

УДК 621.3-1/-8

РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ, ОТВЕЧАЮЩЕГО ЗА АВТОМАТИЧЕСКОЕ УПРАВЛЕНИЕ МАГАЗИНОМ СОПРОТИВЛЕНИЙ P4831

Третьяков Михаил Андреевич, студент, направление подготовки 11.03.04 Электроника и нанoeлектроника, Оренбургский государственный университет, Оренбург
e-mail: ms.miha12345@gmail.com

Научный руководитель: **Фролов Сергей Сергеевич**, кандидат технических наук, доцент кафедры промышленной электроники и информационно-измерительной техники, Оренбургский государственный университет, Оренбург
e-mail: frolovsergey7@mail.ru

***Аннотация.** Данный проект необходим для системы автоматизации калибровки или же поверки каналов измерения сопротивлений программируемого логического контроллера с помощью эталонного магазина сопротивлений. Необходимость автоматизации заключается в трудоемкости процесса – 100–200 измерений для одного канала, что делает процесс долгим и неэффективным. Решается вопрос разработки программного обеспечения системы на языке программирования Си, для микроконтроллера фирмы «Миландр», отвечающего за автоматическое управление эталонным магазином сопротивлений. Представлены схемы подключения периферийных модулей к микроконтроллеру, разработаны алгоритмы работы программы. Предлагается решение задачи приема передачи данных, а также их хранение в памяти микроконтроллера с использованием кольцевого буфера, а также вариант фильтрации принятых данных от ошибок. Разработанная программа проверена средствами автономной и комплексной отладки.*

***Ключевые слова:** поверка каналов, каналы сопротивления, каналы ПЛК, калибровка каналов, аналоговые каналы, программирование микроконтроллеров, автоматизация измерений.*

***Для цитирования:** Третьяков М. А. Разработка программного обеспечения, отвечающего за автоматическое управление магазином сопротивлений P4831 // Шаг в науку. – 2023. – № 2. – С. 67–73.*

DEVELOPMENT OF SOFTWARE RESPONSIBLE FOR THE AUTOMATIC MANAGEMENT OF THE RESISTANCE STORE P4831

Tretyakov Mikhail Andreevich, student, training program 11.03.04 Electronics and Nanoelectronics, Orenburg State University, Orenburg
e-mail: ms.miha12345@gmail.com

Research advisor: **Frolov Sergey Sergeevich**, Candidate of Technical Sciences, Associate Professor of the Department of Industrial Electronics and Information and Measuring Technology, Orenburg State University, Orenburg
e-mail: frolovsergey7@mail.ru

***Abstract.** This project is necessary for an automation system for calibration or verification of PLC resistance measurement channels using a reference resistance store. The need for automation lies in the complexity of the process – 100–200 measurements for one channel, which makes the process long and inefficient. The issue of developing the system software in the C programming language for the microcontroller of the company “Milander”, responsible for the automatic control of the reference resistance store, is being solved. Schemes of connection of peripheral modules to the microcontroller are presented, algorithms of the program operation are developed. The solution of the problem of receiving data transmission, as well as their storage in the memory of the MC, using a ring buffer, as well as the option of filtering the received data from errors is proposed. The developed program has been tested by means of autonomous and complex debugging.*

Key words: channel verification, resistance channels, PLC channels, channel calibration, analog channels, microcontroller programming, measurement automation.

Cite as: Tretyakov, M. A. (2023) [Development of software responsible for the automatic management of the resistance store P4831]. *Shag v nauku* [Step into science]. Vol. 2, pp. 67–73.

Материал статьи связан с автоматизацией тестирования, регулировки и поверки каналов и модулей аналогового ввода программируемого логического контроллера (далее – ПЛК), предназначенных для выполнения измерений электрических унифицированных сигналов термо-, тензо- и других преобразователей сопротивления. При тестировании применяется магазин эталонов сопротивлений.

Необходимость автоматизации заключается в трудоемкости процесса поверки или же калибровки каналов сопротивлений ПЛК. Зачастую в данном процессе необходимо сравнивать большое количество значений, считанных с модулей аналогового ввода с значениями, выставленными на эталонном

магазине сопротивлений. Количество этих значений может превышать сотни точек отсчета, что делает процесс выставления значений на декадах магазина сопротивлений достаточно трудоемким и занимающим долгое время процессом. Решить данную задачу можно с помощью автоматизации выставления значений на декадах, путем их автоматического поворота на нужное положение в соответствии с полученным значением по протоколу связи.

Структурная схема, составленная автором, разрабатываемой командой АСУ ПРО автоматизированной системы представлена на рисунке 1. Работа автора состояла в разработке программного обеспечения для микроконтроллера системы.

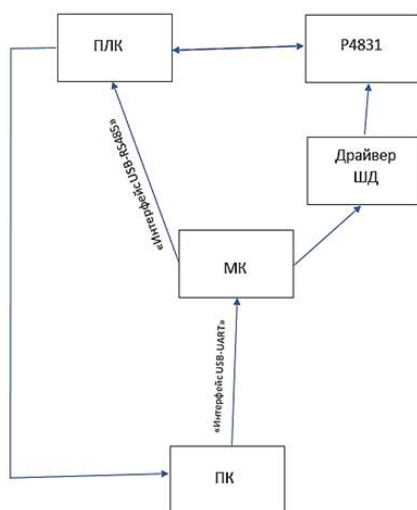


Рисунок 1. Структурная схема подключения автоматизированного, эталонного магазина сопротивлений к ПЛК

Источник: разработано автором

Микроконтроллер получает значение необходимого сопротивления от ПК по интерфейсу UART и передает значения шагов, необходимых для поворота каждого из двигателей, на драйверы, подключенные к двигателям, поочередно проворачивая каждый из двигателей, начиная изменения с того, который отвечает за младший разряд значения сопротивления.

Задачей автора являлась разработка программного обеспечения (ПО) для МК. В настоящей работе представлено описание результатов его разработки.

Разработанное ПО обеспечивает выполнение всех функций измерительной системы рисунка 1. В данной статье уделено внимание лишь некоторым из них – связи между ПК и МК и управлению шаговыми двигателями, непосредственно связанных с процессом поверки, и подпрограммы, реализующие их.

В данной работе использовался микроконтроллер «K1986BE92QI» на базе ядра «ARM-Cortex M3» компании «Миландр», предоставленный предприятием заказчика. Отладка и тестирование выполнялись на отладочной плате LDM-K1986BE92QI.

Данная отладочная плата имеет все необходимое для разработки программного обеспечения, порты подключения, средства для использования встроенной периферии.

Подключение шагового двигателя к отладочной плате производилось с помощью драйвера

«A4988». Была изучена спецификация, а также схема подключения данного драйвера к микроконтроллеру и двигателю.

Подключение драйвера шагового двигателя к отладочной плате осуществляется контактами, показанными на рисунке 2.

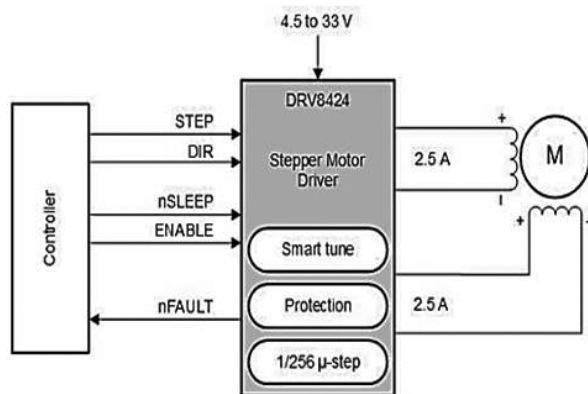


Рисунок 2. Подключение драйвера ШД к отладочной плате

Источник: разработано автором

EN – включение и выключение модуля («+0В» – включен, «+5В» – выключен);

STEP – управляющий вывод, при каждом положительном импульсе;

DIR – управляющий вывод, если подать «+5В» двигатель будет вращаться по часовой стрелке, а если подать «+0В» – против часовой стрелки «+2В», «2А», «+1В» и «1А» – подключение обмоток двигателя;

VDD & GND – питание внутренней логики от «+3В» до «+5,5В»;

RST – сброс драйвера;

SLP – вывод включения спящего режима, если подтянуть его к низкому состоянию, драйвер перейдет в спящий режим.

Разработанный автором алгоритм работы программы представлен на рисунках 3 и 4.



Рисунок 3. Алгоритм основной программы

Источник: разработано автором

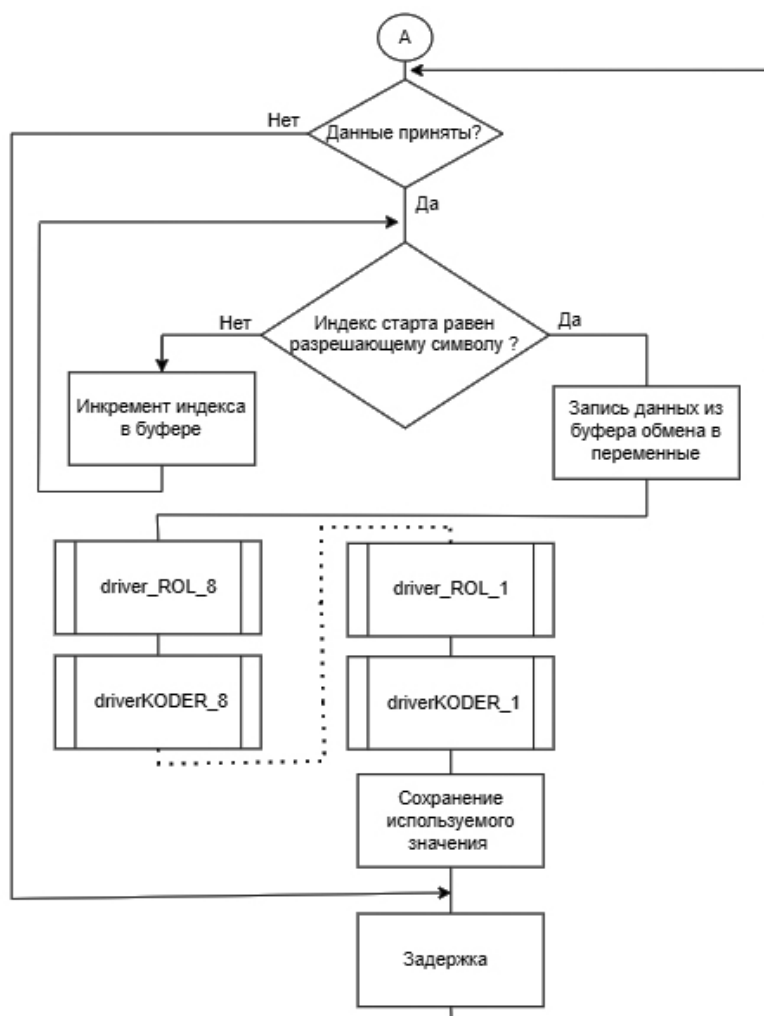


Рисунок 4. Алгоритм основной программы

Источник: разработано автором

Рассмотрим некоторые особенности интерфейса приемопередачи между ПК и отладочной платой, посредством которого в МК поступает информация о сопротивлении.

Приём данных осуществляется по прерыванию по заполнению буфера приёмника UART [1].

Для хранения принятых значений сопротивления магазина решено использовать кольцевой буфер. Кольцевой буфер, или циклический буфер – это структура данных, использующая единственный буфер фиксированного размера таким образом, как будто бы после последнего элемента сразу же снова идет первый [4]. Такая структура предоставляет возможность буферизации потоков данных без потери информации в связи с переполнением. Буфер настроен так, что запись начинается с нулевого индекса. Вторая запись в буфер перезапишет занятые ячейки в буфере в случае, если ее размер будет превышать размер буфера. Принцип

работы буфера представлен на рисунке 5, разработанном автором.

Также необходимо задать определенные правила отправки данных с ПК, а также по приему данных. Для исключения помех настраиваем прием только с символа «:» или же 0x3A в коде ASCII обозначающего, что после него идет значение сопротивления до символа «.» или же 0x2E в коде ASCII, обозначающего конец. Также эти символы необходимы для того, чтобы в буфере отделять одно значение сопротивления от другого.

Для моделирования отправки данных при отладке в ПК использовали программу Terminal1_9_b. Максимальное сопротивление, регулируемое магазином сопротивлений – это шестизначное значение плюс три символа после запятой, в связи с этим отправляем десять символов, включая запятую, разделяющую целую и десятичную части, а также разрешающий символ «:» и конечный символ «.».

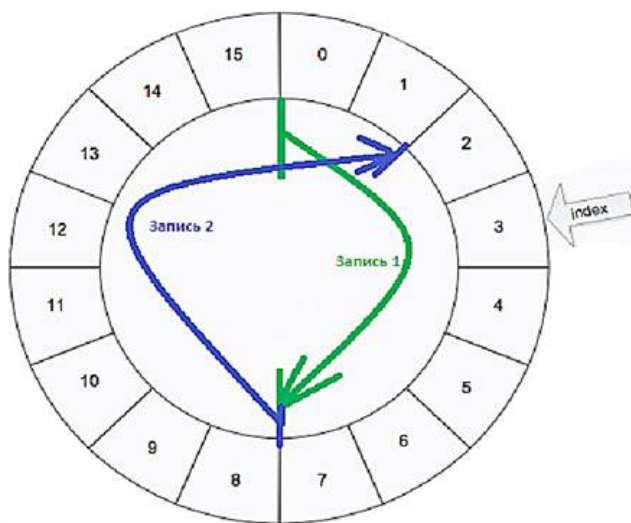


Рисунок 5. Кольцевой буфер
Источник: разработано автором

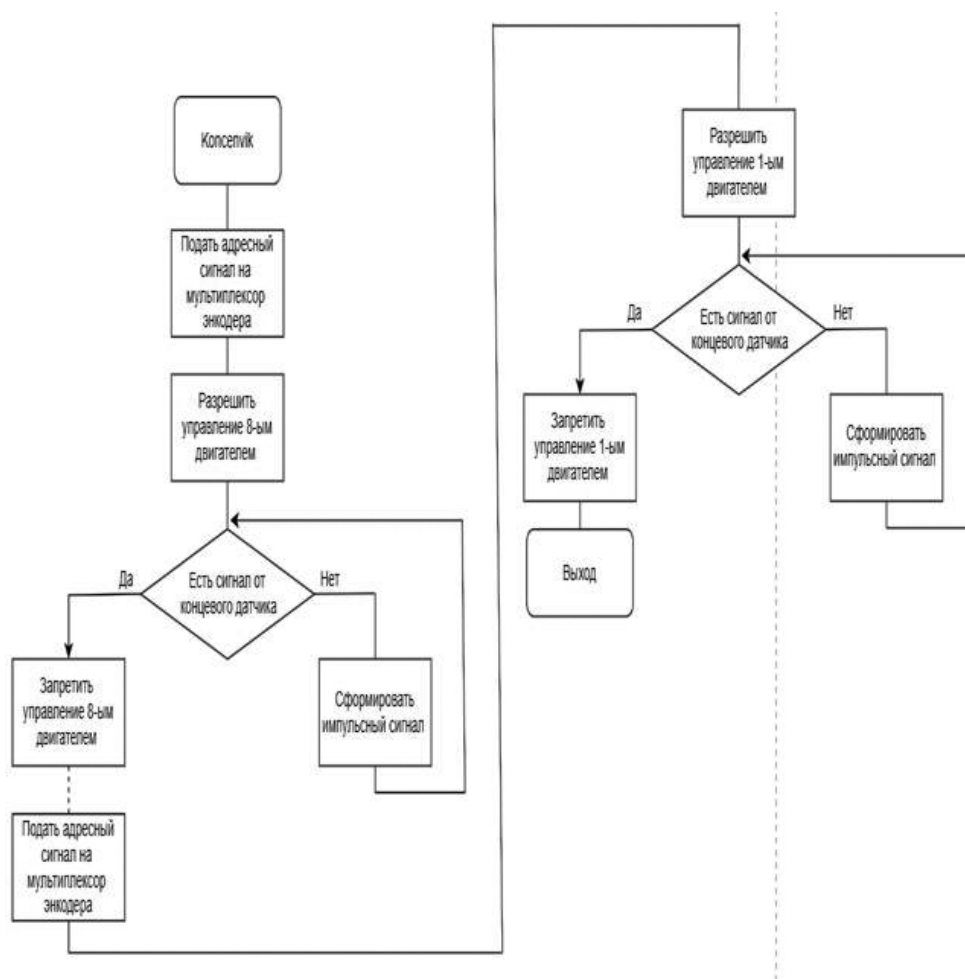


Рисунок 6. Алгоритм подпрограммы перевода двигателей в нулевое положение
Источник: разработано автором

После выполнения предварительных настроек работы МК необходимо запретить работу двигателей путем подачи на их разрешающий порт ENABLE единичного сигнала. Управление двигателями происходит путем выставления определенных битов в регистре данных RXTX [2].

Программа переходит из основного модуля в подпрограмму, переводящую двигатели в нулевое положение перед началом работы, для исключения некорректных выставлений значений из-за того, что двигатель находился в случайном положении, а не в нулевом. Разработанный автором алгоритм подпрограммы представлен на рисунке 6.

Далее реализован бесконечный цикл ожидания информации о сопротивлении, в котором происходит проверка условия, приняты ли данные. Если нет – программа возвращается в начало цикла ожидания.

Если данные были приняты, происходит проверка принятых данных на наличие разрешающего символа «<»». Если символ не равен размещаемому, происходит инкремент порядкового номера принятого символа в буфере и возврат к проверке символа. В случае же когда принятый символ соответствует началу, будет происходить запись из буфера принятых данных в переменные, для дальнейшей работы с ними.

Для реализации движения двигателя на «шаговый» контакт драйвера «STEP» необходимо подавать импульсы. Реализовано это с помощью подачи высокого потенциала, задержки на определенное время, затем подача низкого потенциала. Временем задержки будет определяться скорость вращения двигателя [5].

Для выбора направления движения необходимо на контакт «DIR» драйвера подать постоянный сигнал или же наоборот обнулить его.

Для последовательного управления несколькими двигателями, от младшего к старшему, было решено на контакт «ENABLE» драйвера подавать постоянный сигнал, а при выборе драйвера «обнулять» его, для того, чтобы перевести драйвер в рабочее состояние.

Количество шагов, совершенных двигателем, будет определяться значением, отвечающим за повтор циклов подачи импульсов на порт «STEP».

Комплексная отладка ПО и аппаратной части системы происходила с использованием программной среды Keil uVision 5, одного шагового двигателя печатной головки «EM-462» от принтера компании «Erson», а также с одним драйвером. Проведение комплексной отладки необходимо для проверки корректности работы системы [3]. В микроконтроллерах имеются интерфейсы JTAG/SWD, через которые можно осуществлять отладку ПО в среде Keil. Для реализации интерфейса JTAG/SWD предназначены обычные пользовательские выводы микроконтроллера – выводы на порте А и выводы на порте В. Таким образом, при необходимости можно выбрать те выводы, которые не задействованы в данный момент для основной функции, выполняемой на микроконтроллере. Отладочный интерфейс позволяет произвести первоначальную загрузку программы в память микроконтроллера и запуск программы на исполнение (в том числе и пошаговое), просмотр значений переменных, областей памяти и внутренних регистров процессорного ядра и периферии [6, с. 24–25]. Для тестирования необходимо было поочередно подключать к различным пинам отладочной платы драйвер с подключенным ШД. Передача значения задания сопротивления выполнялась из программы Terminal1_9_b, затем фиксировалась запись данных в буфер, а затем фиксировалось количество шагов двигателя, таким образом происходила отладка.

Результаты отладки показали корректность функционирования разработанной программы. Полученный результат – полученное ПО – позволит завершить отладку завершённой системы автоматизации поверки аналоговых каналов сопротивления в ООО «АСУ ПРО» после поступления оставшихся 7 драйверов и двигателей, и установки последних в конструкцию магазина.

Литература

1. Бирюкова О. В., Корецкая И. В. Влияние электромагнитных помех на работу контрольно-измерительного модуля // Системы синхронизации, формирования и обработки сигналов. – 2022. – Т. 13. – № 5. – С. 26–34. – EDN: BTWQJW.
2. Боскумбаев Л. С. Разработка программно-аппаратного комплекса «Управление шаговым двигателем» // Наука, техника и образование. – 2014. – № 5(5). – С. 105–107. – EDN: TEUTXJ.
3. Гуньков С. А., Шепель В. Н., Трипкош В. А. Моделирование работы систем с целью проведения комплексной отладки программного обеспечения // Университетский комплекс как региональный центр образования, науки и культуры: Материалы Всероссийской научно-методической конференции, Оренбург, 23-25 января 2019 года. – Оренбург: Оренбургский государственный университет, 2019. – С. 721–724. – EDN: YZBFVR.

4. Карпова И. П. Хранение данных системой автономных устройств // Информационные технологии. – 2013. – № 5. – С. 29–34. – EDN: QAQWSR.
5. Колосков С. Ю., Захаров П. В., Леготкин Н. В. Управление униполярным шаговым двигателем с помощью USB персонального компьютера // Информационные технологии. Радиоэлектроника. Телекоммуникации. – 2014. – № 3. – С. 151–154. – EDN: UZNHGD.
6. Лопарев М., Шумилин С. Отладка программного обеспечения в микроконтроллерах 1986BE8 и 1986BE81 // Компоненты и технологии. – 2017. – № 7(192). – С. 24–25. – EDN: ZEMZET.

Статья поступила в редакцию: 16.10.2022; принята в печать: 05.06.2023.

Автор прочитал и одобрил окончательный вариант рукописи.