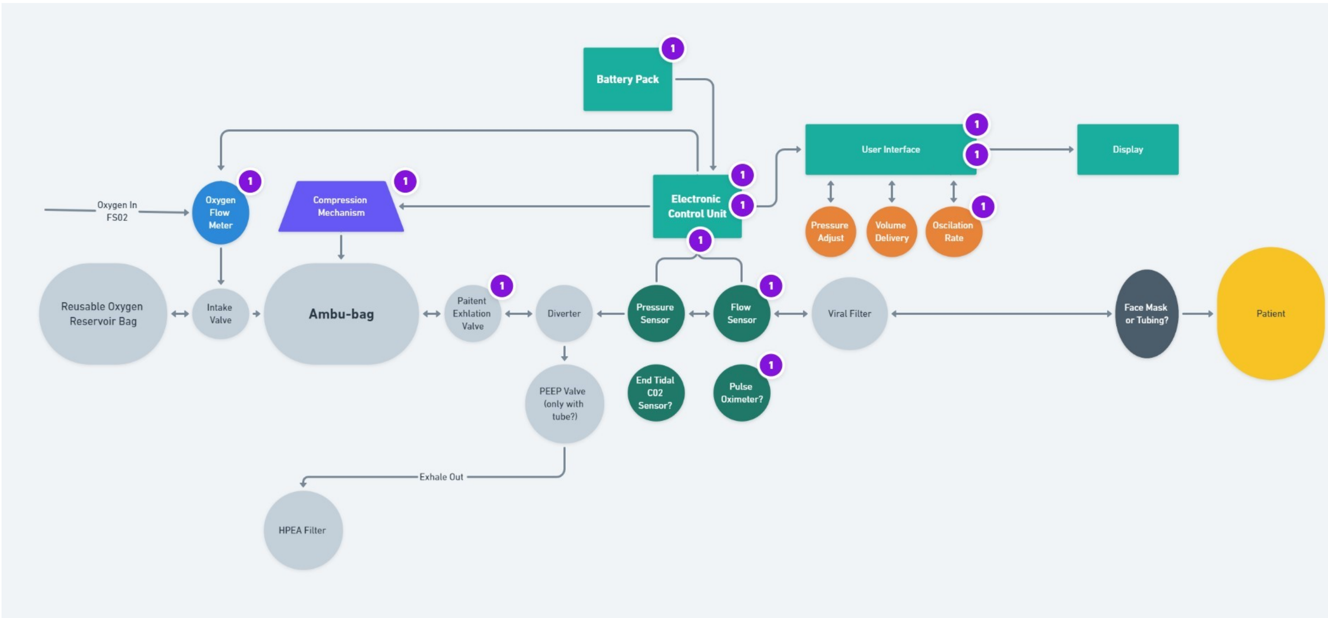
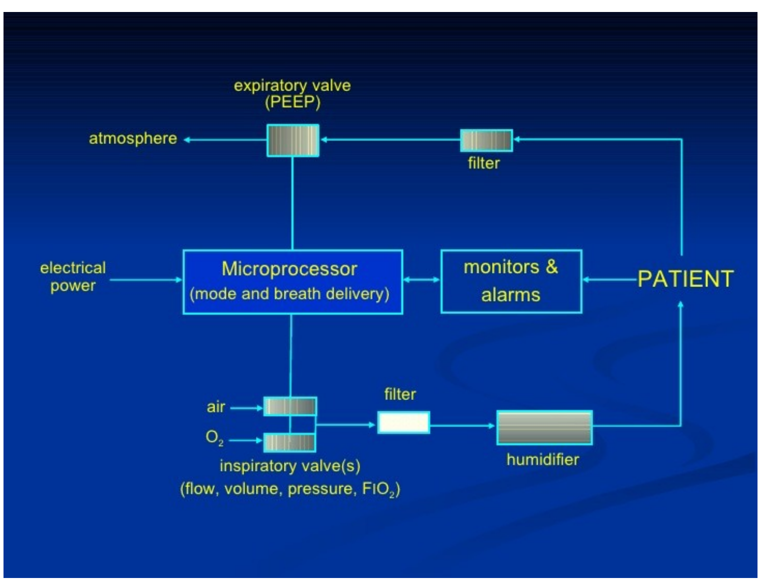
Organization: Opensourceventilator.ie
Document: breakdown and analysis of electrical and software needs
Author(s): Paul McGowan, Steve Siebert, (add your name if you add content!)
Version: 1.0



Purpose:
The purpose of this document is to break down the flowchart above into component parts and form a segmented specification for hardware and software. This information may then be distributed and the larger task of developing the system can be broken down and tackled by the plethora of talented volunteers on the project team Slack page.

Other sources:
In working on complicated projects, it is worthwhile to see how others have done / explain the same concepts. Here is an alternative diagram from https://www.slideshare.net/scribeofegypt/anatomy-of-the-ventilator – accessed March 20th, 2020.



Oxygen flow meter:
The most common method of Oxygen flow metering, or mass flow metering, is a thermal flow meter. The basic principle is that the gas flow passes a filament and the temperature of the filament is related to the amount of Oxygen present. I believe it gets hotter as more oxygen passes. Example sensor:
https://www.digikey.com/product-detail/en/renesas-electronics-america-inc/FS1012-1100-NG/800-3790-ND/7645218

Compression mechanism:
This is the physical mechanism that squeezes the bag. Currently being prototyped (#6 and #7 as of March 20th, 2020), the consensus is a NEMA stepper motor, which will involve a pulsed DC drive of either unipolar or bipolar means at frequencies (estimated) <= 1kHz with (estimated) 10-50% duty cycle. The amount of current necessary per pulse is TBD.

Insert any other electrical specs or motor examples here.

Pressure sensor:
A pressure sensor, potentially a piezoelectric/diaphragm-type, is necessary to monitor pressure between apparatus and patient. Potentially digital or analog output. This is usually either a digital count between a minimum and maximum, or a conditioned analog output as a percentage of some maximum.

A digikey search honing in on a potential candidate:
https://www.digikey.com/products/en/sensors-transducers/pressure-sensors-transducers/512?k=pressure+sensor&k=&pkeyword=pressure+sensor&sv=0&pv1878=1211&pv1878=2087&pv1878=2171&pv1878=2406&pv1878=32407&pv1878=33119&pv1878=36058&pv1878=41594&pv1878=51779&pv1878=52273&pv1878=58520&pv1878=59969&pv1878=168620&pv1878=127644&pv1878=135508&pv1878=202501&pv1878=174935&pv1878=176796&pv1878=176936&pv1878=249096&pv1878=230290&pv1878=234537&pv1878=267210&pv1878=250771&pv1878=269078&pv1878=269197&pv1878=270753&pv1878=271035&pv1878=297914&pv1878=298341&pv1878=74299&sf=1&FV=1989%7C0%2C-8%7C512%2C218%7C307133%2C218%7C338588%2C218%7C338649%2C218%7C350336%2C218%7C400888%2C218%7C419488&quantity=&ColumnSort=1000011&page=1&stock=1&rohs=1&pageSize=25

Perhaps this component would be good as a start:
https://www.digikey.com/product-detail/en/omron-electronics-inc-emc-div/2SMPP-02/Z3059-ND/2504247

This OMRON component will need its analog output (0-31mV) to be scaled and filtered to meet the demands of a microcontroller ADC input.

Flow sensor:
Again, a flow sensor is needed and may be this device:
FS1012-1020-NG

However, as the flow rate may be affected by humidity, a humidity sensor may also be necessary to correct the readings. *Propose to team and look into this.*

Electronic control unit:
There are a wide variety of available devices for use to fill this need. Below is a list of minimum specifications, keeping in mind all of the sensors that are necessary to support, the motor drive, the user interface, etc.

Motor drive:
Brute-force pulses into transistor drivers or gated pulse-width modulation may be the two main sources of stepper motor drive signaling. These signals may or may not need to be optically-isolated from the motor drive circuitry itself.

*__Pulse-width modulation or precision timing__ capability for this purpose.

Flow sensors(2), pressure sensor (1), Oxygen pulse oximeter (1), End tidal CO2 sensors:
Though the exact configuration is not yet known, one can err on the side of caution and propose a minimum amount of capabilities:

*__Between 5 and 10 ADC channel__ (10 bit minimum?); monitoring both sensor input, power supply levels, battery charge characteristics, and other yet unknown parameters

*__(Synchronous) serial port capabilities__; SPI, I2C, UART for debugging and/or networking capability (debug mode or normal mode?)

*__General pulse I/O with interrupt-on-change capabilities__ for use with limit-switch sorts of I/O, user interface, LED indication, bit-banging for 7-segment display via shift registers, etc. etc.

*__A second serial port__ for interfacing to a multi-channel DAC or one of several variable potentiometers for use in trimming/calibrating analog signal conditioning circuitry

*__Quadrature encoder module or similar__ functionality to maintain the current position of the motor (note that if a stepper does not step, the pulse would be counted but the position would be unknown; care must be taken as to how we do this)

*__CTMU or equivalent way__ to gather user “button presses”, however they may arrive and through whichever mechanical (or electrical) means decided upon

- **IOC button presses**? Cheap, needs hardware; pull-ups built-in, software filtering
- **Capacitive touch pads** on PCB? Cheap, needs software resources; timer and CTMU peripheral or external cap and current source
- **Rotary encoder**? Hardware and software resources, not as cheap, pull-ups built-in, software filtering, IOC CPU-intensive for short bursts
- **Membrane switch**? Cheap in mass quantity, needs hardware, pull-ups built-in, software filtering, may not last as long as other options

Capacitive sensing least ESD sensitive if board layout done properly, and hardware cheap (free copper!) but software resources and stability obviously a concern. Ways to deal with this, though.

I am personally familiar with 8-bit PICs; 18F4550, 16F877a, 16F1937, 12F683,...the list goes on. I see a device with no less than 44 pins, as a first guess, based on the list above. I/O can be expanded with 8-bit I2C expander ICs, and I2C with addressable devices; SPI with 1-of-8 decoder chips...will have to extrapolate and figure it out. MPLAB has automatically-generated drivers for their peripherals on most chips, and Microchip has plenty of following for support.

Arduino is naturally another popular contender, as is the ESP32 for its integrated IoT capabilities.

Steve Siebert and I are looking into those three options.

Display:
In this particular application, I think the local electronic control unit should have user diagnostic LEDs and/or a character LCD, along with a data output port to a more capable graphical LCD solution.

Whether this is a Rpi-zero, PC running linux, (whatever)pad with an app,...this is TBD. If I had to do something quickly, it would be C# form app using their chart control for easily-generated and controlled line strip charts. I have run C# via wine on raspian with ease in the past. However, others will know the more convenient and efficient method of solving this problem. Please add info here!

Overall thoughts on architecture of the system:
The want and/or need of an IoT-capable system has been expressed by Steve Siebert and myself, and we envisioned at this time single self-contained electronic systems that have the capability of publishing results via request from a main hub, or via asynchronous means to conserve battery life on portable units. In either case, he has some ideas he is working on and perhaps he can elaborate here.

Battery power:
For portable applications, an efficient solution would allow battery power to be utilized. Though the time constraint is not yet known (to me), perhaps a good rule-of-thumb to start is a 5 hour trip in an ambulance or other transportation to a hospital.

Steve and I discussed battery technology; in the interest of time, SLA (sealed lead-acid) battery technology is a bit heavy and antiquated, but the charge circuitry is very straightforward and these batteries are available practically everywhere. A 5-10 AH battery @ 12V may be plenty for the 50% duty cycle (estimate) motor needs and electronic control subsystem(s). Potential connection to solar panel might become a need as well; although I have not seen this come up thus far. I imagine that, in the interest of the application, we would prefer to stick to 24V and below to protect the user and to keep with commonly-available boost and buck regulators, should we need them. Stepping 12 VDC to 24 VDC for low duty cycle motor drive is not too big a task, but 12 VDC to 60VDC+ introduces new issues and inefficiencies without quality parts and lots of thought.

TODO:
More information on which sensors, what interfaces, mounting, constraints, costs
More information on “how big the box” is, user interface, power consumption
More information on power requirements, lifetime on battery, battery tech, etc.
More information on motor specifications and position sensing
More information on overall system architecture
etc etc.