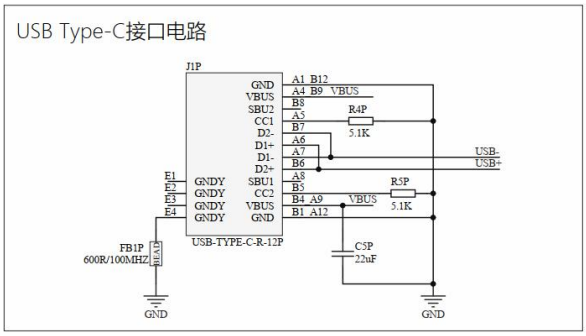
STM32 MCU Development

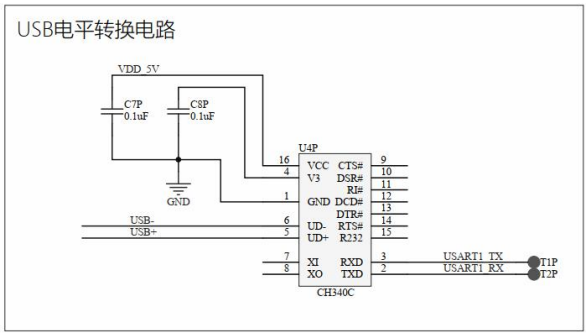
STM32 单片机开发

-- STM32 串口调试输出

目录

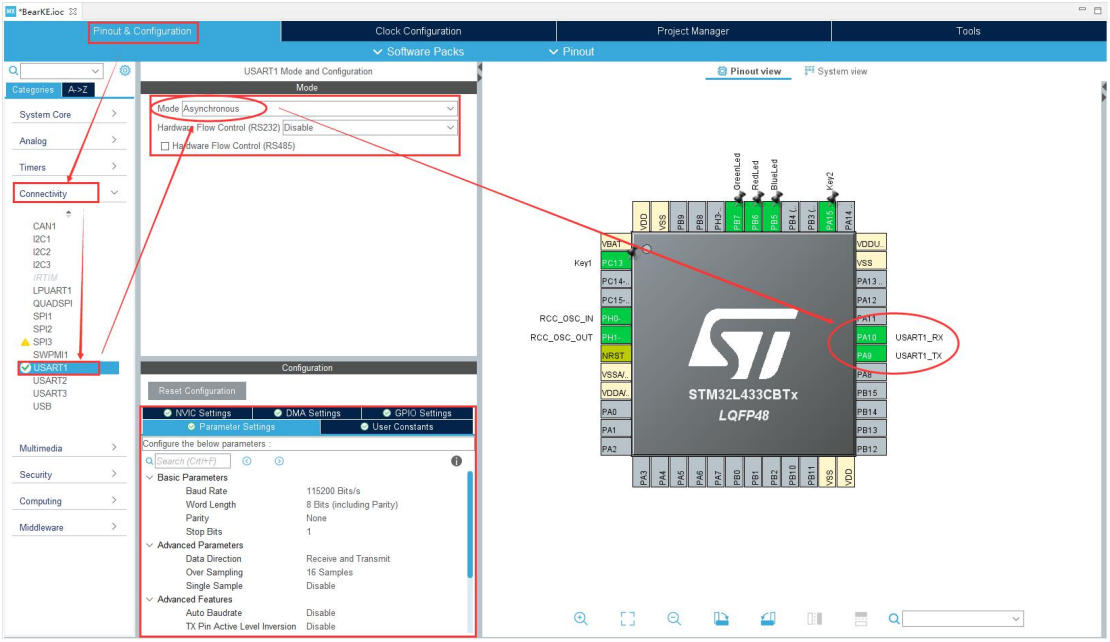
1. 原理图分析





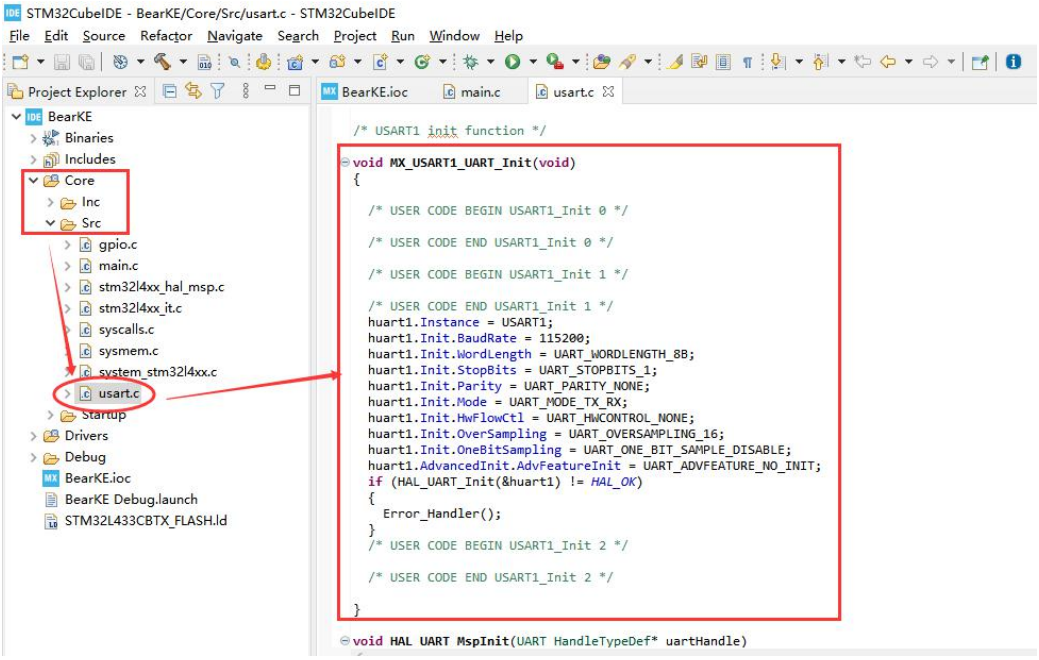
2. printf 实现

2.1. 配置使能串口 USART1



2.2. 自动生产代码

任意位置按 Ctrl+S 将开始自动生成代码。



2.3. 修改 usart.h

修改 usart.h，添加 stdio.h 头文件包含：

```
... ...
/* USER CODE BEGIN Includes */
#include <stdio.h>
/* USER CODE END Includes */
... ...
```

2.4. 修改 usart.c 实现 printf

修改 usart.c 文件，添加 printf()函数的重定向实现：

```
... ...
/* USER CODE BEGIN 1 */
#ifdef __GNUC__
#define PUTCHAR_PROTOTYPE int __io_putchar(int ch)
#else
#define PUTCHAR_PROTOTYPE int fputc(int ch, FILE *f)
#endif
PUTCHAR_PROTOTYPE
{
        HAL_UART_Transmit(&huart1 , (uint8_t *)&ch, 1, 0xFFFF);
        return ch;
}
/* USER CODE END 1 */
... ...
```

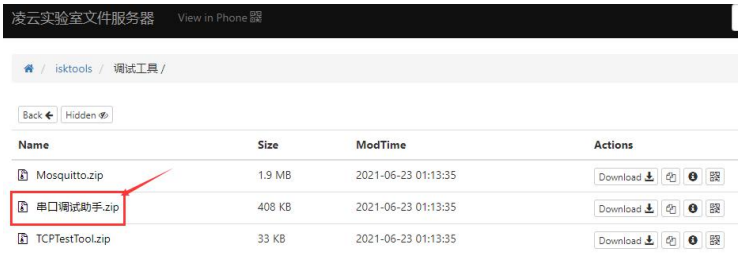
2.5. 修改 main.c 源代码

修改 main.c 文件，添加 printf()函数调用：

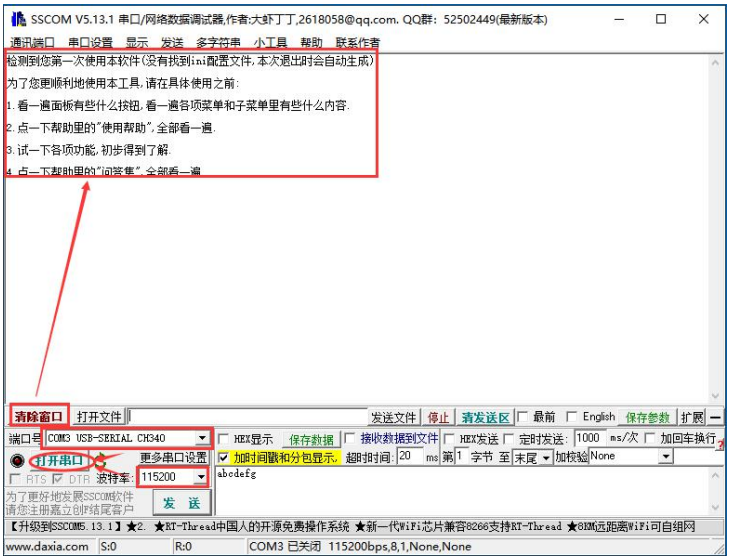
```
... ...
/* USER CODE BEGIN WHILE */
  sysled_hearbeat();
  beep_start(3, 300);
  printf("Start BearKE1 5G NB-IoT Board Example Program v1.0\r\n");
  while (1)
  {
    /* USER CODE END WHILE */

    /* USER CODE BEGIN 3 */
  }
  /* USER CODE END 3 */
}
```

2.6. 运行测试

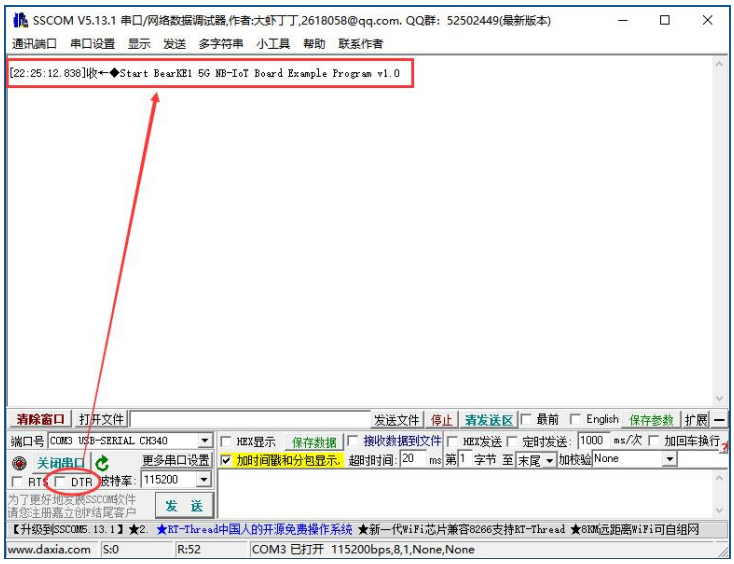
从实验室下面站点下载串口调试助手软件，通过该软件我们可以监听串口输出：
http://studio.iot-yun.club:2211/isktools/%E8%B0%83%E8%AF%95%E5%B7%A5%E5%85%B7



直接运行串口调试助手，打开开发板对应的串口设备，并设置波特率为 115200：



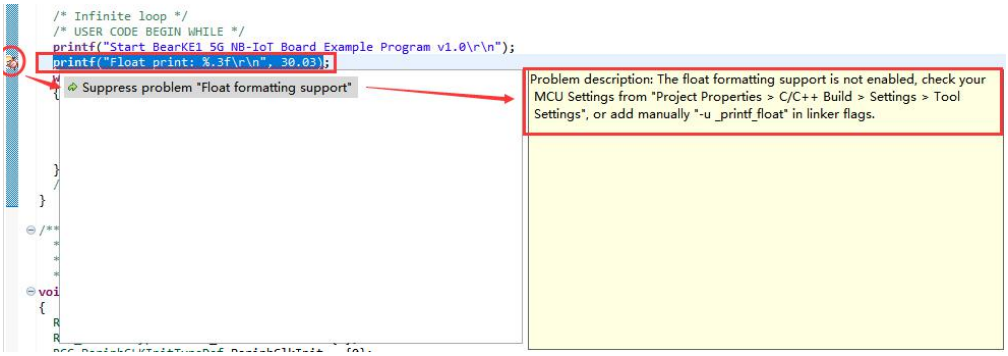
注意不要勾选的 DTR 选项，然后重新烧录程序并运行，我们可以从软件上看到 printf()输出：



2.7. 浮点数输出

由于单片机的硬件资源限制(内存太小)，所以 STM32 单片机里 C 语言标准库的 printf() 并不支持浮点数的输出。今后的代码调试我们可能需要用到浮点数，这里我们实现浮点数打印。

修改一下 main.c 代码，添加浮点数打印，编辑器左侧就会提示默认不支持浮点数，并提示解决方法：



如果要使能浮点数打印的话，则需要添加链接选项“-u _printf_float“ 支持。这个配置好后，提示的错误消失，并能正常打印浮点数了。