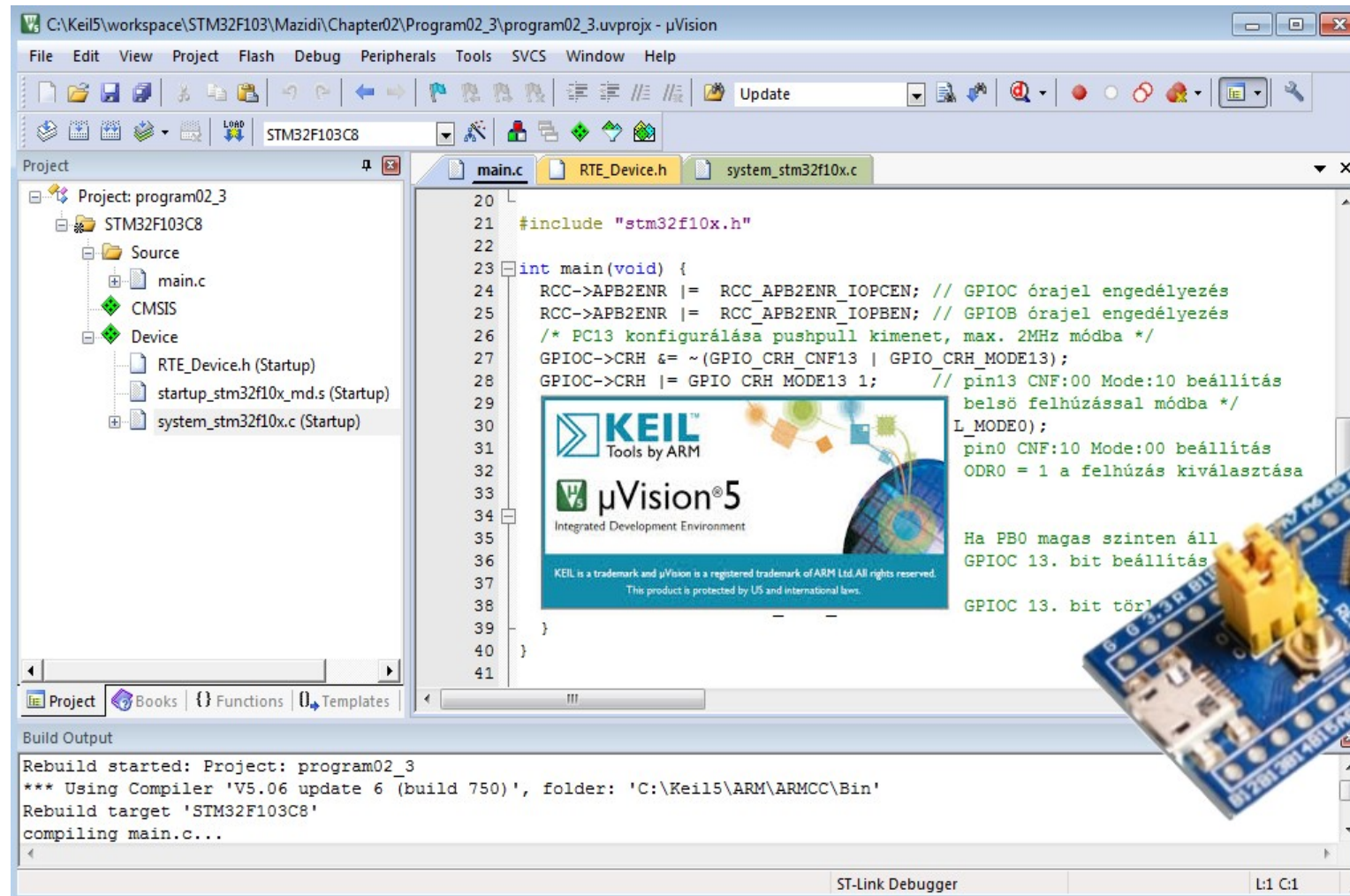













# STM32 mikrovezérlők programozása ARM Keil környezetben



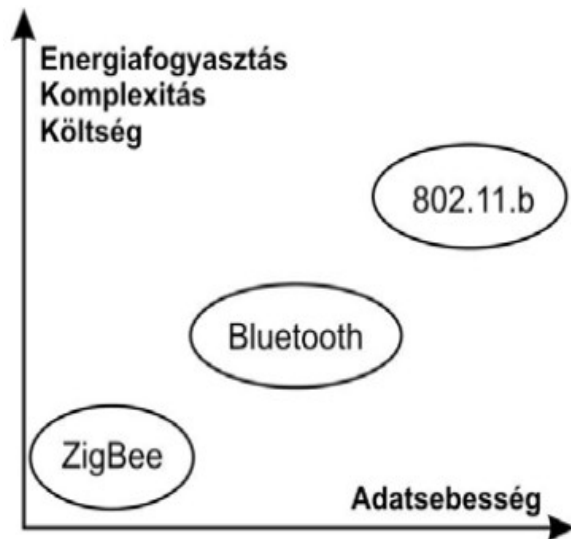
## 15. Vezeték nélküli kommunikáció (XBee/ZigBee)

# Felhasznált és ajánlott irodalom

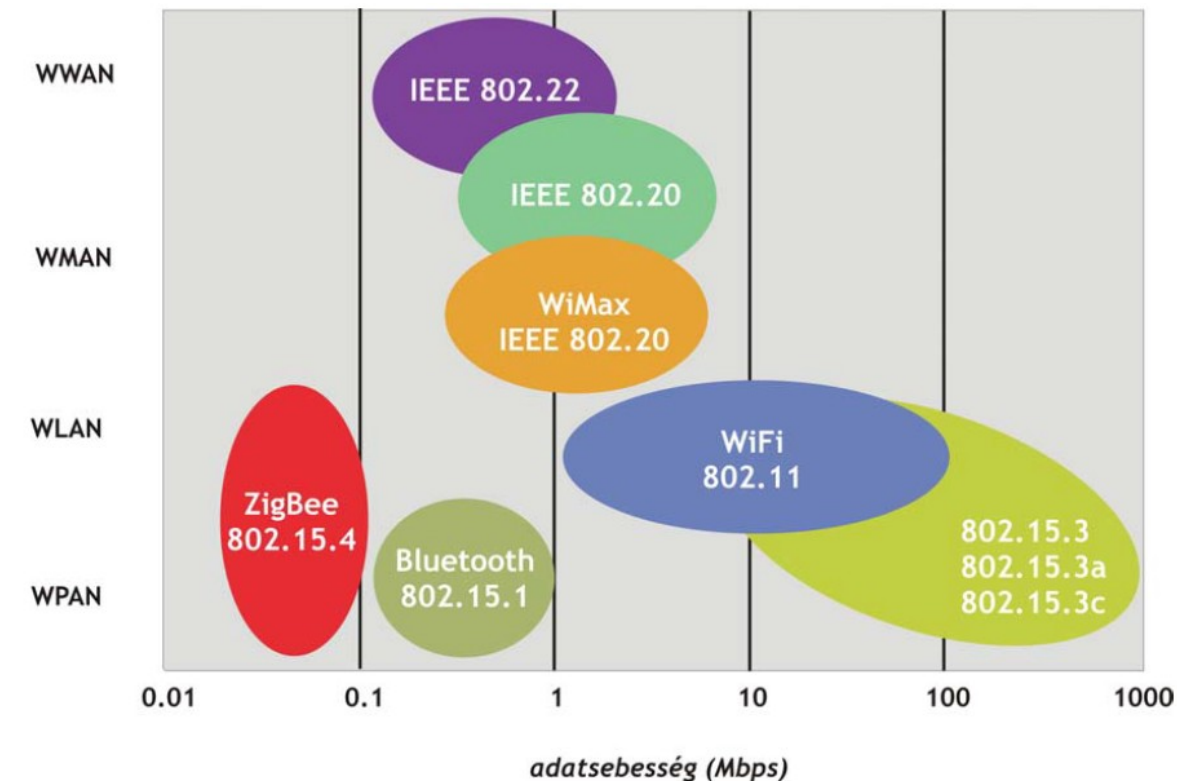
- Joseph Yiu: The Definitive Guide To The ARM CORTEX-M3 
- Muhammad Ali Mazidi, Shujen Chen, Eshragh Ghaemi: STM32 Arm Programming for Embedded Systems 
- Alexander Tarasov: Курс «Штудыям STM32» 
- ARM Keil MDK Getting started 
- **STM32F103C8** adatlap és termékinfo 
- **STM32F103** Family Reference Manual 
- **Varga Attila Károly**: ZigBee vezeték nélküli kommunikációs szabvány 
- **ZigBee Alliance**: ZigBee Technical Presentation 
- **Cytron Technologies**: XBee Series 2 point-to-pointcommunication 
- **DIGI International**: Zigbee RF Modules XBEE2, XBEEPRO2, PRO S2B 
- **DIGI International**: XTCU Configuration and Test Utility Software 

# ZigBee hálózatok

- A közepes és nagysebességű adatátvitelhez (hang, kép, videó), és PC-s hálózatok kiszolgálásához számos vezeték nélküli kommunikációs szabvány áll már rendelkezésre
- A szenzorok és a vezérlő egységek speciális igényeit kielégítő vezeték nélküli kommunikációs szabványokat a 2002-ben alakult **ZigBee Alliance** dolgozta ki („Egyszerűen működő vezeték nélküli vezérlés”)
- A **ZigBee** az IEEE 802.15.4 szabványon alapuló protokoll, amely (elvileg) lehetővé teszi különböző gyártóktól származó termékek együttműködését

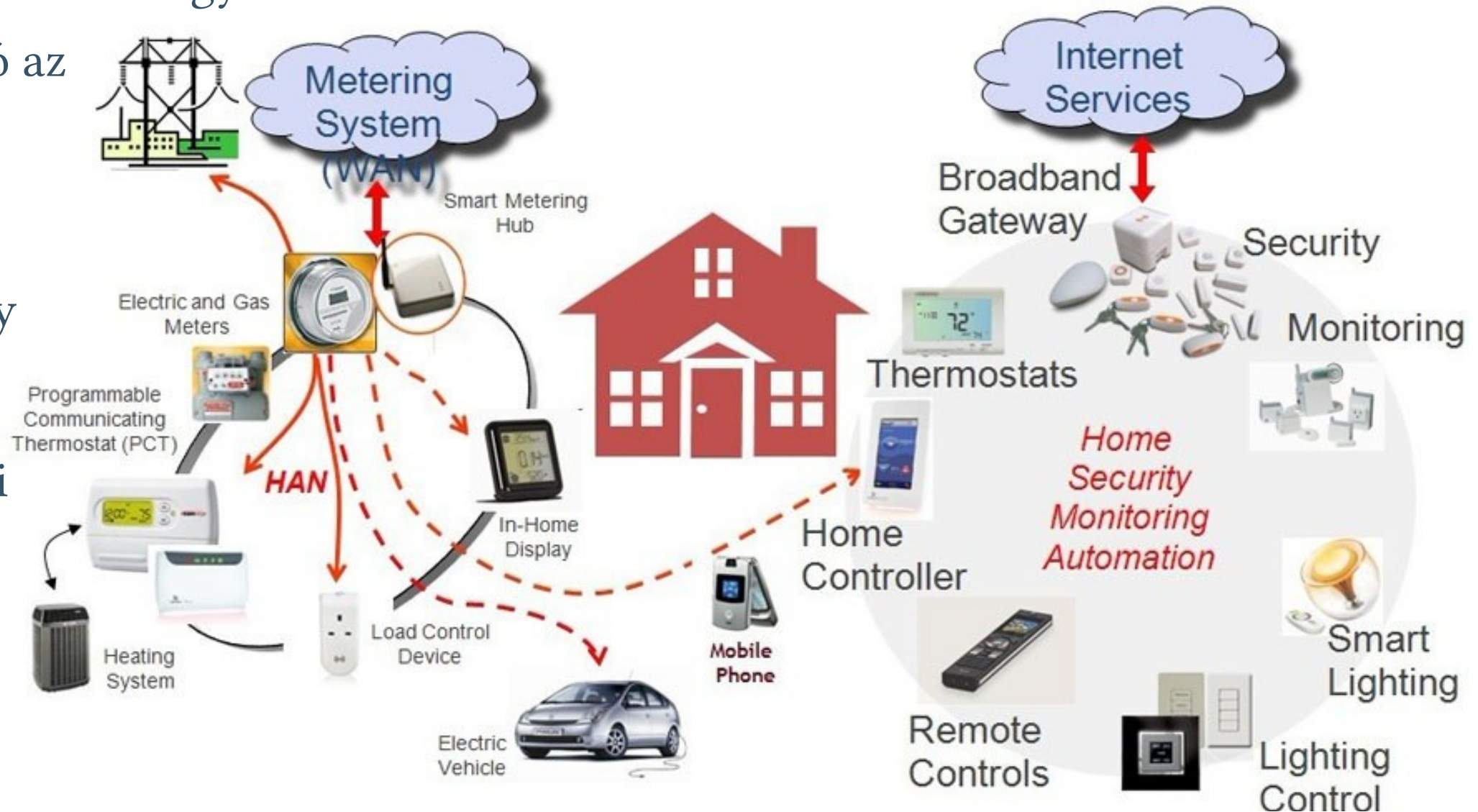


	Adatsebesség	Hatótáv	Alkalmazás
ZigBee	20 - 250 Kbps	10 - 100 m	Vezeték nélküli szenzor hálózatok
Bluetooth	1 - 3 Mbps	2 - 10 m	Vezeték nélküli headset
IEEE 802.11b	1 - 11 Mbps	30 - 100 m	Vezeték nélküli Internet kapcsolat



# Vezeték nélküli kommunikáció az okosotthon projektekben

- A **ZigBee** IOT eszközök fontos szerepet kapnak az okosotthon megoldásokban, mint vezeték nélküli szenzorok vagy vezérlők
- Ugyanakkor látható az igény az átjárásra a LAN vagy WAN hálózatokra is
- Fontos követelmény a biztonságos (titkosított) kommunikáció, ami nyilvánvalóan bonyolítja (és drágítja) az eszközöket



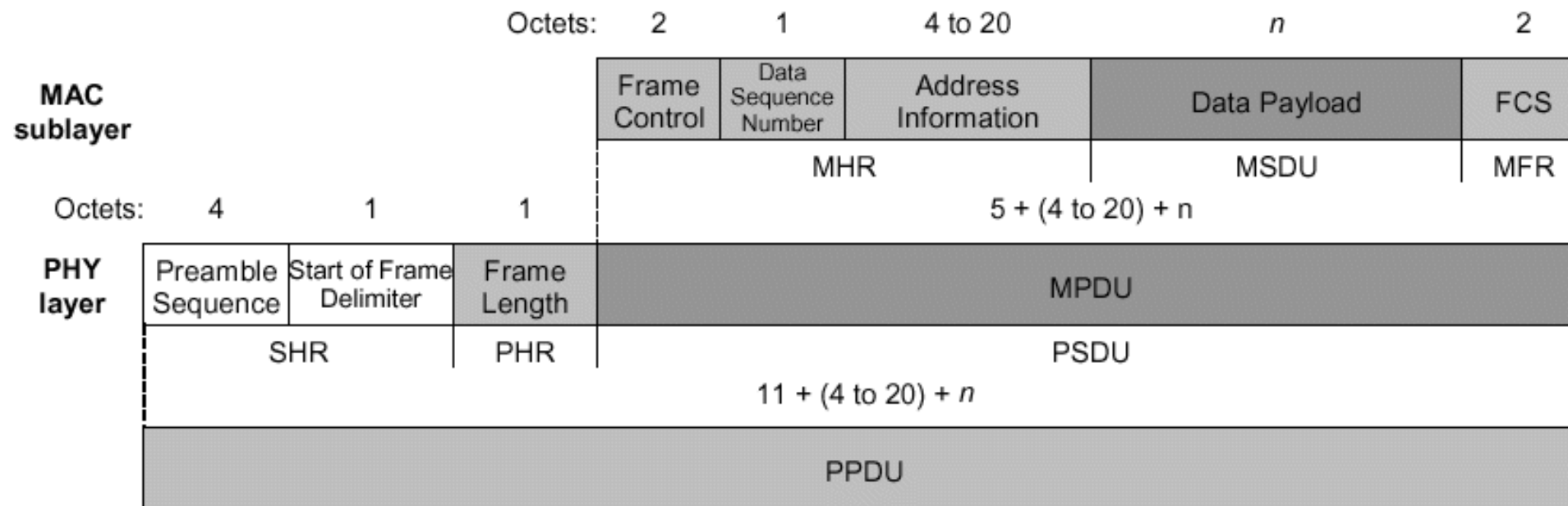
# Hogyan működik a ZigBee hálózat?

- IEEE 802.15.4 az ISM (ipari, tudományos, orvosi) nem engedélyköteles felhasználású frekvencia-sávokban kommunikálhatnak az eszközök

Frequency Band	License Required?	Geographic Region	Data Rate	Channel Number(s)
868.3 MHz	No	Europe	20kbps	0
902-928 MHz	No	Americas	40kbps	1-10
2405-2480 MHz	No	Worldwide	250kbps	11-26

# IEEE 802.15.4 Adatkeret Formátum

- Csomagonként legfeljebb 102 bájt adattovábbítási lehetőséget biztosít
- Adatsorrend számozás lehetővé teszi a csomagok nyomkövetését
- Adatkeret ellenőrző jelsorozat (FCS Frame Check Sequence) hibadetektálást biztosít
- min. 16 bájt = 128 bit = 0.512 ms @ 250 kbps
- max. 133 bájt = 1064 bit = 4.256 ms @ 250 kbps



# IEEE 802.15.4 eszköztípusok

## ■ Teljes funkcionalitású eszköz (FFD - Full function device)

- ❖ Több eszközzel kommunikálhat
- ❖ Általában hálózati tápellátású (mindig bekapcsolt állapotban van)
- ❖ Üzeneteket továbbküldhet (router)



## ■ Korlátozott funkcionalitású eszköz (RFD - Reduced Function Device)

- ❖ Korlátozott funkcionalitás a költségek vagy a komplexitás csökkentése érdekében
- ❖ Csak a szülőeszközzel kommunikál
- ❖ Kevesebb memóriát és erőforrást igényel
- ❖ Alvó (energiatakarékos) módba kapcsolhat
- ❖ Általában a hálózati végpontként használható



# 802.15.4 hálózati topológia modellek

## ■ Star (csillag) elrendezés jellemzői:

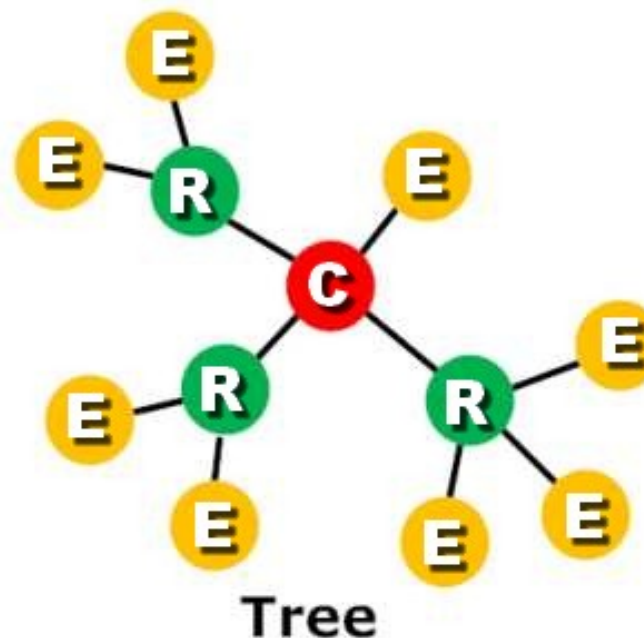
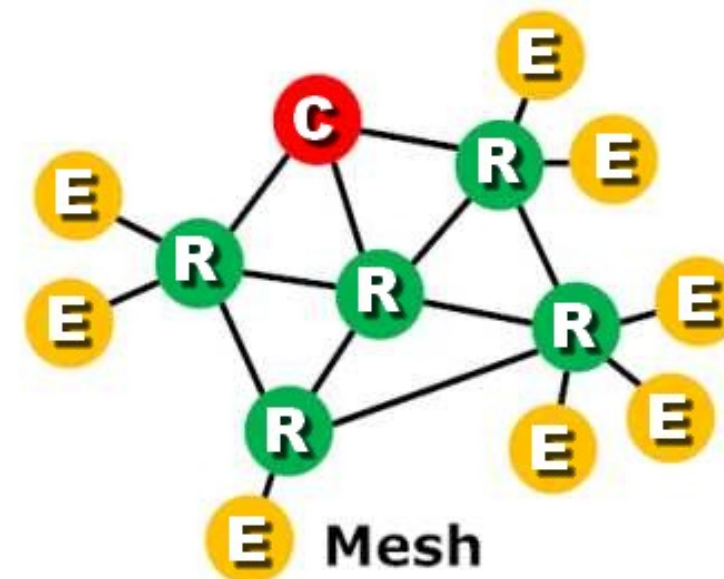
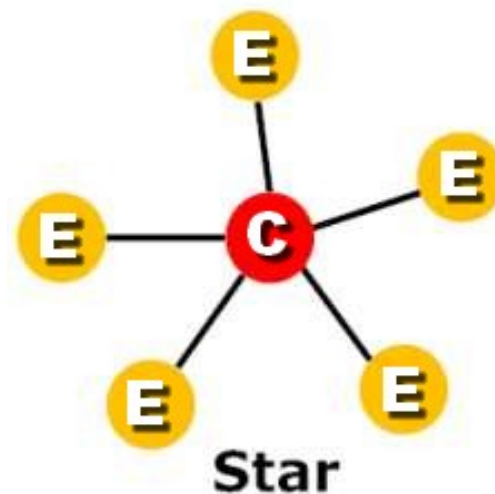
- ❖ Egyszerű felépítés
- ❖ Korlátozott kiterjedésű
- ❖ A koordinátor szűk keresztmetszet lehet

## ■ Tree (fa) elrendezés jellemzői:

- ❖ Kiterjeszti a hálózat elérhetőségét
- ❖ Szűk keresztmetszet még mindig lehet

## ■ Mesh (háló) elrendezés jellemzői:

- ❖ Komplex felépítés
- ❖ Nagy megbízhatóság
- ❖ Tehermentesíti a szűk keresztmetszetet

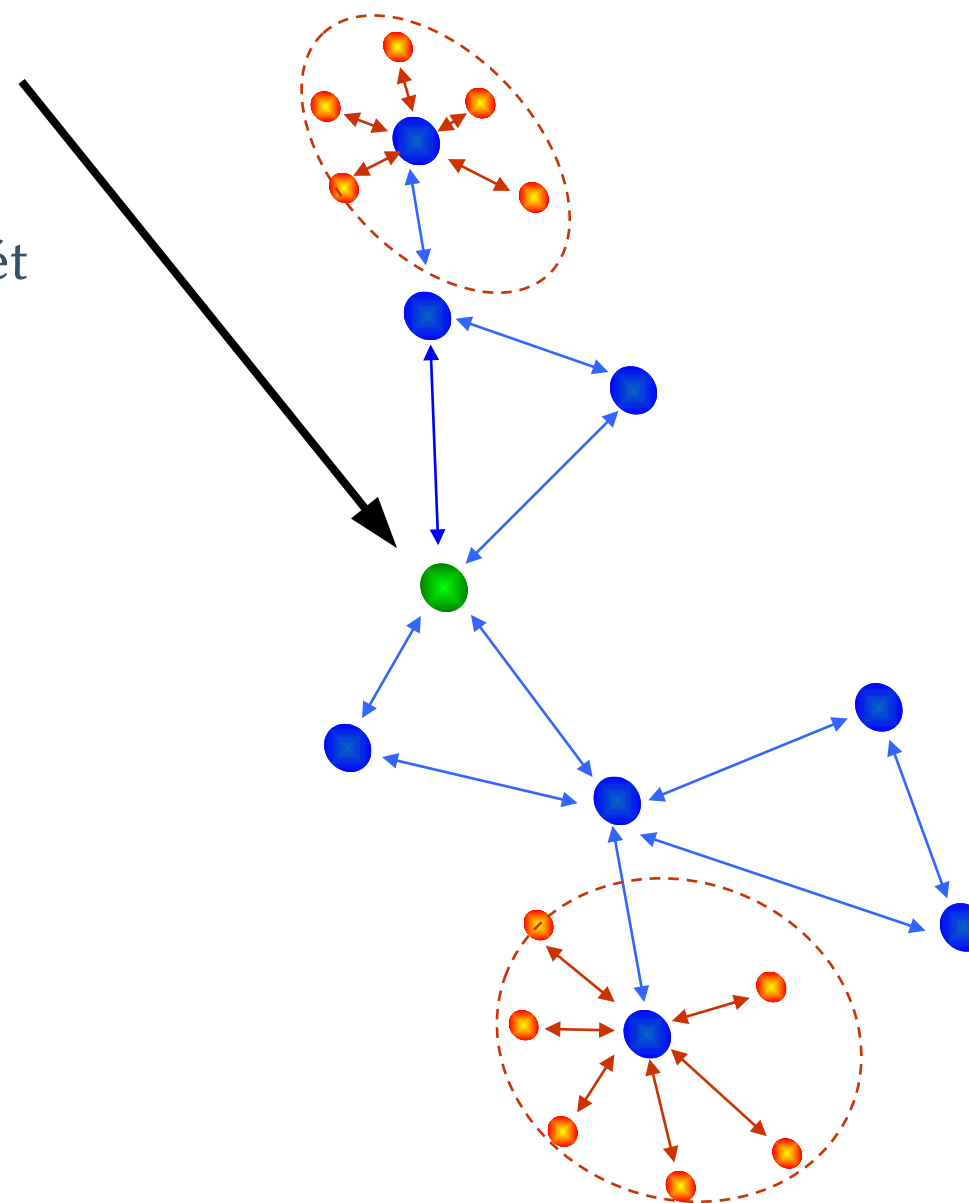




# ZigBee PAN Coordinator

## A ZigBee hálózat koordinátora

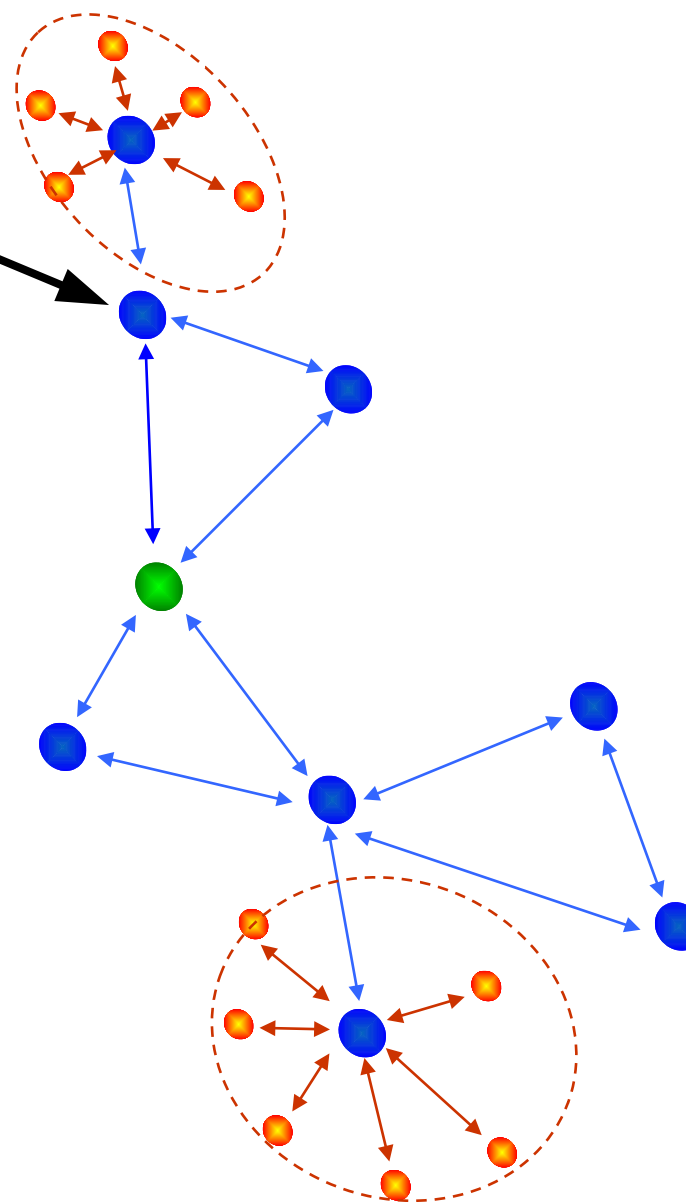
- „Birtokolja” a hálózatot
  - ❖ kezdeményezi az önszervező hálózat felépülését
  - ❖ megnyitja a hálózatot a csatlakozáshoz
  - ❖ kiosztja a hálózati címeket
  - ❖ kézbesítésig tárolja az üzeneteket
  - ❖ biztonsági központként működhet
- Teljes funkcionalitású eszköz – FFD
- Hálózati táplálású (mindig elérhető)
- Egyéb funkcionalitással is rendelkezhet
  - ❖ szenzor
  - ❖ monitor



# ZigBee Router

## A ZigBee hálózat csomópontja

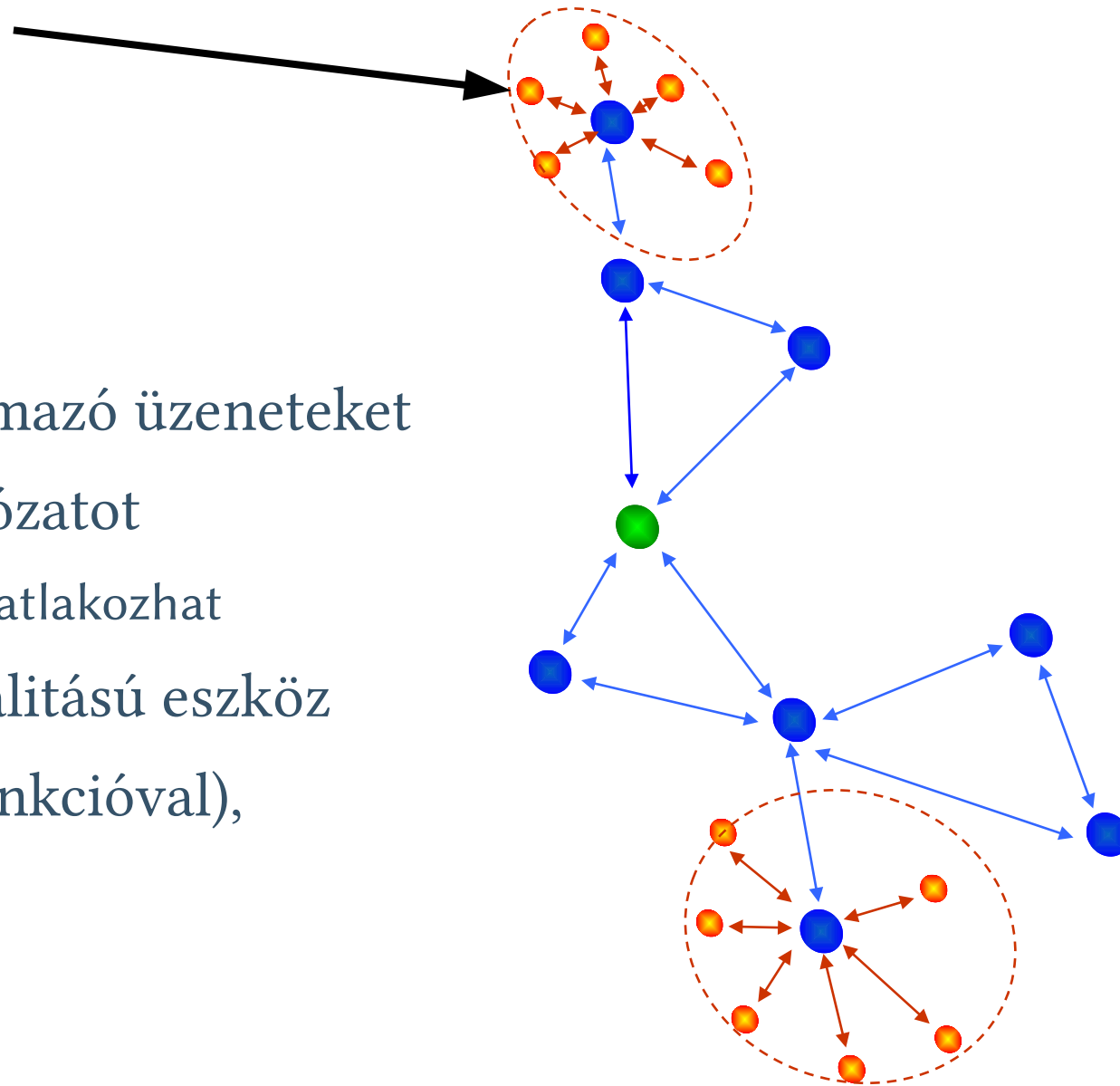
- Elirányítja/továbbítja az üzeneteket
- Nem birtokolja vagy indítja a hálózatot
  - ❖ Megkeresi a hálózatot, amelyhez csatlakozhat
- Teljes funkcionalitású eszköz – FFD
- Hálózati táplálású (mindig elérhető)
- Egyéb funkcionalitással is rendelkezhet
  - ❖ szenzor
  - ❖ monitor



# ZigBee End Device

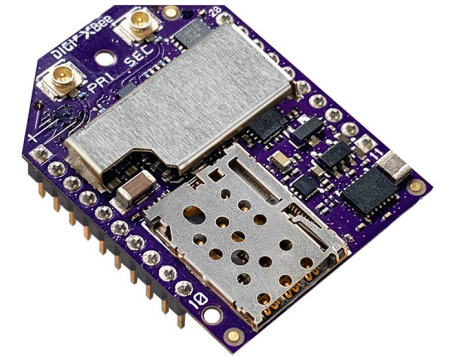
## A ZigBee hálózat végpontja

- Egyedi eszközfunkció, mint pl.
  - ❖ szenzor
  - ❖ monitor
- Nem irányít/továbbít mástól származó üzeneteket
- Nem birtokolja vagy indítja a hálózatot
  - ❖ Megkeresi a hálózatot, amelyhez csatlakozhat
- Teljes, vagy korlátozott funkcionalitású eszköz
- Elemes táplálású is lehet (alvás funkcióval), ezért nem mindig elérhető



# Xbee eszközök

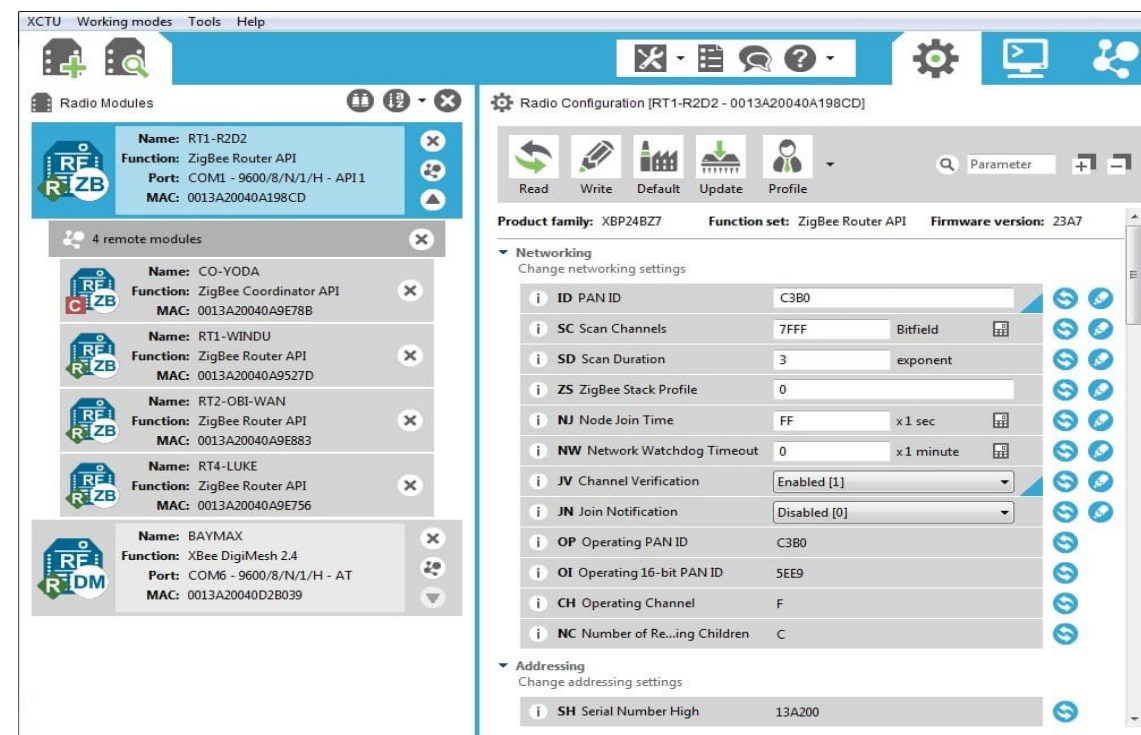
- A **DIGI International** sokféle vezeték nélküli kommunikációs eszközt gyárt **Xbee** elnevezéssel, a ZigBee, Bluetooth, WiFi, 3G, 4G és 5G hálózatok eszközei számára
- Az **Xbee** eszközök azonos formafaktorial készülnek, így a fejlesztés elején nem kell eldönteni, hogy milyen protokollt és frekvenciát használó modult válasszunk
- A szabványos hálózatok mellett a DIGI saját, egyedi hálózati protokollt is fejlesztett (**DigiMesh**), szélesítve a lehetőségek választékát
- Mi a **ZigBee** kompatibilis **Xbee Pro S2** (*XBP24-ZB termékcsalád*) modulokkal végeztünk kísérleteket. Meg kell jegyezni, hogy ezek megjelenése óta a **ZigBee** szabvány és az **Xbee** eszközök is tovább fejlődtek, ezért az **Xbee Pro S2** modulok új fejlesztéshez nem javasoltak, helyettük inkább a Digi XBee 3 Zigbee 3 modullal célszerű fejleszteni



# DIGI XBee fejlesztői eszközök

## ■ XCTU – Xbee Configuration & Test Utility

- ❖ ingyenes, multi-platmos alkalmazás Windows, MacOS és Linux kompatibilitással
- ❖ **Grafikus hálózat megjelenítés** az egyszerű és kényelmes hálózatkonfiguráláshoz
- ❖ **Firmware frissítés és konfigurálás**
- ❖ **API Frame Builder** megkönnyíti az XBee API üzenetcsomagok összeállítását és dekódolását



## ■ Digi XBee 3 Zigbee Mesh Kit

- ❖ Fejlesztői készlet 3 db **XBee 3 ZigBee 3** kártyával, 3 db USB-UART átalakító adapter kártyával és 3 db antennával

(ehhez hasonló, de olcsóbb eszközkészlettel fogunk kísérletezni mi is a továbbiakban)

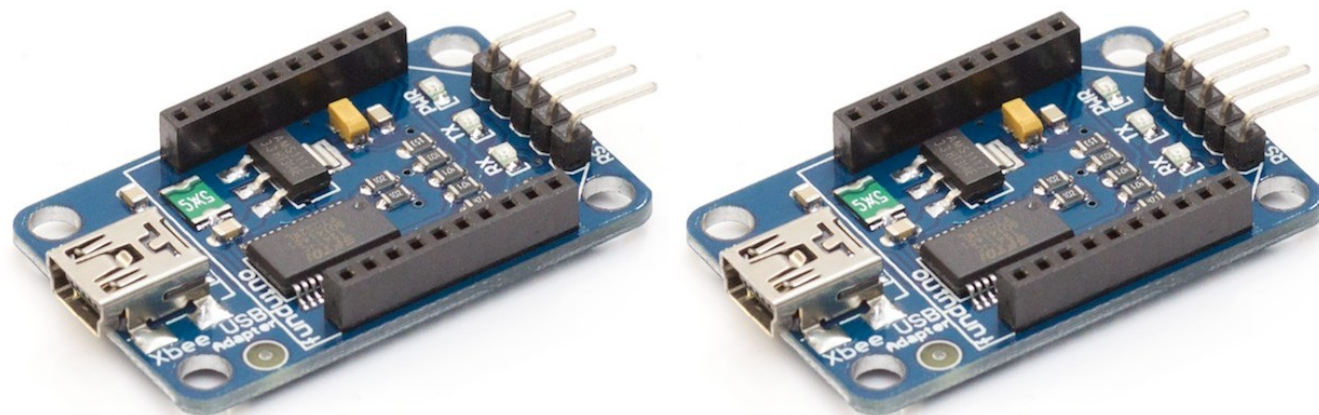
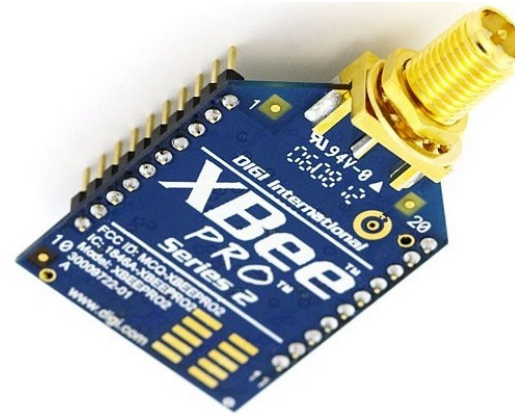


# Egyszerű pont-pont kapcsolat kialakítása

## ■ Hozzávalók:

- ❖ 2 db XBee Pro S2 modul
- ❖ 2 db 2.4 GHz-es antenna (anya)
- ❖ 2 db Funduino XBee USB adapter
- ❖ 2d USB mini kábel

- A számítógépre telepítsük fel az XCTU konfigurációs platform szoftvert!  
(jelenleg az **XCTU v.6.5.0** a legfrissebb kiadás)



# Az XCTU program kezelése

Modulok felismertetése

Eszközök

Konfigurálás

Console mód

Hálózati nézet

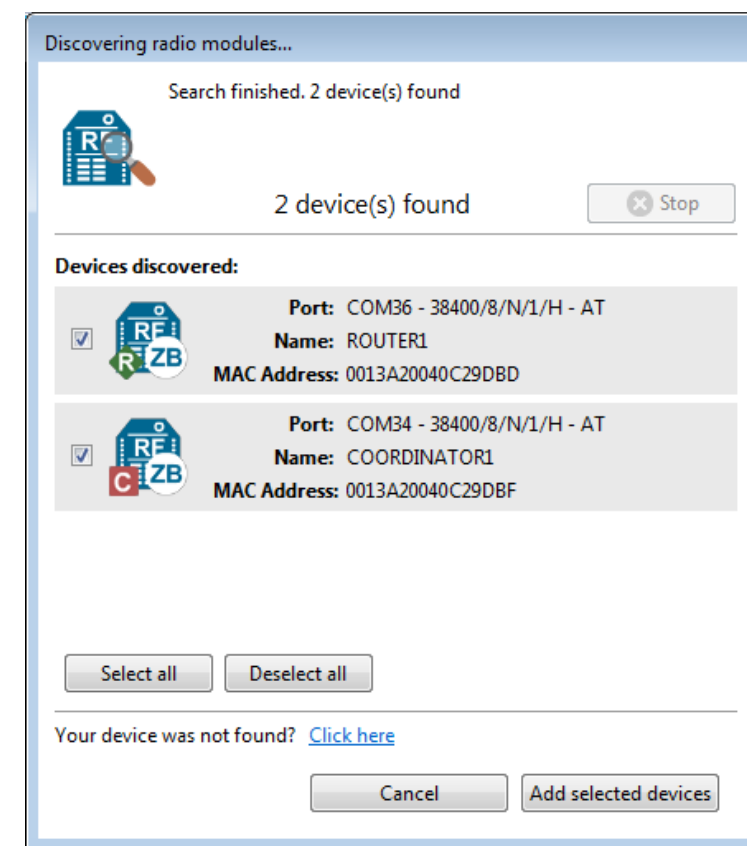
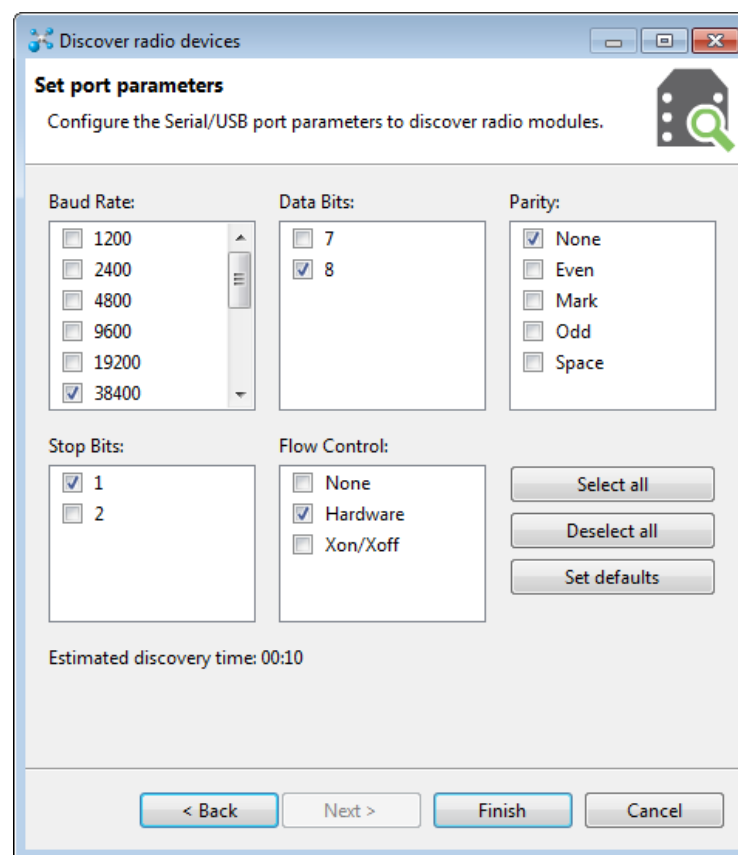
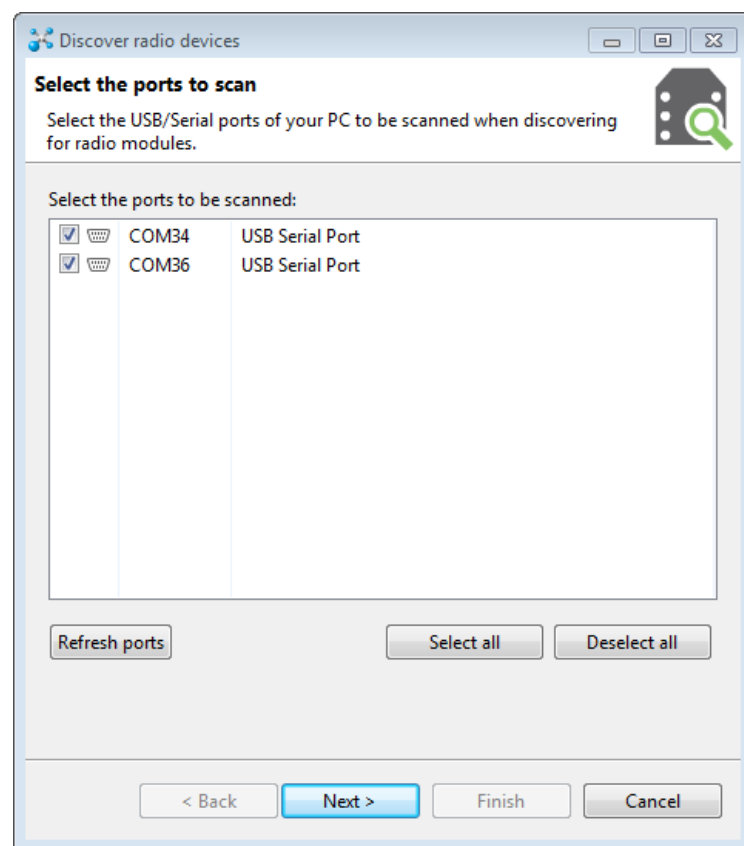
The screenshot displays the XCTU software interface. On the left, the 'Radio Modules' panel lists two modules: 'ROUTER1' (ZigBee Router AT) and 'COORDINATOR1' (ZigBee Coordinator AT). The 'COORDINATOR1' module is selected. The main area shows the 'Radio Configuration' for 'COORDINATOR1'. It includes a toolbar with 'Read', 'Write', 'Default', 'Update', and 'Profile' buttons. Below this, the 'Networking' section is expanded, showing a table of settings:

Parameter	Value	Unit/Type
ID PAN ID	1234	
SC Scan Channels	3FFF	Bitfield
SD Scan Duration	3	exponent
ZS ZigBee Stack Profile	0	

Ezek ún. helyi modulok, amelyek USB-UART illesztőn keresztül kapcsolódnak a számítógéphez

# A modulok felismertetése

- Csatlakoztatás után az **XCTU** programban kattintsunk a nagyítós ikonra (*Discover radio modules*)!
- Jelöljük be mindegyik soros portot és kattintsunk a **Next** gombra!
- Hagyjuk meg az alapértelmezett beállításokat és kattintsunk a **Finish** gombra!
- Sikeres felismertetés után kattintsunk a **Select all**, majd az **Add selected devices** gombokra!





# Firmware frissítés (ha szükséges...)

- A **ZigBee** hálózatokban az egyik modulnak koordinátor szerepet kell kapnia
- Ha szükséges, frissítsük a firmware-t az **Update** gombra kattintva, s a felbukkanó ablakban válasszuk ki az **XBP24-ZB** termékcsaládot, azon belül a **Zigbee Coordinator AT** firmware legfrissebb kiadását (20A7), majd kattintsunk az **Update** gombra!
- **Megjegyzés:** Az AT jelzésű firmware az ún. **transzparens módú** firmware

The screenshot displays the XCTU software interface. On the left, the 'Radio Modules' list shows two modules: 'COORDINATOR1' (ZigBee Coordinator AT) and 'ROUTER1' (ZigBee Router AT). The 'Radio Configuration' panel for 'COORDINATOR1' is active, showing options for 'Read', 'Write', 'Default', 'Update', and 'Profile'. The 'Update' button is circled in red. Below these options, the 'Product family' is set to 'XBP24-ZB' and the 'Function set' is 'ZigBee C...nator AT'. The 'Networking' section shows settings for 'ID PAN ID' (1234), 'SC Scan Channels' (3FFF), and 'SD Scan Duration' (3). An 'Update firmware' dialog box is open in the foreground, titled 'Update the radio module firmware'. It prompts the user to 'Configure the firmware that will be flashed to the radio module.' Below this, it asks to 'Select the product family of your device, the new function set and the firmware version to flash:'. A table lists the available options:

Product family	Function set	Firmware version
XBP24-B	ZigBee Coordinator AT	20A7 (Newest)
XBP24-SE	ZigBee End Device API	20A0
XBP24-ZB	ZigBee End Device AT	208C
	ZigBee End Device Analog IO	2070
	ZigBee End Device Digital IO	2064
	ZigBee End Device PH	2041
	ZigBee Router API	2021

At the bottom of the dialog, there is a link 'Can't find your firmware? [Click here](#)' and a 'View Release Notes' button.

# Firmware frissítés (ha szükséges...)



- A második modul **router** legyen, ehhez az **XBP24-ZB** termékcsalád **Zigbee Router AT** firmware legfrissebb kiadását (22A7) telepítsük az **Update** funkcióval!
- Ha a firmware frissítéssel végeztünk, folytassuk a modulok konfigurálásával!

The screenshot shows the XCTU software interface. The main window displays the 'Radio Configuration' for a device named 'ROUTER1 - 0013A20040C29DBD'. The 'Radio Modules' list on the left shows two modules: 'COORDINATOR1' (ZigBee Coordinator AT) and 'ROUTER1' (ZigBee Router AT). The 'Update firmware' dialog is open, showing the following table:

Product family	Function set	Firmware version
XBP24-B	ZigBee Router API	22A7 (Newest)
XBP24-SE	ZigBee Router AT	22A0
XBP24-ZB	ZigBee Router AT (WALL RT)	228C
	ZigBee Router Sensor	2270
	ZigBee Router/End Device Analog IO	2264
	ZigBee Router/End Device Digital IO	2242
	ZigBee Router/End Device Sensor	2241

# A modulok konfigurálása

- **PAN ID** (Personal Area Network ID): 0 és 0xFFFF közötti, mindkét modulra azonos szám
- **CH** (operating channel): a 2.4GHz 802.15.4 sáv valamelyik csatornája (azonosan legyen!)
- **Címzés**: minden modul rendelkezik saját **MAC** azonosítóval (**SH, SL**), illetve egy általunk adott tetszőleges hálózati azonosítóval (**NI**). Az ellenoldali MAC címet (Destination Address) pedig a (**DH,DL**) paraméterekkel állítható be

 <b>Name:</b> COORDINATOR1 <b>Function:</b> ZigBee Coordinator AT <b>Port:</b> COM34 - 38400/8/N/1/H - AT <b>MAC:</b> 0013A20040C29DBF	 <b>Name:</b> ROUTER1 <b>Function:</b> ZigBee Router AT <b>Port:</b> COM36 - 38400/8/N/1/H - AT <b>MAC:</b> 0013A20040C29DBD
<b>ID PAN ID:</b> 1234 <b>CH Operating channel:</b> 14 <b>SH:</b> 0013A200 <b>SL:</b> 40C29DBF <b>DH:</b> 0013A200 <b>DL:</b> 40C29DBD	<b>ID PAN ID:</b> 1234 <b>CH Operating channel:</b> 14 <b>SH:</b> 0013A200 <b>SL:</b> 40C29DBD <b>DH:</b> 0013A200 <b>DL:</b> 40C29DBF

Red arrows indicate the mapping of SH to SL and SL to SH. Blue arrows indicate the mapping of DL to DH and DH to DL.

Serial Interfacing  
Change modem interfacing options

i BD Baud Rate	38400 [5]
i NB Parity	No Parity [0]
i SB Stop Bits	One stop bit [0]
i RO Packetization Timeout	3 x character times
i D7 DIO7 Configuration	CTS flow control [1]
i D6 DIO6 Configuration	RTS flow control [1]

# Kommunikációs próba

- 1) Kattintsunk a Terminál ikonra (konzol mód)!
- 2) Kapcsolódjunk a hálózatra a COORDINATOR1 modullal!
- 3) Válasszuk le COORDINATOR1 terminál ablakát!
- 4) Csatlakozzunk a ROUTER1 modullal is!

## Transzparens mód:

Amit az egyik modulnál beírnunk, azt a másik modul veszi és kiírja

A +++ beírása AT parancsmódba kapcsol, ATCN pedig kilép belőle

ATNI parancs az NI (Network ID) lekérdezésére szolgál

ATCN parancs visszatérés a Console Módba

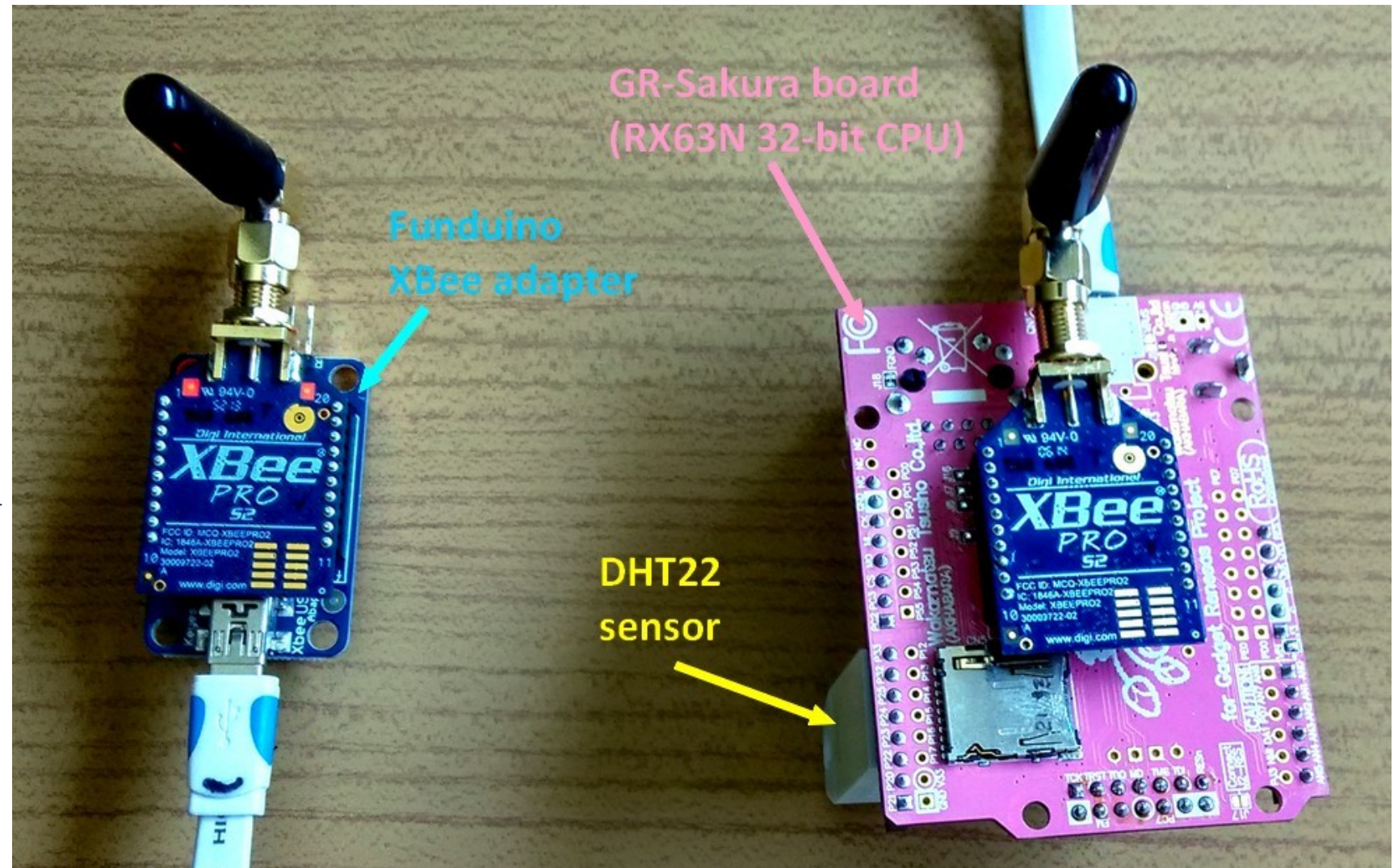
Ha eddig sikeresen eljutottunk, akkor az Xbee modulok készen állnak a munkára!

The screenshot shows the XCTU software interface. The top toolbar contains several icons, with the terminal icon (a green monitor with a white cursor) circled in red and labeled '1.'. Below the toolbar, there are two terminal windows. The left window is titled 'ROUTER1 - 0013A20040C29DBD' and has a green terminal icon circled in red and labeled '4.'. The right window is titled 'COORDINATOR1 - 0013A20040C29DBF' and has a green terminal icon circled in red and labeled '2.'. In the ROUTER1 window, the 'Attach' icon (a green monitor with a white cursor) is circled in red and labeled '3.'. The console logs in both windows show the following text:

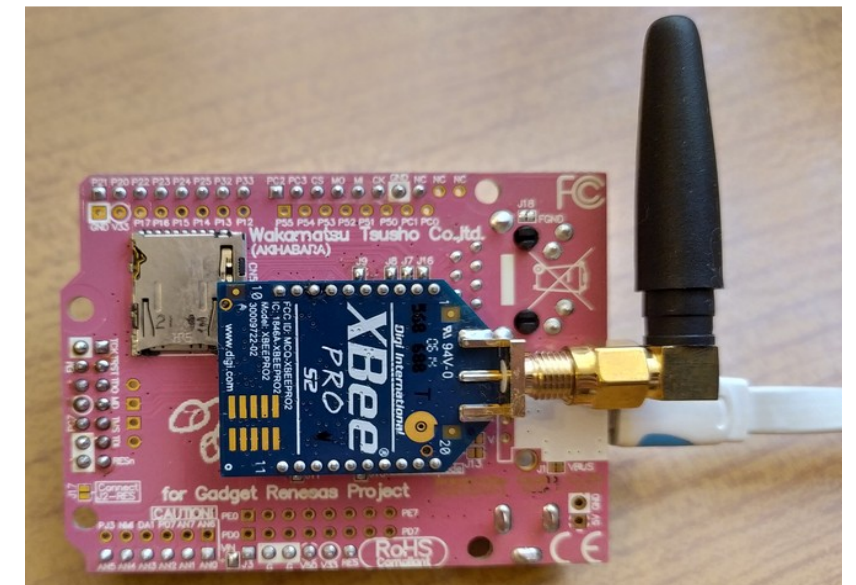
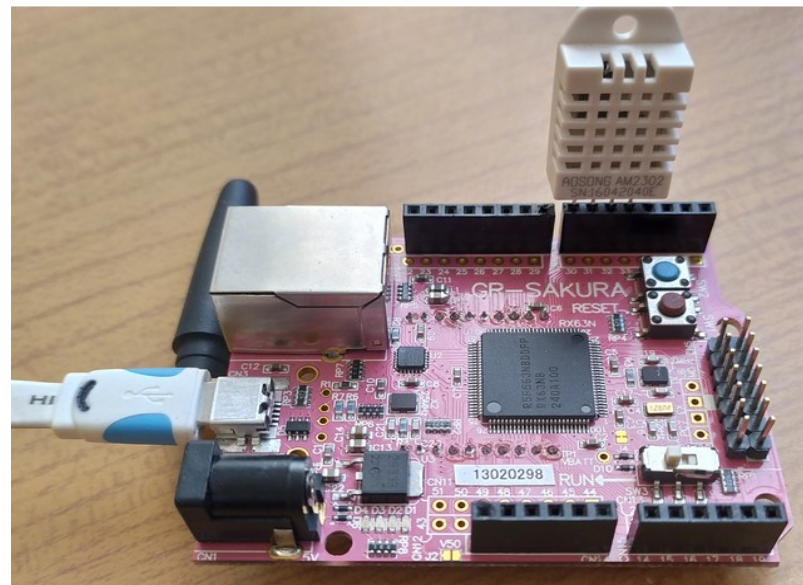
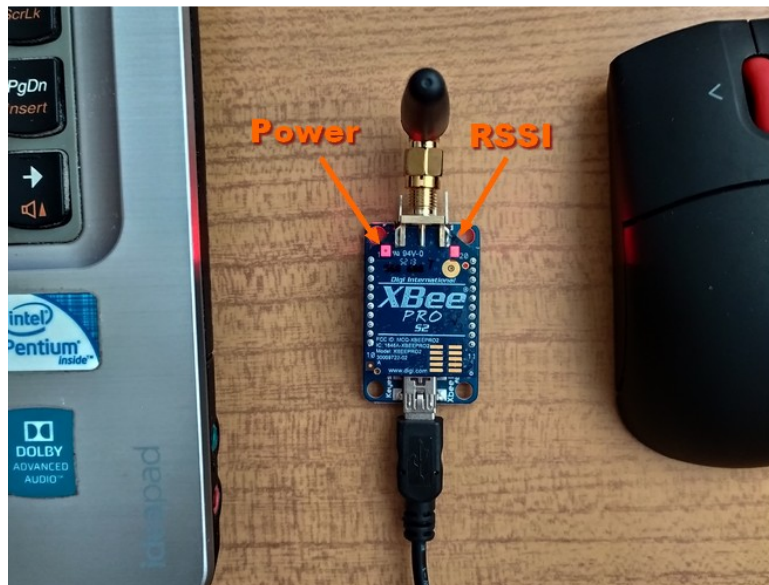
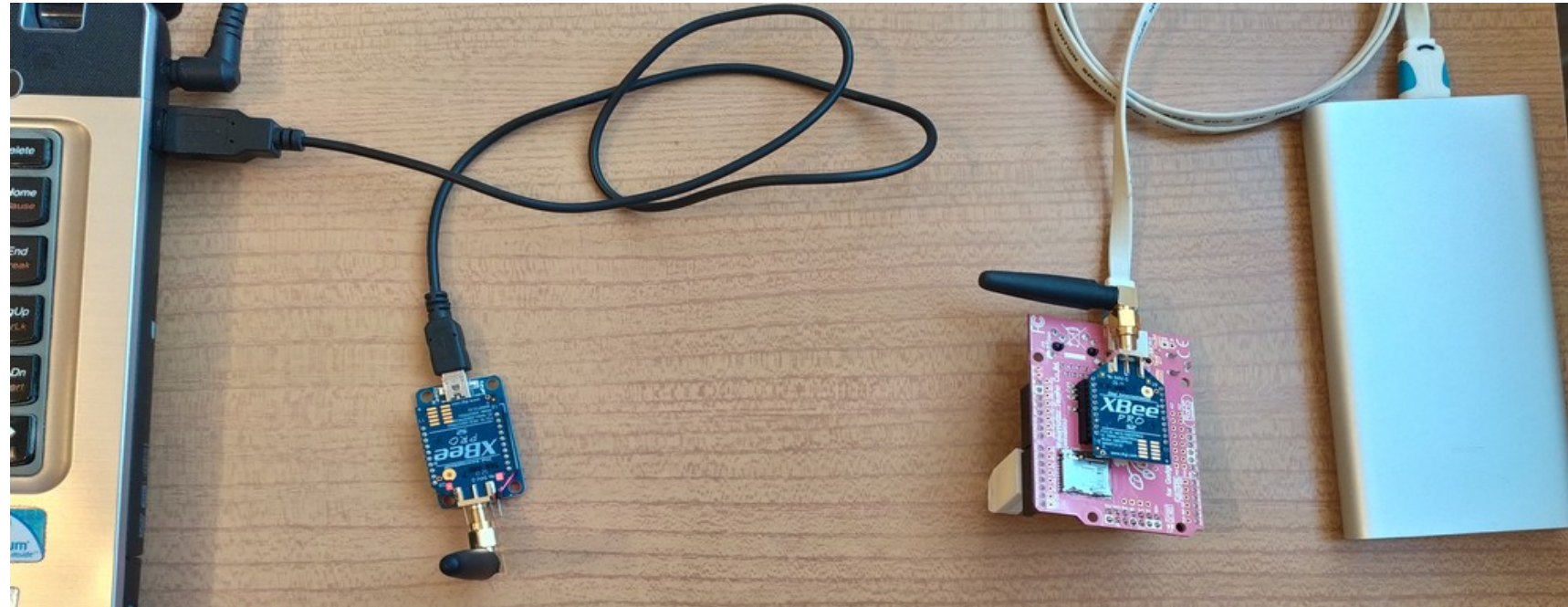
```
Good morning! 47 6
Hi there! 48 6
+++OK 2B 2
ATNI 41 5
ROUTER1 52 4
ATCN 41 5
OK 4F 4
Continue to work... 43 6
Rvoo an
```

# Vezeték nélküli hőmérő

- Első kísérletünkhöz egy Xbee foglalattal rendelkező **Renesas GR-Sakura** kártyát használtunk és egy **DHT22** hőmérséklet és relatív páratartalom mérő szenzort. Ehhez a kártyához a **ROUTER1** modult csatlakoztattuk.
- A szenzor kiolvasásához és az eredmény soros porton történő kiküldéséhez írunk kell egy kis programot.
- A **COORDINATOR1** modul az USB adapter kártyán keresztül továbbra is a számítógéphez csatlakozzon!
- A számítógépen egy **Processing** programmal jelenítjük meg az eredményt.



# További képek az elrendezésről



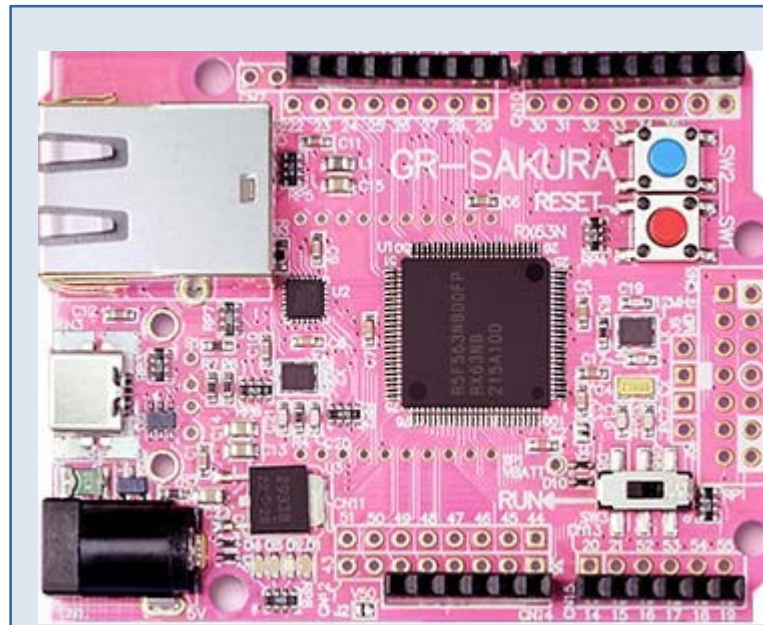
# DHT22\_lazy\_log.ino

```
#include <stdint.h>
#include "DHT.h"
#define DHTPIN 6           // what pin we're connected to
#define DHTVCC 7          // must be set to high
#define DHTGND 4          // must be set to low
#define DHTTYPE DHT22    // DHT-22 (AM2302)

DHT dht(DHTPIN, DHTTYPE,20);

void setup() {
  pinMode(DHTVCC,OUTPUT);  digitalWrite(DHTVCC,HIGH); // Provides VCC for DHT-22
  pinMode(DHTGND,OUTPUT); digitalWrite(DHTGND, LOW); // Provides GND for DHT-22
  Serial3.begin(38400);    // Xbee module connects to Serial3 UART port
  delay(500);
  dht.begin();
}

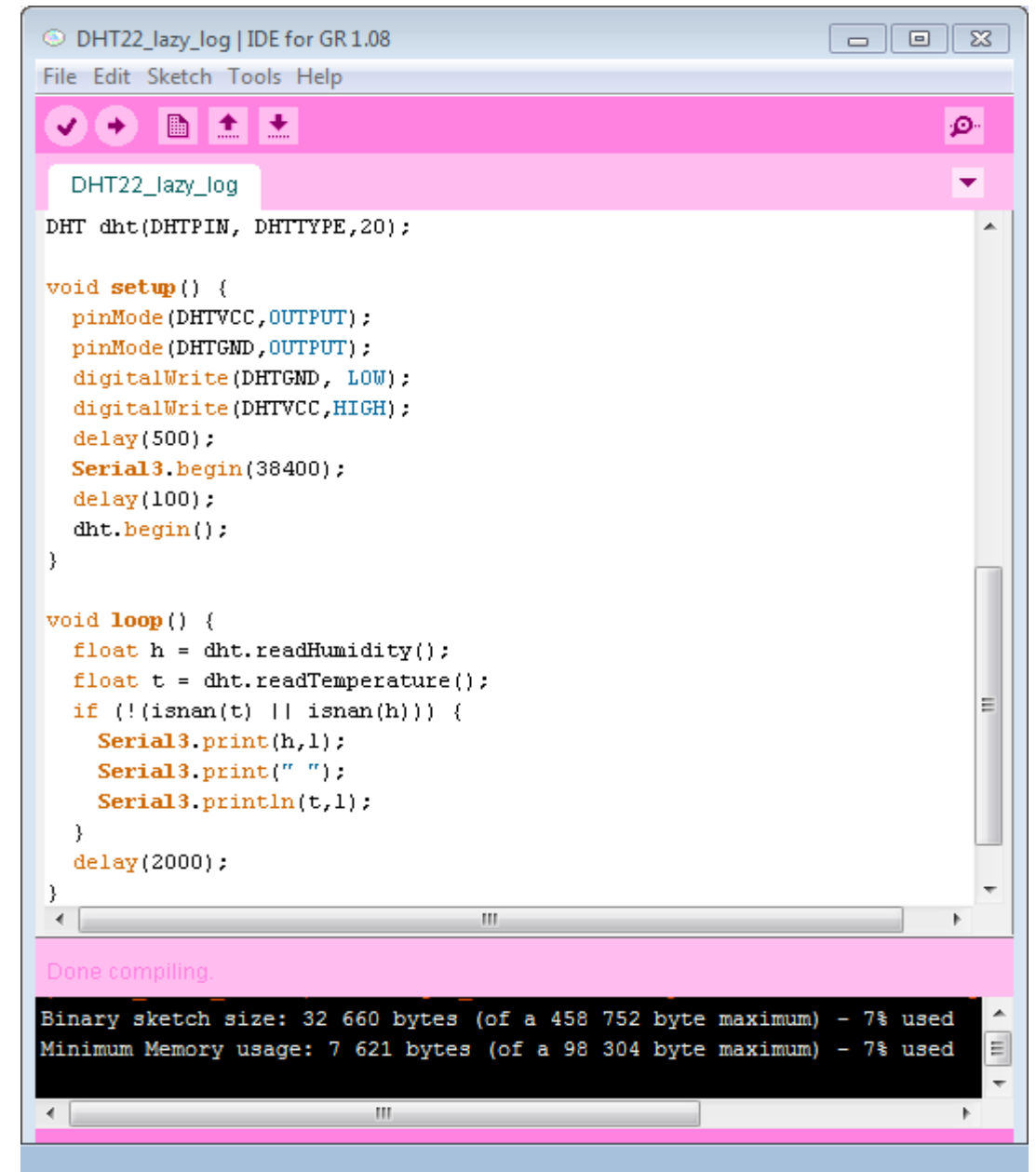
void loop() {
  float h = dht.readHumidity();
  float t = dht.readTemperature();
  if (!(isnan(t) || isnan(h))) {
    Serial3.print(h,1);
    Serial3.print(" ");
    Serial3.println(t,1);
  }
  delay(2000);
}
```



**GR-Sakura:** Arduino kompatibilis,  
nagy teljesítményű kártya  
CPU: RX63N 32 bites  
RAM: 128K, ROM: 1 M  
USB device/host  
Ethernet 100/10 Mbps  
Micro SD slot  
Xbee ready

# IDE for GR 1.0.8

- A fejlesztői rendszer az Arduino IDE-re hasonlító **IDE for GR** volt, amely offline használható és innen tölthető le: [IDE for Gadget Renesas](#)
- A másik lehetőség a **GR-Sakura Web Compiler** online fejlesztői rendszerének a használata, amelyhez itt találunk egy rövid leírást: [GR-SAKURA Special Project: Sketch on Web Compiler](#)
- A programletöltés USB-n keresztül történik: A RESET gomb megnyomásakor a kártya USB flash drive-ként látható a számítógép felől, melybe egyszerűen belemásoljuk a lefordított programot
- A kártya tápellátásához 5 V-ot kell biztosítani power bankból vagy 5V-os dugasztápból



```
DHT22_lazy_log | IDE for GR1.08
File Edit Sketch Tools Help
DHT22_lazy_log
DHT dht(DHTPIN, DHTTYPE,20);

void setup() {
  pinMode(DHTVCC,OUTPUT);
  pinMode(DHTGND,OUTPUT);
  digitalWrite(DHTGND,LOW);
  digitalWrite(DHTVCC,HIGH);
  delay(500);
  Serial3.begin(38400);
  delay(100);
  dht.begin();
}

void loop() {
  float h = dht.readHumidity();
  float t = dht.readTemperature();
  if (!(isnan(t) || isnan(h))) {
    Serial3.print(h,1);
    Serial3.print(" ");
    Serial3.println(t,1);
  }
  delay(2000);
}
```

Done compiling.

Binary sketch size: 32 660 bytes (of a 458 752 byte maximum) - 7% used  
Minimum Memory usage: 7 621 bytes (of a 98 304 byte maximum) - 7% used



# A vezeték nélküli kapcsolat ellenőrzése

- A számítógéphez kapcsolódó **COORDINATOR1** modul és az **XCTU** program segítségével ellenőrizzük, hogy a **DHT-22** szenzorral mért adatok megérkeznek-e
- Az alábbi ábrán közelítőleg 34 % relatív páratartalom és 28.4 °C hőmérséklet értékek érkezése látható

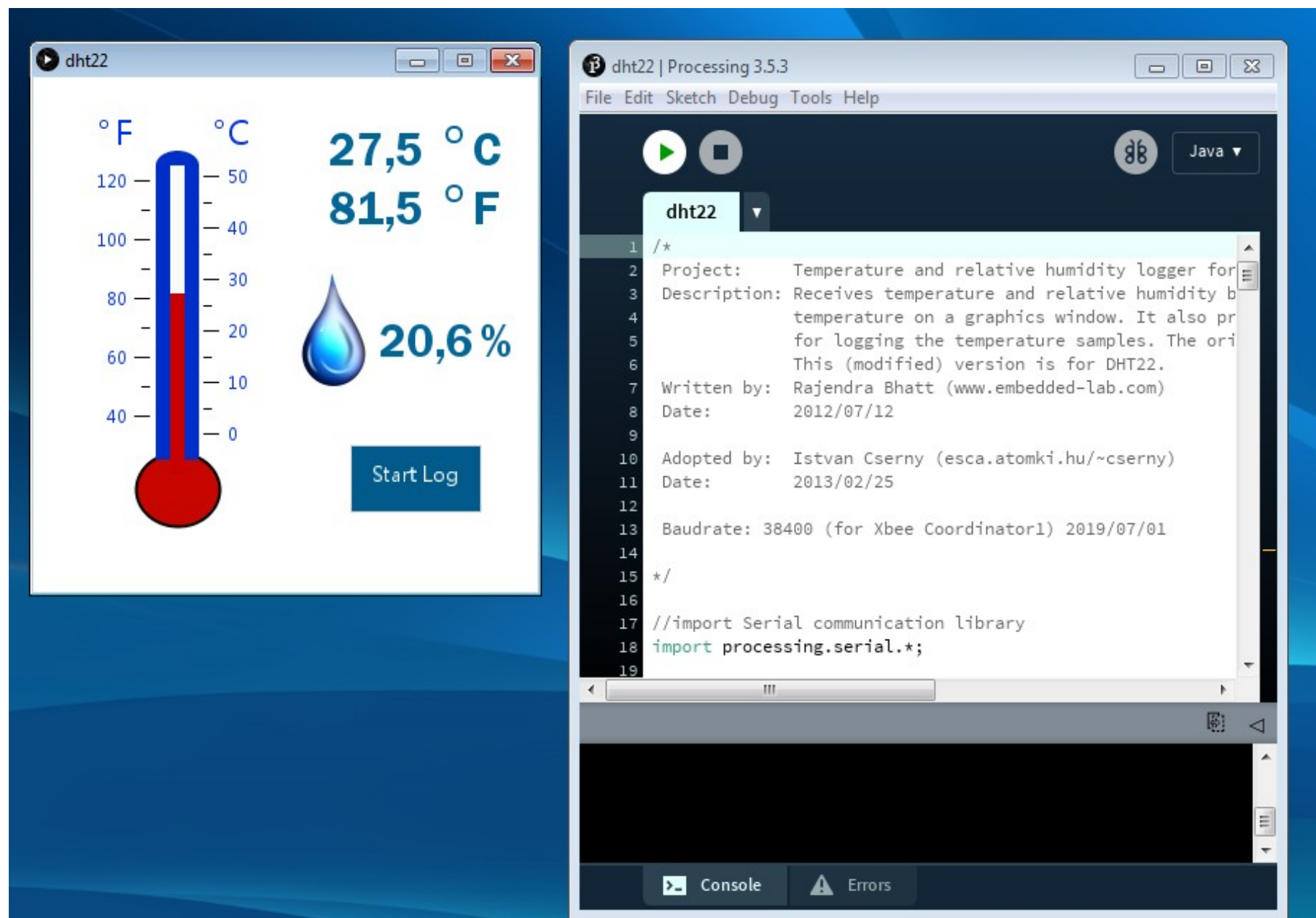
The screenshot displays the XCTU software interface. The main window title is 'XCTU' and the menu bar includes 'XCTU', 'Working Modes', 'Tools', and 'Help'. The interface is divided into several sections:

- Radio Modules:** A list of connected modules. The selected module is 'COORDINATOR1' with the following details:
  - Name: COORDINATOR1
  - Function: ZigBee Coordinator AT
  - Port: COM34 - 38400/8/N/1/H - AT
  - MAC: 0013A20040C29DBF
- Module Controls:** Includes 'Close', 'Record', and 'Detach' buttons. There are also status indicators for CTS, CD, DSR, DTR, RTS, and BRK.
- Console log:** A text area showing the received data. The data is presented in two columns:

34.0	28.4	33	34	2E	30	20	32	38	2E	34	0D	0A
34.1	28.4	33	34	2E	31	20	32	38	2E	34	0D	0A
34.1	28.4	33	34	2E	31	20	32	38	2E	34	0D	0A
34.2	28.4	33	34	2E	32	20	32	38	2E	34	0D	0A
34.1	28.3	33	34	2E	31	20	32	38	2E	33	0D	0A
34.1	28.3	33	34	2E	31	20	32	38	2E	33	0D	0A
34.2	28.4	33	34	2E	32	20	32	38	2E	34	0D	0A
- Statistics:** Shows 'Tx Bytes: 0' and 'Rx Bytes: 77'.

# Grafikus megjelenítés Processing segítségével

- A program eredeti változatát **Rajendra Bhatt** dolgozta ki DHT11-hez
- A DHT-22 miatt az input kezelést és a páratartalom kiírását módosítani kellett
- A **StartLog** gombra kattintva naplózás is kérhető
- Az alábbi sor hatására automatikusan a legkisebb sorszámú soros portot nyitja meg: `Dev_Board = new Serial(this, Serial.list()[0], 38400);`



# XBee modulok API módban

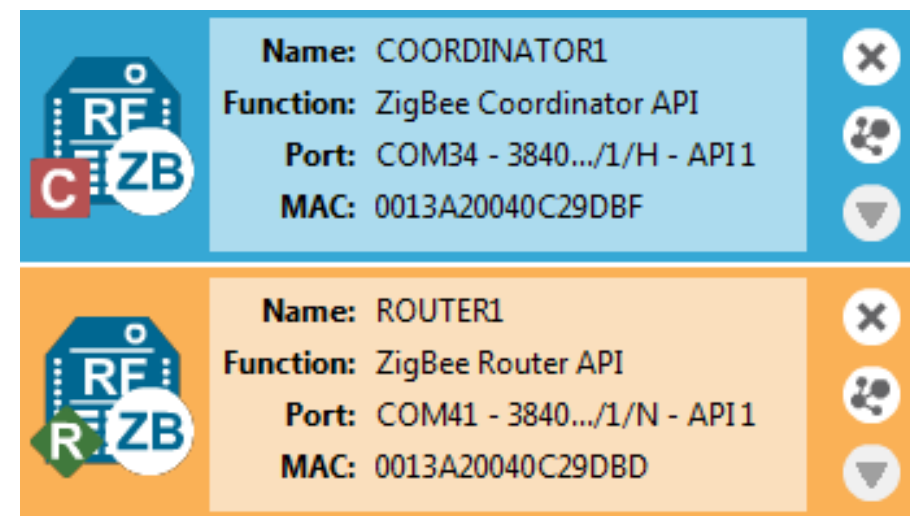
- Az **AT** módban működő modulokkal csak pont – pont kapcsolatot építhetünk ki
- Hálózat kialakítást vagy az **XBee** modulok további lehetőségeinek kiaknázását csak az úgynevezett **API módban** tudjuk megvalósítani. Ehhez a moduljaink firmware-ét le kell cserélni az **XCTU** alkalmazás **Update** funkciójával: ZigBee Coordinator API, ZigBee Router API, illetve ZigBee End Device API firmware-t kell választani

The screenshot displays the XCTU software interface. On the left, the 'Radio Modules' list shows two modules: 'COORDINATOR1' (ZigBee Coordinator AT) and 'ROUTER1' (ZigBee Router AT). The main area shows the 'Radio Configuration' for 'COORDINATOR1'. The 'Update' button in the configuration toolbar is circled in red. An 'Update firmware' dialog box is open, showing a table for selecting the product family, function set, and firmware version.

Product family	Function set	Firmware version
XBP24-B	ZigBee Coordinator AT	23A7 (Newest)
XBP24-SE	ZigBee End Device API	23A0
XBP24-ZB	ZigBee End Device AT	238C
	ZigBee End Device Analog IO	2370
	ZigBee End Device Digital IO	2364
	ZigBee End Device PH	2341
	ZigBee Router API	2321

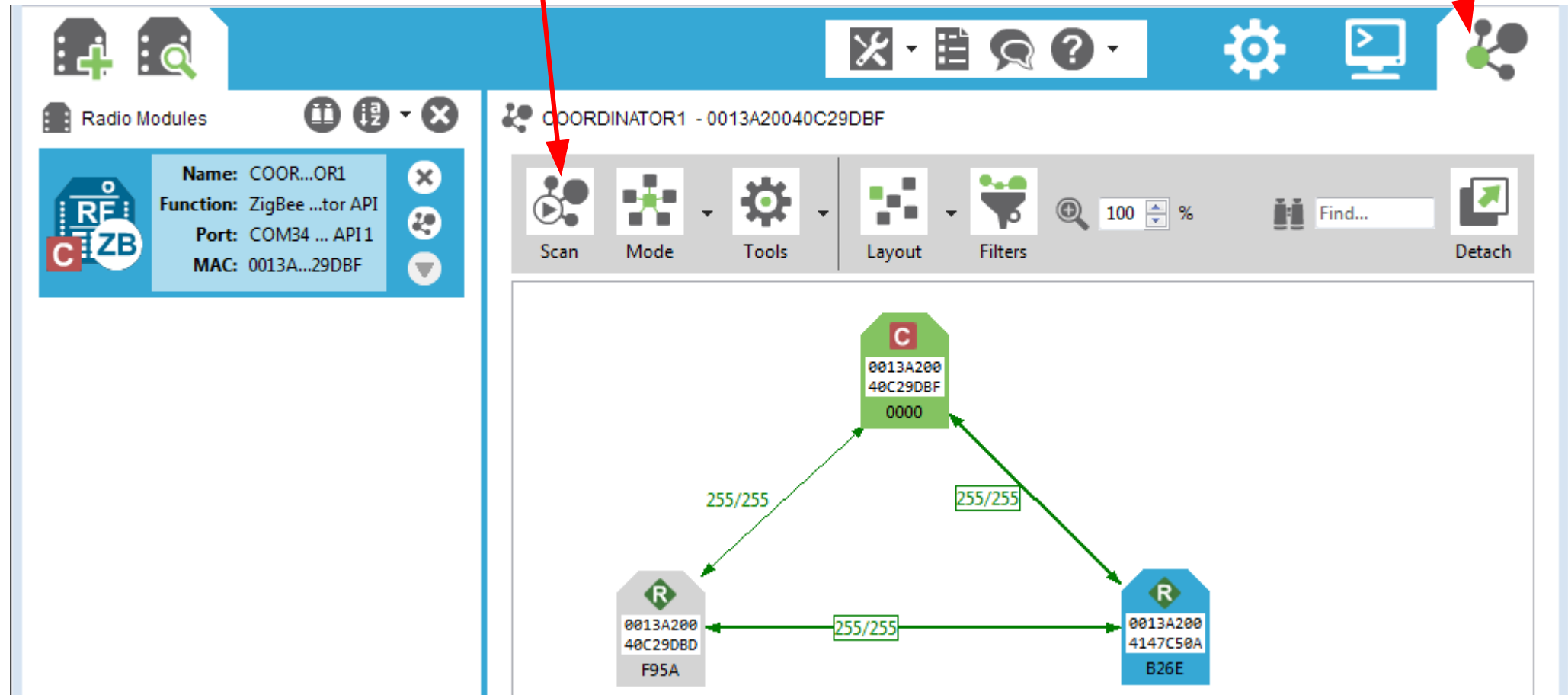
# Modulok konfigurálása API módban

- API módban a **DL**, **DH** paramétereknek nincs szerepe, a címzés az üzenetsomagok része
- Bár nem kötelező, de célszerű a hálózat azonosítóját beállítani (**PAN ID**). A modulok enélkül is tudnak csatlakozni, de fölöttebb kínos lenne, ha egy eszköz a szomszéd lakás hálózatára kapcsolódna a mi koordinátorunk helyett
- Router eszközöknél a **JV** és **JN** paramétereket állítsuk Enabled (1) módba!
- Mindegyik modulnál állítsuk be egységesen az **AP** paramétert (alapértelmezetten 1)
  - ❖ 1: API mode enabled
  - ❖ 2: API mode enabled with escaping
- Állítsuk be a soros port sebességét (esetünkben 38 400 baud)!
- Ha végeztünk minden modul konfigurálásával, akkor moduljaink készen állnak az **API** módú használatra



# A hálózat felépülése és felderítése

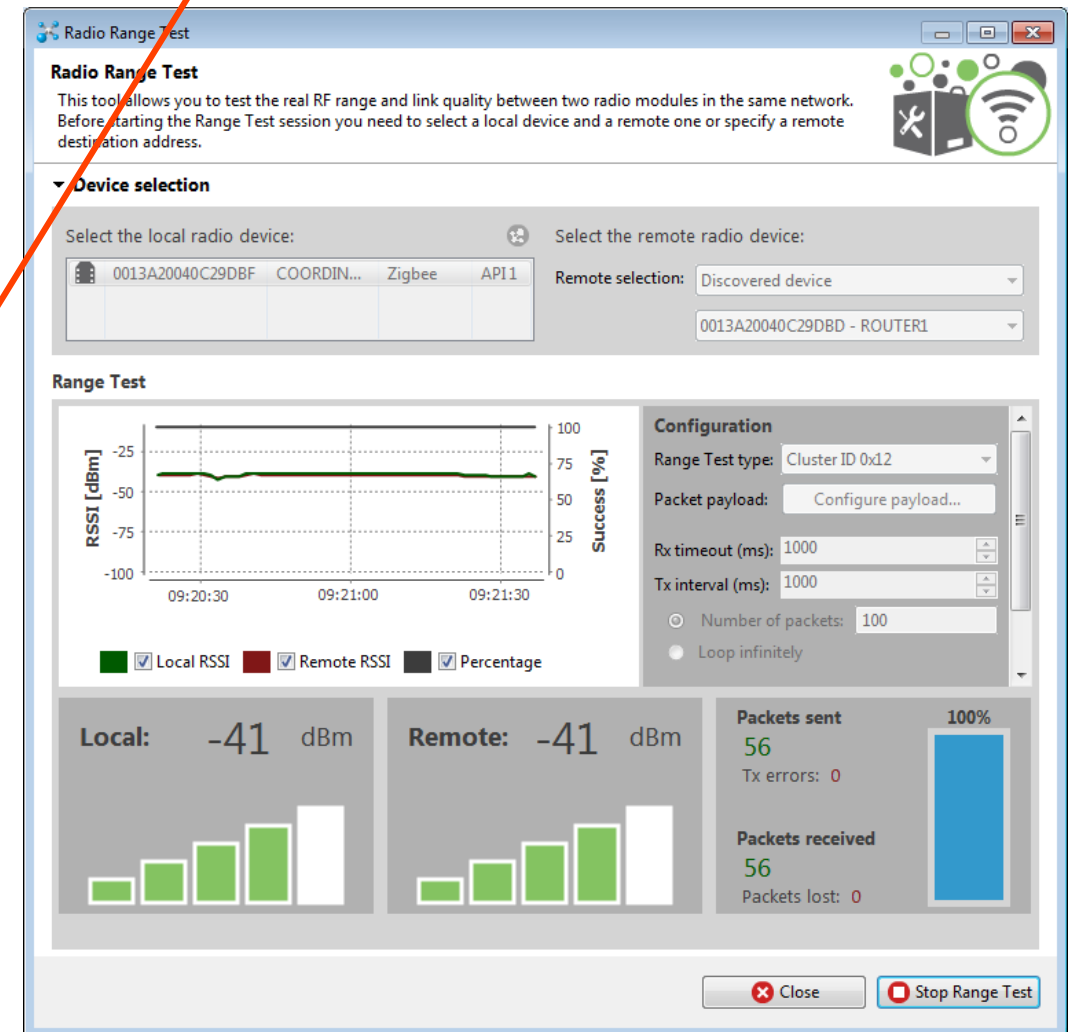
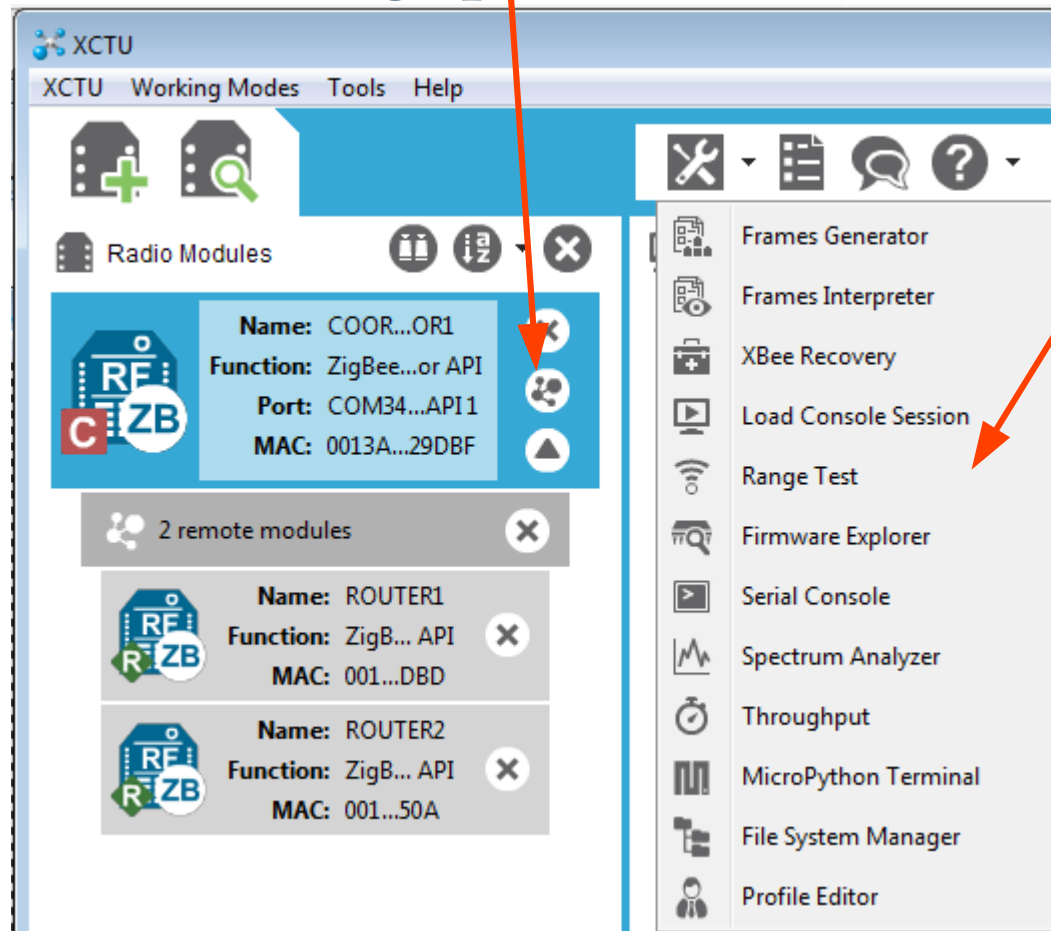
- Az **XCTU** programban csak a **COORDINATOR1** modult csatlakoztatjuk, s a **Network mode** lapot kiválasztva, azon belül kattintsunk a **Scan** gombra!
- A hálózat grafikus elrendezését az egérrel, illetve a **Layout** választógombbal módosíthatjuk
- A koordinátor mindig 0000 címet kap
- Esetünkben a két router egymással is kommunikál
- Egyikük **B26E**, a másik **F55A** ID-t kapott a hálózat felépülése során



# Teszt funkciók az XCTU programban

- A COORDINATOR1 modul „Discover radio nodes in the same network” gombjára kattintva derítsük fel az elérhető modulokat, s ezután használhatjuk a **Tools** menüben található tesztek (Range test, Throughput test)

- A jobboldali ábrán egy **Range test** eredménye látható, nulla csomagvesztéssel

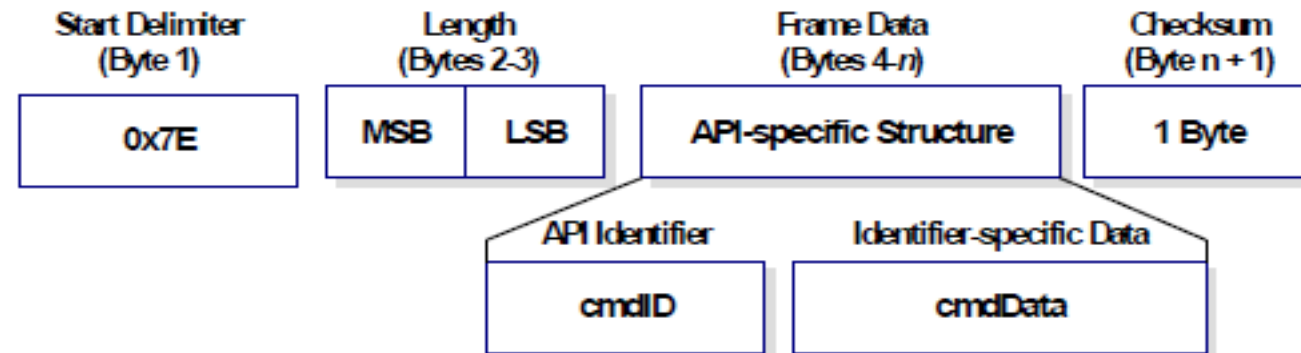


# Hogyan zajlik a kommunikáció API módban?

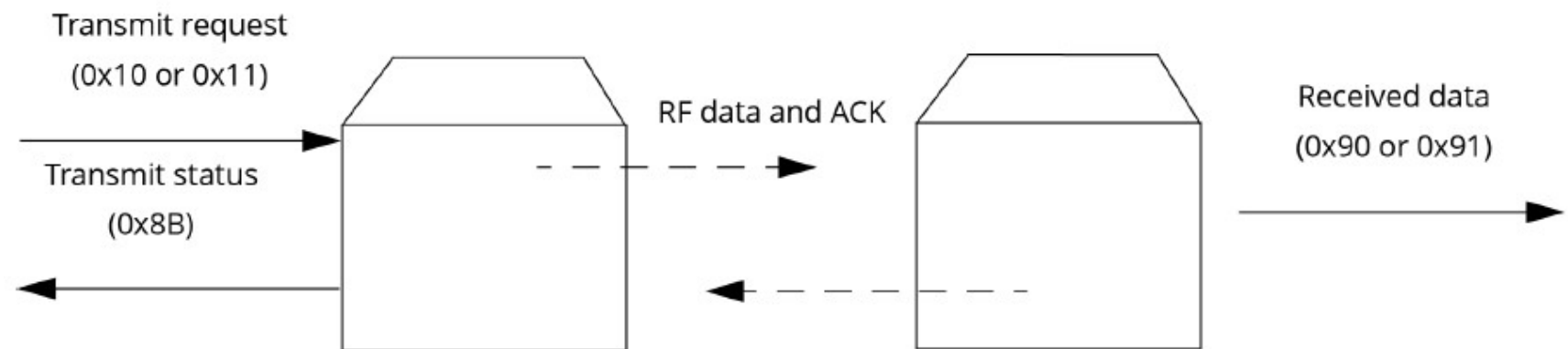
- API módban az üzenetek keretezett csomagokban továbbíthatók, például API 1 módban:



- Az adatkeret is stukturált, felépítése az üzenet típusától függ:



- Adatküldés RF kapcsolaton keresztül



# API kódok és csomagformátumok

Remote AT Command Request (API 1)

7E 00 0F 17 01 00 00 00 00 00 00 FF FF FF FE 02 4E 49 53

Start delimiter: 7E

Length: 00 0F (15)

Frame type: 17 (Remote AT Command Request)

Frame ID: 01 (1)

64-bit dest. address: 00 00 00 00 00 00 FF FF

16-bit dest. address: FF FE

Command options: 02

AT Command: 4E 49 (NI)

Checksum: 53

A COORDINATOR1-ből  
küldött broadcast ATNI  
parancs lekéri a modulok  
nevét és címét

Remote Command Response (API 1)

7E 00 16 97 01 00 13 A2 00 41 47 C5 0A B2 6E 4E 49 00 52 4F 55 54 45 52 32 91

Start delimiter: 7E

Length: 00 16 (22)

Frame type: 97 (Remote Command Response)

Frame ID: 01 (1)

64-bit source address: 00 13 A2 00 41 47 C5 0A

16-bit source address: B2 6E

AT Command: 4E 49 (NI)

Status: 00 (Status OK)

Response: 52 4F 55 54 45 52 32 (ROUTER2)

Checksum: 91

ROUTER2 válasza

Remote Command Response (API 1)

7E 00 16 97 01 00 13 A2 00 40 C2 9D BD F9 5A 4E 49 00 52 4F 55 54 45 52 31 5A

Start delimiter: 7E

Length: 00 16 (22)

Frame type: 97 (Remote Command Response)

Frame ID: 01 (1)

64-bit source address: 00 13 A2 00 40 C2 9D BD

16-bit source address: F9 5A

AT Command: 4E 49 (NI)

Status: 00 (Status OK)

Response: 52 4F 55 54 45 52 31 (ROUTER1)

Checksum: 5A

ROUTER1 válasza

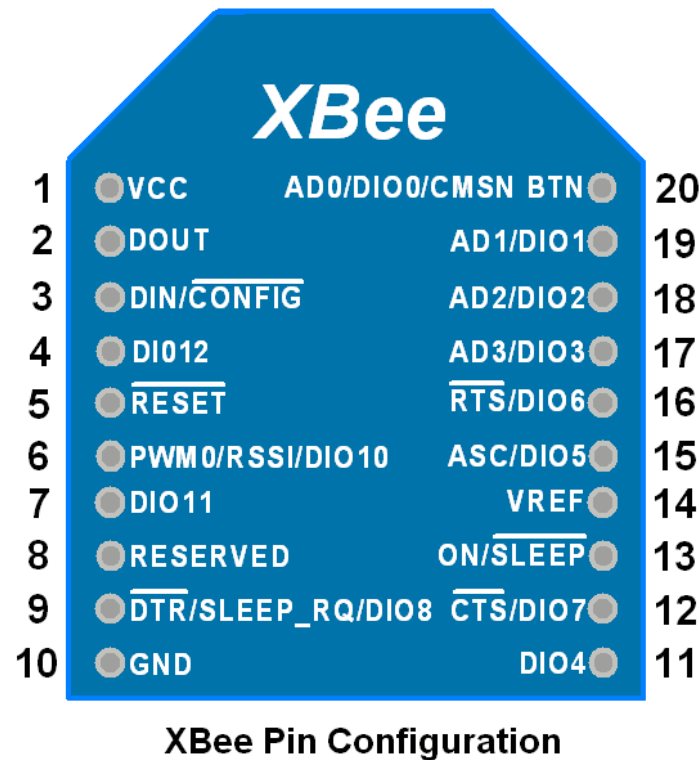
API Frame Names	API ID
AT Command	0x08
AT Command - Queue Parameter Value	0x09
Zigbee Transmit Request	0x10
Explicit Addressing Zigbee Command Frame	0x11
Remote Command Request	0x17
Create Source Route	0x21
AT Command Response	0x88
Modem Status	0x8A
Zigbee Transmit Status	0x8B
Zigbee Receive Packet (AO=0)	0x90
Zigbee Explicit Rx Indicator (AO=1)	0x91
Zigbee IO Data Sample Rx Indicator	0x92
XBee Sensor Read Indicator (AO=0)	0x94
Node Identification Indicator (AO=0)	0x95
Remote Command Response	0x97
Over-the-Air Firmware Update Status	0xA0
Route Record Indicator	0xA1
Many-to-One Route Request Indicator	0xA3



# Használjuk az Xbee modulok I/O funkcióit is!

- API módban az Xbee modulok szabad kivezetéseit felhasználhatjuk analóg jelek mérésére, vagy digitális I/O műveletekre
- Az alábbiakban a ROUTER2 modult konfiguráltuk úgy, hogy  $0x2000$  ( $8192_{10}$ ) milliszekundumonként jelentse az AD1 analóg, és a DIO4 digitális bemenetek állapotát, s jelentse a tápfeszültség értékét is, ha kisebb, mint a megadott határérték
- Az időzített adatküldés a DH, DL párossal megadott címre történik, vagy 0 cím esetén a koordinátor modul kapja meg

Az Xbee modul tehát önmagában is felhasználható!



**I/O Settings**  
Modify DIO and ADC options

D0 AD0/DIO0 Configuration	Commissioning Button [1]	
D1 AD1/DIO1 Configuration	ADC [2]	
D2 AD2/DIO2 Configuration	Disabled [0]	
D3 AD3/DIO3 Configuration	Disabled [0]	
D4 DIO4 Configuration	Digital Input [3]	
D5 DIO5/Assoc Configuration	Associated indicator [1]	
P0 DIO10 Configuration	RSSI PWM Output [1]	
P1 DIO11 Configuration	Disabled [0]	
P2 DIO12 Configuration	Disabled [0]	
PR Pull-up Resistor Enable	1FFF	
LT Associate LED Blink Time	0	x10 ms
RP RSSI PWM Timer	28	x100 ms
DO Device Options	1	Bitfield

**I/O Sampling**  
Configure IO sampling parameters

IR IO Sampling Rate	2000	x1 ms
IC Digital IO Change Detection	0	
V+ Supply Voltage Threshold	B55	

# Használjuk az Xbee modulok I/O funkcióit is!

- Csatlakoztassuk a **COORDINATOR1** modult az **XCTU** program **Console** módjában, akkor a terminál ablakban láthatjuk a beérkező üzeneteket
- Az üzenetek értelmezéséhez vegyük igénybe az **XCTU** segítségét!
- Az alábbiakban bemutatjuk egy üzenetsomag elemzését, melyet a **Tools** menü **Frame Interpreter** eszköze állította elő a hexadecimálisan megadott bájt sorozatból
- A forráscím alapján ez az üzenet a **ROUTER2** modul jelentése

The screenshot shows the XCTU software interface. At the top, it displays 'COORDINATOR1 - 0013A20040C29DBF' and 'ROUTER1 - 0013A20040C29DBD'. Below this are control buttons for 'Close', 'Record', and 'Detach', along with status indicators for 'CTS', 'CD', 'DSR', 'DTR', 'RTS', and 'BRK'. On the right, it shows 'Tx frames: 0' and 'Rx frames: 6'. The main area is divided into 'Frames log' and 'Frame details'. The 'Frames log' table shows several received frames, all identified as 'IO Data Sample RX Indicator'. The 'Frame details' panel shows the following values: DIO4/AD4 digital value: High; DIO1/AD1 analog value: 02 0E (526); Power supply value: 0A D4; Checksum.

```
IO Data Sample RX Indicator (API 1)
7E 00 16 92 00 13 A2 00 41 47 C5 0A B2 6E 01 01 00 10 82 00 10 02 0F 0A D4 AE
Start delimiter: 7E
Length: 00 16 (22)
Frame type: 92 (IO Data Sample RX Indicator)
64-bit source address: 00 13 A2 00 41 47 C5 0A
16-bit source address: B2 6E
Receive options: 01
Number of samples: 01
Digital channel mask: 00 10
Analog channel mask: 82
DIO4/AD4 digital value: High
DIO1/AD1 analog value: 02 0F (527)
Power supply value: 0A D4
Checksum: AE
```

} DIO4 = High  
AD1 = 527  
Vcc = 0xAD4 = 3248 mV

# Merre tovább?

---

- A továbblépéshez programkönyvtárakat kell beszerezni (vagy meg kell írni) az **Xbee/ZigBee API** felület kezeléséhez
- Programkönyvtárak
  - ❖ [XBee library for Digi XBee ZB Modules in API Operation mode](#) (STM32, mbed)
  - ❖ [DIGI Xbee ANSI C library](#)
  - ❖ [DIGI Xbee Java library](#)
  - ❖ [Xbee Arduino library](#)
- Hasznos könyvek
  - ❖ Matthijs Kooijman: [Building Wireless Sensor Networks Using Arduino](#)
  - ❖ Robert Faludi: [Building Wireless Sensor Networks: With Zigbee, Xbee, Arduino, And Processing](#)

# THE GENERIC STM32F103 PINOUT DIAGRAM

## LEGEND

POWER
GROUND
PHYSICAL PIN
PIN NAME
CONTROL
ANALOG
TIMER & CHANNEL
USART
SPI
I2C
CAN BUS
USB
MISC
BOARD HARDWARE

- 5V tolerant
- Not 5V tolerant
- ~ PWM pin
- ⋯ Alternate function
- ⚠ PC13,PC14,PC15: Sink max 3mA, source 0mA, max 2mhz, max 30pF

Absolute MAX 150mA total source/sink for entire CPU

Max ±20mA per pin, ±8mA recommended

