实验手册：

1. UART串口实验

实验内容：使用串口进行一个回射实验

实验平台：keil5 fs\_11C14开发板

实验目的：

让学生掌握UART的使用

让学生掌握UART如何配置

让学生熟悉UART通信理论

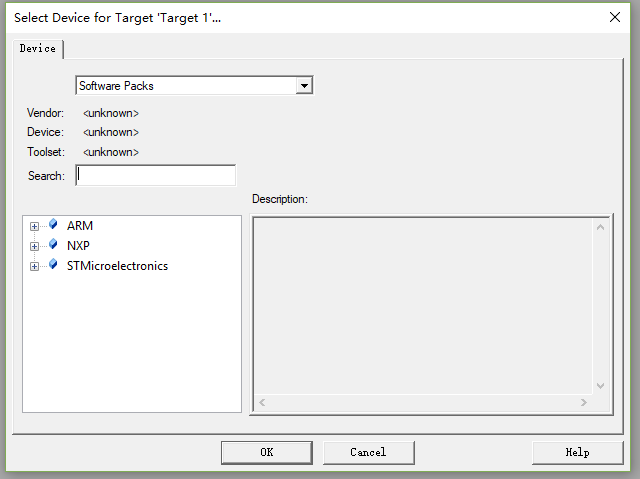
实验步骤：

<1>新建工程目录

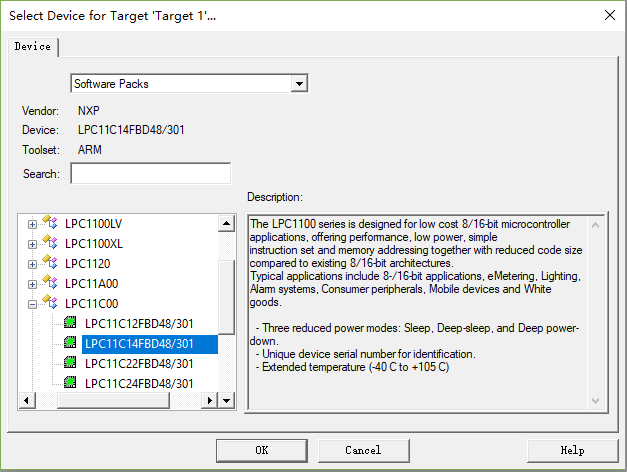
<2>使用keil5新建工程：

新建工程：

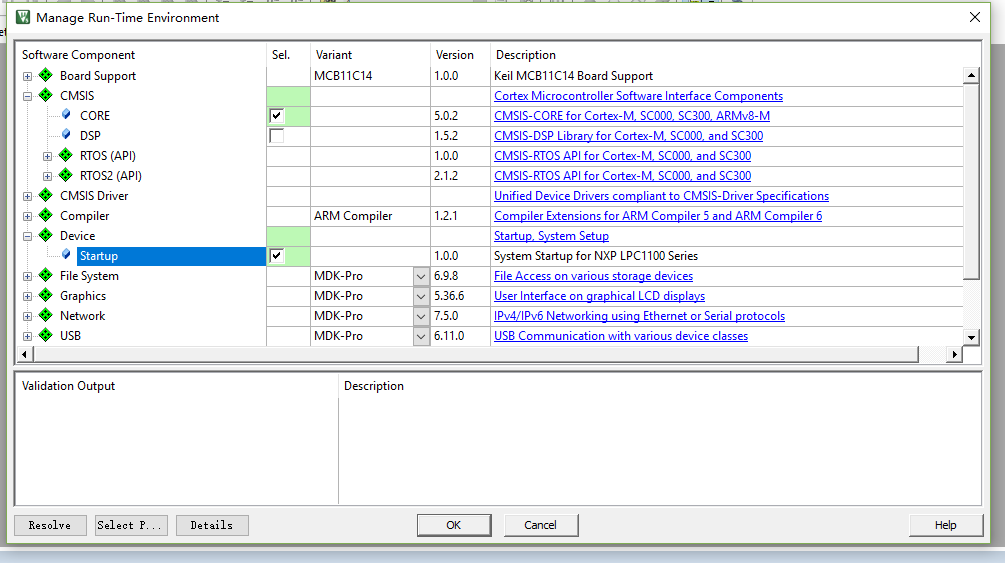
单击Project🡪new µVision Project



选择芯片类型：NXP 11C14：



点击OK，配置CMSIS和Device如下：

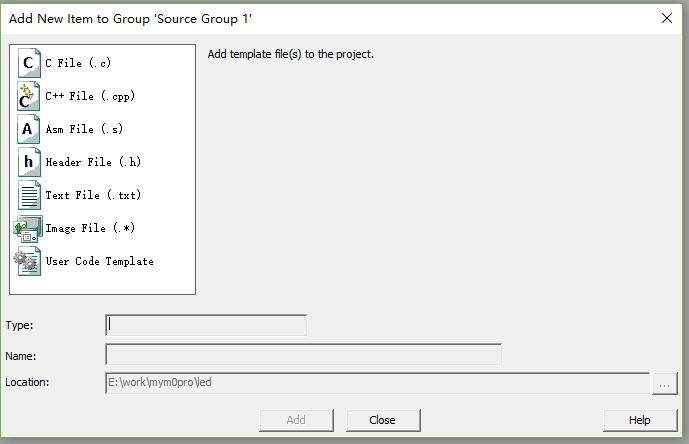


CMSIS：微控制软件接口标准(Cortex Microcontroller Software Interface Standard)

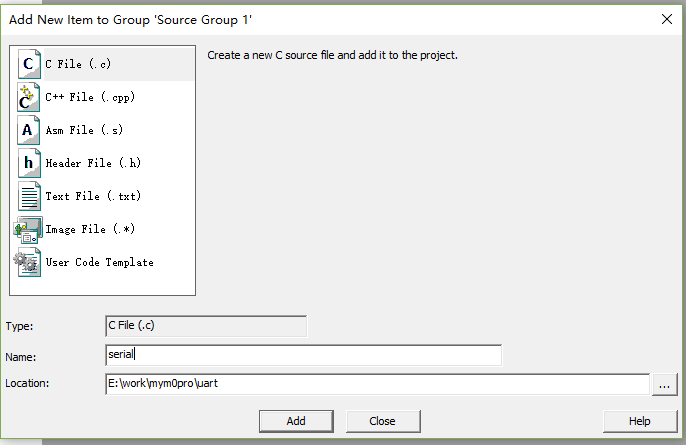
Device：设备

添加新文件：

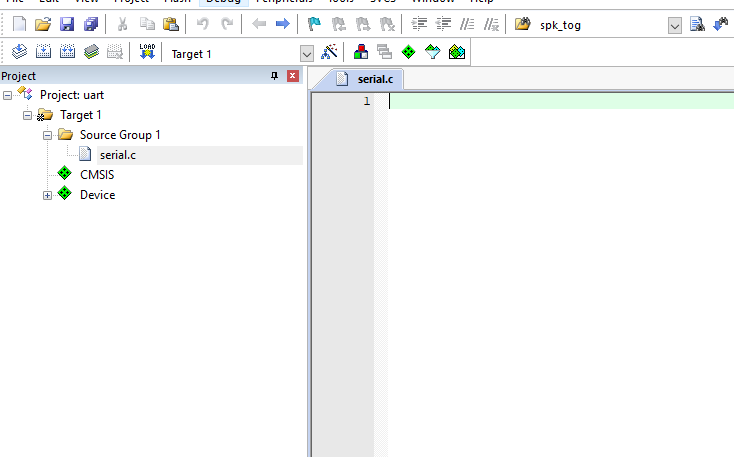
选中Source Group 1右键—>Add New Item to Group ‘Source Group 1’



新建serial.c文件：

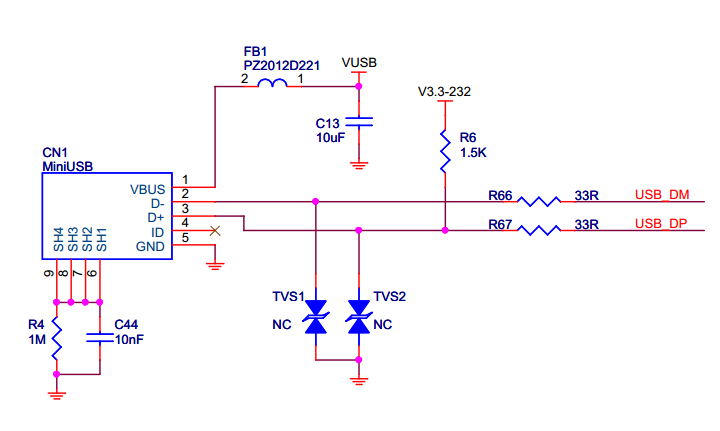


点击add，生成效果如下：



<3>查看板子的原理图，查看CN1的电路：

打开文件fs\_11c14 v6.pdf,找到如下图会看见spk1电路的电路图：



这个CN1有两个功能，我们只关注其串口功能：串口的功能就在这几个端口上：

VBUS：电源

D-:差分信号线

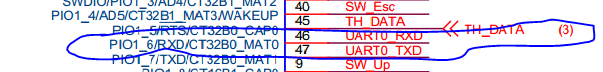
D+:差分信号线

GND：接地

差分信号：差分传输是一种信号传输的技术，区别于传统的一根信号线一根地线的做法，差分传输在这两根线上都传输信号，这两个信号的振幅相等，相位相差180度，极性相反。在这两根线上传输的信号就是差分信号。差分信号的抗干扰能力强，但是有一个缺点就是占用的线路长。

我们关注的是USB\_DM和USB\_DP

然后顺着该图去看cpu管脚图片：

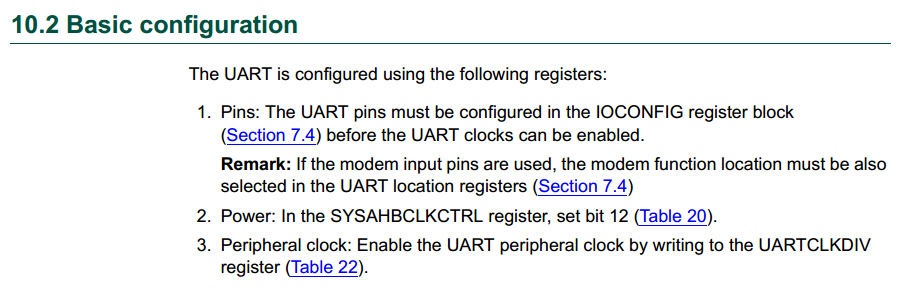


会看到UART的读写管脚分别为：

UART0\_RXD: PIO1\_6/RXD/CT32B0\_MAT0

UART0\_TXD: PIO1\_7/TXD/CT32B0\_MAT1

想要做UART裸机开发，则必须学习其开发流程：开发流程见实验手册：10.2



基础配置：

1. UART 管脚配置
2. SYSAHBCLKCTRL配置
3. Peripheral clock配置

串口参数配置：

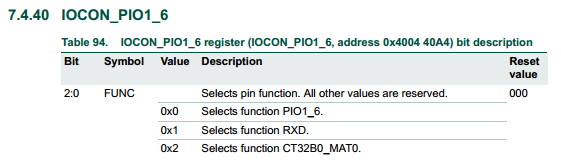
1. 波特率
2. 奇偶校验位
3. 数据位
4. 停止位
5. 流控制

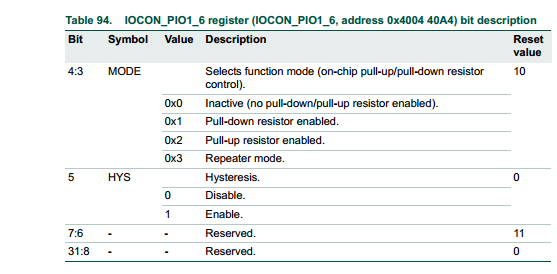
串口读：

串口写：

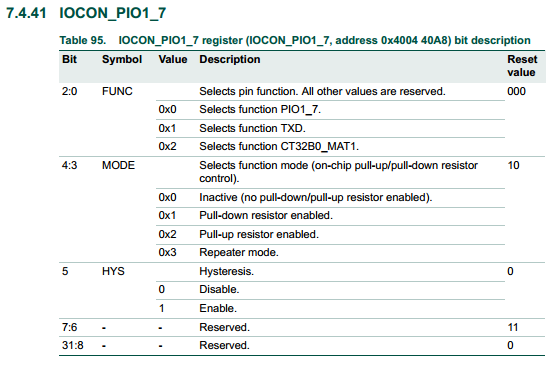
<4>去芯片手册中查找PIO1\_1的绝对地址。

芯片手册文件名为：LPC111x\_C1x User manual R3.pdf



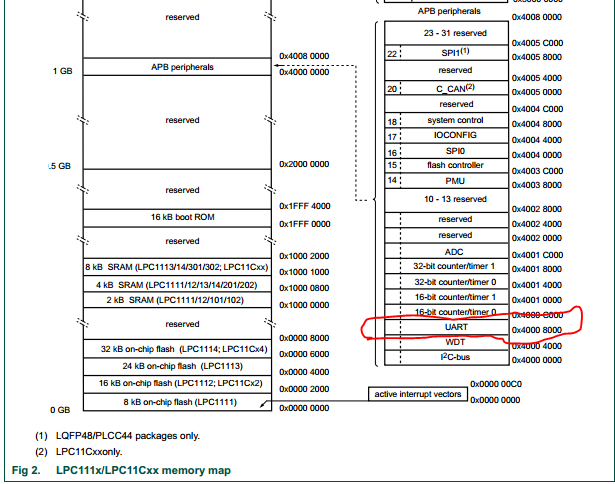


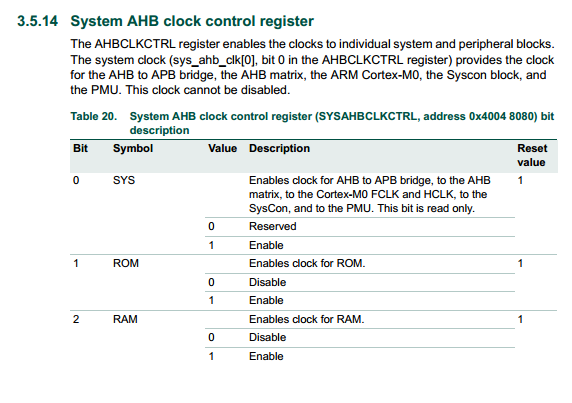
此图见第九章，7.4.40

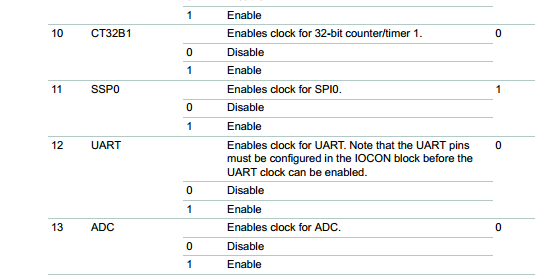


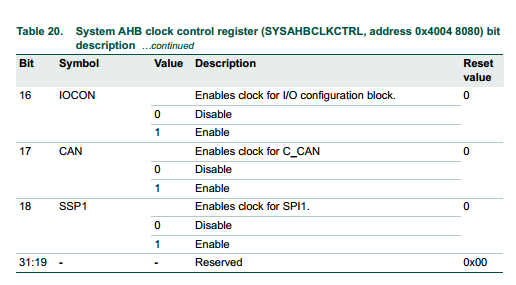
此图见7.4.41

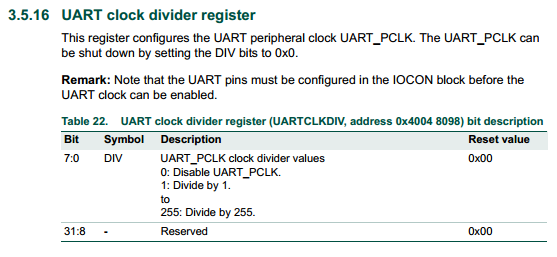
此图在2.2 Memory map

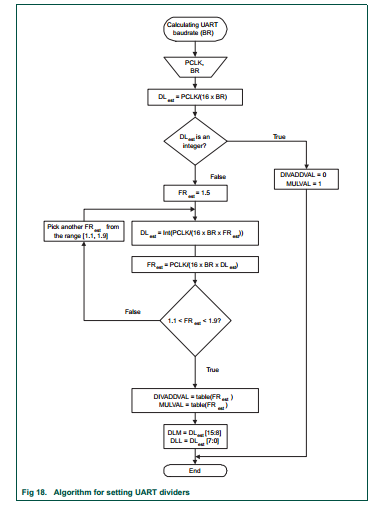


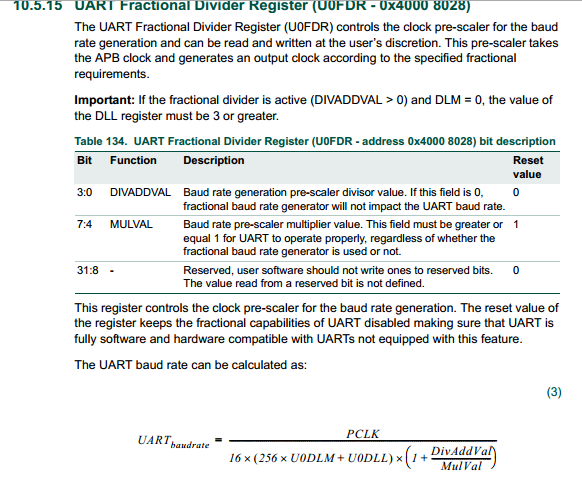


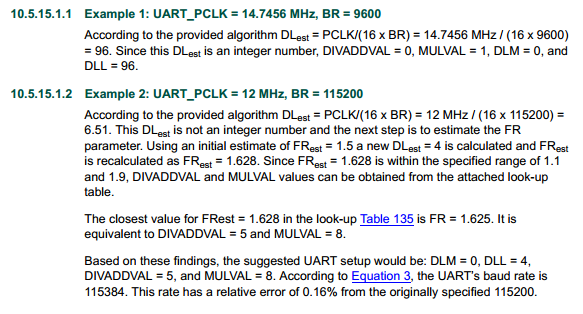












PCLK=48MHz

BR = 115200

DLest = 48000000/(16 \* 115200) = 6.51\*4

DLest 不是一个整数

FRest = 1.5

DLest = Int(48000000/(16 \* 115200 \* 1.5)) = 17

FRest = 48000000/(16 \* 115200 \* 17) = 1.53

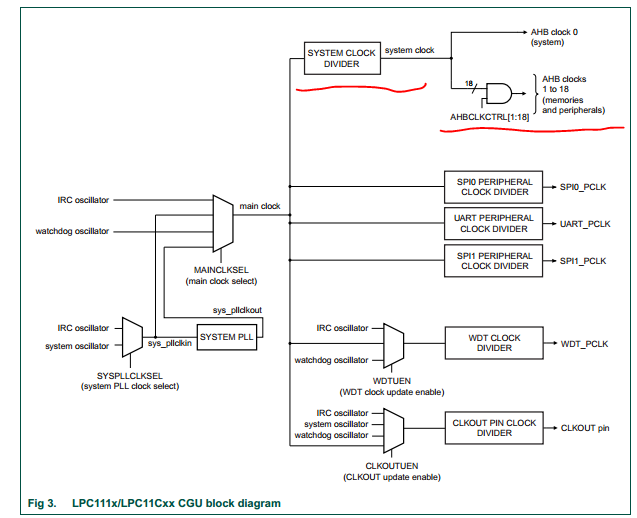
DIVADDVAL = table(FRest) = table(1.627) = 8

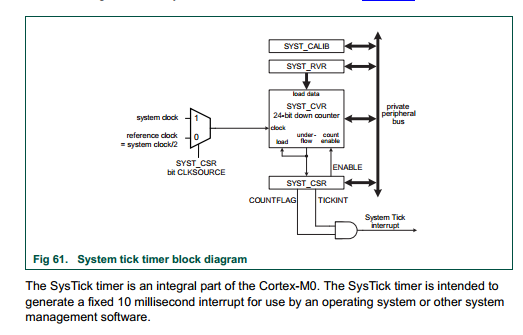
MULVAL = table(FRest) = table(1.627) = 15

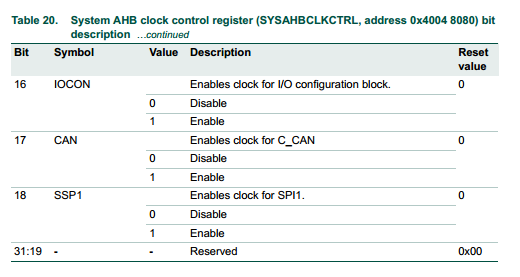
DLM = DLest[15:8] = 0

DLL = DLest[7:0] = 17

注释：DLest = 4 = 0x0016; 高8位为f，低8位为8



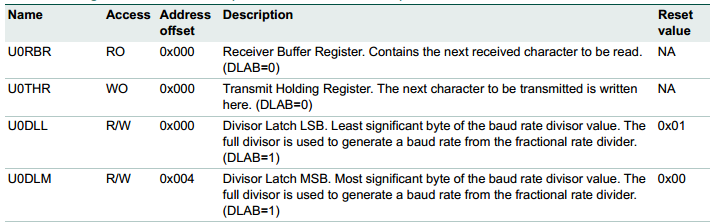


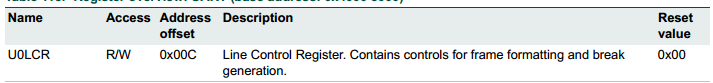


所以我们就得先将系统定时器使能，使能的寄存器为LPC\_SYSCON->SYSAHBCLKCTRL其绝对地址为：0x40048080 其16bit是控制IOCON：使能I/O configuration block的。所以会有以下代码：

LPC\_SYSCON->SYSAHBCLKCTRL |= 1 << 16;

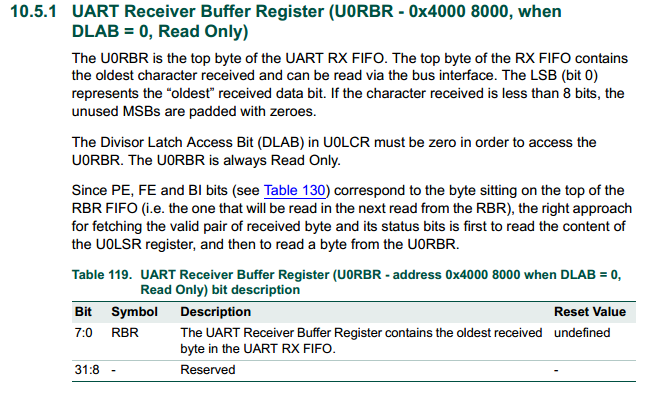
代码如下图所示：



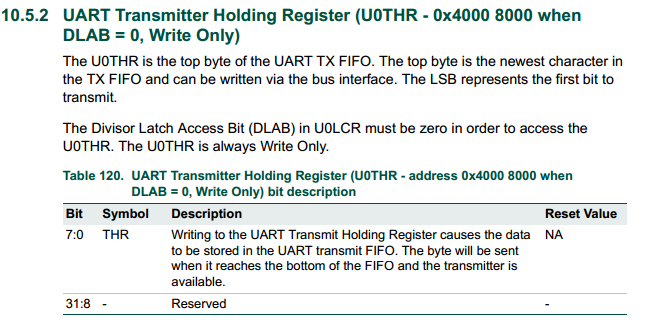




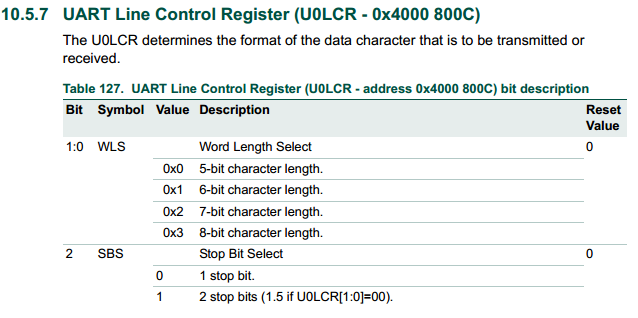


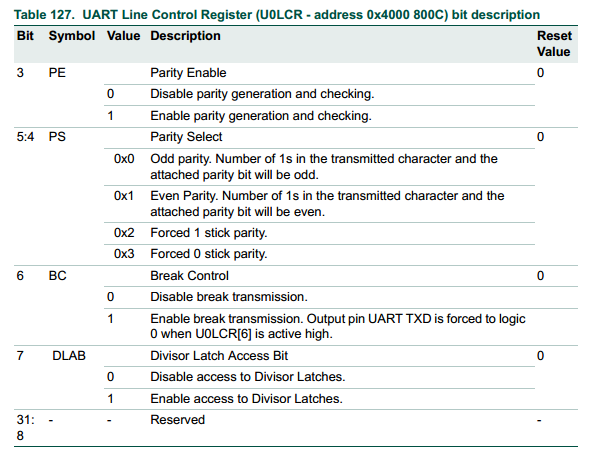


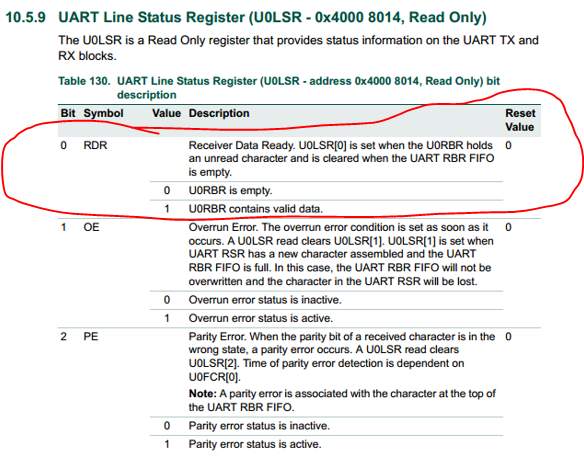
可以从该寄存器中取得串口缓存中的数据。

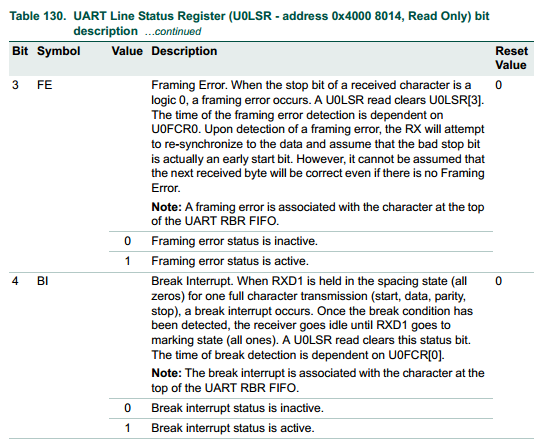


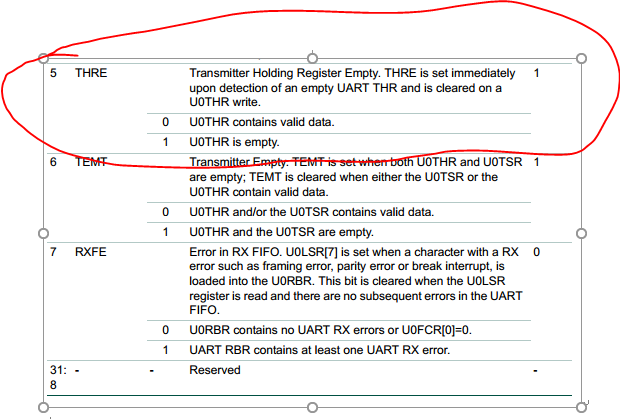
该寄存器用来存储想要发送的字符。

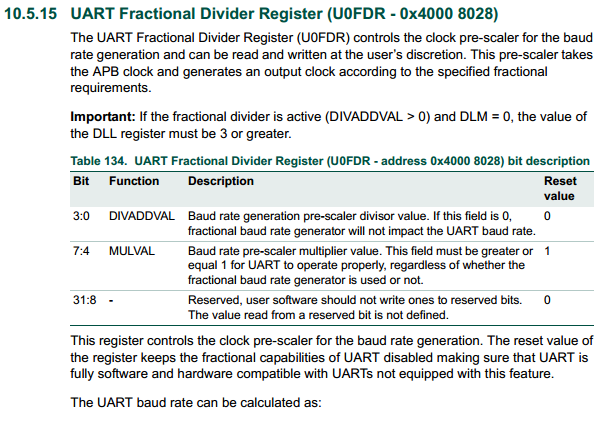


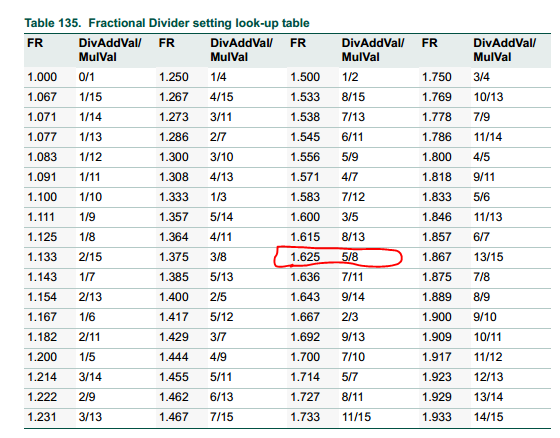


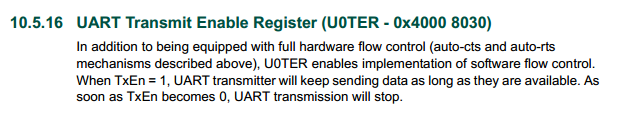


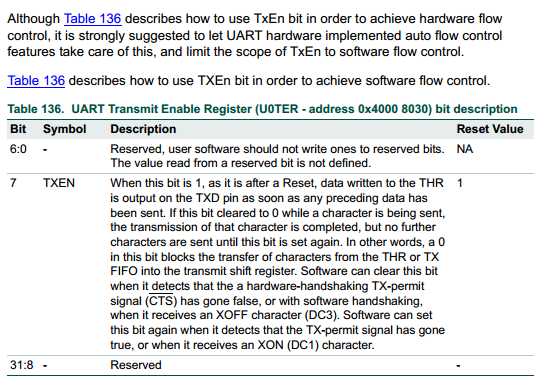




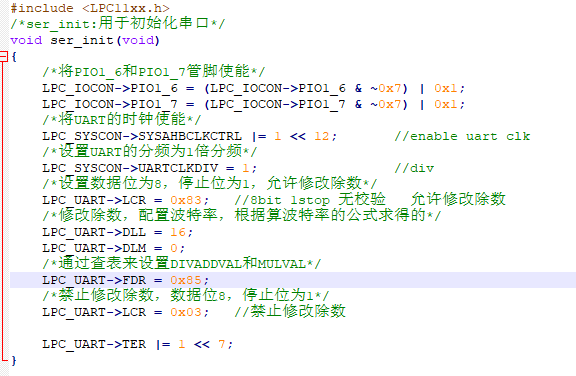








**通过以上的信息，可以用来实现串口的初始化：**



**LPC\_UART->DLL = 17; LPC\_UART->FDR = 0xf8;**

