The insides of an automatic defibrillator

Paramedic CU-ER1

brian@security-bits.de ~~ www.security-bits.de ~~ @BadgeWizard

About Me

- Love coming to Brazil
  - Again and again and again
- Hacker
  - Hardware, Embedded, Telco/Cellular
  - Need to hack more :-(
- Officially OT Security in railroads
  - Last years talk at H2HC
- Love playing with new stuff
  - And a defibrillator has been on my list for a long time

Medical Stuff

Not a Doctor :) my understanding

- In contrast to what I used to think for a very long time, a defibrillator doesn't jump start the heart
  - It rather resets it
- During arrhythmia the hearts drops out of it's pretty clean beat and, depending on the actual issue is by far too fast or simply badly out of rhythm
  - Thus, not the correct amount of oxygen for the brain

Medical Stuff

Not a Doctor :) my understanding

- The defibrillator gives the heart a significant jolt
- Thus cramping and stopping it for a moment and giving it the chance to drop back into rhythm
- And as oxygen is pretty important, the quicker the heart gets back into rhythm the better

Stopping a heart

- Is dangerous and literally lethal
- Depending on the country you're in, it might only be legal for doctors to use the defibrillator, in others instructed medical personnel i.e. EMTs

- The combination of time critical and only be medical personnel can be an issue

AED
Automatic External Defibrillator

- Making defibrillation available to “everyone”
- Making defibrillation available by far quicker
- But how?
  - And how safe?
  - I know probably everybody watching this wants to shock something!

Why >External<?

- The image doesn’t show a pacemaker!
- It’s an implantable cardioverter-defibrillator
  - ICD or also AICD (trademarked)
- Instead of providing a stable rhythm it just gives the heart a shock, when necessary

The How

- The AED monitors the heartbeat
- When necessary, it enables the shock button
- It only shocks, when the button was pressed
- Shocking at will, is not possible

Today's Guest

CU-ER 1

Today's Guest

- Paramedic CU-ER 1
- Why? Honestly, eBay find
- Produced in 2004
  - My unit
- Apparently been in use in a doctor's practice
- It's German
- The stickers came from me

From the outside

- 5 Buttons
- Connector for the pads
- 3 LEDs
- Display
- IR
- SM-Card slot
- RS-232
- Power
- (Battery Slot)
- Speaker / Microphone

Working outside->in

- Defensive approach
- Do everything one can before opening the device
  - It’s a medical device, it might actually have tamper protection
- Make sure, even if the device gets bricked, there is at least some insight

Plan:

1. Buttons & Display
2. Serial
3. SmartMedia Card
4. IR

General Use

- After pressing the on switch it tells you to attach the pads
- Shows a picture where the pads go
- Then shows the EKG

A note on ?fear? ?smartness?

- It’s a medical device, well it was one, when I got it
- It’s made to stop a heart
- I honestly didn’t connect the Pads to myself
- Just felt wrong

Device Information

Usage

Settings

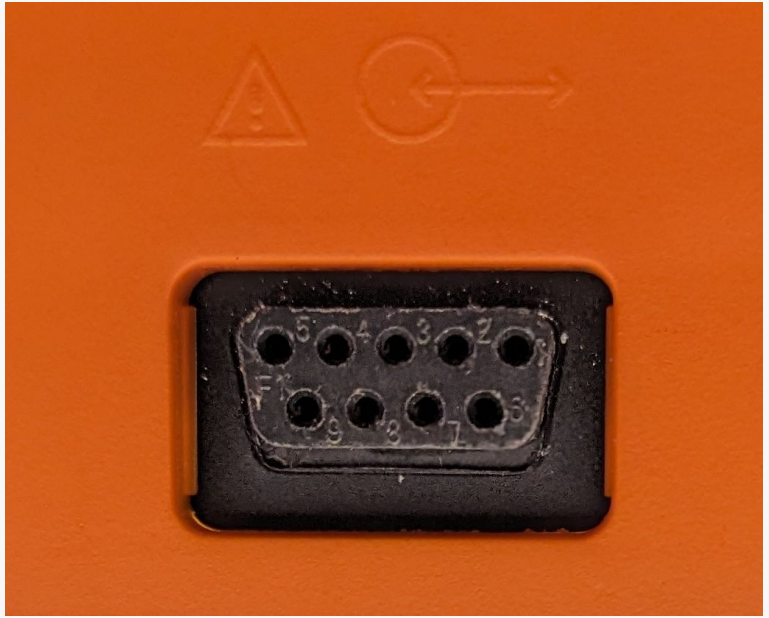
Communications

Buttons

- Switching the device on holding up and down will start the self test mode
- Not interesting
  - Unless: Error 0156
  - Missing Battery, can be resolved by a self test with battery
- Nothing else interesting to be found

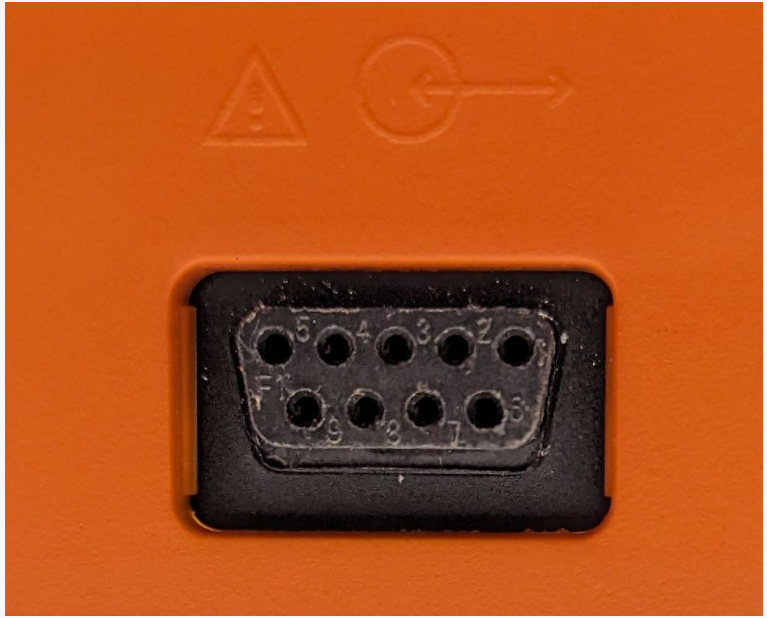
Serial Port

- The manual states a specific serial cable from the manufacturer is required
- Thus, the careful approach
  - Logic Analyzer



Serial Port

- The manual states a specific serial cable from the manufacturer is required
- Thus, the careful approach
  - Logic Analyzer
- Neither the print, nor transfer function result in any reaction on the port
- Might be some magic necessary,
- Postponed until later… :-(



SmartMedia Card

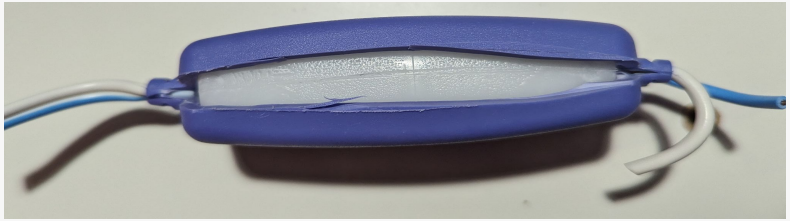
- Complains, the card isn’t correctly formatted
  - Played with various formats, no luck
- Matches the note in the instructions
- Sadly don’t have a valid card, so dead end

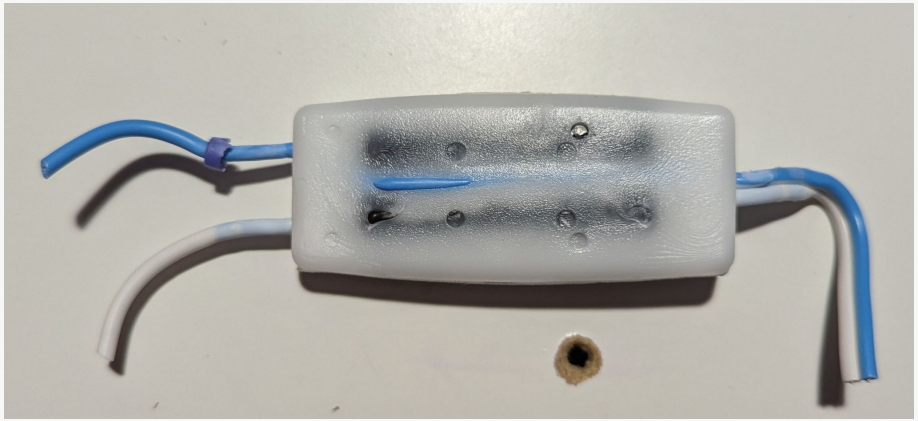
Connector for the pads

- 2 pin connector
- Not interesting?

Connector for the pads

- 2 pin connector
- Not interesting?
- Well, we have the kids adapter
  - With reduced energy
- So....





Child Connector

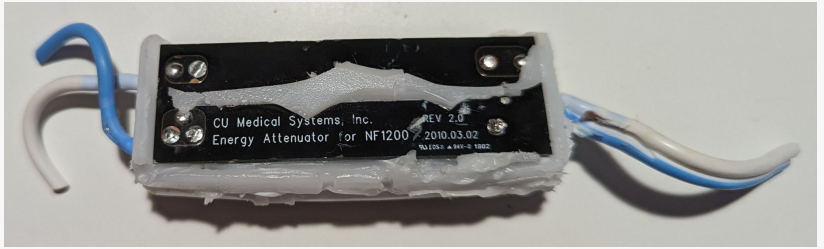
After adding

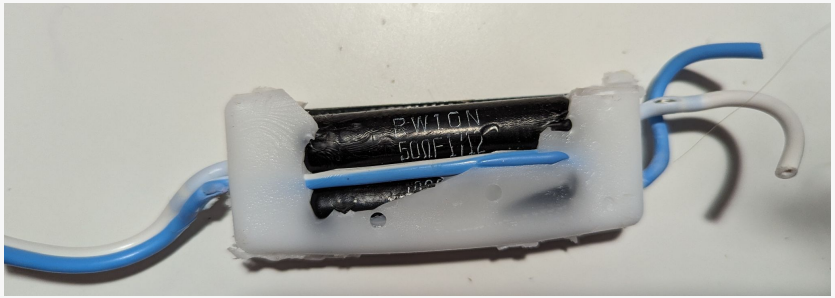
- Acetone didn’t do anything...so...
- Dremel / Multitool
- Axe head as a chisel and a hammer
- Knive
- Heat gun

Child Connector

Child Adapter

- “Energy Attenuator for NF1200”
- Designed in 2010
  - Probably the newest part in the set
- 2 pretty large resistors on the white line
- 50 Ohms





KOA SPEER ELECTRONICS, INC.

RW

coat-insulated miniature precision power wirewound resistors

applications and ratings

Part Designation	Power Rating U	Power Rating V	Resistance Range (Ω) D±0.5% (E24 • E96 25x10^n•50x10^n)	F±1% (E24 • E96 25x10^n•50x10^n)	H±3% (E24 & 25x10^n•50x10^n)	J±5% (E24 & 25x10^n•50x10^n)	T.C.R. (ppm/°C)	Max. Working Voltage	Max. Overload Voltage
RW1/2	0.5W	—	10 - 2.61k	10 - 2.61k	0.47 - 2.7k	0.47 - 2.7k	+20/-50: R≥10Ω +50/-70: 1Ω≤R<10Ω +400/-90: R<1Ω	80V	150V
RW1/2N			—	10 - 2.37k	10 - 2.4k	10 - 2.4k			
RW1	1.0W	—	1 - 5.11k	1 - 5.11k	0.1 - 5.1k	0.1 - 5.1k		130V	300V
RW1N			—	10 - 3.74k	10 - 3.6k	10 - 3.6k			
RW2	2.0W	3.0W	1 - 10k	1 - 10k	0.1 - 10k	0.1 - 10k		140V	500V
RW2N			—	15 - 10k	10 - 10k	10 - 10k			
RW3	3.0W	5.0W	1 - 15k	1 - 15k	0.1 - 15k	0.1 - 15k		200V	600V
RW3N			—	15 - 15k	15 - 15k	15 - 15k			
RW5	5.0W	7.0W	1 - 30.1k	1 - 30.1k	0.1 - 30k	0.1 - 30k		400V	700V
RW5N			—	20 - 29.4k	20 - 30k	20 - 30k			
RW7	7.0W	10W	1 - 45.3k	1 - 45.3k	0.1 - 47k	0.1 - 47k		600V	800V
RW7N			—	36 - 44.2k	36 - 43k	36 - 43k			
RW10	10W	14W	1 - 60.4k	1 - 60.4k	0.1 - 62k	0.1 - 62k		1000V	1500V
RW10N			—	62 - 49.9k	62 - 51k	62 - 51k			

Operating Temperature Range: Characteristic U: -55°C ~ +275°C, V: -55°C ~ +350°C

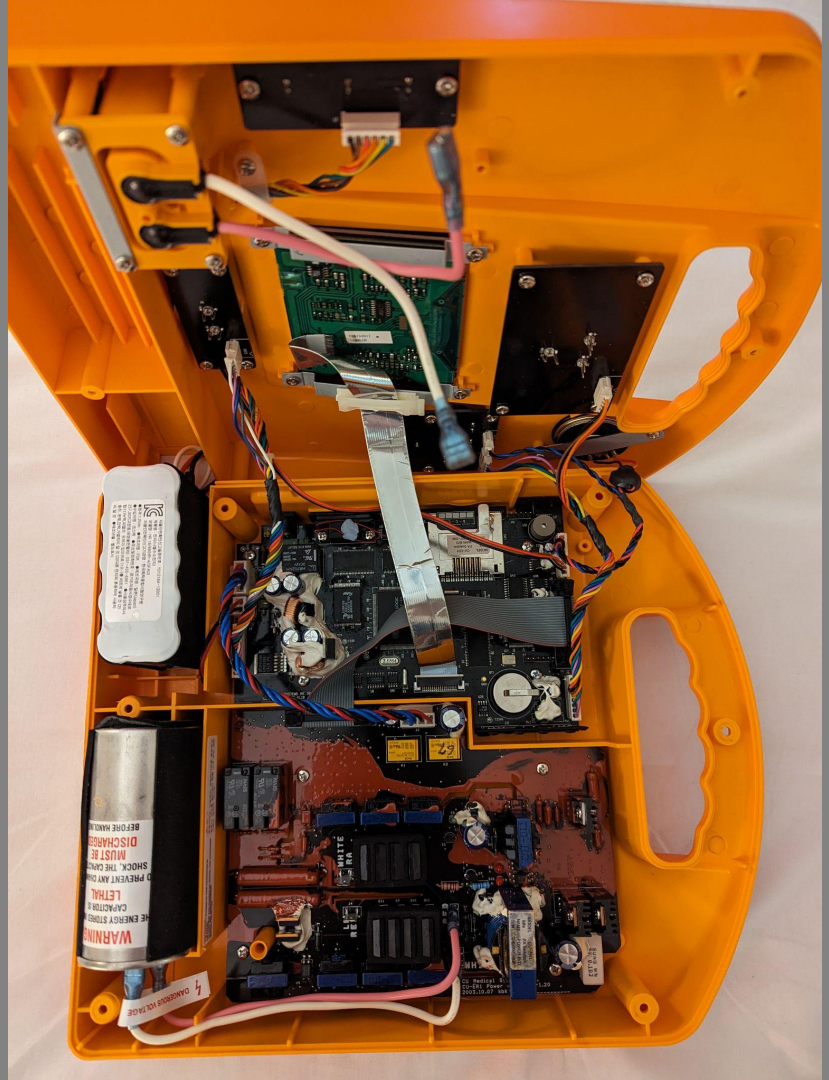
Child Connector

IR

- Didn’t have a serial IR Adapter available, so skipped it
- Should also be the same as the serial connector

The Inside

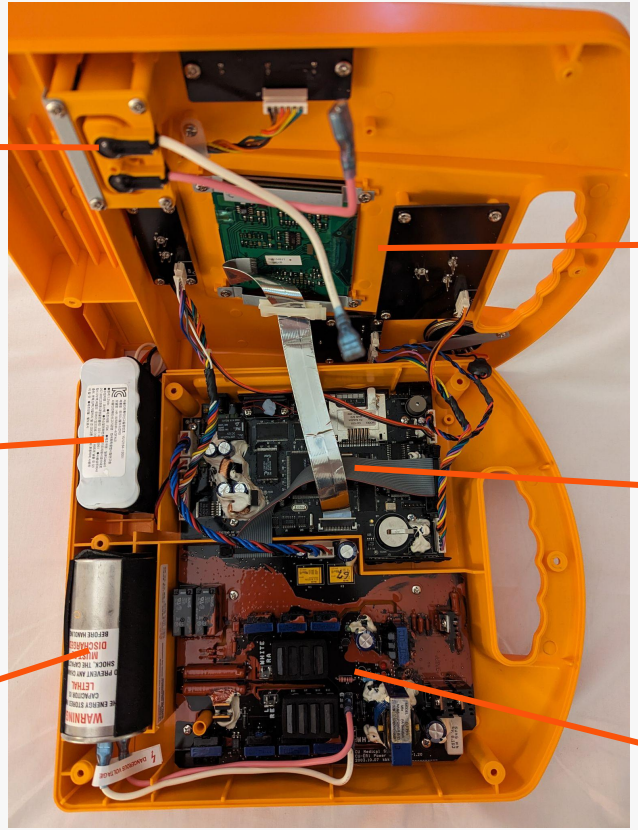
6 screws later...



Pads

Battery

Cap

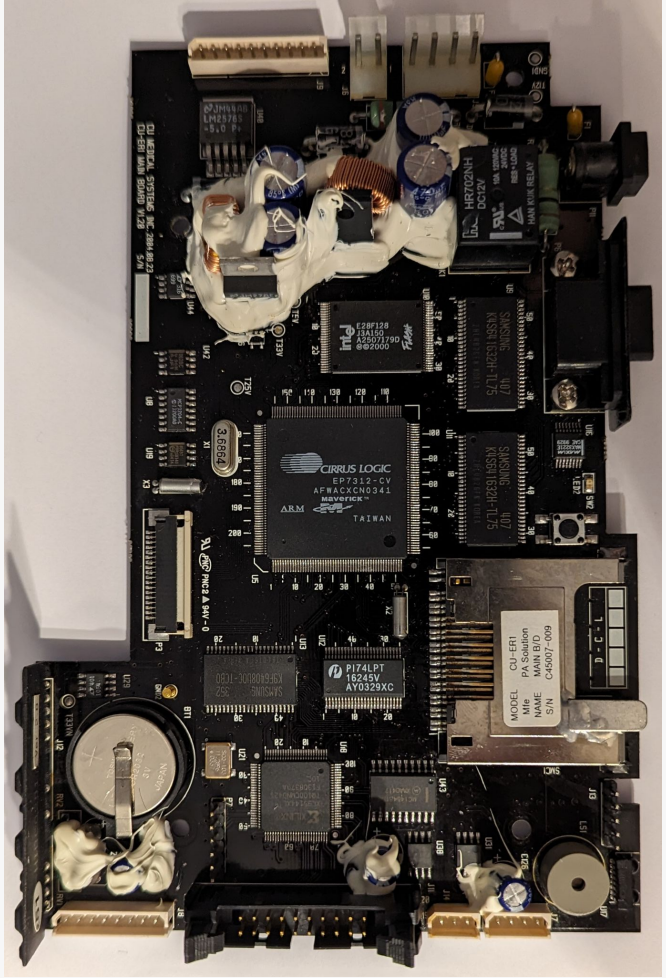


Display &
Buttons

Control
PCB

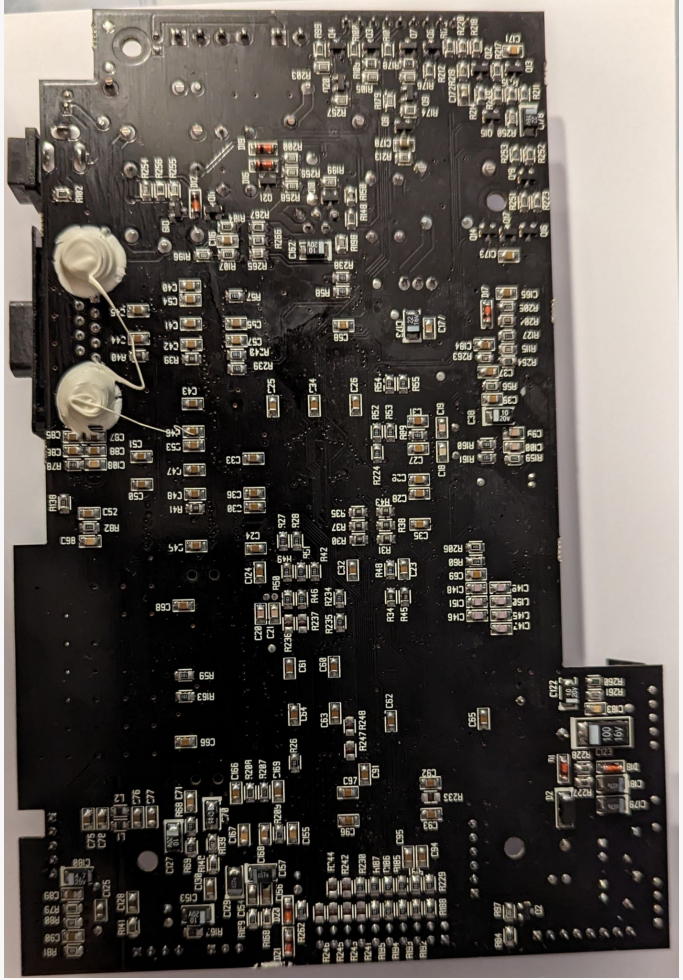
Power
PCB

Device



Top

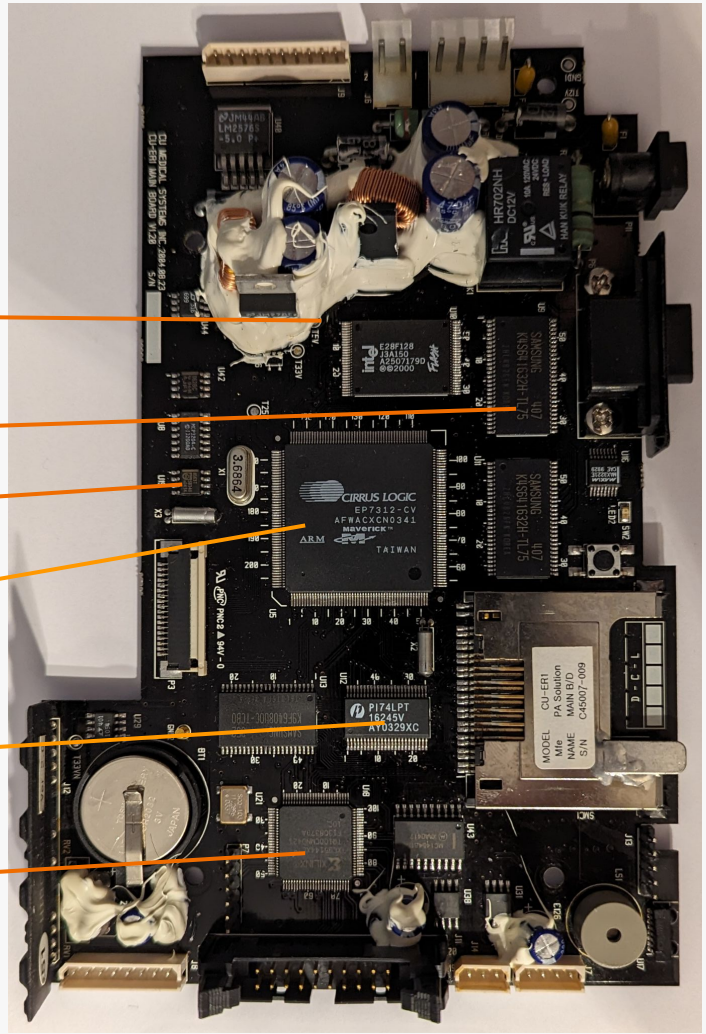
Bottom



Control PCB

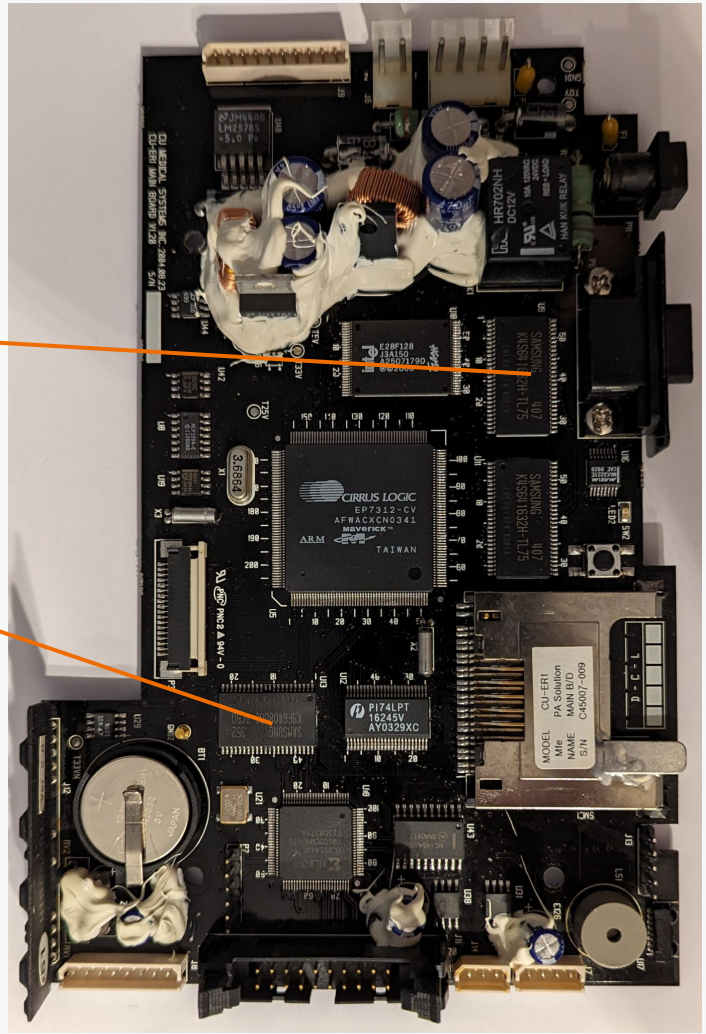
Control PCB Components

- Intel E28F128J3A-150
  - Flash
- Samsung K4S641632H-TL75
  - RAM
- PCF8563
  - RTC/Clock/Calendar
- Cirrus Logic EP7312
  - µC
- PI74LPT 16245V
  - Level Shifter
- Xilinx XC95144XL
  - CPLD



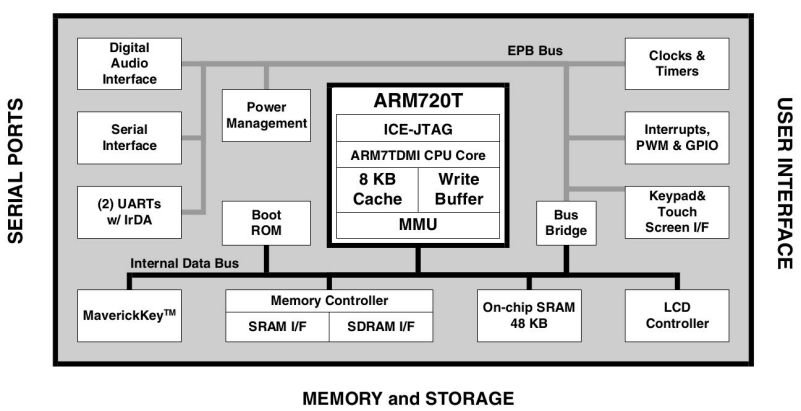
The one I missed

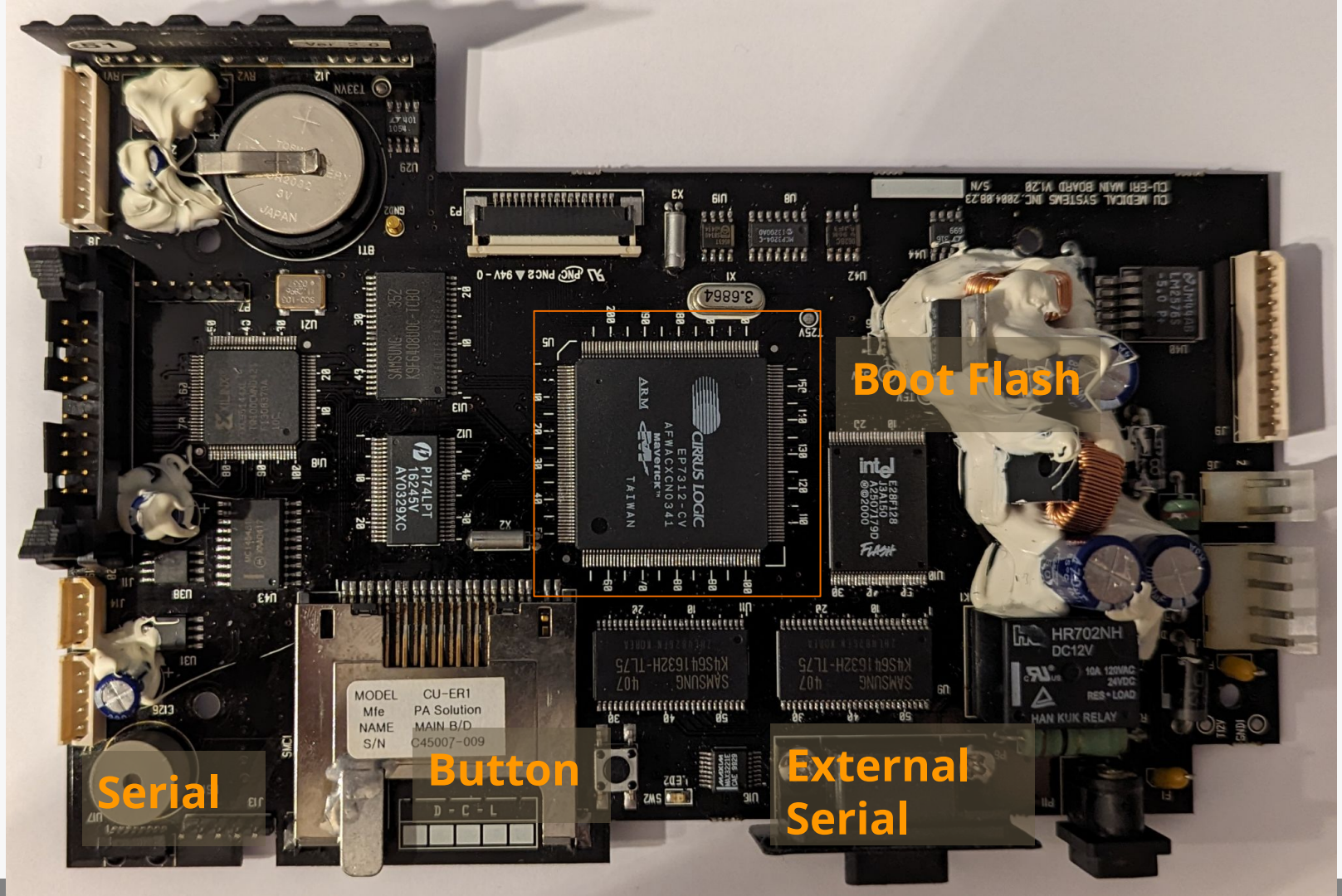
- Samsung K4S641632H-TL75
  - RAM
- Samsung K9F6408U0C
  - 8M NAND Flash
  - Used for Settings etc.



Cirrus Logic EP7312

- No exposed JTAG
  - Pins are otherwise used
- UART
  - UART 1 connected to external Serial via level shifter
  - UART 2 connected to J13
- Connected to Intel E28F128J3A-150
- Button on pin 155, nMEDCHG/nBROM
  - Boot from internal bootrom on press



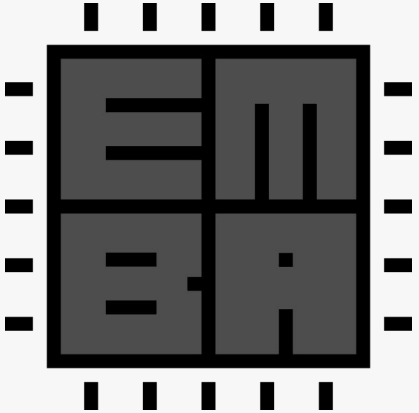

Boot Flash
Serial
Button
External
Serial

Memory Access

- Neither JTAG nor extraction via bootloader
- → Soldering Iron
  - Memory programmer & adapter
- Works!
  - A few strings from the device

```
CU Medical Systems, Inc.
Patient    :_____________________________
Age        :__________
Sex        :__________
Operator   :_____________________________
Device On     :
Elapsed Time  :
Total Shocks
*Device Information.
  Model : CU-ER1
  S/N   :
```

Data Extraction



- Automated approach
  - EMBA and FACT
- Results:

Data Extraction

- Automated approach
  - EMBA and FACT
- Results:

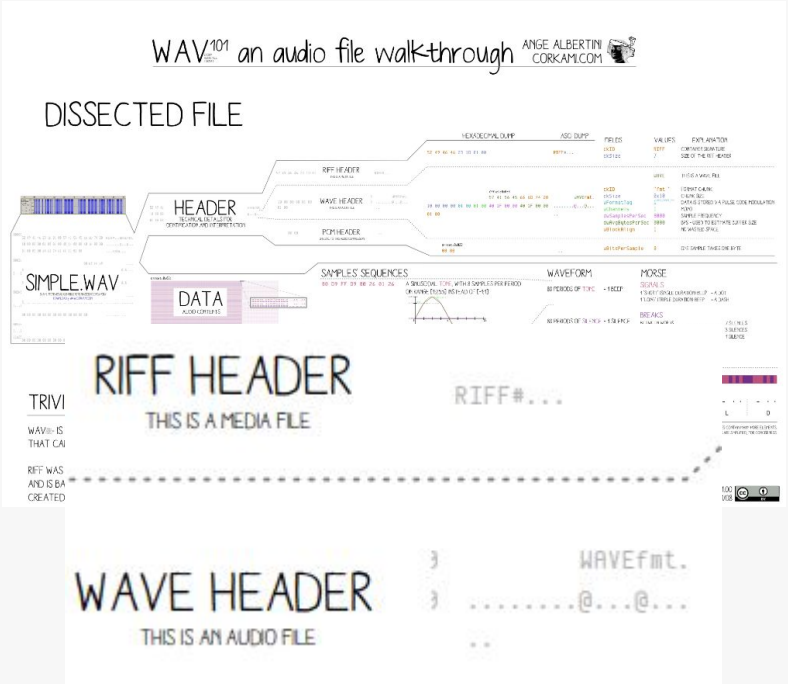
Audio

- While scrolling through the strings

Audio

- While scrolling through the strings
  - Ange Albertinis posters are a good reminder
- There are multiple wav headers in the image
- Thus: foremost to the rescue

RIFF
WAVEfmt
dataA
fact
LISTB
INFOISFT5
GoldWave (C) Chris S. Craig,
http://www.goldwave.com

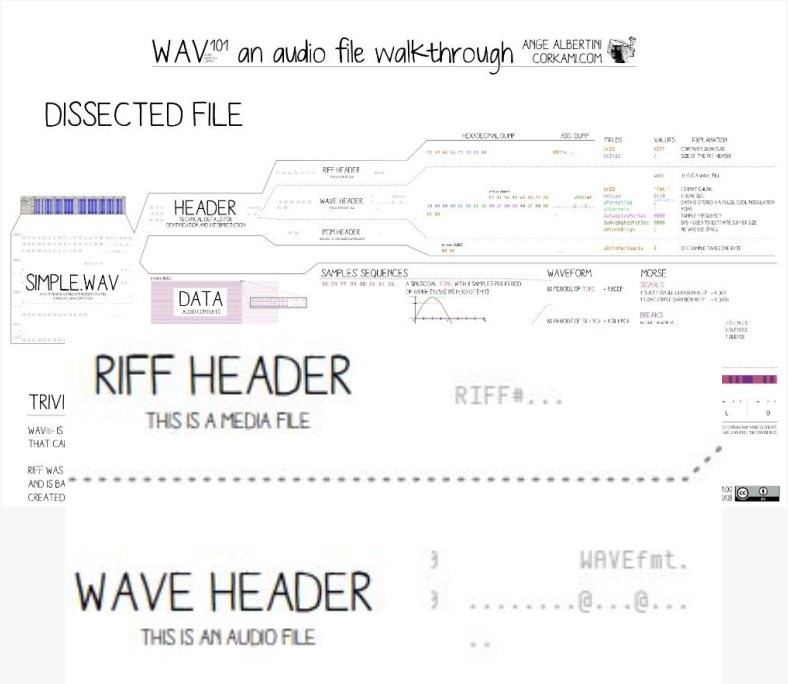


Audio

RIFF
WAVEfmt
dataA
fact
LISTB
INFOISFT5
GoldWave (C) Chris S. Craig,
http://www.goldwave.com

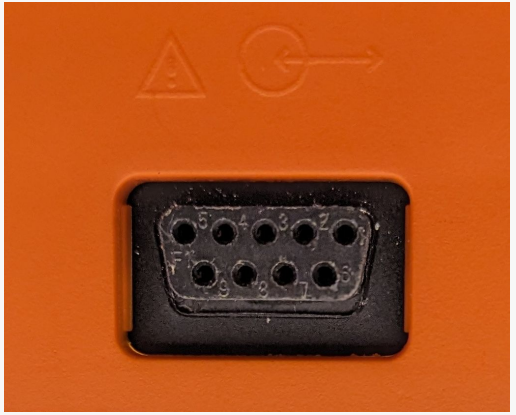
- While scrolling through the strings
  - Ange Albertinis posters are a good reminder
- There are multiple wav headers in the image
- Thus: foremost to the rescue

```
Length: 16 MB (16777216 bytes)

Num      Name (bs=512)          Size      File Offset     Comment

0:      00000532.wav            9 KB          272524
1:      00000551.wav           12 KB          282370
2:      00000575.wav           28 KB          294904
3:      00000632.wav            8 KB          324070
4:      00000708.wav            8 KB          362996
5:      00000725.wav           16 KB          371492
6:      00000759.wav           14 KB          388688
7:      00000841.wav           18 KB          430824
8:      00000877.wav           29 KB          449398
9:      00000936.wav           34 KB          479658
10:     00001006.wav            4 KB          515328
11:     00001016.wav            1 KB          520278
12:     00001162.wav           18 KB          595324
13:     00001199.wav           10 KB          614138
14:     00001221.wav           12 KB          625202
15:     00001246.wav           13 KB          638242
16:     00001274.wav            6 KB          652512
```



Serial Port

- My previous enemy
- The serial port is isolated using a MAX232E
  - Shifting the 3.3V from the EP7312 to actual RS232 levels
- Still nothing to be seen on the bus
  - Thought it might be broken and I replaced the MAX232E
  - No difference



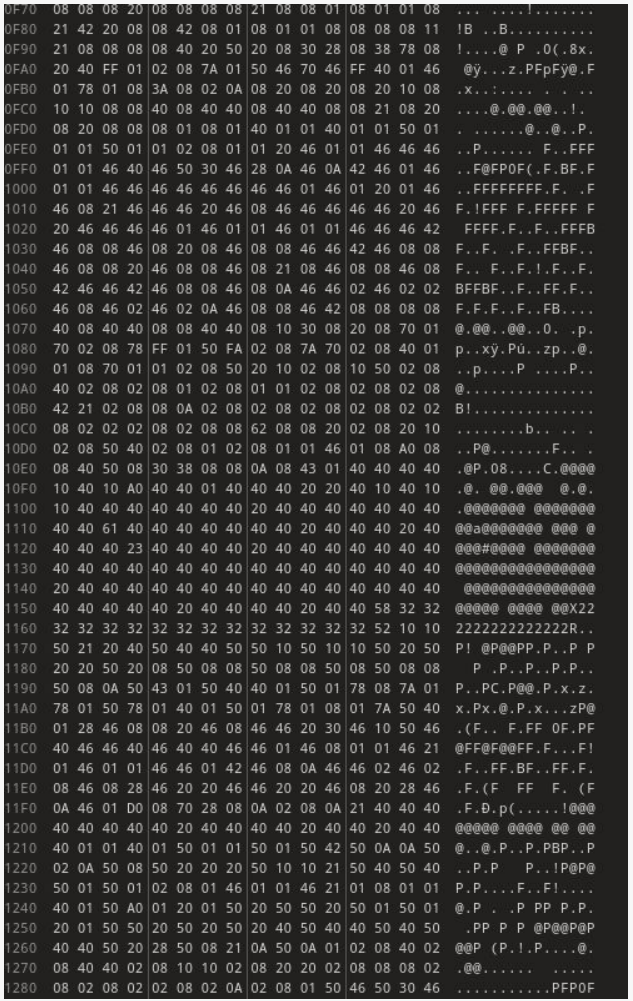
Pin	Signal
1	GND
2	
3	DOUT from MAX232
4	RIN to MAX232
5	R210, 12V PullUp
6	2.5V
8	
9	

Serial Port

- Lazy Approach
  - Fly wires to the Pins on the MAX232
  - Logic Analyzer & USB to Serial Adapter

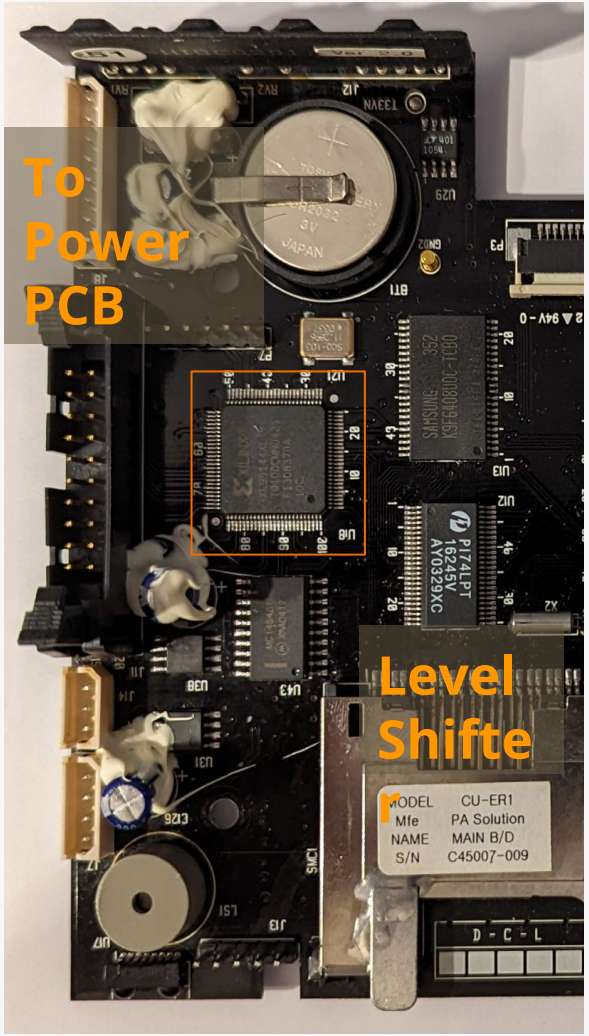
Serial Port

- Lazy Approach
  - Fly wires to the Pins on the MAX232
  - Logic Analyzer & USB to Serial Adapter
- Print works!
- Software connection does nothing
  - I guess the communication has to be initialized by the software
  - Sadly didn’t have a copy



CPLD

- Connected to the main controller via a Level Shifter
- Connected directly to the Power PCB via multiple control lines



CPLD

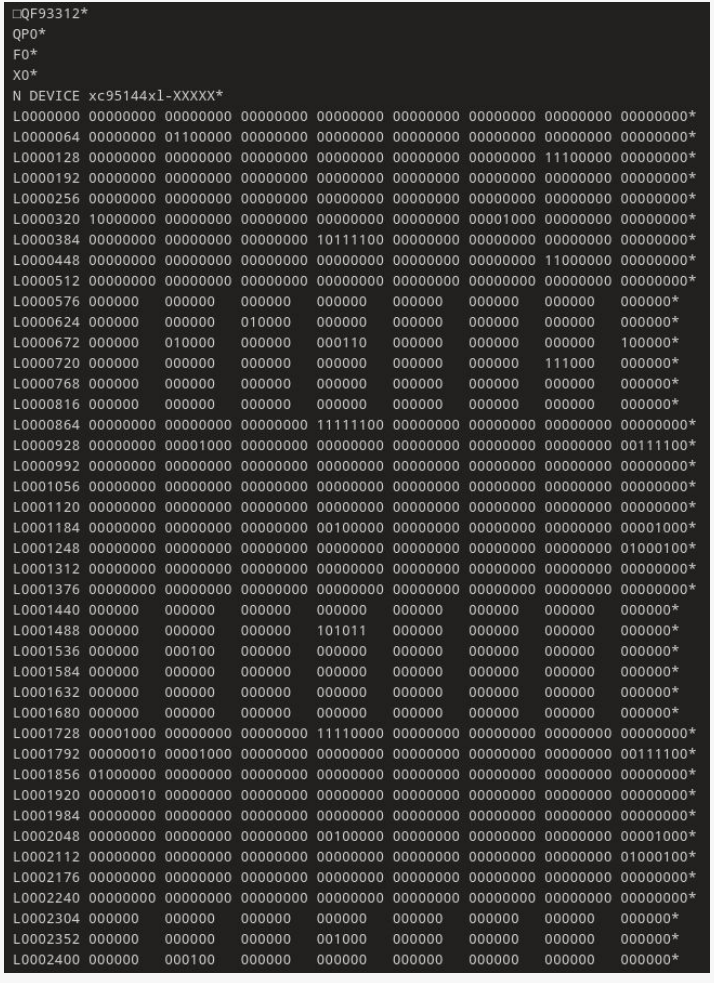
- Connected to the main controller via a Level Shifter
- Connected directly to the Power PCB via multiple control lines
- JTAGulator confirms a functioning JTAG interface
  - P7: 1/TDO, 2/TDI, 3/TMS, 4/TCK, 5/GND, 6/VCC

```
JTAG> J
Enter starting channel [0]:
Enter ending channel [4]:
Possible permutations: 120

Bring channels LOW before each permutation? [y/N]:
Press spacebar to begin (any other key besides Enter to abort)...
JTAGulating! Press any key to abort...

TDI: 1
TDO: 0
TCK: 3
TMS: 2
Device ID #1: 0100 1001011000001000 00001001001 1 (0x49608093)


JTAG combined scan complete.
```

```
JTAG> D
TDI not needed to retrieve Device ID.

Enter TDO pin [0]:
Enter TCK pin [3]:
Enter TMS pin [2]:

Device ID #1: 0100 1001011000001000 00001001001 1 (0x49608093)
-> Manufacturer ID: 0x049
-> Part Number: 0x9608
-> Version: 0x4

IDCODE listing complete.
```

CPLD

- Being a Xilinx, I went back to the good old Xilinx ISE
- And a Digilent JTAG Adapter
- Reading the CPLD failed in Impact
  - But the command line works



```
□QF93312*
QP0*
F0*
X0*
N DEVICE xc95144xl-XXXXX*
L0000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000*
L0000064 00000000 01100000 00000000 00000000 00000000 00000000 00000000 00000000*
L0000128 00000000 00000000 00000000 00000000 00000000 00000000 11100000 00000000*
L0000192 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000*
L0000256 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000*
L0000320 10000000 00000000 00000000 00000000 00000000 00001000 00000000 00000000*
L0000384 00000000 00000000 00000000 10111100 00000000 00000000 00000000 00000000*
L0000448 00000000 00000000 00000000 00000000 00000000 00000000 11000000 00000000*
L0000512 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000*
L0000576 000000   000000   000000   000000   000000   000000   000000   000000*
L0000624 000000   000000   010000   000000   000000   000000   000000   000000*
L0000672 000000   010000   000000   000110   000000   000000   000000   100000*
L0000720 000000   000000   000000   000000   000000   000000   111000   000000*
L0000768 000000   000000   000000   000000   000000   000000   000000   000000*
L0000816 000000   000000   000000   000000   000000   000000   000000   000000*
L0000864 00000000 00000000 00000000 11111100 00000000 00000000 00000000 00000000*
L0000928 00000000 00001000 00000000 00000000 00000000 00000000 00000000 00111100*
L0000992 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000*
L0001056 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000*
L0001120 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000*
L0001184 00000000 00000000 00000000 00100000 00000000 00000000 00000000 00001000*
L0001248 00000000 00000000 00000000 00000000 00000000 00000000 00000000 01000100*
L0001312 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000*
L0001376 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000*
L0001440 000000   000000   000000   000000   000000   000000   000000   000000*
L0001488 000000   000000   000000   101011   000000   000000   000000   000000*
L0001536 000000   000100   000000   000000   000000   000000   000000   000000*
L0001584 000000   000000   000000   000000   000000   000000   000000   000000*
L0001632 000000   000000   000000   000000   000000   000000   000000   000000*
L0001680 000000   000000   000000   000000   000000   000000   000000   000000*
L0001728 00001000 00000000 00000000 11110000 00000000 00000000 00000000 00000000*
L0001792 00000010 00001000 00000000 00000000 00000000 00000000 00000000 00111100*
L0001856 01000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000*
L0001920 00000010 00000000 00000000 00000000 00000000 00000000 00000000 00000000*
L0001984 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000*
L0002048 00000000 00000000 00000000 00100000 00000000 00000000 00000000 00001000*
L0002112 00000000 00000000 00000000 00000000 00000000 00000000 00000000 01000100*
L0002176 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000*
L0002240 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000*
L0002304 000000   000000   000000   000000   000000   000000   000000   000000*
L0002352 000000   000000   000000   001000   000000   000000   000000   000000*
L0002400 000000   000100   000000   000000   000000   000000   000000   000000*
```

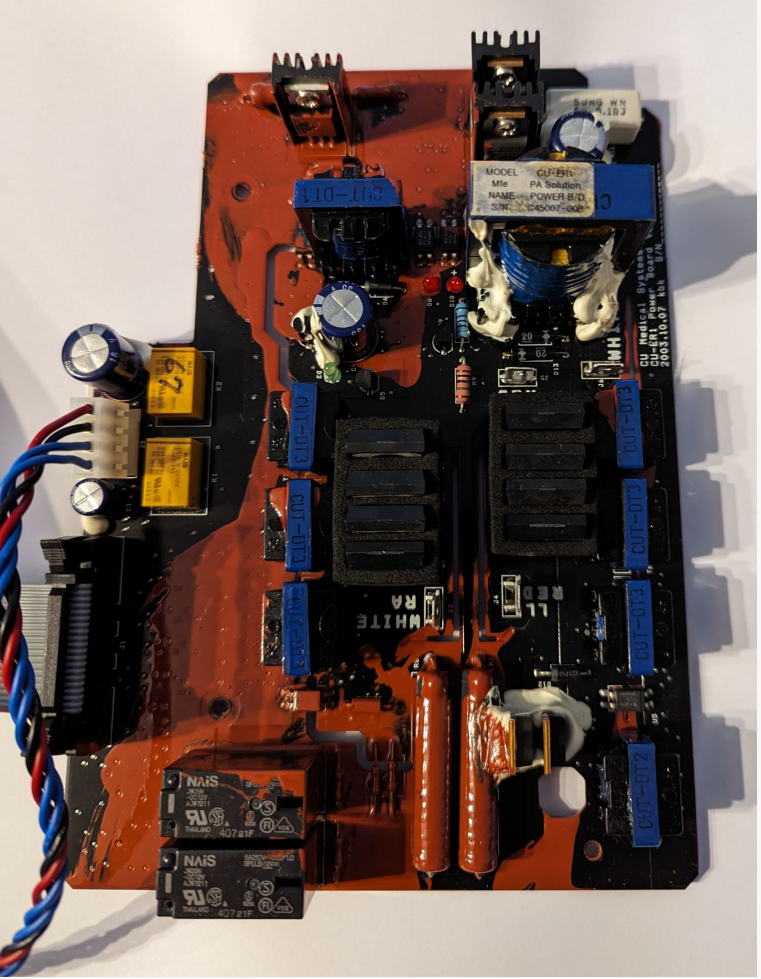
Firmware Update

- According to the manual and eBay, where I got mine, the device supports updates
- After doing some reading, it seems the updates are performed by service technicians during maintenance
- My guess: Serial and the BootBotton due the job

Maverick from CIRRUS LOGIC

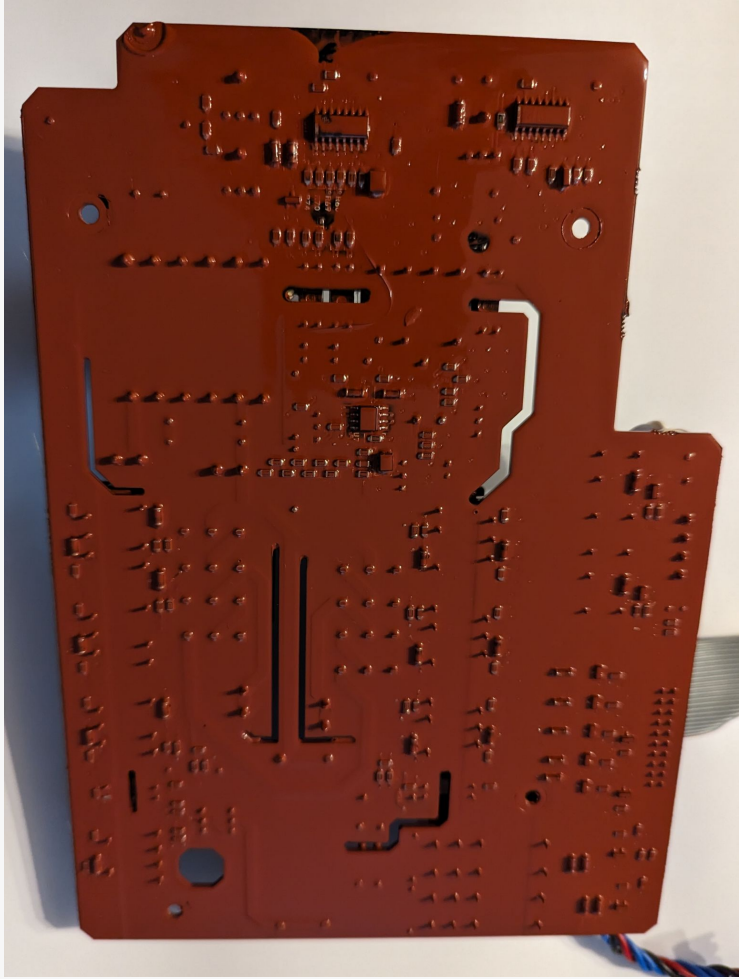
8. APPENDIX A: BOOT CODE

```
00000000          uart_boot_base
00000000 E3A0C102         MOV     r12, #HwRegisterBase ; R12 = 0x80000000
00000004
00000004 E3A08201         MOV     r8, #InternalRamBase ; R8 = 0x10000000
00000008 E2889B02         ADD     r9, r8, #ImageSize ; R9 = 0x10000800
0000000C
0000000C          ;;; The remaining code is functionally identical to the 7111 boot code
0000000C
0000000C          ;;; First, initialize HW control of UART
0000000C
0000000C 00000480 Hw_UARTDR1    EQU     0x0480
0000000C
0000000C 000004C0 Hw_UBRLCR1    EQU     0x04c0
0000000C 00000017 Hw_BR9600     EQU     0x00000017  ; 9600 baud divisor = 23
0000000C 0000000B Hw_BR9600_13  EQU     0x0000000b  ; 9600 baud divisor = 11
0000000C 00060000 Hw_WRDLEN8    EQU     0x00060000
0000000C
0000000C E3A00C01         MOV     r0, #Hw_UART1EN ; Enable UART
00000010 E58C0100         STR     r0, [r12, #Hw_SYSCON]
00000014
00000014 E28C1D45         ADD     r1, r12, #Hw_SYSFLG2 ; (was LDR, ADD in 7111 code)
00000018 E5917000         LDR     r7, [r1]     ; R7 = SYSFLG2
0000001C
0000001C E3170040         TST     r7, #Hw_CKMODE
00000020 13A0000B         MOVNE   r0, #Hw_BR9600_13 ; Load 13 MhZ value if bit set
00000024 03A00017         MOVEQ   r0, #Hw_BR9600 ; If not set, load other divisor
00000028 E3800806         ORR     r0, r0, #Hw_WRDLEN8 ; Insert 8-bit character mode
0000002C
0000002C E58C04C0         STR     r0, [r12, #Hw_UBRLCR1]
00000030
00000030 0000003C StartFlag EQU     '<'
00000030 0000003E EndFlag EQU       '>'
00000030
00000030          ;;; Send ready signal
00000030 E3A0003C         MOV     r0, #StartFlag
00000034 E58C0480         STR     r0, [r12, #Hw_UARTDR1]
00000038
00000038          ;;; Receive the data
00000038          ;;; Store bytes at R9 address, stop loop when R8 == R9
00000038          ;;; Leaves R8 set to 0x10000800
00000038
00000038          ;;; Wait for byte to be available
00000038
00000038          uart_ready_loop
00000038 E59C1140         LDR     r1, [r12, #Hw_SYSFLG] ; Spin, if Rx FIFO is empty
0000003C E3110501         TST     r1, #Hw_URXFE1
00000040 1AFFFFFC         BNE     uart_ready_loop
00000044
00000044          ;;; Read the data, store it, and accumulate checksum
00000044 E59C0480         LDR     r0, [r12, #Hw_UARTDR1] ; Read data
00000048 E4C80001         STRB    r0, [r8], #1 ; Save it in memory
0000004C E1580009         CMP     r8, r9
00000050 BAFFFFF8         BLT     uart_ready_loop ; Do more if end of buffer not reached
00000054
00000054          ;;; All received, send end flag
```

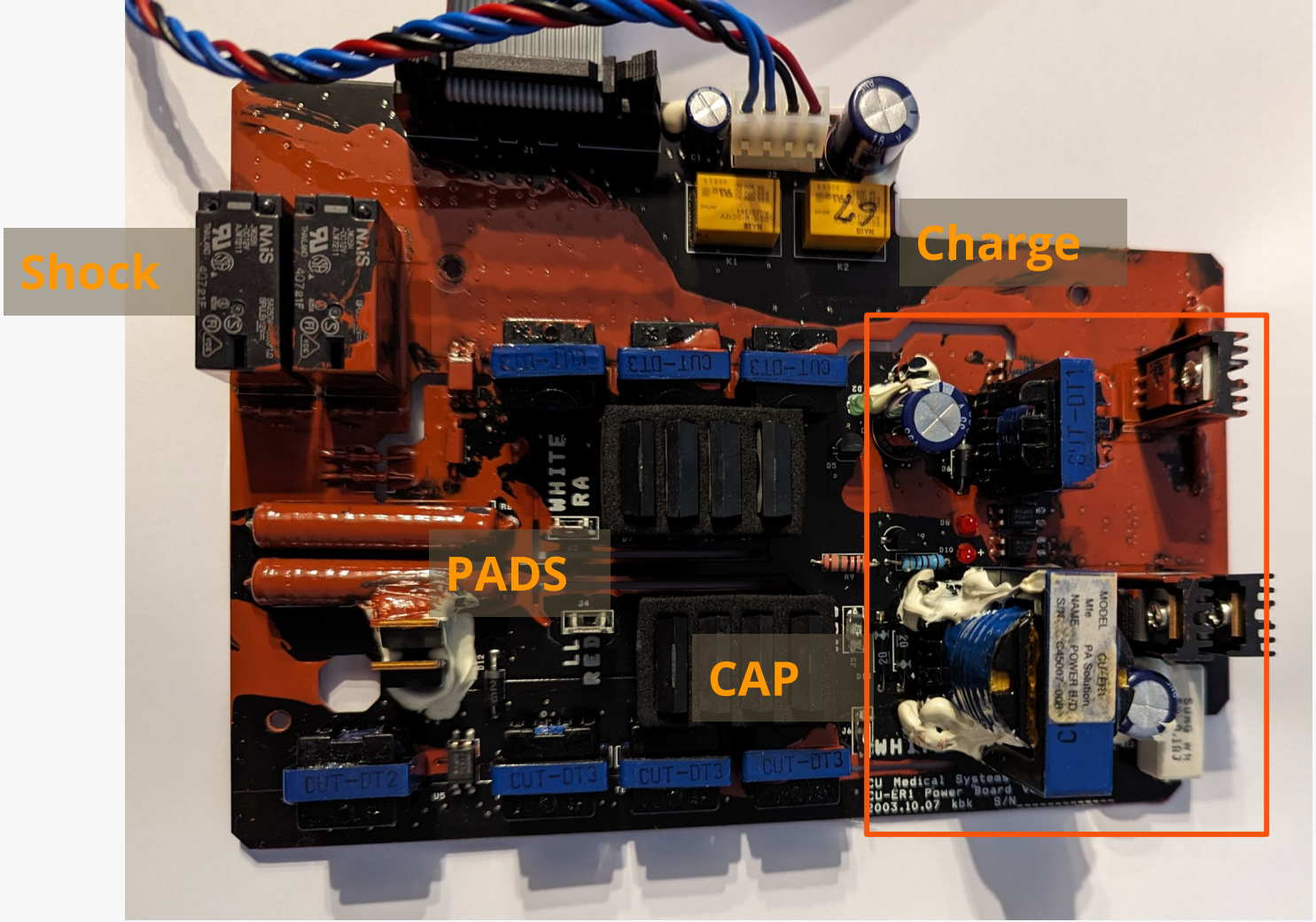


Top

Bottom

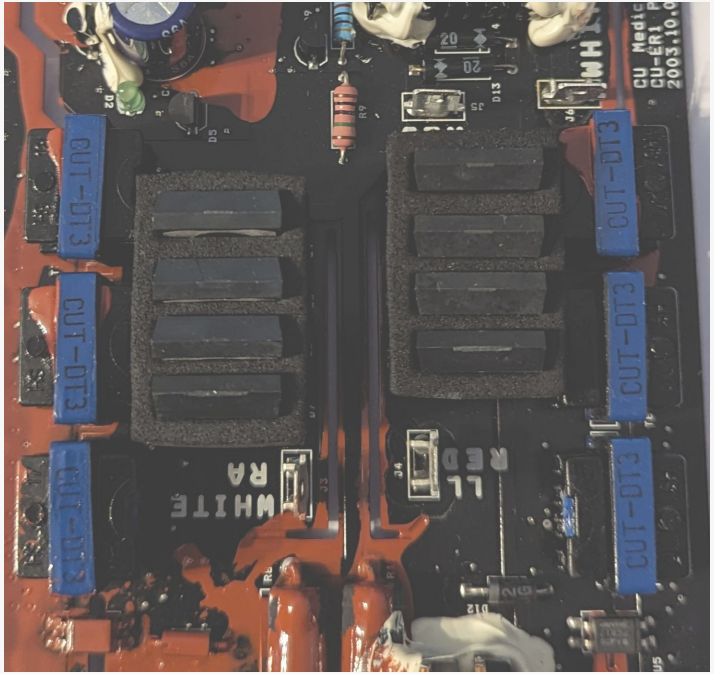


Power PCB


Shock
Charge
PADS
CAP

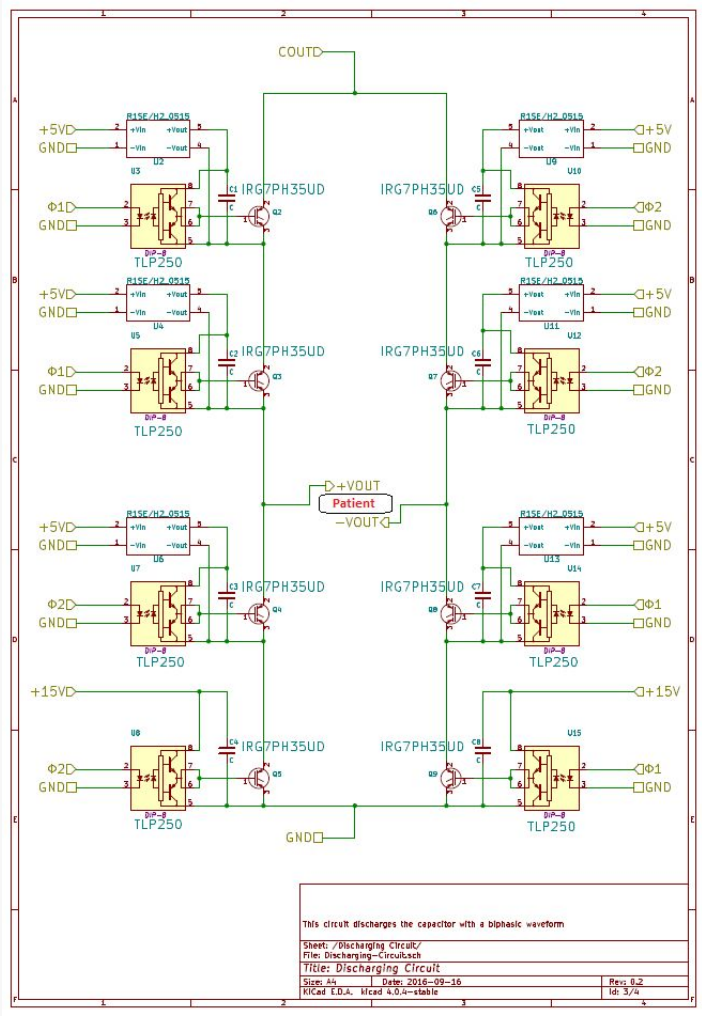
Circuit

- The shape of the circuit itself shows a bit of symmetry
  - And might remind somebody of the letter H
- And that’s just what it is, an H-Bridge



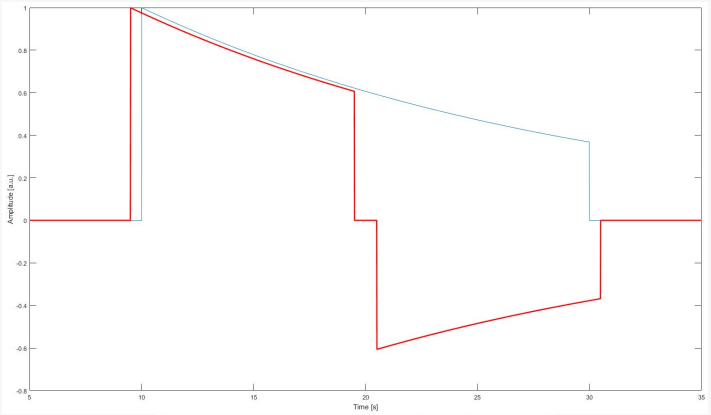
Circuit

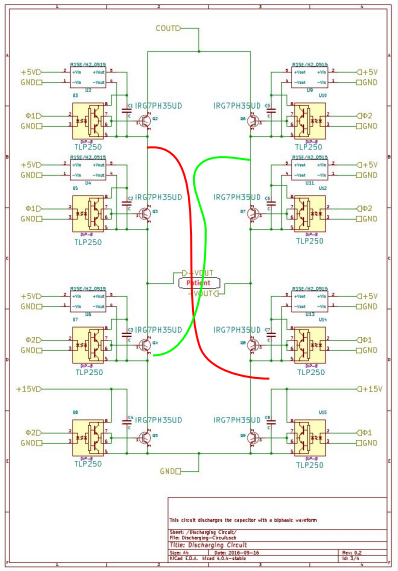
- There is a nicely document project called Open AED on GitHub, which was financed by the EU
- Resulting in a setup, which looks very similar to the commercial solution
- And also brings further explanation



Biphasic Defibrilation
(Yet again, not a doctor)

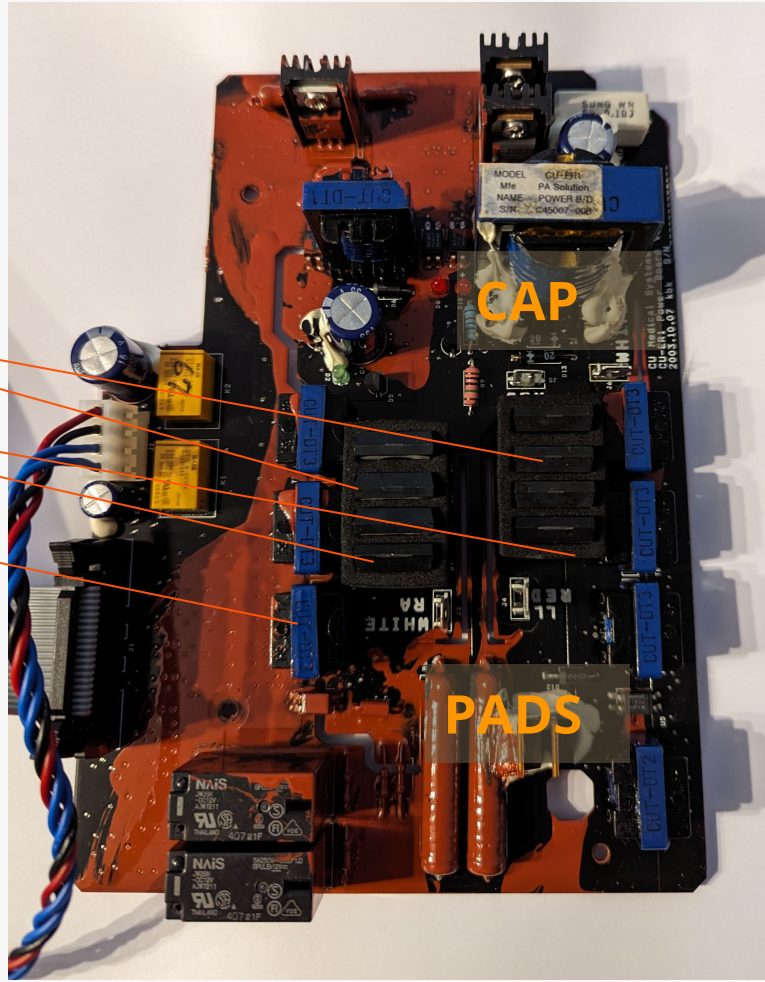
- So the current state of the art defibrillation is not a single, one directional pulse
- But includes reversing the polarity
  - Alternating polarity is exactly what an H-Bridge is used for when i.e. running motors





Applicable Components

- 3 x YXIS CS30-16io Thyristors
- IRG4PH50 Transistor
- Transformersf



Summary & Security

Well . . .

Security?

- Cirrus Logic EP7312
  - Well, Nope
- Xilinx XC95144XL
  - Nope

Shameful?

- The device was created to solve a problem and that’s what it does
- It’s from a time when there was next to no focus on security for embedded devices
- The core focus is on safety, thus all the self-checks etc.

Risky?

- Who would ever reprogram a defibrillator?....

Thanks for your Time
Questions?

Slides, Ressources etc. will be posted on

https://security-bits.de

next week

Sources

- Slide 3: https://commons.wikimedia.org/wiki/File:Ventricular_fibrillation.png
- Slide 4: Movie Crank, Google search
- Slide 6: https://tesladownunder.com/Red%20Alert%20Tesla.htm#Construction
- Slide 7: https://commons.wikimedia.org/wiki/File:Implantable_cardioverter-defibrillator.jpg
- Slide 15: FTDI DS_UT232R-200(500) Datasheet
- Slide 16: MAX3221E Datasheet
- Slide 50, 51: https://github.com/CentroEPiaggio/Open-Automated-External-Defibrillator