

顺序控制图源代码形成方案设计与实现

王国栋,陈宏君,刘克金

(南京南瑞继保电气有限公司,江苏 南京 211102)

摘要:为实现顺序控制功能和状态转换功能图形程序转换成可执行的 C 语言程序,文中设计了顺序控制图的图形化编程符号、定义符号的 C 代码范式并实现 C 代码生成方法。首先,将顺序控制图形化页面分为任务页面、执行步页面和功能图子页面,可清晰地组织可视化程序;其次,设计多种符号块实现程序步顺序执行和转移控制;最后,根据图形化编程符号的拓扑顺序和控制逻辑生成对应的 C 语言代码。进一步提出基于双 switch-case 顺序控制图的任务函数代码生成模式,在第一个 switch 结构中更新状态变量的值,在第二个 switch 结构中根据新的状态变量值运行相应的控制函数,可在同一个运行周期内完成状态切换和动作执行。提高了嵌入式顺序功能程序的运行效率,适用于高实时嵌入式系统的顺序控制。

关键词:顺序控制;双 switch-case;代码生成;状态切换

中图分类号:TP31

文献标识码:A

文章编号:1673-629X(2016)03-0149-05

doi:10.3969/j.issn.1673-629X.2016.03.035

Design and Implementation of Source Code Generation for SFC Graphical Program

WANG Guo-dong, CHEN Hong-jun, LIU Ke-jin

(NR Electric Co., Ltd., Nanjing 211102, China)

Abstract: To generate source code for SFC graphical program, it presents a source code generation scheme which designs graphical programming symbols for Sequential Function Chart (SFC) and defines C code normal form for the symbols and generates real C code finally in this paper. Firstly, the scheme divides sequential function chart pages into three types including task page, execution step page and functional graph page, and manages all pages hierarchically. Then it designs various graphical symbols to achieve sequential execution and transfer control of SFC graph. Finally, it generates C code according to the topological structure and control logic of these graphical symbols. This scheme further promotes source code generation approach for dual switch-case SFC graph page by updating the value of status variables in the first switch structure and executing corresponding control functions in the second switch structure according to the updated status variables, thus completing status switching and function execution in exactly one operation cycle. It increases execution efficiency of embedded sequential function programs, which is suitable for sequence control of high real-time embedded systems.

Key words: SFC; dual switch-case; code generating; state switching

0 引言

顺序控制图是指用于设计实现某个顺序控制功能或者状态转换功能的图形程序,它将整个控制流程分割为一系列的控制步,并描述出生效的执行顺序和控制条件。图形化的顺序控制图已经在工业控制等领域中得到了应用^[1-5]。文献[1]介绍了基于 SIMENS 公司生产的 S7-200 系列机型,利用指令和内部寄存器实现顺序编程的方法。文献[2]介绍了 PLC 的 SFC 在送料小车的应用实例。文献[3]提出了数组式编程方法,通过指针寻址功能对数组进行逻辑运算实现控

制功能。文献[4]通过在上位机运用全组态配置和后台程序搭配的设计方法,实现了混合流程的顺序控制。文献[5]基于 IEC 61850 标准对顺序控制的过程状态和操作配置进行建模,设计了顺序控制功能标准化的状态和控制模型,适用于智能变电站顺序控制。文献[6-9]介绍了控制保护系统的可视化编程软件设计理念,采用图形化符号搭建装置功能。

上述文献并未阐述图形化顺序控制图到嵌入式装置运行所需的源代码生成技术。文中设计了一套基于 C 语言流程控制结构的图形化符号块,并提出了双

收稿日期:2015-06-08

修回日期:2015-09-14

网络出版时间:2016-02-18

基金项目:国家“863”高技术发展计划项目(2015AA050101)

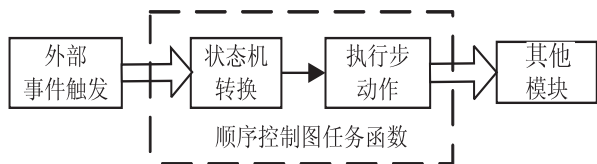
作者简介:王国栋(1988-),男,硕士,工程师,研究方向为可视化编程软件。

网络出版地址: <http://www.cnki.net/kcms/detail/61.1450.TP.20160218.1630.030.html>

switch-case 架构的顺序控制图代码生成方案,可在一个周期内完成状态切换和动作执行,适用于实时控制系统。

1 顺序控制图原理和相关概念

顺序控制图的原理如图 1 所示。相关的基本概念有事件、状态、转换、动作,外部事件触发状态转换,之后执行相关动作,并且将输出或指令传递给其他模块^[10-14]。



(1) 事件 (Event): 在时间和空间上占有一定的位置,事件通常会引起状态的变迁,促使状态机从一种状态切换到另一种状态,通常指其他输出变量值发生变化。

(2) 状态 (State): 是对象在其生命周期中的一种状况,处于某个特定状态中的对象会满足相关条件,执行某些动作或者等待某些事件。

(3) 转换 (Transition): 是两种状态之间的一种关系,对象将在状态 1 中执行一定的动作,并将在满足某个特定条件下进入状态 2。

(4) 动作 (Action): 指状态机可以执行一些原子操作,在顺序控制图中是运行一个执行步函数。

2 顺序控制图元件

文中,顺序控制图对应的代码用元件进行封装。元件是基于面向对象的设计思想,它包括一组数据定义和对数据进行处理的数据函数,处理完后输出传递给其他元件作为输入源^[6]。其源代码包括 h/c 文件,在 h 文件中定义数据结构体,包括输入变量、输出变量和参数等对外接口,在 c 文件中定义相关函数,包括构造函数、初始化函数、功能块函数、页面函数、任务函数等。

顺序控制图元件:用于设计实现某个顺序控制功能或者状态转换功能的元件,外部事件触发其状态的转换,并执行相应的动作。顺序控制元件的输入、输出和参数变量来自于顺序执行控制页面和执行步页面。顺序控制图元件的核心组成部分为:

- (1) 一个状态转换变量,根据状态变量的当前值运行相应执行步页面功能;
- (2) 状态转换控制的规则,通过设计顺序执行控制页面来实现。

3 顺序控制图页面设计

顺序控制图页面类型有顺序执行控制页面、执行步页面、功能图子页面。其页面功能设计如下:

(1) 顺序执行控制页面:用于设计顺序控制或状态转换的页面(可支持多页)。顺序执行控制页面通过执行步和转移块的前后连接关系实现状态转换的规则。

(2) 执行步页面:执行步页面代表执行步块所对应的功能,代码生成时将该页面的功能封装成一个函数,供顺序控制任务函数调用。执行步页面通常由各种功能块组成计算或逻辑功能,原则上不包含子元件块。

(3) 功能图子页面:在一个层次化的设计中用于代表一个子功能图页面的图块,将子功能图页面中的输入输出作为该图块的输入输出显示在执行步页面上。当执行步的页面功能比较复杂时,可以通过若干功能图子页面进行分解,然后在执行步页面连接子页面的输入输出,在代码生成时,将子页面对应的符号复制到执行步页面。

4 顺序控制符号块设计

用于顺序控制的图形符号有:起始符号块、执行步块、无条件转移块、IF 块、单 IF_ELSE 块、双 IF_ELSE 块、SWITCH_CASE 块。图 2 是各种符号的图形化形态,其中 SWITCH_CASE 块根据分支数目动态调整显示。

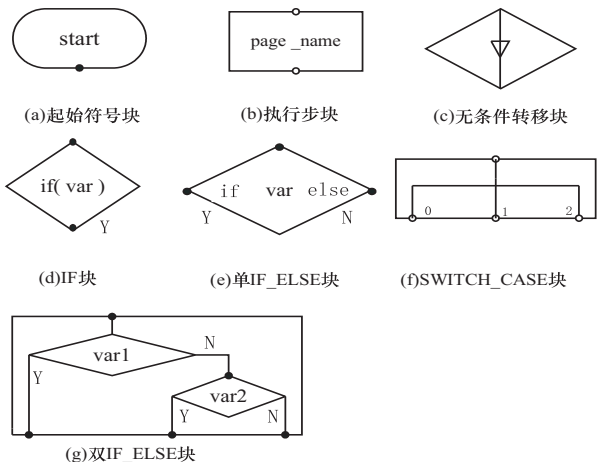


图 2 顺序控制符号块

各个符号功能设计如下:

(1) 起始符号块:在顺序执行控制页面中,起始符号块用于自动形成一个状态机变量,该变量作为元件中的成员变量,初始化为 0。

(2) 执行步块:代表一个执行步页面,对应一个状态,点击可以进入相应的功能图设计页面。执行步块表示运行该执行步页面的功能函数。元件的结构定义

为 SFCComp,则执行步块对应的函数形式为:

```
void runPageIndex(SFCComp * pcp)
```

(3)无条件转移块:表示根据上一状态,直接跳转到另外一个状态值。其代码原型为:

```
pcp->sta=new_stat;
```

(4)IF块:根据某个变量值来决定是否转移或转移到哪一个状态。判断变量可以是程序设计页面中的输入变量、输出变量、成员变量,为二值(0、1)类型。IF块需要指定判断变量名。其代码原型为:

```
if(pcp->var){pcp->state=stax;}
```

(5)单 IF_ELSE 块:根据判断变量的值决定执行哪条分支。单 IF_ELSE 对应的代码原型为:

```
if(pcp->var){pcp->state=stax;}
```

```
else{pcp->state=stay;}
```

(6)双 IF_ELSE 块:有三个输出点,需要指定两个条件判断变量,支持在 ELSE 分支中有子 IF_ELSE 条件。其代码原型为:

```
if(pcp->var1)
```

```
{pcp->state=stax;}
```

```
else{
```

```
if(pcp->var2)
```

```
{pcp->state=stay1;
```

```
else
```

```
{pcp->state=stay2;
```

```
}
```

(7)SWITCH_CASE 块:需要用户设定 case 的分支数,和每个分支的设置值,其图形需要根据分支数动态生成。对应的代码原型为:

```
switch(pcp->var)
```

```
{
```

```
case value_0:pcp->state=stax;break;
```

```
case value_1:pcp->state=stay;break;
```

```
default;break;
```

```
}
```

5 顺序控制图代码生成规则设计

5.1 页面规则设计

顺序执行控制页面的规则如下:

(1)顺序执行控制页面的功能封装成元件的一个任务函数(默认为任务等级1的函数),在每个任务周期里执行此任务函数,该函数决定状态跳转的规则;

(2)该页面允许存在的符号有起始符号块、执行步块、转移条件块、连接线、输入块、输出块和页面链接符号(指存在多页的情形);

(3)该页面的连接线不表示数据流的关系,只表示执行顺序和状态切换的规则;

(4)该页面符号的输出连接点只能连接一条连接线,表示只跳转到一个状态,输入连接点可以连接多条

连接线,表示可以由不同的其他状态转移到本状态;

(5)一个执行步块的后继一般为转移块(条件转移和无条件转移),若存在多个执行步块顺序串联在一起,则这些执行步块等同为一个状态;

(6)只能有一个起始符号块,至少有一个执行步块和条件转移块;

(7)转移块的后继必须是执行步块,IF 块和无条件转移块必须有后继,IF_ELSE、SWITCH_CASE 必须至少有一个分支为执行步块的后继;

(8)执行步块的后继一般为转移块,如果是转移块,转移条件需设定为一个状态值。如果存在多个转移块串联,则这些转移块的状态值相同,必须是一个整体,不能有分支。

执行步页面的规则如下:

(1)该页面可以包含输入块、输出块、参数块、常量块、功能图子块、页面链接符号、连接线、功能块等符号;

(2)该页面设计的功能相对独立、清晰,允许各功能图页面之间通过页面链接符号进行数据流交换(定义为元件中的私有变量);

(3)所有功能图页面功能块的任务等级都相同,功能块之间的连接线变量定义为函数中的局部变量。

功能图子页面的规则为:子页面内不再包含功能图子块符号,只支持一层嵌套关系。

5.2 双 switch-case 控制函数结构设计

顺序控制页面任务函数的代码,采用两个 switch 结构,第一个 switch 结构更新状态变量的值,第二个 switch 结构根据新的状态变量,运行相应的控制执行函数,控制执行函数运行在同一个周期。以图3的顺序控制页面为例,有3个状态,2个执行步页面,3个条件判断跳转符号。

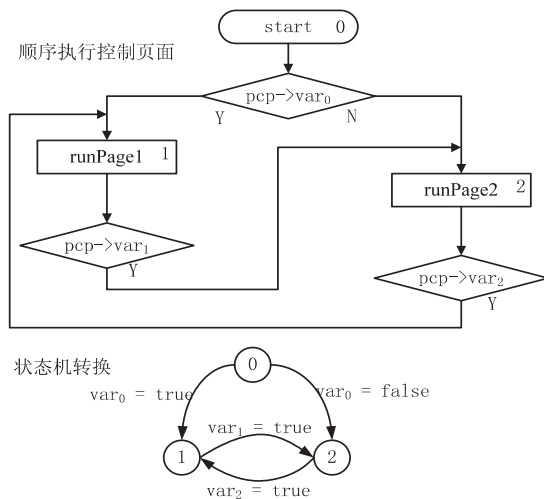


图3 顺序执行控制页面示例

其C函数代码原型定义为:

```
void sfcControlFunc(SfcComp * pcp)
```

```
{
switch(pcp->state) //transfer the state
{
case 0:
if(pcp->var0)
{ pcp->state = 1; }
else
{ pcp->state = 2; }
break;
case 1:
if(pcp->var1)
{ pcp->state = 2; }
break;
case 2:
if pcp->var2)
{ pcp->state = 1; }
break;
default:
pcp->state = 0;
break;
}
switch(pcp->state) //action
{
case 0: break;
case 1: runPage1(pcp); break;
case 2: runPage2(pcp); break;
default: break;
}
}
```

6 代码形成设计

6.1 模块划分

- 顺序控制图程序代码形成工具,划分为 5 个部分:
- (1) 符号数据模块:对程序设计页面中的各种符号进行建模。
 - (2) 数据组织模块:负责对页面、元件文件数据进行管理。
 - (3) 数据处理模块:形成元件内符号的连接关系并对符号进行拓扑排序;形成层次化元件关系;检查图形化程序设计规则;对预编译语句进行解析处理。
 - (4) 代码生成模块:形成元件的 h/c 源代码和配置文件。
 - (5) 编译链接模块:形成每个处理器的 MakeFile,并调用交叉编译器,形成可执行程序。

基于常规 FBD 模式的功能图代码形成,已有文献阐述^[6],下节阐述顺序控制图任务函数代码形成算法。

6.2 顺序控制任务函数代码形成算法

顺序控制任务函数是元件中最关键的周期运行函

数,该算法步骤如图 4 所示。

如图 4 所示,读取控制页面的数据,解析相关符号和连接线信息,形成各符号的数据类实例。之后从起始符号开始,按照深度优先遍历的排序算法,填充与符号输入点关联的前驱符号链表、填充与符号输出点关联的后继符号链表。之后汇总执行步个数,并设置各个执行步对应的状态编码。然后形成第 1、2 个 switch-case 语句结构代码,最后输出 C 代码文本。

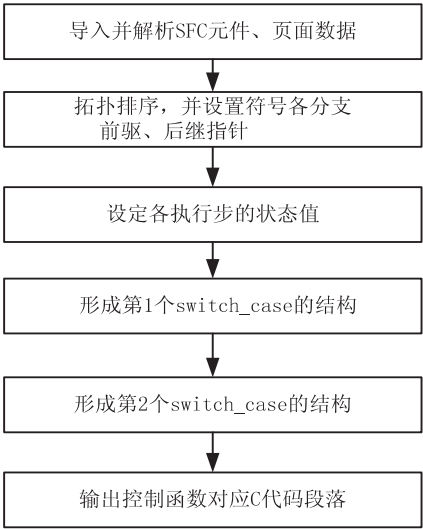


图 4 顺序控制任务函数形成步骤

第 1 个 switch_case 结构的关键算法代码为:

```
for(int i = 1; i <= stateNum; i++) {
m_txt += " case" + QString::number(i) ;
CSFCRunBlock * prb = searchStepRunBlock(i) ;
bool bHasSameNextSybl = false;
for(int j = i+1; j <= stateNum; j++) {
CSFCRunBlock * ptmp = searchStepRunBlock(j) ;
if(prb && ptmp) {
if(prb->m_nextSybl == ptmp->m_nextSybl) {
bHasSameNextSybl = true;
break;
} } }
if(bHasSameNextSybl) { m_txt += " : \n" ; }
else
{
m_txt += " : \n { \n" ;
QString str;
if(prb) makeSfcBlockCode(str, prb->m_nextSybl) ;
m_txt += str + " } \nbreak; \n \n" ;
} } }
```

第二个 switch_case 结构的关键算法代码为:

```
for(i = 1; i <= stateNum; i++) {
QList<CSFCRunBlock * > list;
m_txt += " case" + QString::number(i) ;
m_txt += " : \n" ;
searchStepRunBlockList(list, i) ;
```



```
foreach( CSFCRunBlock * prb,list)
{
    QString fname; getPageDocFuncName( fname,prb );
    QString refName = prb->getRefPageName( );
    CPage * pdoc = pcp->searchPage( refName );
    if( pdoc ) m_txt+= fname;
}
m_txt+=" break; \n \n" ;
}
```

7 结束语

文中提出一种基于 C 语言流程结构符号的方案实现顺序控制图的可视化建模方法,将页面分为顺序功能页面、执行步页面、功能图子页面,通过层次模型管理可视化程序,设计了顺序控制代码转换规则,并以双 switch-case 的代码模式提高了顺序控制图的运行效率。基于文中设计的顺序控制图已经在直流输电保护控制装置中进行了应用,满足了高实时性要求的顺序控制需求。

参考文献:

- [1] 王丁磊,郭 涛. PLC 顺序控制的编程方法及应用[J]. 机电一体化,2011(4):102-104.
- [2] 齐传刚,熊永超,邵 娜. PLC 的 SFC 图在编写顺序控制梯形图中的应用[J]. 辽宁工程技术大学学报:自然科学版,2006,25:234-236.
- [3] 时文飞. 数组式编程方法在 PLC 编程中的应用[J]. 电气传动,2014,44(1):85-87.

(上接第 148 页)

- [1] , IEEE, 2007; 120–125.
- [6] Dong N, Nguyen T, Xue Y. Multimedia wireless transmission with network coding [C]//Packet video 2007. Lausanne, Switzerland; IEEE, 2007; 326–335.
- [7] Seferoglu H, Markopoulou A. Video-aware opportunistic network coding over wireless networks[J]. IEEE Journal on Selected Areas in Communications, 2009, 27(5) : 713–728.
- [8] Seferoglu H, Markopoulou A. Opportunistic network coding for video streaming over wireless [C]//Packet video 2007. Lausanne, Switzerland; IEEE, 2007; 191–200.
- [9] Sundararajan J K, Sadeghi P, Médard M. A feedback-based adaptive broadcast coding scheme for reducing in-order delivery delay [C]//Proc of IEEE workshop on network coding, theory and application. Lausanne; IEEE, 2009; 1–6.
- [10] Ho T, Medard M, Koetter R, et al. A random linear network coding approach to multicast[J]. IEEE Transactions on Information Theory, 2006, 52(10) : 4413–4430.
- [11] Sadeghi P, Shams R, Traskov D. An optimal adaptive network

- [4] 翁元. 基于上位机的混合流程顺序控制系统[J]. 计算机系统应用, 2014, 23(6): 42-47.
- [5] 樊陈, 倪益民, 窦仁晖, 等. 智能变电站顺序控制功能模块化设计[J]. 电力系统自动化, 2012, 36(17): 67-71.
- [6] 刘克金, 陈宏君, 冯亚东, 等. 新一代控制保护系统可视化编程软件设计与实现[J]. 工业控制计算机, 2014, 27(10): 82-84.
- [7] 卢娟, 李兆成, 陈钢杰, 等. 保护与测控领域通用可视化平台的设计与实现[J]. 电力系统自动化, 2005, 29(4): 58-61.
- [8] 邓秋娥, 杜奇壮, 卢娟. 可视化编程在微机保护中的实现[J]. 继电器, 2008, 36(3): 1-4.
- [9] 姚成, 黄国方, 周邵亮, 等. 软 PLC 技术应用于智能保护测控装置的实现方案[J]. 电力系统自动化, 2010, 34(23): 115-118.
- [10] 冯亚东, 陈宏君, 刘克金, 等. 一种顺序功能图的可视化实现方法: CN, 201210036915. 9[P]. 2012-07-25.
- [11] Kernighan B K, Ritchie D M. The C programming language[M]. Beijing: China Machine Press, 2006.
- [12] Aho A V, Lam M S, Sethi R, et al. Compilers: principles, techniques, and tools[M]. 2nd ed. Beijing: China Machine Press, 1986.
- [13] Lewis R W. Programming industrial control systems using IEC 1131-3[D]. London: The Institution of Electrical Engineers, 1995.
- [14] Hoppe G. Software motion control with IEC61131-3 motion profile interface the system independent set of function blocks[C]//Proc of TECH/EXPO technology update conference. [s. l.]: [s. n.], 2001: 463-472.

coding scheme for minimizing decoding delay in broadcast erasure channels[J]. EURASIP Journal on Wireless Communications & Networking, 2010, 2010:50–50.

[12] Wang C C. On the capacity of 1-to-K broadcast packet erasure channels with channel output feedback[J]. IEEE Transactions on Information Theory, 2010, 58(2):1347–1354.

[13] Sorour S, Valaee S. Coding opportunity densification strategies for instantly decodable network coding[J]. IEEE Transactions on Communications, 2013, 61:5077–5089.

[14] Harris J M, Hirst J L, Mossinghoff M J. Combinatorics and graph theory[M]//Undergraduate texts in mathematics. Berlin; Springer, 2008.

[15] Yu M, Sadeghi P, Aboutorab N. On the throughput and decoding delay performance of instantly decodable network coding[J]. IEEE/ACM Trans on Networking, 2013, 67:1309–1310.

[16] Bron C, Kerbosch J. Algorithm 457: finding all cliques of an undirected graph[J]. Communication of the ACM, 1973, 16(9):575–577.