## 串口助手、图像传输协议暨协议参数限制说明文档

### 串口助手说明文档

Git仓库地址：<https://ggit:ohhggit@git.ohhere.xyz/serial_assistant>

默认分支：master

使用说明：

1. 安装`Python 3.6+`
2. 安装`requirements.txt`中的依赖
3. 设置`read\_serial.py`中的常数参数
4. 运行`read\_serial.py`
5. 关闭在控制台命令行中输入`Control+C`即停止通信，并将代码写入`data`目录下的`get\_image.c`源代码文件
6. 查看`data`目录下对应文件夹内的的`\*.bmp`图片和`get\_image.c`源代码文件
7. 调用`get\_image.c`源代码文件中的`pixel \*\*get\_image(unsigned int image\_index)`函数获取指向图像二维数组`pixel image[IMG\_HEIGHT][IMG\_WIDTH]`的指针，该函数具有时不变性。

其他注意事项及说明见仓库下README.md文件。

### 图像传输协议说明文档

1. 协议的假设及对图像的要求：
   1. 注意图像的像素个数**必须**是`8`的倍数。
   2. 传输图像时，协议完全占用对应串口，该串口不得用于传输其他数据，否则图像将无法解析。
   3. 协议仅能传输二值化后的灰度图像。
2. 协议的传输规则（编码方案）：
   1. 传输每一帧图像前或，传输帧头标志0xABAAABAAAB，再开始传输图像。
   2. 传输每一帧图像时，自上到下依次传输各行的像素。
   3. 传输每一行像素时，自左至右依次传输各个像素。
   4. 传输每个像素时，若该像素的值大于二值化阈值，则传输比特1，否则传输比特0。
   5. 每帧图像中，传输的比特不重叠地形成若干个字节，以字节为单位传送，若图像中出现字节0xAA，则将其替换成0xABAAAAAB。
3. 参考解码方案：
   1. 将字节流以帧头标志0xABAAABAAAB为分隔符，切割成若干个不含分隔符地字节数组。
   2. 对任意字节数组，将其中的字节段0xABAAAAAB替换成字节0xAA，形成解码后的字节数组。
   3. 若解码后的字节数组的长度为图像像素数的八分之一，则该数组为合法的图像帧（比特数组）。
   4. 将合法的图像帧（比特数组）中的比特按行切割，形成与图像大小一致的二维比特数据，该二维数据即是原图像。
4. 编码实现和相关函数：
   1. LQMT9V034.h
      1. IMAGEH：图像高度。
      2. IMAGEW：图像宽度。
      3. IMAGE\_SIZE\_BY\_8：图像像素个数的八分之一，即表达二值化图像的所需的最小字节数。
   2. LQMT9V034.c的void UARTSendPicture(uint8\_t tmImage[IMAGEH][IMAGEW])函数：发送图像tmImage。其中的变量threshhold指定从串口发送图像时，二值化的阈值。
5. 解码实现：见上述“串口助手说明文档”一节。

### 协议参数限制说明文档

1. 图像的像素个数**必须**是`8`的倍数。
2. 即使某些时间段内没有正在传输的图像帧，用于传输图像的串口也不可传输其他数据，而应保持空闲，否则可能导致图像无法正常解析。