Proceedings of the National Conference
On Undergraduate Research (NCUR) 2018
University of Central Oklahoma
Edmond, Oklahoma, USA
April 5-7, 2018

Developing a Real-Time Patient Monitor Data Import System

Mohamed Hozayen, Shermeen Nizami, Amente Bekele, and Kevin Dick
Department of Systems and Computer Engineering
Carleton University
Ottawa, ON, K1S 5B6, Canada

Faculty Advisor: Dr. James R. Green

Abstract

This paper develops platform agnostic, user-friendly, open source software applications *Patient Monitor Data Importer* (PMDI) and *parserPMDI* to import physiologic data from patient monitors in real-time. These applications form part of a larger research project that is investigating the novel use of the pressure-sensitive mat (PSM) technology for continuous patient monitoring in the neonatal intensive care unit (NICU). To evaluate the potential for PSM for respiration and patient movement detection, gold standard physiologic data are required. This research is being conducted at the Children's Hospital of Eastern Ontario (CHEO), Ottawa, Canada, where physiologic data are acquired using the Infinity Delta patient monitor series (Dräger Medical, Germany). Currently, one must purchase proprietary data logging software to acquire data from these monitors. The PMDI and *parserPMDI* have been developed as open source applications in our lab to (a) import multiple physiologic data streams from patient monitors; (b) import alarm data associated with each physiologic data stream; (c) display data parameters and acquisition status in a graphical user interface; (d) export the data in decimal format to a CSV file; (e) be robust to temporary cable disconnections; and (f) facilitate data logging from a variety of patient monitors through modular expansion of the open source code. These interoperable applications programmed in Java communicate with the monitor to import physiologic data in hexadecimal messages. The message is then parsed into a decimal table and stored as a CSV file. Currently, these two applications enable researchers to collect and parse physiologic data from the Dräger Infinity Delta patient monitors. However, their modular design facilitates retargeting to import data from monitoring equipment from other vendors. The current PMDI imports and parses discrete parameters such as heart and pulse rates and discrete alarm events. The Java application source code has been released as open source enabling other researchers to use them for real-time patient data acquisition and analysis.

Keywords: real-time, physiologic data import, application

1. Introduction

Research on patient monitoring devices requires access to physiologic data acquired by the monitors. Proprietary data logging software are available for purchase from independent vendors to acquire data from these monitors. This research develops two interoperable, modular, open source Java applications (apps), *Patient Monitor Data Import* (PMDI) and *parserPMDI*, that can import, parse and store physiologic monitor data. This paper also presents a variety of tests of these apps in the lab and in a real-time hospital setting. Both apps are currently in use in a large research project which is evaluating the use of the pressure sensitive mat (PSM) technology as a continuous patient monitoring modality. The apps are deployed on a laptop at the patient bedside in a novel multi-modal data acquisition environment in the neonatal intensive care unit (NICU) at the Children's Hospital of Eastern Ontario (CHEO), Ottawa, Canada. This research has been approved by the Research Ethics Board at CHEO.

Figure 1 depicts the data collection framework on the right and a real patient environment on the left. Using four different modalities, we simultaneously collect PSM data, annotated patient events at the patient bedside, physiologic and associated alarm data from patient monitors, and video recordings of patient movement, routine care and clinical interventions. This rich multidimensional dataset is used to train and test machine learning algorithms for automated PSM- and video-based estimation. Contact pressure data from the PSM is used to detect and estimate the following parameters in neonates: (a) heart rate (HR), (b) respiratory rate (RR), (c) patient movements and limb detection, and (d) patient-related events including routine care and clinical interventions [1]. Video recordings is also used to estimate these parameters. Both the PSM and video are contactless technologies. The data collection framework on the right in Figure 1 also depicts Clinical Event Annotations (CEA). CEA is an mHealth app that is developed for use by a bedside researcher or nurse to record annotations of patient related events in real-time [2]. These annotations are used as gold standard data for research validation purposes.

Physiologic patient data is confounded by several factors including electrical, mechanical, environmental and physiologic artifacts [3]. In particular, motion artifact arising from patient movement is a well-known source of interference in physiologic data streams such as blood oxygen saturation (SpO_2) measured by pulse oximeters [4,5]. Patient movements may be self-initiated or externally-induced through routine care or clinical interventions. Motion artifacts can cause patient monitors to trigger clinically irrelevant alarms. False alarms increase the noise and alarm fatigue in the clinical unit and desensitize clinicians to patient alarms [6]. Existing patient monitors have no means of continuously detecting patient movement unobtrusively in a contactless, non-invasive fashion. PSM is a non-contact, non-invasive technology that can be unobtrusively placed beneath the patient's bedsheet. The research hypothesis is that PSM technology can be used for continuous detection of patient movement, where PSM data will be used to identify the periods of patient movement. There is no known method that applies data fusion between PSM data and other physiologic data streams acquired from patient monitors to reduce the impact of motion artifacts on alarms generated by the patient monitors. The overarching goal of the research project is to detect motion artifacts using PSM data and fuse this information with the physiologic data streams acquired from patient monitors to reduce the number of false alarms generated by the patient monitors [7]. Thereby, reducing the false alarm rates on patient monitors, alarm fatigue, noise, and staff desensitization in patient monitoring environments. This, in turn, bears the potential to enhance the quality of care and safety at the patient bedside.

The main contribution of this paper is the development of the custom *Patient Monitor Data Import* (PMDI) and *parserPMDI* application software. PMDI forms an essential part of the larger project described above. It provides the following functionality to the project: (a) imports multiple physiologic data streams from patient monitors; (b) imports alarm data associated with each physiologic data stream; (c) displays data parameters and acquisition status in a graphical user interface (GUI); (d) exports the data in decimal format to a CSV file; (e) is robust to temporary cable disconnections; and (f) facilitates data logging from a variety of patient monitors through modular expansion of the open source code. In this research, the PMDI app is used to acquire multiple physiologic data streams from the Dräger patient monitor at the (local hospital name) NICU through the RS232 interface at 1 Hz. The patient monitor data acquired and saved by the PMDI app are currently being used in the PSM- and video-based estimation research that

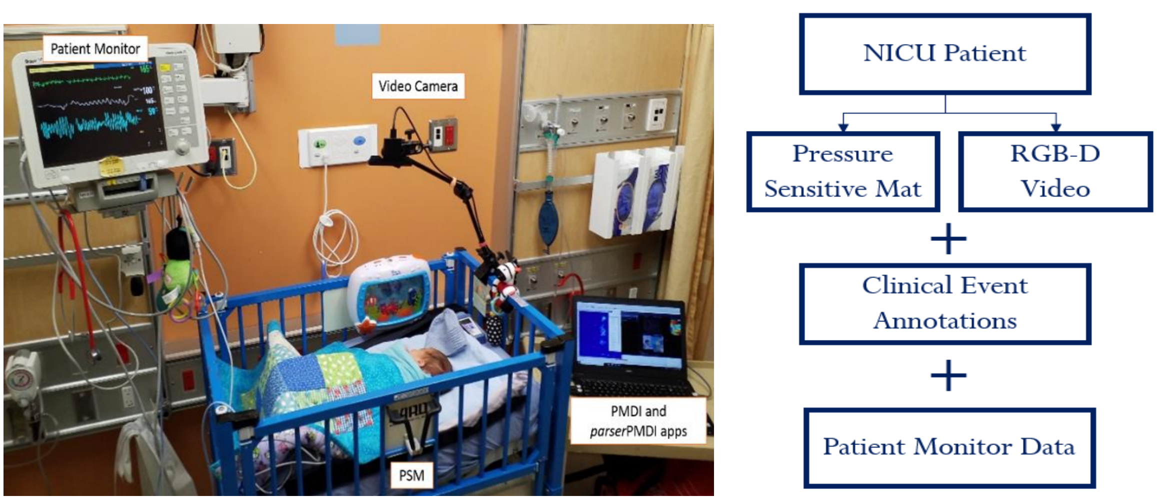

Figure 1. Left: Multi-modal patient monitoring and data acquisition environment set up at the CHEO NICU.
Right: Data collection framework

is described above. In the long term, these data will be fused with PSM data to reduce the false alarm rates of the patient monitors.

2. Methods

This section describes software development of the PMDI and *parserPMDI* applications. The PMDI application programmed in Java exchanges hexadecimal messages with the patient monitor to import physiologic data. The *parserPMDI* applications then parses the hexadecimal message to convert it to a user-friendly decimal format which it then stores as a CSV file. Following that is a description of the hardware connections that are required to import the physiologic data from the Infinity Delta patient monitor series by Dräger Medical. The section concludes with a a testing session of the apps in the lab.

2.1. Software Design and Development

The PMDI application is designed using software engineering principles including separation of concerns, modularity, anticipation of change, and incrementality. The PMDI application is installed on a laptop for use at the patient bedside. It imports and stores patient data locally on the laptop in real-time. These data are used as gold standard for research, thus requiring high fidelity and integrity. The PMDI app must be reliable to ensure there is no data loss or failure during recording. It is required to be robust in handling software data import errors or hardware disconnection errors. It must report to the user any such errors by generating alerts in real-time.

The PMDI system was developed as a concurrent system for real-time environments. The application employs threads wherein each thread handles one task. There was one thread dedicated to each of the following: data import, display of real-time values, control of the communication window, and continuous checking for hardware and software failures. To avoid deadlocks, shared global variables among threads were only being used for reading. In the event of a software or hardware failure the application icon starts to flash red in the taskbar to alert the user.

The PMDI app has a wide variety of users ranging from researchers, engineers to clinicians. Thus, a user-friendly and intuitive interface is paramount. The PMDI graphical user interface (GUI) consists of multiple sub-screens that display the following information. A sub-screen to display the decimal data values and their corresponding alarm status being imported in real-time. A communication window that displays text-based messages to the user, including messages during run-time and error messages. A set of buttons to interface with the background processes to (a) select a communication port on the laptop, (b) to open the port, (c) test the port connection, (d) refresh the ports, (e) select the directory to save the data file, (f) start the recording, and (g) view or verify the current directory in use during the recording. Clicking each button generates a text message in the communication window. The user could therefore test the connection before starting a session, select the directory where the data file is to be committed, and observe the communication window for all messages in real-time. The information displayed by the sub-screens indicate whether the system is running smoothly or has encountered a problem. Figure 2 shows the GUI of the PMDI with all the sub-screens. Note that the data display sub-screen containing HR, SpO_2, PLs (or PR), and RR is blank as the communication port on the laptop is not yet opened and requires hardware connectivity as explained below.

Each patient recording is named by the date and time at the recording onset and is therefore unique to that session. Also, each message received within the session is timestamped. The *parserPMDI* application is built using definitions of data variables specified in the Infinity Delta export protocol [8]. This app parses the hexadecimal message to convert it to a user-friendly decimal format which is then stored as a CSV file on the hard drive.

2.2. Hardware Connectivity

The Infinity Delta patient monitor comes with an RS232 data logging port labelled as X3 on a docking station number 5206110. Note that port X3 is a requirement for digital data import from the monitor. Other docking stations which do not have port X3 do not support digital data import. The second hardware requirement is a proprietary X3 cable manufactured by Dräger. One end of the X3 cable connects to the X3 port and the other end to a RS232-USB converter. The converter is then connected to the USB communication port on the laptop. The monitor is configured to send back the patient monitor data if and only if the hexadecimal message "0x00 0xA5 0x02 0x00 0x77 0x1E" is sent through the X3 cable to port X3 in the docking station. Once the hardware is connected and the PMDI app is launched, the communication window displays if the connection is successful.

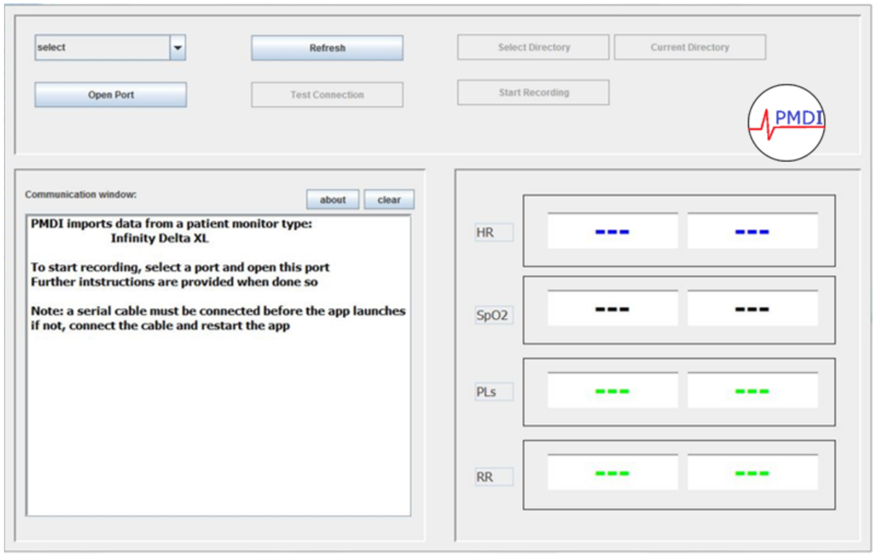

Figure 2: PMDI GUI with its logo on the top right

When the user starts importing data, there are input and output streams in the PMDI backend that are responsible to send the hexadecimal messages and receive the patient monitor data. The PMDI sends the hexadecimal message once every second (1 Hz) and saves it to the hard-drive upon reception of the response. This leads to a high CPU consumption as the PMDI is continuously using the laptop system bus to transfer data from cached memory to the hard-drive. However, this leads to high-fidelity by avoiding data loss in case of any disruption that might result in the system or the laptop disconnecting or powering down.

2.3. Testing

A real-time test study was run in the lab. Figure 3 shows the set up of the test study. In this figure, the Infinity Delta XL patient monitor is shown on the left, and the PMDI GUI on the laptop is shown on the right. Physiologic data imported in real-time by the PMDI is identical to that shown on the patient monitor. This validates the run-time operation of the app. Furthermore, the communication window in Figure 3 displays the messages exchanged between the PMDI app and the monitor.

Figure 3. Left: Infinity Delta XL patient monitor; Right: Real-time test of PMDI showing data imported from the monitor

Figure 4 shows a data packet from the test study received at 19:43:05 on April 01, 2018. Figure 4 also demonstrates how these data are parsed based on the Dräger export protocol wherein the IDs for physiologic data packets are given [8]. For example, 0x01 is the ID for heart rate data. Following that are the bytes of data containing the HR value. These bytes are highlighted in red in Figure 5. These data show that the HR at that time is 0x36 0x37 in hexadecimal which is 67 in decimal. The same idea is applied with respect to knowing other physiologic data packets IDs as follows:

- 0x60 is an ID for Respiratory Rate: 0x32 0x36 → 26 decimal (green highlight)
- 0x64 is an ID for SpO2: 0x39 0x38 → 98 decimal (blue highlight)
- 0x65 is an ID for Pulse Rate: 0x36 0x36 → 66 decimal (yellow highlight)

This data packet is parsed using the *parser*PMDI. Figure 5 depicts the parsed file that is stored by the app. The yellow highlighted row validates the real-time data that were imported as shown in Figure 3.

```
2018/04/01 19:43:05, Read data:
0xA5, 0xC6, 0x00, 0x77, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x09, 0x0E, 0x01, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x01, 0x00, 0x00, 0x36, 0x37, 0x00, 0x10, 0x02, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x03, 0x00,
0x00, 0x00, 0x04, 0x00, 0x80, 0x00, 0x18, 0x04, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xC1, 0x00,
0x80, 0x00, 0xC2, 0x00, 0x80, 0x00, 0xC3, 0x00, 0x80, 0x00, 0xC4, 0x00, 0x80, 0x00, 0x18, 0x04,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xC5, 0x00, 0x80, 0x00, 0xC6, 0x00, 0x80, 0x00, 0xC7, 0x00,
0x80, 0x00, 0xC8, 0x00, 0x80, 0x00, 0x14, 0x03, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xC9, 0x00,
0x80, 0x00, 0xCA, 0x00, 0x80, 0x00, 0xCB, 0x00, 0x80, 0x00, 0x14, 0x03, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0xD8, 0x00, 0x80, 0x00, 0xD9, 0x00, 0x80, 0x00, 0xDA, 0x00, 0x80, 0x00, 0x18, 0x04,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x21, 0xFE, 0x80, 0x00, 0x22, 0xFE, 0x80, 0x00, 0x23, 0xFE,
0x80, 0x00, 0x24, 0xFE, 0x80, 0x00, 0x0E, 0x01, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x60, 0x00,
0x00, 0x32, 0x36, 0x00, 0x14, 0x02, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x64, 0x00, 0x00, 0x39,
0x38, 0x00, 0x65, 0x00, 0x00, 0x36, 0x36, 0x00, 0xA5,
```

Figure 4. Data packet imported by the PMDI app from the patient monitor at 19:43:05 on April 01, 2018. Physiologic data packet IDs are highlighted for HR, RR, SpO_2, and PR

	A	B	C	D	E	F
1	Date/Time	HR (bpm)	RR (RPM)	SPO2 (%)	PLs (bpm)	
107	4/1/2018 19:44	68	42	99	66	
108	4/1/2018 19:44	68	37	99	66	
109	4/1/2018 19:44	68	37	99	66	
110	4/1/2018 19:44	67	37	99	66	
111	4/1/2018 19:44	67	37	99	66	
112	4/1/2018 19:44	65	28	99	66	
113	4/1/2018 19:44	65	28	99	66	
114	4/1/2018 19:44	65	28	99	66	
115	4/1/2018 19:44	64	28	99	65	
116	4/1/2018 19:45	64	22	99	65	
117	4/1/2018 19:45	62	22	99	65	
118	4/1/2018 19:45	62	22	99	65	
119	4/1/2018 19:45	62	24	99	65	
120	4/1/2018 19:45	62	24	99	62	
121	4/1/2018 19:45	63	24	99	62	
122	4/1/2018 19:45	63	24	99	62	
123	4/1/2018 19:45	62	19	99	62	
124	4/1/2018 19:45	62	19	99	62	

Figure 5. Final output after using the *parser*PMDI app

3. Results

This section presents results of using both apps to import and store real-time data at the patient bedside. To date, the PMDI app has been used to acquire 272.3 MB of physiologic data from a total of eleven patients in the NICU at CHEO. Mean recording period is four hours per patient, during which 24.75 MB of data are recorded on average. These data typically include the heart rate (HR) in beats per minute (bpm), respiratory rate in breaths per minute (brpm), arterial oxygen saturation (SpO_2) percentage and pulse rate in beats per minute (bpm). Each data stream is accompanied by its associated alarm status stream. The PMDI app is user-set to acquire multiple physiologic data streams from the Dräger patient monitor through its RS232 interface at 1 Hz. Table I presents results of patient data import using the PMDI in a single instance. Patient identifiers have been removed to preserve patient confidentiality. A packet of hexadecimal data, like that shown in Table I, is received every second throughout the patient recording. The PMDI stores the entire patient data set in a single CSV file.

Once the patient recording is complete, the CSV file is run through the *parserPMDI* application to parse all messages within the recording and convert those to decimal values that are then stored is a user-friendly CSV. A single packet of hexadecimal data, as shown in Table I, is converted to one row of the output file as shown in Table II. The same color coding is used to highlight the physiologic parameters as used in Figure 4. The parsed CSV file will consist of as many rows as there were packets of data received during a patient recording. Each row is timestamped.

4. Discussion

The PMDI and *parser*PMDI applications developed in this paper are a major contribution to our ongoing research project in the NICU. The data imported, parsed and stored by these two apps are used as the gold standard when validating the use of the PSM and video in continuous patient monitoring in the NICU. As demonstrated in this paper

Table 1: Anonymized patient data acquired by the PMDI app during a single instance or second of data import

0xA5	0x02	0x01	0x77	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00
0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x0B	0x0F	0x01	0x00	0x00	0x00	0x00	0x00	0x00
0x01	**0x00**	**0x00**	**0x31**	**0x34**	**0x35**	0x00	0x10	0x02	0x00	0x00	0x00	0x00	0x00	0x00	0x03
0x00	0x00	0x00	0x04	0x00	0x00	0x00	0x18	0x04	0x00	0x00	0x00	0x00	0x00	0x00	0xC1
0x00	0x00	0x00	0xC2	0x00	0x00	0x00	0xC3	0x00	0x00	0x00	0xC4	0x00	0x00	0x00	0x18
0x04	0x00	0x00	0x00	0x00	0x00	0x00	0xC5	0x00	0x00	0x00	0xC6	0x00	0x00	0x00	0xC7
0x00	0x00	0x00	0xC8	0x00	0x00	0x00	0x14	0x03	0x00	0x00	0x00	0x00	0x00	0x00	0xC9
0x00	0x00	0x00	0xCA	0x00	0x00	0x00	0xCB	0x00	0x00	0x00	0x14	0x03	0x00	0x00	0x00
0x00	0x00	0x00	0xD8	0x00	0x00	0x00	0xD9	0x00	0x00	0x00	0xDA	0x00	0x00	0x00	0x18
0x04	0x00	0x00	0x00	0x00	0x00	0x00	0x21	0xFE	0x00	0x00	0x22	0xFE	0x00	0x00	0x23
0xFE	0x00	0x00	0x24	0xFE	0x00	0x00	0x21	0x0B	0x00	0x00	0x00	0x00	0x00	0x00	0x5B
0x00	0x00	0x38	0x31	0x00	0x5C	0x00	0x00	0x35	0x32	0x00	0x5D	0x00	0x00	0x36	0x36
0x00	0xF2	0x00	0x01	0x11	0x27	0x1E	0x00	0x0E	0x01	0x00	0x00	0x00	0x00	0x00	0x00
0x60	**0x00**	**0x00**	**0x36**	**0x36**	0x00	0x15	0x02	0x00	0x00	0x00	0x00	0x00	0x00	**0x64**	**0x00**
0x00	**0x37**	**0x32**	0x00	**0x65**	**0x00**	**0x00**	**0x31**	**0x34**	**0x35**	0x00	0x19	0x03	0x00	0x00	0x00
0x00	0x00	0x00	0x96	0x00	0x00	0x33	0x34	0x00	0x66	0x00	0x00	0x36	0x31	0x00	0x95

Table 2.: Parsed data produced by the parserPMDI app for one second of data imported from the patient monitor

HR (bpm)	RR (brpm)	SpO_2 (%)	PR (bpm)
145	66	72	145

the software is robust, reliable and accurate having undergone several design iterations and test cycles. The apps are open source with an MIT license on GitHub at the following link: github.com/MohamedHozayen/PMDI/. This allows researchers to freely access, use and modify the apps to fit their data import requirements.

5. Future Work

The current version of PMDI imports streaming physiologic data and their associated alarm status, since this information is completely encoded in each data packet imported from the monitor. The current version of the GUI displays the physiologic data value in a sub-screen as shown in the test study. This sub-screen is already designed with a column for displaying the concurrent alarm status to the right of each physiologic variable as depicted by the dotted lines in Figure 3. Through future work we shall decode the alarm status bytes in the hexadecimal message to display the alarm status in real-time. Moreover, the PMDI app can be modified to import continuous analog data from patient monitors, such as the electrocardiograph (ECG) and photoplethysmography (PPG) waveform data. This would allow for a richer dataset to be imported and analysed. The hardware configuration would include a connection to the Infinity Delta XL monitor's port labelled as X10 and an analog to digital converter to store these data on the laptop. Finally, accommodating other patient monitor types shall make the PMDI more universal and usable. This would require further investigation into the different patient monitors used at hospitals and their proprietary export protocol.

6. Conclusion

The PMDI and *parser*PMDI applications are being already used successfully for importing and saving physiologic patient data in the NICU. The PMDI imports multiple physiologic data streams from patient monitors in real-time, imports alarm data associated with each physiologic data stream, displays data parameters and acquisition status in a graphical user interface, and saves the data in a decimal format to a user-friendly CSV file format. The PMDI forms an essential part of the larger project which is evaluating the use of the PSM technology as a continuous patient monitoring modality in the NICU at CHEO.

7. References

1. Bekele A, Nizami S, Dosso YS, et al. Real-time Neonatal Respiratory Rate Estimation using a Pressure-Sensitive Mat. In: Proc. of IEEE Int. Symp. Med. Meas. Appl. (MeMeA), Rome, Italy; 2018.
2. Nizami S, Basharat A, Shaukat A, et al. CEA: Clinical Event Annotator mHealth Application for Real-time Patient Monitoring. In: Proc. of 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), Honolulu, HI, USA. Honolulu, HI, USA; 2018.
3. Nizami S, Green JR, McGregor C. Implementation of Artifact Detection in Critical Care: A Methodological Review. IEEE Rev Biomed Eng. 2013;6:127-142. doi:10.1109/RBME.2013.2243724.
4. Nizami S, Greenwood K, Barrowman N, Harrold J. Performance Evaluation of New-Generation Pulse Oximeters in the NICU: Observational Study. Cardiovasc Eng Technol. 2015 ; 6(3):383-391. doi:10.1007/s13239-015-0229-7.
5. Clarke GWJ, Chan ADC, Adler A. Effects of motion artifact on the blood oxygen saturation estimate in pulse oximetry. In: Medical Measurements and Applications (MeMeA), 2014 IEEE International Symposium On. IEEE; 2014:1-4. doi:10.1109/MeMeA.2014.6860071.
6. Sowan AK, Tarriela AF, Gomez TM, Reed CC, Rapp KM. Nurses' Perceptions and Practices Toward

Clinical Alarms in a Transplant Cardiac Intensive Care Unit: Exploring Key Issues Leading to Alarm Fatigue. JMIR Hum factors. 2015;2(1):e3. doi:10.2196/humanfactors.4196.
7. Nizami S, Green JR, Giffen PR. Detecting quality of physiologic data using contact pressure data for alarm generation. US Patent filed 15927844, 2018.
8. Infinity Gateway VF5.0 Protocol Handbook, Dräger Medical Systems Inc Telford, U.S.A.; 2007.