Systemy wbudowane Wykład 6 - transmisje szeregowe: UART i pochodne Przemek Błaśkiewicz 22 kwietnia 2018 1/57 Komunikacja szeregowa http://websdr.org 2/57

Notes

Notes

Notes

Notes

Rodzaje transmisji

Rodzaje transmisji

simplex/sympleks

simplex/sympleks

Komunikacja taka, że Rx i Tx są niewymienne: jednokierunkowa.

Komunikacja taka, że Rx i Tx są niewymienne: jednokierunkowa.

duplex/dupleks

Komunikacja, gdzie Rx i Tx zamieniają się rolami: dwukierunkowa.

Rodzaje transmisji	Notes
simplex/sympleks	
Komunikacja taka, że Rx i Tx są niewymienne: jednokierunkowa.	
duplex/dupleks	-
Komunikacja, gdzie Rx i Tx zamieniają się rolami: dwukierunkowa. Półdupleks (half duplex): komunikacja jest naprzemienna.	
5/57	
Rodzaje transmisji	Manage
	Notes
simplex/sympleks Komunikacja taka, że Rx i Tx są niewymienne: jednokierunkowa.	
duplex/dupleks	
Komunikacja, gdzie Rx i Tx zamieniają się rolami: dwukierunkowa.	
Półdupleks (half duplex): komunikacja jest naprzemienna. Pełny dupleks (full duplex): komunikacja w obie strony jest niezależna	
педанедна	
6/57	
0/9/	
UART	
	Notes
7/57	
UART	
	Notes
 Universal Asynchronous Receiver and Transmitter 	

		Notes
Universal Asynchronous Receiver and Transmitter		
 służy do przesyłania informacji bit po bicie (na jednej linii) 		
posiada bufory we/wy dla wygody szybszych transmisjiodciąża procesor w zadaniu komunikacji		
 nie zajmuje się poziomami logicznymi/napięciowymi na liniach RS-232, RS-485, IrDA, Bluetooth tryb SPP, modem 		
• 13-232, 13-403, 110A, Diaetotti tiyb 311, Illodelli		
	13/57	
IADTatronaéai	_	
JART - wnętrzności "Paczka danych":		Notes
r aczna danych .		
	14 / 57	
LART	_	
JART - wnętrzności "Paczka danych":		Notes
• bit START		
	15 / 57	
IART	_	
JART - wnętrzności "Paczka danych":		Notes
bit START		
• 5,6,7 lub 8 bitów DANYCH		

UART - wnętrzności Notes "Paczka danych": bit START 5,6,7 lub 8 bitów DANYCH • 1 bit PARZYSTOŚCI (opcjonalnie) 17 / 57 UART - wnętrzności Notes "Paczka danych": bit START • 5,6,7 lub 8 bitów DANYCH • 1 bit PARZYSTOŚCI (opcjonalnie) • 1, 1.5 lub 2 bity STOP 18 / 57 UART - wnętrzności Notes "Paczka danych": bit START • 5,6,7 lub 8 bitów DANYCH • 1 bit PARZYSTOŚCI (opcjonalnie) • 1, 1.5 lub 2 bity STOP PARZYSTOŚĆ (opcja) START DANE STOP D1 D2 D3 D4 D5 D7 PRÓBKOWANIE Bit parzystości parzystej i nieparzystej Notes Bit parzystości Przyjmuje 0 dla **parzystej** liczby jedynek w słowie, 1 w p.p.

Bit parzystości parzystej i nieparzystej Notes Bit parzystości Przyjmuje 0 dla parzystej liczby jedynek w słowie, 1 w p.p. Bit nieparzystości Przyjmuje 0 dla **nieparzystej** liczby jedynek w słowie, 1 w p.p. 21 / 57 Bit parzystości parzystej i nieparzystej Notes Bit parzystości Przyjmuje 0 dla parzystej liczby jedynek w słowie, 1 w p.p. Bit nieparzystości Przyjmuje 0 dla nieparzystej liczby jedynek w słowie, 1 w p.p. 22 / 57 Bit parzystości parzystej i nieparzystej Notes Bit parzystości Przyjmuje 0 dla ${\bf parzystej}$ liczby jedynek w słowie, 1 w p.p. Bit nieparzystości Przyjmuje 0 dla nieparzystej liczby jedynek w słowie, 1 w p.p. function mxor(signal A : std_logic_vector) return std_logic variable tmp : std_logic; begin tmp := '0'; for i in A'range loop tmp := tmp xor A(i); end loop; tmp := tmp xor '0'; return tmp; end mxor; Transmisja UART Notes baud (Bd) wym. bod – ilość *zmian* stanu medium na sekundę, związana z wysyłaniem symbolu kodowego (nie bitu informacji!).

Transmisja UART Notes baud (Bd) wym. bod – ilość zmian stanu medium na sekundę, związana z wysyłaniem symbolu kodowego (nie bitu informacji!). W przypadku gdy symbol kodowy jest równoważny bitowi baud = bitrate. bezczynna linia jest w stanie "Hi" 25 / 57 Transmisja UART Notes baud (Bd) wym. bod – ilość zmian stanu medium na sekundę, związana z wysyłaniem symbolu kodowego (nie bitu informacji!). W przypadku gdy symbol kodowy jest równoważny bitowi $\mathsf{baud} = \mathsf{bitrate}.$ bezczynna linia jest w stanie "Hi" ullet dane do wysłania wchodzą równolegle do UART ightarrow rejestr przesuwny 26 / 57 Transmisja UART baud (Bd) wym. bod – ilość *zmian* stanu medium na sekundę, związana z wysyłaniem symbolu kodowego (nie bitu informacji!). W przypadku gdy symbol kodowy jest równoważny bitowi baud = bitrate. bezczynna linia jest w stanie "Hi" ullet dane do wysłania wchodzą równolegle do UART ightarrow rejestr wysyłany jest bit START (Lo) Transmisja UART baud (Bd) wym. bod – ilość *zmian* stanu medium na sekundę, związana z

wysyłaniem symbolu kodowego (nie bitu informacji!). W przypadku gdy symbol kodowy jest równoważny bitowi baud = bitrate.

- bezczynna linia jest w stanie "Hi"
- ${\color{blue} \bullet}$ dane do wysłania wchodzą równolegle do UART \rightarrow rejestr
- wysyłany jest bit START (Lo)
- odbiornik wykrywa stan Lo jeśli linia jest przez min 1/2 czasu trwania bitu w stanie Lo

Notes				
	Votes			
Votes				
Notes				
Notes				
Votes				
Notes				
Notes				
Notes				
Votes				
Votes				
Notes				
	Votes			
	VOICS			

Transmisja UART

baud (Bd)

wym. bod – ilość zmian stanu medium na sekundę, związana z wysyłaniem symbolu kodowego (nie bitu informacji!). W przypadku gdy symbol kodowy jest równoważny bitowi baud = bitrate.

- bezczynna linia jest w stanie "Hi"
- dane do wysłania wchodzą równolegle do UART → rejestr przesuwny
- wysyłany jest bit START (Lo)
- odbiornik wykrywa stan Lo jeśli linia jest przez min 1/2 czasu trwania bitu w stanie Lo
- przez ustaloną ilość bitów do odbioru medium jest samplowane mniej więcej w środku przedziału

29 / 57

Transmisja UART

baud (Bd)

wym. bod – ilość zmian stanu medium na sekundę, związana z wysyłaniem symbolu kodowego (nie bitu informacji!). W przypadku gdy symbol kodowy jest równoważny bitowi baud = bitrate.

- bezczynna linia jest w stanie "Hi"
- wysyłany jest bit START (Lo)
- odbiornik wykrywa stan Lo jeśli linia jest przez min 1/2 czasu trwania bitu w stanie Lo
- przez ustaloną ilość bitów do odbioru medium jest samplowane mniej więcej w środku przedziału
- o dane odebrane ładowane są do rejestru przesuwnego

30 / 57

Transmisja UART

baud (Bd)

wym. bod – ilość zmian stanu medium na sekundę, związana z wysyłaniem symbolu kodowego (nie bitu informacji!). W przypadku gdy symbol kodowy jest równoważny bitowi baud = bitrate.

- bezczynna linia jest w stanie "Hi"
- ullet dane do wysłania wchodzą równolegle do UART o rejestr przesuwny
- wysyłany jest bit START (Lo)
- odbiornik wykrywa stan Lo jeśli linia jest przez min 1/2 czasu trwania bitu w stanie Lo
- przez ustaloną ilość bitów do odbioru medium jest samplowane mniej więcej w środku przedziału
- o dane odebrane ładowane są do rejestru przesuwnego
- po odebraniu bitu STOP dane przesyłane są do rejestru odbiorczego

31 / 57

Transmisja UART

n UART input clock cycles, where n = divisor in DLH:DLL

UART input clock

Each bit lasts 16 BCLK cycles.

When receiving, the UART samples the bit in the 8 th cycle.

UART T.TXD.

UART T

Notes		
Notes		
Notes		
Notes		

Transmisja UART

Baud Rate	Divisor Value	Actual Baud Rate	Error (%)
2400	3906	2400.154	0.01
4800	1953	4800.372	0.01
9600	977	9595.701	-0.04
19200	488	19211.066	0.06
38400	244	38422.131	0.06
56000	167	56137.725	0.25
128000	73	129807.7	0.33
3000000	3	3125000	4.00

 \rightarrow 150MHz, div=3906, oversampl=16, 7 bitów danych, 1 parzystości

33 / 57

Notes

Transmisja UART

	Actual Baud Rate	Error (%)
3906	2400.154	0.01
1953	4800.372	0.01
977	9595.701	-0.04
488	19211.066	0.06
244	38422.131	0.06
167	56137.725	0.25
73	129807.7	0.33
	1953 977 488 244 167	1953 4800.372 977 9995.701 489 1921.066 244 38422.131 167 56137.725

 \rightarrow 150MHz, div=3906, oversampl=16, 7 bitów danych, 1 parzystości

(150000000/3906)/16 = 2400.154)

34 / 57

Transmisja UART

Table 2-1 Baud Rate Examples for 150-MHz UART Input Clock and 16× Oversampling Mode				
Baud Rate	Divisor Value	Actual Baud Rate	Error (%)	
2400	3906	2400.154	0.01	
4800	1953	4800.372	0.01	
9600	977	9595.701	-0.04	
19200	488	19211.066	0.06	
38400	244	38422.131	0.06	
56000	167	56137.725	0.25	
128000	73	129807.7	0.33	

- \rightarrow 150MHz, div=3906, oversampl=16, 7 bitów danych, 1 parzystości
 - $\bullet \ (150000000/3906)/16 = 2400.154)$
 - $\bullet \ \mathsf{START} + \mathsf{7*DATA} + \mathsf{PARITY} + \mathsf{STOP} \ \to 10 \ \mathsf{bit\'ow}$

35 / 57

Transmisja UART

Table 2-1 Baud Ra	ite Examples for 150-MHz UART In	150-MHz UART Input Clock and 16× Oversampling Mode		
Baud Rate	Divisor Value	Actual Baud Rate	Error (%)	
2400	3906	2400.154	0.01	
4800	1953	4800.372	0.01	
9600	977	9595.701	-0.04	
19200	488	19211.066	0.06	
38400	244	38422.131	0.06	
56000	167	56137.725	0.25	
128000	73	129807.7	0.33	

- \rightarrow 150MHz, div=3906, oversampl=16, 7 bitów danych, 1 parzystości
 - (150000000/3906)/16 = 2400.154)
 - $\hspace{0.1in} \bullet \hspace{0.1in} \mathsf{START} + \mathsf{7*DATA} + \mathsf{PARITY} + \mathsf{STOP} \hspace{0.1in} \to \hspace{0.1in} \mathsf{10} \hspace{0.1in} \mathsf{bit\'ow}$
 - ullet 2400Bd ightarrow 2400 zmian (tu: bitów) na sekundę

Notes	
Notes	
Notes	

Transmisja UART

Table 2-1	Input Clock and 16× Oversampling Mo	ode	
Baud Rate	Divisor Value	Actual Baud Rate	

Baud Rate	Divisor Value	Actual Baud Rate	Error (%)
2400	3906	2400.154	0.01
4800	1953	4800.372	0.01
9600	977	9595.701	-0.04
19200	488	19211.066	0.06
38400	244	38422.131	0.06
56000	167	56137.725	0.25
128000	73	129807.7	0.33
3000000	3	3125000	4.00

 \rightarrow 150MHz, div=3906, oversampl=16, 7 bitów danych, 1 parzystości

- (150000000/3906)/16 = 2400.154)
- $\hspace{0.1cm} \bullet \hspace{0.1cm} \mathsf{START+7*DATA+PARITY+STOP} \hspace{0.1cm} \to \hspace{0.1cm} \mathsf{10} \hspace{0.1cm} \mathsf{bit\'ow} \\$
- ullet 2400Bd ightarrow 2400 zmian (tu: bitów) na sekundę
- $2400 \frac{bit}{sek} / 10 \frac{bit}{paczka} = 240 \frac{paczka}{sek}$

37 / 57

UART i inne

• sygnały komunikacyjne: RTS, CTS (handshake)

38 / 57

UART i inne

- sygnały komunikacyjne: RTS, CTS (handshake)
- USART

UART i inne

- sygnały komunikacyjne: RTS, CTS (handshake)
- USART
 - synchronizuje Rx/Tx na podstawie ciągu danych

Notes	
Notes	
Notes	
Notes	
Notes	
······	

DUART, OCTARTbit-banging - software serial

		Notes
 sygnały komunikacyjne: RTS, CTS (handshake) USART synchronizuje Rx/Tx na podstawie ciągu danych 		
• w trakcie braku transmisji potrzebne są "pingi" (ASCII SYN 0x16)		
	41 / 57	
ART i inne	_	
AIXT I IIIIE		Notes
sygnały komunikacyjne: RTS, CTS (handshake)USART		
 synchronizuje Rx/Tx na podstawie ciągu danych w trakcie braku transmisji potrzebne są "pingi" (ASCII SYN 0x16) 		
 zwiększona przepustowość: brak START/STOP 		
	42 / 57	
ADT	_	
ART i inne		Notes
sygnały komunikacyjne: RTS, CTS (handshake)USART		
 synchronizuje Rx/Tx na podstawie ciągu danych w trakcie braku transmisji potrzebne są "pingi" (ASCII SYN 0x16) 		
 zwiększona przepustowość: brak START/STOP DUART, OCTART 		
	43 / 57	
	_	
ART i inne		Notes
sygnały komunikacyjne: RTS, CTS (handshake)USART		
 synchronizuje Rx/Tx na podstawie ciągu danych w trakcie braku transmisji potrzebne są "pingi" (ASCII SYN 		
0x16) ⊛ zwiększona przepustowość: brak START/STOP		

Notes	
	—
Notes	
Notes	
	<u> </u>

RS-232



45 / 57

RS-232



 $\ \, \bullet \,$ Jest to $standard\ połączenia\ urządzeń\ (nazwy\ styków,\ poziom\ sygnałów)$

46 / 57

RS-232



- $\, \bullet \,$ Jest to $standard \, połączenia urządzeń (nazwy styków, poziom sygnałów) <math display="inline">\,$
- Logiczne 1 jako napięcia od -3 do -15V, logiczne 0 jako 3 -

47 / 57

RS-232



- Jest to standard połączenia urządzeń (nazwy styków, poziom sygnałów)
- Logiczne 1 jako napięcia od -3 do -15V, logiczne 0 jako 3 -15V
- \bullet Typowo \pm 5 V, \pm 10 V, \pm 12 V, \pm 15 V

Notes		
	 	
Notes		
Notes		



- $\,\circ\,$ Jest to $standard\ połączenia\ urządzeń\ (nazwy\ styków, poziom\ sygnałów)$
- Logiczne 1 jako napięcia od -3 do -15V, logiczne 0 jako 3 -15V
- $_{\rm 0}$ Typowo \pm 5 V, \pm 10 V, \pm 12 V, \pm 15 V
- Potrzebne konwertery napięć!

49 / 57

Notes

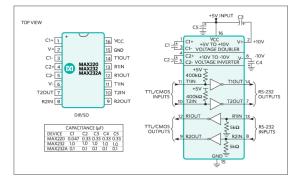
RS-232



- Jest to standard połączenia urządzeń (nazwy styków, poziom sygnałów)
- Logiczne 1 jako napięcia od -3 do -15V, logiczne 0 jako 3 -15V
- ${\tt o}$ Typowo \pm 5 V, \pm 10 V, \pm 12 V, \pm 15 V
- Potrzebne konwertery napięć!
- $\bullet \ \ Wykorzystywane \ konwertery \ RS-232 \leftrightarrow USB$

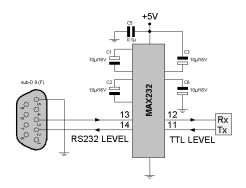
50 / 57

Rodzina MAXów



51/57

Rodzina MAXów



Notes		
Notes		
Notes		



53 / 57

Inne protokoły... SPI, I^2C ,one-wire, oraz kontrolery w następnych odcinkach

54 / 57



55 / 57



- Arduino + RF-433 + biblioteka RadioHead
- SDR: RTL8232U + gqrx (ew. GnuRadio)
- ullet analizator stanów logicznych Saleae

Notes		
Notes		
Notes		
Notes		
Notes		
Notes		
Notes		
INCLUS		

Notes			

Rzeczy do zapamiętania

- rodzaje transmisji, Bd,
- UART: tryby, działanie, ramka danych

57 / 57

• "radzenie sobie" z RS-232

Notes	
Notes	
Notes	
Notes	
Notes	
-	