如何黑掉无人机

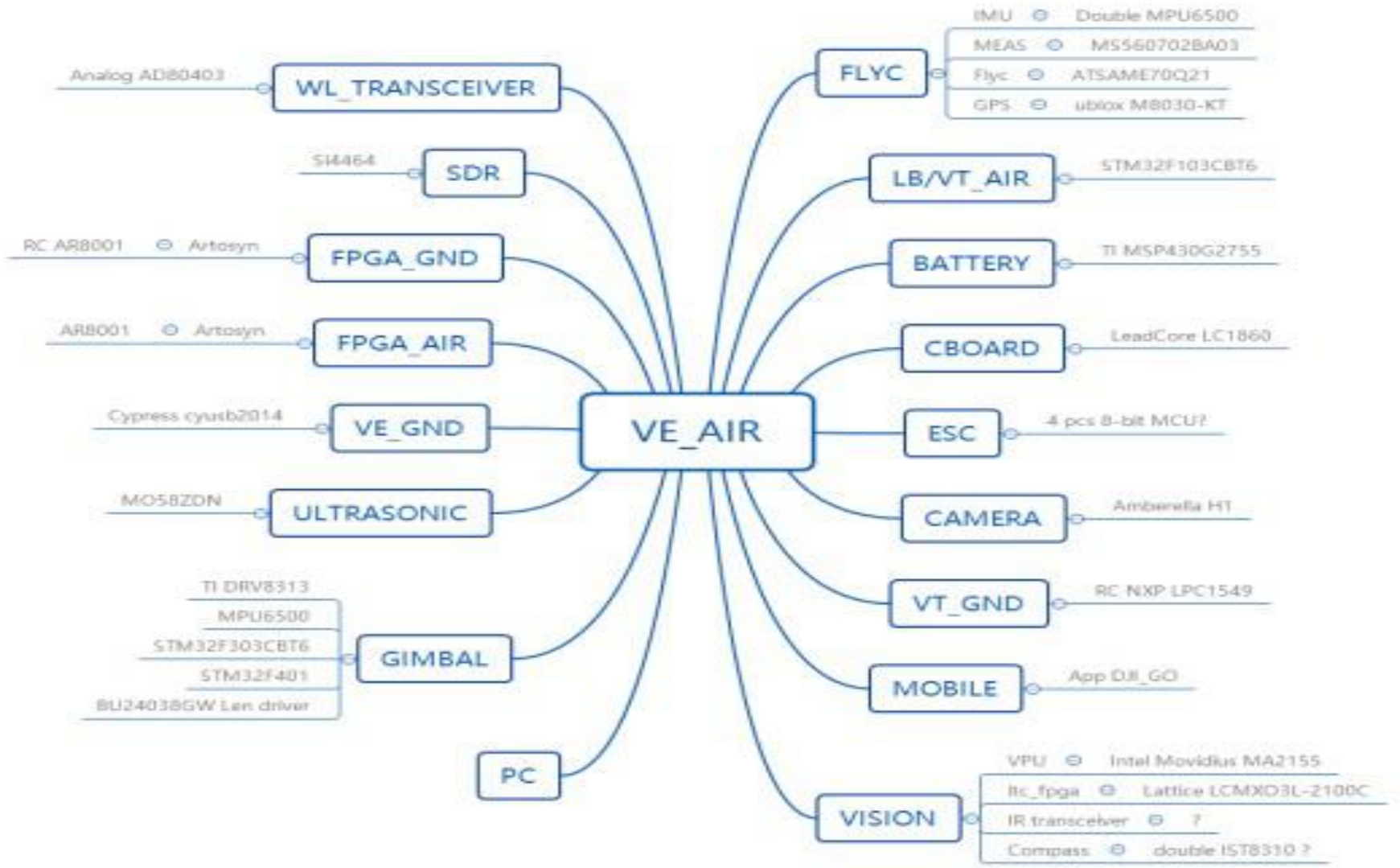
谢君 2017.11.18

无人机是个复杂的系统工程

DJI无人机

- 飞控系统
  - 电调控制
  - 陀螺仪
  - 加速度计
  - 气压计
  - GPS
  - 马达
- 避障测距定位系统
  - 视觉
  - 超声波
  - 红外
  - 指南针
- 图像采集
  - 航拍摄像头
  - 云台
  - 图像压缩
- 无线通信
  - 图像传输
  - 控制传输
- 电源管理
  - 智能电源管理

DJI P4P架构


VE_AIR
WL_TRANSCEIVER
Analog AD80403
SDR
Si4464
FPGA_GND
RC AR8001
Artosyn
FPGA_AIR
AR8001
Artosyn
VE_GND
Cypress cyusb2014
ULTRASONIC
MO58ZDN
GIMBAL
TI DRV8313
MPU6500
STM32F303CBT6
STM32F401
BU24038GW Len driver
PC
FLYC
IMU
Double MPU6500
MEAS
MS560702BA03
Flyc
ATSAME70Q21
GPS
ublox M8030-KT
LB/VT_AIR
STM32F103CBT6
BATTERY
TI MSP430G2755
CBOARD
LeadCore LC1860
ESC
4 pcs 8-bit MCU?
CAMERA
Ambarella H1
VT_GND
RC NXP LPC1549
MOBILE
App DJI_GO
VISION
VPU
Intel Movidius MA2155
ltc_fpga
Lattice LCMXO3L-2100C
IR transceiver
?
Compass
double IST8310 ?

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

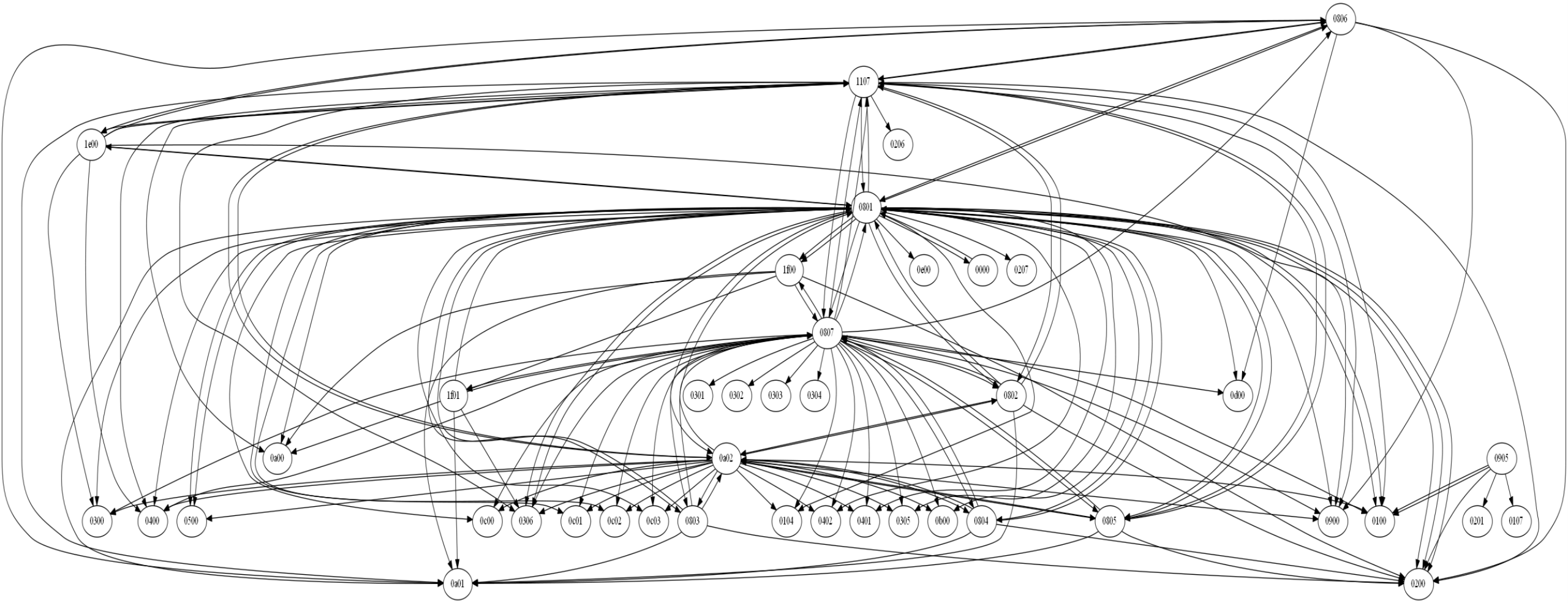
例如03表示飞控，06表示6号子功能

- 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 ma2155
0803 ltc_fpga
0804 ultrasonic
0805 dji_vision
0806 dji_decoding
0807 test_diag
0a00 pc con
0a02 firmware update from pc
1107 ma2155 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


0806
1107
1e00
0206
0801
1f00
0e00
0000
0207
0807
1f01
0301
0302
0303
0304
0802
0d00
0a00
0a02
0905
0300
0400
0500
0c00
0306
0c01
0c02
0c03
0803
0104
0402
0401
0305
0b00
0804
0805
0900
0100
0201
0107
0a01
0200

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 00 06 03 01  PV2.............
03 02 03 02 05 43 02 00 00 00 8F 8B ·····  .....C....弸e
```

DJI构架-模块间通信

- 通信协议Logic&V1

  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 ->

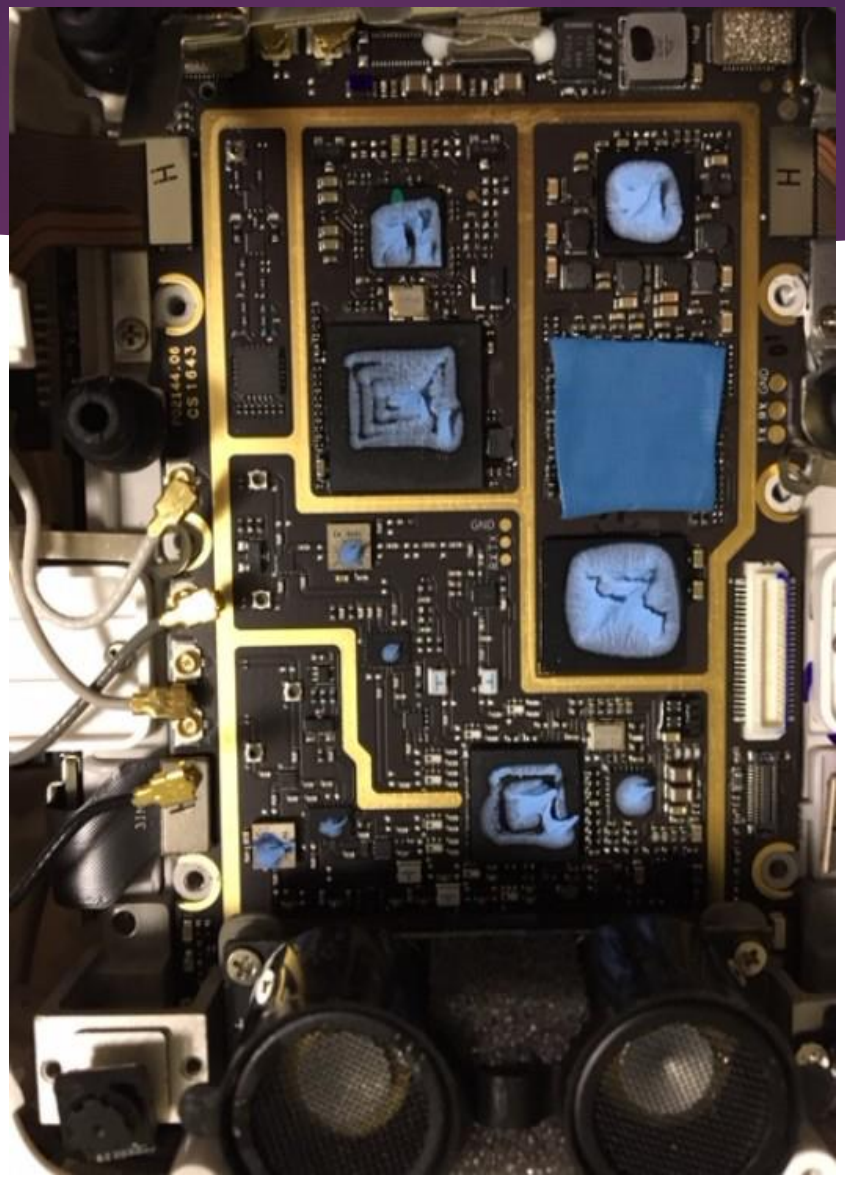
  pkt len 2bytes -> packet len-6

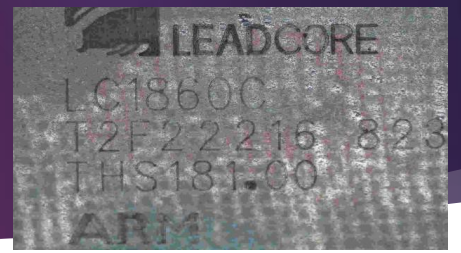
```
def gen_host(d):
    c=(d>>5)|((d&0x1f)<<8)
    return c

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

```
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
```

怎么root？

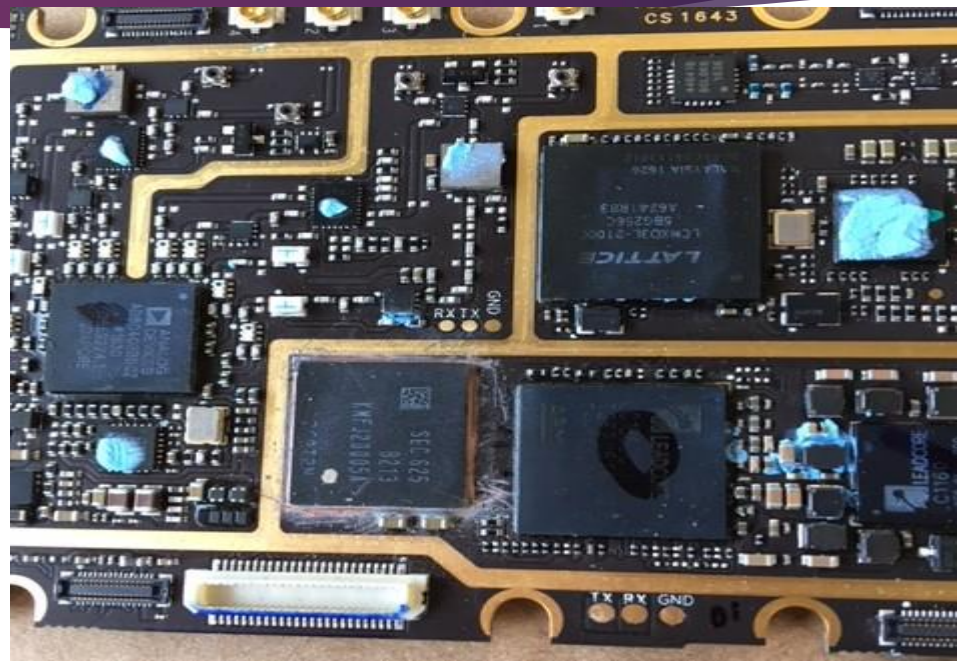
1.软件漏洞执行adb_en.sh脚本

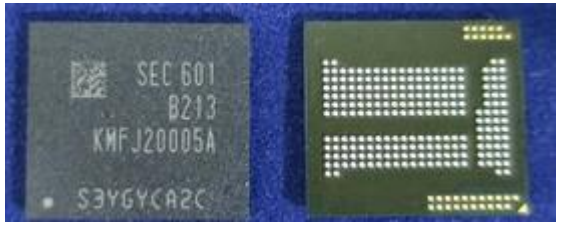
2.如果能够修改start_dji_system.sh脚本

“Debug=true” 也可以达到root目的

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

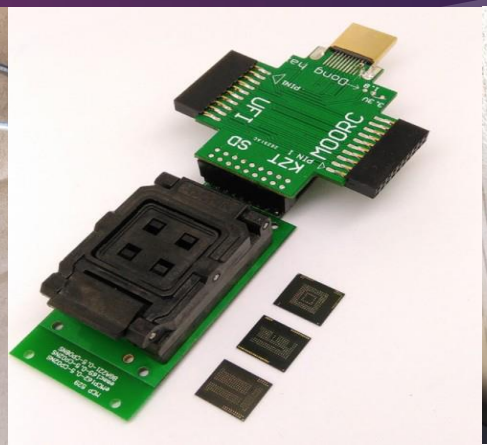


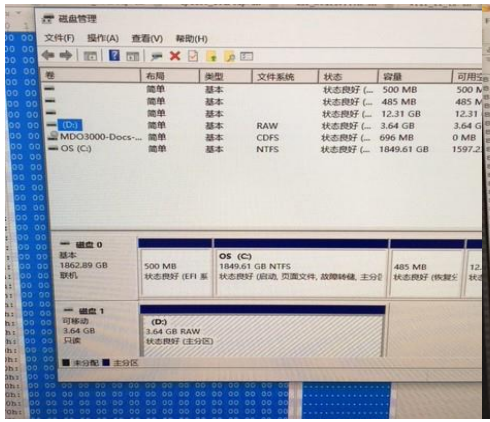
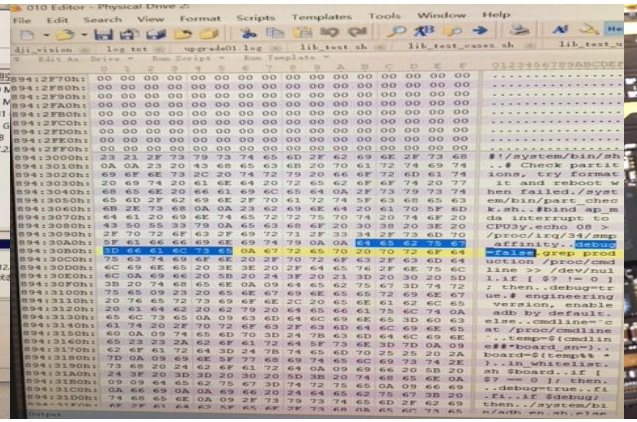
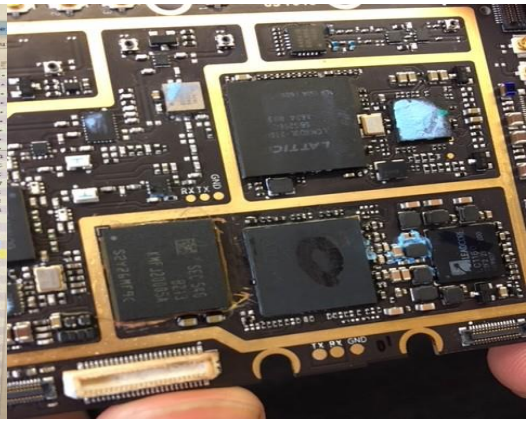


Root无人机

- 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

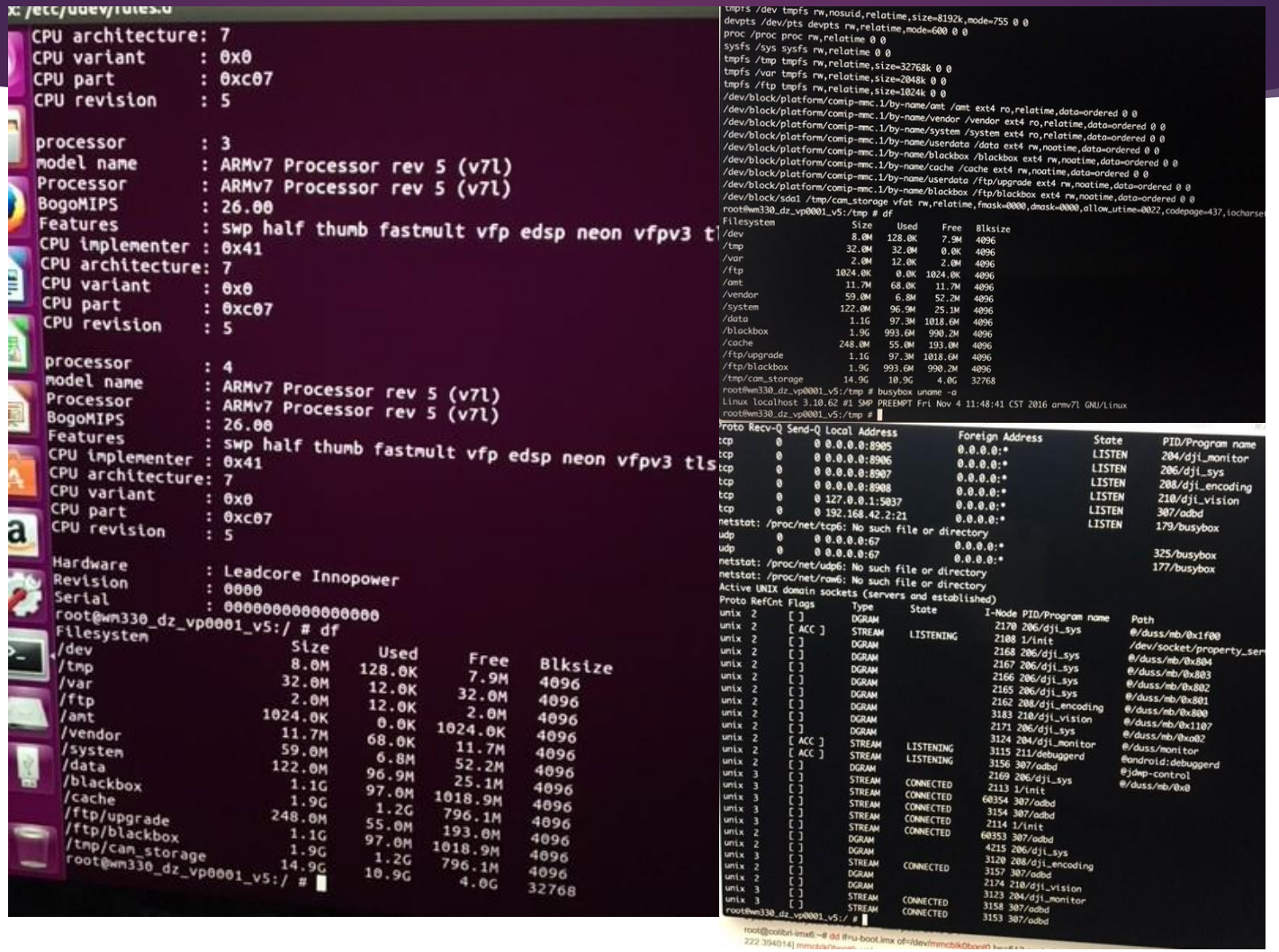




Root无人机

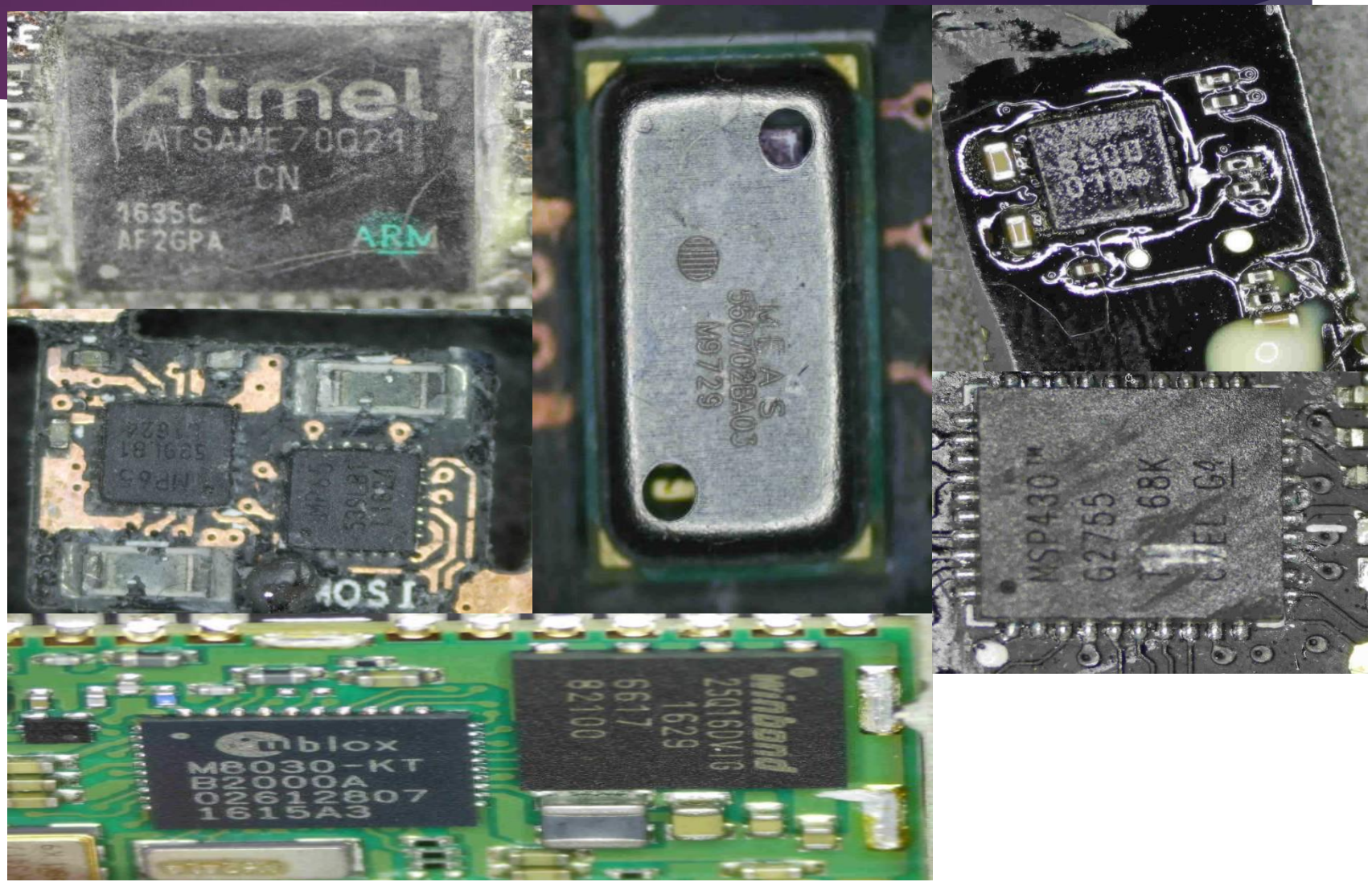
- finally



飞控系统

- 飞控MCU　Atmel ATSAME 70Q21
- 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

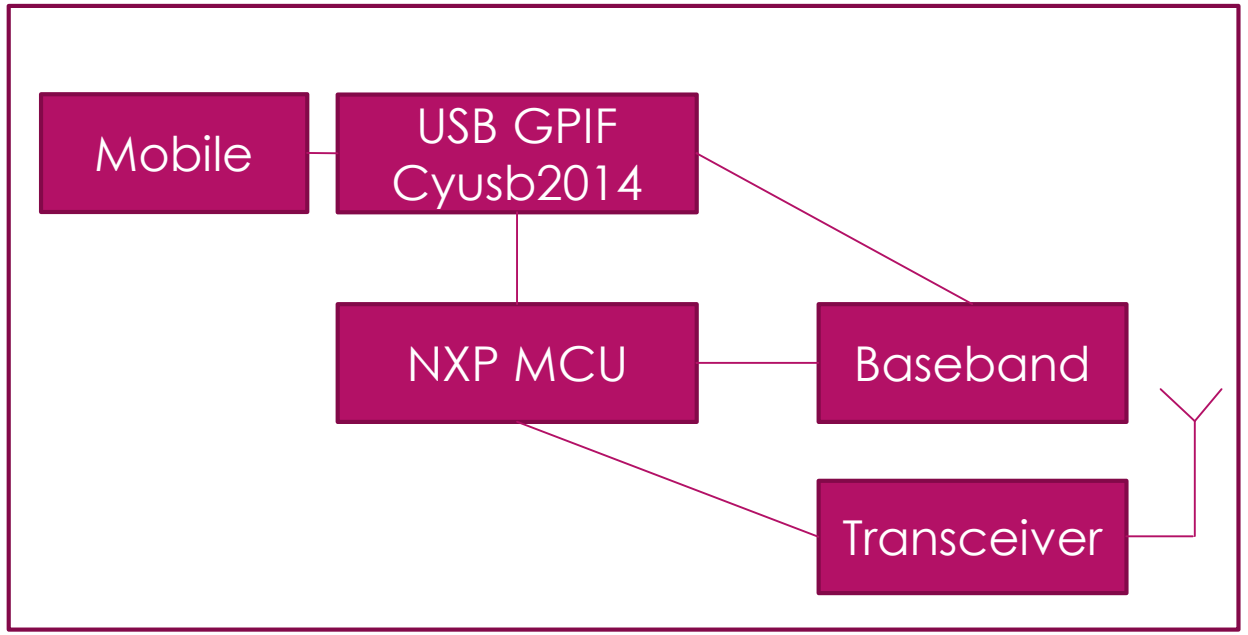


视觉系统

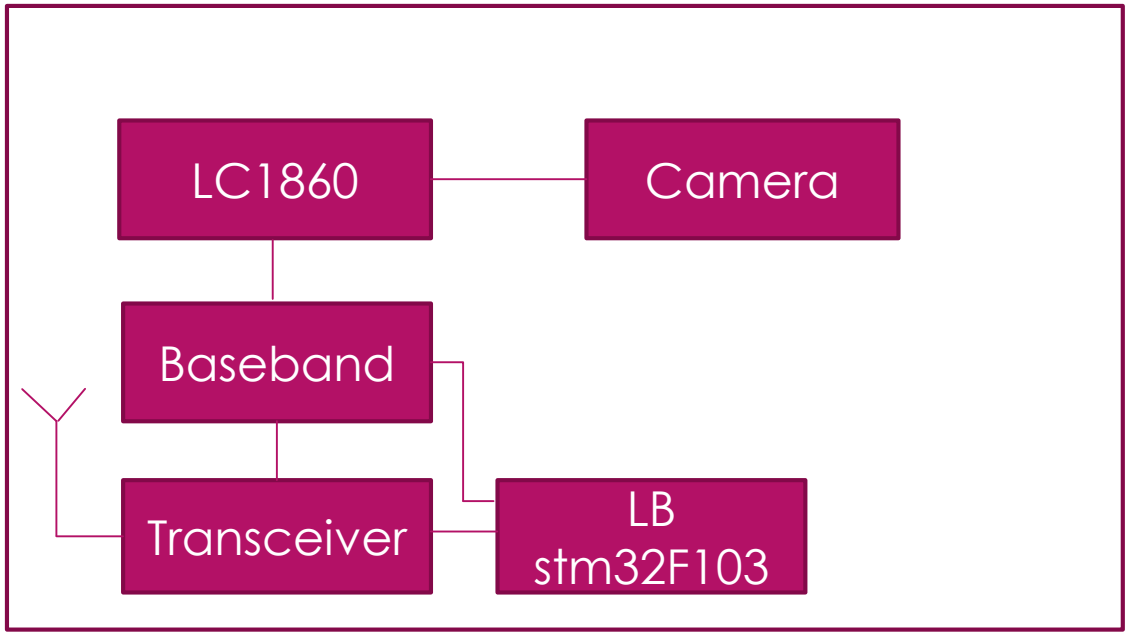
- MA2155深度视觉处理
  - 前后下障碍物识别与测距
  - 人物,姿态,水,ROI识别
- 红外双目避障
  - 左右红外识别
- 超声波
  - 检测下方障碍物

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
2.404-2.4768 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

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]

```
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
```

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

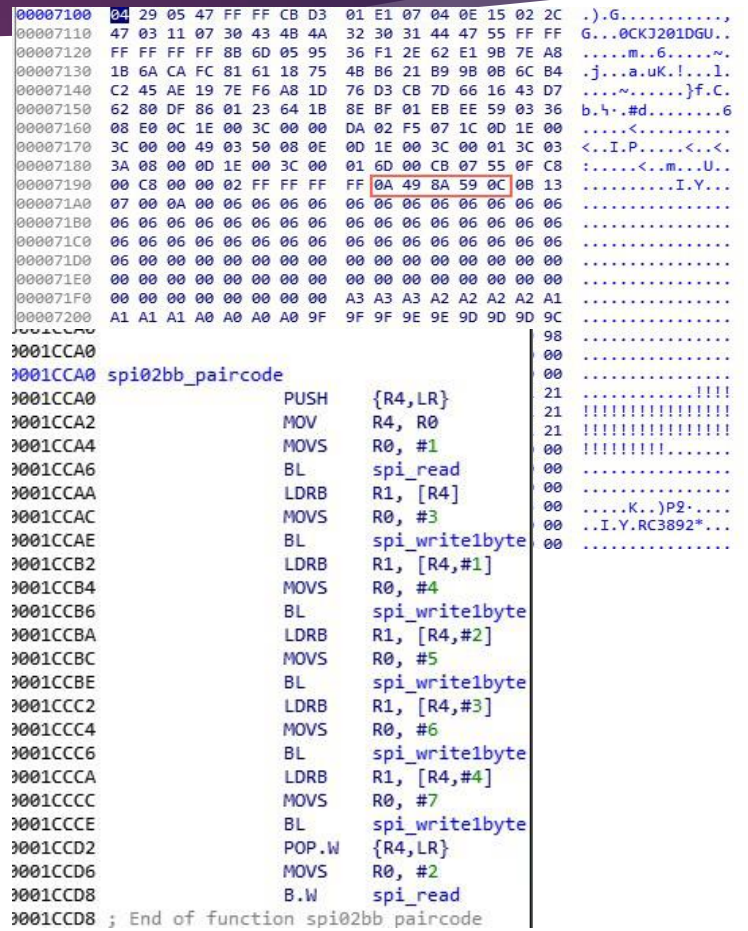
['5745.0', '5803.0', '5764.0', '5821.0', '5782.0', '5743.0', '5801.0', '5761.0', '5819.0', '5780.0', '5741.0', '5798.0', '5759.0', '5817.0', '5778.0', '5738.0', '5796.0', '5757.0', '5814.0', '5775.0', '5736.0', '5794.0', '5755.0', '5812.0', '5773.0', '5734.0', '5791.0', '5752.0', '5810.0', '5771.0', '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

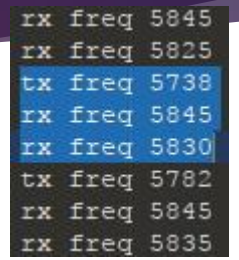
- 写入5字节的配对码到baseband SPI地址3，4，5，6，7

- 写入5字节的配对码



无线通信系统架构

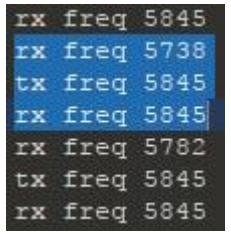
- RC与Air同步原理
- RC跳频循环设置

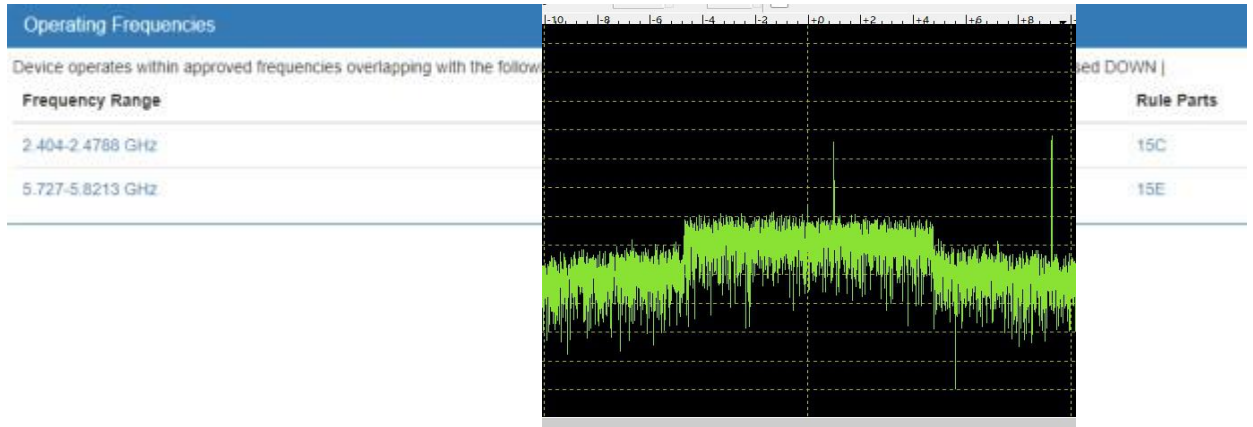


rx跳频设置----- 0.3ms---tx跳频设置----3.075ms-----rx图传设置----13.69ms---rx下一跳频设置

- 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



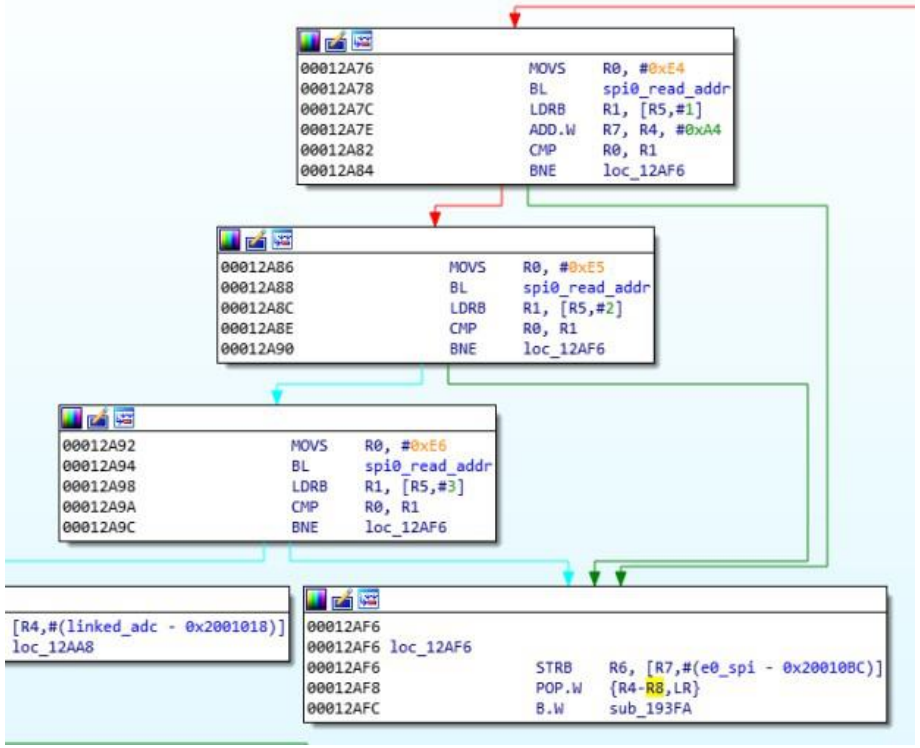


2.400 - 2.483 GHz和5.725 - 5.825 GHz

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

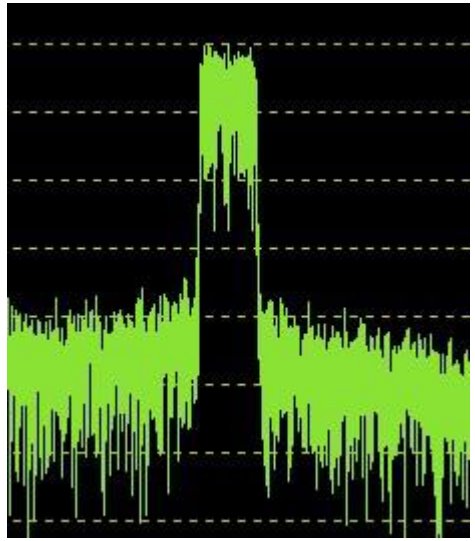
无线通信系统架构

- 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0x1000000
- 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

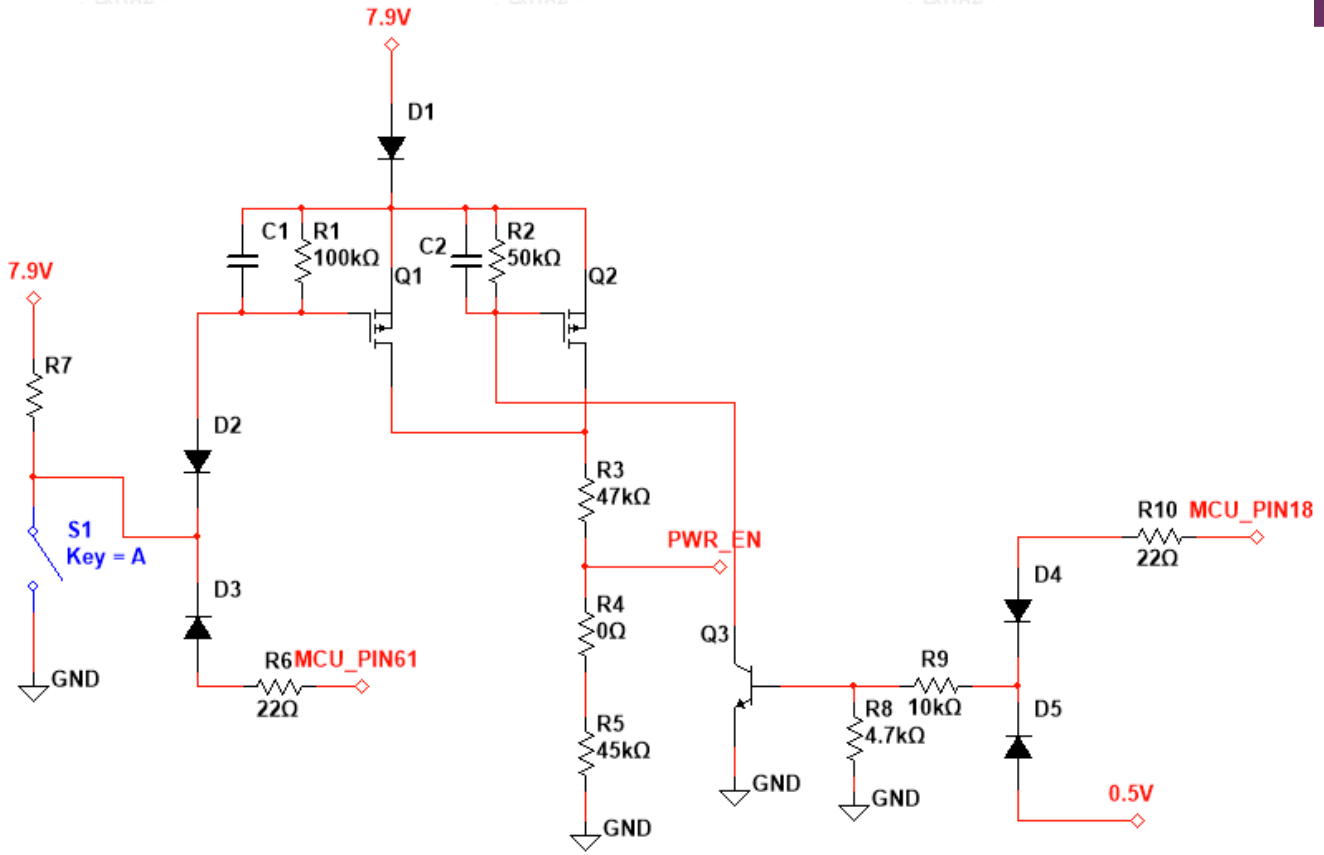
```
SUB             SP, SP, #0x20
BL              sub_800D7B8
MOVS            R1, #1
MOV             R0, R1
BL              RCC_APB2PeriphClockCmd
MOVS            R1, #1
LDR             R0, =0x300400 ; Full SWJ Disabled (JTAG-DP + SW-DP)
BL              GPIO_PinRemapConfig
ADD             R0, SP, #0x20+var_14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

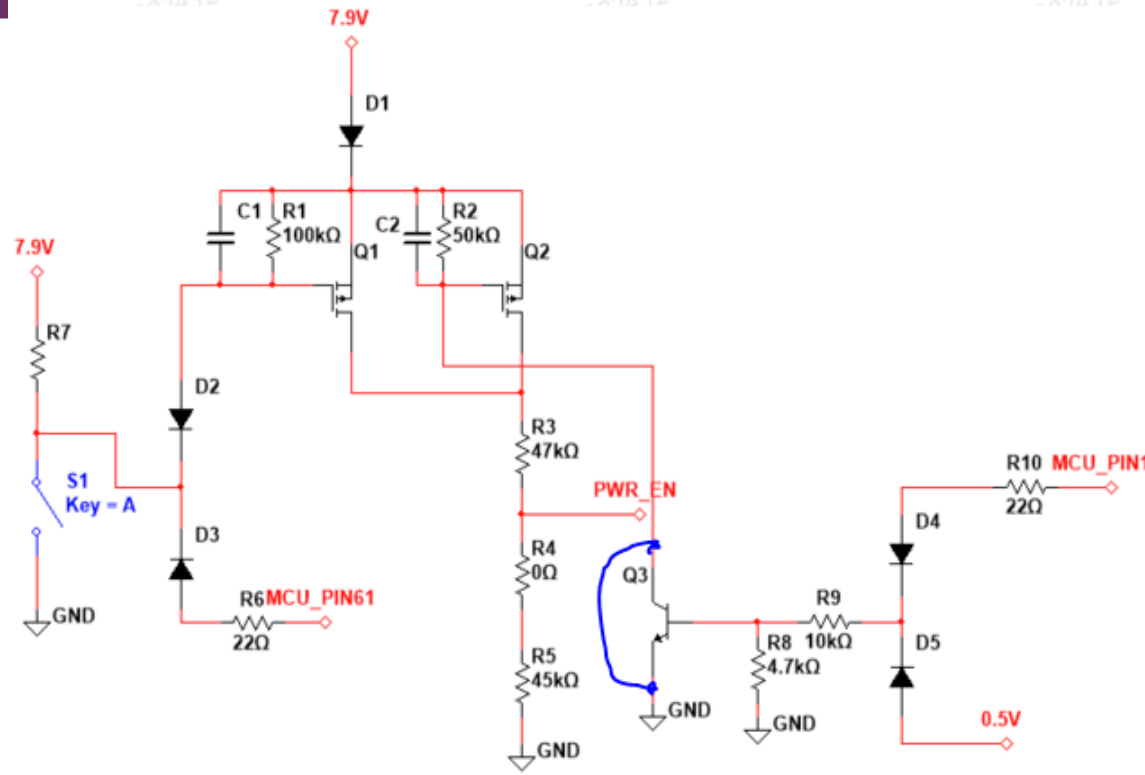
```
00BE14 set_jreset_io_disable                ; CODE XREF: sub_800D588+52↓p
00BE14
00BE14 var_8            = -8
00BE14 var_6            = -6
00BE14 var_5            = -5
```

```
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
```

无人机反调技术

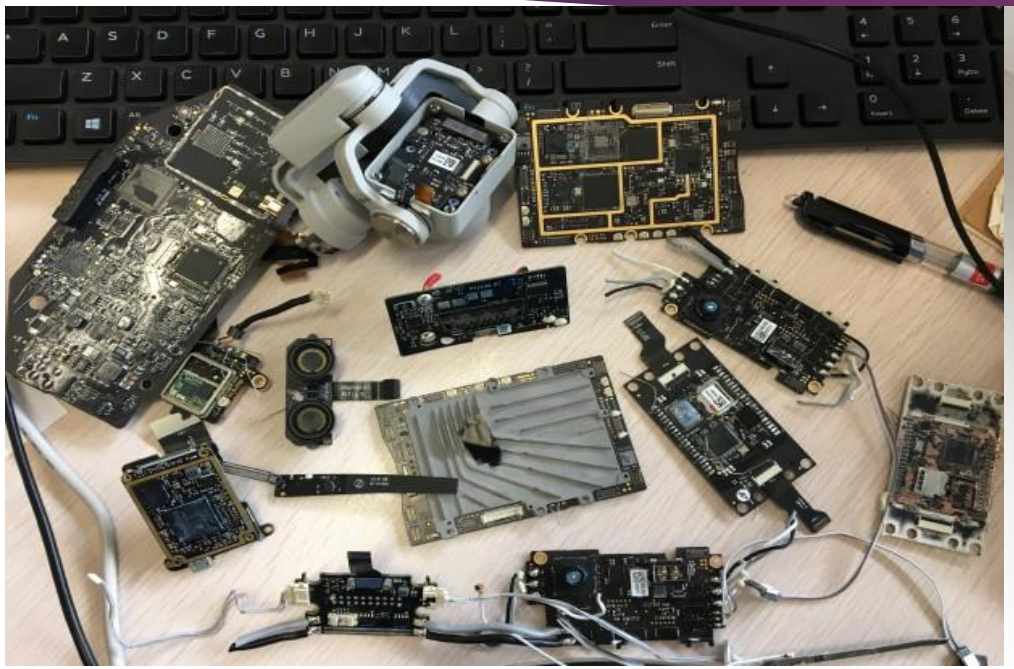

7.9V
D1
C1
R1
100kΩ
C2
R2
50kΩ
Q1
Q2
7.9V
R7
D2
S1
Key = A
D3
GND
R6
MCU_PIN61
22Ω
R3
47kΩ
PWR_EN
R4
0Ω
Q3
R5
45kΩ
GND
R9
R8
10kΩ
4.7kΩ
GND
GND
R10
MCU_PIN18
22Ω
D4
D5
0.5V


7.9V
D1
C1
R1
100kΩ
C2
R2
50kΩ
Q1
Q2
7.9V
R7
D2
S1
Key = A
D3
GND
R6
MCU_PIN61
22Ω
R3
47kΩ
PWR_EN
R4
0Ω
Q3
R5
45kΩ
GND
R9
R8
10kΩ
4.7kΩ
GND
GND
R10
MCU_PIN1
22Ω
D4
D5
0.5V

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