

Министерство науки и высшего образования Российской
Федерации
Федеральное государственное автономное образовательное
учреждение высшего образования
«Национальный исследовательский ядерный университет
«МИФИ»

УДК 539.12.01

ОТЧЕТ
О НАУЧНО-ИССЛЕДОВАТЕЛЬСКОЙ РАБОТЕ
РАСШИРЕНИЕ СИСТЕМЫ ТЕМПЕРАТУРНОГО
КОНТРОЛЯ
СПЕКТРОМЕТРИЧЕСКОГО МАГНИТА СПАСЧАРМ

Научный руководитель

к.ф.-м.н.,

ведущий научный сотрудник НИЦ _____

КИ-ИФВЭ

П. А. Семёнов

Выполнил работу _____

М. В. Гани

Москва, 2026

Содержание

ВВЕДЕНИЕ И ПОСТАНОВКА ЗАДАЧИ	3
Цель работы	3
Состояние работ на текущий момент	4
АППАРАТНАЯ ЧАСТЬ РАЗРАБАТЫВАЕМОГО МОДУ- ЛЯ	4
Используемая плата	4
Структура измерительного канала	5
Особенности DS2480B	6
ВЫБОР ПРОГРАММНОЙ АРХИТЕКТУРЫ	7
Первоначальный вариант с FreeRTOS	7
Принятая архитектура	7
ПРОГРАММНАЯ РЕАЛИЗАЦИЯ	8
Обмен по USART3 через прерывания	8
Инициализация DS2480B	9
Реализация ведущего устройства 1-Wire	10
Работа с DS18B20	10
Периодический опрос датчиков	11
ОТЛАДКА И ВОЗНИКШИЕ ПРОБЛЕМЫ	12
Чувствительность DS2480B к последовательности об- мена	12
Почему временно отказались от FreeRTOS	13
Меры, принятые для повышения устойчивости	13
ПОЛУЧЕННЫЕ РЕЗУЛЬТАТЫ	14
ДАЛЬНЕЙШИЕ РАБОТЫ	14
ЗАКЛЮЧЕНИЕ	15
СПИСОК ЛИТЕРАТУРЫ	16

ВВЕДЕНИЕ И ПОСТАНОВКА ЗАДАЧИ

Температурный контроль спектрометрического магнита СПАСЧАРМ относится к системе медленного контроля установки. Для такой системы важны не только сами значения температуры, но и устойчивость их получения. Компьютерная часть системы должна регулярно получать актуальные данные, а микроконтроллерный узел не должен зависать из-за сбоя одного датчика или кратковременной ошибки обмена.

В предыдущем варианте работы рассматривалась общая архитектура температурного контроля на базе EPICS, Raspberry Pi, микроконтроллерных модулей и датчиков температуры. На этом этапе работа была сосредоточена на микроконтроллерной части системы: STM32F207VGT6, преобразователе DS2480B и цифровых датчиках DS18B20 на шине 1-Wire. Сначала нужно добиться устойчивого обмена с датчиками, а уже после этого добавлять Modbus и интеграцию с EPICS.

Цель работы

Целью данного этапа является разработка и отладка программной части для чтения нескольких датчиков температуры DS18B20 через преобразователь DS2480B с использованием микроконтроллера STM32F207VGT6.

Для достижения цели были поставлены следующие задачи:

1. подготовить программную структуру проекта для работы с USART3 микроконтроллера STM32F207VGT6;
2. реализовать часть программы, отвечающую за обмен с DS2480B через UART с использованием прерываний;
3. реализовать базовые операции DS2480B: инициализацию, калибровку, переключение между режимом команд и режимом данных, сброс шины 1-Wire и передачу битов/байтов;
4. реализовать операции ведущего устройства 1-Wire,

- включая поиск устройств на общей шине;
5. реализовать работу с DS18B20: определение режима питания, запуск преобразования температуры, чтение внутренней памяти датчика и проверку контрольной суммы CRC;
 6. организовать периодический опрос датчиков с хранением последних измеренных значений;
 7. оценить применимость FreeRTOS для данного узла и выбрать более устойчивую схему для текущего этапа отладки.

Состояние работ на текущий момент

На текущем этапе обмен по Modbus еще не реализован. Это сделано сознательно: сначала необходимо стабилизировать работу с DS2480B и датчиками DS18B20. Преобразователь DS2480B оказался чувствительным к порядку инициализации, паузам между командами и переключению режимов. Поэтому основное внимание было уделено именно обмену с датчиками. Ошибка даже в одном байте может привести к неверному определению ответа датчика, сбою поиска ROM-кодов или ошибке чтения внутренней памяти DS18B20.

В результате подготовлен набор исходных файлов для проекта STM32CubeIDE. Проект разделен на несколько простых частей, чтобы отдельно проверять обмен с DS2480B, операции 1-Wire и опрос датчиков.

АППАРАТНАЯ ЧАСТЬ РАЗРАБАТЫВАЕМОГО МОДУЛЯ

Используемая плата

В качестве микроконтроллерного узла используется плата на базе STM32F207VGT6. У этого микроконтроллера достаточно интерфейсов UART/USART и производительности для задач медленного контроля. Для связи с преобразователем DS2480B используется USART3.

Фотография платы, на которой велась отладка, приведе-

на на рисунке 1. Именно на этой плате проверялся обмен STM32 с DS2480B и наблюдались сигналы на линии 1-Wire осциллографом.

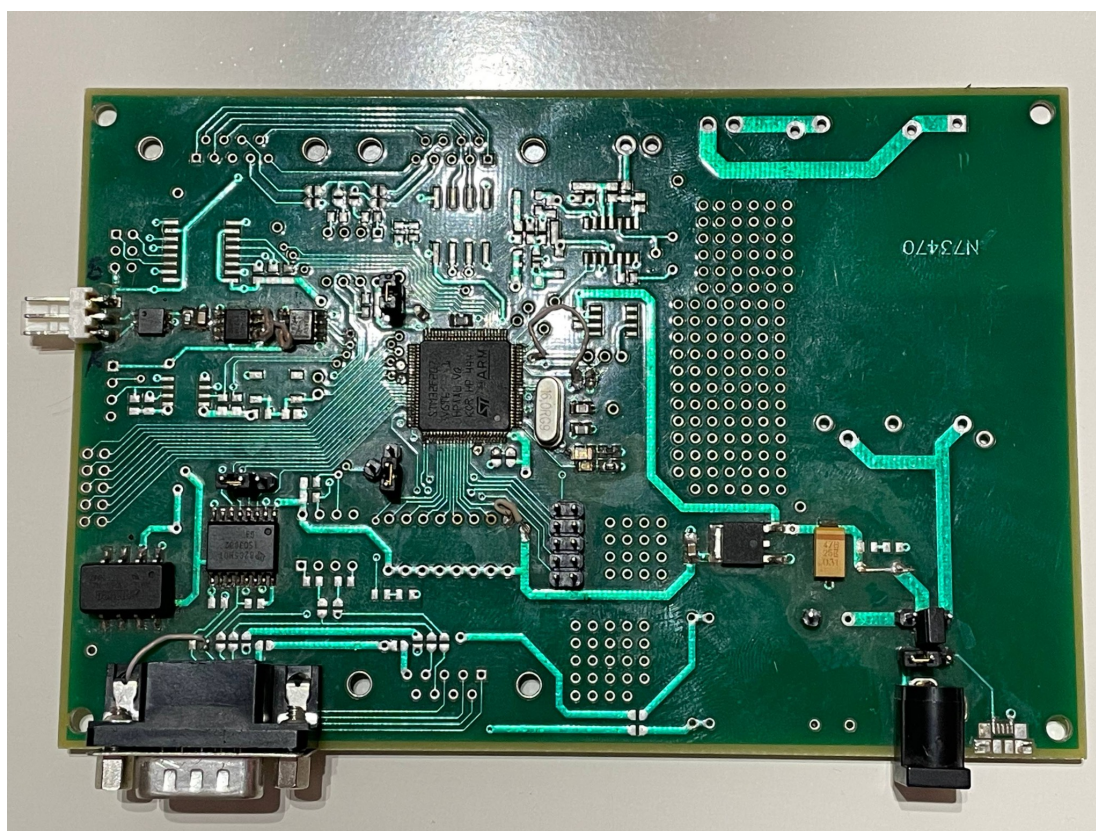


Рис. 1. Плата на базе микроконтроллера STM32F207VGT6, используемая для отладки подсистемы 1-Wire

Структура измерительного канала

Измерительный канал строится по следующей цепочке:

STM32F207VGT6 → USART3 → DS2480B → шина 1-Wire → DS18B20

STM32 не формирует импульсы 1-Wire напрямую. Эту задачу выполняет DS2480B. Микроконтроллер отправляет ему команды по UART, а DS2480B уже формирует сброс шины, интервалы чтения и записи, а также усиленную подтяжку линии питания, если она нужна для паразитного питания датчиков. Такой подход удобен для реальной системы, но требует аккуратного соблюдения последовательности команд DS2480B.

Основные аппаратные элементы канала приведены в таблице 1.

Таблица 1. Основные элементы измерительного канала

Элемент	Назначение
STM32F207VGT6	Управление обменом, хранение последних значений температур, дальнейшая выдача данных по Modbus.
USART3	Последовательный интерфейс для связи с DS2480B. На этапе отладки выбран режим 8N1 и скорость 9600 бод.
DS2480B	Преобразователь UART в 1-Wire. Формирует сброс шины, передает биты и байты, управляет усиленной подтяжкой линии питания.
DS18B20	Цифровые датчики температуры на шине 1-Wire. Идентифицируются 64-битным ROM-кодом.

Особенности DS2480B

DS2480B нельзя рассматривать как обычный переходник UART/1-Wire. У него есть собственная логика работы и два основных режима: режим команд и режим данных. В режиме команд передаются команды для самого DS2480B: сброс шины, настройка параметров, включение и выключение усиленной подтяжки линии питания. В режиме данных байты уже уходят на шину 1-Wire к датчикам.

После включения питания или полного программного сброса DS2480B требуется калибровка. Поэтому первый сброс на скорости 9600 бод относится к настройке самого преобразователя, а не к опросу датчиков на шине 1-Wire.

ВЫБОР ПРОГРАММНОЙ АРХИТЕКТУРЫ

Первоначальный вариант с FreeRTOS

Сначала рассматривался вариант с FreeRTOS. Идея была понятной: одна задача могла бы опрашивать датчики, другая - отвечать за будущий Modbus, а остальные части программы работали бы отдельно. Для законченного контроллера такая схема в целом удобна.

Но при первичной отладке DS2480B этот вариант оказался неудобным. Преобразователь чувствителен к порядку команд и к паузам между отдельными действиями. В системе с RTOS появляются дополнительные задержки: переключение задач, системный таймер, участки HAL с запретом прерываний и служебные паузы внутри драйверов. Для обычного UART-обмена это часто не мешает, но при отладке DS2480B из-за этого сложнее понять, где находится ошибка: в подключении, в порядке команд или в работе планировщика.

Поэтому на текущем этапе FreeRTOS была временно исключена, а программа сделана проще. UART работает через прерывания, а команды DS2480B и 1-Wire выполняются последовательно. Такой вариант проще проверять пошагово и сравнивать с описанием DS2480B.

Принятая архитектура

В итоговой структуре программа разделена на несколько частей. Это позволяет не смешивать в одном файле работу UART, команды DS2480B, алгоритм 1-Wire и периодический опрос температуры.

Таблица 2. Основные части программы

Файл/часть программы	про-	Назначение
----------------------	------	------------

<code>ow_uart3_transport</code>	Обмен с DS2480B через USART3. Передача и прием выполняются по одному байту через обработчики прерываний HAL UART.
<code>ds2480b</code>	Команды DS2480B: калибровка, режим команд/режим данных, сброс шины, передача одного бита, передача байтов данных, усиленная подтяжка линии питания.
<code>ow_master</code>	Базовые операции 1-Wire: сброс шины, запись/чтение бита, запись/чтение байтов, а также команды поиска и выбора ROM-кода.
<code>ds18b20</code>	Работа с DS18B20: поиск датчиков, запуск измерения, чтение данных и проверка результата.
<code>temp_service</code>	Периодический опрос датчиков и хранение последних значений температуры для будущего обмена по Modbus.

Главный принцип выбранной структуры простой: сначала сделать надежный обмен с датчиками, а уже потом добавлять внешние протоколы. В текущем виде проект уже содержит буфер последних температур и счетчики ошибок. Это пригодится при добавлении карты регистров Modbus.

ПРОГРАММНАЯ РЕАЛИЗАЦИЯ

Обмен по USART3 через прерывания

Обмен с DS2480B по USART3 реализован в файлах `ow_uart3_transport.c/.h`. При инициализации программа сохраняет указатель на `UART_HandleTypeDef` и сразу включает прием одного байта через `HAL_UART_Receive_IT`. Передача также выполняется по одному байту через

HAL_UART_Transmit_IT.

Для DS2480B это удобно тем, что каждая операция сводится к передаче команды и приему ответа с заданным таймаутом. Обмен считается завершенным, когда переданы все байты и принято ожидаемое количество ответных байтов. Если ответ не пришел, срабатывает таймаут, и программа фиксирует ошибку.

Упрощенно логика обмена выглядит следующим образом:

1. приложение запускает обмен и передает буфер команд;
2. первый байт отправляется через HAL_UART_Transmit_IT;
3. после завершения передачи обработчик завершения передачи отправляет следующий байт;
4. обработчик приема складывает принятые байты в буфер;
5. функция опроса проверяет, завершился обмен или истек таймаут.

Такой подход не блокирует работу самого UART-драйвера и при этом остается достаточно простым. В дальнейшем эту схему можно будет перевести в полностью неблокирующий автомат состояний, но для отладки DS2480B текущий вариант понятнее.

Инициализация DS2480B

Ключевой частью проекта стала корректная инициализация DS2480B. После включения питания преобразователь нужно перевести в понятное начальное состояние. В коде это сделано через функцию DS2480B_CalibrateAfterReset. Перед калибровкой USART3 явно переводится на скорость 9600 бод. Затем отправляется команда 0xC1.

Эта операция выделена отдельно, чтобы не смешивать настройку DS2480B с обычным сбросом шины 1-Wire. Если перепутать эти два действия, дальнейший обмен с датчиками может нарушиться.

Реализация ведущего устройства 1-Wire

Файл `ow_master` скрывает особенности DS2480B и дает остальной программе обычные операции 1-Wire. В нем реализованы сброс шины, запись и чтение битов, запись и чтение байтов, а также выбор датчика по ROM-коду. Отдельно реализован поиск устройств, позволяющий найти несколько датчиков на одной шине.

Поиск ROM-кодов не использует встроенный ускоренный поиск DS2480B. Это сделано специально: так алгоритм проще проверять при отладке. Микроконтроллер последовательно читает пару битов `id_bit` и `cmp_id_bit`, выбирает направление поиска и записывает выбранный бит. При расхождении запоминается позиция последнего нуля. За несколько проходов это позволяет получить ROM-коды всех устройств.

Для датчиков DS18B20 затем дополнительно проверяется семейный код `0x28` и контрольная сумма ROM-кода. Это позволяет отфильтровать неверно считанные коды и уменьшить вероятность ложного обнаружения устройства.

Работа с DS18B20

В файле `ds18b20.c` реализованы операции, необходимые для опроса температуры:

1. поиск датчиков DS18B20 на шине;
2. определение режима питания датчика;
3. запуск преобразования температуры;
4. чтение внутренней памяти датчика;
5. проверка контрольной суммы CRC8;
6. пересчет исходного кодового значения в миллиградусы Цельсия.

В коде предусмотрена работа как с обычным питанием датчиков, так и с паразитным питанием. Для паразитного питания добавлена усиленная подтяжка линии через DS2480B. На текущем этапе отладка в основном ведется на обычном питании, потому что этот вариант проще и позволяет сначала

ла проверить базовый обмен.

Время преобразования выбирается по разрешению датчика. Для разрешения 12 бит используется задержка 750 мс. После запуска преобразования программа ждет завершения измерения и только потом читает внутреннюю память каждого датчика.

Периодический опрос датчиков

Файл `temp_service.c` реализует простой автомат состояний. Он не зависит от Modbus и EPICS, а занимается только датчиками. Основные состояния приведены в таблице 3.

Таблица 3. Состояния модуля опроса температур

Состояние	Назначение
UNINITIALIZED	Модуль создан, но поиск датчиков еще не выполнен.
IDLE	Ожидание следующего периода измерения.
CONVERTING	Всем датчикам отправлена команда запуска преобразования, идет ожидание завершения измерения.
READING	Последовательное чтение внутренней памяти каждого найденного датчика с выбором по ROM-коду.
ERROR	Ошибка отладки или обнаружения устройств. В текущем варианте автоматическое восстановление еще не выполняется.

При успешном чтении для каждого датчика обновляются исходное значение температуры, температура в миллиградусах Цельсия, время последнего обновления и признак на-

личия датчика. При ошибке увеличивается счетчик ошибок обмена. Это пригодится для Modbus: в регистры можно будет отдавать не только температуру, но и состояние датчика.

ОТЛАДКА И ВОЗНИКШИЕ ПРОБЛЕМЫ

Чувствительность DS2480B к последовательности обмена

Основная практическая сложность была связана с тем, что ответы по UART не всегда означали корректную работу всей цепочки. При отладке приходилось отдельно проверять передачу команд, сигналы на линии 1-Wire и смысл данных, полученных от датчиков.

Типовые проблемы, которые возникали на этом этапе:

1. обмен по UART между STM32 и DS2480B мог присутствовать, но на выходе DS2480B отсутствовали импульсы на линии 1-Wire; это проверялось осциллографом;
2. отсутствовал ожидаемый импульс присутствия после сброса шины 1-Wire;
3. при наличии нескольких датчиков срывался поиск ROM-кодов;
4. датчики иногда возвращали набор байтов, но эти данные не интерпретировались как корректная температура;
5. возникали ошибки контрольной суммы CRC при чтении внутренней памяти датчика;
6. обмен мог зависать после неправильного переключения между режимом команд и режимом данных;
7. было трудно отличить ошибку в последовательности команд от аппаратной проблемы на линии.

После этого отладка велась не по одному случайному удачному чтению, а по всей последовательности операций: инициализация DS2480B, сброс шины, поиск ROM-кодов, запуск преобразования и чтение внутренней памяти датчика. Такой порядок позволял понять, на каком именно этапе возникает сбой.

Почему временно отказались от FreeRTOS

FreeRTOS может пригодиться в дальнейшем, но для текущей отладки она была лишним усложнением. Сначала нужно было получить понятный и повторяемый обмен с DS2480B без дополнительных задержек от планировщика и служебных частей HAL.

Поэтому текущий вариант сделан без FreeRTOS. После стабилизации обмена с датчиками к ней можно будет вернуться и перенести опрос температуры в отдельную задачу.

Меры, принятые для повышения устойчивости

Для повышения устойчивости были приняты следующие решения:

1. зафиксирована начальная скорость 9600 бод для первичной отладки;
2. калибровка DS2480B выделена в отдельную функцию;
3. первая команда сброса после полного сброса не используется как обычный сброс шины;
4. все переходы между режимом команд и режимом данных выполняются явно;
5. для каждого обмена задан таймаут;
6. прием и передача USART3 вынесены в обработчики HAL;
7. поиск устройств выполняется без встроенного ускоренного поиска DS2480B, чтобы алгоритм было проще проверять;
8. чтение внутренней памяти датчика обязательно проверяется по контрольной сумме CRC8.

Эти меры пока не закрывают все возможные ошибки, но дают понятную основу для дальнейшей отладки. Дальнейшая работа должна быть направлена на автоматическое восстановление после сбоев: полный сброс DS2480B, повторный поиск датчиков и сохранение последнего корректного значения при временной ошибке.

ПОЛУЧЕННЫЕ РЕЗУЛЬТАТЫ

На текущем этапе подготовлена программная основа для микроконтроллерной части подсистемы температурного контроля. Основные результаты приведены ниже.

1. Подготовлены файлы проекта для STM32F207VGT6 и USART3.
2. Реализован обмен по UART с использованием прерываний HAL.
3. Реализован драйвер DS2480B с учетом калибровки после сброса, режима команд/режима данных и управления усиленной подтяжкой линии питания.
4. Реализованы базовые операции 1-Wire и работа с DS18B20, включая поиск ROM-кодов, запуск измерения, чтение данных и проверку контрольной суммы CRC8.
5. Реализован периодический опрос температуры с состояниями IDLE, CONVERTING, READING и хранением последних значений.
6. Выделены основные места для дальнейшей отладки: инициализация DS2480B, переключение режимов, поиск ROM-кодов и проверка контрольной суммы CRC8.
7. Сделан вывод, что на этапе первичной отладки обмен с DS2480B лучше вести без FreeRTOS, чтобы исключить неконтролируемые задержки и упростить отладку.

Текущая структура уже подготовлена к будущему добавлению Modbus. Для этого потребуются связать массив найденных датчиков и последние значения температур с картой регистров. Опрос температуры можно оставить независимым: он будет только обновлять буфер, а Modbus будет читать уже готовые значения.

ДАЛЬНЕЙШИЕ РАБОТЫ

Следующий этап работы должен быть направлен на доведение текущего проекта первичной отладки до устойчивого

модуля медленного контроля. Для этого требуется:

1. добавить автоматическое восстановление после ошибок DS2480B и ошибок контрольной суммы CRC;
2. определить окончательный формат хранения температур и статусов;
3. разработать карту регистров Modbus для температур, ROM-кодов и диагностических счетчиков;
4. реализовать ведомое устройство Modbus RTU на отдельном USART;
5. провести проверку обмена STM32 с компьютерной частью системы;
6. после стабилизации обмена с датчиками вернуться к вопросу FreeRTOS или полностью неблокирующего автомата состояний.

Важно, что Modbus не должен запускать измерение температуры напрямую. Правильнее хранить последние измеренные значения в памяти микроконтроллера, а по Modbus отдавать уже готовые данные. Тогда запрос по Modbus не будет ждать DS18B20, у которых преобразование температуры при 12 битах занимает до 750 мс.

ЗАКЛЮЧЕНИЕ

В ходе текущего этапа НИРС была выполнена работа с микроконтроллерной частью подсистемы температурного контроля спектрометрического магнита СПАСЧАРМ. Вместо расширенного описания всей системы медленного контроля основное внимание было уделено конкретному узлу: STM32F207VGT6, USART3, DS2480B и датчикам DS18B20.

Показано, что основная сложность текущего этапа связана не с пересчетом температуры, а с устойчивым обменом через DS2480B. Поэтому была выбрана более простая схема без FreeRTOS, но с UART на прерываниях. Такой вариант легче проверять по шагам и удобнее использовать при первичной отладке.

Подготовленный код дает основу для дальнейшей интеграции с Modbus и EPICS. После завершения отладки обмена с датчиками можно будет добавить карту регистров Modbus, связать ее с буфером последних температур и включить модуль в общую систему медленного контроля.

СПИСОК ЛИТЕРАТУРЫ

1. Maxim Integrated. DS2480B Serial 1-Wire Line Driver with Load Sensor. Datasheet.
2. Maxim Integrated. DS18B20 Programmable Resolution 1-Wire Digital Thermometer. Datasheet.
3. STMicroelectronics. RM0033 Reference manual. STM32F205/215, STM32F207/217 advanced Arm-based 32-bit MCUs.
4. STMicroelectronics. STM32F2 HAL and low-layer drivers documentation.
5. EPICS Collaboration. EPICS Application Developer's Guide.