Supporting information

Digitisation of a Modular Plug and Play 3D Printed Continuous Flow System for Chemical Synthesis

Mireia Benito Montaner, Matthew R. Penny and Stephen T. Hilton*

UCL School of Pharmacy, 29-39 Brunswick Square, London WC1N 1AX

† Electronic supplementary information (ESI) available

*Corresponding author: s.hilton@ucl.ac.uk

Mireia Benito Montaner: https://orcid.org/0009-0001-6406-9248
Matthew R. Penny: https://orcid.org/0000-0003-3572-1783
Stephen T. Hilton: https://orcid.org/0000-0001-8782-4499

GITHUB LINK: GitHub - sthilton/Proteus-Flow-Digital-

TABLE OF CONTENTS

S.1 Experimental 10

S.1.1 General Experimental and analysis 10

S.2 Digital Continuous Flow System Design 11

S.2.1 Design of Digital Flow base 11

S.2.2 Ultimaker Cura settings and Tinkercad design for 3D printed components 12

S.2.2.1 Digital Flow Base 12

S.2.2.2 Base Supports 13

S.2.2.3 Lid for Digital Flow Base 14

S.2.3. Assembly of the 3D printed and electronic components Digital Base Flow 15

S.2.3.1 Wire connections 19

S.2.4 Bill of Materials 21

S.2.5 Meguno Interface Platform 22

S.2.6 Arduino code 25

S.3 Phototemperature sensor Module 30

S.3.1 Ultimaker Cura settings and Tinkercad designs for 3D printed components of the Phototemperature Sensor Module 30

S.3.1.1 Phototemperature Sensor Module 31

S.3.1.2 Lid for the Phototemperature Sensor Module 32

S.3.1.3 Reactor Support and air cooling system bottom part 33

S.3.1.4 Reactor Support and air cooling system top part 35

S.3.3 Assembly of the 3D printed and electronic components of the Phototemperature Sensor Module 36

S.3.4 Bill of Materials 39

S.4 Flow rate sensor Module 40

S.4.1 Design of the flow rate Sensor Module 40

S.4.2 Ultimaker Cura settings and Tinkercad designs for 3D printed components of the flow rate sensor module 41

S.4.2.1 Flow rate Sensor Module 42

S.4.2.2 Lid for the Flow rate sensor module 43

S.4.3 Assembly of the 3D printed and electronic components of the flow rate sensor module 43

S.4.4 Bill of Materials 46

S.5 Three Temperature sensor Module 46

S.5.1 Design of the flow rate Sensor Module 46

S.5.2 Ultimaker Cura settings and Tinkercad designs for 3D printed components of the 3 Temperature Sensor Module 47

S.5.2.1 Three Temperature Sensor Module 48

S.5.2.2 Lid for the Three Temperature Sensor Module 49
S.5.3 Assembly of the 3D printed and electronic components of the 3 Temperature Sensor Module 50
S.5.4 Bill of Materials 51
S.6 Scale up version of Digital Continuous Flow System Design 52
S.6.1 Design of Scale up version 52
S.6.2 Tinkercad design and assembly of the base block for the scale up version 52
S.6.2.1 Tinkercad design of scale up Digital Flow Base 53
S.6.2.2 Wiring connections of scale up Digital Flow Base 54
S.6.3. Tinkercad design and assembly of the 3D printed and electronic components scale up Injection block 55
S.6.3.1 Tinkercad design of scale up Digital Injection Block 56
S.6.3.2 Tinkercad design of scale up Digital Injection Block 57
S.6.3.3 Assembly of the 3D printed and electronic components to the scale up Digital Injection Block 59
S.6.4 Design of the 3D printed scale up bottle holder 62
S.6.5 Assembly of Scale up flow system 65
S.6.6 Bill of Materials 65
S.6.7 Meguno Link Interface Platform 66
S.6.8 Arduino Codes 77
S.7 Ultimaker Cura Settings 85
S.7.1. PLA 3D Printing settings without support 85
S.7.2. PLA 3D Printing settings with support 87
S.7.3. PP 3D Printing settings 89
S.8 Desings for 3D printed reactors 91
S.8.1 CDR for the Phototemperature module with a temperature sensor slot 91
S.8.2 CDR with a temperature sensor slot 92
S.8.3 CDR with a probe slot inserted into the channel 93
S.9 Technical data of electronics 94
S.9.1 Arduino boards 94
S.9.2 Relay Shield 95
S.9.3 Temperature Sensor 95
S.9.3.1 Calibration 96
S.9.4 Flow rate sensor 96
S.9.4.1 Flow rate sensor calibration Methanol 99
S.9.4.2 Flow rate sensor calibration IPA 100
S.9.4.3 Flow rate sensor calibration Acetone 101
S.9.4.4 Flow rate sensor calibration Ethanol 102

S.9.4.5 Flow rate sensor calibration Acetonitrile 103
S.10 General reaction procedures 104
S.11 References 113

SUPPLEMENTARY FIGURES

Supplementary Figure 1: Design and fitting of the Digital Base flow on the continuous flow system (Red Square). A) Front view; B) Top view; C) Side view, D) Analog continuous flow system 11
Supplementary Figure 2: 3D printed continuous flow system with the Three Temperature module inserted. A) Front view. B) Side view 12
Supplementary Figure 3: Technical Drawing Showing the Dimensions of the Digital Base Flow Design 12
Supplementary Figure 4: Tinkercad design illustrating the Digital Base Flow. 13
Supplementary Figure 5: Tinkercad design illustrating the planned support for the base component. 13
Supplementary Figure 6: Technical Drawing Showing the Dimensions of the Digital Base Flow lid Design 14
Supplementary Figure 7: Tinkercad design illustrating the planned lid for the base block. 14
Supplementary Figure 8: 3D printed components of the Digital Base flow. A) Bottom view; B) Front view; C) Top view; D) Digital Base flow lid; E) Back view. 15
Supplementary Figure 9: Electronic components 15
Supplementary Figure 10: installation of DB15 and kessil lamp connector. A) Back view; B) Front view 16
Supplementary Figure 11: installation of the Arduino board. 16
Supplementary Figure 12: installation of the Relay shield. 16
Supplementary Figure 13: Arduino wire connections. 17
Supplementary Figure 14: power supply wire connections to the lid. 17
Supplementary Figure 15: A) power supply wire connections to the base; B) power supply wire connections to the Arduino 17
Supplementary Figure 16: Installation of lid 18
Supplementary Figure 17: completion of Digital flow base. A) Back view; B) Lateral view; C) Front view 18
Supplementary Figure 18: Fritzing design showing all the wire connections of the Digital base 19
Supplementary Figure 19: Connection manager tab 23
Supplementary Figure 20: Monitor tab 23
Supplementary Figure 21: MegunoLink Interface of the project using the Arduino Uno Board. A) Welcome screen tab. B) Temperature monitoring tab. C) Flow rate monitoring tab. D) Phototemperature monitoring tab. 24
Supplementary Figure 22: CommandHandler file 26
Supplementary Figure 23: A) Phototemperature module inserted in the digital flow system. B) Phoototemperature module. C) Reactor holder and air cooler 30
Supplementary Figure 24: Tinkercad design illustrating the planned Phototemperature Senor Module 31
Supplementary Figure 25: Technical Drawing Showing the Dimensions of the Phototemperature Sensor Module. 31
Supplementary Figure 26: Tinkercad design illustrating the planned lid for the Phototemperature Module sensor. 32
Supplementary Figure 27: Technical Drawing Showing the Dimensions of the Phototemperature Sensor Module lid 32
Supplementary Figure 28: Tinkercad design illustrating the planned Reactor Support and air cooling system bottom part 33
Supplementary Figure 29: Technical Drawing Showing the Dimensions of the Phototemperature Module reactor holder bottom part 34

Supplementary Figure 30: Tinkercad design illustrating the planned Reactor Support and air cooling system top part 35
Supplementary Figure 31: Technical Drawing Showing the Dimensions of the Phototemperature Module reactor holder top part 35
Supplementary Figure 32: 3D printed and electronic components of the Phototemperature sensor Module 36
Supplementary Figure 33: Installation of 15 pin male connector (Step 1) 36
Supplementary Figure 34: Installation of Step 2. 37
Supplementary Figure 35: Installation of Step 3. 37
Supplementary Figure 36: Wire connections of the sensor 37
Supplementary Figure 37: completion of flow rate sensor module 38
Supplementary Figure 38: reactor support covered in aluminium film 38
Supplementary Figure 39: reactor support bottom part with mirror 38
Supplementary Figure 40: 3D printed reactor on top of the reactor support bottom part 38
Supplementary Figure 41: reactor support covered in aluminium film 39
Supplementary Figure 42: Flow rate module inserted in the digital flow system. 41
Supplementary Figure 43: Tinkercad design illustrating the planned flow rate Senor Module 42
Supplementary Figure 44: Technical Drawing Showing the Dimensions of the flow rate sensor Module 42
Supplementary Figure 45: Tinkercad design illustrating the planned lid for the flow rate sensor Module. 43
Supplementary Figure 46: 3D printed and electronic components of the flow rate sensor Module 44
Supplementary Figure 47: Installation of 15 pin male connector (Step 1) 44
Supplementary Figure 48: Installation of Step 2. 45
Supplementary Figure 49: Wire connections of the sensor 45
Supplementary Figure 50: completion of flow rate sensor module 45
Supplementary Figure 51: Flow rate module inserted in the digital flow system. 47
Supplementary Figure 52: Tinkercad design illustrating the planned Three Temperature Senor Module 48
Supplementary Figure 53: Technical Drawing Showing the Dimensions of the 3 Temperature Sensor Module 48
Supplementary Figure 54: Tinkercad design illustrating the planned lid for the Three Temperature Module sensor. 49
Supplementary Figure 55: 3D printed plug-in Three temperature module. A) Module assembled B) Electronics 50
Supplementary Figure 56: Design and fitting of the Scale up system with the new versions of the Bottle holder and Base and injection block on the continuous flow system (Red blocks). A) Lateral view; B) Side view; C) Front view. 52
Supplementary Figure 57: Tinkercad design illustrating the Digital Base Flow. 53
Supplementary Figure 58: Technical Drawing Showing the Dimensions of the Arduino Mega base block. 53
Supplementary Figure 59: Arduino Mega board wire connections 54
Supplementary Figure 60: Tinkercad design illustrating the scale up Injection block version. 56
Supplementary Figure 61: Technical Drawing Showing the Dimensions of the Injection block 56
Supplementary Figure 62: Tinkercad design illustrating the scale up Injection block lid. 57
Supplementary Figure 63: Technical Drawing Showing the Dimensions of the Injection block lid 57

Supplementary Figure 64: 6 way loop valve 58
Supplementary Figure 65: 3D printed components of the Scale up injection block. 59
Supplementary Figure 66: Components of the Injection block 59
Supplementary Figure 67: installation of union body, lock washer and nut to the injection block lid 60
Supplementary Figure 68: installation of two L-type valves 60
Supplementary Figure 69: installation of two 6-port loop injection valves 60
Supplementary Figure 70: installation of tubbing from the L-type valve to the union body ... 61
Supplementary Figure 71: installation of the waste tube 61
Supplementary Figure 72: installation of two 2mL loop tubing. 61
Supplementary Figure 73: installation of the Relay shield. 62
Supplementary Figure 74: lid secured to the injection block 62
Supplementary Figure 75: Tinkercad design illustrating the scale up bottle holder 63
Supplementary Figure 76: Technical Drawing Showing the Dimensions of the bottle holder for the scale up system 63
Supplementary Figure 77: Three-way valves operation A) Valve closed. B) Valve opened. 64
Supplementary Figure 78: Assembly step by step of the different blocks 65
Supplementary Figure 79: Connection manager tab 67
Supplementary Figure 80: Monitor tab 68
Supplementary Figure 81: MegunoLink Interface of the project. A) Welcome screen tab. B) System configuration tab. C) System control tab. D) Temperature monitoring tab. E) Phototemperature monitoring tab. F) Flow rate monitoring tab. 69
Supplementary Figure 82: Cura preview file showing the Phototemperature Senor Module with the support placement to 3D print. 89
Supplementary Figure 83: Tinkercad design illustrating the planned reactor for the Phototemperature module 91
Supplementary Figure 84: Tinkercad design illustrating the planned reactor with a temperature sensor slot 92
Supplementary Figure 85: Tinkercad design illustrating the planned reactor with a probe slot inserted into the channel 93
Supplementary Figure 86: Flow rate sensor liquid chemical compatibilities. 4 97

SUPPLEMENTARY TABLES

Supplementary Table 1: Ultimaker Cura Settings for 3D printing of Base component 13
Supplementary Table 2: Ultimaker Cura Settings for 3D printing of Support for the base component 13
Supplementary Table 3: Ultimaker Cura Settings for 3D printing of the lid for the Digital Flow Base 14
Supplementary Table 4: Wire connections of Relay and Arduino shield 20
Supplementary Table 5: Bill of Materials for the Digital Flow Base component......... 21
Supplementary Table 6: Ultimaker Cura Settings for 3D printing of the Phototemperature sensor module 31
Supplementary Table 7: Ultimaker Cura Settings for 3D printing of the lid for the Phototemperature Sensor Module......... 33
Supplementary Table 8: Ultimaker Cura Settings for 3D printing of the Reactor Support and air cooling system bottom part 34
Supplementary Table 9: Ultimaker Cura Settings for 3D printing of the Reactor Support and air cooling system top part 36
Supplementary Table 10: Bill of Materials for the Phototemperature Sensor Module 40
Supplementary Table 11: Ultimaker Cura Settings for 3D printing of the flow rate sensor module 43
Supplementary Table 12: Ultimaker Cura Settings for 3D printing of the lid for the flow rate Sensor Module lid 43
Supplementary Table 13: Bill of Materials for the flow rate sensor Module......... 46
Supplementary Table 14: Ultimaker Cura Settings for 3D printing of the Three Temperature sensor module 49
Supplementary Table 15: Ultimaker Cura Settings for 3D printing of the lid for the Three Temperature Sensor Module......... 49
Supplementary Table 16:Bill of Materials for the 3 Temperature Sensor Module 51
Supplementary Table 17: Ultimaker Cura Settings for 3D printing of the base block for the scale up version 53
Supplementary Table 18: Wire connections of Relay shield and Arduino Mega Borad......... 55
Supplementary Table 19: Ultimaker Cura Settings for 3D printing the injection block for the scale up version 57
Supplementary Table 20: Ultimaker Cura Settings for 3D printing the lid of injection block for the scale up version 58
Supplementary Table 21: Ultimaker Cura Settings for 3D the bottle holder for the scale up version 64
Supplementary Table 22:Bill of Materials for the 3 Temperature Sensor Module 66
Supplementary Table 23: Flow rate of Methanol at different pressures and capillaries......... 71
Supplementary Table 24: Flow rate of IPA at different pressures and capillaries 72
Supplementary Table 25: Flow rate of Ethanol at different pressures and capillaries 73
Supplementary Table 26: Flow rate of Acetone at different pressures and capillaries 74
Supplementary Table 27: Flow rate of Water at different pressures and capillaries......... 75
Supplementary Table 28: Flow rate of Acetonitrile at different pressures and capillaries......... 76
Supplementary Table 29: PLA 3D Printing settings without support......... 86
Supplementary Table 30: PLA 3D Printing settings with support......... 88
Supplementary Table 31: PP 3D Printing settings......... 90
Supplementary Table 32: Ultimaker Cura Settings for 3D printing of the reactor for the Phototemperature module......... 91
Supplementary Table 33: Ultimaker Cura Post processing modification settings for 3D printing of the reactor for the Phototemperature module 91

Supplementary Table 34: Ultimaker Cura Settings for 3D printing of the reactor with a temperature sensor slot 92
Supplementary Table 35: Ultimaker Cura Post processing modification settings for 3D printing of the reactor with a temperature sensor slot 92
Supplementary Table 36: Ultimaker Cura Settings for 3D printing of the reactor inserted into the channel 93
Supplementary Table 37: Ultimaker Cura Post processing modification settings for 3D printing of the reactor inserted into the channel 93
Supplementary Table 38: Arduino Uno Wifi Rev 2 and Arduino Mega Features. 2 94
Supplementary Table 39: Relay shield Features. 3 95
Supplementary Table 40: Temperature sensor Features 95
Supplementary Table 41: Supplementary Table 26: Temperature calibrations 96
Supplementary Table 42: Flow rate Sensor Features. 4 96
Supplementary Table 43: Flow rate of Methanol calibrations 99
Supplementary Table 44: Flow rate of IPA calibrations 100
Supplementary Table 45: Flow rate of Acetone calibrations 101
Supplementary Table 46: Flow rate of Ethanol calibrations 102
Supplementary Table 47: Flow rate of Acetonitrile calibrations 103

SUPPLEMENTARY GRAPHS

Supplementary Graph 1: Selected capillaries to measure the flow rate of Methanol at specific pressures ... 71
Supplementary Graph 2: Selected capillaries to measure the flow rate of IPA at specific pressures ... 72
Supplementary Graph 3: Selected capillaries to measure the flow rate of Ethanol at specific pressures ... 73
Supplementary Graph 4: Selected capillaries to measure the flow rate of Acetone at specific pressures ... 74
Supplementary Graph 5: Selected capillaries to measure the flow rate of Water at specific pressures ... 75
Supplementary Graph 6: Selected capillaries to measure the flow rate of Acetonitrile at specific pressures ... 76
Supplementary Graph 7: relation between Temperature from three different temperature sensors and a thermometer ... 96
Supplementary Graph 8: Correlation of the flow rates of multiple solvents and the liquid flow sensor value at room temperature. ... 98
Supplementary Graph 9: Correlation of the flow rates of Methanol and the liquid flow sensor value at room temperature ... 99
Supplementary Graph 10: Correlation of the flow rates of IPA and the liquid flow sensor value at room temperature ... 100
Supplementary Graph 11: Correlation of the flow rates of Acetone and the liquid flow sensor value at room temperature ... 101
Supplementary Graph 12: Correlation of the flow rates of Ethanol and the liquid flow sensor value at room temperature ... 102
Supplementary Graph 13: Correlation of the flow rates of Acetonitrile and the liquid flow sensor value at room temperature. ... 103

S.1 Experimental

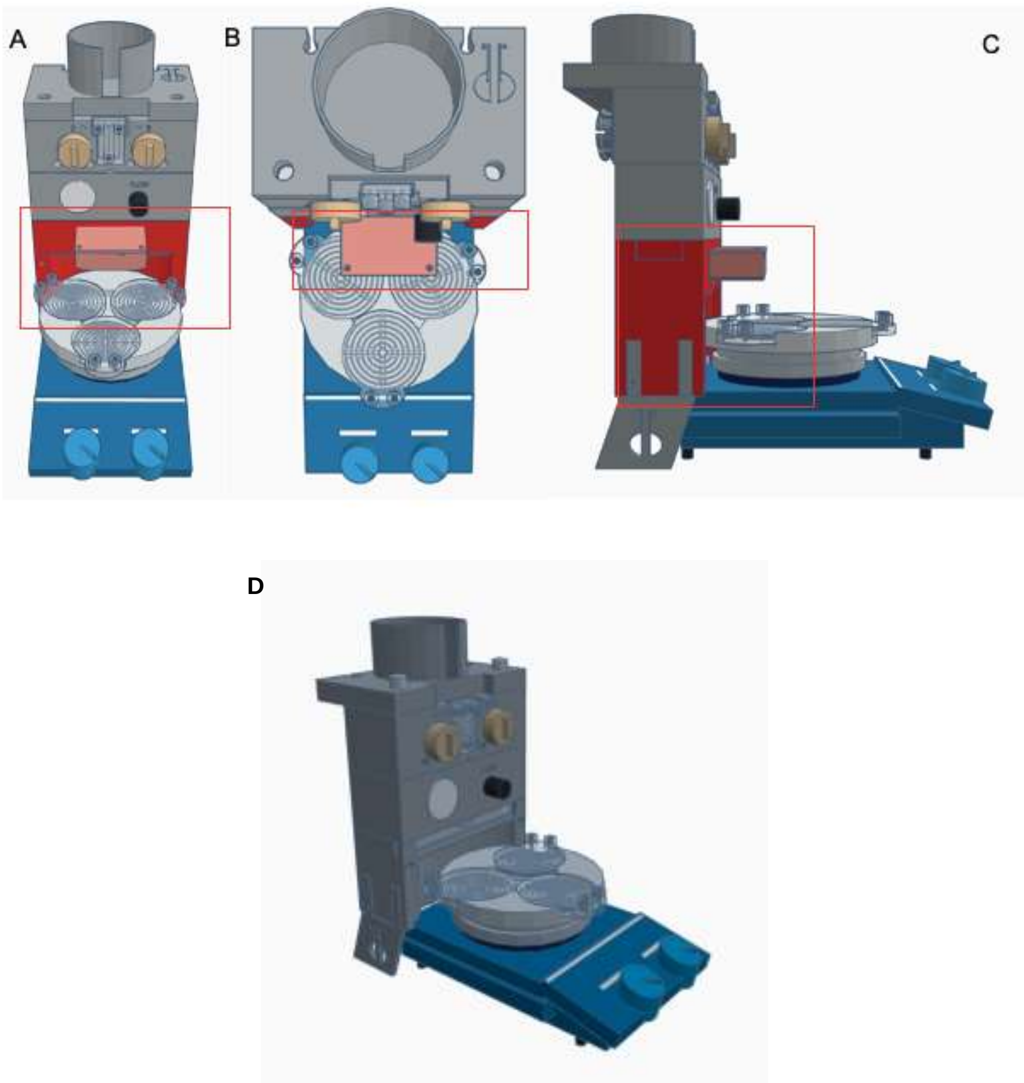
S.1.1 General Experimental and analysis

Stirrer hotplates used in this research were an IKA RCT basic stirrer hotplate and an IKA RCT digital stirrer hotplate. The continuous flow systems used in this research was based on that developed by the Hilton group.[1] All reagents and solvents were purchased from Sigma-Aldrich, Fluka or VWR and used without further purification. The electronic components were purchased from Arduino, Amazon or Keyestudio. Polypropylene (2.85 mm filament) and PLA (2.85 mm filament) was purchased from 3DGBire.[2] 3D Printing was carried out on an Ultimaker 3 3D printer. Metal capillary tubing was purchased from Agilent Technologies. Analytical TLC was carried out on Merck silica gel 60 F_{254} pre-coated plastic plates. Short wave UV (245 nm) were used to visualise components. ^{1}H and ^{13}C NMR data were recorded on a Bruker AV400 spectrometers. Spectra were recorded in deuterochloroform and referenced to residual $CHCl_3$ (^{1}H, 7.26 ppm; ^{13}C, 77.16 ppm). ^{1}H, and ^{13}C spectral data were visualized and processed using MestReNova software. Chemical shifts are expressed in ppm (δ) relative to the standard and coupling constants (J) in Hz. The Arduino IDE software version used was 1.8.19, and the MegunoLink library version used was 1.39.0.

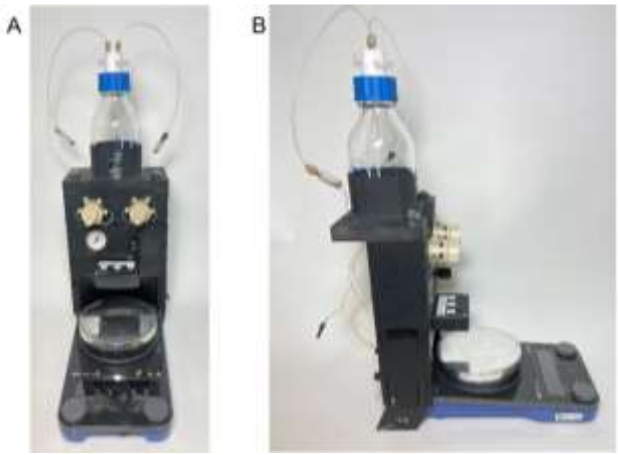
S.2 Digital Continuous Flow System Design

S.2.1 Design of Digital Flow base

The Digital flow base was designed using the freeware web-based application Tinkercad (Autodesk) software.[3] The components were 3D printed using an Ultimaker 3 – 3D printer and PLA (2.85 mm) filament. The base block was designed to replace the previously used base block from the previously described flow system.[1] The internal space was created to have enough space for an Arduino board and Relay shield and wiring connections. This block has a cavity in the front that allows each of the three modules to fit together and be easily interchanged (**Sup. Fig. 1**).



Supplementary Figure 1: Design and fitting of the Digital Base flow on the continuous flow system (Red Square). A) Front view; B) Top view; C) Side view, D) Analog continuous flow system.

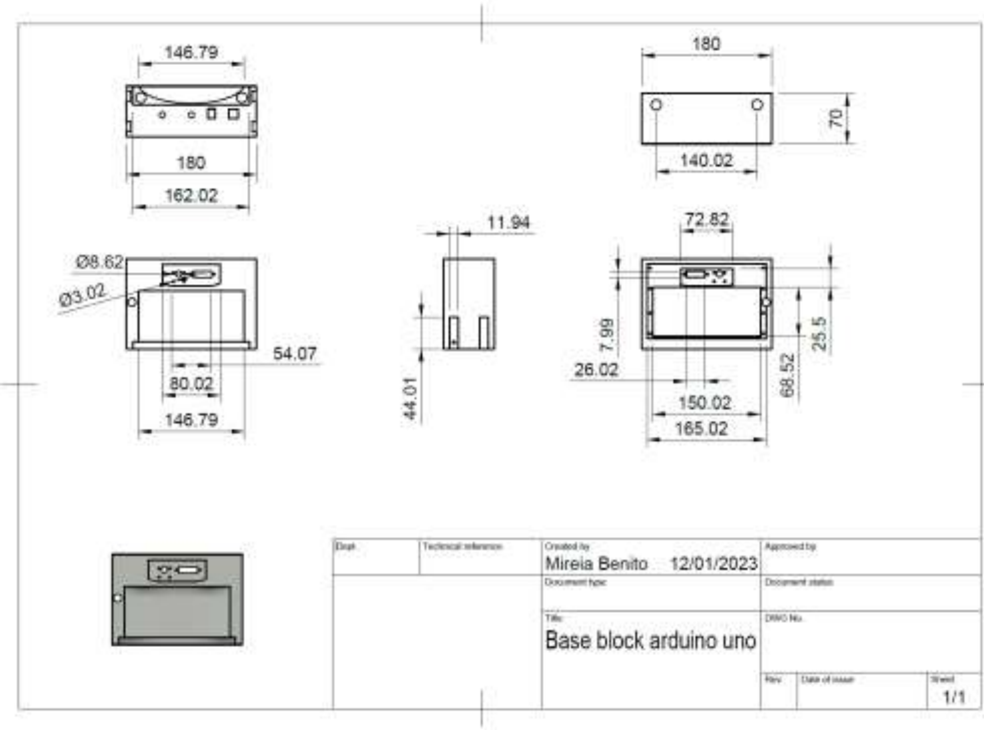


Supplementary Figure 2: 3D printed continuous flow system with the Three Temperature module inserted. A) Front view. B) Side view.

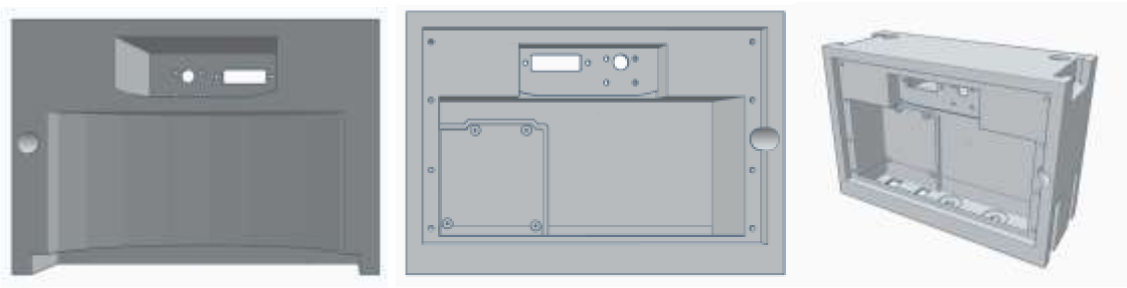
S.2.2 Ultimaker Cura settings and Tinkercad design for 3D printed components

Once the design of the components was completed in Tinkercad, they were exported individually as an STL (Standard Tessellation Language) file and uploaded to Cura software, sliced and 3D printed. The printing settings used for 3D printing the following component are shown in section 7.1.[4]

S.2.2.1 Digital Flow Base



Supplementary Figure 3: Technical Drawing Showing the Dimensions of the Digital Base Flow Design.

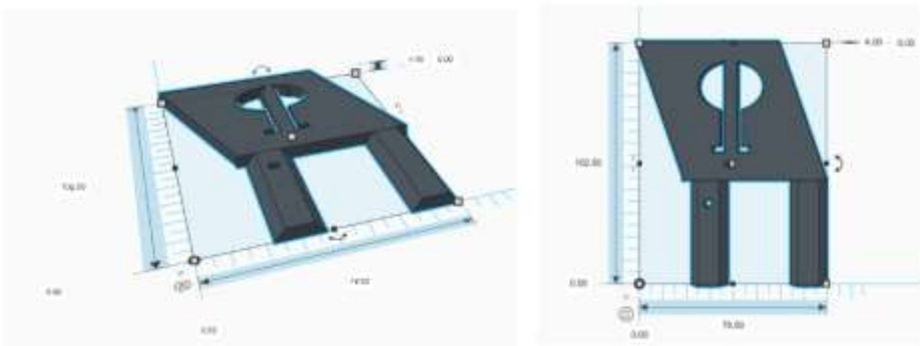


Supplementary Figure 4: Tinkercad design illustrating the Digital Base Flow.

Material	PLA
Time	1 day 18 h 17 min
Quantity	442 g – 55.89 m
Profile	Normal 0.15 mm
Support required	Yes

Supplementary Table 1: Ultimaker Cura Settings for 3D printing of Base component.

S.2.2.2 Base Supports

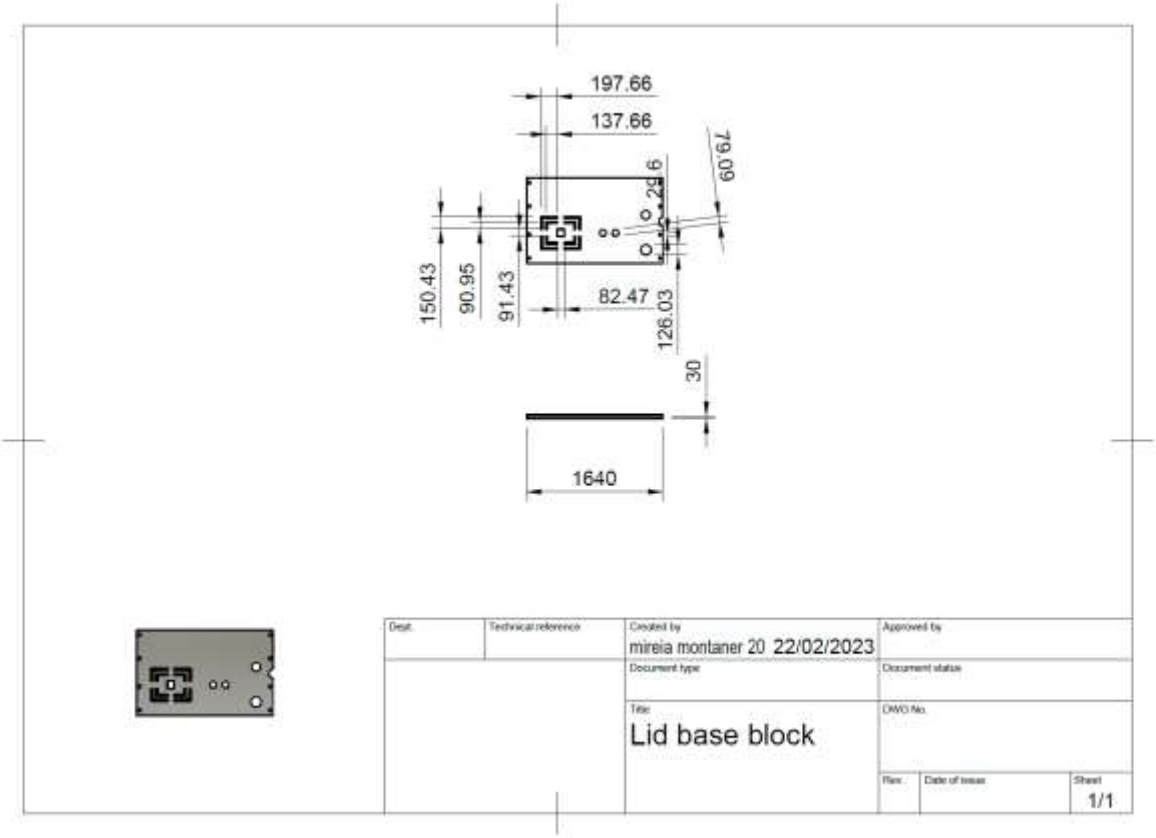


Supplementary Figure 5: Tinkercad design illustrating the planned support for the base component.

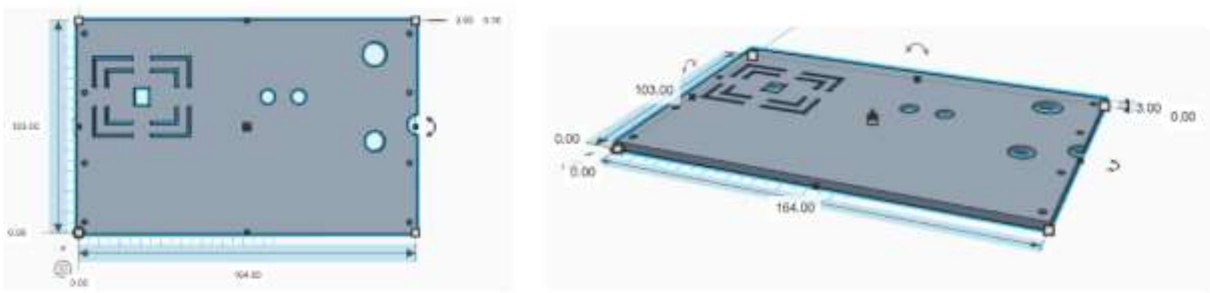
Material	PLA
Time	3 h 46 min
Quantity	41 g – 5.16 m
Profile	Fast 0.2 mm
Support required	No

Supplementary Table 2: Ultimaker Cura Settings for 3D printing of Support for the base component.

S.2.2.3 Lid for Digital Flow Base



Supplementary Figure 6: Technical Drawing Showing the Dimensions of the Digital Base Flow lid Design.



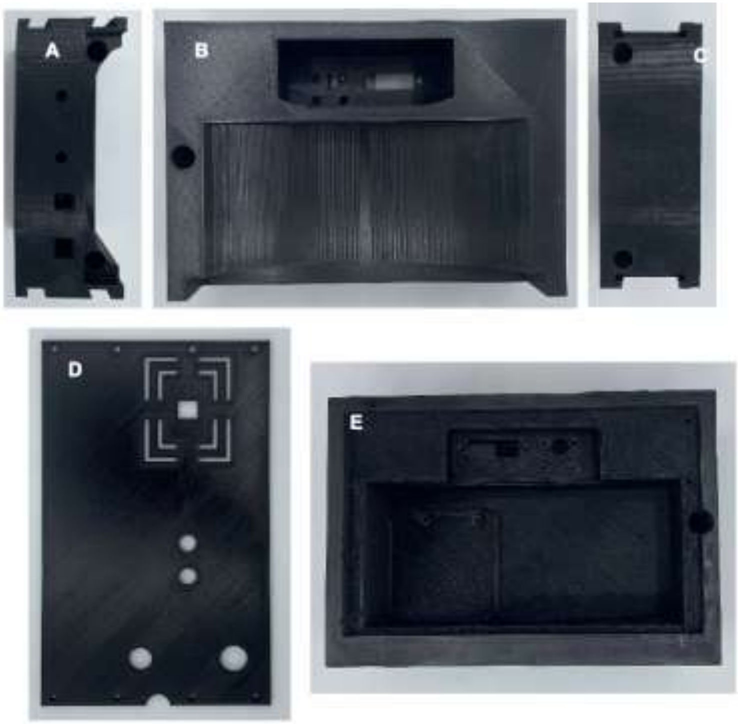
Supplementary Figure 7: Tinkercad design illustrating the planned lid for the base block.

Material	PLA
Time	5 h 24 min
Quantity	61 g – 7.68 m
Profile	Fast 0.2 mm
Support required	No

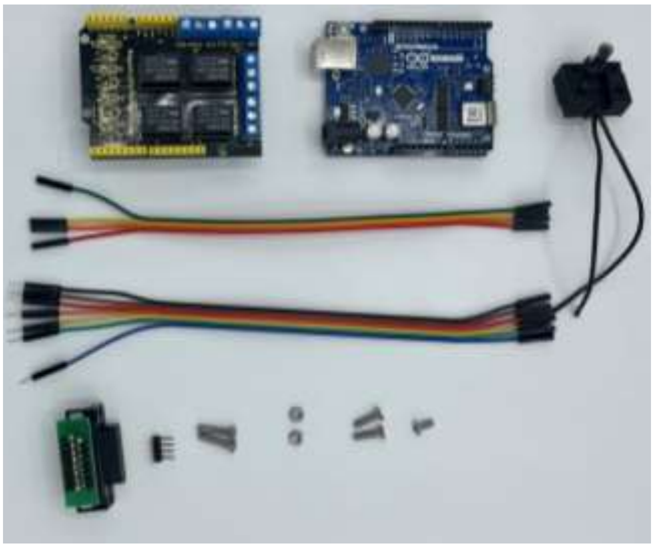
Supplementary Table 3: Ultimaker Cura Settings for 3D printing of the lid for the Digital Flow Base.

S.2.3. Assembly of the 3D printed and electronic components Digital Base Flow

To assemble the Digital Flow base, the 3D printed components were first 3D printed before assembly. The electronic components are built around an Arduino Uno board. The diagram below shows all the 3D printed components (**Sup. Fig. 8**).

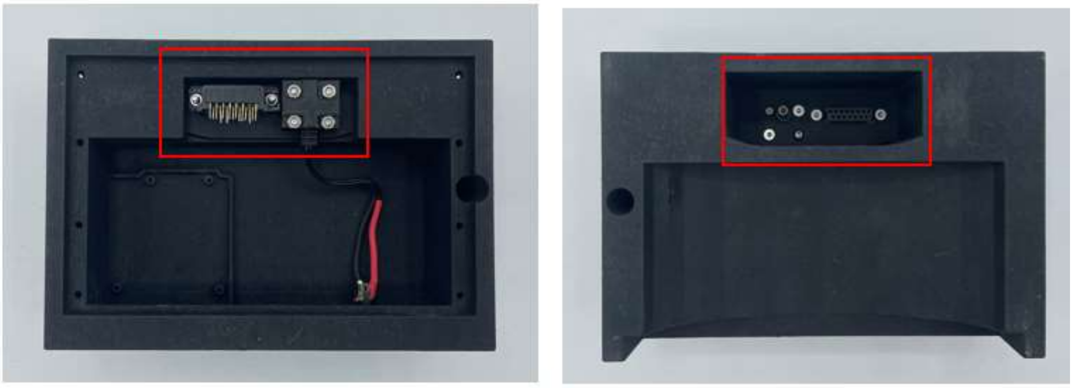


Supplementary Figure 8: 3D printed components of the Digital Base flow. A) Bottom view; B) Front view; C) Top view; D) Digital Base flow lid; E) Back view.



Supplementary Figure 9: Electronic components.

Step 1: secure the DB15 connector to the base with two M3*8 screws and nuts and the connector for the Kessil lamp with four M3*10 screws and associated nuts.



Supplementary Figure 10: installation of DB15 and Kessil lamp connector. A) Back view; B) Front view.

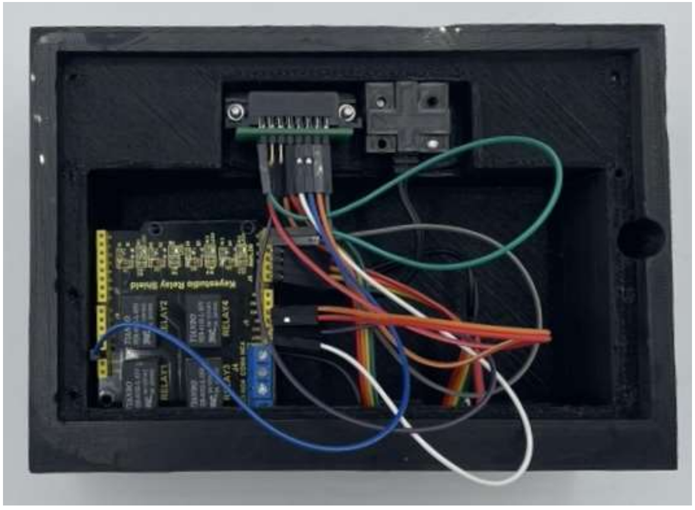
Step 2: secure the Arduino board with four M3*4 screw to the base.

Supplementary Figure 11: installation of the Arduino board.

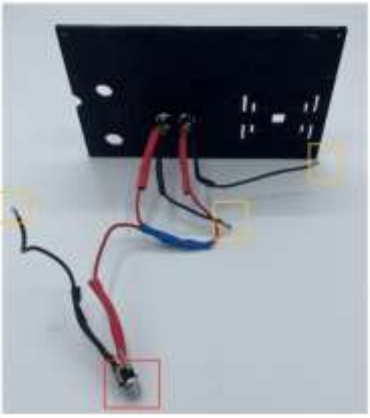
Step 3: mount the relay shield on top of the Arduino board.

Supplementary Figure 12: installation of the relay shield.

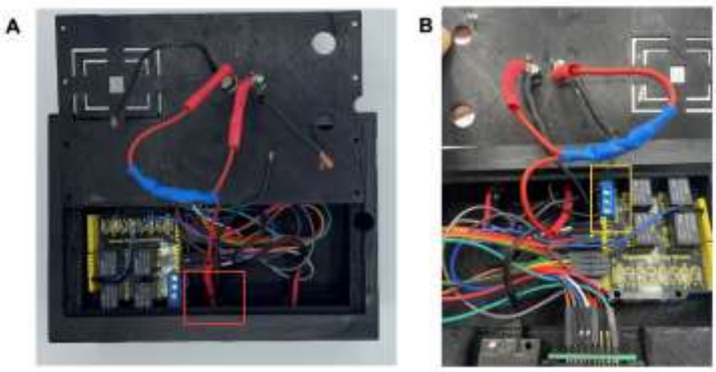
Step 4: the Arduino and power supply wiring is completed in accordance with the Fritzing diagram (Supplemental Figure 17 and table 4).



Supplementary Figure 13: Arduino wire connections.



Supplementary Figure 14: power supply wire connections to the lid.

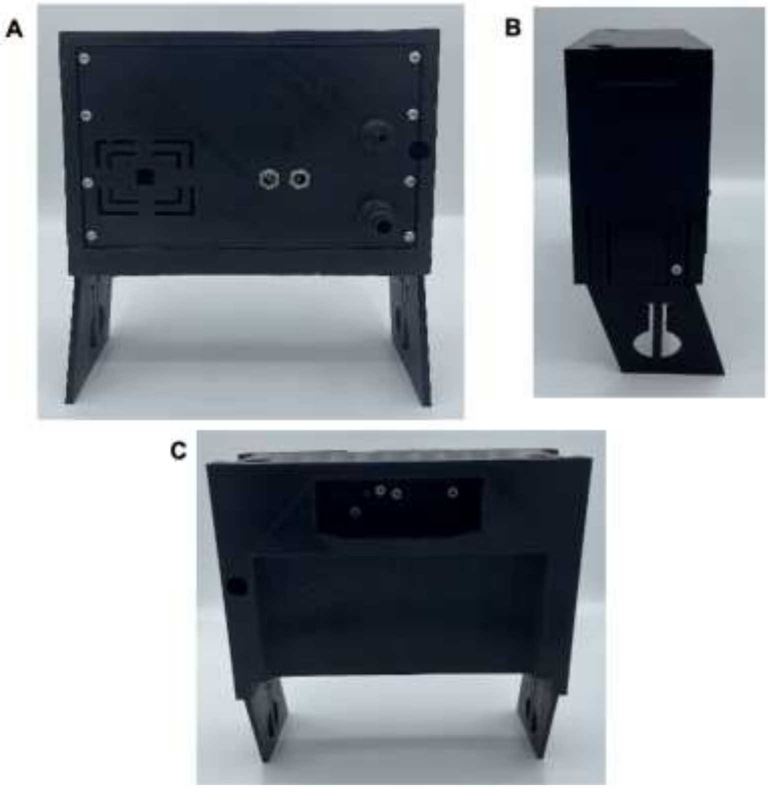


Supplementary Figure 15: A) power supply wire connections to the base; B) power supply wire connections to the Arduino.

Step 5: secure the lid with eight M3*8 screws

Supplementary Figure 16: Installation of lid.

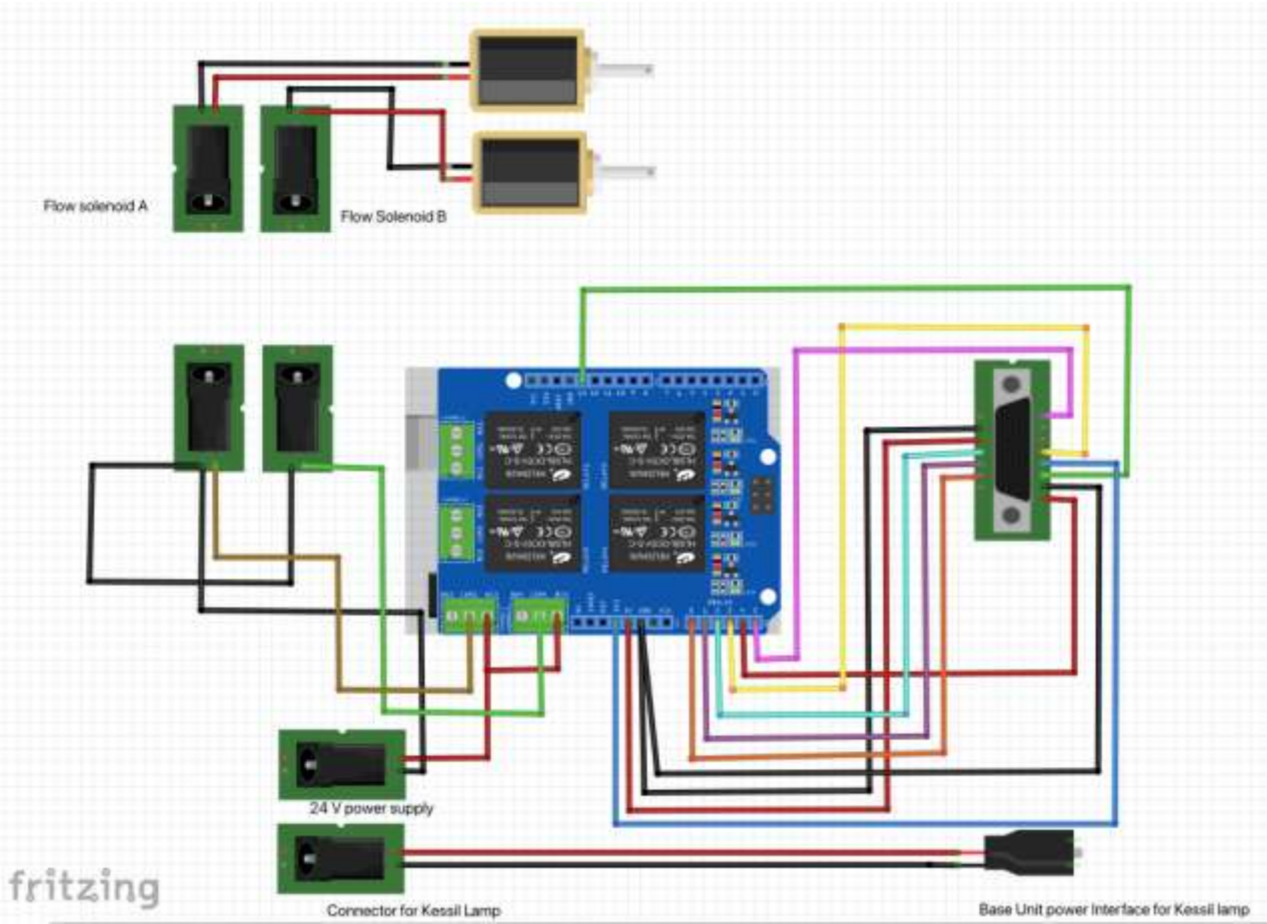
Step 6: secure the support legs to the base with one M3*6 screw on each side.



Supplementary Figure 17: completion of Digital flow base. A) Back view; B) Lateral view; C) Front view.

S.2.3.1 Wire connections

The Fritzing diagram shows all wire connections of the Digital Flow Base including the power supply, Kessil lamp and the microcontroller connections to the DB15 connector.



Supplementary Figure 18: Fritzing design showing all the wire connections of the Digital base.

The following table shows the wire connections from the Arduino board to the DB15 connector and to the sensors of the three different modules (Flow and temperature sensors). Temperature sensors 1–3 correspond to the three temperature modules, while temperature sensor 4 corresponds to the Phototemperature module.

Arduino pin	**DB15 connector**	**Sensor pin**	**Type of sensor**
Analog 5	1	SCL	Flow sensor
Not used	2	Not used	Not used
Not used	3	Not used	Not used
Analog 3	4	SIG	Temperature sensor 4
3V	5	VDD	Flow sensor
Digital 13	6		
GND	7	GND	Temperature sensor 4, flow sensor
Analog 4	8	SDA	Flow sensor
Not used	9	Not used	Not used
GND	10	GND	Temperature sensor 1-2-3
5V	11	UCC	Temperature sensor 1-2-3-4
Analog 2	12	SIG	Temperature sensor 3
Analog 1	13	SIG	Temperature sensor 2
Analog 0	14	SIG	Temperature sensor 1
Not used	15	Not used	Not used

Supplementary Table 4: Wire connections of Relay and Arduino shield.

S.2.4 Bill of Materials

Component Number	Designator	Component	Units	Cost per unit – (GBP £)	Total cost	Source of materials
1	3D printed structure	Digital flow base	1	£23.7	£23.7	3dgbire
2	3D printed structure	Digital flow base lid	1	£2.94	£2.94	3dgbire
3	3D printed structure	Digital flow base supports	2	£1.8	£3.6	3dgbire
4	Electronics	Keyestudio 4 channel Relay shield	1	£19.19	£19.19	Amazon
5	Electronics	Arduino UNO Wifi Rev2	1	£45	£45	Amazon
6	Electronic	Male 12v DC Power Jack Adapter Connector	1	£0.43	£0.43	Amazon
7	Electronic	DB 15 pin connector female	1	£8.99	£8.99	Amazon
8	Electronic	F-M Dupont Wires	7	£0.05	£0.35	Amazon
9	Electronic	F-F Dupont wire	4	£0.05	£0.2	Amazon
10	Fastener	M3*4	1	£0.02	£0.02	Ali express
11	Fastener	M3*8	4	£0.02	£0.08	Ali express
12	Fastener	M3*10	2	£0.02	£0.04	Ali express

Total cost:	**£104.54**

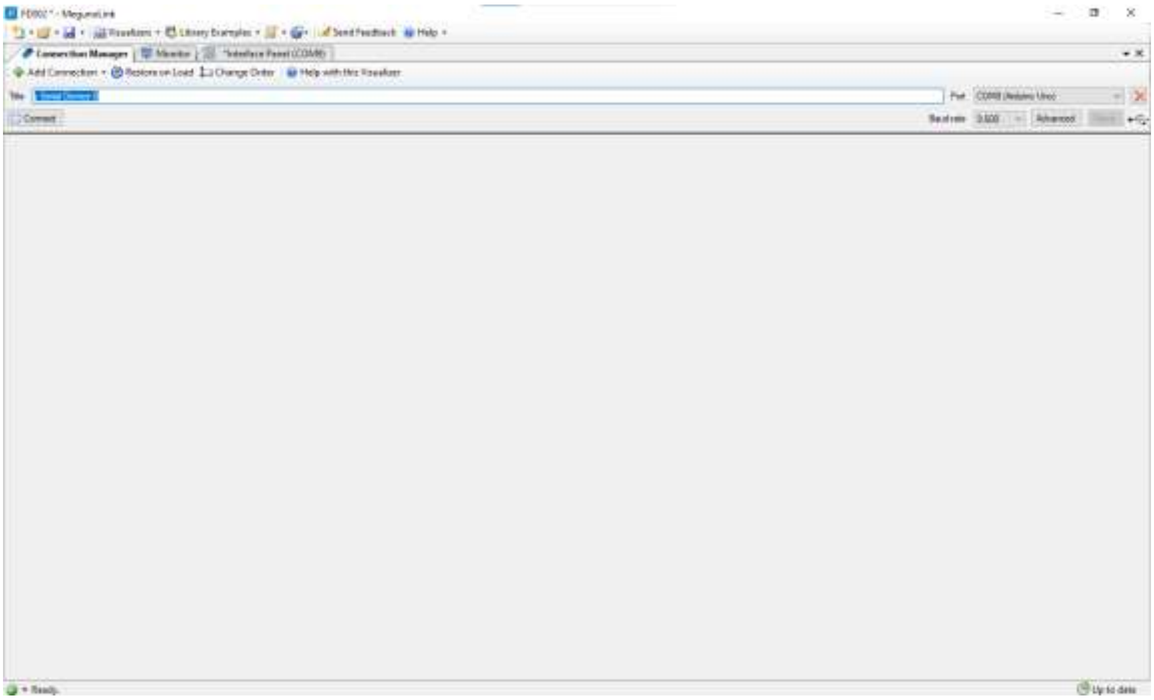
Supplementary Table 5: Bill of Materials for the Digital Flow Base component.

S.2.5 Meguno Interface Platform

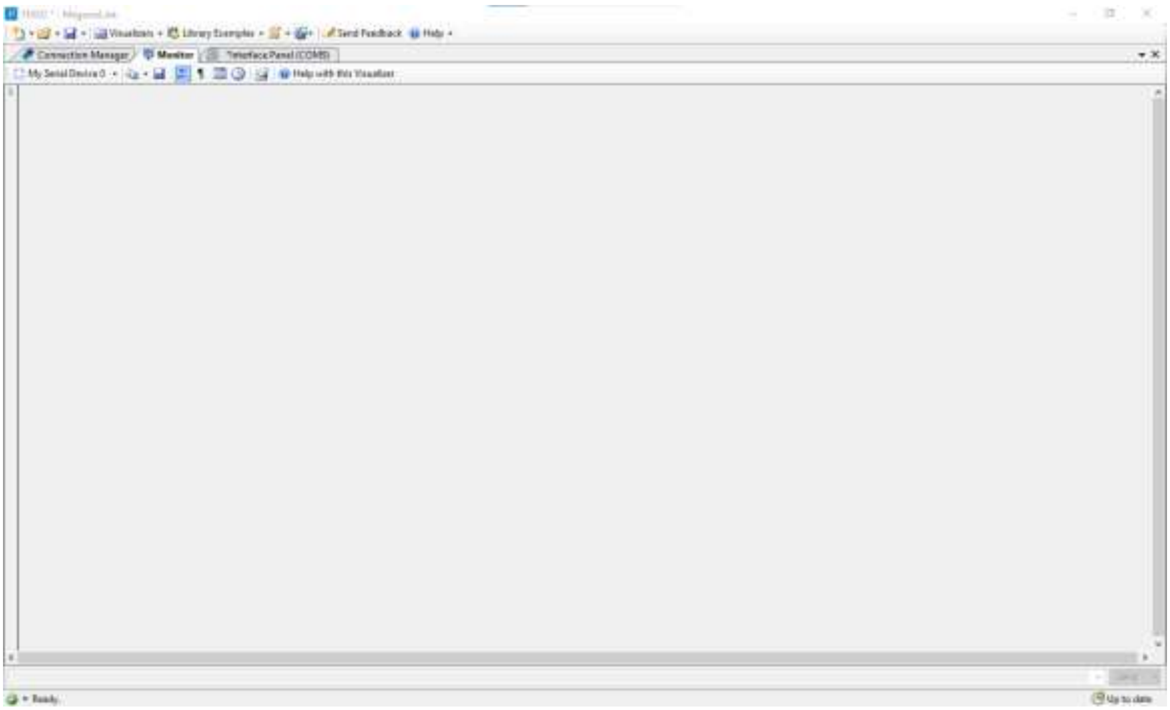
The interface platform that was used to communicate between the Arduino microcontroller and the PC is MegunoLink.[5] This platform can read the data from the sensors and display it in a human readable way. The platform software designed contains 3 tabs:

- Connection Manager (**Sup. Fig. 19**): the platform connects to the Arduino through this tab.
- Monitor (**Sup. Fig. 20**)
- Interface panel (**Sup. Fig. 21**):
    - Welcome screen (**Sup. Fig. 21A**)
    - Temperature monitoring (**Sup. Fig. 21B**): plots the data from the three temperature sensors when the Three Temperature sensor module is plugged.
    - Flow rate monitoring (**Sup. Fig. 21C**): this tab plots the data from the flow rate sensor for when the flow rate module is plugged. Before starting to plot the data, the solvent needs to be selected from the list. All calibrations for the different solvents can be found in section 10.4.
    - PhotoTemperature monitoring (**Sup. Fig. 21D**): plots the data from the temperature sensors when the PhotoTemperature sensor module is plugged.

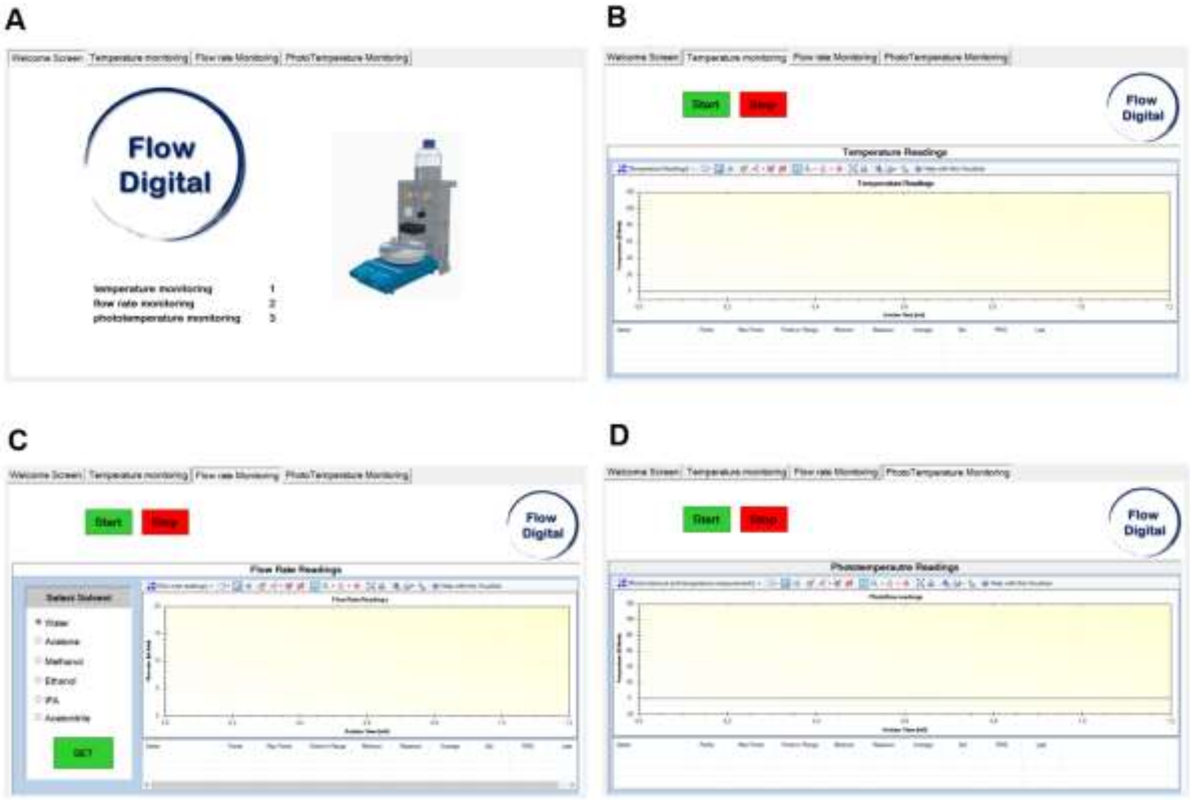
The three monitoring tabs include a start and stop button in order to only record and monitor the sensor data when the button Start is pressed.



Supplementary Figure 19: Connection manager tab.



Supplementary Figure 20: Monitor tab.



Supplementary Figure 21: MegunoLink Interface of the project using the Arduino Uno Board. A) Welcome screen tab. B) Temperature monitoring tab. C) Flow rate monitoring tab. D) Phototemperature monitoring tab.

S.2.6 Arduino code

The MegunoLink website contains a set of functions and methods that can be added to the Arduino code to send and receive data from the interface panel. The Arduino sketch was written using the Arduino IDE software. The three libraries included to the Arduino code were Wire, MegunoLink and CommandHandler. The wire library is used to establish communication with I2C devices which is used by the flow rate sensor. The communication is established using pins A4 for SDA (Serial Data Pin) and A5 for SCL (Serial Clock pin) on the Arduino Uno board. The MegunoLink library allows the Arduino to send data and commands to the Interface panel and its visualizers like plotting a sensor data. The CommandHandler library, in contrast, decodes and processes the commands received from MegunoLink and send them to the Arduino, such as when pressing a button or from a visualizer in the Interface panel. The library can then call an Arduino function or modify an Arduino variable in response to this command. The maximum number of commands and variables that can be called from the Arduino code is stored in the CommandHandler.h file (**Sup. Fig. 22**). When the library is first downloaded, the default values that can be called are 10 commands, 30 command buffers, and 10 variables. The command store size stores the command names and functions while the variable store size stores the map from command name to variable. If more commands have to be handled, these limits can be increased by editing the file, but this will result in increased memory usage. An additional command will require 4 bytes of RAM, whereas an additional variable will require 8 bytes of RAM on an Arduino.

```
#pragma once
#include <Arduino.h>
#include <Stream.h>

#include "utility/CommandDispatcherBase.h"
#include "utility/StreamParser.h"

template <int MAX_COMMANDS = 200, int CP_SERIAL_BUFFER_SIZE = 150, int MAX_VARIABLES=200> class
CommandHandler : public MLP::CommandDispatcherBase, public MLP::StreamParser
{
  // Array of commands we can match & dispatch.
  MLP::CommandCallback m_Commands[MAX_COMMANDS];

  // Array of variables we can match & set/print
  MLP::VariableMap m_Variables[MAX_VARIABLES];

  // Buffer for data received.
  char m_achBuffer[CP_SERIAL_BUFFER_SIZE];

public:

  CommandHandler(Stream &rSourceStream = Serial, char chStartOfMessage = '!', char chEndOfMessage =
'\r')
    : CommandDispatcherBase(m_Commands, MAX_COMMANDS, m_Variables, MAX_VARIABLES)
    , StreamParser(*(static_cast<MLP::CommandDispatcherBase *>(this)), m_achBuffer,
sizeof(m_achBuffer), rSourceStream, chStartOfMessage, chEndOfMessage)
  {
  }
};
```

Supplementary Figure 22: CommandHandler file.

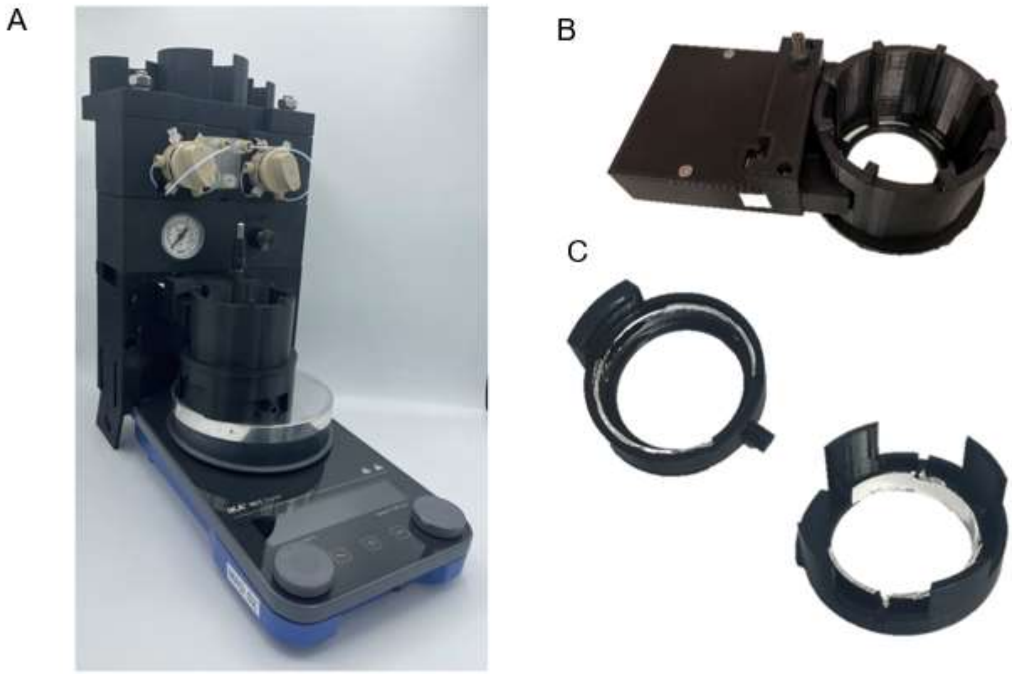
```
#include "MegunoLink.h"
#include "CommandHandler.h"
#include <Wire.h>

#define LEDPIN 13
#define beta 3950
#define resistance 10

int ThermistorPin = 0;
int Vo;
float R1 = 2252;
float logR2, R2, T;
float A = 1.484778004e-03, B =
2.348962910e-04, C = 1.006037158e-07;
unsigned long StartMillis = 0;
float Solvent;
float TotalMillis;
float Eflowrate;
boolean TemperatureStart = false;
boolean TemperatureStop;
boolean FlowStart = false;
boolean FlowStop;
boolean PhotoStart = false;
boolean PhotoStop;
const int ADDRESS = 0x08;
const float SCALE_FACTOR_FLOW =
500.0;
uint16_t sensor_flow_value;
CommandHandler<>
SerialCommandHandler;
long LastSent;

const unsigned SendInterval = 200;

XYPlot TempPlot("Temperature Readings"),
FlowPlot ("Flow rate readings"),
PhotoPlot("Photochemical and temperature
measurements");
InterfacePanel MyPanel;

void
Cmd_SETSYSTEM(CommandParameter
&Parameters)
{Solvent=Parameters.NextParameterAsInte
ger(Solvent); }

void
Cmd_TemperatureStart(CommandParamet
er &params){
 TemperatureStart = true;   }

void
Cmd_TemperatureStop(CommandParamete
r &params){
 TemperatureStart = false;   }

void Cmd_FlowStart(CommandParameter
&params){
 FlowStart = true;   }

void Cmd_FlowStop(CommandParameter
&params){
 FlowStart = false;   }

void Cmd_PhotoStart(CommandParameter
&params){
 PhotoStart = true;   }

void Cmd_PhotoStop(CommandParameter
&params){
 PhotoStart = false;   }

void setup(){

Serial.begin(9600);
Wire.begin();
Serial.println("MegunoLink Pro - Turning
Solenoids on and off");
Serial.println("----------------------------");
SerialCommandHandler.AddCommand(F("S
ETSYSTEM"), Cmd_SETSYSTEM);
SerialCommandHandler.AddCommand(F("T
emperatureStart"), Cmd_TemperatureStart);
SerialCommandHandler.AddCommand(F("T
emperatureStop"), Cmd_TemperatureStop);
SerialCommandHandler.AddCommand(F("F
lowStart"), Cmd_FlowStart);
SerialCommandHandler.AddCommand(F("F
lowStop"), Cmd_FlowStop);
SerialCommandHandler.AddCommand(F("P
hotoStart"), Cmd_PhotoStart);
SerialCommandHandler.AddCommand(F("P
hotoStop"), Cmd_PhotoStop);
```

```
LastSent = millis();

TempPlot.SetSeriesProperties("ADCValue1"
, Plot::Red, Plot::Solid, 2, Plot::Square);
TempPlot.SetSeriesProperties("ADCValue2"
, Plot::Blue, Plot::Solid, 2, Plot::Square);
TempPlot.SetSeriesProperties("ADCValue3"
, Plot::Green, Plot::Solid, 2, Plot::Square);
FlowPlot.SetSeriesProperties("Flow rate",
Plot::Magenta, Plot::Solid, 5, Plot::Circle);
PhotoPlot.SetSeriesProperties("ADCValue4"
, Plot::Black, Plot::Solid, 2, Plot::Triangle);

int ret;
do {
  Wire.beginTransmission(ADDRESS);
  Wire.write(0xFE);
  ret=Wire.endTransmission();
  } while (ret !=0);
}

void loop() {

SerialCommandHandler.Process();

if ((TemperatureStart == true)&& ((millis() -
LastSent) > SendInterval)) {
  LastSent=millis();
  long temp1 =1023 - analogRead (A0);
  float sensor1 = beta /(log(((1025.0 * 10 /
temp1) - 10) / 10) + beta / 298.0) - 273.0;
  long temp2 =1023 - analogRead (A1);
  float sensor2 = beta /(log(((1025.0 * 10 /
temp2) - 10) / 10) + beta / 298.0) - 273.0;
  long temp3 =1023 - analogRead (A2);
  float sensor3 = beta /(log(((1025.0 * 10 /
temp3) - 10) / 10) + beta / 298.0) - 273.0;
  TempPlot.SendData("ADCValue1",
millis(),sensor1);
  TempPlot.SendData("ADCValue2",
millis(),sensor2);
  TempPlot.SendData("ADCValue3",
millis(),sensor3);
  }
if (TemperatureStop== false){
  }

if ((FlowStart == true)&& ((millis() -
LastSent) > SendInterval)&& (Solvent == 1))
{
  LastSent=millis();
   int ret;
    Wire.beginTransmission(ADDRESS);
    Wire.write(0x36);
    Wire.write(0x08);
    ret = Wire.endTransmission();
    Wire.requestFrom(ADDRESS, 9);
    sensor_flow_value  = Wire.read() << 8;
    sensor_flow_value |= Wire.read();
    float flow_value =
((int16_t)sensor_flow_value)/SCALE_FACT
OR_FLOW;
    FlowPlot.SendData("Flow rate",
millis(),flow_value);
  }

else if ((FlowStart == true)&& ((millis() -
LastSent) > SendInterval)&& (Solvent == 2))
{
  LastSent=millis();
   int ret;
    Wire.beginTransmission(ADDRESS);
    Wire.write(0x36);
    Wire.write(0x08);
    ret = Wire.endTransmission();
    Wire.requestFrom(ADDRESS, 9);
    sensor_flow_value  = Wire.read() << 8;
    sensor_flow_value |= Wire.read();
    float flow_value = ((5*(pow(10,(-
5))))*(pow((int16_t)sensor_flow_value,2))+(0
.0051*sensor_flow_value)+0.0215);
    FlowPlot.SendData("Flow rate",
millis(),flow_value);
  }

else if ((FlowStart == true)&& ((millis() -
LastSent) > SendInterval)&& (Solvent == 3))
{
  LastSent=millis();
   int ret;
    Wire.beginTransmission(ADDRESS);
    Wire.write(0x36);
    Wire.write(0x08);
    ret = Wire.endTransmission();
    Wire.requestFrom(ADDRESS, 9);
    sensor_flow_value  = Wire.read() << 8;
    sensor_flow_value |= Wire.read();
    float flow_value = (pow(10,(-
5))*(pow((int16_t)sensor_flow_value,2))+(0.
0071*sensor_flow_value)-0.0918);
```

```
    FlowPlot.SendData("Flow rate", millis(),flow_value);
    FlowPlot.SendData("sensor", millis(),sensor_flow_value);
  }

  else if ((FlowStart == true)&& ((millis() - LastSent) > SendInterval)&& (Solvent == 4))
{
  LastSent=millis();
   int ret;
    Wire.beginTransmission(ADDRESS);
    Wire.write(0x36);
    Wire.write(0x08);
    ret = Wire.endTransmission();
    Wire.requestFrom(ADDRESS, 9);
    sensor_flow_value  = Wire.read() << 8;
    sensor_flow_value |= Wire.read();
    float flow_value = ((2*(pow(10,(-5))))*(pow((int16_t)sensor_flow_value,2))+(0.0085*sensor_flow_value)-0.1453);
    FlowPlot.SendData("Flow rate", millis(),flow_value);
  }

  else if  ((FlowStart == true)&& ((millis() - LastSent) > SendInterval)&& (Solvent == 5))
{
  LastSent=millis();
   int ret;
    Wire.beginTransmission(ADDRESS);
    Wire.write(0x36);
    Wire.write(0x08);
    ret = Wire.endTransmission();
    Wire.requestFrom(ADDRESS, 9);
    sensor_flow_value  = Wire.read() << 8;
    sensor_flow_value |= Wire.read();
    float flow_value = ((4*(pow(10,(-5))))*(pow((int16_t)sensor_flow_value,2))+(0.0054*sensor_flow_value)-0.0335);
    FlowPlot.SendData("Flow rate", millis(),flow_value);
  }

  else if  ((FlowStart == true)&& ((millis() - LastSent) > SendInterval)&& (Solvent == 6))
{
  LastSent=millis();
   int ret;
    Wire.beginTransmission(ADDRESS);
    Wire.write(0x36);
    Wire.write(0x08);
    ret = Wire.endTransmission();
    Wire.requestFrom(ADDRESS, 9);
    sensor_flow_value  = Wire.read() << 8;
    sensor_flow_value |= Wire.read();
    float flow_value = ((3*(pow(10,(-5))))*(pow((int16_t)sensor_flow_value,2))+(0.0022*sensor_flow_value)+0.1508);
    FlowPlot.SendData("Flow rate", millis(),flow_value);
     FlowPlot.SendData("sensor", millis(),sensor_flow_value);
  }

if ((PhotoStart == true)&& ((millis() - LastSent) > SendInterval)) {
  LastSent=millis();
  long temp4 =1023 - analogRead (A3);
  float sensor4 = beta /(log((((1025.0 * 10 / temp4) - 10) / 10) + beta / 298.0) - 273.0;
  PhotoPlot.SendData("ADCValue4", millis(),sensor4);
  }
if (PhotoStop== false){
  }

}
```

S.3 Phototemperature sensor Module

The Phototemperature Module was designed with one temperature adapter and a lamp support. The module fits perfectly into the Digital flow base, which will connect to the Arduino through the DB15 Male connector. The lid was designed to have a hole in order to be able to plug and unplug the temperature sensor to the adapter. The system was modified from that described previously.[6]

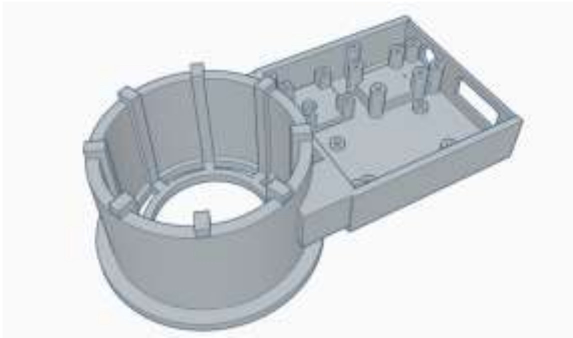


Supplementary Figure 23: A) Phototemperature module inserted in the digital flow system. B) Phototemperature module. C) Reactor holder and air cooler**.**

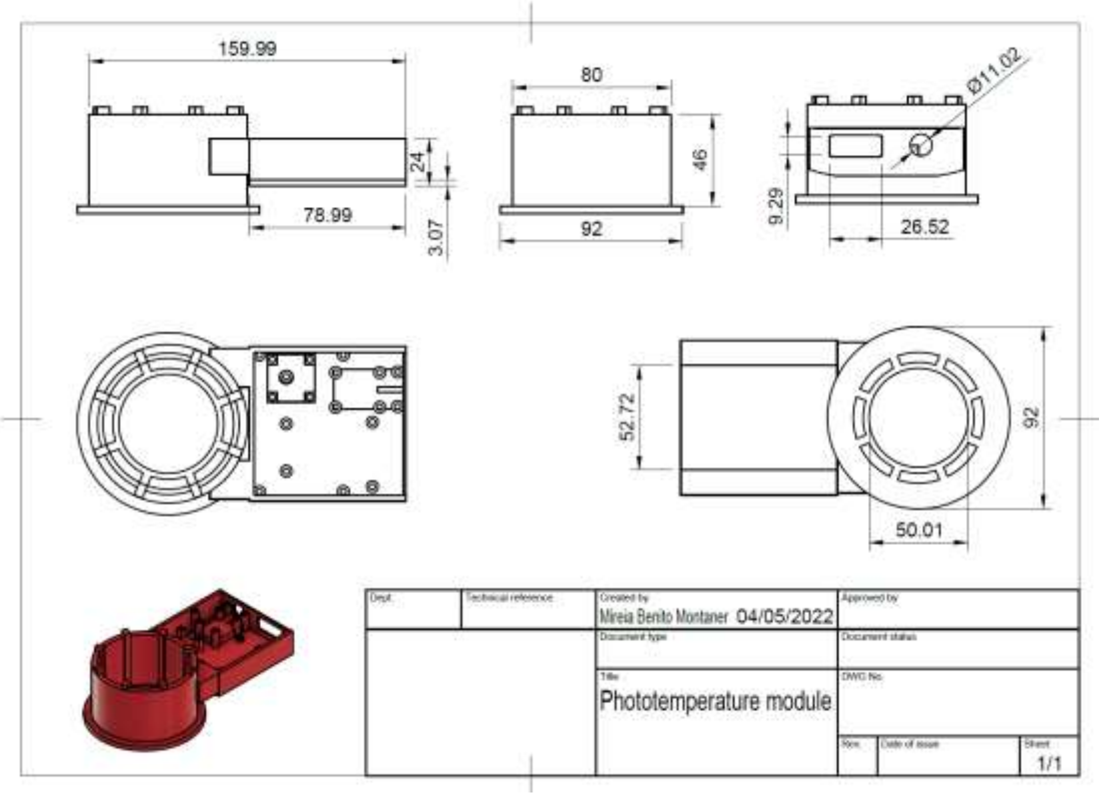
S.3.1 Ultimaker Cura settings and Tinkercad designs for 3D printed components of the Phototemperature Sensor Module

Once the design of the components was completed in Tinkercad, they were exported individually as an STL (Standard Tessellation Language) file and uploaded to Cura software. The printing settings used to 3D print the following component are shown in the tables from section 7.1 and 7.2. Finally, the models were sliced and USB-connected to the printer.

S.3.1.1 <u>Phototemperature Sensor Module</u>



Supplementary Figure 24: Tinkercad design illustrating the planned Phototemperature Senor Module.



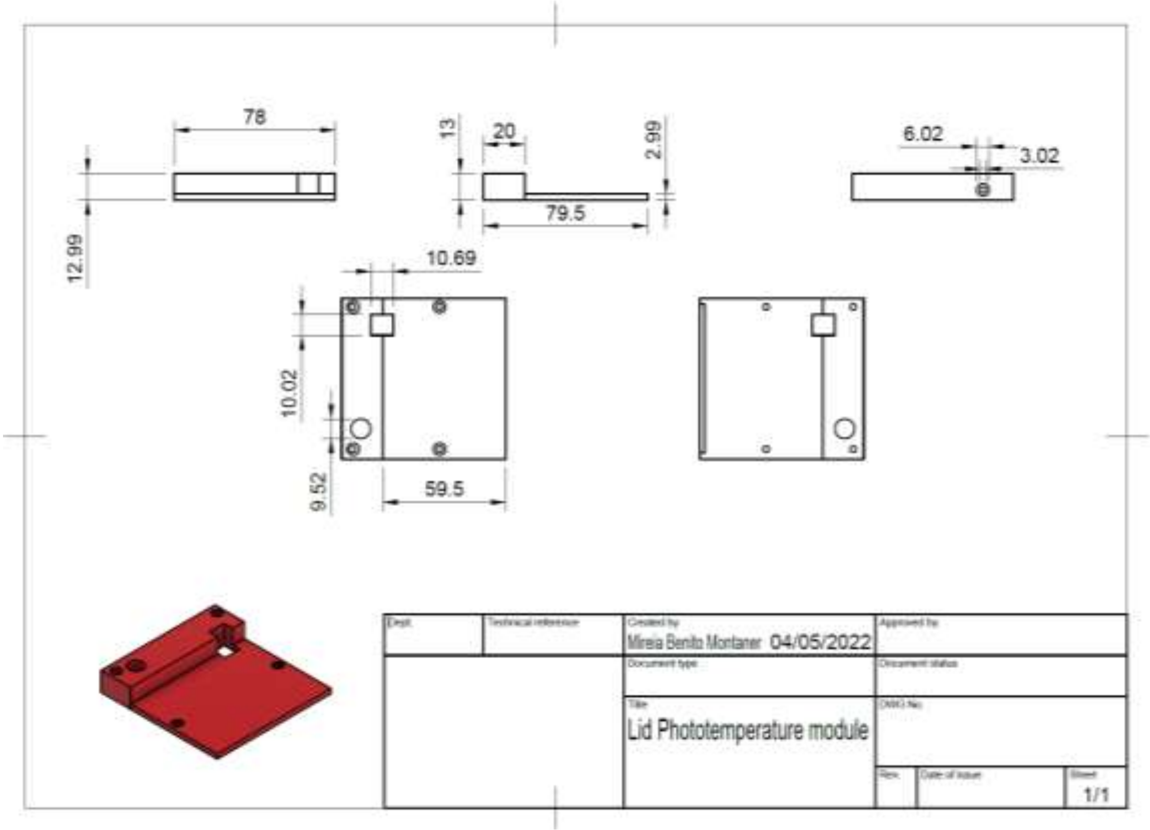
Supplementary Figure 25: Technical Drawing Showing the Dimensions of the Phototemperature Sensor Module.

Material	PLA
Time	18 h 31 min
Quantity	131 g – 16.53 m
Profile	Normal 0.15 mm
Support required	Yes

Supplementary Table 6: Ultimaker Cura Settings for 3D printing of the Phototemperature sensor module.

S.3.1.2 Lid for the Phototemperature Sensor Module

Supplementary Figure 26: Tinkercad design illustrating the planned lid for the Phototemperature Module sensor.

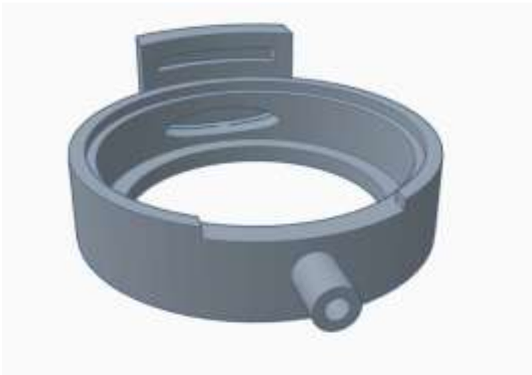


Supplementary Figure 27: Technical Drawing Showing the Dimensions of the Phototemperature Sensor Module lid.

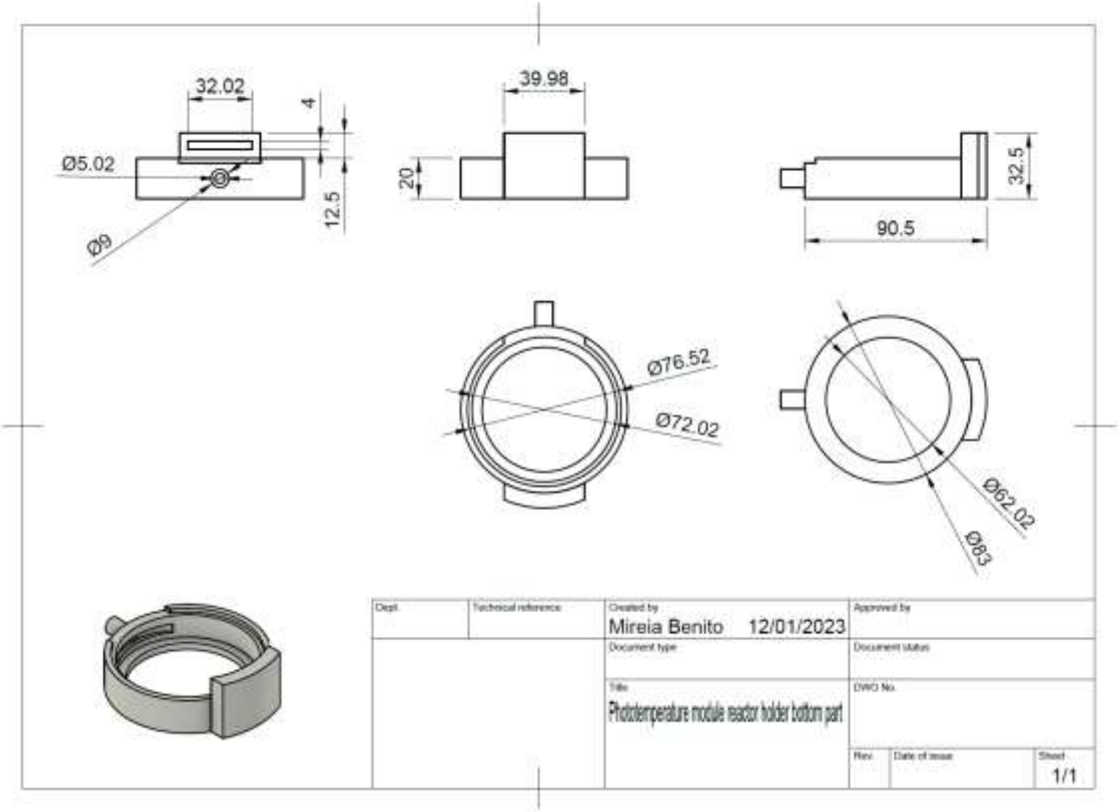
Material	**PLA**
Time	2 h 43 min
Quantity	31 g – 3.95 m
Profile	Fast 0.2 mm
Support required	No

Supplementary Table 7: Ultimaker Cura Settings for 3D printing of the lid for the Phototemperature Sensor Module.

S.3.1.3 Reactor Support and air cooling system bottom part



Supplementary Figure 28: Tinkercad design illustrating the planned Reactor Support and air cooling system bottom part.

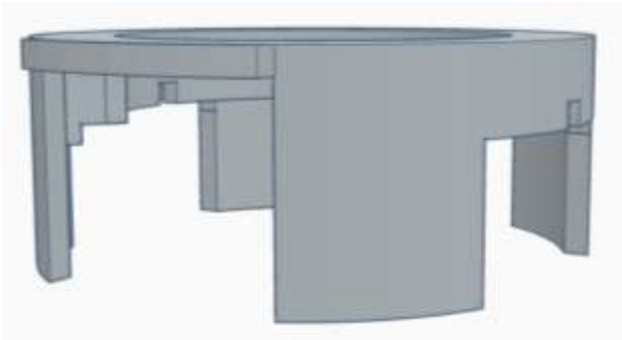


Supplementary Figure 29: Technical Drawing Showing the Dimensions of the Phototemperature Module reactor holder bottom part.

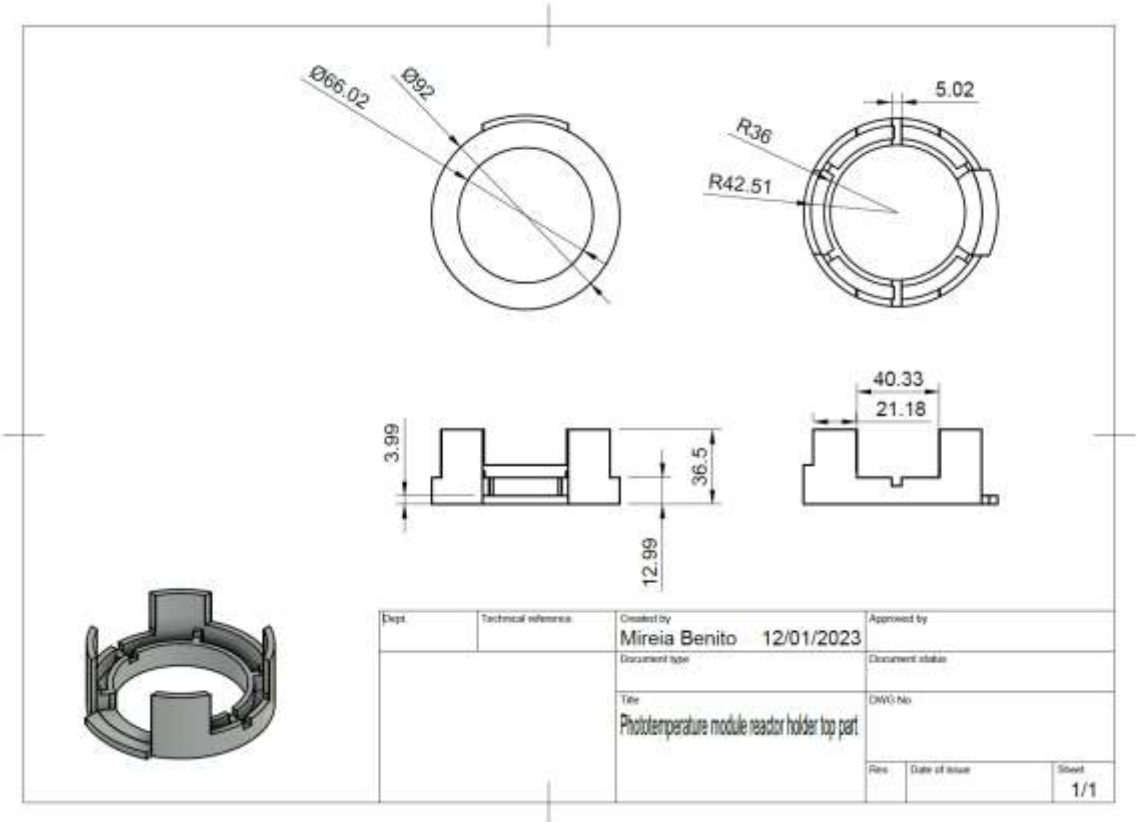
Material	PLA
Time	3 h 20 min
Quantity	37 g – 4.73 m
Profile	Fast 0.2 mm
Support required	No

Supplementary Table 8: Ultimaker Cura Settings for 3D printing of the Reactor Support and air cooling system bottom part.

S.3.1.4 <u>Reactor Support and air cooling system top part</u>



Supplementary Figure 30: Tinkercad design illustrating the planned Reactor Support and air cooling system top part.



Supplementary Figure 31: Technical Drawing Showing the Dimensions of the Phototemperature Module reactor holder top part.

Material	PLA
Time	3 h 32 min
Quantity	40 g – 5.05 m
Profile	Fast 0.2 mm
Support required	No

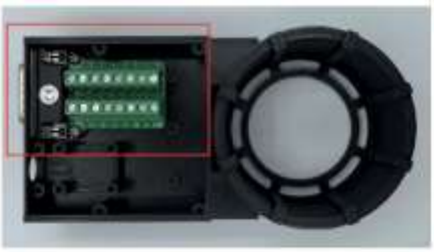
Supplementary Table 9: Ultimaker Cura Settings for 3D printing of the Reactor Support and air cooling system top part.

S.3.3 Assembly of the 3D printed and electronic components of the Phototemperature Sensor Module

To assemble the Digital Flow base, the 3D printed components must first be printed and ready for assembly. The diagram below illustrates the components required to set up the module (**Sup. Fig. 32**).

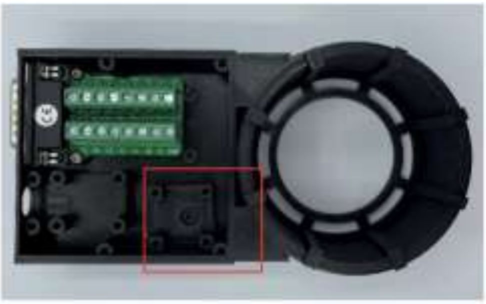
Supplementary Figure 32: 3D printed and electronic components of the Phototemperature sensor Module.

Step 1: The DB15 pin connector male is secured with two M3*5 screws to the base of the Phototemperature sensor Module.



Supplementary Figure 33: Installation of 15 pin male connector (Step 1).

Step 2: The support for the Temperature sensor is secured to the base of the Phototemperature sensor Module.



Supplementary Figure 34: Installation of Step 2.

Step 3: The Temperature sensor adapter is secured to the base of the Phototemperature sensor Module.

Supplementary Figure 35: Installation of Step 3.

Step 4: Wiring is completed in accordance with the Fritzing diagram (**Sup. Fig. 18 and Table 4**).

Supplementary Figure 36: Wire connections of the sensor.

Step 5: The lid is secured to the base unit with two M3*12 (middle) and two M3*16 screws.

Supplementary Figure 37: completion of flow rate sensor module.

Step 6: cover the internal parts of the two reactor supports with aluminum film.

Supplementary Figure 38: reactor support covered in reflective aluminium film.

Step 7: add mirror to the reactor Support and air cooling system bottom part.

Supplementary Figure 39: reactor support bottom part with mirror.

Step 8: place the 3D printer PP reactor on top of the reactor Support and air cooling system bottom part.

Supplementary Figure 40: 3D printed reactor on top of the reactor support bottom part.

Step 7: add the reactor Support and air cooling system top part on top of the 3D printed reactor. Once it has been assembled, the PhotoTemp module can be placed on top (**Sup. Fig. 23A**).

Supplementary Figure 41: reactor support covered in reflective aluminium film.

S.3.4 Bill of Materials

Component Number	Designator	Component	Units	Cost per unit – (GPB - £)	Total cost - (GPB - £)	Source of materials
1	3D printed Structure	Temperature sensor module	1	£7.08	£7.08	3dgbire
2	3D printed Structure	Temperature sensor module lid	1	£1.5	£1.5	3dgbire
3	3D printed Structure	Support and air cooling system top part	1	£2.03	£2.03	3dgbire
4	3D printed Structure	Support and air cooling system bottom part	1	£2.16	£2.16	3dgbire
5	Electronic	Female 12v DC Power Jack Adapter Connector	1	£0.43	£0.43	Amazon
6	Electronic	DB 15 pin connector male	1	£5.51	£5.51	Amazon

7	Electronic	Waterproof NTC Thermistor Temperature Adapter Module	1	£0.93	£0.93	Ali express
8	Electronic	F-M Dupont Wires	3	£0.05	£0.15	Amazon
9	Fastener	M3*12	2	£0.02	£0.04	Ali express
10	Fastener	M3*16	2	£0.02	£0.04	Ali express
11	Fastener	M3*5	2	£0.02	£0.04	Ali express
12	Mirror	Circular mirror	1	£0.99	£0.99	Amazon
13	Tape	Aluminium tape	20cm	£0.02	£0.02	Amazon

Total cost:	**£20.92**

Supplementary Table 10: Bill of Materials for the Phototemperature Sensor Module.

S.4 Flow rate sensor Module

S.4.1 Design of the flow rate Sensor Module

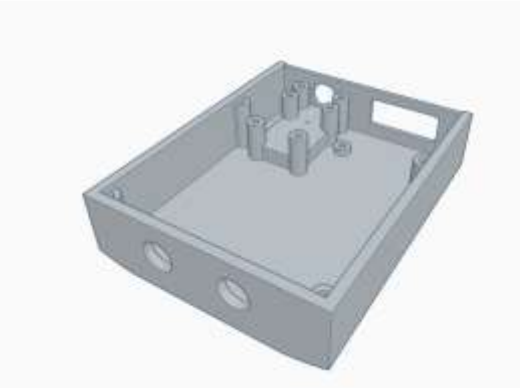
The Flow rate Module was designed with one SLF3C-1300F flow rate sensor. It contains two holes on the front side to connect the system flow path to the sensor without requiring the module to be modified. It also connects to the Arduino in the Digital flow base, through the DB15 Male connector.

Supplementary Figure 42: Flow rate module inserted in the digital flow system.

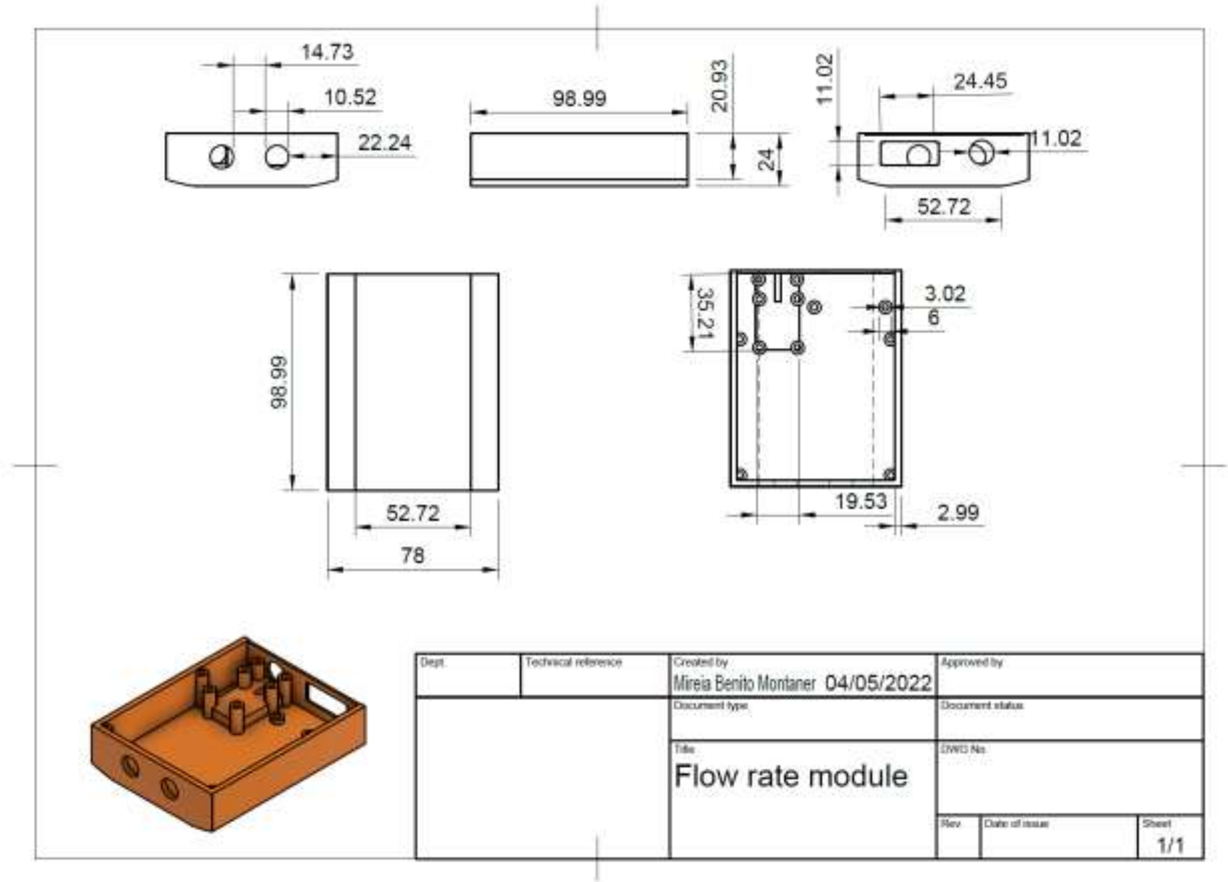
S.4.2 Ultimaker Cura settings and Tinkercad designs for 3D printed components of the flow rate sensor module

Once the design of the components was completed in Tinkercad, they were exported individually as an STL (Standard Tessellation Language) file and uploaded to Cura software. The printing settings used to 3D print the following component are shown in the tables from section 7.1. Finally, the models were sliced and USB-connected to the printer.

S.4.2.1 Flow rate Sensor Module



Supplementary Figure 43: Tinkercad design illustrating the planned flow rate Senor Module.

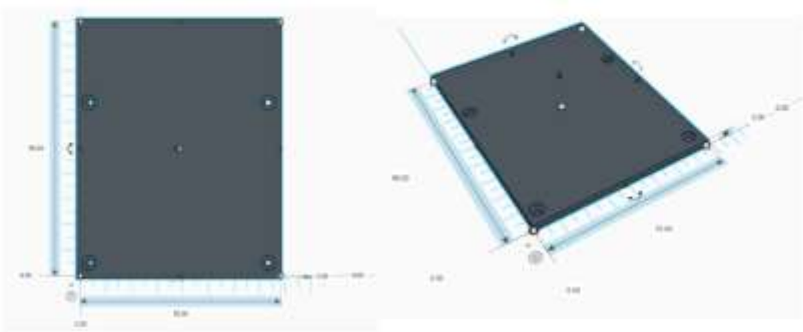


Supplementary Figure 44: Technical Drawing Showing the Dimensions of the flow rate sensor Module.

Material	PLA
Time	8 h 15 min
Quantity	56 g – 7.05 m
Profile	Normal 0.15 mm
Support required	No

Supplementary Table 11: Ultimaker Cura Settings for 3D printing of the flow rate sensor module.

S.4.2.2 Lid for the Flow rate sensor module



Supplementary Figure 45: Tinkercad design illustrating the planned lid for the flow rate sensor Module.

Material	PLA
Time	2 h 30 min
Quantity	28 g – 3.58 m
Profile	Fast 0.2 mm
Support required	No

Supplementary Table 12: Ultimaker Cura Settings for 3D printing of the lid for the flow rate Sensor Module lid.

S.4.3 Assembly of the 3D printed and electronic components of the flow rate sensor module

To assemble the Digital Flow base, the 3D printed components must first be printed and ready for assembly. The diagram below illustrates the components required to set up the module (**Sup. Fig. 46**).

Supplementary Figure 46: 3D printed and electronic components of the flow rate sensor Module.

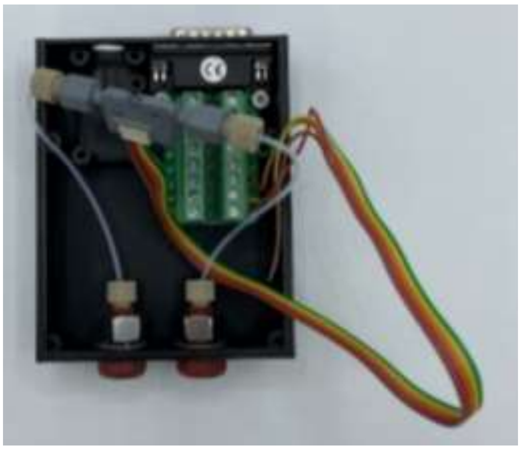
Step 1: The 15 pin connector (male) is secured with two M3*5 screws to the base of the Flow rate sensor Module.

Supplementary Figure 47: Installation of 15 pin male connector (Step 1).

Step 2: slide the union/manifold body through the two front holes, thread the lock washer and nut onto the union/manifold. Tighten the nut until snug.

Supplementary Figure 48: Installation of Step 2.

Step 3: Wiring is completed in accordance with the Fritzing diagram (**Sup. Fig. 18** and **Table 4**).



Supplementary Figure 49: Wire connections of the sensor.

Step 4: The lid is secured to the base unit with four M3*14 screws.

Supplementary Figure 50: completion of flow rate sensor module.

S.4.4 Bill of Materials

Component Number	Designator	Component	Unit	Cost per unit – (GPB - £)	Total cost -	Source of materials
1	3D printed Structure	Temperature sensor module	1	£2.88	£2.88	3dgbire
2	3D printed Structure	Temperature sensor module lid	1	£1.44	£1.44	3dgbire
3	Electronic	15 pin connector male	1	£5.51	£5.51	Amazon
4	Electronic	SLF3S-1300F flow rate sensor	1	194.18	£194.18	Digikey.co.uk
5	Fastener	Union body	2	£20.78	£41.56	Cole-palmer
6	Fastener	Lock washer				
7	Fastener	Nut				
8	Fastener	M3*14	4	£0.02	£0.08	Ali express
9	Fastener	M3*5	2	£0.02	£0.04	Ali express

Total cost:	**£245.69**

Supplementary Table 13: Bill of Materials for the flow rate sensor Module.

S.5 Three Temperature sensor Module

S.5.1 Design of the flow rate Temperature Sensor Module

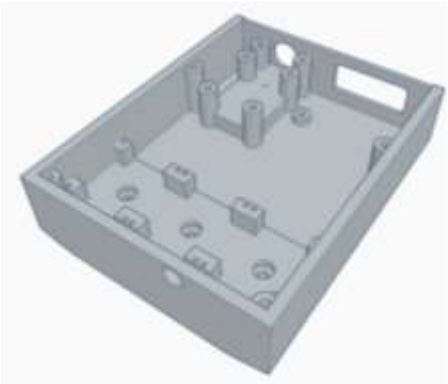
The Temperature Sensor Module was designed to accommodate three temperature sensor adaptors. The module fits into the Digital flow base, which connects to the Arduino through the DB15 Male connector. The lid was designed to have the three holes in order to be able to plug and unplug the waterproof temperature sensors to the adapters.

Supplementary Figure 51: Flow rate module inserted in the digital flow system.

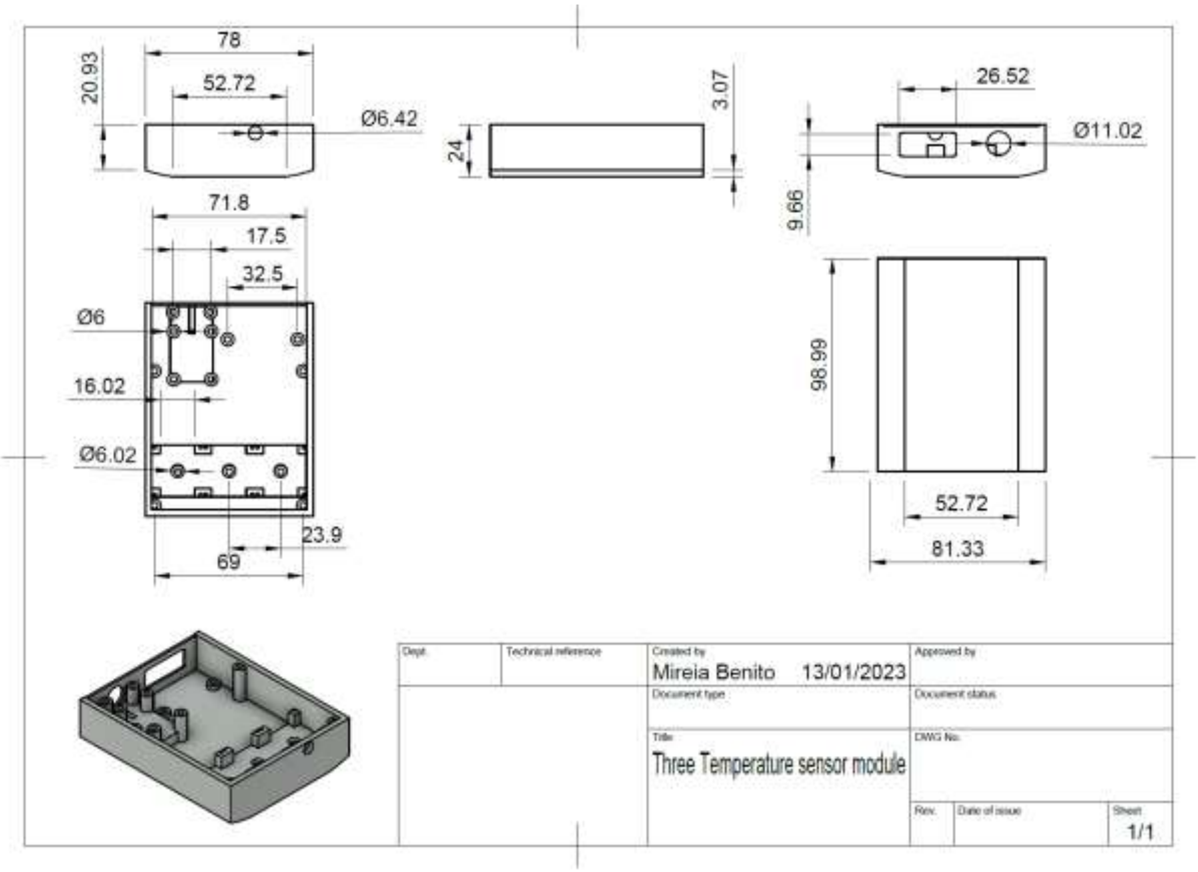
S.5.2 Ultimaker Cura settings and Tinkercad designs for 3D printed components of the 3 Temperature Sensor Module

Once the design of the components was complete in Tinkercad, they were exported individually as an STL (Standard Tessellation Language) file and uploaded to Cura software. The printing settings used to 3D print the following component are shown in the tables from section 7.1. Finally, the models were sliced and USB-connected to the printer.

S.5.2.1 <u>Three Temperature Sensor Module</u>



Supplementary Figure 52: Tinkercad design illustrating the planned Three Temperature Senor Module.

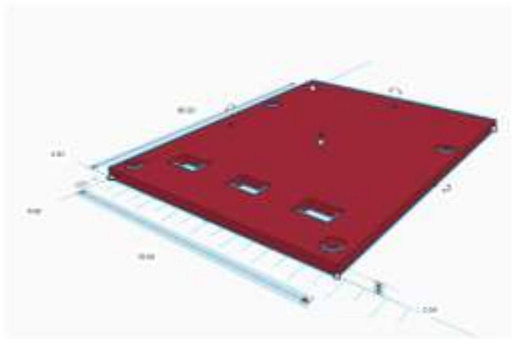


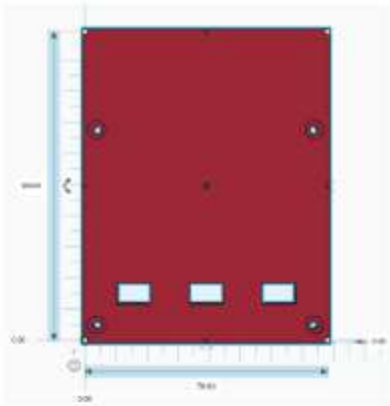
Supplementary Figure 53: Technical Drawing Showing the Dimensions of the 3 Temperature Sensor Module.

Material	PLA
Time	9 h 38 min
Quantity	64 g – 8.05 m
Profile	Normal 0.15 mm
Support required	No

Supplementary Table 14: Ultimaker Cura Settings for 3D printing of the Three Temperature sensor module.

S.5.2.2 Lid for the Three Temperature Sensor Module





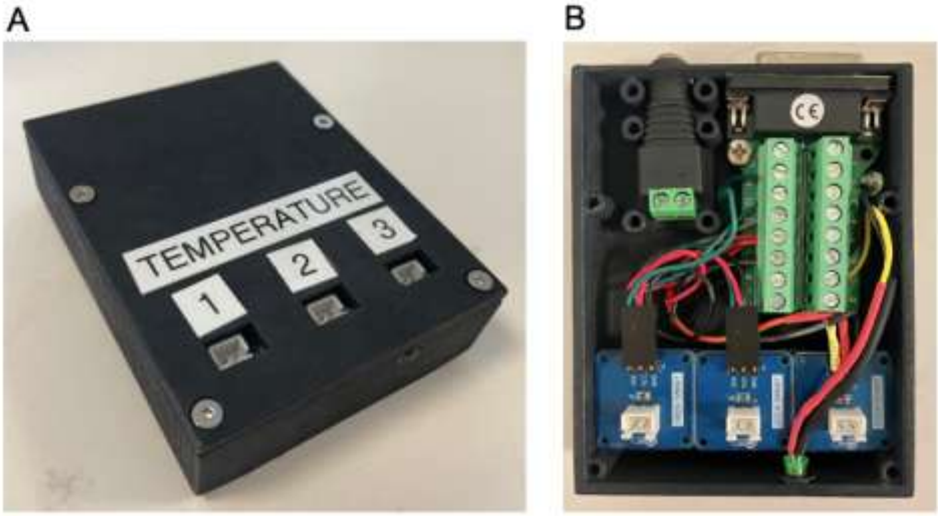
Supplementary Figure 54: Tinkercad design illustrating the planned lid for the Three Temperature Module sensor.

Material	PLA
Time	2 h 29 min
Quantity	28 g – 3.48 m
Profile	Fast 2 mm
Support required	No

Supplementary Table 15: Ultimaker Cura Settings for 3D printing of the lid for the Three Temperature Sensor Module.

S.5.3 Assembly of the 3D printed and electronic components of the 3 Temperature Sensor Module

The assembly of the three temperature sensor module can be followed through the same steps as described for the Phototemperature sensor.



Supplementary Figure 55: 3D printed plug-in Three temperature module. A) Module assembled B) Electronics.

S.5.4 Bill of Materials

Component Number	Designator	Component	Number	Cost per unit – (GPB - £)	Total cost -	Source of materials
1	3D printed Structure	3 temperature sensor module	1	£2.88	£2.88	3dgbire
2	3D printed Structure	3 temperature sensor module lid	1	£1.38	£1.38	3dgbire
3	Electronic	Female 12v DC Power Jack Adapter Connector	1	£0.43	£0.43	Amazon
4	Electronic	15 pin connector male	1	£5.51	£5.51	Amazon
5	Electronic	Waterproof NTC Thermistor Temperature Adapter Module	3	£0.93	£2.79	Amazon
6	Electronic	F-M Dupont Wires	9	£0.05	£0.45	Amazon
7	Fastener	M3*14	4	£0.02	£0.08	Ali express
8	Fastener	M3*5	2	£0.02	£0.04	Ali express

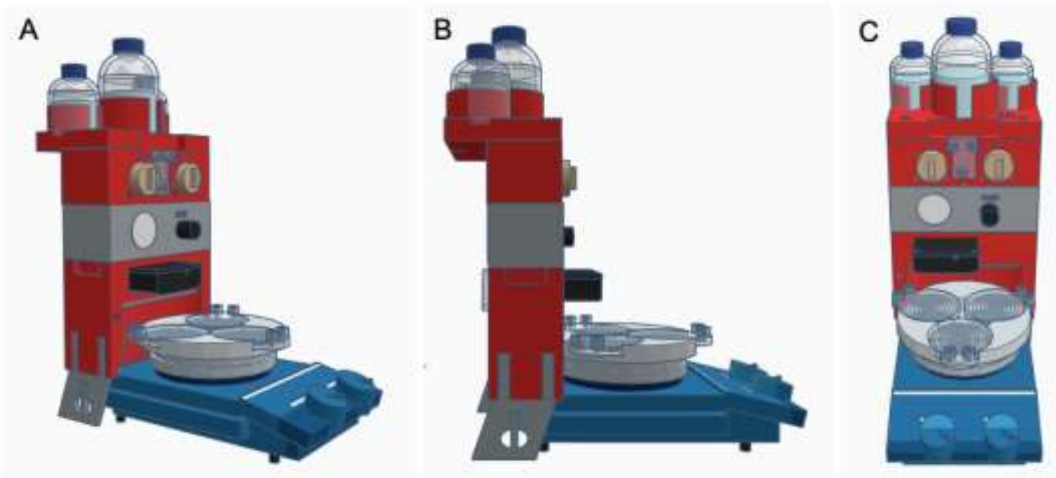
Total cost:	**£13.56**

Supplementary Table 16: Bill of Materials for the 3 Temperature Sensor Module.

S.6 Scale up version of Digital Continuous Flow System Design

S.6.1 Design of Scale up version

The scale up system was mainly designed to enable the injection of higher reactant volumes for larger scale reactions. The base block was slightly modified to accommodate an Arduino Mega in the rear of the interior. As the Arduino mega has larger dimensions than the Arduino Uno Rev 2, the module slot in the front cavity was shifted to the side (**Sup. Fig. 56**). The base block lid used was the same as the previous block designed and the electronics (**Sup. Fig. 6-7**). The injection block was also modified from the previous versions as it required the incorporation of two valves at the back interior to automate the injection of solvent or reactant. The new version of the bottle holder contains three bottle holders, one for a 250 mL Duran bottle and two for 100 mL Duran bottles.

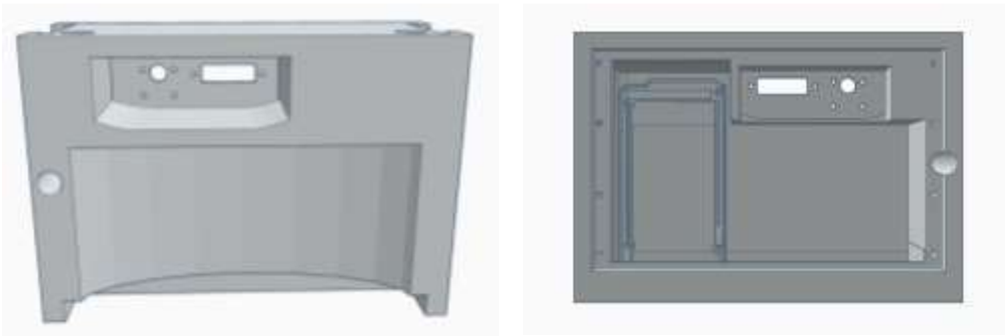


Supplementary Figure 56: Design and fitting of the Scale up system with the new versions of the Bottle holder and Base and injection block on the continuous flow system (Red blocks). A) Lateral view; B) Side view; C) Front view.

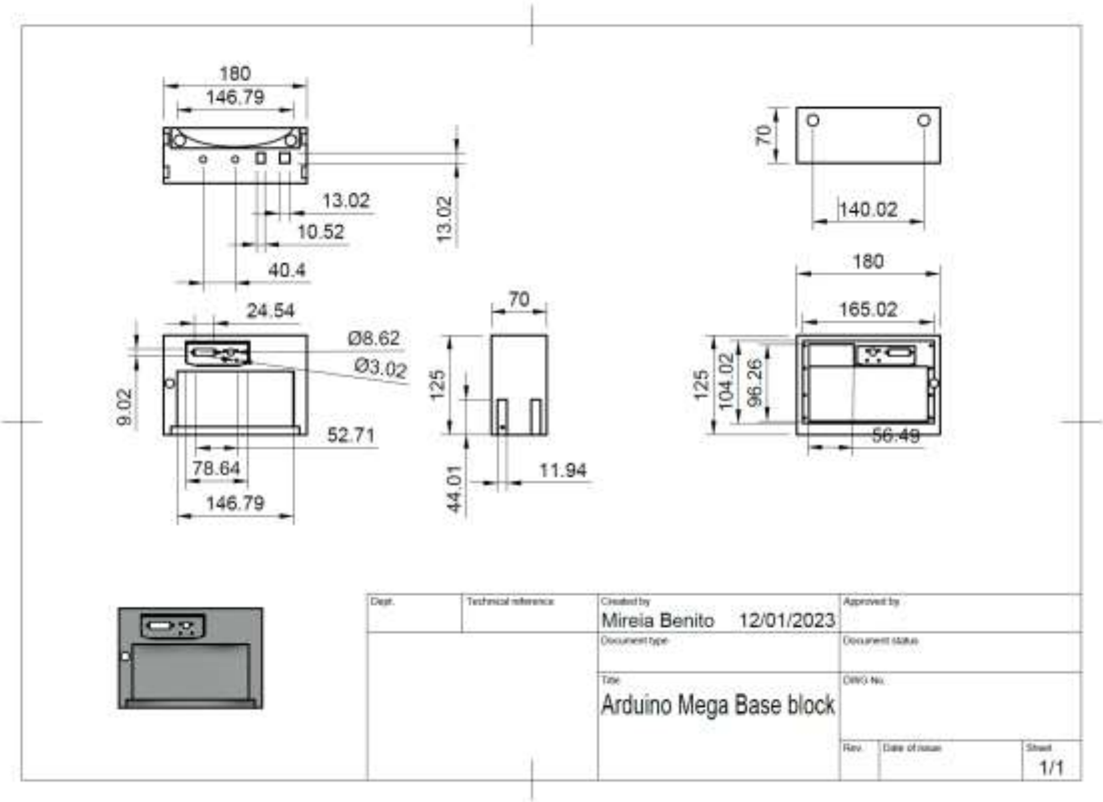
S.6.2 Tinkercad design and assembly of the base block for the scale up version

Once the design of the components was complete in Tinkercad, they were exported individually as an STL (Standard Tessellation Language) file and uploaded to Cura software. The printing settings used to 3D print the following component are shown in the tables from section 7.1. Finally, the models were sliced and USB-connected to the printer.

S.6.2.1 Tinkercad design of scale up Digital Flow Base



Supplementary Figure 57: Tinkercad design illustrating the Digital Base Flow.



Supplementary Figure 58: Technical Drawing Showing the Dimensions of the Arduino Mega base block.

Material	**PLA**
Time	1 day 19 h 25 min
Quantity	444 g – 56.16 m
Profile	Normal 0.15 mm
Support required	Yes

Supplementary Table 17: Ultimaker Cura Settings for 3D printing of the base block for the scale up version.

S.6.2.2 Wiring connections of scale up Digital Flow Base

The assembly of the 3D printed components of the Scale up Digital Flow base and the electronics can be followed in the same manner as section S.2.3. The electronic components are built around an Arduino Mega board instead of an Arduino Uno (**Sup. Fig. 59**).

Supplementary Figure 59: Arduino Mega board wire connections.

The following table shows the wire connections from the Arduino Mega board to the DB15 connector and to the sensors of the three different modules (Flow and temperature sensors). Temperature sensors 1–3 correspond to the three temperature modules, while temperature sensor 4 corresponds to the Phototemperature module.

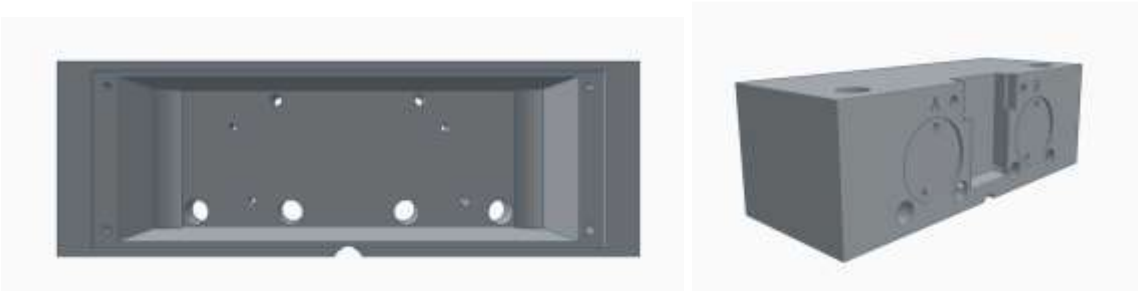
Arduino pin	DB15 connector	Sensor pin	Type of sensor
21	1	SCL	Flow sensor
Not used	2	Not used	Not used
Not used	3	Not used	Not used
Analog 3	4	SIG	Temperature sensor 4
3V	5	VDD	Flow sensor
Digital 13	6		
GND	7	GND	Temperature sensor 4, flow sensor
20	8	SDA	Flow sensor
Not used	9	Not used	Not used
GND	10	GND	Temperature sensor 1-2-3
5V	11	UCC	Temperature sensor 1-2-3-4
Analog 2	12	SIG	Temperature sensor 3
Analog 1	13	SIG	Temperature sensor 2
Analog 0	14	SIG	Temperature sensor 1
Not used	15	Not used	Not used

Supplementary Table 18: Wire connections of Relay shield and Arduino Mega Board.

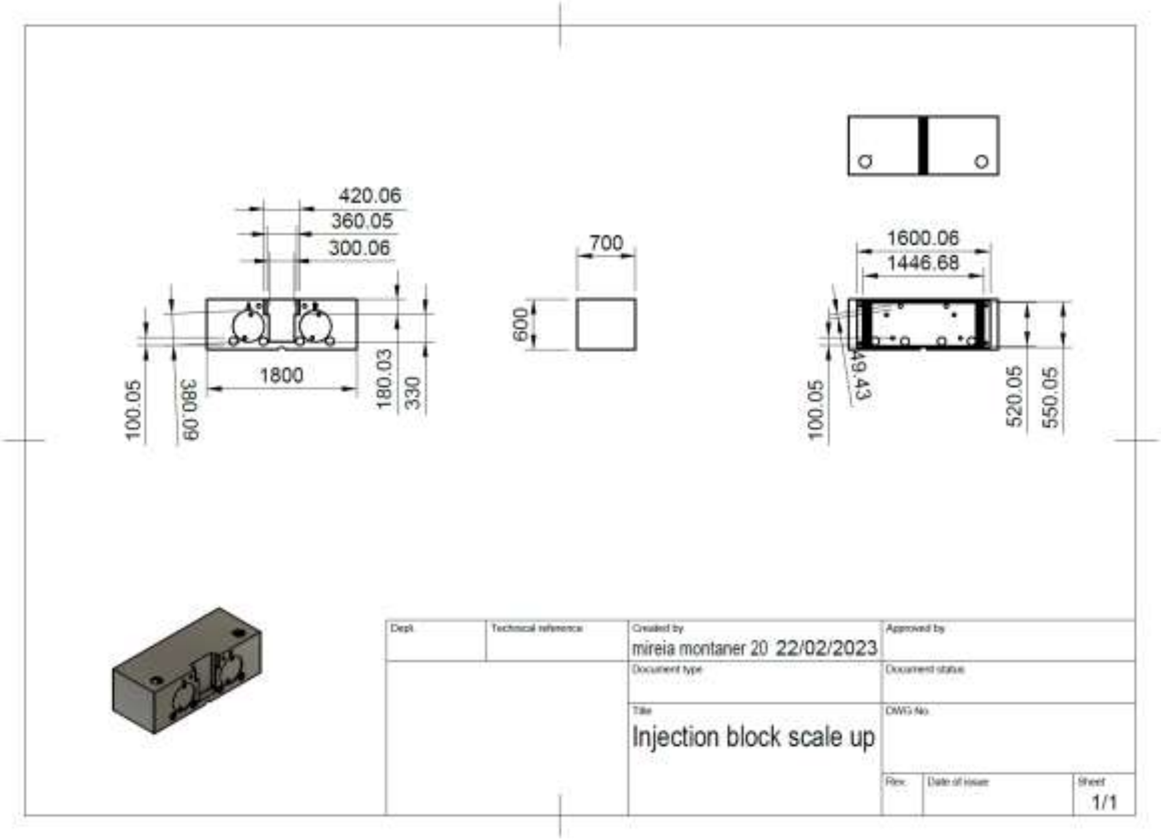
S.6.3. Tinkercad design and assembly of the 3D printed and electronic components scale up Injection block

The design of the Scale up injection block was designed to fit in the two 2 mL loops and two L-type valves to automate the injection of solvent or reactant. Once the design of the components was completed in Tinkercad, they were exported individually as an STL (Standard Tessellation Language) file and uploaded to Cura software. The printing settings used to 3D print the following component are shown in the tables from section 7.1. Finally, the models were sliced and USB-connected to the printer. After the components were 3D printed, the electronics and wires were assembled.

S.6.3.1 Tinkercad design of scale up Digital Injection Block



Supplementary Figure 60: Tinkercad design illustrating the scale up Injection block version.



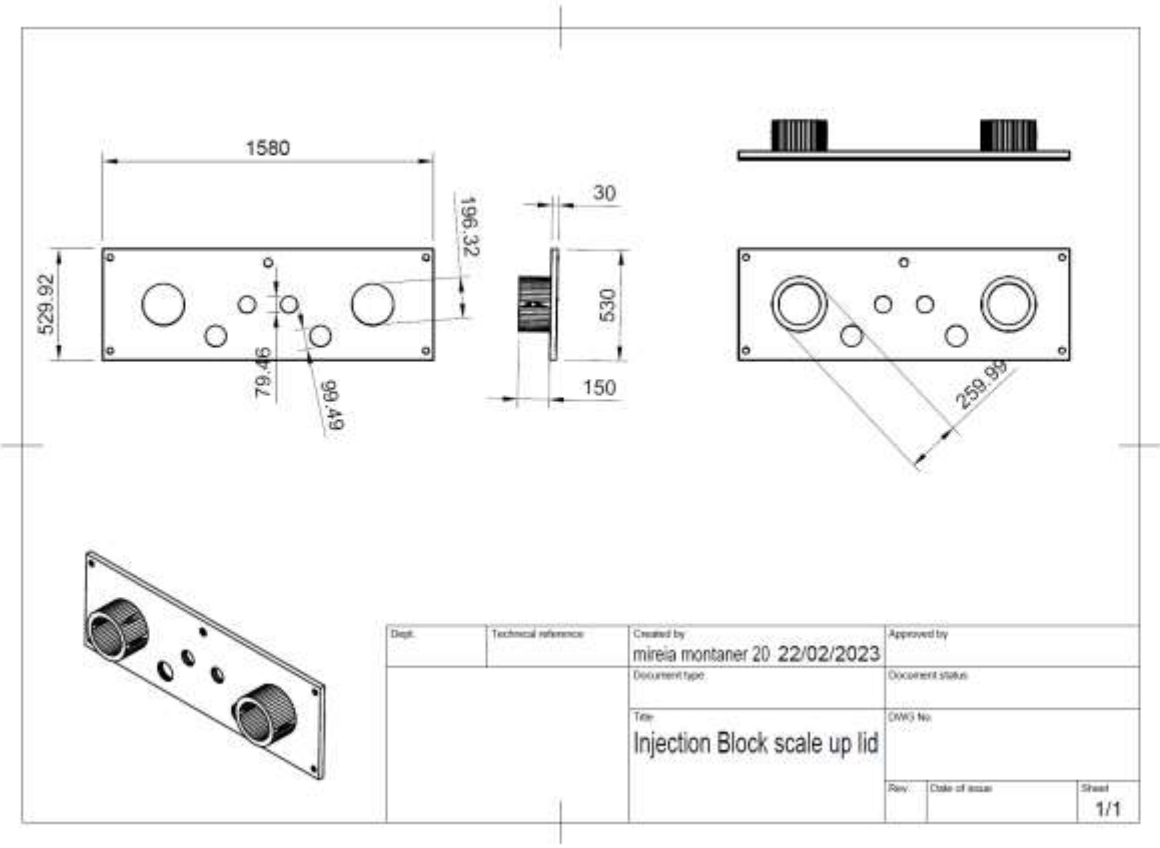
Supplementary Figure 61: Technical Drawing Showing the Dimensions of the Injection block.

Material	PLA
Time	18 h 35 min
Quantity	166 g – 21.10 m
Profile	Normal 0.15 mm
Support required	No

Supplementary Table 19: Ultimaker Cura Settings for 3D printing the injection block for the scale up version.

S.6.3.2 Tinkercad design of scale up Digital Injection Block

Supplementary Figure 62: Tinkercad design illustrating the scale up Injection block lid.

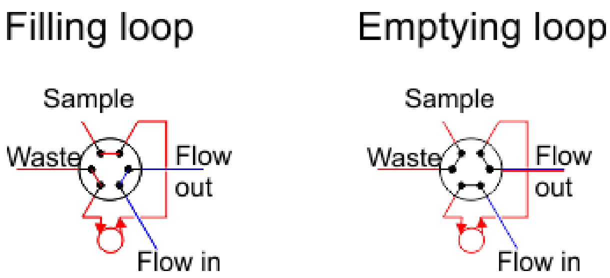


Supplementary Figure 63: Technical Drawing Showing the Dimensions of the Injection block lid.

Material	PLA
Time	4 h 14 min
Quantity	30 g – 3.77 m
Profile	Normal 0.15 mm
Support required	No

Supplementary Table 20: Ultimaker Cura Settings for 3D printing the lid of injection block for the scale up version.

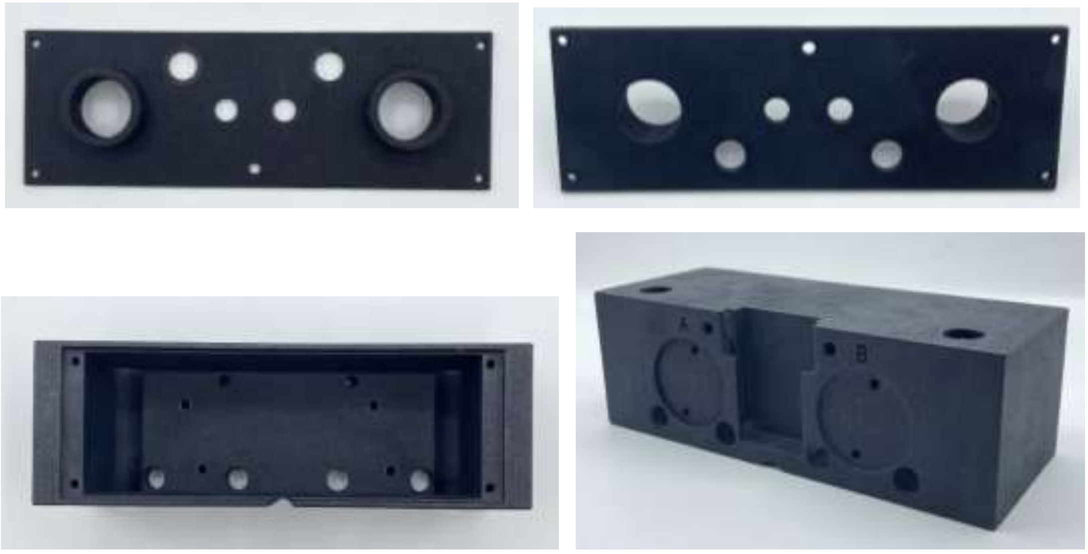
The following diagram illustrates the 6 ports from the valves which consist of: sample, waste flow in and flow out and (3). The valve features six ports arranged evenly around its circumference. Each port is linked to either the adjacent clockwise or anticlockwise port. By manipulating the valve, the neighboring port connection can be altered. In the left image, the sample is being charged into the loop, while in the right image, it can be redirected into the flow.



Supplementary Figure 64: 6 way loop valve.

S.6.3.3 Assembly of the 3D printed and electronic components to the scale up Digital Injection Block

To assemble the injection block for the scale up version, the 3D printed components must first be printed and ready for assembly (**Sup. Fig. 65**).



Supplementary Figure 65: 3D printed components of the Scale up injection block.

The Figure below illustrates the components required to set up the module (**Sup. Fig. 66**).

Supplementary Figure 66: Components of the Injection block.

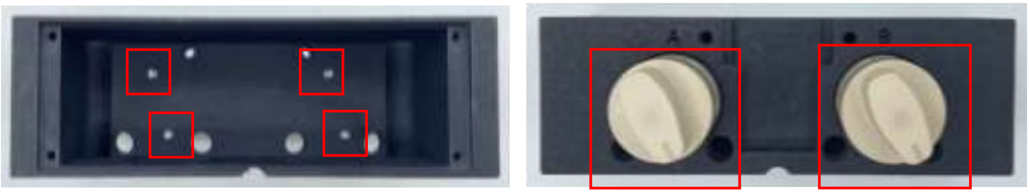
Step 1: installation of the union body, lock washer and nut to the injection block lid

Supplementary Figure 67: installation of union body, lock washer and nut to the injection block lid.

Step 2: connect the two L-type valve to the injection block lid

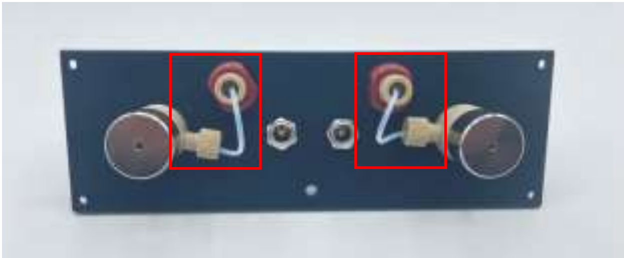
Supplementary Figure 68: installation of two L-type valves.

Step 3: secure the two 6-port loop injection valves to the injection block with two 934 M3 screws for each valve.



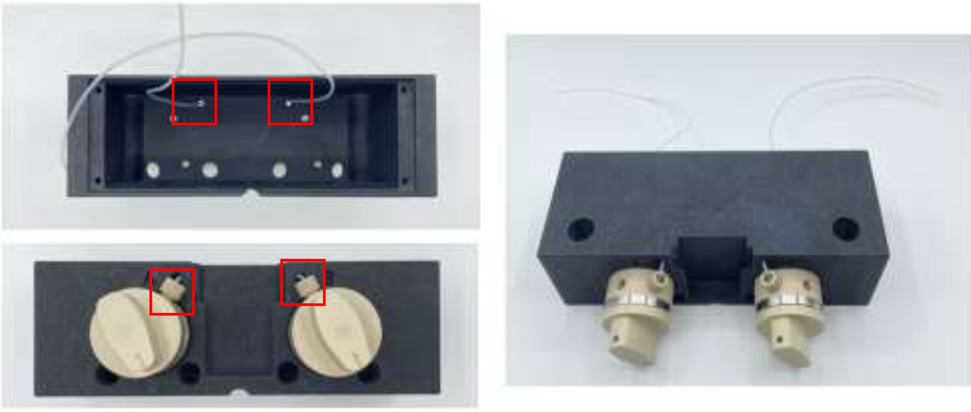
Supplementary Figure 69: installation of two 6-port loop injection valves.

Step 4: connect two small tubes, one for each valve, from the middle port to the union body



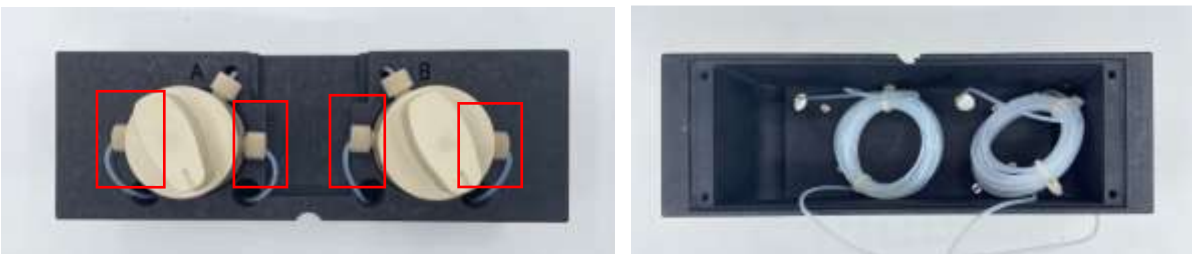
Supplementary Figure 70: installation of tubbing from the L-type valve to the union body.

Step 5: a tubing is connected to the waste port of the valve.



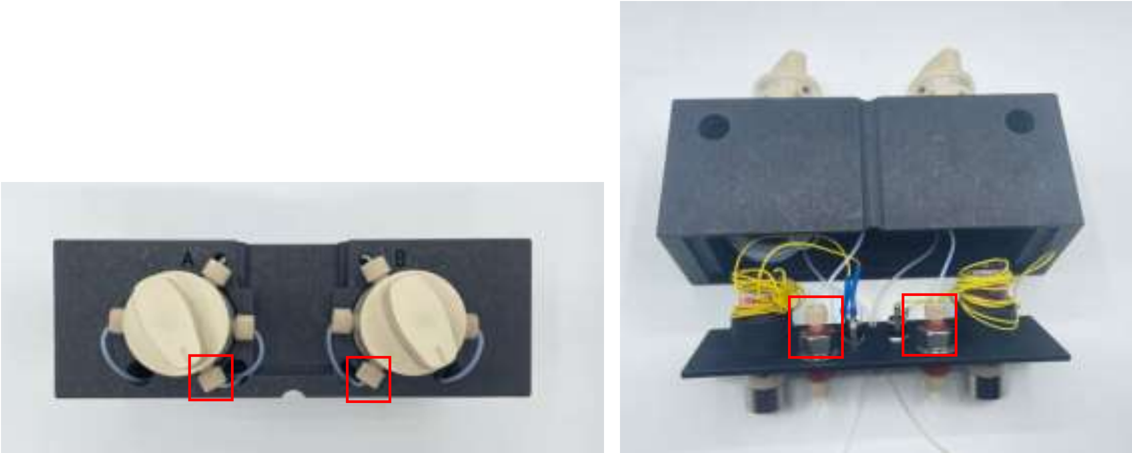
Supplementary Figure 71: installation of the waste tube.

Step 6: the two 2mL tubing loops are connected to the 6 way look valve.



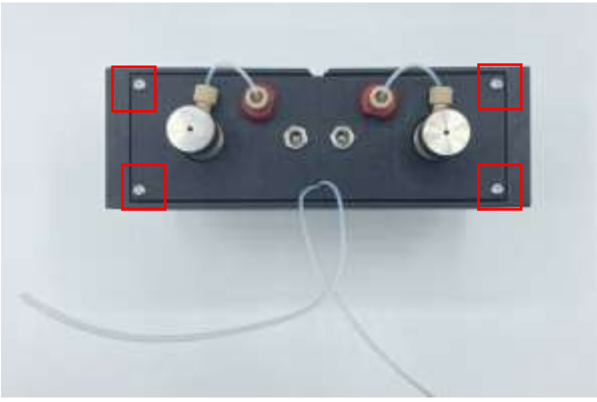
Supplementary Figure 72: installation of two 2mL loop tubing.

Step 7: connection of the sample tubing from the 6 port valve in the injection block to the lid



Supplementary Figure 73: installation of the Relay shield.

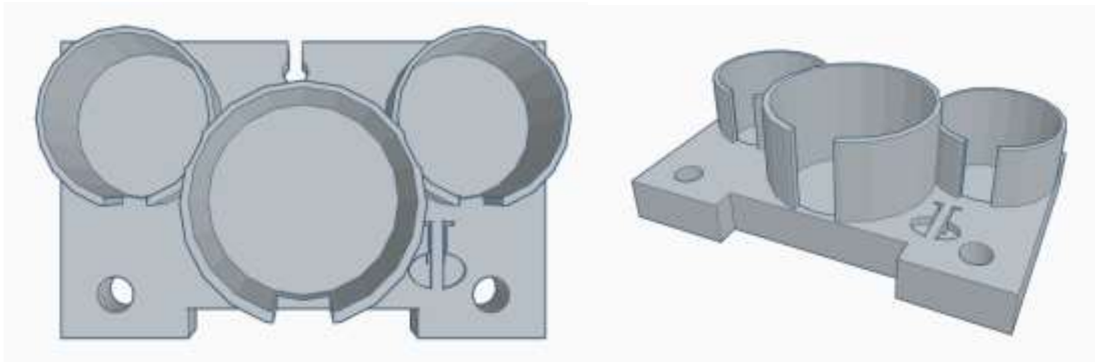
Step 8: secure the lid with two M3*8 and two M3*10 screws



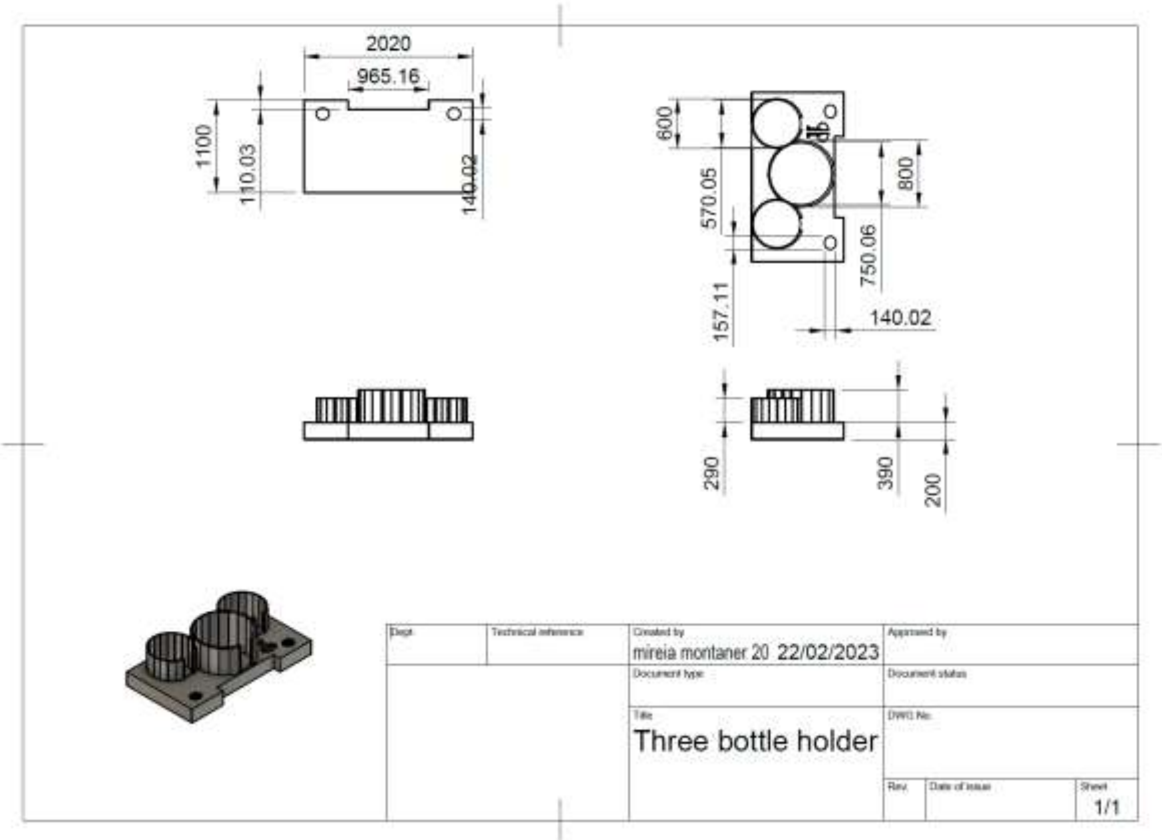
Supplementary Figure 74: lid secured to the injection block.

S.6.4 Design of the 3D printed scale up bottle holder

Once the design of the components was completed in Tinkercad, they were exported individually as an STL (Standard Tessellation Language) file and uploaded to Cura software. The printing settings used to 3D print the following component are shown in section 7.1. Finally, the models were sliced and USB-connected to the printer.



Supplementary Figure 75: Tinkercad design illustrating the scale up bottle holder.

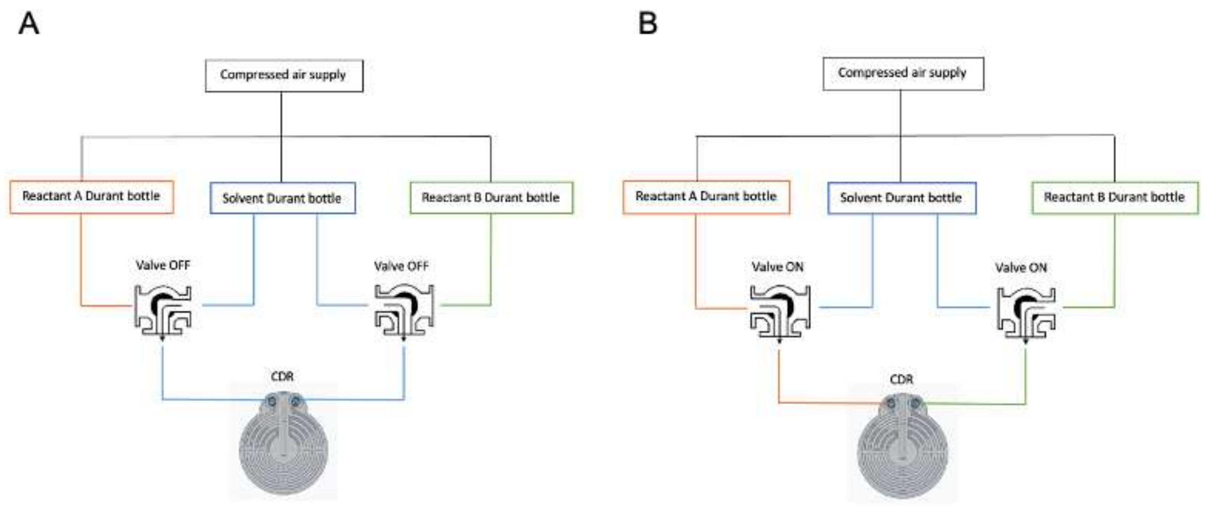


Supplementary Figure 76: Technical Drawing Showing the Dimensions of the bottle holder for the scale up system.

Material	PLA
Time	20 h 35 min
Quantity	174 g – 21.92 m
Profile	Normal 0.15 mm
Support required	No

Supplementary Table 21: Ultimaker Cura Settings for 3D the bottle holder for the scale up version.

The Duran pressure bottles are connected to two three-way L-type solenoid valves powered by a 24V power supply. The two valves are located in the back end of the injection block and regulates which fluid flows into the mixing chip. The L-type valve has two 90 degrees positions that connect one side port and the centre (**Sup. Fig. 77**). The valves opening and closing are controlled by the MegunoLink platform interface. When the valve is closed, solvent flows, and when it is open, reactant flows.



Supplementary Figure 77: Three-way valves operation A) Valve closed. B) Valve opened.

S.6.5 Assembly of Scale up flow system

Supplementary Figure 78: Assembly step by step of the different blocks.

S.6.6 Bill of Materials

Component Number	Designator	Component	Units	Cost per unit – (GPB - £)	Total cost -	Source of materials
1	3D printed structure	Base block	1	£24.00	£24.00	3dgbire
2	3D printed structure	Base block lid	1	£2.95	£2.95	3dgbire
3	3D printed structure	Base block supports	2	£1.8	£3.6	3dgbire
4	3D printed structure	Injection block	1	£8.97	£8.97	3dgbire
5	3D printed structure	Injection block lid	1	£1.62	£1.62	3dgbire
6	3D printed structure	Bottle holder	1	£9.40	£9.40	3dgbire
7	Electronics	Keyestudio 4 channel Relay shield	1	£19.19	£19.19	Amazon
8	Electronics	Arduino Mega	1	£48.37	£48.37	Amazon

9	Electronic	Male 12v DC Power Jack Adapter Connector	1	£0.43	£0.43	Amazon
10	Electronic	DB 15 pin connector female	1	£8.99	£8.99	Amazon
11	Electronic	F-M Dupont Wires	7	£0.05	£0.35	Amazon
12	Electronic	F-F Dupont wire	4	£0.05	£0.2	Amazon
13	Electronic	L-type valve	2	£219.88	£439.76	Kinesis
14	Electronic	Diba Omnifit® Loop Injection Valve, 6-port	2	£288.86	£577.72	Cole Palmer
15	Fastener	M3*4	1	£0.02	£0.02	Ali express
16	Fastener	M3*8	6	£0.02	£0.06	Ali express
17	Fastener	M3*10	4	£0.02	£0.08	Ali express
18	Fastener	934 M3	4	£0.02	£0.08	Ali express
19	Fastener	Union body	2	£20.78	£41.56	Cole-Palmer
20	Fastener	Lock washer				
21	Fastener	Nut				

Total cost:	**£1187.35**

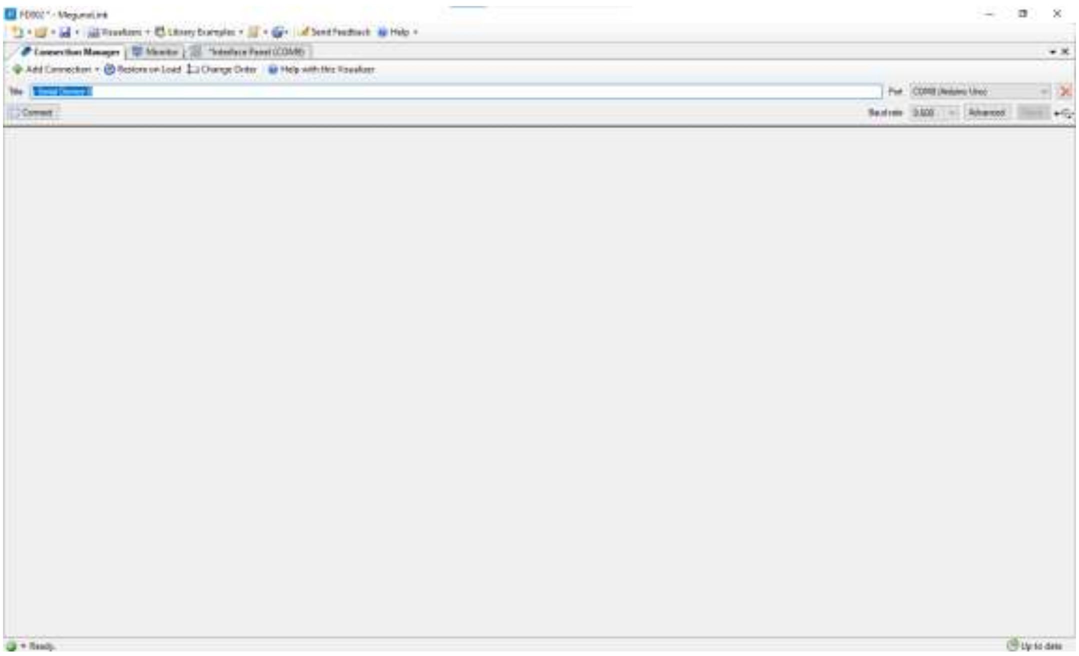
Supplementary Table 22: Bill of Materials for the 3 Temperature Sensor Module.

S.6.7 Meguno Link Interface Platform

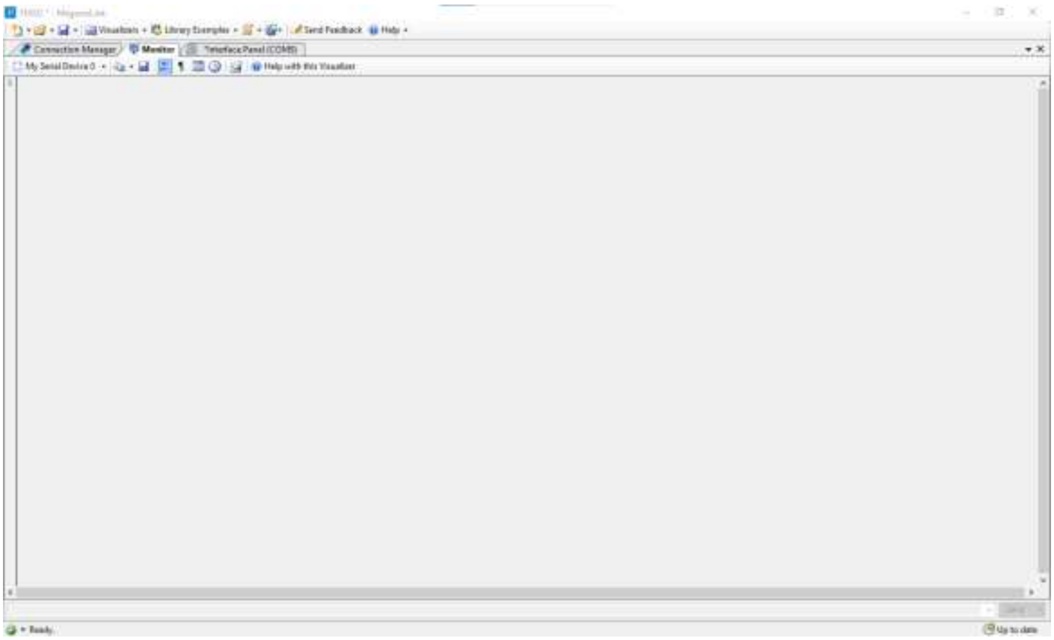
The interface platform that we used to communicate between the Arduino microcontroller and the PC is MegunoLink. This platform can read the data from the sensors and display it in a human readable way. The platform software that we designed contains 3 tabs:

- Connection Manager (**Sup. Fig. 79**): the platform connects to the Arduino through this tab.
- Monitor (**Sup. Fig. 80**)
- Interface panel:
  - Welcome/Splash screen (**Sup. Fig. 81A**)

- System configuration (**Sup. Fig. 81B**): determines how the flow system is set up, the solvent used, amount and type of reactors and the collection variables.
- System control (**Sup. Fig. 81C**): determines the reaction variables. These influence the overall duration of the reaction and set how long the valves must be open to allow reactants to flow into the reaction path.
- Reaction monitoring (**Sup. Fig. 81D**): when the Three Temperature module is connected, the measurements from the sensor will be displayed in this tab.
- Flow monitoring (**Sup. Fig. 81E**): when the Flow rate module is connected, the measurements from the sensor will be displayed in this tab.
- Photoflow monitoring (**Sup. Fig. 81F**): when the PhotoTemperature module is connected, the measurements from the sensor will be displayed in this tab.



Supplementary Figure 79: Connection manager tab.



Supplementary Figure 80: Monitor tab.

Supplementary Figure 81: MegunoLink Interface of the project. A) Welcome screen tab. B) System configuration tab. C) System control tab. D) Temperature monitoring tab. E) Phototemperature monitoring tab. F) Flow rate monitoring tab.

The following equation was used to determine the amount of time the valve must be open for the reactant to flow into path in order to automate the scale up injection system.

$$Flow\ rate\ \left(\frac{volume\ in\ ml}{time\ in\ min}\right) = a\ \times Pressure\ (PSI) + b\ \rightarrow Time\ (min) = \frac{Volume\ (mL)}{a\ \times Pressure\ (PSI) + b}$$

Equation 1: Flow rate and pressure linear equation where a and b are constant numbers.

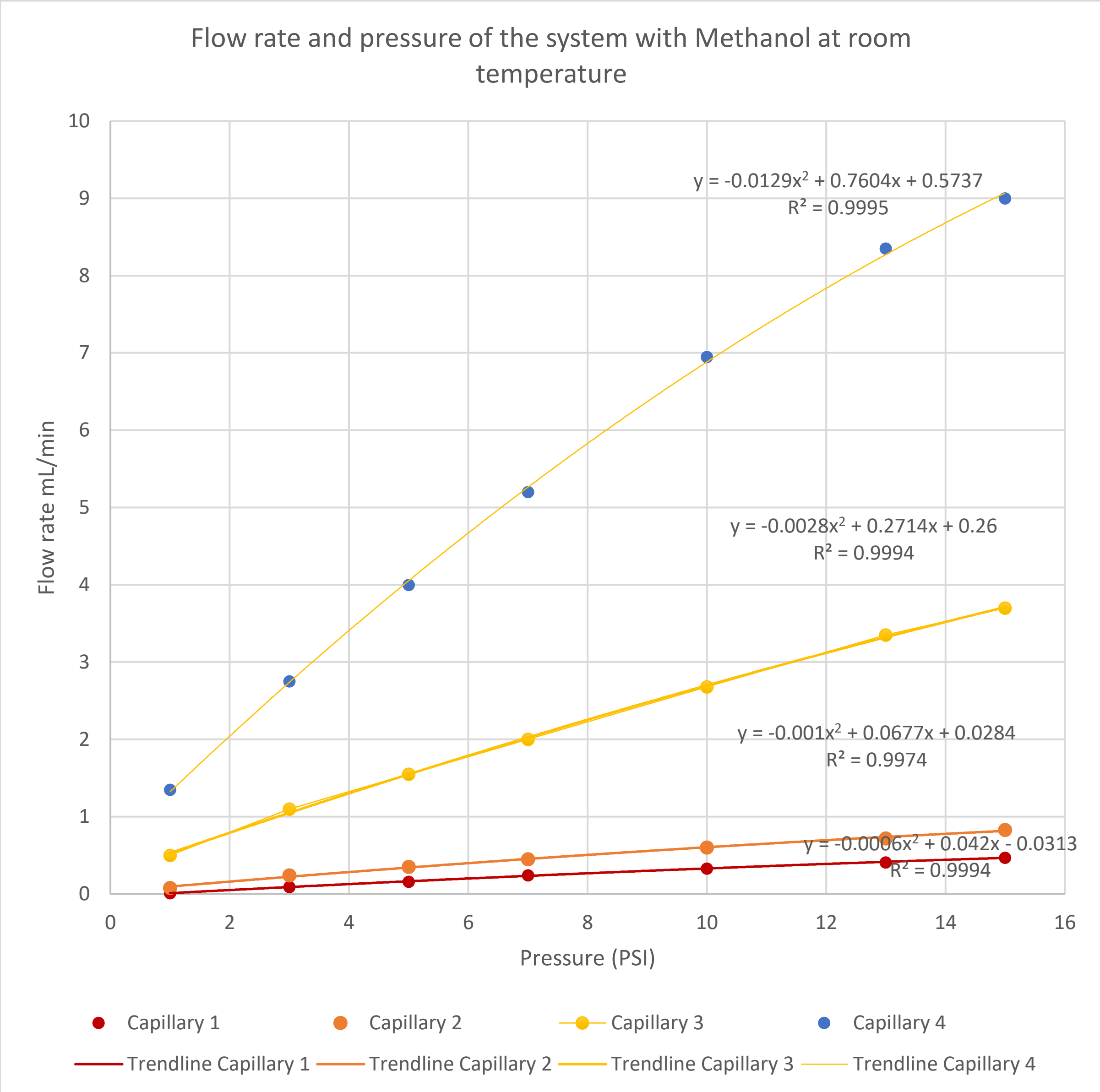
All solvent's flow rate and pressure correlations were measured, resulting in different equations for each solvent (Graph 3-8). Each solvent was allocated a number ranging from 1 to 6, thus when a solvent is selected on the interface, the data sent to the Arduino corresponds to that number. This approach simplified the code and facilitatede the communication between the MegunoLink platform and the Arduino board. To calculate the overall time of the reaction (*TotalTime*), the volume corresponds to the *TotalVolume* + *InjectionVolume*. The *TotalVolume* is equal to the volume values of the CRD used, precollect, postcollect and system volumes inserted on the Configuration tab. The *InjectionVolume* value is entered on the Control tab and represents the amount of reactant to be injected into the flow path. Since the time value is calculated in minutes, and the code employs the time function in millis(), the time value must be multiplied by 60000 to convert minutes to milliseconds.

```
if (Solvent==1){
 TotalTime=(((TotalVolume+InjectionVolume)/((0.018*Pressure)+0.0215))*60000);
 TotalMillis=(StartMillis+TotalTime);
 Eflowrate=((0.0168*Pressure)+0.02);
 if (millis() - StartMillis >= ((InjectionVolume/((0.018*Pressure)+0.0215))*60000)) {
   digitalWrite(SolenoidAPIN,LOW);
   digitalWrite(SolenoidBPIN,LOW);
   ReactionRUN = false;
 }
```

However, when calculating the time for the valve to remain open, the volume corresponds only to the *InjectionVolume*. When this period of time has passed, the valve closes.

```
if (Solvent==1){
 TotalTime=(((TotalVolume+InjectionVolume)/((0.018*Pressure)+0.0215))*60000);
 TotalMillis=(StartMillis+TotalTime);
 Eflowrate=((0.0168*Pressure)+0.02);
 if (millis() - StartMillis >= ((InjectionVolume/((0.018*Pressure)+0.0215))*60000)) {
   digitalWrite(SolenoidAPIN,LOW);
   digitalWrite(SolenoidBPIN,LOW);
   ReactionRUN = false;
 }
```

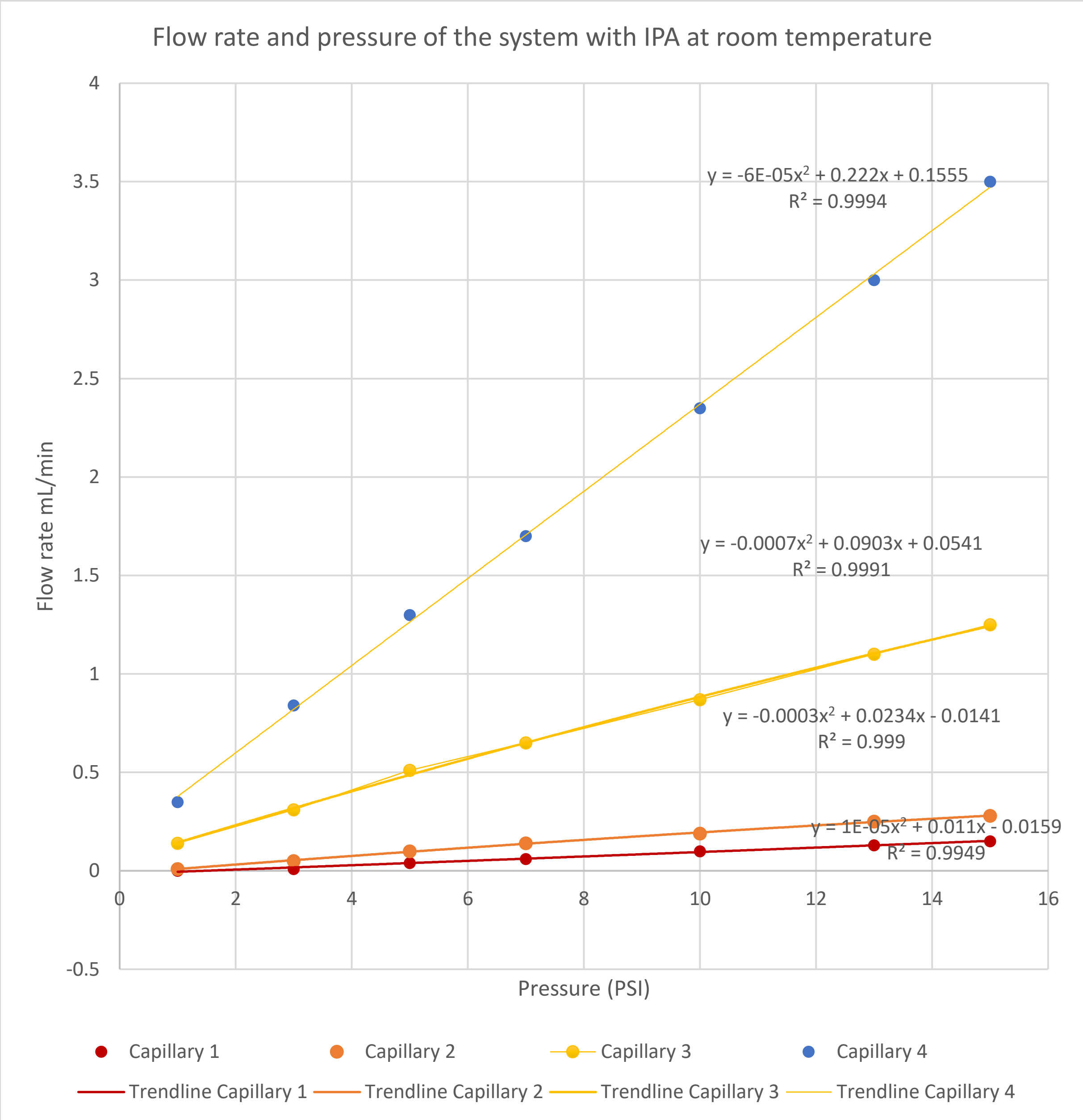
This part of the code corresponding to the reaction time control was constructed with *if* and *else if* statements. The first *if* statement is regarding the capillary used and inside each capillary statement, there is a second *if* statement. The second statement refers to the solvent used which will determine the *TotalTime* calculation. As previously stated, each solvent was assigned a number between 1 and 6, so when a solvent is selected in the System Control tab, the assigned number is sent to the Arduino code.



Supplementary Graph 1: Selected capillaries to measure the flow rate of Methanol at specific pressures.

Pressure	Capillary 1	Capillary 2	Capillary 3	Capillary 4
1	0.01	0.08	0.5	1.35
3	0.09	0.24	1.1	2.75
5	0.16	0.35	1.55	4
7	0.24	0.45	2	5.2
10	0.33	0.6	2.68	6.95
13	0.41	0.72	3.35	8.35
15	0.47	0.83	3.7	9

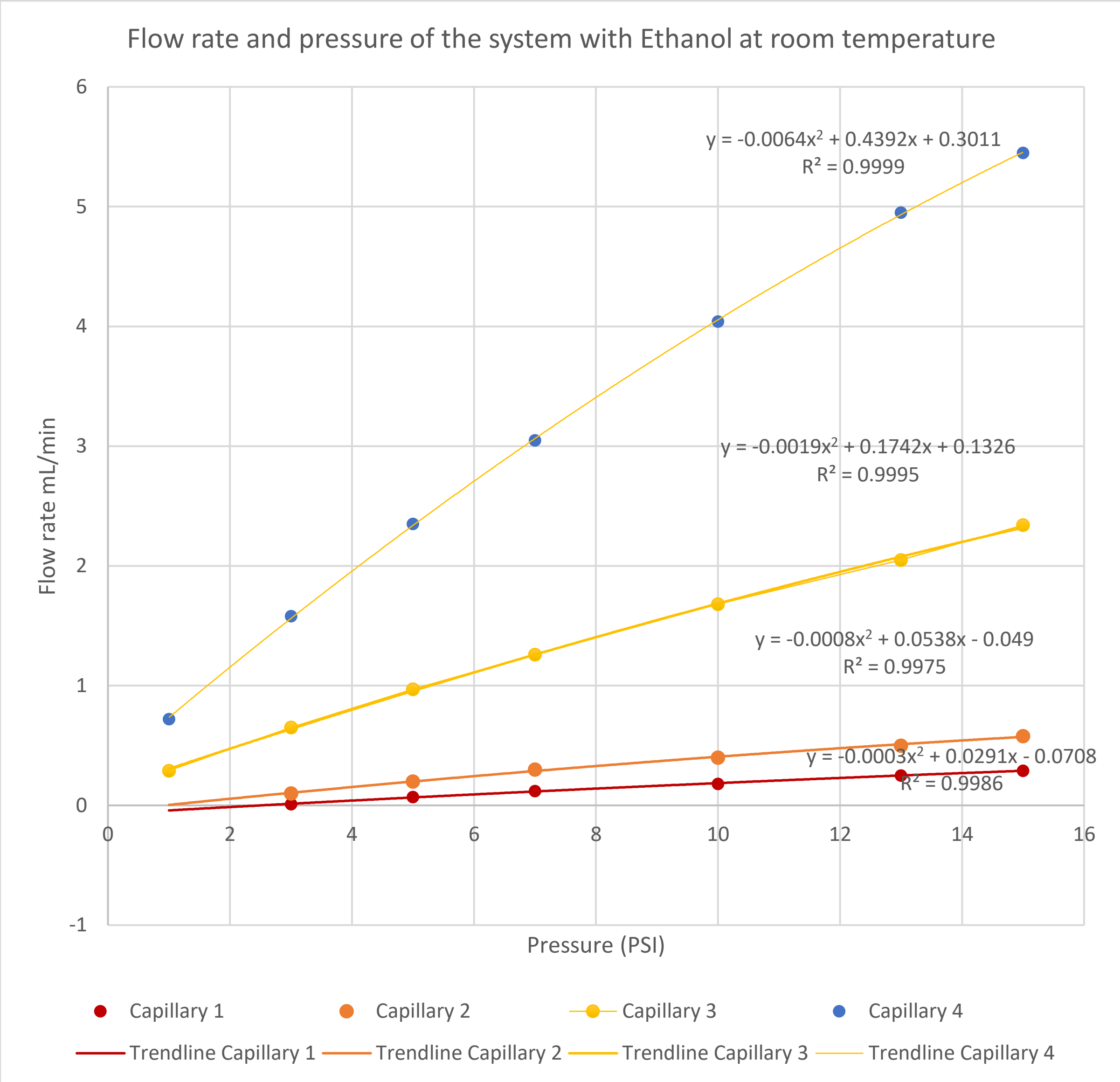
Supplementary Table 23: Flow rate of Methanol at different pressures and capillaries.



Supplementary Graph 2: Selected capillaries to measure the flow rate of IPA at specific pressures.

Pressure	Capillary 1	Capillary 2	Capillary 3	Capillary 4
1	0	0.01	0.14	0.35
3	0.01	0.05	0.31	0.84
5	0.04	0.1	0.51	1.3
7	0.06	0.14	0.65	1.7
10	0.1	0.19	0.87	2.35
13	0.13	0.25	1.1	3
15	0.15	0.28	1.25	3.5

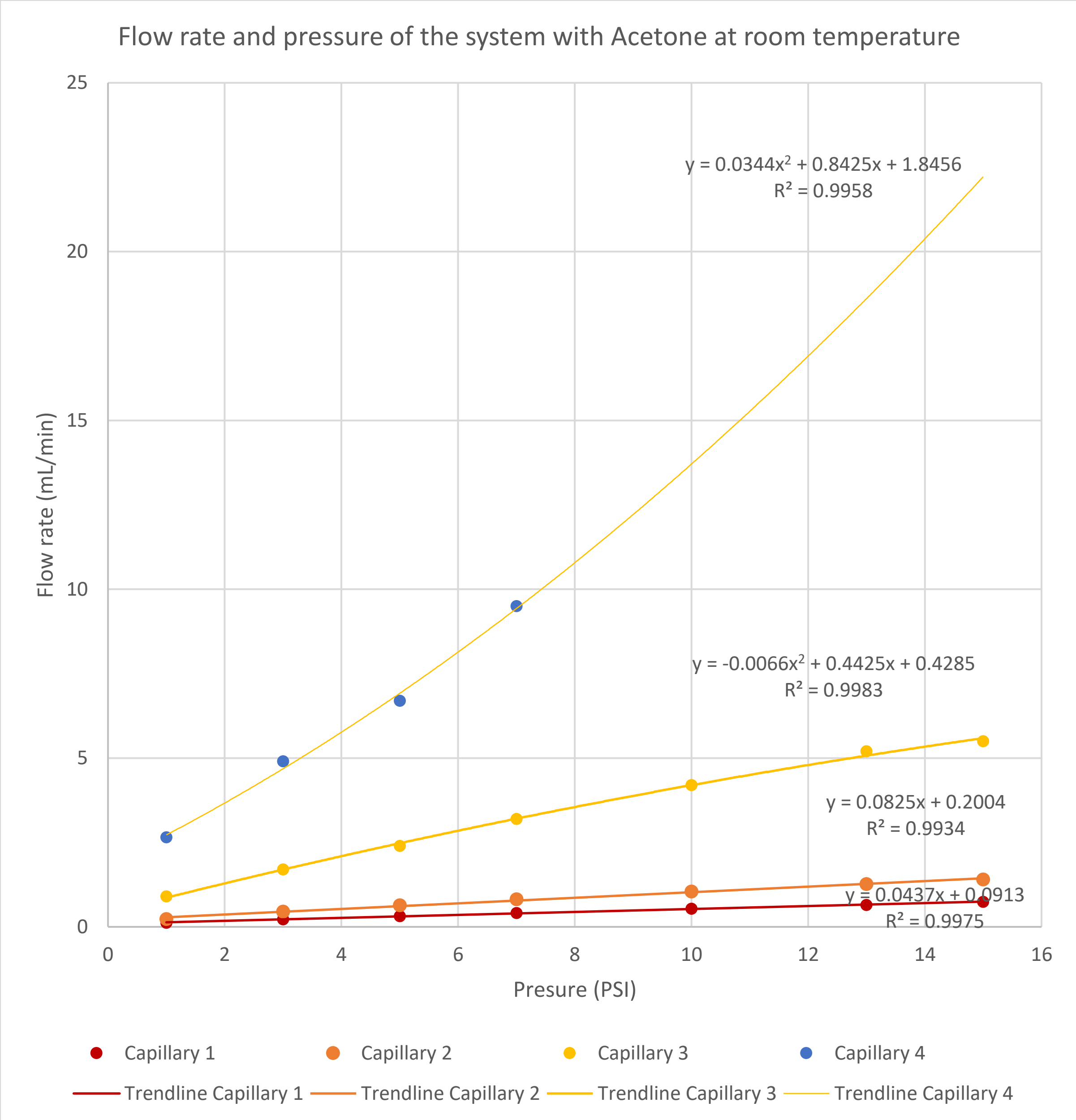
Supplementary Table 24: Flow rate of IPA at different pressures and capillaries.



Supplementary Graph 3: Selected capillaries to measure the flow rate of Ethanol at specific pressures.

Pressure	Capillary 1	Capillary 2	Capillary 3	Capillary 4
1			0.29	0.72
3	0.01	0.1	0.65	1.58
5	0.07	0.2	0.97	2.35
7	0.12	0.3	1.26	3.05
10	0.18	0.4	1.68	4.04
13	0.25	0.5	2.05	4.95
15	0.29	0.58	2.34	5.45

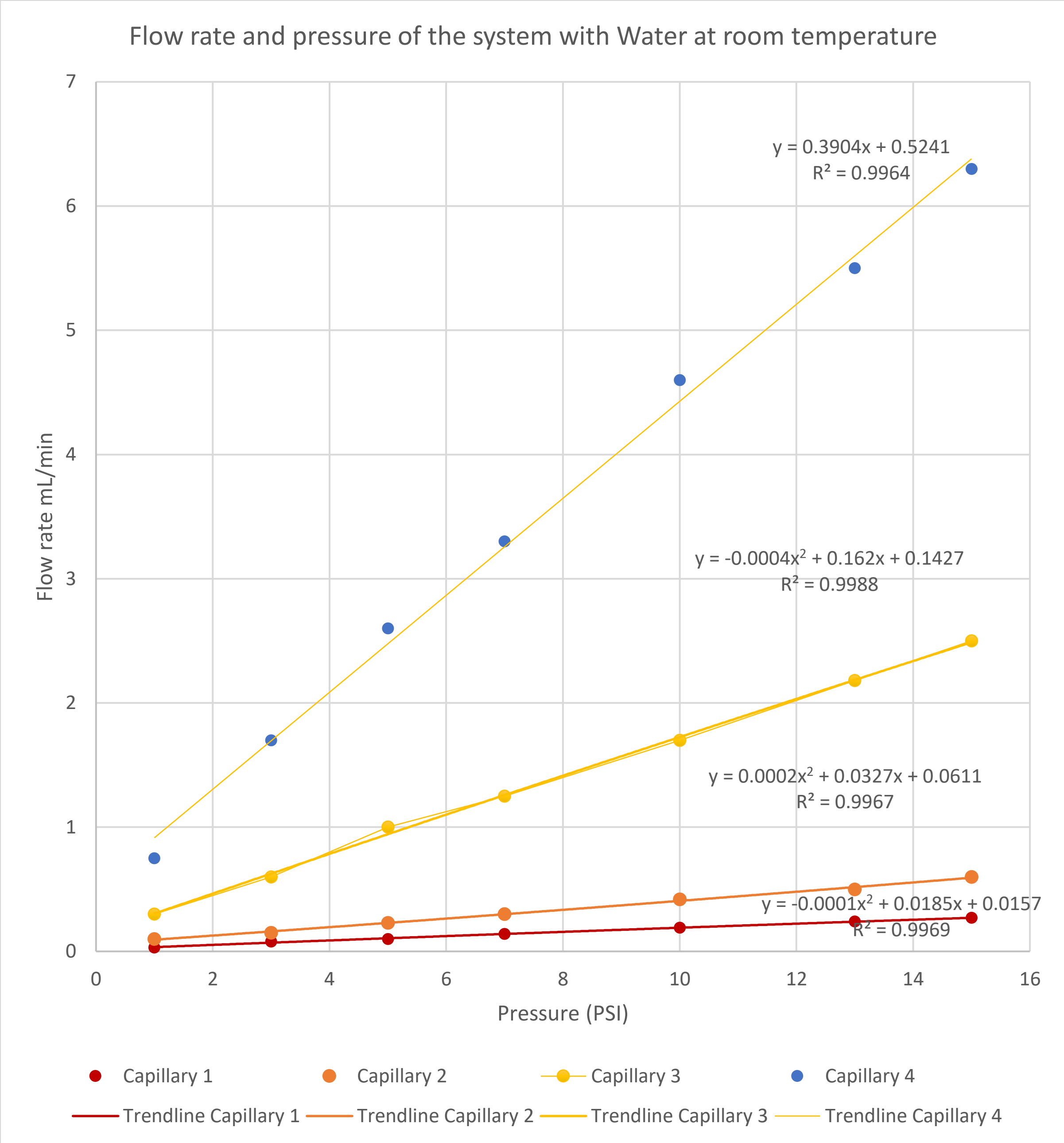
Supplementary Table 25: Flow rate of Ethanol at different pressures and capillaries.



Supplementary Graph 4: Selected capillaries to measure the flow rate of Acetone at specific pressures.

Pressure	Capillary 1	Capillary 2	Capillary 3	Capillary 4
1	0.12	0.23	0.9	2.65
3	0.22	0.45	1.7	4.9
5	0.32	0.64	2.4	6.7
7	0.41	0.82	3.2	9.5
10	0.54	1.05	4.2	-
13	0.65	1.27	5.2	-
15	0.74	1.4	5.5	-

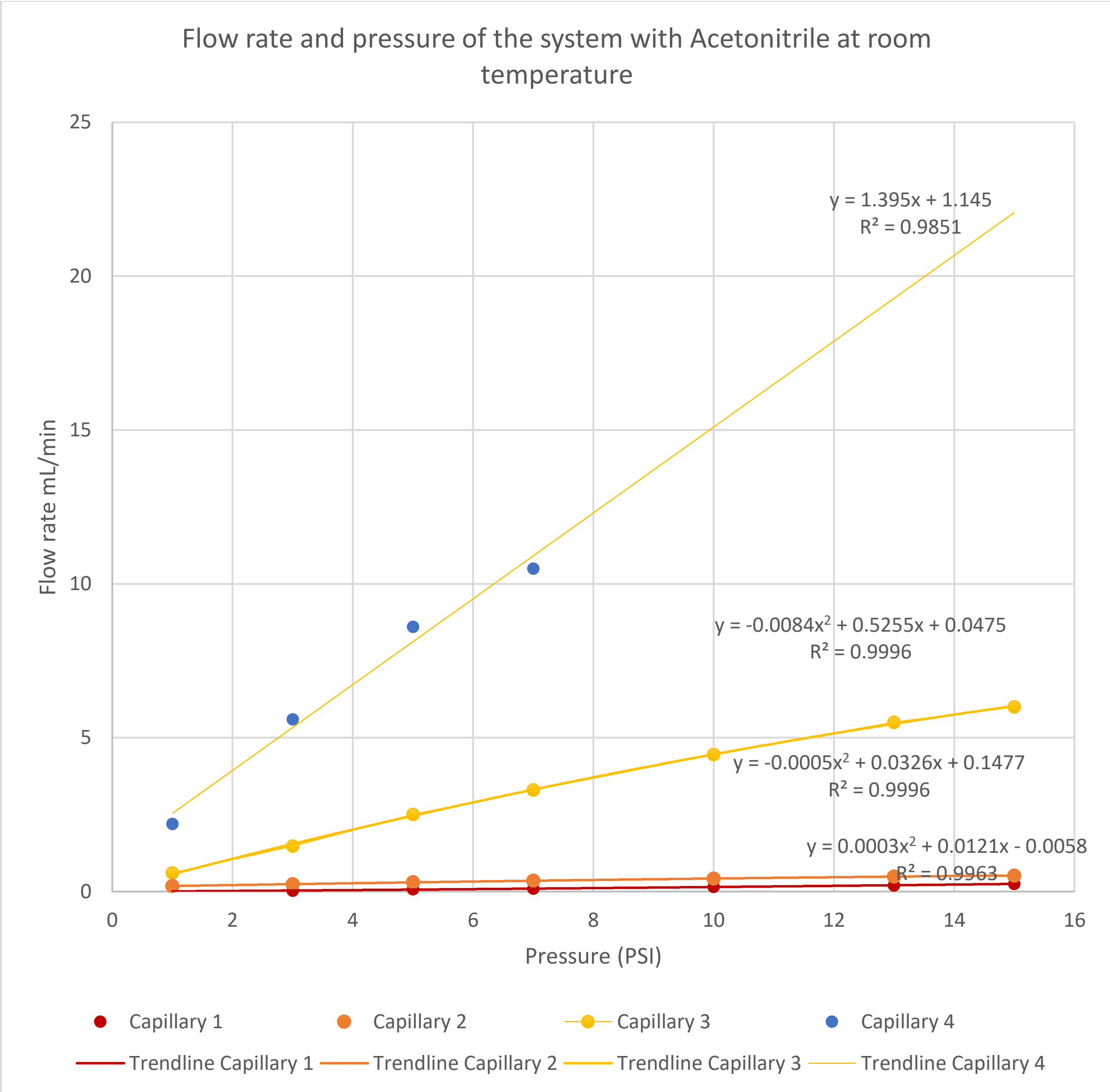
Supplementary Table 26: Flow rate of Acetone at different pressures and capillaries.



Supplementary Graph 5: Selected capillaries to measure the flow rate of Water at specific pressures.

Pressure	Capillary 1	Capillary 2	Capillary 3	Capillary 4
1	0.03	0.1	0.3	0.75
3	0.08	0.15	0.6	1.7
5	0.1	0.23	1	2.6
7	0.14	0.3	1.25	3.3
10	0.19	0.42	1.7	4.6
13	0.24	0.5	2.18	5.5
15	0.27	0.6	2.5	6.3

Supplementary Table 27: Flow rate of Water at different pressures and capillaries.



Supplementary Graph 6: Selected capillaries to measure the flow rate of Acetonitrile at specific pressures.

Pressure	Capillary 1	Capillary 2	Capillary 3	Capillary 4
1		0.18	0.6	2.2
3	0.03	0.24	1.48	5.6
5	0.07	0.3	2.5	8.6
7	0.09	0.35	3.3	10.5
10	0.15	0.42	4.45	-
13	0.2	0.49	5.5	-
15	0.25	0.52	6	-

Supplementary Table 28: Flow rate of Acetonitrile at different pressures and capillaries.

S.6.8 Arduino Codes

The following code controls the three different modules and sends the measurements to the Meguno Link Interface Platform.

```
#include "MegunoLink.h"
#include "CommandHandler.h"
#include <Wire.h>

#define SolenoidAPIN 4
#define SolenoidBPIN 5
#define LEDPIN 13
#define beta 3950
#define resistance 10

int ThermistorPin = 0;
int Vo;
float R1 = 2252;
float logR2, R2, T;
float A = 1.484778004e-03, B =
2.348962910e-04, C = 1.006037158e-07;
unsigned long StartMillis = 0;
float TotalMillis;
float Pressure;
float Capilliary;
float InjectionVolume;
float Solvent;
float CDRType1;
float CDRType3;
float LOOPA;
float LOOPB;
float PreCollect;
float PostCollect;
float FinalWash;
float SystemVolume;
float TotalVolume;
float TotalTime;
float Progress;
float Eflowrate;
boolean ReactionRUN = false;
boolean TemperatureStart = false;
boolean TemperatureStop;
boolean FlowStart = false;
boolean FlowStop;
boolean PhotoStart = false;
boolean PhotoStop;
const int ADDRESS = 0x08;
const float SCALE_FACTOR_FLOW =
500.0;
uint16_t sensor_flow_value;
CommandHandler<>
SerialCommandHandler;
long LastSent;

const unsigned SendInterval = 200;

XYPlot TempPlot("Temperature Readings"),
FlowPlot ("Flow rate readings"),
PhotoPlot("Photochemical and temperature
measurements");
InterfacePanel MyPanel;
void
Cmd_SolenoidAON(CommandParameter
&Parameters)
{ digitalWrite(SolenoidAPIN,HIGH);
digitalWrite(LEDPIN, HIGH); }

void
Cmd_SolenoidAOFF(CommandParameter
&Parameters)
{ digitalWrite(SolenoidAPIN,LOW);
digitalWrite(LEDPIN,LOW); }

void
Cmd_SolenoidBON(CommandParameter
&Parameters)
{ digitalWrite(SolenoidBPIN,HIGH); }

void
Cmd_SolenoidBOFF(CommandParameter
&Parameters)
{ digitalWrite(SolenoidBPIN,LOW);}

void
Cmd_ContinuousRUNON(CommandParam
eter &Parameters)
{ digitalWrite(SolenoidAPIN,HIGH);
 digitalWrite(SolenoidBPIN,HIGH);}
```

```
void
Cmd_ContinuousRUNOFF(CommandPara
meter &Parameters)
{ digitalWrite(SolenoidAPIN,LOW);
digitalWrite(SolenoidBPIN,LOW);

MyPanel.SetProgress(("ReactionProgressB
ar"), 0); }

void
Cmd_ReactionRUN(CommandParameter
&Parameters)
{Pressure=Parameters.NextParameterAsInt
eger(Pressure);
InjectionVolume=Parameters.NextParamete
rAsInteger(InjectionVolume);
Capilliary=Parameters.NextParameterAsInte
ger(Capilliary);
digitalWrite(SolenoidAPIN,HIGH);
digitalWrite(SolenoidBPIN, HIGH);
ReactionRUN = true;
StartMillis = millis();
MyPanel.SetText(("TotalVolume"),(TotalVolu
me+InjectionVolume));
MyPanel.SetProgress(("ReactionProgressB
ar"), 0);
MyPanel.SetText(F("Progress"), ((millis()-
StartMillis)/TotalMillis)*100);
MyPanel.SetText(F("TotalTime"),
(TotalTime));
MyPanel.SetText(F("Eflowrate"),
(Eflowrate));
}

void
Cmd_SETSYSTEM(CommandParameter
&Parameters)
{CDRType1=Parameters.NextParameterAsI
nteger(CDRType1);
CDRType3=Parameters.NextParameterAsIn
teger(CDRType3);
LOOPA=Parameters.NextParameterAsInteg
er(LOOPA);
LOOPB=Parameters.NextParameterAsInteg
er(LOOPB);
PreCollect=Parameters.NextParameterAsInt
eger(PreCollect);
PostCollect=Parameters.NextParameterAsI
nteger(PostCollect);
FinalWash=Parameters.NextParameterAsInt
eger(FinalWash);
SystemVolume=Parameters.NextParameter
AsInteger(SystemVolume);
Solvent=Parameters.NextParameterAsInteg
er(Solvent);

if (CDRType1>=1){
TotalVolume=((CDRType1*3)+(PreCollect)+
(PostCollect)+(SystemVolume)); }

else if (CDRType3>=1){
TotalVolume=((CDRType3*2.8)+(PreCollect
)+(PostCollect)+(SystemVolume)); }
}

void
Cmd_TemperatureStart(CommandParamet
er &params){
 TemperatureStart = true;   }

void
Cmd_TemperatureStop(CommandParamete
r &params){
 TemperatureStart = false;   }

void Cmd_FlowStart(CommandParameter
&params){
 FlowStart = true;   }

void Cmd_FlowStop(CommandParameter
&params){
 FlowStart = false;   }

void Cmd_PhotoStart(CommandParameter
&params){
 PhotoStart = true;   }

void Cmd_PhotoStop(CommandParameter
&params){
 PhotoStart = false;   }


void setup(){

Serial.begin(9600);
Wire.begin();

Serial.println("MegunoLink Pro - Turning
Solenoids on and off");
```

```
Serial.println("-----------------------------");
SerialCommandHandler.AddCommand(F("SolenoidAON"), Cmd_SolenoidAON);
SerialCommandHandler.AddCommand(F("SolenoidAOFF"), Cmd_SolenoidAOFF);
SerialCommandHandler.AddCommand(F("SolenoidBON"), Cmd_SolenoidBON);
SerialCommandHandler.AddCommand(F("SolenoidBOFF"), Cmd_SolenoidBOFF);
SerialCommandHandler.AddCommand(F("ContinuousRUNON"),
Cmd_ContinuousRUNON);
SerialCommandHandler.AddCommand(F("ContinuousRUNOFF"),
Cmd_ContinuousRUNOFF);
SerialCommandHandler.AddCommand(F("SETSYSTEM"), Cmd_SETSYSTEM);
SerialCommandHandler.AddCommand(F("ReactionRUN"), Cmd_ReactionRUN);
SerialCommandHandler.AddCommand(F("TemperatureStart"), Cmd_TemperatureStart);
SerialCommandHandler.AddCommand(F("TemperatureStop"), Cmd_TemperatureStop);
SerialCommandHandler.AddCommand(F("FlowStart"), Cmd_FlowStart);
SerialCommandHandler.AddCommand(F("FlowStop"), Cmd_FlowStop);
SerialCommandHandler.AddCommand(F("PhotoStart"), Cmd_PhotoStart);
SerialCommandHandler.AddCommand(F("PhotoStop"), Cmd_PhotoStop);
pinMode(SolenoidAPIN,OUTPUT);
pinMode(SolenoidBPIN,OUTPUT);

LastSent = millis();

TempPlot.SetSeriesProperties("ADCValue1", Plot::Red, Plot::Solid, 2, Plot::Square);
TempPlot.SetSeriesProperties("ADCValue2", Plot::Blue, Plot::Solid, 2, Plot::Square);
TempPlot.SetSeriesProperties("ADCValue3", Plot::Green, Plot::Solid, 2, Plot::Square);
FlowPlot.SetSeriesProperties("Flow rate",
Plot::Magenta, Plot::Solid, 5, Plot::Circle);
PhotoPlot.SetSeriesProperties("ADCValue4", Plot::Black, Plot::Solid, 2, Plot::Triangle);

int ret;
do {
  Wire.beginTransmission(ADDRESS);
  Wire.write(0xFE);
  ret=Wire.endTransmission();
  } while (ret !=0);
}

void loop() {

SerialCommandHandler.Process();

if ((ReactionRUN == true)&&(Capilliary == 1)) {
 if (Solvent==1){
TotalTime=(((TotalVolume+InjectionVolume)/((0.018*Pressure)+0.0215))*60000);
  TotalMillis=(StartMillis+TotalTime);
  Eflowrate=((0.0168*Pressure)+0.02);
  if (millis() - StartMillis >=
((InjectionVolume/((0.018*Pressure)+0.0215))*60000)) {
    digitalWrite(SolenoidAPIN,LOW);
    digitalWrite(SolenoidBPIN,LOW);
    ReactionRUN = false;  }
}

 else if (Solvent==2){
TotalTime=(((TotalVolume+InjectionVolume)/((0.0407*Pressure)+0.085))*60000);
  TotalMillis=(StartMillis+TotalTime);
  Eflowrate=((0.0437*Pressure)+0.0913);
  if (millis() - StartMillis >=
((InjectionVolume/((0.0407*Pressure)+0.085))*60000)) {
    digitalWrite(SolenoidAPIN,LOW);
    digitalWrite(SolenoidBPIN,LOW);
    ReactionRUN = false;  }
}

  else if (Solvent==3){
TotalTime=(((TotalVolume+InjectionVolume)/((0.0291*Pressure)-0.005))*60000);
  TotalMillis=(StartMillis+TotalTime);
  Eflowrate=((0.0325*Pressure)-0.0065);
  if (millis() - StartMillis >=
((InjectionVolume/((0.0291*Pressure)-0.005))*60000)) {
    digitalWrite(SolenoidAPIN,LOW);
    digitalWrite(SolenoidBPIN,LOW);
    ReactionRUN = false;  }
}
```

```
  else if (Solvent==4){
TotalTime=(((TotalVolume+InjectionVolume)
/((0.02156*Pressure)-0.04609))*60000);
  TotalMillis=(StartMillis+TotalTime);
  Eflowrate=((0.0229*Pressure)-0.049);
  if (millis() - StartMillis >=
((InjectionVolume/((0.02156*Pressure)-
0.04609))*60000)) {
    digitalWrite(SolenoidAPIN,LOW);
    digitalWrite(SolenoidBPIN,LOW);
    ReactionRUN = false;  }
}
  else if (Solvent==5){
TotalTime=(((TotalVolume+InjectionVolume)
/((0.0095*Pressure)-0.014))*60000);
  TotalMillis=(StartMillis+TotalTime);
  Eflowrate=((0.0112*Pressure)-0.0165);
  if (millis() - StartMillis >=
((InjectionVolume/((0.0095*Pressure)-
0.014))*60000)) {
    digitalWrite(SolenoidAPIN,LOW);
    digitalWrite(SolenoidBPIN,LOW);
    ReactionRUN = false;  }
}

  else if (Solvent==6){
TotalTime=(((TotalVolume+InjectionVolume)
/((0.01424*Pressure)-0.02072))*60000);
  TotalMillis=(StartMillis+TotalTime);
  Eflowrate=((0.0178*Pressure)-0.0259);
  if (millis() - StartMillis >=
((InjectionVolume/((0.01424*Pressure)-
0.02072))*60000)) {
    digitalWrite(SolenoidAPIN,LOW);
    digitalWrite(SolenoidBPIN,LOW);
    ReactionRUN = false;   }
}
MyPanel.SetText(F("Progress"), ((millis()-
StartMillis)/TotalTime)*100);
MyPanel.SetText(F("TotalTime"),
(TotalTime/60000));
MyPanel.SetText(F("Eflowrate"),
(Eflowrate));
}

else if ((ReactionRUN ==
true)&&(Capilliary== 2)){

 if (Solvent==1){
TotalTime=(((TotalVolume+InjectionVolume)
/((0.0375*Pressure)+0.0562))*60000);
  TotalMillis=(StartMillis+TotalTime);
  Eflowrate=((0.0357*Pressure)+0.0535);
  if (millis() - StartMillis >=
((InjectionVolume/((0.0375*Pressure)+0.056
2))*60000)) {
    digitalWrite(SolenoidAPIN,LOW);
    digitalWrite(SolenoidBPIN,LOW);
    ReactionRUN = false;  }
}
 else if (Solvent==2){
TotalTime=(((TotalVolume+InjectionVolume)
/((0.0825*Pressure)+0.2004))*60000);
  TotalMillis=(StartMillis+TotalTime);
  Eflowrate=((0.0825*Pressure)+0.2004);
  if (millis() - StartMillis >=
((InjectionVolume/((0.0825*Pressure)+0.200
4))*60000)) {
    digitalWrite(SolenoidAPIN,LOW);
    digitalWrite(SolenoidBPIN,LOW);
    ReactionRUN = false;  }
}

else if (Solvent == 3){
TotalTime=(((TotalVolume+InjectionVolume)
/((0.0513*Pressure)+0.0713))*60000);
  TotalMillis=(StartMillis+TotalTime);
  Eflowrate=((0.0513*Pressure)+0.0713);
  if (millis() - StartMillis >=
((InjectionVolume/((0.0513*Pressure)+0.071
3))*60000)) {
    digitalWrite(SolenoidAPIN,LOW);
    digitalWrite(SolenoidBPIN,LOW);
    ReactionRUN = false;  }
  }

  else if (Solvent==4){
TotalTime=(((TotalVolume+InjectionVolume)
/((0.03492*Pressure)+0.00324))*60000);
  TotalMillis=(StartMillis+TotalTime);
  Eflowrate=((0.0388*Pressure)+0.0036);
  if (millis() - StartMillis >=
((InjectionVolume/((0.03492*Pressure)+0.00
324))*60000)) {
    digitalWrite(SolenoidAPIN,LOW);
    digitalWrite(SolenoidBPIN,LOW);
    ReactionRUN = false;  }
}
```

```
  else if (Solvent==5){
TotalTime=(((TotalVolume+InjectionVolume)
/((0.01696*Pressure)-0.00308))*60000);
  TotalMillis=(StartMillis+TotalTime);
  Eflowrate=((0.0193*Pressure)-0.0035);
  if (millis() - StartMillis >=
((InjectionVolume/((0.01696*Pressure)-
0.00308))*60000)) {
    digitalWrite(SolenoidAPIN,LOW);
    digitalWrite(SolenoidBPIN,LOW);
    ReactionRUN = false;  }
  }

  else if (Solvent==6){
TotalTime=(((TotalVolume+InjectionVolume)
/((0.0244*Pressure)+0.1691))*60000);
  TotalMillis=(StartMillis+TotalTime);
  Eflowrate=((0.0244*Pressure)+0.1691);
  if (millis() - StartMillis >=
((InjectionVolume/((0.0244*Pressure)+0.169
1))*60000)) {
    digitalWrite(SolenoidAPIN,LOW);
    digitalWrite(SolenoidBPIN,LOW);
    ReactionRUN = false;  }
  }

MyPanel.SetText(F("Progress"), ((millis()-
StartMillis)/TotalTime)*100);
MyPanel.SetText(F("TotalTime"),
(TotalTime/60000));
MyPanel.SetText(F("Eflowrate"),
(Eflowrate));
}

else if ((ReactionRUN == true)&&(Capilliary
== 3)) {

 if (Solvent==1){
TotalTime=(((TotalVolume+InjectionVolume)
/((0.1559*Pressure)+0.1585))*60000);
  TotalMillis=(StartMillis+TotalTime);
  Eflowrate=((0.01559*Pressure)+0.1585);
  if (millis() - StartMillis >=
((InjectionVolume/((0.1559*Pressure)+0.158
5))*60000)) {
    digitalWrite(SolenoidAPIN,LOW);
    digitalWrite(SolenoidBPIN,LOW);
    ReactionRUN = false;  }
}

 else if (Solvent==2){
TotalTime=(((TotalVolume+InjectionVolume)
/((0.32*Pressure)+0.71))*60000);
  TotalMillis=(StartMillis+TotalTime);
  Eflowrate=((0.3364*Pressure)+0.7051);
  if (millis() - StartMillis >=
((InjectionVolume/((0.32*Pressure)+0.71))*6
0000)) {
    digitalWrite(SolenoidAPIN,LOW);
    digitalWrite(SolenoidBPIN,LOW);
    ReactionRUN = false; }
}

else if (Solvent==3){
TotalTime=(((TotalVolume+InjectionVolume)
/((0.238*Pressure)+0.395))*60000);
  TotalMillis=(StartMillis+TotalTime);
  Eflowrate=((0.2268*Pressure)+0.3761);
  if (millis() - StartMillis >=
((InjectionVolume/((0.238*Pressure)+0.395))
*60000)) {
    digitalWrite(SolenoidAPIN,LOW);
    digitalWrite(SolenoidBPIN,LOW);
    ReactionRUN = false; }
}

  else if (Solvent==4){
TotalTime=(((TotalVolume+InjectionVolume)
/((0.1332*Pressure)+0.1992))*60000);
  TotalMillis=(StartMillis+TotalTime);
  Eflowrate=((0.1436*Pressure)+0.2123);
  if (millis() - StartMillis >=
((InjectionVolume/((0.1332*Pressure)+0.199
2))*60000)) {
    digitalWrite(SolenoidAPIN,LOW);
    digitalWrite(SolenoidBPIN,LOW);
    ReactionRUN = false; }
}
  else if (Solvent==5){
TotalTime=(((TotalVolume+InjectionVolume)
/((0.221*Pressure)+0.158))*60000);
  TotalMillis=(StartMillis+TotalTime);
  Eflowrate=((0.0784*Pressure)+0.085);
  if (millis() - StartMillis >=
((InjectionVolume/((0.221*Pressure)+0.158))
*60000)) {
    digitalWrite(SolenoidAPIN,LOW);
    digitalWrite(SolenoidBPIN,LOW);
    ReactionRUN = false;   }
}
```

```
  else if (Solvent==6){
TotalTime=(((TotalVolume+InjectionVolume)
/((0.411*Pressure)+0.319))*60000);
  TotalMillis=(StartMillis+TotalTime);
  Eflowrate=((0.3891*Pressure)+0.4028);
  if (millis() - StartMillis >=
((InjectionVolume/((0.411*Pressure)+0.319))
*60000)) {
    digitalWrite(SolenoidAPIN,LOW);
    digitalWrite(SolenoidBPIN,LOW);
    ReactionRUN = false;  }
}
MyPanel.SetText(F("Progress"), ((millis()-
StartMillis)/TotalTime)*100);
MyPanel.SetText(F("TotalTime"),
(TotalTime/60000));
MyPanel.SetText(F("Eflowrate"),
(Eflowrate));
}

else if ((ReactionRUN == true)&&(Capilliary
== 4)) {

  if (Solvent==1){
TotalTime=(((TotalVolume+InjectionVolume)
/((0.401*Pressure)+0.550))*60000);
  TotalMillis=(StartMillis+TotalTime);
  Eflowrate=((0.3904*Pressure)+0.5241);
  if (millis() - StartMillis >=
((InjectionVolume/((0.401*Pressure)+0.550))
*60000)) {
    digitalWrite(SolenoidAPIN,LOW);
    digitalWrite(SolenoidBPIN,LOW);
    ReactionRUN = false;  }
}
 else if (Solvent==2){
TotalTime=(((TotalVolume+InjectionVolume)
/((1.1175*Pressure)+1.4675))*60000);
  TotalMillis=(StartMillis+TotalTime);
  Eflowrate=((1.1175*Pressure)+1.4675);
  if (millis() - StartMillis >=
((InjectionVolume/((1.1175*Pressure)+1.467
5))*60000)) {
    digitalWrite(SolenoidAPIN,LOW);
    digitalWrite(SolenoidBPIN,LOW);
    ReactionRUN = false;  }
}

 else if (Solvent==3){
TotalTime=(((TotalVolume+InjectionVolume)
/((0.606*Pressure)+1.23))*60000);
  TotalMillis=(StartMillis+TotalTime);
  Eflowrate=((0.5513*Pressure)+1.1187);
  if (millis() - StartMillis >=
((InjectionVolume/((0.606*Pressure)+1.23))*
60000)) {
    digitalWrite(SolenoidAPIN,LOW);
    digitalWrite(SolenoidBPIN,LOW);
    ReactionRUN = false; }
}

  else if (Solvent==4){
TotalTime=(((TotalVolume+InjectionVolume)
/((0.3093*Pressure)+0.5240))*60000);
  TotalMillis=(StartMillis+TotalTime);
  Eflowrate=((0.3362*Pressure)+0.5696);
  if (millis() - StartMillis >=
((InjectionVolume/((0.3093*Pressure)+0.524
0))*60000)) {
    digitalWrite(SolenoidAPIN,LOW);
    digitalWrite(SolenoidBPIN,LOW);
    ReactionRUN = false; }
}

  else if (Solvent==5){
TotalTime=(((TotalVolume+InjectionVolume)
/((0.221*Pressure)+0.158))*60000);
  TotalMillis=(StartMillis+TotalTime);
  Eflowrate=((0.221*Pressure)+0.158);
  if (millis() - StartMillis >=
((InjectionVolume/((0.221*Pressure)+0.158))
*60000)) {
    digitalWrite(SolenoidAPIN,LOW);
    digitalWrite(SolenoidBPIN,LOW);
    ReactionRUN = false; }
}

  else if (Solvent==6){
TotalTime=(((TotalVolume+InjectionVolume)
/((1.395*Pressure)+1.145))*60000);
  TotalMillis=(StartMillis+TotalTime);
  Eflowrate=((1.395*Pressure)+1.145);
  if (millis() - StartMillis >=
((InjectionVolume/((1.395*Pressure)+1.145))
*60000)) {
    digitalWrite(SolenoidAPIN,LOW);
    digitalWrite(SolenoidBPIN,LOW);
    ReactionRUN = false; }
}
```

```
MyPanel.SetText(F("Progress"), ((millis()-StartMillis)/TotalTime)*100);
MyPanel.SetText(F("TotalTime"), (TotalTime/60000));
MyPanel.SetText(F("Eflowrate"), (Eflowrate));
}

if ((TemperatureStart == true)&& ((millis() - LastSent) > SendInterval)) {
  LastSent=millis();
  long temp1 =1023 - analogRead (A0);
  float sensor1 = beta /(log(((1025.0 * 10 / temp1) - 10) / 10) + beta / 298.0) - 273.0;
  long temp2 =1023 - analogRead (A1);
  float sensor2 = beta /(log(((1025.0 * 10 / temp2) - 10) / 10) + beta / 298.0) - 273.0;
  long temp3 =1023 - analogRead (A2);
  float sensor3 = beta /(log(((1025.0 * 10 / temp3) - 10) / 10) + beta / 298.0) - 273.0;
  TempPlot.SendData("ADCValue1", millis(),sensor1);
  TempPlot.SendData("ADCValue2", millis(),sensor2);
  TempPlot.SendData("ADCValue3", millis(),sensor3);
  }
if (TemperatureStop== false){
  }

if ((FlowStart == true)&& ((millis() - LastSent) > SendInterval)&& (Solvent == 1))
{
  LastSent=millis();
   int ret;
    Wire.beginTransmission(ADDRESS);
    Wire.write(0x36);
    Wire.write(0x08);
    ret = Wire.endTransmission();
    Wire.requestFrom(ADDRESS, 9);
    sensor_flow_value  = Wire.read() << 8;
    sensor_flow_value |= Wire.read();
    float flow_value = ((int16_t)sensor_flow_value)/SCALE_FACTOR_FLOW;
    FlowPlot.SendData("Flow rate", millis(),flow_value);
  }

else if ((FlowStart == true)&& ((millis() - LastSent) > SendInterval)&& (Solvent == 2))
{
  LastSent=millis();
   int ret;
    Wire.beginTransmission(ADDRESS);
    Wire.write(0x36);
    Wire.write(0x08);
    ret = Wire.endTransmission();
    Wire.requestFrom(ADDRESS, 9);
    sensor_flow_value  = Wire.read() << 8;
    sensor_flow_value |= Wire.read();
    float flow_value = ((5*(pow(10,(-5))))*(pow((int16_t)sensor_flow_value,2))+(0.0051*sensor_flow_value)+0.0215);
    FlowPlot.SendData("Flow rate", millis(),flow_value);
  }

else if ((FlowStart == true)&& ((millis() - LastSent) > SendInterval)&& (Solvent == 3))
{
  LastSent=millis();
   int ret;
    Wire.beginTransmission(ADDRESS);
    Wire.write(0x36);
    Wire.write(0x08);
    ret = Wire.endTransmission();
    Wire.requestFrom(ADDRESS, 9);
    sensor_flow_value  = Wire.read() << 8;
    sensor_flow_value |= Wire.read();
    float flow_value = (pow(10,(-5))*(pow((int16_t)sensor_flow_value,2))+(0.0071*sensor_flow_value)-0.0918);
    FlowPlot.SendData("Flow rate", millis(),flow_value);
    FlowPlot.SendData("sensor", millis(),sensor_flow_value);
  }

  else if ((FlowStart == true)&& ((millis() - LastSent) > SendInterval)&& (Solvent == 4))
{
  LastSent=millis();
   int ret;
    Wire.beginTransmission(ADDRESS);
    Wire.write(0x36);
    Wire.write(0x08);
    ret = Wire.endTransmission();
    Wire.requestFrom(ADDRESS, 9);
```

```
    sensor_flow_value  = Wire.read() << 8;
    sensor_flow_value |= Wire.read();
    float flow_value = ((2*(pow(10,(-
5))))*(pow((int16_t)sensor_flow_value,2))+(0
.0085*sensor_flow_value)-0.1453);
    FlowPlot.SendData("Flow rate",
millis(),flow_value);
  }

  else if  ((FlowStart == true)&& ((millis() -
LastSent) > SendInterval)&& (Solvent == 5))
{
  LastSent=millis();
   int ret;
    Wire.beginTransmission(ADDRESS);
    Wire.write(0x36);
    Wire.write(0x08);
    ret = Wire.endTransmission();
    Wire.requestFrom(ADDRESS, 9);
    sensor_flow_value  = Wire.read() << 8;
    sensor_flow_value |= Wire.read();
    float flow_value = ((4*(pow(10,(-
5))))*(pow((int16_t)sensor_flow_value,2))+(0
.0054*sensor_flow_value)-0.0335);
    FlowPlot.SendData("Flow rate",
millis(),flow_value);
  }

  else if  ((FlowStart == true)&& ((millis() -
LastSent) > SendInterval)&& (Solvent == 6))
{
  LastSent=millis();
   int ret;
    Wire.beginTransmission(ADDRESS);
    Wire.write(0x36);
    Wire.write(0x08);
    ret = Wire.endTransmission();
    Wire.requestFrom(ADDRESS, 9);
    sensor_flow_value  = Wire.read() << 8;
    sensor_flow_value |= Wire.read();
    float flow_value = ((3*(pow(10,(-
5))))*(pow((int16_t)sensor_flow_value,2))+(0
.0022*sensor_flow_value)+0.1508);
    FlowPlot.SendData("Flow rate",
millis(),flow_value);
     FlowPlot.SendData("sensor",
millis(),sensor_flow_value);
  }

if ((PhotoStart == true)&& ((millis() -
LastSent) > SendInterval)) {
  LastSent=millis();
  long temp4 =1023 - analogRead (A3);
  float sensor4 = beta /(log(((1025.0 * 10 /
temp4) - 10) / 10) + beta / 298.0) - 273.0;
  PhotoPlot.SendData("ADCValue4",
millis(),sensor4);
  }
if (PhotoStop== false){
  }

if ((millis()-StartMillis)/TotalTime < 1){
 MyPanel.SetText(F("Progress"), ((millis()-
StartMillis)/TotalTime)*100);
  }

else if ((millis()-StartMillis)/TotalTime == 1){
  MyPanel.SetText(F("Progress"), 100);
  }

else if ((millis()-StartMillis)/TotalTime > 1){
  MyPanel.SetText(F("Progress"), 100);
  }
MyPanel.SetProgress(F("ReactionProgress
Bar"), ((millis()-StartMillis)/TotalTime)*100);
```

S.7 Ultimaker Cura Settings

S.7.1. PLA 3D Printing settings without support

Material	**PLA**	**Profile**	Normal 0.15 mm
Print Core	AA 0.4	**Speed**	
Quality		Print Speed	70 mm/s
Layer height	0.15 mm	Infill Speed	70 mm/s
Initial layer height	0.27 mm	Wall Speed	55 mm/s
Line Width	0.35 mm	Outer Wall Speed	50 mm/s
Wall Line Width	0.35 mm	Inner Wall Speed	55 mm/s
Outer Wall Line Width	0.35 mm	Top/Bottom Speed	40 mm/s
Inner Wall(s) Line Width	0.3 mm	Travel Speed	250 mm/s
Top/Bottom Line Width	0.35 mm	Initial Layer Speed	20 mm/s
Infill Line Width	0.42 mm	Skirt/Brim Speed	20 mm/s
Initial Layer Line Width	120 %	Enable Acceleration Control	Tick
Shell		Enable Jerk Control	Tick
Wall Extruder	Not overridden	**Travel**	
Outer Wall Extruder	Not overridden	Enable Retraction	Tick
Inner Wall Extruder	Not overridden	Retract at Layer Change	Blank
Wall Thickness	1 mm	Retraction Distance	6.5 mm
Wall Line Count	3	Retraction Speed	25 mm/s
Top/Bottom Extruder	Not overridden	Combing Mode	All
Top/Bottom Thickness	1 mm	Avoid Printer Parts When traveling	Tick
Top thickness	1 mm	Avoid Supports When Traveling	Blank
Top layers	10	Travel Avoid Distance	3 mm
Bottom Thickness	1 mm	Z Hop When Retracted	Tick
Bottom Layers	10	Z Hop Only Over Printed Parts	Tick
Optimize Wall Printing Order	Tick	Z Hop Height	2 mm
Fill Gaps Between Walls	Everywhere	Z Hop After Extruder Switch	Tick
Horizontal Expansions	0 mm	**Cooling**	
Enable ironing	Blank	Enable Print Cooling	Tick
Infill		Fan Speed	100 %
Infill Extruder	Not overridden	Regular Fan Speed	100 %

Infill Density	20 %
Infill Line Distance	6.3 mm
Infill Patter	Triangles
Infill Line Multiplier	1
Infill overlap Percentage	0 %
Infill layer Thickness	0.1mm
Gradual Infill Steps	0
Material	
Printing Temperature	205 ºC
Printing Temperature Initial Layer	210 ºC
Initial Printing Temperature	195 ºC
Final Printing Temperature	190 ºC
Build Plate Temperature	60 ºC
Build Plate Temperature Initial Layer	60 ºC
Standby Temperature	100 ºC
Dual Extrusions	
Enable Prime Tower	Blank
Special Modes	
Surface Mode	Normal
Spiralize Outer Contour	Blank
Build Plate Adhesion	
Enable Prime Blob	Tick
Build Plate Adhesion Type	Brim
Build Plate Adhesion Extruder	Extruder 1
Brim Width	7 mm
Brim Line Count	17
Brim Only on Outside	Tick

Maximum Fan Speed	100 %
Regular/Maximum Fan Speed Threshold	10 s
Initial Fan Speed	0 %
Regular Fan Speed at Height	0.47 mm
Regular Fan Speed at Layer	4
Minimum Layer Time	5 s
Minimum Speed	5 mm/ s
Lift Head	Blank
Support	
Generate Support	Tick
Support Extruder	Extruder 1
Support Infill Extruder	Extruder 1
First Layer Support Extruder	Extruder 1
Support Interface Extruder	Extruder 1
Support Structure	Normal
Support Placement	Touching Build plate
Support Overhang Angle	60 º
Support Pattern	Zig Zag
Support Density	15 %
Support Horizontal Expansion	0 mm
Support Infill Layer Thickness	0.1 mm
Gradual Support Infill Steps	0
Enable Support Interface	Blank
Enable Support Roof	Blank
Enable Support Floor	Blank
Experimental	
Make Overhang printable	Blank
Use adaptive layers	Blank

Supplementary Table 29: PLA 3D Printing settings without support.

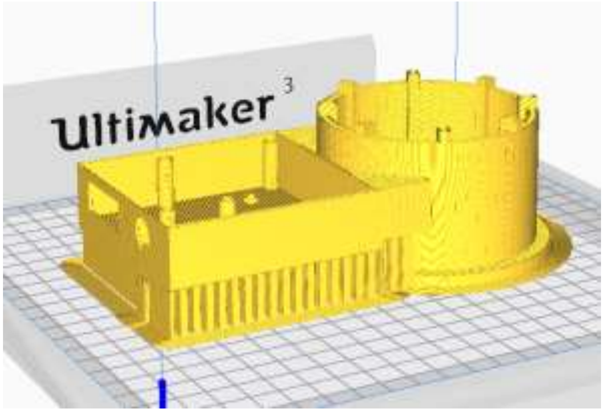
S.7.2. PLA 3D Printing settings with support

Material	**PLA**	**Quantity**	131 g – 16.53 m
Print Core	AA 0.4	**Speed**	
Quality		Print Speed	70 mm/s
Layer height	0.15 mm	Infill Speed	70 mm/s
Initial layer height	0.27 mm	Wall Speed	55 mm/s
Line Width	0.35 mm	Outer Wall Speed	50 mm/s
Wall Line Width	0.35 mm	Inner Wall Speed	55 mm/s
Outer Wall Line Width	0.35 mm	Top/Bottom Speed	40 mm/s
Inner Wall(s) Line Width	0.3 mm	Travel Speed	250 mm/s
Top/Bottom Line Width	0.35 mm	Initial Layer Speed	20 mm/s
Infill Line Width	0.42 mm	Skirt/Brim Speed	20 mm/s
Initial Layer Line Width	120 %	Enable Acceleration Control	Tick
Shell		Enable Jerk Control	Tick
Wall Extruder	Not overridden	**Travel**	
Outer Wall Extruder	Not overridden	Enable Retraction	Tick
Inner Wall Extruder	Not overridden	Retract at Layer Change	Blank
Wall Thickness	1 mm	Retraction Distance	6.5 mm
Wall Line Count	3	Retraction Speed	25 mm/s
Top/Bottom Extruder	Not overridden	Combing Mode	All
Top/Bottom Thickness	1 mm	Avoid Printer Parts When traveling	Tick
Top thickness	1 mm	Avoid Supports When Traveling	Blank
Top layers	10	Travel Avoid Distance	3 mm
Bottom Thickness	1 mm	Z Hop When Retracted	Tick
Bottom Layers	10	Z Hop Only Over Printed Parts	Tick
Optimize Wall Printing Order	Tick	Z Hop Height	2 mm
Fill Gaps Between Walls	Everywhere	Z Hop After Extruder Switch	Tick
Horizontal Expansions	0 mm	**Cooling**	
Enable ironing	Blank	Enable Print Cooling	Tick
Infill		Fan Speed	100 %
Infill Extruder	Not overridden	Regular Fan Speed	100 %
Infill Density	20 %	Maximum Fan Speed	100 %

Infill Line Distance	6.3 mm	Regular/Maximum Fan Speed Threshold	10 s
Infill Patter	Triangles	Initial Fan Speed	0%
Infill Line Multiplier	1	Regular Fan Speed at Height	0.47 mm
Infill overlap Percentage	0 %	Regular Fan Speed at Layer	4
Infill layer Thickness	0.1 mm	Minimum Layer Time	5 s
Gradual Infill Steps	0	Minimum Speed	5 mm/s
Material		Lift Head	Blank
Printing Temperature	205 ºC	**Support**	
Printing Temperature Initial Layer	210 ºC	Generate Support	Tick
Initial Printing Temperature	195 ºC	Support Extruder	Extruder 1
Final Printing Temperature	190 ºC	Support Infill Extruder	Extruder 1
Build Plate Temperature	60 ºC	First Layer Support Extruder	Extruder 1
Build Plate Temperature Initial Layer	60 ºC	Support Interface Extruder	Extruder 1
Standby Temperature	100 ºC	Support Structure	Normal
Dual Extrusions		Support Placement	Touching Build plate
Enable Prime Tower	Blank	Support Overhang Angle	60º
Special Modes		Support Pattern	Zig Zag
Surface Mode	Normal	Support Density	15 %
Spiralize Outer Contour	Blank	Support Horizontal Expansion	0 mm
Build Plate Adhesion		Support Infill Layer Thickness	0.1 mm
Enable Prime Blob	Tick	Gradual Support Infill Steps	0
Build Plate Adhesion Type	Brim	Enable Support Interface	Blank
Build Plate Adhesion Extruder	Extruder 1	Enable Support Roof	Blank
Brim Width	7 mm	Enable Support Floor	Blank
Brim Line Count	17	**Experimental**	
Brim Only on Outside	Tick	Make Overhang printable	Blank
		Use adaptive layers	Blank

Supplementary Table 30: PLA 3D Printing settings with support.

This design requires support printing because a part of it is not in contact with the plate and thus will not print correctly.



Supplementary Figure 82: Cura preview file showing the Phototemperature Senor Module with the support placement to 3D print.

S.7.3. PP 3D Printing settings

Material	**PP**	**Profile**	Normal 0.15 mm
Print Core	AA 0.4	**Speed**	
Quality		Print Speed	25 mm/s
Layer height	0.15 mm	Infill Speed	25 mm/s
Initial layer height	0.27 mm	Wall Speed	25 mm/s
Line Width	0.38 mm	Outer Wall Speed	25 mm/s
Wall Line Width	0.38 mm	Inner Wall Speed	25 mm/s
Outer Wall Line Width	0.38 mm	Top/Bottom Speed	25 mm/s
Inner Wall(s) Line Width	0.38 mm	Travel Speed	300 mm/s
Top/Bottom Line Width	0.38 mm	Initial Layer Speed	15 mm/s
Infill Line Width	0.38 mm	Skirt/Brim Speed	15 mm/s
Initial Layer Line Width	120 %	Enable Acceleration Control	Tick
Shell		Enable Jerk Control	Tick
Wall Extruder	Not overridden	**Travel**	
Outer Wall Extruder	Not overridden	Enable Retraction	Tick
Inner Wall Extruder	Not overridden	Retract at Layer Change	Blank
Wall Thickness	1.14 mm	Retraction Distance	6.5 mm
Wall Line Count	3	Retraction Speed	35 mm/s
Top/Bottom Extruder	Not overridden	Combing Mode	All
Top/Bottom Thickness	1.1 mm	Avoid Printer Parts When traveling	Tick

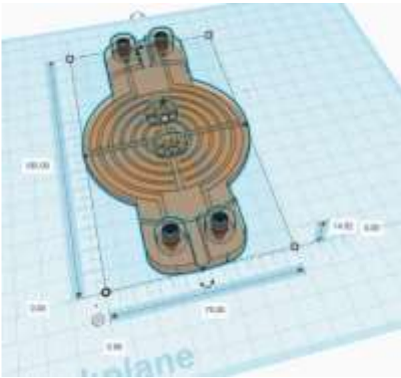
Top thickness	1.1 mm	Avoid Supports When Traveling	Blank
Top layers	0	Travel Avoid Distance	3 mm
Bottom Thickness	1.1 mm	Z Hop When Retracted	Tick
Bottom Layers	999999	Z Hop Only Over Printed Parts	Tick
Optimize Wall Printing Order	Tick	Z Hop Height	2 mm
Fill Gaps Between Walls	Everywhere	Z Hop After Extruder Switch	Tick
Horizontal Expansions	0mm	**Cooling**	
Enable ironing	Blank	Enable Print Cooling	Tick
Infill		Fan Speed	20 %
Infill Extruder	Not overridden	Regular Fan Speed	20 %
Infill Density	100 %	Maximum Fan Speed	100 %
Infill Line Distance	0.76 mm	Regular/Maximum Fan Speed Threshold	7 s
Infill Patter	Octet	Initial Fan Speed	0 %
Infill Line Multiplier	1	Regular Fan Speed at Height	0.87 mm
Infill overlap Percentage	0 %	Regular Fan Speed at Layer	6
Infill layer Thickness	0.15 mm	Minimum Layer Time	7 s
Gradual Infill Steps	0	Minimum Speed	2.5 mm/s
Material		Lift Head	Blank
Printing Temperature	235 ºC	**Support**	
Printing Temperature Initial Layer	240 ºC	Generate Support	Blank
Initial Printing Temperature	230 ºC	**Build Plate Adhesion**	
Final Printing Temperature	225 ºC	Enable Prime Blob	Tick
Build Plate Temperature	85 ºC	Build Plate Adhesion Type	Brim
Build Plate Temperature Initial Layer	90 ºC	Build Plate Adhesion Extruder	Extruder 1
Standby Temperature	100 ºC	Brim Width	20 mm
Dual Extrusions		Brim Line Count	44
Enable Prime Tower	Blank	Brim Only on Outside	Tick
Special Modes		**Experimental**	
Surface Mode	Normal	Make Overhang printable	Blank
Spiralize Outer Contour	Blank	Use adaptive layers	Blank

Supplementary Table 31: PP 3D Printing settings**.**

S.8 <u>Desings for 3D printed reactors</u>

Different types of reactors were designed to be able to fit a temperature sensor or a temperature probe sensor inside the CDR or the channels. Once the design of the reactors was completed in Tinkercad, they were exported individually as an STL (Standard Tessellation Language) file and uploaded to Cura software. The settings used to print each component are shown in shown in section 7.3. The printing of these reactors need to have post processing modifications which are specified in each reactor section. Finally, the models were sliced and USB-connected to the printer. All the reactors were printed with white Polypropylene 2.85mm filament.

S.8.1 CDR for the Phototemperature module with a temperature sensor slot



Supplementary Figure 83: Tinkercad design illustrating the planned reactor for the Phototemperature module.

Material	PP
Time	6 h 8 min
Quantity	26 g – 4.63 m
Profile	Normal 0.15 mm

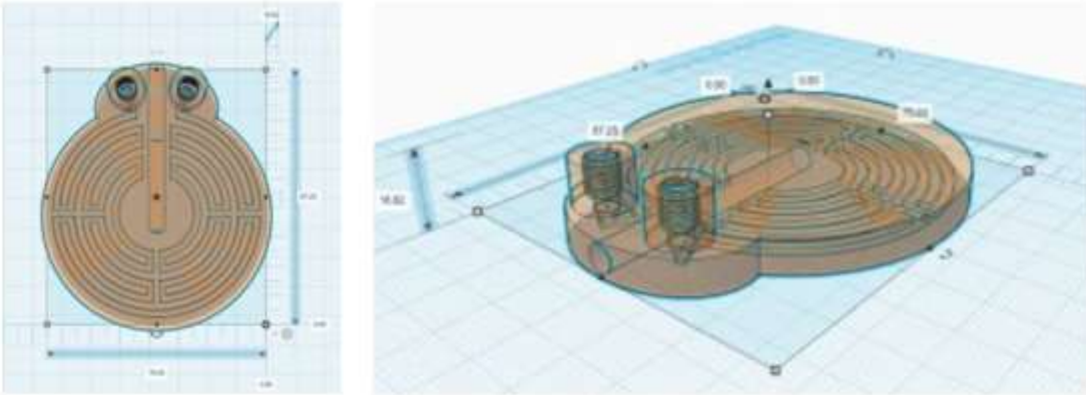
Supplementary Table 32: Ultimaker Cura Settings for 3D printing of the reactor for the Phototemperature module.

Post processing plugin modifications – change At Z 5.2.1 :

Layer number 2	Flow rate	115 ºC
Layer number 14	Flow rate	120 ºC
Layer number 38	Flow rate	100 ºC
	Change extruder 1 Temp	220 ºC

Supplementary Table 33: Ultimaker Cura Post processing modification settings for 3D printing of the reactor for the Phototemperature module.

S.8.2 CDR with a temperature sensor slot



Supplementary Figure 84: Tinkercad design illustrating the planned reactor with a temperature sensor slot.

Material	PP
Time	6 h 55 min
Quantity	30 g – 5.26 m
Profile	Normal 0.15 mm

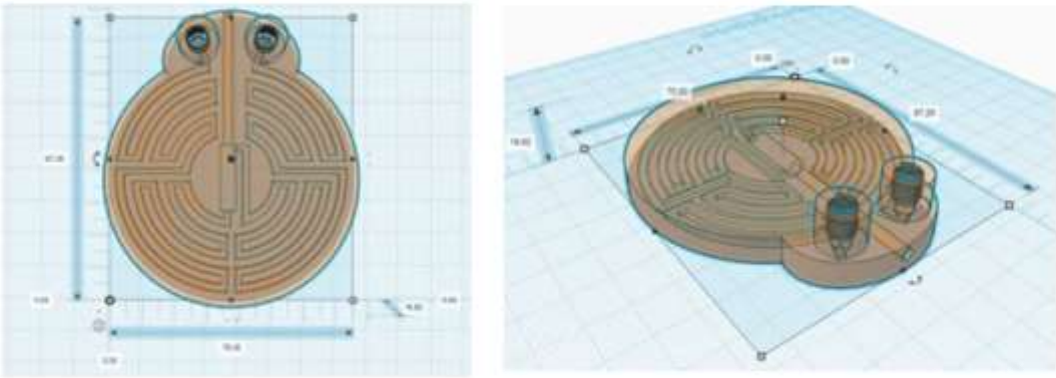
Supplementary Table 34: Ultimaker Cura Settings for 3D printing of the reactor with a temperature sensor slot.

Post processing plugin modifications – change At Z 5.2.1 :

<table>
<tr><td>Layer number 2</td><td>Flow rate</td><td>115 ºC</td></tr>
<tr><td>Layer number 27</td><td>Flow rate</td><td>120 ºC</td></tr>
<tr><td rowspan="2">Layer number 49</td><td>Flow rate</td><td>100 ºC</td></tr>
<tr><td>Change extruder 1 Temp</td><td>220 ºC</td></tr>
</table>

Supplementary Table 35: Ultimaker Cura Post processing modification settings for 3D printing of the reactor with a temperature sensor slot.

S.8.3 CDR with a probe slot inserted into the channel



Supplementary Figure 85: Tinkercad design illustrating the planned reactor with a probe slot inserted into the channel.

Material	PP
Time	7 h 3 min
Quantity	30 g – 5.35 m
Profile	Normal 0.15 mm

Supplementary Table 36: Ultimaker Cura Settings for 3D printing of the reactor inserted into the channel.

Post processing plugin modifications – change At Z 5.2.1 :

Layer number 2	Flow rate	115 ºC
Layer number 27	Flow rate	120 ºC
Layer number 49	Flow rate	100 ºC
	Change extruder 1 Temp	220 ºC

Supplementary Table 37: Ultimaker Cura Post processing modification settings for 3D printing of the reactor inserted into the channel.

S.9 Technical data of electronics

S.9.1 Arduino boards

The following table lists the technical specifications of the Arduino Uno Wi-Fi Rev 2 and Arduino Mega 2560:

Arduino Board		Arduino Uno Wiffi Rev 2	Arduino Mega
Microcontroller		ATmega4809	ATmega2560
Pins	Built-in LED Pin	25	13
	Digital I/O input pins	14	54
	Analog input pins	6	16
	PWM pins	5	
Connectivity	Bluetooth and Wi-Fi	Nina W102 uBlox module	
Communication	UART	Yes	
	I2C	Yes	
	SPI	Yes	
Power	I/O Voltage	5V	
	Input Voltage	7-12V	
	DC current per I/O Pin	20mA	
Memory	ATmega4809	6KB SRAM, 48KB flash, 256 bytes EEPROM	8 KB SRAM, 256 KB flash, 4KB EEPROM
	Nina W102 uBlox module	448 KB ROM, 520KB SRAM, 2MB Flash	
Dimensions	Weight	25 g	37 g
	Width	53.4 mm	53.3 mm
	Length	68.6 mm	101.52 mm

Supplementary Table 38: Arduino Uno Wifi Rev 2 and Arduino Mega Features. 2.

S.9.2 Relay Shield

The relay shield used is the Keyestudio 4-channel Relay shield.

DC Power Interface	5V
Size	69 mm x 54 mm x 26 mm
Weight	39.4 g
Contact capacity	AC120V / 3A ; DC24V/3A
Relay channels	4 channels
Compatibility	UNO R3 boards

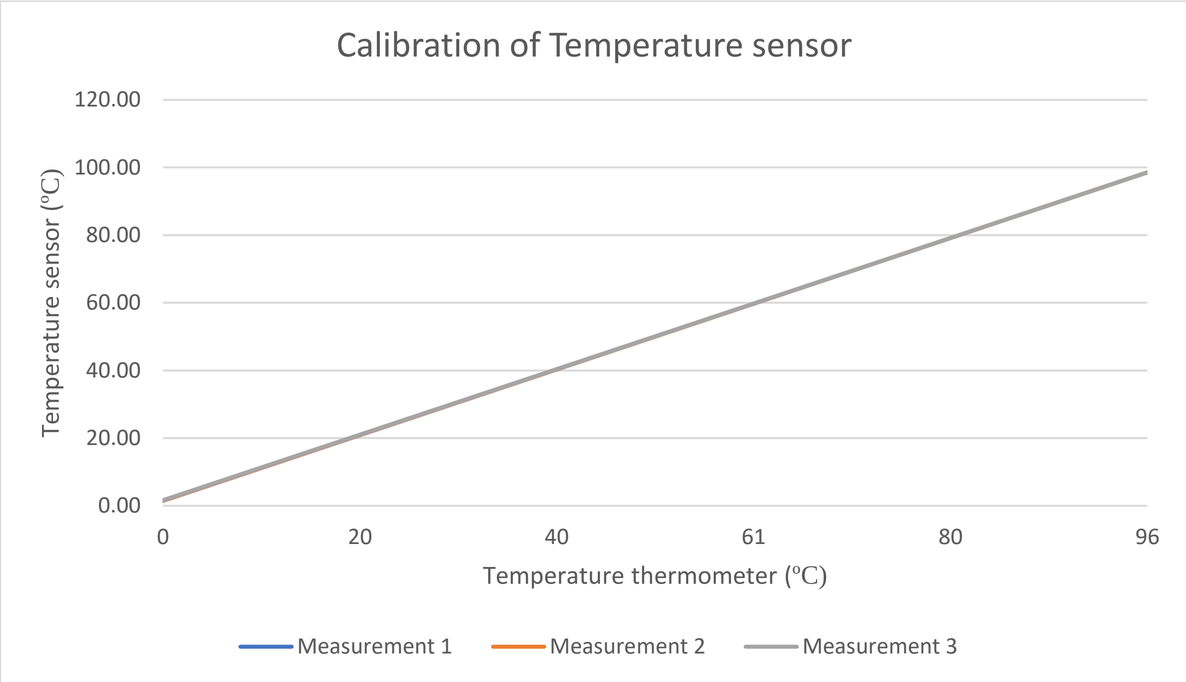
Supplementary Table 39: Relay shield Features. 3.

S.9.3 Temperature Sensor

DC Power Interface	5V
Size	65 mm x 40 mm x 15 mm
Weight	9.2 g
Waterproof	Yes
Compatibility	UNO R3 boards

Supplementary Table 40: Temperature sensor Features.

S.9.3.1 Calibration



Supplementary Graph 7: relation between temperature from three different temperature sensors and a thermometer.

TEMPERATURE (ºC)	0 ºC	20 ºC	40 ºC	61 ºC	80 ºC	96 ºC
Sensor 1	0.92	20.32	40.41	61.34	80.60	96.50
Sensor 2	0.96	20.27	40.66	61.40	80.17	96.72
Sensor 3	1.04	20.45	40.57	61.45	80.79	96.29

Supplementary Table 41: Supplementary Table 26: Temperature calibrations.

S.9.4 Flow rate sensor

The flow rate sensor used is an SLF3S-1300F Liquid Flow sensor from Sensirion.

H_2O Full scale flow rate	±40 mL/min
H_2O Sensor output limit	±65 mL/min
Accuracy	±5% of the measured value
Typical Supply Voltage DC	3.5 V
Operating Temperature	+5 … +50 ºC

Supplementary Table 42: Flow rate Sensor Features.[7]

		Product Family	SLF3S	LD20	LPG10	SLI, LG16	SLS, LS32	SLQ-QT	SLG
		Wetted Materials	PPS, Stainless steel, Epoxy	PEI, LCP, Epoxy	Glass	Glass or Quartz, PEEK, (FEP)	Stainless steel, PEEK, PTFE	Quartz, PFA, (PCTFE)	Quartz, Titanium, PEEK
Type	Chemical	Concentration							
Aqueous solutions	NaCl (Sodium chloride)	0.9%	+	+	+	+	+	+	+
	NaOH (Sodium hydroxide)	1 M (=4%)	o	-	-	-	+	-	-
	HCl (Hydrochloric acid)	1 M (=3%)	o	-	o	o	-	+	o
	H2O2 (Hydrogen peroxide)	30%	-	-	+	+	o[1]	+	+
	NaClO (Sodium hypochlorite)	30%	o	o	o	o	o	o	o
	$Ca(ClO)_2$ (Calcium hypochlorite)	32%	o	o	o	o	o	o	o
Organic solvent	Ethanol IPA (Isopropyl alcohol or 2-propanol) Methanol	100%	+	+	+	+	+	+	+
	ACN (Acteonitrile)	10 - 100%[2]	o/-	-	+	+	+	+	+
	THF (Tetrahydrofuran)	10 - 100%	o/-	-	+	o	+/o	+	o
	DMSO (Dimethylsulfoxid)	10 - 100%	o/-	-	+	o	+/o	+	o
	Acetone	100%	+	-	+	+	+	+	+
	Toluene	100%	o	-	+	o	o	+	o
	Heptane	100%	+	+	+	+	+	+	+
	Gasoline Diesel fuel Kerosene (Jet fuel)	100%	+	o	+	+	+	+	+
Other	Mineral oil, Vegetable oil	100%	+	+	+	+	+	+	+
	Silicone oil	100%	+	+	+	+	+	+	+
	Engine oil, Lubricant oil	100%	+	o	+	+	+	+	+

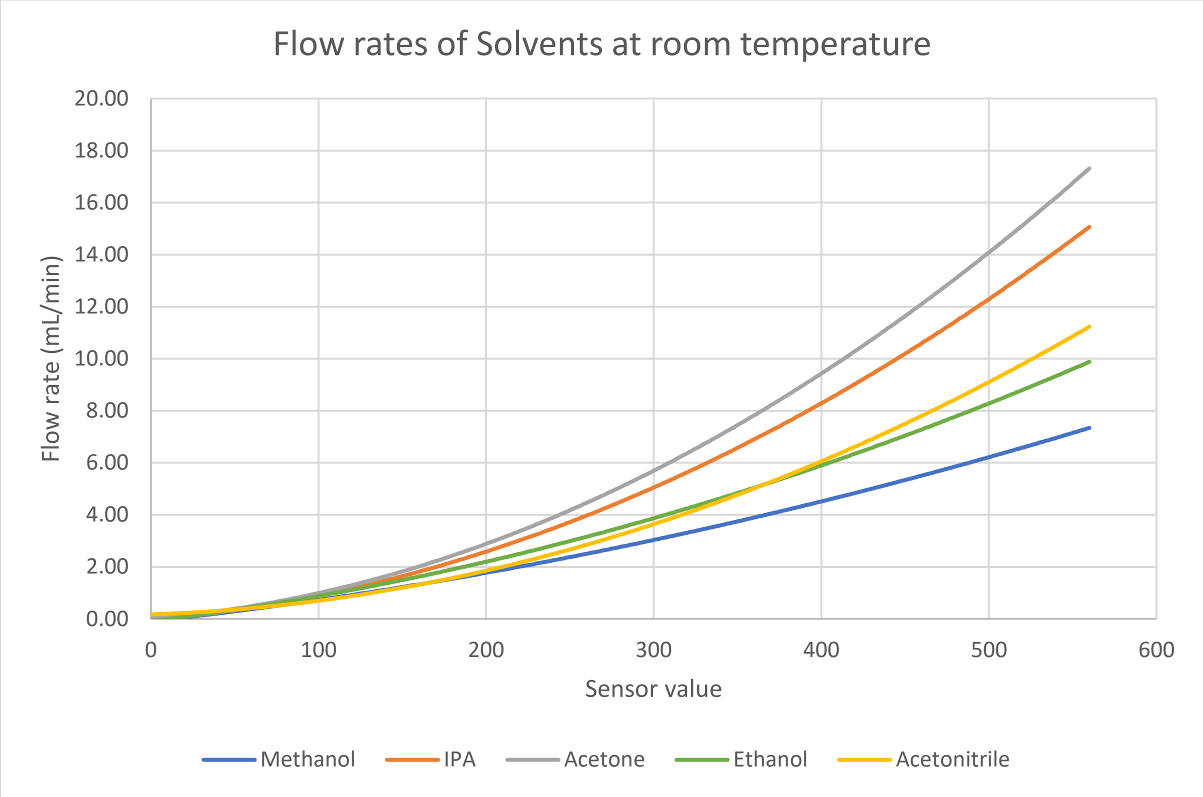
+ : most likely suitable for long term usage
o : possibly suitable for short/medium duration usage
- : likely not suitable

Supplementary Figure 86: Flow rate sensor liquid chemical compatibilities.[7]

The flow rate sensor SLF3C-1300F was also calibrated. Almost all liquid sensors available on the market have been calibrated with water; however, if other solvents are to be used, the sensor must first be calibrated for each one in order to provide accurate measurements. The Arduino code was modified to remove the conversion of the sensor value to the flow rate so that we could only read the sensor value and correlate it to the flow rate of the solvent under test. The calibrated solvents were selected based on which ones were most probable to be used in reactions; these included isopropanol (IPA), methanol, ethanol, acetone, and acetonitrile. The table of compatibilities (**Sup. Fig. 86**) indicates that all of them, except for acetonitrile, can be used for extended periods of time.

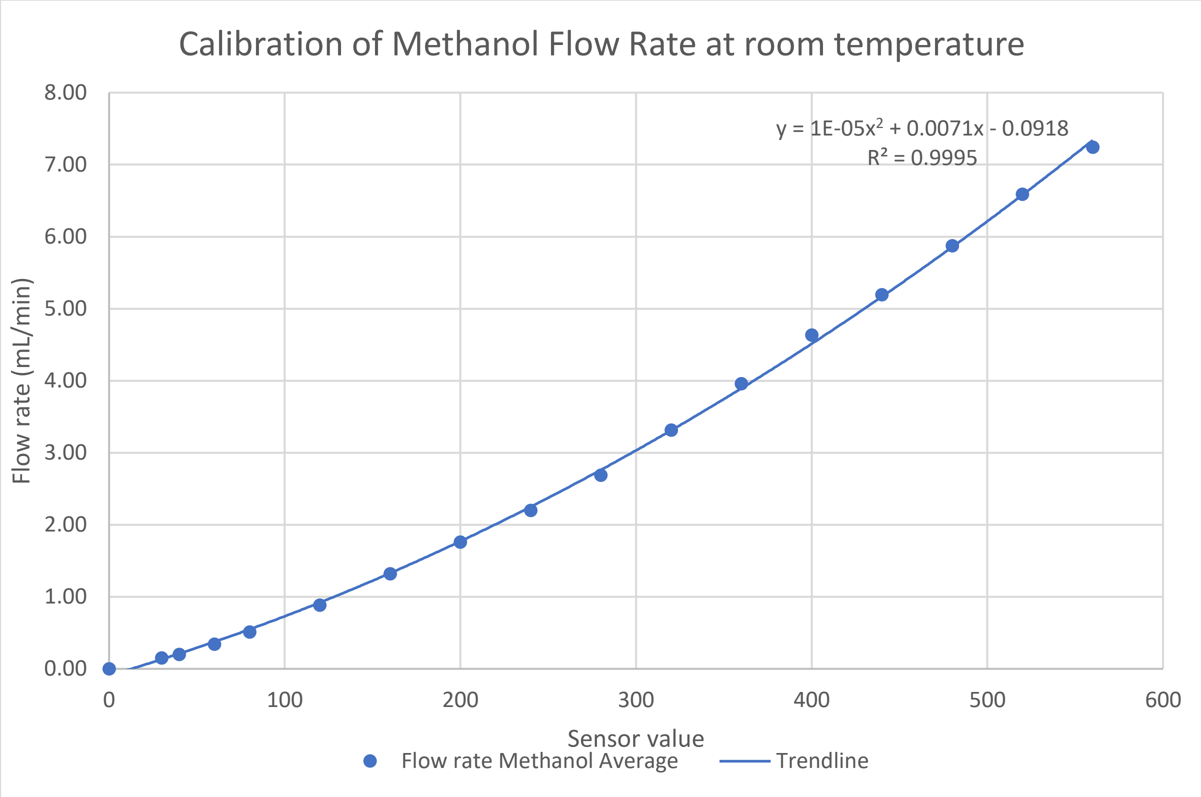
To obtain the correlation between the flow rate and the sensor value, we measured the flow rate of the solvent at room temperature in triplicate at specific sensor value until the entire sensor value range was covered. The 3D printed flow chemistry system uses PEEK capillaries as a back-pressure regulator, however they tend to expand when used with some solvents, such as acetone, so metal capillaries were used to provide a more accurate measurement.

The data from each solvent is shown in the following Graph. In order to calibrate the sensor with acetonitrile, which is not highly compatible with the sensor, fewer values were measured, and the sensor was flushed with ethanol after each measurement to prevent potential damage.



Supplementary Graph 8: Correlation of the flow rates of multiple solvents and the liquid flow sensor value at room temperature.

S.9.4.1 Flow rate sensor calibration Methanol

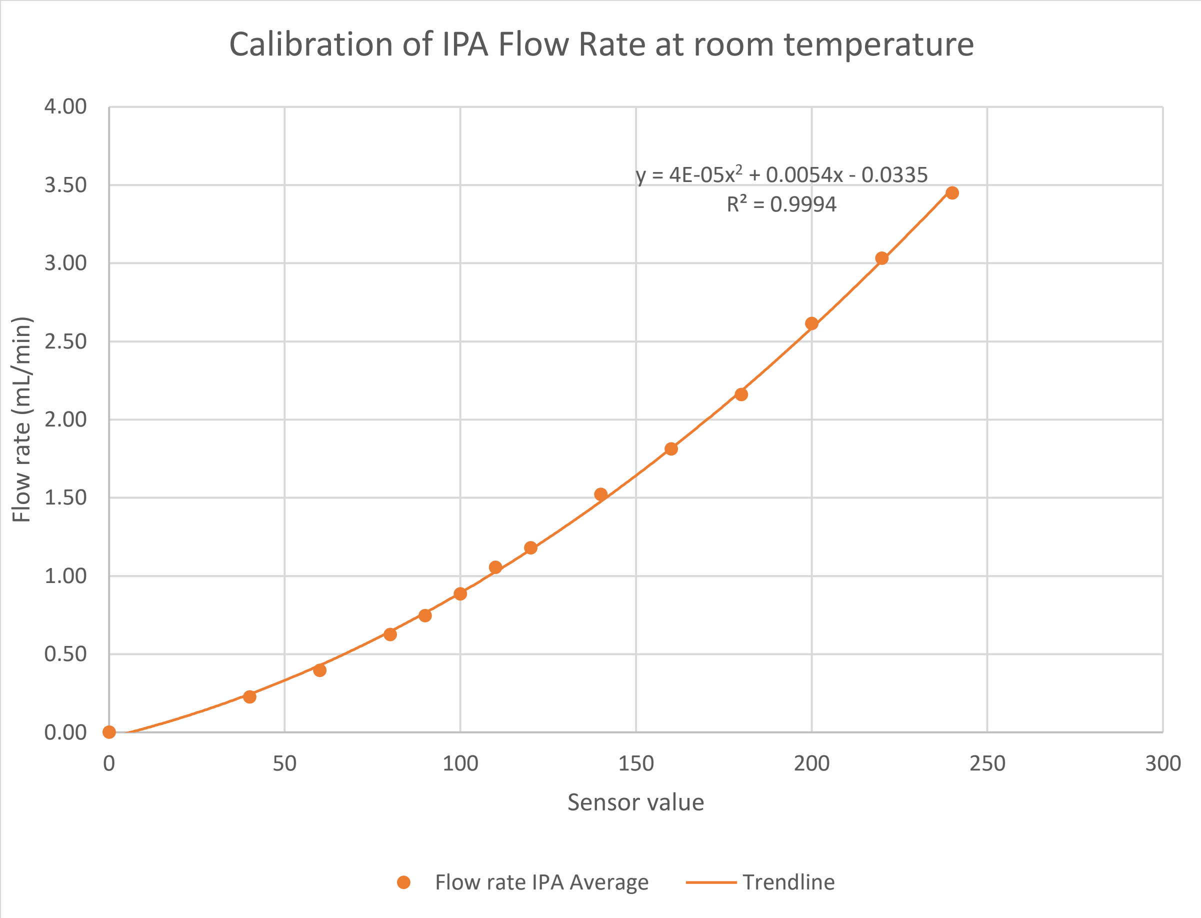


Supplementary Graph 9: Correlation of the flow rates of Methanol and the liquid flow sensor value at room temperature.

Flow rate Sensor value	Flow rate 1	Flow rate 2	Flow rate 3	Flow rate Methanol Average
0	0	0	0	0
30	0.15	0.15	0.15	0.15
40	0.18	0.21	0.21	0.20
60	0.34	0.34	0.35	0.34
80	0.51	0.52	0.50	0.51
120	0.89	0.89	0.87	0.88
160	1.30	1.34	1.32	1.32
200	1.74	1.77	1.77	1.76
240	2.22	2.17	2.21	2.20
280	2.66	2.69	2.71	2.69
320	3.28	3.37	3.29	3.32
360	3.98	3.96	3.93	3.96
400	4.50	4.71	4.69	4.64
440	5.26	5.10	5.23	5.19
480	5.77	5.87	6.00	5.88
520	6.53	6.69	6.55	6.59
560	7.14	7.38	7.19	7.24

Supplementary Table 43: Flow rate of Methanol calibrations.

S.9.4.2 Flow rate sensor calibration IPA

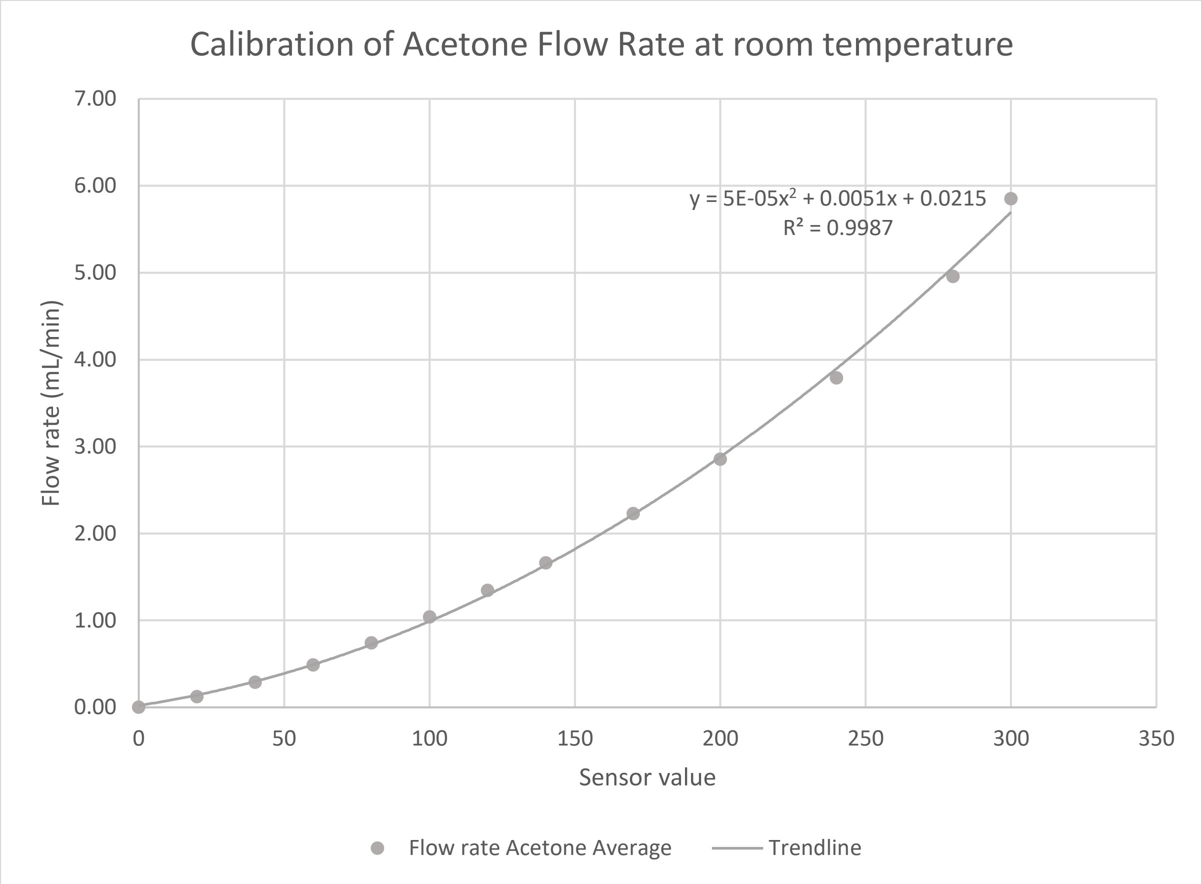


Supplementary Graph 10: Correlation of the flow rates of IPA and the liquid flow sensor value at room temperature.

Flow rate Sensor value	Flow rate IPA 1	Flow rate IPA 2	Flow rate IPA 3	Flow rate IPA Average
0	0.00	0.00	0.00	0.00
40	0.22	0.22	0.24	0.23
60	0.38	0.40	0.40	0.39
80	0.62	0.62	0.64	0.63
90	0.75	0.75	0.74	0.75
100	0.89	0.88	0.88	0.88
110	1.05	1.06	1.06	1.05
120	1.19	1.14	1.20	1.18
140	1.49	1.53	1.55	1.52
160	1.84	1.77	1.82	1.81
180	2.12	2.19	2.17	2.16
200	2.63	2.63	2.58	2.61
220	3.05	2.99	3.06	3.03
240	3.43	3.44	3.48	3.45

Supplementary Table 44: Flow rate of IPA calibrations.

S.9.4.3 Flow rate sensor calibration Acetone

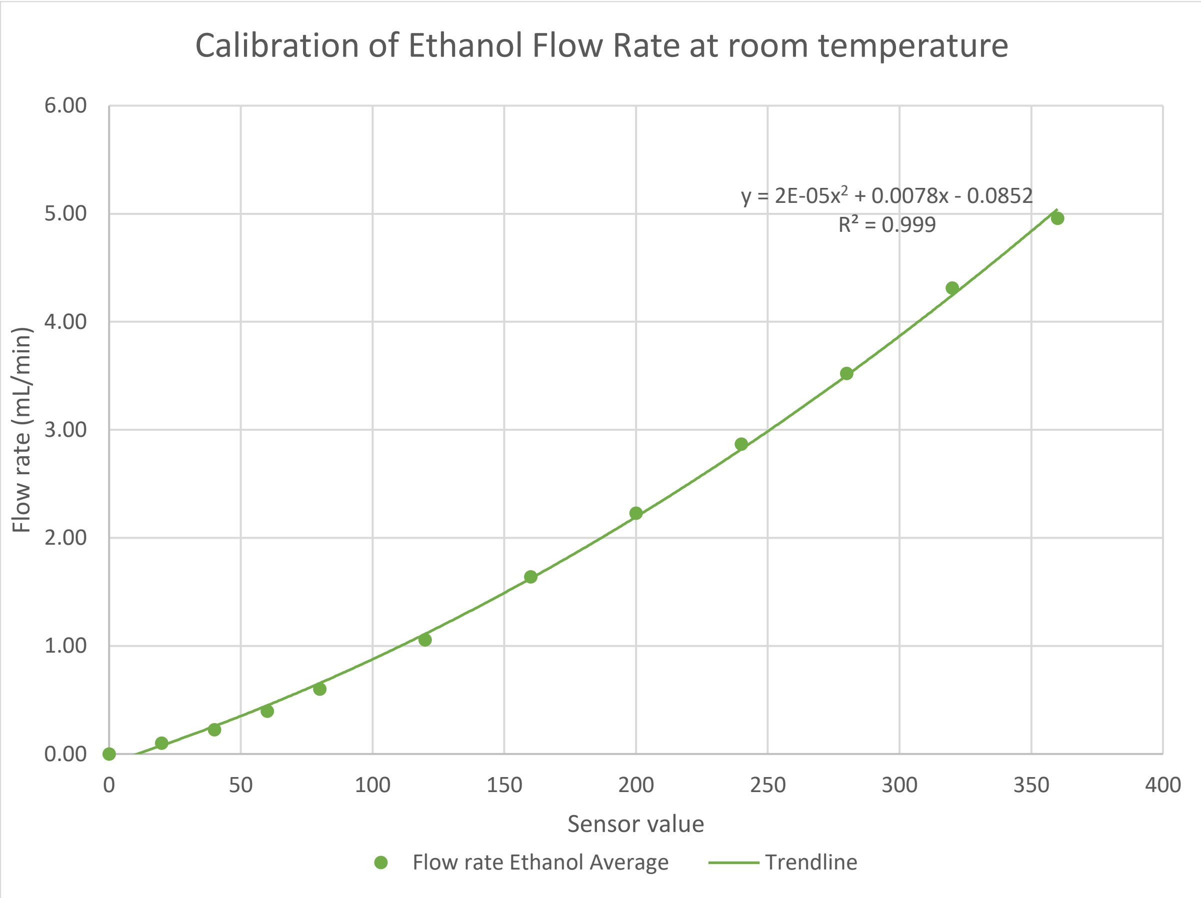


Supplementary Graph 11: Correlation of the flow rates of Acetone and the liquid flow sensor value at room temperature.

Flow rate Sensor value	Flow rate Acetone 1	Flow rate Acetone 2	Flow rate Acetone 3	Flow rate Acetone Average
0	0	0	0	0.00
20	0.12	0.12	0.12	0.12
40	0.29	0.28	0.30	0.29
60	0.45	0.50	0.51	0.49
80	0.74	0.74	0.73	0.74
100	1.00	1.05	1.06	1.04
120	1.37	1.31	1.34	1.34
140	1.65	1.66	1.66	1.66
170	2.23	2.22	2.22	2.23
200	2.84	2.82	2.91	2.86
240	3.81	3.75	3.81	3.79
280	4.98	4.92	4.96	4.95
300	5.92	5.77	5.87	5.85

Supplementary Table 45: Flow rate of Acetone calibrations.

S.9.4.4 Flow rate sensor calibration Ethanol

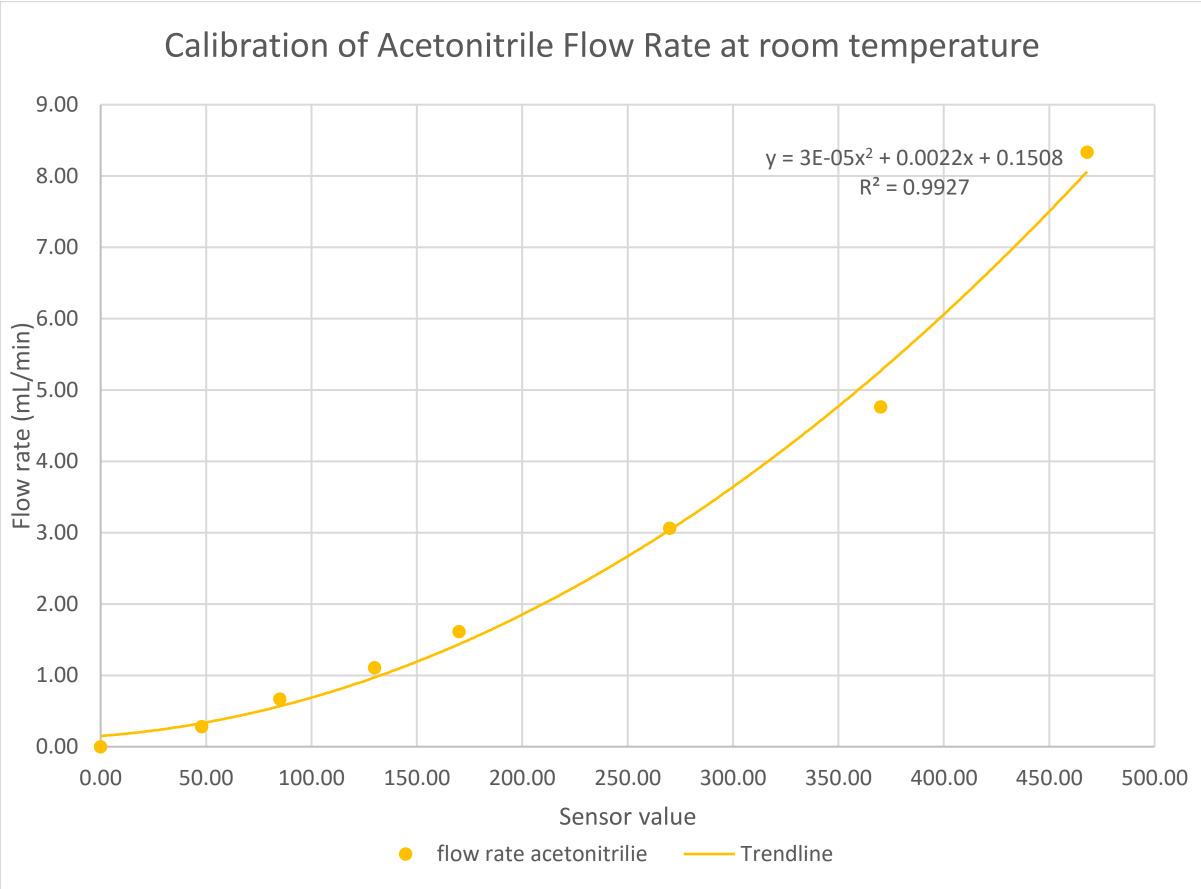


Supplementary Graph 12: Correlation of the flow rates of Ethanol and the liquid flow sensor value at room temperature.

Flow rate Sensor value	Flow rate 1	Flow rate 2	Flow rate 3	Flow rate Ethanol Average
0	0	0	0	0.00
20	0.09	0.10	0.11	0.10
40	0.22	0.21	0.24	0.22
60	0.39	0.40	0.40	0.40
80	0.60	0.60	0.60	0.60
120	1.07	1.05	1.05	1.06
160	1.61	1.64	1.67	1.64
200	2.23	2.23	2.22	2.23
240	2.82	2.85	2.92	2.87
280	3.51	3.48	3.57	3.52
320	4.32	4.31	4.30	4.31
360	4.95	4.96	4.96	4.96

Supplementary Table 46: Flow rate of Ethanol calibrations.

S.9.4.5 Flow rate sensor calibration Acetonitrile



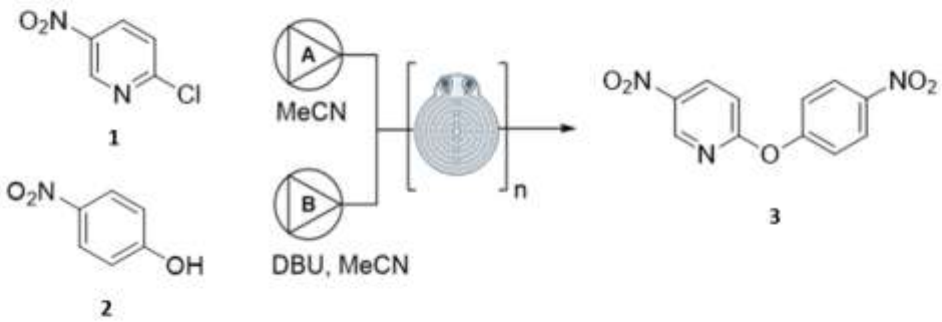
Supplementary Graph 13: Correlation of the flow rates of Acetonitrile and the liquid flow sensor value at room temperature.

Flow rate sensor	flow rate Acetonitrile
0.00	0.00
48.00	0.28
85.00	0.67
130.00	1.11
170.00	1.61
270.00	3.06
370.00	4.76
468.00	8.33

Supplementary Table 47: Flow rate of Acetonitrile calibrations.

S.10 General reaction procedures

Scheme 1: Synthesis of 5-Nitro-2-(4-nitrophenoxy)pyridine (**3**)[1]



A solution of 2-chloro-5-nitropyridine (0.07 g, 0.442 mmol) **1** in acetonitrile (2 mL) was injected into loop A. A solution of 4-nitrophenol (0.08 g, 0.574 mmol) **2** and DBU (0.13 mL, 0.861 mmol) in acetonitrile (2 mL) was injected into loop B. The temperature on the hotplate was set to 65 °C, 75 °C and 100 °C in order for the second CDR internal temperature to be 55 °C, 65 °C and 85 °C respectively. The first 15 minutes only ACN was running through the CDRs. Once the hotplate temperature was reached and stabilized, the valves were turned to the left to run the reaction for 60 min approximately. They were made to pass through two to six CDRs (3D printed in PP) at 5 psi (0.33 mL/min). A 120 mm RED capillary was fitted after the last CDR in order to maintain the constant flow.[1] The reaction mixture was then concentrated under reduced pressure to give the crude product.

Entry 1: the residue was purified via Biotage (9:1 Hex/EtOAc; snap 10 g column) to give 5-nitro-2-(4-nitrophenoxy)pyridine **3** (0.025 g, 0.097 mmol, 22 % yield) as a white solid.

Entry 2: the residue was purified via Biotage (9:1 Hex/EtOAc; snap 10 g column) to give 5-nitro-2-(4-nitrophenoxy)pyridine **3** (0.031 g, 0.119 mmol, 27 % yield) as a white solid.

Entry 3: the residue was purified via Biotage (9:1 Hex/EtOAc; snap 10 g column) to give 5-nitro-2-(4-nitrophenoxy)pyridine **3** (0.044 g, 0.168 mmol, 38 % yield) as a white solid.

Entry 4: the residue was purified via Biotage (9:1 Hex/EtOAc; snap 10 g column) to give 5-nitro-2-(4-nitrophenoxy)pyridine **3** (0.044 g, 0.168 mmol, 38 % yield) as a white solid.

Entry 5: the residue was purified via Biotage (9:1 Hex/EtOAc; snap 10 g column) to give 5-nitro-2-(4-nitrophenoxy)pyridine **3** (0.074 g, 0.283 mmol, 64 % yield) as a white solid.

Entry 6: the residue was purified via Biotage (9:1 Hex/EtOAc; snap 10 g column) to give 5-nitro-2-(4-nitrophenoxy)pyridine **3** (0.080 g, 0.305 mmol, 69 % yield) as a white solid.

Entry 7: the residue was purified via Biotage (9:1 Hex/EtOAc; snap 10 g column) to give 5-nitro-2-(4-nitrophenoxy)pyridine **3** (0.1078 g, 0.413 mmol, 93 % yield) as a white solid.

Entry 8: Scale up

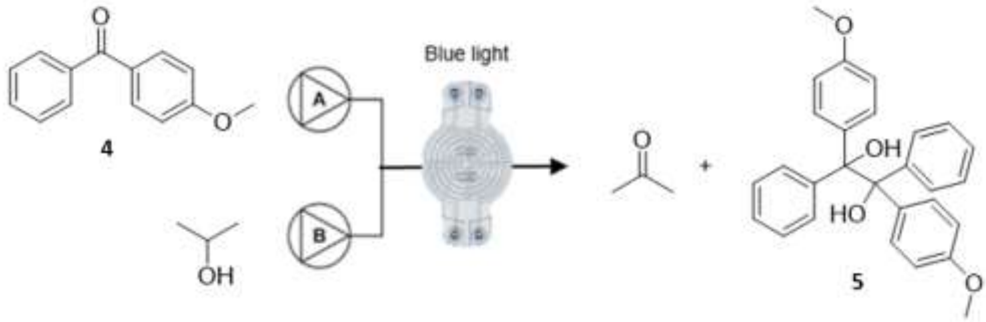
A Duran pressure bottle was charged with ACN (250 mL). Bottle A was charged with 80 mL of a solution 1 2-chloro-5-nitropyridine (containing 2.8 g) **1** in ACN and connected to loop A. Bottle B was charged with 80 mL a solution of 4-nitrophenol (containing 3.2 g) **2** in the presence of DBU (containing 5.2 mL) in ACN and connected to loop B. The temperature on the hotplate was set at 100 °C in order for the second CDR internal temperature to be 83 °C. The system was set up at 6 PSI, 6 CDRs and a 120 mm RED capillary was fitted at the end of the 6th reactor in order to maintain the constant flow. All tubing was filled with acetonitrile prior to carrying out the reaction. The scaled up version of the SnAr reaction was controlled through the Megunolink platform:

- System Volume: 2 mL
- Loop volume: 2 mL (Loop A) and 2 mL (Loop B)
- Pre Collect: 5 mL
- Post Collect: 10 mL
- Injection Volume: 60 mL

The residue was purified via Biotage (9:1, Hex/EtOAc; zip sphere 45 column) to give 5-nitro-2-(4-nitrophenoxy)pyridine (1.557 g, 5.96 mmol, 90 % yield) as a white solid.

Mp 95-98 ºC; 1H NMR (400 MHz, Chloroform-d) δ 9.03 (d, J = 2.8 Hz, 1H), 8.56 (dd, J = 9.0, 2.8 Hz, 1H), 8.38 – 8.30 (m, 2H), 7.40 – 7.32 (m, 2H), 7.18 (d, J = 9.0 Hz, 1H). 13C NMR (101 MHz, Chloroform-d) δ 165.4 , 157.5 , 145.2 , 144.6 , 141.1 , 135.4 , 125.6 , 122.1 , 112.2

Scheme 2: Synthesis of 1,2-bis(4-methoxyphenyl)-1,2-diphenylethane-1,2-diol (**4**)[8]



Entry 9: a solution of 4-methoxybenzophenone (0.176 g, 0.829 mmol) **4** in 5 mL of acetone and 5 mL of 2-propanol was reacted in the presence of acetic acid to obtain the product **5**. 10 mL of a 0.2 M stock solution was taken for reaction and 5 drops of acetic acid added as a catalyst. The solution was placed into two 2 mL syringes, each one injected to a different loop (A and B) and passed through one photoreactor at 0.1 mL/ min. The lamp used was a Kessil 395 nm lamp, blue light and maximum intensity while being air cooled, placed 30 mm above CDR, mirror and reflective coating. A 120mm RED capillary was fitted after the photoreactor in order to maintain the constant flow. The reaction took 100 minutes to be completed. The reaction mixture was then concentrated under reduced pressure to give the crude product. The residue was purified by column chromatography via Biotage (9:1 Hex/EtOAc; snap 10 g column) to give 1,2-bis(4-methoxyphenyl)-1,2-diphenylethane-1,2-diol (0.108 g, 0.253 mmol, 61%) as a colorless oil.

Entry 10: Scale up

A 250 mL Duran pressure bottle was charged with 60mL of a solution of 4-methoxybenzophenone **4** (1.1 g, 5.18 mmol) in acetone (30 mL) and 2-propanolol (30 mL) in the presence of acetic acid (20 drops) to obtain the product **5**. Bottle A and B were charged with 80 mL of a 1:1 solution of 2-propranol and acetone. The duran foil was covered with foil to avoid contact with light. The solution was passed through one photoreactor at 0.066 mL/ min. The lamp used was 395 nm, blue light and maximum intensity while being air cooled, placed 30 mm above CDR, mirror and reflective coating. A 120mm RED capillary was fitted after the photoreactor in order to maintain the constant flow. The reaction took 375 minutes to be completed. The scaled up version of the SnAr reaction was controlled through the Megunolink platform:

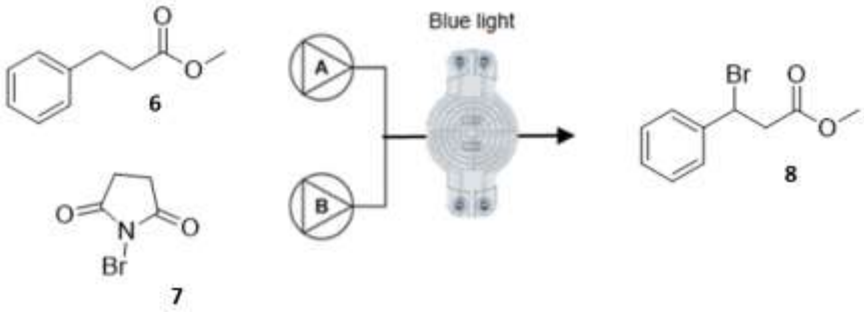
- System Volume: 2 mL
- Loop volume: 2 mL (Loop A) and 2 mL (Loop B)

- Pre Collect: 5 mL
- Post Collect: 5 mL
- Injection Volume: 25 mL

The residue was purified by column chromatography via Biotage (9:1 Hex/EtOAc; snap 10 g column) to give 1,2-bis(4-methoxyphenyl)-1,2-diphenylethane-1,2-diol (0.9057 g, 2.124 mmol, 82%) as a colorless oil.

1H NMR (400 MHz, Propanol) δ 7.31 (dd, J = 8.7, 4.5 Hz, 2H), 7.17 (dd, J = 8.6, 3.4 Hz, 5H), 6.74 – 6.68 (m, 2H), 3.76 (d, J = 4.8 Hz, 3H). 13C NMR (101 MHz, CDCl3) δ 158.5, 158.4, 144.6, 144.6, 136.5, 136.4, 130.0, 129.9, 128.7, 128.6, 127.4, 127.3, 112.7, 112.7, 99.9, 83.0, 55.2, 55.2.

Scheme 3: Synthesis of methyl 3-bromo-3-phenylpropanoate (**8**)[9]



Entry 11: solution of methyl 3-phenylpropanoate **6** (0.470 mL) and NBS **7** (0.561 g) were dissolved in acetonitrile (10 mL) to make a 10 mL stock solution of 0.5 M. The solution was placed into two 2 mL syringes (4 mL in total), each one injected to a different loop (A and B) and passed through the photoreactor at 5 mL/ min. CDR was air cooled. Kessil A160WE (lamp A), 0% colour (blue), 100% intensity placed 30 mm above CDR, mirror & reflective coating. A 120 mm RED capillary was fitted after the last CDR in order to maintain the constant flow. After the injection of the reactants into the flow path, the reaction was collected over a period of 4 minutes. The residue was purified via Biotage (2-5% Hex/EtOAc; snap 10 g column) to give methyl 3-bromo-3-phenylpropanoate (0.43 g, 1.79 mmol, 89%) **8** as a colorless oil.

Entry 12: Scale up

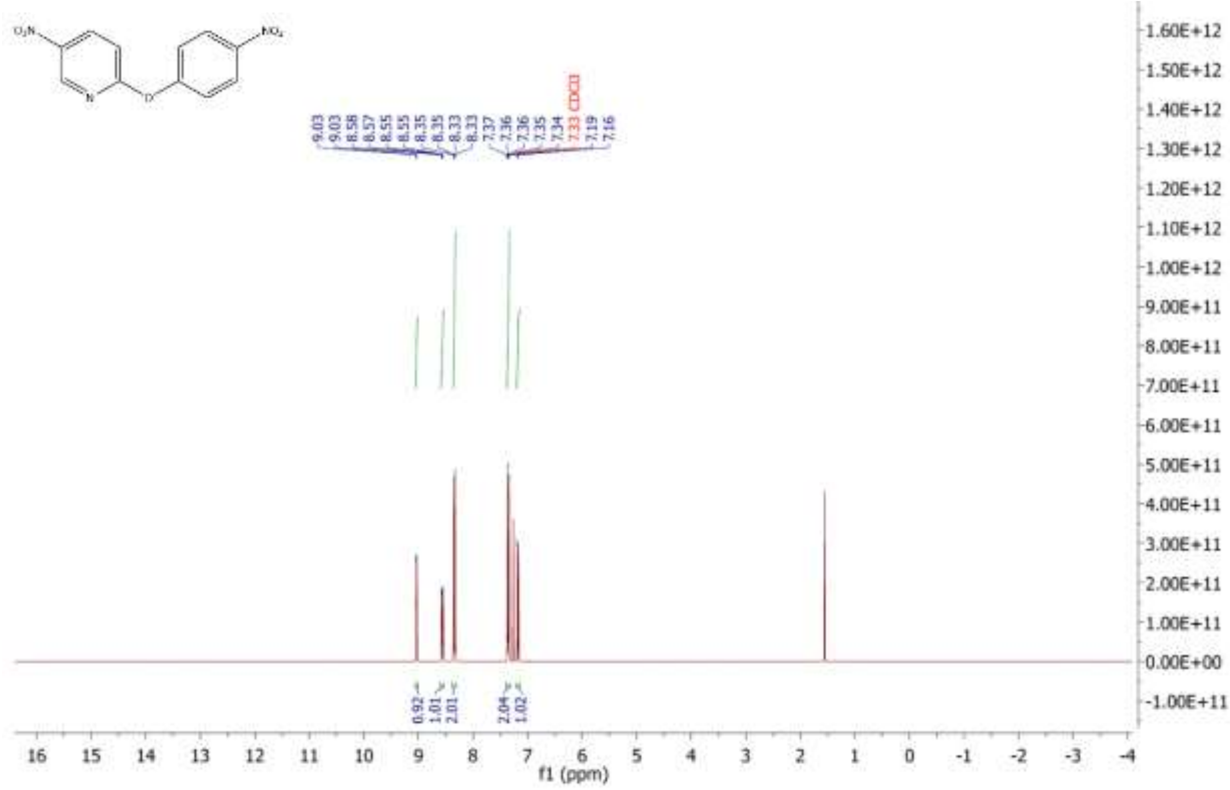
A 250 mL Duran pressure bottle was charged with 50 mL of a solution of methyl 3-phenylpropanoate **6** (containing 2.05 g) and NBS **7** (containing 2.33 g) were dissolved in acetonitrile (50 mL) to make a 50 mL stock solution of 0.25 M. Bottle A and B were charged with 80 mL of acetonitrile. The solution passed through the photoreactor at 5 mL/ min. CDR was air cooled. Kessil A160WE (lamp A), 0% colour (blue), 100% intensity placed 30 mm above CDR, mirror & reflective coating. A 120mm RED capillary was fitted after the last CDR in order to maintain the constant flow. After the injection of the reactants into the flow path, the reaction was collected over a period of 20 minutes. The scaled up version of the SnAr reaction was controlled through the Megunolink platform:

- System Volume: 2 mL
- Loop volume: 2 mL (Loop A) and 2 mL (Loop B)
- Pre Collect: 5 mL
- Post Collect: 5 mL
- Injection Volume: 40 mL

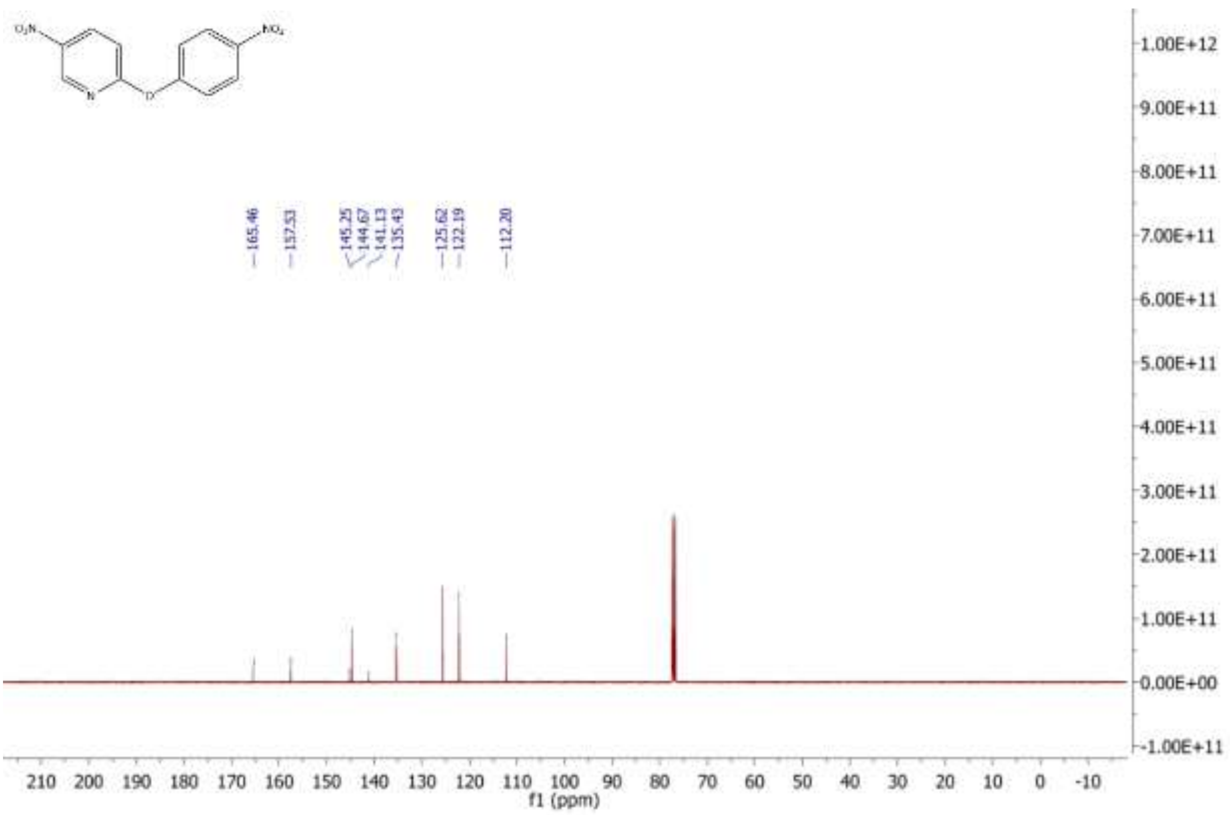
The residue was purified via Biotage (2-5% Hex/EtOAc; snap 50 g column) to give methyl 3-bromo-3-phenylpropanoate (2.07 g, 8.52 mmol, 85%) **8** as a yellow oil.

1H NMR (400 MHz, Chloroform-d) δ 7.47 – 7.25 (m, 5H), 5.45 – 5.37 (m, 1H), 3.70 (s, 3H), 3.36 (dd, J = 16.2, 9.1 Hz, 1H), 3.22 (dd, J = 16.2, 6.1 Hz, 1H). 13C NMR (101 MHz, CDCl3) δ 169.9, 140.5, 128.6, 126.9, 51.8, 47.6, 44.4.

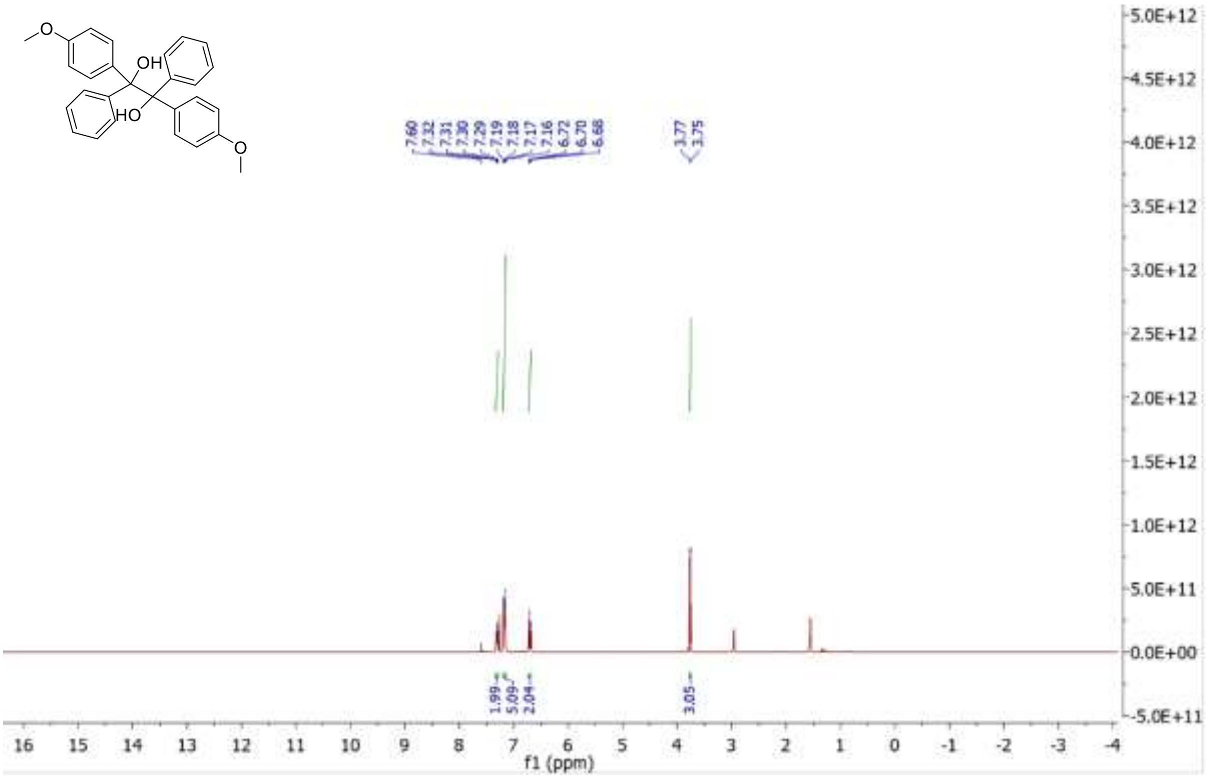
H-NMR Product 3



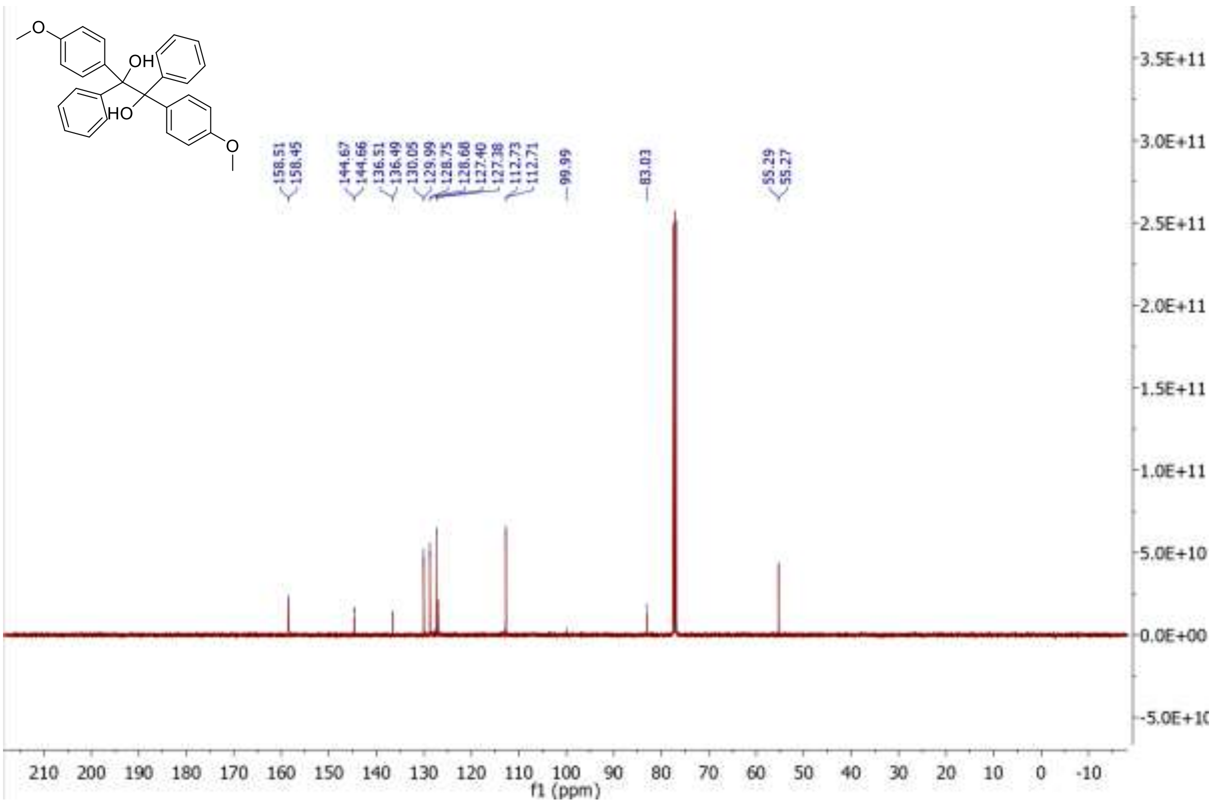
C-NMR Product 3



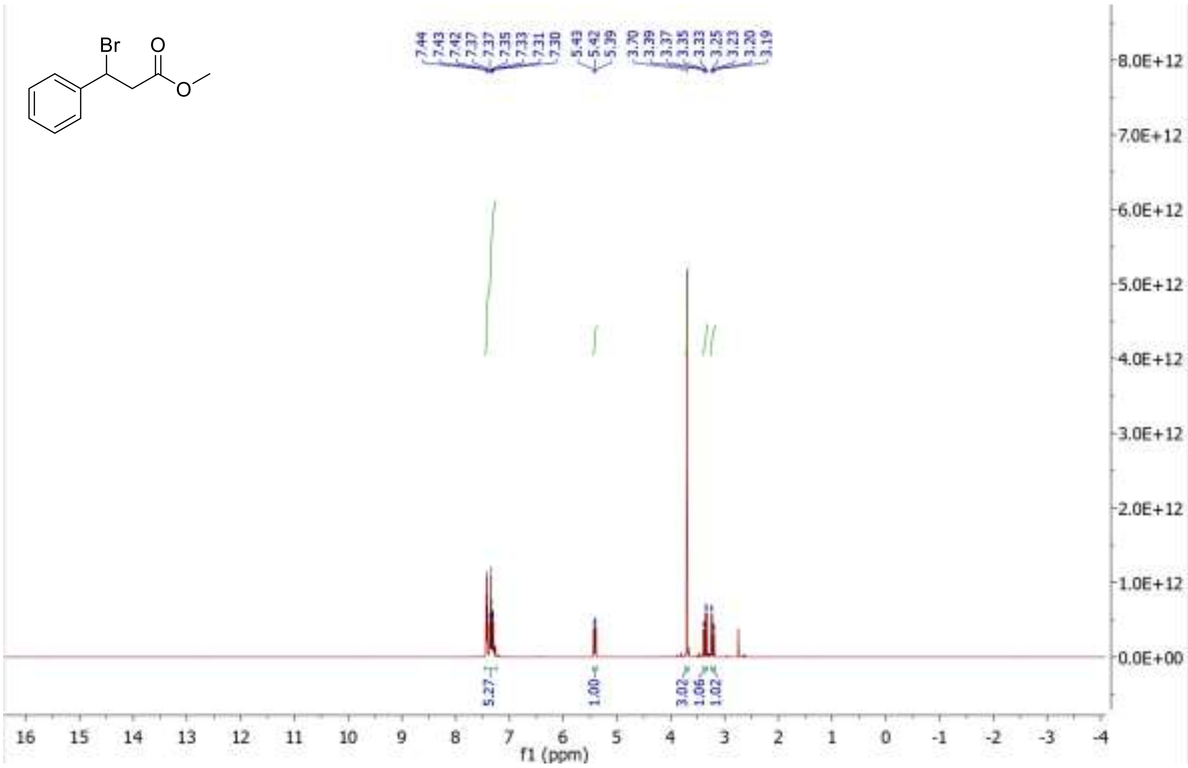
H- NMR Product 5



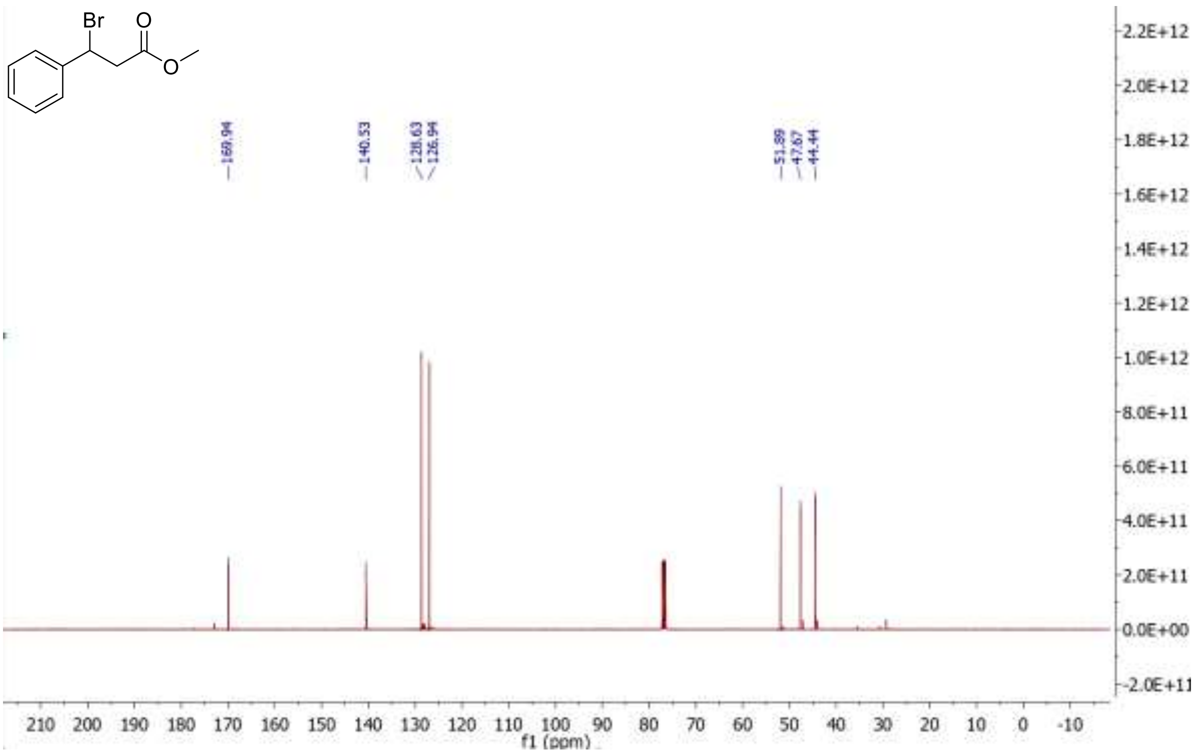
C- NMR Product 5



H- NMR Product 8



C- NMR Product 8



S.11 References

1) Penny MR, Rao ZX, Peniche BF, Hilton ST Modular 3D printed compressed air driven continuous-flow systems for chemical synthesis. *Eur J Org Chem* **2019**:3783–3787.
2) https://3dgbire.com/pages/about-us Accessed 8th April 2023.
3) https://www.tinkercad.com/ Accessed 8th April 2023.
4) https://ultimaker.com/software/ultimaker-cura/ Accessed 8th April 2023.
5) https://www.megunolink.com/ Accessed 8th April 2023.
6) Walmsley L, Hilton S, Sellier E, Penny M, Maddox D. Control and Monitoring of Temperature in 3D-Printed Circular Disk Reactors for Continuous Flow Photochemistry using Raspberry Pi Based Software. *ChemRxiv*. Cambridge: Cambridge Open Engage; **2021**.
7) https://sensirion.com/products/catalog/SLF3C-1300F/ Accessed 8th April 2023.
8) Kate Volpe K., Podlesny, E. E., Modernization of a Photochemical Reaction for the Undergraduate Laboratory: Continuous Flow Photopinacol Coupling *J. Chem. Educ*. **2020**, 97, 2, 586–591
9) Penny M, Hilton S. 3D Printed Reactors and Kessil Lamp Holders for Flow Photochemistry: Design and System Standardization . ChemRxiv. Cambridge: Cambridge Open Engage; **2021**.