# 雷霆MicroPython固件开发的工作评估

## 工作内容评估

1. 固件更新需要使用到esptool进行固件程序烧录，与原本的模式一样；
2. 需要设计功能库、接口库（由于接口的独特性，普通库硬件控制会影响到设备硬件，要考虑提醒用户不能未经过考虑就使用通用MicroPython库）
3. 代码文件的下载，需要下载到MicroPython的文件系统里面。

Adafruit MicroPython Tool (ampy) - Utility to interact with a CircuitPython or MicroPython board over a serial connection.（安装命令：pip install adafruit-ampy）

1. IDE在线调试设备的方法：使用python命令行（终端为USB-UART，可以参考pycom的VScode插件工具Pymakr）工作方式执行相关代码，甚至是单语句测试执行。
2. 需要在系统上编程实现交互逻辑，这里的代码可能使用底层系统的C语言编写。需要访问MicroPython的文件系统，实现多程序程序选择执行，所以要能通过底层C语言控制MicroPython的运行来代替命令行的执行功能，还要能修改启动脚本boot.py或main.py，甚至自定义启动逻辑。
3. 在雷霆上重写MicroPython的固件程序，有些交互逻辑和功能可以重定义，所以需要规划清晰，从用户体验出发，定义好产品应用。

## 二、工作重点、难点：

1. 固件：原来的硬件设备驱动程序都是在Arduino上编写，移植到基于ESP-IDF的MicroPython固件，需要大概2周的时间；
2. 固件：底层系统的分析和优化，并做相关改动以适应我们产品的需求，需要约1周时间；
3. 固件：编写MicroPython的虚拟机底层代码，并测试相关命令，需要约2周时间；
4. 产品：重新优化相关功能，设计产品应用、系统交互，并编码实现，需要1周时间；
5. 产品：需要预留1周时间进行产品调整；
6. IDE：搭建平台环境需要1周时间，实现相关固件烧录、运行脚本、烧录脚本；
7. IDE：重写语句块生成代码，需要2周时间，并实现语句调试运行的功能；
8. IDE：实现代码编程、图形编程的切换功能，需要2周时间；

### 三、可行性分析：

从功能上来讲，这个思路是可以执行的，市面上基于ESP32模块的产品均支持MicroPython的功能，可参考的产品较多。存在的疑点在算法执行效率上面，由于MicroPython代码执行效率低，进行算法编写时会有一定的局限性，根据市场来讲，使用我们的产品进行大规模算法的概率很低；而且部分常用算法可以在系统底层使用C语言内置。