

## **ОПИСАНИЕ ПРОЕКТИРУЕМОЙ ПРОГРАММЫ МИКРОКОНТРОЛЛЕРА**

### **ОГЛАВЛЕНИЕ**

1. Введение
2. Основные характеристики устройства
3. Особенности программы
  - 3.1. Инициализация
  - 3.2. Загрузчик
  - 3.3. Пользовательская программа
4. Алгоритм программы
5. Функции периферии
6. Функция записи во FLASH память
7. Функция чтения из FLASH памяти
8. Функция записи на SD карту
9. Функция чтения из SD карты
10. Функция сканирования АЦП
11. Примеры

## 1. Введение

В данном описании будут обозначены основные требуемые характеристики устройства и алгоритм проектируемой программы.

Проектируемая программа предназначена для работы на платформе S-600, в комплектацию которой входит:

- Основной модуль S-600A со схемой питания от батареи (перемычка USB-OTG отсутствует, микросхема ЗУ Li-ION отсутствует)
- Дополнительный модуль S-600B с установленным модемом NEOWAY N11
- Дополнительный модуль S-600C с выносным модулем S-600D
- Батарея Li-SOCl<sub>2</sub>

## 2. Основные характеристики устройств

Проектируемое устройство представляет собой регистратор изменения и измеритель физических величин, способный хранить и передавать измеренные значения посредством сети сотовой связи. Устройство планируется использовать вне помещения на открытом воздухе, что предполагает наличие осадков, больших перепадов температур, нахождение в рабочем состоянии без внешнего источника питания. Программа устройства должна быть оптимизирована под минимально возможное энергопотребление.

Основные требуемые характеристики представлены ниже:

- Измерение температуры с разрешением до 0,1 градуса.
- Измерение относительной влажности с разрешением до 0,1 %.
- Измерение абсолютного давления с разрешением до 0,1 гПа.
- Измерение наклона устройства с разрешением до 0,01 градуса.
- Измерение микроускорений до 50 mG.
- Измерение концентрации монооксида углерода с разрешением до 0,1 ppm.
- Измерение относительной чистоты воздуха.
- Хранение измеренных данных.
- Передача архива событий по расписанию и тревоге через сотовую сеть.
- Время автономной работы в течение 5 лет.
- Возможность удаленного конфигурирования и обновления ПО.
- Наличие интуитивно-понятного интерфейса «машина – человек».
- Индикация режимов работы внешним светодиодом.

### 3. Особенности программы

#### 3.1. Инициализация

Программа проектируемого устройства должна содержать обязательный блок инициализации, гарантирующий стабильную работу устройства на платформе S-600. Инициализация представляет собой набор подпрограмм, выполняющих начальные установки и настройки периферии микроконтроллера и дополнительных модулей, проверка работоспособности устройства в целом. Инициализация выполняется однократно при подаче питания либо после сброса. В случае сброса, программа инициализации должна определить причину сброса, принять меры по исправлению ошибки (если таковая есть) и передать её на сервер. Программа инициализации может входить в состав загрузчика либо выполняться отдельно. Инициализация должна выполняться раньше выполнения загрузчика. В случае использования разъема программирования в качестве пользовательского GPIO (в данной реализации не актуально), при выполнении Test-Point на вывод BOOTO (подача высокого уровня), разъем программирования должен принудительно отключать функции пользовательского GPIO. Основные обязательные этапы инициализации приведены ниже:

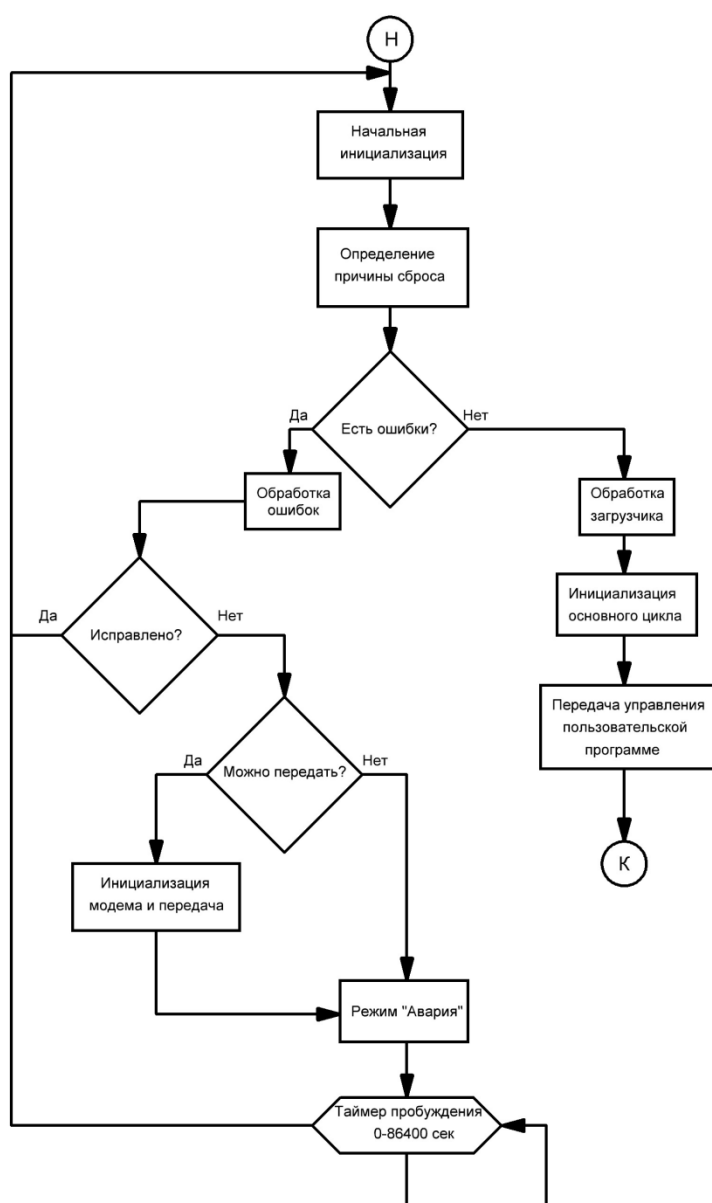
- Проверка основного тактового генератора на кварцевом резонаторе 8 МГц.
- Проверка дополнительного тактового генератора на кварцевом резонаторе 32,768 кГц.
- Конфигурирование портов ввода-вывода согласно таблице 1 «Описания» в состояние «выключено всё».
- Выполнение загрузчика, если запущен.
- Поэтапная проверка периферии:
  - Конфигурирование АЦП, измерение напряжения батареи.
  - Конфигурирование интерфейса QSPI и проверка FLASH памяти.
  - Конфигурирование интерфейса SDMMC. Детектирование наличия карты и её проверка.
  - Конфигурирование интерфейса USB, измерение АЦП канала питания USB, проверка подключения.
  - Перевод периферии в режим «Выключено».
- Поэтапная проверка дополнительных модулей:
  - Переход на тактирование от основного кварцевого генератора.

- Конфигурирование интерфейса UART, подача питания и проверка связи с модемом.
- Проверка работоспособности модема.
- Перевод модема в режим «Выключено» и переход на тактирование от внутреннего генератора.
- Конфигурирование интерфейса I2C, подача питания.
- Чтение регистра «WHO\_AM\_I» датчика давления LPS22HH.
- Чтение регистра «WHO\_AM\_I» акселерометра LIS3DSH.
- Выполнение «Self-Test» акселерометра LIS3DSH.
- Проверка наличия «АСК» при обращении по адресу датчика температуры SHT30.
- Включение «Heater» и проверка работоспособности.
- Перевод шины I2C и датчиков в режим «Выключено».
- Конфигурирование шины SPI, подача питания.
- Подача питания на датчик TGS8100.
- Чтение «STATUS» регистра AD7798.
- Переключение канала на AVdd, преобразование.
- Проверка бита наличия ИОН.
- Измерение АЦП микроконтроллера напряжения ИОН.
- Подача питания на инклинометр, запуск измерения и анализ данных.
- Измерение при помощи AD7798 значения каналов 1 и 2 (оси X и Y) и сравнение с ранее полученными через интерфейс SPI.
- Перевод инклинометра в режим «Выключено»
- Переключение AD7798 на канал 3 и измерение концентрации, анализ.
- Преобразование АЦП микроконтроллера значения датчика ADPS9005.
- Преобразование АЦП микроконтроллера значения датчика TGS8100.
- Перевод всех ранее включенных модулей в режим «Выключено».
- Анализ данных.
- Диагностика найденных ошибок и исправление если возможно.
- Переход к пользовательской программе.

Каждый этап сопровождается отметкой со статусом выполнения в системный журнал. В процессе выполнения инициализации, после этапа конфигурирование портов, выполняется световая индикация статуса выполнения. В случае выполнения инициализации с ошибками (низкий разряд батареи, отсутствие связи с датчиком...), программа инициализации должна сохранить данные, и если возможно дальнейшее выполнение программы, то передать сведения во время сеанса связи на сервер.

На рисунке 1 представлен алгоритм начальной инициализации.

Рисунок 1.



### **3.2. Загрузчик**

Загрузчик позволяет обновлять программу микроконтроллера. Получение программы возможно через USB-DFU, USB – диск, UART интерфейс, чтение образа с SD-карты или FLASH памяти. Процесс перезаписи программы должен контролироваться, должна быть сделана резервная копия старой программы, которая восстанавливается в случае ошибки обновления. Загрузчик может запуститься только во время инициализации.

Удаленное обновление программы выполняется в следующей последовательности:

- получение новой программы и сохранение её в энергонезависимой памяти
- пользовательская программа завершает все подпрограммы и выполняет RESET
- в процессе выполнения инициализации выполняется запуск загрузчика
- обновление программы и выполнение RESET
- выполнение инициализации и переход к пользовательской программе
- стирание образов и временных данных, после успешного обновления

### **3.3. Пользовательская программа**

Выполнение пользовательской программы начинается сразу, после выполнения инициализации. Основные функции выполнения программы:

- опрос инклинометра
- опрос датчика температуры и влажности SHT30
- опрос акселерометра LIS3DSH
- опрос датчика давления LPS22HH
- опрос датчика освещенности (?)
- опрос датчика чистоты воздуха (?)
- опрос датчика CO (?)
- сохранение данных в памяти
- передача данных через модем
- индикация выносным светодиодом

Каждая функция выполняется по расписанию или асинхронно, если сработал триггер (тревога). После выполнения функции, интерфейс или датчик переводится в состояние «Выключено», если не будут осуществляться дальнейшие действия или продолжительное чтение.

## **4. Алгоритм программы**

Пользовательская программа выполняется в соответствии с заданным алгоритмом, определяющим промежуточные состояния, выполнение условий, запуск триггеров, обработку данных, сохранении и передачу данных. В программе должен присутствовать основной цикл, в ходе выполнения которого возможны ветвления. Перед запуском основного цикла выполняется подпрограмма инициализации цикла.

На рисунке 2 представлен прототип основного цикла.

Рисунок 2.



Планировщик в алгоритме выполняет создание сигнала «Пробуждения» микроконтроллера, перевод его в «Сон» и генерацию временных интервалов. Каждый временной интервал вычисляется на основании текущего режима работы.

Сигнал, генерируемый планировщиком, запускает компаратор интервалов «Запуск по времени», который в свою очередь разрешает выполнение функций периферии. Данные, полученные после выполнения функции, попадают в блок сравнения «Обработка порогов», где создается событие, которое попадает в «Триггер событий».

Триггер событий это набор состояний, после обработки пороговых значений. Может представлять собой битовый набор, где позиция бита указывает на принадлежность к определенному событию. Сигнал, создаваемый триггером событий, проходит маску, разрешает запись в архив событий (вместе со значениями), меняет режим работы и может асинхронно запустить синхронизацию с

удаленным сервером. Сигнал, изменяющий режим работы, одновременно влияет на планировщик, изменяющий генерацию временных интервалов согласно установленному режиму.

В процессе синхронизации с сервером, устройством могут быть получены данные, которые необходимо принять и обработать. Это может быть обновление программы устройства (изменение логики работы), изменение режима работы (изменение выполнения логики работы) или настроек (настройка параметров функций). Обновление программы устройства подразумевает стирание пользовательской программы и/или загрузчика с последующей начальной инициализацией. Изменение режима работы представляет собой корректировку выполнения существующей логики работы и является фиксированной. Изменение настроек влияет на результат выполнения логических функций. Разграничение вышеперечисленных задач выполняет блок «Распределения данных».

Пример:

Начальное состояние: устройство находится в режиме «Норма», когда отсутствуют тревожные события.

Воздействие: однократное срабатывание порога акселерометра LIS3DSH.

Выполнение:

- Планировщик пробуждает микроконтроллер
- Компаратор интервалов определяет, что в данный момент будет запущено чтение данных с акселерометра
- Происходит чтение данных с акселерометра
- Детектируется наличие превышения порога вибраций
- Событие, минуя компаратор интервалов, запускает чтение данных с инклинометра и передается в триггер событий
- Происходит чтение данных с инклинометра
- Детектирования изменения угла наклона нет
- Событие передается в триггер событий
- Триггер событий в свою очередь, выдает сигнал изменения режима работы инклинометра (уменьшает интервал запуска)
- В архив событие поступают значения текущего угла и детектированного ускорения с пометкой «Внимание»
- После выполнения функций в текущем интервале (если следующий ещё не начал действовать), планировщик переводит микроконтроллер в режим «Сон»
- Дальнейшее пробуждение микроконтроллера покажет, что событие с акселерометра было сброшено, триггер событий выдаст сигнал изменения режима работы (увеличит интервал запуска инклинометра), в архив будет помещено событие

## 5. Функции периферии

Каждое обращение к периферии (чтение или передача данных) представляет собой законченную функцию, состоящую из следующих этапов:

- подача питания на периферию, если необходимо
- выполнение конфигурирования, если необходимо
- чтение данных/передача данных
- перевод в режим «Выкл»
- отключение питания, если необходимо



## **6. Функция записи во FLASH память**

1. Подача питания путём установки низкого уровня на затвор транзистора (вывод PD14 установить в лог. 0)
2. Сконфигурировать интерфейс QSPI
3. Конфигурация памяти, если необходимо и выдержать паузу на внутреннюю инициализацию микросхемы памяти
4. Выполнить операцию чтения и выполнить сравнение данных. Если данные изменять не надо, то переход на пункт 7
5. Выполнить операцию записи
6. Выполнить проверку записи данных. Если возникла ошибка, создать событие и выполнить цикл исправления ошибки (например «Remap»)
7. Если в очереди записи имеются ещё данные, то переход на пункт 4
8. Выдержать паузу на внутренние процессы микросхемы памяти, если необходимо
9. Отключить питание, установив высокий уровень на порту PD14
10. Выключить интерфейс QSPI, выводы перевести в нейтральное состояние (цифровой вход с подтяжкой на шину VSS, в зависимости от токов утечки выводов микросхемы памяти)

## **7. Функция чтения из FLASH памяти**

1. Подача питания путём установки низкого уровня на затвор транзистора (вывод PD14 установить в лог. 0)
2. Сконфигурировать интерфейс QSPI
3. Конфигурация памяти, если необходимо и выдержать паузу на внутреннюю инициализацию микросхемы памяти
4. Выполнить операцию чтения
5. Если необходимо дальнейшее чтение, то переход на пункт 4
6. Выдержать паузу на внутренние процессы микросхемы памяти, если необходимо
7. Отключить питание, установив высокий уровень на порту PD14
8. Выключить интерфейс QSPI, выводы перевести в нейтральное состояние (цифровой вход с подтяжкой на шину VSS, в зависимости от токов утечки выводов микросхемы памяти)

## **8. Функция записи на SD карту**

1. Подача питания путём установки низкого уровня на затвор транзистора (вывод PD3 установить в лог. 0)
2. Сконфигурировать интерфейс SDMMC. Проверка состояния вывода PD4 на состояние наличия карты (лог. 0 указывает на наличие установленной карты)
3. Конфигурация SD карты, если необходимо и выдержать паузу на внутреннюю инициализацию.
4. Выполнить операцию чтения и выполнить сравнение данных. Если данные изменять не надо, то переход на пункт 7
5. Выполнить операцию записи
6. Выполнить проверку записи данных. Если возникла ошибка, создать событие и выполнить цикл исправления ошибки (например «Remap»)
7. Если в очереди записи имеются ещё данные, то переход на пункт 4
8. Выдержать паузу на внутренние процессы, если необходимо
9. Отключить питание, установив высокий уровень на порту PD3
10. Выключить интерфейс SDMMC, выводы перевести в нейтральное состояние (цифровой вход с подтяжкой на шину VSS, в зависимости от токов утечки)

## 9. Функция чтения из SD карты

1. Подача питания путём установки низкого уровня на затвор транзистора (вывод PD3 установить в лог. 0)
2. Сконфигурировать интерфейс SDMMC. Проверка состояния вывода PD4 на состояние наличия карты (лог. 0 указывает на наличие установленной карты)
3. Конфигурация SD карты, если необходимо и выдержать паузу на внутреннюю инициализацию
4. Выполнить операцию чтения
5. Если необходимо дальнейшее чтение, то переход на пункт 4
6. Выдержать паузу на внутренние процессы, если необходимо
7. Отключить питание, установив высокий уровень на порту PD3
8. Выключить интерфейс SDMMC, выводы перевести в нейтральное состояние (цифровой вход с подтяжкой на шину VSS, в зависимости от токов утечки)

## 10. Функция сканирования АЦП

1. Подача питания на измеряемый модуль. Для канала измерения АКБ/Батареи, это установить низкий логический уровень на порту PD13, для датчика чистоты воздуха порт PD5, для датчика освещенности порт PD6
2. Сконфигурировать АЦП, выбрать канал измерения
3. Сконфигурировать ИОН (использовать внешний от ADR291 если включен)
4. Выдержать паузу на переходные процессы
5. Запустить преобразование несколько раз и вычислить среднее
6. Отключить питание модуля. Перевести соответствующие порты в режим цифрового входа и подтяжкой к шине VDD
7. Выключить АЦП и ИОН

## 11. Примеры реализации начальной стадии работы с устройством

### Общее описание

- Используя DIP переключатель выбирается режим работы.
- Режим работы считывается однократно при включении (подаче питания или reset).
- Считывание DIP переключателя производится после подачи лог. 0 на GND\_SW PD13. После считывания, порт переводится в аналоговый режим.
- После подачи питания, чтения режима и начальной инициализации сделать паузу на 1 секунду.
- Обозначение режима «XXX» соответствует входному логическому состоянию портов МК. Так, переключатель в состоянии OFF-ON-OFF (1, 2, 3) даст комбинацию «101».
- Сокращения: AI – аналоговый вход, IN – цифровой вход, PU – подтяжка на шину VDD, PD – подтяжка на шину VSS, OD – выход открытый исток, Hi – выход с высоким уровнем, Lo – выход с низким уровнем, AF – функция.
- Так, например, IN\_PU – цифровой вход с подтяжкой на шину VDD, Lo-PU – выход с низким уровнем и подтяжкой на шину VDD.
- Начальная инициализация портов:  
PORTA (PA15 – PA0): AI, AF, AF, AI, AI, AI, AI, AI, AI, AI, AI, AI, Lo, Lo, AI, AI.  
PORTB (PB15 – PB0): AI, AI, AI, AI, Hi, Lo, Lo, Hi, AI, AI, AI, AI, AI, Hi, AI, AI.  
PORTC (PC15 – PC0): AF, AF, AI, AI, AI, AI, AI, AI, AI, AI, AI, AI, AI, AI, AI, AI.  
PORTD (PD15 – PD0): AI, Hi, AI, AI, AI, AI, AI, AI, Lo, AI, AI, AI, Hi, AI, AI, AI.  
PORTE (PE15 – PE0): AI, AI, AI, AI, AI, AI, AI, AI, AI, AI, AI, AI, AI, AI, AI, AI.  
PORTH (PH1 – PH0): AF, AF.  
VREF+: выключено.
- При переходе в FAULT мигает красный 200мс выкл – 200мс вкл в течении 5 секунд и затем reset.

#### **- Режим 0 (111) «Тестирование энергопотребления».**

Цель: получение реального минимального значения потребления тока.

- Подача питания и старт МК.
- Начальная инициализация портов.
- Включение зеленого светодиода.
- Пауза 1 секунду.
- Переход в режим сна с пробуждением от RTC (интервал 1 минуту).
- После пробуждения выполняется software reset и цикл повторяется.

#### **- Режим 1 (110) «Тестирование цепей питания».**

Цель: измерение напряжений и калибровка обратных связей DC-DC, проверка поведения цепей питания при коммутации.

- Подача питания и старт МК.
- Начальная инициализация портов.
- Включение зеленого светодиода.

Цикл.

- Инициализация портов: PD14 – Lo, PD13 – Lo, PD12 – Hi, PD7 – Hi, PD3 – Lo
- Пауза 10 секунд.
- Начальная инициализация портов.
- Инициализация портов: PA2 – Hi, PB10 – Hi, PB11 – Lo, PB2 – Lo, PB8 – Lo.
- Пауза 10 секунд.
- Начальная инициализация портов.

#### **- Режим 2 (101) «Тестирование кварцевых осцилляторов».**

Цель: настроить кварцевые генератора и проверить их стабильность.

- Подача питания и старт МК.
- Начальная инициализация портов.
- Запуск осцилляторов на кварцевых резонаторах и проверка их работы (флаги).
- Если оба осциллятора запустились, мигание зеленым 1 секунда – 1 секунда. Если только на 8 МГц, последовательность 2 секунды пауза – 1 секунда свечения. Если только на 32,768 кГц, то 1 секунда паузы, 2 секунды свечения. Если ни один, то постоянное свечение красным. МК тактируется от внутреннего RC с PLL.
- В процессе выполнения программы, будут проверены уровни управления кварцами, частоты, стабильность работы при повышении температуры (до 70 градусов), проверены емкости Cs.

#### **Режим 3 (100) «Тестирование максимального потребления».**

Цель: измерение напряжений, стабильности цепей питания, измерение тока.

- Подача питания и старт МК.
- Начальная инициализация портов.

- Включение всех модулей: PD14 – Lo, PD13 – Lo, PD12 – Hi, PD7 – Hi, PD3 – Lo, PA2 – Hi, PB10 – Hi, PB11 – Lo, PB2 – Lo, PB8 – Lo, PC7 – Lo, PC6 – Lo, PD15 – Lo, PE1 – Hi, PE0 – Lo, PBO – Lo.
- Бесконечный цикл.

#### **- Режим 4 (011) «Тестирование QSPI».**

Цель: проверка работы FLASH памяти.

- подача питания и старт МК.
- Начальная инициализация портов.

Цикл.

- подача питания на FLASH память, инициализация интерфейса и процессинг данных.
- Включение зеленого если норма, включение красного, если ошибка.
- Пауза 3 секунды.
- Выключение питания, деинициализация интерфейсов, включение синего светодиода.
- Пауза 3 секунды.

#### **- Режим 5 (010) «Тестирование SDMMC».**

Цель: проверка работы SD карты.

- подача питания и старт МК.
- Начальная инициализация портов.

Цикл.

- подача питания на SD карту, инициализация интерфейса и процессинг данных.
- Включение зеленого если норма, включение красного, если ошибка.
- Пауза 3 секунды.
- Выключение питания, деинициализация интерфейсов, включение синего светодиода.
- Пауза 3 секунды.

#### **- Режим 6 (001) «Тестирование USB» (не обязательно).**

Цель: проверка работы интерфейса USB.

- подача питания и старт МК.
- Начальная инициализация портов.
- Инициализация интерфейса USB как виртуальный COM порт и включение красного светодиода.
- При подключении к шине USB и инициализации включить зеленый.
- При приеме любых данных от COM с таймаутом более 50 мс, выдать какой-нибудь ASCII ответ, типа «Hi, I am STM!CR+LF». При передаче включить синий светодиод на 50 мс.
- На ПК будет запущена терминалка, которая будет отсылать каждые 100-200 мс сообщение и ждать ответ. Отсутствие ответа будет маркером для лога USB.

#### **- Режим 7 (000) «Тестирование бысродействия» (не обязательно).**

Цель: узнать реальное быстродействие МК.

- Подача питания и старт МК.
- Начальная инициализация портов.
- Создать на основе платформы «card-reader» для FLASH и SD карты.
- Если карта не вставлена (на платформе есть ключ), в качестве masstorage используется FLASH память. Если карта вставлена, соответственно карта, при этом питание с FLASH снимается.
- Если питание есть, а линка по USB нет, то светиться красный вне зависимости от выбранной памяти.
- Если есть подключение по USB и используется SD карта, светится зеленый, и если есть активность красные вспышки на 50-100 мс каждый 1 секунду. Если используется FLASH, вместо зеленого синий.
- На ПК будет запущен файловый менеджер со скриптом, который будет периодически записывать и читать тестовый файл, при этом записывая в лог скорость.