# 基于 JSON 的软 PLC 梯形图存储

谢俊达* 蒋 近 钱美容
XIE Jun-da JIANG Jin QIAN Mei-rong

**摘 要** 关于软 PLC 梯形图文件的保存，传统软件都是以 XML 文件的形式来存储。本文介绍了一种轻量级的数据交换格式 JSON 替代 XML 作为梯形图的元件的存储格式。详细阐明了 PLC 梯形图程序与 JSON 文件之间的转换实现思路。通过实例实现了梯形图文件与 JSON 文件之间的转换。

**关键词** 软 PLC；JSON

doi：10.3969/j.issn.1672-9528.2018.h2.023

## 0 引言

可编程控制器（PLC）是一种基于工业控制的计算机或嵌入式 PC 的通用自动控制装置，其拥有抗干扰能力强，可靠性高，结构简单，通用性强等特点。软 PLC 编译器在工业自动控制领域中有广泛的应用 [1]。

JSON(JavaScript Object Notation) 是一种轻量级的数据交换格式。它是基于 ECMAScript (w3c 制定的 js 规范）的一个子集，其文本格式的存储与数据表示采用了独立于编程语言的形式。因其简洁和清晰的层次结构使得 JSON 成为了优秀的数据交换语言。其拥有易于读写，易于机器的解析和生成的特点，其在 web 领域的应用有效地提升了网络传输效率 [2]。

## 1 存储梯形图的数据结构

梯形图是 PLC 编程的图形编程语言，它具有实用、形象、直观等特点，这些特点让 PLC 编程人员能够熟练地使用并进行 PLC 程序的编写。然而我们需要将梯形图语言输入到计算机中，必须要有其相对应的数据结构，即将梯形图所对应的逻辑关系用计算机语言表示出来，生成数据文件，通过对这些数据文件的解析生成 PLC 的执行代码。

为了能将梯形图语言在计算机中表示并实现其逻辑关系，有的软 PLC 编译器将梯形图表示为有向图，利用拓扑排序将梯形图与指令表之间的逻辑关系转换。常见的梯形图数据结构采用二叉树存储方式。

二叉树的节点有元件，并联块和串联块等，但二叉树对于一些特殊的功能块会比较难以处理。由于 JSON 的自描述性，用户可以非常方便地对梯形图功能块用自己的标签来定义，这弥补了二叉树描述文件的不足。

## 2 用 JSON 描述梯形图

### 2.1 JSON 格式的特点

JSON 作为 web 应用中最常见的数据交换格式，逐渐取代了 XML 的地位，它拥有如下特点：

1、轻量级的数据交换格式。

2、人们读写更加容易。

3、易于机器的解析和生成。

4、JSON 支持多语言。包括：Python，C，C++，Java，JavaScript 等。

这些特点让 JSON 作为梯形图的逻辑关系表示非常容易，并且使得跨平台的数据表示更为便利。其自描述性也让用户自定义标记及文档结构，这些特点使得 JSON 成为了最热门的数据交换格式。因此 JSON 被设计作为 PLC 梯形图的存储格式，能够轻松的满足 PLC 程序跨平台的要求。各个厂家可以按照相应的规则来将 JSON 文件中的逻辑关系转换为自己的梯形图或者助记符程序。

### 2.2 将梯形图表示为 JSON 文件

由 JSON 文件结构可知，用 JOSN 格式来描述梯形图是可行的。在梯形图编辑器中，梯形图元件都有其相应的位置坐标，其表示的是各个元件之间的位置和逻辑关系。同时，相应的元件拥有其相应的元件名称，元件标号和参数。由于 JSON 格式的特点是 key-value（属性－值）的形式，因此我们可以将相应的元件名称或功能块的名称定义为 JSON 格式的属性值，这样可以将梯形图的逻辑关系用 JSON 文件描述出来。

以一个串并联开闭触点梯形图程序为例如图 1：NET1 中的输出 Q0.0 接了一个串联块，该串联块包含了一个独立元件 I0.3 和一个并联块组成。这个并联块又包含了拥有两个

* 湘潭大学 湖南湘潭 411105



?1994-2019 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

元件 I0.1 和 I0.2 的串联块和单独一个元件 I0.0 的串联块。NET2 中则是由一个串联块组成，它包含了触点 I0.4 和触点 I0.5。

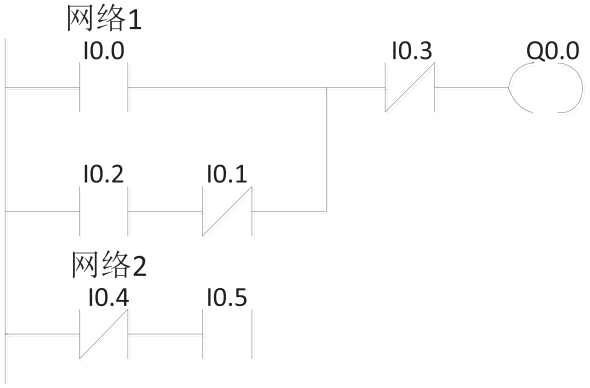


图 1 串并联开闭触点梯形图程序

## 3 JSON 与梯形图的转换算法实现

### 3.1 用前序遍历二叉树转 HashMap 算法

基于文献 [5] 的转换算法，我们可以将梯形图转换为二叉树结构。二叉树的结构如图 2 所示。而要转换为 JSON 文件格式，我们必须将二叉树结构转换为数组，HashMap 等易于转换的数据结构。梯形图转 JSON 算法流程图如图 3。根据二叉树的特点，我们可以将二叉树通过前序遍历的算法转换为 HashMap 的数据结构，具体的算法规则为：对二叉树进行前序遍历，每遍历到一个节点 node，创建一个 key 为当前节点值，value 为 HashMap1 的 HashMap 结构，判断其有无左子树，若有则将该 HashMap1 的 key 值设为 left，value 值设为 node.left，右子树判断同理。最后递归调用 node.left 和 node.right。算法流程图如图 4 所示。

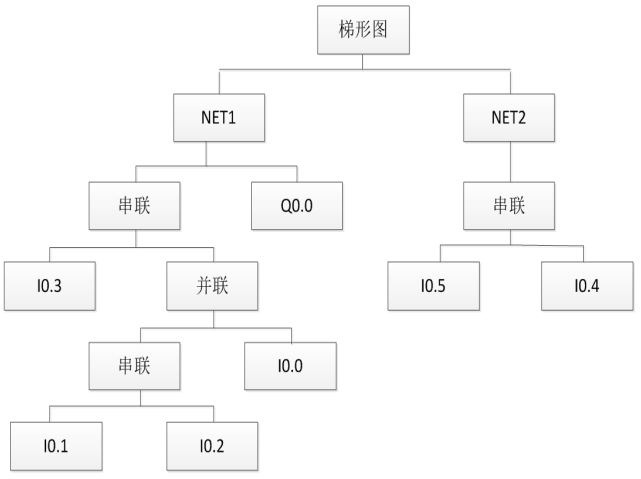


图 2 梯形图的二叉树结构

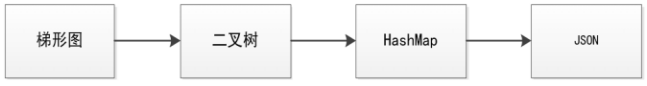


图 3 梯形图转 JSON 算法流程图

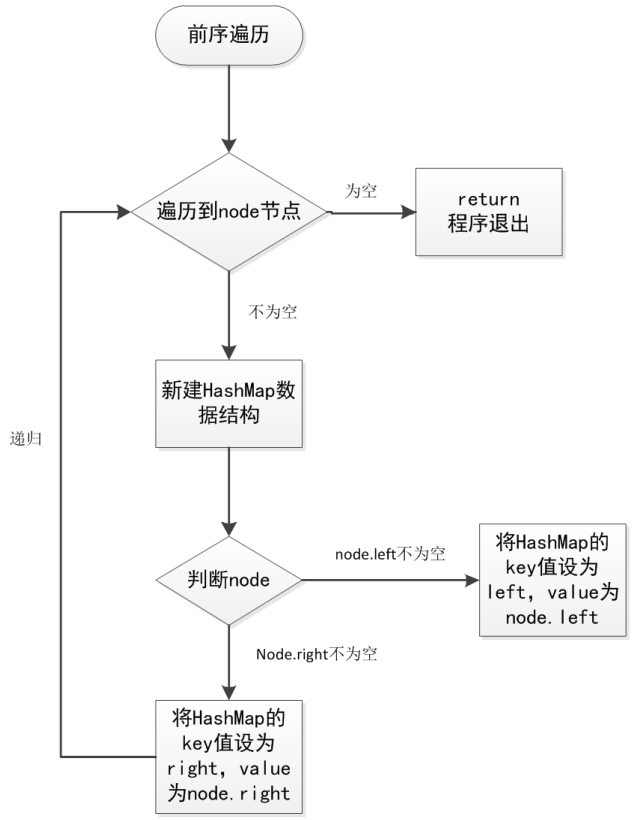


图 4 前序遍历二叉树转 HashMap 算法

其伪代码如下：

```
HashMap dict = new HashMap();                    //新建一个 HashMap 结构
void travel(node) {
    if (node == null) {
        return
    }
    dict[node.value] = new HashMap();
// 以 node 为 key 新建一个值为 Hash 的结构
    if (node.left) {
        dict[node.value][“left”] = node.left;
    }
    if (node.right) {
        dict[node.value][“right”] = node.right;
    }
    travel(node.left);
// 递归遍历
    travel(node.right);
}
```

根据上述算法，可得梯形图键值对（HashMap）表示规则如下：



?1994-2019 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

```
{
  NET1: {
    串联块：{
      并联块：{
        串联块：{
          常开触点：{
            X 坐标：“1”，
            Y 坐标：“1”，
            标号：“I0.2”
          },
          常闭触点：{
            X 坐标：“2”，
            Y 坐标：“1”，
            标号：“I0.1”
          }
        },
        常开触点：{
          X 坐标：“1”，
          Y 坐标：“2”，
          标号：“I0.0”
        }
      },
          常闭触点：{
            X 坐标：“3”，
            Y 坐标：“1”，
            标号：“I0.3”
          }
        },
        输出：{
          X 坐标：“4”，
          Y 坐标：“1”，
          标号：“Q0.0”
        }
      },

  NET2: {
    串联块：{
      常闭触点：{
        X 坐标：“1”，
        Y 坐标：“3”，
        标号：“I0.4”
      },
      常开触点：{
        X 坐标：“3”，
        Y 坐标：“3”，
        标号：“I0.5”，
      }
    }
  }
}
```

3.2 HashMap 转 JSON 文本

我们可知，以上键值对数据结构即为 JAVA HashMap 对象，我们利用 JAVA 的 fastJson 库即可完成 JAVA 对象与 JSON 文本之间的转换：

```
import com.alibaba.fastjson.JSON;
String jsonString = JSON.toJSONString(obj);
```

转换为 JSON 格式 jsonString 如下：

{“NET1”:{“串联块”:{“并联块”:{“串联块”:{“常开触点”:{“X 坐标”:”1”,”Y 坐标”:”1”,”标号”:”I0.2”},”常闭触点”:{“X 坐标”:”2”,”Y 坐标”:”1”,”标号”:”I0.1”}},”常开触点”:{“X 坐标”:”1”,”Y 坐标”:”2”,”标号”:”I0.0”}},”常闭触点”:{“X 坐标”:”3”,”Y 坐标”:”1”,”标号”:”I0.3”}},”输出”:{“X 坐标”:”4”,”Y 坐标”:”1”,”标号”:”Q0.0”}},”NET2”:{“串联块”:{“常闭触点”:{“X 坐标”:”1”,”Y 坐标”:”3”,”标号”:”I0.4”},”常开触点”:{“X 坐标”:”3”,”Y 坐标”:”3”,”标号”:”I0.5”}}}}

3.3 JSON 文本转梯形图算法

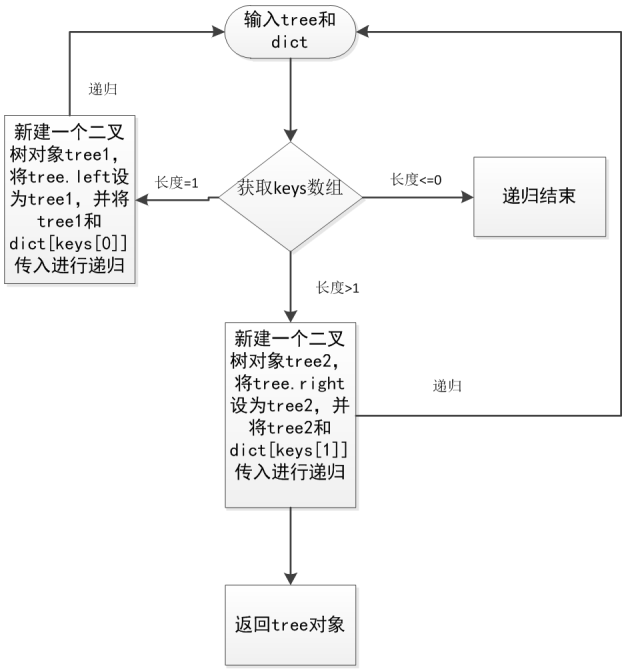


图 5 HashMap 转二叉树算法

我们同样利用 fastJson 库来将 JSON 对象转换为 HashMap 对象，我们的目标是转换为指令表语言，那么首先必须将 HashMap 对象转为二叉树数据结构，再利用后续遍历可得



?1994-2019 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

指令表。算法描述：创建一个二叉树对象 tree，获取 HashMap 对象的 keys 数组，若数组长度小于等于 0，则算法退出。若大于 0，则新建一个二叉树对象 tree1，value 为对数组的第一个对象的值，tree 的左子树设为 tree1，将 HashMap 对象与 tree1 传入函数进行递归。若大于 1，则新建一个二叉树对象 tree2，value 为对数组第二个对象的值，tree 的右子树设为 tree2，将 HashMap 对象与 tree2 传入函数进行递归。

程序流程图如图 4 所示。

程序伪代码如下：

```
Tree getTree(tree, dict) {
  var keysArray = Object.keys(dict);
  if (keysArray.length <= 0) {
    return;
  }
  var flag = 1;
   if ((keysArray[0] && dict[keysArray[0]] instanceof Object) || (keysArray[0] && keysArray.indexOf(“标号”) >= 0)) {
    var tree1 = new Tree();
    tree1.value = keysArray[0];
    if (keysArray.indexOf(“标号”) >= 0) {
      tree1.value = dict[“标号”];
      flag = dict[“标号”];
    }
    tree.left = tree1;
    getTree(tree.left, dict[keysArray[0]]);
  }
   if ((keysArray[1] && dict[keysArray[1]] instanceof Object) || (keysArray[1] && keysArray.indexOf(“标号”) >= 0)) {
    var tree2 = new Tree();
    tree2.value = keysArray[1];
    if (keysArray.indexOf(“标号”) >= 0) {
      tree2.value = dict[“标号”];
      if (dict[“标号”] === flag) {
        tree2.value = null;
      }
    }
    tree.right = tree2;
    getTree(tree.right, dict[keysArray[1]]);
  }
  return tree;
}
```

通过上述算法得到梯形图的二叉树形式后，我们可以采用后序遍历的方式获得程序的指令表。

## 4 梯形图程序转换为 JSON 与 XML 文件的对比

我们知道 JSON 文本比 XML 文件体积要小，具体对比我们可以通过测试同一梯形图程序分别转换为 JSON 和 XML 文件来得知。首先将图 1 的梯形图程序通过传统的 PLC 软件转换为 XML 文件，得到 plc.xml 文件。接着通过本文提出的转换算法转换相同的梯形图程序，可得 plc.json 文件。通过对比可得，XML 文件的大小为 550 字节，JSON 文件的大小为 430 字节，由此我们可以得知 JSON 文件比 XML 文件体积小了将近 21.8%。

## 5 结束语

本文提出了一种基于 JSON 的软 PLC 梯形图存储结构，以 JSON 文件保存的 PLC 程序，体积与传统软件采用的 XML 文件格式方法相比更小，且符合 IEC6113-1 的编程标准，利用 JSON 跨平台，自描述性的特点，可以轻松地在任意语言平台上转换存储。后文提出了二叉树转 HashMap 算法，采用前序遍历的递归形式，再运用 JSON 库转换为 JSON 文件。由 HashMap 转换为二叉树的算法，采用了递归的形式进行转换，达到了转换的目的。最后通过一个梯形图实例，将梯形图程序转换为 JSON 格式文本。

## 参考文献

[1] 魏跃国；陈彬兵 .PLC 技术的发展趋势，科技信息（科学教研），2014 年 22 期 .

[2] 仇小花，秦栓栓，邱果；基于 WEB 开发中的 XML 与 JSON 数据传输格式研究 [J]. 信息技术与信息化，2017,04:123-125.

[3] 程周著 .PLC 技术与应用 [M]. 福建：福建科学技术出版社，2015.

[4] 马远佳；基于 .NET 的 PLC 程序与 XML 文件的转换 [J]. 中国仪器仪表，2008，3(5)：54-57.

[5] 葛芬，吴宁 . 基于 AOV 图及二叉树的梯形图与指令表互换算法 [J]. 南京航空航天大学学报，2006,38(6):754-758.

【作者简介】

谢俊达，出生年份：1993 年，性别：男，籍贯：湖南株洲，工作单位：湘潭大学，职务：学生，专业学位：控制工程，研究方向：软 PLC 编译器。

（收稿日期：2018-01-22）



?1994-2019 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net