



Hardware Article

RespiCo: A novel, flexible, and stand-alone electronic respiratory coaching device

Kanchan Kulkarni^a, John H. Nichols^{b,d}, Antonis A. Armoundas^{a,d}, Jesse D. Roberts Jr.^{a,b,c,d,*}^a Cardiovascular Research Center, Massachusetts General Hospital, Boston, MA, United States^b Department of Anesthesia, Critical Care and Pain Medicine, Massachusetts General Hospital, Boston, MA, United States^c Department of Pediatrics, Massachusetts General Hospital, Boston, MA, United States^d Harvard Medical School, Harvard University, Cambridge, MA, United States

ARTICLE INFO

Article history:

Received 28 March 2022

Received in revised form 5 July 2022

Accepted 6 July 2022

Keywords:

Respiratory muscle training

Mindfulness breathing

Athletic conditioning

And respiratory rate control

ABSTRACT

Conscious respiratory pattern and rate control is desired by patients with some forms of pulmonary disease that are undergoing respiratory muscle conditioning and rehabilitation, by practitioners of meditation hoping to improve mindfulness and wellbeing, by athletes striving to obtain breathing control in order to increase competitiveness, and by engineers and scientists that wish to use the data from breathing subjects to test hypotheses and develop physiological monitoring systems. Although prerecorded audio sources and computer applications are available that guide breathing exercises, they often suffer from being inflexible and allow only limited customization of the breathing cues. Here we describe a small, lightweight, battery-powered, microprocessor-based respiratory coaching device (RespiCo), which through wireless or wired connections, can be easily customized to precisely guide subjects to breathe at desired respiratory rates using specific breathing patterns through visual, auditory, or haptic cues. Digital signals can also be captured from the device to document the breathing cues provided by the device for research purposes. It is anticipated that this device will have important utility for those who wish to be guided to breathe in a precise manner or in research and development of physiologic monitoring systems.

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

Specifications table

Hardware Name	RespiCo (<u>R</u> espiratory <u>C</u> oach)
Subject Area	Medical
Hardware Type	Other: Physical computing via an embedded system
Open Source License	MIT
Cost of Hardware	\$54.14, prototype version
Source File Repository	OSF

* Corresponding author at: Cardiovascular Research Center, Massachusetts General Hospital, Boston, MA, United States.

E-mail address: roberts@cvrc.mgh.harvard.edu (J.D. Roberts Jr.).

Hardware in context

Breathing using specific respiratory rates and patterns has long been thought to produce health benefits. Metered breathing during respiratory muscle training and rehabilitation has been determined to improve pulmonary function in patients with some forms of lung disease. For example, breathing at low respiratory rates (RRs) with prolonged exhalation periods while simultaneously concentrating on diaphragm muscle use (*diaphragmatic breathing*), or while exhaling against mild upper airway resistance caused by pursed lips (*pursed lip breathing*), can improve respiratory muscle function in patients with parenchymal [1,2] and airway [3] diseases. Consciously breathing at low RRs can also reduce dyspnea and improve pulmonary gas exchange and exercise tolerance in patients with congestive heart failure [4]. In addition to helping people with known pulmonary disease, controlled breathing has additional physiologic performance and wellness benefits. For example, slow deep breathing has been observed to improve oxygen uptake and reduce pulmonary artery pressure in subjects acclimating to high altitude [5]. For many years [6], breath-control exercises have been employed during mindfulness breathing as a means to increase self-awareness, decrease systemic pain [7], and achieve a calm state by Yoga and Zen practitioners. Moreover, there is emerging interest by athletes and their coaches in controlling respiratory efforts during physical activities as a way to enhance physical performance [8–10]. Finally, controlled breathing can aid scientific investigation. Scientists have guided breathing in subjects that are undergoing a wide variety of physiological and psychological experiments (e.g. [11,12]). This includes subjects that are being instructed to breathe using different patterns and over a wide range of RRs to gather data for developing and testing new physiologic monitoring systems. This is particularly true if the monitoring system is to be employed with patients that exhibit wide alterations in RRs, such as those with Cheyne-Stokes breathing and abnormalities in the respiratory central pattern generator (CPG; e.g. Parkinson's disease, Rett syndrome, and Alzheimer's disease [13,14]).

Currently, very few devices are available to precisely direct subjects to breathe using specific patterns and rates, and the ones that exist offer very limited flexibility, portability, and ease of use. There are a variety of on-line sources that provide information about how to perform respiratory muscle conditioning and rehabilitation (e.g. [15,16]). These guides are intended to supplement and reinforce instructions that are initiated by health care providers. But an important limitation of such breathing instruction is that it requires the patients to employ mental counting schema ("counts") to meter the breathing pattern, and chronoception studies [17–19] suggest that this approach is unlikely to be able regulate the subject's RR in a consistent manner. Moreover, this form of breathing control can be challenging to some patients. This is because at the same time they need to perform counting to control the respiratory pattern, the patients need to also be mindful of thoracic wall and abdominal muscle movements and lip pursing. This can lead to cognitive overload, thereby causing inconsistent exercising and decreased compliance with the prescribed exercises. Audio recordings, applications for smart devices and computers [20], and a few discrete devices are available to be used to coach breathing. But few of these tools permit the user to have complete and flexible control of the breathing patterns, intervals, cycles, and session lengths that are sometimes desired. One device (e.g. RESPeRATE) uses RR measurements that are detected by a sensor belt that encircles the upper abdomen to control melodies that are played through headphones to guide the user to breathe at ~ 10 breaths per minute (BPM). Studies suggest that this type of breathing feedback can improve blood pressure, with a recent critical review highlighting the need for additional studies to support that conclusion [21]. In the past, scientists and engineers have used audio recordings and metronomes to guide breathing rates of subjects during experiments that collected physiologic data (e.g. [12,22,23]). However, we are not aware of a system that can be used to instruct a wide range of relevant respiratory patterns and rates, including the ability to precisely direct the duration of inhalation or exhalation, or that provides a digital output signal that can be recorded to confirm the breathing cues that was provided. Importantly, a major shortfall of the methods and devices described above is that they do not provide a variety of sensory output cues. They appear to only provide breathing cues from one or two outputs. It is highly desirable to have the options of visual, auditory, and haptic outputs so that users with sensory disabilities can benefit from the respiratory pattern and rate control or can participate in the experimental work.

In this report, we present, characterize, and demonstrate a novel electronic respiratory coaching system, *RespiCo*, that is small, inexpensive, easily constructed and modified, and that enables a subject to receive precise respiratory cues, over a wide range of breathing patterns and rates, using auditory, visual, and haptic outputs. The system offers the user choices between commonly used respiratory muscle training, mindfulness breathing, athletic breathing conditioning patterns and rates, or the option to define fully customizable breathing patterns and RR sequences. Also, digital output from the device can be captured to document the breathing cues that it provided. Using a smart device or computer connected to *RespiCo* using a Wi-Fi or hard-wired serial connection, the user, practitioner, engineer, or scientist can select these device output features.

Hardware description

The technical objective of this project was to design a lightweight, stand-alone, battery or USB-powered, inexpensive electronic device that can be employed to guide human subjects to breathe using precise respiratory patterns and rates. To be flexible, the device features were made to be easily configurable, via Wi-Fi and/or wired connections, thereby providing extensive customization of the breathing cues. Importantly, the device was designed to provide breathing cues through a

wide range of sensory modes: auditory, visual, and haptic. To extend its use in an experimental setting, the device was also designed to provide outputs through serial and digital ports through which data could be collected to confirm the RespiCo output signals and performance. To our knowledge, no similar device exists today that provides these important features.

The schematic for RespiCo, highlighting its technical details, is provided in Fig. 1. The heart of the device is a multifunctional ESP32-WROOM-32 microprocessor (Espressif Systems). Although several microprocessors could be used for this application, the ESP32 one was chosen because it is engineered for mobile and wearable applications and consumes very low levels of power. Moreover, it has easily accessible hardware timers that can be used to regulate the timing of the RespiCo output cues. The ESP32 microprocessor also supports a high degree of connectivity; the RespiCo uses its 802.11b/g/n HT40 Wi-Fi transceivers and UARTs for communication with the user. This processor has been used in several control and monitoring devices that are detailed elsewhere in this journal (e.g. [24–26]).

In this project, we employed the Adafruit Huzzah32–ESP32 Feather Board to house the ESP32 microprocessor, which has the benefit of incorporating a battery- and USB-powered energy management system, a lithium ion polymer battery JST PH connector and charging system, and a micro-USB cable connector and serial USB-to-TTL signal converter system. In this development board, these systems and the ESP32 can be powered via a battery, when one is connected. The development board also has through-hole pads for soldering two single rows of interfacing pins. These pins permit the board to be inserted into a breadboard, or to have prototype boards attached on top. Breadboard and prototype board RespiCo forms are shown in the demonstration videos.

Several of the ESP32 general purpose input and output (GPIO) pins are brought out to through-hole connectors of the development board. Through regulating the voltages of these GPIO pins, custom software running on the ESP32 regulates the activity of the discrete output devices of RespiCo, namely, a LED, passive piezoelectric buzzer, and haptic motor. Moreover, the ESP32 enable pin is connected through a switch that allows the user to shut down the processor between uses. A diffused common anode RGB LED controlled by a set of GPIOs enables the visual cue output, which is programmed to emit

RespiCo

Kulkarni et al.

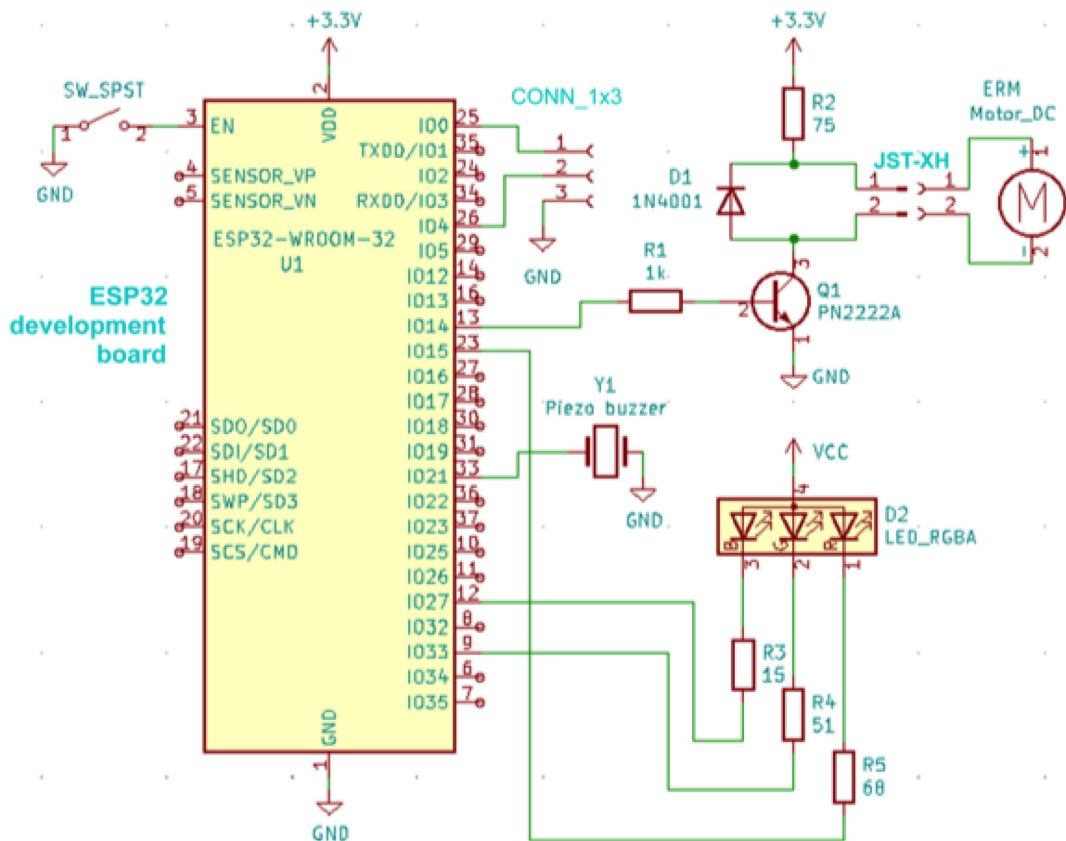


Fig. 1. Schematic drawing of device. A wiring diagram of the device was made using KiCad, an open source Electronic Design Automation software package. The ESP32 general purpose input and output pins (IO, in green) and their mapping to a Huzzah32 – ESP32 Feather Board pins (in red) are shown. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

different light colors to signal inhalation, exhalation, and breath holding to the user. A passive piezoelectric buzzer that is controlled by one of the GPIO pins mediates the auditory output and generates different tones that signal inhalation, exhalation, and an impending change in breathing cue rate. A low-cost eccentric rotating mass (ERM) vibration motor provides haptic output to inform the breathing patterns of the user. Lastly, headers that are connected to GPIO pins are also supplied that output TTL digital signals corresponding to the inhalation and exhalation breathing cycles made by the device. This will allow advanced users to capture detailed and time-stamped device output data for research and development purposes. Additional data obtained during breathing instruction is provided and automatically saved through a Wi-Fi or hard-wire connected smartphone or computer as ASCII text. Advanced users can adjust software flags in a [customizations.py](#) file to regulate the details about the device sensory and ASCII output as desired.

RespiCo is designed for, but not limited to the following applications:

- The system can be used for coaching patients, meditators, and athletes to breathe using specific respiratory patterns and rates. The small size, lightweight, and inexpensive nature of the device means that few barriers can exist to its use.
- Researchers can employ the system to obtain precise breathing data from subjects while testing the efficacy of specific respiratory pattern and rate exercises to improve the respiratory muscle function, mindfulness, and athletic outcomes. In this regard, the effects of respiratory rate control could be determined using physiologic or other biofeedback monitoring systems.
- Engineers and scientists can use RespiCo to guide subjects to breathe in various respiratory patterns over a wide range of RRs to perform relevant experiments for research and development of advanced physiological monitoring systems.

Design files

Design file name	File type	Open Source License	Location of file
Schematic.pdf	pdf	Public domain	Available with the article and at: https://osf.io/u2gfc
RespiCo_prototype.pdf	pdf	Public domain	Available at: https://osf.io/u2gfc
RespiCo_breadboard_boot.py	MicroPython script	MIT	
RespiCo_breadboard_config.py	MicroPython script	MIT	
RespiCo_protoboard_boot.py	MicroPython script	MIT	
RespiCo_protoboard_config.py	MicroPython script	MIT	
RespiCo_customizations.py	MicroPython script	MIT	
RespiCo.mpy	MicroPython bytecode		
RespiCo_patternData.csv	csv	MIT	
RespiCo_rrData.csv	csv	MIT	
Supplemental_info.pdf	pdf	MIT	
Demonstration videos	mp4	MIT	
Fig. 4 Fig. 5_gray_scale.pdf	pdf	MIT	
RespiCo_pattern_phase_timings.xlsx	xlsx	MIT	
RespiCo_pattern_phase_time_data.pdf	pdf	MIT	

- Schematic.pdf (also, Fig. 1) shows the RespiCo components and interconnections.
- RespiCo_prototype.pdf provides information about how the device might be soldered together onto a prototype board, which then mates with the development board.
- RespiCo_protoboard_boot.py and RespiCo_breadboard_boot.py are MicroPython scripts that when installed as “boot.py” on their target builds initialize GPIO ports and set up the RespiCo as a Wi-Fi access point.
- RespiCo_breadboard_config.py and RespiCo_protoboard_config.py are MicroPython scripts that map the ESP32 development board’s GPIO pins to the discrete output devices on the breadboard and prototype board hardware versions of RespiCo, respectively. The user can change these mappings if they use another board or want to rearrange the devices on a custom circuit board. The RespiCo_breadboard_config.py file maps the GPIO pins to the devices as shown in the schematic and breadboard version of RespiCo; the RespiCo_protoboard_config.py file defines the mappings of the pins and devices in the prototype version described in the RespiCo_prototype.pdf file.

- RespiCo_customizations.py is a MicroPython script that contains the variables that the user is most likely to customize. These customizations are described below.
- RespiCo.mpy is a MicroPython bytecode for the main script that serves the user interface and controls the RespiCo output devices.
- RespiCo_patternData.csv and RespiCo_rrData.csv are files containing the data shown in Fig. 4 and Fig. 5, respectively.
- Supplemental information.pdf is provided to help the user become familiarized with the ESP32, the development board, and MicroPython.

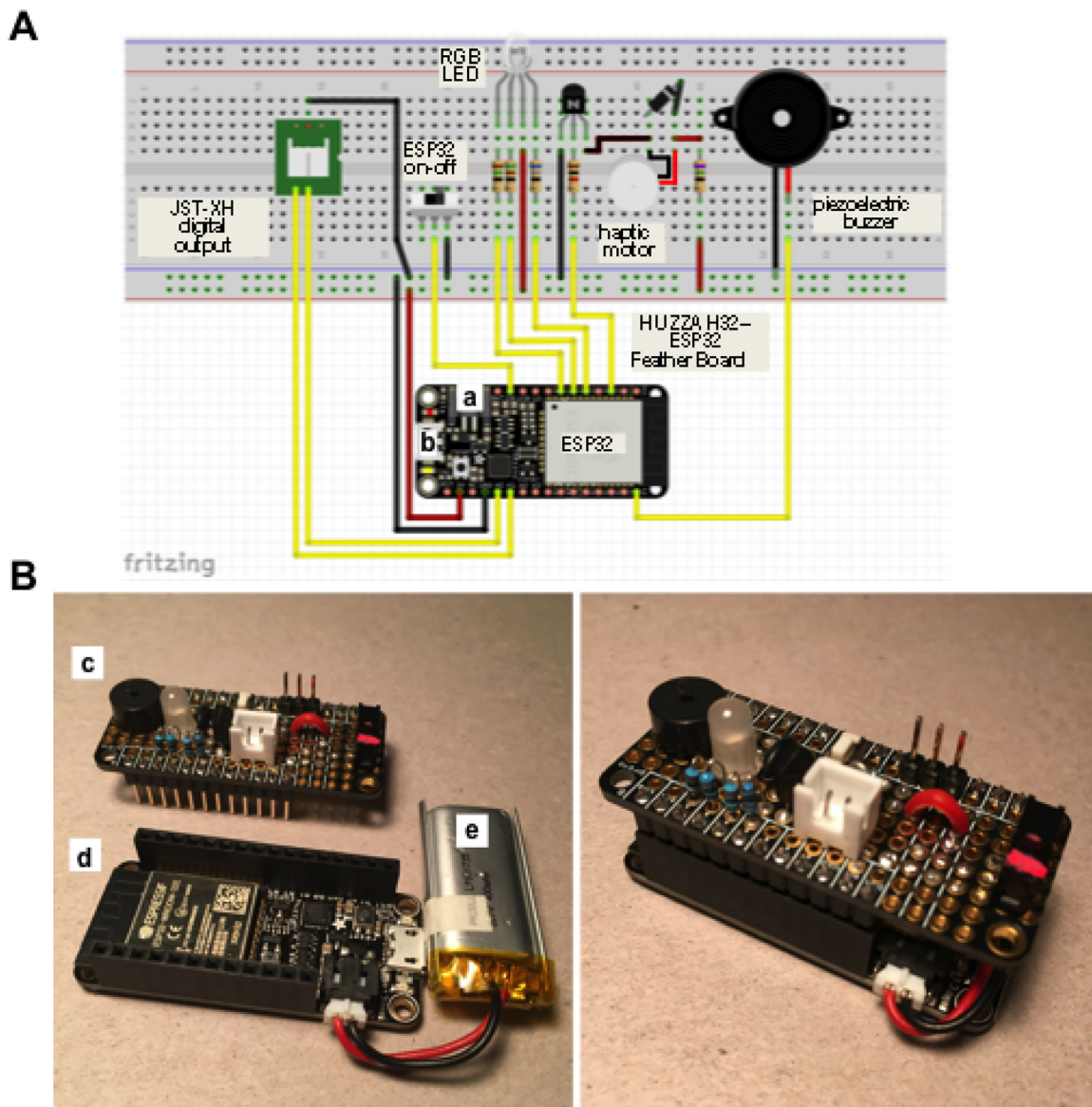


Fig. 2. RespiCo. **A.** RespiCo breadboard – showing interconnections between the HUZAZH32–ESP32 Feather Board and the discrete electrical components. The figure was made using Fritzing (v 0.9.4), an open-source hardware initiative, and annotated. Red and black represent power leads; yellow represents signal leads; the JST PH connector for a lithium ion polymer battery power connection (a), and a mini USB connector provides power and a serial connection (b), if desired. **B.** RespiCo prototype: Left –the top board containing the discrete electronic components and connectors (c), HUZAZH32–ESP32 Feather Board (d), and connected battery (e); Right – the assembled RespiCo prototype. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

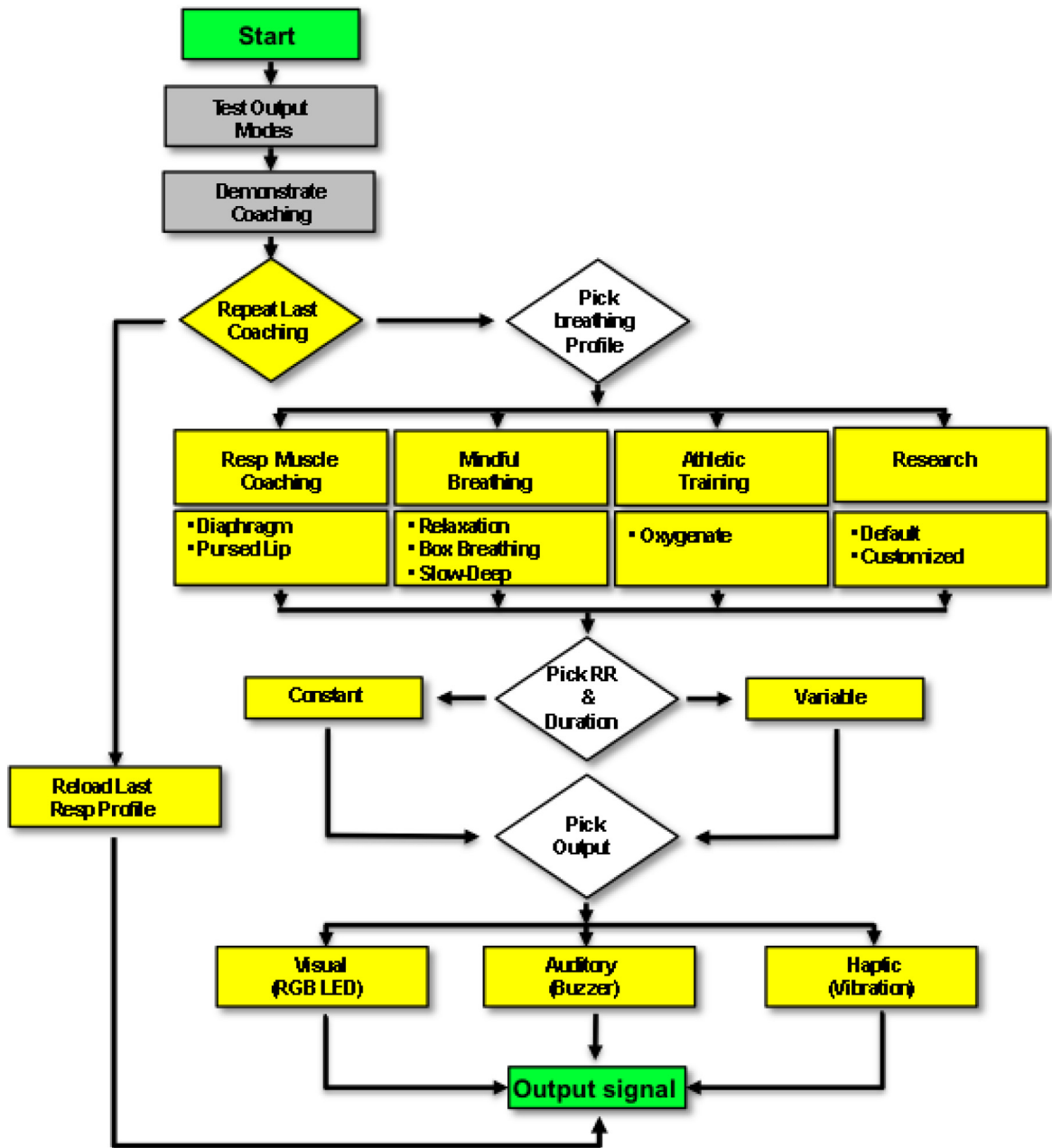


Fig. 3. Flow diagram of the user interface. After a Wi-Fi or wired serial connection with the device is established and the RespiCo script is started, the user is provided with options to test the outputs and observe a demonstration of the coaching that can be provided. Afterwards, the subject is given the option to reuse the most recent coaching parameters, or to enter new custom parameters, followed by the desired coaching session. Subsequently, the subject can repeat the session, enter new parameters and start a new session, or stop the coaching entirely. The testing and demonstration options can be removed from the user interface, and the coaching outputs can be streamlined, if desired, by simple changes to flags in the *RespiCo_customizations.py* file.

- Three demonstration videos are supplied. One video shows the RespiCo constructed on a breadboard and, through Wi-Fi control, demonstrating the options to test the outputs and to demonstrate the device outputs. Another video is provided to show how a Wi-Fi interface can be used for the user to select, initiate, and then receive the device output signals. Also, a demonstration video is provided showing the assembly of a prototype board version of the RespiCo, with the output devices soldered to a circuit board, and the use of Wi-Fi to select and start breathing training sessions.
- [Fig. 4_Fig. 5_gray_scale.pdf](#) is a gray scale version of [Figs. 4 and 5](#).
- [RespiCo_pattern_phase_timings.xlsx](#) provides the algorithmically derived inhalation, inhalation pause, exhalation, and exhalation pause durations at 12–30 BPM.
- [RespiCo_pattern_phase_time_data.pdf](#) provides the data shown in [Fig. 4](#) in a tabular form.

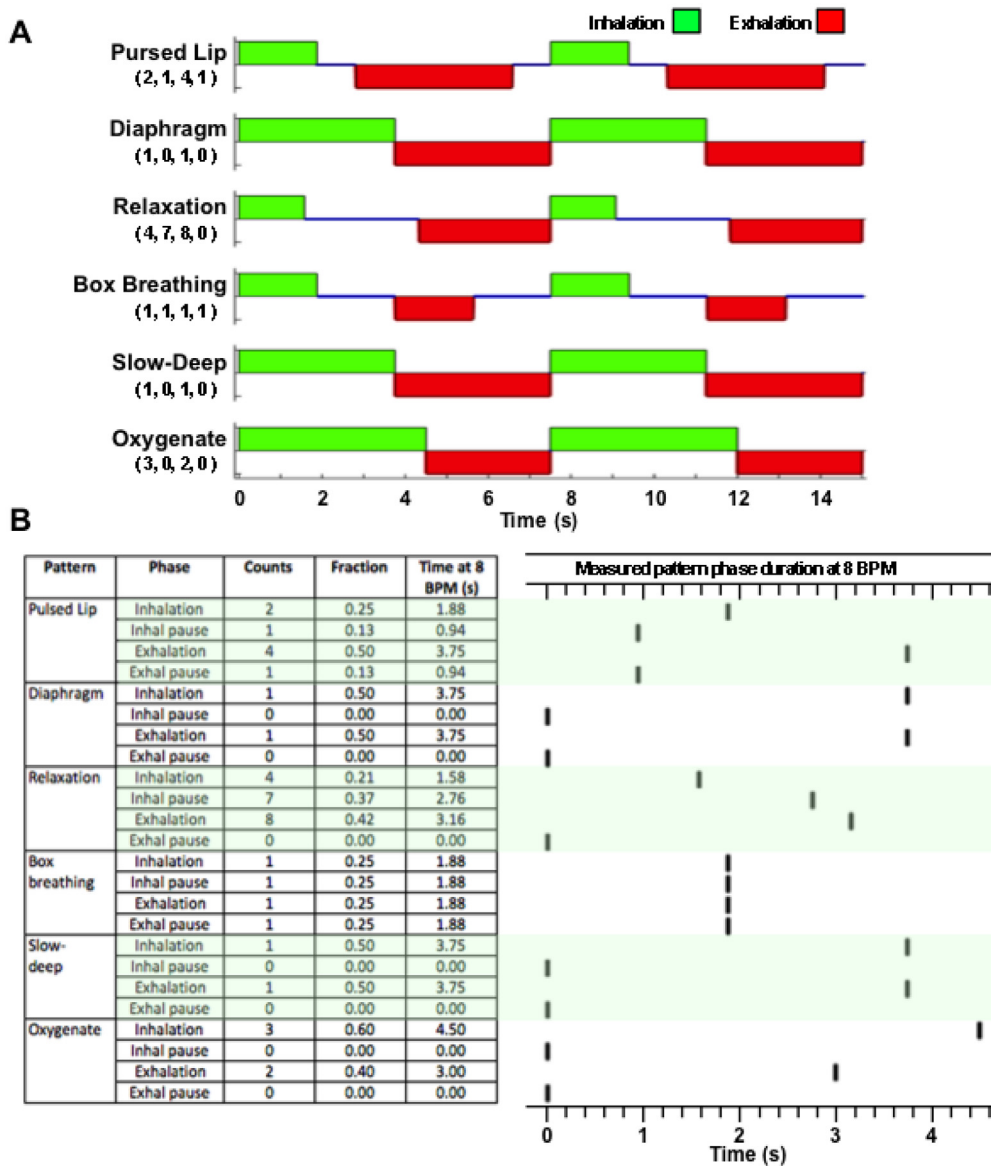


Fig. 4. Respiratory patterns produced by RespiCo. A. RespiCo was programmed to produce breathing phase cues for the indicated patterns at a rate of 8 BPM and the inhalation and exhalation signal outputs were measured at its digital ports. Respiration patterns corresponding to two breath cycles are shown; inhalation and exhalation output signals are color-coded as indicated, and the relative proportion of breath cycle (counts) for the inhalation, inhalation pause, exhalation, and exhalation pause phases are indicated within the parentheses. B. (Left panel) Table summarizes the pattern phase timings, indicating the pattern phase duration in counts and breath cycle fractions, and the calculated timings at 8 BPM. (Right panel) Horizontal boxplot depicts the duration of the RespiCo digital output signals for each pattern phase, referenced in the adjacent table, that was measured with a logic analyzer ($N = 5$ breathing cycles per pattern phase).

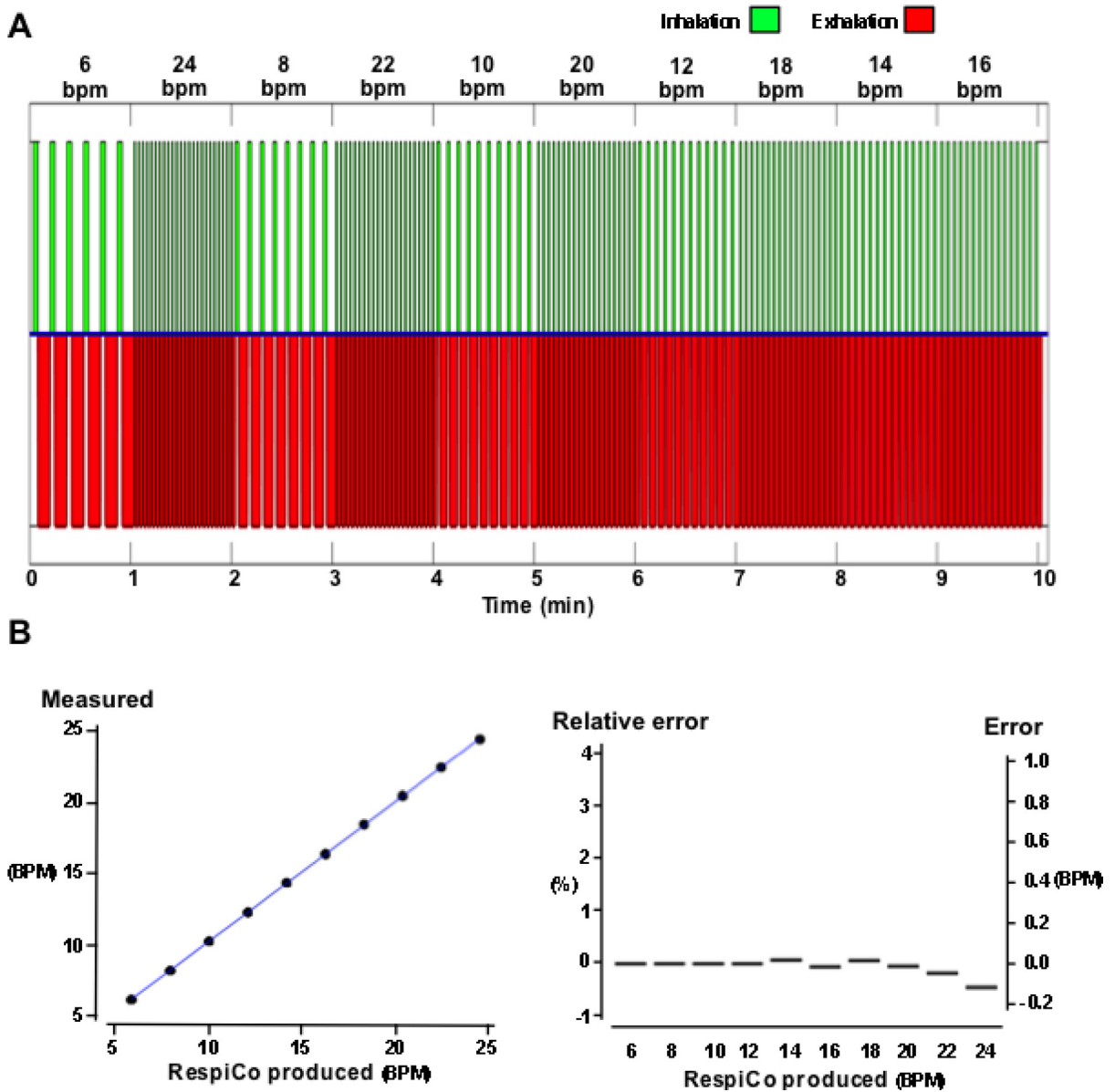


Fig. 5. Validation of respiratory rates generated by RespiCo. **A.** Illustration of RespiCo breathing cues in a physiologically relevant range of respiratory rates (RRs). The RespiCo was instructed to produce respiratory cycle cues for one minute at each of the indicated RRs. The outputs were measured at the RespiCo inhalation and exhalation digital ports; the inhalation and exhalation signals are color-coded as described above. To aid in the figure representation, a gray-scale version of it is in the *Supplemental materials*. **B.** RespiCo RR cue performance. Signals were obtained from the RespiCo inhalation cue digital port using an external pulse analyzer while the device was programmed to produce signals for the indicated rates over 30 min, and the RRs were derived. Left: linear regression of the RespiCo-produced and external device-measured RR data; right: boxplots of the error between the produced and measured RRs with and without normalization to the RespiCo-produced cue rates, as indicated. The variance of the data was too small for the individual data points and boxplot features to be distinguished.

Bill of materials

Designator	Component	#	Cost/unit USD	Total cost	Source of materials
<i>RespiCo breadboard -</i>					
U1	HUZZAH32 – Esp32- WROOM-32 Feather Board	1	\$19.95	\$19.95	Adafruit 3405
Female headers	Feather female header set	1	\$0.95	\$0.95	Adafruit 2886
Breadboard	Breadboard	1	\$5.95	\$5.95	Adafruit 239
Jumper wires	Male/male jumper wire set	1	\$3.95	\$3.95	Adafruit 759
USB cable	USB A-plug to Micro-USB data cable	1	\$4.95	\$4.95	Adafruit 2185
Wire	22 AWG flexible 2 conductor wire	3'	\$0.60/ft	\$1.80	Tuofeng 12AWG Amazon ASIN B0981NF1HV
SW_SPST	SPST slide switch	1	\$0.95	\$0.95	Adafruit 805
CONN_1x3	1 × 3 male header pins	3	\$4.95 / 360	\$0.04	Adafruit 392
R1	1 kOhm, 1/8 W, +/- 5 % value	1	\$0.01	\$0.01	Ltvystore LYSB01MDJQ1II- CMPTRACCS Amazon ASIN B01MDJQ1II
R2	75 Ohm, 1/8 W, +/- 5 % value	1	\$0.01	\$0.01	
R3	15 Ohm, 1/8 W, +/- 5 % value	1	\$0.01	\$0.01	
R4	51 Ohm, 1/8 W, +/- 5 % value	1	\$0.01	\$0.01	
R5	68 Ohm, 1/8 W, +/- 5 % value	1	\$0.01	\$0.01	
JST-XH	Male and female connectors	2	\$0.03	\$0.06	IZOKEE IZKEGB11-US Amazon ASIN B0791DJMW9
Y1	Passive piezo-electric buzzer	1	\$1.06	\$1.06	Uxcell a15091400ux0103 Amazon ASIN B01811WBNQ
D2	Diffuse output RGB common anode LED	1	\$0.50	\$0.50	FEDY FD-5WSRGB-A Adafruit 302
ERM Motor_DC	Brushless, eccentric rotating mass motor	1	\$1.40	\$1.40	uxcell A14061100ux0057 Amazon ASN B00PZYMCT8
D1	1 N4001 flyback diode	1	\$0.15	\$0.15	Adafruit 755
Q1	PN2222A NPN bipolar transistor	1	\$0.20	\$0.20	Adafruit 756
Battery	Lithium-ion polymer battery, 400mAh	1	\$6.95	\$6.95	Adafruit 3898
<i>RespiCo prototype, additional components -</i>					
Custom circuit board	FeatherWing proto, includes loose male headers	1	\$4.95	\$4.95	Adafruit 2884
CONN_1x3	1 × 3 male header pins	3	\$4.95 / 360	\$0.04	Adafruit 392
SW_tactile	Tactile switch button	1	\$4.95 / 20	\$0.25	Adafruit 1489

Build instructions

The RespiCo build requires assembling the hardware and installing python interpreter and the controlling software. First, solder the female headers onto the HUZZAH32-ESP32 Feather Board. During the initial assembly and testing, RespiCo can be constructed with development board's GPIO pins connected to devices plugged into a breadboard (Fig. 2A). After one is satisfied with the build and function of the system, a prototype version of RespiCo can be made (Fig. 2B). To make the prototype, a 1 × 3 male header is substituted for the JST-XH digital output connector used in the breadboard build and the JST-XH connector is now used for the haptic motor connector. In addition, an optional momentary switch is added to the circuit. The discrete electronic components are then placed on a custom circuit board and then the male headers are soldered onto the board. After checking the soldered connections, the custom board can be snapped onto the development board via the headers while sandwiching a battery. More details about the prototype construction, including a soldering diagram, are given in RespiCo_prototype.pdf. The device can be controlled by user input through a Wi-Fi or USB serial connection, or can be fully stand-alone, and on start-up provide previously determined respiratory pattern and rate control signals.

Assembling the hardware

The connections between the control board and the peripheral components of the breadboard version of RespiCo are shown in Fig. 2A. An RGB LED is used to provide the visual breathing cues to the subject. The values for the resistors limiting the current through the LED are adjusted according to the manufacturer's device specification. They can also be adjusted to regulate the brightness of the output to a level that is desired by the subject. For the prototype, the visual output is built using a common anode RGB LED. However, should the user wish to customize the colors that are used to cue the breathing pattern, for example to avoid red and green signaling colors, a RGBW integrated light source (e.g. SK6812) could be used. They would also reduce the need for resistors and GPIO ports, although at the expense of an increase in power consumption.

A passive piezoelectric buzzer is used to provide auditory cues to the subject. The unit chosen was chosen such that it can be driven at the 3 V output of the ESP32 GPIO pin. Although many of these buzzers operate at a resonant frequency of $\sim 2,300$ Hz, the frequency of the pulse width modulated signal driving the buzzer in RespiCo, is programmed to differentially signal inhalation and exhalation, and when a change in RR cues is imminent.

An inexpensive and lightweight DC-powered ERM motor is used in RespiCo as a haptic device to signal inhalation and exhalation cues to the user. Since the motor's power requirement (100 mA at 3 V DC) exceeds the output power that the ESP32 GPIO pins can provide, it is powered using a general-purpose transistor operated at threshold by one of the ESP32 GPIO pins. To dissipate inductive flyback current when the motor is shut off, a snubbing diode is incorporated in the circuit. To facilitate connection of the haptic device to RespiCo, while being mindful of the polarity, flexible two-wire cable is soldered to the leads of the motor and the junctions protected using heat shrink-wrap. The other ends of the connecting wire are crimped individually in the pins of a male JST-XH connector; the pins are then inserted into the connector housing. This connector is inserted into a female JST-XH one in the circuit when the haptic output is desired.

Headers are also provided to two GPIO pins of the ESP32 and to ground, to permit the recording of the 3 V signals with an oscilloscope or chart recorder in synchrony with the outputs produced by the device. One of the pins signals inhalation, while the other indicates an exhalation cue. The utility of these outputs to document device function is shown in the demonstration videos.

If the builder wishes to assemble the prototype version of RespiCo, the components shown in Fig. 2A are soldered onto a custom circuit board, as shown in Fig. 2B according to the extensive instructions provided in RespiCo_prototype.pdf. Attaching the circuit board onto the HUZZAH32-ESP32 Feather Board, while sandwiching an appropriately sized battery between it, then assembles the RespiCo prototype. The assembly is shown in the RespiCo prototype demonstration video; the constructed device is shown in the right panel of Fig. 2B.

RespiCo can be operated using USB and/or battery power sources. A single pole single throw (SPST) switch is connected to the enable pin (EN) of the ESP32 microprocessor, which shuts down the device when it is connected to ground. The microprocessor can be reset using the momentary button on the HUZZAH32-ESP32 Feather Board or by power cycling the processor using the SPST switch.

Installing the firmware and controlling software

MicroPython interpreted scripts are used to control the RespiCo. MicroPython is an open-source [27] Python 3 compiler and runtime interpreter that implements a subset of the standard Python library. It is engineered to be memory efficient and is optimized to run on select microprocessors, including the ESP32. During the RespiCo device build, the MicroPython interpreter and RespiCo bytecode and supporting scripts need to be installed on the ESP32 via a USB serial connection with a host computer. For this, a serial connection is established with the ESP32 through the USB port of the HUZZAH32-ESP32 Feather Board, the MicroPython firmware from an on-line repository at [MicroPython.org](https://micropython.org) is installed onto the chip, and the scripts are loaded. It is critical to use a high-quality USB data cable and not a power-only one for this work. All the scripts and bytecode that control RespiCo are provided in the online source indicated in the Design Files table, and links to information that can aid in this aspect of the RespiCo construction are provided in *Supplemental_info.pdf*. In this implementation, we use MicroPython v1.17.

Testing the build

At this point, we assume that the user has been successful in installing the interpreter and scripts and is able to communicate with the ESP32 on RespiCo using the serial USB connection and the MicroPython interactive top level or language shell, a read-evaluate-print loop (REPL). To aid in testing the hardware build and the software, we have incorporated two options in the *respiCo.mpy* bytecode that can be used to assess the output functions of the device and to demonstrate its auditory and visual outputs. To launch the RespiCo software, enter `"import RespiCo"`, without the quote marks, at the REPL interface prompt (see the *Demonstration videos* for an example of how to do this). The user is then presented with options to test the outputs and demonstrate the output features features. If these testing options are no longer desired, then the *RespiCo_customizations.py* file can be replaced with another version in which flags are set to stop those options from being offered when the main RespiCo bytecode runs.

Testing the stand-alone mode

To control and operate RespiCo in a fully stand-alone mode, an REPL port needs to be enabled, that works through a Wi-Fi web socket connection (WebREPL). To enable this, during a serial REPL connection with the device enter “import webrepl_setup”, without the quote marks, at the REPL prompt and then return to start the WebREPL setup module. Then follow the prompts to enable WebREPL to be started when the device is booted and to select a WebREPL password. At this point, the ESP32 microprocessor needs to be restarted, and a battery supply or a simple USB power cable connected to RespiCo. Load the MicroPython WebREPL client onto the host smart device or computer. This client permits a Wi-Fi-enabled smart device or computer to access the WebREPL interface made by MicroPython running on the RespiCo via WebSockets. This can be accomplished by downloading the client and related tools from <https://github.com/micropython/webrepl> and running them locally, or access to the client can be achieved through the web by entering <https://MicroPython.org/webrepl/> into a browser on the smart device or computer. MicroPython currently does not permit https connections; so make sure that the browser that is being used to host the client does not default to that type of a connection. After the WebREPL client is activated, establish a Wi-Fi connection to the RespiCo device access point. It is setup by code in the *boot.py* files to be discoverable as “RespiCo”; the access point has the password: “password”. This can be changed if desired in the *WebREPL_cfg.py* file that is created using the *webrepl_setup* module. Next, establish a socket connection to the RespiCo’s WebREPL port by clicking on the ‘Connect’ button on the http connection shown by the client in the browser window and entering the password that was chosen when the *webrepl_setup* module was run. Note that the password that is entered into this prompt is not echoed. At that point, you should have a similar interface with the RespiCo MicroPython interpreter that was observed via the one established using the USB serial connection.

From a safety standpoint, RespiCo should only be used when its outputs will not interfere with other critical tasks. Moreover, the electrical safety of the haptic device needs to be considered if it is placed in direct physical contact of the user.

Operation instructions

A flow diagram of the RespiCo user interface is shown in Fig. 3. First establish a serial or Wi-Fi REPL connection with RespiCo. If the TEST_DEMO flag is set in the *RespiCo_customizations.py* file, which is default, then the user is first given the option to test the auditory, visual, haptic, and digital outputs of the device and then to execute a short demonstration of a breathing training session that cycles through the visual and auditory outputs. If desired, this flag can be set to *False* to disable these demonstration modes.

When encountering the main RespiCo program loop, the user is given the option to either reuse the prior output protocol, if it exists, or define a new one to be used during the current session. To create a new protocol, the user first chooses between training *features*. These include respiratory muscle training, mindfulness breathing, or athletic training. Each of these selections leads the user to breathing *pattern* options that have been described in the literature for the feature. But if the user chooses the research feature, they are prompted to pick either a default pattern or a custom one. The default pattern can be designed by the research team and put in the *RespiCo_customizations.py* file for reuse during other experiments. If the custom feature is chosen, however, no predefined pattern is offered. In this case, if the user is prompted to define it according to their current needs. After the patterns are selected, the user selects the RR they would like RespiCo to use to produce breathing cues. This includes a default, pre-defined single or series of variable RRs. They also have the option to define a custom single or series of RRs. The user is then asked to select the duration of the breathing instruction; they can select either a default training period or enter a custom one. Lastly, the user then decides what output mode they would like from the device: visual through the RGB LED, auditory through the passive piezoelectric buzzer, or haptic through a connected device. For this build, the haptic output is provided by a coin-size, vibrating ERM motor, which can be attached to the user’s skin. To facilitate the use of RespiCo in a research and development environment, serial output is also available from RespiCo. If the user chooses the visual or audible output, digital output representing the inhalation and exhalation cycles are also served via two GPIO pins of the device. Additional features are provided to enhance the utility of the device. If the auditory output is chosen, then differences in the pitch of the buzzer output instruct the user whether to inhale or exhale. Moreover, if a variable RR is selected to inform the breathing, a decreased inspiratory pitch level is provided by RespiCo two inhalations before the impending RR change. For the visual output, an RGB LED is used, which emits green light to instruct inhalation, blue to signal breath holding, and red to direct exhalation. The haptic motor is also driven by RespiCo at different voltages to cause vibration at two levels instructing inhalation and exhalation. After the breathing instruction session is completed, the user is given the option to continue or to stop the training. If additional instruction is desired, then the subject is offered the choice to repeat the same or select a new set of breathing instruction parameters. The device can then be shut down using the sliding switch, which also conserves battery power if one is used, or by disconnecting the USB power cable.

Through modifiable flags in the *RespiCo_customizations.py* file, the device can be instructed to execute several options that are reported through the Wi-Fi or serial cable interfaces. For example, the device can be configured to provide detailed information about the timings of the respiratory patterns and rates, and send to the user through the serial port instructions that can be read by an attached terminal about how to perform the respiratory muscle training and mindfulness breathing. Moreover, if documentation of the precise breathing rates produced by the device is desired, the software can be configured to

measure the rate through a separate internal timer, and transfer the information through the serial port as well as save the data to a file entitled "RR_data.txt" that is later accessible through the device's directory.

Characterization and Validation

RespiCo was validated using two approaches. The timing of the breathing pattern phase and RR cues from the device were determined by capturing the outputs from its digital ports using pulse analyzers. Also, the effectiveness by which the device directed human subject breathing rates was determined by measuring their RRs using spirometry.

Each breath cycle instructed by RespiCo is broken down into inhalation, inhalation pause, exhalation, and exhalation pause phases; the duration of each phase is determined algorithmically by RespiCo as a simple product of its relative portion of the breath, determined by counts or fractions, and the duration of the respiratory cycle. Examples of these calculated timings across a wide range of RRs are shown in RespiCo_pattern_timings.xlsx in the Supplemental materials. To characterize the accuracy of the phase cues, digital signals from the RespiCo output ports were acquired at a 20 kHz sampling rate using a logic analyzer (HiLetgo 3-01-0015) and analyzed using open source software (pulseview [28]). The patterns were tested at 8 BPM, because this RR is often used during physiological studies [29]. Two cycles of the respiratory pattern signals that were output by RespiCo are shown in Fig. 4A. To validate the accuracy of the phase timings, the output signals were collected for 5 breath cycles and the duration of each pattern phase was determined. Fig. 4B shows the expected phase timings in the table and the measured phase durations in the boxplot. These data show that the device produced accurate breath phase cues of the expected duration with very little variance in their measurement values. The mean time error for the phases was < 0.01 s (data shown in RespiCo_pattern_time_data.pdf in Supplemental materials).

To validate the accuracy of the RR signals produced by RespiCo, we measured the inhalation cue rate provided by the device's output ports over 30 min while it was instructed to produce a wide range of RR signals. We used 6–24 BPM RRs because this range includes the normal RRs of adults (12–18 BPM, [30]) and likely to be used by builders of this device. The rate of the RespiCo inhalation digital output signal at the varying produced RRs was measured using an external pulse analyzer with 10^{-3} s resolution. As seen in Fig. 5A, a linear relationship was detected between the RespiCo-produced and external device-measured RRs, with $R^2 = 1.0$ and coefficients (95 % confidence bounds): slope 0.9946 (0.9945, 0.9948) and intercept 0.0672 (0.0644, 0.0701). We next determined the RR error generated by the device. The error was defined as (measured RR - produced RR); the relative error was defined as $100 \times (\text{error} / \text{produced RR})$. As seen in Fig. 5B, the RespiCo errors were very small and had little variance. But at the higher RespiCo produced RRs, the measured values deviated from the ideal ones. An increasing negative error of the device-produced RR cues was detected at RRs greater than 18 BPM. Nevertheless, even at a RR of 24 BPM, the relative error was < 0.5 %, representing ~ 0.1 fewer BPM signaled by RespiCo.

To test the clinical utility of the device, RespiCo was used to instruct four adult subjects to breathe according to auditory cues, which were time-stamped to allow the determination of the produced RRs. Simultaneously, the subject RRs were measured spirometrically using a clinical device whose clock had been synchronized with RespiCo. The subjects had a tight-fitting mask applied over their nose and mouth, which was held in place with a head strap, to capture their inhaled and exhaled gases. The mask was connected via an Ayre's T-piece to a ~ 6 foot-long semi-closed gas-breathing circuit, which was continually flushed with ~ 10 LPM O₂. In turn, the breathing circuit was connected to an anesthesia machine (Dräger Apollo). The anesthesia machine monitor integrated the breath-induced change in gas flow in the breathing circuit to derive the RR. The RR data were collected from the anesthesia machine spirometer using software provided by the manufacturer (Dräger proto.exe); the software also decoded and error checked the Dräger RS 232 Medibus protocol-formatted RR datagrams. The subjects were instructed to breathe according to the RespiCo-produced cues while they were in supine and then standing positions. A cue pattern of 3, 0, 7, 0 breathing phase counts was chosen because it approximated the inspiratory: expiratory time ratio observed in the experimental subjects. The auditory cue feature of RespiCo was used for this experiment because it was likely to produce the most consistent stimuli for breathing given the restrictions incurred by the face-mask, breathing circuits, and subject position changes. The alternating low and high RR cues was used to mitigate potential changes in the subject's CO₂ levels associated with changes in their minute ventilation at the more extreme breathing rates.

The subject position did not influence the measured RRs. The relative errors between the RespiCo produced and measured RRs pooled across all subjects, were not different in the experimental positions (t -test $P = 0.1683$). Accordingly, the measured supine and standing RR data were combined for each subject in performing the subsequent analyses. Fig. 6 shows the RRs of the subjects measured using spirometry in response to each of the desired breathing rates instructed by RespiCo. There was a high concordance and low variance between the RespiCo-instructed and measured RRs for all subjects. In particular, the mean relative errors (%) between the RRs instructed by RespiCo and those measured in the subjects by spirometry were: 11.62, 4.87, 3.89, 2.36, 1.27, 0.65, 0.20, 0, 0, and 0 for RRs of 6, 8, 10, 12, 14, 16, 18, 20, 22, and 24 BPM, respectively.

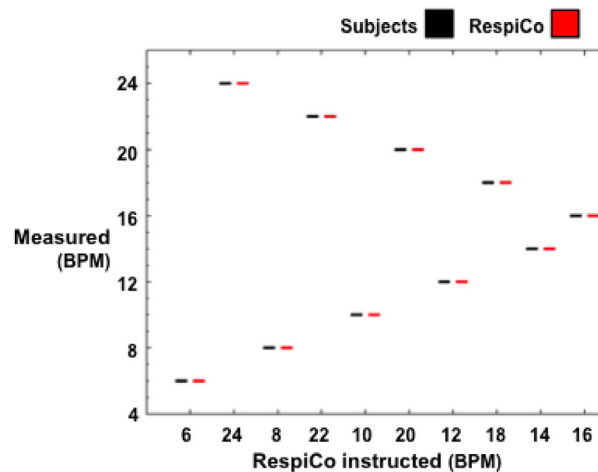


Fig. 6. RespiCo performance using human subjects. RespiCo accurately stimulates adult human subjects to breathe at desired respiration rates. Four subjects were coached to breathe as instructed by the auditory output from the device at the indicated rates while they were in supine and standing positions. Their RRs were quantified by spirometry using a standard ventilator; the RespiCo cue rate was determined by analyzing the time-stamps associated with the breathing cues. Because the errors associated with subject position were not different across the RRs, those data were combined and shown. Boxplots depict the spirometrically measured (black) and pulse-analyzer measured RespiCo coached (red) RRs. A high concordance was observed within and between the subject and RespiCo RRs; the tight data did not allow the boxplot features to be distinguished. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

In summary, we describe and demonstrate a novel, inexpensive, highly flexible device that provides precise respiratory pattern and rate instruction that is optimized for respiratory muscle training, mindfulness breathing, athletic breathing optimization, and experimental use.

Human and animal rights

The work described was carried out in accordance with The Code of Ethics of the World Medical Association (Declaration of Helsinki). The Mass General Brigham Human Research Committee (MGB) and the Institutional Review Board of the MGB approved the study. Informed consent obtained for experimentation with human subjects and the privacy rights of the human subjects were always observed.

CRedit authorship contribution statement

Kanchan Kulkarni: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Resources, Data curation, Writing – original draft, Writing – review & editing, Visualization. **John H. Nichols:** Conceptualization, Validation, Investigation, Writing – review & editing. **Antonis A. Armoundas:** Conceptualization, Investigation, Resources, Data curation, Writing – review & editing. **Jesse D. Roberts Jr.:** Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Resources, Data curation, Writing – original draft, Writing – review & editing, Visualization, Supervision, Project administration, Funding acquisition.

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.

Acknowledgments

The National Institutes of Health (Grants R01HL135335, R21HL137870, and R21EB026164 to A.A.A.), and the MGH Department of Anesthesia, Critical Care, and Pain Medicine (J.H.N. and J.D.R.) supported this work.

References

- [1] A.J.R. van Gestel, M. Kohler, J. Steier, S. Teschler, E.W. Russi, H. Teschler, The effects of controlled breathing during pulmonary rehabilitation in patients with COPD, *Respiration* 83 (2) (2012) 115–124.
- [2] A.E., Holland, et al., *Breathing exercises for chronic obstructive pulmonary disease*. Cochrane Database Syst Rev, 2012. **10**: pp. CD008250.

- [3] T.A. Santino, et al., *Breathing exercises for adults with asthma*. Cochrane Database Syst Rev, 2020. **3**: pp. CD001277.
- [4] L. Bernardi, G. Spadacini, J. Bellwon, R. Hajric, H. Roskamm, A.W. Frey, Effect of breathing rate on oxygen saturation and exercise performance in chronic heart failure, *Lancet* 351 (9112) (1998) 1308–1311.
- [5] G. Bilo, et al., Effects of slow deep breathing at high altitude on oxygen saturation, pulmonary and systemic hemodynamics. *PLoS One*, 2012. **7**(11): p. e49074.
- [6] R.S. Fitzgerald, N.S. Cherniack, Historical perspectives on the control of breathing, *Compr Physiol* 2 (2) (2012) 915–932.
- [7] J. Ebnezar, R. Nagarathna, B. Yogitha, H.R. Nagendra, Effects of an integrated approach of hatha yoga therapy on functional disability, pain, and flexibility in osteoarthritis of the knee joint: a randomized controlled study, *J Altern Complement Med* 18 (5) (2012) 463–472.
- [8] P. Holm, A. Sattler, R.F. Fogosi, Endurance training of respiratory muscles improves cycling performance in fit young cyclists, *BMC Physiol* 4 (2004) 9.
- [9] S.K. Illi, U. Held, I. Frank, C.M. Spengler, Effect of respiratory muscle training on exercise performance in healthy individuals: a systematic review and meta-analysis, *Sports Med* 42 (8) (2012) 707–724.
- [10] E. Bernardi et al, Respiratory muscle endurance training improves breathing pattern in triathletes, *Ann. Sports Med. Res.* 1 (2014) 1003.
- [11] s. Soni, L.N. Joshi, A. Datta, Effect of controlled deep breathing on psychomotor and higher mental functions in normal individuals, *Indian J Physiol Pharmacol* 59 (1) (2015) 41–47.
- [12] Bloomfield, D.M., et al., *Comparison of spontaneous vs. metronome-guided breathing on assessment of vagal modulation using RR variability*. *Am J Physiol Heart Circ Physiol*, 2001. **280**(3): p. H1145-50.
- [13] J.M. Ramirez, C.S. Ward, J.L. Neul, Breathing challenges in Rett syndrome: lessons learned from humans and animal models, *Respir Physiol Neurobiol* 189 (2) (2013) 280–287.
- [14] A. D'Arrigo, S. Floro, F. Bartesaghi, C. Casellato, G.F. Sferazza Papa, S. Centanni, A. Priori, T. Bocci, Respiratory dysfunction in Parkinson's disease: a narrative review, *ERJ Open Res* 6 (4) (2020) 00165-2020.
- [15] *Diaphragmatic breathing*. 2022 [cited 2022 Jan 25]; Available from: <https://my.clevelandclinic.org/health/articles/9445-diaphragmatic-breathing>.
- [16] *Pursed lip breathing*. 2022 [cited 2022 Jan 25]; Available from: <https://my.clevelandclinic.org/health/articles/9443-pursed-lip-breathing>
- [17] D.G. Doehring, Accuracy and consistency of time-estimation by four methods of reproduction, *Am J Psych* 74 (1) (1961) 27–35.
- [18] F.T. Bruss, L. Ruschendorf, On perception of time, *Gerontology* 56 (2009) 361–370.
- [19] S. Grondin, Timing and time perception: A review of recent behavioral and neuroscience findings and theoretical directions, *Attent. Percept. Psychophys.* 72 (3) (2010) 561–582.
- [20] S. Pospos, I.T. Young, N. Downs, A. Iglewicz, C. Depp, J.Y. Chen, I. Newton, K. Lee, G.A. Light, S. Zisook, Web-Based Tools and Mobile Applications To Mitigate Burnout, Depression, and Suicidality Among Healthcare Students and Professionals: a Systematic Review, *Acad Psychiatry* 42 (1) (2018) 109–120.
- [21] K.R. Mahtani, T. Beinortas, K. Bauza, D. Nunan, Device-Guided Breathing for Hypertension: a Summary Evidence Review, *Curr Hypertens Rep* 18 (4) (2016).
- [22] M. Pagani, F. Lombardi, S. Guzzetti, O. Rimoldi, R. Furlan, P. Pizzinelli, G. Sandrone, G. Malfatto, S. Dell'Orto, E. Piccaluga, Power spectral analysis of heart rate and arterial pressure variabilities as a marker of sympatho-vagal interaction in man and conscious dog, *Circ Res* 59 (2) (1986) 178–193.
- [23] E.G. Collins, C. Jelinek, S. O'Connell, J. Butler, D. Reda, F. Laghi, The Effect of Breathing Retraining Using Metronome-Based Acoustic Feedback on Exercise Endurance in COPD: A Randomized Trial, *Lung* 197 (2) (2019) 181–188.
- [24] C. Watson, S. Senyo, All-in-one automated microfluidics control system., *HardwareX* 5 (2019) e00063.
- [25] I. Koene, V. Klar, R. Viitala, IoT connected device for vibration analysis and measurement, *HardwareX* e00109 (2020) 1–15.
- [26] T. Abuzairi, A. Irfan, Basari, COVENT-Tester, A low-cost, open source ventilator tester, *HardwareX* 9 (2021) e00196.
- [27] George, D.P. *MicroPython license*. 2022 [cited 2022 Feb 3]; Available from: <https://raw.githubusercontent.com/micropython/micropython/master/LICENSE>.
- [28] sigrok, sigrok - a portable, cross-platform, Free/Libre/Open-Source signal analysis software suite.
- [29] M.A. Russo, D.M. Santarelli, D. O'Rourke, The physiological effects of slow breathing in the healthy human, *Breathe (Sheff)* 13 (4) (2017) 298–309.
- [30] MedlinePlus. *Vital signs*. [cited 2022 June 6]; Available from: <https://medlineplus.gov/ency/article/002341.htm>.



Kanchan Kulkarni received her M.S. in Electrical Engineering (2013) and Ph.D. in Biomedical Engineering (2017) from the University of Minnesota. She completed a 3-year Research Fellowship (2020) at the Cardiovascular Research Center at the Massachusetts General Hospital (MGH) and Harvard Medical School. Currently, she is a Postdoctoral Researcher at the University of Bordeaux and L'institut De Rythmologie Et Modélisation Cardiaque, France. Dr. Kulkarni's research interests include cardiac electrophysiology, prevention and control of cardiac arrhythmias, and neuromodulation techniques for cardiovascular disease management.



John Nichols received his A.B at Dartmouth College in Engineering modified with Economics, M.D. from the Johns Hopkins School of Medicine, and completed clinical training in anesthesia and pediatric anesthesia at Harvard Medical School. In 2017, he joined the Department of Anesthesia at MGH. Currently he is the Chief of Pediatric Anesthesiology in that Department and in the MassGeneral Hospital for Children. Dr. Nichols is an Instructor in Anesthesiology at Harvard Medical School. His research focus is on device development for patient monitoring, specifically in developing novel methods of quantitative respiratory monitoring, and their combination with machine learning to predict patient events and outcomes.



Antonis Aroundas was born in Mytilini, Greece. He received the B.S. degree in electrical engineering from the National Technical University of Athens in 1991, and M.S. degree in biomedical engineering from Boston University in 1994. He received his Ph.D. degree in nuclear Engineering, from the Massachusetts Institute of Technology (MIT) in 1999. He has been an American Heart Association-sponsored Postdoctoral Fellow in the Department of Medicine Division of Molecular Cardiobiology and the Department of Biomedical Engineering at the Johns Hopkins University. Currently, he is a Principal Investigator in the Cardiovascular Research Center at MGH, a Visiting Scientist at MIT, and an Associate Professor of Medicine at Harvard Medical School. Dr. Aroundas's research interests include the development of biomedical signal processing and machine learning algorithms aiming to improve the diagnosis and treatment of cardiovascular and respiratory disease.



Jesse Roberts Jr. received his M.S. in biochemistry, and M.D. at the University of Rochester in 1982. After completing clinical training in pediatrics at the University of California in San Diego, in neonatology at Brown University, and in anesthesia and pediatric anesthesia at Harvard Medical School, he joined the Departments of Anesthesia, Pediatrics, and Medicine at MGH in 1990. He completed National Institutes of Health (NIH)-sponsored postdoctoral physician-scientist training in molecular biology and then established an NIH National Heart Lung Institute supported laboratory studying pediatric pulmonary developmental and vascular diseases in the Cardiovascular Research Center at MGH. He is an Associate Professor at Harvard Medical School. Dr. Roberts's research interests include novel mechanisms, technologies, and therapeutics of pediatric lung diseases.