

**ИНСТРУКЦИЯ ПО УСТАНОВКЕ
ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ
ПМУ
643.ГЖИК.00020**

АННОТАЦИЯ

Настоящий документ представляет инструкцию по установке программного обеспечения ПМУ 643.ГЖИК.00020, предназначенного для платы пульта местного управления (ПМУ) ГЖИК.301411.111. В документе приведена структура проекта программы. Описана методика сборки, загрузки, настройки и проверки исправности программы.

СОДЕРЖАНИЕ

1. Общие сведения о программе	4
1.1 Технические требования	4
1.2 Содержимое диска	5
1.3 Подготовка технических средств к работе.....	6
1.4 Подготовка программных средств к работе.....	9
2. Структура программы.....	12
3. Настройка программы.....	14
3.1 Базовые операции	14
3.2 Примеры типового использования	23
4. Проверка программы.....	27
4.1 Визуальный контроль.....	27
4.2 Проверка с использованием управляющего интерфейса.....	27
5. Дополнительные возможности.....	29
6. Сообщения.....	31
6.1 Сообщения, выдаваемые при сборке программы.....	31
6.2 Сообщения, выдаваемые в ходе работы программы.....	31
Перечень терминов и сокращений	32

1. ОБЩИЕ СВЕДЕНИЯ О ПРОГРАММЕ

Данная программа предназначена для оперативного управления преобразователем частоты (ПЧ), отображения на дисплее его основных параметров и настройки (конфигурирования). Данный документ представляет собой инструкцию по установке ПО – набор инструкций по установке, настройке, запуску программы и т.п. Документ описывает программное обеспечение ПМУ в составе СУ ПЧ.

1.1 Технические требования

Технические и программные средства, необходимые для разработки, модификации, сборки, отладки, загрузки и проверки программы:

- ПК под управлением ОС «Windows» не ниже версии 7, удовлетворяющий следующим требованиям: центральный процессор AMD или Intel с тактовой частотой не ниже 1.5 ГГц, объём ОЗУ не ниже 1 Гб, объём свободного пространства на жёстком диске не ниже 1 Гб, цветной монитор, наличие как минимум одного порта USB;
- JTAG-эмулятор, удовлетворяющий требованиям, перечисленным в подразделе 1.3 данного руководства;
- адаптер USB-to-CAN, удовлетворяющий требованиям подраздела 1.3 данного руководства;
- среда разработки «Vector IDE» версии 1.3.3;
- программа управляющего интерфейса «OSCT» версии не ниже 5.8.0.

Задачи, решаемые при помощи этих средств, приведены в таблице 1 .

Таблица 1 – Применяемые технические и программные средства

Задача	Применяемые для решения задачи программные средства			Применяемые для решения задачи аппаратные средства		
	Vector IDE	OSCT	CANwise	JTAG-эмулятор	Адаптер USB-to-CAN	USB
Разработка, редактирование исходных текстов программы	*					
Сборка выходного файла программы	*					
Загрузка программы	Первичная	*		*		
	Последующие	*	*	*	*	
	Через интерфейс CAN		*		*	
Отладка	JTAG	*		*		
	CAN (отладка в реальном времени)		*		*	
	USB		*			*
Осциллографирование		*			*	*
Ведение лог файлов сетевого обмена		*			*	*
Просмотр журналов аварий/событий		*			*	*
Параметрирование		*			*	*
Активация встроенного загрузчика		*	*		*	

1.2 Содержимое диска

Установочные файлы программных средств, драйверов технических средств и руководства пользователей к ним находятся на компакт-диске. Структура каталогов диска приведена в таблице 2 .

Таблица 2 – Структура каталогов компакт-диска с ПО

Имя каталога	Имя файла	Описание
CANwise	canwise-3.10-win32.exe	Установочный файл программы «CANwise»

Имя каталога	Имя файла	Описание
	CANwise руководство пользователя.pdf	Руководство пользователя программы «CANwise»
Marathon CAN-bus-USBnp	chai-2.6.0-x86-xp-vista-win7.exe	Установочный файл драйвера адаптера «Marathon USB-bus-CANnp» для 32-разрядных систем под управлением ОС «Windows» 7 / 8 / 10
	chai-2.11.2-x64-win7-8-10.exe	
	chai-2.14.0-x64-win81-10.exe	
	Руководство пользователя Marathon CAN-bus-USBnp.pdf	Руководство пользователя адаптера «Marathon USB-bus-CANnp»
Projects	KEAZ_PU.zip	Архив проекта программного обеспечения ПМУ
Projects\Outputs	KEAZ_PU.elf	Выходной файл проекта
ST-Link V2	stlinkv2_driver.zip	Архив с установочными файлами драйвера JTAG-эмулятора «ST-Link\V2»
OSCT	OSCT_CD v5.8.1.zip	Архив программы «OSCT»
	OSCT руководство пользователя.pdf	Руководство пользователя программы «OSCT»
	OSCT руководство пользователя.docx	
COODEdit4	COODEdit4_KEAZ.zip	Архив программы «COODEdit4»
	COODEdit руководство пользователя.pdf	Руководство пользователя программы «COODEdit4»
Vector IDE	VectorIDEv1.3.3_setup.exe	Установочный файл среды разработки «Vector IDE»
	VectorIDE руководство пользователя.pdf	Руководство пользователя среды разработки «Vector IDE»

1.3 Подготовка технических средств к работе

Требования к техническим средствам приведены в таблице 3

Таблица 3 – Необходимые технические средства

Тип устройства	JTAG-эмулятор	Адаптер USB-to-CAN
Производитель	STMicroelectronics	Marathon
Модель	ST-LINK/V2	CAN-bus-USBnp
Поддерживаемая версия ОС	Windows 7 / Vista / 8 / 10 / 11	Windows 7 / Vista / 8 / 10 / 11
Сайт производителя	https://st.com	http://can.marathon.ru/

Тип устройства	JTAG-эмулятор	Адаптер USB-to-CAN
Краткое описание	ST-LINK/V2 является внутрисхемным отладчиком/программатором для семейства микроконтроллеров на базе ядра ARM Cortex-M4F.	Устройство, подключаемое к ПК через USB интерфейс, предназначенное для использования в качестве контроллера распределенной системы сбора данных и управления на основе сети CAN-bus.

1.3.1 Установка драйверов JTAG-эмулятора

Сначала необходимо установить оригинальный драйвер «stlinkv2_driver.zip» из каталога «ST-Link V2» компакт-диска. Позже, при установке среды разработки «Vector IDE» (п.1.4.1), этот драйвер будет заменён на совместимую версию.

Более подробно процесс установки описан в документе «VectorIDE руководство пользователя.pdf» в разделе 2.

1.3.2 Установка драйверов адаптера USB-to-CAN

Установочные файлы драйверов для адаптеров USB-to-CAN «Marathon CAN-bus-USBnp» расположены в каталоге «Marathon CAN-bus-USBnp» компакт-диска с ПО.

Руководство пользователя адаптера USB-to-CAN «Marathon CAN-bus-USBnp» расположено в каталоге «Marathon CAN-bus-USBnp» компакт-диска и называется «Руководство пользователя Marathon CAN-bus-USBnp.pdf».

Для установки драйверов запустить один из файлов, находящихся внутри каталога:

- для 32-разрядной ОС «Windows» версий XP/Vista/7 необходимо запустить файл «chai-2.6.0-x86-xp-vista-win7.exe»;
- для 64-разрядной ОС «Windows» версий 7 и выше необходимо запустить файл «chai-2.11.2-x64-win7-8-10.exe»;
- для 64-разрядной ОС «Windows» версий 8.1 и выше необходимо запустить файл «chai-2.14.0-x64-win81-10.exe».

В процессе установки необходимо следовать указаниям программы-установщика. Не рекомендуется менять предлагаемый по умолчанию путь установки.

После этого может потребоваться вручную указать расположение драйверов при первом подключении адаптера USB-to-CAN «CAN-bus-USBnp». Для этого нужно выполнить следующие шаги:

- 1) подключить адаптер USB-to-CAN к USB-порту ПК;
- 2) открыть диспетчер устройств Windows;
- 3) проверить наличие двух устройств «CAN-bus-USBnp interface» в разделах «Другие устройства» и «Контроллеры USB». Если два устройства «CAN-bus-USBnp interface» находятся в разделе «Контроллеры USB», то означает, что драйвера успешно установлены. Если устройств находятся в разделе «Другие устройства», то необходимо перейти к следующему шагу;
- 4) нажав правой кнопкой на одном из устройств «CAN-bus-USBnp interface», выбрать пункт «Обновить драйверы»;
- 5) в открывшемся окне выбрать вариант «Выполнить поиск драйверов на этом компьютере»;
- 6) нажать кнопку «Обзор» и указать путь до каталога, куда были установлены драйвера и библиотека «СНАИ». По умолчанию это каталог «C:\Program Files (x86)\СНАИ-xxxx», где «xxxx» - версия драйверов;
- 7) нажать «Далее» и дождаться окончания процесса установки драйверов, после чего нажать кнопку «Закреть»;
- 8) повторить шаги 4-7 для второго устройства.

1.4 Подготовка программных средств к работе

Требования к программным средствам приведены в таблице 4 .

Таблица 4 – Требуемые программные средства

Название	«Vector IDE»	«OSCT»	CANwise
Версия	1.3.3	5.8.1	3.10
Поддерживаемая версия ОС	Windows 7/Vista/8/10/11	Windows 7/Vista/8/10/11	Windows 7/Vista/8/10/11
Сайт производителя	https://motorcontrol.ru		http://can.marathon.ru/
Краткое описание	Интегрированная среда разработки программного обеспечения для встраиваемых систем. Среда сконфигурирована для микроконтроллеров серии K1921BK фирмы АО «НИИЭТ», однако при желании можно разрабатывать и отлаживать проекты для других контроллеров на ядре ARM.	Набор программных средств, позволяющих осуществлять мониторинг и параметрирование CANopen-совместимых устройств посредством персонального компьютера (ПК), подключаемого через физический интерфейс CAN посредством переходника USB-to-CAN.	Универсальное приложение тестирования и конфигурирования сети CAN, с графическим пользовательским интерфейсом. Поставляется с плагином монитора сети CANmonitor, который позволяет работать с CAN на канальном уровне.

1.4.1 Установка среды разработки «Vector IDE»

Установочный файл программы «Vector IDE» расположен в каталоге «Vector IDE» компакт-диска с программным обеспечением и называется «VectorIDEv1.3.3_setup.exe».

Руководство пользователя программы «Vector IDE» расположено в каталоге «Vector IDE» компакт-диска и называется «VectorIDE руководство пользователя.pdf».

Для установки программы следует запустить файл «VectorIDEv1.3.3_setup.exe» и следовать указаниям. Важно: в пути установки не должно быть круглых скобок, поэтому недопустимо устанавливать среду разработки внутри каталога «C:/Program Files (x86)». Во время установки программы будет предложено выбрать JTAG-эмулятор.

По завершении установки будет предложено запустить программу «Zadig», которая осуществляет замену оригинальных драйверов JTAG-эмуляторов «J-Link» и «ST-Link V2» на совместимые с «Vector IDE». Рекомендуется сначала установить оригинальный драйвер «stlinkv2_driver.zip» из каталога «ST-Link V2» компакт-диска, а уже потом запустить программу «Zadig».

Более подробно процесс установки описан в документе «VectorIDE руководство пользователя.pdf» в разделе 2.

1.4.2 Установка программы «OSCT»

Архив с программой «OSCT» расположен в каталоге «OSCT» оптического диска с ПО и называется «OSCT_CD v5.8.1.zip».

Руководство пользователя программы «OSCT» расположено в каталоге «OSCT» компакт-диска и называется «OSCT руководство пользователя.pdf».

Программа «OSCT» не требует установки. Достаточно распаковать архив в удобный для пользователя каталог. О дополнительных настройках можно прочитать в «OSCT руководство пользователя.pdf» в разделе 5.

1.4.3 Установка программы «COOEdit4»

Архив с программой «COOEdit4» расположен в каталоге «COOEdit4» оптического диска с ПО и называется «COOEdit4_KEAZ.zip».

Руководство пользователя программы «COOEdit4» расположено в каталоге «COOEdit4» компакт-диска и называется «COOEdit руководство пользователя.pdf».

Программа «COOEdit4» не требует установки. Достаточно распаковать архив в удобный для пользователя каталог.

1.4.4 Установка программы CANwise

Установочный файл программы «CANwise» расположен в каталоге «CANwise» компакт-диска с ПО и называется «canwise-3.10-win32.exe».

Для установки программы «CANwise» необходимо запустить установочный файл и следовать указаниям.

Руководство пользователя программы «CANwise» расположено в каталоге «CANwise» компакт-диска и называется «CANwise руководство пользователя.pdf».

2. СТРУКТУРА ПРОГРАММЫ

Архив с проектом программного обеспечения ПМУ «KEAZ_PU.zip» расположен в каталоге «Projects» оптического диска с ПО.

Проект программы разработан в виде логических модулей, каждый из которых выполняет определённую функцию. Логика работы каждого модуля описана в соответствующем файле исходного кода (расширение «.c»); интерфейс модуля описан в соответствующем заголовочном файле (расширение «.h»).

Для удобства ориентирования в структуре проекта, файлы, участвующие в сборке программы, распределены по нескольким подкаталогам, в зависимости от их назначения. В таблице 5 приведён список подкаталогов.

Таблица 5 – Каталоги и файлы корневой директории проекта

Содержимое архива с проектом	Комментарий
.settings/..	Подкаталог, содержащий языковые настройки проекта программы
cmd/..	Подкаталог, содержащий файлы компоновки памяти
CMSIS/..	Заголовочные файлы Cortex Microcontroller Software Interface Standard – содержит описание всех регистров микроконтроллера, таблицу векторов прерываний и некоторый стартовый код, который выполняется перед передачей управления функции main()
Images/..	Графические файлы с «иконками», а также python-скрипт для их преобразования в битовые массивы на языке C
include/..	Подкаталог, содержащий стандартные заголовочные файлы
Profiles/..	Подкаталог с профилем словаря объектов CANopen, используемый при работе с программой-редактором словаря «COOEdit4»
src/..	Подкаталог, содержащий стандартные файлы исходного кода, отвечающие за инициализацию целевого микроконтроллера
USB_cdc	Подкаталог, содержащий ядро драйвера USB
Vinclude/..	Подкаталог, содержащий заголовочные файлы, разработанные исполнителем проекта
Vsrc/..	Подкаталог, содержащий файлы исходного кода, разработанные исполнителем проекта
startup_gd32f405.S	Стартовый код, выполняемый после сброса процессора
KEAZ_PU Debug.launch	Файл для запуска отладочной сессии проекта
.cproject	Настройки для компиляции проекта программы
.project	Файл проекта для среды разработки «Vector IDE»

Файлы «.sproject», «.project» и подкаталог «.settings/», расположенные в корневом каталоге проекта, содержат в себе различные настройки проекта и информацию, необходимую для импортирования проекта в среду разработки «Vector IDE». Эти файлы автоматически генерируются средой разработки.

3. НАСТРОЙКА ПРОГРАММЫ

При выполнении типовых задач при работе с ПМУ, таких как первоначальная настройка, обновление программы и других, требуется в определённой последовательности выполнять различные базовые операции, которые подробно описаны в подразделе 3.1.

3.1 Базовые операции

3.1.1 Подключение цифрового питания 24 В к ПМУ

Подключение напряжения питания 24 В обеспечивает работу цифровых цепей, включая работу микроконтроллера. Подключение данного напряжения необходимо для любых взаимодействий с СУ ПЧ.

Источник питания с постоянным напряжением 24 В должен иметь мощность не менее 2 Вт. Подача питания осуществляется через разъём «X2» ПМУ. Описание сигналов разъёма «X2» приведено на схеме электрической принципиальной ГЖИК.301411.111 ЭЗ.

Если ПМУ установлен в ПЧ, то питание на ПМУ поступает со стороны базовой платы управления ГЖИК.301411.109.

3.1.2 Подключение JTAG-эмулятора к ПМУ

При помощи JTAG-эмулятора осуществляется загрузка программы в память микроконтроллера и её отладка. Подключение JTAG-эмулятора необходимо при загрузке программы средствами среды разработки «Vector IDE».

Подключение ПМУ к JTAG-эмулятору «ST-LINK/V2» осуществляется штатным 20-контактным кабелем: один конец кабеля подключается к разъёму «STM32» эмулятора, другой – к разъёму «X1» платы ПМУ.

3.1.3 Подключение адаптера USB-to-CAN к ПМУ

Интерфейс связи CAN с протоколом CANopen используется для взаимодействия пользователя с программой, загруженной в память микроконтроллера, посредством управляющего интерфейса «OSCT».

Для подключения адаптера необходимо соединить сигналы «CAN1_HIGH» и «CAN1_LOW» 9-контактного разъёма SUB-D9F адаптера «CAN-bus-USBnp» с сигналами «CANH» и «CANL» разъёма «X2» ПМУ соответственно.

Описание сигналов разъёма «X2» платы ПМУ приведено на схеме электрической принципиальной ГЖИК.301411.111 ЭЗ.

Описание сигналов 9-контактного разъёма SUB-D9F адаптера «CAN-bus-USBnp» приведено в руководстве пользователя адаптера USB-to-CAN «Marathon CAN-bus-USBnp».

Если ПМУ установлен в ПЧ, то адаптер USB-to-CAN подключается к сервисному разъёму «X18» базовой платы, описание сигналов которого приведено на схеме электрической принципиальной ГЖИК.301411.109 ЭЗ.

3.1.4 Подключение ПК к ПМУ по интерфейсу USB

Если возможность подключения адаптера USB-to-CAN к ПМУ (п.3.1.3) отсутствует, можно подключить ПК к ПМУ непосредственно по интерфейсу USB, с помощью стандартного USB-кабеля типа «mini USB». Кабель подключается к разъёму «X3» платы ПМУ. При таком подключении эмулируется работа интерфейса CAN с протоколом CANopen, за исключением функции загрузки ПО.

3.1.5 Установка соединения между ПК и ПМУ

Перед установкой соединения необходимо подключить ПК к ПМУ либо через адаптер USB-to-CAN (см. п.3.1.3), либо напрямую через интерфейс USB (см. п.3.1.4).

Для установки соединения необходимо выполнить следующие шаги:

- если подключение осуществляется через адаптер USB-to-CAN, то его необходимо подключить к USB-порту ПК штатным USB-кабелем «Type A»;
- убедиться, что на ПМУ подано питание;
- запустить программу «OSCT»;
- если подключение осуществляется через адаптер USB-to-CAN, то в настройках интерфейса связи необходимо выбрать модуль связи «MARATHON» и скорость 250 kBit/s (см. «OSCT руководство пользователя.pdf», раздел 7.2);
- если ПМУ подключен напрямую к ПК через USB-интерфейс, то необходимо выбрать модуль связи «USB RS» (см. «OSCT руководство пользователя.pdf», раздел 7.2);
- подключиться к устройству, как это описано в разделе 6.1 «OSCT руководство пользователя.pdf»;
- если подключение к устройству выполняется впервые, дождаться окончания загрузки словаря объектов устройства.

3.1.6 Импорт проекта

Перед внесением изменений и сборкой проекта его необходимо импортировать в среду разработки «Vector IDE», если этого не было сделано ранее. Для этого необходимо выполнить следующие шаги:

- запустить среду разработки «Vector IDE»;
- открыть рабочую область, созданную в процессе установки «Vector IDE»;
- в меню среды «File» выбрать пункт «Import...»;
- в открывшемся окне раскрыть раздел «General», выбрать пункт «Existing Projects into Workspace» и нажать кнопку «Next >»;
- в открывшемся окне «Import» установить кнопку на вариант «Select archive file», справа от поля ввода нажать кнопку «Browse...» и указать путь до zip-архива с проектом;

- снять галочки «Close newly imported projects upon completion» и «Add project to working sets»;
- нажать кнопку «Finish».

После этих действий импортированный проект появится в окне «Project Explorer», в код проекта можно будет вносить изменения и производить сборку проекта.

3.1.7 Сборка проекта

Перед сборкой проекта, его необходимо импортировать в рабочую область среды разработки «Vector IDE», как описано в п. 3.1.6, если этого не было сделано ранее.

Чтобы запустить процесс сборки проекта, необходимо нажать на его имени в окне «Project Explorer» правой кнопкой мыши и выбрать пункт «Build Project».

Во время сборки откроется окно «Console», в которое будет выводиться информация о процессе сборки. При успешном завершении сборки в этом окне появится строка синего цвета с текстом «[HH:MM:SS] Build Finished. 0 errors, 0 warnings. (took [X_time])», где «HH:MM:SS» - время окончания сборки, «X_time» - длительность сборки.

Результатом сборки проекта является выходной файл программы с расширением «.elf», который сохраняется в каталоге «Debug» в проекте. Если после успешной сборки таковой файл отсутствует, необходимо обновить содержимое проекта, нажав на имени проекта в окне «Project Explorer» правой кнопкой мыши и выбрав пункт «Refresh».

3.1.8 Редактирование исходных файлов проекта программы

Изменения в файлы проекта необходимо вносить в среде разработки «Vector IDE». Перед внесением изменений в код проекта, его необходимо импортировать в среду, если этого не было сделано ранее. Для внесения изменений в файлы проекта необходимо выполнить следующие действия:

- в окне «Project Explorer» среды разработки «Vector IDE» дважды щёлкнуть на файл, в который планируется вносить изменения, после чего откроется окно редактирования файла;
- внести необходимые изменения;
- сохранить изменения, открыв меню «File» и выбрав пункт «Save».

В целях идентификации различных версий программного обеспечения после внесения всех желаемых изменений в файлы проекта, необходимо присвоить макроопределению «VERSION_SHORT_HASH» в файле «include/Version.h» уникальное (то есть такое, которое ранее не присваивалось этому макросу) значение перед запуском сборки. После завершения сборки рекомендуется добавить это значение к имени выходного файла программы. Также рекомендуется поддерживать в актуальном состоянии макроопределения VERSION_YEAR, VERSION_MONTH и VERSION_DAY, определяющие дату сборки ПО (она отображается в программе «OSCT»).

Рекомендуется хранить все выходные файлы, подлежащие загрузке в ПМУ в отдельном каталоге. В таком случае при замене ПМУ можно будет определить выходной файл нужной версии по его имени, сопоставив имя выходного файла со значением параметра «GIT_VersionShortHash» (группа «СЛУЖЕБНЫЕ») в словаре объектов устройства (см. п. 3.2.3).

3.1.9 Загрузка программы в память микроконтроллера с использованием JTAG-эмулятора через среду разработки «Vector IDE»

Если память микроконтроллера не содержит в себе ранее загруженной программы с корректно работающим алгоритмом обновления программы по интерфейсу CAN, то загрузить программу в память можно только при помощи JTAG-эмулятора. Сделать это можно средствами среды разработки «Vector IDE», а также при помощи специальных командных файлов. Способ загрузки программы через среду «Vector IDE» можно использовать в том случае, когда необходимо загрузить только что со-

бранный в открытом проекте выходной файл программы. Если же требуется загрузить один из ранее собранных или любой произвольный выходной файл, то следует использовать специальные командные файлы, работа с которыми описана в п.3.1.10.

Здесь рассмотрен способ загрузки выходного файла программы в память микроконтроллера через среду разработки «Vector IDE».

Перед загрузкой программа должна быть собрана, как описано в п.3.1.7.

Для этого необходимо выполнить следующие шаги:

- снять питание с ПМУ;
- подключить JTAG-адаптер к разъёму «X1» ПМУ;
- подключить JTAG-адаптер к USB-порту ПК;
- подать цифровое питание на ПМУ;
- запустить программу «Vector IDE»;
- в окне «Project Explorer» раскрыть содержимое проекта, щёлкнув левой кнопкой мыши по значку «>» слева от названия проекта;
- правой кнопкой мыши нажать на файл с расширением «KEAZ_PU Debug.launch» и в меню «Debug As...» выбрать пункт «KEAZ_PU Debug».

После этих действий будет запущена отладочная сессия, а среда разработки перейдёт в режим отладки «Debug». Во время отладочной сессии программа будет загружена в память микроконтроллера. По окончании загрузки, программа будет остановлена в начале функции «void main (void)». Если проводить отладку программы не требуется, то нужно нажать на символ красного квадрата «Stop» на панели инструментов в верхней части окна среды разработки.

После этого необходимо снять и снова подать питание на ПМУ, чтобы программа начала своё выполнение.

3.1.10 Загрузка произвольного выходного файла программы в память микроконтроллера с использованием JTAG-эмулятора

Если у программиста имеется ранее собранный выходной файл программы, то его можно загрузить при помощи специальных командных файлов, входящих в состав

«Vector IDE». Процесс загрузки произвольных выходных файлов подробно описан в разделе 3.8 «VectorIDE руководство пользователя.pdf».

3.1.11 Загрузка программы в память микроконтроллера с использованием адаптера USB-to-CAN

Если в память микроконтроллера уже загружена программа, которая корректно работает, но которую по тем или иным причинам необходимо обновить/заменить, то загрузить программу в память можно двумя способами. Первый способ – при помощи JTAG-эмулятора с использованием среды разработки «Vector IDE», как это описано в пунктах 3.1.9 и 3.1.10 данного руководства. Второй способ – при помощи адаптера USB-to-CAN с использованием программы «OSCT».

Перед загрузкой программа должна быть собрана, как описано в п. 3.1.7.

Для обновления программы средствами программы «OSCT» необходимо выполнить следующие шаги:

- установить соединение между ПК и ПМУ (см. п. 3.1.5);
- далее, следуя указаниям разделов 6.1 и 6.8 «OSCT руководство пользователя.pdf», загрузить программу в память контроллера.

Если во время загрузки программы произошли непредвиденные сбои (например, перебой в питании ПМУ), из-за которых установка соединения между ПК и ПМУ стала невозможна, то загрузить программу можно либо используя JTAG-эмулятор, как это описано в пунктах 3.1.9 и 3.1.10, либо воспользовавшись встроенным загрузчиком (п. 5).

3.1.12 Сохранение значений параметров словаря CANopen в энергонезависимую память ПМУ

В программе предусмотрена возможность сохранения значений словаря параметров CANopen в энергонезависимую память (EEPROM). После запуска программа проверяет наличие сохранённых в энергонезависимой памяти значений и, если они

найденны, загружает эти значения. Если сохранённых значений нет, то будут загружены значения по умолчанию, которые содержатся в исходном коде программы. При этом сохраняются значения только редактируемых параметров. Значения параметров, которые доступны только на чтение, сохранить нельзя.

Для сохранения параметров в энергонезависимую память необходимо выполнить следующие шаги:

- установить соединение между ПК и ПМУ (п. 3.1.5);
- если требуется, изменить значения необходимых параметров, как это описано в разделе 6.2 «OSCT руководство пользователя.pdf»;
- нажать на кнопку «Сохранить» сверху справа в основном окне программы «OSCT» и дождаться завершения операции (ход выполнения операции сохранения индицируется зелёной полосой под рядом кнопок «Загрузить» / «Сохранить» / «По умолчанию»).

3.1.13 Сохранение значений параметров словаря CANopen в файл

Если в памяти контроллера уже содержится исправно работающая программа, которую по тем или иным причинам требуется обновить, то перед началом процесса загрузки программы рекомендуется сохранить на жёсткий диск ПК значения параметров словаря CANopen, так как они могут быть утрачены после обновления программы. Впоследствии значения параметров можно будет загрузить из этого файла.

Для сохранения значений параметров необходимо выполнить следующие шаги:

- установить соединение между ПК и ПМУ (см. п. 3.1.5);
- следуя указаниям раздела 6.7 «OSCT руководство пользователя.pdf», сохранить значения параметров в файл на жёстком диске.

3.1.14 Загрузка значений параметров словаря CANopen из файла

Если на жёстком диске имеется файл с сохранёнными значениями параметров словаря CANopen, то в случае необходимости их можно загрузить в память микроконтроллера.

Для загрузки значений параметров необходимо выполнить следующие шаги:

- установить соединение между ПК и ПМУ (см. п. 3.1.5);
- следуя указаниям раздела 6.7 «OSCT руководство пользователя.pdf», загрузить значения параметров из файла на жёстком диске;
- сохранить параметры в энергонезависимой памяти микроконтроллера, нажав на кнопку «Сохранить» сверху справа в основном окне программы «OSCT»;
- перезагрузить ПМУ, сняв и затем снова подав питание.

3.1.15 Чтение и изменение номера узла устройства в сети CAN

Каждое из устройств, находящихся в сети CANopen, должно иметь уникальный номер узла (сетевой адрес). По умолчанию ПМУ имеет адрес 101.

Чтобы узнать номер узла устройства достаточно установить соединение между ПК и ПМУ (п. 3.1.5). Номер узла отображается в «OSCT» в названии вкладки устройства.

Чтобы изменить номер узла, нужно выполнить следующие шаги:

- установить соединение между ПК и ПМУ (п. 3.1.5);
- в левой части основного окна программы «OSCT» открыть группу параметров «CAN»;
- отредактировать значение параметра «CAN NODE-ID [2000.00]», вписав нужный номер узла;
- подключиться к узлу с новым номером;
- сохранить значения параметров в энергонезависимую память (см. п. 3.1.13).

Подробнее с редактированием параметров словаря можно ознакомиться в разделе 6 «OSCT руководство пользователя.pdf».

3.2 Примеры типового использования

3.2.1 Первоначальная настройка ПМУ

Первоначальная настройка проводится для ПМУ, в памяти микроконтроллера которого не содержится никакой программы. Это может потребоваться для нового, только что изготовленного, устройства.

В том случае, если известна версия программы, которую нужно загрузить в ПМУ, и выходной файл программы с такой версией был ранее собран, то загружать этот файл в память следует при помощи специальных командных файлов, руководствуясь указаниями п.3.1.10.

Если же программу нужной версии можно собрать из исходных файлов проекта, который в данный момент импортирован в «Vector IDE», или же требований к версии программы не предъявляется вовсе, то следует собрать проект (см. п.3.1.7) и загрузить его в память микроконтроллера, руководствуясь указаниями п.3.1.9.

При первоначальной настройке ПМУ загрузить программу можно только с использованием JTAG-эмулятора. Это можно сделать при помощи среды разработки «Vector IDE» в том случае, если из импортированного проекта можно собрать выходной файл той версии, которая требуется.

Процесс первоначальной настройки должен выполняться в следующей последовательности:

- 1) собрать программу актуальной версии (п.3.1.7);
- 2) загрузить программу при помощи JTAG-эмулятора (п.3.1.9);
- 3) установить соединение между ПК и ПМУ (п. 3.1.5);
- 4) обновить, при необходимости, словарь объектов устройства, нажав в главном окне «OSCT» кнопку «Обновить словарь»;
- 5) загрузить значения параметров словаря CANopen из файла, если таковой имеется (п. 3.1.14);

- 6) проверить номер узла устройства в сети CAN (п. 3.1.15);
- 7) сохранить значения параметров в энергонезависимую память (п. 3.1.13);
- 8) перезагрузить ПМУ сняв и затем снова подав питание;
- 9) обновить словарь объектов устройства, нажав в главном окне «OSCT» кнопку «Обновить словарь».

3.2.2 Обновление программы ПМУ в составе изделия

Непосредственный доступ к разъёмам ПМУ в составе ПЧ без вскрытия его корпуса невозможен, поэтому обновление программы в таком случае следует производить с использованием адаптера USB-to-CAN через выведенный наружу сервисный разъём «X18» базовой платы (п.3.1.3). Процесс обновления программы должен выполняться в следующей последовательности:

- 1) установить соединение между ПК и ПМУ (п.3.1.5);
- 2) сохранить значения параметров словаря в файл на жёстком диске (п. 3.1.13);
- 3) собрать проект программы (п. 3.1.7);
- 4) загрузить собранную программу в память микроконтроллера по интерфейсу CAN (п. 3.1.11);
- 5) обновить, при необходимости, словарь объектов устройства, нажав в главном окне «OSCT» кнопку «Обновить словарь»;
- 6) загрузить значения параметров словаря CANopen из файла, полученного в шаге 2 (п. 3.1.14);
- 7) сохранить значения параметров в энергонезависимую память (п. 3.1.13);
- 8) перезагрузить ПМУ, сняв и затем снова подав питание;
- 9) повторно установить соединение между ПК и ПМУ (п. 3.1.5);
- 10) обновить словарь объектов устройства.

3.2.3 Замена ПМУ

При замене ПМУ, в память микроконтроллера заменяющего ПМУ рекомендуется загружать программу такой же версии, что и в заменяемом. В противном случае работоспособность заменяющего ПМУ и всей системы СУ ПЧ в целом не гарантируется.

Заменяющий ПМУ подготавливается к работе по методике, описанной в п. 3.2.1. Все нижеследующие действия применимы только в том случае, когда имеется возможность установить соединение между ПК и заменяемым ПМУ.

Чтобы определить выходной файл программы такой же версии, что и ПО, загруженное в заменяемый ПМУ, необходимо произвести следующие действия:

- 1) установить соединение между ПК и заменяемым ПМУ (п. 3.1.5);
- 2) в списке групп параметров в левой части основного окна программы «OSCT» выбрать группу «СЛУЖЕБНЫЕ»;
- 3) в основном окне на вкладке «Параметры» найти параметр «GIT_VersionShortHash [2002.05]»;
- 4) запомнить или записать шестнадцатеричное значение вышеуказанного параметра, состоящее из 8 символов. Для этого нажать правой кнопкой мыши на имени параметра, выбрать пункт контекстного меню «В других форматах». Шестнадцатеричное значение параметра отображается в поле «Шестнадцат-й вид»;
- 5) найти выходной elf-файл программы, в конце имени которого находится 8-ми символьная последовательность (см. п.3.1.8), совпадающая со значением, полученным в предыдущем шаге. Версия программы этого выходного файла соответствует версии программы, загруженной в заменяемый ПМУ (при условии актуального значения константы VERSION_SHORT_HASH, см. п.3.1.8).

Замена ПМУ выполняется в следующей последовательности:

- 1) установить соединение между ПК и заменяемым ПМУ (п. 3.1.5);

- 2) сохранить значения параметров словаря в файл на жёстком диске (п. 3.1.13);
- 3) определить версию ПО, загруженного в заменяемый ПМУ (см. шаги выше);
- 4) найти нужный выходной файл (см. шаги выше), либо собрать проект, версия которого совпадает с версией программы заменяемого ПМУ (п. 3.1.7);
- 5) загрузить собранную программу в память микроконтроллера заменяющего ПМУ, используя JTAG-эмулятор (п. 3.1.9) или адаптер USB-to-CAN (п. 3.1.11);
- 6) установить соединение между ПК и заменяющим ПМУ (п. 3.1.5);
- 7) загрузить в словарь объектов заменяющего ПМУ значения параметров из файла, сохранённого в шаге 2 (п. 3.1.14);
- 8) сохранить параметры в энергонезависимой памяти заменяющего ПМУ (п. 3.1.12);
- 9) перезагрузить заменяющий ПМУ, сняв и затем снова подав питание;
- 10) повторно установить соединение между ПК и заменяющим ПМУ (п. 3.1.5);
- 11) обновить словарь объектов устройства заменяющего ПМУ;
- 12) извлечь заменяемый ПМУ из ПЧ и установить вместо него заменяющий ПМУ;

4. ПРОВЕРКА ПРОГРАММЫ

4.1 Визуальный контроль

Общую оценку работоспособности программы можно дать, визуально оценив отображение информации на дисплее, реакцию на нажатие кнопок и поведение светодиодов, расположенных на ПМУ.

Непосредственно после подключения цифрового питания ПМУ с загруженной в память программой ожидает появления базовой платы в сети CAN, отображая на дисплее статус «ИНИЦИАЛИЗАЦИЯ». При обнаружении базовой платы с ней устанавливается связь и на дисплее отображается т.н. «главный экран» – статус ПЧ и значения его основных параметров.

Реакцию на кнопки можно проверить путём вызова меню «Опции» (левая функциональная клавиша) или главного «Меню» (правая функциональная клавиша) и навигации по ним.

Зелёный светодиод VD1 отображает статус «Готов» (горит постоянно) и «Работа» (мигает с периодом 1 сек). Красный светодиод VD2 отображает статус «Норма» (не горит), «Предупреждение» (мигает с периодом 1 сек) и «Авария» (горит постоянно).

Таким образом гарантировано, что отсутствуют серьёзные сбои с цифровым питанием, тактированием микроконтроллера и общим функционированием программы.

4.2 Проверка с использованием управляющего интерфейса

Более детальная проверка правильности функционирования программы осуществляется с использованием программы управляющего интерфейса «OSCT». Перед началом проверки необходимо установить соединение между ПК и ПМУ (п. 3.1.5).

4.2.1 Проверка назначения программы

Убедиться, что в верхней левой части основного окна программы «OSCT», над кнопками «Подключиться» и «Обновить словарь» отображается правильное название устройства: «ПМУ», а в выпадающем списке справа от кнопки «Обновить словарь» выбран профиль «ПМУ-КЭАЗ-400.xml».

Таким образом гарантировано, что загруженная программа соответствует назначению силового преобразователя.

4.2.2 Проверка версии программы

При условии, что программист следовал рекомендациям по записи версии программы в имени выходного файла, как описано в п. 3.1.8 данного руководства, убедиться в правильности версии загруженной программы можно следующим образом:

- в списке групп параметров в левой части основного окна программы «OSCT» выбрать группу «СЛУЖЕБНЫЕ»;
- в основном окне на вкладке «Параметры» найти параметр «GIT_VersionShortHash [2002.05]»;
- запомнить или записать шестнадцатеричное значение вышеуказанного параметра, состоящее из 8 символов. Для этого нажать правой кнопкой мыши на имени параметра, выбрать пункт контекстного меню «В других форматах». Шестнадцатеричное значение параметра отображается в поле «Шестнадцат-й вид»;
- сопоставить значение параметра с именем загруженного выходного файла программы.

Если значение параметра совпадает с числом, указанным в имени выходного файла программы, значит именно этот файл был загружен в память микроконтроллера.

5. ДОПОЛНИТЕЛЬНЫЕ ВОЗМОЖНОСТИ

В программе предусмотрен встроенный загрузчик ПО через интерфейс CAN. Основная функция встроенного загрузчика – восстановление работоспособности ПМУ после сбоя в питании во время прошивки, или после загрузки неисправной программы в память микроконтроллера. Встроенный загрузчик позволяет восстановить работоспособность ПМУ по сети CAN без необходимости подключения JTAG-эмулятора, что важно, если доступ к ПМУ сильно затруднен (например, он установлен в ПЧ).

Встроенный загрузчик располагается в нулевом секторе FLASH-памяти микроконтроллера. При нормальной процедуре загрузки программы в память микроконтроллера через программу «OSCT» нулевой сектор не стирается. Таким образом гарантировано, что даже если во время процесса прошивки ПМУ произойдет сбой по питанию, встроенный загрузчик сохранит работоспособность.

Во время загрузки ПМУ управление сначала получает встроенный загрузчик, который в режиме «listen only» слушает CAN-линию на скорости 250 кбит/с в течение 100 мс. Если за это время по CAN-линии придет специальная посылка, то встроенный загрузчик загрузит в ОЗУ микроконтроллера код обновления программы по интерфейсу CAN (входит в состав бутлоадера) и передаст ему управление. Таким образом, ПМУ станет доступен по сети CAN и программу ПМУ можно будет обновить. В случае, если через 100 мс специальная посылка не получена, встроенный загрузчик передает управление в программу пользователя.

Для перевода встроенного загрузчика в режим загрузки программы по интерфейсу CAN необходимо послать специальную посылку (значимыми являются не только идентификатор, но и данные), содержание которой приведено в таблице 6 :

Таблица 6 – Содержание специальной посылки

ID	F/F	RTR	LEN	D0	D1	D2	D3	D4	D5	D6	D7
0x1FFF0BAD	EFF	нет	8	0x12	0x34	0x56	0x78	0x9A	0xBC	0xDE	0xF0

Данное сообщение может быть отправлено через адаптер USB-to-CAN «CANbus-USBnr» с использованием программы «CANwise». Инструкции по отправке сообщений в сеть приведены в разделе 5 «CANwise руководство пользователя.pdf».

Альтернативный вариант – использовать управляющий интерфейс «OSCT» через меню «Сервис» → «Программирование» → «Активировать CAN-загрузчик». Подробные инструкции приведены в окне активации (там в качестве примера указана скорость 125 кбит/с, но в нашем случае используется 250 кбит/с).

6. СООБЩЕНИЯ

6.1 Сообщения, выдаваемые при сборке программы

Во время сборки программы в среде «Vector IDE» автоматически открывается окно «Console», в которое выводится стандартная информация набора кросс-средств компиляции «GCC» о процессе сборки: текущая стадия сборки, указание вызова той или иной программы (компилятор, компоновщик и т. д.) и список параметров, переданный этой программе.

При возникновении ошибок во время сборки, информация об этих ошибках отображаются в окне «Console» на красном фоне. По этой информации можно определить, по какой причине произошла ошибка. Рассмотрение всех возможных ошибок выходит за рамки данного документа. Наиболее распространённые ошибки описаны в разделе 7 «VectorIDE руководство пользователя.pdf».

При успешном завершении сборки в окне «Console» появится строка синего цвета с текстом «[HH:MM:SS] Build Finished. 0 errors, 0 warnings. (took [X_time])», где «HH:MM:SS» – время окончания сборки, «X_time» – длительность сборки.

6.2 Сообщения, выдаваемые в ходе работы программы

Сообщениями, выдаваемыми в ходе работы программы можно считать информацию, отображаемую на дисплее согласно требованиям к интерфейсу пользователя – см. «Техническое задание на составную часть опытно-конструкторской работы: Разработка программного обеспечения системы управления преобразователем частоты с векторным и скалярным управлением».

ПЕРЕЧЕНЬ ТЕРМИНОВ И СОКРАЩЕНИЙ

В настоящем документе применяются следующие сокращения:

ОЗУ	– оперативное запоминающее устройство
ПМУ	– пульт местного управления
ПО	– программное обеспечение
ПЧ	– преобразователь частоты
СУ	– система управления
EEPROM	– Electrically Erasable Programmable Read-Only Memory (электрически стираемая программируемая постоянная память)

В настоящем документе применяются следующие термины:

Микроконтроллер	– микросхема интегральная К1921ВК01Т2, выполняющая роль центрального процессора ПМУ
Среда разработки	– набор программных средств, включающих в себя текстовый процессор, набор кросс-средств компиляции для сборки проекта программы, программный отладчик для загрузки программы в память целевого микроконтроллера
Проект программы	– набор исходных файлов, из которых может быть собран файл программы
Сборка программы	– процесс создания файла программы, получаемый путём компиляции, ассемблирования и компоновки исходных файлов программы
Выходной файл программы	– выходной файл с расширением «.elf», получаемый после сборки программы, и который может быть загружен в память микроконтроллера
Управляющий интерфейс	– набор программных средств, позволяющих осуществлять мониторинг и настройку устройств посредством ПК, подключаемого через физический интерфейс

- Встроенный загрузчик – алгоритм, являющийся частью программы для ПМУ, служащий для восстановления работоспособности ПМУ после сбоя в питании во время прошивки, или после загрузки неисправной программы в память микроконтроллера
- JTAG-эмулятор – техническое средство, служащее для загрузки программы в память микроконтроллера и внутрисхемной отладки через стандартизированный интерфейс JTAG
- Адаптер USB-to-CAN – устройство, подключаемое к ПК через USB интерфейс, предназначенное для обмена данными между ПК и устройствами, объединёнными в сеть по интерфейсу CAN

