

Anne Isabelle “Anya” Lovelace

Hacking for fun (and not just for profit)

ou, a jornada de uma hacker



whoami

- anne isabelle “anya”
- /bin/nologin
- user id 65534 (nobody)
- eu faço coisas interessantes
nas horas vagas
- autista, trans e garota gato

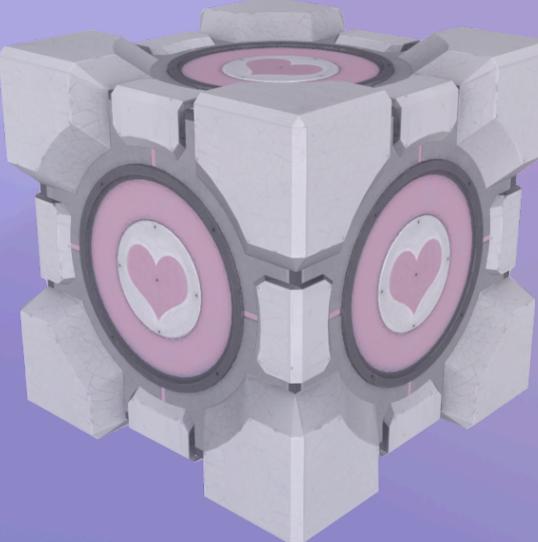




**push %rbp
mov %rsp, %rbp
sub 8, %rsp**

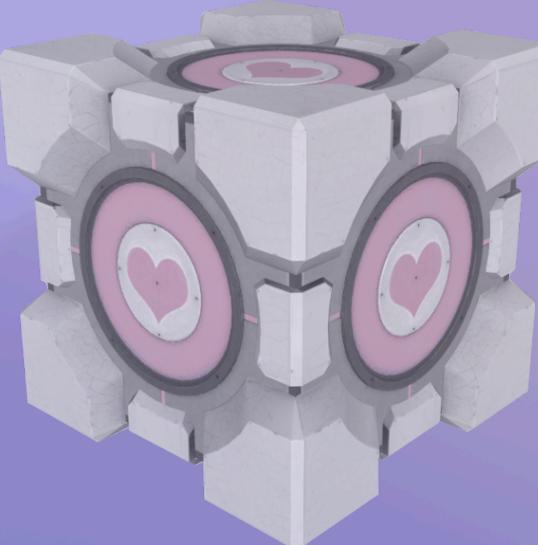
eu não sou uma profissional de sec

- **eu não trabalho com segurança desde 2018**
- **eu não sei o que tá acontecendo no LinkedIn ou no mundo corporativo**
- **eu não sei quais são as melhores certificações de cyber**



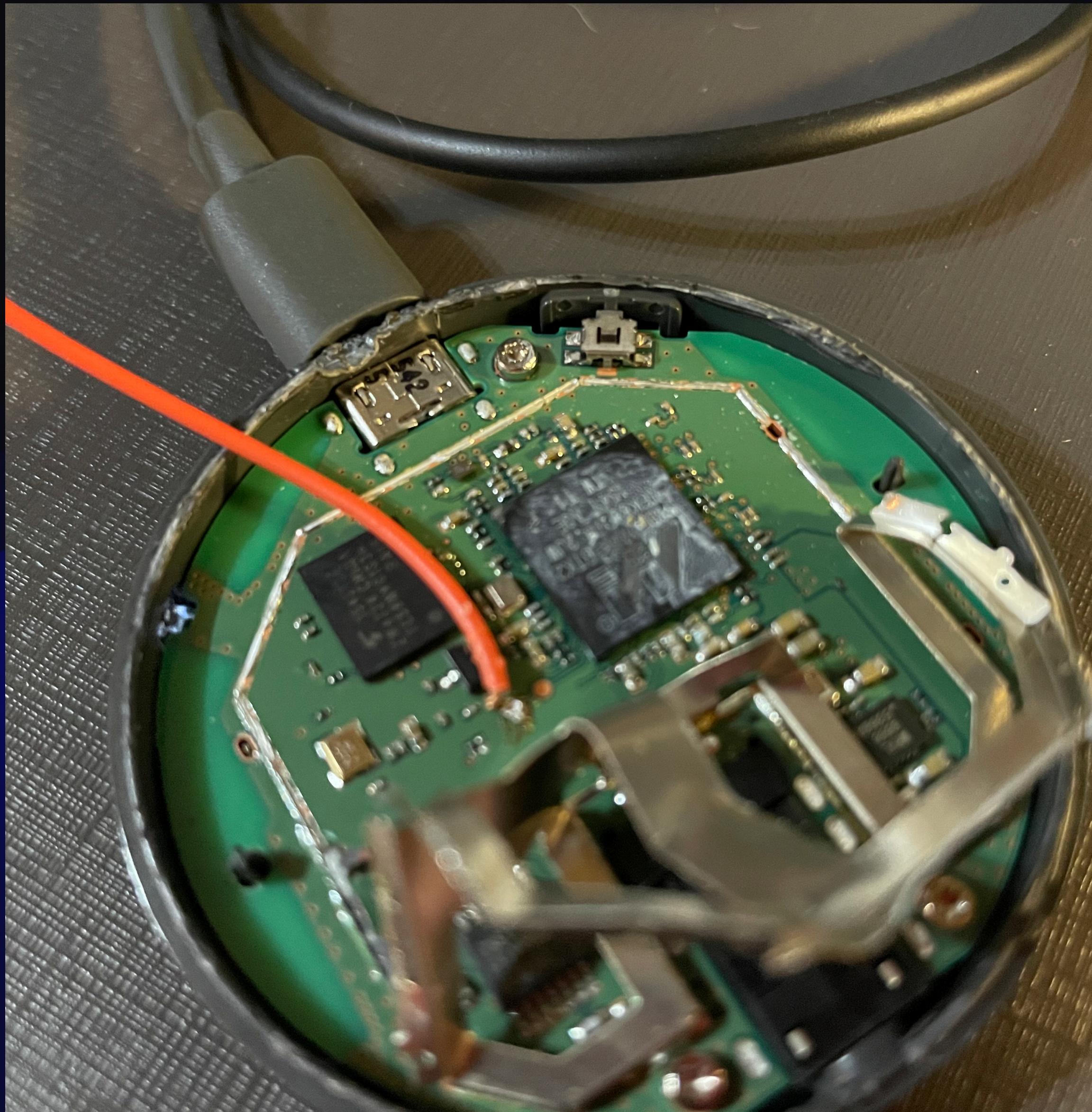
eu sou curiosa

- **eu gosto de entender como coisas funcionam**
- **eu gosto de subverter o funcionamento das coisas**
- **eu gosto de estudar sobre coisas que poucas pessoas se interessam em estudar**
- **hiperfocos são meu modo default (por conta da neurodivergência)**



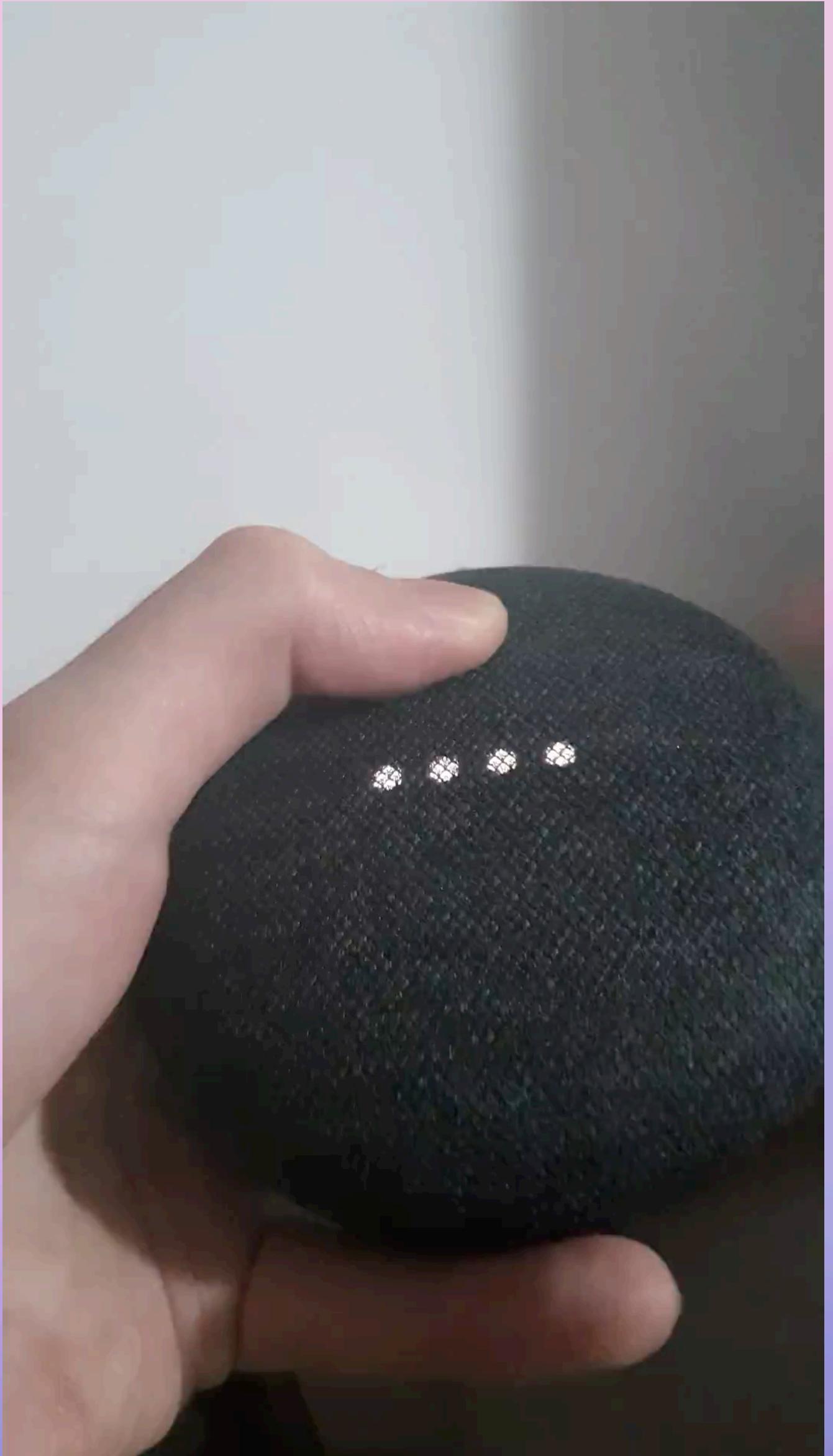
chromecast

Originalmente publicado no
Tramoia.sh 0x01



o dia em que meu aparelho espião da google morreu

- em um determinado dia, a Google lançou uma atualização ruim que brickou meu Google Home.**
- isso me fez ter a seguinte dúvida: o que diabos esses aparelhos rodam?**



o dia em que meu aparelho espião da google morreu

- encontrei certo dia um post de alguém que teve o mesmo problema e decidiu abrir o aparelho pra analisar.

Evan's Techie-Blog

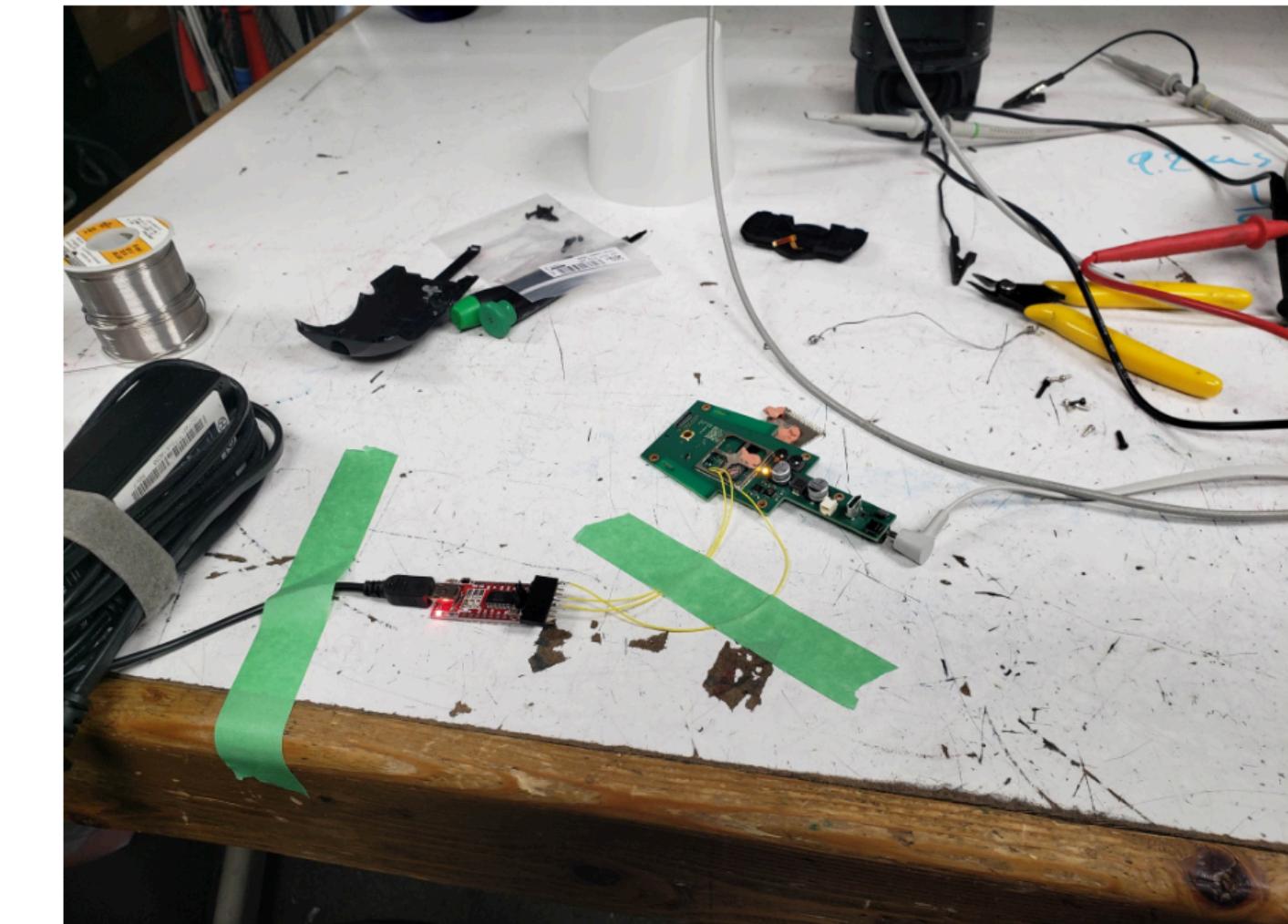
Hacks, repairs, arcade games, sci-fi, and some very bad ideas with possibly humorous consequences

[« Designing a watchdog timer](#)

[Strange parallel ROM pinouts »](#)

Google Home Autopsy

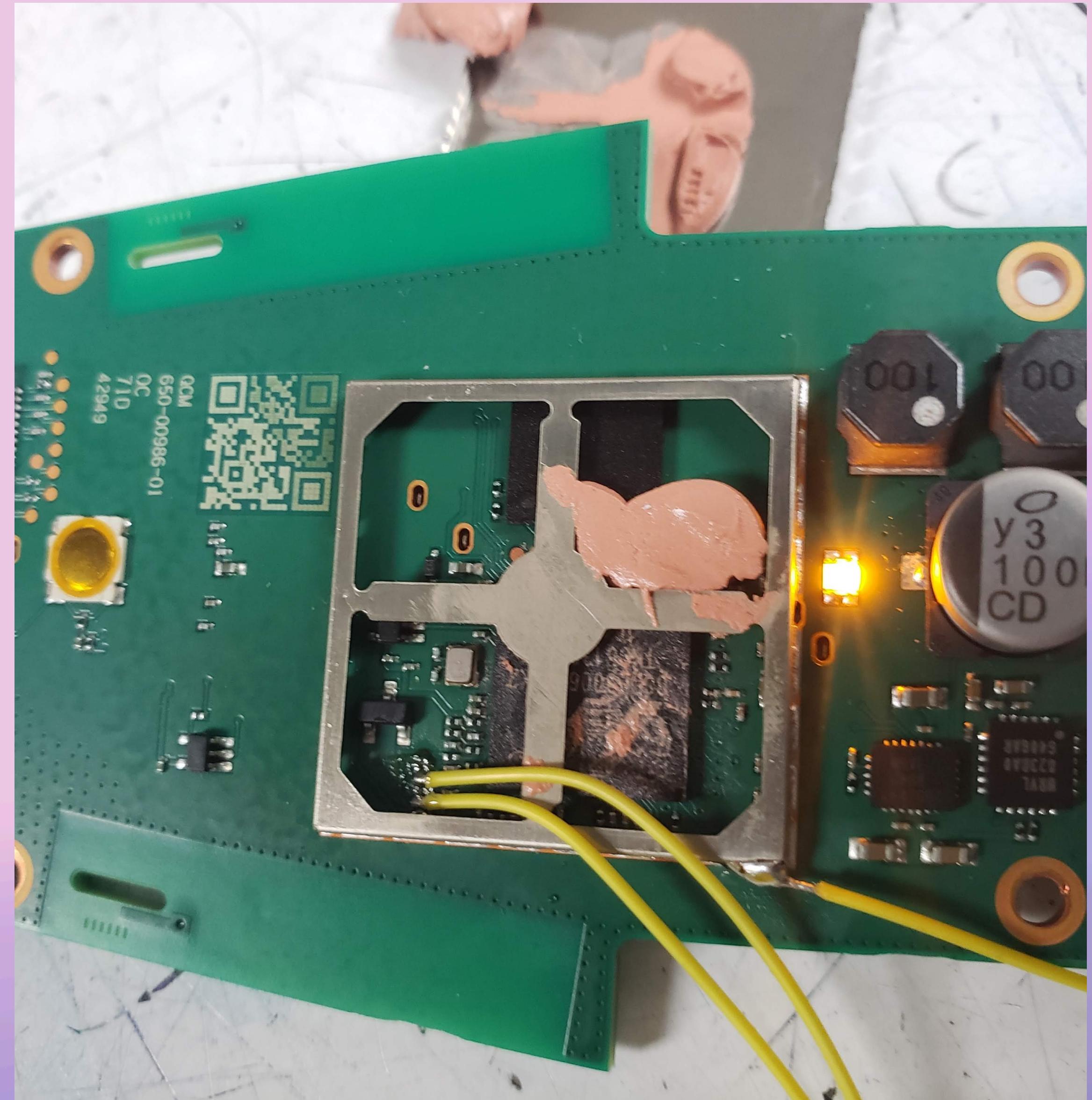
I had a google home die the other week. The internet claimed it could have been a bad OTA flash and that I should get support to replace it even out of warranty. I called and found out that (at least my rep) would extend the warranty out to two years, but not three which I would need. I also found out that there aren't any warranty void stickers inside or anything so I would have taken one of my google homes that was in warranty and swapped out the misbehaving motherboard but that seemed overly complicated and I thought I maybe had a solution anyway.



One thing I did remember was that the google home has a 'hidden' usb micro port that is for 'service'. After some research it seems that no one has hacked this thing using that port, but there are reports that you can

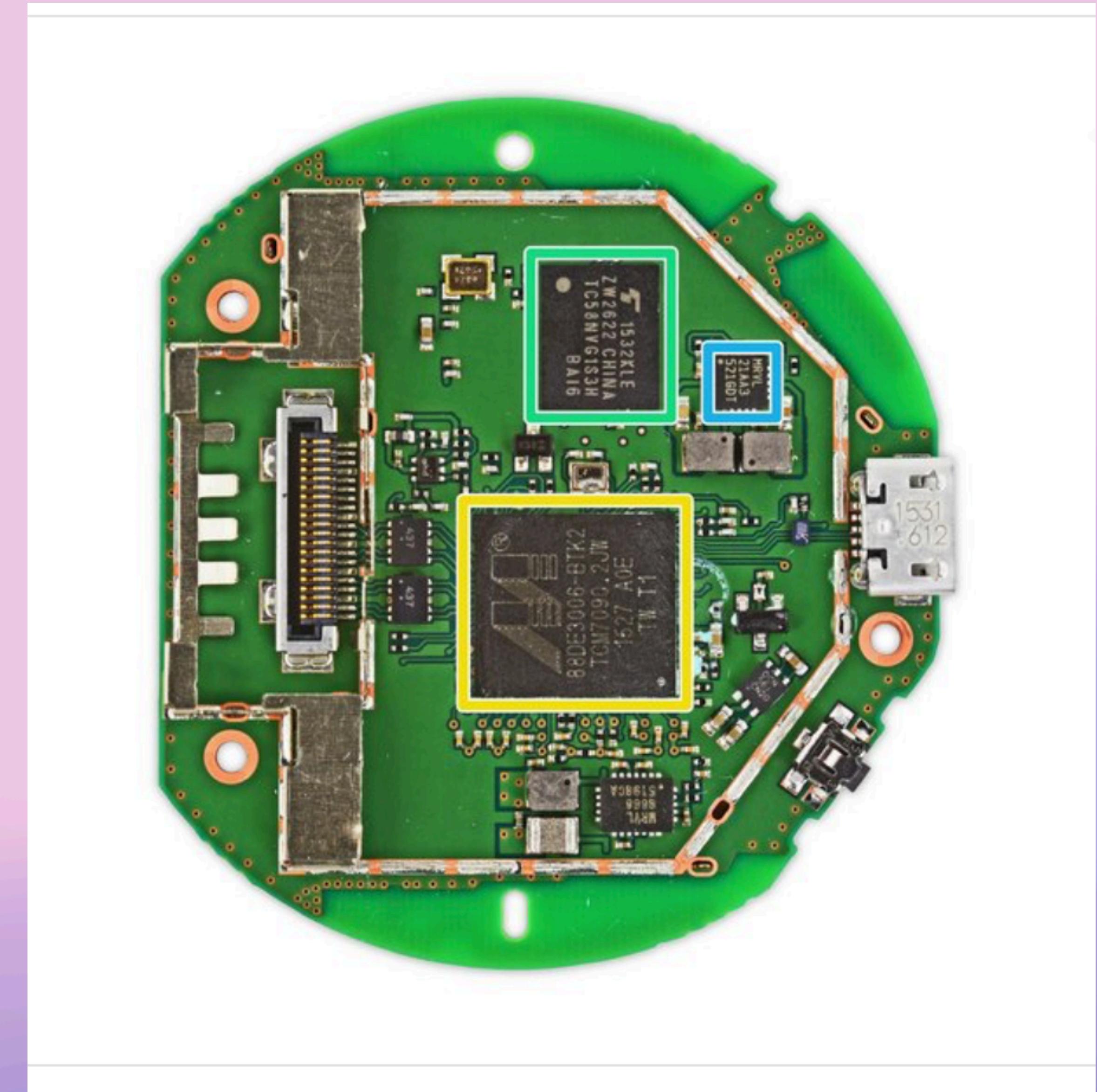
o dia em que meu aparelho espião da google morreu

- o post tinha uma foto de cabinhos próximos ao processador, embaixo do shield de RF.**
- esses cabinhos foram soldados em test-pads pelo autor do post, e com isso ele conseguiu analisar os logs do aparelho via serial.**



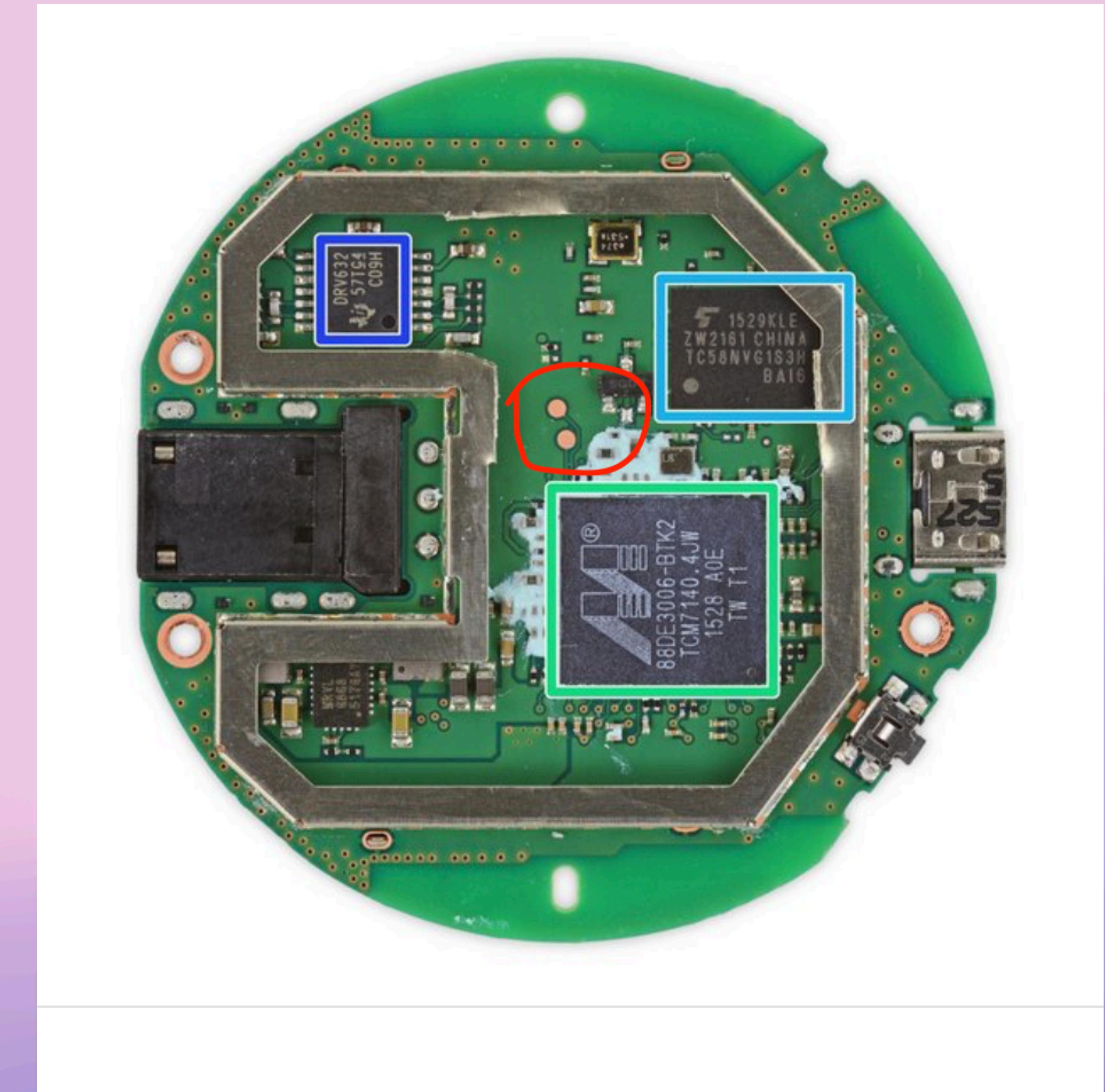
teardowns

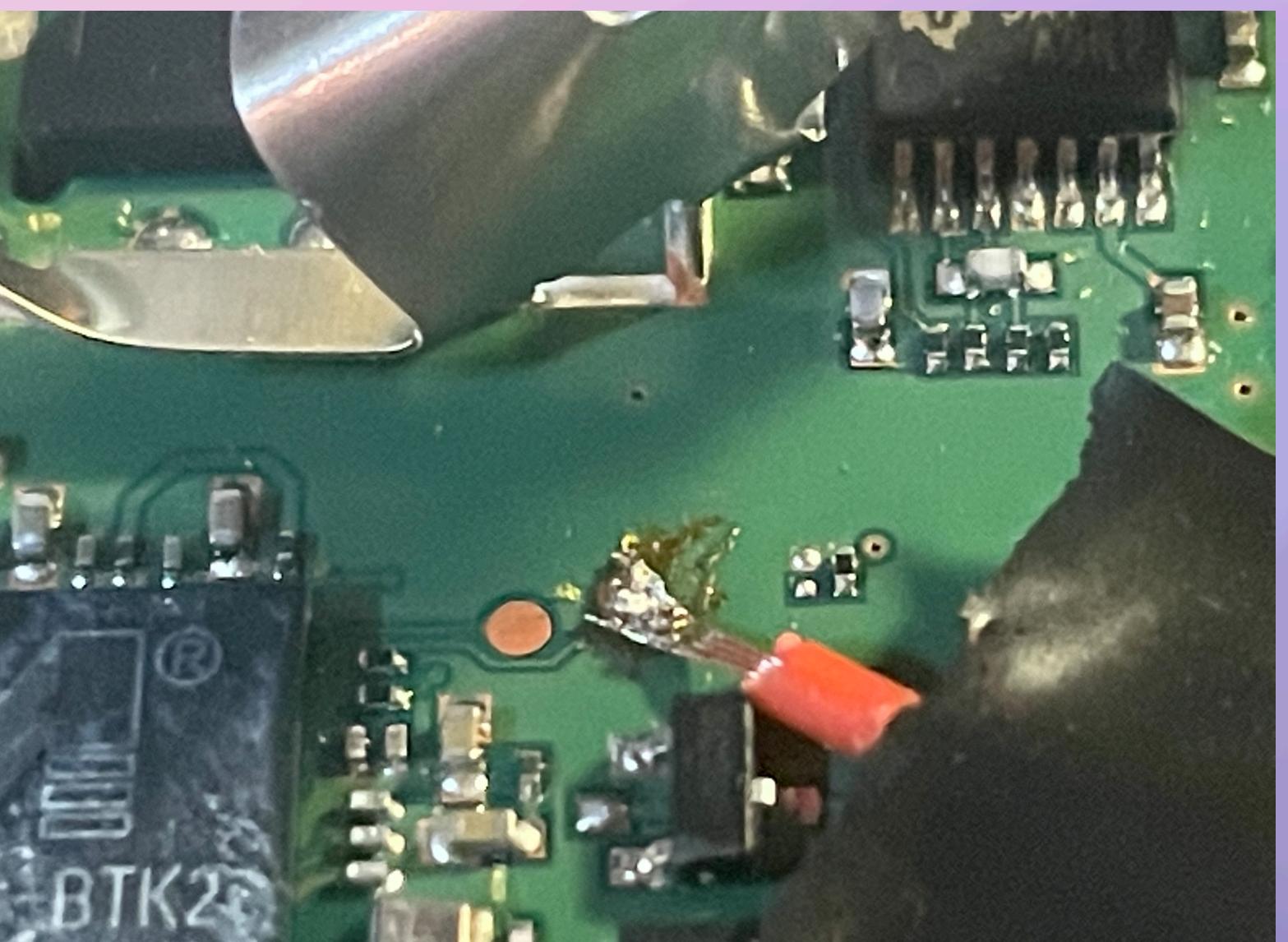
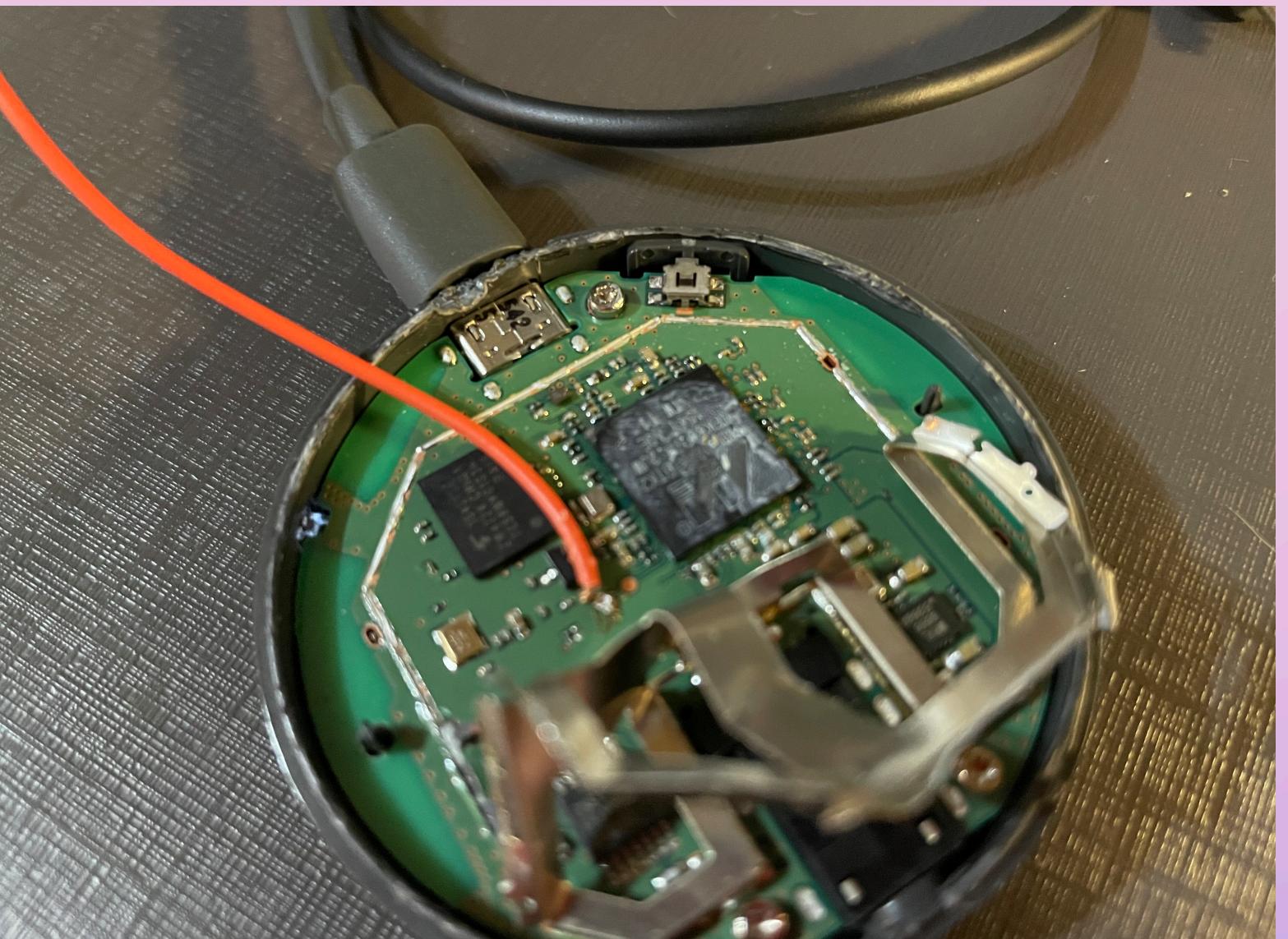
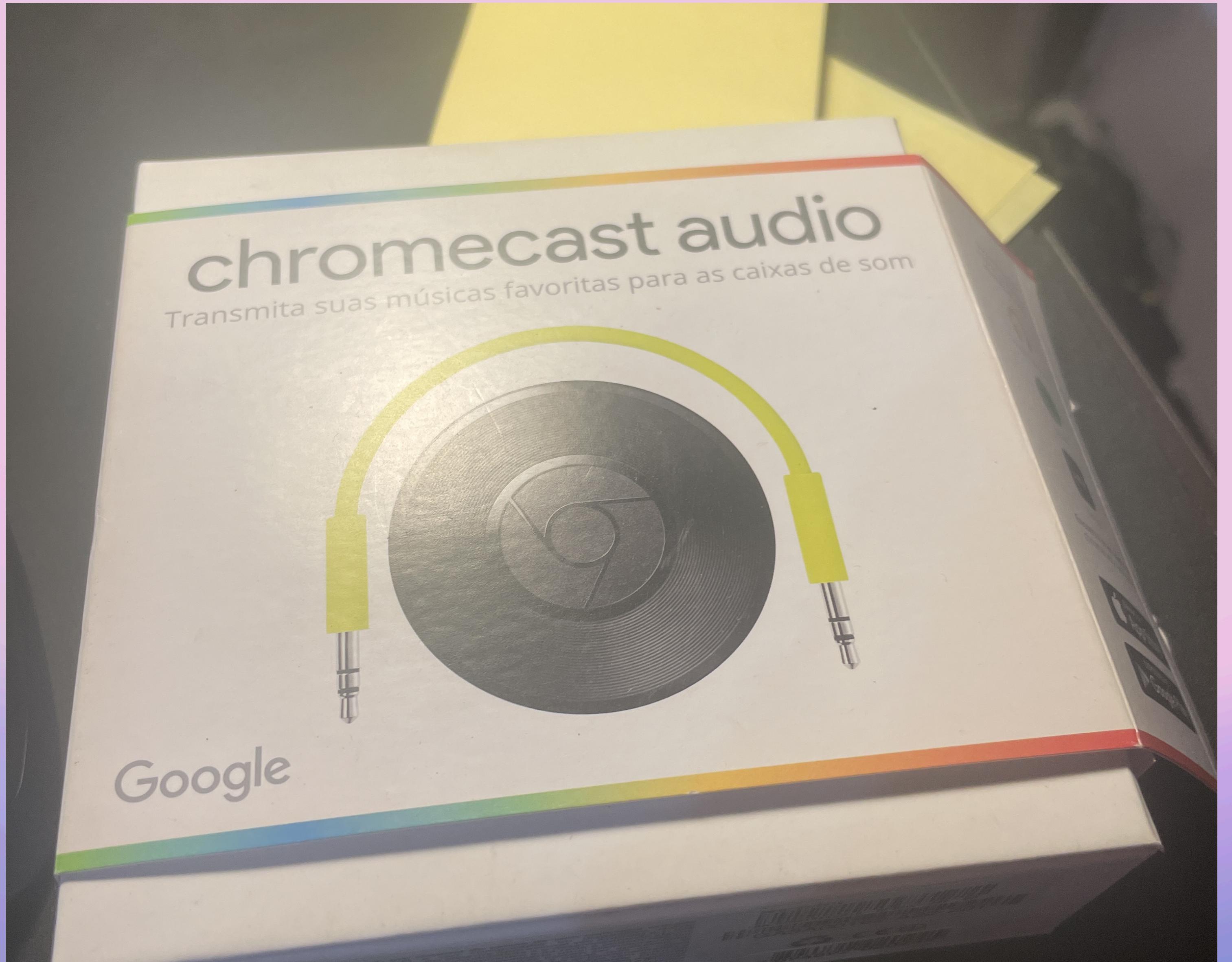
- eu procurei na internet mais informações sobre o SoC que esses aparelhos rodam (um Marvell ARMADA 1500 Mini Plus) e imaginei que fossem aparelhos muito parecidos com chromecasts.
- decidi então procurar teardowns no iFixIt

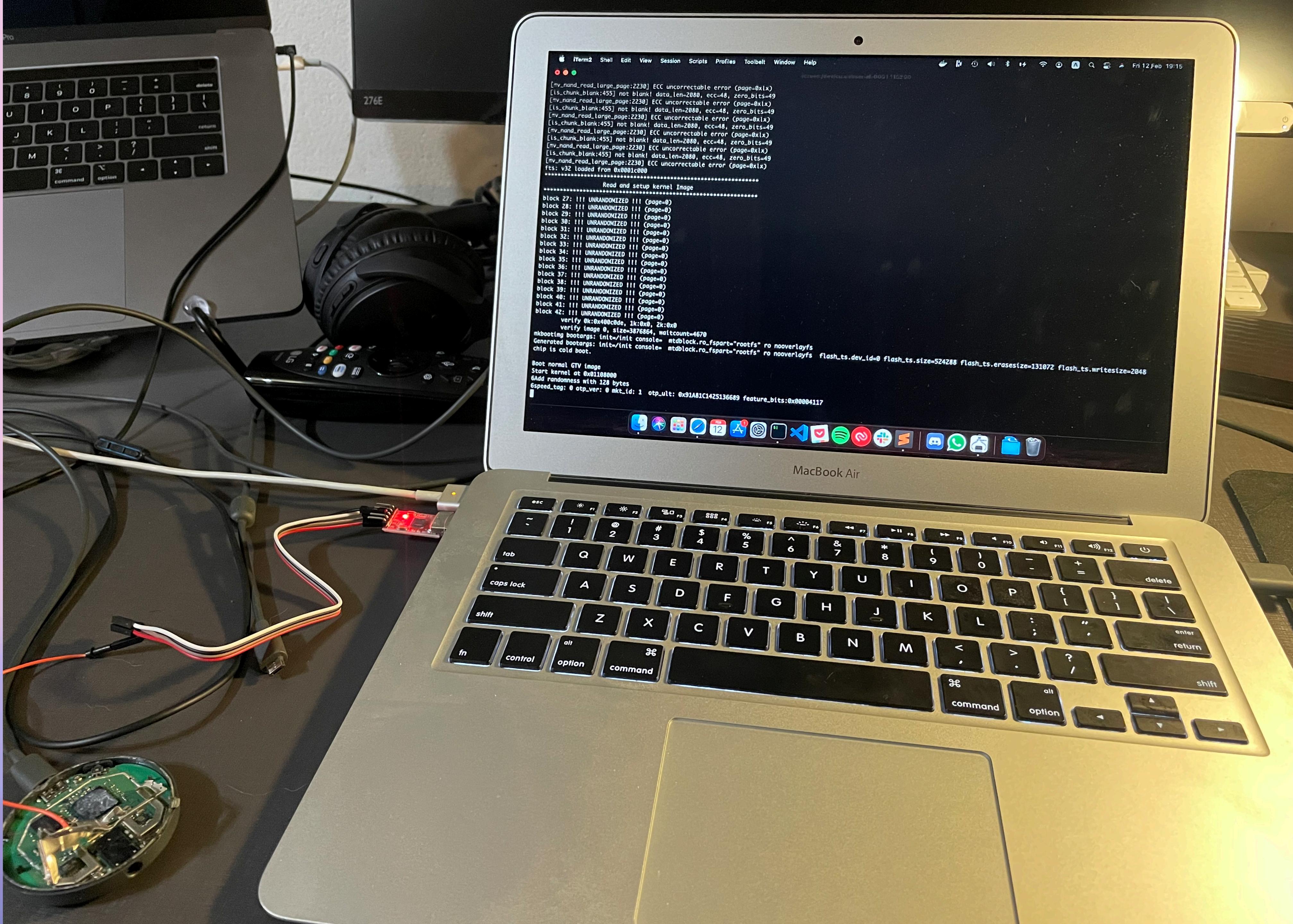


teardowns

- eu tinha um Chromecast 2 em casa, mas ele não tinha nenhum testpad.
- porém, o chromecast audio tinha!





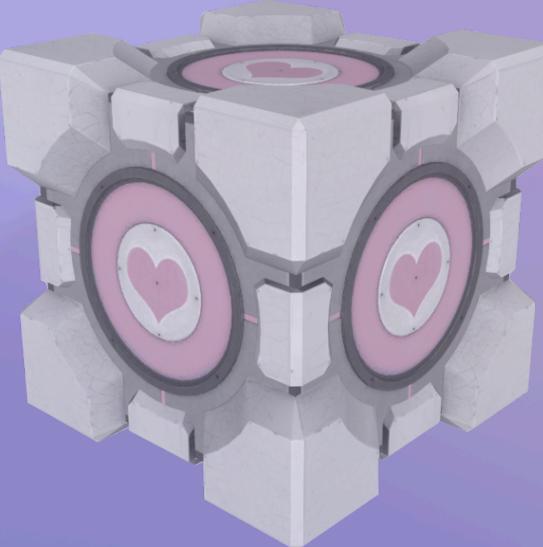


sauce plis



sauce

- o colega do post deixou alguns logs do boot do Google Home no blog, e com isso eu consegui encontrar um source do bootloader no Pastebin.
- de certa forma, tudo isso deveria ser OSS, mas a Google guardou o código de forma bem inconveniente – num tar.gz dentro de um Google Drive



Open source licenses > Hardware open source licenses > Google Chromecast & Android TV

Google Chromecast & Android TV

Chromecast with Google TV Devices

- [Open Source Code](#)

Chromecast Devices

- [Open Source Code](#)
- [Open Source Licenses](#)

Google Cast for Android TV

- [Open Source Code](#)
- [Open Source Licenses](#)

 Give feedback about this article

Hardware open source

-  [Google Chromecast](#)
-  [Google Nest speakers and displays](#)
-  [Nest Wifi, Google WiFi](#)
-  [Nest Cameras \(2021 models and later\)](#)
-  [Nest products open source](#)
-  [Onn cameras open source](#)

[Google Drive Link to Open Source Code for Chromecasts, Nest Speakers and Displays, and Nest Wifi.](#)

[Google Drive Link to Open Source Code for Chromecast with Google TV.](#)

[Google Drive Link to Open Source Code for Nest Cameras \(2021 models and later\)](#)

Nome 

 Kernel_Bootloader_SDK 

 Nest Audio 

 Nest Hub with Sleep Sensing 

 Nest Mini 

 eigen.tgz  

boot por USB

- Com o código do bootloader, eu consegui conferir algo muito interessante – como tentar fazer boot por USB.
- O aparelho tem um botão de reset que, quando pressionado com um cabo OTG e um pendrive, ele tenta recuperar a imagem do pendrive.
- Antes de ter acesso aos logs, eu tentei inferir as condições usando LEDs do pendrive como side-channels.
- Eu consegui acertar os magicos usando a ferramenta xxd e dd do UNIX pra pular todas as verificações de magic.

```
if (boot_src == BOOT_SRC_USB){  
    /* TODO(kolla): Enable checks for ver, mkt_id etc.*/  
    if (img_magic != CPU_IMG_CODE_MAGIC)  
        return -1;  
    if (code_type != BCM_IMG_USBIMG_TYPE)  
        return -1;  
    if (img_udata != CPU_IMG_USB_USRDATA)  
        return -1;
```

Os defines são:

```
#define CPU_IMG_CODE_MAGIC      0xC0DE  
#define BCM_IMG_KERNEL_TYPE    4  
#define BCM_IMG_USBIMG_TYPE   5  
#define CPU_IMG_USB_USRDATA   0xA33A
```

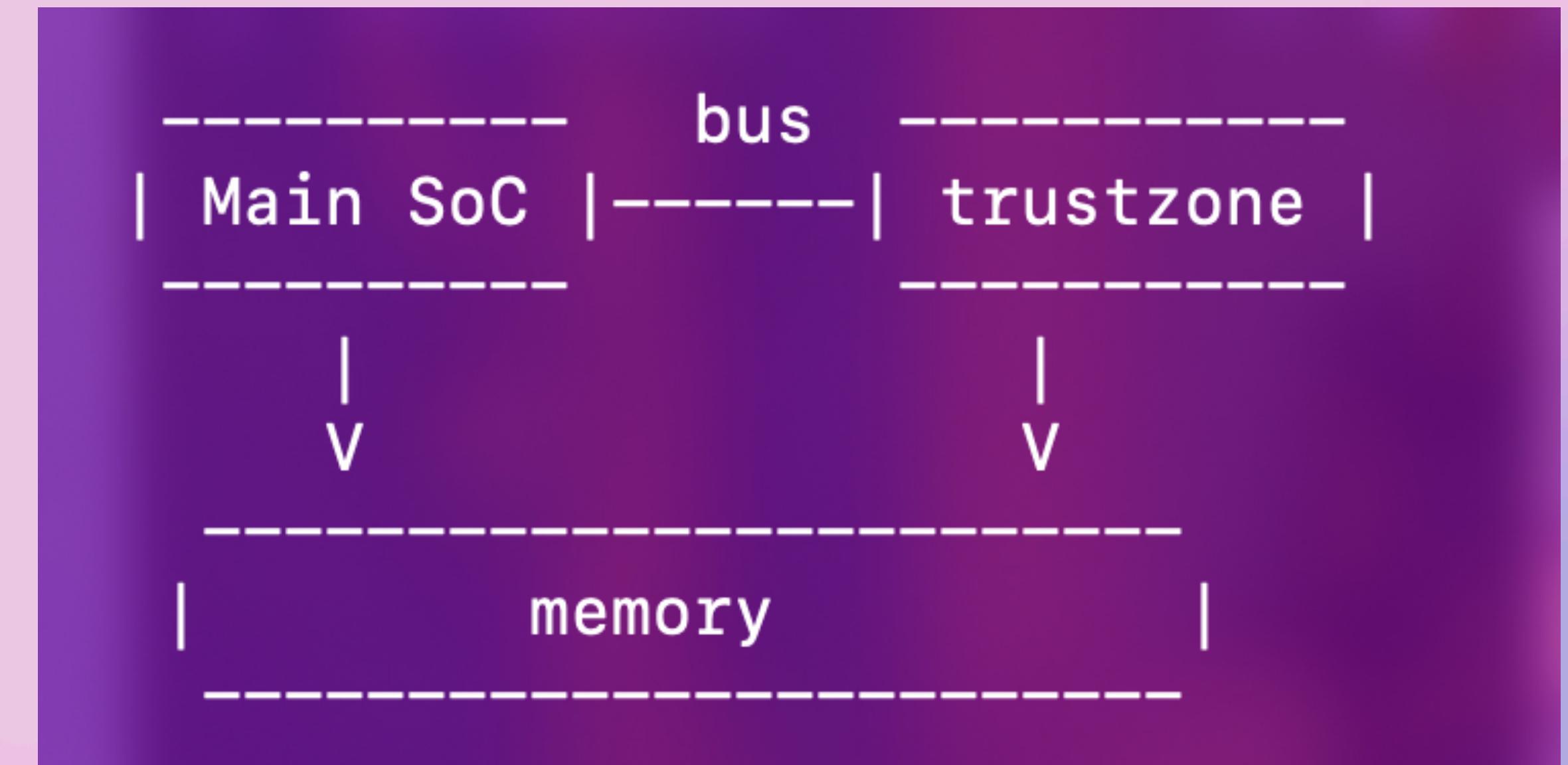
hitting a crypto wall

- Em algum trecho do código de boot, é feita uma verificação usando um módulo de criptografia (i.e. processador criptográfico). Essa verificação confere a assinatura da imagem android que o aparelho vai executar.

```
/* Verify image header */
ret = bootimg_hdr_verify(k_buff_img, boot_src);
if (ret) {
    lgpl_printf("ERROR: Boot image verify header failed!ret=0x%x\n", ret);
    return -1;
}
ret = bcm_image_verify(bcm_img_type, (unsigned) k_buff, (unsigned) k_buff);
if (ret) {
    lgpl_printf("ERROR: Verify k_buff image failed!ret=0x%x\n", ret);
    return -1;
}
```

Hitting a crypto wall

- A função que executa essa verificação mapeia endereços de memória em comum entre os dois processadores (o SoC onde o sistema operacional executa e o trustzone, onde são realizadas operações criptográficas.



Hitting a crypto wall

- O processador executa um comando no coprocessador e espera por um hardware interrupt. Se o retorno do interrupt for zero, o boot prossegue.
- Infelizmente, não consegui descobrir nenhuma forma de bypassar isso :(

```
...
mb->primitive_command_parameter0 = type;
mb->primitive_command_parameter1 = src;
mb->primitive_command_parameter2 = dst;
mb->secure_processor_command = BCM_PI_IMAGE_VERIFY;

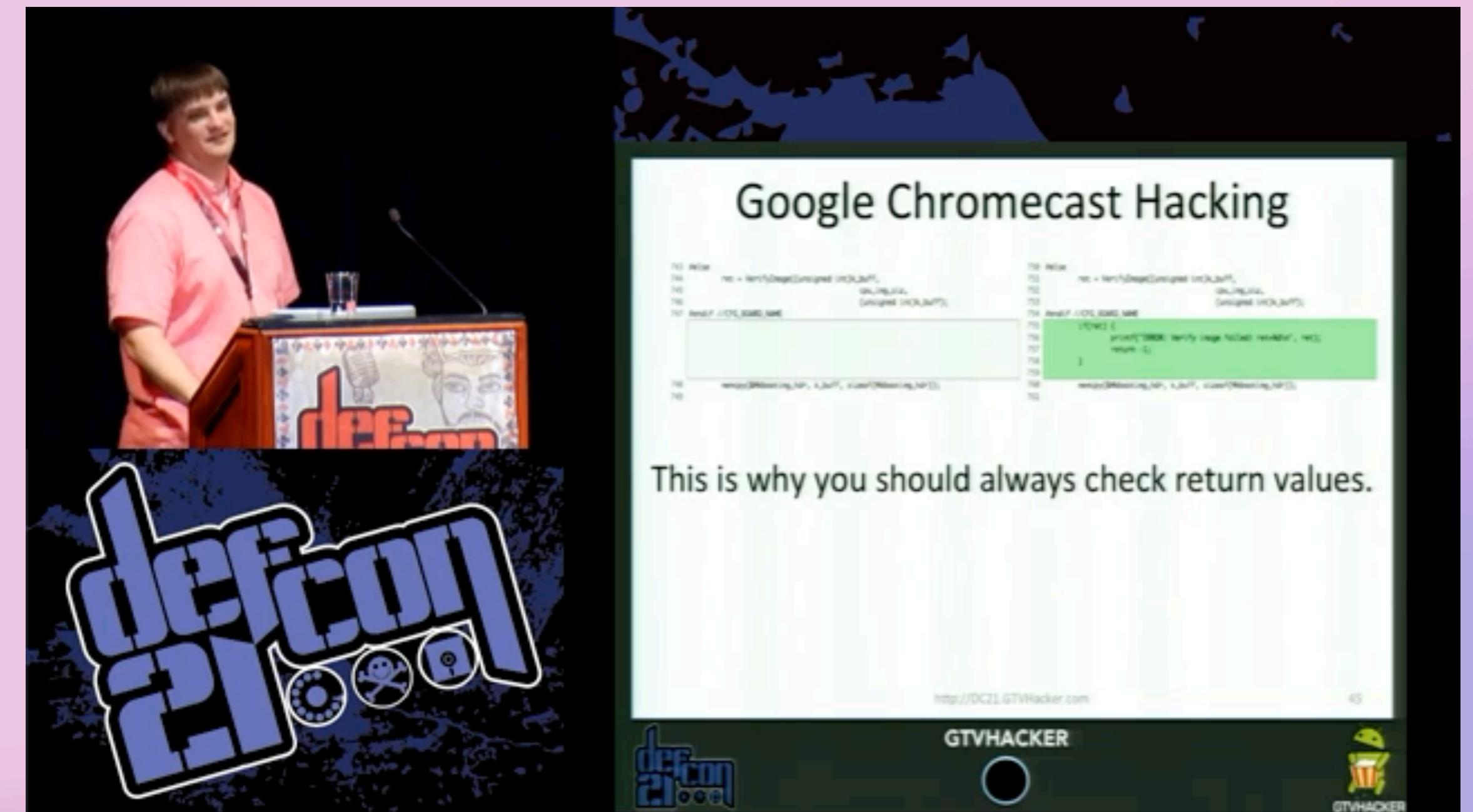
Depois, existe um loop que espera até um hardware interrupt acontecer:
for (waitForCount=0; ; waitForCount++) // Wait_For_WTM_Complete( 0x10000, pCtrl );
{
    //if ((mb->command_fifo_status & BCM_STATUS_BCM_CMD_FLIP) != status)
        //break;
    // wait for "command complete" or timeout
    if( mb-> host_interrupt_register & 0x1 )
        break;
    berlin_delay(100);
}

Por vim, a gente retorna o status
status = mb->command_return_status;

Lá na função load_android_image, é verificado se o retorno disso é zero, e se for, prossegue pro boot.
if (ret) {
    lgpl_printf("ERROR: Verify k_buff image failed!ret=0x%x\n", ret);
    return -1;
}
```

Previously...

- O pessoal do exploitee.rs tinha conseguido fazer um exploit no Chromecast V1 explorando o simples fato de terem esquecido de verificar o endereço de retorno da verificação criptográfica.



Trabalhos futuros

- A ideia de compartilhar um projeto incompleto de hacking é a de inspirar outros pesquisadores a continuar o projeto.
- A google continua lançando devices baseados numa arquitetura similar, então projetos novos podem ser explorados.
- Além disso, alguns projetos parecidos avançaram mais e introduziram técnicas muito incríveis, como o NANDbug do Courk.





**push %rdi
mov %rbp, %rsp
pop %rbp
ret**

computer love

- esse estudo não teve motivações acadêmicas, profissionais ou monetárias. Eu simplesmente queria saber muito o que uma caixa opaca como um Google Home brickado poderia ter.
- talvez a minha única motivação tenha sido o amor por computadores

KRAFTWERK



COMPUTER LOVE

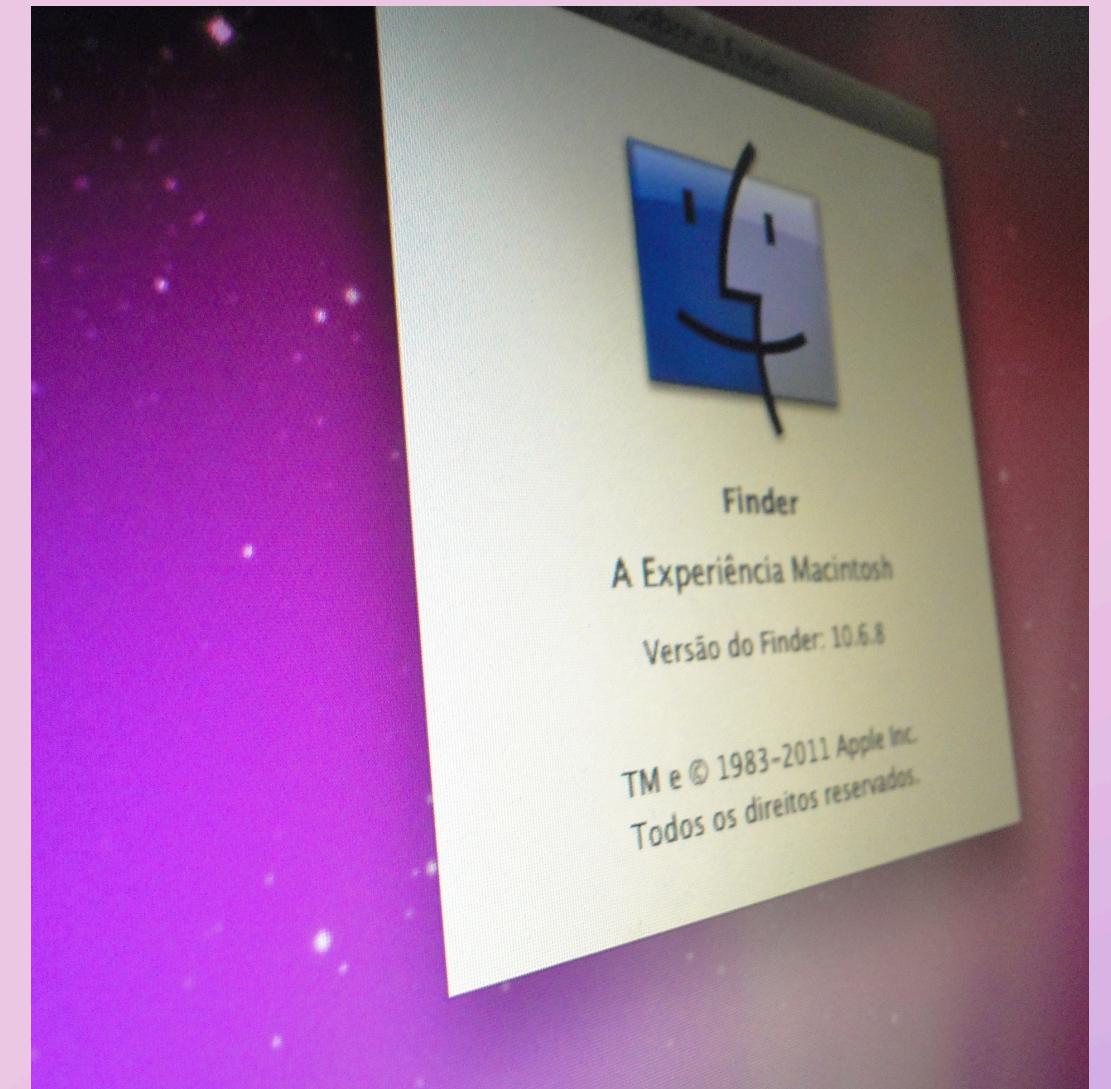
só nos compiuter

- talvez por conta do autismo e acesso a computadores cedo, eu desenvolvi um apreço à computação.
- aprendi a programar aos 13, fiz ETEC, fiz faculdade na área e tenho 9 anos de experiência de mercado.



hackintoshes e versões beta do Windows

- **meus hiperfocos sempre tinham motivações muito bobas. Algumas eram ver versões beta do Windows rodando numa máquina virtual (que aprendi sobre aos 8 anos de idade), outras eram rodar versões do Mac OS X em computadores Intel comuns.**



a computação ainda pode ser divertida

- eu sempre tento imaginar o quanto divertido deveria ser trabalhar na Bell Labs na época do UNIX, ou na Apple na época do Macintosh.
- boa parte dos engenheiros que desenvolveram as bases da computação moderna fizeram isso por diversão ou por necessidade.



Ken Thompson e Dennis Ritchie



Pushing the Limits of Technology: The Ken Thompson and Dennis Ritchie Story (National Inventors Hall of Fame)

technophobia

- todos temos boletos pra pagar, a vida pode ser dura, complicada e difícil. Só por trabalharmos com computação, já somos de certa forma privilegiados. Ainda mais se a gente gosta da área.
- porém, o mercado ultimamente tem sido estressante. Layoffs tem acontecido, IAs tem sido empurradas goela abaixo em todos nós, algumas pessoas já estão se aproveitando da precarização do nosso trabalho por vibe coding.
- perdi a esperança de continuar trabalhando nessa área muitas vezes...



computerlove

- mas a computação continua divertida. Eu continuo voltando nas coisas que gosto de fazer com computadores e me lembrando de que elas me dão prazer.
- o estudo do chromecast não chegou a terminar, mas saber que minhas técnicas vão acender luzes na cabeça de outras pessoas me faz feliz.



- lembrem-se: sejam pessoas curiosas. Queiram aprender mais, não só para suas carreiras, mas porque o mundo é cheio de fenômenos interessantes. Perguntem, procurem respostas, não se contentem com “porque sim”. Não perguntam as coisas pra IAs. Não esperem respostas mastigadas. O caminho para as respostas é coberto de conhecimento.

eof

linkedin.com/in/retpolanne
github.com/retpolanne
blog.retpolanne.com