# 嵌入式系统实验报告



|  |  |
| --- | --- |
| 实验名称： | Lab2 MDK软件开发流程 |
| 姓 名： | 江姝潼 |
| 学 号： | 2019211653 |
| 学 院(系)： | 计算机学院 |
| 专 业： | 网络工程 |
| 指导教师： | 戴志涛、刘健培 |

2021年 11 月 5 日

# 实验目的

在上一个实验Lab1中，使用的是预先配置好的 example 工程，概览了 MDK下程序的编译、调试，本实验将使用MDK创建自定义的应用程序，并使用MDK常用的调试工具，通过练习熟悉MDK的常用开发功能

# 实验环境

* FS-STM32F407开发平台
* ST-Link 仿真器
* RealView MDK5.23集成开发软件
* 串口调试工具
* PC机Window10 (64bit)

# 实验要求

1.创建新工程，并练习使用各种调试手段对程序进行观察分析。

2.完成实验步骤 2.6，将代码与运行结果截图贴在作业答卷里。

# 实验原理

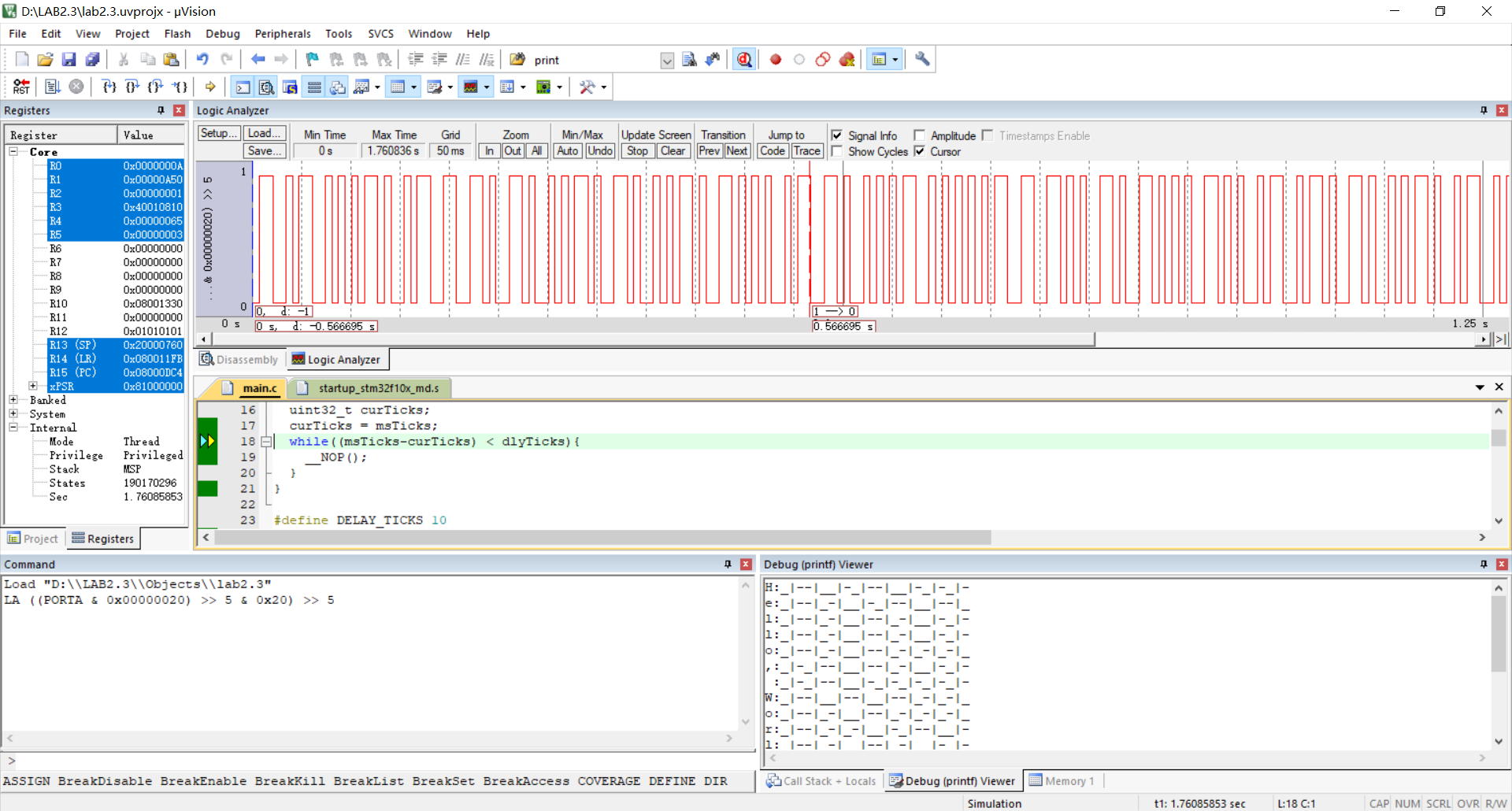
本实验的开发环境是以工程（project）为单位管理整个开发过程的，本实验便要求创建并编译运行工程。创建MDK的 STM32 工程可以通过STM32F1 的标准外设库为模板、或使用MDK的工程创建向导还可以使用 STM32Cube 配置和创建MDK项目。

基本开发流程包括创建新工程、添加代码、编译和链接、设置工程选项、下载程序到芯片/模拟器、执行程序与调试验证，了解了开发流程对我们后面的工程编写奠定了扎实的基础。

本次实验分别用曼彻斯特编码和差分曼彻斯特编码将字符串编成0-1bit的跳变变化，并用LED灯的亮灭直观地看出编码结果。

# 实验步骤

1、输入实验报告中所给的关于曼彻斯特编码的代码，并编译运行查看结果。



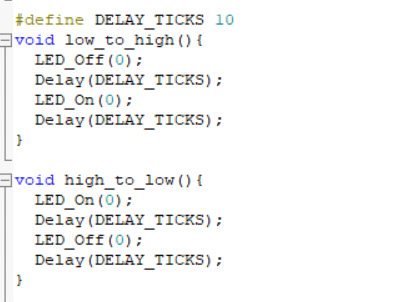
从左下角printf处可以看到程序输出，和逻辑分析器的输出是一致的，可以用于检验编码的正确性。



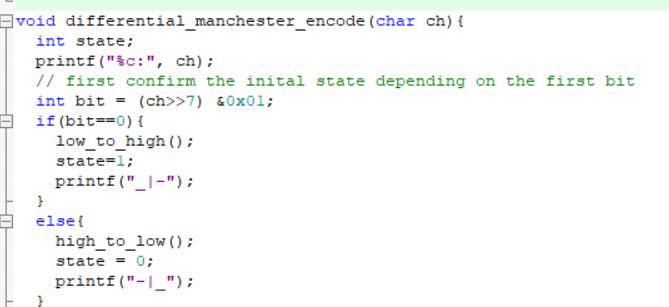


2、下面编写差分曼彻斯特编码方式：

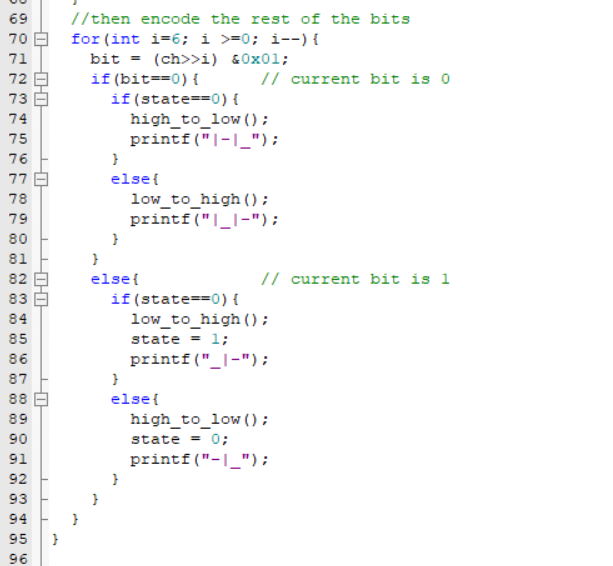
首先将低到高跳变和高到低跳变分别打包成两个函数：low\_to\_high() high\_to\_low()，方便后续程序调用。



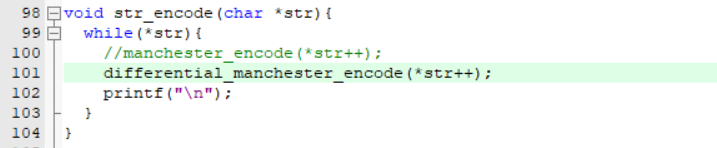
3、差分曼彻斯特编码需要先确定第一位的状态，所以在编码字符时，需要先确定第一位的状态。将第一个比特的数据先单独获取出来，若为0则从低到高变化，1则从高到低变化，后面的就看有没有跳变来决定了。代码如下：



4、接下来编码一个字符的剩余位数，1代表没有跳变，0代表有跳变，用一个state变量存储上一个比特的编码结果，来判断跳变/未跳变时应当处于的状态，并分类讨论，代码如下：

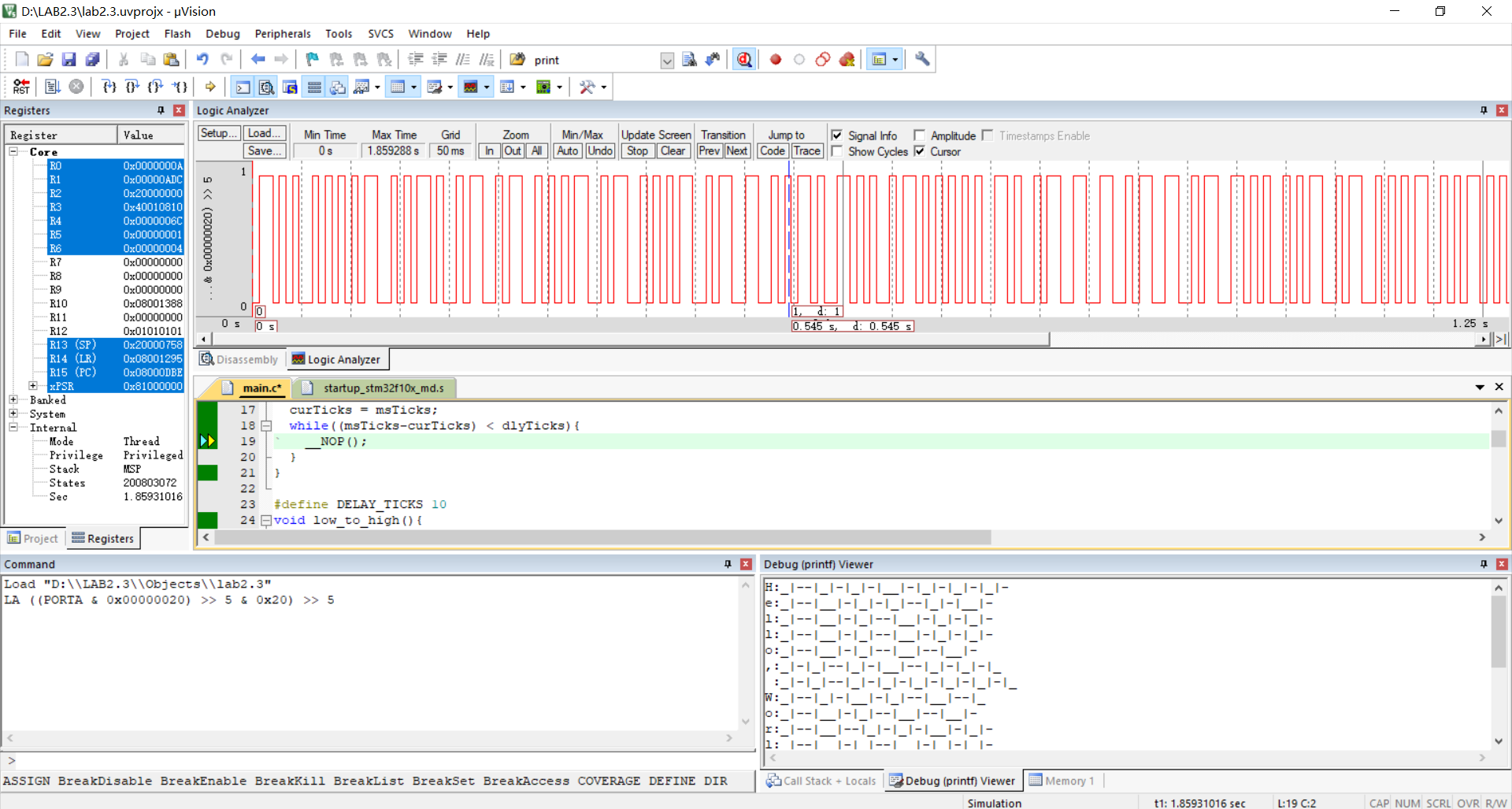


5、在str\_encode函数中调用差分曼彻斯特编码即可：

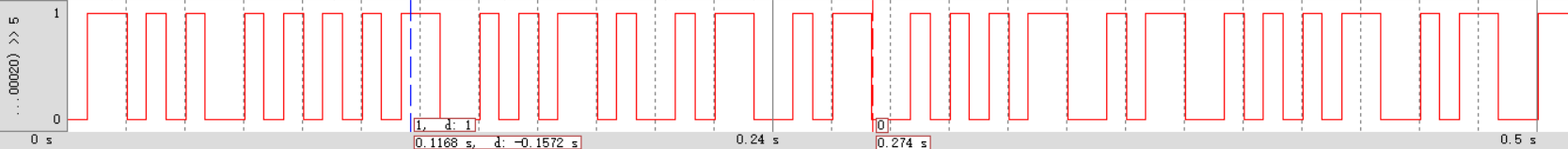


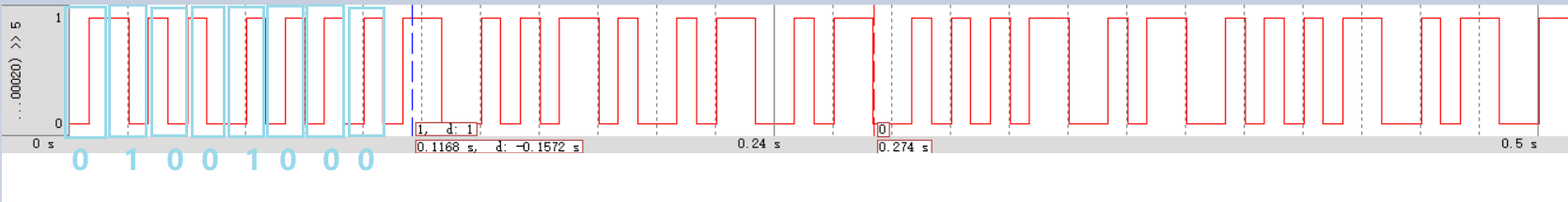
# 实验结果与分析

本实验需要将“Hello,World!\r\n”字符用差分曼彻斯特方式将字符串编码，并用LED的亮灭代表0-1比特，在输出流中输出结果，程序运行结果如下：



下面验证运行结果的正确性，首先看H字母，其ASC2码为72，用二进制表示为01001000，从波形验证中可以看出，其输出结果是正确的。





在输出框也可以看到输出结果:



# 实验总结

本次实验通过创建新工程，并练习使用各种调试手段对程序进行观察分析，了解了MDK下程序的编译、调试、运行与结果分析的过程，熟悉MDK的常用开发功能，为后面的实验奠定了基础。本次通过演示曼彻斯特编码来自己实现差分曼彻斯特编码，让我了解了arm中c语言的编码方式，虽然基本语法相同，但是需要查阅所使用硬件的原始代码文件，去阅读代码注释去了解有哪些功能，并把功能和代码结合起来。例如在本次实验中，我想如果有一个可以获取灯当前状态的函数 (类似于LED.state?)，就不用我自己存储状态了，于是我查阅了LED\_NUCLEO-F103RB.c文件，发现LED中并没有规定这样的函数；但是在文件中我通过注释了解到了所有函数的作用、参数和返回值，这让我之后对硬件所具有的函数有了更多的了解。

