SIMTrace v2

MITM for your phone

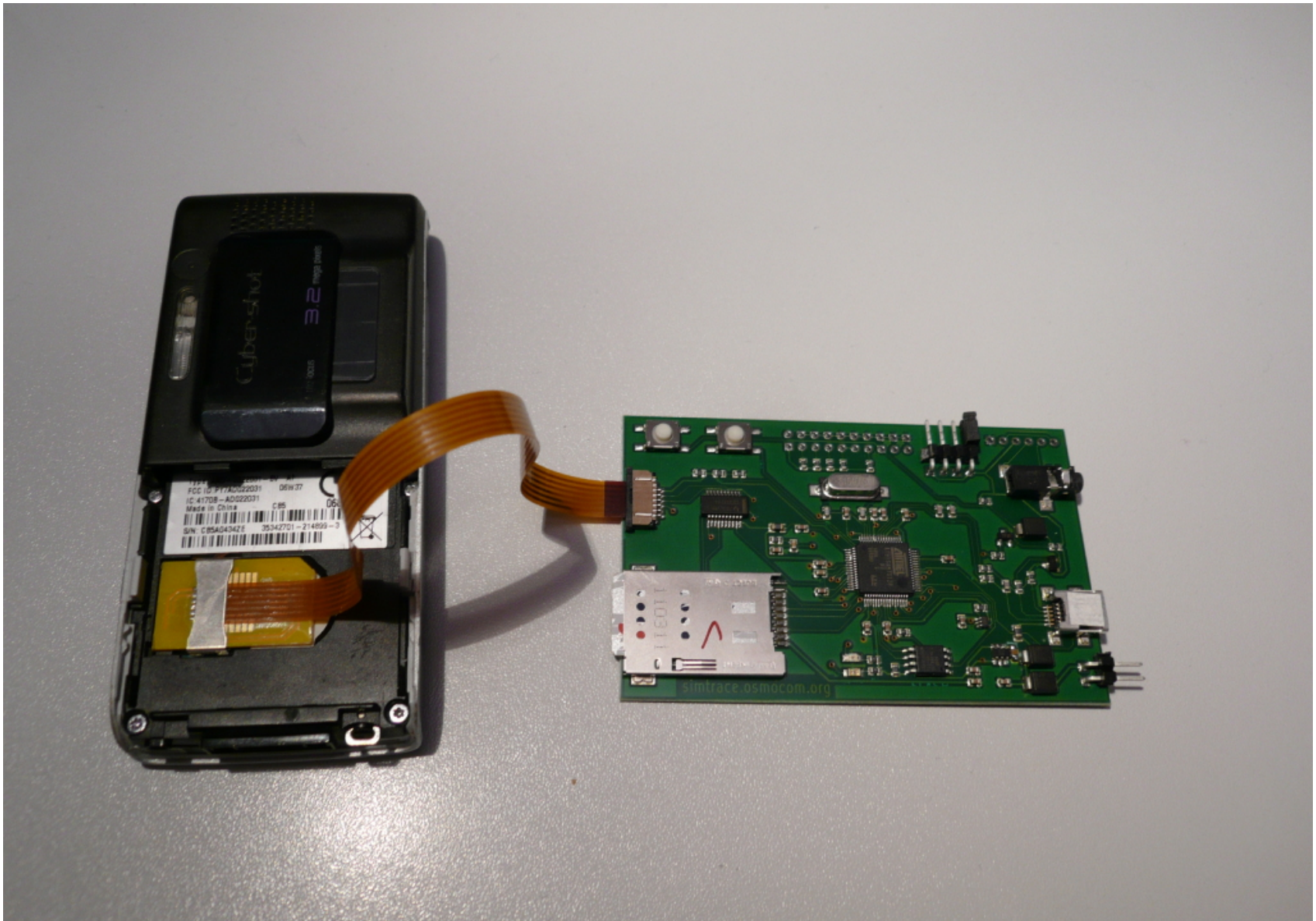
Christina Quast

sysmocom &
Department of Security in Telecommunications
(Institute of Technology Berlin)

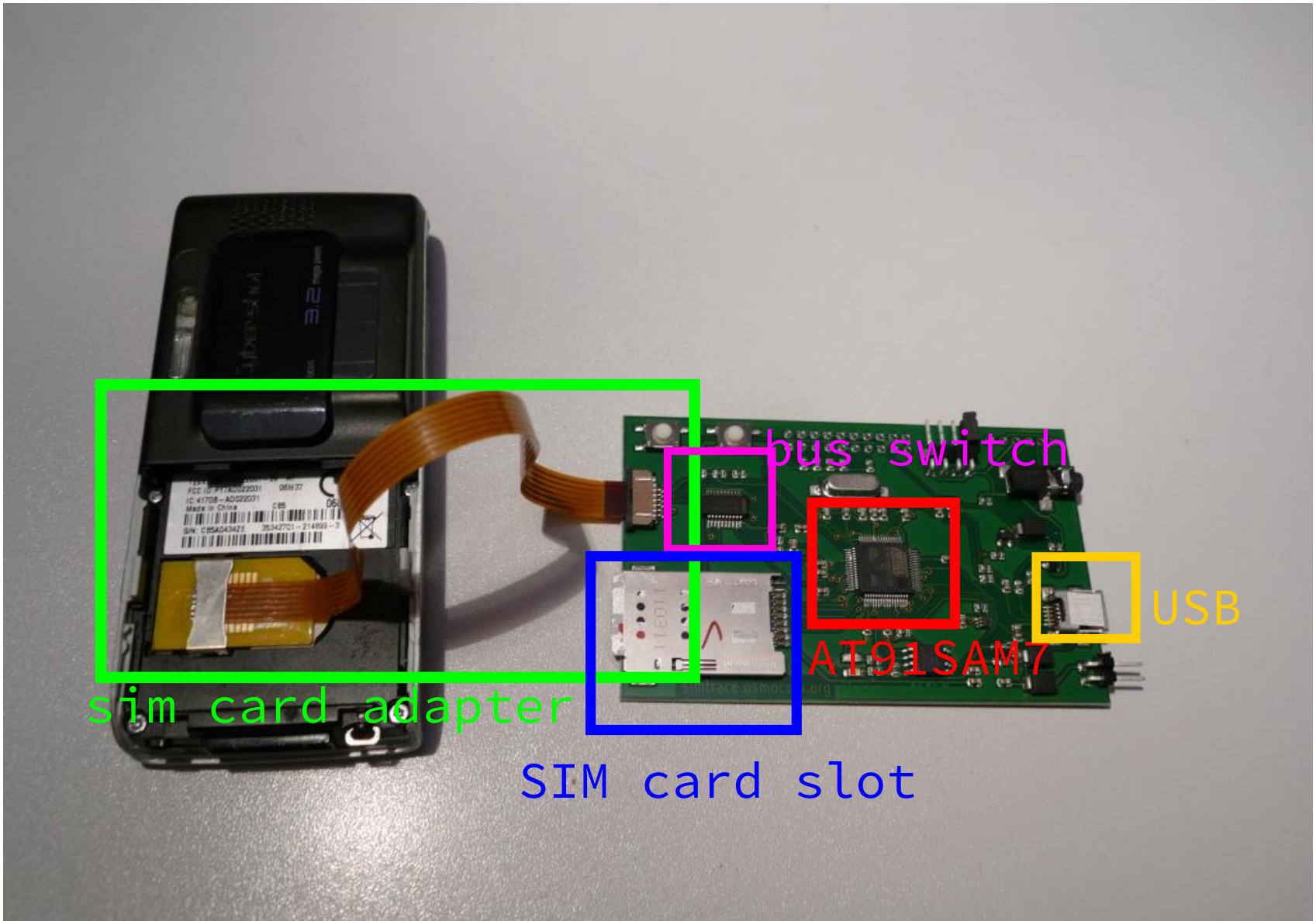
Blackhoodie 0x7e1

January 12, 2018

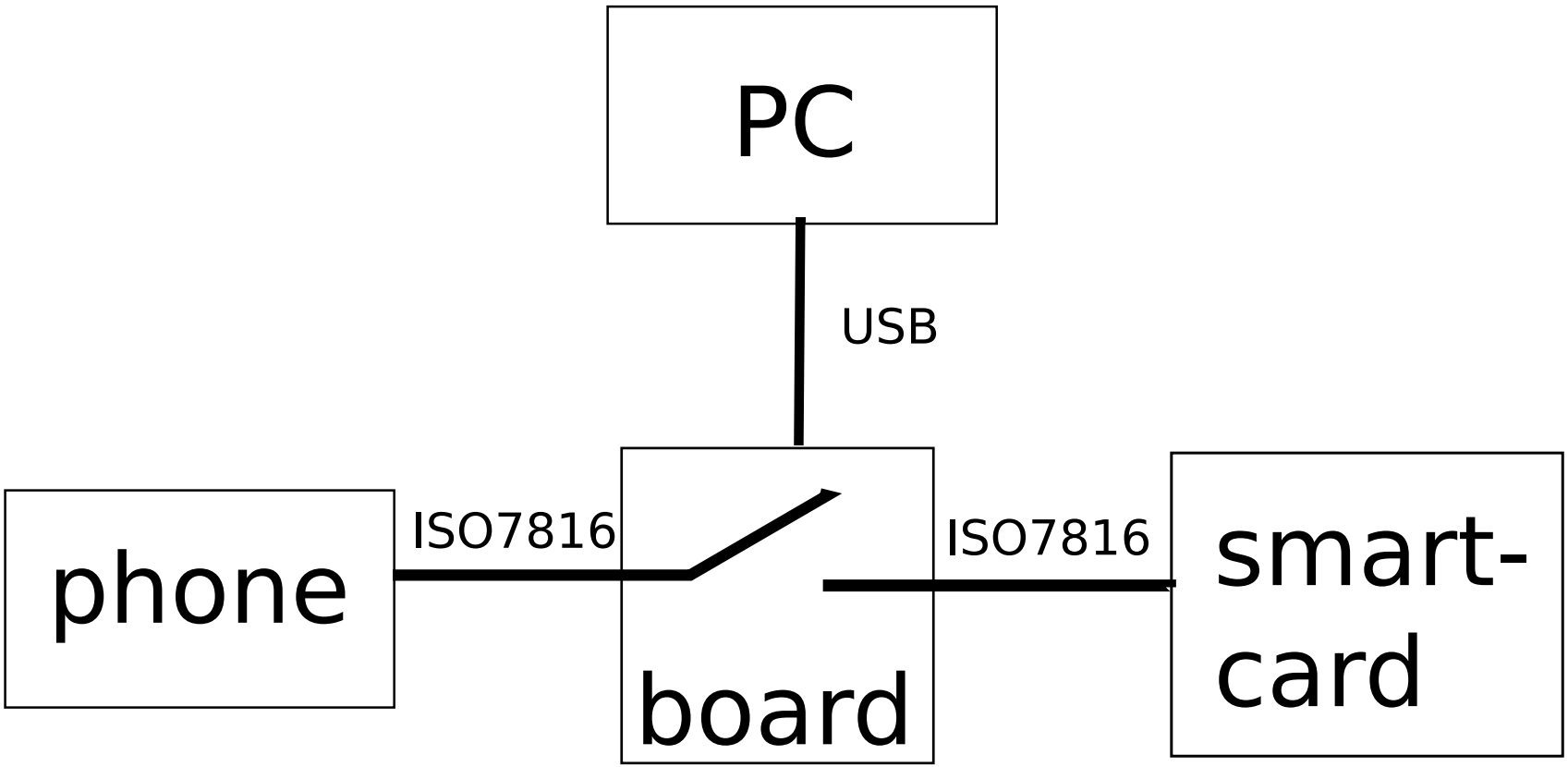
What is SIMTrace?



What is SIMTrace?



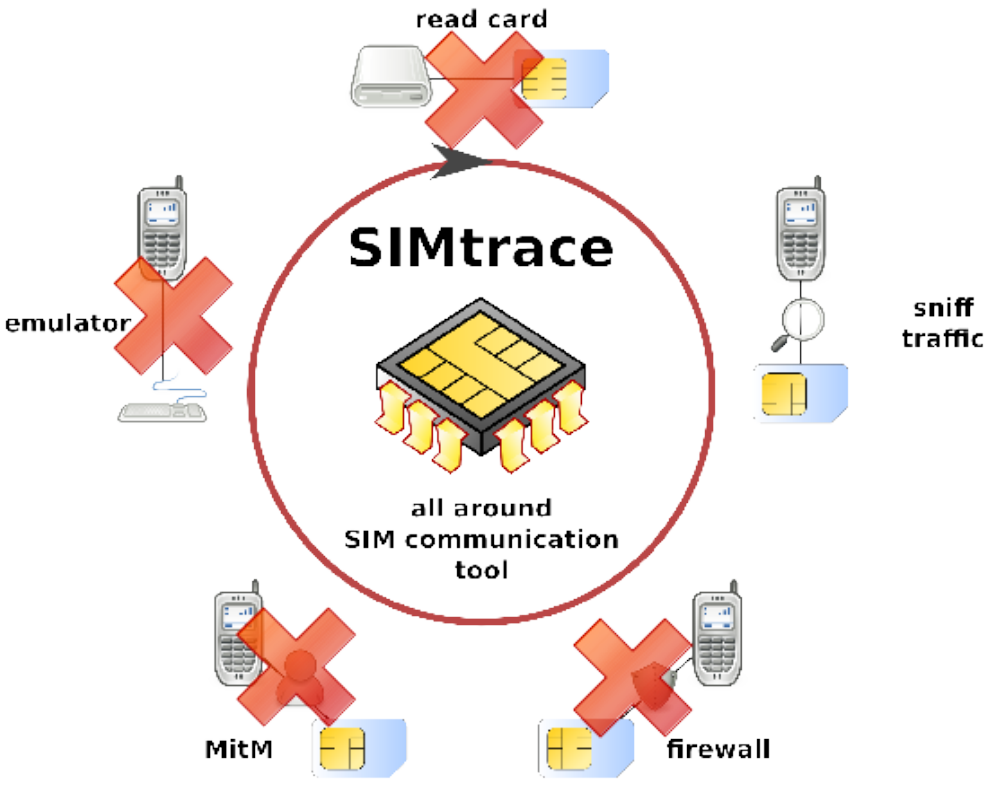
What is SIMTrace?



Basic functions

osmocomSIMTRACE

http://simtrace.osmocom.org

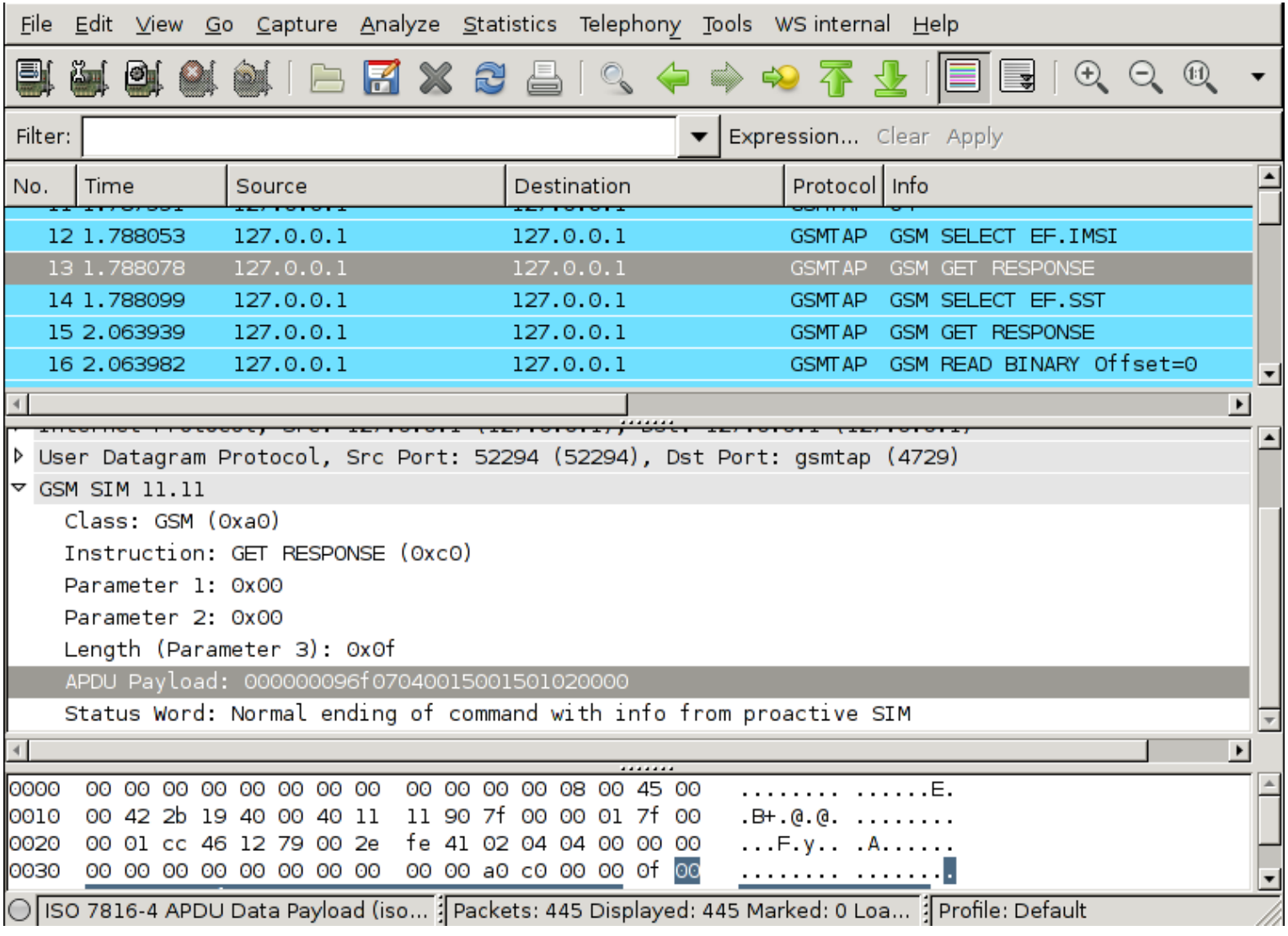


monitor, analyze and use the power of SIM

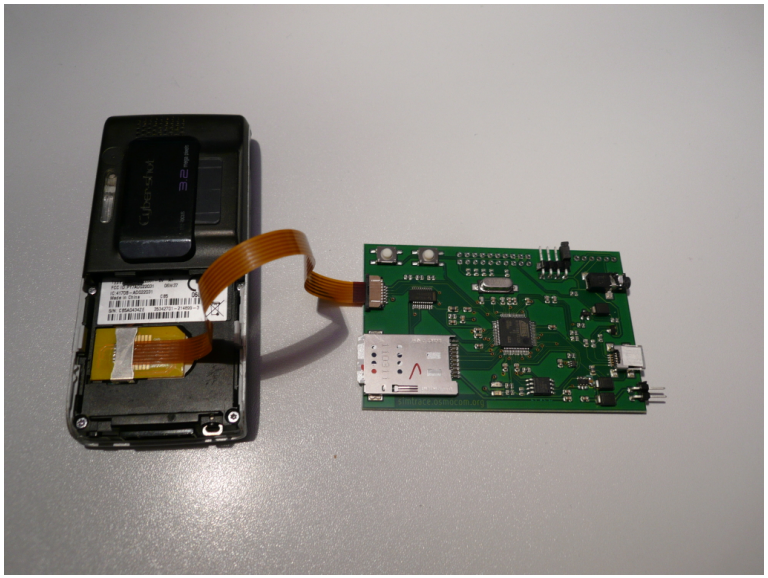
Back in the old days

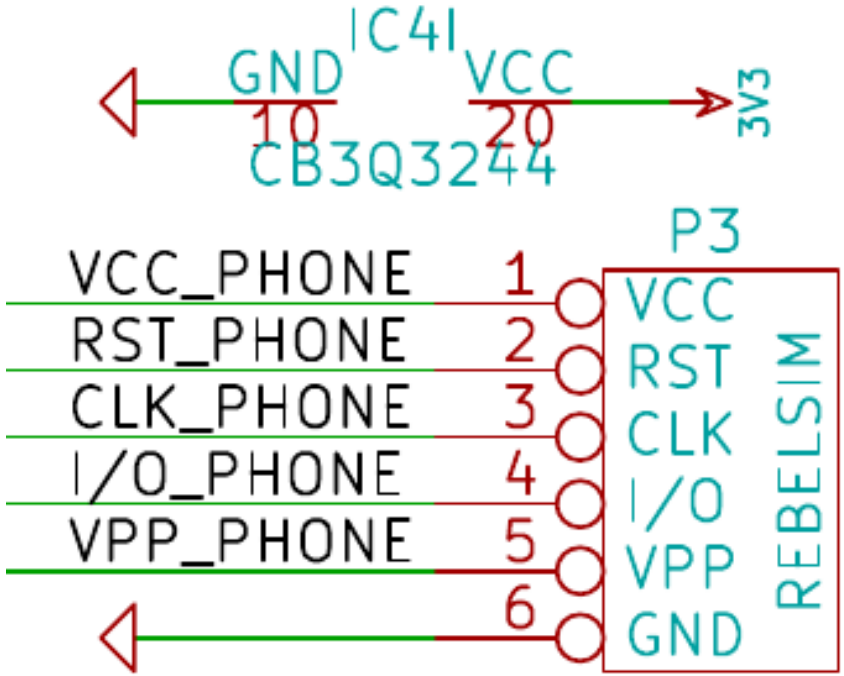
```
# sudo ./simtrace
APDU: (9):  a0 a4 00 00 02 6f 07 9f 0f
APDU: (22):  a0 c0 00 00 0f 00 00 00 09 6f 07 04 00 15 00 15 01 02 00 00 91 78
APDU: (9):  a0 a4 00 00 02 6f 38 9f 0f
APDU: (22):  a0 c0 00 00 0f 00 00 00 09 6f 38 04 00 15 00 55 01 02 00 00 91 78
APDU: (16):  a0 b0 00 00 09 ff 3f ff ff 00 00 3f 03 00 91 78
APDU: (9):  a0 a4 00 00 02 6f ad 9f 0f
APDU: (8):  a0 b0 00 00 01 00 91 78
APDU: (9):  a0 a4 00 00 02 6f 07 9f 0f
APDU: (16):  a0 b0 00 00 09 08 49 06 20 11 49 00 11 06 91 78
APDU: (9):  a0 a4 00 00 02 6f 7e 9f 0f
APDU: (18):  a0 b0 00 00 0b ff ff ff ff 64 f0 00 ff fe 00 03 91 78
APDU: (9):  a0 a4 00 00 02 6f 78 9f 0f
APDU: (9):  a0 b0 00 00 02 00 01 91 78
APDU: (9):  a0 a4 00 00 02 6f 74 9f 0f
APDU: (23):  a0 b0 00 00 10 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 91 78
APDU: (9):  a0 a4 00 00 02 6f 20 9f 0f
APDU: (16):  a0 b0 00 00 09 ff ff ff ff ff ff ff ff 07 91 78
APDU: (9):  a0 a4 00 00 02 6f 30 9f 0f
APDU: (22):  a0 c0 00 00 0f 00 00 00 f0 6f 30 04 00 11 00 55 01 02 00 00 91 78
```

Wireshark integration

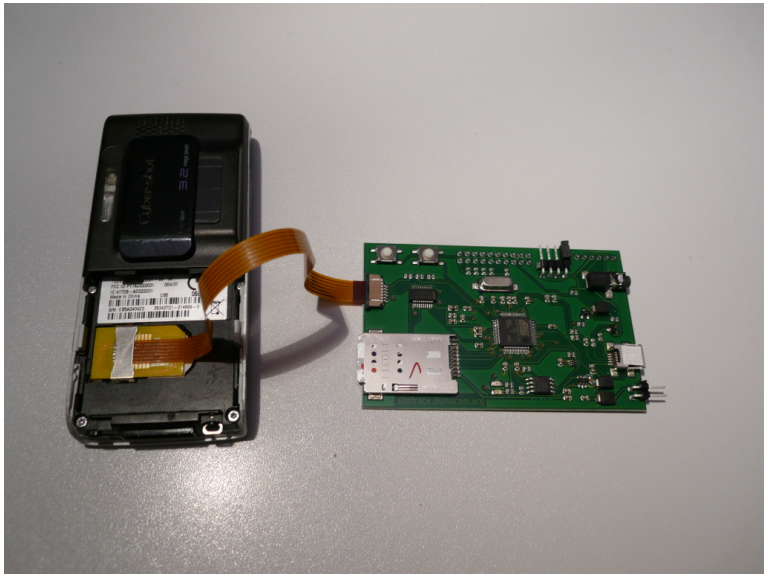


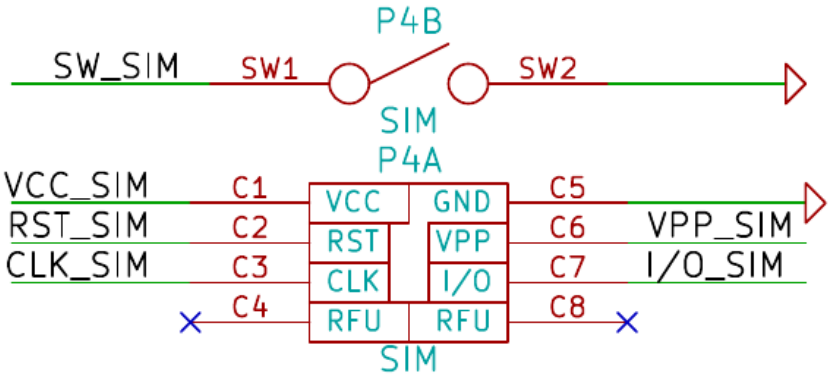
Phone connector



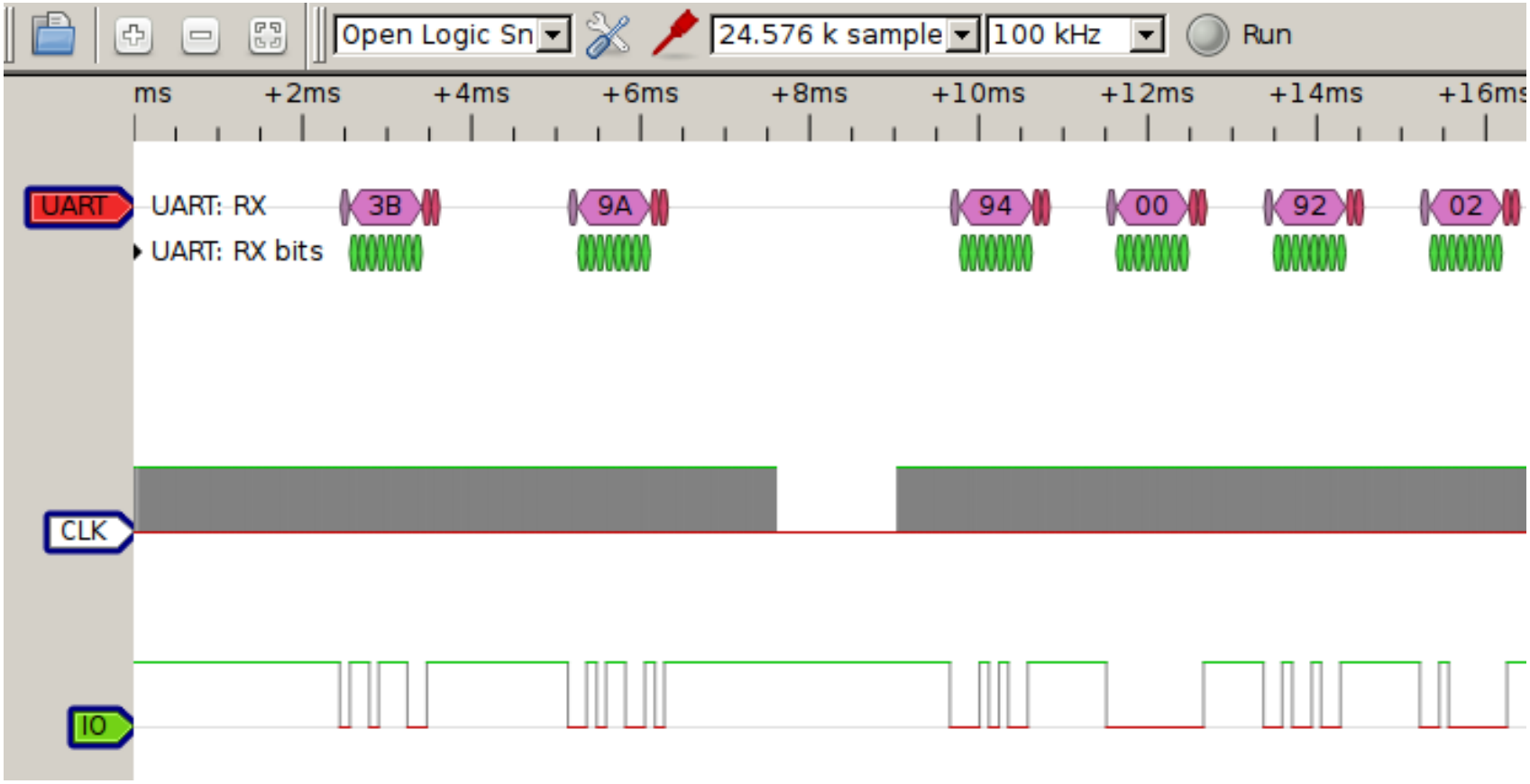


SIM connector pad





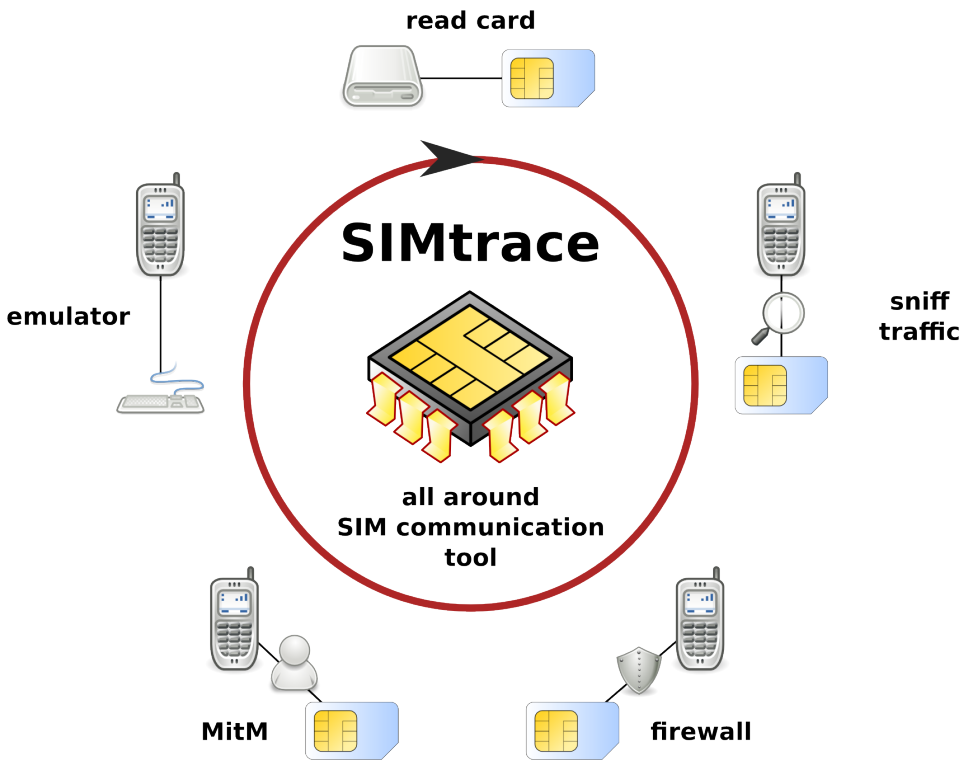
Serial interface



Basic functions

osmocomSIMTRACE

http://simtrace.osmocom.org



monitor, analyze and use the power of SIM

Motivation

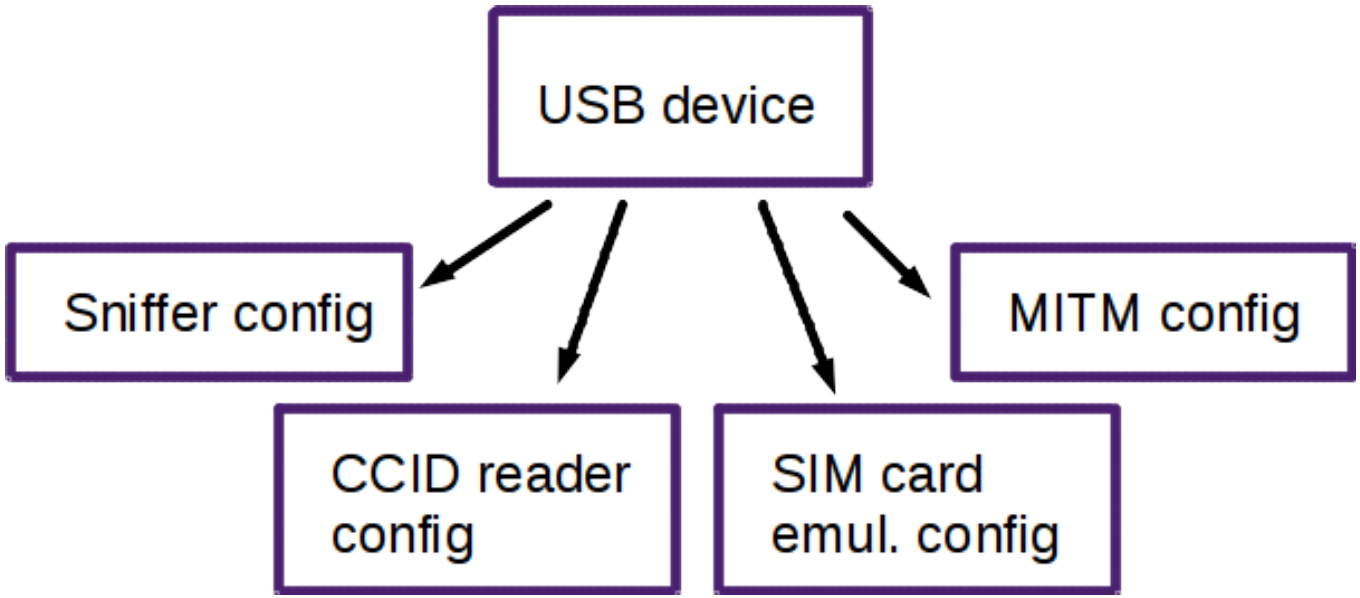
- Upgrade hardware and software to support MITM, CCID reader, smart card emulation mode
- Phone and smartcard can be in different physical spots, connected over the Internet

Hardware Upgrade

Chip	AT91SAM7	AT91SAM3S
ARM Core	ARM7TDMI	ARM Cortex-M3
Instruction set	ARMv4	ARMv7
# USB endpoints	4	8

Table: Endpoints used for each SIMtrace mode

USB configurations



main.c

```
1  typedef struct {
2      void (* configure) ( void );
3      void (* init) ( void );
4      void (* exit) ( void );
5      void (* run) ( void );
6  } conf_func;
7
8  conf_func config_func_ptrs[] = {
9  /*  CFG_NUM_SNIFF */
10     {Sniffer_configure, Sniffer_init, Sniffer_exit, Sniffer_run},
11 /*  CFG_NUM_CCID */
12     {CCID_configure, CCID_init, CCID_exit, CCID_run},
13 /* CFG_NUM_PHONE */
14     {Phone_configure, Phone_init, Phone_exit, Phone_run},
15 /* CFG_NUM_MITM */
16     {MITM_configure, MITM_init, MITM_exit, MITM_run},
17 };
```

Listing 1: Function pointer struct per mode

main.c

```
1   while(1) {
2   /* ...  */
3       if (last_simtrace_config != simtrace_config) {
4           config_func_ptrs[last_simtrace_config-1].exit();
5           config_func_ptrs[simtrace_config-1].init();
6           last_simtrace_config = simtrace_config;
7       } else {
8           config_func_ptrs[simtrace_config-1].run();
9       }
10  }
```

Listing 2: Function pointer struct per mode

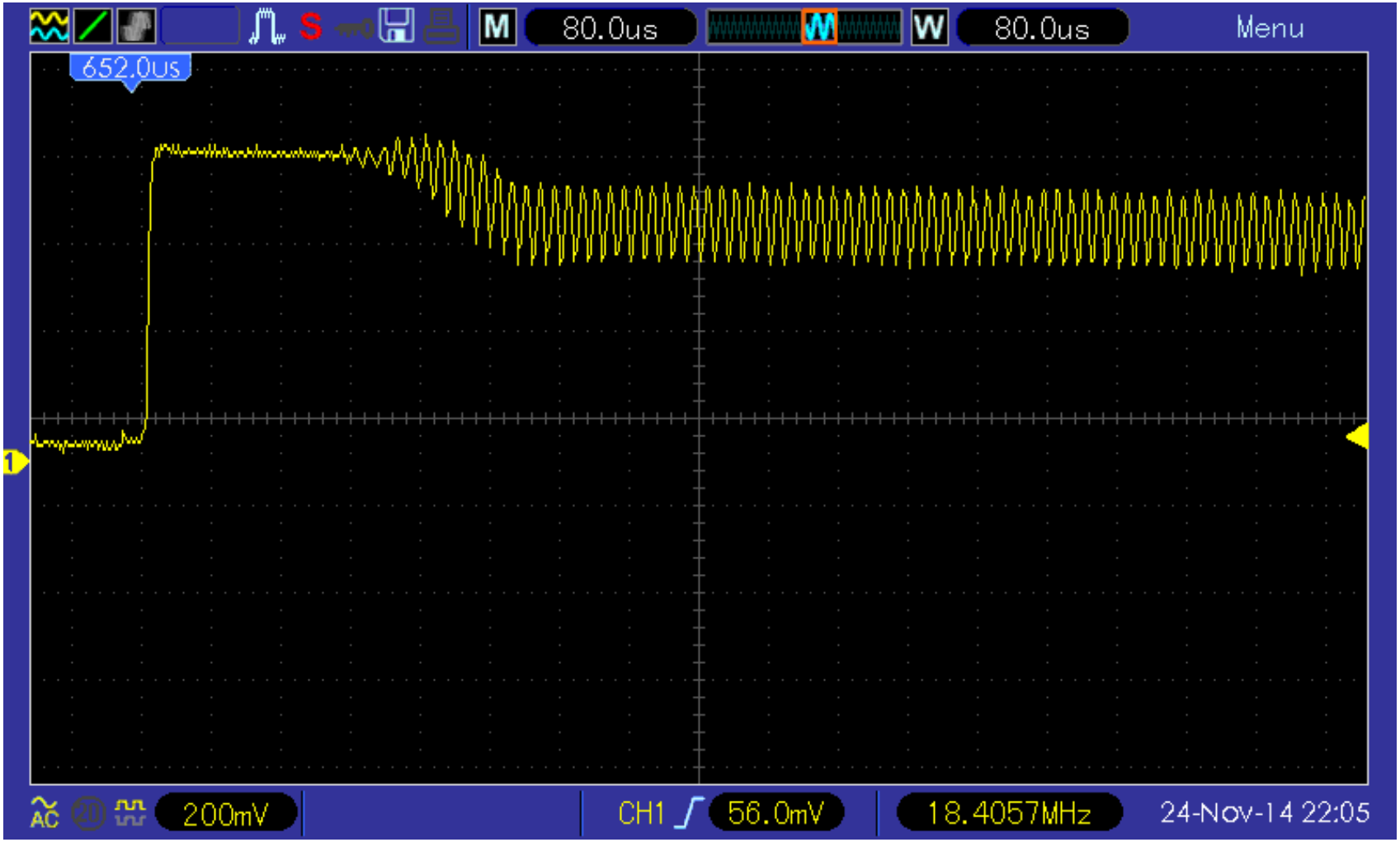
```
# ./simtrace.py -h
usage: simtrace.py [-h] [-C {sniff,ccid,scard_emul,mitm}]
    [-s] [-S] [-p] [-m]

optional arguments:
  -h, --help            show this help message and exit
  -C {sniff,ccid,scard_emul,mitm},
        --conf {sniff,ccid,scard_emul,mitm}
                        Set USB config
  -s, --sniff           Sniff communication!
  -S, --select_file     Transmit SELECT cmd! (Test CCID)
  -p, --phone           Emulates simcard
  -m, --mitm            Intercept communication (MITM)
```

Listing 3: simtrace.py

Work done

- Upgraded hardware: AT91SAM7 $\Rightarrow$ AT91SAM3S
- Implemented at91lib based firmware
- Implemented PoC-MitM attack

Crystal oscillator



Listing 1: Off by two?

```
/* In static void RDRtoPCDatablock\_ATR( void ):     */
        if( length > 5 ) {
-         ccidDriver.ProtocolDataStructure[1] = Atr[5]&0x0F; // TD(1)
-         ccidDriver.bProtocol = Atr[5]&0x0F;              // TD(1)
+         ccidDriver.ProtocolDataStructure[1] = Atr[3]&0x0F; // TD(1)
+         ccidDriver.bProtocol = Atr[3]&0x0F;              // TD(1)
```

Listing 2: Off by one

```
@@ -216,7 +216,7 @@ uint16_t ISO7816_XfrBlockTPDU_T0(const uint8_t *pAPDU,
     ISO7816_SendChar( pAPDU[4] ); /* P3 */

     /* Handle the four structures of command APDU */
-    indexApdu = 4;
+    indexApdu = 5;

         if( wLength == 4 ) {
             cmdCase = CASE1;
```

main.c

Listing 3: Function pointer struct per mode

```
@@ -457,7 +457,7 @@ void USBDDriver_Initialize(
 /*     ... */
     if (pInterfaces != 0) {

-        memset(pInterfaces, sizeof(pInterfaces), 0);
+        memset(pInterfaces, 0, sizeof(*pInterfaces));
     }
```

Listing 4: Diff of ICC_INSERTED_EVENT value

```
  #define ICC_NOT_PRESENT                         0x00
  #define ICC_PRESENT                             0x01
  #define ICC_CHANGE                              0x02
- #define ICC_INSERTED_EVENT                      ICC_PRESENT+ICC_CHANGE
+ #define ICC_INSERTED_EVENT                      0x01

@@ -136,12 +136,18 @@ void CCIDDriver_Initialize( void )
//-------------------------------------------------------------------------
static void RDRtoPCSlotStatus( void )
{
      // Header fields settings
      ccidDriver.sCcidMessage.bMessageType = RDR_TO_PC_SLOTSTATUS;
      ccidDriver.sCcidMessage.wLength    = 0;
-     ccidDriver.sCcidMessage.bStatus    = ccidDriver.SlotStatus;
+     if (ccidDriver.SlotStatus == ICC_INSERTED_EVENT) {
+         ccidDriver.sCcidMessage.bStatus = 0;  /* ICC present and active card */
+     } else if (ccidDriver.SlotStatus == ICC_NOT_PRESENT) {
+         ccidDriver.sCcidMessage.bStatus = 2;  /* No ICC present*/
+     } else{
+         TRACE_ERROR("Strange bStatus");
+         ccidDriver.sCcidMessage.bStatus = 0;
+     }
```

Video time!

Code is (hopefully) goint to be released soon...ish! (Ping me: simtrace(a)chrysh.de)