



Katedra Inżynierii Systemów, Sygnałów i Elektroniki

Wydział Elektryczny

Zachodniopomorski Uniwersytet Technologiczny w Szczecinie

LABORATORIUM

TECHNIKA MIKROPROCESOROWA

**Port transmisji szeregowej USART
ATmega**

Opracował:
Tomasz Miłośławski

1. Cel ćwiczenia

Celem ćwiczenia jest zapoznanie się ze sposobami komunikacji mikrokontrolera rodziny AVR ATmega przez wbudowany port szeregowy USART (*ang. Universal Synchronous and Asynchronous serial Receiver and Transmitter*) z komputerem PC oraz innymi urządzeniami posiadającymi port szeregowy oraz jego konfiguracją i oprogramowaniem.

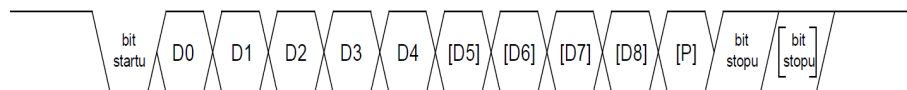
2. Charakterystyka portu szeregowego USART AVR ATmega

Wbudowany port szeregowy umożliwia przesyłanie informacji szeregowej w jednym z trzech trybów:

- **tryb asynchroniczny** – dwupiętowa szeregowo transmisja asynchroniczna przez linie **TxD** (tor nadawczy) i **RxD** (tor odbiorczy),
- **tryb synchroniczny** – dwupiętowa szeregowo transmisja synchroniczna przez linie **TxD** (tor nadawczy) i **RxD** (tor odbiorczy) synchronizowana sygnałem taktującym **XCK** w trybie nadrzędnym lub podrzędnym,
- **tryb wieloprocesorowy** – łączenie wielu mikrokontrolerów przez magistralę szeregową bazującą na liniach **TxD** i **RxD** z wyróżnionym układem nadrzędnym,

Prędkość transmisji we wszystkich trybach pracy jest ustalana programowo.

Ramka danych do transmisji składa się z bitu startu (synchronizującego transmisję), bitów (słowa) danych do transmisji (w kolejności od LSB do MSB), opcjonalnego bitu parzystości i jednego bitu lub dwóch bitów stopu. Słowo danych do transmisji może mieć długość 5, 6, 7, 8 lub 9 bitów.



Rys.1. Ramka danych USART AVR.

O konfiguracji portu szeregowego decydują ustawienia bitów w rejestrach **UCSRA**, **UCSRB** i **UCSRC**. Buforem danych jest rejestr **UDR**. Jego zapisanie inicjuje transmisję ramki danych. Odczyt odebranych danych z **UDR** powoduje skasowanie zawartości bufora.

- UCSRA. 7** – **RXC** – znacznik odebrania danych
- UCSRA. 6** – **TXC** – znacznik przesłania danych
- UCSRA. 5** – **UDRE** – znacznik gotowości rejestru UDR
- UCSRA. 4** – **FE** – znacznik błędu ramki transmisyjnej
- UCSRA. 3** – **DOR** – znacznik przepełnienia bufora wejściowego
- UCSRA. 2** – **PE** – znacznik błędu parzystości
- UCSRA. 1** – **U2X** – podwojenie prędkości transmisji
- UCSRA. 0** – **MPCM** – tryb komunikacji wieloprocesorowej

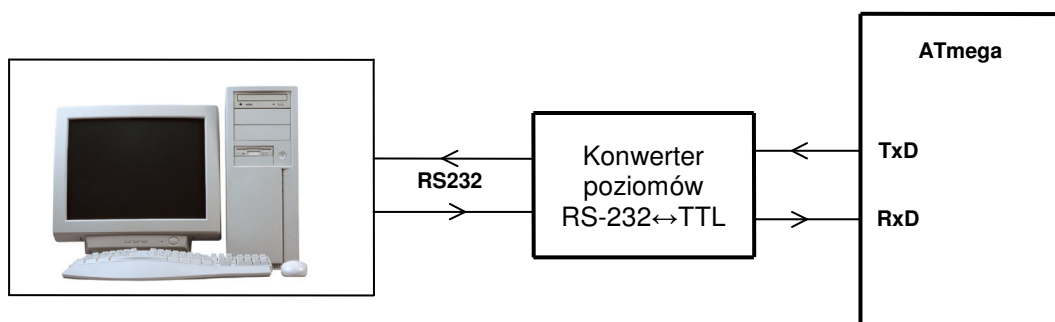
- UCSRB. 7 - RXCIE** – włączenie obsługi przerwania odbiornika
- UCSRB. 6 - TXCIE** – włączenie obsługi przerwania nadajnika
- UCSRB. 5 - UDRIE** – włączenie obsługi przerwania UDRE
- UCSRB. 4 - RXEN** – włączenie odbiornika
- UCSRB. 3 - TXEN** – włączenie nadajnika
- UCSRB. 2 - UCSZ2** – długość słowa w ramce danych
- UCSRB. 1 - RXB8** – bit rozszerzenia rejestru UDR odbiornika
- UCSRB. 0 - TXB8** – bit rozszerzenia rejestru UDR nadajnika

- UCSRC. 7 - URSEL** – wybór rejestru
- UCSRC. 6 - UMSEL** – wybór trybu pracy
- UCSRC. 5 - UPM1** – włączenie bitu parzystości
- UCSRC. 4 - UPM0** – typ bitu parzystości
- UCSRC. 3 - USBS** – liczba bitów stopu
- UCSRC. 2 - UCSZ1** – długość słowa w ramce danych
- UCSRC. 1 - UCSZ0** – długość słowa w ramce danych
- UCSRC. 0 - UCPOL** – polaryzacja sygnału taktującego

Wybór długości słowa jest ustawiany za pomocą konfiguracji bitów UCSZ2..0, co przedstawia Tabela 1:

Tabela 1

UCSZ2	UCSZ1	UCSZ0	Długość słowa w ramce danych
0	0	0	5-bitowe
0	0	1	6-bitowe
0	1	0	7-bitowe
0	1	1	8-bitowe
1	1	1	9-bitowe



Rys. 2. Połączenie mikrokontrolera ATmega z komputerem PC za pomocą interfejsu RS232C w trybie asynchronicznym.

3. Określenie prędkości transmisji portu szeregowego USART AVR

Do określenia prędkości transmisji służą rejestry UBRRH i UBRL, w których zapisana jest wartość UBRR. Tabela 2 zawiera wzory pozwalające dobrać wartość UBRR dla różnych trybów pracy portu szeregowego.

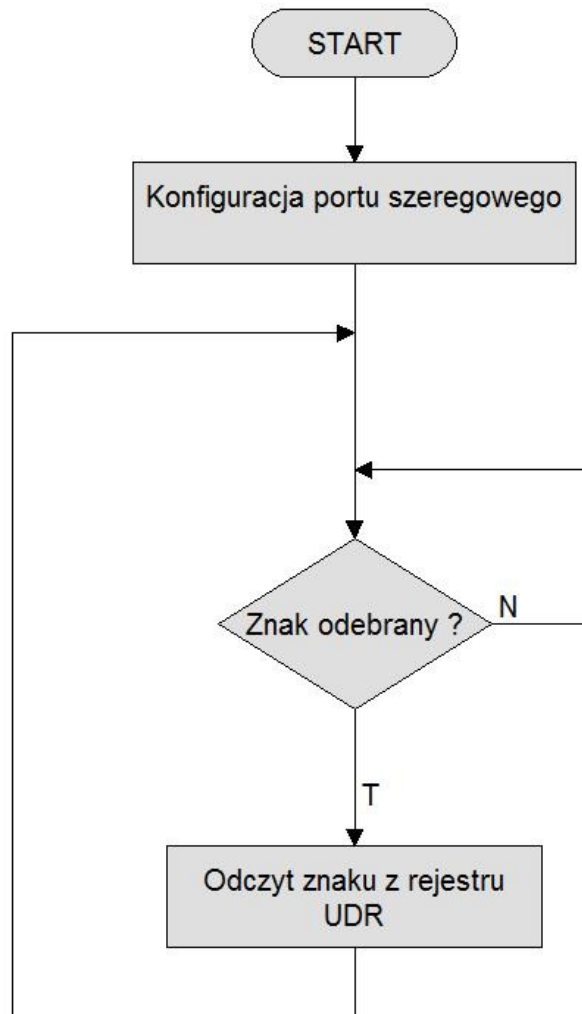
Tabela 2

Tryby pracy USART	Wzór na wartość UBRR	Wzór na prędkość transmisji
Asynchroniczny (UMSEL = 0 i U2X = 0)	$UBRR = f_{osc} / (16 * BAUD) - 1$	$BAUD = f_{osc} / (16 * (UBRR + 1))$
Asynchroniczny podwójnej prędkości (UMSEL = 0 i U2X = 1)	$UBRR = f_{osc} / (8 * BAUD) - 1$	$BAUD = f_{osc} / (8 * (UBRR + 1))$
Synchroniczny (UMSEL = 1 i U2X = 0)	$UBRR = f_{osc} / (2 * BAUD) - 1$	$BAUD = f_{osc} / (2 * (UBRR + 1))$

4. Ćwiczenia programowe

Poniżej przedstawiono przykładowy programy komunikacji zestawu dydaktycznego ZL3AVR z komputerem PC za pomocą łącza pracującego w standardzie RS232 przez złącze COM1 lub COM2 z wykorzystaniem programu Hyper Terminal. Port szeregowy mikrokontrolera ATmega pracuje w trybie asynchronicznym.

Program „odbiera” znaki transmitowane z komputera PC do modułu ZL3AVR poprzez wpisywanie ich na klawiaturze w programie Hyper Terminal. Odebrany znak wyświetlany jest w postaci binarnej za pomocą diod LED sterowanych przez PORTA. Algorytm programu przedstawiono poniżej:



Rys.3. Algorytm programu odbierającego znak.

```

#include <avr/io.h>
#include <util/delay.h>

#define UART_BAUD_RATE 9600 /* 9600 baud */
#define UART_BAUD_SELECT (F_CPU/(UART_BAUD_RATE*16l)-1)

int main(void)
{
  DDRA = 0xff;
  UCSRB = (1 << RXEN) | (1 << TXEN);
  UCSRC = (1 << URSEL) | (1 << UCSZ1) | (1 << UCSZ0);
  /* set baud rate to 9600 */
  UBRRH = (unsigned char) (UART_BAUD_SELECT >> 8);
  UBRRL = (unsigned char) (UART_BAUD_SELECT);

  while (1)
  {
    while (!(UCSRA & (1 << RXC)));
    PORTA = UDR;
  }
}

```

ZADANIE 1

Napisać program odbierający pojedynczy znak z terminala komputera PC i odsyłający ten sam znak z powrotem do terminala, tzw. echo. Prędkość transmisji 2400bit/s, tryb asynchroniczny portu szeregowego. Do wysyłania znaków z klawiatury komputera PC należy wykorzystać program Hyper Terminal.

ZADANIE 2

Napisać program dekodujący znaki będące kodami ASCII przycisków 0, 1, 2, 3, 4, 5, 6, 7 klawiatury komputera PC przesyłane za pomocą łącza RS-232 do zestawu dydaktycznego ZL3AVR. Stan wciśniętego przycisku powinien być sygnalizowany za pomocą odpowiedniej diody LED podłączonej do portu PORTA (0 - PA0, 1 - PA1, 2 - PA2, 3 - PA3, 4 - PA4, 5 - PA5, 6 - PA6, 7 - PA7) w zestawie dydaktycznym. Prędkość transmisji 4800bit/s, tryb asynchroniczny portu szeregowego.

ZADANIE 3

Napisać program odbierający i wysyłający komunikat wieloznakowy na zasadzie „echa” z sumą kontrolną EX-OR. Komunikat powinien mieć następujący format:

\$abc*[CHSH][CHSL][CR][LF]

gdzie:

\$	- jeden znak ASCII, znacznik początku komunikatu
abc	- trzy znaków ASCII, treść komunikatu,
*	- jeden znak ASCII, znacznik końca treści komunikatu
[CHSH][CHSL]	- dwa znaki ASCII reprezentujące 8-bitową liczbę zapisaną w kodzie heksadecymalnym, będącą sumą kontrolną EX-OR,
[CR]	- pierwszy znacznik końca komunikatu,
[LF]	- drugi znacznik końca komunikatu.

Suma kontrolna liczona jest w następujący sposób:

$$\text{CHS} = 0 \text{ xor } a \text{ xor } b \text{ xor } c.$$

Do wysyłania komunikatu należy wykorzystać program Hyper Terminal – wysyłanie pliku. Prędkość transmisji 4800bit/s, tryb asynchroniczny portu szeregowego.

5. Opis sygnałów gniazda DB9 łącza RS232C

1 – DCD- Data Carrier Detected	- sygnał wykrycia nośnej
2 – RXD- Receive Data	- odbiór danych
3 – TXD- Transmit Data	- transmisja danych
4 – DTR - Data Terminal Ready	- przetworzono dane (gotowość odbioru)
5 – GND - Signal Ground	- masa
6 – DSR - Data Set Ready	- wypełniony bufor (gotowość transmisji)
7 – RTS - Request to Send Data	- żądanie wysyłania
8 – CTS - Clear to Send Data	- gotowość wysyłania
9 – RI - Ring indicator	- wskaźnik dzwonka

6. Tablica kodów ASCII (American Standard Code for Information Interchange)

Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	NUL (null)	32	20	040	 	Space	64	40	100	@	@	96	60	140	`	`
1	1	001	SOH (start of heading)	33	21	041	!	!	65	41	101	A	A	97	61	141	a	a
2	2	002	STX (start of text)	34	22	042	"	"	66	42	102	B	B	98	62	142	b	b
3	3	003	ETX (end of text)	35	23	043	#	#	67	43	103	C	C	99	63	143	c	c
4	4	004	EOT (end of transmission)	36	24	044	$	\$	68	44	104	D	D	100	64	144	d	d
5	5	005	ENQ (enquiry)	37	25	045	%	%	69	45	105	E	E	101	65	145	e	e
6	6	006	ACK (acknowledge)	38	26	046	&	&	70	46	106	F	F	102	66	146	f	f
7	7	007	BEL (bell)	39	27	047	'	'	71	47	107	G	G	103	67	147	g	g
8	8	010	BS (backspace)	40	28	050	((72	48	110	H	H	104	68	150	h	h
9	9	011	TAB (horizontal tab)	41	29	051))	73	49	111	I	I	105	69	151	i	i
10	A	012	LF (NL line feed, new line)	42	2A	052	*	*	74	4A	112	J	J	106	6A	152	j	j
11	B	013	VT (vertical tab)	43	2B	053	+	+	75	4B	113	K	K	107	6B	153	k	k
12	C	014	FF (NP form feed, new page)	44	2C	054	,	,	76	4C	114	L	L	108	6C	154	l	l
13	D	015	CR (carriage return)	45	2D	055	-	-	77	4D	115	M	M	109	6D	155	m	m
14	E	016	SO (shift out)	46	2E	056	.	.	78	4E	116	N	N	110	6E	156	n	n
15	F	017	SI (shift in)	47	2F	057	/	/	79	4F	117	O	O	111	6F	157	o	o
16	10	020	DLE (data link escape)	48	30	060	0	0	80	50	120	P	P	112	70	160	p	p
17	11	021	DC1 (device control 1)	49	31	061	1	1	81	51	121	Q	Q	113	71	161	q	q
18	12	022	DC2 (device control 2)	50	32	062	2	2	82	52	122	R	R	114	72	162	r	r
19	13	023	DC3 (device control 3)	51	33	063	3	3	83	53	123	S	S	115	73	163	s	s
20	14	024	DC4 (device control 4)	52	34	064	4	4	84	54	124	T	T	116	74	164	t	t
21	15	025	NAK (negative acknowledge)	53	35	065	5	5	85	55	125	U	U	117	75	165	u	u
22	16	026	SYN (synchronous idle)	54	36	066	6	6	86	56	126	V	V	118	76	166	v	v
23	17	027	ETB (end of trans. block)	55	37	067	7	7	87	57	127	W	W	119	77	167	w	w
24	18	030	CAN (cancel)	56	38	070	8	8	88	58	130	X	X	120	78	170	x	x
25	19	031	EM (end of medium)	57	39	071	9	9	89	59	131	Y	Y	121	79	171	y	y
26	1A	032	SUB (substitute)	58	3A	072	:	:	90	5A	132	Z	Z	122	7A	172	z	z
27	1B	033	ESC (escape)	59	3B	073	;	;	91	5B	133	[[123	7B	173	{	{
28	1C	034	FS (file separator)	60	3C	074	<	<	92	5C	134	\	\	124	7C	174	|	
29	1D	035	GS (group separator)	61	3D	075	=	=	93	5D	135]]	125	7D	175	}	}
30	1E	036	RS (record separator)	62	3E	076	>	>	94	5E	136	^	^	126	7E	176	~	~
31	1F	037	US (unit separator)	63	3F	077	?	?	95	5F	137	_	_	127	7F	177		DEL

Źródło: www.LookupTables.com

Tablica kodów ASCII rozszerzona

128	Ç	144	É	161	í	177	⌘	193	⌞	209	⌠	225	β	241	±
129	ü	145	æ	162	ó	178	⌘	194	⌟	210	⌡	226	Γ	242	≥
130	é	146	Æ	163	ú	179		195	⌠	211	⌢	227	π	243	≤
131	â	147	ô	164	ñ	180	⌡	196	—	212	⌣	228	Σ	244	∫
132	ä	148	ö	165	Ñ	181	⌢	197	+	213	⌤	229	σ	245	∫
133	à	149	ò	166	ª	182	⌣	198	⌡	214	⌥	230	μ	246	+
134	â	150	û	167	º	183	⌤	199	⌢	215	⌦	231	τ	247	±
135	ç	151	ù	168	¿	184	⌥	200	⌣	216	⌧	232	Φ	248	°
136	ê	152	—	169	—	185	⌦	201	⌤	217	⌨	233	⊖	249	.
137	ë	153	Ö	170	¬	186	⌧	202	⌥	218	〈	234	Ω	250	.
138	è	154	Û	171	½	187	⌨	203	⌦	219	■	235	δ	251	√
139	ï	156	£	172	¼	188	〈	204	⌧	220	■	236	∞	252	—
140	î	157	₣	173	¡	189	〉	205	=	221	■	237	φ	253	²
141	ì	158	—	174	«	190	⌫	206	⌨	222	■	238	ε	254	■
142	Ä	159	ƒ	175	»	191	⌬	207	⌞	223	■	239	∩	255	
143	Å	160	á	176	⌘	192	⌭	208	⌟	224	α	240	≡		

Źródło: www.LookupTables.com