Mushroom Maintain Manual

BY JESSE

1 main函数

main函数的建立建立在fx-serial.c和fx-serial.h上，这两个文件提供了一些API，使用这些API便可获得PLC中的数据，以及控制相关的开关。

具体的细节可以细看fx-serial.* 相关的源代码。主要涉及到的是优先队列，以及串口通信读写。以后的扩展无需涉及到这里面的代码。

main函数的设计思路是，一个main函数的主线程，同时单开一个用于控制器的控制线程。在main的主线程中，主要的是负责采集以及发送。

1.1 控制器相关线程

我们在控制线程中，接收上层数据中心发送的指令，这个指令包含了**控制器号**以及即将修改的**控制器状态**。我们使用controller_set()来设置控制器状态，同时返回执行后的结果，成功则为1，失败则为0。这个返回的结果同样是通过TCP/IP进行传输的。同时我们刚刚修改的控制器进行状态读取，确保设置是成功的。

在控制器的设计中，可以加入每次执行后的日志记录，大体的方案是使用时LOG宏，编写出一个输出格式，然后将输出显示到标准输入上，同时将该输入保存到本地的文件中。

1.2 数据采集部分

在设计数据采集的时候，主要是通过房间号进行的一个采集，每个传感器的号直接注册到了几个数组中，因此通过遍历数组进行采集，同时把采集到的数据保存的同一个数组中room_info[]。由于同一类型的传感器比较多，比如房间一的温度传感器有五个，此时我们发送出去的温度只需要一个，因此需要对数据做一个整合。

有一点需要注意的是，我们实际读取到的raw数据是没有经过转换的，需要使用对应的公式进行线性变换。大概有如下的五个公式：

$$T_1 = \frac{\text{raw} \times 12}{1000}$$

$$H = \frac{\text{raw}}{10}$$

$$\text{CO}_2 = \text{raw} \times 10$$

$$L = \text{raw} \times 10$$

$$T_2 = \frac{\left(\text{raw} - \frac{30000}{110}\right) \times 110}{1000}$$

这五个公式是用于进行线性变换的，基本每个公式对应一个函数，其中T有两个，这是因为有一个房间的温度值有所不同,具体的可以参看PLC每个寄存器对应的类别（svn上可寻）。

经过换算后的正确值是存放在room_info中。接着是对数据进行分类打包,对应的不同房间都有一个结构体与之对应，我们将数据打包好。进行序列化。序列化的内容参看proto/文件夹里面的相关定义，如若要修改修改,可以在这里面重新创建新的Message。

运行的方式参看svn中DataCollection中readme

1.3 网络通信

在数据经过序列化后，所要做的就是将消息进行拼接发送，消息的定义是一个完整的package。有标头MUSHROOM，数据长度，消息header，以及传感器数据。因此使用memmove进行拼接，将每个部分按顺序拼接到同一个buf中，最后将buf通过send或者safewrite(自定义的安全读写)函数发出.这里需要注意的是不要使用有关任何格式化输出的函数，比如sprintf等。序列化后的字符串，可能包含一些尾零，这样就会将字符串截断,从而变得不完整。

在网络通信的另一个需要注意的地方是,构建一个有能够检测服务器是否在线的一个能力,我们需要是对套接字进行属性设置，使用setsockopt设置SO_KEEPALIVE. 具体的设置参考google或者《unix网络编程》

2 后期维护

在此基础上，可以继续的扩展的是，添加各种实际运行所需要的命令。比如重启命令，获取当前控制板的状态等等。同时在结构上，main函数的设计可以继续的优化，当前的版本的设计还是比较差的，关系比较复杂,错乱。如果有时间可以考虑将main函数重新实现下。

对于上述所说的扩展,可以实现成一个回调函数的样式,由于目前为了测试能通,在main的构思上缺少一些好的构思,这是一个需要考虑的一个地方。

2.1 调试

有关调试的部分，如果以后系统使用我们的系统在跑的话，那就直接远程telnet上去，然后进行调试，这是很方便的，相关的ip以及账号使用在文末给出。如果没有使用我们的系统的时候，为了不每次测试都要跑过去，那么现在我实现了一个本地自定义参数的方法。数据从本地读取，虽然只有一行，但是我们对数据的大小是可控的,这对数据中心的测试是有好处的。

这个方案的实现是在本地重新建立一个txt文件，对其进行读写，里面存放的内容是1～22号传感器的值。其中需要注意的是，前四个是19～22号，代表的是光带的值。在main函数中我是用的rand_data 这个函数来实现的。方式很简单，就是读取文档内容，对文档内容进行分割、转换类型，接着直接跳过calc_data函数，直接使用dumproom_info进行插入。让每次room_info保存对应的房间内的内容。这个实现我是保村在main.c.bak中,如果要使用这个文件时,只需将原来的main.c和这个互换下位置。

2.2 硬件

硬件上，实验室有一块fx1s的PLC但是没有FT232R USB转串口线，以及一根SGI九孔线，如果要使用这个，可以考虑到淘宝上购买一个，同时可以自行接线，如果不清楚这根线是啥样的，可以到东湖的ARM板上看,接线方式见如下的图片：

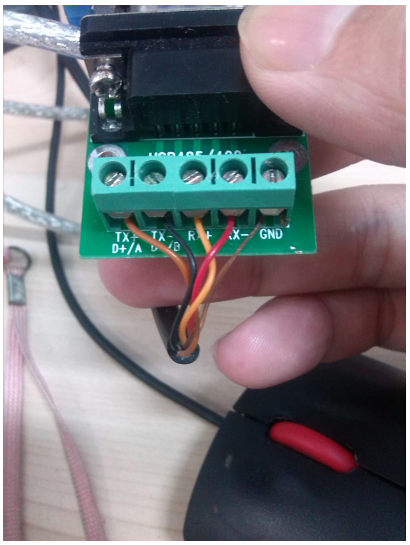

图 1. 接线方式

同时要注意的是去东湖调试时，在插SGI口的时候，要先关闭掉开关，这是必须要注意的。

3 一些参数

1. ARM板的IP：10.28.92.26 网关：10.28.92.254 子网掩码：255.255.255.0 网关：61.155.18.30
2. 传感器读数：灭菌间温度0-120 其他温度：-30~80 湿度0~100 光照0~10000 co2浓度 0~10000
3. 控制器编号：

   进风：

   50、53 、56、64、72

   排风：

   51、54、57、65、73

   温控(空调)：

   52、55、58、66、74

   加湿器：

   59、67、75

   光照：

   60、61、62、69、70、71、77、78、79

   抑制机：

   63、68、76

   培养一 排风二 80

   培养二 排风二 81

培养三 排风二 82

控制状态为：1 - 开 0 -关

4. 传感器编号：

房间	温度	湿度	光照	CO2
培养一	3、4、5、6、7	8	19、20、21、22	9、10
培养二	11	12	-	13、14
培养三	15	16	-	17、18
灭菌间	1	-	-	-
接种间	2	-	-	-

表格 1. 传感器编号