Nokia TRAU (TCSM2): the working state

Nokia TCSM2 is a bank-of-TRAUs apparatus with E1 interfaces.

(Terminology: in Nokia's interpretation, a TRAU is just one voice/data channel; a bank of these TRAU channels is Transcoder and Submultiplexer - TCSM.)

There is now a working TCSM2 setup in Mother Mychaela's lab:

* Put together from individually-bought pieces;
* Getting it into a working state was quite a journey!

But now that it works, we can play with it:

* Basic TRAU functionality: feed it TRAU-UL frames on Ater, decoded output shows up on A as PCM samples; the encoder takes PCM input from A and puts out TRAU-DL frames on Ater.

* ... but TCSM2 also supports TFO, which is the interesting part:

  - TFO protocol
  - TFO transform

Hardware bring-up: the tale of two chassis

Back In The Day: a "proper" TCSM2 rack is 1.8 m tall, 0.8 m wide and 0.5 m deep: too big for most hobbyists, including yours truly. :)

Further consideration: I didn’t have the luxury of picking up a decommissioned setup at zero cost, instead I had to individually buy every single component - hence figuring out the minimum required hardware config was essential.

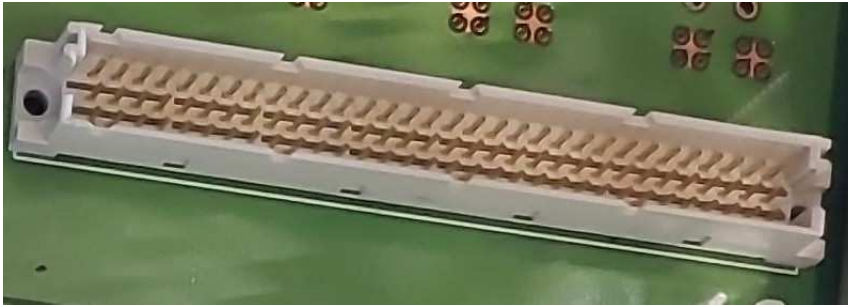
Here is the minimal functional config:

```
+------------+
|  TC1C unit |
+---------+--+
          |
          | interchassis cable
          |
+---------+--+
| ET1TC unit |
+------------+
```

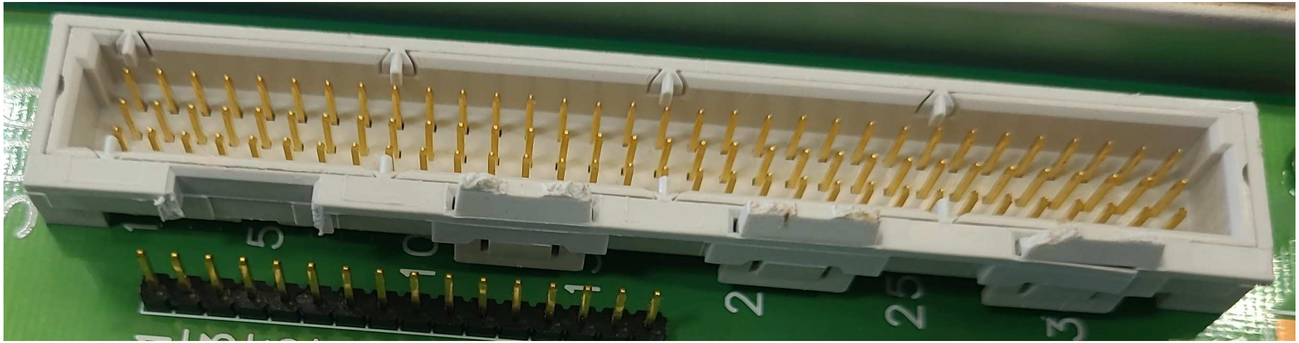
But there are still two separate chassis involved, and a custom interchassis cable had to be constructed!

Making the interchassis cable: connector conspiracy begins

Standard DIN 41612 connector with male pins (TC1C PSC1 slot):

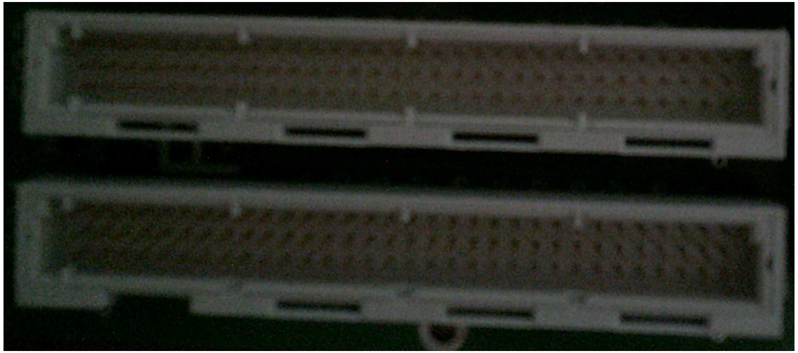


Deceptively similar non-standard connector for interchassis wiring:

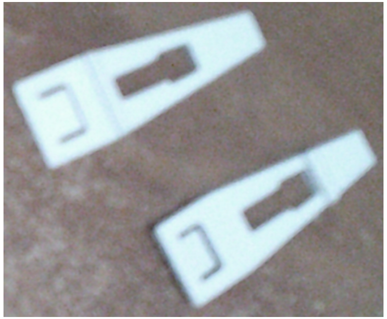


More on these mystery connectors

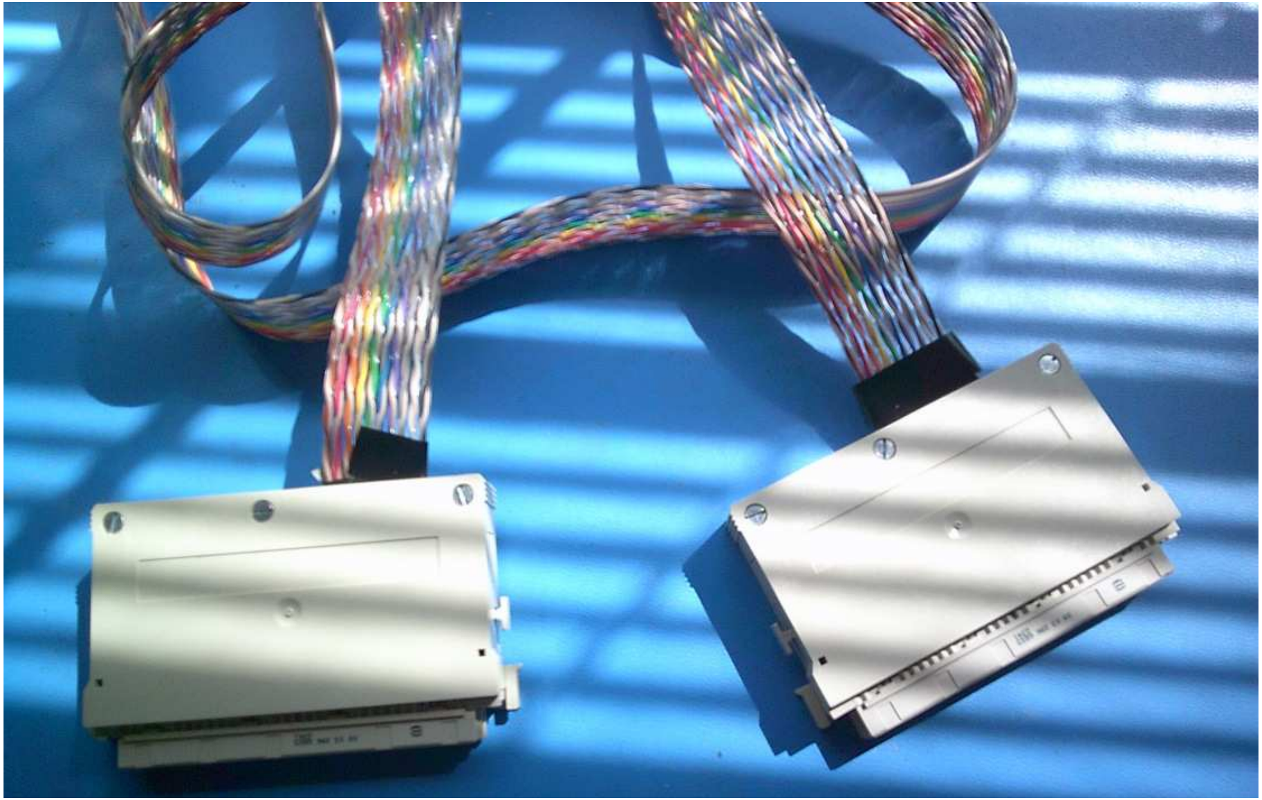
Back face of ET1TC backplane, more connectors of this strange type:



And here are the strange plastic posts that go into those slots on the side:



Finished interchassis cable: behold the beauty



ET2E-* modules for 2xE1 interface: so many variants

	120 ohm balanced E1 (3x16 weird connector)	75 ohm coaxial E1 (Mini-SMB connectors)
1st gen	ET2E (plain)	ET2E-C
2nd gen	ET2E-S	ET2E-SC
3rd gen	ET2E-T	ET2E-TC

The initial quote I got from shields-s offered me the choice of ET2E-C, ET2E-S or ET2E-T, all at the same price of $335 each.

First attempt: I bought ET2E-T. Aside from connector conspiracy, it refused to communicate with TRCO even in self-test mode. Firmware mismatch? Wrong jumper settings? Defective module? Could be anything...

Second try: I bought ET2E-C, and got better luck:
- self-test started passing as soon as I put it in;
- once I got those Mini-SMB connectors sorted out, E1 interfaces came to life!

Nokia TCSM2 firmware structure

TRCO card (main controller):

- 512 KiB immutable boot ROM (OTP chip), 1 MiB writable flash
- supplies firmware from its flash to other modules

Transcoder DSP cards:

- NO on-board flash or EPROM, TRCO pushes firmware on every boot

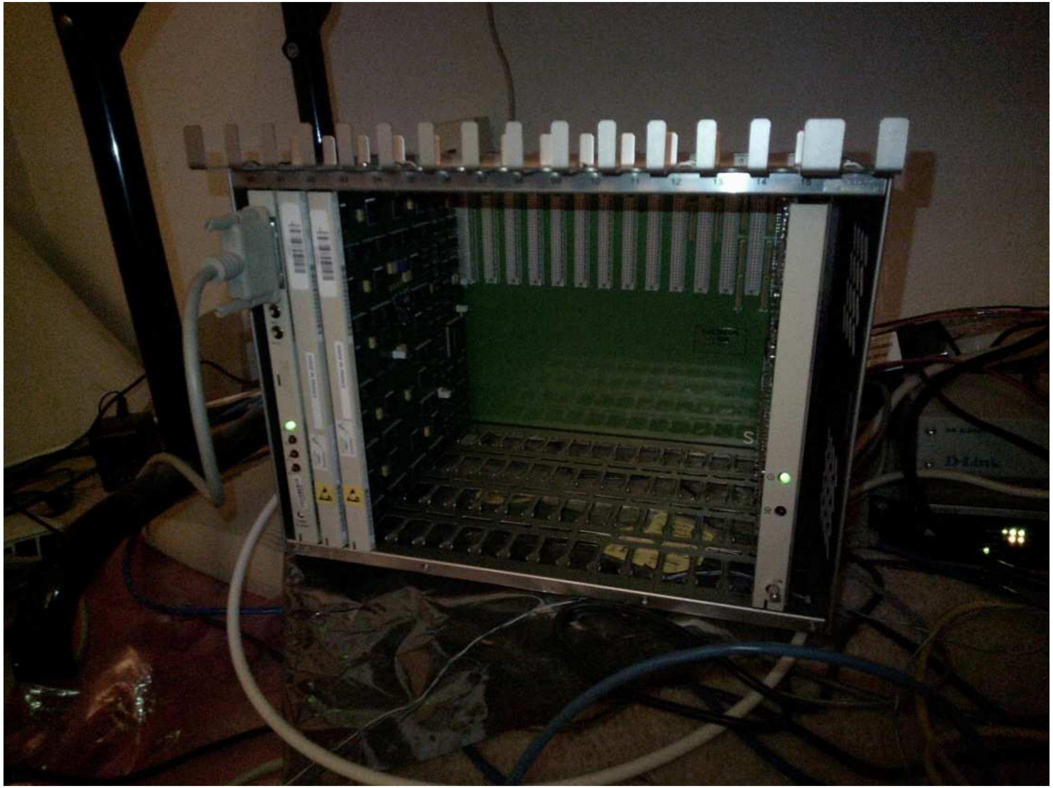
Classic ET2E modules, likely ET2A too:

- 256 KiB immutable boot ROM (UV EPROM chip)
- RAM-based firmware fed from TRCO on every boot via HDLC

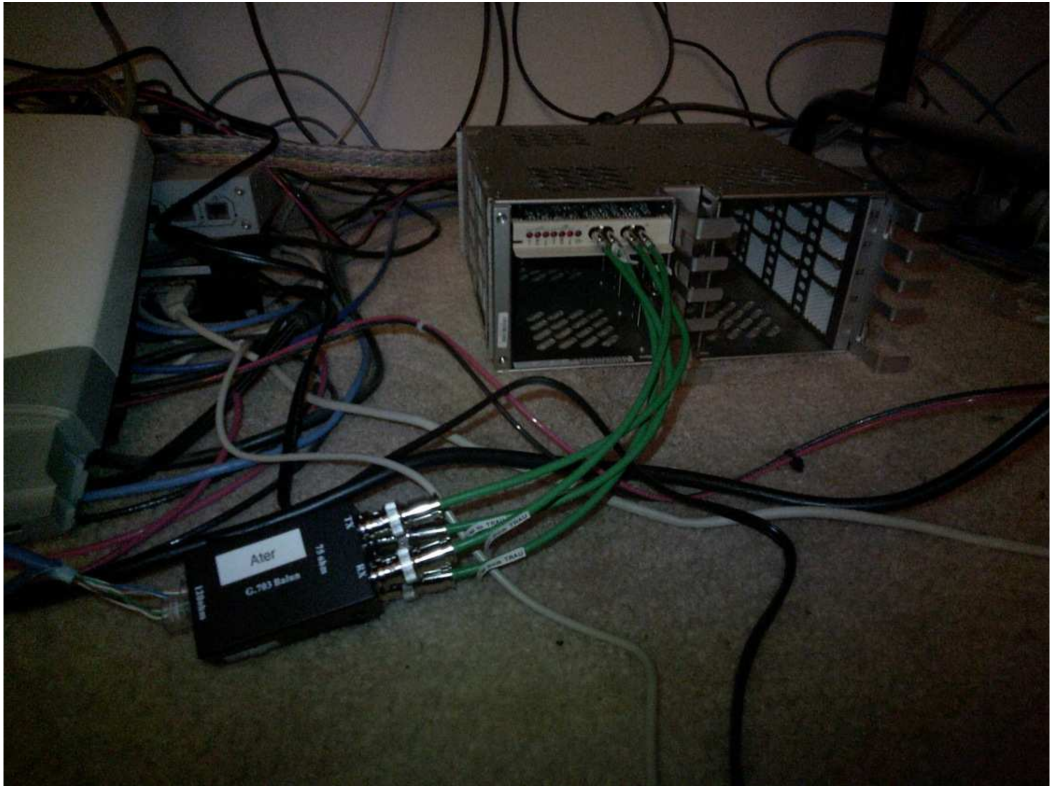
ET2E-T module, the one I never got to work:

- Soldered flash chip on the module, no more socketed EPROM or OTP ROM
- A different fw image for ET2E-T & ET2A-T is present on TRCO
- Is that on-module flash writable in normal operation? Is the architecture based around pushing fw updates into that flash instead of running operational fw in RAM, loaded by an immutable boot ROM?

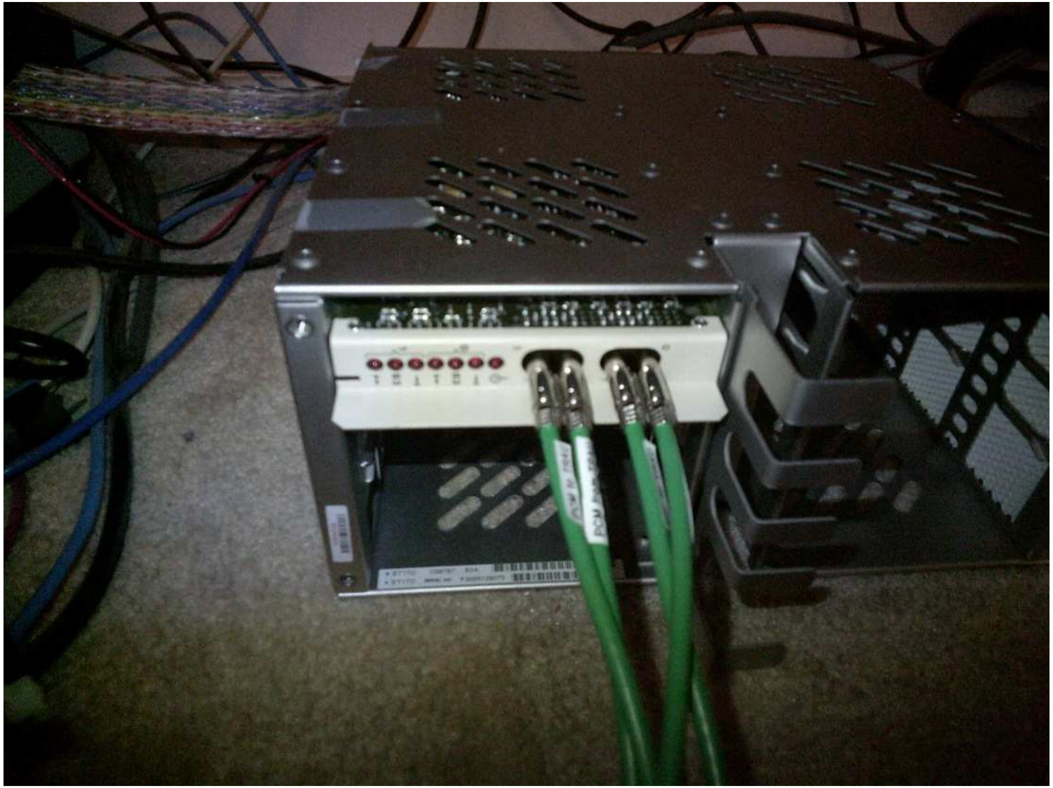
Photos of working setup: TC1C chassis



Photos of working setup: ET1TC chassis and E1 bring-out



Photos of working setup: ET2E-C connections close-up



TCSM2 box in operation

Management interface:

- RS-232 port on TRCO card, connected to a 1980s MicroVAX machine running 4.3BSD UNIX, my favorite 1980s OS - retronetworking paired with retrocomputing!

- 9600 baud, plain ASCII terminal interface.

E1 interfaces:

- Both A and Ater interfaces are connected to an icE1usb adapter;
- The host driving the icE1usb is an RPi5, one of the few that support icE1usb with both E1 ports active;
- All of my custom test programs work through osmo-e1d - not braving DAHDI.

Standalone operation:

- Neither MSC nor BSC are present;
- I do have a Nokia InSite BTS, but it is not used simultaneously with TRAU experiments;
- All experiments are done with ad hoc test programs running on the RPi5.

A and Ater interfaces, essential function of TRAU

```
MSC <-- A i/f --> TCSM <-- Ater i/f --> BSC <-- Abis i/f --> BTS
```

- BSC switches 16 kbit/s or 8 kbit/s channels from Abis to Ater;
- TCSM transcodes or rate-adapts each 16 kbit/s or 8 kbit/s channel on Ater to corresponding 64 kbit/s channel on A interface.

Circuits from MSC to BSC, passing through TCSM, are STATIC!

PCM circuit type A:

- A fixed number of circuits are statically configured for 16 kbit/s submux on Ater, supporting FR/EFR speech and full-rate data.

PCM circuit type B:

- Another fixed block of circuits are statically configured for 8 kbit/s submux on Ater, supporting HR speech and data.

Each MSC-side E1 trunk has to be statically configured as one circuit type. Having only one MSC-side E1 means having to reconfigure the box through RS-232 management i/f each time I switch between playing with FR/EFR or HR!

Additional PCM circuit types

PCM circuit type C:

- Documentation says it shares DSP fw with circuit types D and E, supports both FR/EFR/CSD like type A and HR like type B - but we don’t have this DSP fw.

PCM circuit types D and E:

- Documentation says these circuit types support HSCSD, 2x with type D and 4x with type E - but again, we don’t have this DSP fw image.

PCM circuit type F:

- Same 16 kbit/s submux on Ater as with types A and C, but the DSP fw supports AMR speech.

Choice of fixed circuit blocks for AMR vs non-AMR is once again static!

- Circuit type A supports FR/EFR/CSD, but no AMR;
- Circuit type F supports AMR, but not the others.

Life cycle of a single TRAU channel

With PCM circuit type A, each individual TRAU channel can do FR speech, EFR speech or CSD - how does it know which mode is needed when?

Each TRAU channel comes to life when it receives TRAU-UL frames on Ater, switched to it by the BSC from Abis. The type of these frames tells the TRAU which mode it needs to initialize into.

The TRAU channel remains alive for as long as TRAU-UL frames keep coming - once these frames cease, the channel deactivates and goes back to idle.

During active state:

- Speech encoder takes PCM samples from A i/f and emits TRAU-DL frames on Ater;
- Speech decoder takes TRAU-UL frames Ater and emits decoded PCM samples on A.

During inactive state:

- The Ater subslot assigned to this TRAU emits constant 2’b01 (16 kbit/s) or constant zero bits (8 kbit/s);
- The A timeslot assigned to this TRAU emits constant 0x54 octets.

Basic transcoding function (without TFO)

HRv1 and EFR codecs:

- The TRAU does implement in-band encoder and decoder homing functions, as required by the specs.

- In the decoder direction, the output is NOT a bit-exact match to official A-law test sequences. (Testing only basic speech decoding, no DTXu or BFIs!)

- In light of that mismatch, no encoder test sequences were attempted.

FRv1 codec:

- In-band homing feature is optional in the specs, NOT implemented on this TRAU.

- Lack of in-band homing mechanism means no ability to run test sequences against a "black box" implementation without special "test mode" backdoors.

In light of these findings, the basic (non-TFO) speech transcoding function of this Nokia TRAU becomes rather UNinteresting - no avenues for doing anything fun with it. :(

CSD: the Rate Adaptor part of TRAU

Support for CSD in Nokia TCSM2:

* PCM circuit type A supports data calls on 16 kbit/s submultiplexing, with both 16 kbit/s intermediate rate (TCH/F9.6) and 8 kbit/s intermediate rate (TCH/F4.8 and TCH/F2.4). Both modes tested and found to work as expected.

* PCM circuit type A also supports 14.4 kbit/s data mode; this mode has likewise been tested and found to work correctly, emitting A-TRAU frames on PCM side. Inspired libosmotrau implementation of RAA’ function!

* PCM circuit type B supports half-rate data calls on 8 kbit/s submultiplexing - interesting indeed! Does anyone know if InSite BTS supports HR data too?

* TCSM2 architecture supports multislot HSCSD with circuit types D and E - but we don’t have the necessary firmware.

* PCM circuit type F (the one added for AMR) rejects CSD just like how it rejects FRv1 and EFR - it is really AMR only.

TFO: ending on a cliffhanger

TFO is Tandem-Free Operation - see GSM 08.62 in R98 or R99, or 3GPP TS 28.062 in Rel4 and later.

TFO is also the main reason for my interest in historical GSM TRAU equipment: I spent about $3800 and 4.5 months of work to acquire this Nokia TCSM2 gear and bring it into a working state, all for the primary purpose of playing with its TFO feature.

TFO as implemented in classic GSM TRAUs for FR/HR/EFR codecs is really two topics of interest:

- TFO protocol: how two TRAUs connected by 64 kbit/s PSTN/ISDN discover each other and negotiate the possibility of entering Tandem-Free Operation;

- TFO transform from UL to DL: how the TRAU on each end fills in synthetic "speech" frames for call leg B DL in places where the incoming frame stream from call leg A UL has BFI frame gaps or DTXu pauses.

TFO is such a big topic that I am saving it for a separate presentation: yes, just like how it went in _One Thousand and One Nights_.