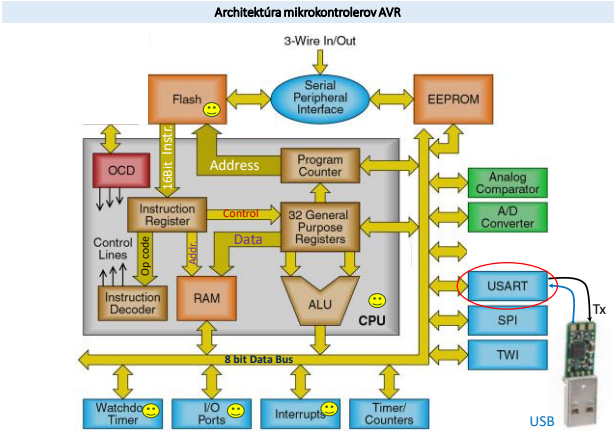


Týždeň	Prednášky	Cvičenia
1.	Úvod do predmetu-Číselné sústavy v elektronike; kódovanie čísel- Jazyk C- Úvod do digitálnej elektroniky.	Instalácia:— IDE Microchip Studio –driver pre Arduino- kit
2.	Polovodičové pamäte	Práca s Microchip Studio, definovanie projektu
3.	Atmega328; pamäť, generovanie hodinových signálov; manažment napájania; reset	Práca s Microchip Studio preklad; simulácia projektu
4.	Vstupy-výstupy	Práca so vstupmi a výstupmi
5.	Externé a interné prerušenia; Watchdog časovač	Práca s prerušeniami Výber zadania na zápočet
6.	Sériové komunikačné prostriedky USART – Tx	Práca s USART – TX. Práca na zadani
7.	Sériové komunikačné prostriedky USART - Rx	Práca s USART – RX. Práca na zadani
8.	Časovače – počítadlá	Práca s TCO. Práca na zadani
9.	Časovače – počítadlá	Práca s TCO. Práca na zadani
10.	A/D prevodník	Programovanie A/D. Práca na zadani
11.	EEPROM, FUSE bity	Programovanie EEPROM. Práca na zadani
12.	I2C zbernica	Zápočet - Praktická práca s MCU
13.	Predtermín	Zápočet - Praktická práca s MCU

Základy mikroprocesorovej techniky

1



Základy mikroprocesorovej techniky

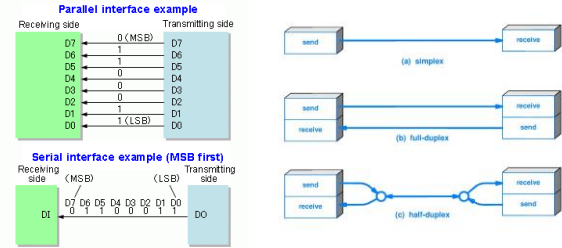
2

ATmega328P – sériové komunikačné prostriedky (90-104)

Prepojenie elektronických obvodov, medzi ktorými sa prenášajú dáta, môže byť realizované pomocou paralelnej alebo sériovej zbernice.

Paralelná zbernica sa skladá z viacerých paralelných vodičov (8, 16, 24, 32), po ktorých sa prenášajú dáta od vysielateľa k prijímaču. Komunikácia je synchronná, pričom synchronizáciu vysielateľa a prijímateľa zabezpečujú riadiace signály. Paralelná komunikácia umožňuje prenos viacerých bitov súčasne.

Sériová zbernica je tvorená jedným (Simplex, Half-Duplex) alebo dvoma (Full-Duplex) dátovými vodičmi. Sériová komunikácia znamená prenos dát po jednotlivých bitoch za sebou.

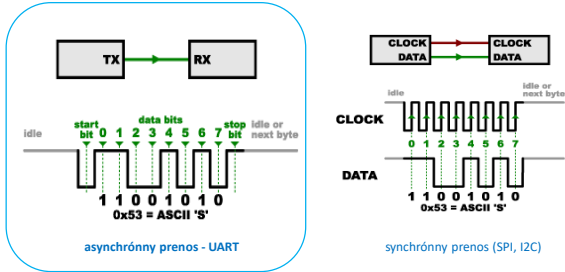


Základy mikroprocesorovej techniky

3

ATmega328P – sériové komunikačné prostriedky

- Prenos na sériovej zbernici môže byť:
- **asynchronný** - bez použitia synchronizačného signálu, pričom riadenie sa vykonáva pomocou riadiacich bitov, ktoré sa prenášajú spoločne s dátovými bitmi,
 - **synchronný** - s použitím synchronizačného signálu (hodinového signálu).



Základy mikroprocesorovej techniky

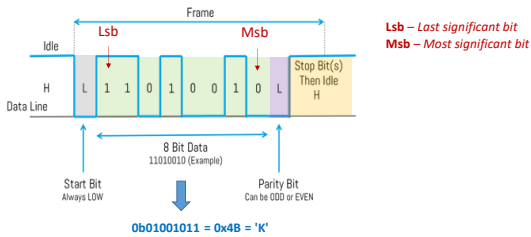
4

Asynchrónny sériový prenos – Frame Formats

Asynchrónny prenos dát sa realizuje vysielaním resp. prijímaním tzv. rámcov (frame) s vopred nastavenou prenosovou rýchlosťou.

Frame je tvorený skupinou dátových bitov, synchronizačnými bitmi (štart, stop) a voľiteľným paritným bitom na kontrolu chyby prenosu:

- 1 **štartovací bit** – vždy log. 0,
- 5, 6, 7, 8 alebo 9 **dátových bitov**,
- žiadny, párny alebo nepárny **paritný bit**. Parita je kontrolný mechanizmus na určenie integrity dátových bitov a vyjadruje, či je počet log. 1 v dátových bitoch párny alebo nepárny,
- 1 alebo 2 **stop bity** – vždy log. 1.



Základy mikroprocesorovej techniky

5

ATmega328P – Univerzálny Synchronný/Asynchrónny Prijímač/Vysielač – USART (24.)

UDR0 – USART Data Register 0

UCSRA, UCSRB, UCSRC
USART Control and Status Register 0A, (0B, 0C)

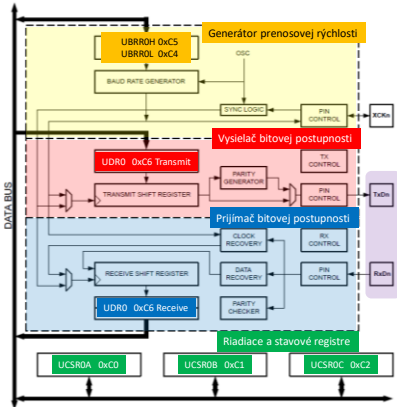
UBRR0L – USART Baud Rate Register 0 Low

UBRR0H – USART Baud Rate Register 0 High

UBRR0 = UBR0H * 0x100 + UBR0L

Režimy USART-u a určenie prenosovej rýchlosti: (Table 24-1)

- **Asynchrónny normálny mód**
$$\text{Baud} = \frac{f_{osc}}{16 \cdot (UBRR0 + 1)}$$
- **Asynchrónny 2x rýchlosť mód**
$$\text{Baud} = \frac{f_{osc}}{8 \cdot (UBRR0 + 1)}$$
- **Synchronný Master SPI mód (MSPIM)**
$$\text{Baud} = \frac{f_{osc}}{2 \cdot (UBRR0 + 1)}$$



Základy mikroprocesorovej techniky

6

ATmega328P – režim UART – Frame Formats (24.5)

Nastavenie parametrov sériového prenosu po resete MCU (UCSROC, UCSROB):

- Asynchrónny prenos (UART).
- Žiadna parita.
- Jeden stop bit.
- 8 – dátových bitov.
- UCSRO.UCPOL=0 v asynchrónnom móde.

Name: UCSROC
Offset: 0xC2
Reset: 0x06
Property: -
USART Control and Status Register 0 C

Bit	7	6	5	4	3	2	1	0
	UMIELO1	UMIELO0	UPM01	UPM00	USBS0	UCSZ01 / UDORD0	UCSZ00 / UCPH00	UCPOL0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	1	1	0

Name: UCSROB
Offset: 0xC1
Reset: 0x00
Property: -
USART Control and Status Register 0 B

Bit	7	6	5	4	3	2	1	0
	RXCIF0	TXCIF0	UDRE0	TXDR0	TXEN0	UDRE0	RXCIF0	TXDR0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Základy mikroprocesorovej techniky

7

ATmega328P – režim UART – Inicializácia (24.6)

USART sa musí najprv inicializovať, až potom je možné uskutočniť komunikáciu. Postup je nasledovný:

- Nastavenie režimu USART a frame formát v registri UCSROC,
- Nastavenie prenosovej rýchlosti v registri UBRR0 v bitoch za sekundu (baud),
- Povolenie vysielanie dát (Tx) a (alebo) prijímania dát (Rx) v registri UCSROB,
- Povolenie / zakázanie prerušenie MCU od činnosti USART v registri UCSROB.

$$Baud = \frac{f_{osc}}{16 * (UBRR0 + 1)}$$

Bit 7 – RXCIE0: RX Complete Interrupt Enable
Bit 6 – TXCIE0: TX Complete Interrupt Enable
Bit 5 – UDRE0: USART Data Register Empty Interrupt Enable
Bit 4 – RXEN0: Receiver Enable
Bit 3 – TXEN0: Transmitter Enable
Bit 2 – UCSZ02: Character Size
Bit 1 – RXB80: Receive Data Bit 8
Bit 0 – TXB80: Transmit Data Bit 8

Name: UCSROB
Offset: 0xC1
Reset: 0x00
Property: -
USART Control and Status Register 0 B

Bit	7	6	5	4	3	2	1	0
	RXCIF0	TXCIF0	UDRE0	RXCIF0	TXCIF0	UDRE0	RXCIF0	TXCIF0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

9 bitový prenos

Zmena funkcie vývodov PD1 a PD0 na Tx a Rx

Základy mikroprocesorovej techniky

8

ATmega328P – režim UART – Inicializácia (24.6)

USART sa musí najprv inicializovať, až potom je možné uskutočniť komunikáciu. Postup je nasledovný:

- Nastavenie režimu USART a frame formát v registri UCSROC,
- Nastavenie prenosovej rýchlosti v registri UBRR0 v bitoch za sekundu (baud),
- Povolenie vysielanie dát (Tx) a (alebo) prijímania dát (Rx) v registri UCSROB,
- Povolenie / zakázanie prerušenie MCU od činnosti USART v registri UCSROB.

```
#define FOSC 16000000UL
#define BAUD 9600
#define UBRR FOSC/16/BAUD-1

void USART_Init (unsigned int ubrr)
{
    // set baud rate
    UBRRH = (unsigned char)(ubrr>>8);
    UBRRL = (unsigned char) ubrr;
    // Enable receiver and transmitter;
    UCSRB = (1 << RXEN0) | (1 << TXEN0);
    //Set Mode UART, Frame format: 8data, 1stop
    UCSRC = (1 << UCSZ01) | (1 << UCSZ00);
}

int main (void) {
    USART_Init (UBRR);
    while(1) {
    }
```

$$Baud = \frac{f_{osc}}{16 * (UBRR0 + 1)}$$

$$UBRR0 = \frac{f_{osc}}{16 * baud} - 1$$

Zmena funkcie vývodov PD1 a PD0 na Tx a Rx

Základy mikroprocesorovej techniky

9

ATmega328P – režim UART – prenosová rýchlosť (24.11)

Register UBRR0[11:0] je 12-bitový register určený na nastavenie prenosovej rýchlosti. Volíme rýchlosť s najmenšou chybou. Pre vývojovú dosku Arduino UNO (f_{XTAL} = 16MHz) volíme prenosovú rýchlosť napr. 9600 baud. (Table 24-7)

Baud Rate [bps]	f _{osc} = 16.0000MHz				f _{osc} = 18.4320MHz				f _{osc} = 20.0000MHz			
	UBRRn	Error	UBRRn	Error	UBRRn	Error	UBRRn	Error	UBRRn	Error	UBRRn	Error
2400	416	-0.1%	832	0.0%	479	0.0%	959	0.0%	520	0.0%	1041	0.0%
4800	207	0.2%	416	-0.1%	239	0.0%	479	0.0%	259	0.2%	520	0.0%
9600	103	0.2%	207	0.2%	119	0.0%	239	0.0%	129	0.2%	259	0.2%
14.4k	68	0.6%	138	-0.1%	79	0.0%	159	0.0%	86	-0.2%	173	-0.2%
19.2k	51	0.2%	103	0.2%	59	0.0%	119	0.0%	64	0.2%	129	0.2%
28.8k	34	-0.8%	68	0.6%	39	0.0%	79	0.0%	42	0.9%	86	-0.2%
38.4k	25	0.2%	51	0.2%	29	0.0%	59	0.0%	32	-1.4%	64	0.2%
57.6k	16	2.1%	34	-0.8%	19	0.0%	39	0.0%	21	-1.4%	42	0.9%
76.8k	12	0.2%	25	0.2%	14	0.0%	29	0.0%	15	1.7%	32	-1.4%
115.2k	8	-3.5%	16	2.1%	9	0.0%	19	0.0%	10	-1.4%	21	-1.4%
230.4k	3	8.5%	8	-3.5%	4	0.0%	9	0.0%	4	8.5%	10	-1.4%
250k	3	0.0%	7	0.0%	4	-7.8%	8	2.4%	4	0.0%	9	0.0%
0.5M	1	0.0%	3	0.0%	-	-	4	-7.8%	-	4	0.0%	
1M	0	0.0%	1	0.0%	-	-	-	-	-	-	-	
Max(1)	1Mbps		2Mbps		1.152Mbps		2.304Mbps		1.25Mbps		2.5Mbps	

Základy mikroprocesorovej techniky

10

ATmega328P – režim UART – vysielanie cez vývod Tx (24.7) (24.7.1)

- Vysielanie dát cez vývod Tx (PD1) sa povoľuje nastavením bitu vysielania UCSROB.TXEN.
- Vysielanie dát cez vývod Tx sa začne zápisom dátového bajtu do registra UDRO. Pred zápisom do UDRO je potrebné overiť, či je tento register prázdny, testovaním bitu UCSROA.UDRE0.
- Posuvný register zabezpečuje odovšielanie jednotlivých bitov podľa parametrov prenosu.
- Ukončenie vysielania bajtu je signalizované bitom UCSROA.TXC0.

```
void USART_Transmit (unsigned char data)
{
    // Wait for empty transmit buffer
    while( !(UCSRA & (1<<UDRE0)) );
    // Put data into buffer, send data
    UDRO = data;
}
```

Name: UCSROA
Offset: 0xC0
Reset: 0x00
Property: -
USART Control and Status Register 0 A

Bit	7	6	5	4	3	2	1	0
	RXCIF0	TXCIF0	UDRE0	TXDR0	UDRE0	UDRE0	UDRE0	UDRE0
Access	R	R/W	R/W	R	R	R/W	R/W	R/W
Reset	0	0	1	0	0	0	0	0

Základy mikroprocesorovej techniky

11

ASCII tabuľka - vysielanie jednotlivých znakov (8 bitov - bajt)

Kódovanie znakov klávesnice anglickej abecedy s rozlíšením veľkých aj malých písmen, vrátane čísiel a špeciálnych znakov popisuje 7-bitový kód ASCII (American Standard Code for Information Interchange).

Tabuľku kódov tvoria tri časti:

- Riadiace kódy (0x00-0x1F, 0x7F) na riadenie komunikácie (CR, LF, STX, ETX ...)
- Kódy pre písmená a čísla,
- Špeciálne znaky.

```
#define LF 0x0A (line feed-posun o riadok)
#define CR 0x0D (carriage return-navrat vozika)
#define STX 0x02 (start text)
#define ETX 0x03 (end text)

void USART_Transmit (unsigned char data);

int main (void) {
    USART_Transmit (0x41);
    USART_Transmit ('A');
    USART_Transmit (LF);
    USART_Transmit (CR);
}
```

Base-16	ASCII	Base-16	ASCII	Base-16	ASCII	Base-16	ASCII
00	NULL	20	Space	40	@	60	a
01	SOH	21	!	41	A	61	b
02	STX	22	"	42	B	62	c
03	ETX	23	#	43	C	63	d
04	EOF	24	\$	44	D	64	e
05	ENQ	25	%	45	E	65	f
06	ACK	26	&	46	F	66	g
07	BEL	27	'	47	G	67	h
08	BS	28	(	48	H	68	i
09	EM	29	)	49	I	69	j
0A	LF	2A	*	4A	J	6A	k
0B	VT	2B	+	4B	K	6B	l
0C	FF	2C	,	4C	L	6C	m
0D	CR	2D	-	4D	M	6D	n
0E	SO	2E	.	4E	N	6E	o
0F	SI	2F	/	4F	O	6F	p
10	DLE	30	:	50	P	70	q
11	DC1	31	;	51	Q	71	r
12	DC2	32	<	52	R	72	s
13	DC3	33	=	53	S	73	t
14	DC4	34	>	54	T	74	u
15	NAK	35	?	55	U	75	v
16	SYN	36	@	56	V	76	w
17	ETB	37	[	57	W	77	x
18	CAN	38	\	58	X	78	y
19	EM	39	]	59	Y	79	z
1A	SUB	3A	^	5A	Z	7A	{
1B	ESC	3B	_	5B	[	7B	
1C	FS	3C	`	5C	\	7C	}
1D	GS	3D	~	5D	]	7D	~
1E	RS	3E	>	5E	^	7E	-
1F	US	3F	?	5F	_	7F	DEL

Základy mikroprocesorovej techniky

12

Vysielanie textového refazca (n - bajtov)

```
#include <string.h>
#include <stdio.h>

#define LF 0x0A (line feed-posun o riadok)
#define CR 0x0D (carriage return-navrat vozika)
#define STX 0x02 (start text)
#define ETX 0x03 (end text)
//-----
void USART_Transmit (unsigned char data)
{
    // Wait for empty transmit buffer
    while( !(UCSR0A & (1<< UDRE0)) );
    // Put data into buffer, send data
    UDR0 = data;
}
//-----
void USART_Transmit_Text (char *text) {
    unsigned char i;

    for (i=0; i<strlen(text); i++)
        USART_Transmit (text[i]);
}
//-----
int main (void) {
    USART_Transmit_text ("Ahoj!\n\n");
    USART_Transmit_text ("Ahoj");
    USART_Transmit (CR);
}
//-----
sprintf (buffer, "%02d.%02d.%02d:%02d:%02d\n", t.date, t.month, t.hour, t.minute, t.second);
USART_Transmit_Text (buffer);
```

Formátované vysielanie textu je možné realizovať pomocou funkcie *sprintf*, ktorá sa nachádza v knižnici *<stdio.h>*. Táto funkcia umožňuje vytvoriť textový refazec v pamäti MCU (*buffer*) podľa zadanych formátovacích parametrov. Jednotlivé znaky refazca sa následne vyšlú pomocou funkcie *USART_Transmit_text*.



Základy mikroprocesorovej techniky

13

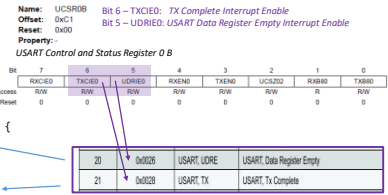
ATmega328P – režim UART – vysielanie s použitím prerušenia (24.7.3)

Vysielač USART má dva príznaky bity, ktoré charakterizujú jeho stav:

- UCSROA.UDRE0 - *USART Data Register Empty* (prázdny dátový register). Tento bit je nastavený na log. 1, ak je UDRO prázdny. Zápisom novej hodnoty (bajtu) do UDRO sa tento bit nastaví na log. 0.
- UCSROA.TXC0 - *Transmit Complete* (vysielanie ukončené). Tento bit je nastavený, keď bol obsah UDRO presunutý do posuvného registra a UDRO register je prázdny.

Obidva príznaky môžu byť použité na generovanie prerušení.

- Ak je bit UCSROB.UDRIE0 nastavený na log. 1, potom ak je prázdny dátový register (UCSROA.UDRE0) vykoná sa prerušenie **20** (*USART Data Register Empty*).
- Ak je bit UCSROB.TXCIE0 nastavený na log. 1, potom ak je vysielanie ukončené (UCSROA.TXC0) vykoná sa prerušenie **21** (*USART Tx Complete*).



Interrupt	Bit	Register
20	UDRIE0	USART_UDRE
21	TXCIE0	USART_TX

```
ISR USART0_UDRE_vect (void) {
}

ISR USART0_TX_vect (void) {
}
```

Základy mikroprocesorovej techniky

14

ATmega328P – režim UART – príklad použitia Tx.

```
UART_Txb.c
int main (void) {
    char i=0;

    wdt_enable(WDTO_25); //povolenie WDT 2sec
    DDRB |= (1<<DDRB5); //nastavenie vyvodu LED na vystup
    USART_Init (UBRR0); //inicializacia UART

    PORTD |= 1<<PORTD2; // Enable PD2 pull-up resistor
    EICRA |= 1<<ISC0B; // Trigger INT0 on falling edge
    EIMSK |= 1<<INT0; // Enable INT0
    sei(); //Enable Global Interrupt

    USART_Transmit(CR); USART_Transmit(LF);
    USART_Transmit(' '); USART_Transmit('Z'); USART_Transmit('M'); USART_Transmit('T');
    USART_Transmit(CR); USART_Transmit(LF);

    while(1) {
        wdt_reset();
        USART_Transmit (CR); USART_Transmit (LF); //novy riadok
        USART_Transmit ('0'+i); USART_Transmit (' '); //vysle index
        if (i < 9) i++;
        else i=0;
        PORTB |= 1<<PORTB5; USART_Transmit ('S'); //vysle 5
        _delay_ms (300);
        PORTB &= ~1<<PORTB5; USART_Transmit ('N'); //Vysle N
        _delay_ms (500);
        PORTB |= 1<<PORTB5; USART_Transmit ('S');
        _delay_ms (100);
        PORTB &= ~1<<PORTB5; USART_Transmit ('N');
        _delay_ms (700);
    }
}
```

Základy mikroprocesorovej techniky

15

ATmega328P – režim UART – príklad vyslania textu.

```
UART_Txb.c
ISR (INT0_vect) {
    unsigned char i = 0, j;

    EIMSK &= ~(1<<INT0); //Dissable INT0
    PORTB |= 1<<PORTB5; //LED_ON
    strcpy (buffer, "ASCII tabulka\n");
    USART_Transmit_string(buffer);
    for (i=0; i<56; i++) {
        wdt_reset();
        for(j=0; j<4; j++) {
            sprintf (buffer, "%03d %03dX %c | ", 0x20+i*4+j, 0x20+i*4+j, 0x20+i*4+j);
            USART_Transmit_string(buffer);
        }
        USART_Transmit (LF); USART_Transmit (CR);
        if (i==23) {USART_Transmit (LF); USART_Transmit (CR);
        }
        PORTB &= ~1<<PORTB5; //LED_OFF;
        EIFR |= 1<<INTIFR; //Clear Flag INT0
        EIMSK |= 1<<INT0; //Enable INT0
    }
}
//-----
int main (void) {
    wdt_enable(WDTO_500MS); //Enable WDT 0.5sec
    DDRB |= 1<<DDRB5; //nastavenie vyvodu PORTB5 na vystup
    PORTD |= 1<<PORTD2; //Enable PD2 pull-up resistor
    EICRA |= 1<<ISC0B; //Trigger INT0 on falling edge
    EIMSK |= 1<<INT0; //Enable INT0
    EIFR |= 1<<INTIFR; //Enable Global Interrupt
    sei(); //inicializacia UART
    USART_Init (UBRR0);

    strcpy (buffer, "UART_Tab.c\n");
    USART_Transmit_string(buffer);

    while(1) {
        wdt_reset();
    }
}
```

Základy mikroprocesorovej techniky

16

3