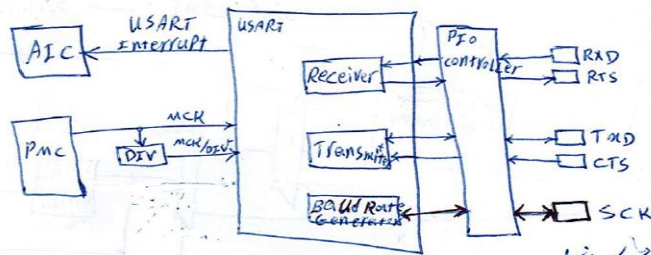


۷۷

ARM

« بنای خدا »

یورت سریال : « USART »
 universal Synchronous Asynchronous Receiver Transmitter
 امکان ارتباط سریال سینکرون یا آسنکرون
 Full Duplex یعنی دو مسیر جدا برای ارسال و دریافت
 میکرو AT91SAM7S64 دارای دو USART می باشد : USART0 و USART1



slave clock درگاه سنجش کارایی دارد.

منبع کلاک USART :
 SCK هم می تواند از یک منبع MCK انتخاب شود

منبع کلاک USART : می تواند MCK (کلاک اصلی) و (MCK/DIV) است

کنترل : « Power management controller » PMC تأمین می شود

تذکره : کلاک USART نیاز به یک میکرو عموماً می شود بنابراین قبل از کار با USART

لازم است کلاک را فعال کنیم.

$$\text{AT91C-BASE-PMC} \rightarrow \text{PMC-PCER} = (1 \ll \text{AT91C-ID-US1})$$

نکته : در یکون عدد ۸ DAR می باشد.

$\text{RXD}_0 \xrightarrow{35} \text{PA.5}$ $\text{TXD}_0 \xrightarrow{34} \text{PA.6}$ $\text{RTS}_0 \xrightarrow{32} \text{PA.7}$ $\text{CTS}_0 \xrightarrow{31} \text{PA.8}$ $\text{SCK}_0 \xrightarrow{44} \text{PA.2}$	Peripheral B	$\text{RXD}_1 \xrightarrow{11} \text{PA.21}$ $\text{TXD}_1 \xrightarrow{14} \text{PA.22}$ $\text{RTS}_1 \xrightarrow{23} \text{PA.24}$ $\text{CTS}_1 \xrightarrow{25} \text{PA.25}$ $\text{SCK}_1 \xrightarrow{15} \text{PA.23}$	Peripheral A
---	--------------	--	--------------

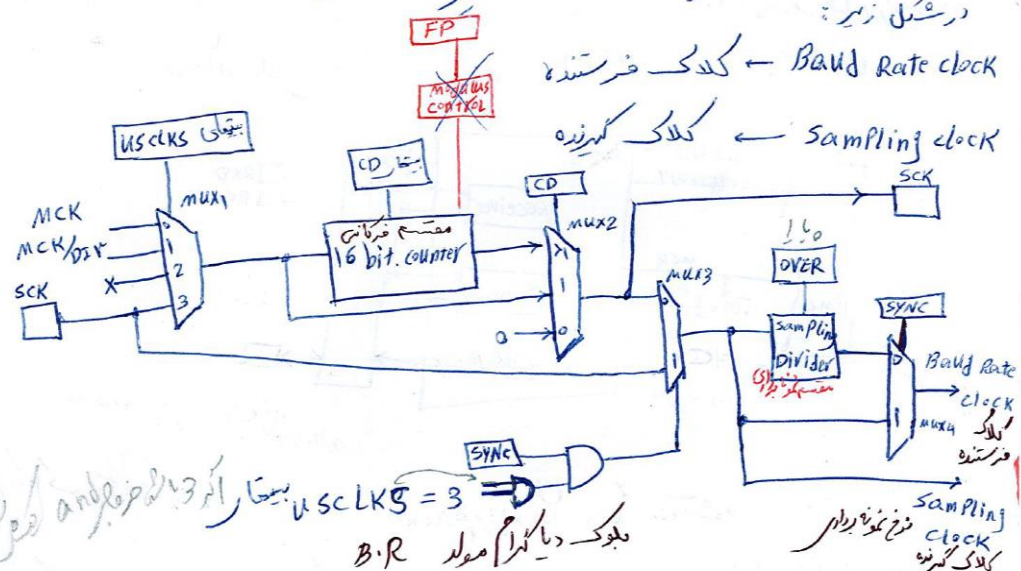
خطوط ورودی و خروجی USART :

۴

۷۸

مولد نرخ باور "BAUD Rate Generator"

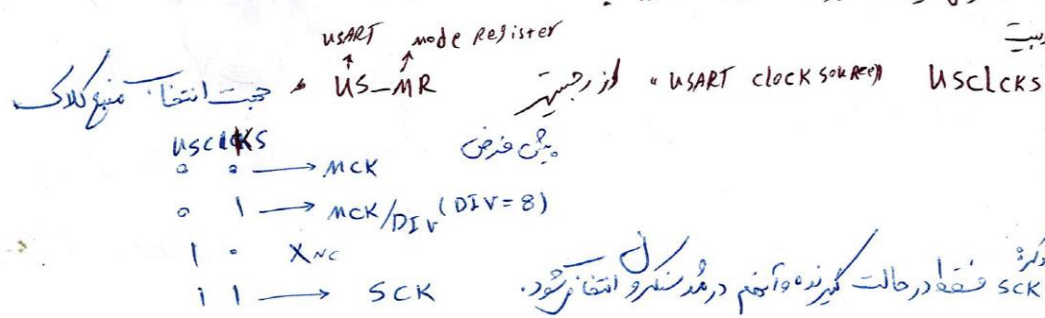
در شکل زیر: وظیفه تولید کلاک شیفت رجیستر ارسال و دریافت را دارد.



بسیار اگر 3 یا 4 فرکانس و فرکانس $USCLKS = 3$ بزرگ دیاگرام مولد B.R

تذکره: بین SCK در فرستنده فرکانس (SCK آخر شکل) و درگیرنده و درون (SCK اول شکل)

معرفی بیهای بلوک مولد B.R :



تذکره: فقط در حالت گیرنده و آنگاه در فرستنده انتخاب شود. $USCLKS$ بیهای CD هم مقدار مقسم فرکانس را مشخص میکند و هم جهت انتخاب بای از ورودیها $US - BRGR$ اگر بزرگتر از ۱ باشد خروجی مقسم نیز انتخاب شود. اگر کوچکتر از ۱ باشد، انتخاب نمیشود و اگر صفر باشد مقدار صفر انتخاب نمیشود.

۳

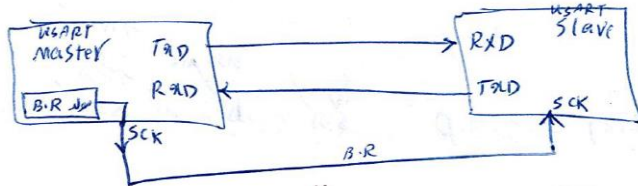
۷۹

نکته

US-MR مد عملیات را مشخص میکند:
این فرقی (مد عملیات) است که
0 → مد عملیات است
1 → مد عملیات است

نکته

تذکره: اگر مد عملیات مستند انتخاب شود اولاً باید در هر دو USART، B.R توسط یک منبع کلاک مشترک تأمین شود در برابر انتخاب B.R توسط MCK یا MCK/DIV میسر نیست
از USART ها تولید شود (master) و از طریق SCK به USART دوم اعمال شود (slave)



تذکره: SCK در USART master به عنوان خروجی و در USART slave به عنوان ورودی عمل میکند

بیکر نیز USART به عنوان master: (در حالت مستند)

USART master و USART های گرفته شده B.R برابر هر دو USART ایجاد کند

با انتخاب کلاک از دو منبع MCK یا MCK/DIV یک منبعی USCLKS

$$\text{Baud Rate clock} = \text{Sampling clock} = \frac{\text{select clock}}{CD}$$

در حالت مستند هر USART می تواند دارنده باشد

تذکره: در Master برابر انتخاب SCK به عنوان خروجی B.R لازم است بیت CLK0
این فرقی مشخص است که SCK به عنوان ورودی برای slave تنظیم شده است. فقط در مد مستند
از جمله US-MR یک کنیم.
تذکره: اگر MCK/DIV به عنوان منبع کلاک انتخاب شده باشد، بزرگتر از 50% Duty cycle B.R
لازم است به رابطه CD زوج انتخاب شود.

۸۰

در دستگیر کردن
USART به عنوان slave

USART slave به USART گفته می شود که B.R خود را از USART دیگری دریافت می کند.

با انتخاب بیت های
 $USCLKS = 3$
 $SYNC = 1$
 $AND = 1$
 SCK به عنوان منبع کلاک B.R ورودی انتخاب می شود.

Baud Rate clock = sampling clock = SCK

در slave
یکپوشه

با انتخاب بیت $SYNC = 0$ می توان انتخاب می شود

نکته: بک بیت OVER همیشه فرکانس کلاک B.R در دستگیر (Baud Rate clock)
 ۱ یا ۱/۱۶ کلاک گیرنده (sampling clock) می شود تا گیرنده فرکانس کلاک
 برابر نمونه بردار داشته باشد.
 $OVER = 0$ نرخ نمونه بردار (گیرنده) ۱۶ برابر نرخ ارسال (فرستنده)
 ۱ ۱ ۱ ۱ ۱ ۱ ۱ ۱ ۱ ۱ ۱ ۱ ۱ ۱ ۱ ۱

Baud Rate clock = $\frac{\text{کلاک انتخابی}}{CD(2 - OVER) \times 16}$

برای کلاک
فرستنده

sampling clock = $\frac{\text{کلاک انتخابی}}{CD}$

Error = $(1 - \frac{\text{نرخ بردارنده}}{\text{نرخ ارسال}}) \times 100\%$

خطا "error"

مثال: اگر $MCK = 8 MHz$ و $B.R = 289600$ در حالت $OVER = 0$ (نرخ بردارنده ۱۶ برابر)

مقدار $CD = 14$ تقریباً صحیح
 $CD = 14$
 $BR = 289611.05$

Error = $(1 - \frac{289600}{289611.05}) \times 100\% = 0.0036\%$

⑤

۸۱

معمولترین فرکانس مورد استفاده در میکروکنترلر:

مکانیزم: $MCK = 40 MHz$ و $CD = 1, 2, 4$ داریم: خروجی مقسم فرکانس

$$\frac{40 MHz}{1} = 40 MHz$$

همانطور که ملاحظه می‌کنیم بین $20 MHz$ و $40 MHz$ رنج دیگری

$$\frac{40}{2} = 20 MHz$$

نداریم، بهر حال داشتن رنج‌های بیشتر از FP

$$\frac{40}{3} = 13,333 MHz$$

در جستجوی $US-BRGR$ استفاده می‌کنیم (پس فرض می‌کنیم).

بنابراین در روابط قبل هر جا CD داریم بجای آن $CD + \frac{FP}{8}$ می‌نویسیم.

$$Baud Rate clock = \frac{\text{کلاک انتخابی}}{8(2-OVER)(CD + \frac{FP}{8})}$$

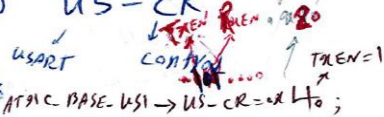
$$sampling clock = \frac{\text{کلاک انتخابی}}{CD + (\frac{FP}{8})}$$

نتیجه: حالت توان بین $CD = 1$ و $CD = 2$ رنج دیگری تعریف کرد.

کنترل فرستنده و گیرنده: (در جستجوی $US-CR$)

۱- فعال ساز فرستنده و گیرنده:

بعد از رست میکرو، گیرنده و فرستنده بصورت پیش فرض غیر فعال می‌شوند. برای نویسی لازم است با نوشتن یک در بیت‌های $RXEN$ و $TXEN$ از رجیستر $US-CR$ گیرنده و فرستنده $USART$ را فعال کند.



۸۲

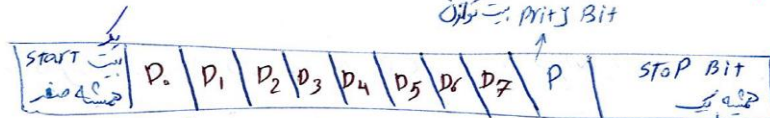
۲- رست کردن نرم افزار گیرنده و فرستنده:

با نوشتن یک درستی‌های RSTRX و RSTTX از رجیستر US-CR
 عملیات ~~فرستنده و گیرنده~~ فرستنده و گیرنده Reset می‌شوند (عملیات ارسال و دریافت منوقت می‌شود)
 نکته: قبل از انجام تنظیمات USART فقط یکبار RSTRX و RSTTX را رست می‌کنیم
 کاراکتر در حال ارسال یا دریافت رست می‌شود
 کاراکتر از فرستنده شروع می‌شود
 $AT91C_BASE_US \rightarrow US_CR = 0x4$
 $RSTRX = 1$
 $RSTTX = 1$
 ۳- غیرفعال ساز فرستنده و گیرنده:
 $RSTRX = 0$
 $RSTTX = 0$

با نوشتن یک درستی‌های RXDIS و TXDIS از رجیستر US-CR
 فرستنده و گیرنده غیرفعال می‌شوند.

تذکر مهم: اگر فرستنده در حال ارسال یک کاراکتر باشد و یا گیرنده در حال دریافت یک کاراکتر باشد، درحین کاراکتر فرستنده غیرفعال ساز باید ابتدا کار قبلی (ارسال یا دریافت) کاراکتر مورد نظر کامل می‌شود، پس فرستنده و یا گیرنده غیرفعال می‌شود.

قالب انتقال اطلاعات در USART



① بیت start: طول یک یا یک نیم یا ۲ بیت (هر بیت با یک یک کلاک B-R ارسال یا دریافت می‌شود)
 Start Bit همیشه یک بیت با مقدار صفر است.

② بیت دیتا: دیتای دیتا می‌تواند طبق جدول مقابل ۵ بیت تا ۸ بیت انتقال می‌دهد.

و اینکار با تنظیم بیت‌های CHRL از رجیستر US-MR می‌شود.

CHRL
 ۰ → ۵ بیت طول کاراکتر
 ۱ → ۶ بیت
 ۲ → ۷ بیت
 ۳ → ۸ بیت

Character Length
 طول کاراکتر

نکته: دیتا می‌تواند ۹ بیت باشد و با یک کلاک بیت MODE از رجیستر US-MR طول دیتا ۹ بیت می‌شود و در CHRL تأثیر ندارد.

7



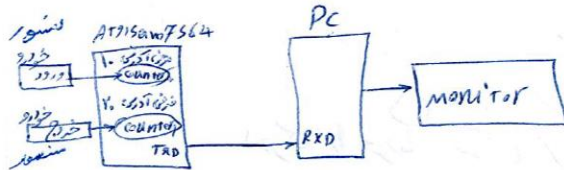
Parity Bit

US-MR

PAR از رجیستر

بیت توان: نوع بیت است و مقدار آن

PAR	Even Parity	Odd Parity	(Space)	(Mark)	No Parity	Multidrop mode
0 0 0	→	→	→	→	→	→
0 0 1	→	→	→	→	→	→
0 1 0	→	→	→	→	→	→
0 1 1	→	→	→	→	→	→
1 0 0	→	→	→	→	→	→
1 0 1	→	→	→	→	→	→
1 1 0	→	→	→	→	→	→
1 1 1	→	→	→	→	→	→



multidrop
برای یادگیری این مد به شکل زیر نگاه کنید:

فرقی کنید بلر ورود و خروج یک بارکنش از کانتر منفی (ورود) و کانتر یک (خروج) استفاده

کرده باشیم چون اطلاعات از یک مسیر مشترک ارسال می شود، لازم است یک بایت آدرس نیز

علاوه بر دیتا ارسال کنیم تا مشخص شود دیتا متعلق به کدام کانال است. با اینکار آدرس دریافتی مشخص می کند دیتا متعلق به کانتر منفی است یا یک (در این سیستم آدرس کانتر منفی برابر ۱۰ و آدرس کانتر یک برابر ۲۰ می باشد) (صفر آدرس ها فرستاده شده است)

multidrop در میکروکنترلر AT91: ~~در میکروکنترلر AT91~~

واحد USART در میکروکنترلر سری AT91 از طریق بیت توان می تواند بین کاراکترهای آدرس

و دیتا فرق بگذارد. برای آدرس بیت توان را یک می کند و برای کاراکترهای دیتا بیت توان منفی

فرستاده می شود. ارسال دیتا به پس فرقی بیت توان منفی است یعنی با ارسال داده از طریق رجیستری آن در رجیستر

US-THR بیت توان منفی ارسال می کند و نیاز به صفر کردن نداریم. ارسال آدرس به اگر بخواهیم آدرس در US-THR بارگذاری کنیم قبل از آن با نوشتن یک در بیت

SEND Address از رجیستر کنترلر US-CR بیت توان یک خواهد شد.

AF

channel status parity Error
 اگر بیت توالی درستی نباشد، گیرنده بیت PARE از رجیستر US-CSR را یک می‌کند که با خواندن رجیستر US-CSR این امر مشخص می‌شود.
 $x = \text{AT91C_BASE_US1} \rightarrow \text{US_CSR} (8080)$

اگر x مخالف صفر باشد یعنی این بیت یک است.
 $\text{if } (x \neq 0)$
 $\text{if } (\text{AT91C_BASE_US1} \rightarrow \text{US_CSR} \& 0x80 \neq 0x80)$
 و با این کار از آدرس یادگرفته بود کاراکتر درستی مطلع می‌شویم.

تذکره: آدرس را خودمان فرض می‌کنیم.
 نکته: با نوشتن یک در بیت RST STA از رجیستر US-CR بیت PARE می‌شود.

بیت STOP: تعدادی یک است.
 Number stop

بک بیتهای NBSTOP از رجیستر US-MR طبق NBSTOP
 0 0 → 1 stop bit
 0 1 → 1.5 stop bit
 1 0 → 2 stop bit
 1 1 → —
 این جدول طول این بیت مشخص می‌کند و همیشه مقدار آن 1 است.

اولین بیت ارسال درستی
 most significant bit

بیت MSBF از رجیستر US-MR اگر یک باشد اولین بیت ارسال درستی با از کمترین بیت خواهد بود و اگر MSBF صفر باشد اولین بیت ارسال درستی کم از کمترین بیت خواهد بود.

نکته: اگر بخواهیم دنیا ارسال را درستی 9 بیت باشد باید بیت MODE از رجیستر US-MR را یک کنیم. (در این حالت دیگر طول کاراکتر ارتباطی با بیتهای CHRL ندارد.)
 9 بیت را در رجیستر ارسال US-THR می‌نویسیم و ارسال می‌کنیم.
 9 بیت دریافت در رجیستر US-RHR

ارسال (عملیات فرستادن) ؟ ^{تعداد ارسال} Transmitter Holding ^{ارسال در هر دو طرف و آستانه ییانات}

مقادیر کاراکتر ارسال را در رجیستر US-THR قرار می دهیم و بعد از این کیفیت رجیستر به (بیت اول)

کیفیت رجیستر ارسال ریخته می شود (اتوماتیک) و در لبه پایینی رونده B.R از بین TxD

بیت ارسال انجام می شود
 $AT91C-BASE-USA \rightarrow US-THR = 26$;

نکته ۱: یک لبه بیت TXRDY از رجیستر US-CSR یعنی رجیستر مقدار US-THR

خالی شده و آماده گرفتن اطلاعات جدید است. و از آن برای کنترل ارسال استفاده می کنیم.

نکته ۲: یک لبه بیت TXEMPTY از رجیستر US-CSR ^{Channel Status} یعنی رجیستر ارسال USART ^{هم US-THR خالی شده}

و $(AT91C-BASE-USA \rightarrow US-CSR \& 0x2) != 0x2$;

تا زمانی که TARDY صفر هیجانبان

با ریختن مقدار جدید مقدار قبلی پاک و TXRDY اتوماتیک صفر می شود و بعد از آن آماده می شود.

الف) در مد اسکرون: در این مد بیت OVER یا نرخ نمونه برداری برابر با ۱۶ برابر B.R ارسال می شود.

نکته: در هشتگونی پالس نمونه برداری START

نکته ۳: اگر بطول نباشد (طول بیت با اندازه استاندارد B.R است) از یک بیت سطح ولتاژ RXD صفر

شود، این سطح ولتاژ صفر را به منزله بیت start تلقی می کنند. ^{نرخ نمونه برداری} Sampling clock

مثلاً برای حالتی که نرخ نمونه برداری ۱۶ برابر B.R ارسال باشد در هشتگونی پالس نمونه برداری

START شناسایی می شود. پس هر یک از بیت های دیتا، بیت تراز و بیت stop هر کدام اینجا

بطور مستقل در ۱۶ پالس از Sampling clock شناسایی می شوند.

یعنی ترتیب برای نرخ نمونه برداری ۸ برابر B.R (ارقاع فوق) ^{بیت B.R} ^{پای ۱۶ عدد ۸} می شود.

4

Receiver Holding
US-RHR

رجیستر دریافت

گرفته پس از دریافت کاراکتر مورد نظر آنرا به رجیستر نگهدارنده دریافت "US-RHR" منتقل می‌کند. (۹ بیت اول)

نکته: با کامل شدن دریافت بیت RXRDY^{اولین بیت} از رجیستر US-CSR یک می‌شود که از آن برای کنترل دریافت استفاده می‌کنیم.

while ((AT91C-BASE-US1->US-CSR & 0x1) != 0x1) ;
RXRDY صفر است اینجا

نکته: با خواندن محتوای این رجیستر اتوماتیک بیت RXRDY صفر می‌شود
 $x = (AT91C-BASE-US1->US-RHR \& 0x1FF)$
بزرگ‌ترین حالت ۹ بیت

خطای OVRE یا overrun error :

اگر قبل از خواندن مقدار رجیستر US-RHR مقدار جدید در رجیستر برنیم بیت OVRE از رجیستر US-CSR یک می‌شود که نشانه خطا و از دست رفتن اطلاعات می‌باشد.

نکته: با یک کردن بیت RSTSTA^{Reset Start} از رجیستر US-CR بیت OVRE یک می‌شود.

ب- گیرنده درمدمیکرو :

در این وضعیت گیرنده در لبه‌های بالارونده clock Baud Rate نمونه برداری می‌کند.
بقیه عملکرد شبیه مدم اسکرون است یعنی همه موارد حقوق و خطای OVRE اینجا تکرار می‌شود.
تنها دقت شده در این حالت در هر لبه بالارونده از B.R یک بیت دریافت می‌شود و سرعت این روشی دقت و سرعت بالا می‌آورد.

خطای بی - توازن :

ای بی توازن؛
درگیر در صورتی که بی توازن در یافتنی باشد توازن حساب شده برابر اندازه باشد بی
PARE از ریشه US-CSR به نشانه خطا یک می شود که در وقفه کاربرد دارد.

CSR از چسب PARE
multidrop در چسب → Parity Error

multidrop → Parity Error

نکات: برادر پاک کرد است PARE باید است RSTSTA از دسترس KS-CR

رایج ترین =
 نکته: قبل از انجام سایر تنظیمات مربوط به USART لازم است پین‌ها را دوباره RSTn و RSTn از رجیستر US-CR
 RSTn, RSTn
 ATn-BASE-US → US-CR = 0x8 → RSTn
 ATn-BASE-US → US-CR = 0x4
 Time -> (حفاظت زمانی) TG
 USART

Time-guard (حفاظت زمانی) ← میزنند

هنگامه بکیر از USART (گیرنده) در درون باسند یا در دیاف کند باسند، سیر ارتباطی لغو شود
بنابراین با تنظیم ۸ بیت = TG فرستنده با گیرنده وصلت می دهد تا دریافت را کامل کند

US-TTGR
 Transmitter Time Guard
 US-THR

نکته: وقتی مقدار ارسال در TXEMPTY و وقتی مقدار ارسال در TXRDY یک شود، اما بیت TXEMPTY و وقتی یک شود گاهی بیت TXRDY (مقدار ارسال) نیز سبزر شود.
در بیشتر ارسال،

درگیرنده ← Time-out (محدت) T_o (در واقع یک عمود افست که در صورت تقاطع با T_o بر روی محور می کشند)
 مدت زمان برای تعداد بالایی که درگیرنده تنظیم می شود تا دریافت سیال کامل گردد. اگر طول مدت دریافت از این
 تعداد بالایی تنظیم می شود. بیت TIME-out از رجیستر US-CSR یک می شود (نشانه پایان محلت)

در این قسمت که می‌تواند سبب اجبار وقفه در USART گردد Receiver استفاده می‌گردد

تذکره: در صورت بروز خطا Time-out دو کار مهم انجام داد:

۱- با نوشتن $STT \rightarrow T_o$ از رجیستر $US-CR$ و سبب T_o را تا دریافت باراکتر جدید متوقف کنیم.

۲- $RET \rightarrow T_o$ و دوباره بارگذاری کنیم.

وقتی $Time-out$ رخ دهد $Reload$ and $Start Time out$



Hamming Error (خطا غالب)

از آنجاکه همیشه بیت STOP باید یک باشد، در صورت نا همخوانی بین فرستنده و گیرنده ممکن است این بیت صفر دریافت شود، در این صورت در گیرنده خطای Framing Error یک شدن بیت FRAME از رجیستر US-CSR اعلام میشود و ایجاد وقفه مربوطه را میکند. تذکره: برای صفر کردن این بیت لازم است بیت RSTSTA را یک کنیم.

↓
در گیرنده US-CR

تعلیقات Break :

در فرستنده به منظور صفر کردن سطح ولتاژ بین TXD انجام میشود.

باید کردن بیت STTBK از رجیستر US-CR فرستنده پس از ارسال کاراکتر
Start Break

فعلی شروع به صفر کردن سطح ولتاژ بین TXD حداقل بطول یک کاراکتر میباشد.

تذکره: زمان ساخت کاراکتر کاراکتر (علا + STOP + تراز + دست بندر)

مجموع زمانی اینجا (طول یک کاراکتر)

نکته: باید کردن بیت STPBK از رجیستر US-CR فرستنده پس از ارسال صفر از بین TXD بطول یک کاراکتر (طول یک کاراکتر را برای ارسال صفر کامل میکند) و دوباره به عملکرد طبیعی باز میگردد.

در گیرنده در حالت طبیعی یک بیت STOP (ت) را در حالت انتظار برای کاراکتر ارسال کننده انتظار دارد. اگر کاراکتر ارسال کننده در وسط سطح انتظار نداشته باشد.

اگر گیرنده بجای انتخاب دیتا، تراز و STOP، صفر دریافت کند، آنرا بعنوان Break در نظر میگیرد. تذکره: برای استفاده کردن RXBRK از رجیستر US-CSR باید مکنز که زمان برای وقفه USART

مکنز که برای کار کردن این بیت لازم است بیت RSTSTA را یک کنیم. از رجیستر US-MR

در این حد کنترل فرستنده در هر سمت درست گیرنده درست مقابل است. Hard ware Hand shaking (دست دادن سخت افزار) (توضیح: RTS در فرستنده و CTS در گیرنده)

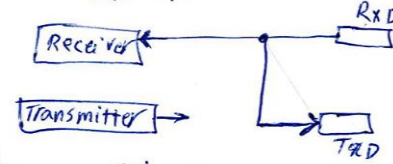
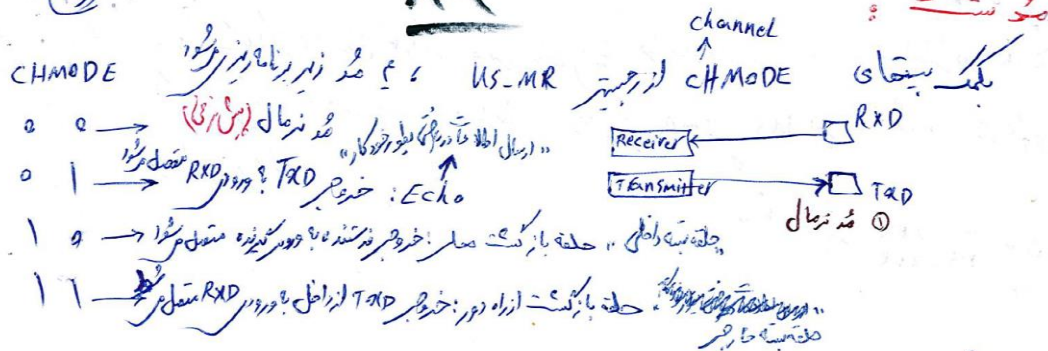


بطوریکه اگر گیرنده از طریق RTS سطح ولتاژ را در CTS صفر کند، مکنز که شروع عملیات ارسال در فرستنده میشود. اگر کاراکتر گیرنده پر شود، RXBUFE یک میشود و گیرنده باید کار RTS و در نتیجه CTS در فرستنده، ارسال را متوقف مکنز.



13

19



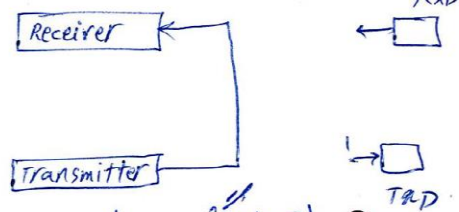
Echo ⑤

"مد ارسال اطلاعات در یک لایه خودکار"

تنظیم واحد فرستنده تأثیر بر TXD ندارد

وبسایت درستی علاوه بر ورودی به گیرنده، اتوماتیک از TXD برگردانده می شود

تنظیم B.R فرستنده نیاز نیست

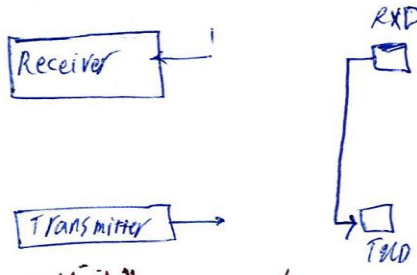


⑤ حلقه بازگشت صاف

"حلقه بسته داخل"

در این حالت به پیکره منبر RXD و TXD نیاز نداریم. (عیناً به نازل)

بازگشت خطا در برد اصلی



④ حلقه بازگشت از راه دور

"حلقه بسته خارج (فرستنده و گیرنده غیرفعال)"

تنظیم B.R در فرستنده و گیرنده تأثیر ندارد

مگر اگر درستی از RXD عیناً از TXD برگردانده می شود

(15)

90

USART1 و با $BR=9400$ ارسال شود

آسترو
مقاله برنامه‌نویسی که ALT Zarei توسط

$$\text{selected clock} = \frac{mck}{\Delta} = \frac{48082185V}{\Delta} = 7.07185V, 120$$

$TXD1 = PA.22$

پیرفرمال A

$$\text{Baud Rate} = \frac{\text{selected clock}}{14 \times CD} = 9400 \Rightarrow CD = 29$$

$RxD1 = PA.21$

پیرفرمال A

over=0 $\rightarrow 14 \times CD$

$CD=29 \rightarrow \text{Baud Rate} = 9424, 2V$

$\Rightarrow \text{خطا} = \left[1 - \frac{9400}{9424, 2V} \right] \times 100\% = 0, 25\%$

تألیف اطلاعات: بیت‌ها، بیت توقف، بیت توان زو

فرع نمونه بردار (B.R) ارسال

```
#include <Atmel/AtmelSAM7S64.h>
unsigned char x[10] = "ALT Zarei";
void main(void)
```

```
{
    start-up
    speed-up
```

```
AT91C_BASE_PMC -> PMC_PCR = (1 << AT91C_ID_US1);
```

```
AT91C_BASE_PIOA -> PIO_PDR = (1 << 21 | 1 << 22);
```

```
AT91C_BASE_PIOA -> PIO_MDER = (1 << 21 | 1 << 22);
```

```
AT91C_BASE_PIOA -> PIO_ASR = (1 << 21 | 1 << 22);
```

```
AT91C_BASE_US1 -> US_CR = 0x28;  $\rightarrow$  RXD  $\rightarrow$  بیت رست اولیه برای ارسال
```

```
AT91C_BASE_US1 -> US_MR = 0x40;  $\rightarrow$  NSTOP=0, NSCLKS=01, CHRL=11, OVER=0
```

```
AT91C_BASE_US1 -> US_BRGR = 0x27;  $\rightarrow$  CD=39
```

```
AT91C_BASE_US1 -> US_CR = 0x40;  $\rightarrow$  TXEN=1 فعال ساز برای ارسال
```

```
while(1)
```

```
{ for(i=0; i<10; i++) {
```

```
while((AT91C_BASE_US1 -> US_CSR & 0x2) != 0x2);
```

```
AT91C_BASE_US1 -> US_THR = x[i];
```

```
} }
```

برنامه‌ار بر روی میکروکنترلر AVR از سیستم جانبی توسط USART دریافت کند. و سپس بر روی پینهای PA₄ و PA₅ ارسال دهد. ~~Sampling clock~~ B.R = 19200

قابلیت‌ها: 1. بیت اطلاعات - 2. بیت توقف و بیت توان قدرت 1. نرخ نمونه برداری 1/8 ارسال B.R

$$\text{select clock} = \frac{\text{MCK}}{N} = \frac{8108480}{N} = 704180, 128$$

TXD → PA.5
RXD → PA.2

$$\text{Baud Rate} = \frac{\text{select clock}}{N \times CD} = 19200 \rightarrow CD = 39 \Rightarrow \text{B.R} = 19200, 128$$

OVER = 1

$$\text{خطا} = \left[1 - \frac{19200}{19200, 128} \right] \times 100\% = 0,5\%$$

```
#include <ATmel/ioa91sam7s64.h>
```

```
void main(void)
```

```
{
```

```
start-up
```

```
speed-up
```

```
AT91C-BASE_PIOA → PIO_PCR = (1 << AT91C-ID_PIOA);
```

```
AT91C-BASE_PIOA → PIO_PCR = (1 << AT91C-ID_US0);
```

```
AT91C-BASE_PIOA → PIO_PER = (1 << 4 | 1 << 11 | 1 << 12 | 1 << 13 | 1 << 14 | 1 << 15);
```

```
AT91C-BASE_PIOA → PIO_OER = ( N );
```

```
AT91C-BASE_PIOA → PIO_PDR = ( N );
```

```
AT91C-BASE_PIOA → PIO_ODSR = 0x0;
```

```
AT91C-BASE_PIOA → PIO_PDR = (1 << 2 | 1 << 5);
```

```
AT91C-BASE_PIOA → PIO_MDER = ( N );
```

```
AT91C-BASE_PIOA → PIO_ASR = ( N );
```

```
AT91C-BASE_PIOA → US_CR = 0x24; RSTRX = 1
```

```
AT91C-BASE_US → US_MR = 0x82250; SYNC = 0, USERXS = 0, CHRL = 0, 1
```

```
AT91C-BASE_US → US_BRGR = 0x27; NBSTP = 1, OVER = 1, PAR = 0, 0
```

```
AT91C-BASE_US → US_CR = 0x0; RXEN = 1, TXEN = 1
```

```
while (AT91C-BASE_US → US_CSR & 0x1) ! = 0x1);
```

```
AT91C-BASE_PIOA → PIO_ODSR = (AT91C-BASE_US → US_RHR & 0xFF) << 1;
```

برای ارسال و دریافت از ATmega8535 توسط USART به صورت قالب زیر و با $B.R = 57400$ به صورت

مد حلقه بسته، داخل ارسال و دریافت و پس از آن PA_4 تا PA_7 پائین داده شود

قالب اطلاعات ۵ بیت دیتا، ۱ بیت توقف و بیت توازن زوج و $S.C = 8 B.R$

۹۵



$$\text{selected clock} = \frac{MCK}{N} = 7007157,125$$

$$B.R = \frac{\text{selected clock}}{N \times CD} \Rightarrow CD \approx 12 \Rightarrow B.R = 57400,72$$

$$\text{خطا} = \left[1 - \frac{57400}{57400,72} \right] \times 100\% = 0,12\%$$

#include <Atmel/ioat91sam7564.h>

void main(void)

{
start-up

speed-up

AT91C_BASE_PMC -> PMC_PCR = (1 << AT91C_ID_PIOA);

AT91C_BASE_PMC -> PMC_PCR = (1 << AT91C_ID_US);

AT91C_BASE_PIOA -> PIO_PER = (1 << 0 | 1 << 1 | 1 << 2 | 1 << 3 | 1 << 4);

N -> PIO_OER = (N);

N -> PIO_OWER = (N);

N -> PIO_ODSR = 0x00;

مد حلقه بسته داخل ارسال و دریافت
پایین داده شود
با دریافت TXD RXD نیاز نیست

AT91C_BASE_US -> US_CR = 0x0C; -> RSTRM = 1
-> RSTTAM = 1

AT91C_BASE_US -> US_MR = 0x8900;

AT91C_BASE_US -> US_BRGR = 0x0D; CD = 13

SVCK = 0x00
USCLKS = 0x01 -> MCKA
CHM = DE = 10 -> مده
CHRL = 0x00
OVR = 1 + 1/4
PAR = 0x00
توازن زوج

AT91C_BASE_US -> US_CR = 0x50; -> RXEN = 1
-> TXEN = 1

while(1)

{
if (AT91C_BASE_US -> US_CSR & 0x2) == 0x2)

AT91C_BASE_US -> US_THR = 0x10;

if (AT91C_BASE_US -> US_CSR & 0x1) == 0x1)

AT91C_BASE_PIOA -> PIO_ODSR = AT91C_BASE_US -> US_RHR & 0x1FF;

۵ بیت ارسال
0x1F