B152B Bioprinter Manual

Software and Hardware operation and overview

Diogo Jahchan Koike – diogo_jk@usp.br
PONTES PROGRAM - PURDUE UNIVERSITY

Contents

Objective

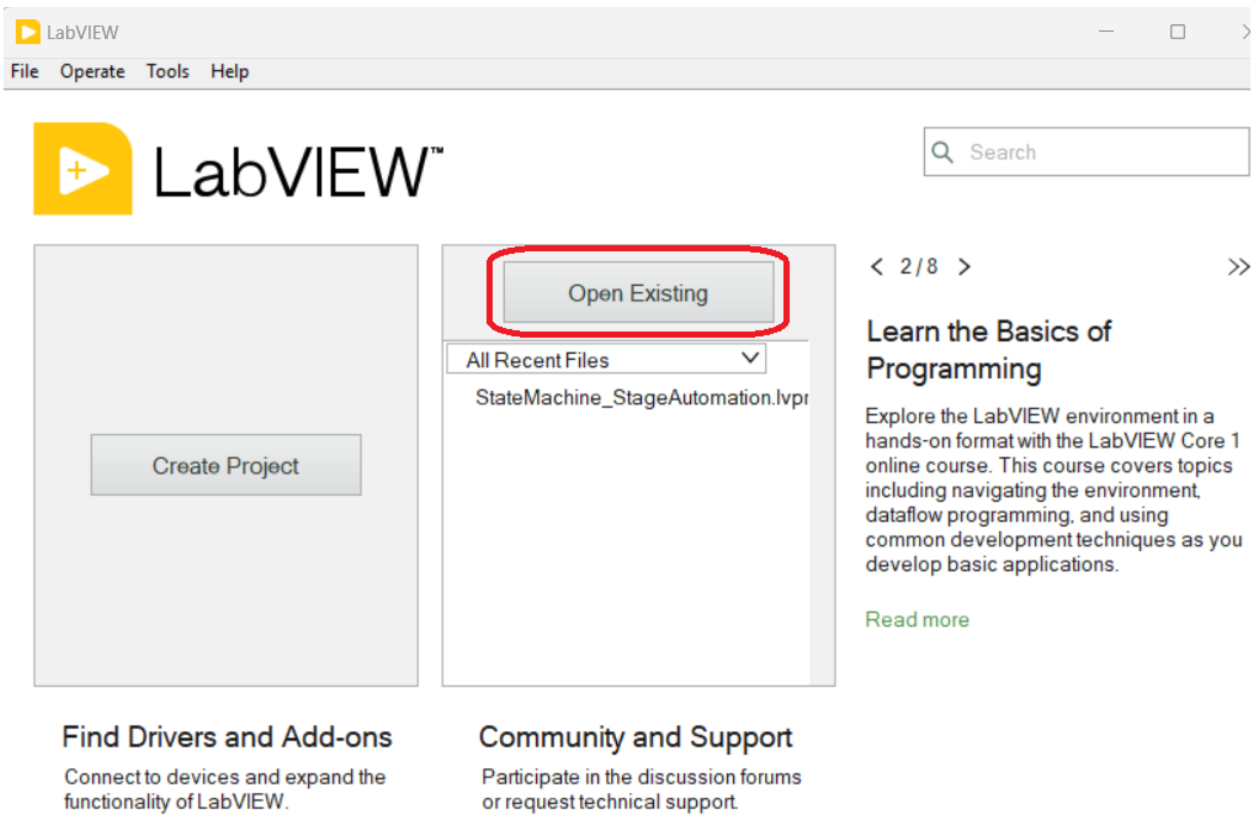
This document has the purpose to guide the use of the LabView Software, regarding the step-by-step for correct printing of tumoroid cells, as well as show the inner workings of the project – with a detailed view of the LabView code for printing and imaging, and the hardware that is integrated in the project.

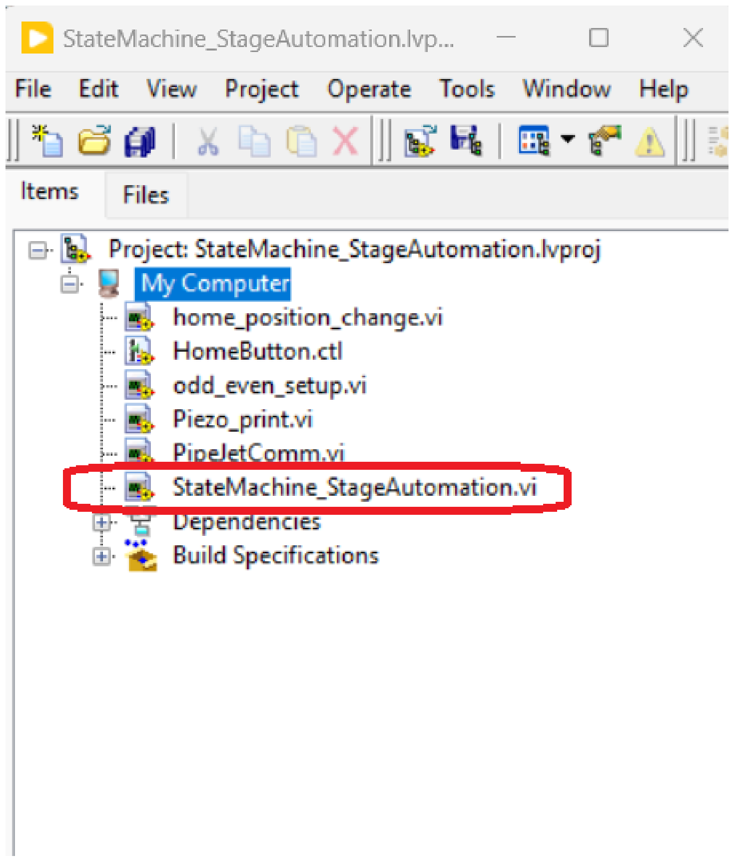
It will also aim to display the current set of problems faced during the conception of the project and give a clear sight of what needs to be fixed and what can be the next steps within the B152B Bioprinter project.

Operation Manual for LabView program

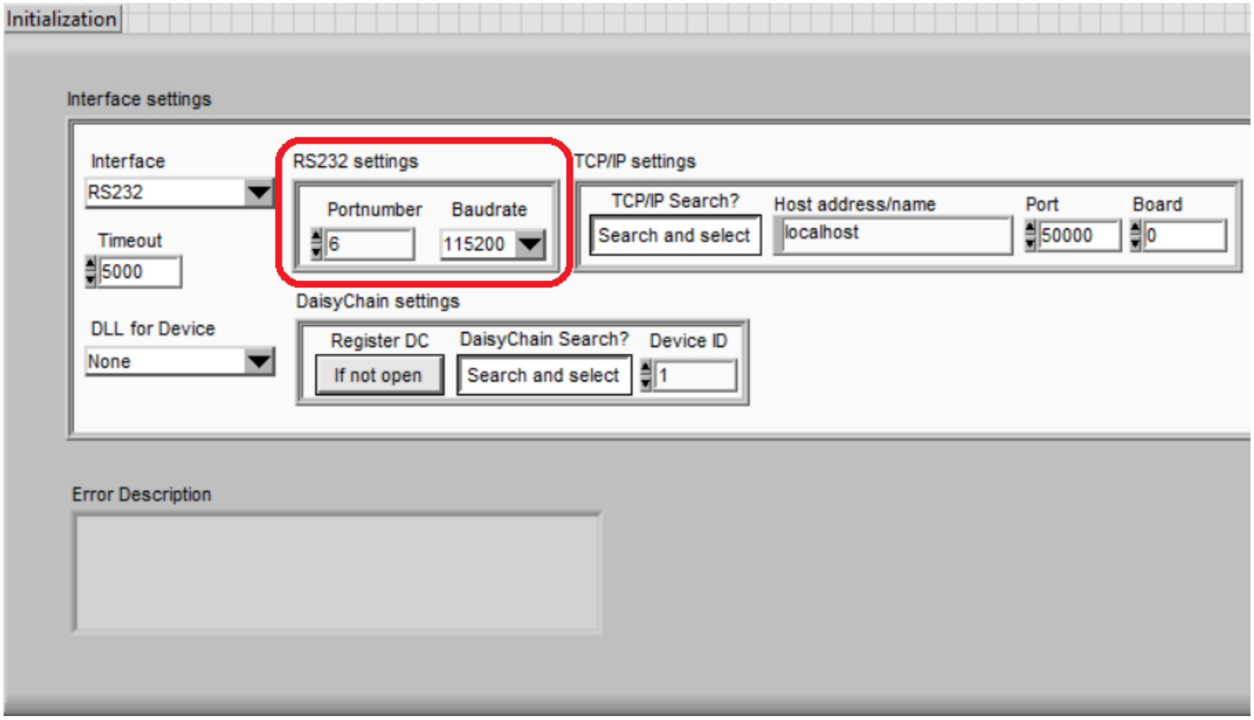
For printing, one must follow these steps – *remember to use LabView 2021 64-bit, use search bar to find it and run it*:

- Open NI LabView 2021 (64-bit)
- Choose to "Open Existing" and open the LabView program VI (titled "StateMachine_StageAutomation.vi") in "Stage_automation" Folder under the "CDFT-Diogo" folder.
    - Alternatively, open the LabView project with the same name and double-click VI.

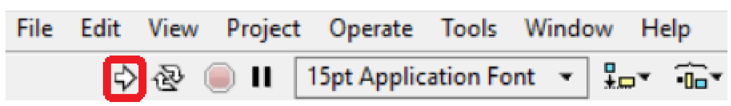




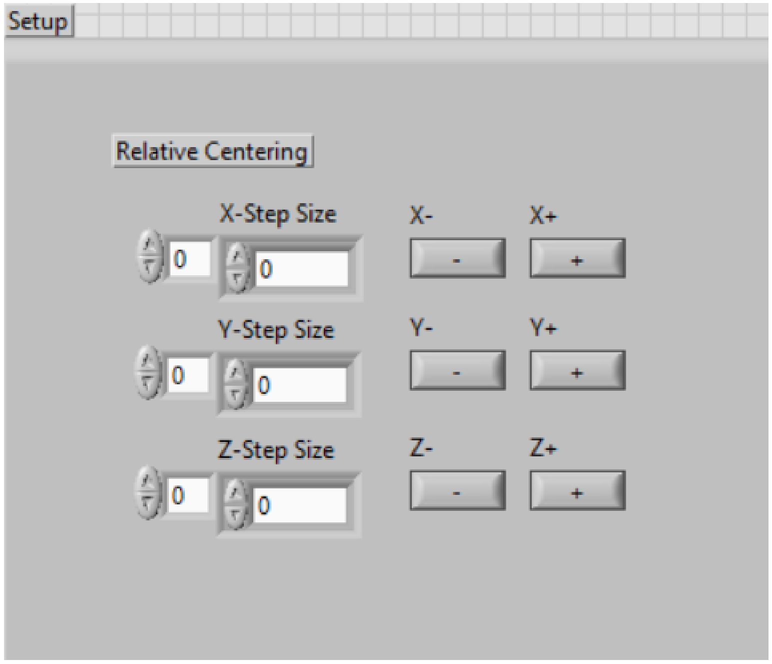
- ***(Skip if already set-up)*** Once successfully opening the LabView program VI, choose the right port for the 3-axis stage controller and set baud rate as 115200 in the "Initialization" tab.
    - To find the correct port, open Device Manager (Start → Control Panel → Hardware and Sound → Device Manager). Look in the Device Manager list, open the category "Ports", and find the matching COM Port.



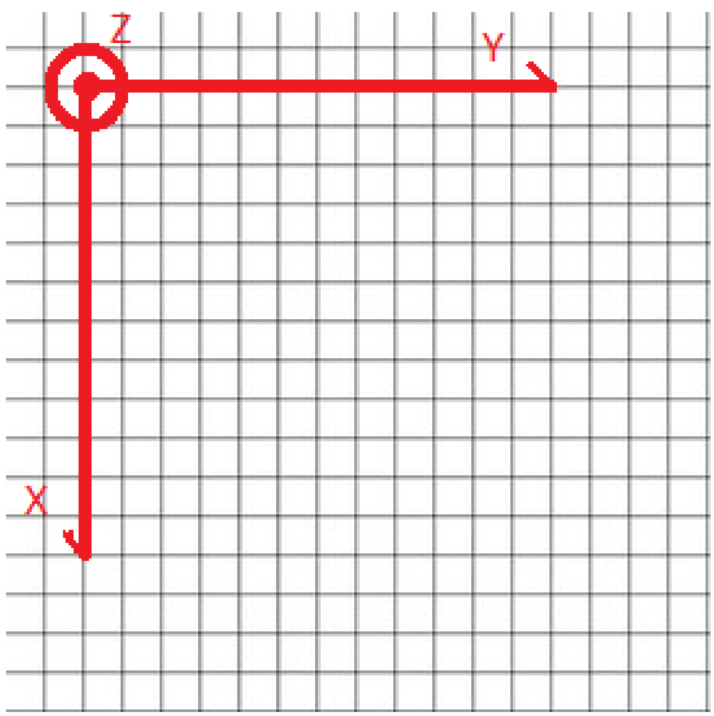
- With the correct initialization settings, press the “Run” button located on the top-right corner of the LabView program.

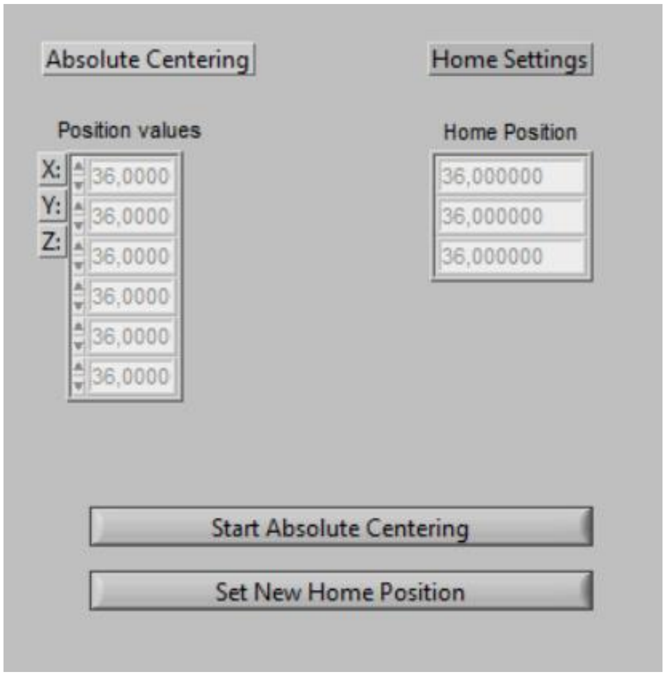


- With the program running - indicated by the “Run” button turning black -, properly move the stage to the desired position. To do that, one can choose to move the table by “steps” or by using absolute position.
    - To move by step, type in the appropriate step for each axis in the “Relative Centering” tab and press the “+” or “-“ button to increase or decrease the position of the axis, respectively.



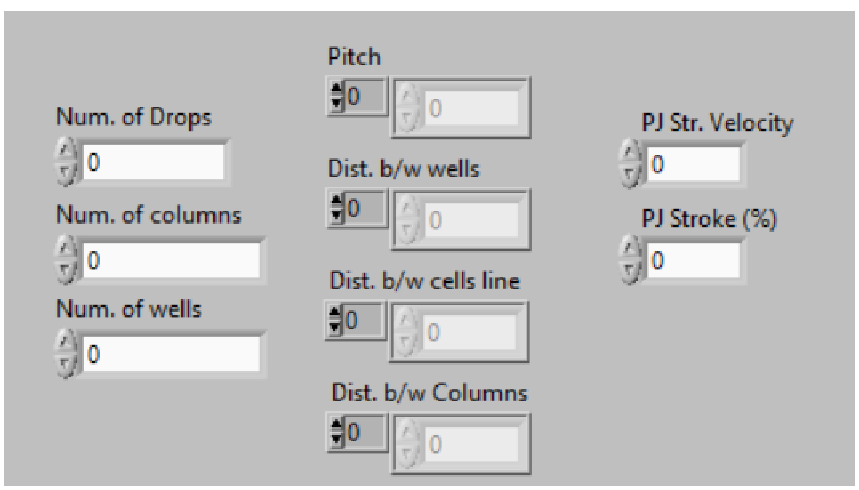
- To move by position, type in the appropriate position for each axis in the "Absolute Centering" tab and press the "Start Absolute Centering" button.
  - *Reminder:* the stage grid for position is shown below as a top-down view. Keep it in mind while setting position values or moving by step.





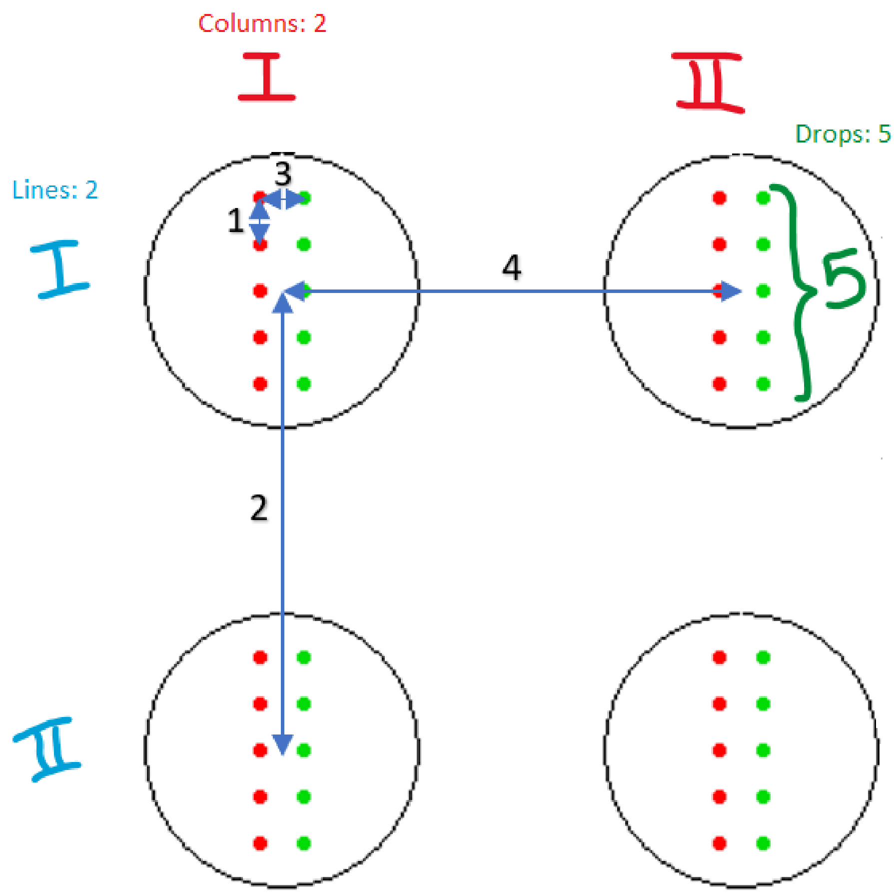
- One can set a new Home position so as to quickly move to a desired position by pressing the “Set New Home Position”, the updated Home position will be then displayed.

- After aligning the tip of the nozzle with the center of the first well, select the appropriate parameters for printing.



- “Num. of Drops”: # of drops in each cell line inside a well (*typ. 5)*.
- “Num. of columns”: # of columns in the well plate.
- “Num. of wells”: # of lines in the well plate.
- “Pitch”: distance between each drop in the same cell line (typ. 0.8).[1]

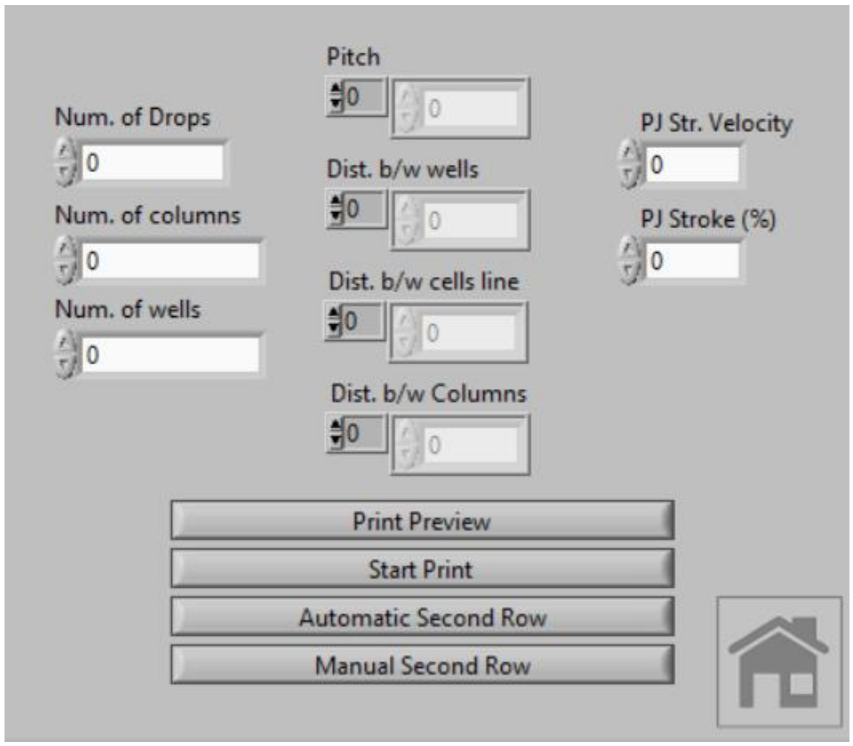
- “Dist. b/w wells”: distance between the centers of a well in two lines.[2]
- "Dist. b/w cells line”: distance between cell lines inside a well (typ. 0.5).[3]
- “Dist. b/w columns”: distance between the centers of a well in two columns.[4]
- “PJ Str. Velocity”: stroke velocity of the PipeJet (typ. 180).
- “PJ Stroke (%)”: stroke length percentage of the PipeJet piezo (typ. 100).



- *Printing starts from the bottom-left well.*

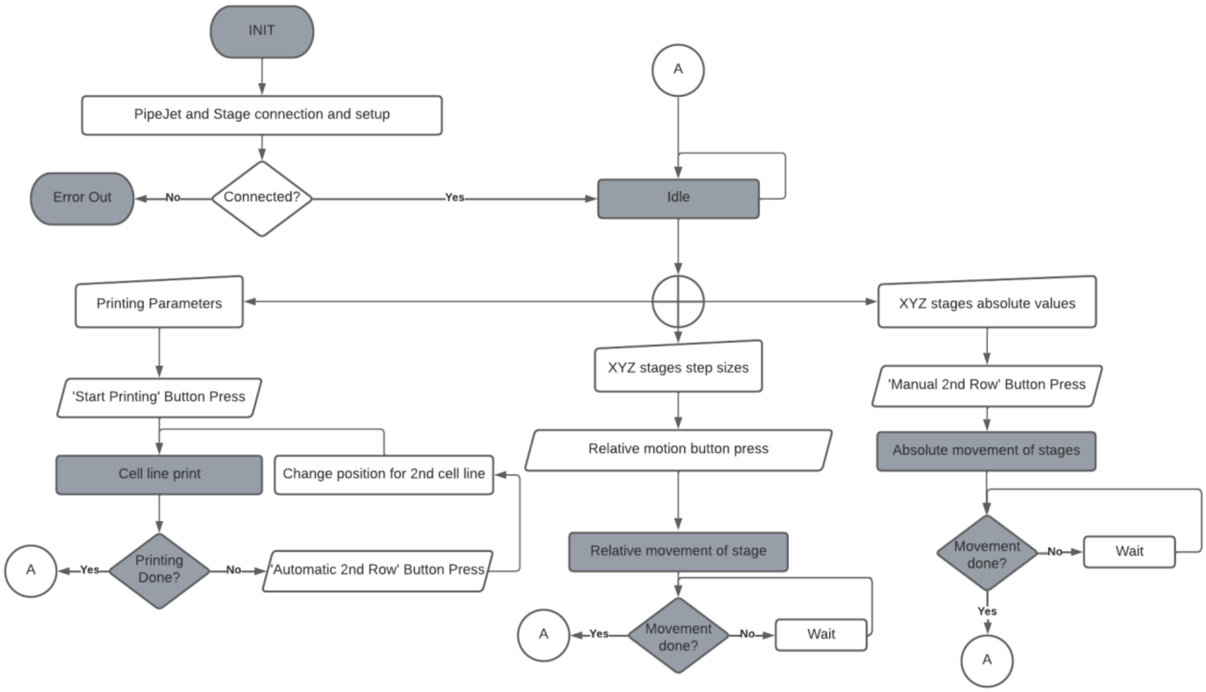
- With the parameters set-up, press the “Start Printing” button. The printer will go over each well and print the first line of cell.
- After that, one can choose between two options for printing: either the "Automatic Second Row” or “Manual Second Row”.
  - “Automatic Second Row" is to be used when there’s no disturbance in the position of the well plate. It will adjust the stage starting position and print the second line of cells. ***Normal operation.***

- "Manual Second Row" is to be used if the well plate was moved during the printing process, losing its starting position and reference. With "Manual Second Row", the user must manually align the cell lines by eye. ***Disturbed operation.***



Software Overview

Printing

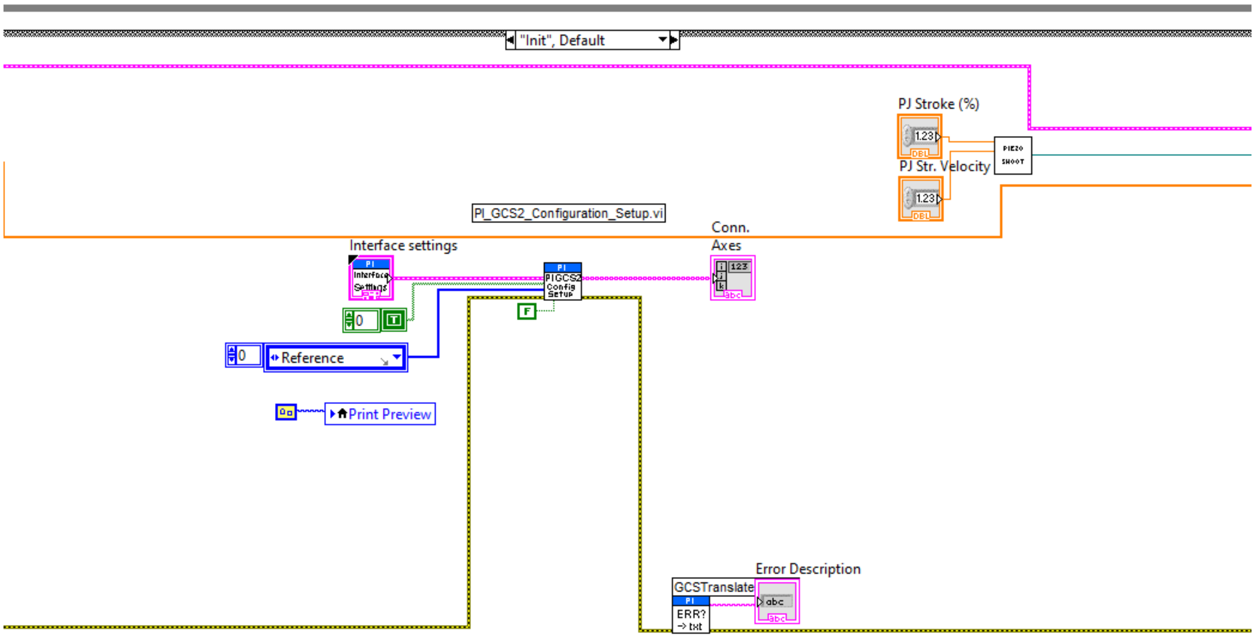


The LabView code for printing is based on an enumerated type state machine with front panel triggers as events and a case structure that will cycle through the states. There are 10 different states that are nested in the case structure:

- “Init” – Sets up PipeJet and Stage communication.
- “Idle” – Idle state that waits for event triggers, has an event structure that is set to read value changes in button presses and change case accordingly.
- “AbsoluteCentering” – Goes to specified absolute position values.
- “RelativeCentering” – Moves specified step size.
- “Home” – Sets Home position.
- “GoHome” – Moves to set Home position.
- “PrintPreview” – 2D picture preview of the printing, with adjustable scale for different well diameters.
- “SingleHead” – Prints line of cells in each well. Starts from the center of the first well.
- “2ndRow” – Shifts starting position of the ink head for the second line of cells.
- “2ndPrint” – Prints line of cells in each well. Starts from operator’s chosen position.

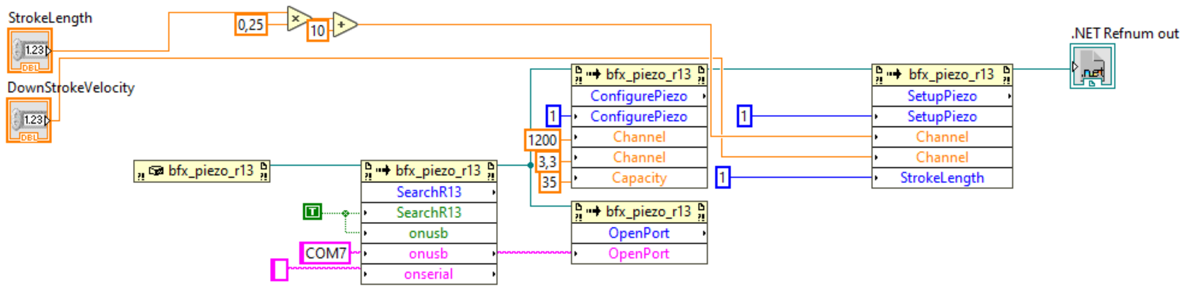
Throughout the code, it is also used shift registers to convey data within cases. One of these instances is in the "Idle" state, where we bundle all user double inputs to later use the printing parameters in the printing states – this helps to format a cleaner code.

Init



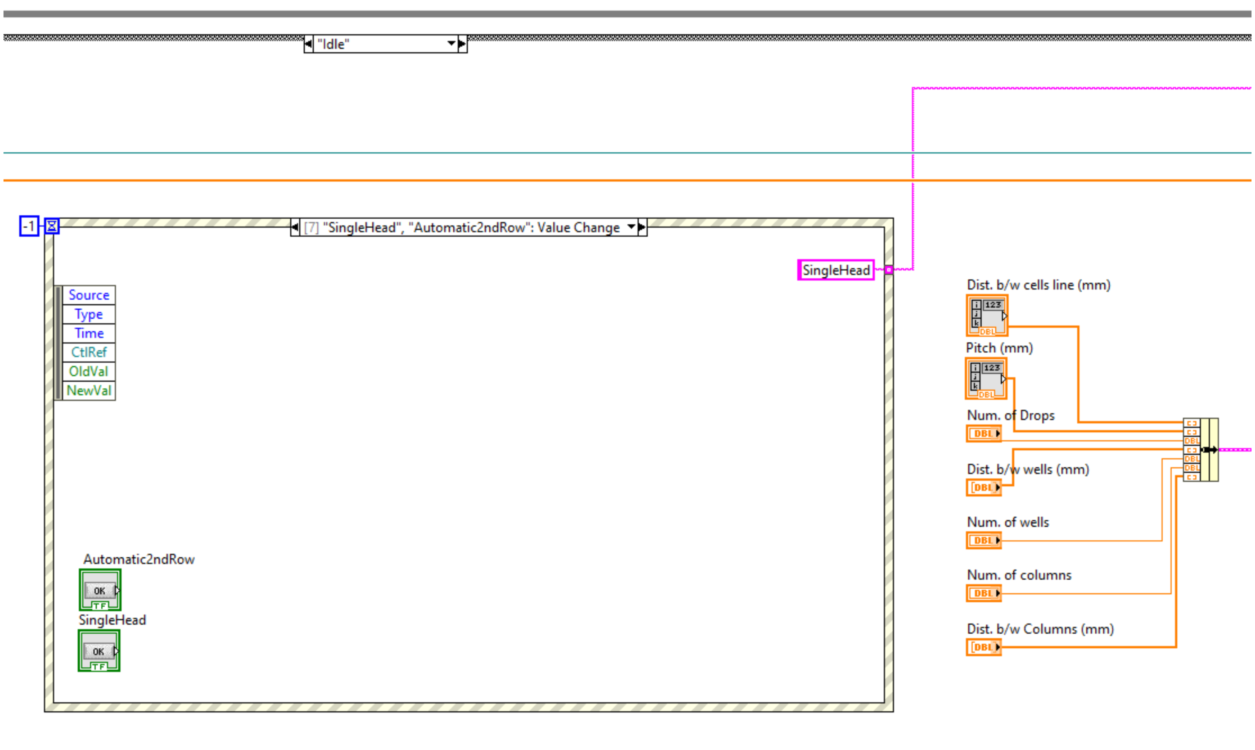
The Init state uses Vis from PI (stages manufacturer) which can be located under "C:\Users\...\ProgramData\PI\LabVIEW", most importantly it uses the "PI_GCS2_Configuration_Setup.vi" which can automatically detect and assign an identifier for each axis connected to the C-884.4DC controller, given the correct connected port through RS-232.

For the PipeJet setup, a subVI is created under the name "Piezo Shoot"; it contains the .NET library used to communicate with the ElectroniX 200 piezo controller, named "bfx_piezo_r13.dll" – *further information regarding the library can be found in the GitHub page.* Special functions are called within the library to successfully set the printing parameters of the PipeJet as well as allow communication with it.



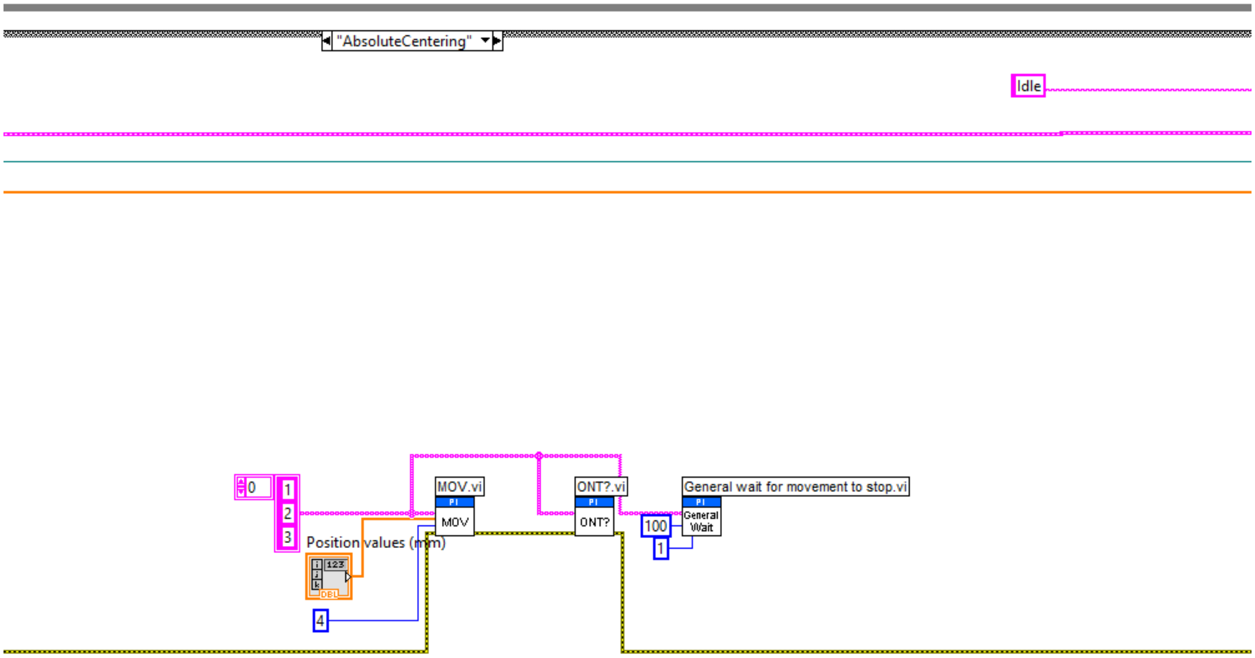
The Piezo Shoot subVI outputs a reference session number that is used to call further functions without losing its reference or creating another reference for each subsequent function call – such as the shoot function.

Idle



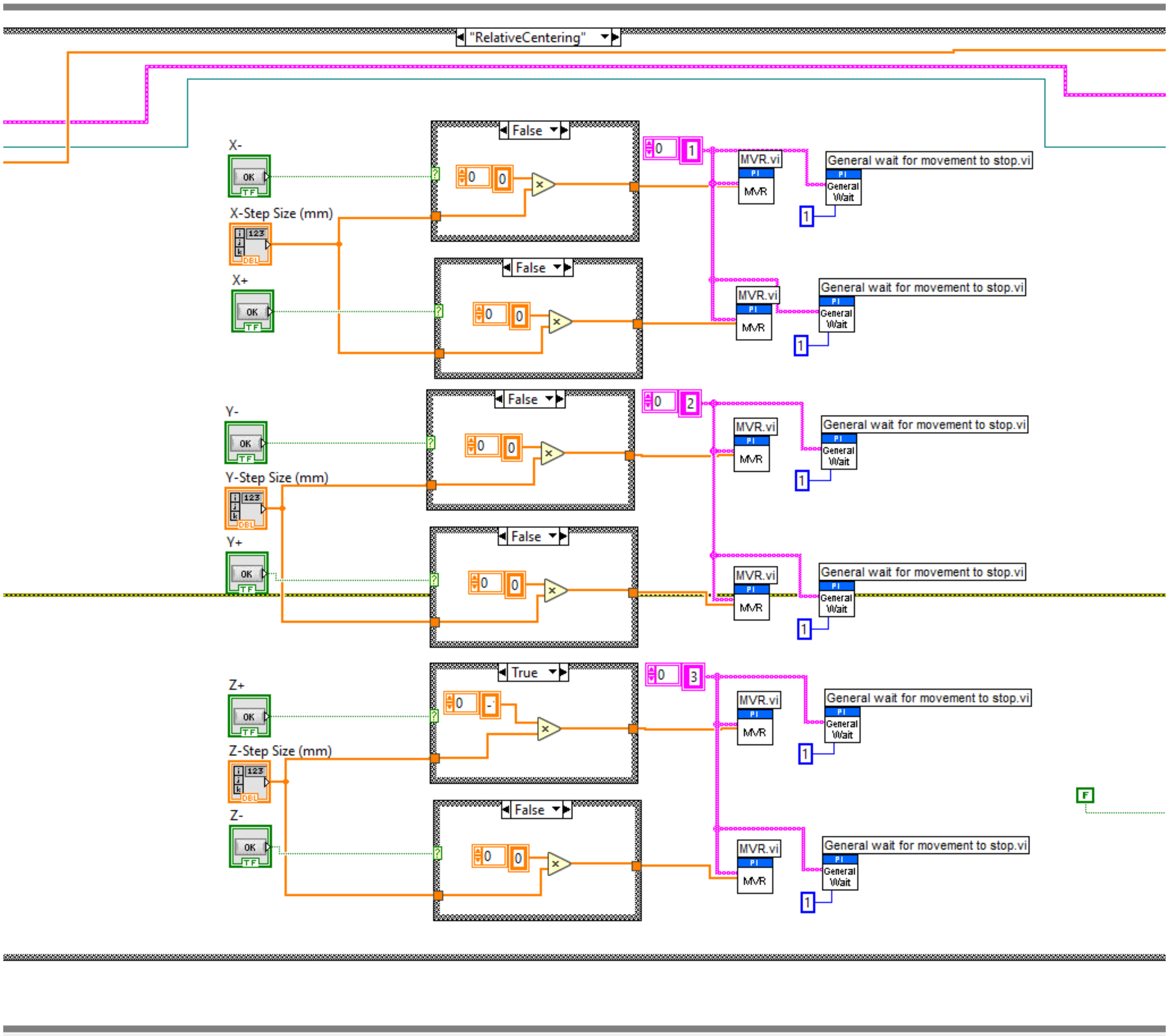
In the Idle state, we have an event structure responsible for detecting event triggers – in our case, a value change caused by a button press – that will accordingly change the value in the case shift register and in turn will cause the case structure to go the resulting state. In this state, we also bundle the user input to facilitate data exchange between states.

Absolute Centering



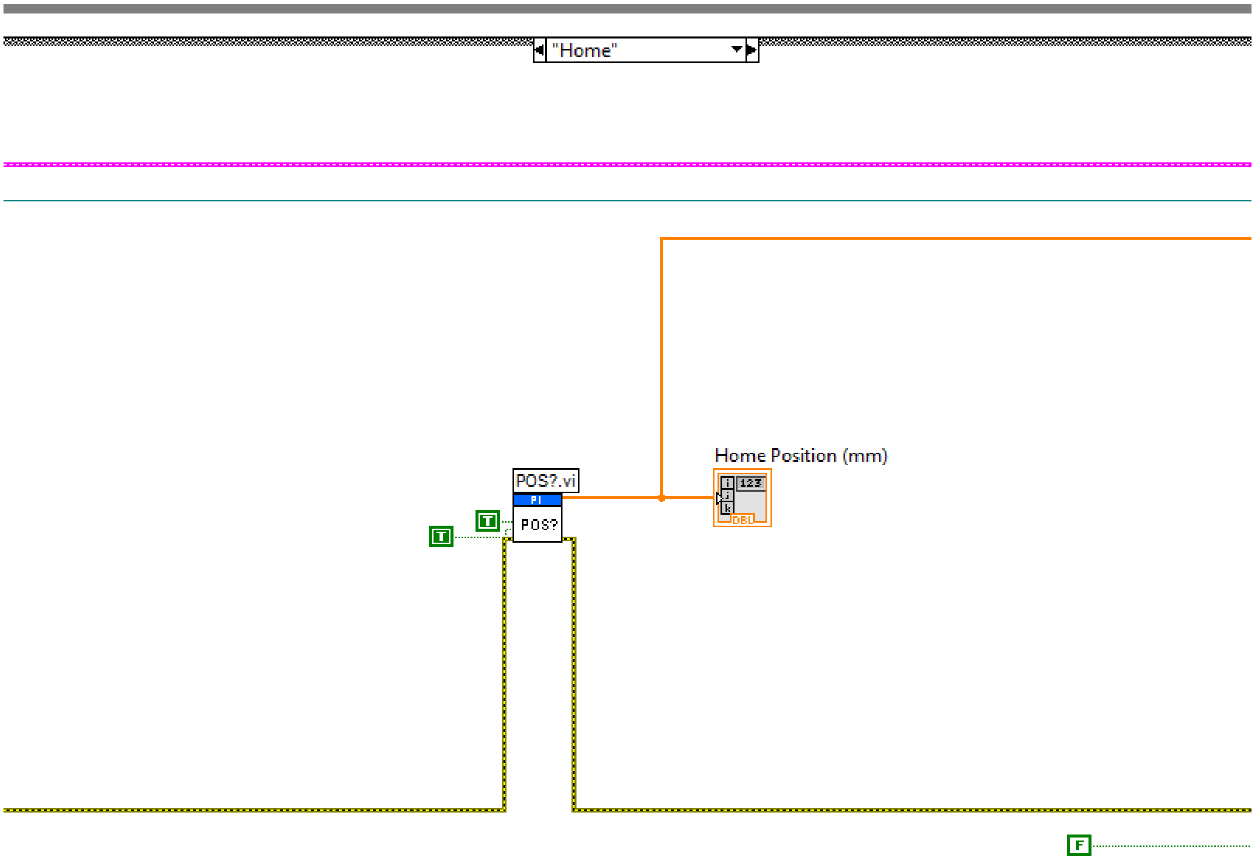
In the Absolute Centering state, the program takes a user input of absolute positions and uses the MOV and ONT? VIs, the former is used to move the stage to the specified position and the latter is used to check if the stage is in the correct position. A general wait function is also called to wait for movement to stop before proceeding.

Relative Centering



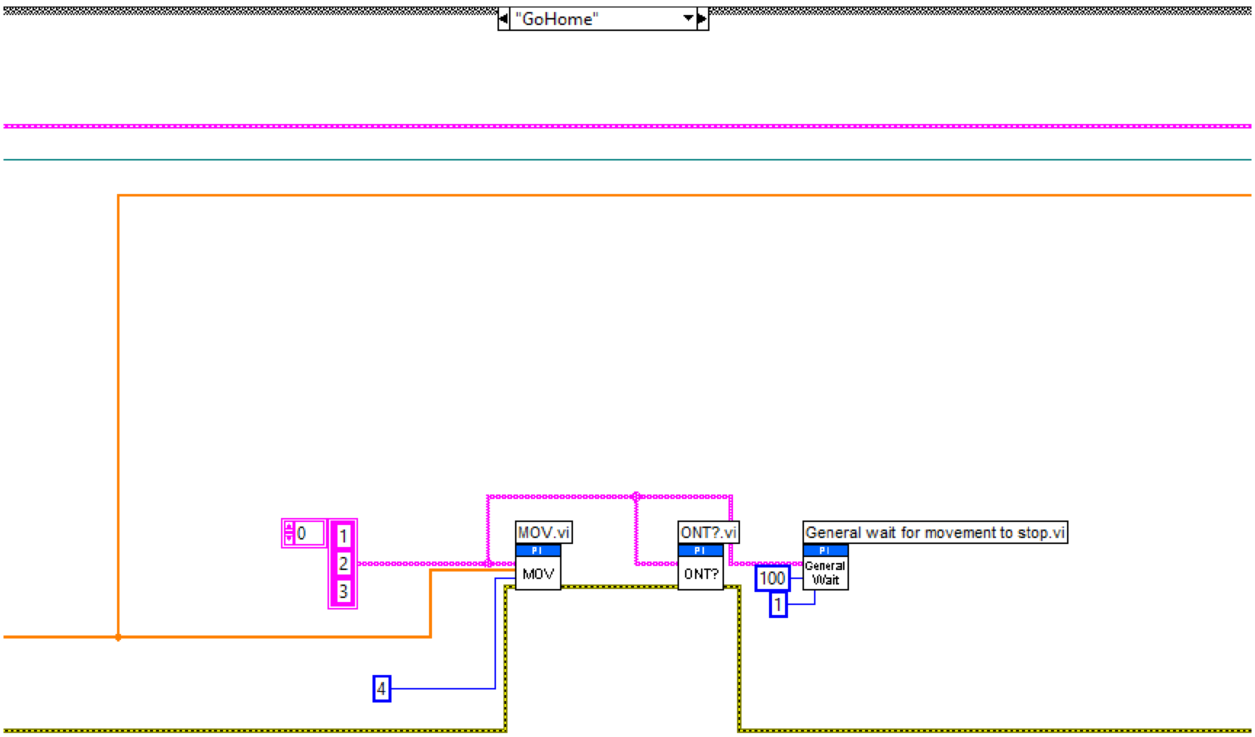
In the Relative Centering state, the program takes the input step size for each axis and uses the user's button presses as conditions for the MVR VI, which moves the axis by the step-size amount.

Home



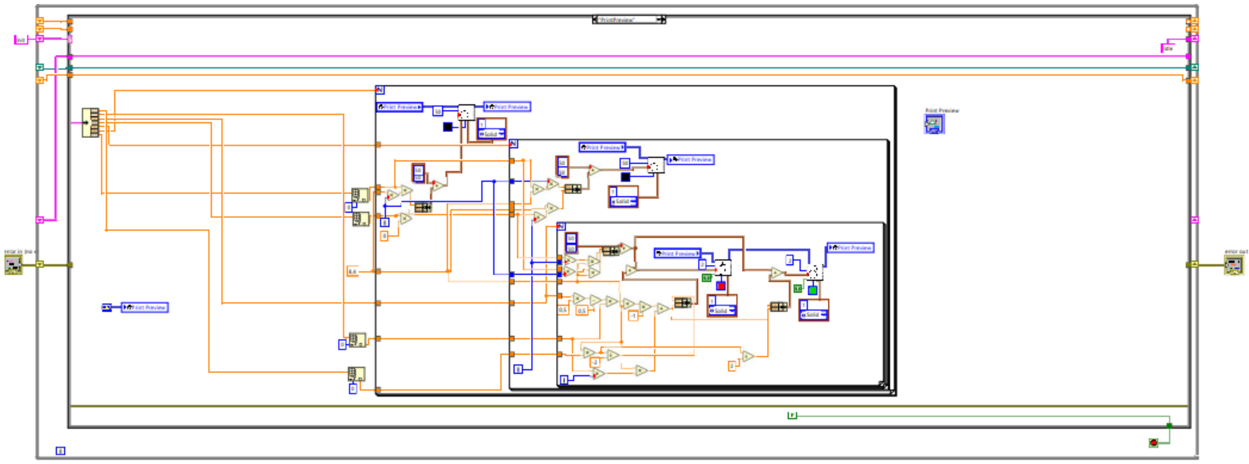
The Home state is responsible for calling the POS? VI – which outputs an array of all axis position values, then showing and storing the current stage position in a shift register.

Go Home



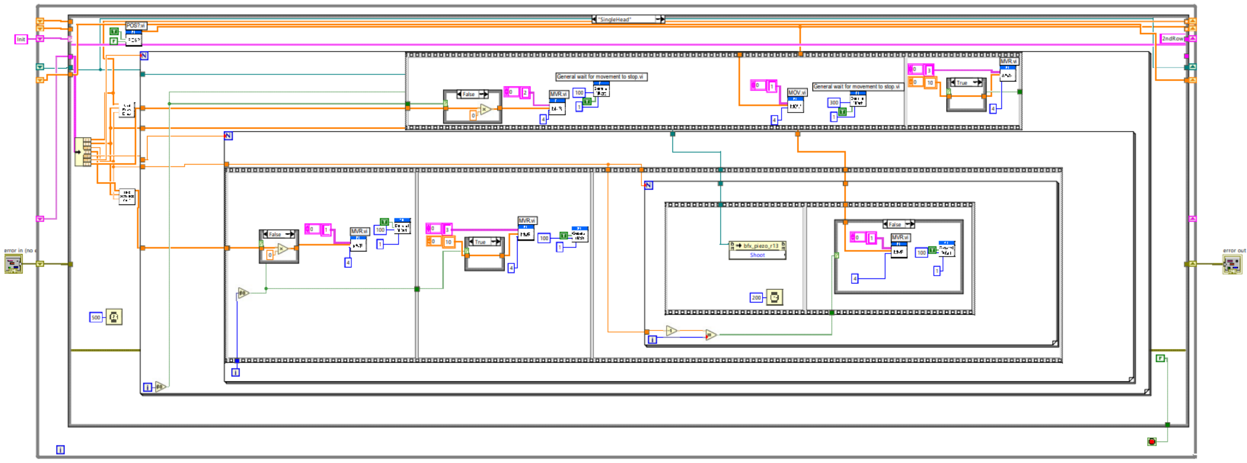
The Go Home state receives the set Home position array as input and moves the stage to it using the MOV VI.

Print Preview



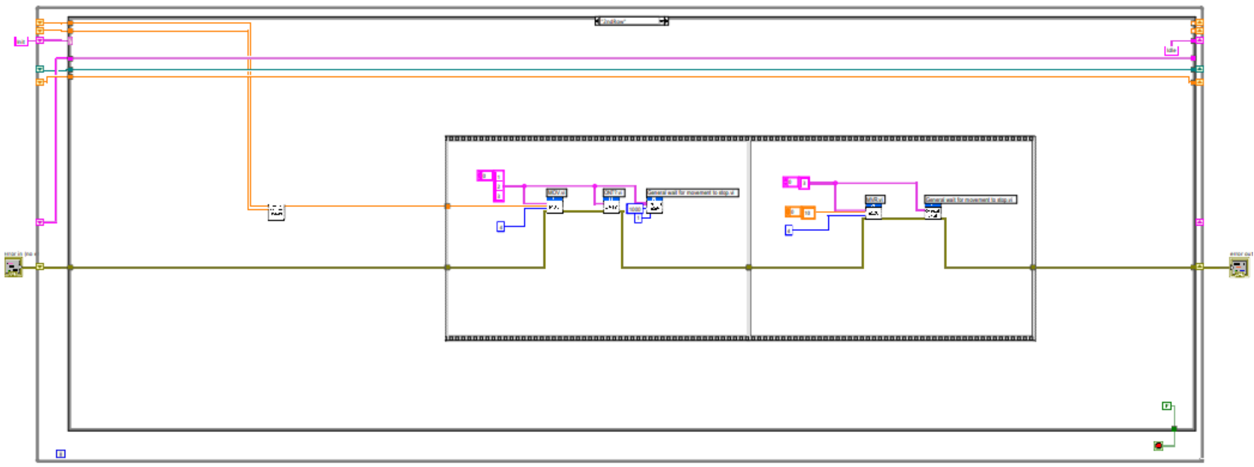
The Print Preview state receives the bundled printing parameters and uses it for drawing the 2D picture of the printing. It has adjustable scale for different diameter's wells and serves as guideline for the user to confirm the chosen printing parameters.

Single Head



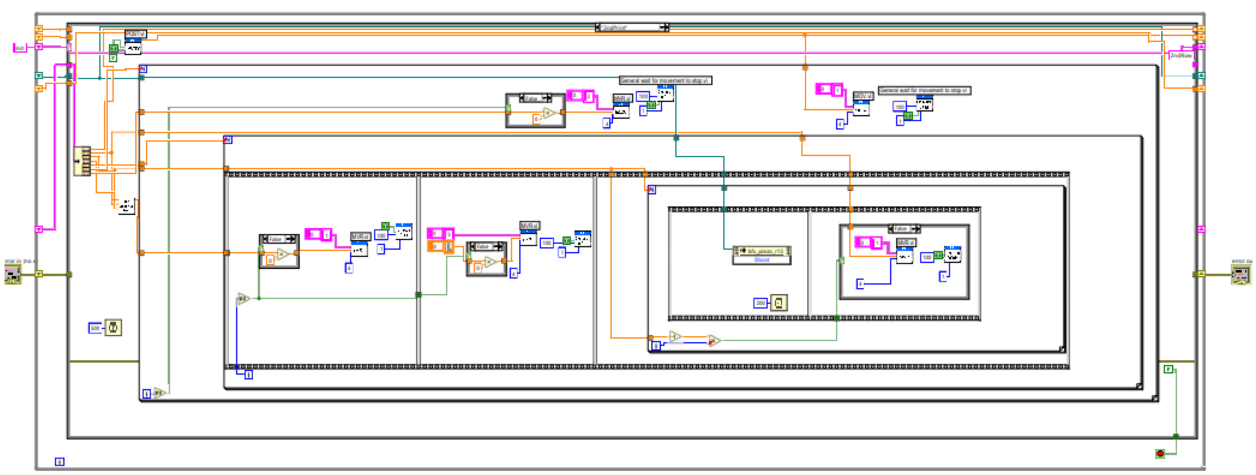
The Single Head state is based on loops according to the printing parameters. We can look at it from an inside-out view: for 'X' number of drops, the stage will move the pitch distance and print each time it stops with the shoot function – at the last interaction it will retract its head to move to another well. For 'Y' number of lines, the stage will move the distance between wells and lower its head to go into the shooting loop. For 'Z' number of columns, the stage will move the distance between columns and go into the line loop. The state also has a 'setup' function, where the user will align the center of the well with the nozzle and the function will move the stage to the correct position for the first line of cells.

2nd Row



The 2nd Row state makes the stage go back to the starting position but changes the Y position to the starting position plus the distance between cells line – virtually creating a new center for the well that is shifted the distance between cells line to the right.

2nd Print

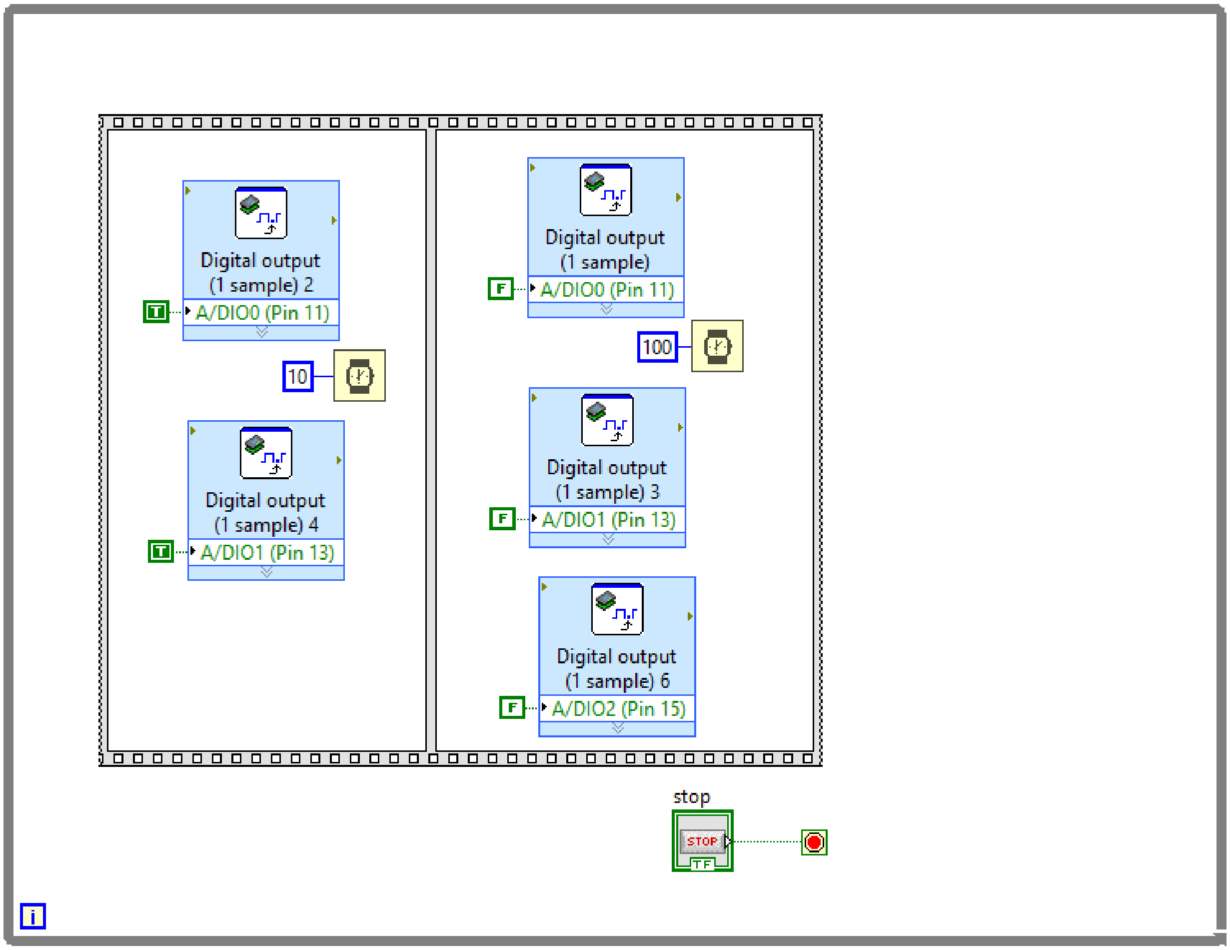


The 2nd Print state is used when the user wants to manually align the second line of cells, it has the same working principle as the Single Head state, however, it does not include the function which moves the stage from its starting position.

Imaging

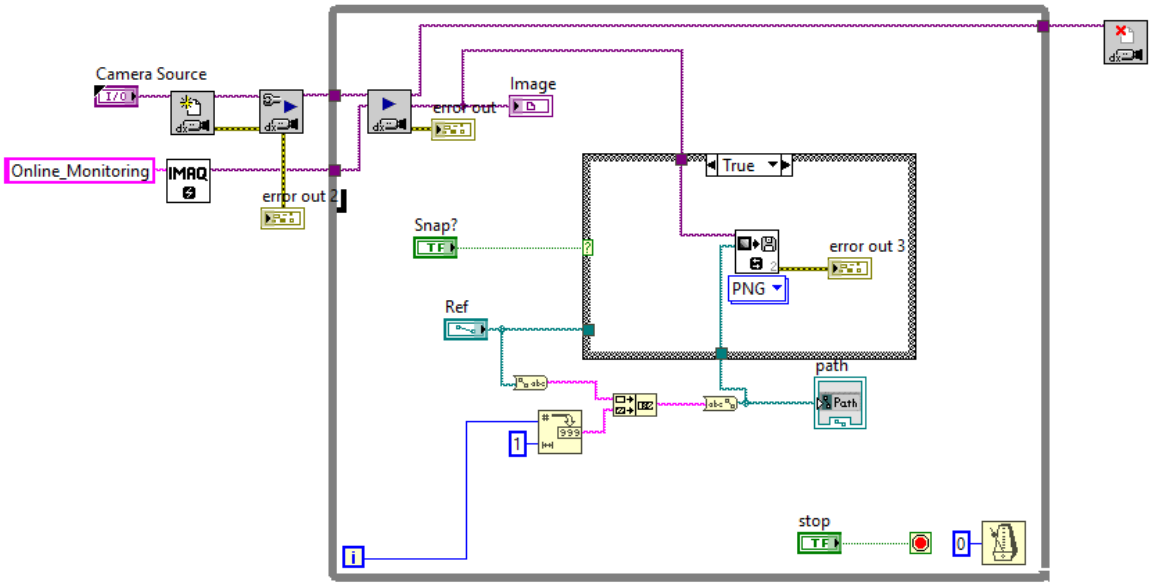
The imaging system code in LabView is responsible for the hardware triggers and images capture and is deployed to our myRio-1900 controller. The system is spread into 2 different VIs, “ImagingSystem.vi” and “ImageCapture.vi”.

ImagingSystem.vi



This VI purpose is to simulate pulse square waves with fixed pulse length. The two outputs are for the strobe LED and the camera trigger (LED lights up and camera takes a picture), this VI must be run for the camera to not be timed-out – as it is set in TTL trigger mode.

ImageCapture.vi

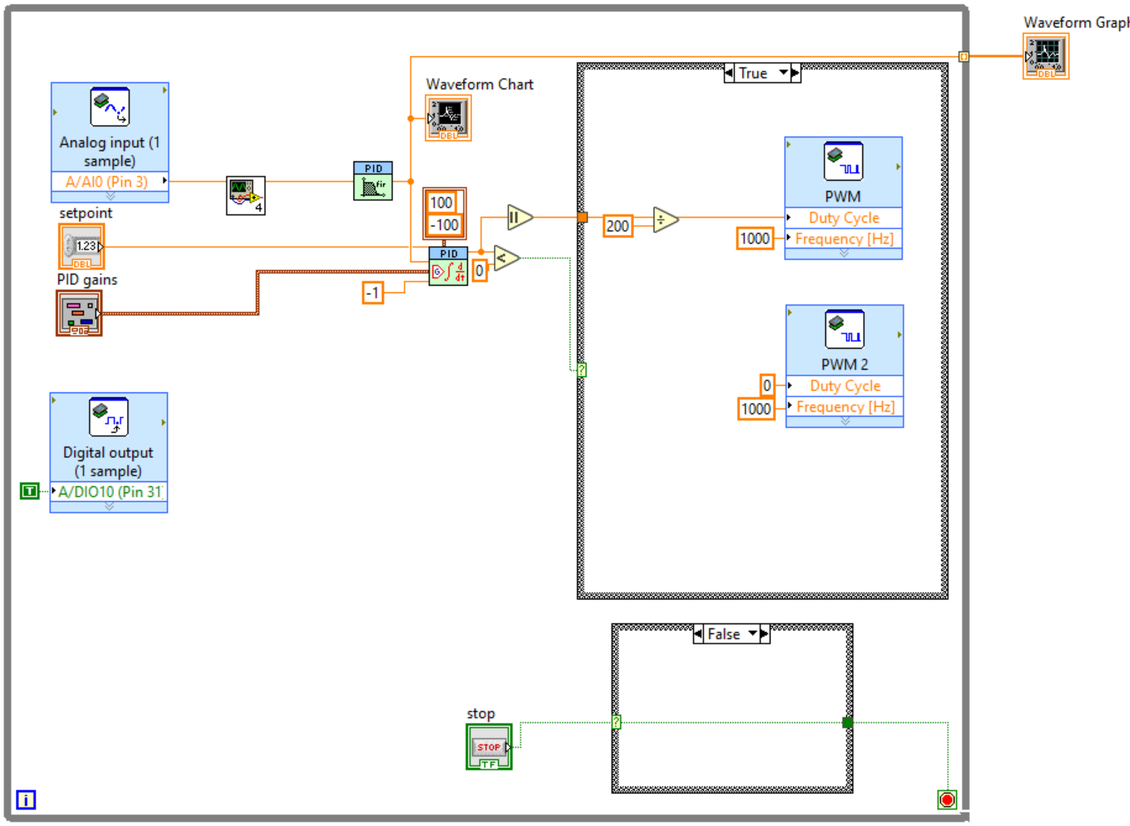


This VI uses IMAQ functions and a compatible camera to show the user the images captured; it also can save those images to a specified file path.

Backpressure control system

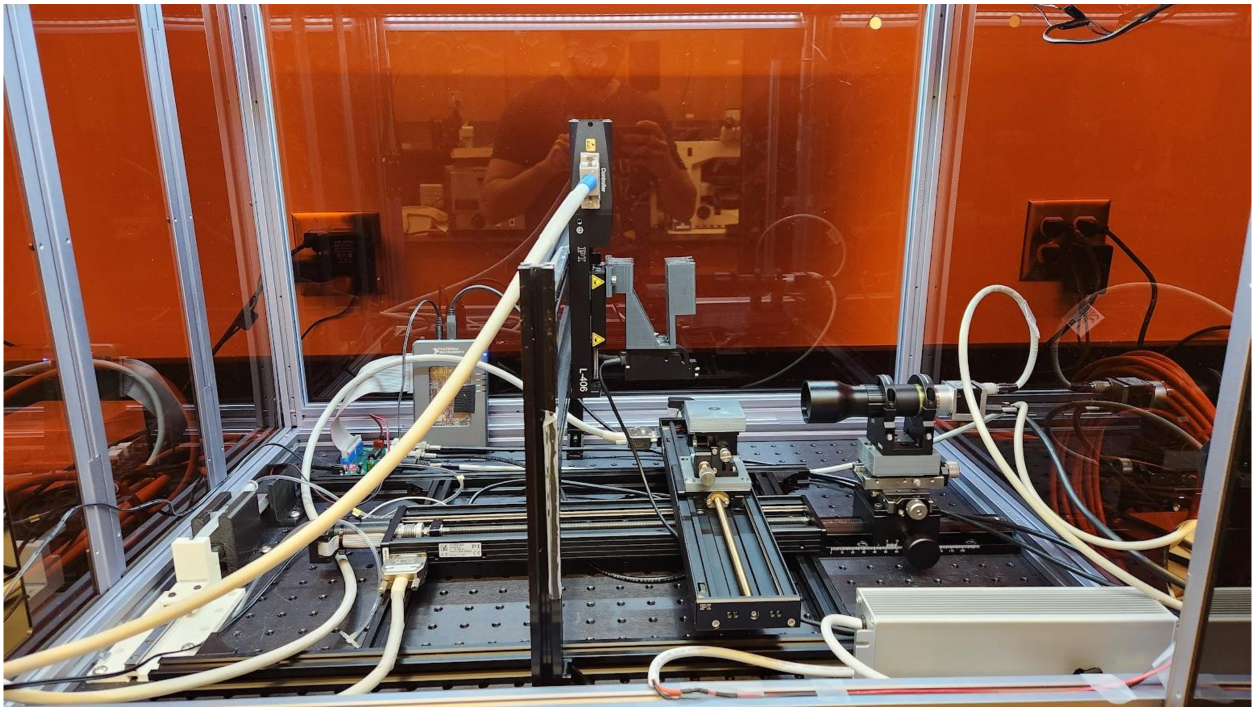
This system is contained within the PID.vi and deployed to the myRio-1900 controller.

PID.vi



The PID VI receives feedback voltage from the pressure sensor, the voltage is then converted to pressure and fed into the PID controller, the output from the PID controller is transformed into a PWM duty cycle that is fed into the DC motor driver – effectively controlling the motor and eventually the pressure. The user chooses the setpoint pressure and the PID parameters.

Hardware Overview



B152B's setup was designed and built during the Jan-May period and comprises of the following systems: printing, imaging and backpressure control. This section aims to showcase all hardware involved in the project.

The whole project is contained within a Snapmaker 2.0 A350 enclosure and sits on a 600x900mm optical aluminum breadboard.

Printing

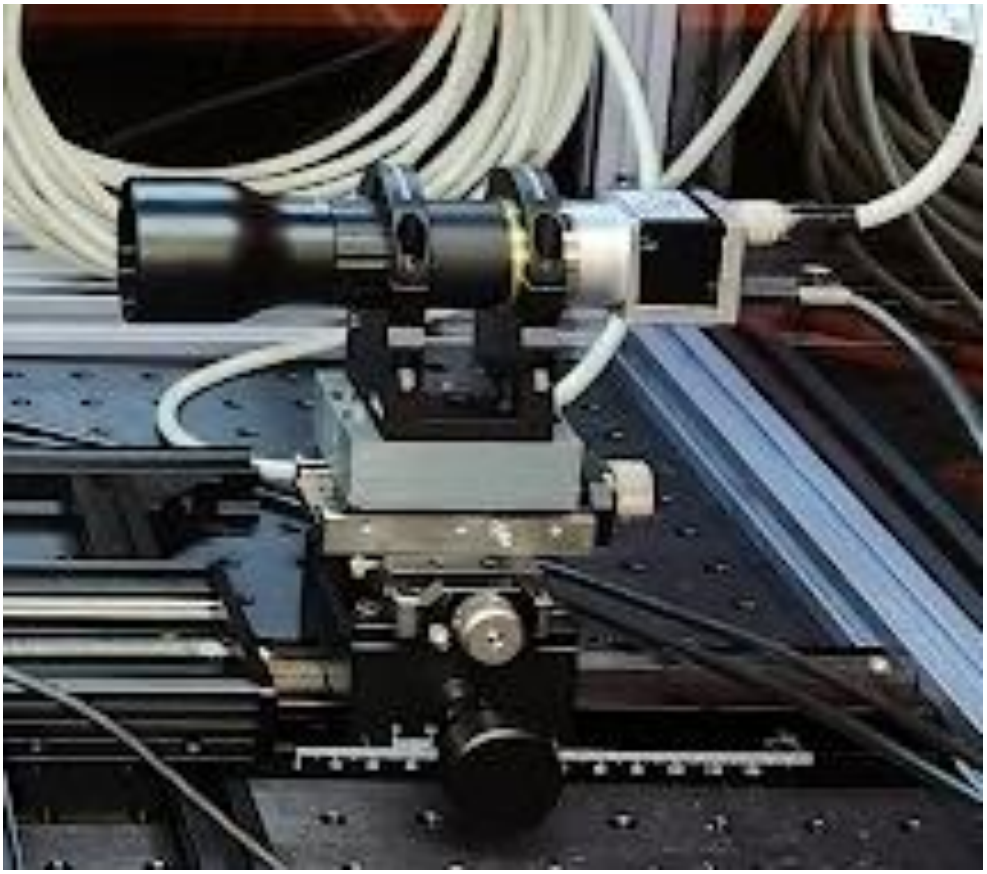
The frame for the stages is built from 2020 aluminum extrusion rails, who connects to each other through aluminum corner brackets. The frame gives a stable base for stage movement, which consists of two VT-80 linear stages for XY-movement and a L-406 linear stage for Z-movement, all of them are controlled by a C-884.4DC 4-axis motion controller.

On the X-stage we have two 3D-printed adapters, one made to fixate the vertical translation stage - for fine adjustment in the table height - into the X-stage and one who is screwed into the vertical stage to act as a table for the printing. On the Z-stage we have a 3D-printed fixture for the PipeJet Nanodispenser.

The PipeJet dispenser is controlled by an Electronix 200 piezo controller and can be activated either through USB communication or through a TTL signal trigger.

Imaging

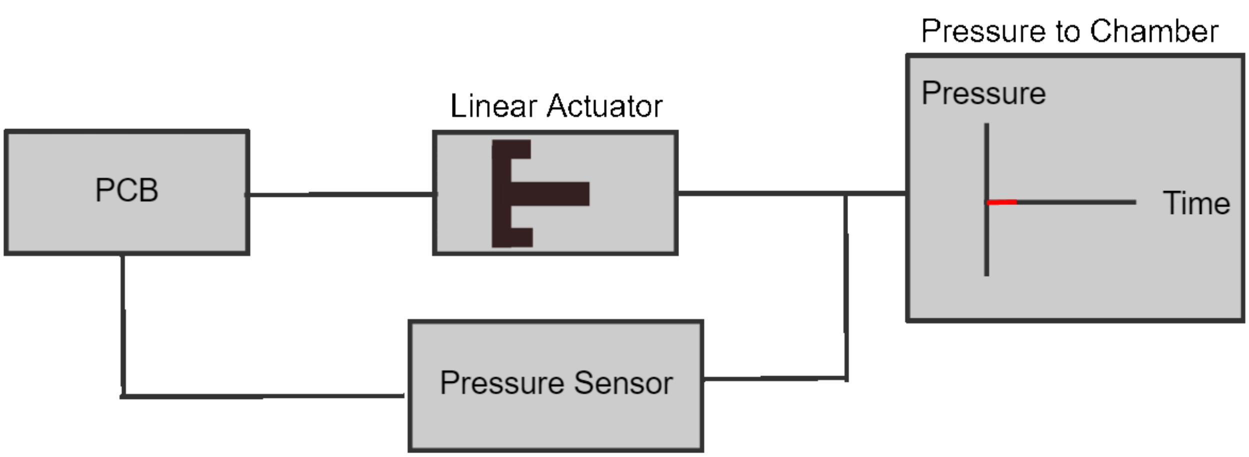
The imaging hardware is composed of a Basler Camera, coupled with a 1.0x Telecentric Lens and mounted to a Clamp mount. The clamp mount is fixated into a 3D-printed adapter and screwed to a fine manual XY-stage for image centering with the nozzle; the XY-stage is then mounted on to a manual one knob stage and track which is screwed to the breadboard.

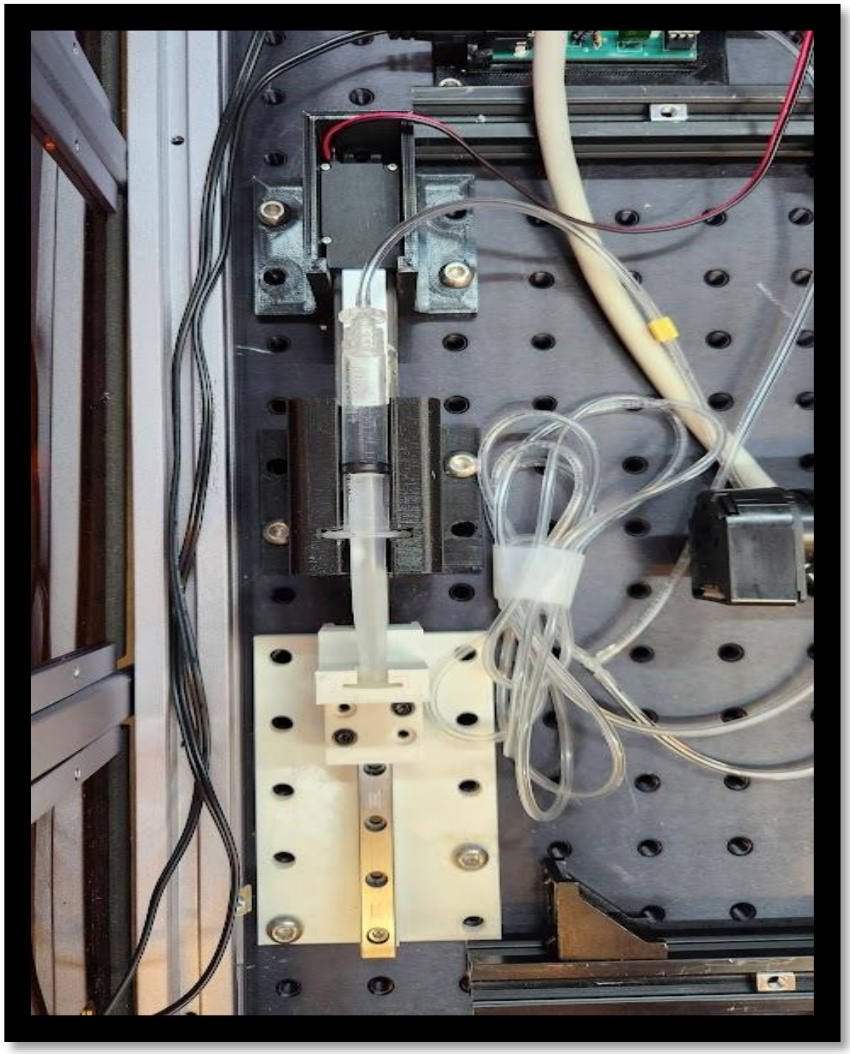


The camera is powered by a USB 3.0 Micro-B to type-A cable and is sent trigger signals through a GPIO cable for the camera.

Backpressure control

The Backpressure is created by the motion of the 6V linear actuator pushing or pulling the syringe into the pressure chamber. It is then controlled by the PCB who drives the linear actuator and who is fed the feedback pressure from the pressure sensor.

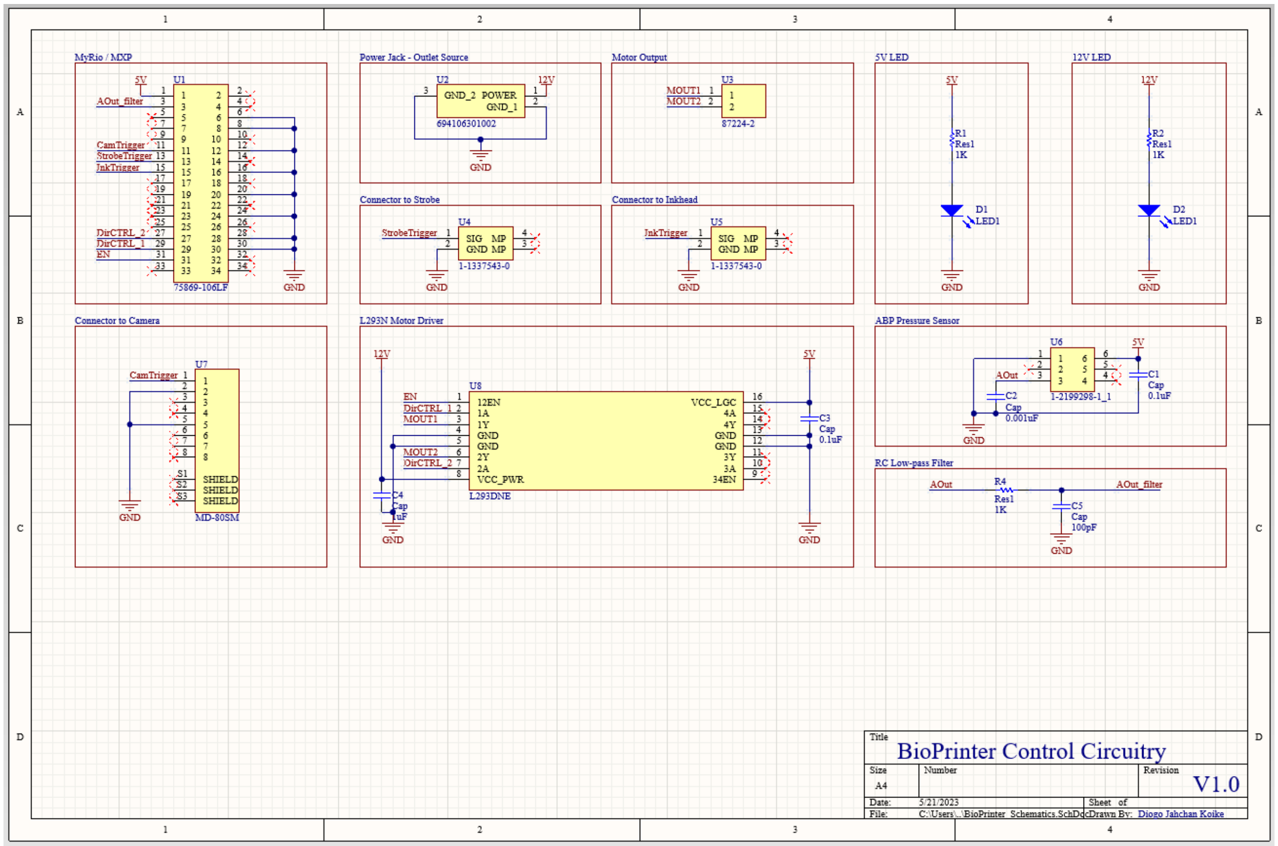


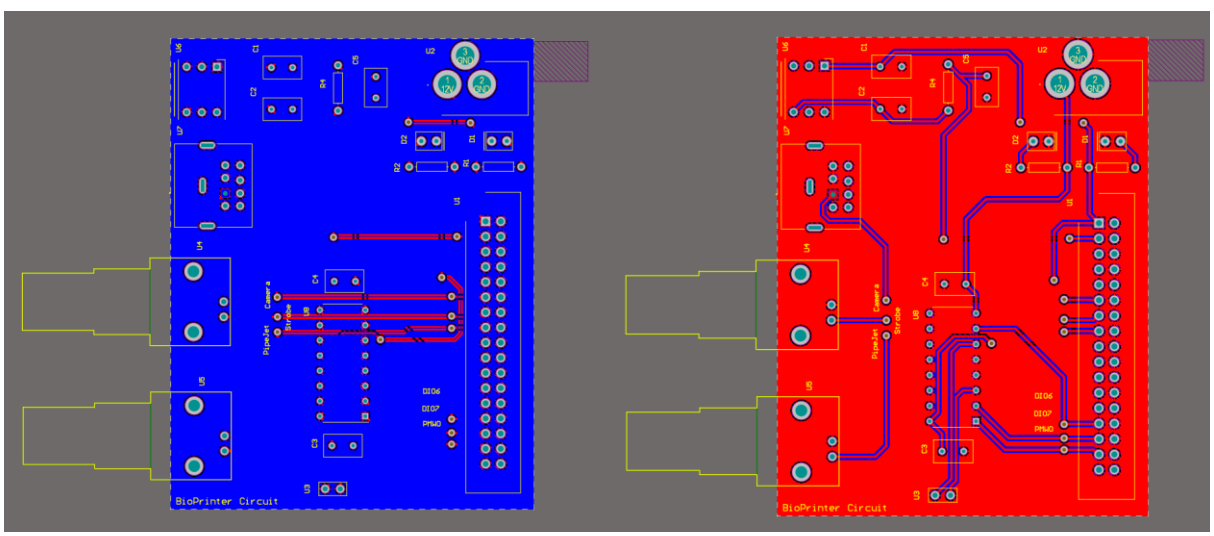


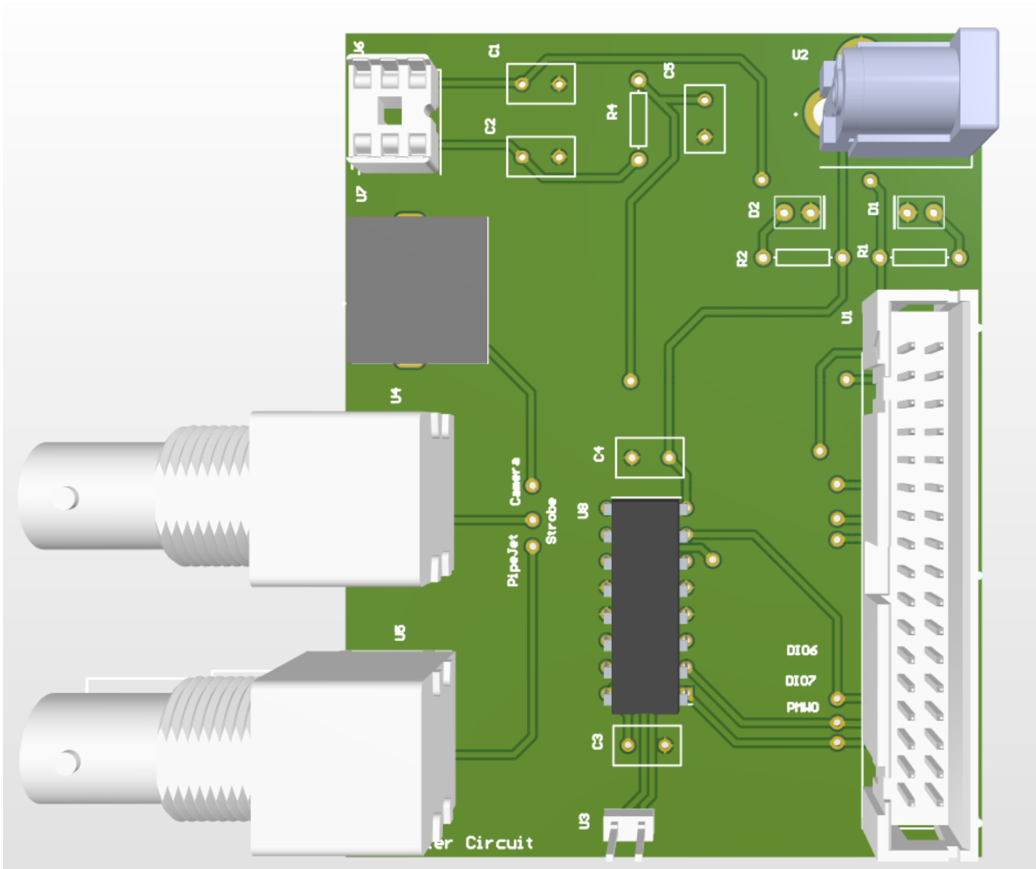
The motor system is powered by a 12V adapter who drives the motor and a 7.5V adapter powers the myRio-1900 that powers the PCB logic-level system.

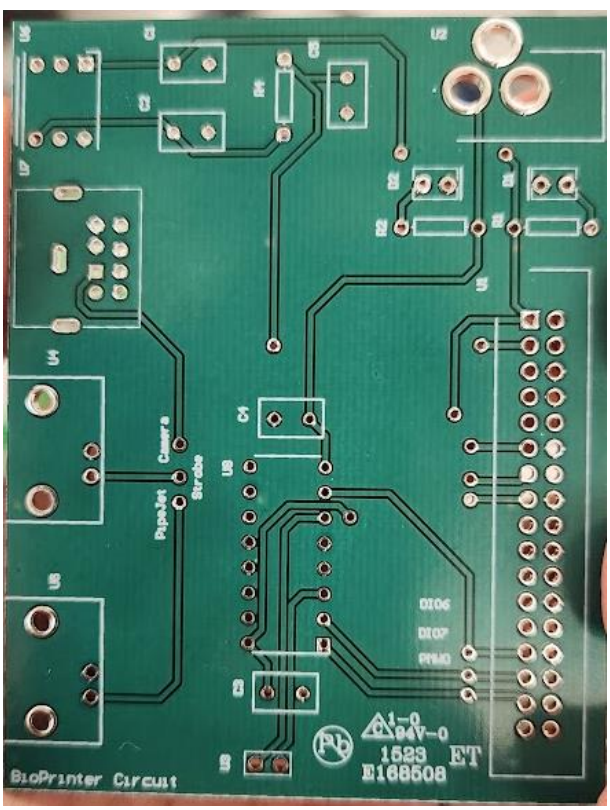
PCB

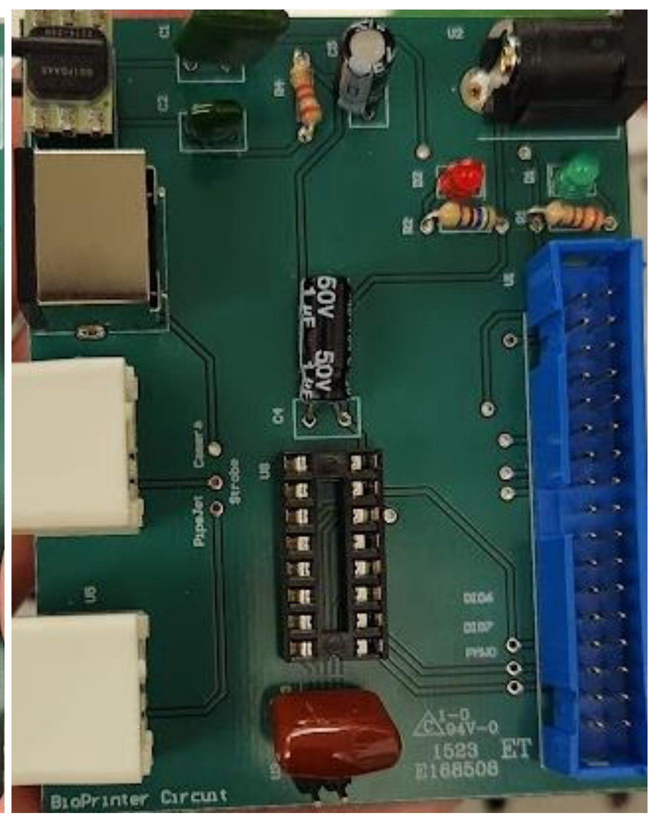
The PCB was designed in Altium Designer and manufactured by Delta Circuits. It features 4 outputs: triggers for the camera, PipeJet, strobe LED and the linear actuator output. It is divided into a logic-level 5V circuitry powered by MyRio and a power-level 12V circuitry powered by an external adapter. The logic circuitry consists of the pressure sensor, RC filter and the enable signals for the motor driver IC.

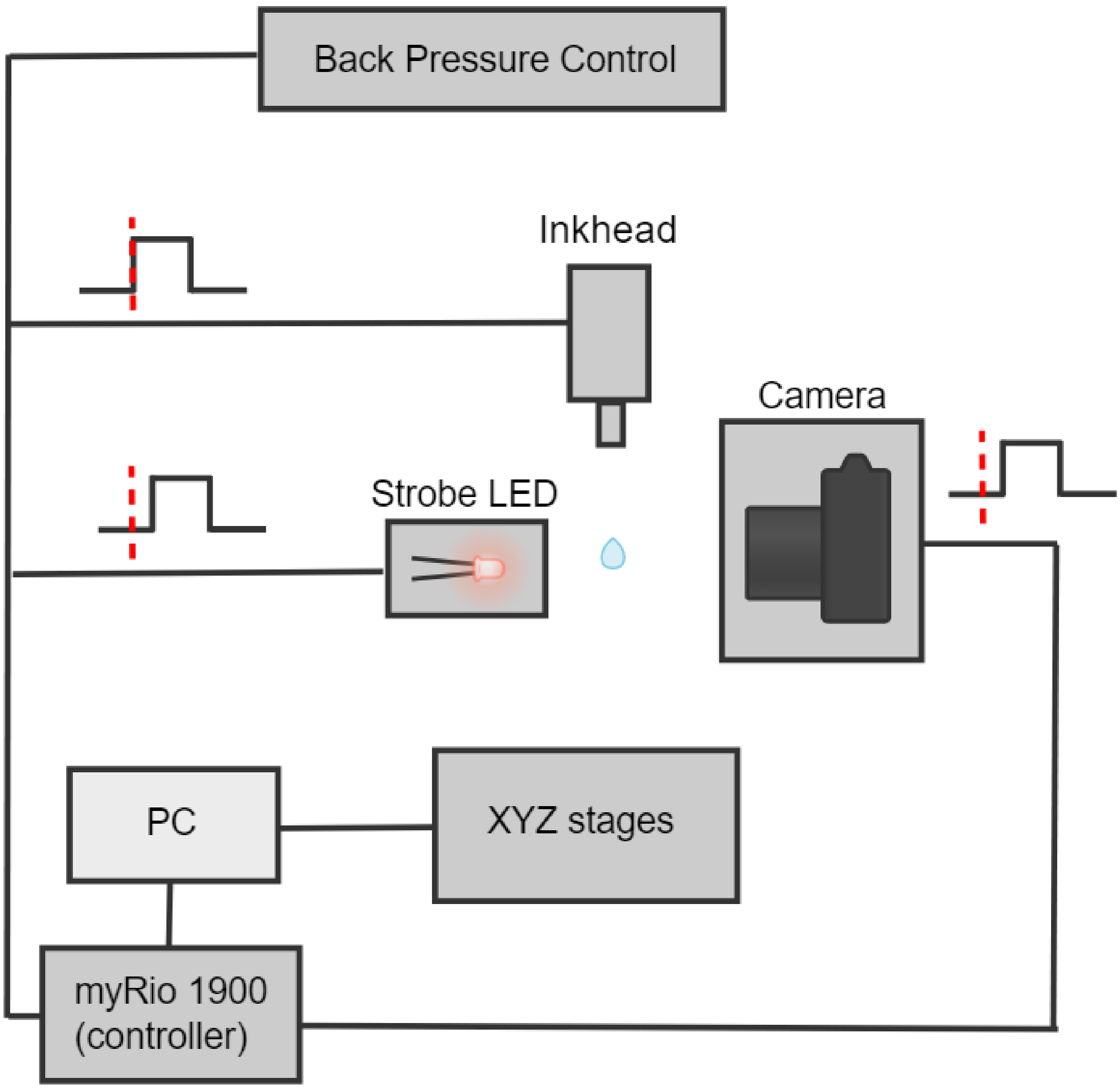

MyRio / MXP
Power Jack - Outlet Source
Motor Output
5V LED
12V LED
Connector to Strobe
Connector to Inkhead
Connector to Camera
L293N Motor Driver
ABP Pressure Sensor
RC Low-pass Filter
75869-106LF
694106301002
87224-2
1-1337543-0
MD-80SM
L293DNE
1-2199298-1_1
Title
BioPrinter Control Circuitry
Size
A4
Number
Revision
V1.0
Date: 5/21/2023
Sheet of
Drawn By: Diogo Jahchan Koike


BioPrinter Circuit
BioPrinter Circuit


U2
C1
C2
C5
R4
D2
D1
R2
R1
U1
U4
U5
U8
U3
C4
C3
Camera
Strobe
PipeJet
DIO6
DIO7
PWM0


BioPrinter Circuit
Camera
Strobe
PipeJet
DIO6
DIO7
PWM0
1523 ET
E168508


BioPrinter Circuit
50V
1523 ET
E168508



Part List

Structural

- 1x Snapmaker 2.0 A350 enclosure.
- 1x Thorlabs 600x900mm aluminum breadboard.
- 8x 2020 aluminum extrusion rails.
- 10x 2020 aluminum corner brackets.
- 1x set of M5 Allen screws.

- 1x set of M6 Allen screws.

Imaging

- 1x Basler ace acA2000-165um camera.
- 1x SilverTL 1.0X Telecentric Lens.
- 1x One Knob Stage & 125mm Track.
- 1x Mounting Clamp, 1.75” Centerline
- 1x Large X-Y Axis Leadscrew Drive.
- 1x generic white LED.
- 1x 24V LED strip power supply.
- 1x GPIO & Power Cable for Basler ace cameras.
- 1x USB 3.0 Micro-B to Type-A cable.

Printing

- 1x L-406 Compact Linear Stage (Z-Axis).
- 2x 300mm VT-80 Linear Stage (XY-Axis).
- 1x C-884.4DC 4-axis motion controller.
- 1x PipeJet Nanodispenser.
- 1x Control Electronix200.
- 1x 40x40mm Metric Z-Axis stage.

Backpressure control

- 2x Syringe.
- 1x 6V Linear actuator.
- 1x myRio-1900
- 1x BioPrinter Circuitry PCB

PCB

- 5x ceramic capacitors[1]
- 1x Red LED.
- 1x Green LED.
- 3x Resistors[1]
- 1x 75869-106LF Amphenol Connector.
- 1x 694106301002 Power Connector Jack.
- 1x 87224-2 Header Connector 2POS.
- 1x 1-1337543-0 BNC Connector Jack.
- 1x MD-80SM Connector
- 1x L293DNE DC Brushed Motor Driver IC

[1]*Consult PCB schematics for correct values.*

Troubleshooting

Stuck stages

During normal operation, the stages might get stuck at something or accidentally hit their limit switches. In that case, the LabView program won't work as the servos of the stage will be turned off. To turn them on again, open the manufacturer software PIMikroMove, connect to the correct port and choose to do automatic referencing – this will bring the stage back to its zero position and correctly reference the stage once again. With that done., you can close the software and open the LabView program.

Stuck Linear Actuator

After extensive periods of use, the Actuator may get stuck at a certain position, being unable to move it using the motor driver and PWM signals from myRio. In that case, unplug the linear actuator from the PCB header output and apply a constant voltage in the actuator terminals from a bench power supply – the actuator should move and then you can connect the motor terminals to the PCB's headers.

Connection Failed to Stage or PipeJet controller

Make sure the cables are properly connected and the port number is correctly set. If everything is correct, make sure that no other programs are currently using the port, i.e., having both PIMikroMove software and LabView program open at the same time or having the BFX control software and LabView open as well.

Electronix200

There are currently two BFX Control Electronix200 controllers, one of them has installed older firmware which can be successfully connected to the LabView program, however the other with newer firmware is unable to connect to the LabView program.

PCB Design

In the first iteration of the PCB design, there are two mistakes to be corrected: the pins on the "1-1337543-0 BNC Connector Jack" are swapped – signal is where GND is supposed to be – and the pressure sensor pads should be further apart in length as it was mistaken for a generic DIP package (having to crunch the IC legs to fit the DIP socket).

Next Steps

For further iterations of the BioPrinter project, the focus should be on fully integrating the imaging system into the printing system for live drop capture during the printing process. As well as improving the printing velocity, so that – for a large quantity of wells – the printing is done before the drops dry out; this should be achieved by improving the code for printing and possibly using a different strategy for stage movement.

Quality-of-life changes should be something to aim for; for example, having a self-contained program that can reference the stages itself without relying on external software from the manufacturer (refer to the PI LabView Vis manual for Vis that can provide these functions).

Inquiries should be made to the BFX Electronix200 manufacturer, Biofluidix, as to the compatibility issues we have with the newer firmware Electronix200. After that, the controller should be put in Trigger mode, and eventually we aim for it to be fully controllable by the PCB trigger signals.

Lastly, the mistakes on the PCB should be corrected and reviewed by a peer before production. All documentation, code and designs can be found under the GitHub project 'B152B' in the following link: https://github.com/DiogoJKoike/B152B

Special Thanks

I would like to extend my heartfelt gratitude to all the individuals who have played a significant role in the successful conclusion of my internship program research. Their unwavering support, guidance, and expertise have been instrumental in shaping my research journey and enriching my overall learning experience.

I am immensely grateful to Dr. George Chiu for his invaluable mentorship throughout my internship program. His profound knowledge, insightful feedback, and unwavering encouragement have been crucial in guiding me towards the successful completion of my research. His expertise in the field has not only inspired me but has also broadened my horizons, fostering a deeper understanding of the subject matter.

I would like to express my sincere appreciation to Natalia Ospina Muñoz for her indispensable assistance and continuous support during my internship program. Her dedication, patience, and expertise have been instrumental in helping me navigate through various challenges, offering valuable insights and suggestions along the way. I am truly grateful for her unwavering commitment to my growth and development.

My heartfelt thanks go to Jie Wang for their invaluable contributions and collaboration throughout my internship program. Jie's passion, dedication, and commitment to excellence have been truly inspiring. Their willingness to share their knowledge, provide constructive feedback, and engage in insightful discussions have significantly enriched my research experience, leaving an indelible impact on my professional growth.

I would also like to extend my deepest appreciation to the Pontes program office for creating and overseeing the internship program. Their vision, dedication, and meticulous planning have provided me with a remarkable opportunity to gain hands-on experience and apply theoretical knowledge in a practical setting. I am profoundly grateful for their efforts in designing and executing a program that has empowered me to enhance my skills and broaden my perspectives.

To all the professors, mentors, colleagues, and friends who have contributed to my internship program experience, I extend my sincere thanks. Your unwavering support, insightful discussions, and encouragement have been invaluable in shaping my research journey and personal growth.

Lastly, I would like to express my gratitude to my family and loved ones for their support, understanding, and encouragement throughout this internship program. Their belief in my abilities and constant motivation have been the cornerstone of my success.

Thank you all for being an integral part of my internship program and for your immeasurable contributions. Your support and guidance have made a lasting impact on me, and I am truly grateful for the opportunity to have worked alongside such incredible individuals.