

ICS 71.040.40

CCS G 04

T/GAIA

广东省分析测试协会团体标准

T/GAIA XXXX—XXXX

表面化学分析 扫描探针显微镜数据存储格式

Surface chemical analysis — Data storage format for scanning probe microscopy

(征求意见稿)

(本草案完成时间: 2023 年 9 月 30 日)

XXXX - XX - XX 发布

XXXX - XX - XX 实施

广东省分析测试协会 发布

目 次

前 言.....	III
引 言.....	IV
1 范围.....	5
2 规范性引用文件.....	5
3 术语和定义.....	5
4 格式概述.....	6
5 格式描述.....	6
5.1 总体结构.....	6
5.2 数据头.....	7
5.2.1 数据头的结构及特点.....	7
5.2.2 数据文件头.....	7
5.2.3 数据信息头.....	7
5.2.4 默认彩色表.....	9
5.3 数据阵列.....	10
5.4 参数表.....	10
5.4.1 概述.....	10
5.4.2 参数表表头.....	10
5.4.3 基本参数表.....	11
5.4.4 扩展参数表.....	17
5.4.5 关联参数表.....	21
5.4.6 特殊格式参数表.....	24
附录 A 格式示例.....	27
A.1 示例-1: 数据文件头.....	27
A.2 示例-2: 数据信息头.....	27
A.3 示例-3: 颜色定义表.....	27
A.4 示例-4: 数据阵列.....	28
A.5 示例-5: 参数表头.....	28
A.6 示例-6: 通用参数表头.....	29
A.7 示例-7: 普测量显示信息表头.....	29
A.8 示例-8: 关联参数表头.....	29
A.9 示例-9: 实验参数信息.....	30
A.10 示例-10: 纵坐标信息.....	30
A.11 示例-11: 测量位置.....	30
A.12 示例-12: 图像显示信息.....	31
A.13 示例-13: 图像显示配色表.....	31
A.14 示例-14: 数据来源.....	32
A.15 示例-15: 处理参数.....	32
A.16 示例-16: 插件信息.....	32

参 考 文 献..... 34

前 言

本文件按照GB/T 1.1—2020《标准化工作导则 第1部分：标准化文件的结构和起草规则》的规定起草。

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。

本文件由广东省分析测试协会提出。

本文件由广东省分析测试协会归口。

本文件起草单位：中山大学物理学院

广州中源仪器技术有限公司

中山大学分析测试中心

暨南大学理工学院

华南理工大学分析测试中心

本文件主要起草人：丁喜冬（中山大学物理学院）

肖章武（广州中源仪器技术有限公司）

赵亮兵（广州中源仪器技术有限公司）

陈 建（中山大学分析测试中心）

龚 力（中山大学分析测试中心）

谢伟广（暨南大学理工学院）

刘金超（华南理工大学分析测试中心）

引 言

在表面分析测试领域，许多商用的扫描探针显微镜（SPM）在各种不同的环境下使用。SPM的控制、数据的采集和处理都可以由带有存储器的计算机来实现，所有的数据及参数都记录在数字化的文件中。由于所用硬件平台不同，各制造商所生产的SPM大都使用专用的数据格式。这些数据格式大多互不兼容，给数据的存储、处理、传输、交换、共享等带来了不便。基于文本的SPM数据格式，虽然能够解决数据的传输、交换、共享等方面的问题，但存储效率不高且不利于数据的快速处理。各制造商所使用的SPM数据格式通常采用二进制存储，更加适合用于数据的存储和分析处理，但存在兼容性、扩展性等方面的问题，不便于数据的传输、交换和共享。因此，需要一种用于SPM数据的存储和共享的标准格式，主要应用于SPM中的实时数据存储、高效分析处理和数据共享等计算处理领域。

表面化学分析

扫描探针显微镜数据存储格式

1 范围

本文件规定了在计算机上存储扫描探针显微镜（SPM）数据的一种标准格式。该格式以二进制和文本相混合的形式进行编码，将SPM采集的数据、参数和关联信息进行紧凑地存储，适合于对海量数据的存储和共享。该格式的数据结构支持随机读写，适合于对SPM数据进行高效的分析处理和读写。

本格式是为SPM的数据存储而设计的，包括扫描隧道显微镜（STM）、原子力显微镜（AFM）、静电力学显微镜（EFM）、磁力显微镜（MFM）等，以及其它采用尖锐针尖在样品表面扫描的相关SPM表面分析方法。本格式涵盖了SPM的单通道成像、多通道成像、单点谱和多点谱的数据。

2 规范性引用文件

下列文件中的内容通过文中的规范性引用而构成本文件必不可少的条款。其中，注日期的引用文件，仅该日期对应的版本适用于本文件；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 36052-2018 表面化学分析 扫描探针显微镜数据传送格式

本文件的主体部分包括格式描述和相关约定。

附录A给出带有注释的格式示例，附录B给出数据格式示例。

3 术语和定义

本文件没有需要界定的术语和定义。

ISO 和 IEC 使用的标准化术语数据库保存在下列网址中：

——ISO 在线浏览平台：<https://www.iso.org/obp>

——IEC 电工百科：<http://www.electropedia.org>

下列缩略语适用于本文件。

SPM：扫描探针显微镜（Scanning Probe Microscope）

AFM：原子力显微镜（Atomic Force Microscope）

STM: 扫描隧道显微镜 (Scanning Tunneling Microscope)

EFM: 静电力显微镜 (Electrostatic Force Microscope)

MFM: 磁力显微镜 (Magnetic Force Microscope)

BMP: 位图文件 (Bitmap)

4 格式概述

这里先对该格式进行概述，以便在上下文中理解第5章中给出的格式描述。

SPM 数据存储格式的基本构想是该格式用普通计算机可读、可写、可共享；为了紧凑存储数据、减小存储空间、利于海量存储，该格式中的数据主要采用基于二进制的存储方式；为了实现对数据的高效分析处理，该格式的数据结构应支持数据的随机读写。

为了有足够的灵活性以适应未来SPM衍生仪器的扩展、有足够的通用性以适用各种被测物理量，本格式中的数据参数部分兼容现有的表面化学分析的国际标准ISO/DIS 28600:2011 Surface chemical analysis – Data transfer format for scanning probe microscopy，或国家标准GB/T 36052-2018 表面化学分析 – 扫描探针显微镜数据传送格式。

5 格式描述

5.1 总体结构

SPM 数据存储格式采用与标准的 Windows 的 BMP (Bitmap) 文件格式相类似的文件结构。为方便用户理解格式与编写数据处理程序，本格式采用与 Windows 的 BMP 文件格式相类似的描述方法。

SPM 数据存储格式的总体结构包括三个部分，分别是：数据头、数据阵列、参数表。

SPM 数据存储格式中的前两个部分相当于标准的 Windows 的 BMP 文件格式部分，采用二进制存储。其中，SPM 数据存储格式中的数据头（第一部分）类似于 Windows 的 BMP 文件格式中的文件头，数据阵列（第二部分）则类似于 Windows 的 BMP 文件格式中的位图的字节阵列。

SPM 数据存储格式中的参数表（第三部分）是添加在 Windows 的 BMP 文件尾部的参数描述部分。参数表包括 SPM 数据获取、存储、显示和分析处理的参数，用于对 SPM 的图像数据进行结构化的描述，主要采用二进制存储。

SPM 数据存储格式建议保存为*.spm 文件，即以.spm 为文件扩展名。

SPM数据存储格式 = 数据头 + 数据阵列 + 参数表

5.2 数据头

5.2.1 数据头的结构及特点

SPM数据存储格式中的数据头以Windows的BMP文件格式中的文件头为基础进行定义。为了能够同时涵盖单通道与多通道图像数据、单点与多点的谱数据，并能够在同一文件中存储SPM的参数和关联信息，以及提供适当的兼容性和扩展性，对原BMP文件格式中的数据头的结构和定义进行了修改。

数据头包括三个部分，即：数据文件头、数据信息头和默认彩色表等。这三个部分大致对应于BMP文件格式中的前三个部分（文件头），即：位图文件头(bitmap-file header)、位图信息头(bitmap-information header)和彩色表(color table)。

BMP（英文Bitmap的简写）是Windows操作系统中的标准图像文件格式。BMP是一种与硬件设备无关的图像文件格式，能够支持1, 4, 8, 16, 24, 32位的图像存储，并且支持非压缩和压缩这两种存储方式，数据是以Little Endian依次存储。本格式中的数据头定义继承了BMP文件格式的这些特点。

数据头 = 数据文件头 + 数据信息头 + 默认彩色表

5.2.2 数据文件头

数据文件头包含文件头类型、文件大小、存放位置等信息，共占14个字节。数据文件头包括4个属性，各个属性规定如下：

(*1*) bitmap file identifier=字符 (*2个字节的内容用来识别位图数据的类型*)；

数据文件类型标识符（对应BMP的文件类型标识符），偏移量0000h。0x4d42='BM'，固定不变，表示本格式类似于Windows支持的BMP文件格式。

(*2*) file Size=整数 (*4个字节的内容用来表示数据阵列部分的字节数*)

4字节表示的数据部分的大小，偏移量0002h。

(*3*) data type identifier=字符 (*4个字节的内容用来识别SPM数据的类型*)；

4字节表示SPM图像数据的类型标识符，即表示是否SPM的图像数据。偏移量0006h。

① 0x0000：表示是SPM单通道成像的图像数据；该数据格式与Windows支持的BMP格式兼容。

② 'MPMC'：表示是SPM的多通道图像数据；数据格式与BMP格式不兼容。

③ 'SPMC'：表示是SPM的多通道谱数据；数据格式与BMP格式不兼容。

④ 'USPM'：表示是SPM的图像或谱的数据；数据格式保留给用户自定义。

(*4*) Bitmap Data Offset=整数 (*4个字节的内容用来表示从文件开始到实际的图像数据之间的字节的偏移量，即表示数据头部分的字节数*)；

5.2.3 数据信息头

数据信息头包含位图数据存放的信息，包括位图信息头长度、位图宽度和高度、位图的位面数、每个像素的位数、压缩说明、位图数据的大小、比例尺（水平和垂直的分辨率）、颜色索引数、重要颜色索引数等共11个属性，各个属性规定如下：

(*5*) bitmap head Size=整数 (*4个字节表示数据头中的数据信息头的长度*)

用4字节表示的数据信息头的长度的字节数，偏移量000Eh。

(*6*) bitmap width=整数 (*4个字节表示位图宽度，以像素为单位*)

用4字节表示的位图宽度，偏移量0012h。对于SPM的图像数据，表示每行数据个数。对于SPM的谱数据，表示每个谱的测量点数。

(*7*) bitmap height=整数 (*4个字节表示位图高度，以像素为单位*)

用4字节表示的位图高度，偏移量0016h。对于SPM的图像数据，位图高度 = 图像行数 * 图像个数。对于SPM的谱数据，位图高度 = 控制变量数量 + 待测变量数量 * 测量遍数 * 位置数量。

为使数据从上往下显示，Bitmap height 需要采用负整数，其值通常等于位图高度的相反数。

(*8*) bitmap planes=整数 (*2个字节表示位图的位面数*)

用2字节表示位图的位面数，偏移量001Ah。对于SPM图像数据或谱数据，bitmap planes=0x0001。

(*9*) bit count=整数 (*2个字节表示每个像素数据的位数*)

用2字节表示位图的每个像素数据位数，偏移量001Ch。其值通常可为1、4、8、16、24、32。

当数据文件头中的SPM数据的类型data type identifier =0x0000时，默认bit count=24。

当数据文件头中的SPM数据的类型data type identifier = 'MPMC' 或 'SPMC' 时，默认bit count=32。

当bit count=24，每个像素的数据格式如下：低16位存储无符号整数B，高8位存储0x00；该像素的SPM数据的值 = B。

当bit count=32，每个像素的数据格式如下：低24位存储无符号整数，其值为基数B；高8位存储有符号整数；其值为以2为底的指数N；该像素的SPM数据的值 = $B * 2^N$ 。指数N通常为0。如过N为0不足以无损地表示出所有数据，则所有数据的N值应尽量统一，其绝对值应尽量采用0、4、8等数值。

(*10*) data compression=整数 (*4个字节表示数据的压缩方式*)

用4字节表示说明SPM的图像数据的压缩的类型（每个像素的数据位数），偏移量001Eh。通常选择=0(相当于BMP图像格式中的BI_RGB，表示没有压缩)。压缩说明：=0(不压缩)，1(RLE 8，8位RLE压缩)，2(RLE 4，4位RLE压缩)，3(Bitfields，位域存放)。RLE简单地说是采用像素数+像素值的方式进行压缩。

(*11*) bitmap data Size=整数 (*4个字节用来表示位图数据的大小*)

用4字节表示位图数据的大小，偏移量0022h。该数必须是4的倍数，数值上应等于：一行所占的字节数×位图高度。

(*12*) bitmap X scale=整数 (*4个字节表示位图的水平比例尺或分辨率*)

用4字节表示位图的水平(X方向)的分辨率或谱图正向测量点数，偏移量0026h。表示位图数据的水平分辨率时，单位是像素/毫米。

(*13*) bitmap Y scale=整数 (*4个字节表示位图的垂直的比例尺或分辨率*)

用4字节表示位图的垂直(Y方向)的分辨率或谱图反向测量点数, 偏移量002Ah。表示位图数据的垂直分辨率时, 单位是像素/毫米。

(*14*) color used=整数 (*4个字节表示SPM图像数据使用的颜色索引数*)

用4字节表示SPM图像数据在显示时实际使用的颜色索引数(设为0的话, 则说明使用彩色表中的所有颜色)。对于典型的SPM图像数据(如data type identifier='MPMC'), 这4字节表示默认的成像通道(通常为第0号或1号成像通道)的SPM图像数据在显示时实际使用的彩色表的颜色数目(通常不为0或1)。对于SPM图像数据中与BMP兼容的位图数据(如data type identifier=0x0000时), 这4字节的定义与BMP格式中的定义相同, 即表示位图实际使用的彩色表中的颜色索引数(为0说明使用所有调色板项)。对于单点谱数据(如data type identifier='SPMC'), 这用4字节表示所有谱线在数据显示时实际使用的颜色索引数(通常每条谱线只使用1个颜色)。

(*15*) color Important =整数 (*4个字节表示重要的颜色索引数*)

用4字节表示SPM图像数据在显示时重要数据段使用的颜色数目(如果是0, 表示都重要)。对于SPM的多通道图像数据(如data type identifier='MPMC'), 通常将SPM图像数据按数值大小分为若干个数据段, 其中有一个重要数据段; “重要颜色数目”用于指定该重要数据段使用的颜色数目(0表示都重要, 即重要数据段和其它数据段所分配的颜色索引数尽量均匀分布)。对于BMP兼容的位图数据(如data type identifier=0x0000时), 这4字节的定义与BMP格式中的定义相同。对于谱数据(在data type identifier='SPMC'), 由于每条谱线通常使用不同的颜色值, color Important的值通常为0(表示都重要)。

5.2.4 默认彩色表

默认彩色表表示SPM图像数据在显示时默认使用的颜色值。对于本格式标准定义的SPM数据, 均可能使用或不使用默认彩色表。对于SPM的多通道图像数据(如data type identifier='MPMC'), 这用4字节表示默认成像通道(通常为第0号或1号成像通道)的SPM图像在显示时默认使用的彩色表。对于SPM图像数据中与BMP兼容的位图数据(如data type identifier=0x0000时), 这4字节的定义与BMP格式中相应字段的定义相同。对于单点谱数据(如data type identifier='SPMC'), 也使用默认彩色表。

(*16*) color index={RGB颜色值} (*每个索引项用4个字节表示RGB颜色值*)

彩色表中的每个颜色值均占用4个字节的存储空间。像素颜色的颜色可用类似BMP格式中的RGBQUAD结构来定义。RGBQUAD结构描述由R、G、B相对强度组成的颜色。颜色表中的颜色索引号默认按照0、1、2、3、...的顺序依次排列。

5.3 数据阵列

紧跟在数据头之后的是位图数据的字节阵列。图象的每一扫描行由表示图象像素的连续的字节组成，同一通道的数据连续存储，每一行的字节数取决于位图数据的颜色数目和用像素表示的图象宽度。扫描行是由顶向下存储的，这就是说，阵列中的第一个字节表示位图左上角的像素，而最后一个字节表示位图右下角的像素（即采用正向DIB，扫描行是由顶向下存储的）。同时，每一扫描行的字节数必需是4的整倍数，也就是DWORD对齐的。

5.4 参数表

5.4.1 概述

紧跟在数据阵列之后的是参数表。参数表包括SPM图像或谱图等数据的获取、存储、显示和分析处理等所需的参数。对于不同类型的SPM图像数据（data type identifier= 0x0000、'SPMC'、'SPMC'或'USPM'），参数表的内容和格式可以不同。本格式文本重点描述典型的SPM图像数据（data type identifier='SPMC'或'SPMC'）中的、默认参数表的格式定义。在参数表的最后部分，也简要描述与BMP格式兼容的SPM图像数据（data type identifier= 0x0000）的特殊参数表的格式定义。

本格式描述的默认参数表主要包括以下3个方面的内容：SPM数据获取的基本参数和扩展参数、以及SPM数据存储、处理、分析所需的关联参数。SPM数据获取的基本参数包含《GB/T 36052-2018 表面化学分析-扫描探针显微镜数据传送格式》中的128个“文件头项目”组成的基本参数表。基本参数表的描述采用与该国家标准《GB/T 36052-2018 表面化学分析-扫描探针显微镜数据传送格式》相同的术语、词汇标准和元语言。

参数表采用结构化的描述方法，包含参数表头、基本参数表、扩展参数表和关联参数表。参数表数据项也采用结构化的数据存储方式。参数表数据项的数据类型可以是计算机程序语言支持的基本数据类型，包含整型、浮点、字符串、字节串等，也可以包含由以上基本数据类型组成的结构数据类型。

参数表 = 参数表头 + 基本参数表 + 扩展参数表 + 关联参数表

5.4.2 参数表表头

参数表头包括以下内容：

(*1*) Total Parameters Identifier=字符（*4个字节的内容用来识别参数表的类型*）；
参数表标识符='PARS'，暂固定不变并保留升级。

(*2*) Total Parameters Size=整数（*4个字节的内容用来表示整个参数表的字节数*）

(*3*) Total Parameters Number=整数（*4个字节的内容用来表示整个参数表数据的个数*）

(*4*) Max Data Value=整数（*4个字节的内容用来表示位图数据阵列中数据的最大值*）

数据格式与数据阵列中每像素数据的格式一致。

(*5*) Base Parameters Offset=整数 (*4个字节的内容用来表示从参数表头开始到基本参数表之间的字节的偏移量*)

(*6*) Extend Parameters Offset=整数 (*4个字节的内容用来表示从基本参数表开始到扩展参数表开始之间的字节的偏移量*)

(*7*) RelationParameters Offset=整数 (*4个字节的内容用来表示从扩展参数表开始到关联参数表开始之间的字节的偏移量*)

5.4.3 基本参数表

基本参数表采用结构化的描述方法，包含基本参数表头、基本参数表项。

5.4.3.1 基本参数表表头

(*1*) Base Parameters Identifier=字符 (*4个字节的内容用来识别参数表的类型*)；
基本参数表标识符='BASE'，暂固定不变并保留升级。

(*2*) Base Parameters Size=整数 (*4个字节的内容用来表示基本参数表数据的字节数*)

(*3*) Base Parameters Number=整数 (*4个字节的内容用来表示基本参数表数据的项数*)

基本参数表数据的项数=0x0100，暂固定不变并保留升级。

5.4.3.2 基本参数表数据

(*1*) format identifier = 'ISO/TC201SPM Data Transfer Format'；(* 版本号*)；

(*2*) label line = 'general information'，EOL；(* 基本信息*)；

(*3*) institution identifier = 字符串；(* 对数据负责的机构,例如 NIMS*)；

(*4*) instrument model identifier = 字符串；(* 数据采集设备名称*)；

对于商用SPM系统，应该标明制造商名称和机器研发代码以识别使用的仪器；对于自制 SPM，应标明自制或机器研发代号以用于识别。

(*5*) operator identifier = 字符串；(* 标明操作人员,例如:Fujita*)；

(*6*) experiment identifier = 字符串；(* 标明实验名称*)；

通常来说,为了原始数据的转移,由原始文件名做实验名称是合适的。

(*7*) comment line = 字符串；(*SPM数据概要*)；

(*8*) experiment mode = ('MAP_SC'|'MAP_MC'|'SPEC_SC'|'SPEC_MC'|)，EOL；(*SPM

测量方式:MAP_SC—单通道规则二维数据;MAP_MC—多通道规则或不规则二维数据;SPEC_SC—单点单通道谱图;SPEC_MC—单点多通道谱图*)；

(*9*) year in full= 整数；(* 公历年,例如 '2008' *)；

(*10*) month = 整数；(* 月*)；

(*11*) day of month = 整数；(* 日*)；

(*12*) hours = 整数; (*24小时制*);

(*13*) minutes = 整数; (*分*);

(*14*) seconds = 整数; (*秒*);

(*15*) Number of hours in advance of Greenwich Mean Time = 整数; (*格林威治时间, 上述七项数据采集的日期和时间是必不可少的, 以采集最后一个数据点的时间记录, 如果前六项中某些数值是未知的, 用-1记录*);

(*16*) label line = 'scan information', EOL; (*扫描信息*);

(*17*) scan mode = ('REGULAR MAPPING' | 'IRREGULAR MAPPING'), EOL; (*X-Y平面扫描类型: REGULAR MAPPING—规则扫描: 在X-Y平面二维光栅扫描, 探针沿快扫轴规则运动, X、Y坐标值省略不计; IRREGULAR MAPPING—不规则扫描: 在X-Y平面二维向量扫描, 探针不规则运动, 每个点的X、Y坐标应该加入数组中*);

(*18*) scanning system = ('open-loop scanner' | 'XY closed-loop scanner' | 'XYZ closed-loop scanner'), EOL; (*说明扫描系统类型的字符串*);

通常使用基于压电部件的位置扫描器对探针进行定位。没有闭环控制的扫描系统叫做开环扫描系统; 带有位置传感器和反馈控制的系统叫做闭环扫描系统。

(*19*) scanner type = ('sample XYZ scan' (样品在XYZ方向扫描) | 'probe XYZ scan' (探针在XYZ方向扫描) | 'sample XY scan and probe Z scan' (样品在XY方向扫描的同时探针在Z方向扫描) | 'sample Z scan and probe XY scan' (样品在Z方向扫描的同时探针在XY方向扫描)), EOL; (*扫描类型*);

(*20*) fast scan axis = ('X' | 'Y'), EOL; (*获取图像每一行数据的扫描轴, 即快扫轴*);

(*21*) fast scan direction = text line; (*快扫轴扫描方向, 例如: 'left to right' (从左到右), 'right to left' (从右向左), 'bottom to top' (从下向上) 或 'top to bottom' (从上向下)*); 一个图像仅对应一个快扫方向, 在两个方向上合并的图像需在(*7*)行做注释;

(*22*) slow scan axis = ('X' | 'Y'), EOL; (*与快扫轴相反的轴, 即慢扫轴*);

(*23*) slow scan direction = 字符串; (*慢扫轴方向, 例如: 'bottom to top' (从下到上), 'top to bottom' (从上到下), 'left to right' (从左到右), 'right to Left' (从右到左)*);

(*24*) Number of discrete x coordinates available in full map = 整数; (*图像X轴像素点个数, 例如 '256' '512' *);

(*25*) Number of discrete y coordinates available in full map = 整数; (*图像Y轴像素点个数*);

(*26*) physical unit of x axis = 单位; (*X轴物理单位, 例如: 'nm' 'um' 'V' *);

(*27*) physical unit of y axis = 单位; (*Y轴物理单位, 例如: 'nm' 'um' 'V' *);

若扫描器已经校准,可以用长度单位,如nm,若扫描器没有校准,在压电陶瓷上施加的适当电压作为单位。

- (*28*) range of x axis = 实数; (* 表明 X 轴扫描宽度的实数*);
- (*29*) range of y axis = 实数; (* 表明 Y 轴扫描宽度的实数*);
- (*30*) physical unit of X Offset = 单位; (* X 轴偏移量单位,例如: ‘nm’ ‘um’ ‘V’ *);
- (*31*) physical unit of Y Offset = 单位; (* Y 轴偏移量单位,例如: ‘nm’ ‘um’ ‘V’ *);
- (*32*) Offset of X axis = 实数; (* 一个实数,表明 X 轴方向原点相对中点偏移量*);
- (*33*) Offset of Y axis = 实数; (* 一个实数,表明 Y 轴方向原点相对中点偏移量*);
- (*34*) rotation angle = 实数; (* 沿X 轴逆时针旋转至样品的角度*);
- (*35*) physical unit of scan speed = 单位; (* 沿快扫轴扫描速度单位,例如: ‘nm/s’ *);
- (*36*) scan speed = 实数; (* 扫描速度*);
- (*37*) physical unit of scan rate = 单位; (* 扫描频率单位,例如: ‘Hz’ *);
- (*38*) scan rate = 实数; (* 沿快扫轴的扫描频率*);
- (*39*) SPM technique = 文本行; (*SPM 测量技术, 例如:
 BEEM = ballistic electron beam microscopy, 弹道电子束显微镜
 CPAFM = conductive probe atomic force microscopy, 导电原子力显微镜
 contact mode AFM = contact mode atomic force microscopy, 原子力显微镜接触模式
 DFM = dynamic force microscopy, 动态力显微镜
 EFM = electrostatic force microscopy, 静电力显微镜
 FMM = force modulation microscopy, 力调制显微镜
 FFM = friction force microscopy, 摩擦力显微镜
 FM-AFM = frequency modulation atomic force microscopy, 频率调制模式原子力显微镜
 IC-AFM =intermittent contact mode atomic forc emicroscopy, 间歇接触模式原子力显微镜
 NC-AFM = non-contact mode atomic force microscopy, 非接触模式原子力显微镜
 KFM = Kelvin force Microscopy, 开尔文探针力显微镜
 MFM =magnetic force microscopy, 磁力显微镜
 LFM =lateral force microscopy, 横向力显微镜
 SCM = scanning capacitance microscopy, 扫描电容显微镜
 SSRM = scanning spreading resistance microscopy, 扫描扩散电阻显微镜
 STM = scanning tunneling microscopy, 扫描隧道显微镜
 SThM = scanning thermal microscopy, 扫描热显微镜
 NSOM = near-field scanning optical microscopy, 近场扫描光学显微镜
 SNOM = scanning near-field optical microscopy, 扫描近场光学显微镜等*);

(*40*) bias voltage contact = (‘sample biased’ | ‘tip biased’), EOL; (* 指明加偏压的电极: sample biased—样品加偏压, tipbiased—探针加偏压*);

(*41*) bias voltage = 实数; (* 偏压大小, 单位是 V*);

(*42*) Number of setitems = 整数; (*SPM 固定参数的个数*);

(*43*) set parameter(s) = 字符串; (* 标识每个固定参数, 以逗号分开, 例如: free-oscillation amplitude (自由振幅), drive frequency (驱动频率)*);

(*44*) unit(s) of set parameter(s) = 单位 ;(* 各固定参数的单位, 以逗号分开, 例如: ‘nm’ , ‘Hz’ *);

(*45*) value of set parameter = 实数; (* 各固定参数的值, 以逗号分开, 例如: ‘100, 100000’ *);

(*46*) calibration comment for set parameter = 字符串; (* 各固定参数的校准, 以逗号分开, 例如: ‘SP1是 CV1乘以仪器读出值, SP2是 CV2乘以仪器读出值’ *);

(*47*) calibration for set parameter= real Number; (* 校准值, 以逗号分开, 例如: ‘1.054, 0.965’ *);

(*48*) label line = ‘environment description’ , EOL; (* 环境描述*);

(*49*) environment mode = 字符串 ;(* 实验环境, 例如: ‘UHV(超高真空)’ , ‘air(大气)’ , ‘liquid (液体)’ , ‘controlled atmosphere (可控气氛)’ 等*);

(*50*) sample temperature = 实数; (* 样品环境温度, 以 K 为单位表达*);

(*51*) surroundings pressure = 实数; (* 样品环境大气压, 以Pa为单位表达*);

(*52*) environment humidity = 实数; (* 相对湿度*);

相对湿度是一定温度下, 局部水蒸气压力与饱和水蒸气压力的比, 用百分比来表示。

(*53*) comment line = 字符串; (* 其他环境说明*);

(*54*) label line = ‘probe description’ , EOL; (* 探针描述*);

(*55*) probe identifier = 字符串; (* 用来采集数据的探针名称*);

(*56*) probe material = 字符串; (* 探针材料, 例如: Si, Si₃N₄, W, Pt-Ir等*);

(*57*) normal spring constant = 实数; (* 探针弹性系数, 以 N/m 表示*);

(*58*) resonance frequency = 实数; (* 探针共振频率, 以 Hz表示*);

(*59*) cantilever sensitivity = 实数; (* 悬臂Z方向的偏差, 以nm 表示, 在力谱测量下, 微悬臂的灵敏度以 V/nm 表示, 可以将微悬臂偏差的单位由 V 转换成nm。*);

(*60*) angle between probe and X axis = 实数; (* 探针与X 轴正方向逆时针旋转角度*);

(*61*) angle between probe vertical movement and Z axis in X azimuth = 实数; (* 探针垂直运动与Z轴在X 方位的夹角*);

(*62*) angle between probe vertical movement and Z axis in Y azimuth = 实数; (* 探针垂直运动与Z轴在Y方位的夹角*);

(*63*) comment line = 文本行; (* 其他探针信息*);

(*64*) label line = ‘sample description’, EOL; (* 样品信息*);

(*65*) sample identifier = 字符串; (* 样品名称, 例如: ‘Si(001) surface: P-doped 0.01ohm·cm’ 等*);

(*66*) species label= 字符串; (* 样品化学成分, 例如: ‘Si’ *);

(*67*) comment line = 字符串; (* 样品其他信息*);

(*68*) label line = ‘single-channel mapping description’, EOL; (* 单通道图像描述*);

(*69*) Z axis channel = 字符串; (* 实验模式为 MAP_SC 的 Z 轴输入信号, 例如: ‘height’, ‘tunnelling current’, ‘the Number of photons’ 等 *);

(*70*) physical unit of Z axis channel = 单位; (*Z轴物理单位, 例如: ‘nm’, ‘nA’, ‘c/s’ *); (*71*) comment line = 字符串; (*Z轴其他信息*);

(*72*) label line = ‘spectroscopy description’, EOL; (* 谱图描述*);

(*73*) spectroscopy mode = 字符串; (*SPM 谱图测量模式, 例如:

I-V spectroscopy—传导表面与探针间电流随电压变化,

I-Z spectroscopy—传导表面与探针间电流随 Z高度变化,

force-distance curve—探针与样品间力与探针高度的变化*);

(*74*) spectroscopy scan mode = (‘REGULAR’ | ‘IRREGULAR’), EOL; (* ‘REGULAR’ —横坐标间隔均匀, 探针位置固定, 可以省略横坐标, ‘IRREGULAR’ —横坐标间隔不均匀, 探针位置固定, 横坐标值应加入数组*);

(*75*) abscissa label = 字符串; (* 横坐标标题, 例如: ‘sample bias voltage’ *);

(*76*) abscissa units = 单位; (* 谱横坐标单位, 例如: V*);

(*77*) abscissa start = 实数; (* 谱横坐标起始值*);

(*78*) abscissa end = 实数; (* 终点值*);

(*79*) abscissa increment = 实数; (* 横坐标增量*);

(*80*) calibration constant for abscissa = 实数; (* 横坐标校准常数*);

(*81*) Number of points in abscissa = 整数; (* 横坐标测量点个数*);

(*82*) Number of ordinate items = 不小于1的整数; (* 纵坐标个数*); 在SPEC_MC 和 multi-channel spectra模式下, 数量应是二个或以上。

(*83*) ordinate label(s) = 字符串; (* 各纵坐标以逗号分开, 如: ‘tunnelling current, phase shift’ *);

(*84*) ordinate unit(s) = 单位; (* 纵坐标单位以逗号分开, 例如: ‘nA, degree’ *);

- (*85*) calibration constant(s) for ordinate = 实数; (* 纵坐标校准常数*);
- (*86*) comment line = 字符串; (* 谱图说明*);
- (*87*) label line = ‘data Treatment description’, EOL; (* 数据处理描述*);
- (*88*) data Treatment = (‘rawdata’ (原始数据) | ‘pre-treateddata’ (预处理数据) | ‘post-treated data’ (处理后的数据)), EOL; (* 数据处理类型*);
- (*89*) plain correction = 字符串; (* 平坦矫正法, 用于矫正样品倾斜和不期望的平坦假象*);
- (*90*) numerical filtering = 字符串; (* 数字滤波方法, 例如: ‘Fourier filtering’, ‘parametric low pass filter with a weight factor of 1.3’, ‘median filtering with a kernel Sizeof 1×5’ 等*);
- (*91*) image reconstruction = 字符串; (* 图像重建, 例如: ‘blind reconstruction’ *);
- (*92*) comment line = 字符串; (* 其他数据处理说明*);
- (*93*) label line = ‘multi-channel mapping description’, EOL; (* 多通道图像描述*);
- (*94*) Number of data channels = 大于1的整数; (* 通道数, 实验模式为: MAP_MC*);
- (*95*) 1st data channel= 字符串; (*1通道信号*);
- (*96*) 1st data channel unit = 单位; (*1通道信号单位*);
- (*97*) 1st data channel comment= 字符串; (*1通道信息*);
- (*98*) 2nd data channel = 字符串; (*2通道信号*);
- (*99*) 2nd data channel unit= 单位; (*2通道信号单位*);
- (*100*) 2nd data channel comment= 字符串; (*2通道信息*);
- (*101*) 3rd data channel = 字符串; (*3通道信号*);
- (*102*) 3rd data channel unit= 单位; (*3通道信号单位*);
- (*103*) 3rd data channel comment = 字符串; (*3通道信息*);
- (*104*) 4th data channel = 字符串; (*4通道信号*);
- (*105*) 4th data channel unit = 单位; (*4通道信号单位*);
- (*106*) 4th data channel comment = 字符串; (*4通道信息*);
- (*107*) 5th data channel = 字符串; (*5通道信号*);
- (*108*) 5th data channel unit= 单位; (*5通道信号单位*);
- (*109*) 5th data channel comment = 字符串; (*5通道信息*);
- (*110*) 6th data channel = 字符串; (*6通道信号*);
- (*111*) 6th data channel unit = 单位; (*6通道信号单位*);
- (*112*) 6th data channel comment = 字符串; (*6通道信息*);
- (*113*) 7th datachannel = 字符串; (*7通道信号*);

- (*114*) 7th data channel unit= 单位; (*7通道信号单位*);
- (*115*) 7thdatachannelcomment= 字符串;(*7通道信息*);
- (*116*) 8th data channel = 字符串; (*8通道信号*);
- (*117*) 8th data channel unit = 单位; (*8通道信号单位*);
- (*118*) 8th data channel comment = 字符串; (*8通道信息*);
- (*119*) comment line = 字符串; (* 描述多通道的全部信息*);
- (*120*) 空字符串;
- (*121*) 空字符串;
- (*122*) 空字符串;
- (*123*) 空字符串;
- (*124*) 空字符串;
- (*125*) 空字符串;
- (*126*) 空字符串;
- (*127*) 空字符串; (*120到127为未来扩展版本保留行*);
- (*128*) end of header identifier = ‘end of header’ , EOL; (* 头文件结束*);

5.4.4 扩展参数表

扩展参数表表示SPM数据获取中的基本参数以外的扩展参数，用于补充基本参数表中未准确表述的参数。扩展参数表采用结构化的描述方法，包含基本参数表头、基本参数表项。

5.4.4.1 扩展参数表表头

(*1*) Extend Parameters Identifier=字符 (*4个字节的内容用来识别参数表的类型*)；

扩展参数表标识符='EXTD'，暂固定不变并保留升级。

(*2*) ExtendParameters Size=整数 (*4个字节的内容用来表示扩展参数表的总字节数，不包含扩展参数表表头*)

(*3*) Extend Parameters Number=整数 (*4个字节的内容用来表示扩展参数表数据的项数*)

5.4.4.2 扩展参数表数据

扩展参数表项包括图像显示信息表、谱测量及显示信息表、图像数据显示彩色表、实验扩展参数表、用户自定义信息表等。

5.4.4.2.1 图像显示信息表

图像显示信息表用来记录当前图像显示所需的信息，以便根据这些信息还原当前图像的显示效果。图像显示信息表包括一个图像显示信息表表头、若干个图像显示信息表数据。如果当前图像文件包含了多个成像通道的多幅图像，则每一幅图像均需要一套图像显示信息的数据。

5.4.4.2.1.1 图像显示信息表表头

(*1*) Channels Parameters Identifier=字符 (*4个字节的内容用来识别参数表的类型*)；
参数表类型标识符='IMAG'，暂固定不变并保留升级。

(*2*) Channels Size=整数 (*4个字节的内容用来表示图像显示信息表数据的字节数*)

(*3*) Channels Number=整数 (*4个字节的内容用来表示图像显示信息表数据的项数*，即表示有多少个信号通道*)

5.4.4.2.1.2 图像显示信息表数据 (Image Display)

(*1*) Label = 字符串；(* 成像信号标签 *)；

(*2*) Channel No = 整数；(* 信号通道编号 *)；

(*3*) Pass No = 整数；(* 扫描遍次 *)；

(*4*) Unit = 字符串；(* 成像信号数据的单位，如nm, mV等 *)；

(*5*) Data Start = 实数；(* 成像信号数据的起始值 *)；

(*6*) Data End = 实数；(* 成像信号数据的终止值 *)；

(*7*) Display Start = 实数；(* 成像信号数据显示的起始值 *)；

(*8*) Display End = 实数；(* 成像信号数据显示的终止值 *)；

(*9*) Important Data Start = 实数；(* 成像信号重要数据的起始值 *)；

(*10*) Important Data End = 实数；(* 成像信号重要数据的终止值 *)；

(*11*) Display color used = 整数 (*4个字节表示使用的颜色数目*)

用4字节表示本通道的SPM图像数据在显示时实际使用的颜色索引数（设为0的话，则说明使用彩色表中的所有颜色）。

(*12*) Display color Important = 整数 (*4个字节表示重要数据使用的颜色数目*)

用4字节表示本通道的SPM图像数据在显示时重要数据段使用的颜色数目；如果是0，表示都重要，即重要数据段和其它数据段所分配的颜色索引数尽量均匀分布。

(*13*) Display color Identifier = 整数；(* 彩色表索引号，默认值0，表示使用文件头里面的默认彩色表 *)；

(*14*) Display color Number = 整数 (*4个字节的内容用来表示上述彩色表索引号所对应的彩色表的数据项数*)

(*15*) Comment = 字符串；(* 成像信号说明文字，可选 *)；

5.4.4.2.2 谱测量显示信息表

谱测量显示信息表用来记录当前谱曲线的测量及显示的相关信息，以便根据这些信息还原当前谱曲线的显示效果。谱测量显示信息表包括一个谱测量显示信息表表头、若干个谱测量显示信息表数据。如果当前谱文件包含了多个待测变量、多遍及多个测量位置的多条谱曲线，则每一条谱曲线均需要一套谱测量显示的信息数据。

5.4.4.2.2.1 谱测量显示信息表表头

(*1*) Spectra Parameters Identifier = 字符 (*4个字节的内容用来识别参数表的类型*)；
参数表类型标识符='SPEC'，暂固定不变并保留升级。

(*2*) Spectra Parameters Size = 整数 (*4个字节的内容用来表示谱测量显示信息表数据的字节数*)

(*3*) Spectra Number = 整数 (*4个字节的内容用来表示谱测量显示信息表数据的项数*)

表示谱曲线的数量，通常情况下 谱曲线数量 = 待测变量数量 * 测量遍数 * 位置数量。

(*4*) Ordinates Number = 整数 (*4个字节的内容用来表示纵坐标信息表数据的项数，即表示有多少个待测变量或纵坐标*)

(*5*) Times Number = 整数 (*4个字节的内容用来表示测量遍数，即表示每个变量测量的遍数*)

(*6*) Positions Number = 整数 (*4个字节的内容用来表示测量位置数据的项数*，即表示有多少个测量位置*)

(*7*) Display color used = 整数 (*4个字节表示使用的颜色数目*)

用4字节表示谱曲线数据在显示时实际使用的颜色数目（设为0的话，则说明使用彩色表中的所有颜色）。用4字节表示本谱曲线在数据显示时实际使用的颜色索引数（每条谱线通常只使用1个颜色，最多使用2个颜色），即颜色数目等于谱曲线的数量或该数量的两倍。

(*8*) Display color Important = 整数 (*4个字节表示每条谱曲线使用的颜色数目*)

用4字节表示每条谱曲线数据在显示时使用的颜色数目。通常设为0或者2，表示都重要，即每条谱线使用2个不同的颜色值；也可以设为1表示每条谱曲线只使用1个颜色。

(*9*) Display color Identifier = 整数 (* 彩色表索引号，默认值0，表示使用文件头里面的默认彩色表 *)；

(*10*) Display color Number = 整数 (*4个字节的内容用来表示上述彩色表索引号所对应的彩色表的数据项数*)

5.4.4.2.2.2 谱纵坐标信息表数据 (Ordinate)

(*1*) Label = 字符串；(* 纵坐标标签 *)；

(*2*) Unit = 字符串；(* 纵坐标的单位，如nm, mV等 *)；

(*3*) Calibration Constant = 实数; (* 纵坐标校准常数 *);

(*4*) Comment = 字符串; (* 纵坐标说明文字, 可选 *);

5.4.4.2.2.3 谱测量位置表数据 (Position)

(*1*) X = 实数; (* 测量位置的X坐标 *);

(*2*) Y = 实数; (* 测量位置的Y坐标 *);

(*3*) Z = 实数; (* 测量位置高度 *);

(*4*) Times = 整数; (* 该测量位置的测量次数 *);

(*5*) Unit = 字符串; (* 测量位置X、Y、Z的单位, 如nm, mV等 *);

(*6*) Comment = 字符串; (* 测量位置说明文字, 可选 *);

5.4.4.2.2.4 谱曲线显示信息表数据 (Spectrum Display)

(*1*) Label = 字符串; (* 谱曲线信息显示标签 *);

(*2*) Spectrum No = 整数; (* 谱曲线编号, 从0开始编号 *);

(*3*) Data Start = 实数; (* 谱曲线数据的起始值 *);

(*4*) Data End = 实数; (* 谱曲线数据的终止值 *);

(*5*) Display Start = 实数; (* 谱曲线数据显示的起始值, 对应纵坐标的起始值 *);

(*6*) Display End = 实数; (* 谱曲线数据显示的终止值, 对应纵坐标的终止值 *);

将本谱曲线数据在显示时映射到纵坐标的起始值和终止值。

(*7*) Display color used = 整数 (*表示每条谱曲线使用的颜色数目*)

表示本谱曲线数据在显示时使用的颜色数目。通常设置为2, 表示每条谱线使用2个不同的颜色值; 也可以设置为1表示每条谱曲线只使用1个颜色。

(*8*) Comment = 字符串; (* 谱曲线数据显示说明文字, 可选 *);

5.4.4.2.3 实验扩展参数表

实验参数表包括一个实验参数表表头、一个或多个实验参数表数据。

5.4.4.2.3.1 实验扩展参数表表头

(*1*) Experiment Parameters Identifier=字符 (*4个字节的内容用来识别参数表的类型*);

实验参数表标识符='EXPR', 暂固定不变并保留升级。

(*2*) Experiment Parameters Size=整数 (*4个字节的内容用来表示实验参数信息表数据的总字节数*)

(*3*) Experiment Parameters Number=整数 (*4个字节的内容用来表示实验参数表的项数, 即表示有多少套实验参数数据*)

5.4.4.2.3.2 实验参数表数据 (Experiment parameter)

- (*1*) Label = 字符串; (* 配置参数标签 *);
- (*2*) Unit = 字符串; (* 配置参数的单位, 如nm, mV等 *);
- (*3*) Value = 实数; (* 配置参数值 *);
- (*4*) Comment = 字符串; (* 配置参数说明文字, 可选 *);

5.4.4.2.4 图像数据显示彩色表

图像数据显示彩色表包括一个图像显示彩色表头、若干个图像数据显示彩色表数据。

5.4.4.2.4.1 图像数据显示配色表表头

(*1*) Palettes Parameters Identifier=字符 (*4个字节的内容用来识别参数表的类型*);
图像数据显示彩色表标识符='PALT', 暂固定不变并保留升级。

(*2*) Palettes Size =整数 (*4个字节的内容用来表示图像显示彩色表数据的字节数*)

(*3*) Palettes Number=整数 (*4个字节的内容用来表示图像显示彩色表数据的项数*)
表示彩色表的数量。

5.4.4.2.4.2 图像显示配色表数据 (Palette)

(*1*) Identifier = 整数; (* 图像显示彩色表索引号, 从1开始编号 *);

默认彩色表 (0号) 存储在文件头中。

(*2*) Number = 整数 (*4个字节的内容用来表示上述彩色表索引号所对应的彩色表的数据项数*)

(*3*) Colors = 整数列表; (* 多个颜色值的数据列表 *);

(*4*) Comment = 字符串; (* 图像显示配色表说明文字, 可选 *);

5.4.5 关联参数表

关联参数表是对SPM获取数据和参数的补充描述, 或者对SPM数据之后的分析处理的描述, 或者对SPM数据之间的关联关系的描述, 包括对相应的数据获取设备或分析处理软件等的接口的描述。

5.4.5.1 关联参数表头

关联参数表头用于对当前的SPM数据文件进行分类和标识。

(*1*) Relation Parameters Identifier=字符 (*4个字节的内容用来识别参数表的类型*);
关联参数表标识符='RELA', 暂固定不变并保留升级。

(*2*) Relation Parameters Size=整数 (*4个字节的内容用来表示关联参数表数据的字节数*)

(*3*) Relation Parameters Number=整数 (*4个字节的内容用来表示关联参数表数据的项数*)

(*4*) Data Identifier = 字符串; (* 16个字节, SPM数据的唯一标识 *);

(*5*) Auxiliary Identifier = 字符串 (*16个字节, SPM数据的辅助标识, 主要包含数据格式及自定义的标识符*) ;

SPM数据的辅助标识 (CSPM、CSPM、ISO28600、ChinaSPM-ZungWin、ISO28600-SYSU, 其中ChinaSPM、ISO28600属于格式标识符)。

(*6*) Subdata Coordinates 1 = 整数; (*2个字节的内容用来表示子数据坐标1的最大数*)

例如, 用于指定包含几副图像或者几个谱, 0表示未指定;

(*7*) Subdata Coordinates 2 = 整数; (*2个字节的内容用来表示子数据坐标2的最大数*) ;

(*8*) Subdata Coordinates 3 = 整数; (*2个字节的内容用来表示子数据坐标3的最大数*) ;

(*9*) Subdata Coordinates 4 = 整数; (*2个字节的内容用来表示子数据坐标4的最大数*) ;

5.4.5.2 关联参数表数据

关联参数表项包括数据来源表、插件信息表、处理参数表。

5.4.5.2.1 数据来源表

数据来源表包括一个数据来源表表头、若干个 (可以为0个) 数据来源表数据。

5.4.5.2.1.1 数据来源表表头

(*1*) Parameters List Identifier=字符 (*4个字节的内容用来识别参数表的类型*) ;

数据来源表标识符='DTSR', 暂固定不变并保留升级。

(*2*) Data Source Parameters Size=整数 (*4个字节的内容用来表示数据来源表数据的字节数*) ;

(*3*) Data Source Parameters Number=整数 (*4个字节的内容用来表示数据来源表数据的项数*) ;

5.4.5.2.1.2 数据来源表数据 (Data Source)

(*1*) Data Identifier = 字符串; (* SPM数据的唯一标识*) ;

(*2*) Format Identifier = 字符串 (*输出的、附加数据的格式类型*) ;

数据格式及自定义标识符(ChinaSPM、ISO28600、ChinaSPM-ZungWin、ISO28600-SYSU, 其中ChinaSPM、ISO28600属于格式标识符)。

(*3*) Subdata Coordinates 1 = 整数; (*2个字节的内容用来表示子数据坐标1*) ,

例如, 用于指定使用其中的第几副图像或者第几个谱, 0表示未指定;

(*4*) Subdata Coordinates 2 = 整数; (*2个字节的内容用来表示子数据坐标2*) ;

(*5*) Subdata Coordinates 3 = 整数; (*2个字节的内容用来表示子数据坐标3*) ;

(*6*) Subdata Coordinates 4 = 整数; (*2个字节的内容用来表示子数据坐标4*) ;

5.4.5.2.2 插件信息表

插件表用于对得到当前的SPM数据文件所使用的插件进行说明，描述SPM数据的处理过程和关联关系。SPM的数据获取设备、分析处理软件等均可称为插件，如SPM的后处理插件、驱动插件、格式插件、显示插件等。用户可以定义新的插件。插件表对相应插件的特征、接口、功能等进行描述，并可保存插件处理的参数、以及处理前和处理后的SPM数据文件之间的关联关系。

插件信息表包括一个插件信息表表头、若干个（可以为0个）插件信息表数据。

5.4.5.2.2.1 插件信息表表头

(*1*) Parameters List Identifier=字符（*4个字节的内容用来识别参数表的类型*）；
插件信息表标识符='PLUG'，暂固定不变并保留升级。

(*2*) Plugin Parameters Size=整数（*4个字节的内容用来表示插件信息表数据的字节数*）；

(*3*) Plugin Parameters Number=整数（*4个字节的内容用来表示插件信息表数据的项数*）；

5.4.5.2.2.2 插件信息表数据 (Plugin)

插件表包含有关于插件类型、插件名称、插件标识或版本、处理类型、插件数量、插件中的数据通道使用情况、插件及插件中的数据的大小等信息。

(*1*) SPM Plugin Name=字符串（* 插件名称 *）；

SPM数据文件类型标识符（PLUG0000、PLUG-RAW、PLUGPOST、PLUGCSPM、PLUGWin0、PLUGBenY）。

(*2*) Plugin Type=整数（*4个字节的内容用来表示插件的类型*）

插件类型包括：后处理插件（1）、驱动插件（2）、格式插件（3）、显示插件（4）等。

(*3*) Plugin Identifier=字符串（*插件的唯一标识*）

(*4*) Parent Plugin Identifier=字符串（*父插件的唯一标识*）

当插件没有父插件时本属性为空

(*5*) SPM Data Format=字符串（* 用来识别输出数据的格式类型 *）；

SPM数据文件格式标识符（CSPM、CSPM0000、CSPMMPMC、CSPMSPMC、SYSU、ZungWin、ISO28600）。

(*6*) Plugin Content = 字节串；（* 简短的、自定义插件的内容，可选 *）；

(*7*) Comment = 字符串；（* 插件信息说明文字，可选 *）；

5.4.5.2.3 处理参数表

处理参数表用于存放用户数据处理过程中的参数项，可包含多种用户数据处理参数，如数据标注、图像滤波等。处理参数表包括一个处理参数表表头、若干个（可以为0个）处理参数表数据。

5.4.5.2.3.1 处理参数表表头

(*1*) Parameters List Identifier=字符（*4个字节的内容用来识别参数表的类型*）；
处理参数表标识符='TRMT'，暂固定不变并保留升级。

(*2*) Treatment Parameters Size=整数(*4个字节的内容用来表示处理参数表数据的字节数*);

(*3*) Treatment Parameters Number=整数(*4个字节的内容用来表示处理参数表数据的项数*);

5.4.5.2.3.2 处理参数表数据 (Treatment parameter)

(*1*) Label = 字符串; (* 数据处理参数标签 *);

(*2*) Plugin Identifier = 字符串 (*数据处理使用插件的唯一标识*);

(*3*) Plugin Parameters = 字符串列表 (* 数据处理参数列表 *);

(*4*) Comment = 字符串 (* 数据处理参数说明文字, 可选 *);

5.4.6 特殊格式参数表

特殊格式参数表是用于替代默认参数表的、描述特定的SPM图像数据类型 (data type identifier=0x0000) 的特殊格式的参数表。特殊参数表主要对这种特殊的SPM图像数据的获取进行描述, 是主要用于SPM图形数据存储的、简化的特殊版本。特殊参数表包括实验仪器信息、扫描范围、扫描速度、实纵坐标信息、其它实验参数等信息。

特殊格式参数表的数据项的数据类型可以是计算机程序语言支持的基本数据类型, 包含整型、浮点、字符串、字节串等, 也可以包含由以上基本数据类型组成的结构数据类型。

5.4.6.1 特殊格式参数表内容

(*1 *) Version = 字符串; (* 控制软件版本 *)

(*2 *) Date = 字符串; (* 数据生成时间 *)

(*3 *) sTitle = 字符串; (* 数据标题 *)

(*4 *) Image width = 整数; (* X方向分辨率 *)

(*5 *) Image height = 整数; (* Y方向分辨率 *)

(*6 *) Ref = 实数; (* 反馈参考点 *)

(*7 *) Bias = 实数; (* 偏压, V *)

(*8 *) P.G. = 整数; (* 比例增益值 *)

(*9 *) I.G. = 整数; (* 积分增益值 *)

(*10*) ScanSpeed = 实数; (* 扫描速度, 每秒行数 *)

(*11*) Source = 字符串; (* 数据来源, 正向扫描或回扫 *)

(*12*) ScanSize = 实数; (* XY方向扫描范围, nm *)

(*13*) ScanX0 = 实数; (* X方向中心偏移, nm *)

(*14*) ScanY0 = 实数; (* Y方向中心偏移, nm *)

(*15*) HeightScale = 实数; (* 高度数据范围, nm *)

- (*16*) StartHeightScale = 实数; (* 起始高度, nm *)
- (*17*) Machine = 字符串; (* 工作模式, 接触或轻敲 *)
- (*18*) MaxValue = 整数; (* 数据最大值 *)
- (*19*) Scan Angle = 整数; (* 扫描角度 *)
- (*20*) Start Lift Height = 实数; (* 起始抬起高度, nm *)
- (*21*) Keep Lift Height = 实数; (* 保持抬起高度, nm *)
- (*22*) Signal Gain = 实数; (* 成像信号增益 *)
- (*23*) Signal Brightness = 实数; (* 成像信号亮度 *)
- (*24*) ZLimit = 整数; (* 可用于Z方向高度限制 *)
- (*25*) Filter = 整数; (* 可用于滤波器定义 *)
- (*26*) Surface Fit = 整数; (* 可用于预定义的曲面拟合 *)
- (*27*) Auto Refresh = 字符串; (* 是否自动刷新图像 *)
- (*28*) Sensitivity = 实数; (* 系统灵敏度, mV/nm *)
- (*29*) Auto ZLimit = 字符串; (* 可用于自动Z方向高度限制 *)
- (*30*) Tapping Amplitude = 实数; (* 轻敲自由振幅 *)
- (*31*) Lift Tap. Amplitude = 实数; (* 抬起时自由振幅 *)
- (*32*) Tapping Drive Amplitude = 实数; (* 轻敲激励电压, V *)
- (*33*) Lift Drive Amplitude = 实数; (* 抬起时激励电压, V *)
- (*34*) Tapping Frequency = 实数; (* 震动频率, kHz *)
- (*35*) Tapping Phase = 实数; (* 相位差, 角度 *)
- (*36*) Lift Tap. Frequency = 实数; (* 抬起时震动频率, kHz *)
- (*37*) Lift Tap. Phase = 实数; (* 抬起时相位差 *)
- (*38*) Serial Number = 字符串; (* AFM仪器序列号 *)
- (*39*) Checksum = 整数; (* 软件校验和 *)
- (*40*) Dsp Version = 字符串; (* DSP固件版本号 *)
- (*41*) Online Version = 字符串; (* 控制台版本号 *)
- (*42*) Scanner Name = 字符串; (* 扫描器名称 *)
- (*43*) LookAhead Gain = 整数; (* 可用于反馈信号增益 *)
- (*44*) Max Scope = 实数; (* XY方向扫描器最大范围 *)
- (*45*) Z Scale = 实数; (* Z方向扫描器最大范围 *)
- (*46*) X Fast = 实数; (* X方向快扫伸缩系数 *)
- (*47*) Y Fast = 实数; (* Y方向快扫伸缩系数 *)
- (*48*) X Slow = 实数; (* X方向慢扫伸缩系数 *)

(*49*) Y Slow = 实数; (* Y方向慢扫伸缩系数 *)

(*50*) AX = 实数; (* X方向校正系数1 *)

(*51*) BX = 实数; (* X方向校正系数2 *)

(*52*) CX = 实数; (* X方向校正系数3 *)

(*53*) DX = 实数; (* X方向校正系数4 *)

(*54*) AY = 实数; (* Y方向校正系数1 *)

(*55*) BY = 实数; (* Y方向校正系数2 *)

(*56*) CY = 实数; (* Y方向校正系数3 *)

(*57*) DY = 实数; (* Y方向校正系数4 *)

(*58*) TipType = 字符串; (* 探针类型 *)

附录 A 格式示例

A.1 示例-1：数据文件头

SPM数据文件头可用SPMDATAFILEHEADER结构来定义：

```
typedef struct tagSPMDATAFILEHEADER { /* sdfh */
    UINT16 bfType;
    UINT32 sfSize;
    UINT32 sfDataType;
    UINT32 sfOffBits;
} SPMDATAFILEHEADER;
```

A.2 示例-2：数据信息头

SPM数据信息头可用SPMDATAINFOHEADER结构来定义：

```
typedef struct tagSPMDATAINFOHEADER { /* bmih */
    INT32 biHeadSize;
    INT32 biWidth;
    INT32 biHeight;
    UINT16 biPlanes;
    UINT16 biBitCount;
    UINT32 biCompression;
    UINT32 biDataSize;
    UINT32 biXScale;
    UINT32 biYScale;
    UINT32 biDataClrUsed;
    UINT32 biDataImportant;
} SPMDATAINFOHEADER;
```

A.3 示例-3：颜色定义表

颜色定义表中像素的颜色用RGBQUAD结构来定义。对于调色板中的每个表项，描述RGB的值用下述方法：1字节用于蓝色分量，1字节用于绿色分量，1字节用于红色分量，1字节用于填充符(设置为0)。

RGBQUAD结构描述由R、G、B相对强度组成的颜色，定义如下：

```
typedef struct tagRGBQUAD { /* rgbq */
    BYTE rgbBlue;
    BYTE rgbGreen;
    BYTE rgbRed;
    BYTE rgbReserved;
} RGBQUAD;
```

A.4 示例-4：数据阵列

下面的代码可以完成第一个字节表示位图左上角的像素，而最后一个字节表示位图右下角的像素，即正常的显示状态，便于操作。

```
int m, n;
unsigned char k;
m = BMPPIC.SPMDDataInfoHead.biWidth; //24
n = BMPPIC.SPMDDataInfoHead.biHeight; //96, 24*96 = 2304 bytes
for(int i=0; i < n/2; i++)
{
    for(int a=0; a < m; a++)
    {
        k = pbufout1[m*(n-i-1) + a];
        pbufout1[m*(n-i-1) + a] = pbufout1[m*i + a];
        pbufout1[m*i + a] = k;
    }
}
```

A.5 示例-5：参数表头

SPM数据的参数表头中可用SPMTOTALPARAMETERSHEADER结构来定义：

```
typedef struct tagSPMTOTALPARAMETERSHEADER {
    UINT32 saParametersIdentifier; // 参数表的类型
    INT32 saParametersSize; // 参数表字节数
    INT32 saParametersNumber; // 参数表数量
    UINT32 saMaxDataValue; // 位图数据的最大值
```

```

INT32 saBaseParametersOffset; // 基本参数表的偏移量(从参数表头开始)
INT32 saExtendParametersOffset; // 扩展参数表的偏移量(从参数表头开始)
INT32 saRelationParametersOffset; // 关联参数表的偏移量(从参数表头开始)
} SPMTOTALPARAMETERSHEADER;

```

A.6 示例-6：通用参数表头

SPM数据的子参数表头中可用SPMSUBPARAMETERSHEADER结构来定义：

```

typedef struct tagSPMSUBPARAMETERSHEADER {
    UINT32 saParametersIdentifier; // 参数表的类型
    INT32 saParametersSize; // 参数表字节数
    INT32 saParametersNumber; // 参数表数量
} SPMSUBPARAMETERSHEADER;

```

通用参数表头适用于基本参数表、扩展参数表及其子参数表、关联参数表的子参数表的表头。

A.7 示例-7：普测量显示信息表头

SPM数据的关联参数表头中可用SPMSPECTRAHEADER结构来定义：

```

typedef struct tagSPMSPECTRAHEADER {
    UINT32 saParametersIdentifier; // 参数表的类型
    INT32 saParametersSize; // 插件表字节数
    INT32 saSpectraNumber; // 谱曲线显示信息表数据项数量
    INT32 saOrdinatesNumber; // 纵坐标（待测变量）表项数量
    INT32 saTimesNumber; // 每个位置测量遍次数
    INT32 saPositionsNumber; // 测量位置表项数量
    INT32 saDisplayColorUsed; // 谱曲线数据在显示时使用的颜色数目
    INT32 saDisplayColorEach; // 每条谱曲线数据在显示时使用的颜色数目
    INT32 saPaletteIdentifier; // 对应彩色表索引号
    INT32 saPaletteNumber; // 对应彩色表颜色数量
} SPMSPECTRAHEADER;

```

A.8 示例-8：关联参数表头

SPM数据的关联参数表头中可用SPMDATARELATIONHEADER结构来定义：

```

typedef struct tagSPMDATARELATIONHEADER {
    UINT32 saParametersIdentifier; // 参数表的类型

```

```

    INT32 saParametersSize;    // 插件表字节数
    INT32 saParametersNumber; // 插件表数据的项数
    BYTE saDataIdentifier[16]; // 数据的唯一标识
    BYTE saDataFormat[16];    // 数据的格式类型
    UINT16 saSubdataCoordinates1; //
    UINT16 saSubdataCoordinates2; //
    UINT16 saSubdataCoordinates3; //
    UINT16 saSubdataCoordinates4; //
} SPMDATARELATIONHEADER;

```

A.9 示例-9：实验参数信息

SPM数据的实验参数信息，可用SPMEXPERIMENTPARAMETER结构来定义：

```

typedef struct tagSPMEXPERIMENTPARAMETER {
    String spLabel;        // 参数名称
    String spUnit;        // 参数的单位
    double spValue;       // 参数值
    double spCalibration; // 校准的参数值
    String spComment;     // 参数的说明
} SPMEXPERIMENTPARAMETER;

```

A.10 示例-10：纵坐标信息

SPM数据的纵坐标信息表，可用SPMORDINATE结构来定义：

```

typedef struct tagSPMORDINATE { /* ep */
    String spLabel;        // 参数名称
    String spUnit;        // 参数的单位
    double spCalibration; // 校准常数
    String spComment;     // 参数的说明
} SPMORDINATE;

```

A.11 示例-11：测量位置

SPM数据的测量位置，可用SPMPOSITION结构来定义：

```

typedef struct tagSPMPOSITION { /* ep */

```

```

double spX;        // 测量位置的X坐标
double spY;        // 测量位置的Y坐标
double spZ;        // 测量位置高度
UINT32 spTimes;   // 该测量位置的测量次数
String spUnit;     // 测量位置X、Y、Z的单位
String spComment; // 参数的说明
} SPMPOSITION;

```

A.12 示例-12：图像显示信息

SPM数据的图像显示信息表，可用SPMIMAGESIGNAL结构来定义：

```

typedef struct tagSPMIMAGESIGNAL {
    String spLabel;        // 成像信号名称
    String spUnit;        // 成像信号的单位
    UINT32 spChannelNo;   // 信号通道编号
    UINT32 spPassNo;      // 扫描遍次
    double spMinimum;     // 信号最小值
    double spMaximum;     // 信号最大终止值
    double spDisplayStart; // 成像信号数据显示的起始值
    double spDisplayEnd;  // 成像信号数据显示的终止值
    double spImportantStart; // 重要数据起始值
    double spImportantEnd; // 重要数据终止值
    INT32 spDisplayColorUsed; // 使用的颜色数目
    INT32 spDisplayColorImportant; // 重要数据使用的颜色数目
    INT32 spPaletteIndex; // 配色表索引号
    INT32 spPaletteNumber; // 配色表颜色数
    String spComment;     // 成像信号的说明
} SPMIMAGESIGNAL;

```

A.13 示例-13：图像显示配色表

SPM数据的图像显示配色表，可用SPMPALETTE结构来定义：

```

typedef struct tagSPMPALETTE {
    UINT32 spIndex;        // 配色表索引号

```

```

        List<UINT32> spColors;    // 颜色值列表
        String spComment;        // 参数的说明
    } SPMPALETTE;

```

A.14 示例-14：数据来源

SPM数据的数据来源中可用SPMDATASOURCE结构来定义：

```

typedef struct tagSPMDATASOURCE {
    Bytes saDataIdentifier;    // 数据的唯一标识
    Bytes saDataFormat;        // 数据的格式类型
    UINT16 saSubdataCoordinates1;    //
    UINT16 saSubdataCoordinates2;    //
    UINT16 saSubdataCoordinates3;    //
    UINT16 saSubdataCoordinates4;    //
} SPMDATASOURCE;

```

A.15 示例-15：处理参数

SPM数据的处理参数，可用SPMTREATMENTPARAMETER结构来定义：

```

typedef struct tagSPMTREATMENTPARAMETER {
    String spLabel;            // 参数的说明
    Bytes spPluginIdentifier;    // 参数的说明
    String spPluginParameters;    // 参数的说明
    String spComment;          // 参数的说明
} SPMTREATMENTPARAMETER;

```

A.16 示例-16：插件信息

SPM数据的插件信息，可用SPMPLUGIN结构来定义：

```

typedef struct tagSPMPLUGIN {
    String spPluginName;        // 插件名称
    UINT32 spPluginType;        // 插件类型， post-treated data, 显示等;
    Bytes spPluginIdentifier;    // 插件的唯一标识
    Bytes spParentPluginIdentifier;    // 父插件的唯一标识
    Bytes spDataFormat;          // 输出数据的格式标识符
} SPMPLUGIN;

```

```
Bytes spPluginContent;    // 自定义插件的内容, 可选
String spComment;        // 插件表参数的字节数
} SPMPPLUGIN;
```

参 考 文 献

- [1] ISO 18115-2-2013 , Surface chemical analysis. 词汇表. 第2部分:扫描-探针显微镜检查中应用的术语
- [2] ISO/DIS 28600:2011 Surface chemical analysis — Data transfer format for scanning-probe microscopy
- [3] GB/T 36052-2018 表面化学分析 扫描探针显微镜数据传送格式
-