



دانشگاه صنعتی اصفهان  
دانشکده برق و کامپیوتر

# دستورکار آزمایشگاه طراحی مدارهای واسط

## آزمایش دوازدهم

تهیه کننده: مهران صفایانی

پاییز ۹۵

# آشنایی با پروتکل ارتباطی UART(RS232)

❖اهداف:

- آشنایی با پروتکل ارتباطی UART و امکانات آن
- راه اندازی پروتکل UART در میکروکنترلر LPC1768

❖مقدمه:

پروتکل دیگری که در این آزمایشگاه مورد استفاده قرار می گیرد پروتکل ارتباطی UART است.ارتباط سریال (UART) یکی مهم ترین پروتکل های ارتباطی است که به وسیله ی آن می توان با تجهیزات جانبی و ماژول های الکترونیکی مختلفی ارتباط برقرار کرد. اکثر ماژول های ارتباطی که نیاز به نرخ ارسال اطلاعات با حجم و سرعت کم می باشند از این پروتکل استفاده می کنند.

## ❖واحد سریال UART در میکروکنترلر LPC1768

میکروکنترلر مذکور دارای ۴ پورت سریال مجزا است. شایان ذکر است که UART1 قابلیت ارتباط به صورت مودم و UART0,2,3 قابلیت ارتباط مادون قرمز را دارند.

در تصویر زیر خلاصه ای از قابلیت های ارتباط سریال با UART0,2,3 در این میکرو آمده است :

- Data sizes of 5, 6, 7, and 8 bits.
- Parity generation and checking: odd, even mark, space or none.
- One or two stop bits.
- 16 byte Receive and Transmit FIFOs.
- Built-in baud rate generator, including a fractional rate divider for great versatility.
- Supports DMA for both transmit and receive.
- Auto-baud capability
- Break generation and detection.
- Multiprocessor addressing mode.
- IrDA mode to support infrared communication.
- Support for software flow control.

در این تصویر قابلیت های ارتباطی پورت سریال UART1 آمده است :

- Full modem control handshaking available
  - Data sizes of 5, 6, 7, and 8 bits.
  - Parity generation and checking: odd, even mark, space or none.
  - One or two stop bits.
  - 16 byte Receive and Transmit FIFOs.
  - Built-in baud rate generator, including a fractional rate divider for great versatility.
  - Supports DMA for both transmit and receive.
  - Auto-baud capability
  - Break generation and detection.
  - Multiprocessor addressing mode.
  - RS-485 support.
- 

## ❖ راه اندازی و استفاده از بلوک سریال

جهت راه اندازی و استفاده از UART در میکرو lpc1768 به وسیله رجیسترها باید ابتدا جهت پایه ها با رجیستر PINSEL در مد سریال قرار داد و سپس کلاک این بلوک را با رجیستر PCLKSEL تنظیم نمود و در ادامه با استفاده از رجیسترهای DLL , DLM , FCR و LCR نرخ ارسال ارتباط سریال آسنکرون مشخص می شود و پس از پیکربندی با رجیستر THR می توان داده ها را ارسال و با استفاده از رجیستر RBR جهت دریافت اطلاعات استفاده کرد. در صفحه بعدی لیست کاملی از رجیستر های UART مشاهده خواهید کرد.

با توجه به تعداد زیاد رجیسترها و مشکلاتی در راه اندازی به صورت مستقیم از کتابخانه UART جهت استفاده از ارتباط سریال استفاده می کنیم.

برای استفاده از این تابع ابتدا نرخ ارسال Baud Rate را برابر با ۹۶۰۰ می گذاریم:

```
#define UART0_BPS 9600
```

**Table 271. UART0/2/3 Register Map**

Generic Name	Description	Access	Reset value <sup>[1]</sup>	UARTn Register Name & Address
RBR (DLAB =0)	Receiver Buffer Register. Contains the next received character to be read.	RO	NA	U0RBR - 0x4000 C000 U2RBR - 0x4009 8000 U3RBR - 0x4009 C000
THR (DLAB =0)	Transmit Holding Register. The next character to be transmitted is written here.	WO	NA	U0THR - 0x4000 C000 U2THR - 0x4009 8000 U3THR - 0x4009 C000
DLL (DLAB =1)	Divisor Latch LSB. Least significant byte of the baud rate divisor value. The full divisor is used to generate a baud rate from the fractional rate divider.	R/W	0x01	U0DLL - 0x4000 C000 U2DLL - 0x4009 8000 U3DLL - 0x4009 C000
DLM (DLAB =1)	Divisor Latch MSB. Most significant byte of the baud rate divisor value. The full divisor is used to generate a baud rate from the fractional rate divider.	R/W	0x00	U0DLM - 0x4000 C004 U2DLM - 0x4009 8004 U3DLM - 0x4009 C004
IER (DLAB =0)	Interrupt Enable Register. Contains individual interrupt enable bits for the 7 potential UART interrupts.	R/W	0x00	U0IER - 0x4000 C004 U2IER - 0x4009 8004 U3IER - 0x4009 C004
IIR	Interrupt ID Register. Identifies which interrupt(s) are pending.	RO	0x01	U0IIR - 0x4000 C008 U2IIR - 0x4009 8008 U3IIR - 0x4009 C008
FCR	FIFO Control Register. Controls UART FIFO usage and modes.	WO	0x00	U0FCR - 0x4000 C008 U2FCR - 0x4009 8008 U3FCR - 0x4009 C008
LCR	Line Control Register. Contains controls for frame formatting and break generation.	R/W	0x00	U0LCR - 0x4000 C00C U2LCR - 0x4009 800C U3LCR - 0x4009 C00C
LSR	Line Status Register. Contains flags for transmit and receive status, including line errors.	RO	0x60	U0LSR - 0x4000 C014 U2LSR - 0x4009 8014 U3LSR - 0x4009 C014
SCR	Scratch Pad Register. 8-bit temporary storage for software.	R/W	0x00	U0SCR - 0x4000 C01C U2SCR - 0x4009 801C U3SCR - 0x4009 C01C
ACR	Auto-baud Control Register. Contains controls for the auto-baud feature.	R/W	0x00	U0ACR - 0x4000 C020 U2ACR - 0x4009 8020 U3ACR - 0x4009 C020
ICR	IrDA Control Register. Enables and configures the IrDA mode.	R/W	0x00	U0ICR - 0x4000 C024 U2ICR - 0x4009 8024 U3ICR - 0x4009 C024
FDR	Fractional Divider Register. Generates a clock input for the baud rate divider.	R/W	0x10	U0FDR - 0x4000 C028 U2FDR - 0x4009 8028 U3FDR - 0x4009 C028
TER	Transmit Enable Register. Turns off UART transmitter for use with software flow control.	R/W	0x80	U0TER - 0x4000 C030 U2TER - 0x4009 8030 U3TER - 0x4009 C030

توابع مهم و کاربردی کتابخانه **UART** عبارتند از:

**UARTx\_Init()** : جهت پیکر بندی پورت سریال X است.

**UARTx\_SendByte()** : جهت ارسال یک بایت اطلاعات استفاده می شود.

**UARTx\_GetChar()** : جهت دریافت یک کاراکتر از ورودی استفاده می شود.

**UARTx\_SendString()** : جهت ارسال یک رشته به خروجی استفاده می شود.

**UARTx\_SendChar()** : جهت ارسال کاراکتر به خروجی استفاده می شود.

در ادامه به بررسی بخش اول این کتابخانه و تابع مهم **UART0\_Init** می پردازیم. در ابتدای این کتابخانه با استفاده از عبارت **#define** قسمت های مختلف کلاک را معرفی می نماید:

```
#define FOSC                12000000 //External Clock
#define FCCLK               (FOSC * 8)
#define FCCO                (FCCLK * 3)
#define FPCLK              (FCCLK / 4)
#define UART0_BPS          9600
#define UART2_BPS          9600
```

در صورتی که کلاک اسیلاتور خارجی ما عددی غیر از 12MHz باشد کافی است مقدار **FOSC** را با مقدار فرکانس اسیلاتور خارجی مقداردهی کنیم. در حالت عادی همین مقدار داده شده درست می باشد.

سپس به بررسی تابع مهم و اصلی **UART0\_Init** می پردازیم. با دو دستور زیر استفاده پورت های **P0.2** و **P0.3** را به عنوان پورت های دریافت و ارسال سریال مشخص می نماییم:

```
LPC_PINCON->PINSEL0 |= (1 << 4); /* Pin P0.2 used as TXD0 (Com0) */
LPC_PINCON->PINSEL0 |= (1 << 6); /* Pin P0.3 used as RXD0 (Com0) */
```

سپس با دستورات زیر تنظیمات مربوط به **Stop bit** و بیت **parity** و ... را انجام می دهیم:

```
LPC_UART0->LCR = 0x83; // 1stop bit , 8bit character length ,disable
parity , Enable access to divisor latches
```

سپس با دستورات زیر به محاسبه و مقدار دهی رجیستر های **Divisor latches** می پردازیم:

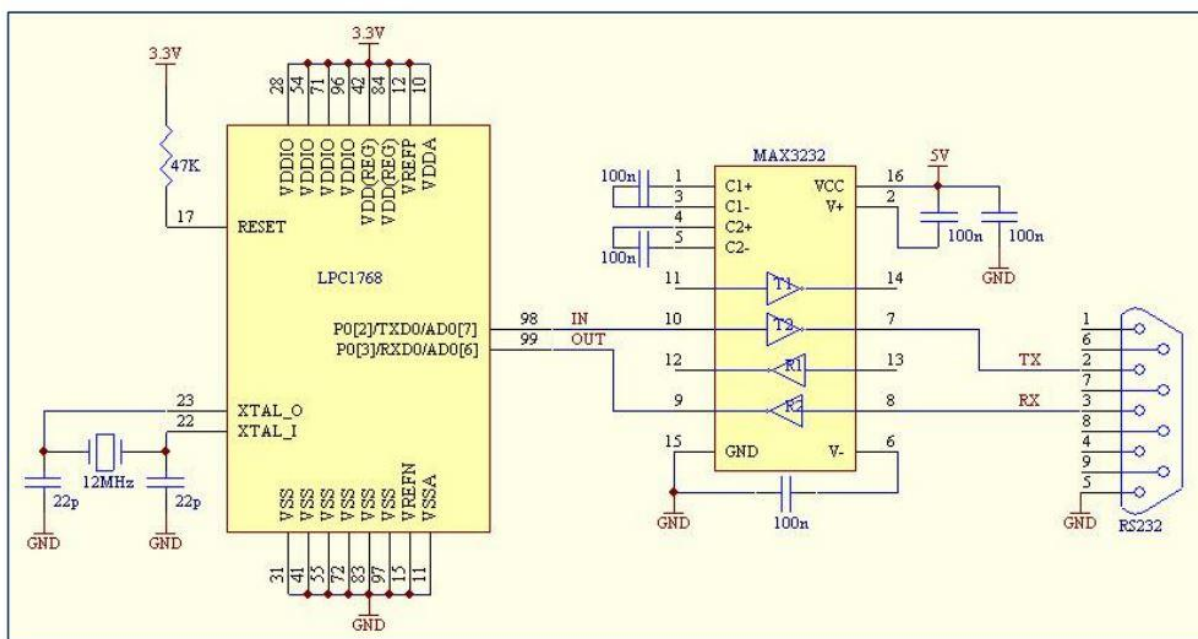
```
uint16_t usFdiv;
usFdiv = (FPCLK / 16) / UART0_BPS;
LPC_UART0->DLM = usFdiv / 256;
LPC_UART0->DLL = usFdiv % 256;
```

سپس با دستورات زیر دسترسی به رجیستر های **Divisor latches** را غیرفعال می نماییم و همچنین به ریست کردن **FIFO** دریافت و ارسال می پردازیم:

```
LPC_UART0->LCR = 0x03;
LPC_UART0->FCR = 0x06;
```

**برنامه نمونه:** برنامه ای بنویسید که اسم و فامیل شما را با پورت سریال یک به کامپیوتر بفرستد و در ادامه به صورت پیوسته یک کارکتر از کامپیوتر بگیرد و دوباره به کامپیوتر بفرستد و در کامپیوتر نتیجه را با یک نرم افزار مانیتور پورت سریال ببیند. برای دیدن نتیجه ارسال و دریافت داده ها از نرم افزار Termite استفاده نمایید و تنظیمات آن را مطابق با تنظیمات کتابخانه استفاده شده قرار دهید.

نکته مهم با توجه به اینکه ارتباط سریال میکروکنترلرها TTL است ، جهت تبدیل این منطق به منطق RS232 و ارتباط بین میکرو و کامپیوتر از چیپ MAX3232 استفاده می شود.



```
#include <LPC17xx.h>
#include <stdio.h>
#include "uart.h"
void Delay (uint32_t Time){
    uint32_t i;
    i = 0;
    while (Time--){
        for (i = 0; i < 5000; i++);
    }
}
```

```

int main(void) {
    int i;
    SystemInit();
    UART0_Init();
    UART0_SendString("\r\nMyName Family\r\n\n");
    while(1) {
        UART0_SendString("your character= ");
        i=UART0_GetChar();
        UART0_SendByte((char) i);
        UART0_SendString("\r\n");
    }
}

```

---

## ❖ دستور کار:

- I. به وسیله ارتباط سریال دو میکرو را به هم متصل نمایید و میکروی اول پیامی را به میکرو دوم می فرستد و میکرو دوم پیام دریافتی را بروی LCD چاپ و پیام دریافت شده را برای میکرو اول ارسال می کند.
- II. در رابطه با وسایل جانبی که به وسیله ارتباط سریال می توان به میکرو متصل نمود تحقیق کنید و سعی کنید یکی از این ماژول ها را با میکرو LPC1768 راه اندازی نمایید. از جمله ماژول های SIM800 و WIFI و RF و...