Emergency Ventilators

Implementation details

NOTE

The concept demonstrated in this document is based on professional experiences in the design and development of life-support ventilators. The concept has not been prototype tested and will need validating. Many details are yet incomplete in the current draft. The concept is open source – free to use, evolve or transfer in parts into other designs.

Issue/change record

|  |  |  |  |
| --- | --- | --- | --- |
| Issue | Date | Author | Reason of issue/summary |
| 0.1 | 13/04/2020 | Carlos Pardo | First draft |
| 0.2 | 19/04/2020 | David Ortiz | Added Controller Initialization routine. Updated I2C protocol description. |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Contents

[Introduction 3](#_Toc37756879)

[1 Controller algorithms 3](#_Toc37756880)

[1.1 Pressure control 3](#_Toc37756881)

[1.2 Air flow / Volume measure 3](#_Toc37756882)

[1.3 display & buttons functionality 3](#_Toc37756883)

[1.4 power on/off management 3](#_Toc37756884)

[2 Monitor algorithms 3](#_Toc37756885)

[2.1 Pressure monitoring 3](#_Toc37756886)

[2.2 alarm control 3](#_Toc37756887)

[2.3 display & buttons functionality 3](#_Toc37756888)

[3 Controller <-> monitor communication 4](#_Toc37756889)

[3.1 i2c information 4](#_Toc37756890)

[4 assembly 4](#_Toc37756891)

[4.1 placement of components 4](#_Toc37756892)

[4.2 3d printed parts 4](#_Toc37756893)

# Introduction

This document contains information about the implementation not described neither in the specification, neither in the user manual

# Controller algorithms

## Initialization

Every time the controller starts perform a self-test according to the following sequence:

* 5V supply level of the Controller is measured during 500ms.
  + Max and Mean values are stored. Mean value is calculated as (Max+Min)/2.
  + The Mean value must be between 4.75 and 5.25V.
  + Max-Min must be below 100mV.
  + If any of the levels is outside limits, an error will be displayed and Controller will restart the initialization routine.
* After Vdd measurement, the controller will start blinking the LCD backlight, and signal the user to leave the output airway open to perform the pressure test. Pressure test will start once Breath button is pressed for 2 seconds.
  + After Breath button is pressed, the pressure sensors will be measured during 500ms.
    - Max, Min and Mean values will be measured as in the Vdd case.
    - It will be checked that Mean value lies within limits, and that the difference between Max and Min is small. Mean value will be internally stored and used as sensor offset level.
  + After this, the output SV2 and SV3 valves will be opened during 0.5 ms.
    - Flow rate will be measured and displayed on the screen.
    - If flow rate is too low or too high an error will be displayed and the Initialization routine will be restarted.
    - If flow sensor output is saturated during the flow measurement, no flow value will be displayed. The Initialization routine will report an error and will be restarted.
* Finally, if Initialization is correct, the user will have the possibility of restarting it again (useful during setting-up of the internal pressure regulator), or exit the initialization.

After initialization is finished, the Controller will go to STOP state. In this state only the menu for setting the different parameters is enabled, but the Air Flow will be closed. A message will appear to indicate that Breath key must be pressed to start normal operation.

Once Breath key is pressed for >2 seconds, normal operation of the controller will start.

## Pressure control

## Air flow / Volume measure

## display & buttons functionality

## power on/off management

# Monitor algorithms

## Pressure monitoring

## alarm control

## display & buttons functionality

LCD is an I2C controlled Display. The following information is sent via I2C to control the display:

I2C Address: 0x27

I2C WriteByte:

Bits: 7. 6. 5. 4. 3. 2. 1. 0

Data: D7.D6.D5.D4.LED.E.RW.RS

# Controller <-> monitor communication

## i2c information

The controller communicates with the Monitor every 50ms.

To allow for bidirectional communication, each 50ms the controller will first perform an I2C write burst to the Monitor, and then it will perform a read burst from the Monitor to get its status.

The content of the write command will be as follows:

Master: Controller

Slave: Monitor

Monitor Address: 0xA0 (8 bit address) 0x50 ( 7 bit address)

Num of bytes: 10

Information to be send:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **BYTE** | **BIT** | **NAME** | **SIZE** | **DESCRIPTION** |
| **0** | 7:1 | ADDR | 7 | I2C address of monitor: 0x50 |
| **0** | 0 | R/W | 1 | Write always: 0 |
| **1** | 7 | IDLE | 1 | Initialization Active |
| **1** | 6 | PCAL | 1 | Pressure calibration state |
| **1** | 5 | VCAL | 1 | Volume calibration state. All valves open during this state. |
| **1** | 4 | RUN | 1 | RUN/STOP status. |
| **1** | 3 | MODE | 1 | Operating Mode. 0 → Pressure-control. 1 → Volume-control. |
| **1** | 2:0 | RFU | 3 | Reserved for future use. |
| **2** | 7 | IP CH | 1 | IP Changed. |
| **2** | 5:0 | IP | 1 | IP Value set by User. |
| **3** | 7 | PEEP CH | 1 | PEEP Changed. |
| **3** | 5:0 | PEEP | 1 | PEEP Value set by User. |
| **4** | 7 | BPM CH | 1 | BPM Changed. |
| **4** | 5:0 | BPM | 1 | BPM Value set by User. |
| **5** | 7 | PMAX CH | 1 | PMAX Changed. |
| **5** | 5:0 | PMAX | 1 | PMAX Value set by User. |
| **6** | 7 | VMAX CH | 1 | VMAX Changed. |
| **6** | 5:0 | VMAX | 1 | VMAX Value set by User. |
| **7** | 7 | LVA CH | 1 | Low Volume alarm changed. |
| **7** | 6:0 | LVA | 6 | Low Volume alarm value. Given in units of 10ml. |
| **8** | 7 | HVA CH | 1 | High Volume alarm changed. |
| **8** | 6:0 | HVA | 6 | High Volume alarm value. Given in units of 10ml. |
| **9** | 7 | VDDA | 1 | Controller Alarm. 5V line outside limits. |
| **9** | 6:0 | RFU | 7 | Reserved for Future Use. |
|  |  |  |  |  |
| **~~-~~** | ~~-~~ | ~~-~~ | ~~-~~ | ~~-~~ |
| **~~1~~** | ~~5:0~~ | ~~IP~~ | ~~6~~ | ~~Pressure in mBar during inspiration period~~ |
| **~~2~~** | ~~7~~ | ~~LP~~ | ~~1~~ | ~~Alarm: Low pressure (No patient or no input air )~~ |
| **~~2~~** | ~~6~~ | ~~HP~~ | ~~1~~ | ~~Alarm: Pressure too high (Bad functioning or patient cough)~~ |
| **~~2~~** | ~~5:0~~ | ~~EP~~ | ~~6~~ | ~~Pressure in mBar during expiration period~~ |
| **~~3~~** | ~~7~~ | ~~B12~~ | ~~1~~ | ~~Battery voltage bellow 12.5V~~ |
| **~~3~~** | ~~6~~ | ~~B10~~ | ~~1~~ | ~~Battery voltage bellow 10V~~ |
| **~~3~~** | ~~5~~ | ~~B8~~ | ~~1~~ | ~~Battery voltage bellow 8V~~ |
| **~~3~~** | ~~4:0~~ | ~~BR~~ | ~~5~~ | ~~Breath rate in Breath per Minute units~~ |
| **~~4~~** | ~~7~~ | ~~CONF~~ | ~~1~~ | ~~In configuration state~~ |
| **~~4~~** | ~~6:0~~ | ~~VOL~~ | ~~7~~ | ~~Volume Setting in 10 ml units~~ |
| **~~5~~** | ~~7~~ |  |  |  |
| **~~5~~** | ~~6:0~~ | ~~MVOL~~ | ~~7~~ | ~~Volume measured in 10 ml units~~ |

The content of the read command will be as follows:

Master: Controller

Slave: Monitor

Monitor Address: 0xA0 (8 bit address) 0x50 ( 7 bit address)

Num of bytes: 3

Information to be send:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **BYTE** | **BIT** | **NAME** | **SIZE** | **DESCRIPTION** |
| **0** | 7:1 | ADDR | 7 | I2C address of monitor: 0x50 |
| **0** | 0 | R/W | 1 | Write always: 1 |
| **1** | 7 | IDLE | 1 | Initialization Active |
| **1** | 6 | PCAL | 1 | Pressure calibration state |
| **1** | 5 | VCAL | 1 | Volume calibration state. All valves open during this state. |
| **1** | 4 | RUN | 1 | RUN/STOP status. |
| **1** | 3 | MODE | 1 | Operating Mode. 0 → Pressure-control. 1 → Volume-control. |
| **1** | 2:0 | RFU | 3 | Reserved for future use. |
| **2** | 7 | IP CH | 1 | IP Change Ack. |
| **2** | 6 | PEEP CH | 1 | PEEP Changed. |
| **2** | 5 | BPM CH | 1 | BPM Changed. |
| **2** | 4 | PMAX CH | 1 | PMAX Changed. |
| **2** | 3 | VMAX CH | 1 | VMAX Changed. |
| **2** | 2 | LVA CH | 1 | Low Volume alarm changed. |
| **2** | 1 | HVA CH | 1 | High Volume alarm changed. |
| **2** | 0 | BCKSND | 1 | Request to sound backup sounder. |

# assembly

## placement of components

## 3d printed parts