ICS ICS 35.240.50
J07

中华人民共和国国家标准

GB/T XXXXX—XXXX

装备制造业 生产过程射频识别技术 第3部分 系统应用接口规范

Equipment Manufacturing Industry -- Radio Frequency Identification for Manufacturing Process -- Part 3 : System Application Interface Specification

（征求意见稿）

XXXX－XX－XX 发布　　XXXX－XX－XX 实施

中华人民共和国国家质量监督检验检疫总局
中国国家标准化管理委员会　发布

前　言

本标准按照 GB/T1.1-2009 给出的规则起草。

请注意本文件的某些内容可能涉及专利。本文件的发布机构不承担识别这些专利的责任。

本标准由中国机械工业联合会提出。

本标准由全国自动化系统与集成标准化技术委员会（SAC/TC159）归口。

本标准起草单位：中国科学院自动化研究所、北京机械工业自动化研究所、威海北洋电气集团股份有限公司、齐齐哈尔轨道交通装备有限责任公司、国家射频识别产品质量监督检验中心、全国信息技术标准化技术委员会

本标准主要起草人：谭杰等

装备制造业 制造过程射频识别技术 第3部分 系统应用接口规范

1 范围

本标准规定了射频识别技术在装备制造业生产过程中对读写设备设备信息查询、参数配置、设备指令、读写命令以及状态报告等五类接口。

本标准适用于装备制造业生产过程中射频识别系统的软件实现与快速部署。

2 规范性引用文件

下列文件对于本文件的应用是必不可少的。凡是注日期的引用文件，仅所注日期的版本适用于本文件。凡是不注日期的引用文件，其最新版本（包括所有的修改单）适用于本文件。

GB/T XXXXX-XXXX 装备制造业 生产过程射频识别技术 第1部分 电子标签技术要求及应用规范

GB/T XXXXX-XXXX 装备制造业 生产过程射频识别技术 第2部分 读写设备技术要求及应用规范

ISO/IEC19762-3：2005 信息技术 自动识别和数据采集（AIDC）技术 词汇 第 3 部分：射频识别（Information technology - Automatic identification and data capture (AIDC) techniques - Harmonized vocabulary - Part 3: Radio frequency identification）

3 术语和定义

下列术语和定义适用于本文件。

3.1 标签 tag

用于物体或物品标识、具有信息存储功能、能接收读写设备的电磁场调制信号，并返回响应信号的数据载体。

3.2 读写设备 reader/writer

一种用于从射频标签获取数据和向射频标签写入数据的电子设备，通常具有冲突仲裁、差错控制、信道编码、信道解码、信源编码、信源译码和交换源端数据等过程。

3.3 标签类型 tag type

读写设备与射频标签之间的空中接口协议类型。

3.4 射频识别 radio frequency identification

在频谱的射频部分，利用电磁耦合或感应耦合，通过各种调制和编码方案，与射频标签进行通信，并读取射频标签信息的技术。

3.5 射频识别系统 radio frequency identification system

一种自动识别和数据采集系统，包含1个或者多个读写设备以及1个或者多个标签，其中，数据传输通过对电磁场载波信号的适当调制实现。

3.6 工作参数 working parameter

读写设备工作时为达到某种性能或者实现某种功能而所需的项目配置。

3.7 通讯接口类型 communication port type

读写设备与管理层数据交流的通讯接口类型。

注：主要有EIA-RS-232C、EIA-RS-485、以太网口等接口类型。

3.8 逻辑地址 logic address

用于表征读写设备的通讯地址。如IP地址。

3.9 人工控制模式—Manual control mode

读写设备工作模式之一，是指读写设备通过物理通讯接口接收到人工控制命令后才进行工作，并通过原接口返回命令执行结果。

3.10 自动读取模式—Auto Read mode

读写设备工作模式之一，是指读写设备作为一种USB人机接口设备使用，读写设备无须通过人工控制命令后才可进行工作。

4 符号和缩略语

下列符号和缩略语适用于本标准。

HID – Human Interface Device

5 数据结构

在接口规范中，采用与编程语言相关的数据类型来描述指令的参数和返回值，主要的数据类型，除了C语言基本数据类型外，自定义的数据类型包括如下：

5.1 读写设备频率数据类型（ReaderFrequency）

读写设备频率数据类型是一个结构体，包含2个字段，其形式如下：

```
struct ReaderFrequency
{
    Integer FrequencyIndex;
    integer FrequencyValue;
};
```

其中FrequencyIndex为读写设备工作频率类型，取值为1（低频段）、2（中频段）、3（超高频段）、4（微波）、5（保留）；FrequencyValue为具体频率值。

5.2 物理接口数据类型（PhysicalInterType）

读写设备的物理接口主要包括EIA-RS-485、以太网口、无线网、USB HID等，其数据类型是一个结构体，包含6个字段，其形式如下：

```
Union PhysicalInterType
{
  RS232_PROT_TYPE = 0x01,
  RS485_PROT_TYPE,
  ETH_PROT_TYPE,
  WIRELESS_PROT_TYPE,
  USB_PROT_TYPE,
  UNDIFINED_TYPE
};
```

其中RS232_PROT_TYPE为EIA-RS-232C接口类型；RS485_PROT_TYPE为EIA-RS-485接口类型；ETH_PROT_TYPE 以太网接口类型；WIRELESS_PROT_TYPE表示无线网络接口类型；USB_PROT_TYPE为USB接口类型；UNDIFINED_TYPE为其它未定义类型。

5.3 EIA接口参数数据类型（EIAInterParameters）

具有EIA-RS-232C或EIA-RS-485接口的读写设备参数数据类型，该数据类型是一个结构体。

```
Struct InterfaceParametersType
{
    long integer Baudrate;
    integer DataBits;
    integer Parity;
    float StopBits;
};
```

其中：

Baudrate参数的取值范围为{600，1200，2400，4800，9600，19200，38400，56000，115200}。

DataBits参数的取值范围为{6，7，8}。

Parity参数：0-无奇偶校验；1-奇校验；2-偶校验。

StopBits参数：0 – 不使用停止位；1 – 使用停止位。

5.4 通讯逻辑地址数据类型（ReaderLogicAddress）

读写设备的通讯逻辑地址主要包括EIA-RS-485、以太网口的通讯地址，其数据类型是一个结构体，包含6个字段，其形式如下：

```
struct ReaderLogicAddress
{
    InterfaceParametersType PortType;
    integer ReaderRS485_Address;
    string ReaderIP_Address;
    string ReaderMAC_Address;
    string ReaderMark_Address;
    string ReaderGateWay;
    string ReaderDNS;
```

};

其中PortType为读写设备的物理端口类型；ReaderRS485_Address为RS_485地址，取值为0-32；ReaderIP_Address为以太网IP地址；ReaderMAC_Address为读写设备MAC地址；ReaderMark_Address为子网掩码；ReaderGateWay为网关；ReaderDNS为域名系统。

当ReaderComPortType为物理端口RS_485类型时，ReaderRS485_Address有效，ReaderIP_Address，ReaderMark_Address,ReaderGateWay,ReaderDNS无效;当PortType为以太网类型时,ReaderIP_Address，ReaderMark_Address，ReaderGateWay，ReaderDNS有效，ReaderRS485_Address无效。

5.5 读写设备信息数据类型（ReaderInfo）

读写设备信息数据类型是一个结构体，包含3个字段，其形式如下：

```
struct ReaderInfo
{
    integer ReaderNumber;
    ReaderFrequency WorkingFreq;
    integer AntennaCount;
};
```

其中ReaderNumber为读写设备的编号；WorkingFreq为读写设备的工作频率；AntennaCount为读写设备的工作天线数量。

5.6 错误报告类型（ErrorReportType）

读写设备返回错误报告的数据类型（DWORD），其定义见表2。

表1 错误报告类型

0x01	ERROR_INVALID_COMMAND	无效指令
0x02	ERROR_COMMAND_INTERRUPT	当前指令无法执行
0x03	ERROR_READER_UNKNOWN	找不到指定读写设备
0x04	ERROR_ANTENNA_NUMBER	天线数量错误
0x05	ERROR_PARAMETER_INPUT	指令参数错误
0x06	ERROR_READER_INVALIDATE	读写设备工作不正常
0x07	ERROR_NON_SPECIFIED_TYPE_TAG	读取到非指定类型标签
0x08	ERROR_TAG_DAMAGED_OR_MOVEDAWAY	标签有损坏内存或者未读取完毕标签被移走
0x09-0xffffffff	保留	保留

6 接口说明

6.1 信息查询接口 -- 读写设备信息查询

6.1.1 目的

查询指定读写设备的自有信息。适用于以人工控制模式和自动读取模式进行工作的读写设备。

6.1.2 说明

实现需求：必须实现

C语言语法函数描述如下：

DWORD *GetReaderDescription*(Integer ConfigID, struct * m_pDevAttribute)

参数：

ConfigID,指定的读写设备编号，如果取为-1，则指所有读写设备。

m_ pDevAttribute，返回读写设备信息数据类型结构体指针。

返回值：

a)0， 操作成功

b)非 0， 操作失败，返回错误报告类型

6.2 信息查询接口--读写设备逻辑地址信息查询

6.2.1 目的

查询指定读写设备的逻辑地址。适用于以人工控制模式和自动读取模式进行工作的读写设备。

6.2.2 说明

实现需求：必须实现

C语言语法函数描述如下：

DWORD *GetLogicAddress* (Integer ConfigID, struct * m_ pDevLogicAddr)

参数：

ConfigID,指定的读写设备编号，如果取为-1，则指所有读写设备。

m_ pDevLogicAddr，返回通讯逻辑地址数据类型指针。

返回值：

a)0， 操作成功

b)非 0， 操作失败，返回错误报告类型

6.3 信息查询接口--读写设备工作天线查询

6.3.1 目的

查询指定读写设备的工作天线状态。适用于以人工控制模式进行工作的读写设备。

6.3.2 说明

实现需求：必须实现

C语言语法函数描述如下：

DWORD *GetAntennaCount* (Integer ConfigID, DWORD * m_ pAntennaCount)

参数：

ConfigID,指定的读写设备编号。

m_ pAntennaCount，返回读写设备工作天线数量。其数值二进制比特位代表端口天线可用性。例：Bit 0等于1表示端口1天线可用，等于0表示端口1天线不可用；Bit 6等于1表示端口7天线可用，等于0表示端口7天线不可用。

返回值：

a)0， 操作成功
b)非 0， 操作失败，返回错误报告类型

6.4 参数设置接口--设置读写设备当前的读取命令间隔时间

6.4.1 目的

设置读写设备新的当前命令间隔时间。适用于以人工控制模式进行工作的读写设备。

6.4.2 说明

实现需求：必须实现
C语言语法函数描述如下：
DWORD *SetCommandInterval* (Integer ConfigID,integer Period_t)
参数：
ConfigID,指定的读写设备编号，如果取为-1，则指所有读写设备。
Period_t,读写设备当前的读取命令间隔时间，单位是毫秒。
返回值：
a)0， 操作成功
b)非 0， 操作失败，返回错误报告类型

6.5 参数设置接口--设置读写设备单次可读取标签数

6.5.1 目的

设置超高频工作频段的读写设备最大可以识别的标签数量。适用于以人工控制模式进行工作的读写设备。

6.5.2 说明

实现需求：必须实现
C语言语法函数描述如下：
DWORD *SetMaxReadCountLimit* (Integer ConfigID,integer Max_num)
参数：
ConfigID,指定的读写设备编号，如果取为-1，则指所有读写设备。
Max_num, 读写设备单次可识别标签数。
返回值：
a)0， 操作成功
b)非 0， 操作失败，返回错误报告类型

6.6 参数设置接口--设置读写设备逻辑地址

6.6.1 目的

设置读写设备逻辑通信地址。适用于以人工控制模式和自动读取模式进行工作的读写设备。

6.6.2 说明

实现需求：必须实现
C语言语法函数描述如下：
DWORD *SetMaxReadCountLimit* (Integer ConfigID,stuct ReaderLogicAddress)
参数：ConfigID,指定的读写设备编号。
ReaderLogicAddress, 读写逻辑通信地址。
返回值：
a)0， 操作成功
b)非 0， 操作失败，返回错误报告类型

6.7 参数设置接口—设置天线收发功率

6.7.1 目的

设置读写设备指定工作天线的收发功率。适用于以人工控制模式进行工作的读写设备。

6.7.2 说明

实现需求：必须实现
C语言语法函数描述如下：
DWORD *SetAntennaPower* (Integer AntennaNumber, integer Power_num)
参数：
AntennaNumber,指定工作天线编号，-1为当前所有工作天线。
Power_num，天线收发功率值。
返回值：
a)0， 操作成功
b)非 0， 操作失败，返回错误报告类型

6.8 设备指令接口 – 重启读写设备

6.8.1 目的

重新启动指定读写设备。适用于以人工控制模式进行工作的读写设备。

6.8.2 说明

实现需求：可选
C语言语法函数描述如下：
DWORD *RebootReader*(Integer ConfigID)
参数：ConfigID,指定的读写设备编号，如果取为-1，则指所有读写设备。
返回值：
a)0， 操作成功
b)非 0， 操作失败，返回错误报告类型

6.9 设备指令接口 – 打开读写设备

6.9.1 目的

给指定读写设备上电。适用于以人工控制模式进行工作的读写设备。

6.9.2 说明

实现需求：可选

C语言语法函数描述如下：

DWORD *OpenReader*(Integer ConfigID)

参数：ConfigID,指定的读写设备编号，如果取为-1，则指所有读写设备。

返回值：

a)0， 操作成功

b)非 0， 操作失败，返回错误报告类型

6.10 设备指令接口 – 初始化读写设备

6.10.1 目的

设置指定的读写设备的工作参数。适用于以人工控制模式进行工作的读写设备。

6.10.2 说明

实现需求：可选

C语言语法函数描述如下：

DWORD *ReaderInit*(Integer ConfigID ,struct *m_pReaderConf)

参数：ConfigID,指定的读写设备编号。

m_pReaderConf，读写设备配置信息结构体指针，包含读写设备地址，工作频率等信息，需要根据读写设备功能设置。

返回值：

a)0， 操作成功

b)非 0， 操作失败，返回错误报告类型

6.11 设备指令接口 – 关闭读写设备

6.11.1 目的

关闭指定的读写设备，使之停止工作。适用于以人工控制模式进行工作的读写设备。

6.11.2 说明

实现需求：可选

C语言语法函数描述如下：

DWORD *CloseReader*(Integer ConfigID)

参数：ConfigID,指定的读写设备编号。如果取为-1，则指所有读写设备。

返回值：

a)0， 操作成功

b)非 0， 操作失败，返回错误报告类型

6.12 读写命令接口 – 读取标签

6.12.1 目的

读取指定位置、指定长度的标签数据。适用于以人工控制模式进行工作的读写设备。

6.12.2 说明

实现需求：必选

C语言语法函数描述如下：

DWORD ***ReadTag*** (BYTE m_Type,DWORD m_StartBlock,DWORD m_PreTotalNum, DWORD* m_ReadBlockNum, BYTE* m_pBuffer, BYTE* m_TagUID = NULL)

参数：

m_Type，0 对所有标签进行操作，1 对指定标签进行操作。

m_Startblock，操作的标签内存起始块号。

m_PreTotalNum，预读取总块数。

m_ReadBlockNum，实际读取到的块数。

m_pBuffer，读取数据首指针。

m_TagUID，标签标识符，只有m_Type为1时，此参数才有意义。

返回值：

a)0， 操作成功

b)非 0， 操作失败，返回错误报告类型

6.13 读写命令接口– 写标签数据

6.13.1 目的

向标签指定位置写入指定长度的数据。适用于以人工控制模式进行工作的读写设备。

6.13.2 说明

实现需求：必选

C语言语法函数描述如下：

DWORD ***WriteTag***(BYTE m_Type,DWORD m_StartBlock,DWORD m_TotalNum, BYTE* m_pBuffer, BYTE* m_TagUID = NULL))

参数：

m_Type，0 对所有标签进行操作，1 对指定标签进行操作。

m_Startblock，操作的标签内存起始块号。

m_TotalNum，写入总块数。

m_pBuffer，写入数据首指针。

m_TagUID，标签标识符，只有m_Type为1时，此参数才有意义。

返回值：

a)0， 操作成功

b)非 0， 操作失败，返回错误报告类型

6.14 读写命令接口 – 触发接收数据

6.14.1 目的

外部读写设备接入时，触发此函数等待接收标签数据。适用于以自动读取模式进行工作的读写设备。

6.14.2 说明

实现需求：必选

C语言语法函数描述如下：

DWORD *AutoTag* (BYTE m_Type,BYTE* m_pBuffer, BYTE* m_TagUID = NULL)

参数：

m_Type，0 对所有标签进行操作，1 对指定标签进行操作。

m_pBuffer，读取数据首指针。

m_TagUID，标签标识符，只有m_Type为1时，此参数才有意义。

返回值：

a)0， 操作成功

b)非 0， 操作失败，返回错误报告类型

6.15 状态报告接口--读写设备工作状态报告

6.15.1 目的

当前指定读写设备的状态报告。适用于以人工控制模式和自动读取模式进行工作的读写设备。

6.15.2 说明

实现需求：可选

C语言语法函数描述如下：

DWORD *SelfTesting*(Integer ConfigID, bool bOpen, char* TestingReport)

参数：ConfigID,指定的读写设备编号。如果取为-1，则指所有读写设备。

TestingReport,读写设备自检报告

BOpen, 参数为true时，如果读写设备的自检检测功能已开启，则返回当前工作状态；如果读写设备的自检检测功能已关闭，则启动此功能后，返回当前工作状态。参数为false时，关闭读写设备的自检检测功能。

返回值：

a)0， 操作成功，返回自检检测信息

b)非0， 操作失败，返回错误报告类型