## DL\_02串口配置：

配置是有多步的，进入配置是上电的时候按着按键，配置内容的切换就是长按按键，配置的内容有，波特率，信道，主从机。波特率对应的led状态在背面有，信道要求一致，主从机根据实际的来就好。

点对点是双方都可以同时发送接收的，按着说明来的话，但是要配置成一个A，一个B。只有一个A和一个B可以通信，不能一个A多个B，一个A和一个A。

点对点模式和广播模式两种。

## 代码

串口是本质，无线只不过是用了不同的罢了，但是信息的发送和接收都是通过串口来做。

串口的发送简单，但是接收呢？之前一直都是发送方，现在还要控制接收呢？不仅仅是一方发送一方接收，而且要双方可以都发送和接收，最后如果可以还要双方可以同时发送和接收。

我需要自己整理协议吗？

之前章程写的那个而已直接用，我要去看看那是怎么做的

## 知识碎片

所谓缓冲区，在自己写通信协议的时候其实都是自己定的。根据需求来，问题不大。

拆分成字节来传输，接收到之后就要考虑重组，现在不需要考虑数丢包，和乱序，真是化解了很多。所以只要编号，然后每隔多少之后就提取一次就好了.

C++，联合union。“联合”是一种特殊的类，也是一种构造类型的[数据结构](http://lib.csdn.net/base/datastructure)。在一个“联合”内可以定义多种不同的数据类型， 一个被说明为该“联合”类型的变量中，允许装入该“联合”所定义的任何一种数据。这看起来和数据结构很一致。但是不同就在于联合内的变量是共享内存的，一个联合占的总内存是联合里面内存占用最多的变量的所占的内存，而结构是每个变量都有自己的内存，一个结构的内存是总内存的和。所以联合虽然每个变量都可以被赋值，但是每次只能有一个被赋值，而且，新的赋值会冲掉旧的赋值，那么数组呢？不会还有交换的作用吧？两个同样内存大小的变量，对一个做处理，然后调用的时候调用另外一个，也是访问了这个内存，然后就调出去了，天哪，感觉和指针类似的效果，而且更加偷天换日。

联合体里，共存数组和其他类型时，数组存放数据的方式是首地址存低位，逐渐上去，所以幅值的时候赋给首地址就是放在低位，赋给后面的地址就是放在高位。取数据也是这样。所以双方就可以约定是先传低位还是先传高位。

Sizeof计算的数据类型的字节数。

数组的定义必然要指明大小，除非是用指针定义一个头指针。但是有时候需要数组的大小岁变量的变化而变化呢？这时候初始化就有用了，把变量先初始化，那么就具体了，后面还可以继续修改。而这也是初始化的必要性吧。

C++类的思想的时候可能有些地方看起来有些无厘头甚至无理取闹，但是却还是不得不做，因为这样才能叫c++呀，独立，多态，继承。一个就是明明继承类用到了父类的变量，但是却不把变量公有，而是用函数去获取或者是保护状态，就是为了能够维持彼此的独立性。还有就是很多时候类中使用另一个类，不是继承而是调用，就是维持累的干净，简单，只提供必要的接口，还有的时候，明明可以写在一个类里面，但是却分成多个类，再组合，也是为了类的独立性，可继承性，全部写在一起，类的精髓就开始丢了，所以学会合适的划分工程要求也很重要。还有就是有时候只是把几个接口放到一起，就又成了一个函数，甚至只是再封装了一个接口。不要说怎么这么简单？封装好了之后就是这么简单，直接调用，如果有人封装了一大堆库函数，你调用起来也是这么简单。所以用轮子就是这么简单。

代码逻辑好像都是从结果开始，先排除了特殊情况和异常情况，才开始其他。FPGA是，程序逻辑也是。

指针到数组数据，之前是+1，就直接是下一个数据了。现在这个咋更直接，传入指针直接当数组用？数组的访问原理，本来就是从首地址加上偏移量来吗？传入的指针竟然可以直接当数组用？理论上不是不知道指针指向的是什么吗？就只是一个点，蕴含着下面的地址而已？错了，那是链表，但是这个不是！！！，我把链表当做指针了。或者把所有的指针都认作了链表。传入首地址访问，是建立在所有的数据都已经确定的基础上，那么自然而已首地址移位访问，而且，就算是指针++的形式，不也是这样吗？

所以，学习一个新的程序的东西，从存储结构，逻辑结构上来考虑，应该是最好最明了的吧。

通信一定要合理利用已有的的东西，用最简单，资源消耗最小的方式到达目的。