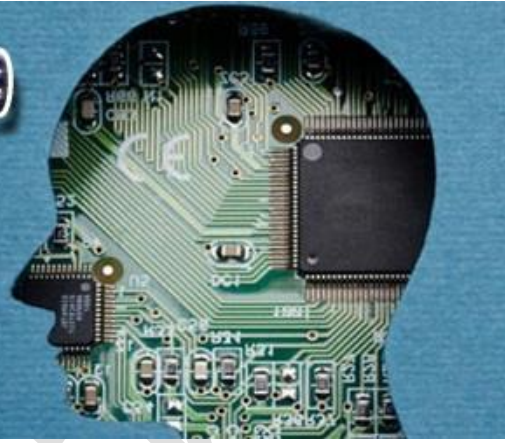


وسایت شخصی علی اکبر خلیلی

آموزش، مقالات، پروژه‌ها

الکترونیک تخصصی و صنعتی، دیجیتال، هوش مصنوعی

www.a-khalili.ir



UART

تراشه های LPC17XX دارای ۴ پورت سریال مجزا از هم می‌باشند.

خصوصیات واحد UART این میکروکنترلر:

طول دیتا 5, 6, 7, 8 بیت داده

توانایی تولید parity

16 بایت رجیستر FIFO برای ارسال و دریافت

پشتیبانی از DMA برای ارسال و دریافت

قابلیت تنظیم خودکار باود

حالت ادرس دهی چند پردازنده‌ای

پشتیبانی از ارتباط مادون قرمز

پشتیبانی از کنترل جریان داده نرم افزاری

پایه‌های مربوط به UART:

Port Pin	Pin Number	PINSEL_FUNC_0	PINSEL_FUNC_1	PINSEL_FUNC_2	PINSEL_FUNC_3
P0.02	98	GPIO	TXD0	ADC0[7]	
P0.03	99	GPIO	RXD0	ADC0[6]	
P2_0	48	GPIO	PWM1[1]	TXD1	
P2.1	49	GPIO	PWM1[2]	RXD1	
P0.10	62	GPIO	TXD2	SDA2	MAT3[0]
P0.11	63	GPIO	RXD2	SCL2	MAT3[1]
P0.0	82	GPIO	CAN1_Rx	TXD3	SDA1
P0.1	85	GPIO	CAN1_Tx	RXD3	SCL1



مراحل پیکره بندی:

- تغذیه: بعد از ریست، UART0,1 فعال بوده و UART2,3 فعال نمی‌باشند که باید با استفاده از PCONP فعال شوند.
- تنظیم کلاک با استفاده از PCLKSEL
- تنظیم نرخ باود با استفاده از رجیسترهای UART
- تنظیم رجیسترهای FIFO
- تنظیم و انتخاب پین‌ها
- تنظیم DMA در صورت نیاز

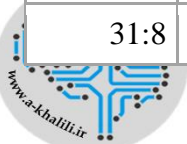
رجیسترهای واحد UART :

هرکدام از رجیسترها، بطور مجزا برای هرکدام از کانال‌ها وجود دارند

Register	Description
RBR (DLAB=0)	رجیستر بافر دریافت کننده اطلاعات. آخرین مقدار دریافتی در این رجیستر نگهداری می‌شود
THR (DLAB=0)	رجیستر نگهدارنده مقدار انتقالی
DLL (DLAB=1)	بایت کم ارزش مقسم فرکانس جهت تنظیم باود
DLM (DLAB=1)	بایت پر ارزش مقسم فرکانس، جهت تعیین باود
IER (DLAB=0)	رجیستر فعالسازی وقفه
IIR	رجیستر نشان دهنده وقفه
FCR	رجیستر کنترل FIFO
LCR	رجیستر کنترل خط
LSR	رجیستر وضعیت خط
SCR	یک رجیستر ذخیره‌سازی ۸ بیتی موقتی برای نرم افزار
ACR	رجیستر کنترل خودکار نرخ باود
ICR	رجیستر کنترل ارتباط مادون قرمز
FDR	رجیستر مقسم کسری. تولید کلاک برای مقسم نرخ باود
TER	رجیستر فعالسازی ارسال، این رجیستر برای خاموش یا روشن کردن ارسال کننده UART ، برای کنترل جریان داده نرم افزاری بکار می‌رود.

رجیستر UxRBR :

Bit	Symbol	Description
7:0	RBR	بافر دریافت ورودی، محتوی قدیمی ترین اطلاعات دریافتی از رجیسترهای FIFO می‌باشد. CPU ، اطلاعات را از این رجیستر دریافت می‌کند.
31:8	•	reserved



دسترسی:

I = LPC_UARTx -> RBR ;

: ساختار رجیستر DLL , DLM

Table 274: UARTn Divisor Latch LSB register (U0DLL - address 0x4000 C000, U2DLL - 0x4009 8000, U3DLL - 0x4009 C000 when DLAB = 1) bit description

Bit	Symbol	Description	Reset Value
7:0	DLLSB	The UARTn Divisor Latch LSB Register, along with the UnDLM register, determines the baud rate of the UARTn.	0x01
31:8	-	Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	NA

Table 275: UARTn Divisor Latch MSB register (U0DLM - address 0x4000 C004, U2DLM - 0x4009 8004, U3DLM - 0x4009 C004 when DLAB = 1) bit description

Bit	Symbol	Description	Reset Value
7:0	DLMSB	The UARTn Divisor Latch MSB Register, along with the U0DLL register, determines the baud rate of the UARTn.	0x00
31:8	-	Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	NA

از این دو رجیستر برای تعیین مقسم مورد نیاز برای تنظیم باود استفاده می‌شود.

: رجیستر UxTHR

Bit	Symbol	Description
7:0	THR	رجیستر نگهداری اطلاعات ارسالی. این رجیستر اولین رجیستر از بافرهای FIFO ارسالی می‌باشد و اطلاعات موجود در این رجیستر، آخرین بایت برای ارسال می‌باشد.
31:8	-	reserved

دسترسی:

LPC_UARTx -> THR = ch ; // send char

: رجیستر UxTER

Bit	Symbol	Description
6:0	-	reserved
7	TXEN	هنگامی که این بیت یک باشد، داده ای که در THR قرار می‌گیرد، از طریق پین TXD ارسال می‌شود. هنگامی که این بیت صفر شود، اگر داده ای در حال ارسال باشد، ارسال آن را به پایان می‌رساند و داده دیگری را قبل از یک شدن دوباره این بیت ارسال نخواهد کرد. این بیت بعد از ریست یک می‌باشد.

دسترسی:

LPC_UARTx -> TER = 0x80 ; //stop sending

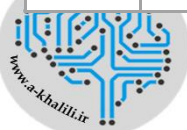
UxFCR : رجیستر کنترل FIFO



Bit	Symbol	Description	
0	FIFO Enable	0	رجیسترهای FIFO در ابتدا غیر فعال می‌باشند و نباید از آنها استفاده نمود
		1	رجیسترهای FIFO فعال می‌شوند و می‌توان از آنها استفاده نمود
1	RX FIFO reset	0	تاثیری روی رجیسترهای FIFO نمی‌گذارد.
		1	تمام رجیسترهای RX FIFO را ریست می‌کند
2	TX FIFO reset	0	تاثیری روی رجیسترهای FIFO نمی‌گذارد.
		1	تمام رجیسترهای TX FIFO را ریست می‌کند
3	DMA Mode select	هنگامی که بیت فعالساز FIFO ست باشد این بیت باعث انتخاب حالت DMA می‌شود.	
5:4	-	reserved	
7:6	RE Trigger level	این دو بیت، تعداد بایت‌های دریافتی را که باعث فعال شدن وقفه یا درخواست DMA می‌شوند را تعیین می‌کند	
		00	یک کاراکتر
		01	۴ کاراکتر
		10	۸ کاراکتر
		11	۱۴ کاراکتر
31:8	-	reserved	

ساختار رجیستر UxLCR :

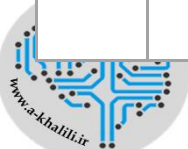
Bit	Symbol	Description	
1:0	Word Length Select	00	طول داده ۵ بیت
		01	طول داده ۶ بیت
		10	طول داده ۷ بیت
		11	طول داده ۸ بیت



2	STOP Bit Select	0	یک بیت برای stop bit
		1	دو بیت stop
3	Parity Enable	0	عدم تولید parity , checking
		1	فعالسازی مولد parity , checking با توجه به بیت 5:4 از همین رجیستر
5:4	Parity Select	00	Odd parity : اگر تعداد یک ها فرد باشد parity تولید می شود.
		01	Even parity : اگر تعداد یک ها زوج باشند، parity تولید می شود.
		10	چسباندن مقدار یک برای parity
		11	چسباندن مقدار صفر برای parity
6	Break Control	0	غیر فعال کردن توقف انتقال
		1	فعال کردن توقف انتقال. با یک شدن این بیت، پین TX مربوط به UART مورد نظر، صفر می شود.
7	DLAB Divisor Latch Access Bit	0	عدم دسترسی به رجیسترهای مقسم یا DLL , DLM
		1	ایجاد دسترسی به رجیسترهای مقسم
31:8			reserved

ساختار رجیستر UxLSR :

Bit	Symbol	Description
0	Receiver Data Ready (RDR)	این بیت هنگامیکه UxRBR داده‌ی خوانده نشده‌ی را نگهداری کند، ست می شود و هنگامی که رجیستر UxRBR خالی باشد صفر می شود.
1	Overrun Error (OE)	این بیت زمانی ست می شود که داده جدیدی وارد رجیستر UxRSR شود و رجیستر UxRBR نیز پر باشد. در این حالت، اطلاعاتی روی رجیستر UxRBR مجدداً نوشته خواهد شد و دیتای موجود در رجیستر UxRSR از بین خواهد رفت. خواندن این رجیستر (UxLSR) باعث پاک شدن این بیت خواهد شد.
2	Parity Error (PE)	اگر بیت Parity از دیتای دریافتی دارای خطا باشد، این بیت ست خواهد شد. این بیت بر اساس بررسی رجیستر UxRBR تأثیر می پذیرد. خواندن این رجیستر (UxLSR) باعث پاک شدن این بیت خواهد شد.



3	Framing Error (FE)	اگر بیت stop از داده دریافتی مقدار صفر داشته باشد، این بیت ست می‌شود. این بیت براساس بررسی رجیستر UxRBR تاثیر می‌پذیرد خواندن این رجیستر (UxLSR) باعث پاک شدن این بیت خواهد شد.
4	Break Interrupt (BI)	هنگامی که بلوک سریال، یک داده که تمام بیت‌های آن اعم صفر باشد دریافت کند این توقف رخ می‌دهد. در این حالت بخش دریافت دیتا در بلوک UART در حالت بیکار قرار می‌گیرد و تا یک داده با تمام بیت‌های یک دریافت نکند فعال نمی‌شود
5	Transmitter Holding Register Empty (THRE)	ان بیت بلافاصله پس از خالی شدن رجیستر UxTHR ست شده و با پر شدن آن صفر می‌شود.
6	Transmitter Empty (TEMT)	اگر هر دو رجیستر UxTHR و UxTSR همزمان با هم خالی باشند، این بیت ست می‌شود. ولی اگر در هر یک از دو رجیستر ذکر شده دیتا وجود داشته باشد، این بیت صفر می‌شود.
7	Error in RX FIFO (RXFE)	اگر هر دیتایی با یکی از خطاهای PE, FE, BI وارد رجیستر UxRBR شود این بیت ست خواهد شد. خواندن این رجیستر (UxLSR) و عدم وجود خطا در رجیسترهای FIFO باعث پاک شدن این بیت خواهد شد.
31:8		reserved

ساختار رجیستر UxLSR :

Bit	Symbol	Description						
0	IrDAEn	0 مد IrDA در UARTx غیر فعال بوده و UARTx در حالت استاندارد و عادی کار می‌کند.						
		1 مد IrDA در UARTx فعال می‌باشد.						
1	IrDAInv	0 اطلاعات ورودی سریال معکوس نمی‌شود						
		1 اطلاعات ورودی سریال معکوس می‌شود						
2	FixPulseEn	در صورت ست شدن این بیت، حالت تثبیت عرض پالس IrDA فعال می‌شود. این خاصیت با استفاده از بیت‌های ۳ تا ۵ از همین رجیستر تعیین می‌شود.						
5:3	PulseDin	اگر بیت ۲ از همین رجیستر ست شده باشد، می‌توان عرض پالس ارسال IrDA را پیکره‌بندی نمود. در حالتی که بیت FixPulse صفر باشد، عرض پالس فرستنده از فرمول زیر بدست می‌آید.						
$Pulse\ Width\ (us) = \frac{3}{16 \times Baud\ Rate}$								
<table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td>000</td> <td>2*TPCLK</td> </tr> <tr> <td>001</td> <td>4*TPCLK</td> </tr> <tr> <td>010</td> <td>8*TPCLK</td> </tr> </table>			000	2*TPCLK	001	4*TPCLK	010	8*TPCLK
000	2*TPCLK							
001	4*TPCLK							
010	8*TPCLK							

			011	16*TPCLK
			100	32*TPCLK
			101	64*TPCLK
			110	128*TPCLK
			111	256*TPCLK
31:6		•		reserved

TPCLK = period of Peripheral CLK

ساختار رجیستر UxIER :

Bit	Symbol	Description				
0	RBR Interrupt Enable	وقفه دریافت اطلاعات ورودی برای کانال متناظر را فعال می‌کند <table border="1"> <tr> <td>0</td> <td>وقفه (اطلاعات ورودی در دسترس است) غیر فعال می‌شود.</td> </tr> <tr> <td>1</td> <td>وقفه (اطلاعات ورودی در دسترس است) فعال می‌شود.</td> </tr> </table>	0	وقفه (اطلاعات ورودی در دسترس است) غیر فعال می‌شود.	1	وقفه (اطلاعات ورودی در دسترس است) فعال می‌شود.
0	وقفه (اطلاعات ورودی در دسترس است) غیر فعال می‌شود.					
1	وقفه (اطلاعات ورودی در دسترس است) فعال می‌شود.					
1	THRE Enable	وقفه خالی شدن رجیستر THR را فعال می‌کند. وضعیت این وقفه از رجیستر UxLSR قابل بررسی است. <table border="1"> <tr> <td>0</td> <td>وقفه خالی شدن رجیستر THR غیر فعال است.</td> </tr> <tr> <td>1</td> <td>وقفه خالی شدن رجیستر THR فعال است.</td> </tr> </table>	0	وقفه خالی شدن رجیستر THR غیر فعال است.	1	وقفه خالی شدن رجیستر THR فعال است.
0	وقفه خالی شدن رجیستر THR غیر فعال است.					
1	وقفه خالی شدن رجیستر THR فعال است.					
2	RX Line Status Interrupt Enable	وقفه وضعیت خط RX را فعال می‌کند. وضعیت این وقفه از رجیستر UxLSR[4:1] قابل بررسی است. <table border="1"> <tr> <td>0</td> <td>وقفه وضعیت خط RX را غیر فعال می‌کند.</td> </tr> <tr> <td>1</td> <td>وقفه وضعیت خط RX را فعال می‌کند.</td> </tr> </table>	0	وقفه وضعیت خط RX را غیر فعال می‌کند.	1	وقفه وضعیت خط RX را فعال می‌کند.
0	وقفه وضعیت خط RX را غیر فعال می‌کند.					
1	وقفه وضعیت خط RX را فعال می‌کند.					
7:3		reserved				
8	ABEOIntEn	فعال‌ساز وقفه موفقیت در تنظیم خودکار باود				
9	ABTOIntEn	فعال‌ساز وقفه Time-out تنظیم خودکار باود				
31:10		reserved				



ساختار رجیستر UxIIR :

Bit	Symbol	Description
0	IntStatus	وضعیت وقفه. توجه شود این بیت با صفر فعال می‌شود. وقفه معوق با استفاده از بیت‌ها ۱ تا ۳ همین رجیستر تعیین می‌شود
		0 حداقل یک وقفه معوق شده است.
		1 هیچ وقفه ای معوق نشده است.
3:1	THRE Enable	نشان دهنده نوع وقفه
		011 خطا یا وضعیت در خط دریافت RLS
		010 دسترسی به اطلاعات دریافتی ایجاد شد.
		110 شاخص Time-out
		001 وقفه THRE
5:4	-	reserved
7:6	FIFO Enable	کپی‌های بیت صفر از رجیستر FCR هستند.
8	ABEOInt	این بیت با پایان موفقیت آمیز تنظیم خودکار باود، در صورتی که وقفه متناظر با آن فعال باشد، ست می‌شود.
9	ABTOInt	این بیت اگر تنظیم خودکار باود دچار Time-out شود و وقفه متناظر با آن فعال باشد، ست می‌شود.
31:10	•	reserved

مثال ۱: برنامه ای بنویسید که اعداد بین ۰ تا ۲۵۵ را با استفاده از UART0 به کامپیوتر ارسال کند.

نکات مهم برای پیکره بندی UART :

1. باید ابتدا نقش پایه‌ها را تعیین کنیم.
2. به رجیستر PINMODE نباید کاری داشته باشیم و وضعیت pull up پایه‌ها در وضعیت پیشفرض قرار داشته باشند.
3. تنظیمات مربوط به FIFO باید انجام شود و حافظه FIFO مربوط به ارسال و دریافت خالی شوند، چون ممکن است از آخرین بار روشن بودن میکرو، داده در آنها باقی مانده باشد.
4. پس از آن باید وضعیت دیتا را تعیین کنیم.
5. در مرحله بعد باید ابتدا فرکانس کاری واحد uart مورد نظر را بخوانیم و بعد متناسب با آن مقدار باود را بسازیم.
6. برای ساختن مقدار باود باید بیت DLAB ست باشد که در هنگام تنظیم وضعیت دیتا این کار را انجام می‌دهیم. و بعد از تنظیم باود دوباره آن را صفر می‌کنیم.

