



BUDAPESTI MŰSZAKI ÉS GAZDASÁGTUDOMÁNYI EGYETEM  
VILLAMOSMÉRNÖKI ÉS INFORMATIKAI KAR  
MÉRÉSTECHNIKA ÉS INFORMÁCIÓS RENDSZEREK TANSZÉK

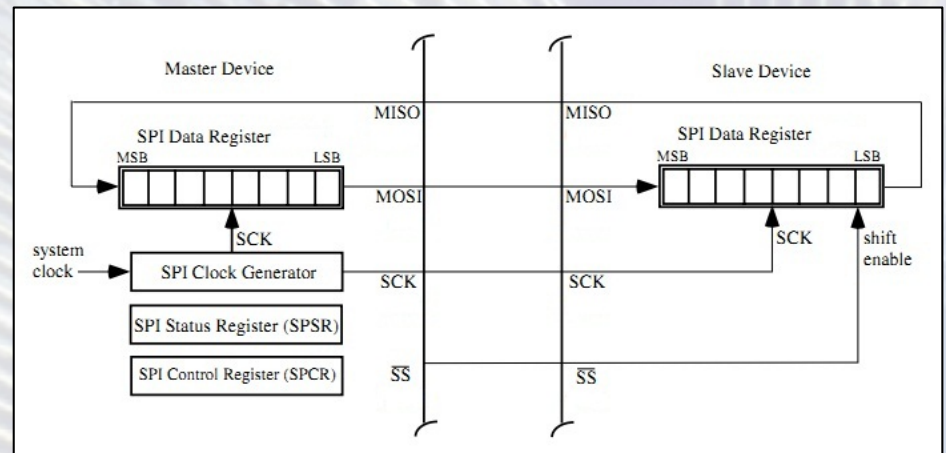
# Digitális technika (VIMIAA02)

## 11. laboratórium

Raikovich Tamás  
BME MIT

# Az SPI soros adatátvitel

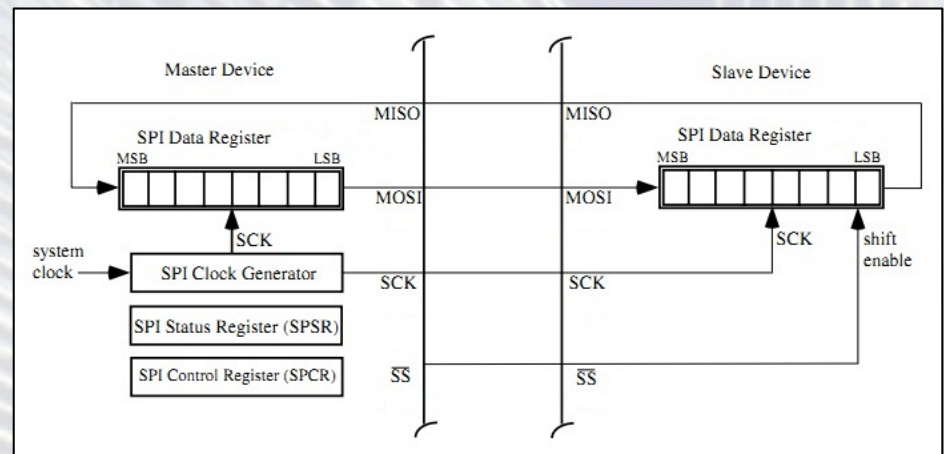
- Az SPI soros adatátviteli interfész elsősorban áramkörön belüli, áramkörök közötti adatátvitelre szolgál
- 4 vezetékes szinkron interfész
  - CSn, SSn: periféria kiválasztó jel
  - SCK: soros órajel
  - MOSI: master kimenet, slave bemenet
  - MISO: master bemenet, slave kimenet



# Az SPI soros adatátvitel

Az SPI soros adatátviteli interfész nagyon flexibilis, sok különböző megvalósítása létezik

- 4 féle üzemmód
- Tetszőleges számú bájt átvitele
- Egyirányú kapcsolat (pl. csak bemenet)
- Több bites kiterjesztés
- Busz topológia
- Láncba kapcsolás
- Nagy sebesség



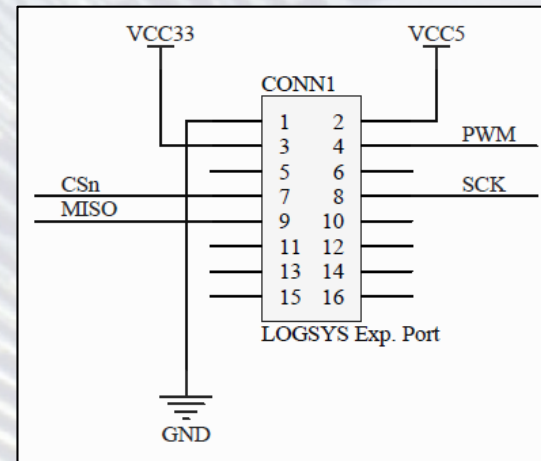
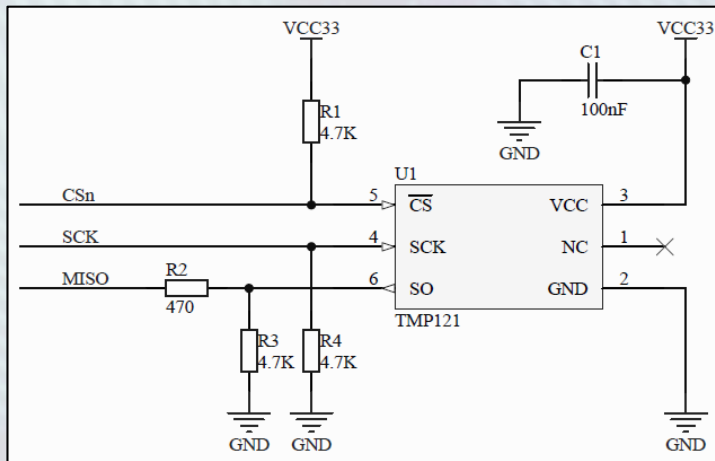


# Az SPI soros adatátvitel

- A MiniRISC processzor nem rendelkezik hardveres SPI perifériával
- Az SPI interfész a GPIO perifériával „bit-banging” módban realizálható
- Ez a megoldás rugalmas, tetszőleges előíráshoz könnyen adaptálható
  - Az elérhető sebesség konfiguráció és protokoll függő
- Az SPI interfész szoftveres használatát a TMP121 hőmérő illesztésén keresztül nézzük meg

# A TMP121 hőmérő modul

- A hőmérő modul kapcsolási rajza az alábbi ábrán látható
- Az „A” bővítőcsatlakozót használva a bekötés a következő
  - CSn → GPIOA[0], SCK → GPIOA[1], MISO → GPIOA[2]
  - Mi legyen a port bitek iránya és kezdeti értéke?
  - Készítsük el a GPIO A portot konfiguráló szubrutint

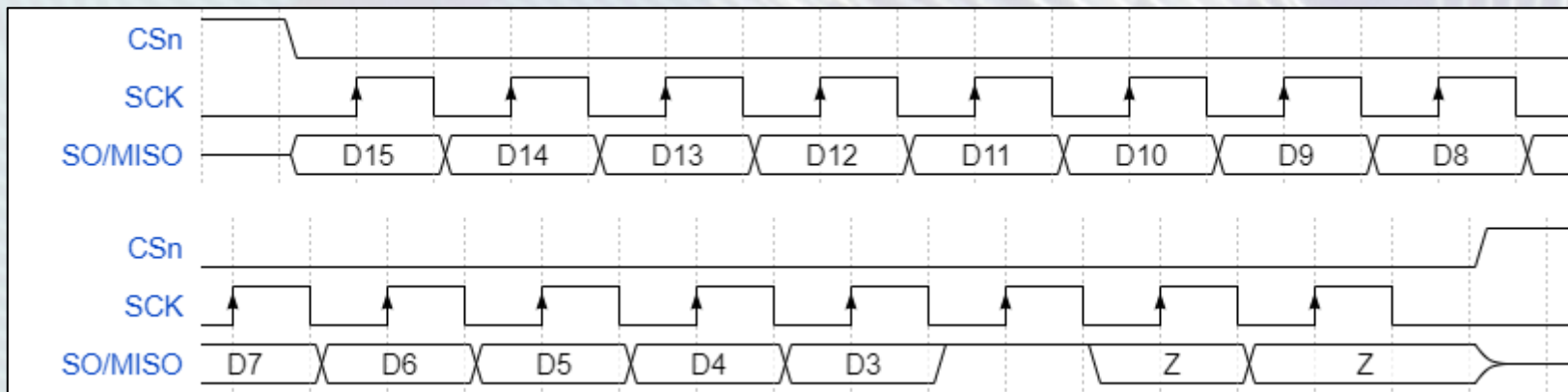


**A**

15 (I) C[3]	13 (I/O) A[6]	11 (I/O) A[4]	9 (I/O) A[2]	7 (I/O) A[0]	5 (I/O) C[1]	3 (PWR) +3,3 V	1 (PWR) GND
16 (I) C[4]	14 (I/O) A[7]	12 (I/O) A[5]	10 (I/O) A[3]	8 (I/O) A[1]	6 (I/O) C[2]	4 (I/O) C[0]	2 (PWR) +5 V

# A hőmérséklet beolvasása

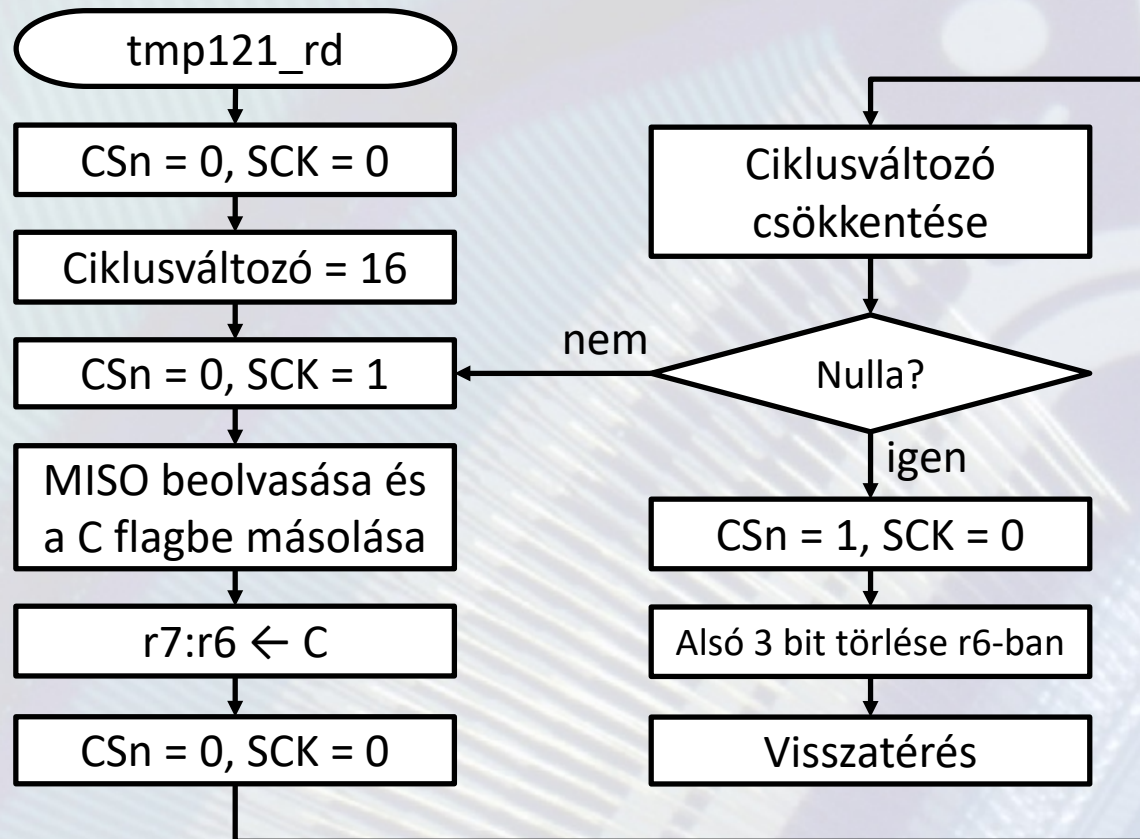
- A TMP121 SPI kommunikációjának idődiagramja és a gyakorlaton megbeszéltek alapján készítsük el a hőmérséklet beolvasó szubrutint
  - A beolvasott 16 bites érték r7:r6-ba kerüljön
- Milyen formátumú a beolvasott hőmérséklet adat?
  - Az első gyakorlat szorgalmi feladata volt





# A hőmérséklet beolvasása

## Egy lehetséges megvalósítás folyamatábrája



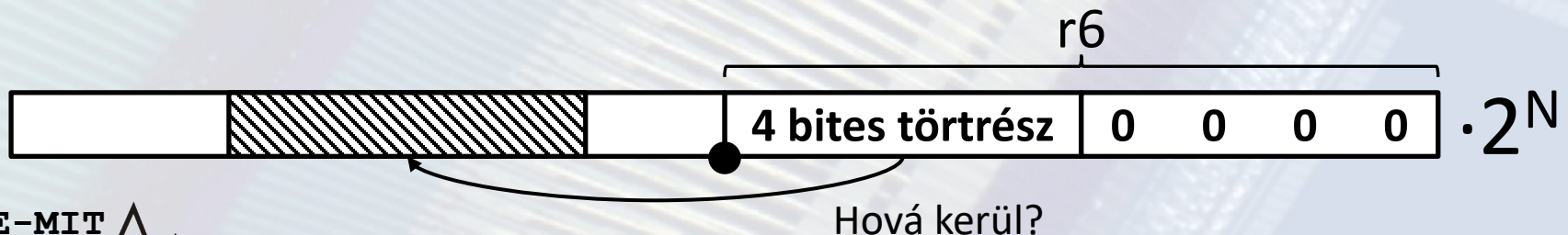
# A hőmérséklet megjelenítése

- Forgatással az egészrészből csak az alsó 8 bitet tartjuk meg, ez számunkra elegendő
  - A törtrész így r6 felső 4 bitjébe kerül
- A hőmérséklet egészrészét és törtrészét 2-2 digiten jelenítjük meg a hétszegmenses kijelzőn
  - A negatív értékekkel most nem foglalkozunk
  - A kijelző kezeléséhez a 9. laboron elkészített szubrutint használjuk a tizedespontok vezérlésével kiegészítve (*disp\_wr*)
- Az egészrész megjelenítéséhez bináris → BCD konverzió szükséges
  - Az ezt elvégző *bcd\_conv* szubrutin rendelkezésre áll



# A hőmérséklet megjelenítése

- **Mi a helyzet a törtrész megjelenítésével?**
  - A tört számok bináris  $\rightarrow$  BCD átalakításához nem használható a *bcd\_conv* szubrutin
- **A törtrészből 2 digités egész értéket százzal való szorzással kaphatunk**
  - Mely résszorzatok (N értékek) szükségesek?
  - Hogy lehet ezeket hatékonyan előállítani?
  - Készítsük el a *frac\_mul100* szubrutint



# A hőmérséklet megjelenítése

- A főprogramban inicializáljuk a GPIO A portot és végtelen ciklusban végezzük el a hőmérséklet beolvasását és megjelenítését
- A DIG2 és DIG1 digitek között a tizedespont legyen bekapcsolva
- A TMP121 hőmérő maximális konverziós ideje 640 ms, ezért két beolvasás között teljen el kb. 1 s
  - Készítsük el ehhez a *wait\_1s* szoftveres várakozási szubrutint az előző laboron tanultak alapján