About...

Specter :

* An entity of service
* Enjoys reversing black-box hardware

About...

Specter :

* An entity of service
* Enjoys reversing black-box hardware
* Heating Engineer

Play video1.mp4

Stephen Chavez introduces Himself in SAS2017 clip

YOUR READING LIST


This Guy Let Me Control His Hacked Wheelchair With An Xbox Gamepad


BOX CEO Aaron Levie Talks Trump, Tech, And How To Stay Nimble As A Public Software Company

Active on LinkedIn


Apple's Cunning Trick To Make You Buy The New iPhone 7S

Active on LinkedIn

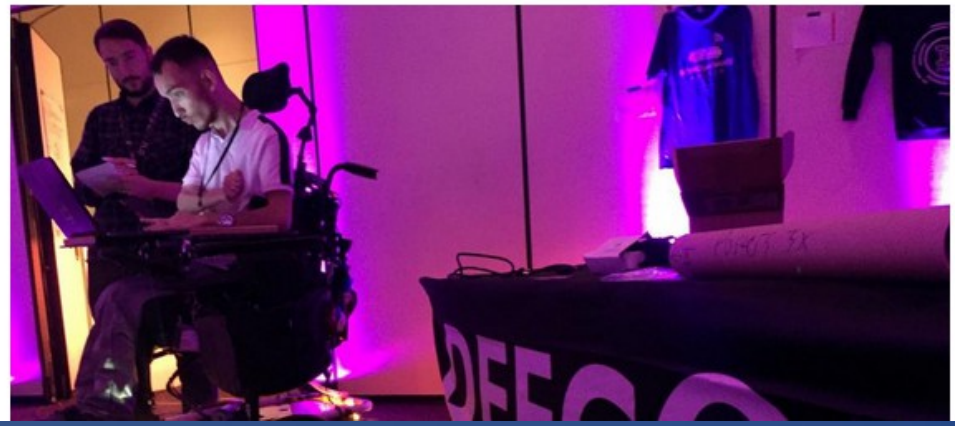
AUG 6, 2016 @ 03:50 PM 2,757

The Little Black Book of Billionaire

This Guy Let Me Control His Hacked Wheelchair With An Xbox Gamepad

Thomas Fox-Brewster, FORBES STAFF

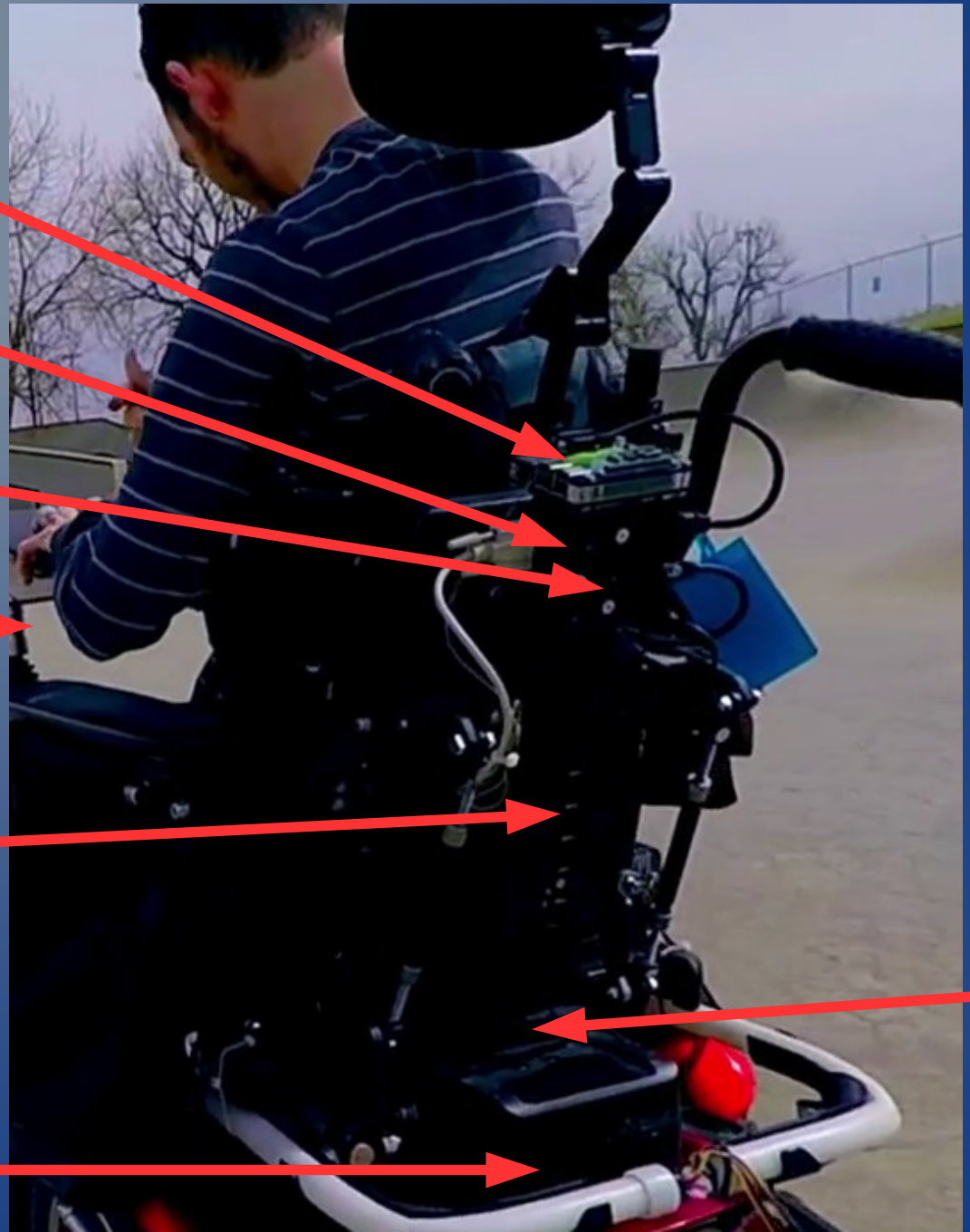
I cover crime, privacy and security in digital and physical forms. FULL BIO



GeekPwn 2016, Shanghai

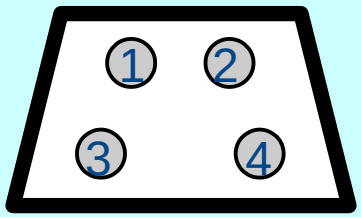

Forbes
YOUR READING LIST
This Guy
His Hack
With An
BOX CE
Talks Tru
How To
Public Se
Active on Linked
Apple's
Make Yo
iPhone 7
Active on Linked
Kaspersky SAS 2017
HAT IS R-NET/CANBUS?
R-NET Protocol
On-the-fly-configuration
Advanced motor control algorithms
Uses special ID's to allow chair modules to act on certain messages
CAN BUS (Controller Area Network)
Vehicle protocol standard
Network based
Any device on the same CAN BUS network can send/receive CAN messages
Has no security/authentication
– Inject our own messages
Kaspersky
SECURITY
ANALYST
SUMMIT

R-net devices


Pi3
Output Module
Bluetooth Module
Joystick Module
R-net Hub
Seating / Lighting Module
Power Module

R-Net interface and topology

Female / Device side



1) CAN Lo
2) CAN Hi
3) +24VDC
4) GND(-)

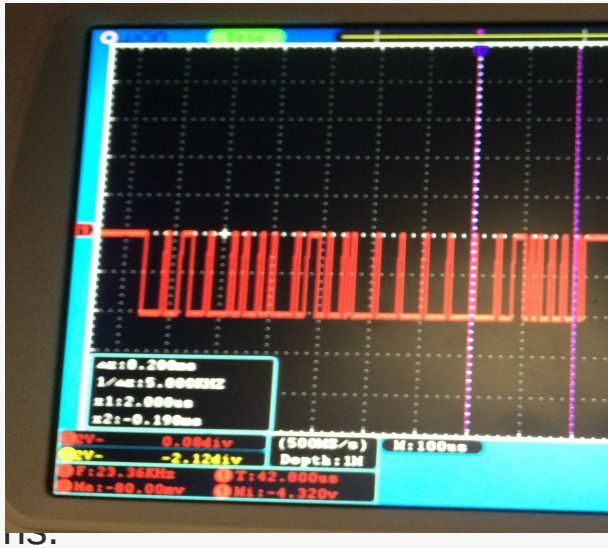
PM

Hub

Pi3 + PiCan2

R-NET rides on CANBUS 2.0B

Differential pair. Dominant and recessive bits.



dominant is a logical 0 (actively driven to a voltage by the transmitter)
recessive is a logical 1 (passively returned to a voltage by a resistor)

Frame oriented. IDs: 11bits(standard frame)

11+18bits(extended frame). Data can be 0 to 8 bytes.

Speeds: R-net is at 125Kbps. Max 1Mbps for Can 2.0B

FrameID represents message priority.
If multiple messages attempt to xmit at the same time, the lowest ID wins.

Protocol chips do the work.
CAN protocol is built in to many SOCs (Beaglebone) and MCUs(ARM Cortex M3/M4.)

Acknowledge bit (@ end of frame) is set by any receiving device.

Errors in transmission can be instantly detected. We tried bit banging to kill frames.
This instantly causes an error condition and the frame is resent (no timeout).

There are no addresses implicit in CAN protocol.
This makes it difficult to determine what is source/destination.

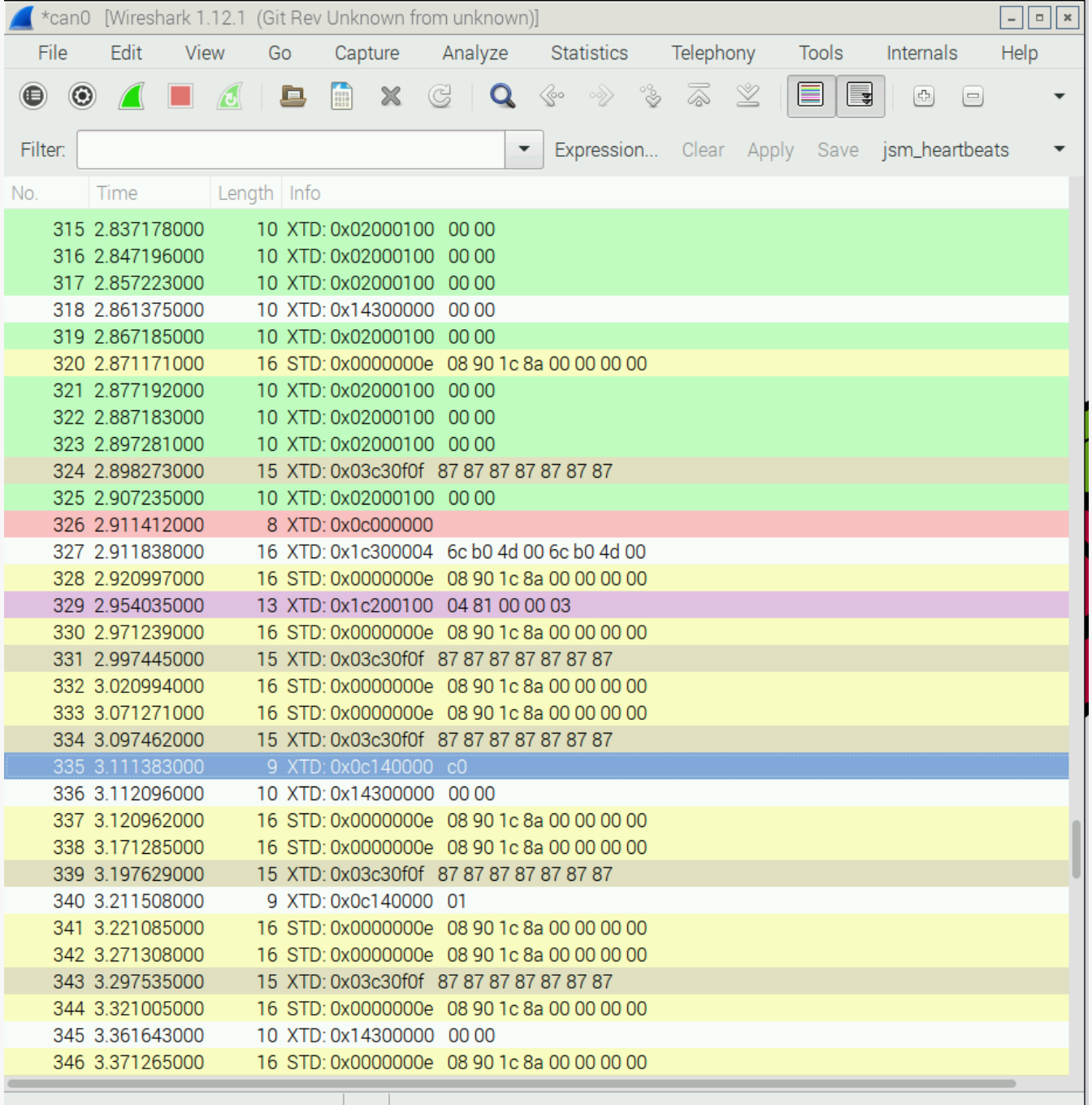
R-NET CAN frame examples

Horn beep:
$ cansend can0 0C040100# ;sleep .2; cansend can0 0c040101#

Set maximum power to 50%:
$ cansend can0 0A040100#32; cansend can0 181c0100#0260000000000000

Random battery levels:
$ cangen can0 -I 1C0C0100 -L 1 -e -g 100

Change from mode “0” to mode “1”:
$ cansend can0 061#40400000; sleep .1; cansend can0 061#00410000



JSMerror exploit

Green = JoyXY frames
Yellow = JSM heartbeats
Red = Injected frame

JSM is in “drive” mode
Outputs JoyXY frames...
until a JSM network error is triggered.

JSM continues to output heartbeat frames but stops outputting JoyXY frames.
At the point of error we can take up the rhythm with injection.

Synchronizing our spoofed JoyXY frames may be done by clocking the last JSM JoyXY frame prior to inducing the JSM error.

Reversing R-NET

STARTUP and NETWORK CONFIG frames:

```
     000#R                :PMtx sleep all devices
     002#R                :PMtx sleep all devices
     00C#                 :JSMtx test canbus connection.  Checks for ack on bus prior to JSM wake
     04M#00000000         :JSMrx select modemap M for parameter exchange.  See: 78M#...  causes
     04M#80000000         :JSMtx end parameter exchange for mode M.
     7B3#                 :PMtx global request for configuration mode
     7B1#                 :PMtx drop to config mode 1
     7B0#                 :JSMtx PMtx drop to config mode 0 --- ends capability
PARAMETER EXCHANGE frames:
     78M#2P810000Xx00Vv00:JSMtx check if pointer Xx sub Vv exists
     79M#4P81000000000000:PMtx yes, pointer exists
     79M#CP81000000000000:PMtx no, pointer does not exist
     79M#2P8C0000asciitxt:PMtx text chunk used for cJSM display messages.  Only prese
     78M#4P8F000000000000:JSMtx request "pointer" from PM.  Pointer address set with 78M#2P81...
     79M#2P8F0000XxYy0000:JSMtx XxYy = "pointer" returned by PM. Response to 78M#408F000000000000
     79M#C1810000280000000:PMtx Error: address not found.
     78M#208000001M000000:JSMtx programming header issued prior to capability

SERIAL NUMBER enumeration/confirmation:
     1FRSTtUu#                  :JSMtx/rx PMtx/rx SerialNumber exchange. R=Subsequence
```

Play video2.mp4

POC demo @ SSD Jul '16

SmartWheels

[EECS 149/249A Class Project]

Tomás Vega
tomas.vega@berkeley.edu

Corten Singer
cortensinger@berkeley.edu

James Musk
jamesmusk@berkeley.edu

Yash Shah
yshah@berkeley.edu

Department of Electrical Engineering and Computer Science
University of California
Berkeley, CA

ABSTRACT

We develop a self-driving, target-following, obstacle-avoiding wheelchair and discuss the design considerations that were incorporated into the creative process. We use a Raspberry Pi 3 Model B with a PiCAN 2 shield to send commands to the wheelchair via the R-net protocol. An iPhone application is used to track AprilTags (2D barcodes developed for robotics applications) with its camera and send data to the Raspberry Pi. Finally, we use three ultrasonic sensors attached to the Raspberry Pi to detect obstacles.

control of his wheelchair. He was eager to share his research [6] with us and help us gain access into our wheelchair's control system. He was also the motivation behind the features we wanted to implement in our smart wheelchair. Stephen suffers from bilateral open-cleft schizencephaly, a rare brain condition that leaves him non-verbal and paralyzed except for one hand. Thus, Stephen cannot speak and navigate simultaneously. Our solution allows Stephen to let his wheelchair autonomously follow his friends while avoiding obstacles so that he can participate in conversation without the burden of stopping each time he wants to contribute. Further, our intelligent wheelchair design has incredible potential for visually impaired wheelchair users who have trouble navigating in unfamiliar terrain. Globally, there are 39 million blind people and 246 million have low vision [5]. It is estimated that 1 in 10 visually impaired

Play video3.mp4

SmartWheels Demo

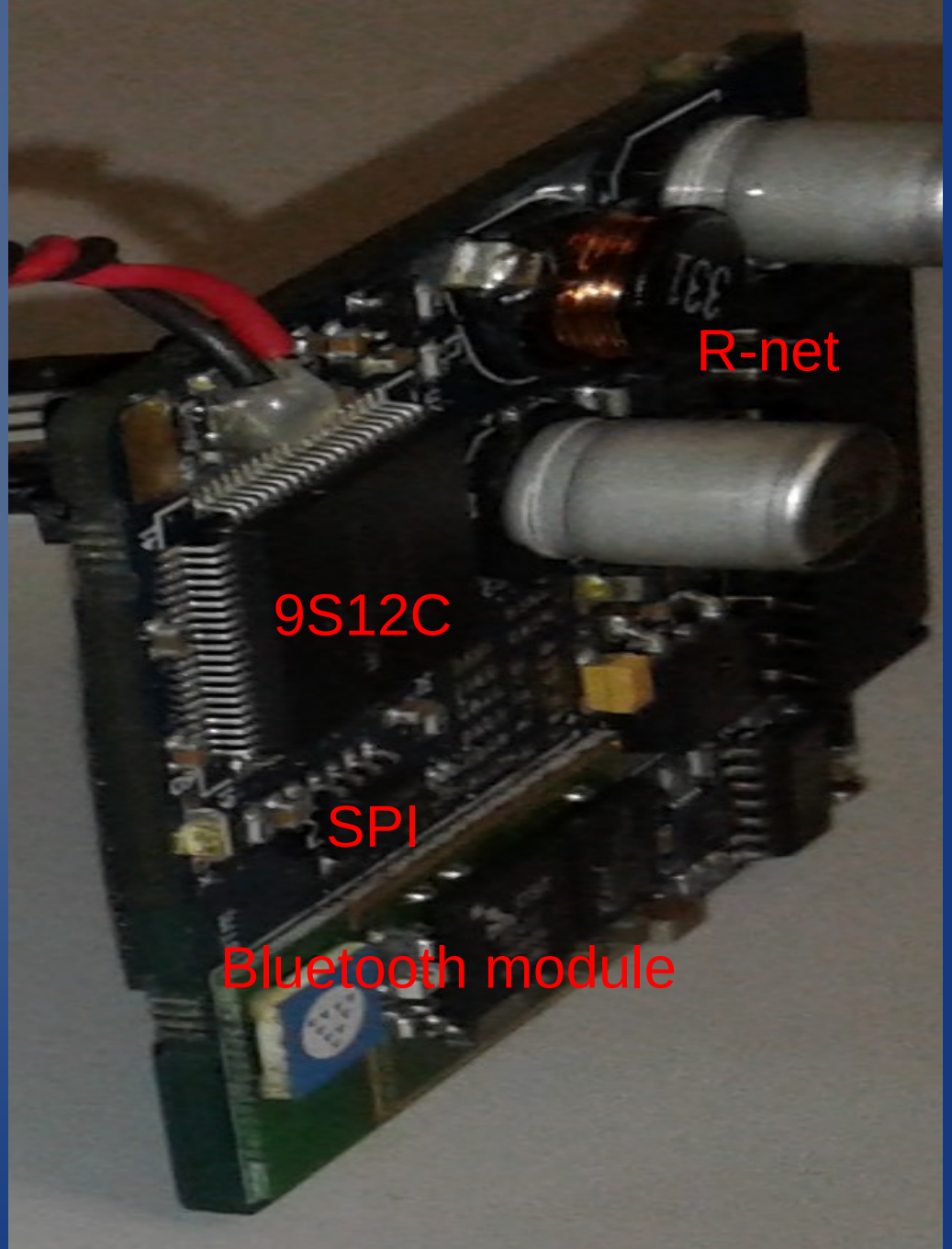
What about a headline grabbing remote exploit without modifications to the chair?

The only R-net wireless device


D51111.03
BP12060172
PG DRIVES TECHNOLOGY LTD
MADE IN ENGLAND
BT-MOUSE
Mouse module
"iDevice" module
SW1
SW2

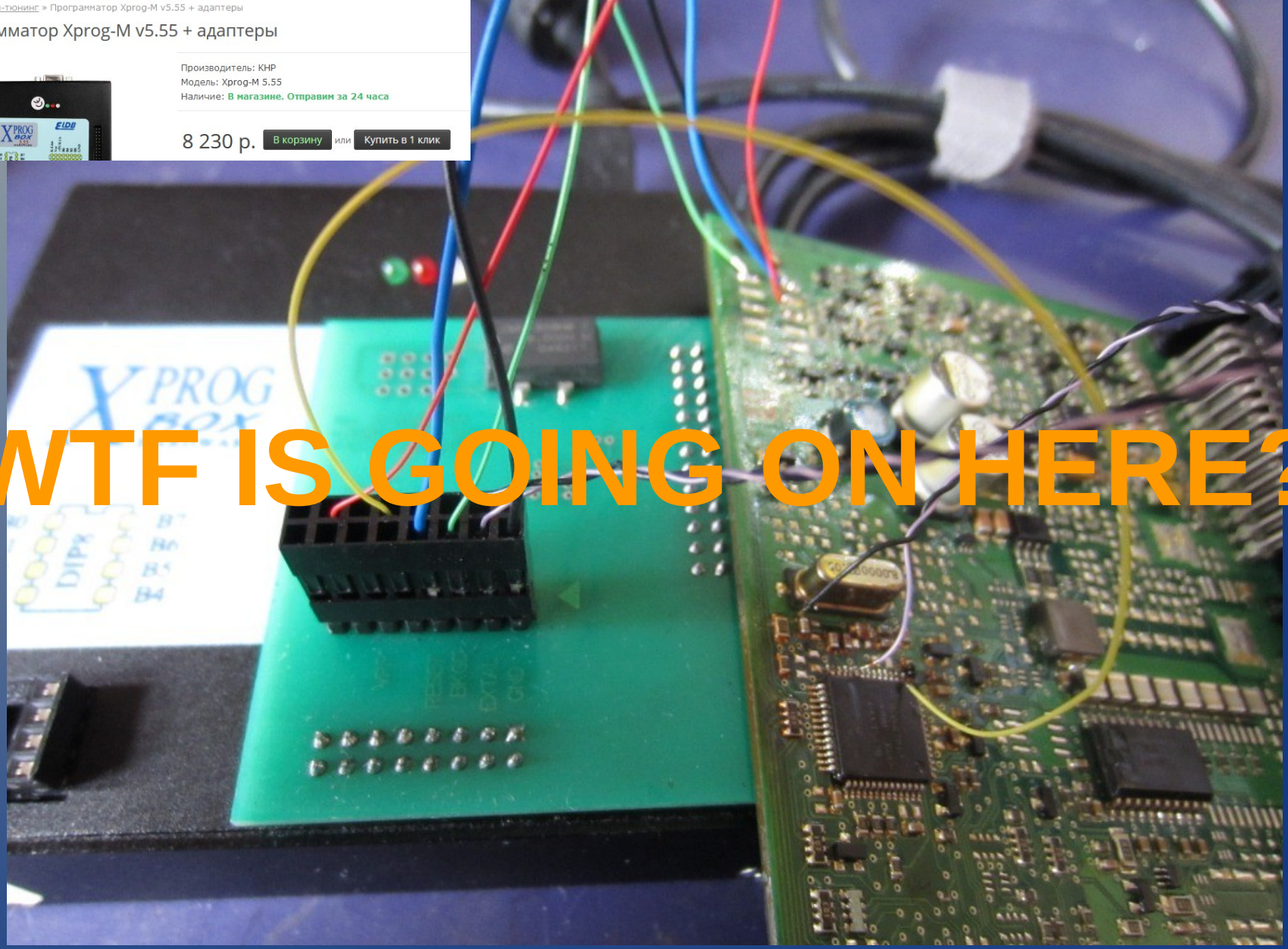

9S12C
LED JSMs
Monochrome JSMs
ARM7
Color JSMs
Toggles & Pots
Omni
EL PM90
80(90)
5
Seating Module
MC908 +
56F83 DSP
PM80/120
Encoder Module
Attendant
2L
4L
6L
Intelligent Seating/
Lighting Module
2
4
6
Intelligent Seating
Module
XC164 Infineon
Lighting Module
MC908GZ
Output Module
9S12C
Bluetooth Module

R-net Bluetooth Mouse Internals



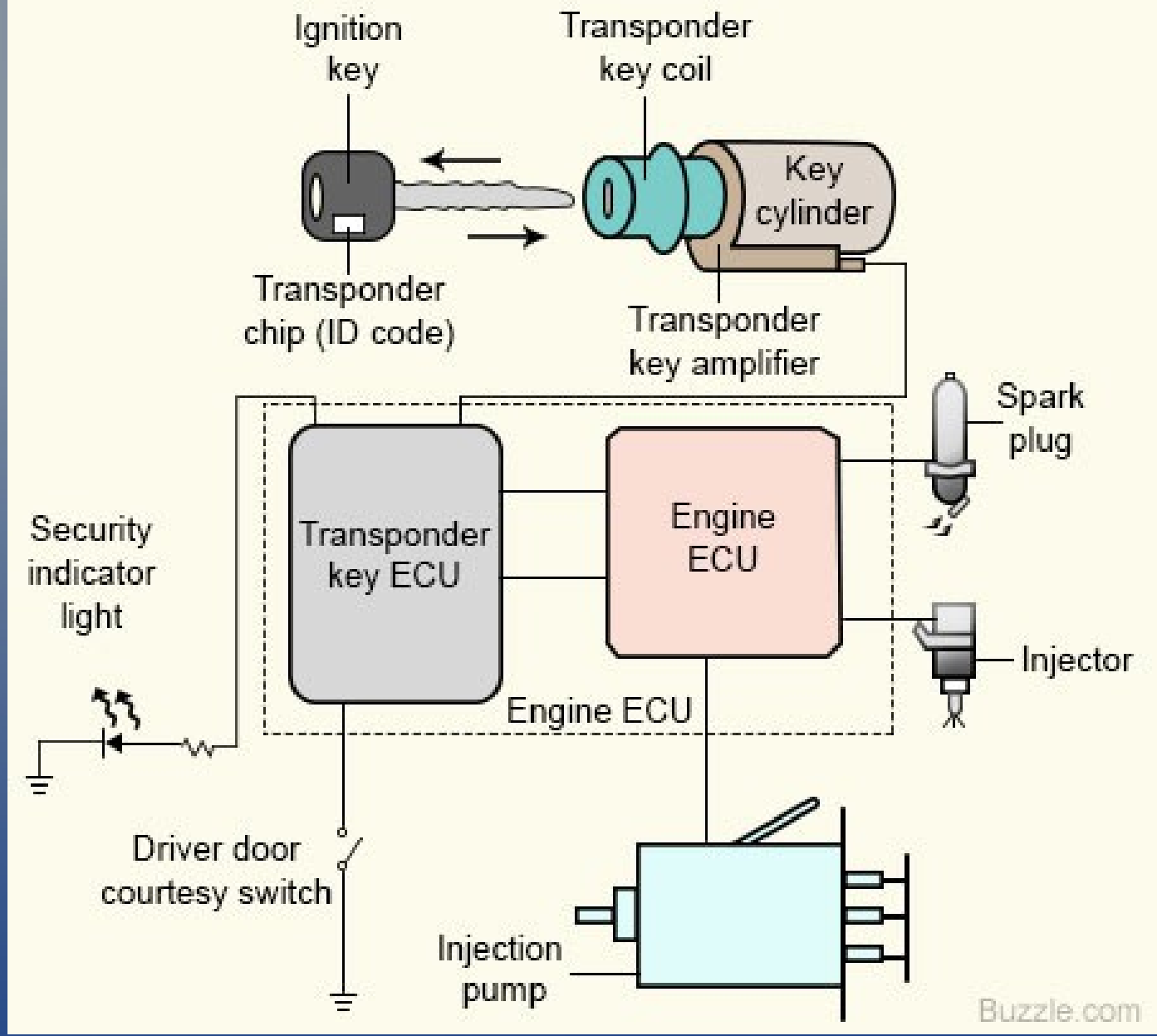
Google Image search: “MC9S12”

some interesting results


Главная » Чип-тюнинг » Программатор Xprog-M v5.55 + адаптеры
Программатор Xprog-M v5.55 + адаптеры
Производитель: KHP
Модель: Xprog-M 5.55
Наличие: В магазине. Отправим за 24 часа
8 230 р.
В корзину
или
Купить в 1 клик
WTF IS GOING ON HERE?


G-POWER

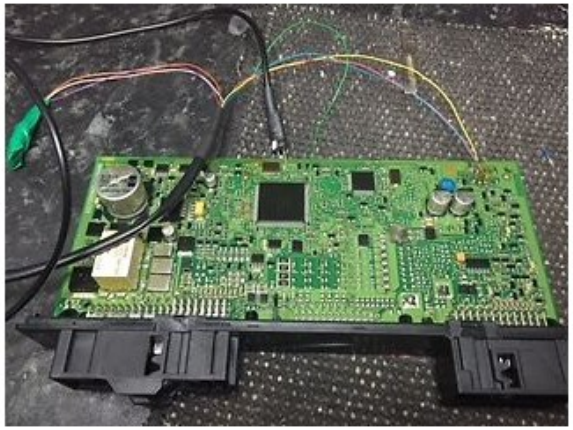

BMW
CAS4


Ignition key
Transponder key coil
Key cylinder
Transponder chip (ID code)
Transponder key amplifier
Spark plug
Security indicator light
Transponder key ECU
Engine ECU
Injector
Engine ECU
Driver door courtesy switch
Injection pump
Buzzle.com

Send ECU + $$ = Get FLASH

ebay Shop by category Search... All

Back to search results | Listed in category: eBay Motors > Automotive Tools & Supplies > Diagnostic Service Tools > Other Diagnostic Service Tools



BMW CAS4 READING SERVICE 100% SUCCESS

Item condition: **New other (see details)**

Quantity: 1 3 available / 1 sold

Price: **GBP 120.00**
Approximately US $149.85

Buy It Now

Add to cart

17 watching

Add to watch list

Add to collection

5 inquiries	Longtime member	Fast and safe shipping

Shipping: GBP 16.99 (approx. US $21.22) Expedited Shipping to United States | See details

Item location: Leeds, West Yorkshire, United Kingdom

Ships to: United Kingdom and many other countries | See details

Does not lift legs of mcu.

Clock injected at component

HELLO AND THANKS FOR LOOKING AT LISTING
ON OFFER IS A CAS 4 READING SERVICE

ARE YOU HAVING TROUBLE READING CAS 4 MODULES WITH 5M48H & 1N35H UP TO 2015 ?

Remove cas 4 & send it or bring it to us here at Leeds West Yorkshire
and we will do the rest .
We won't lift any legs or cut any tracks or damage your cas 4 in any way
We can read data flash & program flash & will send you files to make key if your an auto locksmith
Or we can send it back with info changed to what you need for dash correction
(If you want cluster doing too send me eeprom or eeprom data and i can change for you at extra cost)

We will test it first on bench then afterwards before shipping
Please only send ones that have not been attempted to read first
or you will be charged to test it & if it fails will be charged for return postage .

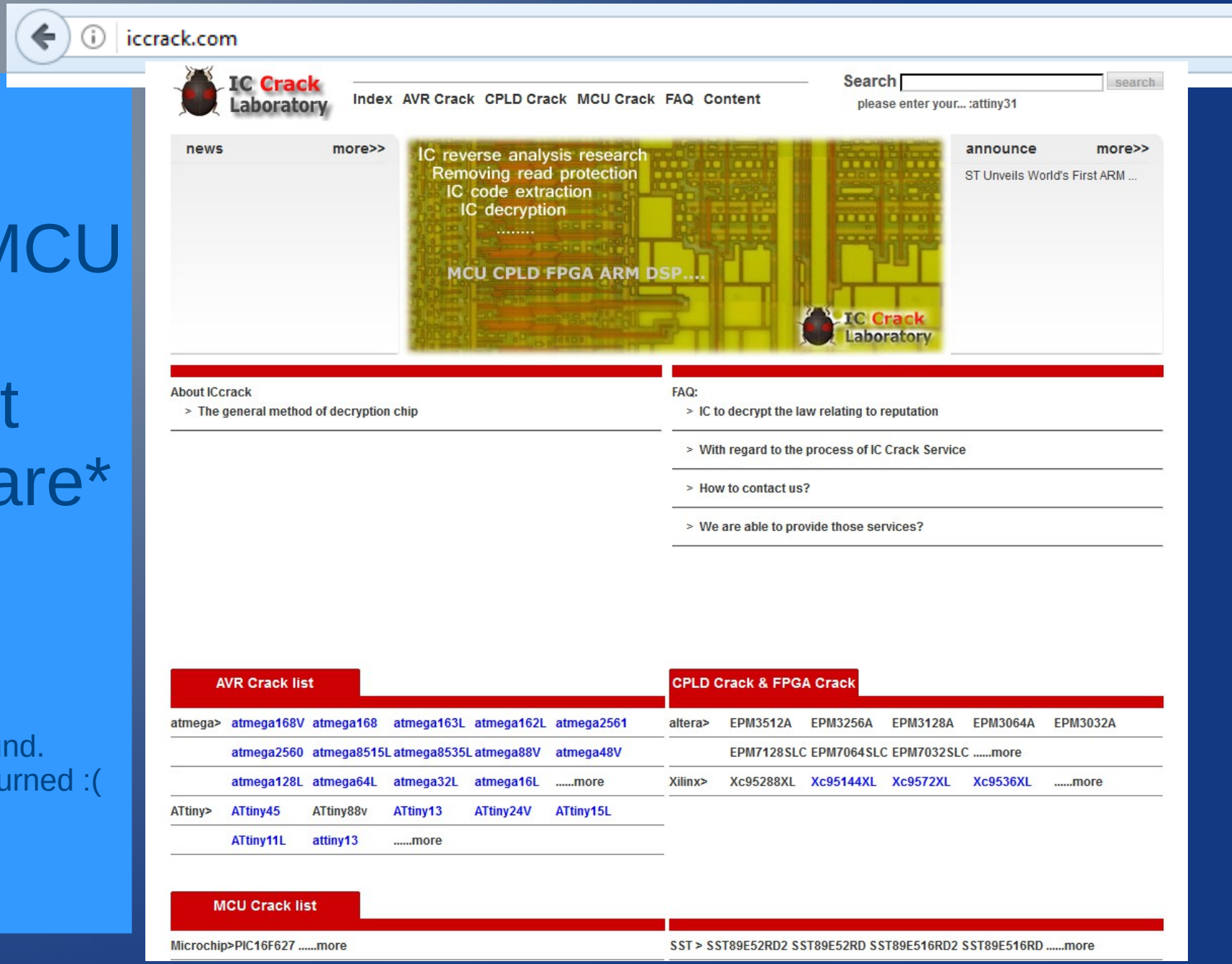
We also offer same day service inclusive of fully insured special delivery return postage

Any info contact Anthony on 077707 23332

Thanks for looking

Send MCU = Get Firmware*

* or full refund.
Chip is not returned :(



MCU crack services

MCU Reverse Services

- MCU Reverse Engineering
- Break Copy Protection in MCU
- Atmel MCU Crack
- MICROCHIP MCU Attack
- CYPRESS MCU Attack
- Freescale IC Crack
- NXP IC Attack

Extraction
U Break
ode Extraction
Unlock
Crack

Buy From Us?

HING DAHL TECHNOLOGY HK CO.,LTD

Sitemap | Company profile | MCU crack Blog

ecryption | MCU crack unlock/SCM crack unlock | PCB clone | IC Programmer | Service/FAQ | IC unlock news

IC unlock crack decipher Service FAQ and Service

We work on IC MCU unlock crack break decipher Code extraction 10 years,Until now we service for many countries from all over the world customer,10 years decipher experience, High speed,High success rate, High quality in communitiesofpractice! so blive us and choose us, you are good lucky!

Service Process:

1. Please send an inquiry by email, what's IC MCU CHIP MODEL you have;

hddworld HING

CU crack Blog

Home | IC unlock crack/IC Chip decryption | nlock news

Home

Difficult chip decryption:

- C8051 chip crack unlock
- Samsung MCU unlock crack
- STC_STC12C unlock
- Cypress/Cy8c MCU crack
- SPMC65 SPMC75
- MSP430
- CPLD un
- ARM MCU
- MASK Ro
- DSP Chip
- IC MCU Pro
- Hilo Syste
- Bee prog
- EasyPro

The buyer is liar: (veryone take care and remenber this liar)

Name:Lin

Address:via togni, 6, brescia, Brescia, Italy

Phone:00393333336633

The cheat bought goods then return back wrong and bad things to us.

CIRCUIT engineering

Home | Who We Are | Frequently Asked Questi

MCU Crack | DSP Crack | AVR Crack | CPLD Crack | FPGA Crack | IC Crack

Keywords | Search

You are here: IC clone, MCU Crack, Microcontroller Unlock Service Provider

crack mcu heximal eeprom crack m

Product Categories

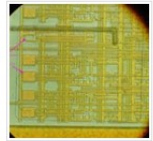
- MCU Crack
- DSP Crack
- AVR Crack
- CPLD Crack
- FPGA Crack
- IC Crack

Live Support Chat

Customer Testimonials

"We are very pleased with the

Hot Products



Crack MCU PIC12C672 Heximal

Copy Protected PIC12C509A Code

Crack IC PIC18F2221 Firmware

Extract Chip PIC16F527 Heximal

break-ic.com

Microcontroller reverse engineer

Everything they make, We can break!

HOME | COMPANY | PCB COPY | MCU HACK | FAQ | CONTACT US | Disassembler Software

WHY US ?

World first mcu hack company

- In business since 1998
- Reversed tens of thousands of chips
- Copied thousands of pcbs
- Foreseen all pertential problems
- Integrity with payments

Learn More

Thanks MikaTech ! Nice art!

Send MCU = Get Firmware*

* or full refund.
Chip is not returned :(

iccrack.com

IC Crack Laboratory

Index AVR Crack CPLD Crack MCU Crack FAQ Content

Search search

please enter your... :attiny31

news more>>

IC reverse analysis research
Removing read protection
IC code extraction
IC decryption
.....
MCU CPLD FPGA ARM DSP....

announce more>>

ST Unveils World's First ARM ...

About ICcrack
> The general method of decryption chip

FAQ:
> IC to decrypt the law relating to reputation

AVR Crack list

atmega>	atmega168V	atmega168	atmega163L	atmega162L
	atmega2560	atmega8515L	atmega8535L	atmega88V
	atmega128L	atmega64L	atmega32L	atmega16L
ATtiny>	ATtiny45	ATtiny88v	ATtiny13	ATtiny24V
	ATtiny11L	attiny13	.....more	

MCU Crack list

Microchip>PIC16F627 .....more

HING DAHL TECHNOLOGY HK CO.,LTD

Sitemap

ecryption | MCU crack unlock/SCM crack unlock | PCB clone | IC Programmer

vice

IC unlock crack decipher Service FAQ and Ser

We work on IC MCU unlock crack break decipher Code extraction 10
for many countries from all over the world customer,10 years deciph
speed,High success rate, High quality in communitiesofpractice! so
are good lucky!

Service Process:

Home About Us MCU Crack PCB Cloning Contact us

Chip Works

MCU Reverse Enginering

Chip Works Technology Limited

- PCB Reverse Engineering and Cloing
- PCB Assembly Prototyping
- PCB Manufacturing
- Atmel Microcontroller Crack
- Microchip IC Crack
- Altera Chip Decryption
- Toshiba Microcontroller Cloning
- Winbond IC Dump
- NEC MCU Crack

For Car Industry

Chip Works helped global clients to copy a wide range of tools for car diagonsis, auto locksmith, engine tuning

Learn More

MICROCONTROLLER HACK SERVICE

Actel, ATMEL, ALTERA, CYPRESS, CORERIVER, CYGNAL, FUJITSU, freescale, HOLTEK, HYUNDAI, LG, MAXIM, MDT, MYSON, MXIC, NEC, NXP, PHILIPS, SAMSUNG

MOTOROLA Series MCU reverse engineering

34. MC68HC11/711 Series: MC68HC11A8(AB95T), MC
MC68HC11E9(1B60R), MC68HC11E9(D82R), MC68HC1
MC68HC11F1(2F37E), MC68HC11F1(E87J), MC68HC11
MC68HC11KA4(1E59B), MC68HC11KS2(0H95B), MC68
MC68HC711E20(1H19S), MC68HC711E9(4K81H), MC6
MC68HC711K4(K59D), MC68HC711PH8(0H30R), MC68
MC68HC11P2(2E74J), MC68HC711P2(1E53M)...
35. MC68HC705 Series: MC68HC705C4, 68HC705C4,
MC68HC705C8A, 68HC705C8A, MC68HC705C9, 68HC
68HC705SR3, MC68HC705J1A, 68HC705J1A, MC68HC
MC68HC705X16, MC68HC705X32, MC68HC05B6, MC6
MC68HC05X32
36. MC68HC908 Series: MC68HC05H12(0H57A), MC68
MC68HC908AZ60A(3K85K), MC68HC908AS60(3J74Y),
MC68HC08AZ60(1J35D), MC68HC08AZ32(0J66D), MC6
MC68HC08AZ32A(1L52H), MC68HC908QT1, MC68HC9
MC68HC908QY2, MC68HC908QY3, MC68HC908QY4, M
37. MC68HC912/9S12 Series: MC68HC912DC128A(3K
MC68HC912DG128A(3K91D), MC68HC912DG128(0K5
MC68HC912D60(0K13J), MC68HC912D60(4F73K), MC
MC9S12DG128B(0L85D), MC9S12DT128B(0L85D), MC
MC9S12DT128B(1L85D), MC9S12DG256C(2K79X), MC
MC9S12H256(1K78X), MC9S12DP256C(2K79X), MC9S

ORANGE5

Orange5 is a professional programming device for memory and microcontrollers.
Unique feature of the current series programmers is built-in macrolanguage for writing down protocols, which gives fast and easy capability to add new types of microschemes, precisely meeting manufacturers' requirements to read/write algorithms.

Orange 5 base set- 350 EURO whitout including VAT

Additional Adapters for Orange 5

Additional Software for Orange 5

Technical Info

- USB power supply (USB2.0/3.0).
- Universal easy to plug panel ZIF16 for EEPROM
- Control of contacts in the sockets

Location: /CARPROG/Software/MCU

S2.4 - CarProg Motorola (Freescale) 912 and 9S12 MPU programmer

Product Information

CarProg Motorola and Freescale 912 and 9S12 series MPU internal EEPROM programmer.

Supported MCUs:

912B32 (1H91F, 0J38M)
912B32 (3H91F)
912B32 (4J54E)
912B32 (9H91F)
912D60 (0K13J)
912D60 (0K75F)
912D60 (4F73K)
912DB128 (0L85D)
912DC128 (0K50E)
912DG128 (3K91D)
912DG128 (5H55W)
9S12H128 (1K78X)
9S12DB128 (0L85D)
9S12DG128 (1L59W)
9S12DG256B (1K79X)
9S12XDP512 (0L15Y)
9S12D64 (2L86D)
9S12DT128 (3L40K)

We always try to do one step ahead.....

Special Offers | Bookmark | Contact | Sitemap

ELDB

Home | Log In | Account | Basket | Checkout

Keywords | All Categories | Go » | Advanced Search »

Euro

SHOPPING CART

0 items

CATEGORIES

- Metalworker Tools
- XPROG
- Misc products

CATEGORIES INFO

- Mini Lathe/Mill
- NEWS
- XPROG
- DOWNLOADS
- FAQ

Welcome to ELDB electronics store

Padelis Floudas
@ Edward Karpicz on Feb 5, 2014 1:30 PM

For urls on programmers that do bypass the security on these MCUs is following:
http://www.xprog-m.com/product.php?id_product=50
http://www.scorpio-lk.com/eng/orange5_main_eng.html
and at the bottom there is a "general recommendation":
http://www.eldb.eu/index.php?route=information/information&information_id=15

These tools are sold as vehicle tools many years now and they are worth the money. There are more tools that do the same thing like "UPA programmer".
The data that i have would be garbage if i reprogrammed the unit with those data and it wouldnt work. You could say that "how do you know that the data are erased?" and the discussion would go on for ever.

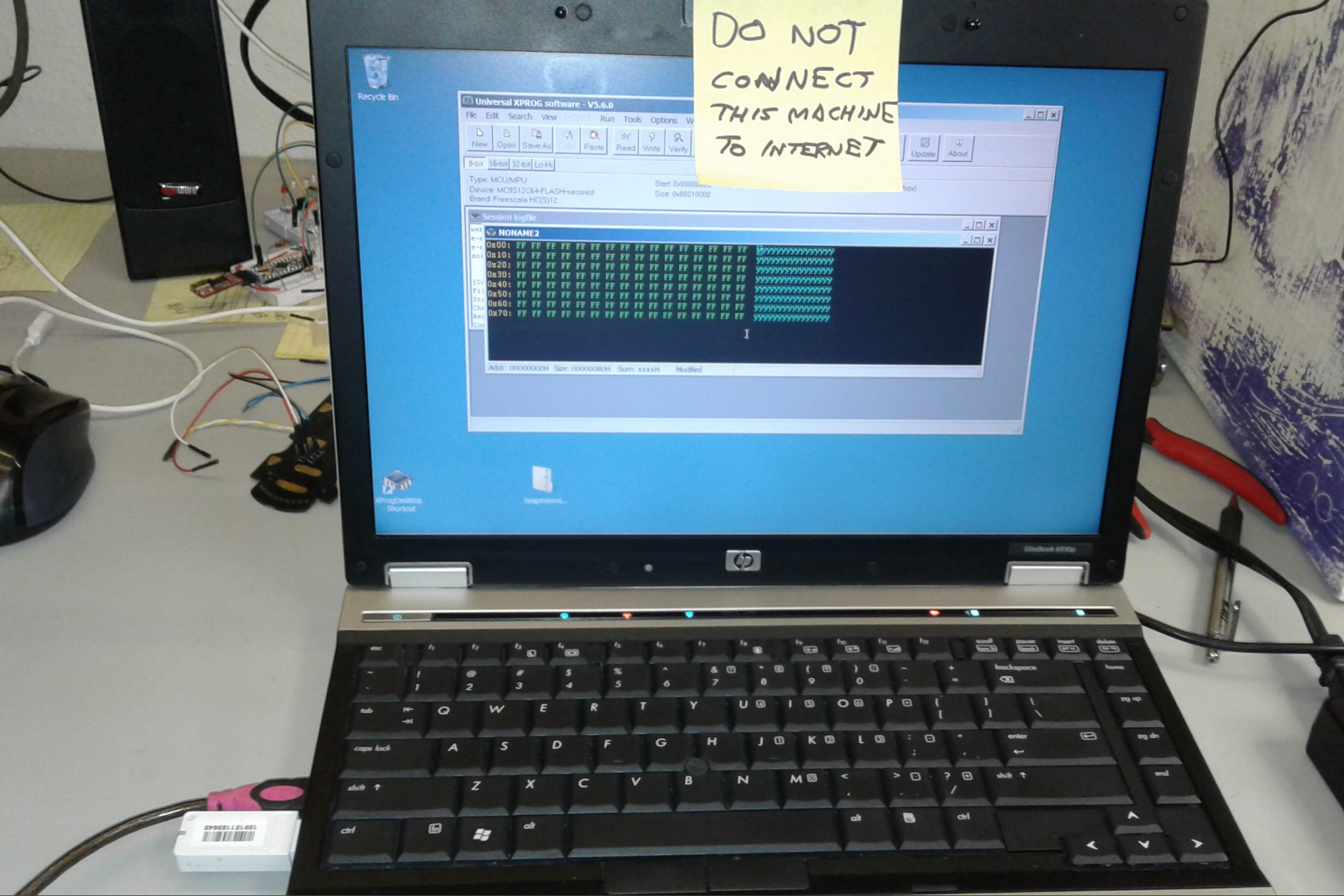
What is strange is that as an Electrical Engineer and programmer i cant understand how these tools bypass the security. This is why i started this discussion here. More brains work better than one.

lama NP
@ Padelis Floudas on Feb 7, 2014 7:02 AM

Hi,
I hope you understand I will not continue with thinking how to hack the MCU.

Also to Edward:
First of all, there's absolutely no 100% secure application, micro, device, etc. If there's enough demand to "hack" something, it will be done. Just ask Microsoft and the XBOX, Apple and the iPhone or the DVD industry. Saying that our parts are 100% secure could be a lie. So, the answer on the question is really not important

"as an Electrical Engineer and programmer ***i cant understand how these tools bypass the security.*** This is why i started this discussion here."


DO NOT
CONNECT
THIS MACHINE
TO INTERNET
Recycle Bin
Universal XPROG software - V5.6.0
File Edit Search View Project Run Tools Options W
New Open Save As Copy Paste Read Write Verify Update About
8-bit 16-bit 32-bit Lo-Hi
Type: MCU/MPU
Device: MC9S12C64-FLASH-secured
Brand: Freescale HC(S)12
Start: 0x00000000
Size: 0x00010000
Session logfile
NONAME2
0x00: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
0x10: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
0x20: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
0x30: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
0x40: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
0x50: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
0x60: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
0x70: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
Addr: 00000000H Size: 00000080H Sum: xxxxH Modified

Cracked Xprog software

Notice:

1. If your X-Prog is older version,cannot use this newest dongle to update to V5.60

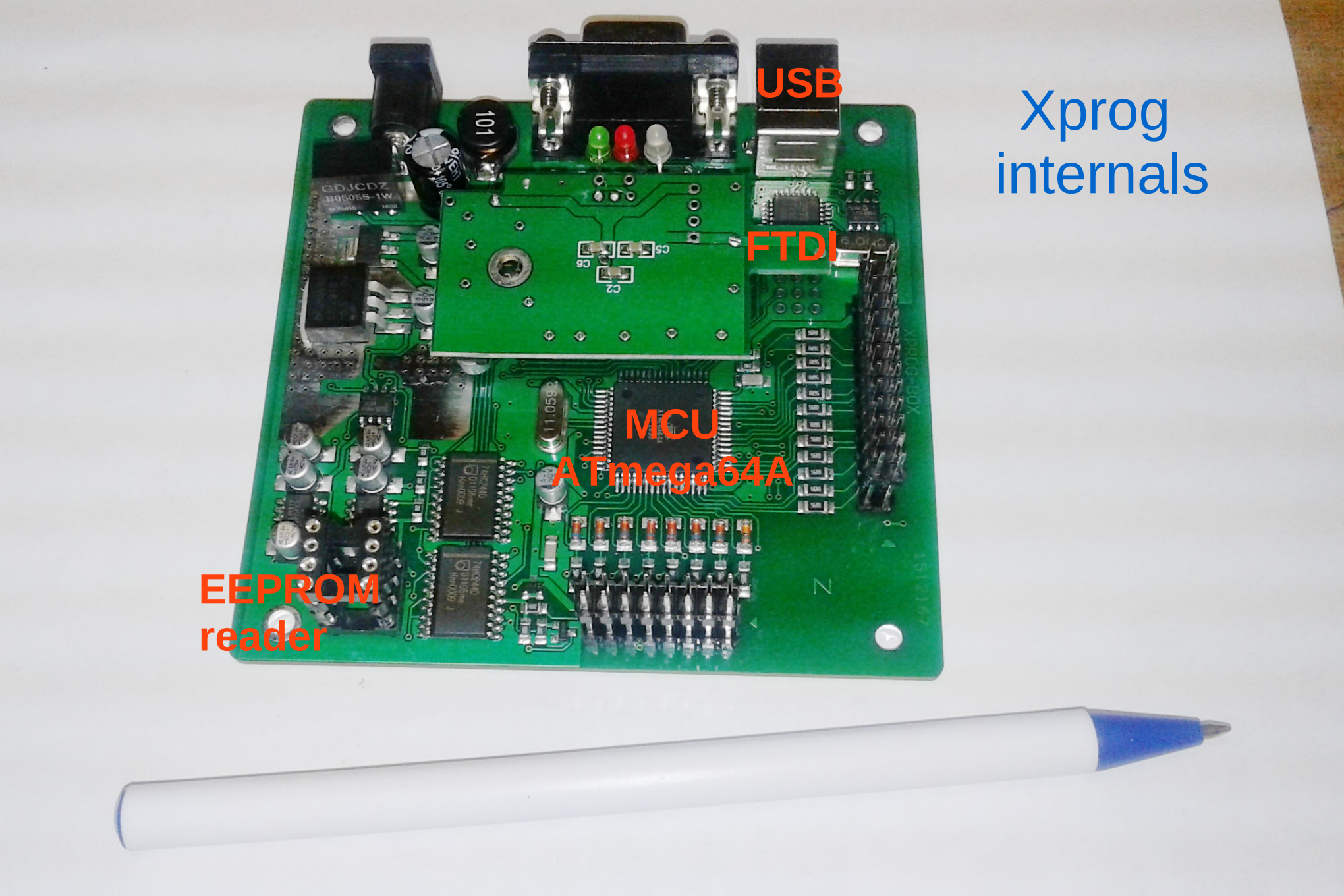
2. Close all of the computer anti-software. If not, X-prog software may be killed.

3. **Disconnect the internet**. If not, the ***internet may damage the hardware***.

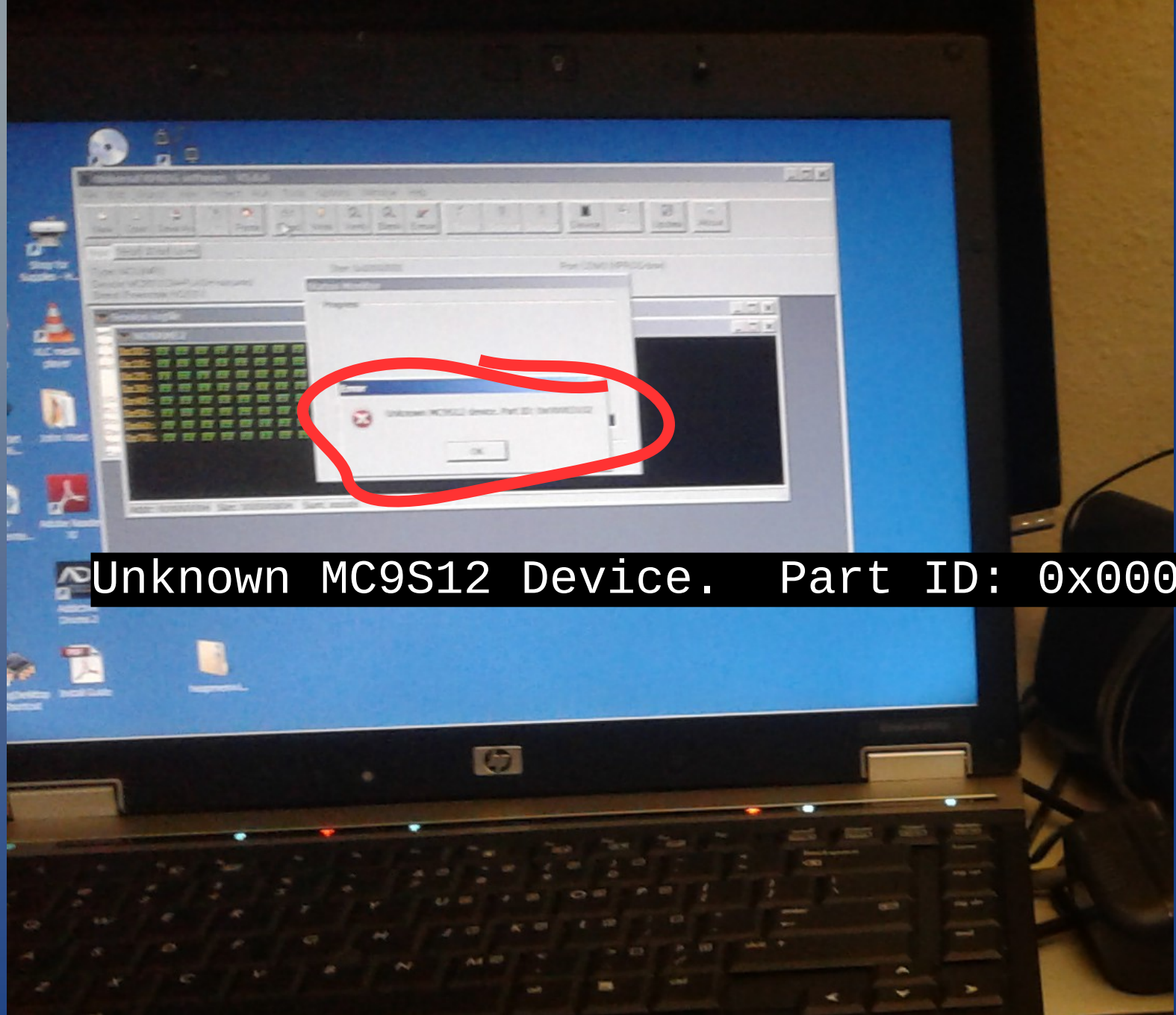
4. Uninstall all the old xprog-m, or xprog box software, make sure that your PC only install our software for our xprog-box 5.70 , our xprog-box hardware cannot work with other lower or higher software. If not, the hardware will be damaged, and will lose its warranty...

5. **Never try to UPDATE**, the hardware will be damaged if you want to try to upgrade it online, without any warranty.

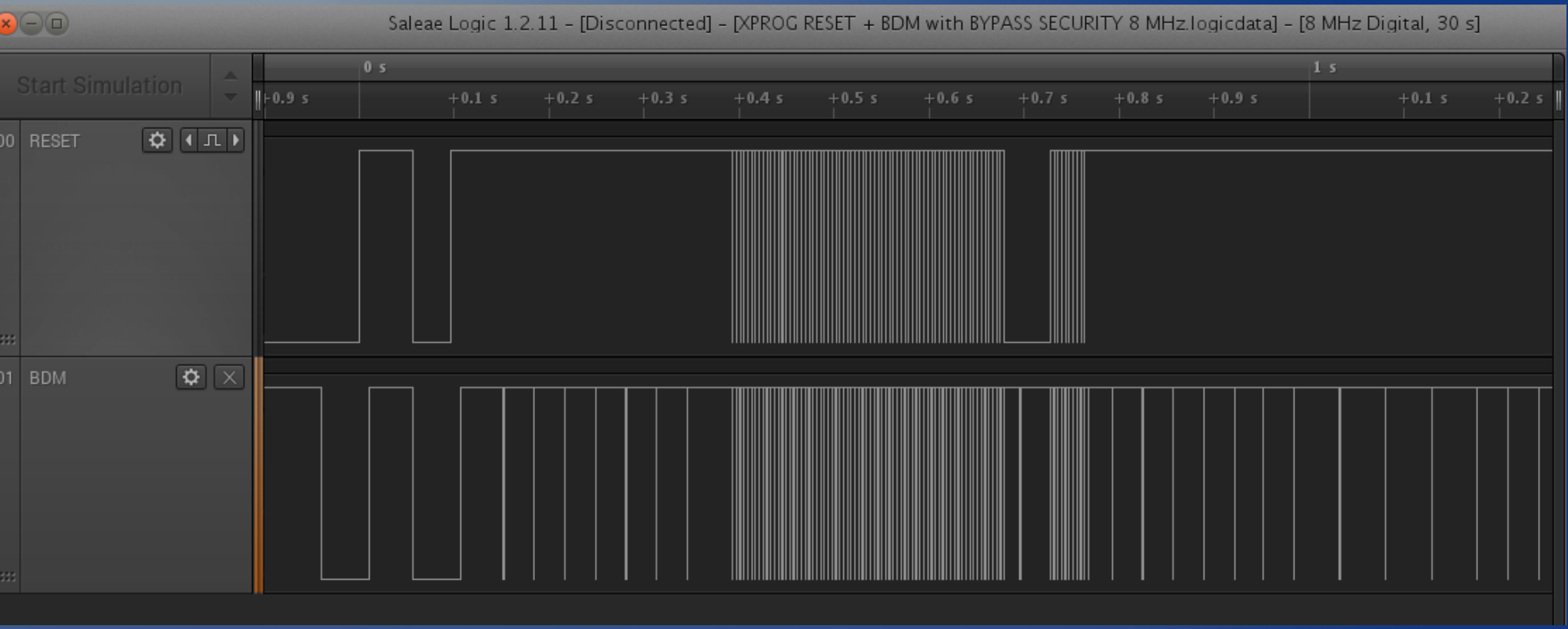
6. We cannot refund, cannot exchange, can not repair if you do not listen to those advices. It means that you accept those conditions if you have ordered our xprog-box.

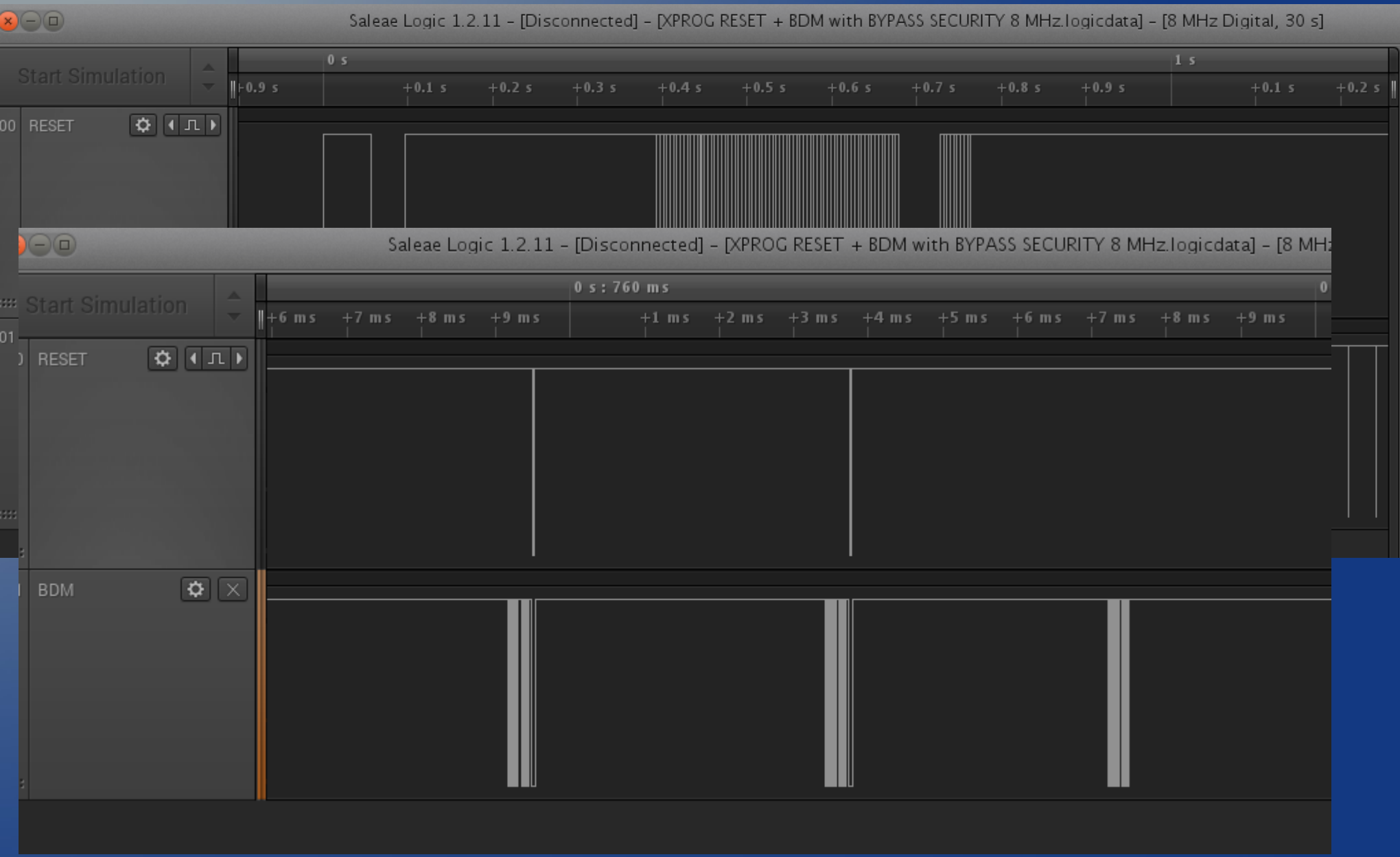


Xprog internals

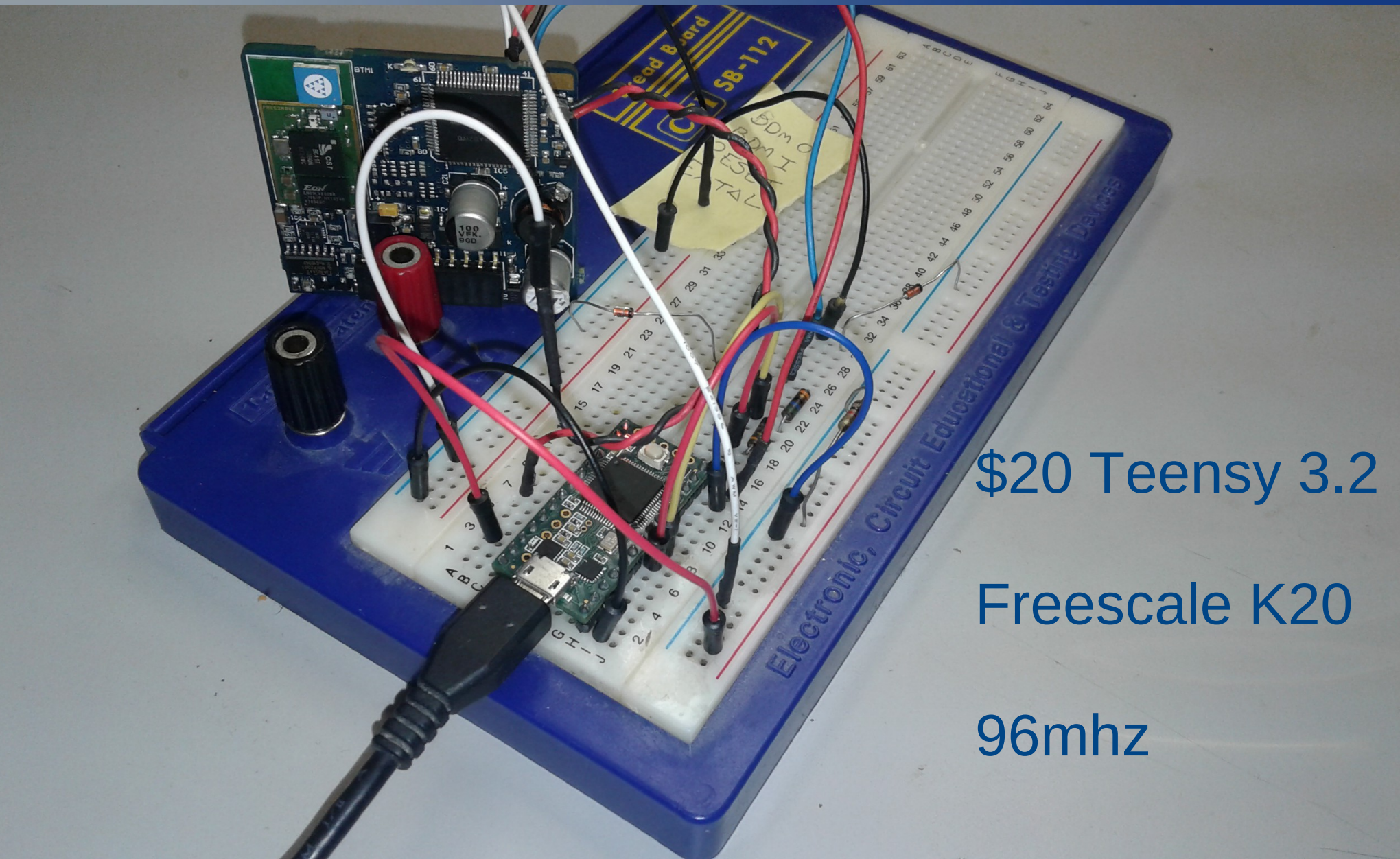

Unknown MC9S12 Device. Part ID: 0x00003102

GLITCHING THE HCS12 RESET-PHASE STATE MACHINE BY PULLING RESET HIGH AT CRITICAL POINTS

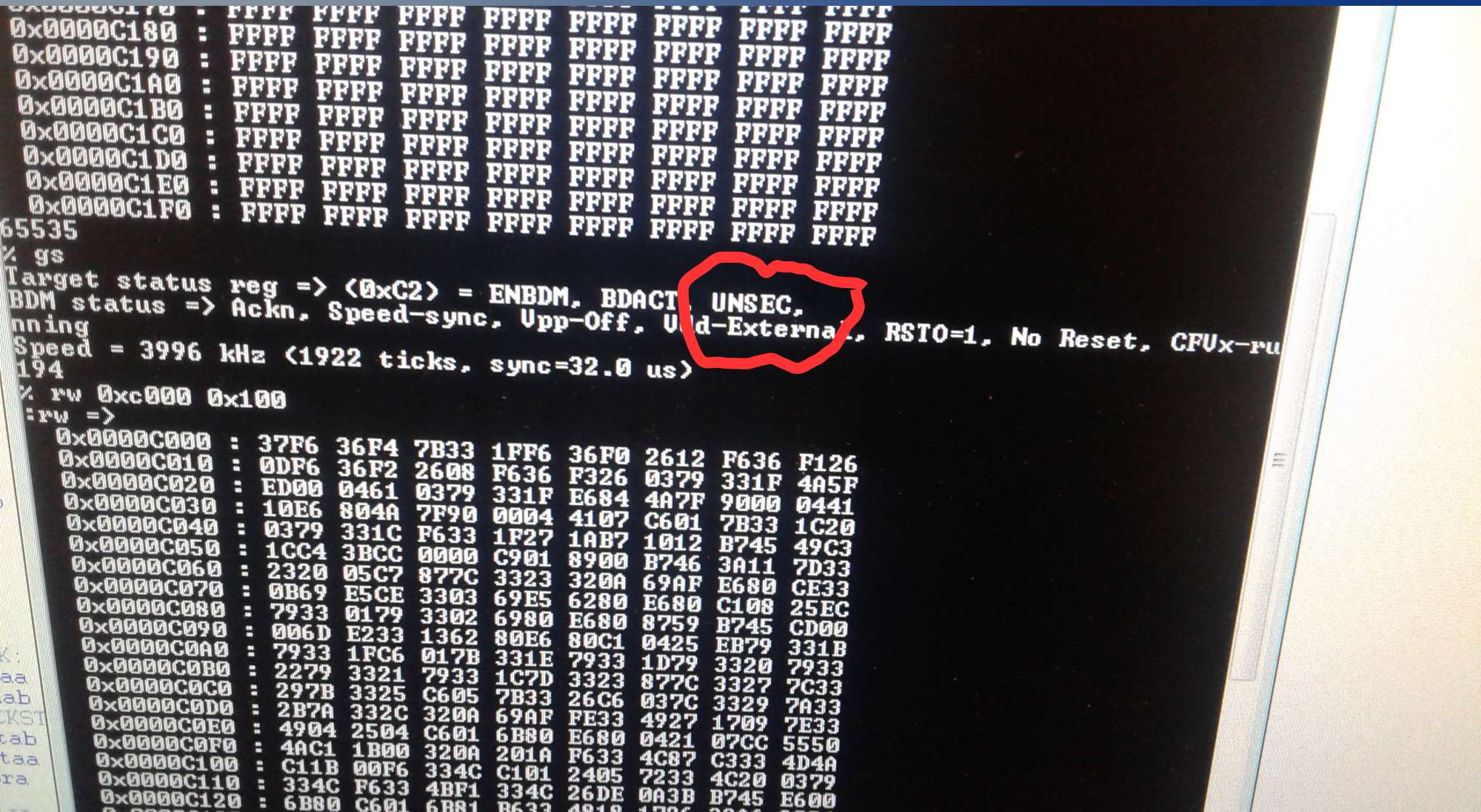



Saleae Logic 1.2.11 - [Disconnected] - [XPROG RESET + BDM with BYPASS SECURITY 8 MHz.logicdata] - [8 MHz Digital, 30 s]
Start Simulation
0 s
1 s
+0.9 s
+0.1 s
+0.2 s
+0.3 s
+0.4 s
+0.5 s
+0.6 s
+0.7 s
+0.8 s
+0.9 s
+0.1 s
+0.2 s
RESET
Saleae Logic 1.2.11 - [Disconnected] - [XPROG RESET + BDM with BYPASS SECURITY 8 MHz.logicdata] - [8 MHz
Start Simulation
0 s : 760 ms
+6 ms
+7 ms
+8 ms
+9 ms
+1 ms
+2 ms
+3 ms
+4 ms
+5 ms
+6 ms
+7 ms
+8 ms
+9 ms
RESET
BDM

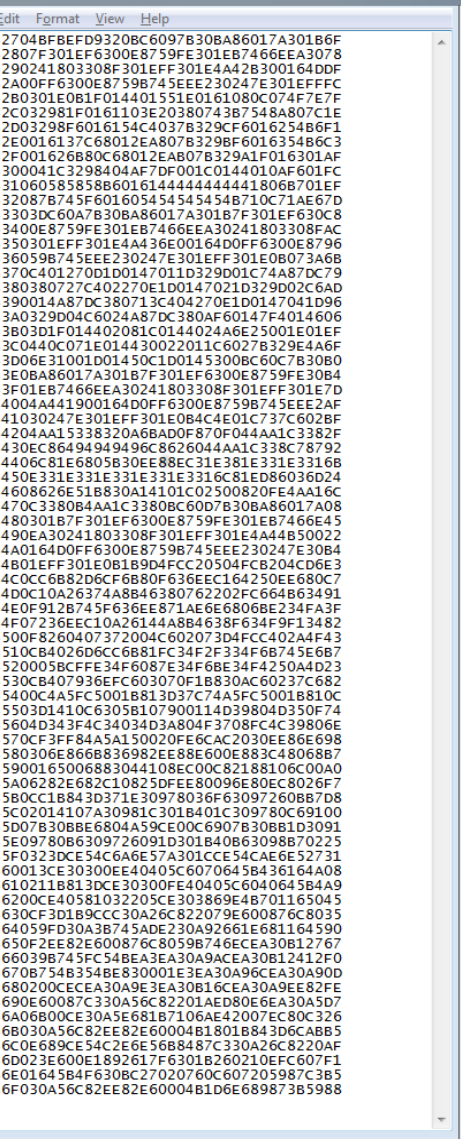
```
0.751430375      '0xe4ff01'      READ_BD_BYTE    FF01 = FFFF     RSTGAP: 39143
0.751610625                                      SEC!=0
0.751762375 - 0.75178175         RESET   194
0.75513675       '0xe4ff01'      READ_BD_BYTE    FF01 = FFFF     RSTGAP: 33550
0.755317                                         SEC!=0
0.755468625 - 0.755488375        RESET   197
0.759162         '0xe4ff01'      READ_BD_BYTE    FF01 = FFFF     RSTGAP: 36736
0.7934225                                        SEC!=0
0.759494125 - 0.759513625        RESET   195
0.7951375 - 0.759513875          RESET   1
0.763426         '0xe4ff01'      READ_BD_BYTE    FF01 = FFFF     RSTGAP: 39121
0.76360625                                       SEC!=0
0.76375775 - 0.763777375         RESET   196
0.767221875      '0xe4ff01'      READ_BD_BYTE    FF01 = 00CA     RSTGAP: 34445
0.767402125                                      SEC!=0
0.7922245        '0xe4ff01'      READ_BD_BYTE    FF01 = 00CA     RSTGAP: 284471
0.79240475                                       SEC!=0
0.824448625      '0xe0ff0f'      READ_BYTE       FF0F = FFFC
0.856687625      '0xe00101'      READ_BYTE       0101 = 00FC
0.8889145        '0xe0ff0d'      READ_BYTE       FF0D = FFFF
0.921142125      '0xe00104'      READ_BYTE       0104 = FF00
0.951355875      '0xe00010'      READ_BYTE       0010 = 0900
0.98358225       '0xc808001234'  WRITE_WORD      0800 = 1234
1.031914625      '0xc808025a7a'  WRITE_WORD      0802 = 5A7A
1.08026625       '0xe80800'      READ_WORD       0800 = 1234
1.128606875      '0xe80802'      READ_WORD       0802 = 5A7A
1.176947         '0xe4ff01'      READ_BD_BYTE    FF01 = 00CA     RSTGAP: 4131696
1.17712725                                       SEC!=0
1.209184875      '0xc4ff0100ca'  WRITE_BD_BYTE   FF01 = 00CA
1.24149775       '0xe00013'      READ_BYTE       0013 = 00CD
1.271623125      '0xe8001a'      READ_WORD       001A = 3102
```

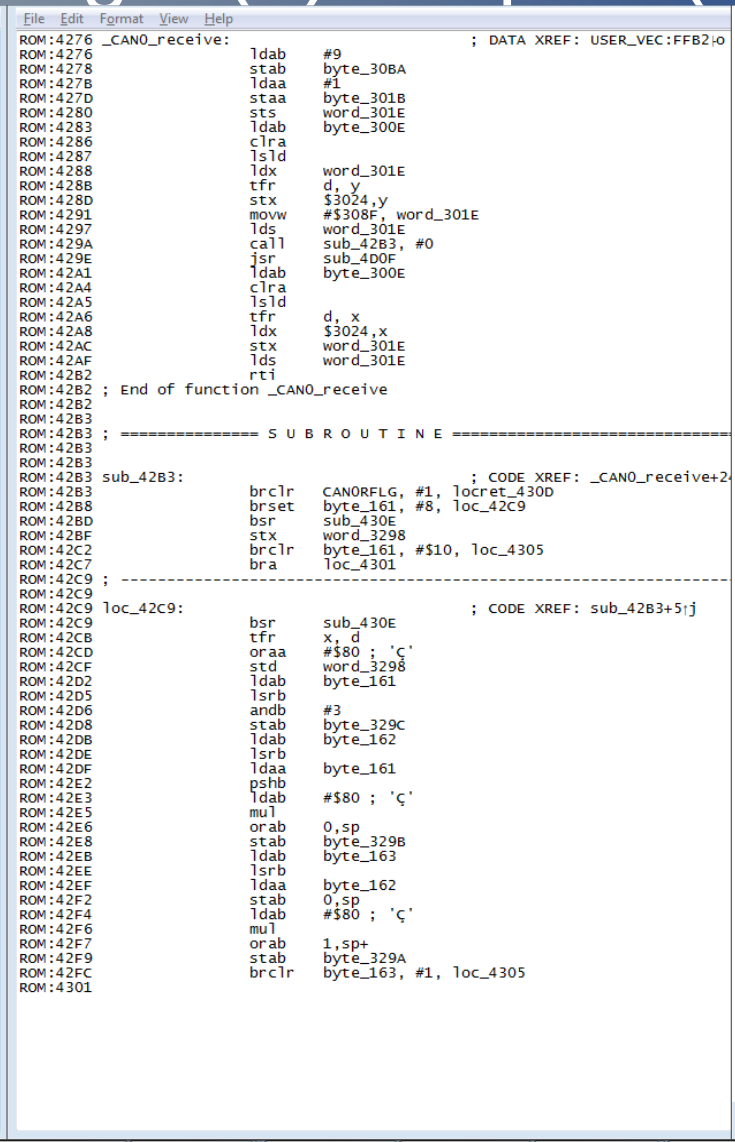

$20 Teensy 3.2
Freescale K20
96mhz

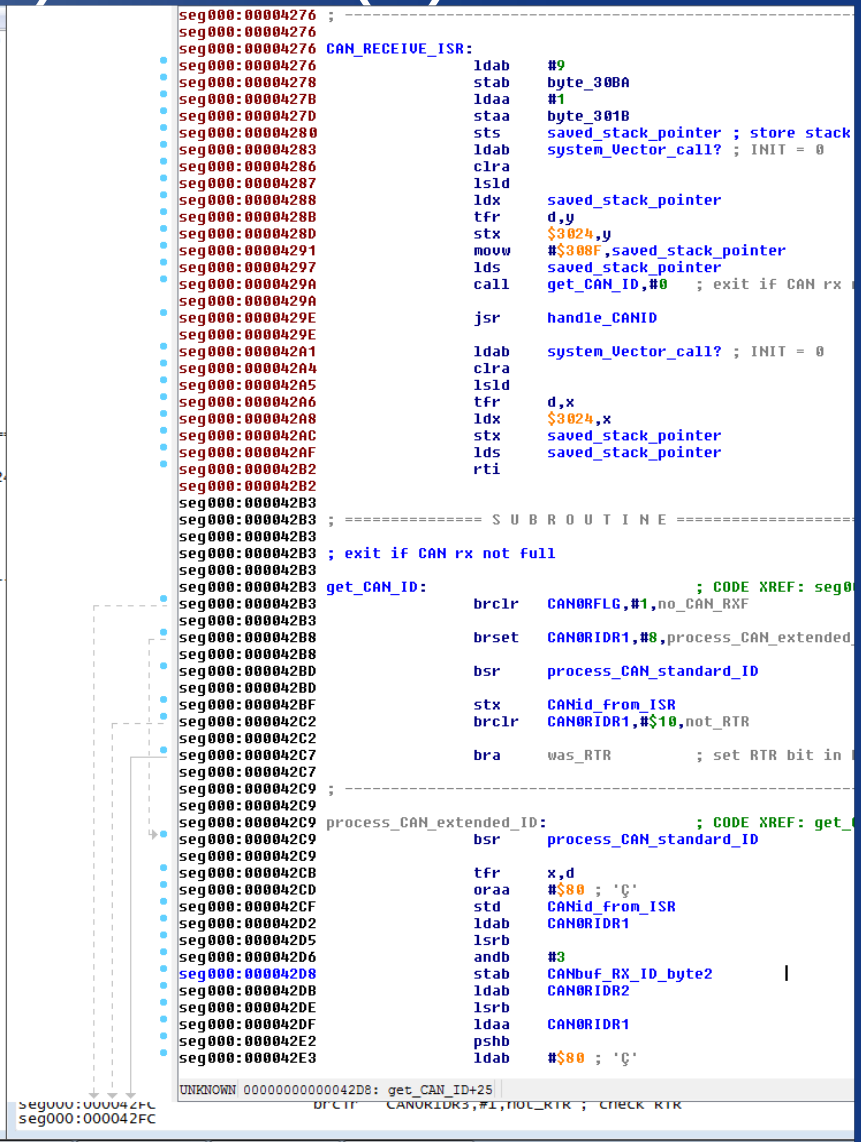
Reading 9s12 after bypass security



Reversing: (1)dump ... (2)diss ... (3)name







MCUs on PowerWheelchair

Power Module – Freescale MC56F83

Joystick Module – NXP LH75411-NOQ
32-BIT ARM7

IO module – Freescale MC908GZ

Bluetooth “iDevice” - Freescale 9S12C

Light Module (not OEM) -
SAF-XC164CS
ATMEL AT90Can128

```
; =============== S U B R O U T I N E =======================================

; exit if CAN rx not full

get_CAN_ID:                             ; CODE XREF: seg000:0000429A↑p
                brclr   CAN0RFLG,#1,no_CAN_RXF

                brset   CAN0RIDR1,#8,get_CAN_extended_ID ; Extended: ID20 ID19 ID18 SRR=1 IDE=1 ID17 ID16 ID15

                bsr     mung_CANMSB_into_X ; X = 00000 ID10 ID9 ID8 / ID7 ID5 ID6 ID4 ID3 ID2 ID1 ID0
                                        ; X = 00000 ID28 ID27 ID26 / ID25 ID24 ID23 ID22 ID21 ID20 ID19 ID18

                stx     CANRIDtop11     ; EXT RTR 0 0  0 ID28/10 ID27/9 ID26/8  ID25/7 ID24/6 ID23/5 ID22/4  ID21/3 ID20/2 ID19/1 ID18/0
                brclr   CAN0RIDR1,#$10,not_RTR ; Extended: ID20 ID19 ID18 SRR=1 IDE=1 ID17 ID16 ID15

                bra     was_RTR         ; set RTR bit in buffer

; ---------------------------------------------------------------------------

get_CAN_extended_ID:                    ; CODE XREF: get_CAN_ID+5↑j
                bsr     mung_CANMSB_into_X ; X = 00000 ID10 ID9 ID8 / ID7 ID5 ID6 ID4 ID3 ID2 ID1 ID0
                                        ; X = 00000 ID28 ID27 ID26 / ID25 ID24 ID23 ID22 ID21 ID20 ID19 ID18

                tfr     x,d
                oraa    #$80 ; 'Ç'      ; set EXTENDED_FRAME bit in D
                std     CANRIDtop11     ; EXT RTR 0 0  0 ID28/10 ID27/9 ID26/8  ID25/7 ID24/6 ID23/5 ID22/4  ID21/3 ID20/2 ID19/1 ID18/0
                ldab    CAN0RIDR1       ; Extended: ID20 ID19 ID18 SRR=1 IDE=1 ID17 ID16 ID15
                lsrb
                andb    #3              ; B = ID17 ID16
                stab    CANbuf_RX_XID_17to16 ; ID17 ID16
                ldab    CAN0RIDR2       ; ID14 ID13 ID12 ID11 ID10 ID9 ID8 ID7
                lsrb                    ; B = 0 ID14 ID13 ID12 ID11 ID10 ID9 ID8
                ldaa    CAN0RIDR1       ; Extended: ID20 ID19 ID18 SRR=1 IDE=1 ID17 ID16 ID15
                                        ; Standard: ID2 ID1 ID0 RTR IDE=0 0 0 0
                pshb
                ldab    #$80 ; 'Ç'      ; << 7
                mul                     ; B = ID15 0 0 0 0 0 0 0
                orab    0,sp
                stab    CANbuf_RX_XID_15to8 ; ID15 ID14 ID13 ID12 ID11 ID10 ID9 ID8
                ldab    CAN0RIDR3       ; ID6 ID5 ID4 ID3 ID2 ID1 ID0 RTR
                lsrb                    ; drop RTR bit
                ldaa    CAN0RIDR2       ; ID14 ID13 ID12 ID11 ID10 ID9 ID8 ID7
                stab    0,sp
                ldab    #$80 ; 'Ç'
                mul                     ; B = ID7 0 0 0  0 0 0 0
                orab    1,sp+
                stab    CANbuf_RX_XID_7to0 ; ID7 ID6 ID5 ID4 ID3 ID2 ID1 ID0
                brclr   CAN0RIDR3,#1,not_RTR ; check RTR
```

dicesoft.net

gofundme.com/stephenandlani

github.com/redragonx

sp3ct3r@protonmail.com