



# 看雪 2017 安全开发者峰会

Kanxue 2017 Security Developer Summit

2000-2017



看雪峰会  
2017

# 如何黑掉无人机

谢君@阿里安全

# 关于我

- ▶ ID: vessial
- ▶ 阿里安全IoT安全研究负责人
- ▶ 安全领域从业12年
- ▶ 2016发现整个小米智能硬件远程利用漏洞
- ▶ 2015发现整个Broadlink智能硬件远程利用漏洞

无人机是个复杂的系统工程

# DJI无人机

飞控系统

避障测距定位系统

图像采集

无线通信

电源  
管理

电调  
控制

陀螺  
仪

加速  
度计

气压  
计

GPS

马达

视觉

超声  
波

红外

指南  
针

航拍  
摄像  
头

云台

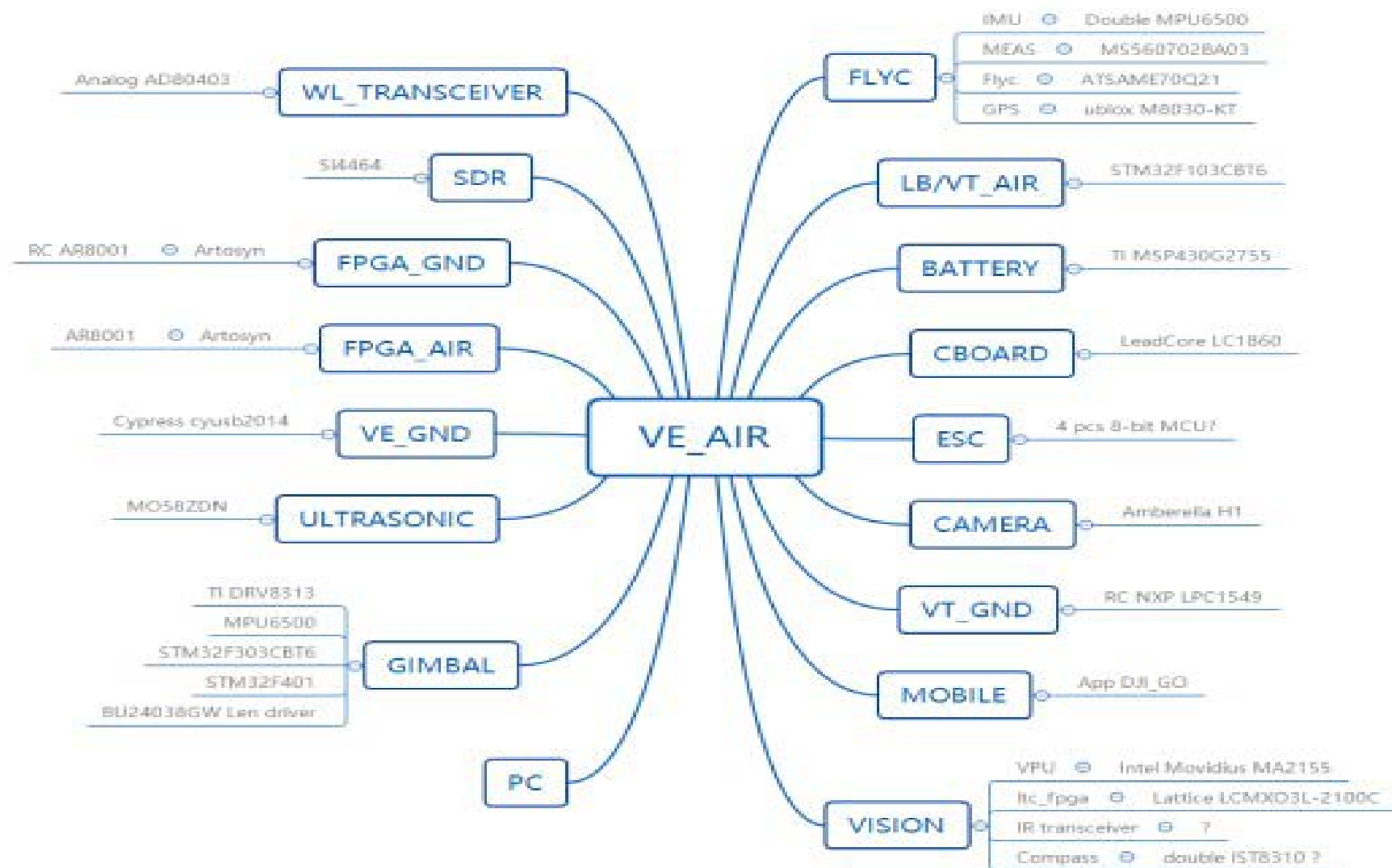
图像  
压缩

图像  
传输

控制  
传输

智能  
电源  
管理

# DJI P4P架构





# DJI构架-模块间通信

- ▶ 两个字节表示通信源地址或者目的地址

子系统编号	子功能编号
-------	-------

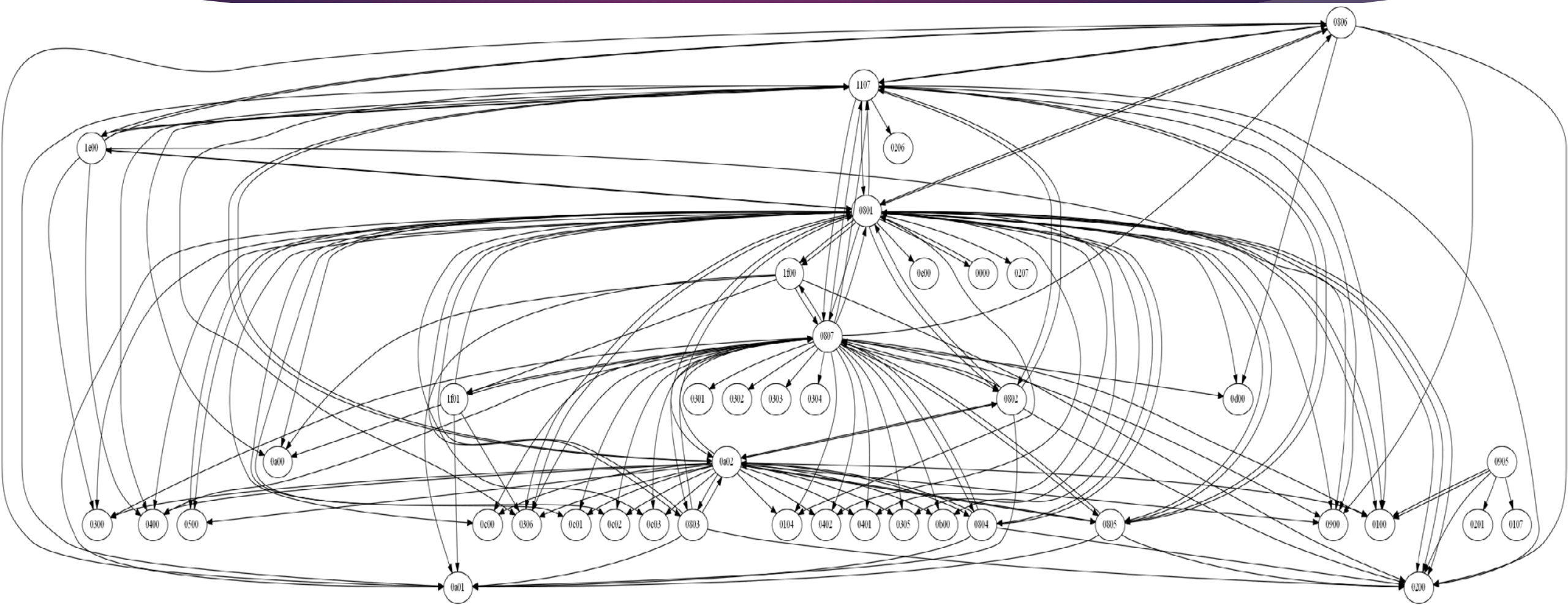
例如03表示飞控，06表示6号子功能

- ▶ 总共不超过32个子系统
- ▶ 通道方式
  - local,IP,WL,UART,I2C,SPI,CAN,ADB
  - HPI,USB,IAP2
- ▶ 通信协议
  - Logic
  - V0
  - V1
  - NAL
  - MAVLink

```
; "whoami" 00
; "camera" 01
; "mobile" 02
; "flight" 03
; "gimbal" 04
; "cboard" 05
; "rc" 06
; "network" 07
; "ve_air" 08
; "vt_air" 09
; "pc" 0a
; "battery" 0b
; "esc" 0c
; "ve_gnd" 0d
; "vt_gnd" 0e
; "s_to_p_air" 0f
; "s_to_p_gnd" 10
; "mvision" 0x11
; "bvision" 0x12
; "fpga_air" 0x13
; "fpga_gnd" 0x14
; "simulator" 0x15
; "null"
; "null"
; "null"
; "null"
; "mavlink" 0x1a
; "null"
; "glass" 0x1c
; "blackbox" 0x1d
; "test" 0x1e
; "all" 0x1f
```

```
0100 camera
0200 mobile
0400 gimbal
0500 cboard
0801 dji_sys
0802 ma2l55
0803 ltc_fpga
0804 ultrasonic
0805 dji_vision
0806 dji_decoding
0807 test_diag
0a00 pc con
0a02 firmware update from pc
1107 ma2l55 vision
1105 dji_flight for NFZ
0300 fly control
0301
0302
0303
0304
0305
0306
0900 LightBridge
0905 download service
0b00 battery
0c00 ESC 1
0c01 ESC 2
0c02 ESC 3
0c03 ESC 4
0d00 RC cyusb
0e00 RC LPC1549
1f00 all
```

# DJI构架-模块间通信



# DJI构架-模块间通信

## ► 通信协议Logic&V1

message v1 format

Magic Header 1byte → 0x55

Packet Len 10bits → less than 0x400

version 1byte&4 → 4 in P4 &P4P

CRC8 cksum 1 byte → crc8(header[0:3])

Seq 2bytes → random

src ID 1byte → 0A → 0A00

dst ID 1byte → 0E → 0E00

Attr ID 1byte →

cmd ID 2bytes → 00 01 version request

sub\_cmd ID 1byte optional

body len 2bytes optional

body few bytes optional

crc16 cksum 2bytes → crc16(pkt[:-2])

```
55 0D 04 33 0A 0E 13 27 40 00 01 16 72 .....U..3...'@...r
55 2C 04 36 0E 0A 13 27 80 00 01 00 10 50 34 5F ...U,.6...'€....P4_
50 56 32 00 00 00 00 00 00 00 00 00 06 03 01 ...PV2.....
03 02 03 02 05 43 02 00 00 00 8F 8B .....C.....e
```



# DJI构架-模块间通信

## ► 通信协议Logic&VI

message Logic Format

Magic header 4bytes -> 21 12 ad de

src ID 2bytes ->

dst ID 2bytes ->

Seq 2bytes ->

Attr ID 1byte ->

CMD ID 2bytes ->

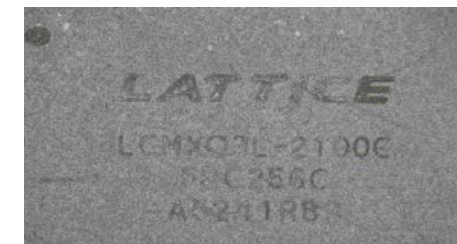
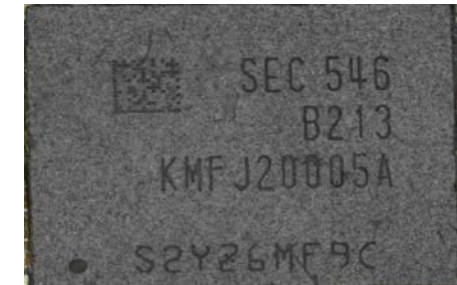
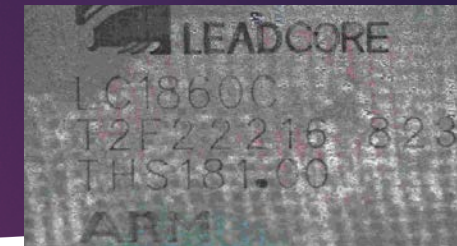
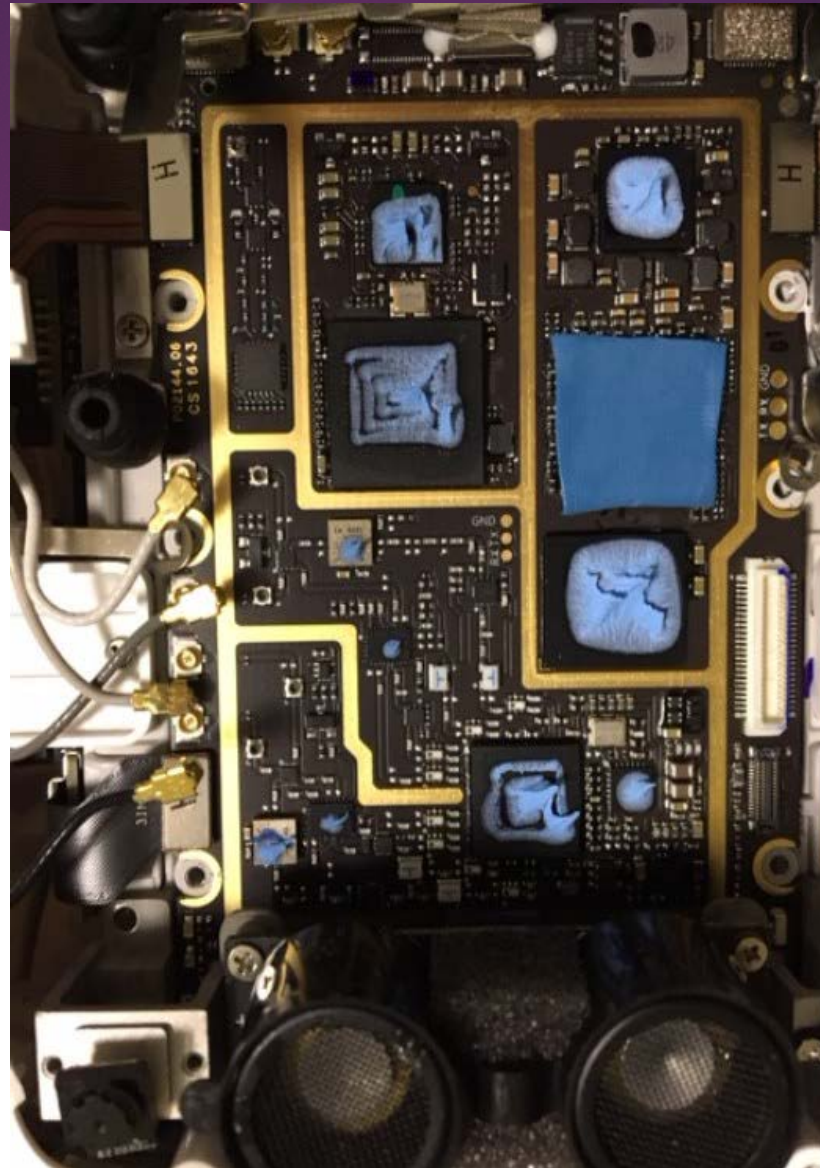
body few bytes ->

pkt len 2bytes -> packet len-6

```
def gen_host(d):  
    c=(d>>5)|((d&0x1f)<<8)  
    return c  
  
gen_host(0xf1)==0x1107  
gen_host(0x09)==0x0900
```

# CenterBoard

- ▶ CenterBoard MCU LC1860
- ▶ Vision MA2155+Lattice LCMXO3L2100C
- ▶ Memory&Storeage Samsung eMCP(eMMC+DRAM) KMFJ20005A 4GB
- ▶ SiliCon LABS Si4464



# Root无人机

为什么要root无人机？

更方便研究无人机，方便研究其它模块

目标root CenterBoard MCU LC1860系统

嵌入式linux，配备adb，默认不开启

关键脚本Start\_dji\_system.sh，adb\_en.sh

怎么root？

1.软件漏洞执行adb\_en.sh脚本

2.如果能够修改start\_dji\_system.sh脚本

“Debug=true” 也可以达到root目的

```
debug=false
grep production /proc/cmdline >> /dev/null
if [ $? != 0 ]; then
    debug=true # engineering version, enable adb by default
else
    cmdline='cat /proc/cmdline'
    temp=${cmdline##*board_sn=}
    board=${temp%% *}
    in_whitelist.sh $board
    if [ $? == 0 ]; then
        debug=true
    fi
fi

if $debug; then
    /system/bin/adb_en.sh
else
    setprop sys.usb.config rndis,mass_storage,bulk,acm
fi
```

```
#!/system/bin/sh

level=$1
: ${level:=NonSecurePrivilege}

# mark debug enable
mkdir -p /tmp/dji
echo $level > /tmp/dji/secure_debug

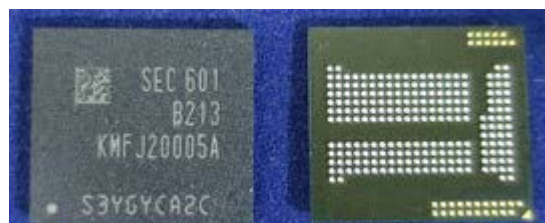
# init adb device serial
if [ -f /data/dji/cfg/adb_serial ]; then
    serial=`cat /data/dji/cfg/adb_serial`
    busybox printf "$serial" > /sys/class/android_usb/android0/iSerial
fi

setprop service.adb.root 1
setprop service.adb.tcp.port -1
setprop sys.usb.config rndis,mass_storage,bulk,acm,adb
busybox devmem 0xe10093d0 8 0x40 #enable uart
sleep 1
busybox udhcpd
```



# Root无人机

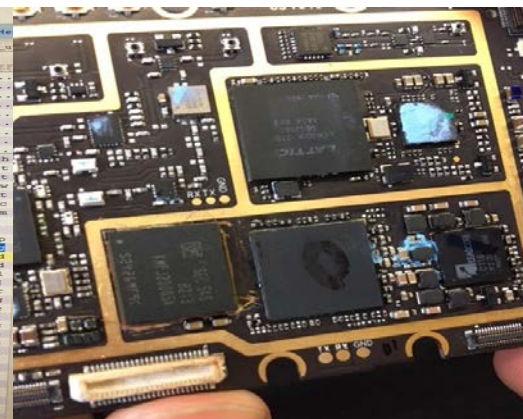
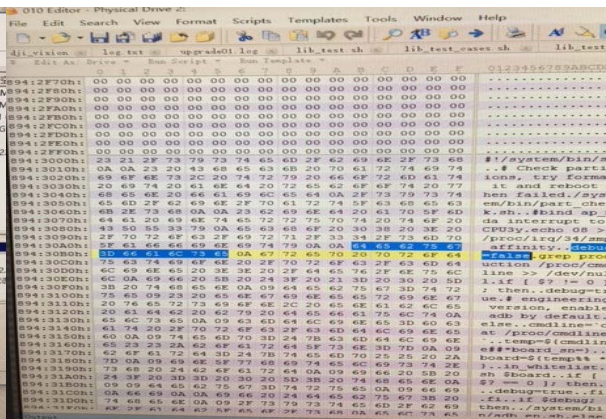
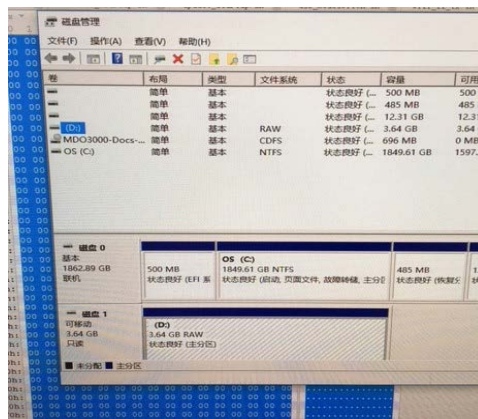
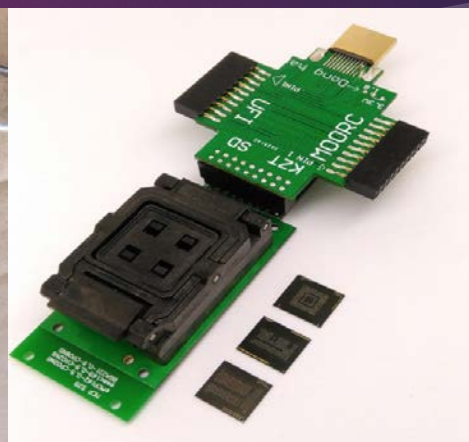
- ▶ 通过修改start\_dji\_system.sh脚本达到root效果
- ▶ 比较通用的root方法，需要软硬结合
- ▶ 目标就是修改存储器里面的内容





# Root无人机

- ▶ 风枪吹下eMCP
- ▶ eMMC读卡器读取Raw data
- ▶ 找到文件存放的合适分区
- ▶ Ext4挂载system分区
- ▶ 修改start\_dji\_system.sh文件
- ▶ System img写回Raw Data文件
- ▶ eMMC读卡器写回eMCP
- ▶ eMCP焊接回芯片座





# Root无人机

▶ finally

```
x: /etc/passwd
CPU architecture: 7
CPU variant : 0x0
CPU part : 0xc07
CPU revision : 5

processor : 3
model name : ARMv7 Processor rev 5 (v7l)
Processor : ARMv7 Processor rev 5 (v7l)
BogoMIPS : 26.00
Features : swp half thumb fastmult vfp edsp neon vfpv3 t
CPU implementer : 0x41
CPU architecture: 7
CPU variant : 0x0
CPU part : 0xc07
CPU revision : 5

processor : 4
model name : ARMv7 Processor rev 5 (v7l)
Processor : ARMv7 Processor rev 5 (v7l)
BogoMIPS : 26.00
Features : swp half thumb fastmult vfp edsp neon vfpv3 t
CPU implementer : 0x41
CPU architecture: 7
CPU variant : 0x0
CPU part : 0xc07
CPU revision : 5

Hardware : Leadcore Innopower
Revision : 0000
Serial : 0000000000000000
root@wm330_dz_vp0001_v5:/ # df
Filesystem      Size      Used      Free    Blksize
/dev            8.0M      128.0K      7.9M      4096
/tmp           32.0M      12.0K      32.0M      4096
/var            2.0M      12.0K      2.0M      4096
/ftp           1024.0K      0.0K     1024.0K      4096
/ant           11.7M      68.0K     11.7M      4096
/vendor         59.0M      6.8M      52.2M      4096
/system        122.0M      96.9M      25.1M      4096
/data          1.1G      97.3M     1018.6M      4096
/blackbox       1.9G      993.6M      990.2M      4096
/cache         248.0M      55.0M      193.0M      4096
/ftp/upgrade   1.1G      97.3M     1018.6M      4096
/ftp/blackbox  1.1G      993.6M      990.2M      4096
/tmp/cam_storage 14.9G      10.9G      4.0G      32768
root@wm330_dz_vp0001_v5:/ #

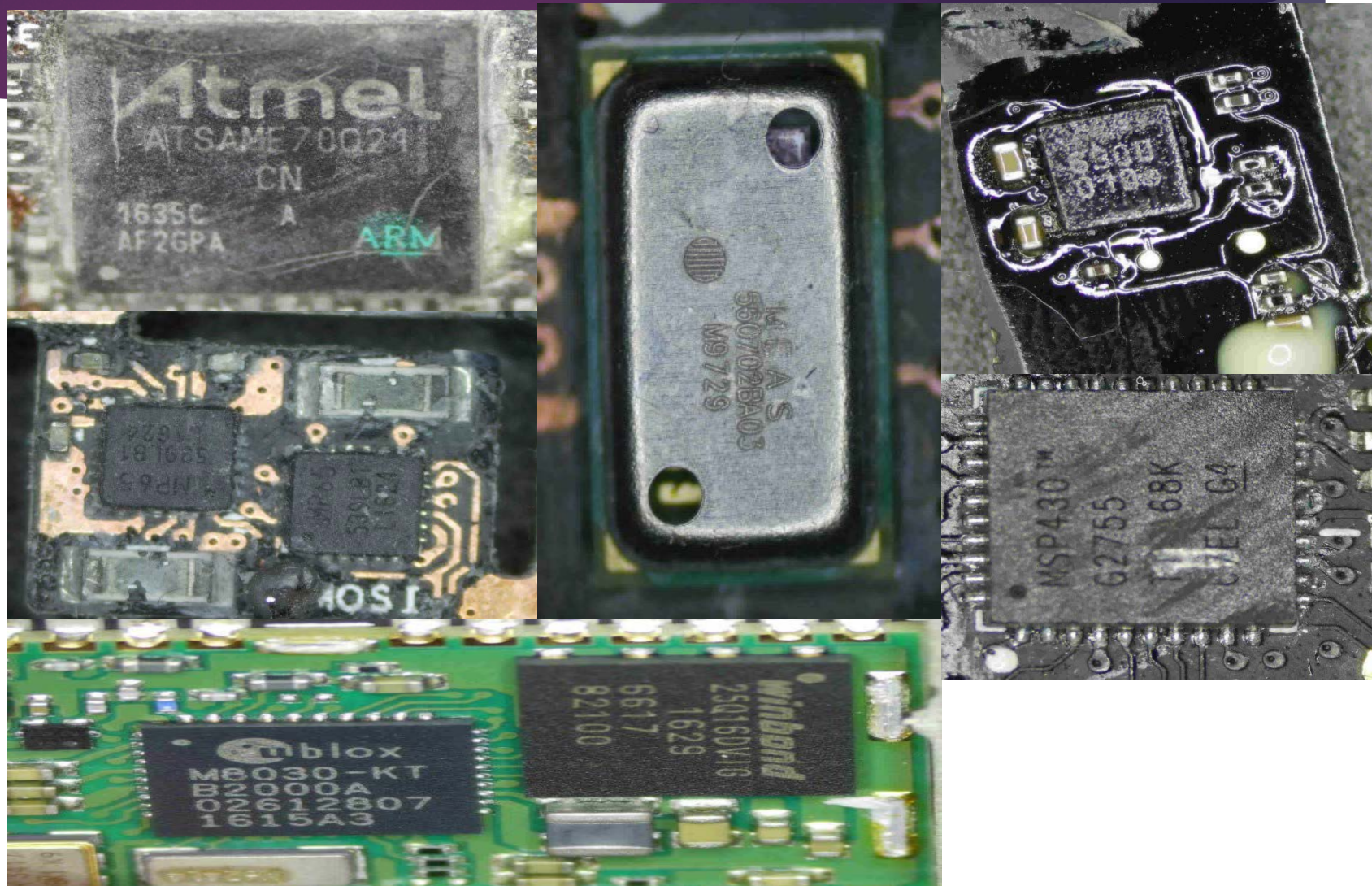
root@wm330_dz_vp0001_v5:/tmp # df
Filesystem      Size      Used      Free    Blksize
/dev            8.0M      128.0K      7.9M      4096
/tmp           32.0M      12.0K      32.0M      4096
/var            2.0M      12.0K      2.0M      4096
/ftp           1024.0K      0.0K     1024.0K      4096
/ant           11.7M      68.0K     11.7M      4096
/vendor         59.0M      6.8M      52.2M      4096
/system        122.0M      96.9M      25.1M      4096
/data          1.1G      97.3M     1018.6M      4096
/blackbox       1.9G      993.6M      990.2M      4096
/cache         248.0M      55.0M      193.0M      4096
/ftp/upgrade   1.1G      97.3M     1018.6M      4096
/ftp/blackbox  1.1G      993.6M      990.2M      4096
/tmp/cam_storage 14.9G      10.9G      4.0G      32768
root@wm330_dz_vp0001_v5:/tmp #

root@wm330_dz_vp0001_v5:/tmp # netstat -tlnp
Active UNIX domain sockets (servers and established)
Proto RefCnt Flags      Type       State
tcp      0      0 0.0.0.0:8905      0.0.0.0:*   LISTEN
tcp      0      0 0.0.0.0:8906      0.0.0.0:*   LISTEN
tcp      0      0 0.0.0.0:8907      0.0.0.0:*   LISTEN
tcp      0      0 0.0.0.0:8908      0.0.0.0:*   LISTEN
tcp      0      0 0.127.0.0:5037    0.0.0.0:*   LISTEN
tcp      0      0 0.192.168.42:21  0.0.0.0:*   LISTEN
udp      0      0 0.0.0.0:67        0.0.0.0:*
udp      0      0 0.0.0.0:67        0.0.0.0:*
netstat: /proc/net/tcp6: No such file or directory
netstat: /proc/net/udp6: No such file or directory
Active UNIX domain sockets (servers and established)
Proto RefCnt Flags      Type       State
unix 2      [ ]      DGRAM      1-Node PID/Program name Path
unix 2      [ ]      DGRAM      2170 206/dji_sys @/duss/mb/0x1f00
unix 2      [ ]      DGRAM      2188 1/init /dev/socket/property_ser
unix 2      [ ]      DGRAM      2168 206/dji_sys @/duss/mb/0x804
unix 2      [ ]      DGRAM      2167 206/dji_sys @/duss/mb/0x803
unix 2      [ ]      DGRAM      2166 206/dji_sys @/duss/mb/0x802
unix 2      [ ]      DGRAM      2165 206/dji_sys @/duss/mb/0x801
unix 2      [ ]      DGRAM      2162 208/dji_encoding @/duss/mb/0x800
unix 2      [ ]      DGRAM      3183 210/dji_vision @/duss/mb/0x1107
unix 2      [ ]      DGRAM      3124 204/dji_monitor @/duss/mb/0x002
unix 2      [ ]      DGRAM      3115 211/debugger @/duss/monitor
unix 2      [ ]      DGRAM      3156 307/adbd /android:debuggerd
unix 3      [ ]      DGRAM      2169 206/dji_sys @/duss/mb/0x0
unix 3      [ ]      DGRAM      2113 1/init
unix 3      [ ]      DGRAM      60394 307/adbd
unix 3      [ ]      DGRAM      3154 307/adbd
unix 3      [ ]      DGRAM      2114 1/init
unix 2      [ ]      DGRAM      60353 307/adbd
unix 2      [ ]      DGRAM      4215 206/dji_sys
unix 2      [ ]      DGRAM      3120 208/dji_encoding
unix 2      [ ]      DGRAM      3157 307/adbd
unix 3      [ ]      DGRAM      3124 210/dji_vision
unix 3      [ ]      DGRAM      3123 204/dji_monitor
unix 3      [ ]      DGRAM      3158 307/adbd
root@wm330_dz_vp0001_v5:/tmp #
```



# 飞控系统

- ▶ 飞控MCU Atmel ATSAME70Q21
- ▶ IMU MP6500(陀螺仪+加速度计)
- ▶ 气压计 MEAS MS560702BA03
- ▶ Ublox GPS模块
- ▶ Double IST8310指南针模块
- ▶ 智能电源
- ▶ 电调ESC





# 视觉系统

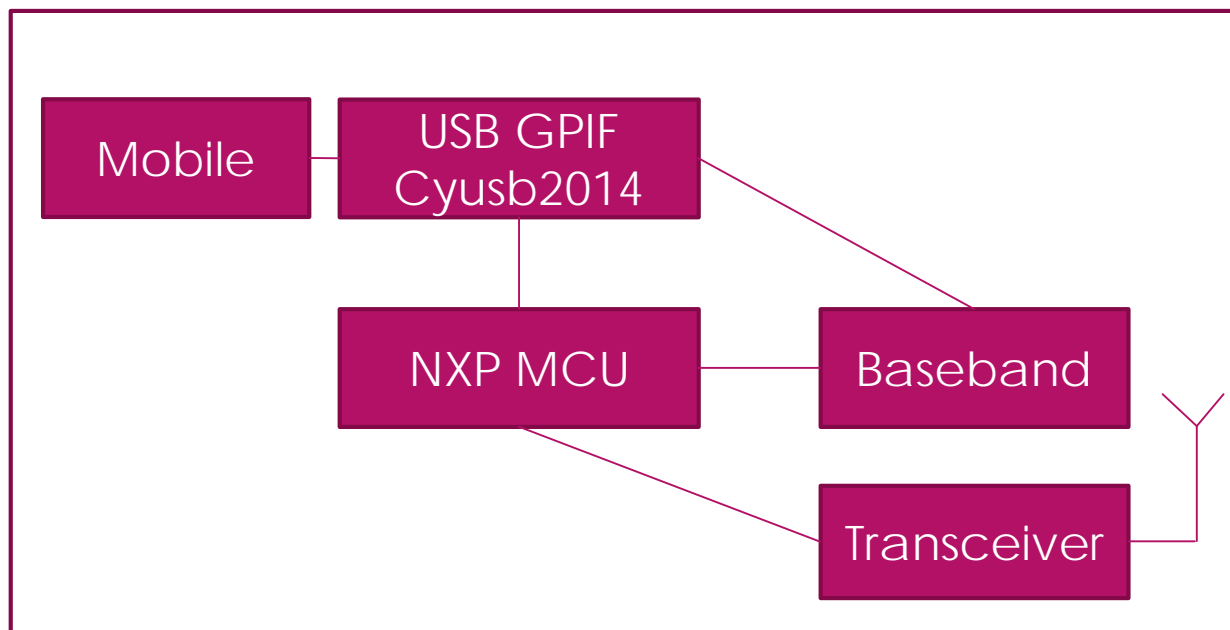
- ▶ MA2155深度视觉处理
  - ▶ 前后下障碍物识别与测距
  - ▶ 人物,姿态,水,ROI识别
- ▶ 红外双目避障
  - ▶ 左右红外识别
- ▶ 超声波
  - ▶ 检测下方障碍物



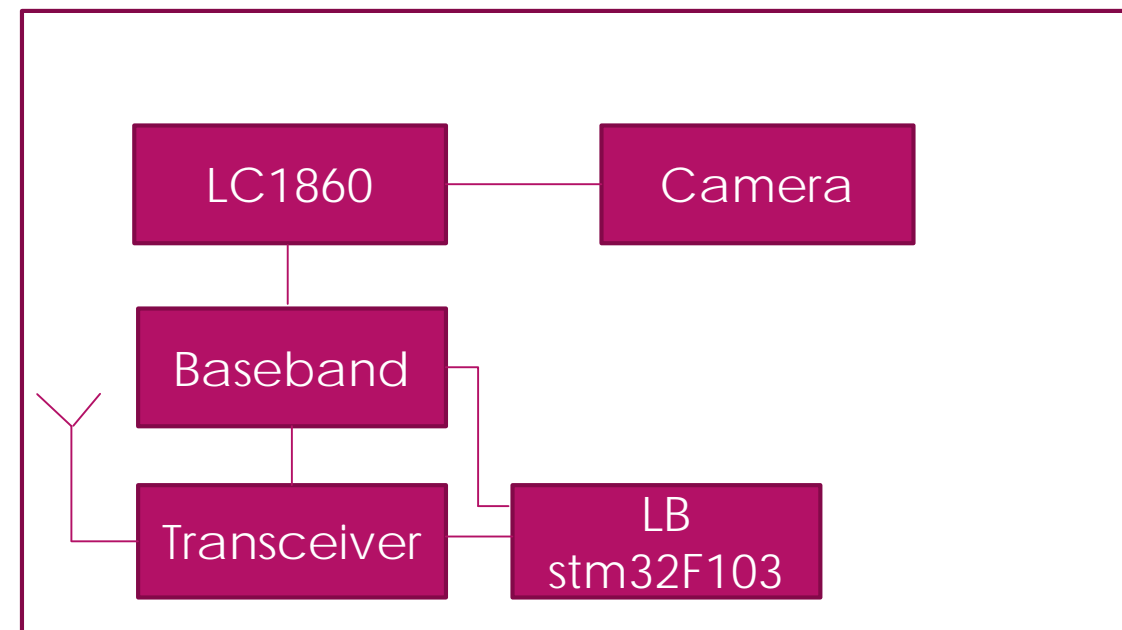


# 无线通信系统架构

RC



Aircraft



# 无线通信系统架构

- ▶ 图传通道&数控通道
- ▶ OFDM宽带传输
- ▶ 1Mhz&10Mhz带宽
- ▶ RC 1Tx&2Rx signal path channel
- ▶ Air 1Tx&2Rx signal path channel
- ▶ Transceiver AD80403 SPI config same as Ad9361
- ▶ 数控通道2.4G 24个跳频序列, 5.8G 12个跳频序列
- ▶ 数控通道2.4G 45个频点, 5.8G 42个频点
- ▶ 每个频点切换间隔14ms, 1s切换71个频点

Operating Frequencies		
Device operates within approved frequencies overlapping with the following cellular bands: LTE 255, Unlicensed NII-3 DOWN   LTE 46, TD Unlicensed DOWN		
Frequency Range	Power Output	Rule Parts
2.404-2.4788 GHz	566 mW	15C
5.727-5.8213 GHz	175 mW	15E



# 无线通信系统架构

- ▶ Pair&link
- ▶ 配对码由RC MCU ID生成
- ▶ 通过RC ID生成跳频序列

```
freq hop seed generation
4pro type==3

a=crc8(a1,4,0x33)
a1=[0xa7,0x9f,0x4e,0x09],rc id gened
```

```
table_58[1,5,0x0b,0x0d,0x11,0x13,0x17,0x19,0x1d,0x1f,0x25,0x29]
table_24[1,2,4,7,8,0x0b,0x0d,0x0e,0x10,0x11,0x13,0x16,0x17,0x1a,0x1c,0x1d,0x1f,0x20,0x22,0x25,0x26,0x29,0x2b,0x2c]
```

```
if band==2
    b=a%0x0c
    c=table_58[b]
if band==0
    b=a%0x18
    c=table_24[b]
init_seed=c
```

```
if band==2 and type==3
    hops=0x2a
if band==0 and type==3
    hops=0x2d
```

RC id generation algorithm

IAP\_chip\_ReadUID get 16 bytes

b1=crc8(uid,0x77)

b2=crc16(uid,0x3692)

b3=xor(uid,)

RC\_id=[b2[0:1],b1,b3]

all of 5.8ghz freqz tx hop list

```
1
['5732.0', '5734.0', '5736.0', '5738.0', '5741.0', '5743.0', '5745.0', '5748.0', '5750.0', '5752.0', '5755.0', '5757.0', '5759.0', '5761.0', '5764.0', '5766.0', '5768.0', '5771.0', '5773.0', '5775.0', '5778.0', '5780.0', '5782.0', '5784.0', '5787.0', '5789.0', '5791.0', '5794.0', '5796.0', '5798.0', '5801.0', '5803.0', '5805.0', '5807.0', '5810.0', '5812.0', '5814.0', '5817.0', '5819.0', '5821.0', '5727.0', '5729.0']
5
['5750.0', '5761.0', '5773.0', '5784.0', '5796.0', '5807.0', '5819.0', '5734.0', '5745.0', '5757.0', '5768.0', '5780.0', '5791.0', '5803.0', '5814.0', '5729.0', '5741.0', '5752.0', '5764.0', '5775.0', '5787.0', '5798.0', '5810.0', '5821.0', '5736.0', '5748.0', '5759.0', '5771.0', '5782.0', '5794.0', '5805.0', '5817.0', '5732.0', '5743.0', '5755.0', '5766.0', '5778.0', '5789.0', '5801.0', '5812.0', '5727.0', '5738.0']
11
['5778.0', '5803.0', '5732.0', '5757.0', '5782.0', '5807.0', '5736.0', '5761.0', '5787.0', '5812.0', '5741.0', '5766.0', '5791.0', '5817.0', '5745.0', '5771.0', '5796.0', '5821.0', '5750.0', '5775.0', '5801.0', '5729.0', '5755.0', '5780.0', '5805.0', '5734.0', '5759.0', '5784.0', '5810.0', '5738.0', '5764.0', '5789.0', '5814.0', '5743.0', '5768.0', '5794.0', '5819.0', '5748.0', '5773.0', '5798.0', '5727.0', '5752.0']
13
['5787.0', '5817.0', '5750.0', '5780.0', '5810.0', '5743.0', '5773.0', '5803.0', '5736.0', '5766.0', '5796.0', '5729.0', '5759.0', '5789.0', '5819.0', '5752.0', '5782.0', '5812.0', '5745.0', '5775.0', '5805.0', '5738.0', '5768.0', '5798.0', '5732.0', '5761.0', '5791.0', '5821.0', '5755.0', '5784.0', '5814.0', '5748.0', '5778.0', '5807.0', '5741.0', '5771.0', '5801.0', '5734.0', '5764.0', '5794.0', '5727.0', '5757.0']
17
['5805.0', '5748.0', '5787.0', '5729.0', '5768.0', '5807.0', '5750.0', '5789.0', '5732.0', '5771.0', '5810.0', '5752.0', '5791.0', '5734.0', '5773.0', '5812.0', '5755.0', '5794.0', '5736.0', '5775.0', '5814.0', '5757.0', '5796.0', '5738.0', '5778.0', '5817.0', '5759.0', '5798.0', '5741.0', '5780.0', '5819.0', '5761.0', '5801.0', '5743.0', '5782.0', '5821.0', '5764.0', '5803.0', '5745.0', '5784.0', '5727.0', '5766.0']
19
['5814.0', '5761.0', '5805.0', '5752.0', '5796.0', '5743.0', '5787.0', '5734.0', '5778.0', '5821.0', '5768.0', '5812.0', '5759.0', '5803.0', '5750.0', '5794.0', '5741.0', '5784.0', '5732.0', '5775.0', '5819.0', '5766.0', '5810.0', '5757.0', '5801.0', '5748.0', '5791.0', '5738.0', '5782.0', '5729.0', '5773.0', '5817.0', '5764.0', '5807.0', '5755.0', '5798.0', '5745.0', '5789.0', '5736.0', '5780.0', '5727.0', '5771.0']
23
['5736.0', '5789.0', '5745.0', '5798.0', '5755.0', '5807.0', '5764.0', '5817.0', '5773.0', '5729.0', '5782.0', '5738.0', '5791.0', '5748.0', '5801.0', '5757.0', '5810.0', '5766.0', '5819.0', '5775.0', '5732.0', '5784.0', '5741.0', '5794.0', '5750.0', '5803.0', '5759.0', '5812.0', '5768.0', '5821.0', '5778.0', '5734.0', '5787.0', '5743.0', '5796.0', '5752.0', '5805.0', '5761.0', '5814.0', '5771.0', '5727.0', '5780.0']
25
['5745.0', '5803.0', '5764.0', '5821.0', '5782.0', '5743.0', '5801.0', '5761.0', '5819.0', '5780.0', '5741.0', '5798.0', '5759.0', '5817.0', '5778.0', '5738.0', '5796.0', '5757.0', '5814.0', '5775.0', '5736.0', '5794.0', '5755.0', '5812.0', '5773.0', '5734.0', '5791.0', '5752.0', '5810.0', '5771.0', '5732.0', '5789.0', '5750.0', '5807.0', '5768.0', '5729.0', '5787.0', '5748.0', '5805.0', '5766.0', '5727.0', '5784.0']
```



# 无线通信系统架构

- ▶ RC初始化->NXP LPC1549
- ▶ RC初始化baseband的SPI寄存器
- ▶ RC初始化Transceiver的SPI寄存器
- ▶ RC固定跳频广播
- ▶ Air初始化->STM32F103
- ▶ Air初始化baseband的SPI寄存器
- ▶ Air初始化Transceiver的SPI寄存器
- ▶ Air挑选一个信噪比高的频点监听
- ▶ Air收到RC信号并同步跳频序列

- ▶ 写入5字节的配对码到baseband SPI地址3, 4, 5, 6, 7

- ▶ 写入5字节的配对码

```
00007100 04 29 05 47 FF FF CB D3 01 E1 07 04 0E 15 02 2C .).G.....,
00007110 47 03 11 07 30 43 4B 4A 32 30 31 44 47 55 FF FF G...0CKJ201DGU..
00007120 FF FF FF FF 8B 6D 05 95 36 F1 2E 62 E1 9B 7E A8 .....m..6.....~
00007130 1B 6A CA FC 81 61 18 75 48 B6 21 B9 98 0B 6C B4 ...j...a.uK.!...l.
00007140 C2 45 AE 19 7E F6 A8 1D 76 D3 CB 7D 66 16 43 D7 ...~.....}f.C.
00007150 62 80 DF 86 01 23 64 1B 8E 8F 01 EB EE 59 03 36 b..#d.....6
00007160 08 E0 0C 1E 00 3C 00 00 DA 02 F5 07 1C 0D 1E 00 .....<.....
00007170 3C 00 00 49 03 50 08 0E 0D 1E 00 3C 00 01 3C 03 <..I.P.....<..
00007180 3A 08 00 0D 1E 00 3C 00 01 6D 00 CB 07 55 0F C8 :.....<..m...U..
00007190 00 C8 00 00 02 FF FF FF FF 0A 49 8A 59 0C 0B 13 .....I.Y...
000071A0 07 00 0A 00 06 06 06 06 06 06 06 06 06 06 06 .....
000071B0 06 06 06 06 06 06 06 06 06 06 06 06 06 06 06 .....
000071C0 06 06 06 06 06 06 06 06 06 06 06 06 06 06 06 .....
000071D0 06 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
000071E0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
000071F0 00 00 00 00 00 00 00 00 A3 A3 A3 A2 A2 A2 A1 .....
00007200 A1 A1 A1 A0 A0 A0 A0 9F 9F 9F 9E 9E 9D 9D 9C .....
00007210 .....
00007220 .....
0001CCA0 .....
0001CCA0 spi02bb_paircode
0001CCA0 PUSH {R4,LR}
0001CCA2 MOV R4, R0
0001CCA4 MOVS R0, #1
0001CCA6 BL spi_read
0001CCAA LDRB R1, [R4]
0001CCAC MOVS R0, #3
0001CCAE BL spi_writelbyte
0001CCB2 LDRB R1, [R4,#1]
0001CCB4 MOVS R0, #4
0001CCB6 BL spi_writelbyte
0001CCBA LDRB R1, [R4,#2]
0001CCBC MOVS R0, #5
0001CCBE BL spi_writelbyte
0001CCC2 LDRB R1, [R4,#3]
0001CCC4 MOVS R0, #6
0001CCC6 BL spi_writelbyte
0001CCCA LDRB R1, [R4,#4]
0001CCCC MOVS R0, #7
0001CCCE BL spi_writelbyte
0001CCD2 POP.W {R4,LR}
0001CCD6 MOVS R0, #2
0001CCD8 B.W spi_read
0001CCD8 ; End of function spi02bb_paircode
```



# 无线通信系统架构

- ▶ RC与Air同步原理
- ▶ RC跳频循环设置

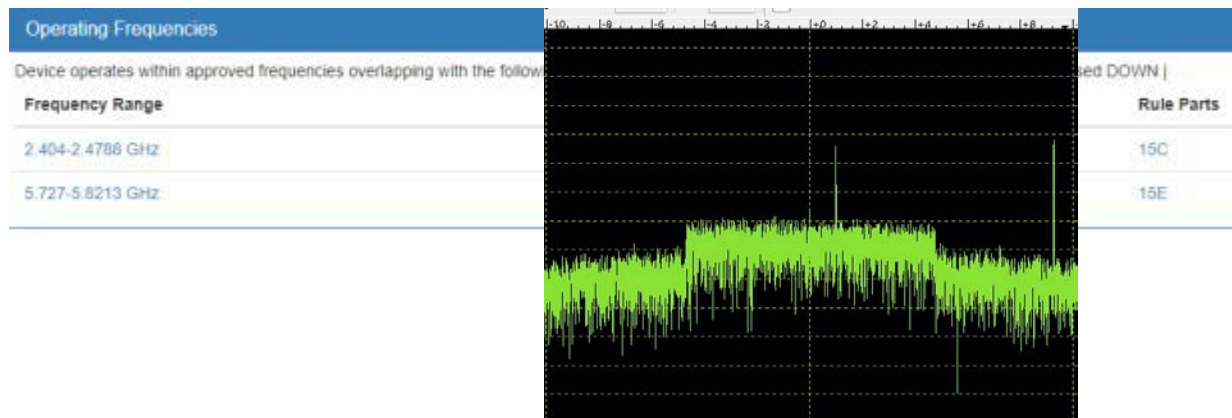
```
rx freq 5845
rx freq 5825
tx freq 5738
rx freq 5845
rx freq 5830
tx freq 5782
rx freq 5845
rx freq 5835
```

rx跳频设置----- 0.3ms---tx跳频设置----3.075ms-----rx图传设置----13.69ms---rx下一跳频设置

- ▶ Air跳频循环设置

rx图传设置----- 0.3ms---rx跳频设置----2.91ms ----tx图传数传设置 ----10.78ms---rx图传设置

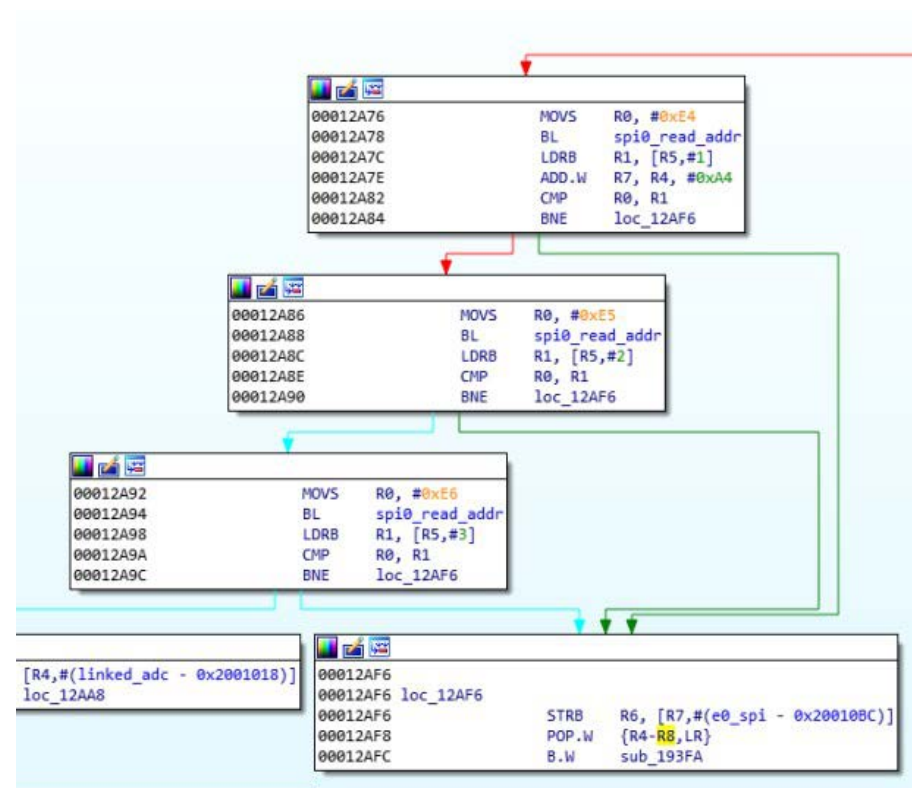
```
rx freq 5845
rx freq 5738
tx freq 5845
rx freq 5845
rx freq 5782
tx freq 5845
rx freq 5845
```



2.400 - 2.483 GHz和5.725 - 5.825 GHz  
2.400 - 2.483 GHz (无干扰、无遮挡)  
FCC : 7000 m  
CE : 3500 m  
SRRC : 4000 m  
5.725 - 5.825 GHz (无干扰、无遮挡)  
FCC : 7000 m  
CE : 2000 m  
SRRC : 5000 m

# 无线通信系统架构

- ▶ RC验证Air配对
- ▶ RC读取baseband SPI地址（0xe4,0xe5,0xe6）值进行比较



# 如何劫持一架无人机

- ▶ 通过破解配对码，码字空间0x10000000
- ▶ 改造RC固件，固定频点每隔10ms初始化baseband SPI配对码寄存器
- ▶ 每隔10ms通过SDR重放固定频点Air信号
- ▶ 读取baseband SPI寄存器（0xe4,0xe5,0xe6）值进行比较
- ▶ 结果写入到串口
- ▶ 破解需要离线操作，不能实时
- ▶ 理论上保守估计破解最长只需要46小时
- ▶ 破解成功可以远程修改飞机的配对码，并完全控制无人机

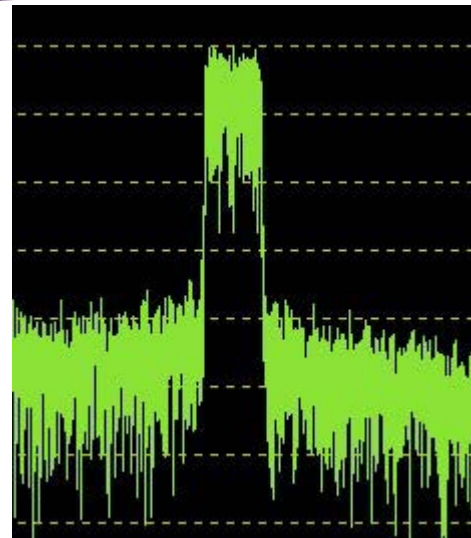
Todo list:

成功注入代码到RC固件，测试通过

编写破解算法代码

构造RC固件

测试验证



# 无人机反调对抗技术

- ▶ Stm32 IO remap defeat SWD debugging
- ▶ 硬件外设检测，ADC粒度检测
- ▶ Atmel芯片Security Bit
- ▶ 开关保护电路对抗SWD reset信号

```
SUB      SP, SP, #0x20
BL      sub_800D7B8
MOVS    R1, #1
MOV     R0, R1
BL      RCC_APB2PeriphClockCmd
MOVS    R1, #1
LDR     R0, =0x300400 ; Full SWJ Disabled (JTAG-DP + SW-DP)
BL      GPIO_PinRemapConfig
ADD     R0, SP, #0x20+var_14
```

```
00BE14 set_jreset_io_disable ; CODE XREF: sub_800D588+524p
00BE14
00BE14 var_8 = -8
00BE14 var_6 = -6
00BE14 var_5 = -5
```

```
PUSH    {R3,LR}
MOVS    R1, #1
MOVS    R0, #4
BL      RCC_APB2PeriphClockCmd
MOV.W   R0, #0x8000
STRH.W  R0, [SP,#8+var_8]
MOVS    R0, #3
STRB.W  R0, [SP,#8+var_6]
MOVS    R0, #4
STRB.W  R0, [SP,#8+var_5]
MOV     R1, SP
LDR     R0, =0x40010800
BL      GPIO_Init ; PA15 disable JTDI
POP     {R3,PC}
```

```
PUSH    {R3,LR}
MOVS    R1, #1
MOVS    R0, #8
BL      RCC_APB2PeriphClockCmd
MOVS    R0, #0x10
STRH.W  R0, [SP,#8+var_8]
MOVS    R1, #3
STRB.W  R1, [SP,#8+var_6]
STRB.W  R0, [SP,#8+var_5]
LDR     R0, =0x40010C0C
MOV     R1, SP
SUBS    R0, #0xC
BL      GPIO_Init ; PB4
```



100



# 后续研究

- ▶ OTA升级协议破解
- ▶ 构造某些硬件模块固件并升级（智能电池，LightBridge）
- ▶ Fuzzing飞控的协议接口
- ▶ SDR模拟无线通信

# Q&A



Thanks  
[vessialq@gmail.com](mailto:vessialq@gmail.com)