

UART kommunikáció

Tartalom

1	Bevezetés	3
1.1	USART/UART	3
1.2	Adat küldés és fogadás	4
1.3	UART használata AVR mikrovezérlővel.....	6
1.3.1	UDR regiszter	7
1.3.2	UCSRA státusz és kontroll regiszter.....	8
1.3.3	UCSRB státusz és kontroll regiszter.....	8
1.3.4	UCSRC státusz és kontroll regiszter	9
1.3.5	UBRR baud rate regiszter	9
2	Fogalomtár	10

1 Bevezetés

A beágyazott rendszerek világában, ugyan úgy, mint a „hagyományos” PC-k esetében is számos kommunikációs lehetőség létezik. Az adatküldést két csoportra lehet osztani: soros és párhuzamos kommunikációra. Az adatforgalom byte vagy byteokból álló adatcsomagok formájában jutnak el az adótól a vevőig.

Soros kommunikáció esetén az adat csomagok szekvenciálisan, egymás után bitről bitre kerülnek elküldésre ugyan azon az adatvezetéken. Ennek nagy előnye, hogy az egyes eszközök között elegendő 1db vezeték, így akár több méteres távolságra is könnyedén ki lehet építeni az adatkapcsolatot. Iparban számos soros kommunikációs protokollt alkalmaznak (RS-232/422/485).

Párhuzamos kommunikáció esetén 1db byte, bitekre felbontva egyszerre 8db adatvezetéken kerül kiküldésre. Tegyük fel, hogy 1bit kiküldése 1us ideig tart. Soros kommunikáció esetén, 1 byte átküldéséhez 8us idő szükséges, míg párhuzamos esetén a teljes byte 1us idő alatt eljut a küldőtől a fogadóig. Viszont a gyorsaságnak ára is van, mert 8db adatvezetékre van szükség, illetve további szinkronizáló/vezérlő vezetéket is be kell vonni a rendszerbe.

A továbbiakban a soros UART kommunikáció kerül bemutatásra.

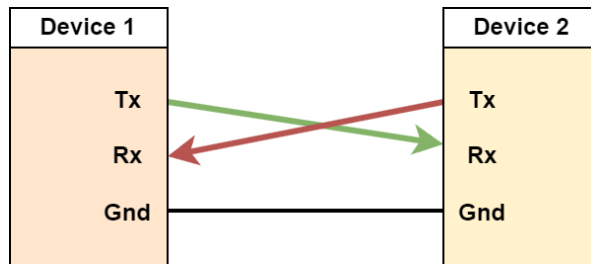
1.1 USART/UART

Szinte minden mikrovezérlőben megtalálható az úgynevezett USART/UART. Jelentése: Universal Synchronous Asynchronous Receiver Transmitter. Alap céláramkörként van megvalósítva a mikrovezérlőn belül, segítségével könnyedén kiépíthető egy másik mikrovezérlővel, számítógéppel, szenzorral stb. a soros kommunikáció.

Az alábbiakban az UART (Universal Asynchronous Receiver Transmitter) aszinkron kommunikációs protokoll kerül bemutatásra, mégpedig AVR mikrovezérlő esetén.

A legtöbb kommunikációs protokoll szinkronizált adatátvitelt valósít meg, azaz a kommunikációban résztvevő eszközök között valamilyen úton módon egy közös szinkronizáló órajelvezeték is ki van építve. (Megj.: Az USART az UART szinkron változata). Az aszinkron kommunikációnak meg van az előnye és a hátránya is. Előnye, hogy egy vezeték megspórolható, hátránya viszont az, hogy előre meghatározott frekvencián kell a küldőnek az adatot küldenie és ugyan olyan frekvencián kell a vevőnek a beérkező adatokat mintavételeznie! Az OSI – Open System Interconnection modellhez hasonlóan, a következő 5 réteget lehet kiemelni: Fizikai, elektromos, logikai, adat, alkalmazási réteg.

Az UART esetében a fizikai réteg összesen három vezetéket tartalmaz. Rx (fogadó), Tx (küldő) illetve GND (föld).



1.1. ábra: Két eszköz összekötése

Az elektromos rétegben definiált feszültségszint mikrovezérlőtől függően a legtöbbször 3.3V vagy 5V. Ebből következik, hogy a logikai rétegben a logikai 1-nek vagy 3.3V vagy 5V felel meg. Az adat rétegben kerül meghatározásra a kommunikációs sebesség, adatbitek és speciális bitek számát. Pl.: 9600 bps, 8 adat bit és 1 START és 1 STOP bit. Az alkalmazási réteget tekintve, az adatok a következő csomagban kerülnek elküldésre: START bit, adat, STOP bit. Az előzőek alapján egy adatcsomag 10 bitből áll.

1.2 Adat küldés és fogadás

A következőkben az alábbi paraméterekkel rendelkező kommunikáció kerül ismertetésre:

- átviteli sebesség 9600bps
- 1db START bit
- 1db STOP bit
- 8bit adat.

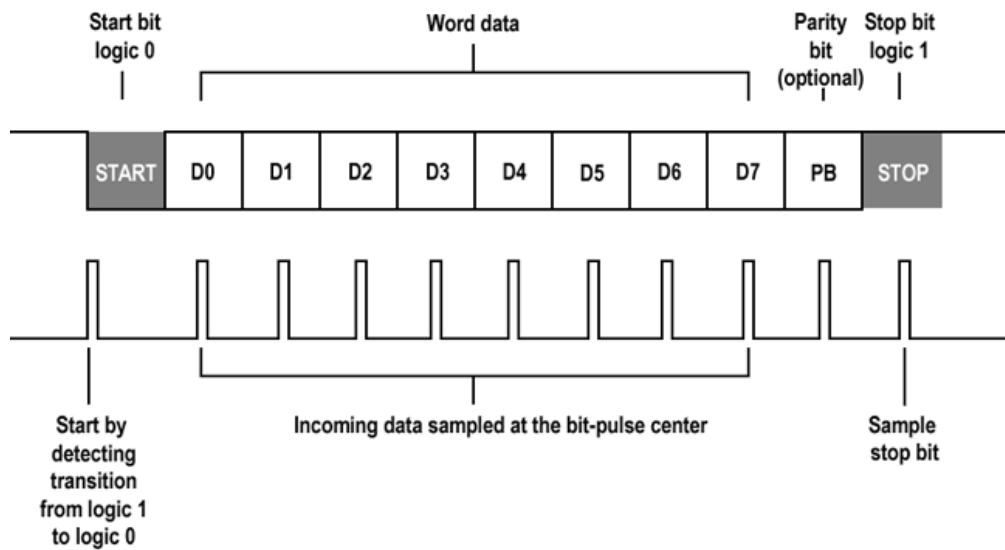
Aszinkron módban két eszköz között nincs közös órajel vezeték. Ahhoz, hogy a kommunikáció működjön a két eszköznek ugyan azt az átviteli sebességet (baudrate) kell a buszon használnia. 9600 bps esetén egy bit átküldéséhez $1/9600 = 104\mu s$ idő kell. Rossz baudrate megválasztása esetén a fogadó fél nem tudja megfelelő módon minta vételezni a beérkező biteket, hibás adatokat eredményezhet.

A protokollt tekintve az adatok keretbe beágyazva kerülnek továbbításra a fogadó fél számára. Egy standard keret a következőképpen épülhet fel:

1 START	DATA 0	DATA 1	DATA 2	DATA 3	DATA 4	DATA 5	DATA 6	DATA 7	1 STOP
---------	--------	--------	--------	--------	--------	--------	--------	--------	--------

Az adó elsőnek a STOP bitet küldi el. Ezzel jelzi a fogadó számára azt, hogy adatot szeretne küldeni, készüljön fel a fogadásra. Ezután következnek az adatbitek a legvégén pedig egy STOP bittel zárja a küldő a kommunikációt.

Mind a két eszköznek ugyan azt az adatformátumot kell használni, tehát 1 START, 8 adat és 1 STOP bit. Ezeknek a paramétereknek a beállítása a fejlesztő feladata!!



1.2. ábra: Adat küldés és fogadás

Az 1.2 ábrán látható az adat küldés és fogadás menete.

Adat küldés:

- Várakozó állapotban az adatvonal (Tx) logikai 1 szinten van, azaz 3.3V vagy 5V
- Küldéskor a küldő eszköz 9600bps esetén 104us ideig logikai 0 értékre állítja. Ez felel meg a START bitnek.
- A START bit kiküldése után következik a 8db ADAT bit.
- A kommunikáció egy STOP bit-el záródik, azaz a küldő eszköz legalább 104us ideig logikai 1 értéken tartja az adatvonalat. Több adatküldés esetén, a STOP bit után azonnal kiküldésre az újabb START bit.
- A kommunikáció befejeztével az adatvonal, az utolsó STOP bit után, logikai 1 értéken marad.

Adat fogadás:

- Várakozó állapotban a fogadó eszköz Rx vonala logikai 1 értéken van. Amikor beérkezik az első START bit, az Rx vonal 104us ideig logikai 0 értéket kap. A vevő ezt észleli és 9600bps esetén $1/9600/2=52\mu s$ után mintavételezi az Rx vonalat. Amennyiben a beérkező jel észlelését követően, 52us idő után is logikai 0 értéket olvas, a vevő megbizonyosodik arról, hogy ténylegesen adatküldés van.
- Ezután a vevő 104us időközönként mintavételezi az Rx vonalat, mindaddig amíg a STOP bit be nem érkezik.

Az aszinkron kommunikációnak tehát nagyon fontos paramétere a megfelelő átviteli sebesség beállítása! Belátható, hogy ha a fogadó eszköz másik baudrate értéket használ, akkor a fél periódus idővel való elcsúsztatás, mintavételezés nem fog megfelelően működni.

Az 1.2 ábrán látható egy PB (paritás) bit is. A fentiekben a legelterjedtebb 8N1 formátum került bemutatásra, de a következő bitek fordulhatnak elő:

- START bit: mindig 1db
- ADAT: lehet 5, 7, 8, 9 bit
- PARITÁS bit: mindig 1db
- STOP bit: 1 vagy 2db.

Szabványos baudrate értékeket a következő táblázat tartalmazza:

Baudrate	Bits/s	Bit duration	Speed
4800 bauds	4800 bits/s	208.333 μ s	600 bytes/s
9600 bauds	9600 bits/s	104.167 μs	1200 bytes/s
19200 bauds	19200 bits/s	52.083 μ s	2400 bytes/s
28800 bauds	28800 bits/s	34.722 μ s	3600 bytes/s
38400 bauds	38400 bits/s	26.042 μ s	4800 bytes/s
57600 bauds	57600 bits/s	17.361 μ s	7200 bytes/s
76800 bauds	76800 bits/s	13.021 μ s	9600 bytes/s
115200 bauds	115200 bits/s	8.681 μ s	14400 bytes/s

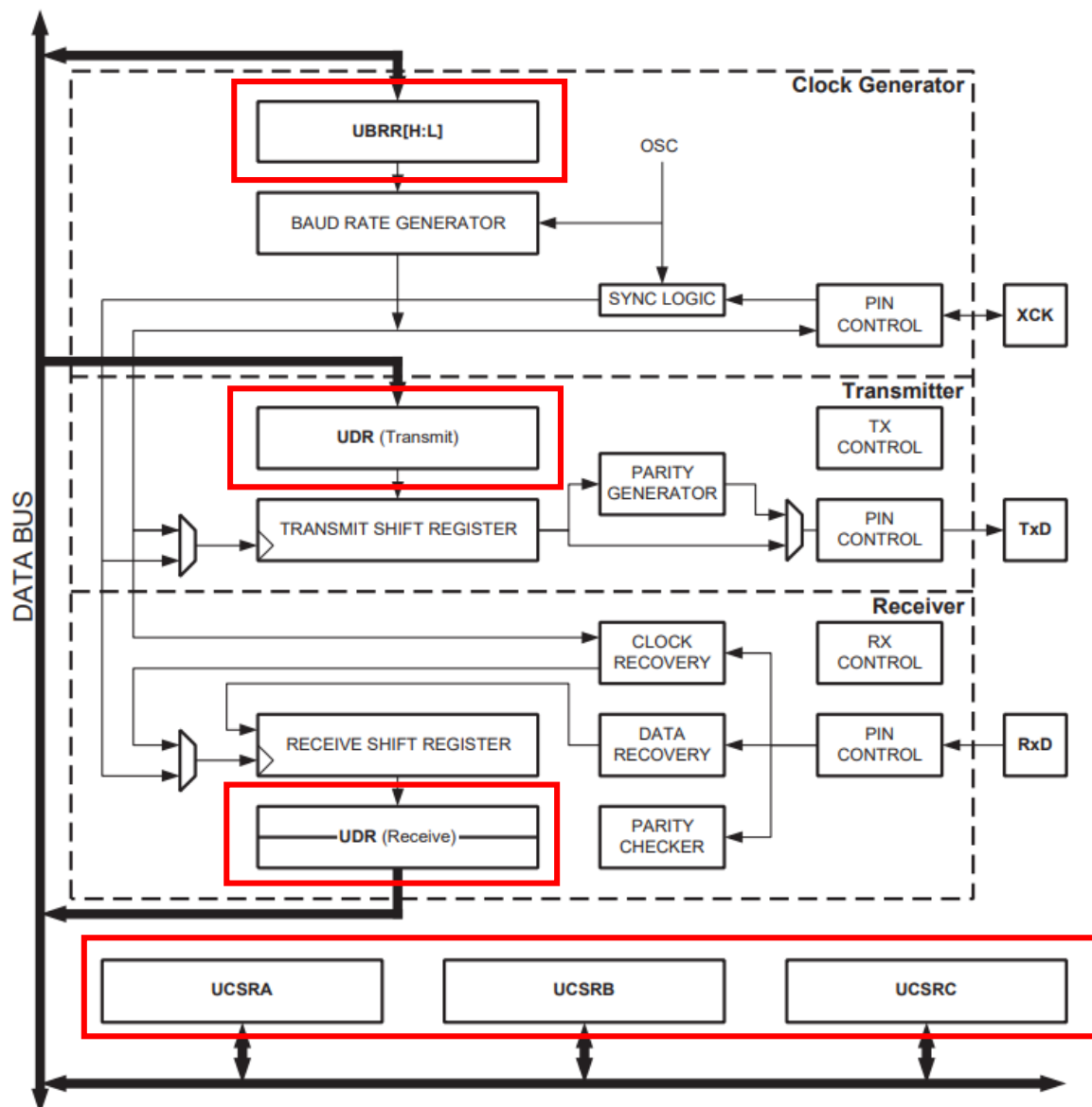
1.3 UART használata AVR mikrovezérlővel

Mint minden mikrovezérlő az AVR termékcsalád is rendelkezik soros kommunikációt elősegítő perifériával, USART-al. A protokollt tekintve, mivel nagyon egyszerű ezért könnyedén szoftveresen is implementálni lehet. A logikát leprogramozva, kisebb sebességgel, de egy GPIO porton keresztül is meg lehet valósítani a soros kommunikációt. A külön perifériának az az előnye, hogy a processzortól függetlenül lehet elvégezni az adat küldést és fogadást. Szoftverfejlesztés során a fejlesztőnek csak az adatok „adagolásával” kell foglalkoznia, a tényleges adat továbbítást az USART periféria végzi. Ugyan ez van fogadó oldalon is. Függetlenül a processzortól történik az adatok fogadása, majd miután minden bit beérkezett pl. egy megszakítás segítségével jelzi azt a processzor számára, hogy az adatot ki kell olvasni a megfelelő tárhelyből (regiszterből).

AVR mikrovezérlők adatlapját megtekintve, az alábbi 3 fő regiszter tartozik az USART-hoz:

- **UDR** (USART Data Register): adat regiszter
- **UCSRA, UCSRB, UCSRC** (USART Control and Status Register): kontrol és státusz regiszter
- **UBRRH** és **UBRRL** (USART Baud Rate Register): átviteli sebesség beállításához tartozó 16 bites (2x8 bit) regiszter.

A következő ábrán az ATmega2560 mikrovezérlő USART blokkdiagramja látható:

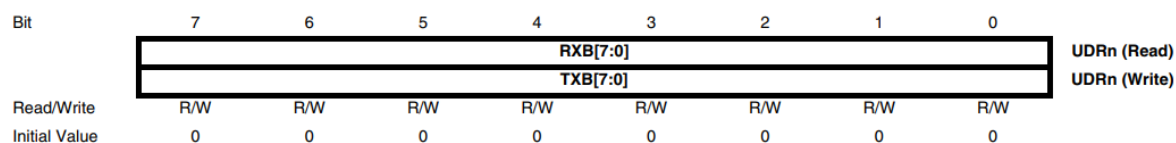


1.3. ábra: AtMega 2560 USART blokkdiagram

A következő alfejezetekben részletezésre kerülnek a regiszterek, illetve azok a konfigurációs bitek, amik a legelterjedtebb 8N1 framet használó, normál aszinkron üzemmódú UART kommunikáció beállításához szükségesek.

1.3.1 UDR regiszter

Az **UDR** regiszterből valójában, ahogy az az 1.3 ábrán is látható két darab van. Két 8 bites regiszterről van szó, egy a küldő egy pedig a fogadó egységben.



1.4. ábra: USART I/O Data Register

A két **UDR** buffer regiszter ugyan azon az I/O címen érhető el. Küldés esetén, az elküldendő byte-ot ebbe a regiszterbe kell a programozónak betölteni. Betöltés után átkerül egy küldő shiftregiszterbe, ahonnan bitről bitre kerül elküldésre az adat. Fogadáskor, a fogadott adat szintén egy fogadó shiftregiszterbe kerül bele. Ha minden adat beérkezett, akkor kerül áttöltésre az UDR regiszterbe, ahonnan már kiolvasható.

1.3.2 UCSRA státusz és kontroll regiszter

Bit	7	6	5	4	3	2	1	0	
	RXCn	TXCn	UDREn	FEn	DORn	UPEn	U2Xn	MPCMn	UCSRnA
Read/Write	R	R/W	R	R	R	R	R/W	R/W	
Initial Value	0	0	1	0	0	0	0	0	

1.5. ábra: UCSRA Control and Status Register

Az UCSRA regiszterben a következő két fontosabb paraméter beállításához tartozó bitek találhatóak:

- **RXCn** (Receive Complete): ez a bit jelzi (1-re billen) azt, hogy az UART beolvasta a beérkező adatot, a shiftregiszterből átkerült az UDR regiszterbe, azaz ki kell olvasni belől
- **TXCn** (Transmit Complete): ez a bit jelzi (1-re billen) azt, hogy az UART befejezte az adat küldést, azaz újabb adatot lehet betölteni az UDR regiszterbe.
- **UDRE** (Data Register Empty): ez a bit jelzi azt (1-re billen), hogy az UDR regiszter üres, azaz betölthető az új adat

1.3.3 UCSRB státusz és kontroll regiszter

Bit	7	6	5	4	3	2	1	0	
	RXCIEn	TXCIEn	UDRIEn	RXENn	TXENn	UCSZn2	RXB8n	TXB8n	UCSRnB
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Az **UCSRB** regiszterben a következő két fontosabb paraméter beállításához tartozó bitek találhatóak:

- **RXCIE** (RX Complete Interrupt Enable): Ezzel a bittel lehet engedélyezni azt a megszakítást, ami akkor váltódik ki, ha az adat fogadás befejeződött
- **TXCIE** (TX Complete Interrupt Enable): Ezzel a bittel lehet engedélyezni azt a megszakítást, ami akkor váltódik ki, ha az adat küldés befejeződött
- **RXEN** (Receiver enable): Vevő áramkör engedélyezése
- **TXEN** (Transmitter enable): Adó áramkör engedélyezése
- **UCSZ** (Character Size): Az UCSZ bitekkel lehet beállítani azt, hogy a frame-ben az adat hány bitből álljon (5, 6, 7, 8). Figyelem, további UCSZ bit található az UCSRC regiszterben!

1.3.4 UCSRC státusz és kontroll regiszter

Bit	7	6	5	4	3	2	1	0	
	UMSELn1	UMSELn0	UPMn1	UPMn0	USBSn	UCSZn1	UCSZn0	UCPOLn	UCSRnC
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	1	1	0	

Az **UCSRC** regiszterben a következő két fontosabb paraméter beállításához tartozó bitek találhatók:

- **UMSEL** (USART Mode Select): ezzel a bittel lehet kiválasztani, hogy szinkron vagy aszinkron üzemmódban működjön (aszinkron módban használjuk)
- **USBS** (Stop Bit Select): STOP bitek számát beállító bit (1 vagy 2 STOP bit lehet)
- **UPM** (Parity Mode): paritás módot beállító bit
- **UCSZ** (Character Size): Az UCSZ bitekkel lehet beállítani azt, hogy a frame-ben az adat hány bitből álljon (5, 6, 7, 8). Figyelem, további UCSZ bit található az UCSRB regiszterben!

1.3.5 UBRR baud rate regiszter

Bit	15	14	13	12	11	10	9	8	
	-	-	-	-	UBRR[11:8]				UBRRHn
	UBRR[7:0]								UBRRLn
	7	6	5	4	3	2	1	0	
Read/Write	R	R	R	R	R/W	R/W	R/W	R/W	
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	

Az **UBRR** regiszter 2db 8 bites regiszterből áll: **UBRRL**, **UBRRH**. Ennek a regiszternek a segítségével lehet, az úgynevezett előosztást beállítani. Az előosztás a rendszer fő órajeléből történik. A gyártó által biztosított adatlapban megtalálható, hogy a főórajelből, adott oszcillátor mellett milyen értéket kell betölteni az UBRR regiszterbe. Aszinkron normál üzemmódban, a következő egyenletet kell felhasználni:

$$baudrate = \frac{f_{osc}}{16(UBRR+1)} \quad (1)$$

ahol,

- baudrate: általunk kiválasztott baudrate érték (pl.: 9600)
- f_{osc} : mikroporcesszorhoz tartozó oszcillátor frekvenciája (pl.: 8000000UL = 8MHz),
- UBRR: ubrr regiszter értéke (ezt az értéket keressük)

Az (1) rendezzük úgy, hogy baloldalt az **UBRR** álljon:

$$UBRR = \frac{f_{osc}}{16baudrate} - 1 \quad (2)$$

2 Fogalomtár

Soros kommunikáció:

- az adat csomagok szekvenciálisan, egymás után bitről bitre kerülnek elküldésre ugyan azon az adatvezetéken.

Párhuzamos kommunikáció:

- 1 byte, bitekre felbontva egyszerre 8db adatvezetéken kerül kiküldésre.

Szinkronizált adatátvitel:

- a kommunikációban résztvevő eszközök között egy közös szinkronizáló órajelvezeték is ki van építve

Aszinkron adatátvitel:

- a kommunikációban résztvevő eszközök között nincs közös szinkronizáló órajelvezeték kiépítve

Előosztó:

- A főórajel leosztását lehet vele beállítani.

Baudrate:

- Átviteli sebesség, mértékegysége bit/s