# Ventilator Example with Closed-Loop Control on Low Cost Hardware

## Disclaimer

**This example may not represent an implementable design, and no validation has been done. The purpose of the example is to provide a starting point for designers working on ventilators showing how interfacing between the real-time controller and the system model can be done, how a real-time controller can be defined in Simulink® and Stateflow™ and how a full system model can be used to support the design process.**

## Overview

This example is a modified version of ssc\_medical\_ventilator that includes closed loop pressure control. It is intended to show how a simple controller can be built on a readily available Arduino device. It is also intended to illustrate programming patterns useful in building embedded controllers of this type.

It is written using R2019b and also requires the Simulink Support Package for Arduino to be installed. It consists of the following:

## medicalVentilatorSystemModel.slx

The simulation of the ventilator physical model and its control. Run this model to simulate the controller. You can also use the Simulation Data Inspector to see internal measurements from the controller once the simulation has finished.

## medicalVentilatorSystemParams.m

This is a MATLAB file that contains some of the ventilator and lung parameters you would want to tune as the control device is developed. If a user/developer is familiar with Simulink Data Dictionaries, then best practice would be to migrate these parameters to a Simulink Data Dictionary.

## controlModel.slx

This is the model of just the controller. It is referenced from the other models. The model contains annotations to help the reader understand the purpose of the different parts of the controller. This model also has a test harness on the top level for open-loop testing of the controller.

## controlParams.m

This is a MATLAB file that contains some of the control parameters you would want to tune as a device of this type is developed. If a user/developer is familiar with Simulink Data Dictionaries, then best practice would be to migrate these parameters to a Simulink Data Dictionary.

## controlLibrary.slx

The controller uses some blocks repeatedly, such as filters. Blocks that are repeated should be stored in a library so they can be edited in one place.

## controlArduino.slx

This is a model that is used to build the code that can be deployed onto the Arduino hardware (Mega 2560, but would probably work on other boards too). This builds a hex file that can be programmed onto the device.

## How to run

To get started with the example, first open the project file ventilator.prj. You can then open medicalVentilatorSystemModel.slx, simulate and inspect the logged data and scopes. The hardware build process can be tested by opening controlArduino.slx and pressing the build icon. To run the model at four dial settings and see performance run plotResults.m.

**Please remember this is intended as an example for people to build on if they are working on ventilator projects and is not a working design in itself.**