

《嵌入式系统》实验一：温湿度传感器

陈俊哲 2020010964

梁 烨 2020080093

田正祺 2020080095

1 实验内容

使用 STM32MP157 开发板、ST-Link、OLED 显示单元、温湿度传感器、功能拓展板、以及 STM32CubeIDE 实现一个温湿度测量仪。

2 实验过程

2.1 插线和引脚部分

通过 RGB 测试案例，小组搞清楚了 ST-Link、开发板、拓展板、电池的连线方式，同时也找准了连接拓展模块以及其对应的 STM32CubeIDE 中的引脚的方式。按照相同的方式，对 OLED 显示单元和温湿度传感器分别插线并设置引脚。先找到两个拓展模块对应的原理图，找到 OLED 的 SCL 和 SDA 对应的线，以及温湿度对应的 DQ 线。然后将它们连到功能拓展板对应的 P3（有 SDA、SCL 标识）和 P5（D0）。再按照对应的拓展板 P1 接口上的标号，与开发板的 I/O 部分接口进行连接，完成硬件部分的连接。使用到的原理图部分如图 1、2、3 所示。随后在 STM32CubeIDE 中的 Pinout & Configuration 部分，进行引脚的设置。在开发板原理图中，它提供了对应的引脚号，如 P3 中编号 11 的接口对应的引脚号便是 PB7，而小组将温湿度传感器接到的便是编号 11，所以直接在软件中设置 PB7，设置为 GPIO_INPUT、Cortex-M4 状态。对于 OLED，虽然原理图上标识的是 I2C6 SDA 和 I2C6 SCL 对应的 PZ6、PZ7，但小组一开始设置的时候没注意，设置成了 I2C2。后来发现也能正常使用便没有进行修改。时钟部分就完全按照作业文档进行设置的。然后用和 RGB 案例时相同的方式，进行生成代码。

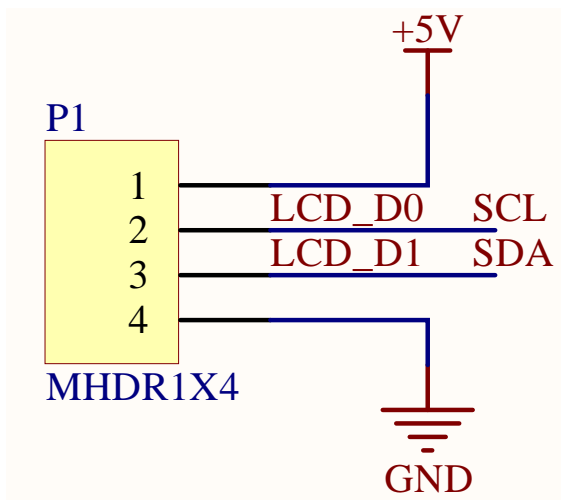


图 1: OLED P1 原理图

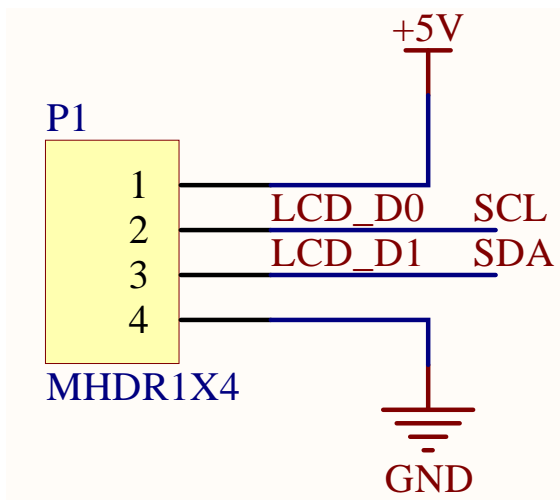


图 2: 温湿度传感器 P1 原理图

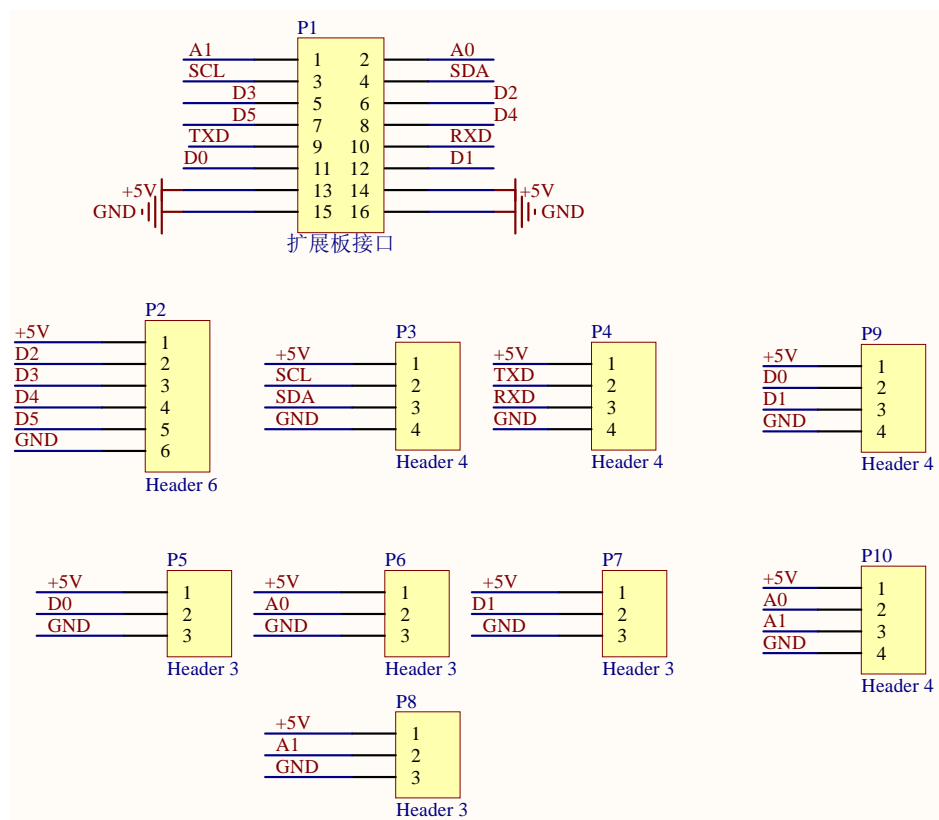


图 3: 开发板 I/O 部分原理图和功能拓展板原理图



图 4: 最终硬件插线情况

2.2 代码部分

代码部分可以分为三个部分：DHT11 函数部分，main.c 中的初始化部分，以及 while(1) 内的部分。

对于 DHT11 函数部分，因为看到最终的 temp 和 humi 引用参数都是 uint16_t，本小组的思路是先正常读取整数部分，然后将整数部分整体向左移位占据高八位，将第八位变成小数部分。然后再 while(1) 对 16 位的 temp 和 humi 再进行数据处理。因此再 DHT11 函数部分，实现的就相对简单，只是再 DHT11_Read_Data 部分将 40bit 的数据分成 5 个 8 byte 分别进行读取。在读取小数位对应的数据前将整数位进行向左移位。DHT11_Read_Byte 函数就是调用 DHT11_Read_Bit 八次，每次将结果向左移一位。比较复杂的是 DHT11_Read_Bit 函数。需要通过查看时序图，来对一开始的两段信号进行忽略。数据 0 和数据 1 的区别在于它们的长短。因为数据 0 一般是 26-28 μ s 的高电平，所以本小组的策略是将开始输出数据后的前 30 μ s 放过去，就看第 31 μ s 的时候。倘若那个时候是高电平，则是 1，反之则为 0。参考的时序图如下：

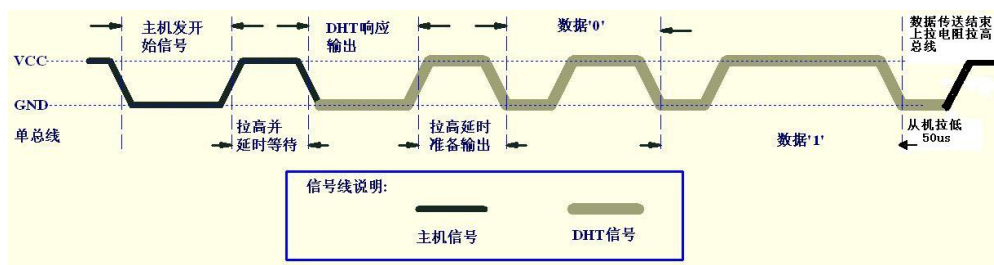


图 5: DHT11 传感器通讯时序图

对于 main.c 初始化部分，主要是对 delay、OLED、DHT11 分别进行初始化，然后再提前设置一些后续需要用到的变量。其中 delay 初始化调用的是 HAL_User_Delay_init 函数，参数传的是和时钟信号对应的 209MHz。OLED 就调用 init 函数，DHT11 则调用 DHT11_WHILE 函数。对于 while(1) 内的代码部分，因为每次调用 DHT11_Read_Data 前都需要调用 DHT11_Check 来检查 DHT11 的状态，所以需要有一个 while 循环在前面确保 DHT11 可以检测到。然后对于 OLED 显示的内容，本小组把显示的部分分成了 7 个部分：3 个英文字符串的显示（“temp:”、“humid:”、“lab1:”），3 个数字部分（温度整数、温度小数、湿度整数），以及一个字符“.”。在这里需要对我们之前 DHT11_Read_Data 函数留下的两个 uint16_t 整数 temp 和 humi 进行处理。它们的高八位就是整数部分，所以在显示整数前要把两个都右移 8 位。对于小数部分则是用 mod 256 的方式得出。最后在每次显示后需要 delay 一段时间，再重新进行调用，我们设置的是 300ms，为了在测试以及录视频的时候更快的看到变化。当然，也不能忘了清空 OLED 来避免一些问题以及调用 DHT11_Rst 函数来进行重置。

2.3 测试部分

OLED 可以显示，调用的 OLED_ShowString 和 OLED_ShowInt32Num 正确无误。温湿度传感器可以接收到正确的数据（手放上去温度慢慢上升、湿度快速上升，手拿开数据会慢慢下降直至稳定状态）。演示视频有两个，一个是有 OLED_Clear 的版本（即最终提交的版本），因为涉及到刷新，所以会一闪一闪的，不是很好拍照。另一个是没有 OLED_Clear 的版本。

3 实验结果

OLED 可以显示, 调用的 `OLED_ShowString` 和 `OLED_ShowInt32Num` 正确无误。温湿度传感器可以接收到正确的数据 (手放上去温度慢慢上升、湿度快速上升, 手拿开数据会慢慢下降直至稳定状态)。

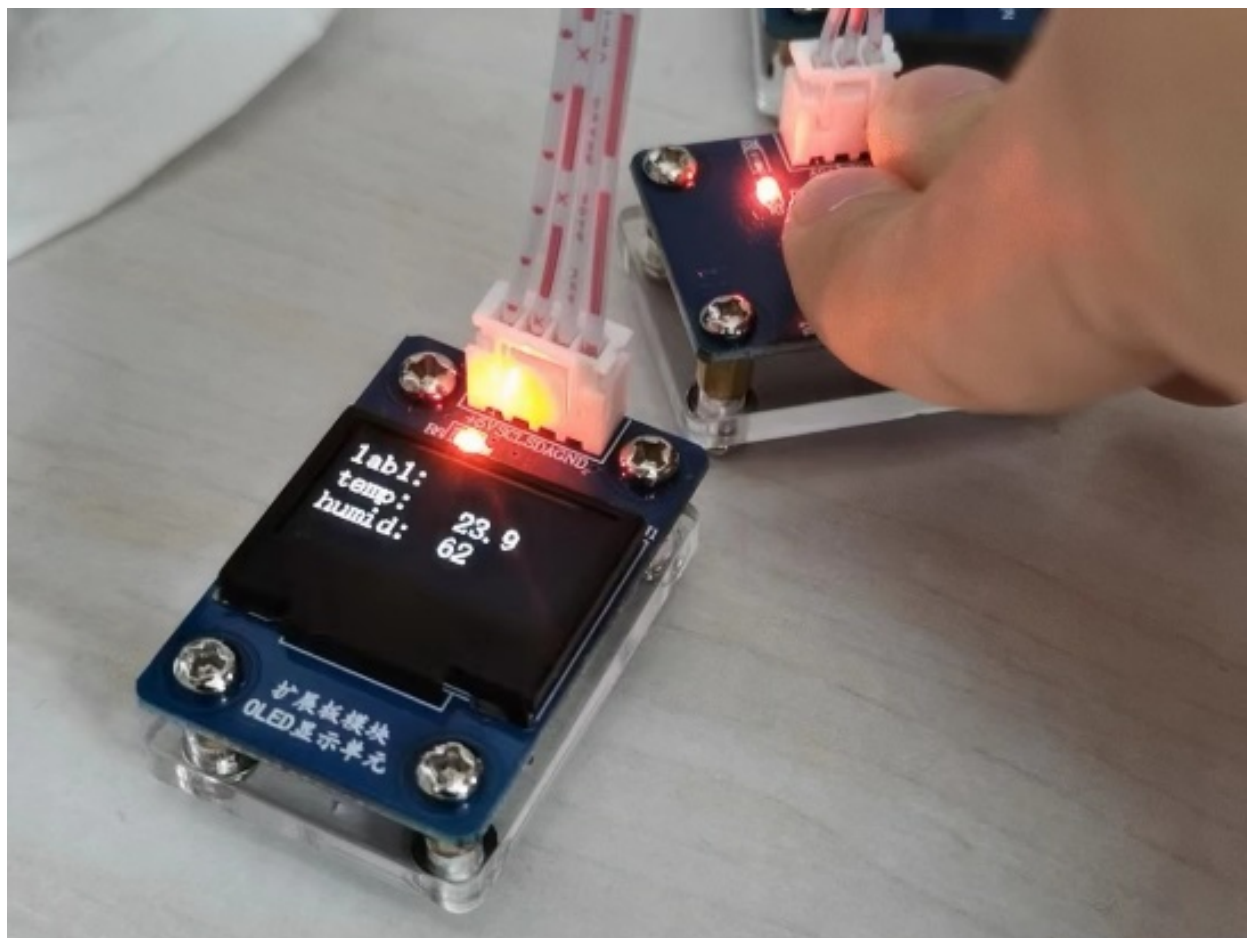


图 6: 实验结果

演示视频有两个, 一个是有 `OLED_Clear` 的版本 (即最终提交的版本), 因为涉及到刷新, 所以会一闪一闪的, 不是很好拍照。另一个是没有 `OLED_Clear` 的版本。

4 过程中遇到的问题及解决方案

本次实验中，小组遇到的问题较多，有些简单的问题在于对设备不是很熟悉，在查阅了课程讨论群中的 FAQ、实验文档、原理图之后便快速解决。下面列出的是一些在实验过程遇到的比较困难的问题。

1. Debug 时卡在 `SystemClock_Config`，运行的时候却没有任何问题。

参考了下面链接中所提供的解决方案：<https://blog.csdn.net/tuxinbang1989/article/details/100826820>，将 `SYS` 中的 `Timebase source` 设为 `Systick` 后即可。同时每次运行后想要 debug 也需要对开发板进行重启。

2. 初始化 `delay` 时，`HAL_User_Delay_init` 不知道怎么设参数

查看了往年的 FAQ，然后看了群内其他同学对这个函数的讨论，最终设置了和时钟设置相同的 209MHz。

3. `DHT11_Check` 在 `while(1)` 里面不通过，但是在初始化阶段的 `DHT11_WHILE` 却可以正常通过。

这个问题非常奇怪，我们求助了助教。助教让我们请教了其他组同学的代码，然后发现在 `main` 这两段代码的部分唯一的区别就只是我们试图在 `DHT11_WHILE` 初始化后对 `OLED` 进行 `show` 操作，想对屏幕显示进行一个初始化。可能是函数互相影响的问题，以至于在 `DHT11_WHILE` 初始化 `DHT11` 后，到 `DHT11_Check` 之间最好没有任何其他拓展模块函数的调用。我们最终去掉了屏幕初始化的部分，选择将 `DHT11` 传感器初始化、`DHT11_Check`、读取后，再统一对 `OLED` 进行操作。

4. `DHT11` 通过 `DHT11_Check` 后只输出 1

尝试了换个传感器，重新配置引脚，结果都没解决。最后发现是读写顺序和调用 `DHT11_Rst` 的问题。

5. 湿度理应是 < 90 的，但手放上去后，会比 90 多一点。

我们之前设置了对湿度数据的判定，即若大于 90 则表示 `DHT11_Read_Data` 失败，从而导致传感器湿度数据大于 90 时会黑屏（因为没有数据了）。后来去掉了这个限制，发现湿度确实会稍微超过 90 一些，大概 92、93。这个应该是正常现象。别的时候传感器也很正常。

5 心得体会

本次实验感觉经历了三个不同的阶段：刚发下实验工具箱的时候的兴奋、开始编写代码测试出问题却找不到问题根源时的迷茫、以及最终做完实验后的成就感。硬件总是能在一开始让人诞生浓浓的兴趣，毕竟可以除了编写代码，可以捣鼓一堆小玩意儿。实验过程中就会有点抓瞎。实验本身涉及到的知识非常多，并且总有一些预料之外的事情会发生，这些都是调试阶段抓瞎的原因。光确保插线和引脚配置没问题就确认了好几次，每次出现预料之外的问题都会觉得会不会是硬件哪里我们配置错了。对于软件部分，我们也是尽可能用 `OLED` 来作为一个显式输出，来查看 `DHT11` 函数返回的值，进行调试。总的来说，在本次实验中，本小组对嵌入式系统有了进一步的认识与了解，对设备进行了实际的操作，收获颇丰。