|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 实验名称 | | USART实验 | | 指导教师 | | 徐斌 | |
| 实验类型 | 综合 | | 实验学时 | 2 | 实验时间 | | 2023/11/10 |
| 姓名 | 赵俊杰 | | | 学号 | 202121018060 | | |
| 一 实验目的与要求  **目的：**UART通用异步收发传输器（Universal Asynchronous Receiver/Transmitter)。UART是一种通用串行数据总线，用于异步通信。该总线双向通信，可以实现全双工传输和接收。作为把并行输入信号转成串行输出信号的芯片，UART通常被集成于其他通讯接口的连结上。在单片机，嵌入式开发系统，大型设备中，串口发挥了极大的作用。本实验重点学习串口的协议内容，掌握STM32中的串口外设的使用方法。掌握在调试过程中，使用可靠的串口进行代码调试。  1、分析串口波特率的产生方式。假如串口时钟为90M，需要得到115200的波特率。计算出USARTDIV的值。（详述计算过程）  2、分析发送流程和接收流程，重要函数需要进行分析。  3、运行串口发送程序。（参考实验5-1串口发送小实验-F429），串口截图。  4、运行串口接收程序。（参考实验5-2串口接收小实验-F429），串口截图。 | | | | | | | |
| **二 实验仪器与器材**  1、电脑  2、软件  MDK521A.exe  芯片支持包：Keil.STM32F4xx\_DFP.2.9.0.pack  STlink驱动(dpinst\_amd64.exe)  CH340驱动  3、硬件  STM32F429阿波罗开发板  ST-LINK-V2下载器  USB Cable | | | | | | | |
| **三 实验内容及步骤**（包括实验原理、步骤、接线图、记录表格、数据处理等内容）  **1. 串口波特率的产生方式及USARTDIV计算过程**  串口波特率的计算方式涉及到USART的时钟源以及分频器的设置。在STM32中，常见的时钟源包括系统时钟（SYSCLK）和外部时钟源（HSE，HSI等）。  波特率发生器的工作原理是将串口时钟频率除以一个称为USARTDIV的预分频系数，然后再除以16（因为每个字节包含起始位、停止位和8个数据位）来得到最终的波特率。  实际波特率 = 串口时钟频率 / (USARTDIV × 16)  USARTDIV=90M/16/115200=48.8  **2. 串口发送和接收流程**  发送流程：  1. 初始化串口：配置串口的参数，包括波特率、数据位、停止位、奇偶校验等。  2. 调用发送函数：使用相关的库函数或直接操作寄存器，将数据写入发送缓冲区。  3. 等待发送完成：等待发送缓冲区为空，表示数据已成功发送。  接收流程：  1. 初始化串口：配置串口的参数，包括波特率、数据位、停止位、奇偶校验等。  2. 启用串口接收中断（可选）：如果需要异步接收，可以启用串口接收中断。  3. 等待接收：使用相关的库函数或直接操作寄存器，等待接收缓冲区有数据可读。  4. 读取接收数据：从接收缓冲区读取数据。  在STM32中，常见的库函数包括：  发送数据的函数：  HAL\_UART\_Transmit()：用于发送数据到串口。  HAL\_UART\_Transmit\_IT()：用于以中断方式发送数据到串口。  HAL\_UART\_Transmit\_DMA()：用于以DMA方式发送数据到串口。  接收数据的函数：  HAL\_UART\_Receive()：用于从串口接收数据。  HAL\_UART\_Receive\_IT()：用于以中断方式从串口接收数据。  HAL\_UART\_Receive\_DMA()：用于以DMA方式从串口接收数据。  **3.** **运行串口发送程序**    **4.** **运行串口接收程序** | | | | | | | |
| **四 实验小结、思考**（包括感想、体会与启示）  **小结：**  1、串口波特率计算：  根据实验要求，串口时钟为90M，需要得到115200的波特率。通过计算，得知USARTDIV的值为约 48.8。  2、发送与接收流程分析：  发送流程：通过初始化串口参数，调用发送函数（如 HAL\_UART\_Transmit()），等待发送完成，可以实现将数据通过串口发送。发送函数的调用和等待发送完成是关键步骤，确保数据按照预期发送。  接收流程：初始化串口参数，启用接收中断（可选），等待接收数据，然后通过接收函数（如 HAL\_UART\_Receive()）读取接收缓冲区的数据。在接收中断方式下，中断服务函数负责处理接收到的数据。  3、运行串口发送与接收程序：  参考实验5-1和5-2的串口发送和接收小实验，在STM32上成功运行了串口发送程序和串口接收程序。通过截图记录串口发送和接收的数据，确保数据的正确性和可靠性。  **感想：**  1. 波特率的精确计算：  在串口通信中，波特率的精确计算至关重要，尤其在高速通信中。通过仔细计算和调整 USARTDIV的值，可以确保实际波特率与目标波特率尽可能接近，提高通信的准确性。  2. 串口通信的重要性：  串口通信在嵌入式系统和单片机应用中扮演着重要的角色。通过学习串口的协议内容和掌握STM32中串口外设的使用方法，我深刻理解了串口在数据交互中的灵活性和实用性。  3. 调试过程中的串口应用：  在实际开发中，使用可靠的串口进行代码调试是一种高效的手段。通过串口输出调试信息，可以实时监测程序的运行状态，加速故障排除过程。  通过本次实验，我不仅对UART通信有了更深刻的理解，还学到了在STM32平台上如何配置和使用串口外设。这些知识对于嵌入式系统开发将会是极为有益的。 | | | | | | | |