I received a Flex3000 app. 3 weeks ago from a friend of mine. The Flex was totally dead. He couldn't tell me what really happened and already gave it up. His intention was to keep it as „spare parts".

Situation:

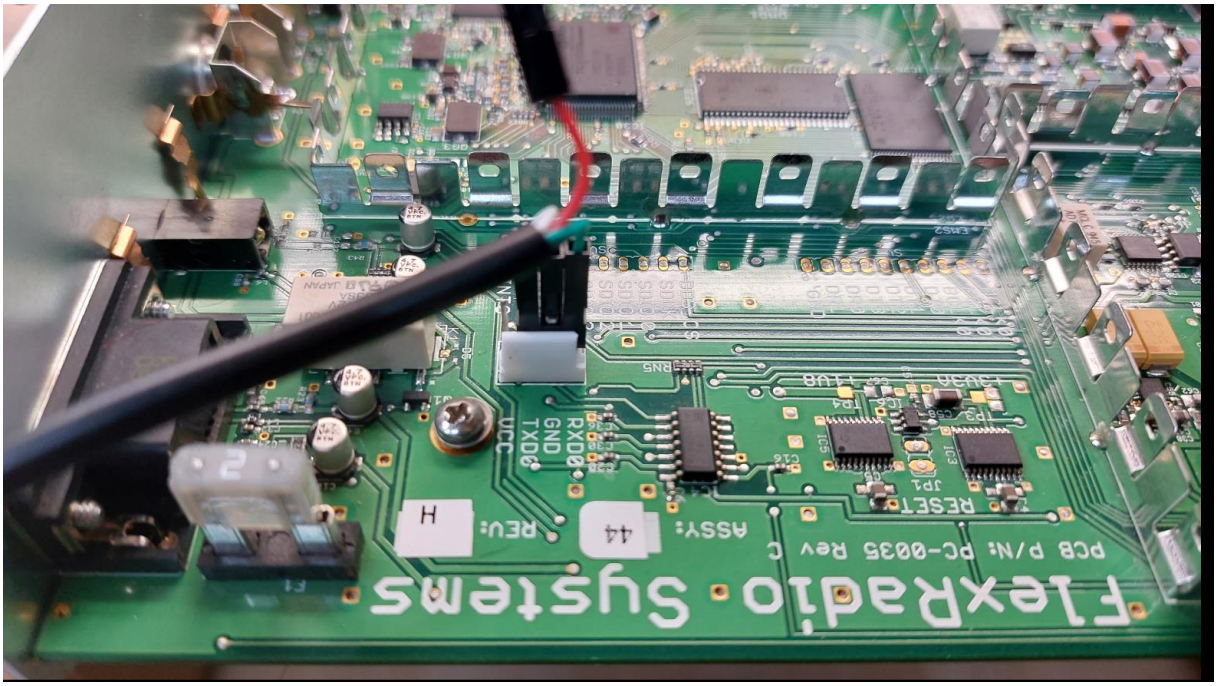
- After powering up the blue LED turned on but no further relay-click as usually noticeable.
- Connecting it to a PC – it was not noticed by the firewire-driver
- Looking at the firewire socket - it seems that somebody forced the plug upside down into the socket.

After opening the Flex I found the following situation:

One of the two EMI-Inductors of the firewire bus was loose – after de-soldering, it showed up as faulty (one path open circuit): I placed two jumpers there – will fix that later.

Still no firewire response…

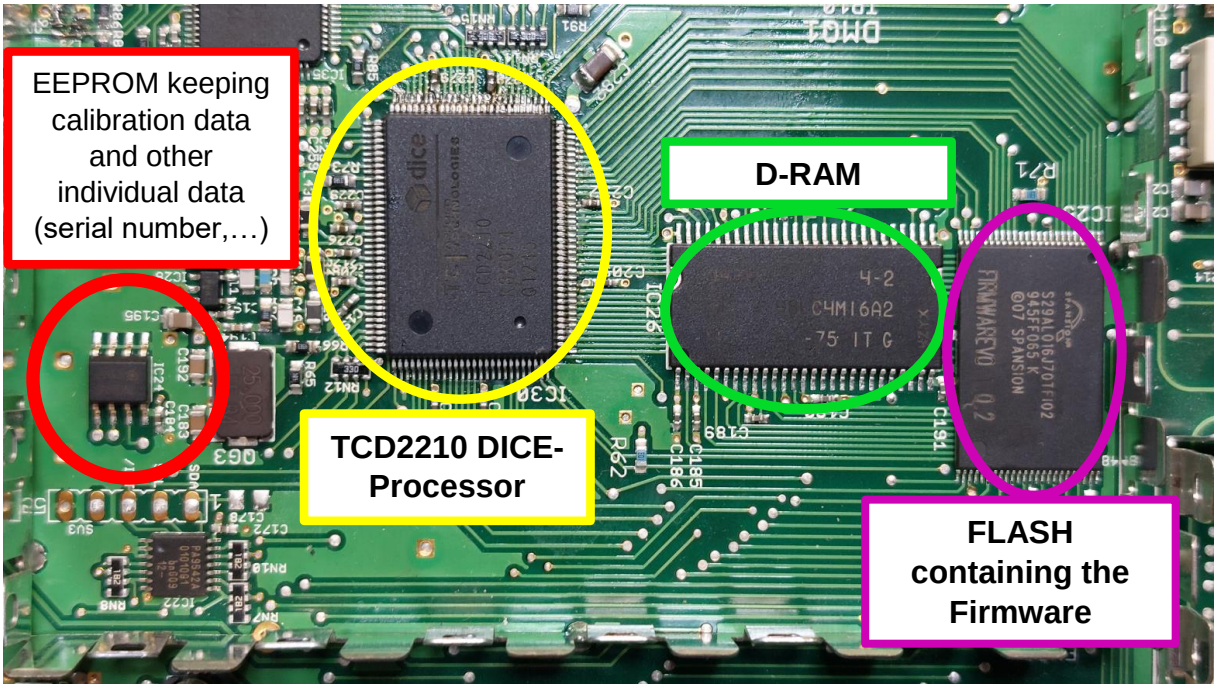
I connected a RS232 adaptor to the internal serial port as it can be seen here:



Settings at this “serial-debug-port” are: 115.2 KBaud, 8, N, 1 – no flow control (that is what worked for me)

Inside the Flex there is a TCD2210 ARM-based firewire “processor/controller”. Documentation is very rare available and all what I notice here are more or less assumptions.

Here an overwiev of the different components inside he flex:



What I found is, that that there is a RedBoot bootstrap loader or “OS” working in the TCD2210. I could find some documentation about the commands on the internet.

The following dump shows the output during the start up sequence of the Flex.

For the broken flex it stops after the “fis – load – dice” command. So it loads the Flex-firmware from flash to the internal RAM but doesn’t start it (?). The final “go” command is missing in the script.

Dump from the Terminal:

```
Watchdog Int installed!!!

RedBoot(tm) bootstrap and dbgenvironment [RORM]
Non-certified release, version v2_0 - built 12:23:07, Sep 29 2006

Platform: TCAT DICE/VB (ARM7TDMI)
Copyright (C) 2000, 2001, 2002, Red Hat, Inc.

RAM: 0x00000000-0x0080000 x00016528-0x007e00available
== Executigbot script in 20.000scnds - enter ^C toaot
RedBoot> fis load dice
RedBoot> go   <------------this command was missing!!!
Could not initalize environment - will use deault values
                                    **********************************
* myMode board configuration         *
* Board is configue to mode: 0      *
* Name: DICE JR EM2/28              *
* Use: dierver.dump              *
**************************************
routeMid
Initializing Hardware.TRX.......PA.....

******************************************************
* FLEX5000 Command LineInterfaceSstem              *
* Copyright 205-2012 by FlexRadio Systems, Inc.        *
******************************************************
* Model: FLEX-3000  Serial Number: 0610-30  r2.1.4.12   *
******************************************************
>routeLow
```

I fixed that by the following command sequence, which writes a new permanent startup script:

In Green my input:

```
RedBoot> fconfig
Run script at boot: false t
      Boot script:
Enter script, terminate with empty line
>> fis load dice
>> go
    Boot script timeout: 0 10
Use BOOTP for network configuration: false .
Update RedBoot non-volatile configuration - continue (y/n)? y
```

After doing this and restarting the flex …. There was the relay click :-)

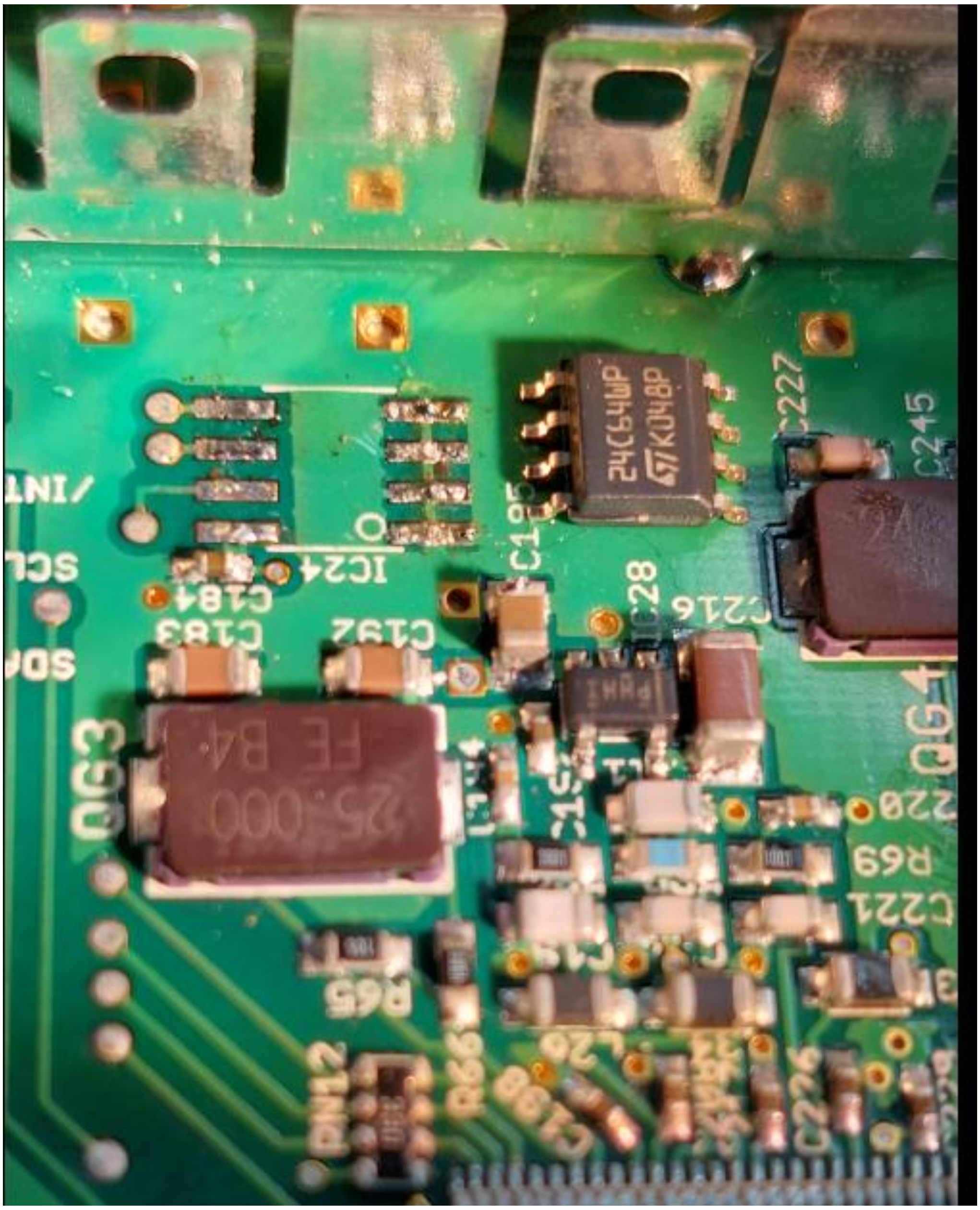
And… the flex showed up at the firewire port!!!

After starting PowerSDR – RX was working!!

But no TX - and, the Serial number was totally corrupted (255255-65535).

What is this?

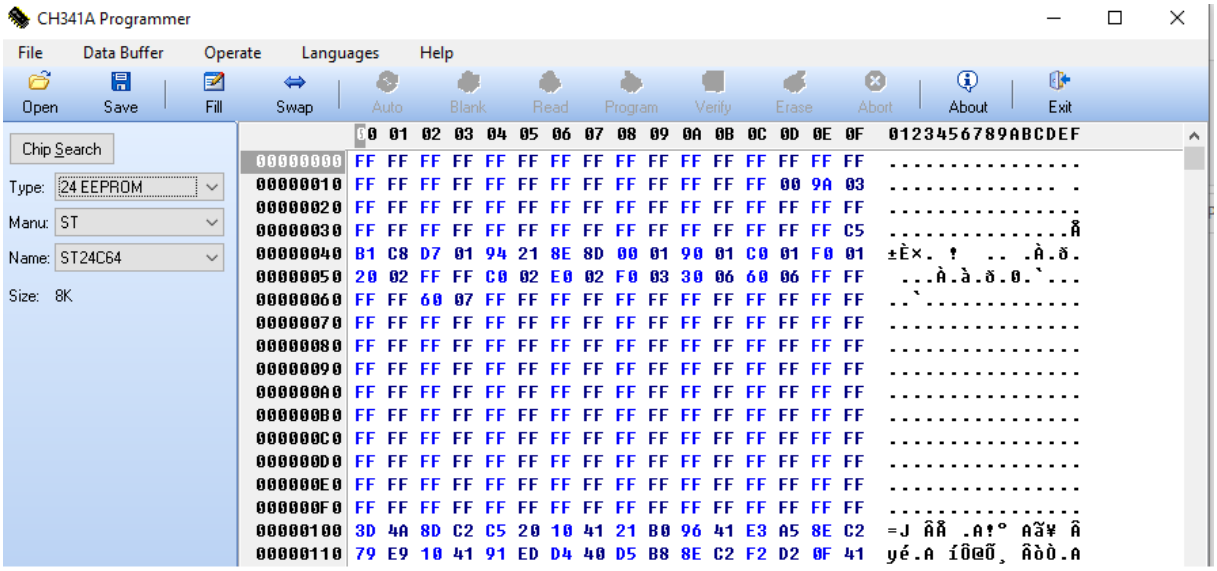
After some investigations I found, that the serial number is coded in the EEprom (26c64) you can see at the following foto:



EEPROM desoldered and new one waiting to be placed

After closing PowerSDR I could find an EEProm dump in the backup folder.

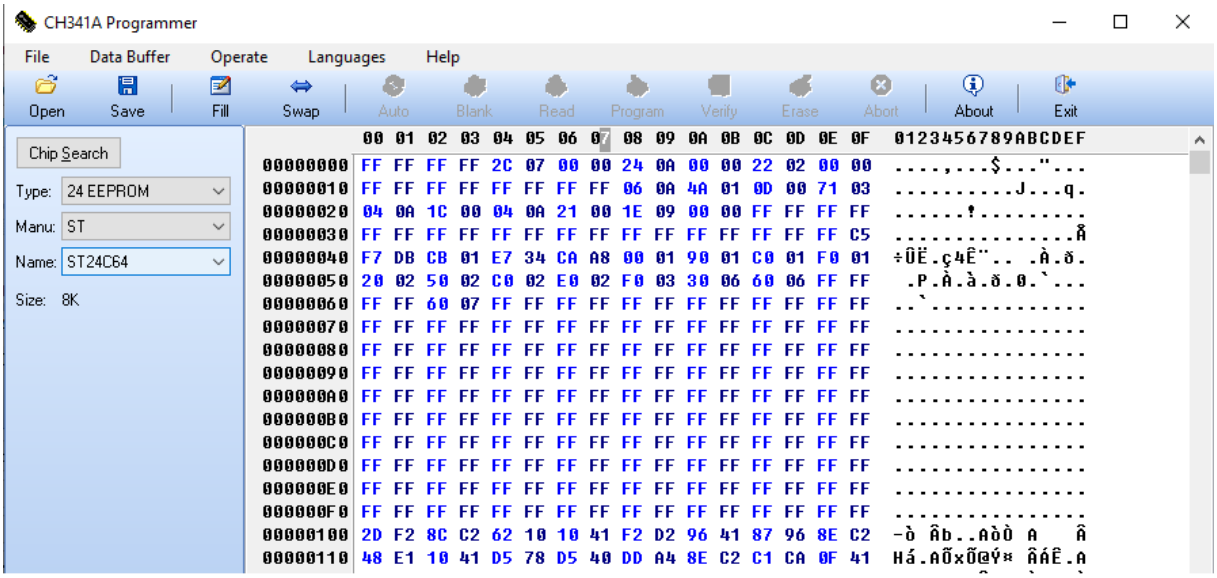
The content looked corrupted as you can see here (check the first 4 lines):



(The screendump above comes from reading the eeprom after desoldering)

Also the calibration data looked corrupted….

Comparing this to a “normal” EEprom backup which looks like this:



How to fix this??

I was lucky to get an EEprom-backup from the pre-pre owner of the flex, who still had this on his harddisk.

But how to get this back into the EEProm?

Trying to do this by the standard procedures fail, because all the tools failed connecting to the flex. This might be due to the corrupted eeprom and the wrong serial number.

So, for me that was a dead-lock!

The only way for me was, to program a new eeprom with the original content and to replace it.

Pretty straight forward, but more problems to come:

The eeprom dump in the backup folder was plain ascii(!) separated by commas. Additionally containing line and row numbering. A format, which could not be read by an eeprom programmer software directly.

I did the necessary conversion by excel.

First I read the csv-file into a spread sheet

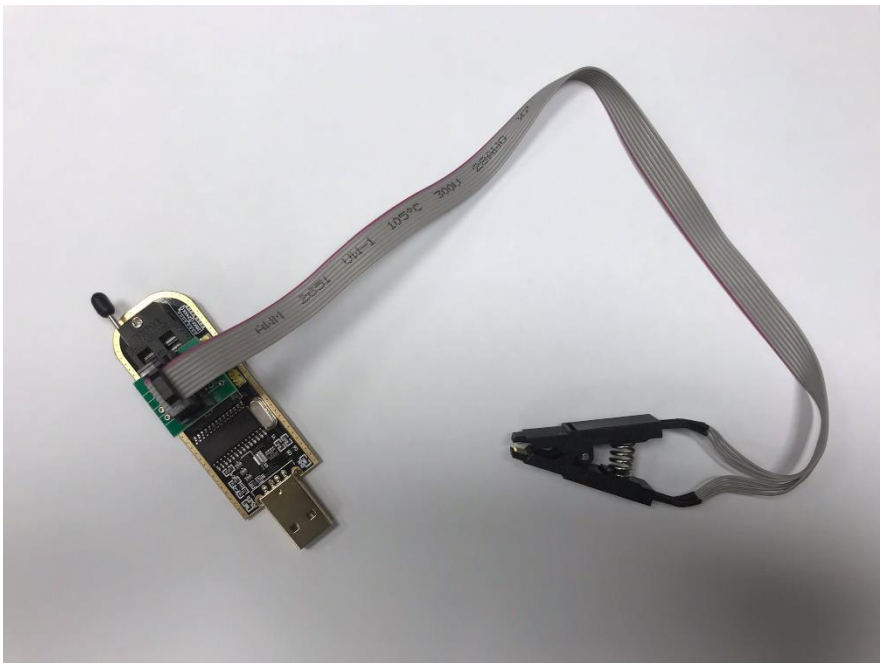
Than deleted the first row and first column (Containing the numbering)

After that used the HEXINDEZ conversion function transfering the data into a 2nd spreadsheet.

Finally I wrote a small macro to dump this content in to a binary file byte by byte.

The resulting binary could be read by a standard eeprom-programming-tool (CH341a-based)

I used the one shown on the following fotos (app. 15$)



After replacing the eeprom the next problem occurred. Starting up PowerSDR I got many error-messages regarding eeprom access – did I do something wrong??

I couldn’t find any fault – the eeprom content was exactly what it was before the crash…

Very strange…

But now after the eeprom replacement, I was able to use the firmware update tool.

I wrote the current firmware to the flex, restarted and wow, everything worked fine – RX and TX!