Don’t Whisper My Chips

Colin O’Flynn – Dalhousie University – Halifax, NS. Canada.

WTF ARE WE DOING?

Objective: Learn about all sorts of 'physical' layer attacks.

Critical Difference: Everything I'm showing you is *open source* and freely available. Most of the hardware is commercially available (it's difficult for hardware to be free), but you can DIY it too if you wish, or use existing tools (e.g. oscilloscope).

www.NewAE.com

Open Source Tools Posted to:

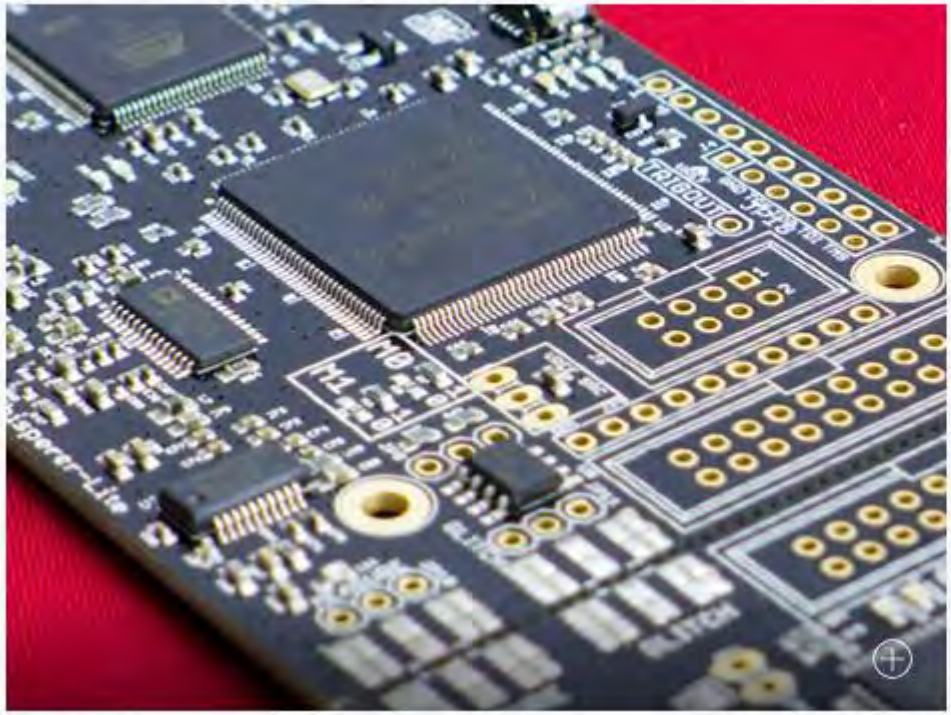
www.ChipWhisperer.com


SPRINGFIELD
SAUSAGE FACTORY
KIDS, COME SEE HOW
IT'S MADE!

Who Am I?

KICKSTARTER!

ChipWhisperer-Lite: A New Era of Hardware Security Research



Embedded security - is it an oxymoron? Learn the truth through a series of hands-on labs targeting computer and electrical engineers.

⊕ **Add link**

Created by

Colin O'Flynn

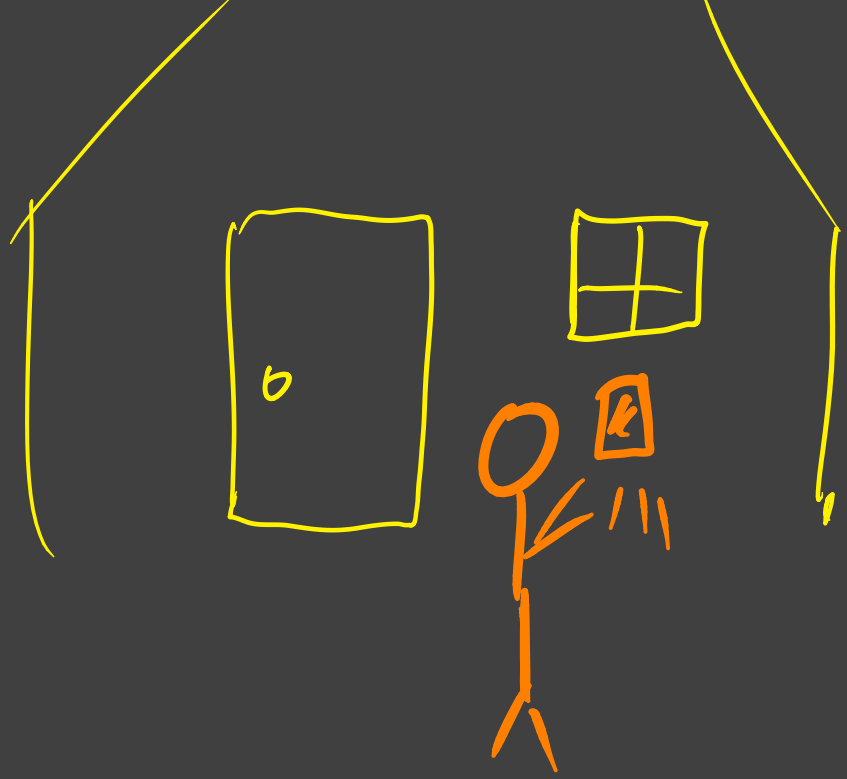
331 backers pledged $88,535 to help bring this project to life.

EMBEDDED?

Embedded System:

Not an Embedded System:

THE SIDE CHANNEL

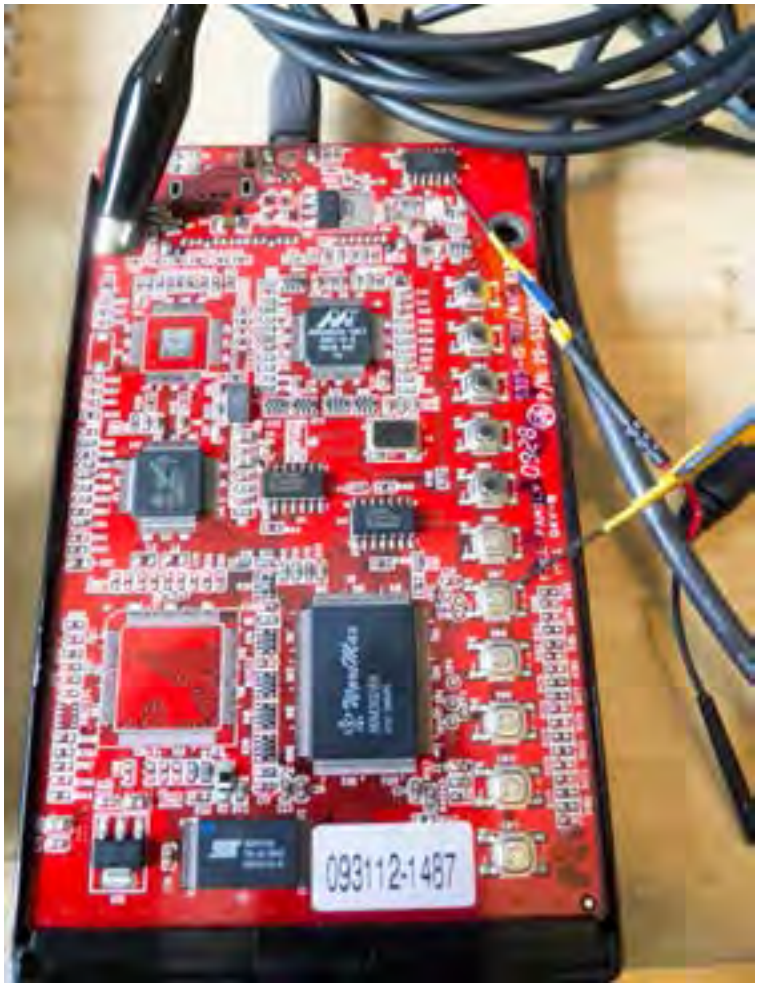


Timing Attacks

```
unsigned char correctpin[6] = {1,2,3,4,5,6};
unsigned char enteredpin[6];

read_pin_from_buttons(enteredpin);

for (i = 0; i < 6; i++){

        if (correctpin[i] != enteredpin[i]){
                return;

        }
}
```

(REDACTED)

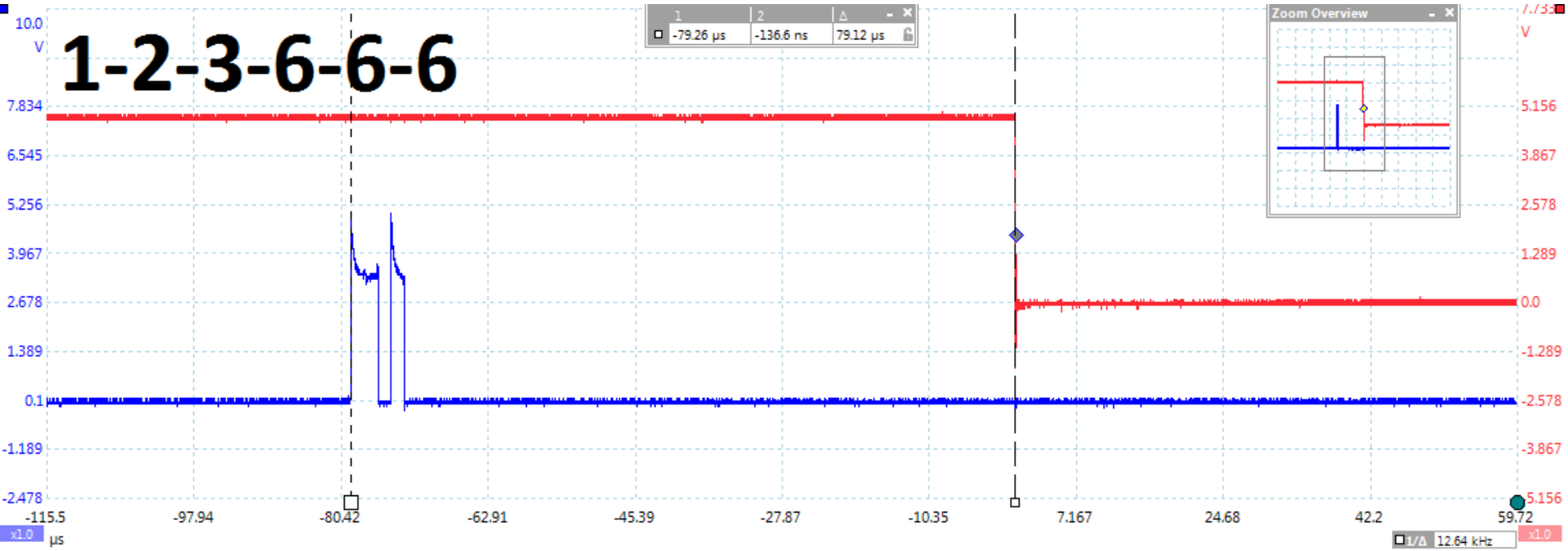
START WITH TIME

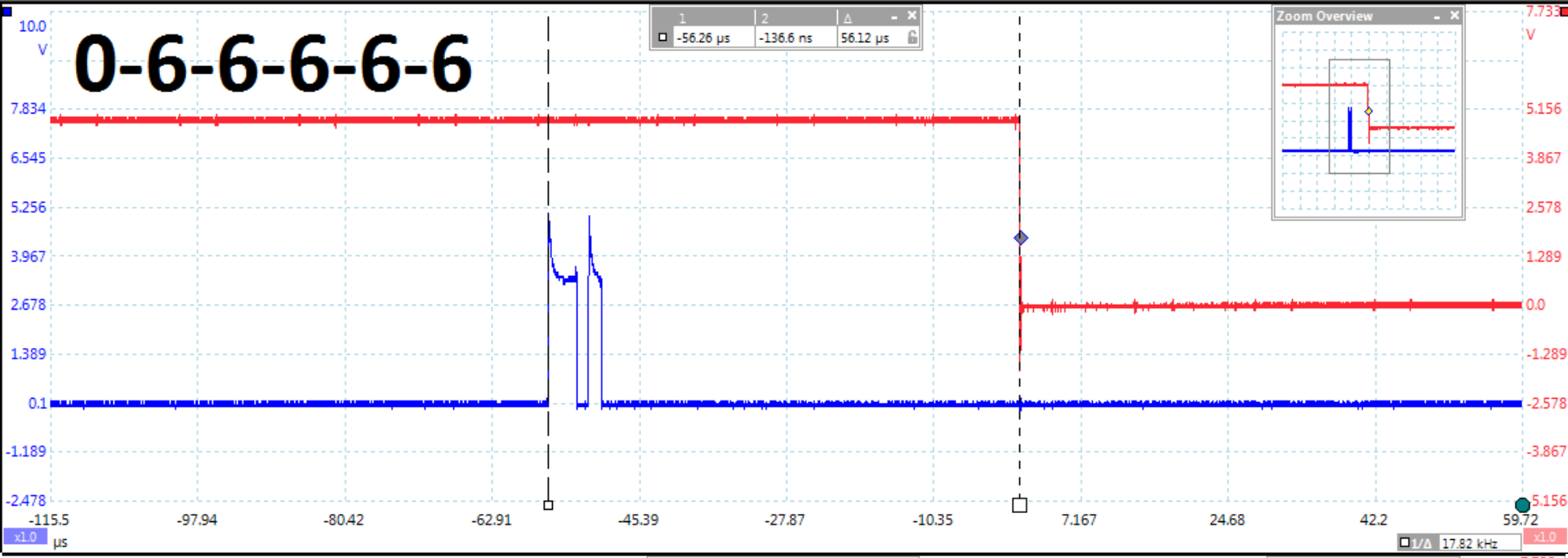


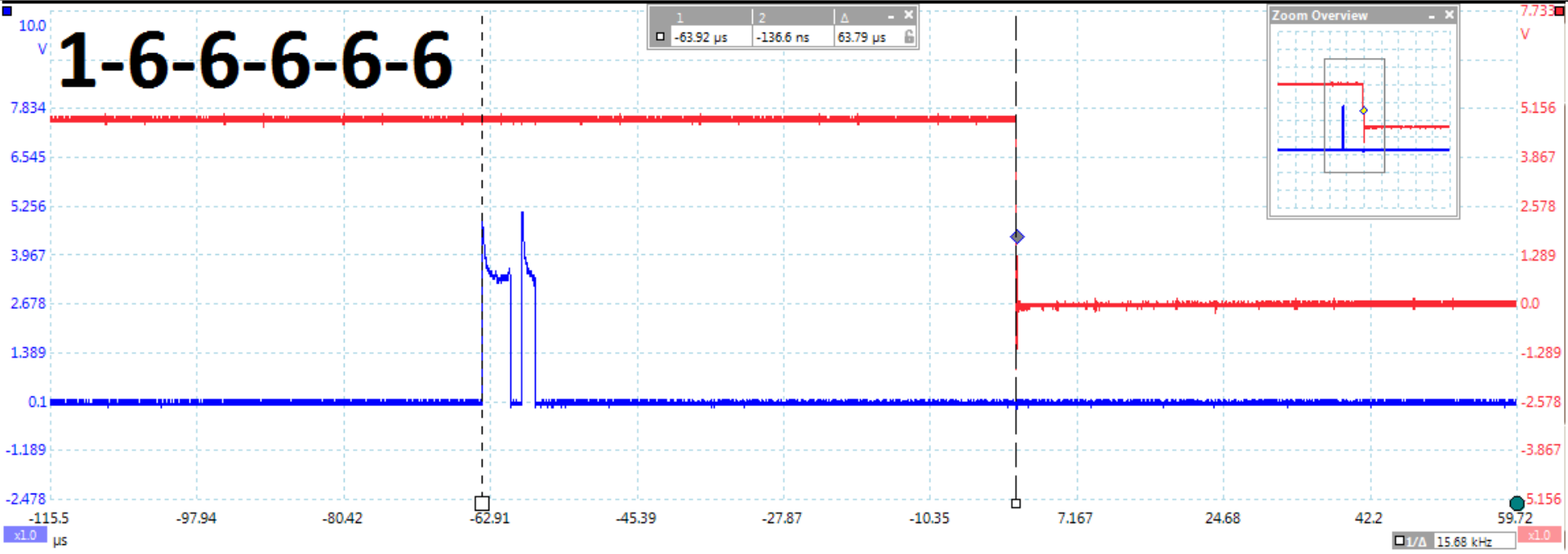
START WITH TIME

- Important: Must discover when microcontroller detects button press, not when user presses button
  - Need to do some slight reverse-engineering to discover if using multiplexing, how often polling of buttons, etc

Time to Measure: Time




0-6-6-6-6-6
1
2
Δ
-56.26 µs
-136.6 ns
56.12 µs
Zoom Overview
10.0
V
7.834
6.545
5.256
3.967
2.678
1.389
0.1
-1.189
-2.478
7.733
V
5.156
3.867
2.578
1.289
0.0
-1.289
-2.578
-3.867
5.156
-115.5
-97.94
-80.42
-62.91
-45.39
-27.87
-10.35
7.167
24.68
42.2
59.72
µs
1/Δ
17.82 kHz


1-6-6-6-6-6
1
2
Δ
-63.92 µs
-136.6 ns
63.79 µs
Zoom Overview
10.0
V
7.834
6.545
5.256
3.967
2.678
1.389
0.1
-1.189
-2.478
7.733
V
5.156
3.867
2.578
1.289
0.0
-1.289
-2.578
-3.867
5.156
-115.5
-97.94
-80.42
-62.91
-45.39
-27.87
-10.35
7.167
24.68
42.2
59.72
µs
1/Δ
15.68 kHz

Preventing Attacks

- Don't compare input directly to saved password
  - Ideally: use hashes, also prevents attacker from reading out memory
- Don't give any response when 'wrong'
  - No timing information 'easily' available
  - Depends how you define 'easily' though….

Simple Power Analysis

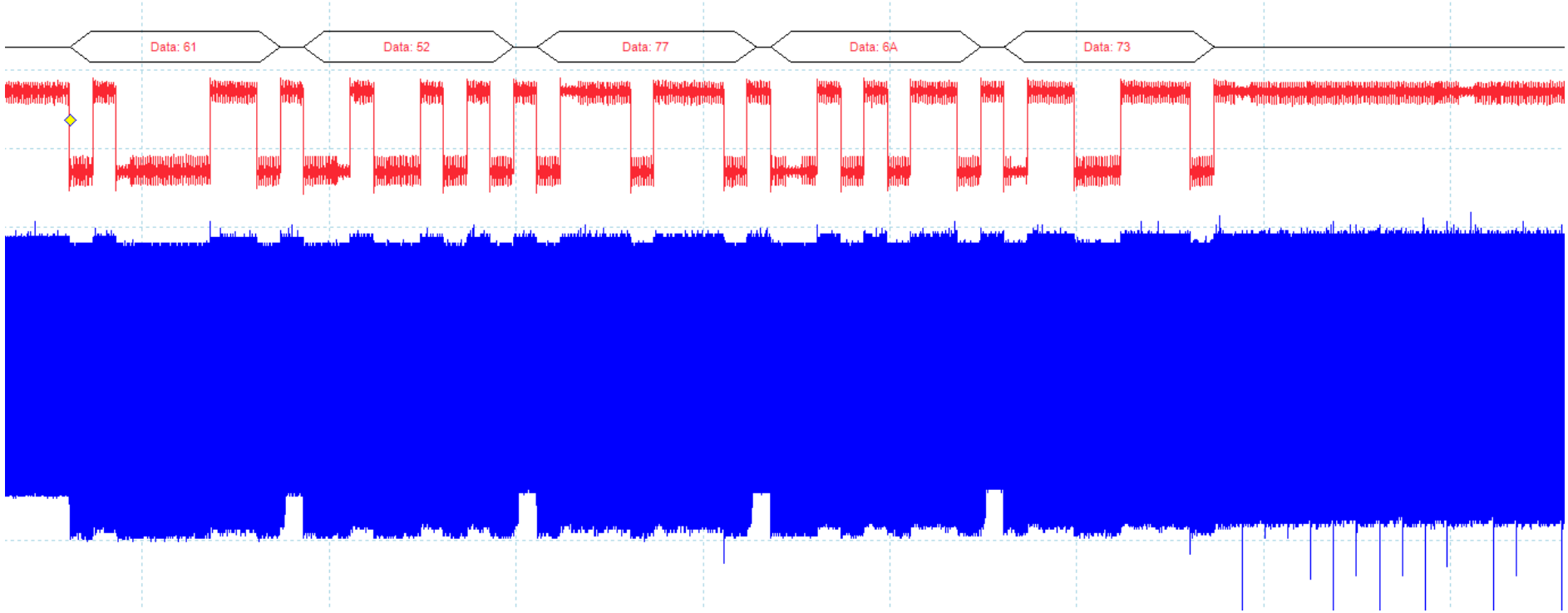
- TinySafeBoot (TSB) is Awesome Bootloader for AVRs
  - See http://jtxp.org/tech/tinysafeboot_en.htm

- **DOES NOT** claim any sort of cryptographic protection
  - BUT – What if someone uses it *expecting* it to be 'secure'?

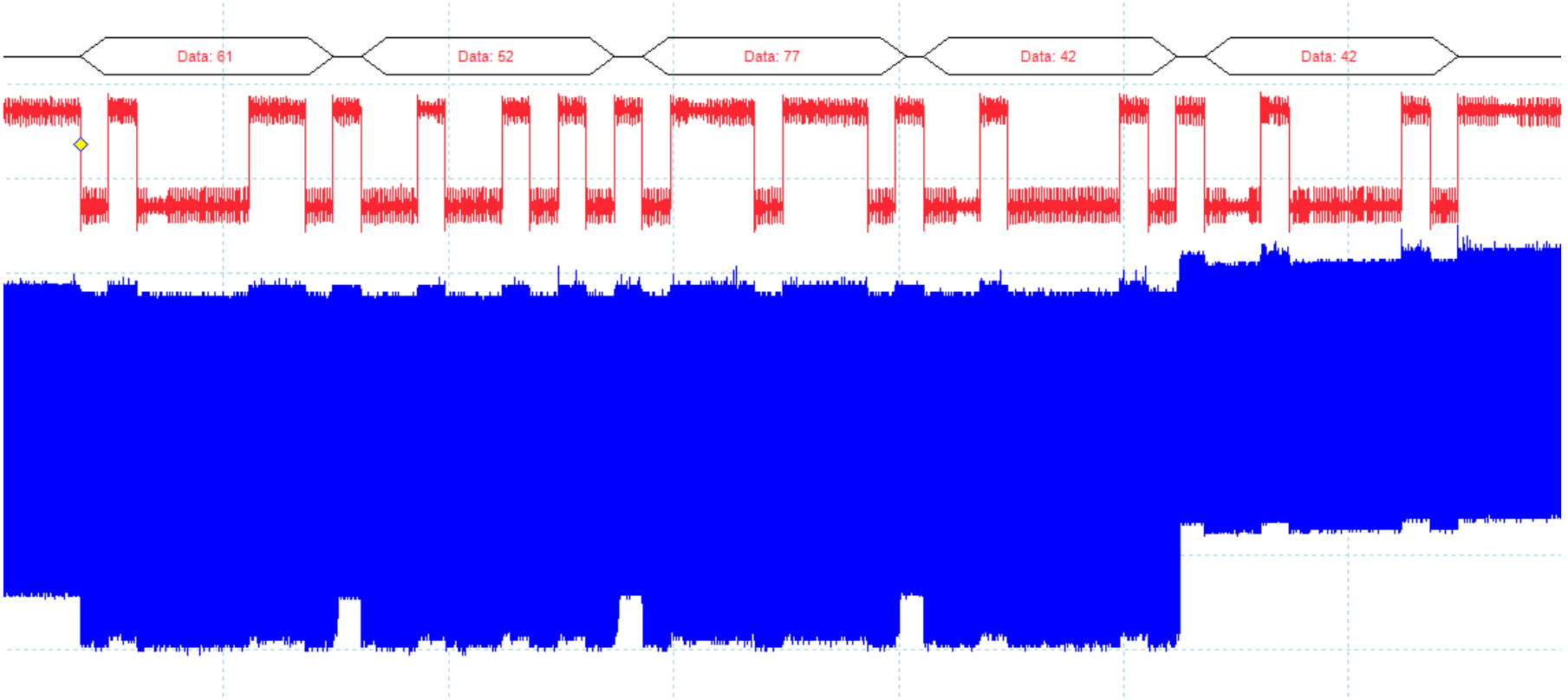
Simple Power Analysis

```
CheckPW:

chpw1:
        lpm tmp3, z+                    ; load character from Flash
        cpi tmp3, 255                   ; byte value (255) indicates
        breq chpwx                      ; end of password -> okay
        rcall Receivebyte               ; else receive next character
chpw2:
        cp tmp3, tmp1                   ; compare with password
        breq chpw1                      ; if equal check next character
        cpi tmp1, 0                     ; or was it 0 (emergency erase)
chpwl:  brne chpwl                      ; if not, loop infinitely
        rcall RequestConfirmation       ; if yes, request confirm
        brts chpa                       ; not confirmed, leave
        rcall RequestConfirmation       ; request 2nd confirm
        brts chpa                       ; can't be mistake now
        rcall EmergencyErase            ; go, emergency erase!


        rjmp Mainloop

chpa:
        rjmp APPJUMP                    ; start application
chpwx:
;        rcall SendDeviceInfo
```

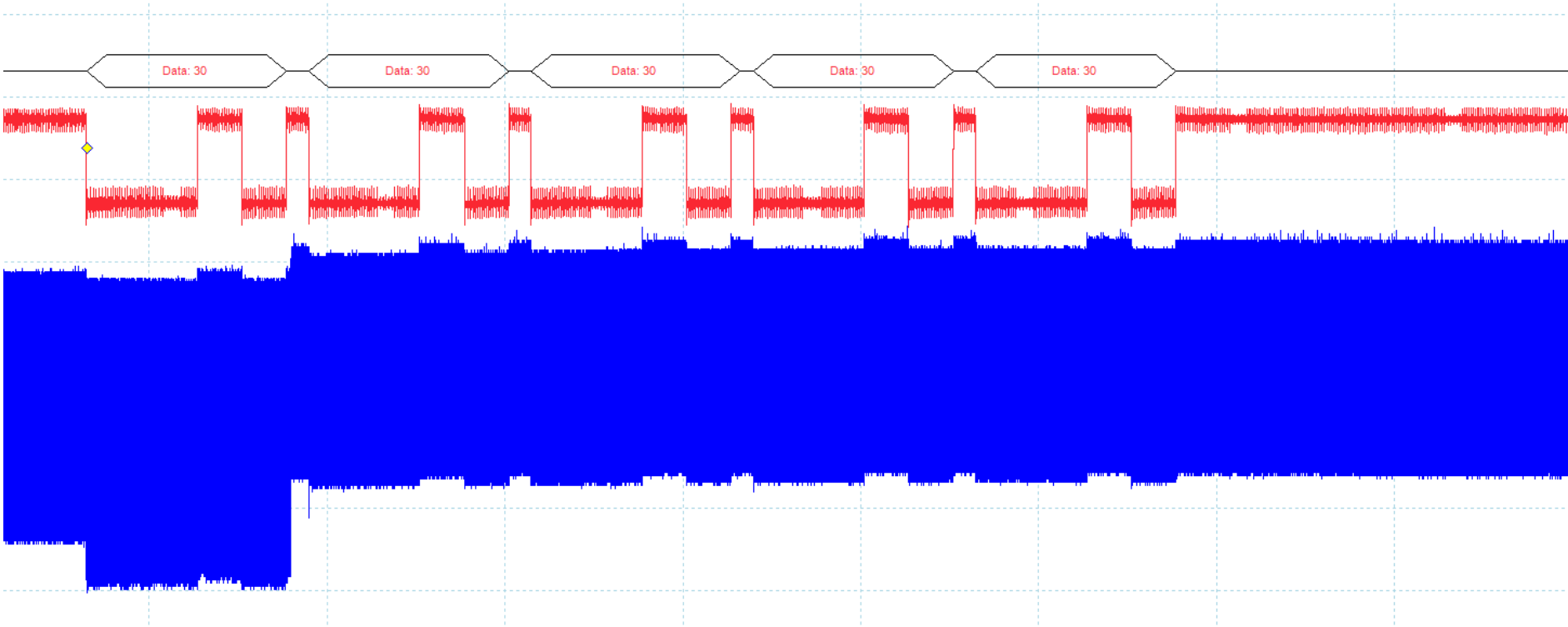
Simple Power Analysis



Simple Power Analysis



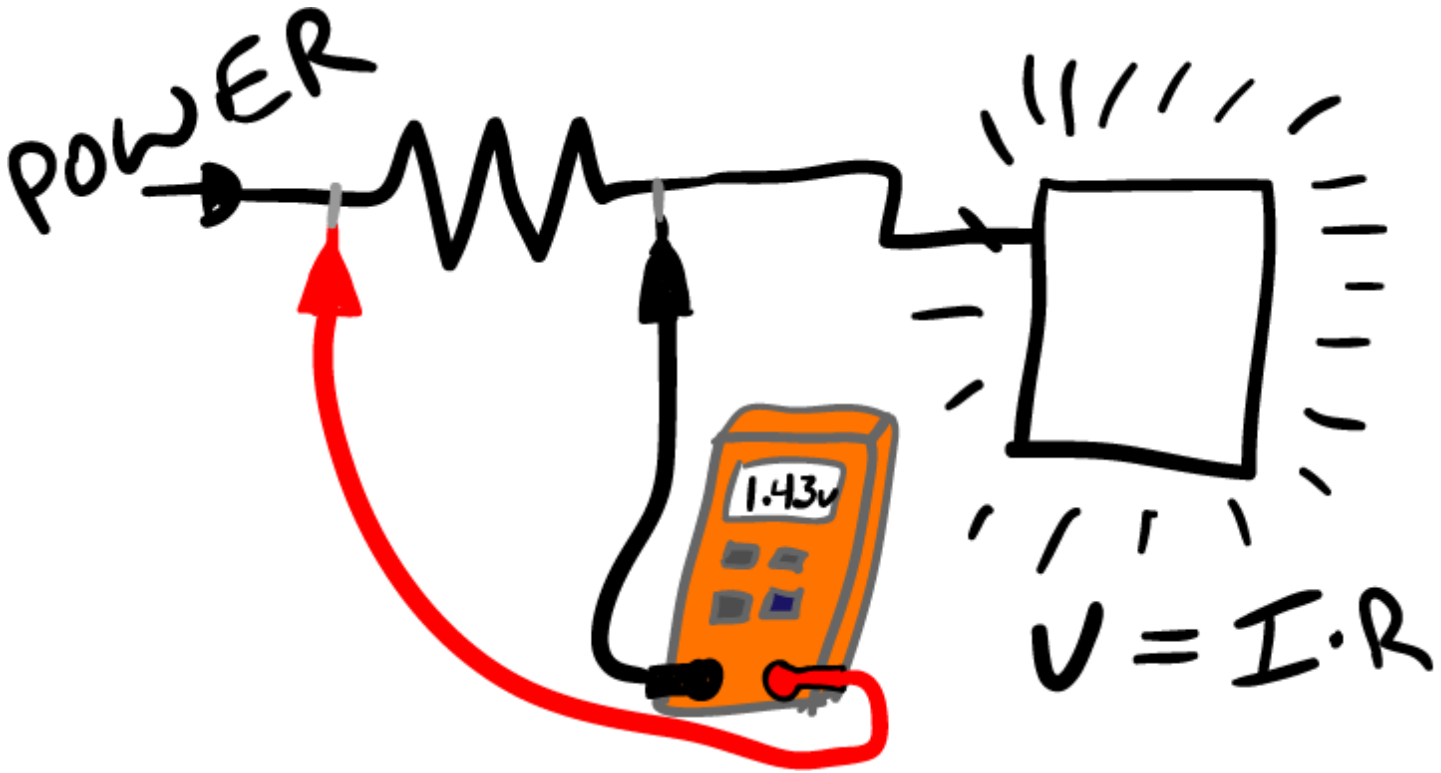
Simple Power Analysis



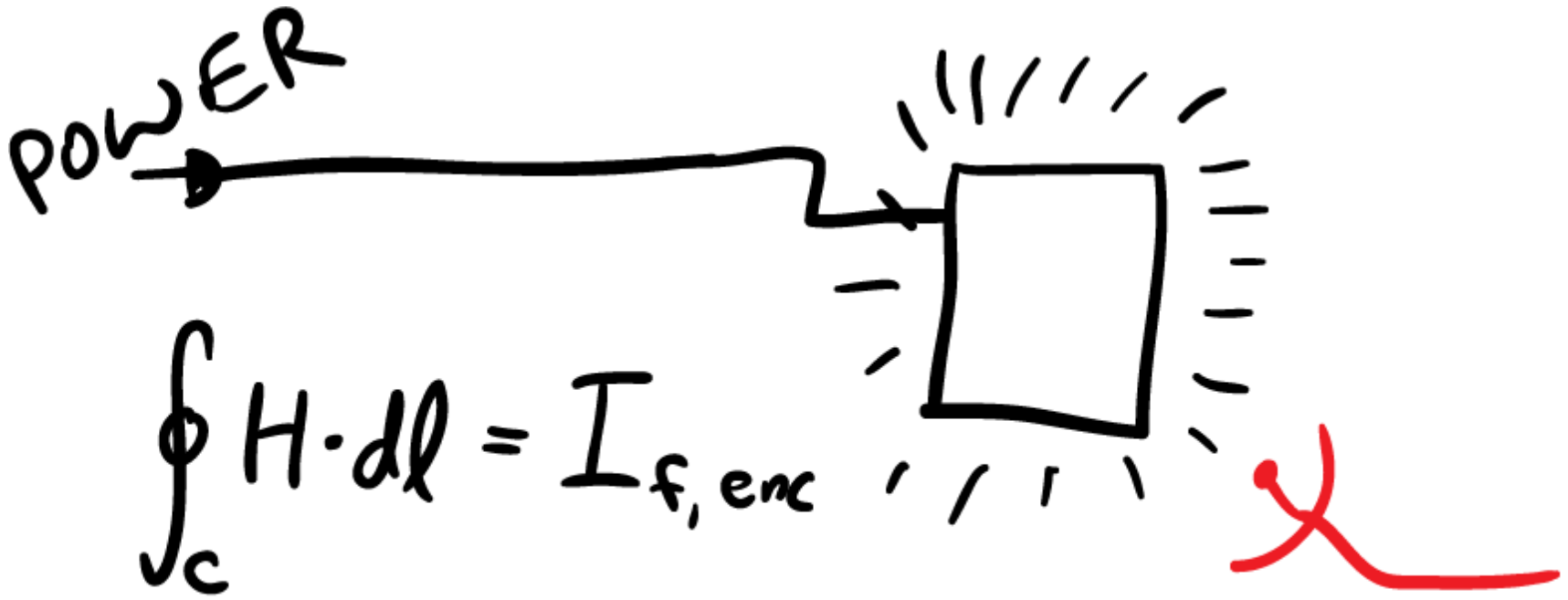
Simple Power Analysis

- 256 x N guesses required, can reduce this if password comes from ASCII-printable characters
  - Even if N=256 (very long password!), and can attempt only 1/second, still breakable in <24 hours

MEASURING POWER

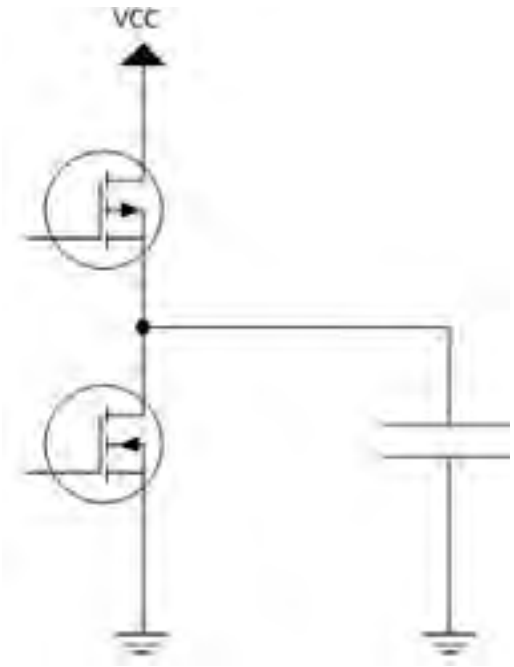


Simple Power Analysis

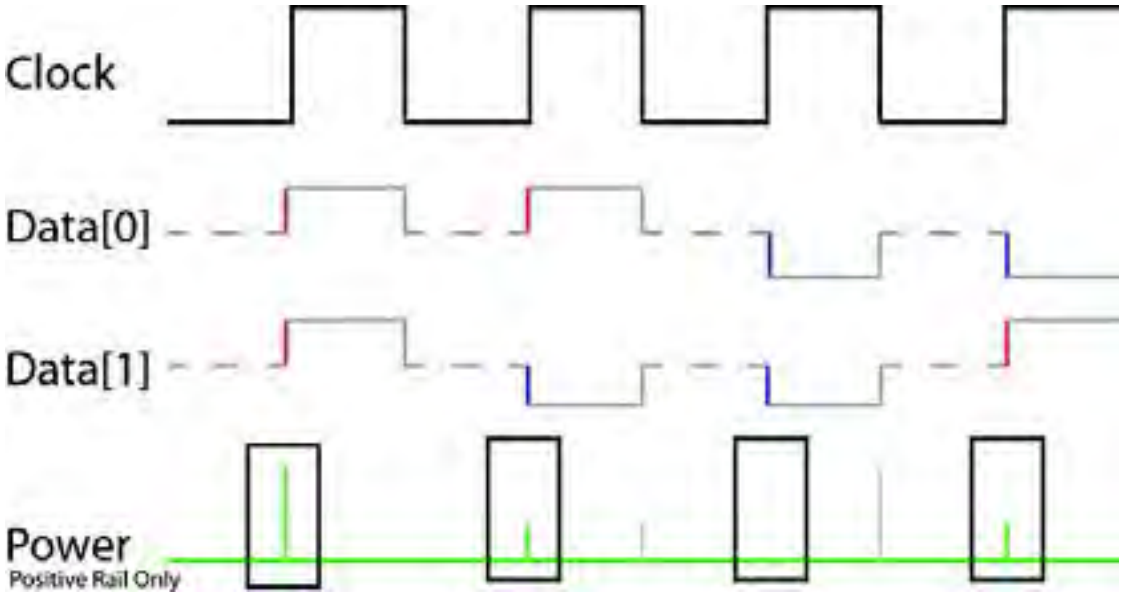


EM PROBE

Differential Power Ana.



(a)



(b)

DPA - XOR

Assume user is 'encrypting' a 1-byte piece of data by XORing with a 1-byte secret key (EF), and we cannot observe output of XOR. This becomes:

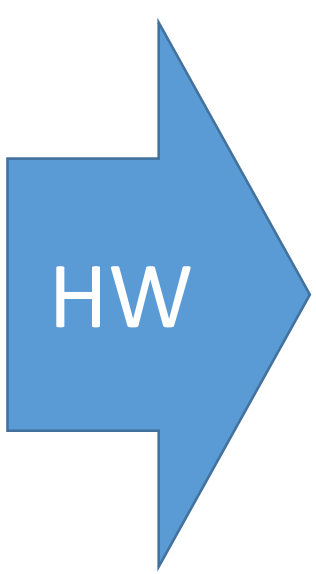
$88 \oplus EF = 67$

$56 \oplus EF = B9$

$32 \oplus EF = DD$

$A6 \oplus EF = 49$

$35 \oplus EF = DA$



5

5

6

3

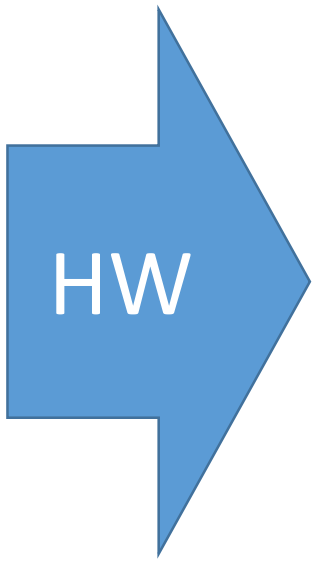
5

observations

DPA - XOR

Of course our **ACTUAL** observations are…

$88 \oplus XX = ?$
$56 \oplus XX = ?$
$32 \oplus XX = ?$
$A6 \oplus XX = ?$
$35 \oplus XX = ?$



5
5
6
3
5

observations

DPA - XOR

Guess each possibility for key, check what gets actual HW we observed

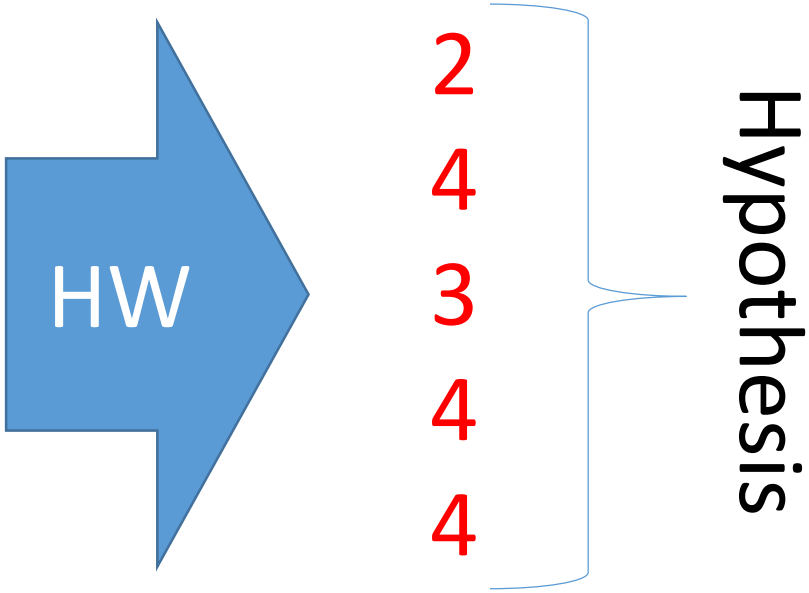
$88 \oplus 00 = 88$

$56 \oplus 00 = 56$

$32 \oplus 00 = 32$

$A6 \oplus 00 = A6$

$35 \oplus 00 = 35$



DPA - XOR

Guess each possibility for key, check what gets actual HW we observed

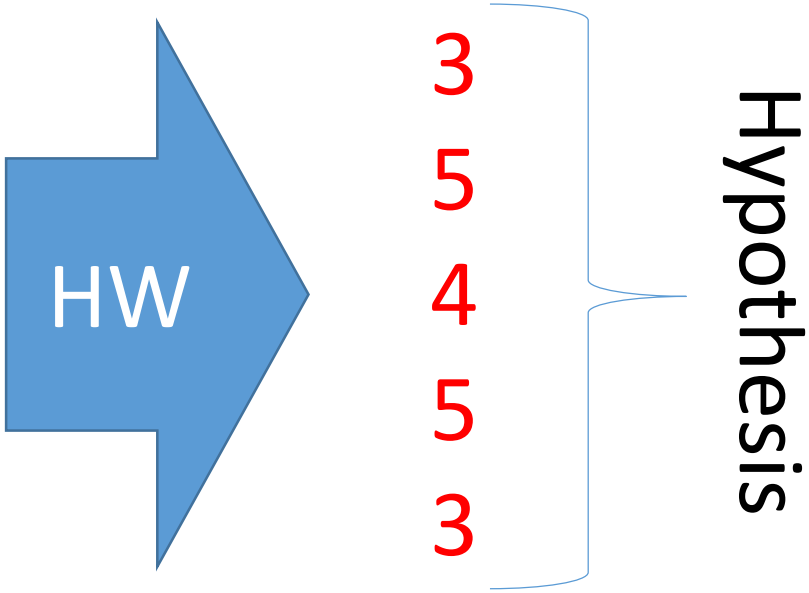
$88 \oplus 01 = 89$

$56 \oplus 01 = 57$

$32 \oplus 01 = 33$

$A6 \oplus 01 = A7$

$35 \oplus 01 = 34$



DPA - XOR

Assume user is 'encrypting' a 1-byte piece of data by XORing with a 1-byte secret key (EF), and we cannot observe output of XOR. Observed Result?

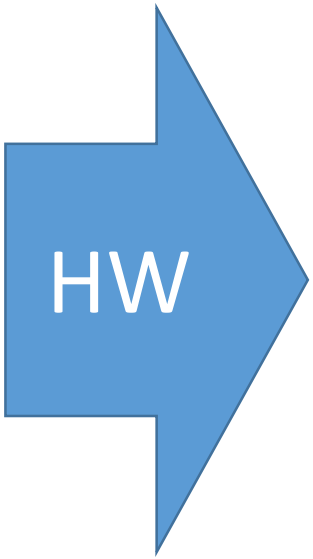
$88 \oplus EF = 67$ → 5

$56 \oplus EF = B9$ → 5

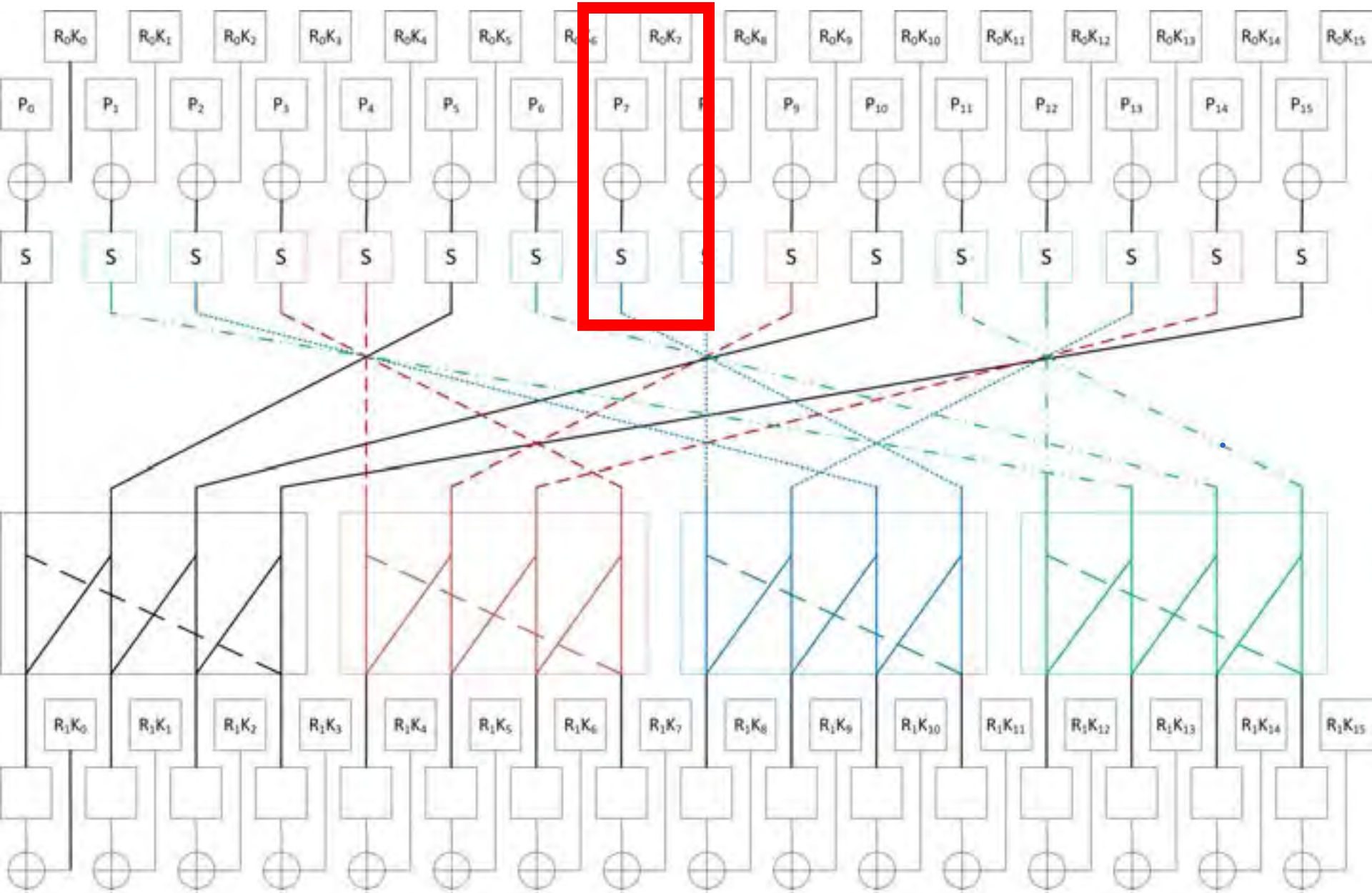
$32 \oplus EF = DD$ → 6

$A6 \oplus EF = 49$ → 3

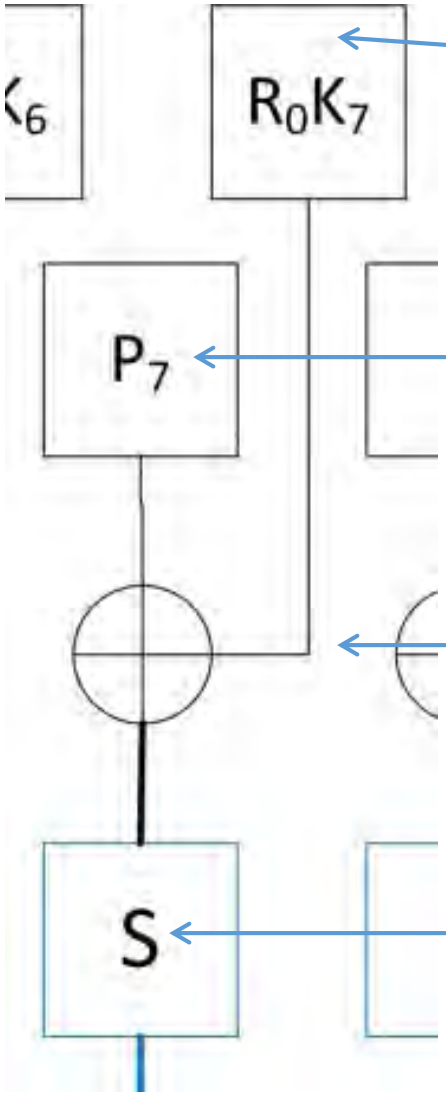
$35 \oplus EF = DA$ → 5



AES-128


R0K0
R0K1
R0K2
R0K3
R0K4
R0K5
R0K7
R0K8
R0K9
R0K10
R0K11
R0K12
R0K13
R0K14
R0K15
P0
P1
P2
P3
P4
P5
P6
P7
P9
P10
P11
P12
P13
P14
P15
S
R1K0
R1K1
R1K2
R1K3
R1K4
R1K5
R1K6
R1K7
R1K8
R1K9
R1K10
R1K11
R1K12
R1K13
R1K14
R1K15

AES-128



1-Byte of Key (Subkey)

1-Byte of Input (Plaintext)

Bitwise XOR

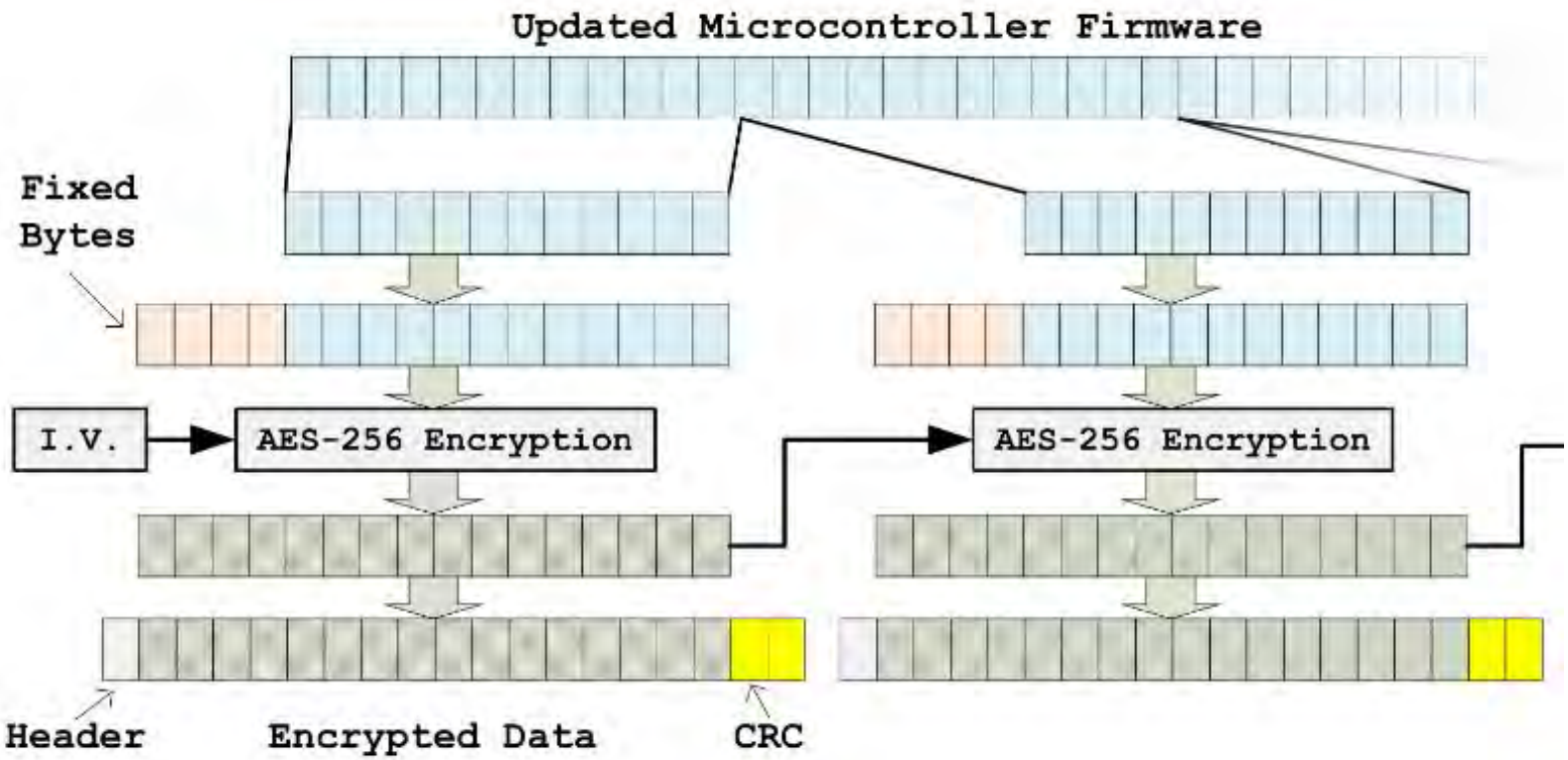
Substitution-Box (Lookup Table)

DEMO TIME!

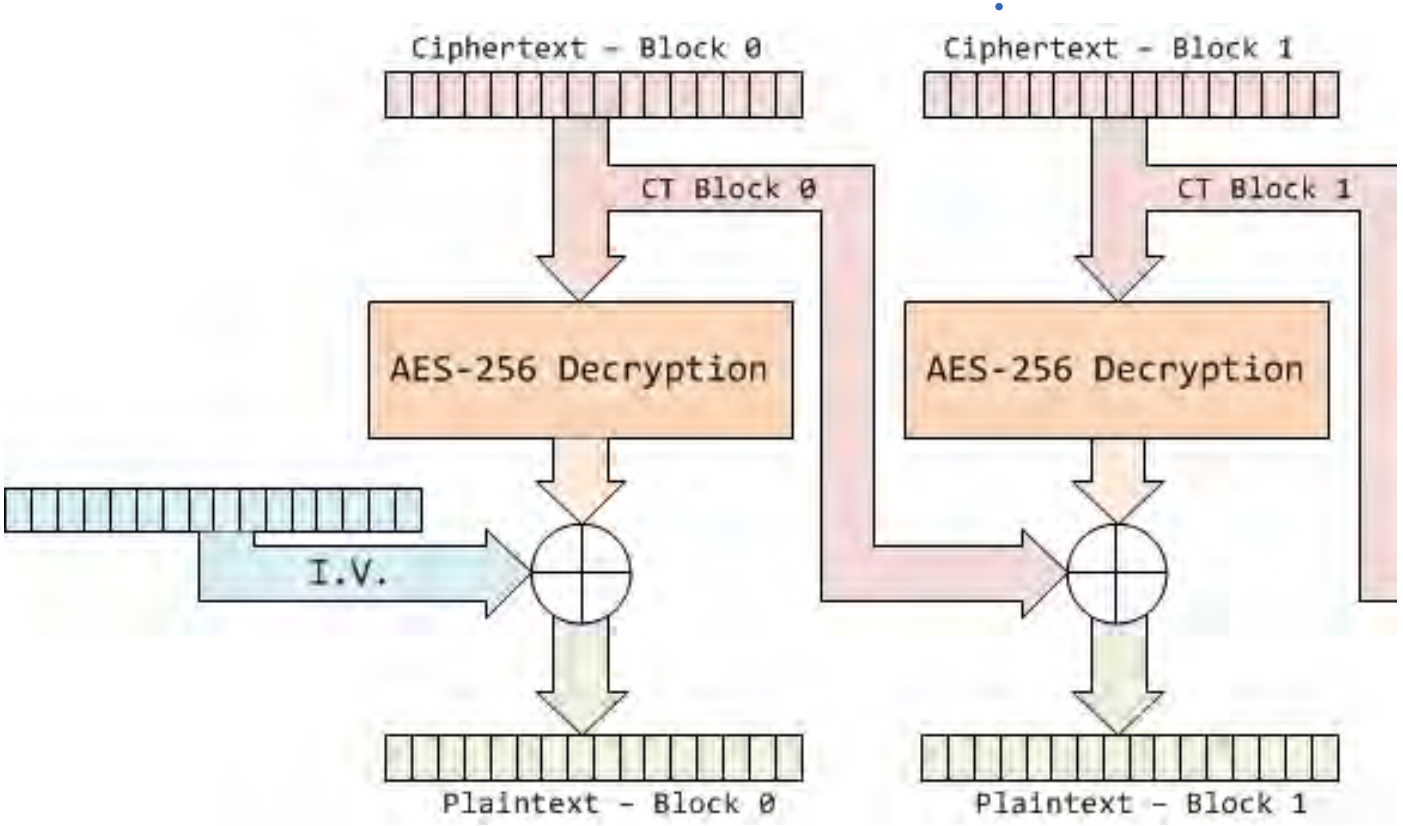
AES-256 Bootloader (Example #1)

Tutorial:
http://newae.com/sidechannel/cwdocs/tutorialaes256boot.html
Paper (CCECE 2015):
https://eprint.iacr.org/2014/899.pdf

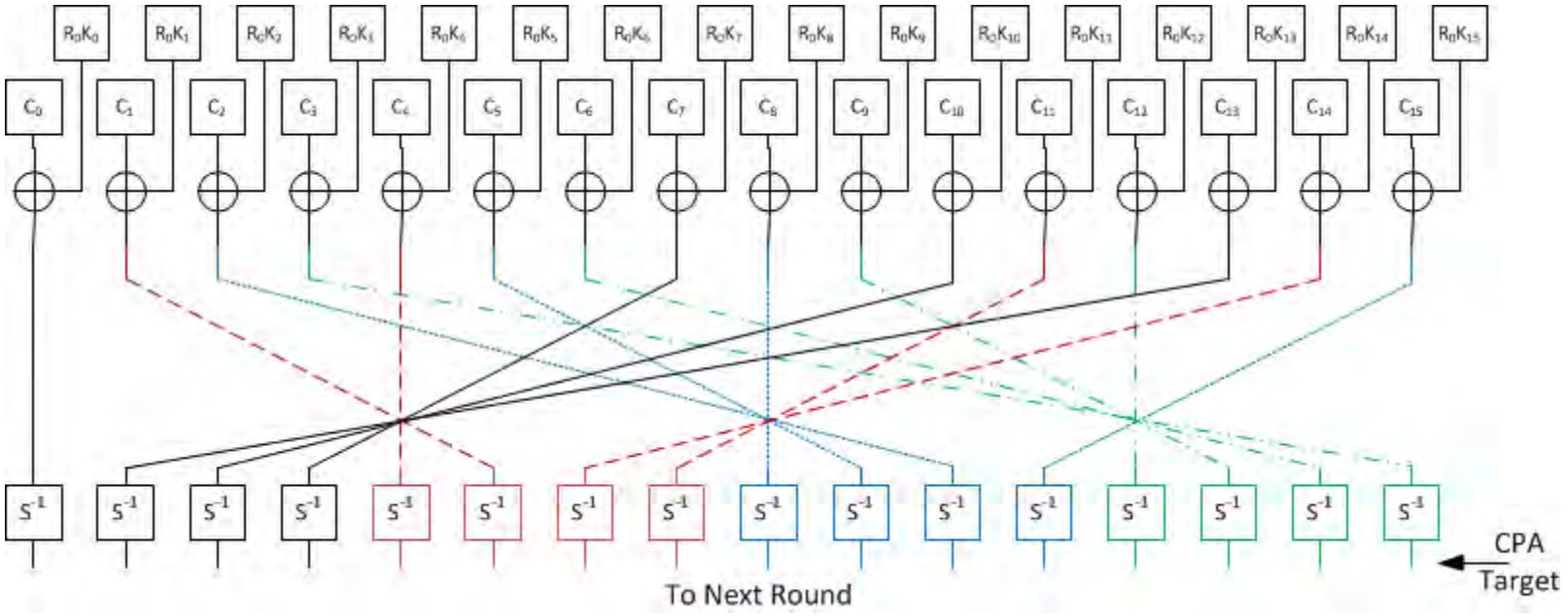
Bootloader Format



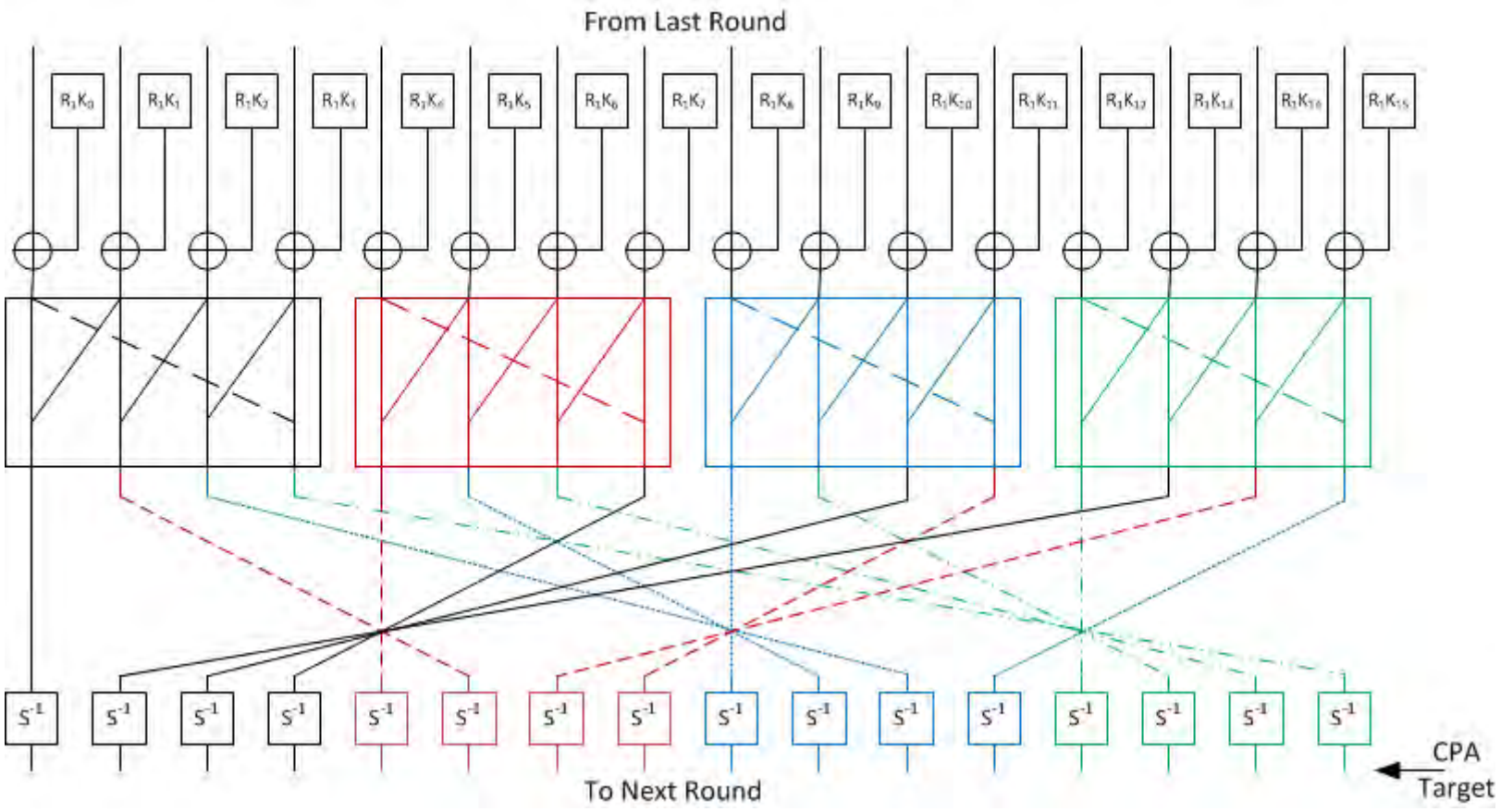
AES IN C.B.C. MODE



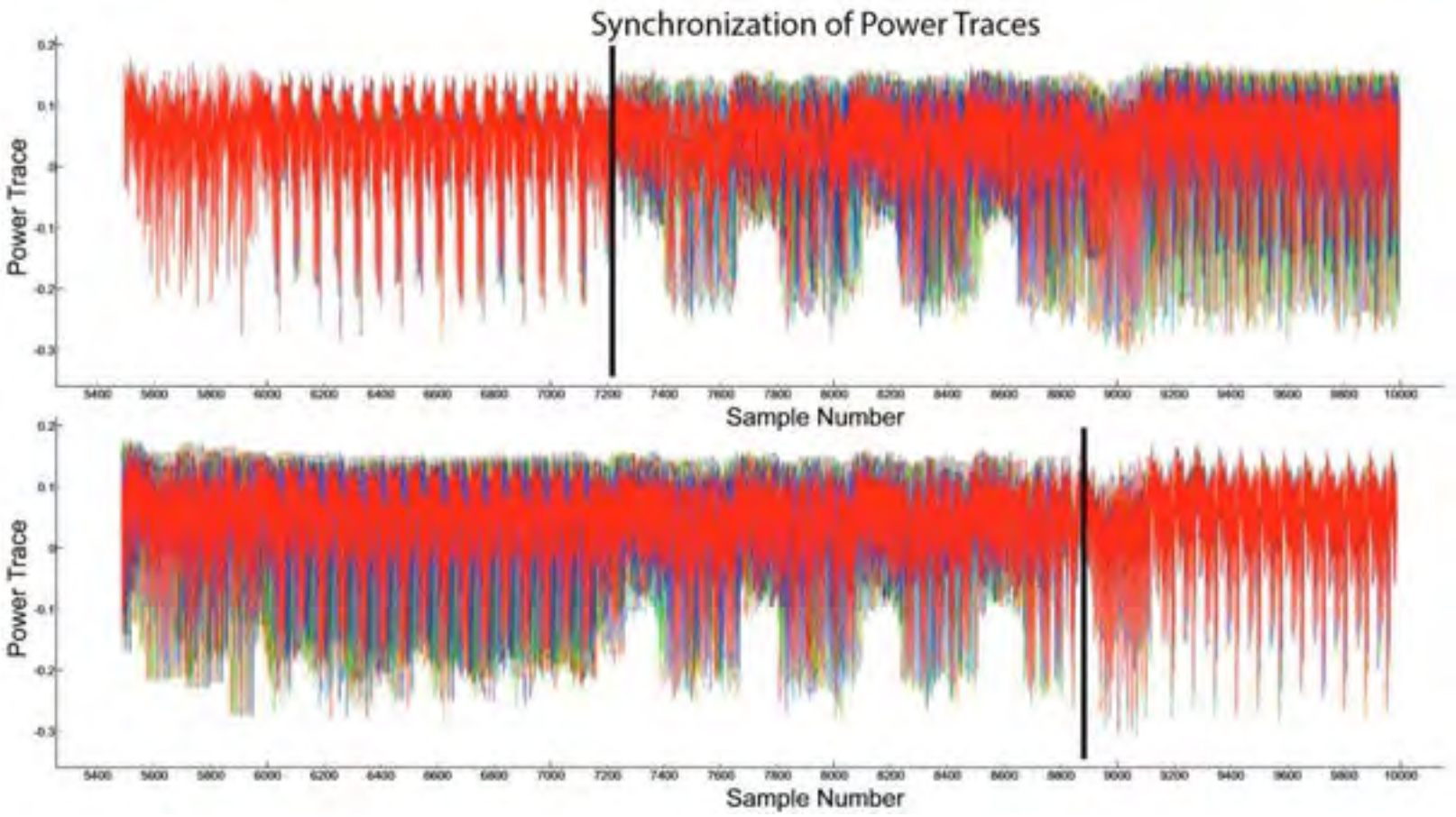
AES DECRYPTION



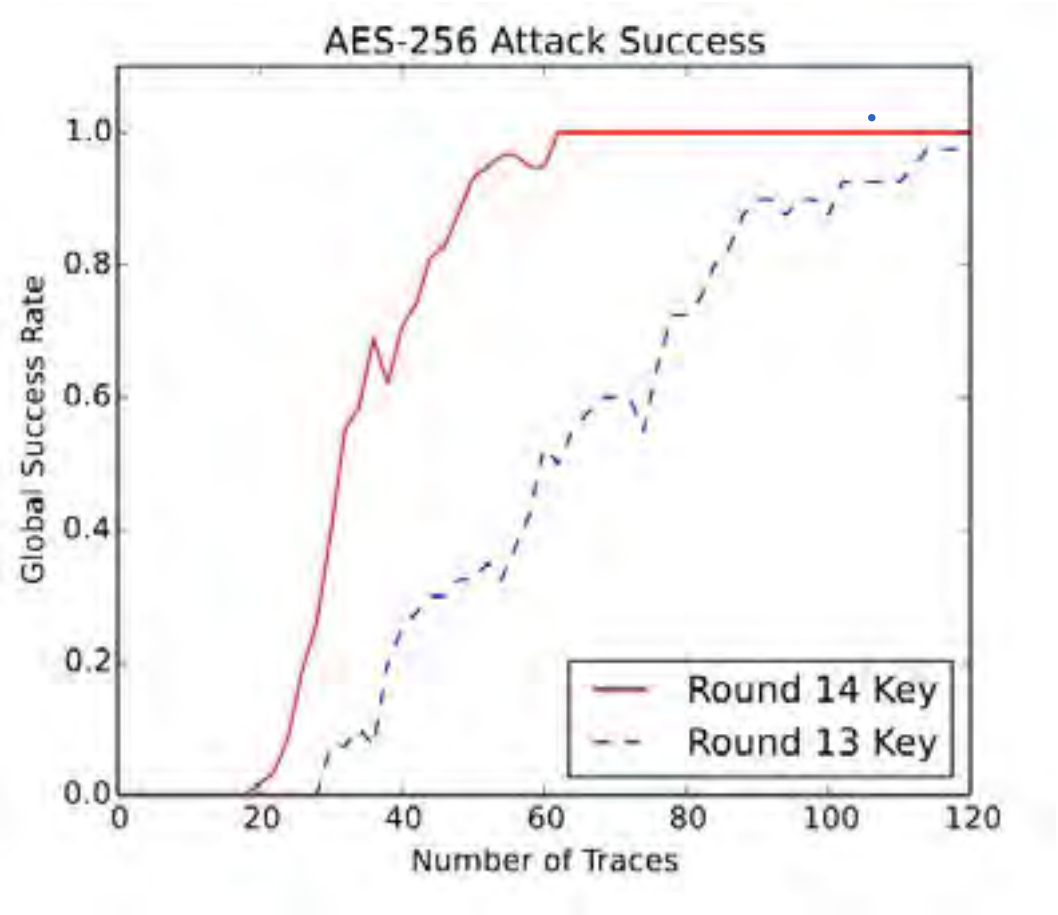
cont'd


From Last Round
R_1K_0
R_1K_1
R_1K_2
R_1K_3
R_1K_4
R_1K_5
R_1K_6
R_1K_7
R_1K_8
R_1K_9
R_1K_{10}
R_1K_{11}
R_1K_{12}
R_1K_{13}
R_1K_{14}
R_1K_{15}
S^{-1}
To Next Round
CPA
Target

RESYNC.


Synchronization of Power Traces
Power Trace
Sample Number
Power Trace
Sample Number

Success Rate



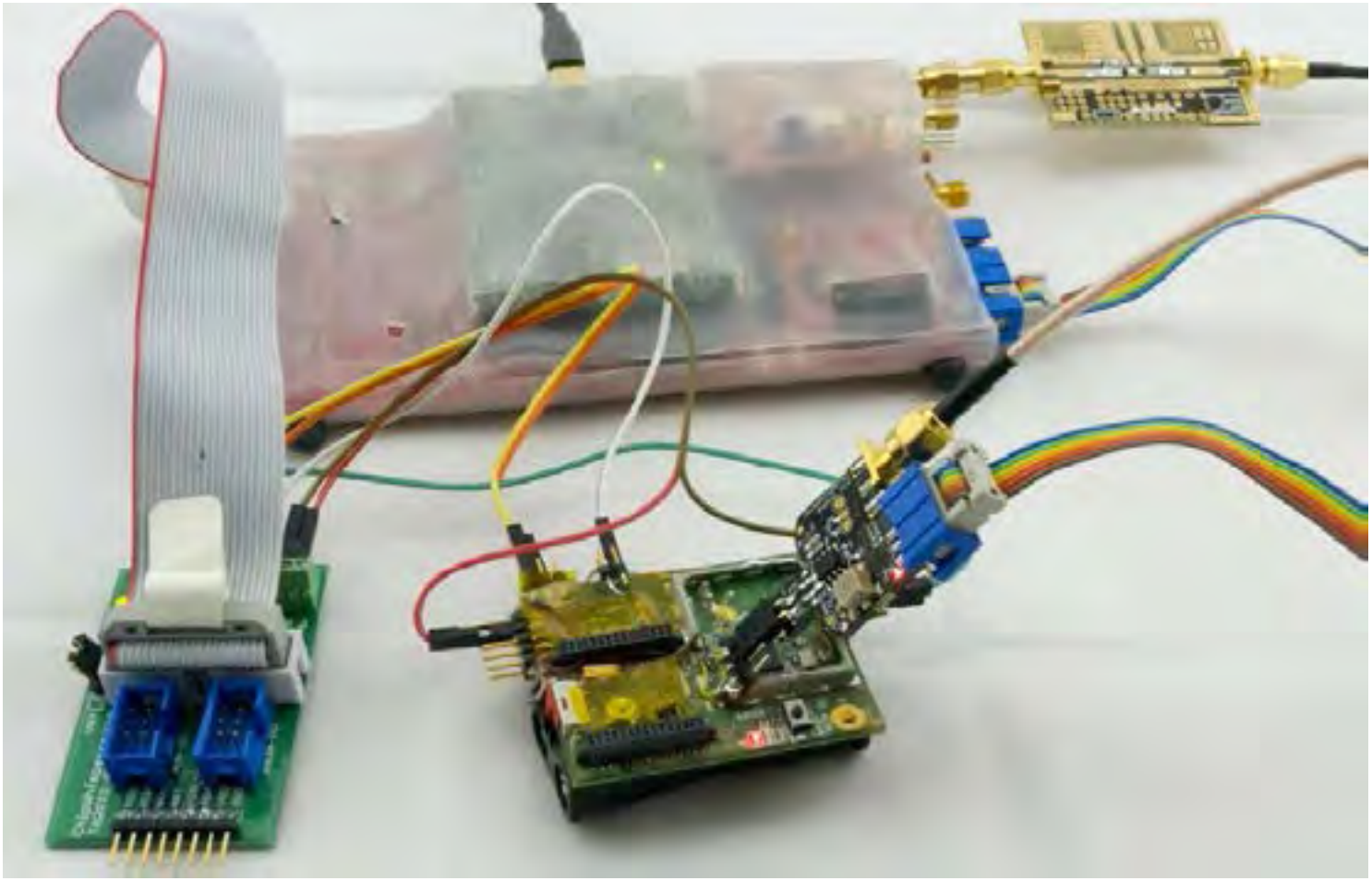
IEEE 802.15.4 (Example #2)

ZigBee (ZigBee IP, ZigBee Pro, RF4CE, etc.)
WirelessHART
MiWi
ISA100.11a
6LoWPAN
Nest Weave
JenNet
Thread
Atmel Lightweight Mesh
IEEE 802.15.5
DigiMesh

802.15.4 Node

http://eprint.iacr.org/2015/529

Platform

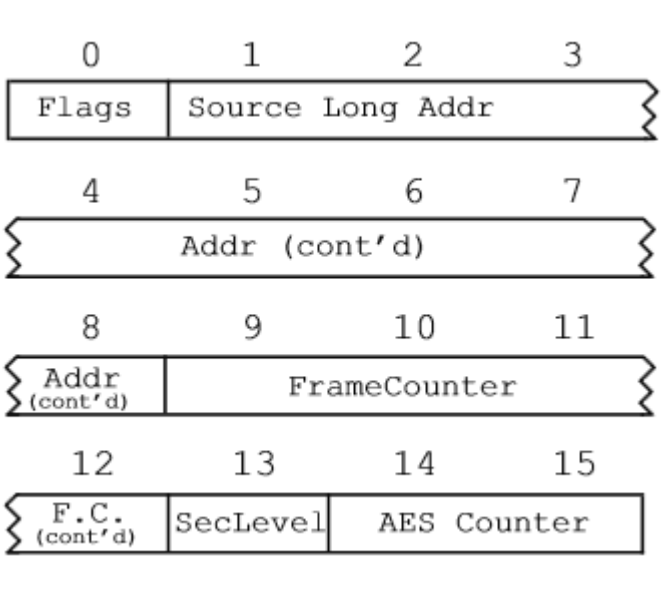


REGULAR OPS

IEEE 802.15.4 Wireless Stack: Frame Decryption Procedure:

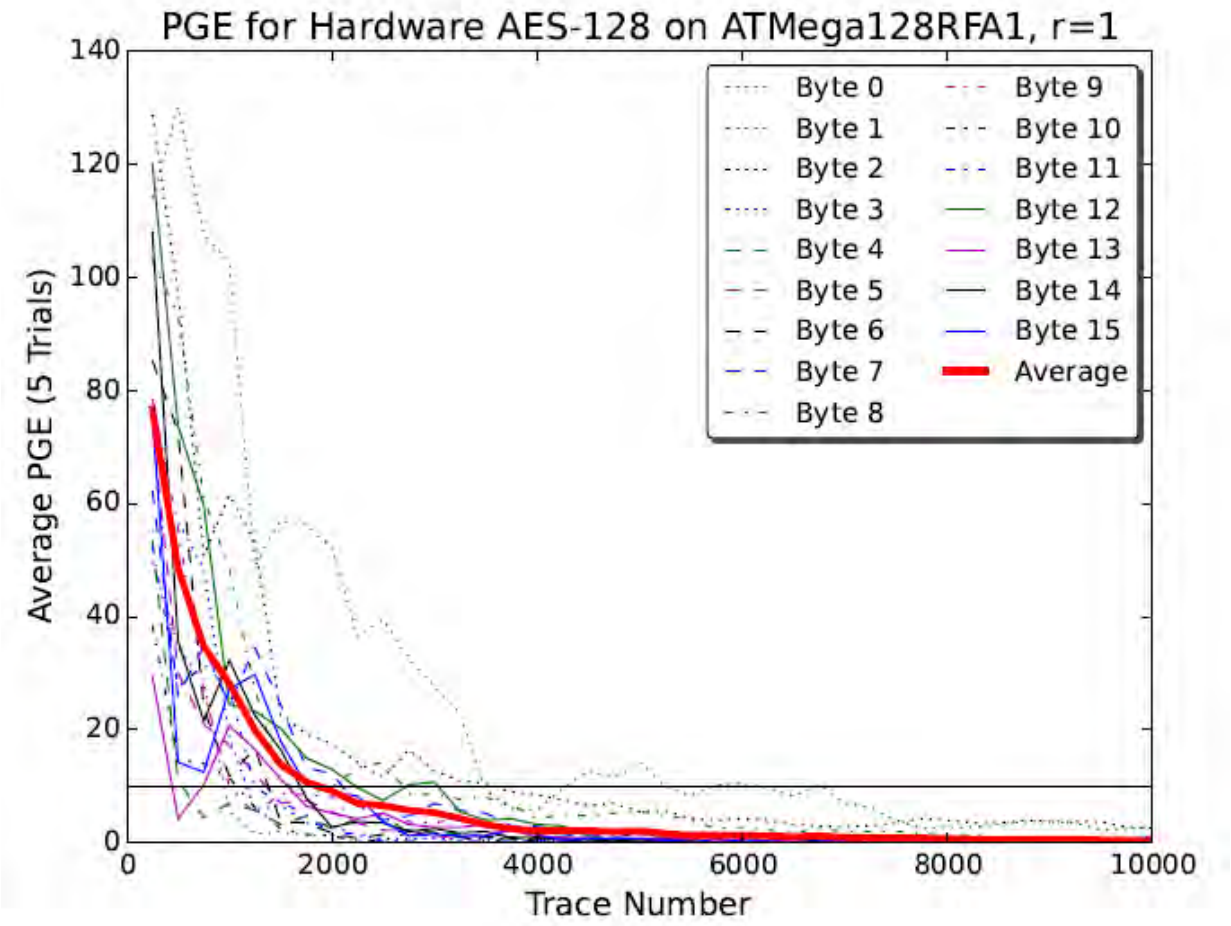
1. Validate headers and security options.
2. Check that the received frame counter is numerically greater than the last stored frame count.
3. Look up the secret key based on message address and/or key index.
4. Decrypt the payload (and MAC if present).
5. Validate the MAC (if present).
6. Store the frame counter.

AES Input

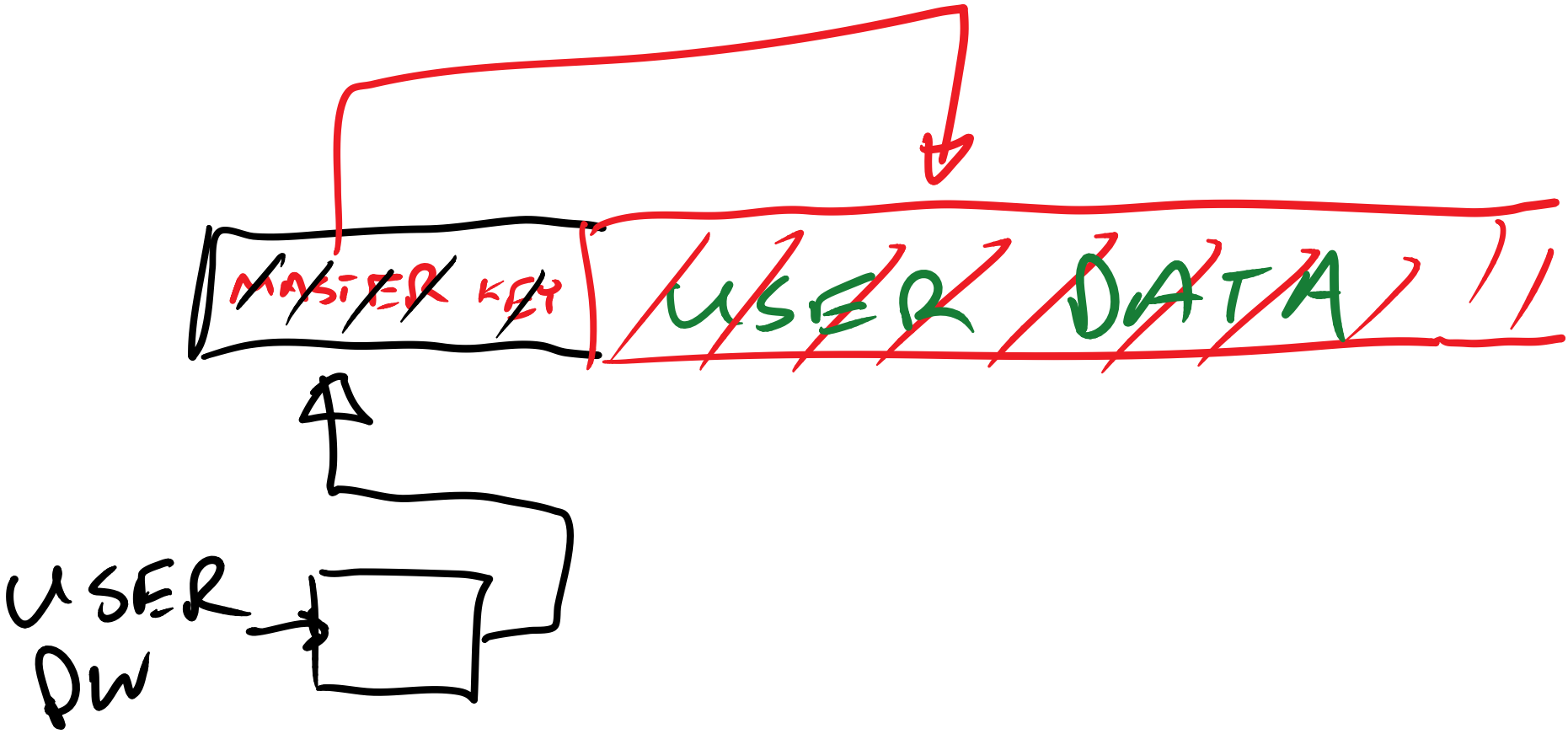


Input to AES Block

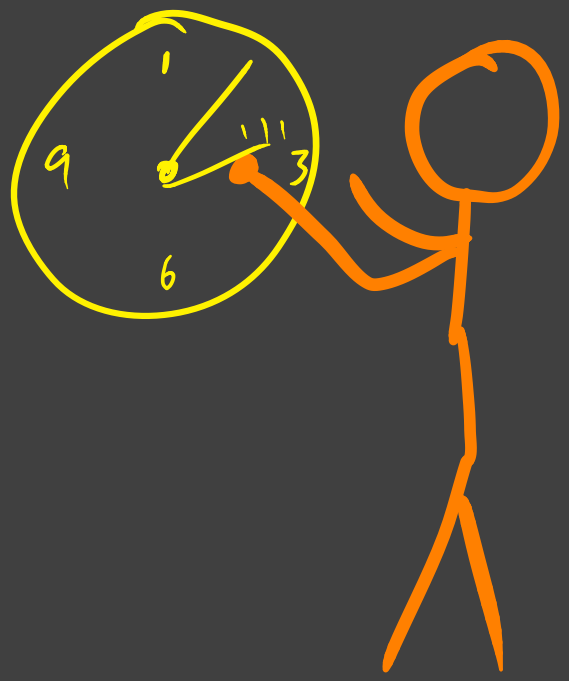
REAL-LIFE?


PGE for Hardware AES-128 on ATMega128RFA1, r=1
Average PGE (5 Trials)
Trace Number
0
20
40
60
80
100
120
140
0
2000
4000
6000
8000
10000
Byte 0
Byte 1
Byte 2
Byte 3
Byte 4
Byte 5
Byte 6
Byte 7
Byte 8
Byte 9
Byte 10
Byte 11
Byte 12
Byte 13
Byte 14
Byte 15
Average

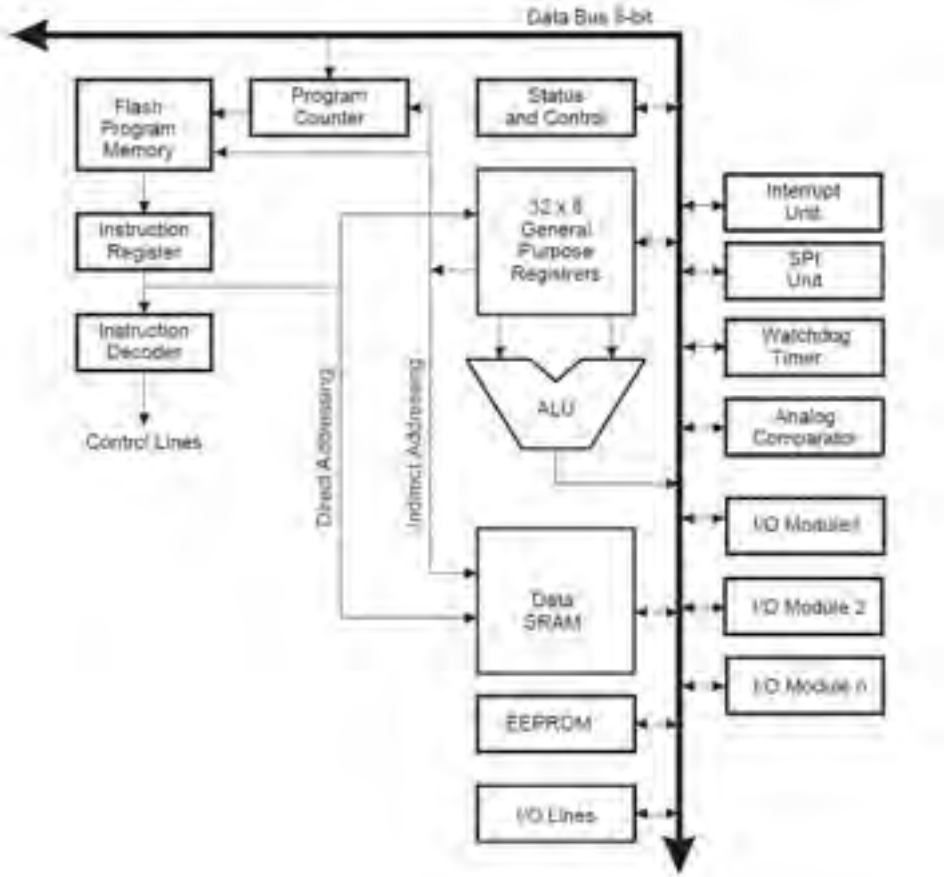
ENCRYPTED DRIVE (EXAMPLE #3)



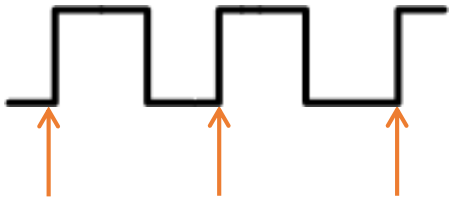
Clock Glitching

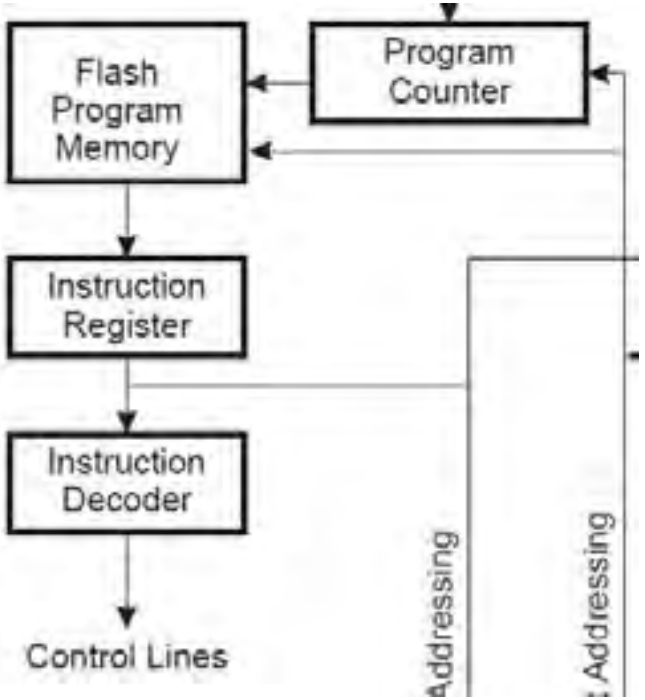


Clock Glitching

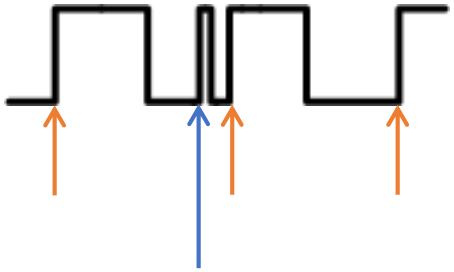


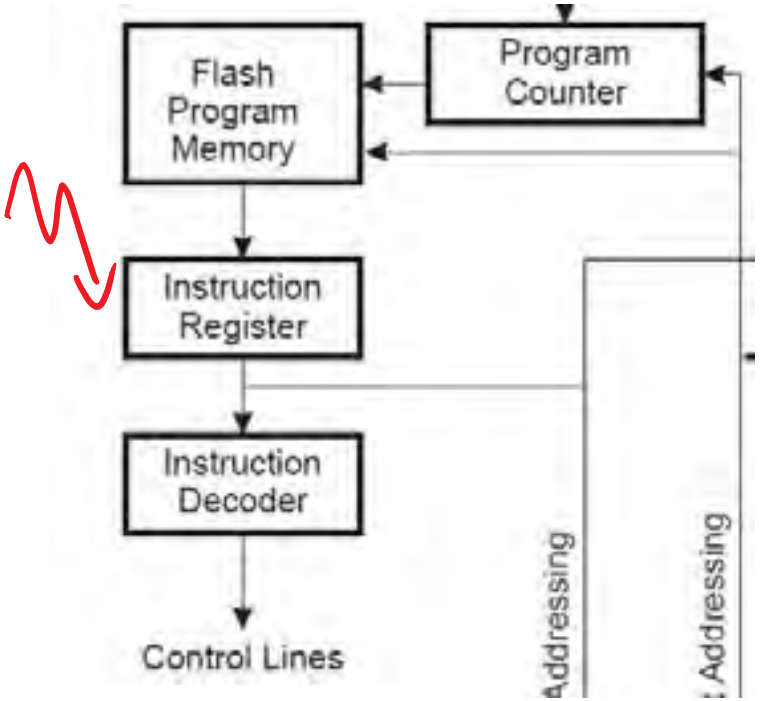
Clock Glitching



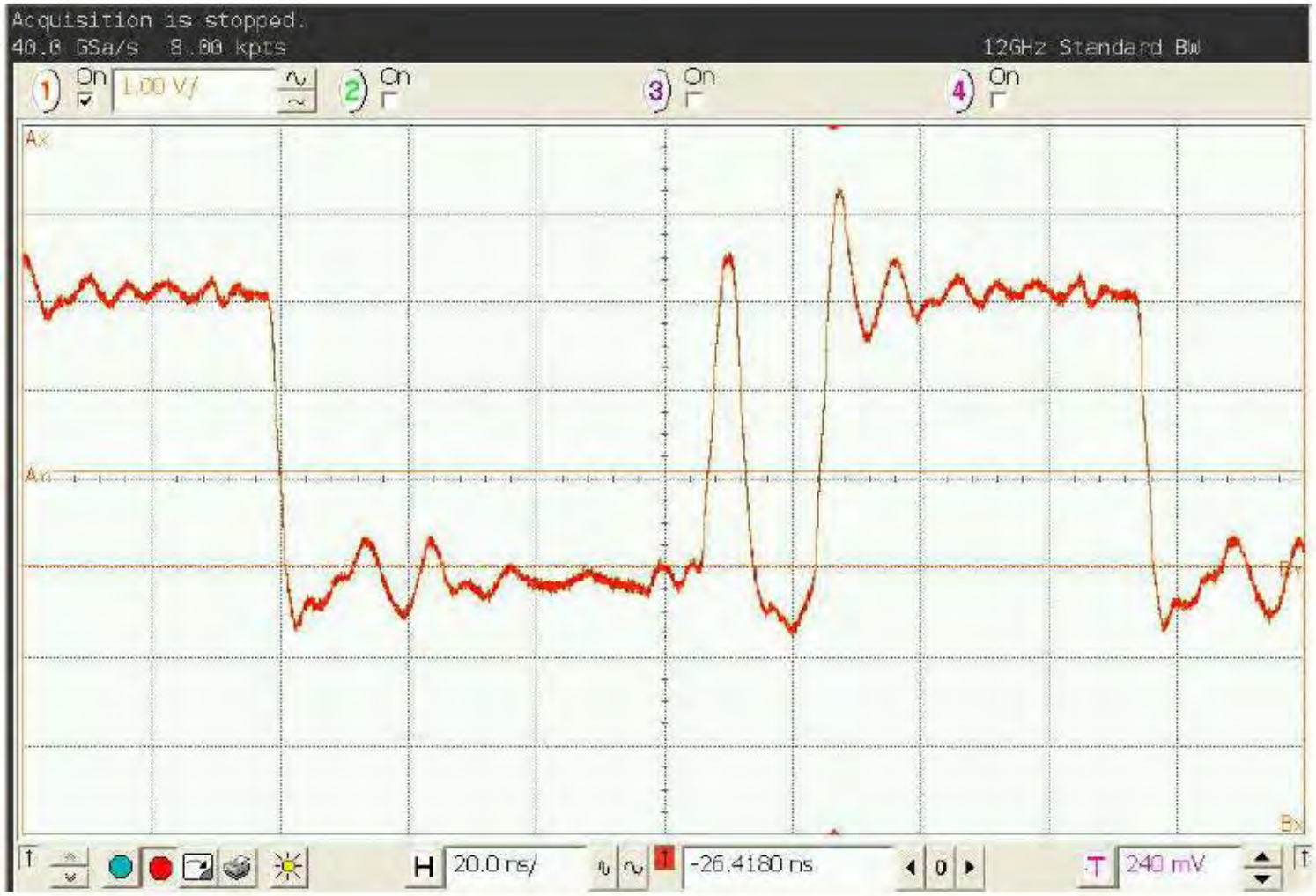


Clock Glitching

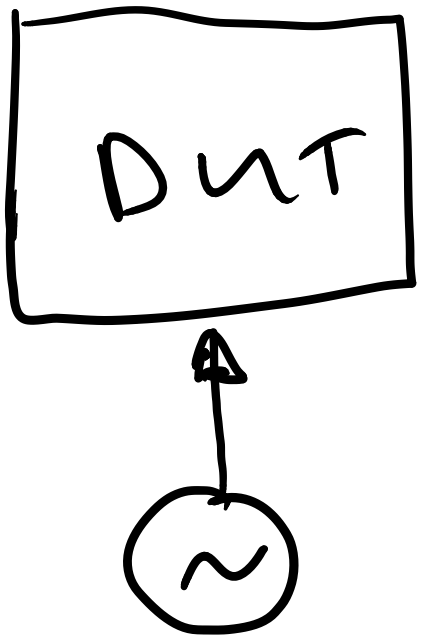




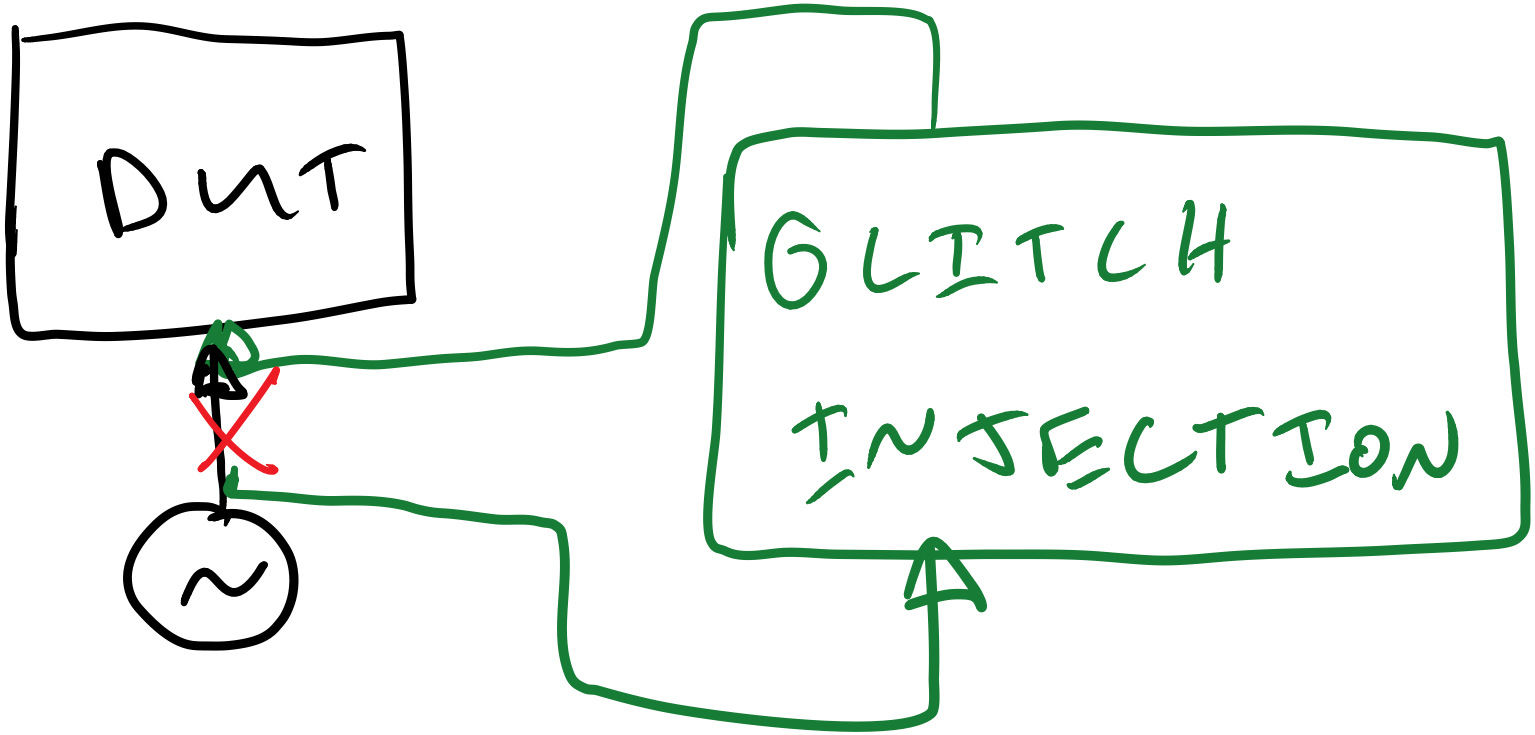
Example Clock Glitch



Example Clock Glitch



Example Clock Glitch



Linux

linux-util-2.24

```
/*
 *   auth.c -- PAM authorization code, common between chsh and chfn
 *   (c) 2012 by Cody Maloney <cmaloney@theoreticalchaos.com>
 *
 *   this program is free software.  you can redistribute it and
 *   modify it under the terms of the gnu general public license.
 *   there is no warranty.
 *
 */

#include "auth.h"
#include "pamfail.h"

int auth_pam(const char *service_name, uid_t uid, const char *username)
{
                if (uid != 0) {
                                pam_handle_t *pamh = NULL;
                                struct pam_conv conv = { misc_conv, NULL };
                                int retcode;

                                retcode = pam_start(service_name, username, &conv, &pamh);
                                if (pam_fail_check(pamh, retcode))
                                                return FALSE;

                                retcode = pam_authenticate(pamh, 0);
                                if (pam_fail_check(pamh, retcode))
                                                return FALSE;

                                retcode = pam_acct_mgmt(pamh, 0);
                                if (retcode == PAM_NEW_AUTHTOK_REQD)
                                                retcode =
                                                    pam_chauthtok(pamh, PAM_CHANGE_EXPIRED_AUTHTOK);
                                if (pam_fail_check(pamh, retcode))
                                                return FALSE;

                                retcode = pam_setcred(pamh, 0);
                                if (pam_fail_check(pamh, retcode))
                                                return FALSE;

                                pam_end(pamh, 0);
                                /* no need to establish a session; this isn't a
                                 * session-oriented activity...  */
                }
                return TRUE;
}
```

ANDROID

```
162
163     private void verifyPasswordAndUnlock() {
164         String entry = mPasswordEntry.getText().toString();
165         if (mLockPatternUtils.checkPassword(entry)) {
166             mCallback.keyguardDone(true);
167             mCallback.reportSuccessfulUnlockAttempt();
168         } else if (entry.length() > MINIMUM_PASSWORD_LENGTH_BEFORE_REPORT ) {
169             // to avoid accidental lockout, only count attempts that are long enough to be a
170             // real password. This may require some tweaking.
171             mCallback.reportFailedUnlockAttempt();
172             if (0 == (mUpdateMonitor.getFailedAttempts()
173                     % LockPatternUtils.FAILED_ATTEMPTS_BEFORE_TIMEOUT)) {
174                 long deadline = mLockPatternUtils.setLockoutAttemptDeadline();
175                 handleAttemptLockout(deadline);
176             }
177         }
178         mPasswordEntry.setText("");
179     }
180
181     // Prevent user from using the PIN/Password entry until scheduled deadline.
182     private void handleAttemptLockout(long elapsedRealtimeDeadline) {
183         mPasswordEntry.setEnabled(false);
184         mKeyboardView.setEnabled(false);
```

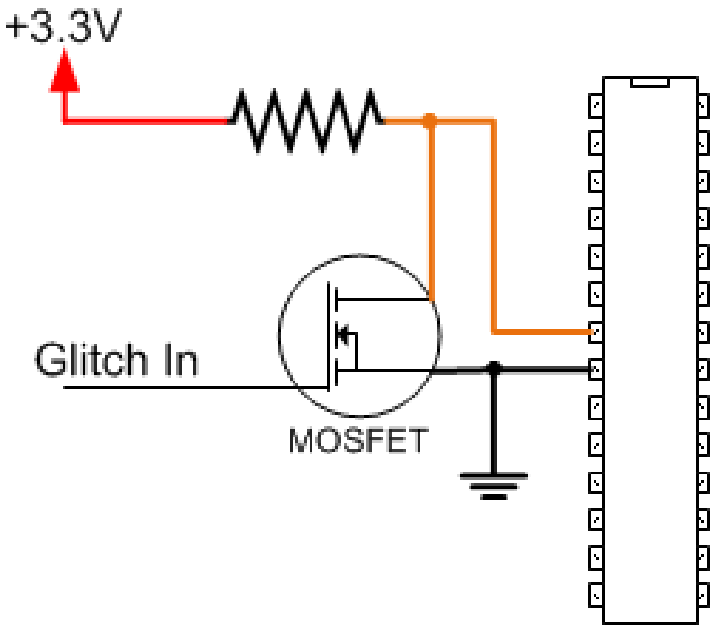
<Clock Glitching Movie>

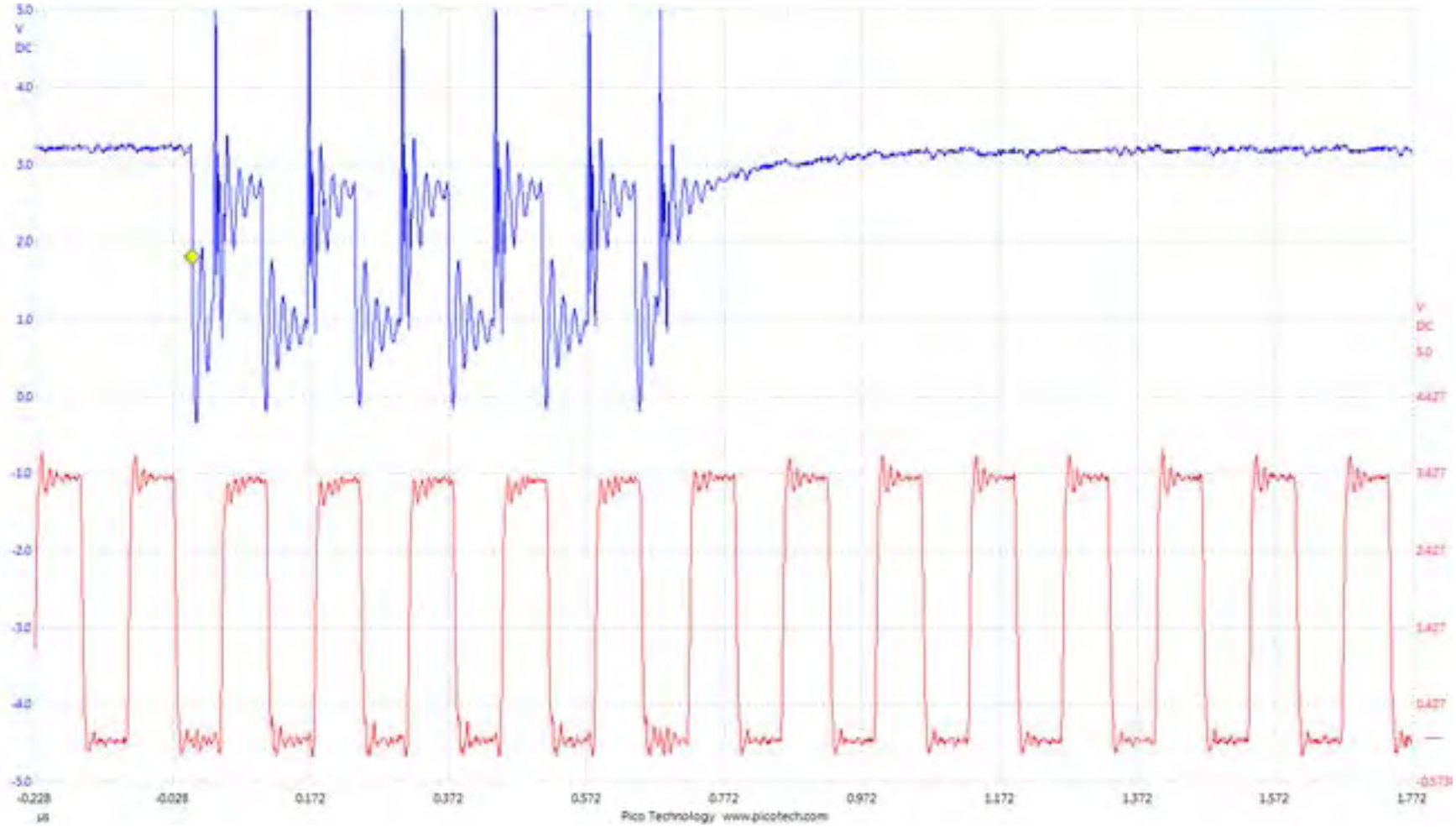
POWER GLITCHING

Demo Code

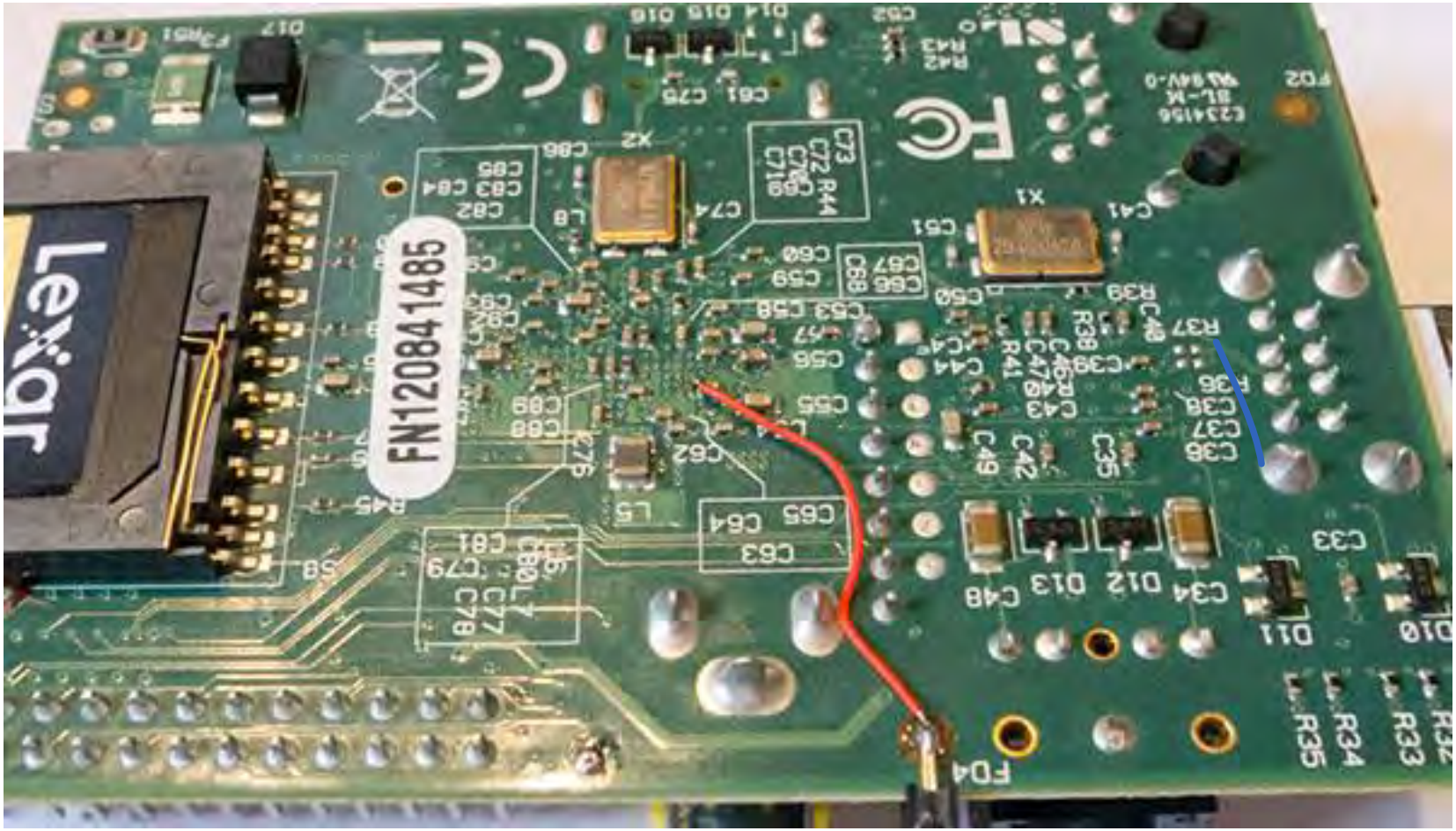
```
int i,j,count;

while(1){
    count = 0;

    for (j = 0; j < 5000; j++){
        for (i = 0; i < 5000; i++){
            count++;
        }
    }

    printf("%d %d %d\n", count, i, j);
}
```

EMBEDDED TARGET

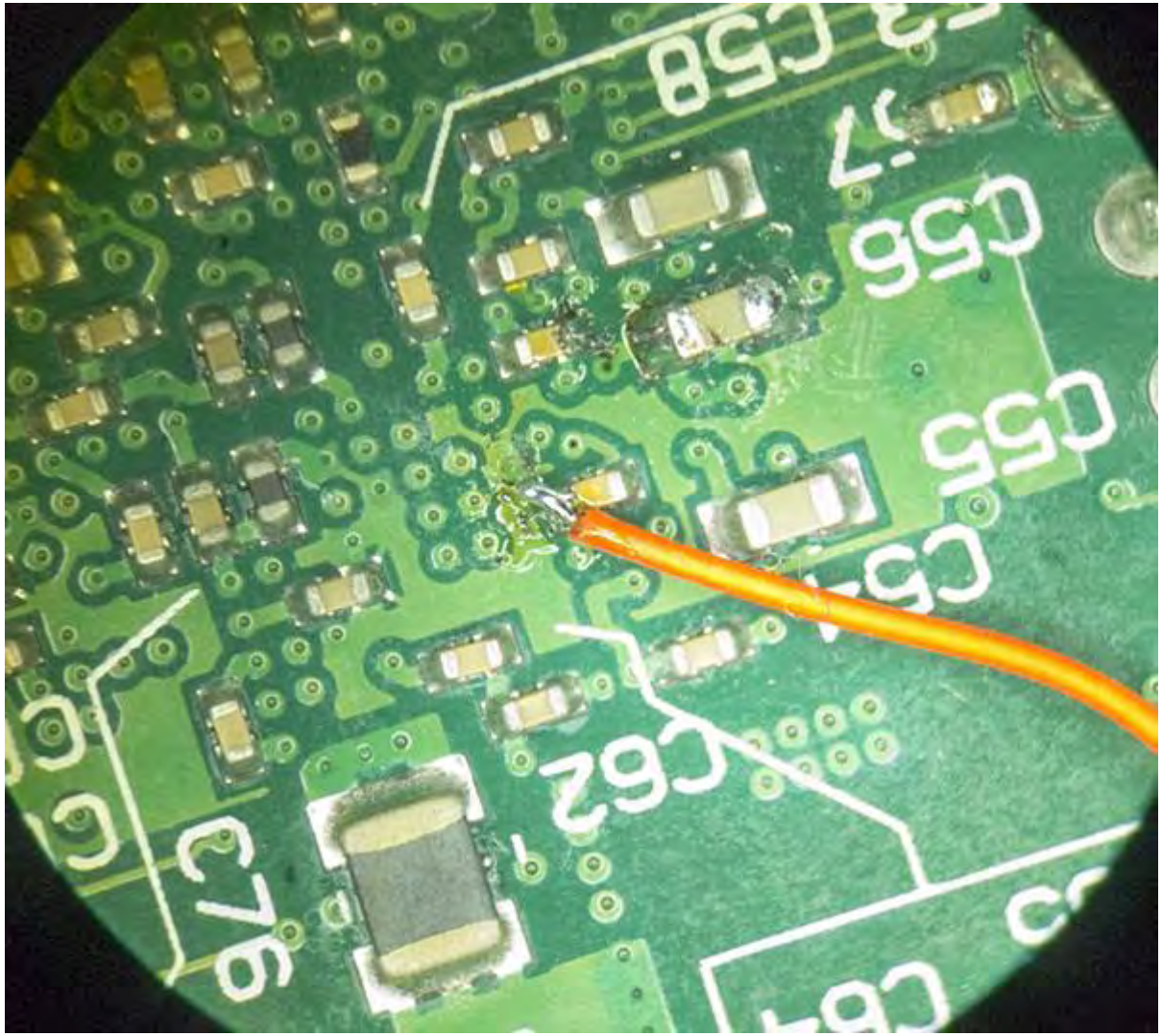



5.0
V
DC
4.0
3.0
2.0
1.0
0.0
-1.0
-2.0
-3.0
-4.0
-5.0
-0.228
µs
-0.028
0.172
0.372
0.572
0.772
0.972
1.172
1.372
1.572
1.772
Pico Technology www.picotech.com

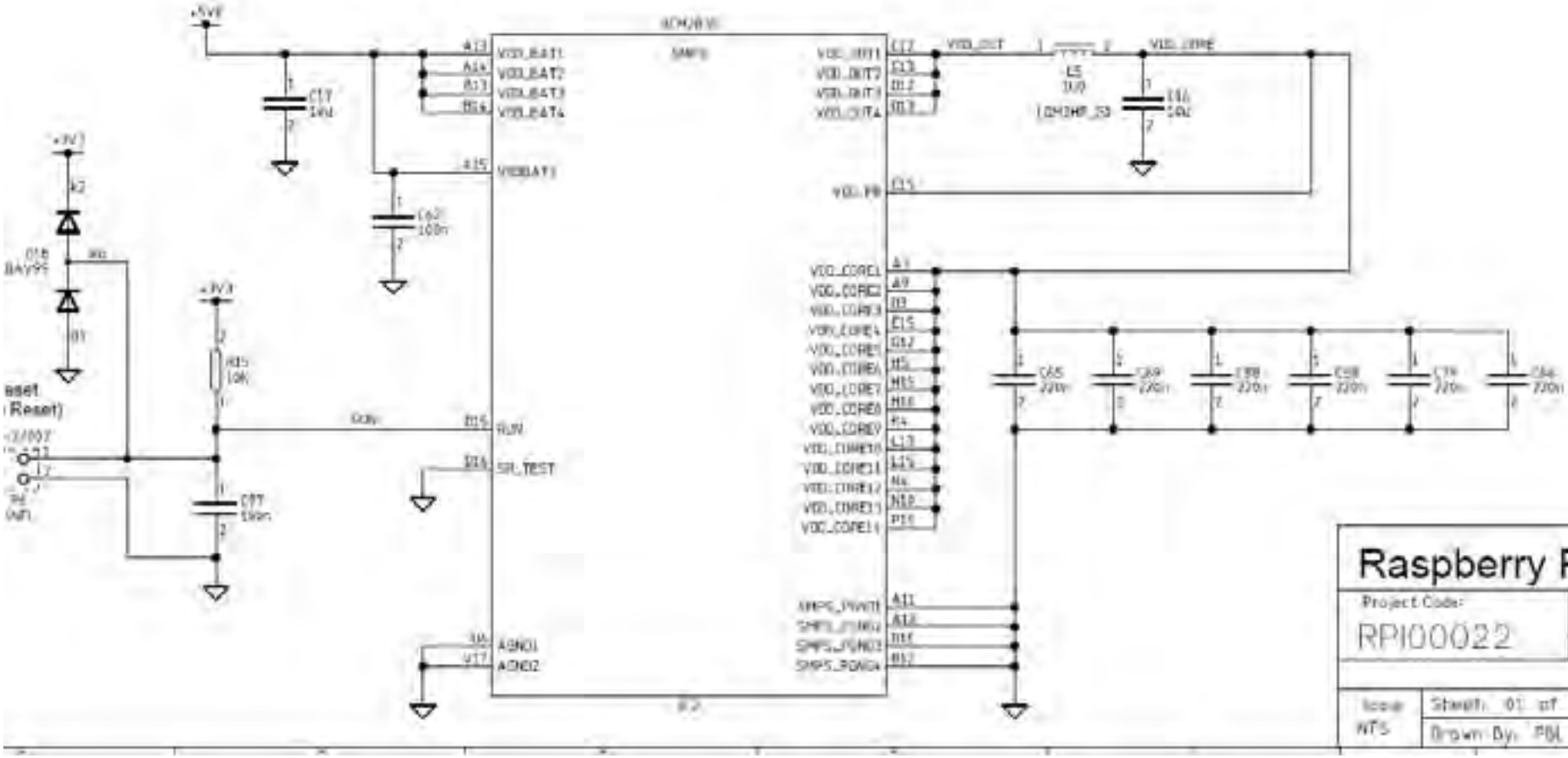
RASPBERRY PI



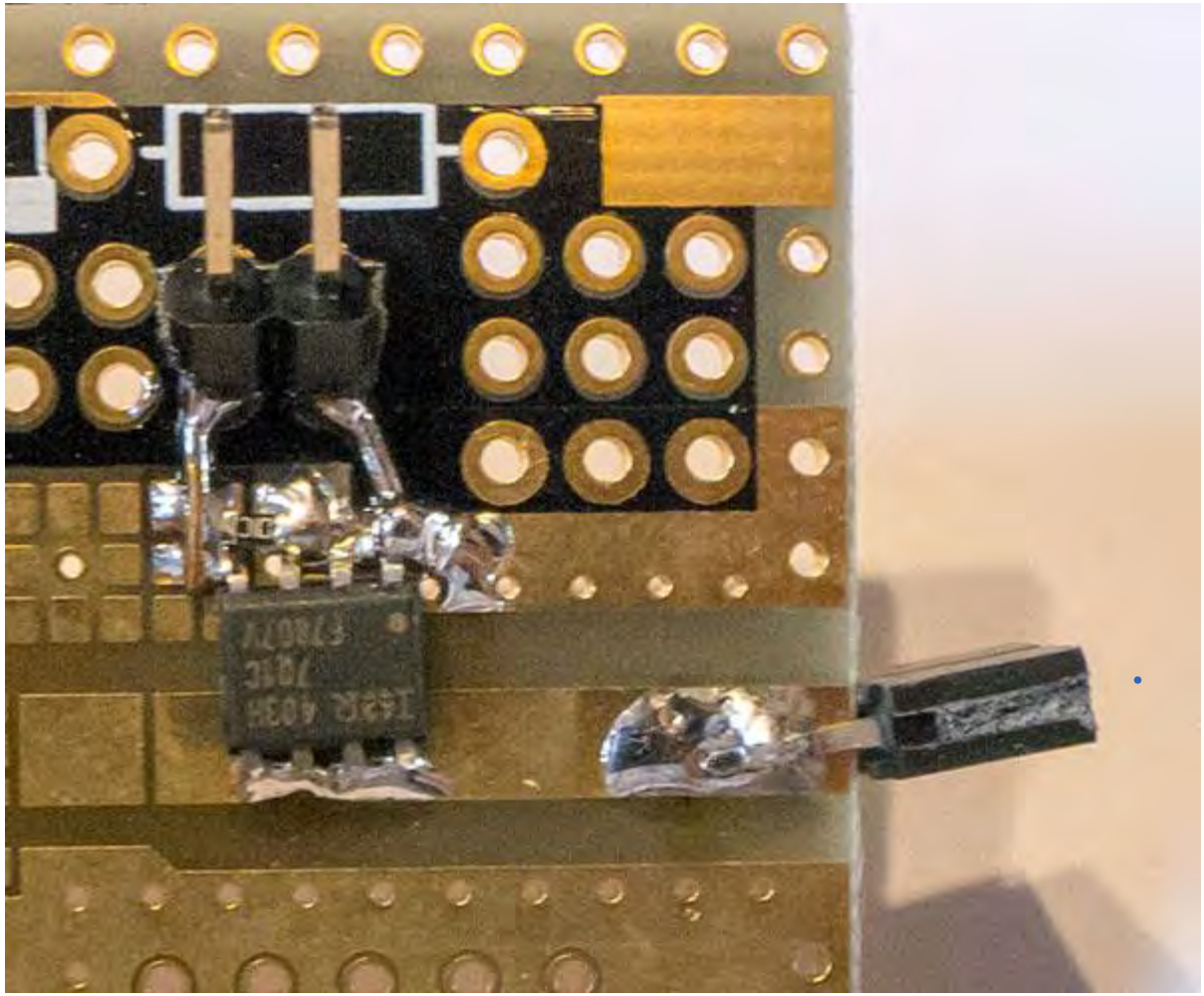
RASPBERRY PI



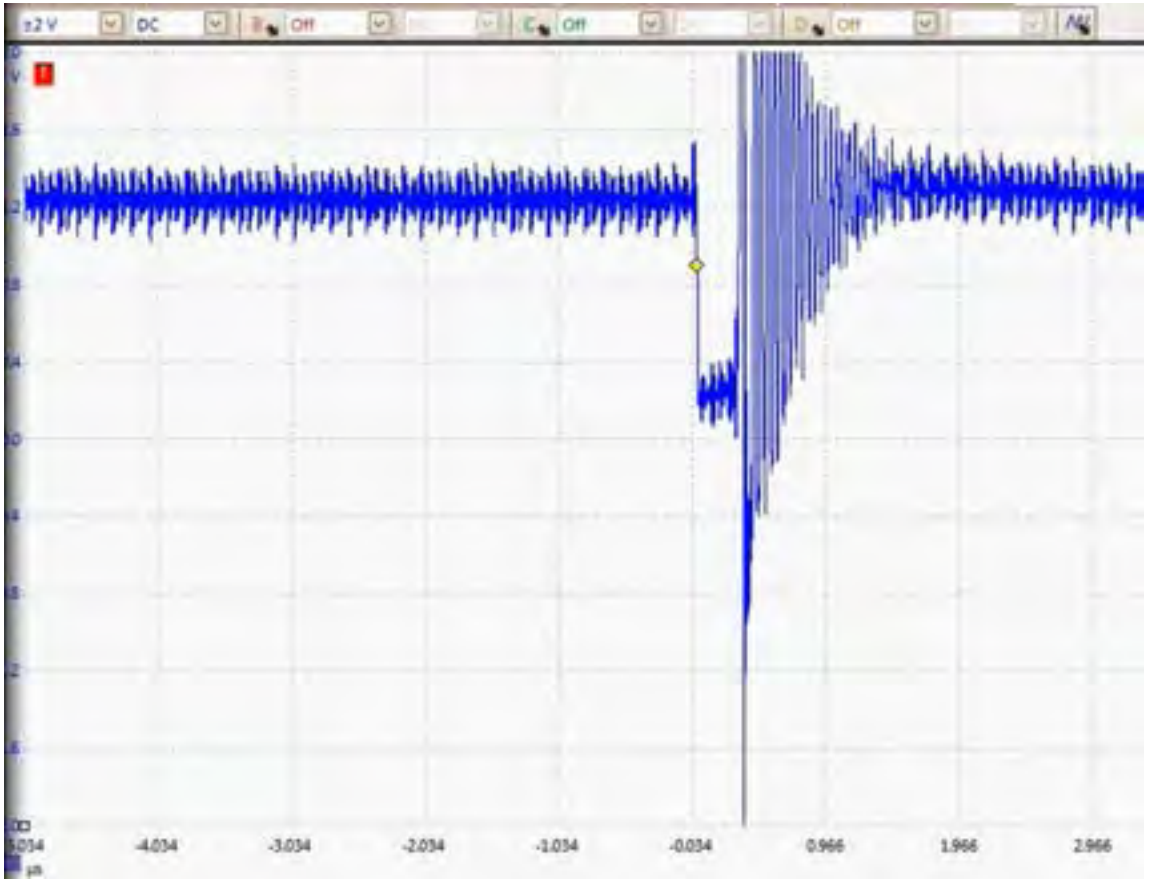
RASPBERRY PI



Glitch Tool



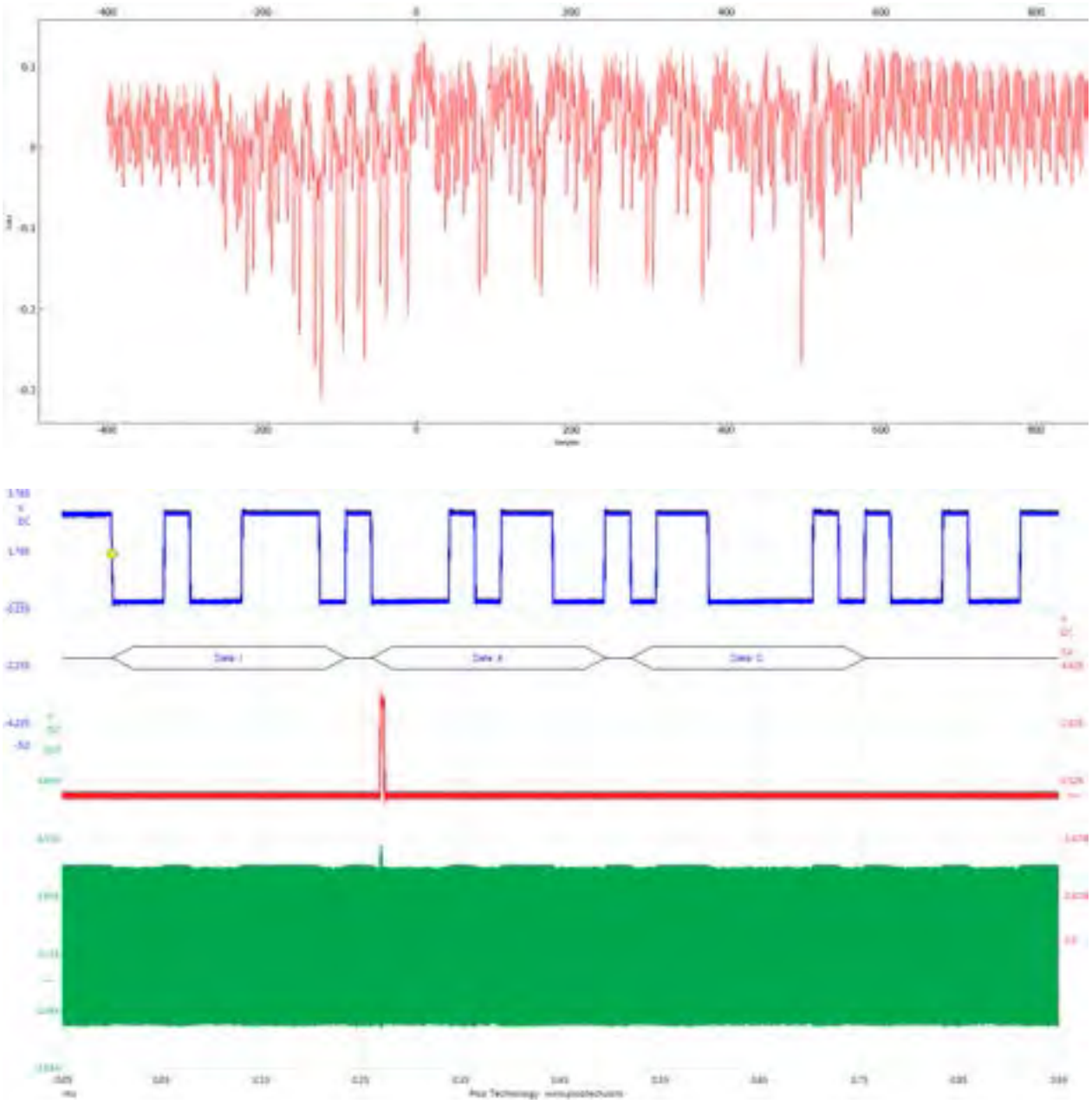
120MHz glitch clock (mul = 4, div = 1)
38 Cycles of clock being glitched = 315nS glitch



<Rapberry Pi Movie>

<Android Movie>

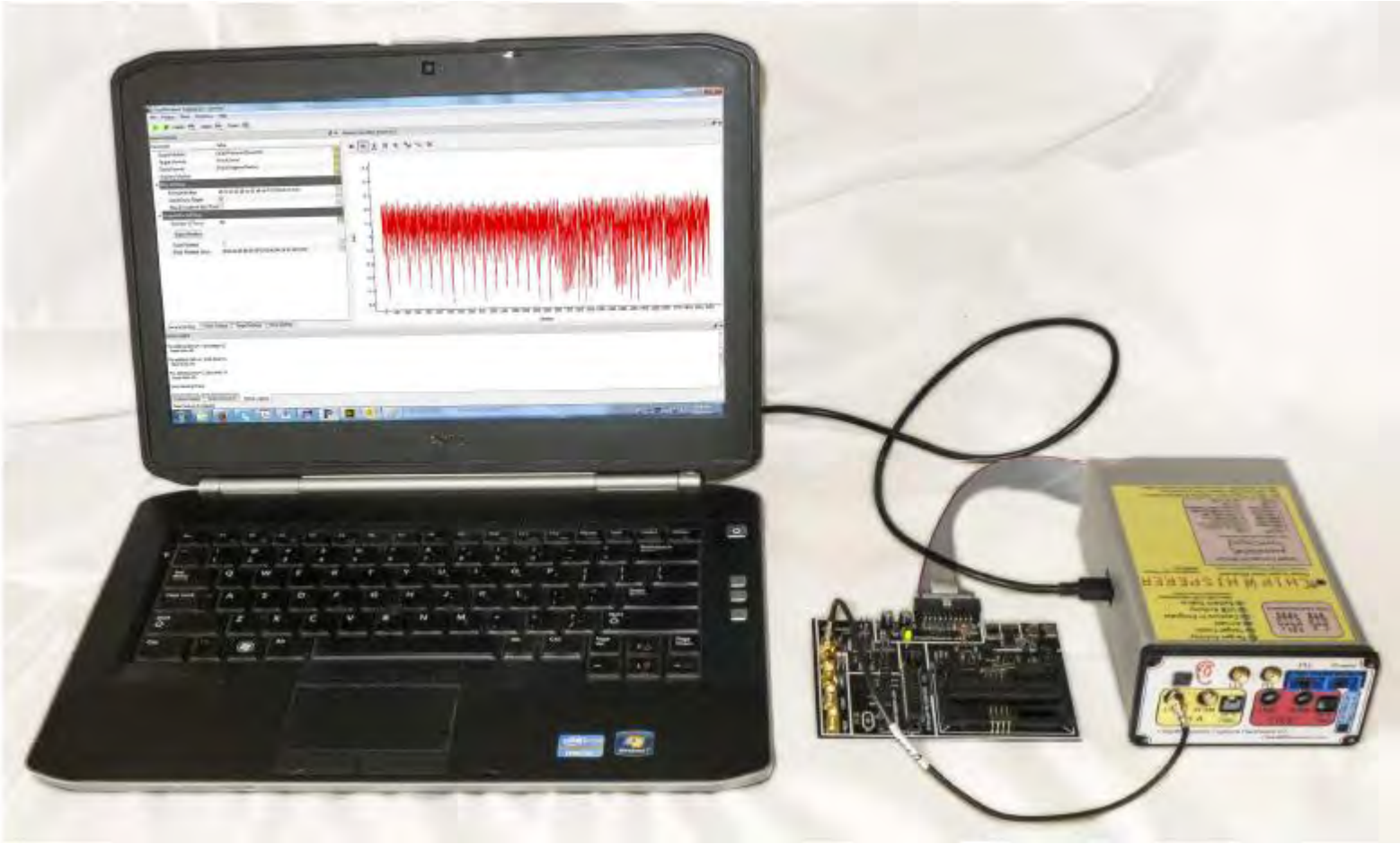
TRIGGERING



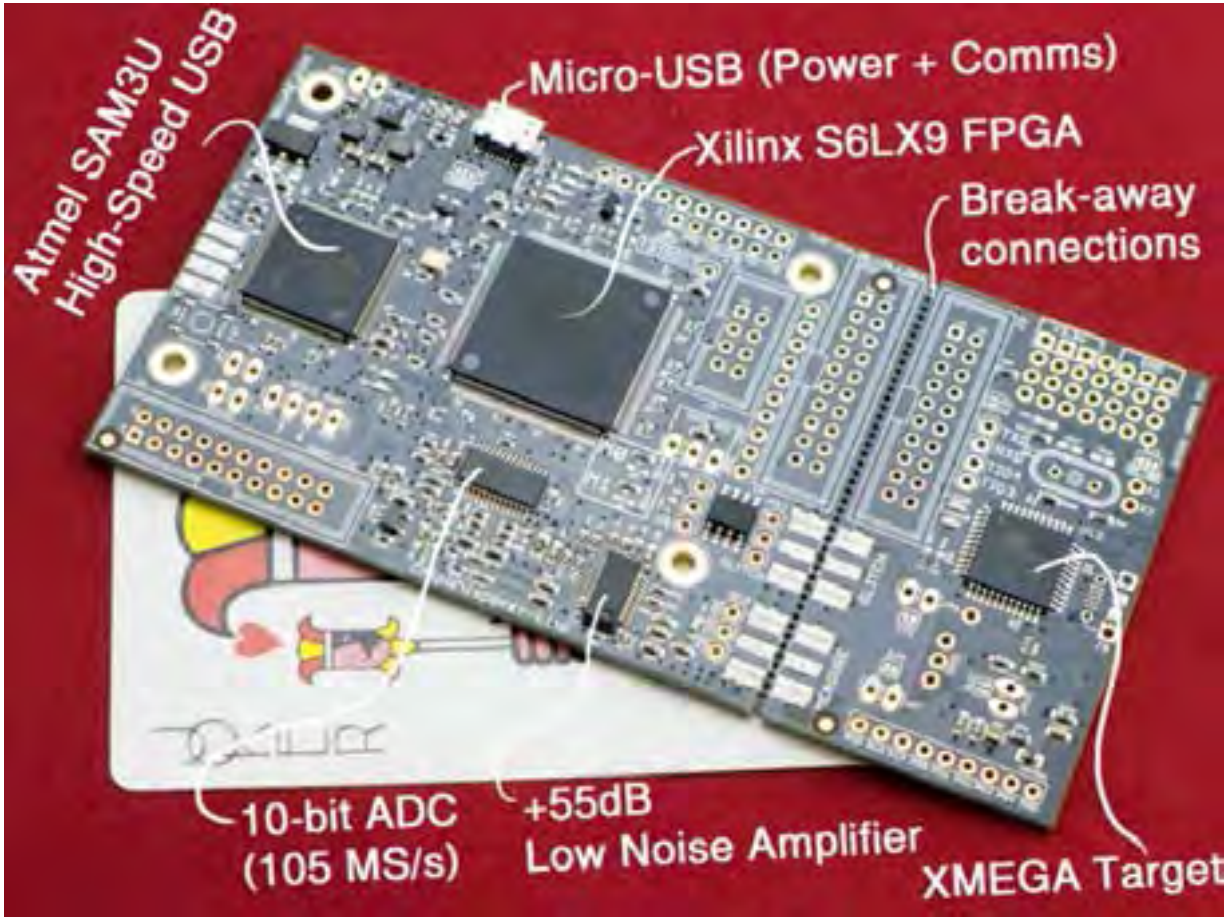
Open Source Tools

DPA HARDWARE

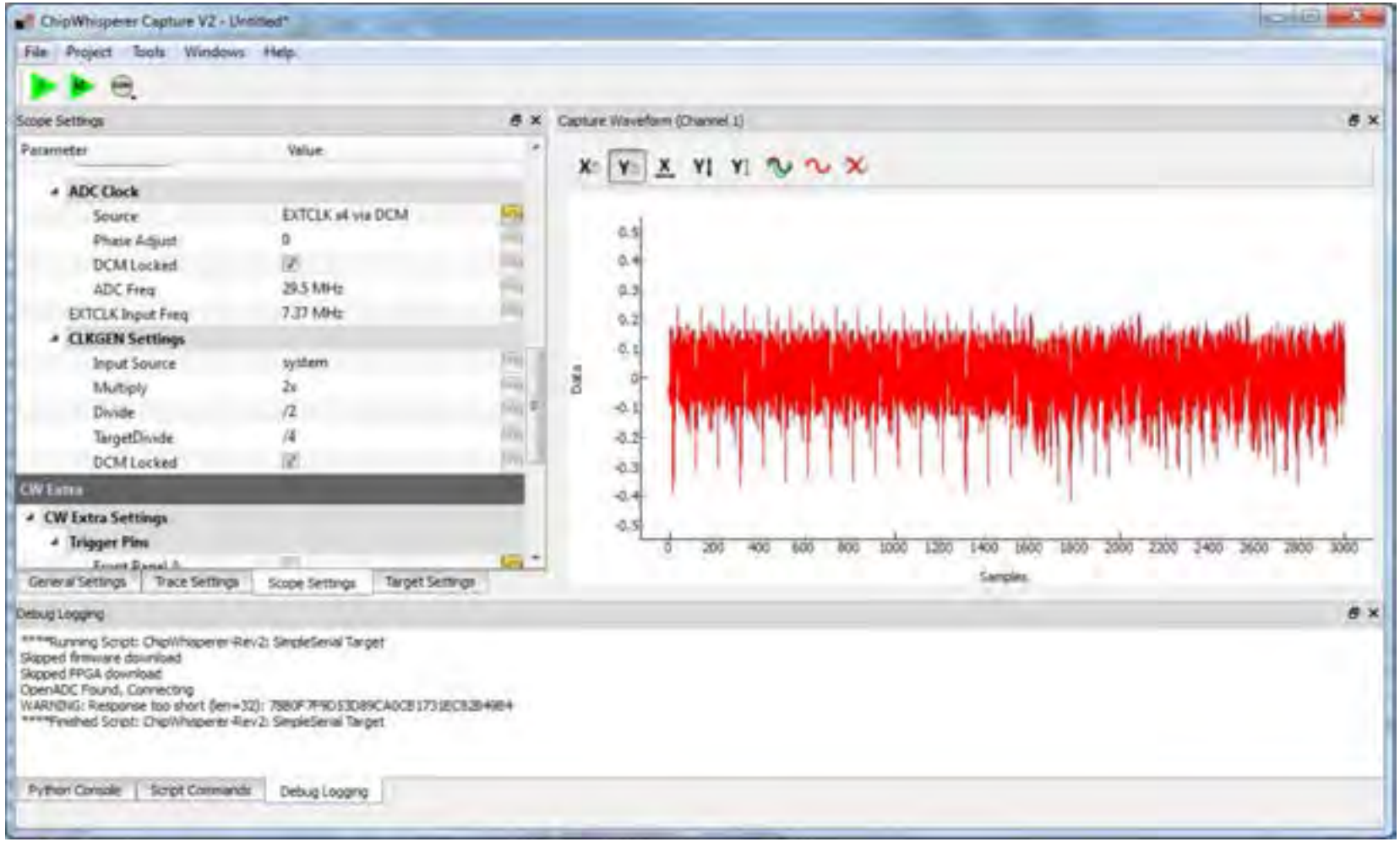
DPA LAB SETUP



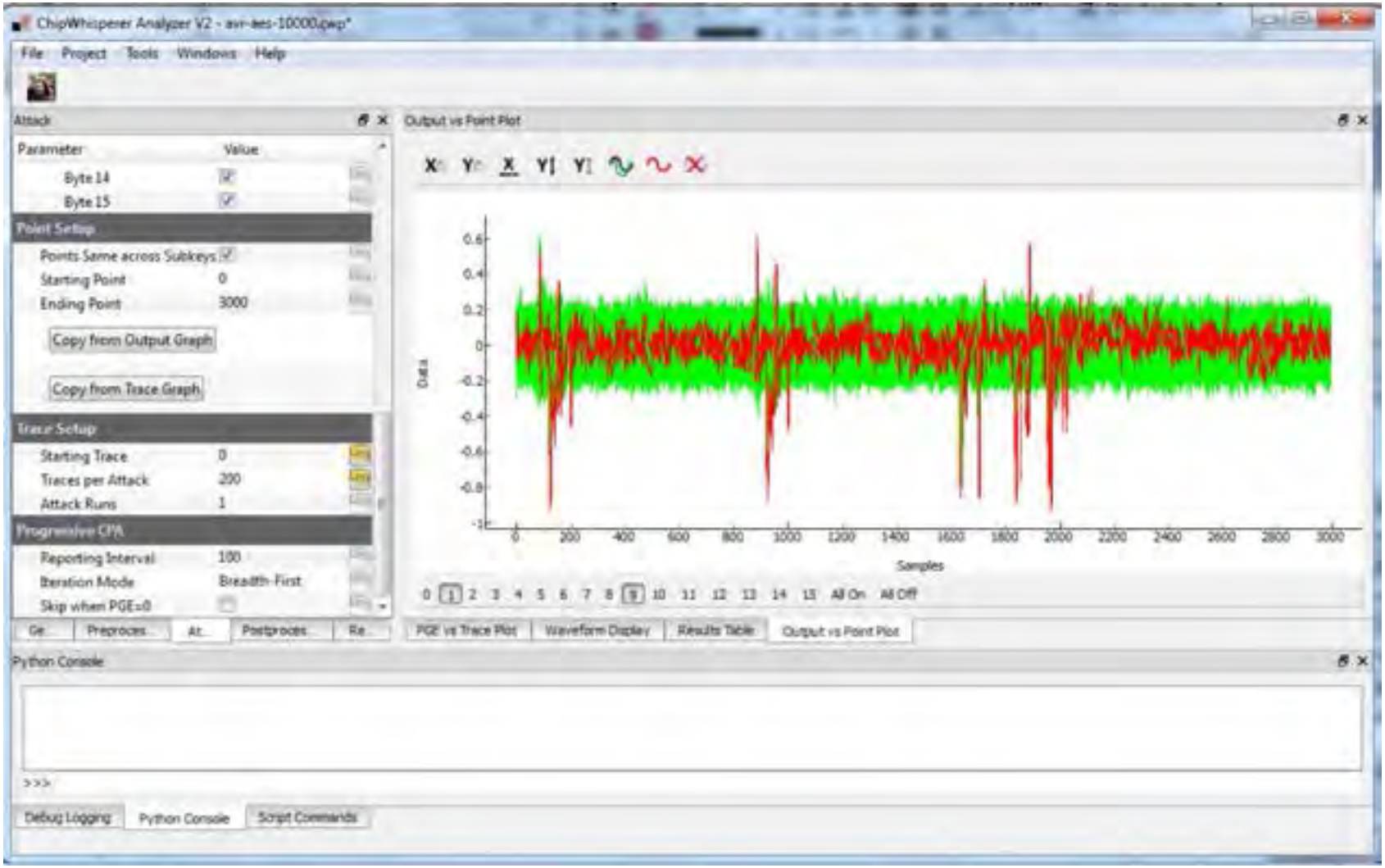
The New Thing



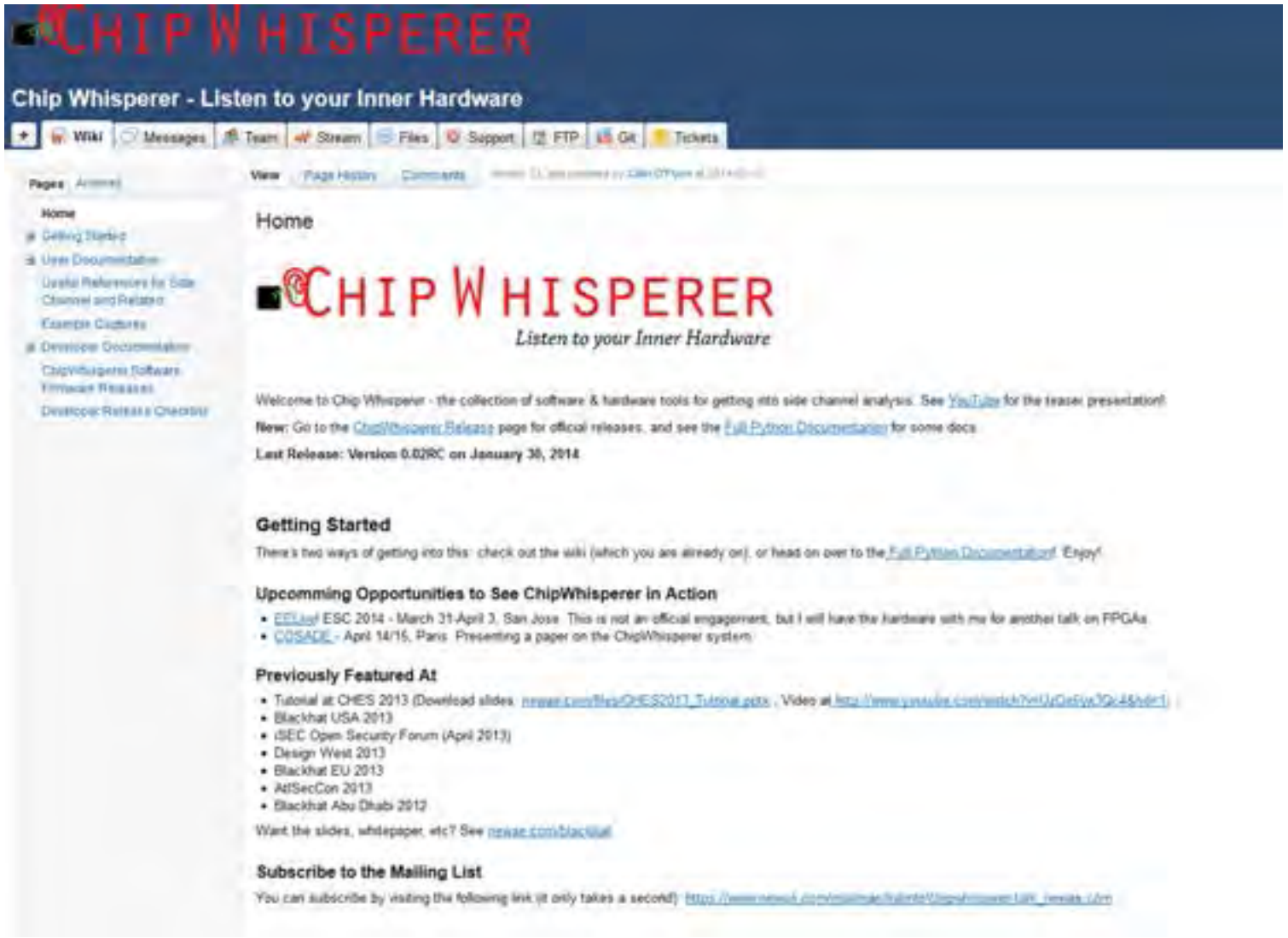
DPA SOFTWARE



DPA SOFTWARE



Documentation (!!)



Documentation (!!)

ChipWhisperer Main Documentation

CHIPWHISPERER

Listen to your Inner Hardware

Welcome to the ChipWhisperer documentation. This documentation is auto-generated from the ChipWhisperer source, and includes information

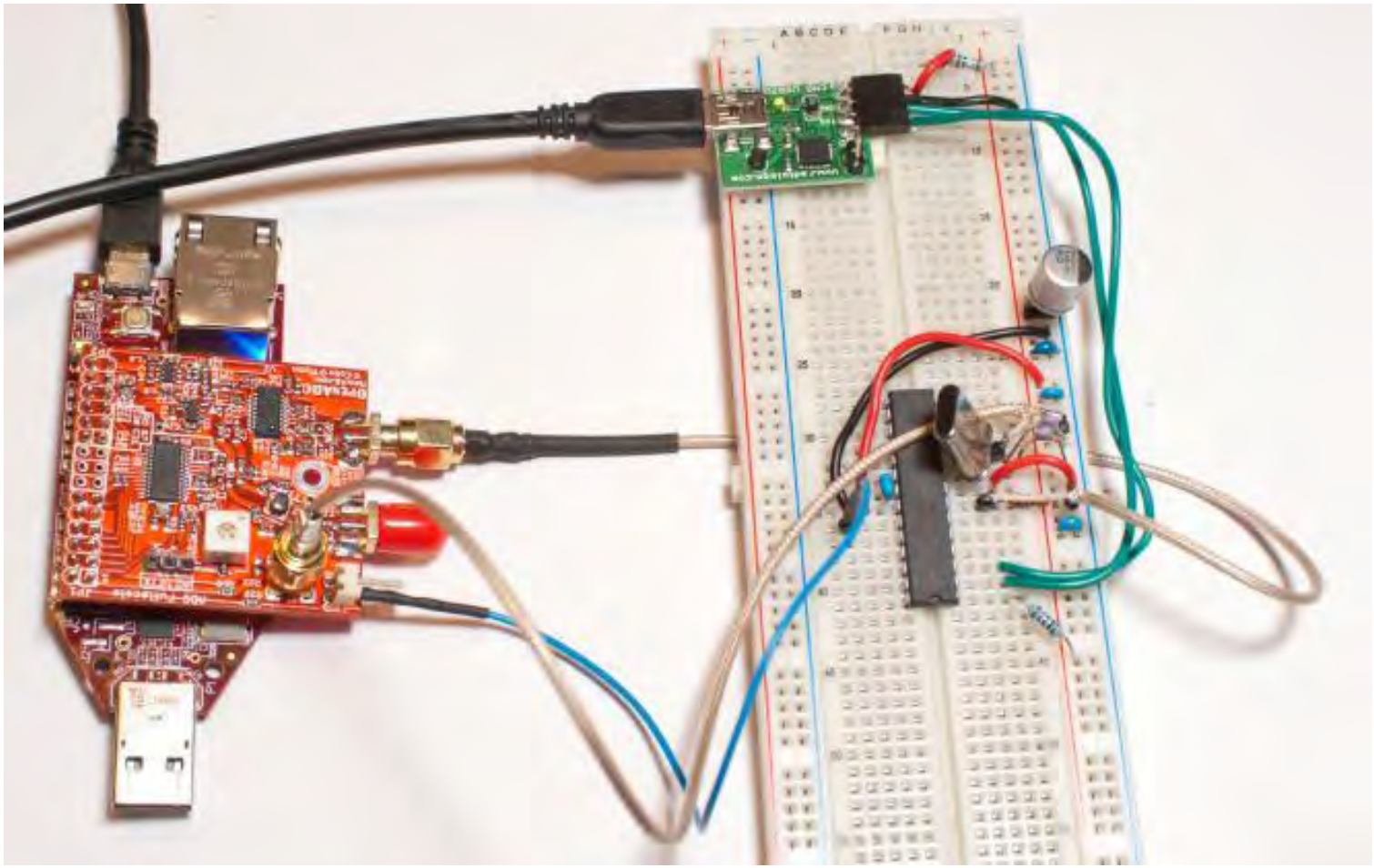
Additionally, there is a ChipWhisperer Wiki which has information such as notes on building your own hardware and various other tid-bits. Gene notes which may change frequently (e.g. building guides, latest errata, release information, latest conferences CW is being demoed at, etc).

Some of the documents here will make reference to youtube videos. You can see the NewAE Technology Inc. YouTube Channel for 'official' videos video & put some links here! There is also some videos of ChipWhisperer things in action on Colin O'Flynn's YouTube Channel.

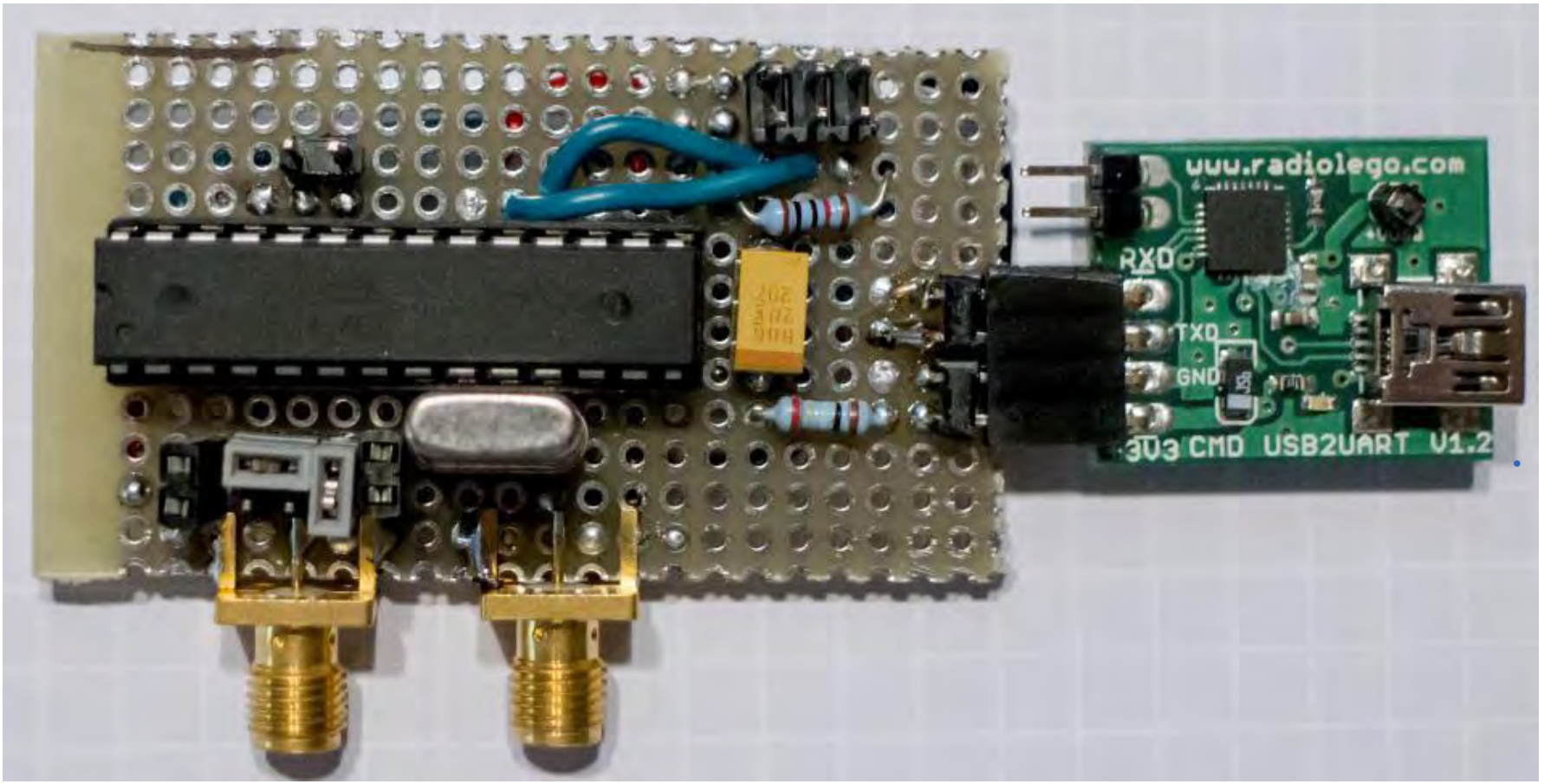
If you are interested in buying premade hardware, see NewAE Technology Inc.'s Embedded Hardware Security Division, although again because th Remember this project *does not* require specific hardware to work, and can be used with almost any oscilloscope along with a microcontroller ta

Full Table of Contents:

Nuttin' Fancy Version



Another DIY Example



OR WITH A SCOPE

What Now?

Call Me Maybe

Find those tools:
www.ChipWhisperer.com

Find me:
www.OFlynn.com

Buy Stuff:
www.newae.com

Twitter: @colinoflynn
Email: coflynn@newae.com