

рассмотренных алгоритмах сводится к единице или двум, поиск ведется локально от точки к точке, селекция проводится по специальным правилам, а «память» алгоритмов ограничивается лишь последними лучшими решениями.

С другой стороны, используя такие механизмы, как «план охлаждения» в методе отжига, или аналогичные эвристики в пороговом и потоковом алгоритмах, методе рекордных оценок можно более эффективно управлять скоростью сходимости процесса поиска. Однако эти процедуры не обладают свойством параллелизма, который, наоборот, является отличительной чертой эволюционных алгоритмов.

Данный анализ эвристических программ для решения оптимизационных задач проводился в рамках разработки образовательного контента по дисциплине «Методы оптимизации».

**Е.С. Синютин, Ю.И. Волочков**

### **ПРИМЕНЕНИЕ ОПЕРАЦИОННЫХ СИСТЕМ РЕАЛЬНОГО ВРЕМЕНИ ДЛЯ ПРОЕКТИРОВАНИЯ МИКРОПРОГРАММНОГО ОБЕСПЕЧЕНИЯ АВТОНОМНЫХ КАРДИОГРАФОВ И МОБИЛЬНЫХ ПОЛИГРАФОВ**

Автономные кардиографы, такие как, например, монитор Холтера, раньше имели только одно назначение – длительная регистрация одного, двух или трех каналов ЭКГ и запись на твердотельный или магнитный носитель. От постановки до считывания пользователь мог иметь только одну возможность взаимодействия с прибором – нажатие на тревожную кнопку. Факт и время нажатия регистрировались и сохранялись на носителе. Нажатие на тревожную кнопку могло означать все что угодно – это зависело от договоренности с врачом-диагностом. Чаще всего монитор даже не обладал индикацией валидности наложения электродов, что часто приводило к большому числу артефактов в получаемой записи или вся запись могла оказаться артефактной [1].

С развитием вычислительной техники появилась возможность расширить базовую функцию автономного кардиографа. Самым простым, но очень эффективным нововведением является встраивание индицирующего элемента. Это может быть жидкокристаллический индикатор, на который выводятся самые важные сведения о состоянии прибора, ходе записи и, возможно, о состоянии самого пациента. Для расширения опций по настройке записи прибор может быть оснащен пленочной клавиатурой. Эти два улучшения приводят к значительному усложнению алгоритма программы микроконтроллера, который является основой устройства. В табл.1 приведены три типа автономных устройства для съема, записи и хранения ЭКГ с различным набором функций.

Таблица 1

Краткое описание функций устройства	Входящие потоки данных	Исходящие потоки данных	Внешние прерывания	Прерывания периферийных устройств
Регистрация 2-х каналов ЭКГ, регистрация факта нажатия кнопки	ЭКГ (в буфер)	ЭКГ (во внешнюю флэш-память)	Нажатие тревожной кнопки	Прерывания SPI порта (работа с внешним АЦП и флэш-памятью), прерывание таймера
Регистрация 2-х каналов ЭКГ, регистрация факта нажатия кнопки, отображение текущего состояния на ЖКИ, поддержка клавиатуры	ЭКГ (в буфер), нажатие клавиш	ЭКГ (во внешнюю флэш-память), вывод информации на ЖКИ	Нажатие тревожной кнопки, дополнительные прерывания по кнопкам клавиатуры	Прерывания SPI порта (работа с внешним АЦП и флэш-памятью), прерывание таймера
Регистрация 2-х каналов ЭКГ, регистрация факта нажатия кнопки, отображение текущего состояния на ЖКИ, поддержка клавиатуры, регистрация 2-х дополнительных параметров (акселерограмма и GPS позиционирование)	ЭКГ (в буфер), акселерограмма (в буфер), данные GPS (в буфер), нажатие клавиш	ЭКГ (во внешнюю флэш-память), акселерограмма (в буфер), данные GPS (в буфер), вывод информации на ЖКИ	Нажатие тревожной кнопки, дополнительные прерывания по кнопкам клавиатуры, прерывание от GPS-модуля	Прерывания SPI порта (работа с внешним АЦП и флэш-памятью), прерывания UART порта (обмен с GPS модулем) прерывание таймера

Как видно из таблицы, первый прибор имеет всего два потока данных, причем однородных (единый формат и заданная частота поступления данных). Внешнее прерывание одно, причем нажатие тревожной кнопки достаточно редкое событие (максимум 3-6 раз в час). Внутренние прерывания тесно связаны с потоками данных, прерывание от таймера используется для формирования точной частоты дискретизации. Алгоритм микропрограммы для такой системы представлен на рис. 1.

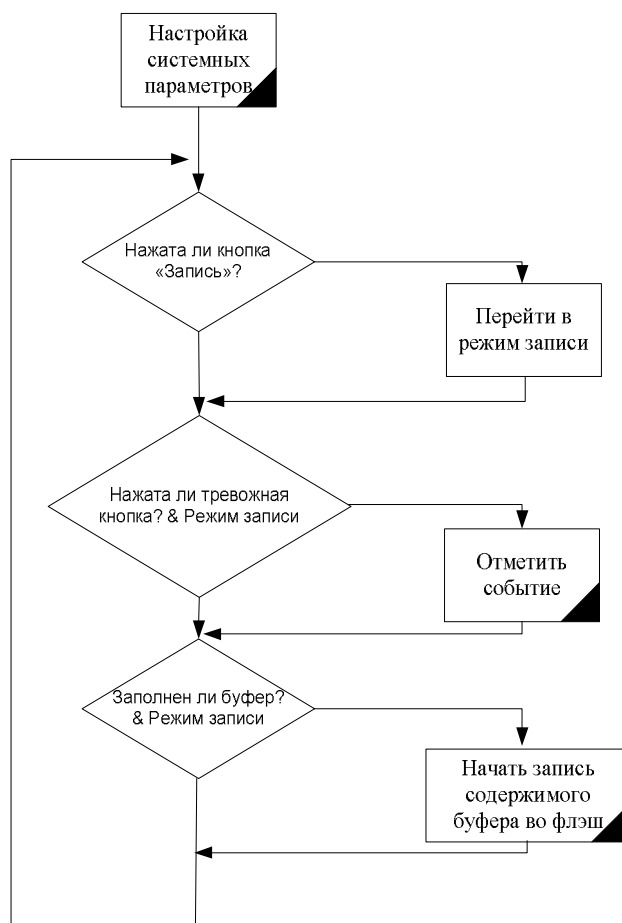


Рис. 1. Алгоритм микропрограммы

На блок схеме не указаны прерывания от периферийных устройств, фактически, используя схему «бесконечного цикла», можно оставить только одно прерывание, генерируемое таймером. По этому прерыванию осуществляется сэмплинг и запись полученных данных в буфер. При заполнении буфера необходимо выставить флаг, который затем анализируется в «бесконечном цикле». Если буфер заполнен – вместо него подставляется чистый, а заполненный буфер переписывается во флэш, затем они снова меняются.

Прерывание нажатия тревожной кнопки так же преобразовывается в событие, отслеживаемое в «бесконечном цикле», так как это редкое и достаточно длительное событие. Следует обратить внимание на новое событие, отслеживаемое в цикле – это нажатие кнопки «Запись», по которому инициируется режим записи устройства. Разработчик может сделать две разные кнопки – «Запись» и «Тревога», а может сделать одну кнопку – после включения питания она является кнопкой «Запись», а после инициирования записи становится кнопкой «Тревога».

При таком алгоритме устройство будет иметь несколько режимов работы, например: режим ожидания (после включения питания), режим записи, режим передачи данных (если используется несъемная флэш-память).

Во втором устройстве, представленном в табл. 1, появляется два дополнительных потока данных, и эти потоки неоднородны. Их невозможно синхронизировать по единственному прерыванию таймера. Необходимо использовать второй таймер или добавлять новые флаги и события в «бесконечный цикл». Если действовать в рамках подобной идеологии, то любые дополнительные режимы и функции можно организовать, только вводя дополнительные флаги и усложняя логику бесконечного цикла, причем резко усложняется тестирование, если эти режимы должны включаться по событиям (возможны взаимные блокировки). Из литературы известно [2], что наличие трех потоков данных и пяти асинхронных прерываний приводит к ситуации, когда операционная система реального времени (ОСРВ) становится более экономичным решением, особенно с учетом надежности программного продукта и возможной его модификации.

Для современных микроконтроллеров на рынке присутствует большое количество ОСРВ, как специализированных (под конкретные семейства микроконтроллеров), так и портируемых практически под любой контроллер соответствующей архитектуры. В табл. 2 приведены некоторые наиболее популярные из существующих ОСРВ [3,4,5].

Таблица 2

Операционная система	Язык	Тип ОС	Типичный объем памяти выделяемый на..			Количество уровней приоритета	Стоимость
			одну задачу	одно событие	диспетчер задач		
EMBOS	ANSI C	Кооперативная (явное переключение задач)	16 Б	6 Б	5 Б	15	1400 у.е.
UROS	ANSI C	Разделение времени, программирование по событиям	14 Б	4 Б	6 Б	15	400 у.е.
Salvo	ANSI C	Кооперативная (явное переключение задач)	4-12 Б	3-4 Б	4-6 Б	15	1200 у.е.

При использовании ОСРВ каждый дополнительный режим или функция может стать либо отдельной исполняемой задачей, либо дополнением к уже существующей. Использование набора стандартных задач может существенно облегчить разработку ПО, а также модификацию существующих устройств для выполнения новых задач. При использовании ОСРВ мы получим задачи и потоки данных, представленные на рис. 2.

Как видно из рис. 2, в данную систему разработчик легко сможет добавлять новые потоки данных или процессы, взаимодействующие с потоками или между собой. Причем каждый процесс может быть унифицирован как блок, с изменяемыми характеристиками. Например, процессы запись данных в буфер и запись из буфера в память могут быть фактически одинаковыми блоками с различными настройками. При использовании ОСРВ один таймер будет использоваться операционной системой для генерации системных тиков, квантов времени, величина которых зависит от быстродействия системы.

Для современных мобильных полиграфов с числом каналов более двух, средствами индикации и управления режимами, использование ОСРВ представляется более целесообразным, чем разработка уникальной программы, использующей принцип «round robin». В качестве ОСРВ для малопотребляющего МПК типа MSP-430 при программировании в среде IAR Embedded Workbench C можно использовать ОСРВ Salvo.

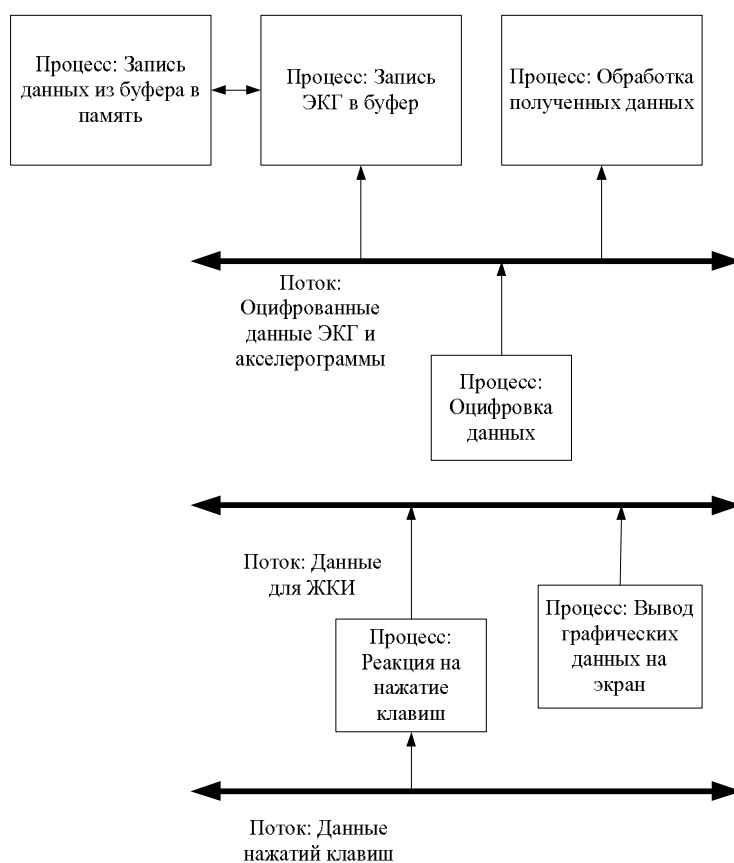


Рис. 2. Распределение потоков данных между процессами

### БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Макаров Л.М.* Холтеровское мониторирование. 2-е изд. – М.: ИД «Медпрактика-М». 2003. – 340 с.
2. *Сорокин С.* Системы реального времени. – СТА 2/97. – М.: СТА-ПРЕСС, 1997. – С. 28-29.
3. [www.salvo.com](http://www.salvo.com).
4. [www.embos.com](http://www.embos.com).
5. [www.ufos.com](http://www.ufos.com).

**Е.А. Башков, О.Л. Вовк**

### **ОЦЕНКА ТОЧНОСТИ КОНТЕКСТНОГО ПОИСКА В КОЛЛЕКЦИЯХ КЛАСТЕРИЗИРОВАННЫХ ИЗОБРАЖЕНИЙ**

Тенденция постоянного увеличения количества изображений, доступных в мировой паутине Internet, повышает интерес исследователей к разработке эффективных механизмов поиска изображений. На данный момент разработано множество систем поиска, основанных на текстовых описаниях изображений. Примерами могут служить такие системы, как система поиска коллекции живописи Национальной галереи искусств США [1], поисковая система AltaVista [2, 3]. Однако технология поиска, основанная на текстовых описаниях, ассоциированных с изображением, может быть применена только при поиске внутри ограниченной коллекции, а не в вычислительной сети Internet, так как неоднозначность соответствия между визуальным содержанием и текстовым описанием снижает показатели точности и полноты поиска [4]. Поэтому для решения задачи поиска изображений во всемирной сети принято использовать механизм контекстного поиска изображений.

Поиск изображений по контекстному содержимому [5] – набор технологий для извлечения семантически связанных изображений из базы данных, основанный на автоматическом выделении характеристик изображения. То есть в базе данных выполняется поиск изображений визуально сходных с заданным изображением (изображением-шаблоном, изображением-образцом) по заданному набору свойств (характеристик, визуальных примитивов). Одной из наиболее используемых технологий контекстного поиска является технология, основанная на сравнении характеристик отдельных регионов (кластеров, областей) изображений.

Цель данной работы – проверить устойчивость статистических методов кластеризации изображений к основным модификациям изображений-образцов поиска путем расчета характеристики точности контекстного поиска.

В рамках поставленной цели выделено решение следующих задач:

- постановка задачи контекстного поиска кластеризированных изображений;
- постановка задачи оценки результатов контекстного поиска;