

ООО «СОФТИТЕК»

450112, Респ. Башкортостан, г. Уфа

ул. Мира, д. 61, офис 223

E-mail: info@softitech.ru

Тел.: +7 (347) 216-43-53

www.softitech.ru

ИНН 0277961766, КПП 027701001

ОГРН 1230200011249



Операционная система

Softitech Embedded Linux

Руководство пользователя

Уфа 2025

Содержание

1. ВВЕДЕНИЕ	3
1.1 Особенности операционной системы	3
1.2 Установка ОС на ПЛК S01.PLC.01	3
2. Основные принципы функционирования Softitech Embedded Linux	8
2.1 Принцип функционирования процессов в ОС	8
2.2 Организация файловой системы Softitech Embedded Linux	10
2.3 Физические носители (диски) и логические разделы	11
3. Краткое описание ОС Softitech Embedded Linux	12
3.1 Базовые функциональные характеристики и возможности Softitech Embedded Linux	12
3.2 Целевая аудитория ОС	12
4 Функциональные характеристики Softitech Embedded Linux	13
4.1 Основные предустановленные пакеты	13
5. Начало работы и запуск системы Softitech Embedded Linux	14
5.1 Общая информация	14
5.2 Работа с командной оболочкой Bash	15
5.3 Основные команды командной оболочки Bash	15
5.4. Конфигурация интерфейсов	19
5.4.1 Конфигурирование Ethernet интерфейсов	19
5.4.2 Конфигурирование системных часов и часового пояса	21

1. ВВЕДЕНИЕ

Добро пожаловать в документацию дистрибутива Softitech Embedded Linux ОС! Этот материал рассчитан как на новичков, так и на опытных пользователей. В руководстве подробно рассматриваются этапы подготовки системы к установке, сам процесс установки дистрибутива, а также дальнейшие шаги по его настройке и использованию.

Руководство соответствует актуальной информации на текущий момент, однако возможно, что часть информации может быть дополнена для более наглядного понимания информации. Если вы обнаружите недоработки или неточности, они будут оперативно исправлены при очередном обновлении документации.

1.1 Особенности операционной системы

Операционная система (ОС) Softitech Embedded Linux представляет собой специализированную встраиваемую ОС на базе ядра Linux и содержит специально отобранный набор компонентов и модулей программного обеспечения (ПО). Данная ОС создана и оптимизирована исключительно для работы на программируемых логических контроллерах (ПЛК) Softitech серии S01.PLC.01 и обеспечивает выполнение их прикладного ПО, а также реализует возможности этих ПЛК, предоставляя удобные функции доступа к аппаратной части.

Традиционно большинство программных компонентов этой ОС написано на языке Си, поэтому язык этот является основным, однако также есть и поддержка языка Python, поскольку для него разработано много аппаратных библиотек, а сам язык прост и лаконичен для написания прикладных программ и системных скриптов.

В Softitech Embedded Linux заложена поддержка работы с интегрированной средой разработки (IDE) CODESYS 3.5

Особенностью IDE CODESYS 3.5 является поддержка языков программирования стандарта МЭК 61131-3, к которым относятся:

- язык лестничных диаграмм LD;
- язык диаграмм функциональных блоков FBD;
- язык списка инструкций IL;
- язык структурированного текста ST;
- язык диаграммы последовательных функций SFC;
- язык Continuous Function Chart, основанный на базе FBD с непрерывным выполнением алгоритмов и гибкой системой функциональных блоков.

1.2 Установка ОС на ПЛК S01.PLC.01

Для установки ОС Softitech Embedded Linux на ПЛК Softitech серии S01.PLC.01 требуется персональный компьютер, который по своим параметрам не уступает следующему:

- процессор разрядности 32/64 семейства x86/x64;
- ОЗУ 1 ГБ;
- 200 Мб свободного пространства;
- операционная система Windows 7/Windows 10
- наличие свободного порта ethernet или возможность подключить интерфейс USB-ethernet;
- наличие 2-х свободных разъемов USB.

Также потребуются дополнительные устройства:

- материнская плата StarterKit SK-iMX6ULL-NANO-2E-MB компании StarterKit с проводом USB Type-A – USB Type-C – плата имеет слот PCI-E для установки процессорного модуля из S01.PLC.01, на который будет устанавливаться данная ОС;
- интерфейс USB-UART любой модели;
- FLASH накопитель USB Type-A объёмом не менее 16Гб, с файловой системой exFAT.

На ПК должно находиться следующее ПО (помимо ОС):

- PuTTY – программа-терминал, позволяющая видеть системные сообщения, исходящие от процессорной платы;
- Zadig – программа для установки драйвера WinUSB, дающего ПК доступ к материнской плате SK-iMX6ULL-NANO-2E-MB;
- набор фирменных утилит FEL компании StarterKit, содержащие скрипты для прошивки процессора;
- драйвер интерфейса USB-UART соответствующей модели.

Подготовьте FLASH накопитель USB Type-A. А именно:

- отформатируйте её стандартными средствами в файловой системе exFAT;
- перепишите в корневой каталог носителя прилагаемые файлы скомпилированного образа: «sdcard.img» и «autorun.sh»

Извлеките из устройства S01.PLC.01 процессорный модуль SK-A40i-NANO-2E:

- извлеките маленькой плоской отверткой пружину (1) (рис.1);



Рис. 1. Снятие пружины и скобы



Рис 2. Снятие верхних коннекторов

- потяните скобу (2) наверх (рис. 1) и снимите её;
- снимите два верхних коннектора (рис. 2);
- разведите в стороны две половинки корпуса пальцами через верхние разъёмы (рис. 3);
- извлеките плату модуля (рис. 4);

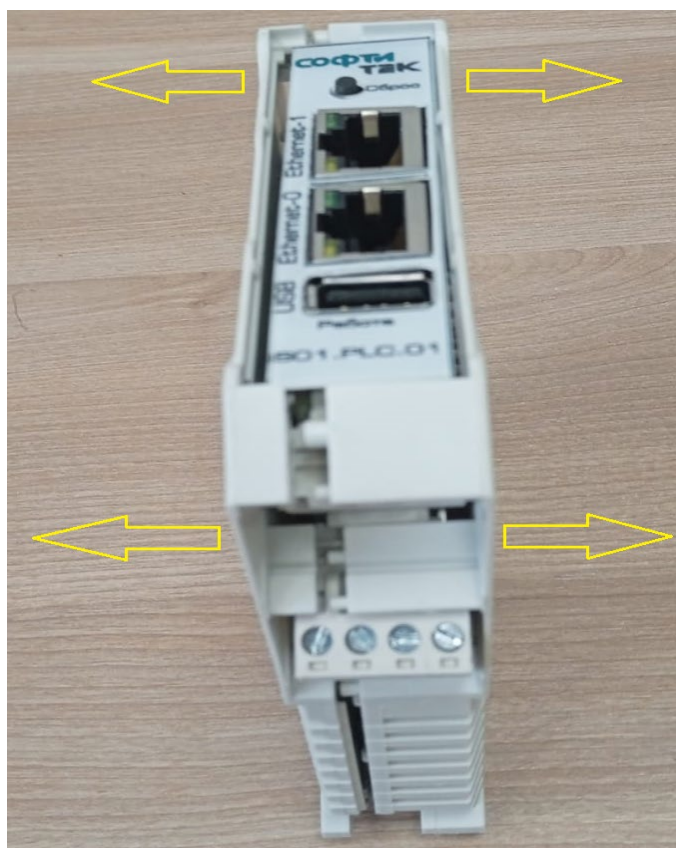


Рис. 3. Разборка корпуса

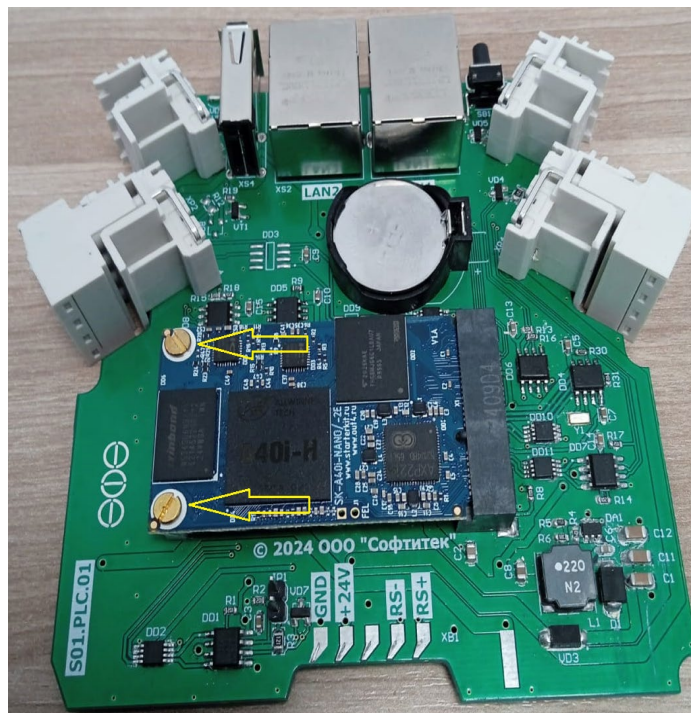


Рис. 4. Снятие процессорного модуля

- меленькой плоской отверткой открутите 2 винта (рис. 4) и вытащите процессорный модуль.

Подготовьте аппаратную часть устройства для прошивки (см. рис.5):



Рис. 5. Материнская плата SK-iMX6ULL-NANO-2E-MB с процессорным модулем SK-A40i-NANO-2E

- разместите на столе материнскую плату SK-iMX6ULL-NANO-2E-MB;
- установите в разъем PCI-E процессорный модуль;

- подсоедините кабелем USB Type A – micro USB материнскую плату к ПК;
- присоедините интерфейс USB – UART к консольному порту ПЛК и ПК.

Установите драйвер WinUSB для связи с материнской платой, для этого:

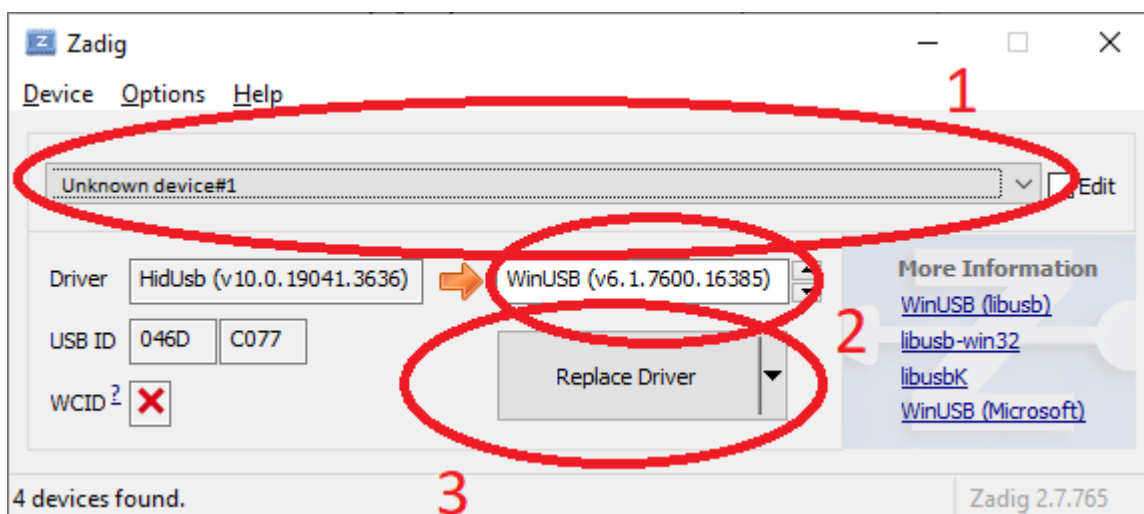


Рис. 6. Программа Zadig

- запустите программу Zadig-2.7.exe из директории DFU (рис. 6);
- в меню «Options» установите флажок «List All Devices», при наличии во всплывающем списке (1) пункта «Unknown device #1» - выберите его, проверьте, чтобы в поле «Driver» справа стояло значение WinUSB (2), затем нажмите кнопку «Replace driver» (3)

Запустите процесс загрузки ПО, для этого:

- установите переключку на контакт «FEL» процессорного модуля (рис. 2),
- нажмите кнопку «Reset» на плате разработчика
- запустите файл «boot.bat» из директории FEL
- дождитесь, когда в консоли ПЛК появится надпись «Starting kernel»
- вытащите переключку «FEL»

Прошивка окончена!

2. Основные принципы функционирования Softitech Embedded Linux

Взаимодействие с операционной системой осуществляется через текстовый интерфейс, где пользователь вводит команды и получает ответы в символьном формате. Управление системой происходит через специальные оболочки (shell), которые взаимодействуют с базовыми компонентами программного обеспечения. Система оптимизирована для минимального использования ресурсов, сохраняя при этом все необходимые функциональные возможности.

Базовые библиотеки представляют собой набор предварительно скомпилированных модулей, которые автоматически подключаются по мере необходимости. Эти библиотеки обеспечивают выполнение ключевых операций и выступают связующим звеном между приложениями и ядром системы. Они собираются с учетом минимальных требований к размеру и производительности, что особенно важно для устройств с ограниченными ресурсами.

Ядро создает защитный барьер между пользователями и аппаратным обеспечением, скрывая сложность работы с оборудованием.

Реализуется многоуровневая система безопасности, которая защищает как данные, так и аппаратные ресурсы от несанкционированного доступа.

ОС позволяет создать скрипты для автоматизации рутинных операций, таких как запуск сервисов или настройка сетевых интерфейсов. В ОС минимизировано использование памяти и процессорного времени за счет тщательно оптимизированных компонентов. Пользователь взаимодействует с оборудованием только через строго контролируемые интерфейсы, таким образом нет необходимости глубокого понимания работы аппаратного обеспечения.

2.1 Принцип функционирования процессов в ОС

Процессы играют ключевую роль в работе операционной системы, выполняя различные задачи. Программа представляет собой набор машинного кода и данных, хранящихся в исполняемом файле на диске. Однако процесс — это не просто программа, а более сложная структура, включающая программный счетчик, регистры центрального процессора, стеки для временных данных (например, параметры функций, адреса возврата и сохраненные переменные) и другие элементы. Текущая исполняемая программа или процесс полностью определяет деятельность микропроцессора в данный момент времени.

Softitech Embedded Linux — это многозадачная операционная система, которая эффективно управляет процессами, каждый из которых в системе является независимой задачей со своими правами и обязанностями. Одним из основных преимуществ такой архитектуры является изоляция: если один процесс завершится аварийно, это не повлияет на работу других. Каждый процесс выполняется в собственном виртуальном адресном пространстве, что предотвращает прямое взаимодействие между разными

процессами, для безопасного общения которых используются механизмы, контролируемые ядром системы, такие как межпроцессное взаимодействие (IPC).

За время своего существования процесс может использовать множество системных ресурсов. Например, он задействует CPU для выполнения инструкций, физическую память для хранения данных, файловые системы для чтения и записи файлов, а также физические устройства, такие как сетевые интерфейсы или RTC. Операционная система должна постоянно отслеживать состояние всех процессов и распределять системные ресурсы справедливо между ними. Это обеспечивает стабильность и производительность всей системы.

Softitech Embedded Linux предоставляет широкий набор инструментов для управления процессами и их анализа. Эти инструменты позволяют запускать, останавливать, приостанавливать, возобновлять, а также получать подробную информацию об их состоянии. Вот некоторые популярные утилиты для работы с процессами:

Утилита	Пример вызова	Описание
top	top	показывает динамическое представление активных процессов с детальной информацией об использовании ресурсов.
ps	ps aux	выводит детализированный список всех запущенных процессов в системе (один раз и статично)
kill	kill PID	отправляет сигнал процессу с номером PID для его завершения
pkill	pkill name	отправляет сигнал процессу с именем name для его завершения
nice	nice -n 10 prg	Назначение приоритета процессу. Например, запуск программы prg с низким приоритетом
pidof	pidof process	Показывает номер процесса в системе по его имени (process)
lsof	lsof /tmp/file	Выводит список открытых файлов, используемых процессами. В примере: показать кто открыл файл /tmp/file
Ps	Ps -U user	Показывает все задачи, запущенные пользователем user
htop	htop	Отображает состояние об использовании ресурсов ОС
dmesg	dmesg	Отображает сообщения ядра ОС, которая содержит в т.ч. информацию о запуске, завершении или ошибках процессов

2.2 Организация файловой системы Softitech Embedded Linux

Основные разделы корневой файловой системы представлены следующими директориями:

Директория	Описание
/bin	Содержит основной набор утилит, необходимых для работы системы
/dev	Хранит файлы устройств (device files), которые представляют физические или виртуальные устройства
/etc	Содержит конфигурационные файлы системы и программ
/home	Личные директории пользователей, где хранятся их данные, настройки и программы
/include	Содержит заголовочные файлы (*.h) для разработки программ на языках C/C++
/lib	Библиотеки, необходимые для работы базовых команд и системных утилит, в том числе загружаемые модули ядра, используемые при загрузке ОС
/lib32	Библиотеки для поддержки 32-битных приложений на 64-битной системе
/libexec	Внутренние исполняемые файлы, используемые другими программами, но не предназначенные для прямого вызова пользователем
/media	Точки монтирования для съемных носителей (например, USB-flash)
/mnt	Временные точки монтирования для файловых систем
/opt	Установка сторонних приложений и программ, которые не входят в стандартный дистрибутив
/proc	Виртуальная файловая система, предоставляющая информацию о состоянии ядра и процессов
/root	Домашняя директория суперпользователя (root)
/run	Хранит временные данные, актуальные только для текущей сессии
/sbin	Системные утилиты, предназначенные для администрирования (например, fdisk, ifconfig)
/srv	Данные, предоставляемые сервисами (например, веб-серверами или FTP)
/sys	Виртуальная файловая система, предоставляющая информацию об устройствах и драйверах
/tmp	Временные файлы, создаваемые программами, обычно физически располагается в ОЗУ и стирается при перезапуске

/usr	Директория предназначена для размещения всех установленных программ. Локально установленные приложения хранятся в /usr/local
/var	Переменные данные, изменяющиеся в процессе работы системы

2.3 Физические носители (диски) и логические разделы

ЕММС flash диск программируемого логического контроллера (ПЛК), который используется как диск долговременного хранения, может быть разделен на несколько логических разделов. Каждый из этих разделов функционирует как независимый виртуальный диск, что позволяет эффективно организовывать хранение данных и управлять ими.

Основная идея разделения диска заключается в том, чтобы использовать один физический носитель для различных целей так, чтобы изменения, вносимые на один диск никак не затрагивали состояние другого, в том числе допустимо наличие разных файловых систем на них. Например, если необходимо установить несколько операционных систем на одном устройстве, можно создать для каждой из них свой логический раздел. Каждая система будет работать в своем выделенном пространстве, не пересекаясь с другой. Это обеспечивает изоляцию данных и предотвращает конфликты между системами, хотя физически данные будут храниться на одном носителе. Это также экономит средства и не занимает дополнительных аппаратных ресурсов устройства.

Все устройства хранения данных, подключенные к ПЛК, включая встроенные ЕММС-диски и USB-накопители, отображаются в директории /dev. Каждое устройство получает уникальное обозначение, например:

/dev/sda — первое устройство на шине USB.

/dev/sdb — второе устройство.

/dev/mmcblk0 — типичное обозначение для ЕММС-дисков.

/dev/mmcblk0p1, /dev/mmcblk0p2 — разделы на ЕММС-диск

Для работы с дисками и разделами в ОС предусмотрены следующие команды:

Утилита	Описание утилиты
lsblk	Отображает список блочных устройств и их разделов в виде дерева
fdisk	Интерактивная утилита для создания, изменения и удаления разделов на MBR-дисках. Пример: fdisk -l показывает подробную информацию о дисках и их разделах
mount	Подключение раздела («монтирование») к определенной директории, например: mount /dev/sda1 /mnt/data – монтирование устройства /dev/sda1 в директорию /mnt/data
umount	Отключение раздела («размонтирование»), например: umount /mnt/data

fsck	Проверка и исправление ошибок файловой системы. Пример: fsck /dev/sda1
df	Показывает общее использование дискового пространства на всех подключенных файловых системах. Она предоставляет информацию о доступном и занятом месте на каждом разделе. Например, df -h
du	Показывает использование дискового пространства для конкретных файлов и директорий. Она помогает определить, какие файлы или папки занимают больше всего места. Пример: du -sh

3. Краткое описание ОС Softitech Embedded Linux

3.1 Базовые функциональные характеристики и возможности Softitech Embedded Linux

ОС Softitech Embedded Linux обеспечивает возможность:

- 1) выполнять прикладное программное обеспечение на аппаратной платформе ПЛК S01.PLC.01;
- 2) функционирования внешних интерфейсов ПЛК (RS-485, Ethernet, CAN, USB);
- 3) обращения к аппаратным интерфейсам ПЛК (RS-485, Ethernet, CAN, USB);
- 4) функционирования файловой системы ОС;
- 5) обращения к встроенным устройствам ПЛК (часы реального времени, FRAM память, ЕММС и т.д.)
- 6) обмена информацией между прикладным ПО и ПЛК;
- 7) удалённого управления по протоколу SCP или SSH;
- 8) взаимодействия пользователя и аппаратной части ПЛК (т.н. «человек-машинный интерфейс или HMI).

3.2 Целевая аудитория ОС

С учётом специфики данного ПЛК, под нужды которого была создана данная ОС, в целевую аудиторию этой ОС потенциально входят:

- 1) системные интеграторы и инженеры автоматизации (компании и специалисты, внедряющие ПЛК в промышленные системы управления);
- 2) технические специалисты и администраторы (сотрудники, отвечающие за поддержку, обновление и диагностику уже развернутых систем).
- 3) студенты и исследователи, обучающиеся в области встраиваемых систем и промышленной автоматизации.

4 Функциональные характеристики Softitech Embedded Linux

ОС разработана для поддержки ПЛК серии S01.PLC.01 (на данный момент это единственное поддерживаемое устройство, однако разработчик продукта оставляет за собой право добавлять в будущем поддержку других устройств)

4.1 Основные предустановленные пакеты

Пакет	Описание
base system utilities (coreutils, findutils, util-linux)	coreutils содержит базовые системные команды: cat, cp, mv, rm, ls, chmod, chown, mkdir, rmdir, touch, dd, df, du, basename, dirname, echo, false, true, test, unlink, sync findutils содержит утилиту find для поиска файлов и директорий и утилиту xargs, выполняющую команды для найденных файлов util-linux содержит утилиты для работы с файловыми системами (mount, umount), устройствами (dmesg, lsblk, lscpu, lsusb), сетью (hostname, hostnamectl), временем (timedatectl).
networking tools (iproute2, openssh, net-tools)	iproute2 содержит утилиты для управления сетью: ip, ss, tc. openssh содержит утилиты для безопасного удаленного доступа: ssh, scp, sftp, ssh-agent, ssh-add, ssh-keygen, ssh-copy-id. net-tools содержит утилиты для работы с сетью: netstat, ping, ping6, traceroute.
text processing tools (grep, sed, awk, diff)	утилиты для поиска текста в файлах или потоках (grep, egrep, fgrep) поточковый текстовый редактор sed обработка текстовых данных awk утилиты сравнения файлов (diff, diff3, cmp, sdiff) утилиты обработки текста stiff: cut, paste, tr, uniq, wc, expand, unexpand, fmt, fold.
shell environment (bash)	командная оболочка bash
system management (systemd)	управление службами и системными процессами: systemctl, journalctl, systemd-

	analyze, systemd-cat, systemd-cgls, systemd-detect-virt, systemd-firstboot, systemd-mount, systemd-nspawn
development tools (gcc, python, make)	компилятор для C/C++: gcc, g++ интерпретатор Python: python, python3, python3.10 автоматизация сборки программ make библиотеки для C/C++, Python
CAN-utils package	утилиты для работы с CAN-шиной: candump, cansend, canbusload, canfdtest, cangen, cangw, canlogserver, canplayer, cansequence, cansniffer.
monitoring tools (htop, vmstat)	Утилиты для мониторинга процессов: ps, top, uptime, free, slabtop, htop, vmstat.
minicom	Утилиты для работы с последовательными портами: minicom, picocom.
текстовые редакторы	nano, vi, vim, vimdiff
tar	работа с архивами данных
GPIO tools	Утилиты для работы с GPIO: gpioget, gpioreset, gpiomon, gpiodetect.
Softitech tools	Набор фирменных утилит для работы с modbus протоколом: mb_write_x, mb_read_x, mb_adr_search, mb_adr_set, test_rs485 Набор фирменных утилит для тестирования модулей серии S01: testPLC, testDO16, testDI16, testRTD04, testAI08, testAO08

5. Начало работы и запуск системы Softitech Embedded Linux

5.1 Общая информация

Для работы с ОС Softitech Embedded Linux необходимо ознакомиться с данным руководством и руководством по эксплуатации ПЛК S01.PLC.01. Предполагается, что пользователь имеет представление и опыт работы с командной оболочкой bash, традиционной для многих версий семейства Linux и прошёл инструктаж по технике безопасности при обращении с электроприборами.

После включения ПЛК к электрической сети осуществляется запуск Softitech Embedded Linux; загрузка происходит в течение 20-30 секунд (в это время контроллер недоступен для любых команд извне), о завершении загрузки сигнализирует светодиод «Работа» на

передней панели ПЛК. После этого пользователь может подключиться к ПЛК через любой ethernet порт по протоколу SFTP или SSH, используя программу-терминал¹ (например, встроенная в Windows утилита ssh). При этом надо указать логин (по умолчанию «root»), после чего пользователю будет предложено ввести пароль (см. рис. 5.1) – по умолчанию «root».

```
root@172.120.42.32's password:
Welcome to Softitech S01.PLC.01!
#
```

Рис. 5.1 Вход в систему пользователя

Система даёт три попытки ввода логина и пароля в случае некорректных значений, для следующих попыток надо повторно подключаться к ПЛК.

5.2 Работа с командной оболочкой Bash

Bash — это командная оболочка (shell) и язык сценариев (scripting language), используемая в операционных системах Linux. Она является посредником между пользователем и ядром ОС, или т.н. «интерпретатором командной строки», который принимает команды, понятные человеку и преобразует их в данные для ядра ОС. Bash позволяет не только воспринимать команды, но и автоматизировать их ввод через написание сценариев (скриптов). Помимо ввода данных с клавиатуры или из скрипта, bash также способен принимать и отправлять данные на другие устройства ввода-вывода (например, порты RS485, CAN и пр.)

В ОС Softitech Embedded Linux не предусмотрено других оболочек командной строки кроме bash и sh (более ранний вариант bash), графического интерфейса пользователя не предусмотрено. Управление ОС таким образом осуществляется только через терминальную утилиту (далее по тексту «Терминал»), которая связывается с ПЛК через физический порт Ethernet по протоколу SSH или SFTP.

Терминал воспринимает команды пользователя, которые вызывают встроенные программы ОС, которые, в свою очередь, были заранее установлены в системе в составе пакетов ПО (см. п.4.1) – например: «ps», «ls», «cat» и др. Результат выполнения пользователь видит также в Терминале. Выполнение команд, как и ввод осуществляется последовательно.

5.3 Основные команды командной оболочки Bash

Группа команд для работы с файловой системой

¹ Пример подключения к ПЛК в операционной системе Windows 10 при помощи встроенной утилиты ssh:

1) нажмите Windows+R, введите «cmd», нажмите ENTER

2) введите команду

```
$ ssh root@xxx.xxx.xxx.xxx,
```

где вместо xxx.xxx.xxx.xxx введите IP-адрес ПЛК (по умолчанию это 192.168.1.10 для ethernet1 и 192.168.1.11 для ethernet2), нажмите ENTER

3) на вопрос системы «Are you sure you want to continue connecting (yes/no/[fingerprint])?» ответьте утвердительно вводом «yes» и нажатием ENTER

Команда	Описание
cp	копирует файлы и директории
mv	перемещает или переименовывает файлы и директории
rm	удаляет файлы и директории
mkdir	создаёт новую директорию
rmdir	удаляет пустую директорию
touch	создаёт пустой файл или обновляет метку времени существующего файла
ls	выводит список файлов и директорий
findmnt	отображает смонтированные файловые системы
df	показывает использование дискового пространства
du	показывает использование дискового пространства для файлов и директорий
chmod	изменяет права доступа к файлам и директориям
chown	изменяет владельца файла или директории
chgrp	изменяет группу файла или директории
chattr	изменяет атрибуты файлов на уровне файловой системы (например, делает файл неизменяемым)
lsattr	просматривает атрибуты файлов
mount	монтирует файловые системы
umount	размонтирует файловые системы
mountpoint	проверяет, является ли директория точкой монтирования
ln	создает жесткие или символические ссылки
readlink	показывает целевой файл символической ссылки
file	определяет тип файла
stat	показывает подробную информацию о файле (размер, права, время изменения и т.д.)
lsblk	выводит список блочных устройств (дисков и разделов)
fdisk	утилита для управления разделами дисков
mktemp	создает временные файлы или директории
sync	синхронизирует данные с диском
rsync	синхронизирует файлы и директории между локальными и удаленными системами
dd	копирует и преобразует данные на низком уровне
realpath	показывает абсолютный путь к файлу
grep -ir	поиск файлов в каталоге по шаблону
mc	вызов файлового менеджера Midnight Commander, имеющий вид, напоминающий классический Norton Commander

Группа команд для осуществления настройки сетевых интерфейсов и работе с сетью

Команда	Описание
ifconfig	утилита для настройки сетевых интерфейсов
ip	Универсальная утилита для управления сетевыми интерфейсами, маршрутизацией и туннелями
netstat	Утилита для просмотра сетевых подключений, таблиц маршрутизации и статистики
ping	Проверка доступности хостов через ICMP
traceroute	Отображение маршрута пакетов до целевого хоста
networkctl	Утилита для управления сетевыми интерфейсами через systemd-networkd
arp	сопоставление IP-адреса и MAC-адреса в локальной сети

Группа команд для настройки системного времени

Команда	Описание
timedatectl	Управление системным временем и часовым поясом
ntpdate	Синхронизация времени с NTP-сервером
hwclock	утилита для работы с микросхемой RTC

Группа команд для работы с удалённым доступом

Команда	Описание
ssh	подключение к удаленному серверу по протоколу SSH
scp	копирование файлов через SSH
sftp	безопасная передача файлов
ssh-keygen	генерация ключей SSH
ssh-agent	управление ключами SSH
ssh-copy-id	копирование ключа на удаленный сервер

Группа команд для перезагрузки и выключения ПК

Команда	Описание
shutdown	управление выключением или перезагрузкой ПК
shutdown -h now	выключение системы
shutdown -r +5	перезагрузка через 5 минут
reboot	перезагрузка ПК

Группа команд для работы с текстом

Команда	Описание
cat	выводит содержимое файла на экран или объединяет файлы
echo	выводит текст или переменные на экран
fmt	форматирует текст, переносит строки

fold	переносит строки по указанной ширине
expand	преобразует табуляцию в пробелы (expand) и наоборот (unexpand)
grep	ищет строки, содержащие указанный шаблон
egrep	расширенный поиск с поддержкой регулярных выражений
fgrep	поиск фиксированных строк без интерпретации регулярных выражений
diff	сравнивает два файла и показывает различия
diff3	сравнивает три файла
cmp	сравнивает два файла побайтово
sdiff	показывает различия между двумя файлами в виде двух колонок
tr	заменяет или удаляет символы
cut	вырезает части строк (например, столбцы)
paste	объединяет строки из нескольких файлов
sed	поточковый текстовый редактор для выполнения операций замены, удаления и т.д.
awk	язык обработки текстовых данных для сложных манипуляций
vim	популярный среди программистов текстовый редактор
nano	простой и удобный текстовый редактор, ориентированный на новичков

Группа команд для работы со сжатыми файлами

Команда	Описание
tar	создание, просмотр и распаковка tar-архивов
gzip	сжатие файла в формате .gz
gunzip	распаковка gzip-архивов .gz
gzexe	создание самораспаковывающихся скриптов из gzip-сжатых файлов
xz	сжатие файла в формате .xz
unxz	распаковка xz-архивов .xz
xzcat	вывод содержимого xz-архива без распаковки
split	разделение больших файлов на части (полезно для создания многотомных архивов)

Группа специализированных команд для ПЛК Softitech S01.PLC.01

Команда	Описание
mb_adr_set	задание modbus адреса устройству
mb_adr_search	поиск modbus адреса, назначенному устройству (если устройство единственное на шине)
mb_write_x	запись регистра в указанное modbus устройство
mb_read_x	чтение регистра из указанного modbus устройства

test_rs485	тестирование работоспособности заданного modbus устройства, подключенного к одному из RS485 портов
testPLC	скрипт автоматического тестирования модуля S01.PLC.01
testDI16	скрипт автоматического тестирования модуля S01.DI16.01
testDO16	скрипт автоматического тестирования модуля S01.DO16.01
testRTD04	скрипт автоматического тестирования модуля S01.RTD04.01
testAI08	скрипт автоматического тестирования модуля S01.AI08.01
testAO08	скрипт автоматического тестирования модуля S01.AO08.01

5.4. Конфигурация интерфейсов

5.4.1 Конфигурирование Ethernet интерфейсов

Заводские настройки ethernet портов ПЛК имеют следующую конфигурацию:

Параметр	Конфигурация Ethernet1	Конфигурация Ethernet2
IP-адрес	192.168.1.10	192.168.1.11
маска подсети	255.255.255.0	255.255.255.0
шлюз	0.0.0.0	0.0.0.0
DNS	192.168.1.0	192.168.1.0

Соответственно, для подключения к ПЛК Ваш ПК должен иметь ту же подсеть и маску подсети, что и порт ПЛК.

Пользователь в любой момент может изменить IP-адрес и маску подсети. Для этого необходимо отредактировать файл, отвечающий за настройку соответствующего интерфейса:

Путь к файлу	Назначение файла
/etc/systemd/network/10-eth0.network	Конфигурация интерфейса eth0 (порт Ethernet-0)
/etc/systemd/network/10-eth1.network	Конфигурация интерфейса eth1 (порт Ethernet-1)

Откройте файл для конфигурации соответствующего порта в любом удобном для Вас редакторе (например, nano или vim) и откорректируйте значения. Рассмотрим содержимое файла /etc/systemd/network/10-eth0.network у ПЛК,

```
[Match]
```

```
Name=eth0
```

```
[Network]
```

```
Address=192.168.1.10/24
```

Для изменения IP-адреса и маски подсети измените содержимое значения Address. Стоит отметить, что маска подсети обычно именуется в традиционной форме, например: 255.255.255.0, однако наряду с этим используется формат CIDR, представляющий более краткую запись, например /24. В указанных файлах используется именно формат записи масок CIDR.

Часто используемые маски подсети

Маска в традиционной форме	Маска в формате CIDR
255.0.0.0	/8
255.255.0.0	/16
255.255.255.0	/24
255.255.255.128	/25
255.255.255.252	/30

Если Вам требуется задать 2 адреса DNS **8.8.8.8** и **8.8.4.4**, а также gateway **192.168.1.1** следует дописать опции в конец файла, например:

```
[Match]
Name=eth0
```

```
[Network]
Address=192.168.1.10/24
Gateway=192.168.1.1
DNS=8.8.8.8
DNS=8.8.4.4
```

Вы можете настроить дополнительные маршруты через секцию [Route]:

```
[Route]
Gateway=192.168.1.1
Destination=10.0.0.0/8
```

Здесь указан маршрут для подсети **10.0.0.0/8** через шлюз **192.168.1.1**.

Если вы хотите использовать DHCP вместо статического IP-адреса, замените секцию [Network]:

```
[Network]
DHCP=yes
```

Если вы хотите назначить фиксированный MAC-адрес интерфейсу, добавьте параметр MACAddress:

```
[Link]
```

```
MACAddress=00:1A:2B:3C:4D:5E
```

После внесения необходимых изменений нужно сохранить файл (если редактор vim, то нажмите клавишу ESC, наберите :wq, нажмите клавишу ENTER; если редактор nano, то нажмите комбинацию клавиш CTRL+O, нажмите клавишу ENTER, выйдите из редактора nano по нажатию сочетания CTRL+X). Новая конфигурация будет принята системой после перезагрузки сетевого сервиса systemd-networkd, для этого требуется ввести команду

```
$ systemctl restart systemd-networkd
```

Для просмотра текущей конфигурации порта Ethernet-0 введите:

```
$ ip addr show eth0
```

5.4.2 Конфигурирование системных часов и часового пояса

На материнской плате ПЛК Softitech S01.PLC.01 есть микросхема RTC (real time clock), питающаяся от собственного источника питания (элемент CR2032), что гарантирует работу встроенных часов независимо от наличия внешнего питания ПЛК. Для просмотра текущего времени, фиксируемого аппаратными часами, введите:

```
$ hwclock
```

Помимо аппаратных часов ОС Softitech Embedded Linux имеет системные часы, которые синхронизируются с аппаратными при загрузке ОС, после чего и внутренние и аппаратные часы работают друг от друга независимо. Узнать показание системных часов можно с помощью команды

```
$ date
```

Для установки системного времени следует ввести команду:

```
$ date -s '2025-04-22 16:45:10'
```

При этом, если поменять системное время/дату, следует также вручную передать эти значения и в аппаратные часы, в противном случае после перезагрузки будет восстановлено время из RTC.

Команды для работы с временем, датой и часовыми поясами

Команда	Описание
date	отображает текущее системное время и дату

<code>date -s 'YYYY-MM-DD HH:MM:SS'</code>	устанавливает новые системные время и дату, YYYY – год, MM – месяц, DD – день, HH – час, MM – минута, SS – секунда
<code>timedatectl status</code>	отображение текущего времени, даты, часового пояса
<code>timedatectl list-timezones</code>	отображение списка доступных часовых поясов
<code>timedatectl set-timezone Asia/Yekaterinburg</code>	установка нового часового пояса, в примере – Asia/Yekaterinburg
<code>hwclock</code>	показывает текущую дату и время на RTC
<code>hwclock -s</code>	установить/синхронизировать время с часов реального времени (RTC) на системные часы
<code>hwclock -w</code>	установить/синхронизировать время с системных часов на часы реального времени

Стоит отметить, что в аппаратных часах время может храниться в формате локального времени, или в формате UTC. Настройка правильного формата времени в RTC (UTC или локальное время) предотвращает ошибки при переключении часовых поясов или переходе на летнее/зимнее время, поэтому по умолчанию время на RTC хранится в формате UTC, переключение режима хранения времени осуществляется командой:

```
$ timedatectl set-local-rtc <0|1>
```

где параметр «0» означает что время в RTC будет храниться в формате UTC (рекомендуется), а «1» - в формате локального времени.

Проверить статус можно командой

```
$ timedatectl status
```

Об этом статусе будет говорить строка «RTC in local TZ»: no – UTC, yes – локально.