     |                     | 6              |
| Keypad                     | 4. D44     |                     | 2              |
| Keypad                     | 5. D43     |                     | 7              |
| Keypad                     | 6. D42     |                     | 3              |
| Keypad                     | 7. D35     |                     | 4              |
| Keypad                     | 8. D36     |                     | 0              |
| Keypad                     | 9. D34     |                     | 5              |
| Keypad                     | 10. D36    |                     | 1              |

In Fig. 9 it is shown an enlarged view of the connectors used on the motherboard for RC servos control. The four 4×1 headers with wires were placed respectively on:

1. Pins 2, 4, 6, 8 of AUX3
2. Pins 1, 3, 5, 7 of AUX3
3. Pins 4, 6, 8, 10 of Keypad
4. Pins 3, 5, 7, 9 of Keypad

4.1.6. Hot plate interface
The heater we used possesses a 5 pin female DIN plug suitable for external temperature controls. In our model (AREX, Velp Scientifica), the heater is off when one pin is short circuited to GND. Heater also needs a further jumper (see Fig. 10). A complete list of the parts needed is in Table 7.

We choose a transistor solid-state design to limit power requests by the interface and to increase reliability compared to relays. Galvanic insulations are not necessary because they work at low voltage.

The power supply is drawn by the heater to avoid unwanted commutation spikes reaching the motherboard. The very few components are not critical and are available all around the World.

In substance, the 5 V on/off signal coming from the motherboard is negated by the first NPN BJT (Q1) and is amplified and adapted by Q2 for the heater that runs at 12 V. Q2 acts like a switch. The output signal is directly sent to the heater that has a negated logic operation.

Before using the hotplate, make sure that the PID parameters are correct for your setup. In order to do that, set a temperature and check the temperature increase over time. In the case of temperature overshoot or slow heating, change the PID settings. It is possible to change the parameters by sending the following Gcode command:
In the above example, the proportional parameter is set to 100, the integrative parameter to 1.5 and the derivative parameter to 800.

4.2. CPU

The basement is a 3D printed part (Table 1, fourth row, Fig. 11) that is used as a support for electronic components shown in Table 8. The power supply 144 W–12 V must be inserted in the central part of the basement. A Raspberry Pi 3 can be fixed on the side part. The other part is reserved for future expansions (in our case it holds the motherboard of a liquid handling robot).
For a description of the software used on the Raspberry Pi 3 represented in Section 5.

4.3. pH and electrodes interface

The external module for pH/voltage was exploited by using two affordable interfaces for electrodes (see Table 9a). Power supply for the unit could be taken directly from the Arduino board, due to their low power consumption. In Fig. 12 the schematic shows the connections.

Connect Arduino to USB and, via Arduino IDE, flash the firmware (ph2.ino). To interface the device to the CORRO program (see chapter 6), it is only needed to fill the corresponding fields in the configuration.txt file (USB port and 9600 baud).

The interface can be placed inside the 3D printed box (Table 1 fifth row).

4.4. Load cell interface

A balance is always useful when dealing with chemistry. According to needs, a low-cost serial balance could be easily built and interfaced to CORRO by using the scheme in Fig. 13.

As an example of a possible application, the vessel of the waste materials could be placed onto a load cell. In this way it is possible to monitor the quantity of wastes produced.

In fact, several load cells having different weight ranges could be found on the market. In Table 9b we suggest the use of a load cell having a full scale of 100 g. This is suitable for measuring weight in the range 50–100 g with a good precision. To interface the device with the CORRO program (Section 6), it is only needed to fill the corresponding fields in the configuration.txt file (USB port and 9600 baud).

To program the interface, connect Arduino to USB and, via Arduino IDE, flash the firmware (serialscale.ino). Please note that different load cells need to be calibrated, it means to find the correct calibration factor (see serialscale.ino source code). An easy method to find such a parameter (after finding a calibration weight) is the use of the SparkFun_HX711_Calibration sketch.

The interface can be placed inside the 3D printed box (Table 1 sixth row).
5. Configuration and use

The core of the system is the Raspberry PI 3 at which other modules are attached and controlled. The Raspberry PI 3 was prepared with the Raspbian Operating System and realVNC activated for enabling the remote control via PC. Please refer to the instructions present on the Raspberry’s website.

In order to remotely control the Raspberry PI 3, a network cable should be connected to a router having the DHCP or keyboard/mouse/screen. We set up the Raspberry PI 3 to have a fixed IP on the Wi-Fi. Obviously, your Wi-Fi name and password should be inserted manually at this time.

Python v.3.5.3 is needed for the installation of control software. Pronterface is then installed (follow the instructions on the website) and, finally, our control software, CORRO, is copied. In order to link correctly the program to the Pronterface’s libraries open CORRO as a text file and find the following line:

```
sys.path.append('/home/pi/Printrun') #put here your path to Printrun software.
```

The text inside the append command should be modified according to your Pronterface installation directory.

In our installation, the Arduino modules (Syringebot and sensors) were directly programmed from the Raspberry PI 3, so the Arduino IDE was also installed.

An ISO copy of the whole SD flash memory card of our installation is released on request so as to avoid troubles in the installation process. Obviously the operating system and all other programs installed are not being updated but in any case it is ready to use.

After the preparation of Raspberry Pi 3, the Syringebot firmware is flashed into the Syringebot microcontroller. Please carefully check the name of the USB/serial device and remember to choose AT Mega 2560 in the Arduino IDE before flashing.

For the electrodes controller, follow the above instructions but choose Arduino UNO microcontroller and flash the file ph2.ino.

If everything is correct, by using the Pronterface program the Syringebot can be tested by sending directly the GCode commands (choose the correct serial/USB port and set up 250,000 baud rate). This could also be performed by using the Pronterface suite.

As an example, if the RC servos should be reset before the mounting of the valves, the following GCode command must be sent:

```
M280 P0 S0
```

In particular, the above commands should move the RC servo number 0 (P0) at position 0 degrees (S0). To reset also other RC servos substitute P0 with P1 or the number of the RC servo to reset. Please take into consideration that RC servos numbering starts from 0 and is thus in the range 0–11.

Important: the purge exit from the lower exit of each valve block must be let free (i.e. not connected to tubes) to a drain collector connected to waste. In fact, the purge exit could expel some liquid and draw up air to fill the syringe.
5.1. Syringebot operations, use of the CORRO program

IMPORTANT: at every cold start of the Syringebot each syringe must be ‘homed’. If this operation is not executed, the robot does not know the position of the syringe plungers and no operations should be made. In fact, there is also the possibility of damaging hardware. Homing means that the software must move syringe plungers toward their endstops in order to set the highest position as zero. In fact, hardware has no detectors able to measure the position of the plunger during the process. In the homing procedure, all valve blocks must be set to connect the syringe to the purge position so that air can be in taken. In the last phase the syringe plunger is moved to the lower position and valves are closed.

All these actions are performed by the macro called ‘Home ALL’.

Since the chemicals are connected with tubes to the reaction vessel, in order to fill the inlet tube during the first loading and to empty the outlet tube into the reaction vessel, two procedures named, respectively, priming and purging should be carried out.

The priming procedure should be called when a reagent is loaded into the syringe for the first time or when the inlet tube contains a chemical considered degraded. The priming procedure is composed of two phases. In the first phase the valve block connects the inlet tube to the syringe and the syringe draws up an amount of liquid that is greater than the inlet tube volume. In the second phase, the valve block connects the syringe to the purge exit and the excess liquid present in the syringe is expelled. The procedure is followed by a third step in which all valves are closed (Fig. 14).

The syringe purge procedure must instead always follow the last measured addition of one chemical to the reaction vessel. Without this operation the volume added to the vessel is less than expected because some chemical is still inside the outlet tube.

The purge procedure is composed of two phases. In the first phase the valve block connects the syringe to the purge output and a volume of air greater than the outlet tube volume is drawn up in the syringe. In the second phase the air taken by the syringe is pushed into the outlet tube and all the chemical goes into the reaction vessel. As usual, the procedure is followed by a third step with all the valves closed (Fig. 15).

It is very important to know and understand the functioning of the above-mentioned procedures to use the Syringebot program interface properly (see Fig 16).
Fig. 15. Purge procedure sequence.

Fig. 16. CORRO graphical user interface.
5.1.1. Syringe volume calibration

Another important procedure to be executed before starting a chemical reaction is the calibration of the volume withdrawn by each syringe. In fact, the firmware of the Syringebot is based on the plunger movement in millimetres, so, depending on the volume of the syringe, a correlation between the millimetres and the volume must be found. An easy procedure to start with is simply to measure the distance between the 0 sign and the maximum volume sign printed onto the syringe body. Write this value into the file configuration.txt (see below) together with the syringe total volume (separate values with a semicolon). Repeat this process for each type of syringe used. The numbers of lines inserted in the section must correspond to the number of total syringes inserted in the second line of the configuration file.

```
#total syringes
6
#syringe 0 to max sign distance (mm);syringe total volume (ml)
92.5;60
59;10
92.5;60
59;10
92.5;60
59;10
#inlet tube volume (ml)
10
#outlet tube volume (ml)
10
#schematic image of the current configuration
test.png
#USB port for the Syringebot
/dev/ttyUSB1
#baud rate for Syringebot
250000
#USB port for electrodes interface (leave a blank line if not present)
/dev/ttyACM0
#baud rate for electrodes (leave blank if not used)
9600
#USB port for robot (leave a blank line if not present)
#baud rate for robot (leave blank if not used)
#USB port for load cell (leave a blank line if not present)
#baud rate for load cell (leave blank if not used)
```

To complete the setup, values for the volume of the inlet and outlet tube must be inserted according to the tubes used. Kindly do not alter the structure of the file since only a basic parsing is performed by the program.

To check the calibration before starting a chemical reaction, connect with CORRO to the Syringebot, then perform a prime for the syringe and take the necessary volume with the syringe putting the solvent in a cylinder or balance for the test. Be sure also that the outlet tube is completely empty at the end of the task. If it is not so, the outlet tube volume must be increased in the file configuration.txt. Please note that the inlet and/or outlet volume must not be higher than the syringe volume. If this is the case, procedures prime and purge should be called more times in order to have a pumped volume greater than tube volume.

5.2. Graphical control

The program is thought to work in two modes: expert and normal user. In the expert mode many functions are present like direct GCode sending and macro programming, while in the normal user mode a schematic image can be loaded and used for the interface by clicking on its parts.

Modes can be switched runtime by clicking on F1, F2 and left ALT buttons. To control the Syringebot via graphical interface, a couple of PNG images should be created. The first image represents the scheme as it is visible from the user (Fig. 17, left), while the second image is used for the associations of colours to macros (Fig. 17, right) and it is called mask. In the mask file, each different active area must have a different colour. Both the images must be 800×600 pixels. If it is necessary to work with bigger images, the CORRO source should be modified.

The image filename of the scheme must be inserted into the configuration.txt file. The mask image should have the same name but with -mask added to the name (example: test.png and test-mask.png).

The first time the image is created it is not associated with the macros. To create associations, launch the program and click on the parts with the right mouse button. Please note that the macro name is bound with the RGB colour of the mask image, so for each colour only one macro could be used. The macro connection can be deleted by clicking on the associated
zone with the middle button of the mouse. Colour to macro connections is automatically saved when the program is closed (click yes when it is asked to save). Associations are saved in the text file `imagename.binds.txt`.

After associations, by clicking on the image with the left mouse button, the corresponding macro is activated.

As a further help with the programming, the RGB values of the mask colour are passed as arguments to the macro. In Fig. 17, syringes have the red channel of RGB colour that is transformed in the syringe number (after a division by ten). In this case, all the syringes are associated with the same macro.

![Fig. 17. Example of graphical interface (left) and its mask file (right).](image)

### 5.3. Application note. Automatic titration

The SyringeBot, equipped with the pH/electrode interface, was used for this demonstration.

A solution (50 ml) of hydrochloric acid 0.2 M ca. was added in a 250 ml beaker placed over a stirrer and syringe 1 (syringe having 60 ml of total volume) was used for the addition of a solution of potassium hydroxide (1 M).

It is important to note that before starting the procedure, syringe 1 must be primed and also a volume greater (or equal) than the outlet tube volume must be expelled by the outlet tube. This ensures that both inlet and outlet tubes are full of liquid; allowing the exit of a drop as soon as the syringe starts the addition of the potassium hydroxide solution.

This was made by using the following macro (useful for syringe 1):

````
buffer
macro "prime syringe X"1
macro "syrY_X.ml"15,1
print
Before executing the macro, remove the outlet tube from the beaker.
After the completion of the operation, the outlet tube of syringe 1 could be inserted in the beaker.
After this operation, the macro titration (shown below) is finally called. Please note that for successive titrations there is no need to call again the priming procedure.
;
: below parameters to be set
:
  eval $1$,20; total volume in mL to add  
  eval $2$,1; syringe number  
  eval $step$,20; number of additions to make  
  eval $pause$,1; pause in seconds between additions  
:
: above parameters to be setted
:
: procedure starts here
buffer
  eval $pause$,eval$pause$*1000; transforms the value in milliseconds
getsyringeparms $2$
  eval $valve1$,($2$-1)*2
  eval $valve2$,$valve1$+1
  eval $ml$,$syringemax$-$1$*($syringemax$/$syringevol$)
```
macr... valve_X_exit_2valve1 send G1 $axisname$m3$m1$; take $ml$ milliliters send M400,0; wait for task completion macro "valve_X_exit_1"valve1$ml$ macro "valve_X_exit_2"valve2$ml$ for $cyc$ $step$
eval $vol$,($cyc$-1)*($1$/step$); calculate the volume in the syringe for each addition eval $ml$,$syringemax$-$vol$*$syringemax$/$syringevol$ send G1 $axisname$m3$m1$; drop send M400,0 send G4 $pause$,0; wait (number in milliseconds) next macro "valve_X_close"valve1$ml$ macro "valve_X_close"valve2$ml$ print

Please note that in the header of the procedure, titration parameters could be changed according to the needs. Result is shown in Fig. 18. Data is taken from the log.txt file.

Fig. 18 shows the example of automatic titration done by the Syringebot using volumetric syringe present in the robot. There is a purging unit for cleaning the valves and liquid passage tubes cleaned by the necessary liquid. In our case, NaOH (strong base) is taken in a syringe for measurement and adding according to the parameters which is given an instruction. The beaker containing HCl is placed in a stirring bar attached with thermocouple and pH meter. The pH values obtained as a function of time with respect to NaOH added. It is a best sensitive example for proving the caliber of the robot we assembled.

6. Conclusion and outlook

The hardware described here is based on low cost, disposable materials (valves, syringes) readily available on the market. The parts needed for its building are mostly printed (with affordable 3D printers) while other parts are components commonly used in 3D printers. With respect to other similar open hardware described in the literature, this is the cheapest example of interface with medium to big size chemical reactors, also considering the possibility of extension to sensors of different types. The graphical user interface presented here could be a help to young chemists or also experienced chemists that could work online with reduced risk for people. The Syringebot can also be considered a module for bigger workstations having movable pipettes or robotic arms. The possible limitation of this hardware is the chemical compatibility of the disposable material used. In our example, the less resistant parts to chemicals are the Tygon tubes and the body of valves made in polycarbonate. Obviously, parts might be substituted with glass or polyfluorinated polymers. Unfortunately, the latter materials fall in the more expensive parts because they are thought for special applications like HPLC and thus the cost increases by one or two orders of magnitude. Polyethylene or polypropylene valves and tubing might be the cheapest replacement while maintaining a chemical inertia but now for us they were impossible to find on the market. Our future work will be on the addition of other modules such as filtration, liquid/liquid separation or rotavapor interfaces as well as a liquid chromatography interface.

7. Funding

Authors declare no funding to mention.
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

Authors gratefully acknowledge Christian Bartz (“DerAndere”) for helping them in the modification of his firmware Marlin2ForPipetBot (https://github.com/DerAndere1/Marlin) that is the base of the firmware used here. Authors wish to acknowledge also Boris Landoni for his help in finding the proper electronic components and Salvatore Aprea for the glassware and plasticware supplies.

Appendix A. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.ohx.2022.e00310.

References

[1] A. Gervasi, P. Cardol, P.E. Meyer, Open-hardware wireless controller and 3D-printed pumps for efficient liquid manipulation, HardwareX 9 (2021) e00199, https://doi.org/10.1016/j.ohx.2021.e00199.
[2] D.A.V. Medina, L.F. Rodriguez Cabal, F.M. Lanaes, A.J. Santos-Neto, Sample treatment platform for automated integration of microextraction techniques and liquid chromatography analysis, HardwareX 5 (2019) e00056, https://doi.org/10.1016/j.ohx.2019.e00056.
[3] 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.
[4] J.P. Efremson, S. Li, M.D. Lynch, BioSamplir: An open source, low cost automated sampling system for bioreactors, HardwareX 9 (2021) e00177, https://doi.org/10.1016/j.ohx.2021.e00177.
[5] M.C. Carvalho, Miau, a microbalance autosampler, HardwareX 10 (2021) e00215, https://doi.org/10.1016/j.ohx.2021.e00215.
[6] F. Nejatimoharrami, A. Faina, K. Sty, New Capabilities of EvoBot: A Modular, Open-Source Liquid-Handling Robot, SLAS TECHNOLOGY: Translating Life Sciences, Innovation. 22 (5) (2017) 500–506, https://doi.org/10.1177/2472630316689285.
[7] S. Eggert, P. Mieszczanek, C. Meinert, D.W. Hutmacher, OpenWorkstation: A modular open-source technology for automated in vitro workflows, HardwareX 8 (2020) e00152, https://doi.org/10.1016/j.ohx.2020.e00152.
[8] C. Liu, J. Xie, W. Wu, M. Wang, W. Chen, S.B. Idres, J. Rong, L.-W. Deng, S.A. Khan, J. Wu, Automated synthesis of prexasertib and derivatives enabled by continuous-flow solid-phase synthesis, Nat. Chem. 13 (5) (2021) 451–457, https://doi.org/10.1038/s41557-021-00662-w.
[9] J.M. Parrilla-Gutierrez, S. Tsuda, J. Grizou, J. Taylor, A. Henson, L. Cronin, Adaptive artificial evolution of droplet protocells in a 3D-printed fluidic chemorobotic platform with configurable environments, Nat Commun. 8 (2017) 1144, https://doi.org/10.1038/s41467-017-01161-8.
[10] J.W. Taylor, S.A. Eghtesadi, L.J. Points, T. Liu, L. Cronin, Autonomous model protocell division driven by molecular replication, Nat. Commun. 8 (2017) 237, https://doi.org/10.1038/s41467-017-00177-4.
[11] J.M. Granda, L. Donina, V. Dragone, D.-L. Long, L. Cronin, Controlling an organic synthesis robot with machine learning to search for new reactivity, Nature 559 (7714) (2018) 377–381, https://doi.org/10.1038/s41586-018-0307-y.
[12] K. Sanderson, Automation: Chemistry shoots for the Moon, Nature 568 (7753) (2019) 577–579, https://doi.org/10.1038/d41586-019-01246-y.
[13] M. Peplow, Organic synthesis: The robo-chemist, Nature 512 (7512) (2014) 20–22, https://doi.org/10.1038/512020a.