**FIGHT COVID19 – Autonolous Ambu Support**

**v.0.2**

**Sergio Sestili**

**Abstract**

This document presents a general design for an Automatic **Ambu** Support, aimed to automatize its usage in emergency conditions where there are not enough personnel to assist people who are in need for a breath support.

Several projects are flourishing in these days. This document is meant to be a contribution with some ideas, totally for free forever, available to all community.

It is important to clarify that this project is not the description of a design of a certified medical product.

**Requirements**

In the last days I’ve been thinking at various ways to automatize Ambu operations, and during this path I realized that a reliable Ambu support must match the following minimum requirements:

1. Simulate movement for velocity and pressure as human hand does
2. Provide flexibility and dynamic parametrization to adjusts operation
3. Be built for long duration usage
4. Have a user interface as much intuitive as possible
5. Being affordable
6. Being quick to produce
7. Based on off the shelf commercial affordable components or 3D printable components
8. Possibly provide wifi connection for remotely alerting or managing

**The main idea**

In order to match the above defined requirements, I came up with two different solutions.

Solution 1

This first solution is based on 3D print parts to build a custom structure to house an Ambu

1. Mechanic: based on technologies used to build 3D printers, with Step Motors
2. Electronic: use a Raspberry PI 3+ as control computer together with a motor driver board
3. Software: python on raspberry
4. GUI: a tablet or mobile phone can be used to manage the device

The main idea is

* to build a structure that contains an Ambu
* with two step motors having two simulated-hand to gently press the Ambu
* the motors are directly controlled by the raspberry that implements the “business logic”
* an initial calibration phase is needed
  + to position the hands in proximity of the Ambu for defining
    - min rest position for each hand and
    - max movement position for each hand

Step motor 1

Threaded rod

Threaded rod

Simulated hand

Ambu

Frame to fix Ambu

To facial mask

Raspberry

Simulated hand

Step motor 2

Fig. 1 - Solution 1 sketch

This solution is easy to build and possibly cheap, but it requires some time to build the mechanical part. A quickest solution is the following

Solution 2

This solution uses as mechanical module an entire 3D printer

1. Mechanic: Use a small 3D printer, i.e. Monoprice
2. Electronic: use a Raspberry PI 3+ as control computer that
   1. Implements the business logic and
   2. interface to the 3D printer to send G-code commands in real time
3. Software: Octoprint + Octoprint API + python business logic on raspberry
4. GUI: a tablet or mobile phone can be used to manage the device

The main idea is:

* Ambu is installed firmly in the print plate of the 3d Printer
* Instead of the extruder, there is a simulated-hand that will press the Ambu
* the Z axis of the 3D printer is the movement to be controlled by raspberry in order to press the Ambu
* the raspberry software will implement the “business logic” and will command directly the printer sending G-code commands via serial connection
* a pre-production calibration phase is needed to position the hand in proximity of the Ambu to define min rest position and max position

This solution may cost a little more because it requires a commercial 3D printer (and uses just a fraction of the 3d printer features), but it is quickest to be realized. In addition, by interfacing with an already produced 3D printer, the mechanic and electronic part is already tested and reliable.

Off the shelf

3D Printer

Raspberry

Simulated hand

Ambu

Serial connection

To facial mask

Fig. 2 - Solution 2 sketch

**Business Logic**

The main module of the software that runs on raspberry is defined as AMBUManager. It commands motors (solution 1) or 3D Print (solution 2) to gently press the Ambu and simulate human operation.

The following configuration parameters are to be implemented

* Ambu press velocity (proportional to the gas pressure injected)
* Ambu press duration (proportional to the gas volume injected)
* inspiration end pause (pressed Ambu)
* espiration end pause (not pressed Ambu, is the min rest position for the simulated hand/s)

**User Interface**

Both solutions need a GUI to let the human to

* initialize the system
* periodically adjusts working parameters

Such interface can be implemented with a web application running on the raspberry that updates a configuration file that is used bu the business logic module.

**Sensors**

For incrementing software reliability, several sensors can be applied to be included in the raspberry business logic, but all this will need more time to consolidate the software itself