<< 可视状态机用户使用手册 >>

1 可视状态机介绍

Visual State Machine(简称VSM)是一种有限状态机,它的状态和从状态到状态的转换可以很容易地用GUI 软件用可视对象来表示。例如PLC/RTU产品在其用户级软件功能定制中采用VSM进行编程。用户使用VSM 模型创 建他们的应用程序,并将VSM 程序下载到例如PLC/RTU 产品中。程序下载后,二进制代码存储在闪存中,并在 每次上电时自动运行,直到下次被其他程序覆盖。这里的Visual State Machine特指Raynix Ltd.提供的PC软件。安装此软件后,用户可以为egPLC/RTU系列产品创建、编辑、编译、链接、下载定制程序。

例如PLC/RTU系列产品用系统空间和用户空间来配置它们的CPU和内存资源。运行 VSM 代码的 32 位微处理器 嵌入用户空间,用于执行用户程序。该微处理器支持 32 位浮点数据表示和浮点数据操作。状态转换指令本质上 便于支持从状态到状态的 VSM 程序流切换。

一个状态由一段代码执行组成,紧随其后的CPU空闲状态是代码执行序列。在当前状态结束时执行从当前状态到下一个状态的转换。 VSM软件用于创建和编辑每个状态的C代码,集成软件功能模块包括检查程序语法,编译代码,汇编代码,下载二进制代码。

2 安装 VSM 软件

VSM Windows PC 软件随附一个安装副本"Setup_VSM.msi"。可以从 Windows 双击该程序以启动并执行安装 过程。但是,如果主机 PC 机器中没有dotNet组件,用户可以开始运行"setup.exe"程序,该程序将通过从 dotNet 网站下载必要的组件来指导安装过程。 VSM 软件的安装既快速又简单。安装完成后,PC桌面会出现一 个图标,表示VSM的快捷方式。注意:在安装的路径名中不能包含空格或中文和一些特殊字符,目录路径名称沿 用传统unix的习惯。



在安装目录下,软件自带一个"blink"示例,位于"example"子目录下。此示例仅包含一个文件"blink.vsm"。 该文件可由Visual State Machine软件加载,用户可以对其进行编辑修改,编译汇编并下载到egPLC/RTU系列 产品中—演示自定义输入/输出闭环控制过程和自定义其他功能。

3 浏览示例

启动可视状态机并从文件菜单加载现有项目,然后将"blick.vsm"加载到 GUI 软件中。

1





我们可以从图形状态选项卡中看到"blink.vsm"有 2 个状态(S0 和 S1)和两个切换(从 S0 到 S1,反之从 S1 到 S0)。

我们还可以看到 S0 状态是六边形物体,而不是圆形物体 S1。六边形状态是初始状态,这里是程序的入口。每个程序只有一个初始状态,即使程序中只有一个状态!

State_0 选项卡显示在 S0 中执行的代码, State_1 选项卡显示在 S1 中执行的代码。这里的代码是简单的 OI 函数,用于打开和关闭 egPLC5888 设备的 8-YOUT 固态继电器开关。并且在每个状态程序将保持空闲等待 1 秒! 无条件状态转换会导致输出灯闪烁。

此应用程序中没有全局变量或过程。

程序编写好后,我们可以从"工具"菜单检查语法、编译和汇编代码,最终将程序下载到设备的闪存中。

"语法检查"会自动扫描整个 VSM 状态和代码,并在状态栏上报告结果。

"编译"将编译 VSM 程序并在状态栏上报告结果。

"汇编"将程序组装和链接并在状态栏上报告结果。

当上述所有过程都通过"下载"菜单项点击将二进制代码发送到设备的闪存中。下载程序后,设备将在启动设备时运行用户程序。固化在闪存中的程序下电之后不会丢失。

用户现在可以看到 LED 闪烁!

2



4 编写程序

从"文件"开始"创建新项目",然后用户可以添加新状态、添加新转换和编辑代码。代码为C语言,VSM软件包提供了一套预定义的系统库接口,不需要"#include"关键字!预定义系统库定义如下。

```
/*
* 从一段代码过渡到另一段代码
* addr 是代码的标签,或函数/子例程的名称。
* 这用于 VSM 从状态切换到状态。这执行一个转到功能
* 无回报
*/
void switch_to (long addr);
/*
* 延迟时间以毫秒为单位,1个tick需要10ms,所以实际延迟时间是round
* 以毫秒为单位下降到最接近的 10 次整数。
*/
void br_delay(int ms);
/*
* 基本通过给定地址从IO口输入长数据
*/
long br_inport (long addr) ;
/*
* 基本按给定地址输出长数据到IO口
*/
void br outport (long addr, long dat);
/*
* 设置RS485主端口波特率,默认9600
*/
void br_set_baudrate(long bd);
/*
*从 RS485 主端口接收或发送字符
* 对于receive,它是一个同步阻塞函数,等待字符直到得到
*/
char br_recv_char(void);
void br_send_char(char c);
/*
* 向RS485主端口发送一个块数据,字符串长度和块数据长度
* 不超过32字节,否则配置长消息短
* 按用户。用户可根据波特率计算发送时间表
*/
void br_send_string(char *s);
```



```
void br_send_block(char *s, int num);
/*
*从 RS485 主端口读取一行字符并存储到字符串中,直到出现 '\r' 或 '\n'
* 满足。
* 一个以空字符结尾的字符串存储在 st[]
*/
char *br_gets(char *st);
/*
* 将格式化的数字打印到 RS485 主端口
* 支持的格式是
* "%i" 表示 int 和十六进制输出
* "%1" 表示长并以十六进制输出
* "%c"代表字符
* "%s" 表示字符串
*/
void br_printf(char *fmt, ...);
/*
* 从需要固定长度的人类可读字符串文字中读取十六进制数字
*/
long br_get_hex(char *s, int n);
```

添加新状态

在可视状态图形选项卡上,用户可以在背景区域上单击鼠标右键,弹出一个上下文菜单,即"添加新状态"命令。 将新状态添加到项目后,可以通过对象的上下文菜单修改其属性。它显示的内容如下。

1	删除这个状态
	路径起始点
	路径终结点
	设置初始状态
	编辑状态名称

从上下文菜单中,可以删除该状态对象,可以更改该状态对象的名称,还可以将该状态标记为程序的初始状态。

因此,用户可以创建从状态到状态的转换。转换由源状态和目标状态定义。因此转换是由状态对创建的,甚至更 多的源状态和目标状态可以是相同的状态!

在状态标签页中自动生成状态转换代码,用户可以细化转换条件以正确控制他们的程序!

在任何情况下,在每个状态代码执行结束时都必须遇到转换状态 switch_to() 函数。

语法检查和编译后,项目可以保存在磁盘上,以备将来使用和交流。



5 构建和下载程序

当用户开始构建程序时,软件会在内部将所有状态都安装在一个单独的C 文件中,然后对 C 文件进行逐步的一系列处理。在语法检查期间,系统将报告显示每条错误消息(如果有)。当没有错误时,状态栏显示"语法检查通过"信息。此消息表明可以沿着整个处理链执行下一个过程。

程序构建过程步骤将包括语法检查、编译、组装和下载。在下载例如PLC/RTU 设备之前,应将其连接到USB 端口或通过TCP/IP 网络!

6 用于 USB 和网络设备的 VSM

一旦 USB 数据线从主机 PC 连接到例如 PLC/RTU的USB 端口设备就逻辑连接到 PC 主机。因此,在这种情况 下,默认执行程序下载以访问 USB 端口。下载 TCP/IP 端口设备时,应在对话框设置框中预先提供目标 IP 地址。此 IP 地址不用于目标设备设置更新,而是为 Visual State Machine 软件提供设备的目标地址。当 网络端口设备连接到网络并且此 IP 地址在连接字段中指定且不存在 USB 端口(例如 PLC/RTU 设备)时,下载过程将转发到 TCP/IP 网络。

总而言之,当 USB 设备连接时,下载方向转到 USB 设备,否则,下载方向通过指定的IP 地址搜索网络。

7 语言规范

可视状态机使用 ANSI/C 语法的一个子集,它不支持枚举数据结构。它支持 4 种基本的有符号数据类型,分别 是 char (8 位有符号整数)、int (16 位有符号整数)、long (32 位有符号整数)和 float (32 位 IEEE-754 浮点数) . VSM支持组合数据结构s,比如结构、联合、数组等。

表达式中变量之间的运算应基于相同的数据类型进行,VSM 不会自动进行数据类型转换。任何数据类型转换都应 使用显式约定。

VSM 支持宏定义和包含预处理的头文件,但是目录和文件名约定应符合Unix命名约定而不是 Windows 命名约定。文件名或路径名中不允许有空格或"-"。 VSM 在源代码级别的严格要求有助于去除任何含糊不清的语言代码并生成和生成稳定且明确的二进制文件!

5