

1 ECONWIN 来历

ECONWIN 产生于很多的思考和工作中长期经验，我以前是做通信行业 C++ 软件开发，07 年初转向设备行业，做 linux 下面的嵌入式设备，08 年正式进入工控行业，学习了很多的平台（DOS，windows，linux，单片机，各类 PLC），获益很多，但是学习中的艰辛和使用中项目期限的压力，各位经历过的可想而知。

开发人员为了做一个产品，经常进行反复的学习，反复的对比，有时系统有问题，因为是闭源，根本无法修改，感觉是浪费生命在这些东西上面。

我一直在想，能否一劳永逸的使用一个平台解决这些问题呢？经过反复的思考，都难以获得一个很好的解决方案，2013 年看到贝加莱的 powerlink 总线方案，我眼前一亮，这不就是我一直追求的东西么！

我决定采用通用平台上的简易以太网协议来实现一个通用控制器，因为控制器使用以太网扩展连接外部的模块，还有控制器成本也很便宜，所以命名为 ECON，有以太网（Ethernet）和便宜（Easy）控制器的含义。

2 系统设计哲学 KISS

经过长年的工控系统的开发，我特别推崇 UNIX 的设计哲学-KISS，对于 ECONWIN 的设计架构，我尝试将 KISS 原则贯穿于整个体系之中。

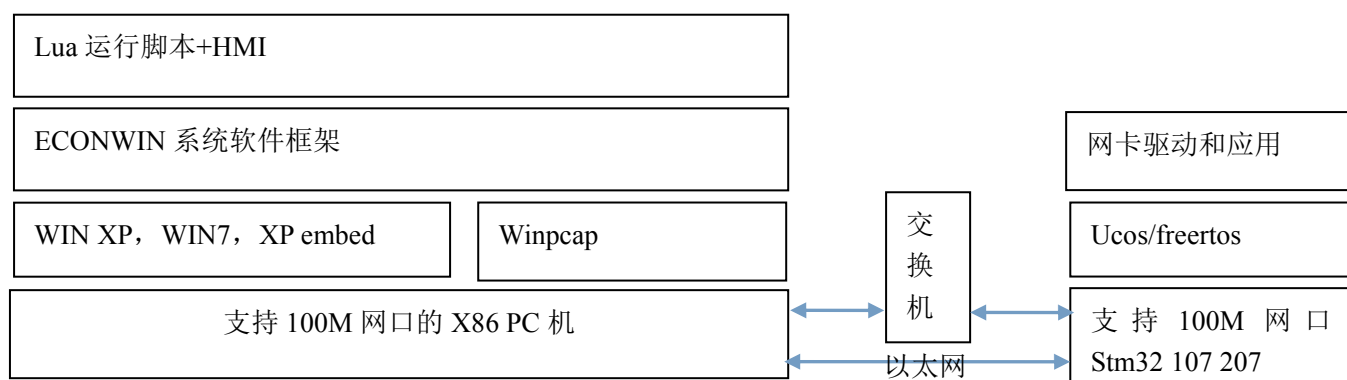
通用平台主机我选择 PC 机或者工控机，这个平台是目前性价比最好的硬件平台，window 或者 linux 都是非常非常成熟的运行体系，虽然普通系统实时性不够好，但是加上实时补丁或者设计得够巧妙后，系统的实时性还是可以的（linux 可以得到 1ms 的硬实时，windows 可以得到 1ms 的软实时），又加上 PC 平台的容易获得（现在人基本人手一台 PC 机或者笔记本），开发者可以在 10 分钟内搭建系统并开始设计验证。

通用平台上的开发环境，我采用 VC2003+Qt 作为系统开发 IDE，lua 脚本为系统的运行脚本，这些都是目前软件工业界能找得到最好的最成熟的东西，VC 开发环境从易用性还有性能上都是最好的，Qt 是开源的体系框架，作为 ECONWIN 的设计基础，刚好适合。Lua 脚本和 c/c++ 的完美契合、lua 脚本的简易语法、lua 运行高性能等都是我选择他的理由。

系统外围扩展总线采用以太网，由于网络的普遍应用，1000M 网口目前都是 pc 机的标配，100M 交换机的廉价和高性能，使基于太网的系统扩展，变得廉价而且易得。

外围模块采用 stm32 107、stm32 207 等 arm 芯片，stm32 芯片为高速的 risc 芯片，体系成熟稳定，使用者众多，107，207 等型号自身带了以太网 mac 器件，编程简单、性能优良，开发者可以快速的定制自己的外围模块。

3 ECONWIN 体系架构



4 快速体验 ECONWIN

在下面的项目托管网页中

<https://github.com/huzhiwen28/econwin/>

 **Download ZIP**

下载系统包，解压

1 安装以太网驱动

在文件夹 econwin/econwinpackage/网口驱动程序/ 中运行 WinPcap_4_1_2.exe，此是以太网驱动。

2 安装 lua IDE

在文件夹 econwin/econwinpackage/lua IDE 中会找到安装文件和 lua 学习教材，运行安装文件会安装 lua IDE。

3 系统运行

在文件夹 econwin/econwinpackage/econwin/中运行 econwin.exe 文件。

4 开始随意探索系统的功能吧

econwinpackage\econwin\hmi 文件夹中保存了 HMI 项目，其实是 lua 方式配置文件

econwinpackage\econwin\luasystem 文件夹中保存了系统的基础服务文件

econwinpackage\econwin\plugins 文件夹中保存了 qt 库

Fileregister.lua 文件是用户自己配置系统包含哪些用户定义的 lua 脚本的文件

Globalvar.lua 是用户自定义 lua 空间的全局变量的文件，这些变量的定义是方便在变量监控中显示

Para.lua 是系统使用的配置文件，用户最好不要编辑

usertask1.lua 是用户自己定义的 lua 脚本任务

5 ECONWIN 界面

5.1 主界面

运行程序后，主界面如下

主界面主要为 HMI 编辑功能和 HMI 界面运行功能，系统自己会记忆 HMI 项目名称和运行状态。



5.2 系统界面

点击 **系统** 菜单，弹出系统界面

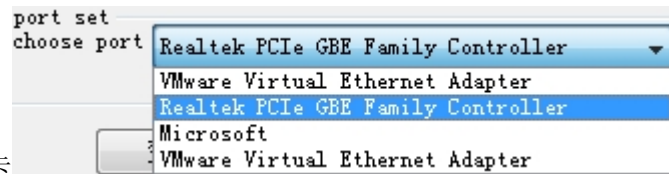


系统界面有如下功能

- 1) 启动、停止系统运行

使用   按钮

2) 选择系统总线使用的网卡



在系统停止运行状态下面选择网卡

系统目前只支持有线网卡!

3) 进入变量监控界面

点击  按钮进入

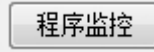
4) 进入任务监控界面

点击  按钮进入

5) 进入模块监控界面


点击  按钮进入

6) 进入程序监控界面

点击  按钮进入

5.3 变量监控界面

变量监控界面，是对系统中定义的全局变量监控的界面，目前支持简单的变量（lua 脚本和 c++ 中的全局变量）监控和设置值。

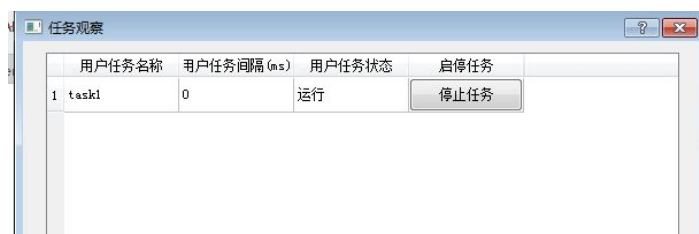


	变量名	变量类型	变量值
1	错误通讯次数	unsigned int	69415
2	bint	unsigned int	200
3	cint	int	0
4	dushort	unsigned short	0
5	eshort	short	0
6	full	unsigned long long	0
7	gll	long long	0
8	hul	unsigned long	0
9	il	long	0
10	jf	float	2.5
11	kd	double	-0.13
12	a	double	200
13	b	double	0
14	zhishi deng1	double	0
15	zhishi deng2	double	0
16	zhishi deng3	double	0

返回

5.4 任务监控界面

任务监控界面，是对系统中定义的任务监控（lua 任务和 C++任务）的功能界面，可以起到和停止任务



	用户任务名称	用户任务间隔(ms)	用户任务状态	启停任务
1	task1	0	运行	<div>停止任务</div>

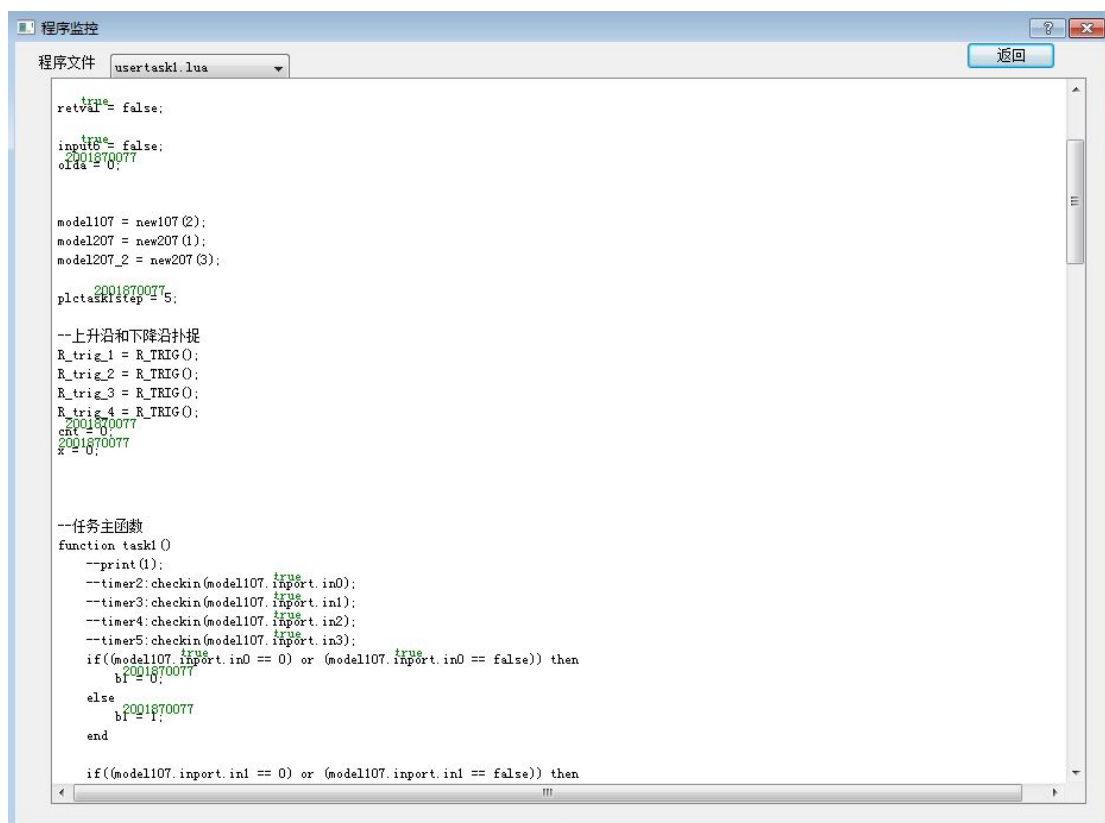
5.5 模块监控界面

模块监控界面，可以在其中系统中定义的模块的端口状态，并设置输出口的值。



5.6 程序监控界面

对 lua 脚本程序进行监控的界面，可以查看 lua 脚本中各个变量的实时刷新值。



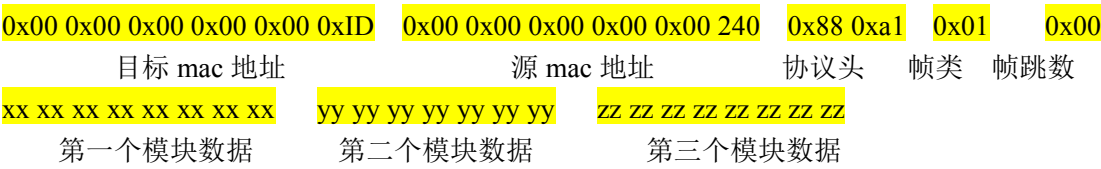
6 ECONWIN 通信协议

ECONWIN 通信协议目前比较简单，分为 3 类，PDO 帧（01），SDO 帧（02），广播帧（03）
PDO 帧和 SDO 帧由 PC 发出后，顺序经过外部模块进行数据交换(ethercat 方式)，最后一个节点返回到 PC 机，广播帧由 PC 机发出，外部模块读自己的数据，无需应答。

PDO 是 PC 和外围模块自动间隔时间刷新的帧，主要做内存刷新。
SDO 是用户触发的或者模块启动时设置的通信帧，一般用来设定外部模块参数。
广播帧是用户触发的通信帧，用来做模块的同步。

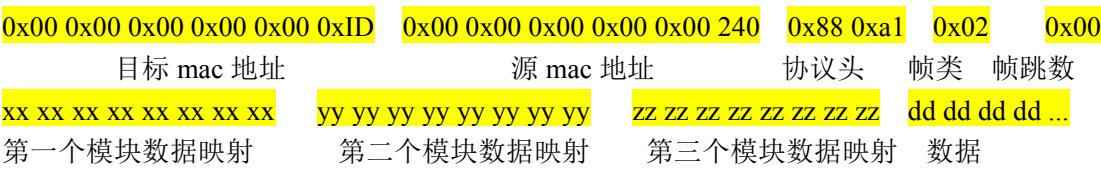
注意：目前系统只实现了 SDO

PDO 帧格式



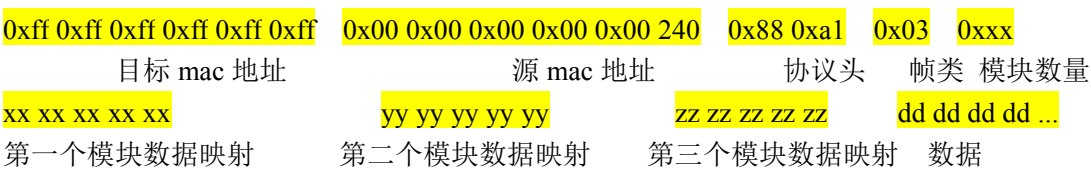
PDO 帧中的每个模块数据，会根据模块中的映射参数进行解析，而映射参数是通过启动的时候，PC 使用 SDO 帧设置到外部模块上的。

SDO 帧格式



数据映射为 8 个字节，格式如下
WADDRL WADDRH WLen WDeviceADDR RADDRL RADDRH RLen
RDeviceADDR
WADDRL：写数据在帧中地址的低字节 WADDRH：写数据在帧中地址的高字节 WLen：写数据的长度 WDeviceADDR：写设备中的内存地址 RADDRL：读数据在帧中地址的低字节 RADDRH：读数据在帧中地址的高字节 RLen：读数据的长度 RDeviceADDR：读设备中的内存地址

广播帧



模块数据映射为 5 个字节，格式如下

ID WADDRL WADDRH WLen WDeviceADDR

ID: 为外部模块 id WADDRL: 写数据在帧中地址的低字节 WADDRH: 写数据在帧中地址的高字节 WLen: 写数据的长度 WDeviceADDR: 写设备中的内存地址