

УДК 004.67

ОСОБЕННОСТИ РЕАЛИЗАЦИИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ КОМПЬЮТЕРИЗИРОВАННОГО КОМПЛЕКСА ОЦЕНКИ СКОРОСТИ РАСПРОСТРАНЕНИЯ ПУЛЬСОВОЙ ВОЛНЫ

А.В. ЛЕБЕДЕВСКИЙ

Белорусский государственный университет (г. Минск, Республика Беларусь)

Аннотация. Обсуждаются проблемы реализации программного обеспечения верхнего и нижнего уровней комплекса оценки скоростей распространения пульсовой волны в магистральных сосудах и сосудах микроциркуляторного русла. Предложен алгоритм регистрации исключительных ситуаций и передачи информации о них на ПО верхнего уровня без использования механизма исключений. Описан способ форматирования строк, выделенных в статической области памяти. Обсуждается вопрос определения оптимального размера буферов для хранения данных, поступающих от датчиков с различными (отличающимися на порядок) скоростями передачи данных. Определена и обоснована структура данных для оптимального взаимодействия с потоками данных в ПО верхнего уровня.

Ключевые слова: фотоплетизмограмма, обработка исключений, форматирование строк, управление памятью, данные с гетерогенной структурой.

FEATURES OF PROGRAM IMPLEMENTATION FOR COMPUTERIZED COMPLEX OF PULSE WAVE VELOCITY ESTIMATION

A. LEBEDEVSKIY

Belarusian State University (Minsk, Republic of Belarus)

Annotation. The problems of software realization of the high and low levels of the system for estimation of pulse wave propagation velocities in the main vessels and vessels of the microcirculatory channel are discussed. An algorithm for registering exceptional situations and transferring information about them to the high level without using the exception mechanism is proposed. The method of formatting the lines allocated in the static memory area is described. The question of determining the optimal size of buffers for storing data coming from sensors with different (differing by an order of magnitude) data transfer rates is discussed. The data structure for optimal interaction with data streams in high level software is defined and reasoned.

Keywords: photoplethysmogram, exception handling, string formatting, memory management, data with heterogeneous structure.

Введение

Сердечно-сосудистые заболевания лидируют по количеству смертей в мире. В 2022 году вышло Постановление Минздрава РБ о проведении диспансеризации взрослого населения старше 39 лет, главная цель которой – определение факторов риска развития заболеваний на ранней стадии. Специализированной аппаратуры для проведения оценки состояния сердечно-сосудистой системы, кроме обычно принятых (например ЭКГ), на текущий момент нет. Для эффективного проведения диспансеризации требуется недорогая диагностическая аппаратура, удобная в применении при массовом скрининге населения. Важно, чтобы она могла выпускаться белорусскими предприятиями.

На данный момент на кафедре квантовой радиофизики и оптоэлектроники факультета радиофизики и компьютерных технологий БГУ создается компьютеризированный комплекс, предназначенный для оценки временной задержки прихода пульсовой волны в различные ткани. В разрабатываемом комплексе используется набор датчиков, с которых микроконтроллером асинхронно осуществляется считывание данных с учетом с учетом

временных ограничений каждого из датчиков. Эти потоки через канал связи по USB направляются в ноутбук (на верхний уровень программного обеспечения), где обрабатываются, визуализируются и сохраняются в памяти ПК. Чтобы оперативно получать и оценивать отклик организма на функциональное тестовое воздействие, визуализацию на ПК получаемых результатов требуется осуществлять в реальном масштабе времени.

Управление работой используемых датчиков и связь с ноутбуком осуществляет микроконтроллер STM32F103C8T6 на отладочной плате Blue Pill. Для регистрации данных, поступающих с датчиков, и передачи их на верхний уровень создана прошивка (программа функционирования микроконтроллера, нижний уровень программного обеспечения (ПО) разрабатываемого комплекса) на языке программирования C. На верхнем уровне ПО для визуализации и обработки информации используется система компьютерной математики Matlab, позволяющая оперативно разрабатывать алгоритмы обработки и представления получаемых результатов, обеспечивая при этом требуемое быстродействие. В комплексе используются три интегральных модуля: два пульсоксиметрических модуля MAX30102 и модуль ЭКГ AD8232, который подключен через 12-битное 4-канальное АЦП MCP3204.

В процессе разработки ПО приходится решать ряд проблем, от учета особенностей языка программирования Matlab (где передача параметров в функцию по ссылке невозможна без написания пользовательского класса), до ограничений возможностей используемого микроконтроллера (необходимость выделять буфера для датчиков так, чтобы данные от каждого датчика хранились в течение хотя бы 0,2 секунды, поскольку при меньшем периоде заполнения буфера, верхний уровень ПО может не обеспечить своевременный прием данных).

Цель доклада – рассмотреть проблему гибкого и нетребовательного по ресурсам механизма сбора ошибок на ПО нижнего уровня, а также задачу выбора структуры данных и поиска алгоритмов для получения, обработки и визуализации данных с гетерогенной структурой в унифицированном для всех данных виде на ПО верхнего уровня.

Созданные и внедренные в ПО решения могут быть интересны разработчикам программного обеспечения синхронного сбора и обработки данных с гетерогенной структурой для программно-аппаратных комплексов на основе микроконтроллера.

Обработка ошибок

В процессе функционирования ПО нижнего уровня появляется множество ситуаций, при которых дальнейшее выполнение программы или невозможно (нет свободных буферов для временного сохранения данных перед передачей их на ПК), или оно требует дополнительных действий (например, необходимость увеличить счетчик ошибок при нарушении целостности передачи данных по шине I2C). Для сообщения о таких исключительных ситуациях на встраиваемых системах обычно используют коды ошибок. Однако в таком случае разработчику необходимо вручную вести учет всех возможных исключительных ситуаций для обеспечения уникальности кодов ошибок.

Альтернативой является использование механизма обработки исключений. Недостатками исключений являются: накладные расходы на оперативную память и непредсказуемость времени обработки исключения. Даже при использовании механизма обработки исключений, предложенного Renwick et al. [1], который оптимизирован по размеру генерируемого вспомогательного кода и ограничивает максимальное время обработки исключения, остается открытой проблема невозможности захвата исключений из разных потоков (из основного потока и из прерывания).

Наиболее оптимальным вариантом решения проблемы является внедрение сущности «накопитель ошибок», который оперирует ошибками в строковом виде. Информация об ошибке в таком случае представляется или как заданная вручную разработчиком строка или как невыполненное условие, записанное в виде строки выполняемого кода (используя оператор «#» препроцессора языка C [2, p. 16]). В конце эксперимента в случае, если за время процесса измерения произошли ошибки, накопитель ошибок высылает на верхний уровень текстовое представление всех зарегистрированных ошибок, которые произошли во время измерения. На ПО верхнего уровня эти данные могут быть использованы для выявления проблемы, которая привела к данному ряду ошибок и принятия мер по ее устранению. Так как накопитель ошибок

в программе нижнего уровня существует в единственном экземпляре, исключительные ситуации появляющиеся в прерываниях микроконтроллера могут быть зарегистрированы таким же образом, как и ошибки в основном потоке программы. Так как микропроцессор является однопоточным, для синхронизации доступа к накопителю ошибок из основного потока программы и из прерываний не требуется использование примитивов синхронизации, а достаточно отключения оптимизаций для переменной, хранящей количество текущих зарегистрированных ошибок (т.е. объявления переменной с модификатором `volatile`).

Гибкое задание источников данных для измерений

Для управления потоками входных данных (каждый из которых представляет собой одну измеряемую величину в виде «сырых» (необработанных) данных) в унифицированном виде важно определить алгоритм взаимодействия программы верхнего уровня с потоками данных и структуру данных, обеспечивающую возможность хранения информации о потоках данных с разнородной структурой и однозначное определение их типов. Для этого каждому потоку данных присвоен уникальный однобайтовый идентификационный номер (`id`), одинаковый как на верхнем, так и на нижнем уровнях; и на верхнем уровне работа с источниками данных проводится исключительно через структуру данных «кортеж» – отображение, где ключом является `id` измерения, а значением – хранимая информация. Таким образом, считываемые данные, поступающие от каждого источника данных, можно сохранять в кортеже выходных данных по переданному вместе с данными значению `id`. А конфигурировать набор используемых в текущем эксперименте устройств оптимально путем записи параметров источника данных по соответствующему для данного измерения значению `id` в отдельный кортеж параметров.

Заключение

Разработан компьютеризированный комплекс оценки скоростей распространения пульсовой волны в магистральных сосудах и сосудах микроциркуляторного русла, включающий три датчика: два пульсоксиметрических модуля МАХ30102 и модуль ЭКГ AD8232. Рассмотрен ряд проблем, которые возникают перед разработчиком программного обеспечения обработки и анализа данных, поступающих от набора датчиков. Основные из них:

- наличие гибкого и нетребовательного по ресурсам механизма сбора ошибок;
- оптимизация архитектуры ПО, чтобы накопление, обработка и визуализация потоков данных с гетерогенной структурой осуществлялось в унифицированном для всех данных виде.

Предлагаемые в докладе решения описанных проблем могут быть полезны разработчикам ПО как нижнего уровня, выполняющего задачи получения данных, их первичной обработки и передачи на ПК, так и ПО верхнего уровня, выполняющего задачи приема, обработки и визуализации потоков данных в единообразном виде.

Список литературы

1. Low-Cost Deterministic C++ Exceptions for Embedded Systems / J. Renwick, T. Spink, B. Franke // Proceedings of the 28th International Conference on Compiler Construction (CC '19), February 16–17, 2019, Washington, DC, USA. ACM, New York, NY, USA, 11 pages.
2. The C Preprocessor [Electronic Resource] / R.M. Stallman, Z. Weinberg // Free Software Foundation. Mode of access: <https://gcc.gnu.org/onlinedocs/cpp.pdf>. Date of access: 16.10.2024, 88 p.