Universal asynchronous receiver/transmitter (UART)

Vgrajeni sistemi

Aleš Čep

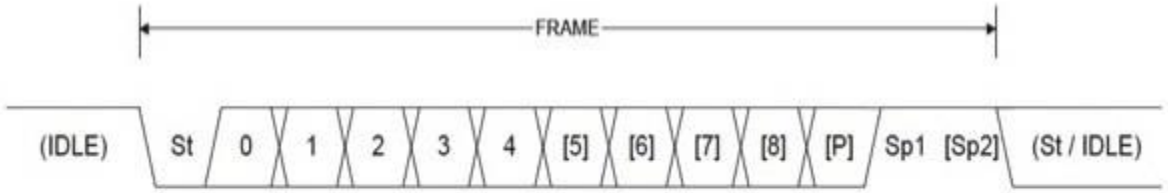
Komunikacijski protokoli

- Pri povezavi več naprav
- Obstaja več protokolov:
  - SPI (full-duplex, hiter, 3 žice, naslavljanje z chip set, običajno pri manj napravah)
  - U(S)ART (asinhroni serijski protokol, zelo razširjen, paketno pošiljanje, običajno med dvema)
  - I2C (TWI) (half-duplex, 2 žici, naslavljanje z naslovom (običajno 7bitni))
  - 1-wire (poceni, počasen, 1 žica, daljše razdalje, napajanje se prenaša skupaj s podatki na isti liniji)

Kaj je UART in kaj USART?

- Universal Synchronous/Asynchronous Receiver/Transmitter
- Oba sta vrsti serijske povezave
- Običajno se uporabljata za povezavo 2 naprav
- USART je namenjen sinhroni serijski povezavi
- UART je asinhrona serijska komunikacija, ki je poceni, preprosta za uporabo in zelo razširjena.
- Ne potrebuje ure in potrebuje zelo malo povezav – dve povezavi za pošiljanje podatkov (RX in TX).
- Paketno pošiljanje

Zgradba paketa



St	Start bit, always low.
(n)	Data bits (0 to 8).
P	Parity bit. Can be odd or even.
Sp	Stop bit, always high.
IDLE	No transfers on the communication line (RxD or TxD). An IDLE line must be high.

- Paket je sestavljen vsaj iz:
  - 1 začetni bit (aktivno 0)
  - Podatkovni biti (5-9)
  - Končen bit

http://i1.wp.com/maxembedded.files.wordpress.com/2013/09/frame-format.jpg

Zgradba paketa

- Paket je sestavljen iz več podatkovnih bitov. To število je običajno med 5 in 9 (na STM32F429I se lahko nastavi 8 ali 9 bitov).
- Paritetni bit
- Stop bit (koliko bitov želimo dodati na koncu – običajno so možnosti 1 ali 2 bita).

Baud rate (hitrost pošiljanja)

- Gre se za asinhrono komunikacijo, zato mora naprava, ki bere, vedeti, kdaj mora brati. Dogovoriti se je potrebno za hitrost.
- Obstajajo *dogovorjene* vrednost glede hitrosti (Tx/Rx baud) prenosa (2400,4800,9600,115200,…)

Baud rate - izračun

- Običajno je potrebno za konfiguracijo *baud rate* izračunati eno celo število (približek), s čimer so hitrosti omejene.
- Na STM32F4 je na možnost velika izbira možnih hitrosti in konfiguracij, saj lahko konfiguracijo podamo z realnim številom. Enačba je zapisana spodaj. Izračunati je potrebno USARTDIV. Rezultat se zapiše v register BRR.

$$\text{Tx/Rx baud} = \frac{f_{CK}}{8 \times (2 - \text{OVER8}) \times \text{USARTDIV}}$$

Pariteta

- ZELO preprost mehanizem za preverjanje pravilnosti prenosa (najde neujemanje za 1 (oz liho število) bit)
- PCE bit v CR1 (Paritetni bit je eden izmed podatkovnih bitov.)
- V poslanem podatku naj bo sodo/liho število postavljenih bitov
- Stanje je lahko (PS bit v CR1):
  - soda pariteta (1010001 **1**)
  - liha pariteta (1010001 **0**)

Oversampling

- Z nastavljanje *oversampling*-a želimo preprečiti, da bi *zgrešili* kakšen začeten bit.
- Uporablja se za *iskanje* začetnega bita, za zmanjševanje šuma.
- Najbolj pogosto deluje po principu volitev – v enem časovnem vzorcu se npr. 16 preveri stanje signala. Vzamejo se sredinske tri vrednost in izbere se tista, ki se pojavi največkrat
- STM32F4 podpira dve možnosti: 8 ali 16

Registri – Status register (SR)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved						CTS	LBD	TXE	TC	RXNE	IDLE	ORE	NF	FE	PE
						rc_w0	rc_w0	r	rc_w0	rc_w0	r	r	r	r	r

- TXE – znak „poslan“, lahko se pošlje nov
- RXNE – znak je bil sprejet in je pripravljen za branje

Registri – Data register (DR)

- Namenjen je pošiljanju in prejemanju podatkov.
- Za pošiljanje v register zapišete znak:

```
WRITE_REG(USART1->DR, znak);
```

- Za branje pa register preberete in rezultat shranite v spremenljivko:

```
char znak = READ_REG (USART1->DR);  ali
char znak = USART1->DR;
```

Registri – Baud rate register (BRR)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DIV_Mantissa[11:0]												DIV_Fraction[3:0]			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits 31:16 Reserved, must be kept at reset value

Bits 15:4 **DIV_Mantissa[11:0]**: mantissa of USARTDIV
These 12 bits define the mantissa of the USART Divider (USARTDIV)

Bits 3:0 **DIV_Fraction[3:0]**: fraction of USARTDIV
These 4 bits define the fraction of the USART Divider (USARTDIV). When OVER8=1, the DIV_Fraction3 bit is not considered and must be kept cleared.

- Nastavitev hitrosti pošiljanja podatkov (b/s)
- DIV_Mantissa: 12 bitni podatek (celi del števila)
- DIV_Fraction: 4 bitni podatek (realni del števila)
- Poglavje: 30.3.4

BRR:

$$\text{Tx/Rx baud} = \frac{f_{CK}}{8 \times (2 - \text{OVER8}) \times \text{USARTDIV}}$$

- Tx/Rx baud: hitrost (4800,9600,115200, …)
- f_{CK} : privzeto na našem procesorju je 16 MHz
- OVER8: konfiguracijski bit v registru CR1
- Kot pomoč vam je tabela 136, v kateri so že preračune vrednosti za zgornjo enačbo.

BRR Primer:

Table 136. Error calculation for programmed baud rates at f_{PCLK} = 16 MHz or f_{PCLK} = 24 MHz, oversampling by 16(1)

Oversampling by 16 (OVER8 = 0)							
Baud rate		f_{PCLK} = 16 MHz			f_{PCLK} = 24 MHz		
S.No	Desired	Actual	Value programmed in the baud rate register	% Error = (Calculated - Desired) B.rate / Desired B.rate	Actual	Value programmed in the baud rate register	% Error
1	1.2 KBps	1.2 KBps	833.3125	0	1.2	1250	0
2	2.4 KBps	2.4 KBps	416.6875	0	2.4	625	0

- DIV_Mantissa: 833
- DIV_Fraction: 5 ($0{,}3125_{(10)} = 0101_{(2)}$)

Registri – Control register 1(CR1)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OVER8	Reserved	UE	M	WAKE	PCE	PS	PEIE	TXEIE	TCIE	RXNEIE	IDLEIE	TE	RE	RWU	SBK
rw	Res.	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

- OVER8: oversampling izbira
- UE: omogoči se U(S)ART
- M: nastavi se sestava paketa (1,8,n ali 1,9,n)
- PCE: omogoči se pariteta
- PS: izbira paritete
- TE: omogoči se pošiljanje
- RE: omogoči se prejemanje

Registri – Control register 2(CR2)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	LINEN	STOP[1:0]		CLKEN	CPOL	CPHA	LBCL	Res.	LBDIE	LBDL	Res.	ADD[3:0]			
	rw	rw	rw	rw	rw	rw	rw		rw	rw	rw	rw	rw	rw	rw

- STOP: število stop bitov

Dokumentacija

- U(S)ART se nahaja v poglavju 30 (str. 969)
- Postopek konfiguracije za pošiljanje znaka najdete v podpoglavju 30.3.2.
- Postopek konfiguracije za prejemanje znaka najdete v podpoglavju 30.3.3.
- Na STM32F429 plošči je U(S)ART1 povezan na pina PA9 in PA10 (Specifikacije plošče STM32F429I Discovery na strani 28 levo).