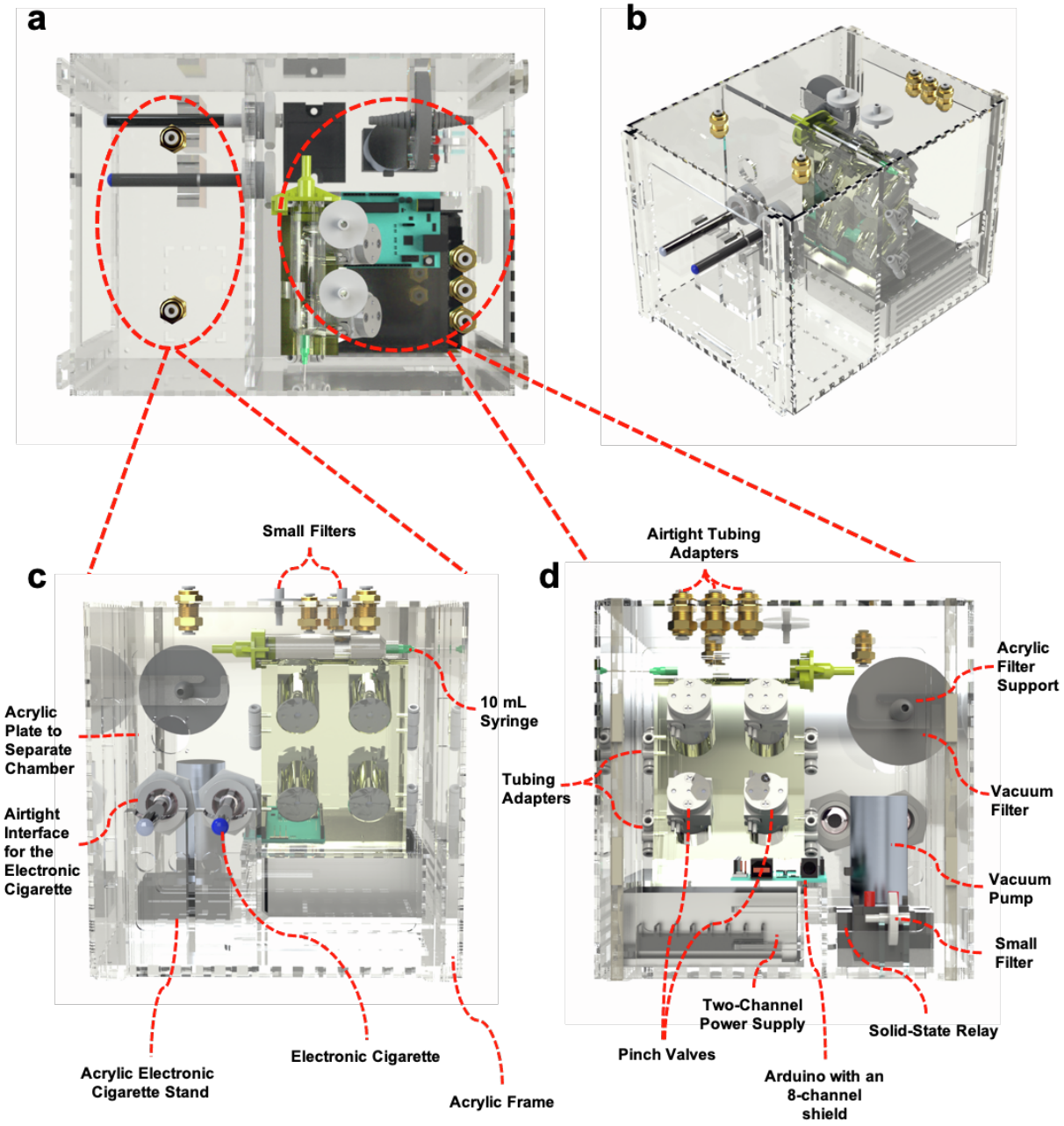
iScience, Volume *24*

Supplemental information

A robotic system for real-time analysis of inhaled submicron and microparticles

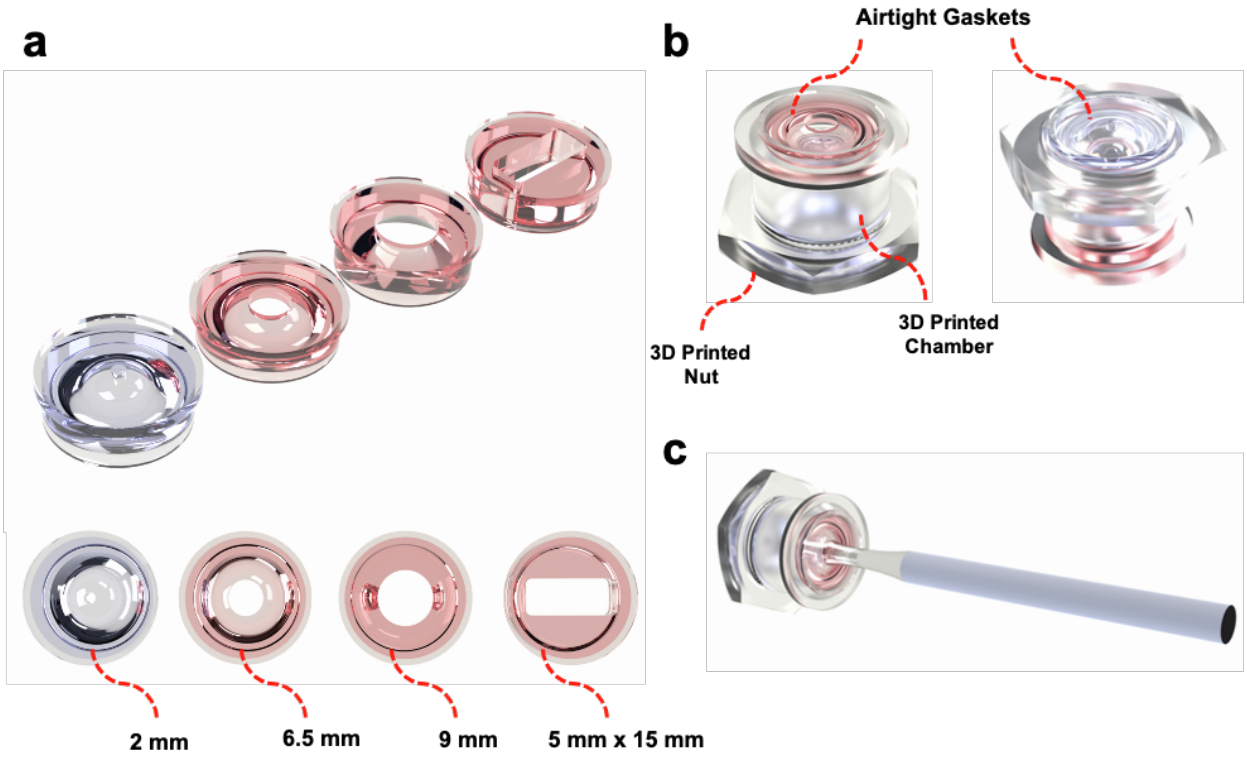
Alexander J. Kaiser, Cassie Salem, Bob J. Alvarenga, Anthony Pagliaro, Kelly P. Smith, Luis G. Valerio Jr., and Kambez H. Benam

SUPPLEMENTARY FIGURES AND FIGURE LEGENDS

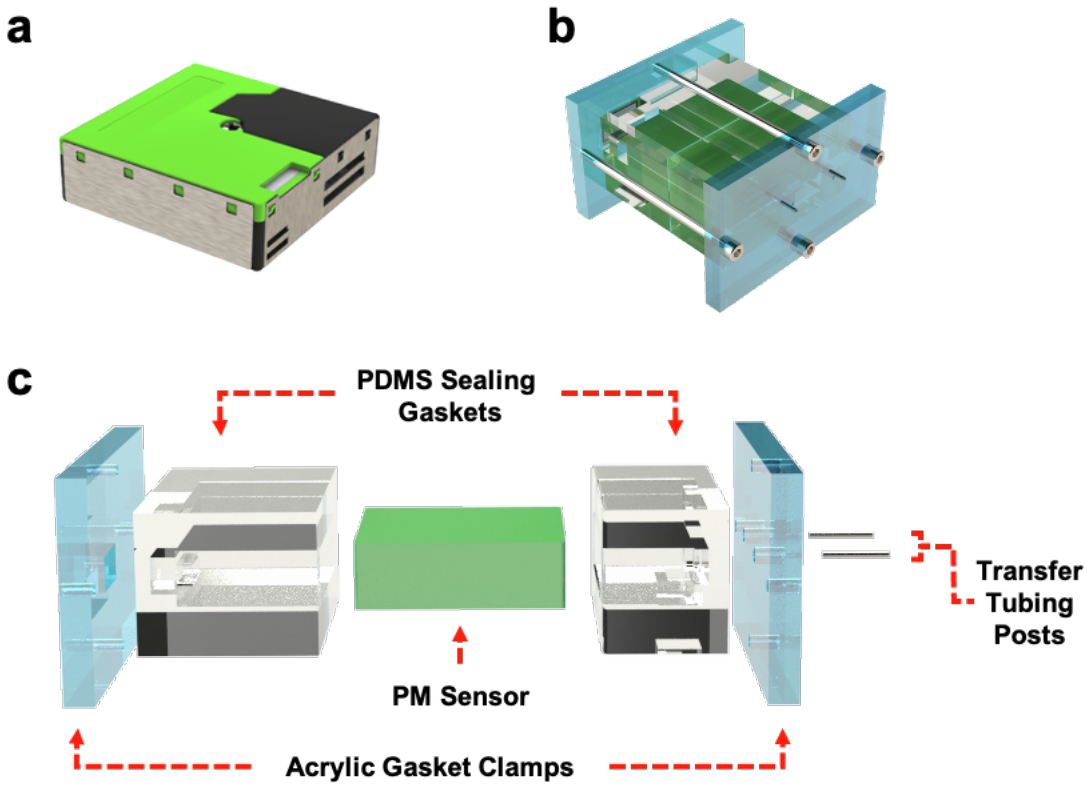

a
b
c
d
Small Filters
Airtight Tubing Adapters
Acrylic Plate to Separate Chamber
Airtight Interface for the Electronic Cigarette
10 mL Syringe
Tubing Adapters
Acrylic Filter Support
Vacuum Filter
Vacuum Pump
Small Filter
Acrylic Electronic Cigarette Stand
Electronic Cigarette
Acrylic Frame
Pinch Valves
Two-Channel Power Supply
Solid-State Relay
Arduino with an 8-channel shield

Supplementary Figure 1. Vaping Robot Design, Related to Figure 1. (**a**) The system used for these vapor exposure studies allowed for different vaping profiles reflecting both healthy and diseased lungs to be mimicked, enabling a smoking topography that would be present in an analogous (human) EC user. The system is divided into two chambers, the chamber that contains the ECs and the chamber that contains the pneumatics and electronics. (**b**) The vape exposure device that was used, delivered proper vapor dilution, with proper vaping profiles, to the PM Sensor in-line with the system. Inhaling vapor from any EC device is accomplished when negative pressure is created by the contracting of the diaphragm. Negative pressure in the vape exposure system is created by the vacuum pump, allowing vapor to be “inhaled” into the reservoir. (**c**) The reservoir that holds the vapor is a biocompatible syringe with a capacity analogous to vapor intake volume. The reservoir and pinch valves are holstered with a VeroClear™ 3D printed part for ease of access, operation, and maintenance. The reservoir can be removed, where the inlet and outlet connections can be optionally interfaced with the Dilution Robot, that contains its own internal reservoir. The vape exposure system is also equipped with various filters for clean air breaths and the incubator in which the system is contained, provides a filtered air supply. The inner shelving units and outer casing of the system were made from clear acrylic for visibility and for ease of use of the material. (**d**) The sampling rate and flow of vapor through the exposure system is controlled by a system of pinch valves and tubing. The pinch valves allow for different vaping profiles to be driven through the system via the firmware. The pneumatics system was designed in order to perform the following key functions: generate a sample of aerosolized vapor inside the reservoir, purge air from inside the

reservoir, flow vapor sample into the PM Sensor, take clean inter-puff interval inhales, and properly exhaust this vapor out of the system. The electronic hardware driving the system consists of an Arduino microcontroller, an 8-channel relay shield, a solid-state relay, a vacuum pump, four pinch valves, and a 2-channel power supply. A low current, Arduino compatible, relay shield is used to propagate the voltage needed to power all electrical components. This is needed because the Arduino only has a supply voltage of 5V, whereas most of the electronic components need upwards of 12V to operate. A solid-state relay is also used to prevent the vacuum from causing a short in the circuit. The vacuum required a higher current than the relay shield could provide, so it acts as an isolated switch that can use a low current circuit to switch on the higher current circuit powering the vacuum pump. The Arduino code used was developed to specifically respond to the Breathing-Emulator system driving the vapor into the PM Sensor used for this study.

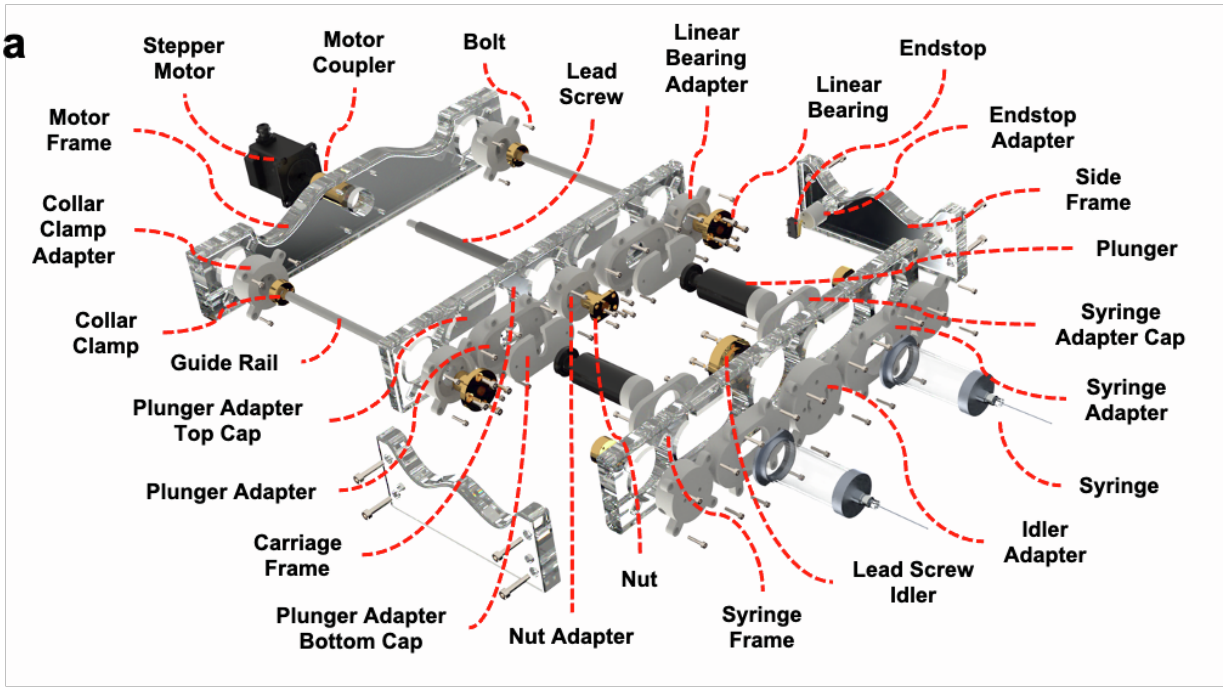

a
b
Airtight Gaskets
3D Printed
Nut
3D Printed
Chamber
c
2 mm
6.5 mm
9 mm
5 mm x 15 mm

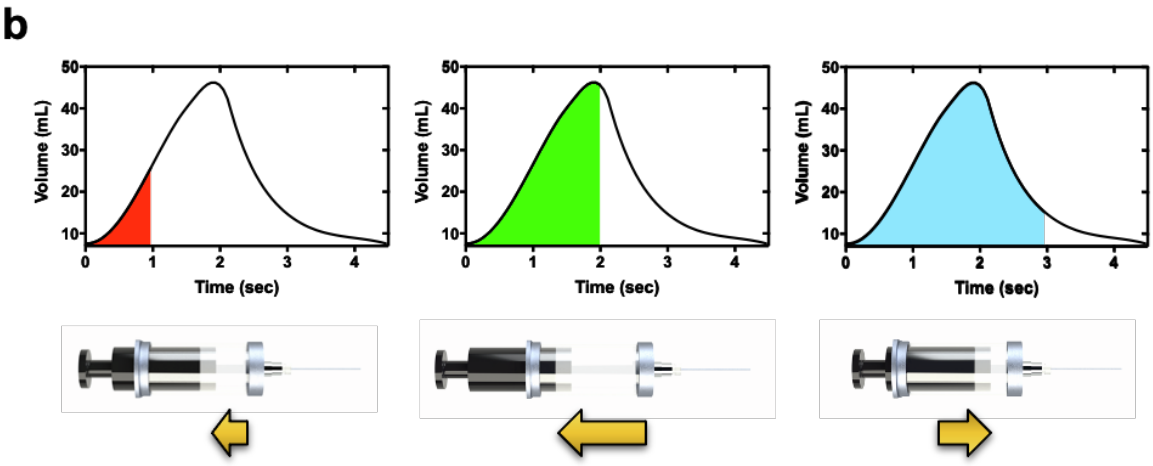
Supplementary Figure 2. Electronic Cigarette Sealing Gasket Design, Related to Figure 1. The vape exposure system was designed with the aim to closely mimic normal use by an EC user. The mouth is modeled using a Polydimethylsiloxane (PDMS) gasket to interface with the EC and a PolyJet 3D printed part to hold the gaskets in place. As well as being biocompatible, PDMS was chosen because it is inert, moldable and can provide an airtight seal. (**a**) Using PDMS as the interface also allows for quick and relatively cheap fabrication, meaning that various sizes and shapes of molds can easily be made to interface different ECs or tubing. The modularity of this design also allows for minor adjustments to be made quickly to account for small variances in tubing or cartridges. (**b**) Each gasket is designed to fit into a small 3D printed chamber with an airtight seal. Using a coupled 3D printed nut, we were able to fasten the outside of the chamber to the acrylic back plate to couple the generated vapor to the rest of the system. (**c**) This interface has been essential in testing and interfacing different types of EC cartridges with our system.


a
b
c
PDMS Sealing Gaskets
Transfer Tubing Posts
PM Sensor
Acrylic Gasket Clamps
d
Connector
Photoelectric Converter
Light Trap
Photodiode
Lens
Laser Source
Laser Drive Circuit
Air/Vapor Flow Direction
Inlet
Outlet

Supplementary Figure 3. Particulate Matter Sensor and Its Enclosure, Related to Figure 1. (**a**) The PM Sensor used was a Sensirion SPS30 optical sensor with internal air circulation. This PM Sensor was chosen for this application since it is capable of detecting a range of different particle sizes, from 300 nm to 10.0 μm. It can report mass concentration and particle count values for 300 nm – 1 μm, 1 μm – 2.5 μm, 2.5 μm – 4 μm, and 4 μm – 10 μm in real-time, allowing for direct monitoring of concentration levels during the experiment. (**b**) The factory casing for the PM Sensor features large inlet and outlet ports for increased airflow. In order for the PM Sensor to be compatible with our experimental setup, an airtight housing was designed, which isolated the inlet and outlet ports from one another. This allowed for the control of airflow through the sensor, and ensured that vapor samples passed through the PM Sensor were not being double counted, offering particle concentration data that is more representative of the sample analyzed at any particular time-point. (**c**) The housing was comprised of a cast polydimethylsiloxane (PDMS) shell, acrylic plates, and tube-interfacing posts. The PDMS shell consists of two pieces, each modeled after the sensor's factory casing, allowing for a flush, airtight fit against the sensor. In order to further ensure that there was no air leaking from the casing, the PDMS shell was compressed between a set of two acrylic plates using 40 mm bolts. The plates have cutouts for the insertion of two interface posts, which allow for the tubing used in the experimental setup to connect to the inlet and outlet of the sensor, as well as an opening for the electronics wiring to pass through. (**d**) The PM Sensor uses an internal channel to direct airflow from the inlet to pass through a sensing channel, across a laser. Once a particle passes over the laser, the particle causes the laser light to be

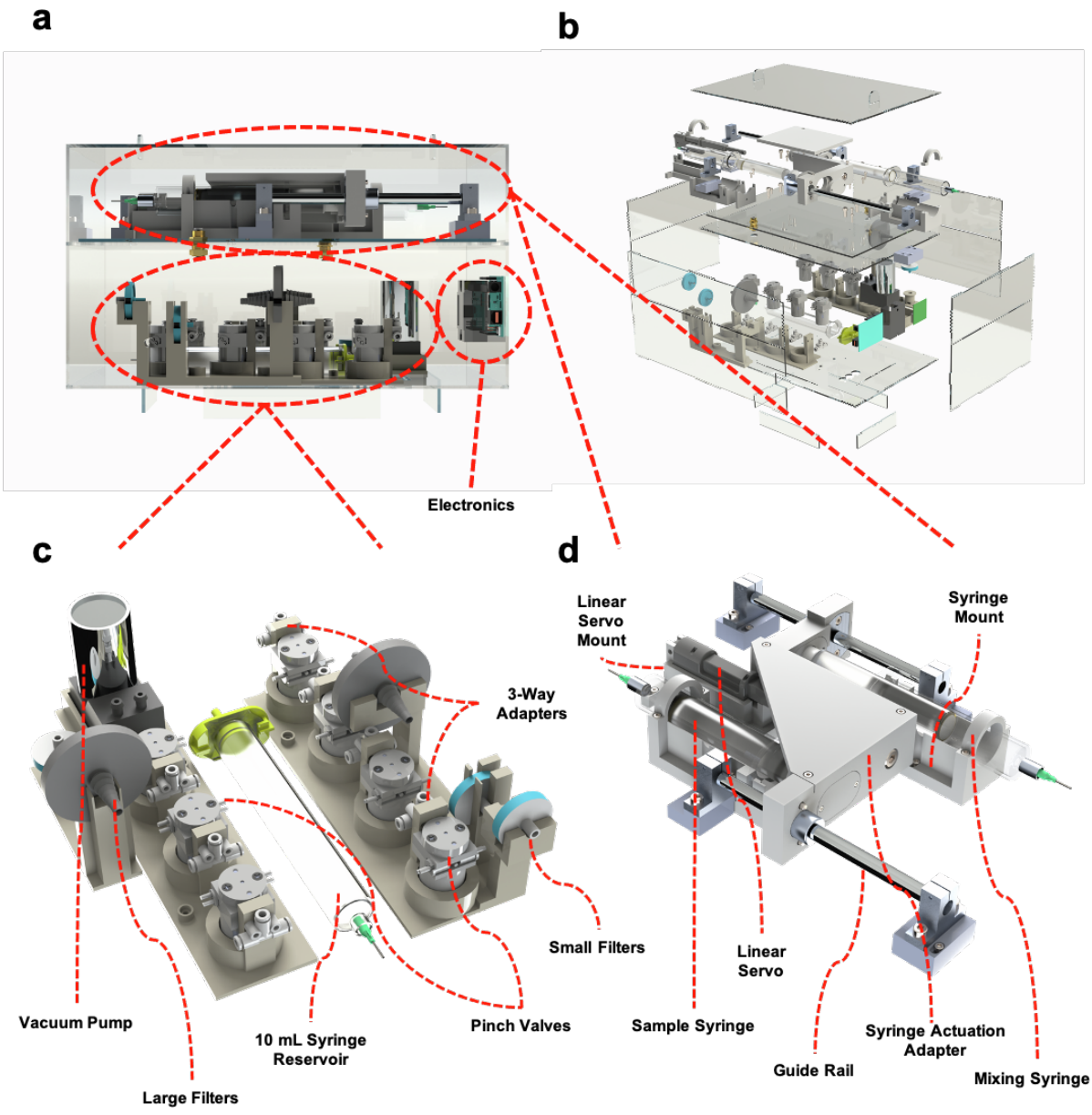
scattered, and this scattered light is sensed by a photodiode, which then sends a signal to a photoelectric converter that calculates the size of the particle sensed based on the amount of scattered light that was detected. This information is then sent to the connected computing unit. After passing through the sensing channel, the air collects in the outlet channel, where a fan helps prevent air from flowing back into the sensing channel, promoting unidirectional airflow through the sensor.


a
Stepper Motor
Motor Coupler
Bolt
Lead Screw
Linear Bearing Adapter
Endstop
Linear Bearing
Endstop Adapter
Motor Frame
Side Frame
Collar Clamp Adapter
Plunger
Collar Clamp
Syringe Adapter Cap
Guide Rail
Syringe Adapter
Plunger Adapter Top Cap
Syringe
Plunger Adapter
Idler Adapter
Carriage Frame
Lead Screw Idler
Nut
Plunger Adapter Bottom Cap
Nut Adapter
Syringe Frame


b
Volume (mL)
Time (sec)
50
40
30
20
10
0
1
2
3
4

Supplementary Figure 4. Breathing-Emulator Components and Functionality, Related to Figure 1. (**a**) The Breathing-Emulator consists of a frame and mechanical actuation components that drive a plunger back and forth to pull and push air from within air-tight syringes. The actuation process is controlled by a stepper motor, which connects to a motor coupler that turns a lead screw, which is connected to a lead screw idler to allow for the lead screw to freely rotate. The lead screw actuates a nut that is attached to the carriage through the nut adapter. The carriage is stabilized through two linear bearing adapters that connect to the linear bearings which glide along the two guide rails. This allows for the rotational movement of the motor to be converted into linear motion. While the carriage is actuated by the motor, plungers for the syringes are mounted to the carriage through the plunger adapters and are actuated simultaneously. The syringes are fixed and mounted to the syringe frame through the syringe adapters. In both the case of the plunger adapter and the syringe adapters, additional caps are attached, which pin the plunger and the syringe to the carriage and syringe frame respectively, this prevents displacement of either component from its adapter. One side of the frame mounts two endstops, which are on opposite sides of the carriage stroke limits. The endstops are mechanical switches that connect two electrical lines when activated and are used to detect the seek limits of the Breathing-Emulator, for both avoiding damage due to over seeking and for homing the system at boot. (**b**) The Breathing-Emulator operates based off of a generated set of data that describes the number of steps for the motor to take, a step is a small fraction of a rotation, and the delay rate between each step. This profile is described by a series of steps and delays that are paired to give dynamic actuation. The steps and delays are calculated based off

the mechanical and electrical attributes of the system and the desired breathing profile. The breathing profile is initially generated as a flow-volume loop and mathematically converted to volume as a function of time, through modeling the flow-volume loop as a first order non-linear differential equation. This allows for the syringe plunger to be actuated to displace the volume in the syringe to match the graph at the specific time points.


a
b
Electronics
c
3-Way
Adapters
Small Filters
Vacuum Pump
10 mL Syringe
Reservoir
Pinch Valves
Large Filters
d
Linear
Servo
Mount
Syringe
Mount
Linear
Servo
Sample Syringe
Guide Rail
Syringe Actuation
Adapter
Mixing Syringe

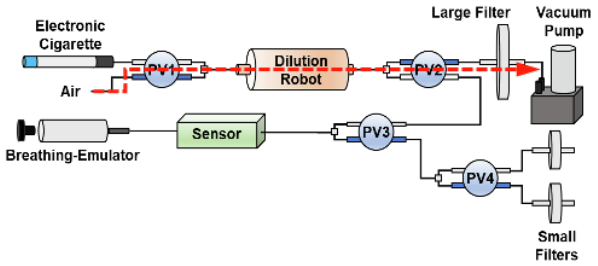
Supplementary Figure 5. Dilution Robot Mechanical Design, Related to Figure 1. (**a**) The Dilution Robot is capable of diluting the vapor produced from the Vaping Robot with filtered air to a specified ratio. The system achieves this through two syringes being actuated in a top chamber and a series of pinch valves that is responsible for controlling the fluidic flow in the bottom chamber. All electronics are also located in the bottom chamber. (**b**) An exploded view of Dilution Robot displays the mechanical frame and internal components of the dilution system. The frame is mounted on top of the Vaping Robot and is supported by fins on the bottom side of the Dilution Robot. (**c**) The bottom chamber pinch valve array unit contains 7 pinch valves and three-way adapters, a vacuum pump, 2 large filters, 3 small filters, and the 55 mL reservoir. The pinch valves are connected to one another following the fluidic flow diagram, with the 1st and 5th pinch valve connecting to the reservoir and the 3rd and 7th pinch valves connecting the upper chambers mixing and sampling syringes. All pinch valves, three-way adapters, filters, and the vacuum pump are mounted within a custom 3D printed support. (**d**) The top chamber consists of two 30 mL syringes, the mixing syringe and sampling syringe, each of these syringes are connected to ports that fluidically link to the bottom chamber pinch valve array unit. Both the syringes are mechanically coupled and driven by a single servo motor. Specifically, the syringes are oriented in opposite directions, so that when one syringe's plunger is being actuated to release air, the other syringe is taking in air, and vice versa. The plungers are actuated through linear motion of the syringe actuation adapter. The adapter is bolted into the linear servo, where the linear servo is resting on a custom 3D printed linear servo mount, that aligns the actuation arm with the bolt interface on the syringe actuation

adapter. The adapter is linearly guided by two guide rails on each edge of the adapter. The syringes rest in a custom syringe mount that aligns the barrel of the syringe with the plungers being supported by the syringe actuations adapter.

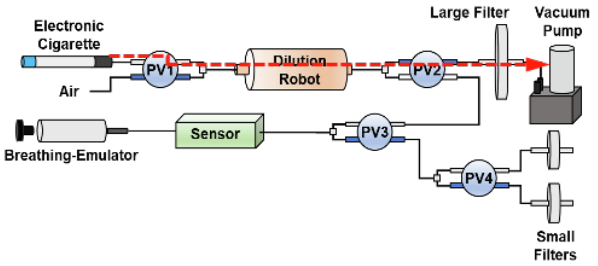
a

State	PV1	PV2	PV3	PV4	Vacuum Pump	Breathing-Emulator Sync
Purge	ON	ON	ON	N/A	ON	OFF
Vapor	OFF	ON	ON	N/A	ON	OFF
Sample	ON	OFF	OFF	N/A	OFF	Inhale
Inhale	N/A	N/A	ON	OFF	OFF	Inhale
Exhale	N/A	N/A	ON	ON	OFF	Exhale

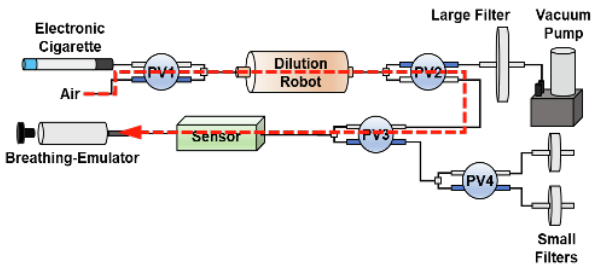
b



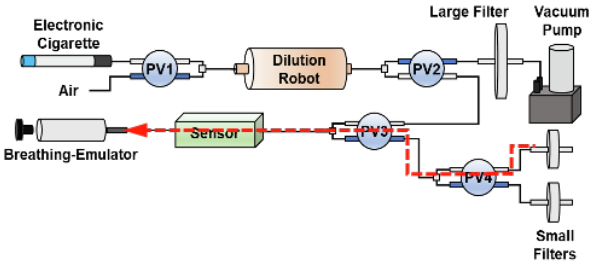
c



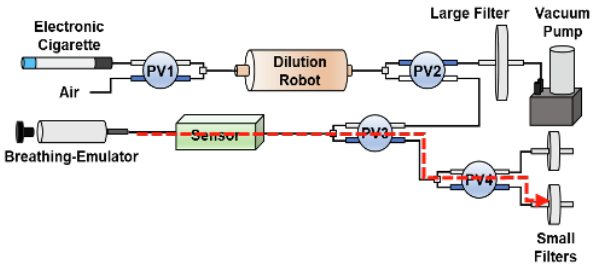
d



e



f



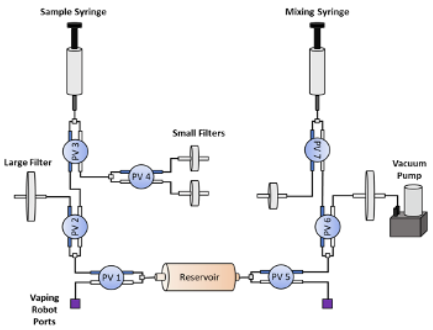
Supplementary Figure 6. Vaping Robot Control System and Fluidic Flow Diagram, Related to Figure 1. (**a**) There are four pinch valves and a vacuum pump that are regulated by an array of relays. The pinch valves are two-way pinch valves, where one line is pinched and the other is open, when the relay signal is toggled the lines that are pinched and open are toggled. The vacuum pump is active when the relay on signal is set and deactivated when the relay off signal is set. The system state machine is categorized into five states, Purge, Vapor, Sample, Inhale, and Exhale. The Purge state is responsible for evacuating vapor within the reservoir of the Dilution Robot. The Vapor state is responsible for filling the reservoir with EC vapor. Both the Purge and Vapor states run independently of the Breathing-Emulators position within a breathing cycle, additionally both states require the vacuum pump to be active. The Sample state is responsible for fluidically connecting the Breathing-Emulator to the reservoir in order to sample the vapor with the inline particle sensor. This occurs when the Breathing-Emulator is retracting the gas and thus requires being synced to the Inhale state. The Inhale and Exhale states can run independently or in conjunction with the above states and are synchronized with the Breathing-Emulator's respiratory cycle. Specifically, the Inhale state of the Vaping Robot is active when the Breathing-Emulator is retracting gas and the Exhale states is active during gas expulsion. Due to the Purge, Vapor, and exhale portion of the Sample state not requiring synchronization with the Breathing-Emulator the Inhale and Exhale states are running simultaneously while these states are active. The Sample state is synchronized with the Breathing-Emulator during the inhale portion of the breathing cycle and thus overrides the currently active Inhale state, specifically pinch valve (PV) 3. (**b**) The Purge state pulls air from the

Inhalation Exposure Chamber into the system through PV1, the reservoir within the Dilution Robot, PV2, the large filter, and then to the vacuum pump. During this time PV3 and PV4 are independently controlled by the Inhale and Exhale states. (**c**) The Vapor state activates the EC, through negative pressure, and pulls the vapor into the reservoir through PV1. The negative pressure is generated from the vacuum pump fluidically connecting the reservoir to the vacuum pump through PV2. Just as in the Purge state, PV3 and PV4 are controlled independently. (**d**) During the Breathing-Emulator's inhale, the Sample state fluidically connects the Breathing-Emulator to the reservoir through PV2 and PV3, which allows for the vapor to be sampled into the inline particle sensor. PV1 connects the reservoir to the air within the Inhalation Exposure Chamber to allow for pressure relief from the inhale. During the Breathing-Emulator's exhale the Sample state is no longer active and is overridden by the Exhale state. (**e**) The Inhale state fluidically connects the Breathing-Emulator to the inhale filter to allow purified air to be inhaled. The air flows from the filter through PV4, PV3, the particle sensor, to the Breathing-Emulator. (**f**) The exhale state fluidically connects the Breathing-Emulator to the exhale filter to expel filtered gas. The air flows from the Breathing-Emulator through the particle sensor, PV3, PV4, to the exhale filter.

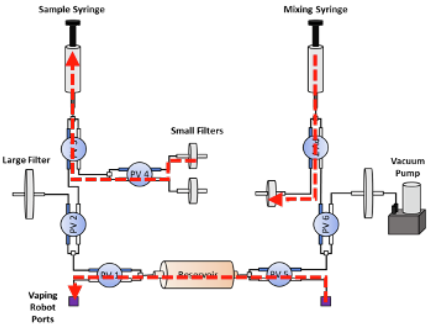
a

State	PV1	PV2	PV3	PV4	PV5	PV6	PV7	Vacuum Pump	Sample Syringe
VR Connect	OFF	N/A	OFF	ON	OFF	N/A	ON	OFF	Fill
Mix	ON	OFF	ON	N/A	ON	ON	OFF	OFF	Drain
Sample	ON	OFF	ON	N/A	ON	ON	OFF	OFF	Partial Fill
Evacuate	ON	ON	OFF	ON	ON	OFF	ON	ON	Fill
Empty	ON	ON	OFF	OFF	ON	ON	OFF	OFF	Drain

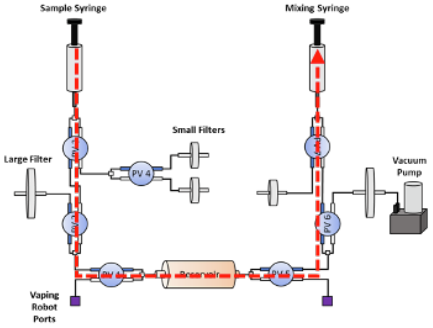
b



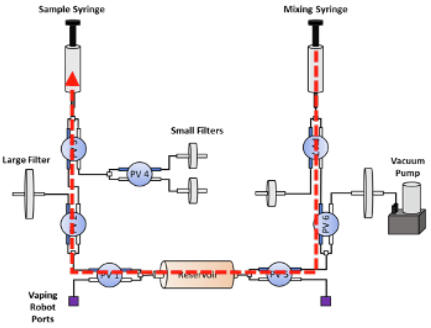
c



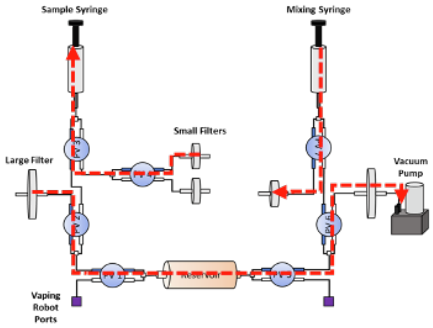
d



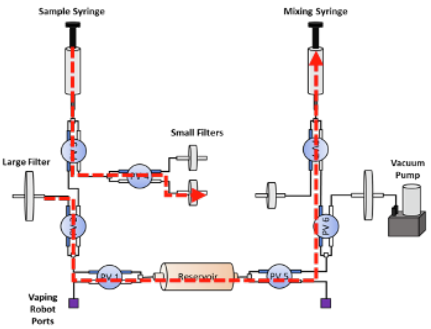
e



f



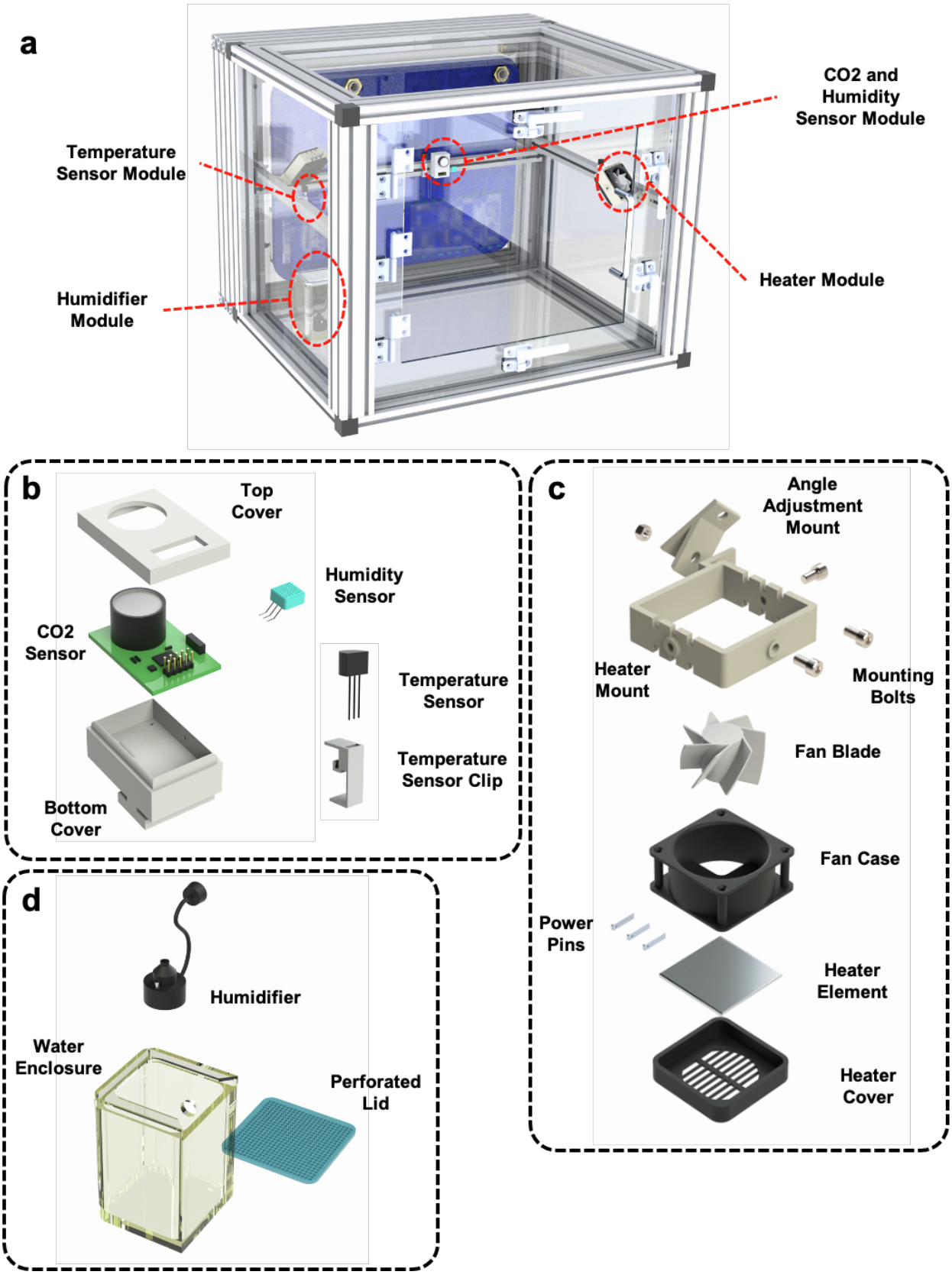
g



Supplementary Figure 7. Dilution Robot Control System and Fluidic Flow Diagram, Related to Figure 1. (**a**) The system has 5 individual states that can be transitioned to, which correspond to specific states the vacuum, pinch valves, and linear servo are in. The first state Vaping Robot (VR) Connect is a standby state. VR Connect is responsible for fluidically connecting the Dilution Robot reservoir to the Vaping Robot. This state is active when the Dilution Robot is waiting for the Vaping Robot to fill the reservoir with vapor. The Mix state is responsible for mixing the air within the reservoir without changing concentration or volume. The Sample state is responsible for sampling a calculated volume of vapor from the reservoir. The Evacuate state is responsible for removing all air in the mixing syringe, collecting fresh filtered air into the sample syringe and to remove all vapor for the reservoir. The Empty state is an optionally enabled state that pulls in air from the reservoir into the mixing syringe and exhausts all air in the sample syringe. (**b**) Similar to the Vaping Robot the Dilution Robot contains two-way pinch valves, where when one line is open the other is closed with the ability to switch between the open and closed line. The pinch valves are connected to three-way adapters in order to split and combine tubing. PV1 and PV5 are responsible for connecting the reservoir to the Vaping Robot or to the pinch valve array for the Dilution Robot. PV2 is responsible for connecting the reservoir to either the large intake filter or to the sample syringe. PV3 is responsible for connecting the sample syringe to reservoir or to the inhale and exhale filters, the selection of the exhale or inhale filter is through PV4. PV6 is responsible for connecting the reservoir to the vacuum pump or to the mixing syringe. PV7 is responsible for connecting the mixing syringe to the exhale filter or the reservoir. (**c**) VR Connect state is set on boot or transitioned to after the Mix state

or Evacuate state. During the VR Connect state PV1 and PV5 is connected to the Vaping Robot, PV3 and PV4 connect the sample syringe to the inhale filter, and PV7 connects the mixing syringe. After fluidically connecting the pinch valves the plunger is actuated fully to pull filtered air into the sample syringe and to evacuate air from the mixing syringe into the exhale filter. This state leaves 30 mL of filtered air into the sample syringe and the mixing syringe empty. (**d**) The Mix state transitions from the Evacuate state during the end of the dilution and is transitioned to from VR Connect during the start of the dilution procedure. The dilution starts and stop procedure is activated through the Start/Stop interrupt that is electrically connected to the Vaping Robot. The Mix state is responsible for transferring the filter air from the sample syringe through the reservoir and into the mixing syringe, through PV3, PV2, PV1, reservoir, PV5, PV6, and then PV7. (**e**) The Sample state is transitioned to from the Mix state. The Sample state pulls in a calculated, based off dilutions settings, volume of vapor from the reservoir into the sample syringe. During this time, the mixing syringe is actuated to keep the internal volume constant without introducing any further dilutions, since the mixing syringe also contains mixed vapor, due to the last state activated being the Mix state. The fluidic flow is through PV7, PV6, PV5, reservoir, PV1 PV2, and then PV3. (**f**) The Evacuate state is invoked after the Sample state and is responsible for further dilution of the vapor in the sample syringe while pulling in filtered air into the remaining 30 mL of volume in the sample syringe, in addition to emptying both the mixing syringe and the reservoir. The Evacuate state can also optionally be called after the Empty state, in this case the sample syringe is not filled with diluted vapor but filled only with filtered air. The fluidic connection is through the large intake filter to, PV2, PV1,

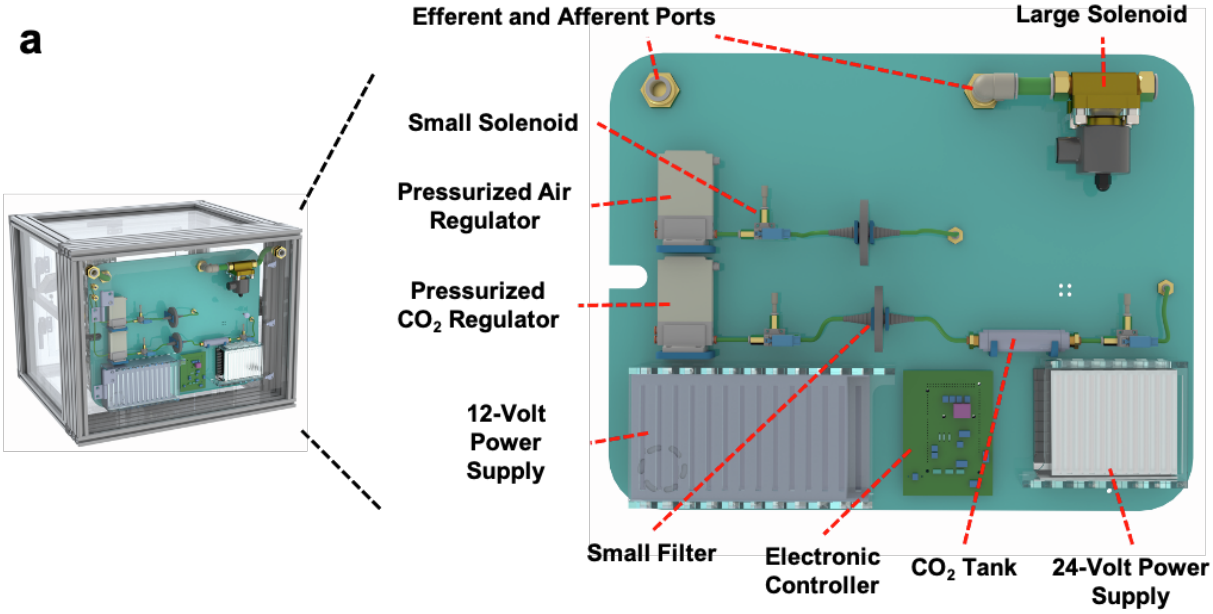
reservoir, PV5, PV6, through the vacuum pump filter and into the vacuum. Concurrently the sample syringe pulls filtered air from the inhale filter, through PV4, PV3, and into the sample syringe, and the mixing syringe evacuates air from the mixing syringe through PV4 into the exhale filter. (**g**) The Empty state is an optional state that can be invoked during the beginning of the standby procedure of the Dilution Robot. When the Vaping Robot has filled the reservoir with vapor and the Dilution Robot has diluted the vapor and the Vaping Robot has sampled the vapor into the particle sensor, the Vaping Robot activates the Stop Dilution procedure in the Dilution Robot. The Stop Dilution procedure can either transition directly to VR Connect, if the Vaping Robot is responsible for clearing the reservoir, or can transition to Empty, Evacuate, and then to VR Connect if the Dilution Robot is responsible for clearing the reservoir. The Empty state fluidically connects the large intake filter to PV2, PV1, reservoir, PV5, PV6, PV7, to the mixing syringe and connects the exhale filter to PV4 to PV5 to the sample syringe. The state empties the sample syringe and fills the mixing syringe. This allows the Evacuate state to be called next which will fill the sample syringe with clean air as it was fully evacuated from the Empty state.


a
CO2 and Humidity Sensor Module
Temperature Sensor Module
Heater Module
Humidifier Module
b
Top Cover
Humidity Sensor
CO2 Sensor
Temperature Sensor
Temperature Sensor Clip
Bottom Cover
c
Angle Adjustment Mount
Heater Mount
Mounting Bolts
Fan Blade
Fan Case
Power Pins
Heater Element
Heater Cover
d
Humidifier
Water Enclosure
Perforated Lid

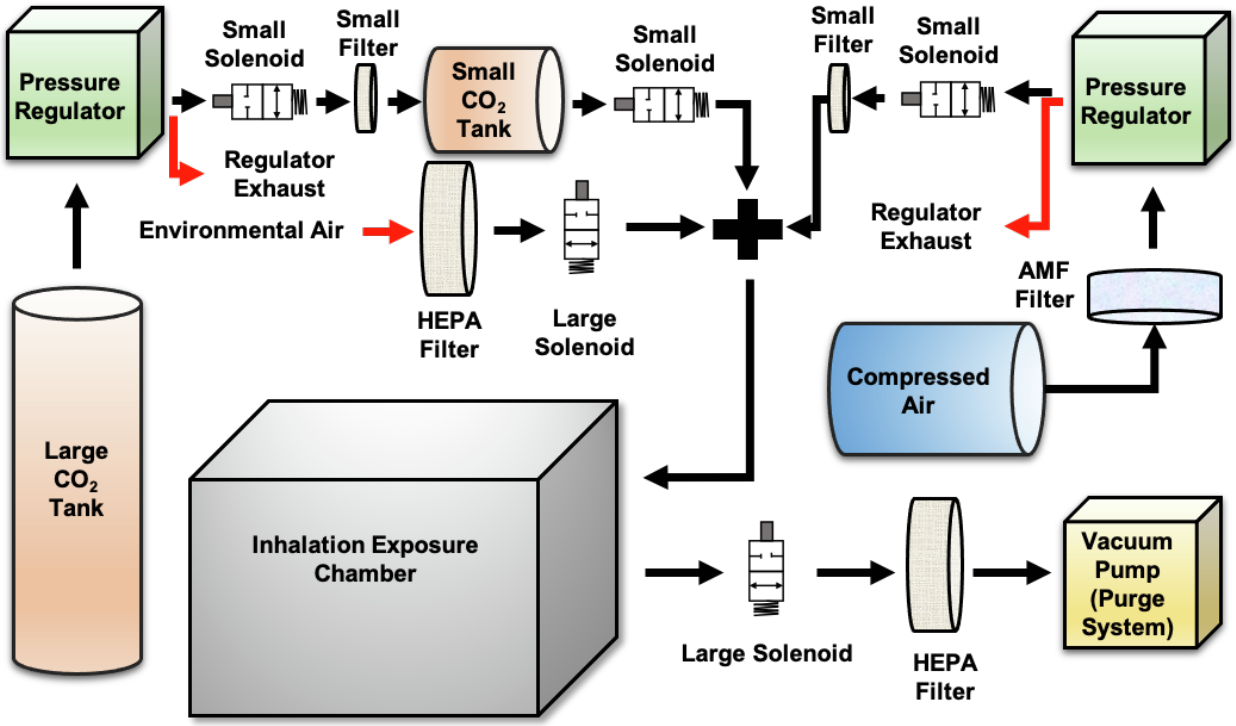
Supplementary Figure 8. Inhalation Exposure Chamber and Environmental Control Components, Related to Figure 1. (**a**) The Inhalation Exposure Chamber is a double paned acrylic enclosure supported by an aluminum frame and gaskets for sealing the chamber. Within the front of the Inhalation Exposure Chamber is the main door, which when latched shut, compresses a gasket to seal the chamber. Inside is a mounting rail on the back and side panels of the inner walls, on the mounting rails rests all the Inhalation Exposure Chamber sensors and the heating modules, the rails also are used to route wires from the components to the back where the component compartment is located. Each of the side walls has a heater module mounted near the corner, specifically mounted in the back corner on the left side and front corner on the right side. The carbon dioxide sensor and humidity sensor are located towards the center of the mounting rail on the back wall. The temperature sensors are mounted on the center of each of the three mounting rails and one in the top corner of the front panel, totaling to 4 temperature sensors. The humidity module is located on the left side of the Inhalation Exposure Chamber and is a non-mounted component that can be placed anywhere within the chamber. (**b**) The carbon dioxide module is a carbon dioxide sensor with a circuit board that the humidity sensor can plug into. The circuit board then communicates to the Inhalation Exposure Chamber controller through a serial port communication protocol. The carbon dioxide sensor is housed within a top and bottom cover, where the bottom cover slots into the mounting rail on the back wall. Similarly, the temperature sensors are mounted with a custom clip, which clips to the mounting rail. The temperature sensor communicates to the Inhalation Exposure Chamber's controller through a 1-wire communication protocol. (**c**) The heater module

flows air past a thermal element to heat the environment. There is a cover to protect from contact with the heater element, and both the heater element and the cover are adjacent to the fan case. Both the heater element and fan are powered by three power pins. Within the fan frame is a blade that rotates to move the air through the heater element. The fan and heater element are both mounted within a custom heater mount that is fixed with bolts. The heater mount is bolted to an angle adjustment mount which allows for the angle of the heater element to be swiveled in various directions. (**d**) The humidifier unit consists of a water enclosure container where a humidifier is submerged in water. The enclosure has a perforated lid that allows for the humidified air to exit the enclosure while preventing any larger droplets from escaping. The humidifier unit is powered through two wires which has an embedded plug to prevent leaking through the wire port of the water enclosure.

a


Efferent and Afferent Ports
Large Solenoid
Small Solenoid
Pressurized Air Regulator
Pressurized CO_2 Regulator
12-Volt Power Supply
Small Filter
Electronic Controller
CO_2 Tank
24-Volt Power Supply

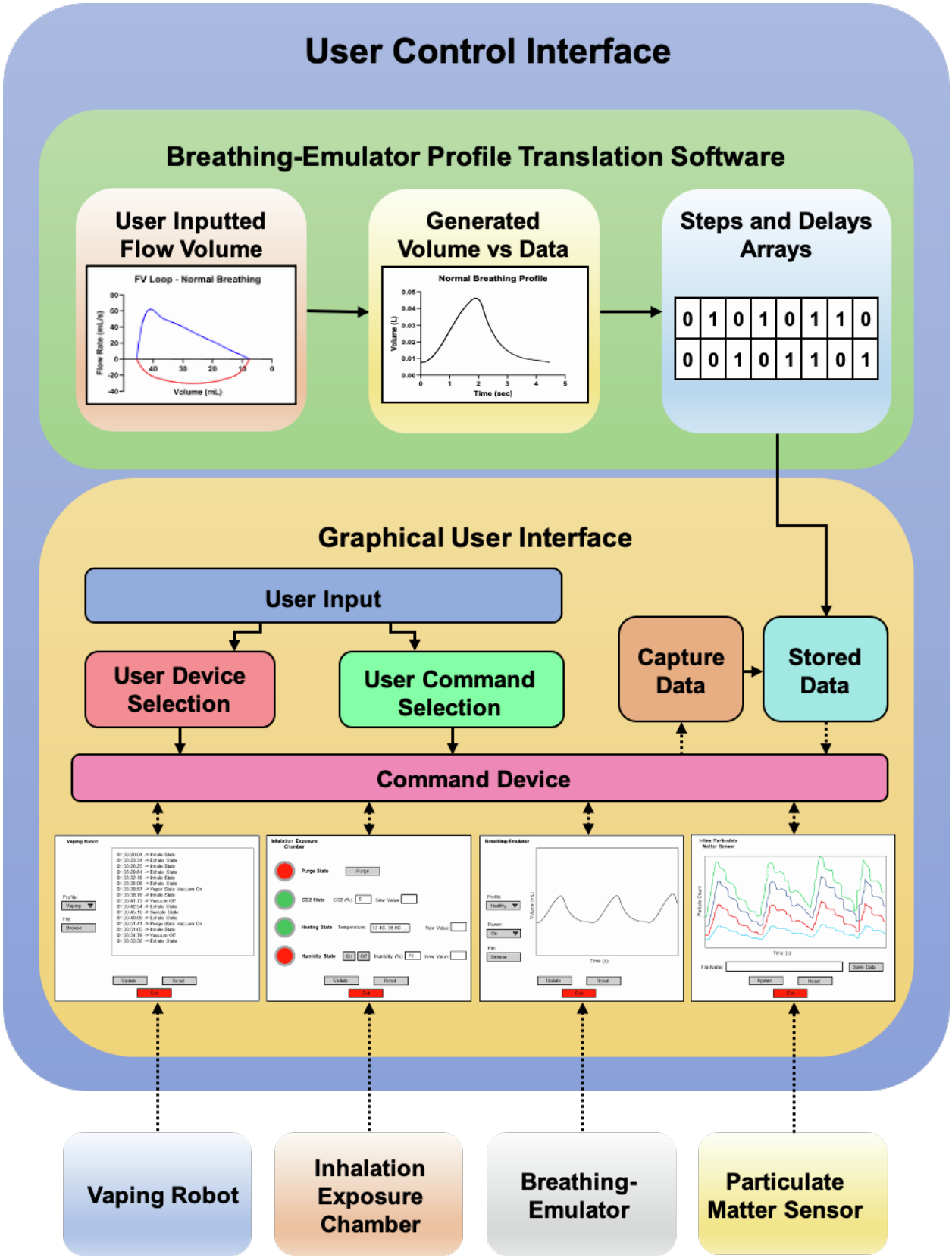
b


Pressure Regulator
Small Solenoid
Small Filter
Small CO_2 Tank
Small Solenoid
Small Filter
Small Solenoid
Pressure Regulator
Regulator Exhaust
Environmental Air
HEPA Filter
Large Solenoid
Regulator Exhaust
AMF Filter
Compressed Air
Large CO_2 Tank
Inhalation Exposure Chamber
Large Solenoid
HEPA Filter
Vacuum Pump (Purge System)

Supplementary Figure 9. Component Compartment System and Diagram for Inhalation Exposure Chamber, Related to Figure 1. (**a**) The Inhalation Exposure chamber has three afferent ports and one efferent port for air to flow in and out of the chamber. One set of afferent and efferent ports are for large air flow that is used during the purge activation of the Inhalation Exposure Chamber. The other two smaller afferent ports are for air and carbon dioxide injection, respectively. One of the large solenoids used is on-board the Inhalation Exposure Chamber and is activated as a pressure relief to take in filtered air during the purge activation. Both the pressurized air and the carbon dioxide gas is each connected to a pressure regulator that can dynamically change the pressure being applied to further downstream components. The output of each of the pressure regulators is connected to small solenoids that apply or disable the pressure in a binary fashion. The solenoid for the air pressure is used to inject filtered air into the chamber to allow for adjustments in air conditions within the Inhalation Exposure Chamber when either the temperature, humidity, or carbon dioxide levels are too high. The small solenoid for the carbon dioxide is used to pressurize a downstream tank, the tank is then connected to another small solenoid that controls the injection of carbon dioxide. All of these components mentioned as well as the sensors and devices in the main chamber are controlled through an on-board electronic controller, which is a custom printed circuit board plugged into an Arduino. The electronic controller is powered by both a 12-volt power supply, which is used to power the controller and most components, and a 24-volt power supply, which is used to power the humidifier module. (**b**) There are three main inputs into the Inhalation Exposure Chamber, filtered pressurized air, carbon dioxide, and filtered air drawn in by negative pressure induced

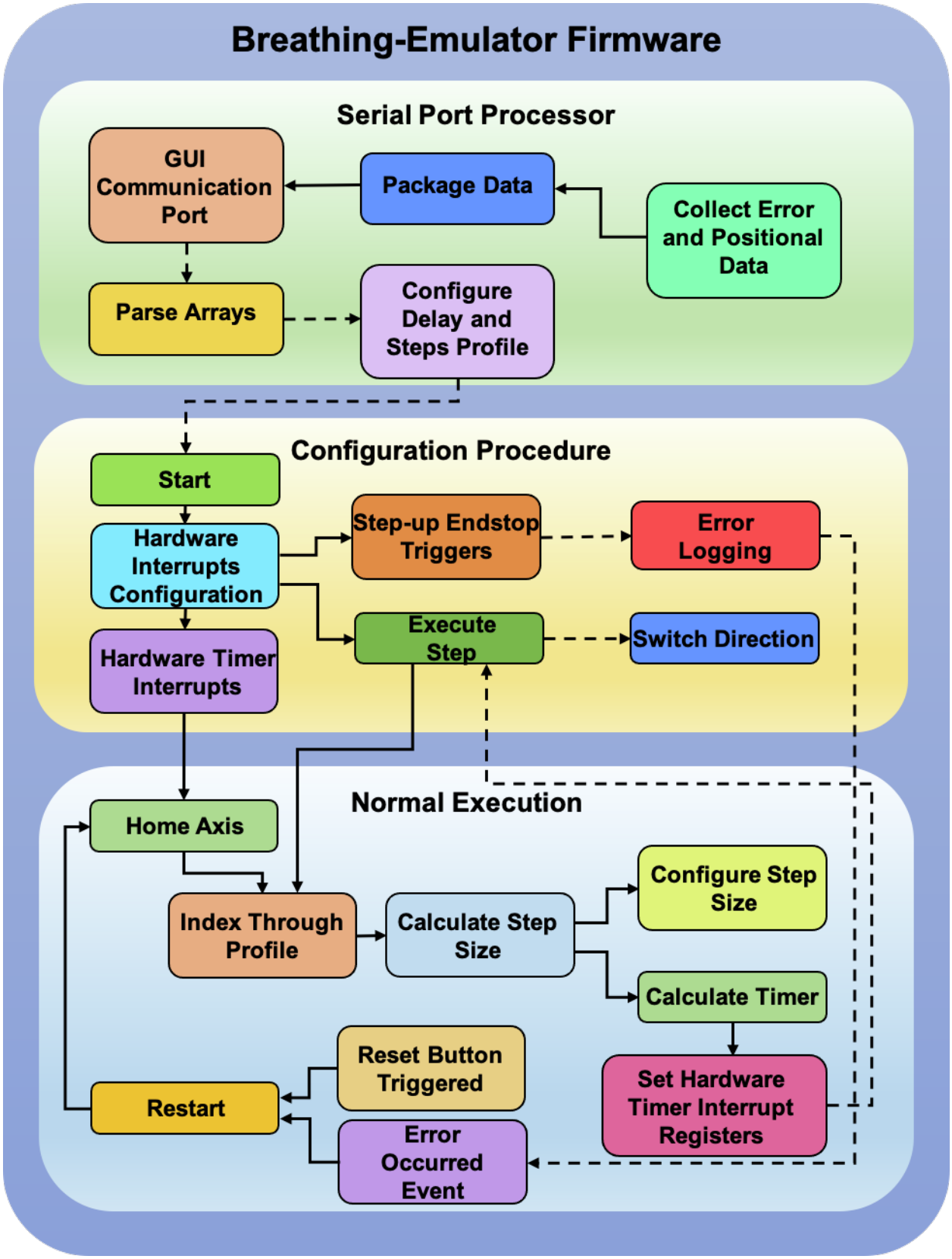
from the vacuum pump purge system. The carbon dioxide is provided by an external pressurized tank which flows through a pressure regulator, a small solenoid, a filter, and into a small tank. When the Inhalation Exposure Chamber has determined that more carbon dioxide is required the pressure regulator will adjust the pressure to the optimal calculated value, open the first solenoid to allow carbon dioxide to pressurize the small tank, once pressurized the first solenoid closes and the second solenoid opens allowing for a controlled amount of carbon dioxide to be injected into the Inhalation Exposure Chamber. The air injection system is similar, except that there is no tank or second solenoid, and the pressure is supplied by a pump that passes the air through a dehumidifying filter. The air injection capabilities are only being used as a fail-safe for if conditions become too elevated from the setpoints, as temperature, humidity, and carbon dioxide levels can only be reduced by adding in dehumidified ambient atmospheric filtered air. As a result, air injection is controlled by the regulator and first solenoid only, as the higher rate of injection required cannot be supplied by the small tank injection design used in the carbon dioxide injection system. In the event that the carbon dioxide or the air pressure regulators' output pressure exceeds the setpoint pressure, small amounts of gas will be injected into the environment (red arrows) to correct for the internal overshoot in pressure. The third afferent connection to the Inhalation Exposure Chamber is the pressure relief for the purge activation. Environmental air (red arrow) is pulled through a HEPA filter, through the large on-board solenoid, and into the chamber. The large solenoid is off except during purge activations or gas injections to prevent carbon dioxide from escaping the chamber. The Inhalation Exposure Chamber has a single efferent port that is connected to an external large

solenoid, a HEPA filter and a high flow rate vacuum pump, all of which make up the external purge system. Both large solenoids are activated in tandem with each other during the purge activation to prevent an internal pressure change from within the Inhalation Exposure Chamber.


User Control Interface
Breathing-Emulator Profile Translation Software
User Inputted Flow Volume
FV Loop - Normal Breathing
Flow Rate (mL/s)
Volume (mL)
Generated Volume vs Data
Normal Breathing Profile
Volume (L)
Time (sec)
Steps and Delays Arrays
0 1 0 1 0 1 1 0
0 0 1 0 1 1 0 1
Graphical User Interface
User Input
User Device Selection
User Command Selection
Capture Data
Stored Data
Command Device
Vaping Robot
Inhalation Exposure Chamber
Breathing-Emulator
Particulate Matter Sensor

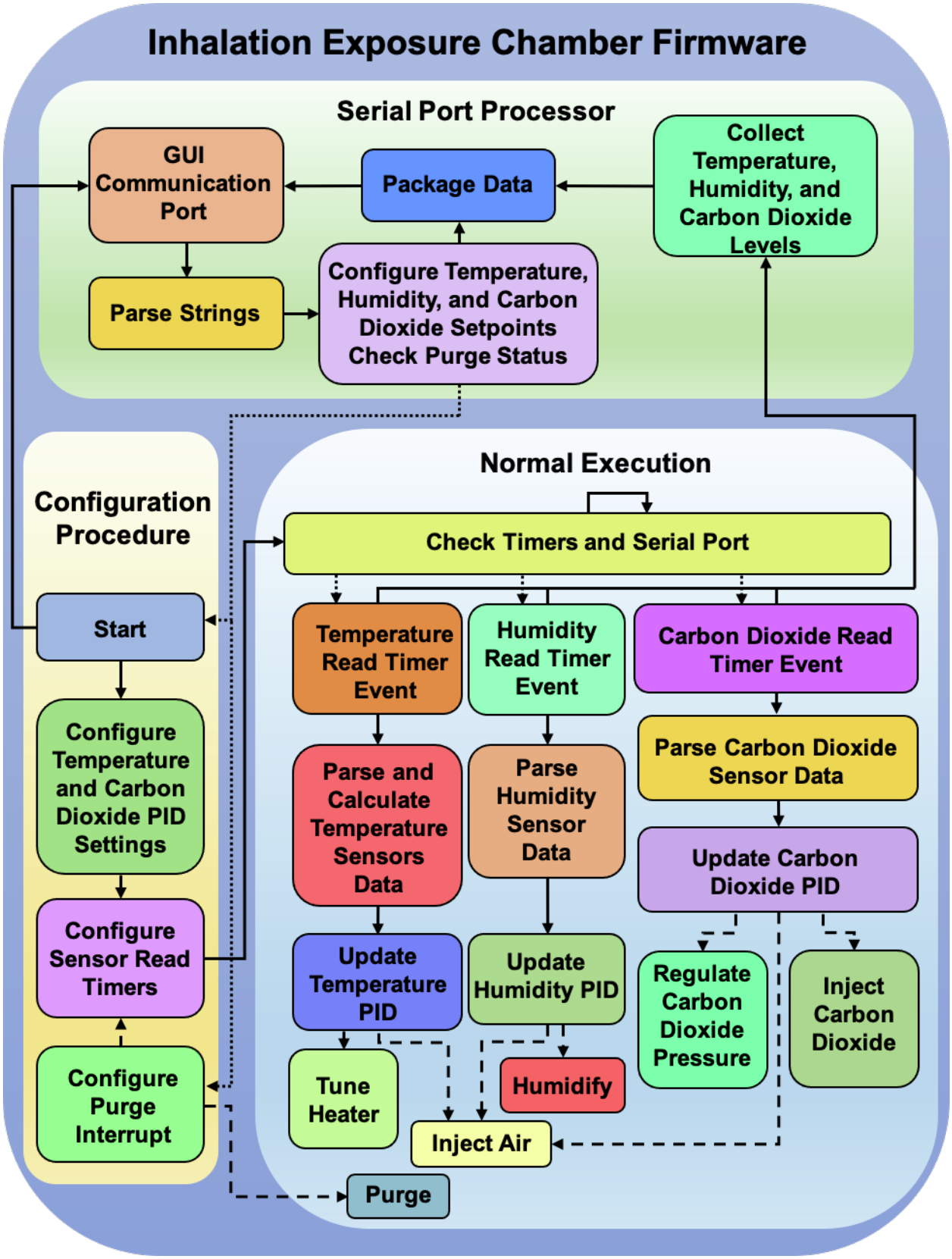
Supplementary Figure 10. User Control Interface Data Flow Diagram, Related to Figure 1. The User Control Interface is responsible for calculating the data that commands the Breathing-Emulator and is derived from the breathing state flow volume loop. The user saves generated flow volume loop profiles to a directory. When the translation software begins it loads the profiles and calculates the volume as a function of time by modeling the flow volume loop as a first order non-linear differential equation. This data is then processed into a series of delay rate and step counts, these parameters are derived from both the volume versus time profile and physical dimensions and properties of the Breathing-Emulator. The profiles are then saved to another directory to be accessed by users and other programs. In addition to running the Breathing-Emulator Profile Translation Software, the User Control Interface hosts the Graphical User Interface (GUI) which allows the user to interact with a custom software program to command the Breathing-Emulator, Inhalation Exposure Chamber, Inline PM Sensor, and the Vaping Robot. The GUI has four separate windows that each have a custom panel for all the custom devices connected. Each of the GUI modules, that correspond to a device, in tandem both bidirectionally communicate to the intended peripheral device through serial port communication while also bidirectionally communicating with the GUI device command interface layer. The GUI interprets the user's inputs to process the intended device, which is based off the window selected, and the commands for that device, which is controlled through the custom buttons and textboxes. Each of these outputs communicates with the device command interface layer, to then update the GUI windows and the peripheral devices if needed. The GUI is also capable of both loading and saving data. The GUI is capable of loading the data

generated from the Breathing-Emulator Profile Translation Software and communicating the data to the Breathing-Emulator in real-time. Data that is also output from the peripheral devices is communicated to the device command interface layer which then saves the data to be analyzed later. The Vaping Robot GUI window allows the user to select the vaping profile that will be sent to the Vaping Robot. Additionally, the GUI will capture and display the state transition data being transmitted from the Vaping Robot. The Inhalation Exposure Chamber allows the user to input settings to regulate both the temperature and the carbon dioxide and humidity levels, in addition to activating a purge of the chamber. The GUI also displays the current environmental variables within the chamber, which is processed by the output of the Inhalation Exposure Chamber. The Breathing-Emulator GUI allows the user to select the saved profile generated from the Breathing-Emulator Profile Translation Software and transmits it as a new profile to the Breathing-Emulator, the system also displays the volumetric displacement data of the Breathing-Emulator. The Inline PM Sensor GUI records the transmitted particle data and graphically displays the data as well as allows the user to save the data for post processing. All arrows in Supplementary Figure related to the firmware flow diagrams follow the same format, solid lines indicate when the module depicted as a block is called, it will call the pointed to module once per call, the thick dashed arrows indicate that the first module interacts with the pointed to module through hardware based method, and the thin dashes arrow indicates that the module call is conditional.


Breathing-Emulator Firmware
Serial Port Processor
GUI Communication Port
Package Data
Collect Error and Positional Data
Parse Arrays
Configure Delay and Steps Profile
Configuration Procedure
Start
Hardware Interrupts Configuration
Step-up Endstop Triggers
Error Logging
Hardware Timer Interrupts
Execute Step
Switch Direction
Normal Execution
Home Axis
Index Through Profile
Calculate Step Size
Configure Step Size
Calculate Timer
Reset Button Triggered
Set Hardware Timer Interrupt Registers
Restart
Error Occurred Event

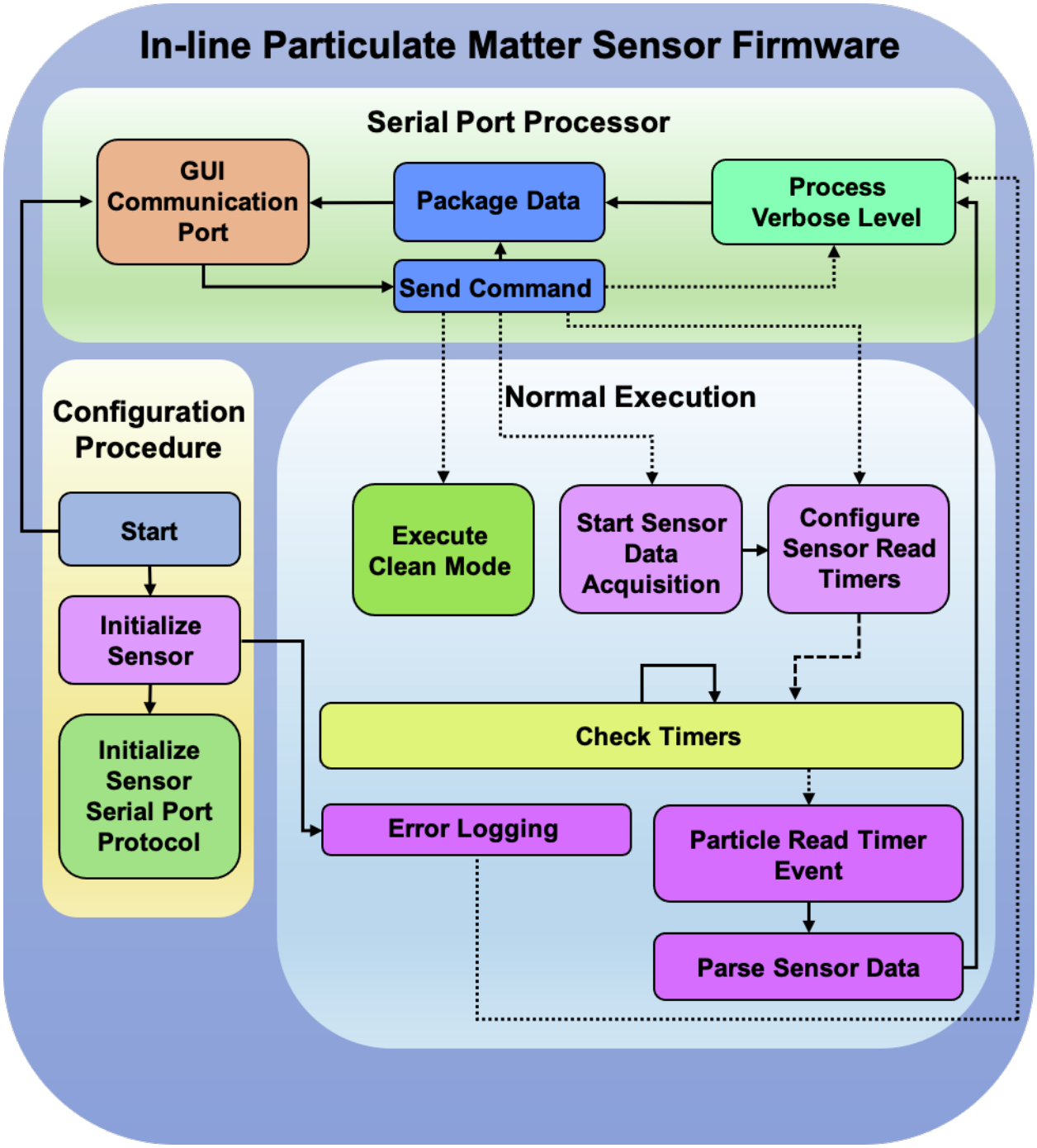
Supplementary Figure 11. Breathing-Emulator Firmware Flow Diagram, Related to Figure 1. All custom devices' firmware, including the Breathing-Emulator, can be categorized into three sections: the serial port processor, the configuration procedure, and the normal execution. When the system is first powered on, the firmware begins at the start module, which enables the GUI communication and initializes the hardware interrupts. The GUI communication line is now running in tandem with other sections of the firmware, where the port is constantly being monitored for data from the GUI, which is parsed and used to reconfigure the Breathing-Emulator profile. The communication port is also transmitting data to the GUI, where the data is generated by the Breathing-Emulator. The data includes both positional data and potential error messages. While the GUI communication port thread is running, the execution thread has configured the hardware interrupts, which consists of both the timer to execute a step and the interrupt to detect when the endstop has been triggered. After the timer interrupts are configured the home axis procedure is executed. The procedure configures the motor driver hardware and then takes controlled steps until the endstop has sent a signal indicating that it has been reached. After the home axis procedure has finished, the main loop of the normal execution cycle starts, where the breathing profile is indexed through one at a time, each time, executing a step. First the step size is calculated, alternating through different step sizes allows for an increased dynamic range of both velocity and precision. The step size data and the data in the current index of the breathing profile is then used to calculate the hardware timer values which control the rate of steps. Base off the timers values a step will occur at a programmed periodic rate. The steps are executed at the programmed rate, until the

number of steps in the current index was reached, at this point the profile index is incremented and the new step size is calculated. While this normal execution thread is running, the number of steps are also being counted and when the number of steps corresponding to a full stroke length are reached the motor control is configured to change directions, transitioning from inhale to exhale or vice versa. In the event that the endstop is triggered while not in the home axis procedure then the event is treated as an error. When errors are logged the information is passed up to the serial port processor layer as well as to the error event occurred module, which will then trigger a restart where the axis is homed before beginning the main loop. The restart event can also be triggered through the reset button being pressed on the board.

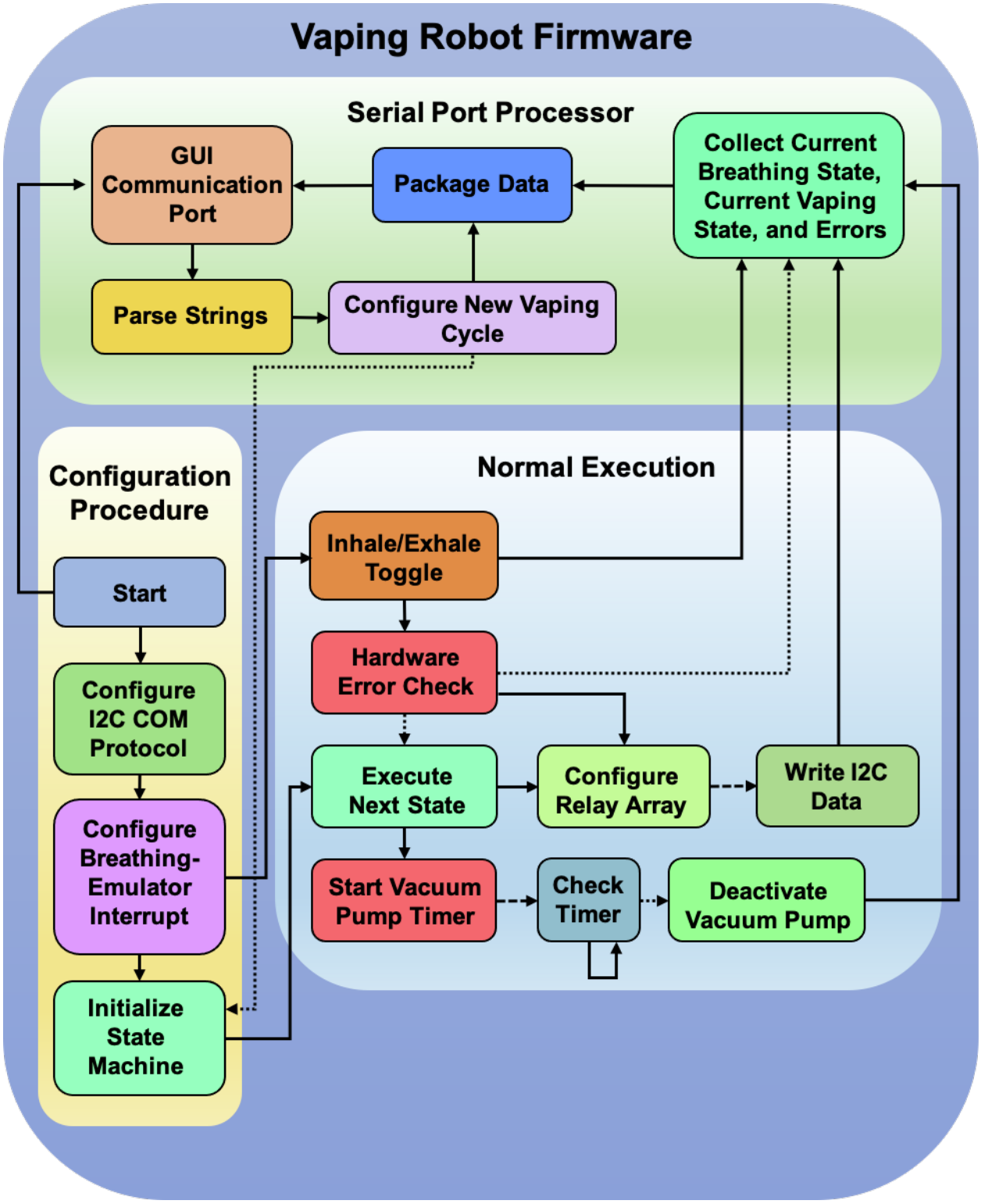

Inhalation Exposure Chamber Firmware
Serial Port Processor
GUI Communication Port
Package Data
Collect Temperature, Humidity, and Carbon Dioxide Levels
Parse Strings
Configure Temperature, Humidity, and Carbon Dioxide Setpoints Check Purge Status
Configuration Procedure
Start
Configure Temperature and Carbon Dioxide PID Settings
Configure Sensor Read Timers
Configure Purge Interrupt
Normal Execution
Check Timers and Serial Port
Temperature Read Timer Event
Humidity Read Timer Event
Carbon Dioxide Read Timer Event
Parse and Calculate Temperature Sensors Data
Parse Humidity Sensor Data
Parse Carbon Dioxide Sensor Data
Update Carbon Dioxide PID
Update Temperature PID
Update Humidity PID
Regulate Carbon Dioxide Pressure
Inject Carbon Dioxide
Tune Heater
Humidify
Inject Air
Purge

Supplementary Figure 12. Inhalation Exposure Chamber Firmware Flow Diagram, Related to Figure 1. Similar to the Breathing-Emulator firmware, the Inhalation Exposure Chamber firmware begins at the start module. The GUI communication port is then initialized which enables data being transferred from the GUI to be parsed and used to configure the Inhalation Exposure Chamber's settings, including temperature, humidity, and carbon dioxide setpoints, as well as wither the system should purge the chamber. When a setpoint is changed through the GUI the module will re-configure the setpoints. The module that configures the setpoints is also called after start. Once all setpoints have been configured, the sensor read timers are configured which are responsible for periodically triggering an event to read the data from the various sensors, this includes the temperature sensors, the humidity sensor, and the carbon dioxide sensor. When a read timer event has occurred, and the data is read and parsed from the sensors it is passed to the serial port processor which packages the data and sends the data to the GUI. Specifically, when the temperature read timer event occurs, four temperature sensors are parsed and averaged, the average temperature data is used to update the PID control system that regulates the power delivered to the heater module, which regulates both fan speed and heater power. When the humidity read timer has expired, the humidity sensor's output data is parsed and used to input into the humidity PID control system, which controls the humidity module. Similar to the other two modules, when the carbon dioxide read timer expires the carbon dioxide sensor data is parsed and used as an input into the PID control system. The carbon dioxide PID control system has the ability to both regulate the carbon dioxide pressure in the small tank and the periodic release rate of the

pressurized carbon dioxide from the tank into the chamber. Both the pressure setpoint and the injection rate will control the amount of carbon dioxide injected into the system. In the event that either the temperature, humidity, or carbon dioxide levels are too high, the designed non-passive method of lowering these values is by injecting filtered dehumidified ambient air, which will have both lower temperature, humidity, and carbon dioxide levels than the current and setpoint values. In the event that the purge state is activated through the GUI, the configure purge interrupt is called. This both activates the purge module to evacuate all the air within the chamber and suspends the data acquisition and processing of all the sensors, which prevents the system from interpreting the rapid change in the environment as part of normal operation. Once the purge is complete the system is restored to normal operation.

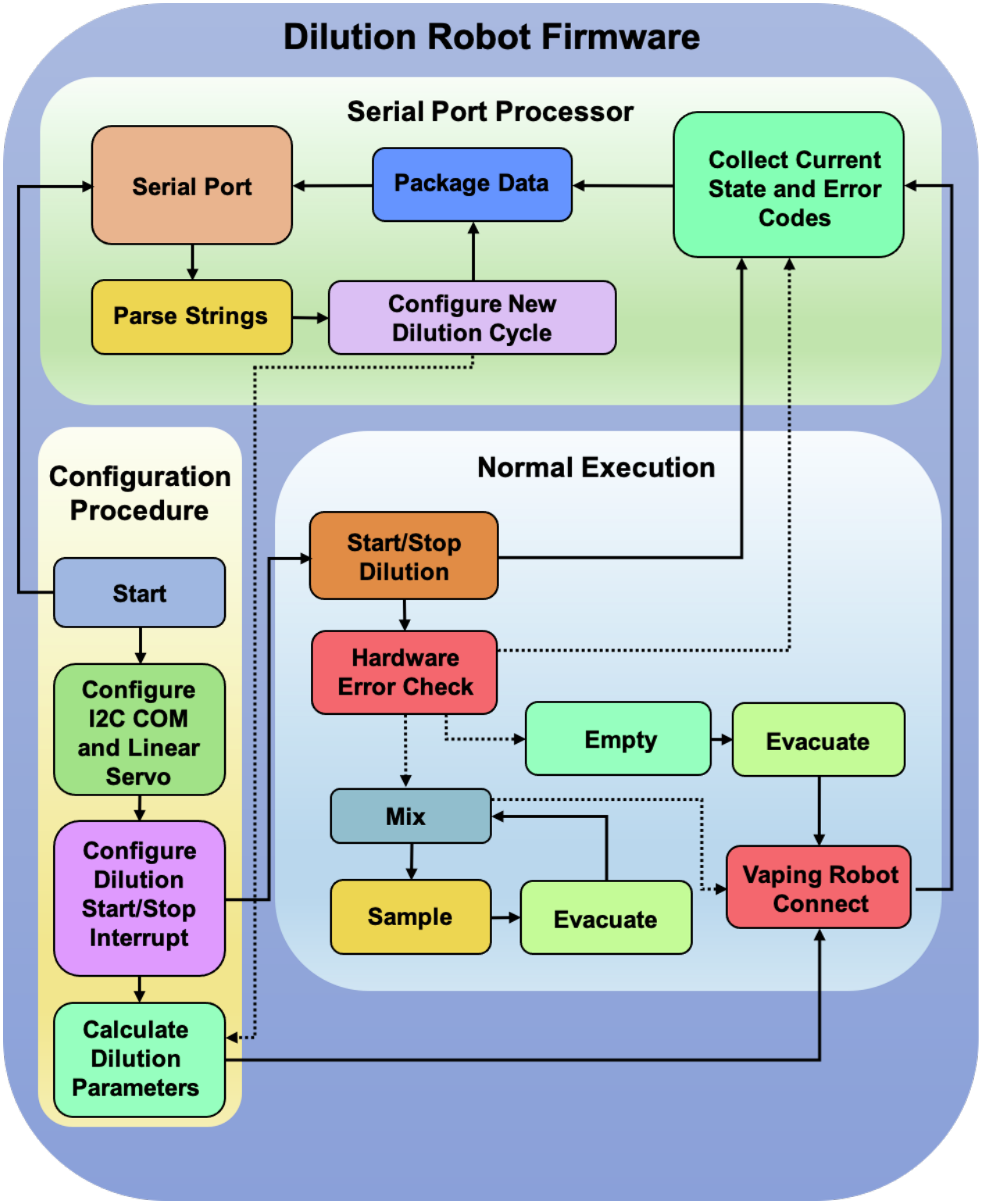

In-line Particulate Matter Sensor Firmware
Serial Port Processor
GUI Communication Port
Package Data
Process Verbose Level
Send Command
Configuration Procedure
Start
Initialize Sensor
Initialize Sensor Serial Port Protocol
Normal Execution
Execute Clean Mode
Start Sensor Data Acquisition
Configure Sensor Read Timers
Check Timers
Error Logging
Particle Read Timer Event
Parse Sensor Data

Supplementary Figure 13. Inline Particulate Matter Sensor Firmware Flow Diagram, Related to Figure 1. As with the other modules, while the system is initializing the GUI communication port in a newly created thread, the PM Sensor is initialized. During this process there are multiple error checks that if triggered will report to the verbose level processor which will select and package the data to send to the GUI. If there are no errors, the system will initialize the sensor serial port protocol. At this point the system is ready for normal execution. When a command comes in from the GUI to begin collecting data, the command does not need to be parsed as the GUI automatically sends the command in the format that the sensor is expecting. There are four main commands: execute clean mode, start sensor data acquisition, configure read sensor timers, and set verbose level. The execute clean mode command is used to clean the sensor after heavy use. The start sensor data acquisition command is used to begin collecting data. The configure sensor read timers is used to program the sample rate of data, if start sensor data acquisition is called it will set a default value for the configure sensor read timers. The set verbose level command is used to configure the process verbose level module, which filters out messages and data based off the level of information set to be reported. Once the sensor data acquisition has started and the sensor read timers have been configured, the system will periodically sample the data from the PM Sensor and parse it into both the particle counts and concentrations for all the particle distribution sizes. This data is then passed to the process verbose level module, which will package the data and send it to the GUI.


Vaping Robot Firmware
Serial Port Processor
GUI Communication Port
Package Data
Collect Current Breathing State, Current Vaping State, and Errors
Parse Strings
Configure New Vaping Cycle
Configuration Procedure
Start
Configure I2C COM Protocol
Configure Breathing-Emulator Interrupt
Initialize State Machine
Normal Execution
Inhale/Exhale Toggle
Hardware Error Check
Execute Next State
Configure Relay Array
Write I2C Data
Start Vacuum Pump Timer
Check Timer
Deactivate Vacuum Pump

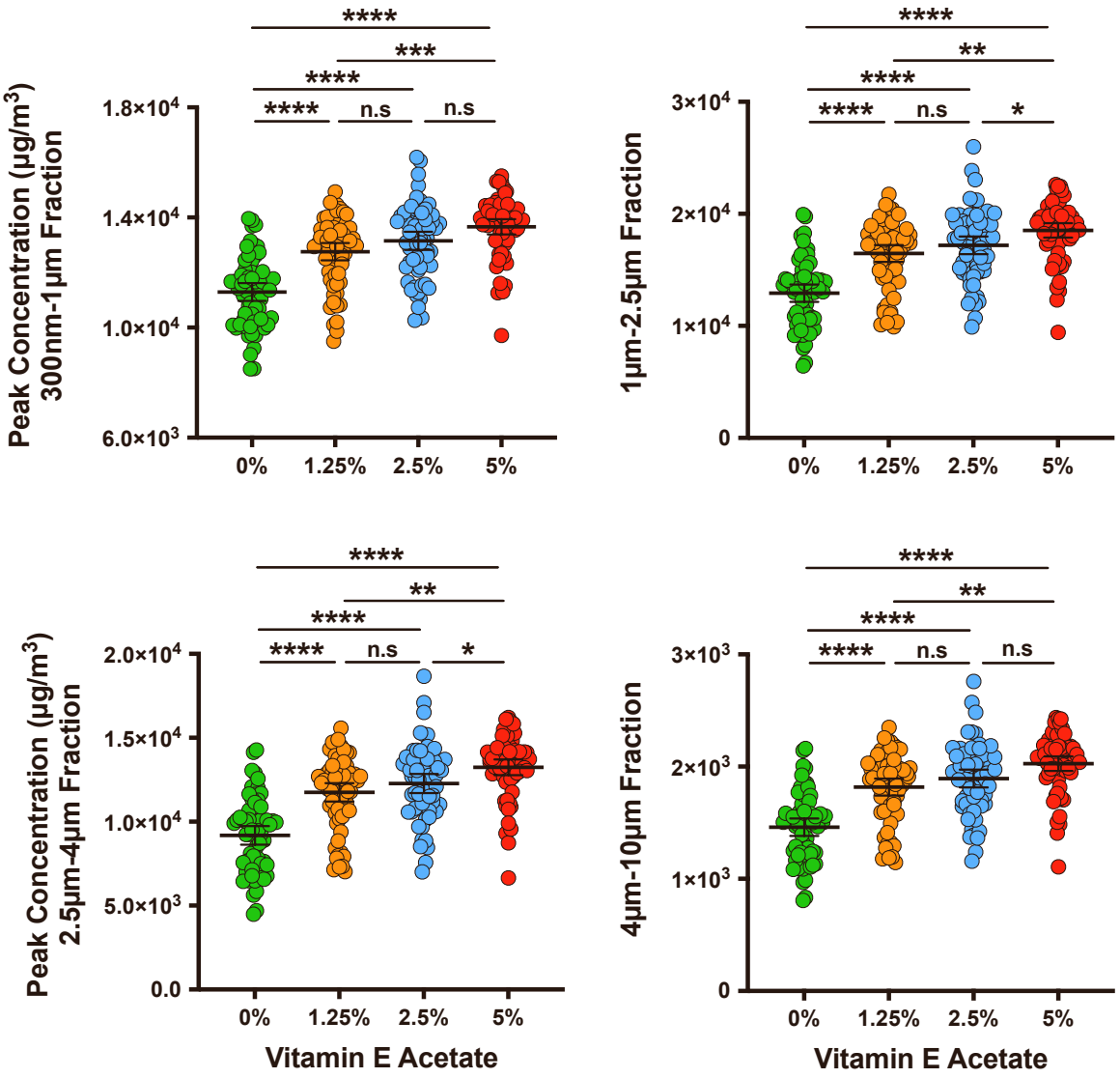
Supplementary Figure 14. Vaping Robot Firmware Flow Diagram, Related to Figure 1. Once the start module has configured the GUI communication port, the system will configure the I2C COM protocol, followed by configuring the Breathing-Emulator interrupt, which is responsible for configuring the hardware to detect the transition from inhale to exhale and exhale to inhale. After, the initialize state machine module will be called which is responsible for configuring the firmware to traverse to the appropriate state when a breathing cycle has finished. Initialize state machine is called once after start and is also called and updated with a new profile when a new profile is parsed from the GUI. After initialize state machine is called, the system is in normal operation. The state machine transitions with the execute next state module, which is dictated by the breathing cycle, specifically when the system changes from exhale of the last cycle to inhale of the new cycle. Due to the physical nature of the electronics there are checks on the hardware to validate that there was not an issue with detecting the transition from inhale to exhale or exhale to inhale. If errors are detected this information is corrected and the corrected error information is passed to the serial port processor which will package the data and send it to the GUI. Similarly, just as the state machine is being controlled by the transition of one breathing cycle to the next, controlling the fluidic connection of the inhale and exhale filters are also dictated by the breathing cycle. Specifically, when the inhale/exhale toggle module is triggered and any errors that occurred were corrected, this calls the configure relay array module, which sets the data to toggle the pinch valve responsible for controlling the inhale and exhale filter connection. While the configure relay array module is called every time due to the inhale/exhale toggle module, the execute next state module is only called half the time.

When the execute next state module calls the configure relay array module the same iteration as the inhale/exhale toggle module, the execute next state will override any conflicting pinch valve configurations, which only occur during the Breathing-Emulator's inhale while the Sample state is active. This relay configuration is packaged and written to both the I2C port to activate the relays, as well as the serial port processor to inform the user of the event, through the GUI. Depending on the current active state, the vacuum pump may also be required to be activated, specifically during the Vapor and Purge states. When this is required, the vacuum pump is activated at the start of the state transition, however the timer for the vacuum pump is independent of the breathing cycle and is on for a programmed amount of time, 3 seconds in our experiment. Thus, when the start vacuum pump timer module is called, the vacuum pump is activated, and the timer, which is on a separate thread, waits until it has expired before deactivating the vacuum pump. When the system toggles the inhale/exhale state, detects hardware errors, writes data to the relays through the I2C port, or activates and deactivates the vacuum pump this information is reported to the serial port processor which is passed to the GUI.

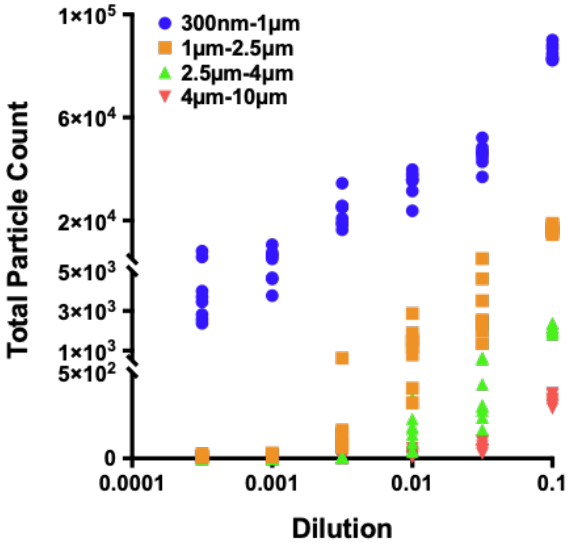

Dilution Robot Firmware
Serial Port Processor
Serial Port
Package Data
Collect Current State and Error Codes
Parse Strings
Configure New Dilution Cycle
Configuration Procedure
Start
Configure I2C COM and Linear Servo
Configure Dilution Start/Stop Interrupt
Calculate Dilution Parameters
Normal Execution
Start/Stop Dilution
Hardware Error Check
Empty
Evacuate
Mix
Sample
Evacuate
Vaping Robot Connect

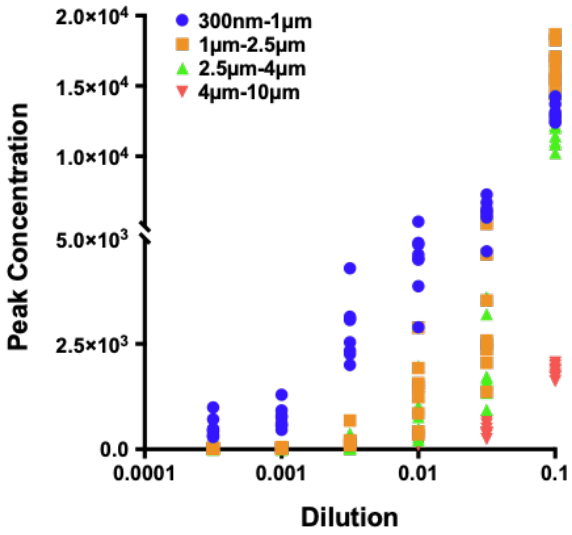
Supplementary Figure 15. Dilution Robot Firmware Flow Diagram, Related to Figure 1. The boot sequence of the Dilution system starts with configuring the main serial port and the I2C serial port communications in addition to configuring the linear servo. Next the firmware configures the Start/Stop interrupts. This interrupt is electrically connected to the Vaping Robot. The begin dilution procedure, Start Dilution, is invoked by the Vaping Robot after the Vaping Robot has filled the Dilution Robot's reservoir with vapor. The standby dilution procedure, Stop Dilution, is invoked by the Vaping Robot after the Vaping Robot has finished the Sample state, where the vapor was delivered to the PM Sensor. The standby dilution procedure can be configured to directly transition to the VR Connect state if the Vaping Robot is responsible for removing the vapor from the reservoir. Or the standby dilution procedure can invoke Empty, Evacuate, and then VR Connect, which will empty the reservoir with the Dilution Robot. After the Start/Stop interrupt is configured the system calculates the dilution parameters based off the programmed dilution ratio, some of the parameters include sample of vapor drawn into the sample syringe during the Sample state and number of serial dilutions to achieve setpoint. Once the dilution parameters have been calculated the system transitions to the VR Connect state and begins normal execution. Once the Start interrupt has been activated, hardware checks are performed, and the Mix state is invoked. Then the Sample state is called followed by the Evacuate and the Mix state again. Depending on the number of serial dilutions calculated the Sample, Evacuate, then Mix states can be repeated again until the targeted dilution is reached, at that point the system invokes the VR Connect state. State transition information is sent to the serial port processor along with any hardware errors that were detected. When the Stop Dilution interrupt is

activated, hardware checks are performed and the system can either invoke VR Connect directly or call the Empty, Evacuate, and then VR Connect states. The serial port processor is constantly monitoring the main serial port. The user can enter command(s) to invoke the Start/Stop Dilution interrupt in addition to changing the dilution ratio setpoint. Data that is sent from the hardware checks and state transitions is also sent out of the main serial port for diagnostics.

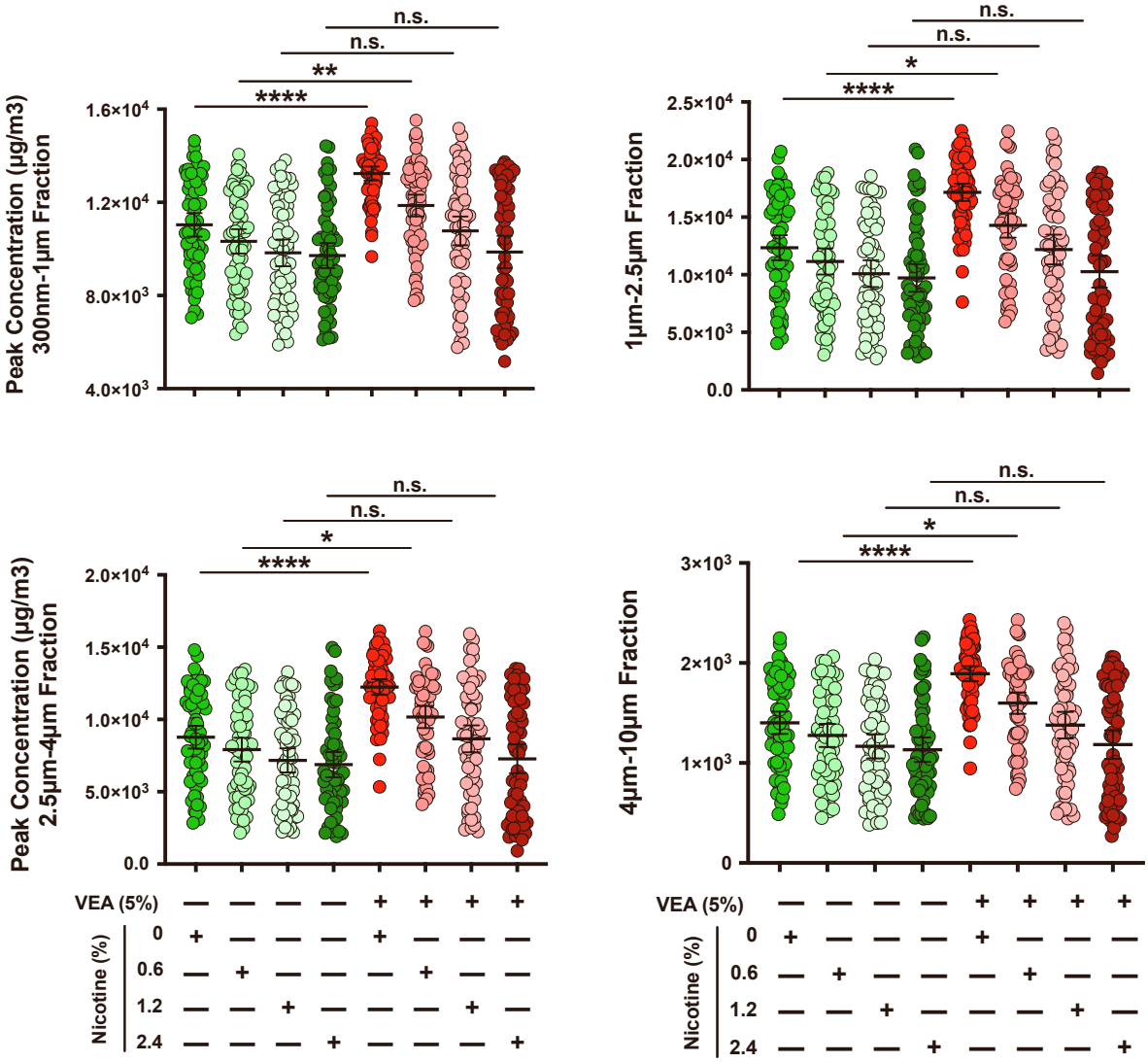

Peak Concentration (µg/m3) 300nm-1µm Fraction
1.8×10^4
1.4×10^4
1.0×10^4
6.0×10^3
****
***
****
****
n.s
n.s
0%
1.25%
2.5%
5%
1µm-2.5µm Fraction
3×10^4
2×10^4
1×10^4
0
****
**
****
****
n.s
*
Peak Concentration (µg/m3) 2.5µm-4µm Fraction
2.0×10^4
1.5×10^4
1.0×10^4
5.0×10^3
0.0
****
**
****
****
n.s
*
Vitamin E Acetate
4µm-10µm Fraction
3×10^3
2×10^3
1×10^3
0
****
**
****
****
n.s
n.s
Vitamin E Acetate

Supplementary Figure 16. Impact of Vitamin E Acetate on Inhaled Particle Peak Concentrations, Related to Figure 2. Plotted Distributions of the peak concentration over each inter-puff interval for particles within 300 nm – 1 μm, 1 μm – 2.5 μm, 2.5 μm – 4 μm, and 4 μm – 10 μm in size for the different VEA concentrations (v/v at 0%, 1.25%, 2.5%, and 5% all prepared in 50/50 PG/VG). Kruskal-Wallis test demonstrated statistical significance ($p < 0.0001$). Post-hoc analysis (Dunn's multiple comparison test) demonstrated statistically significant differences between all data sets except 1.25% and 2.5% for all particle size distributions and between 2.5% and 5% for the particle distribution of 4 μm – 10 μm . Bars show mean and 95% confidence intervals for each data set.

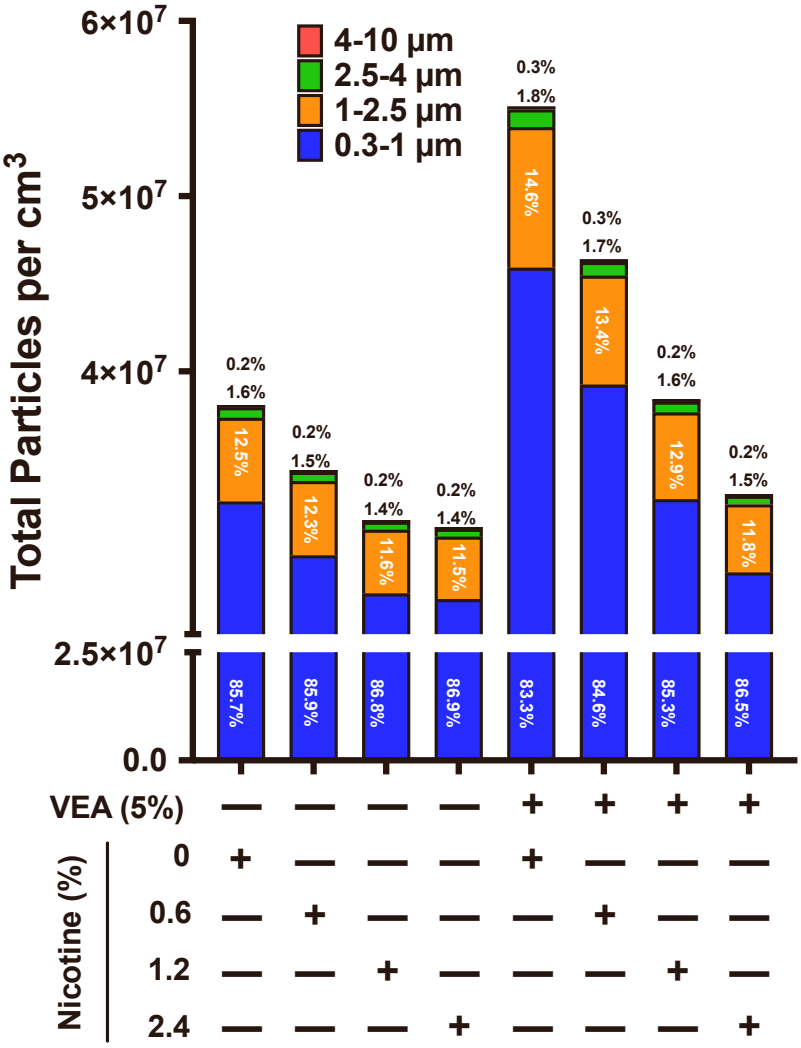

300nm-1µm
1µm-2.5µm
2.5µm-4µm
4µm-10µm
Total Particle Count
1×10^5
6×10^4
2×10^4
5×10^3
3×10^3
1×10^3
5×10^2
0
0.0001
0.001
0.01
0.1
Dilution


300nm-1µm
1µm-2.5µm
2.5µm-4µm
4µm-10µm
Peak Concentration
2.0×10^4
1.5×10^4
1.0×10^4
5.0×10^3
2.5×10^3
0.0
0.0001
0.001
0.01
0.1
Dilution

Supplementary Figure 17. Particle Distribution of Varying Dilution Ratios, Related to Figure 3. In order to characterize the accuracy of the PM Sensor coupled with the Dilution Robot over a wide range of dilutions, data was collected at the following dilutions: 0.1, 0.0316, 0.01, 0.00316, 0.001, and 0.000316. Due to the wide range of dilutions, the data is plotted in a Logarithmic axis (both X and Y axis). The data is for the four particle distribution sizes, 300 nm – 1.0 μm, 1.0 μm – 2.5 μm, 2.5 μm – 4.0 μm, and 4.0 μm – 10.0 μm. A general linear trend is observed for all particle distributions validating the dilution procedure. Both the total particles counted per cm^3 over each inter-puff interval and peak concentration over each inter-puff interval versus dilution are plotted.

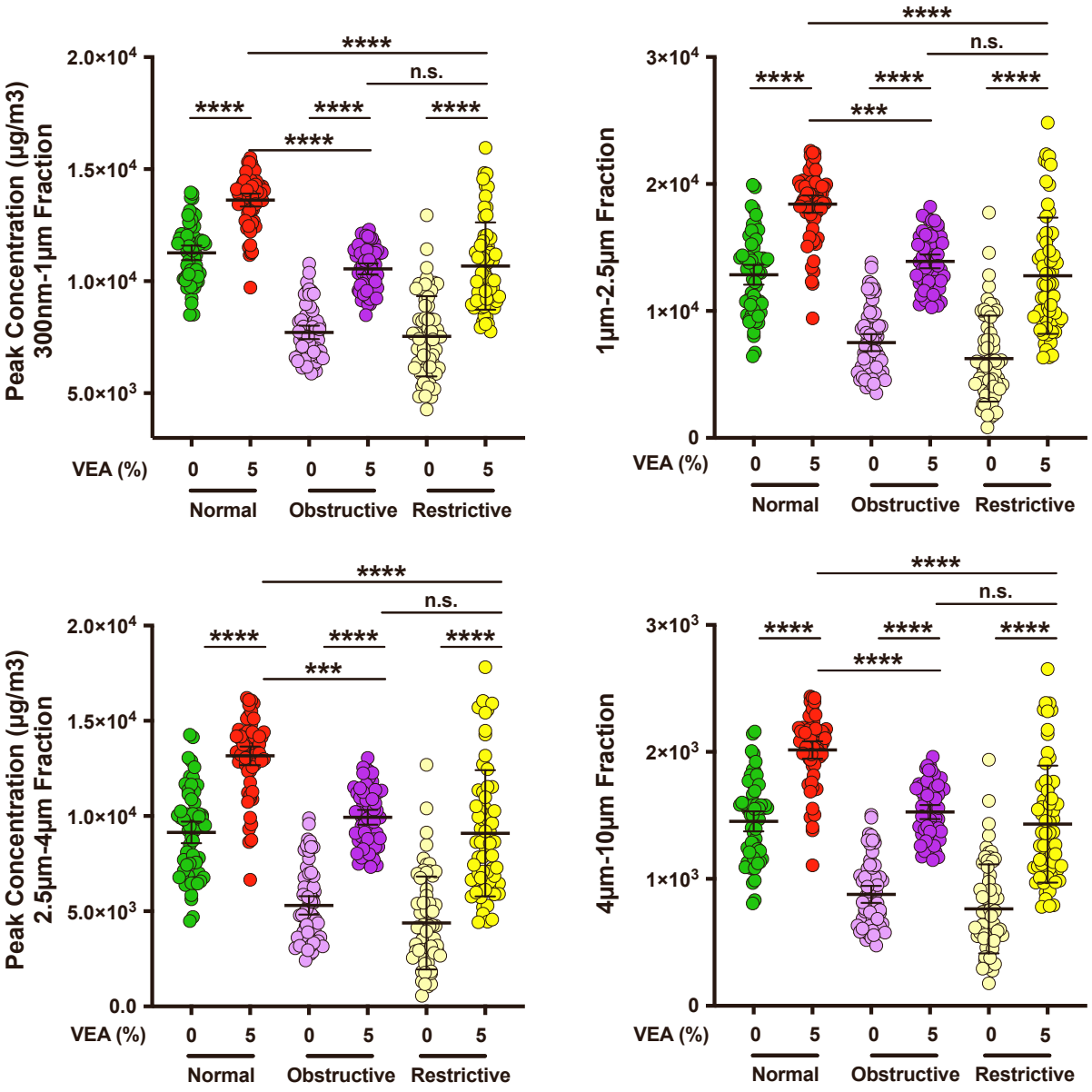

Peak Concentration (µg/m3)
300nm-1µm Fraction
1.6×10^4
1.2×10^4
8.0×10^3
4.0×10^3
n.s.
n.s.
**
****
1µm-2.5µm Fraction
2.5×10^4
2.0×10^4
1.5×10^4
1.0×10^4
5.0×10^3
0.0
n.s.
n.s.
*
****
Peak Concentration (µg/m3)
2.5µm-4µm Fraction
2.0×10^4
1.5×10^4
1.0×10^4
5.0×10^3
0.0
n.s.
n.s.
*
****
4µm-10µm Fraction
3×10^3
2×10^3
1×10^3
0
n.s.
n.s.
*
****
VEA (5%)
Nicotine (%)
0
0.6
1.2
2.4

Supplementary Figure 18. Impact of Nicotine Alone or In Combination with Vitamin E Acetate on Particle Peak Concentrations Generated from Electronic Cigarette, Related to Figure 4. Plotted Distributions peak concentration over each inter-puff interval for particles within 300 nm – 1 μm, 1 μm – 2.5 μm, 2.5 μm – 4 μm, and 4 μm – 10 μm in size for the different VEA and nicotine concentrations (0% or 5% (v/v) VEA with 0%, 0.6%, 1.2%, or 2.4% (wt/v) nicotine all prepared in 50/50 PG/VG). Kruskal-Wallis test demonstrated statistical significance ($p < 0.0001$). Post-hoc analysis (Dunn's multiple comparison test) demonstrated statistically significant differences between 0% VEA and 5% VEA at both 0% and 0.6% nicotine for all particle size distributions. Bars show mean and 95% confidence intervals for each data set.


Total Particles per cm3
6×10^7
5×10^7
4×10^7
2.5×10^7
0.0
4-10 µm
2.5-4 µm
1-2.5 µm
0.3-1 µm
0.2%
1.6%
12.5%
85.7%
0.2%
1.5%
12.3%
85.9%
0.2%
1.4%
11.6%
86.8%
0.2%
1.4%
11.5%
86.9%
0.3%
1.8%
14.6%
83.3%
0.3%
1.7%
13.4%
84.6%
0.2%
1.6%
12.9%
85.3%
0.2%
1.5%
11.8%
86.5%
VEA (5%)
Nicotine (%)
0
0.6
1.2
2.4

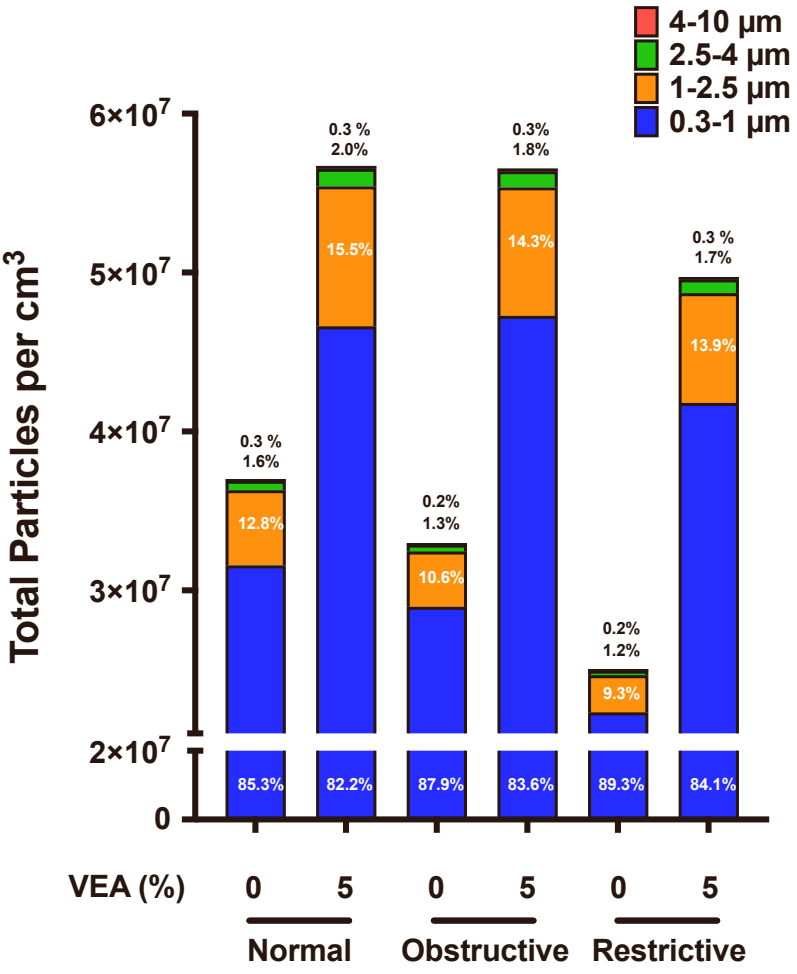
Supplementary Figure 19. Impact of Nicotine Alone or In Combination with Vitamin E Acetate Content on Cumulative Total Sub-micron and Microparticles Generated from Electronic Cigarette Over a Representative Vaping Regiment, Related to Figure 4. Total combined particles per cm^3 were counted at a sampling rate of 1.3 seconds over the course of seven independent vaping sessions (at 9 puffs per session) in the presence of a range of nicotine doses (0%–2.4% wt/v in 50/50 PG/VG) with (5% v/v) or without VEA. In addition, for each condition, the particle distribution size profile was segmented to display relative measured abundance of each size fraction as a percentage of total particles that were counted. The 0% VEA and 0% nicotine combination yielded 3.26×10^7 particles/cm^3 within 300 nm – 1 µm (85.7% of total), 4.77×10^6 particles/cm^3 within 1 µm – 2.5 µm (12.5% of total), 5.96×10^5 particles/cm^3 within 2.5 µm – 4 µm (1.6% of total), and 9.51×10^4 particles/cm^3 within 4 µm – 10 µm (0.2% of total). The 0% VEA and 0.6% nicotine combination yielded 2.95×10^7 particles/cm^3 within 300 nm – 1 µm (85.9% of total), 4.23×10^6 particles/cm^3 within 1 µm – 2.5 µm (12.3% of total), 5.28×10^5 particles/cm^3 within 2.5 µm – 4 µm (1.5% of total), and 8.42×10^4 particles/cm^3 within 4 µm – 10 µm (0.2% of total). The 0% VEA and 1.2% nicotine combination yielded 2.73×10^7 particles/cm^3 within 300 nm – 1 µm (86.8% of total), 3.64×10^6 particles/cm^3 within 1 µm – 2.5 µm (11.6% of total), 4.54×10^5 particles/cm^3 within 2.5 µm – 4 µm (1.4% of total), and 7.25×10^4 particles/cm^3 within 4 µm – 10 µm (0.2% of total). The 0% VEA and 2.4% nicotine combination yielded 2.70×10^7 particles/cm^3 within 300 nm – 1 µm (86.9% of total), 3.56×10^6 particles/cm^3 within 1 µm – 2.5 µm (11.5% of total), 4.44×10^5 particles/cm^3 within 2.5 µm – 4 µm (1.4% of total), and 7.10×10^4 particles/cm^3 within 4 µm – 10 µm (0.2% of total). The 5% VEA and 0% nicotine

combination yielded 4.59×10^7 particles/cm^3 within 300 nm – 1 µm (83.3% of total), 8.03×10^6 particles/cm^3 within 1 µm – 2.5 µm (14.6% of total), 1.01×10^6 particles/cm^3 within 2.5 µm – 4 µm (1.8% of total), and 1.60×10^5 particles/cm^3 within 4 µm – 10 µm (0.3% of total). The 5% VEA and 0.3% nicotine combination yielded 3.924×10^7 particles/cm^3 within 300 nm – 1 µm (84.6% of total), 6.22×10^6 particles/cm^3 within 1 µm – 2.5 µm (13.4% of total), 7.78×10^5 particles/cm^3 within 2.5 µm – 4 µm (1.7% of total), and 1.24×10^5 particles/cm^3 within 4 µm – 10 µm (0.3% of total). The 5% VEA and 1.2% nicotine combination yielded 3.27×10^7 particles/cm^3 within 300 nm – 1 µm (85.3% of total), 4.94×10^6 particles/cm^3 within 1 µm – 2.5 µm (12.9% of total), 6.18×10^5 particles/cm^3 within 2.5 µm – 4 µm (1.6% of total), and 9.84×10^4 particles/cm^3 within 4 µm – 10 µm (0.2% of total). The 5% (v/v) VEA and 2.4% nicotine combination yielded 2.85×10^7 particles/cm^3 within 300 nm – 1 µm (86.5% of total), 3.89×10^6 particles/cm^3 within 1 µm – 2.5 µm (11.8% of total), 4.85×10^5 particles/cm^3 within 2.5 µm – 4 µm (1.5% of total), and 7.75×10^4 particles/cm^3 within 4 µm – 10 µm (0.2% of total).


Peak Concentration (µg/m3)
300nm-1µm Fraction
2.0×10^4
1.5×10^4
1.0×10^4
5.0×10^3
****
n.s.
****
****
****
****
VEA (%)
0
5
0
5
0
5
Normal
Obstructive
Restrictive
1µm-2.5µm Fraction
3×10^4
2×10^4
1×10^4
0
****
n.s.
****
****
****
***
Peak Concentration (µg/m3)
2.5µm-4µm Fraction
2.0×10^4
1.5×10^4
1.0×10^4
5.0×10^3
0.0
****
n.s.
****
****
****
***
4µm-10µm Fraction
3×10^3
2×10^3
1×10^3
0
****
n.s.
****
****
****
****

Supplementary Figure 20. Impact of Healthy versus Diseased Breathing on Cumulative Total Sub-micron and Microparticles Generated from Electronic Cigarette Over a Representative Vaping Regiment, Related to Figure 5. Total combined particles per cm^3 were counted at a sampling rate of 1.3 seconds over the course of seven independent vaping sessions (at 9 puffs per session) in the absence or presence of VEA (5% v/v in 50/50 PG/VG) for healthy, restrictive and obstructive pulmonary breathing profiles. For each condition, the particle distribution size profile was segmented to display relative measured abundance of each size fraction as a percentage of total particles that were counted. The 0% VEA at normal breathing yielded 3.16×10^7 particles/cm^3 within 300 nm – 1 µm (85.3% of total), 4.73×10^6 particles/cm^3 within 1 µm – 2.5 µm (12.8% of total), 5.91×10^5 particles/cm^3 within 2.5 µm – 4 µm (1.6% of total), and 9.43×10^4 particles/cm^3 within 4 µm – 10 µm (0.3% of total). The 5% VEA at normal breathing yielded 4.66×10^7 particles/cm^3 within 300 nm – 1 µm (82.2% of total), 8.81×10^6 particles/cm^3 within 1 µm – 2.5 µm (15.5% of total), 1.11×10^6 particles/cm^3 within 2.5 µm – 4 µm (2.0% of total), and 1.76×10^5 particles/cm^3 within 4 µm – 10 µm (0.3% of total). The 0% VEA at obstructive breathing yielded 2.90×10^7 particles/cm^3 within 300 nm – 1 µm (87.9% of total), 3.48×10^6 particles/cm^3 within 1 µm – 2.5 µm (10.6% of total), 4.33×10^5 particles/cm^3 within 2.5 µm – 4 µm (1.3% of total), and 6.93×10^4 particles/cm^3 within 4 µm – 10 µm (0.2% of total). The 5% VEA at obstructive breathing yielded 4.73×10^7 particles/cm^3 within 300 nm – 1 µm (83.6% of total), 8.09×10^6 particles/cm^3 within 1 µm – 2.5 µm (14.3% of total), 1.01×10^6 particles/cm^3 within 2.5 µm – 4 µm (1.8% of total), and 1.61×10^5 particles/cm^3 within 4 µm – 10 µm (0.3% of total). The 0% VEA at restrictive breathing

yielded 2.23×10^7 particles/cm^3 within 300 nm – 1 μm (89.3% of total), 2.33×10^6 particles/cm^3 within 1 μm – 2.5 μm (9.3% of total), 2.89×10^5 particles/cm^3 within 2.5 μm – 4 μm (1.2% of total), and 4.63×10^4 particles/cm^3 within 4 μm – 10 μm (0.2% of total). The 5% VEA at restrictive breathing yielded 4.18×10^7 particles/cm^3 within 300 nm – 1 μm (84.1% of total), 6.91×10^6 particles/cm^3 within 1 μm – 2.5 μm (13.9% of total), 8.65×10^5 particles/cm^3 within 2.5 μm – 4 μm (1.7% of total), and 1.38×10^5 particles/cm^3 within 4 μm – 10 μm (0.3% of total). Kruskal-Wallis test demonstrated statistical significance ($p < 0.0001$). Post-hoc analysis (Dunn's multiple comparison test) showed statistically significant differences between all conditions except normal 0% VEA versus both obstructive and restrictive 5% VEA, and obstructive versus restrictive at both 0% and 5% VEA. Error bars indicate mean and 95% confidence intervals for each data set.


4-10 µm
2.5-4 µm
1-2.5 µm
0.3-1 µm
Total Particles per cm3
6×10^7
5×10^7
4×10^7
3×10^7
2×10^7
0
0.3 %
1.6%
12.8%
85.3%
0.3 %
2.0%
15.5%
82.2%
0.2%
1.3%
10.6%
87.9%
0.3%
1.8%
14.3%
83.6%
0.2%
1.2%
9.3%
89.3%
0.3 %
1.7%
13.9%
84.1%
VEA (%)
0
5
0
5
0
5
Normal
Obstructive
Restrictive

Supplementary Figure 21. Influence of Healthy versus Diseased Breathing in the Absence or Presence of Vitamin E Acetate on Inhaled Particle Peak Concentrations, Related to Figure 5. Plotted Distributions of the peak concentration over each inter-puff interval for particles within 300 nm – 1 μm, 1 μm – 2.5 μm, 2.5 μm – 4 μm, and 4 μm – 10 μm in size for the different VEA concentrations (0% or 5% (v/v) VEA (prepared in 50/50 PG/VG) with normal, obstructive, and restrictive breathing profiles).