

Акционерное общество  
«Научно-исследовательский и проектно-конструкторский институт  
информатизации, автоматизации и связи на железнодорожном транспорте»  
(АО «НИИАС»)

**УТВЕРЖДАЮ**



Первый заместитель  
Генерального директора

Е. Н. Розенберг

" 26 " сентября 20 23 г.

**Тестирующий комплекс  
на базе аппаратуры системы АБТЦ-МШ**

**Мульти СТ**

**Программа мультисервисного терминала**

Программное обеспечение

**Руководство оператора**

Лист утверждения  
RU.НАБП.62.01.29.037 34 ЛУ

Начальник комплекса  
НТК СУ и ОБДП

 А. В. Марков

" 26 " сентября 20 23 г.

УТВЕРЖДЕН

RU.НАБП.62.01.29.037 34 ЛУ

**Тестирующий комплекс  
на базе аппаратуры системы АБТЦ-МШ**

**Мульти СТ  
Программа мультисервисного терминала**

Программное обеспечение

Руководство оператора

RU.НАБП.62.01.29.037 34

Листов 13

## Содержание

1 Назначение .....	3
2 Конфигурация Мульти СТ: .....	4
3 Установка программы.....	5
4 Запуск программы .....	5
5 Использование программы.....	5
5 Команда «Restart».....	8
6 Команда «Status» .....	9
7 Команда «Stop».....	10
8 Команда «Start» .....	11
9. Неисправности и совершенствование ПО .....	12
Лист регистрации изменений.....	13

## **1 Назначение**

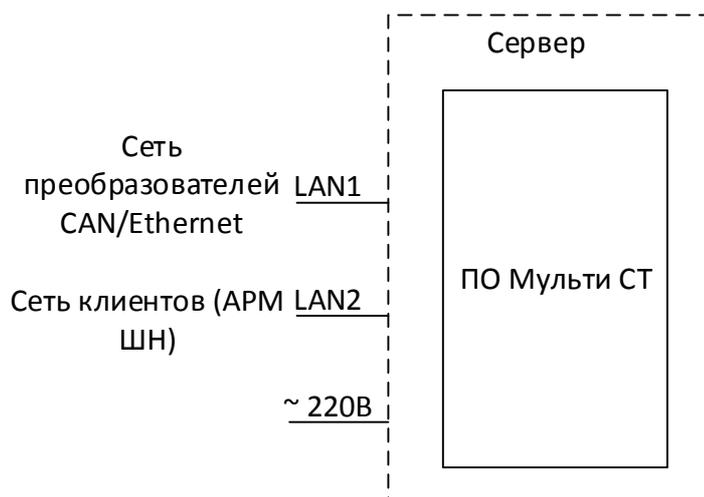
Программное обеспечение «Мульти СТ» входит в состав тестирующего комплекса системы АБТЦ-МШ и позволяет осуществлять функции чтения, и передачи по запросу данных CAN с использованием преобразователей CAN/Ethernet.

ПО «Мульти СТ» реализует следующие основные функции:

- прием информации через Ethernet интерфейс;
- организация серверной части;
- передача в реальном времени CAN потока через Ethernet всем подключённым клиентам;
- вывод на экран диагностической информации по запросу оператора.

## 2 Конфигурация Мульти СТ:

- Сервер с двумя портами LAN;
- Операционная среда Linux.



Мульти СТ портом LAN1 подключен к сети преобразователей CAN/Ethernet. В конфигурации программы при её поставке в составе контейнера определенным образом сопоставлены IP адреса преобразователей CAN/Ethernet с комплектами системы АБТЦ-МШ.

Порт LAN2 используется для передачи данных клиентам, таким как: АРМ ШН, ДТ, РМЭ СЦБ, Диагностическому шлюзу.

Программа осуществляет передачу данных от CAN/Ethernet к клиентам, преобразуя протокол увязки и рассылая эти данные с определенных IP адресов, назначенных порту LAN2.

### 3 Установка программы

Перед установкой программы на сервере должна быть установлена операционная система с ядром Linux.

В операционной системе должна быть установлена программа Docker для поддержки контейнеризации.

Интерфейс LAN1 должен быть настроен для автоматического получения параметров от DHCP сервера.

Интерфейс LAN2 должен быть настроен для статической IP адресации без использования DHCP сервера в сети. IP адреса, назначенные интерфейсу должны находиться в диапазоне от 192.168.1.1 до 192.168.1.17.

Развернуть контейнер с использованием скрипта установки и запустить его.

### 4 Запуск программы

Программа представляется в виде службы multyst операционной системы. Эта служба запускается автоматически при старте контейнера.

### 5 Использование программы

После запуска программы можно пользоваться программой по назначению без дополнительных действий.

Поскольку программа выполняется на сервере, то доступ к диагностической информации о программе имеется через протокол SSH. Для подключения SSH может использоваться любой клиент SSH, в настоящем РО описывается подключение с использованием общедоступного приложения Putty, в котором предварительно сконфигурировано подключение SSH с именем «ip» к адресу 192.168.1.1:22 с АРМ ШН.

На рабочем столе навести курсор манипулятора «мышь» на ярлык «putty»



и нажать дважды левой клавишей. Появится окно конфигурации

программы (Рисунок 1). В этом окне в поле выбрать строчку «ip» и нажать кнопку «Load».

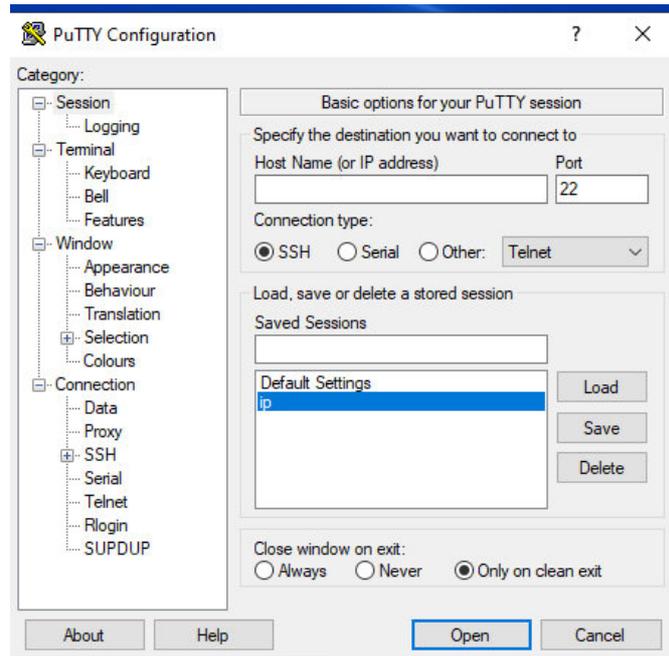


Рисунок 1 – Окно конфигурации программы шаг 1

В строке «Host name» появиться IP-адрес (Рисунок 2), далее нажать кнопку «Open». Для сохранения конфигурации (в случае, если она не была сохранена ранее, то необходимо вручную ввести IP-адрес первого сервисного терминала 192.168.1.1, в поле Saved Sessions ввести «ip» и нажать кнопку «Save».

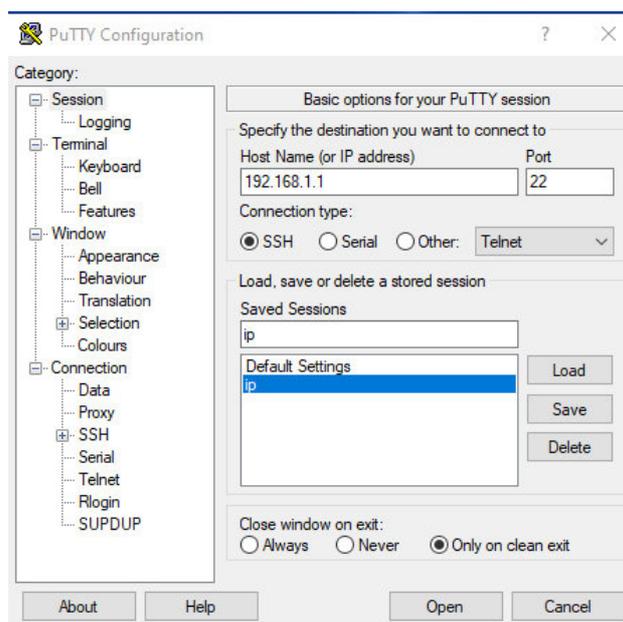


Рисунок 2 – Окно конфигурации программы шаг 2

Появляется окно для ввода логина и пароля (Рисунок 3).

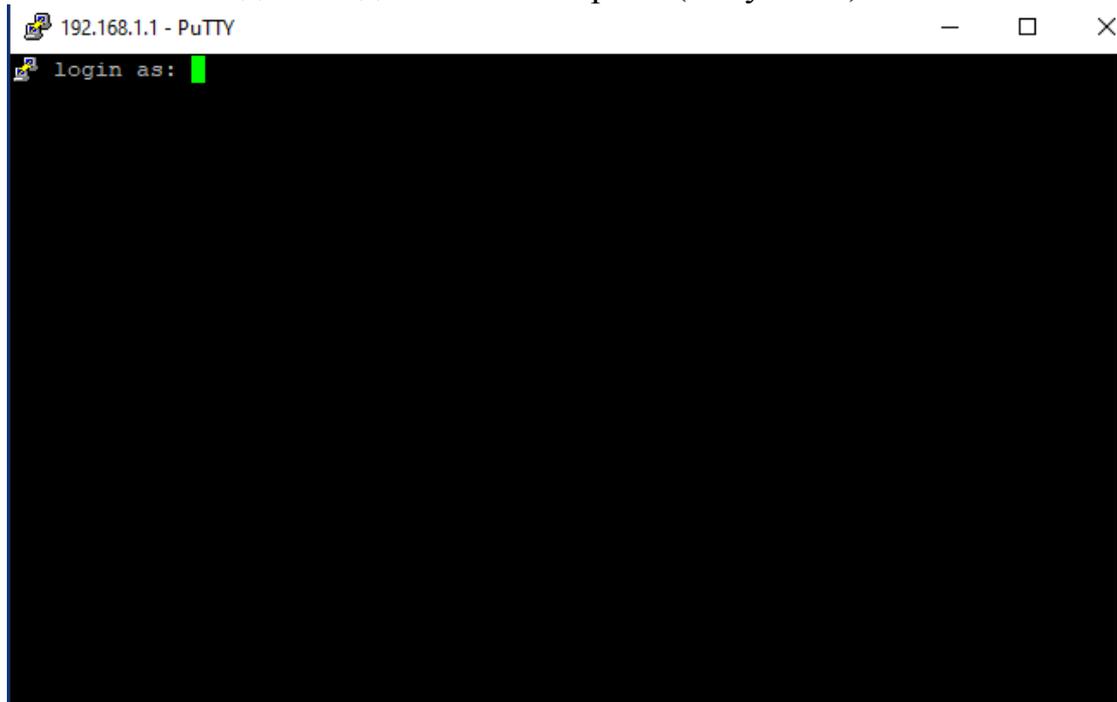


Рисунок 3 –Окно ввода логина и пароля

В появившемся окне вводим логин: admin.

Далее вводим пароль пользователя. По умолчанию для первого запуска пароль admin.

На экране появится окно приветствия и конфигурацией сети (Рисунок 4).

```
IPv4 address for enp0s8: 192.168.1.6
IPv4 address for enp0s8: 192.168.1.7
IPv4 address for enp0s8: 192.168.1.8
IPv4 address for enp0s8: 192.168.1.9
IPv4 address for enp0s8: 192.168.1.10
IPv4 address for enp0s8: 192.168.1.11
IPv4 address for enp0s8: 192.168.1.12
IPv4 address for enp0s8: 192.168.1.13
IPv4 address for enp0s8: 192.168.1.14
IPv4 address for enp0s8: 192.168.1.15
IPv4 address for enp0s8: 192.168.1.16
IPv4 address for enp0s8: 192.168.1.17

145 updates can be installed immediately.
2 of these updates are security updates.
To see these additional updates run: apt list --upgradable

Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your
Internet connection or proxy settings

Last login: Mon Jul 31 12:26:36 2023 from 192.168.1.29
admin@multystb:~$ sudo vi /usr/local/bin/multyst/config.xml
```

Рисунок 4 – Окно приветствия

## 5 Команда «Restart»

В ситуации, когда завис сервис «Мульти СТ», не реагирует на команды и Status показывает «Failed» необходимо перезапустить сервис. Для этого в строке (Рисунок 5) ввести команду «sudo systemctl restart multyst.service», далее нажать «Enter», ввести пароль пользователя и снова «Enter». Сервис перезагрузится.

```
admin@multysta:~$ sudo systemctl restart multyst.service
[sudo] password for admin:
admin@multysta:~$ █
```

Рисунок 5 – Команда «Restart»

## 6 Команда «Status»

Для просмотра статуса сервиса необходимо ввести команду «`sudo systemctl status multyst.service`». Индикация статуса отобразится зеленым цветом в строке «Active:» (Рисунок 6).

Возможные состояния:

- норма active(running);
- запускается активация activating;

```
admin@multysta:~$ sudo systemctl status multyst.service
● multyst.service - MultySt Service
   Loaded: loaded (/etc/systemd/system/multyst.service; enabled; vendor pre
   Active: active (running) since Wed 2023-07-26 11:01:28 UTC; lmin 45s ago
   Main PID: 11071 (service)
     Tasks: 37 (limit: 596)
    Memory: 21.6M
    CGroup: /system.slice/multyst.service
           └─11071 /bin/sh /usr/local/bin/multyst/service
             └─11077 /usr/bin/dotnet /usr/local/bin/multyst/MultiST.dll

Jul 26 11:01:39 multysta service[11077]: 2023-07-26 11:01:39.1884|INFO|[Nias>
Jul 26 11:01:39 multysta service[11077]: 2023-07-26 11:01:39.1919|INFO|[Nias>
Jul 26 11:01:58 multysta service[11077]: 2023-07-26 11:01:58.5516|WARN|Пытаем>
Jul 26 11:01:58 multysta service[11077]: 2023-07-26 11:01:58.6054|INFO|[Nias>
lines 1-20/20 (END)
```

Рисунок 6 – Команда «Status»

## 7 Команда «Stop»

Перед выключением питания, перед физическим перезагрузкой Сервера необходимо остановить сервис. В строку ввести «sudo systemctl stop multyst.service» (Рисунок 7).

```
admin@multysta:~$ sudo systemctl stop multyst.service
admin@multysta:~$ sudo systemctl status multyst.service
• multyst.service - MultySt Service
   Loaded: loaded (/etc/systemd/system/multyst.service; enabled; vendor pre>
   Active: inactive (dead) since Wed 2023-07-26 11:05:38 UTC; 24s ago
   Process: 11071 ExecStart=/usr/local/bin/multyst/service (code=killed, sig>
   Main PID: 11071 (code=killed, signal=TERM)

Jul 26 11:01:39 multysta service[11077]: 2023-07-26 11:01:39.1884|INFO|[Nias>
Jul 26 11:01:39 multysta service[11077]: 2023-07-26 11:01:39.1884|INFO|[Nias>
Jul 26 11:01:39 multysta service[11077]: 2023-07-26 11:01:39.1919|INFO|[Nias>
Jul 26 11:01:39 multysta service[11077]: 2023-07-26 11:01:39.1919|INFO|[Nias>
Jul 26 11:01:39 multysta service[11077]: 2023-07-26 11:01:39.1919|INFO|[Nias>
Jul 26 11:01:58 multysta service[11077]: 2023-07-26 11:01:58.5516|WARN|Пытаем>
Jul 26 11:01:58 multysta service[11077]: 2023-07-26 11:01:58.6054|INFO|[Nias>
Jul 26 11:05:38 multysta systemd[1]: Stopping MultySt Service...
Jul 26 11:05:38 multysta systemd[1]: multyst.service: Succeeded.
Jul 26 11:05:38 multysta systemd[1]: Stopped MultySt Service.
lines 1-16/16 (END)
```

Рисунок 7 – Команда «Stop»

## 8 Команда «Start»

В строку ввести «sudo systemctl start multyst.service» (Рисунок 8).

```
admin@multysta:~$ sudo systemctl start multyst.service
admin@multysta:~$ sudo systemctl status multyst.service
• multyst.service - MultySt Service
   Loaded: loaded (/etc/systemd/system/multyst.service; enabled; vendor pre>
   Active: active (running) since Wed 2023-07-26 11:07:58 UTC; 3s ago
   Main PID: 11187 (service)
   Tasks: 22 (limit: 596)
   Memory: 16.9M
   CGroup: /system.slice/multyst.service
           └─11187 /bin/sh /usr/local/bin/multyst/service
             └─11190 /usr/bin/dotnet /usr/local/bin/multyst/MultiST.dll

Jul 26 11:07:58 multysta systemd[1]: Started MultySt Service.
lines 1-11/11 (END)
```

Рисунок 8 – Команда «Start»

## **9. Неисправности и совершенствование ПО**

При возникновении неисправностей необходимо перезапустить ПО командой «Restart». При необходимости перезапустить сервер. При сохраняющейся неисправности необходимо обратиться к АО «НИИАС» - разработчику программного обеспечения.

Совершенствование программного обеспечения, в том числе расширение функционала, улучшение производительности, графического интерфейса, устранение ошибок производится по результатам тестирования, согласования и утверждения технического задания на доработку программного обеспечения. Разработка, доработка программного обеспечения осуществляется АО «НИИАС» с привлечением инженера-программиста, обладающего необходимыми и достаточными знаниями для выполнения своих обязанностей в соответствии с должностной инструкцией, утвержденной в установленном порядке. Обязательным условием является знание языков программирования, умение работать с алгоритмами, представленными в виде графов переходов и блок-схем.

