Weighing in on Issues with “Cloud Scale”

Michael Coppola

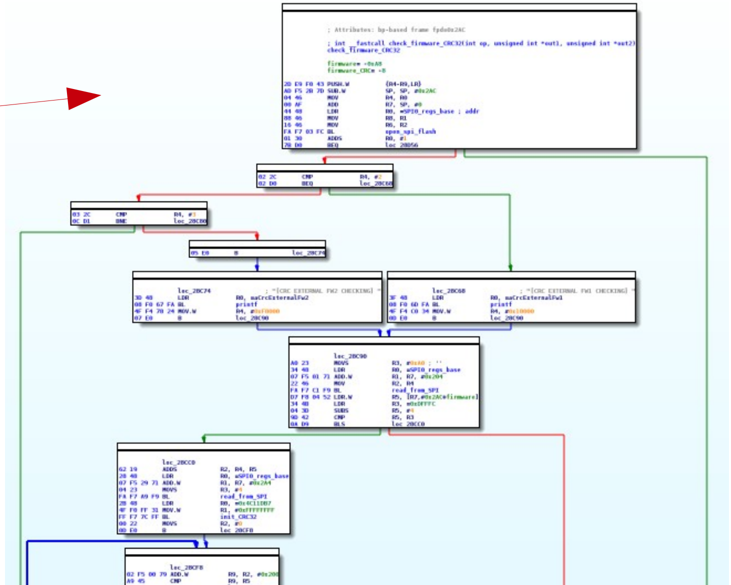
Ekoparty 2013

Who am I?

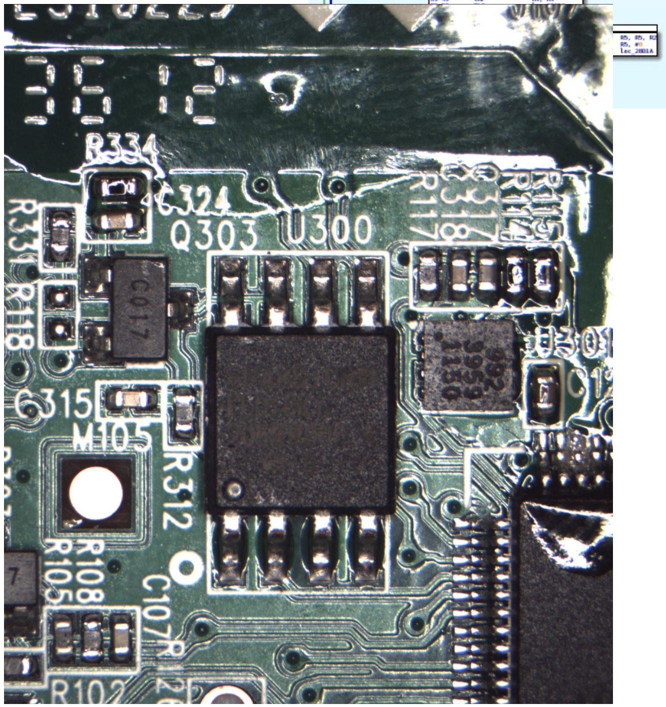
- #
- Student at Northeastern University, USA
- CTF every now and then
- http://poppopret.org/

In this presentation

- A bit of this
- A bit of that
- Successes, failures
- Tool development
- Mad t3chn1quez



```
; Attributes: bp-based frame fpd=0x2AC

; int __fastcall check_firmware_CRC32(int op, unsigned int *out1, unsigned int *out2)
check_firmware_CRC32

firmware= -0xA8
firmware_CRC= -8

2D E9 F0 43 PUSH.W      {R4-R9,LR}
AD F5 2B 7D SUB.W       SP, SP, #0x2AC
04 46       MOV         R4, R0
00 AF       ADD         R7, SP, #0
44 48       LDR         R0, =SPI0_regs_base ; addr
88 46       MOV         R8, R1
16 46       MOV         R6, R2
FA F7 03 FC BL          open_spi_flash
01 30       ADDS        R0, #1
78 D0       BEQ         loc_28D56

02 2C       CMP         R4, #2
02 D0       BEQ         loc_28C68

03 2C       CMP         R4, #3
0C D1       BNE         loc_28C80

05 E0       B           loc_28C74

            loc_28C74               ; "[CRC EXTERNAL FW2 CHECKING] "
3D 48       LDR         R0, =aCrcExternalFw2
08 F0 67 FA BL          printf
4F F4 70 24 MOV.W       R4, #0xF0000
07 E0       B           loc_28C90

            loc_28C68               ; "[CRC EXTERNAL FW1 CHECKING] "
3F 48       LDR         R0, =aCrcExternalFw1
08 F0 6D FA BL          printf
4F F4 C0 34 MOV.W       R4, #0x10000
0D E0       B           loc_28C90

            loc_28C90
A0 23       MOVS        R3, #0xA0 ; ' '
34 48       LDR         R0, =SPI0_regs_base
07 F5 01 71 ADD.W       R1, R7, #0x204
22 46       MOV         R2, R4
FA F7 C1 F9 BL          read_from_SPI
D7 F8 04 52 LDR.W       R5, [R7,#0x2AC+firmware]
34 4B       LDR         R3, =0xDFFFC
04 3D       SUBS        R5, #4
9D 42       CMP         R5, R3
0A D9       BLS         loc_28CC0

            loc_28CC0
62 19       ADDS        R2, R4, R5
28 48       LDR         R0, =SPI0_regs_base
07 F5 29 71 ADD.W       R1, R7, #0x2A4
04 23       MOVS        R3, #4
FA F7 A9 F9 BL          read_from_SPI
28 48       LDR         R0, =0x4C11DB7
4F F0 FF 31 MOV.W       R1, #0xFFFFFFFF
FF F7 7C FF BL          init_CRC32
00 22       MOVS        R2, #0
0D E0       B           loc_28CF8

            loc_28CF8
02 F5 00 79 ADD.W       R9, R2, #0x200
A9 45       CMP         R9, R5

                        R5, R5, R2
                        R5, #0
                        loc_28D1A
```

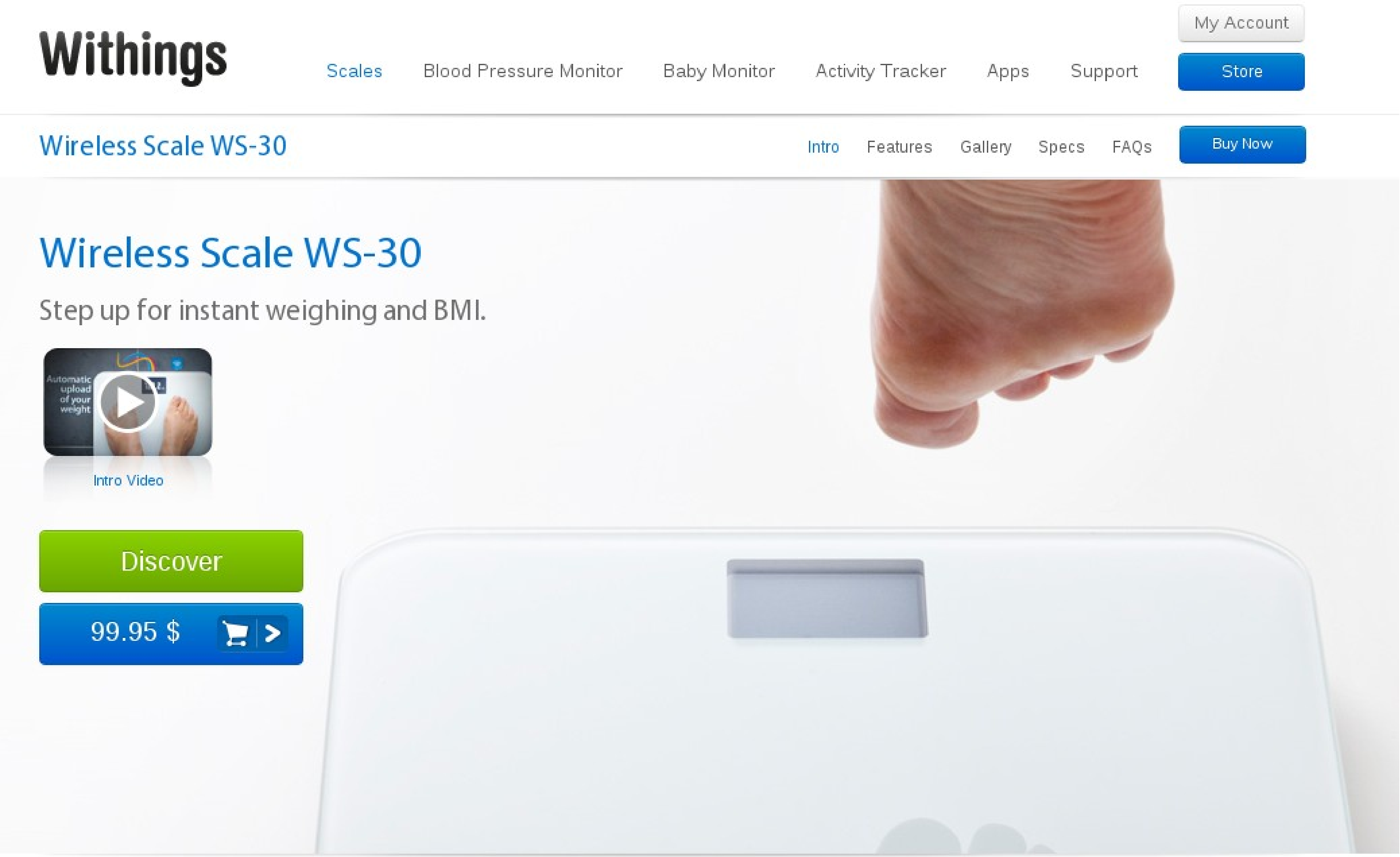


Cloud scaling

+

OMG CLOUD

=



Wireless Scale WS-30

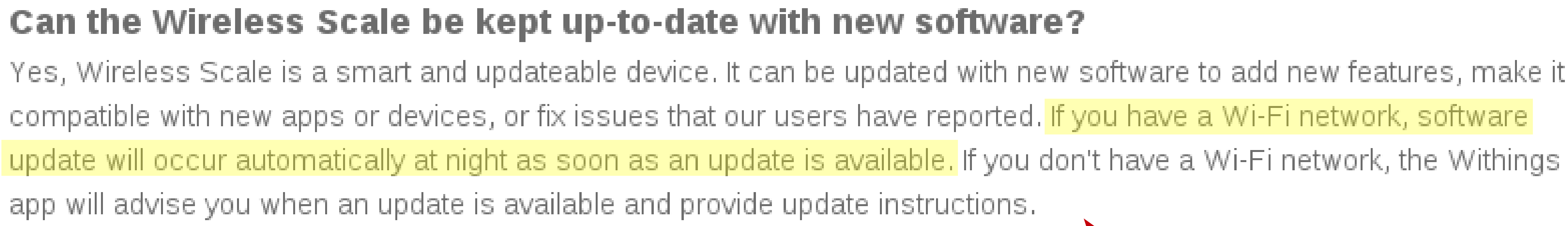
Step up for instant weighing and BMI.

UNITED STATES CYBER COMMAND
9ec4c12949a4f31474f299058ce2b22a

Attack surface

- WiFi / Bluetooth driver and application
- Network communications
- Application input parsing
- No network services (no open ports)

Firmware updates



Can the Wireless Scale be kept up-to-date with new software?

Yes, Wireless Scale is a smart and updateable device. It can be updated with new software to add new features, make it compatible with new apps or devices, or fix issues that our users have reported. If you have a Wi-Fi network, software update will occur automatically at night as soon as an update is available. If you don't have a Wi-Fi network, the Withings app will advise you when an update is available and provide update instructions.

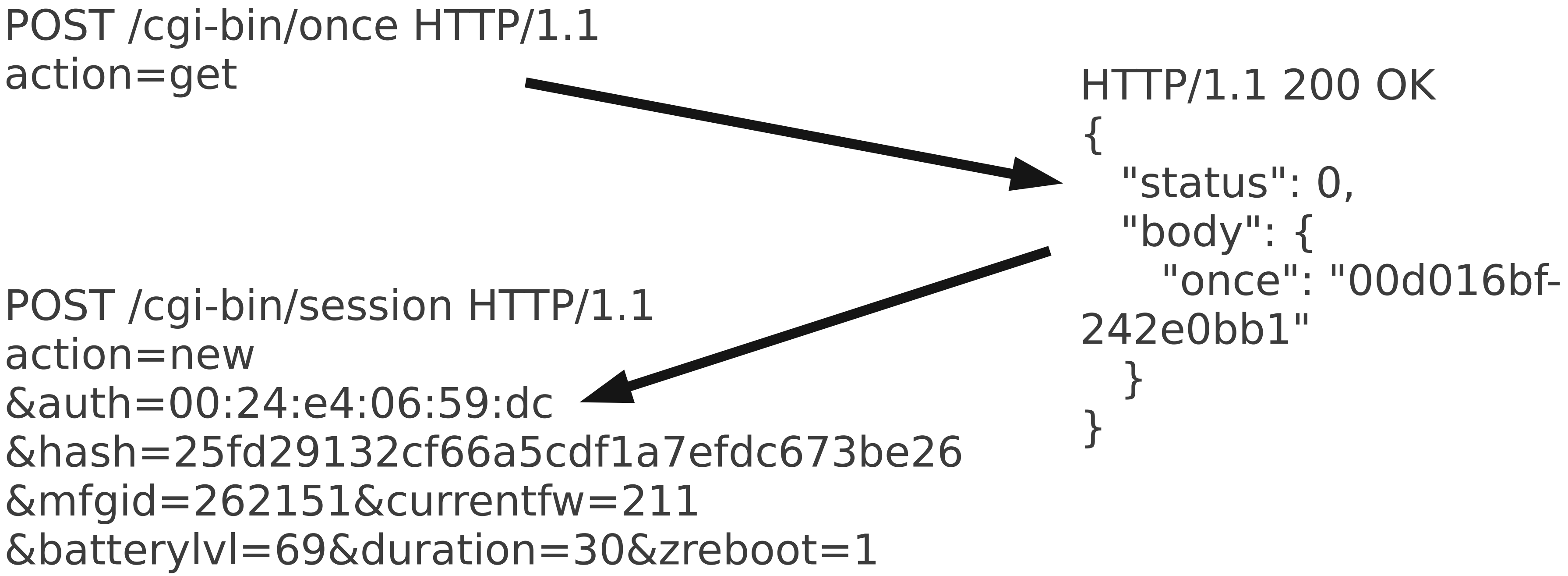
99% chance this was implemented horribly

Sniffing network traffic

- Associate to WiFi using config from Bluetooth
- DNS lookup for scalews.withings.net
- JSON-based protocol over plaintext HTTP

Sniffing network traffic

- Challenge-handshake authentication
- Send device info (MAC, fw version, battery)



```
POST /cgi-bin/once HTTP/1.1
action=get
```

```
HTTP/1.1 200 OK
{
   "status": 0,
   "body": {
      "once": "00d016bf-
242e0bb1"
   }
}
```

```
POST /cgi-bin/session HTTP/1.1
action=new
&auth=00:24:e4:06:59:dc
&hash=25fd29132cf66a5cdf1a7efdc673be26
&mfgid=262151&currentfw=211
&batterylvl=69&duration=30&zreboot=1
```

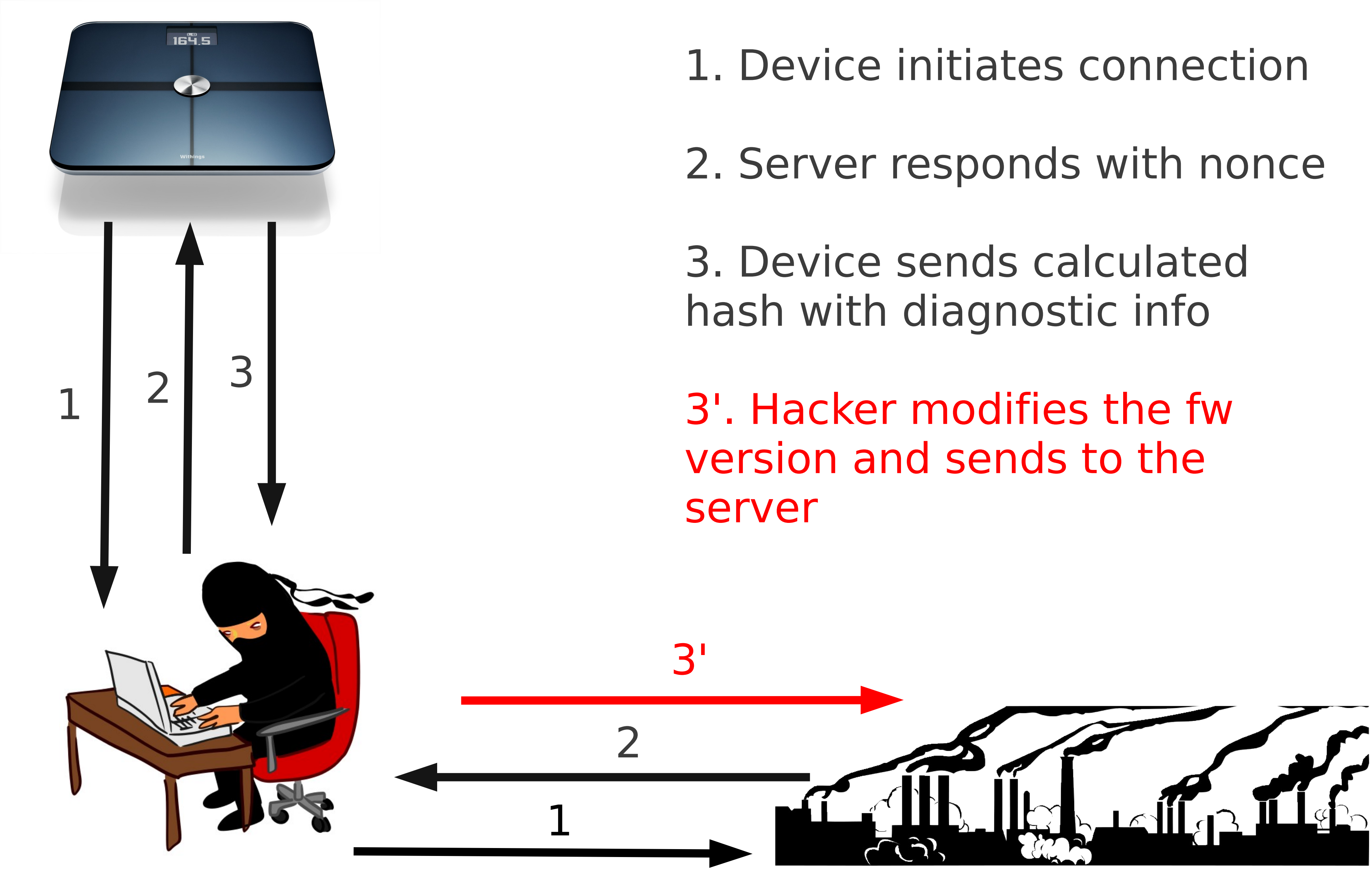
MITM'ing network traffic

- We want the firmware image
  - Maybe sending a lesser fw version will initiate an update

- We don't know how to complete the handshake, so we still need the device

- DNS spoof the device, interpose ourselves in the session

DNS spoofing the handshake



1. Device initiates connection

2. Server responds with nonce

3. Device sends calculated hash with diagnostic info

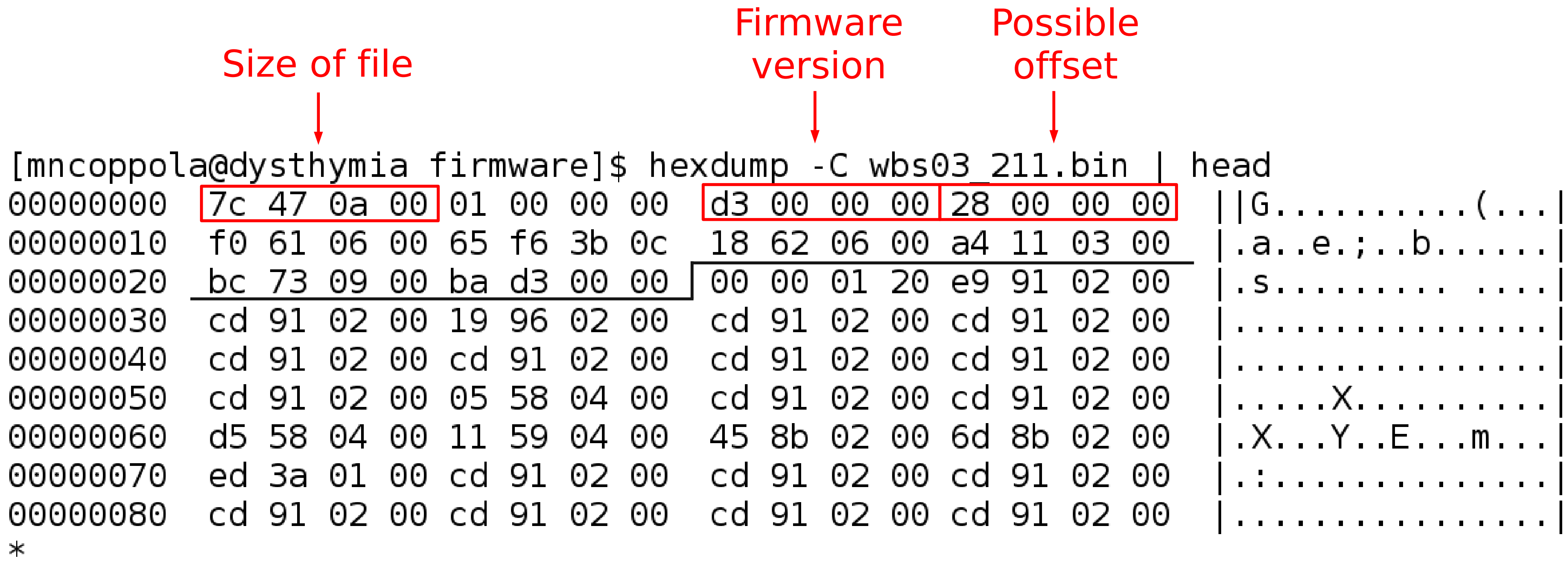
3'. Hacker modifies the fw version and sends to the server

The response

{"status":0,"body":{"sessionid":"8051-51492c4d-730e4ff3","sp":
{"users":[]},"ind":{"lg":"en_GB","imt":1,"stp":1,"f":0,"g":97918},"syp":
{"blc":"http:\/\/fw.withings.net\/wbs03_211.bin","utc":1363749965},"ctp"
:{"goff":-14400,"dst":0,"ngoff":0}}}

Firmware header

- No results from binwalk
- Lots of strings → likely no encryption or compression
- Multiple null padded sections → likely multiple objects packaged together



Size of file

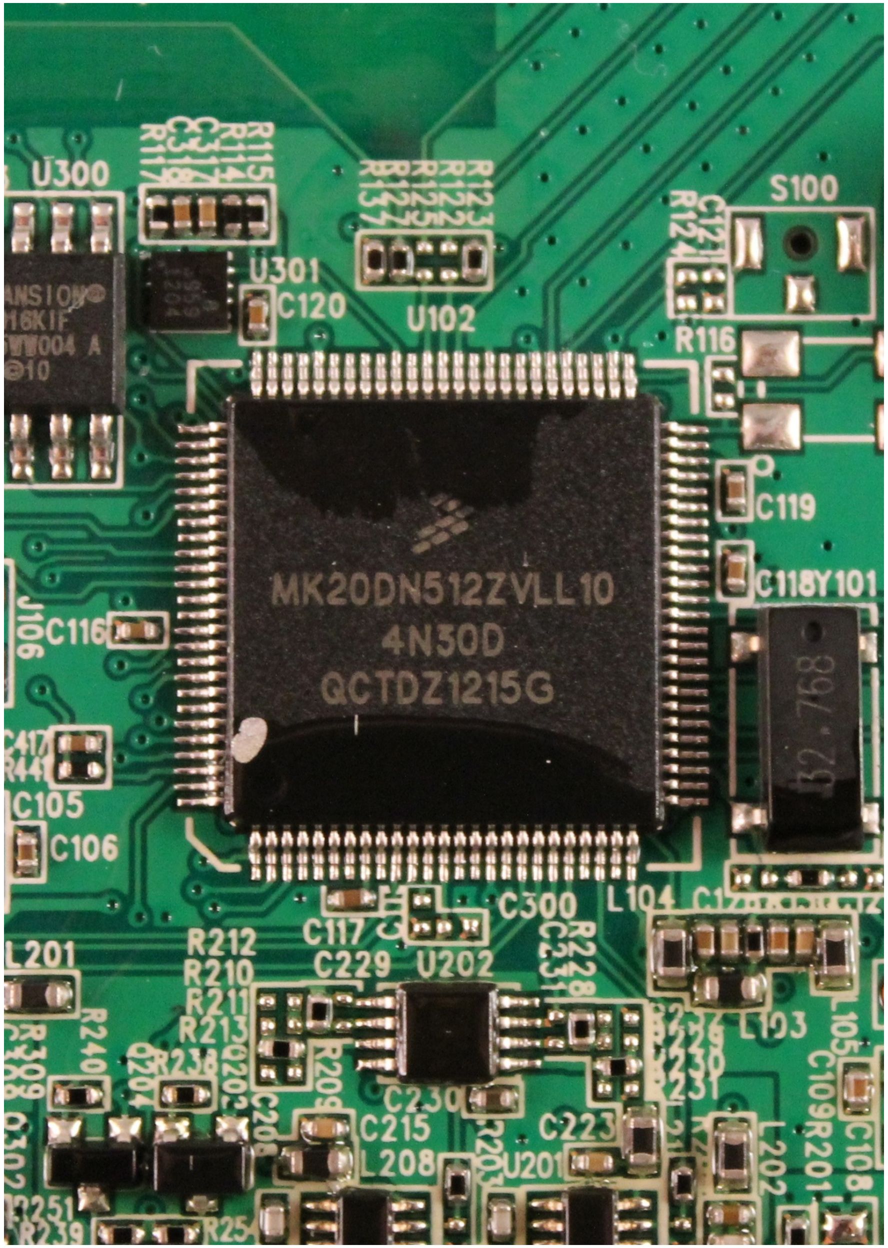
Firmware version

Possible offset

```
[mncoppola@dysthymia firmware]$ hexdump -C wbs03_211.bin | head
00000000  7c 47 0a 00 01 00 00 00  d3 00 00 00 28 00 00 00  ||G..........(...|
00000010  f0 61 06 00 65 f6 3b 0c  18 62 06 00 a4 11 03 00  |.a..e.;..b......|
00000020  bc 73 09 00 ba d3 00 00  00 00 01 20 e9 91 02 00  |.s......... ....|
00000030  cd 91 02 00 19 96 02 00  cd 91 02 00 cd 91 02 00  |................|
00000040  cd 91 02 00 cd 91 02 00  cd 91 02 00 cd 91 02 00  |................|
00000050  cd 91 02 00 05 58 04 00  cd 91 02 00 cd 91 02 00  |.....X..........|
00000060  d5 58 04 00 11 59 04 00  45 8b 02 00 6d 8b 02 00  |.X...Y..E...m...|
00000070  ed 3a 01 00 cd 91 02 00  cd 91 02 00 cd 91 02 00  |.:..............|
00000080  cd 91 02 00 cd 91 02 00  cd 91 02 00 cd 91 02 00  |................|
*
```

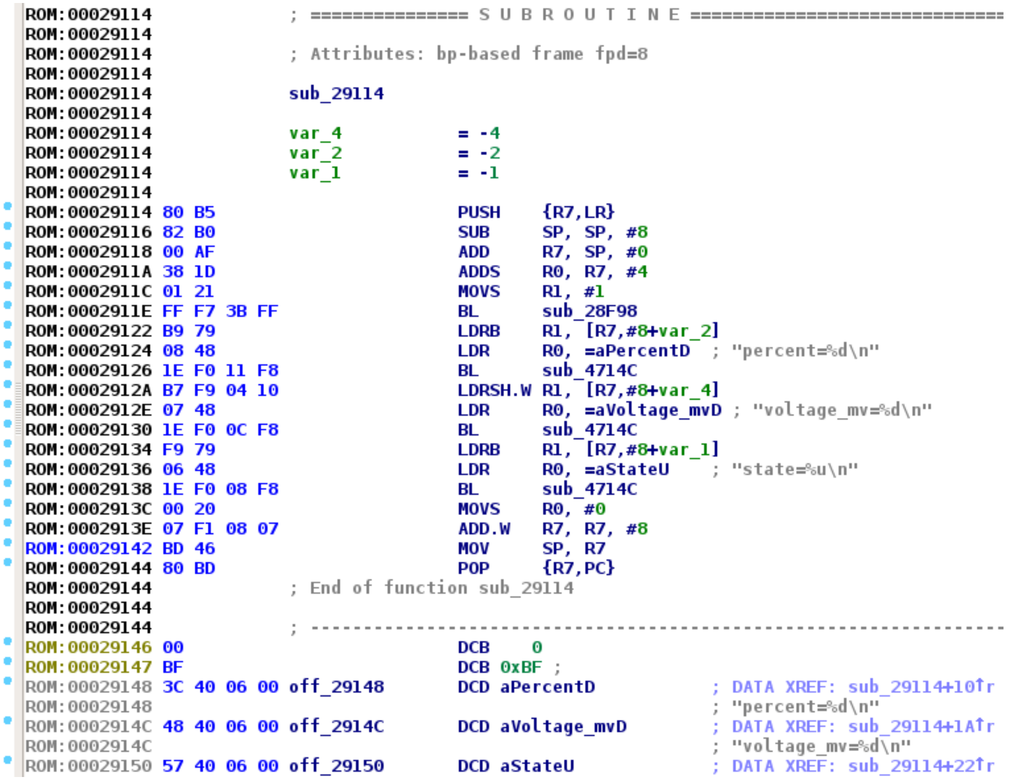
Identifying the MCU

- MK20DN512ZVLL10
- Freescale Kinetis K20 family
- ARM Cortex-M4 (ARMv7)
- Memory-mapped peripheral registers

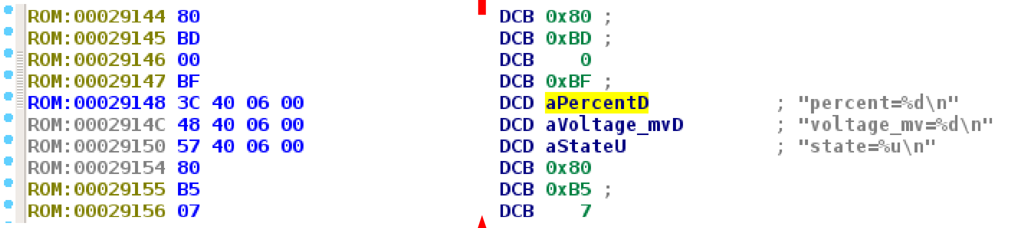


Locating code blocks

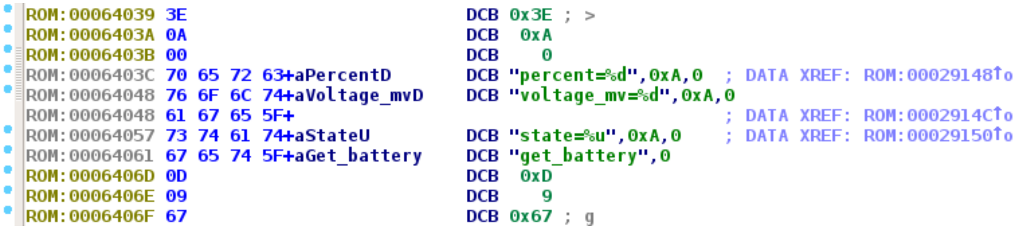
- Find a dense area of bytes and start disassembling
- Common bytes:
  - ARM: 0xe*
  - Padding: 0xbf00 (nop)
- Byte search the addresses of strings and disassemble backwards



```
ROM:00029114                          ; =============== S U B R O U T I N E ==============================
ROM:00029114
ROM:00029114                          ; Attributes: bp-based frame fpd=8
ROM:00029114
ROM:00029114                          sub_29114
ROM:00029114
ROM:00029114                          var_4           = -4
ROM:00029114                          var_2           = -2
ROM:00029114                          var_1           = -1
ROM:00029114
ROM:00029114 80 B5                                    PUSH    {R7,LR}
ROM:00029116 82 B0                                    SUB     SP, SP, #8
ROM:00029118 00 AF                                    ADD     R7, SP, #0
ROM:0002911A 38 1D                                    ADDS    R0, R7, #4
ROM:0002911C 01 21                                    MOVS    R1, #1
ROM:0002911E FF F7 3B FF                              BL      sub_28F98
ROM:00029122 B9 79                                    LDRB    R1, [R7,#8+var_2]
ROM:00029124 08 48                                    LDR     R0, =aPercentD  ; "percent=%d\n"
ROM:00029126 1E F0 11 F8                              BL      sub_4714C
ROM:0002912A B7 F9 04 10                              LDRSH.W R1, [R7,#8+var_4]
ROM:0002912E 07 48                                    LDR     R0, =aVoltage_mvD ; "voltage_mv=%d\n"
ROM:00029130 1E F0 0C F8                              BL      sub_4714C
ROM:00029134 F9 79                                    LDRB    R1, [R7,#8+var_1]
ROM:00029136 06 48                                    LDR     R0, =aStateU    ; "state=%u\n"
ROM:00029138 1E F0 08 F8                              BL      sub_4714C
ROM:0002913C 00 20                                    MOVS    R0, #0
ROM:0002913E 07 F1 08 07                              ADD.W   R7, R7, #8
ROM:00029142 BD 46                                    MOV     SP, R7
ROM:00029144 80 BD                                    POP     {R7,PC}
ROM:00029144                          ; End of function sub_29114
ROM:00029144
ROM:00029144                          ; ---------------------------------------------------------------
ROM:00029146 00                                       DCB    0
ROM:00029147 BF                                       DCB 0xBF ;
ROM:00029148 3C 40 06 00 off_29148                    DCD aPercentD           ; DATA XREF: sub_29114+10↑r
ROM:00029148                                                                  ; "percent=%d\n"
ROM:0002914C 48 40 06 00 off_2914C                    DCD aVoltage_mvD        ; DATA XREF: sub_29114+1A↑r
ROM:0002914C                                                                  ; "voltage_mv=%d\n"
ROM:00029150 57 40 06 00 off_29150                    DCD aStateU             ; DATA XREF: sub_29114+22↑r
```

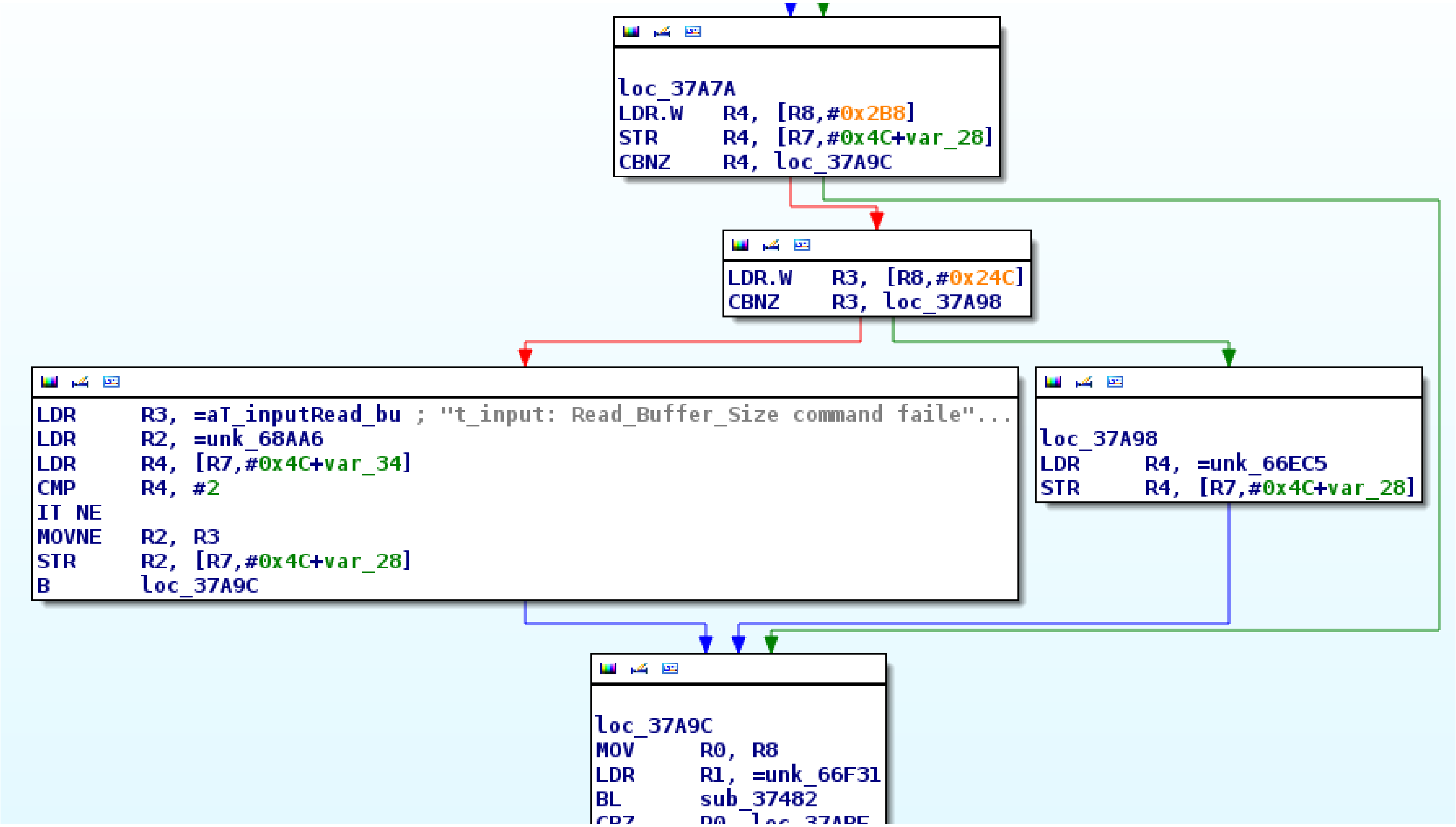


```
ROM:00029144 80                             DCB 0x80 ;
ROM:00029145 BD                             DCB 0xBD ;
ROM:00029146 00                             DCB    0
ROM:00029147 BF                             DCB 0xBF ;
ROM:00029148 3C 40 06 00                    DCD aPercentD           ; "percent=%d\n"
ROM:0002914C 48 40 06 00                    DCD aVoltage_mvD        ; "voltage_mv=%d\n"
ROM:00029150 57 40 06 00                    DCD aStateU             ; "state=%u\n"
ROM:00029154 80                             DCB 0x80
ROM:00029155 B5                             DCB 0xB5 ;
ROM:00029156 07                             DCB    7
```



```
ROM:00064039 3E                         DCB 0x3E ; >
ROM:0006403A 0A                         DCB  0xA
ROM:0006403B 00                         DCB    0
ROM:0006403C 70 65 72 63+aPercentD      DCB "percent=%d",0xA,0  ; DATA XREF: ROM:00029148↑o
ROM:00064048 76 6F 6C 74+aVoltage_mvD   DCB "voltage_mv=%d",0xA,0
ROM:00064048 61 67 65 5F+                                       ; DATA XREF: ROM:0002914C↑o
ROM:00064057 73 74 61 74+aStateU        DCB "state=%u",0xA,0    ; DATA XREF: ROM:00029150↑o
ROM:00064061 67 65 74 5F+aGet_battery   DCB "get_battery",0
ROM:0006406D 0D                         DCB  0xD
ROM:0006406E 09                         DCB    9
ROM:0006406F 67                         DCB 0x67 ; g
```

Things aren't lining up...



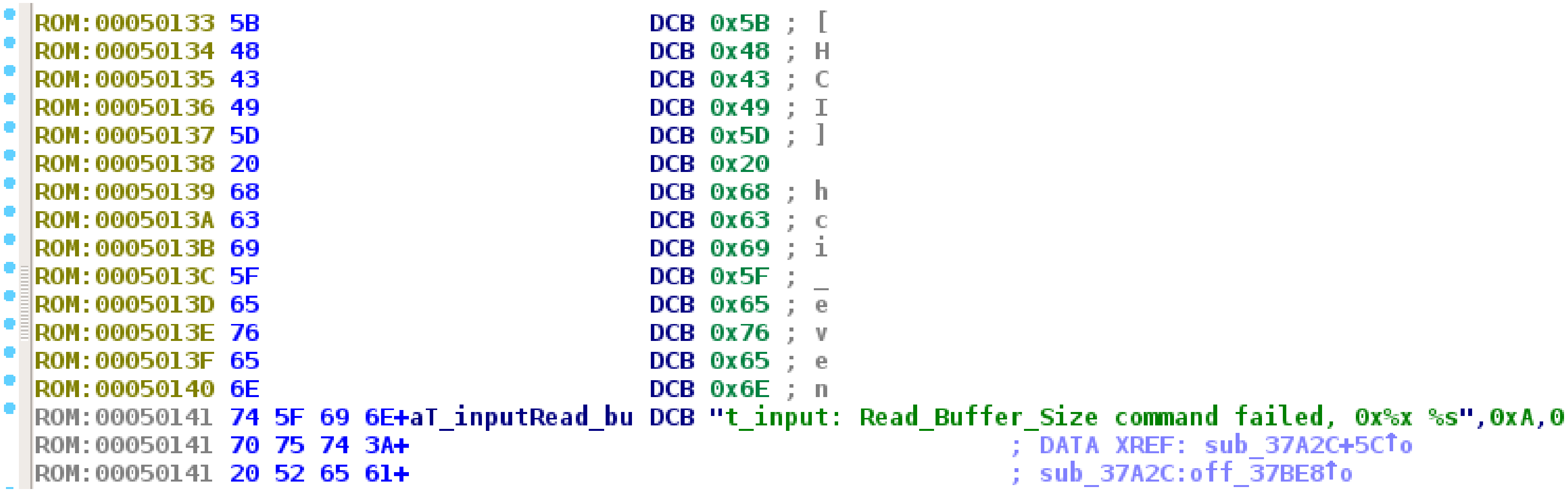
```
loc_37A7A
LDR.W    R4, [R8,#0x2B8]
STR      R4, [R7,#0x4C+var_28]
CBNZ     R4, loc_37A9C
```

```
LDR.W    R3, [R8,#0x24C]
CBNZ     R3, loc_37A98
```

```
LDR      R3, =aT_inputRead_bu ; "t_input: Read_Buffer_Size command faile"...
LDR      R2, =unk_68AA6
LDR      R4, [R7,#0x4C+var_34]
CMP      R4, #2
IT NE
MOVNE    R2, R3
STR      R2, [R7,#0x4C+var_28]
B        loc_37A9C
```

```
loc_37A98
LDR      R4, =unk_66EC5
STR      R4, [R7,#0x4C+var_28]
```

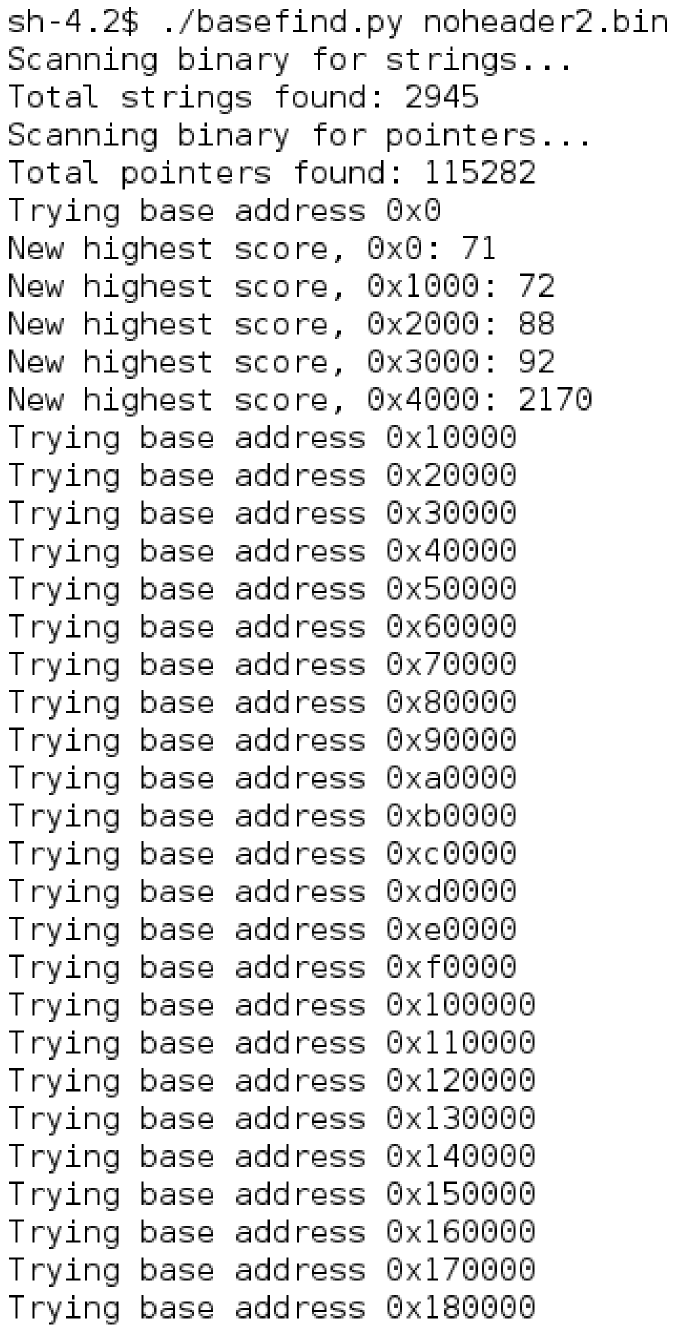
```
loc_37A9C
MOV      R0, R8
LDR      R1, =unk_66F31
BL       sub_37482
CBZ      R0, loc_37ABE
```



```
ROM:00050133 5B                          DCB 0x5B ; [
ROM:00050134 48                          DCB 0x48 ; H
ROM:00050135 43                          DCB 0x43 ; C
ROM:00050136 49                          DCB 0x49 ; I
ROM:00050137 5D                          DCB 0x5D ; ]
ROM:00050138 20                          DCB 0x20
ROM:00050139 68                          DCB 0x68 ; h
ROM:0005013A 63                          DCB 0x63 ; c
ROM:0005013B 69                          DCB 0x69 ; i
ROM:0005013C 5F                          DCB 0x5F ; _
ROM:0005013D 65                          DCB 0x65 ; e
ROM:0005013E 76                          DCB 0x76 ; v
ROM:0005013F 65                          DCB 0x65 ; e
ROM:00050140 6E                          DCB 0x6E ; n
ROM:00050141 74 5F 69 6E+aT_inputRead_bu DCB "t_input: Read_Buffer_Size command failed, 0x%x %s",0xA,0
ROM:00050141 70 75 74 3A+                                        ; DATA XREF: sub_37A2C+5C↑o
ROM:00050141 20 52 65 61+                                        ; sub_37A2C:off_37BE8↑o
```

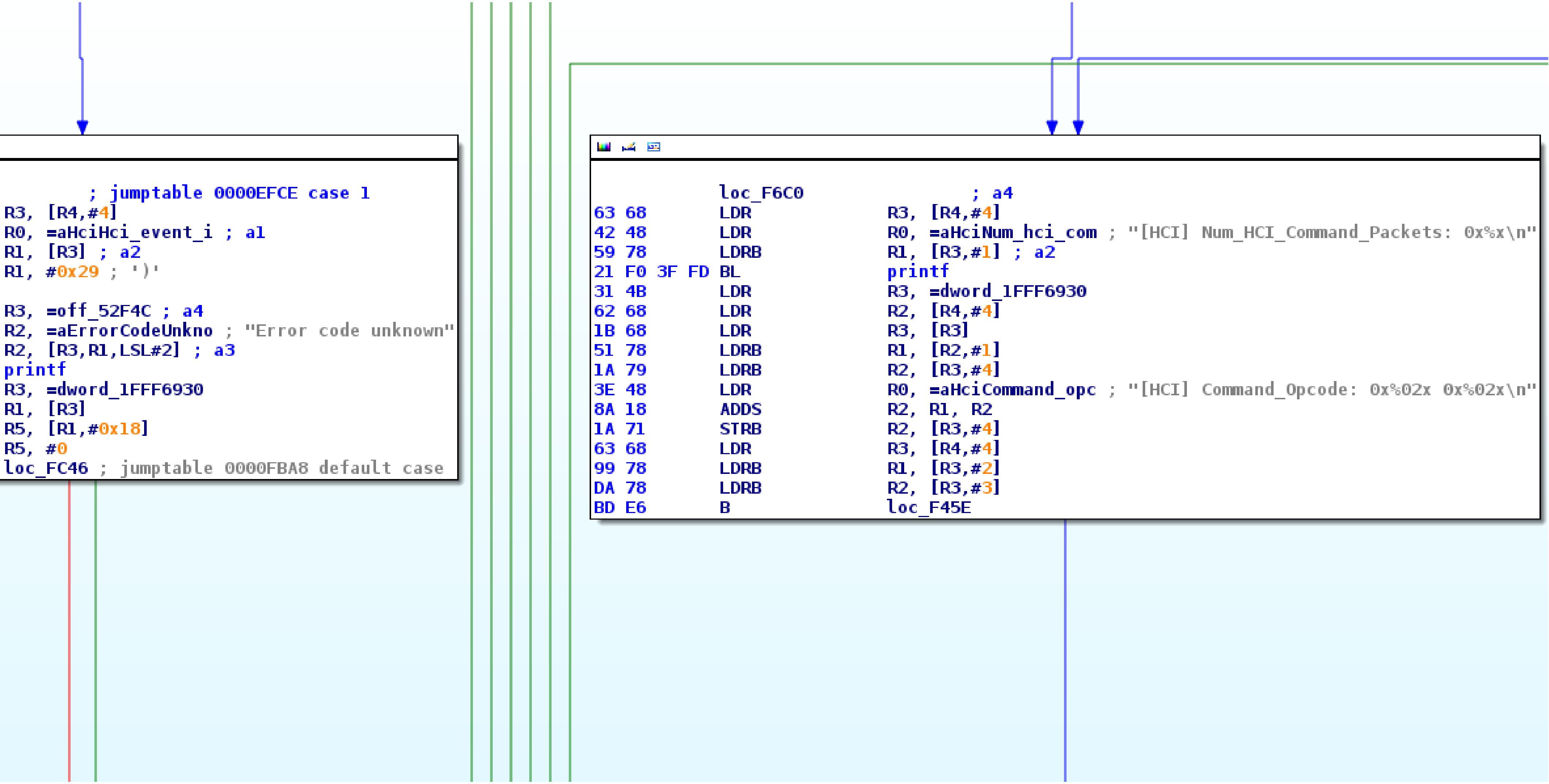
basefind.py

- Every dword in file is treated as a pointer
- Does base + dword point to the beginning of a string?
- Repeat for all possible base addresses
- Highest score is likely the correct base address



```
sh-4.2$ ./basefind.py noheader2.bin
Scanning binary for strings...
Total strings found: 2945
Scanning binary for pointers...
Total pointers found: 115282
Trying base address 0x0
New highest score, 0x0: 71
New highest score, 0x1000: 72
New highest score, 0x2000: 88
New highest score, 0x3000: 92
New highest score, 0x4000: 2170
Trying base address 0x10000
Trying base address 0x20000
Trying base address 0x30000
Trying base address 0x40000
Trying base address 0x50000
Trying base address 0x60000
Trying base address 0x70000
Trying base address 0x80000
Trying base address 0x90000
Trying base address 0xa0000
Trying base address 0xb0000
Trying base address 0xc0000
Trying base address 0xd0000
Trying base address 0xe0000
Trying base address 0xf0000
Trying base address 0x100000
Trying base address 0x110000
Trying base address 0x120000
Trying base address 0x130000
Trying base address 0x140000
Trying base address 0x150000
Trying base address 0x160000
Trying base address 0x170000
Trying base address 0x180000
```

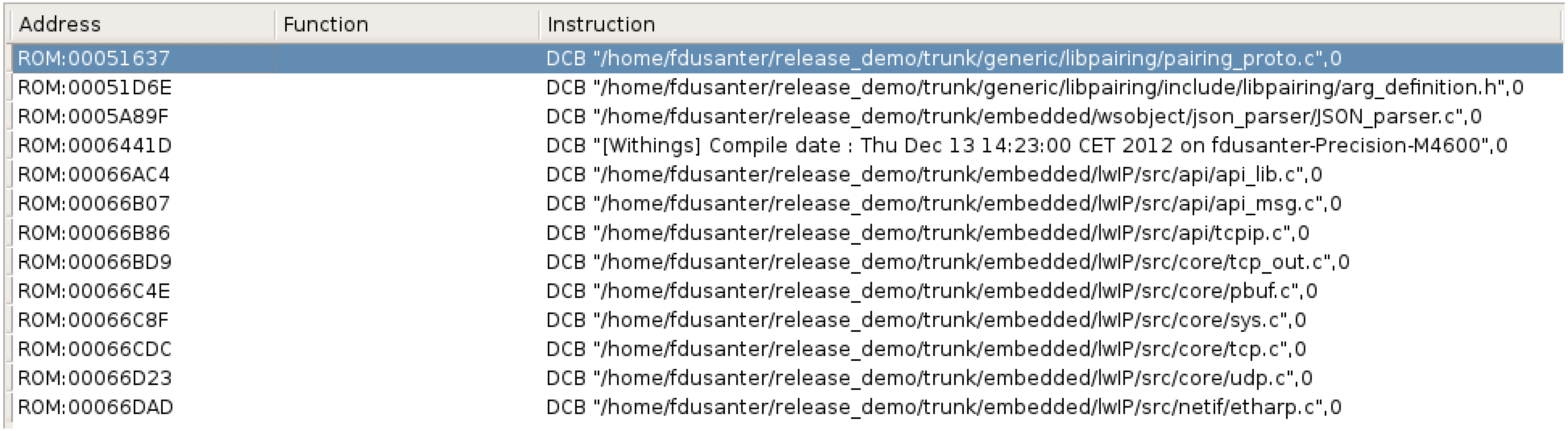
Base address 0x4000



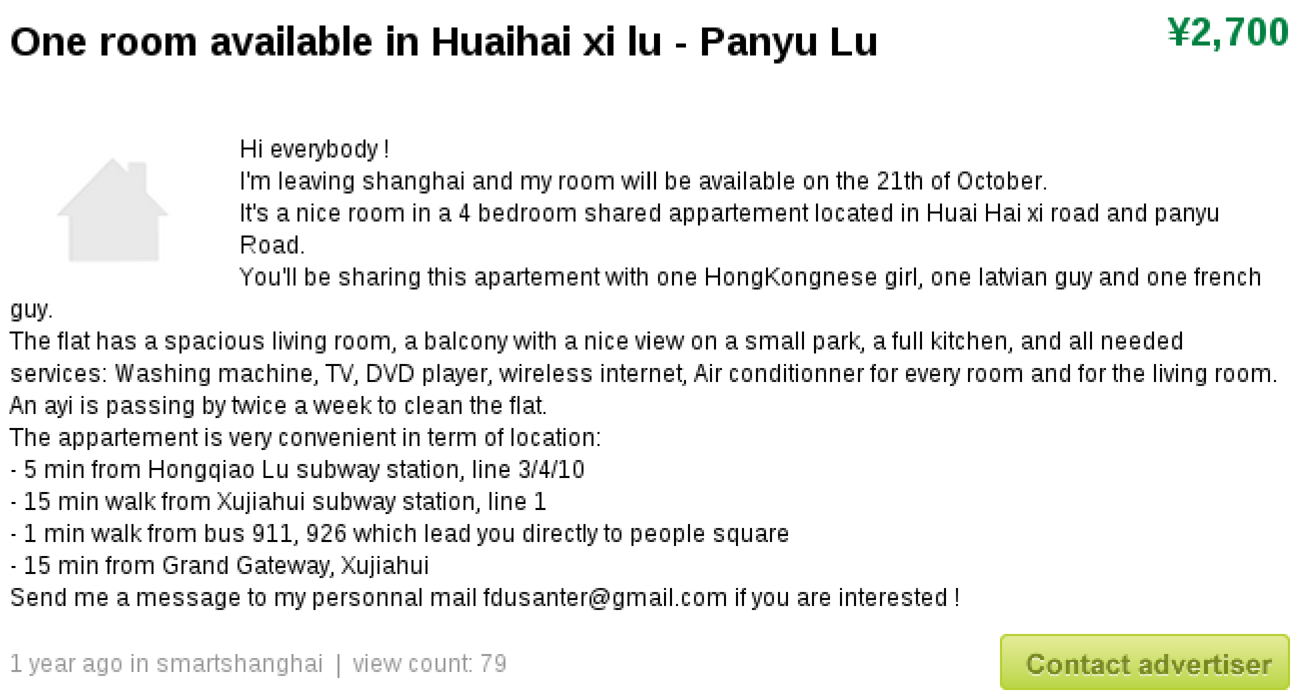
```
      ; jumptable 0000EFCE case 1
R3, [R4,#4]
R0, =aHciHci_event_i ; a1
R1, [R3] ; a2
R1, #0x29 ; ')'

R3, =off_52F4C ; a4
R2, =aErrorCodeUnkno ; "Error code unknown"
R2, [R3,R1,LSL#2] ; a3
printf
R3, =dword_1FFF6930
R1, [R3]
R5, [R1,#0x18]
R5, #0
loc_FC46 ; jumptable 0000FBA8 default case
```

```
             loc_F6C0                ; a4
63 68        LDR             R3, [R4,#4]
42 48        LDR             R0, =aHciNum_hci_com ; "[HCI] Num_HCI_Command_Packets: 0x%x\n"
59 78        LDRB            R1, [R3,#1] ; a2
21 F0 3F FD  BL              printf
31 4B        LDR             R3, =dword_1FFF6930
62 68        LDR             R2, [R4,#4]
1B 68        LDR             R3, [R3]
51 78        LDRB            R1, [R2,#1]
1A 79        LDRB            R2, [R3,#4]
3E 48        LDR             R0, =aHciCommand_opc ; "[HCI] Command_Opcode: 0x%02x 0x%02x\n"
8A 18        ADDS            R2, R1, R2
1A 71        STRB            R2, [R3,#4]
63 68        LDR             R3, [R4,#4]
99 78        LDRB            R1, [R3,#2]
DA 78        LDRB            R2, [R3,#3]
BD E6        B               loc_F45E
```

d0x d0x d0x



Address	Function	Instruction
ROM:00051637		DCB "/home/fdusanter/release_demo/trunk/generic/libpairing/pairing_proto.c",0
ROM:00051D6E		DCB "/home/fdusanter/release_demo/trunk/generic/libpairing/include/libpairing/arg_definition.h",0
ROM:0005A89F		DCB "/home/fdusanter/release_demo/trunk/embedded/wsobject/json_parser/JSON_parser.c",0
ROM:0006441D		DCB "[Withings] Compile date : Thu Dec 13 14:23:00 CET 2012 on fdusanter-Precision-M4600",0
ROM:00066AC4		DCB "/home/fdusanter/release_demo/trunk/embedded/lwIP/src/api/api_lib.c",0
ROM:00066B07		DCB "/home/fdusanter/release_demo/trunk/embedded/lwIP/src/api/api_msg.c",0
ROM:00066B86		DCB "/home/fdusanter/release_demo/trunk/embedded/lwIP/src/api/tcpip.c",0
ROM:00066BD9		DCB "/home/fdusanter/release_demo/trunk/embedded/lwIP/src/core/tcp_out.c",0
ROM:00066C4E		DCB "/home/fdusanter/release_demo/trunk/embedded/lwIP/src/core/pbuf.c",0
ROM:00066C8F		DCB "/home/fdusanter/release_demo/trunk/embedded/lwIP/src/core/sys.c",0
ROM:00066CDC		DCB "/home/fdusanter/release_demo/trunk/embedded/lwIP/src/core/tcp.c",0
ROM:00066D23		DCB "/home/fdusanter/release_demo/trunk/embedded/lwIP/src/core/udp.c",0
ROM:00066DAD		DCB "/home/fdusanter/release_demo/trunk/embedded/lwIP/src/netif/etharp.c",0



One room available in Huaihai xi lu - Panyu Lu ¥2,700

Hi everybody !
I'm leaving shanghai and my room will be available on the 21th of October.
It's a nice room in a 4 bedroom shared appartement located in Huai Hai xi road and panyu Road.
You'll be sharing this apartement with one HongKongnese girl, one latvian guy and one french guy.
The flat has a spacious living room, a balcony with a nice view on a small park, a full kitchen, and all needed services: Washing machine, TV, DVD player, wireless internet, Air conditionner for every room and for the living room.
An ayi is passing by twice a week to clean the flat.
The appartement is very convenient in term of location:
- 5 min from Hongqiao Lu subway station, line 3/4/10
- 15 min walk from Xujiahui subway station, line 1
- 1 min walk from bus 911, 926 which lead you directly to people square
- 15 min from Grand Gateway, Xujiahui
Send me a message to my personnal mail fdusanter@gmail.com if you are interested !

1 year ago in smartshanghai | view count: 79

Firmware subsystems

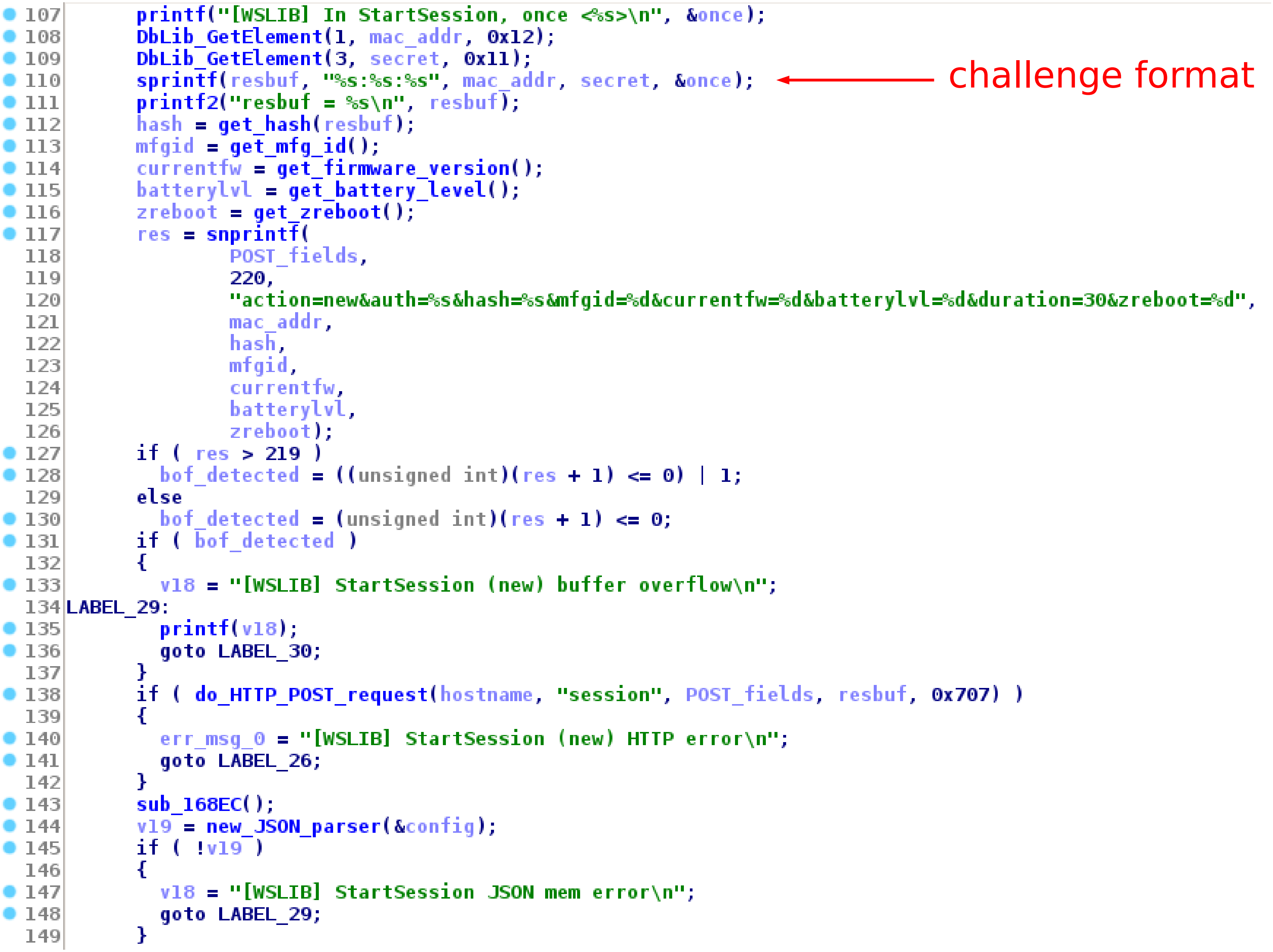
- LibCURL – HTTP library
- lwIP – TCP/IP stack
- DbLib – Key=value store
- WsLib – Interact with Withings web services
- CnLib – Network connection management
- UsLib – User management
- LibPairing – Bluetooth pairing

DbLib_GetElement()

Index	Name	Size
1	mac_address	0x12
2	hostname	0x40
3	secret	0x11
4	time_constant	0xc
5	users	0x280
6	uid	0x26
7	pref_lang	0x6
8	scale_constant	0x10
9	ssid	0x6c
10	IP config	0x16
11	mfg_id	0x4
12	calibration	0x14e
14	last connection	0x20
16		0x2c

Index	Name	Size
17	factory_mode	0x4
19	debug traces	0x0
20	factory weight verif	0x4c
22	wifi_country	0x4
24	wpa_key	0x50
25	battery level	0x8
27	wifi_delay	0x4
28	Bluetooth config	0x16
32		0x4
33	calibration parameters	0x4e
34		0x10
37		0x4
38		0x82

Device association request

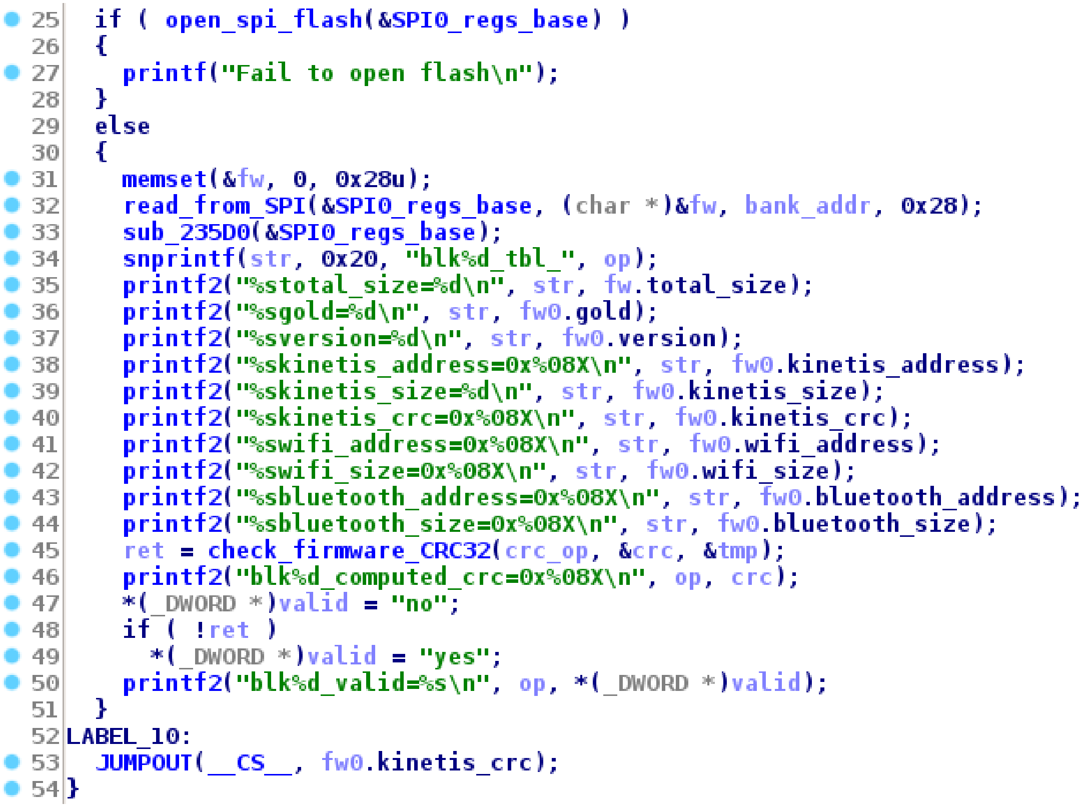


```
      printf("[WSLIB] In StartSession, once <%s>\n", &once);
      DbLib_GetElement(1, mac_addr, 0x12);
      DbLib_GetElement(3, secret, 0x11);
      sprintf(resbuf, "%s:%s:%s", mac_addr, secret, &once);
      printf2("resbuf = %s\n", resbuf);
      hash = get_hash(resbuf);
      mfgid = get_mfg_id();
      currentfw = get_firmware_version();
      batterylvl = get_battery_level();
      zreboot = get_zreboot();
      res = snprintf(
              POST_fields,
              220,
              "action=new&auth=%s&hash=%s&mfgid=%d&currentfw=%d&batterylvl=%d&duration=30&zreboot=%d",
              mac_addr,
              hash,
              mfgid,
              currentfw,
              batterylvl,
              zreboot);
      if ( res > 219 )
        bof_detected = ((unsigned int)(res + 1) <= 0) | 1;
      else
        bof_detected = (unsigned int)(res + 1) <= 0;
      if ( bof_detected )
      {
        v18 = "[WSLIB] StartSession (new) buffer overflow\n";
LABEL_29:
        printf(v18);
        goto LABEL_30;
      }
      if ( do_HTTP_POST_request(hostname, "session", POST_fields, resbuf, 0x707) )
      {
        err_msg_0 = "[WSLIB] StartSession (new) HTTP error\n";
        goto LABEL_26;
      }
      sub_168EC();
      v19 = new_JSON_parser(&config);
      if ( !v19 )
      {
        v18 = "[WSLIB] StartSession JSON mem error\n";
        goto LABEL_29;
      }
```

challenge format

Cracking the firmware header

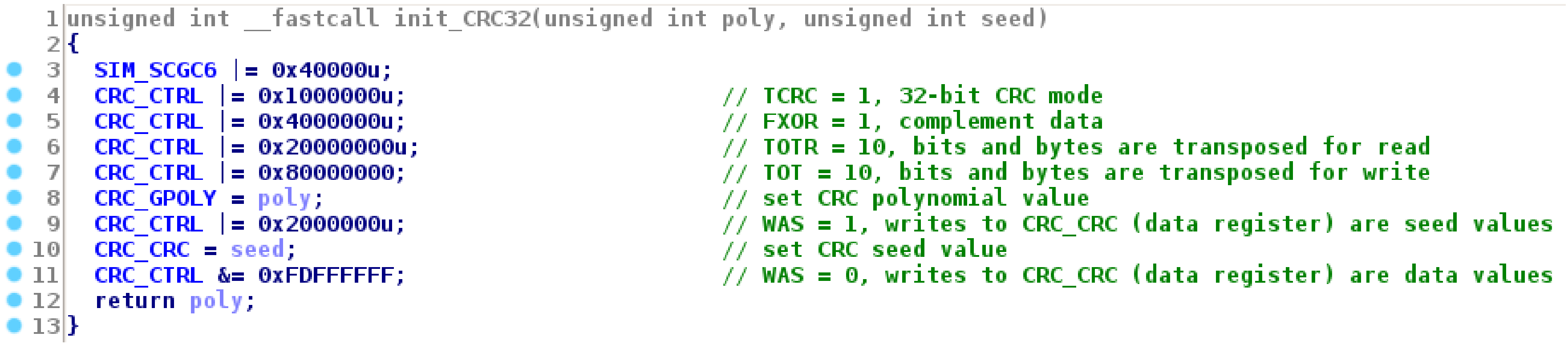
```
[mncoppola@dysthymia firmware]$ hexdump -C wbs03_211.bin | head -n3
00000000  7c 47 0a 00 01 00 00 00  d3 00 00 00 28 00 00 00  ||G.........(...|
00000010  f0 61 06 00 65 f6 3b 0c  18 62 06 00 a4 11 03 00  |.a..e.;..b......|
00000020  bc 73 09 00 ba d3 00 00  00 00 01 20 e9 91 02 00  |.s.......... ....|
[mncoppola@dysthymia firmware]$ hexdump -C wbs03_211.bin | tail -n3
000a4760  d1 1c 60 01 e0 13 68 0b  60 14 60 10 bd 00 00 4e  |..`...h.`.`....N|
000a4770  fc 04 ff ff ff ff ff ff  65 68 f3 4f              |........eh.O|
000a477c
```



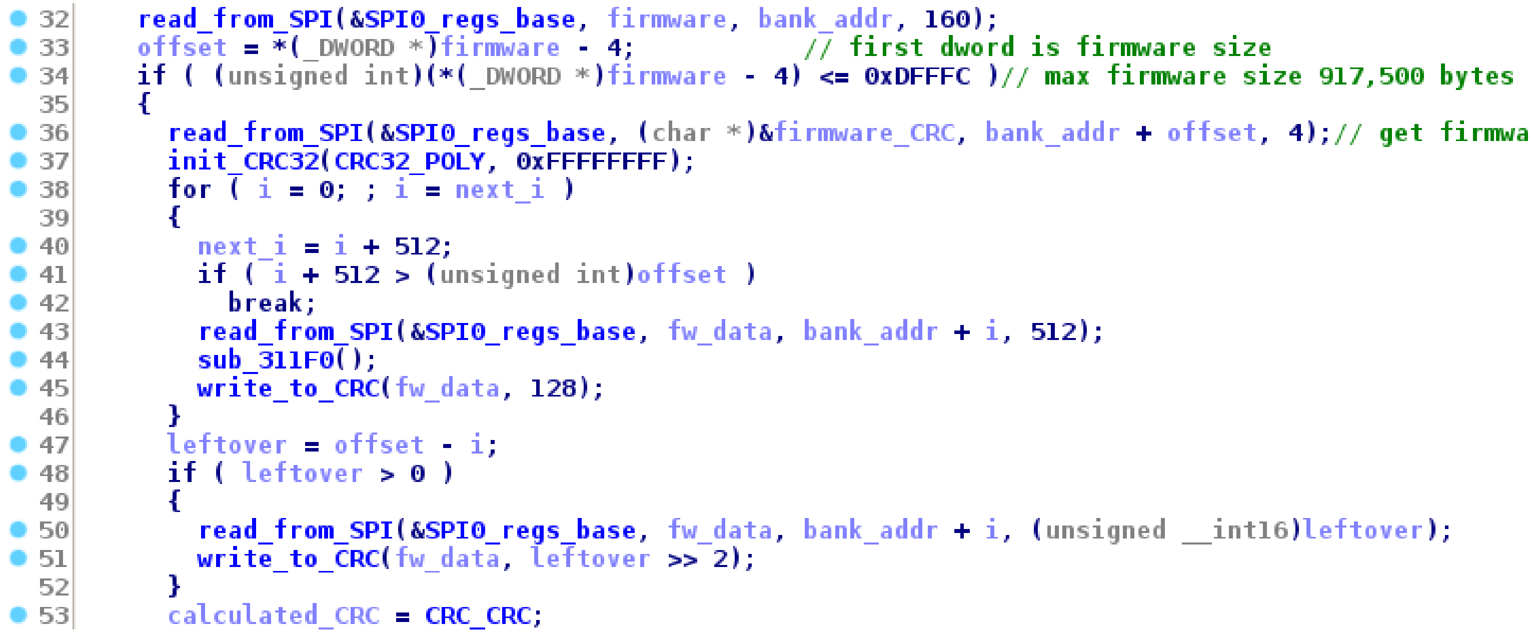
```
  if ( open_spi_flash(&SPI0_regs_base) )
  {
    printf("Fail to open flash\n");
  }
  else
  {
    memset(&fw, 0, 0x28u);
    read_from_SPI(&SPI0_regs_base, (char *)&fw, bank_addr, 0x28);
    sub_235D0(&SPI0_regs_base);
    snprintf(str, 0x20, "blk%d_tbl_", op);
    printf2("%stotal_size=%d\n", str, fw.total_size);
    printf2("%sgold=%d\n", str, fw0.gold);
    printf2("%sversion=%d\n", str, fw0.version);
    printf2("%skinetis_address=0x%08X\n", str, fw0.kinetis_address);
    printf2("%skinetis_size=%d\n", str, fw0.kinetis_size);
    printf2("%skinetis_crc=0x%08X\n", str, fw0.kinetis_crc);
    printf2("%swifi_address=0x%08X\n", str, fw0.wifi_address);
    printf2("%swifi_size=0x%08X\n", str, fw0.wifi_size);
    printf2("%sbluetooth_address=0x%08X\n", str, fw0.bluetooth_address);
    printf2("%sbluetooth_size=0x%08X\n", str, fw0.bluetooth_size);
    ret = check_firmware_CRC32(crc_op, &crc, &tmp);
    printf2("blk%d_computed_crc=0x%08X\n", op, crc);
    *(_DWORD *)valid = "no";
    if ( !ret )
      *(_DWORD *)valid = "yes";
    printf2("blk%d_valid=%s\n", op, *(_DWORD *)valid);
  }
LABEL_10:
  JUMPOUT(__CS__, fw0.kinetis_crc);
}
```

Total size: 673660 bytes
Gold status: 0x1
Firmware version: 211
Kinetis address: 0x28
Kinetis size: 418288 bytes
Kinetis CRC: 0x0c3bf665
WiFi address: 0x66218
WiFi size: 201124 bytes
Bluetooth address: 0x973bc
Bluetooth size: 54202 bytes
Firmware CRC: 0x4ff36865

Reversing the CRC validation

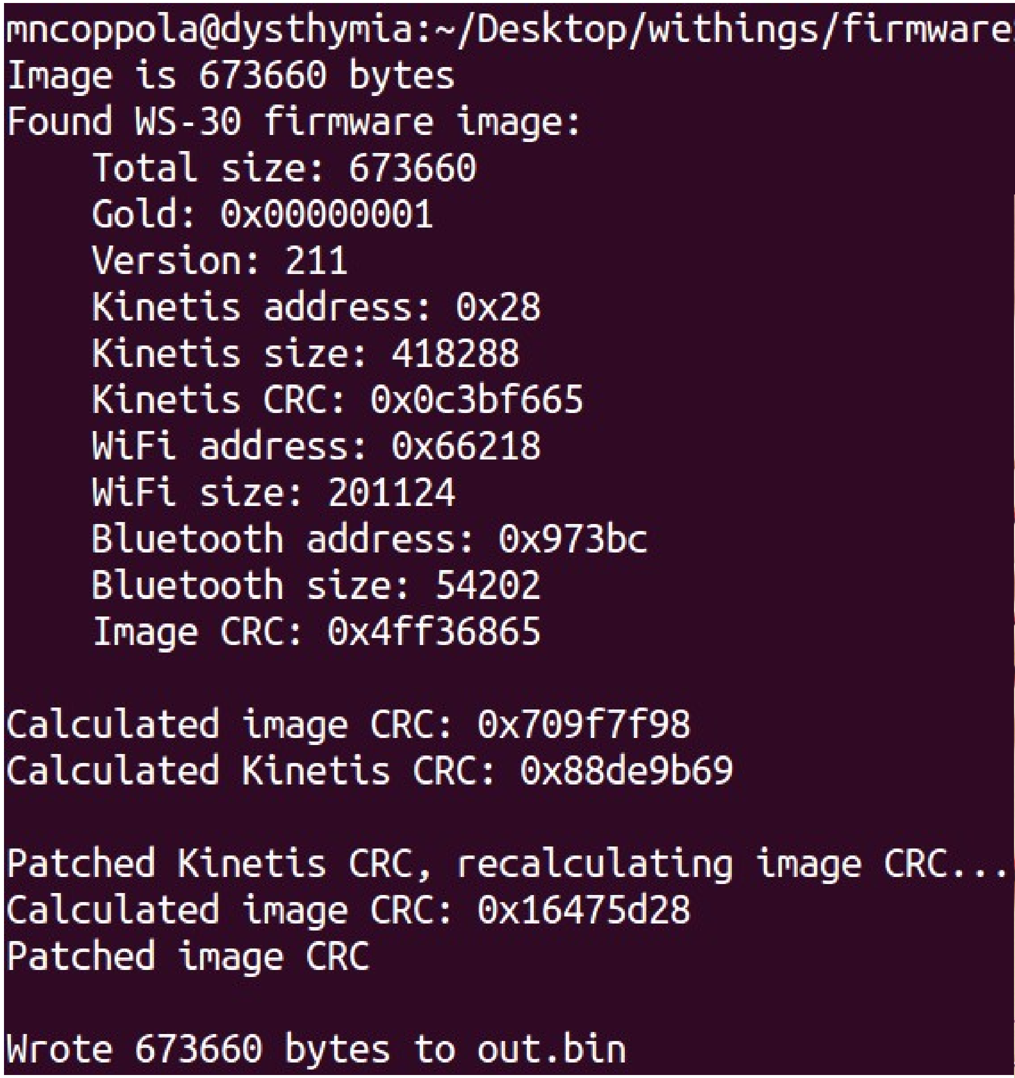


```
unsigned int __fastcall init_CRC32(unsigned int poly, unsigned int seed)
{
  SIM_SCGC6 |= 0x40000u;
  CRC_CTRL |= 0x1000000u;                       // TCRC = 1, 32-bit CRC mode
  CRC_CTRL |= 0x4000000u;                       // FXOR = 1, complement data
  CRC_CTRL |= 0x20000000u;                      // TOTR = 10, bits and bytes are transposed for read
  CRC_CTRL |= 0x80000000;                       // TOT = 10, bits and bytes are transposed for write
  CRC_GPOLY = poly;                             // set CRC polynomial value
  CRC_CTRL |= 0x2000000u;                       // WAS = 1, writes to CRC_CRC (data register) are seed values
  CRC_CRC = seed;                               // set CRC seed value
  CRC_CTRL &= 0xFDFFFFFF;                       // WAS = 0, writes to CRC_CRC (data register) are data values
  return poly;
}
```

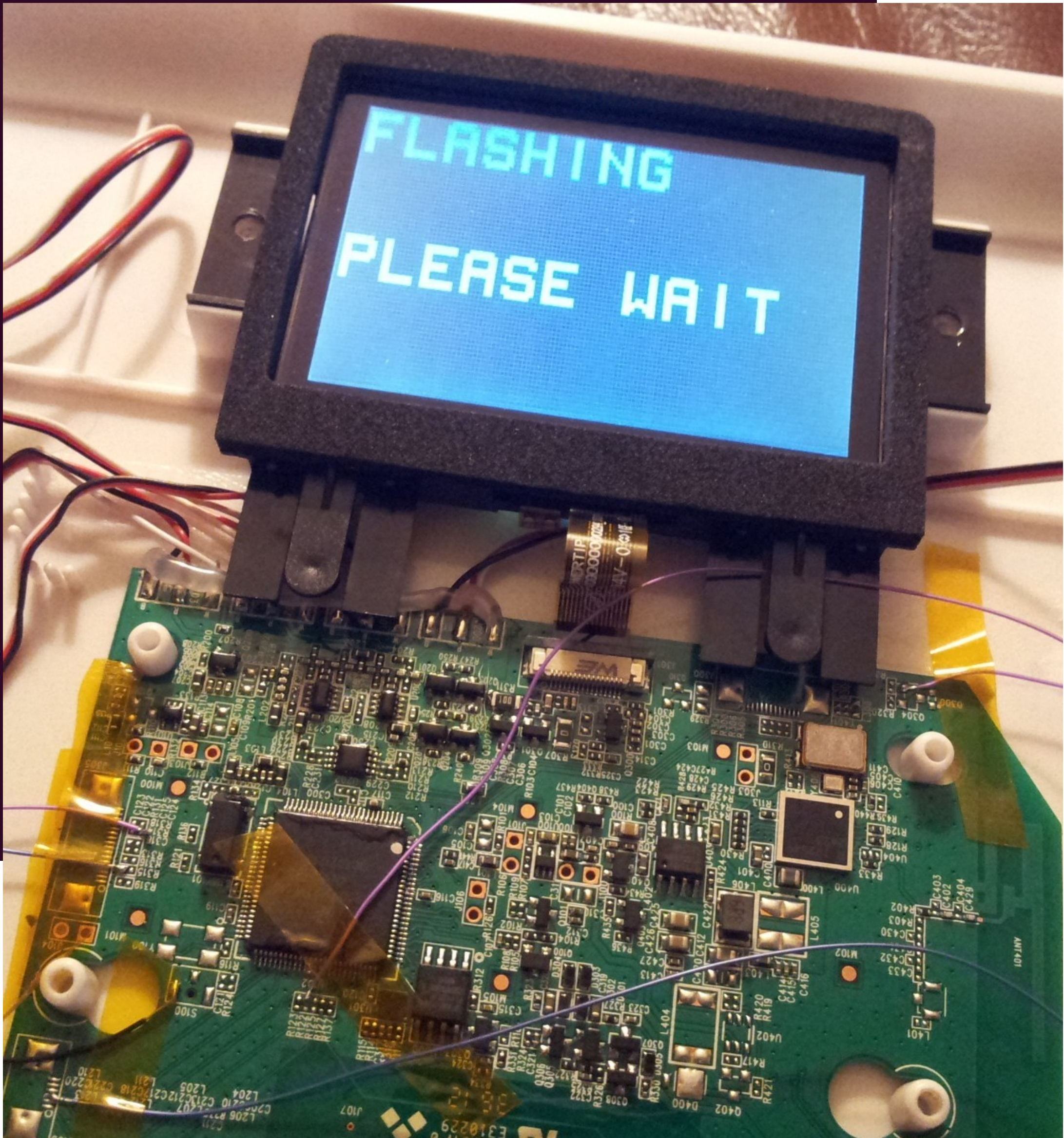


```
    read_from_SPI(&SPI0_regs_base, firmware, bank_addr, 160);
    offset = *(_DWORD *)firmware - 4;           // first dword is firmware size
    if ( (unsigned int)(*(_DWORD *)firmware - 4) <= 0xDFFFC )// max firmware size 917,500 bytes
    {
      read_from_SPI(&SPI0_regs_base, (char *)&firmware_CRC, bank_addr + offset, 4);// get firmware checksum
      init_CRC32(CRC32_POLY, 0xFFFFFFFF);
      for ( i = 0; ; i = next_i )
      {
        next_i = i + 512;
        if ( i + 512 > (unsigned int)offset )
          break;
        read_from_SPI(&SPI0_regs_base, fw_data, bank_addr + i, 512);
        sub_311F0();
        write_to_CRC(fw_data, 128);
      }
      leftover = offset - i;
      if ( leftover > 0 )
      {
        read_from_SPI(&SPI0_regs_base, fw_data, bank_addr + i, (unsigned __int16)leftover);
        write_to_CRC(fw_data, leftover >> 2);
      }
      calculated_CRC = CRC_CRC;
```

Crafting arbitrary images



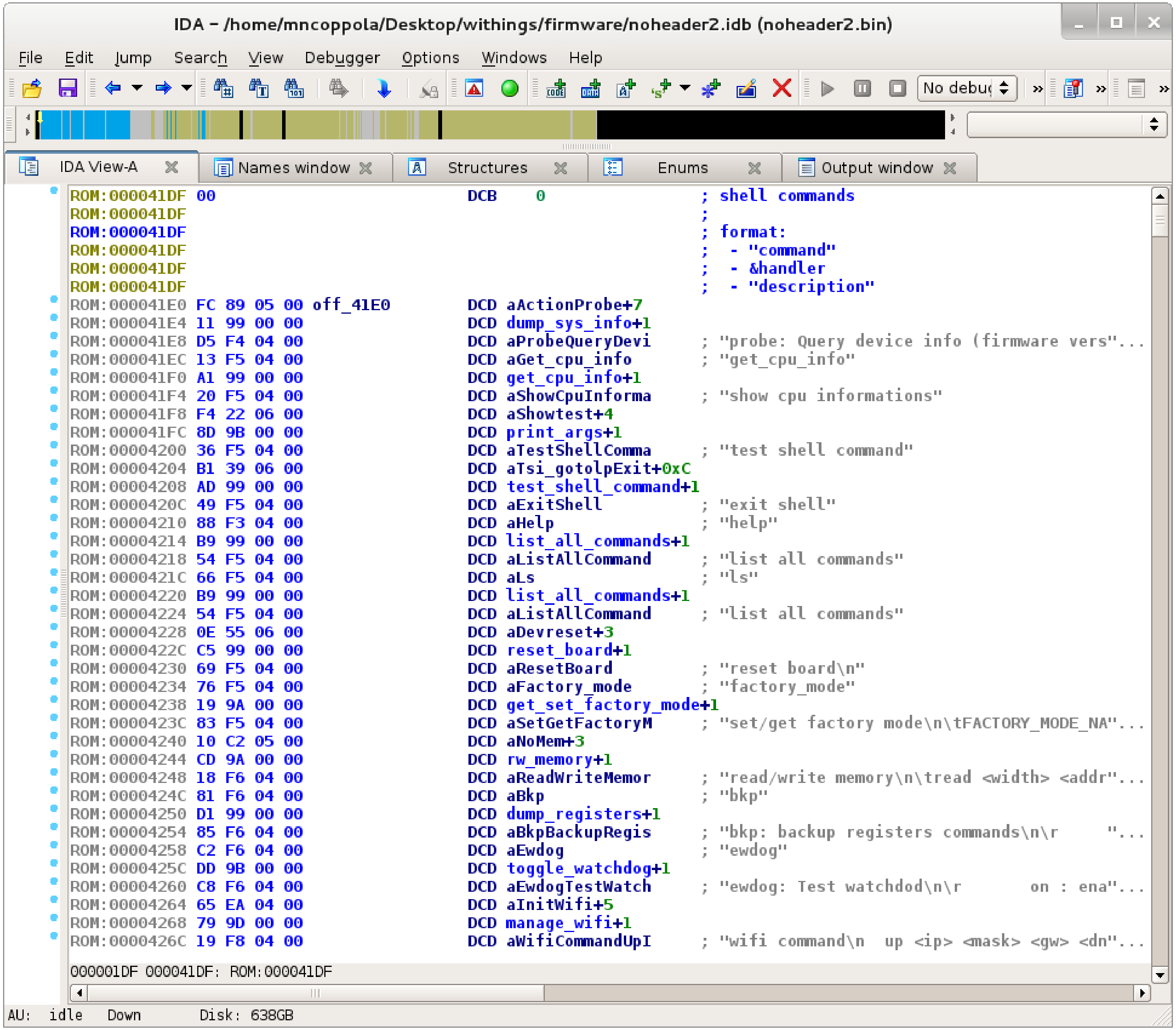
```
mncoppola@dysthymia:~/Desktop/withings/firmware$ ./image_chksum modified_211.bin out.bin
Image is 673660 bytes
Found WS-30 firmware image:
    Total size: 673660
    Gold: 0x00000001
    Version: 211
    Kinetis address: 0x28
    Kinetis size: 418288
    Kinetis CRC: 0x0c3bf665
    WiFi address: 0x66218
    WiFi size: 201124
    Bluetooth address: 0x973bc
    Bluetooth size: 54202
    Image CRC: 0x4ff36865

Calculated image CRC: 0x709f7f98
Calculated Kinetis CRC: 0x88de9b69

Patched Kinetis CRC, recalculating image CRC...
Calculated image CRC: 0x16475d28
Patched image CRC

Wrote 673660 bytes to out.bin
```



Let's fuzz this thing

- We want some delicious 0dayz
- Send it invalid JSON / garbage values
- We need introspection
- Debug console?

Debug console!

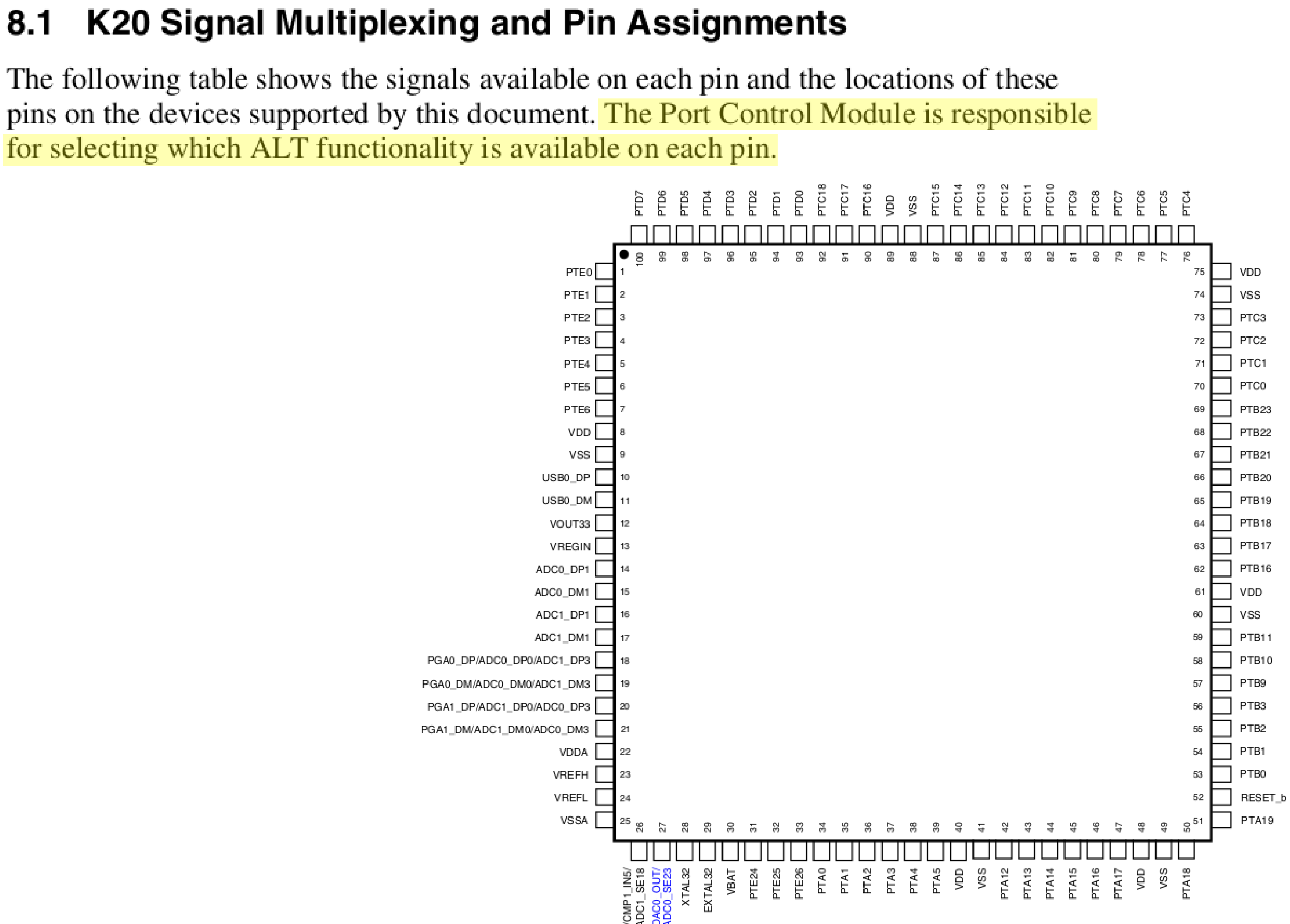

IDA - /home/mncoppola/Desktop/withings/firmware/noheader2.idb (noheader2.bin)
File Edit Jump Search View Debugger Options Windows Help
No debug
IDA View-A
Names window
Structures
Enums
Output window

```
ROM:000041DF 00                            DCB     0                ; shell commands
ROM:000041DF                                                        ;
ROM:000041DF                                                        ; format:
ROM:000041DF                                                        ;  - "command"
ROM:000041DF                                                        ;  - &handler
ROM:000041DF                                                        ;  - "description"
ROM:000041E0 FC 89 05 00 off_41E0          DCD aActionProbe+7
ROM:000041E4 11 99 00 00                   DCD dump_sys_info+1
ROM:000041E8 D5 F4 04 00                   DCD aProbeQueryDevi      ; "probe: Query device info (firmware vers"...
ROM:000041EC 13 F5 04 00                   DCD aGet_cpu_info        ; "get_cpu_info"
ROM:000041F0 A1 99 00 00                   DCD get_cpu_info+1
ROM:000041F4 20 F5 04 00                   DCD aShowCpuInforma      ; "show cpu informations"
ROM:000041F8 F4 22 06 00                   DCD aShowtest+4
ROM:000041FC 8D 9B 00 00                   DCD print_args+1
ROM:00004200 36 F5 04 00                   DCD aTestShellComma      ; "test shell command"
ROM:00004204 B1 39 06 00                   DCD aTsi_gotolpExit+0xC
ROM:00004208 AD 99 00 00                   DCD test_shell_command+1
ROM:0000420C 49 F5 04 00                   DCD aExitShell           ; "exit shell"
ROM:00004210 88 F3 04 00                   DCD aHelp                ; "help"
ROM:00004214 B9 99 00 00                   DCD list_all_commands+1
ROM:00004218 54 F5 04 00                   DCD aListAllCommand      ; "list all commands"
ROM:0000421C 66 F5 04 00                   DCD aLs                  ; "ls"
ROM:00004220 B9 99 00 00                   DCD list_all_commands+1
ROM:00004224 54 F5 04 00                   DCD aListAllCommand      ; "list all commands"
ROM:00004228 0E 55 06 00                   DCD aDevreset+3
ROM:0000422C C5 99 00 00                   DCD reset_board+1
ROM:00004230 69 F5 04 00                   DCD aResetBoard          ; "reset board\n"
ROM:00004234 76 F5 04 00                   DCD aFactory_mode        ; "factory_mode"
ROM:00004238 19 9A 00 00                   DCD get_set_factory_mode+1
ROM:0000423C 83 F5 04 00                   DCD aSetGetFactoryM      ; "set/get factory mode\n\tFACTORY_MODE_NA"...
ROM:00004240 10 C2 05 00                   DCD aNoMem+3
ROM:00004244 CD 9A 00 00                   DCD rw_memory+1
ROM:00004248 18 F6 04 00                   DCD aReadWriteMemor      ; "read/write memory\n\tread <width> <addr"...
ROM:0000424C 81 F6 04 00                   DCD aBkp                 ; "bkp"
ROM:00004250 D1 99 00 00                   DCD dump_registers+1
ROM:00004254 85 F6 04 00                   DCD aBkpBackupRegis      ; "bkp: backup registers commands\n\r     "...
ROM:00004258 C2 F6 04 00                   DCD aEwdog               ; "ewdog"
ROM:0000425C DD 9B 00 00                   DCD toggle_watchdog+1
ROM:00004260 C8 F6 04 00                   DCD aEwdogTestWatch      ; "ewdog: Test watchdod\n\r       on : ena"...
ROM:00004264 65 EA 04 00                   DCD aInitWifi+5
ROM:00004268 79 9D 00 00                   DCD manage_wifi+1
ROM:0000426C 19 F8 04 00                   DCD aWifiCommandUpI      ; "wifi command\n  up <ip> <mask> <gw> <dn"...
```

```
000001DF 000041DF: ROM:000041DF
```

```
AU:  idle   Down       Disk: 638GB
```

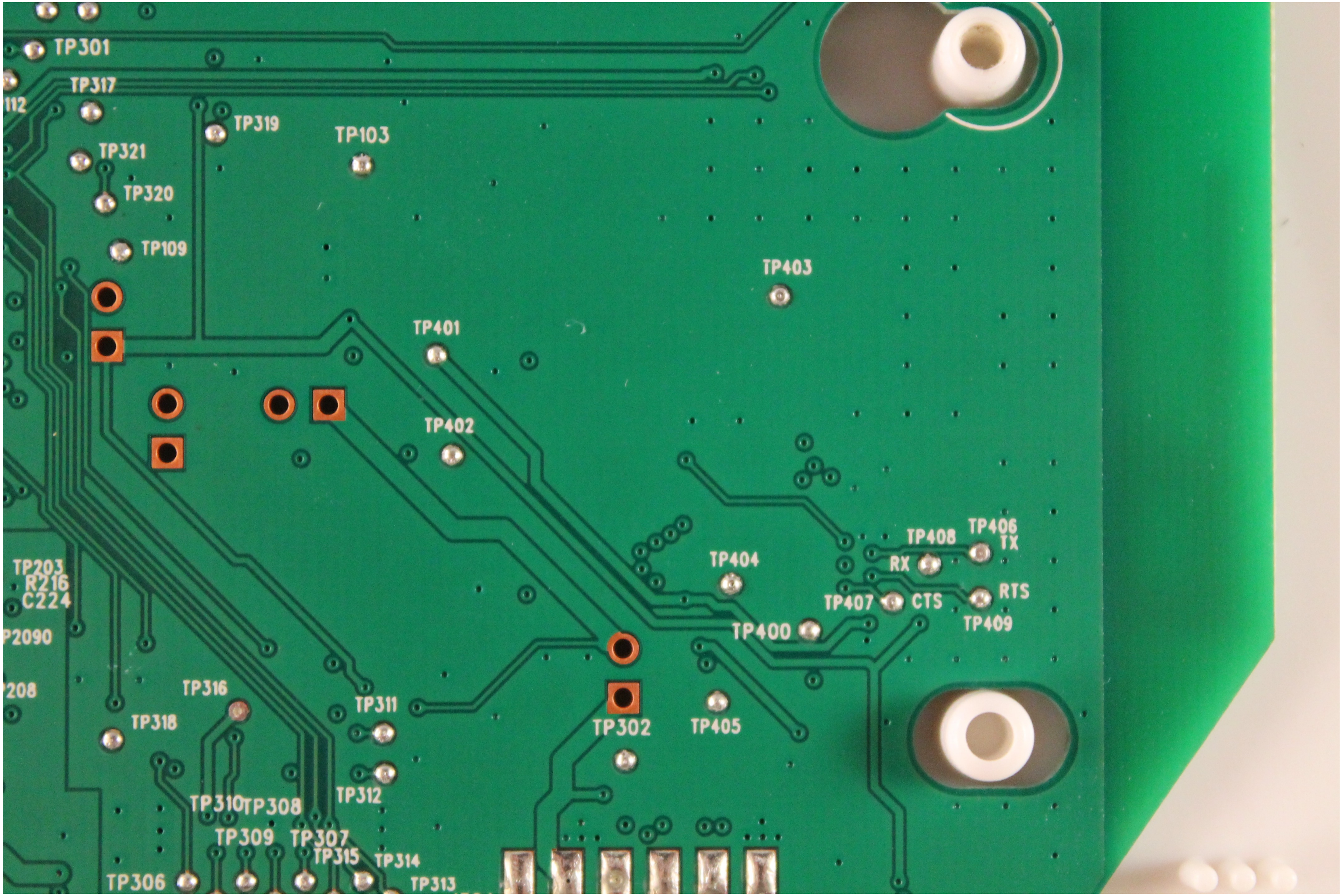
UART in the data sheet



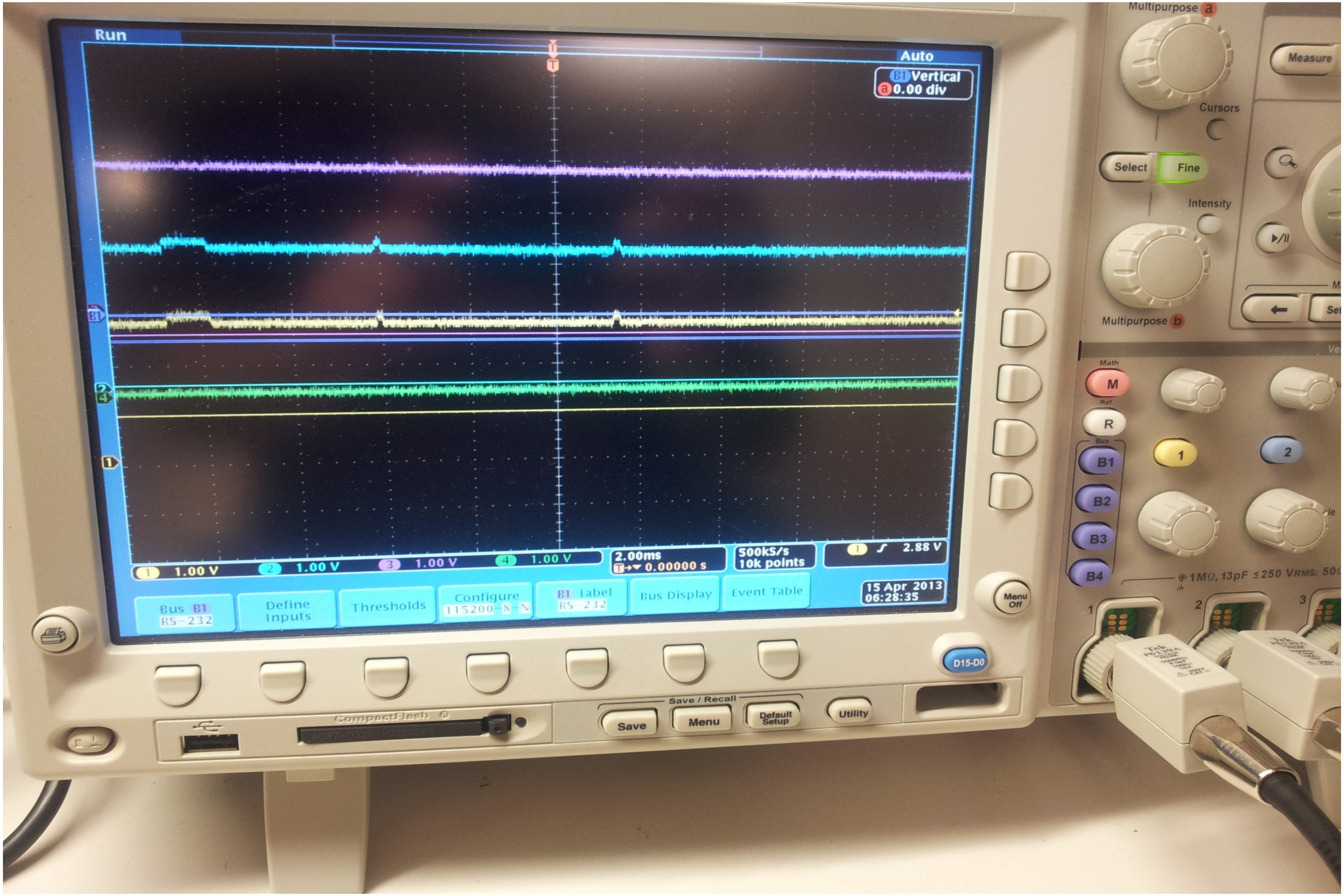
8.1 K20 Signal Multiplexing and Pin Assignments

The following table shows the signals available on each pin and the locations of these pins on the devices supported by this document. The Port Control Module is responsible for selecting which ALT functionality is available on each pin.

Spot the UART!



Not a great signal...



Introspection-less

- Code execution on the device would solve all the problems

- Hook the ARM hard fault handler to send crash dumps over the wire

- Re-purpose WsLib to easily send HTTP requests with register context info

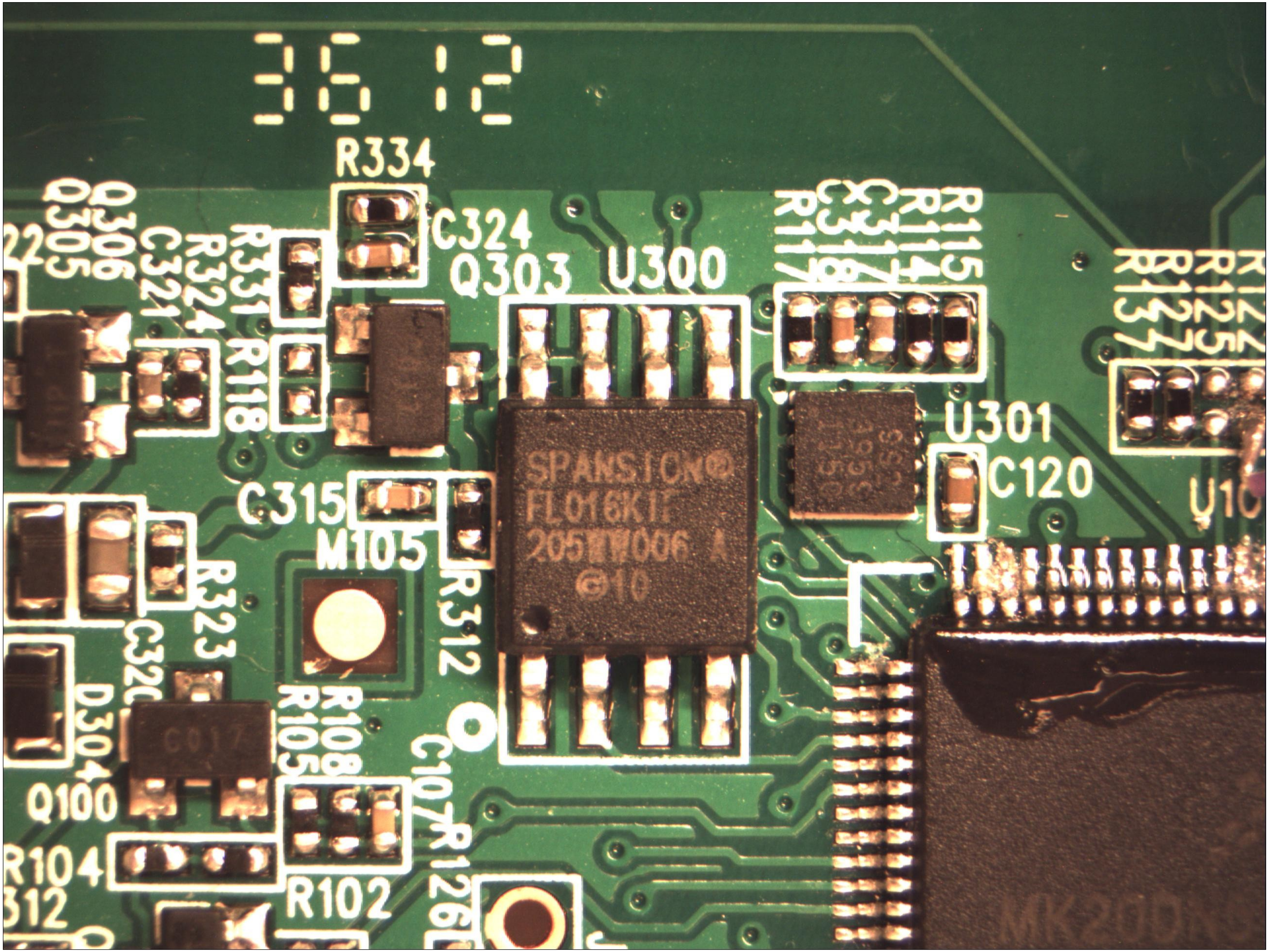
Hard Fault Handler

```
struct processor_status {
    unsigned int r0;
    unsigned int r1;
    unsigned int r2;
    unsigned int r3;
    unsigned int r12;
    unsigned int lr;
    unsigned int pc;
    unsigned int psr;
};

#define sprintf ((int (*)(char *, char *, ...))0x47915)
#define do_HTTP_POST_request ((int (*)(char *, char *, char *, char *, int))0x16425)

void hard_fault_handler ( void )
{
    /* All function pointers +1 for thumb */
    char buf[100];
    struct processor_status *regs;

    sprintf(buf, 0x5c22c, regs->r0, regs->r1, regs->r2, regs->r3, regs->r12, regs->lr, regs->pc, regs->psr);

    do_HTTP_POST_request("hacker", "fault", buf, 0, 0);
}
```

Pushing updates

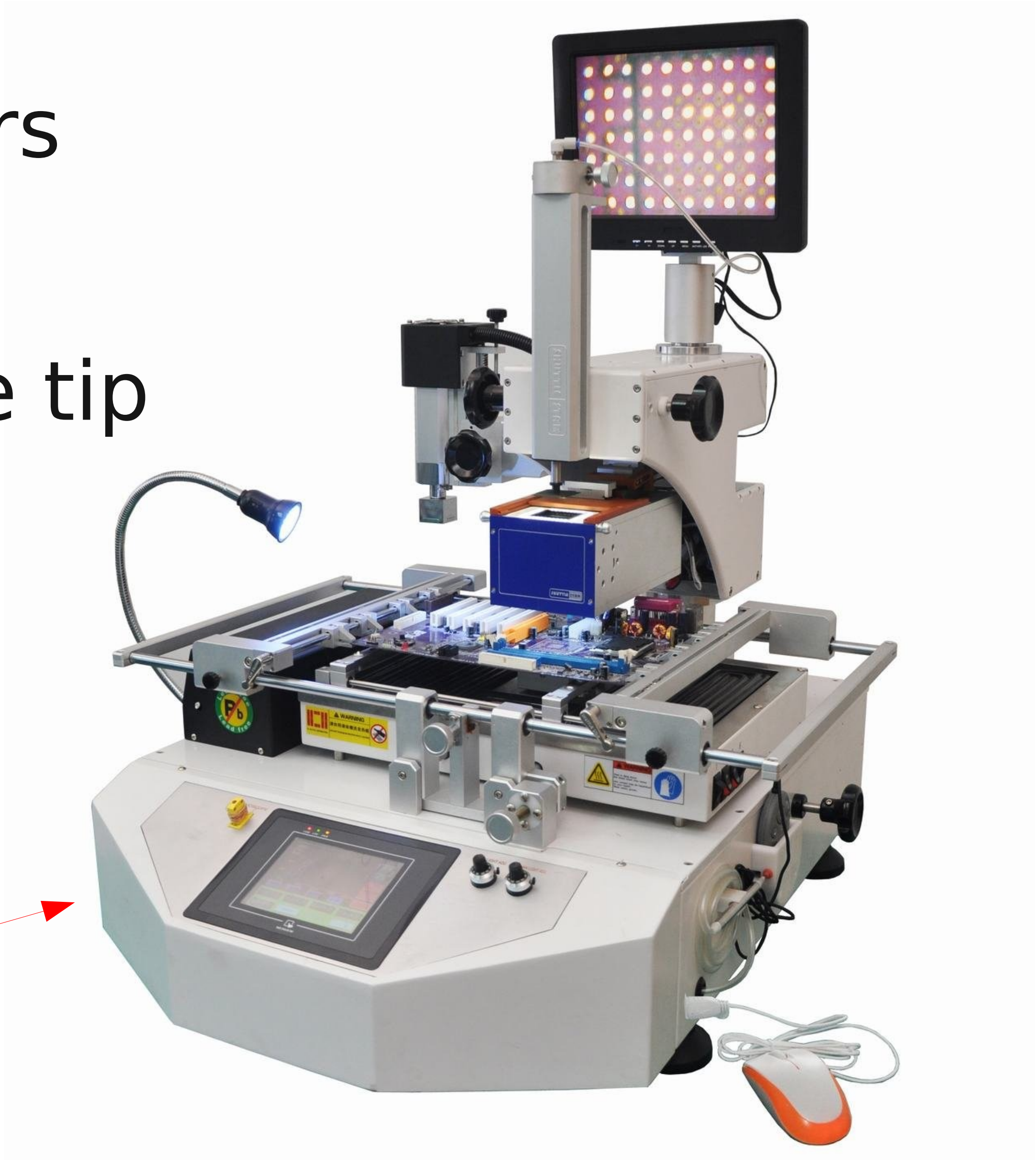
- We'll probably brick it at some point
- Need a recovery plan
- No bootloader we can break into
- Dump (and reprogram) the flash memory!

Identify the flash chip

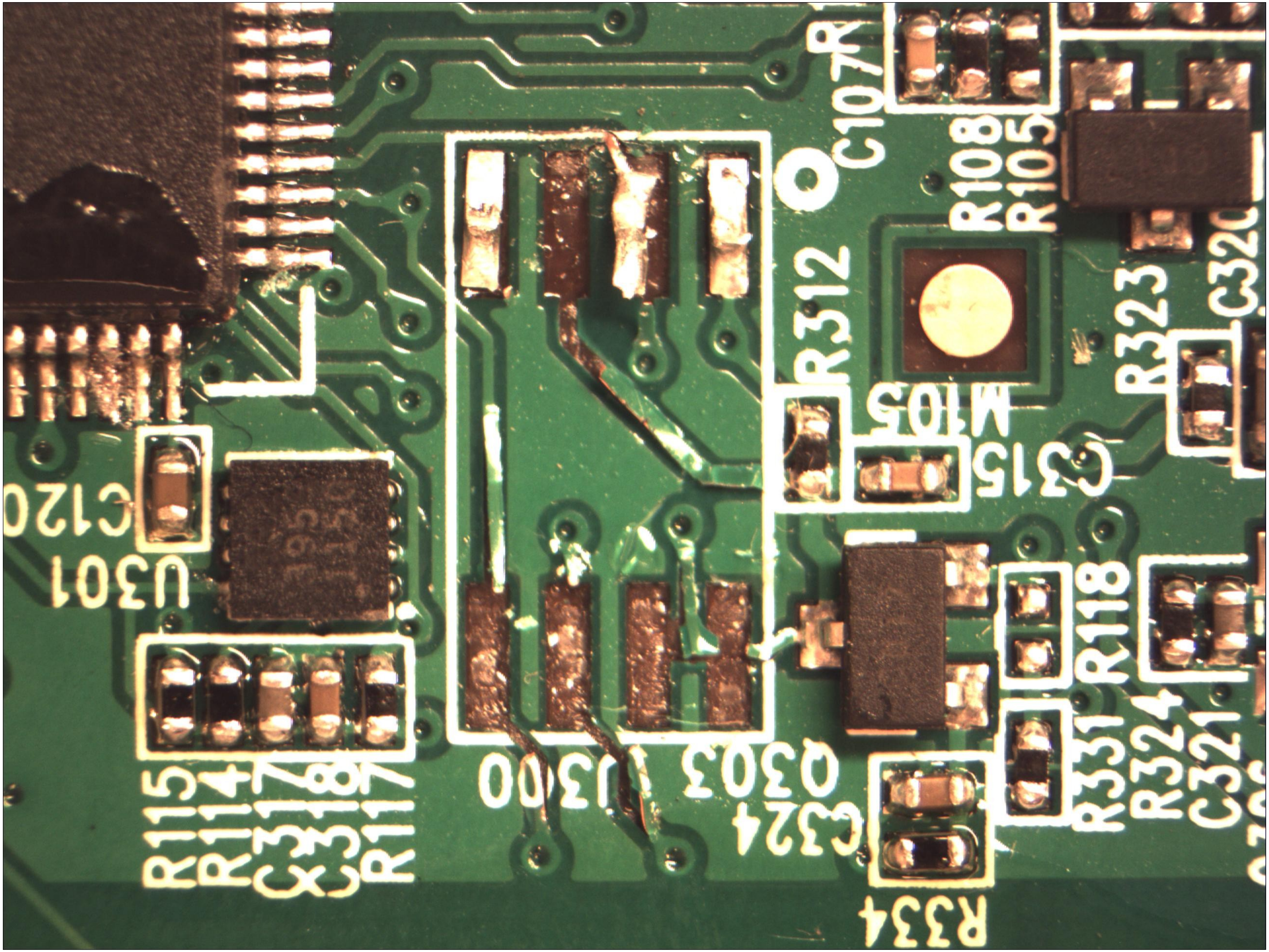


Desolder the flash chip

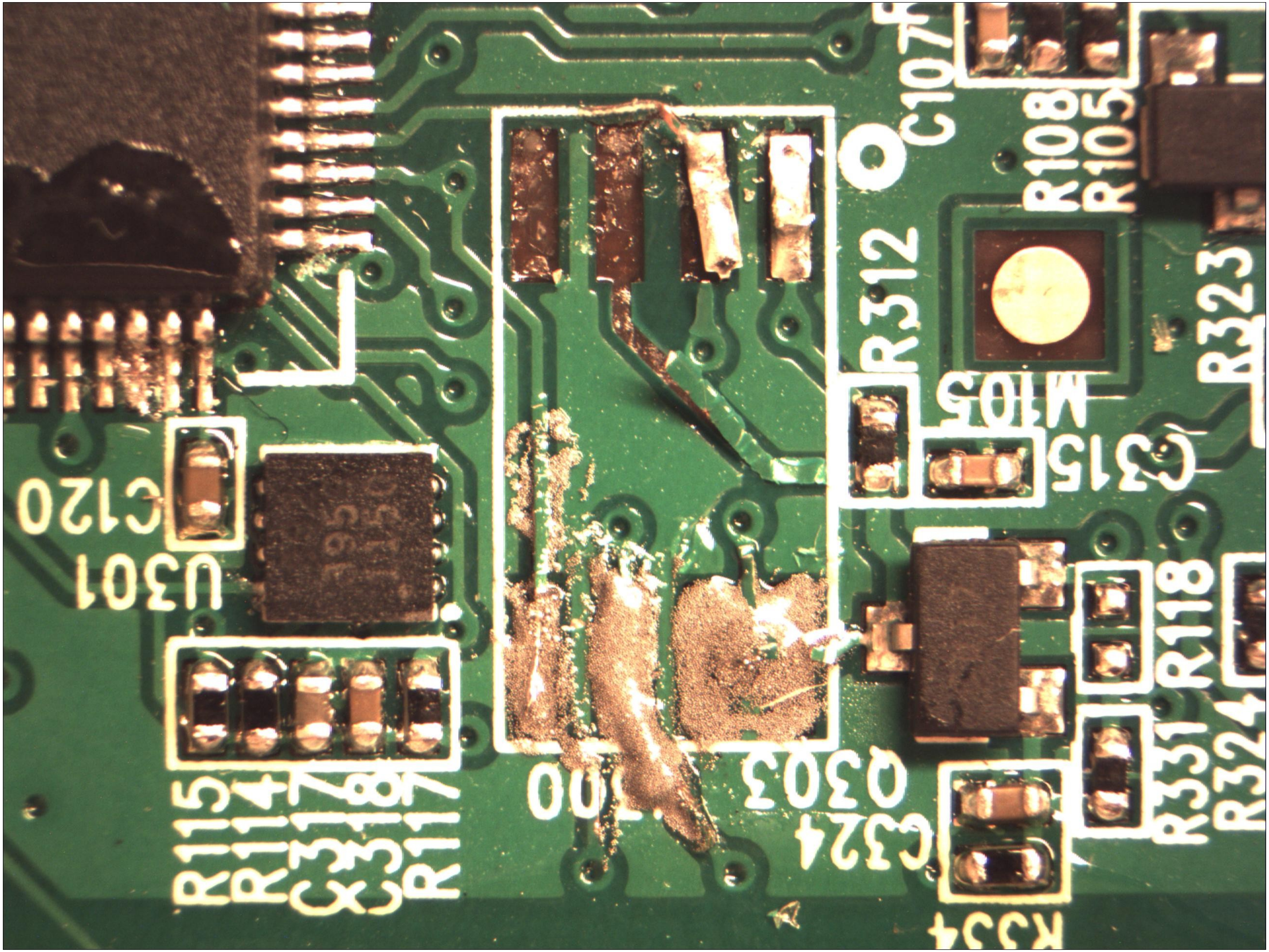
- Heat gun + tweezers
- Soldering iron blade tip
- Solder wick
- Rework station



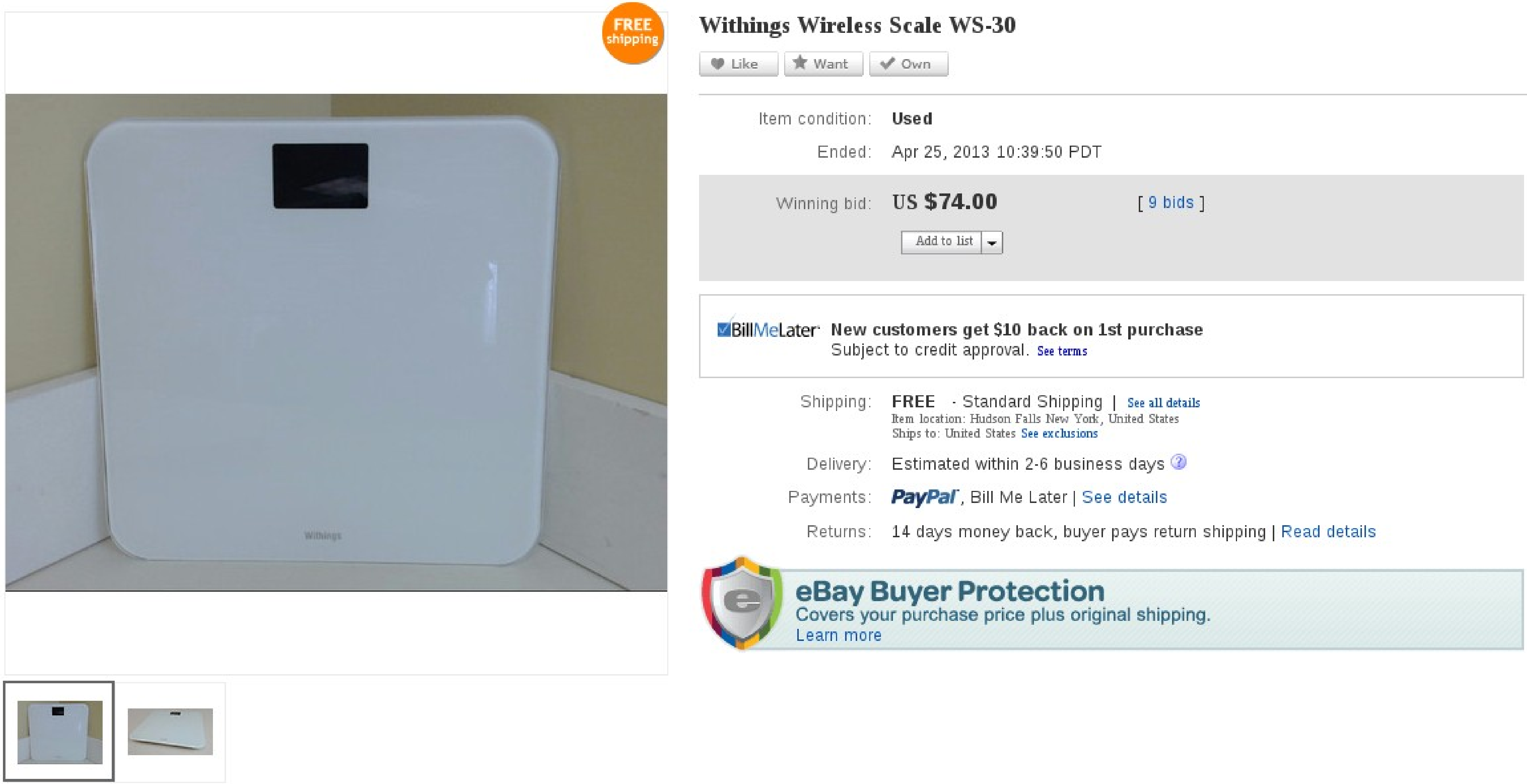
This means you did it wrong



This means you fixed it wrong



Fuck it, new scale



Withings Wireless Scale WS-30

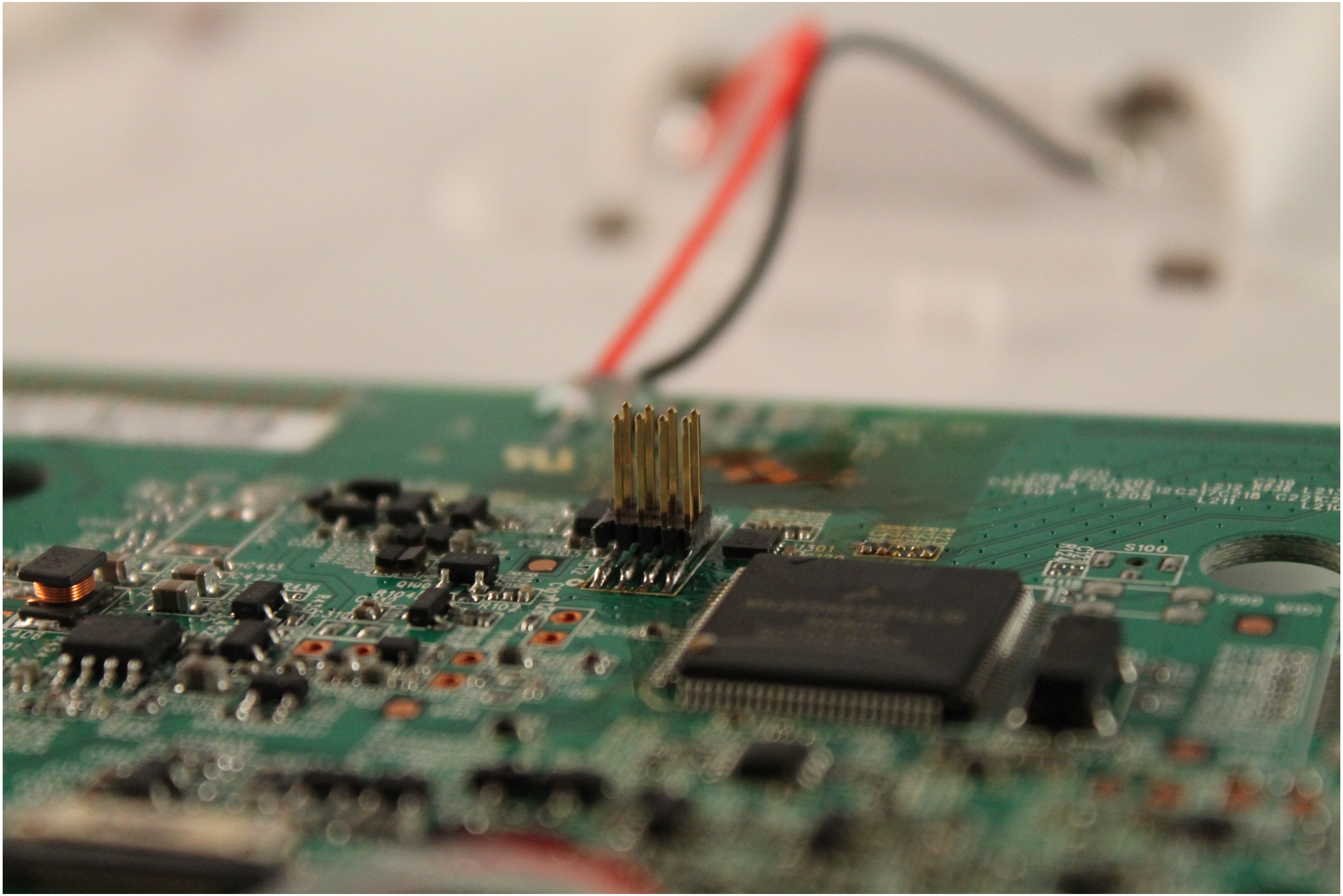
Item condition:	**Used**
Ended:	Apr 25, 2013 10:39:50 PDT
Winning bid:	**US $74.00** [9 bids]

BillMeLater **New customers get $10 back on 1st purchase**
Subject to credit approval. See terms

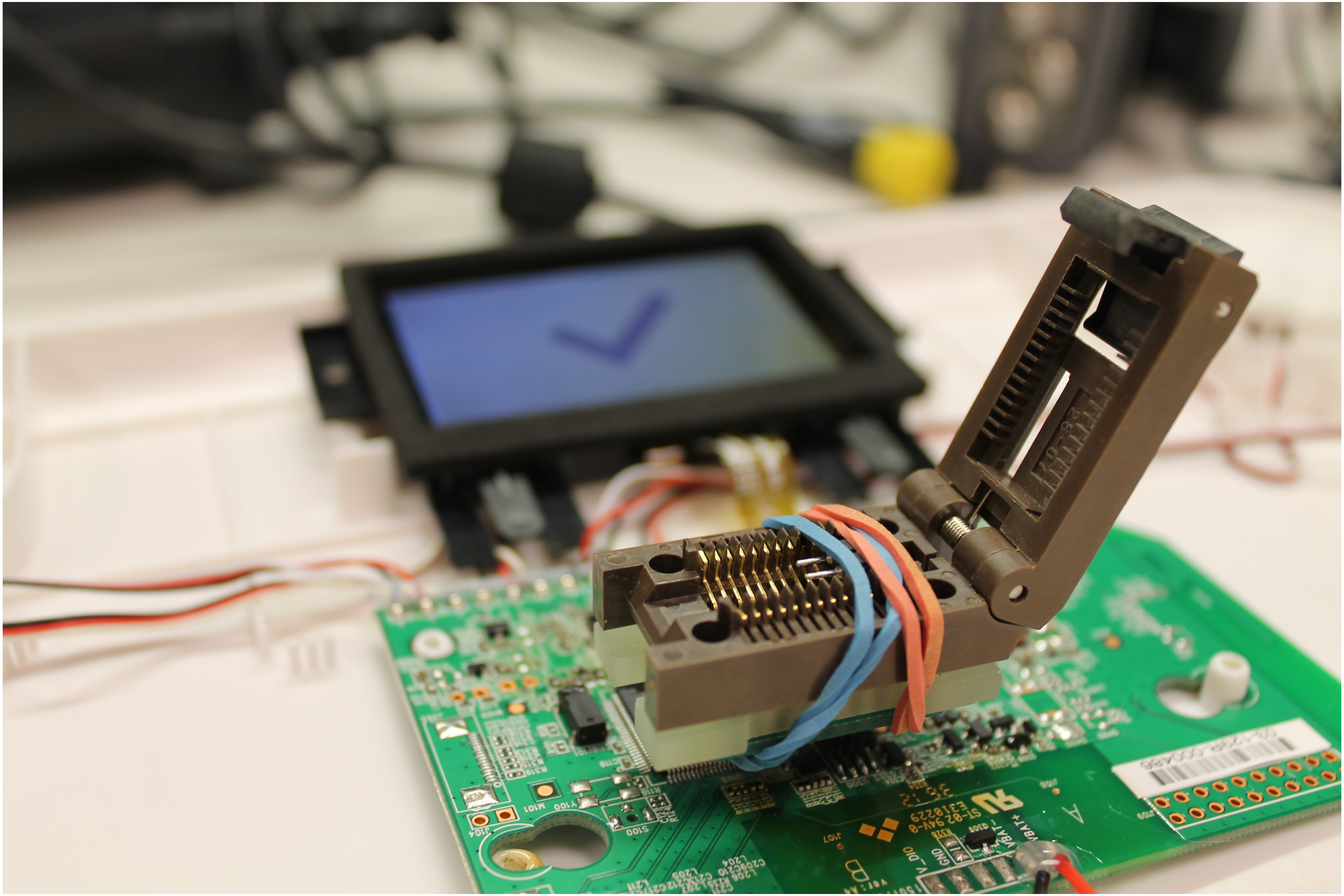
Shipping:	FREE - Standard Shipping | See all details Item location: Hudson Falls New York, United States Ships to: United States See exclusions
Delivery:	Estimated within 2-6 business days
Payments:	PayPal, Bill Me Later | See details
Returns:	14 days money back, buyer pays return shipping | Read details

eBay Buyer Protection
Covers your purchase price plus original shipping.
Learn more

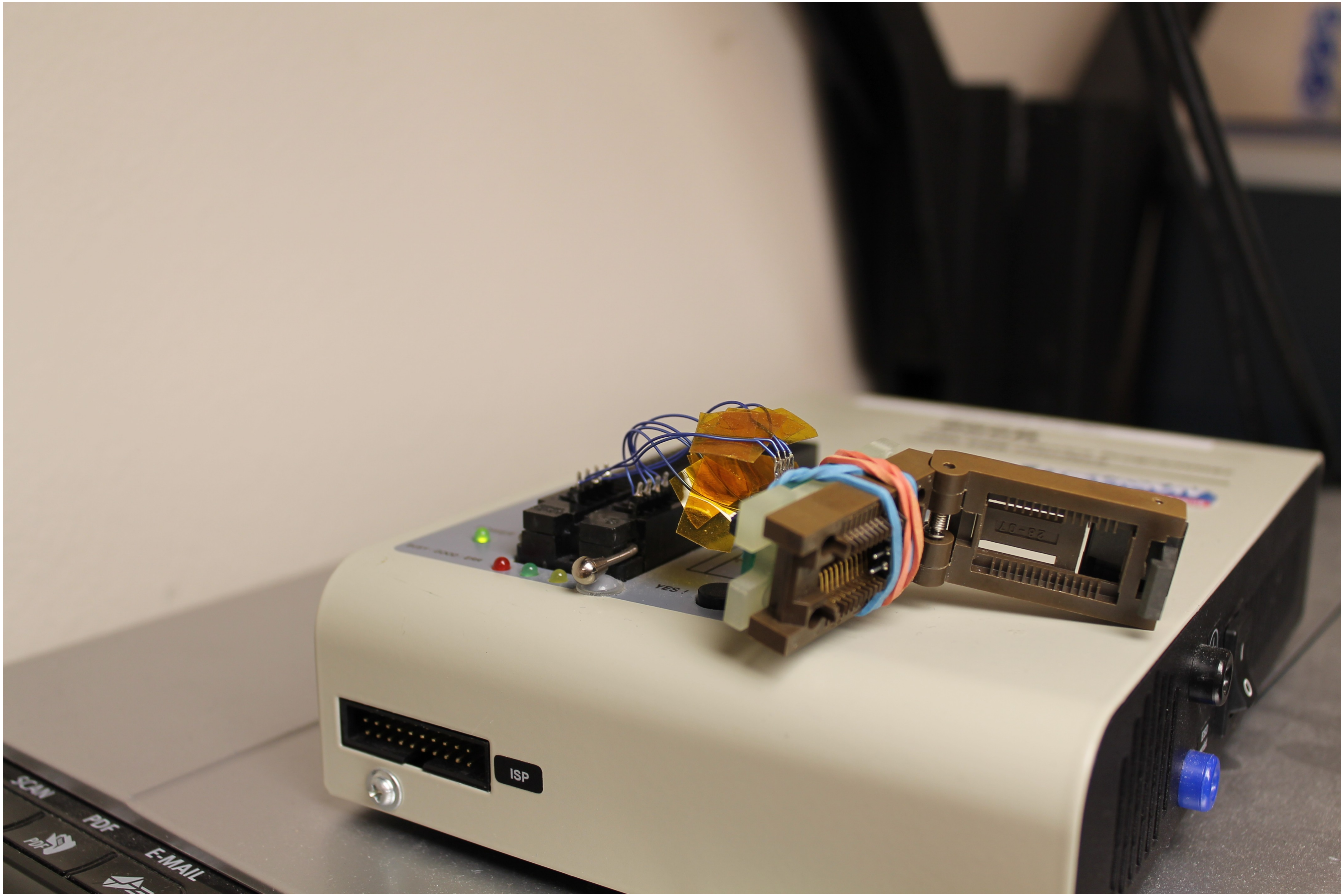
Replace chip with pins



Yup, it still works



Connect to USB programmer



Score!

```
[mncoppola@dysthymia flash]$ hexdump -C scale.bin | head
00000000  0b 00 04 00 07 00 04 00  01 00 12 00 30 30 3a 32  |............00:2|
00000010  34 3a 65 34 3a 30 36 3a  35 39 3a 64 63 00 03 00  |4:e4:06:59:dc...|
00000020  11 00 36 34 39 61 31 36  62 66 39 37 37 64 33 62  |..649a16bf977d3b|
00000030  33 65 00 02 00 20 00 73  63 61 6c 65 77 73 2e 77  |3e... .scalews.w|
00000040  69 74 68 69 6e 67 73 2e  6e 65 74 3a 38 30 2f 63  |ithings.net:80/c|
00000050  67 69 2d 62 69 6e 00 21  00 4e 00 57 7e 01 00 00  |gi-bin.!.N.W~...|
00000060  00 48 42 00 00 c8 42 00  00 16 43 00 00 00 00 00  |.HB...B...C.....|
00000070  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
00000090  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 03  |................|
```

secret

Challenge format:
<mac_addr>:<secret>:<nonce>

MAC Address: 00:24:e4:06:59:dc
Secret: 649a16bf977d3b3e
Nonce: 00d016bf-242e0bb1

HASH: 25fd29132cf66a5cdf1a7efdc673be26

DEMO TIME

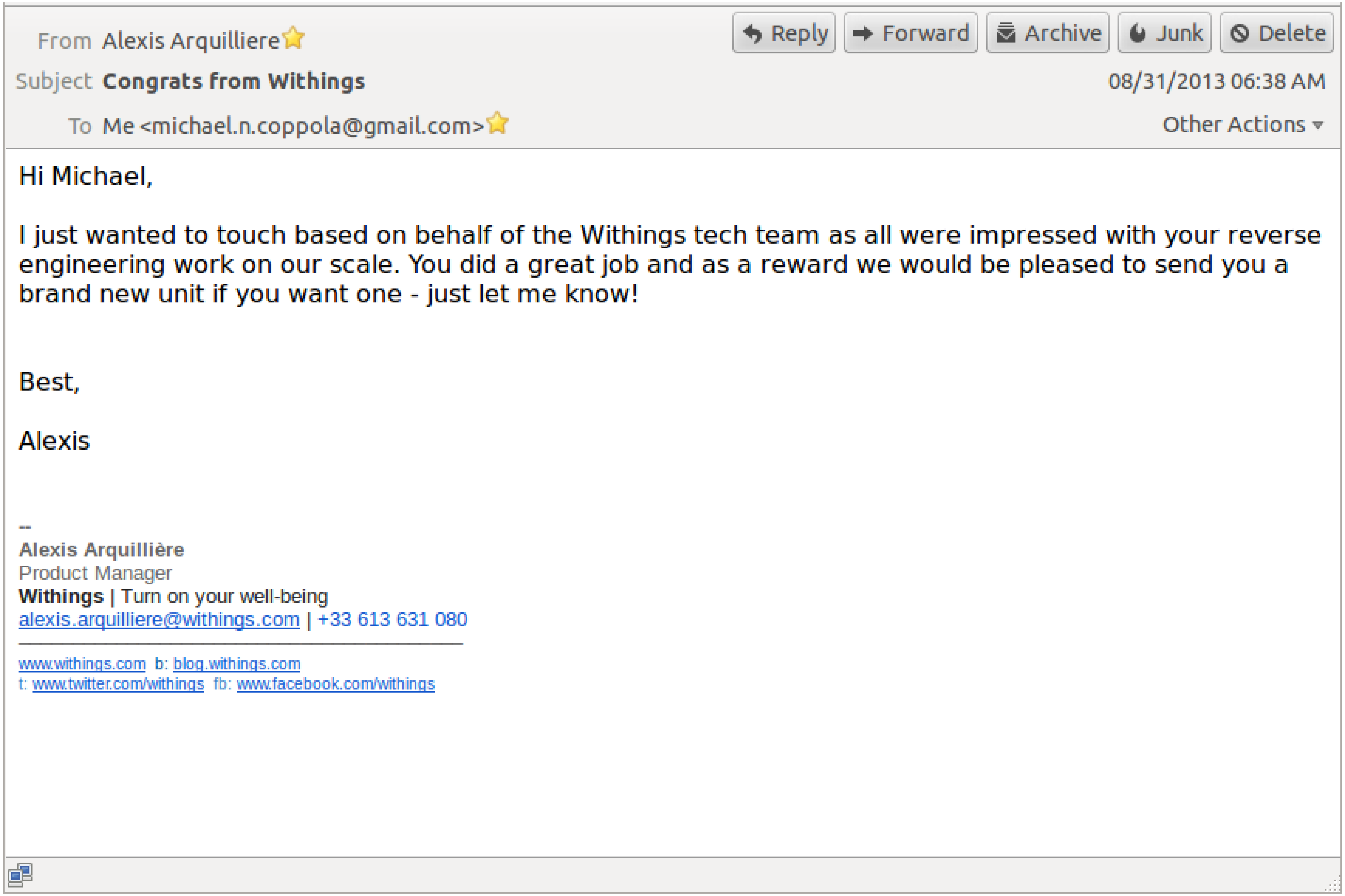
Lessons learned

- ARM compilers are aggressive
  - Reference middle of strings
  - Inline everything possible, even data
- Strings are your friends
  - Find base address
  - Find code blocks
  - Determine symbol names, branch purposes, debugging info
- Lots of help from hardware data sheets and reference manuals
- Embedded system security sucks

Thanks

- Albert Cahalan
- Rob Jerrell
- Jordan Wiens
- Andrew Watts
- Paul Furtado

Thanks to Withings



From Alexis Arquilliere

Subject **Congrats from Withings**

08/31/2013 06:38 AM

To Me <michael.n.coppola@gmail.com>

Hi Michael,

I just wanted to touch based on behalf of the Withings tech team as all were impressed with your reverse engineering work on our scale. You did a great job and as a reward we would be pleased to send you a brand new unit if you want one - just let me know!

Best,

Alexis

--
Alexis Arquillière
Product Manager
Withings | Turn on your well-being
alexis.arquilliere@withings.com | +33 613 631 080

www.withings.com b: blog.withings.com
t: www.twitter.com/withings fb: www.facebook.com/withings

Greetz

- #busticati
- Marauders
- bliss, thing2

Questions?

@mncoppola
poppopret.org