**LvData🡨🡪MemoryData**

**使用说明**

**编制：xxx**

**2020年05月15日**

协议修改历史

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **序号** | **修改内容** | **修改人** | **日期** | **协议版本** |
| 1 | 编制 |  |  | V1.00 |
| 2 |  |  |  |  |
| 3 |  |  |  |  |
| 4 |  |  |  |  |
| 5 |  |  |  |  |
| 6 |  |  |  |  |
| 说明：  1：协议版本号采用三个数据位标识。例：V1.00  2：每修改一次协议，需要将协议文件的版本号加1，同时要在协议修改历史中写出修改的内容。 | | | | |

# 概述

LvData🡨🡪MemoryData函数库主要实现LabVIEW数据类型与数据内存格式之间的转换，利用统一的内存格式作为数据接口，解决了LabVIEW与其他编程环境进行数据交互的问题。根据应用场景不同分类PTR Mode(指针传递模式)和NOPTR Mode(无指针传递模式)。

# LvData🡨🡪MemoryData数据类型

## 2.1 PTR Mode

PTR Mode将复杂的数据类型通过指针传递，并没有传输真实的原始数据，而是通过指针访问内存空间获取数据信息。因此PTR Mode只能在本机使用，不能作为远程传输。针对LabVIEW在测试测量过程的应用，主要支持的数据类型如下表所示：

|  |  |  |
| --- | --- | --- |
| 变量类型 | 变量长度 | 说明 |
| Boolean | 1个字节(8位) | 0表示False，1表示True |
| I8 | 1个字节(8位) | 有符号字节型 |
| U8 | 1个字节(8位) | 无符号字节型 |
| I16 | 2个字节(16位) | 有符号短整型 |
| U16 | 2个字节(16位) | 无符号短整型 |
| I32 | 4个字节(32位) | 有符号整型 |
| U32 | 4个字节(32位) | 无符号整型 |
| I64 | 8个字节(64位) | 有符号长整型 |
| U64 | 8个字节(64位) | 无符号长整型 |
| SGL | 4个字节(32位) | 单精度浮点型 |
| DBL | 8个字节(64位) | 双精度浮点型 |
| Timestamp | 8个字节(64位) | 时间戳。使用1904.1.1 00:00:00 作为基准，DBL的值表示当前时间举例基准时间的差值，单位为秒(s). |
| Array | **PTR Mode:**  12个字节（96位） | **PTR Mode:**  长度=数据长度(U32)+数据首地址(U64) |
| Waveform | **PTR Mode:**  12个字节（96位） | Waveform = Array(T0(DBL) + deltaT(DBL) +Data(DBL Array))  **PTR Mode:**  长度=数据长度(U32)+数据首地址(U64) |
| Cluster | 根据组成元素变化 | Cluster 的组成元素按照顺序排列，长度为组成元素长度之和。 |
| String | 1024个字节，当作U8Array解析 | 不包含长度信息，长度小于1024时，空位补0 |
| Path | 1024个字节，当作U8Array解析 | 不包含长度信息，长度小于1024时，空位补0 |

举例：

|  |  |  |  |
| --- | --- | --- | --- |
| LabVIEW数据 | C语言数据 | Binary数据格式 |  |
|  | struct NI\_Array{  unsigned long Len;  unsigned long long Addr;  }; | 04000000  80043F1500000000 | 数组长度(4Byte) Len  数组首地址(8Byte) Addr |
|  | struct NI\_Array{  unsigned long Len;  unsigned long long Addr;  }; | 06000000  303D481500000000 | Array(T0(DBL)+deltaT(DBL)+Data(DBL Array))  数组长度(4Byte) Len  数组首地址(8Byte) Addr |
|  | struct ClusterType{  int para\_I32;  double para\_DBL;  char para\_Str[1024];  struct NI\_Array{  unsigned long Len;  unsigned long long Addr;  };  }; | DD040000  8D976E1283C0F33F  54657374537472696F6700…  04000000  C03B3F1500000000 | 整型I32（4Byte）,  双精度浮点数DBL（8Byte）  字符串String(1024 Byte)  数组长度Array.Len(U32)  数组地址Array.Addr(U64) |

## 2.2 NOPTR Mode

NOPTR Mode将所有的原始数据信息转换为内存格式，直接传递原始数据信息，既可以在本机使用，也可以进行远程数据传输。针对LabVIEW在测试测量过程的应用，主要支持的数据类型如下表所示：

|  |  |  |
| --- | --- | --- |
| 变量类型 | 变量长度 | 说明 |
| Boolean | 1个字节(8位) | 0表示False，1表示True |
| I8 | 1个字节(8位) | 有符号字节型 |
| U8 | 1个字节(8位) | 无符号字节型 |
| I16 | 2个字节(16位) | 有符号短整型 |
| U16 | 2个字节(16位) | 无符号短整型 |
| I32 | 4个字节(32位) | 有符号整型 |
| U32 | 4个字节(32位) | 无符号整型 |
| I64 | 8个字节(64位) | 有符号长整型 |
| U64 | 8个字节(64位) | 无符号长整型 |
| SGL | 4个字节(32位) | 单精度浮点型 |
| DBL | 8个字节(64位) | 双精度浮点型 |
| Timestamp | 8个字节(64位) | 时间戳。使用1904.1.1 00:00:00 作为基准，DBL的值表示当前时间举例基准时间的差值，单位为秒(s). |
| Array | 根据元素和数据长度变化 | 长度=数据长度(U32)+数组长度(n)\*元素变量长度 |
| Waveform | 根据数据长度变化,可以当作DBL 解析，第一位为T0，第二位为deltaT，以后的数据为 Data。 | Waveform = Array(T0(DBL) + deltaT(DBL) +Data(DBL Array))  长度=数据长度(U32)+数组长度(n)\*元素变量长度 |
| Cluster | 根据组成元素变化 | Cluster 的组成元素按照顺序排列，长度为组成元素长度之和。 |
| String | 1024个字节，当作U8Array解析 | 不包含长度信息，长度小于1024时，空位补0 |
| Path | 1024个字节，当作U8Array解析 | 不包含长度信息，长度小于1024时，空位补0 |

举例：

|  |  |  |  |
| --- | --- | --- | --- |
| LabVIEW数据 | C语言数据 | Binary数据格式 |  |
| Graphical user interface  Description automatically generated | double para\_Array[4] | 04000000  0000000000001440  000000000000F03F  0000000000000040  0000000000000840 | 数组长度(4Byte) Len：4  DBL Array[0] 5.00  DBL Array[1] 1.00  DBL Array[2] 2.00  DBL Array[2] 3.00 |
|  | double para\_wave[6] | 06000000  0000008864BBEB41  000000000000F03F  000000000000F03F  0000000000000040  0000000000000840  0000000000001040 | Array(T0(DBL)+deltaT(DBL)+Data(DBL Array))  数组长度(U32) ：6  TimeStamp:12:00:00PM 12/12/2021  Dt：1.00  Waveform[0]:1.00  Waveform[0]:2.00  Waveform[0]:3.00  Waveform[0]:4.00 |
| Graphical user interface  Description automatically generated | struct ClusterType{  int para\_I32;  double para\_DBL;  char para\_Str[1024];  double para\_Arr[4]  }; | DD040000  8D976E1283C0F33F  54657374537472696F67000…  04000000  0000000000000000  000000000000F03F  0000000000000040  0000000000000840 | 整型I32（4Byte）,  双精度浮点数DBL（8Byte）  字符串String(1024 Byte)  数组长度Array.Len(U32)：4  DBL Array[0] 0.00  DBL Array[1] 1.00  DBL Array[2] 2.00  DBL Array[3] 3.00 |

# API使用说明

## LvData🡪MemoryData

Graphical user interface, diagram

Description automatically generatedDiagram

Description automatically generated

PTR Mode

Graphical user interface, diagram

Description automatically generated

NOPTR Mode

**PTR Mode:** PTR模式为了传输内存数据地址，需要进行内存管理。传输数据前需要申请内存，数据使用后需要释放内存。因此，PTR模式需要和Mem\_ObtainPTRQueue.vi以及Mem\_ReleasePTRQueue,vi配合使用。

Diagram, text

Description automatically generated

**NOPTR Mode:** 直接对数据进行转换，LabVIEW数据连接Payload。

Diagram

Description automatically generated with medium confidence

## MemoryData🡪LvData

A picture containing text, indoor, screenshot

Description automatically generated

A picture containing text, indoor, computer, screenshot

Description automatically generatedA picture containing text, indoor, screenshot

Description automatically generated

PTR Mode

NOPTR Mode

PTR Mode与NOPTR Mode:具有相同的接口，对内存数据进行LabVIEW数据类型转换时，需要连接内存数据，以及LabVIEW数据原型，从而对内存数据进行解析。

Diagram

Description automatically generated Diagram

Description automatically generated with medium confidence

## PTR🡪LvData

提供一套针对数据指针操作的API, 可以通过数组的内存地址和长度信息获取内存中的数据，主要支持的数组类型如下：

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| I8 | I16 | I32 | I64 | U8 | U16 | U32 | U64 | SGL | DBL | Char | String | waveform |

输入数组的长度及首地址，选择数组类型，通过内存地址，数组长度和数组类型从内存中解析数据。

A picture containing text, screenshot, computer

Description automatically generated