

# Wireless Hacking with HackCUBE & HackCUBE-Special



Jie Fu KunZhe Chai  
From PegasusTeam

# Brief introduction of HackCUBE

## UNICORNCUBE 无限可能性之“盒”

### MouseJack

MouseJack is a class of vulnerabilities that affects the vast majority of wireless, non-Bluetooth keyboards and mice.

### BadUSB

BadUSB is a keystroke injection tool disguised as a generic flash drive.

### RfCat

RfCat USB Radio Dongle is custom hardware designed for use with the custom RfCat firmware written by Atlas.



### Aircrack-NG

Aircrack-ng is a complete suite of tools to assess WiFi network security.

### Kismet

Kismet is a network detector, packet sniffer, and intrusion detection system for 802.11 wireless LANs.

### Mfoc

MFOC is an open source implementation of "offline nested" attack by Nethemba.

## KALI & RASPBERRY PI & ARDUINO

将“攻防零件”重新组合



(2x) Dual-Band (2.4/5 GHz) 802.11 a/b/g/n/ac Radios (BCM43438 and RTL8822BU Chipsets)

(2x) Bluetooth BCM43438(V2.1+EDR/v3.0/v3.0+HS/v4.1) and RTL8822BU(4.1 + HS)

(2x) Dual-Band (125Khz/13.56Mhz)NFC/RFID Radios (EM4095 and PN532 Chipsets)

(2x) Dual-Band(433/315Mhz) Sub-GHz Radios (CC1101 Chipsets)

(2x) ATmega32u4/BCM2835 1Ghz 512MB RAM

2.4GHZ Radio transceiver (nRF24L01 Chipsets)

(3x) USB Host Port (USB-HID and 2x USB 2.0)

500W Pixels Camera module



WIFI



BLUETOOTH



NFC



RF



HID



CAMERA



LED



BUZZER



MICROPHONE



BATTERY

# 防御

抵御无线射频攻击  
根据频谱仪溯源恶意干扰源  
有效防御汽车中继攻击  
可阻断未知射频信号









# Two specific attack cases



Fixed-code brute force attack to parking bar

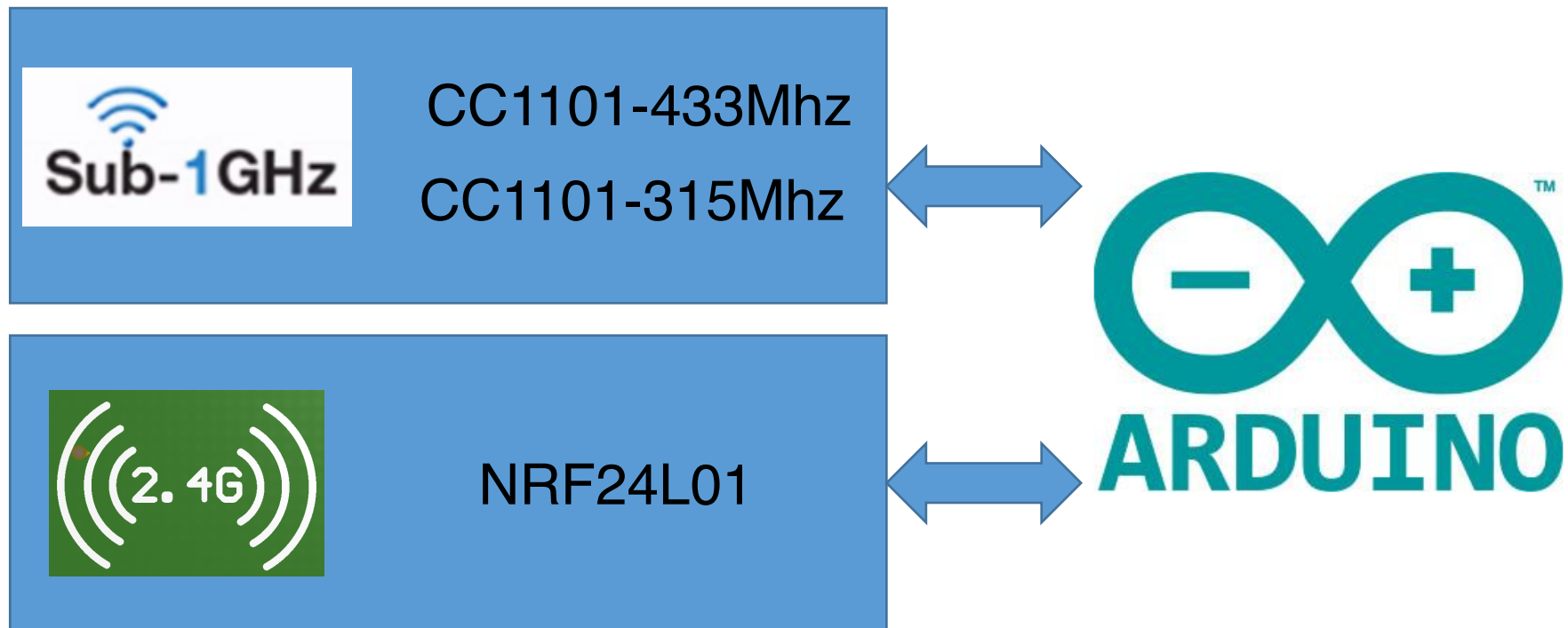


Attack to Entrance guard system

# Example 1: Fixed-code brute force attack to parking bar



# Resources of HackCUBE for attacking Sub-1Ghz & 2.4GHz

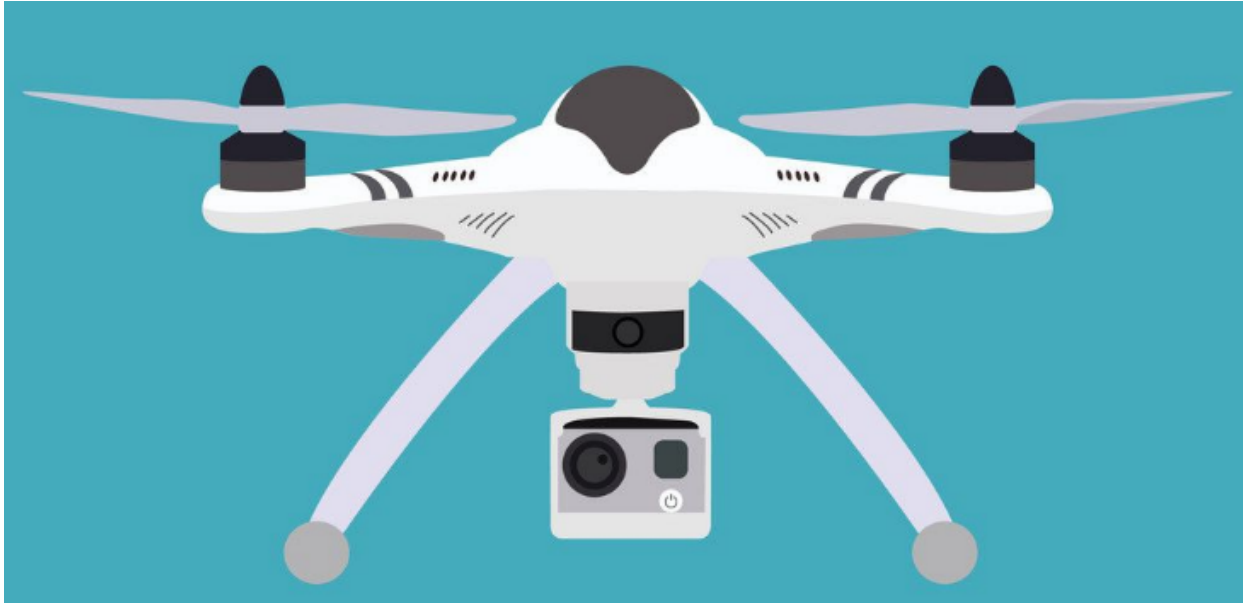




# Sub-1GHz radio usage in our daily life



# 2.4GHz Radio usage in our daily life



# Basic knowledge of Remote Keyless Entry

## 1. Fixed code remote control

- Send same data every time
- Data is not encrypted
- Widely used in
  - Safety Guard System
  - Smart Home System

## ~~2. Rolling code remote control~~

- ~~➤ Send rolling data~~
- ~~➤ Data is encrypted~~
- ~~➤ Widely used in Automobile entrance guard system~~

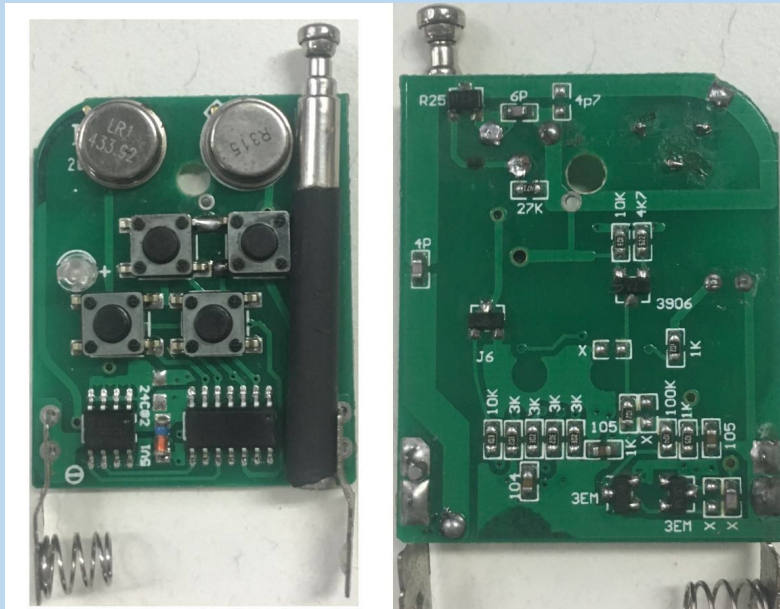
If you are interested in the Rolling code remote control, please refer to <https://www.youtube.com/watch?v=p3SJP-7LSNs&t=2807s>



# Two types of Fixed code remote control

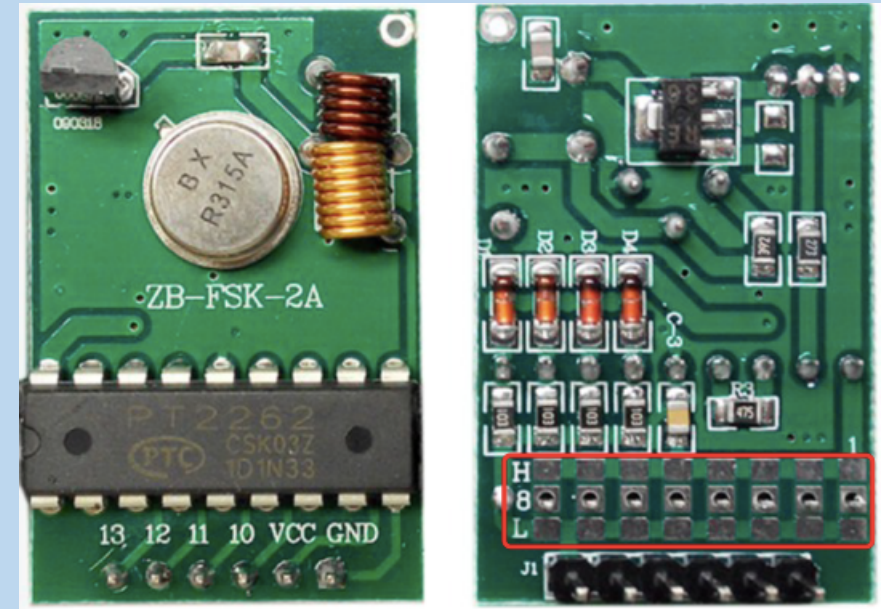
## 1. Changeless coding

address is fixed by the Semiconductor manufacturer



## 2. Changeable coding

can change address by soldering to 3 different states





# How to attack the parking bar?

In fact , the parking bar system is a fixed-code remote control.

## **Method 1:**

~~Sniff the signal when the guard control the parking bar, then replay it using the HackCUBE or any other SDR tools.~~

## **Method 2:**

Reverse analysis signal, then forge all the data.

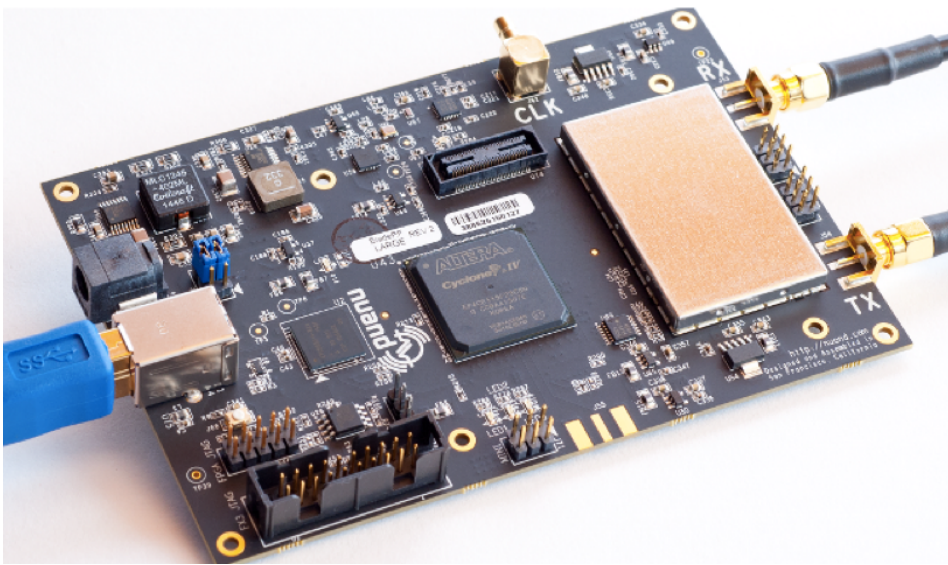
# How to find the operating frequency?



HackRF USRP

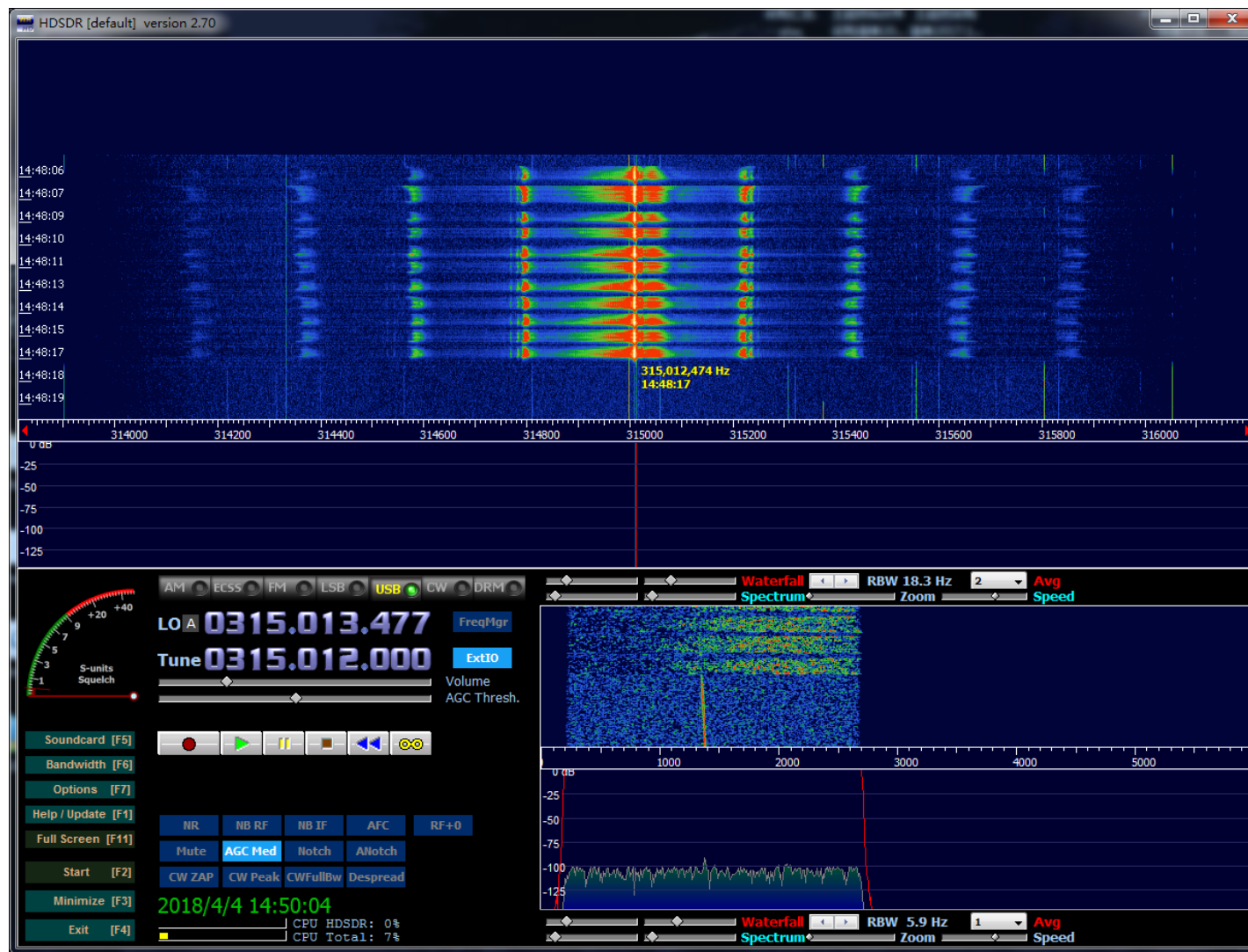


BladeRF RTL2832U





The frequency is 315MHz



# Analyzing the signal--Changeless coding

The original signal



The signal can be divided into 4 parts

**Preamble:** used by receiver to sync with the transmitter

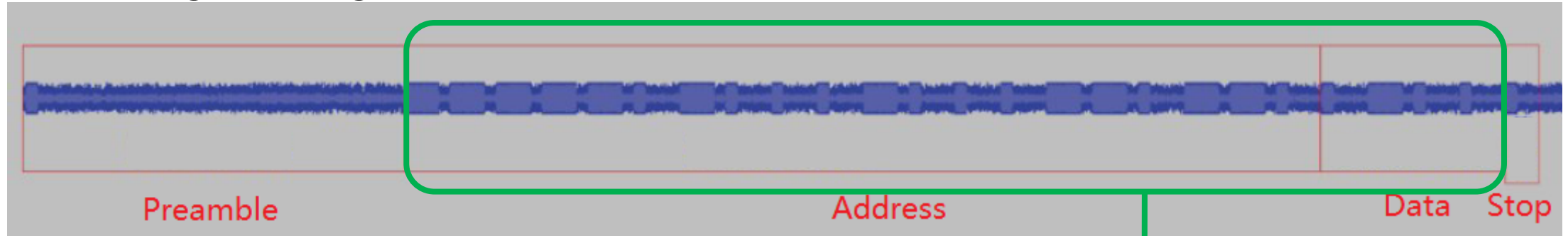
**Address:** Identification code

**Data:** function code

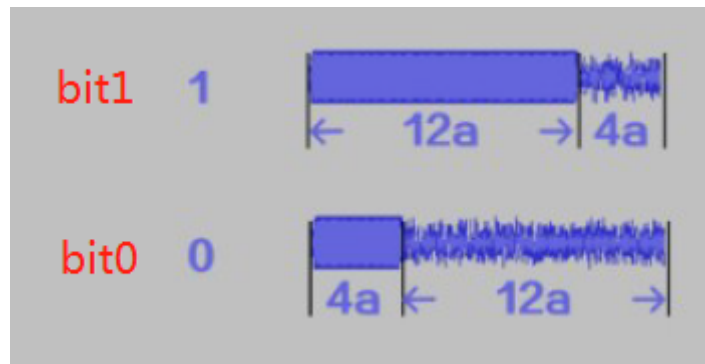
**Stop:** stop bit

# Decoding the signal--Changeless coding

The original signal



Coding format



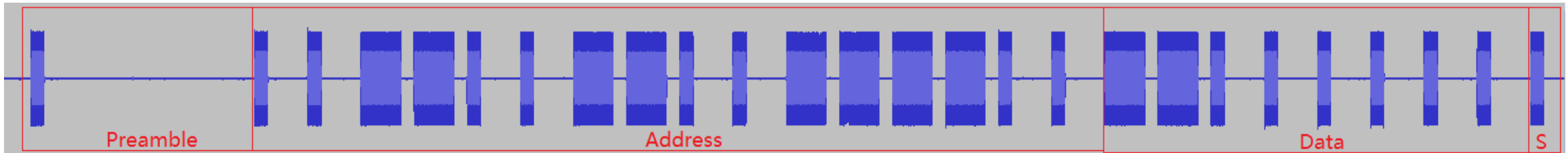
Decoded data





# Analyzing the signal--Changeable coding

The original signal



The signal can be divided into 4 parts:

**Preamble:** used by receiver to sync with the transmitter

**Address:** Identification code

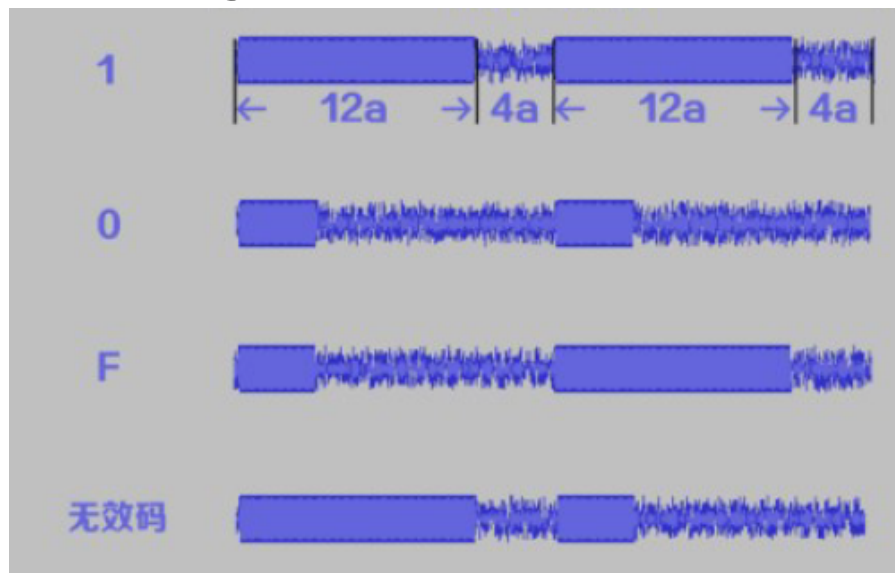
**Data:** function code

**Stop:** stop bit It is same as the changeless code.

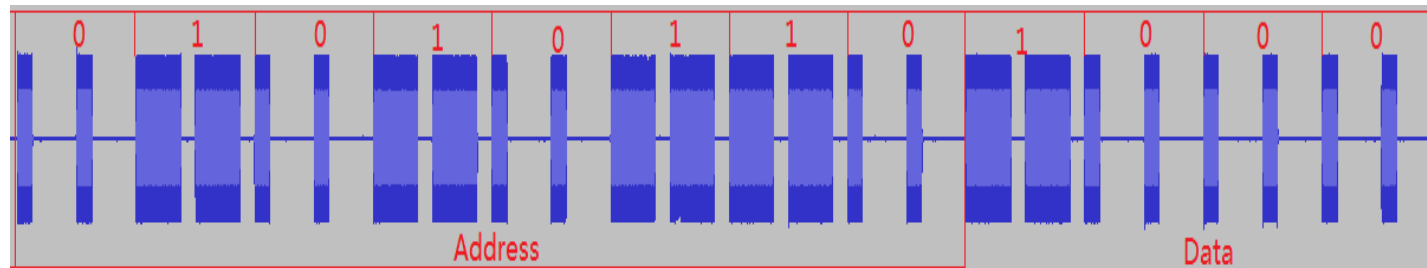
# Decoding the signal--Changeable cod



## Coding format



## Decoded data



# Changeable coding VS changeless coding

## Changeable coding

1. 8 bit address & 3 states  
total numbers  $3^8=6551$
2. It take about 10 minute to  
carry out brute force attack

## Changeless coding

1. 20 bit address & 2 states  
total numbers  $2^{20}=1048576$
2. It is not easy to carry  
out brute force attack

# Get your hands dirty;)

Following the steps

Step1: Plug the MicroUSB to power the HackCUBE

Step2: Power the LEGO-based parking bar model

Step3: Connect to the AP of the HackCUBE

SSID: HackCUBE\_XX:XX:XX (MAC address)

key: hackcube123

Step 4: open the browser, enter 192.168.2.3

Step5: select the RF tab

Step6: click the attack in bottom of this web



# Example 2: attacking RFID



# Resources of HackCUBE for attacking RFID



**transceiver module for contactless communication at 13.56 MHz, 6 different operating modes:**

- 1.ISO/IEC 14443A/MIFARE® Reader/Writer
- 2.FeliCa Reader/Writer
- 3.ISO/IEC 14443B Reader/Writer
- 4.ISO/IEC 14443A/MIFARE Card MIFARE Classic® 1K or MIFARE Classic 4K card emulation mode
- 5.FeliCa Card emulation

**Read/Write analog front end for 125kHz RFID**

Multiple transponder protocol compatibility (Ex: EM4102, EM4200, EM4450 and EM4205/EM4305)

# RFID usage in our daily life



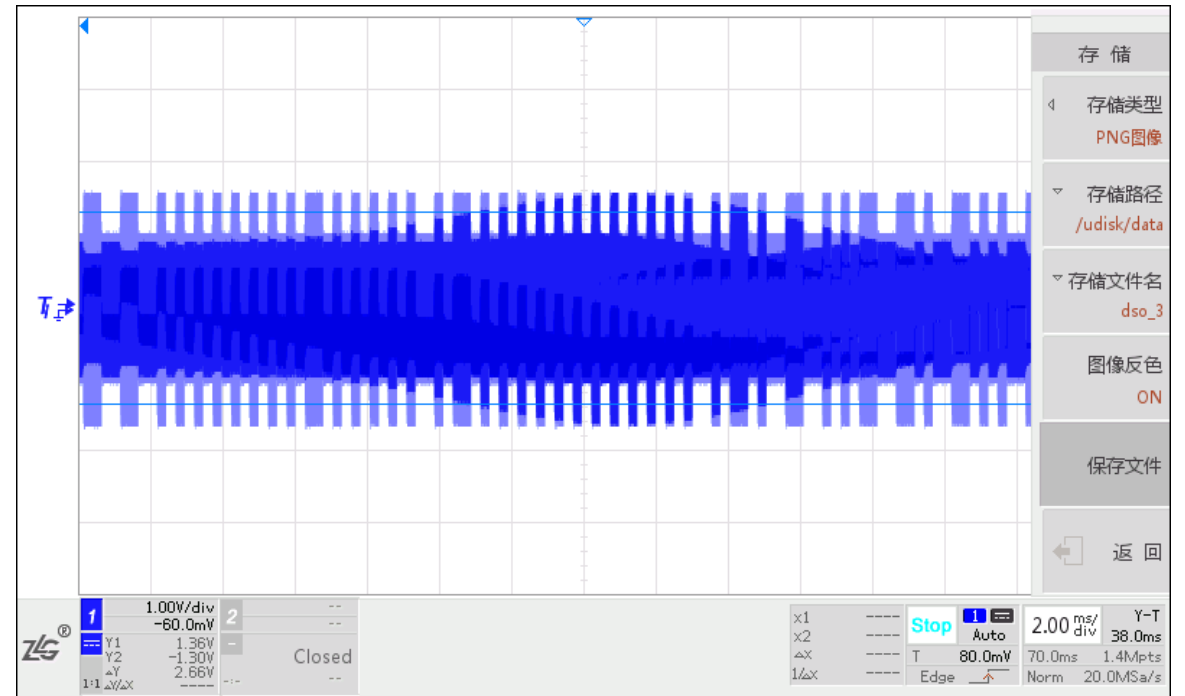
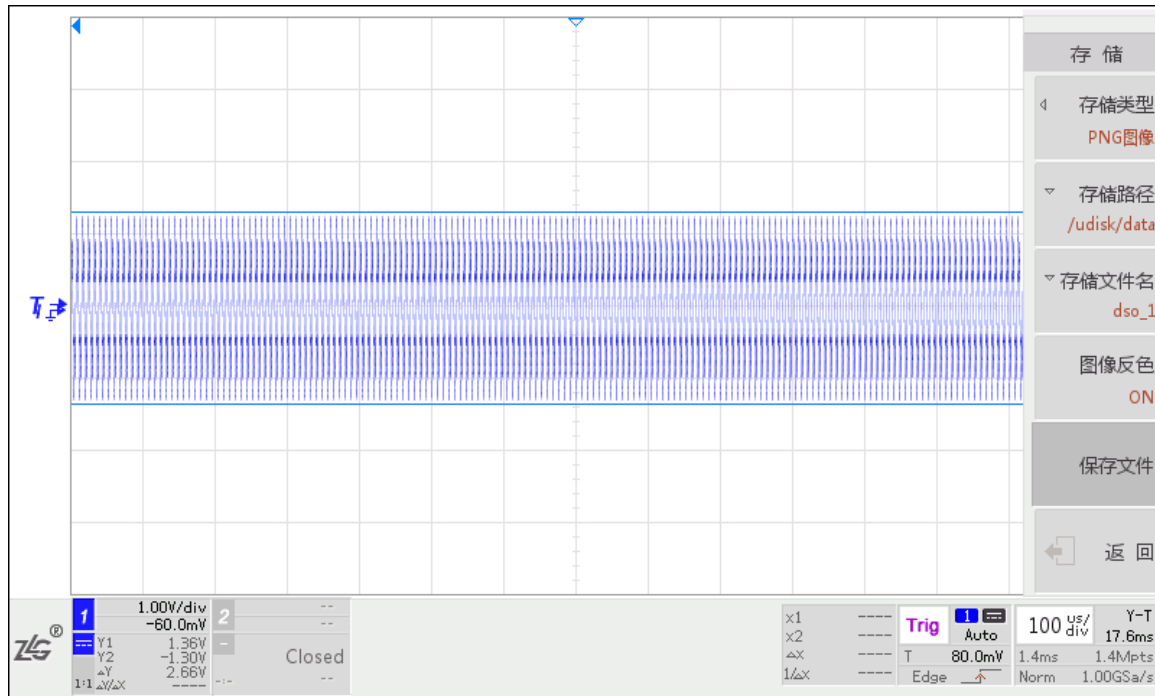
# What can we do with the HackCUBE?

1. Read the 125Khz ID tag
  2. Write to T5577 card with any card number from the stored card data or the inputted data
  3. Emulate as cards with any card number from the stored card data or the inputted data
- .....
- Any function you want to add which works at 125KHz



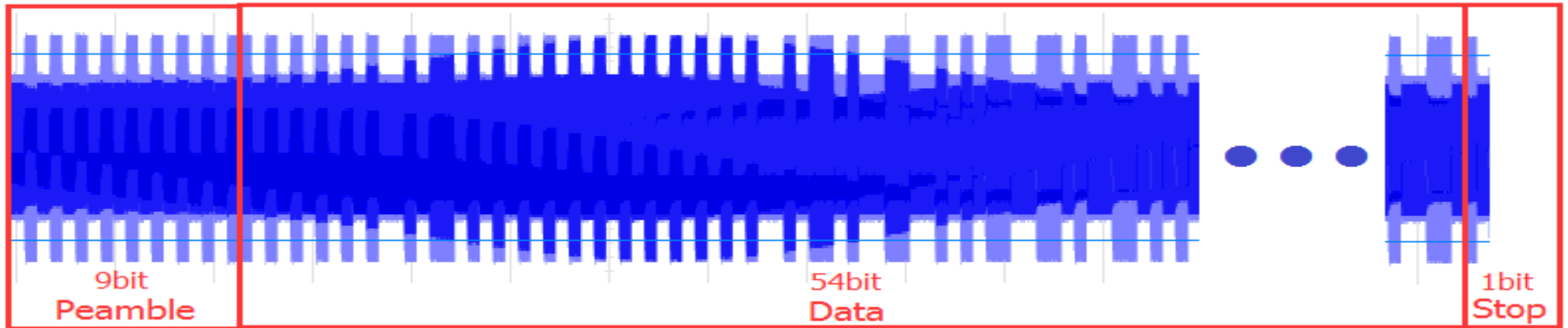
# Read ID(125KHz) tag

Signal of the reader without any tag    Put a tag close to the reader



# Analyzing the signal

The original signal



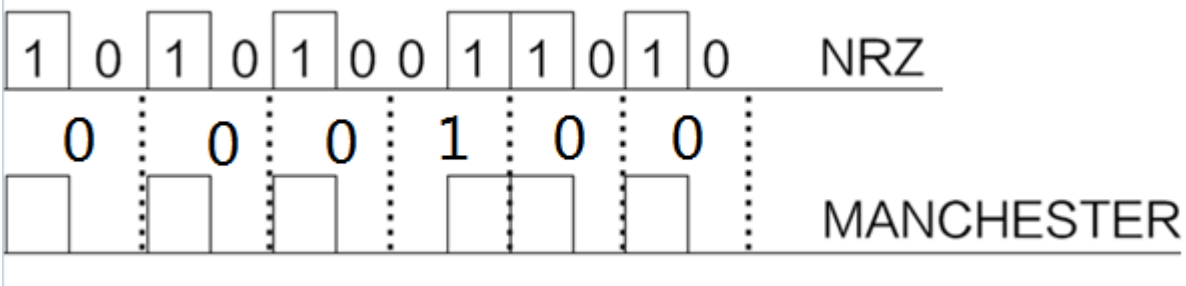
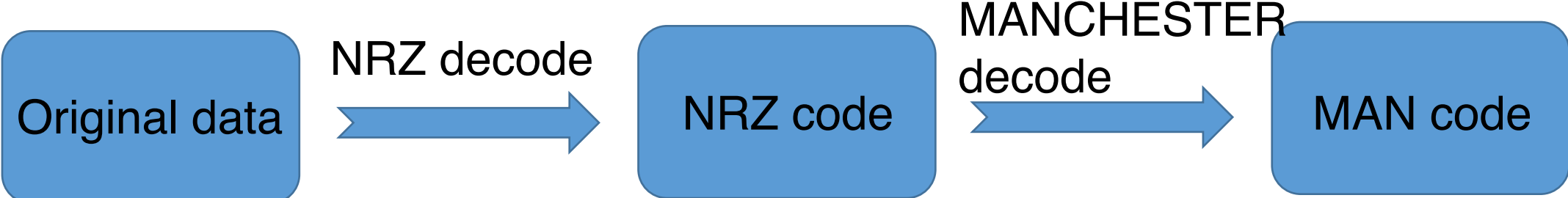
The signal can be divided into 3 parts

**Preamble:** Consist of 9 bit1

**Data:** data + parity check

**Stop:** always bit0

# Encoding format



```

NRZ decode 0101010101010101010110101010101001011010101010101010101010100
101
10100110010101100110100110101001011001010101100110100101011001
MANCHESTER 10
R decode 111111111000001100000000000001100101110100100011011110100111010
  
```



**Preamble** : sync (9 bit1)

**VD** : vendor identity(8 bit)

**ID** : identity(32bit)

**R** : row parity check  
(number of bit1 & 0x01)

**L** : column parity check  
(number of bit1 & 0x01)

**S** : stop(bit0)

1	1	1	1	1	1	1	1	1	9 bit preamble				
8 bit vendor identity				D00	D01	D02	D03	PR0	10 bit row parity check				
				D10	D11	D12	D13	PR1					
32 bit data				D20	D21	D22	D23	PR2					
				D30	D31	D32	D33	PR3					
				D40	D41	D42	D43	PR4					
				D50	D51	D52	D53	PR5					
				D60	D61	D62	D63	PR6					
				D70	D71	D72	D73	PR7					
				D80	D81	D82	D83	PR8					
				D90	D91	D92	D93	PR9					
				4 bit column parity check				PC0	PC1	PC2	PC3	S0	stop bit



# Get the tag number

1	1	1	1	1	1	1	1	1	9 bit preamble
8 bit vendor identity				0	0	0	0	0	10 bit row parity check
				0	1	1	0	0	
32 bit data				0	0	0	0	0	
				0	0	0	0	0	
				0	1	1	0	0	
				1	0	1	1	1	
				0	1	0	0	1	
				0	0	0	1	1	
				0	1	1	1	1	
				0	1	0	0	1	
4 bit column parity check				1	1	0	1	0	stop bit

Extract the VD & ID:

00000110

00000000

01101011

01000001

01110100

Tag : 06 00 6B 41 74

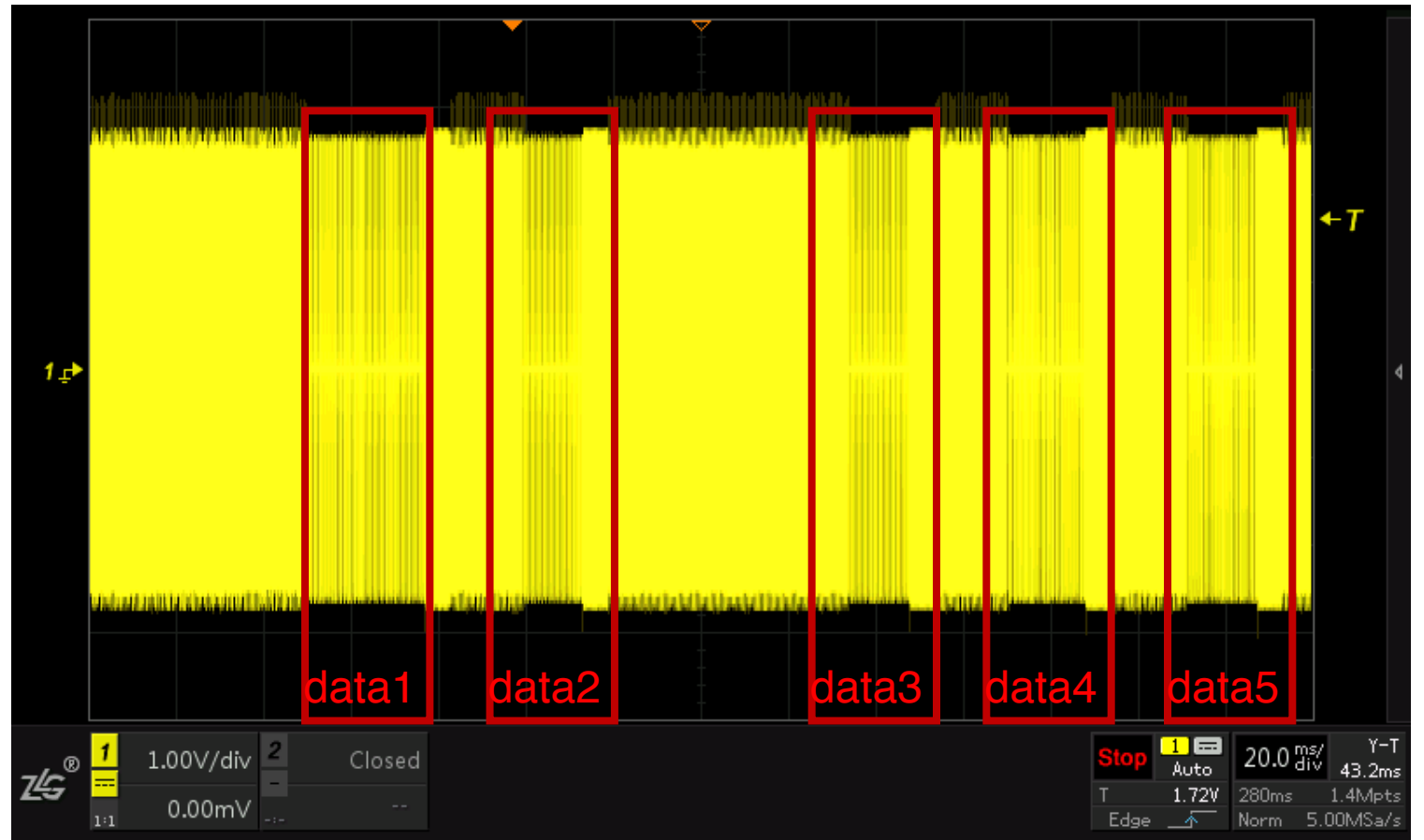
# Emulate as ID(125KHz) tag

The protocol is similar to reading card  
Just control the EM4095 chip as this protocol

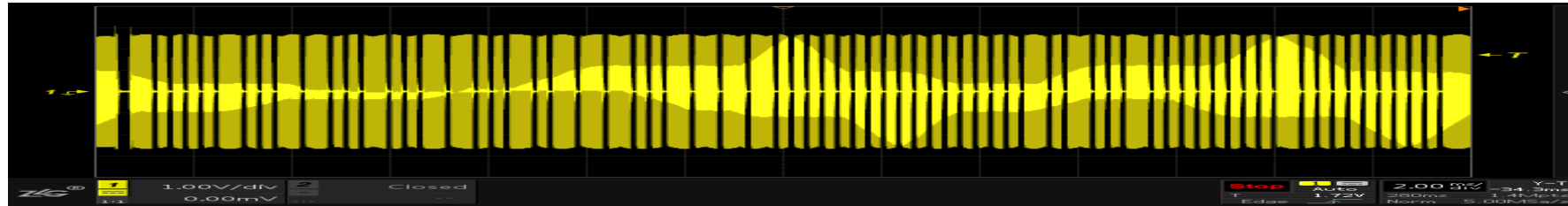
Warning : this emulation is a active emission, not similar to the real tag. It probably can't be recognized by the reader.

# Write ID(125KHz) tag

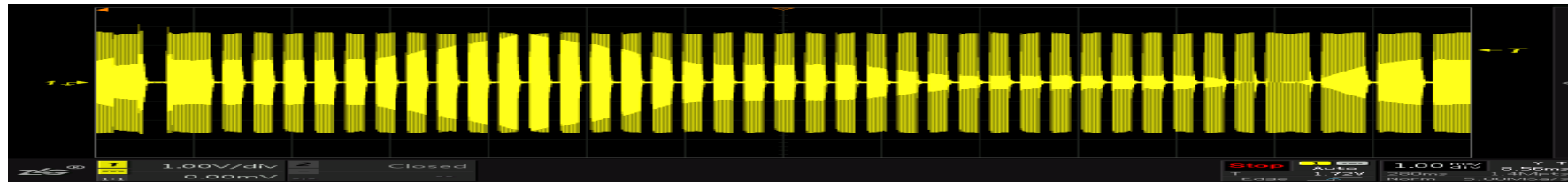
When we write to a writable tag, the captured signal is as the right picture



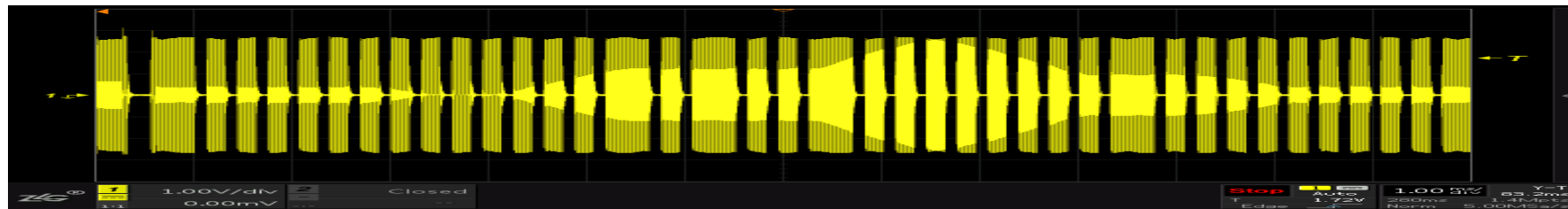
# Details of the signal



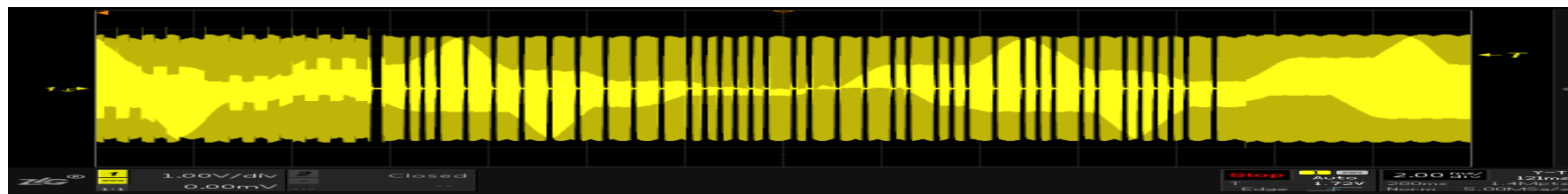
Data 1



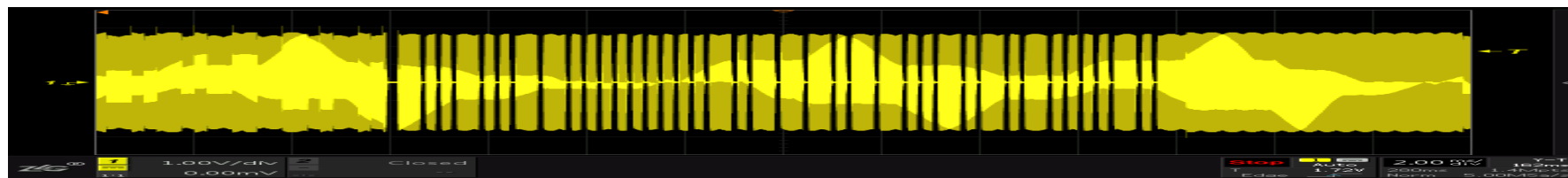
Data 2



Data 3



Data 4



Data 5



# Decoding the signal

The wide signal is bit1, and the narrow signal is bit0. The 5 data can be decoded as follow:

Data 1 : 100 00100110 01001100 10111001 11100000 00000000 00010100 10000000 01000000 000  
 Data 2 : 100 00000000 00000000 00000000 00000000 111  
 Data 3 : 100 00000000 00010100 10000000 01000000 000  
 Data 4 : 100 11111111 10000010 11101001 10010100 001  
 Data 5 : 100 10011000 00000010 11101100 10101000 010

signal	Preamble	data	address
Data1	100	00100110 01001100 10111001 11100000 00000000 00010100 10000000 01000000	000
Data2	100	00000000 00000000 00000000 00000000	111
Data3	100	00000000 00010100 10000000 01000000	000
Data4	100	11111111 10000010 11101001 10010100	001
Data5	100	10011000 00000010 11101100 10101000	010

The first 3 bits is the preamble

The last 3 bits is the writing address

# The function of each signal

1. The first data is written to block 0, 0x264CB9E0 0x00148040, this is the configuration
2. The second data(0x00000000) is the password of the tag, which is written to block 7
3. The third data(0x00148040) is also the configuration of the tag
4. The fourth and fifth data is the data, which we want to write to the tag. The written address of the tag is block 1 and block 2

# The data written to block 1 and block 2

Data 4 : 100 11111111 10000010 11101001 10010100 001

Data 5 : 100 10011000 00000010 11101100 101

1	1	1	1	1	1	1	1	1	9 bit preamble			
8 bit vendor identity				0	0	0	0	0				
				1	0	1	1	1				
32 bit data				0	1	0	0	1				
				1	0	0	1	0				
				1	0	0	1	0				
				0	1	1	0	0				
				0	0	0	0	0				
				0	0	1	0	1				
				1	1	0	1	1				
				0	0	1	0	1				
				4 bit column parity check				0	1	0	0	0
									stop bit			

After written the tag is  
0x 0b 49 96 02 d2

# Get your hands dirty;)

Following the steps

Step1: Plug the MicroUSB to power the HackCUBE-Special

Step2: Connect to the AP of the HackCUBE-Special

SSID: HackCUBE\_XX:XX:XX (MAC address)

key: hackcube

Step3: open the browser, enter 192.168.5.1

Step4: select the NFC tab

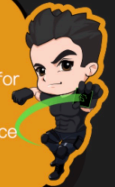
Step5: put the tag close to the antenna of NFC

(opposite of the Logo)

Then you can find the reading tag on the web.

### Warning

This equipment shall be only used for penetration testing of non-real systems. Please use it in accordance with local laws and regulations.



# HackCUBE-Special LF

RFID Low Frequency System(125Khz,EM41XX&T5577)  
Risk Evaluation

## Read ID



VID

ID

Simulate

## Emulate ID(Test)



VID

ID

## Write ID

Write

VID

ID

## Brute Attack(Test)



VD

Start ID

End ID

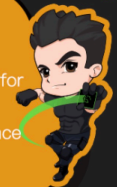
VD

Start ID

Stop ID

### Warning

This equipment shall be only used for penetration testing of non-real systems. Please use it in accordance with local laws and regulations.



# HackCUBE-Special HID

Human Interface Device Risk Evaluation

## Load Script

Key

Submit

## List Script

info	name	run
29d172a6	lock	execute
d2f392f1	cmatrix	execute
9209993f	Shellcode	execute



HAC is NOWN  
HACK. LEARN. FOR THE FUTURE.



LF

RF

HID

Setup

**Warning**

This equipment shall be only used for penetration testing of non-real systems. Please use it in accordance with local laws and regulations.



# HackCUBE-Special RF

Wireless System  
(2.4Ghz,315Mhz&433Mhz&868/915Mhz) Risk Evaluation

## Sniffer Data

Freq	Pac	Modu	Func	Data	Play
------	-----	------	------	------	------

## Transmit Data

Start

Freq	Protocol
315Mhz	PT226X
Data	Func
Data	Func

## Transmit Brute



Freq	Protocol	
315Mhz	PT226X	
Start	End	Func
Start address	Stop address	Func

LF

RF

HID

Setup

**Warning**

This equipment shall be only used for penetration testing of non-real systems. Please use it in accordance with local laws and regulations.



NFC Power  RF Power

**Frequency Setting**

315000000

**Protocol**

固定码

Settings Update

**Lighting Effects**

...

**Lighting Colours**

#ffffff

**Lighting Brightness**

128



效果更新



# 低频安全

## 门禁卡数据读取

Any questions?

[zhujiu1234@gmail.com](mailto:zhujiu1234@gmail.com)

Thank you~