

ОБ ИНТЕРПРЕТАЦИИ И ХРАНЕНИИ УПРАВЛЯЮЩИХ ПОСЛЕДОВАТЕЛЬНОСТЕЙ ПЕРЕДАЧИ ДАННЫХ В ТЕЛЕМЕТРИЧЕСКОЙ ЗАБОЙНОЙ СИСТЕМЕ

Гараев Р.А.

*Гараев Рашит Аюпович - кандидат физико-математических наук, доцент,
кафедра вычислительной техники и защиты информации,
Уфимский университет науки и технологий, г. Уфа*

Аннотация: статья посвящена аспекту разработки ПО внутрискважинных микропроцессорных модулей установок наклонно-направленного бурения, связанному с хранением и обработкой управляющих последовательностей передачи данных (УППД). Типовой внутрискважинный зонд включает в себя инклинометр и прочие измерительные модули. Распространенными терминами для обозначения подобных систем являются “Measurement While Drilling (MWD)” и “Logging While Drilling (LWD)”. Для обмена с наземным оборудованием в MWD/LWD используются несколько видов канала передачи, самым популярным из которых является гидроканал, имеющий скорость передачи на уровне долей бита в секунду, что обычно приводит к необходимости передавать данные только в направлении “забой-устье” и ограничивать состав и последовательность передаваемых параметров при помощи типовых УППД.

Ключевые слова: MWD/LWD системы для наклонно-направленного бурения, гидроканал, управляющие последовательности передачи данных, микроконтроллеры, многопоточность, FreeRTOS.

ON THE INTERPRETATION AND STORAGE OF DATA TRANSMISSION CONTROL SEQUENCES IN THE TELEMETRY DOWNHOLE SYSTEM

Garaev R.A.

*Garaev Rashit Ayupovich - Candidate of Physical and Mathematical Sciences, Associate Professor,
Department of Computer Science and Information Security,
Ufa University of Science and Technology, Ufa*

Abstract: the article is devoted to the aspect of software development for downhole microprocessor modules of directional drilling rigs, associated with the storage and processing of control sequences for data transmission (CDTS). A typical downhole probe includes an inclinometer and other measurement modules. Common terms for such systems are. “Measurement While Drilling (MWD)” and “Logging While Drilling (LWD)”. MWD/LWD uses several types of transmission channel to exchange with surface equipment, the most popular of which is the hydrochannel, which has a transmission rate of fractions of a bit per second, which usually leads to the need to transmit data only in the bottom-hole direction and limit the composition and the sequence of transmitted parameters using typical UPPDs.

Keywords: MWD/LWD systems for directional drilling, hydraulic channel, control sequences for data transmission, microcontrollers, multithreading, FreeRTOS.

УДК 004.428.4

При разработке реальной MWD-системы для установки наклонно-направленного бурения [1] со структурой, включающей основной модуль телесистемы на базе микроконтроллера 1986VE92QI [2] и модули инклинометра и гамма-модуля, на модуль телесистемы (см. рис.1) возложена задача быть ведущим устройством в сети (по крайней мере при нахождении в скважине). Этот модуль при помощи пакетов протокола Modbus периодически опрашивает остальные модули и формирует информационный поток, передаваемый вверх по гидроканалу. Низкая скорость передачи в канале приводит к необходимости тщательно выбирать состав и порядок передачи конкретных параметров.

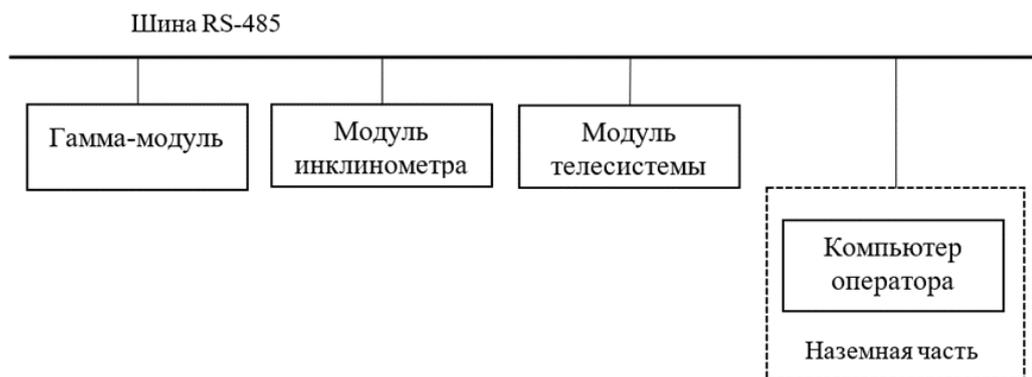


Рис.1. MWD-система на основе канала RS-485 (с подключенным компьютером оператора после подъема скважинных модулей на поверхность).

Однонаправленность канала не дает возможность оперативно менять этот порядок непосредственно с поверхности при нахождении бура в скважине. Почти единственный вариант вносить изменения связан с необходимостью использовать периодические моменты подъема элементов MWD-системы на поверхность (главным образом, по причине разряда аккумуляторных батарей). Подключение в эти моменты компьютера оператора позволяет изменять и эти УППД. В разработке для формализованного описания и задания УППД использован синтаксис, широко применяемый в аналогичных MWD-системах [3].

Для практической реализации данной технологии необходимо решить несколько задач, в частности, выбрать способ и формат энергонезависимого хранения УППД, а также обеспечить на уровне управляющего ПО телесистемы (ТС) корректную и своевременную обработку последовательностей.

С учетом большого количества воздействий, на которые оперативно должно реагировать разрабатываемое ПО ТС, было принято решение использовать разбиение кода на совокупность нескольких задач (или, строго говоря, программных потоков), работающих под управлением ОС реального времени FreeRTOS [4]. Такой подход существенно снизил трудоемкость разработки и обеспечил высокую надежность и устойчивость разработанного кода. Один из потоков был использован в качестве интерпретатора УППД. Потенциально возможно решение, в котором ПО компьютера оператора, в соответствии с новыми пожеланиями изменяло бы ПО ТС и перезаписывало бы его в ПЗУ программ микроконтроллера (МК). Такой подход неудобен хотя бы по причине необходимости целиком менять версию ПО, что было бы невозможно сделать через интерфейс RS-485. Гораздо удобней записывать саму новую УППД с привлечением МК в энергонезависимую память на базе ЭСППЗУ (флэш ПЗУ) [5] в исходной форме или подвергнутую некоторой предварительной обработке (прекомпиляции). В реально разработанном ПО был выбран особенно удобный с точки зрения программы компьютера оператора вариант передачи для хранения во флэш-ПЗУ УППД в исходной текстовой форме.

Поток интерпретатор [6] использует для связи с потоком, готовящим и передающим данные по гидроканалу механизм очередей (он же выбран для организации взаимодействия и всех прочих потоков приложения ТС). Запись в очередь потока интерпретатора "xQueueInterp" (см. файл "Interp.c" [6]) используется для указания конкретной УППД, поэлементное чтение и интерпретация которой актуальны на данный момент. Информацию о каждом очередном параметре, который должен быть отправлен согласно УППД, интерпретатор помещает в небольшую, всего из 4 элементов, очередь "xQueueNext". Очередь читается из потока, отправляющего данные по гидроканалу. Таким образом, в "активном состоянии" поток интерпретатор успевает подготовить не более 4-х элементов в выходной очереди и далее блокируется планировщиком ОС. Разблокировка происходит каждый раз при прочтении хотя бы 1 элемента потоком передачи по гидроканалу. Операция чтения из xQueueNext выполняется сразу после завершения достаточно длительной операции (от секунд до нескольких десятков секунд) передачи очередного параметра по гидроканалу. При такой логике никогда не возникает ситуации неготовности информации об очередном параметре перед началом его передачи по гидроканалу.

Синтаксис УППД [3] предполагает поддержку, в том числе, выражений вида:

"2{3{aTFA:6:p}2{gama:8:p}} BatV:8:p 2{3{aTFA:6:p}1{gama:8:p}} Temp:8:p Bat2:1:n",

здесь помимо названий типовых параметров через двоеточие фигурируют разрядности параметров и наличие или отсутствие битов четности. Использование фигурных скобок подразумевает возможность изменения порядка передачи параметров и многократное повторение блоков, заключенных в скобки. С учетом этого прохождение всей даже короткой и компактной последовательности из нескольких параметров может приводить к реальной отправке многих десятков значений за время от нескольких до десятков минут. За это время практически перед передачей каждого очередного значения модуль ТС успевает запросить и получить от прочих модулей пакет modbus с множеством параметров или измерить некоторые параметры сам, например, в вышеприведенном примере BatV:8:p соответствует 8-ми разрядному коду, содержащему информацию о напряжении на батарее питания. Такой алгоритм обеспечивает максимальную "свежесть" передаваемой по гидроканалу информации. Наиболее рациональная обработка подобных УППД заключается

в их интерпретации непосредственно ПО МК ТС в реальном времени без предварительного преобразования в простую непрерывную последовательность, не содержащую фигурных скобок и чисел, задающих количество повторов. Такая “развернутая” УППД могла бы занимать иногда очень много места в ПЗУ программ или ОЗУ МК.

Интерпретация выражений с изменением порядка, задаваемого скобками, в общем случае может выполняться разными способами, в частности популярна методика с эмуляцией процессора со стековой архитектурой. В рассматриваемом случае был выбран вариант с повторными рекурсивными вызовами одной и той же функции каждый раз при встрече с открывающимися скобками с учетом и значения счетчика повторов. В файле “Interp.c” [6] использована функция `uint16_t interprSeq(void)`, в определении которой можно обнаружить рекурсивный вызов ее же самой. Указанная функция используется при этом для обработки не исходной текстовой УППД, а прекомпилированной, более компактной записи, полученной путем её парсинга и замены текстовых последовательностей типа `aTFA, Inc, Azm, MagF, gama, Temp` и т.п. на компактные однобайтовые записи с установленным или сброшенным старшим битом, в зависимости от наличия или отсутствия бита четности для параметра в исходной УППД.

Указанная прекомпиляция выполняется вызовом функции `int16_t preCmpl(void)`, которая проверяет и корректность УППД, в момент приема пакета Modbus с исходной текстовой УППД модулем ТС от компьютера оператора по каналу RS-485, перед записью в ЭСППЗУ. Таким образом гарантируется, что внутри скважины не возникнут проблемы с некорректной УППД. Результат в виде компактной байтовой последовательности записывается в ОЗУ МК ТС и может перезаписываться вновь при перезапуске МК по сторожевому таймеру путем нового вызова функции `preCmpl`.

Для хранения УППД и различных иных параметров, задающих режим работы разработанной MWD-системы, а также для сохранения объемной каротажной информации, передача которой в реальном времени невозможна по гидроканалу в силу низкой скорости, использован флэш-накопитель на базе отечественной ИМС 1636PP52У того же производителя, что и МК (АО «ПКК Миландр»). ИМС удовлетворяет основным требованиям по объему, энергопотреблению и условиям эксплуатации. При выборе способа организации данных, хранимых в накопителе, было принято решение отказаться от использования традиционной системы FAT или конкретно FAT12 с учетом скромного объема памяти 128К x 8. Основными причинами здесь послужили очень большой объем минимального блока, для которого возможно стирание - 1 сектор в 64 Кбайт, что вдвое больше, чем весь объем ОЗУ используемого МК. В результате операция записи любого нового файла в место, занятое ранее удаленным файлом, не могла бы быть реализована только силами МК без помощи компьютера оператора. Кроме того, с учетом не очень большого разнообразия видов хранимой информации нет смысла пользоваться привычной файловой системой с именованными файлами, каталогами, таблицами размещения и загрузочными записями. Вместо этого пространство матрицы накопителя заполняется информационными блоками со структурой, включающей заголовок, область данных и хвостовик:

<pre>typedef struct { <u>epromBlHd Hdr;</u> <u>uint8_t Dat[124];</u> } <u>epromBl;</u></pre> <p style="text-align: center;"><i>Информационный блок</i></p>	<pre>typedef struct { <u>uint8_t siz0;</u> <u>uint8_t siz1;</u> <u>uint8_t type;</u> <u>uint8_t attrib;</u> } <u>epromBlHd;</u></pre> <p style="text-align: center;"><i>Заголовок блока</i></p>
--	---

Рис. 2. Структуры блока и заголовка.

Для ускорения операции поиска нужного блока по коду типа защита при помощи CRC применяется только к полям `type`, `attrib` и данным. Поиск заголовка следующего блока реализуется за счет использования в текущем блоке поля «размер области данных в блоке». Поле «тип блока», помимо смысла хранимой информации, описывает и состояние блока. Обнуление поля `type` с некоторым исходным значением означает, что данный блок перестал быть актуальным, например, замещается новым блоком с аналогичной по смыслу информацией, т.е. обнуление типа «удаляет» блок, реально не стирая его. При этом учтено, что для флэш-ПЗУ изменение содержимого битов из состояний лог. 1 в лог. 0 возможно, в отличие от обратного перехода из 0 в 1, для которого потребуются стирание, как минимум целого сектора (1636PP52У) или страницы (1636PP3/4У). Сразу после стирания байты содержат код FFh. При таком подходе новый аналогичный по

смыслу «удаленному» блок размещается уже в области с большим адресом. В итоге при достаточно продолжительном повторении процедуры записи однотипных блоков адрес новой копии постепенно достигнет верхнего края адресов флэш-ПЗУ и дальнейшее использование ЭСППЗУ для записи станет возможно только после стирания, как минимум, сектора или всей ИМС. При работе с не слишком большими информационными блоками большую часть времени операции поиска, чтения и записи, даже с заменой однотипных блоков новыми, будут выполняться без необходимости реального стирания секторов/страниц или всего массива памяти. Эпизодические операции очистки всей памяти ИМС или сектора/страницы при почти полном заполнении флэш-ПЗУ должны будут выполняться при взаимодействии с компьютером оператора, когда полезная информация из стираемых секторов сначала может быть скопирована в память компьютера. В результате требования к объему свободного ОЗУ МК значительно снижаются.

Описанный подход к организации информации в ЭСППЗУ реализован в виде библиотеки функций на языке Си. Кроме базовых низкоуровневых операций, связанных с чтением/записью байтов и последовательностей байтов, а также стирания секторов и всей ИМС ЭСППЗУ, в библиотеке представлены функции для работы с информационными блоками, представленными на рис. 2:

- seek4Blk2R - поиск очередного блока для чтения и формирование смещения до следующего;
- rdBlk – считывание тела (полезная информация) блока с проверкой CRC;
- seekFree – поиск начальной точки для записи блока;
- wrBlk2Free – запись нового блока с указанного смещения с формированием CRC;
- mdFBlkHdr – запись/модификация заголовка блока;
- rplceBlk – замена блока заданного типа на аналогичный с новым содержимым ("перезапись").

Библиотека в виде соответствующих заголовочного файла и файла на языке Си, а также все необходимые вспомогательные файлы, включая функцию расчета CRC, размещена в репозитории [7] и может быть использована без ограничений.

Список литературы / References

1. Бескабельные измерительные системы для исследований нефтегазовых скважин (теория и практика) / А. А. Молчанов, Г. С. Абрамов. — М.: ОАО «ВНИИОЭНГ», 2004. — 516 с.
2. Спецификация микроконтроллеров серии 1986BE9x и K1986BE9x. [Электронный ресурс]. Режим доступа: <https://ic.milandr.ru/upload/iblock/65a/z76ysnhw675i5cdkbciklitx70gwifr2/1986BE9X.pdf/> (дата обращения: 10.02.2022).
3. *Kripa Nidhi, David Erdos*. Advanced Mud Pulse Telemetry: M-Ary Encoding for MWD Tools [Chapter 2]. [Электронный ресурс]. Режим доступа: <https://www.erdosmiller.com/advanced-mud-pulse-telemetry-mary/> (дата обращения: 16.02.2023).
4. *Курниц А.* FreeRTOS — операционная система для микроконтроллеров. [Электронный ресурс]. Режим доступа: http://easyelectronics.ru/img/ARM_kurs/FreeRTOS/Kurniz.pdf/ (дата обращения: 16.02.2023).
5. Микросхема ЭППЗУ Flash-типа с последовательным интерфейсом 1636PP52У, K1636PP52У, K1636PP52УК. [Электронный ресурс]. Режим доступа: <https://ic.milandr.ru/> (дата обращения: 10.02.2022).
6. *Гараев Р.А.* Программный поток под FreeRTOS для интерпретации управляющих последовательностей передачи данных по гидроканалу в составе программного обеспечения системы MWD на базе K1986BE92QI (MILANDR) [Электронный ресурс]. Режим доступа: https://github.com/garaevra22/MWD_Sequence_Interp/ (дата обращения: 04.02.2023).
7. *Гараев Р.А.* Библиотека для работы с отечественной ИМС ЭППЗУ (флэш 128К x 8) 1636PP52 по интерфейсу SPI в системе на базе отечественного микроконтроллера (МК) K1986BE92QI (MILANDR) [Электронный ресурс]. Режим доступа: [https://github.com/garaevra22/miscella/tree/main/K1986BE92%2B1636%D0%A0%D0%A052\(Milandr\)/](https://github.com/garaevra22/miscella/tree/main/K1986BE92%2B1636%D0%A0%D0%A052(Milandr)/) (дата обращения: 16.03.2023).