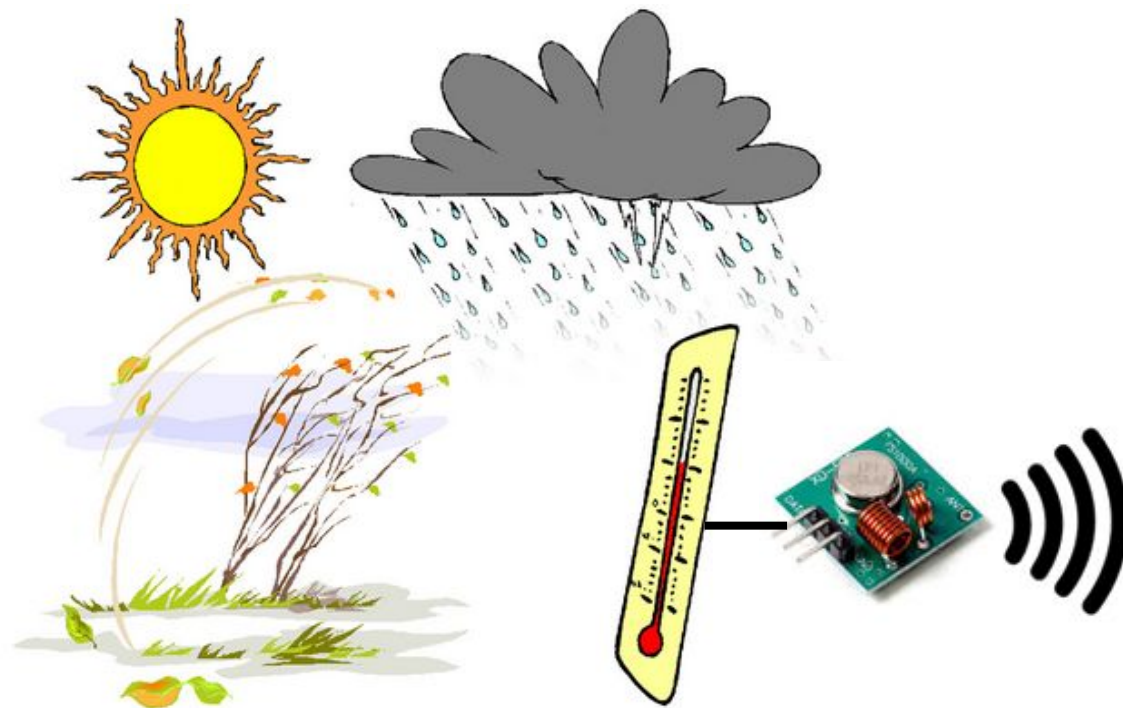
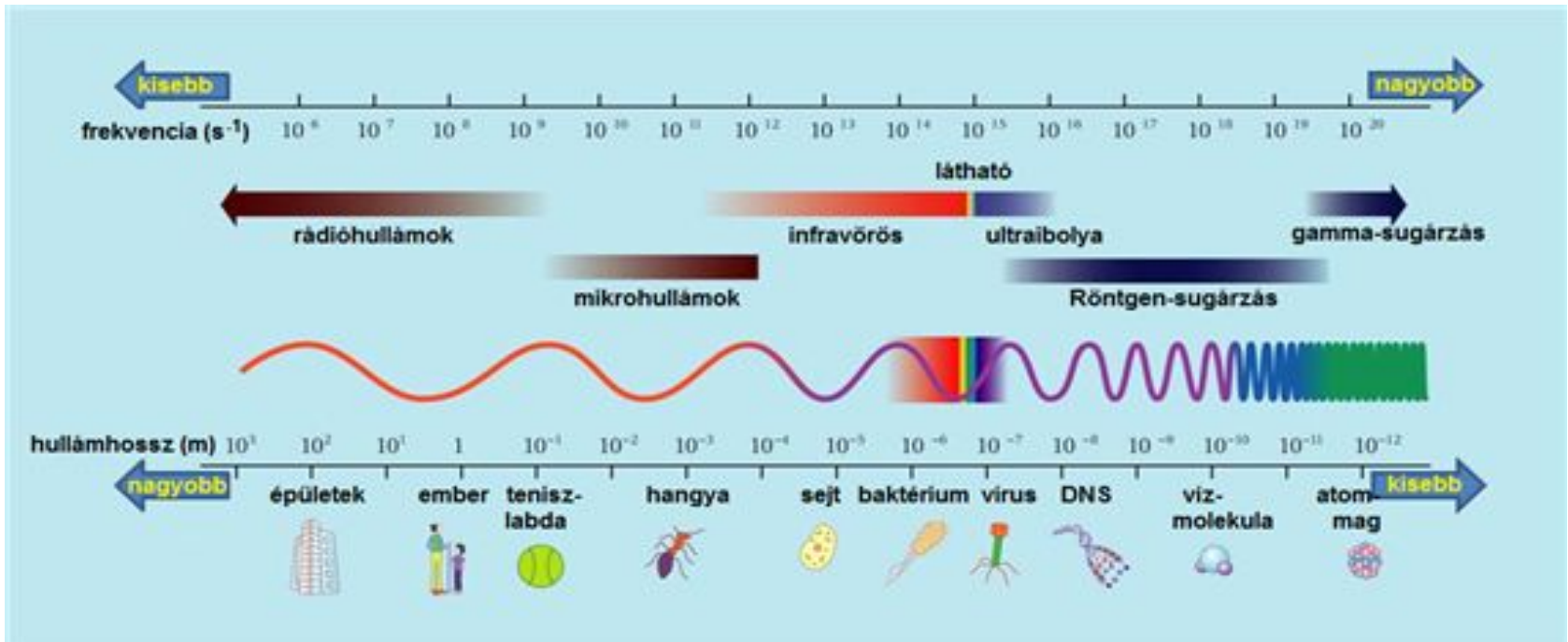


Időjárásállomás külső érzékelőjétől érkező rádiójel feldolgozása



Az elektromágneses sugárzás spektruma

- Az elektronok mozgása elektromágneses hullámokat kelt
- Adattovábbítás főleg a rádióhullámokkal (7 kHz – 5 GHz)

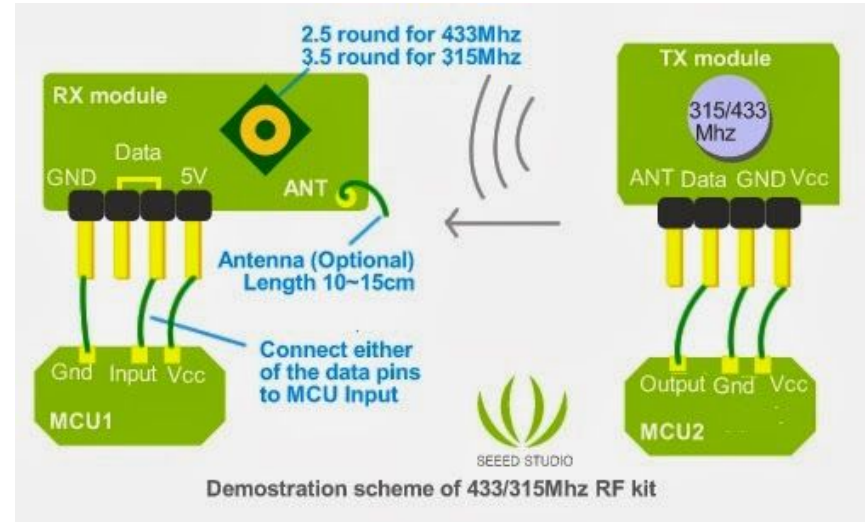


ISM sávok

- A nemzetközi egyezmények külön frekvencia sávokat biztosítanak az ipari, tudományos és orvosi alkalmazások számára (pl. rádiófrekvenciás fűtés, mikrohullámú sütő, diathermás kezelés)
- Az utóbbi években rohamosan nő azoknak az alkalmazásoknak a száma, amelyek az eredeti céltól eltérően (rövidtávú) kommunikációra használja fel ezeket a frekvencia tartományokat. A vezeték nélküli telefonok, bluetooth eszközök, NFC, garázsajtó nyitók, baba-monitorok, vezeték nélküli hálózatok (WiFi) mind **ISM sávot** használnak, mivel ehhez nem kell külön kormányzati engedélyt beszerezni.
- Az **ISM sávok** közepes frekvenciái: 6.78 MHz, 13.56 MHz, 27.12 MHz, 40.68 MHz, **433.92 MHz** (USA 915 MHz), 2.45 GHz, 5.8 GHz, stb.

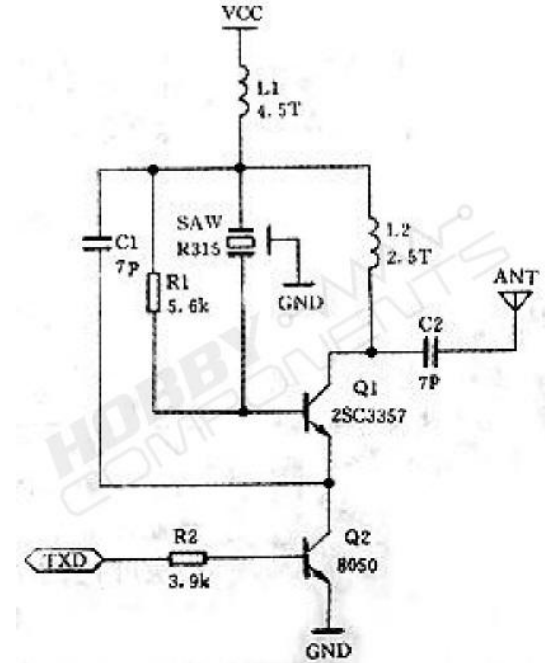
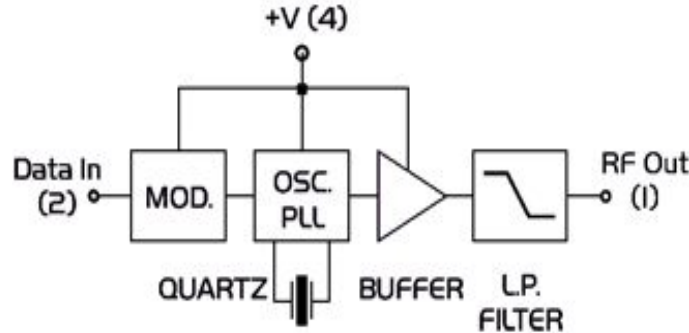
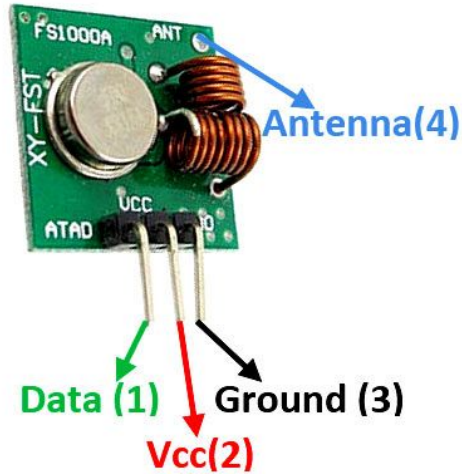
Egyszerű adatátvitel 433 MHz-en

- A hullámhossz kb. 70 cm
- Az optimális antennahossz = $\lambda/4 = 17,3$ cm
- Az FS1000A (máshol MX-FS-03V / MX-05V néven fut) adó-vevő pár olcsó megoldást nyújt



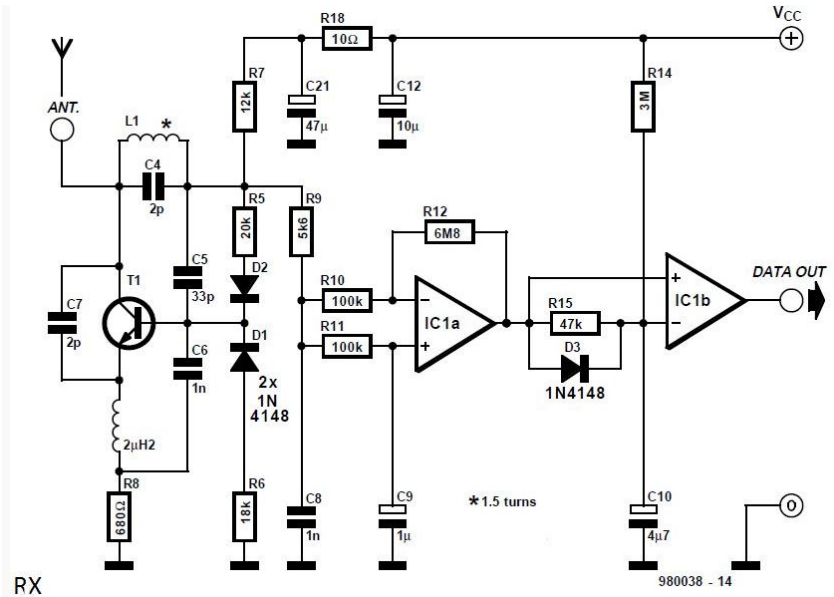
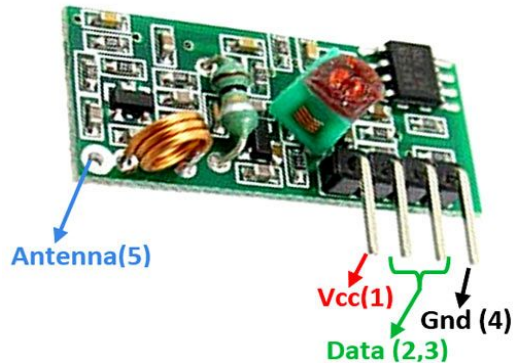
Adó modul

- A digitális adatjellel modulált adó magas szintű bemenetnél sugároz, alacsony szintű bemenetnél nem sugároz (OOK/ASK moduláció)



Vevő modul

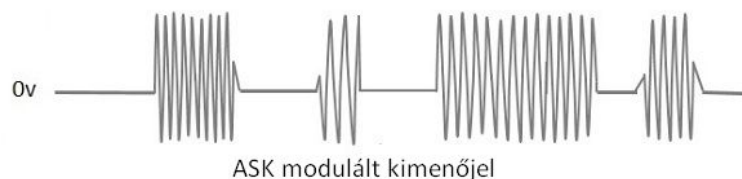
- A vevő modul feladata, hogy az antennával felfogott jelet felerősítse, majd demodulálja
- Az egyszerű (ASK) moduláció miatt a demoduláció csupán egyenirányítást és szűrést jelent



Nem feltétlenül egyezik meg az aktuális kapcsolással!

ASK moduláció és demoduláció

- Az ASK moduláció lényegében a 433 MHz-es vivőfrekvencia „szaggatását” jelenti
- A demodulációnál az egyenirányítás és szűrés után egy komparátor teszi digitálissá a kimenőjelet



Modulált
ASK jel



Aszinkron ASK detektor

Alkalmazási példák

- Távvezérelhető jelfogó (lámpa, konnektor, kapunyitó motor stb távvezérléséhez)
- Több (max 25) távvezérlőre tanítható
- Háromféle módban működhet:
 - Addig kapcsol, amíg nyomjuk
 - Ki/be kapcsolás egy gombbal
 - Ki és bekapcsolás külön gombbal



AC 85V ~ 250V

Alkalmazási példák

- Rádiófrekvenciás átjáró a 433 MHz-es és a WiFi eszközök között
- Okos otthon megoldásokhoz



Alkalmazási példák



- Időjárásjelző állomás kültéri érzékelővel
- A kommunikáció a kültéri és a beltéri egység között 433 MHz-en történik
- Kevesebb energiát igényel, mint a WiFi kapcsolat



Aleco WS 1700 protokoll

A protokoll elérhető a pilight.org címen, Weather Stations pont alatt a teljes protokoll us -ben megadva, a páros pozíción a jel adás ideje, páratlanon a szünet

- Weather Stations
 - Aleco WSD17
 - Aleco WS1700
 - Aleco WX500

```
540 1890 540 3780 540 1890 540 3780 540 3780 540 3780 540 3780 540 3780 540 1890 540 1890 540 3780 540 3780 540 3780 540 1890 540 1890
540 1890 540 1890 540 1890 540 1890 540 1890 540 3780 540 3780 540 1890 540 1890 540 4050 540 1890 540 4050 540 4050 540 1890 540 1890
540 4050 540 1890 540 3780 540 1890 540 3780 540 3780 540 9180
```

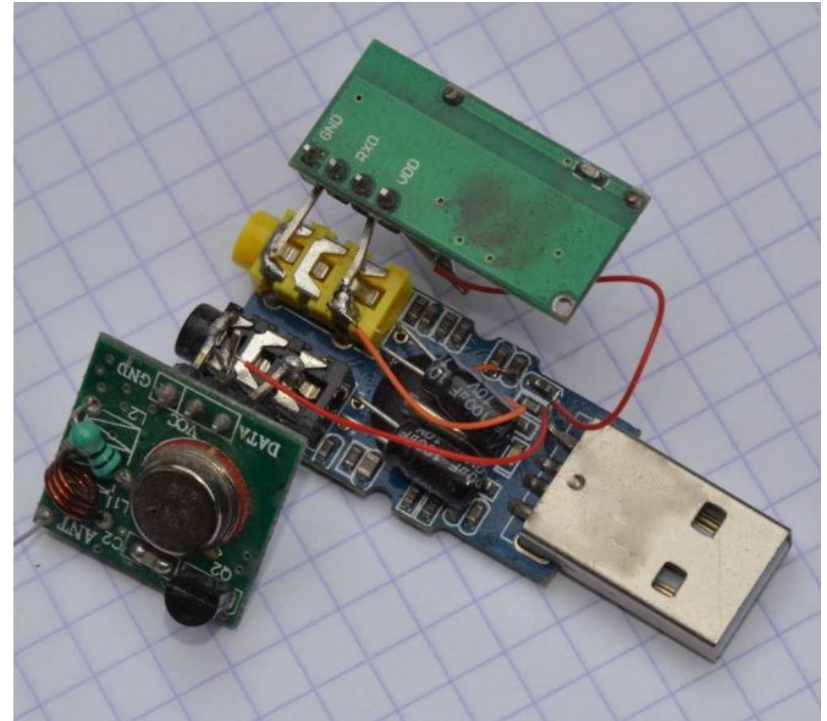
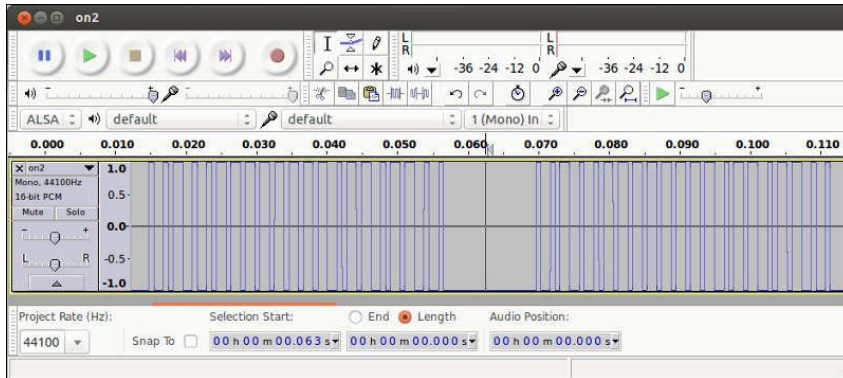
az 540 1890 a 0-át jelenti, az 540 3780 pedig az 1-et, az 540 9180 pedig a szinkron jel

- 0 - 3 header 0101 (digoo esetén 1001)
- 4 - 11 ID, minden elemkivétellel új ID generálódik
- 12 elem állapota
- 13 TX mód
- 14 - 15 csatorna száma
- 16 - 27 hőmérséklet tízzel megszorozva, legfelső bit negatív
- 28 - 35 páratartalom

RF protokoll egyszerű dekódolása USB hangkártya segítségével

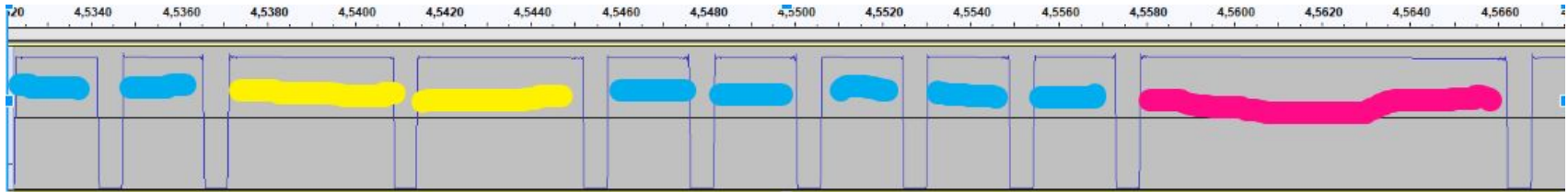
Olcsó \$1.25 -os USB hangkártya és 433.92 MHz vevő segítségével egy egyszerű dekódoló készíthető, a megjelenítéshez az Audacity nevű nyílt forráskódú program használható

https://rurandom.org/justintime/w/Cheapest_ever_433_Mhz_transceiver_for_PCs

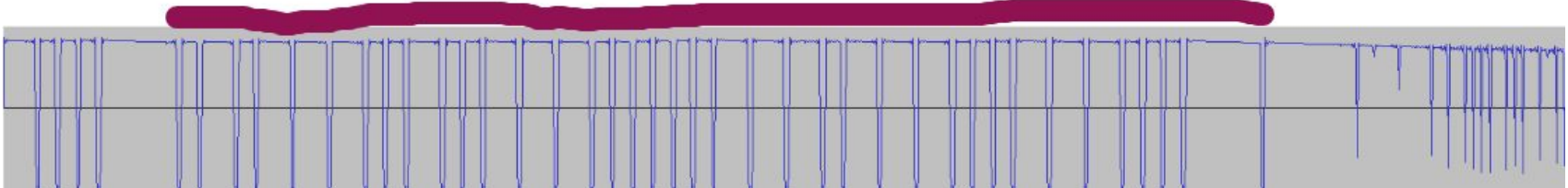


Aleco WS 1700 protokoll

egymás után háromszor küldi el a teljes adatsort minden adatsort egy ~8 ms szinkronjellel fejezi be. Az 1-es ~4 ms -os jelszűnet, a 0-ás ~2 ms-os jelszűnet jelzi.



A teljes adatsor 36 bit



Aleco WS 1700 adat dekódolása micropython segítségével

az ismételt küldést kihasználva a szinkron jelet kezdő jelként használjuk, a státuszt SYNCED-re állítva megtörténik a jelhossz letárolás, majd ha minden adat letárolásra került (elérte a 36 bitet), a státuszt DATA-ra állítva megtörténik a jelkiértékelés és a dekódolás

```
SYNC
SYNC
temp: 22.4C, humy: 43%
100111100110000000001110000000101011
```

```
1 import pyb , time, array
2
3 bsize=37 ; bi=0 ; buf=array.array("I") # segéd buffer a debughoz
4 for i in range(bsize):
5     buf.append(0)
6 # az időzítések tartományként vannak meghatározva
7 LP=range(200,800) ; APP=range(7000,10000) ; ASP=range(1500,2500) ; ALP=range(3500,4500) ; protwidth=36
8 ert=None ; temp=humy=ch=0 ; elozi=time.time.time_ticks_us() ; up = False
9 # a module adat lába ide csatlakozik
10 rfpin = pyb.Pin('Y12', pyb.Pin.IN , pyb.Pin.PULL_NONE)
11 def f():
12     global elozi,ert, rfpin, up
13     up=rfpin()==1
14     ert = time.time_ticks_us() - elozi
15     elozi=time.time_ticks_us()
16
17 T_SER = T_WRK = time.time_ticks_ms()
18 # az adat láb megszakításának beállítása az előbb megadott funkcióra, fel és lefutó ág is kell
19 ext = pyb.ExtInt(rfpin, pyb.ExtInt.IRQ_RISING_FALLING, pyb.Pin.PULL_NONE, lambda t: f() )
20 STEP_IDLE = 0 ; STEP_SYNCED = 1 ; STEP_DATA = 2 ; STEP_PRINT = 3 ; step = STEP_IDLE
21 while True:
22     if ert!=None:
23         if not up:
24             if ert not in LP:
25                 if step==STEP_SYNCED:
26                     print("#NOSYNC",bi,ert)
27                     step=STEP_IDLE
28             elif step==STEP_IDLE and ert in APP:
29                 bi=0 ; step=STEP_SYNCED ; print("SYNC")
30             elif step==STEP_SYNCED:
31                 buf[bi]=ert ; bi+=1
32                 if bi==protwidth: step=STEP_DATA
33             if step==STEP_DATA:
34                 s="" ; err=False
35                 for i in range(protwidth):
36                     if buf[i] in ASP: s+="0"
37                     elif buf[i] in ALP: s+="1"
38                 else:
39                     s+="_"
40                     err=True
41             if not err:
42                 temp = int( s[16:28] , 2 )
43                 if (temp > 3840): temp -= 4096
44                 temp/=10
45                 humy = int( s[28:36] , 2 )
46             else:
47                 print("ERR:" + s )
48             step=STEP_IDLE
49         ert=None
```

Aleco WS 1700 protokollt használó külső érzékelő



DANIU 433MHz Wireless Weather Station Digital
Thermometer Humidity Sensor

https://www.banggood.com/DANIU-433MHz-Wireless-Weather-Station-Digital-Thermometer-Humidity-Sensor-p-965559.html?rmmnds=search&cur_warehouse=CN

(drága, több mint \$5, majdnem \$10)

Digoo DG-R8S 433MHz Wireless Digital
Hygrometer Thermometer Weather Station
Sensor for DG-TH8888pro

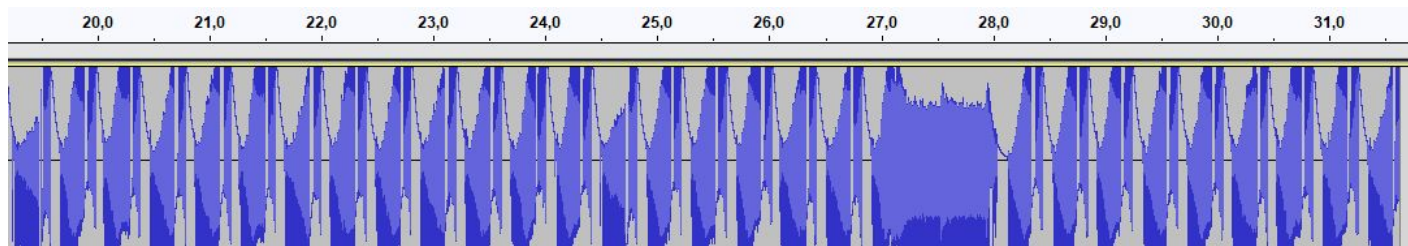
https://www.banggood.com/Digoo-DG-R8S-433MHz-Wireless-Digital-Hygrometer-Thermometer-Weather-Station-Remote-Sensor-p-1139603.html?rmmnds=myorder&ID=512288&cur_warehouse=CN

(általában \$5 alatti az ára)

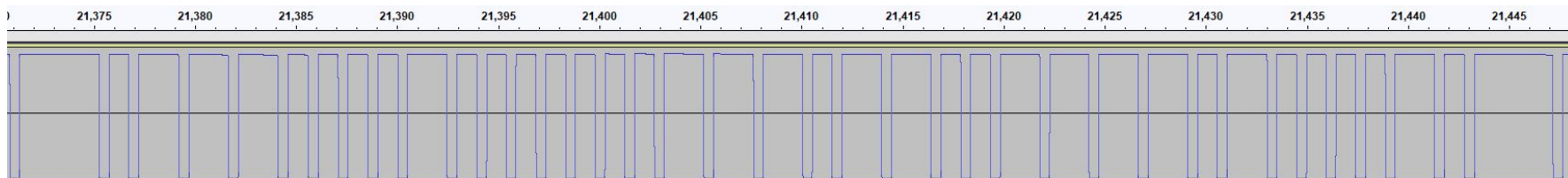


Digoo R8B protokoll visszafejtése

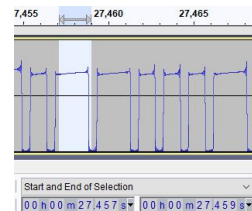
A hangkártyával elkapott jelekből egyértelműen látszik az adatküldés



Nagyítva az elküldött adatokra jól látszik, hogy két szinkron jel között ismét 36 bit adat található.



Kijelöléssel megállapítható a jelszűnethossz, ami 1000us 0 esetén 2000us 1 esetén és 4000us a szinkronjel



Digoo R8B protokoll feldolgozása micropython segítségével

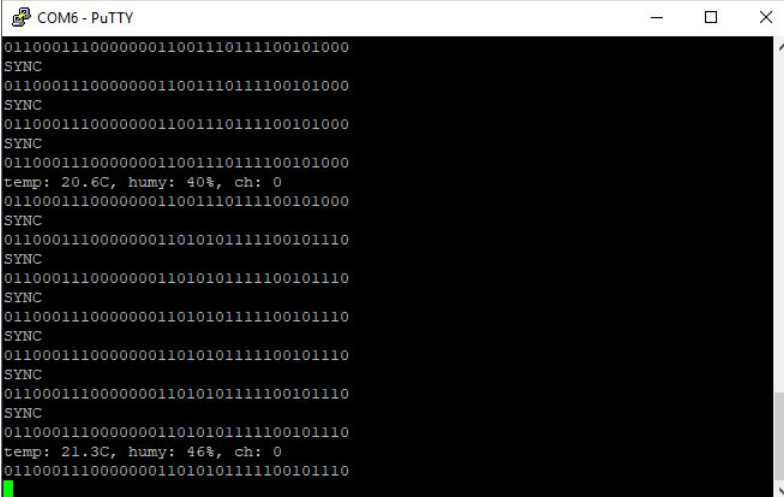
A python kódot módosítani kell az új jelhossz értékeivel

```
# az időzítések tartományként vannak meghatározva  
LP=range(200,800) ; APP=range(3000,5000) ; ASP=range(500,1500) ; ALP=range(1500,2500) ; protwidth=36
```

A módosított programot futtatva megkapjuk a 36 bitnyi adatsort bináris string-ként

Érdeemes a csatornát átkapcsolni, elemkivétellel és anélkül több adatot begyűjteni

Mivel a hőmérsékletet nem ismerjük (ezen a hardveren nincs kijelző) hűteni vagy épp melegíteni a szenzort



```
COM6 - PuTTY  
011000111000000011001110111100101000  
SYNC  
011000111000000011001110111100101000  
SYNC  
011000111000000011001110111100101000  
SYNC  
011000111000000011001110111100101000  
temp: 20.6C, humy: 40%, ch: 0  
011000111000000011001110111100101000  
SYNC  
011000111000000011010101111100101110  
SYNC  
011000111000000011010101111100101110  
SYNC  
011000111000000011010101111100101110  
SYNC  
011000111000000011010101111100101110  
SYNC  
011000111000000011010101111100101110  
temp: 21.3C, humy: 46%, ch: 0  
011000111000000011010101111100101110
```

Digoo R8B protokoll kiértékelése

Javascript-ben írt programmal a kapott értékeket kijelölés után az adott pozíción visszafejtve megkapjuk az eszköz által mért adatokat

A protokoll R8B esetén

- 10 - 11 csatorna
- 12 - 23
hőmérséklet
- 28 - 35
páratartalom

kezd: 12, vege: 24

```
1 001101111000000100011101111100101010
2 001101111000000100010011111100101000
3 001101111000000100001101111100101001
4 001101111000000100001100111100101100
5 010001101001000100010001111100101110
6 0100010010010001000100011111100101111
7 01000111010000100010100111100101111
8 010001011010000100010010111100110001
9 010001011010000100000100111100101011
10 010001011010000100000010111100101011
11 010001011010000100000001111100101011
12 010001011010000011111100111100101100
13 010001011010000011111101111100101011
14 010001011010000011111110111100101011
15
```

```
1 000100011101 285
2 000100010011 275
3 000100001101 269
4 000100001100 268
5 000100010001 273
6 000100010011 275
7 000100010100 276
8 000100010010 274
9 000100000100 260
10 000100000010 258
11 000100000001 257
12 000011111100 252
13 000011111101 253
14 000011111110 254
15
```

Futtat

Ment

Digoo R8B külső érzékelő

Digoo DG-R8B 433MHz Wireless Digital Hygrometer Thermometer
Weather Station Sensor for DG-TH3330

Előnye

- olcsó, \$5 alatt megkapható
- AA elem és a kijelző hiánya miatt sokáig bírja
- kevésbé sérülékeny

Hátránya

- nincs kijelző, így nem olvashatók le az értékek

