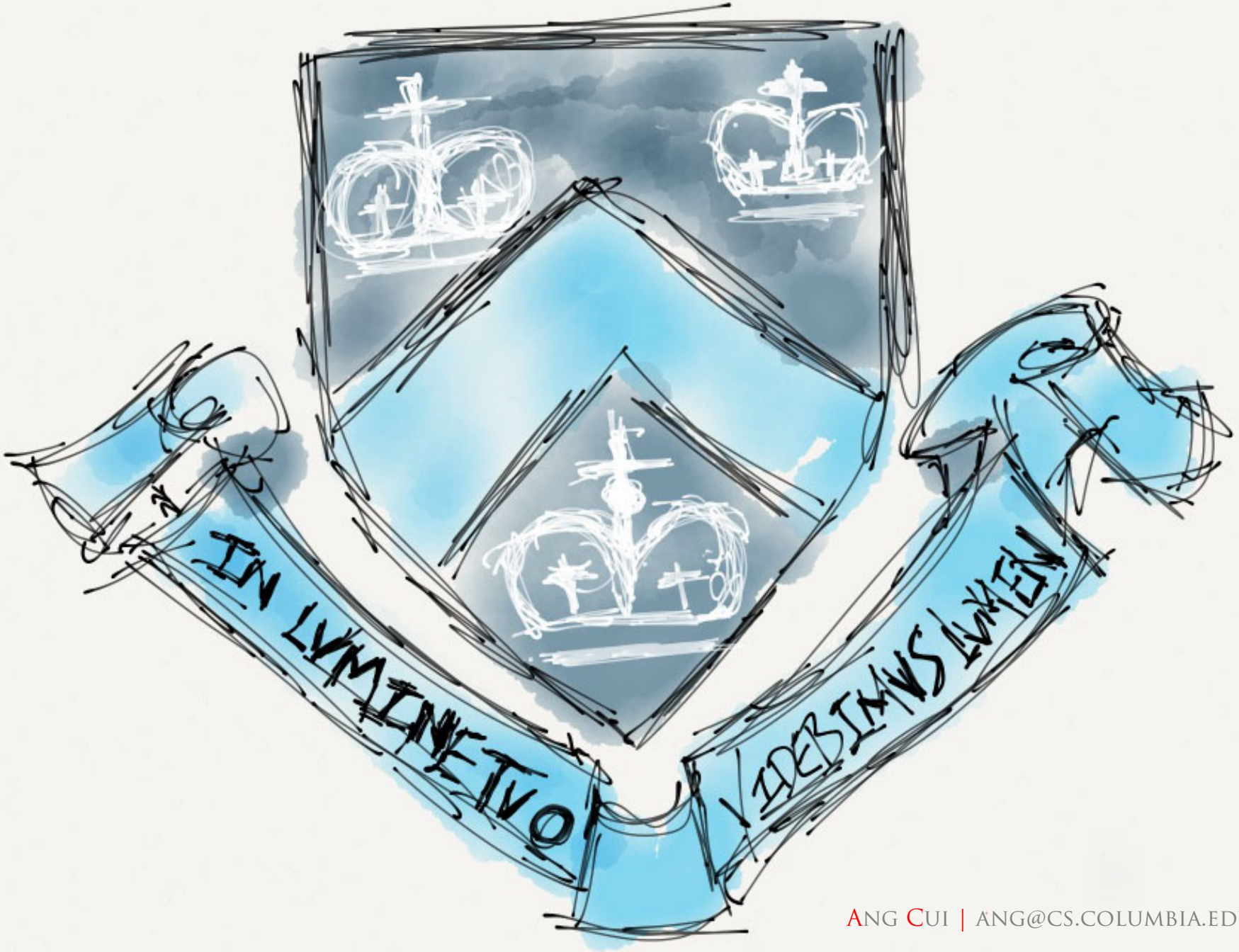
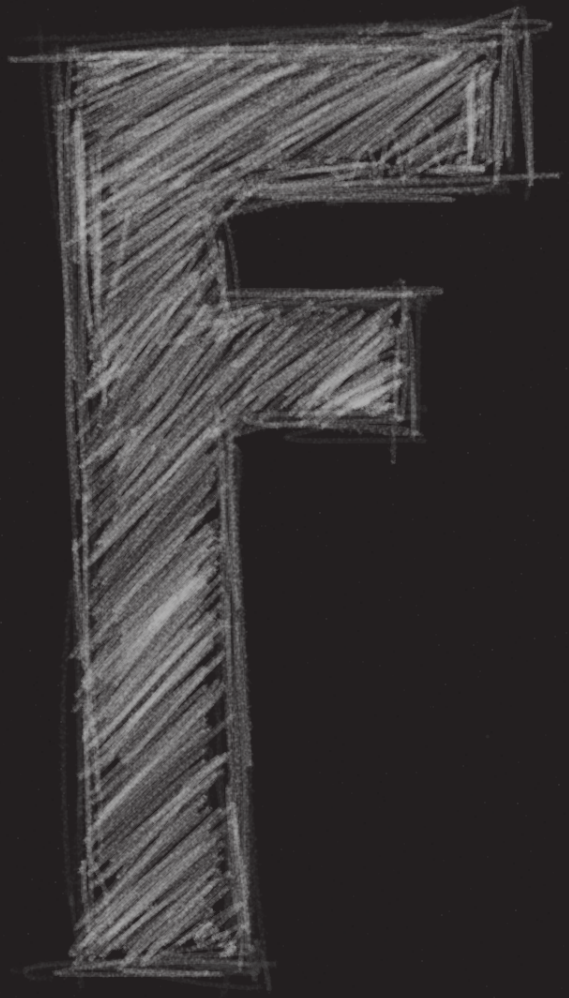


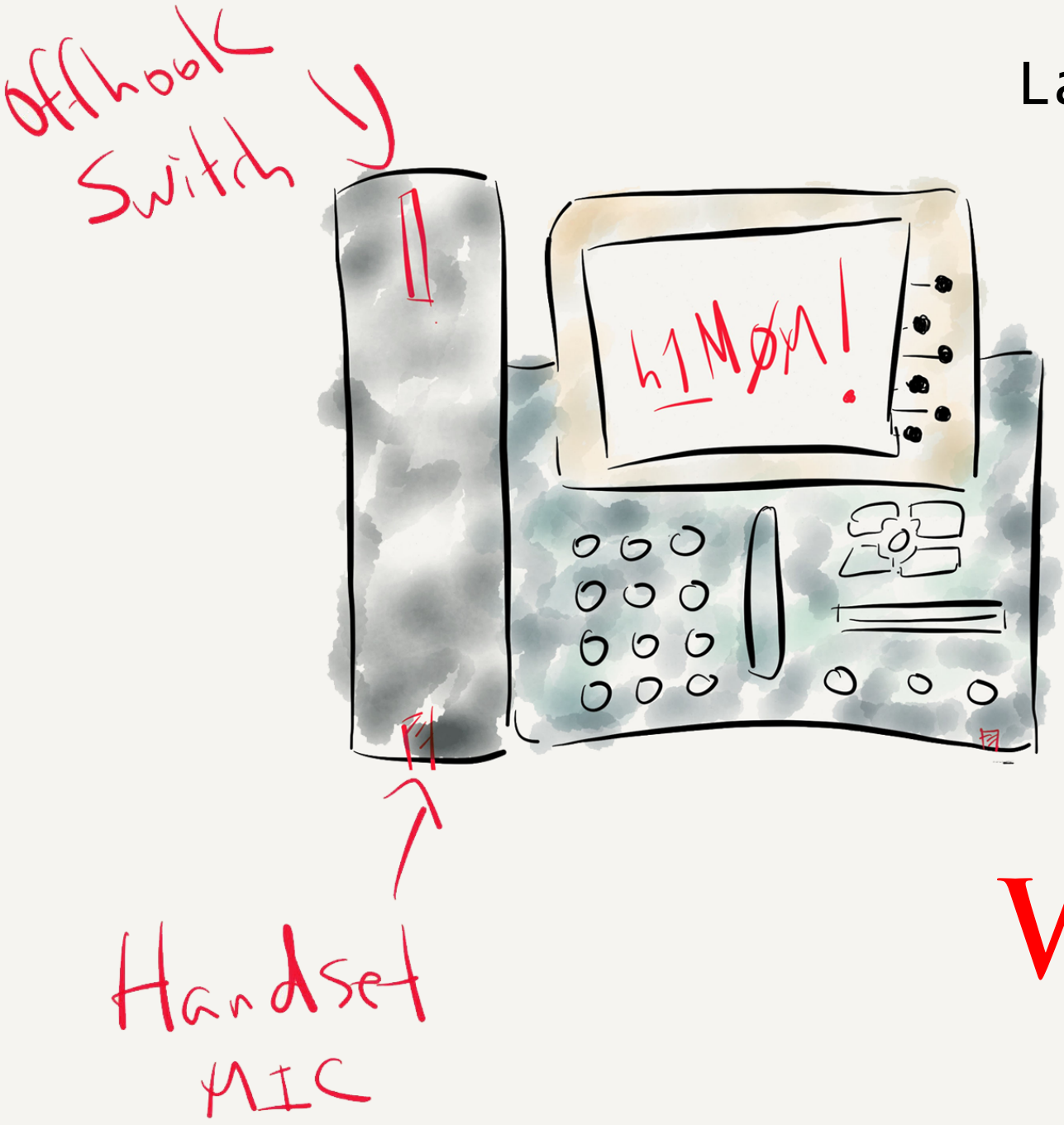
ANG CUI | ANG@CS.COLUMBIA.EDU



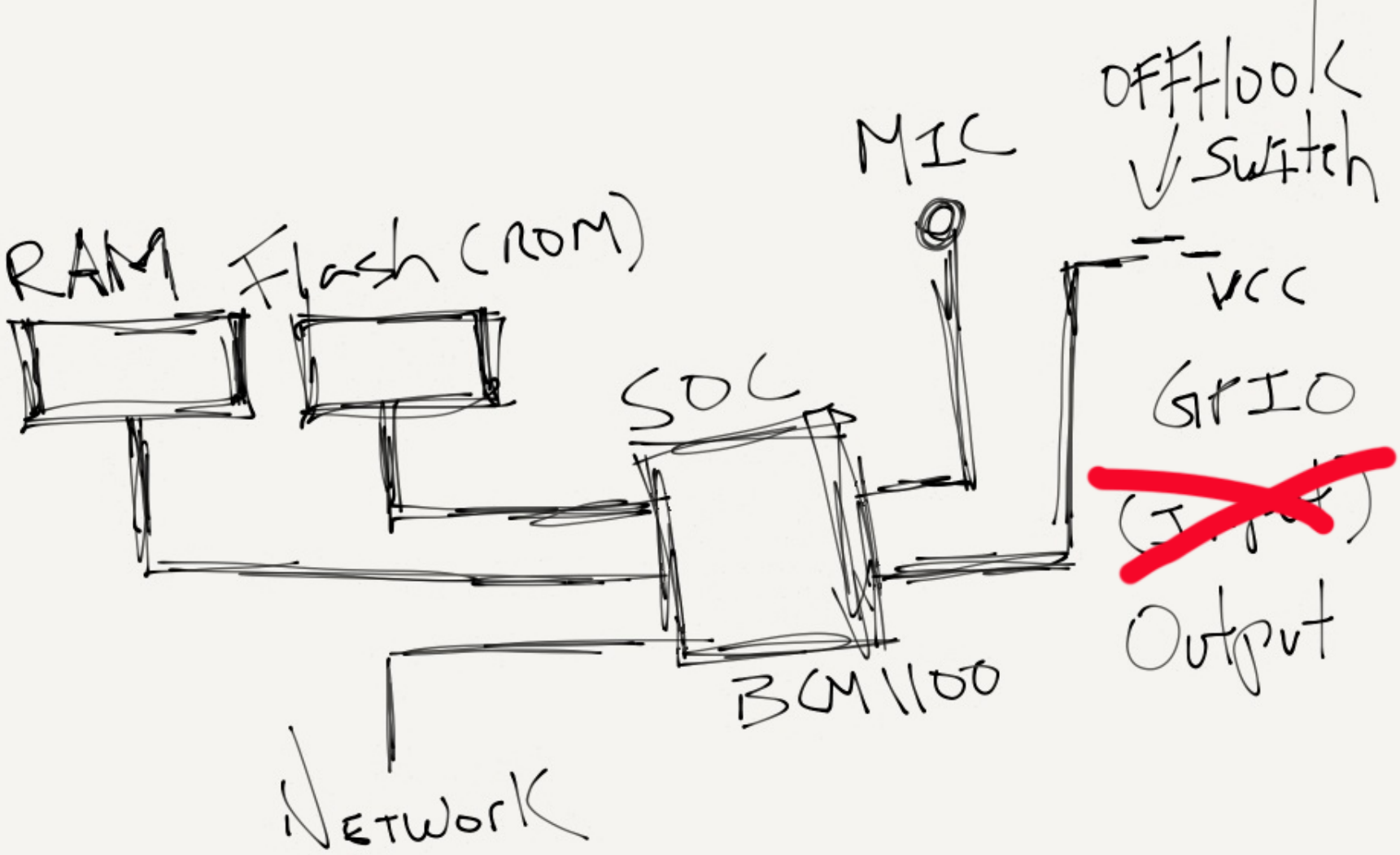
irmware Fat Camp!

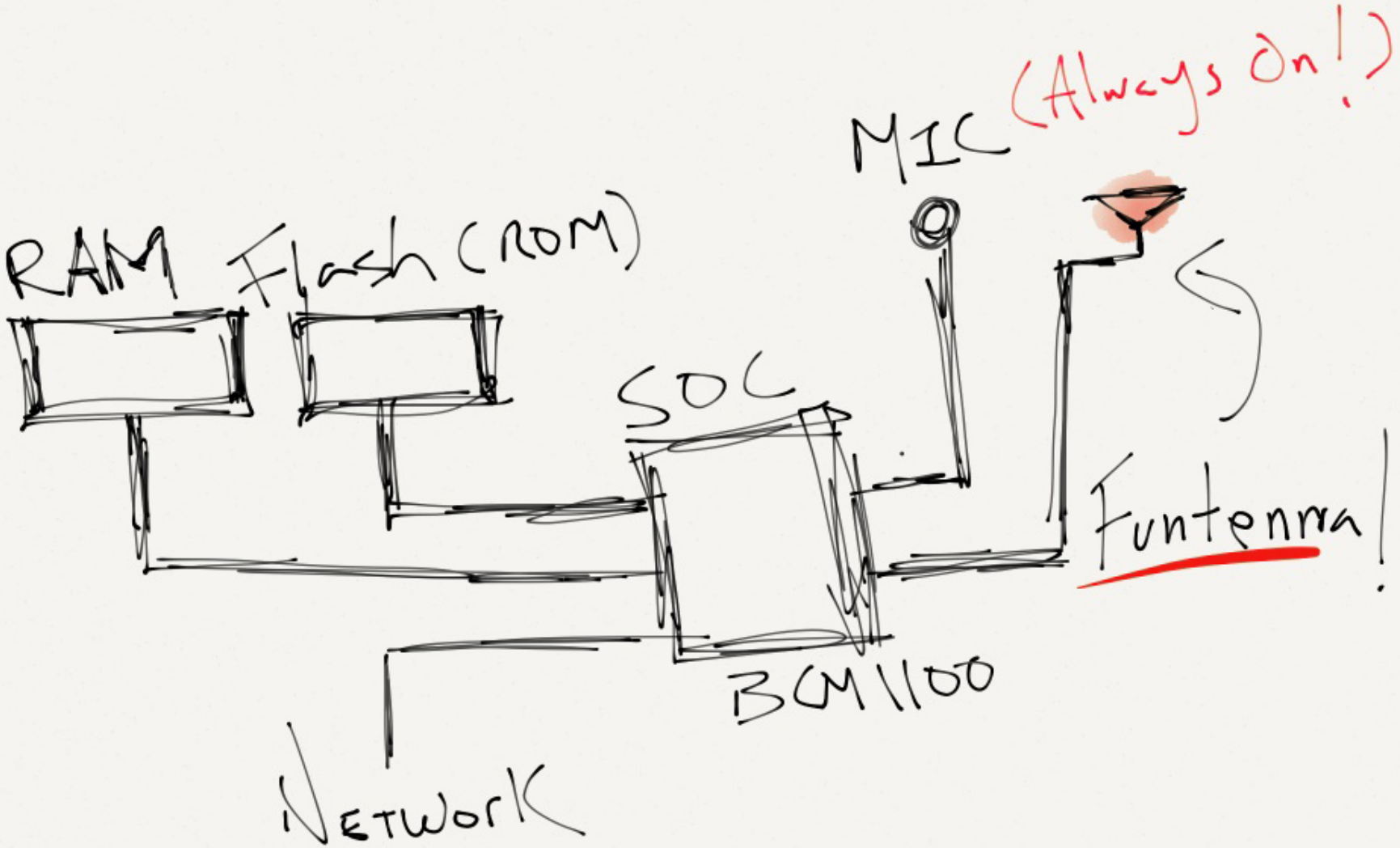
With Zach Newman
zjn2101@columbia.edu

Last year…



WIN!


RAM
Flash (ROM)
SOC
MIC
OFFHOOK
Switch
VCC
GPIO
Output
BCM1100
NETWORK


MIC (Always On!)
RAM
Flash (ROM)
SOC
Funtenna!
BCM1100
Network


怪しい伝説
Discovery
CHANNEL
h1M@N!

Funtenna

Noun:

1: Software payload that intentionally causes its host hardware to act as an improvised RF transmitter using existing hardware, which are typically not designed for electromagnetic emanation.

2: Software which intentionally causes compromising emanation.

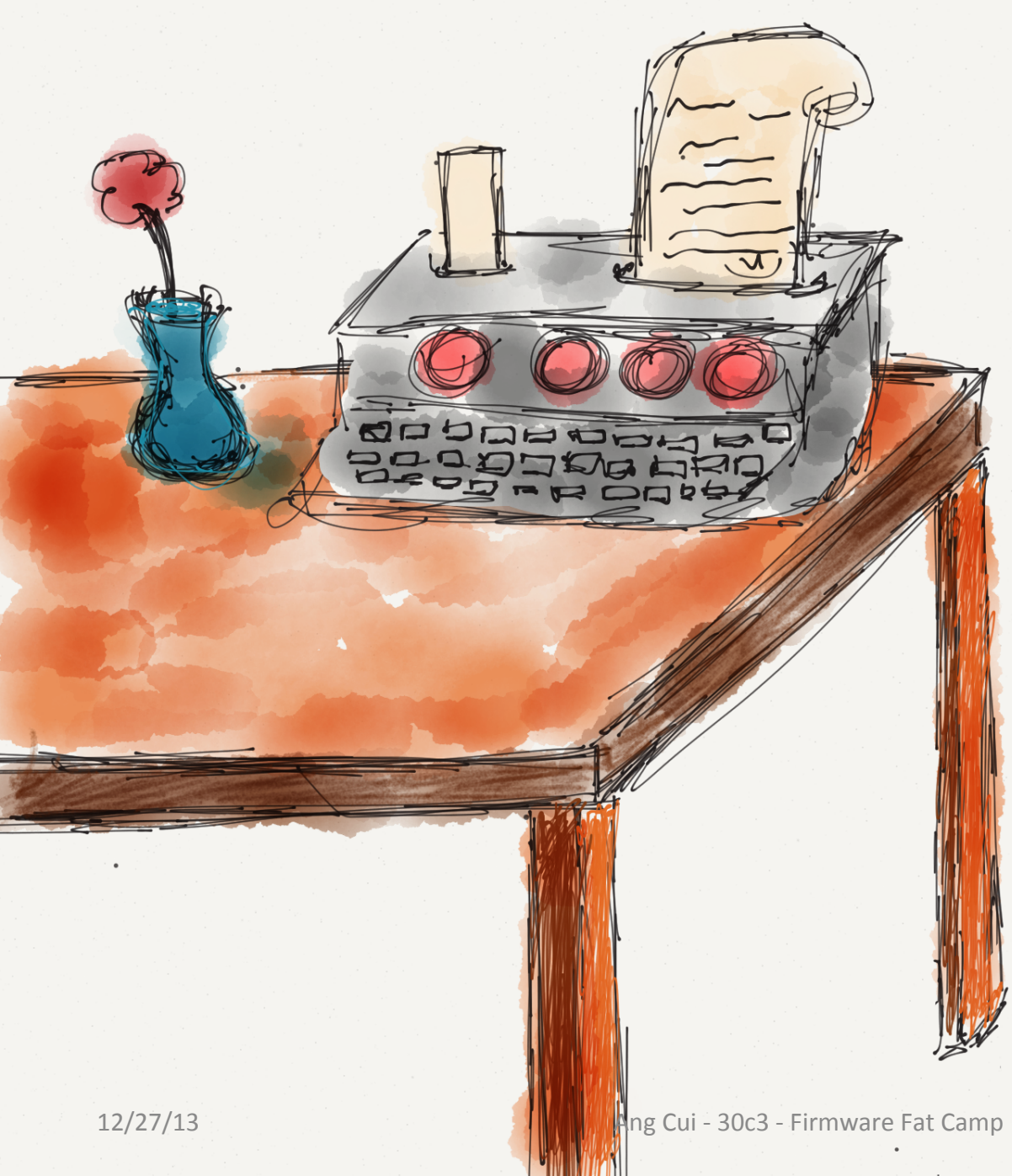
Prior Art

- SigInt Awesomesauce, et. al
  - TEMPEST Timeline: http://cryptome.org/tempest-time.htm

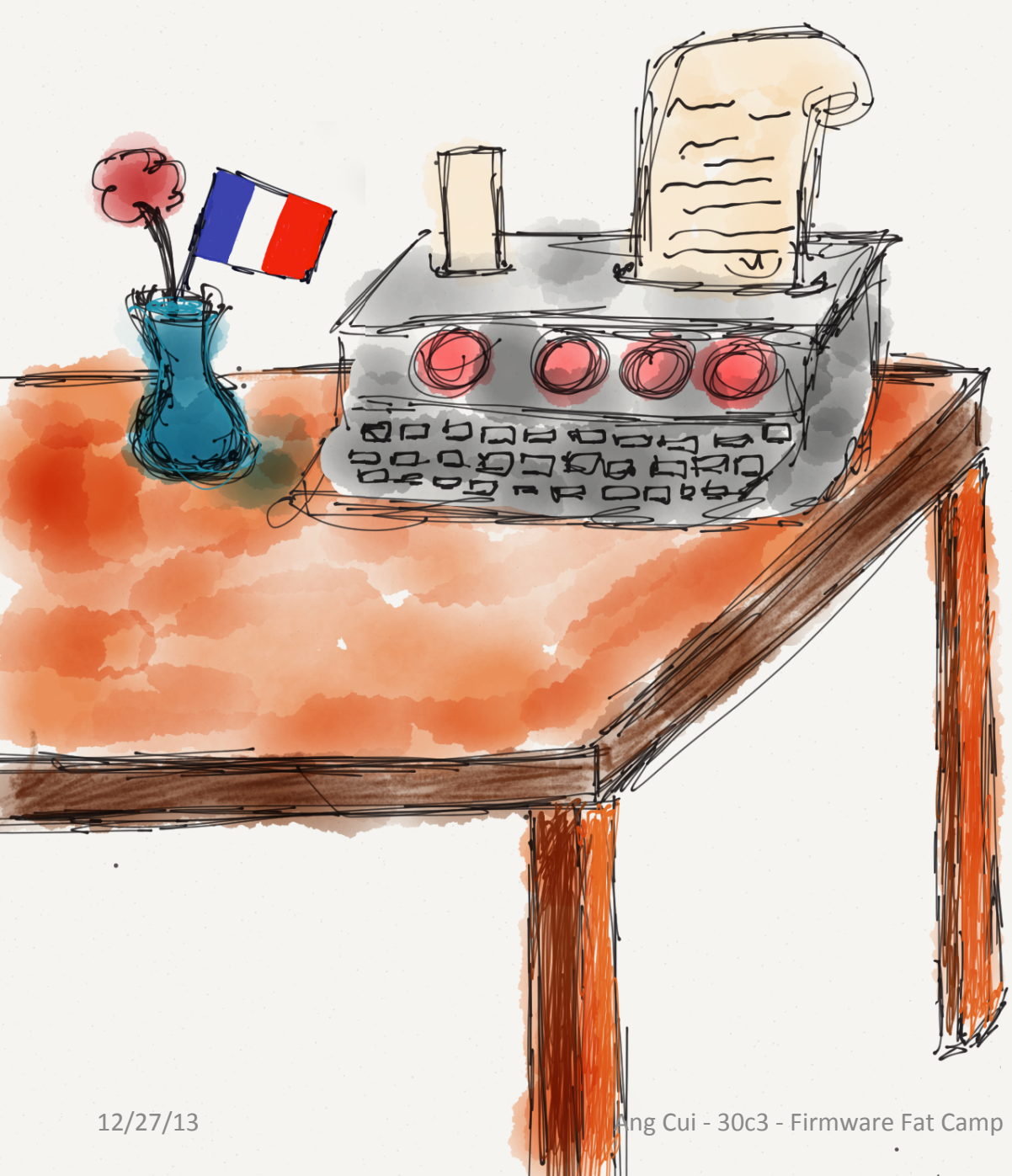
Peter Wright

MI5 Counter Intelligence
Author: Spycatcher

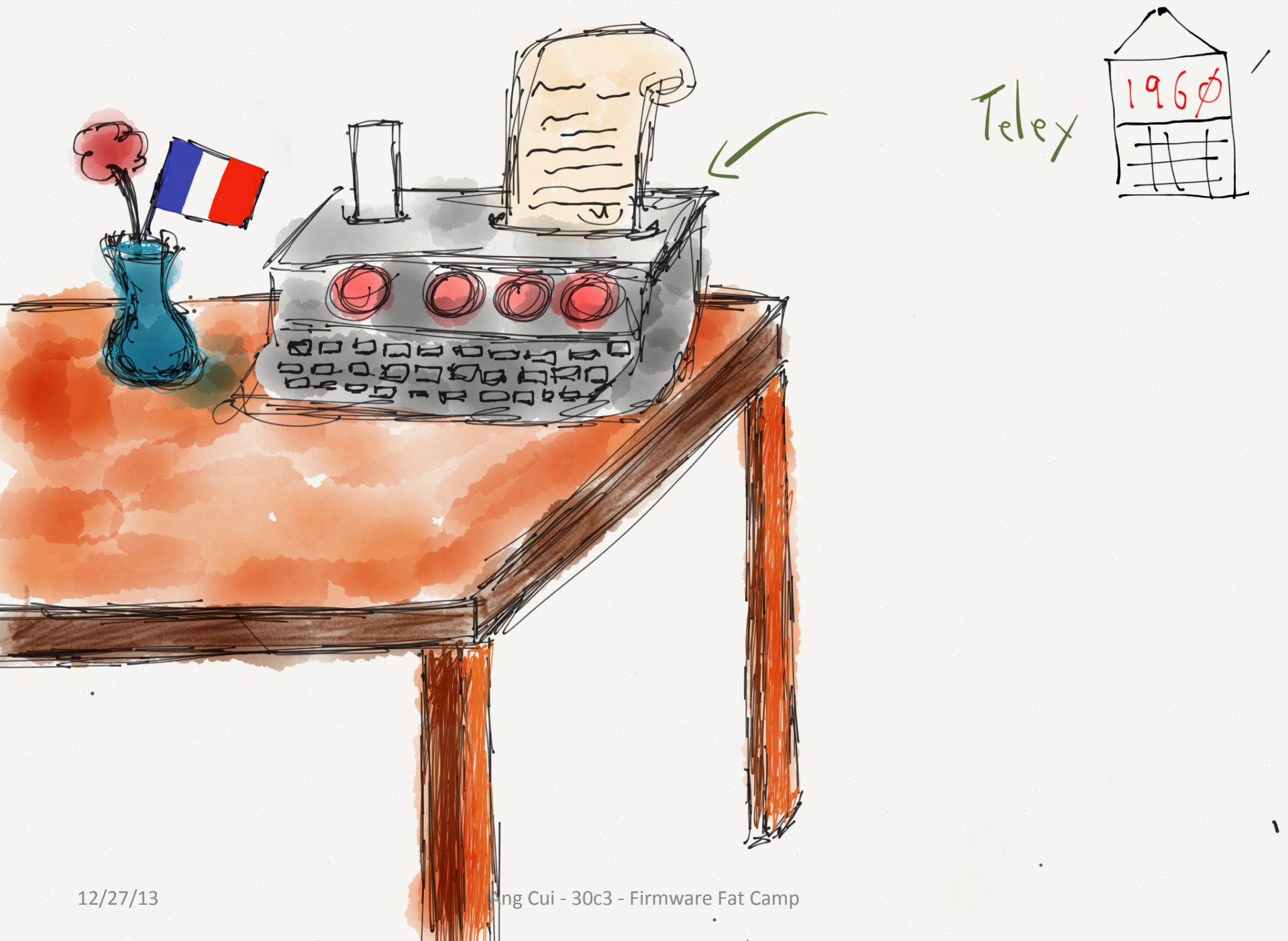

1960

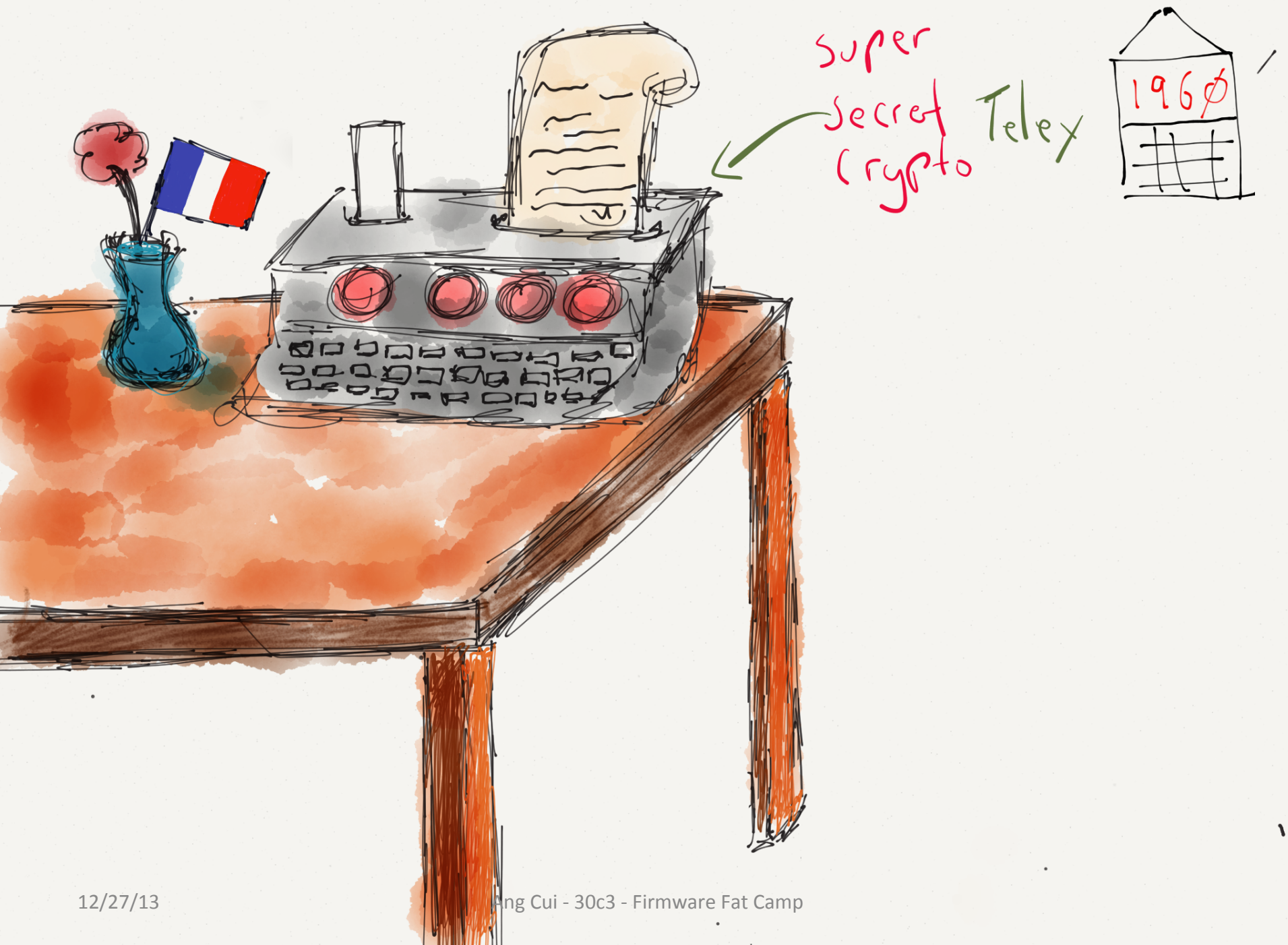


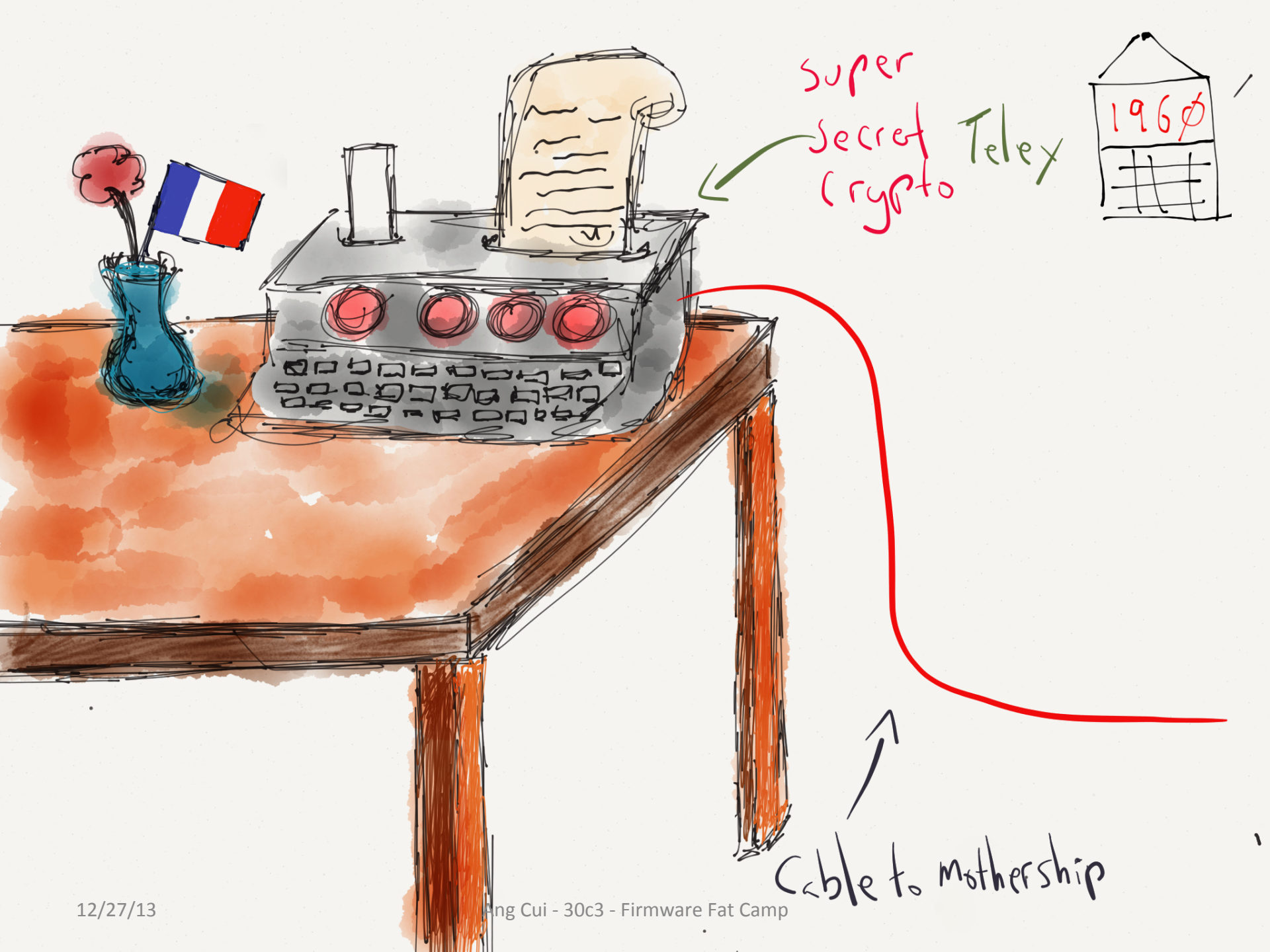

1960

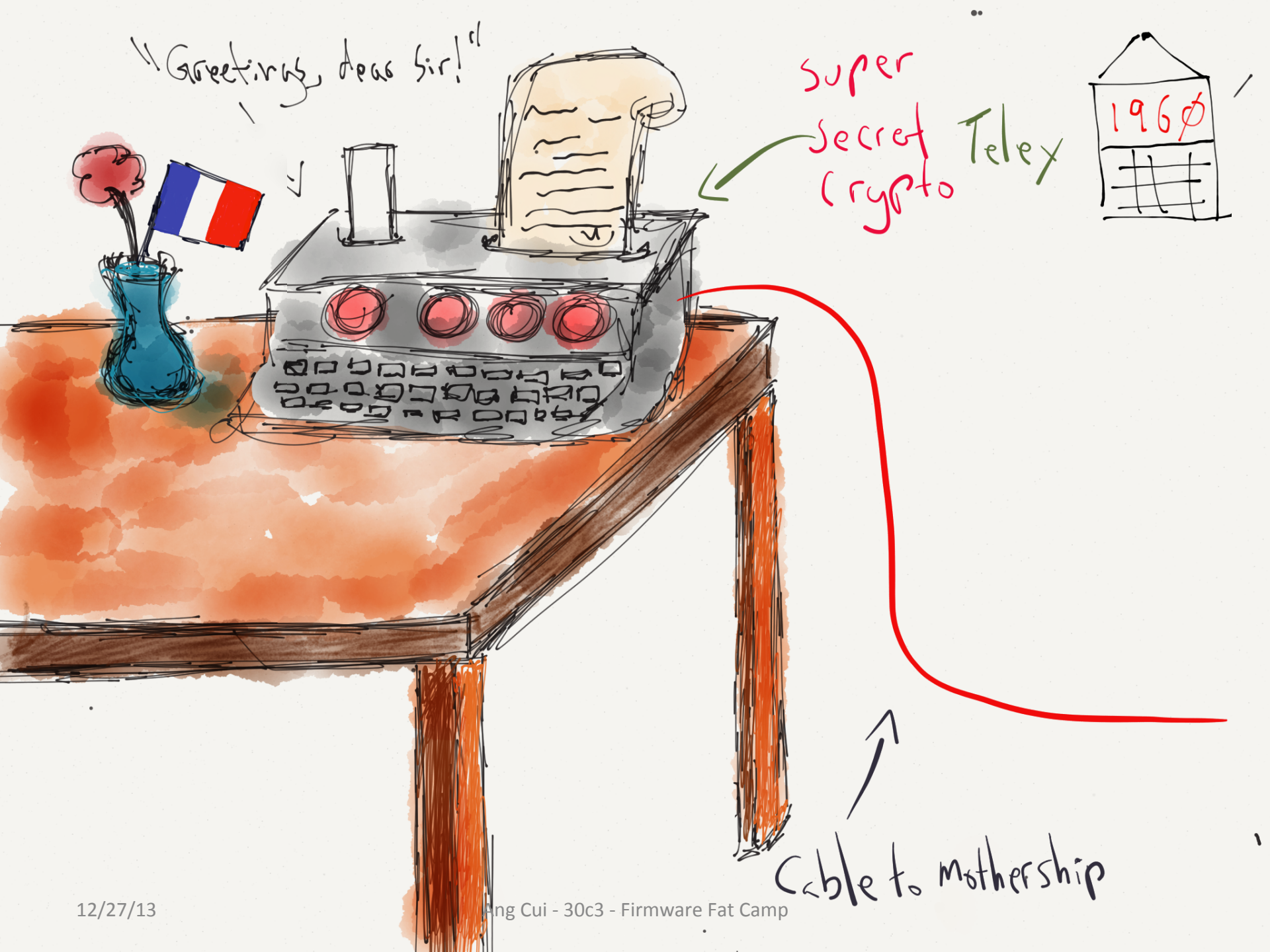


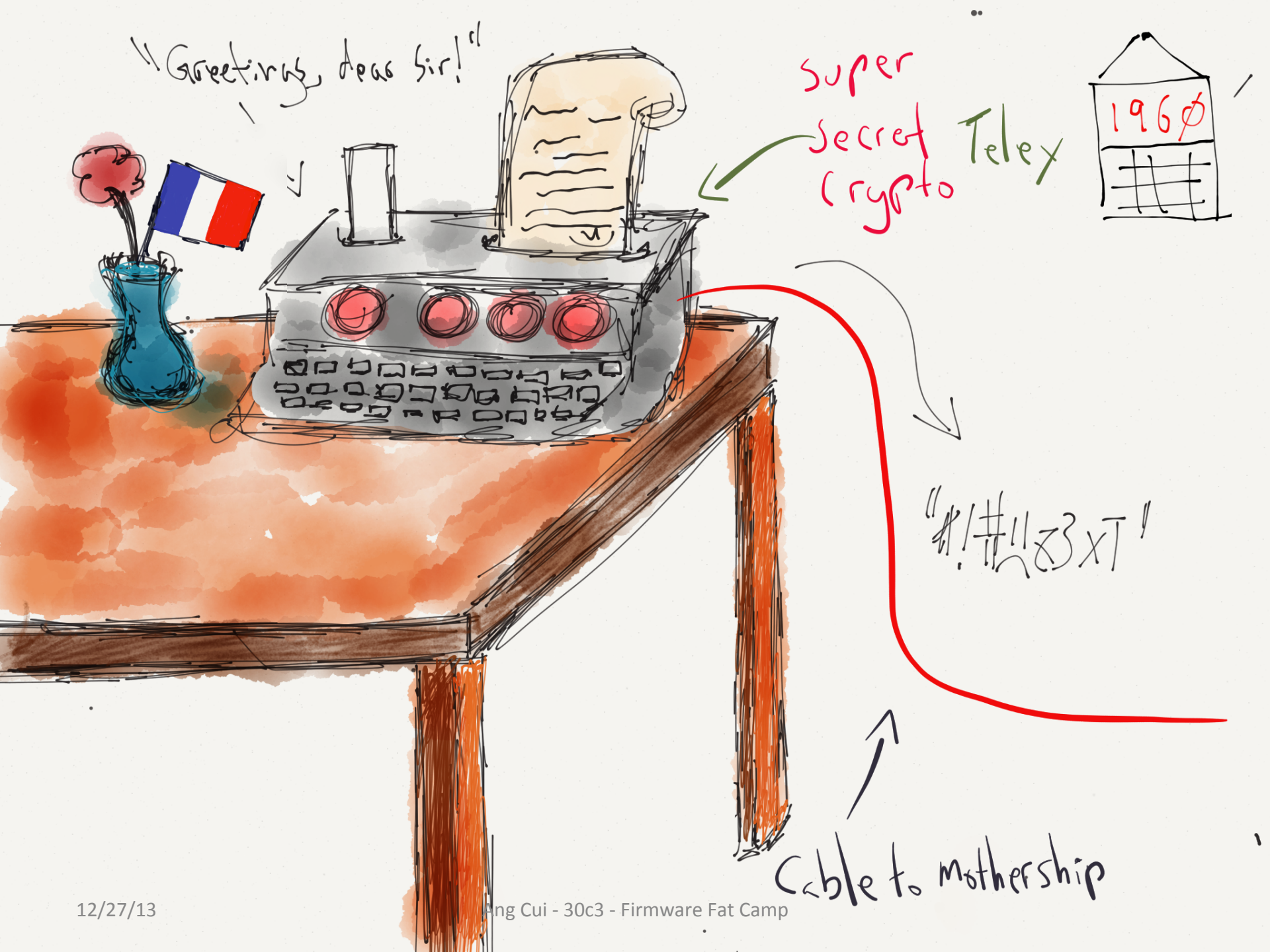

1960

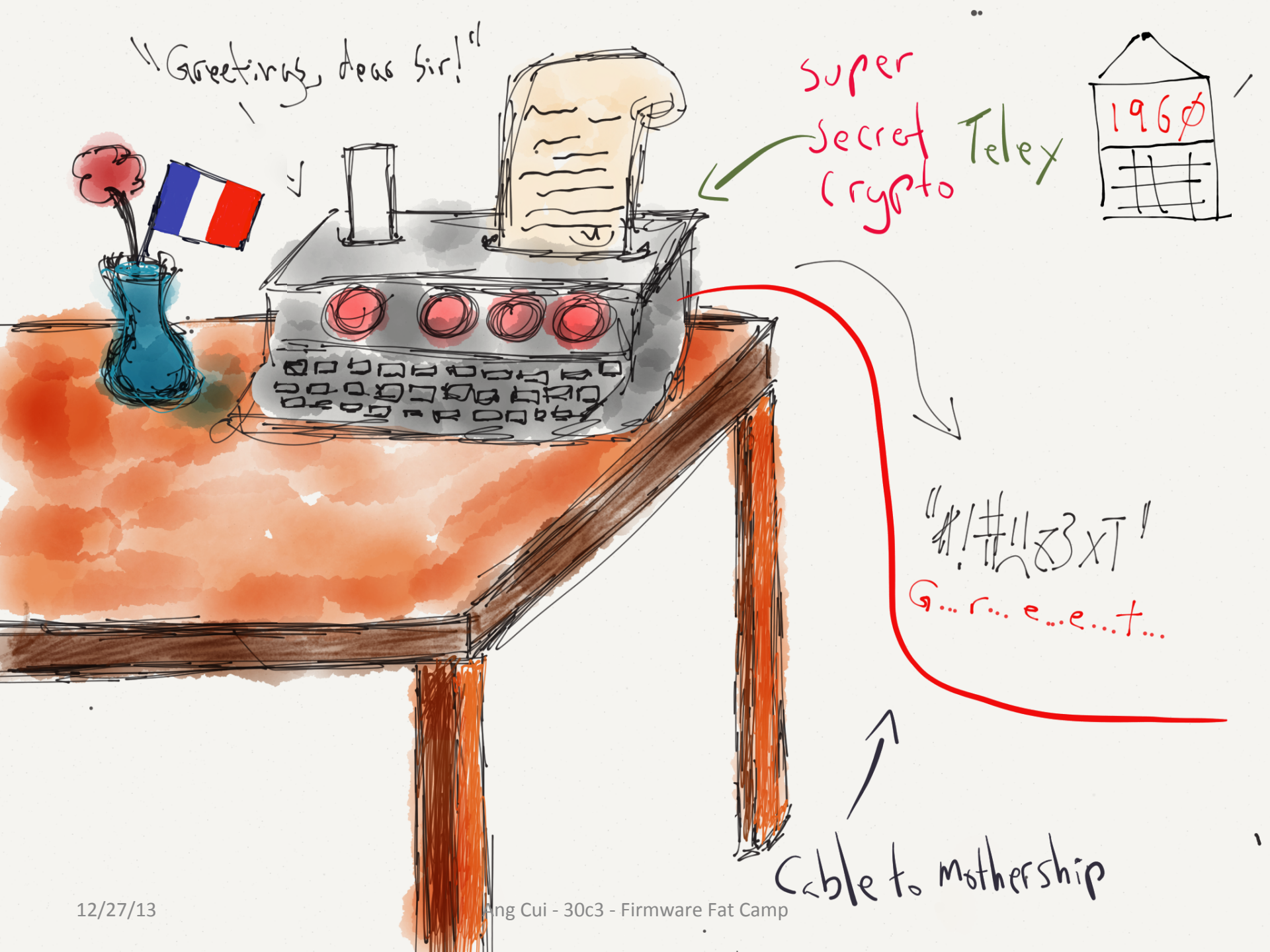

Telex
1960


super
secret crypto Telex
1960


super
secret crypto
Telex
1960
Cable to mothership


"Greetings, dear Sir!"
super
secret
crypto
Telex
1960
Cable to mothership


"Greetings, dear Sir!"
super
secret crypto
Telex
1960
"#!#!!z3xT"
Cable to mothership


"Greetings, dear Sir!"
super
secret crypto Telex
1960
"#!#!!&3xT!"
G...r...e...e...t...
Cable to mothership

Prior Art

- SigInt Awesomesauce, et. al
- Van Eck Phreaking, etc

Prior Art

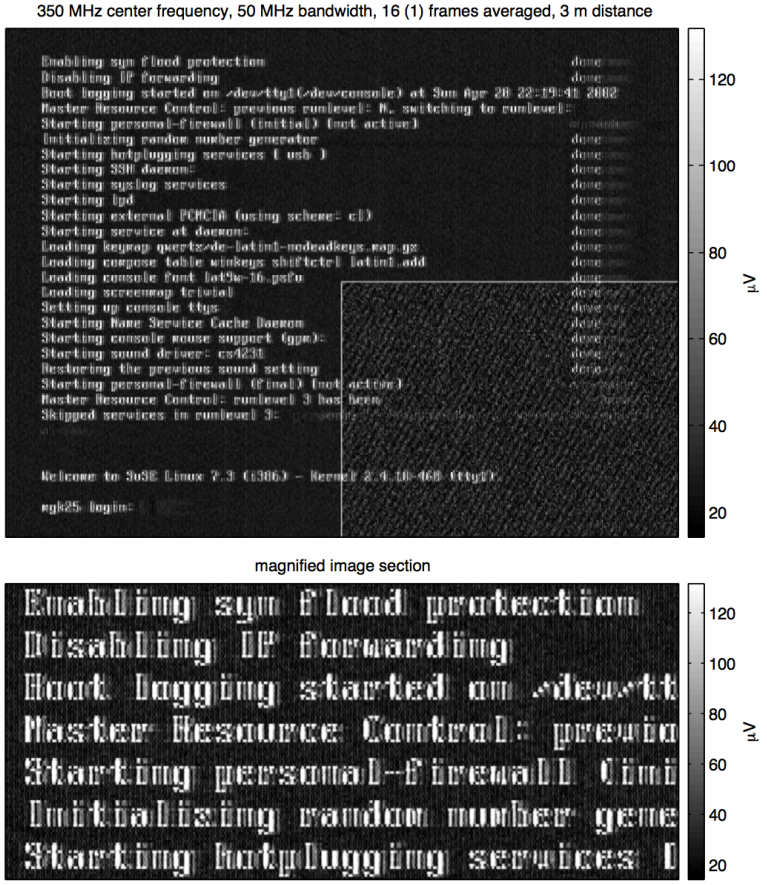
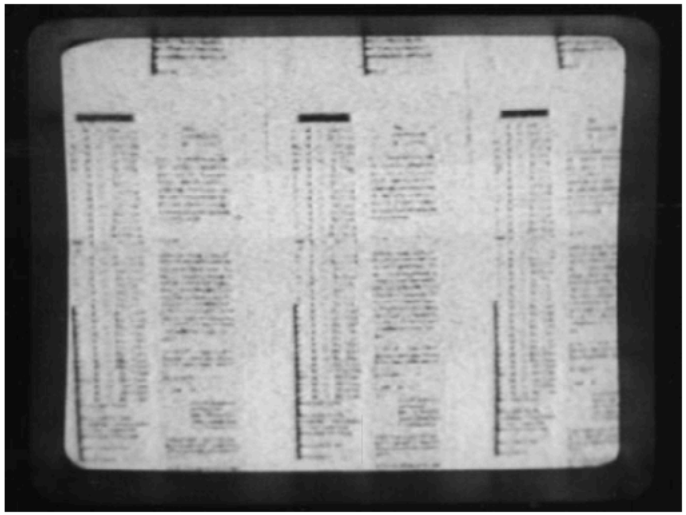


Fig. 1. Eavesdropped Linux boot screen visible on the LCD of a Toshiba 440CDX laptop (log-periodic antenna, vertical polarization).



Electromagnetic Eavesdropping Risks of Flat-Panel Displays

Markus G. Kuhn

University of Cambridge, Computer Laboratory,
15 JJ Thomson Avenue, Cambridge CB3 0FD, United Kingdom
http://www.cl.cam.ac.uk/~mgk25/

Prior Art

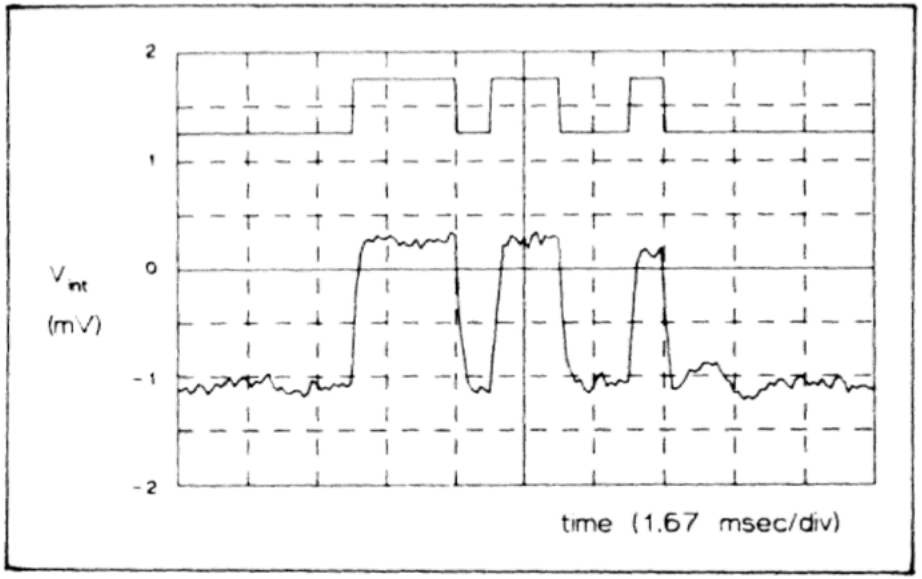


Fig. 4. Original and intercepted data signal at 7 m and 98 MHz (FM band).

The Threat of Information Theft by Reception of Electromagnetic Radiation from RS-232 Cables

Peter Smulders

Prior Art

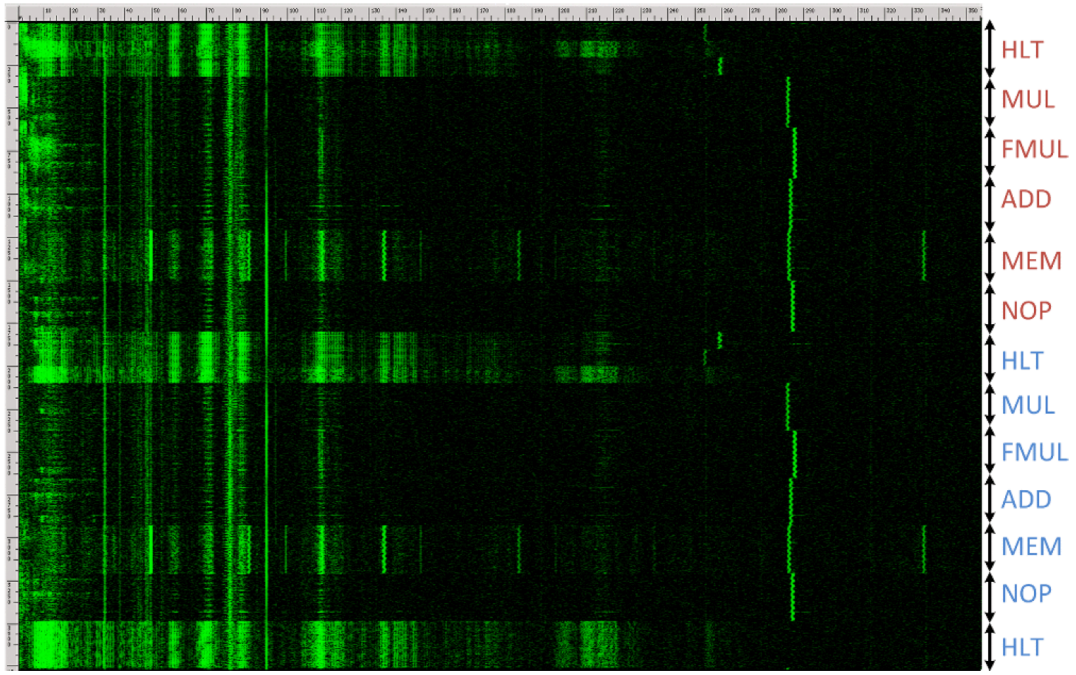


Figure 7: Acoustic measurement frequency spectrogram of a recording of different CPU operations using the Brüel&Kjær 4939 microphone capsule. The horizontal axis is frequency (0–310 kHz), the vertical axis is time (3.7 sec), and intensity is proportional to the instantaneous energy in that frequency band.

RSA Key Extraction via Low-Bandwidth Acoustic Cryptanalysis*

Daniel Genkin
Technion and Tel Aviv University
danielg3@cs.technion.ac.il

Adi Shamir
Weizmann Institute of Science
adi.shamir@weizmann.ac.il

Eran Tromer
Tel Aviv University
tromer@cs.tau.ac.il

Funtenna p0c

- Reference implementation to demonstrate feasibility

Funtenna p0c

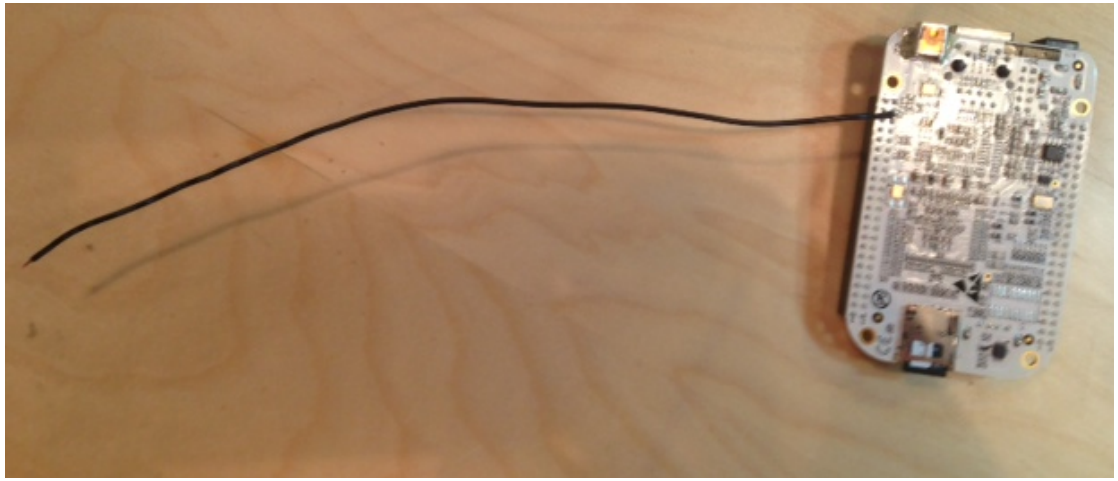
- Reference implementation to demonstrate feasibility
- Open hardware & software

- Use hardware found on nearly all embedded devices

Funtenna p0c

- Reference implementation to demonstrate feasibility
- Open hardware & software

- Use hardware found on nearly all embedded devices



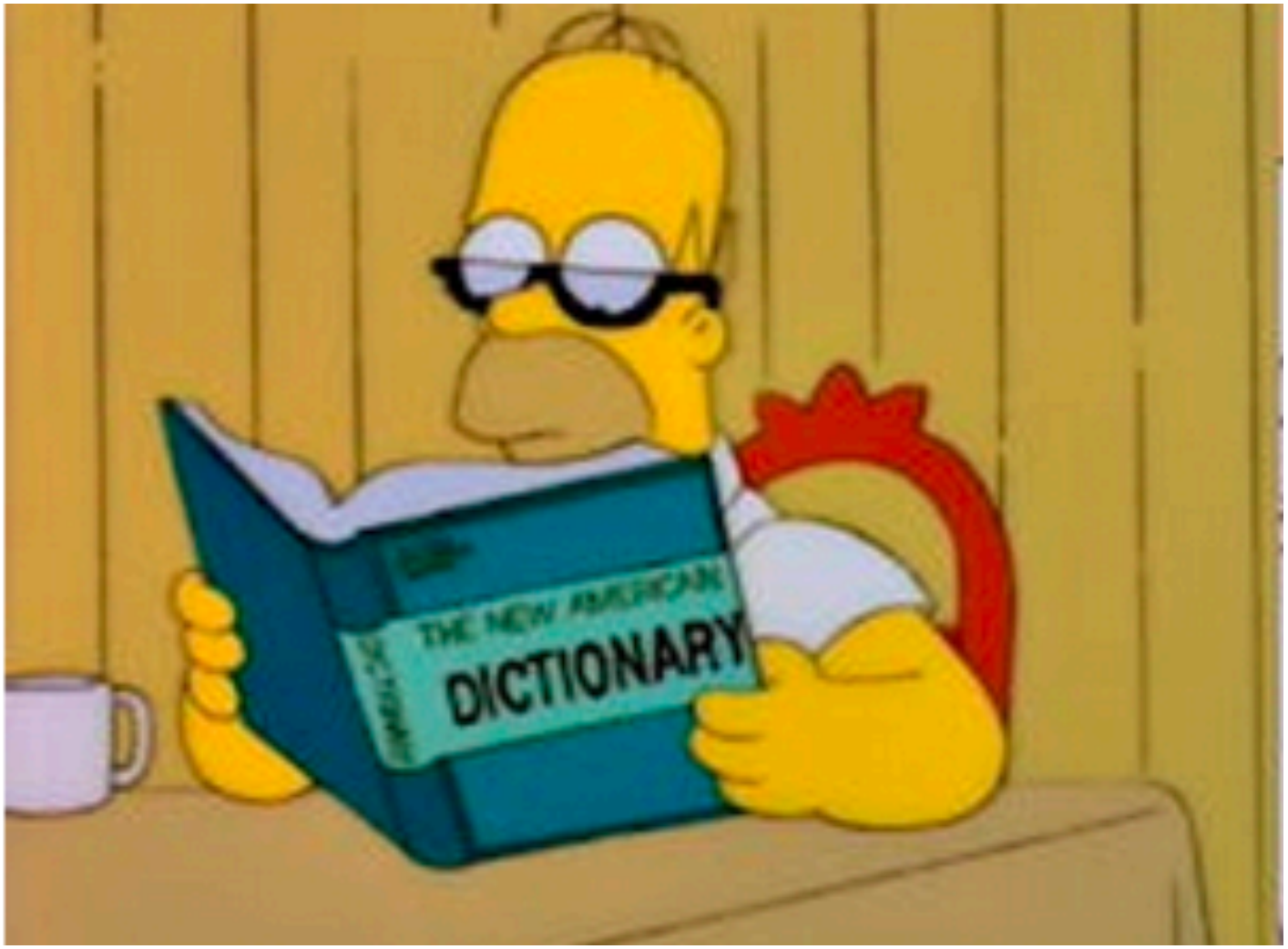
BeagleBone - GPIO

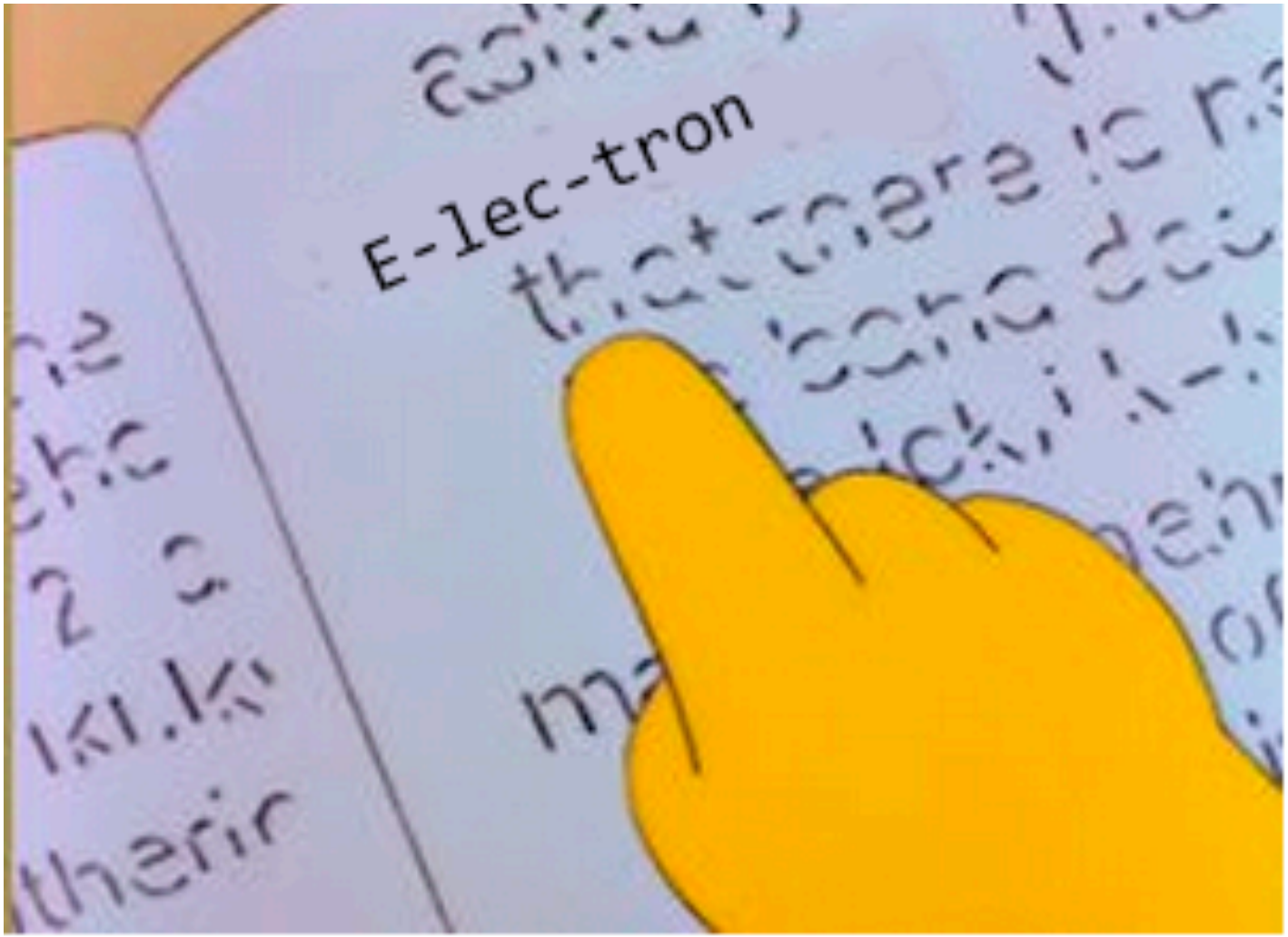

SDR


Advanced
SDR

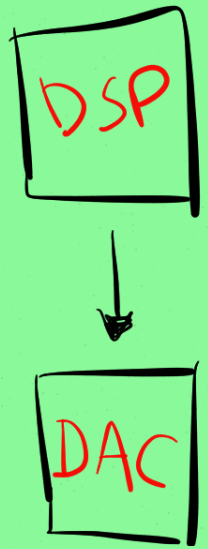

RADIO


Beginning
RADIO
VIA 9GAG.COM

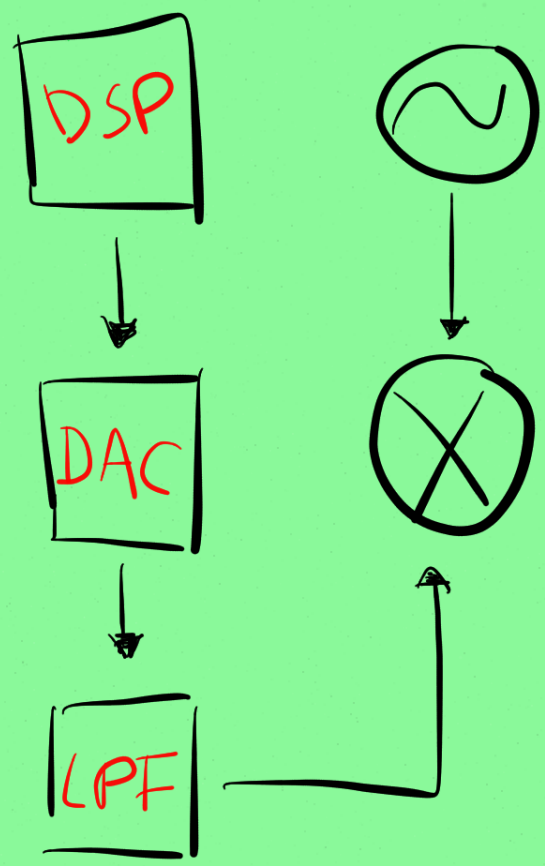

DICTIONARY

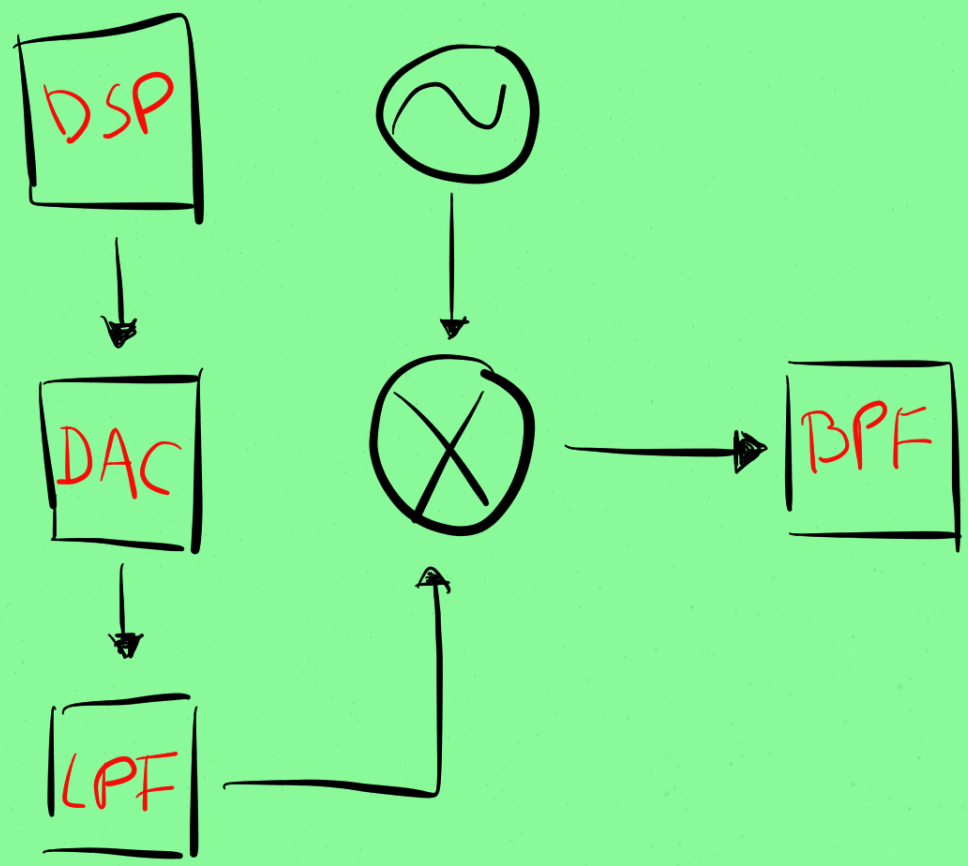

E-lec-tron

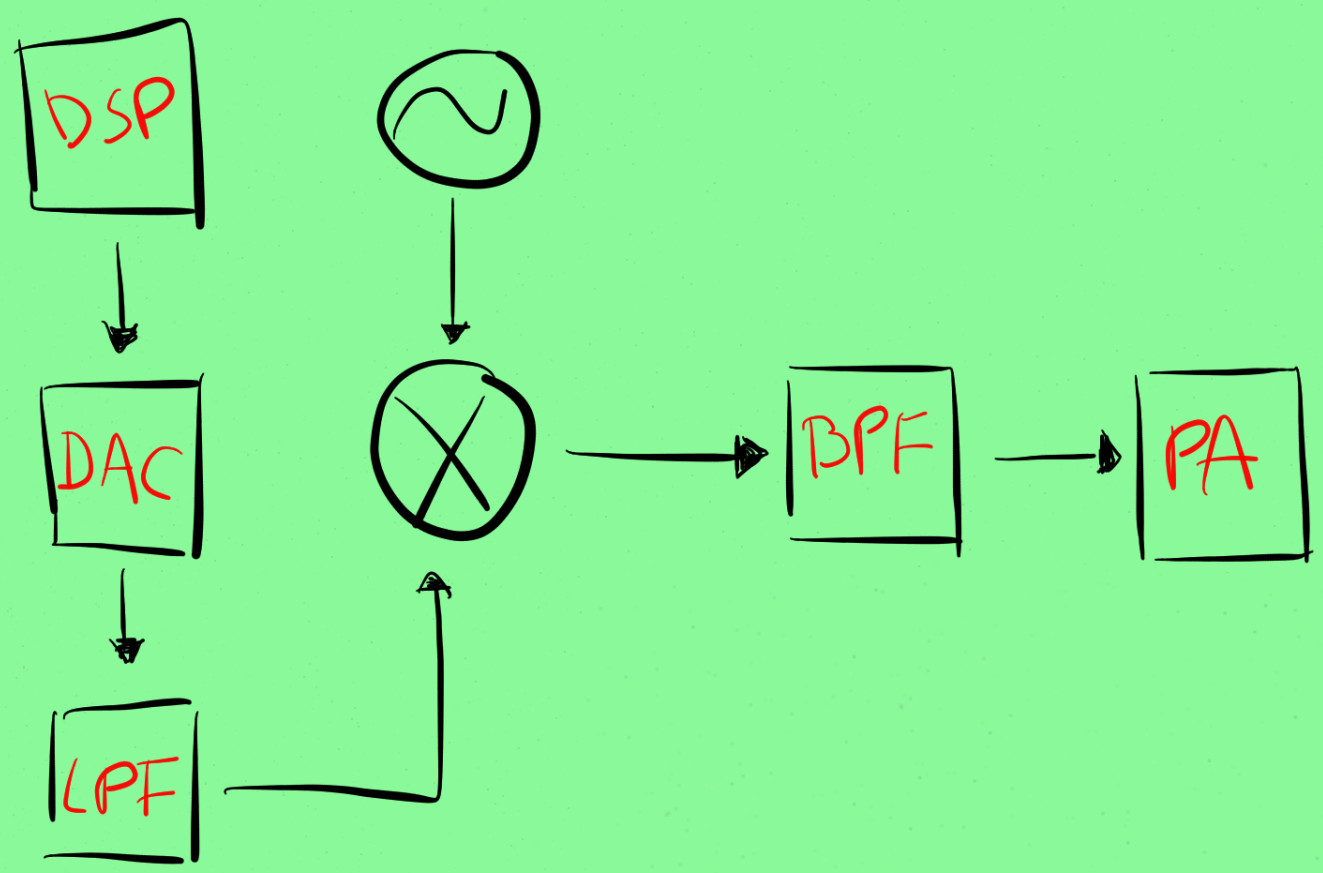

DSP

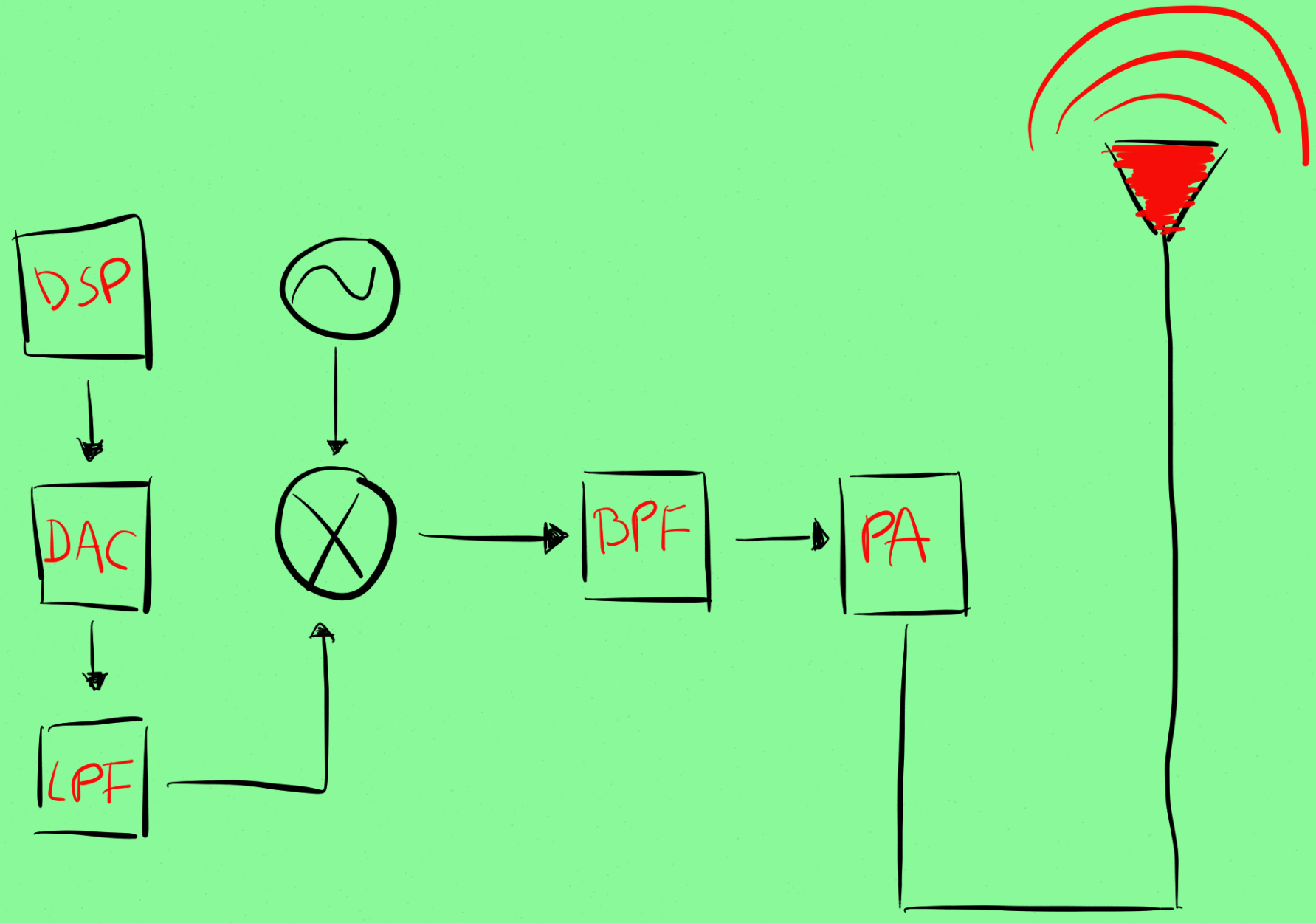

DSP
DAC


DSP
DAC
LPF


DSP
DAC
LPF


DSP
DAC
LPF
BPF


DSP
DAC
LPF
BPF
PA


DSP
DAC
LPF
BPF
PA

DAC?

DAC?

Wire…

LPF?

LPF?

Wire…

Oscillator?

Oscillator?

sorta…

Power Amp?

Power Amp?

Wire…

BPF?

BPF?

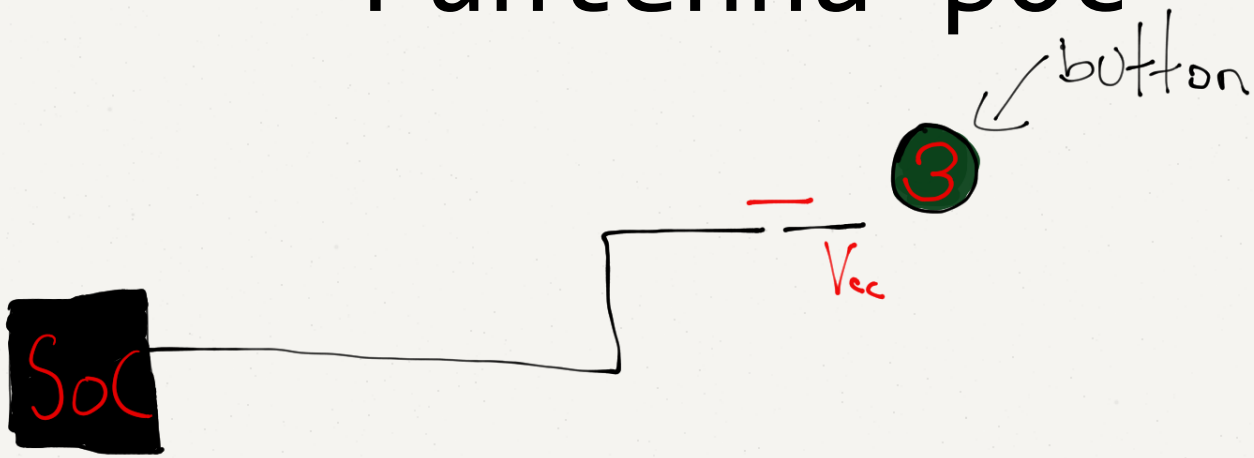
Wire…

Antenna?

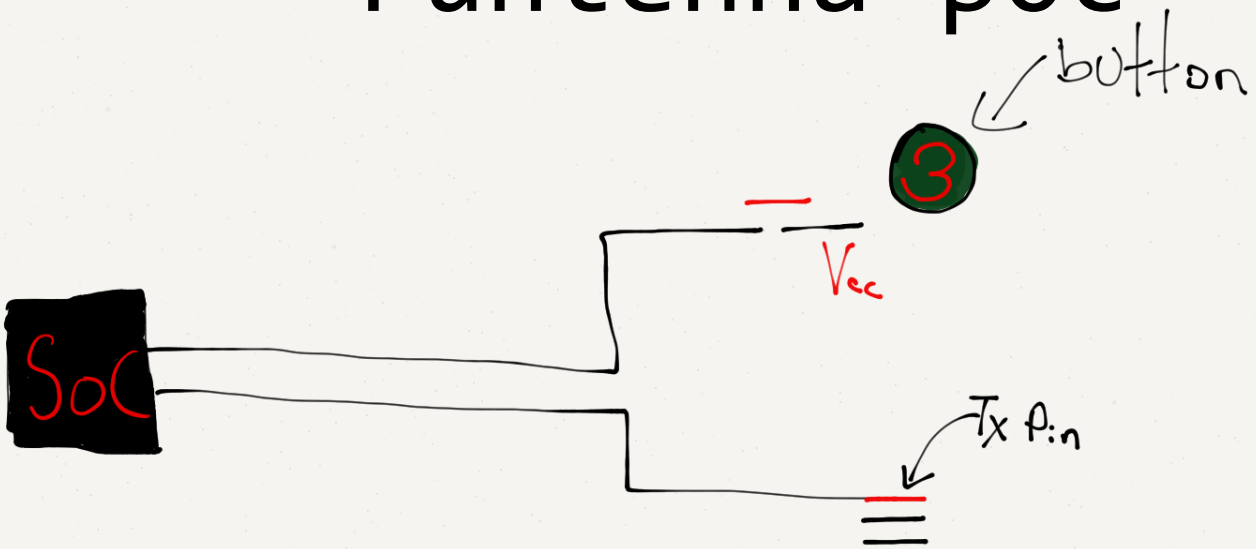
Wire!

Funtenna p0c

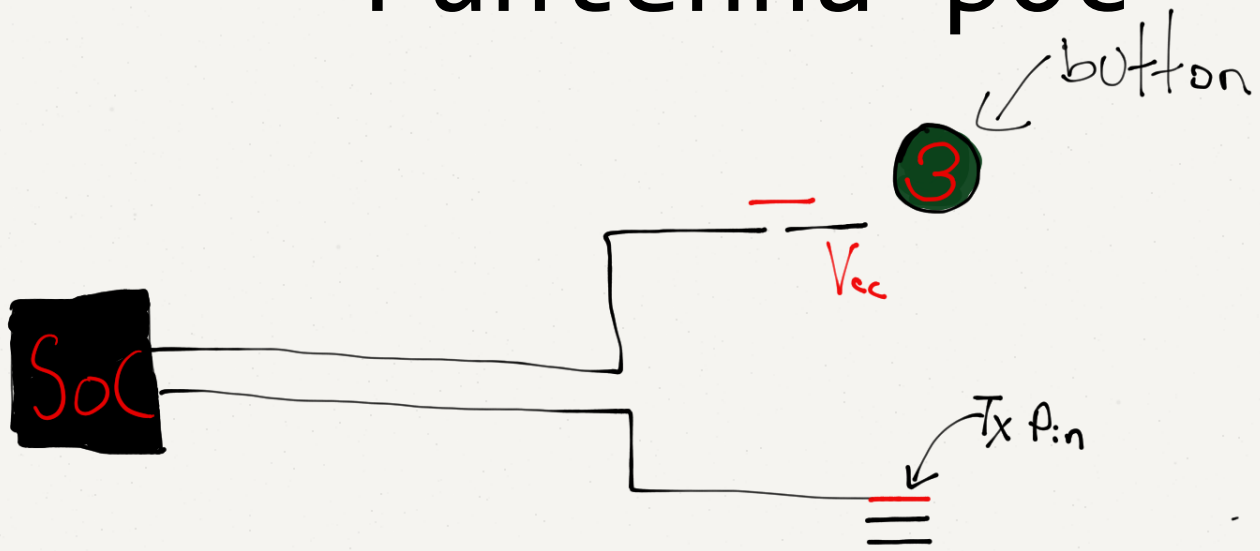
Funtenna p0c

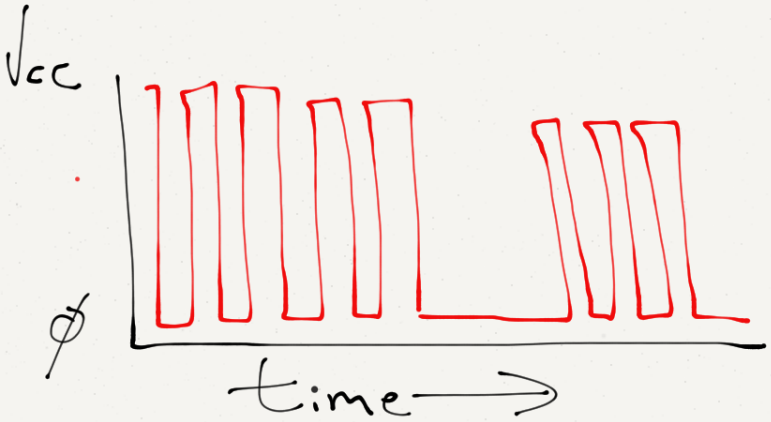


Funtenna p0c

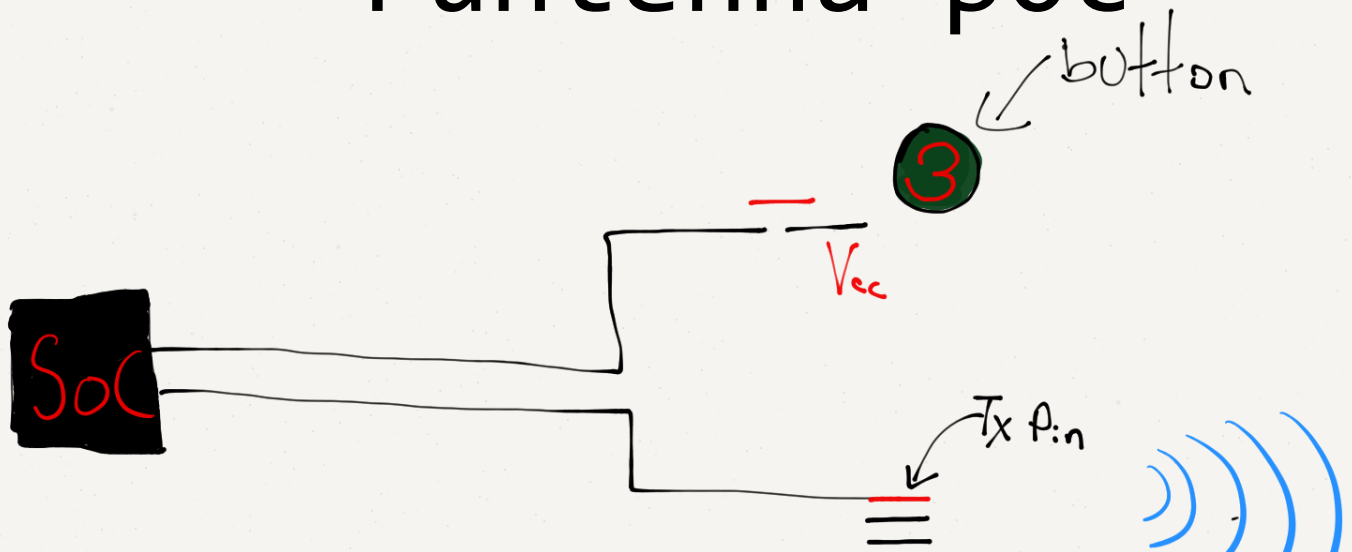


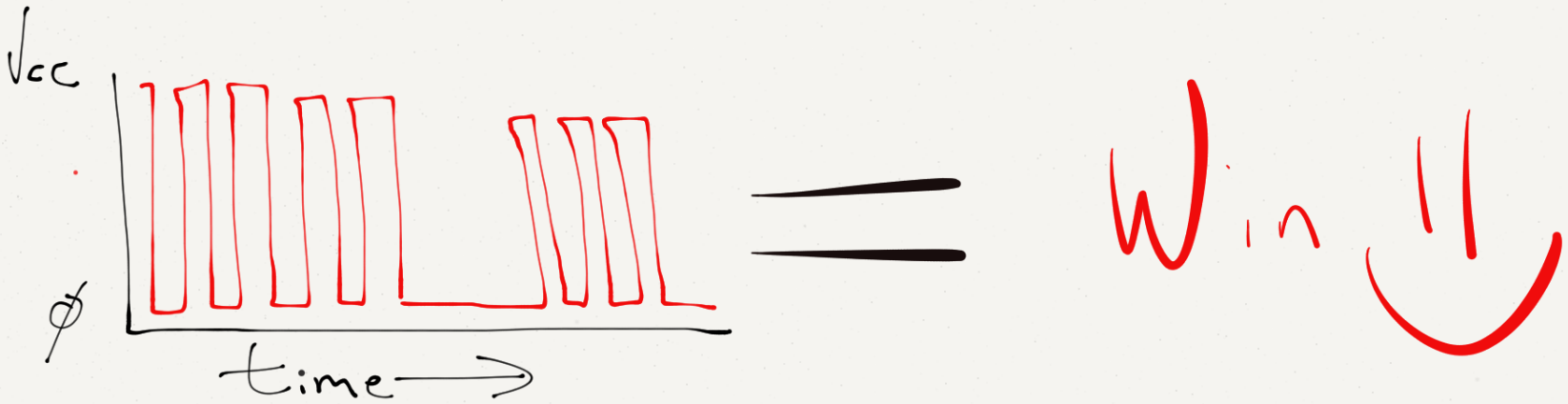
Funtenna p0c



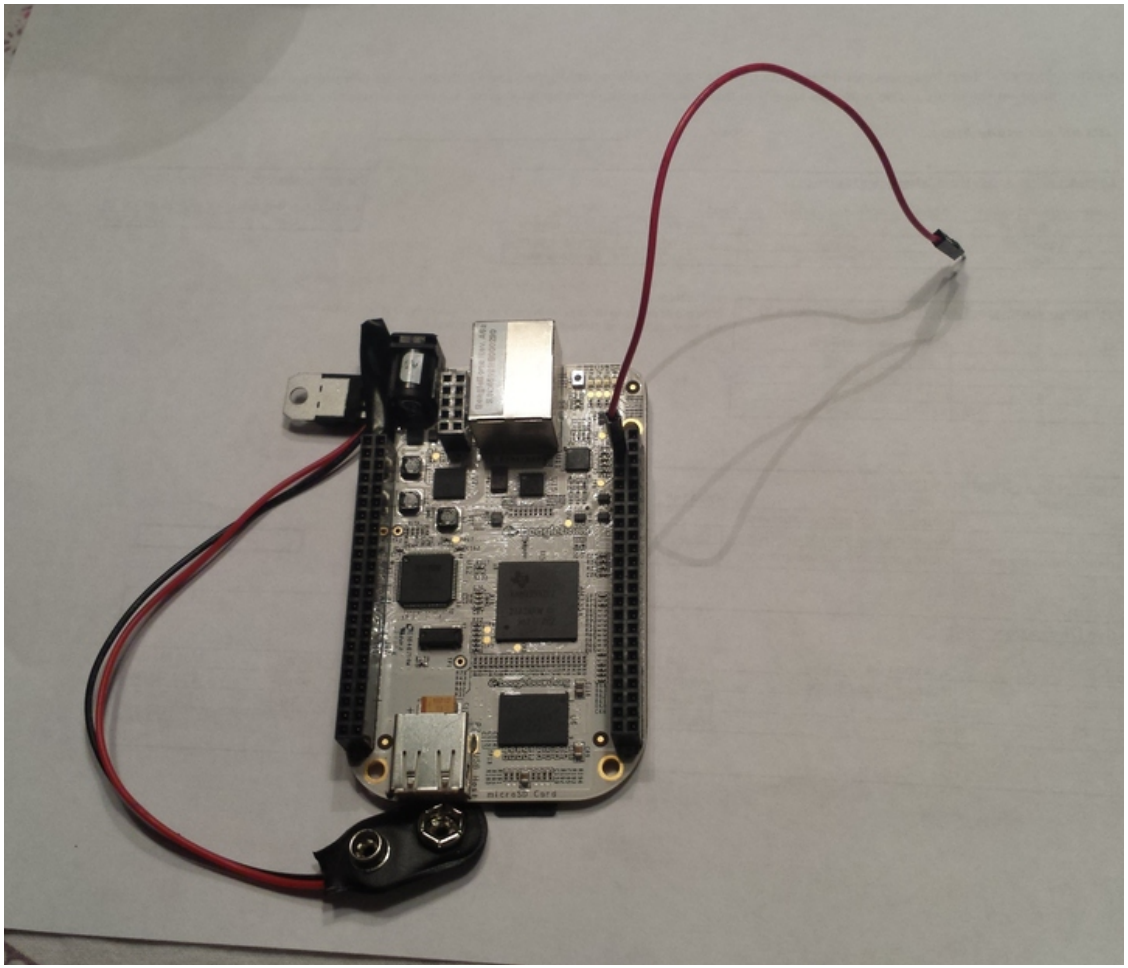


Funtenna p0c



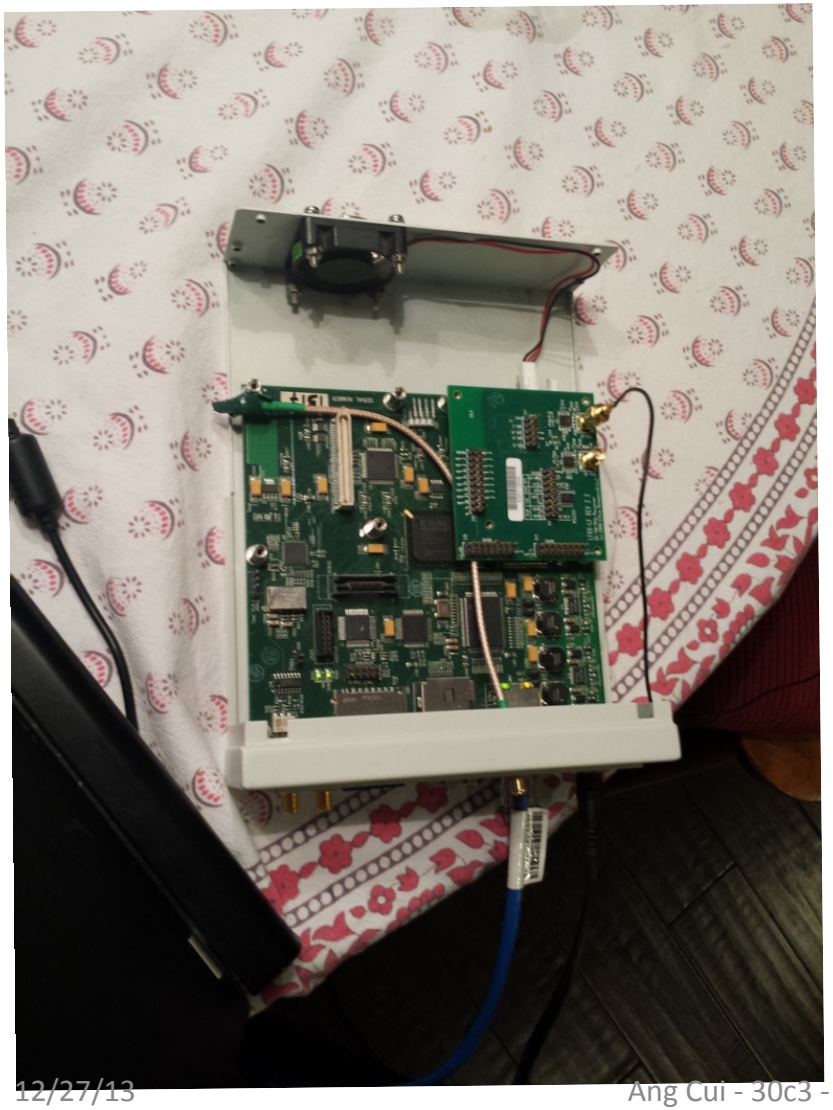


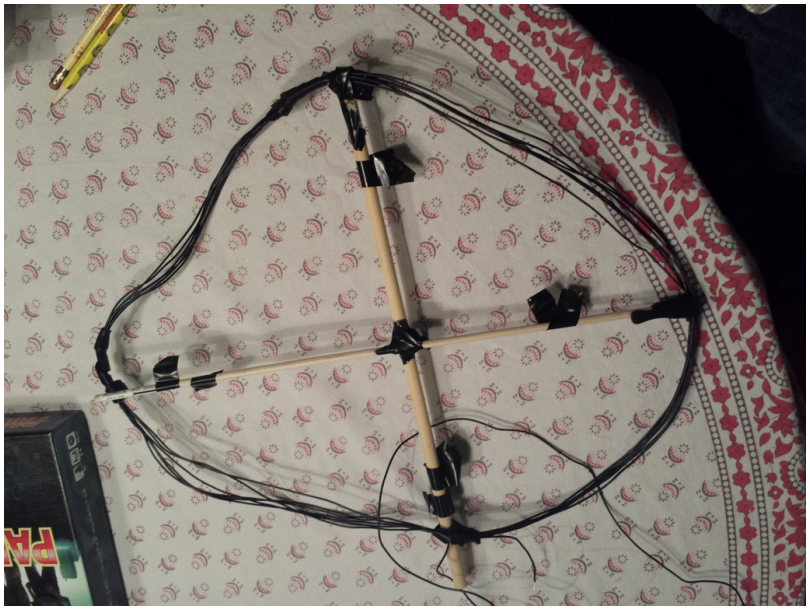
Reference Implementation



www.funtenna.org

Reference Implementation

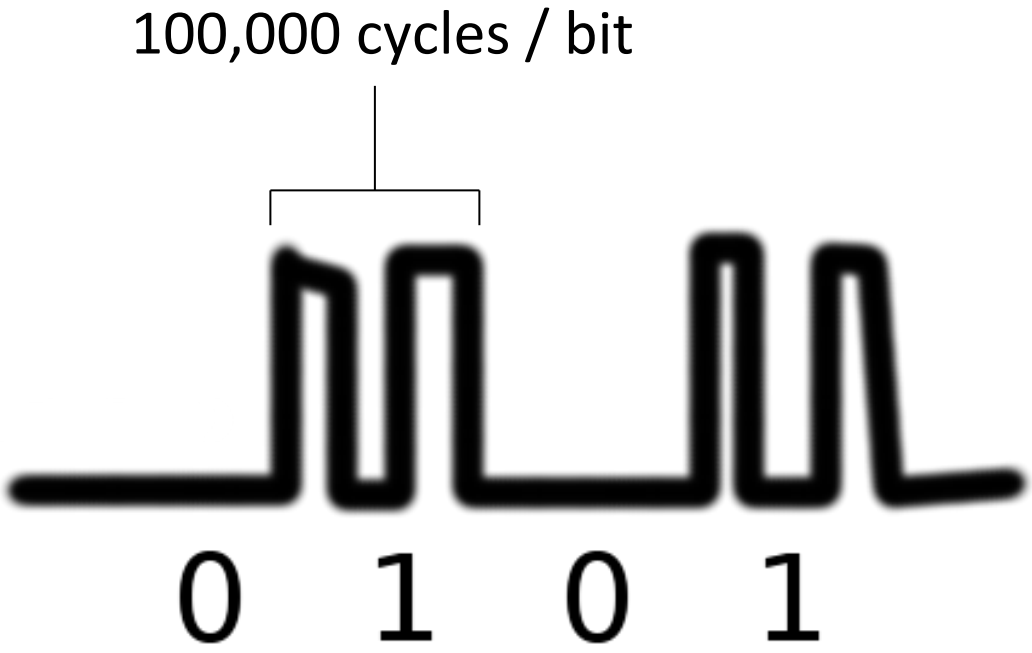




DEMO

On-Off Keying

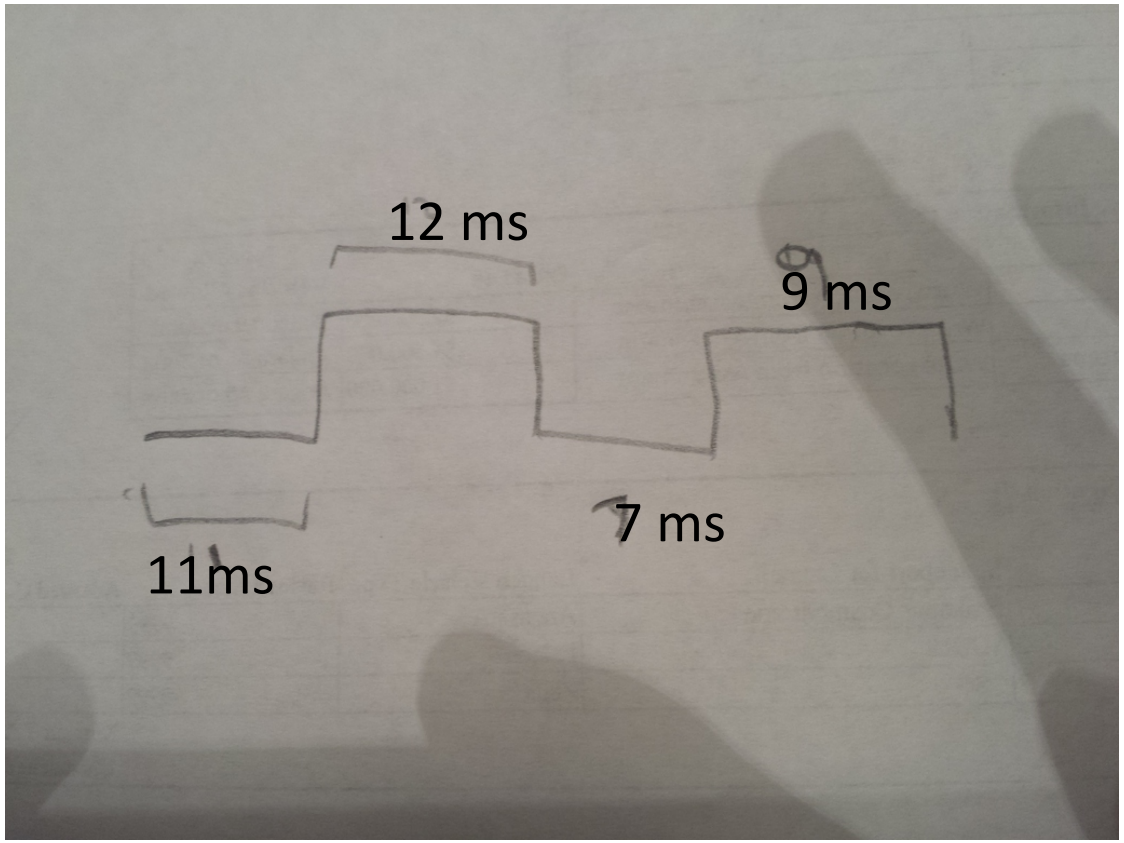
On-Off Keying



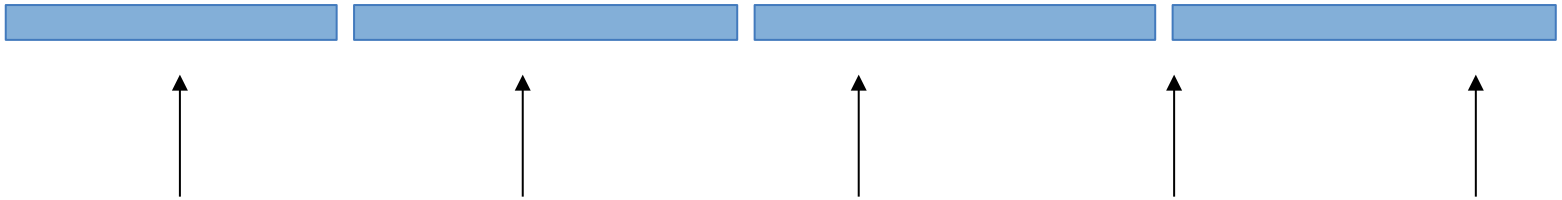
Message Format

- Prologue: 11001100
- Message length
- Hamming[7,4] code with extra parity bit

Results: Irregular Symbol Duration



Results: Sample Drift



Results: Bandwidth

12.5 MHz at 100,000 cycles / bit

= 125 bits / sec

~= 16 bytes / sec

Theoretical Bandwidth

1,500 cycles / bit

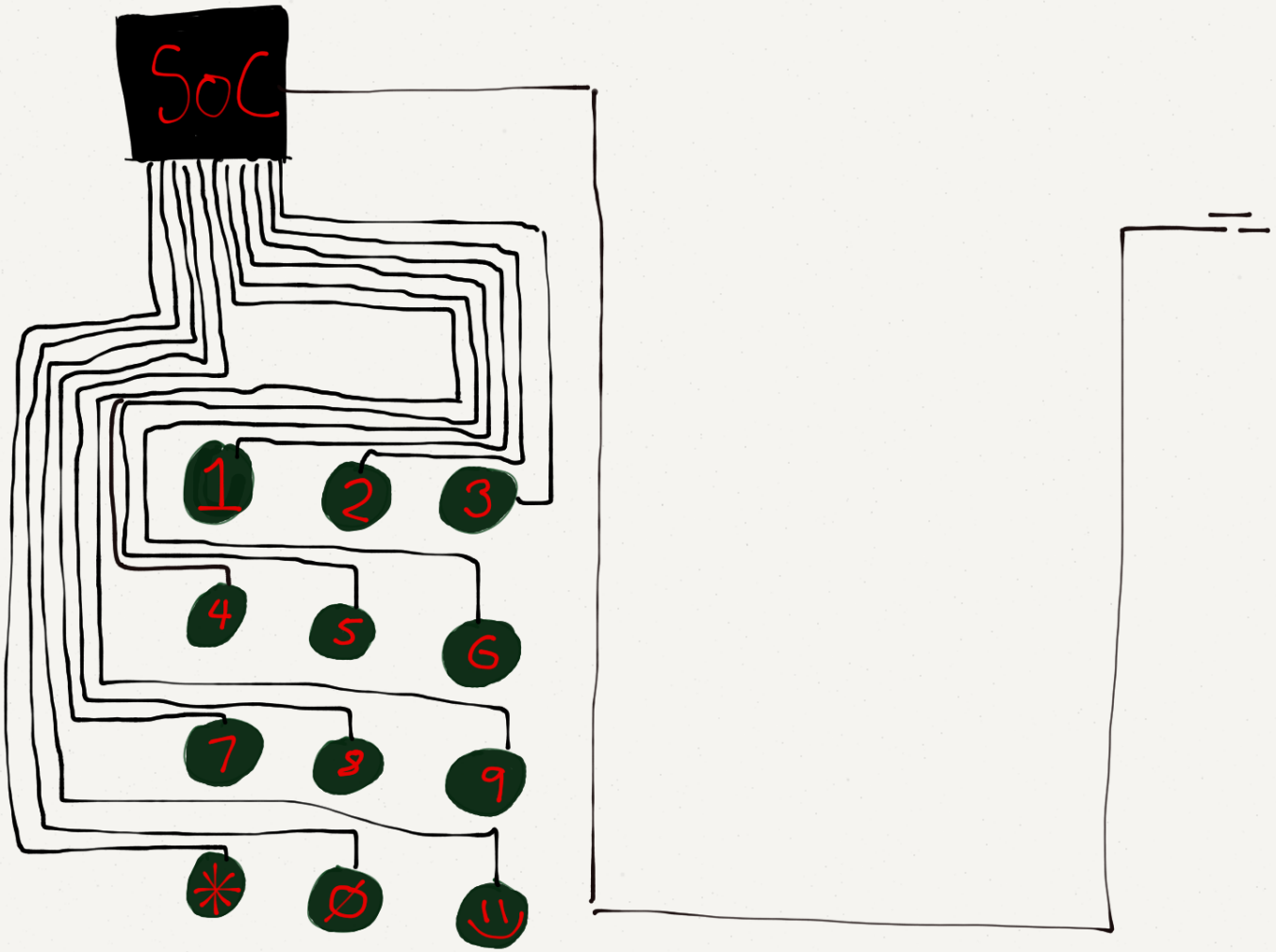
~= 8.3 kbit / sec

Other Improvements

- Frequency Shift Keying
- Protocol improvements:
  - Packets of fixed size (fix sample drift; a little more efficient)
- Regulate duration better

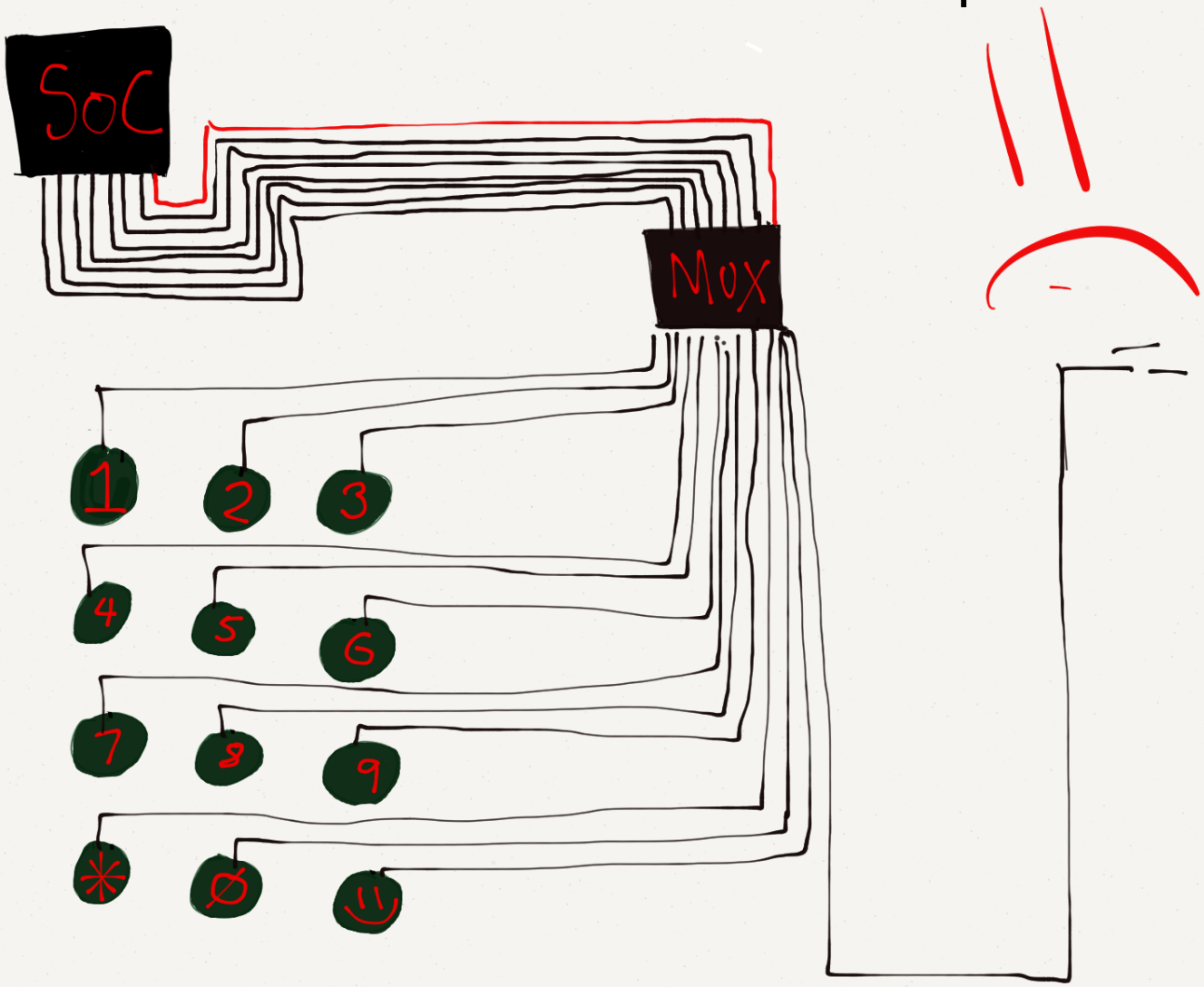
Funtenna p0c meets reality

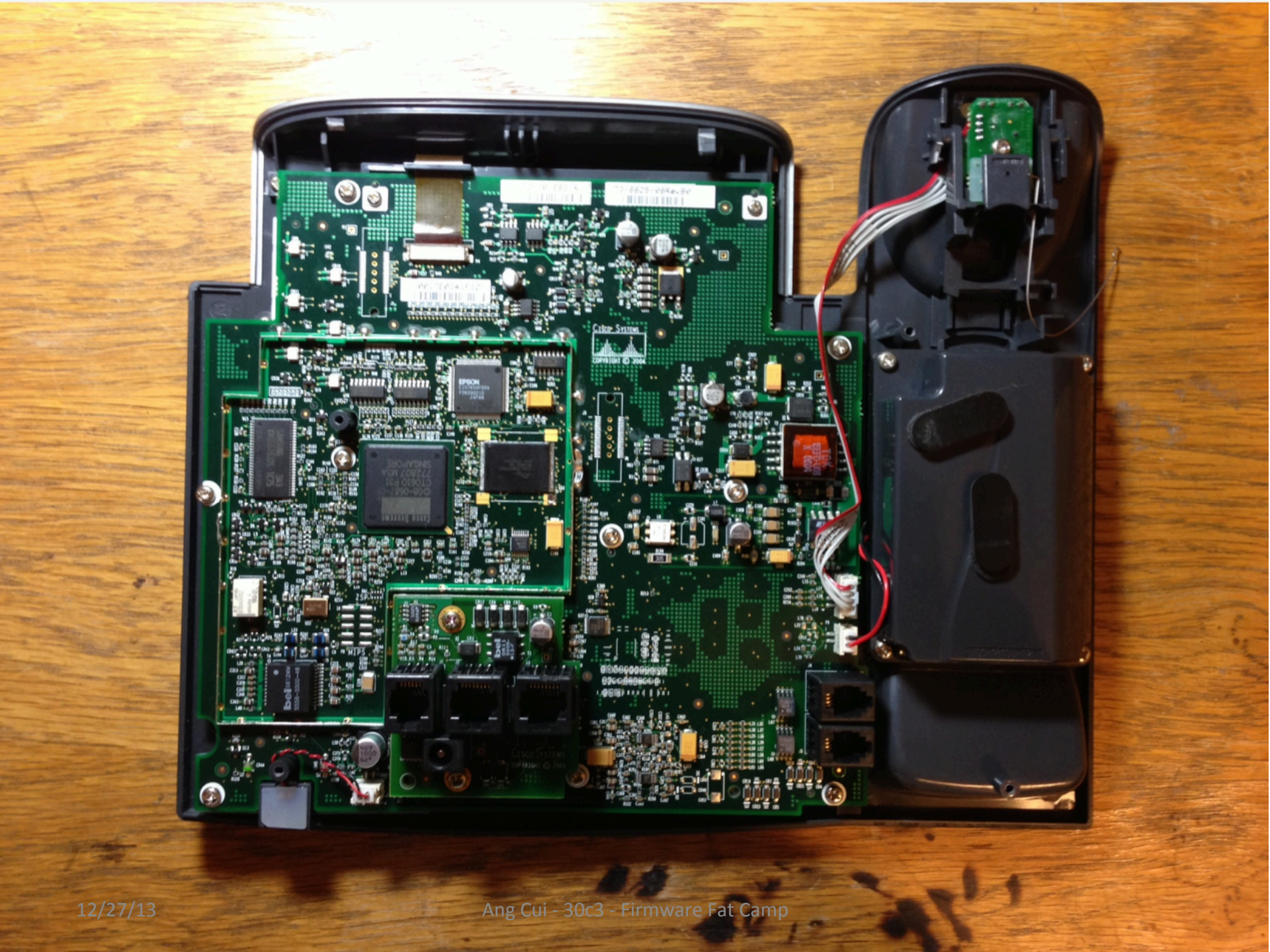
Would be so nice if GPIO worked this way…

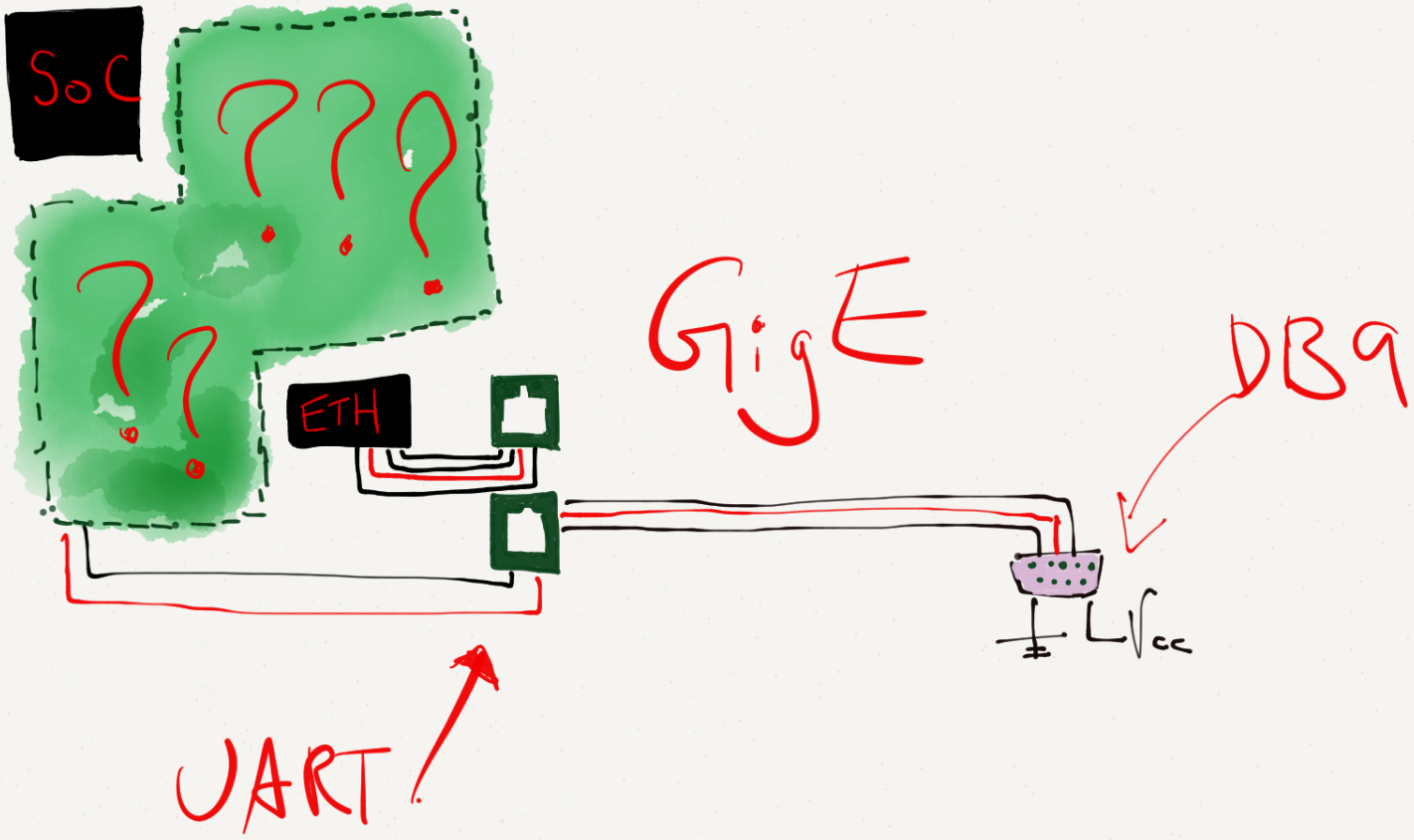


But GPIO input usually MUX'd

But GPIO input usually MUX'd

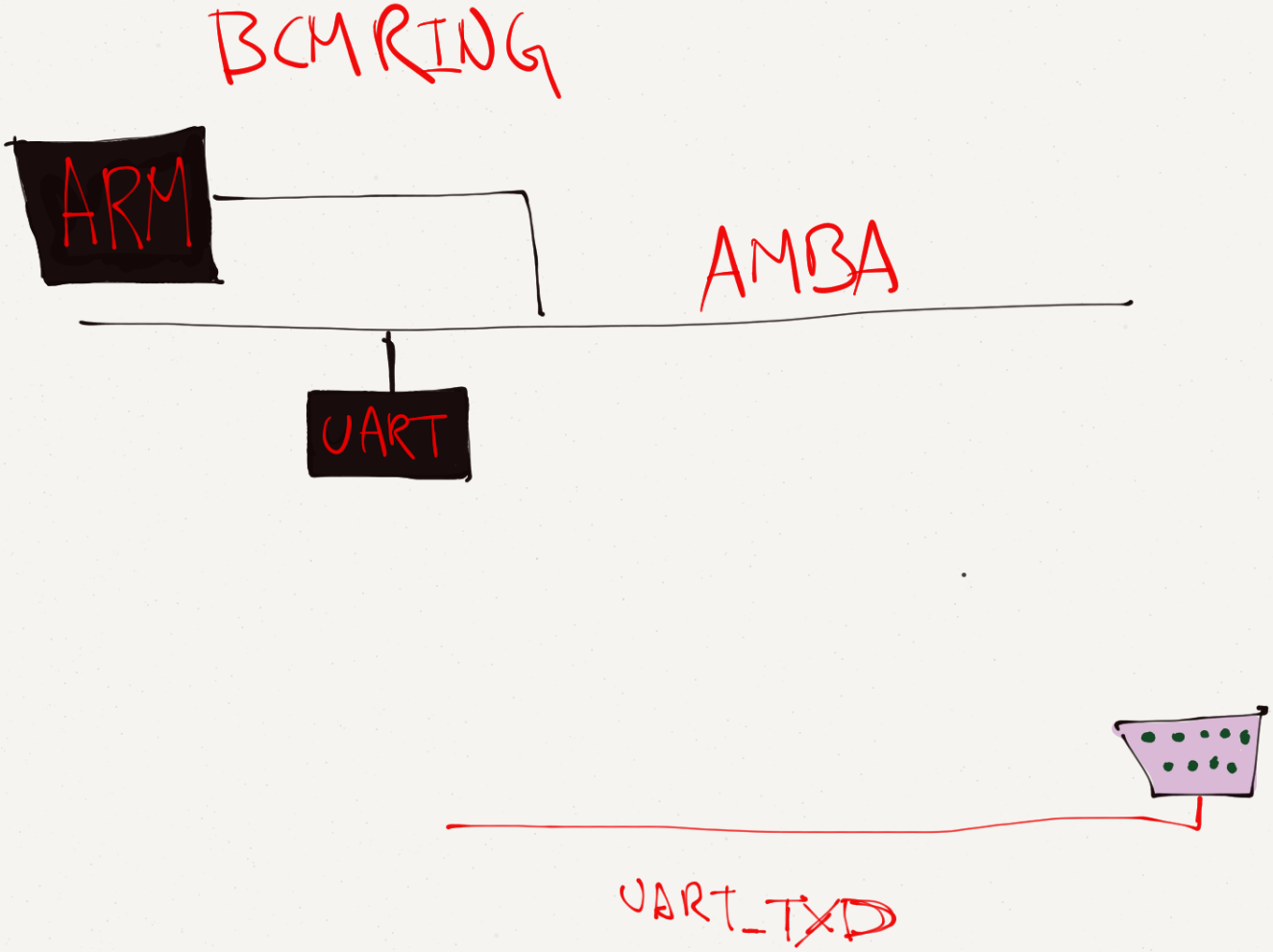





SoC
???
ETH
GigE
DB9
Vcc
UART

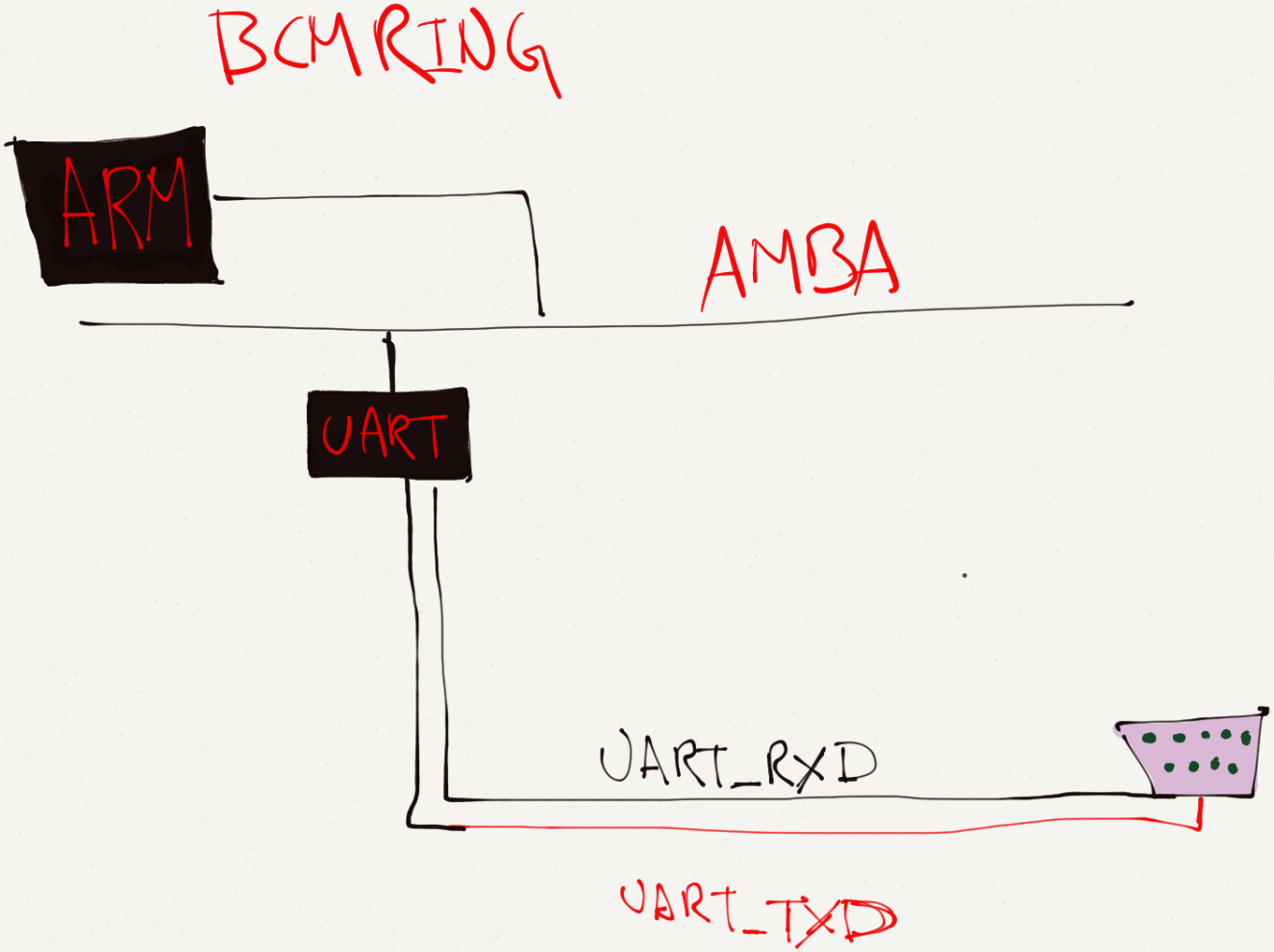
ARM / Linux / PL011 UART EXAMPLE

ARM / Linux / PL011 UART EXAMPLE

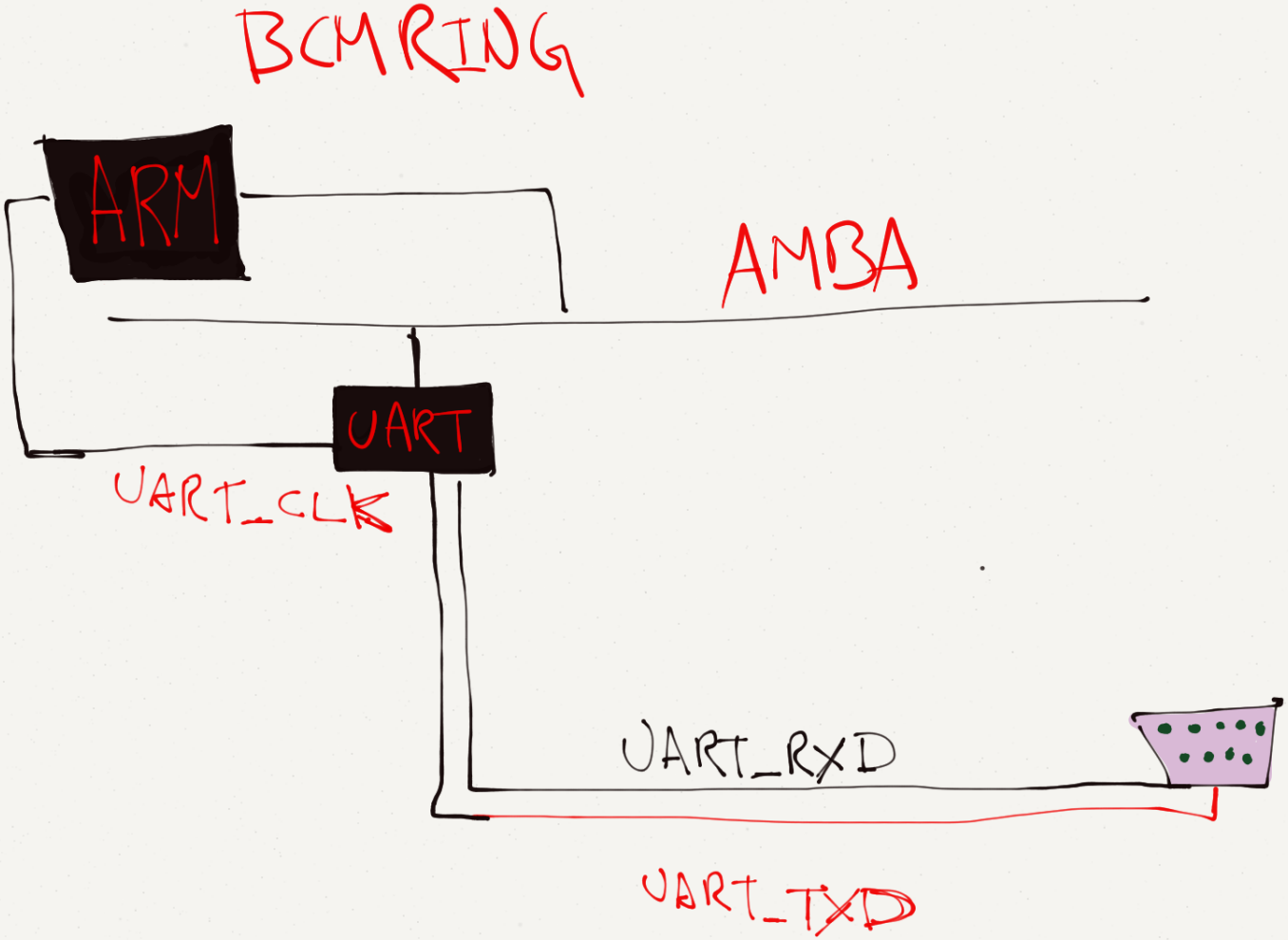


AMBA: Advanced Microcontroller Bus Architecture

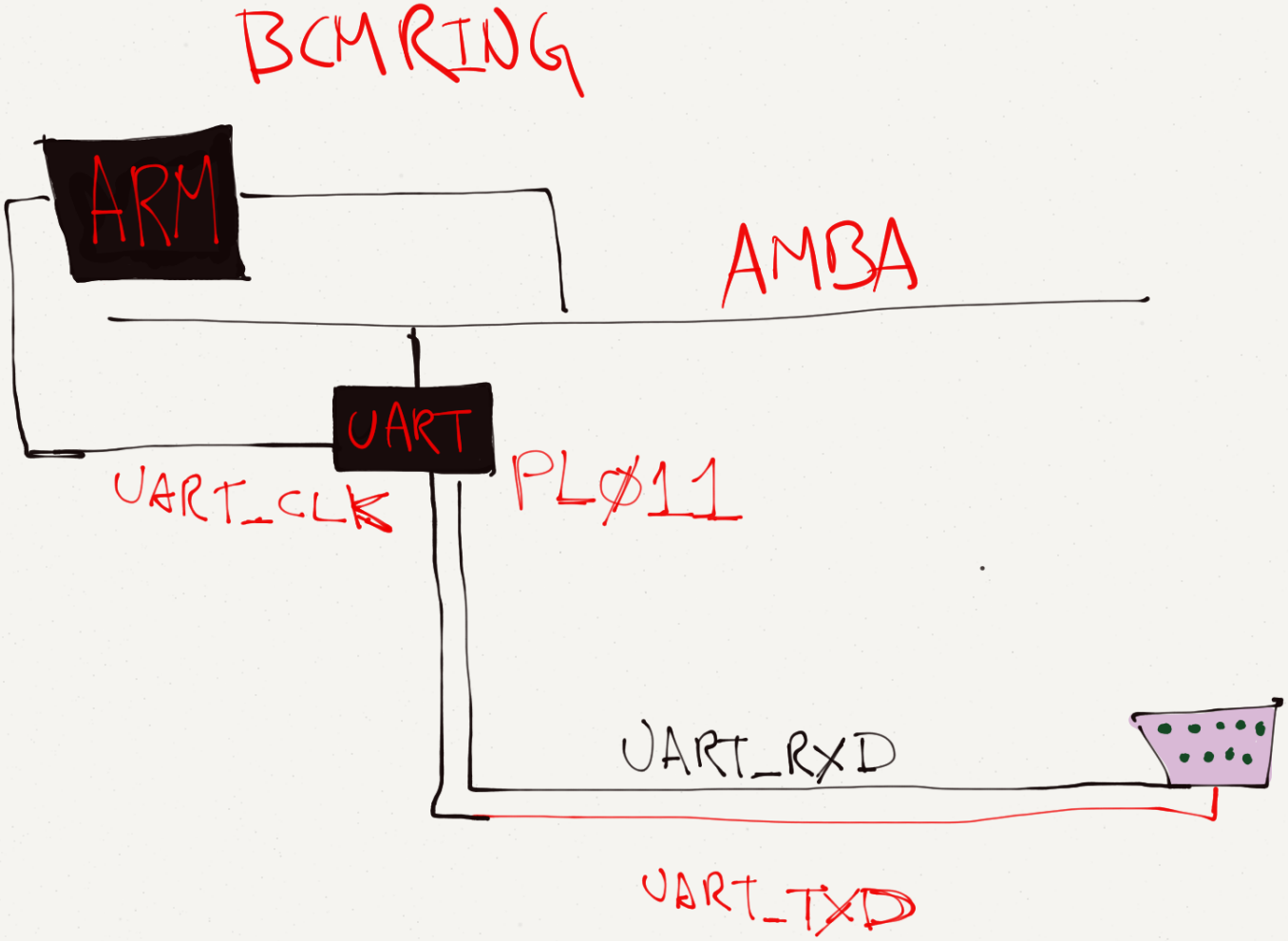
ARM / Linux / PL011 UART EXAMPLE



ARM / Linux / PL011 UART EXAMPLE

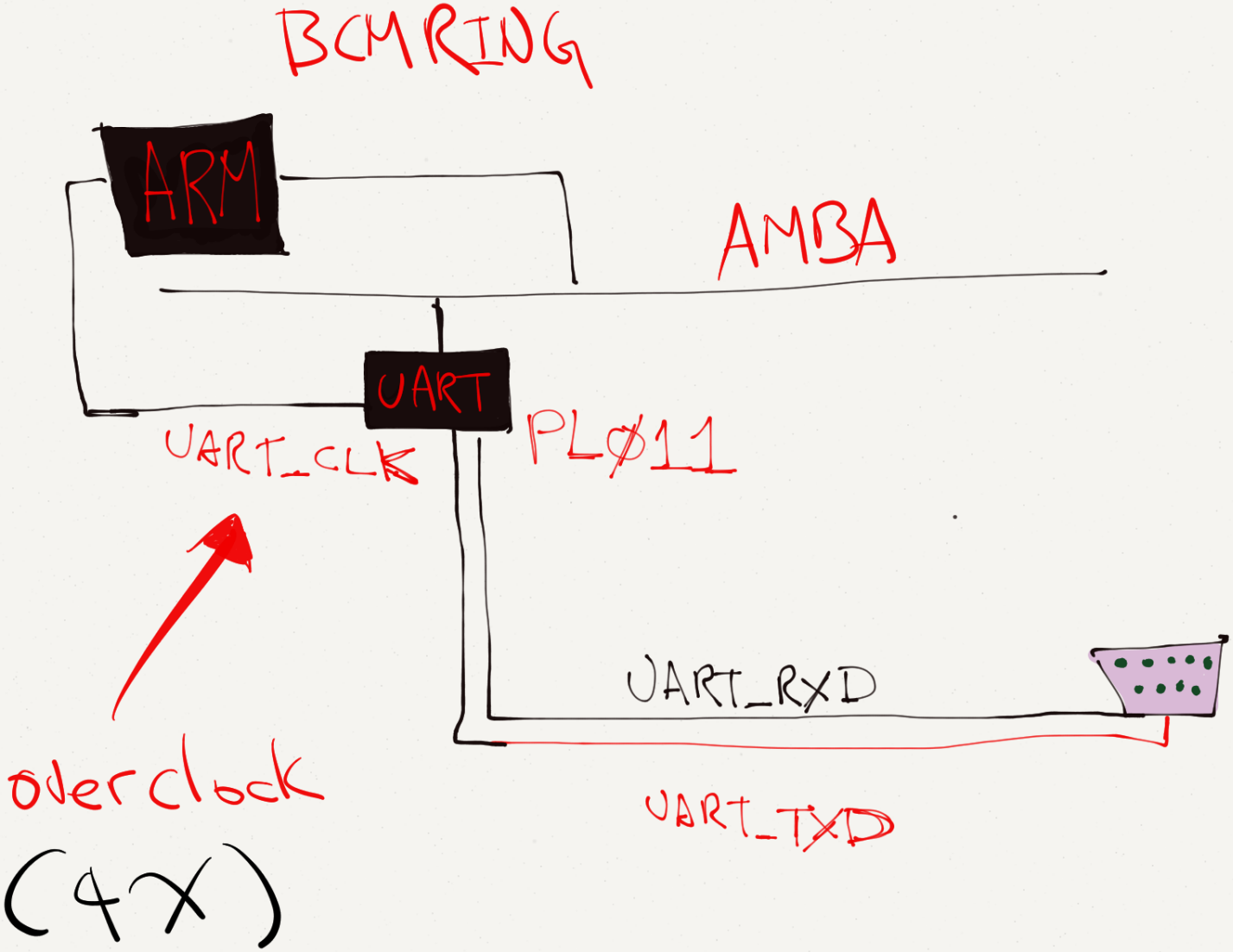


ARM / Linux / PL011 UART EXAMPLE



PL011 Datasheet http://infocenter.arm.com/help/topic/com.arm.doc.ddi0183f/DDI0183.pdf

ARM / Linux / PL011 UART EXAMPLE



lxr.free-electrons.com/source/arch/arm/mach-bcmring/csp/chipc/chipcHw.c?v=2.6.32#L261

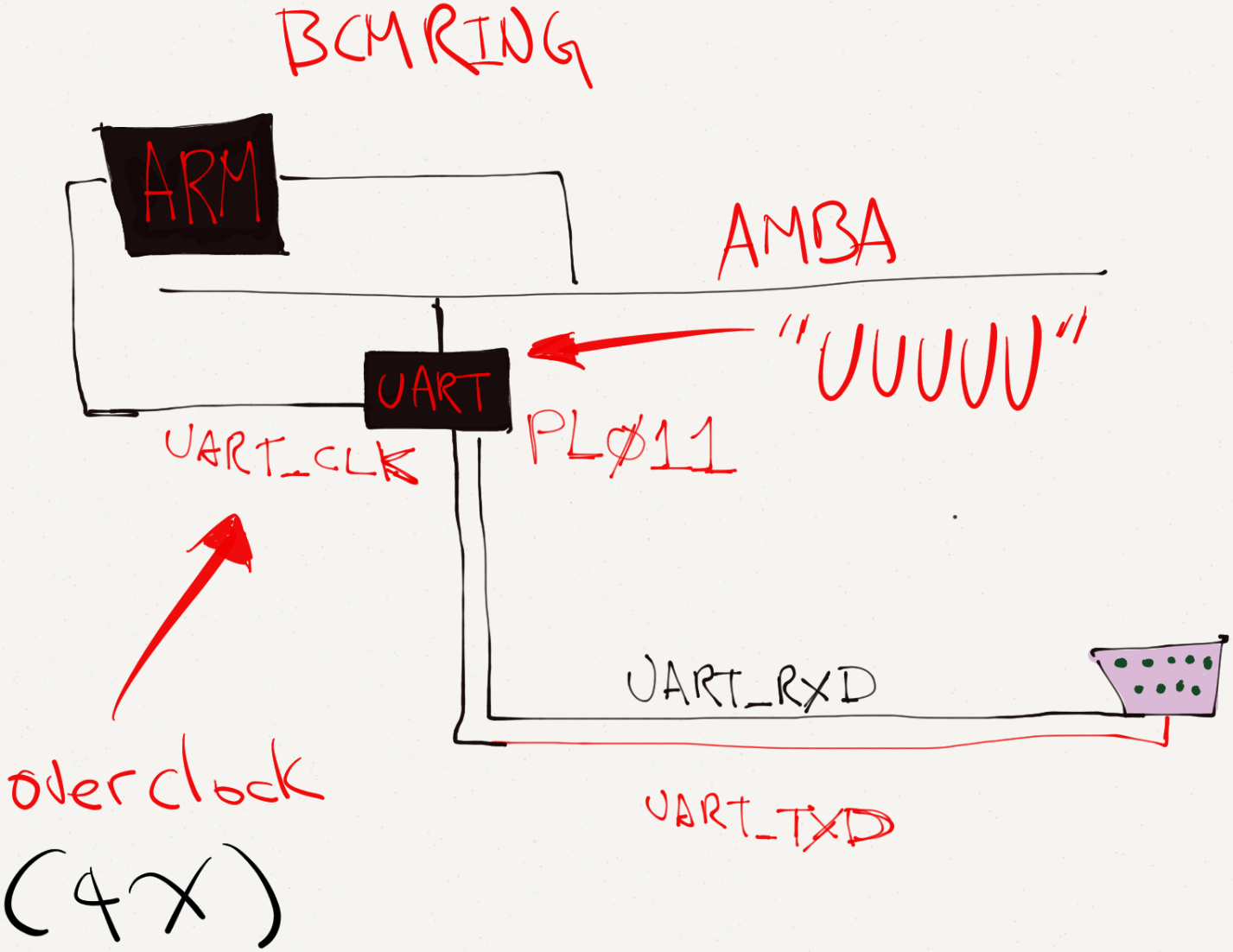
```
260 /****************************************************************************/
261 chipcHw_freq chipcHw_setClockFrequency(chipcHw_CLOCK_e clock,   /*  [ IN ] Configurable clock */
262                                         uint32_t freq    /*  [ IN ] Clock frequency in Hz */
263     ) {
```

lxr.free-electrons.com/source/arch/arm/mach-bcmring/include/mach/csp/chipcHw_def.h?v=2.6.32#L60

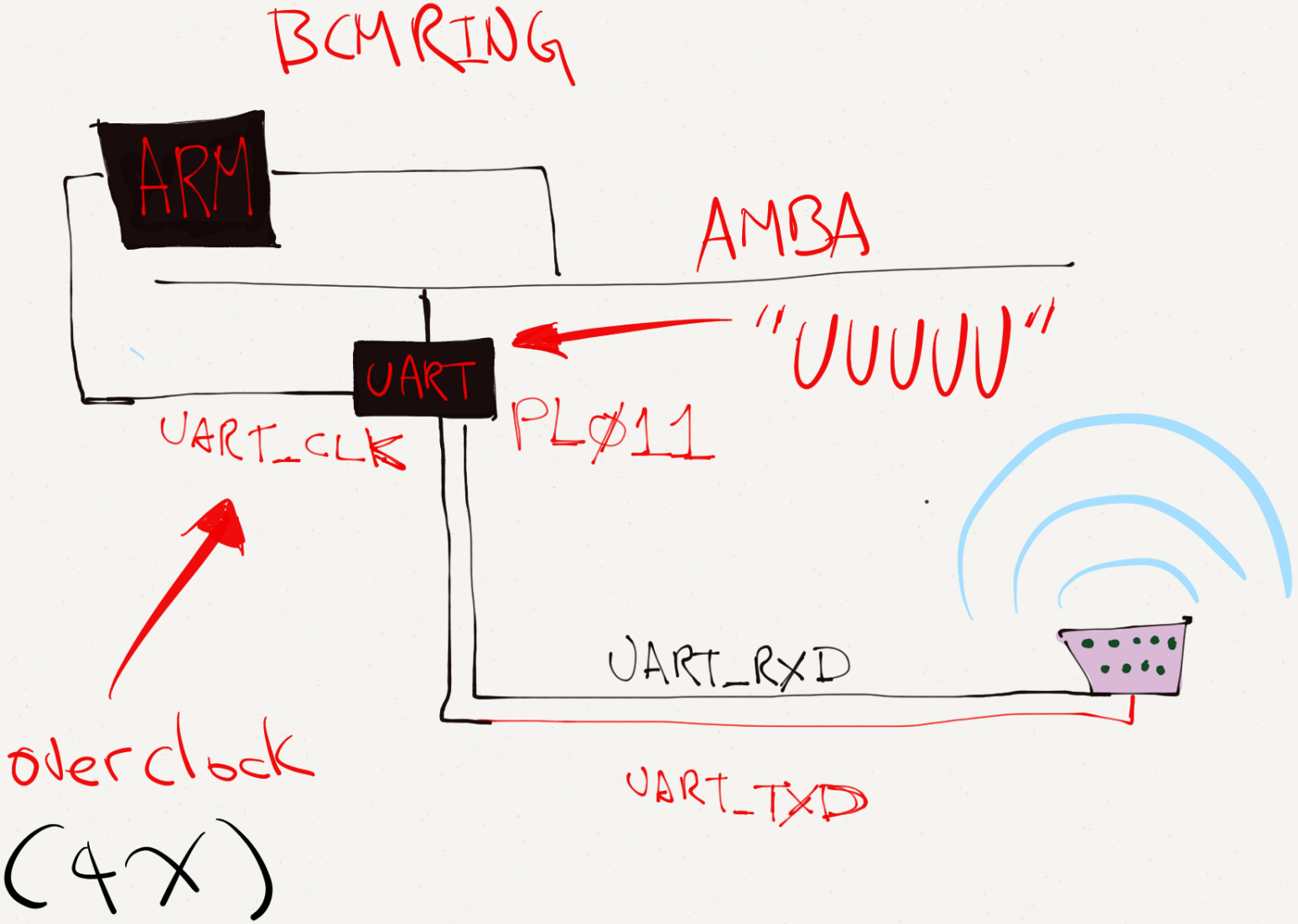
```
31 typedef uint32_t chipcHw_freq;
32
33 /* Configurable miscellaneous clocks */
34 typedef enum {
35         chipcHw_CLOCK_DDR,      /* DDR PHY Clock */
36         chipcHw_CLOCK_ARM,      /* ARM Clock */
37         chipcHw_CLOCK_ESW,      /* Ethernet Switch Clock */
38         chipcHw_CLOCK_VPM,      /* VPM Clock */
39         chipcHw_CLOCK_ESW125,   /* Ethernet MII Clock */
40         chipcHw_CLOCK_UART,     /* UART Clock */
41         chipcHw_CLOCK_SDIO0,    /* SDIO 0 Clock */
42         chipcHw_CLOCK_SDIO1,    /* SDIO 1 Clock */
43         chipcHw_CLOCK_SPI,      /* SPI Clock */
44         chipcHw_CLOCK_ETM,      /* ARM ETM Clock */
45
46         chipcHw_CLOCK_BUS,      /* BUS Clock */
47         chipcHw_CLOCK_OTP,      /* OTP Clock */
48         chipcHw_CLOCK_I2C,      /* I2C Host Clock */
49         chipcHw_CLOCK_I2S0,     /* I2S 0 Host Clock */
50         chipcHw_CLOCK_RTBUS,    /* DDR PHY Configuration Clock */
51         chipcHw_CLOCK_APM100,   /* APM100 Clock */
52         chipcHw_CLOCK_TSC,      /* Touch screen Clock */
53         chipcHw_CLOCK_LED,      /* LED Clock */
54
55         chipcHw_CLOCK_USB,      /* USB Clock */
56         chipcHw_CLOCK_LCD,      /* LCD CLock */
57         chipcHw_CLOCK_APM,      /* APM Clock */
58
59         chipcHw_CLOCK_I2S1,     /* I2S 1 Host Clock */
60 } chipcHw_CLOCK_e;
61
```

CLOCK_UART = 0x5

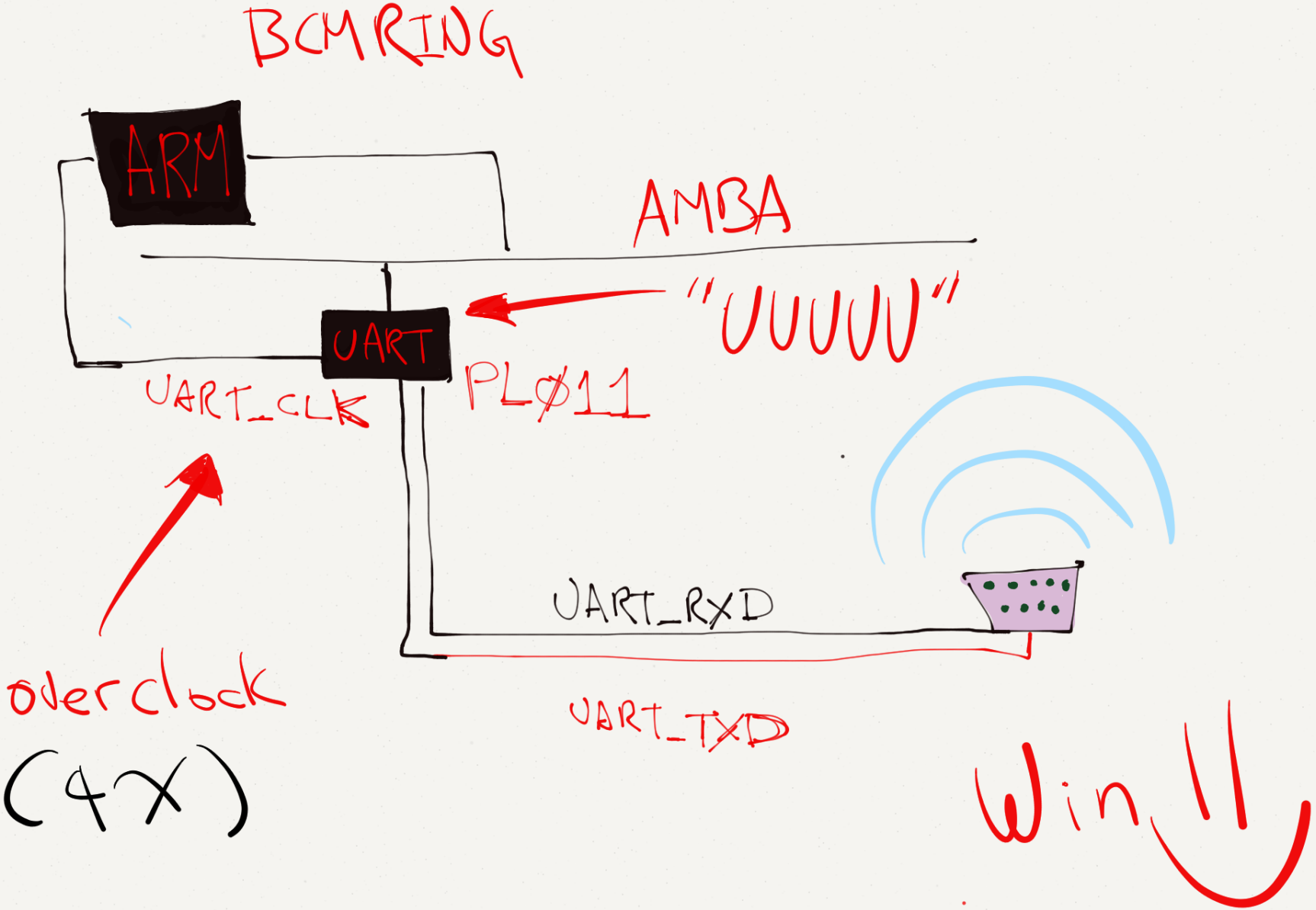
ARM / Linux / PL011 UART EXAMPLE



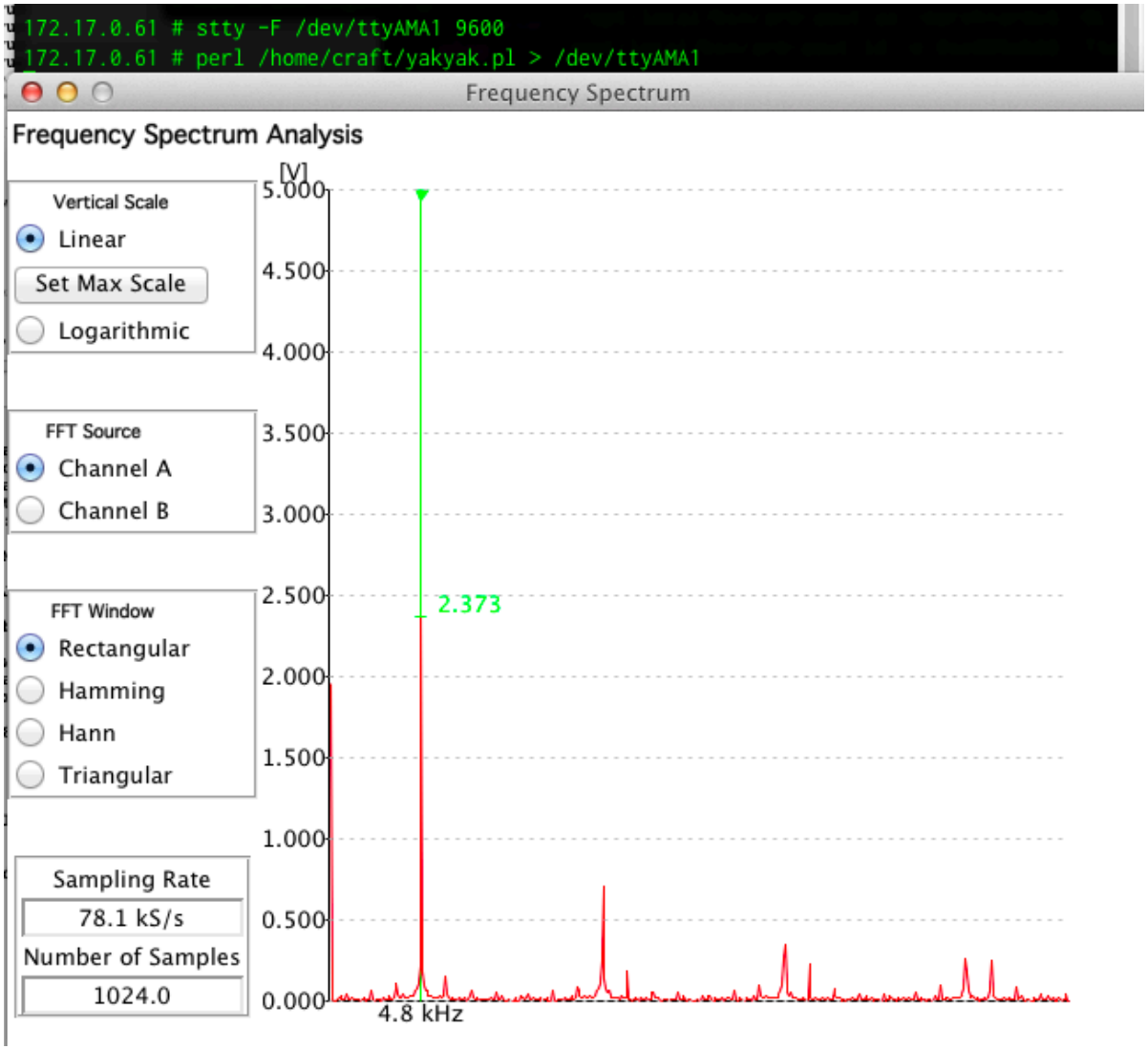
ARM / Linux / PL011 UART EXAMPLE



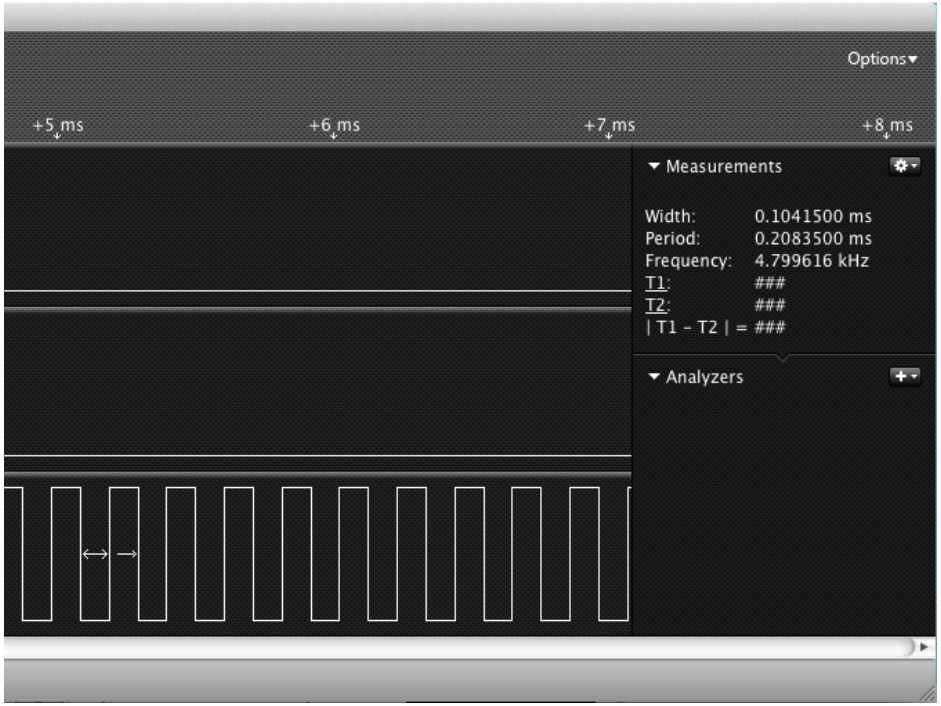
ARM / Linux / PL011 UART EXAMPLE

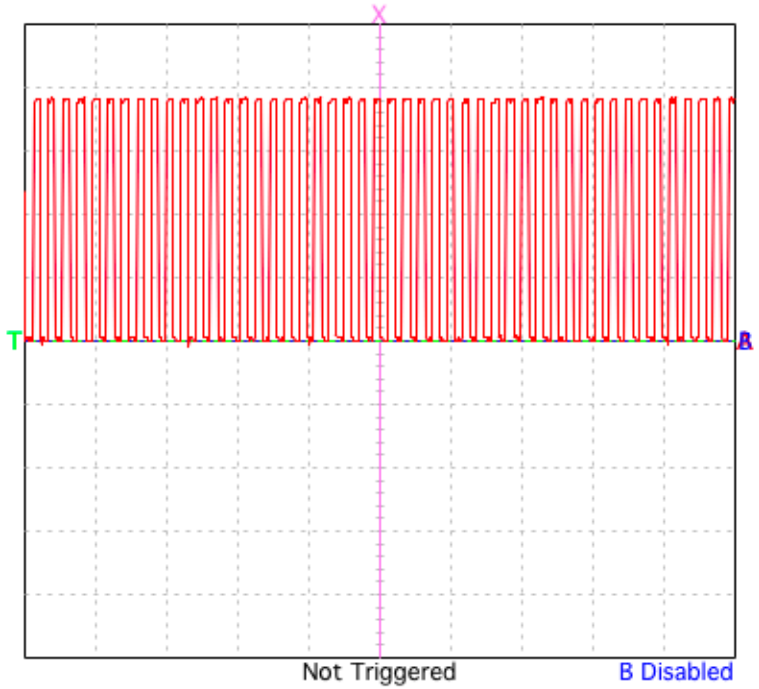


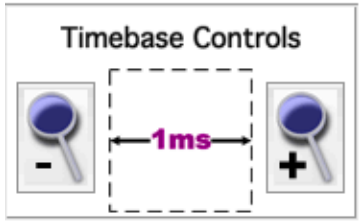
UART Funtenna 9600 baud



UART Funtenna 9600 baud

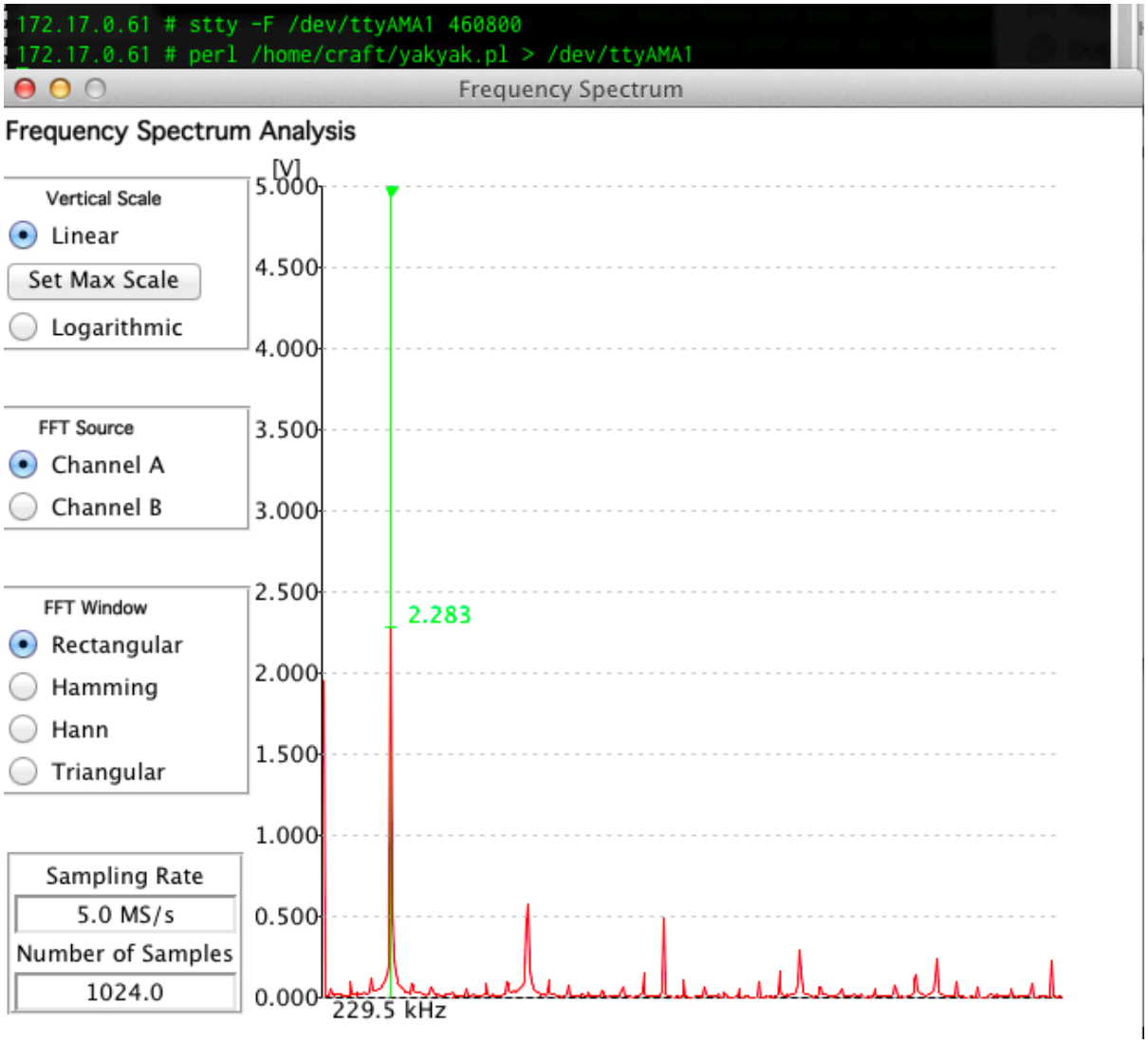




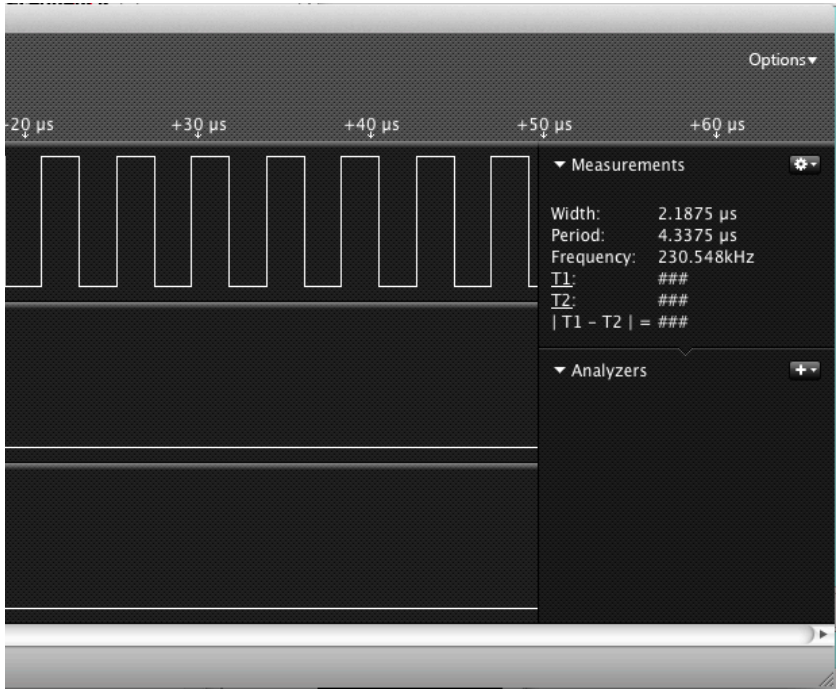


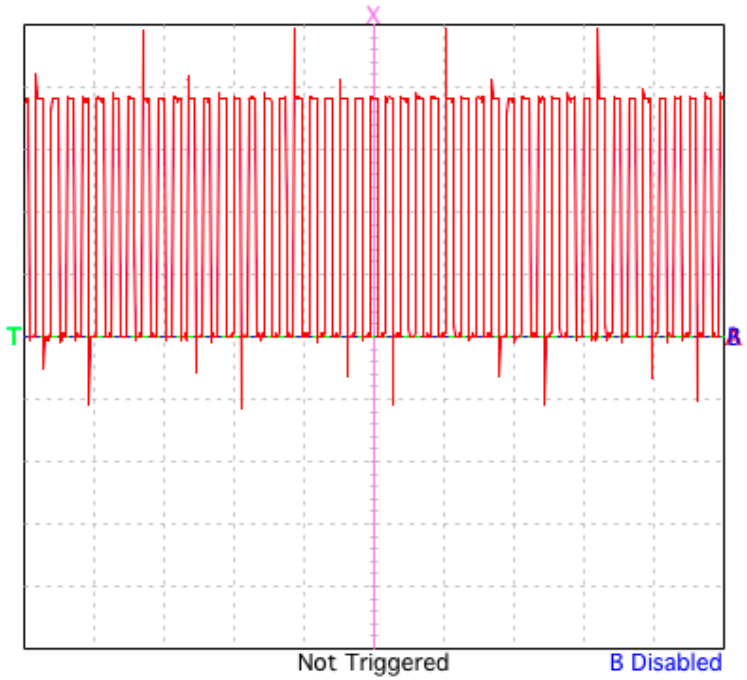
“UUUUUUUUUUUU”

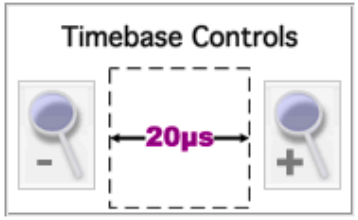
UART Funtenna 460800 baud



UART Funtenna 460800 baud

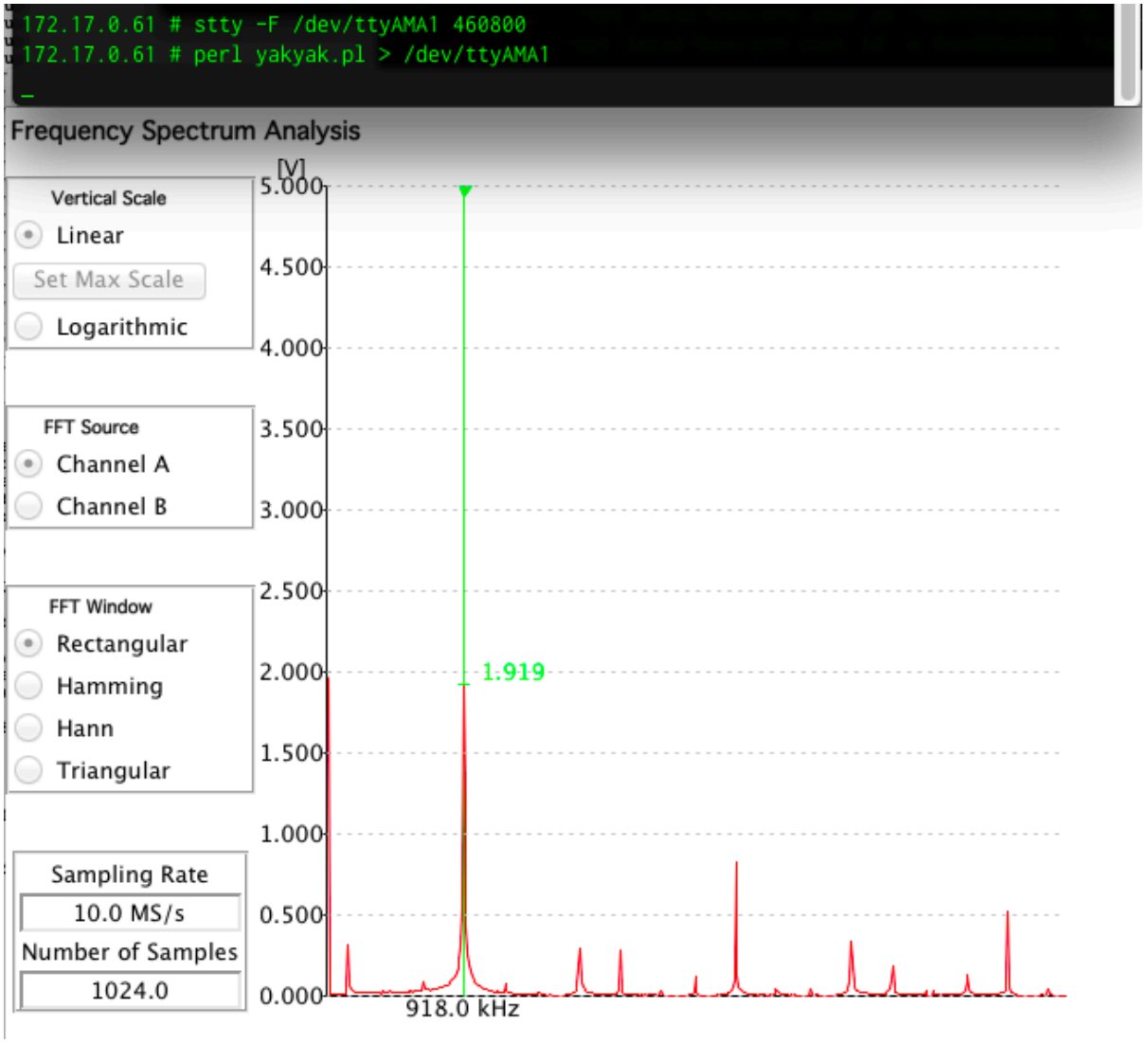




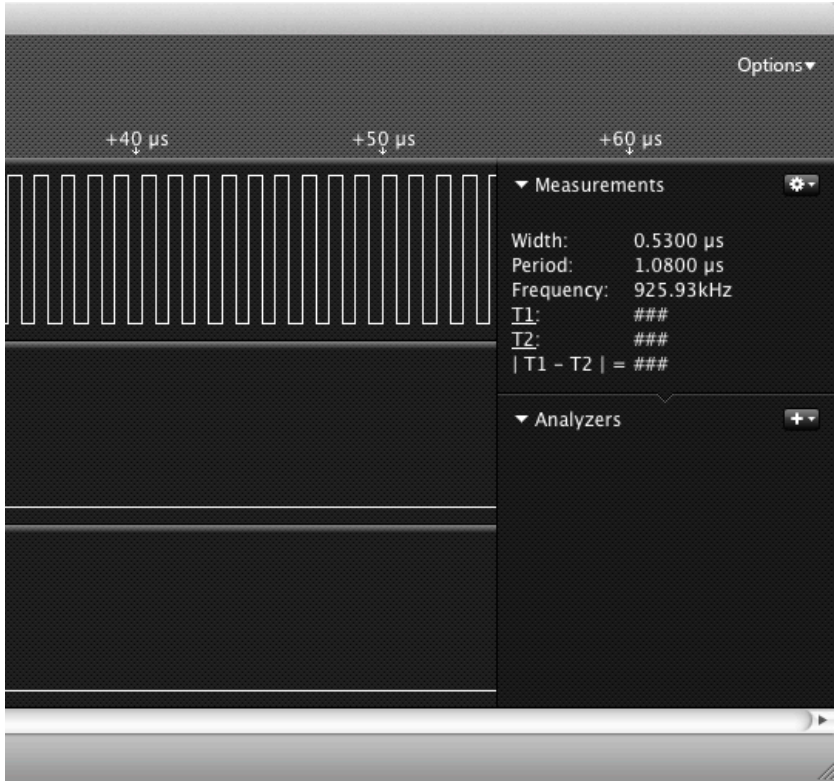


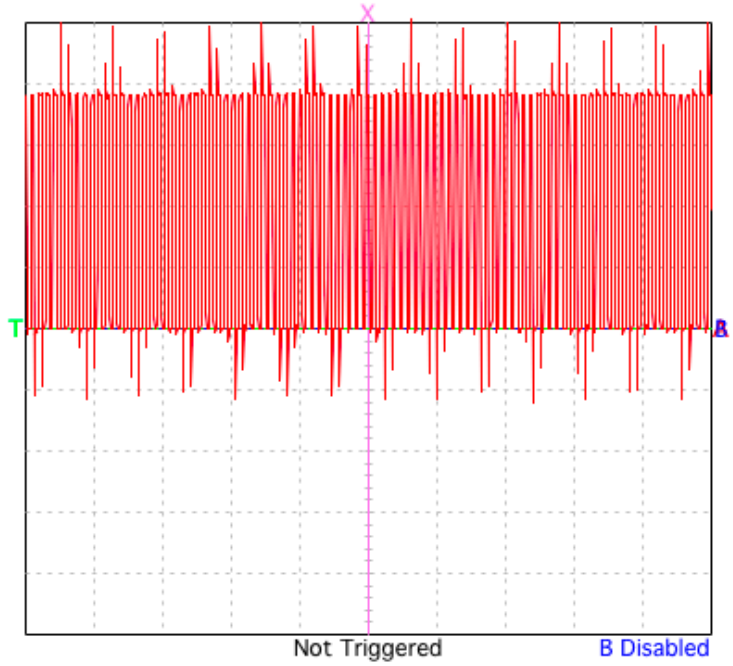
“UUUUUUUUUUUU”

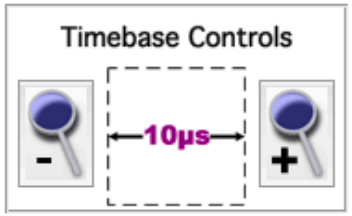
UART Funtenna “1843200” baud



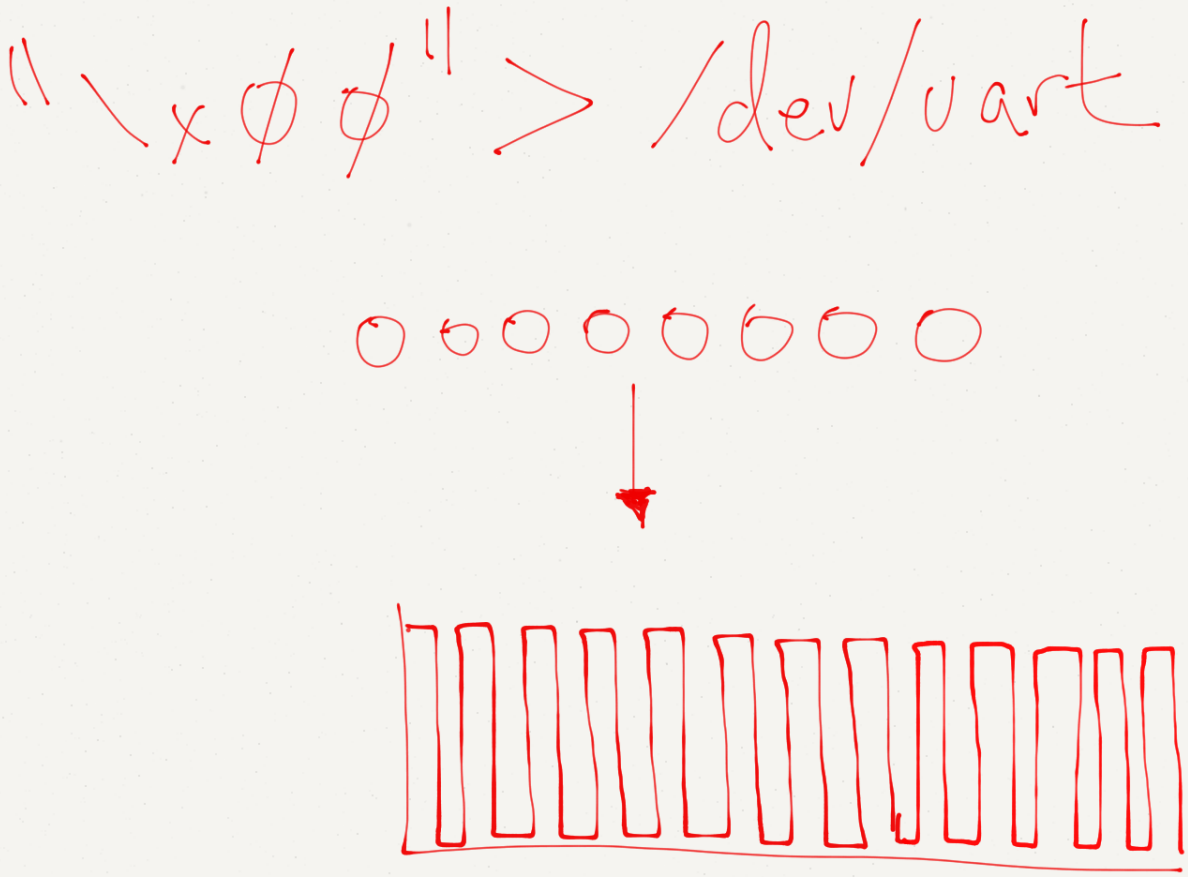
UART Funtenna “1843200” baud





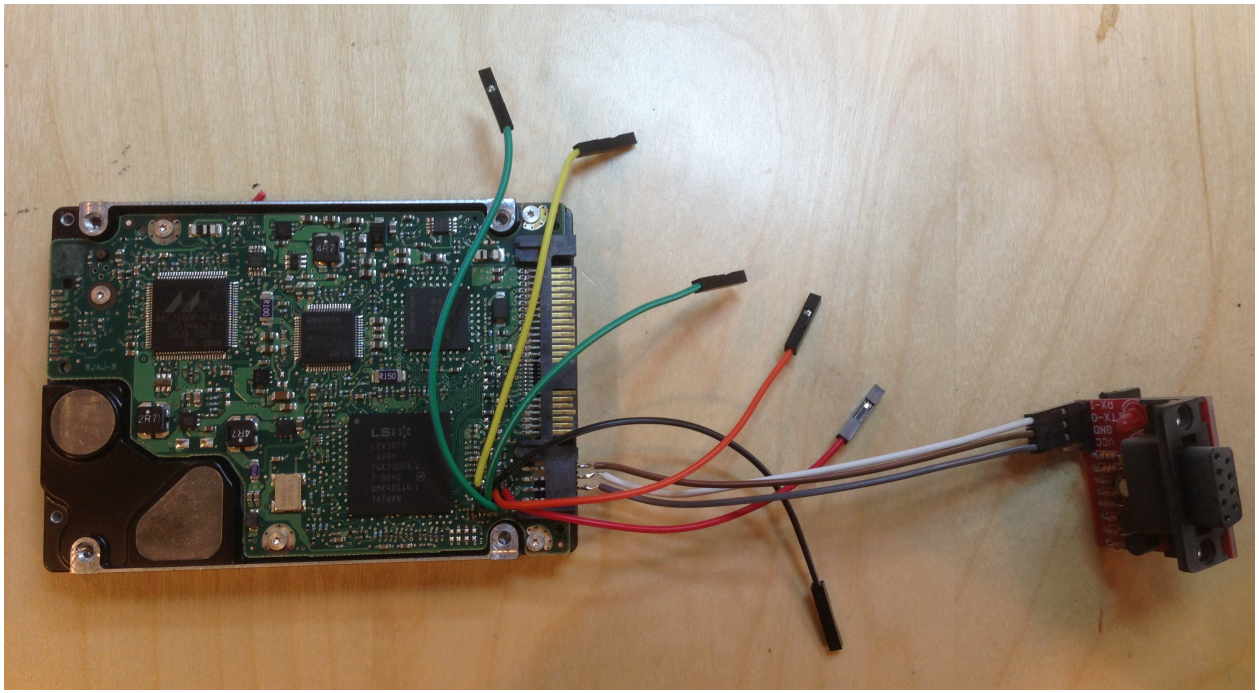


“UUUUUUUUUUUU”



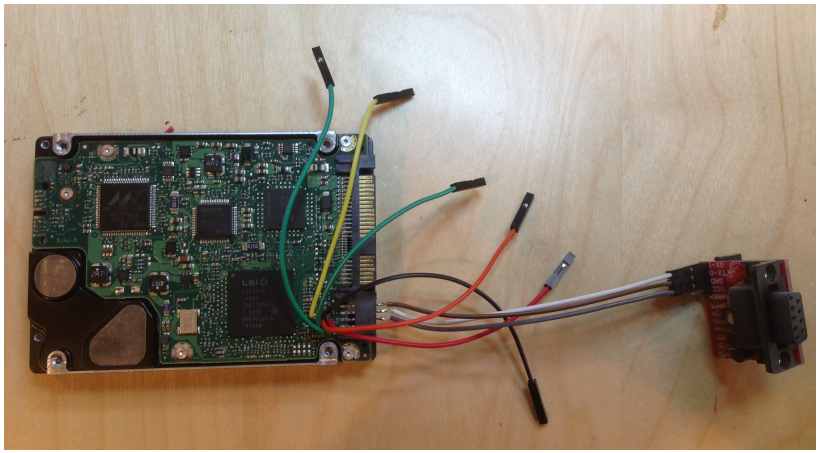
~ 1.8Mhz Carrier

Other Potential Funtennas



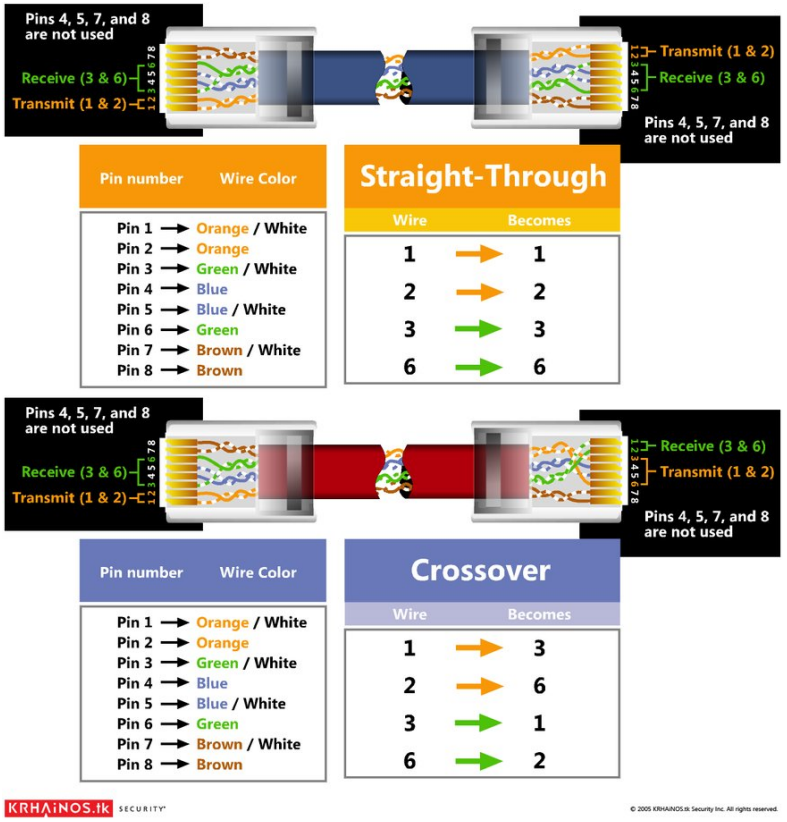
UART on HDD
“Implementation and Implications of a Stealth Hard-Drive Backdoor”

Other Potential Funtennas



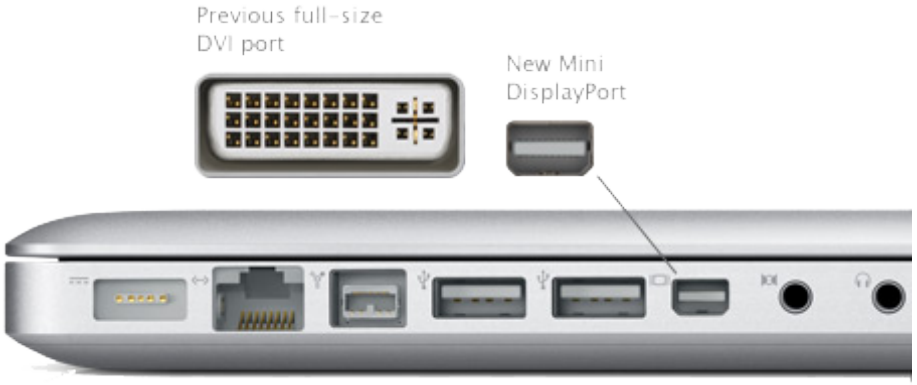
UART on HDD

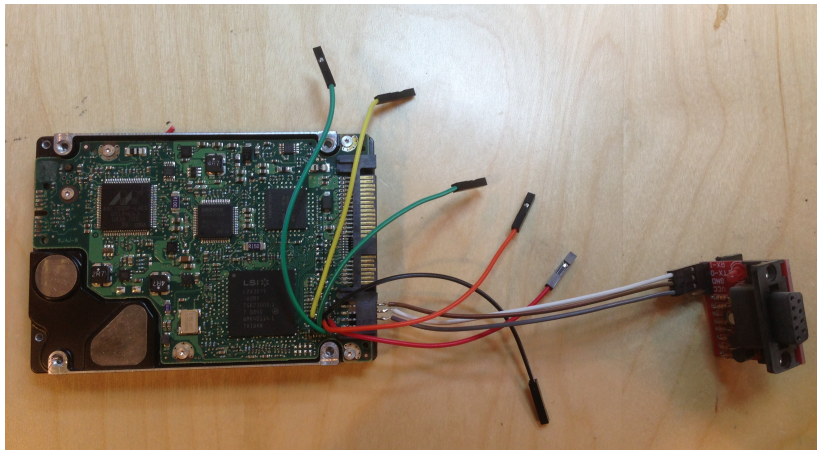
“Implementation and Implications of a Stealth Hard-Drive Backdoor”

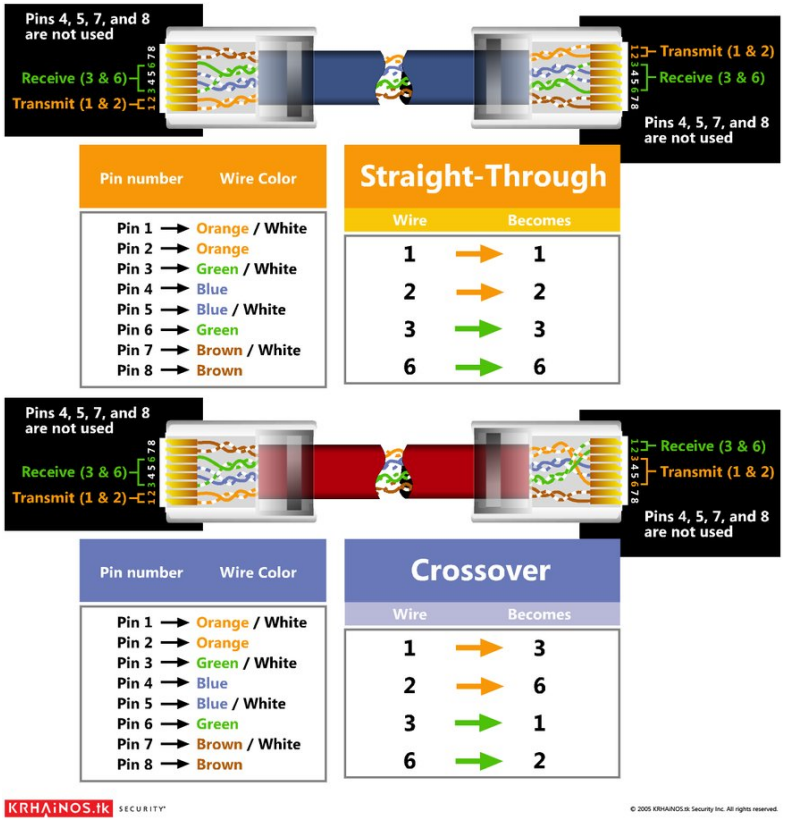


Ethernet Interface

Other Potential Funtennas







Ethernet Interface

UART on HDD

“Implementation and Implications of a Stealth Hard-Drive Backdoor”

irmware Fat Camp!

AUTOTOMIC
BINARY
STRUCTURE
RANDOMIZATION

AUTOTOMY

BINARY

STRUCTURE

RANDOMIZATION

AUTOTOMIC

BINARY STRUCTURE RANDOMIZATION

Idea 1: Remove code that should never execute

Idea 1: Remove code that should never execute

MGCP
VRRP
HSRP
EIGRP
HTTP/S
SIP
SCCP
DHCPServer
Etc, etc…

Idea 1: Remove code that should never execute

MGCP
VRRP
HSRP
EIGRP
HTTP/S
SIP
SCCP
DHCPServer
Etc, etc…

Lawful Intercept

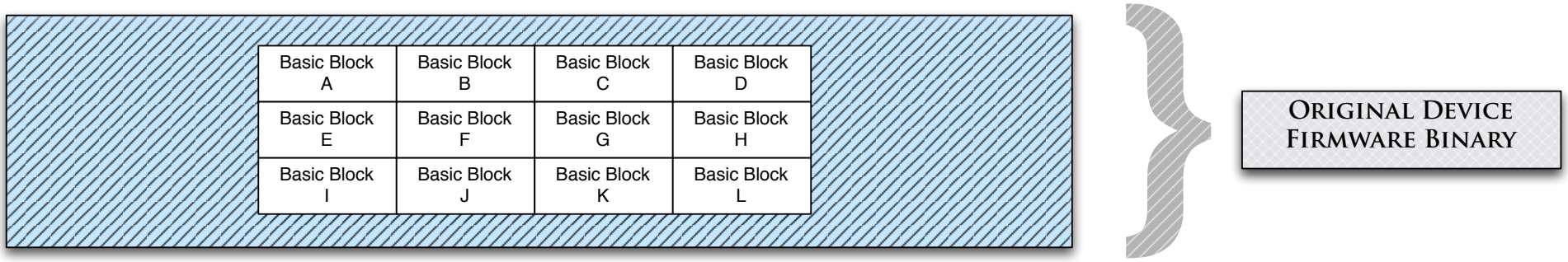
Idea 1: Remove code that should never execute

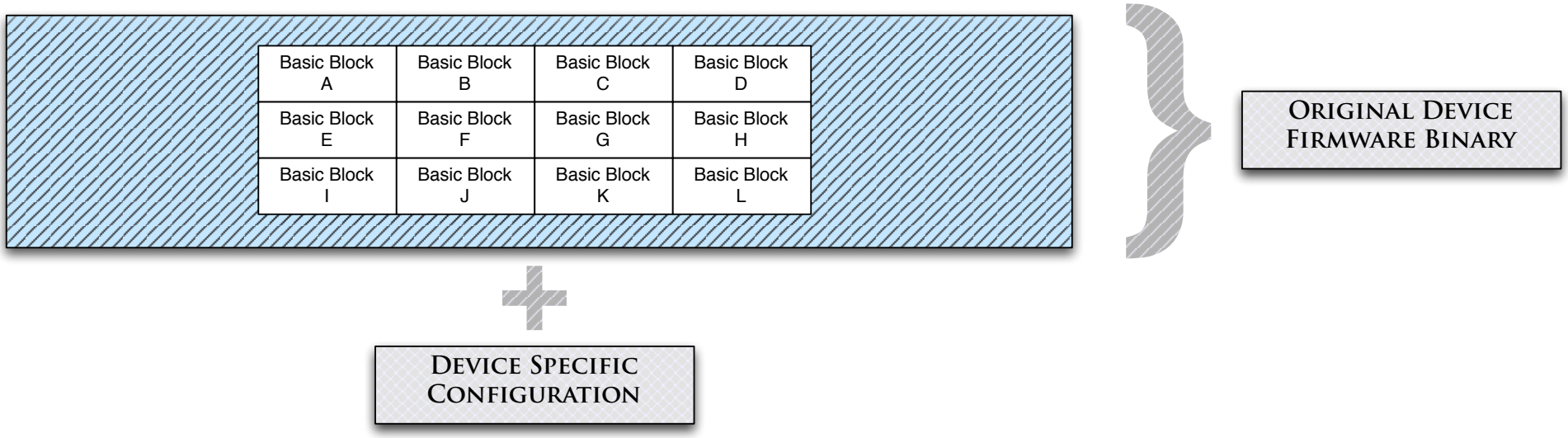
MGCP
VRRP
HSRP
EIGRP
HTTP/S
SIP
SCCP
DHCPServer
Etc, etc…

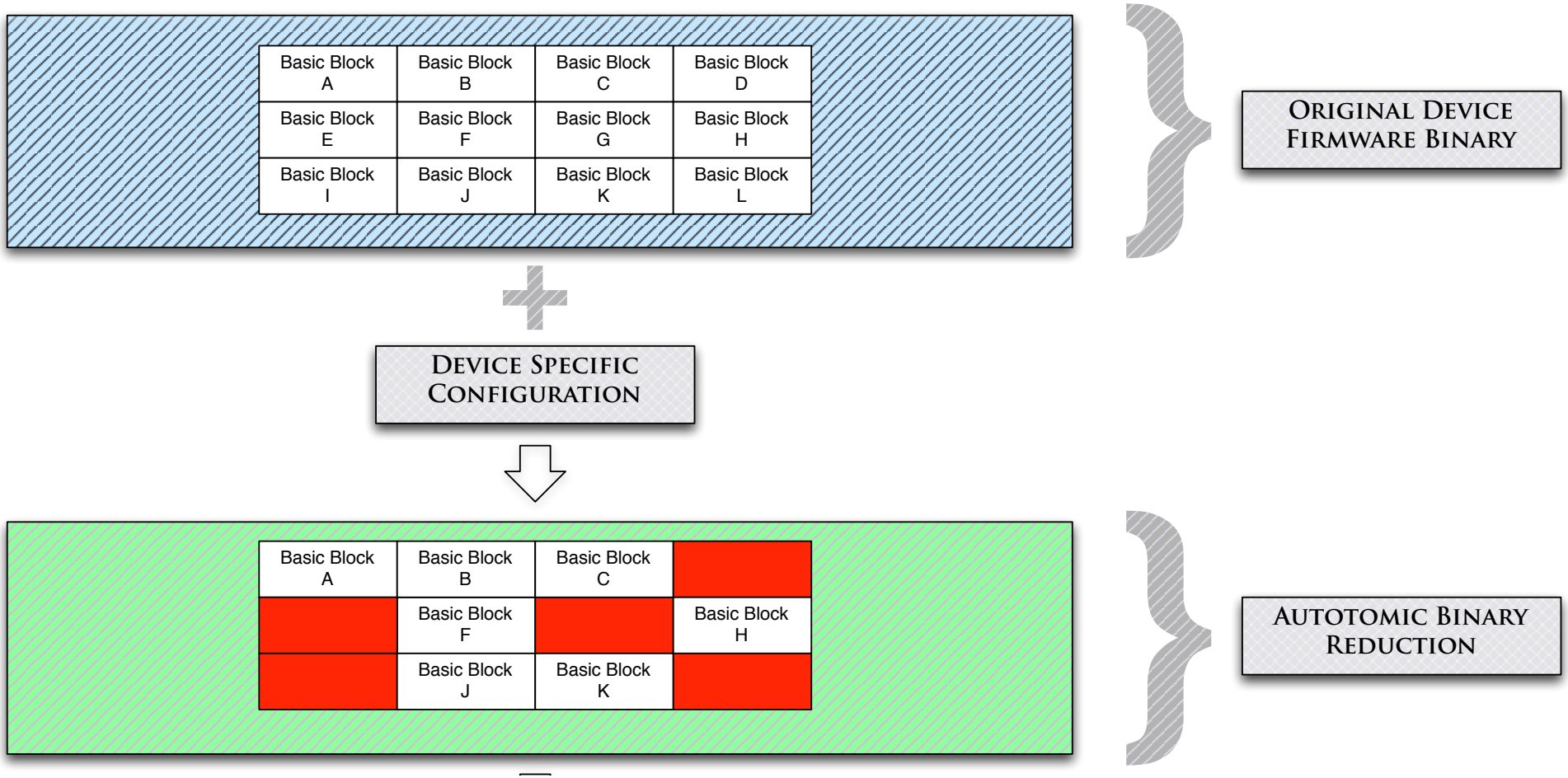
Lawful Intercept

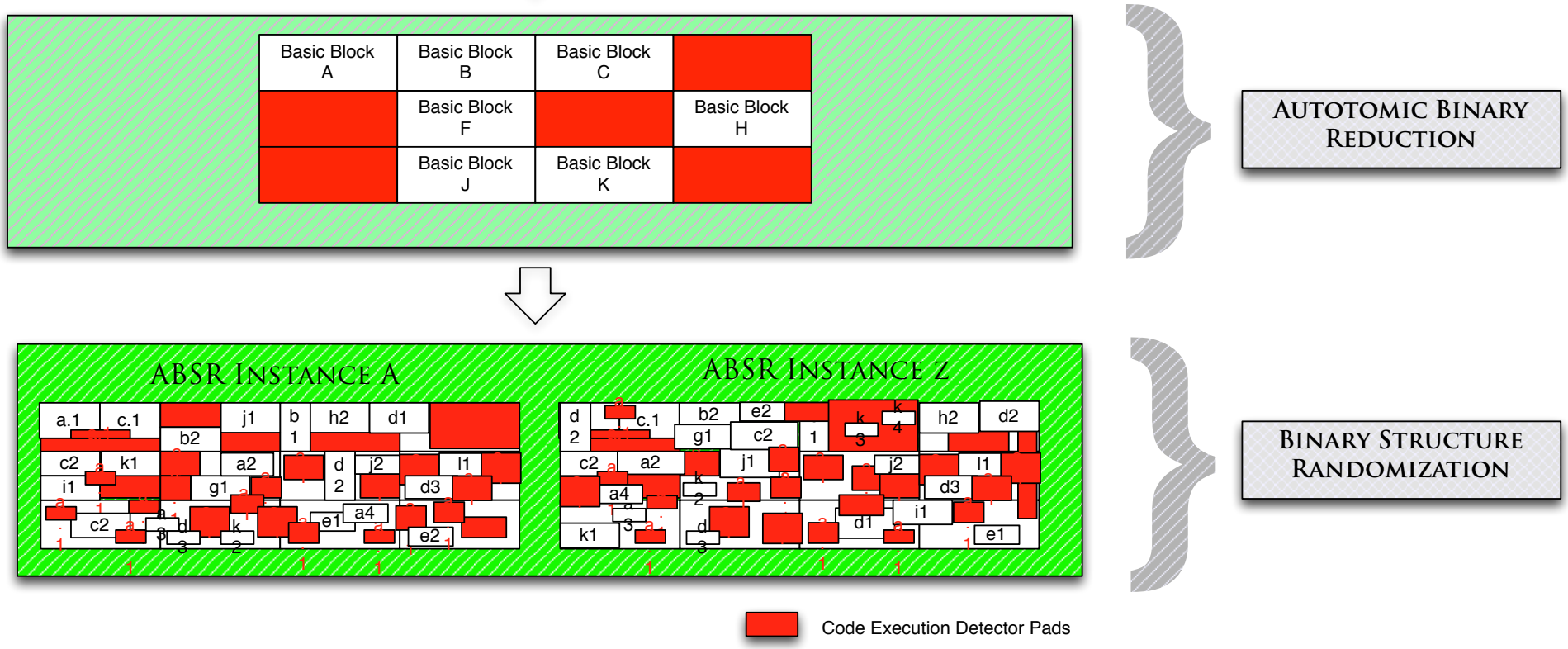
Feature not configured? Remove it from the firmware.

Idea 2: Use “empty space” in firmware for binary randomization


Basic Block A
Basic Block B
Basic Block C
Basic Block D
Basic Block E
Basic Block F
Basic Block G
Basic Block H
Basic Block I
Basic Block J
Basic Block K
Basic Block L
Original Device Firmware Binary


Basic Block A
Basic Block B
Basic Block C
Basic Block D
Basic Block E
Basic Block F
Basic Block G
Basic Block H
Basic Block I
Basic Block J
Basic Block K
Basic Block L
Original Device Firmware Binary
Device Specific Configuration


Basic Block A
Basic Block B
Basic Block C
Basic Block D
Basic Block E
Basic Block F
Basic Block G
Basic Block H
Basic Block I
Basic Block J
Basic Block K
Basic Block L
Original Device Firmware Binary
Device Specific Configuration
Basic Block A
Basic Block B
Basic Block C
Basic Block F
Basic Block H
Basic Block J
Basic Block K
Autotomic Binary Reduction

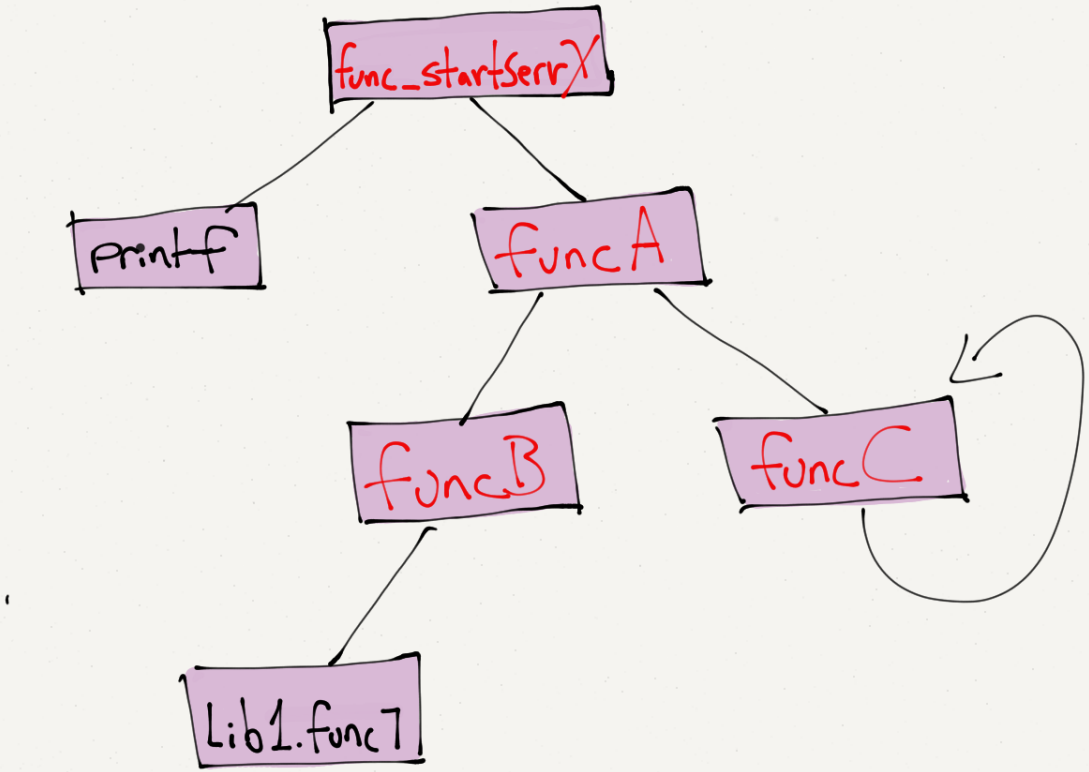

Basic Block A
Basic Block B
Basic Block C
Basic Block F
Basic Block H
Basic Block J
Basic Block K
AUTOTOMIC BINARY REDUCTION
ABSR INSTANCE A
ABSR INSTANCE Z
BINARY STRUCTURE RANDOMIZATION
Code Execution Detector Pads

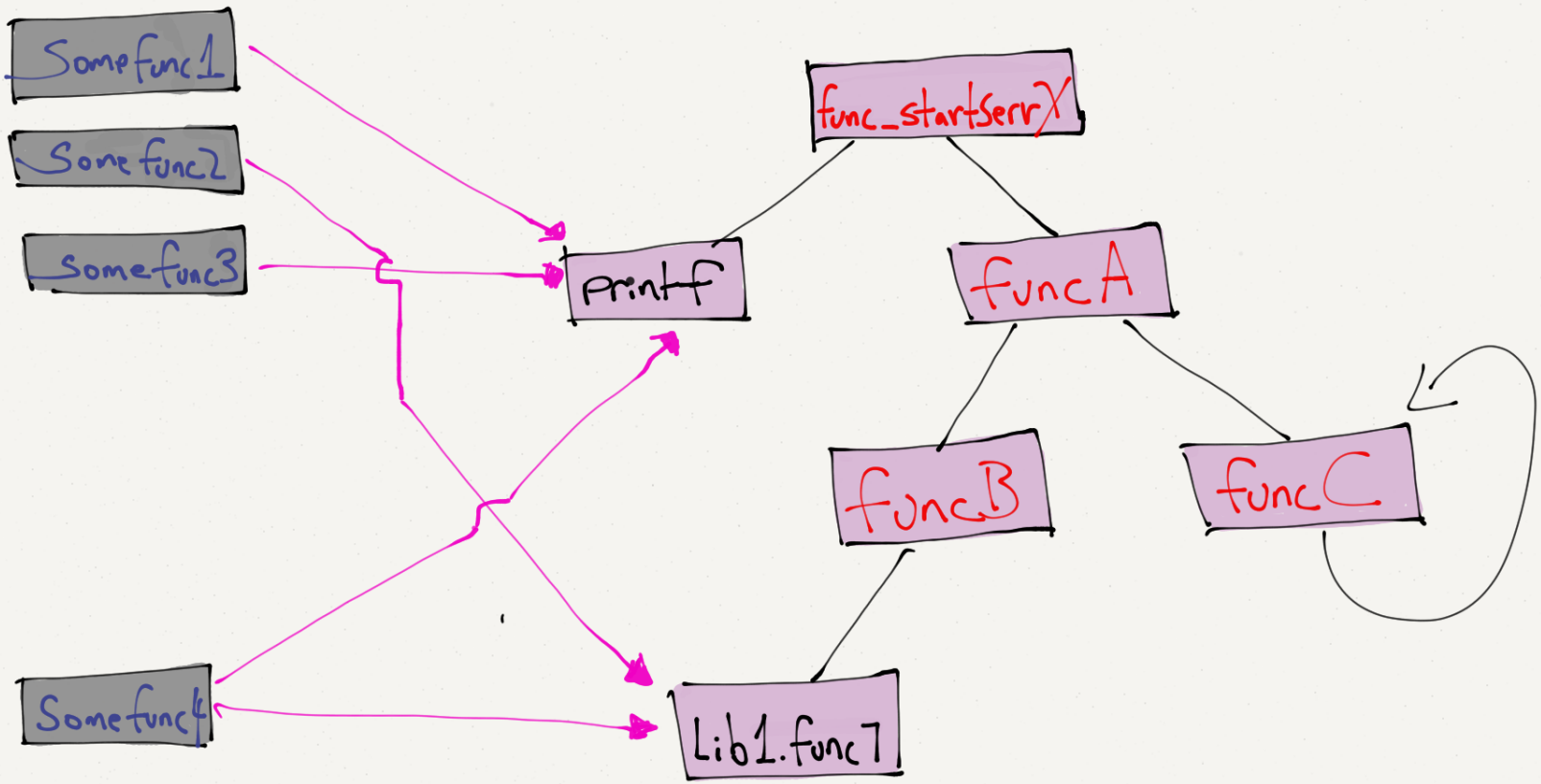
AUTOTOMY

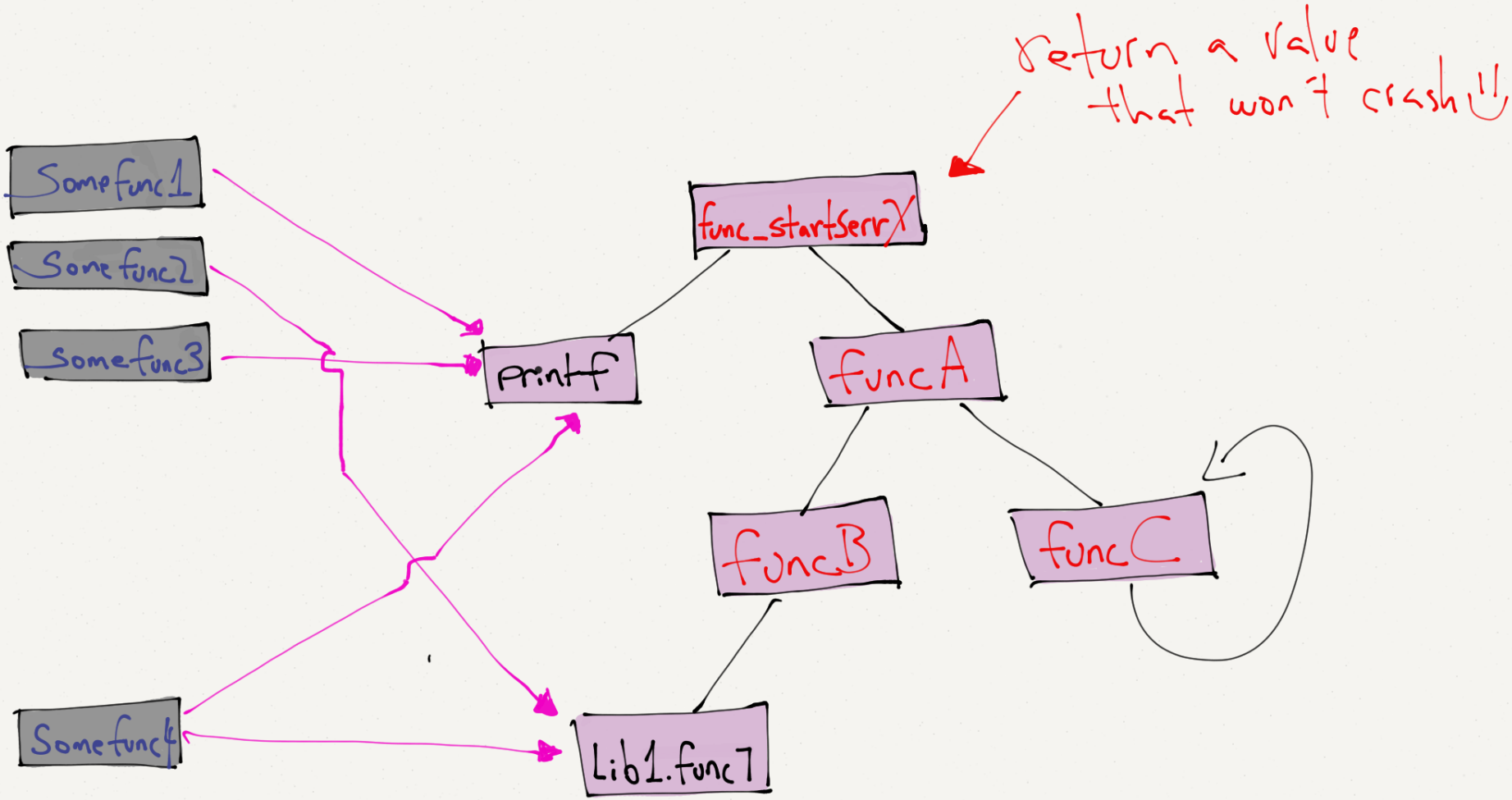
BINARY

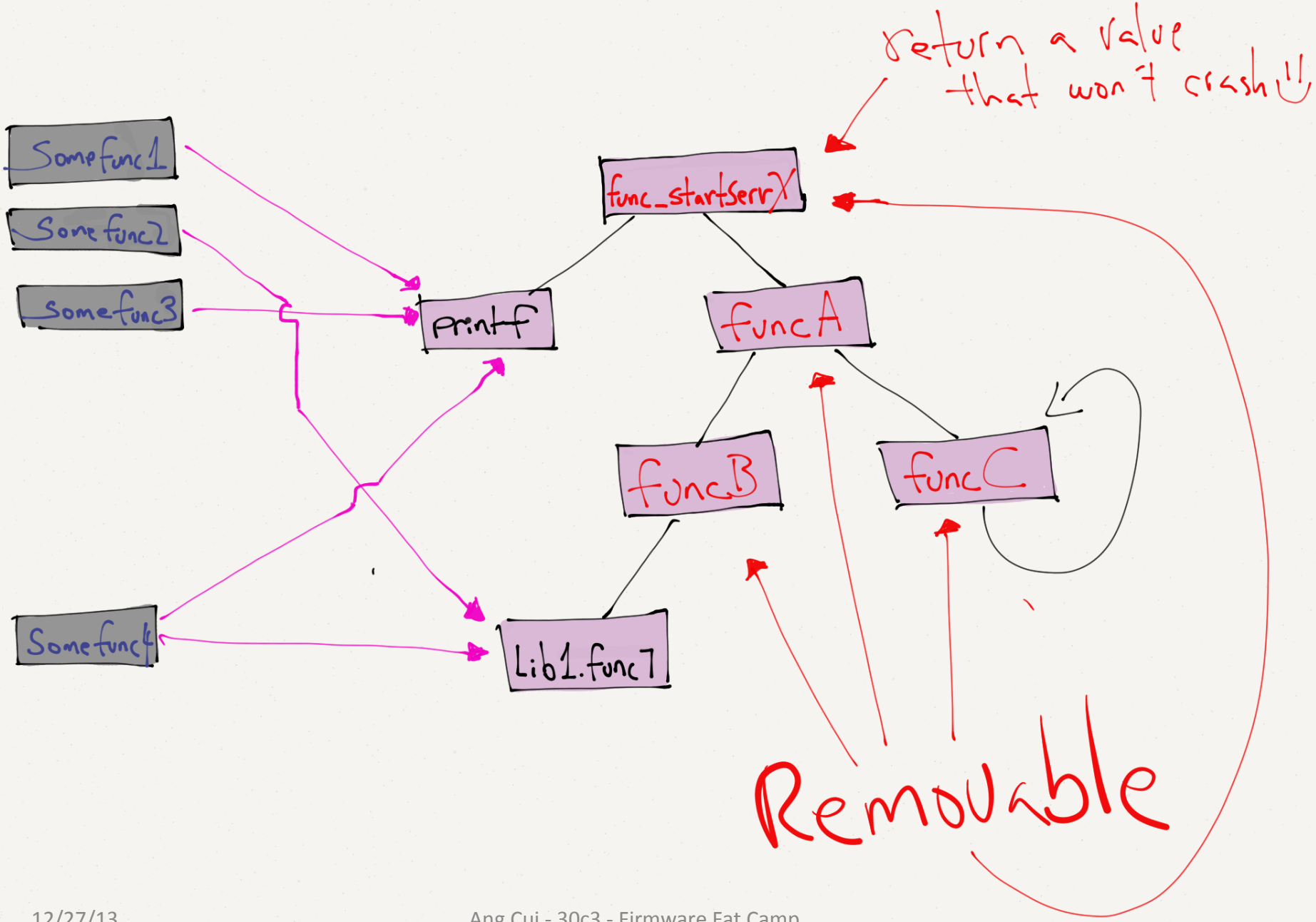
STRUCTURE

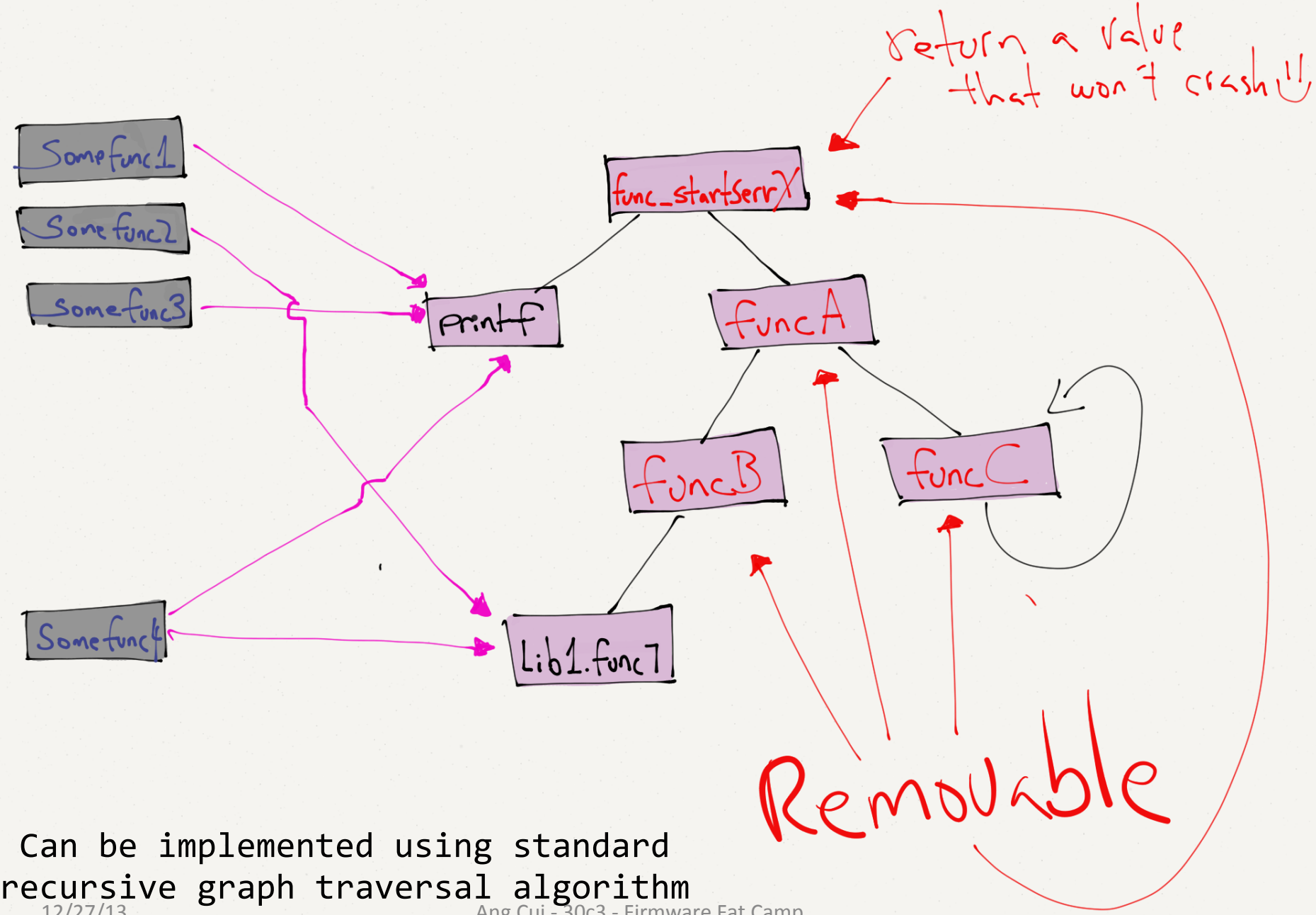
RANDOMIZATION


func_startServer
printf
funcA
funcB
funcC
Lib1.func7


Somefunc1
Somefunc2
Somefunc3
Somefunc4
printf
func_startServ
funcA
funcB
funcC
Lib1.func7

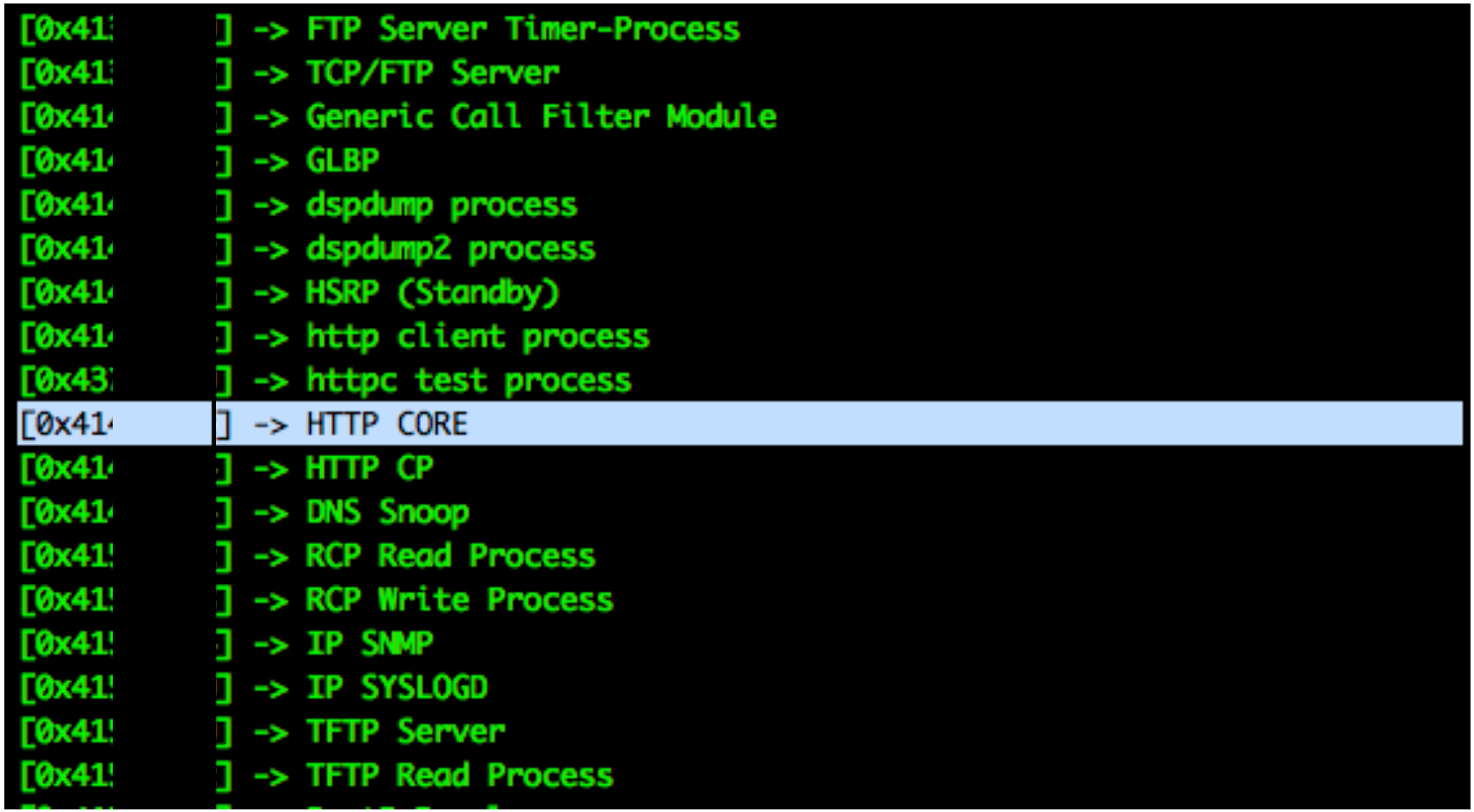

return a value
that won't crash!!
Somefunc1
Somefunc2
Somefunc3
Somefunc4
printf
func_startServX
funcA
funcB
funcC
Lib1.func7


return a value
that won't crash!!
Somefunc1
Somefunc2
Somefunc3
Somefunc4
printf
func_startServX
funcA
funcB
funcC
Lib1.func7
Removable



Can be implemented using standard recursive graph traversal algorithm

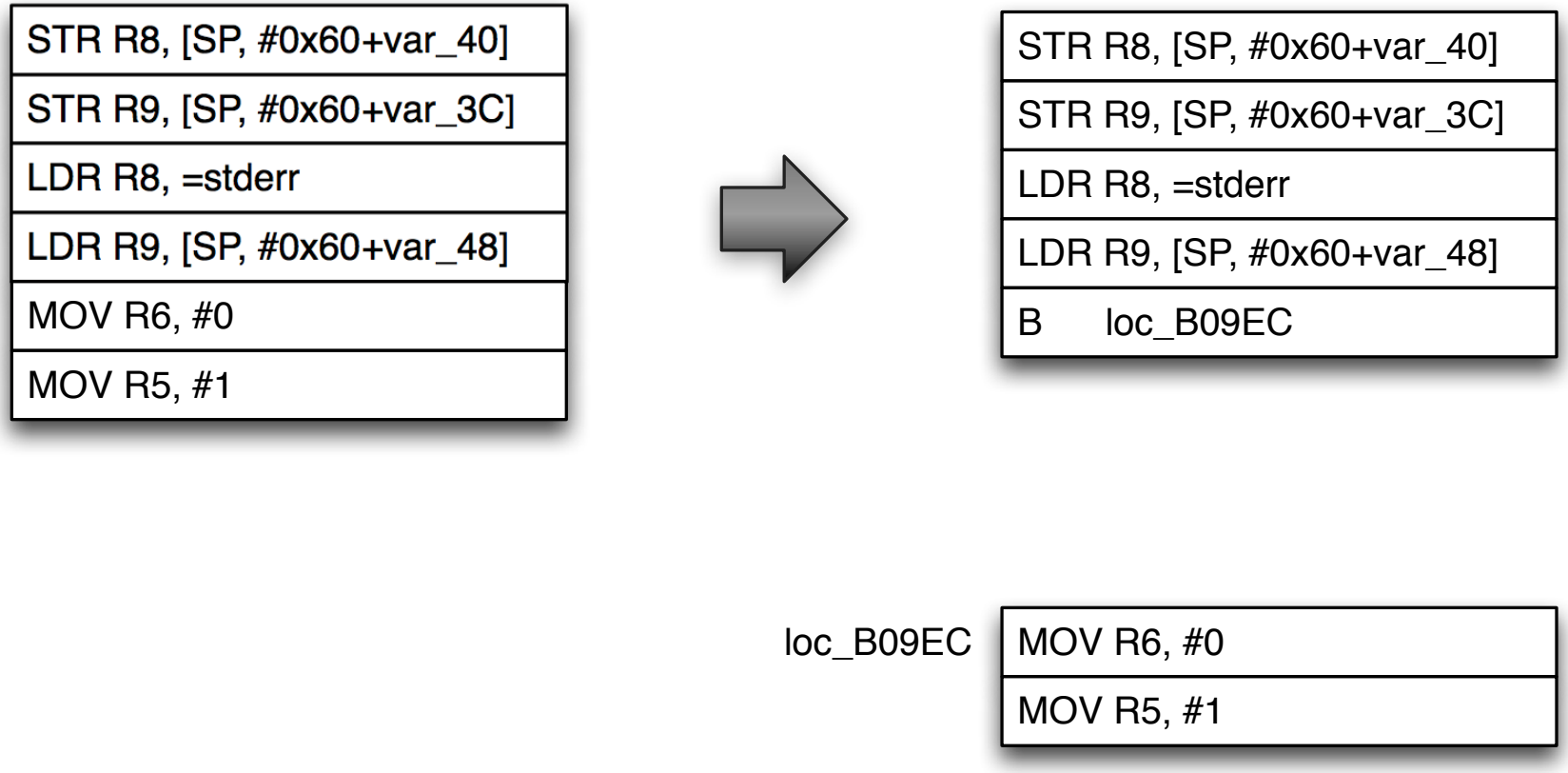
Autotomy: Intractable for the general case, but feasible for most specific cases.



Specific addresses redacted –(

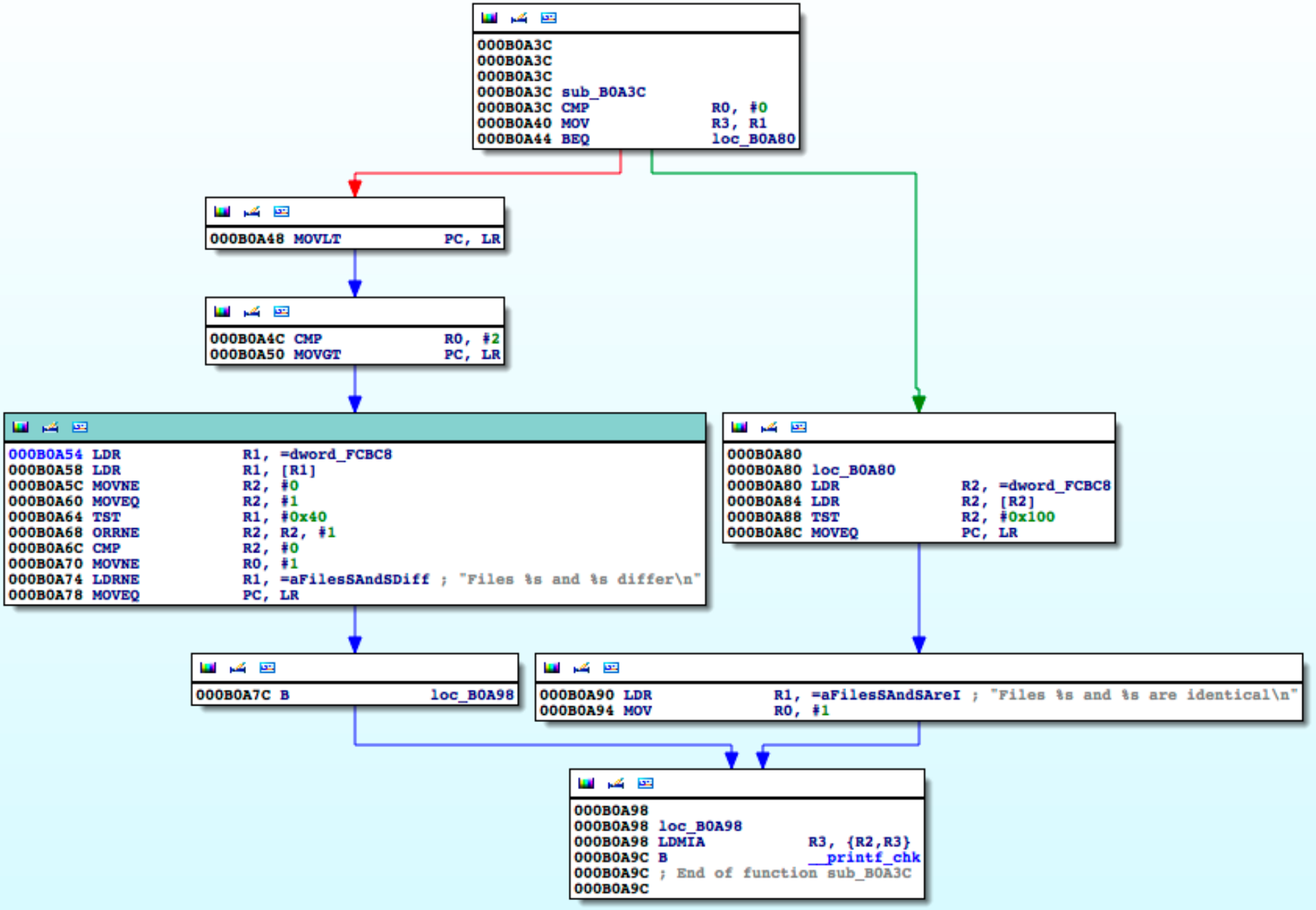
AUTOTOMIC

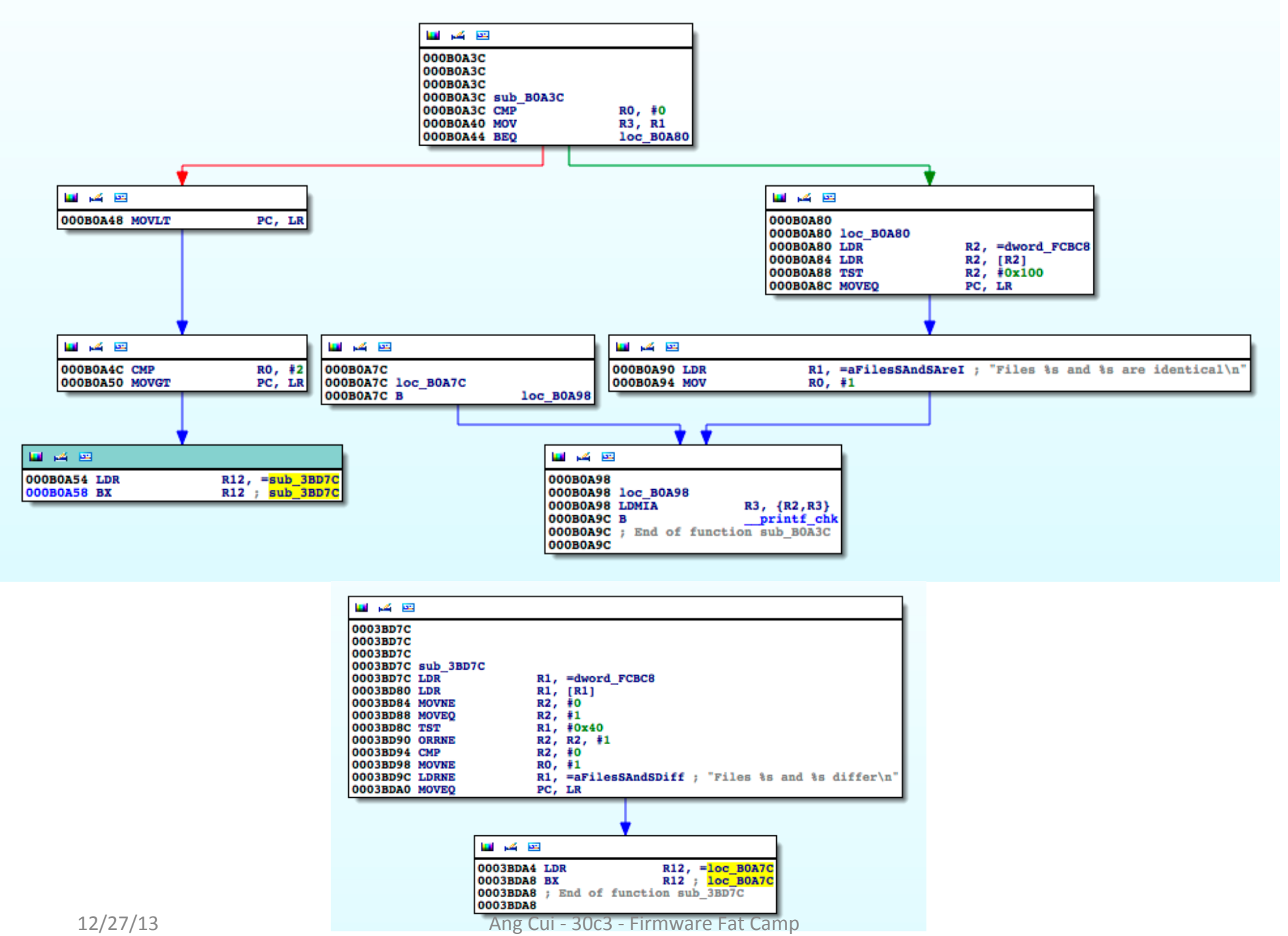
BINARY STRUCTURE RANDOMIZATION



Transform: Basic Block Split

Transform: Basic Block Swap


000B0A3C
000B0A3C
000B0A3C
000B0A3C sub_B0A3C
000B0A3C CMP R0, #0
000B0A40 MOV R3, R1
000B0A44 BEQ loc_B0A80
000B0A48 MOVLT PC, LR
000B0A4C CMP R0, #2
000B0A50 MOVGT PC, LR
000B0A54 LDR R1, =dword_FCBC8
000B0A58 LDR R1, [R1]
000B0A5C MOVNE R2, #0
000B0A60 MOVEQ R2, #1
000B0A64 TST R1, #0x40
000B0A68 ORRNE R2, R2, #1
000B0A6C CMP R2, #0
000B0A70 MOVNE R0, #1
000B0A74 LDRNE R1, =aFilesSAndSDiff ; "Files %s and %s differ\n"
000B0A78 MOVEQ PC, LR
000B0A80
000B0A80 loc_B0A80
000B0A80 LDR R2, =dword_FCBC8
000B0A84 LDR R2, [R2]
000B0A88 TST R2, #0x100
000B0A8C MOVEQ PC, LR
000B0A7C B loc_B0A98
000B0A90 LDR R1, =aFilesSAndSAreI ; "Files %s and %s are identical\n"
000B0A94 MOV R0, #1
000B0A98
000B0A98 loc_B0A98
000B0A98 LDMIA R3, {R2,R3}
000B0A9C B __printf_chk
000B0A9C ; End of function sub_B0A3C
000B0A9C

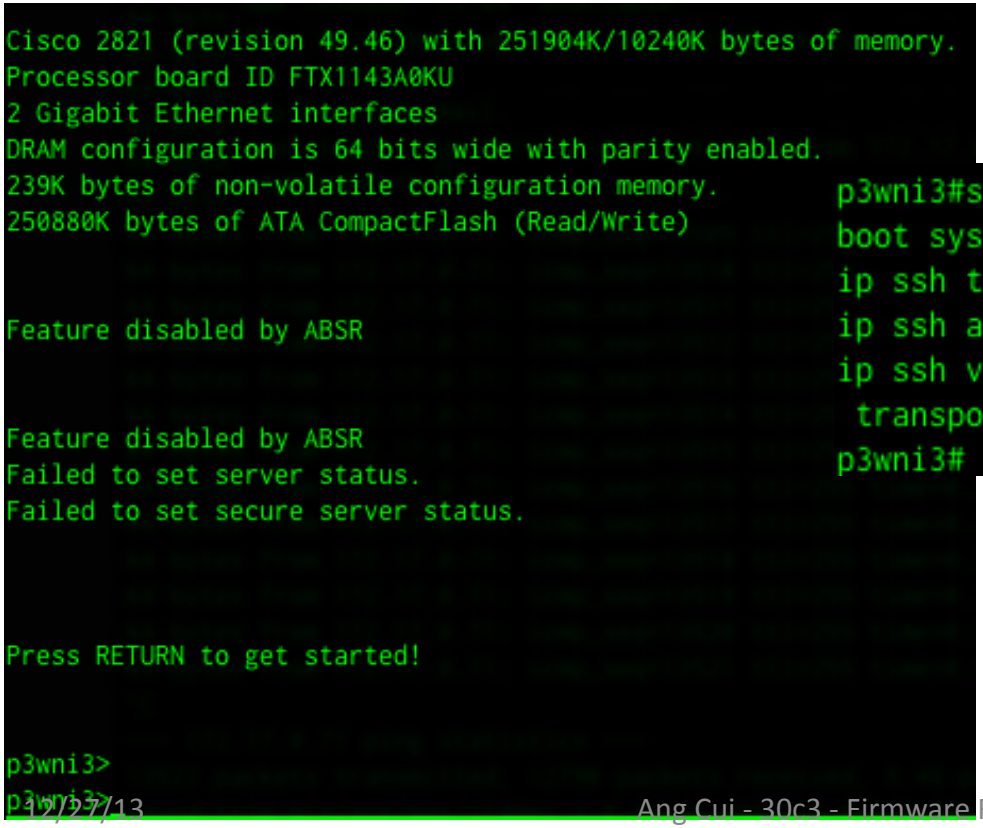

000B0A3C
000B0A3C
000B0A3C
000B0A3C sub_B0A3C
000B0A3C CMP R0, #0
000B0A40 MOV R3, R1
000B0A44 BEQ loc_B0A80
000B0A48 MOVLT PC, LR
000B0A80
000B0A80 loc_B0A80
000B0A80 LDR R2, =dword_FCBC8
000B0A84 LDR R2, [R2]
000B0A88 TST R2, #0x100
000B0A8C MOVEQ PC, LR
000B0A4C CMP R0, #2
000B0A50 MOVGT PC, LR
000B0A7C
000B0A7C loc_B0A7C
000B0A7C B loc_B0A98
000B0A90 LDR R1, =aFilesSAndSAreI ; "Files %s and %s are identical\n"
000B0A94 MOV R0, #1
000B0A54 LDR R12, =sub_3BD7C
000B0A58 BX R12 ; sub_3BD7C
000B0A98
000B0A98 loc_B0A98
000B0A98 LDMIA R3, {R2,R3}
000B0A9C B __printf_chk
000B0A9C ; End of function sub_B0A3C
000B0A9C
0003BD7C
0003BD7C
0003BD7C
0003BD7C sub_3BD7C
0003BD7C LDR R1, =dword_FCBC8
0003BD80 LDR R1, [R1]
0003BD84 MOVNE R2, #0
0003BD88 MOVEQ R2, #1
0003BD8C TST R1, #0x40
0003BD90 ORRNE R2, R2, #1
0003BD94 CMP R2, #0
0003BD98 MOVNE R0, #1
0003BD9C LDRNE R1, =aFilesSAndSDiff ; "Files %s and %s differ\n"
0003BDA0 MOVEQ PC, LR
0003BDA4 LDR R12, =loc_B0A7C
0003BDA8 BX R12 ; loc_B0A7C
0003BDA8 ; End of function sub_3BD7C
0003BDA8

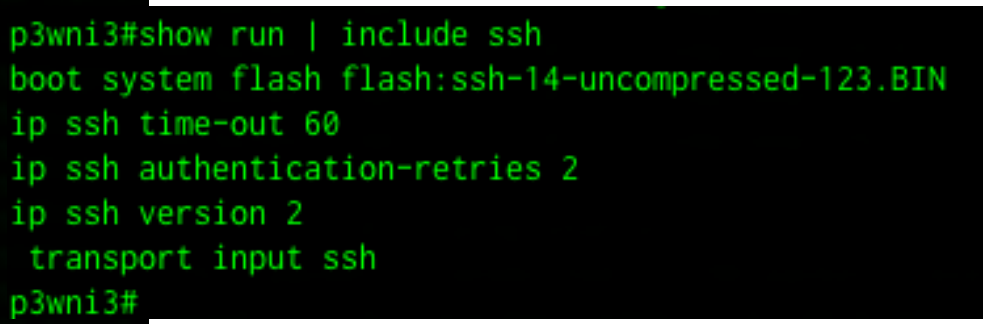
Demo

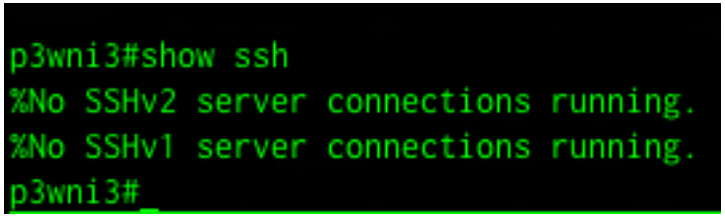
- Removing HTTP & SSH servers from Cisco IOS
- Cisco 1841, IOS 12.4

Demo

- Removing HTTP & SSH servers from Cisco IOS



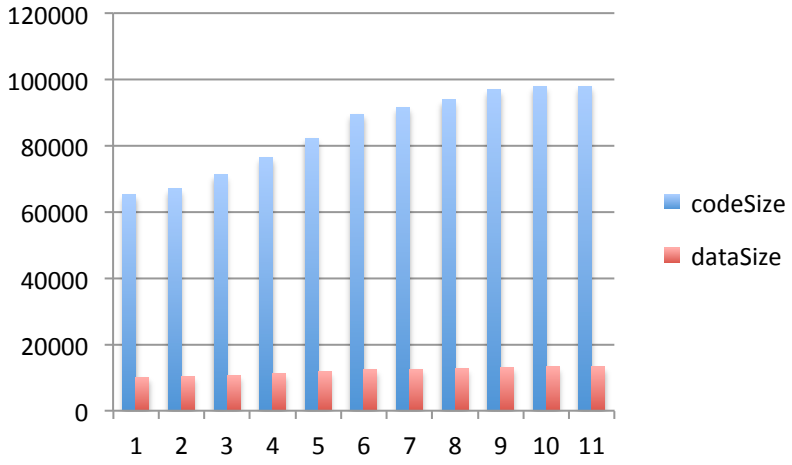




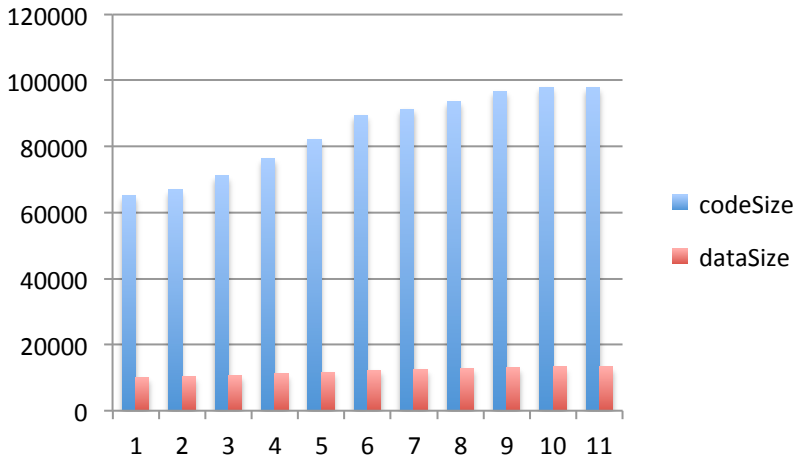
Demo

```
p3wni3#sh ip http server status
HTTP server status: Disabled
HTTP server port: 80
HTTP server authentication method: enable
HTTP server access class: 0
HTTP server base path:
Maximum number of concurrent server connections allowed: 5
Server idle time-out: 180 seconds
Server life time-out: 180 seconds
Maximum number of requests allowed on a connection: 1
HTTP secure server capability: Present
HTTP secure server status: Disabled
HTTP secure server port: 443
HTTP secure server ciphersuite: 3des-ede-cbc-sha des-cbc-sha rc4-128-md5 rc4-128-sha
HTTP secure server client authentication: Disabled
HTTP secure server trustpoint:

p3wni3#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
p3wni3(config)#ip http server
Failed to set server status.

p3wni3(config)#end
p3wni3#sh ip http server status
HTTP server status: Disabled
HTTP server port: 80
HTTP server authentication method: enable
HTTP server access class: 0
HTTP server base path:
Maximum number of concurrent server connections allowed: 5
Server idle time-out: 180 seconds
Server life time-out: 180 seconds
Maximum number of requests allowed on a connection: 1
HTTP secure server capability: Present
HTTP secure server status: Disabled
HTTP secure server port: 443
HTTP secure server ciphersuite: 3des-ede-cbc-sha des-cbc-sha rc4-128-md5 rc4-128-sha
HTTP secure server client authentication: Disabled
HTTP secure server trustpoint:
p3wni3#
```

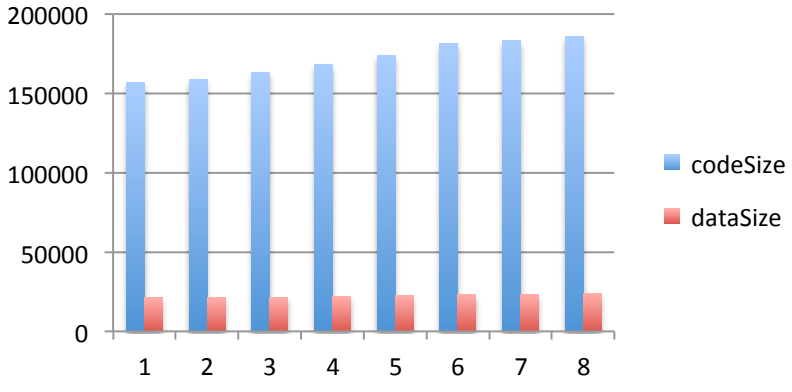
Binary Autotomy: Cisco IOS Features



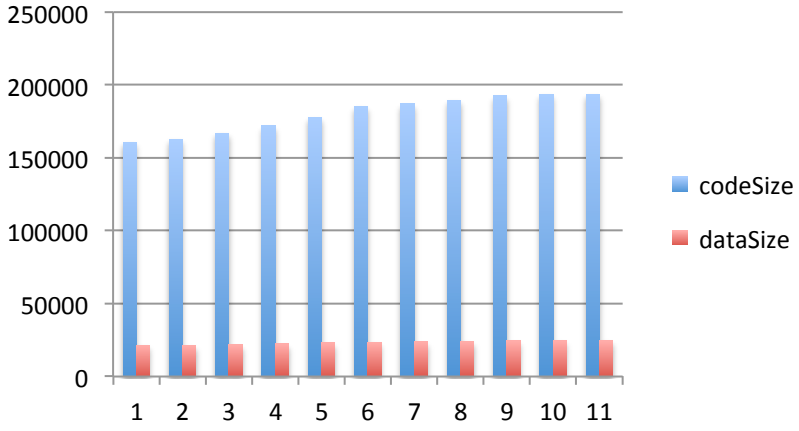
HTTP



HTTP | SSH

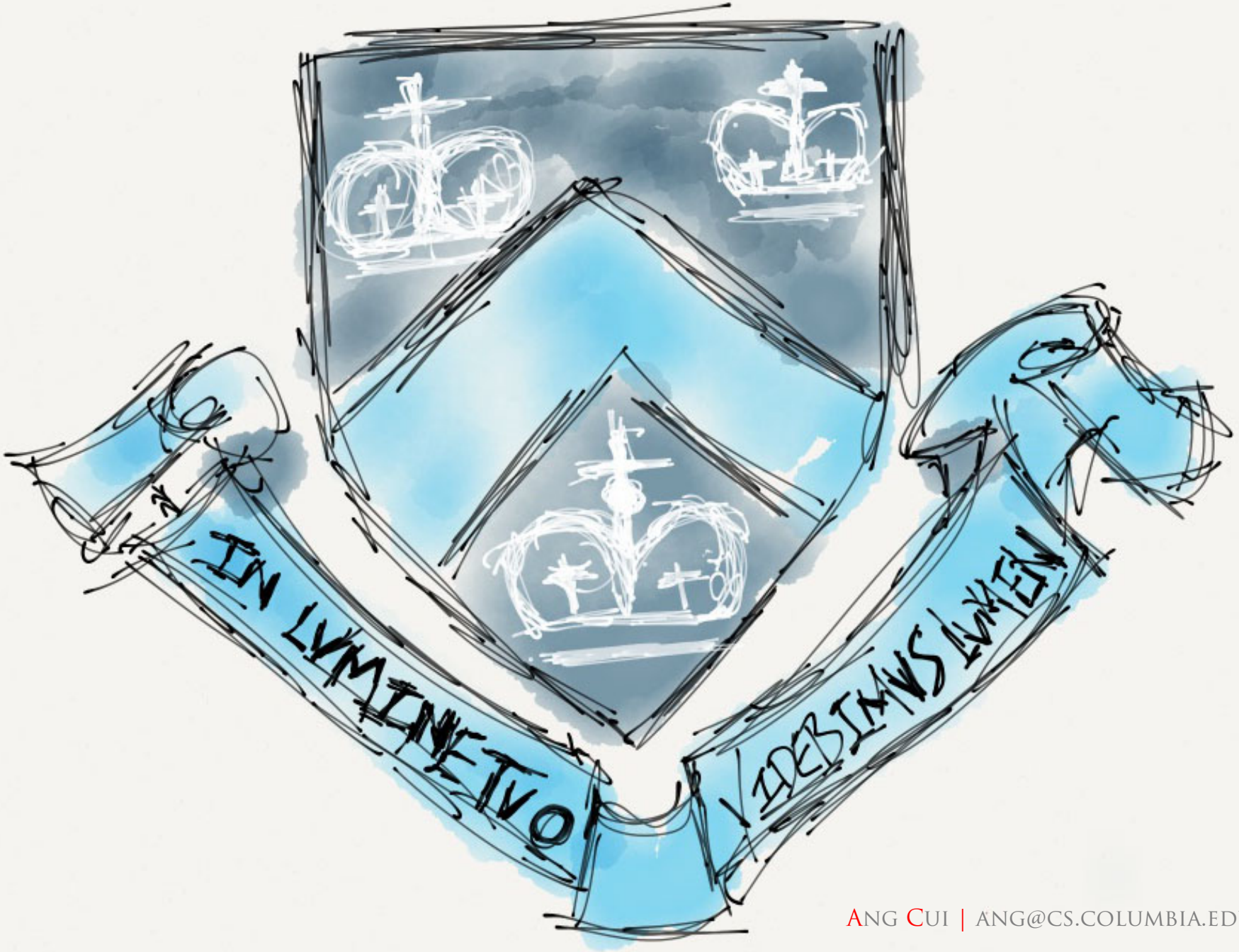


HTTP|SSH|X.25|ATM



HTTP|SSH|X.25|ATM|SCP

X-axis: Levels of recursion
Y-axis: Bytes removed



ANG CUI | ANG@CS.COLUMBIA.EDU