



OpenHumidistat: Humidity-controlled experiments for everyone

Lars B. Veldscholte^{a,*}, Sissi de Beer^a



^a Sustainable Polymer Chemistry Group, Department of Molecules & Materials MESA+ Institute for Nanotechnology, University of Twente, P.O. Box 217, 7500 AE Enschede, The Netherlands

ARTICLE INFO

Article history:

Keywords:

Humidity
Controller
Electronics
Pneumatics
PID

ABSTRACT

Humidity control is a crucial element for a wide variety of experiments. Yet, often naive methods are used that do not yield stable regulation of the humidity, are slow, or are inflexible. PID-based electropneumatic humidistats solve these problems, but commercial devices are not widespread, typically proprietary and/or prohibitively expensive. Here we describe OpenHumidistat: a free and open-source humidistat for laboratory-scale humidity control that is affordable (€500) and easy to build. The design is based around mixing a humid and dry air flow in varying proportions, using proportional solenoid valves and flow sensors to control flow rates. The mixed flow is led into a measurement chamber, which contains a humidity sensor to provide feedback to the controller, to achieve closed-loop humidity control.

© 2022 The Author(s). Published by Elsevier Ltd. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

Specifications

Hardware name	OpenHumidistat
Subject area	<ul style="list-style-type: none"> • General • Engineering and material science • Chemistry and biochemistry • Biological sciences • Environmental, planetary and agricultural sciences • Educational tools and open source alternatives to existing infrastructure
Hardware type	Measuring physical properties and in-lab sensors
License	Firmware/software: GNU GPL v3.0 Hardware: CERN-OHL-S v2 Documentation: CC BY-SA 4.0
Cost of hardware	€500
Source file repository	10.17605/OSF.IO/F5U6E
OSHW certification UID	NL000006

Abbreviations: PID, Proportional, integral, derivative (control); PWM, Pulse-width modulation; PCB, Printed circuit board; PV, Process variable; SP, Setpoint; CV, Control variable; FCE, Final control element; FC, Flow controller; HC, Humidity controller; K_p , Proportional gain; K_i , Integral gain; K_d , Derivative gain; K_{ff} , Feed-forward gain; \dot{V} , (Volumetric) flowrate; UI, User interface; I/O, Input/output; RAM, Random-access memory; ADC, Analog-to-digital converter; VCCS, Voltage-controlled current sink; EDA, Electronic design automation; MOSFET, Metal-oxide-semiconductor field-effect transistor; IC, Integrated circuit; THT, Through-hole technology; SMT, Surface-mount technology; DC, Direct current; PU, Poly(urethane); PVC, Poly(vinyl chloride); DI, Deionized (water); GNU GPL, GNU General Public License; CERN OHL, CERN Open Hardware License; CC, Creative Commons.

* Corresponding author.

E-mail address: lb.veldscholte@utwente.nl (L.B. Veldscholte).

<https://doi.org/10.1016/j.ohx.2022.e00288>

2468-0672/© 2022 The Author(s). Published by Elsevier Ltd.

This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

1. Hardware in context

Humidity control is essential for a plethora of biological, chemical, and physical experiments [1], some of which simply require a stable, constant humidity, while in others the humidity might be an experimental variable which is varied. In either case, careful humidity control is necessary. A device that accomplishes this is called a humidity controller or *humidistat* (analogous to a thermostat for temperature).

Systems designed to adjust the humidity in experiments are often based on equilibria of water vapour with salt solutions [2,3], local heating of a water bath in a reservoir chamber [4,5], or using an air stream composed of a mixture of approximately dry and water-saturated air [6–9] that is manually mixed in varying ratios. The first method provides stable control, but is very inflexible: for every value of humidity, a specific salt is required. This moreover means that when an experiment calls for sweeping a range of humidities, the salt solution has to be manually swapped out for another one when moving to the next humidity point, interrupting the experiment [10]. Depending on the size of the chamber, equilibration also can take quite long. For this reason, this method is mainly suited for experiments which only require long-term stabilisation of the humidity on a single value in a air-tight chamber. The other methods are more suited to experiments requiring changing humidities, but typically yield poor control due to the lack of (automatic) feedback. This, combined with the typically long response times makes them arduous to regulate manually and prone to external disturbances.

Therefore, humidistats based on electropneumatic devices have been developed. In these devices, electronically-driven valves are used to mix a humid and dry air stream in varying proportions. A humidity sensor in a chamber is used to provide feedback to the controller actuating the valves. The advantages of such humidistats are fast settling times, good disturbance rejection, broad attainable humidity range, flexibility and portability, and easy, intuitive operation. Electropneumatic humidistats are commercially available, but are often unsuitable to many labs because they are typically prohibitively expensive, proprietary, and/or tailored to a specific experimental setup.

An open-source design of an electropneumatic humidistat using solenoid valves driven by an Arduino was published by Boulogne [11]. This was followed by the HumidOSH [12], which is a self-contained environmental chamber utilising a similar principle. In this design, ordinary (non-proportional) solenoid valves are opened/closed for varying time periods to control the ratio of humid and dry air led into a chamber. This control method, called time-proportional control, can be regarded as PWM (Pulse-Width Modulation), except on longer timescales. This method then relies on the inherent (diffusion-driven) time averaging properties of the comparatively long residence time of the chamber to attain a stable humidity. For this reason, the minimum settling times are inherently limited: Boulogne reports settling times on the order of 1/2h, and the HumidOSH takes several days. Reducing the residence time of the chamber is not an option since in that case, the resulting chamber humidity will fluctuate significantly over time. Hence, when quicker dynamics or small chambers¹ are desired, a more delicate control technique is required.

In our previous publication [13], we presented an affordable open-source humidistat for lab-scale applications based on an Arduino microcontroller, a set of proportional solenoid valves, a gas washing bottle, and a humidity sensor. Using closed-loop feedback, humidity control is achieved in a small measurement chamber by regulating the ratio of dry and humid air flows. Because this design utilises *proportional* solenoid valves which allow for analog control of the flow rate through the valve, stable humidity control can be achieved even in small chambers. It achieved fast regulation of humidity over a wide range (10–95%) with settling times within 30s.

Here, we improve upon that design, primarily by moving to cascade control, in which the individual flow rates are closed-loop controlled as well for improved robustness and dynamic performance. This is comparable to the approach taken by Gaponenko et al. [10], but avoiding the use of (expensive) off-the-shelf mass flow controllers. Furthermore, many small improvements to both the hardware and firmware are made, most notably an upgraded humidity sensor, a much larger graphical display, and a more powerful microcontroller. Meanwhile, the firmware is (re) written to include the cascade PID control loop with feed-forward and a new graphical user interface to match the new display, while care is taken to keep the code modular and fully configurable, such that it is compatible with (any combination of) both 'original' and upgraded components. Finally, by including comprehensive build instructions, printed circuit board (PCB) designs for the electronic circuits, and 3D printable models for the enclosure, this paper also facilitates reproducibility of the device.

2. Hardware description

2.1. Principle

The working principle behind the humidistat is the mixing of a humid and dry airflow in varying proportions. In order to adjust the flow rates of humid and dry air, proportional solenoid valves are utilised, which provide analog electronic control over the flow rates. A PID controller running on a Teensy LC microcontroller board [14] actuates the valves in order to attain the desired humidity in a chamber, which contains a humidity sensor to provide feedback to the controller (Fig. 1).

¹ E.g. those with a volume of several cL, as is usual for cells used for analytical techniques such as contact angle measurement, quartz crystal microbalance, ellipsometry, atomic force microscopy, neutron reflectometry, etc.

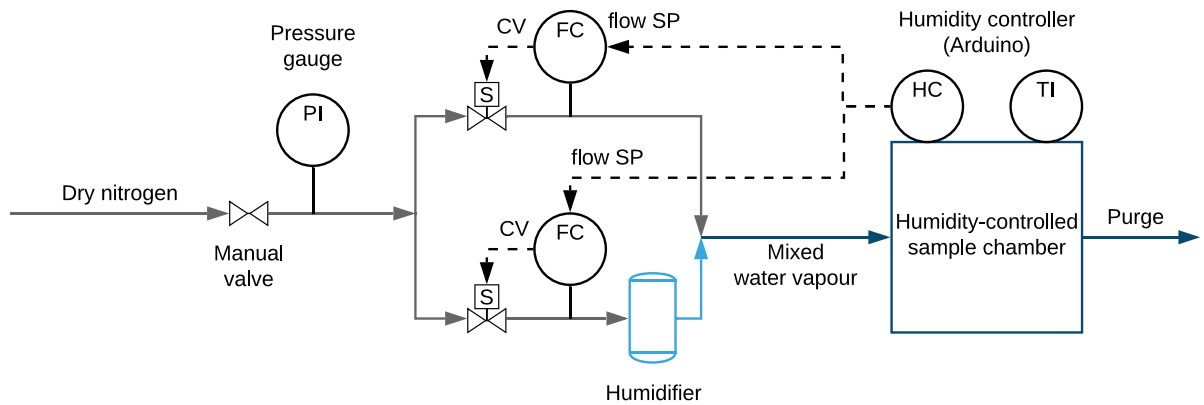


Fig. 1. Piping and instrumentation diagram of the humidistat. Starting from the left, pressure-regulated dry nitrogen is fed into the device. The stream is split into two. One of the streams is humidified, while the other is not. Solenoid valves inserted in the streams modulate the flow rates of humidified and dry nitrogen, after which they are combined again and led into the chamber. A controller acting on the humidity in the chamber sets setpoints of flow controllers that determine the ratio of humid and dry flows. PI, TI, FC, and HC stand for pressure indicator, temperature indicator, flow controller, and humidity controller, respectively.

2.1.1. PID control

PID control is a closed-loop control scheme, which means that feedback is employed to control some process variable (PV from here on). The letters P, I, and D stand for proportional, integral, and derivative respectively, and refer to the three actions included in the control scheme. The PID controller seeks to close the error between the PV and the user-desired value called the setpoint (SP) by actuating some final control element (FCE) whose state is represented by a control variable (CV). Often, the FCE is a valve, or sometimes a electric device such as a heater. [15–17].

The CV is the sum of the three principal components of the PID controller: the proportional term, which is given by the instantaneous error between the PV and SP; the integral term, which is given by the integral of the error over time; and the derivative term, which is given by the rate of change of the error with respect to time. Each term is multiplied by its respective gain (K_p , K_i , and K_d), as expressed by the following equation:

$$u(t) = K_p e(t) + K_i \int_0^t e(x) dx + K_d \frac{de(t)}{dt}. \quad (1)$$

Here, $u(t)$ is the control variable and $e(t) = r(t) - y(t)$ is the error given by the difference between the SP ($r(t)$) and PV ($y(t)$). x is a bound variable within the integral. In practice, in digital PID controllers, this continuous PID equation is discretised and the integral and derivative are approximated by finite difference methods. [17].

The PID controller can be augmented by a feed-forward action, which does not incorporate feedback, but tries to (partially) control the plant by using some pre-known model of it. An example of this is simply including the setpoint multiplied by a feed-forward gain K_{ff} :

$$u(t) = K_{ff} r(t) + K_p e(t) + K_i \int_0^t e(x) dx + K_d \frac{de(t)}{dt}. \quad (2)$$

2.1.2. Controller design

In the first version of the humidistat [13], a single control loop exists: the humidity controller. It directly regulates the humidity in the chamber by actuating the two valves as FCEs (Fig. 2). While this works reasonably well after careful tuning of the PID gains, it suffers from lack of robustness. Most notably, it is sensitive to the feed pressure, and finding an universal ‘sweet spot’ for the PID gains is difficult [13].

The main reason for the challenges in tuning the controller stems from imperfect actuators. The proportional solenoid valve response is non-linear and also exhibits rather severe hysteresis (from plunger stiction). This adversely affects the controller performance in several ways: first of all, the non-linearity results in a operating point-dependent process gain. Hence, the optimal PID gains for moderate humidity values are different from those for extreme humidity values. For example, because of the S-shaped flow rate-CV curve, a tuning parameter set that provides good performance for extreme humidity values might cause unstable/underdamped behaviour around moderate values. Moreover, the total flow rate is only constant if the valve response is linear. Non-linearities in the valve response cause variations in the total flow rate with CV, which in turn alter the process dynamics since the plant’s response time and dead time depend on the total flow rate. Finally, hysteresis in the valve response can induce oscillation of the controller [18].

An elegant way to improve the dynamic performance and stability of a PID controller is to employ cascade control (Fig. 3). In that case, there is one outer control loop which controls humidity, and two inner loops that control flow: one for each

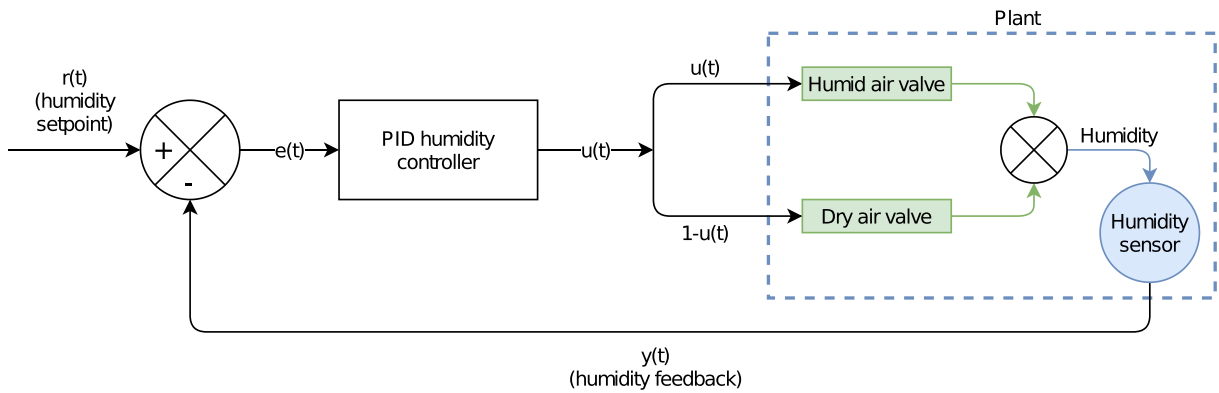


Fig. 2. Block diagram of the single control loop.

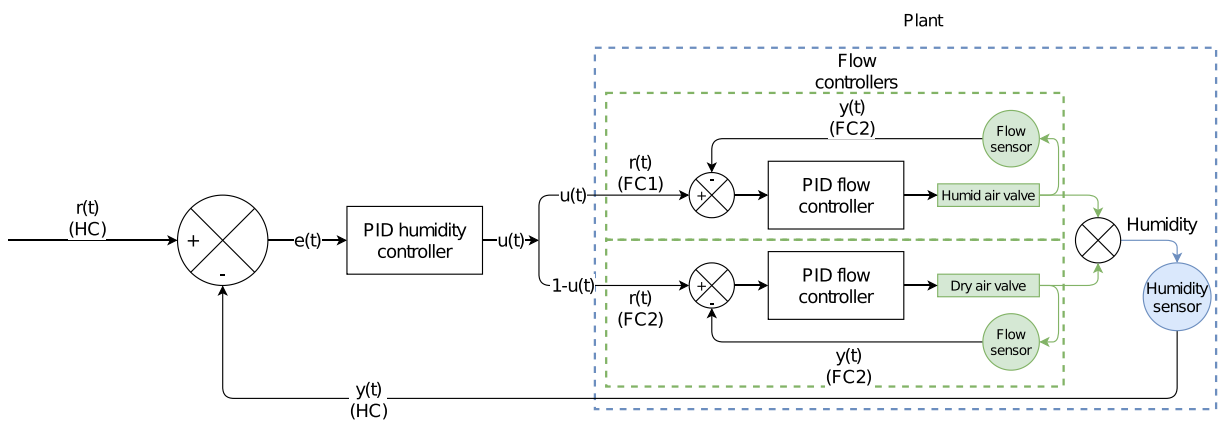


Fig. 3. Block diagram of the cascade control loop.

valve. The outer controller does not directly actuate the valves, but its output is fed into the (flow rate) setpoints of the two inner control loops [15,17]. As such, the inner control loops essentially constitute custom flow controllers. Naturally, such a setup requires flow sensors besides just a humidity sensor.

A cascade controller can improve dynamic performance by exploiting a separation of characteristic timescales in the plant: the inner loops, controlling flow, have much faster dynamics than the outer loop because the inner PVs (flow rates) respond much quicker to changes than the outer PV (chamber humidity) [15,17]. Moreover, a cascade controller is better able to deal with hysteresis in the plant [19].

Another advantage of using a cascade controller is that the actuator non-linearities are much less of a concern since they are dealt with in the inner loop [17]; the flow rates are closed-loop controlled. As a result, the total flow rate can be kept constant despite non-linear valve response [15]. This is difficult to achieve without closed-loop flow rate control: one could try to compensate for the non-linear valve response, but that requires carefully characterising the flow rate-CV curve, which is complicated by hysteresis and dependence on external factors such as the feed pressure. Closed-loop control also removes the need for carefully calibrating the zero-flow solenoid duty cycle.

2.2. Components

2.2.1. Microcontroller board

As a microcontroller board, a Teensy LC [14] is used, which is similar in terms of price to the more popular Arduino Uno which was used in the previous version, but features much improved specifications: it is faster, has more flash and RAM and more I/O pins. This allows for a more complex UI. A comparison between the specifications of the Teensy LC and Arduino Uno is given in Table 1.

OpenHumidistat can be used fully standalone, but it is possible to connect it to a PC to monitor and/or log the controller parameters using a Python utility.

Table 1
Comparison between specifications of the Arduino Uno and Teensy LC.

		Arduino Uno	Teensy LC
Price		€20 (official) €5 (compatible)	€13
Microcontroller		ATmega328P	ARM Cortex-M0+
Clock speed		16 MHz	48 MHz
Memory	Flash	32 kB	62 kB
	SRAM	2 kB	8 kB
Operating voltage		5 V	3.3 V
Digital I/O pins		14	27
ADC	Channels	6	13
	Resolution	10 bit	12 bit
PWM	Channels	6	10
	Resolution	8 bit	16 bit (at < 732Hz)

2.2.2. UI

The 16x2 character display from the previous version has been upgraded to a larger, graphical 128x64 LCD.

Meanwhile, similarly to the previous version, a 5-button keypad with 4 directional and a 'select' button is used. This keypad is implemented as a resistance ladder, such that it only occupies a single analog input pin on the microcontroller.

If desired, this keypad could relatively simply be interchanged for an alternative input method, such as a joystick or rotary encoder.

2.2.3. Sensors

The humidity sensor was upgraded from the DHT22/AM2302 to the more reliable and accurate Sensirion SHT-85 [20]. This sensor, although more expensive, is also significantly smaller, which makes it easier to fit to measurement chambers.

In order to provide flow rate feedback for closed-loop flow control, MEMS flow sensors from Omron (model D6F-P0010A1) were used. These sensors output the flow rate as an analog voltage signal, and have a maximum measurable flow rate of 1 min⁻¹.

2.2.4. Valves

In order to achieve analog control over the flow rates, proportional solenoid valves were used. Unchanged from the previous version, PVQ30-series miniature proportional solenoid valves from SMC were used.

These valves are driven by a (smoothed) PWM current that is generated by a custom solenoid valve driver (see 3.2).

2.2.5. Humidifier

One of the two air streams is humidified by bubbling it through water in two gas washing bottles. Two bottles are connected in series to ensure sufficient water vapour saturation at high flow rates.

This method is simple and does not require any electronics. Moreover, unlike ultrasonic humidifiers, it strictly produces vapour, and no aerosols.

2.3. Summary

In summary, OpenHumidistat features:

- Standalone control of humidity in a chamber
- Optional monitoring/logging of controller state using Python utility on a PC connected over USB
- High accuracy and good disturbance rejection through closed-loop control
- Broad attainable humidity range of approximately 10-90% (depending on humidifier efficacy and feed gas conditions)
- Easy and intuitive operation
- Excellent versatility and portability: can be applied to a wide range of experimental setups

3. Design files

3.1. Design files summary

Table 2 contains a summary of the design files and links to the repositories they are stored in.

Table 2
Summary of source files.

Name	Type	License	Location
Hardware	KiCAD project (schematic, board layout)	CERN-OHL-S v2	https://doi.org/10.17605/OSF.IO/UF3BN
Firmware/software	Firmware: C++ code Software: Python code	GNU GPL v3	https://doi.org/10.17605/OSF.IO/U3YNG
Enclosure	3D model (SolidWorks source, STL)	CERN-OHL-S v2	https://doi.org/10.17605/OSF.IO/T49EH

3.2. Electronic circuits

Electronic circuit schematics and PCB designs are provided for two boards: a mainboard and a solenoid driver board. Schematics of the circuits are shown in Fig. 4 and 5 respectively.

The microcontroller board (a Teensy LC) can be plugged onto the mainboard, which provides headers to connect to all peripherals (solenoid driver board, sensors, keypad, display). Besides headers, it contains a number of pull-up/down resistors

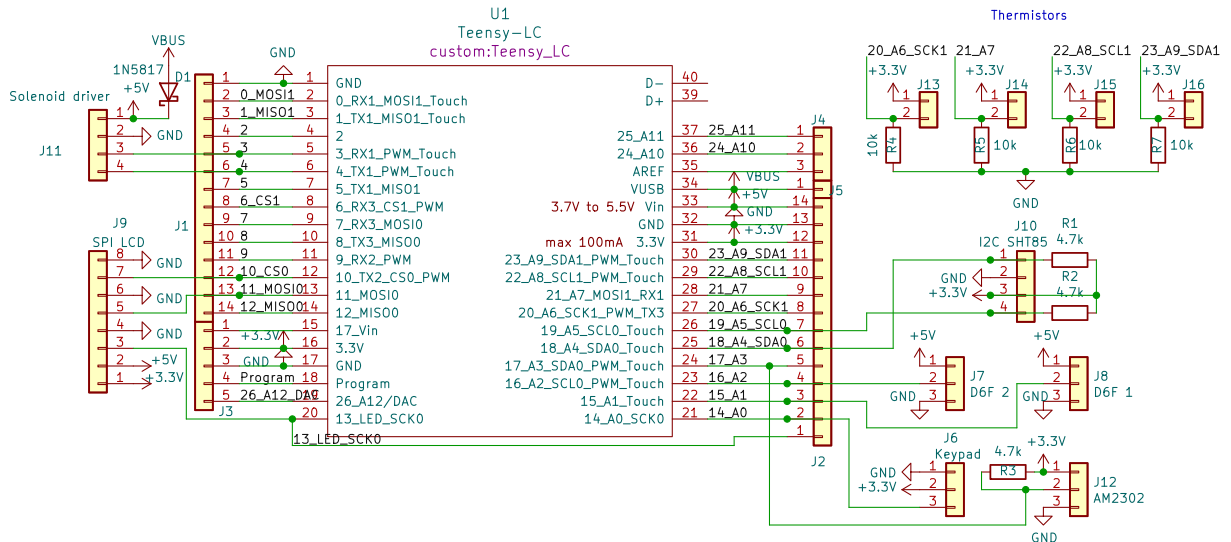


Fig. 4. Circuit schematic for the mainboard.

Voltage-controlled current sink solenoid valve driver (x2)

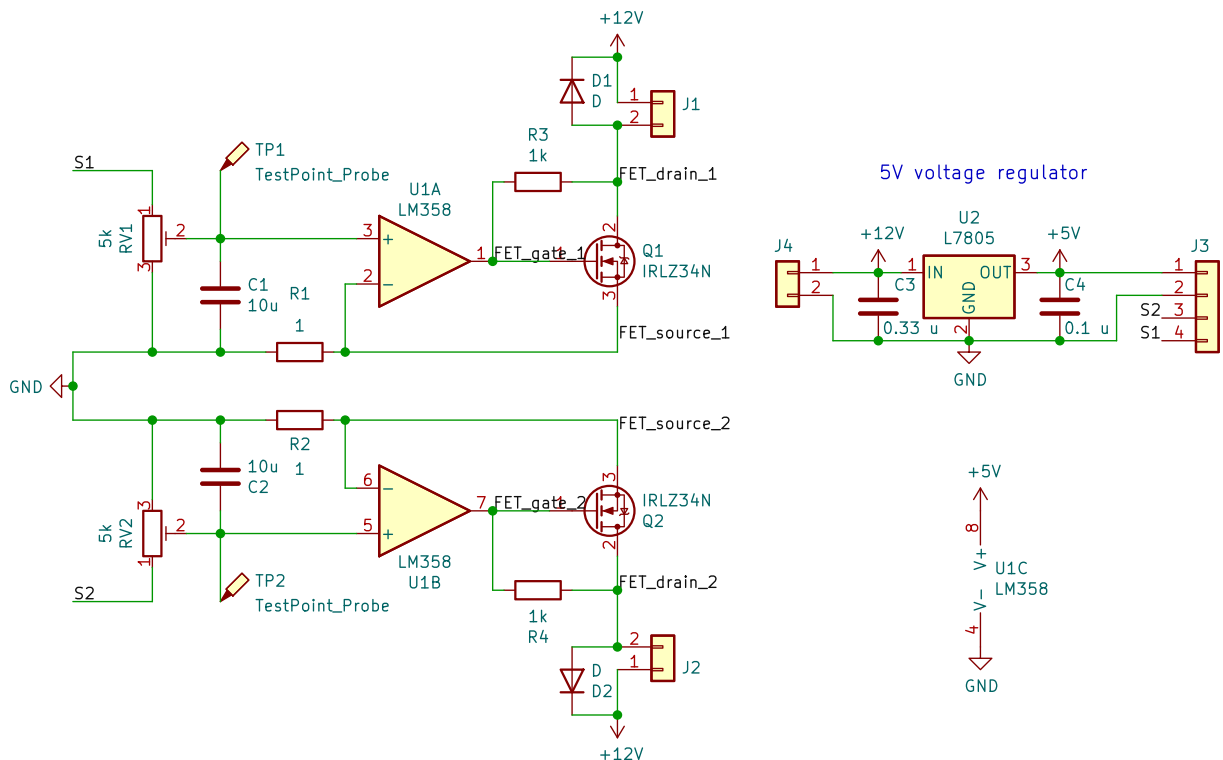


Fig. 5. Circuit schematic for the solenoid driver.

and a Schottky diode between the USB V_{BUS} and the device's $+5V$ net, to avoid potentially backpowering the USB host in case the device is powered by its own power supply and USB simultaneously.

The solenoid driver board receives low-power PWM control signals from the microcontroller, and uses that to drive the (two) solenoid valves. More specifically, it comprises a 2-channel VCCS (Voltage-Controlled Current Sink), to drive the solenoid valves in a constant-current manner. This is necessary because the solenoids heat up considerably during normal operation, and this consequently alters their resistance. Without a constant-current driver, the solenoid current, which is the parameter that controls the flow rate, would drift with temperature. Design details about the VCCS are available in the supplementary material of the previous publication [13].

The solenoids require 12V. This is provided by an external power supply, which is connected to header J4 on the solenoid driver board. To allow the microcontroller and peripherals also to be powered through this power supply, the solenoid driver board includes a linear voltage regulator which creates a $+5V$ line from the $+12V$ supply.

An archive containing Gerber and drill files of the PCB designs, as well as source files of the circuit schematics and PCB designs which can be opened in the free EDA KiCad [21] are made available under the CERN-OHL-S.

3.3. Firmware

The firmware running on the microcontroller is written in C++ with the Arduino Core framework and the PlatformIO [22] build system. It is written using the object-oriented programming paradigm, and as such consists of a number of modular classes.

The source is available under the GNU GPL v3.

3.4. Enclosure

CAD models for the enclosure are provided: source files (SolidWorks) and STL files for 3D printing.

4. Bill of materials

The full bill of materials is available at <https://dx.doi.org/10.17605/OSF.IO/F5U6E>.

We note that electronic components are listed with their prices per unit, but are commonly purchased in larger quantities (usually at least 10). This means that the actual cost might be inflated somewhat if one does not already have access to these (rather common) electronic components. Still, since these components are relatively cheap, even having to purchase 10 times the required amount does not increase the total price of the device by much.

Electronic and pneumatic components can be sourced from retailers such as RS Components, Mouser, Farnell, and Digikey. The gas washing bottles can be sourced from laboratory glassware distributors such as VWR and Fisher Scientific.

A summarised version of the BoM can be found in Table 3. It can be seen that the custom parts of the device are very cheap to build, and the majority of the total cost of the device comes from off-the-shelf components such as the humidity sensor, the flow sensors, gas washing bottles, and most notably the proportional solenoid valves. The total cost of the materials needed to build OpenHumidistat is less than €500.

Table 3
Summarised bill of materials.

Qty	Item	Unit price (€)	Ext. price (€)
1	Solenoid driver board (incl. components)	1.67	1.67
1	Mainboard (incl. components)	0.35	0.35
1	Teensy LC microcontroller board	13.00	13.00
1	128x64 ST7920 LCD	9.00	9.00
1	Keypad	0.75	0.75
1	12 V 1A DC power supply	8.00	8.00
1	Humidity sensor (Sensirion SHT-85)	26.35	26.35
2	MEMS flow sensor (Omron D6F-P0010A1)	45.38	90.76
2	Proportional solenoid valve (SMC PVQ31-6G-16-01F)	82.16	164.32
2	Gas washing bottle	70.00	140.00
	Cabling and connectors	9.04	9.04
	Pneumatic tubing and couplings	24.35	24.35
		Grand total	489.10

5. Build instructions

5.1. Materials and tools

Refer to the BoM (Section 4) for a comprehensive list of materials to order.

The PCBs can be ordered for quite cheap from various PCB manufacturers: at the cheapest fab houses, 5 units can be purchased for merely \$2 (excluding shipping). To do so, submit the Gerber and drill files (see Section 3.2) to a manufacturer of your choice. An online price comparison tool can be found at <https://pcbshopper.com>.

Building the device requires access to some basic tools. For the electronics, these are:

- Soldering iron (+ solder)
- Wire cutters
- Stripping pliers
- Crimping tools for:
 - DuPont connectors
 - JST connectors
 - Modular connectors
- Tweezers
- Multimeter

For the enclosure, STL files are provided, intended to be 3D printed. Alternatively, the enclosure can be made in a different way, such as by modifying a universal enclosure. For this, one needs (access to) a workshop with some (machine) tools.

Additionally, some commonplace mounting materials (not listed on the BoM) are required, such as screws, nuts, and standoffs.

5.2. Flashing the microcontroller

1. Configure the OpenHumidistat firmware by modifying `src/config.h` as required.
2. Flash the OpenHumidistat firmware to the Teensy microcontroller by connecting it to a PC using a USB (Type-A to Micro-B) cable.

Detailed instructions can be found in the `README.md` file included with the firmware (see Section 3.3).

5.3. Electronics

After obtaining the electronic components and the PCBs, the boards can be assembled. When populating the PCBs with the components, use the reference designators printed in silkscreen on the PCB and the BoM (or circuit schematic) to look up which component goes where. Fig. 6 can also be used as a visual reference.

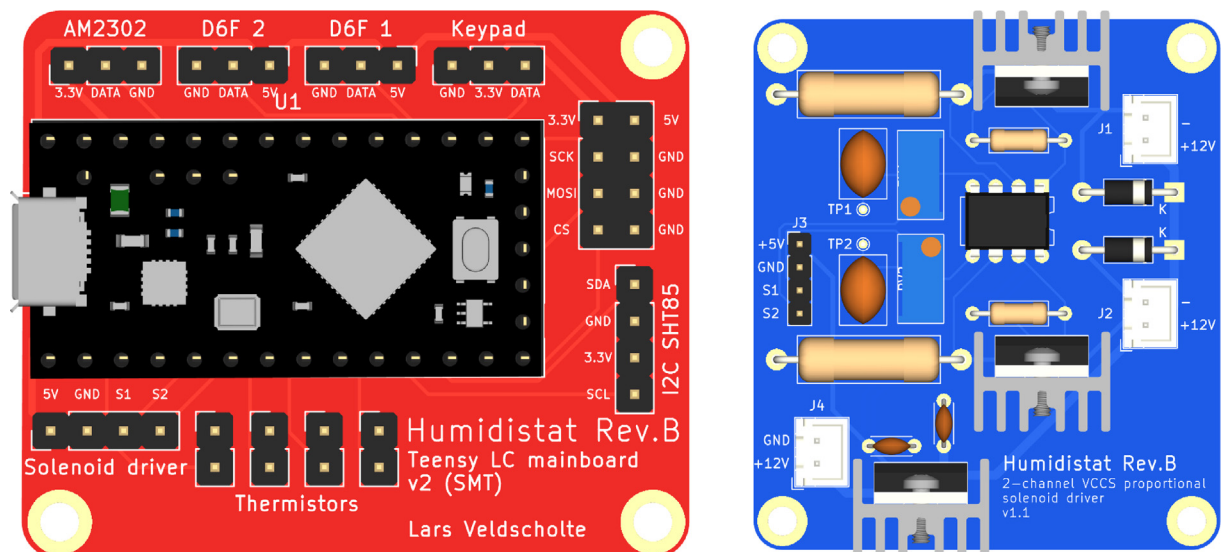


Fig. 6. Renders of the two PCBs (not to scale).

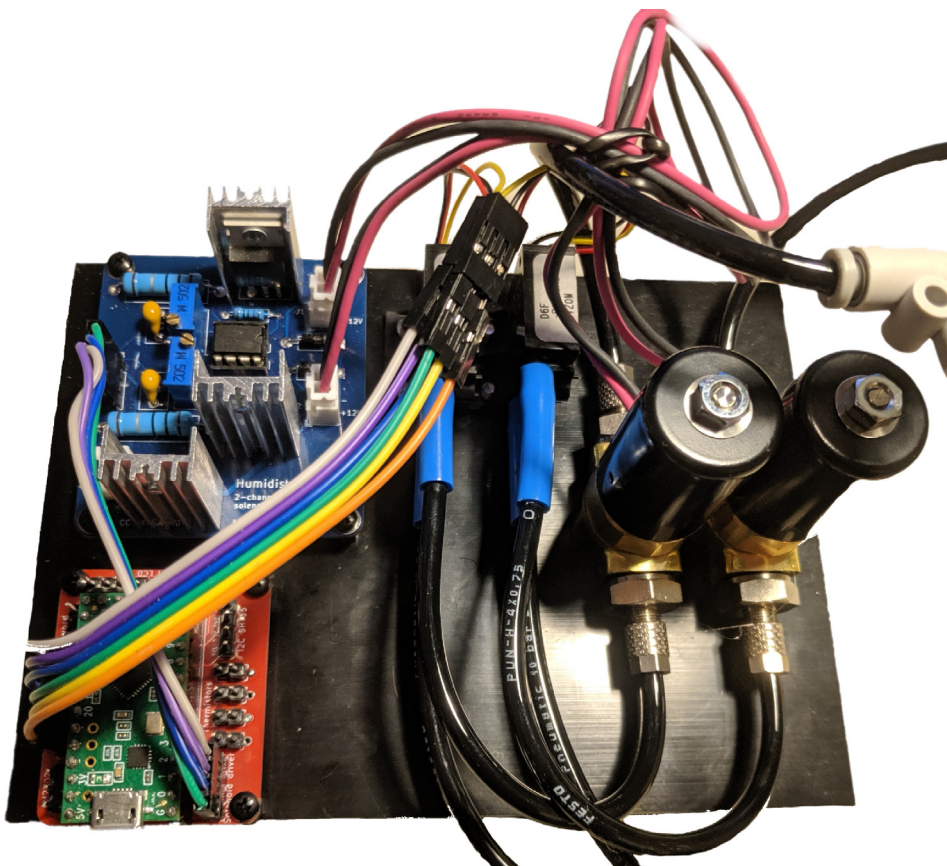


Fig. 8. Baseplate with components installed and partially connected.



Fig. 9. Baseplate installed in the 3D printed enclosure, with frontpanel open.

5.4.1. Humidity sensor(s)

1. Create the humidity sensor cable: solder one end of a length (1-2m recommended, depending on the desired distance between the OpenHumidistat and the measurement chamber) of 4-wire data cable to the SHT-85 sensor and crimp a 6P4C plug on the other end. Mind the pin order: it should match that of the header marked I2C SHT85 on the mainboard.
2. Make a cable to connect the female modular jack (which will be mounted in the front panel) to the mainboard: crimp a 4-pin female DuPont connector on one end, and solder the wires on the other end to the pins of the female modular jack. Again, mind the order of the pins: the cable should preserve it along the entire path from the header on the mainboard, to the actual SHT-85 sensor. Use a multimeter to verify the continuity of the connections, if required.

For backwards compatibility, the mainboard also includes a header for connecting a DHT22/AM2302 humidity sensor. If desired, the above procedure can be repeated for this sensor type.

5.4.2. Power

Make a cable to connect the barrel jack (which will be mounted in the front panel) to the +12 V header (J4) on the solenoid driver board. For this, wire thicker than that used for the signal cables should be used: at least 0.5 mm² (AWG 20).

1. Crimp a JST-XH female connector to one end of two wires, and solder the other ends to the pins of the barrel jack. Mind the polarity: it should match that of the barrel plug of the power supply used. Usually, it is 'center positive', but verify this: reverse polarity will damage the electronics.
2. Use heat-shrink tubing to insulate the end of the barrel jack.

5.4.3. Solenoid valves

1. Crimp female 2-pin JST-XH connectors to the solenoid valve wires. Mind the polarity: the red wire should correspond to the positive bottom pin on headers J1/J2 (marked +12 V).

5.4.4. Thermistors

Optionally, 4 thermistors can be installed for temperature monitoring of the solenoids and MOSFETs. For this,

1. Crimp female 2-pin DuPont connectors to the thermistor cables.
2. Glue or otherwise attach the thermistors to the solenoid valves and MOSFET heatsinks.

5.4.5. Other peripherals

The other peripherals (display, keypad, flow sensors) are connected using (female–female) DuPont cables, which can be simply used as-is.

Refer to [Table 4](#) for connecting the display module to the mainboard.

Table 4
Pin mapping between mainboard and ST7920 display module.

Mainboard		ST7920 display
#	Label	Label
1	+ 3.3 V	BLA
2	+ 5 V	VCC
3	SCK	E
4	GND	BLK
5	MOSI	RW
6	GND	PSB
7	CS	RS
8	GND	GND

5.5. Testing and calibration

1. Partially assemble the device: install the Teensy onto the mainboard and connect the solenoid driver board, display, keypad, and flow sensors to the mainboard.
2. When connecting the device over USB (connected to the Teensy), the device should power up. Check that the firmware is running, that the interface is shown on the display, and that you can interact with it using the keypad.

3. Calibrate the solenoid driver by adjusting the trimpots. To do this:
 - (a) Make sure there is a 3.3V signal applied to the pins S1/S2 of the solenoid driver board. This is most straightforwardly done by having it connected to the mainboard, and having the microcontroller output a 100% duty cycle signal on both of the channels. This should happen automatically, as the flow controllers' CVs will saturate with no actual flow through the sensors.
 - (b) After verifying on the display that the microcontroller is outputting a 100% duty cycle signal on both channels, use a multimeter to measure the voltage between each of the test points TP1 and TP2, and ground. Adjust each trimpot (RV1 and RV2) until the voltage at the respective test point reads exactly 0.330V.

5.6. Enclosure

1. Print the enclosure using the included STL files (two parts: the enclosure, and the frontpanel)
2. Make a base plate onto which to affix the components.
 - (a) Cut a piece of PVC to size.
 - (b) Drill holes in the base plate for affixing the components to it, and it to the bottom of the enclosure.
3. Screw the components to the baseplate:
 - (a) The PCBs (use 6mm spacers)
 - (b) The flow sensors
 - (c) The solenoid valves. See Fig. 8 for reference.
4. Tap M3 thread in the holes in the enclosure for affixing the frontpanel.
5. Install components into the enclosure:
 - (a) Affix the display board using 6mm spacers.
 - (b) Affix the keypad. You may need to reposition the pin header.
 - (c) Affix the DC barrel jack.
6. Install components into the frontpanel:
 - (a) Glue the modular jacks into place.
 - (b) Affix the pneumatic bulkhead union couplings.
7. You can then slide the baseplate including components into the enclosure, and screw the frontpanel on (after connecting cables and tubing, of course). See 5.8 for details on the pneumatic connections. See Fig. 9 for reference.

5.7. Measurement chamber

A wide variety of measurement chambers can be used, depending on the envisioned application of the humidistat. The only requirements are that it must fit the humidity sensor, and that it should be reasonably closed; if there is too much exchange with ambient air, it might be difficult to reach extreme humidity values. The device is tested with chambers with volumes of several cL, but smaller or larger chambers can be used if desired. Note that larger chambers will have larger residence times, which will in turn limit the speed of the controller.

If for a particular experimental setup it is undesirable or impossible to fit the humidity sensor in the measurement chamber, it also possible to construct a 'pre-chamber' containing just the humidity sensor, which is then connected between the humidistat and the measurement chamber. This pre-chamber should be completely air-tight in this case. The advantage of this configuration is the versatility: the humidistat can be used with practically every experimental setup featuring a flow cell this way without any modification. A potential disadvantage arises when the actual measurement chamber is too open, in which case exchange with the ambient air occurs and the actual humidity will deviate from the measured humidity in the pre-chamber.

5.8. Assembling the device

Once all the individual assemblies are completed, the full device can be assembled.

1. Screw the G1/8 \lt --- \gt 4mm barbed fittings into the solenoid valve ports.
2. Make the pneumatic connections using PU tubing. See Fig. 10 for reference.

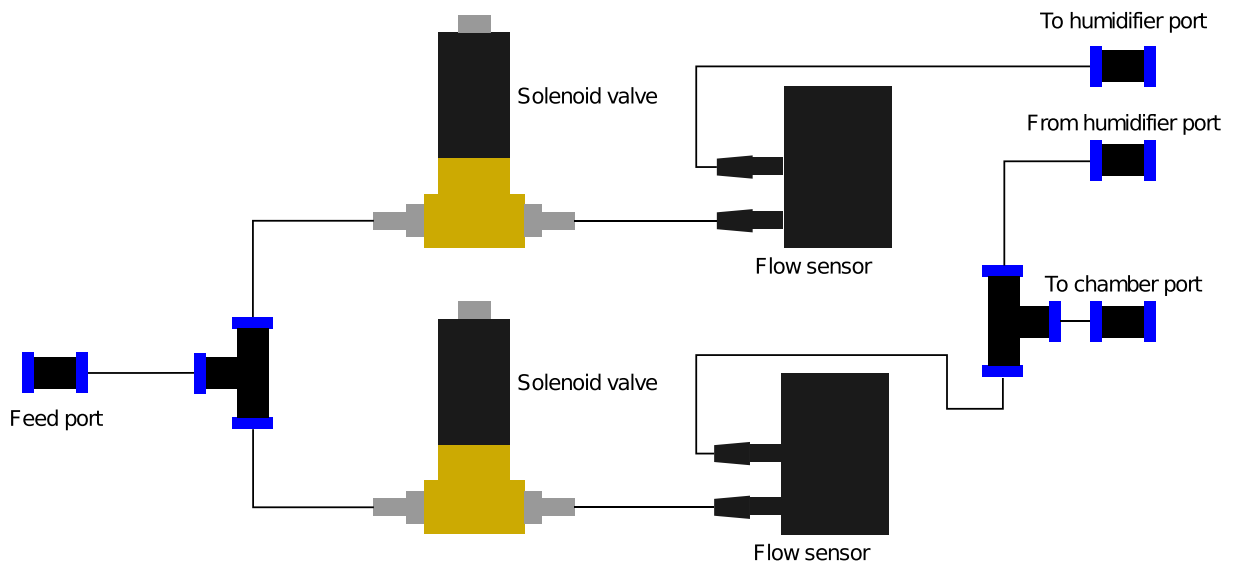


Fig. 10. Diagram illustrating the pneumatic connections between the components.

6. Operation instructions

To turn on the device, connect the external 12V power supply to the DC jack. The microcontroller (and peripherals) can also be powered from the USB port, but for operation of the valves, a 12V supply is required.



Fig. 11. Finished device in 3D printed enclosure.

The display should power up (see Fig. 11). Instruction for the usage of the UI can be found in the `README.md` included with the firmware.

6.1. Setting up the device

6.1.1. Humidifier

1. Fill the gas washing bottles with (DI) water.
2. Connect the two gas washing bottles in series.
3. Connect the gas washing bottles to the designated ports on the device. Make sure the direction is correct: if it is wrong, it will flush water into the system.

6.1.2. Chamber

1. Using PU tubing, connect the outlet port of the device to the measurement chamber.
2. Also connect the humidity sensor installed in this measurement chamber to the device.

6.1.3. Feed

As supply gas, either dry air or nitrogen can be used. Although not strictly required, the solenoid valves perform better when it is pressure-regulated (at 1-2 bar).

1. Connect the nitrogen/air supply to the inlet port of the device.
2. Increase the feed pressure until the OpenHumidistat's flow controllers achieve sufficient flow (recognisable by the stopping of blinking of the flow controller labels on the display). Note that you need at least about 1 bar for the solenoid valves to open.

6.1.4. Testing and tuning

The system contains two solenoid valves and two flow sensors: one set for humid flow, and one for dry flow. Hence, there are 4 ways to connect these, but they must be connected in the right configuration for the controllers to work properly. To test this:

1. Turn on the device.
2. With the humidity controller in manual mode and feed pressure applied to the device's inlet:
 - If the flow controllers manage to successfully regulate the flow rates, the flow sensors correctly correspond to the solenoid valves.
 - If both flow controllers saturate (one CV goes to 0%, while the other goes to 100%), the flow controllers are erroneously trying to control each other's flow. In this case, swap either:
 - the pneumatic connections between the flow sensors and the solenoid valves,
 - the connectors of either the flow sensors (on the mainboard) or the solenoid valves (on the solenoid driver board),
 - or the pin numbers of either the flow sensors or the solenoid driver channels in the firmware's config file.
3. Turn the humidity controller into automatic mode.
 - If the humidity controller manages to regulate the humidity by converging on the SP (even if doing so slowly, or with overshoot), the device is configured correctly.
 - If the humidity controller saturates (the CV goes to either 0% or 100% and stays there), the humid and dry flows are swapped. In this case, swap either:
 - the pin numbers of *both* the flow sensors and solenoid driver channels in the firmware's config file,
 - the electrical connectors of *both* the flow sensors and the solenoid valves
 - or the pneumatic connections of the dry and wet flows (disconnect the tubing from the flow sensor going to the humidifier and connect it directly to the union tee, and disconnect the tubing going directly going to the union tee and connect it to bulkhead union going to the humidifier).
4. Tune the two PID controllers (outer humidity controller and inner flow controllers) appropriately for the total flow rate and measurement chamber used by adjusting the controller parameters. Defaults for these can be set in `config.h`, but they can also be overridden using the UI itself in a standalone manner. Section 7.1 illustrates the tuning procedure.

6.2. Hazards

Because the outlet of the gas washing bottles is subject to back pressure from the dry flow, there exists a risk of backflow through the gas washing bottles when the inlet of the gas washing bottles is left unconnected, or when the humid solenoid valve is closed (open) and the tubing and/or connections from there to the gas washing bottles are not completely air-tight. This is problematic because it will siphon liquid water out the gas washing bottles into the system.

Users should be alert of this potential issue, which can be spotted by the water level rising in the tubes of the gas washing bottles. If this problem occurs, it is strongly recommended to install a check valve (e.g. SMC AKH, Festo H-QS) in line with the gas washing bottles, which should prevent any backflow.

7. Validation and characterisation

7.1. Tuning

A manual tuning procedure is outlined below. This was performed with a ‘universal pre-chamber’ with a volume of 25 mL, which with the used total flow rate of 2 min^{-1} gives a residence time of 0.75s.

7.1.1. Flow controllers

The inner control loops, constituting flow controllers, are tuned as PI-only controllers with a high K_i , so that they essentially are integral controllers. This works very well because the plant, that is, the flow through the solenoid valve, is a fully self-regulating process that, although non-linear and exhibiting hysteresis, fortunately almost does not have any dead or response time: it responds virtually instantly to changes that the controller makes by actuating the valves. Hence, overshoot is barely of concern and this controller is very stable. Because the signal from the flow sensors is rather noisy and because it is not necessary in this case, derivative action is not used ($K_d = 0$). [15].

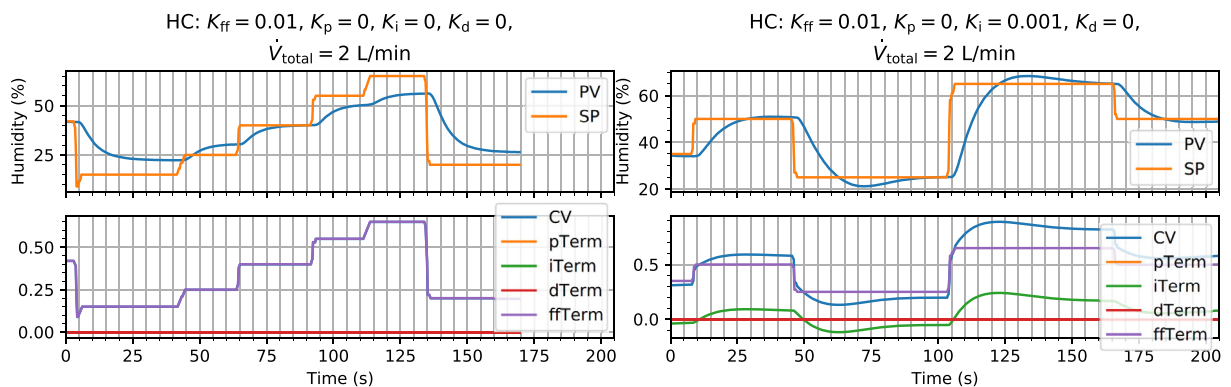
As mentioned in the first section, a major improvement over the single-loop controller is that it is no longer critical to meticulously calibrate the minimum value of the CV (solenoid duty cycle). Whereas this was crucial in the previous design to make sure the resulting total combined flow rate is reasonably constant and no deadband exists at the lower range of the solenoid response, with closed-loop flow control the this zero-flow duty cycle is automatically ‘found’ by the flow controller. In the cascade control scheme this parameter still exists, but solely for the proper functioning of the anti integral windup mechanism.

7.1.2. Humidity controller

For the (outer) humidity controller, is helpful to start off by considering a purely *feed-forward* (ff) controller. In that case, the controller solely acts on the SP, and there is no *feedback*. In other words, the CV is purely a function of the SP.

Feed-forward-only controllers like these are rarely ever sufficient because they cannot deal with external disturbances (after all, if they were, the entire PID controller would be unnecessary), but including a significant feed-forward action in our outer humidity controller turns out to be a useful starting point. We set K_{ff} to 0.01, since that creates a CV of 1 (the maximum) for a SP of 100 (%)². The implicit model behind this feed-forward action is that the resulting humidity of the mixed-together dry and humid air is proportional to the ratio of those two flow rates. This is a reasonable initial guess that nevertheless fails when the ‘dry’ air is slightly moist, the humid air is not fully saturated, or when the plant is confronted with other external disturbances, but that does not matter, since the feedback action (PID) exists to correct for those. The main benefit of partly relying on feed-forward action, is that this can supplant most of the integral action in providing the ‘offset’, yet without the potential for destabilising the controller.

The dynamic response of the feed-forward-only controller is shown in Fig. 12a. Although it is physically impossible for this controller to oscillate or overshoot, the PV values it settles on deviate significantly from the SP (for most values) and the settling times are far from optimal.



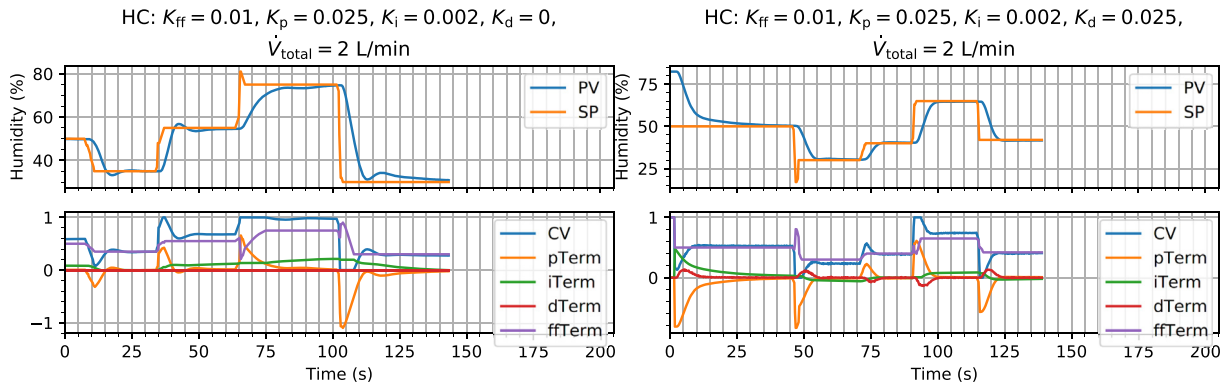
(a) Controller response with feed-forward only. (b) Controller response with feed-forward and integral action.

Fig. 12.

² In the PID code, the CV is normalised between 0 and 1, but the humidity PV/SP runs from 0 to 100.

As an example, a small amount of integral action is then added to help with static deviations from the aforementioned ideal model. The resulting response is shown in Fig. 12b. The controller now eventually homes in on the SP, but only rather slowly, and after considerably overshooting it. This is an inherent effect of high integral action on a plant with some time delay [15,16].

Next, we add proportional action. This makes the controller act briefly on the error between the PV and SP, and has the effect of reducing the accumulation time of the integral action, so the settling times improve (Fig. 13a).



(a) Controller response with feed-forward, proportional, and integral action.

(b) Controller response with feed-forward, proportional, integral, and derivative action.

Fig. 13.

Derivative action is also added to accelerate the dynamic response by reducing overshoot, and finally the response is reasonably satisfactory, with good setpoint tracking, and settling times on the order of 10s (Fig. 13b). An overview of the tuning parameters is given in Table 5

Table 5
Tuning parameters.

	K_{ff}	K_p	K_i (s^{-1})	K_d (s)
FC	0	0.005	0.05	0
HC	0.01	0.025	0.002	0.025

7.2. Robustness

Due to closed-loop flow control, the system is a lot more robust to various feed pressures. Whereas with the single-loop humidistat variations in feed pressure and/or chamber pressure drop led to varying flow rates, which in turn affects the solenoid valve response and chamber response time, the flow controllers in the cascade humidity controller always keep the total flow rate constant despite varying feed pressures. This makes the controller more portable, especially in situations where the feed pressure is not controlled.

7.3. Capabilities

Range	10 % - 90 %
Maximum total flow rate	2 min^{-1}
Settling time	~ 10 s
Accuracy	± 1.5 %
Precision	0.01 %

8. Conclusion and outlook

We presented the design of a free and open-source humidistat. Due to the usage of cascade PID control, in which the dry and humid flow rates are also closed-loop controlled, the dynamic performance and robustness was significantly improved over our previous design in which a single control loop was used: Settling times are improved from ~ 30 s to ~ 10 s.

The versatility of the device makes it broadly applicable, bringing humidity control to a wide variety of experimental and analytical setups.

As a final note, we want to mention the possibility of generalising the OpenHumidistat beyond water vapour. By swapping the water in the gas washing bottles for a different substance, this design could readily function as a vapour pressure controller for arbitrary vapours, provided that an appropriate vapour sensor for it exists.

CRediT authorship contribution statement

Lars B. Veldscholte: Conceptualization, Software, Validation, Investigation, Writing - original draft, Writing - review & editing.

Sissi de Beer: Conceptualization, Supervision, Funding acquisition, Writing - review & editing.

Declaration of Competing Interest

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

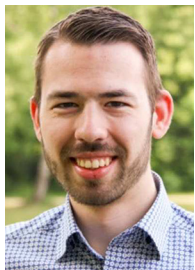
Acknowledgements

The authors extend their thanks to Koen Jorissen for his help with CAD modelling, 3D printing, and assembling of the enclosure.

This research received funding from the Dutch Research Council (NWO) in the framework of the ENW PPP Fund for the top sectors and from the Ministry of Economic Affairs in the framework of the 'PPS-Toeslagregeling' regarding the Soft Advanced Materials consortium.

References

- [1] P. Johansson, K. Kalliorinne, P. Marklund, M. Björling, Precise control of operating conditions in tribotesting with respect to trace humidity and contact temperature, *MethodsX* 8 (2021) 101362, <https://doi.org/10.1016/j.mex.2021.101362>, issn: 2215-0161 (Visited on 2021-10-15).
- [2] L. Greenspan, Humidity fixed points of binary saturated aqueous solutions, *Journal of Research of the National Bureau of Standards Section A: Physics and Chemistry* (1977) 89–96.
- [3] M. Brió Pérez, M. Cirelli, S. de Beer, Degrafting of Polymer Brushes by Exposure to Humid Air, *ACS Applied Polymer Materials* 2 (8) (2020) 3039–3043, <https://doi.org/10.1021/acsapm.0c00474>.
- [4] L. Perino-Gallice, G. Fragneto, U. Mennicke, T. Salditt, F. Rietord, Dewetting of solid-supported multilamellar lipid layers, *Eur. Phys. J. E* 8 (3) (2002) 275–282, <https://doi.org/10.1140/epje/j2001-10101-0>.
- [5] S. Micciulla et al, One-step procedure for the preparation of functional polysaccharide/fatty acid multilayered coatings, *Communications Chemistry* 2 (2019) 61, <https://doi.org/10.1038/s42004-019-0155-y>.
- [6] S. de Beer, D.T. Mannetje, S. Zantema, F. Mugele, Instability of Confined Water Films between Elastic Surfaces, *Langmuir* 26 (5) (2010) 3280–3285, <https://doi.org/10.1021/la903051k>.
- [7] A. Chhabra et al, Humidity Effects on the Wetting Characteristics of Poly(N-isopropylacrylamide) during a Lower Critical Solution Transition, *Langmuir* 29 (25) (2013) 8116–8124.
- [8] S. Karpitschka, F. Liebig, H. Riegler, Marangoni Contraction of Evaporating Sessile Droplets of Binary Mixtures, *Langmuir* 33 (19) (2017) 4682–4687, <https://doi.org/10.1021/acs.langmuir.7b00740>.
- [9] H. Atesci et al, Humidity-controlled rectification switching in ruthenium-complex molecular junctions, *Nature nanotechnology* 13 (2) (2018) 117–121, <https://doi.org/10.1038/s41565-017-0016-8>.
- [10] I. Gaponenko, L. Musy, S.C. Muller, P. Paruch, Open source standalone relative humidity controller for laboratory applications, *Engineering Research Express* 1 (2) (2019) 025042, <https://doi.org/10.1088/2631-8695/ab5771>.
- [11] F. Boulogne, Cheap and versatile humidity regulator for environmentally controlled experiments, *The European Physical Journal E* 42 (4) (2019) 51, <https://doi.org/10.1140/epje/j2019-11813-0>, issn: 1292-895X.
- [12] S.K. Lau, J. Subbiah, HumidOSH: A self-contained environmental chamber with controls for relative humidity and fan speed, *HardwareX* 8 (2020) , <https://doi.org/10.1016/j.ohx.2020.e00141> e00141.
- [13] L.B. Veldscholte, R.J. Horst, S. de Beer, Design, construction, and testing of an accurate low-cost humidistat for laboratory-scale applications, *The European Physical Journal E* 44 (4) (2021) 48, <https://doi.org/10.1140/epje/s10189-021-00062-5>.
- [14] Teensy LC development board. url:<https://www.pjrc.com/store/teensylc.html> (visited on 2021-10-25).
- [15] T. Kuphaldt. Lessons In Industrial Instrumentation. Lessons in Industrial Instrumentation Series. Samurai Media Limited, 2017. isbn: 978-988-8407-08-8.
- [16] T. Wescott. PID Without a PhD. 2018. url:<http://wescottdesign.com/articles/pid/pidWithoutAphd.pdf> (visited on 2021-12-12).
- [17] A. Visioli, Practical PID Control. Advances in Industrial Control, isbn:, Springer, London, 2006, <https://doi.org/10.1007/1-84628-586-0>, 978-1-84628-586-8.
- [18] R.B. di Capaci and C. Scali. 'An augmented PID control structure to compensate for valve stiction', in: IFAC-PapersOnLine. 3rd IFAC Conference on Advances in Proportional-Integral-Derivative Control PID 2018 51.4 (2018), pp. 799–804. doi: 10.1016/j.ifacol.2018.06.181.
- [19] C. Li, M.A.A. Shoukat Choudhury, B. Huang, F. Qian, Frequency analysis and compensation of valve stiction in cascade control loops, *Journal of Process Control* 24 (11) (2014) 1747–1760, <https://doi.org/10.1016/j.jprocont.2014.09.009>.
- [20] R. Smith. Compare DHT22, DHT11 and Sensirion SHT71. 2017. url:https://www.kandrsmith.org/RJS/Misc/Hygrometers/calib_dht22_dht11_sht71.html (visited on 2021-02-03).
- [21] KiCad EDA: A Cross Platform and Open Source Electronics Design Automation Suite. url:<https://www.kicad.org/> (visited on 2021-10-25).
- [22] PlatformIO: A professional collaborative platform for embedded development. url:<https://platformio.org/> (visited on 2021-10-25).
- [23] Using External Power and USB with the Teensy USB development board. url:https://www.pjrc.com/teensy/external_power.html (visited on 2021-10-25).



Lars Veldscholte is a PhD student at the Sustainable Polymer Chemistry group at the University of Twente, Enschede, the Netherlands. He obtained his BSc in Chemical Engineering and MSc in Material Science and Engineering under the supervision of dr. Sissi de Beer at the University of Twente. His research focuses on understanding polymer brushes through a combination of experimental and computational chemistry and physics.



Sissi de Beer obtained her PhD degree at the University of Twente (Netherlands) in 2011. After postdoctoral positions at the Forschungszentrum in Jülich (Germany) and the University of Toronto (Canada), she became an assistant professor at the University of Twente in 2016. Together with her students, she develops functional polymer coatings by combining modeling and laboratory experiments.