

带PSI5、SPI和SQUIB的HiL设置 -Seskion GmbH-

内容:

- 1. Seskion HiL设置概述
 - 1.1 Seskion HiL解释
 - 1.2 可能的HiL设置
 - 1.3 FAQ
- 2. 详细信息
 - 2.1 数据准备
 - 2.2 二进制数组的系统模拟数据结构
 - 2.2.1 二进制数组的系统配置文件
 - SPI传感器的定义
 - PSI5传感器的定义
 - 2.2.2 PPF文件
 - 2.2.3 SPF文件
 - 2.2.4 工艺流程控制程序
 - 2.3 通过CSV文件的系统模拟数据结构
 - 2.3.1 CSV数据的系统配置文件
 - SPI传感器的定义
 - PSI5传感器的定义
 - 2.3.2 PPF文件
 - 2.3.3 SPF文件
 - 2.3.4 工艺流程控制程序

版本:	(1.0) 09.12.2021 – 创建
	(1.1) 28.01.2022 – 小改进

带PSI5、SPI和SQUIB的HiL设置

1. Seskion HiL设置概述

1.1 Seskion HiL 解释

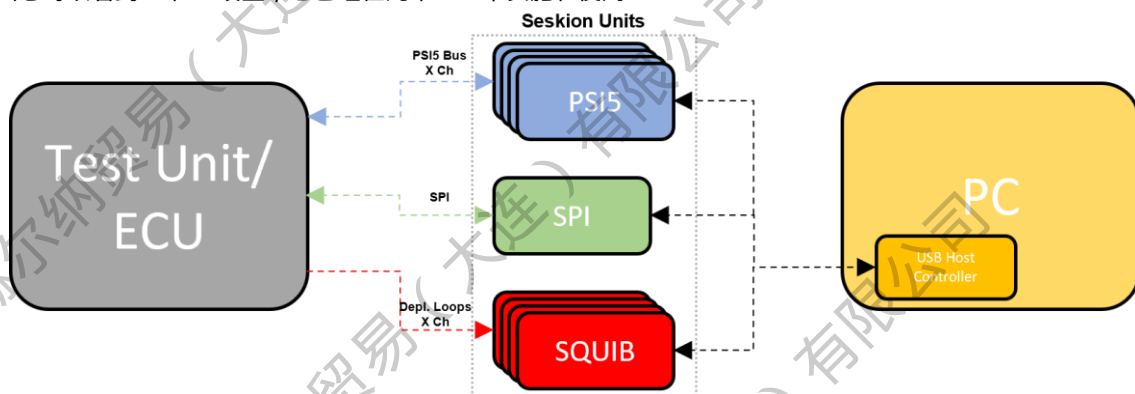
为了生成安全气囊控制单元的完整HiL仿真，必须连接控制单元的内部加速度传感器信号(SPI)和外围加速度和压力传感器信号(PSI5)。此外，通过USB端口/网络连接到模拟器的PC系统用于数据处理。

通常多达6个或更多的PSI5-Simulyzer与几个SPI-Simulyzer一起操作。为了协调，有一个上级配置，它决定使用哪个模拟器，哪个模拟器使用哪个信号跟踪，以及哪些详细配置将与总线参数一起使用。通过使用上级配置，API的操作非常简单。

还可以对传感器信号进行连续控制。

1.2 可能的HiL设置

下面您可以看到一个HiL设置，它已经在几个OEM中实施和使用：



1.3 FAQ

问题：我需要在我的测试ECU的所有通道上模拟同步的PSI5数据。有没有一种方法可以同步多个Simulyzer，使它们同时开始传输？

答：所有Simulyzer都有数字输入/输出，可以产生或接收触发信号。在接收到该触发信号时，在PSI5接口上流式传输模拟数据。

问题：我将拥有来自真实世界的传感器数据，我需要使用Simulyzer工具在模拟中“回放”这些数据。这可能吗？

答：这些数据已经通过USB排队到盒子，所以在触发信号和数据流启动之间没有延迟。流数据可以从.csv文件导入，也可以作为二进制数据在API接口上提供。

问题：多个模拟器需要并行连接在一起，才能在我的测试ECU的所有通道上实现模拟。我可以连接多少个？

答：在PC上并行使用Simulyzer没有限制。当并行使用多个Simulyzer时，我们建议在PC中使用专用USB主机控制器卡，而不是在主板上使用USB连接。

问题：我需要通过LabVIEW或C-API进行控制。您是否可以通过库和API支持实现上述所有要求？

答：我们的API提供了一个接口，可以将更多的Simulyzer分组到所谓的SimulyzerSystem中。该系统可以配置有一个API调用和一个配置文件，其中定义了系统的Simulyzer。

需要一个额外的API调用来向系统提供模拟数据。仅此而已。接收到外部触发器后模拟数据在PSI5接口上进行流式传输。

带PSI5、SPI和SQUIB的HiL设置

2. HiL系统的详细信息视图

实时系统：

提供实际操作的数据通信和剩余总线模拟，并通过触发脉冲同步模拟系统。

PSI5模拟器：

外围加速度或压力传感器信号由PSI5 Simulyzer箱模拟。每个Simulyzer最多可测量6个传感器。可以与实际传感器混合操作。每个待模拟的传感器及其信号必须根据实时系统进行描述。

SPI Simulyzer：

内部加速度传感器通过SPI Simulyzer箱进行模拟。要使用的盒子数量取决于传感器的数量和使用的接口。与真实传感器混合操作是可能的。系统中要模拟的每个传感器及其信号必须根据实时系统进行描述。

触发ECU：

提供点火脉冲。

PC系统：

测量数据的顺序控制和归档/分析的数据技术环境。

2.1 数据准备

通常的PSI5/SPI Simulyzerdata导出功能可用于分析/归档数据。

在协议级别上生成错误是可能的，并且必须已经包含在提供的流数据中或提供的ppf/spf文件中，必须已经创建或定义。

详细描述可在Simulyzer软件的相应帮助系统中找到。

可以通过火花塞盒检测到安全气囊展开（更换火花塞）。

可以与Seskion GmbH协商实现特殊解决方案，例如从GPS传感器或类似传感器提供位置数据。

还支持其他传感器协议，如SENT或DSI3。

2.2 采用二进制数组的系统仿真数据结构

以下示例系统用于描述：

1x PSI5-Simulyzer-Box 序列号.: 2283 通道0已使用 通道1未使用 1. 在插槽0中发送信号 初始化数据： 阶段 2: 42010714A480DF11790017C17037308C 阶段 3: 0x1e7:16 0x0 实时数据信号跟踪：“PSI5_IO_Slot0 Index 1” 2. 在插槽1中发送信号 初始化数据： 阶段 2: 42010714A480DF11790017C17037308D 阶段 3: 0x1e7:16 0x0 实时数据信号跟踪：“PSI5_IO_Slot1 Index 1”	1x SPI-Simulyzer-Box 序列号.: 162 通道0已使用 通道1未使用 1. 传感器信号 信号名称AccX 数据传输开始时出现下降沿 实时数据信号跟踪：“AccX Index 10” 2. 传感器信号 AccY信号 实时数据信号跟踪：“AccY Index 11” 3. 传感器信号 AccX_1信号 实时数据信号跟踪：“AccX_1 Index12”
---	--

带PSI5、SPI和SQUIB的HiL设置

2.2.1. 二进制数组的系统配置文件

系统配置文件是*.xml格式的强制性文件，用于配置HiL系统，参考：

- Simulyzer许可证验证
- 通过spf文件引用的SPI传感器信息（SPI Simulyzer软件生成的SPI配置数据）
- 根据模拟数据定义SPI信号名称。
- 通过ppf文件的PSI5传感器信息（PSI5 Simulyzer软件生成的PSI5配置数据）。
- PSI5信号名称定义根据仿真数据。

示例(SimulyzerSystemConfigurationFile.xml):

```
<?xml version="1.0" encoding="UTF-8"?>
<SimulyzerSystemConfigurationFile>
  <!-- -->
  <BasePath>./</BasePath>
  <LicenseFile>../seskionlicense.xml</LicenseFile>
  <Simulyzer>
    <Type>SPI</Type>
    <SerialNumber>162</SerialNumber>
    <StreamStartPin Polarity="activeLow">1</StreamStartPin>
    <ConfigurationFile>../configs/SPI/Example_Simulation_SPI_Sensor.spf</ConfigurationFile>
    <Sensor>
      <!-- StreamDataIndex index of data array provided by api call SimulyzerSystemSetSignalData for sensor simulation-->
      <!-- first index is for first Hawthorn chip channel 1 -->
      <StreamDataIndex sigName="AccX">10</StreamDataIndex>
      <!-- second index is for first Hawthorn chip channel 2 -->
      <StreamDataIndex sigName="AccY">11</StreamDataIndex>
      <!-- third index is for second Hawthorn chip channel 1 -->
      <StreamDataIndex sigName="AccX_1">12</StreamDataIndex>
    </Sensor>
  </Simulyzer>
  <Simulyzer>
    <Type>PSI5</Type>
    <SerialNumber>2283</SerialNumber>
    <ConfigurationFile>../configs/PSI5/Example_Simulation_PSI5_Sensor.ppf</ConfigurationFile>
    <Sensor>
      <!-- InterfaceIndex = defines the physical PSI5 interface on Simulyzer Box Value 0/1 -->
      <InterfaceIndex>0</InterfaceIndex>
      <!-- SlotIndex 0/1/2/3 defines the timeslot on PSI5 bus-->
      <SlotIndex>0</SlotIndex>
      <!-- InitPhase2Data string of hex encoded data nibbles for sensor initialization, start with first data nybble on left side-->
      <InitPhase2Data>42010714A480DF11790017C17037308C</InitPhase2Data>
      <!-- InitPhase3Data list of hex encoded data for init phase 3, :n behind value defines a repetition count of this value-->
      <InitPhase3Data>0x1e7:16 0x0</InitPhase3Data>
      <!-- StreamDataIndex index of data array provided by api call SimulyzerSystemSetSignalData for sensor simulation-->
      <StreamDataIndex sigName="PSI5_I0_Slot0">2</StreamDataIndex>
    </Sensor>
    <Sensor>
      <InterfaceIndex>0</InterfaceIndex>
      <SlotIndex>1</SlotIndex>
      <InitPhase2Data>42010714A480DF11790017C17037308D</InitPhase2Data>
      <InitPhase3Data> 0x1e7:16 0x0</InitPhase3Data>
      <StreamDataIndex sigName="PSI5_I0_Slot1">1</StreamDataIndex>
    </Sensor>
  </Simulyzer>
</SimulyzerSystemConfigurationFile>
```

Location of the Seskion license file

Definition of the SPI sensor signals

Definition of the PSI5 sensor signals 1

Definition of the PSI5 sensor signals 2

带PSI5、SPI和SQUIB的HiL设置

SPI传感器的定义

对于每个SPI Simulyzer盒，必须根据以下方案进行定义，实时数据中是否存在超过此处定义的传感器信号并不重要。忽略所有未定义的信号。

在该示例中，只使用了一个SPI Simulyzer框，因此只定义了一个以< Simulyzer > 开头和结尾的SPI定义块。每个使用的Simulyzer盒子都必须有一个定义块。

```
<Simulyzer>
  <Type>SPI</Type>
  <SerialNumber>162</SerialNumber>
  <StreamStartPin Polarity="activeLow">1</StreamStartPin>
  <ConfigurationFile>../configs/SPI/Example_Simulation_SPI_Sensor.spf</ConfigurationFile>
  <Sensor>
    <!-- StreamDataIndex index of data array provided by api call SimulyzerSystemSetSignalData for sensor simulation-->
    <!-- first index is for first Hawthorn chip channel 1 -->
    <StreamDataIndex sigName="AccX">10</StreamDataIndex>
    <!-- second index is for first Hawthorn chip channel 2 -->
    <StreamDataIndex sigName="AccY">11</StreamDataIndex>
    <!-- third index is for second Hawthorn chip channel 1 -->
    <StreamDataIndex sigName="AccX_1">12</StreamDataIndex>
  </Sensor>
</Simulyzer>
```

传感器类型

Simulyzer盒的序列号

流启动定义

spf文件的位置/名称

1. 传感器信号“[名称]”>Signallspur<
2. 传感器信号“[名称]”>Signallspur<
3. 传感器信号“[名称]”>Signallspur<

Sensor type:Type of sensor

SPI Simulyzer盒的序列号

streamstartpinpolar -Definition:定义用于数据传输的起始脉冲，Simulyzer盒用作所有连接的Simulyzer盒的主脉冲。

在本例中，将脉冲发送到引脚1的所有Simulyzer盒，并从其下降沿开始数据传输。

如果此streamstart脉冲来自系统或自动，则无需定义。

SPI Simulyzer软件生成的spf文件配置文件的名称/文件名。（请参阅SPI Simulyzer网站帮助）

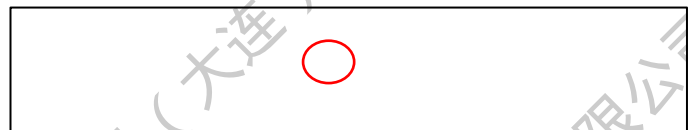
Dateiname:

Dateityp: SPI Simulyzer Project files (*.spf)

Ordner ausblenden

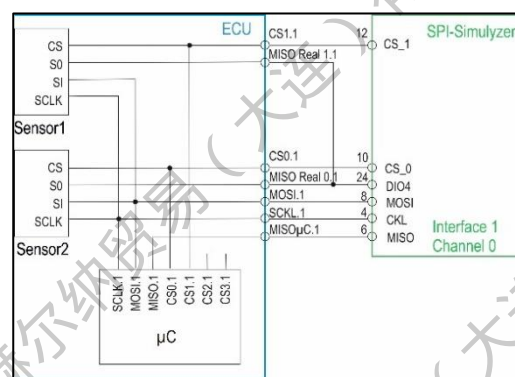
Speichern Abbrechen

从Simulyzer软件生成spf文件:菜单文件-
命令保存为...



信号定义

一个SPI-Simulyzer盒最多可以模拟4个SPI传感器。这4个SPI传感器可以分布在一个或两个接口上。



Example Wiring

信号的定义必须根据二进制数组的数据来完成。

每个信号轨迹都是由二进制数组中唯一定义的信号名称和相应的二进制数组的流索引来定义的。

HiL-Setup with PSI5, SPI and SQUIB

PSI5传感器的定义

对于每个PSI5传感器信号，必须根据以下方案进行定义。实时数据中是否存在超过此处定义的传感器信号并不重要，忽略所有未定义的信号。

在该示例中，仅使用了一个PSI5 Simulyzer盒，因此只有一个以开头和结尾的PSI5定义块定义了<Simulyzer>。每个使用的Simulyzer盒子必须有一个定义块。

(有关数据示例，请参见“2.2.1. 系统配置”第4页)

```

<Simulyzer>
  <Type> PSI5 </Type>
  <SerialNumber> 2283 </SerialNumber>
  <StreamStartPin Polarity="activeLow"> 1 </StreamStartPin>
  <ConfigurationFile> ../configs/PSI5/Example_Simulation_PSI5_Sensor.ppf </ConfigurationFile>
  <Sensor>
    <!-- InterfaceIndex = defines the physical PSI5 interface on Simulyzer Box Value 0/1 -->
    <InterfaceIndex> 0 </InterfaceIndex>
    <!-- SlotIndex 0/1/2/3 defines the timeslot on PSI5 bus -->
    <SlotIndex> 0 </SlotIndex>
    <!-- InitPhase2Data string of hex encoded data nibbles for sensor initialization, start with first data nybble on left side -->
    <InitPhase2Data> 42010714A480DF11790017C17037308C </InitPhase2Data>
    <!-- InitPhase3Data list of hex encoded data for init phase 3, :n behind value defines a repetition count of this value -->
    <InitPhase3Data> 0x1e7:16 0x0 </InitPhase3Data>
    <!-- StreamDataIndex index of data array provided by api call SimulyzerSystemSetSignalData for sensor simulation -->
    <StreamDataIndex sigName="PSI5_I0_Slot0"> 2 </StreamDataIndex>
  </Sensor>
  <Sensor>
    <InterfaceIndex> 0 </InterfaceIndex>
    <SlotIndex> 1 </SlotIndex>
    <InitPhase2Data> 42010714A480DF11790017C17037308D </InitPhase2Data>
    <InitPhase3Data> 0x1e7:16 0x0 </InitPhase3Data>
    <StreamDataIndex sigName="PSI5_I0_Slot1"> 1 </StreamDataIndex>
  </Sensor>
</Simulyzer>
</SimulyzerSystemConfigurationFile>

```

传感器类型

Simulyzer盒的序列号

位置/ ppf文件名称

信号的接口

信号的Slotindex

initphase2中信号的数据

initphase3中信号的数据

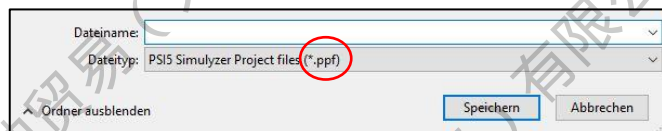
1.传感器“[名称]”的信号

Sensor type: Type of sensor

PSI5-Simulyzer盒的序列号

ppf文件的路径/文件名-它可以从Simulyzer软件生成并描述PSI5传感器。(参见PSI5-Simulyzer网站帮助)

从Simulyzer软件生成ppf文件:菜单文件-命令保存为...



接口:PSI5传感器信号发送的通道(通道0或通道1)

带PSI5、SPI和SQUIB的HiL设置

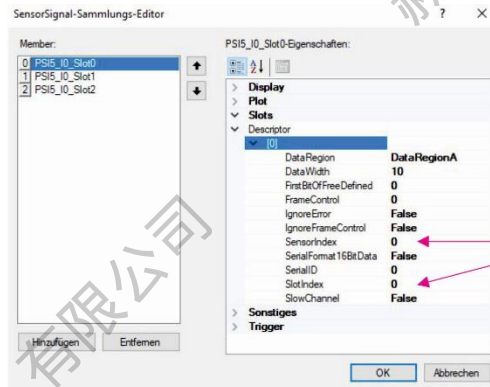
插槽索引：PSI5传感器发送数据的插槽索引。

必须相应地输入在PSI5实时系统中传输数据信号的时隙的分配。

模拟中使用的信号1

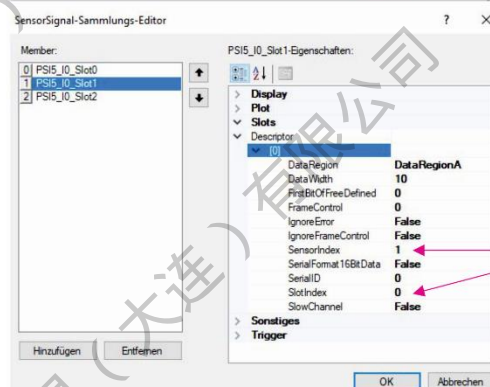
模拟中使用的信号2

模拟中未使用信号3



PSI5 Simulyzer软件
信号采集编辑器

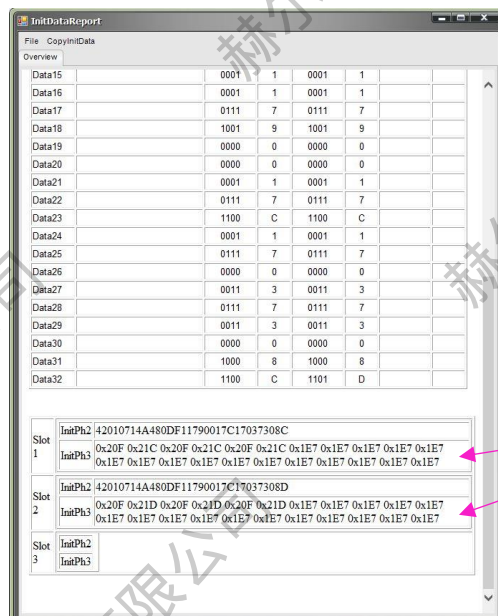
第一个信号来自时
隙0中的传感器0



2nd signal comes from
Sensor 1 in timeslot 0

根据此分配，必须在配置文件中
中进行定义。

初始化阶段2/3：传感器在初始化阶段发送的十六进制数据。数据可以通过
初始化数据报告复制并粘贴到PSI5软件中。



Slot1和Slot2的初始化数据的十六
进制值

用于复制/粘贴传输

流数据索引（信号数据定义）

根据二进制数据阵列中定义的信号数量和信号名称。每个信号轨道由

- 唯一定义的信号名称和
- 定义的二进制数据阵列的1位信号索引。

带PSI5、SPI和SQUIB的HiL设置

2.2.2. PPF文件

ppf文件包含所有特性PSI5总线数据（类型、时隙定义、数据长度、数据计数等）。

该文件可以由PSI5 Simulyzer软件生成。

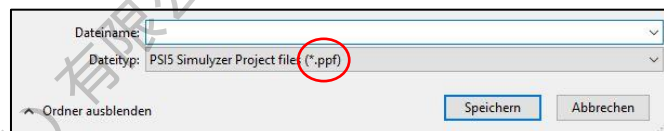
每个使用的PSI5 Simulyzer盒子必须有一个文件

ppf文件示例：

```
<?xml version="1.0" encoding="utf-8"?>
<PSI5SimulyzerProjectFile>
  <Version>1.0</Version>
  <CultureInfo>de-DE</CultureInfo>
  <SensorDataVersion>2</SensorDataVersion>
  <Template>False</Template>
  <ApplicationSettings>
    <CommonTriggerConfig>
      <RefSigTrigger0>0</RefSigTrigger0>
      <RefSigTrigger1>0</RefSigTrigger1>
      <RefSigTrigger2>0</RefSigTrigger2>
      <RefSigTrigger3>0</RefSigTrigger3>
      <Mask>5</Mask>
    </CommonTriggerConfig>
    <UseModel>2</UseModel>
    <ImportedDataMode>0</ImportedDataMode>
    <StreamStartMode>2</StreamStartMode>
    <StreamReplayCount>1</StreamReplayCount>
    <StreamReplayTime>100</StreamReplayTime>
  </ApplicationSettings>
</PSI5SimulyzerProjectFile>
```

从Simulyzer软件生成ppf文件：

菜单文件-命令保存为...



2.2.3. SPF文件

该ppf file包含所有特性PSI5总线数据（波特率、CS分配、数据计数等）。

该文件可以由PSI5 Simulyzer软件生成。

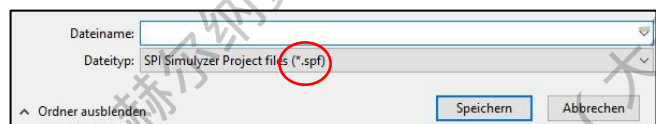
每个使用的SPI Simulyzer盒必须有一个文件。

spf文件示例：

```
<?xml version="1.0" encoding="utf-8"?>
<SPISimulyzerProjectFile>
  <Version>1.1</Version>
  <CultureInfo>de-DE</CultureInfo>
  <SensorDataVersion>2</SensorDataVersion>
  <Template>False</Template>
  <CanConfig>
    <rxontx>0</rxontx>
    <btr0>15</btr0>
    <btr1>0</btr1>
    <cdr>128</cdr>
    <audRate>1000</audRate>
  </CanConfig>
</SPISimulyzerProjectFile>
```

从Simulyzer软件生成spf文件：

菜单文件-命令保存为...



带PSI5、SPI和SQUIB的HiL设置

2.2.4. 工艺流程控制程序

该程序控制整个HiL测试系统并调节整个测量过程。
可能的编程语言：API：ANSI-C、Python、dotNET
假定具有相应编程语言的知识。
原则，API包含以下结构（例如Python）

1. 二进制数组的定义

```
#provide twenty(20) buffers with downsampled data
bufferCount = 20
sampleCnt = (c_int * bufferCount)() #hold the valid count of sample buffer
sampleBuffers = (c_void_p * bufferCount)() #array to store the buffer addresses

for n in range(0,bufferCount):
    sampleCnt[n] = 300; #set sample count
    buff = (c_double * 3000)()
    for k in range(0,2999):
        buff[k] = (n + 1) * 10
    sampleBuffers[n] = cast(pointer(buff),c_void_p) #store buffer address
    buff[sampleCnt[n] - 1] = 0 #set stream data back to zero
```

2. 加载Simulyzer驱动

```
#load the simulyzer library
os.chdir("../tmp")
hnd = cdll.LoadLibrary("../Simulyzer.dll")
print hnd
```

3. 生成测量系统并将其链接到system configuration.xml

```
#create a system handle from system configuration file
syshnd = c_void_p()
sekRetVal = hnd.StartLogging("simu.log",c_int(LoggingWhat[TRACE_APICALLS]+LoggingWhat[TRACE_ERROR]),c_int(LoggingHow[MODE_REOPEN]+LoggingHow[MODE_CONSOLE_OUT]))
print "StartLogging return code: " + str(sekRetVal)

sekRetVal = hnd.SimulyzerSystemCreate(byref(syshnd), "../ExampleSystemConfig.xml")
print "SimulyzerSystemCreate return code: " + str(sekRetVal)
#read back the actual count of initialized simulyzer devices (only for testing)
devCount = c_int(10)
devHnd = (c_void_p * 10)()
hnd.SimulyzerSystemGetDevices(syshnd,byref(devCount),pointer(devHnd))
print devCount
sekRetVal = hnd.SimulyzerSystemSetStreamEndCallback(syshnd,callback_cb,cb_func)
print "SimulyzerSystemSetStreamEndCallback return code: " + str(sekRetVal)
```

4. 登录-检查正确的系统设置和各个组件的准备情况。

5. 测量系统启动命令/触发脉冲启动

6. 测量系统的停止-停止命令

```
#set the data for streaming
hnd.SimulyzerSystemSetSignalData(syshnd,bufferCount,pointer(sampleCnt),pointer(sampleBuffers),1)
#set trigger signal
raw_input("return to continue")
rdy_flag.clear()
hnd.SimulyzerSystemStartStream(syshnd)
rdy_flag.wait(2.0)
print rdy_flag.isSet()
```

个别扩展由用户决定，可以不受限制地执行，鉴于系统进程可以执行。

带PSI5、SPI和SQUIB的HiL设置

示例完整API (Python)

```
from ctypes import *
import os
from SimulizerConstants import *
import threading

rdy_flag = threading.Event()

def py_callback(v_p):
    print "Stream complete"
    rdy_flag.set()
    return

CMPFUNC = WINFUNCTYPE(None, c_void_p)
cb_func = CMPFUNC(py_callback)
callback_ctx = c_void_p()

#provide twenty(20) buffers with downsampled data
bufferCount = 20
sampleCnt = (c_int * bufferCount)() #hold the valid count of sample buffer
sampleBuffers = (c_void_p * bufferCount)() #array to store the buffer addresses

for n in range(0, bufferCount):
    sampleCnt[n] = 300; #set sample count
    buff = (c_double * 3000)()
    for k in range(0, 2999):
        buff[k] = (n + 1) * 10
    sampleBuffers[n] = cast(pointer(buff), c_void_p) #store buffer address
    buff[sampleCnt[n] - 1] = 0 #set stream data back to zero

#load the simulizer library
os.chdir("../tmp")
hnd = cdll.LoadLibrary("../Simulizer.dll")
print hnd

#create a system handle from system configuration file
syshnd = c_void_p()
sekRetVal = hnd.StartLogging("simu.log", c_int(LoggingWhat[TRACE_APICALLS] + LoggingWhat[TRACE_ERROR]), c_int(LoggingHow[MODE_REOPEN] + LoggingHow[MODE_CONSOLE_OUT]))
print "StartLogging return code: " + str(sekRetVal)

sekRetVal = hnd.SimulizerSystemCreate(byref(syshnd), "../ExampleSystemConfig.xml")
print "SimulizerSystemCreate return code: " + str(sekRetVal)
#read back the actual count of initialized simulizer devices (only for testing)
devCount = c_int(10)
devHnd = (c_void_p * 10)()
hnd.SimulizerSystemGetDevices(syshnd, byref(devCount), pointer(devHnd))
print devCount
sekRetVal = hnd.SimulizerSystemSetStreamEndCallback(syshnd, callback_ctx, cb_func)
print "SimulizerSystemSetStreamEndCallback return code: " + str(sekRetVal)

#set the data for streaming
hnd.SimulizerSystemSetSignalData(syshnd, bufferCount, pointer(sampleCnt), pointer(sampleBuffers), 1)
#set trigger signal
raw_input("return to continue")
rdy_flag.clear()
hnd.SimulizerSystemStartStream(syshnd)
rdy_flag.wait(2.0)
print rdy_flag.isSet()

#cleanup all simulizer stuff
hnd.SimulizerSystemDestroy(syshnd)
```

带PSI5、SPI和SQUIB的HiL设置

2.3.通过CSV文件的系统模拟数据结构

以下示例系统用于描述：

1x PSI5-Simulyzer-Box 序列号.: 2283 通道0已使用 通道1未使用 1.在插槽0中发送信号 初始数据： 阶段2: 42010714A480DF11790017C17037308C 阶段3: 0x1e7:16 0x0 实时数据信号跟踪: "PSI5_IO_Slot0 Index 1" 2.信号在插槽1中发送 初始数据： 阶段 2: 42010714A480DF11790017C17037308D 阶段 3: 0x1e7:16 0x0 实时数据信号跟踪: "PSI5_IO_Slot1 Index 1"	1x SPI-Simulyzer-Box 序列号.: 162 通道0已使用 通道1未使用 1.传感器信号 AccX信号 数据传输开始时出现下降沿 实时数据信号跟踪: "AccX Index 10" 2.传感器信号 AccY信号 实时数据信号跟踪: "AccY Index 11" 3.传感器信号 AccX_1信号 实时数据信号跟踪: "AccX_1 Index 12"
---	---

2.3.1.CSV数据的系统配置文件

*xml格式的必选文件，用于在以下方面配置HiL测试系统：

- Simulyzer许可证验证
- 用于数据准备的解码器文件参考SPI信号
- 通过spf文件参考的SPI传感器信息（SPI Simulyzer软件生成的SPI配置数据）
- 根据仿真数据定义SPI信号名称。
- PSI5总线*csv文件参考
- 通过ppf文件的PSI5传感器信息（PSI5 Simulyzer软件生成的PSI5配置数据）
- 用于数据准备的解码器文件参考PSI5信号
- 通过ppf文件的PSI5传感器信息（PSI5 Simulyzer软件生成的PSI5配置数据）
- 根据模拟数据定义PSI5信号名称。

<pre> <?xml version="1.0" encoding="UTF-8"?> <SimulyzerSystemConfigurationFile> <!-- --> <BasePath>./</BasePath> <LicenseFile>./seskionLicense.xml</LicenseFile> <Simulyzer> <Type>SPI</Type> <SerialNumber>4711</SerialNumber> <CSV_DecoderFile>./configs/SPI/Example_Simulation_SPI_CSV_Description.xml</CSV_DecoderFile> <ConfigurationFile>./configs/SPI/Example_Simulation_SPI_Sensor.spf</ConfigurationFile> <Sensor> <!-- StreamDataIndex index of data array provided by api call SimulyzerSystemSetSignalData for SPI simulation --> <!-- first index is for first Hawthorn chip channel 1 --> <StreamDataIndex sigName="AccX">10</StreamDataIndex> <!-- second index is for first Hawthorn chip channel 2 --> <StreamDataIndex sigName="AccY">11</StreamDataIndex> <!-- third index is for second Hawthorn chip channel 1 --> <StreamDataIndex sigName="AccX_1">12</StreamDataIndex> </Sensor> </Simulyzer> <Simulyzer> <Type>PSI5</Type> <SerialNumber>2283</SerialNumber> <StreamStartPin Polarity="activeLow">1</StreamStartPin> <CSV_DecoderFile>./configs/PSI5/Example_Simulation_PSI5_0_0_CSV_Description.xml</CSV_DecoderFile> <CSV_DecoderFile>./configs/PSI5/Example_Simulation_PSI5_0_1_CSV_Description.xml</CSV_DecoderFile> <ConfigurationFile>./configs/PSI5/Example_Simulation_PSI5_Sensor.ppf</ConfigurationFile> <Sensor> <!-- InterfaceIndex = defines the physical PSI5 interface on Simulyzer Box Value 0/1 --> <InterfaceIndex>0</InterfaceIndex> <!-- SlotIndex 0/1/2/3 defines the timeslot on PSI5 bus --> <SlotIndex>0</SlotIndex> <!-- InitPhase2Data string of hex encoded data nibbles for sensor initialization, start with first data nibble on left side --> <InitPhase2Data>42010714A480DF11790017C17037308C</InitPhase2Data> <!-- InitPhase3Data list of hex encoded data for init phase 3, :n behind value defines a repetition count of this value --> <InitPhase3Data>0x1e7:16 0x0</InitPhase3Data> <!-- StreamDataIndex index of data array provided by api call SimulyzerSystemSetSignalData for sensor simulation --> <StreamDataIndex sigName="PSI5_IO_Slot0">2</StreamDataIndex> </Sensor> <Sensor> <InterfaceIndex>0</InterfaceIndex> <SlotIndex>1</SlotIndex> <InitPhase2Data>42010714A480DF11790017C17037308D</InitPhase2Data> <InitPhase3Data>0x1e7:16 0x0</InitPhase3Data> <StreamDataIndex sigName="PSI5_IO_Slot1">1</StreamDataIndex> </Sensor> </Simulyzer> </SimulyzerSystemConfigurationFile> </pre>	Seskion license文件的位置 PSI5传感器的定义 Seskion许可证文件的位置 PSI5传感器信号1的定义 PSI5传感器信号2的定义
--	--

带PSI5、SPI和SQUIB的HiL设置

SPI传感器的定义

对于每个SPI传感器信号，必须根据以下方案进行定义。模拟数据中是否存在超过此处定义的传感器信号并不重要。忽略所有未定义的信号。

在该示例中，只使用了一个SPI Simulyzer框，因此只定义了一个以< Simulyzer > 开头和结尾的SPI定义块。每个使用的Simulyzer盒子都必须有一个定义块



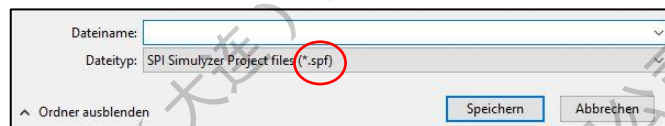
Sensor type: Type of sensor

SPI Simulyzer盒的序列号。

路径/文件名SPI解码器csv文件

SPI Simulyzer软件生成的spf文件配置文件的路径/文件名。（请参阅SPI Simulyzer网站帮助）

从Simulyzer软件创建spf文件：
菜单文件命令另存为。。



信号定义

根据SPI传感器的信号数量和实时数据.csv中定义的信号名称。
每个信号轨道由

- 唯一的、定义的信号名称和
- 2位信号索引
 - 1.index=通道0或1（接口1或2）
 - 2.索引=信号的芯片选择行

带PSI5、SPI和SQUIB的HiL设置

SPI解码器文件

解码文件引用并定义来自仿真数据文件的SPI信号。

```
<?xml version="1.0" encoding="UTF-8"?>
<ImportDescriptionFile>
  <Separator></Separator>
  <MapDefinitionLine>1</MapDefinitionLine>
  <FirstDataLine>2</FirstDataLine>
  <Column>
    <!-- ASG51 Channel 1(Interface 0) 16LSB/g 9.81 m/s*s = 1g 16LSB/(9.81 m/s*s) 1.63LSB/(m/s*s)-->
    <Scale>1.63</Scale>
    <Name>ACCX</Name>
    <Format>double</Format>
  </Column>
  <Column>
    <!-- ASG51 Channel 1(Interface 0) 16LSB/g 9.81 m/s*s = 1g 16LSB/(9.81 m/s*s) 1.63LSB/(m/s*s)-->
    <Scale>1.63</Scale>
    <Name>ACCY</Name>
    <Format>double</Format>
  </Column>
  <!-- AccX -->
  <DataStream>
    <Size>32</Size>
  </DataStream>
  <!-- AccY -->
  <DataStream>
    <Size>32</Size>
    <RefColumn Align="0" Pos="0" Width="16">ACCY</RefColumn>
  </DataStream>
  <!-- AccX_1 -->
  <DataStream>
    <Size>32</Size>
    <RefColumn Align="0" Pos="0" Width="16">ACCX</RefColumn>
  </DataStream>
  <RequiredColumnCount>2</RequiredColumnCount>
</ImportDescriptionFile>
```

PSI5传感器定义

对于每个PSI5传感器信号，必须按照以下方案进行定义。(示例数据见第11页“2.3 CSV数据的系统配置文件”)

```
<Simulizer>
  <Type>PSI5</Type>
  <SerialNumber>2283</SerialNumber>
  <StreamStartPin Polarity="activeLow">1</StreamStartPin>
  <CSV_DecoderFile>../configs/PSI5/Example_Simulation_PSI5_0_0_CSV_Description.xml</CSV_DecoderFile>
  <CSV_DecoderFile>../configs/PSI5/Example_Simulation_PSI5_0_1_CSV_Description.xml</CSV_DecoderFile>
  <ConfigurationFile>../configs/PSI5/Example_Simulation_PSI5_Sensor.ppf</ConfigurationFile>
</Simulizer>

PSI5 Sensor 0
  <!-- InterfaceIndex = defines the physical PSI5 interface on Simulizer Box Value 0/1 -->
  <InterfaceIndex>0</InterfaceIndex>
  <!-- SlotIndex 0/1/2/3 defines the timeslot on PSI5 bus-->
  <SlotIndex>0</SlotIndex>
  <!-- InitPhase2Data string of hex encoded data nibbles for sensor initialization, start with first data nybble on left side-->
  <InitPhase2Data>42010714A480DF11790017C17037308C</InitPhase2Data>
  <!-- InitPhase3Data list of hex encoded data for init phase 3, :n behind value defines a repetition count of this value-->
  <InitPhase3Data>0x1e7:16 0x0</InitPhase3Data>
  <!-- StreamDataIndex index of data array provided by api call SimulizerSystemSetSignalData for sensor simulation-->
  <StreamDataIndex sigName="PSI5_I0_Slot0">2</StreamDataIndex>
</Sensor>

PSI5 Sensor 1
  <InterfaceIndex>0</InterfaceIndex>
  <SlotIndex>1</SlotIndex>
  <InitPhase2Data>42010714A480DF11790017C17037308D</InitPhase2Data>
  <InitPhase3Data>0x1e7:16 0x0</InitPhase3Data>
  <StreamDataIndex sigName="PSI5_I0_Slot1">1</StreamDataIndex>
</Sensor>
```

带PSI5、SPI和SQUIB的HiL设置

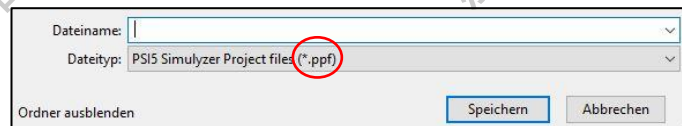
Sensor type: Type of sensor

PSI5-Simulyzer盒的序列号。

StreamstartPinPolarityDefinition (流启动极性定义) : 数据传输的启动脉冲的定义, 该脉冲由模拟器盒用作所有连接的模拟器盒的主机。在该示例中, 脉冲被发送到引脚1处的所有Simulyzer盒, 并且数据传输从其下降沿开始。如果此流启动脉冲来自系统或自动, 则无需

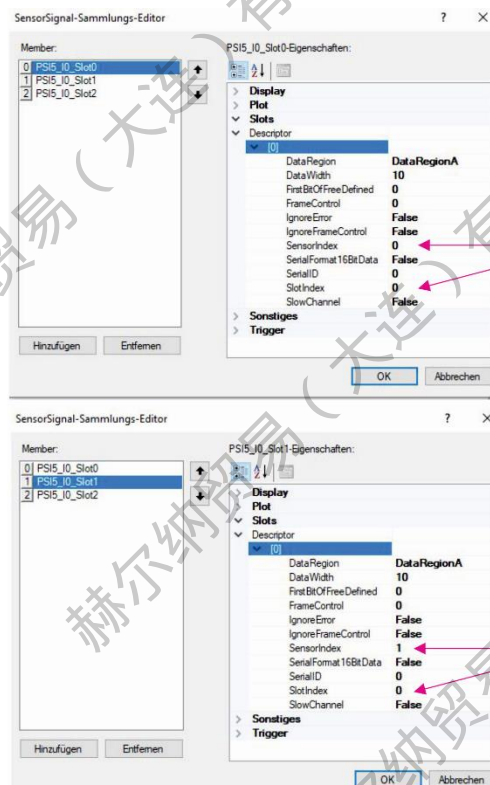
ppf文件的路径/文件名, 该文件可从Simulyzer软件生成, 并描述PSI5传感器
(请参阅PSI5 Simulyzer网站帮助)

创建一个ppf文件从模拟器-软件:
菜单文件-命令保存为...



接口: PSI5传感器信号发送的通道(通道0或通道1)。

槽位索引: PSI5传感器信号发送所在的槽位索引。发送数据信号的时隙的分配是在PSI5仿真系统中完成的, 必须相应地输入。

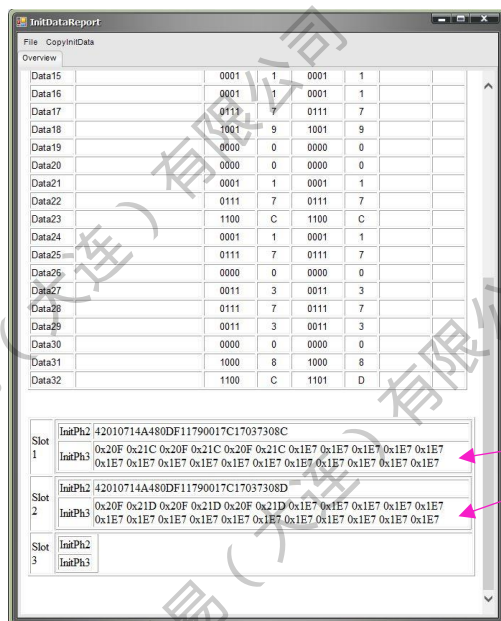


psi5 - simulyzer -软件信号采集编辑器

根据此分配,
必须在配置文件中定义

带PSI5、SPI和SQUIB的HiL设置

初始化阶段2/3：传感器在初始化阶段发送的十六进制数据。
数据可以通过初始化数据报告复制并粘贴到PSI5软件中。



Slot	InitPh2	InitPh3
Slot 1	42010714A480DF11790017C17037308C	0x20F 0x21C 0x20F 0x21C 0x20F 0x21C 0x1E7 0x1E7 0x1E7 0x1E7 0x1E7 0x1E7 0x1E7 0x1E7 0x1E7 0x1E7
Slot 2	42010714A480DF11790017C17037308D	0x20F 0x21D 0x20F 0x21D 0x20F 0x21D 0x1E7 0x1E7 0x1E7 0x1E7 0x1E7 0x1E7 0x1E7 0x1E7 0x1E7 0x1E7
Slot 3		

init数据的十六进制值
用于插槽1和插槽2
用于复制/粘贴传输

Streamdataindex(信号数据定义)

根据CSV中定义的信号数量和信号名称。
每个信号轨道由

- 唯一定义的信号名称和
- 1位信号索引

PSI5解码器文件

PSI5解码器文件参考并定义模拟数据文件中的PSI5信号

```
<?xml version="1.0" encoding="UTF-8"?>
<ImportDescriptionFile>
  <Separator></Separator>
  <MapDefinitionLine>1</MapDefinitionLine>
  <FirstDataLine>2</FirstDataLine>
  <Column>
    <Scale>1.63</Scale>
    <Name>ACCY</Name>
    <Format>double</Format>
  </Column>
  <Column>
    <Scale>1.63</Scale>
    <Name>ACCX</Name>
    <Format>double</Format>
  </Column>
  <DataStream>
    <Size>32</Size>
    <RefColumn Align="0" Pos="0" Width="16">ACCX</RefColumn>
  </DataStream>
  <DataStream>
    <Size>32</Size>
  </DataStream>
  <DataStream>
    <Size>32</Size>
    <RefColumn Align="0" Pos="0" Width="16">ACCY</RefColumn>
  </DataStream>
  <RequiredColumnCount>3</RequiredColumnCount>
</ImportDescriptionFile>
```

分隔符

信号线名称

信号数据起始行

数据值的比例因子

信号标识

数据值的数字格式

数据值的位数

数据值列数

带PSI5、SPI和SQUIB的HiL设置

2.3.2. PPF文件

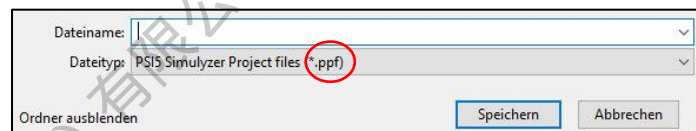
ppf文件包含所有特征数据（类型、时隙定义、数据长度、数据计数等）。
该文件可以由PSI5 Simulyzer软件生成。
每个使用的PSI5 Simulyzer盒子必须有一个文件。

ppf文件示例：

```
<?xml version="1.0" encoding="utf-8"?>
<PSI5SimulyzerProjectFile>
  <Version>1.0</Version>
  <CultureInfo>de-DE</CultureInfo>
  <SensorDataVersion>2</SensorDataVersion>
  <Template>False</Template>
  <ApplicationSettings>
    <CommonTriggerConfig>
      <RefSigTrigger0>0</RefSigTrigger0>
      <RefSigTrigger1>0</RefSigTrigger1>
      <RefSigTrigger2>0</RefSigTrigger2>
      <RefSigTrigger3>0</RefSigTrigger3>
      <Mask>5</Mask>
    </CommonTriggerConfig>
    <UseModel>2</UseModel>
    <ImportedDataMode>0</ImportedDataMode>
    <StreamStartMode>2</StreamStartMode>
    <StreamReplayCount>1</StreamReplayCount>
    <StreamReplayTime>100</StreamReplayTime>
  </ApplicationSettings>
</PSI5SimulyzerProjectFile>
```

从Simulyzer软件创建ppf文件：
菜单文件-命令保存为

...



2.3.3. SPf文件

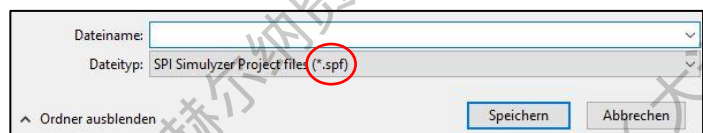
spf文件包含所有特性PSI5总线数据（波特率、CS分配、数据计数等）。
该文件可以由PSI5 Simulyzer软件生成。
每个使用的SPI Simulyzer盒子必须有一个文件。

spf文件示例：

```
<?xml version="1.0" encoding="utf-8"?>
<SPISimulyzerProjectFile>
  <Version>1.1</Version>
  <CultureInfo>de-DE</CultureInfo>
  <SensorDataVersion>2</SensorDataVersion>
  <Template>False</Template>
  <CanConfig>
    <rxontx>0</rxontx>
    <btr0>15</btr0>
    <btr1>0</btr1>
    <cdr>128</cdr>
    <audRate>1000
  </CanConfig>
</SPISimulyzerProjectFile>
```

从Simulyzer创建anspf文件
软件：菜单文件命令另存为

...



带PSI5、SPI和SQUIB的HiL设置

2.3.4. 工艺流程控制程序

该程序控制整个HiL测试系统并调节整个测量过程。

可能的编程语言：API：ANSI-C、Python、dotNET
需要具备相应的编程语言知识。
原则上，API包含以下结构（例如Python）

- 1.加载仿真器驱动程序
- 2.测量系统的生成和与系统configuration.xml的联动
- 3.登录-检查正确的系统配置和单个组件的准备情况
- 4.测量系统启动命令/触发脉冲启动
- 5.测量系统的停止-停止命令

```
from ctypes import *
import os
from SimulizerConstants import *
import threading

rdy_flag = threading.Event()

def py_callback(v_p):
    print "Stream complete"
    rdy_flag.set()
    return

CMPFUNC = WINFUNCTYPE(None, c_void_p)
cb_func = CMPFUNC(py_callback)
callback_ctx = c_void_p()

#load the simulizer library
os.chdir("../tmp")
hnd = cdll.LoadLibrary("../Simulizer.dll")
print hnd

#create a system handle from system configuration file
syshnd = c_void_p()
sekRetVal = hnd.StartLogging("simu log" c_int(LoggingWhat[TRACE_APICALLS]+LoggingWhat[TRACE_ERROR]), c_int(LoggingHow[MODE_REOPEN]+LoggingHow[MODE_CONSOLE_OUT]))
print "Start Logging return code: " + str(sekRetVal)

sekRetVal = hnd.SimulizerSystemCreate(byref(syshnd), ".\\ExampleSystemConfigCSV\\import.xml")
print "SimulizerSystemCreate return code: " + str(sekRetVal)
#read back the actual count of initialized simulizer devices (only for testing)
devCount = c_int(10)
devHnd = (c_void_p * 10)()
hnd.SimulizerSystemGetDevices(syshnd, byref(devCount), pointer(devHnd))
print devCount
sekRetVal = hnd.SimulizerSystemSetStreamEndCallback(syshnd, callback_ctx, cb_func)
print "SimulizerSystemSetStreamEndCallback return code: " + str(sekRetVal)

#set the data for streaming
hnd.SimulizerSystemSetSignalDataFromCSV(syshnd, ".\\data\\crash1.csv", 1)
#set trigger signal
raw_input("return to continue")
rdy_flag.clear()
hnd.SimulizerSystemStartStream(syshnd)
rdy_flag.wait(2.0)
print rdy_flag.isSet()

#cleanup all simulizer stuff
hnd.SimulizerSystemDestroy(syshnd)
```

生成测量系统

加载Simulizer驱动程序

配置文件.xml的路径

测量系统的启动
触发脉冲的启动

模拟数据*.csv文件路径

单独的扩展由用户决定，并且可以在没有系统过程限制的情况下进行。

进一步的资料来源和教程