# Et tu, CPCR-505 ?

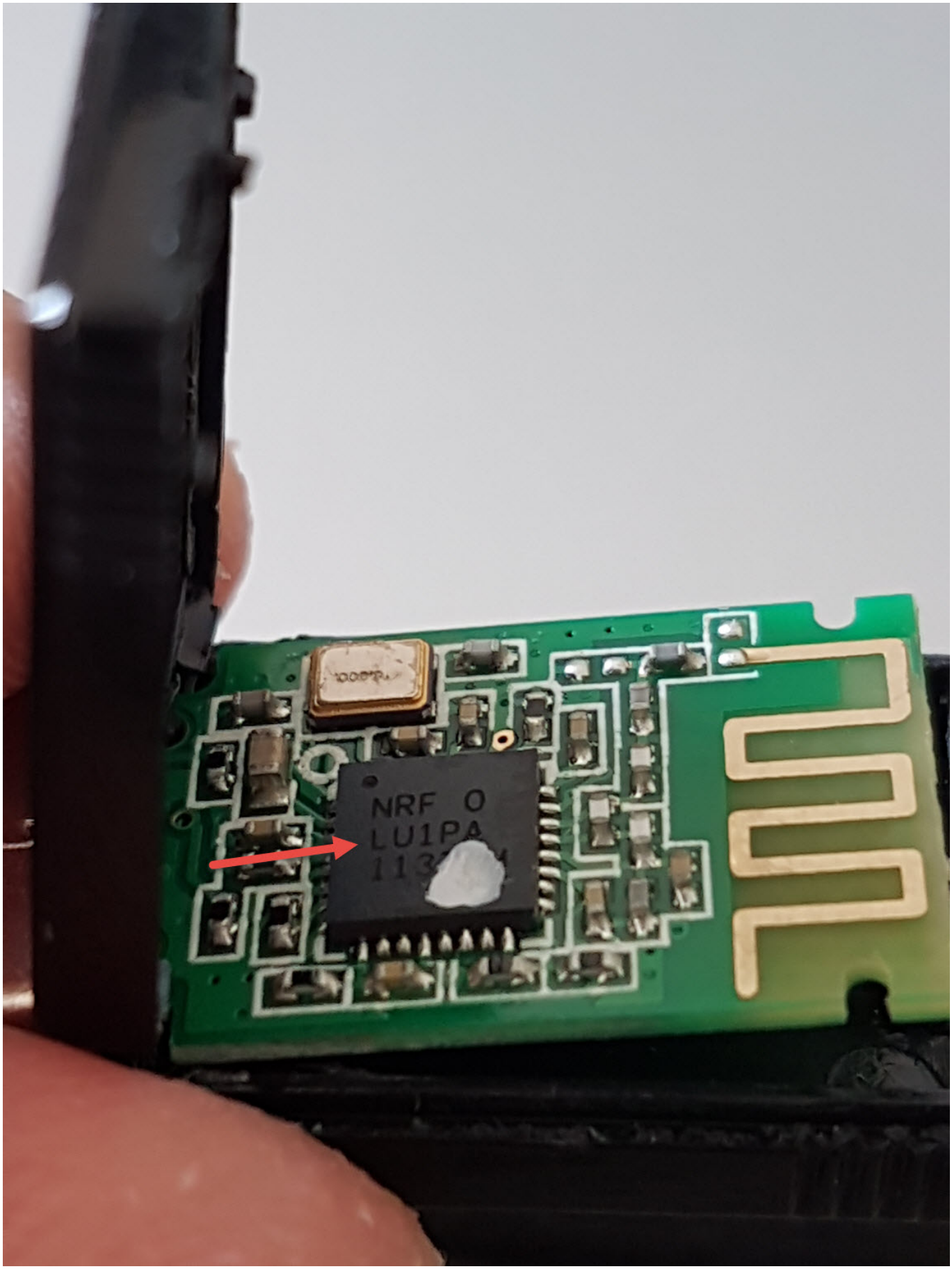
written by Mert SARICA | 1 March 2018
If you are looking for an English version of this article, please visit here.

Yeni yeni sunum davetleri almaya başladığım 2012 yılının başında, ilk iş olarak kendime kablosuz bir sunum kumandası almaya karar vermiştim. Fiyat & performans açısından beni üzmeyecek bir tane ararken, Codegen CPCR-505 sunum kumandası ile karşılaşıp hemen satın almıştım. Emektar sunum kumandam ile geçmiş yıllarda çok sayıda keyifli sunumlara imza atmış olsam da, kendisinin Brütüs gibi beni arkadan hançerleyebileceğini öğrenmem meğer yıllar sonra ortaya çıkacak ve bu yazıya konu olacakmış!

Kablosuz sunum kumandası ile sunum yaparken veya sunum yapan birini izlerken aklımı zaman zaman şu soru kurcalardı; “Sunumu izleyenlerden birisi, sunum kumandasını uzaktan hackleyerek sunumu sabote edebilir mi ?” “Mert sen biraz fazla Mr Robot dizisi izlemişsin galiba” diyenleriniz olduğunu duyar gibi olsam da, geçtiğimiz günlerde bunu öğrenmek için emekter sunum kumandamı

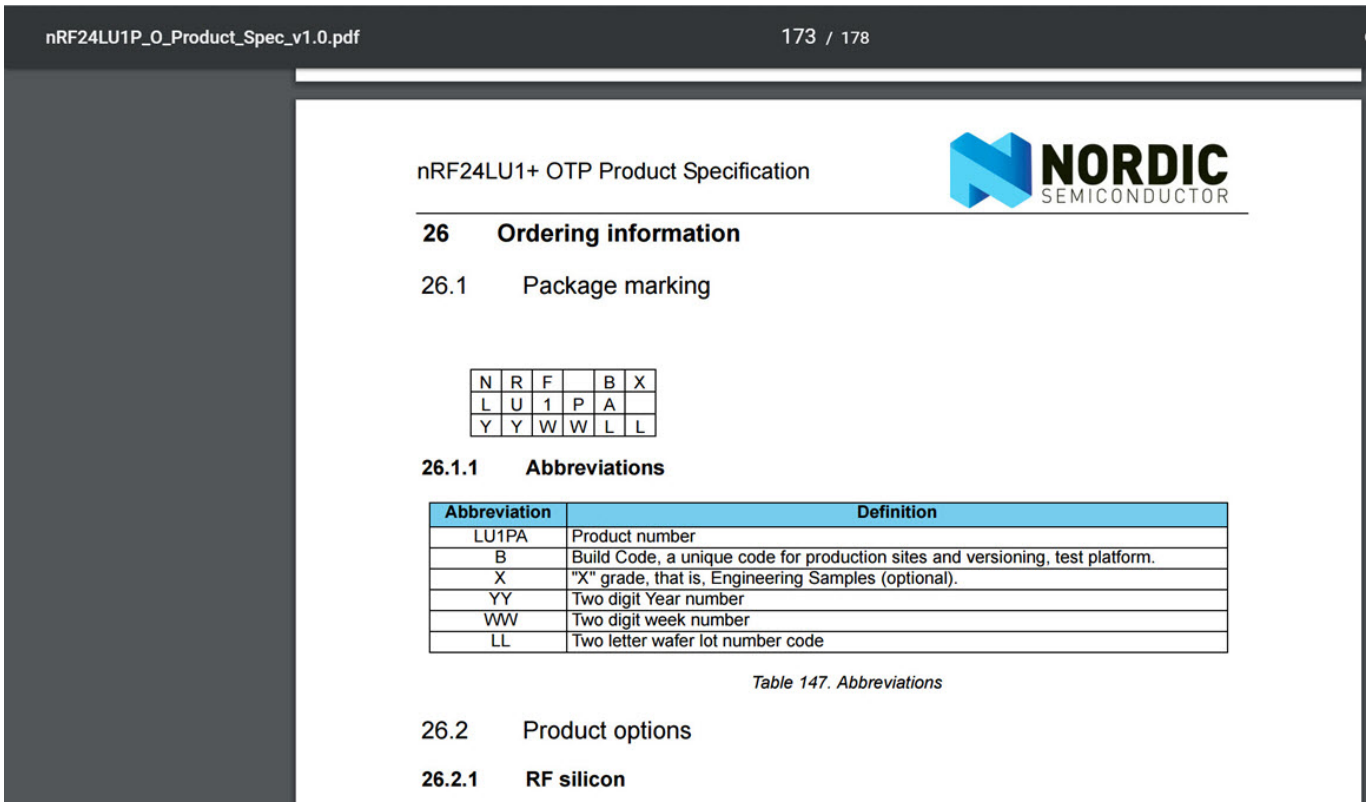
yakından incelemeye karar verdim.

Sunum kumandasının üreticisi olan Codegen firmasının web sitesini incelediğimde, bu kumandanın 2.4 GHz frekans bandında çalıştığı net olarak ifade edilmişti. Kablosuz klavye ve farelerin de bu frekansta çalıştığını Casus Fare başlıklı blog yazım için yapmış olduğum araştırmalardan az çok bildiğimden dolayı, detaylı bilgi toplama adına USB alıcının içini açmakla işe koyuldum.




Alıcının içini açtığımda karşıma nRF24LU1PA 2.4 GHz alıcı-verici çip çıktı. Çipin üreticisi olan Nordic Semiconductor’un web sitesinde bu çipe ait olan belgeye baktığımda, 125 RF kanal desteği, AES şifreleme desteği ve frekans

atlamasına sahip olması ilk dikkatimi çeken noktalar oldu.







## 26 Ordering information

### 26.1 Package marking

| | | | | | |
|---|---|---|---|---|---|
| N | R | F | | B | X |
| L | U | 1 | P | A | |
| Y | Y | W | W | L | L |

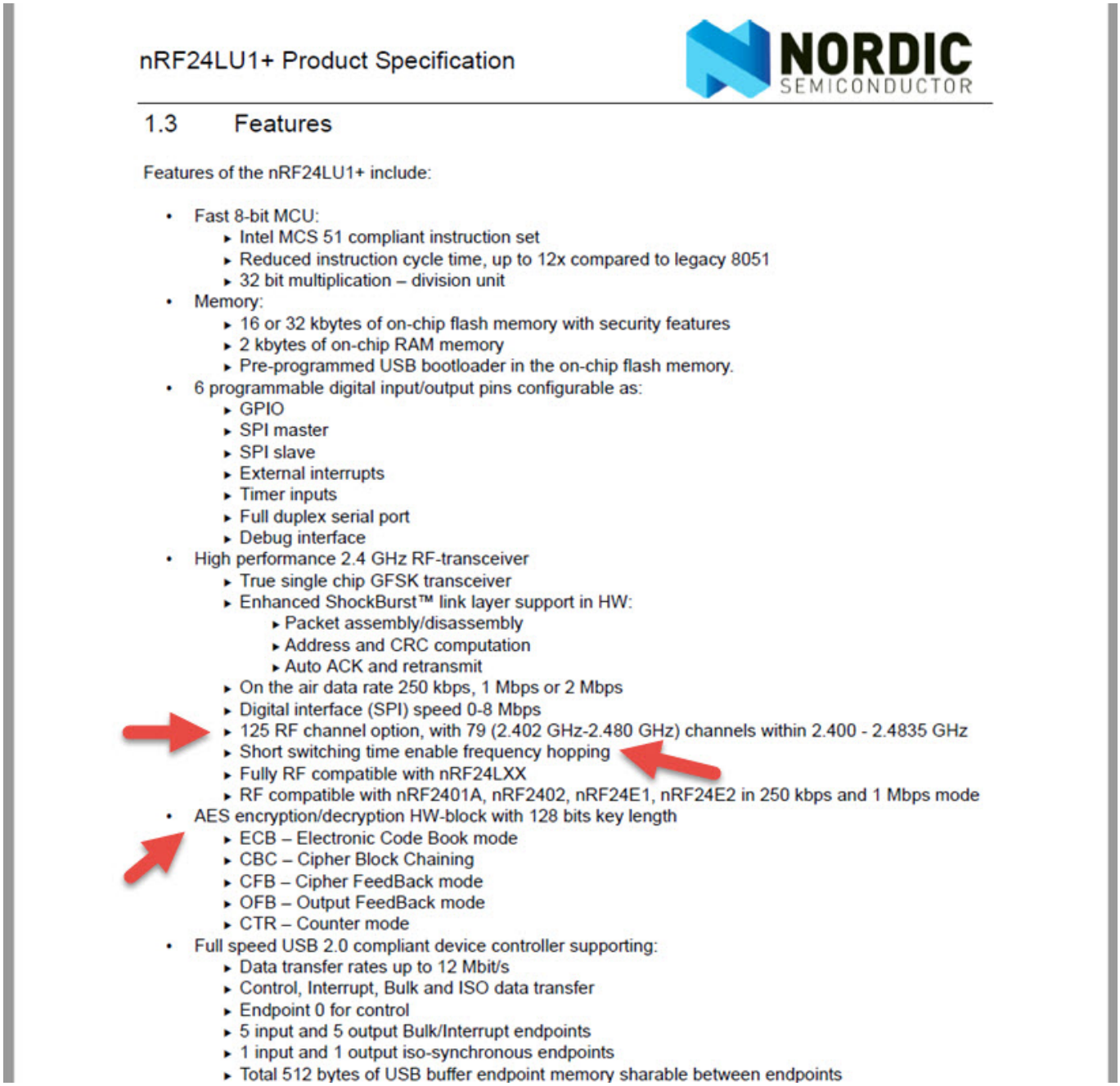
#### 26.1.1 Abbreviations

| Abbreviation | Definition |
|---|---|
| LU1PA | Product number |
| B | Build Code, a unique code for production sites and versioning, test platform. |
| X | "X" grade, that is, Engineering Samples (optional). |
| YY | Two digit Year number |
| WW | Two digit week number |
| LL | Two letter wafer lot number code |

*Table 147. Abbreviations*

### 26.2 Product options

#### 26.2.1 RF silicon



nRF24LU1+ Product Specification

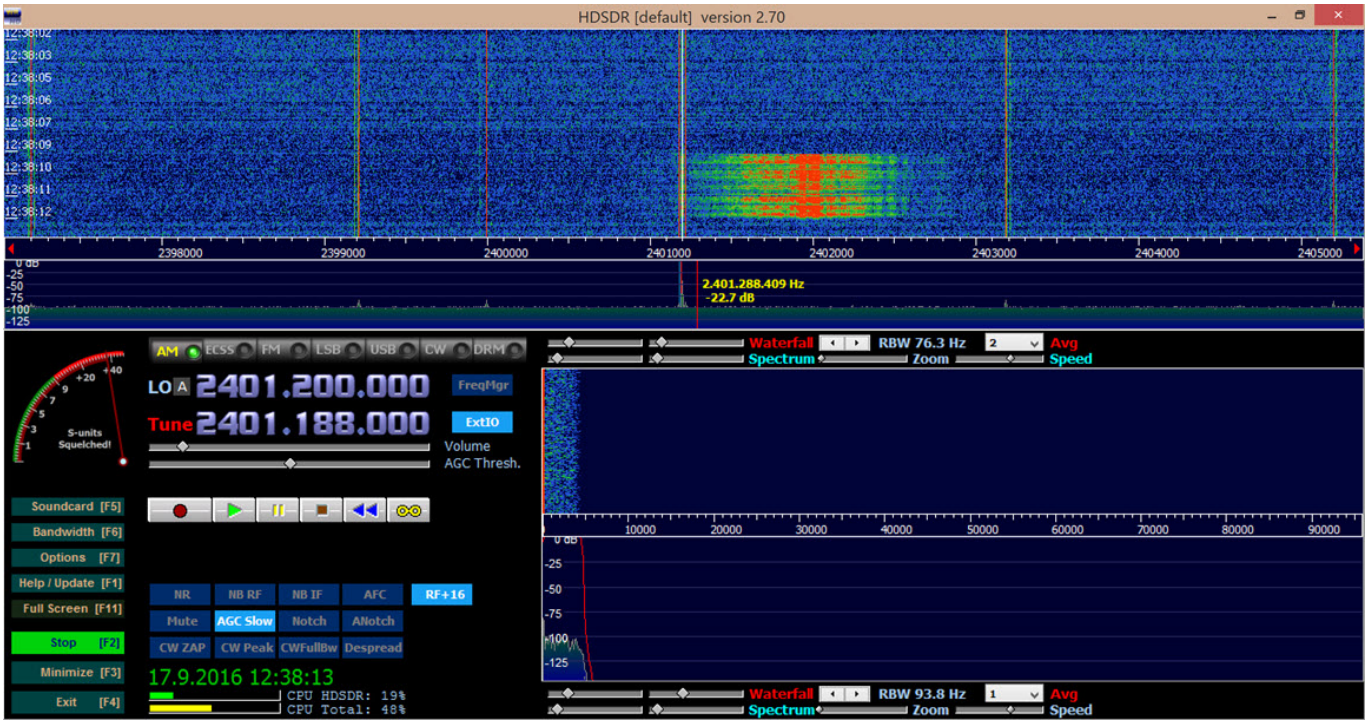
NORDIC SEMICONDUCTOR

## 1.3 Features

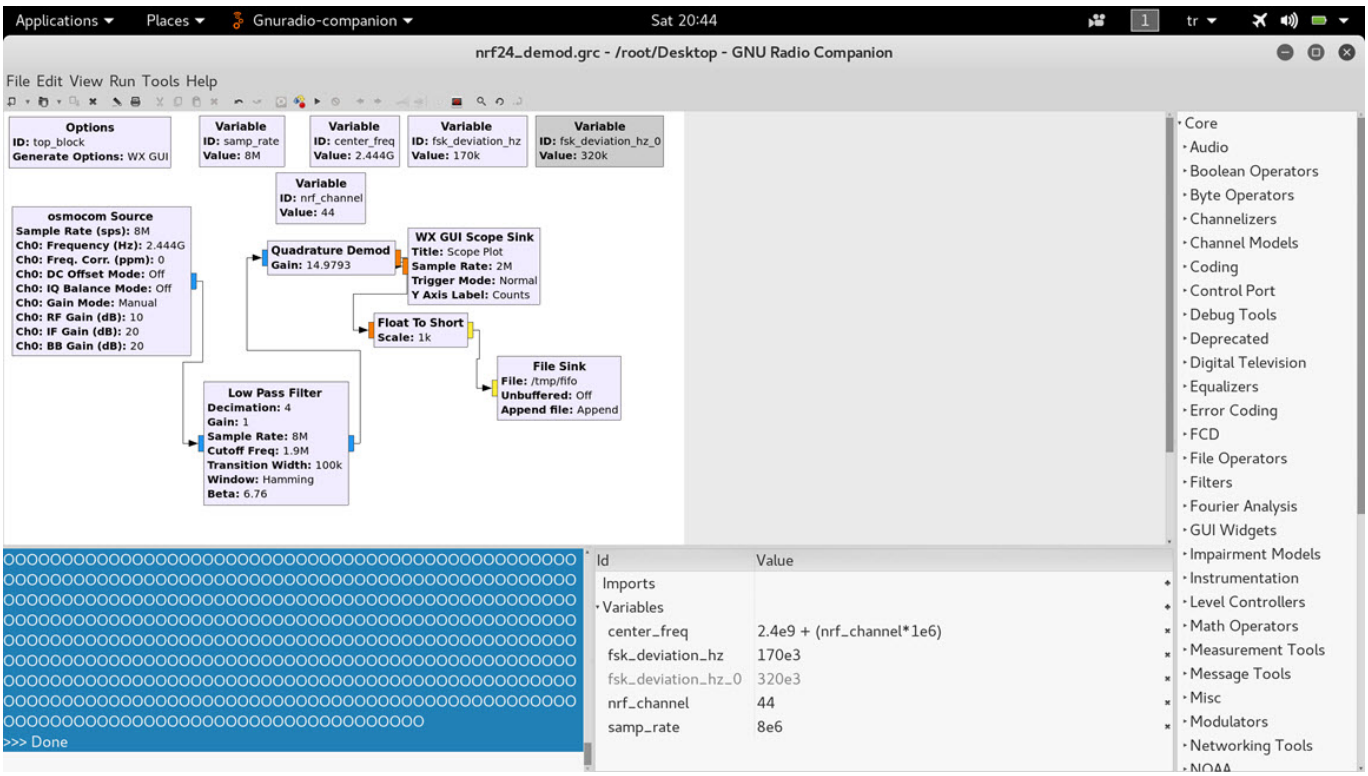
Features of the nRF24LU1+ include:

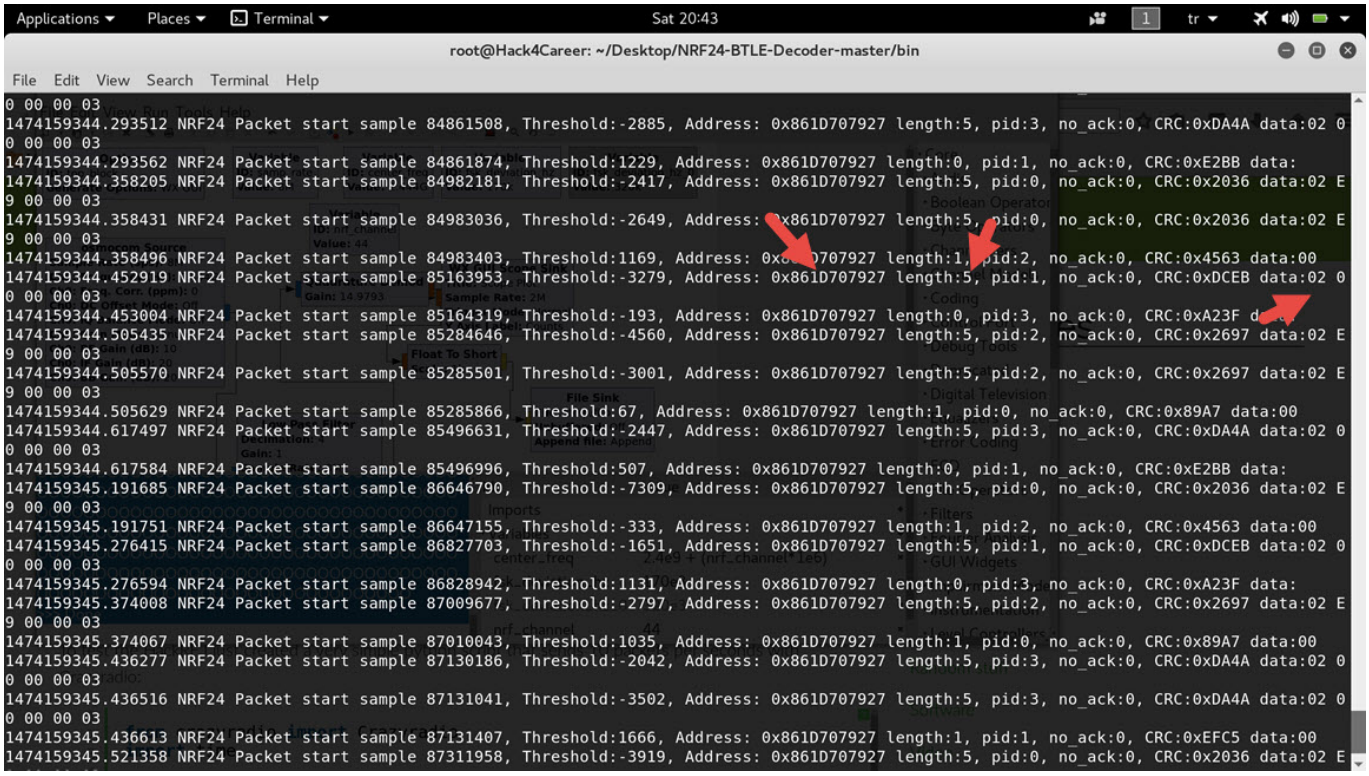
- Fast 8-bit MCU:
  - Intel MCS 51 compliant instruction set
  - Reduced instruction cycle time, up to 12x compared to legacy 8051
  - 32 bit multiplication – division unit
- Memory:
  - 16 or 32 kbytes of on-chip flash memory with security features
  - 2 kbytes of on-chip RAM memory
  - Pre-programmed USB bootloader in the on-chip flash memory.
- 6 programmable digital input/output pins configurable as:
  - GPIO
  - SPI master
  - SPI slave
  - External interrupts
  - Timer inputs
  - Full duplex serial port
  - Debug interface
- High performance 2.4 GHz RF-transceiver
  - True single chip GFSK transceiver
  - Enhanced ShockBurst™ link layer support in HW:
    - Packet assembly/disassembly
    - Address and CRC computation
    - Auto ACK and retransmit
  - On the air data rate 250 kbps, 1 Mbps or 2 Mbps
  - Digital interface (SPI) speed 0-8 Mbps
  - 125 RF channel option, with 79 (2.402 GHz-2.480 GHz) channels within 2.400 - 2.4835 GHz
  - Short switching time enable frequency hopping
  - Fully RF compatible with nRF24LXX
  - RF compatible with nRF2401A, nRF2402, nRF24E1, nRF24E2 in 250 kbps and 1 Mbps mode
- AES encryption/decryption HW-block with 128 bits key length
  - ECB – Electronic Code Book mode
  - CBC – Cipher Block Chaining
  - CFB – Cipher FeedBack mode
  - OFB – Output FeedBack mode
  - CTR – Counter mode
- Full speed USB 2.0 compliant device controller supporting:
  - Data transfer rates up to 12 Mbit/s
  - Control, Interrupt, Bulk and ISO data transfer
  - Endpoint 0 for control
  - 5 input and 5 output Bulk/Interrupt endpoints
  - 1 input and 1 output iso-synchronous endpoints
  - Total 512 bytes of USB buffer endpoint memory sharable between endpoints

2015 yılında Black Hat Güvenlik Konferansı'na gitmişken satın aldığım ve RF Dünyası ve Güvenlik Dünyası blog yazımda da kullanmış olduğum HackRF One cihazı sayesinde sunum kumandası ile bilgisayarıma taktığım USB alıcısının arasındaki haberleşmeyi HDSDR programı yardımı ile izlemeye başladım.



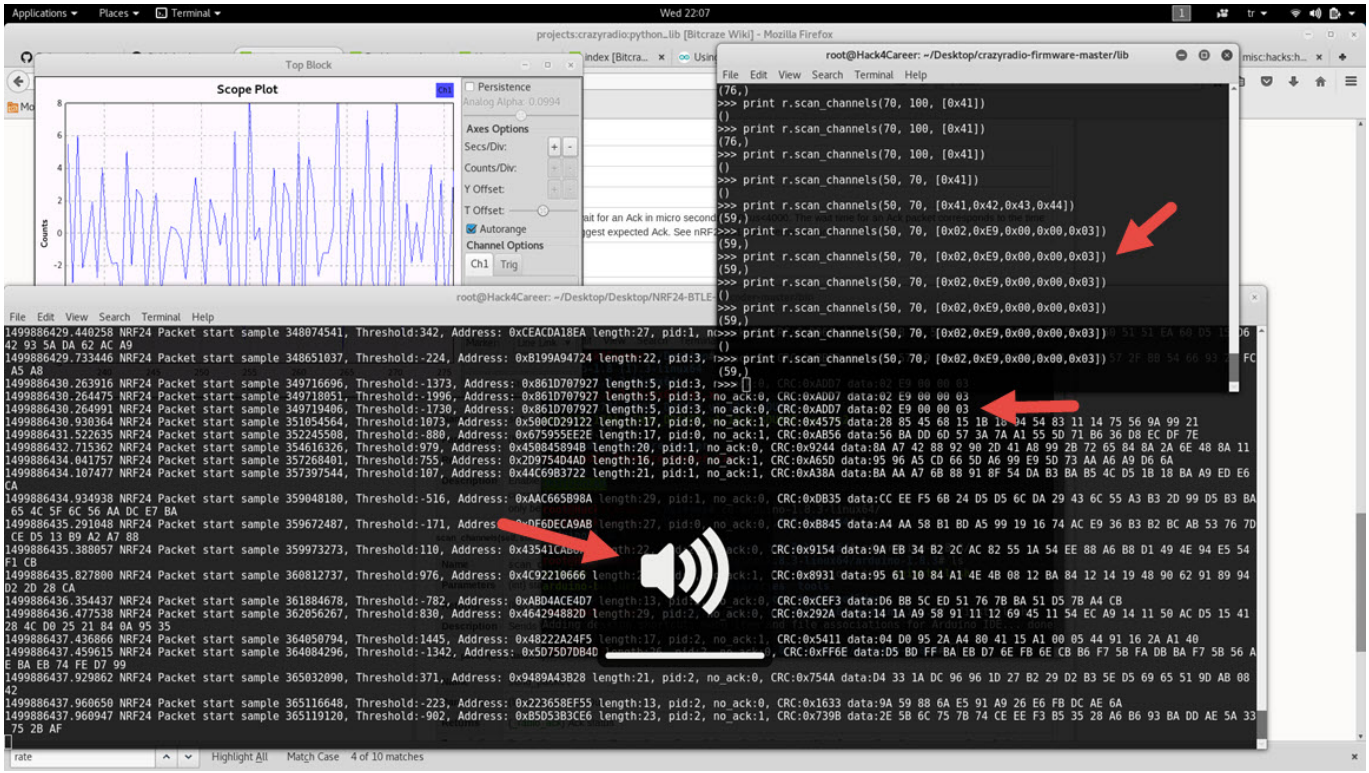

Sunum kumandası frekans atlaması yaptığı için veri paketlerini nasıl yakalayıp, çözeceğim üzerine hindi gibi düşünürken Bitcraze'in Wiki sayfasında tam da aradığım konuyu işleyen bir yazı ile karşılaştım. Yazıda belirtilenleri harfi harfine uygulayarak sunum kumandası ile alıcı arasındaki fare hareketlerinden, sunum kumandasındaki çeşitli butonlara (ses yükseltme, azaltma vs.) basılınca oluşan çeşitli veri paketlerini (adres, boyut, veri) kolayca elde edebildim.






Sıra veri paketlerini sunum kumandasının alıcısına göndermek için hangi aygıtla, cihazla bu işi gerçekleştireceğime karar vermeye gelmişti. Hali hazırda elimde bulunan Arduino Uno R3 ve NRF24L01+ 2.4GHz alıcı verici modül ile bu işi gerçekleştirebilirdim. Bill Gates bir röportajında “Her zaman en tembel insanları işe alırım çünkü tembeller çok karışık işleri bile en kısa yoldan yaparlar.” derken beni kastetmiş olsa gerek ki Arduino, modül ve kablolarla şimdi kim uğraşacak diye hayıflanırken aklıma Casus Fare başlıklı blog yazımda kullandığım CrazyRadio PA USB aygıtı geldi. Python kütüphanesi sayesinde işletim sistemi farketmeksizin 5 satırlık bir kod ile istediğim alıcıya, dilediğim veri paketini göndermem oldukça kolay olacaktı. scan_channels() fonksiyonu sayesinde aşağıdaki kod ile 50-70 arasındaki kanallara (frekans atlama sebebiyle 50-70 arası kanallara göndererek veri paketinin alıcıya ulaşma şansını arttırdım.) ses yükseltme komutu içeren veri paketini gönderdim ve başarıya ulaştım. ;)

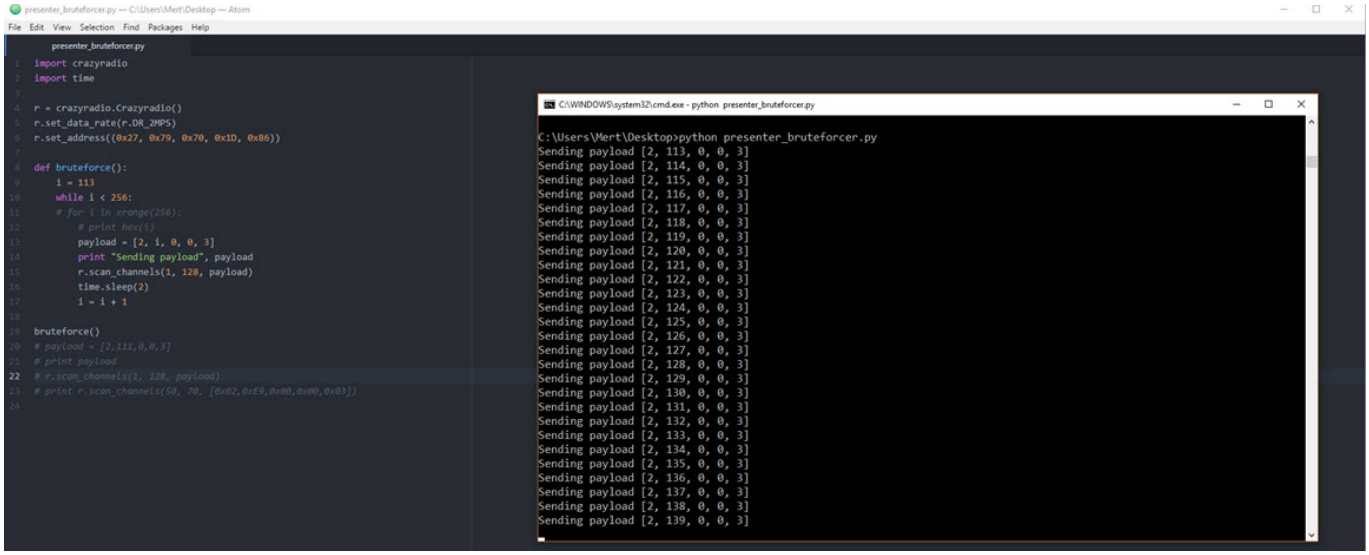
```
import crazyradio
r = crazyradio.Crazyradio()
r.set_data_rate(r.DR_2MPS)
# Alıcı/Verici adresi
r.set_address((0x27, 0x79, 0x70, 0x1D, 0x86))
# Ses yükseltme komutu
print r.scan_channels(50, 70, [0x02,0xE9,0x00,0x00,0x03])
```



Tabii ses arttırarak veya azaltarak bir sunumu sabote etmek pek de mümkün olamayacağı için sunum kumandasının normalde göndermediği ancak sunumun kumandasının alıcısının desteklediği gizli komutları, deneme yanılma (brute-force) yöntemi ile tespit etmeye karar verdim. Bunun için de aşağıdaki Python kodunu yazdım.

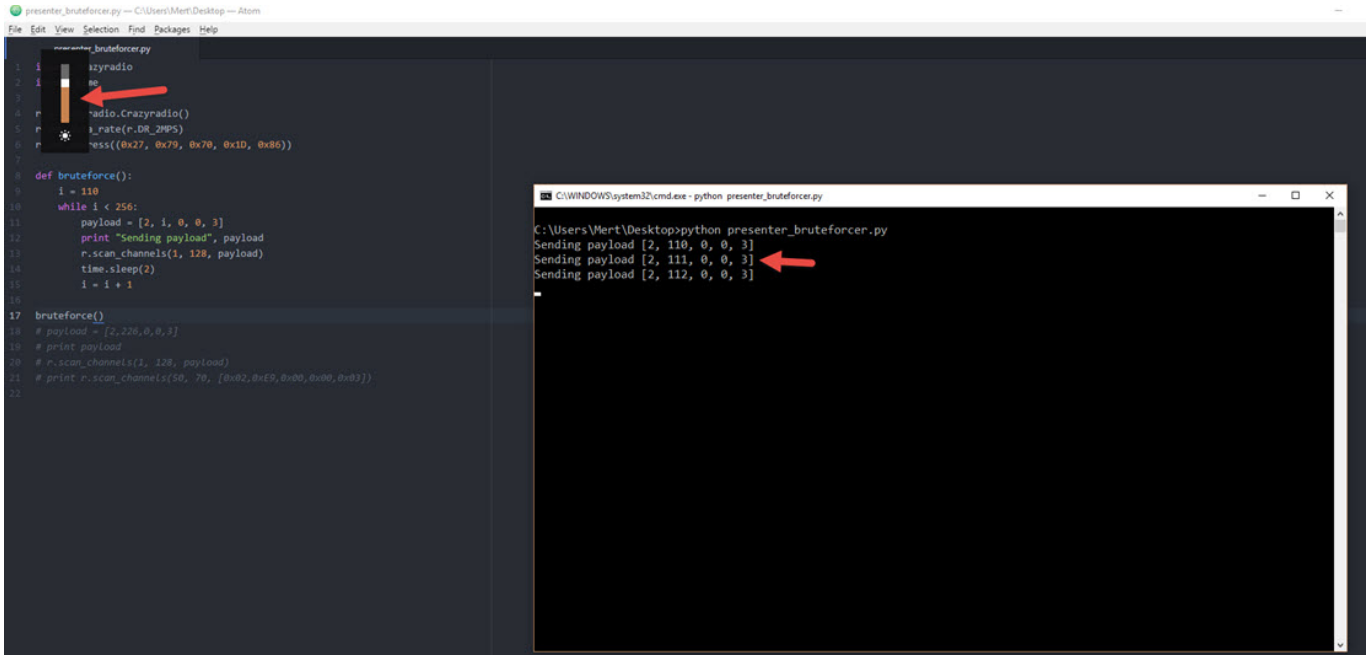
```
import crazyradio
import time

r = crazyradio.Crazyradio()
r.set_data_rate(r.DR_2MPS)
r.set_address((0x27, 0x79, 0x70, 0x1D, 0x86))

def bruteforce():
    i = 0
    while i < 256:
        # Gi
        payload = [2, i, 0, 0, 3]
        print "Sending payload", payload
        r.scan_channels(1, 128, payload)
        time.sleep(2)
        i = i + 1
```

bruteforce()




Çok geçmeden sunum kumandasında var olmayan ancak sunum kumandasının alıcısının desteklediği ve ekran parlaklığını azaltmaya yarayan aşağıdaki komutu keşfettim. Crazy Radio PA USB aygıtı ile bu komutu alıcıya birden fazla gönderdiğimde ekran parlaklığına okumayı oldukça zorlaştıracak seviyeye kadar indirebildim ki bu gerçekten bir sunumu sabote etmek için yeter ve artardı. :)

[2, 111, 0, 0, 3] # Ekran parlaklığını arttırıyor.
[2, 112, 0, 0, 3] # Ekran parlaklığını azaltıyor.




Sunum kumandasının alıcısına şayet Casus Fare blog yazısında olduğu gibi

klavye tuş basma komutları da gönderebilinirse durumun gerçek anlamda sistem güvenliğine tehlikeye atacak bir boyuta gelebileceğini de göz önünde bulundurarak sunum kumandamı bu çalışma sonunda emekli ederek, şifreli haberleşen bir sunum kumandası aramaya koyuldum.

Bunun gibi 2.4 GHz (ISM bandı) frekans bandında, şifresiz gerçekleşen benzer tüm haberleşmelerde (misal dronelar olabilir) bu tür veya benzeri güvenlik zafiyetlerinin ortaya çıkabileceğini de göz önünde bulundurmanızı hatırlatarak bir sonraki yazıda görüşmek dileğiyle herkese güvenli günler dilerim.