

Режим Advanced в конвертере Z-397 WEB

Устройство, подключённое к локальной сети, можно конфигурировать по протоколу TELNET. Для этого необходимо чтобы устройство было правильно сконфигурировано для работы в данной локальной сети, согласно «Руководства по эксплуатации...».

Поиск конвертеров в сети

Для поиска всех устройств, подключённых к локальной сети реализован следующий протокол:

1. Программа, осуществляющая поиск, посылает широковещательный UDP пакет с портом назначения 9000. (Широковещательный пакет – пакет с IP адресом назначения равным 255.255.255.255). В качестве данных, передаваемых в пакете, передаётся строка:

```
SEEK Z397IP
```

2. Устройство отвечает UDP пакетом на IP адрес компьютера, пославшего запрос, UDP порт получателя в этом пакете равен порту отправителя в пакете запроса. В ответном пакете содержится информация об типе устройства, его серийном номере, открытых TCP портах и т.д. Формат информационной строки:

```
Z397WEB-VCP-SW:3.0.69 SN000008 L1_Port:1000 L2_Port:1001 L1_Conn:192.168.1.31
L2_Conn:0.0.0.0 Lock:1
```

| Параметр | Описание |
|----------------------|---|
| Z397WEB | Модель устройства |
| VCP | Функциональные возможности (ALL/VCP/WEB) |
| SW:3.0.69 | Версия прошивки |
| SN000008 | Серийный номер |
| L1_Port:1000 | TCP порт первой линии RS485 |
| L2_Port:1001 | TCP порт второй линии RS485 |
| L1_Conn:192.168.1.31 | Первая линия соединена с компьютером, имеющим IP 192.168.1.31 |
| L2_Conn:0.0.0.0 | Вторая линия не имеет активного соединения |
| Lock:1 | Состояние переключателя "LOCK" |

3. Программа, осуществляющая поиск, получив ответный пакет, анализирует пришедшие данные об устройствах и добавляет отправителя в список найденных устройств.

Установка соединения

Для установки соединения с конвертером без использования драйверов виртуального COM порта необходимо:

1. Если конвертер работает в режиме сервера.

- Создать клиентский сокет.
- Установить параметры сокета в соответствии с IP адресом и TCP портом конвертера.
- Установить соединение с конвертером.

Пример для OS Windows:

```
#define PORT 1000
#define SERVERADDR "192.168.1.10"

SOCKET my_sock;
client_sock=socket ( AF_INET, SOCK_STREAM, 0 );

if ( client_sock < 0 )
{
printf ( " Socket ( ) error %d \n ", WSAGetLastError ( ) );
return -1;
}

sockaddr_in dest_addr;
dest_addr.sin_family=AF_INET;
dest_addr.sin_port=htons ( PORT );
dest_addr.sin_addr=inet_addr ( SERVERADDR)

if (connect( client_sock, ( sockaddr * ) &dest_addr, sizeof ( dest_addr ) ) )
{
printf ( " Connect error %d \n ", WSAGetLastError ( ) );
return -1;
}
printf ( " Соединение с %s успешно установлено\r\n" , SERVERADDR );
```

Пример для *nix

```
int
open_remote_host(char *host, int port)
{
struct sockaddr_in rem_addr;
int len, s, x;
struct hostent *H;
int on = 1;

H = gethostbyname(host);
if (!H)
return (-2);

len = sizeof(rem_addr);
```

```
s = socket(AF_INET, SOCK_STREAM, 0);
if (s < 0)
return s;

setsockopt(s, SOL_SOCKET, SO_REUSEADDR, &on, 4);

len = sizeof(rem_addr);
memset(&rem_addr, '\0', len);
rem_addr.sin_family = AF_INET;
memcpy(&rem_addr.sin_addr, H->h_addr, H->h_length);
rem_addr.sin_port = htons(port);
x = connect(s, (struct sockaddr *) &rem_addr, len);
if (x < 0) {
close(s);
return x;
}
set_nonblock(s);
return s;
}
```

- После соединения произвести конфигурацию конвертера.

2. Если конвертер работает в режиме клиента.

- Создать серверный сокет
- Установить порт сокета в соответствии с портом, настроенным в конвертере
- Ожидать соединения с конвертером.

Пример для OS Windows:

```
#define PORT 1000
#define BACKLOG 4

SOCKET my_sock;
server_sock=socket ( AF_INET, SOCK_STREAM, 0 );

if ( server_sock < 0 )
{
printf (" Socket ( ) error %d \n ", WSAGetLastError ( ) );
return -1;
}

sockaddr_in srv_addr;
srv_addr.sin_family=AF_INET;
srv_addr.sin_port=htons ( PORT );
srv_addr.sin_addr.s_addr = INADDR_ANY;
```

```
if(bind (server_socket , (LPSOCKADDR )&srv_addr, sizeof(srv_address)) == SOCKET_ERROR )
{
    closesocket (server_socket);
    return -1;
}

listen(server_socket,BACKLOG);
```

Пример для *nix

```
create_server_sock(char *addr, int port)
{
    int addrlen, s, on = 1, x;
    static struct sockaddr_in client_addr;

    s = socket(AF_INET, SOCK_STREAM, 0);
    if (s < 0)
        err(1, "socket");

    addrlen = sizeof(client_addr);
    memset(&client_addr, '\0', addrlen);
    client_addr.sin_family = AF_INET;
    client_addr.sin_addr.s_addr = inet_addr(addr);
    client_addr.sin_port = htons(port);
    setsockopt(s, SOL_SOCKET, SO_REUSEADDR, &on, 4);
    x = bind(s, (struct sockaddr *) &client_addr, addrlen);
    if (x < 0)
        err(1, "bind %s:%d", addr, port);

    x = listen(s, 5);
    if (x < 0)
        err(1, "listen %s:%d", addr, port);
    syslog(LOG_NOTICE, "listening on %s port %d", addr, port);

    return s;
}
```

- После соединения произвести конфигурацию конвертера.

Настройка режима работы устройства

После установки соединения конвертер работает в режиме "NORMAL". В этом режиме все данные передаются на линию RS-485. Скорость линии устанавливается в соответствии с конфигурационным файлом устройства. Для изменения скорости линии RS-485, а также для перевода конвертера в режим "[ADVANCED](#)", используется протокол [NVT](#).

Внимание: Для работы по протоколу NVT необходимо чтобы первый пакет, принятый конвертером после установки соединения, содержал NVT команду.

- NVT команды вставляются в поток данных передаваемых по TCP/IP.
- Каждая NVT команда предваряется префиксом в виде символа <IAC>
- Некоторые основные команды (EOF, ABORT, BRK, AYT, NOP, EC) передаются одним байтом, остальные команды передаются в формате <IAC><SB><КОМАНДА><IAC><SE>
- Устройство фильтрует и обрабатывает все команды NVT и они не попадают в выходной поток данных.
- Для посылки символа с кодом 0xFF в потоке данных, его надо удвоить (0xFF 0xFF), для того чтобы он попал в выходной поток.

| Dec | HEX | Shortcut | Описание |
|-----|-----|----------|----------------|
| 240 | F0 | SE | Конец команды |
| 241 | F1 | NOP | No Operation |
| 246 | F6 | AYT | Are You There |
| 250 | FA | SB | Начало команды |
| 255 | FF | IAC | Data Byte 255 |

В данный момент устройство поддерживает только команды установки скорости порта. Остальные команды фильтруются из выходного потока, но не обрабатываются. Для установки скорости порта необходимо послать следующую команду:

| IAC | SB | COM_PORT_OPTION | CAS_SET_BAUDRATE | XX XX XX XX | IAC | SE |
|-----|----|-----------------|------------------|------------------|-----|----|
| FF | FA | 2C | 01 | Скорость 4 байта | FF | F0 |

Например: для скорости 9600 надо послать последовательность:

```
FF FA 2C 01 00 00 25 80 FF F0
```

Для перевода конвертера в режим "[ADVANCED](#)" необходимо установить скорость линии 230400:

```
FF FA 2C 01 00 03 84 00 FF F0
```

Если режим "[ADVANCED](#)" устанавливается для первой линии (iPort1) и при этом вторая линия свободна (отсутствует TCP соединение с ней), то конвертер автоматически переходит в режим объединения линий. В этом режиме работа происходит с обеими линиями как с одной, то есть все контроллеры, подключённые к линиям конвертера доступны для опроса по одному TCP соединению.