**Verisilicon**

**Ble Tool Designe Notes**

**Table of Contents**

[1 Document Information 3](#_Toc529261618)

[1.1 Revision Status 3](#_Toc529261619)

[2 目录文件结构介绍 4](#_Toc529261620)

[3 GUI模块介绍 6](#_Toc529261621)

[3.1 Main page modules 6](#_Toc529261622)

[3.2 Message log page modules 7](#_Toc529261623)

[3.3 Local device page modules 8](#_Toc529261624)

[4 软件框架 9](#_Toc529261625)

[4.1 主框架 9](#_Toc529261626)

[4.1.1 线程框架 9](#_Toc529261627)

[4.1.2 参数传递 9](#_Toc529261628)

[4.2 线程模块 10](#_Toc529261629)

[4.2.1 GUI主线程模块 10](#_Toc529261630)

[4.2.2 uart接收线程模块 10](#_Toc529261631)

[4.2.3 event解析线程模块 10](#_Toc529261632)

[4.2.4 acl数据发送线程模块 10](#_Toc529261633)

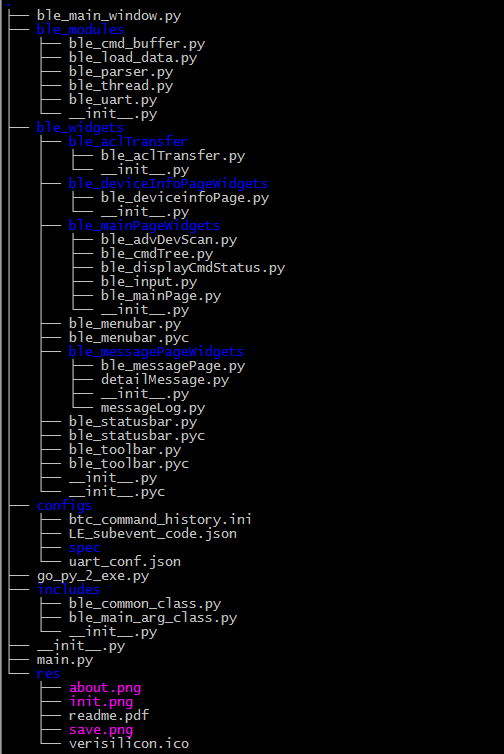
# Document Information

## Revision Status

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Version** | **Date** | **Description of Change** | **Prepared by** | **Reviewed and Approved by** | **Date** |
| 0.1.0 | 2018-11-10 | First version | Jianjun Yang | Jianjun Yang | 2018-11-10 |

# 目录文件结构介绍

如下图所示：



函数入口main.py

**Includes目录**：里面主要包含头文件的定义，其中ble\_main\_arg\_class.py为工具传递的主要的类，里面包含了所有的GUI模块handle已经其他的非GUI的handle。

**Ble\_modules目录**：包含非GUI的模块操作

ble\_cmd\_buffer.py： 加载cmd默认参数的模块

ble\_load\_data.py：加载cmd参数的模块，如cmd有多少个参数，每个参数的大小以及参数size是否可变等信息。

ble\_parser.py：解析event模块，主要功能为解析event事件，如adv事件，connect事件等。

ble\_uart.py：串口操作模块，如rx/tx功能

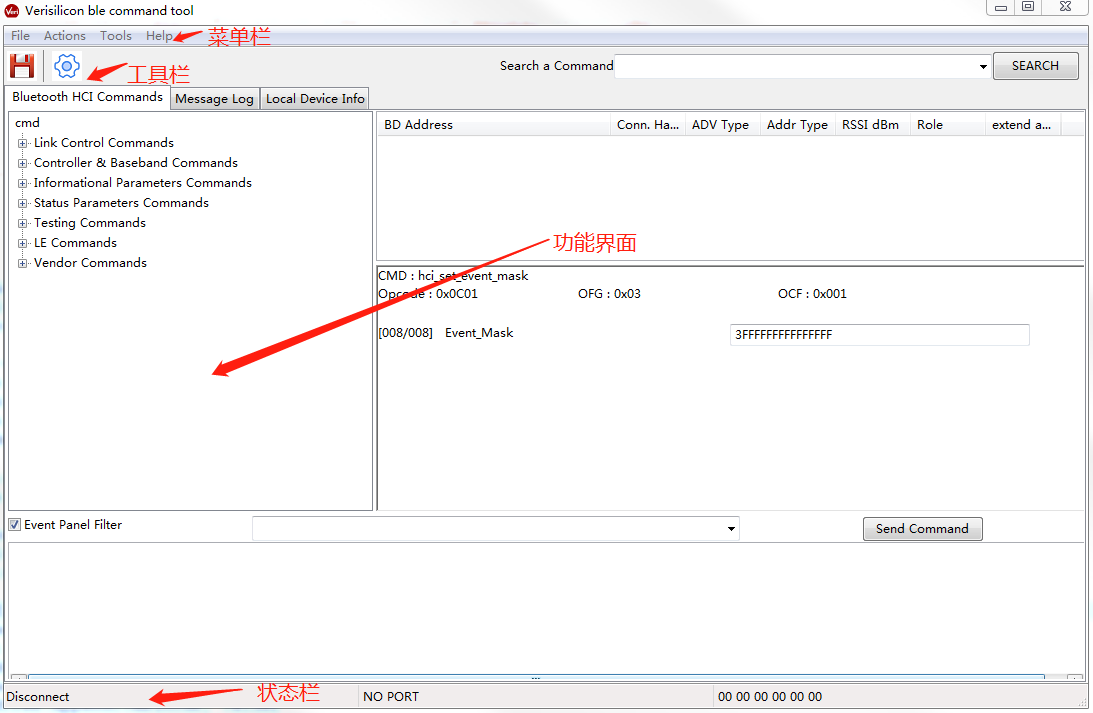
**ble\_widgets目录**：主要为GUI模块相关的操作，里面的子目录分别对应相应的页面界面。

**configs目录**：主要为cmd默认参数文件，LE参数的subevent的json文件以及cmd的data文件。

**res目录**：为一下资源文件，如图标等文件

# GUI模块介绍

整个界面的布局分为4个部分，分别为菜单栏，工具栏主页面和状态栏。主界面布局位于文件ble\_main\_window.py中，界面如下图所示。



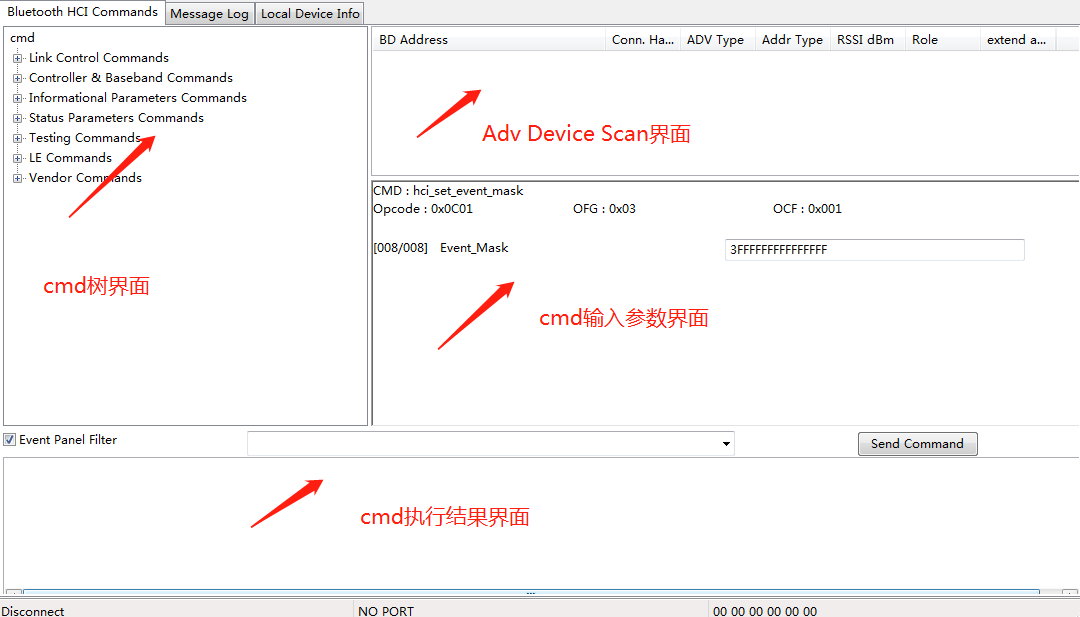
Tool软件的功能页面位置为主页面，主页面采用wx模块的Notebook对象实现，目前的版本只有3个主页面，在适当的时候可以通过增加对象的个数新增功能页面，本次设计主要包含main page、message log page以及local device info page三个功能页面，接下来将分别对其进行讲述。

## Main page modules

主页面主要包含4个部分Cmd树选择界面、cmd输入参数界面、Adv device scan结果界面以及cmd执行界面界面，主界面位于文件:

ble\_widgets\ble\_mainPageWidgets\ble\_mainPage.py

其界面如下图所示：



Cmd树界面: ble\_widgets\ble\_mainPageWidgets\ble\_cmdPage.py

采用模块wx.TreeCtrl构建cmd树

Cmd输入参数界面：ble\_widgets\ble\_mainPageWidgets\ble\_input.py

采用模块wx.TextCtrl构建输入界面，对输入的参数个数扩展无限制。

Adv device scan界面：ble\_widgets\ble\_mainPageWidgets\ ble\_advDevScan.py

采用模块wx.ListCtrl构建adv device scan界面，对于扫描到的adv或者extend adv设备进行显示，并对已经连接成功的设备进行必要的信息输出显示。

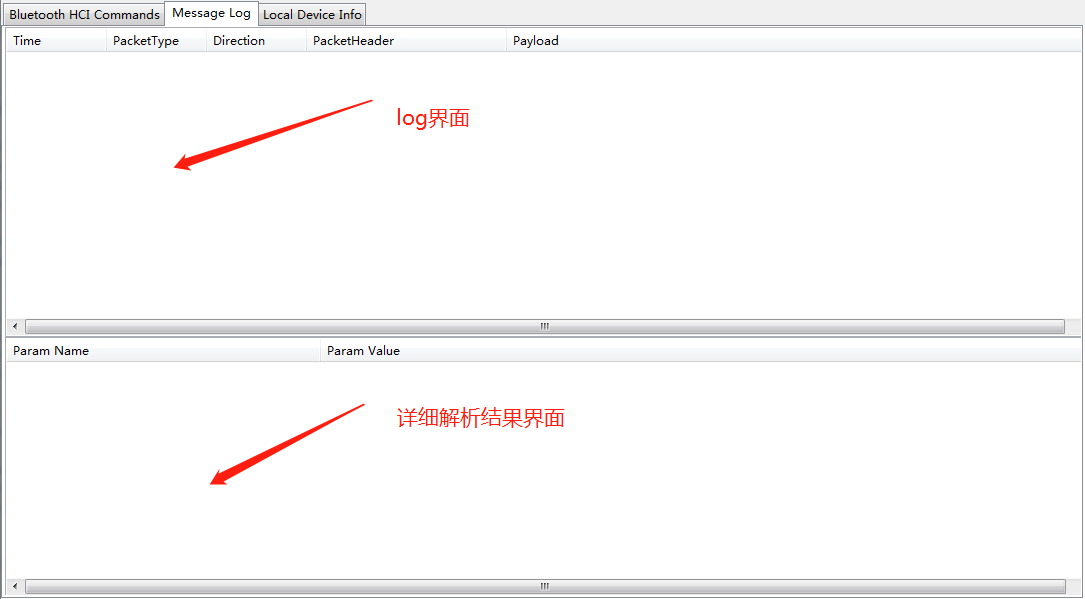
Cmd执行结果界面: ble\_widgets\ble\_mainPageWidgets\ble\_displayCmdStatus.py

采用类Ble\_cmdReturnInfoClass模块构建，其包含3个部分，一个history命令选择，一个send button以及cmd执行结果TreeCtrl模块。

程序在启动过程中，程序会将所有需要的cmds的输入界面都创建好，让其只显示第一个界面在cmd输入线束界面位置，其他的cmds的输入界面为隐藏状态，以后通过点击cmd树选择相应的命令都都是采用这样的原理，隐藏之前的cmd，显示被选中的cmd界面使其显示出来的模式来与用户进行交互。在cmd执行结果界面中点击“Send Command”按钮后，程序会根据当前显示的cmd输入参数界面的index去获取相应的命令的HCI格式的数据通过串口发送出去。

## Message log page modules

Message log主要有两个部门，一个的event包的log界面，另一位为被选中的event解析后的详细信息界面，如下所示：



当解析线程收到一个完整的包后，根据hci的header和payload进行分离，并将信息填充到log界面，当用户选中单个event时，action为解析该event事件并将详细结果显示到详细信息界面。

## Local device page modules

本界面主要为显示本地设备的一些部分信息的界面，如BD addr、蓝牙版本号之类的信息，程序位于文件” ble\_widgets\ble\_deviceInfoPageWidgets\ ble\_deviceinfoPage.py”，类名为:

class Ble\_localInfoListCtrlClass(wx.Frame):

采用模块wx.TreeCtrl对设备的信息采用树的模型进行管理，当新增加属性的时候，只需要调用起内部函数” addAttr”即可添加到GUI界面中。

# 软件框架

## 主框架

主要采用多线程设计。

### 线程框架

主要采用多线程进行设计，程序启动后将会有3个线程开始运行，主要是：

GUI线程：main thread主要负责GUI界面的操作，如text的内容的加载等等

Uart 线程：负责接收串口数据，打包成完整的HCI数据包后将其推入队列

Event解析线程：主要为从uart线程如队列的数据取出来解析，并将相应的结果写入到GUI模块中

如下图所示：



在流程启动recv/parser启程模块，会将parser thread最后需要操作的GUI句柄传递到parser线程以供子线程去操作GUI界面。

### 参数传递

本次设计参数传递主要有3种：

简单函数调用传递

Queue传递：如uart thread到event解析线程的传递

Main arg class传递：其为一个大的类，包含了改程序的所有的对象的类，其定义位于文件：

Includes\ ble\_main\_arg\_class.py，class名为class MAIN\_ARGS\_CLASS。在启动多线程的时候这个class的对象会被传入进去，其后子线程可以通过这个参数对其他的GUI对象进行操作。

## 线程模块

### GUI主线程模块

GUI模块主要负责GUI界面的处理，为便于I/O操作流畅，设计的初衷尽量让主线程做尽可能少的其他事情。在部分界面需要被多个线程操作的情况下设置锁机制，以避免读写脏数据以及程序崩溃。

### uart接收线程模块

该线程在程序启动的时候启动，在端口（串口）被初始化后将会对端口进行数据接收（以串口为例），收到数据后，会进行相应的解析，当收到一个完整的数据包后将其入队列。

### event解析线程模块

该线程主要目的为解析数据，传入的参数为Main arg class对象，也就是整个程序的主要参数对象，当有数据包从uart接收线程过来的时候对其进行解析，并将解析结果写入到GUI界面中，GUI界面的模块handle通过Main arg class传入到该程序中。

### acl数据发送线程模块

该线程只是在发送ACL data的时候才会被启动，主要目的为发送ACL data的数据并计算相应的发送速率并对其进行显示，通过Tools->Tx Data Transfer->Start Transfer启动。传输完成或者被用户cancel/close界面的时候结束该线程。