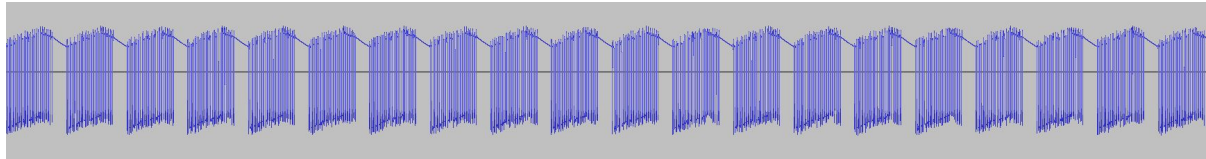
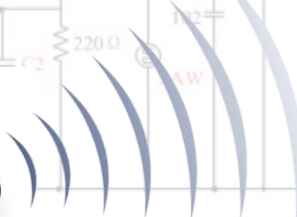
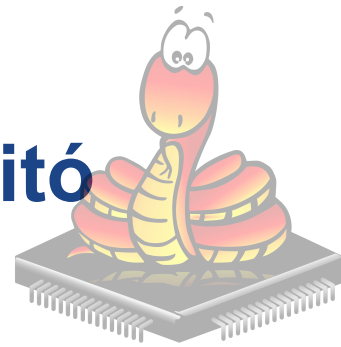
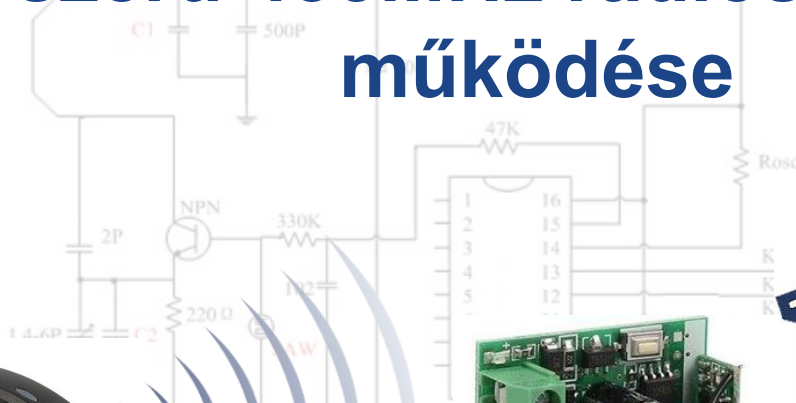
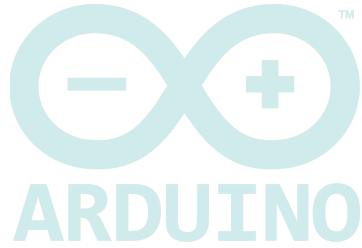


Egyszerű 433MHz rádiós kapunyitó működése



Rádiófrekvenciás kapunyitó

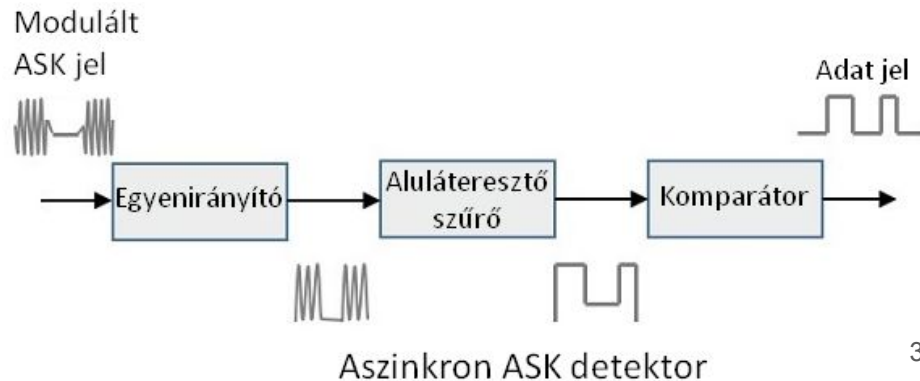
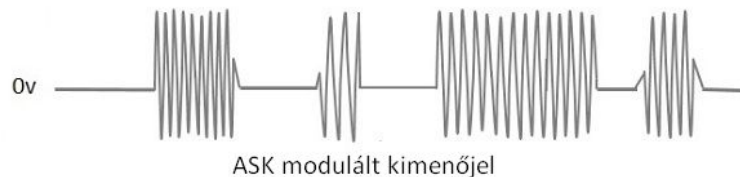
Jellemzői

- alacsony energiafelhasználás, jellemzően 1mW
- könnyen kezelhető
- tág hőmérsékleti határok között is működőképes
- viszonylag nagy hatótávú
- nem szükséges közvetlen rálátás



Rádiófrekvenciás digitális jelátvitel, ASK moduláció

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



Fix kódú encoder IC PT2260, SCT2260, PT2262

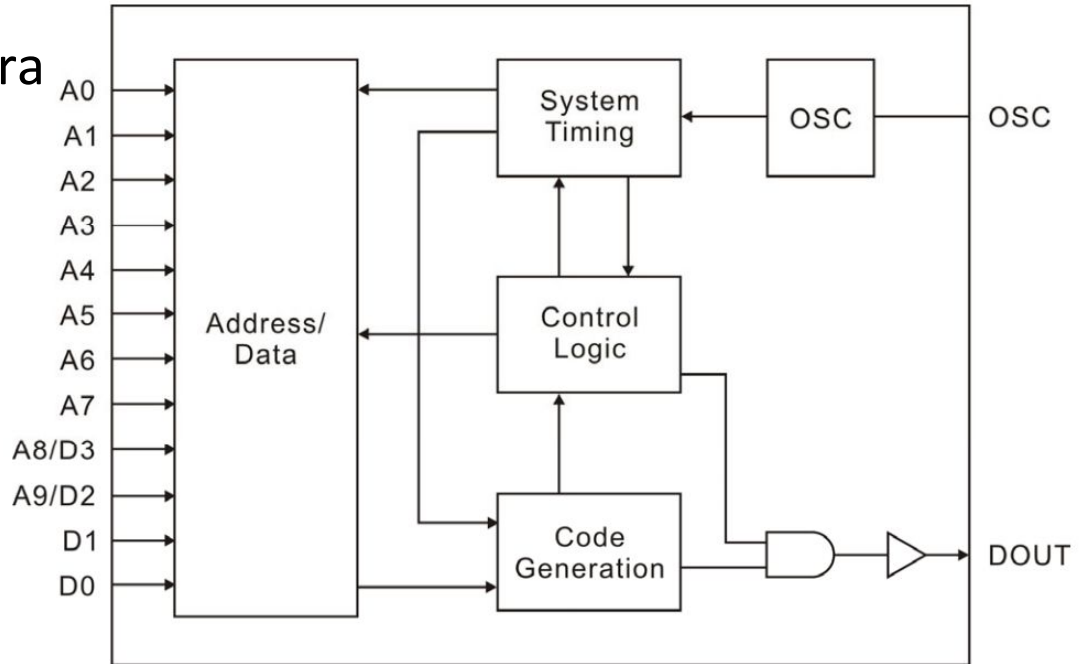
- 8 vagy 10 cím bit az egyediség biztosítására
3 értéket vehet fel, alacsony, magas, lebegtetett
- 4 vagy 2 adat bit

PT2260 max 10V

SCT2260 max 12V

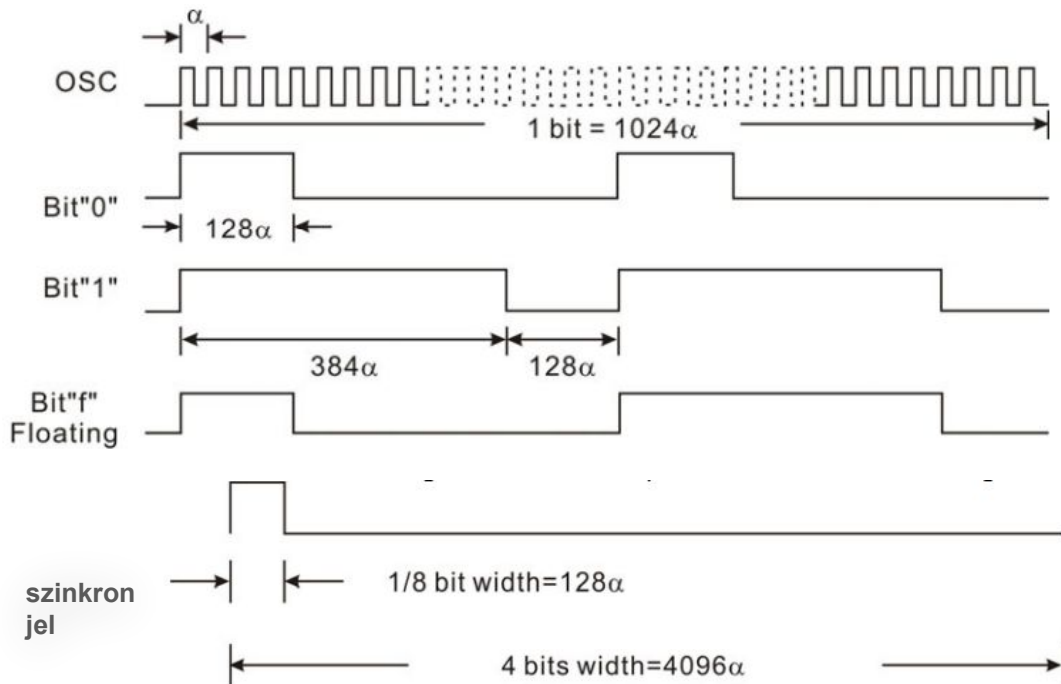
PT2262 max 14V

BLOCK DIAGRAM



2260 IC bitek kódolása

A szinkronjel kivételével 2 magas szint időbeli megoszlása kódolja a küldeni kívánt biteket. A szinkronjel egy rövid magas szint, a továbbiakban alacsony szint, összesen 4 bitnyi hossz



2260 IC lábkiosztása és kiküldött bitek sorrendje

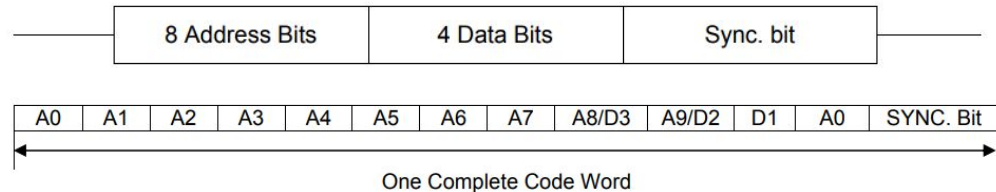
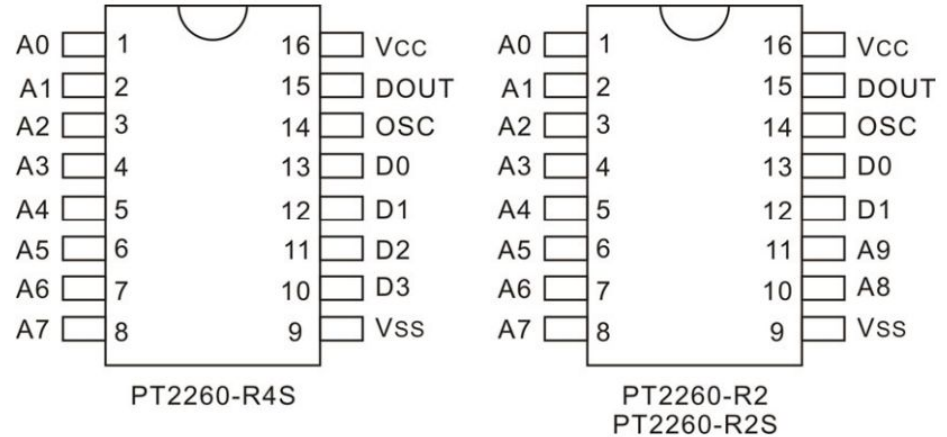
DIP vagy SOP tokozású

16 lábú IC

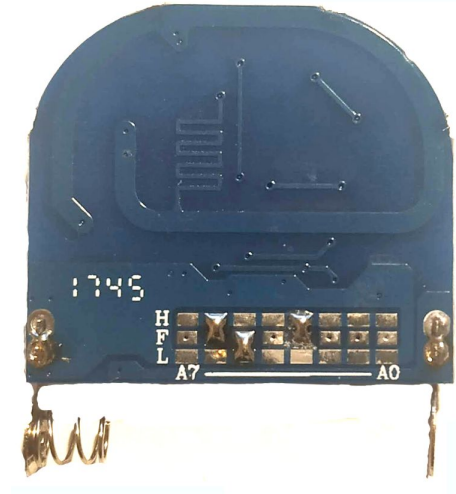
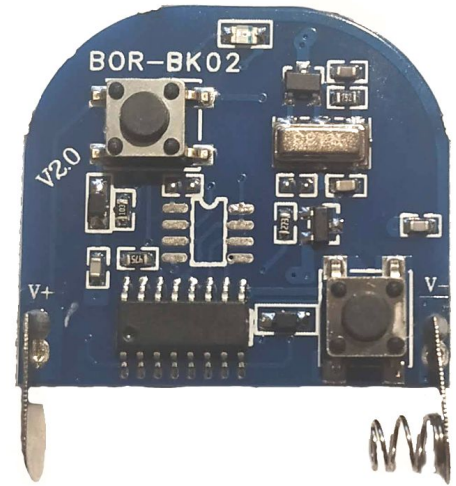
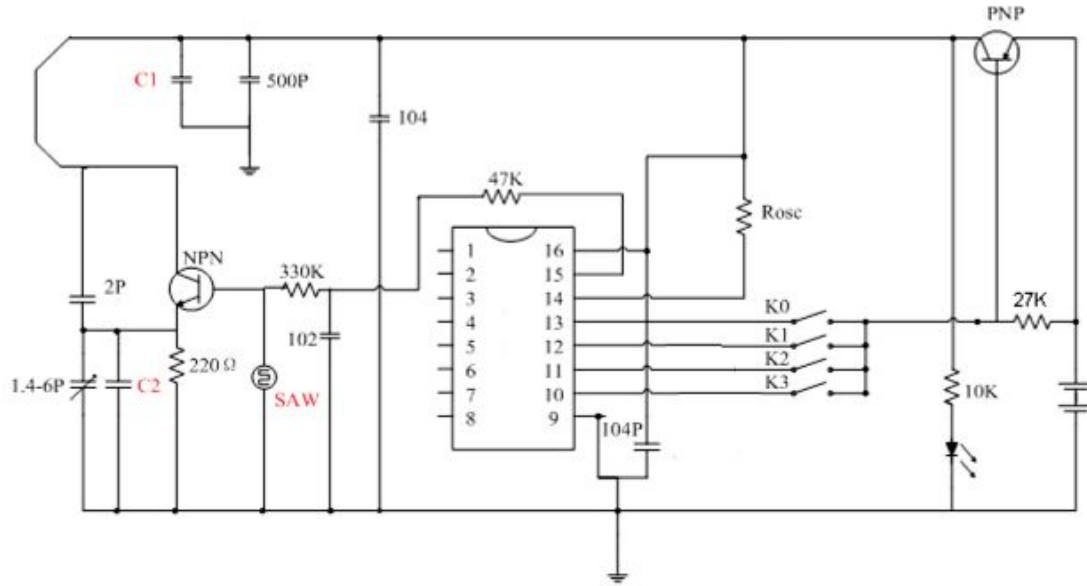
Először a címbitek átvitele történik meg, azt követik az adatbitek, a végén a szinkronizációs bit. Ezt az adatsort ismétli folytatólagosan



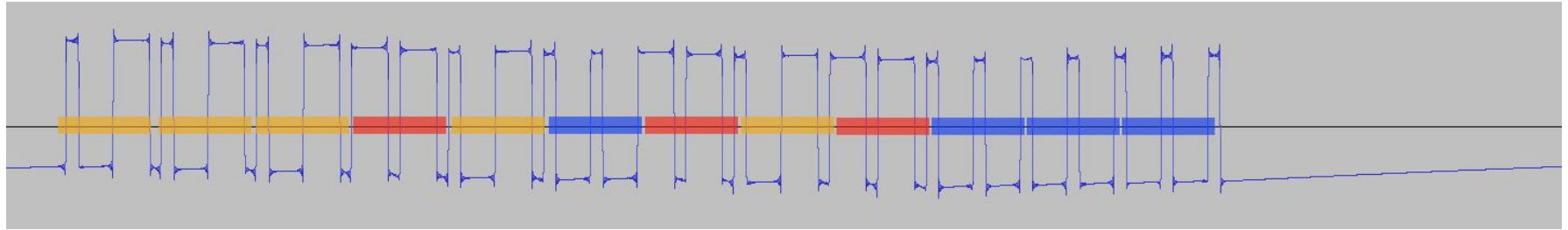
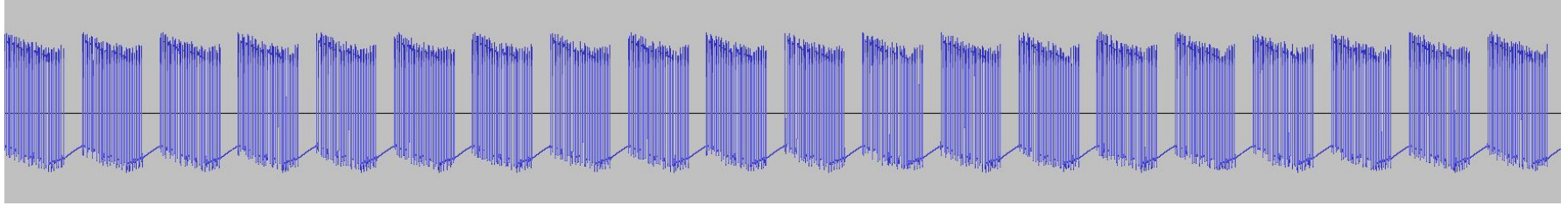
PIN CONFIGURATION



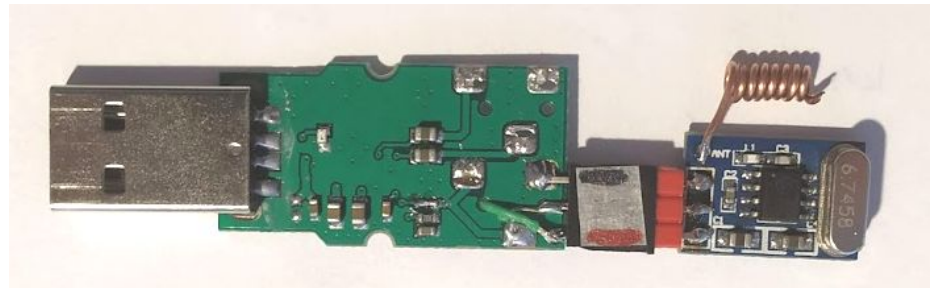
kapunyitó kapcsolási rajza



PT2260 jelek logolása audacity programmal



Elküldött bitek: **F F F 1 F0 1 F** 1000

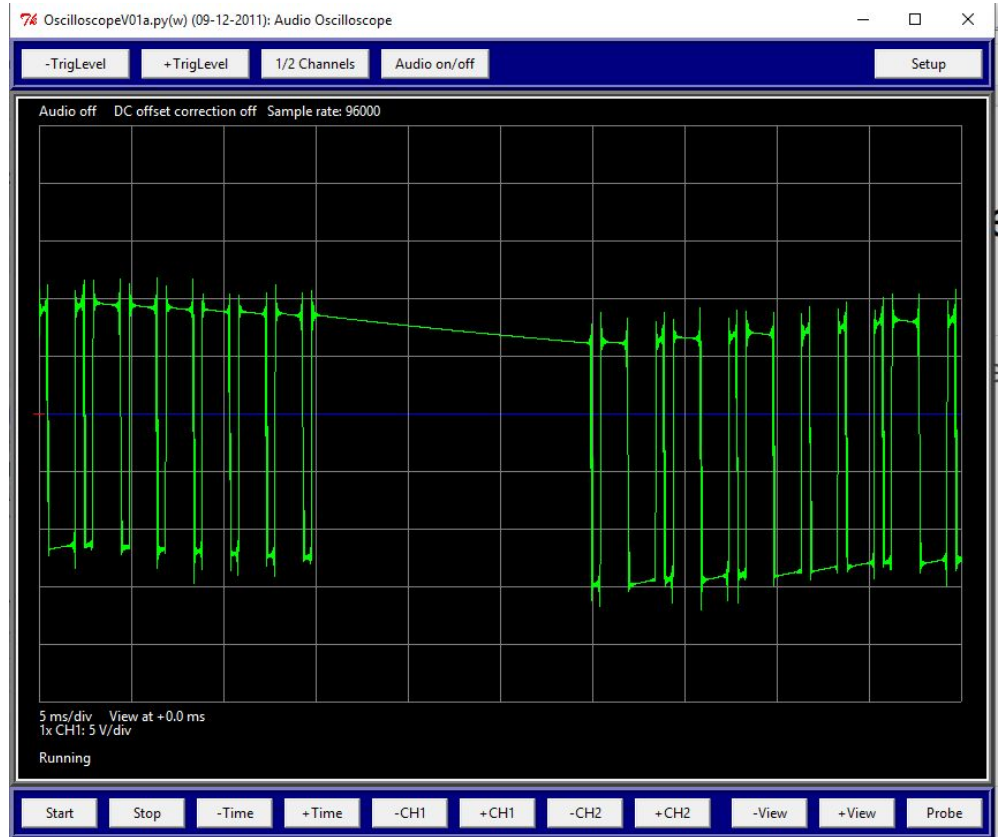
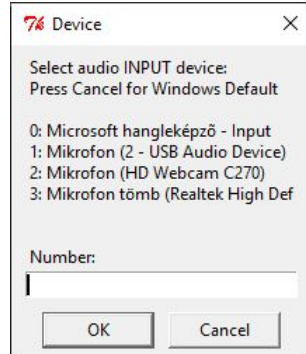


PT2260 jelek megtekintése python oszcilloszkóp programmal

Python2-Oscilloscope-v01a.pyw

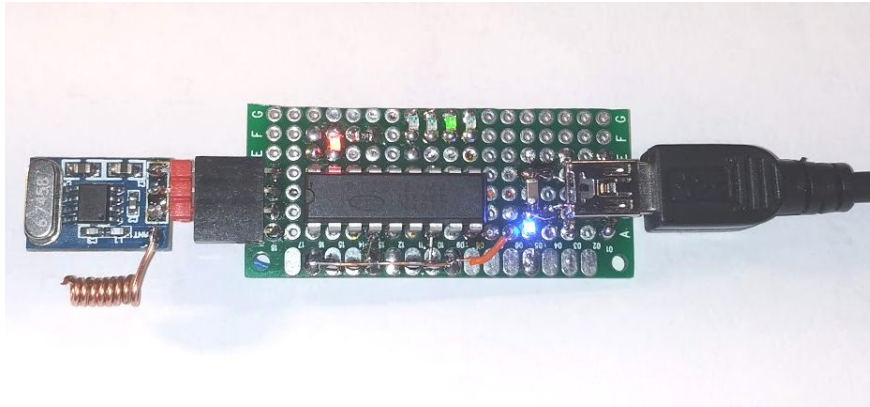
Válasszuk ki az USB Audio Device-t INPUT device-nak, majd megjelenik a fogadott jelek rajza,

a “- Time” gombbal módosítható a megjelenítendő időtartam

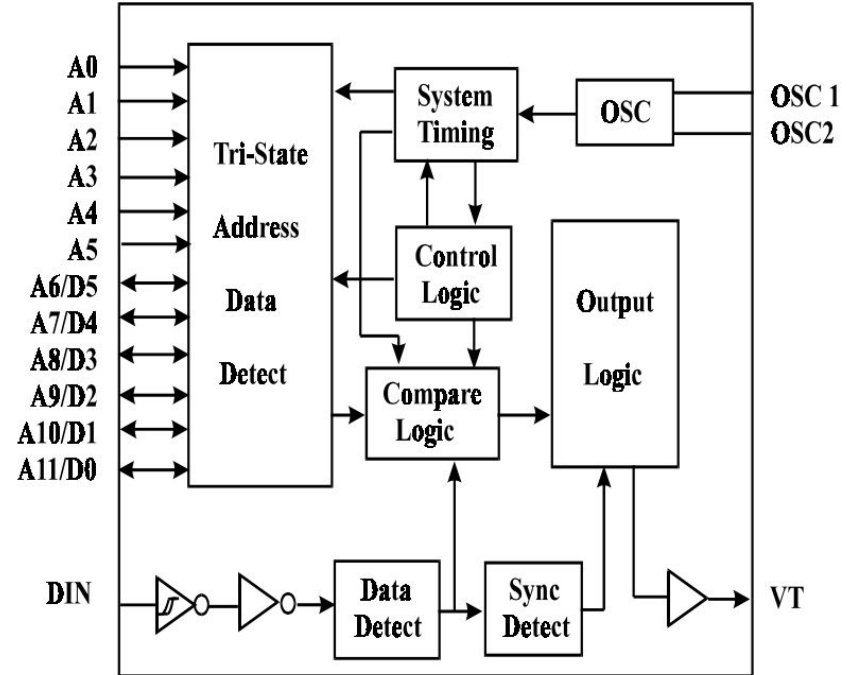


Decoder IC PT2272

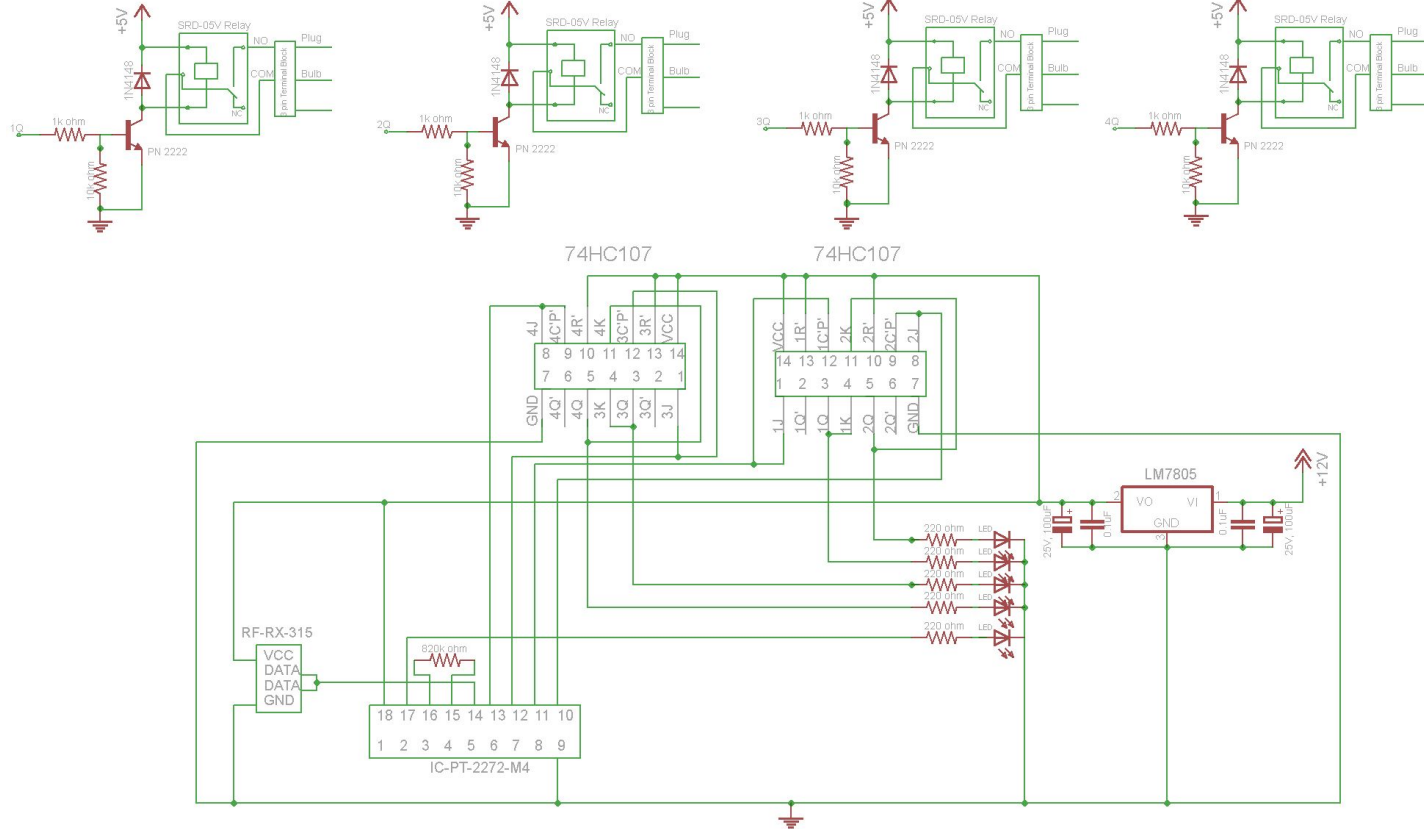
a beállított címmel egyező jelsor vétele esetén az adatlábakon megjelenik az átvitt érték



Block Diagram



PT 2272 IC felhasználása

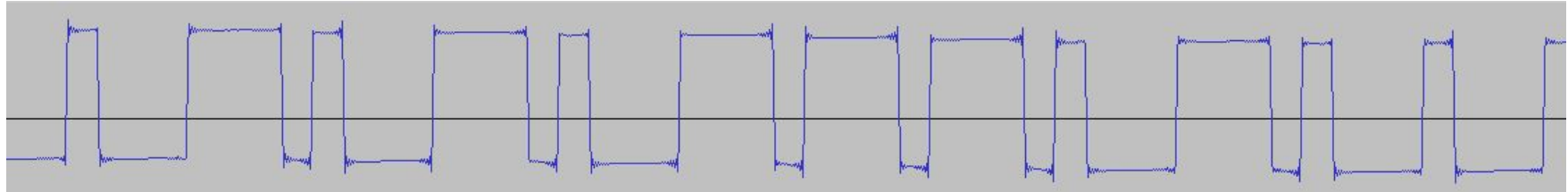


Micro Python kód

```
1 import pyb , time, array
2 dsize = 24 ; bsize = dsize+2 ; buff = array.array("I",[0] * bsize) ;up=False ; ert = None
3 LP=range(1400,1600) ; SP=range(400,600) ; SY=range(14500,15500)
4 rfpin = pyb.Pin('Y12', pyb.Pin.IN , pyb.Pin.PULL_NONE)
5 def f():
6     global elozo,ert, rfpin
7     up=rfpin()==1 ; e=time.ticks_us() ; a = e - elozo ; elozo = e
8     if up: ert = a
9 T_SER = time.ticks_ms() ; tart = 0 ; out = "" ; elozo=time.ticks_us()
10 ext = pyb.ExtInt(rfpin, pyb.ExtInt.IRQ_RISING_FALLING, pyb.Pin.PULL_NONE, lambda t: f() )
11 while True:
12     if ert!=None:
13         buff[tart]=ert ; tart = (tart+1)%bsize
14         if ert in SY and buff[tart] in SY:
15             i = tart ; raw = 0
16             for k in range(24):
17                 i=(i+1)%bsize ; raw <<= 1
18                 if buff[i] in SP: raw|=1
19                 elif buff[i] in LP: pass
20                 else: raw=0 ; break
21             if raw!=0 and out=="":
22                 for k in range(22,-2,-2):
23                     out += {0b00:"0",0b01:"F",0b10:"_",0b11:"1"}[ (raw >> k) & 0b11 ]
24             ert = None
25         if time.ticks_ms()-T_SER>100:
26             T_SER=time.ticks_ms()
27             if out!="": print( out ) ; out = ""
```

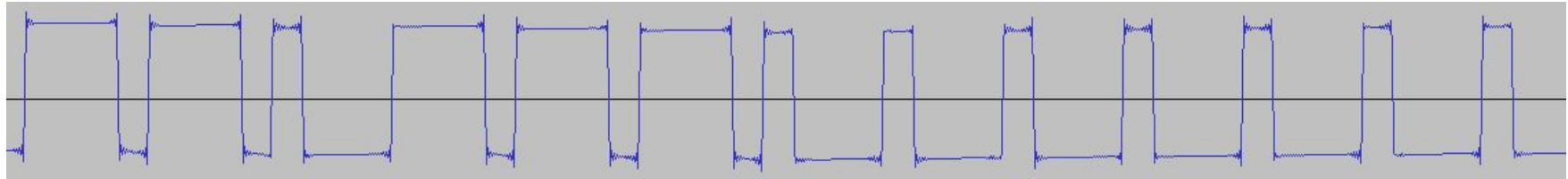
```
>>> import rfs
FFF1F01F1000
FFF1F01F1000
FFF1F01F1000
FFF1F01F1000
FFF1F01F1000
FFF1F01F1000
FFF1F01F0100
FFF1F01F0100
FFF1F01F0100
FFF1F01F0100
FFF1F01F0100
```

Fogadott bitek értelmezése



524 1432 1552 484 508 1464 1532 488 516 1456 1540 488 1536 496 1536 492 508 1464 1532 492 508 1476 500 1464
0 1 0 1 0 1 0 1 1 1 0 1 1 1 0 1 0 1 0 0 0

F F F 1 F 0



1532 500 1528 508 496 1468 1536 496 1532 500 1532 500 504 1476 500 1468 508 1468 504 1464 508 1460 512 1464
1 1 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

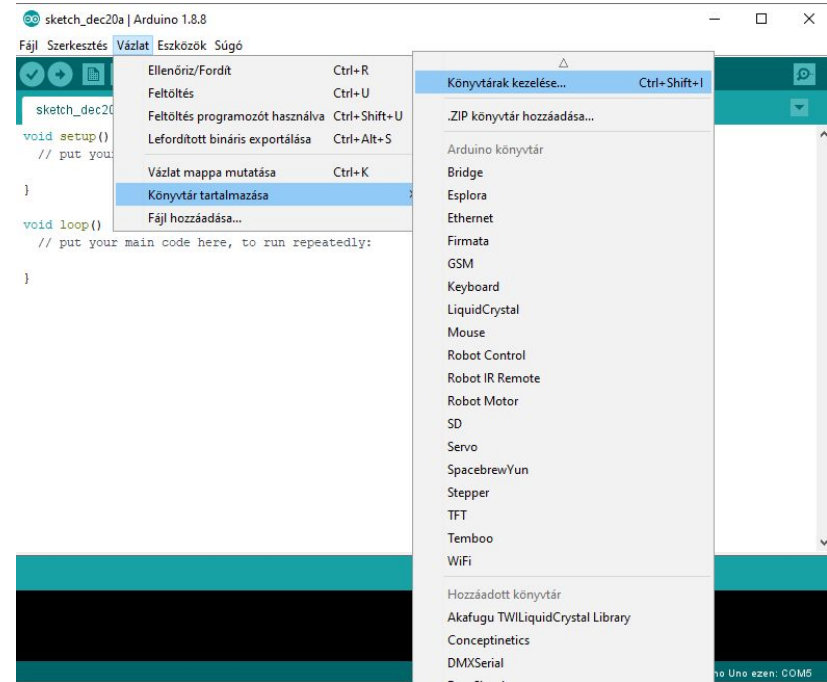
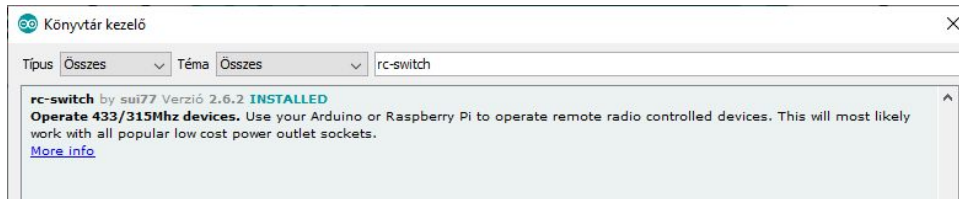
1 F 1 1 0 0 0 0 0 0 0 0

a bináris 010101110100110111000000 decimálisan: 5 721 536

RC switch arduino függvénykönyvtár telepítése

Vázlat menüpont Könyvtár
tartalmazása
ponton belül a Könyvtárak kezelése...

A beviteli mezőbe írjuk be: rc-switch
majd az rc-switch by sui77 pontnál,
nyomjunk a Telepítés gombra



RC switch használata

```
RCSwitch mySwitch = RCSwitch();

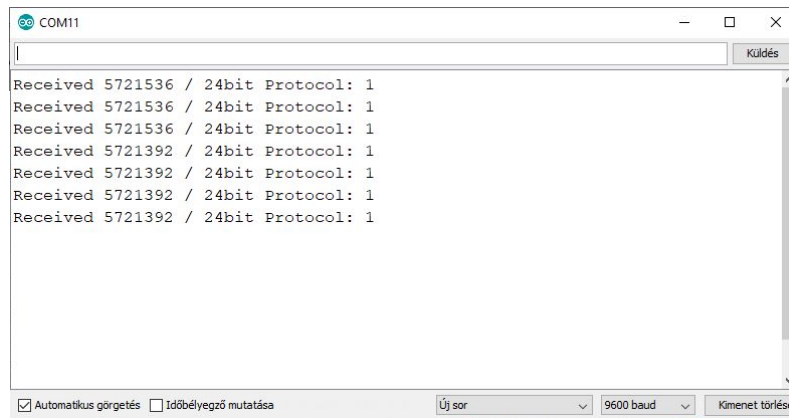
void setup() {
  Serial.begin(9600);
  mySwitch.enableReceive(0); // Receiver on interrupt 0 => that is pin #2
}

void loop() {
  if (mySwitch.available()) {

    int value = mySwitch.getReceivedValue();

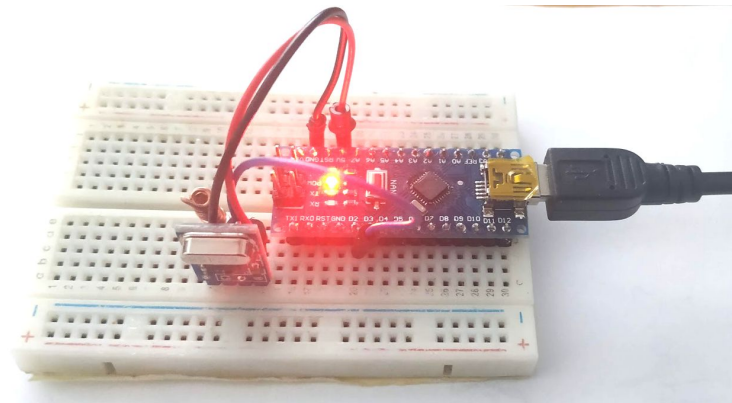
    if (value == 0) {
      Serial.print("Unknown encoding");
    } else {
      Serial.print("Received ");
      Serial.print( mySwitch.getReceivedValue() );
      Serial.print(" / ");
      Serial.print( mySwitch.getReceivedBitlength() );
      Serial.print("bit ");
      Serial.print("Protocol: ");
      Serial.println( mySwitch.getReceivedProtocol() );
    }

    mySwitch.resetAvailable();
  }
}
```



```
COM11
Received 5721536 / 24bit Protocol: 1
Received 5721536 / 24bit Protocol: 1
Received 5721536 / 24bit Protocol: 1
Received 5721392 / 24bit Protocol: 1
Received 5721392 / 24bit Protocol: 1
Received 5721392 / 24bit Protocol: 1
Received 5721392 / 24bit Protocol: 1

 Automatikus görgetés  Időbélyegző mutató
Új sor 9600 baud Kimenet törlése
```



RC switch fogadott adatok kiiratas

```
#include <RCSwitch.h>
```

```
RCSwitch mySwitch = RCSwitch();
```

```
void setup() {  
  Serial.begin(9600);  
  Serial.print("begin");  
  mySwitch.enableReceive(0); // Receiver on interrupt 0 => that is pin #2  
}
```

```
void loop() {  
  if (mySwitch.available()) {  
  
    int value = mySwitch.getReceivedValue();  
  
    if (value == 0) {  
      Serial.print("Unknown encoding");  
    } else {  
      unsigned int *b=mySwitch.getReceivedRawdata();  
      for (int i = 0 ; i<50; i++) { Serial.print( b[i] ); Serial.print(' '); }  
      Serial.println();  
    }  
    mySwitch.resetAvailable();  
  }  
}
```

14996

524 1432 1552 484 F

508 1464 1532 488 F

516 1456 1540 488 F

1536 496 1536 492 H

508 1464 1532 492 F

508 1476 500 1464 L

1532 500 1528 508 H

496 1468 1536 496 F

1532 500 1532 500 H

504 1476 500 1468 L

508 1468 504 1464 L

508 1460 512 1464 L

RC switch kódok küldése

```
#include <RCSwitch.h>
RCSwitch mySwitch = RCSwitch();
void setup() {
  Serial.begin(9600);
  // Transmitter is connected to Arduino Pin #10
  mySwitch.enableTransmit(10);
  // Optional set pulse length.
  // mySwitch.setPulseLength(320);

  // Optional set protocol (default is 1, will work for most outlets)
  mySwitch.setProtocol(1);
  // Optional set number of transmission repetitions.
  mySwitch.setRepeatTransmit(5);
}

void loop() {
  /* See Example: TypeA_WithDIPSwitches */
  mySwitch.sendTriState("FFF1F01F1000"); delay(100);
  mySwitch.sendTriState("FFF1F01F0100"); delay(100);
  mySwitch.sendTriState("FFF1F01F0010"); delay(100);
  mySwitch.sendTriState("FFF1F01F0001"); delay(100);
  delay(500);
}
```

