

643.ЭБУ10.00001-011301

УТВЕРЖДЕН
643.ЭБУ10.00001-011301–ЛУ

Министерство науки и высшего образования Российской Федерации
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«ЮЖНО-УРАЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
(НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)»

МИКРОКОНТРОЛЛЕР ЭЛЕКТРОННОГО БЛОКА УПРАВЛЕНИЯ

Описание программы

643.ЭБУ10.00001-011301

Листов 18

| | | | | |
|--------------|----------------|-------------|--------------|----------------|
| Инов.№ подл. | Подпись и дата | Взам. инв.№ | Инов.№ дубл. | Подпись и дата |
| | | | | |

2024

АННОТАЦИЯ

Настоящий документ является описанием программы микроконтроллера (далее – программа) электронного блока управления, управляющего электромеханическим преобразователем электрогидравлического усилителя мощности.

Структурно документ состоит из шести разделов. В первом разделе указаны обозначение и наименование программы, программное обеспечение, необходимое для функционирования программы, ее основные характеристики, языки программирования, на которых написана программа.

Во втором разделе указаны классы решаемых задач, назначение программы и сведения о функциональных ограничениях на применение.

В третьем разделе описан алгоритм программы, используемые методы, структура программы с описанием функций составных частей и связи между ними.

В четвертом разделе указаны типы электронно-вычислительных машин и устройств, которые используются при работе программы.

В пятом разделе указан способ вызова программы с носителя данных, на котором она расположена, входные точки в программу, сведения об использовании оперативной памяти.

В шестом разделе указаны сведения о характере, организации и предварительная подготовке входных и выходных данных, а также сведения о формате и способах кодирования данных.

СОДЕРЖАНИЕ

| | | |
|---|-----------------------------------|----|
| 1 | Общие сведения | 4 |
| 2 | Функциональное назначение | 5 |
| 3 | Описание логической структуры | 6 |
| 4 | Используемые технические средства | 13 |
| 5 | Вызов и загрузка | 14 |
| 6 | Входные и выходные данные | 15 |

1 ОБЩИЕ СВЕДЕНИЯ

1.1 Программа микроконтроллера электронного блока управления (далее – Программа) обозначается согласно документу, содержащему текст программы – 643.ЭБУ10.00001-011201.

1.2 Программа содержит все функциональные модули, необходимые для её функционирования и не требует для работы дополнительного программного обеспечения.

1.3 Основные характеристики программы.

1.3.1 Необходимый объем flash ПЗУ – не менее 256 кбайт.

1.3.2 Необходимый объем ОЗУ – не менее 128 кбайт.

1.4 В соответствии с требованиями п.п.3.19 ЧТЗ на ЭБУ, программа написана на машинно-ориентированным языком программирования C/C++ с закрытым кодом алгоритма.

2 ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ

2.1 Программа предназначена для реализации функций управления, заданных требованиями к реализации комплексного проекта (далее – ТРКП) и требованиями частного технического задания (далее – ЧТЗ) на электронный блок управления (далее – ЭБУ).

2.2 Функционирование программы ограничено возможностью работы только в составе микроконтроллера типа АРМ32F407VGT6, входящего в состав ЭБУ. Требуемый объем ОЗУ и flash ПЗУ указаны в п.п. 1.3.2.

3 ОПИСАНИЕ ЛОГИЧЕСКОЙ СТРУКТУРЫ

3.1 Алгоритм программы разработан с учетом выполнения требований п.п. 3.1.1 и 3.1.3 ЧТЗ на ЭБУ и заключается в формировании импульсных сигналов управления для полупроводниковых переключающих элементов, обеспечивающих поддержание положения поршня-штока гидроцилиндра, или (в зависимости от заданного режима работы) поддержание положения золотника высокоточной золотниковой пары, соответствующем положению, заданному аналоговым сигналом на входе задания положения, или цифровым кодом, заданным по интерфейсу USB или интерфейсу Ethernet. Для поддержания заданного положения поршня-штока или золотника, используется информация о текущем положении соответствующего элемента, получаемая с помощью обработки сигналов датчиков положения, являющихся линейными дифференциальными трансформаторами напряжения (далее – ЛДТ). В процессе выполнения, алгоритм Программы обеспечивает также формирование импульсного сигнала управления напряжением питания электромеханического преобразователя (далее – ЭМП), входящего в состав электрогидравлического усилителя мощности (далее – ЭГУМ). Для выполнения требований п.п. 3.18 ЧТЗ на ЭБУ, алгоритмы программы при обработке процессов аналого-цифрового и цифро-аналогового преобразования обеспечивает поддержку ввода и вывода аналоговых сигналов в диапазонах:

- по току: от -20 мА до $+20$ мА; от 4 мА до 20 мА;
- по напряжению: от -10 В до $+10$ В; от 0 до 10 В; от 0 до 5 В.

3.2 Используемый метод управления заключается в реализации алгоритма системы управления с обратной связью на основе пропорционального интегрально-дифференциального (ПИД) регулятора.

Структурная схема регулятора приведена на рисунке 1.

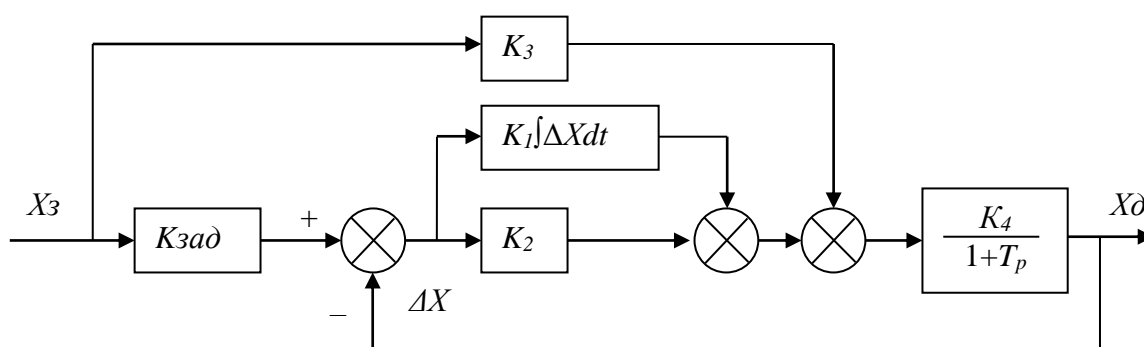


Рисунок 1

Сигнал задания положения $X_з$, задаваемый одним из указанных выше способов, сравнивается с положением, получаемым по датчику $X_д$.

Положение $X_д$ вычисляется в результате обработки сигнала, измеряемого с помощью датчика положения соответствующей координаты, с применением интегрального метода вычисления действующего значения напряжения на вторичной обмотке ЛДТ с усреднением на периоде сигнала возбуждения, подаваемого на первичную обмотку ЛДТ.

На основе получаемой ошибки управления ΔX формируется сигнал управления на ЭМП путем формирования последовательности импульсов с регулируемой скважностью (ШИМ).

Коэффициенты регулятора ($K_1 \dots K_4$) настраиваются в ходе отладки программы и сохраняются в энергонезависимой памяти микроконтроллера ЭБУ.

3.3 Структурно программа состоит из следующих программных модулей:

- модуль инициализации (файл Init.c);
- модуль обработки сигналов и расчета управления (файл arm32f4_dsp.c);
- модуль информационного обмена (файл Modbus.c);
- модуль объявления данных (файл data.c);
- исполняемый модуль (файл main.c).

3.4 В модуле инициализации описаны процедуры, выполняющие начальную настройку функциональных узлов микроконтроллера при старте программы.

Процедура `RCC_Config()` выполняет подключение тактирующих сигналов, получаемых с помощью узла формирования тактовых импульсов с автоматической подстройкой частоты и фазы, к следующим периферийным узлам микроконтроллера:

- порты ввода вывода `GPIOA`, `GPIOB`, `GPIOC`, `GPIOD`, `GPIOE`;
- контроллеры прямого доступа у памяти `DMA1`, `DMA2`;
- блок расчета контрольной суммы `CRC`;
- интерфейс `USB`;
- таймеры-счетчики `TIM1`, `TIM2`, `TIM3`, `TIM4`;
- интерфейсы обмена данными `SPI1`, `SPI3`, `USART1`, `USART2`;
- аналого-цифровые преобразователи `ADC1`, `ADC2`, `ADC3`;
- цифро-аналоговый преобразователь `DAC`;
- системный интерфейс конфигурирования `SYS_CFG`.

Процедура `GPIO_Config()` выполняет инициализацию режима работы портов ввода-вывода микроконтроллера. Для каждого вывода микросхемы `DD1`, в зависимости от его назначения, устанавливается один из следующих режимов работы:

- дискретный вход;
- дискретный выход;
- альтернативный режим (для подключения к внутренним периферийным устройствам микроконтроллера).

Для выводов в режиме дискретного входа или в режиме входа периферийных устройств дополнительно выбирается подключение или отключение встроенных резисторов, подключенных к напряжению питания или к общей цепи микроконтроллера, что позволяет повысить помехоустойчивость входных линий.

Для выводов в режиме дискретного выхода или в режиме выхода периферийных устройств дополнительно выбирается скорость нарастания и спада выходного напряжения, что определяет максимальную частоту выходного сигнала:

10, 50 или 100 МГц. Кроме того, выбирается тип выходного каскада: с открытым стоком (обеспечивает только замыкание вывода на общую цепь), или двухтактный, обеспечивающий замыкание как на общую цепь, так и на цепь питания микроконтроллера.

Процедура TIM_Config() выполняет установку режимов работы таймеров-счетчиков TIM1, TIM2. Для каждого таймера устанавливаются следующие параметры времязадающих регистров:

- коэффициент предварительного делителя тактовых импульсов;
- максимальное значение счетного регистра;
- режим работы счетного регистра.

Далее устанавливаются режимы работы цифровых компараторов:

- величина регистров сравнения;
- разрешение работы и режимы работы выходных каналов сравнения;

После завершения вышеописанных настроек выполняется команда, разрешающая работу каждого таймера-счетчика.

Процедура ADC_Config() выполняет настройку режимов работы встроенных АЦП ADC1, ADC2, ADC3 микроконтроллера DD1. Для получения высокой скорости обработки информации, поступающие от АЦП данные копируются в оперативную память с помощью устройства прямого доступа у памяти DMA0, настройка которого выполняется в первую очередь. Устанавливаются следующие параметры DMA0:

- номер канала;
- адрес и размер буфера в памяти для отсчетов, выполняемых АЦП;
- направление передачи данных;
- автоматический инкремент адреса;
- разрядность передаваемых данных;
- циклический режим передачи;

- порог для определения заполнения буфера;
- разрешение прерываний при заполнении буфера;
- разрешение работы выбранного канала DMA0.

Далее выполняется настройка общего узла АЦП, устанавливаются следующие общие параметры:

- режим одновременной работы;
- дополнительная задержка выборки и запоминания сигнала;
- режим доступа к устройству DMA;
- коэффициент деления тактовой частоты.

Далее для каждого из трех АЦП выполняется:

- конфигурирование входных каналов (очередность и время выборки);
- установка разрешения (12 двоичных разрядов);
- режим непрерывного последовательного сканирования каналов;
- выбор TIM2 в качестве триггера для запуска преобразования;
- выравнивание данных по младшему разряду;
- количество преобразований до ожидания триггера.

Далее подключается встроенный в микроконтроллер датчик температуры (на отдельный канал АЦП) и производится запуск всех трех АЦП.

Процедура DAC_Config() предназначена для настройки работы ЦАП, обеспечивающего формирование синусоидального напряжения для питания датчиков положения и вывод сигнала на аналоговые выходы. Для вывода синусоидального напряжения в память помещаются масштабированные значения синусоидальной функции на одном периоде, которые затем выводятся в регистр данных ЦАП с синхронизацией по таймеру-счетчику TIM2. Таким образом обеспечивается также синхронизация отсчетов АЦП, поскольку в алгоритме управления необходимо рассчитывать и усреднять данные на периоде сигнала возбуждения датчиков положения.

3.5 В модуле обработки сигналов и расчета управления определены переменные и буферы в памяти для хранения обрабатываемых данных, а также

описаны процедуры и функции, непосредственно реализующие алгоритмы управления ЭМП. Функция `LP_Filter(float* z, float x)` предназначена для рекурсивного вычисления текущего выходного значения биквадратного фильтра низких частот 4-го порядка с частотой среза 10 Гц, применяемого для дополнительного усреднения получаемых данных. Аргументами функции являются указатель на массив результатов предыдущих расчетов функции (`float* z`) и текущее входное значение (`float x`).

Функция `BP_Filter(float* z, float x)` предназначена для рекурсивного вычисления текущего выходного значения биквадратного полосового фильтра 4-го порядка с центральной частотой 3000 Гц, применяемого для удаления постоянной составляющей и фильтрации сигналов, поступающих со вторичных обмоток ЛДТ. Аргументами функции являются указатель на массив результатов предыдущих расчетов функции (`float* z`) и текущее входное значение (`float x`).

Процедура `ADC_OnGetData(void* buf)` выполняет обработку данных в памяти, полученных в результате оцифровки с помощью АЦП сигналов со вторичных обмоток датчиков положения, сигналов, поступающих на аналоговые входы, а также напряжений питания и температуры встроенного датчика микроконтроллера. Здесь производится вычисление положения поршня-штока гидроцилиндра и положения золотника ВЗП, а затем, в зависимости от сигнала на дискретном входе выбора датчика положения, по алгоритму, рассмотренному в п.п. 3.2, производится расчет напряжения, которое устанавливается на ЭМП. Эта процедура вызывается из процедуры `DMA2_Stream0_IRQHandler()`, выполняемой при обработке прерывания, поступающего от устройства прямого доступа к памяти DMA2.

3.6 В модуле информационного обмена описаны процедуры и функции, обеспечивающие обмен по протоколам Modbus RTU/UDP/TCP через интерфейсы USB и Ethernet. Функция `CRC16(uint8_t* Buf, uint16_t Count)` предназначена для расчета контрольной суммы принимаемого и передаваемого пакета данных. Аргументами функции являются указатель на адрес начала пакета данных (`uint8_t* Buf`) и длина этого пакета в байтах (`uint16_t Count`). Основной алгоритм обработки

принимаемых пакетов данных реализован с помощью функции `Modbus_GetResponse(uint8_t* pBuf, uint16_t CharCount)`. На входе функция принимает указатель на адрес начала пакета данных (`uint8_t* pBuf`) и длину пакета данных в байтах (`uint16_t CharCount`). Данные для ответного пакета формируются в том же буфере, что и входные данные. Длина выходного пакета данных возвращается в качестве результата функции. Для доступа к Modbus-регистрам в соответствии с картой регистров ЭБУ, используется функция `ECU_AccessRegister(uint16_t addr, int16_t val, charmode)`, аргументами которой являются адрес регистра (`uint16_t addr`), записываемое значение (`int16_t val`) и режим доступа (чтение или запись). Перед выполнением операций чтения или записи производится проверка текущего уровня доступа, записанного в Modbus-регистре 13 и в случае, если текущий уровень меньше требуемого, операция записи игнорируется, а операция чтения возвращает отрицательное число, что является признаком ошибки доступа.

3.7 В модуле объявления данных объявлены и заполнены соответствующими начальными значениями следующие переменные структурного типа, предназначенные для организации доступа к Modbus-регистрам из Программы:

- `ECU_ID_data` – параметры идентификации и безопасности;
- `ECU_control_params` – параметры управления;
- `ECU_reg_params` – параметры регуляторов;
- `ECU_state_data` – параметры текущего состояния;
- `ECU_ref_control` – конфигурация входов задания положения;
- `ECU_output_control` – конфигурация аналоговых выходов;
- `ECU_dither_control` – конфигурация шумовой добавки;
- `ECU_statistics` – доступ к данным статистики;
- `ECU_raw_data` – исходные данные для калибровки;
- `ECU_calib_values` – калибровочные коэффициенты;
- `ECU_internal_params` – параметры управления, скрытые от пользователя;

- ECU_debug_data – отладочные данные;
- ECU_archive_data – доступ к архиву;
- ECU_scope_control – параметры записи осциллограмм.

3.8 В **исполняемом модуле** Программы описана процедура main(), запускаемая при старте Программы и выполняющая вызовы всех остальных процедур и функций, описанных выше в настоящем документе.

4 ИСПОЛЬЗУЕМЫЕ ТЕХНИЧЕСКИЕ СРЕДСТВА

4.1 Для запуска и выполнения Программы используется однокристалльная микро-ЭВМ (микроконтроллер) типа ARM32F407VGT6, имеющая приведенные далее технические характеристики.

Корпус: LQFP100.

Тактовая частота: 168 МГц.

Объем Flash-памяти: 1 МБ.

Объем SRAM-памяти: 192 КБ.

Портов ввода-вывода всего: 80.

Портов ввода-вывода, устойчивых к напряжению 5 В: 78.

АЦП: 3× 12 битный на 24 канала.

ЦАП: 2× 12 битный.

Таймеры-счетчики с функцией ШИМ: 17.

Аппаратные интерфейсы: 3× SPI, 3× I²C, 6× UART, 2× CAN, 1× SDIO.

Диапазон напряжения питания: 1,8...3,6 В.

Максимальный ток вывода (втекающий или вытекающий): 25 мА.

Другие технические средства для запуска и работы программы не требуются.

5 ВЫЗОВ И ЗАГРУЗКА ПРОГРАММЫ

5.1 Программа расположена в энергонезависимой памяти программ микроконтроллера (п.п. 4.1). Объем программы 223 кБ.

5.2 Запуск Программы происходит автоматически при включении питания или при аппаратном сбросе микроконтроллера.

5.3 Точка входа в программу расположена по адресу 0x8000000.

5.4 В процессе работы Программе требуется не менее 128 кБ памяти для хранения данных.

6 ВХОДНЫЕ И ВЫХОДНЫЕ ДАННЫЕ

6.1 Сведения о входных данных Программы.

6.1.1 Дискретный входной сигнал выбора датчика положения. Сигнал поступает на порт ввода-вывода микроконтроллера. Разрешение – 1 бит.

6.1.2 Дискретный входной сигнал разрешения работы. Сигнал поступает на порт ввода-вывода микроконтроллера. Разрешение – 1 бит.

6.1.3 Дискретный входной сигнал сброса ошибок. Сигнал поступает на порт ввода-вывода микроконтроллера. Разрешение – 1 бит.

6.1.4 Аналоговый входной сигнал датчика положения. Сигнал поступает на вход встроенного АЦП микроконтроллера. Разрешение – 12 бит.

6.1.5 Аналоговый входной сигнал задания положения. Сигнал поступает на вход встроенного АЦП микроконтроллера. Разрешение – 12 бит.

6.1.6 Аналоговый входной сигнал контроля напряжения питания ЭБУ. Сигнал поступает на вход встроенного АЦП микроконтроллера. Разрешение – 12 бит.

6.1.7 Аналоговый входной сигнал контроля напряжения питания датчика тока в обмотке якоря ЭМП. Сигнал поступает на вход встроенного АЦП микроконтроллера. Разрешение – 12 бит.

6.1.8 Команды управления, поступающие по интерфейсу USB. Данные записываются в соответствующие Modbus-регистры. Разрешение данных может составлять от 1 до 32 бит. Тип данных: целое число, число с плавающей запятой.

6.1.9 Команды управления, поступающие по интерфейсу Ethernet. Данные записываются в соответствующие Modbus-регистры. Разрешение данных может составлять от 1 до 32 бит. Тип данных: целое число, число с плавающей запятой.

6.1.10 Способ кодирования целых чисел– дополнительный код.

6.1.11 Способ кодирования чисел с плавающей запятой определяется стандартом IEEE754–2008.

6.2 Сведения о выходных данных

6.2.1 Импульсные сигналы управления транзисторами мостовой схемы питания обмотки якоря ЭМП. Разрешение – 1 бит. Всего 4 сигнала.

6.2.2 Импульсные сигналы управления транзисторами преобразователя напряжения для питания обмотки якоря ЭМП. Разрешение – 1 бит. Всего 2 сигнала.

6.2.3 Сигнал индикации процесса позиционирования. Сигнал выводится на порт дискретного ввода-вывода микроконтроллера. Разрешение – 1 бит.

6.2.3 Сигнал индикации завершения позиционирования. Сигнал выводится на порт дискретного ввода-вывода микроконтроллера. Разрешение – 1 бит.

6.2.3 Сигнал индикации ошибки функционирования. Сигнал выводится на порт дискретного ввода-вывода микроконтроллера. Разрешение – 1 бит.

6.2.4 Цифровые значения аналоговых сигналов управления и мониторинга состояния. Аналоговые сигналы выводятся с помощью встроенного цифро-аналогового преобразователя микроконтроллера. Разрешение – 16 бит. Всего 3 числовых значения.

6.2.5 Числовые значения Modbus-регистров, передаваемые по интерфейсу USB. Данные передаются в составе ответного пакета при запросе чтения содержимого Modbus-регистров. Разрешение данных может составлять от 1 до 32 бит. Тип данных: целое число, число с плавающей запятой.

6.2.6 Числовые значения Modbus-регистров, передаваемые по интерфейсу Ethernet. Данные передаются в составе ответного пакета при запросе чтения содержимого Modbus-регистров. Разрешение данных может составлять от 1 до 32 бит. Тип данных: целое число, число с плавающей запятой.

6.2.7 Способ кодирования целых чисел– дополнительный код.

6.2.8 Способ кодирования чисел с плавающей запятой определяется стандартом IEEE754–2008.

