Challenging real-world targets: from iPhone to insulin pump

Dr Sergei Skorobogatov

http://www.cl.cam.ac.uk/~sps32 *email: sps32@cam.ac.uk*

UNIVERSITY OF CAMBRIDGE
Computer Laboratory

Outline

- Introduction
- How to find the hot topic for research?
  - iPhone, car keys and insulin pump: what is common?
- Background
  - Apple iPhone 5C: https://arxiv.org/abs/1609.04327
  - Audi Smart Key: work-in-progress
  - Insulet OmniPod: https://arxiv.org/abs/1709.06026
- iPhone 5C
- Smart Key
- OmniPod
- Limitations and Improvements
- Future work and Collaboration
- Conclusion
- The slides are available online: http://www.cl.cam.ac.uk/~sps32

Introduction

- Hardware Security research since 1995
  - testing microcontrollers and smartcards for security
  - research on semi-invasive attacks (PhD, 2005, Cambridge UK)
  - backdoors in semiconductors (2012)
  - iPhone 5C NAND mirroring (2016)
  - “impossible” solutions for challenges in real-world devices
- Hardware Security is about finding flaws and fixing them
  - preventing attacks on service, data and IP theft
  - what security features are implemented and how secure they are
- Hardware Security challenges
  - data, keys and passwords storage
  - new attack technologies
  - modern fabrication processes (10nm, 14nm, 28nm, 45nm, 65nm)
  - developing countermeasures through understanding of flaws
  - prediction of new attack methods

How to find the hot topic for research?

- iPhone 5C
  - FBI recovered an Apple iPhone 5C used by one of the shooters involved in the December 2015 San Bernardino attack
  - On February 9, 2016, the FBI announced that it was unable to unlock the phone due to its advanced security features, including encryption of user data
  - FBI asked Apple Inc. to create a new version of the phone's iOS operating system that could be installed to disable certain security features. Apple declined due to its policy to never undermine the security features of its products. Legal fight starts…
  - FBI Director James Comey at a press conference on 24 March 2016: "We tried everything we could think of, asked everybody we thought could help, inside and outside the government, before bringing the litigation." "The notion that we didn’t exhaust all alternatives is silly."
  - "I heard that a lot. It doesn’t work," Comey said in response to a reporter’s question about the NAND mirroring technique.
  - On March 28, the FBI said it had unlocked the iPhone with the third party's help
- Challenges
  - look at the problem and find the solution
  - make something “impossible” to attract attention to the research

How to find the hot topic for research?

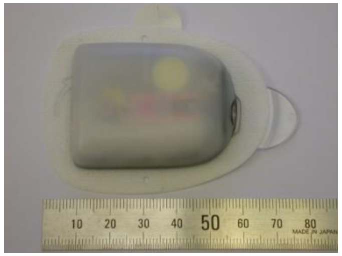
- Audi Smart Key
  - recently bought a used Audi car from Audi dealership which is supposed to be more trustworthy than independent private garage
  - the car was supplied with only 2 keys and the salesman claimed it had only that
  - it is the part of 145-point check that Audi is proud of with their Approved Used cars as they claim “you have the reassurance of knowing that all cars have to meet our meticulous standards”… “145 Exacting checks to pass”; In the paperwork position “17. Check & record all ignition keys coded to the vehicle are present" is ticked
  - accidentally found in car menu: Car systems>Servicing&checks>Initialised keys: 3
  - response from Audi
    - “The spare/3rd key is a valet key, like the one attached, and is indeed programmed to the car. You are unable to start the car with this key, you can only open the driver’s door with it”
    - “…getting your 3rd wallet key coded…”

- Challenges
  - read the User Manual
  - find out what is inside Audi Smart Key
  - investigate the problem
  - evaluate the hardware security of the car key

How to find the hot topic for research?

- Insulet OmniPod wireless tubeless insulin pump
  - improves the life of patients with Type 1 diabetes – no more tubes hanging around
  - the only tubeless insulin pump approved by FDA (important for USA patients)
  - requires manual monitoring of patient’s blood sugar level
  - the controller (PDM) securely communicates with the pump (Pod)
  - total lack of information about their devices from the manufacturer Insulet Corp.
  - wide community of people trying to build an artificial pancreas
    - Loop project: http://loopdocs.org
    - OpenAPS: https://openaps.org/
    - Nightscout Foundation: http://www.nightscoutfoundation.org/
  - so far succeeded with only conventional pump from Medtronic and only old version

- Challenges
  - find the solution to help people
  - find out what is inside the medical devices
  - evaluate the security of custom medical device to help with improvements



What is common between those projects?

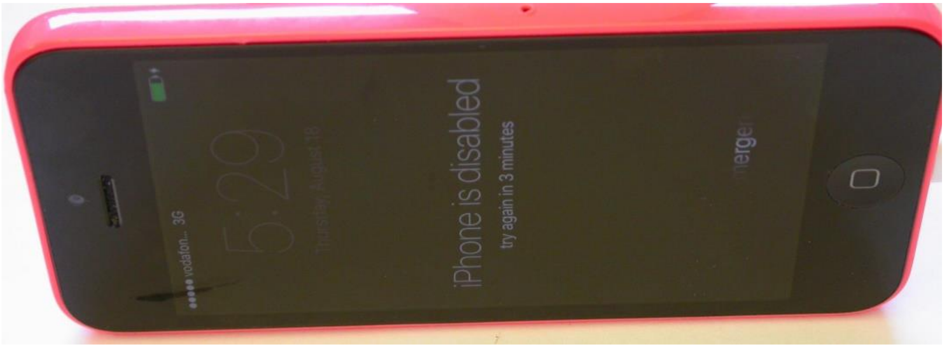
- Security claims
  - virtually unbreakable device
  - strong encryption
  - secure communication
- Challenges
  - verify the claims
  - evaluate the security
  - find the ways for improvements
- Impact
  - expose vulnerabilities
  - bring awareness
  - help communities of people
- Attract attention
  - funding for existing research
  - attract sponsors and collaborators for new research

iPhone 5C: NAND mirroring

- iPhone forensics expert Jonathan Zdziarski previously suggested the FBI could use NAND mirroring to get information off the locked iPhone
- FBI Director James Comey claimed that making a copy of the phone’s chip to get around the passcode "doesn’t work" and the solution would be "software-based."
- Zdziarski responded by cooking up a NAND mirroring proof-of-concept to prove that "copying the back disk content could allow for unlimited passcode attempts." However, he used a jailbreaked phone, and no one has yet demonstrated the actual NAND mirroring attack
- If he is correct and Comey is not, it wouldn’t be the first time the government lied to us. The Justice Department swore it was “impossible” without having Apple to create a backdoor
- FBI Director Comey was not pleased about the WSJ piece and fired back: "You are simply wrong to assert that the FBI and the Justice Department lied about our ability to access the San Bernardino killer’s phone."

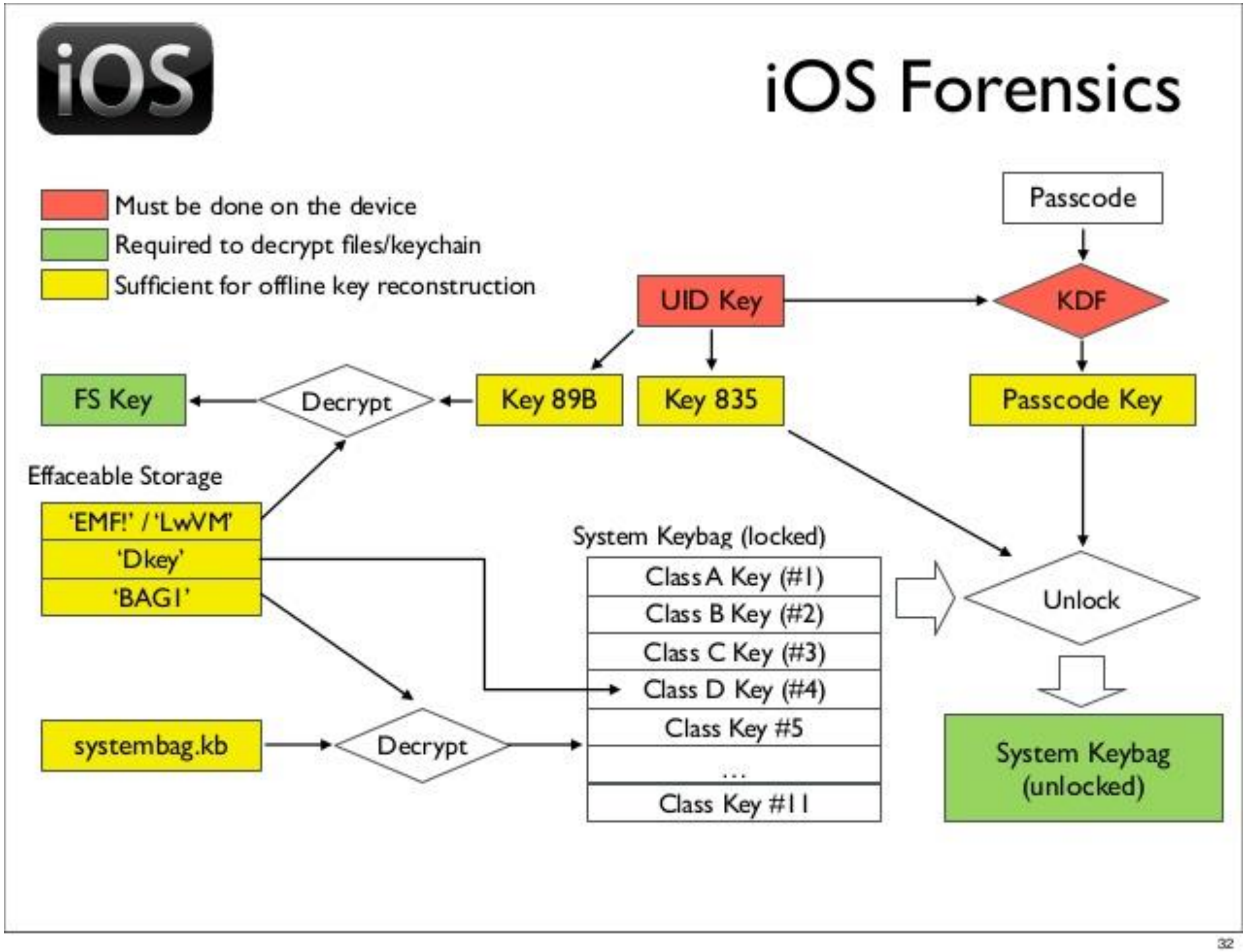
What is known

- iPhone 5C allows 5 passcode attempts without delay
- 6th attempt is delayed by 5 seconds
- 7th attempt is delayed by 1 minute
- 8th attempt is delayed by 5 minutes
- 9th attempt is delayed by 15 minutes
- 10th attempt is delayed by 60 minutes
- Permanent lock could be activated after 10 attempts



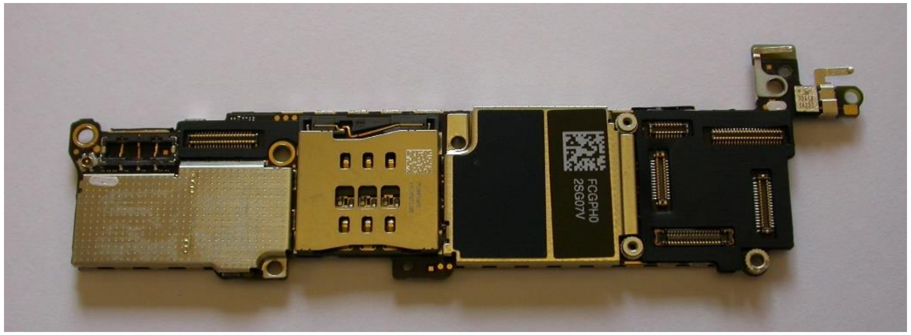
Introduction

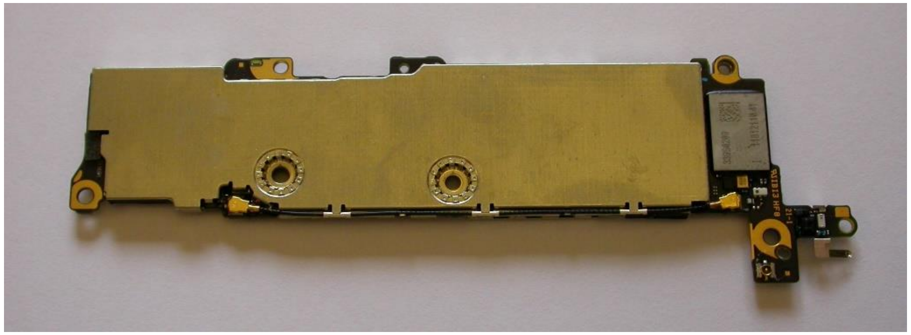
- Apple iPhone 5C security
  - claimed to have hardware that is infeasible to break

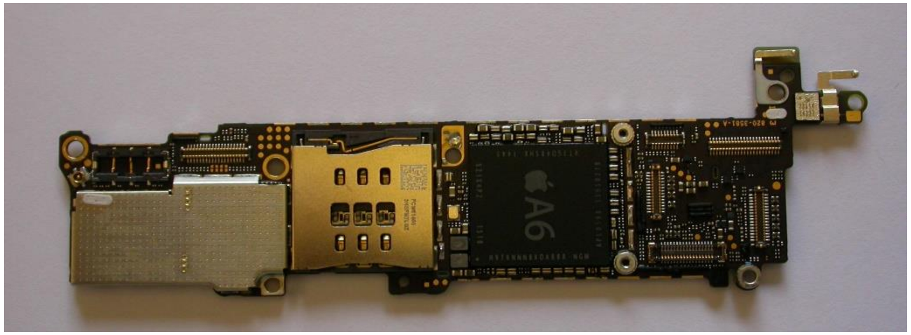


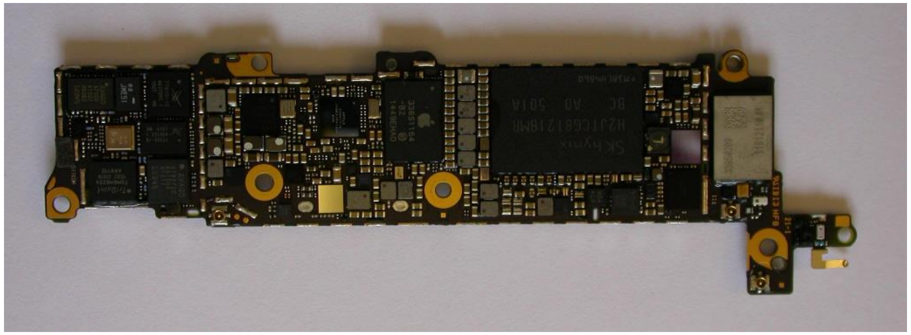
Taking iPhone 5C apart

- Used iPhone 5C cost from £60 on Ebay
- Teardown process is well described by repair geeks
- Shieldings were desoldered with hot air gun



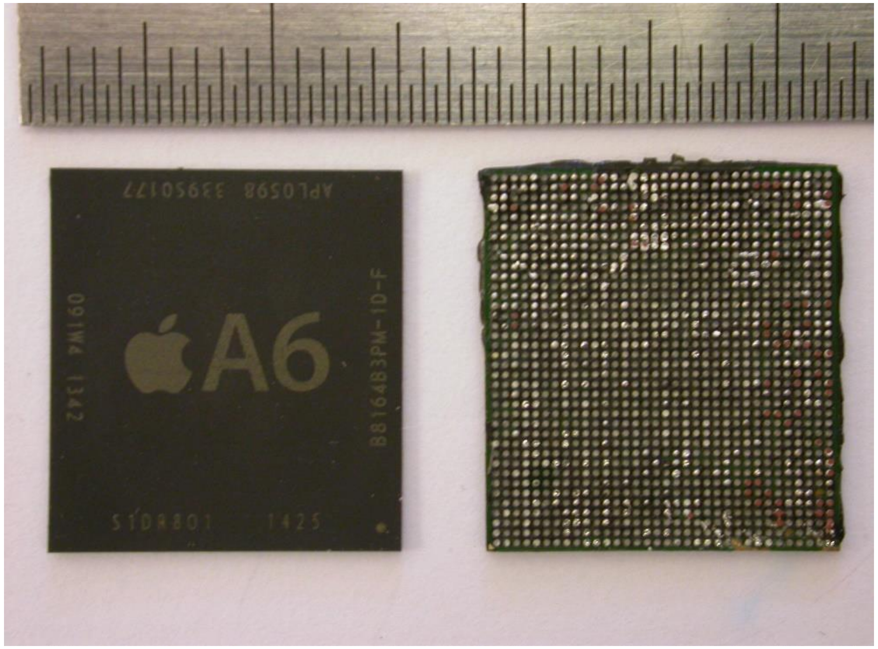


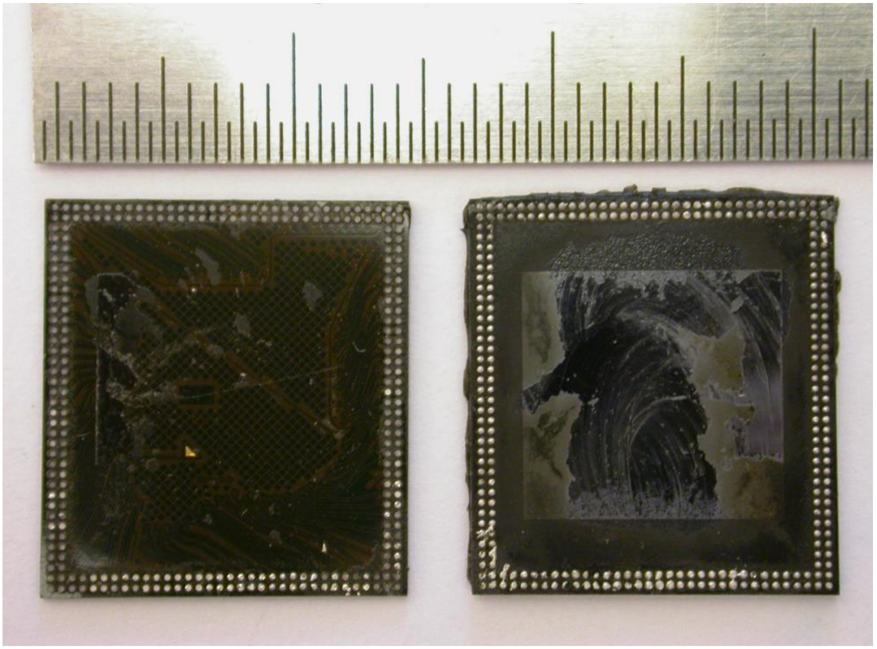




Taking A6 CPU apart

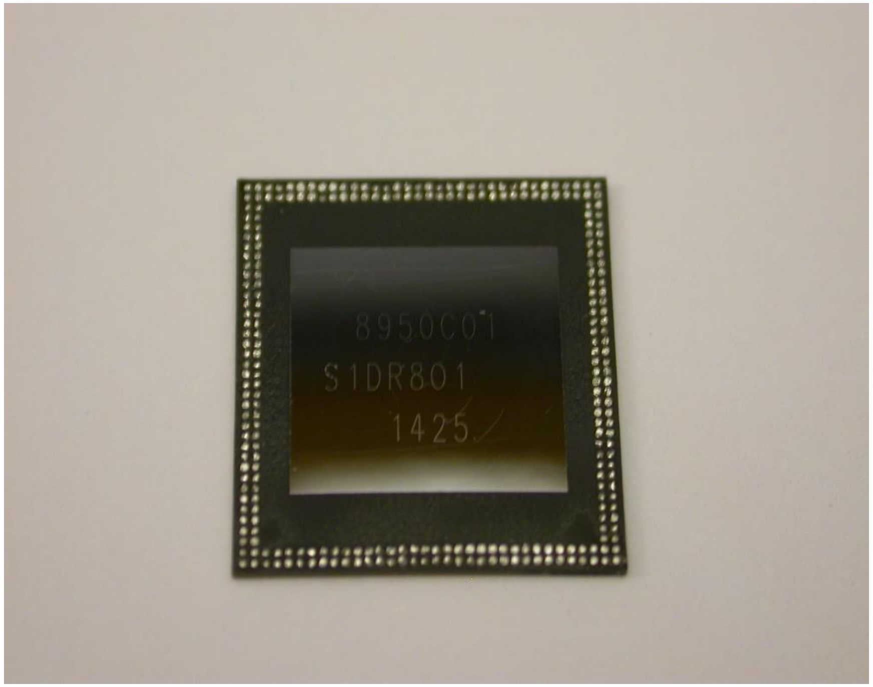
- CPU and NAND Flash chips were glued to PCB
  - just raised the temperature and pulled with tweezers
  - A6 in BGA package with 1292 balls at 0.4mm pitch
  - A6 heated further to take it apart into SoC and SDRAM

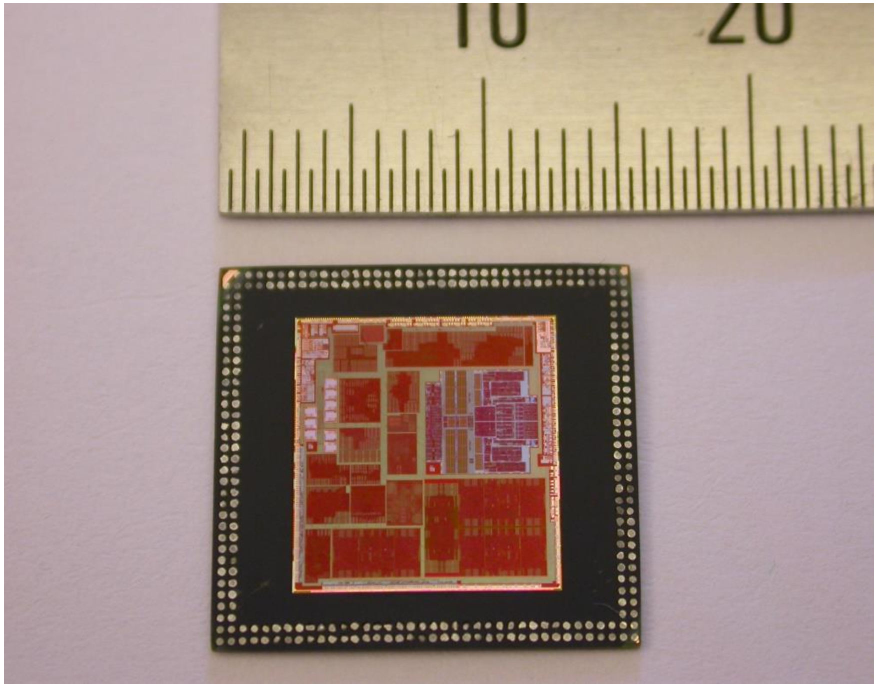




Taking A6 CPU apart

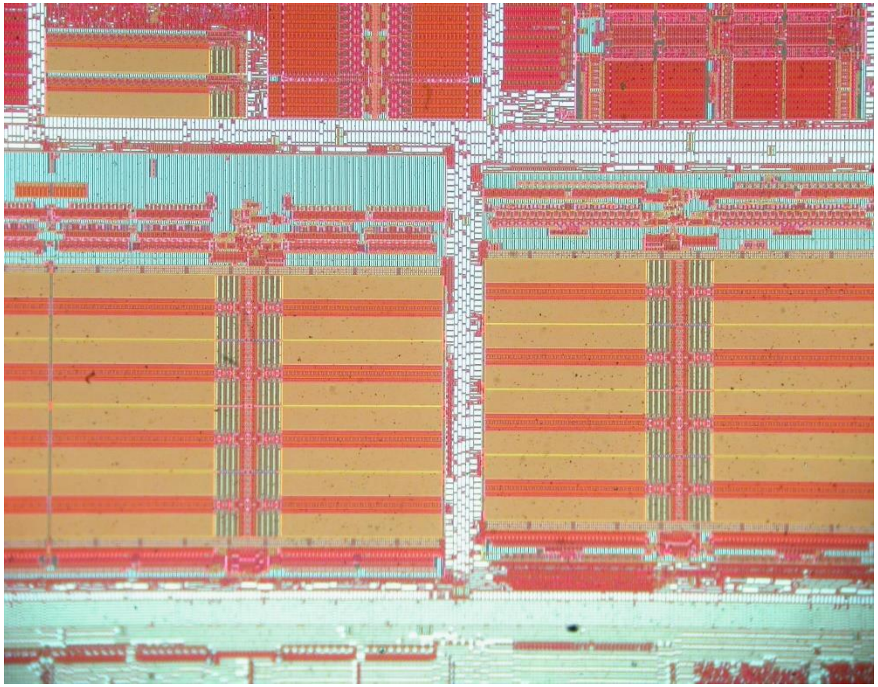
- SoC part of the CPU cleaned, polished and deprocessed
  - photographed with low resolution for outlook image

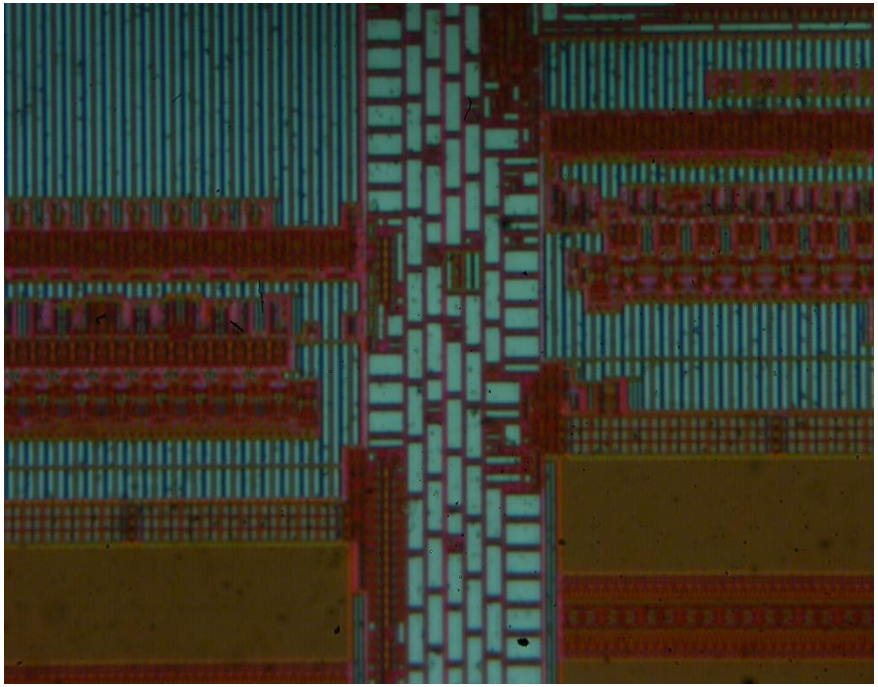




Search for UID

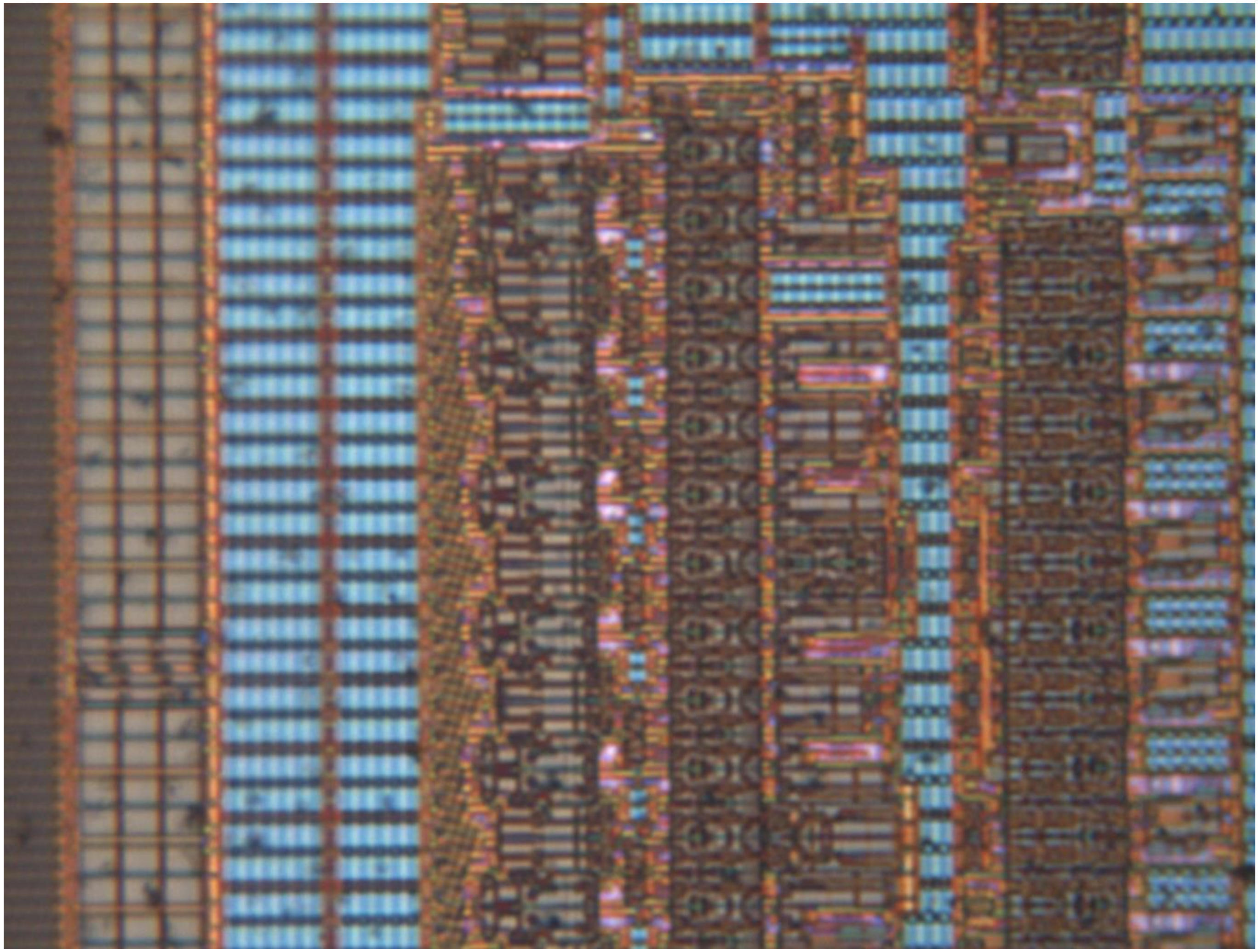
- But A6 SoC is built with 32nm process
  - several generations have passed from 90nm: 65, 55, 45, 40nm
  - some odd findings in ARM area





Search for UID within ARM CPU cores

- Closer look at suspected areas with 160x/1.25W HA

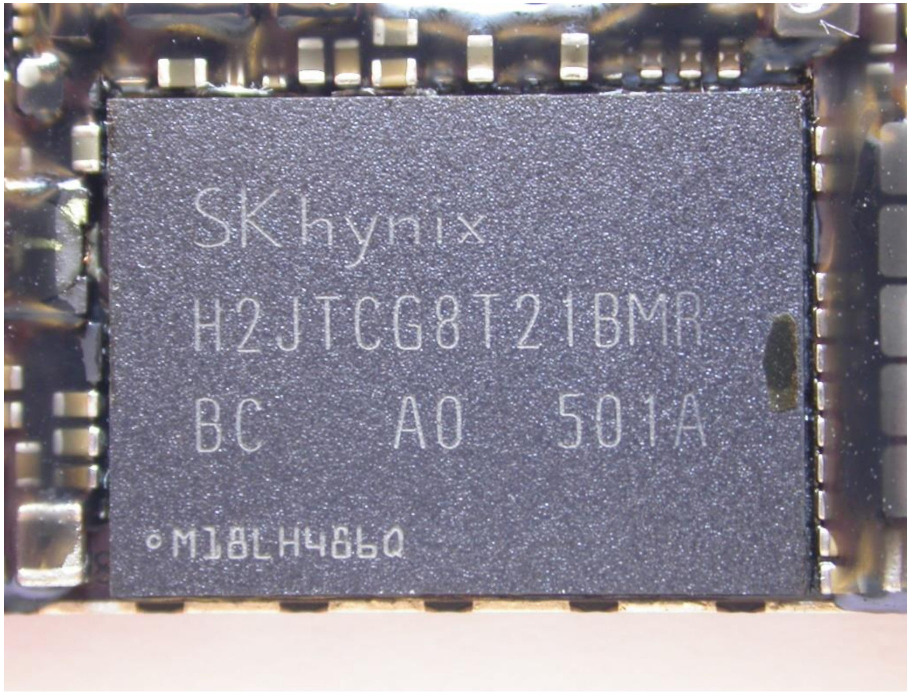


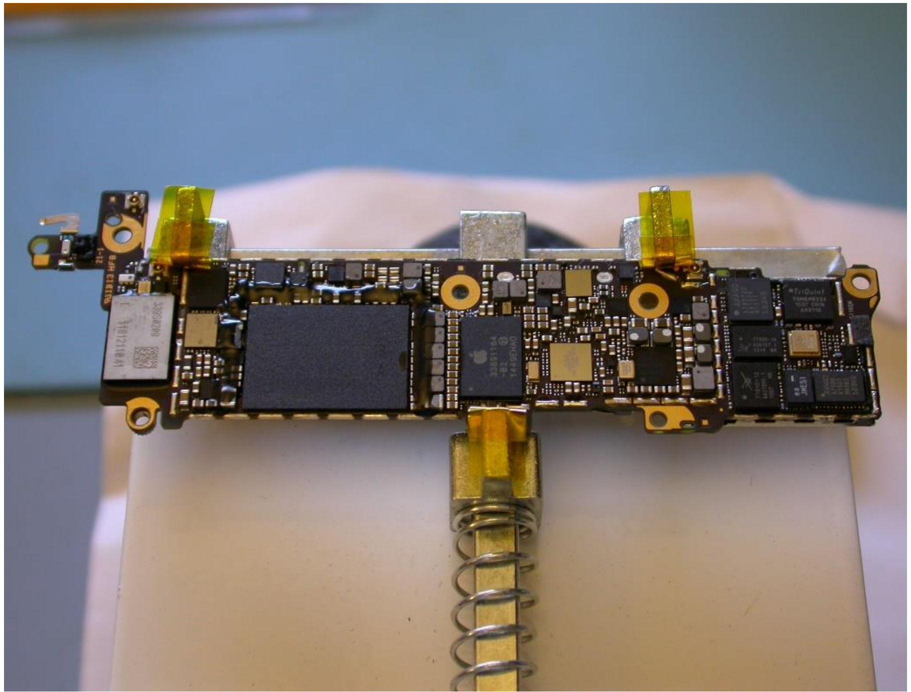
Limitations and improvements

- Best optical microscopes employ immersion lenses, near-field optics and confocal imaging, still at best 100nm
- We will be better off with SEM – modern models resolve 0.5nm resolution
- We don't know eFuse structure
  - can be polisilicon fuse
  - can be M1 or M2 fuse
  - can be M1 to M2 via fuse
- Might be necessary to deprocess down to M2, M2-via-M1, M1 or polysilicon layer
- What about non-invasive attacks?
  - Apple claim there are no backdoors, but eFuses leak a lot for SCA
  - Have they blocked the programming access interface (writeback)?

Taking iPhone 5C apart

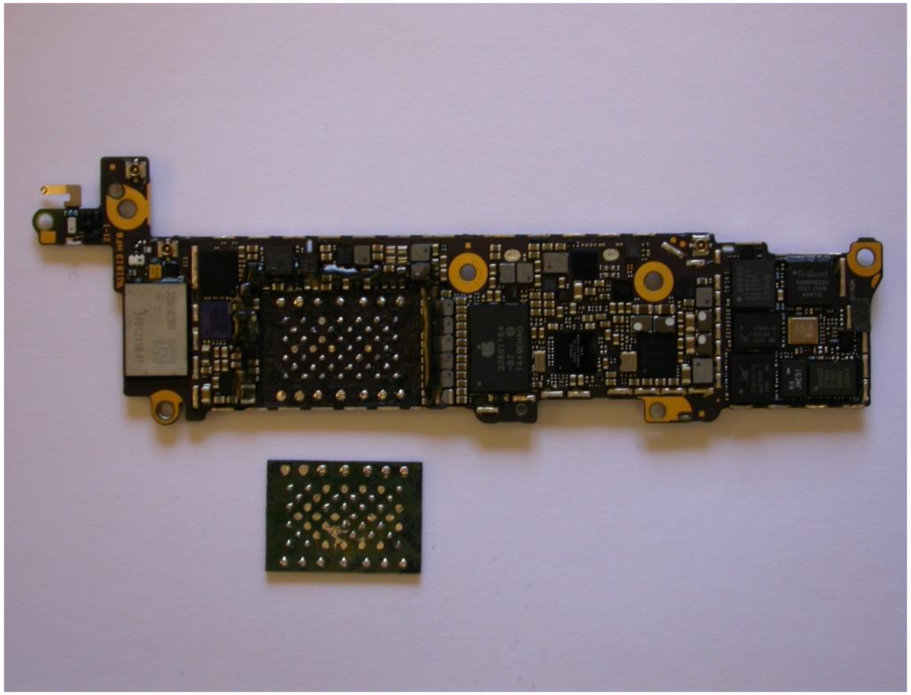
- Locating the NAND flash memory chip
- Desoldering it with hot air gun
- Quite a challenging process
  - epoxy glued LGA package
  - the gap between chip and PCB is only 0.05mm

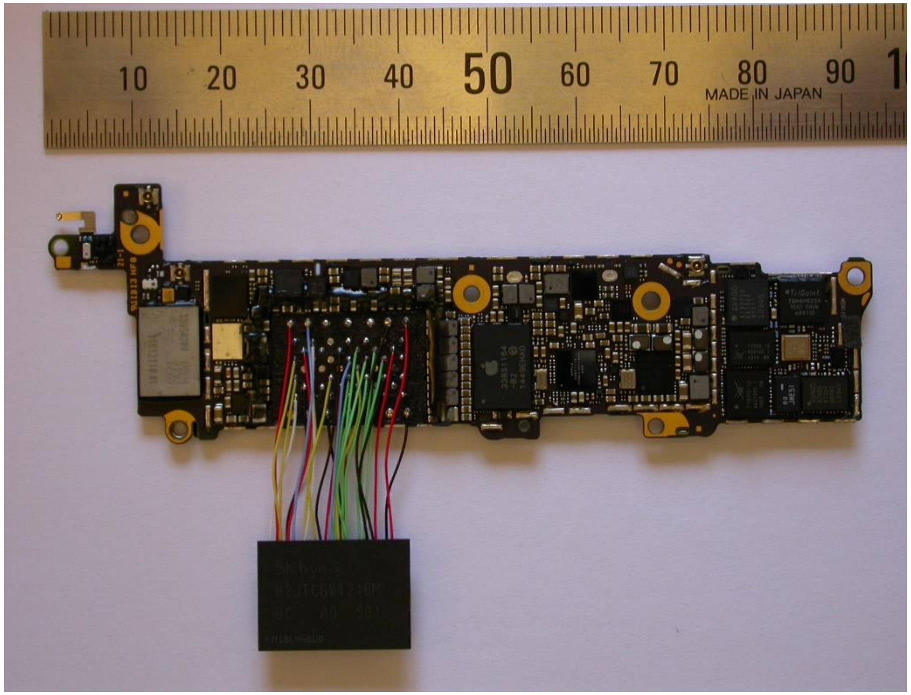




Taking iPhone 5C apart

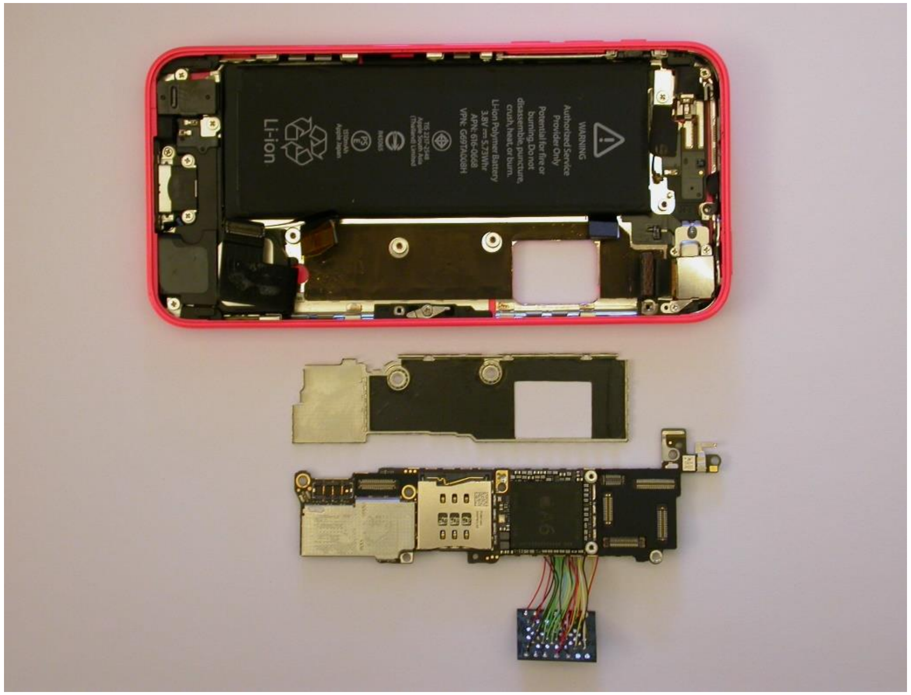
- Cleaning the pads
- Wiring the NAND flash memory chip to the main PCB

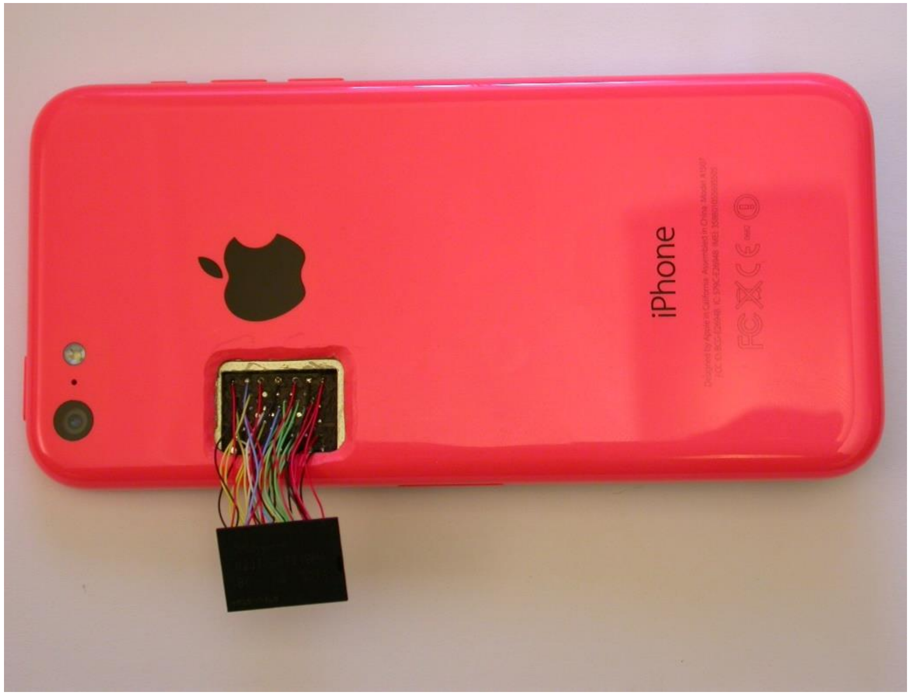




Taking iPhone 5C apart

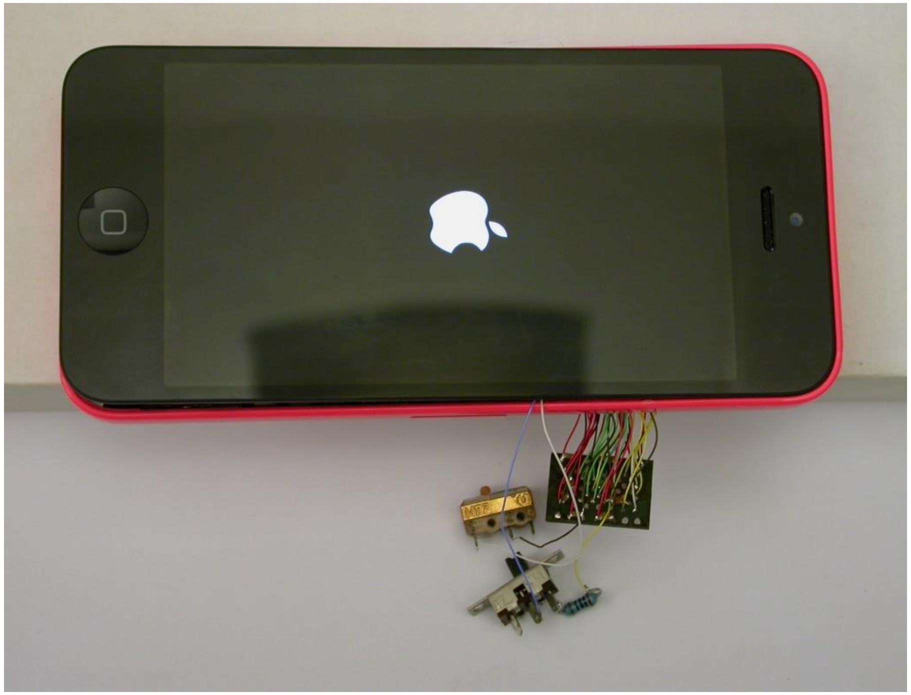
- Make a cut in the housing and assemble the iPhone
- Power up the iPhone... it shows the Logo and crashes
- iTunes cannot Restore the system due to verification error

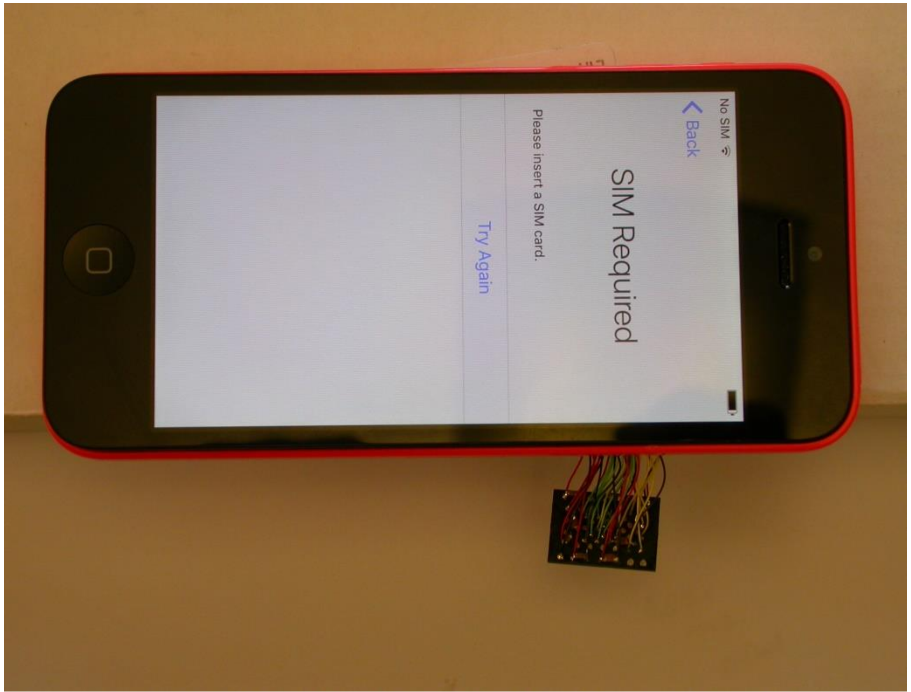




Taking iPhone 5C to work again

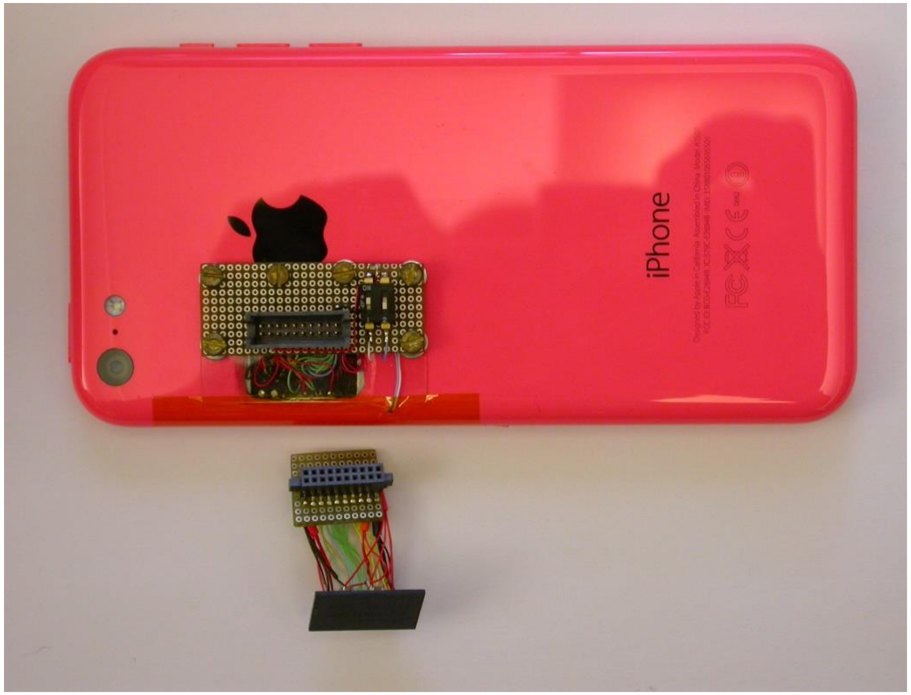
- Power supply wires do not pass sufficient current as it could peak 2A: sorted out by installing 10 bypass caps
- But the NAND was corrupted during multiple OS crashes and didn't boot into Recovery mode for iTunes
- Sorted out with two wires attached to main PCB: HRESET and FORCE_DFU (a bit tricky and requires high voltage to activate)

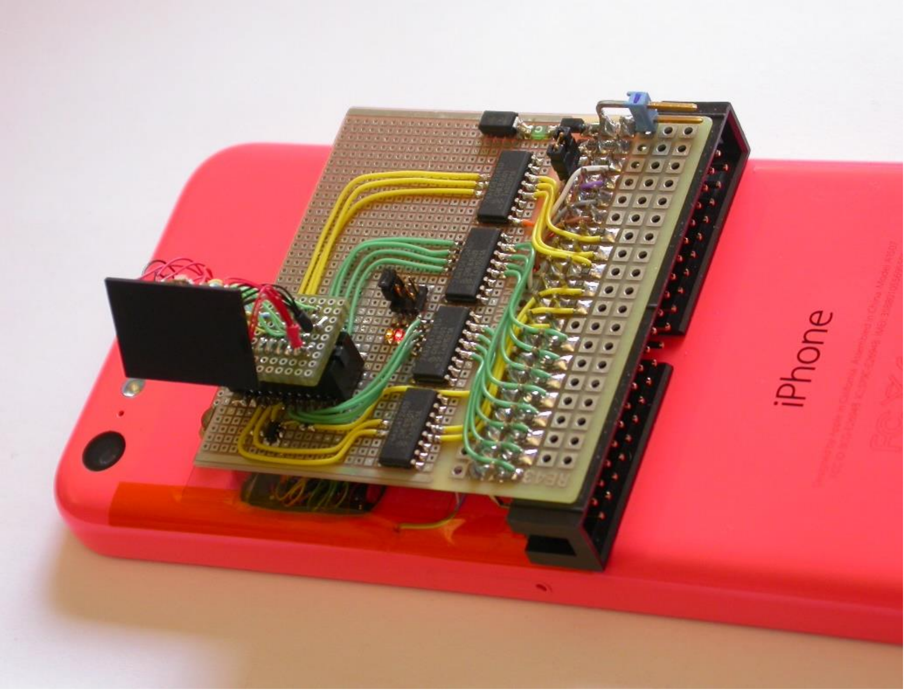




Eavesdropping on iPhone 5C NAND

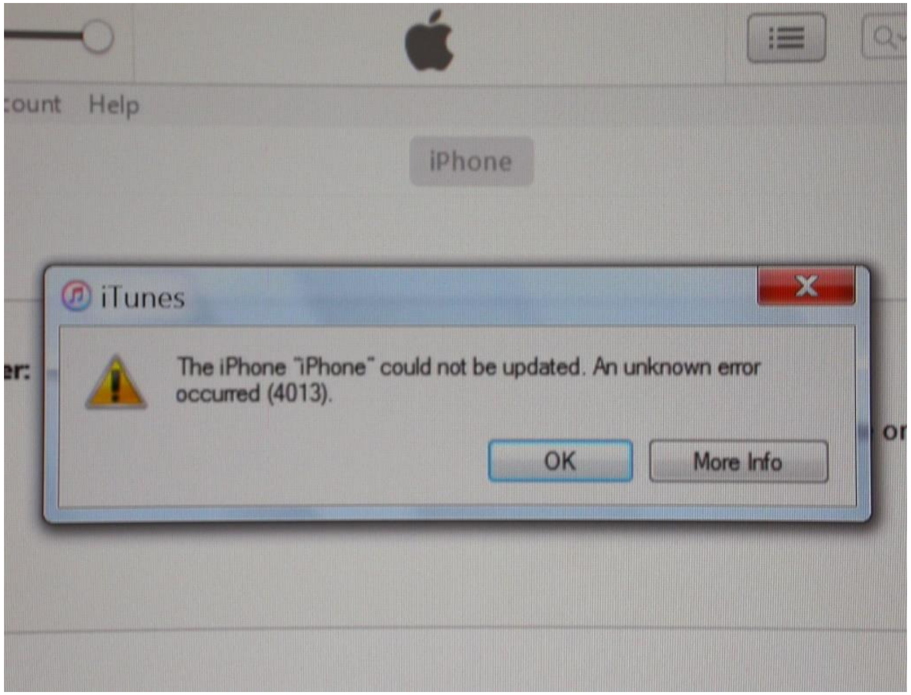
- Replace wires with sockets
- Build an intermediate PCB
- Power up the iPhone and ... it doesn't boot
- Removing of the intermediate PCB doesn't help





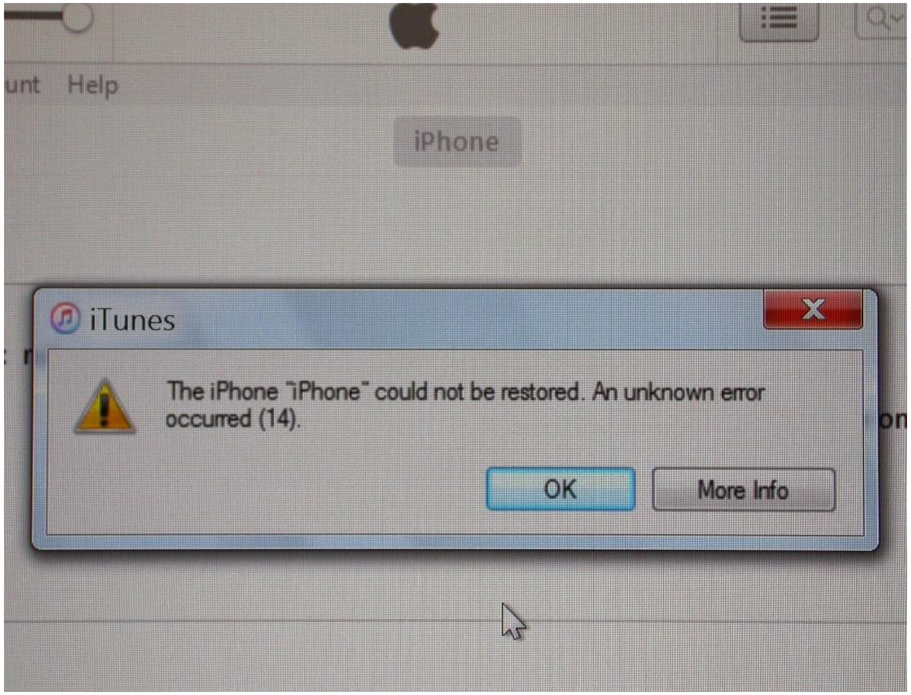
Eavesdropping on iPhone 5C NAND

- The NAND memory contents was seriously corrupted and iTunes was unable to restore it – boot server was crashing
- No solution was found on the Internet except suggestions for PurpleRestore tools which were outdated and didn't run



iTunes

The iPhone "iPhone" could not be updated. An unknown error occurred (4013).

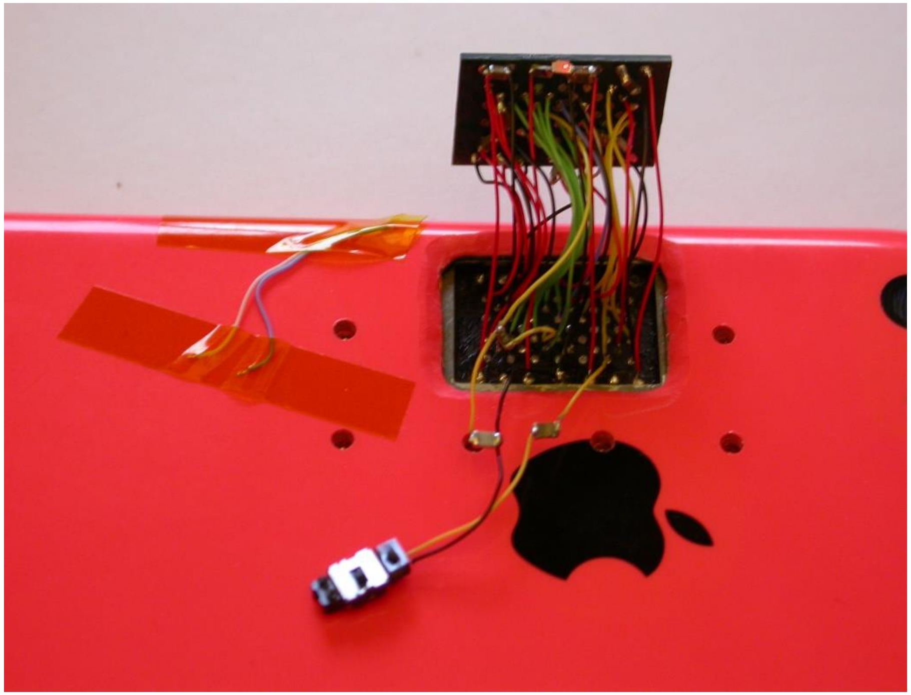


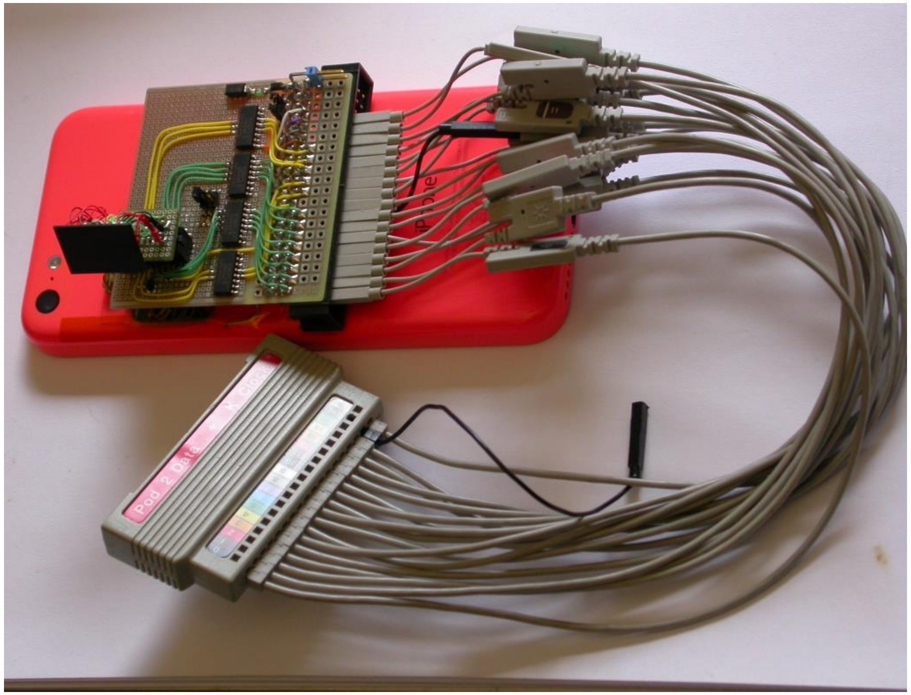
iTunes

The iPhone "iPhone" could not be restored. An unknown error occurred (14).

Eavesdropping on iPhone 5C NAND

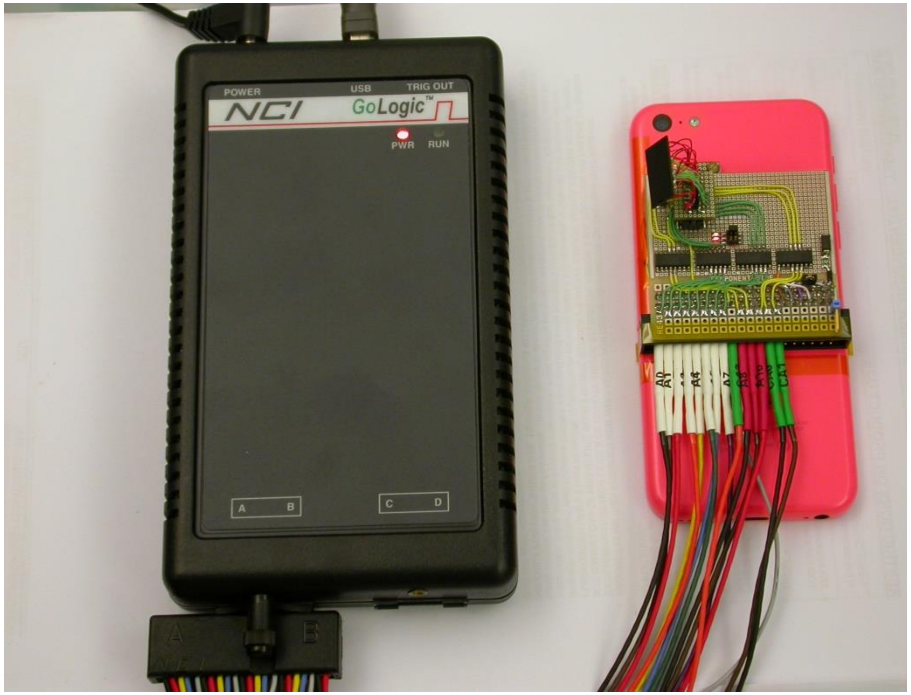
- The problem is Catch22 of iPhone ring of trust
- Itunes trusts iBoot, but verifies it
- iBoot trusts Update Server, but doesn't verify all its parts
- Need to crash the iBoot to force iTunes into deep Recovery

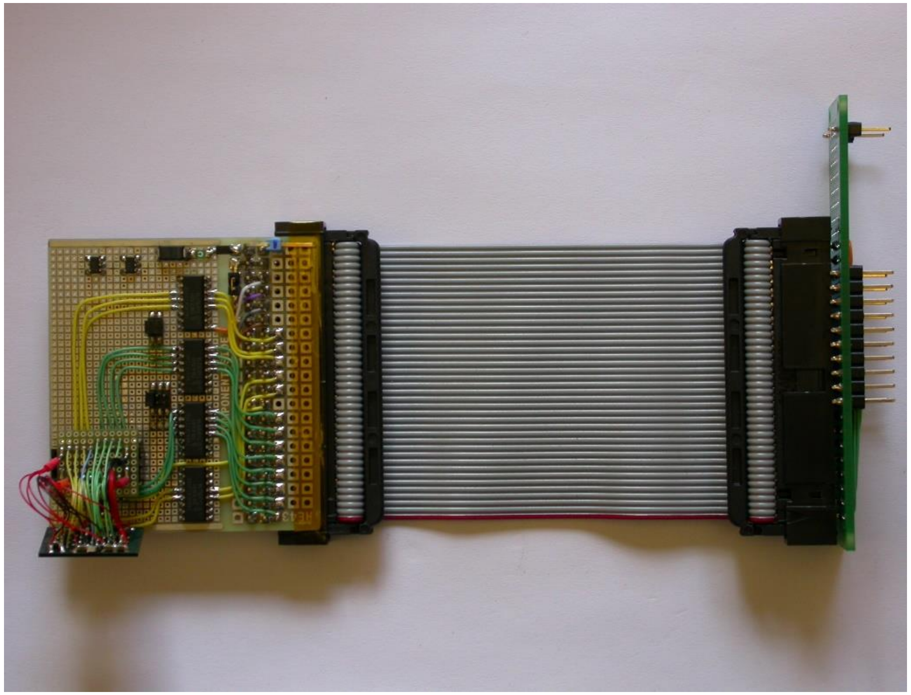




iPhone 5C NAND analysis

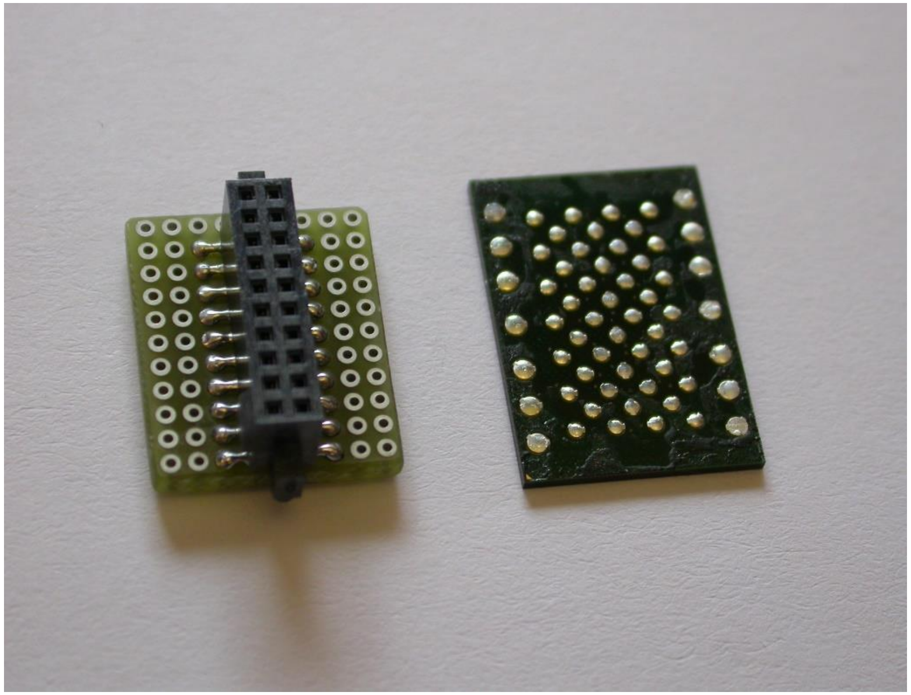
- Once the iPhone is fully operational we can use oscilloscope and logic analyser to understand the communication protocol and commands
- NAND chips in iPhone 5C use proprietary design and non-standard commands

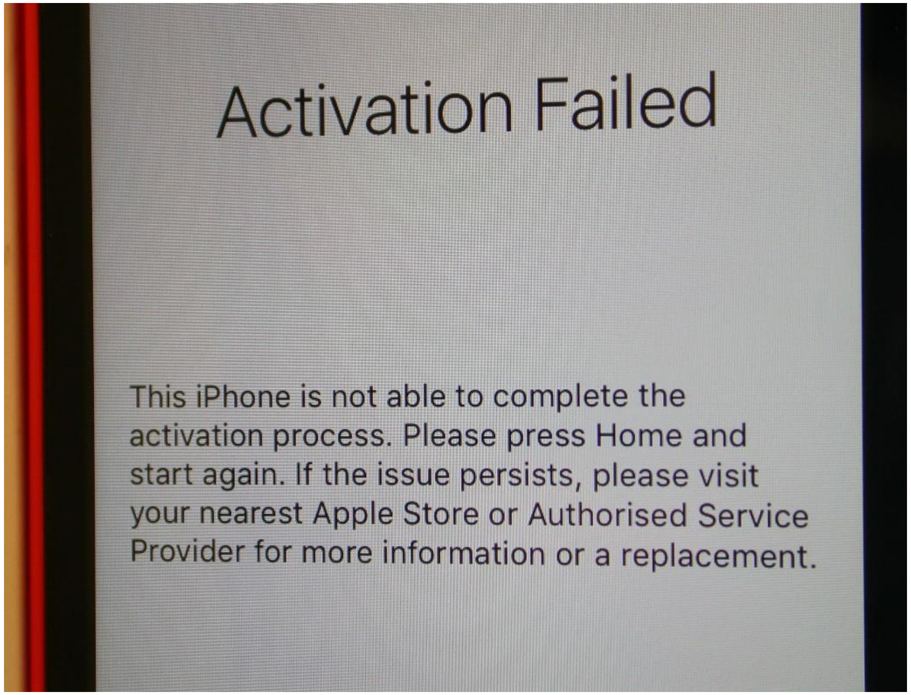




iPhone 5C NAND analysis

- Apple prevented NAND cloning
- NAND chips from other iPhones obviously will not operate
- They will also have unique serial numbers verified by OS
- Neither it would be possible to do recovery with iTunes



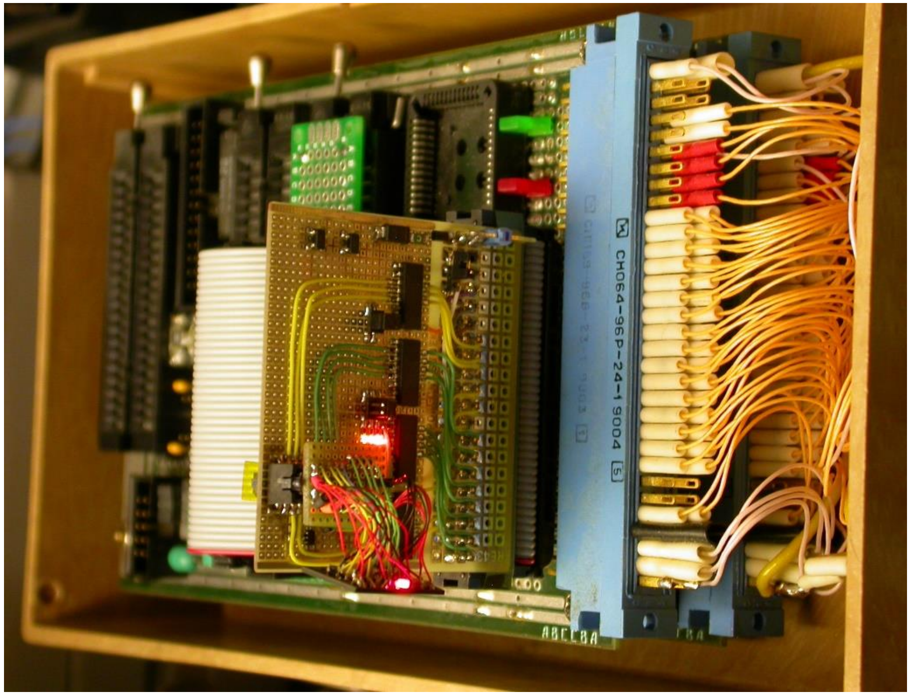


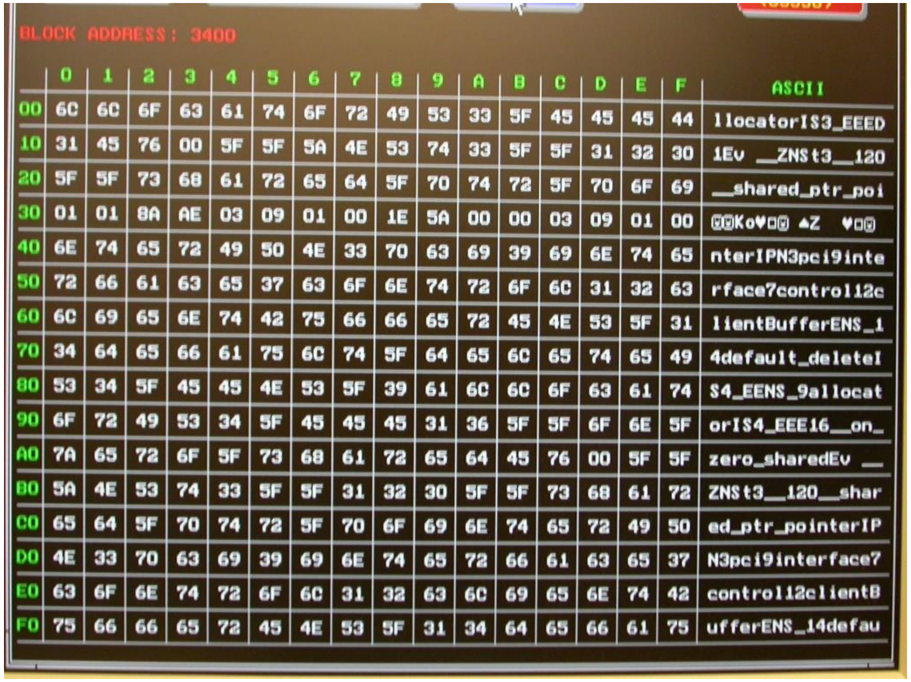
Activation Failed

This iPhone is not able to complete the activation process. Please press Home and start again. If the issue persists, please visit your nearest Apple Store or Authorised Service Provider for more information or a replacement.

iPhone 5C NAND analysis

- First the protocol was emulated in a slow Universal hardware programmer
- Some ASCII text can be seen
- NAND erasing in 4MB blocks of 256 pages, writing in pages
  - page is 16448 bytes: 4 sectors of 4096 bytes + 16 bytes of index
  - index locations are quite unusual: 400–40F in 0000–100F space



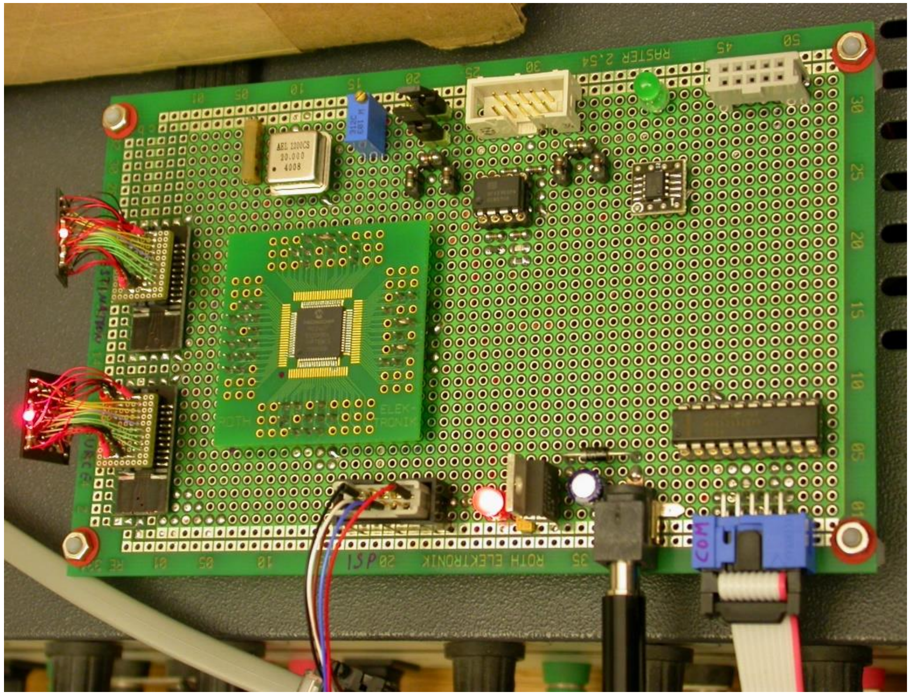


BLOCK ADDRESS: 3400

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	ASCII
00	6C	6C	6F	63	61	74	6F	72	49	53	33	5F	45	45	45	44	llocatorIS3_EEED
10	31	45	76	00	5F	5F	5A	4E	53	74	33	5F	5F	31	32	30	1Ev __ZNSt3__120
20	5F	5F	73	68	61	72	65	64	5F	70	74	72	5F	70	6F	69	__shared_ptr_poi
30	01	01	8A	AE	03	09	01	00	1E	5A	00	00	03	09	01	00	⊡⊡Ko♥□⊡ ▲Z ♥□⊡
40	6E	74	65	72	49	50	4E	33	70	63	69	39	69	6E	74	65	nterIPN3pci9inte
50	72	66	61	63	65	37	63	6F	6E	74	72	6F	6C	31	32	63	rface7control12c
60	6C	69	65	6E	74	42	75	66	66	65	72	45	4E	53	5F	31	lientBufferENS_1
70	34	64	65	66	61	75	6C	74	5F	64	65	6C	65	74	65	49	4default_deleteI
80	53	34	5F	45	45	4E	53	5F	39	61	6C	6C	6F	63	61	74	S4_EENS_9allocat
90	6F	72	49	53	34	5F	45	45	45	31	36	5F	5F	6F	6E	5F	orIS4_EEE16__on_
A0	7A	65	72	6F	5F	73	68	61	72	65	64	45	76	00	5F	5F	zero_sharedEv __
B0	5A	4E	53	74	33	5F	5F	31	32	30	5F	5F	73	68	61	72	ZNSt3__120__shar
C0	65	64	5F	70	74	72	5F	70	6F	69	6E	74	65	72	49	50	ed_ptr_pointerIP
D0	4E	33	70	63	69	39	69	6E	74	65	72	66	61	63	65	37	N3pci9interface7
E0	63	6F	6E	74	72	6F	6C	31	32	63	6C	69	65	6E	74	42	control12clientB
F0	75	66	66	65	72	45	4E	53	5F	31	34	64	65	66	61	75	ufferENS_14defau

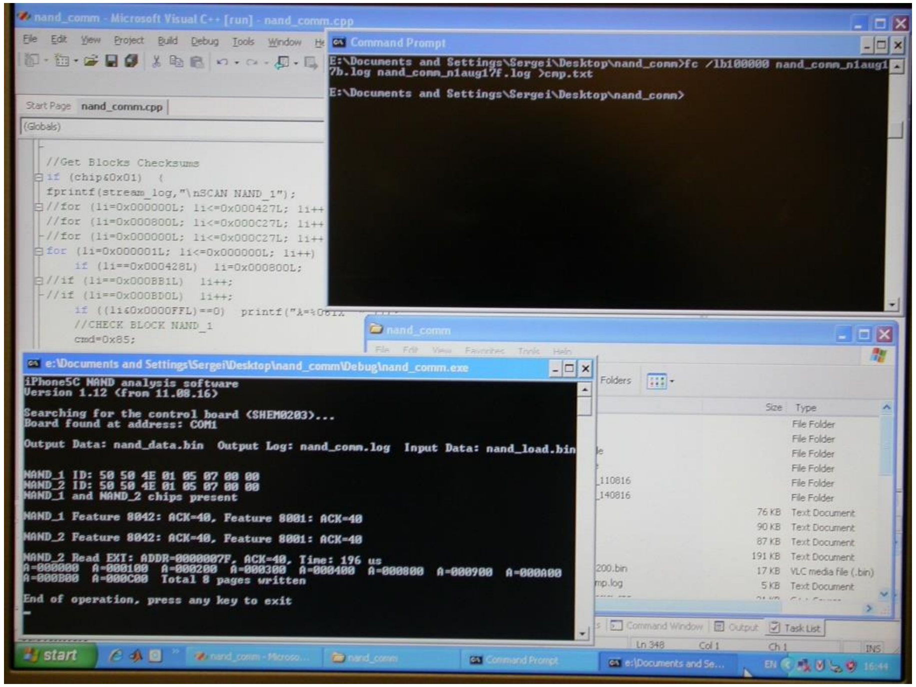
iPhone 5C NAND analysis

- Special hardware board was built to implement fast reading and writing of NAND
  - uses same type of NAND as an image store
  - communicates with NAND at 40MB/s vs iPhone 250MB/s speed
  - does not have fast PC communication, hence, no backups
  - sufficient for verifying the NAND mirroring concept



iPhone 5C NAND mirroring

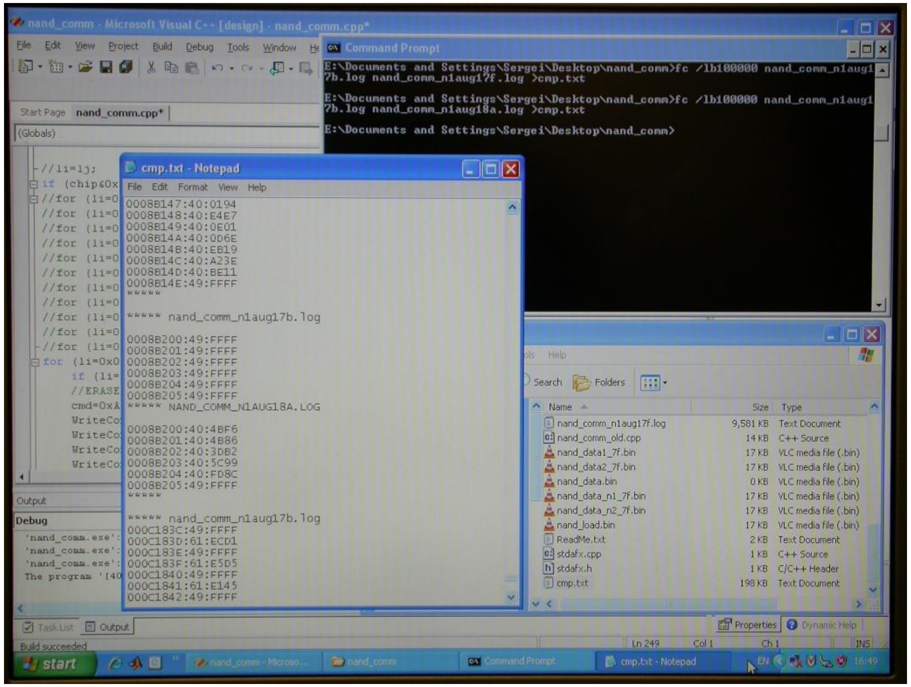
- Mirroring is the process of restoring the original contents of storage from backup copy
  - create backup copy
  - let the system to run, enter 5–6 passcodes then power down
  - scan the NAND chip and identify any changes


nand_comm - Microsoft Visual C++ [run] - nand_comm.cpp
File Edit View Project Build Debug Tools Window He
Start Page nand_comm.cpp
(Globals)
Command Prompt
nand_comm
e:\Documents and Settings\Sergei\Desktop\nand_comm\Debug\nand_comm.exe
start

```
//Get Blocks Checksums
if (chip&0x01)  {
fprintf(stream_log,"\nSCAN NAND_1");
//for (li=0x000000L; li<=0x000427L; li++
//for (li=0x000800L; li<=0x000C27L; li++
//for (li=0x000000L; li<=0x000C27L; li++
for (li=0x000001L; li<=0x000000L; li++)
    if (li==0x000428L)  li=0x000800L;
//if (li==0x000BB1L)  li++;
//if (li==0x000BD0L)  li++;
    if ((li&0x0000FFL)==0)  printf("A=%06lX
    //CHECK BLOCK NAND_1
    cmd=0x85;
```

```
E:\Documents and Settings\Sergei\Desktop\nand_comm>fc /lb100000 nand_comm_n1aug1
7b.log nand_comm_n1aug17f.log >cmp.txt

E:\Documents and Settings\Sergei\Desktop\nand_comm>
```

```
iPhone5C NAND analysis software
Version 1.12 (from 11.08.16)

Searching for the control board (SHEM0203)...
Board found at address: COM1

Output Data: nand_data.bin  Output Log: nand_comm.log  Input Data: nand_load.bin

NAND_1 ID: 50 50 4E 01 05 07 00 00
NAND_2 ID: 50 50 4E 01 05 07 00 00
NAND_1 and NAND_2 chips present

NAND_1 Feature 8042: ACK=40, Feature 8001: ACK=40

NAND_2 Feature 8042: ACK=40, Feature 8001: ACK=40

NAND_2 Read EXT: ADDR=0000007F, ACK=40, Time: 196 us
A=000000  A=000100  A=000200  A=000300  A=000400  A=000800  A=000900  A=000A00
A=000B00  A=000C00  Total 8 pages written

End of operation, press any key to exit
_
```


nand_comm - Microsoft Visual C++ [design] - nand_comm.cpp*
File Edit View Project Build Debug Tools Window He
Start Page nand_comm.cpp*
(Globals)
Command Prompt
cmp.txt - Notepad
File Edit Format View Help
Search Folders
Properties Dynamic Help
Ln 249 Col 1 Ch 1 INS
Build succeeded
start

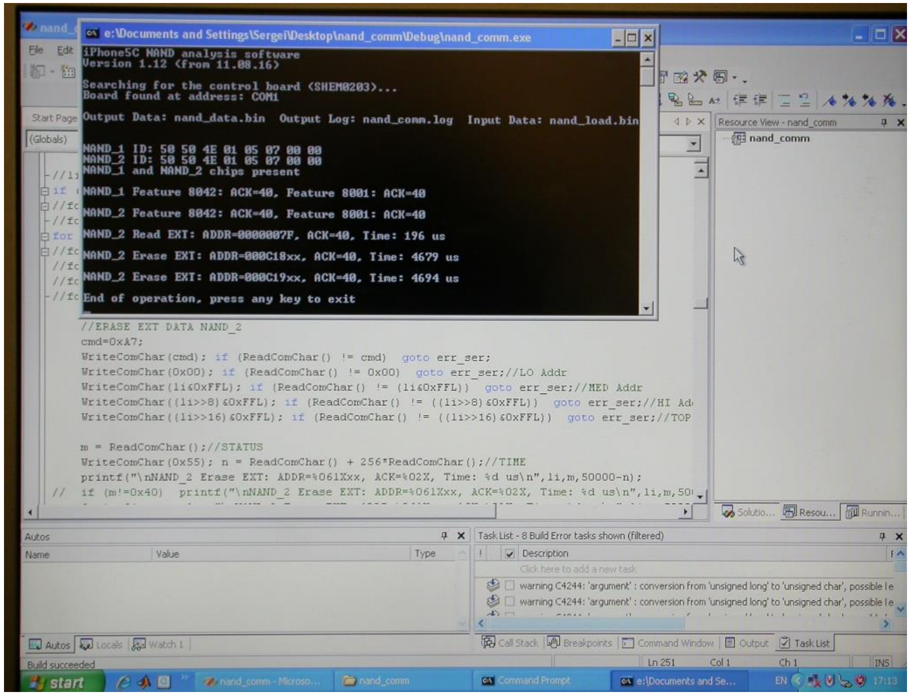
```
E:\Documents and Settings\Sergei\Desktop\nand_comm>fc /lb100000 nand_comm_n1aug1
7b.log nand_comm_n1aug17f.log >cmp.txt

E:\Documents and Settings\Sergei\Desktop\nand_comm>fc /lb100000 nand_comm_n1aug1
7b.log nand_comm_n1aug18a.log >cmp.txt

E:\Documents and Settings\Sergei\Desktop\nand_comm>
```

```
0008B147:40:0194
0008B148:40:E4E7
0008B149:40:0E01
0008B14A:40:0D6E
0008B14B:40:EB19
0008B14C:40:A23E
0008B14D:40:BE11
0008B14E:49:FFFF
*****

***** nand_comm_n1aug17b.log

0008B200:49:FFFF
0008B201:49:FFFF
0008B202:49:FFFF
0008B203:49:FFFF
0008B204:49:FFFF
0008B205:49:FFFF
***** NAND_COMM_N1AUG18A.LOG

0008B200:40:4BF6
0008B201:40:4B86
0008B202:40:3DB2
0008B203:40:5C99
0008B204:40:FD8C
0008B205:49:FFFF
*****

***** nand_comm_n1aug17b.log
000C183C:49:FFFF
000C183D:61:ECD1
000C183E:49:FFFF
000C183F:61:E5D5
000C1840:49:FFFF
000C1841:61:E145
000C1842:49:FFFF
```

Name	Size	Type
nand_comm_n1aug17f.log	9,581 KB	Text Document
nand_comm_old.cpp	14 KB	C++ Source
nand_data1_7f.bin	17 KB	VLC media file (.bin)
nand_data2_7f.bin	17 KB	VLC media file (.bin)
nand_data.bin	0 KB	VLC media file (.bin)
nand_data_n1_7f.bin	17 KB	VLC media file (.bin)
nand_data_n2_7f.bin	17 KB	VLC media file (.bin)
nand_load.bin	17 KB	VLC media file (.bin)
ReadMe.txt	2 KB	Text Document
stdafx.cpp	1 KB	C++ Source
stdafx.h	1 KB	C/C++ Header
cmp.txt	198 KB	Text Document

iPhone 5C NAND mirroring

- Mirroring is the process of restoring the original contents of storage from backup copy
  - erase all blocks that contain changed pages
  - write back pages from the backup
  - plug the NAND back into iPhone and observe the result
  - video: https://youtu.be/tM66GWrwbsY



e:\Documents and Settings\Sergei\Desktop\nand_comm\Debug\nand_comm.exe

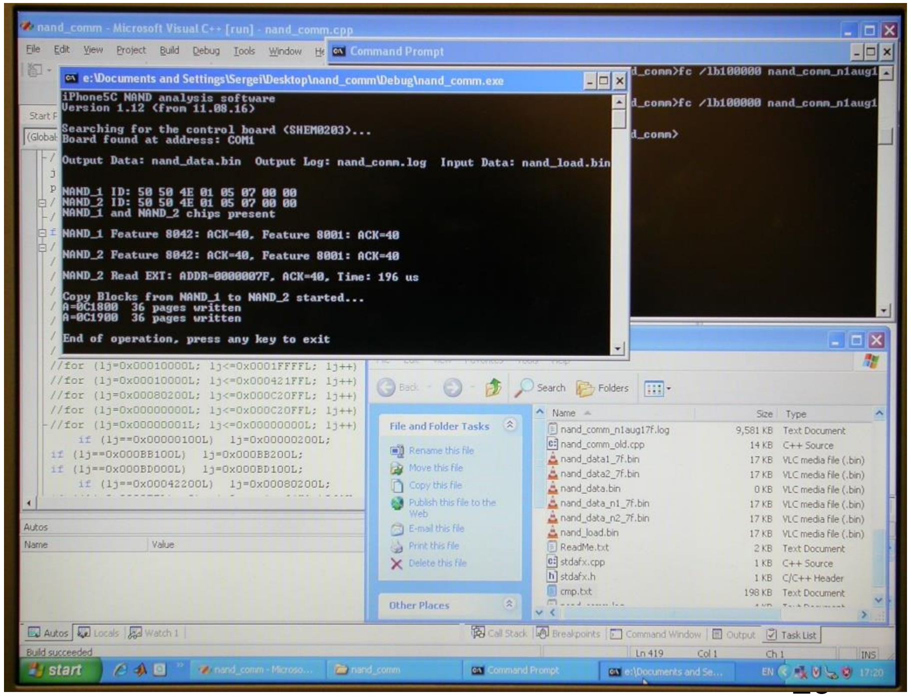
```
iPhone5C NAND analysis software
Version 1.12 (from 11.08.16)

Searching for the control board (SHEM0203)...
Board found at address: COM1

Output Data: nand_data.bin  Output Log: nand_comm.log  Input Data: nand_load.bin


NAND_1 ID: 50 50 4E 01 05 07 00 00
NAND_2 ID: 50 50 4E 01 05 07 00 00
NAND_1 and NAND_2 chips present

NAND_1 Feature 8042: ACK=40, Feature 8001: ACK=40

NAND_2 Feature 8042: ACK=40, Feature 8001: ACK=40

NAND_2 Read EXT: ADDR=0000007F, ACK=40, Time: 196 us

NAND_2 Erase EXT: ADDR=000C18xx, ACK=40, Time: 4679 us

NAND_2 Erase EXT: ADDR=000C19xx, ACK=40, Time: 4694 us

End of operation, press any key to exit
```

```
    //ERASE EXT DATA NAND_2
    cmd=0xA7;
    WriteComChar(cmd); if (ReadComChar() != cmd)  goto err_ser;
    WriteComChar(0x00); if (ReadComChar() != 0x00)  goto err_ser;//LO Addr
    WriteComChar(li&0xFFL); if (ReadComChar() != (li&0xFFL))  goto err_ser;//MED Addr
    WriteComChar((li>>8)&0xFFL); if (ReadComChar() != ((li>>8)&0xFFL))  goto err_ser;//HI Ad
    WriteComChar((li>>16)&0xFFL); if (ReadComChar() != ((li>>16)&0xFFL))  goto err_ser;//TOP

    m = ReadComChar();//STATUS
    WriteComChar(0x55); n = ReadComChar() + 256*ReadComChar();//TIME
    printf("\nNAND_2 Erase EXT: ADDR=%061Xxx, ACK=%02X, Time: %d us\n",li,m,50000-n);
//  if (m!=0x40)  printf("\nNAND_2 Erase EXT: ADDR=%061Xxx, ACK=%02X, Time: %d us\n",li,m,50
```

Task List - 8 Build Error tasks shown (filtered)

```
warning C4244: 'argument' : conversion from 'unsigned long' to 'unsigned char', possible l
warning C4244: 'argument' : conversion from 'unsigned long' to 'unsigned char', possible l
```

Build succeeded



nand_comm - Microsoft Visual C++ [run] - nand_comm.cpp

Command Prompt

```
d_comm>fc /lb100000 nand_comm_n1aug1
d_comm>fc /lb100000 nand_comm_n1aug1
d_comm>
```

e:\Documents and Settings\Sergei\Desktop\nand_comm\Debug\nand_comm.exe

```
iPhone5C NAND analysis software
Version 1.12 (from 11.08.16)

Searching for the control board (SHEM0203)...
Board found at address: COM1

Output Data: nand_data.bin  Output Log: nand_comm.log  Input Data: nand_load.bin


NAND_1 ID: 50 50 4E 01 05 07 00 00
NAND_2 ID: 50 50 4E 01 05 07 00 00
NAND_1 and NAND_2 chips present

NAND_1 Feature 8042: ACK=40, Feature 8001: ACK=40

NAND_2 Feature 8042: ACK=40, Feature 8001: ACK=40

NAND_2 Read EXT: ADDR=0000007F, ACK=40, Time: 196 us

Copy Blocks from NAND_1 to NAND_2 started...
A=0C1800  36 pages written
A=0C1900  36 pages written

End of operation, press any key to exit
```

```
//for (lj=0x00010000L; lj<=0x0001FFFFL; lj++)
//for (lj=0x00010000L; lj<=0x000421FFL; lj++)
//for (lj=0x00080200L; lj<=0x000C20FFL; lj++)
//for (lj=0x00000000L; lj<=0x000C20FFL; lj++)
//for (lj=0x00000001L; lj<=0x00000000L; lj++)
     if (lj==0x00000100L)  lj=0x00000200L;
if (lj==0x000BB100L)  lj=0x000BB200L;
if (lj==0x000BD000L)  lj=0x000BD100L;
     if (lj==0x00042200L)  lj=0x00080200L;
```

Name	Size	Type
nand_comm_n1aug17f.log	9,581 KB	Text Document
nand_comm_old.cpp	14 KB	C++ Source
nand_data1_7f.bin	17 KB	VLC media file (.bin)
nand_data2_7f.bin	17 KB	VLC media file (.bin)
nand_data.bin	0 KB	VLC media file (.bin)
nand_data_n1_7f.bin	17 KB	VLC media file (.bin)
nand_data_n2_7f.bin	17 KB	VLC media file (.bin)
nand_load.bin	17 KB	VLC media file (.bin)
ReadMe.txt	2 KB	Text Document
stdafx.cpp	1 KB	C++ Source
stdafx.h	1 KB	C/C++ Header
cmp.txt	198 KB	Text Document

Build succeeded

Limitations and improvements

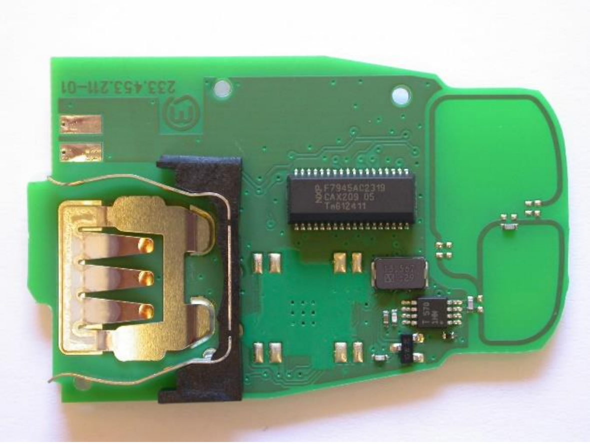
- The mirroring works well for iPhone 5C provided all pages were restored to the original state, any mistakes will result in OS crash during the boot process
- The process can be fully automated to avoid unnecessary rewrites and increase the update speed
- Standard microcontrollers cannot handle NAND at high speed, better to use FPGAs
- NAND storage has limited number of rewrites, usually tens of thousands, better to emulate with FPGA
- Only iPhone 5C was tested so far
- iPhone 5S and 6 have the same type of NAND flash, but iPhone 6S and 7 use more advanced high speed serial NAND chips

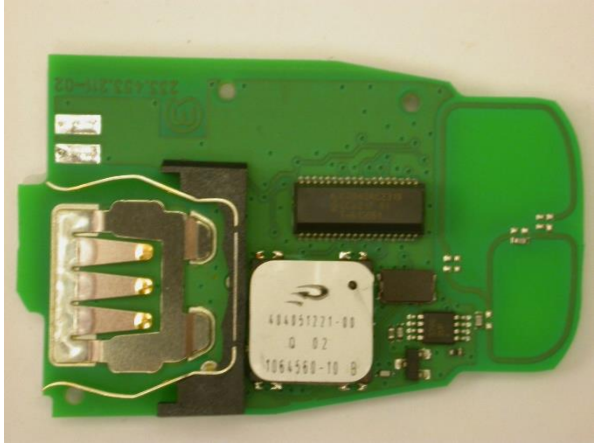
Audi Smart Key security (work-in-progress)

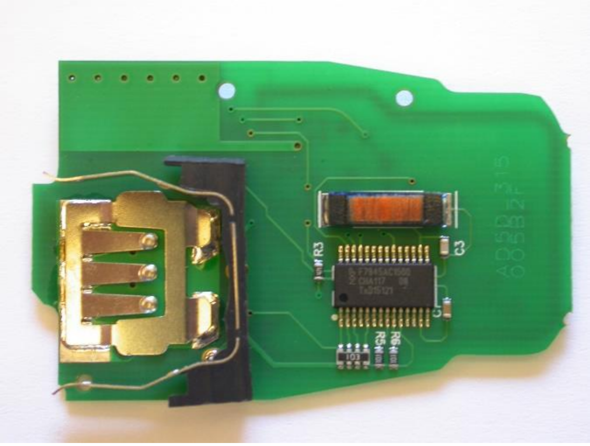
- Based on Philips/NXP PCF7945 transponder IC chip
  - sometime referred to as RKE (remote key entry) device (125kHz + 433/866MHz)
- The same chip is used by other manufacturers (BMW, VW, Porsche etc.)
- There are other similar chips in the NXP line (PCF7941, PCF7952, PCF7953, PCF7961)
- Only abridged datasheets are available
- By default they use HITAG2 protocol, but can be switched to user programmed mode
  - Garcia RV, Balasch J. Gone in 360 Seconds: Hijacking with Hitag2. Usenix 2012
- No information or tools are available for 8-bit RISC MRKII CPU inside those chips
- Challenges
  - look at real samples of keys
  - analyse the PCF794x, PCF795x and PCF796x ICs

Audi Smart Key samples

- Fully functional key
  - original key is based on PCF7945 in SOP with embedded coil antenna
  - cloned key is based on PCF7945 in TSOP with external antenna

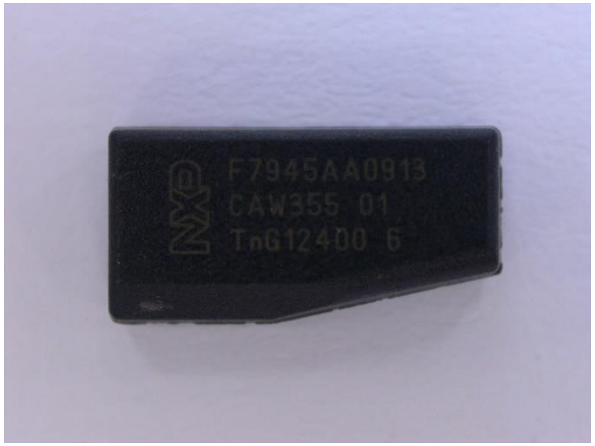


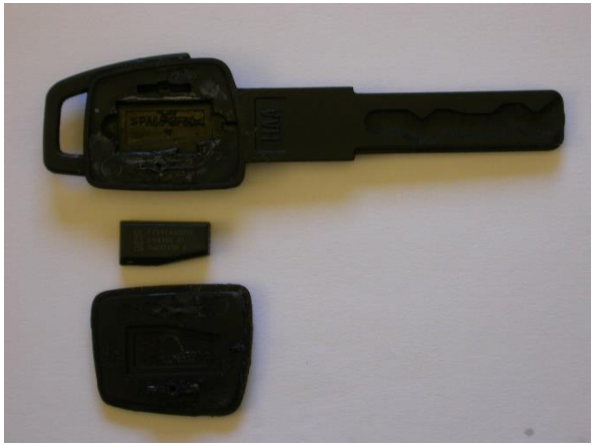


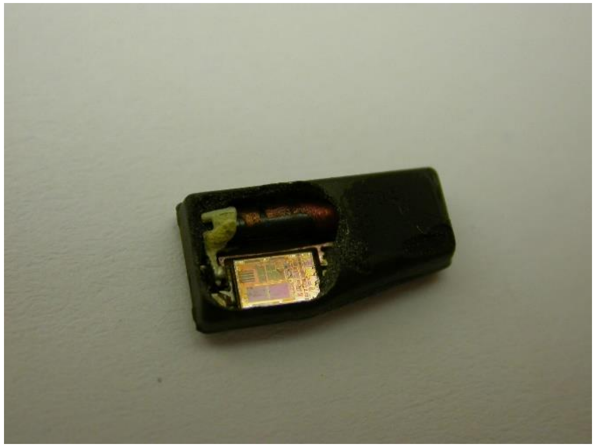


Audi Smart Key samples

- Spare plastic key
  - based on the same PCF7945 but in leadless plastic package with embedded coil antenna

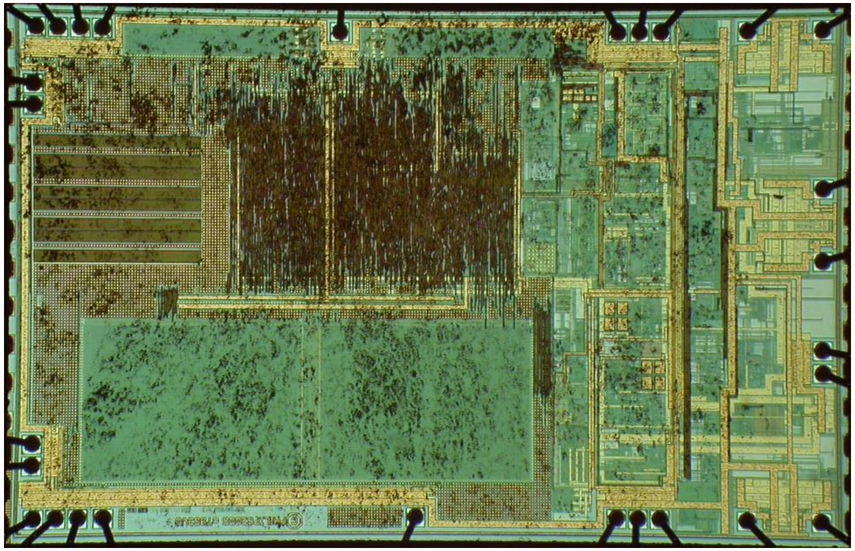


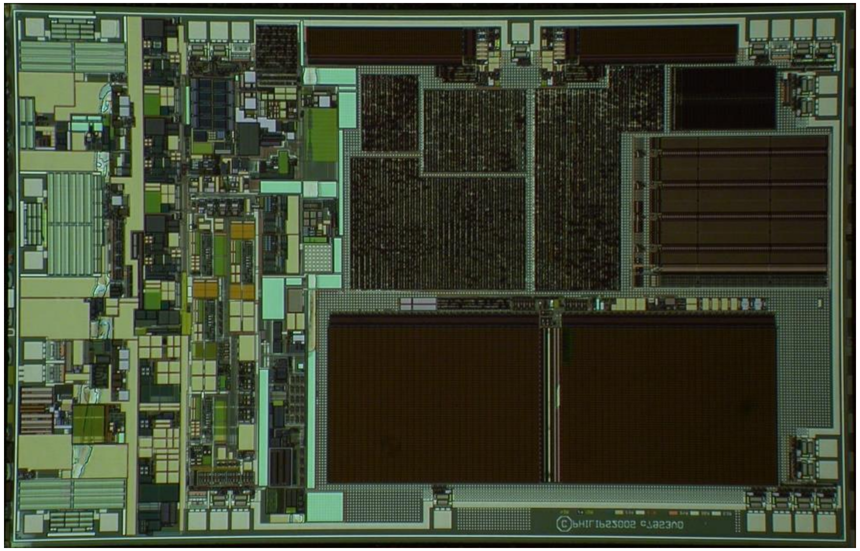




Audi Smart Key ICs

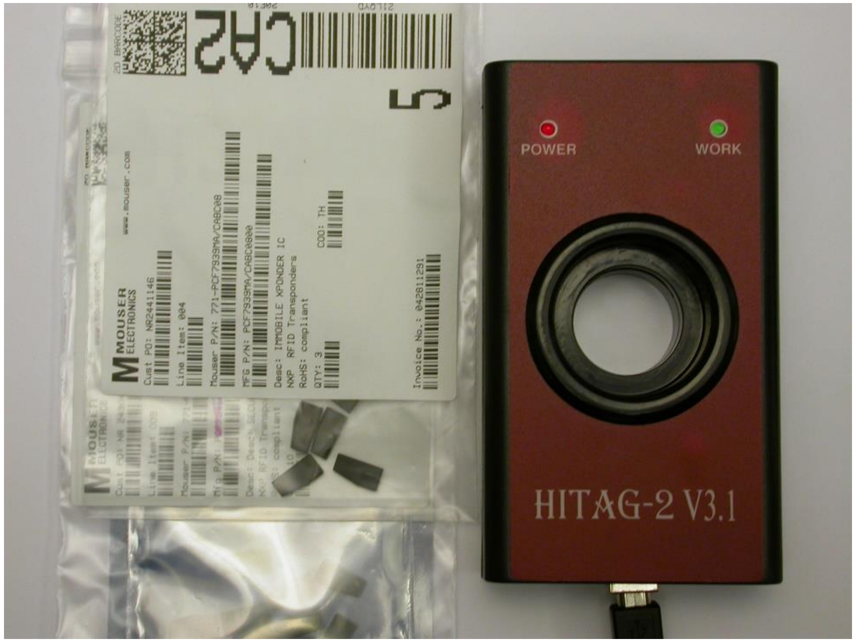
- PCF7945 (or PCF7953?)
  - die marking: ©PHILIPS2005 c7953V0
  - 8kB MaskROM (libraries)
  - 512B EEPROM (ID, keys, settings)
  - 4kB E-ROM (user code)
  - 196B SRAM (variables)
  - 8-bit RISC CPU (MRKII)
  - fabricated with 0.35μm process with 3 metal layers

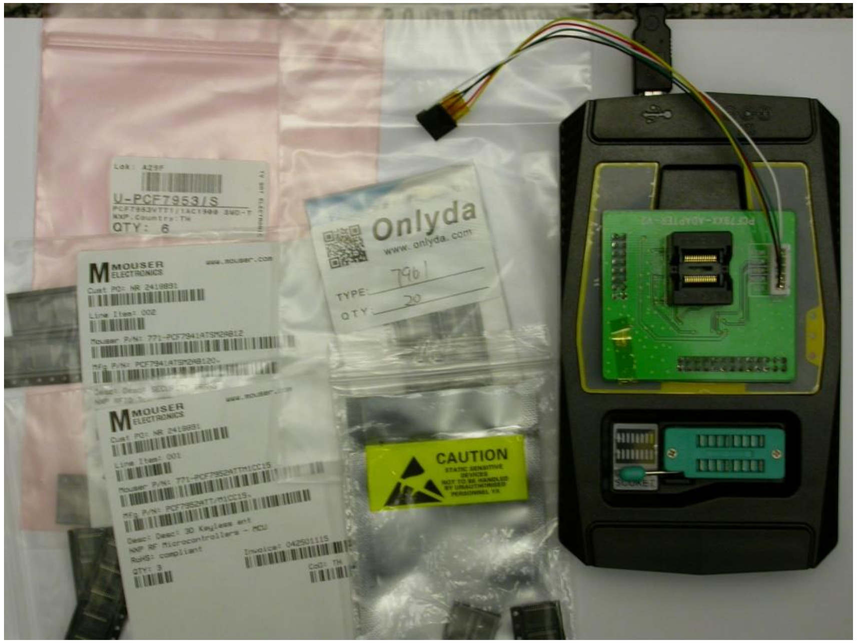




Security Analysis

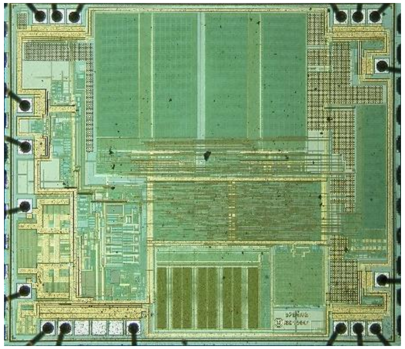
- Samples can be obtained from Ebay, AliExpress, as well as from electronic distributors (Mouser, DigiKey, Avnet etc)
- Programmers can be bought on Ebay and AliExpress
  - HITAG2 only supports transponders
  - VVDI Programmer supports ICs in standard packages



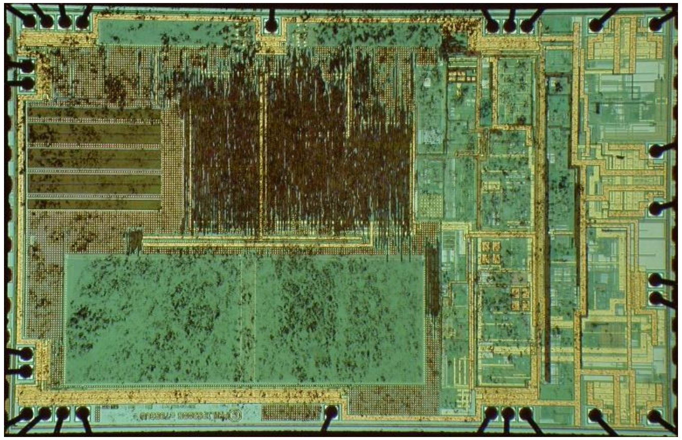


Audi Smart Key ICs

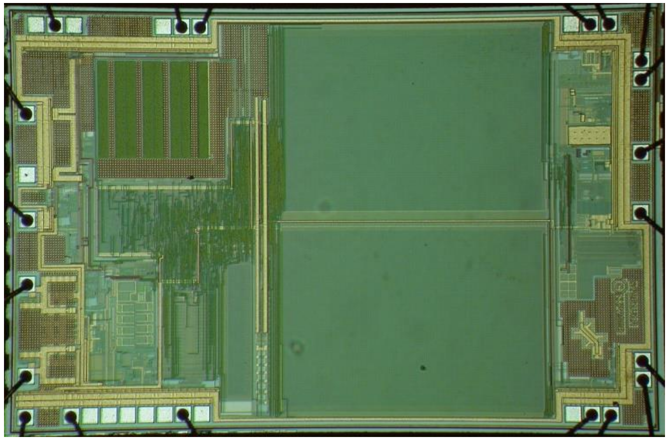
- Other chips from the family: PCF7941, PCF7952, PCF7953, PCF7961
  - different packages
  - different memory sizes
  - similar layout
  - small variation of fabrication process
  - similar memory structure



©NXP2007 c7941V2



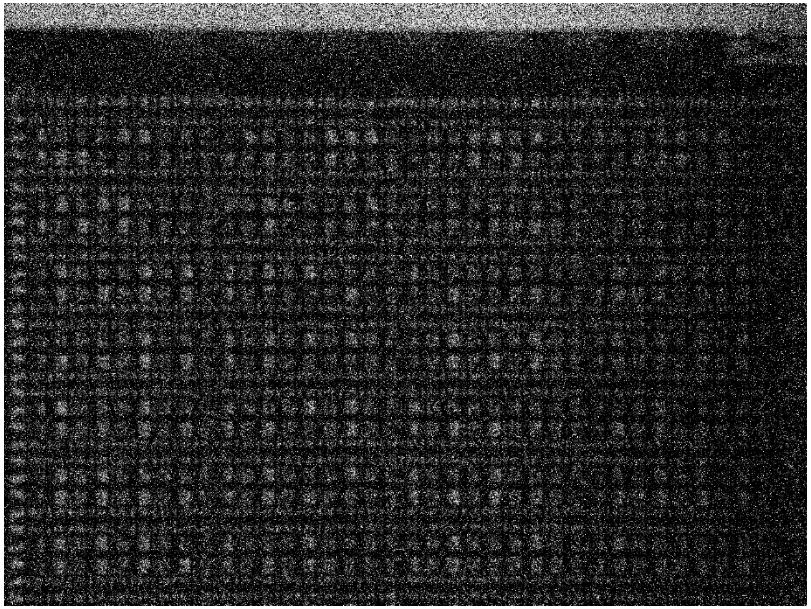
©PHILIPS2005 c7953V0



©NXP2008 c7961V1A

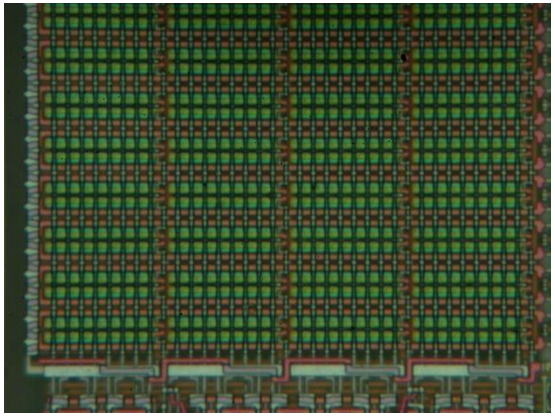
Embedded Memory extraction

- Security protection is based on a certain bit set in EEPROM
  - no known non-invasive and semi-invasive methods that can defeat it
- Invasive methods
  - microprobing will require FIB editing
  - Scanning Probe Microscopy will require sophisticated equipment
  - Scanning Electron Microscopy works well for 0.35μm and larger processes
  - samples can be programmed with test pattern before the analysis



Results

- Initial analysis was performed on PCF7941 chips (easier to obtain)
  - test pattern revealed the physical layout of the EEPROM memory
  - sequential addresses, but scrambled data
  - not even XOR like in other car RKE chips – just swapping the bit lines
- Once the memory layout is decoded, it can be used for real parts
  - Crypto Key is either 48 or 96 bits for HITAG and 128 bit for AES
  - small IC with weak security could protect assets in excess of 100,000€



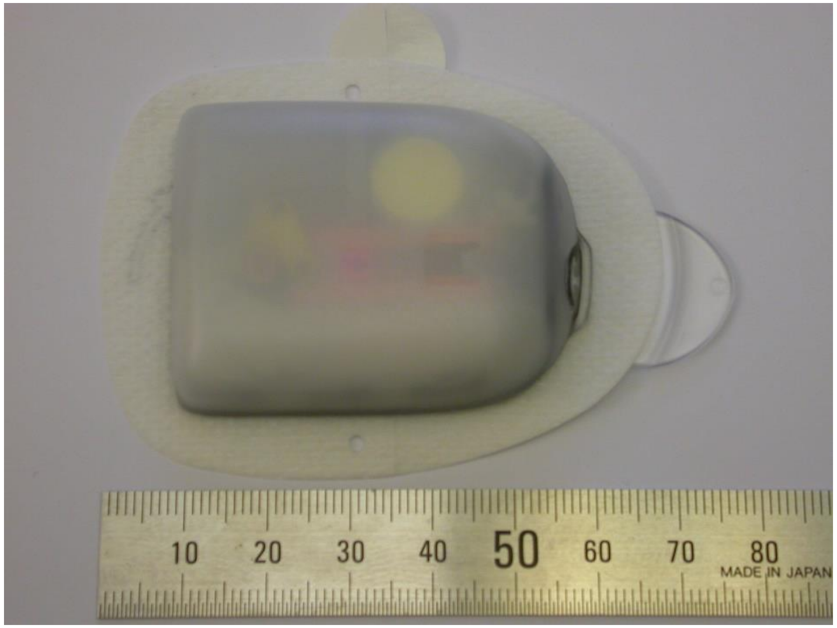
D6 D1 D4 D3 D2 D5 D0 D7 D1 D4 D3 D6 D5 D0 D7 D2 D3 D2 D1 D6 D5 D0 D7 D4 D2 D7 D4 D3 D0 D5 D6 D1

Limitations and improvements

- The attack works well for EEPROM fabricated with 0.35μm process, but too challenging beyond 180nm process
- Proper memory encryption will prevent any use of image
- Lack of documentation makes the analysis too challenging
- What Audi dealership claimed as being a piece of plastic is actually the fully functional key!
  - it can be used to start the engine and drive the car off
  - it has exactly the same functionality as the standard key when its battery is removed
- The protocol is unknown (could be HITAG2, HITAG Pro, custom or AES), but the memory is not encrypted
- Sensitive information can be extracted and used
- Audi dealerships mislead their customers on security

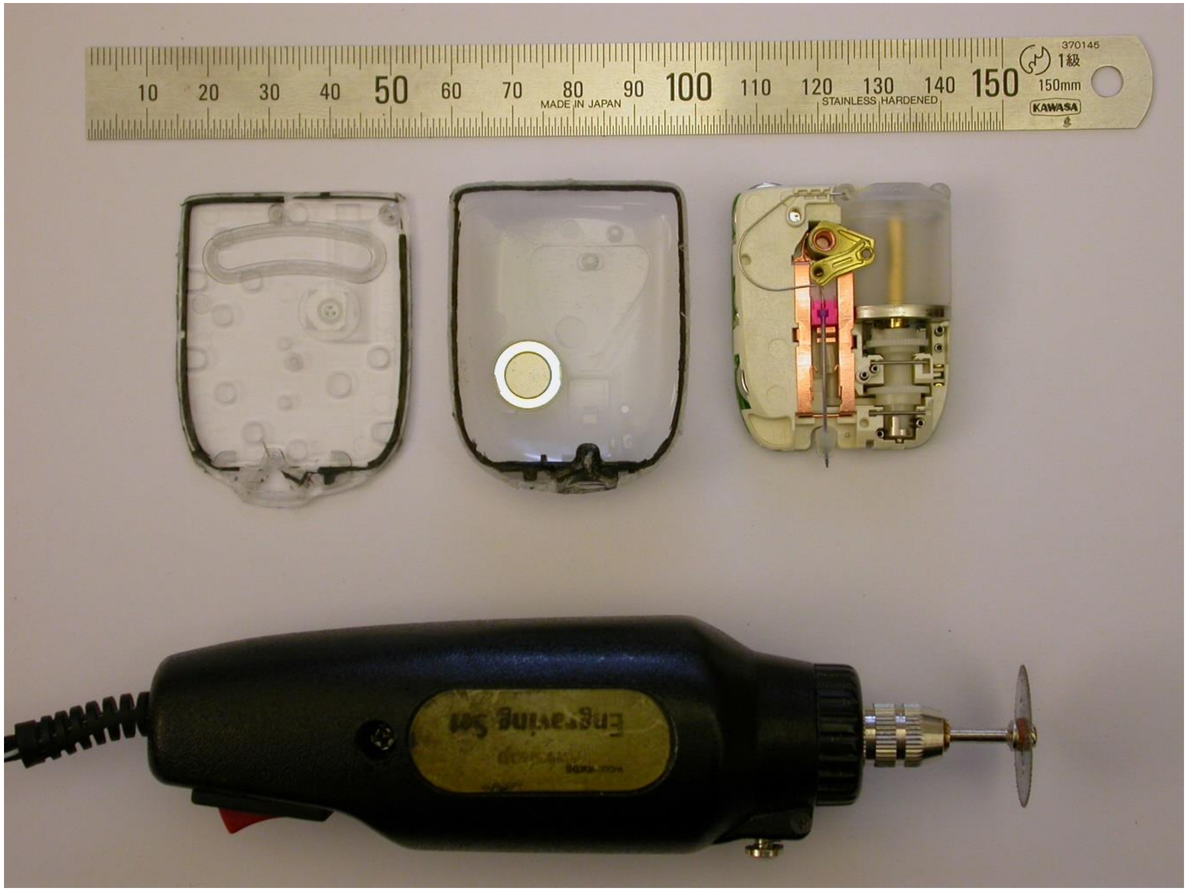
Insulet OmniPod teardown

- Insulet OmniPod wireless tubeless insulin pump
  - complete reverse engineering for the purpose of building compatible open source controller acting as an artificial pancreas (OpenAPS, OpenOmni, Loop project)
- Original setup consists of PDM (Personal Diabetes Monitor) and Pod
  - custom hardware without any details and information



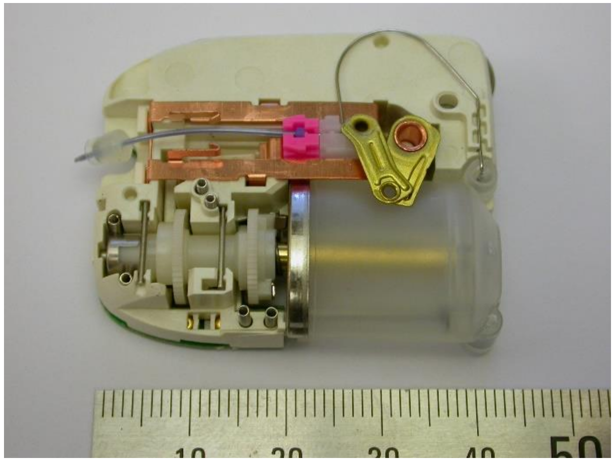
Mechanical teardown

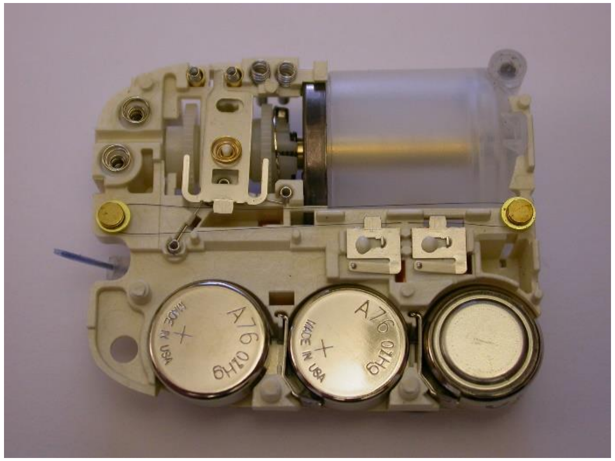
- Engraving tools with circular saw to cut through plastic
  - reveals internal structure for further analysis
  - video: https://youtu.be/ZYSnOnE0Ns4

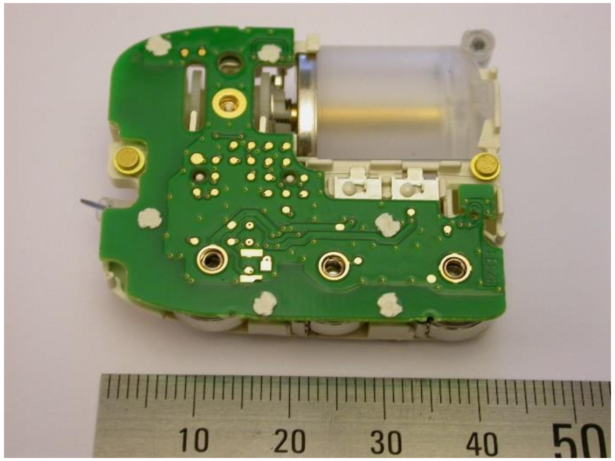


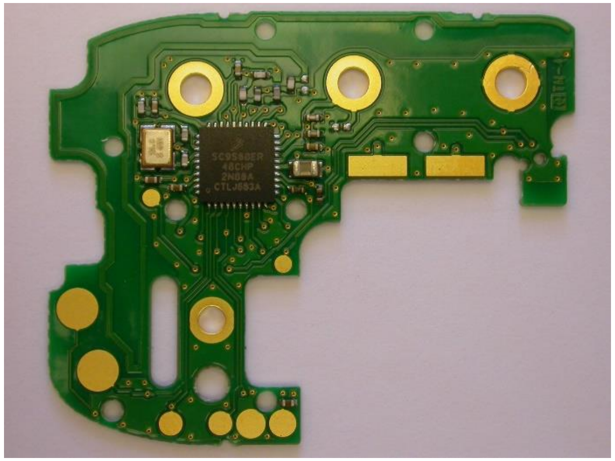
Mechanical teardown

- Inside the Pod
  - gears driven by muscle wire motor
  - cannula insertion mechanism and insulin reservoir with clutch to the gears



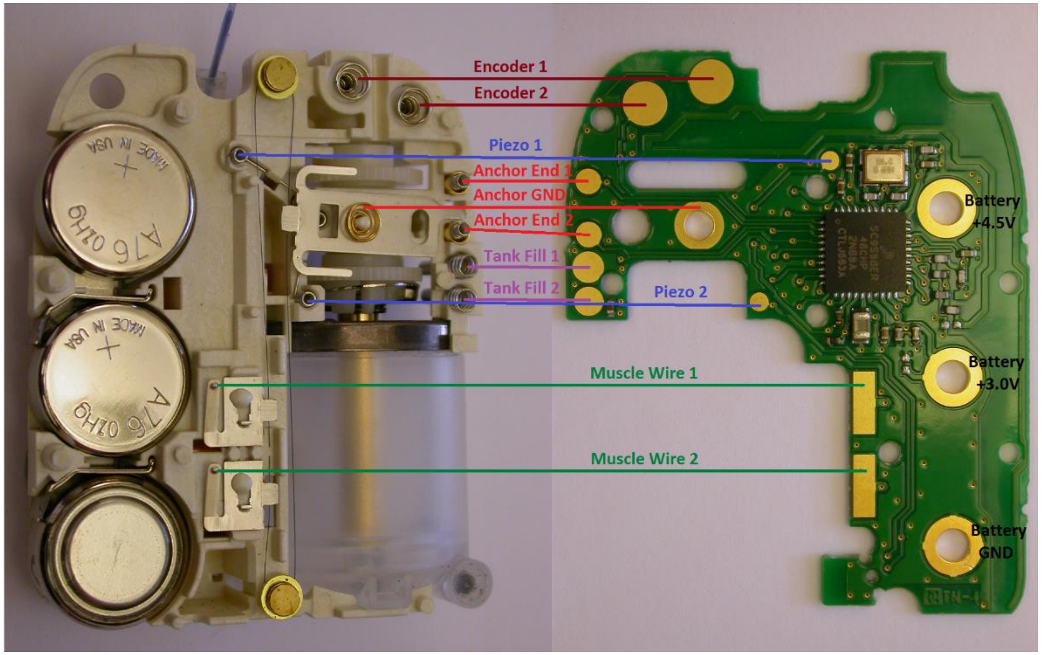


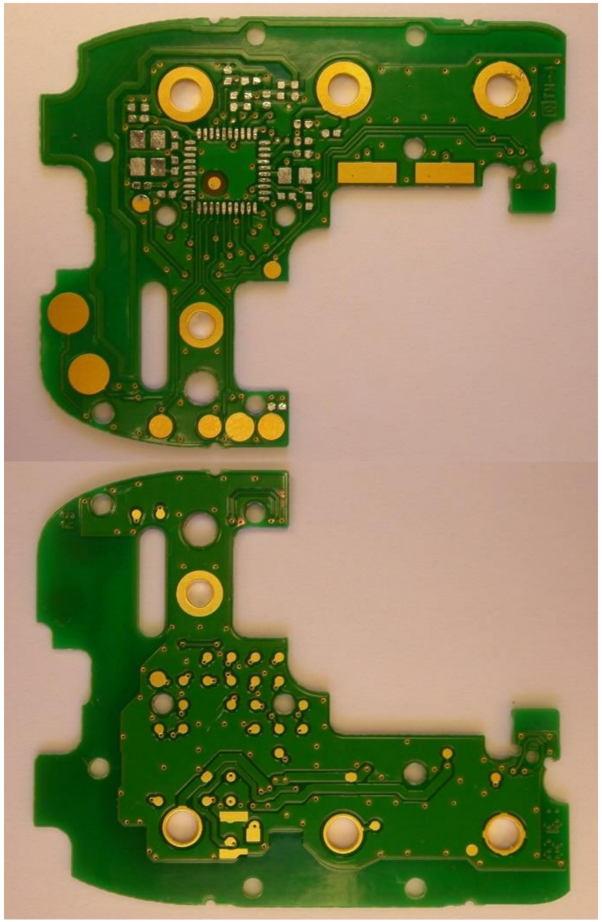




Electrical teardown

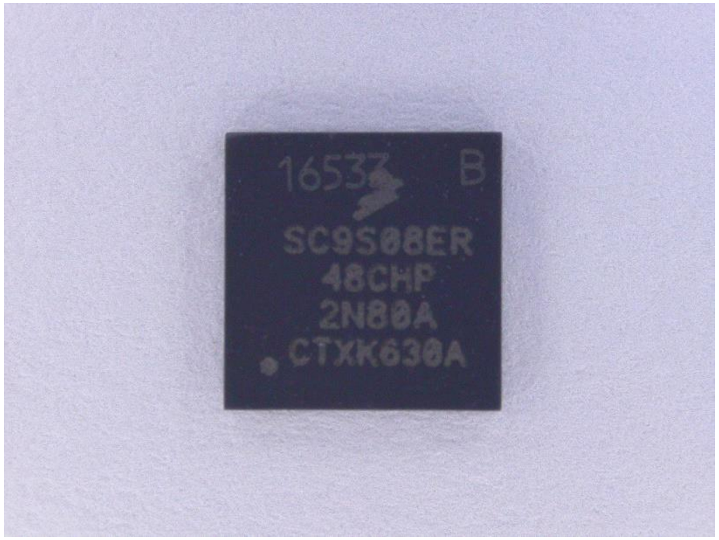
- Tracing connections to the mechanical part
- Creating the schematic by tracing 2-layer PCB

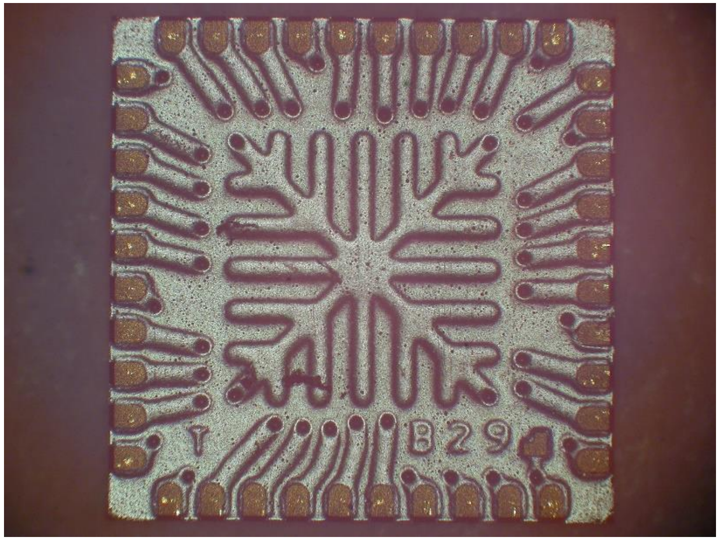




Insulet OmniPod teardown

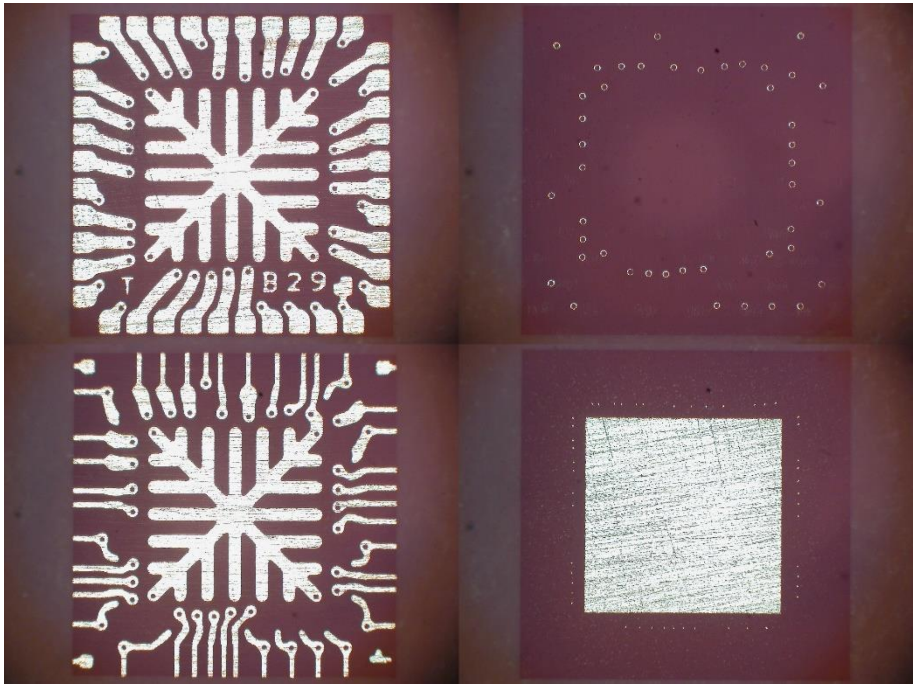
- Pin functions were determined from the PCB and IC measurements
  - connections to pins
  - voltages during operation
  - voltage drop to GND and V_{DD}
- Wireless communication analysis
  - already carried out by OpenOmni community
  - 433.9MHz with FSK-2 modulation, 40625 baud with CRC





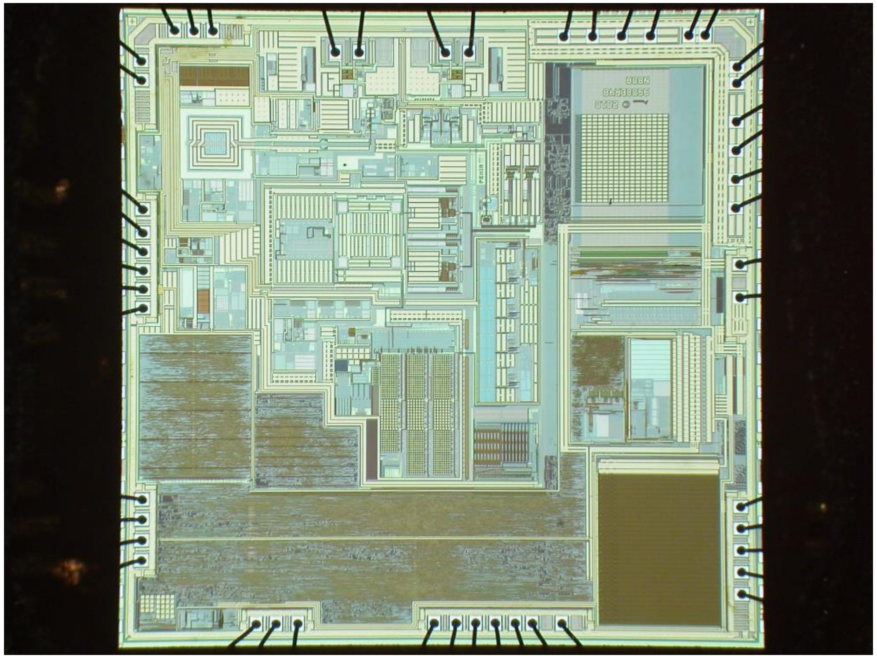
IC teardown

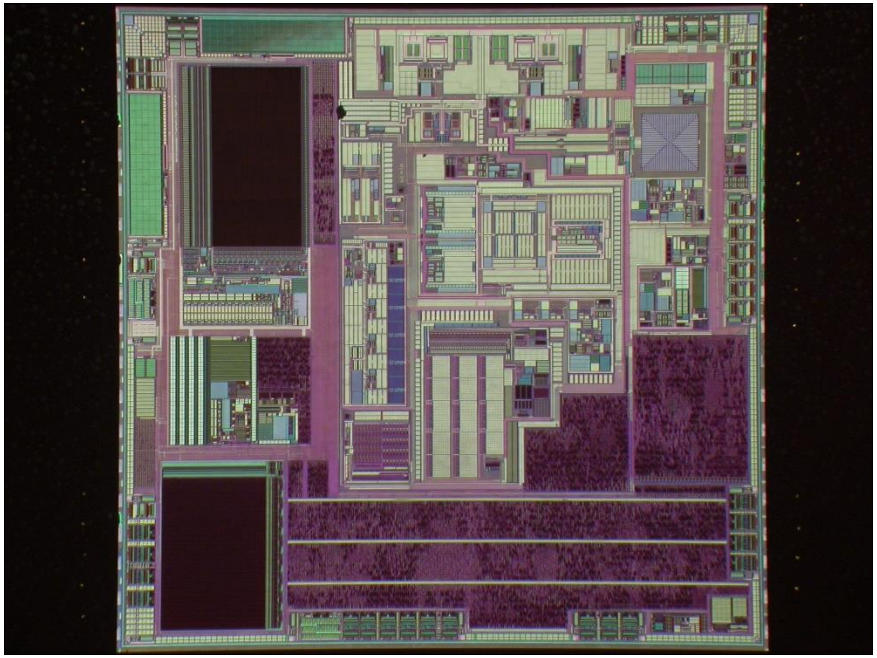
- HCS08 SoC (System-on -Chip) device based on MC9S08QE32 with:
  - LDO regulator
  - buzzer module
  - power MOSFETs
  - RF communication module
- Tracing wiring from pins to pads on the die



IC teardown

- Decapsulation and deprocessing
  - silicon die marking and structures: Freescale ©2010 9S08ER48 N80A
  - connections of bonding wires
  - fabrication process: 0.25μm, 5M, CMOS
  - types and sizes of embedded memory: 48kB Flash, 4kB SRAM



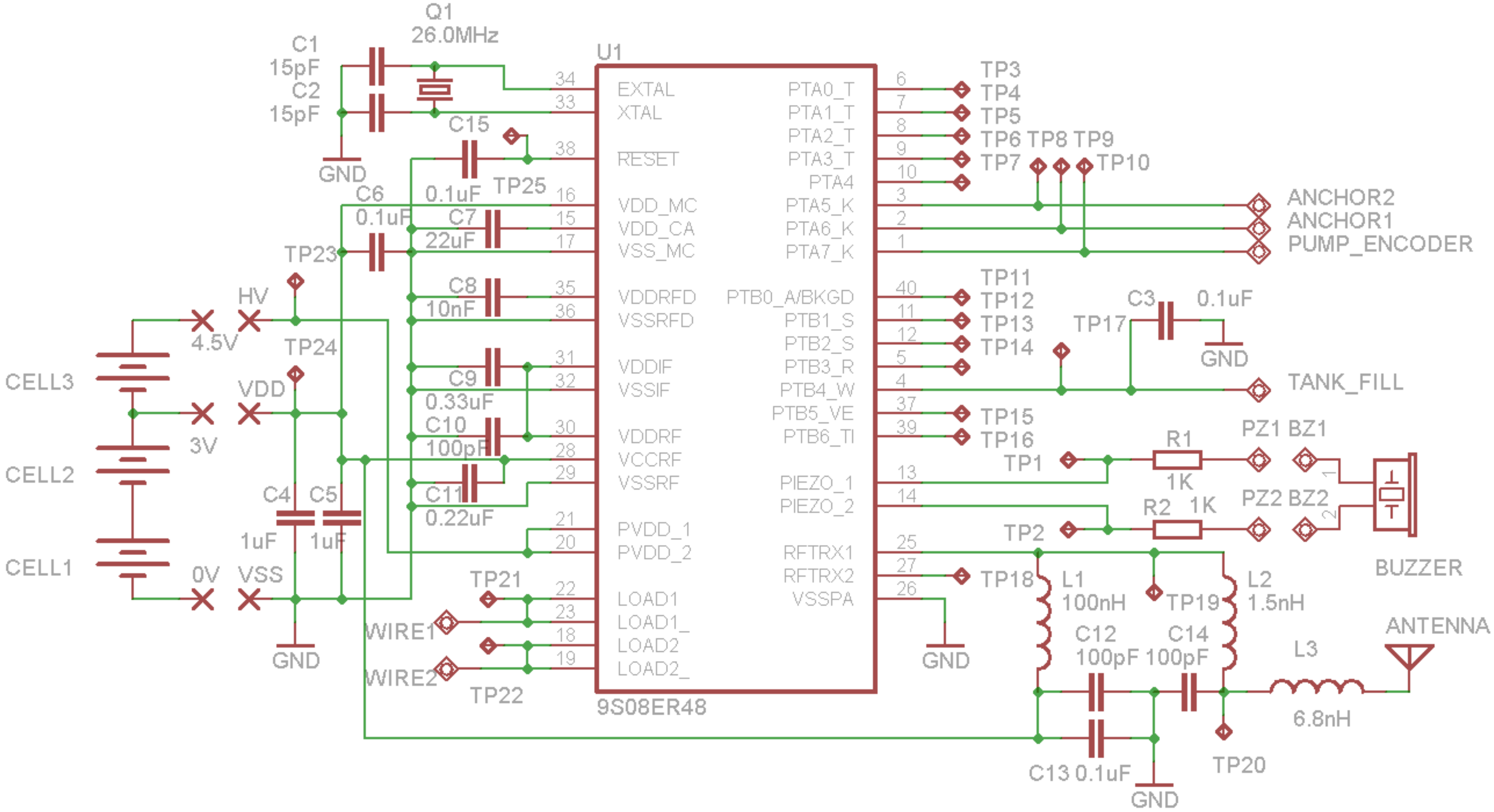


Insulet OmniPod teardown

- Other sources of information
  - Internet search: hidden pages on NXP website which were somehow indexed by Google confirmed the exact fabrication process and memory size
  - Freescale CodeWarrior development tools are free to download: information about all members of HCS08 family including 9S08ER48 such as pinout and functions
  - Universal programmer Elnec BeeProg2 supports 9S08ER48
  - PE Micro Flash programming tools support 9S08ER48
- Identification of the Debug interface
  - can be used for Flash readout and programming
  - only two pins: Reset and BKGD must be at high level during normal operation
  - can be traced on silicon die
  - can be found using universal programmer and QFN40 adapter
  - can be learned from development tools

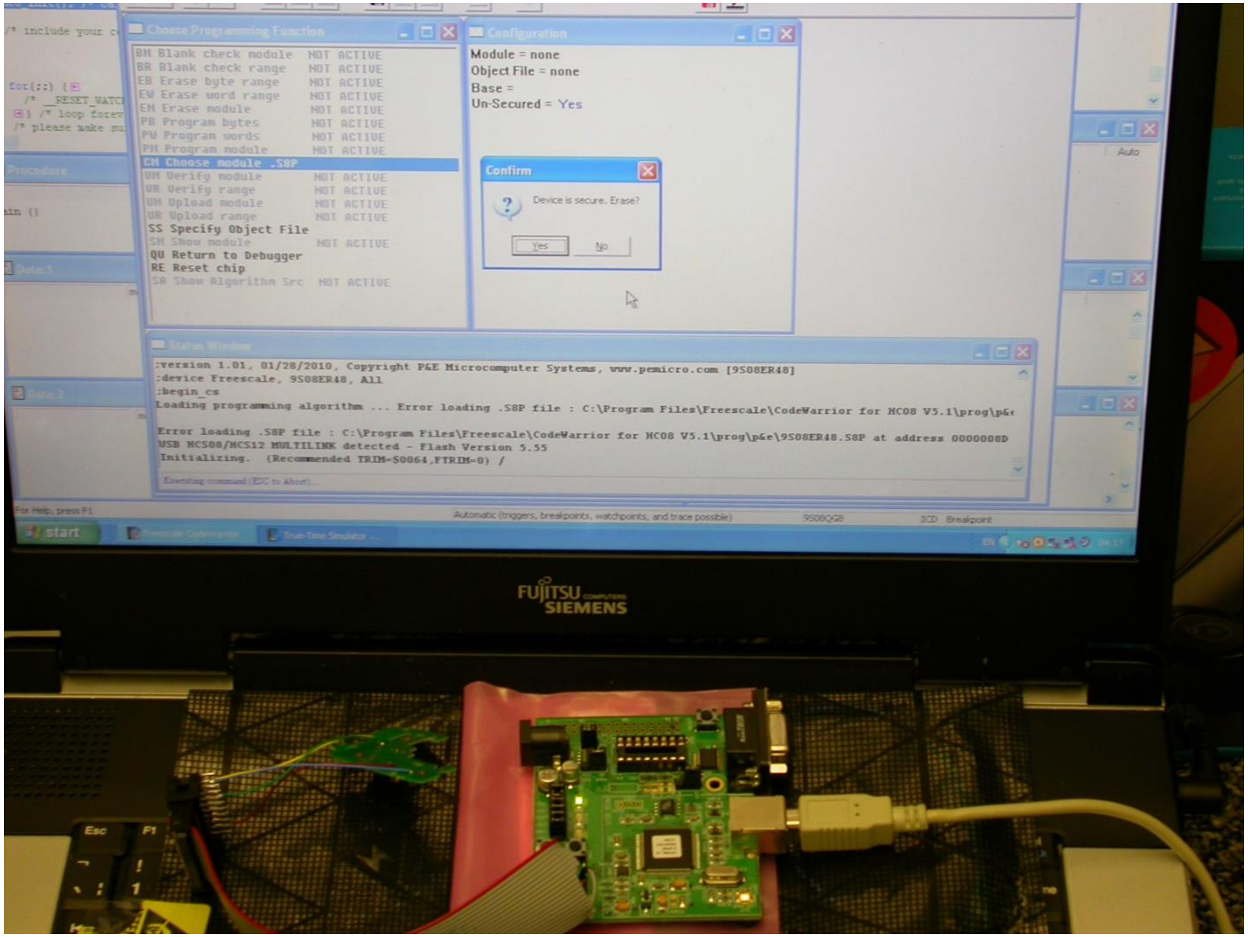
Insulet OmniPod schematic

- All the necessary information is obtained to create it using Eagle
  - PCB layout
  - values of components measured with LCR meter, crystal is marked



Firmware extraction

- Testing the Debug interface with development tools
  - Freescale CodeWarrior and 9S08QG8 Demo board
  - Security was activated on the SoC chip: can only Mass Erase the Flash



Choose Programming Function

```
BM Blank check module   NOT ACTIVE
BR Blank check range    NOT ACTIVE
EB Erase byte range     NOT ACTIVE
EW Erase word range     NOT ACTIVE
EM Erase module         NOT ACTIVE
PB Program bytes        NOT ACTIVE
PW Program words        NOT ACTIVE
PM Program module       NOT ACTIVE
CM Choose module .S8P
VM Verify module        NOT ACTIVE
VR Verify range         NOT ACTIVE
UM Upload module        NOT ACTIVE
UR Upload range         NOT ACTIVE
SS Specify Object File
SM Show module          NOT ACTIVE
QU Return to Debugger
RE Reset chip
SA Show Algorithm Src   NOT ACTIVE
```

Configuration

Module = none
Object File = none
Base =
Un-Secured = Yes

Confirm

Device is secure. Erase?

Yes No

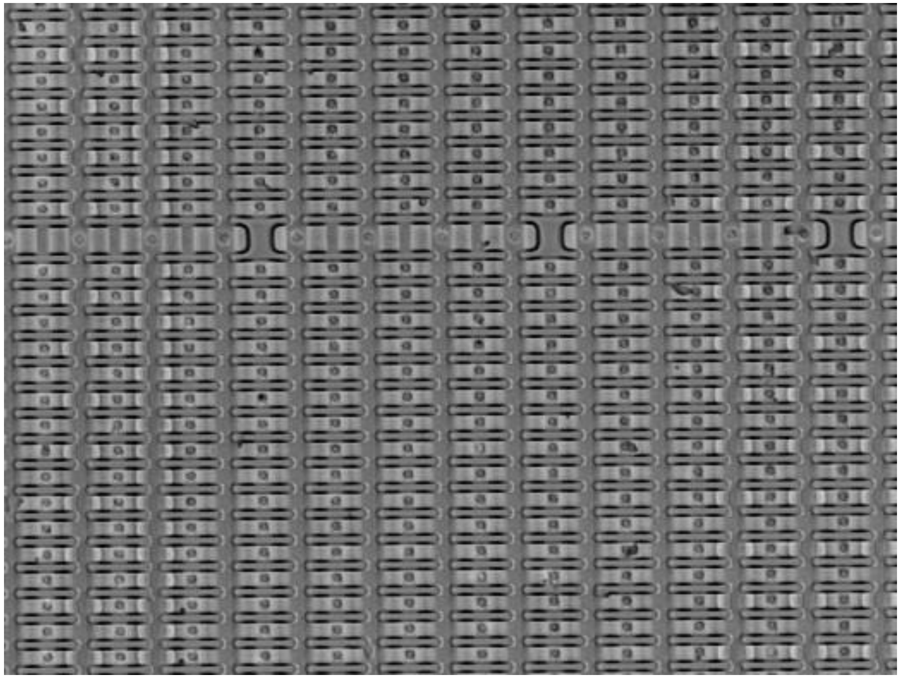
Status Window

```
:version 1.01, 01/28/2010, Copyright P&E Microcomputer Systems, www.pemicro.com [9S08ER48]
:device Freescale, 9S08ER48, All
:begin_cs
Loading programming algorithm ... Error loading .S8P file : C:\Program Files\Freescale\CodeWarrior for HC08 V5.1\prog\p&e

Error loading .S8P file : C:\Program Files\Freescale\CodeWarrior for HC08 V5.1\prog\p&e\9S08ER48.S8P at address 0000008D
USB HCS08/HCS12 MULTILINK detected - Flash Version 5.55
Initializing.  (Recommended TRIM=$0064,FTRIM=0) /
```

Executing command (ESC to Abort)...

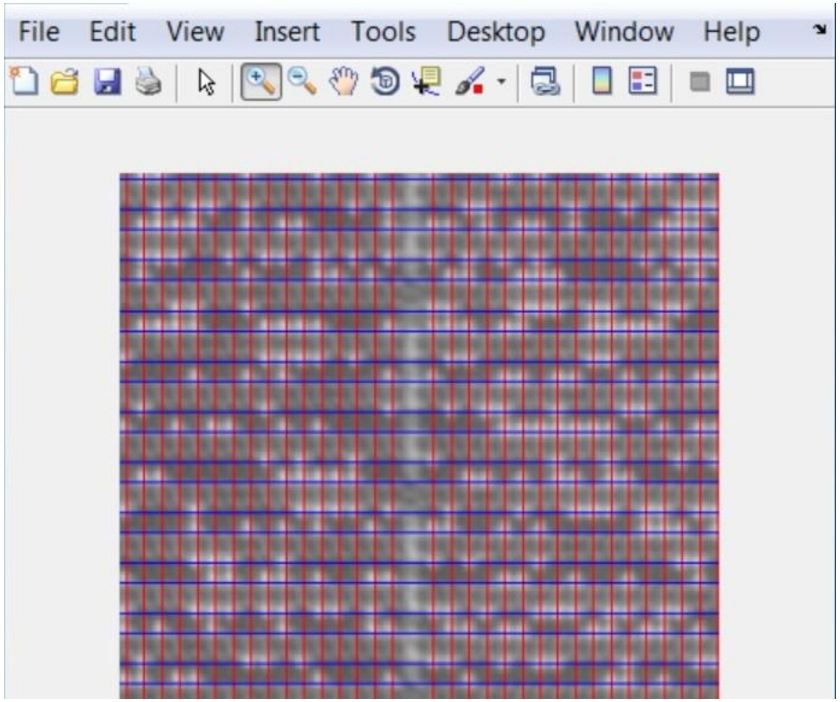
Firmware extraction

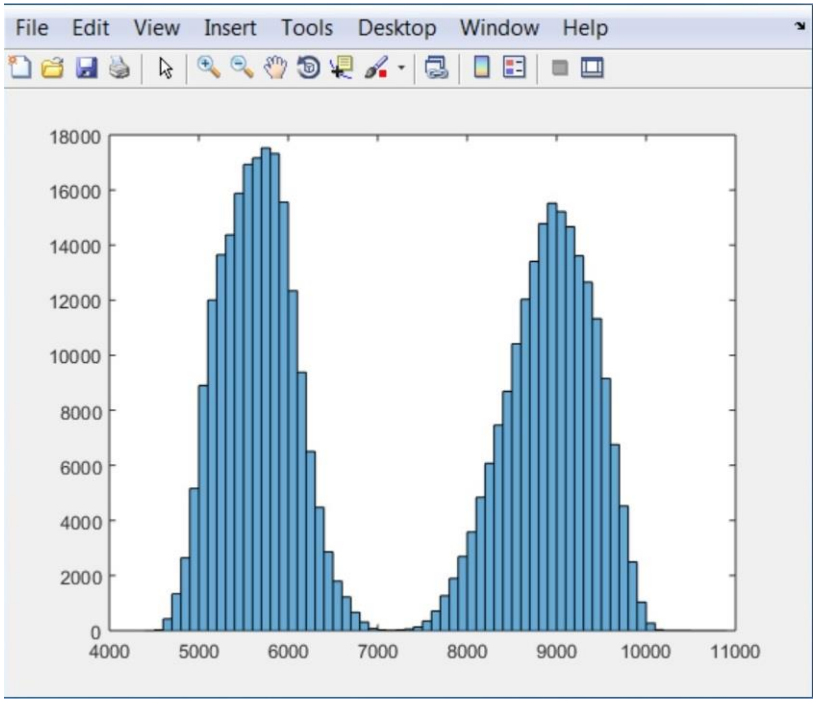
- Non-invasive and Semi-invasive attacks were unsuccessful
- Invasive attacks
  - FIB editing is expensive and requires partial reverse engineering
  - SPM methods require specialised equipment and special sample preparation
  - SEM PVC imaging proved to work down to 180nm
  - backside silicon removal and cleaning



Firmware extraction

- Image processing using Matlab
  - tools to help with image processing
  - histogram to observe distributions of 0s and 1s
  - produces HEX file for testing
  - several samples were processed to produce error-free HEX





Firmware analysis

- HEX Rays IDA Pro supports Freescale HCS08 CPU
  - can produce assembler code listing for further reverse engineering
- Requires configuration file *hcs08.cfg* to correctly name CPU special function registers
  - created using the information found in the directories of the development tools
- Test firmware extraction on real device
  - upload the HEX file into Flash of used Pod and test with PDM
  - video: https://youtu.be/YK6aa4ojl7M
- Debugging the device
  - Flash memory can be downloaded after certain operation to compare with original image
  - development tools allow insertion of conditional breakpoints

Limitations and improvements

- The PVC SEM imaging works well for EEPROM (2T) fabricated with 180nm process and Flash (1T) with 250nm, but too challenging for smaller features
  - 0.35µm Flash: >50,000e$^-$, 16nm Flash: <50e$^-$
  - thin dielectric: <4nm
  - high beam energy: >500eV
  - large memory size: MB vs kB in old devices
  - more efficient methods and solutions are necessary
- Proper memory encryption will pose additional challenges for finding keys and decryption algorithm
- Total lack of documentation will likely make the analysis too challenging with the need for silicon level reverse engineering

Future Work and Collaboration

- More extensive involvement with Failure Analysis methods
  - need more interdisciplinary research
  - make improvements to existing SPM and SEM methods
- Need for closer collaboration between industry and academia
  - test innovative ideas (sometime non-standard and crazy)
- Collaboration with industry
  - bring new ideas and test new methods
  - funding is essential, but it might be possible to go beyond state-of-the-art
- New methods in direct imaging of embedded memory
  - combined methods did work for semi-invasive techniques so should do for invasive
  - more research and development is needed to find new innovative solutions
  - Work-in-Progress webpage for latest breakthrough news: http://www.cl.cam.ac.uk/~sps32/dec_proj.html

Conclusion

- There are many interesting targets for security research
- It is not always true that the latest devices on the market have the best hardware security features
  - you never know what strategy was chosen by the manufacturer
- Do not always take for granted what someone BIG tells you
  - especially when it comes to security and money
- Do not be naive and always try to challenge things if they look odd
  - innovations can drive many aspects, not only the silicon design
- Even if something sounds “impossible” there could be a way
  - “When the Lord closes a door, somewhere He opens a window.” (The Sound of Music, 1965)
- Be creative and innovative!