

УДК 629.3.054.254

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ДЛЯ ИЗМЕРИТЕЛЯ КОНЦЕНТРАЦИИ УГЛЕКИСЛОГО ГАЗА В ВЫДЫХАЕМОМ ВОЗДУХЕ КАПНОМЕТРА «МИКОН» С ДАТЧИКОМ КОНЦЕНТРАЦИИ КИСЛОРОДА

Д. Е. Грошев, В. К. Макуха, С. В. Степанов

Новосибирский государственный технический университет

E-mail: makukha@epu.ref.nstu.ru

Для аппаратного комплекса, включающего измеритель концентрации углекислого газа и кислорода в выдыхаемом воздухе, разработана программная оболочка. В результате врачу-исследователю предоставляется возможность наблюдения в реальном масштабе динамики потребления кислорода и выделения углекислого газа, анализа и хранения полученных зависимостей. Программное обеспечение протестировано в среде MS Windows 98/2000/XP.

Проблема формирования показателя, характеризующего состояние пациента, наиболее остро стоит в медицине критических состояний и, в частности, в анестезиологии, где необходимы экспресс-методы оценки состояния пациента [1].

В анестезиологическом мониторинге наибольший интерес при анализе состава дыхательной газовой смеси представляет определение содержания CO_2 в выдыхаемом воздухе.

Метод капнометрии позволяет в реальном масштабе времени вести измерение величины парциального давления CO_2 (или значение объемной концентрации), в том числе и в момент окончания выдоха — FET CO_2 . Значение содержания CO_2 в выдыхаемом воздухе в этот момент времени соответствует величине $P \text{ CO}_2$ (напряжение углекислого газа) артериальной крови [2].

Важную диагностическую информацию дает анализ текущего значения концентрации CO_2 во времени (капнограммы) в реальном масштабе времени.

Одновременное измерение концентрации CO_2 и O_2 в выдыхаемом воздухе позволяет определить такой параметр, как дыхательный коэффициент. Опираясь на этот параметр, можно оценить состояние организма в целом, а также определить уровень метаболизма.

Капнометр МИКОН (рис. 1) с датчиком кислорода (далее МИКОН) измеряет концентрации CO_2 и O_2 в выдыхаемом воздухе. Для измерения концентрации CO_2 используют метод инфракрасной спектроскопии [3], концентрация кислорода измеряется с помощью электрохимического датчика ДК-21 производства ООО «ИНСОВТ» (г. Санкт-Петербург). Выдыхаемый воздух по газопроводя-

щей трубке попадает к датчикам CO_2 и O_2 , рис. 2. Полученные данные передаются по интерфейсу RS-232 в компьютер, а также отображаются на сегментных индикаторах. Выбор интерфейса RS-232 позволяет подсоединять МИКОН ко всем компьютерам, в том числе и старых моделей.

Для организации взаимодействия прибора с персональным компьютером потребовалось разработать программное обеспечение, которое позволило бы визуализировать получаемые данные, сохранять результаты измерения в базу данных, вести картотеку пациентов, а также просматривать сохраненные результаты.

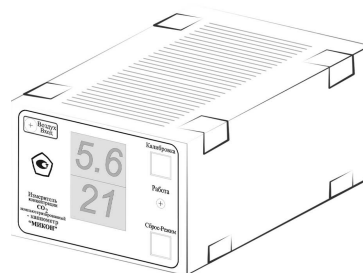


Рис. 1. Внешний вид капнометра МИКОН

Созданная программа автоматически определяет наличие устройства, его модель, и номер порта к которому оно подключено. Для работы с COM портом был выбран компонент TMS Asyn32 от фирмы TMS Software [4].

Программа поддерживает четыре режима работы:

- « $\text{CO}_2 + \text{O}_2$ »;
- « CO_2 »;
- « O_2 »;

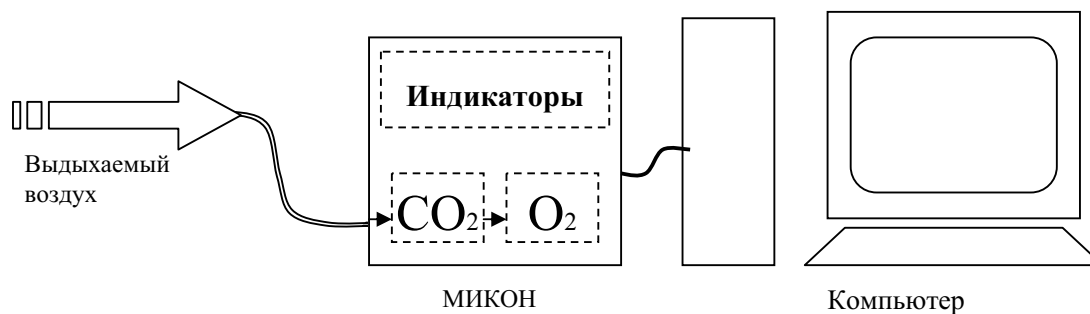


Рис. 2. Общая схема измерений

– «Капнометрия» (отображение капнограммы и графика трендов).

В зависимости от режима программа конфигурирует МИКОН на передачу тех или иных данных пересылкой команд управления. Данные концентрации CO_2 с капнометра приходят каждые 50 мс. Данные O_2 приходят с интервалом 1 с.

После выбора режима в основном меню программа открывает окно выбора пациента из базы данных. По окончании выбора пациента программа автоматически открывает окно измерений и запускает процесс измерения.

Окно измерения содержит три области:

- графики;
- цифровые данные;
- табличные данные по параметрам.

В режиме « $\text{CO}_2 + \text{O}_2$ » отображается два графика. В области измерения отображается график зависимости концентрации CO_2 от времени и справа приводятся цифровые данные по текущей концентрации CO_2 . Масштаб по оси времени определяется установками. Вид информационного окна в режиме « $\text{CO}_2 + \text{O}_2$ » показан на рис. 3.

Аналогичную структуру имеет область измерения O_2 .

Внизу расположена третья область, содержащая табличные данные по параметрам, соответствующим всему времени измерения.

Отображение данных на графиках « CO_2 » и « O_2 » происходит по кольцу, т.е. когда график подходит до края, отображение продолжится с начала, старый график затирается по мере необходимости.

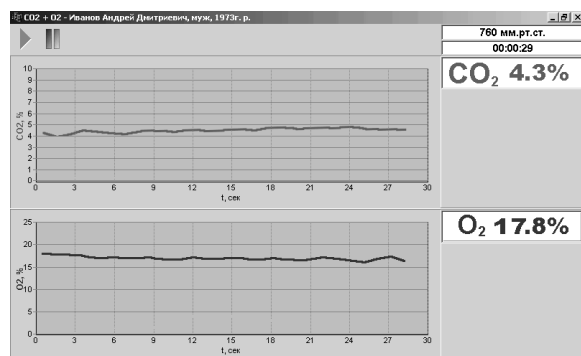


Рис. 3. Вид рабочего окна при измерении концентрации углекислого газа и кислорода из смесительного объема

После остановки измерения данные на графиках « CO_2 » и « O_2 » разворачиваются и отображаются таким образом, что данные, полученные последними, отображаются справа, а данные, полученные первыми – слева.

Результаты измерений после остановки могут быть занесены в базу данных.

Измерения в режимах « CO_2 » и « O_2 » аналогичны измерению в режиме « $\text{CO}_2 + \text{O}_2$ ». Отличия заключается в том, что область измерения содержит не два графика, а один.

В режиме «Капнометрия» окно программы имеет два графика: график концентрации углекислого газа и график трендов. График концентрации CO_2 отображается по кольцу. Для графика трендов при достижении максимального значения на временной оси запрограммировано автомасштабирование, что позволяет проследить тенденцию дыхания на длительном промежутке времени.

В таблице общих параметров кроме данных для CO_2 и FET CO_2 отображаются параметры дыхания: интервал дыхания и частота дыхания.

Результатом работы программы являются, полученные за интервал измерения минимальные, максимальные и средние значения концентраций CO_2 , O_2 или FET CO_2 и параметры дыхания в режиме «Капнометрия».

Отображение графиков на экране осуществляется стандартными средствами C++ Builder с помощью компонента TChart [5].

Программа позволяет вести накопление данных за определенный промежуток времени, выбранный пользователем. Интервал накопления для каждого из режимов задается пользователем в настройках.

После остановки измерения во всех режимах программа переходит в режим работы с маркерами. Перемещение маркеров осуществляется с помощью манипулятора «мышь». При этом на графиках отображаются по две вертикальные сплошные линии. Над каждым маркером находится выноска, показывающая положение маркера на оси абсцисс.

В правой части экрана располагается информация:

- значений измеряемых величин в моменты времени, соответствующих положениям маркеров;
- среднее, минимальное и максимальное значения измеряемой величины за интервал времени, заключенный между маркерами.

Для хранения результатов измерений выбрана база данных Microsoft Access. Программа позволяет сохранять графики проводимых измерений и просматривать полученные результаты.

Хранение результатов измерений в программе осуществляется с помощью контейнеров. Поиск данных и вычисление средних значений организовано через использование библиотек STL C++.

Основную сложность при разработке программы вызвал режим « $\text{CO}_2 + \text{O}_2$ » за счет необходимости постоянного переключения между режимами передачи данных CO_2 и O_2 , синхронизации процессов инициализации МИКОНа и получения данных с прибора. Проблему удалось решить после анализа временных интервалов при работе с COM-портом в среде Windows за счет создания дополнительных задержек при переключении режимов.

Заключение

Разработано программное обеспечение измерителя концентрации углекислого газа и кислорода в выдыхаемом воздухе, работоспособное в среде MS

Windows 98/2000/XP. Была организована работа с оборудованием в реальном масштабе времени.

Реализован алгоритм определения параметров дыхания.

Программа позволяет сохранять результаты измерений и графики полученных зависимостей в базе данных с последующим просмотром.

СПИСОК ЛИТЕРАТУРЫ

1. Калакутский Л.И. Система интегрального анестезиологического мониторинга. — Самара. <http://www.eliman.ru/publ/iy-un98.html>
2. Калакутский Л.И., Манелис Э.Л. Аппаратура и методы клинического мониторинга. — Самара: Самар. гос. аэрокосм. ун-т, 1999. — 160 с.
3. Грошев Д.Е., Макуха В.К., Сокол А.В. Мониторинговый измеритель концентрации углекислого газа в выдыхаемом воздухе «МИКОН» // В сб.: Медицинская техника в Сибири. — Новосибирск: СибНИИЦМТ, 2001. — С. 40–41.
4. TMS Software. <http://www.tmssoftware.com>
5. Архангельский А.Я. Программирование в C++ Builder 5. — М.: Бинум, 2001. — 1152 с.

УДК 681.3.06

ИСПОЛЬЗОВАНИЕ НАБОРА ДИАГРАММ UML ДЛЯ ПОСТРОЕНИЯ МОДЕЛЕЙ ПРОИЗВОДИТЕЛЬНОСТИ

С.А. Дубаков, В.А. Силич

Томский политехнический университет
E-mail: sad@osu.cctpu.edu.ru, vas@osu.cctpu.edu.ru

Рассматривается возможность генерации моделей производительности программного обеспечения на основе диаграмм в нотации UML как одна из базовых составляющих методологии интеграции анализа производительности в процесс разработки. Предложен подход, основанный на методологии Software Performance Engineering (SPE), использующий в качестве исходных данных стандартные элементы UML и ряд расширений.

Основной причиной выявляемых проблем производительности программного обеспечения (ПО) является инерционный подход к данной проблеме на протяжении процесса разработки. Данный подход, именуемый также в литературе как «fix-it-later» («исправим-это-потом»), заключается в том, что вопросы производительности игнорируются до того момента, когда выявляются проблемы. В случае если это происходит (а происходит это практически всегда, за редким исключением), приходится принимать решение об увеличении аппаратных ресурсов, модифицировании исходного кода или перепроектировании системы или ее частей. Подобные затраты часто приводят к срывам сроков и превышению бюджета, а в некоторых случаях достижение требуемой производительности в рамках текущего проекта становится практически невозможным.

Проблемы с производительностью в большинстве случаев возникают вследствие неверных архитектурных решений, а не неэффективного кодирования. Это означает, что они привносятся в проект на ранних этапах разработки, но оказываются невыявленными до момента интеграционного тестирования или передачи системы в эксплуатацию, когда решить их становится гораздо сложнее.

Альтернативу рассмотренному типичному подходу составляет превентивный подход к вопросу производительности программного обеспечения. Это означает, что производительность рассматрива-

ется и учитывается на всех этапах жизненного цикла наряду с остальными ключевыми характеристиками программного обеспечения. Все важные проектные и административные решения должны приниматься в том числе с учетом вопросов производительности. Оценка производительности должна выполняться тем или иным способом при рассмотрении критических архитектурных альтернатив, а менеджмент должен учитывать затраты на рассмотрение и решение вопросов производительности при выделении объемов работ и составлении планов.

Одним из наиболее критичных вопросов при интеграции анализа производительности в процесс разработки программного обеспечения является возможность использования уже полученных артефактов разработки в качестве исходных данных для анализа.

В настоящее время, когда объектно-ориентированный подход стал использоваться повсеместно при проектировании и реализации сложных программных систем, стандартом моделирования де-факто стал Unified Modeling Language (UML). Появившись в 1997 г., UML объединил в себе все лучшее, что было наработано в области моделирования объектно-ориентированного программного обеспечения. UML — открытый стандарт, контролируемый некоммерческим консорциумом OMG (Object Management Group) [1].

Таким образом, очевидно, что для того, чтобы как можно более тесно интегрировать анализ производи-