

# Interfacing

- [E1/T1 Interface](#)
- [FrameRelay over E1](#)

# E1/T1 Interface

## Why not TDMoIP / CESoIP?

Usual E1/T1 over IP protocols use generic frames over IP transfer that makes more than 2 megabits constant rate with up to 8000 packets per second. That approach:

- not suitable for distributed sites with limited internet access
- makes constant parasite load on the servers
- requires delays to compensate jitters
- has issues with asynchronous-to-synchronous transition

For such kind of integrations we prefer to have our own agent software on-site with hardware E1/T1 interfaces.

- good timings driven by E1/T1 clock or GPS
- good framing due to synchronous composition on the site
- less traffic due to transfer only valuable demultiplexed information

## icE1usb



We have chosen [Osmocom icE1usb](#) as a primary interface for our projects for several reasons:

- good compatibility with big range of hardware hosts due to use of USB
- in opposite to most of DAHDI interfaces it is in production
- open-source hardware design
- [open-source user-space driver](#), no need for the kernel driver
- suitable API for our purposes (at least full-frame mode)
- available at [Sysmocom's web shop](#)

User manual can be found [here](#).

# NOTICE

**Please be careful! Or your device will be damaged!** Front-panel RJ45 connectors accepts 120 Ohm E1. Back-panel RJ45 is for GPIO and accepts TTL only. Don't try to connect the device to Ethernet. USB-C port should be connected to computer, since it's just an interface. Minijack connector is only to reprogram firmware. Also please configure properly internal jumpers to master (NE) or slave (NT) E1 mode.

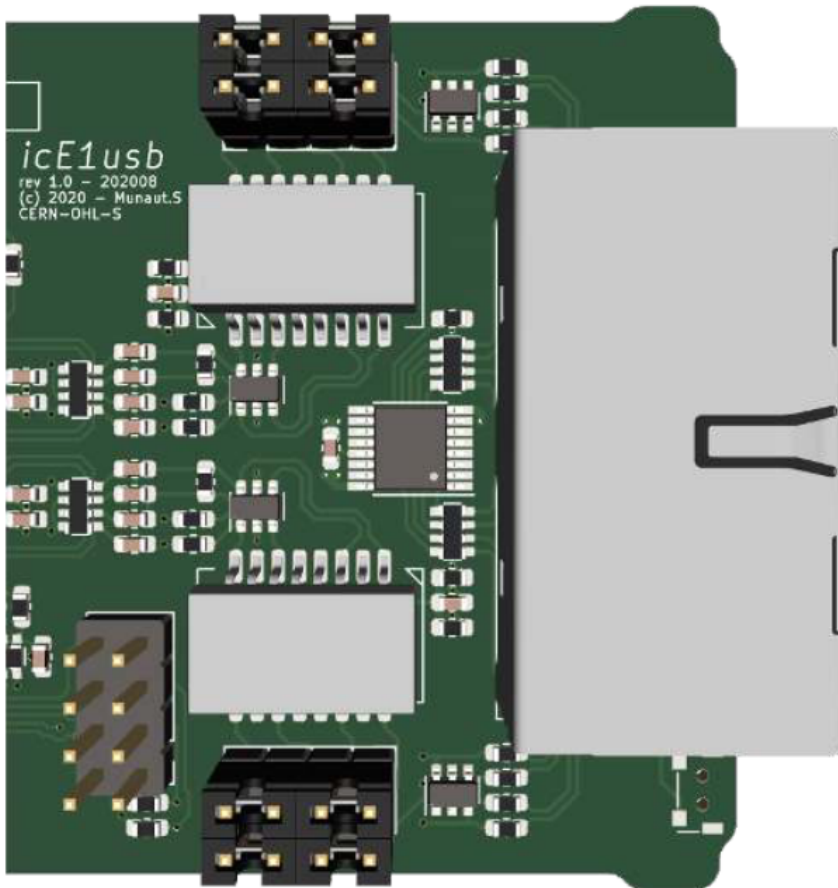
PLEASE READ [USER MANUAL](#) FIRST.

## Hardware configuration

Almost in all our cases you need this interface in NT (Network Termination) mode. In this mode the interface works as a network side and you can use a normal Ethernet cable for connection (no crossing required).

*To change the mode, unscrew the 2 PH0 screws on the side with the 2 RJ45 jacks. You'll also need to unscrew the nut on the SMA jack on the other side. After that, the cover plate and rubber gasket around the 2 RJ45 jacks can be removed. The PCB can be slid out of the case.*

Then swap the jumpers so they match this picture:



**NT mode**

# Driver installation

You can get [Osmocom E1D](#) from [Osmocom's Debian repository](#) (amd64, arm64).

```
sudo apt install extrepo
sudo extrepo enable osmocom-latest
sudo apt update
sudo apt install -y osmo-e1d
```

# Driver configuration

/etc/osmocom/osmo-e1d.cfg

```
log syslog daemon
logging level force-all fatal
e1d
interface 0 icE1usb
usb-serial [interface serial number]
line 0
line 1
```

More information about logging section can be found [here](#), [here](#) and [here](#).  
(If you care about configuration file format, please check the sources [here](#)).

How to find a serial number of connected icE1usb:

```
sudo lsusb -d 1d50:6145 -v 2> /dev/null | grep iSerial
```

Reload a service:

```
sudo systemctl restart osmo-e1d
```

You can also test your configuration from the command line:

```
sudo osmo-e1d -c /etc/osmocom/osmo-e1d.cfg
```

## Recomended settings

### /etc/rsyslog.d/99-e1d.conf

```
if ($programname contains "osmo-e1d") and (($msg contains "Received Only 0 bytes") or ($msg contains "TS
read underflow")) then {
    ~/dev/null
    stop
}
```

### /etc/systemd/system/osmo-e1d.service.d/override.conf

```
[Service]
TimeoutStopSec=2s
```

# Issues with the second E1 port

*Please note that on many XHCI host controllers there seem to be implementation flaws in the XHCI host controller firmware preventing the activation of both icE1usb ports simultaneously. The XHCI controller firmware erroneously claims that there is insufficient bus bandwidth. However, the same icE1usb hardware/firmware works perfectly fine with OHCI, UHCI and EHCI host controllers. See [https://osmocom.org/projects/e1-t1-adapter/wiki/Isochronous\\_USB\\_Issues](https://osmocom.org/projects/e1-t1-adapter/wiki/Isochronous_USB_Issues) for a user-maintained list of USB hosts / controllers and whether or not they work with two E1 ports.*

In most installations it's not a real issue, because of need in a single port only.

We have tested it with following equipment:

- Raspberry Pi 3, 4, CM4 - only first E1 port is working
- Raspberry Pi 5 - both E1 ports are working when connected to blue USB port

# FrameRelay over E1

## FrameRelay-over-E1 Daemon (FRED)

### Intro

- Gateway software to run on on-site E1 connection (**see article [E1/T1 Interface](#)**)
- Bridges IPv4/IPv6/Ethernet packets between Linux kernel and FrameRelay over E1 ([RFC 2427](#), [RFC 2590](#))
- Suitable to connect [Motorola EBTS](#) (**can be used instead of Cisco routers with E1 card installed**)
- Supports FRF.12 (inner and outer) fragmentation for incoming traffic (Frame Relay -> Linux)
- Implements basic DCE-PVC LMI with support of ITU-T Q.933-A, ANSI T1.617-D, GOF (automatic detection)
- Acts via TUN/TAP network interfaces (one per DLCI) on Linux side
- Debian 12 arm64 or amd64, tested on Raspberry Pi CM4, Raspberry Pi 5

### Installation

```
sudo apt install extrepo
sudo extrepo enable osmocom-latest
curl http://packages.tetrapack.online/install/tools/Debian/add-repository.sh | bash
sudo apt install -y osmo-e1d fred
```

### Configuration

Default configuration file is /opt/FRED/default.env

```
#
# Parameters for FRED binary
#

# E1D interface and line in format <interface number>.<line number>
FRED_LINE=0.0

# E1 slot list in format <first slot>-<last slot>
FRED_SLOT_LIST=1-31

# FrameRelay feature list in format <option>[,...]
# FRF.12-OUTER - outer frame sequence counting and fragmentation (incoming frames only supported)
FRED_FEATURE_LIST=

# DLCI list in format <DLCI>:<type>[,...]
# LMI-DCE-PVC
# IP - IP bridge (RFC 2427 for IPv4, RFC 2590 for IPv6)
# ETH - Ethernet bridge (RFC 2427)
FRED_DLCI_LIST=0:LMI-DCE-PVC,16:IP,17:IP

# Path to configuration script to manage IP interface configuration on up and down
FRED_SCRIPT=./dimetra-hook.sh

#
# Parameters for dimetra-hook.sh
#

# Zone ID
DIMETRA_ZONE=1

# Local Site ID
DIMETRA_SITE=12

# FrameRelay DLCIs
DIMETRA_DLCI_PRIMARY=16
DIMETRA_DLCI_SECONDARY=17

# When set, routes will be added to FRR's OSPF prefix list
FRR_PREFIX_LIST=dimetra
```



# Use with Dimetra

FRED package includes prepared scripts to establish connection between EBTS and Dimetra MPVPN network.

```
sudo /opt/FRED/dimetra-setup.sh
```

Best settings for EBTS TSC will be:

```
.sitelink -e1  
.e1config -crdStart 1 -crd 31 -ts16Skip off -portNo 1 -crc on  
reset
```

## Multiple instances

You can have several separated systems connected to a local Linux system. For such case you need to have several .env files for each configuration. use *sudo ./setup.sh install* to register all instances.

Also it's possible to run FRED in command line - *sudo ./run.sh <configuration.env>*