

# 使用 PYTHON 处理采集的波形数据



罗德与施瓦茨示波器采用先进技术以提供可重复的可靠结果。

## 您的任务

开发或测试工程师在日常工作中使用 R&S®RTP、R&S®RTO 或 R&S®RTE 示波器。工程师存储记录的波形以作存档，或进行因功能或性能限制而无法使用示波器的进一步处理工作。将波形存储为 ASCII 格式（文件名后缀为 csv），可能会超出存储容量。传输和处理这种格式的文件也需要花费大量时间。此外，使用 ASCII 格式而非二进制格式时，部分信息会不可用。

## 罗德与施瓦茨解决方案

罗德与施瓦茨实验室用高性能示波器 R&S®RTP、R&S®RTO 和 R&S®RTE 均使用相同的二进制格式存储波形。高性能辅助工具 Python RTxReadBin 程序包可用于处理二进制文件并获取所有必要的详细信息以进行归档记录。

## 应用

为了将波形从仪器传输到 Python 环境，必须先准备好 Python 集成开发环境：

1. 从罗德与施瓦茨网站上下载 RTxReadBin 程序包
2. 在系统中安装程序包
3. 使用示波器采集波形，并将其存储为二进制文件

这将创建两个文件：波形描述文件 (\*.bin) 和波形数据文件 (\*.wfm.bin)。波形数据（通过 RTxReadBin 功能加载）可能包含以下信息：

- ▶ 模拟通道
- ▶ 数字通道（混合信号示波器 (MSO)）
- ▶ 并行总线（通过 MSO 采集）
- ▶ 模拟通道（通过 ZVC 采集）
- ▶ 数学运算波形，含频谱
- ▶ 轨迹（基于自动测量的波形）

对于所有这些信号源，支持多次采集和多个通道（如适用）。

使用罗德与施瓦茨示波器采集数据后，您可以将数据加载到 Python 环境以作进一步处理。但是，您也可以将波形数据保留在示波器中，并在计算机中安装 CIFS 共享以通过网络访问数据，这样更为便捷。在示波器上完成以下准备工作：

1. 转至高级共享
2. 选择公用配置文件并更改以下参数：
  - 启用网络发现
  - 启用文件和打印机共享
  - 启用写入公用文件夹



完成上述步骤后，您可以在 Python 脚本中通过以下文件路径连接到示波器：

```
r'\\<示波器名称或 IP 地址>\Users\Public\Documents\Rohde-Schwarz\RTx\RefWaveforms\  
<文件名>.bin'
```

注意，由于 Python 会解译字符串，因此 \U 字符序列将被解译并导致出错。为规避这个问题，只需在字符串前面添加一个“r”（表示原始字符串）。系统可能要求您提供凭证。确保不是对本地 Windows 域服务器进行身份认证。为此，请在本地帐户 instrument 前添加一个反斜杠：

```
帐户: \instrument  
密码: <示波器密码>
```

或者，您也可以通过 Windows 资源管理器从示波器中下载（拖放）波形。

完成这些准备工作后，您可以通过 Python 导入功能加载模块，并调用 RTxReadBin。最简单的方式是使用文件名。此功能将返回三个参数：垂直波形数据 (y)、水平轴数据 (x) 和采集参数 (S)。

```
> from RTxReadBin import RTxReadBin  
> import numpy  
> y, x, S = RTxReadBin('<波形文件名>.bin')  
> y.shape
```

垂直数据 (y) 是 NumPy 数组，形状为 [<记录长度>, <采集次数>, <活动通道数量>]。MSO 波形的数据类型为布尔型，并行总线的数据类型为使用最高 MSO 索引（最高有效位）进行位压缩的无符号整数 (uint16)。例如，假设 MSO 数据线  $D_{12}$ 、 $D_7$ 、 $D_2$  和  $D_0$  已连接并配置为并行总线。记录的数据字将存储为  $D_{12} \cdot 2^3 + D_7 \cdot 2^2 + D_2 \cdot 2^1 + D_0 \cdot 2^0$ 。所有其他波形将使用 32 位浮点型数据类型。值得一提的是，示波器与 Python 采用不同的方法计算采集历史。示波器从 (1 - <采集次数>) 开始至 0 结束并以此为最后一次采集。Python 则与此相反，从 0 开始至 (<采集次数> - 1) 结束并以此为最后一次采集。

水平数据 (x) 为 64 位浮点型 NumPy 数组，形状为 [<记录长度>]。如果以 x/y 交错格式存储数据，数组将扩展以涵盖所有采集 [<记录长度>, <采集次数>]。如果存储频谱，x 轴将包含频谱的频率轴。

通过访问返回值 y 的形状属性，可以轻松检索记录长度、采集次数和通道数量。

如果无需使用所有参数，可以通过标准 Python 方法忽略部分参数，例如添加列表说明符（见下方示例）或使用下划线作为位置返回参数。

```
> y, x =  
RTxReadBin('<波形文件名>.bin')[0,2]
```

您还可以指定其他两个（可选）参数以减少分配的内存量。第一个参数指定感兴趣的采集。第二个参数指定感兴趣的间隔、时间或频率。这两个参数均需要包含两个项目的列表。注意，采集参数遵循 Python 列表表示法。给定示例中返回两次采集 (2,3)。

```
> y, x, S = RTxReadBin('<波形文件名>.bin',  
acquisitions = [2,4],  
xInterval = [-2e-7,3e-7])
```

最后一个参数 S 提供头文件中可用的所有详细信息。此参数作为 Python 字典实现。对用户而言，某些参数可能并不重要。但是，一些参数有助于进行归档记录和分析。由于 y 向量中的波形数据仅为浮点型数据，因此用户无法据此确定原始设置。但是，用户可以查看报头中的参数，进而获得必要信息以作存档记录和后处理。例如，为了了解测量的噪声限制，需要获取有关垂直刻度、偏移、位置和耦合设置的信息。

相关示例还包括雷达应用中的脉冲重复间隔 (PRI) 分析。使用分段存储捕获每次采集的一个雷达脉冲时<sup>1)</sup>，分段捕获的时间戳列表将列明一段时间内的触发事件，且这些事件在其他情况下不可用 (S['Timestamps'])。根据连续触发事件的时间差得到 PRI 序列。

下方两张图显示了示波器和 Python 环境中加载的波形数据。图中显示了三个信号：

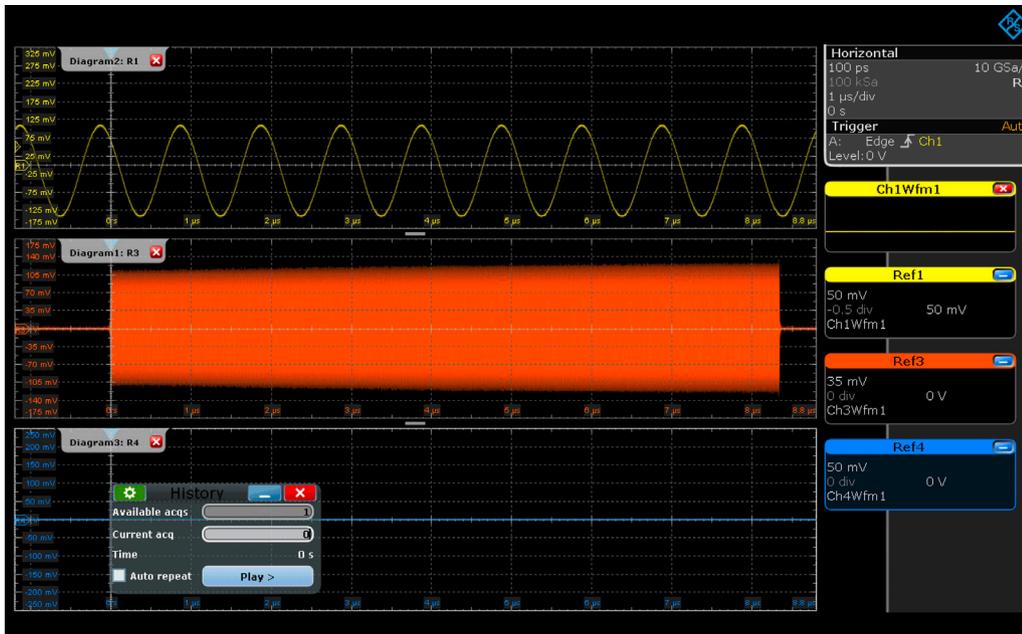
- ▶ 1 MHz 正弦信号，带偏移
- ▶ 交替的正负斜率雷达线性调频信号
- ▶ 噪声，无信号连接

RTxReadBin 功能的优点在于能够处理包含多次采集（历史）的波形文件。加载同一文件时，示波器仅识别最后一次采集，并显示空白的采集历史。

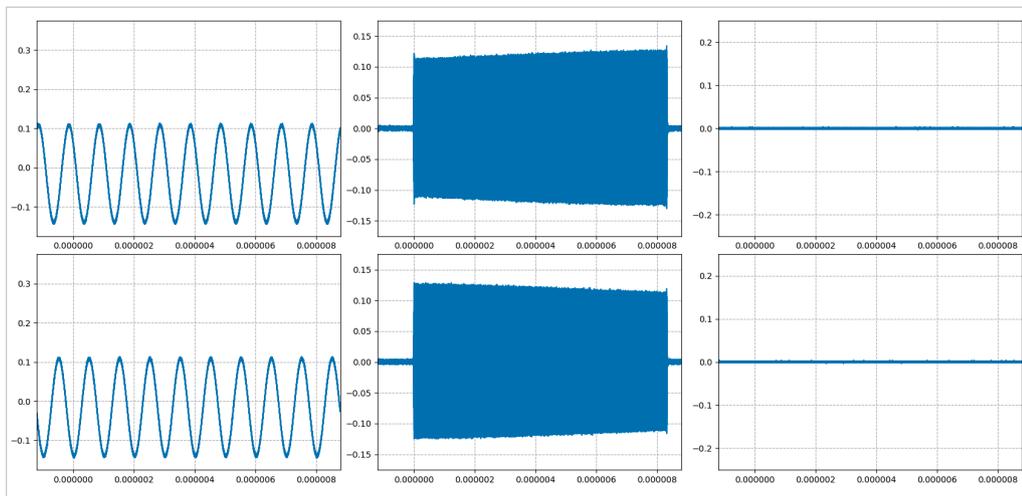
<sup>1)</sup> 访问罗德与施瓦茨网站，阅读《使用示波器解调雷达射频脉冲》应用说明：  
[www.rohde-schwarz.com/applications/demodulating-radar-rf-pulses-with-an-oscilloscope-application-card\\_56279-618819.html](http://www.rohde-schwarz.com/applications/demodulating-radar-rf-pulses-with-an-oscilloscope-application-card_56279-618819.html)

## 摘要

R&S®RTP、R&S®RTO 和 R&S®RTE 示波器功能强大。Python 模块 (RTxReadBin) 支持用户访问存储的波形，进一步扩展了仪器功能。对于需要进行后处理、归档记录或波形比较等后续分析工作的应用，强大的 Python 环境可提供所有波形数据。



三个无采集历史的不同波形加载为参考波形。



三个包含采集历史的不同波形。

## 增值服务

- ▶ 遍及全球
- ▶ 立足本地个性化
- ▶ 可订制而且非常灵活
- ▶ 质量过硬
- ▶ 长期保障

## 关于罗德与施瓦茨公司

作为测试测量、技术系统以及网络安全方面的行业先驱，Rohde&Schwarz technology group 通过领先方案为世界安全联网保驾护航。集团成立于85年前，致力于为全球工业企业和政府部门的客户提供可靠服务。集团总部位于德国慕尼黑，在全球70多个国家和地区设有分支机构，拥有广阔的销售和服务网络。

## 可持续性的产品设计

- ▶ 环境兼容性和生态足迹
- ▶ 提高能源效率和低排放
- ▶ 长久性和优化的总体拥有成本

Certified Quality Management  
**ISO 9001**

Certified Environmental Management  
**ISO 14001**

## 罗德与施瓦茨培训

[www.training.rohde-schwarz.com](http://www.training.rohde-schwarz.com)

## 罗德与施瓦茨客户支持

[www.rohde-schwarz.com/support](http://www.rohde-schwarz.com/support)

