

=====
Release Notes for IVI-C Driver V2021.5.1
=====

Fixed issues

None this release

New Features

None this release

Note IVI-C is not supported for MS46131A

Note the IVI-C driver is being deprecated and will no longer be actively supported with new functionality.

Users are strongly encouraged to use the SCPI command interface for remote programming of ShockLine VNAs.

Please contact Anritsu at <https://www.anritsu.com/test-measurement/contact-us> for any questions or additional support.

=====
Release Notes for IVI-C Driver V2020.10.1
=====

Fixed issues

None this release

New Features

None this release

Note IVI-C is not supported for MS46131A

=====
Release Notes for IVI-C Driver V2020.7.3
=====

Fixed issues

None this release

New Features

None this release

Note IVI-C is not supported for MS46131A

=====

Release Notes for IVI-C Driver V2020.7.2

=====

Fixed issues

None this release

New Features

None this release

Note IVI-C is not supported for MS46131A.

=====

Release Notes for IVI-C Driver V2020.7.1

=====

Fixed issues

None this release

New Features

None this release
Note IVI-C is not supported for MS46131A.

=====
Release Notes for IVI-C Driver V2019.12.3
=====

Fixed issues

None this release

New Features

None this release
Note IVI-C is not supported for MS46131A.

=====
Release Notes for IVI-C Driver V2019.12.1
=====

Fixed issues

None this release

New Features

None this release
Note IVI-C is not supported for MS46131A.

=====

Release Notes for IVI-C Driver V2019.7.1

=====

Fixed issues

1. [SLD-6710] Fixed IVI-C 1-port SOL calibration when used with 3 MS46121 VNAs.

New Features

1. Added IVI-C Commands for High Fidelity mode for 500s model with frequency option 20, 40, 43.

=====

Release Notes for IVI-C Driver V2019.3.2

=====

Fixed issues

1. [SLD-5474] Fixed an issue with IVI function GetAttributeViUInt32 not working correctly with multiple channels.

New Features

1. [SLD-3790] Added IVI-C Commands for actual X1 and X2 points to indicate actual x values use for limit lines.
2. [SLD-4229] Added IVI-C support for Marker Statistics.

=====

Release Notes for IVI-C Driver V2018.9.4

=====

None this release: simply rolling the revision to V2018.9.4 in order to keep it consistent with the ShockLine SW package revision, V2018.9.4.

=====
Release Notes for IVI-C Driver V2018.9.2
=====

None this release: simply rolling the revision to V2018.9.2 in order to keep it consistent with the ShockLine SW package revision, V2018.9.2.

=====
Release Notes for IVI-C Driver V2018.9.1
=====

Fixed issues

1. [SLD-3878] Added ability to do IVI-C based scalar transmission calibration and response on MS46121Bs with option 21.

New Features

1. [SLD-534] Differential impedance transformation control added to the ShockLine software.
2. [SLD-4453] Added IVI-C Commands for turning off unused source when using MS46121B in scalar transmission configuration.

=====
Release Notes for IVI-C Driver V2018.2.1
=====

Fixed issues

1. Corrected failure in Anritsu ShockLine VNA-SOLT Calibration.vi LabView example. (SLD-3088)
2. Corrected autosense issue with Smartcal (SLD-3888)
3. Updated installer to copy both versions (32/64 bit) of C++ libs in both install locations (32 and 64 bit)

New Features

None this release

=====
Release Notes for IVI-C Driver V2017.9.1
=====

Fixed issues

1. ChannelMeasurementSetMixedModeOneDiffPairResponse IVI-C function now works in MATLAB. (SLD-3123)
2. Improved ripple limit functionality in IVI-C. (SLD-3121)
3. Fix for IVI C ChannelSaveState fails to save calibration file directly on C drive. (SLD-3115)
4. Corrected T/F Boolean issue with Set CW Mode IVI command. (SLD-2776)
5. Fixed ANVNA.GetAttributeViString IVI-C function in C#. (SLD-3124)
6. LabView Fix for user can open this front panel without encountering errors. (SLD-3406)
7. Update bundled version of IVI-C Shared Components to 2.4.2 (SLD-3433)
8. Fix for ANVNA_GetTimeDomainDistanceValues in MatLab wrapper (SLD-3512)

New Features

None this release

=====
Release Notes for IVI-C Driver V2017.6.1
=====

Fixed issues

1. Fixes for GetTimeDomainDistanceValues and GateValues (SLD-2558)
2. Fixes for ANVNA_ChannelMeasurementGetMixedModeOneDiffPairResponse and ANVNA_ChannelMeasurementSetMixedModeOneDiffPairResponse commands (SLD-2552)
3. Fixes for several limit functions in Python wrapper (SLD-2559)
4. Fix for GetAttributeString() in C# wrapper (SLD-2774)
5. Fix for ANVNA_SetupCalibration() (SLD-3043)

New Features

1. GetAttributeViString now supports GetSoftwareVersion (SLD-2748, SLD-2749)
2. Added new IVI-C Commands to support Hybrid Cal (SLD-2770)
3. GetAttributeViBoolean now supports ANVNA_ATTR_CHANNEL_SPUR_REDUCTION (SLD-2790)

=====

Release Notes for IVI-C Driver v1.12.0002

=====

Fixed issues

1. Labview IVI-C examples do not function (SLD-2297)

New Features

1. GET functions use new ReferenceChannel instead of ActiveChannel (SLD-2095)

=====

Release Notes for IVI-C Driver v1.12

=====

Fixed issues

1. IVI-C SetManualCalKit types do not match MT elements (SLD-1230)

New features

1. Add functions to manage ripple limits:

```
ANVNA_API ViStatus ANVNA_SetRippleLimitTestingOnOff (ViSession Vi, ViConstString RepCapIdentifier, ViBoolean onOff);
ANVNA_API ViStatus ANVNA_GetRippleLimitTestingOnOff (ViSession Vi, ViConstString RepCapIdentifier, ViPBoolean onOff);
ANVNA_API ViStatus ANVNA_SetRippleLimitTestResultSign (ViSession Vi, ViBoolean onOff);
ANVNA_API ViStatus ANVNA_GetRippleLimitTestResultSign (ViSession Vi, ViPBoolean onOff);
ANVNA_API ViStatus ANVNA_AddDefaultRippleLimitSegment (ViSession Vi, ViConstString RepCapIdentifier);
ANVNA_API ViStatus ANVNA_GetRippleLimitsCount (ViSession Vi, ViConstString RepCapIdentifier, ViPUInt32 readCount);
ANVNA_API ViStatus ANVNA_DeleteRippleLimitSegment (ViSession Vi, ViConstString RepCapIdentifier);
ANVNA_API ViStatus ANVNA_ClearAllRippleLimits (ViSession Vi, ViConstString RepCapIdentifier);
ANVNA_API ViStatus ANVNA_AddRippleLimitSegment (ViSession Vi, ViConstString RepCapIdentifier, ViReal64 x1, ViReal64 x2, ViReal32 ripple);
ANVNA_API ViStatus ANVNA_SetRippleLimitValues (ViSession Vi, ViConstString RepCapIdentifier, ViUInt32 segmentNum, ViReal64 x1, ViReal64 x2, ViReal32 ripple);
ANVNA_API ViStatus ANVNA_GetRippleLimitValues (ViSession Vi, ViConstString RepCapIdentifier, ViUInt32 segmentNum, ViPReal64 x1, ViPReal64 x2, ViPReal32 ripple);
ANVNA_API ViStatus ANVNA_DeleteRippleLimitSegmentAt (ViSession Vi, ViConstString RepCapIdentifier, ViUInt32 segmentNum);
ANVNA_API ViStatus ANVNA_IsRippleLimitTestPass (ViSession Vi, ViConstString RepCapIdentifier, ViUInt32 segmentNum, ViPBoolean passNoPass);
ANVNA_API ViStatus ANVNA_SetRippleLimitX1Val (ViSession Vi, ViConstString RepCapIdentifier, ViUInt32 segmentNum, ViReal64 x1);
ANVNA_API ViStatus ANVNA_GetRippleLimitX1Val (ViSession Vi, ViConstString RepCapIdentifier, ViUInt32 segmentNum, ViPReal64 x1);
ANVNA_API ViStatus ANVNA_SetRippleLimitX2Val (ViSession Vi, ViConstString RepCapIdentifier, ViUInt32 segmentNum, ViReal64 x2);
ANVNA_API ViStatus ANVNA_GetRippleLimitX2Val (ViSession Vi, ViConstString RepCapIdentifier, ViUInt32 segmentNum, ViPReal64 x2);
ANVNA_API ViStatus ANVNA_SetRippleLimitRippleVal (ViSession Vi, ViConstString RepCapIdentifier, ViUInt32 segmentNum, ViReal32 ripple);
ANVNA_API ViStatus ANVNA_GetRippleLimitRippleVal (ViSession Vi, ViConstString RepCapIdentifier, ViUInt32 segmentNum, ViPReal32 ripple);
ANVNA_API ViStatus ANVNA_GetRippleLimitUpperLowerValues (ViSession Vi, ViConstString RepCapIdentifier, ViUInt32 segmentNum, ViPReal32 upper, ViPReal32 lower);
ANVNA_API ViStatus ANVNA_GetRippleLimitFailPointsBuffer (ViSession Vi, ViConstString RepCapIdentifier, ViPReal64 dataUpper, ViUInt32 dataSizeUpper, ViPUInt32
readCountUpper, ViPReal64 dataLower, ViUInt32 dataSizeLower, ViPUInt32 readCountLower);
ANVNA_API ViStatus ANVNA_SetRippleLimitLineActive (ViSession Vi, ViConstString RepCapIdentifier, ViUInt32 segmentNum, ViBoolean onOff);
ANVNA_API ViStatus ANVNA_GetRippleLimitLineActive (ViSession Vi, ViConstString RepCapIdentifier, ViUInt32 segmentNum, ViPBoolean onOff);
ANVNA_API ViStatus ANVNA_SetRippleLimitLinesOnOff (ViSession Vi, ViConstString RepCapIdentifier, ViBoolean onOff);
ANVNA_API ViStatus ANVNA_GetRippleLimitLinesOnOff (ViSession Vi, ViConstString RepCapIdentifier, ViPBoolean onOff);
ANVNA_API ViStatus ANVNA_SetRippleLimitRippleValueFormat (ViSession Vi, ViConstString RepCapIdentifier, ViUInt32 type);
ANVNA_API ViStatus ANVNA_GetRippleLimitRippleValueFormat (ViSession Vi, ViConstString RepCapIdentifier, ViPUInt32 type);
ANVNA_API ViStatus ANVNA_GetRippleLimitRippleMeasurementValue (ViSession Vi, ViConstString RepCapIdentifier, ViUInt32 segmentNum, ViPReal64 data);
```

2. Add defines for Mixed Mode defines for One Differential Pair mode:


```
#define ANVNA_VAL_ANRITSU_VNA_MESUREMENT_RESPONSE_MIXEDMODE_ODP_SDD 0
#define ANVNA_VAL_ANRITSU_VNA_MESUREMENT_RESPONSE_MIXEDMODE_ODP_SCC 1
#define ANVNA_VAL_ANRITSU_VNA_MESUREMENT_RESPONSE_MIXEDMODE_ODP_SDC 2
#define ANVNA_VAL_ANRITSU_VNA_MESUREMENT_RESPONSE_MIXEDMODE_ODP_SCD 3
```

Compatibility issues

1. Signature changed for:

```
ANVNA_API ViStatus ANVNA_GetLowerLimitFailPointsBuffer ( ViUInt32 Vi, ViConstString RepCapIdentifier, ViInt32 dataValueXSize, ViPReal64 dataValueX, ViPInt32 dataValueXActualSize, ViInt32 dataValueYSize, ViPReal32 dataValueY, ViPInt32 dataValueYActualSize, ViInt32 dataValueFailedYSize, ViPReal32 dataValueFailedY, ViPInt32 dataValueFailedYActualSize );
```

```
ANVNA_API ViStatus ANVNA_GetUpperLimitFailPointsBuffer ( ViUInt32 Vi, ViConstString RepCapIdentifier, ViInt32 dataValueXSize, ViPReal64 dataValueX, ViPInt32 dataValueXActualSize, ViInt32 dataValueYSize, ViPReal32 dataValueY, ViPInt32 dataValueYActualSize, ViInt32 dataValueFailedYSize, ViPReal32 dataValueFailedY, ViPInt32 dataValueFailedYActualSize );
```

```
ANVNA_API ViStatus ANVNA_GetLowerTraceLowerLimitFailPointsBuffer ( ViUInt32 Vi, ViConstString RepCapIdentifier, ViInt32 dataValueXSize, ViPReal64 dataValueX, ViPInt32 dataValueXActualSize, ViInt32 dataValueYSize, ViPReal32 dataValueY, ViPInt32 dataValueYActualSize, ViInt32 dataValueFailedYSize, ViPReal32 dataValueFailedY, ViPInt32 dataValueFailedYActualSize );
```

```
ANVNA_API ViStatus ANVNA_GetLowerTraceUpperLimitFailPointsBuffer ( ViUInt32 Vi, ViConstString RepCapIdentifier, ViInt32 dataValueXSize, ViPReal64 dataValueX, ViPInt32 dataValueXActualSize, ViInt32 dataValueYSize, ViPReal32 dataValueY, ViPInt32 dataValueYActualSize, ViInt32 dataValueFailedYSize, ViPReal32 dataValueFailedY, ViPInt32 dataValueFailedYActualSize );
```

into

```
ANVNA_API ViStatus ANVNA_GetLowerLimitFailPointsBuffer ( ViUInt32 Vi, ViConstString RepCapIdentifier, ViInt32 dataValueXSize, ViPReal64 dataValueX, ViPInt32 dataValueXActualSize );
```

```
ANVNA_API ViStatus ANVNA_GetUpperLimitFailPointsBuffer ( ViUInt32 Vi, ViConstString RepCapIdentifier, ViInt32 dataValueXSize, ViPReal64 dataValueX, ViPInt32 dataValueXActualSize );
```

```
ANVNA_API ViStatus ANVNA_GetLowerTraceLowerLimitFailPointsBuffer ( ViUInt32 Vi, ViConstString RepCapIdentifier, ViInt32 dataValueXSize, ViPReal64 dataValueX, ViPInt32 dataValueXActualSize);
```

```
ANVNA_API ViStatus ANVNA_GetLowerTraceUpperLimitFailPointsBuffer ( ViUInt32 Vi, ViConstString RepCapIdentifier, ViInt32 dataValueXSize, ViPReal64 dataValueX, ViPInt32 dataValueXActualSize );
```

=====
Release Notes for IVI-C Driver v1.11.0005
=====

Fixed issues

1. Fix for Python driver
2. IVI-C SetManualCalKit types do not match MT elements (SLD-1230)

=====
Release Notes for IVI-C Driver v1.11.0004
=====

1. Add functions to manage ripple limits:

```
ANVNA_API ViStatus ANVNA_SetRippleLimitTestingOnOff (ViSession Vi, ViConstString RepCapIdentifier, ViBoolean onOff);
ANVNA_API ViStatus ANVNA_GetRippleLimitTestingOnOff (ViSession Vi, ViConstString RepCapIdentifier, ViPBoolean onOff);
ANVNA_API ViStatus ANVNA_SetRippleLimitTestResultSign (ViSession Vi, ViBoolean onOff);
ANVNA_API ViStatus ANVNA_GetRippleLimitTestResultSign (ViSession Vi, ViPBoolean onOff);
ANVNA_API ViStatus ANVNA_AddDefaultRippleLimitSegment (ViSession Vi, ViConstString RepCapIdentifier);
ANVNA_API ViStatus ANVNA_GetRippleLimitsCount (ViSession Vi, ViConstString RepCapIdentifier, ViUInt32 readCount);
ANVNA_API ViStatus ANVNA_DeleteRippleLimitSegment (ViSession Vi, ViConstString RepCapIdentifier);
ANVNA_API ViStatus ANVNA_ClearAllRippleLimits (ViSession Vi, ViConstString RepCapIdentifier);
ANVNA_API ViStatus ANVNA_AddRippleLimitSegment (ViSession Vi, ViConstString RepCapIdentifier, ViReal64 x1, ViReal64 x2, ViReal32 ripple);
ANVNA_API ViStatus ANVNA_SetRippleLimitValues (ViSession Vi, ViConstString RepCapIdentifier, ViUInt32 segmentNum, ViReal64 x1, ViReal64 x2, ViReal32 ripple);
ANVNA_API ViStatus ANVNA_GetRippleLimitValues (ViSession Vi, ViConstString RepCapIdentifier, ViUInt32 segmentNum, ViPReal64 x1, ViPReal64 x2, ViPReal32 ripple);
ANVNA_API ViStatus ANVNA_DeleteRippleLimitSegmentAt (ViSession Vi, ViConstString RepCapIdentifier, ViUInt32 segmentNum);
ANVNA_API ViStatus ANVNA_IsRippleLimitTestPass (ViSession Vi, ViConstString RepCapIdentifier, ViUInt32 segmentNum, ViPBoolean passNoPass);
ANVNA_API ViStatus ANVNA_SetRippleLimitX1Val (ViSession Vi, ViConstString RepCapIdentifier, ViUInt32 segmentNum, ViReal64 x1);
ANVNA_API ViStatus ANVNA_GetRippleLimitX1Val (ViSession Vi, ViConstString RepCapIdentifier, ViUInt32 segmentNum, ViPReal64 x1);
ANVNA_API ViStatus ANVNA_SetRippleLimitX2Val (ViSession Vi, ViConstString RepCapIdentifier, ViUInt32 segmentNum, ViReal64 x2);
```

```
ANVNA_API ViStatus ANVNA_GetRippleLimitX2Val (ViSession Vi, ViConstString RepCapIdentifier, ViUInt32 segmentNum, ViPReal64 x2);
ANVNA_API ViStatus ANVNA_SetRippleLimitRippleVal (ViSession Vi, ViConstString RepCapIdentifier, ViUInt32 segmentNum, ViReal32 ripple);
ANVNA_API ViStatus ANVNA_GetRippleLimitRippleVal (ViSession Vi, ViConstString RepCapIdentifier, ViUInt32 segmentNum, ViPReal32 ripple);
ANVNA_API ViStatus ANVNA_GetRippleLimitUpperLowerValues (ViSession Vi, ViConstString RepCapIdentifier, ViUInt32 segmentNum, ViPReal32 upper, ViPReal32 lower);
ANVNA_API ViStatus ANVNA_GetRippleLimitFailPointsBuffer (ViSession Vi, ViConstString RepCapIdentifier, ViPReal64 dataUpper, ViUInt32 dataSizeUpper, ViPUInt32
readCountUpper, ViPReal64 dataLower, ViUInt32 dataSizeLower, ViPUInt32 readCountLower);
ANVNA_API ViStatus ANVNA_SetRippleLimitLineActive (ViSession Vi, ViConstString RepCapIdentifier, ViUInt32 segmentNum, ViBoolean onOff);
ANVNA_API ViStatus ANVNA_GetRippleLimitLineActive (ViSession Vi, ViConstString RepCapIdentifier, ViUInt32 segmentNum, ViPBoolean onOff);
ANVNA_API ViStatus ANVNA_SetRippleLimitLinesOnOff (ViSession Vi, ViConstString RepCapIdentifier, ViBoolean onOff);
ANVNA_API ViStatus ANVNA_GetRippleLimitLinesOnOff (ViSession Vi, ViConstString RepCapIdentifier, ViPBoolean onOff);
ANVNA_API ViStatus ANVNA_SetRippleLimitRippleValueFormat (ViSession Vi, ViConstString RepCapIdentifier, ViUInt32 type);
ANVNA_API ViStatus ANVNA_GetRippleLimitRippleValueFormat (ViSession Vi, ViConstString RepCapIdentifier, ViPUInt32 type);
ANVNA_API ViStatus ANVNA_GetRippleLimitRippleMeasurementValue (ViSession Vi, ViConstString RepCapIdentifier, ViUInt32 segmentNum, ViPReal64 data);
```

Compatibility issues

1. Signature changed for:

```
ANVNA_API ViStatus ANVNA_GetLowerLimitFailPointsBuffer ( ViUInt32 Vi, ViConstString RepCapIdentifier, ViInt32 dataValueXSize, ViPReal64 dataValueX, ViPInt32
dataValueXActualSize, ViInt32 dataValueYSize, ViPReal32 dataValueY, ViPInt32 dataValueYActualSize, ViInt32 dataValueFailedYSize, ViPReal32 dataValueFailedY, ViPInt32
dataValueFailedYActualSize );
```

```
ANVNA_API ViStatus ANVNA_GetUpperLimitFailPointsBuffer ( ViUInt32 Vi, ViConstString RepCapIdentifier, ViInt32 dataValueXSize, ViPReal64 dataValueX, ViPInt32
dataValueXActualSize, ViInt32 dataValueYSize, ViPReal32 dataValueY, ViPInt32 dataValueYActualSize, ViInt32 dataValueFailedYSize, ViPReal32 dataValueFailedY, ViPInt32
dataValueFailedYActualSize );
```

```
ANVNA_API ViStatus ANVNA_GetLowerTraceLowerLimitFailPointsBuffer ( ViUInt32 Vi, ViConstString RepCapIdentifier, ViInt32 dataValueXSize, ViPReal64 dataValueX, ViPInt32
dataValueXActualSize, ViInt32 dataValueYSize, ViPReal32 dataValueY, ViPInt32 dataValueYActualSize, ViInt32 dataValueFailedYSize, ViPReal32 dataValueFailedY, ViPInt32
dataValueFailedYActualSize );
```

```
ANVNA_API ViStatus ANVNA_GetLowerTraceUpperLimitFailPointsBuffer ( ViUInt32 Vi, ViConstString RepCapIdentifier, ViInt32 dataValueXSize, ViPReal64 dataValueX, ViPInt32
dataValueXActualSize, ViInt32 dataValueYSize, ViPReal32 dataValueY, ViPInt32 dataValueYActualSize, ViInt32 dataValueFailedYSize, ViPReal32 dataValueFailedY, ViPInt32
dataValueFailedYActualSize );
```

into

```
ANVNA_API ViStatus ANVNA_GetLowerLimitFailPointsBuffer ( ViUInt32 Vi, ViConstString RepCapIdentifier, ViInt32 dataValueXSize, ViPReal64 dataValueX, ViPInt32
```

```
dataValueXActualSize );  
  ANVNA_API ViStatus ANVNA_GetUpperLimitFailPointsBuffer ( ViUInt32 Vi, ViConstString RepCapIdentifier, ViInt32 dataValueXSize, ViPReal64 dataValueX, ViPInt32  
dataValueXActualSize );  
  ANVNA_API ViStatus ANVNA_GetLowerTraceLowerLimitFailPointsBuffer ( ViUInt32 Vi, ViConstString RepCapIdentifier, ViInt32 dataValueXSize, ViPReal64 dataValueX, ViPInt32  
dataValueXActualSize);  
  ANVNA_API ViStatus ANVNA_GetLowerTraceUpperLimitFailPointsBuffer ( ViUInt32 Vi, ViConstString RepCapIdentifier, ViInt32 dataValueXSize, ViPReal64 dataValueX, ViPInt32  
dataValueXActualSize );
```

```
=====  
Release Notes for IVI-C Driver v1.11.0002  
=====
```

New features

1. Add defines for Mixed Mode defines for One Differential Pair mode:

```
#define ANVNA_VAL_ANRITSU_VNA_MESUREMENT_RESPONSE_MIXEDMODE_ODP_SDD 0  
#define ANVNA_VAL_ANRITSU_VNA_MESUREMENT_RESPONSE_MIXEDMODE_ODP_SCC 1  
#define ANVNA_VAL_ANRITSU_VNA_MESUREMENT_RESPONSE_MIXEDMODE_ODP_SDC 2  
#define ANVNA_VAL_ANRITSU_VNA_MESUREMENT_RESPONSE_MIXEDMODE_ODP_SCD 3
```

```
=====  
Release Notes for IVI-C Driver v1.10.0008  
=====
```

New features

1. Add function for retrieving the Calibration Kit:

```
ViStatus ANVNA_GetManualCalKit ( ViSession Vi, ViConstString RepCapIdentifier, ViUInt16 PortNumber, ViPUInt32 KitId, ViPUInt32 BBLoad )
```

2. Add functions to set/get dielectric values:

```
#define ANVNA_VAL_ANRITSU_VNA_DIELECTRIC_AIR 0
```

```
#define ANVNA_VAL_ANRITSU_VNA_DIELECTRIC_POLYETHYLENE 1
```

```
#define ANVNA_VAL_ANRITSU_VNA_DIELECTRIC_TEFLON 2
```

```
#define ANVNA_VAL_ANRITSU_VNA_DIELECTRIC_MICROPOROUS_TEFLON 3
```

```
#define ANVNA_VAL_ANRITSU_VNA_DIELECTRIC_OTHER 4
```

```
ANVNA_API ViStatus ANVNA_SetPowerControlDielectric ( ViSession Vi, ViConstString RepCapIdentifier, ViUInt32 type, ViReal64 value );
```

```
ANVNA_API ViStatus ANVNA_GetPowerControlDielectric ( ViSession Vi, ViConstString RepCapIdentifier, ViPUInt32 type, ViPReal64 value );
```

3. Add function for loading calibration kit:

```
ANVNA_API ViStatus ANVNA_LoadCalibrationKit ( ViSession Vi, ViConstString RepCapIdentifier, ViConstString calKitFile, ViConstString label, ViUInt32 type, ViUInt32 calibrationLine, ViUInt32 calibrationMethod );
```

4. Deprecate old calibration kit loading function:

```
ANVNA_API ViStatus ANVNA_LoadCalKit ( ViSession Vi, ViConstString RepCapIdentifier, ViConstString calKitFile, ViUInt32 port );
```

5. Remove following Time Domain functions:

```
ANVNA_EnableTimeDomainOption
```

```
ANVNA_DisableTimeDomainOption
```

```
ANVNA_IsTimeDomainInstalled
```

```
ANVNA_SetTimeDomainLowPassHarmonicSetup
```

6. Add new read only attribute for retrieving the board device id.

```
ANVNA_ATTR_INSTRUMENT_SERIAL_NUMBER
```

7. Add new read-write attribute for system reference: ANVNA_ATTR_SYSTEM_REFERENCE

```
#define ANVNA_VAL_ANRITSU_VNA_REFERENCE_INTERNAL 0
```

```
#define ANVNA_VAL_ANRITSU_VNA_REFERENCE_EXTERNAL 1
```

8. Add ANVNA_VAL_TRIGGER_MODE_SINGLE_SWEEP_AND_HOLD for "Single sweep and hold" GUI state.

```
#define ANVNA_VAL_TRIGGER_MODE_SINGLE_SWEEP_AND_HOLD
```

```
=====  
Release Notes for IVI-C Driver v1.10  
=====
```

New features

1. Defined values for attribute ANVNA_ATTR_CHANNEL_MEASUREMENT_FORMAT was updated to:

```
#define ANVNA_VAL_ANRITSU_VNA_MEASUREMENT_LOG_MAG          0  
#define ANVNA_VAL_ANRITSU_VNA_MEASUREMENT_LIN_MAG         1  
#define ANVNA_VAL_ANRITSU_VNA_MEASUREMENT_PHASE          2  
#define ANVNA_VAL_ANRITSU_VNA_MEASUREMENT_REAL           3  
#define ANVNA_VAL_ANRITSU_VNA_MEASUREMENT_IMAG           4  
#define ANVNA_VAL_ANRITSU_VNA_MEASUREMENT_SWR            5  
#define ANVNA_VAL_ANRITSU_VNA_MEASUREMENT_IMPEDANCE_REAL 6  
#define ANVNA_VAL_ANRITSU_VNA_MEASUREMENT_IMPEDANCE_IMAG 7  
#define ANVNA_VAL_ANRITSU_VNA_MEASUREMENT_IMPEDANCE_MAGNITUDE 8  
#define ANVNA_VAL_ANRITSU_VNA_MEASUREMENT_IMPEDANCE_REAL_IMAG 9  
#define ANVNA_VAL_ANRITSU_VNA_MEASUREMENT_IMPEDANCE_INDUCTANCE 10  
#define ANVNA_VAL_ANRITSU_VNA_MEASUREMENT_IMPEDANCE_CAPACITANCE 11  
#define ANVNA_VAL_ANRITSU_VNA_MEASUREMENT_SMITH_IMPEDANCE_LIN_PHASE 12  
#define ANVNA_VAL_ANRITSU_VNA_MEASUREMENT_SMITH_IMPEDANCE_LOG_PHASE 13  
#define ANVNA_VAL_ANRITSU_VNA_MEASUREMENT_SMITH_IMPEDANCE_REAL_IMAG 14  
#define ANVNA_VAL_ANRITSU_VNA_MEASUREMENT_SMITH_IMPEDANCE 15  
#define ANVNA_VAL_ANRITSU_VNA_MEASUREMENT_SMITH_ADMITTANCE_LIN_PHASE 16  
#define ANVNA_VAL_ANRITSU_VNA_MEASUREMENT_SMITH_ADMITTANCE_LOG_PHASE 17  
#define ANVNA_VAL_ANRITSU_VNA_MEASUREMENT_SMITH_ADMITTANCE_REAL_IMAG 18  
#define ANVNA_VAL_ANRITSU_VNA_MEASUREMENT_SMITH_ADMITTANCE 19
```

```
#define ANVNA_VAL_ANRITSU_VNA_MEASUREMENT_LIN_POLAR_LIN_PHASE      20
#define ANVNA_VAL_ANRITSU_VNA_MEASUREMENT_LIN_POLAR_REAL_IMAG    21
#define ANVNA_VAL_ANRITSU_VNA_MEASUREMENT_LOG_POLAR_LOG_PHASE    22
#define ANVNA_VAL_ANRITSU_VNA_MEASUREMENT_LOG_POLAR_REAL_IMAG    23
#define ANVNA_VAL_ANRITSU_VNA_MEASUREMENT_LOG_MAG_PAHSE         24
#define ANVNA_VAL_ANRITSU_VNA_MEASUREMENT_LIN_MAG_PHASE         25
#define ANVNA_VAL_ANRITSU_VNA_MEASUREMENT_REAL_IMAG              26
#define ANVNA_VAL_ANRITSU_VNA_MEASUREMENT_GROUP_DELAY            27
```

2. Add function to get response type:

```
ANVNA_VAL_ANRITSU_VNA_MESUREMENT_RESPONSE_TYPE_S      0
ANVNA_VAL_ANRITSU_VNA_MESUREMENT_RESPONSE_TYPE_USER_DEFINED  1
ANVNA_API ViStatus ANVNA_ChannelMeasurementGetResponseType ( ViSession Vi, ViConstString RepCapIdentifier, ViPUInt32 ResponseType );
```

3. Add functions to set/get user defined response type:

```
#define ANVNA_VAL_ANRITSU_VNA_MESUREMENT_A1      0
#define ANVNA_VAL_ANRITSU_VNA_MESUREMENT_A2      1
#define ANVNA_VAL_ANRITSU_VNA_MESUREMENT_B1      2
#define ANVNA_VAL_ANRITSU_VNA_MESUREMENT_B2      3
#define ANVNA_VAL_ANRITSU_VNA_MESUREMENT_ONE     4
ANVNA_API ViStatus ANVNA_ChannelMeasurementSetUserDefinedParameter ( ViSession Vi, ViConstString RepCapIdentifier, ViUInt32 numerator, ViUInt32 denominator, ViUInt32 DriverPort );
ANVNA_API ViStatus ANVNA_ChannelMeasurementGetUserDefinedParameter ( ViSession Vi, ViConstString RepCapIdentifier, ViPUInt32 numerator, ViPUInt32 denominator, ViPUInt32 DriverPort );
```

4. Add function to get double markers (lower and upper values):

```
ANVNA_API ViStatus ANVNA_GetMarkerUpLowValue ( ViSession Vi, ViConstString RepCapIdentifier, ViUInt32 markerNum, ViPReal64 up, ViPReal64 low );
```

5. Add function for setting manual calibration kit from the predefined GUI list:

```
ANVNA_API ViStatus ANVNA_SetManualCalKit ( ViSession Vi, ViConstString RepCapIdentifier, ViUInt16 PortNumber, ViUInt32 KitId, ViUInt32 BBLoad );
```

And defines for KitId:

```

#define ANVNA_VAL_ANRITSU_VNA_CALKIT_TWOPPOINTFOUR_MALE          0
#define ANVNA_VAL_ANRITSU_VNA_CALKIT_GPCTHREEPOINTFIVE_MALE     1
#define ANVNA_VAL_ANRITSU_VNA_CALKIT_KCONN_MALE                  2
#define ANVNA_VAL_ANRITSU_VNA_CALKIT_NCONN_MALE                  3
#define ANVNA_VAL_ANRITSU_VNA_CALKIT_SMA_MALE                    4
#define ANVNA_VAL_ANRITSU_VNA_CALKIT_TNC_MALE                    5
#define ANVNA_VAL_ANRITSU_VNA_CALKIT_VCONN_MALE                  6
#define ANVNA_VAL_ANRITSU_VNA_CALKIT_W1CONN_MALE                 7
#define ANVNA_VAL_ANRITSU_VNA_CALKIT_SEVENSIXTEEN_MALE           8
#define ANVNA_VAL_ANRITSU_VNA_CALKIT_GPC7_MALE                   9
#define ANVNA_VAL_ANRITSU_VNA_CALKIT_NCONN75_MALE                10
#define ANVNA_VAL_ANRITSU_VNA_CALKIT_TOSLK50_MALE                11
#define ANVNA_VAL_ANRITSU_VNA_CALKIT_TOSLN50_MALE                12
#define ANVNA_VAL_ANRITSU_VNA_CALKIT_GCS35M_MALE                 13
#define ANVNA_VAL_ANRITSU_VNA_CALKIT_WR10_MALE                   14
#define ANVNA_VAL_ANRITSU_VNA_CALKIT_WR12_MALE                   15
#define ANVNA_VAL_ANRITSU_VNA_CALKIT_WR15_MALE                   16
#define ANVNA_VAL_ANRITSU_VNA_CALKIT_WR28_MALE                   17
#define ANVNA_VAL_ANRITSU_VNA_CALKIT_WR42_MALE                   18
#define ANVNA_VAL_ANRITSU_VNA_CALKIT_WR62_MALE                   19
#define ANVNA_VAL_ANRITSU_VNA_CALKIT_WR75_MALE                   20
#define ANVNA_VAL_ANRITSU_VNA_CALKIT_WR90_MALE                   21
#define ANVNA_VAL_ANRITSU_VNA_CALKIT_WR112_MALE                  22
#define ANVNA_VAL_ANRITSU_VNA_CALKIT_WR137_MALE                  23
#define ANVNA_VAL_ANRITSU_VNA_CALKIT_WR159_MALE                  24
#define ANVNA_VAL_ANRITSU_VNA_CALKIT_WR187_MALE                  25
#define ANVNA_VAL_ANRITSU_VNA_CALKIT_WR229_MALE                  26
#define ANVNA_VAL_ANRITSU_VNA_CALKIT_USERDEFINED1_MALE           27
#define ANVNA_VAL_ANRITSU_VNA_CALKIT_USERDEFINED2_MALE           28
#define ANVNA_VAL_ANRITSU_VNA_CALKIT_USERDEFINED3_MALE           29
#define ANVNA_VAL_ANRITSU_VNA_CALKIT_USERDEFINED4_MALE           30
#define ANVNA_VAL_ANRITSU_VNA_CALKIT_USERDEFINED5_MALE           31
  
```


#define ANVNA_VAL_ANRITSU_VNA_CALKIT_USERDEFINED6_MALE	32
#define ANVNA_VAL_ANRITSU_VNA_CALKIT_USERDEFINED7_MALE	33
#define ANVNA_VAL_ANRITSU_VNA_CALKIT_USERDEFINED8_MALE	34
#define ANVNA_VAL_ANRITSU_VNA_CALKIT_TWOPOINTFOUR_FEMALE	100
#define ANVNA_VAL_ANRITSU_VNA_CALKIT_GPCTHREEPOINTFIVE_FEMALE	101
#define ANVNA_VAL_ANRITSU_VNA_CALKIT_KCONN_FEMALE	102
#define ANVNA_VAL_ANRITSU_VNA_CALKIT_NCONN_FEMALE	103
#define ANVNA_VAL_ANRITSU_VNA_CALKIT_SMA_FEMALE	104
#define ANVNA_VAL_ANRITSU_VNA_CALKIT_TNC_FEMALE	105
#define ANVNA_VAL_ANRITSU_VNA_CALKIT_VCONN_FEMALE	106
#define ANVNA_VAL_ANRITSU_VNA_CALKIT_W1CONN_FEMALE	107
#define ANVNA_VAL_ANRITSU_VNA_CALKIT_SEVENSIXTEEN_FEMALE	108
#define ANVNA_VAL_ANRITSU_VNA_CALKIT_GPC7_FEMALE	109
#define ANVNA_VAL_ANRITSU_VNA_CALKIT_NCONN75_FEMALE	110
#define ANVNA_VAL_ANRITSU_VNA_CALKIT_TOSLK50_FEMALE	111
#define ANVNA_VAL_ANRITSU_VNA_CALKIT_TOSLN50_FEMALE	112
#define ANVNA_VAL_ANRITSU_VNA_CALKIT_GCS35M_FEMALE	113
#define ANVNA_VAL_ANRITSU_VNA_CALKIT_WR10_FEMALE	114
#define ANVNA_VAL_ANRITSU_VNA_CALKIT_WR12_FEMALE	115
#define ANVNA_VAL_ANRITSU_VNA_CALKIT_WR15_FEMALE	116
#define ANVNA_VAL_ANRITSU_VNA_CALKIT_WR28_FEMALE	117
#define ANVNA_VAL_ANRITSU_VNA_CALKIT_WR42_FEMALE	118
#define ANVNA_VAL_ANRITSU_VNA_CALKIT_WR62_FEMALE	119
#define ANVNA_VAL_ANRITSU_VNA_CALKIT_WR75_FEMALE	120
#define ANVNA_VAL_ANRITSU_VNA_CALKIT_WR90_FEMALE	121
#define ANVNA_VAL_ANRITSU_VNA_CALKIT_WR112_FEMALE	122
#define ANVNA_VAL_ANRITSU_VNA_CALKIT_WR137_FEMALE	123
#define ANVNA_VAL_ANRITSU_VNA_CALKIT_WR159_FEMALE	124
#define ANVNA_VAL_ANRITSU_VNA_CALKIT_WR187_FEMALE	125
#define ANVNA_VAL_ANRITSU_VNA_CALKIT_WR229_FEMALE	126
#define ANVNA_VAL_ANRITSU_VNA_CALKIT_USERDEFINED1_FEMALE	127

```
#define ANVNA_VAL_ANRITSU_VNA_CALKIT_USERDEFINED2_FEMALE 128
#define ANVNA_VAL_ANRITSU_VNA_CALKIT_USERDEFINED3_FEMALE 129
#define ANVNA_VAL_ANRITSU_VNA_CALKIT_USERDEFINED4_FEMALE 130
#define ANVNA_VAL_ANRITSU_VNA_CALKIT_USERDEFINED5_FEMALE 131
#define ANVNA_VAL_ANRITSU_VNA_CALKIT_USERDEFINED6_FEMALE 132
#define ANVNA_VAL_ANRITSU_VNA_CALKIT_USERDEFINED7_FEMALE 133
#define ANVNA_VAL_ANRITSU_VNA_CALKIT_USERDEFINED8_FEMALE 134
```

6. Add function for getting the Time Domain Distance and Gate Values:

```
ANVNA_API ViStatus ANVNA_GetTimeDomainDistanceValues ( ViSession Vi, ViConstString RepCapIdentifier, ViInt32 RetValBufferSize, ViPReal64 RetVal, ViPInt32 RetValActualSize );
ANVNA_API ViStatus ANVNA_GetTimeDomainGateValues ( ViSession Vi, ViConstString RepCapIdentifier, ViInt32 RetValBufferSize, ViPReal32 RetVal, ViPInt32 RetValActualSize );
```

7. Add functions to manage limits:

```
ANVNA_API ViStatus ANVNA_SetLimitTestingOnOff ( ViSession Vi, ViConstString RepCapIdentifier, ViBoolean onOff );
ANVNA_API ViStatus ANVNA_GetLimitTestingOnOff ( ViSession Vi, ViConstString RepCapIdentifier, ViPBoolean onOff );
ANVNA_API ViStatus ANVNA_SetLimitTestResultSign ( ViSession Vi, ViBoolean onOff );
ANVNA_API ViStatus ANVNA_GetLimitTestResultSign ( ViSession Vi, ViPBoolean onOff );
ANVNA_API ViStatus ANVNA_ClearAllLimits ( ViSession Vi, ViConstString RepCapIdentifier );
ANVNA_API ViStatus ANVNA_GetLimitsCount ( ViSession Vi, ViConstString RepCapIdentifier, ViPUInt32 count );
ANVNA_API ViStatus ANVNA_AddLimit ( ViSession Vi, ViConstString RepCapIdentifier, ViReal64 x1, ViReal64 x2, ViReal32 y1, ViReal32 y2, ViReal32 radius, ViUInt32 limitType );
ANVNA_API ViStatus ANVNA_SetLimit ( ViSession Vi, ViConstString RepCapIdentifier, ViUInt32 limitSegmentNumber, ViReal64 x1, ViReal64 x2, ViReal32 y1, ViReal32 y2, ViReal32 radius, ViUInt32 limitType );
ANVNA_API ViStatus ANVNA_GetLimit ( ViSession Vi, ViConstString RepCapIdentifier, ViUInt32 limitSegmentNumber, ViPReal64 x1, ViPReal64 x2, ViPReal32 y1, ViPReal32 y2, ViPReal32 radius, ViPUInt32 limitType );
ANVNA_API ViStatus ANVNA_SetLimitType ( ViSession Vi, ViConstString RepCapIdentifier, ViUInt32 limitSegmentNumber, ViUInt32 limitType );
ANVNA_API ViStatus ANVNA_GetLimitType ( ViSession Vi, ViConstString RepCapIdentifier, ViUInt32 limitSegmentNumber, ViPUInt32 limitType );
ANVNA_API ViStatus ANVNA_DeleteLimitSegmentAt ( ViSession Vi, ViConstString RepCapIdentifier, ViUInt32 limitSegmentNumber );
ANVNA_API ViStatus ANVNA_IsLimitTestPass ( ViUInt32 Vi, ViConstString RepCapIdentifier, ViPBoolean passNoPass );
ANVNA_API ViStatus ANVNA_GetLowerLimitBuffer ( ViUInt32 Vi, ViConstString RepCapIdentifier, ViInt32 dataIndexXSize, ViPInt32 dataIndexX, ViPInt32 dataIndexXActualSize, ViInt32 dataValueYSize, ViPReal32 dataValueY, ViPInt32 dataValueYActualSize );
ANVNA_API ViStatus ANVNA_GetUpperLimitBuffer ( ViUInt32 Vi, ViConstString RepCapIdentifier, ViInt32 dataIndexXSize, ViPInt32 dataIndexX, ViPInt32 dataIndexXActualSize,
```

```
ViInt32 dataValueYSize, ViPReal32 dataValueY, ViPInt32 dataValueYActualSize );  
ANVNA_API ViStatus ANVNA_GetLowerLimitFailPointsBuffer ( ViUInt32 Vi, ViConstString RepCapIdentifier, ViInt32 dataValueXSize, ViPReal64 dataValueX, ViPInt32  
dataValueXActualSize, ViInt32 dataValueYSize, ViPReal32 dataValueY, ViPInt32 dataValueYActualSize, ViInt32 dataValueFailedYSize, ViPReal32 dataValueFailedY, ViPInt32  
dataValueFailedYActualSize );  
ANVNA_API ViStatus ANVNA_GetUpperLimitFailPointsBuffer ( ViUInt32 Vi, ViConstString RepCapIdentifier, ViInt32 dataValueXSize, ViPReal64 dataValueX, ViPInt32  
dataValueXActualSize, ViInt32 dataValueYSize, ViPReal32 dataValueY, ViPInt32 dataValueYActualSize, ViInt32 dataValueFailedYSize, ViPReal32 dataValueFailedY, ViPInt32  
dataValueFailedYActualSize );
```

8. Add functions to manage SmartCal & AutoCal:

```
ANVNA_API ViStatus ANVNA_AddSelectedPort ( ViSession Vi, ViUInt16 PortNumber, ViUInt16 CurrentEncodedPorts, ViPUInt16 NewEncodedPorts );  
ANVNA_API ViStatus ANVNA_SetAutoCalDevice ( ViSession Vi, ViConstString RepCapIdentifier, ViConstString comPort, ViConstString characterizationFile, ViUInt16 portLeft,  
ViUInt16 portRight, ViBoolean orientation, ViBoolean autoSenseOn );  
ANVNA_API ViStatus ANVNA_GetAutoCalDevice ( ViSession Vi, ViConstString RepCapIdentifier, ViPChar comPort, ViPChar characterizationFile, ViPUInt16 portLeft, ViPUInt16  
portRight, ViPBoolean orientation, ViPBoolean autoSenseOn );  
ANVNA_API ViStatus ANVNA_SetSmartCalDevice ( ViSession Vi, ViConstString RepCapIdentifier, ViUInt16 NumPorts, ViUInt16 PortA, ViUInt16 PortB, ViUInt16 PortC, ViUInt16  
PortD, ViBoolean autoSenseOn );  
ANVNA_API ViStatus ANVNA_GetSmartCalDevice ( ViSession Vi, ViConstString RepCapIdentifier, ViPUInt16 NumPorts, ViPUInt16 PortA, ViPUInt16 PortB, ViPUInt16 PortC,  
ViPUInt16 PortD, ViPBoolean autoSenseOn );  
ANVNA_API ViStatus ANVNA_SetAdditionalThru ( ViSession Vi, ViConstString RepCapIdentifier, ViUInt16 portA, ViUInt16 portB );  
ANVNA_API ViStatus ANVNA_AddOnePortConnection ( ViSession Vi, ViConstString RepCapIdentifier, ViUInt16 CalibrationType, ViUInt32 portA, ViBoolean resetAccumulation );  
ANVNA_API ViStatus ANVNA_AddTwoPortConnection ( ViSession Vi, ViConstString RepCapIdentifier, ViUInt16 CalibrationType, ViUInt32 portA, ViUInt32 portB, ViBoolean  
resetAccumulation );  
ANVNA_API ViStatus ANVNA_AddThreePortConnection ( ViSession Vi, ViConstString RepCapIdentifier, ViUInt16 CalibrationType, ViUInt32 portA, ViUInt32 portB, ViUInt32 portC,  
ViBoolean resetAccumulation );  
ANVNA_API ViStatus ANVNA_AddFourPortConnection ( ViSession Vi, ViConstString RepCapIdentifier, ViUInt16 CalibrationType, ViUInt32 portA, ViUInt32 portB, ViUInt32 portC,  
ViUInt32 portD, ViBoolean resetAccumulation );  
ANVNA_API ViStatus ANVNA_BeginAutoCalCalibration ( ViSession Vi, ViConstString RepCapIdentifier, ViPBoolean endStatus, ViPUInt16 portA, ViPUInt16 portB );  
ANVNA_API ViStatus ANVNA_ResumeAutoCalCalibration ( ViSession Vi, ViConstString RepCapIdentifier, ViPBoolean endStatus, ViPUInt16 portA, ViPUInt16 portB );  
ANVNA_API ViStatus ANVNA_EndAutoCalCalibration ( ViSession Vi, ViConstString RepCapIdentifier, ViPBoolean endStatus );
```

9. Add function to setup manual calibration in consistent manner with SmartCal and AutoCal:

```
ANVNA_API ViStatus ANVNA_SetupManualCalibration ( ViSession Vi, ViConstString RepCapIdentifier, ViUInt32 calibrationMethod, ViUInt32 calibrationLine );
```

10. Add functions for managing MixedMode and MaxEfficiency Response:

```
#define ANVNA_VAL_ANRITSU_VNA_MESUREMENT_RESPONSE_MIXEDMODE_TWODIFFERENTIALPAIRS      0
#define ANVNA_VAL_ANRITSU_VNA_MESUREMENT_RESPONSE_MIXEDMODE_ONEDIFFERENTIALPAIRONESINGLETON  1
#define ANVNA_VAL_ANRITSU_VNA_MESUREMENT_RESPONSE_MIXEDMODE_ONEDIFFERENTIALPAIRTWO SINGLETONS  2
#define ANVNA_VAL_ANRITSU_VNA_MESUREMENT_RESPONSE_MIXEDMODE_ONEDIFFERENTIALPAIR      3

#define ANVNA_VAL_ANRITSU_VNA_MESUREMENT_RESPONSE_MIXEDMODE_TDP_SD1D1      0
#define ANVNA_VAL_ANRITSU_VNA_MESUREMENT_RESPONSE_MIXEDMODE_TDP_SD1D2      1
#define ANVNA_VAL_ANRITSU_VNA_MESUREMENT_RESPONSE_MIXEDMODE_TDP_SD2D1      2
#define ANVNA_VAL_ANRITSU_VNA_MESUREMENT_RESPONSE_MIXEDMODE_TDP_SD2D2      3
#define ANVNA_VAL_ANRITSU_VNA_MESUREMENT_RESPONSE_MIXEDMODE_TDP_SC1C1      4
#define ANVNA_VAL_ANRITSU_VNA_MESUREMENT_RESPONSE_MIXEDMODE_TDP_SC1C2      5
#define ANVNA_VAL_ANRITSU_VNA_MESUREMENT_RESPONSE_MIXEDMODE_TDP_SC2C1      6
#define ANVNA_VAL_ANRITSU_VNA_MESUREMENT_RESPONSE_MIXEDMODE_TDP_SC2C2      7
#define ANVNA_VAL_ANRITSU_VNA_MESUREMENT_RESPONSE_MIXEDMODE_TDP_SD1C1      8
#define ANVNA_VAL_ANRITSU_VNA_MESUREMENT_RESPONSE_MIXEDMODE_TDP_SD1C2      9
#define ANVNA_VAL_ANRITSU_VNA_MESUREMENT_RESPONSE_MIXEDMODE_TDP_SD2C1     10
#define ANVNA_VAL_ANRITSU_VNA_MESUREMENT_RESPONSE_MIXEDMODE_TDP_SD2C2     11
#define ANVNA_VAL_ANRITSU_VNA_MESUREMENT_RESPONSE_MIXEDMODE_TDP_SC1D1     12
#define ANVNA_VAL_ANRITSU_VNA_MESUREMENT_RESPONSE_MIXEDMODE_TDP_SC1D2     13
#define ANVNA_VAL_ANRITSU_VNA_MESUREMENT_RESPONSE_MIXEDMODE_TDP_SC2D1     14
#define ANVNA_VAL_ANRITSU_VNA_MESUREMENT_RESPONSE_MIXEDMODE_TDP_SC2D2     15

#define ANVNA_VAL_ANRITSU_VNA_MESUREMENT_RESPONSE_MIXEDMODE_ODPOS_SXX      0
#define ANVNA_VAL_ANRITSU_VNA_MESUREMENT_RESPONSE_MIXEDMODE_ODPOS_SXD      1
#define ANVNA_VAL_ANRITSU_VNA_MESUREMENT_RESPONSE_MIXEDMODE_ODPOS_SXC      2
#define ANVNA_VAL_ANRITSU_VNA_MESUREMENT_RESPONSE_MIXEDMODE_ODPOS_SDX      3
#define ANVNA_VAL_ANRITSU_VNA_MESUREMENT_RESPONSE_MIXEDMODE_ODPOS_SCX      4
#define ANVNA_VAL_ANRITSU_VNA_MESUREMENT_RESPONSE_MIXEDMODE_ODPOS_SDD      5
#define ANVNA_VAL_ANRITSU_VNA_MESUREMENT_RESPONSE_MIXEDMODE_ODPOS_SDC      6
```

```

#define ANVNA_VAL_ANRITSU_VNA_MESUREMENT_RESPONSE_MIXEDMODE_ODPOS_SCD 7
#define ANVNA_VAL_ANRITSU_VNA_MESUREMENT_RESPONSE_MIXEDMODE_ODPOS_SCC 8

#define ANVNA_VAL_ANRITSU_VNA_MESUREMENT_RESPONSE_MIXEDMODE_ODPTS_SXX 0
#define ANVNA_VAL_ANRITSU_VNA_MESUREMENT_RESPONSE_MIXEDMODE_ODPTS_SXY 1
#define ANVNA_VAL_ANRITSU_VNA_MESUREMENT_RESPONSE_MIXEDMODE_ODPTS_SYX 2
#define ANVNA_VAL_ANRITSU_VNA_MESUREMENT_RESPONSE_MIXEDMODE_ODPTS_SYY 3
#define ANVNA_VAL_ANRITSU_VNA_MESUREMENT_RESPONSE_MIXEDMODE_ODPTS_SXD 4
#define ANVNA_VAL_ANRITSU_VNA_MESUREMENT_RESPONSE_MIXEDMODE_ODPTS_SXC 5
#define ANVNA_VAL_ANRITSU_VNA_MESUREMENT_RESPONSE_MIXEDMODE_ODPTS_SYD 6
#define ANVNA_VAL_ANRITSU_VNA_MESUREMENT_RESPONSE_MIXEDMODE_ODPTS_SYC 7
#define ANVNA_VAL_ANRITSU_VNA_MESUREMENT_RESPONSE_MIXEDMODE_ODPTS_SDX 8
#define ANVNA_VAL_ANRITSU_VNA_MESUREMENT_RESPONSE_MIXEDMODE_ODPTS_SDY 9
#define ANVNA_VAL_ANRITSU_VNA_MESUREMENT_RESPONSE_MIXEDMODE_ODPTS_SCX 10
#define ANVNA_VAL_ANRITSU_VNA_MESUREMENT_RESPONSE_MIXEDMODE_ODPTS_SCY 11
#define ANVNA_VAL_ANRITSU_VNA_MESUREMENT_RESPONSE_MIXEDMODE_ODPTS_SDD 12
#define ANVNA_VAL_ANRITSU_VNA_MESUREMENT_RESPONSE_MIXEDMODE_ODPTS_SDC 13
#define ANVNA_VAL_ANRITSU_VNA_MESUREMENT_RESPONSE_MIXEDMODE_ODPTS_SCD 14
#define ANVNA_VAL_ANRITSU_VNA_MESUREMENT_RESPONSE_MIXEDMODE_ODPTS_SCC 15
  
```

```

ANVNA_API ViStatus ANVNA_ChannelMeasurementGetMixedModeResponseType ( ViSession Vi, ViConstString RepCapIdentifier, ViPUInt32 ResponseType );
ANVNA_API ViStatus ANVNA_ChannelMeasurementSetMixedModeTwoDifferentialPairsResponse ( ViSession Vi, ViConstString RepCapIdentifier, ViUInt32 Pair1Port1, ViUInt32
Pair1Port2, ViUInt32 Pair2Port1, ViUInt32 Pair2Port2, ViUInt32 Response );
ANVNA_API ViStatus ANVNA_ChannelMeasurementGetMixedModeTwoDifferentialPairsResponse ( ViSession Vi, ViConstString RepCapIdentifier, ViPUInt32 Pair1Port1, ViPUInt32
Pair1Port2, ViPUInt32 Pair2Port1, ViPUInt32 Pair2Port2, ViPUInt32 Response );
ANVNA_API ViStatus ANVNA_ChannelMeasurementSetMixedModeOneDifferentialPairOneSingletonResponse ( ViSession Vi, ViConstString RepCapIdentifier, ViUInt32 Pair1Port1,
ViUInt32 Pair1Port2, ViUInt32 Singleton1, ViUInt32 Response );
ANVNA_API ViStatus ANVNA_ChannelMeasurementGetMixedModeOneDifferentialPairOneSingletonResponse ( ViSession Vi, ViConstString RepCapIdentifier, ViPUInt32
Pair1Port1, ViPUInt32 Pair1Port2, ViPUInt32 Singleton1, ViPUInt32 Response );
ANVNA_API ViStatus ANVNA_ChannelMeasurementSetMixedModeOneDifferentialPairTwoSingletonsResponse ( ViSession Vi, ViConstString RepCapIdentifier, ViUInt32 Pair1Port1,
ViUInt32 Pair1Port2, ViUInt32 Singleton1, ViUInt32 Singleton2, ViUInt32 Response );
ANVNA_API ViStatus ANVNA_ChannelMeasurementGetMixedModeOneDifferentialPairTwoSingletonsResponse ( ViSession Vi, ViConstString RepCapIdentifier, ViPUInt32
  
```

Pair1Port1, ViPUInt32 Pair1Port2, ViPUInt32 Singleton1, ViPUInt32 Singleton2, ViPUInt32 Response);
ANVNA_API ViStatus ANVNA_ChannelMeasurementSetMixedModeOneDifferentialPairResponse (ViSession Vi, ViConstString RepCapIdentifier, ViUInt32 Pair1Port1, ViUInt32
Pair1Port2, ViUInt32 Response);
ANVNA_API ViStatus ANVNA_ChannelMeasurementGetMixedModeOneDifferentialPairResponse (ViSession Vi, ViConstString RepCapIdentifier, ViPUInt32 Pair1Port1, ViPUInt32
Pair1Port2, ViPUInt32 Response);
ANVNA_API ViStatus ANVNA_ChannelMeasurementSetMaxEfficiencyResponse (ViSession Vi, ViConstString RepCapIdentifier, ViUInt32 Port1, ViUInt32 Port2);
ANVNA_API ViStatus ANVNA_ChannelMeasurementGetMaxEfficiencyResponse (ViSession Vi, ViConstString RepCapIdentifier, ViPUInt32 Port1, ViPUInt32 Port2);

Compatibility issues

None

Fixed issues

None

=====
Release Notes for IVI-C Driver v1.09
=====

New features

1. New supported data formats when calling ANVNA_ChannelSaveState/ANVNA_ChannelRecallState:

- ◆ load: s1p, s2p, tdu, tdf
- ◆ save: s1p, s2p, tdu, tdf, txt

2. Add new read-only attribute ANVNA_ATTR_CHANNEL_MEASUREMENT_SMOOTHING_POINTS for retrieving the number of smoothing points.

3. Add new read-only attribute ANVNA_ATTR_CHANNEL_AVERAGING_COUNT for retrieving the number of averaging runs.

4. Add new read-write attribute ANVNA_ATTR_CHANNEL_AVERAGING_TYPE and its possible values:
 ANVNA_VAL_ANRITSU_VNA_AVERAGING_TYPE_PER_POINT = 0 and
 ANVNA_VAL_ANRITSU_VNA_AVERAGING_TYPE_PER_SWEEP = 1.

5: Add Time Domain API constants and functions:

```

/*- Defines for Time Domain Mode definition */
#define ANVNA_VAL_ANRITSU_VNA_TIMEDOMAIN_TYPE_FREQUENCYNOTIMEGATE      0
#define ANVNA_VAL_ANRITSU_VNA_TIMEDOMAIN_TYPE_FREQUENCYTIMEGATE      1
#define ANVNA_VAL_ANRITSU_VNA_TIMEDOMAIN_TYPE_TIMEBANDPASS           2
#define ANVNA_VAL_ANRITSU_VNA_TIMEDOMAIN_TYPE_TIMELOWPASS           3

/*- Defines for Time Domain Trip definition */
#define ANVNA_VAL_ANRITSU_VNA_TIMEDOMAIN_TRIP_AUTO                    0
#define ANVNA_VAL_ANRITSU_VNA_TIMEDOMAIN_TRIP_ONEWAY                  1
#define ANVNA_VAL_ANRITSU_VNA_TIMEDOMAIN_TRIP_ROUNDTRIP              2

/*- Defines for Time Domain Dielectric definition */
#define ANVNA_VAL_ANRITSU_VNA_TIMEDOMAIN_DIELECTRIC_AIR               0
#define ANVNA_VAL_ANRITSU_VNA_TIMEDOMAIN_DIELECTRIC_POLYETHYLENE     1
#define ANVNA_VAL_ANRITSU_VNA_TIMEDOMAIN_DIELECTRIC_TEFLON          2
#define ANVNA_VAL_ANRITSU_VNA_TIMEDOMAIN_DIELECTRIC_MICROPOROUS     3
#define ANVNA_VAL_ANRITSU_VNA_TIMEDOMAIN_DIELECTRIC_OTHER           4
  
```

```

/*- Defines for Time Domain DC Term definition */
#define ANVNA_VAL_ANRITSU_VNA_TIMEDOMAIN_DCTERM_AUTOEXTRAPOLATE      0
#define ANVNA_VAL_ANRITSU_VNA_TIMEDOMAIN_DCTERM_LINEIMPEDANCE      1
#define ANVNA_VAL_ANRITSU_VNA_TIMEDOMAIN_DCTERM_OPEN                2
#define ANVNA_VAL_ANRITSU_VNA_TIMEDOMAIN_DCTERM_SHORT               3
#define ANVNA_VAL_ANRITSU_VNA_TIMEDOMAIN_DCTERM_OTHER               4

/*- Defines for Time Domain DC Method definition */
#define ANVNA_VAL_ANRITSU_VNA_TIMEDOMAIN_DCTERM_METHOD_PHASEONLY    0
#define ANVNA_VAL_ANRITSU_VNA_TIMEDOMAIN_DCTERM_METHOD_MAGPHASE    1
#define ANVNA_VAL_ANRITSU_VNA_TIMEDOMAIN_DCTERM_METHOD_USERDEFINED  2

/*- Defines for Time Domain Window Shape definition */
#define ANVNA_VAL_ANRITSU_VNA_TIMEDOMAIN_WINDOWSHAPE_NOMINAL        0
#define ANVNA_VAL_ANRITSU_VNA_TIMEDOMAIN_WINDOWSHAPE_RECTANGULAR    1
#define ANVNA_VAL_ANRITSU_VNA_TIMEDOMAIN_WINDOWSHAPE_LOWSIDELOBE    2
#define ANVNA_VAL_ANRITSU_VNA_TIMEDOMAIN_WINDOWSHAPE_MINSIDELOBE    3
#define ANVNA_VAL_ANRITSU_VNA_TIMEDOMAIN_WINDOWSHAPE_KAISERBESSEL   4
#define ANVNA_VAL_ANRITSU_VNA_TIMEDOMAIN_WINDOWSHAPE_DOLPHCHEBYSHEV 5

/*- Defines for Time Domain Gate definition */
#define ANVNA_VAL_ANRITSU_VNA_TIMEDOMAIN_GATE_DISPLAY               0
#define ANVNA_VAL_ANRITSU_VNA_TIMEDOMAIN_GATE_OFF                   1
#define ANVNA_VAL_ANRITSU_VNA_TIMEDOMAIN_GATE_ON                     2

/*- Defines for Time Domain Gate Shape definition */
#define ANVNA_VAL_ANRITSU_VNA_TIMEDOMAIN_GATESHAPE_NOMINAL          0
#define ANVNA_VAL_ANRITSU_VNA_TIMEDOMAIN_GATESHAPE_MINIMUM          1
#define ANVNA_VAL_ANRITSU_VNA_TIMEDOMAIN_GATESHAPE_MAXIMUM          2
#define ANVNA_VAL_ANRITSU_VNA_TIMEDOMAIN_GATESHAPE_WIDE             3
#define ANVNA_VAL_ANRITSU_VNA_TIMEDOMAIN_GATESHAPE_KAISERBESSEL     4
#define ANVNA_VAL_ANRITSU_VNA_TIMEDOMAIN_GATESHAPE_DOLPHCHEBYSHEV  5
  
```



```

/*- Defines for Time Unit definition */
#define ANVNA_VAL_ANRITSU_VNA_TIMEDOMAIN_UNIT_DISTANCE          0
#define ANVNA_VAL_ANRITSU_VNA_TIMEDOMAIN_UNIT_TIME             1

/*- Defines for Time Response definition */
#define ANVNA_VAL_ANRITSU_VNA_TIMEDOMAIN_RESPONSE_IMPULSE      0
#define ANVNA_VAL_ANRITSU_VNA_TIMEDOMAIN_RESPONSE_STEP         1

/*- Defines for Time Domain error definition */
#define ANVNA_ERROR_TIMEDOMAIN_NOTALLOWEDINCWMODE              (IVIC_SPECIFIC_ERROR_BASE + 0X0500)
#define ANVNA_ERROR_TIMEDOMAIN_NOTALLOWEDINPOWERSWEEPMODE      (IVIC_SPECIFIC_ERROR_BASE + 0X0501)
#define ANVNA_ERROR_TIMEDOMAIN_LOWPASSREQUIRESHARMONICSWEEP    (IVIC_SPECIFIC_ERROR_BASE + 0X0502)
#define ANVNA_ERROR_TIMEDOMAIN_LOWPASSNOTALLOWEDWAVEGUIDEORMICROSTRIPLINETYPE (IVIC_SPECIFIC_ERROR_BASE + 0X0503)
#define ANVNA_ERROR_TIMEDOMAIN_NOGROUPDELAY                    (IVIC_SPECIFIC_ERROR_BASE + 0X0504)
#define ANVNA_ERROR_TIMEDOMAIN_SETEXCEPTION                    (IVIC_SPECIFIC_ERROR_BASE + 0X0505)
#define ANVNA_ERROR_TIMEDOMAIN_OPTIONNOTINSTALLED              (IVIC_SPECIFIC_ERROR_BASE + 0X0506)
#define ANVNA_ERROR_TIMEDOMAIN_OPTIONINSTALLED                 (IVIC_SPECIFIC_ERROR_BASE + 0X0507)
#define ANVNA_ERROR_TIMEDOMAIN_OPTIONKEYINVALID                (IVIC_SPECIFIC_ERROR_BASE + 0X0508)
#define ANVNA_ERROR_TIMEDOMAIN_MODEFORGATEDATAINVALID          (IVIC_SPECIFIC_ERROR_BASE + 0X0509)
#define ANVNA_ERROR_TIMEDOMAIN_GATESIZEOUTOFALIASFREERANGE     (IVIC_SPECIFIC_ERROR_BASE + 0X0510)
#define ANVNA_ERROR_TIMEDOMAIN_WINDOWSHAPEVALIDATIONFAILED    (IVIC_SPECIFIC_ERROR_BASE + 0X0511)
#define ANVNA_ERROR_TIMEDOMAIN_WINDOWSHAPEINPUTVALIDATIONFAILED (IVIC_SPECIFIC_ERROR_BASE + 0X0512)
#define ANVNA_ERROR_TIMEDOMAIN_GATESHAPEVALIDATIONFAILED      (IVIC_SPECIFIC_ERROR_BASE + 0X0513)
#define ANVNA_ERROR_TIMEDOMAIN_GATESHAPEINPUTVALIDATIONFAILED (IVIC_SPECIFIC_ERROR_BASE + 0X0514)
#define ANVNA_ERROR_TIMEDOMAIN_DISPLAYUNITVALIDATIONFAILED    (IVIC_SPECIFIC_ERROR_BASE + 0X0515)
#define ANVNA_ERROR_TIMEDOMAIN_TRIPDEFINITIONVALIDATIONFAILED (IVIC_SPECIFIC_ERROR_BASE + 0X0516)
#define ANVNA_ERROR_TIMEDOMAIN_RESPONSEVALIDATIONFAILED       (IVIC_SPECIFIC_ERROR_BASE + 0X0517)
#define ANVNA_ERROR_TIMEDOMAIN_EXTRAPOLATIONTYPEVALIDATIONFAILED (IVIC_SPECIFIC_ERROR_BASE + 0X0518)
#define ANVNA_ERROR_TIMEDOMAIN_EXTRAPOLATIONMETHODVALIDATIONFAILED (IVIC_SPECIFIC_ERROR_BASE + 0X0519)
#define ANVNA_ERROR_TIMEDOMAIN_GATEFUNCTIONVALIDATIONFAILED   (IVIC_SPECIFIC_ERROR_BASE + 0X0520)
#define ANVNA_ERROR_TIMEDOMAIN_TRACENUMBERVALIDATIONFAILED    (IVIC_SPECIFIC_ERROR_BASE + 0X0521)

```

```
#define ANVNA_ERROR_TIMEDOMAIN_TYPEVALIDATIONFAILED (IVIC_SPECIFIC_ERROR_BASE + 0X0522)
#define ANVNA_ERROR_TIMEDOMAIN_RAWDATAOUTPUTBUFFERSIZEINCORRECT (IVIC_SPECIFIC_ERROR_BASE + 0X0523)
#define ANVNA_ERROR_TIMEDOMAIN_RAWDATAGATEFUNCTIONINVALID (IVIC_SPECIFIC_ERROR_BASE + 0X0524)
#define ANVNA_ERROR_TIMEDOMAIN_TIMELISTNOTAVAILABLE (IVIC_SPECIFIC_ERROR_BASE + 0X0525)
#define ANVNA_ERROR_TIMEDOMAIN_RAWGATEDATANOTREADY (IVIC_SPECIFIC_ERROR_BASE + 0X0526)
#define ANVNA_ERROR_TIMEDOMAIN_NOTAVAILABLEINTHISFIRMWAREREVISION (IVIC_SPECIFIC_ERROR_BASE + 0X0527)

/*- Time Domain */
ANVNA_API ViStatus ANVNA_EnableTimeDomainOption ( ViSession Vi, ViConstString RepCapIdentifier, ViConstString password );
ANVNA_API ViStatus ANVNA_IsTimeDomainInstalled ( ViSession Vi, ViConstString RepCapIdentifier, ViBoolean on_off );
ANVNA_API ViStatus ANVNA_SetTimeDomainType ( ViSession Vi, ViConstString RepCapIdentifier, ViUInt32 domainType );
ANVNA_API ViStatus ANVNA_GetTimeDomainType ( ViSession Vi, ViConstString RepCapIdentifier, ViPUInt32 domainType );
ANVNA_API ViStatus ANVNA_SetTimeDomainLowPassHarmonicSetup ( ViSession Vi, ViConstString RepCapIdentifier, ViReal64 start, ViReal64 stop, ViUInt32 points );
ANVNA_API ViStatus ANVNA_SetTimeDomainResponse ( ViSession Vi, ViConstString RepCapIdentifier, ViUInt32 response );
ANVNA_API ViStatus ANVNA_GetTimeDomainResponse ( ViSession Vi, ViConstString RepCapIdentifier, ViPUInt32 response );
ANVNA_API ViStatus ANVNA_SetTimeDomainTrip ( ViSession Vi, ViConstString RepCapIdentifier, ViUInt32 tripMode );
ANVNA_API ViStatus ANVNA_GetTimeDomainTrip ( ViSession Vi, ViConstString RepCapIdentifier, ViPUInt32 tripMode );
ANVNA_API ViStatus ANVNA_SetTimeDomainUnit ( ViSession Vi, ViConstString RepCapIdentifier, ViUInt32 unit );
ANVNA_API ViStatus ANVNA_GetTimeDomainUnit ( ViSession Vi, ViConstString RepCapIdentifier, ViPUInt32 unit );
ANVNA_API ViStatus ANVNA_SetTimeDomainRangeStartStop ( ViSession Vi, ViConstString RepCapIdentifier, ViReal64 start, ViReal64 stop );
ANVNA_API ViStatus ANVNA_GetTimeDomainRangeStartStop ( ViSession Vi, ViConstString RepCapIdentifier, ViPReal64 start, ViPReal64 stop );
ANVNA_API ViStatus ANVNA_SetTimeDomainRangeCenterSpan ( ViSession Vi, ViConstString RepCapIdentifier, ViReal64 center, ViReal64 span );
ANVNA_API ViStatus ANVNA_GetTimeDomainRangeCenterSpan ( ViSession Vi, ViConstString RepCapIdentifier, ViPReal64 center, ViPReal64 span );
ANVNA_API ViStatus ANVNA_SetTimeDomainRangeDCTerm ( ViSession Vi, ViConstString RepCapIdentifier, ViUInt32 extrapolationType, ViReal64 extrapolation, ViUInt32 extrapolationMethod );
ANVNA_API ViStatus ANVNA_GetTimeDomainRangeDCTerm ( ViSession Vi, ViConstString RepCapIdentifier, ViPUInt32 extrapolationType, ViPReal64 extrapolation, ViPUInt32 extrapolationMethod, ViPReal64 reflectionCoefficient );
ANVNA_API ViStatus ANVNA_SetTimeDomainRangeProperties ( ViSession Vi, ViConstString RepCapIdentifier, ViUInt32 windowShape, ViReal64 shapeValue, ViReal64 aliasFreeRange );
ANVNA_API ViStatus ANVNA_GetTimeDomainRangeProperties ( ViSession Vi, ViConstString RepCapIdentifier, ViPUInt32 windowShape, ViPReal64 shapeValue );
ANVNA_API ViStatus ANVNA_SetTimeDomainGateStartStop ( ViSession Vi, ViConstString RepCapIdentifier, ViReal64 start, ViReal64 stop );
ANVNA_API ViStatus ANVNA_GetTimeDomainGateStartStop ( ViSession Vi, ViConstString RepCapIdentifier, ViPReal64 start, ViPReal64 stop );
```

```
ANVNA_API ViStatus ANVNA_SetTimeDomainGateCenterSpan ( ViSession Vi, ViConstString RepCapIdentifier, ViReal64 center, ViReal64 span );  
ANVNA_API ViStatus ANVNA_GetTimeDomainGateCenterSpan ( ViSession Vi, ViConstString RepCapIdentifier, ViPReal64 center, ViPReal64 span );  
ANVNA_API ViStatus ANVNA_SetTimeDomainGateProperties ( ViSession Vi, ViConstString RepCapIdentifier, ViUInt32 setup, ViBoolean notch, ViUInt32 shape, ViReal64  
shapeValue );  
ANVNA_API ViStatus ANVNA_GetTimeDomainGateProperties ( ViSession Vi, ViConstString RepCapIdentifier, ViPUInt32 setup, ViPBoolean notch, ViPUInt32 shape, ViPReal64  
shapeValue );
```

6. New native python driver, compatible with all 2.x and 3.x versions.

7. Add Channel Source Power Level functions to set/get power value for each channel and port.

```
ANVNA_API ViStatus ANVNA_SetChannelSourcePowerLevel ( ViSession Vi, ViConstString RepCapIdentifier, ViInt32 PortVal, ViReal64 Val );  
ANVNA_API ViStatus ANVNA_GetChannelSourcePowerLevel ( ViSession Vi, ViConstString RepCapIdentifier, ViInt32 PortVal, ViPReal64 Val );
```

8. Simulation mode: max frequency is 43.5 Ghz and ANVNA_ChannelMeasurementFetchComplex now returns data.

9. Notifications to Shockline Tray Controller now consider multiple boards capabilities.

10. Disconnecting a client closes the session and notifies the Shockline Tray Controller.

11. Implement Index Segmented Sweeping feature.

Compatibility issues

None

Fixed issues

RDF-1 Wrong Compatibility Version in IVI-C Installer
Fix issues for Matlab 64 environment.

=====
Release Notes for IVI-C Driver v1.08
=====

New features

1. Supported new device Anritsu MS46121A.
2. Add new attribute for querying the number of available ports on the device: ANVNA_ATTR_INSTRUMENT_PORTS_COUNT.
3. Calibration API now considers the transmission Line Type; ANVNA_SetupCalibration has a new parameter with corresponding line type:





ANVNA_VAL_ANRITSU_VNA_CALIBRATION_LINETYPE_COAXIAL	0
ANVNA_VAL_ANRITSU_VNA_CALIBRATION_LINETYPE_WAVEGUIDE	1
ANVNA_VAL_ANRITSU_VNA_CALIBRATION_LINETYPE_MICROSTRIP	2
ANVNA_VAL_ANRITSU_VNA_CALIBRATION_LINETYPE_NONDISPERSIVE	3

4. New function for getting calibration line type: ANVNA_GetCalibrationLine
5. New function for loading S1P calibration: ANVNA_LoadS1PKit.

Compatibility issues

Signature change for ANVNA_SetupCalibration function.

Fixed issues

- #OT 23389: Driver installer: message not matching with the current Shockline version
- #OT 23406: ANVNA_SetupCalibration  no error when invalid value for calibration method is used
- #OT 23339: Potentially Harmful SW message
- #OT 23359: Segment function  segment added when incorrect frequency used
- #OT 23358: Segment functions  no errors when incorrect repCapIdentifier is used
- #OT 3494: Add read-only attribute for number of test ports
- #OT 22051: GetMarkerValue - min/max values returned NOK
- #OT 3273: Proper error message when installer doesn't work
- #OT 22808: ANVNA_StartCalibration  4-port calibration type on model MS46522A (2-port)