Gmail **Ken Eckert <eckertkp@gmail.com>**

Re: [HP-Agilent-Keysight-equipment] E4407B restoration project: EEPROMs

4 messages

Techfreakz <alex@techfreakz.net> Sat, Oct 23, 2021 at 7:27 AM
Reply-To: HP-Agilent-Keysight-equipment@groups.io
To: "HP-Agilent-Keysight-equipment@groups.io" <HP-Agilent-Keysight-equipment@groups.io>

Hello zs,

Sorry for the slow reply, I don't check this group very often.

I put about a month in to learning how the look-up is performed, reverse engineering the EEPROM contents and re-generating the correct contents - quite a feat of engineering!

My most important discovery (late in the process) was that there is a serial (RS232) debug port on the CPU card (its the pin header you can see on the rear panel). During the YTF alignment process, there is lots of information written out this port. I made a little cable to convert from the 2mm(?) pitch header to a 9-way D-type.

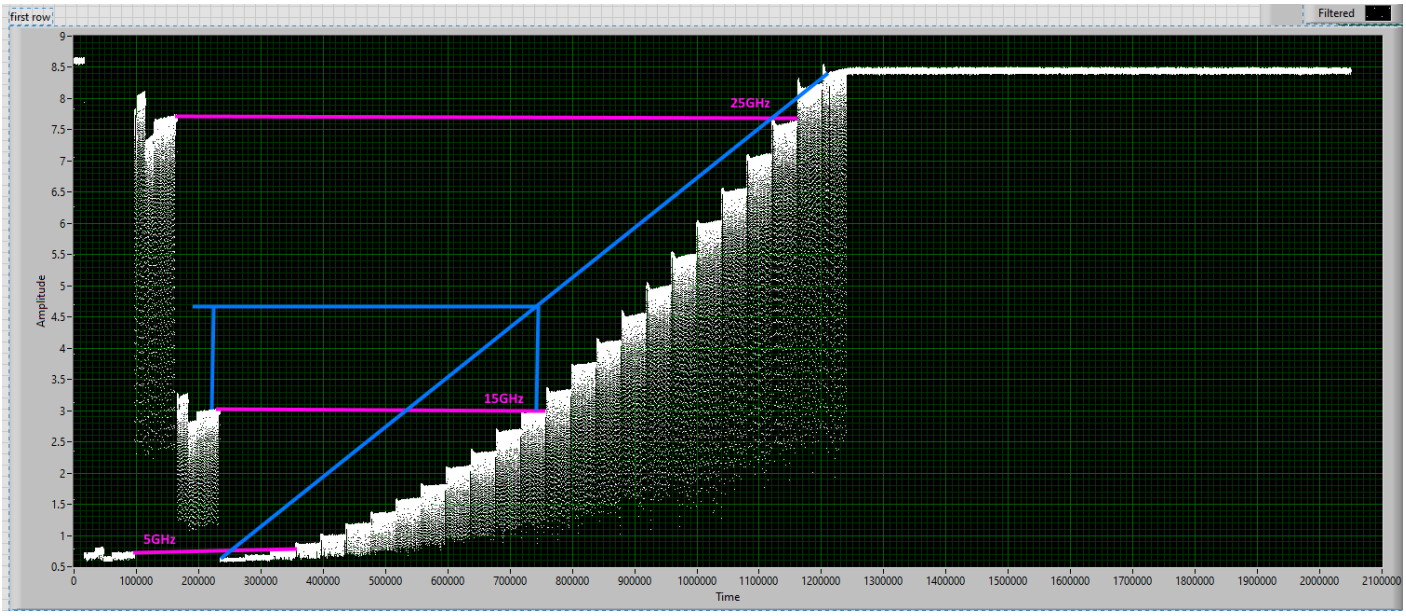
Attached are some captures of the debug output (for the NFA) during boot and (failed) alignment and a successful alignment.

So, what I found (for the NFA), was that the instrument uses a polynomial to perform the look-up of frequency -> YTF tuning voltage.

If I recall correctly, the alignment process does something along the following lines during "YTF Align":

1. It uses the current polynomial constants to set the YTF at the bottom, top and middle of the band, seeping the LO to detect the noise-peak in the receivers pass band.
2. It then steps through many frequency points, performing peaking of the noise (by tuning the YTF) at each frequency.
3. Upon successful completion of the calibration sweep, the polynomial is re-calculated and programmed to the EEPROM (when you click Save, I think).

Here's the YTF voltage monitored over time during the alignment process.



There is actually a significant loop-hole in the YTF alignment process. If the instrument has a fault and the YTF alignment process is run and saved, then the (very) incorrect YTF polynomial values are stored to memory. If the unrelated instrument fault is fixed, the instrument will still never be able to align itself back to a corrected state!

NOTE: Calibration data actually spans the two EEPROMs on instruments with a wide frequency range (e.g. those that go up to 26.5GHz, may the 6 & 13GHz models too). The YTF polynomial only actually has a few coefficients. The majority of the data on the EEPROM is the amplitude correction & mixer bias settings across the band. For the NFA, this data isn't important, as it corrects itself during it measurement process, but the the ESAs this may be more important for accurate measurements. If you just have incorrect YTF coefficients, then you will not need to touch the other EEPROM contents. If this is the case, I would strongly encourage you to back-up the contents of these EEPROMs before making any changes!!

BTW, each section in the EEPROM is checksum'd (I never did determine the CRC calculation used), but I seem to recall the instrument doesn't check this anyway!

The following data is stored in the first (of two) EEPROMs:

Model, serial, description and revision

```
3367 4534 3430 3436 3030 3031 0000 3233
3131 3437 3030 3031 3600 3939 0000 0041
4672 6571 7565 6E63 7920 4578 7465 6E73
696F 6E00 0000 0000 0000 0000 0000 9D3D
```

ALC reference level = 8 bits (written with address = 0xD, register = 0xC)

```
82CA 003F 0087 006F 006F 0093 0080 008F
```

YTF DAC Polynomial (double polynominal in 64bits, X^0, X^1, X^2, X^3)

```
4090 6B3B A000 0000 3E87 BF3E A000 0000
3C8E CD38 C000 0000 BA1C A744 A000 0000
0000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000 0000 0000 0000 0000 0000
4088
```

Unknown. Likely Mixer bias level is in here

```
26FB E76C 8B44 410D EE80 0000 0000 3F19
06E2 1C6A 43EC 3EE0 D12A F7C7 C49D C083
2922 D0E5 6042 BCDA C05A 8E88 2CA3 3E20
42C0 4307 107A BC55 8119 4633 00EF 3FE7
CED9 1687 2B02 BAD0 18E6 0000 0276
BF20 25E7 F115 8171 3EFA 6C92 D051 BC8B
0000 0000 0000 0000 0000 0000 0000 0000
BE60 5A63 F94C A62C 0000 0000 0000 0000
0000 0000 0000 0000 0000 0000 0000 0000
3DC4 2D10 D9C0 A872 0000 0000 0000 0000
4034 1B08 AAC9 6CC6 0000 0000 0000 0000
0000 0000 0000 0000 0000 0000 0000 0000
000F 0055 0000 0000 0001
000F 0055 0000 0000 0001
0D48 0000 0244
0D48 0000 02A8
0D48 0000 02DA
```

Of interest is the "YTF DAC Polynomial". This data is stored as a double precision floating point number (8 x 8 bytes = 64 bits).

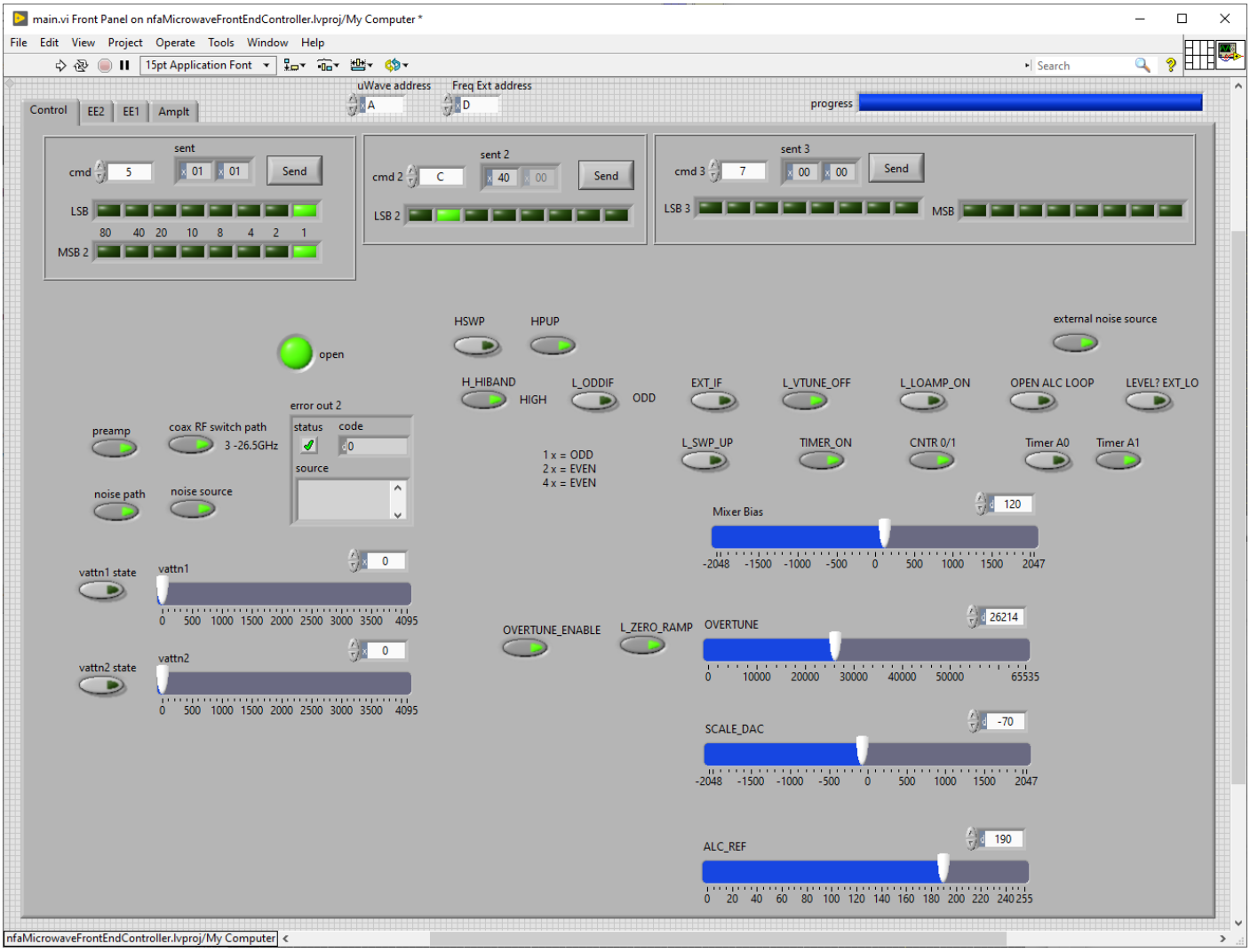
If I recall correctly, the short answer to what I did was to connect the YTF to my Network Analyser to perform an S21 measurement. I connected an external power supply to the YTF control input and swept the control voltage whilst noting the centre frequency of the pass band. A polynomial was fitted to the curve and these coefficients were programmed in to the EEPROM as a more sensible starting point for the YTF Align. I think I only needed the first one or two terms as its very linear. I then re-ran the YTF alignment process several times. Each time, it would get closer to the final characteristic and finally the alignment completed successfully at every frequency.

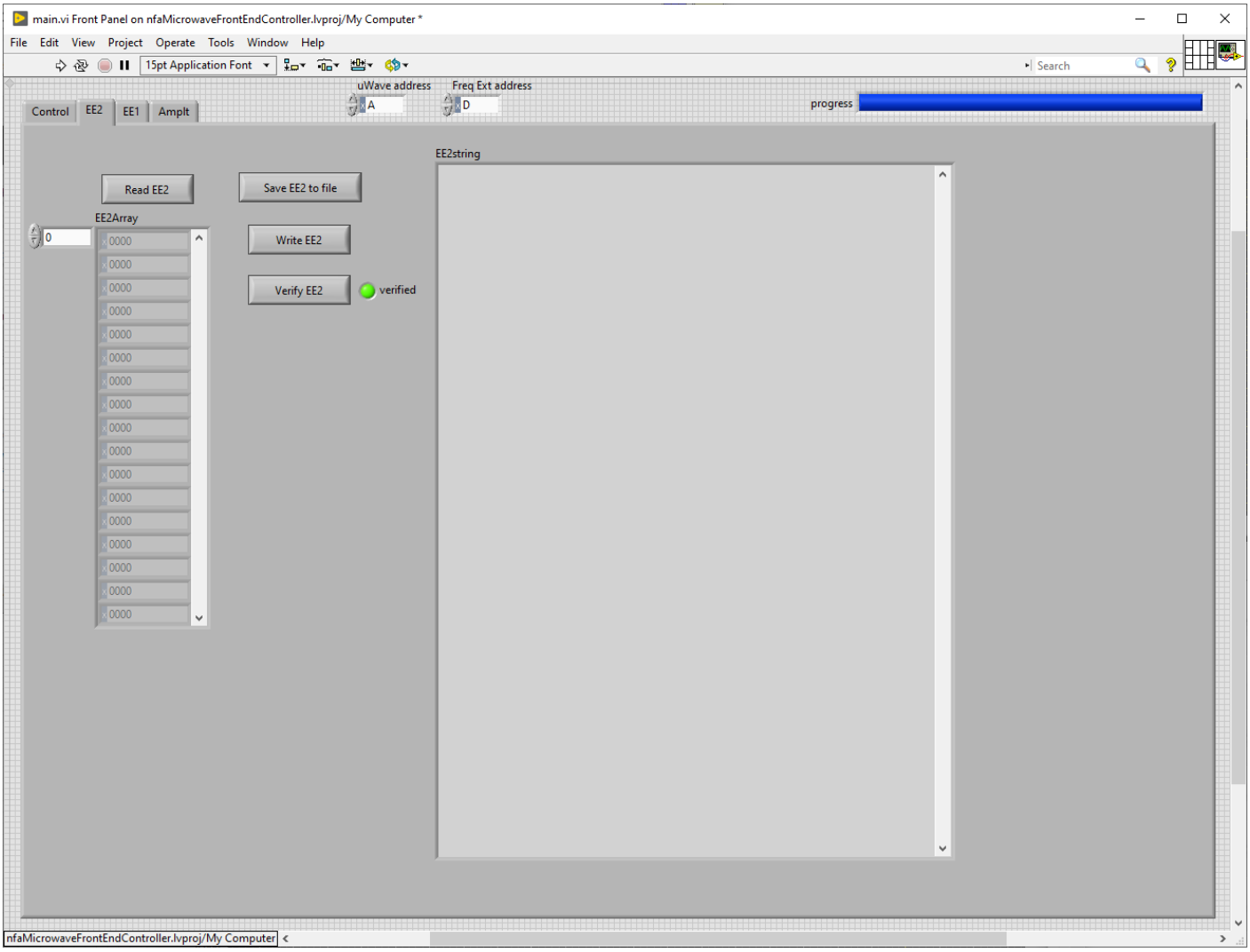
You can run the calibration (even if it fails), note the polynomial generated at the end. Read back the EEPROM and verify what was programmed matches what was computed and shown in the debug log. You can then adjust these coefficient values, as shown above, to bring the alignment back to a correct state.

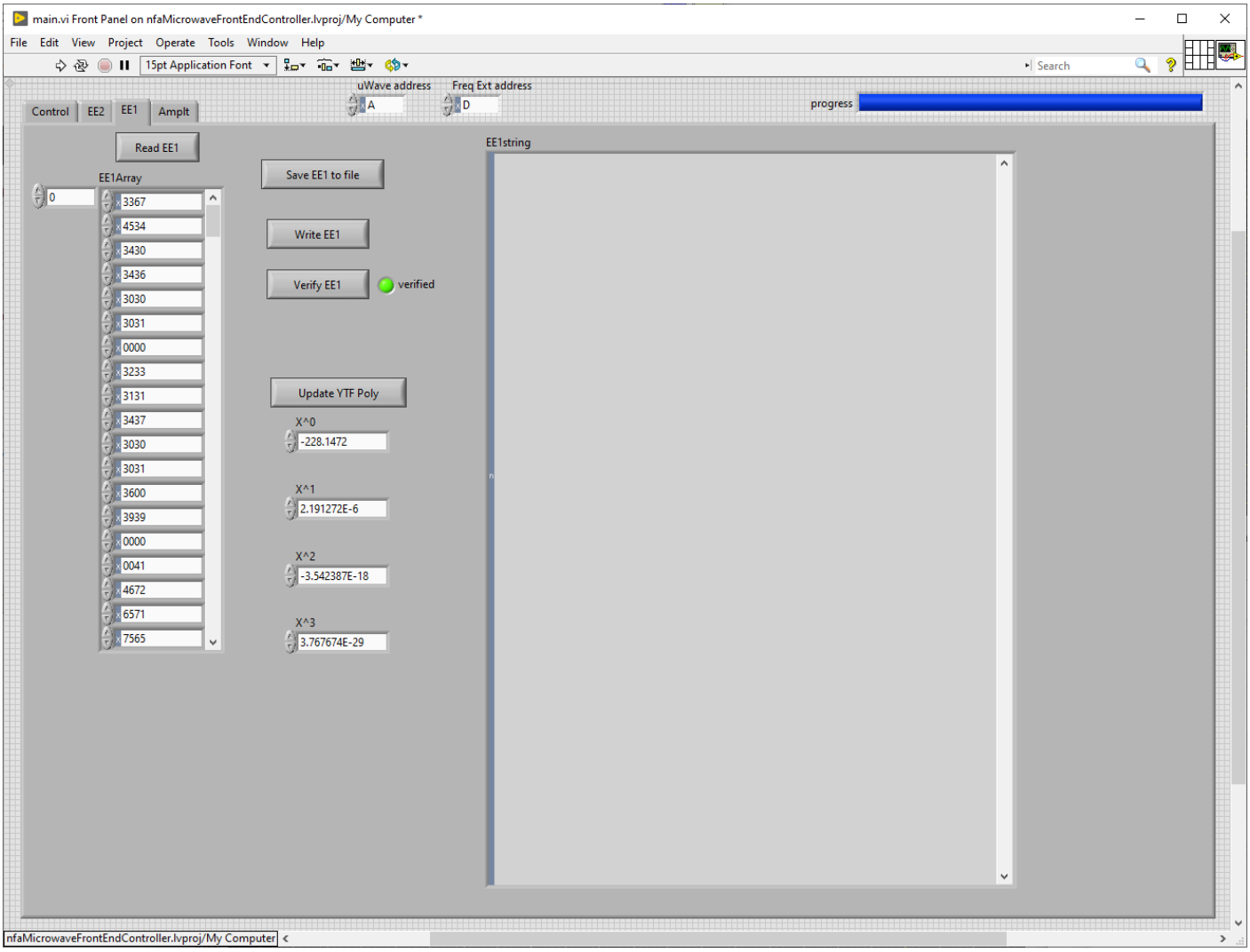
As an aside, I reverse engineered the pseudo-SPI bus that the CPU card uses to talk to all the other cards across the backplane. I then wrote a LabVIEW application with buttons and sliders to control some elements of the operation of the instrument. To do this, I pulled out the CPU card, and made connections to the pseudo-SPI bus on one of the plug-in cards.

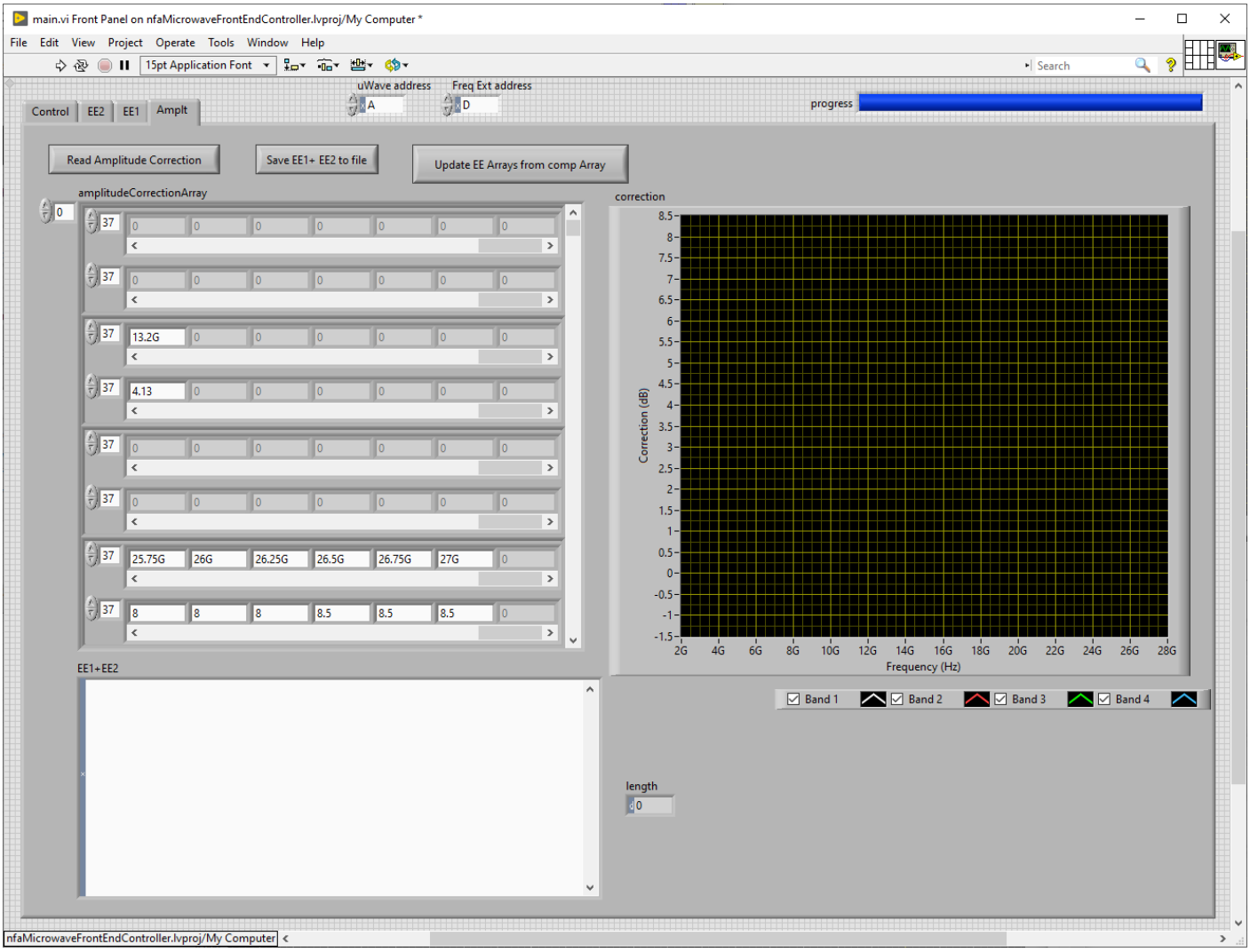
FTDI	ADBUS	Mask	Direction	Value	A7A4 pins
Black	-				90
Grey	4	0x10	1		89
Orange	0	0x01	1		88
White	6	0x40	1		87
Yellow	1	0x02	1		86
Purple	5	0x20	1		85
Green	2	0x04	0		84
Brown	3	0x08	0		34
Blue	7	0x80	0		81

I think the cable I used was one of these: https://ftdichip.com/products/c232hm-ddhsl-0-2/


main.vi Front Panel on nfaMicrowaveFrontEndController.lvproj/My Computer *
File Edit View Project Operate Tools Window Help
15pt Application Font
Search
uWave address
A
Freq Ext address
D
progress
Control
EE2
EE1
Ampit
sent
cmd
5
01
01
Send
LSB
80 40 20 10 8 4 2 1
MSB 2
sent 2
cmd 2
C
40
00
Send
LSB 2
sent 3
cmd 3
7
00
00
Send
LSB 3
MSB
HSWP
HPUP
external noise source
open
H_HIBAND
HIGH
L_ODDIF
ODD
EXT_IF
L_VTUNE_OFF
L_LOAMP_ON
OPEN ALC LOOP
LEVEL? EXT_LO
error out 2
status
code
0
source
preamp
coax RF switch path
3 -26.5GHz
noise path
noise source
1 x = ODD
2 x = EVEN
4 x = EVEN
L_SWP_UP
TIMER_ON
CNTR 0/1
Timer A0
Timer A1
Mixer Bias
120
-2048 -1500 -1000 -500 0 500 1000 1500 2047
vattn1 state
vattn1
0
0 500 1000 1500 2000 2500 3000 3500 4095
vattn2 state
vattn2
0
0 500 1000 1500 2000 2500 3000 3500 4095
OVERTUNE_ENABLE
L_ZERO_RAMP
OVERTUNE
26214
0 10000 20000 30000 40000 50000 65535
SCALE_DAC
-70
-2048 -1500 -1000 -500 0 500 1000 1500 2047
ALC_REF
190
0 20 40 60 80 100 120 140 160 180 200 220 240 255
nfaMicrowaveFrontEndController.lvproj/My Computer


main.vi Front Panel on nfaMicrowaveFrontEndController.lvproj/My Computer *
File Edit View Project Operate Tools Window Help
15pt Application Font
Search
uWave address
A
Freq Ext address
D
progress
Control EE2 EE1 AmpIt
EE2string
Read EE2
Save EE2 to file
EE2Array
0
Write EE2
Verify EE2
verified
nfaMicrowaveFrontEndController.lvproj/My Computer


main.vi Front Panel on nfaMicrowaveFrontEndController.lvproj/My Computer *
File Edit View Project Operate Tools Window Help
15pt Application Font
Search
uWave address
A
Freq Ext address
D
progress
Control EE2 EE1 AmpIt
Read EE1
EE1Array
0
3367
4534
3430
3436
3030
3031
0000
3233
3131
3437
3030
3031
3600
3939
0000
0041
4672
6571
7565
Save EE1 to file
Write EE1
Verify EE1
verified
Update YTF Poly
X^0
-228.1472
X^1
2.191272E-6
X^2
-3.542387E-18
X^3
3.767674E-29
EE1string
nfaMicrowaveFrontEndController.lvproj/My Computer



Amazing what you can do when you put your mind to it 😉

Best regards,
A

From: HP-Agilent-Keysight-equipment@groups.io <HP-Agilent-Keysight-equipment@groups.io> on behalf of zs437442 via groups.io <zs437442=gmail.com@groups.io>
Sent: 01 June 2021 22:43
To: HP-Agilent-Keysight-equipment@groups.io <HP-Agilent-Keysight-equipment@groups.io>
Subject: Re: [HP-Agilent-Keysight-equipment] E4407B restoration project: EEPROMs

Hi Alex,

Were you able to regenerate the YTF alignment/correction data? I am in the need to do the same for my ESA where the YTF needs to to re-adjusted. Did you find out the format of the values stored for that on the freq. extension board, and the best way to regenerate the correct values?

Thanks,

--zs

Groups.io Links:

You receive all messages sent to this group.

View/Reply Online (#119574) | Reply To Group | Reply To Sender | Mute This Topic | New Topic
Your Subscription | Contact Group Owner | Unsubscribe [eckertkp@gmail.com]

_.—,—.—,—

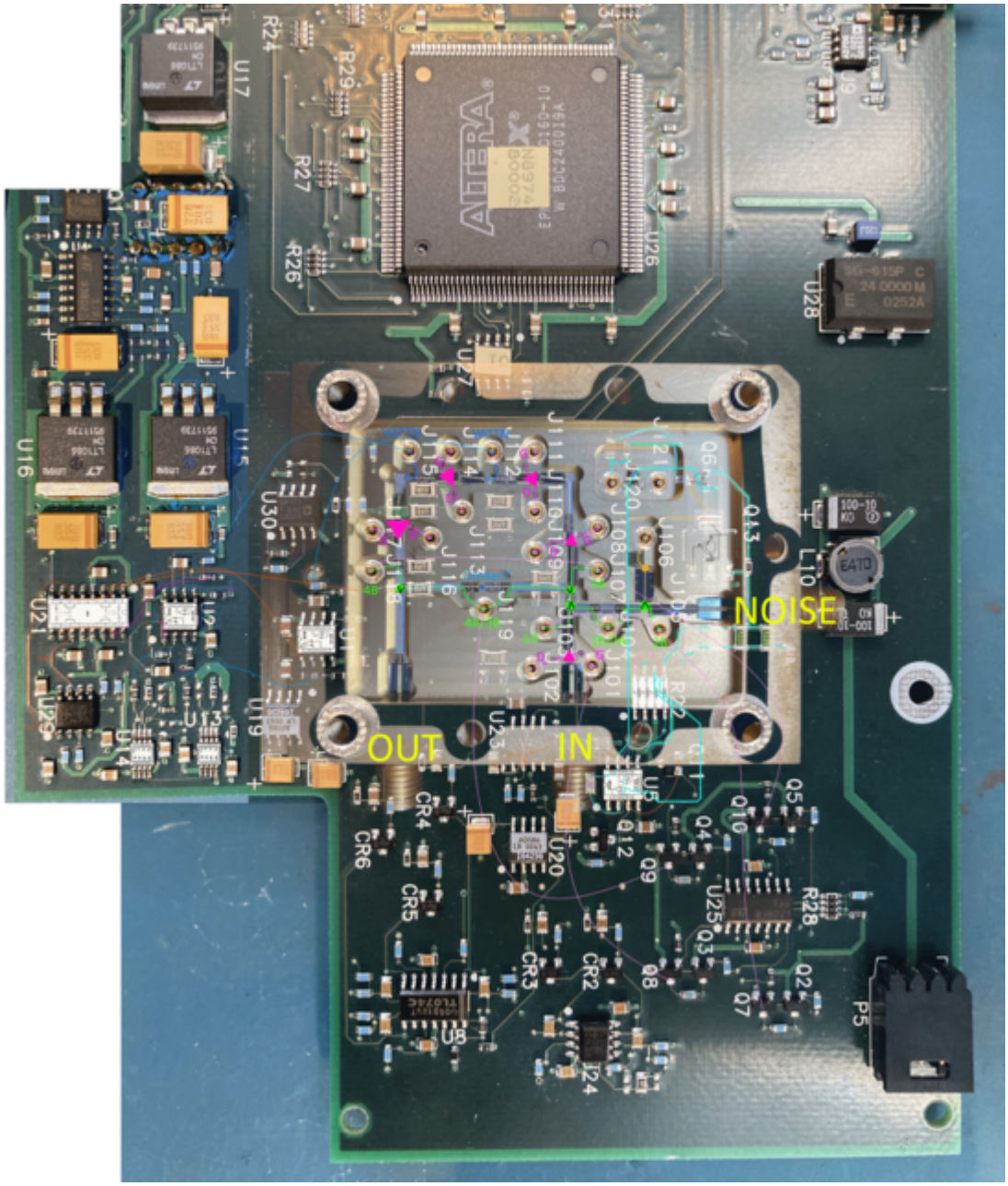
2 attachments

boot and ytf align log.txt
131K

_completely Successful calibration.txt
5K

Techfreakz <alex@techfreakz.net> Sat, Oct 23, 2021 at 8:04 AM
Reply-To: HP-Agilent-Keysight-equipment@groups.io
To: HP-Agilent-Keysight-equipment@groups.io

P.S. If anyone needs to work on the NFA "uWave Front End" module, here's the "chip & wire" and DC PCB reverse engineered!



_.—’—.—’—

Groups.io Links:

You receive all messages sent to this group.

View/Reply Online (#119575) | Reply To Group | Reply To Sender | Mute This Topic | New Topic

[Quoted text hidden]

Techfreakz <alex@techfreakz.net> Sat, Oct 23, 2021 at 8:06 AM
Reply-To: HP-Agilent-Keysight-equipment@groups.io
To: HP-Agilent-Keysight-equipment@groups.io

P.S. If anyone needs to work on the NFA "uWave Front End" module, here's the "chip & wire" and DC PCB reverse engineered!

_.—’—.—’—

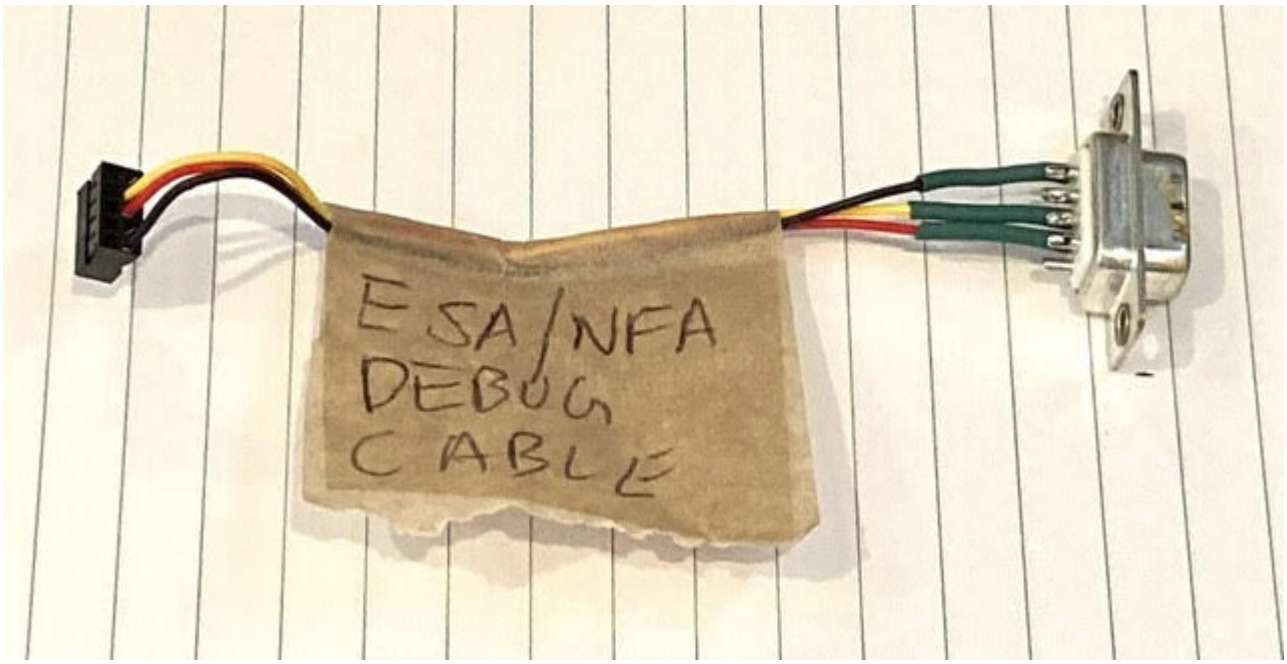
Groups.io Links:

[Quoted text hidden]

uWave module overlay.png
5815K

Techfreakz <alex@techfreakz.net> Sat, Oct 23, 2021 at 8:14 AM
Reply-To: HP-Agilent-Keysight-equipment@groups.io
To: HP-Agilent-Keysight-equipment@groups.io

P.P.S. Here's a photo of the serial debug cable.



_.-._.-._.-._

Groups.io Links:

[Quoted text hidden]