Virtual Vital Signs Monitor

by Barry M. Robinson, http://www.barrymrobinson.ca
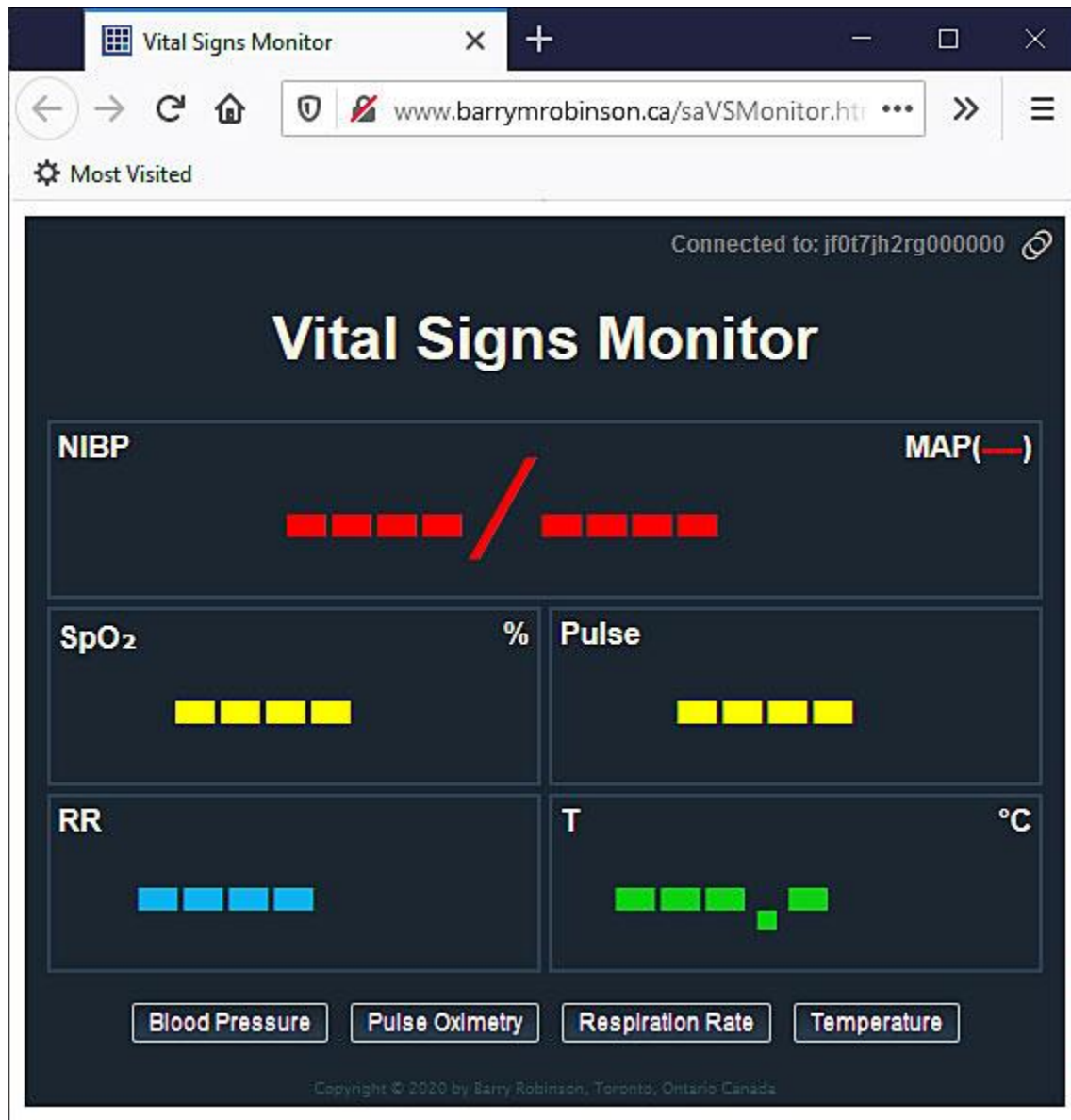
Last update: Saturday, September 12, 2020

This browser-based app allows a learner to operate a vital signs monitor, shown above with the ECG window disabled, in a browser while interacting with a virtual or standardized patient over a separate audio/video link. The connection between the control browser and the monitor browser is a peer-to-peer connection; it does use a server to establish the connection but disconnects from the server once the connection between the two peers is established. All the learner needs is the link, a URL, generated by the Control Window.

When the Control Window is opened, it connects with the peerjs server and is automatically assigned an ID (**jf0t7jh2rg000000** in the example above). The Control Window will generate a link containing the ID which is then sent to the learner. The learner clicks on the link and the Vital Signs Monitor loads and connects with the control window.

Once the monitor is connected, the date and time appears in the response window. The clock in the Control Window is set to 0; synchronized with the remote monitor.

As the scenario progresses, vital signs can be changed on-the-fly. Values can trend (change over time) and the trend can be delayed. Pre-programmed sets of vital signs can be loaded from the controlling computer. As the learner takes the vitals, a time-stamped record appears in the response window. The response window is the area at the top of the Control Window which displays data regarding the current state of the system. Data originating from the Control Window is in the normal font, data sent from the remote Vitals Monitor is *italicized*. The link with the id is in **bold**.

> . . . Control window opened on Sat Sep 12 2020 at 18:53:56 GMT-0400 (Eastern Daylight Time)
>
> Copy the following URL and send to your learner:
>
> **http://www.barrymrobinson.ca/saVSMonitor.html?id=jf0t7jh2rg000000**
>
> *. . . Remote vital signs monitor opened on Sat Sep 12 2020 at 18:57:43 GMT-0400 (Eastern Daylight Time) synchronized*
> *. . . at 01:44 Temperature taken: 37.0°C . . .*
> *. . . at 01:46 Respiration rate taken: 16 . . .*
> *. . . at 02:03 Pulse oximeter probe SPO2: 99% Pulse: 72 . . .*
> *. . . at 02:32 Blood pressure taken 120 / 80 MAP (93) . . .*
> Update vitals at 03:41 . . .
>   Temp: 41.3°C
>   Resp: 12
> . . . end vitals update
> *. . . at 03:52 Temperature taken: 41.3°C . . .*
> *. . . at 03:55 Respiration rate taken: 12 . . .*
> Comment: at 04:24 This is a comment.

The monitor behaves in a realistic manner. When the blood pressure function is activated the learner hears the pump and sees the cuff inflate to approximately 10mmHg over the systolic. The inflation of the cuff is the same interval in seconds regardless of the pressure (the numbers just indicate that it inflates faster). The cuff then deflates at a standard rate (2mmHg/sec) until 10mmHg below the diastolic. The BP is then displayed along with the Mean Arterial Pressure (MAP).

When the SPO2 is activated it pulses at the correct rate for the pulse of the patient and takes several seconds to register the O2 level and pulse. The pulse oximeter probe remains attached to the patient unless the learner presses the button again to remove it. The SPO2 varies in a manner similar to an actual reading; usually about ±1% for values above 80% saturation and ±2% for values below 80%.

Pressing the ECG button, if enabled, connects the leads to the patient and displays a synthetic waveform for Lead lll. If the respiration rate button is pressed while the ECG is enabled *and* connected, the respiration rate (derived from the ECG signal) is displayed.

The monitor is controlled by a technologist or a simulationist from the Control Window. Vitals are displayed in the window and values can be entered or changed. Most vitals can accept four values. These are:

Start: This is the beginning value, depending on the vital sign. It will be available to the learner immediately when the "Update" button is activated.

End: This is the value at the end of the time period (Delay + Trend). If the Trend is 0 then the End value and Delay value will not be used. The vital sign will be set to the Start value only.

Delay: This is the time, in seconds, that the Start value will not change. This can be 0 or larger.

Trend: This is the time, in seconds, that the Start value will begin to trend towards the End value. The change is linear. This can be 0 or larger. **Note*: If the Trend is set to 0, the End value and Delay value are ignored. Only the Start value is set when updated if the Trend is set to 0.*

The ECG is slightly different. Only a Trend time is available which switches the waveform at the end of the trend period. If the Trend is set to 0, only the start value of the waveform is sent when updated.

The blood pressure has one Delay and Trend setting for both the Systolic and Diastolic pressures; this makes sense and keeps things simpler. The Delay and Trend are **set** *and* **displayed** on the Systolic line; these values are only **displayed** on the Diastolic line.

If the ECG is disabled, the Respiration Rate is available by clicking on the "Respiration" button. If the ECG is enabled, then the Respiration Rate is only available if the ECG is attached (activated) since the Respiration Rate is derived from the ECG in this case. Please note that the "RR" on the monitor stands for "Respiration Rate" and **not** the RR interval of the ECG.

After entering the values in the table, use the "Update" button to send the values to the learner's monitor. You must check the box(s) for all the values you wish to send. Use the "Select/Deselect All" if you wish to send all the values at once. The time you updated the monitor and the values sent will be displayed in the control window.

The "Load" button is used to populate the controls with a set of values stored in a text file. This button opens a window on your local computer and lets you search for and load a set of values. This saves time and lets you program scenarios ahead of time.

The "Toggle C/F" lets you change the default temperature unit from Celsius to Fahrenheit and back again. If the temperature is changed from Celsius to Fahrenheit, or vise versa, during a scenario the temperature window will automatically be blanked, and the new temperature unit will be displayed. The program will not convert values between the two systems. The default temperature unit is Celsius; this can be changed by using the setTemperature() JavaScript function to set the initial temperature and unit. This should be added to the Vitals Monitor web page as an onload in the <body> tag; e.g. setTemperature(98.6, false);

At the top of the Control Window is the response window. When you set the vitals, it is recorded in this window with a time stamp. When the learner takes a vital sign, the time and value of the sign are recorded. You can also add comments (immediately below this window) which are time stamped. You can use the information in this window while debriefing.

The peer-to-peer connection process
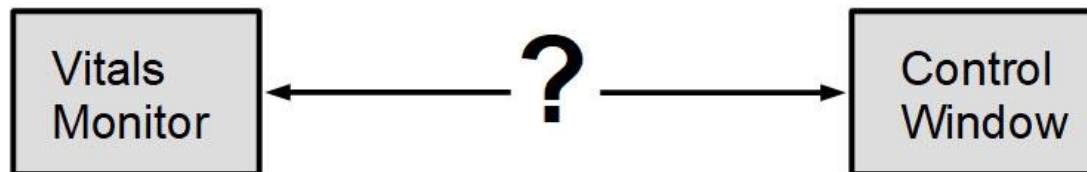
When the Vitals Monitor and the Control Window are first opened, both are independent and are not connected. Usually when two web pages interact, in a chat for example, the information is sent to a server then forwarded from the server to the other web page. For a direct peer-to-peer connection both pages need to know the address of the other so that communication between the two can take place. This requires a *temporary* connection to a server.
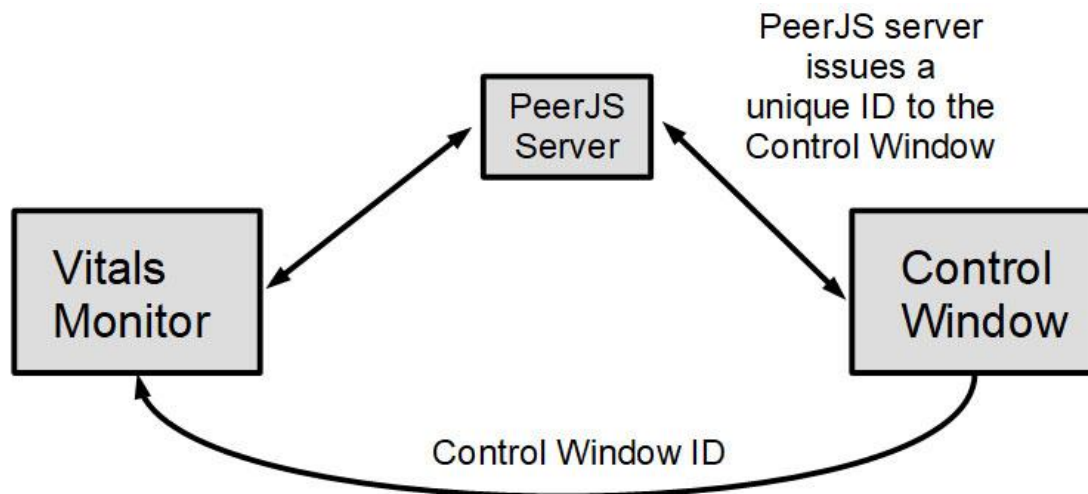
In order to connect the Vitals Monitor to the Control Window, a server must be used temporarily, to mediate the connection. When the Control Window connects to the PeerJS server, the server issues a unique ID to the Control Window which identifies it. This ID is included in a link sent to the learner by any convenient means (email, text, chat etc.). The learner clicks on the link, the Vitals Monitor opens, and it connects with the server. The server then mediates a connection between the Vitals Monitor and the Control Window, then disconnects. No data is sent through the PeerJS server. The two windows now have a direct peer-to-peer connection.
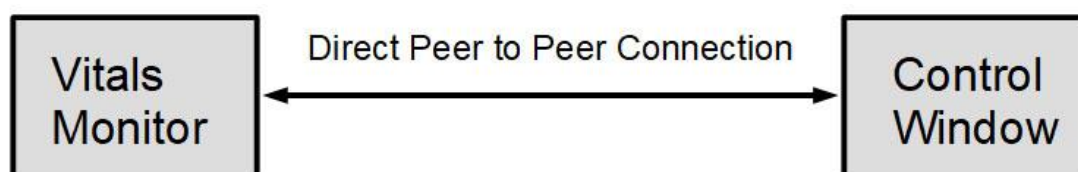
Programming Scenarios

Scenarios can be programmed ahead of time by creating text files which load values for each of the vital signs into the Control Window. A description of the file is displayed in the response window. Each file consists of two lines. The first is a description of what the settings are and the second is a series of name/value pairs. Each pair is separated by a "&" and the pair is in the format "name=value". The first character on the line of the name/value pairs is a question mark "?". When loaded, the values populate the appropriate fields in the lower part of the Control Window. The values do not automatically update so the operator can change fields before updating if necessary. An example is given below.

> **Decreasing respirations from drug administration**
> **?rsps=16&rspe=6&rspd=15&rspt=240**

The above will set a starting rate of 16 respirations per minute, and end rate of 6 respirations per minute. The 16 will be set immediately; it will remain steady at this rate for 15 seconds and then decrease at a linear rate to 6 over a period of 240 seconds (4 minutes).

If there is a problem with the file, the values will not be loaded, and the problem will be noted in the response window.

> **?ecgn='false'&ecgs='NSR'&egce='NSR'&egct=0**
>
> **?syss=120&syse=0&sysd=0&syst=0&dias=80&diae=0**
>
> **?spos=99&spoe=0&spod=0&spot=0**
>
> **?plss=72&plse=0&plsd=0&plst=0**
>
> **?tmps=36.9&tmpe=0&tmpd=0&tmpt=0&cels='true'**
>
> **?rsps=16&rspe=0&rspd=0&rspt=0**

The **minimum** that can be sent is one complete line above; the **maximum** is all the lines above concatenated into a single line without spaces and starting with a single question mark; no other question marks may be in the single line.

ecg stands for the ECG waveform, sys is for systolic blood pressure, dia for diastolic blood pressure, spo for oxygen saturation, pls for pulse, tmp for temperature, and rsp for respirations. The final letter stands for s start value, e end value, d for delay, t for trend, and n for ECG enable. cels is for Celsius and is true

for Celsius and false for Fahrenheit. For the blood pressure only the systolic has a delay and trend since the systolic and diastolic use the same delay and trend times. Checkboxes will be automatically checked for each parameter, but the user must manually update the monitor. These values can be changed after loading.

Using the Vitals Monitor in the stand-alone mode

Although it is intended that the Vitals Monitor and the Control Window be used together (connected mode), the Vitals Monitor can be used by itself in the stand-alone mode. When used in this manner, the vital signs are fixed and cannot be updated over time. This might be useful in an online scenario where a standardized patient is having a routine assessment and only one set of vital signs is needed during the scenario. There is also no time-stamped record of the actions that the learner has performed.

To set the vital signs initial conditions open the Vitals Monitor webpage, **saVSMonitor.html**, using a text editor such as Windows Notepad. You will have to host the webpage on your own website or server since you cannot modify the webpage on someone else's server. Scroll down until you see the following:

```
</head><body onload="
    bpSound = loadSound('./Sounds/pump.mp3');
    initialize();
    setTimeout(function(){conectToPeer(); }, 5000);
    // use the functions below to set initial conditions
    setECG(false);
    setBloodPressure(120, 80);
    setTemperature(true, 37.0);
    setRespirationRate(16);
    setPulseOximetry(99);
    setHeartRate(72);">
```

Only modify the functions beginning with "set" (e.g. setBloodPressure()); <u>**DO NOT**</u> **modify or delete the first three functions or you will disable the webpage!** As of this writing, the ECG function is **not** available so leave this set to false. setBloodPressure() has two variables, the systolic and diastolic pressures. You can change these as needed. The setTemperature() function has two variables; the first is the temperature unit (in this case true = Celsius) and the second is the temperature (in this case 37.0). If you wish to use Fahrenheit, set the first variable to false and the temperature in Fahrenheit, setTemperature(false, 98.6) . Please note that the program will not convert temperatures from one unit to another. The three remaining vial signs, setRespirationRate(), setPulseOximetry(), and setHeartRate() each have only one variable. The maximum value for setPulseOximetry() is 100 percent.

It is possible to set trends and delays using the onload functions, but the time begins the instant the learner starts the monitor, so the functionality is very limited. The following shows a delay of 180 seconds (3 minutes) and a trend of 300 seconds (5 minutes) applied to the vitals.

```
    setECG(false);
    setBloodPressure(120, 80, 90, 50, 180, 300);
    setTemperature(true, 37.0, 41.3, 180, 300);
    setRespirationRate(14, 19, 180, 300);
    setPulseOximetry(99, 96, 180, 300);
    setHeartRate(72, 60, 180, 300);
```

In the above, the blood pressure decreases from 120/80 to 90/50, the temperature, in Celsius, increases from 37.0°C to 41.3°C, the respiration rate increases from 14 to 19, the oxygen saturation decreases from 99% to 96%, and the heart rate decreases from 72bpm to 60bpm.

Running the Virtual Vital Signs Monitor on your server

Very easy. Just upload the following files and create the Sounds directory in your main web directory:
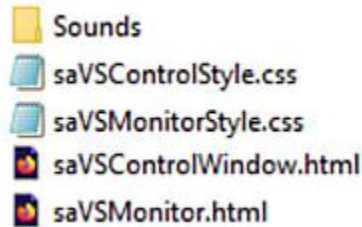
Inside the Sounds folder is the blood pressure pump sound, pump.mp3. If you want to change the location of the sound you will have to modify the line of code in **saVSMonitor.html** that is found immediately under the <body> tag (highlighted in yellow below):

```
</head><body onload="
bpSound = loadSound('./Sounds/pump.mp3');
initialize();
setTimeout(function(){conectToPeer(); }, 5000);
```

This line indicates the location of pump.mp3 relative to the directory where the vital monitor files are stored. Change as necessary.

If you will be using this often, you may wish to host a copy of the peerjs library on your website and possibly run a PeerServer. If you do this you will have to change the <script> line for the peerjs library in **saVSControlWindow.html** <u>and</u> **saVSMonitor.html** which is shown below.

```
<script src="https://unpkg.com/peerjs@1.3.1/dist/peerjs.min.js"></script>
```

You can donate to peerjs to help keep the public server running. For more about peerjs see:
https://peerjs.com/