Programowanie mikrokontrolerów

RS-232

Marcin Engel Marcin Peczarski

Instytut Informatyki Uniwersytetu Warszawskiego

27 listopada 2012

RS-232

- Jeden z najstarszych interfejsów szeregowych
- Pierwotne przeznaczenie to łączenie terminali znakowych z komputerem, często z wykorzystaniem modemów
- Dwa typy urządzeń
  - DTE (ang. *Data Terminal Equipment*) – terminal, komputer
  - DCE (ang. *Data Communication Equipment*, *Data Circuit-terminating Equipment*) – zwykle modem albo inne urządzenie peryferyjne podłączane do komputera
- Prędkości transmisji od kilkudziesięciu b/s do kilkuset kb/s, typowe wartości 1200, 2400, 4800, 9600, 19200, 38400 b/s
- Zasięg do kilkunastu metrów
- Wersja asynchroniczna (bardzo popularna) i synchroniczna (obecnie już bardzo rzadko spotykana)

RS-232, złącza i kable

- Wersja asynchroniczna definiuje 9 drutów: RXD, TXD, CTS, RTS, DSR, DTR, DCD, RI, GND.
- Głównie stosuje się złącza D-Sub 9-pinowe lub 25-pinowe, żeńskie lub męskie, . . .

- . . . ale spotyka się też inne, np. RJ.
- Kabel prosty łączy DTE z DCE.
- Kabel skrzyżowany (ang. *null modem*) łączy DTE z DTE.
- Żeby móc połączyć dowolne dwa urządzenia, trzeba mieć przynajmniej 8 różnych typów kabli, . . .
- . . . ale i to może być za mało, bo te 9 drutów można połączyć ze sobą na wiele sposobów.

RS-232, wersja uproszczona

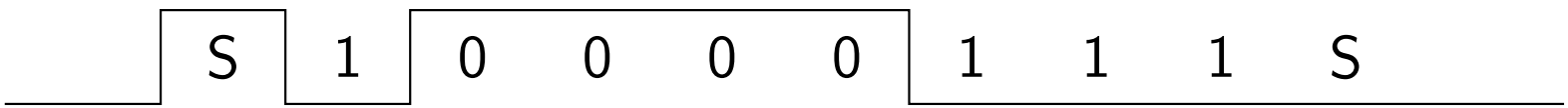
- Prawie każdy mikrokontorler obsługuje RS-232 w wersji uproszczonej, asynchronicznej, 3-drutowej:
  - RxD – odbiór w DTE, nadawanie w DCE
  - TxD – nadawanie w DTE, odbiór w DCE
  - GND – masa

RS-232, protokół komunikacyjny

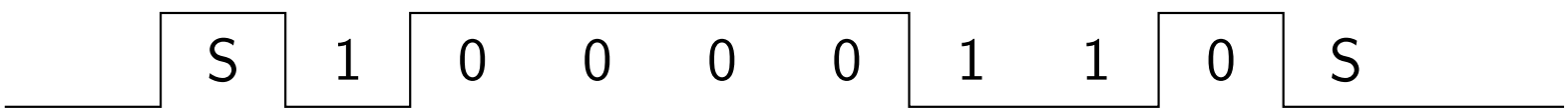
- Sygnalizacja napięciem o wartości bezwzględnej od 3 do 25 V, typowe wartości 5, 10, 12, 15 V
- Dwa poziomy napięcia
  - niski, napięcie ujemne – sygnał mark, logiczna 1, stan off
  - wysoki, napięcie dodatnie – sygnał space, logiczne 0, stan on
- Pojedyncza transmisja asynchroniczna
  - 1 bit startowy, space, logiczne 0
  - 5 do 9 bitów danych, typowo 7 lub 8, najpierw najmniej znaczący (LSB)
  - opcjonalny bit parzystości lub nieparzystości
  - 1 bit lub 1,5 bita lub 2 bity stopu, mark, logiczna 1
- Najczęściej spotykane kombinacje to
  - 7E1 – 7 bitów danych, bit parzystości, 1 bit stopu
  - 8N1 – 8 bitów danych, brak kontroli parzystości, 1 bit stopu
- Dowolnej długości przerwa między kolejnymi transmisjami

RS-232, przebiegi czasowe

- 7E1, litera 'a', kod $61



- 8N1, litera 'a', kod $61



RS-232 w ATmega16 i ATmega32

- Mikrokontrolery ATmega16 i ATmega32 mają na pokładzie USART (Universal Synchronous and Asynchronous serial Receiver Transmitter), który może pracować jako RS-232.

Rejestr UDR

- USART Data Register
- Są osobne rejestry nadawczy i odbiorczy (USART pracuje w trybie full duplex), widziane pod tym samym adresem wejścia-wyjścia.
- Zapis dokonywany jest do rejestru nadawczego.
- Odczyt dotyczy bufora odbiorczego.
- Odbierane bity są gromadzone w rejestrze odbiorczym.
- Odebrane dane są dodatkowo buforowane w dwuelementowej kolejce FIFO.

Rejestry UBRRH i UBRRL

USART Baud Rate Registers

7	6	5	4	3	2	1	0
URSEL	–	–	–	UBRR bity 11:8			
UBRR bity 7:0							

- URSEL musi być ustawiony na 0 (1 oznacza zapis do rejestru UCSRC).
- Bity 6:4 w UBRRH są zarezerwowane i należy je zerować.
- Szybkość transmisji wynosi

$$\mathrm{BAUD} = \frac{f_{\mathrm{OSC}}}{16(\mathrm{UBRR}+1)}.$$

- Wartość UBRR obliczamy ze wzoru

$$\mathrm{UBRR} = \frac{f_{\mathrm{OSC}}}{16 \cdot \mathrm{BAUD}} - 1.$$

Rejestry UBRRH i UBRRL, przykład

- Częstotliwość taktowania 8 MHz.
- Szybkość transmisji 19200 b/s.
- Obliczamy

$$\frac{f_{OSC}}{16 \cdot BAUD} - 1 \approx 25{,}04.$$

- Przyjmujemy $UBRR = 25$.
- Rzeczywista szybkość transmisji wynosi

$$\frac{f_{OSC}}{16(UBRR + 1)} \approx 19231 \text{ b/s}.$$

Specjalene oscylatory kwarcowe dla RS-232

- Chcąc uzyskać szybkość transmisji dokładnie zgodną z normą, należy taktować mikrokontroler oscylatorem kwarcowym o dobranej częstotliwości.
- Można zastosować rezonator kwarcowy o częstotliowści $n \cdot 1{,}8432$ MHz, gdzie $n = 1, 2, 4, 6, 8, 10$.
- Dla ATmega16 i ATmega32 $n \leqslant 8$.
- Np. dla $n = 8$ częstotliowść zegara wynosi 14,7456 MHz i możemy wtedy wybrać UBRR $= 47$. Dostajemy

$$\text{BAUD} = \frac{\text{f}_{\text{OSC}}}{16(\text{UBRR} + 1)} = 19200 \text{ b/s}.$$

Liczba przesyłanych bitów

- Konfigurowana za pomocą bitu UCSZ2 z rejestru UCSRB i bitów UCSZ1, UCSZ0 z rejestru UCSRC.

UCSZ2	UCSZ1	UCSZ0	liczba bitów
0	0	0	5
0	0	1	6
0	1	0	7
0	1	1	8
1	0	0	zarezerwowane
1	0	1	zarezerwowane
1	1	0	zarezerwowane
1	1	1	9

Rejestr UCSRC

USART Control and Status Register C

7	6	5	4	3	2	1	0
URSEL	UMSEL	UPM1	UPM0	USBS	UCSZ1	UCSZ0	UCPOL

- URSEL musi być ustawiony na 1 (0 oznacza zapis do rejestru UBRRH).
- UMSEL – tryb pracy
  - 0 – asynchroniczny
  - 1 – synchroniczny
- UPM1:UPM0 – tryb parzystości
  - 00 – bez bitu parzystości
  - 01 – wartość zarezerwowana
  - 10 – bit parzystości (ang. *even parity*)
  - 11 – bit nieparzystości (ang. *odd parity*)

Rejestr UCSRC, cd.

USART Control and Status Register C

7	6	5	4	3	2	1	0
URSEL	UMSEL	UPM1	UPM0	USBS	UCSZ1	UCSZ0	UCPOL

- USBS – liczba bitów stopu
  - 0 – jeden bit
  - 1 – dwa bity
- UCSZ1:UCSZ0 – liczba przesyłanych bitów
- UCPOL – polaryzacja sygnału zegara, dotyczy trybu synchronicznego

Rejestr UCSRA

USART Control and Status Register A

7	6	5	4	3	2	1	0
RXC	TXC	UDRE	FE	DOR	PE	U2X	MPCM

- RXC – Ustawiony, gdy w buforze odbiorczym są nieprzeczytane dane. Może służyć do wyzwolenia (stanem) przerwania.
- TXC – Ustawiany, gdy zakończy się nadawanie ramki. Może służyć do wyzwolenia (zdarzeniem) przerwania. Zerowany przez wpisane 1 lub przy wywołaniu przerwania.
- UDRE – Ustawiony, gdy rejestr nadawczy jest pusty. Może służyć do wyzwolenia (stanem) przerwania.
- FE – błąd ramki
- DOR – przepełnienie kolejki odbiorczej
- PE – błąd parzystości
- U2X – podwojenie prędkości transmisji, tylko dla trybu asynchronicznego
- MPCM – włączenie trybu komunikacji wieloprocesorowej

Rejestr UCSRB

USART Control and Status Register B

7	6	5	4	3	2	1	0
RXCIE	TXCIE	UDRIE	RXEN	TXEN	UCSZ2	RXB8	TXB8

- RXCIE – włączenie przerwania, gdy odebrano dane (ustawiony bit RXC)
- TXCIE – włączenie przerwania, gdy zakończono nadawanie (ustawiony bit TXC)
- UDRIE – włączenie przerwania, gdy rejestr nadawczy pusty (ustawiony bit UDRE)
- RXEN – włączenie odbiornika, zmiana funkcji wyprowadzenia PD0 na RxD
- TXEN – włączenie nadajnika, zmiana funkcji wyprowadzenia PD1 na TxD
- UCSZ2 – liczba przesyłanych bitów
- RXB8 – dziewiąty odbierany bit
- TXB8 – dziewiąty nadawany bit

Inicjowanie

- Zaczynamy jak zwykle:

```
RS232_init:
  push r16
  push r17
```

- Ustawiamy szybkość transmisji:

```
.equ SYS_FREQ_HZ = 8000000
.equ RS232_BAUD_RATE = 19200
ldi r17, high(SYS_FREQ_HZ/(16*RS232_BAUD_RATE)-1)
ldi r16, low(SYS_FREQ_HZ/(16*RS232_BAUD_RATE)-1)
out UBRRH, r17
out UBRRL, r16
```

Inicjowanie, cd.

- Ustawiamy format ramki:

```
.equ RS232_7E1 = 1<<URSEL | 1<<UPM1 | 1<<UCSZ1
.equ RS232_8N1 = 1<<URSEL | 1<<UCSZ1 | 1<<UCSZ0
ldi r16, RS232_8N1
out UCSRC, r16
```

- Uaktywniamy odbiornik i nadajnik:

```
ldi r16, 1 << RXEN | 1 << TXEN
out UCSRB, r16
```

- Kończymy inicjowanie:

```
pop r17
pop r16
ret
```

Wysłanie pojedynczej ramki

- Rejestr `r16` zawiera dane do wysłania.
- Czekamy, aktywnie `:-(`, dopóki rejestr nadawczy jest niepusty:

```
RS232_transmit_frame:
  sbis UCSRA, UDRE
  rjmp RS232_transmit_frame
```

- Wstawiamy dane do rejestru nadawczego i wracamy:

```
  out UDR, r16
  ret
```

Odebranie pojedynczej ramki

- Rejestr `r16` będzie zawierał odebrane dane.
- Rejestr `r17` będzie zawierał kod błędu.
- Znacznik `Z` będzie ustawiony, gdy nie było błędu, a wyzerowany, gdy wystąpił błąd odbioru.
- Czekamy na odebranie ramki:

```
RS232_receive_frame:
  sbis UCSRA, RXC
  rjmp RS232_receive_frame
```

- Wczytujemy znaczniki statusu transmisji i odebraną wartość (ważna jest kolejność tych operacji):

```
  in r17, UCSRA
  in r16, UDR
```

- Maskujemy bity błędów, modyfikujemy znacznik `Z` i wracamy:

```
  andi r17, 1 << FE | 1 << DOR | 1 << PE
  ret
```

Konfiguracja sprzętu

- 0 w mikrokontrolerze to napięcie 0 V, a 1 to napięcie zasilania.
- Trzeba zastosować konwerter poziomów logicznych, np. układ MAX232.
- Na płytce uruchomieniowej mamy jego odpowiednik ST3232.
- Trzeba odłączyć wyprowadzenia PD0 i PD1 od wyświetlacza LCD przez wyjęcie odpowiednich zworek.
- Trzeba połączyć:
  - wyprowadzenie PD1 (TxD) z wejściem TxD konwertera,
  - wyprowadzenie PD0 (RxD) z wyjściem RxD konwertera,
  - kablem złącze DB9 na płytce uruchomieniowej z odpowiednim złączem w komputerze.

Konfiguracja oprogramowania

- Do komunikacji możemy użyć programu HyperTerminal lub PuTTY.
- Trzeba tylko skonfigurować parametry komunikacji.

Konfiguracja w VMLAB

- 19200 bitów na sekundę, 8 bitów, bez parzystości, 1 bit stopu

  `XRS232 TTY(19200 8) PD0 PD1`

- 19200 bitów na sekundę, 7 bitów, bit parzystości, 1 bit stopu

  `XRS232 TTY(19200 7 1) PD0 PD1`

Propozycje zadań

- Zaimplementować termometr podłączany do komputera przez RS-232. Mikrokontroler:
  - czeka na rozkaz wykonania pomiaru,
  - dokonuje pomiaru,
  - odsyła wynik pomiaru.
- Zaimplementować „instrument muzyczny” sterowany z komputera przez RS-232. Mikrokontroler:
  - odbiera wysokość i długość dźwięku,
  - generuje odpowiedni dźwięk.