Ethernet alapú elosztott jelfeldolgozó rendszer kiépítése

BSc Önálló Laboratórium

Készítette: Dányi Péter

Konzulens: Dr. Orosz György

Méréstechnika és Információs Rendszerek Tanszék

2017.05.10.

Elosztott jelfeldolgozási rendszer

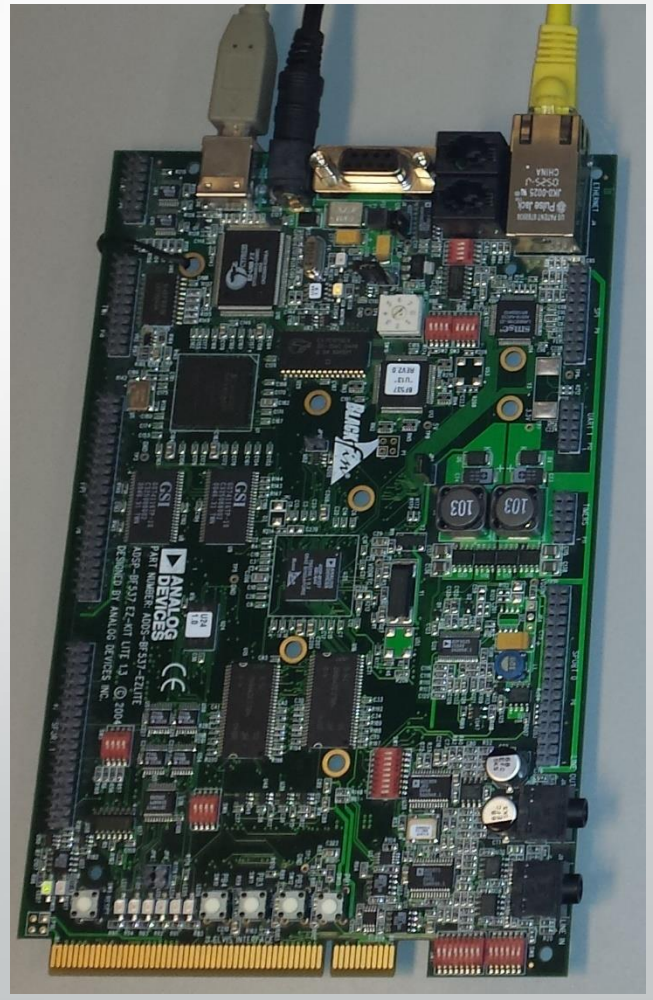
- Manapság igen elterjedt: pl.: IoT rendszerek, akusztikus lokalizáció
- Hálózat:
  - Vezetéknélküli kapcsolat (Bluetooth, ZigBee)
  - Vezetékes kapcsolat (Ethernet, CAN)
- Részegységei:
  - Szenzorok (akár jelfeldolgozó funkcióval)
  - Jelfeldolgozó – DSP, FPGA
  - Hálózatkezelő – szerver, kliens, router
- Mire jó?
  - Több térben elosztott szenzor
  - Az adatok grafikus megjelenítése szerveren

Elosztott jelfeldolgozási rendszer

- **OSI modell –** számítógépes hálózatok réteges szerkezetűek
- **Ethernet** : a hálózat fizikai és adatkapcsolati rétegét valósítja meg
  - Felsőbb rétegek megvalósítását nem célozza (csak hardver)
- **TCP/IP**
  - Megbízható, sorrendhelyes, kapcsolatalapú adatátvitel
  - Nagy erőforrás igény → implementációs probléma kisebb rendszereken
- **LwIP**
  - Csökkentett erőforrásigény, bizonyos funkciók elhagyásával
  - Nyílt forráskodú
  - Beágyazott rendszerekben

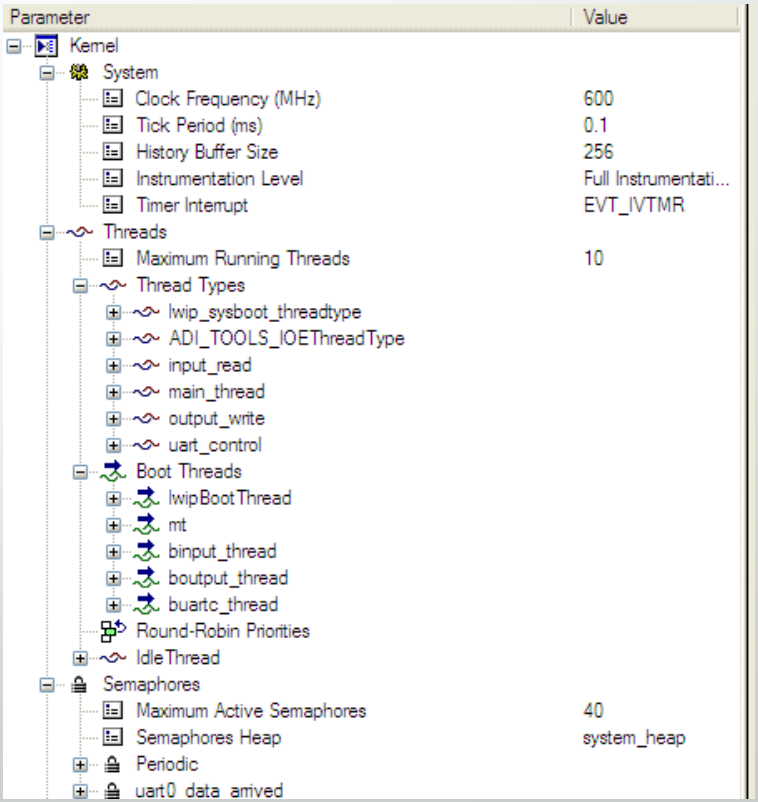
ADSP – BP537 (Blackfin)

- Jelfeldolgozó processzor
  - **600MHz** CLK
  - 96kHz ADC és DAC - **ΣΔ átalakítók**
- Kommunikációs interfészek:
  - **Ethernet,** CAN, SPI, I2C, **UART - debug**

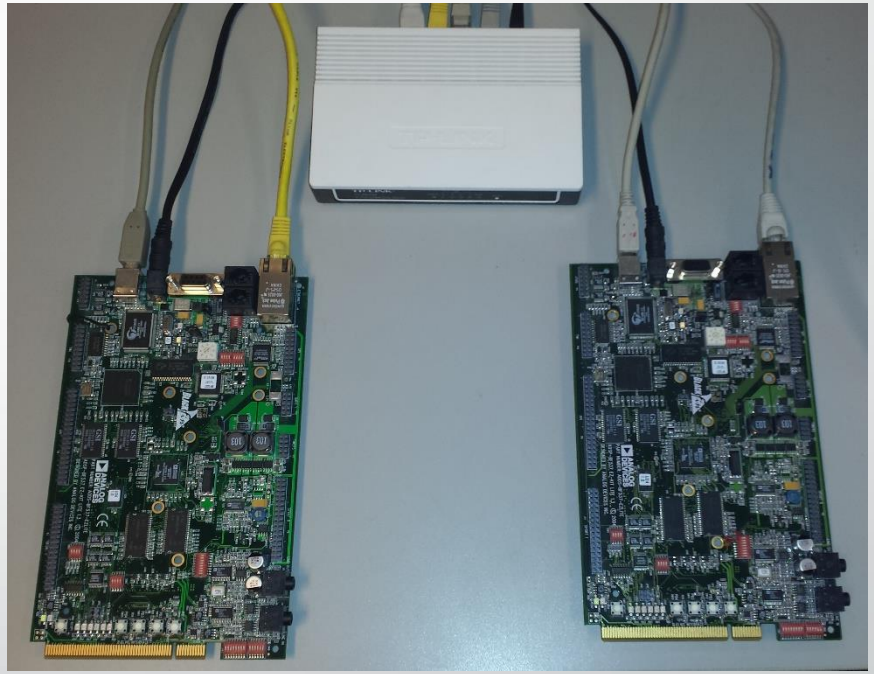


Fejlesztőkörnyezet: Visual DSP++

- VDK operációs rendszer
  - Szálkezelés
  - Driverek az ADC és DAC-hoz
- LwIP támogatás



A megvalósított rendszer

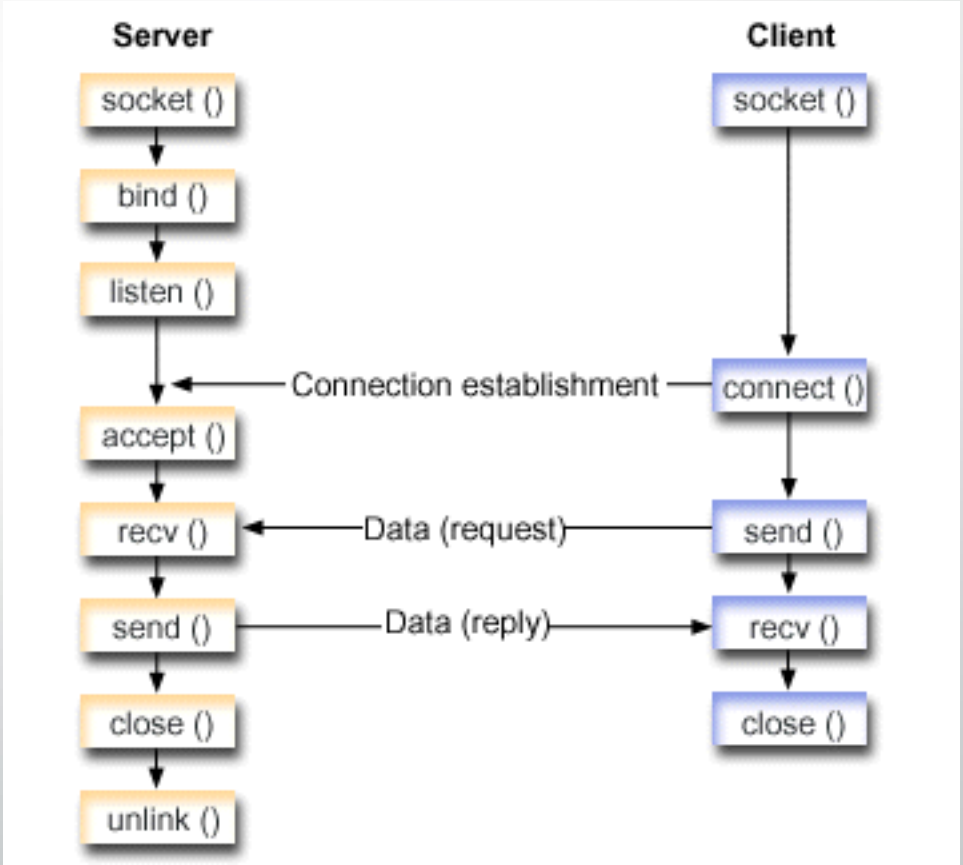


- A két kártya Etherneten kommunikál
- Routeren keresztül
  - MAC-hez IP címet rendel
- TCP kapcsolat:
  - Szerver – kliens
  - A kliens üzeneteket küld a szervernek
- A számítógéppel mindkét DSP össze van kötve
  - Debug és Compile

A kapcsolat felépítése

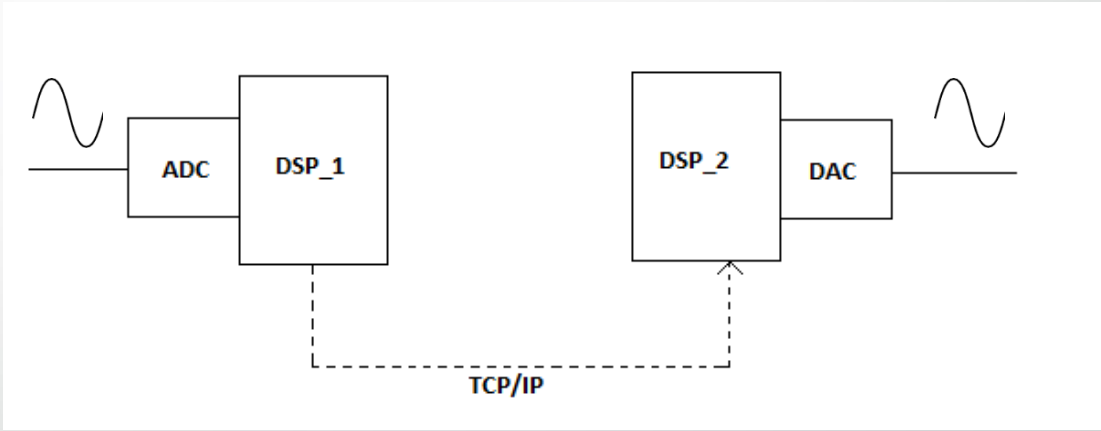
- Socket létrehozása: socket()
- bind() - cím struktúra hozzárendelése
  - Ahol a szerver várja a kapcsolódásokat
- listen()
  - A szerver – socket mód, „hallgatózás”
- connect()
  - A kliens megpróbál csatlakozni
- accept()
  - A szerver elfogadja és hozzárendel egy kliens socketet
- send() ←→ recv()
  - Adatok küldése és fogadása mindkét oldalon

Kapcsolat lezárása: close()



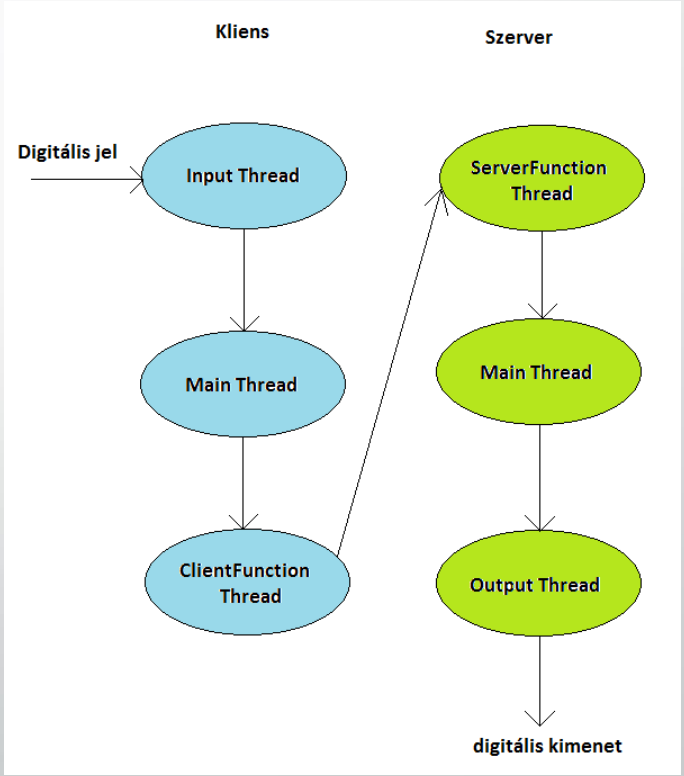
Működő rendszer 1.

- Kliens DSP
  - ADC-n mintavételezi a bejövő jelet
  - Küldi tovább a szervernek
- Szerver DSP
  - Csomagok fogadása
  - Kimeneti jel előállítása
- Ebben az esetben mindkét kártyán történhet jelfeldolgozás

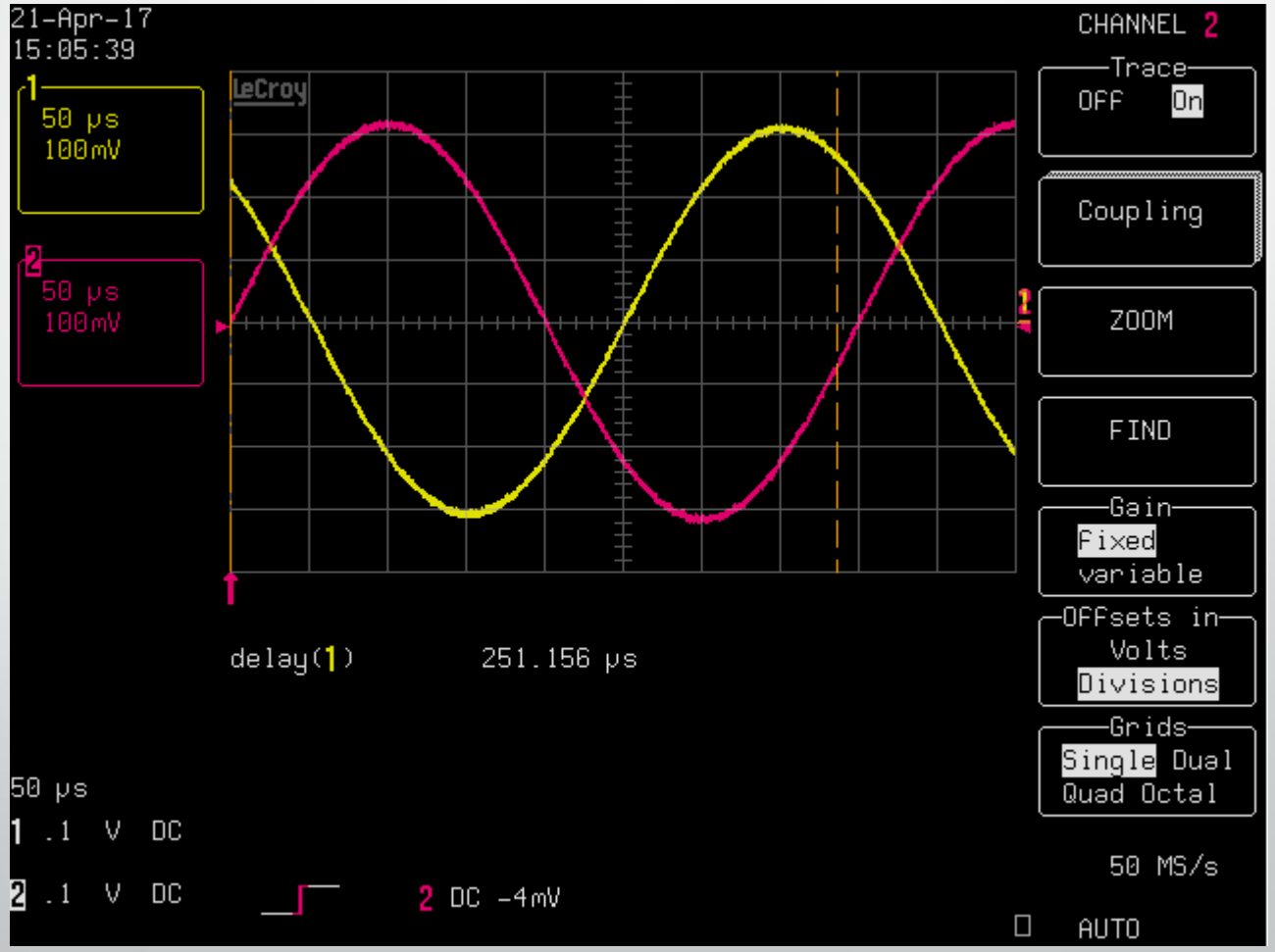


Működő rendszer 2.

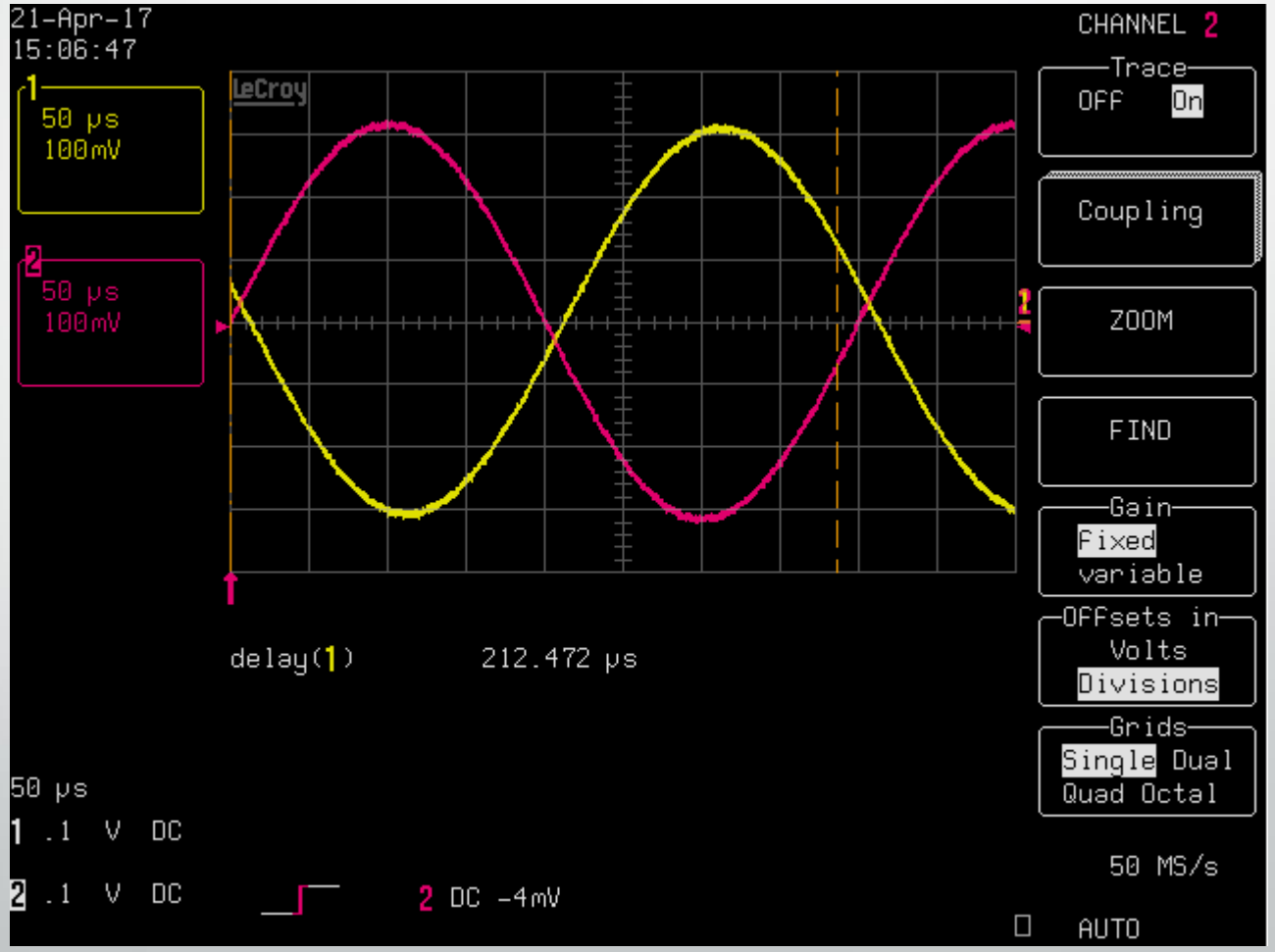
- Kártyánként 3 szál:
- Kliens
  - Input (adatok beolvasása)
  - Main (feldolgozás)
  - ClientFunction (küldés)
- Szerver
  - ServerFunction (fogadás)
  - Main (feldolgozás)
  - Output (kimenet írása)



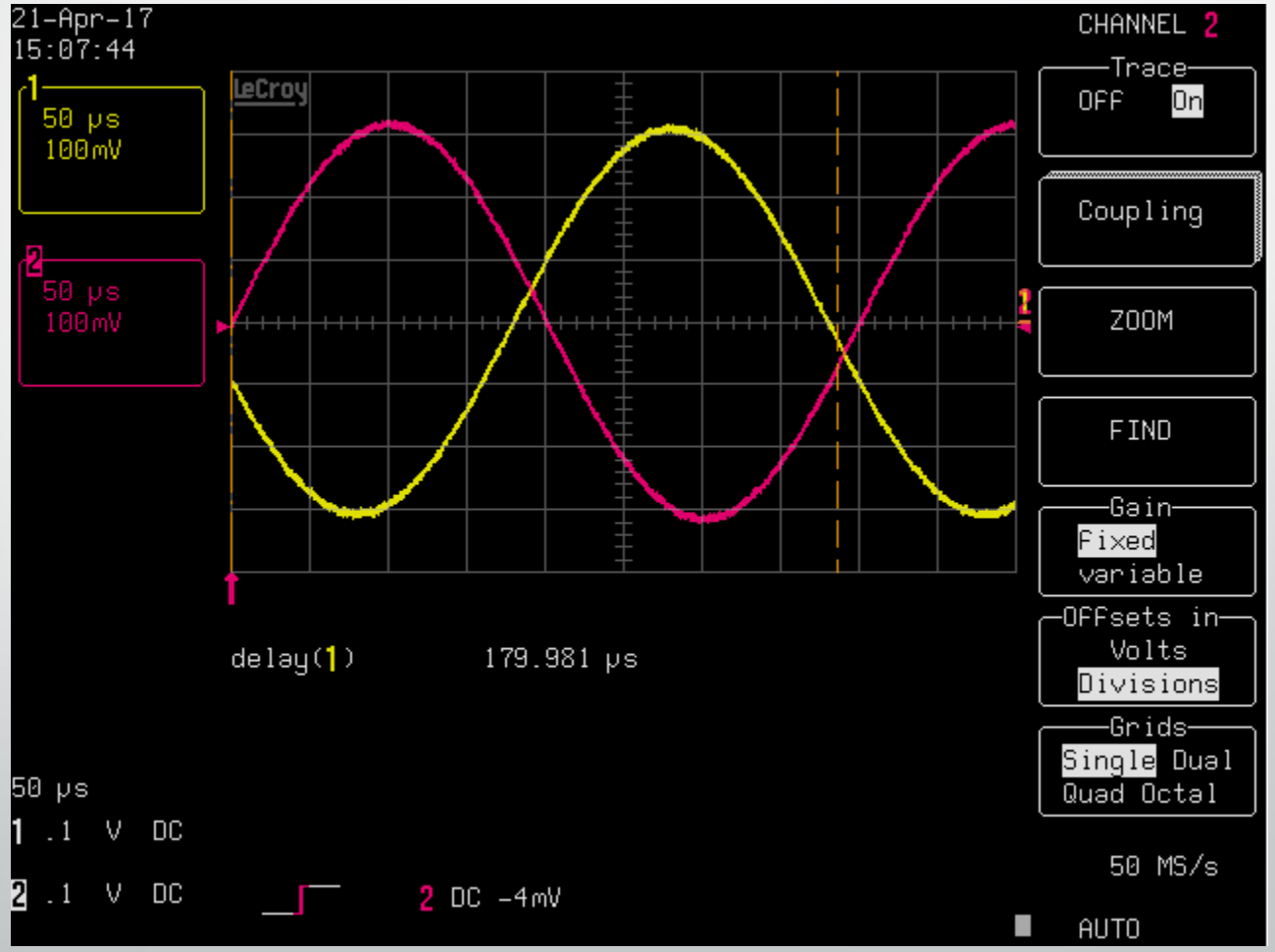
Mérés: Szinkronizáció 1.



Mérés: Szinkronizáció 2.



Mérés: Szinkronizáció 3.

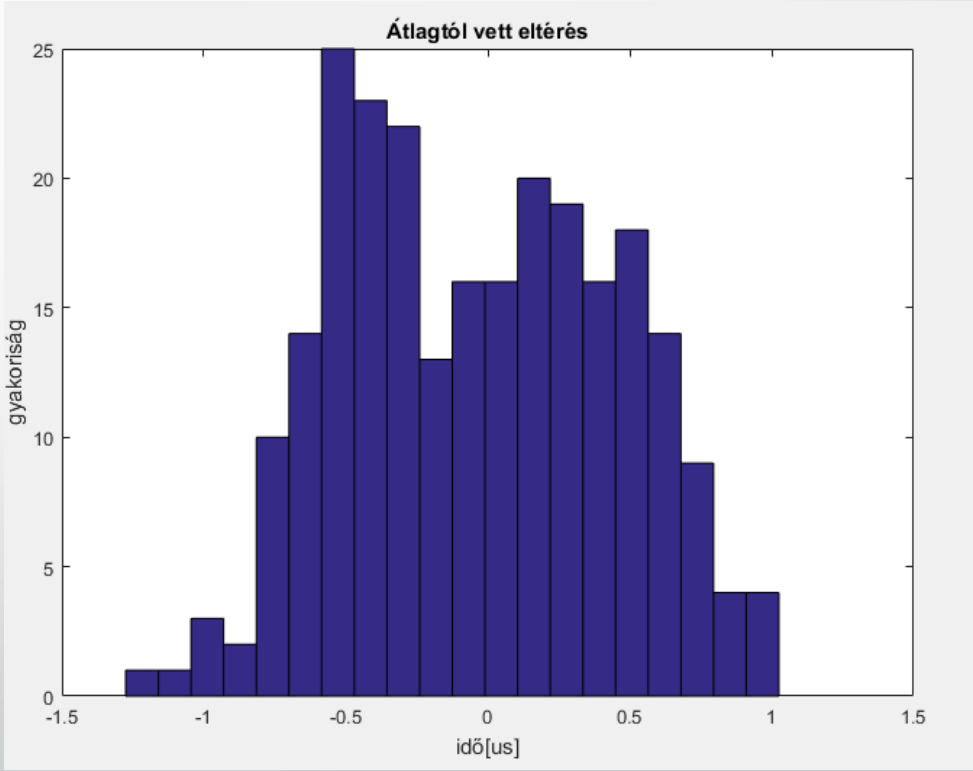


Eredmény

- A két jel egymáshoz képest elcsúszik fázisban
- Drift : Egységnyi sebességhez képesti eltérés
- $\varrho = \frac{\Delta\tau}{\Delta T} \rightarrow \varrho = 0{,}6\text{ppm}$
- Oka: oszcillátorok frekvenciájának eltérése
- Megoldás: pl.: minden csomagban elküldeni a küldés időpontját

Adatátviteli idő mérése

- Timer konfigurálása
- Két üzenet fogadás között eltelt idő: 10ms
- Adatátviteli sebesség: 96kbyte/sec



Összegzés

- A kialakított rendszer alkalmas valós idejű jelfeldolgozási feladatokra
- Elvégzendő feldatok:
  - A hálózat további paramétereinek kimérése
  - Néhány probléma még megoldásra vár: pl. szinkronizáció
  - Esetleg több kliens

Köszönöm a figyelmet!