

无线遥控信号分析入门

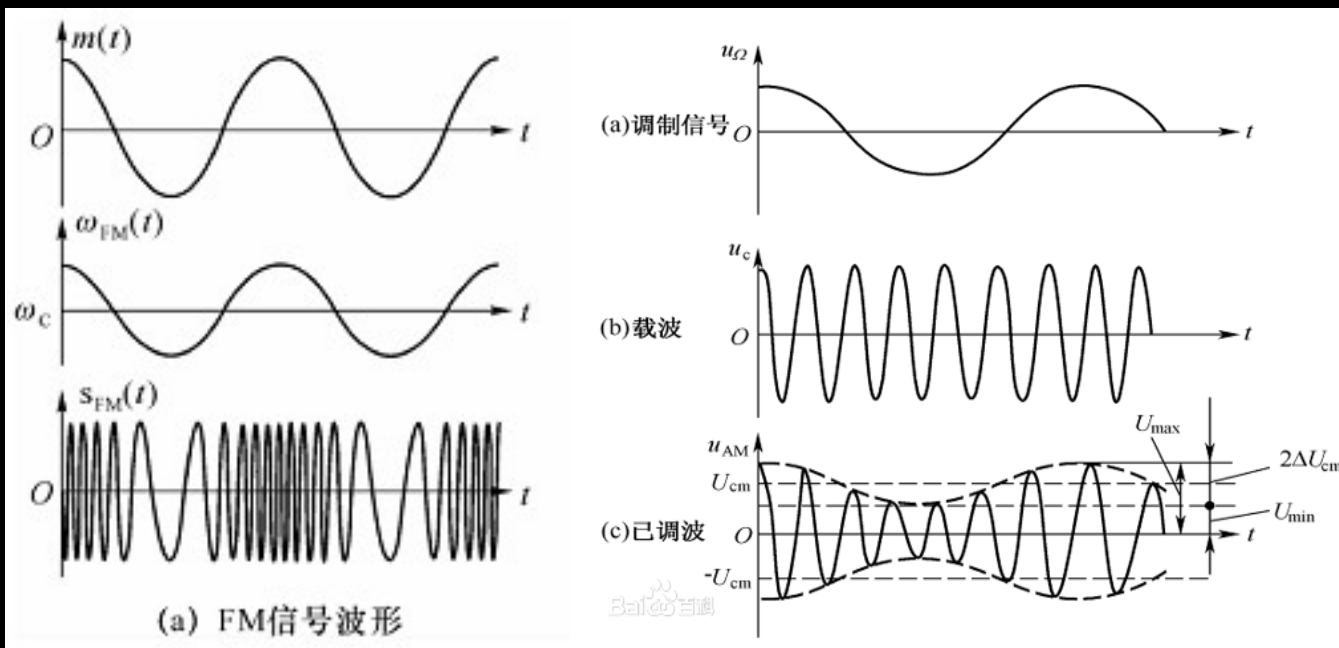
HackPanda@Qingteng

目录

- 无线遥控知识入门
- HackRF及工具使用
- 信号重放攻击
- xx1527信号分析
- eZ430手表开发
- 其他工具介绍

无线遥控知识入门

- 常见频率315、433
- 无线电信号调制
- 遥控器与无线继电器
- 固定码、学习码、滚动码

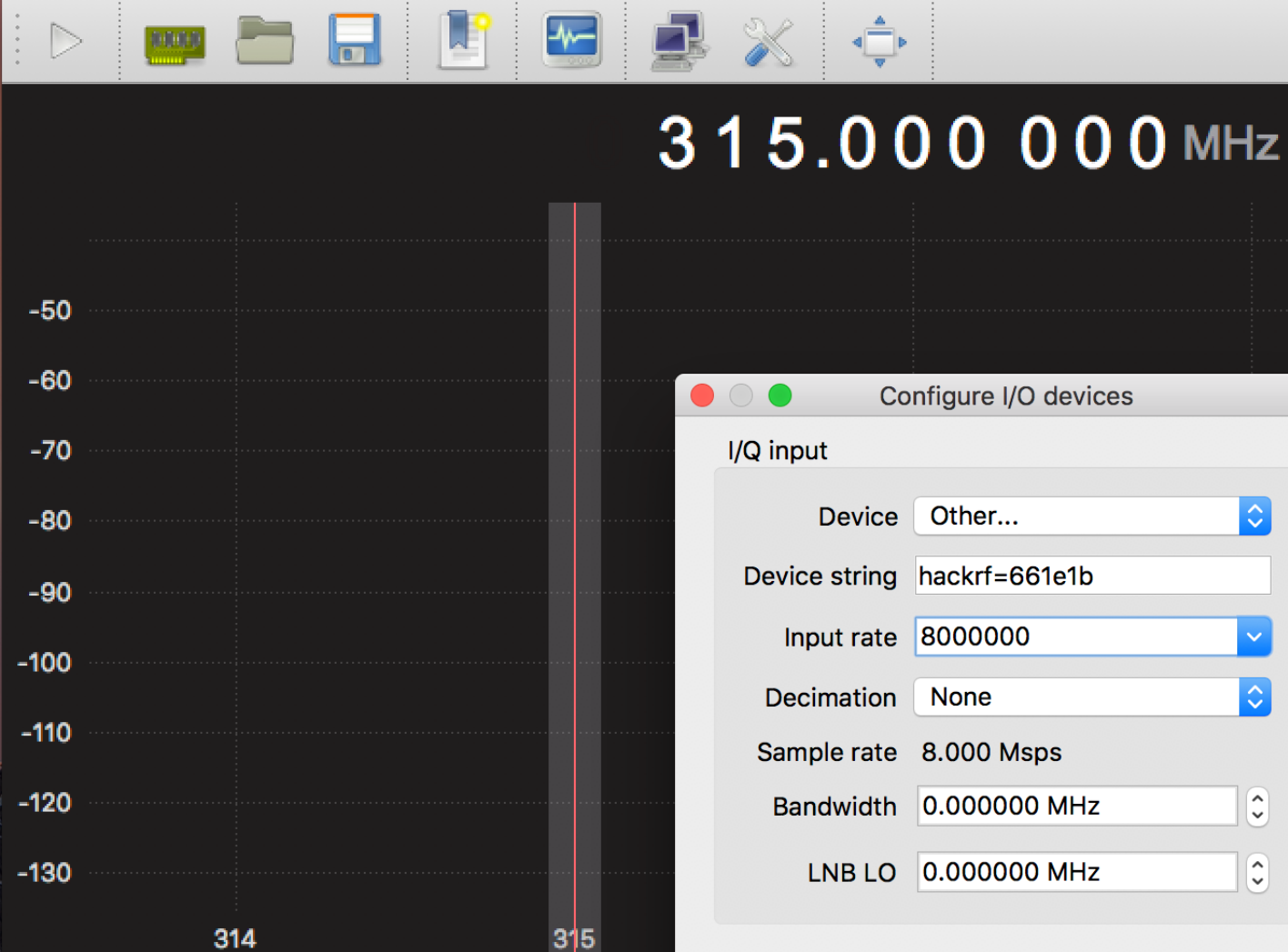


HACKRF及工具使用

- HackRF简介
- gqrx
- hackrf_transfer
- inspectrum

	HackRF		bladeRF		USRP	
			x40	x115	B100 Starter	B200B210
Radio Spectrum	30 MHz – 6 GHz		300 MHz – 3.8 GHz		50 MHz – 2.2 GHz [1]	50MHz – 6 GHz
Bandwidth	20 MHz		28 MHz		16 MHz [2]	61.44 MHz [3]
Duplex	Half		Full		Full	Full2x2 MIMO
Sample Size (ADC/DAC)	8 bit		12 bit		12 bit / 14 bit	12 bit
Sample Rate (ADC/DAC)	20 Msps		40 Msps		64 Msps / 128 Msps	61.44 Msps
Interface (Speed)	USB 2 HS (480 megabit)		USB 3 (5 gigabit)		USB 2 HS (480 megabit)	USB 3 (5 gigabit)
FPGA Logic Elements	[4]		40k	115k	25k	75k 150k
Microcontroller	LPC43XX		Cypress FX3		Cypress FX2	Cypress FX3
Open Source	Everything		HDL + Code Schematics		HDL + Code Schematics	Host Code [5]
Availability	January 2014		Now		Now	Now
Cost	\$300 [6]		\$420	\$650	\$675	\$675 \$1100





-100 -80 -60 -40 -20 0
-120 dBFS

Configure I/O devices

I/Q input

Device Other...

Device string hackrf=661e1b

Input rate 8000000

Decimation None

Sample rate 8.000 Msps

Bandwidth 0.000000 MHz

LNB LO 0.000000 MHz

Audio output

Device Default

Sample rate 48 kHz

Cancel OK

Open file...

Sample rate: 8000000

Spectrogram

FFT size:

Zoom:

Power max:

Power min:

Scales: ☒

Time selection

Enable cursors: ☐

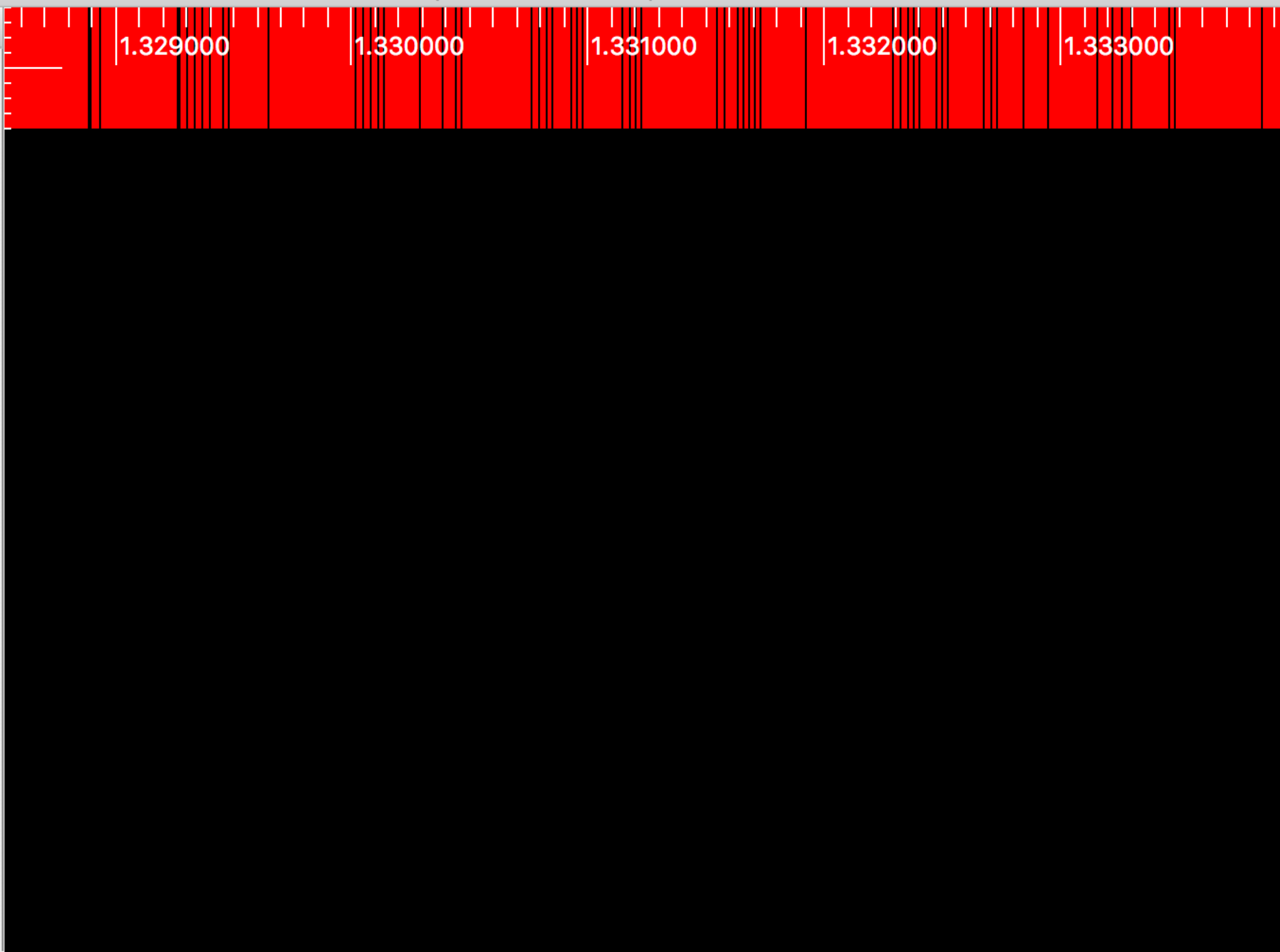
Symbols:

Rate:

Period:

Symbol rate:

Symbol period:




```
117 void InputSource::openFile(const char *filename)
118 {
119     QFileInfo fileInfo(filename);
120     const auto suffix = fileInfo.suffix().toLower();
121     if ((suffix == "cfile") || (suffix == "cf32") || (suffix == "fc32")) {
122         sampleAdapter = std::unique_ptr<SampleAdapter>(new ComplexF32SampleAdapter());
123     }
124     else if ((suffix == "cs16") || (suffix == "sc16") || (suffix == "c16")) {
125         sampleAdapter = std::unique_ptr<SampleAdapter>(new ComplexS16SampleAdapter());
126     }
127     else if ((suffix == "cs8") || (suffix == "sc8") || (suffix == "c8")) {
128         sampleAdapter = std::unique_ptr<SampleAdapter>(new ComplexS8SampleAdapter());
129     }
130     else if ((suffix == "cu8") || (suffix == "uc8")) {
131         sampleAdapter = std::unique_ptr<SampleAdapter>(new ComplexU8SampleAdapter());
132     }
133     else {
134         sampleAdapter = std::unique_ptr<SampleAdapter>(new ComplexF32SampleAdapter());
135     }
136 }
```

inspectrum supports the following file types:

- *.cf32 , *.cfile - Complex 32-bit floating point samples (GNURadio, osmocom_fft)
- *.cs16 - Complex 16-bit signed integer samples (BladeRF)
- *.cs8 - Complex 8-bit signed integer samples (HackRF)
- *.cu8 - Complex 8-bit unsigned integer samples (RTL-SDR)

If an unknown file extension is loaded, inspectrum will default to *.cf32 .

信号重放攻击

使用gqrx确定中心频率



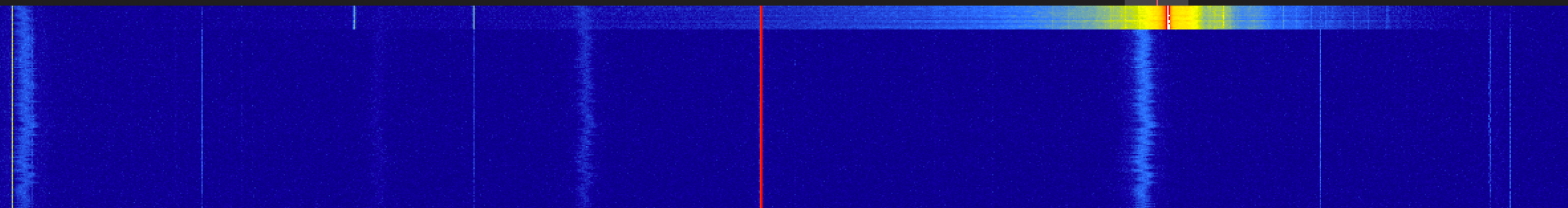
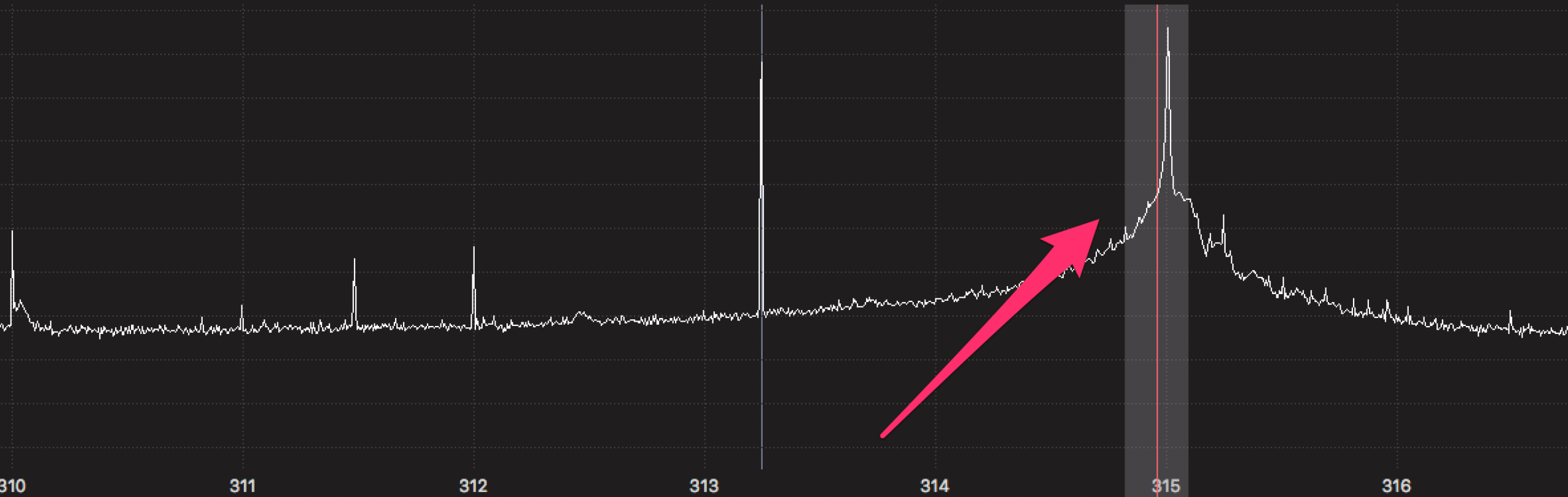
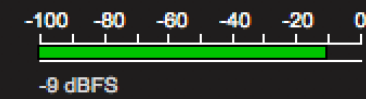
hackrf_transfer录制信号



hackrf_transfer重放信号



314.965 000 MHz



```
ghjk:~ \[REDACTED] hackrf_transfer -r remote.iq -f 315000000 -s 8000000
call hackrf_set_sample_rate(8000000 Hz/8.000 MHz)
call hackrf_set_freq(315000000 Hz/315.000 MHz)
Stop with Ctrl-C
16.0 MiB / 1.000 sec = 16.0 MiB/second
^CCaught signal 2
 5.8 MiB / 0.373 sec = 15.4 MiB/second

Exiting...
Total time: 1.37346 s
hackrf_stop_rx() done
hackrf_close() done
hackrf_exit() done
fclose(fd) done
exit
```

```
ghjk:~ ██████████ hackrf_transfer -t remote.cs8 -f 315000000 -s 8000000 -a 1 -x 47  
call hackrf_set_sample_rate(8000000 Hz/8.000 MHz)  
call hackrf_set_freq(315000000 Hz/315.000 MHz)  
call hackrf_set_amp_enable(1)
```

Stop with Ctrl-C

```
16.0 MiB / 1.005 sec = 15.9 MiB/second  
16.0 MiB / 1.005 sec = 15.9 MiB/second  
16.0 MiB / 1.000 sec = 16.0 MiB/second  
16.0 MiB / 1.001 sec = 16.0 MiB/second  
16.3 MiB / 1.005 sec = 16.2 MiB/second  
16.0 MiB / 1.002 sec = 16.0 MiB/second  
16.0 MiB / 1.000 sec = 16.0 MiB/second  
16.0 MiB / 1.005 sec = 15.9 MiB/second  
16.0 MiB / 1.003 sec = 15.9 MiB/second  
16.3 MiB / 1.005 sec = 16.2 MiB/second  
16.0 MiB / 1.001 sec = 16.0 MiB/second  
3.9 MiB / 1.000 sec = 3.9 MiB/second
```

Exiting... hackrf_is_streaming() result: streaming terminated (-1004)

Total time: 12.03285 s

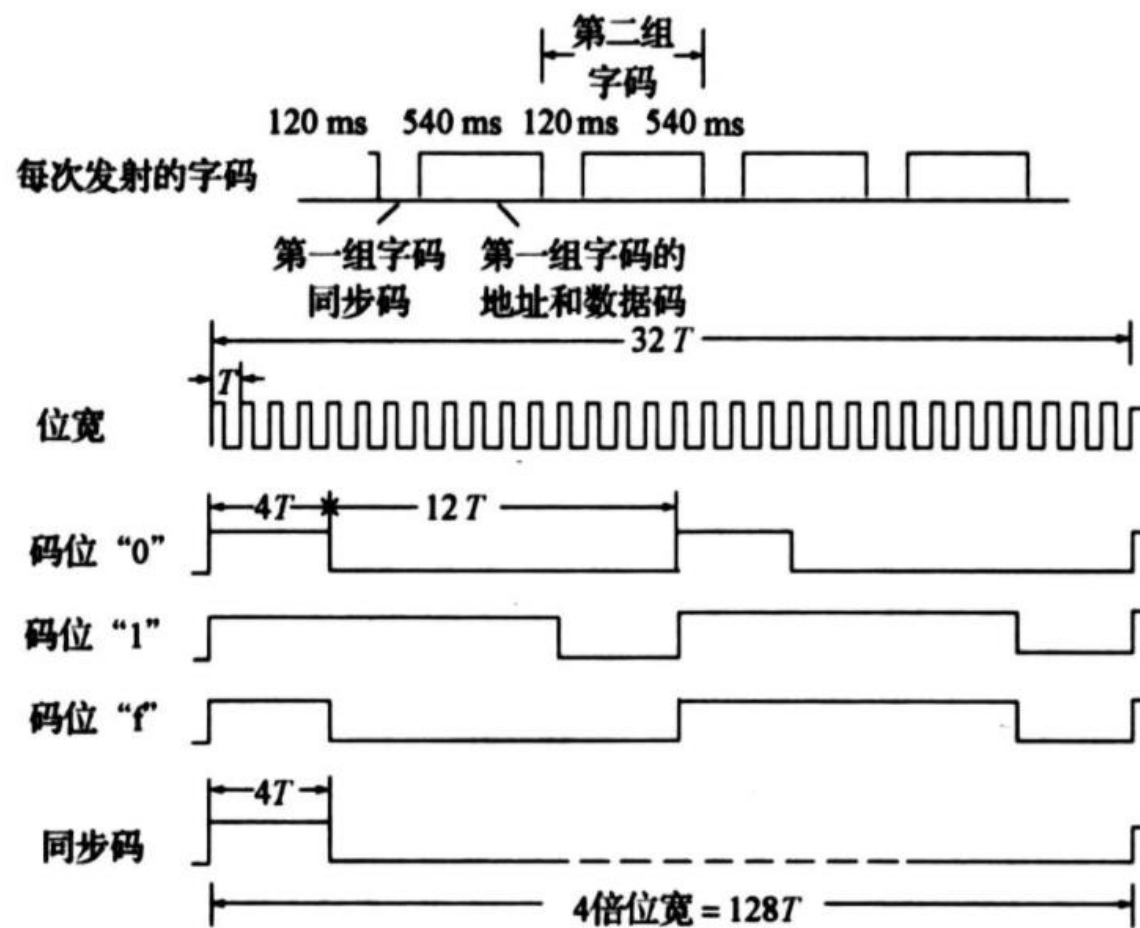
hackrf_stop_tx() done

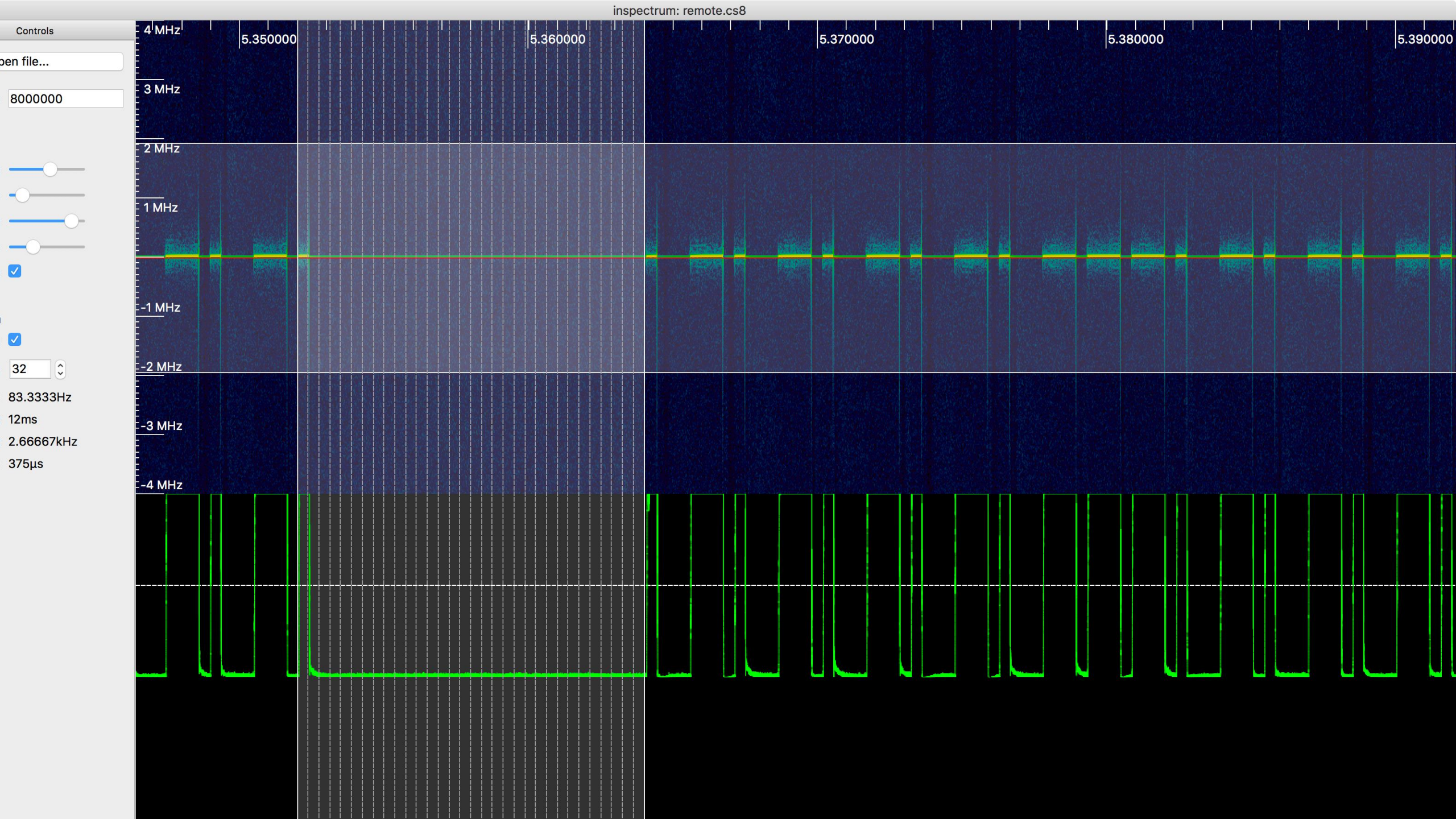
hackrf_close() done



XX1527信号分析

- 学习码遥控器
- 同步码、地址码、数据码



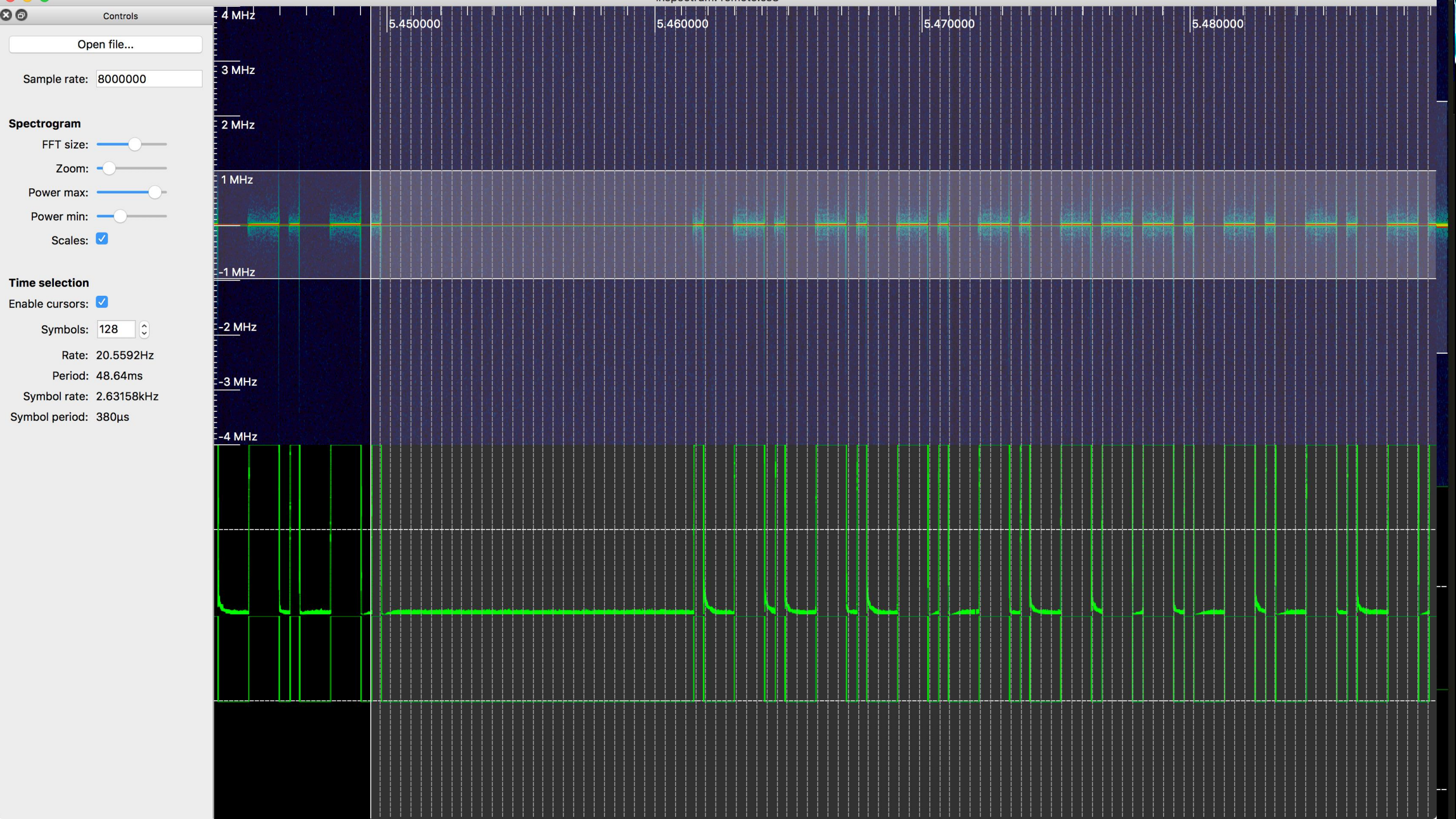


XX1527信号分析

分割信号为同步码、地址码、数据码



对后12位的脉冲宽度识别并转换成高低电平




```
ghjk:~ [REDACTED] inspectrum
```

```
objc[6026]: Class FIFinderSyncExtensionHost is implemented in both /System/Library/PrivateFrameworks/FinderKit.framework/Versions/A/FinderKit (0x7fff8b5b1b68) and /System/Library/PrivateFrameworks/FileProvider.framework/OverrideBundles/FinderSyncCollaborationFileProviderOverride.bundle/Contents/MacOS/FinderSyncCollaborationFileProviderOverride (0x1307d7cd8). One of the two will be used. Which one is undefined.
```

[illegible]

```
2018-03-07 19:44:21.013 inspectrum[6026:1155780] IMKInputSession [0x7f8f5912d490 pres
Context]=0x7f8f53e1d7c0 *NO* NSRemoteViewController to client, NSError=Error Domain=N
process." UserInfo={NSDebugDescription=The connection from pid 0 was invalidated from
2018-03-07 19:44:45.981 inspectrum[6026:1155780] IMKInputSession [0x7f8f5912d490 pres
Context]=0x7f8f53e1d7c0 *NO* NSRemoteViewController to client, NSError=Error Domain=N
process." UserInfo={NSDebugDescription=The connection from pid 0 was invalidated from
```

```

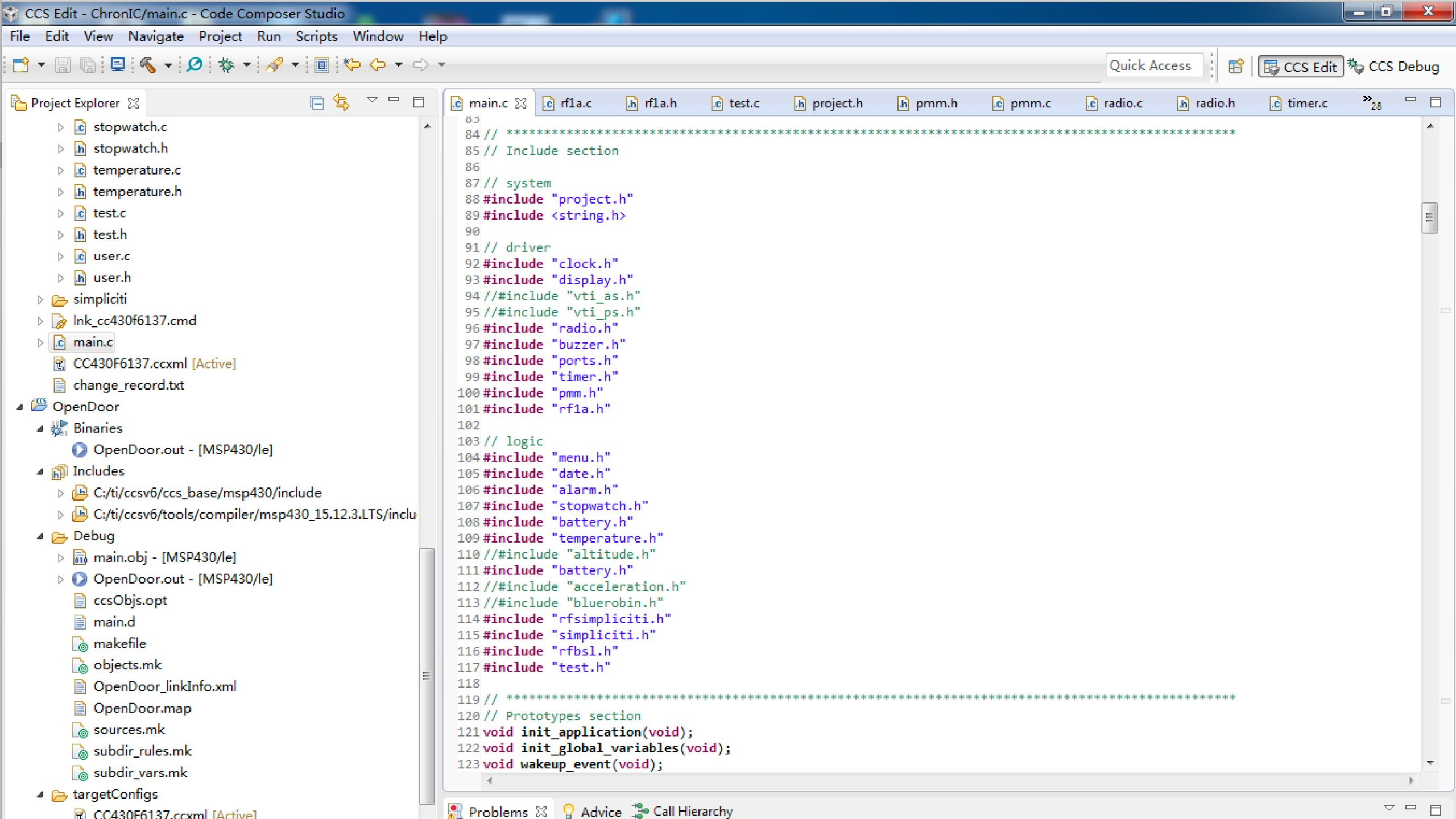
1, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0,
1, 0, 0, 0, 1, 1, 1, 0, f
1, 0, 0, 0, 1, 1, 1, 0, f
1, 0, 0, 0, 1, 1, 1, 0, f
1, 0, 0, 0, 1, 1, 1, 0, f
1, 0, 0, 0, 1, 1, 1, 0, f
1, 1, 1, 0, 1, 1, 1, 0, 1
1, 0, 0, 0, 1, 1, 1, 0, f
1, 0, 0, 0, 1, 1, 1, 0, f
1, 0, 0, 0, 1, 1, 1, 0, f
1, 0, 0, 0, 1, 1, 1, 0, f
1, 0, 0, 0, 1, 1, 1, 0, f
1, 0, 0, 0, 1, 1, 1, 0, f
1, 0, 0, 0, 1, 1, 1, 0, f
fffff1ffff ff
80 00 00 00
8e 8e 8e 8e 8e ee 8e 8e 8e 8e
8e 8e
\x80\x00\x00\x00\x8e\x8e\x8e

```

EZ430手表开发

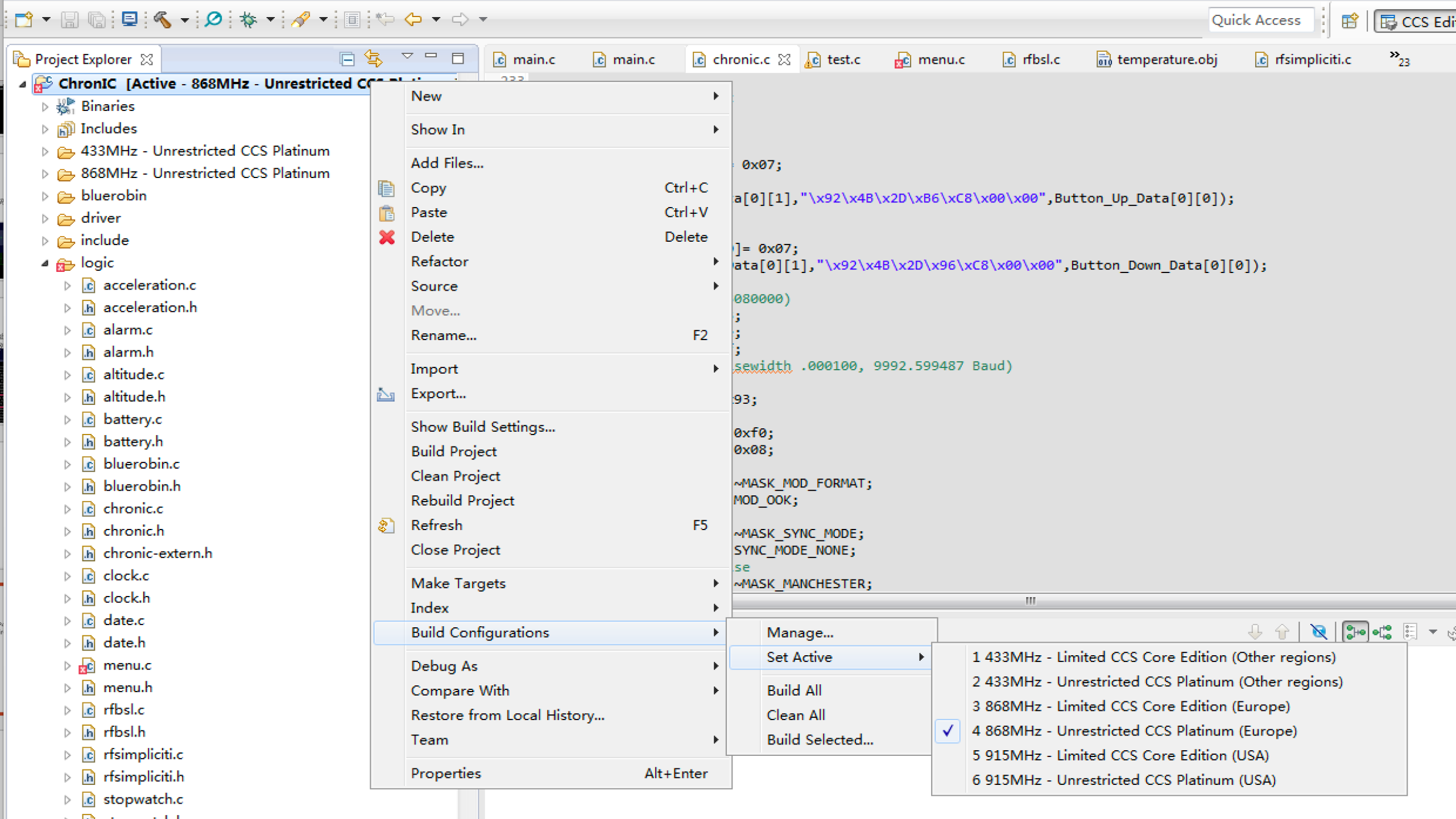
- 基于 CC430F6137、MSP430 的完整功能的运动手表，带可针对定制无线应用 <1GHz wireless transceiver
- 使用CCS开发





EZ430手表开发

- 配置手表菜单
- 定义发射功能函数、显示函数等
- Hex发送电平
- 基于26mhz晶振计算频率、波特率
- 编译并把生成的文件写入手表



Project Explorer

main.c main.c chronic.c test.c menu.c rfbsl.c temperature.obj rfsimpliciti.c »23

ChronIC [Active - 868MHz - Unrestricted CCS Platinum]

- Binaries
- Includes
- 433MHz - Unrestricted CCS Platinum
- 868MHz - Unrestricted CCS Platinum
- bluerobin
- driver
- include
- logic
 - acceleration.c
 - acceleration.h
 - alarm.c
 - alarm.h
 - altitude.c
 - altitude.h
 - battery.c
 - battery.h
 - bluerobin.c
 - bluerobin.h
 - chronic.c
 - chronic.h
 - chronic-extern.h
 - clock.c
 - clock.h
 - date.c
 - date.h
 - menu.c
 - menu.h
 - rfbsl.c
 - rfbsl.h
 - rfsimpliciti.c
 - rfsimpliciti.h
 - stopwatch.c

- New
- Show In
- Add Files...
- Copy Ctrl+C
- Paste Ctrl+V
- Delete Delete
- Refactor
- Source
- Move...
- Rename... F2
- Import
- Export...
- Show Build Settings...
- Build Project
- Clean Project
- Rebuild Project
- Refresh F5
- Close Project
- Make Targets
- Index
- Build Configurations
- Debug As
- Compare With
- Restore from Local History...
- Team
- Properties Alt+Enter

```
0x07;  
a[0][1], "\x92\x4B\x2D\xB6\xC8\x00\x00", Button_Up_Data[0][0]);  
  
]= 0x07;  
ata[0][1], "\x92\x4B\x2D\x96\xC8\x00\x00", Button_Down_Data[0][0]);  
  
080000)  
;  
;  
;  
sewidth .000100, 9992.599487 Baud)  
  
93;  
  
0xf0;  
0x08;  
  
~MASK_MOD_FORMAT;  
MOD_OOK;  
  
~MASK_SYNC_MODE;  
SYNC_MODE_NONE;  
se  
~MASK_MANCHESTER;
```

- Manage...
- Set Active
- Build All
- Clean All
- Build Selected...

- 1 433MHz - Limited CCS Core Edition (Other regions)
- 2 433MHz - Unrestricted CCS Platinum (Other regions)
- 3 868MHz - Limited CCS Core Edition (Europe)
- ☒ 4 868MHz - Unrestricted CCS Platinum (Europe)
- 5 915MHz - Limited CCS Core Edition (USA)
- 6 915MHz - Unrestricted CCS Platinum (USA)

```

main.c chronic.c rfla.c chronic.h chronic-ext... *menu.c timer.c bluerobin.h 31
drivers.lib
134 };
135
136 // Line1 - Alarm
137 const struct menu menu_L1_Alarm = {
138     FUNCTION(sx_alarm),           // direct function
139     FUNCTION(mx_alarm),           // sub menu function
140     FUNCTION(display_alarm),       // display function
141 //     &menu_L1_Temperature,
142     &menu_L1_Doorbell,
143 };
144
145 // Line1 - Doorbell
146 const struct menu menu_L1_Doorbell = {
147     FUNCTION(config_doorbell),     // direct function
148     FUNCTION(dummy),               // sub menu function
149     FUNCTION(display_doorbell),    // display function
150 #else
151     &menu_L1_Test,
152 #endif
153 };
154
155 const struct menu menu_L1_Test = {
156     FUNCTION(config_test),         // direct function
157     FUNCTION(dummy),               // sub menu function
158     FUNCTION(display_test),        // display function
159     &menu_L1_Home,
160 };
161
162 const struct menu menu_L1_Home = {
163     FUNCTION(config_home),         // direct function
164     FUNCTION(dummy),               // sub menu function
165     FUNCTION(display_home),        // display function
166     &menu_L1_Time,
167 };
168
169 //// Line1 - Temperature
170 //const struct menu menu_L1_Temperature = {
171 //    FUNCTION(dummy),               // direct function
172 //    FUNCTION(mx_temperature),      // sub menu function
173 //    FUNCTION(display_temperature)  // display function

```



23

Full License

```

329 void config_test(u8 line)
330 {
331     // gap between data pulses
332     Button_Delay= 0;
333     // how many times to send per button ;
334     Button_Repeat= 6;
335
336     // set button content
337
338     Up_Buttons= 1;
339     // packet length
340     Button_Up_Data[0][0]= 0x10;
341     // payload
342     memcpy(&Button_Up_Data[0][1], "\x80\x00\x00\x00\x8e\x8e\x8e\x8e\x8e\x8e\xee\x8e\x8e\x8e\x8e\x8e\x8e", Button_Up_Data[0][0]);
343
344     Down_Buttons= 1;
345     Button_Down_Data[0][0]= 0x10;
346     memcpy(&Button_Down_Data[0][1], "\x80\x00\x00\x00\x8e\x8e\x8e\x8e\x8e\x8e\xee\x8e\x8e\x8e\x8e\x8e\x8e", Button_Down_Data[0][0]);
347
348     // set frequency (315000000)
349     ChronicRF.freq0= 0x89;
350     ChronicRF.freq1= 0x1D;
351     ChronicRF.freq2= 0x0C;
352     // set data rate (pulsewidth .000376, 2659.574468085 Baud)
353     // drate_m
354     ChronicRF.mdmcfg3= 0xAD;
355     // drate_e
356     ChronicRF.mdmcfg4 &= 0xf0;
357     ChronicRF.mdmcfg4 |= 0x06;
358     // set modulation
359     ChronicRF.mdmcfg2 &= ~MASK_MOD_FORMAT;
360     ChronicRF.mdmcfg2 |= MOD_OOK;
361     // set sync mode
362     ChronicRF.mdmcfg2 &= ~MASK_SYNC_MODE;
363     ChronicRF.mdmcfg2 |= SYNC_MODE_NONE;
364     // set manchester false
365     ChronicRF.mdmcfg2 &= ~MASK_MANCHESTER;
366     display_symbol(LCD_ICON_RECORD, SEG_ON);
367     Emulation_Mode= EMULATION_MODE_TEST;
368 }

```

The TX & RX data rate is programmable from
1.2 to 500 kBaud using the following registers:

- MDMCFG3.DRATE_M
- MDMCFG4.DRATE_E



$$R_{DATA} = \frac{(256 + DRATE_M) \times 2^{DRATE_E}}{2^{28}} \times f_{XOSC}$$

```

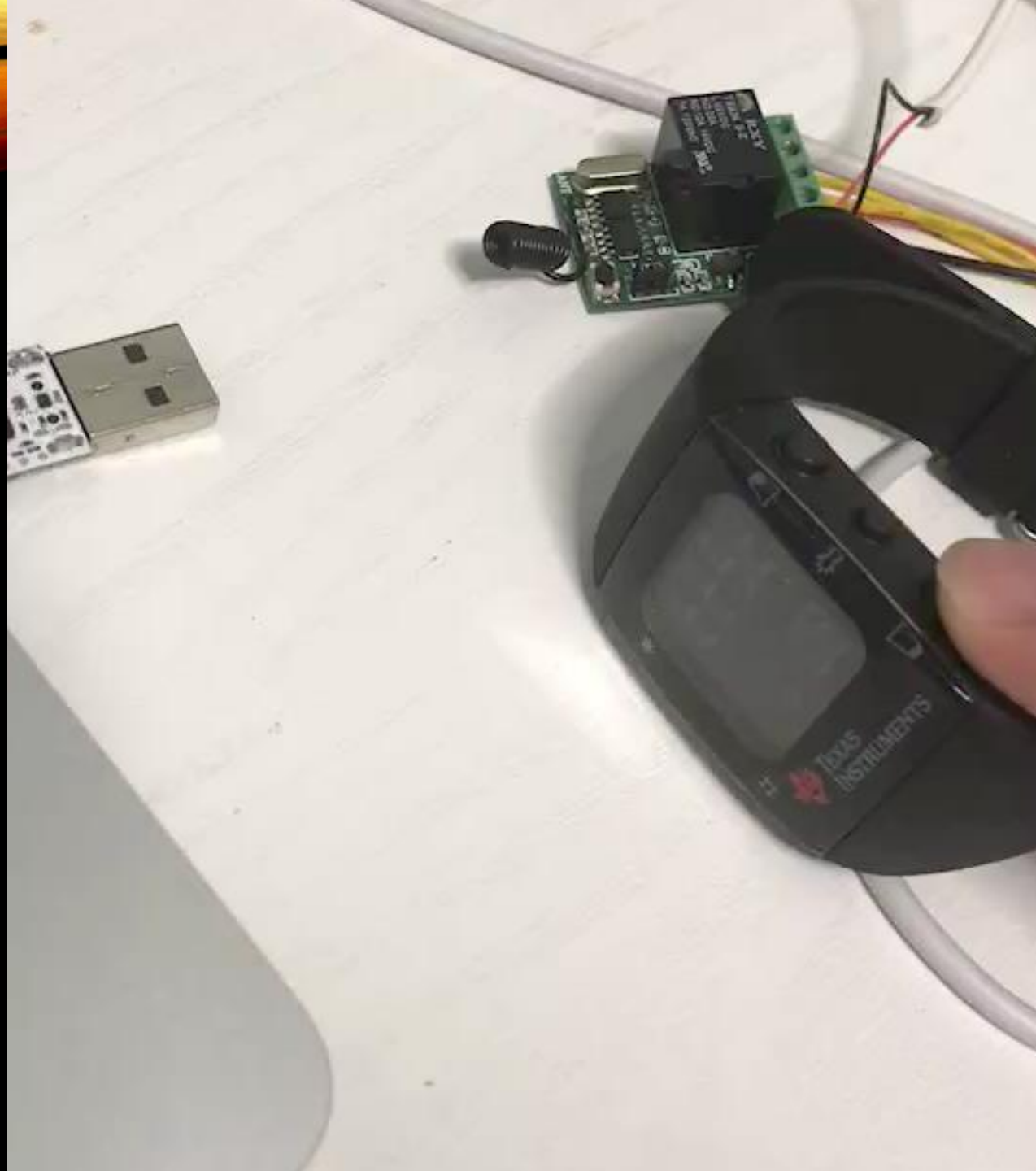
def setfreq(freq):
    mhz= 26
    freqmult = (0x10000 / 1000000.0) / mhz
    num = int(freq * freqmult)
    freq0= num & 0xff
    payload= chr(freq0)
    freq1= (num >> 8) & 0xff
    payload += chr(freq1)
    freq2= (num >> 16) & 0xff
    payload += chr(freq2)
    print '- FREQ2: %02x FREQ1: %02x FREQ0: %02x -' % (freq2, freq1, freq0)

```

```

def setdata rate(drate):
    mhz= 26
    drate_e = None
    drate_m = None
    for e in range(16):
        m = int((drate * pow(2,28) / (pow(2,e)* (mhz*1000000.0))-256) + .5)      # rounded evenly
        if m < 256:
            drate_e = e
            drate_m = m
            break
    if drate_e is None:
        return False, None
    drate = 1000000.0 * mhz * (256+drate_m) * pow(2,drate_e) / pow(2,28)
    print 'drate_e: %02x drate_m: %02x' %(drate_e,drate_m)

```



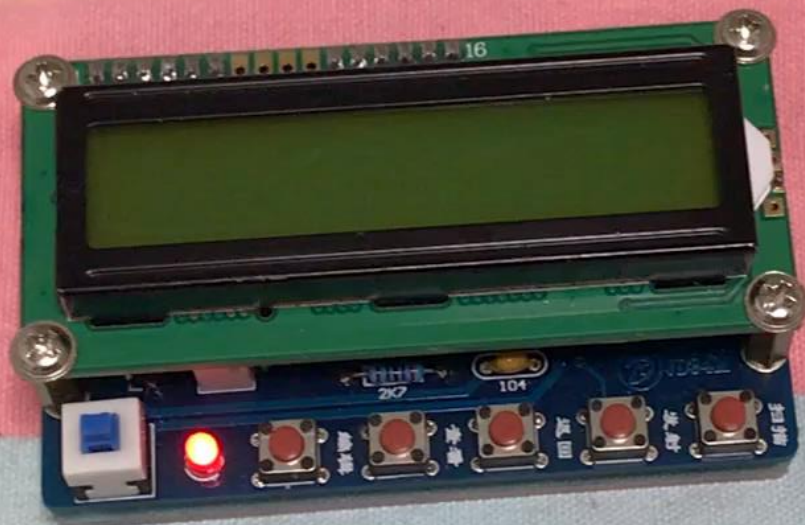
- A890
- 单片机软件解码

正面：发射模块，注意方向



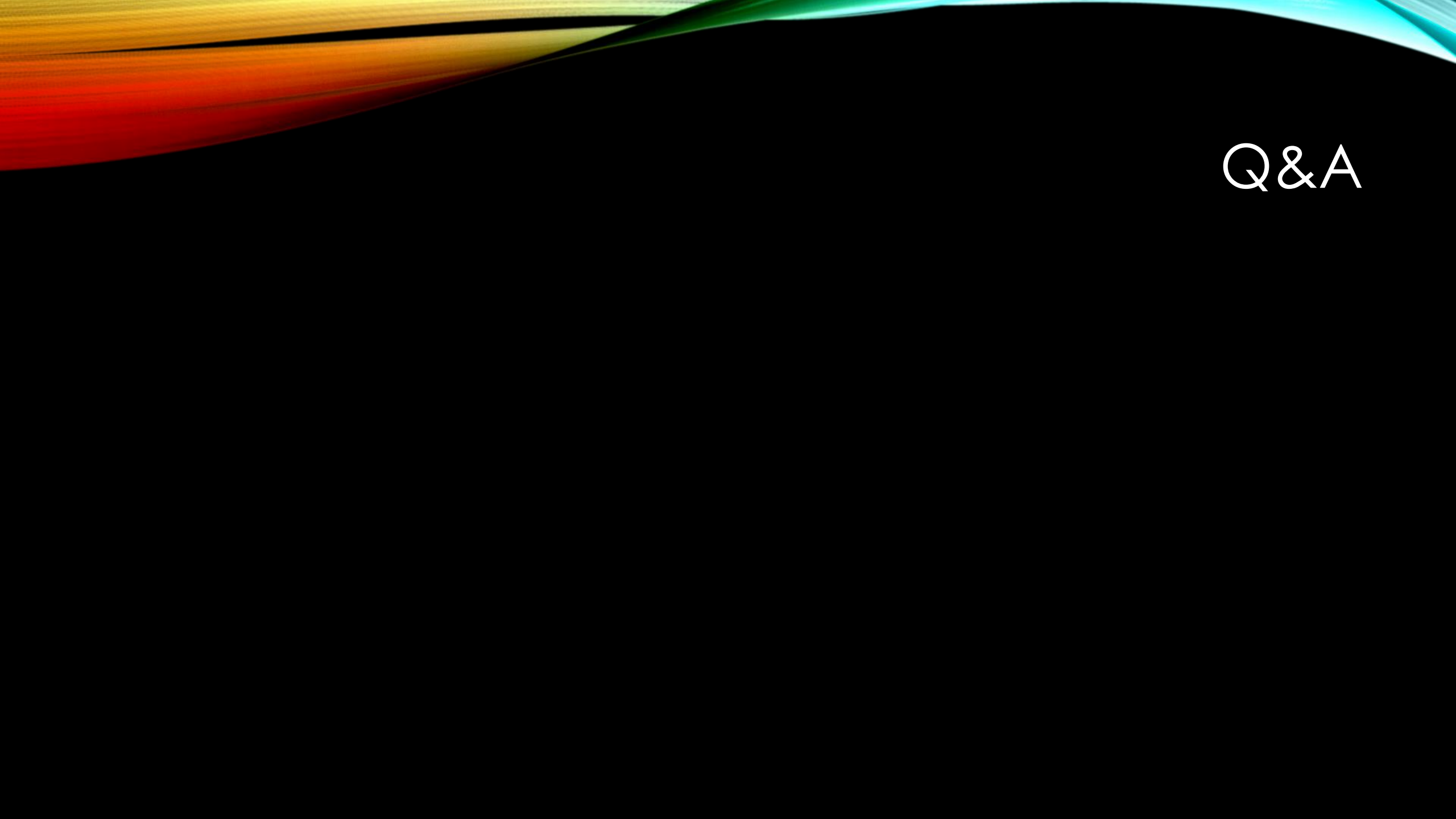
背面：接收模块，注意方向





感谢

- Unicorn Team <https://unicorn.360.com/blog/>
- 雪碧0xroot <https://cn0xroot.com/>
- k1two2 <https://www.cnblogs.com/k1two2/>



Q&A

The background features a solid black field. At the top, there is a decorative, wavy, translucent shape that transitions through a color gradient from yellow and orange on the left to green and blue on the right.

THX