

ARM Cortex magú mikrovezérlők

Mikrovezérlő alapperifériák

Scherer Balázs



Méréstechnika és
Információs Rendszerek
Tanszék

Tartalom

- Általános I/O lábak
- Timer-ek
- SPI
- I2C
- UART
- AD, DA, Komparátor
- CAN

GPIO lábak

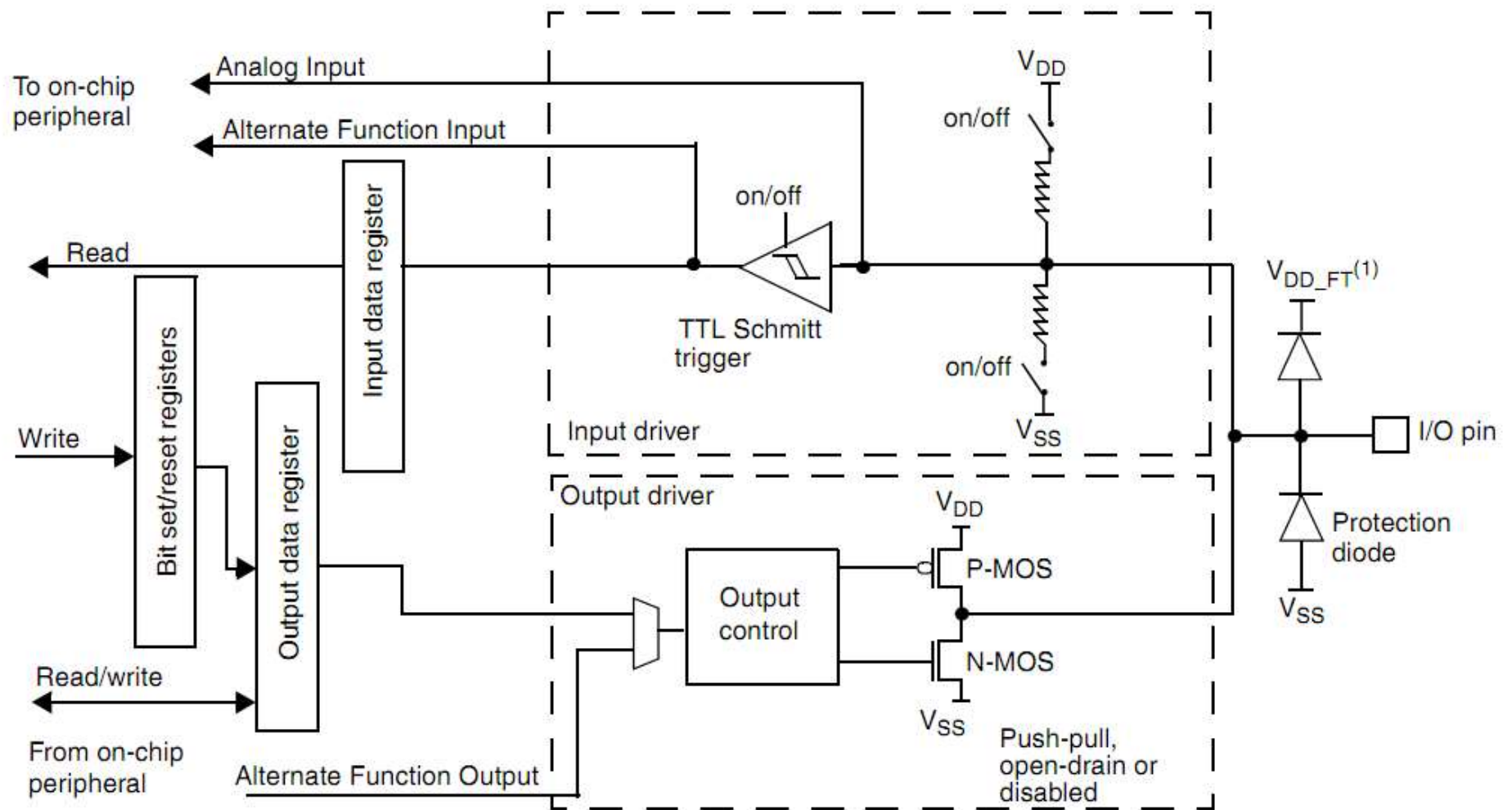
- Alap tulajdonság már a 70-es évektől
- Portokra osztva kezelhetőek:
 - Általában 8 bites portok
 - Ritkábban 32 bites (STM32 16 bites)
- Fontosak az elektronikus tulajdonságok
 - Max, min feszültség:
 - Logikai 0: általában max. 0,1V
 - Logikai 1: általában min. $V_{cc} \cdot 0,7$
 - 5 Volt toleráns vagy nem?
 - Maximális kiadható meghajtó és nyelő áram
 - Kiadható: 10mA, ritkábban 4mA
 - Nyelő áram: 20mA
 - Időzítés
 - Általában egy órajel ciklus alatt vezérelhetőek

GPIO lábak

- Egy GPIO porthoz általában 3 regiszter
 - PIN direction
 - Data Write
 - Data Read
- Manapság nem mindig ilyen egyszerű
 - Port funkció regiszter
 - A legtöbb GPIO lábnak van alternatív funkciója
 - Külön órajel generáló blokk

GPIO lábak

- Egy modern I/O blokk (STM32F)



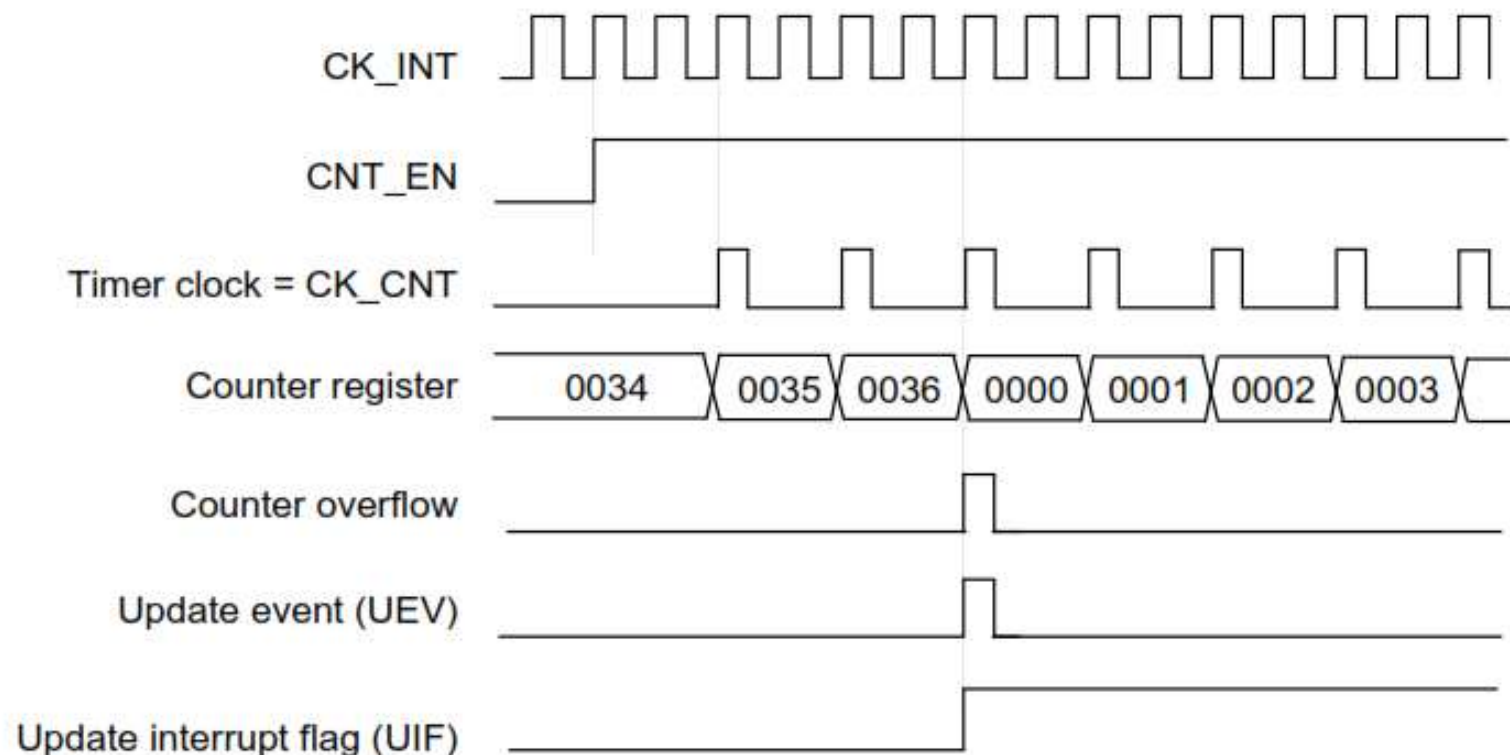
Timer-ek:

- Már a 70-es évektől megjelentek
- 8,16,32 bites változatok
- Különböző órajel források lehetségesek
 - Rendszer órajel
 - Külön Óra kvarc
 - Külső esemény
- Alap funkciók
 - Változtatható előosztó
 - Felfelé/Lefelé számolás
 - Töltés, törlés
 - Automatikus túlcsordulás, vagy megállás
 - Megszakítás kérés

Timer-ek: előosztó működés

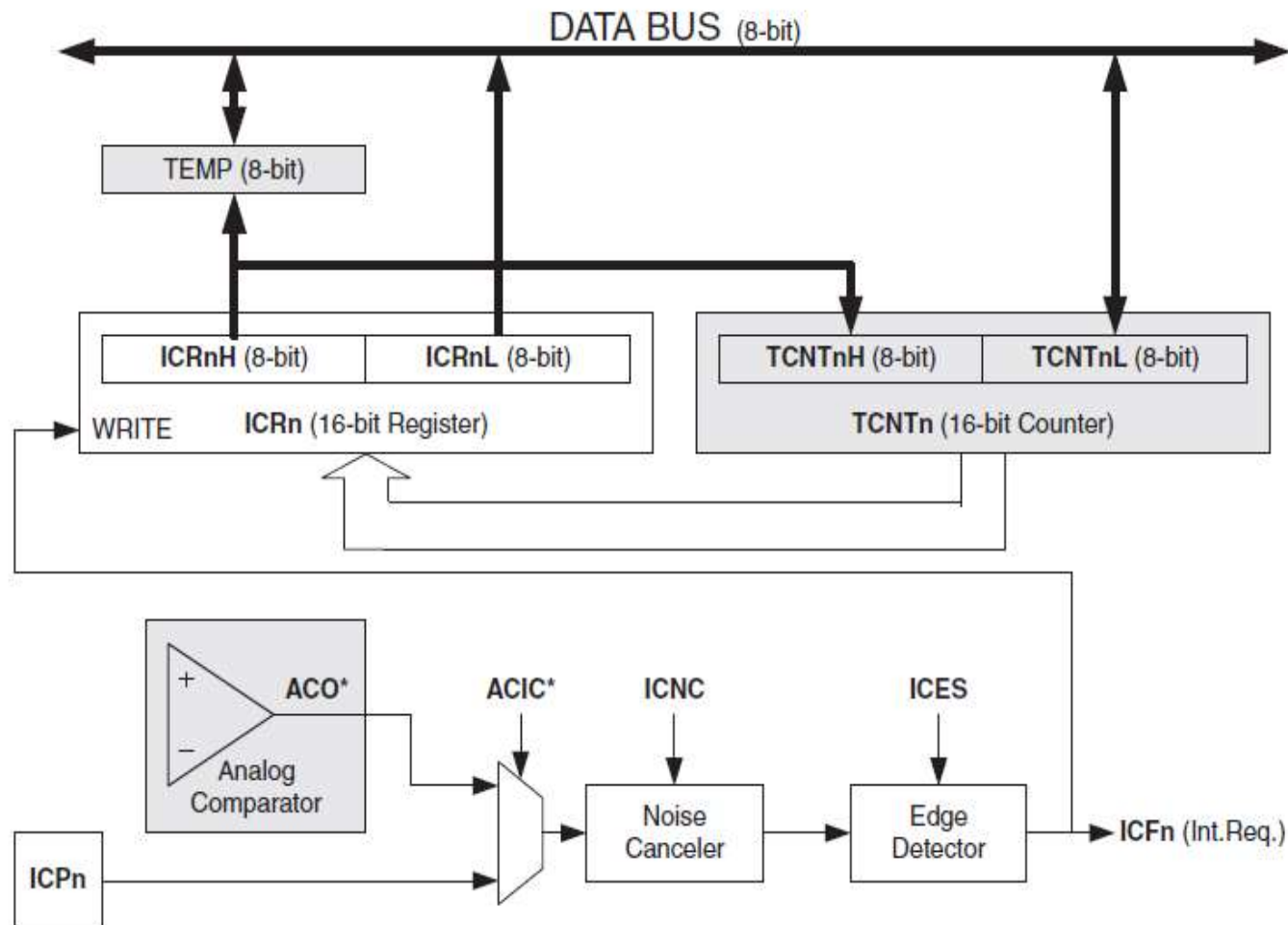
- Előosztó működés közben: valamilyen beállított értékig számol (pl.: Auto Relod value-ig, ami itt 36)

Figure 138. Counter timing diagram, internal clock divided by 2



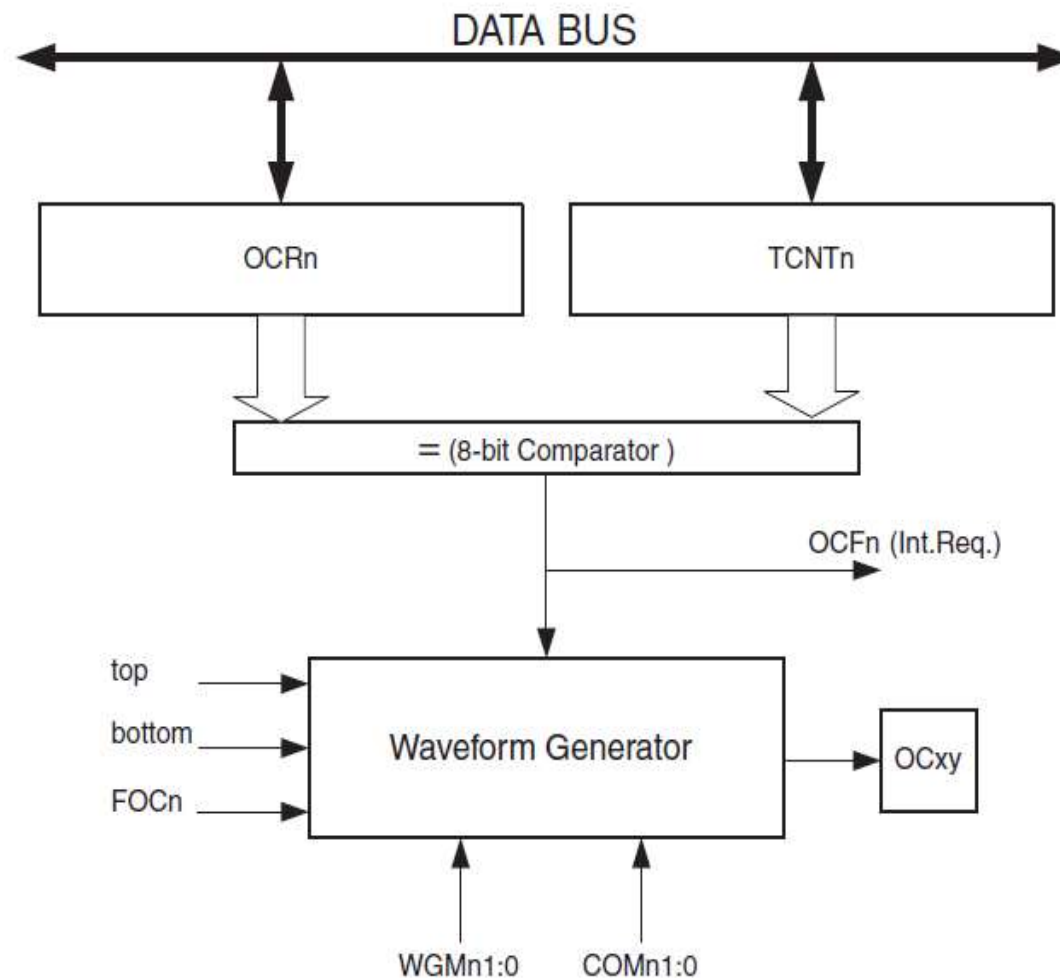
Timer-ek, Kiterjesztett funkciók

- Input capture



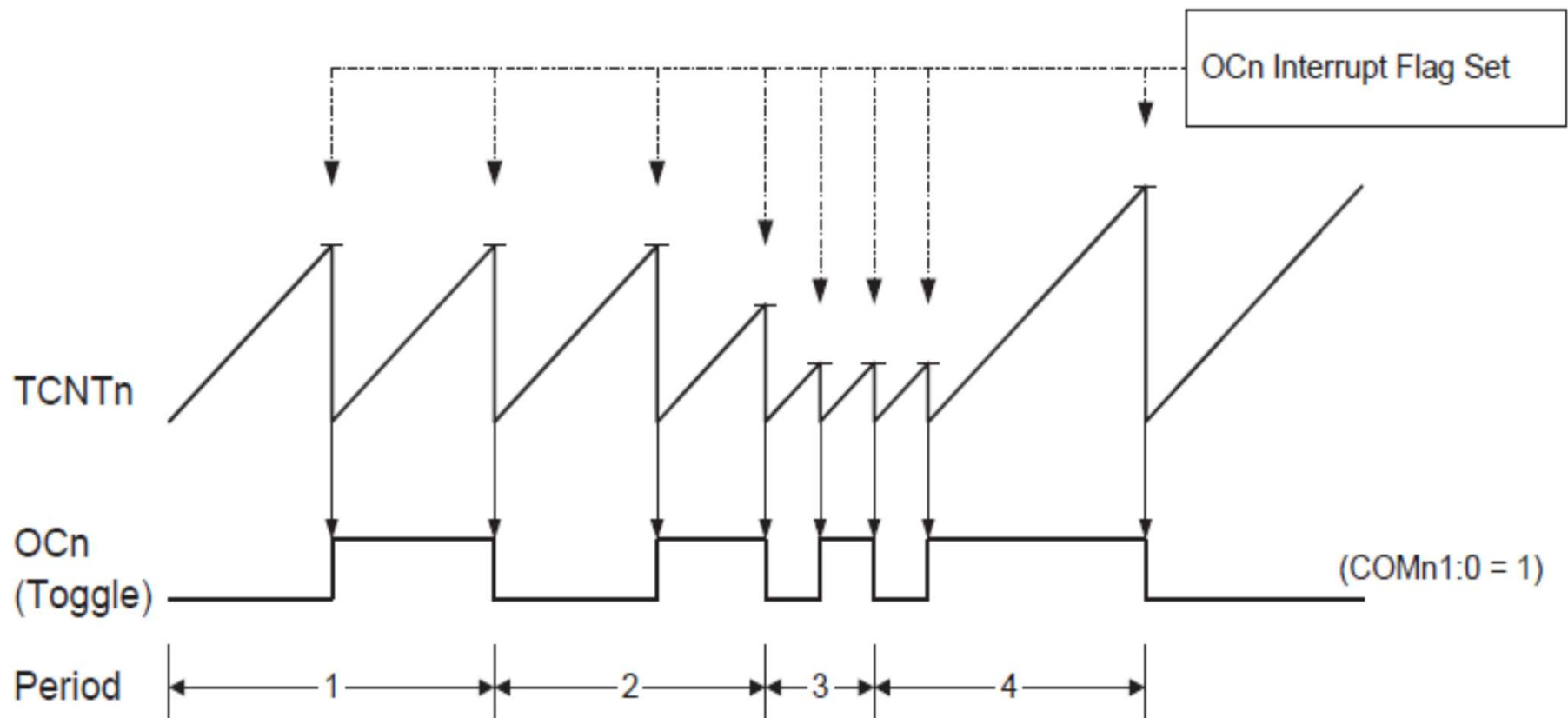
Timer-ek, Kiterjesztett funkciók

- Compare Output



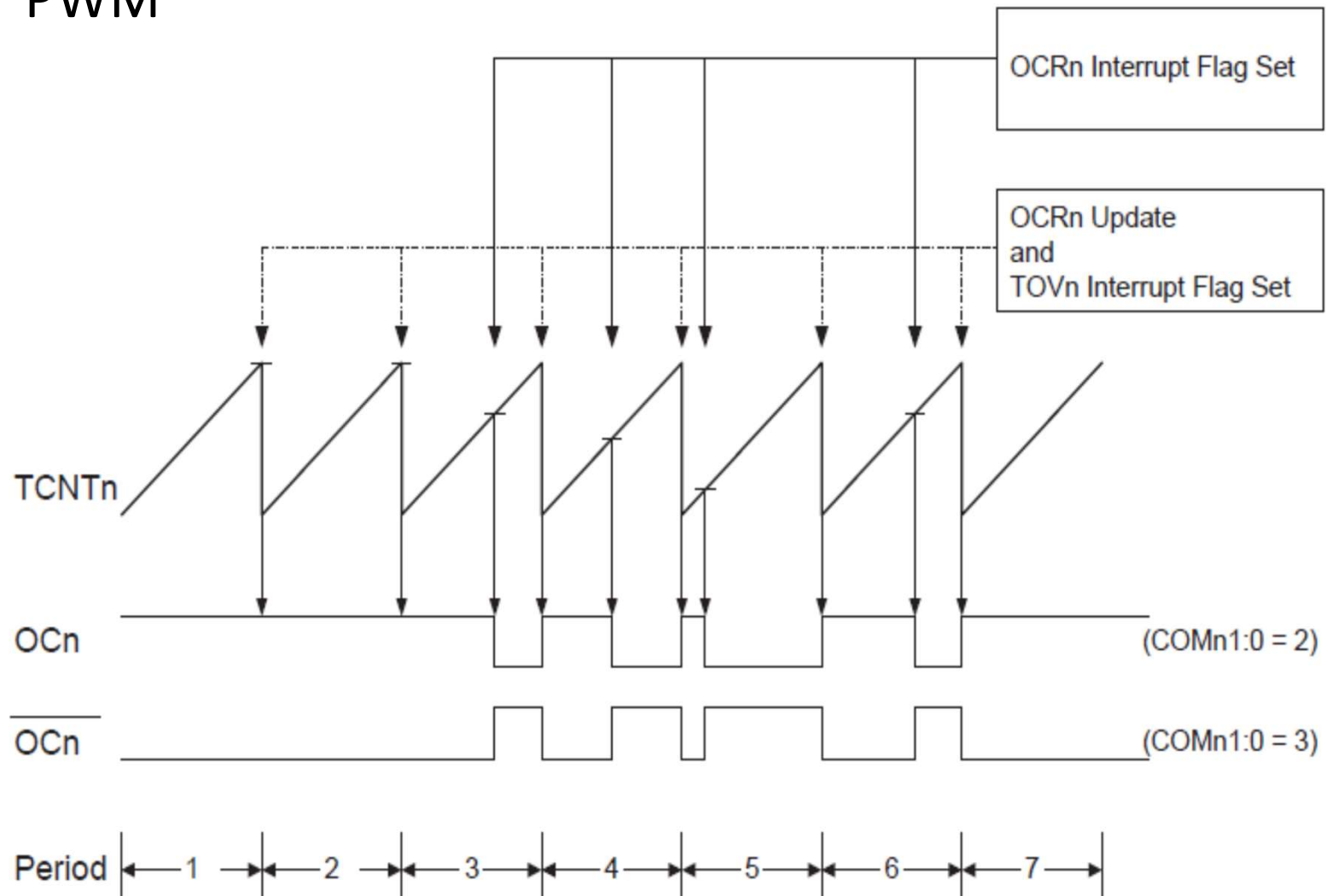
Timer-ek, Kiterjesztett funkciók

- Compare Output



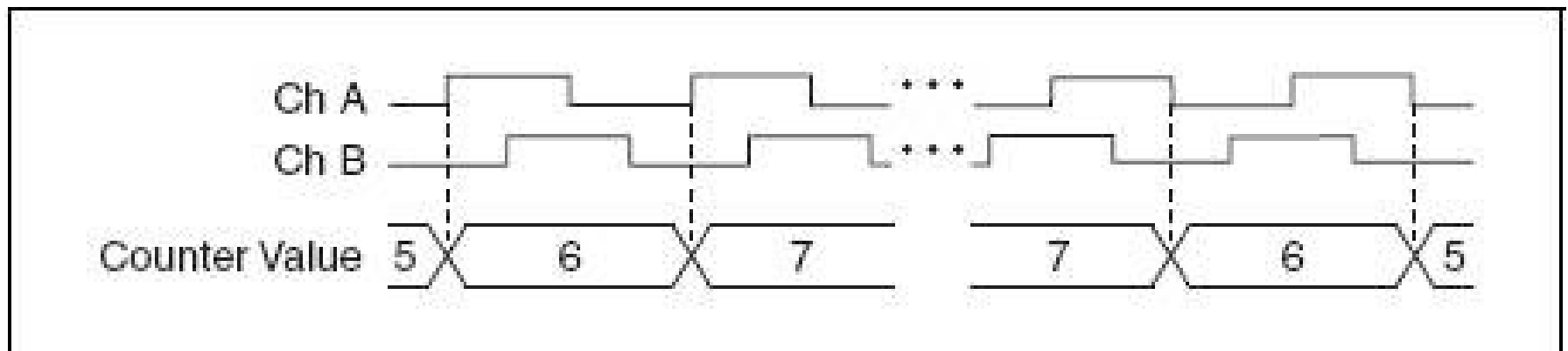
Timer-ek, Kiterjesztett funkciók

■ PWM



Timer-ek, Kiterjesztett funkciók

- Encoder interfész

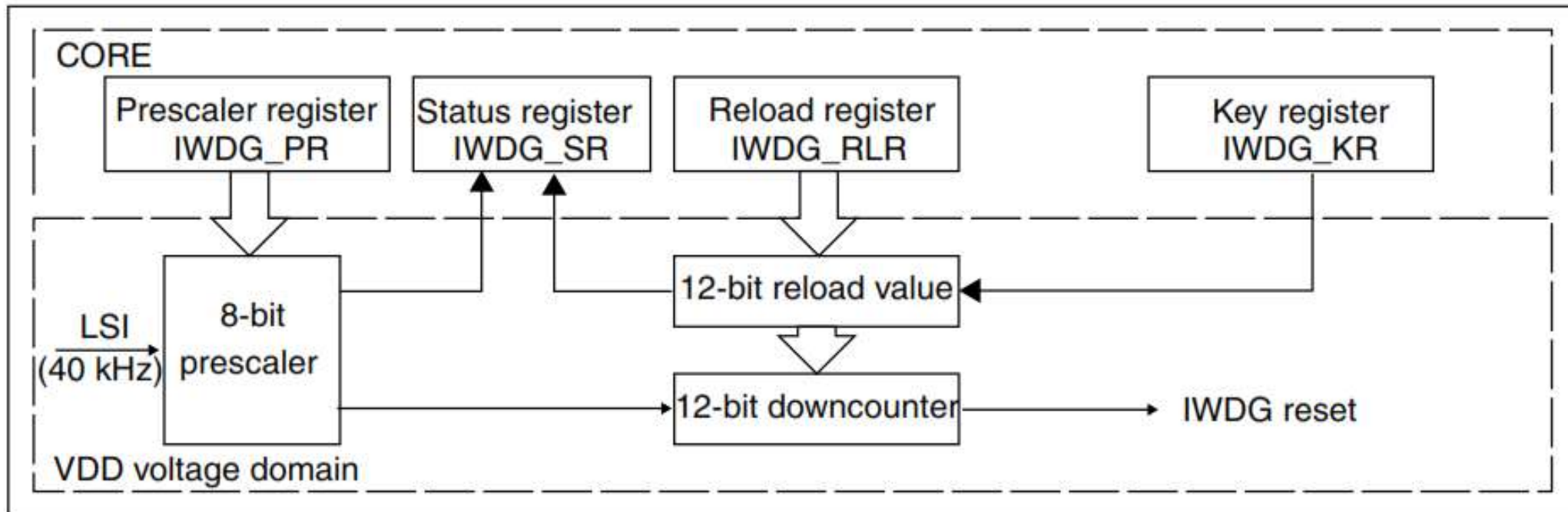


Timer-ek, Kiterjesztett funkciók

- Motor Control PWM-ek
 - Direkt 3 fázisú motor kezelésére optimalizált verziók
 - Timer Array-k
 - STM32, a 16 bites timerek összefűzhetőek tömbökké
- Real Time Clock
 - Külön 32.768 kHz-es kvarc
 - Általában külön tápbemenet
 - Külön elemet szoktak használni
 - Naptári óra funkcionalitás
 - Alarm lehetőség
 - Képes felkelteni a mikrovezérlőt alacsony fogyasztású módból

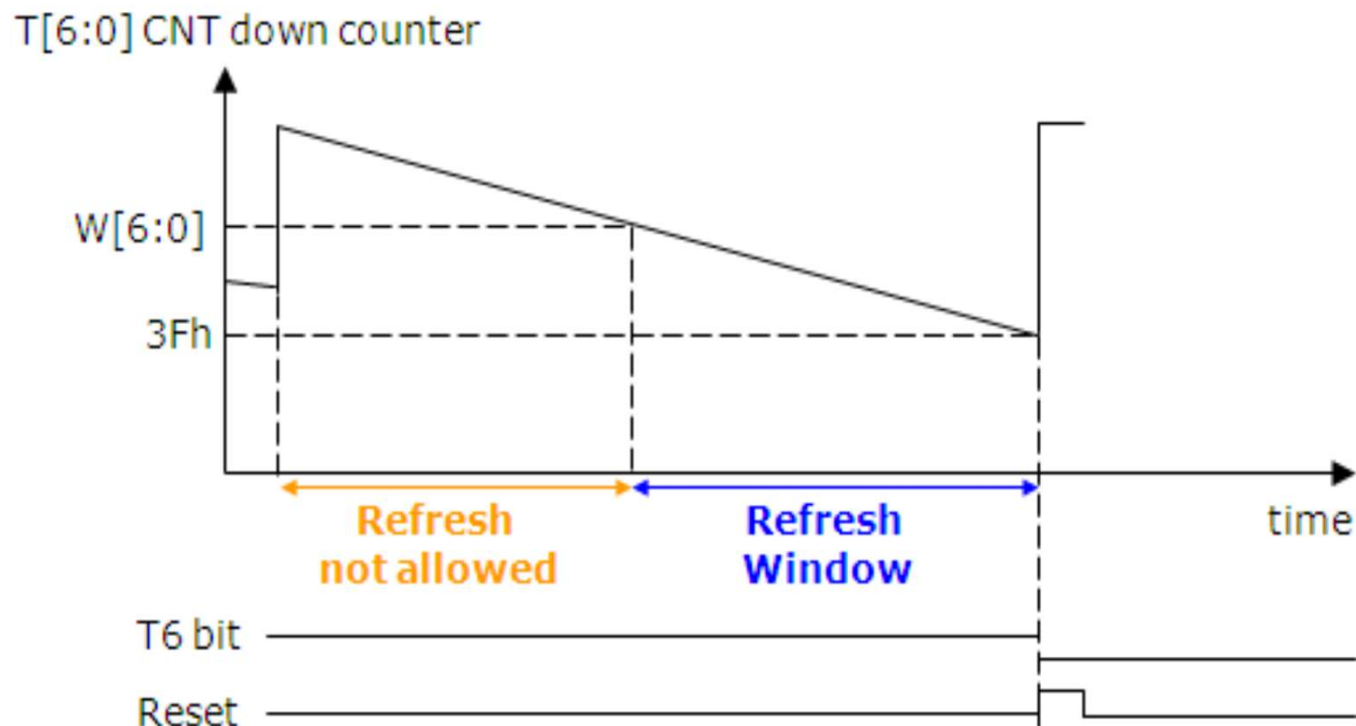
Timer-ek, Kiterjesztett funkciók

- Watchdog timer
 - Ha lejár akkor reset-eli a processzort
 - Fontos a független órajelforrás
 - Mi történik debugg alatt?



Timer-ek, Kiterjesztett funkciók

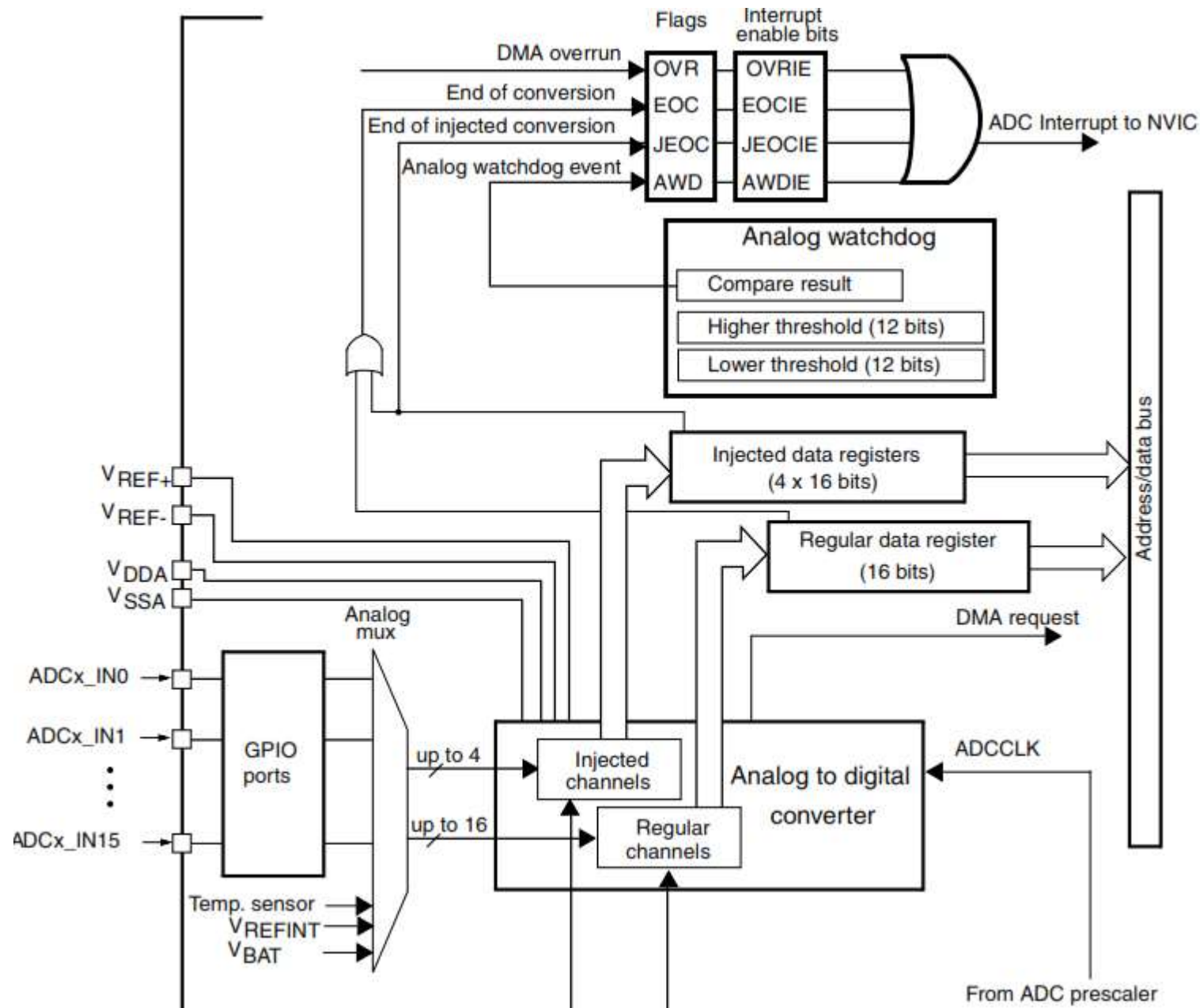
- Ablakozott watchdog timer
 - Ablakozott mód: Ha az újratöltött érték nagyobb a beírt értéknél, akkor is reset



ADC: Analog to Digital Converter

- Külön tápfeszültség bemenet
 - NYÁK tervezésnél figyelni kell rá, induktivitással van csatolva a normál táphoz
- Referencia:
 - Táp
 - belső általában 2,56V
 - Külső
- Leggyakrabban 12, esetleg 16 bites (successive approximációs AD)
 - Az Analog Devices gyártott 16 bites belső AD
- Normál I/O lábakkal közös lábakon
- Általában 1, max. 2 AD konverter sok csatornával
- Sebesség pár MSample/sec-esek.

ADC: Analog to Digital Converter



ADC: Analog to Digital Converter

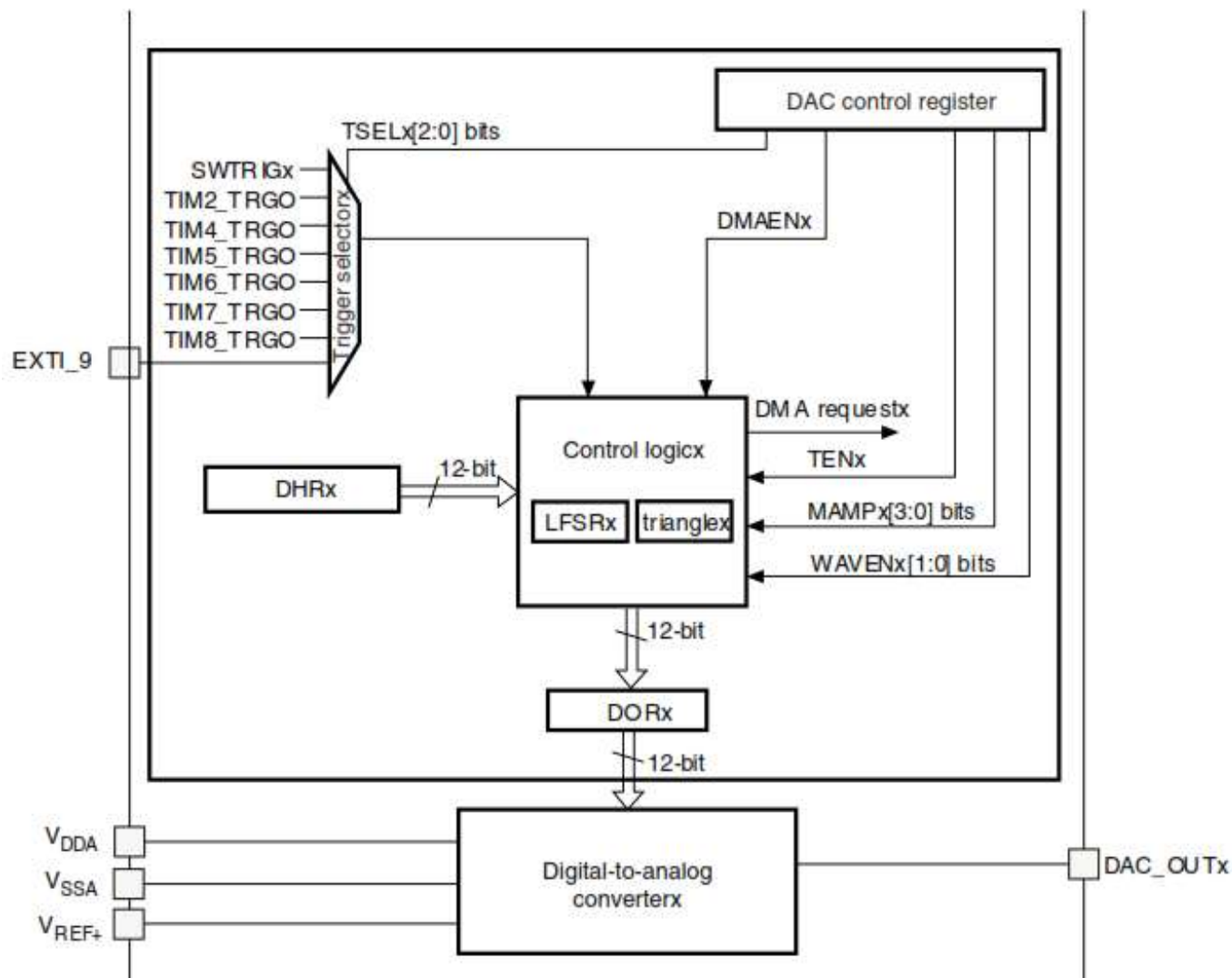
- Adatrendezés



- Több AD szinkronizálása

DAC: Digital to Analog Converter

- Sokszor a használható hullámforma generálásra is

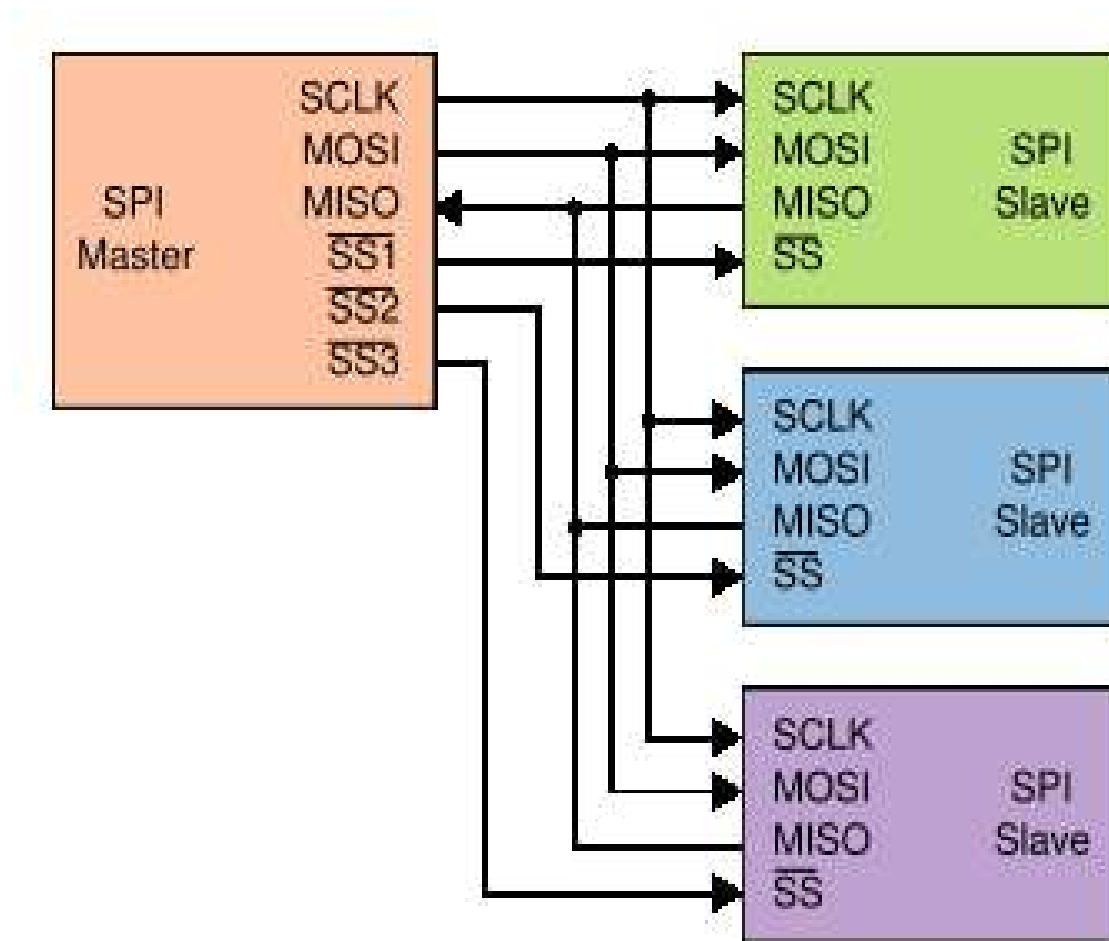


SPI: Serial Peripheral Interface

- Motorola fejlesztette ki
- Master – Slave filozófia
- 4 vezetékes kommunikáció
 - SSEL: Slave Select
 - SCK: Serial Clock
 - MOSI: Master Out Slave In
 - MISO: Master In Slave Out
- Szinkron adatátvitel
 - Adat az órajel lefutó vagy felfutó élére érvényes
- Viszonylag gyors full-duplex kommunikáció
 - SCK lehet MHz feletti is.

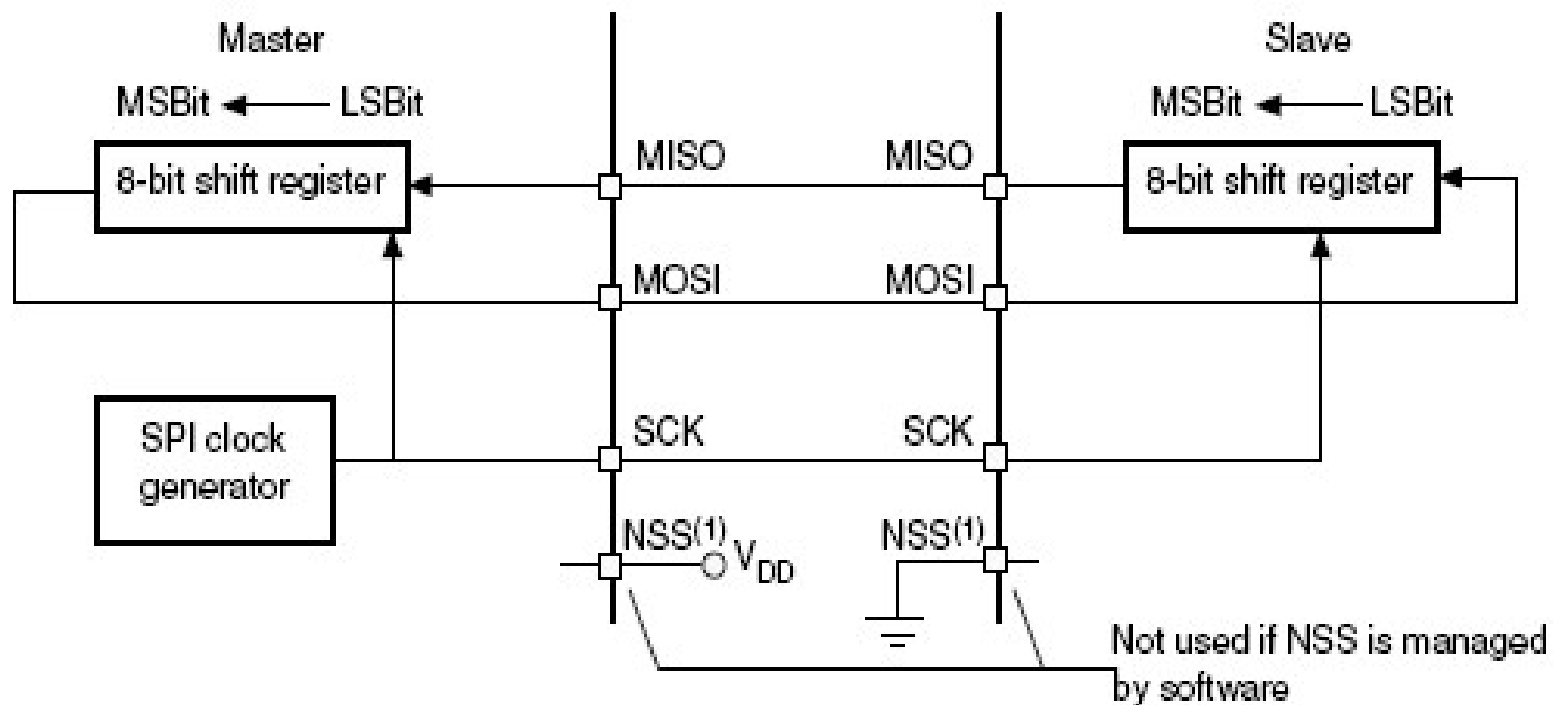
SPI: Serial Peripheral Interface *folytatás*

- Master – Slave filozófia



SPI: Serial Peripheral Interface *folytatás*

- 4 vezetékes átvitel



SPI: Serial Peripheral Interface *folytatás*

- Tipikus SPI-al csatlakozó perifériák
 - AD, DA átalakítók
 - Könnyen lehet galvanikusan izolálni
 - EEPROM-ok
 - Szenzorok
 - Hőmérséklet mérők
 - Gyorsulásmérő
 - Kommunikációs kontrollerek
 - Wifi, CAN, ZigBee stb.
 - Grafikus kontrollerek

SPI 32 bites mikrovezérlőkben

■ STM32F107

- Max. 18MHz full-duplex, 8, 16 bites blokkok.
 - DMA az adási és vételi oldalra is külön
- Hardware támogatott CRC8, CRC16 kalkulálás
 - MMC, SD kártya kezeléshez hasznos

■ LPC18xx Quad SPI interface

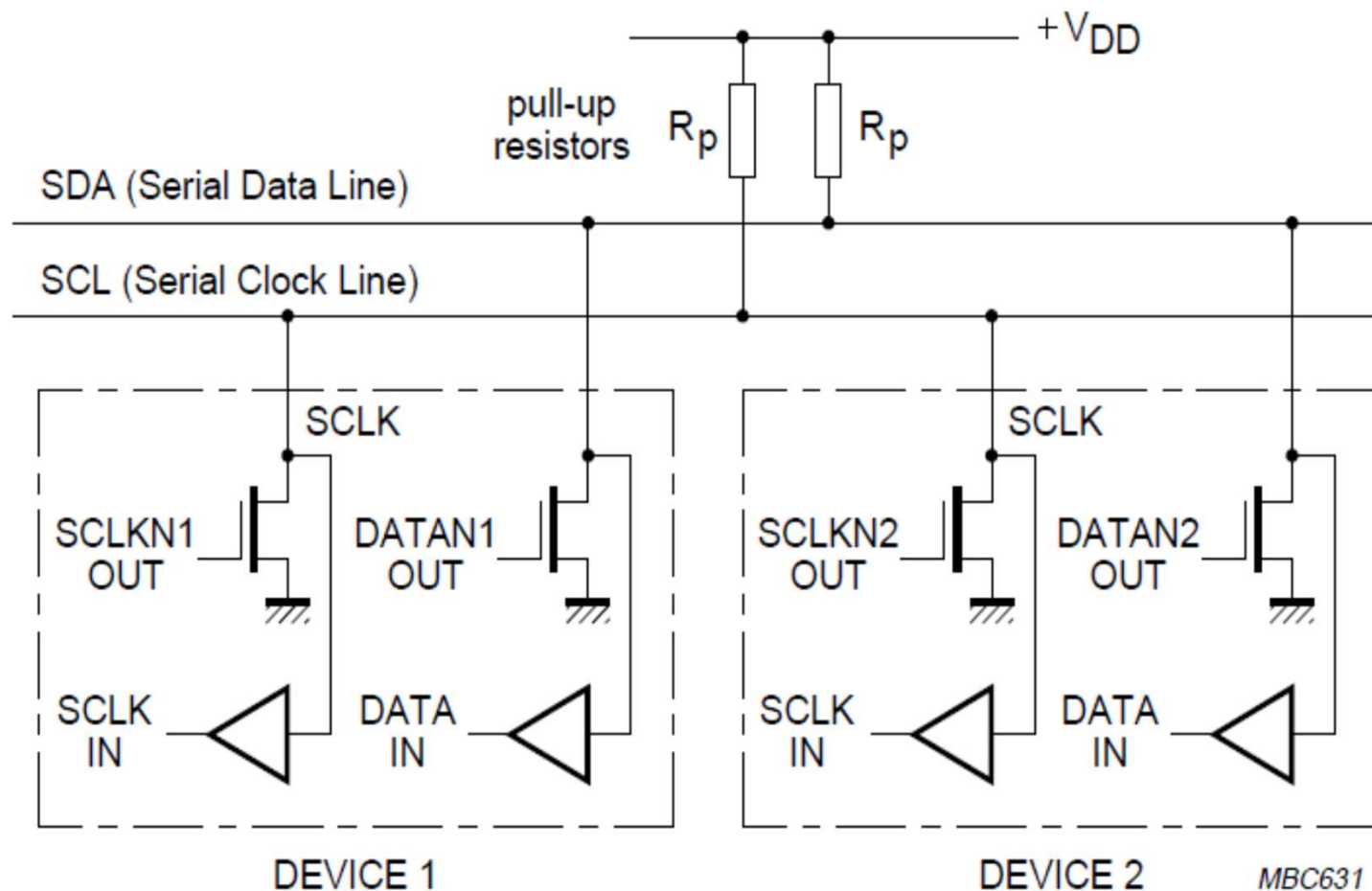
- 1,2,4 bites üzemmód.
- Max 54 Mbit/sec

Inter-Integrated Circuit: I2C

- A szabvány első verziója 1992-ben jelent meg.
- Kétvezetékes, fél-duplex átvitel
 - SCL: Serial Clock Line
 - SDA: Serial Data Line
- Általában 1 master több slave, de lehetséges több master is
- A node-ok számát a busz kapacitása korlátozza (400pF)
- Szabadon választható sebesség
 - 100kbps (standard)
 - 400kbps (fast)
 - 3,6Mbps (high-speed)
- A Slave-eket címezéssel azonosítjuk
 - 7 bites standard címezés
 - 10 bites kiterjesztett címezés (általában nem használt, picit erőszakolás)

Inter-Integrated Circuit: I2C *folytatás*

- Fizikai elrendezés
 - Logikai 1: Vdd, Logikai 0: GND



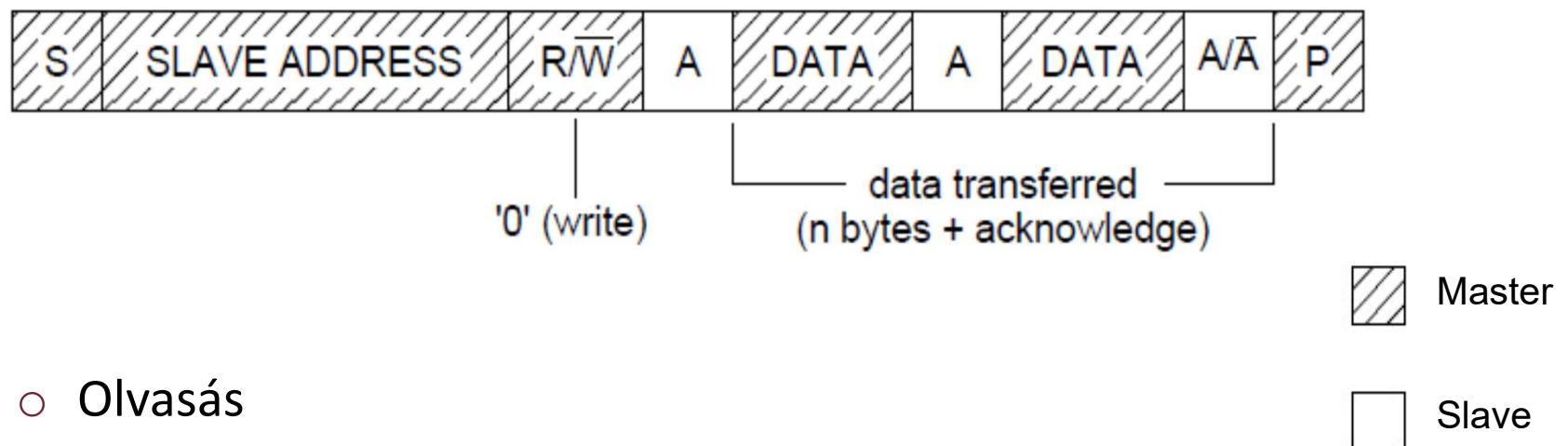
Inter-Integrated Circuit: I2C *folytatás*

- A Start feltétel után adatbiteket viszünk át
- Az SCL jelet csak a Master hajtja meg
- Az SDA nem változhat az SCL magas állapota alatt (adatátvitel)
- Byte centrikus adatátvitel 8 bit + ACK bit
- Az MSB bit először
- Cím byte, majd adat byte-ok
- Több Master esetén arbitráció van (huzalozott ÉS kapcsolat), az arbitráció már a cím alatt el szokott dőlni.

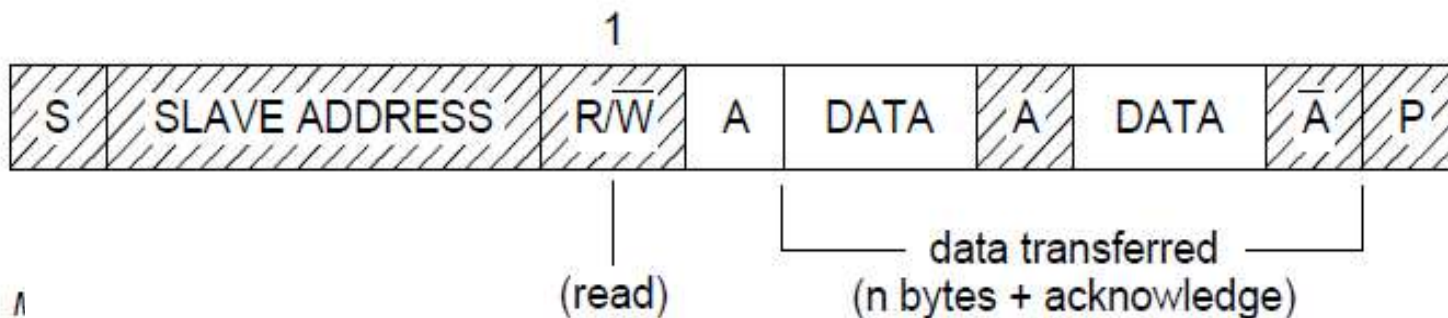
Inter-Integrated Circuit: I2C *folytatás*

■ Adatátvitel

○ Írás



○ Olvasás



Inter-Integrated Circuit: I2C *folytatás*

- Tipikus I2C-vel csatlakozó perifériák
 - AD, DA átalakítók
 - Újabb sorozatoknál
 - EEPROM-ok
 - Szenzorok
 - Hőmérő
 - Gyorsulásmérő
 - Nyomásmérő
 - Páratartalom mérő
 - Port bővítők

- Aszinkron soros átvitel
 - Már a mechanikus telegráf jellegű készülékeknél használtak ilyen átvitelt
 - 1970-es években kezdtek el ilyen aszinkron átvitelt használni
 - 1980 Más-más néven, de a Motorola és az Intel is kihozza saját UART jellegű chipjeit
 - 1990 Első bufferelt UART chipek
 - A mai napig alapfelszerelés az összes mikrovezérlőnél

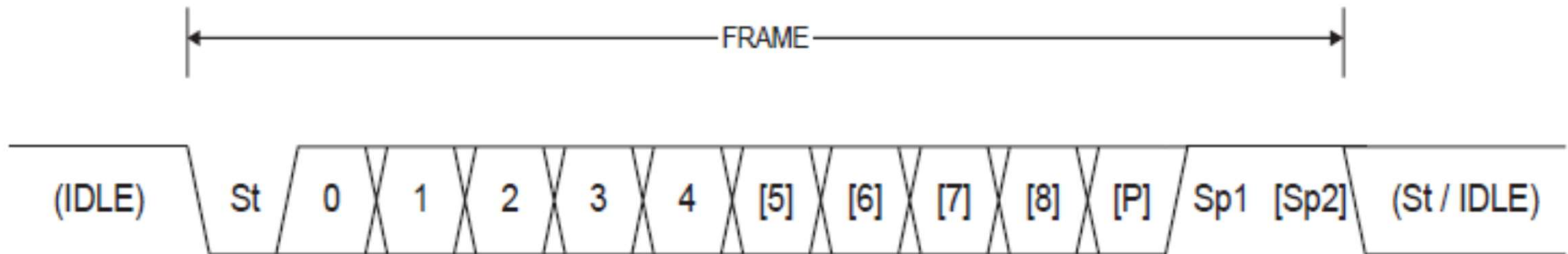
UART *folytatás*

■ Egy UART keret

- Start Bit
- 5, 6, 7, 8 vagy 9 adat bit
- Paritás Bit
- 1, 1.5 vagy 2 Stop Bit

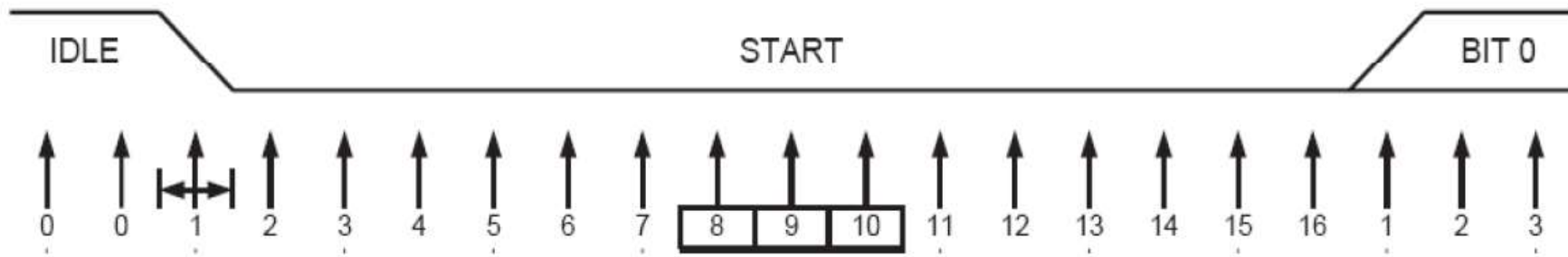
■ Szabványos adatsebességek

- 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200



UART *folytatás*

- UART bit felismerés, vétel.
 - A start bit lefutó élére elinduló mintavételezés
 - A bitszinkronnak az egész kereten keresztül ki kell tartania
 - Óraforrás csúszás problémát okozhat (RC oszcillátor, Hőmérséklet)
 - 2% felett jelentős probléma lehet (gyakorlatban <1%-ot)



UART *folytatás*

- Az UART használata
 - PC terminál, debug, műszer kapcsolat (Rs232)
 - Ipari kommunikációs hálózat (Rs485)
 - IrDA
 - Autóipari kommunikációs hálózatok
 - LIN (Local Interconnect Network)
 - Modem jellegű illesztésre kommunikációs chipekkel
 - GSM, GPS, GPRS modem
 - ZigBee
 - TCP/IP chip
 - Power Line kommunikáció (külön illesztő chip)
 - USB virtuális soros port (FTDI chipek)

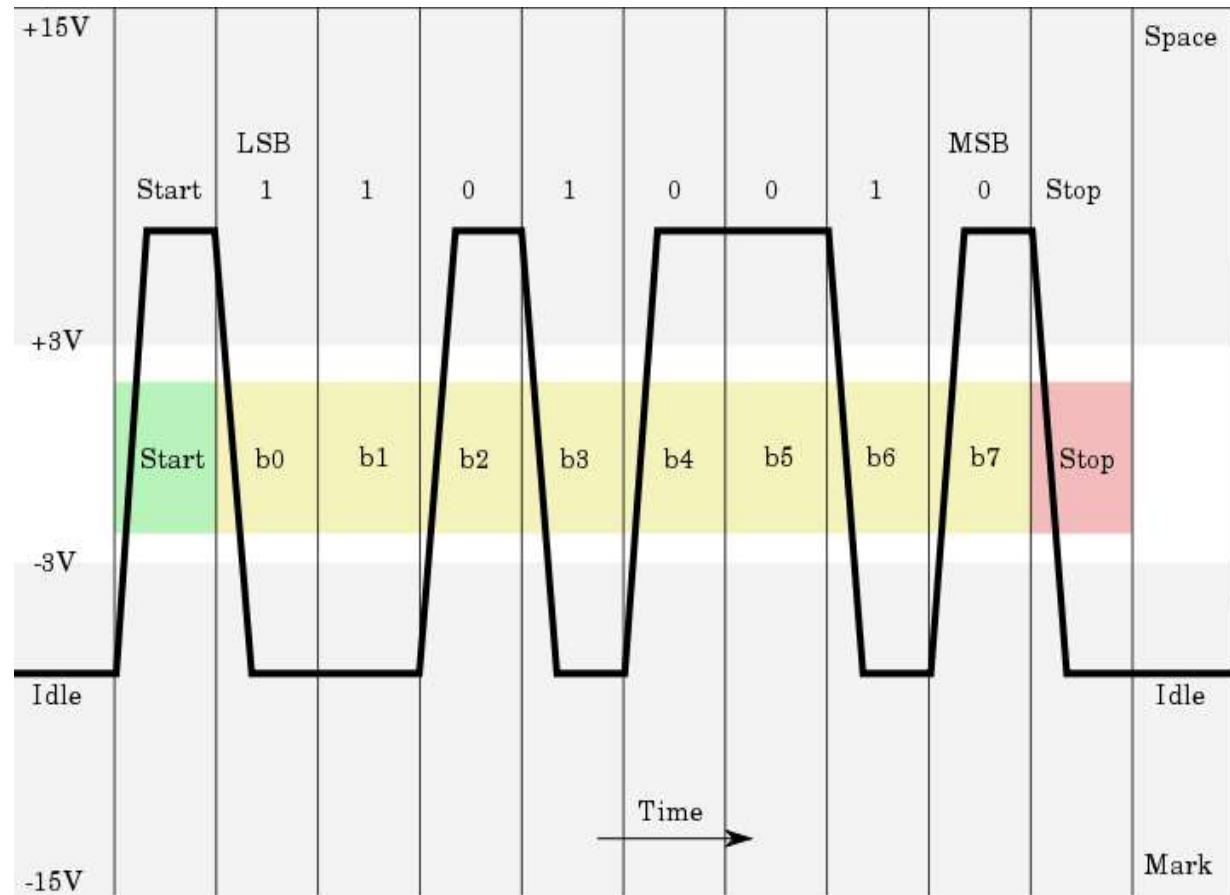
Rs232 (EIA Standard RS-232-C)

■ Logikai fizikai szint kapcsolat

Logikai szint	Adó	Vevő
0	+5...+15	+3...+25
1	-5...-15	-3...-25

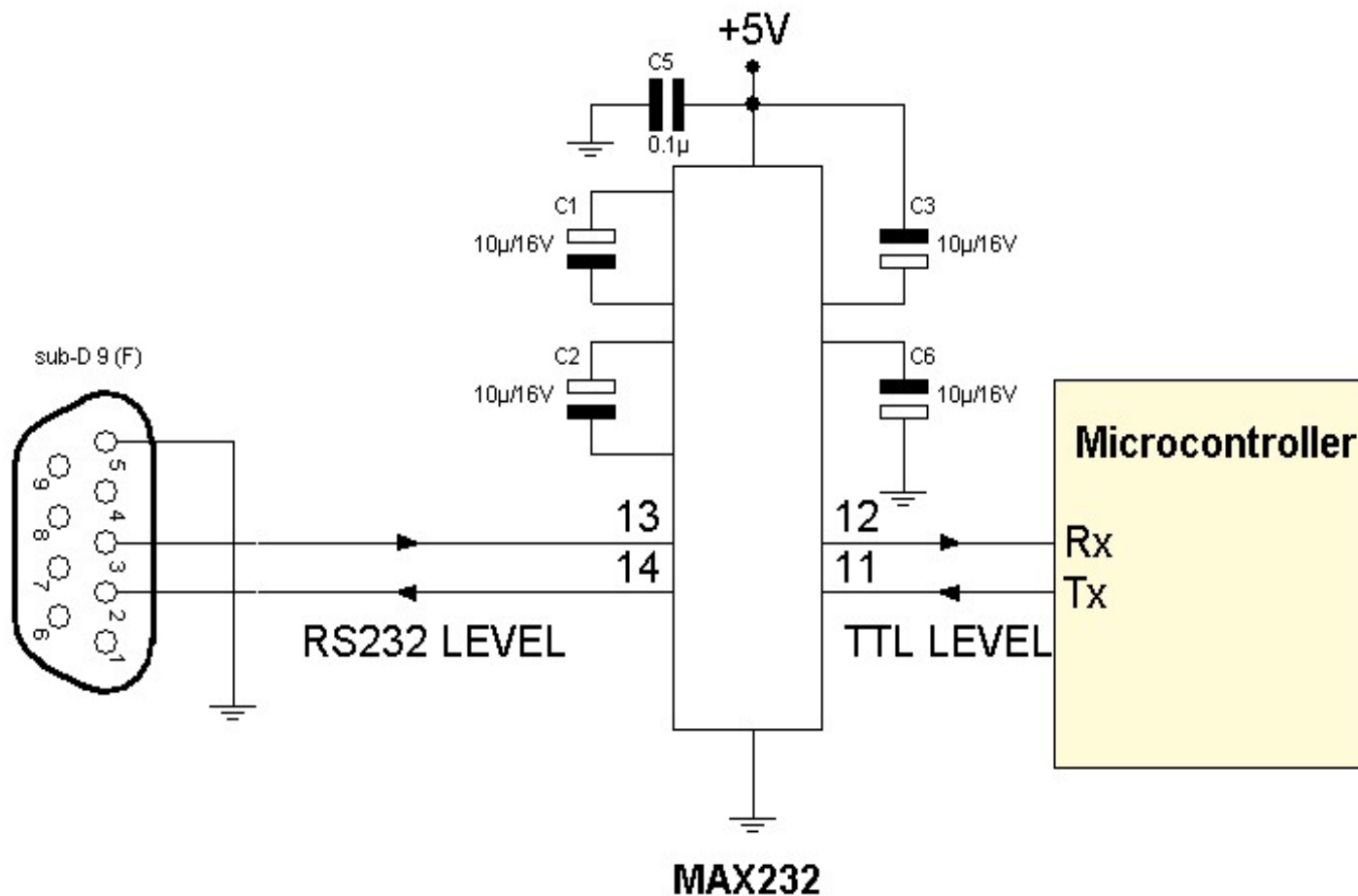
■ Átviteli távolság

Baud rate	Kábelhossz (kb)
19200	15m
9600	150m
4800	300m
2400	1000m



Rs232 (EIA Standard RS-232-C) folytatás

- Rs232 alapú mikrovezérlő – PC kapcsolat

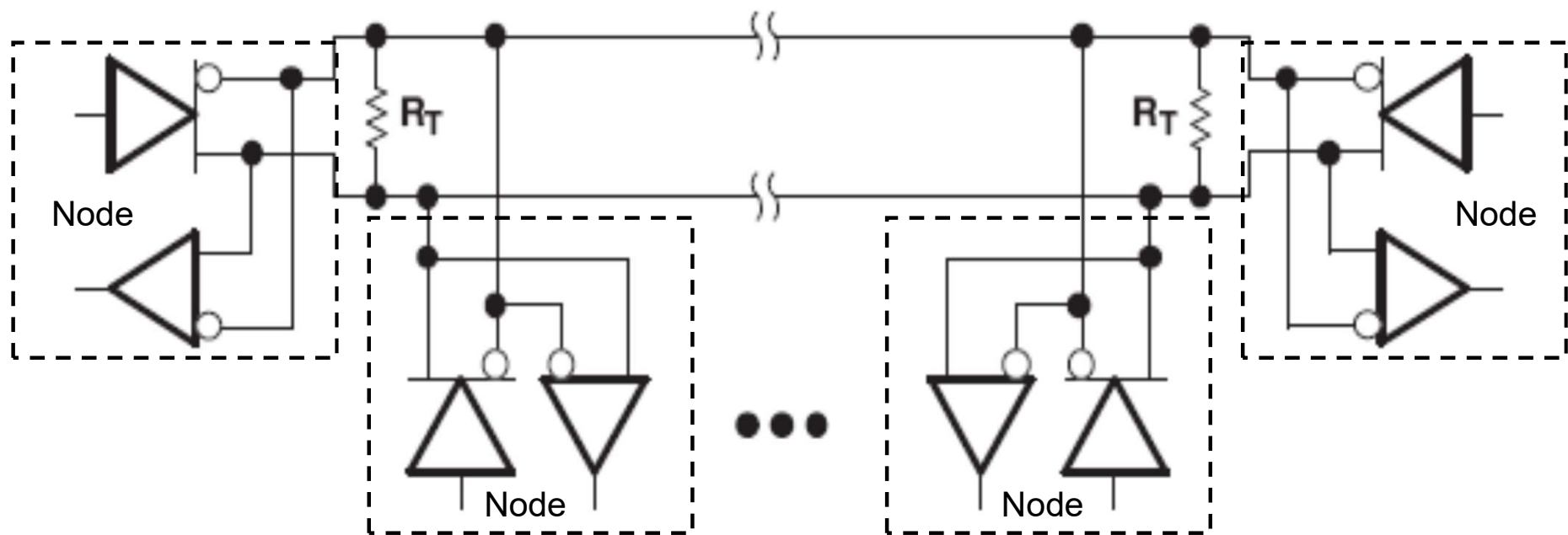


Rs485 (EIA-485)

- Hálózati elrendezés
 - maximum 1200m
- Csak az elektromos karakterisztika
- Nem ad meg adatkapcsolati protokollt
- Sodort érpár differenciális meghajtással
 - Lezáró ellenállás reflexió ellen
- 2 vezetéket használ half-duplex kapcsolatként
 - Általában master-slave protokoll
- Lehetőség van full-duplex kommunikációra is 4 vezetéken

Rs485 (EIA-485)

■ Hálózati elrendezés

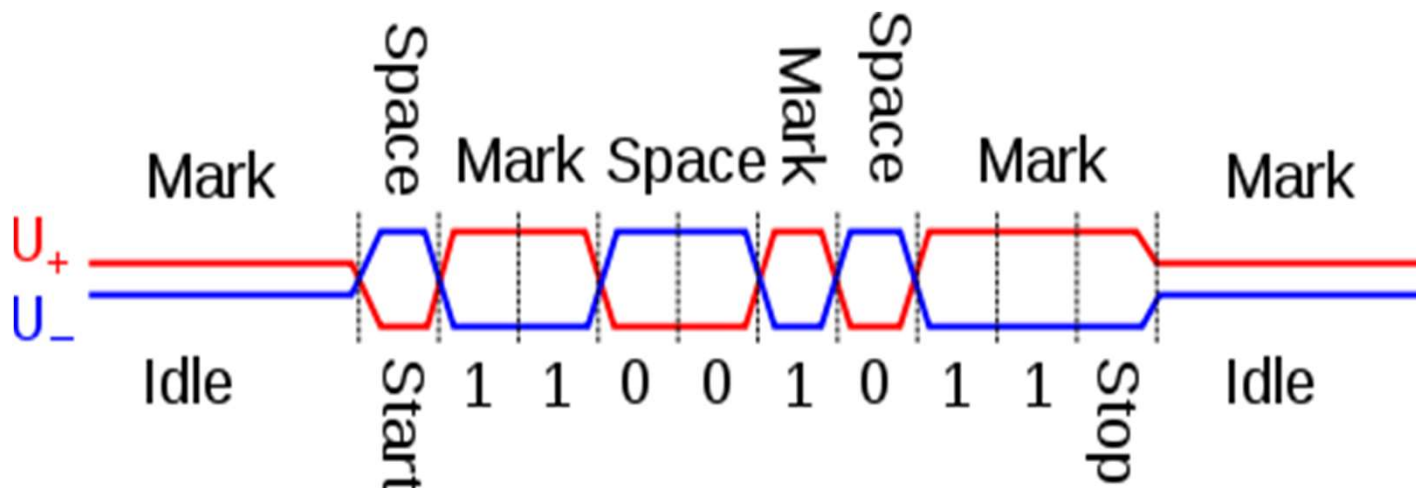
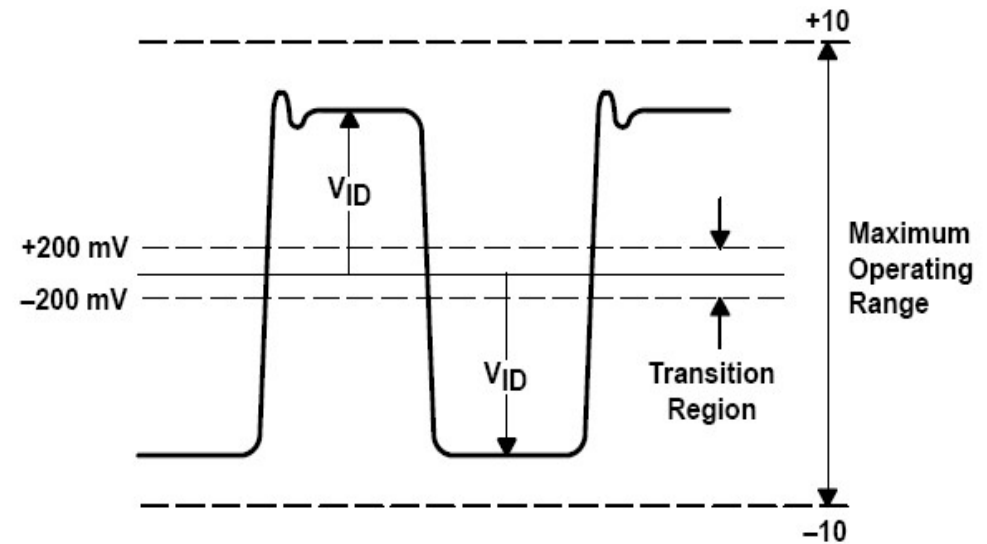


Rs485 (EIA-485) folytatás

■ Logikai fizikai szint kapcsolat

○ Differenciális jel

- **A** az invertált vagy '-' jel
- **B** a nem-invertált vagy '+' jel



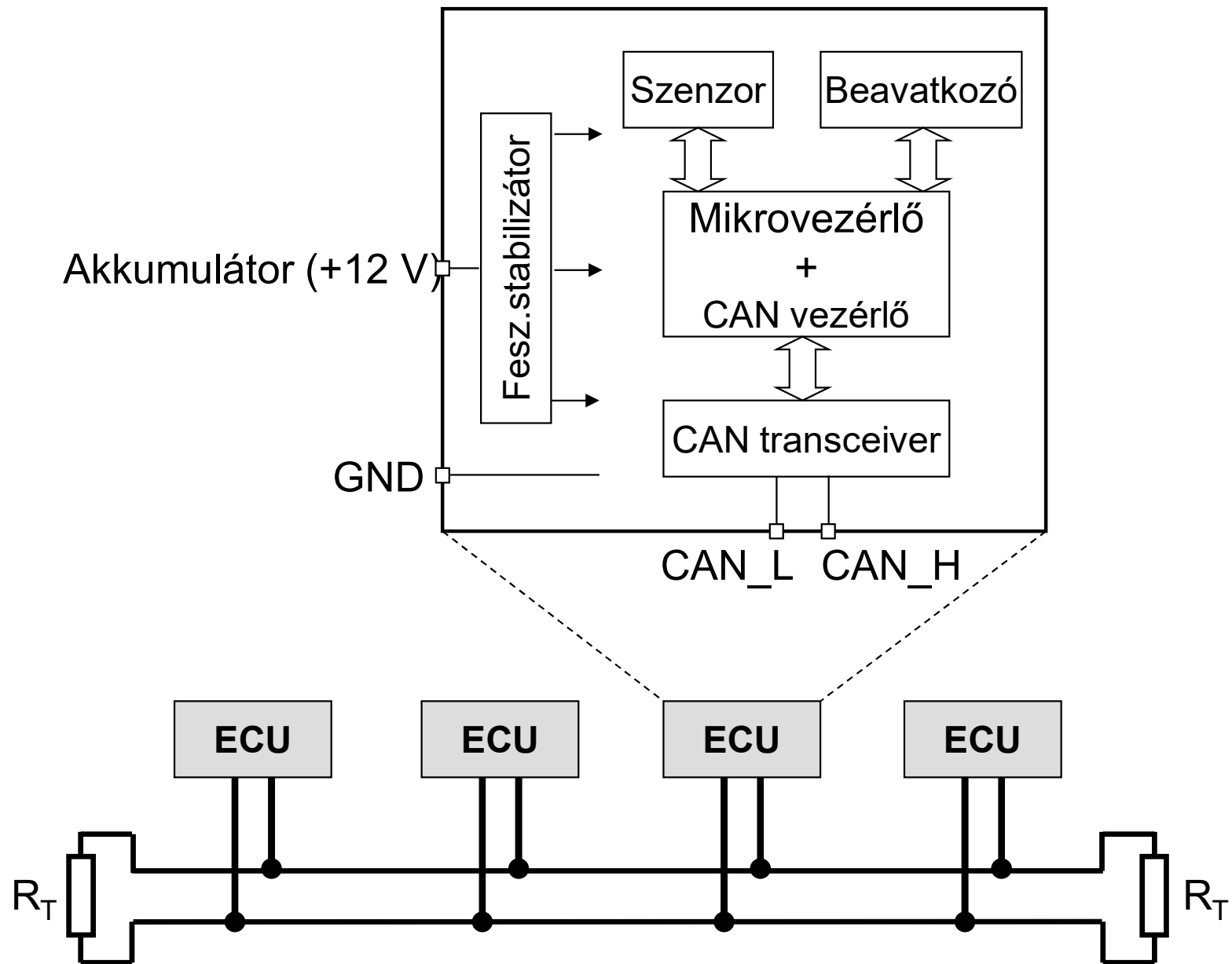
Rs485 (EIA-485) folytatás

- Alkalmazás: Az egyik legolcsóbb ipari busz
 - Újabban a CAN, vagy az Ethernet kezdi kiszorítani
 - Sok az adatkapcsolati szoftver teendő: byte-ok közötti idő, frame-ek közötti idő specifikálása és ellenőrzése.
- PLC-k
- Modbus
- Profibus

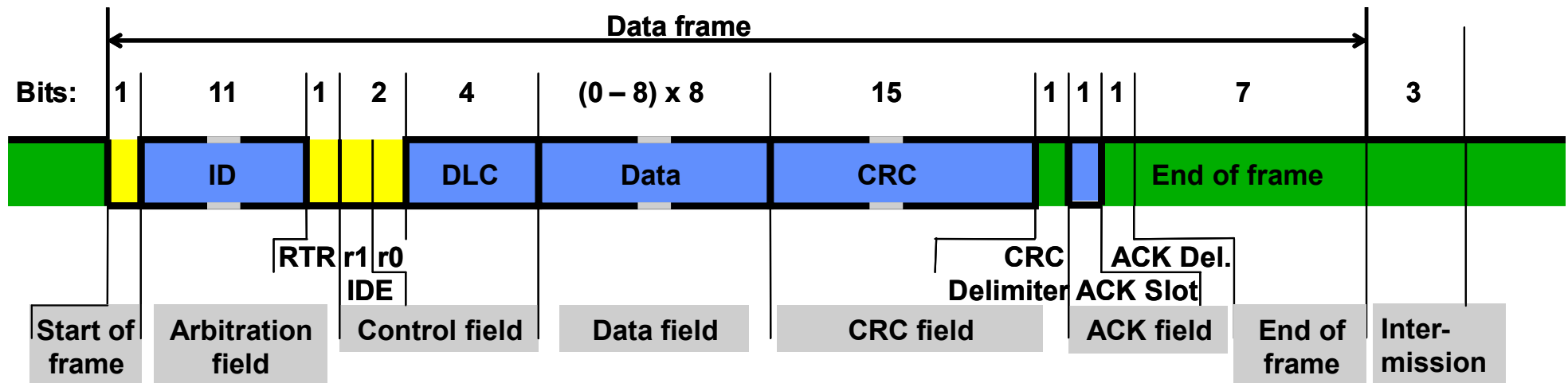
CAN: Controller Area Network

- 1 Mbit/s maximális adatátviteli sebesség (kis sebességű: 10 kbit/s-tól 125 kbit/s-ig, nagy sebességű: 125 kbit/s-tól 1 Mbit/s-ig, jellemzően 500 k)
- Áthidalható távolság:
 - 100 m (330 ft), 500 kbit/s
 - 200 m (650 ft), 250 kbit/s
 - 500 m (1600 ft), 125 kbit/s
 - 6 km (20000 ft), 10 kbit/s
- Többféle adatátviteli közeg, legtöbbször csavart érpár
- Non-Return to Zero (NRZ) bitkódolás bitbeszúrással és bitkiejtéssel (bit-stuffing)
- Rövid, változó hosszúságú keretek (0–64 bit hosszú adatmező, 0–8 adatbyte).

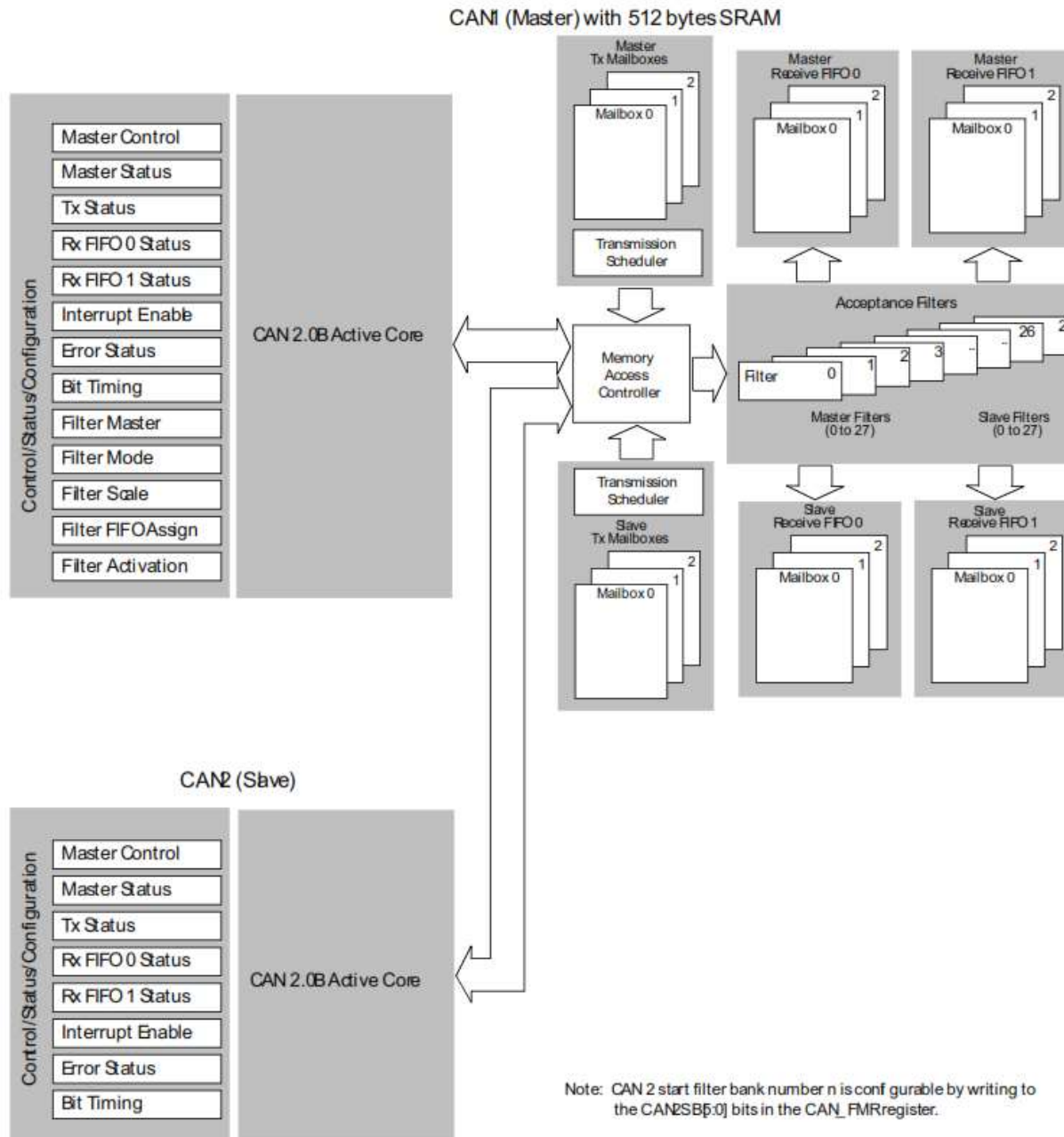
Egy CAN hálózat (példa)



Data frame – 2.0A / Standard CAN



Egy CAN controller



ARM Cortex magú mikrovezérlők

Egyszerű firmware libraryk

Scherer Balázs



Méréstechnika és
Információs Rendszerek
Tanszék

Doxygen

Automatikus dokumentáció generálás: Doxygen

- 1997-ben jelent meg az első verzió v0.1
- A komment – dokumentáció közötti inkonzisztenciát igyekszik feloldani
- Többfajta kommentezési stílust képes kezelni
- C stílusú JAVA doc szerű kommentek

```
/**  
... text ...  
*/
```

- C stílusú ! kommentek

```
/*!  
... text ...  
*/
```

Doxygen kommentek példa

```
/*! \fn void UART1_Init(unsigned long baud_rate, void (*handler)(void));  
 * \brief An inicialisation function to redirect STDIO to UART1  
 * \param baud_rate: Baudrate in bit/sec  
 * \param handler: Callback function for receiving UART characters with IT  
 * \return nothing  
 */
```

```
/** @fn void UART1_Init(unsigned long baud_rate, void (*handler)(void));  
 * @brief An inicialisation function to redirect STDIO to UART1  
 * @param baud_rate: Baudrate in bit/sec  
 * @param handler: Callback function for receiving UART characters with IT  
 * @return nothing  
 */
```

Csoportosítás

- Alapvetően file szintű dokumentálás, ahhoz, hogy logikai egység szintű dokumentációt csináljunk szükséges a csoportok megalkotása
- Szinte az összes firmware library ezt alkalmazza

```
/** @addtogroup Az én csoportom  
@{  
*/  
  
.....  
/**  
@}  
*/
```

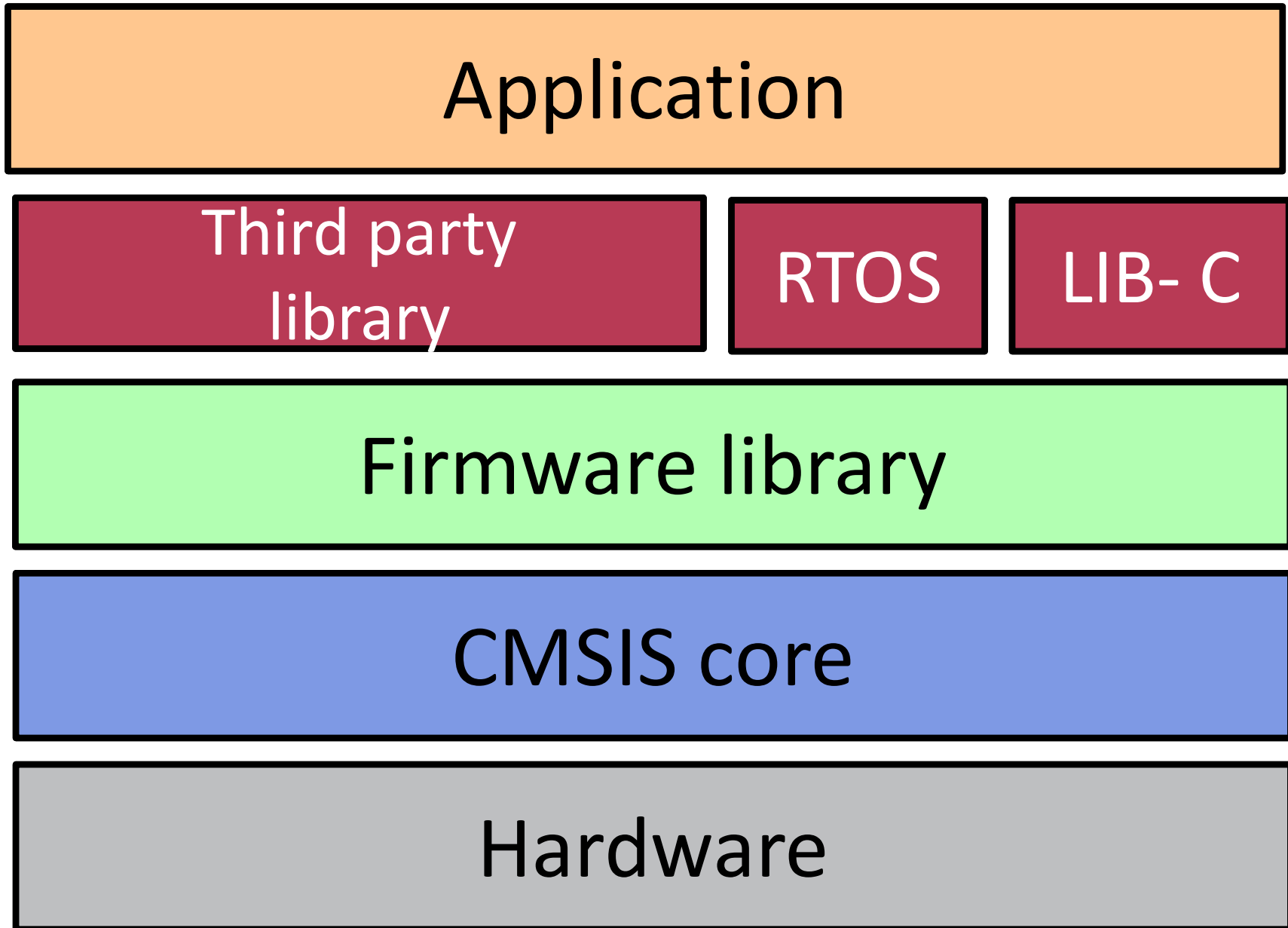

Csoportosítás, struktúra, hivatkozás

```
/** @brief I2C Init structure definition */
typedef struct
{
    uint32_t I2C_ClockSpeed;      /*!< Specifies the clock frequency */
    uint16_t I2C_Mode;           /*!< Specifies the I2C mode.
    This parameter can be a value of @ref I2C_mode */
} I2C_InitTypeDef;
```

```
/** @defgroup I2C_mode
@{
*/
#define I2C_Mode_MASTER          1
#define I2C_Mode_SLAVE          0
/**
@}
*/
```

A LIB-C elhelyezkedése a szoftver architektúrában

UART printf



UART printf

- Syscall minimal porting
 - iprintf, printf különbségek

```
60 __attribute__((used))
61 int _read(int file, char *ptr, int len)
62 {
63     return 0;
64 }
65 __attribute__((used))
66 int _write(int file, char *ptr, int len)
67 {
68     return len;
69 }
70 |
```