

DOI:10.3969/j.issn.1001-4551.2019.01.012

一种基于 OtoStudio 的数控系统 文件读写功能的软件开发^{*}

胡春晓,梅志千^{*},李向国

(河海大学 机电工程学院,江苏 常州 213022)

摘要:针对目前传统数控软件移植性、扩展性差,程序编写效率效率低等问题,对伺服压力机数控系统的文件读写模块进行了研究。提出了一种基于 IEC61131-3 工业控制语言标准的程序设计方法,对伺服压力机数控系统的文件读写功能进行了设计;在对读写文件的内容格式进行分析的基础上,选择 OtoStudio 软件作为开发平台,采用结构化文本编程语言和模块化的设计方法,对文件读写模块进行了编程,开发出了一种基于 OtoStudio 的数控系统文件读写软件,并在伺服压力机硬件平台上进行了文件读取和写入功能测试。研究表明:基于 OtoStudio 开发平台编写的文件读写模块,成功实现了工艺文件读写,执行速度快;基于模块化设计的程序逻辑简单、扩展性好、可移植性强,具有较高的实际应用价值。

关键词:文件读写;模块化设计;数控系统

中图分类号:TH39

文献标志码:A

文章编号:1001-4551(2019)01-0054-06

Software development of file reading and writing function of CNC system based on OtoStudio

HU Chun-xiao, MEI Zhi-qian, LI Xiang-guo

(College of Mechanical and Electrical Engineering, HoHai University, Changzhou 213022, China)

Abstract: Aiming at the problems existed in the traditional numerical control software, such as lack of good portability and scalability, programming inefficiency etc, a study on the filing reading and writing module of servo pressure CNC system was carried on. Based on IEC61131-3 industrial control standard, a program design method was proposed to design the function of file reading and writing for the servo pressure CNC system. On the basis of the analysis about the content format of the reading and writing files, OtoStudio software adopted as the development platform, filing reading and writing module was programmed by employing structural text programming language and modular design method, a file reading and writing software was developed. The test of file reading and writing worked out on the servo pressure hardware platform. The results indicate that the file reading and writing module programmed based on OtoStudio is achieved successfully to realize the process file reading and writing, while the execution correct results and fast speed are attained. The program design based on modular design is simple, extensible, and portable with high practical application value.

Key words: file reading and writing; modular design; CNC system

0 引 言

数控机床可极大地提高零件的加工质量和加工效

率,是装备制造业的研究热点^[1]。其中,数控系统作为数控机床的控制核心,其控制精度、功能与可靠性直接影响了机床的整体性能和性价比^[2]。

目前,绝大部分数控技术仍封闭在系统框架内部,

收稿日期:2018-08-30

基金项目:常州市特种机器人及智能技术重点实验室基金资助项目(M20133004)

作者简介:胡春晓(1990-),男,江苏沭阳人,硕士研究生,主要从事伺服压力机数控系统方面的研究。E-mail:215683953@qq.com

通信联系人:梅志千,男,教授,硕士生导师。E-mail:meizq@hhuc.edu.cn

不具备移植性、兼容性和二次开发的可能,影响了整个系统的维护、集成与扩展,而且各厂商之间的数控系统软件、硬件模块和体系结构遵循各自独特的标准,系统之间不能相互替代与共享^[3]。为摆脱这些问题,开放性数控系统的概念被提了出来。

与传统数控系统相比,开放性数控系统具有可扩展性、可移植性和互换性等优点。它降低了数控系统的开发难度,允许客户根据自身的需求对数控系统的功能进行扩展和更改,缩短了开发周期^[4]。

针对开放式数控系统,国内外许多学者进行了研究,丁许等^[5]通过 Visual Studio2010.NET 开发工具,采用 C#编程语言对数控系统软件进行了开发,数控系统结构采用了控制器 + PC 方式;李麟^[6]通过 QT 开发工具,采用 C++ 语言设计了伺服压力机数控系统软件,硬件上采用 STM32 控制器,提出了嵌入式、模块化的数控系统。虽然他们对数控系统的硬件结构采用了模块化的方式,具有开放性的优点,但数控软件的开发平台和编程语言并不一致,互相之间缺乏良好的移植性和互换性,同时程序的编写效率和扩展性较低。

IEC61131-3 是第一个为工业自动化控制系统的软件设计提供标准化编程语言的国际标准。基于此标准开发的软件开发平台具有良好的结构化编程环境,可缩短程序开发周期,提高软件开发效率,程序移植性和重用性好^[7]。

本研究将 OtoStudio 作为软件开发平台,对数控系统的文件读写功能进行程序设计,规定读写功能针对的文件内容结构,编程语言选择结构化文本(ST),采用模块化的编程方法,并在以驱控一体化控制器为核心的伺服压力机平台上进行读写功能的验证。

1 GTSD14 系列智能伺服驱动器简介

驱控一体化控制器集工业 PC、运动控制和伺服驱动于一体,与传统的一个控制器 + 多个驱动器的多轴控制模式相比,驱控一体化控制器可避免硬件重复浪费、系统调试与维修困难和抗干扰能力差等问题,具有很高的使用价值和研究价值。

本文驱控一体化控制器选用固高科技(深圳)有限公司的 GTSD14 系列智能伺服驱动器。该驱动器具有以下特点^[8]:电流、速度和位置全闭环控制,支持多轴运动控制;高度集成的系统架构极大简化了电气设计,提高了设备的性能和可靠性;配套的软件开发平台 OtoStudio,基于 WINCE 操作系统,满足客户对实时性、安全性和开放性的要求;支持高速本地 IO 和远程扩展 IO,带有 CAN 总线、eHMI 显示、USB、gLink、LAN 等控

制单元接口;可实现整套数控系统的开发。

2 文件操作功能的设计

2.1 OtoStudio 简介

OtoStudio 是一款基于 IEC61131-3 工业控制语言标准开发的组态软件。它包含控制方案编辑器、仿真调试器和离线仿真调试工具^[9],支持 6 种编程语言:结构化文本(ST)、顺序功能图(SFC)、梯形图(LD)、指令表(IL)、连续功能图(CFC)和功能块图(FBD)。此外,它还具有灵活的人机界面组态。因此本文选择 OtoStudio 作为软件开发平台来进行文件读写功能的程序编写。

2.2 读写功能的程序设计

本文采用 ST 编程语言编写读写程序,编写方式采用模块化,以便于后续功能的修改、删除和增加。

2.2.1 读取功能的程序设计

本文读写程序针对的文件内容结构规定如下:

[参数段 1](* 标题行 *)

参数名 11 = 参数 11(* 参数行,参数:参数 11 *)

参数名 12 = 参数 12(* 参数行,参数:参数 12 *)

.....

[参数段 2](* 标题行 *)

参数名 21 = 参数 21(* 参数行,参数:参数 21 *)

参数名 22 = 参数 22(* 参数行,参数:参数 22 *)

.....

.....

[参数段 N](* 标题行 *)

参数名 N1 = 参数 N1(* 参数行,参数:参数 N1 *)

参数名 N2 = 参数 N2(* 参数行,参数:参数 N2 *)

.....

[读取结束段](* 标题行 *)

读取结束标志 = 1(* 参数行,参数为:1 *)

读取程序(ReadFile)要实现目标内容的读取,需依次调用 OtoStudio 的 3 个库函数:SysFileOpen、SysFileRead 和 SysFileClose。SysFileOpen 能打开已有的文件,它的返回值是一个文件编号,若打开错误,将返回 0,此文件编号将作为 SysFileWrite、SysFileRead、SysFileClose 中一个输入变量;SysFileRead 能从 SysFileOpen 打开的文件中读取数据;SysFileClose 用于关闭 SysFileOpen 打开的文件。

程序 ReadFile 的读取功能具体通过调用函数 ReadParam 来实现。函数 ReadParam 设有 2 个输入:BOOL 型变量 Enb(使能)和 STRING 型变量 FilePath

(目标文件路径)。在此函数中,首先通过库函数 SysFileOpen 打开输入 FilePath 对应的目标文件,若此库函数返回值为 0,函数 ReadParam 返回值置为 FALSE,参数读取失败;若返回值不为 0,则执行函数 TakeoutParam,读取目标文件中所有参数,若函数 TakeoutParam 返回值为 TRUE,参数读取成功,函数 ReadParam 返回值置为 TRUE;否则,参数读取失败,函数 ReadParam 返回值置为 FALSE。

函数 TakeoutParam 设有 1 个输入:DWORD 型变量 FN(上述库函数 SysFileOpen 的返回值),此函数的程序设计流程如图 1 所示。

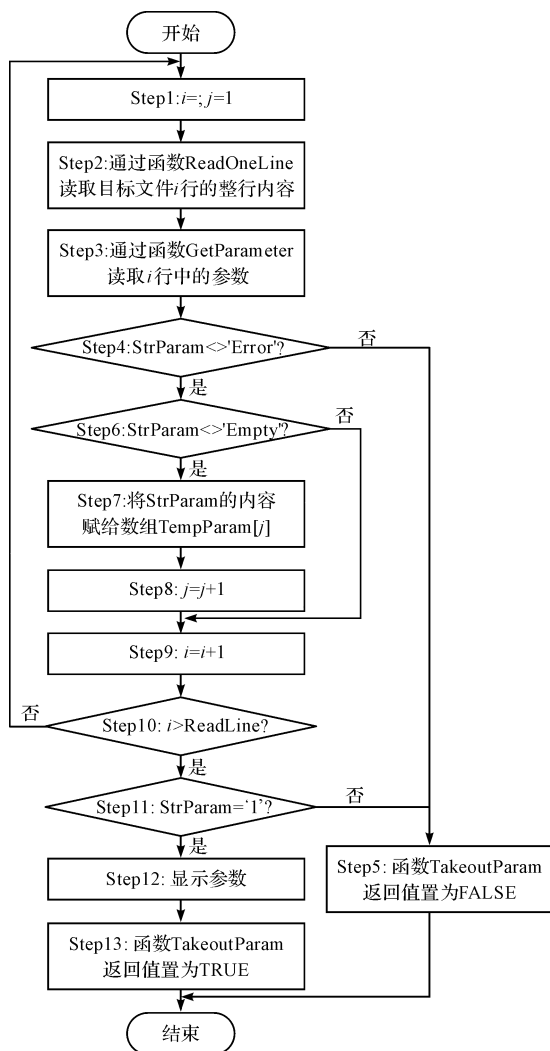


图 1 函数 TakeoutParam 的程序设计流程图

现结合上列的内容结构规定,对图 1 进行解释。首先,假设内容结构规定中的参数为 INT 型,针对内容结构规定中参数的显示,先定义结构体 Vshow:

TYPE Vshow;

STRUCT

var11: INT;(* 对应内容结构的参数 11 *)

var12: INT;(* 对应内容结构的参数 12 *)

.....

var11: INT;(* 对应内容结构的参数 21 *)

var12: INT;(* 对应内容结构的参数 22 *)

.....

varN1: INT;(* 对应内容结构的参数 N1 *)

varN2: INT;(* 对应内容结构的参数 N2 *)

END_STRUCT

END_TYPE

在上述结构体定义中,由于内容结构规定中的读取结束标志 1 只用于判断内容是否读取结束,所以不需要定义对应的变量。

再定义结构体 Vshow 的全局型实例 V,用于显示内容结构规定中的参数,其声明如下:

VAR_GLOBAL

V: Vshow;

END_VAR

以内容结构规定的第 2 行参数读取为例,在图 1 的 Step2 中,通过函数 ReadOneLine 读取出第 2 行;参数名 11 = 参数 11MYMRMYMN。函数 ReadOneLine 设有 1 个输入:DWORD 型变量 ReValue(函数 TakeoutParam 输入 FN 的值),在此函数中,通过输入 FN,执行库函数 SysFileRead 来实现目标文件的 i 行的整行内容读取,并将读取后的整行内容作为函数 ReadOneLine 的 STRING 型返回值。

再执行图 1 中的 Step3,通过函数 GetParameter,从函数 ReadOneLine 返回的字符串中读取参数:参数 11。此函数设有 2 个输入:STRING 型变量 StrOneLine(函数 ReadOneLine 的返回值)、STRING 型变量 StrBegin(i 行对应的格式正确的标题行或参数名)。针对函数 GetParameter 输入 StrBegin 的内容赋入,需定义一个全局 STRING 型数组 A,并依次赋入结构规定的所有标题行(例如:A[1]:=[参数段 1])和参数名(例如:A[2]:=[参数名 11]),然后将 A[i]赋给输入 StrBegin。

在函数 GetParameter 中,首先判断内容格式规定中的 i 行是否为参数行,编写以下程序段:

Pos:=FIND(StrBegin,[]);

若 i 行不为参数行(Pos=1),函数 GetParameter 返回值为字符串:Empty;若 i 行为参数行(Pos<>1),则开始检查参数行的参数名部分是否正确,检查参数名部分的程序段如下:

StrBeginPos:=FIND(StrOneLine,StrBegin);

若参数名部分不正确(StrBeginPos<>1),函数 GetParameter 返回值为字符串:Error;若参数名部分正确(StrBeginPos=1),取出 i 行的参数,并将参数作为

此函数返回值,参数读取程序段编写如下:

```
Len1 := LEN(StrOneLine);
Len2 := LEN(StrBegin);
Str := RIGHT(StrOneLine, (Len1 - Len2 - 1));
GetParameter := LEFT(Str, (Len1 - Len2 - 3));
```

在上述 3 段程序段中,调用了 OtoStudio 的库函数: FIND、LEN、RIGHT 和 LEFT。FIND(STR1, STR2) 用于在一个字符串 STR1 中查找到字符串 STR2; LEN(STR) 用于读取字符串 STR 的长度; RIGHT(STR, SIZE) 用于从字符串 STR 的右边取第一个 SIZE 字节; LEFT(STR, SIZE) 用于在字符串 STR 中,从右边取第一个 SIZE 字节。

在图 1 的 Step4、Step6、Step7 和 Step11 中, STRING 型变量 StrParam 用于存放 GetParameter 的返回值, Step7 中的数组 TempParam 定义如下:

```
VAR_GLOBAL
TempParam: ARRAY[1..30] OF STRING;
END_VAR
```

当内容结构规定中第 2 行的参数,通过 Step7 放入数组变量 TempParam[i] 中后,执行 Step10。

在图 1 的 Step10 中, INT 型全局变量 ReadLine 的值为内容结构规定的总行数。当内容结构规定的所有行读取完毕,即 $i > \text{ReadLine}$, 执行 Step11。

在 Step11 中,当 StrParam = 1 时,表示已成功读完内容结构规定中最后一行,执行 Step12。

在 Step12 中,通过显示函数,将数组 TempParam 中存放的参数进行类型转换,然后对应地赋给结构体变量 V 中的元素,以内容结构规定第 2 行参数的类型转换和赋给的程序段为例:

```
V.var11 := STRING_TO_INT(TempParam[1]);
```

STRING_TO_INT(var) 为 OtoStudio 的一个转换操作符,可将 STRING 型变量的内容转换为 INT 型。

2.2.2 写入功能的程序设计

写入程序(SaveFile)是以文件的形式对目标参数进行保存。此程序功能的实现需依次调用 OtoStudio 的 3 个库函数: SysFileOpen、SysFileWrite 和 SysFileClose,库函数 SysFileWrite 能将数据写入用库函数 SysFileOpen 打开的文件内。

程序 SaveFile 的写入功能具体通过调用函数 SaveParam 来实现。此函数有 2 个输入: BOOL 型变量 Enb(使能)、STRING 型变量 FilePath(目标文件的路径)。在函数 SaveParam 中,首先通过输入 FilePath 执行库函数 SysFileOpen,若此库函数返回值为 0,则参数保存失败,函数 SaveParam 返回值置为 FALSE;否则开

始调用组装函数,以按行组装的方式,将参数组装成 2.2.1 节中的文件内容结构规定,组装函数中的程序编写流程如图 2 所示。

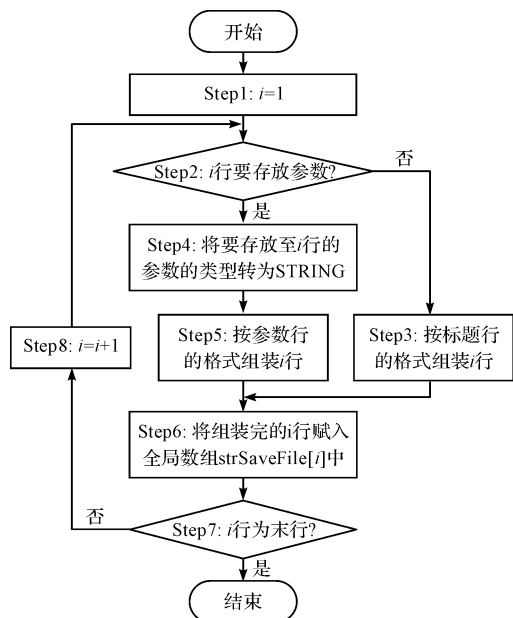


图 2 组装函数的程序编写流程图

在图 2 中的 Step6 中,数组 strSaveFile 定义如下:

```
VAR_GLOBAL
strSaveFile: ARRAY[1..200] OF STRING;
END_VAR
```

现根据 2.2.1 节中的内容结构规定,以第 1 行和第 2 行的程序编写对图 2 进行解释。由于内容结构规定的第 1 行为标题行,根据图 2 中的 Step2、Step3 和 Step6,将 2.2.1 节中数组 A[1] 内的字符串[参数段 1]组合为:[参数段 1]MYMRMYMN,对应的程序段编写如下:

```
strSaveFile[1] := CONCAT(A[1], MYMRMYMN);
```

文件内容结构的第 2 行为参数行,根据图 2 中的 Step2、Step4、Step5 和 Step6,将 2.2.1 节中数组 A[2] 和结构体变量 V.var11 内的参数组合为:参数名 11 = 参数 11MYMRMYMN,对应的程序段编写如下:

```
strVar := INT_TO_STRING(V.var11);
```

```
str := CONCAT(A[2], '=' );
```

```
str := CONCAT(str, strVar);
```

```
str := CONCAT(str, MYMRMYMN);
```

```
strSaveFile[2] := str;
```

上述程序段的 str 为组装函数的 STRING 型局部变量, INT_TO_STRING(var) 为 OtoStudio 的一个转换操作符,可将 INT 型变量的内容转换为 STRING 型,并将转结果作此转换操作符的输出。库函数 CONCAT 能将两个字符串进行组合,并将组合后的字符串作为

此库函数的输出。

然后执行函数 SavetoFile,将组装好的所有行(数组 strSaveFile 的内容)写入目标文件内。该函数有 1 个输入:DWORD 型变量 FN(存放函数 SaveParam 中 SysFileOpen 的返回值)。在函数 SavetoFile 中,通过 FOR 循环程序,重复执行库函数 SysFileWrite,将数组 strSaveFile 的内容写入目标文件内,FOR 循环次数等于 INT 型全局变量 SaveLine 的值,循环结束后,执行库函数 SysFileClose 关闭函数 SavetoFile 输入 FN 对应的文件。

当函数 SavetoFile 执行完毕后,函数 SaveParam 返回值置为 TRUE,参数保存成功。

3 应用实例

3.1 参数的读取

以伺服压力机的工艺文件中参数读取为例。工艺文件中要求显示的参数有:返上死点速度、零件名称、模具型号和七段行程参数。

现要读取一文件名为 CraftFile1 的工艺文件中的参数。此文件在 GTSD14 系列智能伺服驱动器中的地址为:\Hard Disk\CraftWork1\Craftfile1. PRM,参数为:SysMfile(模具型号)、001(零件名称)、4(返上点速度,单位为 mm/ms)和七段行程参数。其中,七段行程参数具体如表 1 所示。

表 1 工艺文件 CraftFile1 中的七段行程参数

行程段	段速度/(mm·ms ⁻¹)	目标角度/(°)	保压时间/ms
1	8	120	0
2	3	150	0
3	2	180	1 000
4	4	230	0
5	0	0	0
6	0	0	0
7	0	0	0

选择和打开文件 CraftFile1 的操作如图 3 所示。



图 3 选择打开功能和目标文件 CraftFile1

在图 3 中,首先在工艺文件参数显示和输入界面 1 中点击功能菜单按钮 2,显示文件操作框 3,再点击打开功能按钮 5,选择文件 CraftFile1,最后点击打开按钮 6,文件操作框 3 消失,运行程序 ReadFile。

通过图 3 操作后,文件 CraftFile1 中要求显示的参数在工艺文件参数显示和输入界面中的显示结果如图 4 所示。



图 4 工艺文件 CraftFile1 中的参数读取显示

在图 4 中,七段行程的段速度和返上死点速度值,从文件 CraftFile1 中读取完毕后,需再乘以 5,通过换算后,才能在界面中显示。

3.2 参数的写入

以伺服压力机的工艺参数的保存为例,现要在图 4 界面中输入以下参数:SysMfile1(模具型号)、002(零件名称)、4(返上死点速度)和七段行程参数。其中,七段行程参数具体如表 2 所示。

表 2 工艺参数中的七段行程参数

行程段	段速度/(mm·ms ⁻¹)	目标角度/(°)	保压时间/ms
1	30	90	0
2	15	150	0
3	10	180	1 500
4	0	0	0
5	0	0	0
6	0	0	0
7	0	0	0

现将上述工艺参数保存至 GTSD14 系列智能伺服驱动器中名为 CraftFile2 的工艺文件内,此文件地址为:\Hard Disk\CraftWork2\CraftFile2. PRM。

本研究对上述工艺参数进行保存操作。首先,在图 4 中,将上述所有工艺参数对应地赋入文本显示框内;然后,点击功能菜单按钮 2,显示文件操作框 3;再选择保存功能按钮 4,此时文件操作框的 3 中的打开按钮 6 对应地替换为保存按钮;再选择文件 CraftFile2;最后,点击保存按钮,文件操作框 3 消失,运行程序 SaveFile。

通过上述的保存操作后,文件 CraftFile2 中的内容截图如图 5 所示。

[Process Configuration] ModelType=SysMfile1 ElementName=002 BackSpeed=4	[Object Vel]	[Object Angle]	[Pressure Tim]	[FileFlag] IniFileFlag=1
	Vel1=6	Angle1=90	Tim1=0	
	Vel2=3	Angle2=150	Tim2=0	
	Vel3=2	Angle3=180	Tim3=1500	
	Vel4=0	Angle4=0	Tim4=0	
	Vel5=0	Angle5=0	Tim5=0	
	Vel6=0	Angle6=0	Tim6=0	
	Vel7=0	Angle7=0	Tim7=0	
(a) 模具类型等部分	(b) 段速度部分	(c) 目标角度部分	(d) 保压时间部分	(e) 读取结束标志部分

图 5 工艺文件 CraftFile2 中的内容

在图 5 中,在工艺文件参数显示和输入界面中输入的七段行程的段速度和返上死点速度值,需再除以 5,通过换算后,才能写入文件 CraftFile2 内。

4 结束语

笔者通过对伺服压力机数控系统的文件读写功能进行研究,开发了一种基于 OtoStudio 的伺服压力机数控系统软件;采用模块化的设计方法,分别编写了文件读取功能模块和文件写入功能模块,并对读写文件的内容格式进行了规定,最后在以驱控一体化控制器为核心的伺服压力机平台上进行了读写功能的验证。

研究表明:基于 OtoStudio 开发平台编写的文件读写功能模块,成功实现了工艺文件读写,执行结果正确、执行速度快;基于模块化设计的程序逻辑简单、扩展性好、可移植性强,作为一款开放性的数控系统软件,具有较高的实际应用价值。

参考文献 (References):

[1] 方晨曦,叶佩青. 面向高速高精加工的高性能数控系统展

望[J]. 航空制造技术,2014,447(3):44-47.
[2] 蔡锐龙,李晓栋,钱思思. 国内外数控系统技术研究现状与发展趋势[J]. 机械科学与技术,2016,35(4):493-500.
[3] 周胜凯,刘 强,李传军,等. 开放式数控系统平台的设计与开发[J]. 机械设计与制造,2015(1):169-171.
[4] 王程鹏. 开放式数控系统的现状与发展趋势[J]. 科技传播,2013(3):220-221.
[5] 丁 许,何 宁,李 亮,等. 基于 PMAC 的微细铣床数控系统软件开发[J]. 制造技术与机床,2016(8):176-180.
[6] 李 麟. 伺服压力机控制系统设计[D]. 武汉:华中科技大学材料科学与工程学院,2013.
[7] 王蔚庭. 解析工业编程语言国际标准 IEC61131-3[J]. 国内外机电一体化技术,2007(2):47-62.
[8] 固高科技(深圳)有限公司. GTSD 通用型智能伺服驱动器用户手册[M]. 深圳:固高科技(深圳)有限公司,2016.
[9] 固高科技(深圳)有限公司. CPAC-OtoStudio 可视化界面开发手册[M]. 深圳:固高科技(深圳)有限公司,2011.

[编辑:周昱晨]

本文引用格式: