头文件扫描功能

驱动组 王彬 2016.06.07

|  |  |  |  |
| --- | --- | --- | --- |
| 版本 | 时间 | 作者 | 说明 |
| V1.0 | 2016.6.07 | 王彬 | 第一版本draft |
|  |  |  |  |

# 1：概述

## 1.1：目的

通过扫描dsp的头文件获取所有变量的地址，并可以将程序中的字符串信息，变量的字节数，变量的模块信息等提取出来作为调试信息使用。这个过程将自动化完成。

## 1.2：目录文件扫描

采用ffsco类进行windows下的目录中的文件扫描。该类可以找到目录下的文件和子目录下的文件。也可以找到目录和子目录等。这里只需要获取头文件即可。对于一些不需要扫描的头文件，可以从文件列表中删除掉。提高效率

## 1.3：头文件的解析

头文件的解析的作用是从扫描到的头文件中找到需要的结构体，枚举类型，联合体类型，宏定义等。

## 1.4：树结构的构建

在得到结构体等信息后，需要将这些信息建立起一个和底层程序一样的结构。该结构是一个结构体，联合体，枚举等的嵌套结构。并且其中有些数组，和结构体数组需要特殊处理。

采用自顶向下的搜索方式，通过递归调用的方式找子元素。

## 1.5：代码自动生成获取地址的文件

当建立好树结构后，目的是实现高级功能，因此需要获取所有变量的地址偏移。方式是通过代码自动生成一个GetAddress.c的文件。因为扫描出的树结构中包含了所有底层代码中的变量字符串信息，所以可以容易生成这个文件。

这个文件是要放到IAR工程中进行编译的。最终将所有变量的地址打印到一个Addr.txt的文本文件中。如果是多轴的情况，那么只需要打印第一个轴的所有变量偏移和其它轴的主框架结构偏移。那么其它轴的所有变量偏移就可以根据第一个轴偏移和该轴的主框架偏移计算得到。

其次，因为IAR工程中有多轴的时候，就会有gSevDrv.sev\_chd[2]这样的主框架结构。如果直接使用gSevDrv.sev\_chd.act\_on的结构编译器会报错，因为编译器会将sev\_chd这个数组名当成是指针，所以需要将之前获取的树结构中的某些“.”操作更改为指针操作“->”，例如gSevDrv.sev\_chd->act\_on。这样生成的文件在IAR中编译才能没有问题。

该文件中并没有考虑bit位的操作，所以并没有打印bit位的地址。因此读取该文件时要注意和之前的list对应。

## 1.6：在IAR中编译运行获取地址

1.5中生成的文件将在IAR的工程中进行编译并运行，生成地址文件。该IAR工程是一个独立的工程，区别于真正的四轴代码工程，该工程的唯一作用是获取地址文件。该工程放在实际Servo工程下面的GetAddr文件夹下。这个工程不能被拷贝到其它目录使用，因为其中很多文件使用相对路径包含Servo工程中的很多文件。

其次，该工程目录下包含其自己的main函数和GetAddress.c.负责打印需要的变量地址。

实际使用时需要考虑exe文件和GetAddress.c和Addr.txt所放的位置，根据实际工程的文件组织再进行调整。

## 1.7：更新地址

更新地址分为更新变量地址和参数地址，其中变量地址就是Addr.txt中的地址，参数地址是铁电中分配的地址。铁电存储器中分配的地址做了预留，可以兼容之前的参数地址。非陪铁电地址的时候需要得到参数的子模块名称。然后根据模块进行分配。注意多个轴的时候需要设置多个轴的地址。

## 1.8：xml文件生成

Xml文件的操作采用c++语言的xml操作库tinyxml。需要增加一些文件到工程中。主要就是对XML进行读写操作

## 1.9：内存操作

在dll中将建立特性表的内存映射，内存以多叉树的链表形式存在，可以进行查找，读取，修改等操作。

# 2：操作流程

## 2.1：总流程图



## 2.2：头文件解析子流程图



## 2.3：建立程序结构树list子流程图

