AXRF RF Subsystem and AutoCal Unit

User Manual

Document part no. 47090/270
AXRF RF Subsystem and AutoCal Unit
User Manual
About this manual

This manual explains how to use the capabilities of the AXRF RF Subsystem and the AutoCal Unit, and version 3.0 or higher of the AXRF software.

Intended audience

Persons engaged in the design and manufacture of RF and microwave systems. Familiarity with the terms used in RF and microwave measurements is assumed.

Document conventions

The following conventions apply throughout this manual:

- **Monospace** examples of syntax and programming examples.
- **Bold** denotes items that you must enter or select. Also denotes default conditions of API parameters.
- **Italic** specialized terms.
Contents

About this manual ................................................................. ii
Intended audience .................................................................... ii
Document conventions ................................................................ ii
Associated documentation ..................................................... vii
Safety instructions ................................................................... viii
Invalid cal data/damage to modules ........................................ viii
Precautions ........................................................................... ix
Précautions .......................................................................... xi
Vorsichtsmaßnahmen ............................................................ xv
Precauzioni ........................................................................... xviii
Precauciones .......................................................................... xxii
Introduction ........................................................................... 1
AXRF RF Subsystem ............................................................. 1
AutoCal Unit .......................................................................... 1
Specifications ......................................................................... 2
AXRF RF Subsystem specification ........................................ 2
AutoCal Unit specification .................................................. 3
Loadboard version ............................................................... 3
Declarations of conformity .................................................... 3
Installation requirements ....................................................... 4
AXRF RF Subsystem cooling .................................................. 4
18-slot chassis ...................................................................... 4
Connecting to supply ........................................................... 4
AXRF RF Subsystem ............................................................. 4
AutoCal Unit .......................................................................... 4
Disconnecting device ........................................................... 4
Hardware installation ............................................................ 6
Available power in chassis .................................................... 6
AXRF RF Subsystem configuration and cabling ..................... 7
Cabling ................................................................................ 7
Input/output ports .................................................................. 8
Required software ................................................................. 9
Cobham PXI module software .............................................. 9
Power sensor software ........................................................ 9
FTDI driver software ........................................................ 9
AXRF RF Subsystem software ............................................. 9
PXI Measurement Suite software ....................................... 9
National Instruments PXI Platform Services software ............ 9
National Instruments NI-VISA ............................................. 9
Cal data ................................................................................ 9
Software installation ............................................................. 10
Installation instructions ......................................................... 10
Programming the AXRF RF Subsystem ................................ 11
General information ............................................................ 11
AXRF administrative syntax ................................................ 12
TevAXRF_Initiate ............................................................... 12
TevAXRF_Close ................................................................. 12
TevAXRF_Isolate ............................................................... 13
TevAXRF_SetFastSettlingMode .......................................... 13
Preface
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Current consumption</td>
<td>104</td>
</tr>
<tr>
<td>Current measurements</td>
<td>104</td>
</tr>
<tr>
<td>Maintenance</td>
<td>105</td>
</tr>
<tr>
<td>Cooling</td>
<td>105</td>
</tr>
<tr>
<td>Cleaning the exterior</td>
<td>105</td>
</tr>
<tr>
<td>Power requirements</td>
<td>105</td>
</tr>
<tr>
<td>Service or assistance</td>
<td>105</td>
</tr>
</tbody>
</table>
## Associated documentation

<table>
<thead>
<tr>
<th>If you want to...</th>
<th>Refer to...</th>
</tr>
</thead>
</table>
| Read initial safety and setup instructions about the AXRF RF Subsystem and AutoCal Unit | Getting Started with AXRF RF Subsystem and AutoCal Unit  
Part no. 47000/303 (printed), 47090/303 (electronic)  
Supplied as a printed document; also on AXRF System CD-ROM 46886/100 and at [www.cobham.com/wireless](http://www.cobham.com/wireless) |
Safety instructions

These requirements apply to both the AXRF RF Subsystem and the AutoCal Unit, unless otherwise specified.

For user safety, read and follow all instructions, WARNINGS, CAUTIONS, and Notes marked in this manual and on the associated instrument before handling or operating the instrument.

Read these safety instructions carefully.

Keep this manual for future reference.

Read the specifications section of this manual for information about the operating environment of the instrument.

When installing/mounting or uninstalling/removing an instrument:

- turn off power and unplug any power cords/cables.

To avoid electrical shock and/or damage to the instrument:

- keep the instrument away from water or liquid sources;
- keep the instrument away from high heat or high humidity;
- keep the instrument properly ventilated (do not block or cover ventilation openings);
- make sure to use recommended voltage and power source settings;
- always install and operate the instrument near an easily accessible electrical socket-outlet;
- secure the power cord (do not place any object on/over the power cord);
- only install/attach and operate the instrument on stable surfaces and/or recommended mountings;
- if the instrument is not to be used for long periods, turn it off and unplug it from its power source.

Never attempt to repair an instrument. Instruments should only be serviced by qualified personnel.

An instrument must be serviced by authorized technicians when:

- the power cord or plug is damaged;
- liquid has penetrated the instrument;
- it has been exposed to high humidity/moisture;
- it is not functioning or does not function according to the user manual;
- it has been dropped and/or damaged;
- it has an obvious sign of breakage.

CAUTION

Invalid cal data/damage to modules

Do not remove any of the installed PXI modules. Removing a module invalidates the system calibration data supplied with the AXRF RF Subsystem, and may damage the module.
Safety

Precautions

These terms have specific meanings in this manual:

**WARNING** information to prevent personal injury.

**CAUTION** information to prevent damage to the equipment.

**Note** important general information.

Hazard symbols

The meaning of hazard symbols appearing on the equipment and in the documentation is as follows:

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Warning" /></td>
<td>Refer to the operating manual when this symbol is marked on the equipment. Familiarize yourself with the nature of the hazard and the actions that may have to be taken.</td>
</tr>
<tr>
<td><img src="image" alt="Dangerous Voltage" /></td>
<td>Dangerous voltage</td>
</tr>
<tr>
<td><img src="image" alt="Toxic Hazard" /></td>
<td>Toxic hazard</td>
</tr>
<tr>
<td><img src="image" alt="Hot Surface" /></td>
<td>Hot surface</td>
</tr>
</tbody>
</table>

**WARNING**

Suitability for use

This equipment has been designed and manufactured by Cobham to generate low-power RF signals for testing radio communications apparatus and to digitize and provide spectrum analysis of RF signals.

If the equipment is not used in a manner specified by Cobham, or if it is damaged, the protection provided by the equipment may be impaired.

Cobham has no control over the use of this equipment and cannot be held responsible for events arising from its use other than for its intended purpose.

General conditions of use

This product is designed and tested to comply with the requirements of BS EN 61010-1 ‘Safety requirements for electrical equipment for measurement, control and laboratory use’, for Class III equipment and is for use in a pollution degree 2 environment. The equipment is designed to operate from an installation category II supply.

Equipment should be protected from the ingress of liquids and precipitation such as rain, snow, etc. When moving the equipment from a cold to a hot environment, it is important to allow the temperature of the equipment to stabilize before it is connected to the supply to avoid condensation forming. The equipment must only be operated within the environmental conditions specified in the data sheet, otherwise the protection provided by the equipment may be impaired.

This product is not approved for use in hazardous atmospheres or medical applications or safety-critical applications.

The enclosure of the host PXI chassis should meet the enclosure requirements of BS EN 61010-1, in order to provide an appropriate fire enclosure for the product.
Safety

**WARNING**

⚠️ Electrical hazards (DC supply voltage)

This equipment conforms with IEC safety Class III, meaning that for continued safety it must only be connected to supplies and signal sources which conform to 'Separated Extra-Low Voltage' (SELV and SELV-E) voltage and insulation requirements. No hazardous voltages are generated internally. See the data sheet for the maximum permitted voltage levels that can be applied.

Do not remove equipment covers as this may result in personal injury. There are no user-serviceable parts inside.

Refer all servicing to qualified personnel.

**WARNING**

⚠️ Heavy equipment

The weight of this equipment exceeds the 18 kg (40 lb) guideline for manual handling by a single person. To avoid the risk of injury, an assessment should be carried out prior to handling which takes account of the load, workplace environment and individual capability, in accordance with European Directive 90/269/EEC and associated National Regulations.

**WARNING**

⚠️ Toxic hazards

Some of the components used in this equipment may include resins and other materials which give off toxic fumes if incinerated. Take appropriate precautions, therefore, in the disposal of these items.

**WARNING**

⚠️ Beryllium copper

It is possible that some mechanical components within this equipment may be manufactured from beryllium copper. This is an alloy with a beryllium content of approximately 5%. It represents no risk in normal use.

The material should not be machined, welded or subjected to any process where heat is involved.

It must be disposed of as "special waste".

It must NOT be disposed of by incineration.

**WARNING**

⚠️ Short circuited/bent pins

Before replacing a module into the chassis, check inside the chassis that no debris is present between pins on the backplane connectors, and that no pins on the backplane connectors are bent or damaged. Check the module’s connectors in the same way. Short-circuits can damage the module and chassis, and may cause fire.

If debris is present, ensure that no power is applied to the chassis, and carefully remove the debris with an antistatic brush.

The module should slide smoothly and easily into the slot. If you feel resistance, remove the module and check for obstructions or damage, and that the module is compatible with the chosen slot.
Safety

WARNING

⚠️ Hot surfaces

Take care when touching a module which has run for a prolonged period; the surface temperature can become high.

CAUTION

⚠️ Static-sensitive components

This equipment contains static-sensitive components which may be damaged by handling.

Note

The AXIe Autocal loadboard unit has been supplied by Cobham for use during calibration of Cobham AXIe subsystems. It is a subassembly intended to be integrated into a customer’s test system and so is not supplied with a protective enclosure.

Integration of the Autocal loadboard unit into a test system, including safety and EMC aspects of the installation, is the responsibility of the system integrator and is outside the control of Cobham.
Précautions

Les termes suivants ont, dans ce manuel, des significations particulières:

**WARNING** contient des informations pour éviter toute blessure au personnel.

**CAUTION** contient des informations pour éviter les dommages aux équipements.

**Note** contient d'importantes informations d'ordre général.

Symboles signalant un risque

La signification des symboles de danger apparaissant sur l'équipement et dans la documentation est la suivante:

<table>
<thead>
<tr>
<th>Symbole</th>
<th>Nature du risque</th>
</tr>
</thead>
<tbody>
<tr>
<td>![Symbole]</td>
<td>Reportez-vous au manuel d'utilisation quand ce symbole apparaît sur l'instrument. Familiarisez-vous avec la nature du danger et la conduite à tenir.</td>
</tr>
<tr>
<td>![Symbole]</td>
<td>Tension dangereuse</td>
</tr>
<tr>
<td>![Symbole]</td>
<td>Danger produits toxiques</td>
</tr>
<tr>
<td>![Symbole]</td>
<td>Surfaces chaudes</td>
</tr>
</tbody>
</table>

**WARNING** Utilisation

Cet équipement a été conçu et fabriqué par Cobham pour générer des signaux RF de faible puissance pour le test d'appareils de radio communications et numériser et analyser le spectre de signaux RF.

La protection de l'équipement peut être altérée s'il n'est pas utilisé dans les conditions spécifiées par Cobham, ou il est endommagé.

Cobham n’a aucun contrôle sur l’usage de l’instrument, et ne pourra être tenu pour responsable en cas d’événement survenant suite à une utilisation différente de celle prévue.

Conditions générales d’utilisation

Ce produit a été conçu et testé pour être conforme aux exigences des normes BS EN 61010-1 "Règles de sécurité pour appareils électriques de mesurage, de régulation et de laboratoire", pour des équipements Classe III et pour une utilisation dans un environnement de pollution de niveau 2. Cet équipement est conçu pour fonctionner à partir d’une alimentation de catégorie II.

Cet équipement doit être protégé de l’introduction de liquides ainsi que des précipitations d’eau, de neige, etc... Lorsqu’on transporte cet équipement d’un environnement chaud vers un environnement froid, il est important de laisser l’équipement se stabiliser en température avant de le connecter à une alimentation afin d’éviter toute formation de condensation. L’appareil doit être utilisé uniquement dans le cadre des conditions d’environnement spécifiées dans la fiche technique, toute autre utilisation peut endommager les systèmes de protection.

Ce produit n’est pas garanti pour fonctionner dans des atmosphères dangereuses ou la sécurité des applications critiques.

L'enceinte du châssis PXI hôte doit répondre aux exigences de l'enceinte de la norme BS EN 61010-1, afin de fournir une enceinte de feu appropriée pour le produit.
Sécurité électrique (tension d'alimentation continue)

Cet équipement est conforme aux normes de sécurité CEI Classe III, c’est-à-dire qu’il ne doit être connecté qu’à des sources d’alimentation ou de signaux qui suivent les recommandations de tension et d’isolement du type ‘Tension extra-faible séparée’ (SELV at SELV-E). Aucune tension dangereuse n’est générée en interne. La fiche technique précise les niveaux de tension maximum acceptables en entrée.

Ne démontez pas le capot de l’instrument, car ceci peut provoquer des blessures. Il n’y a pas de pièces remplaçables par l’utilisateur à l’intérieur.

Faites effectuer toute réparation par du personnel qualifié.

Equipement lourd

Le poids de cet appareil est supérieur à la limite de 18 kg (40 lb), fixée pour le transport par une seule personne. Afin d’éviter tout risque de blessure, il est nécessaire de faire, avant le transport, une évaluation de la charge, des contraintes de l’environnement et des capacités de l’individu, en conformité avec la Directive Européenne 90/269/EEC ainsi que les recommandations Nationales concernées.

Danger produits toxiques

Certains composants utilisés dans cet appareil peuvent contenir des résines et d’autres matières qui dégagent des fumées toxiques lors de leur incinération. Les précautions d’usages doivent donc être prises lorsqu’on se débarrasse de ce type de composant.

Bronze au beryllium

Il est possible que dans cet équipement certaines pièces mécaniques soient à base de bronze au beryllium. Il s’agit d’un alliage dans lequel le pourcentage de beryllium ne dépasse pas 5 %. Il ne présente aucun danger en utilisation normale.

Toutefois, cet alliage ne doit pas être travaillé, soudé ou soumis à un processus qui implique l’utilisation d’une source de chaleur.

En cas de destruction, il sera entreposé dans un container spécial. Il ne devra pas être détruit par incinération.

Court-circuit/broches tordues

Avant de remplacer un module dans le châssis, vérifiez l’intérieur du châssis qu’aucun débris sont présents entre les broches sur les connecteurs de fond de panier, et qu’aucune broche des connecteurs fond de panier ne sont tordues ou endommagées. Courts-circuits peuvent endommager le module et le châssis, et peut provoquer un incendie.

Si des débris sont présents, assurez qu’aucune tension est appliquée sur le châssis, et retirez soigneusement les débris avec une brosse antistatique.
Le module doit glisser en douceur et facilement dans la fente. Si vous sentez une résistance, retirez le module et vérifier s’il ya des obstructions ou des dommages, et que le module est compatible avec la fente choisie.

**WARNING**

⚠️ **Surfaces chaudes**

Faites attention en touchant un module qui a fonctionné pendant une période prolongée; la température de surface peut devenir haute.
Vorsichtsmaßnahmen

Diese Hinweise haben eine bestimmte Bedeutung in diesem Handbuch:

**WARNING** dienen zur Vermeidung von Verletzungsrisiken.

**CAUTION** dienen dem Schutz der Geräte.

**Note** enthalten wichtige Informationen.

Gefahrensymbole

Die Bedeutung der Gefahrensymbole auf den Geräten und in der Dokumentation ist wie folgt:

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Gefahrenart</th>
</tr>
</thead>
<tbody>
<tr>
<td>![symbol]</td>
<td>Gefährliche Spannung</td>
</tr>
<tr>
<td>![symbol]</td>
<td>Warnung vor giftigen Substanzen</td>
</tr>
<tr>
<td>![symbol]</td>
<td>Heiße Oberfläche</td>
</tr>
</tbody>
</table>

**WARNING**

Eignung für Gebrauch

Dieses Gerät wurde von Cobham entwickelt und hergestellt um HF Signale geringer Leistung zum Test von Kommunikationseinrichtungen zu erzeugen und HF Signale zu digitalisieren und Spektrumanalyse an HF Signalen durchzuführen.

Sollte das Gerät nicht auf die von Cobham vorgesehene Art und Weise verwendet werden, oder wenn es beschädigt ist, kann die Schutzfunktion des Gerätes beeinträchtigt werden.

Cobham hat keinen Einfluß auf die Art der Verwendung und übernimmt keinerlei Verantwortung bei unsachgemäßer Handhabung.

Allgemeine Hinweise zur Verwendung


Das Gerät sollte vor dem Eindringen von Flüssigkeiten sowie vor Regen, Schnee etc. geschützt werden. Bei Standortänderung von kalter in wärmere Umgebung sollte das Gerät wegen der Kondensation erst nach Anpassung an die wärmere Umgebung mit dem Netz verbunden werden. Das Gerät darf nur in Umgebungsbedingungen wie im Datenblatt beschrieben, betrieben werden; ansonsten wird der vom Gerät vorgesehene Schutz des Anwenders beeinträchtigt.

Dieses Produkt ist nicht für den Einsatz in gefährlicher Umgebung (z.B. Ex-Bereich) und für sicherheitskritische Anwendungen geprüft.

Das Gehäuse der Host PXI-Chassis sollten die Gehäuse Anforderungen von BS EN 61010-1 erfüllen, um eine geeignete Brandschutzgehäuse für das Produkt zu liefern.
**WARNING**

**Elektrische Schläge (Gleichspannungsversorgung)**


Öffnen Sie niemals das Gehäuse der Geräte das dies zu ernsthaften Verletzungen führen kann. Es gibt keine vom Anwender austauschbare Teile in diesem Gerät.

**WARNING**

**Schweres Gerät**


**WARNING**

**Warnung vor giftigen Substanzen**

In einigen Bauelementen dieses Geräts können Epoxyharze oder andere Materialien enthalten sein, die im Brandfall giftige Gase erzeugen. Bei der Entsorgung müssen deshalb entsprechende Vorsichtsmaßnahmen getroffen werden.

**WARNING**

**Beryllium Kupfer**

Es ist möglich, dass in diesem Gerät sind einige mechanische Komponenten aus Beryllium Kupfer gefertigt. Dies ist eine Verbindung welche aus einem Berylliumanteil von ca. 5 % besteht. Bei normaler Verwendung besteht kein Gesundheitsrisiko.

Das Metall darf nicht bearbeitet, geschweißt oder sonstiger Wärmebehandlung ausgesetzt werden.

Es muß als Sondermüll entsorgt werden.

Es darf nicht durch Verbrennung entsorgt werden.

**WARNING**

**Kurzgeschlossen/verbogene Pins**

Bevor Sie ein Modul in das Chassis ersetzen, überprüfen Sie innerhalb des Chassis, dass kein Trümmer zwischen den Pins auf der Backplane-Steckverbinder präsentieren ist und dass keine Pins auf der Backplane-Steckverbinder verbogen oder beschädigt sind. Kurzschlüsse können Modul und Chassis schädigen, und sind ein Feuergefahr.

Wenn Trümmer vorhanden ist, sicherzustellen, dass keine Strom auf dem Chassis angelegt wird, und entfernen Sie vorsichtig die Trümmer mit einem antistatischen Bürste.

Das Modul sollte glatt und leicht in den Steckplatz gleiten. Wenn Sie auf Widerstand stossen, entfernen Sie das Modul und überprüfen Sie auf Hindernisse oder Schäden, und dass das Modul kompatibel mit dem gewählten Steckplatz ist.
WARNING

Heiße Oberfläche

Vorsicht bei Berührung eines Moduls das während eines verlängerten Zeitraums gelaufen ist; die Oberflächentemperatur kann hoch werden.
Precauzioni

Questi termini vengono utilizzati in questo manuale con significati specifici:

- **WARNING**: riportano informazioni atte ad evitare possibili pericoli alla persona.
- **CAUTION**: riportano informazioni per evitare possibili pericoli all'apparecchiatura.
- **Note**: riportano importanti informazioni di carattere generale.

Simboli di pericolo

Il significato del simbolo di pericolo riportato sugli strumenti e nella documentazione è il seguente:

<table>
<thead>
<tr>
<th>Simbolo</th>
<th>Tipo di pericolo</th>
</tr>
</thead>
<tbody>
<tr>
<td>🚨</td>
<td>Fare riferimento al manuale operativo quando questo simbolo è riportato sullo strumento. Rendervi conto della natura del pericolo e delle precauzioni che dovete prendere.</td>
</tr>
<tr>
<td>⚠️</td>
<td>Tensione pericolosa</td>
</tr>
<tr>
<td>⚠️</td>
<td>Pericolo sostanze tossiche</td>
</tr>
<tr>
<td>⚠️</td>
<td>Superfici ad alta temperatura</td>
</tr>
</tbody>
</table>

**WARNING**

Caratteristiche d’uso

Questo strumento è stato progettato e prodotto da Cobham per generare segnali RF in bassa potenza per testare apparati di radio comunicazione e digitalizzare ed eseguire analisi di spettro su segnali RF.

Se lo strumento non è utilizzato nel modo specificato da Cobham, o è danneggiato, le protezioni previste sullo strumento potrebbero risultare inefficaci.

Cobham non ha il controllo sull’uso di questo strumento e non può essere ritenuta responsabile per eventi risultanti da un uso diverso dallo scopo prefisso.

Condizioni generali d’uso

Questo prodotto è stato progettato e collaudato per rispondere ai requisiti della direttiva BS EN 61010-1 ‘Safety requirements for electrical equipment for measurement, control and laboratory use’ per apparati di classe III, per l’uso in un ambiente inquinato di grado 2. L’apparato è stato progettato per essere alimentato da un alimentatore di categoria II.

Lo strumento deve essere protetto dal possibile ingresso di liquidi quali, ad es., acqua, pioggia, neve, ecc. Qualora lo strumento venga portato da un ambiente freddo ad uno caldo, è importante lasciare che la temperatura all’interno dello strumento si stabilizzi prima di alimentarlo per evitare formazione di condense. Lo strumento deve essere utilizzato esclusivamente nelle condizioni ambientali descritte nella scheda tecnica, in caso contrario le protezioni previste nello strumento potrebbero risultare non sufficienti.

Questo prodotto non è stato approvato per essere usato in ambienti pericolosi o applicazioni safety-critical.

L’involucro del chassis PXI deve soddisfare i requisiti di BS EN 61010-1, al fine di fornire un appropriato al fuoco enclosure per il prodotto.
**Pericoli da elettricità (alimentazione a c.c.)**

Questo strumento rispetta le norme IEC, classe III, e quindi, per una completa sicurezza, deve essere collegato solo ad alimentatori e generatori di segnali che rispettano i requisiti di tensione ed isolamento SELV e SELV-E (Separated Extra-Low Voltage). Nessuna tensione pericolosa è generata al suo interno. Vedi la scheda tecnica per quanto concerne i livelli massimi di tensione applicabili.

Non rimuovete mai le coperture perché così potreste provocare danni a voi stessi. Non vi sono all’interno parti di interesse all’utilizzatore.

Tutte gli interventi sono di competenza del personale qualificato.

**Strumento pesante**

Il peso di questo strumento supera i 18 kg (40 lb) raccomandati come limite per il trasporto manuale da parte di singola persona. Per evitare rischi di danni fisici è bene quindi considerare il carico complessivo, le condizioni del trasporto e le capacità individuali in accordo con la direttiva comunitaria 90/269/EEC e con eventuali regolamenti locali.

**Pericolo sostanze tossiche**

Alcuni dei componenti usati in questo strumento possono contenere resine o altri materiali che, se bruciati, possono emettere fumi tossici. Prendere quindi le opportune precauzioni nell’uso di tali parti.

**Rame berillio**

È possibile che alcuni componenti meccanici in questo strumento sono realizzati in rame berillio. Si tratta di una lega con contenuto di berillio di circa il 5 %, che non presenta alcun rischio in usi normali.

Questo materiale non deve essere lavorato, saldato o subire qualsiasi processo che coinvolge alte temperature.

Deve essere eliminato come “rifiuto speciale”. Non deve essere eliminato tramite “inceneritore”.

**Cortocircuiti/pins piegati**

Prima di sostituire un modulo nello telaio, controllare che all’interno dello telaio nessun detrito sia presente tra i pin dei connettore backplane, e che nessun pin del connettore backplane sia piegato o danneggiato. Cortocircuiti possono danneggiare il modulo lo telaio, e può provocare un incendio.

Se sono presenti detriti, verificare che lo telaio non sia alimentato, e con attenzione rimuovere i detriti con un pennello antistatico.

Il modulo dovrebbe scorrere agevolmente e facilmente nello slot. Se si incontra resistenza, rimuovere il modulo e verificare la presenza di ostruzioni o danni, e che il modulo è compatibile con lo slot scelto.
WARNING

⚠️ Superfici ad alta temperatura

Fare attenzione nel toccare un modulo che ha funzionato per un periodo prolungato; la temperatura in superficie può diventare molto elevata.
Precauciones

Estos términos tienen significados específicos en este manual:

- **WARNING**: contienen información referente a prevención de daños personales.
- **CAUTION**: contienen información referente a prevención de daños en equipos.
- **Note**: contienen información general importante.

Símbolos de peligro

El significado de los símbolos de peligro en el equipo y en la documentación es el siguiente:

<table>
<thead>
<tr>
<th>Símbolo</th>
<th>Naturaleza del peligro</th>
</tr>
</thead>
<tbody>
<tr>
<td>!</td>
<td>Vea el manual de funcionamiento cuando este símbolo aparezca en el instrumento. Familiarícese con la naturaleza del riesgo y con las acciones que deban de tomarse.</td>
</tr>
<tr>
<td>⚡</td>
<td>Voltaje peligroso</td>
</tr>
<tr>
<td>⚠️</td>
<td>Aviso de toxicidad</td>
</tr>
<tr>
<td>🔥</td>
<td>Superficies a altas temperaturas</td>
</tr>
</tbody>
</table>

**WARNING**

Idoneidad de uso

Este equipo ha sido diseñado y fabricado por Cobham para generar señales de VHF y UHF de bajo nivel de potencia para prueba de equipos de radiocomunicaciones y para digitalizar y realizar análisis espectral de señales RF.

Si el equipo fuese utilizado de forma diferente a la especificada por Cobham, o está dañado, la protección ofrecida por el equipo pudiera quedar reducida.

Cobham no tiene control sobre el uso de este equipo y no puede, por tanto, exigirsele responsabilidades derivadas de una utilización distinta de aquellas para las que ha sido diseñado.

Condiciones generales de uso

Este producto ha sido diseñado y probado para cumplir los requerimientos de la normativa BS EN 61010-1 "Requerimientos de la normativa para equipos eléctricos de medida, control y uso en laboratorio", para equipos clase III, para uso en un ambiente con un grado de contaminación 2. El equipo ha sido diseñado para funcionar sobre una instalación de alimentación de categorías II.

Debe protegerse el equipo de la entrada de líquidos y precipitaciones como nieve, lluvia, etc. Cuando se traslada el equipo de entorno frío a un entorno caliente, es importante aguardar la estabilización el equipo para evitar la condensación. Solamente debe utilizarse el equipo bajo las condiciones ambientales especificadas en la Hoja Técnica, en caso contrario la propia protección del equipo puede resultar dañada.

Este producto no ha sido aprobado para su utilización en entornos peligrosos o la seguridad de las aplicaciones críticas.

El recinto del chasis PXI debe satisfacer los requisitos de protección de la norma BS EN 61010-1, con el fin de proporcionar un recinto fuego apropiado para el producto.
Nivel peligroso de electricidad (tensión de alimentación DC)

Este equipo cumple con la norma de seguridad IEC clase III, lo que significa que para total seguridad debe ser conectado a alimentaciones y fuentes de señal que cumplan los requerimientos de tensión y aislamiento “Tensión Separada Extra-Baja” (SELV y SELV-E). Ninguna tensión generada internamente implica riesgo para el operario.

En la Hoja Técnica podrá encontrar los valores máximos permitidos que pueden aplicarse.

No retire las cubiertas del chasis del instrumento, ya que pudiera resultar dañado personalmente. No existen partes que puedan ser reparadas en su interior.

Deje todas las tareas relativas a reparación a un servicio técnico cualificado.

Instrumento pesado

El peso de este equipo excede de los 18 kg (40 lb), lo que debe tenerse en cuenta si va ser transportado manualmente por una sola persona. Para evitar el riesgo de lesiones, antes de mover el equipo deberá evaluar la carga, el entorno de trabajo y la propia capacidad, de acuerdo con la Directiva Europea 90/269/EEC y el Reglamento Nacional Asociado.

Aviso de toxicidad

Alguno de los componentes utilizados en este equipo pudieran incluir resinas u otro tipo de materiales que al arder produjeran sustancias tóxicas. Por tanto, tome las debidas precauciones en la manipulación de esas piezas.

Berilio-cobre

Es posible que algunos componentes mecánicos contenidos en este instrumento incorporan berilio-cobre en su proceso de fabricación. Se trata de una aleación con un contenido aproximado de berilio del 5 %, lo que no representa ningún riesgo durante su uso normal.

El material no debe ser manipulado, soldado, ni sometido a ningún proceso que implique la aplicación de calor.

Para su eliminación debe tratarse como un "residuo especial". El material NO DEBE eliminarse mediante incineración.

Cortocircuito/pines doblados

Antes de reemplazar un módulo en el chasis, compruebe en el interior del chasis que no haya residuos está presente entre los pines de los conectores del panel posterior, y que no pines en los conectores del panel posterior están dobladas o dañadas. Los cortocircuitos pueden dañar el módulo y el chasis, y puede provocar un incendio.

Si los desechos se encuentra presente, asegurarse de que no se les aplica electricidad al chasis, y retirar con cuidado los restos con un cepillo antiestático.

El módulo debe deslizarse suave y fácilmente en la ranura. Si se siente resistencia, retire el módulo y comprobar si hay obstrucciones o daños, y que el módulo es compatible con la ranura elegida.
WARNING

⚠️ Superficies a altas temperaturas

Tenga cuidado al tocar un módulo que ha funcionado por un período prolongado; la temperatura superficial puede llegar a ser alta.

CAUTION

⚠️ Flujo de aire

Coloque una placa ciega de cada ranura no utilizada. Los módulos se pueden recalentarse si el flujo de aire correcta no se mantiene.
Introduction

AXRF RF Subsystem
The AXRF RF Subsystem hardware consists of several PXI cards that are housed in a Cobham chassis.

18-slot PXI chassis (AXRF Option 03)

AutoCal Unit
The AXRF-ACAL8 AutoCal Unit is available in two configurations:

as an enclosed benchtop box, powered by an external +48 VDC power adapter and during calibration, connected by cables to the AXRF RF Subsystem:

or as a loadboard for use with, and powered by, an AXIe Series system.
Specifications

These specifications, provided for the purposes of installation, are a subset of the full specifications provided in AXRF RF Subsystem data sheet 46891/653, available at www.cobham.com/wireless.

AXRF RF Subsystem specification

The AXRF RF Subsystem provides user expansion slots that comply with PXI or PXIe standards.

<table>
<thead>
<tr>
<th>Power supply</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>AC input</td>
<td></td>
</tr>
<tr>
<td>Line voltage</td>
<td></td>
</tr>
<tr>
<td>18-slot chassis (Option 03)</td>
<td>100 to 240 VAC full range</td>
</tr>
<tr>
<td>Line frequency</td>
<td></td>
</tr>
<tr>
<td>18-slot chassis (Option 03)</td>
<td>50 to 60 Hz</td>
</tr>
<tr>
<td>DC power</td>
<td>See Available power in chassis</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Physical</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Slots</td>
<td></td>
</tr>
<tr>
<td>18-slot chassis (Option 03)</td>
<td>18 (one system slot, 17 peripheral slots)</td>
</tr>
<tr>
<td>Dimensions</td>
<td></td>
</tr>
<tr>
<td>18-slot chassis (Option 03)</td>
<td>445.5 mm (W) (17.5 in)</td>
</tr>
<tr>
<td>178 mm (H) (7.0 in)</td>
<td></td>
</tr>
<tr>
<td>465 mm (D) (18.3 in)</td>
<td></td>
</tr>
<tr>
<td>Weight</td>
<td></td>
</tr>
<tr>
<td>18-slot chassis (Option 03)</td>
<td>Approximately 17 kg (37 lb) when populated as Figure 1 (8-port plus embedded controller, Options 06, 09, 11, 19)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Environmental (indoor use only)</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Operating</td>
<td>Temperature: 0 to 40 °C</td>
</tr>
<tr>
<td></td>
<td>Relative humidity: &lt;80 % 5 to 31 °C, decreasing linearly to 50 % at 40 °C</td>
</tr>
<tr>
<td></td>
<td>Altitude: 2000 m</td>
</tr>
<tr>
<td>Storage</td>
<td>Temperature: −20 to 70 °C</td>
</tr>
<tr>
<td></td>
<td>Relative humidity: 5 % to 95 % non-condensing (with Option 03)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Certification</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Safety</td>
<td>BS EN 61010-1</td>
</tr>
<tr>
<td>Electromagnetic compatibility</td>
<td>Emissions: BS EN 55011 Class A Immunity: BS EN 61326-1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>General specifications</th>
<th></th>
</tr>
</thead>
</table>
AutoCal Unit specification

<table>
<thead>
<tr>
<th>Power supply</th>
</tr>
</thead>
<tbody>
<tr>
<td>DC input +48 VDC, 1.5 A max, supplied by external 100 to 240 VAC, 50 to 60 Hz power adapter</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Maximum input level</th>
</tr>
</thead>
<tbody>
<tr>
<td>+20 dBm max</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Physical</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dimensions</td>
</tr>
<tr>
<td>414 mm (W) (16.3 in)</td>
</tr>
<tr>
<td>264 mm (H) (10.4 in)</td>
</tr>
<tr>
<td>210 mm (D) (8.3 in)</td>
</tr>
<tr>
<td>Weight</td>
</tr>
<tr>
<td>9.1 kg (20 lb) approximately</td>
</tr>
</tbody>
</table>

Loadboard version

<table>
<thead>
<tr>
<th>Power supply</th>
</tr>
</thead>
<tbody>
<tr>
<td>DC input +48 VDC, 1.5 A max, supplied by test head or by external 100 to 240 VAC, 50 to 60 Hz line power adapter</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Maximum input level</th>
</tr>
</thead>
<tbody>
<tr>
<td>+20 dBm max</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Physical</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dimensions</td>
</tr>
<tr>
<td>390 mm (W) (15.4 in)</td>
</tr>
<tr>
<td>420 mm (W) (16.5 in) over butterfly screw heads</td>
</tr>
<tr>
<td>180 mm (H) (7.1 in)</td>
</tr>
<tr>
<td>200 mm (D) (7.9 in)</td>
</tr>
<tr>
<td>Weight</td>
</tr>
<tr>
<td>5.0 kg (11 lb) approximately</td>
</tr>
</tbody>
</table>

Declarations of conformity

Copies of EC declarations of conformity for the AXRF RF Subsystem and AutoCal Unit are available on request from Cobham.
Installation requirements

These requirements apply to both the AXRF RF Subsystem and the AutoCal Unit, unless otherwise specified.

AXRF RF Subsystem cooling

Set the fan speed to HIGH for optimal cooling.

18-slot chassis

When mounting the 18-slot chassis (Option 03) in a rack or on a bench, allow at least 1U (44.5 mm/1.75 in) clearance above the primary exhaust outlet on the top of the chassis. Keep other objects or equipment a minimum of 76.2 mm (3 in) away from the primary inlet apertures at the rear of the chassis. Provide 44.5 mm/1.75 in clearance at the sides of the chassis.

WARNING

Connecting to supply

AXRF RF Subsystem

The AXRF RF Subsystem is a Safety Class I product and therefore must be earthed. Use the supplied power cord or an appropriate replacement. Make sure that the equipment is plugged into an outlet socket with a protective earth contact.

Ensure that the AC supply is correctly connected to the line power receptacle. For line power in the range 100 to 240 VAC, 50 to 60 Hz (18-slot PXI chassis, AXRF Option 03), the PSU automatically selects the appropriate range.

No manual selection of the voltage-range is provided.

AutoCal Unit

The AC to +48 VDC external power adapter supplied with the AXRF-ACAL8 AutoCal Unit is a Safety Class II product and therefore does not require a protective earth contact.

Ensure that the AC supply is correctly connected to the power adapter. For line power in the range 100 to 240 VAC, 50 to 60 Hz, the external power adapter automatically selects the appropriate range.

No manual selection of the voltage-range is provided.

Disconnecting device

The detachable power cord is the equipment's disconnecting device, but if the equipment is permanently integrated into a rack or system, an external power switch or circuit breaker is required.

Whatever the disconnecting device, make sure that you can reach it easily and that it is accessible at all times.

Use only an approved power cord that does not exceed three meters.
**WARNING**

⚠️ **Power cord (AXRF RF Subsystem only)**

When the equipment has to be plugged into a Class II (ungrounded) two-terminal socket outlet, the cable should either be fitted with a three-pin Class I plug and used in conjunction with an adapter incorporating a ground wire, or be fitted with a Class II plug with an integral ground wire.

Fasten the ground wire securely to ground. If you ground one terminal on a two-terminal socket, it does not provide adequate protection.

If a molded plug has to be removed from a lead, dispose of it immediately. A plug with bare flexible cords is hazardous if it is inserted into a live socket outlet.

The power cord is the equipment’s disconnecting device.

Use the supplied power cord or an appropriate replacement. The power cord set must be an appropriately rated and approved cord set in accordance with the regulations of the country it is used in. Do not replace a detachable power cord with an inadequately rated cord.

**WARNING**

⚠️ **Standby/on switch**

The switch on the front panel of the AXRF RF Subsystem is a standby switch only, and does not isolate the equipment from the supply. Remove the power cord from the socket outlet to isolate the equipment.

**WARNING**

⚠️ **After repair**

You must verify the safe state of the equipment after repair.

**CAUTION**

**Electrostatic discharge (ESD)**

If you are connecting cables that are not already connected to equipment at either end, make sure that any static electricity in the cable is first discharged to ground.
Hardware installation

Available power in chassis

You can insert additional PXI modules into the AXRF RF Subsystem providing that you do not disturb the modules supplied as part of the system, and that the additional modules do not draw current exceeding that available from the subsystem’s power supplies. Table 1 shows the current capacities of the different voltage supplies within the chassis.

This table shows:

- **Capacity**: the maximum current that may be drawn from each voltage supply;
- **Demand**: the current drawn by each option;
- **Maximum**: the total current drawn when all permissible options that draw the highest current are fitted.

### Table 1  DC current capacity of AXRF RF Subsystem

<table>
<thead>
<tr>
<th>Capacity</th>
<th>3.3 V</th>
<th>5 V</th>
<th>12 V</th>
<th>−12 V</th>
</tr>
</thead>
<tbody>
<tr>
<td>Option 03</td>
<td>40 A</td>
<td>59 A</td>
<td>8 A</td>
<td>4.5 A</td>
</tr>
<tr>
<td>Demand</td>
<td>3.3 V</td>
<td>5 V</td>
<td>12 V</td>
<td>−12 V</td>
</tr>
<tr>
<td>AXRF</td>
<td>0.35 A</td>
<td>0.78 A</td>
<td>0.43 A</td>
<td>&lt;0.1 A</td>
</tr>
<tr>
<td>Option 04</td>
<td>1.75 A</td>
<td>0.02 A</td>
<td>0.02 A</td>
<td>—</td>
</tr>
<tr>
<td>Option 06</td>
<td>1.8 A</td>
<td>16.5 A</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>Option 09</td>
<td>2.25 A</td>
<td>4.75 A</td>
<td>0.4 A</td>
<td>0.1 A</td>
</tr>
<tr>
<td>Option 10</td>
<td>2.6 A</td>
<td>5.2 A</td>
<td>0.75 A</td>
<td>0.13 A</td>
</tr>
<tr>
<td>Option 11</td>
<td>3.95 A</td>
<td>3.75 A</td>
<td>0.36 A</td>
<td>0.57 A</td>
</tr>
<tr>
<td>Option 12</td>
<td>3.95 A</td>
<td>3.75 A</td>
<td>0.36 A</td>
<td>0.57 A</td>
</tr>
<tr>
<td>Option 13</td>
<td>3.8 A</td>
<td>3.9 A</td>
<td>0.85 A</td>
<td>0.71 A</td>
</tr>
<tr>
<td>Option 16</td>
<td>0.07 A</td>
<td>0.03 A</td>
<td>0.4 A</td>
<td>&lt;0.1 A</td>
</tr>
<tr>
<td>Option 17</td>
<td>1.07 A</td>
<td>0.05 A</td>
<td>0.8 A</td>
<td>0.2 A</td>
</tr>
<tr>
<td>Option 19</td>
<td>1.72 A</td>
<td>0.09 A</td>
<td>1.6 A</td>
<td>0.4 A</td>
</tr>
<tr>
<td>Maximum</td>
<td>3.3 V</td>
<td>5 V</td>
<td>12 V</td>
<td>−12 V</td>
</tr>
<tr>
<td></td>
<td>14.37 A</td>
<td>30.22 A</td>
<td>3.99 A</td>
<td>1.91 A</td>
</tr>
</tbody>
</table>

**Example**

The maximum current is drawn on the 5 V rail when Option 06 (internal Corei7 controller), Option 10 (RF digitizer 13 GHz), Option 12 (VSG2), Option 13 (VSG1 high power) and Option 19 (eight RF test ports) are fitted: the total current is 30.22 A. These values are shown in **bold** in the table.

The maximum current available for additional PXI modules in the chassis on the 5 V rail is then (59−30.22) = 28.78 A.
AXRF RF Subsystem configuration and cabling

A typical eight-channel AXRF RF Subsystem is shown in Figure 1. It consists of the following modules:

- Four DRPMs
- One RFSM (Option 19)
- One RFM
- One RF digitizer (Option 9* (6 GHz)). Option 10* (13 GHz) may be used as an alternative.
- One VSG (vector signal generator) (Option 11** (6 GHz)). Option 13** (6 GHz high power) may be used as an alternative.

**Figure 1 AXRF RF Subsystem (8-port) cards and cabling**

* Option 9, Option 10 consist of 3011 RF synthesized signal generator + 303xC wideband RF digitizer.

** Option 11, Option 13 consist of 3010 RF synthesized signal generator + 302xC digital RF signal generator.

Cabling

The AXRF RF Subsystem is supplied with cables and modules installed. Cables should not be removed as this can affect the calibration. The correct cabling is listed below.

**RF source cabling**

- Digital RF signal generator RF OUT to source RFSM IN A
- Source RFSM OUT 1 to CH 1 & 2 DRPM RF IN
- Source RFSM OUT 2 to CH 3 & 4 DRPM RF IN
- Source RFSM OUT 3 to CH 5 & 6 DRPM RF IN
- Source RFSM OUT 4 to CH 7 & 8 DRPM RF IN
RF signal cabling
  CH 1 & 2 DRPM LO IN/RF OUT to RFM CH1
  CH 3 & 4 DRPM LO IN/RF OUT to RFM CH2
  CH 5 & 6 DRPM LO IN/RF OUT to RFM CH3
  CH 7 & 8 DRPM LO IN/RF OUT to RFM CH4
  RFM COM to RF digitizer RF IN

10 MHz signal cabling
  RF synthesized signal generator (3010) 10 MHz I/O to digital RF signal generator 10 MHz I/O
  Digital RF signal generator 10 MHz I/O to RF digitizer 10 MHz I/O
  RF digitizer 10 MHz I/O to RF synthesized signal generator (3011)

LO source cabling
  RF synthesized signal generator (3010) LO OUT to digital RF signal generator LO IN
  RF synthesized signal generator (3011) LO OUT to RF digitizer LO IN

Input/output ports
  Ports A and B on the DRPM are the RF I/O ports for the DUT (device under test).
Required software

The following software is required for correct operation of the AXRF RF Subsystem and AutoCal Unit:

Cobham PXI module software
Provides control of the Cobham RF source and digitizer within the AXRF RF Subsystem.
On PXI Drivers and Documentation CD-ROM 46886/028, supplied with the Cobham PXI module hardware.
Alternatively, there is a direct link to downloadable driver software (requires a Cobham user account) at: www.cobham.com/wireless.

Power sensor software
Provides the USB control of the power meter used within the AXRF AutoCal Unit.
On AXRF System Software CD-ROM 46886/100, supplied with the AXRF RF Subsystem hardware.

FTDI driver software
Provides USB control of the AXRF AutoCal Unit.
On AXRF System Software CD-ROM 46886/100, supplied with the AXRF RF Subsystem hardware.

AXRF RF Subsystem software
Provides the drivers, user-level APIs and control panel for the AXRF.
On AXRF System Software CD-ROM 46886/100, supplied with the AXRF RF Subsystem hardware.

PXI Measurement Suite software
On Cobham Measurement Suite CD-ROM 46886/101, supplied when optional measurement suite(s) are purchased.

National Instruments PXI Platform Services software
Identifies chassis and modules, and provides trigger routing.

National Instruments NI-VISA
Used for configuring, programming, and troubleshooting instrumentation systems.

Cal data
Loss compensation data files for the AXRF RF Subsystem and the AutoCal Unit.
Provided on a USB stick.
Software installation

If an embedded or remote PC is supplied by Cobham, **all software is supplied pre-installed**.

**Note**: the following instructions are relevant only where the software is used for remote PC control and the PC is not supplied by Cobham, or if the PC is changed or the hard disk is corrupted.

**Installation instructions**

Install the AXRF RF Subsystem software on the test computer attached to the chassis as follows:

- Install **NI-VISA** from the PXI Drivers and Documentation CD-ROM.
- If it has not installed automatically already with NI-VISA, install **PXI Platform Services** from the PXI-PCIe Interface Accessory Kit.
- After installation of NI-VISA, start MAX, select Help > Check for Updates, and install any required updates to the National Instruments software.
- Install **Cobham PXI module software** from the PXI Drivers and Documentation CD-ROM.
- Install **AXRF system software** from the AXRF System Software CD-ROM.
- Copy the initial AXRF calibration data files from the AXRF USB stick to the cal data directory. Note that this data is overwritten if a calibration of the system is performed using the AutoCal Unit, as described in the Calibration tab section of this document. The default location of the cal data directory for Windows 7 installations is `C:\ProgramData\TEV\AXI\CalData`.
- Configure the backplane triggers (see Basic calibration procedure).
Programming the AXRF RF Subsystem

General information

The AXRF APIs included in this manual are high-level, MVP-based API calls for use by the end user of this RF subsystem.

All AXRF APIs use the following general naming convention:

```
[Prefix][Instrument Type]_Function()
TevAXRF_Function()
```

where AXRF identifies the subsystem/instrument.

Several types of functionality have both single-channel and multi-channel API versions. The multi-channel versions are important when trying to ensure that the same level is sourced at multiple locations.

AXRF channel(s) is typically the first parameter in the API call. Within these APIs, the channels are direct references to the hardware and thus have fixed names and fixed relationship to their respective DRPM:

- AXRF_CH1, AXRF_CH2
- AXRF_CH3, AXRF_CH4
- AXRF_CH5, AXRF_CH6
- AXRF_CH7, AXRF_CH8

Parameter values are unit-less but units can be included for clarity.

Parameters used to accept returned (measured) values are pointers to arrays that must be large enough to hold all data. The APIs do not perform bounds checking on the arrays. If arrays are not dimensioned correctly, data can be written outside the array and can cause problems.

Given the nature of RF measurements, there is a need for variables/arrays to hold complex (real, imaginary) result data. Complex values are handled within the AXRF software using the `NIComplexNumber` data structure as defined in the driver's header file:

```c
typedef struct NIComplexNumber_struct{
  double real;
  double imaginary;
} NIComplexNumber
```
**AXRF administrative syntax**

The APIs included here are functions normally performed as part of the tester executive administrative infrastructure.

### TevAXRF.Initialize

**Description:** Initialize the AXRF RF Subsystem.
- Read the AXRF Configuration file and initialize all PXI instruments.
- Create all PXI instrument handles for use by the AXRF APIs.
- Isolate all AXRF channels.
- Set all Port Modules to measure mode.

This API is typically called as part of the tester executive functionality that occurs when a test program is loaded.

**Syntax:** `TevAXRF.Initialize(void);`

**Return:** An error code is returned that identifies any AXRF module exhibiting a problem during initialization. Individual module error values are OR'd together into a single return value.

<table>
<thead>
<tr>
<th>Module</th>
<th>Error code</th>
</tr>
</thead>
<tbody>
<tr>
<td>TevDRPM #1</td>
<td>0x00001</td>
</tr>
<tr>
<td>TevDRPM #2</td>
<td>0x00002</td>
</tr>
<tr>
<td>TevDRPM #3</td>
<td>0x00004</td>
</tr>
<tr>
<td>TevDRPM #4</td>
<td>0x00008</td>
</tr>
<tr>
<td>TevRFSM</td>
<td>0x00010</td>
</tr>
<tr>
<td>TevRFM</td>
<td>0x00020</td>
</tr>
<tr>
<td>Cobham SigGen A</td>
<td>0x00040</td>
</tr>
<tr>
<td>Cobham Digitizer</td>
<td>0x00080</td>
</tr>
<tr>
<td>Cobham SigGen B</td>
<td>0x00100</td>
</tr>
<tr>
<td>Scalar Factor Read Error</td>
<td>0x00200</td>
</tr>
<tr>
<td>Vector Factor Read Error</td>
<td>0x00400</td>
</tr>
<tr>
<td>Noise Factor Read Error</td>
<td>0x00800</td>
</tr>
<tr>
<td>Configuration File Read Error</td>
<td>0x10000</td>
</tr>
<tr>
<td>No Hardware Found</td>
<td>0x20000</td>
</tr>
</tbody>
</table>

**Usage:** `TevAXRF.Initialize();`

### TevAXRF.Close

**Description:** Release all handles to the AXRF PXI modules.

This API is typically called as part of the tester executive functionality that occurs when a test program is deleted/unloaded.

**Syntax:** `TevAXRF.Close(void);`

**Usage:** `TevAXRF.Close();`
**TevAXRF_Isolate**

**Description:** Set the specified AXRF_CHANNEL to the isolate mode. This puts the port into the measure high mode with 50 ohm characteristic impedance.

**Syntax:**

```c
TevAXRF_Isolate(
    AXRF_CHANNEL channel
);
```

**Parameters:**

- `channel` An AXRF channel:
  - AXRF_CH1, AXRF_CH2, AXRF_CH3, AXRF_CH4,
  - AXRF_CH5, AXRF_CH6, AXRF_CH7, AXRF_CH8.

**Usage:**

```c
TevAXRF_Isolate(AXRF_CH1);
```

**TevAXRF_SetFastSettlingMode**

**Description:** Normally in isolate mode, the source buffer in the DRPM signal path is gated off. This buffer has a thermal tail of 0.1 to 0.3 dBm over several milliseconds. This thermal tail effect can be reduced by enabling the fast settling mode.

When enabled, the isolate mode leaves the DRPM source buffers gated on. This improves source settling between device runs.

**Syntax:**

```c
TevAXRF_SetFastSettlingMode(
    int state
);
```

**Parameters:**

- `state` state of fast settling mode
  - 0 — OFF (default)
  - 1 — ON

**Usage:**

```c
TevAXRF_SetFastSettlingMode(1);
```
**AXRF source syntax**

The APIs included here control the RF sourcing and signal routing.

**TevAXRF_GateOn**

**Description:** Gate on the selected AXRF source.

Normally all the AXRF sources are gated on. If an individual RF source was gated off using `TevAXRF_GateOff()`, it can be gated back on using this API.

Gating on/off does not change any of the RF source’s other setup parameters (frequency, level, etc.).

**Note:** there is only one set of RF sources for each PXI chassis (that is, shared between all eight channels). The channel parameter is only used to identify the chassis.

**Syntax:**

```c
TevAXRF_GateOn(
    AXRF_CHANNEL channel,
    AXRF_GENERATOR generator
);
```

**Parameters:**

- **channel** An AXRF channel:
  - AXRF_CH1, AXRF_CH2, AXRF_CH3, AXRF_CH4,
  - AXRF_CH5, AXRF_CH6, AXRF_CH7, AXRF_CH8.

- **generator** The selected RF source generator.
  One of the following:
  - AXRF_SRCA_GEN
  - AXRF_SRCB_GEN
  - AXRF_LO_GEN.

**Usage:**

```c
// Gate the Source A generator ON.
TevAXRF_GateOn(AXRF_CH1, AXRF_SRCA_GEN);
```
### TevAXRF_GateOff

**Description:** Gate off the selected AXRF source.

Normally all the AXRF sources are gated on. Individual RF sources can be gated off — the signal generator is disabled. This guarantees no signal is present.

Gating on/off does not change any of the RF source’s other setup parameters (frequency, level, etc.)

**Note:** there is only one set of RF sources for each PXI chassis (that is, shared between all eight channels). The channel parameter is only used to identify the chassis.

**Syntax:**

```cpp
TevAXRF_GateOff(
    AXRF_CHANNEL channel,
    AXRF_GENERATOR generator
);
```

**Parameters:**

- `channel` An AXRF channel:
  - AXRF_CH1, AXRF_CH2, AXRF_CH3, AXRF_CH4,
  - AXRF_CH5, AXRF_CH6, AXRF_CH7, AXRF_CH8.

- `generator` The selected RF source generator. One of the following:
  - AXRF_SRCA_GEN
  - AXRF_SRCB_GEN
  - AXRF_LO_GEN.

**Usage:**

// Gate the Source A generator OFF.
TevAXRF_GateOff(AXRF_CH1, AXRF_SRCA_GEN);

### TevAXRF_Source

**Description:** Set up the specified AXRF_CHANNEL to source an RF signal at the programmed frequency and dBm level.

**Syntax:**

```cpp
TevAXRF_Source(
    AXRF_CHANNEL channel,
    double sourceLevel,
    double freq
);
```

**Parameters:**

- `channel` An AXRF channel:
  - AXRF_CH1, AXRF_CH2, AXRF_CH3, AXRF_CH4,
  - AXRF_CH5, AXRF_CH6, AXRF_CH7, AXRF_CH8.

- `level` +12.0 dBm max.

- `freq` 50MHz to 6GHz.

**Usage:**

TevAXRF_Source(AXRF_CH1, -10.0dBm, 3.4GHz);
TevAXRF_SourceMultiChannel

Description: Set up the specified AXRF_CHANNEL to source an RF signal at the programmed frequency and dBm level.

Note: only one channel on each DRPM can be in the channel array.

Syntax:
```c
TevAXRF_SourceMultiChannel(
   AXRF_CHANNEL channelArray[],
   int numChannels,
   double sourceLevel,
   double freq
);
```

Parameters:
- **channelArray** An array of AXRF channels to source
  - AXRF_CH1, AXRF_CH2, AXRF_CH3, AXRF_CH4,
    AXRF_CH5, AXRF_CH6, AXRF_CH7, AXRF_CH8.
- **numChannels** the number of channels in the channel array:
  - 1 to 4.
- **sourceLevel** +12.0 dBm max.
- **freq** 50MHz to 6GHz.

Usage:
```c
AXRF_CHANNEL mySrcChans[2];
TevAXRF_SourceMultiChannel(mySrcChans, 2, 10.0dBm, 3.4GHz);
```

TevAXRF_SourceMultiChannelMultiLevel

Description: Set up the specified AXRF_CHANNEL to source an RF signal at the programmed frequency. This API allows each specified source channel to have a different level.

Note: only one channel on each DRPM can be in the channel array.

Syntax:
```c
TevAXRF_SourceMultiChannelMultiLevel(
   AXRF_CHANNEL channelArray[],
   int numChannels,
   double sourceArray[],
   double freq
);
```

Parameters:
- **channelArray** An array of AXRF channels to source
  - AXRF_CH1, AXRF_CH2, AXRF_CH3, AXRF_CH4,
    AXRF_CH5, AXRF_CH6, AXRF_CH7, AXRF_CH8.
- **numChannels** the number of channels in the channel array:
  - 1 to 4.
- **sourceArray** An array of levels (in dBm) to be applied to channels:
  - +12.0 dBm max.
- **freq** 50MHz to 6GHz.

Usage:
```c
AXRF_CHANNEL mySrcChans[2];
double myLevels[2];
TevAXRF_SourceMultiChannelMultiLevel(mySrcChans, 2, myLevels, 3.4GHz);
```
**TevAXRF(SourceTwoTone)**

**Description:** Source a two-tone signal on the specified channel (requires AXRF Options 11 and 12). Each tone is specified with a frequency and level.

The two tones are generated using Source A and Source B. The second VSG (AXRF Option 12) is connected to IN B on the RFSM. The individual source signals are then summed together at the front end of the RF splitter module.

**Syntax:**
```
TevAXRF_SourceTwoTone(
    AXRF_CHANNEL channel,
    double sourceLevel_1,
    double freq_1,
    double sourceLevel_2,
    double freq_2
);
```

**Parameters:**
- `channel`: An AXRF channel:
  - AXRF_CH1, AXRF_CH2, AXRF_CH3, AXRF_CH4,
  - AXRF_CH5, AXRF_CH6, AXRF_CH7, AXRF_CH8.
- `sourceLevel_1`: Level of first tone in dBm:
  - +12.0 dBm max.
- `freq_1`: Frequency of first tone:
  - 50 MHz to 6 GHz.
- `sourceLevel_2`: Level of second tone in dBm:
  - +12.0 dBm max.
- `freq_2`: Frequency of second tone:
  - 50 MHz to 6 GHz.

**Usage:**
```
// Source a 2-tone signal on channel 1
TevAXRF_SourceTwoTone(AXRF_CH1, -10.0dBm, 3.4GHz, -12.0dBm, 3.6GHz);
```
AXRF modulation syntax

The APIs included here provide access to and control of the modulation capabilities of the RF source. AXRF Option 11 or Option 13 provides a VSG that includes an arbitrary waveform generator, which can hold up to 128 MSamples of custom modulation waveform data. This data is loaded from modulation files created using **iOCreator**.

**TevAXRF_LoadModulationFile**

**Description:** Load the **iOCreator** modulation file (.aiq extension) into the source generator associated with the specified channel.

Multiple modulation files can be loaded into the source's memory. In a production program, it is common to load all the modulation files as part of the program load function.

**Syntax:**

```c
TevAXRF_LoadModulationFile(
    AXRF_CHANNEL channel,
    char* modulationFile
);
```

**Parameters:**

- **channel** An AXRF channel:
  - AXRF_CH1, AXRF_CH2, AXRF_CH3, AXRF_CH4, AXRF_CH5, AXRF_CH6, AXRF_CH7, AXRF_CH8.
- **modulationFile** Filename of an **iOCreator** .aiq modulation file. Include the full path to the file.

**Usage:**

```c
// Load a modulation file.
TevAXRF_LoadModulationFile(AXRF_CH1, "C:\ModFiles\Ex1.aiq");
```

**TevAXRF_UnloadModulationFile**

**Description:** Unload the specified modulation file (.aiq extension) from the generator associated with the specified pin(s).

To ensure that the source's modulation memory is left in a known state, ensure that all modulation files are unloaded as part of the program unload function.

**Syntax:**

```c
TevAXRF_UnloadModulationFile(
    AXRF_CHANNEL channel,
    char* modulationFile
);
```

**Parameters:**

- **channel** An AXRF channel:
  - AXRF_CH1, AXRF_CH2, AXRF_CH3, AXRF_CH4, AXRF_CH5, AXRF_CH6, AXRF_CH7, AXRF_CH8.
- **modulationFile** Filename of an **iOCreator** .aiq modulation file. Include the full path to the file.

**Usage:**

```c
// Unload a modulation file.
TevAXRF_UnloadModulationFile(AXRF_CH1, "C:\ModFiles\Ex1.aiq");
```
**TevAXRF_StartModulation**

**Description:** Start IQ modulation on the generator associated with the specified pin(s) using a modulation filename previously loaded by `TevAXRF_LoadModulationFile()`.

**Syntax:**
```
TevAXRF_StartModulation(
    AXRF_CHANNEL channel,
    char* modulationFile
);
```

**Parameters:**
- `channel` An AXRF channel:
  - AXRF_CH1, AXRF_CH2, AXRF_CH3, AXRF_CH4,
  - AXRF_CH5, AXRF_CH6, AXRF_CH7, AXRF_CH8.
- `modulationFile` Filename of an IQCreator®.aiq modulation file. Include the full path to the file.

**Usage:**
```
// Start source modulation.
TevAXRF_StartModulation(AXRF_CH1, "C:\ModFiles\Ex1.aiq");
```

**TevAXRF_StopModulation**

**Description:** Disable IQ modulation on the generator associated with the specified channel.

**Syntax:**
```
TevAXRF_StopModulation(
    AXRF_CHANNEL channel,
);
```

**Parameters:**
- `channel` An AXRF channel:
  - AXRF_CH1, AXRF_CH2, AXRF_CH3, AXRF_CH4,
  - AXRF_CH5, AXRF_CH6, AXRF_CH7, AXRF_CH8.

**Usage:**
```
// Stop source modulation.
TevAXRF_StopModulation(AXRF_CH1);
```
TevAXRF_ModulationTriggerArm

Description: In addition to calling APIs to start and stop the modulation source, a variety of PXI signals can be used to trigger/gate the modulation source signal generation. This API sets up (arms) the modulation source to turn on when it detects the specified trigger signal.

   External trigger sources are:
   
   - PXI backplane (PXI trigger bus)
   - LVDS AUXiliary inputs (front-panel DATA connector)
   - TTL TRIG input on front panel (SMB)
   - Serial local bus

   Input, TTL +ve or –ve edge. SMB socket, 50 Ω.

Syntax:
```c
TevAXRF_ModulationTriggerArm(
    AXRF_CHANNEL channel,
    afSigGenDll_rmRoutingMatrix_t triggerSource,
    BOOL gate,
    BOOL negativeEdge
);
```
Parameters:

- **channel**: Used to identify the PXI chassis containing the modulation source. An AXRF channel:
  - AXRF_CH1, AXRF_CH2, AXRF_CH3, AXRF_CH4,
  - AXRF_CH5, AXRF_CH6, AXRF_CH7, AXRF_CH8.

- **triggerSource**: Specify the signal to be used as a trigger for the modulation source. One of the following:

  afSigGenDll_rmPXI_TRIG_0, afSigGenDll_rmPXI_TRIG_1,
  afSigGenDll_rmPXI_TRIG_2, afSigGenDll_rmPXI_TRIG_3,
  afSigGenDll_rmPXI_TRIG_4, afSigGenDll_rmPXI_TRIG_5,
  afSigGenDll_rmPXI_TRIG_6, afSigGenDll_rmPXI_TRIG_7,
  afSigGenDll_rmPXI_STAR,
  afSigGenDll_rmPXI_LBL_0, afSigGenDll_rmPXI_LBL_1,
  afSigGenDll_rmPXI_LBL_2, afSigGenDll_rmPXI_LBL_3,
  afSigGenDll_rmPXI_LBL_4, afSigGenDll_rmPXI_LBL_5,
  afSigGenDll_rmPXI_LBL_6, afSigGenDll_rmPXI_LBL_7,
  afSigGenDll_rmPXI_LBL_8, afSigGenDll_rmPXI_LBL_9,
  afSigGenDll_rmPXI_LBL_10, afSigGenDll_rmPXI_LBL_11,
  afSigGenDll_rmPXI_LBL_12,
  afSigGenDll_rmLVDS_MARKER_1, afSigGenDll_rmLVDS_MARKER_2,
  afSigGenDll_rmLVDS_MARKER_3, afSigGenDll_rmLVDS_MARKER_4,
  afSigGenDll_rmLVDS_AUX_0, afSigGenDll_rmLVDS_AUX_1,
  afSigGenDll_rmLVDS_AUX_2, afSigGenDll_rmLVDS_AUX_3,
  afSigGenDll_rmLVDS_AUX_4,
  afSigGenDll_rmLVDS_SPARE_0, afSigGenDll_rmLVDS_SPARE_1,
  afSigGenDll_rmLVDS_SPARE_2,
  afSigGenDll_rmARB_MARKER_1, afSigGenDll_rmARB_MARKER_2,
  afSigGenDll_rmARB_MARKER_3, afSigGenDll_rmARB_MARKER_4,
  afSigGenDll_rmARB_TRIG,
  afSigGenDll_rmLA_OUT_0, afSigGenDll_rmLA_OUT_1,
  afSigGenDll_rmLA_OUT_2, afSigGenDll_rmLA_OUT_3,
  afSigGenDll_rmLA_OUT_4, afSigGenDll_rmLA_OUT_5,
  afSigGenDll_rmLA_OUT_6, afSigGenDll_rmLA_OUT_7,
  afSigGenDll_rmLSTB_OUT,
  afSigGenDll_rmLA_IN_0, afSigGenDll_rmLA_IN_1,
  afSigGenDll_rmLA_IN_2, afSigGenDll_rmLA_IN_3,
  afSigGenDll_rmLA_IN_4, afSigGenDll_rmLA_IN_5,
  afSigGenDll_rmLA_IN_6, afSigGenDll_rmLA_IN_7,
  afSigGenDll_rmRFOFF_EXT, afSigGenDll_rmMODOFF_EXT,
  afSigGenDll_rmFREEZE_EXT, afSigGenDll_rmGND,
  afSigGenDll_rmSeqStart, afSigGenDll_rmRfBlank,
  afSigGenDll_rmLSTB_IN,
  afSigGenDll_rmFRONT_SMB,
  afSigGenDll_rsmSW,
  afSigGenDll_rmLA_SERIAL,
  afSigGenDll_rmTRIG_GATE_EN,
  afSigGenDll_rmTRIG_GATE_SIG,
  afSigGenDll_rmTRIG_GATE_OUT.
gate Controls trigger/gate behavior.
FALSE ‘Trigger’ mode.
Start the modulation source on the trigger edge (normally Positive).
The source remains on until manually stopped with
TevAXRF_StopModulation().

TRUE ‘Gate’ mode.
The modulation source is enabled whenever the trigger signal is ON
(normally Hi).
The source is disabled whenever the trigger signal is OFF (normally Lo).

Usage: // Setup modulation source to start when it detects a
// positive-going signal on the RF Source front panel
// SMB connector.
TevAXRF_ModulationTriggerArm(AXRF_CH1,
    afSigGenDll_rmFRONT_SMB, // Trigger input from front panel SMB
    FALSE, // Trigger mode
    FALSE); // Positive edge

// Modulation source start if/when positive edge is detected
// . . .
TevAXRF_StopModulation(AXRF_CH1); // Stop modulation source
**AXRF measurement syntax**

The APIs included here control the signal conditioning and routing of the RF signals to be measured as well as the actual digitizing and calculations needed to analyze RF signals.

All RF measurements use the RF digitizer. The two key parameters for the digitizer are:

- Sample size — the number of samples captured for each measurement
- Sample frequency — the rate at which the samples are taken.

The table below summarizes the defaults and programmability of the sample size and sample frequency for the types of RF measurements that can be made by the AXRF RF Subsystem.

<table>
<thead>
<tr>
<th>Measurement type</th>
<th>Sample size</th>
<th>Sample freq</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scalar</td>
<td>Default= 2048 &lt;br&gt; _SetMeasureSamples()</td>
<td>FIXED — 250MHz</td>
</tr>
<tr>
<td>Return loss</td>
<td>Default= 2048 &lt;br&gt; _SetMeasureSamples()</td>
<td>FIXED — 250MHz</td>
</tr>
<tr>
<td>S-parameters</td>
<td>Default= 2048 &lt;br&gt; _SetMeasureSamples()</td>
<td>FIXED — 250MHz</td>
</tr>
<tr>
<td>IQ</td>
<td>_MeasureSetupIQ() &lt;br&gt; Based on sampleFrequency, RBW</td>
<td>_MeasureSetupIQ()</td>
</tr>
<tr>
<td>IQ data</td>
<td>_MeasureArrayIQ0</td>
<td>Default= 250 MHz &lt;br&gt; _SetIQSampleFrequency()</td>
</tr>
<tr>
<td>Noise figure</td>
<td>_MeasureNoiseFigureSetup()</td>
<td>Default= 8 MHz &lt;br&gt; _SetNoiseFigureSampleFrequency()</td>
</tr>
</tbody>
</table>

**IQ mode measurements**

IQ mode measurements provide several benefits over the basic scalar measurement. The additional flexibility of being able to adjust the digitizer sampling frequency, effective resolution bandwidth (RBW) and measurement frequency span are very useful in making accurate, low-power CW measurements for harmonics, etc. But smaller RBWs are generally achieved by using larger sample sizes, which in turn require longer FFT calculation time.

The AXRF Control Panel provides an interactive way to experiment with the IQ mode parameters and to design RF measurements to meet specific needs. Once the proper parameters values are determined, they can be transferred to the API calls in the test program.
TevAXRF_MeasureSetup

Description: Set up the specified channel to measure an RF signal at the specified measure frequency (freq) with the specified measureLevel (in dBm).

*Note: default digitizer sample size for a return loss measurement is 2048.*
Sample size can be adjusted using TevAXRF_SetMeasureSamples().
Sampling frequency is fixed at 250 MHz as IF mode of 3035 is being used.

Syntax: TevAXRF_MeasureSetup(
   AXRF_CHANNEL channel,
   double measureLevel,
   double freq
);

Parameters: channel An AXRF channel. One of the following:
   AXRF_CH1, AXRF_CH2, AXRF_CH3, AXRF_CH4,
   AXRF_CH5, AXRF_CH6, AXRF_CH7, AXRF_CH8.

measureLevel Expected measurement level in dBm:
   +20 dBm max.

freq Expected measurement frequency:
   50 MHz to 6 GHz.

Example: TevAXRF_MeasureSetup(AXRF_CH2, -10.0 dBm, 3.4 GHz);
**TevAXRF_MeasureSetupMultiChannel**

**Description:** Set up the specified channel to measure an RF signal at the specified measure frequency (freq) with the specified measureLevel (in dBm). The TevAXRF_MeasureMultiChannel() API should be used to make a multi-channel measurement.

*Note: Only one channel on each DRPM can be in the channel array.*

*Default digitizer sample size for a scalar measurement is 2048.*

*Sample size can be adjusted using TevAXRF_SetMeasureSamples().*

*Sampling frequency is fixed at 250 MHz.*

**Syntax:**

```c
TevAXRF_MeasureSetupMultiChannel(
    AXRF_CHANNEL channelArray[],
    int numChannels,
    double measureLevel,
    double freq
);
```

**Parameters:**

- **channelArray** An array of AXRF channels to set up for measurement
  - AXRF_CH1, AXRF_CH2, AXRF_CH3, AXRF_CH4,
  - AXRF_CH5, AXRF_CH6, AXRF_CH7, AXRF_CH8.

- **numChannels** The number of channels in the channel array:
  - 1 to 4.

- **measureLevel** Expected measurement level in dBm:
  - +20 dBm max.

- **freq** Expected measurement frequency:
  - 50MHz to 6GHz.

**Usage:**

```c
AXRF_CHANNEL myMeasChans[2];
TevAXRF_MeasureSetupMultiChannel(myMeasChans, 2, -10.0dBm, 3.4GHz);
```
TevAXRF_MeasureSetupIQ

Description: Set up the specified channel to measure an RF signal at the specified measureFrequency with the specified measureLevel (in dBm) using IQ data. The digitizer’s sampling frequency and effective resolution bandwidth can be programmed. The setup also specifies a range of frequency data that can be returned (as input to a display tool for example).

Use TevAXRF_MeasureIQ() to make measurements based on this setup.

Note: this API uses the lower level Cobham function afSpectrumDll() to perform the required FFT. Part of the setup determines the minimum FFT size needed to achieve the specified RBW.

\[
\text{MinimumFFTsize}= 2^{\left\lfloor \log_2 (\frac{\text{sampleFrequency}}{\text{RBW} \times \text{ENB bins}}) \right\rfloor}
\]

where \( \text{ENB bins} = 2.5 \)

The digitizer can generate FFTs with number of points ranging from 128 to 65536.

Syntax:

```c
int TevAXRF_MeasureSetupIQ(
    AXRF_CHANNEL channel,
    double measureLevel,
    double measureFrequency,
    double sampleFrequency,
    double RBW,
    double measurementSpan
);
```

Parameters:

- **channel**: An AXRF channel. One of the following: AXRF_CH1, AXRF_CH2, AXRF_CH3, AXRF_CH4, AXRF_CH5, AXRF_CH6, AXRF_CH7, AXRF_CH8.
- **measureLevel**: Expected measurement level in dBm: +20 dBm max.
- **measureFrequency**: Expected measurement frequency: 50MHz to 6GHz.
- **sampleFrequency**: Sampling frequency for digitizer (see Note): 250 MHz max.
- **RBW**: Resolution bandwidth target (see Note).
- **measurementSpan**: Measurement frequency span (around the center frequency).

Return variable: This API returns the following based on sampleFrequency and RBW:

- **sample size** of data to be collected
- \(-1\) if a valid FFT cannot be met.

Example:

```c
int sampleSize;
sampleSize= TevAXRF_MeasureSetupIQ(
    AXRF_CH2, -10.0 dBm, 3.4 GHz, 200 MHz, 100 KHz, 10 Mhz);
```
TevAXRF_MeasureSetupIQMultiChannel

Description: This is the multi-channel version of TevAXRF_MeasureSetupIQ(). Only one channel on each DRPM can be in the channel array.

Set up the specified channel to measure an RF signal at the specified measureFrequency with the specified measureLevel (in dBm) using IQ data. The digitizer's sampling frequency and effective resolution bandwidth can be programmed. The setup also specifies a range of frequency data that can be returned (as input to a display tool for example).

Use TevAXRF_MeasureIQMultiChannel() to make measurements based on this setup.

Note: this API uses the lower level Cobham function afSpectrumDll() to perform the required FFT. Part of the setup determines the minimum FFT size needed to achieve the specified RBW.

\[
\text{MinimumFFTsize} = 2 \left\lfloor \log_2 \left( \frac{\text{sampleFrequency}}{\text{RBW} \times \text{ENB bins}} \right) \right\rfloor
\]

where \( \text{ENB bins} = 2.5 \)

The digitizer can generate FFTs with number of points ranging from 128 to 65536.

Syntax:

```c
int TevAXRF_MeasureSetupIQMultiChannel(
    AXRF_CHANNEL channelArray[],
    int numChannels,
    double measureLevel,
    double measureFrequency,
    double sampleFrequency,
    double RBW,
    double measurementSpan
);
```

Parameters:

- **channelArray**: An array of AXRF channels to set up for measurement:
  - AXRF_CH1, AXRF_CH2, AXRF_CH3, AXRF_CH4,
  - AXRF_CH5, AXRF_CH6, AXRF_CH7, AXRF_CH8.

- **numChannels**: The number of channels in the channel array:
  - 1 to 4.

- **measureLevel**: Expected measurement level in dBm:
  - +20 dBm max.

- **measureFrequency**: Expected measurement frequency:
  - 50MHz to 6GHz.

- **sampleFrequency**: Sampling frequency for digitizer (see Note):
  - 250 MHz max.

- **RBW**: Resolution bandwidth target (see Note).

- **measurementSpan**: Measurement frequency span (around the center frequency).

Return variable: This API returns the following based on sampleFrequency and RBW:

- sample size of data to be collected
- –1 if a valid FFT cannot be met.

Example:

```c
AXRF_CHANNEL myMeasChans[2];
TevAXRF_MeasureSetupIQMultiChannel(myMeasChans, 2,
    -10.0 dBm, 3.4 GHz, 200 MHz, 100 KHz, 10 Mhz);
```
**TevAXRF_MeasureReturnLossSetup**

**Description:** Set up the specified channel to measure return loss. The measurement is made using an RF signal at the specified frequency (`freq`) with the specified `sourceLevel` (in dBm).

*Note: default digitizer sample size for a return loss measurement is 2048.*

Sample size can be adjusted using `TevAXRF_SetMeasureSamples()`. **Sampling frequency is fixed at 250 MHz.**

**Syntax:**
```
TevAXRF_MeasureReturnLossSetup(
    AXRF_CHANNEL channel,
    double sourceLevel,
    double freq
);
```

**Parameters:**
- **channel** An AXRF channel. One of the following:
  - AXRF_CH1, AXRF_CH2, AXRF_CH3, AXRF_CH4,
  - AXRF_CH5, AXRF_CH6, AXRF_CH7, AXRF_CH8.
- **sourceLevel** Source level in dBm:
  - +20 dBm max.
- **freq** Source frequency:
  - 50MHz to 6GHz.

**Usage:**
```
TevAXRF_MeasureReturnLossSetup(AXRF_CH1, 0.5dBm, 3.4GHz);
```
AXRF measurement syntax

TevAXRF_MeasureReturnLossSetupMultiChannel

Description: Set up the specified channels to measure return loss. The measurement is made using an RF signal at the specified frequency \( f_{\text{req}} \) with the specified level in dBm.

Only one channel from each channel pair (corresponding to a DRPM) can be specified in the channelArray. This is checked by the software driver.

Note: Only one channel on each DRPM can be in the channel array.
Default digitizer sample size for a scalar measurement is 2048. Sample size can be adjusted using TevAXRF_SetMeasureSamples(). Sampling frequency is fixed at 250 MHz.

Syntax: TevAXRF_MeasureReturnLossSetupMultiChannel(
    AXRF_CHANNEL channelArray[],
    int numChannels,
    double level,
    double freq
);

Parameters:

- **channelArray**: An array of AXRF channels to set up for measurement:
  - AXRF_CH1, AXRF_CH2, AXRF_CH3, AXRF_CH4,
  - AXRF_CH5, AXRF_CH6, AXRF_CH7, AXRF_CH8.

- **numChannels**: The number of channels in the channel array:
  - 1 to 4.

- **level**: Source level in dBm:
  - +20 dBm max.

- **freq**: Expected measurement frequency:
  - 50MHz to 6GHz.

Usage:

```c
AXRF_CHANNEL myMeasChans[2];

TevAXRF_MeasureReturnLossSetupMultiChannel(myMeasChans, 2, 0.5dBm, 3.4GHz);
```
Description: Set up the specified DRPM to perform a two-port S-parameter measurement. The measurement setup expects the specified channel to be connected to the device input (Port 1) and the other channel of the same DRPM to be connected to the device output (Port 2). Each DRPM has two channels (ports) making up a 'channel pair': AXRF_CH1 and AXRF_CH2, AXRF_CH3 and AXRF_CH4, AXRF_CH5 and AXRF_CH6, AXRF_CH7 and AXRF_CH8. RF channels 2, 4, 6 and 8 correspond to AXRF Options 16, 17, 18 and 19.

The measurement is made by applying an RF signal to the input port at the specified frequency (freq) with the specified sourceLevel (in dBm). The measureLevel specifies the expected level of the signal at the output port.

**Note:** default digitizer sample size for an S-parameters measurement is 2048. Sample size can be adjusted using TevAXRF_SetMeasureSamples(). Sampling frequency is fixed at 250 MHz.

Syntax:
```
TevAXRF_MeasureSparametersSetup(
    AXRF_CHANNEL channel,
    double sourceLevel,
    double measureLevel,
    double freq
);
```

Parameters:
- **channel** An AXRF channel to be used as Port 1 (input) of the S-parameter measurement. The other channel of the same DRPM is Port 2 (output) of the S-parameter measurement. One of the following: AXRF_CH1, AXRF_CH2, AXRF_CH3, AXRF_CH4, AXRF_CH5, AXRF_CH6, AXRF_CH7, AXRF_CH8.
- **sourceLevel** Input level in dBm: +12 dBm max.
- **measureLevel** Expected level to be measured in dBm.
- **freq** Input frequency: 50MHz to 6GHz.

Usage: TevAXRF_MeasureSparametersSetup(AXRF_CH1, -10dBm, 0dBm, 3.4GHz);
TevAXRF_MeasureSparametersSetupMultiChannel

Description: Set up multiple DRPMs to perform a two-port S-parameter measurement. The measurement setup expects each specified channel to be connected to the device input (Port 1) and the other channel of the same DRPM to be connected to the device output (Port 2). Each DRPM has two channels (ports) making up a 'channel pair': AXRF_CH1 and AXRF_CH2, AXRF_CH3 and AXRF_CH4, AXRF_CH5 and AXRF_CH6, AXRF_CH7 and AXRF_CH8.

The measurement is made by applying an RF signal to all of the input ports at the specified frequency (freq) with the specified sourceLevel (in dBm). The measureLevel specifies the expected level of the signal at the output ports.

Note: only one channel on each DRPM can be in the channel array. The default digitizer sample size for an S-parameters measurement is 2048. Sample size can be adjusted using TevAXRF_SetMeasureSamples(). Sampling frequency is fixed at 250 MHz.

Syntax: TevAXRF_MeasureSparametersSetupMultiChannel(
    AXRF_CHANNEL channelArray[],
    int numChannels
    double sourceLevel,
    double measureLevel,
    double freq
);

Parameters: channelArray An array of AXRF channels to be used as Port 1 (input) of an S-parameter measurement. The other channel of the same DRPM is Port 2 (output) of the S-parameter measurement. One of the following: AXRF_CH1, AXRF_CH2, AXRF_CH3, AXRF_CH4, AXRF_CH5, AXRF_CH6, AXRF_CH7, AXRF_CH8.

numChannels The number of channels in the channel array:
1 to 4.

sourceLevel Input level in dBm:
+12 dBm max.

measureLevel Expected level to be measured in dBm.

freq Input frequency:
50MHz to 6GHz.

Usage: AXRF_CHANNEL myMeasChans[2];

TevAXRF_MeasureSparametersSetupMultiChannel(  
    myMeasChans, 2, -10dBm, 0dBm, 3.4GHz);


TevAXRF_MeasureNoiseFigureSetup

Description: Noise figure measurements are made using an IQ measurement from the AXRF digitizer. The digitizer’s sampling clock is changed to 8 MHz (normally it is 250 MHz).

As part of setting up the measurement, the AXRF source generators are gated OFF. The noise source, which is located on the RF Splitter Module (RFSM), is connected to the sourceChannel. The noise source is gated ON at this point.

The measureChannel is connected to the digitizer. Best results are obtained when the sourceChannel and measureChannel are on the same DRPM (1-2, 3-4, 5-6, 7-8).

The noise figure measurement is made using TevAXRF_MeasureNoiseFigureYfactor().

Syntax: TevAXRF_MeasureNoiseFigureSetup(
    AXRF_CHANNEL sourceChannel,
    AXRF_CHANNEL measureChannel,
    double frequency,
    int numberPoints);

Parameters:

sourceChannel  An AXRF channel. One of the following:
    AXRF_CH1, AXRF_CH2, AXRF_CH3, AXRF_CH4,
    AXRF_CH5, AXRF_CH6, AXRF_CH7, AXRF_CH8.

measureChannel  An AXRF channel. One of the following:
    AXRF_CH1, AXRF_CH2, AXRF_CH3, AXRF_CH4,
    AXRF_CH5, AXRF_CH6, AXRF_CH7, AXRF_CH8.

frequency  Center frequency for the noise measurement.

numPoints  Number of samples to be captured per measurement:
    2 to max of digitizer memory.

Usage: TevAXRF_MeasureNoiseFigureSetup(AXRF_CH1, AXRF_CH2, 1.007e9, 32768);
TevAXRF_MeasureTriggerArm

Description: This API is used to arm the AXRF digitizer when making scalar measurements. Rather than starting a measurement directly from an API call, a variety of PXI signals can be used to trigger the measurement process (specifically the digitizing of the signal). This API selects a PXI signal or other external signal to be used as the digitizer’s trigger. It also arms the digitizer; measurement begins when the selected trigger signal is detected. A timeout is provided to ensure that program execution can continue if the expected trigger signal does not occur within a certain period of time.

Use of the digitizer’s trigger capability requires a specific sequence of API calls as shown in the Usage section below.

Syntax:
TevAXRF_MeasureTriggerArm(
   AXRF_CHANNEL channel,
   afDigitizerDll_tsTrigSource_t triggerSource,
   afDigitizerDll_egpPolarity_t edgeGatePolarity,
   double timeout
);

Parameters:
- channel: An AXRF channel. One of the following:
  AXRF_CH1, AXRF_CH2, AXRF_CH3, AXRF_CH4, AXRF_CH5, AXRF_CH6, AXRF_CH7, AXRF_CH8.
- triggerSource: PXI trigger signal to be used to start a measurement.
  One of the following:
  afDigitizerDll_tsPXI_TRIG_0, afDigitizerDll_tsPXI_TRIG_1, afDigitizerDll_tsPXI_TRIG_2, afDigitizerDll_tsPXI_TRIG_3, afDigitizerDll_tsPXI_TrIG_4, afDigitizerDll_tsPXI_TRIG_5, afDigitizerDll_tsPXI_TRIG_6, afDigitizerDll_tsPXI_TRIG_7, afDigitizerDll_tsPXI_STAR, afDigitizerDll_tsPXI_LBL_0, afDigitizerDll_tsPXI_LBL_1, afDigitizerDll_tsPXI_LBL_2, afDigitizerDll_tsPXI_LBL_3, afDigitizerDll_tsPXI_LBL_4, afDigitizerDll_tsPXI_LBL_5, afDigitizerDll_tsPXI_LBL_6, afDigitizerDll_tsPXI_LBL_7, afDigitizerDll_tsPXI_LBL_8, afDigitizerDll_tsPXI_LBL_9, afDigitizerDll_tsPXI_LBL_10, afDigitizerDll_tsPXI_LBL_11, afDigitizerDll_tsPXI_LBL_12, afDigitizerDll_tsLVDS_MARKER_0, afDigitizerDll_tsLVDS_MARKER_1, afDigitizerDll_tsLVDS_MARKER_2, afDigitizerDll_tsLVDS_MARKER_3, afDigitizerDll_tsLVDS_MARKER_4, afDigitizerDll_tsLVDS_AUX_0, afDigitizerDll_tsLVDS_AUX_1, afDigitizerDll_tsLVDS_AUX_2, afDigitizerDll_tsLVDS_AUX_3, afDigitizerDll_tsLVDS_AUX_4, afDigitizerDll_tsLVDS_SPARE_0, afDigitizerDll_tsSW_TRIG, afDigitizerDll_tsINT_TIMER, afDigitizerDll_tsINT_TRIG, afDigitizerDll_tsFRONT_SMB.
- edgeGatePolarity: Polarity of PXI trigger signal used to start a measurement.
  One of the following:
  afDigitizerDll_egpPositive, afDigitizerDll_egpNegative.
- timeout: Timeout duration.
  Execution continues if trigger has not occurred by this time.
Usage:

double result;

// Setup the measurement
TevAXRF_MeasureSetup(AXRF_CH2, -10.0 dBm, 3.4 GHz);

// Setup and arm the digitizer trigger
TevAXRF_MeasureTriggerArm(AXRF_CH2,
    afDigitizerDll_tspXI_TRIG_1,
    afDigitizerDll_egpPositive,
    500 ms);

// Trigger condition occurs ➔ Digitizer captures signal

// The measure API waits for the trigger to occur or
// for the timeout time to elapse before continuing.

// Final measurement is calculated and returned
result = TevAXRF_Measure(AXRF_CH2);
TevAXRF_MeasureIntTriggerSetup

Description: Provides control of the Digitizer's trigger mode.

By default the digitizer is triggered in the absolute level mode.

In absolute trigger mode, the digitized signal is filtered using an absolute time constant. An internal level trigger is generated when the level of this filtered signal exceeds the absolute level trigger threshold (specified in dBm). The absolute time constant and level settings may affect the trigger delay.

In relative trigger mode, the digitized signal is filtered using both fast and slow time constants. The amplitude difference between the fast and slow time constant filtered signal is compared with the 'relative threshold level'. For a step level change, the difference signal is a pulse with a duration and level determined by the difference between the fast and slow time constants. When the relative threshold trigger level is entered as positive, the difference signal = (fast signal − slow signal); when the relative threshold trigger level is entered as negative, the difference signal = (slow signal − fast signal).

Syntax: TevAXRF_MeasureIntTriggerSetup(
    AXRF_CHANNEL channel,
    afDigitizerDll_itmIntTriggerMode_t internalTriggerMode,
    double thresholdLevel
);

Parameters:

channel An AXRF channel (used to identify Digitizer):
    AXRF_CH1, AXRF_CH2, AXRF_CH3, AXRF_CH4,
    AXRF_CH5, AXRF_CH6, AXRF_CH7, AXRF_CH8.

internalTriggerMode

    afDigitizerDll_itmAbsolute absolute level trigger mode
    afDigitizerDll_itmRelative relative level trigger mode

thresholdLevel the threshold level (in dBm) used to generate an internal trigger.

Usage: TevAXRF_MeasureIntTriggerSetup(AXRF_CH2, afDigitizerDll_itmRelative, 10 dBm);
**TevAXRF_MeasureTriggerOffsetDelay**

**Description:** The capture of Digitizer data can be delayed from the hardware trigger signal. This delay is specified as an offset in number of samples.

**Syntax:**
```
TevAXRF_MeasureTriggerOffsetDelay(
    AXRF_CHANNEL channel,
    int numberOfSamples
);
```

**Parameters:**
- **channel**
  - An AXRF channel (used to identify Digitizer):
  - AXRF_CH1, AXRF_CH2, AXRF_CH3, AXRF_CH4,
  - AXRF_CH5, AXRF_CH6, AXRF_CH7, AXRF_CH8.
- **numberOfSamples**
  - the delay as a number of samples.

**Usage:**
```
TevAXRF_MeasureTriggerOffsetDelay(AXRF_CH2, 2001);
```

**TevAXRF_MeasureTriggerPreEdgeSamples**

**Description:** Specify the number of pre-trigger samples to be captured once a Digitizer trigger signal is detected.

**Syntax:**
```
TevAXRF_MeasureTriggerPreEdgeSamples(
    AXRF_CHANNEL channel,
    int numberOfSamples
);
```

**Parameters:**
- **channel**
  - An AXRF channel (used to identify Digitizer):
  - AXRF_CH1, AXRF_CH2, AXRF_CH3, AXRF_CH4,
  - AXRF_CH5, AXRF_CH6, AXRF_CH7, AXRF_CH8.
- **numberOfSamples**
  - the number of pre-trigger samples to capture.

**Usage:**
```
TevAXRF_MeasureTriggerPreEdgeSamples(AXRF_CH2, 1001);
```
**TevAXRF_MeasureIQTriggerArm**

**Description:** This API is used to arm the AXRF digitizer when making IQ measurements. Rather than starting a measurement directly from an API call, a variety of PXI signals can be used to trigger the measurement process (specifically the digitizing of the signal). This API selects a PXI signal or other external signal to be used as the digitizer’s trigger. It also arms the digitizer; measurement begins when the selected trigger signal is detected. A timeout is provided to ensure that program execution can continue if the expected trigger signal does not occur within a certain period of time.

Use of the digitizer’s trigger capability requires a specific sequence of API calls as shown in the Usage section below.

**Syntax:**
```
TevAXRF_MeasureIQTriggerArm(
    AXRF_CHANNEL channel,
    int numberOfPoints,
    afDigitizerDll_tsTrigSource_t triggerSource,
    afDigitizerDll_egpPolarity_t edgeGatePolarity,
    double timeout
);
```

**Parameters:**
- `channel`: An AXRF channel. One of the following:
  - AXRF_CH1, AXRF_CH2, AXRF_CH3, AXRF_CH4,
  - AXRF_CH5, AXRF_CH6, AXRF_CH7, AXRF_CH8.
- `numberOfPoints`: Number of points to digitize.
- `triggerSource`: PXI trigger signal to be used to start a measurement. One of the following:
  - afDigitizerDll_tsPXI_TRIG_0, afDigitizerDll_tsPXI_TRIG_1,
  - afDigitizerDll_tsPXI_TRIG_2, afDigitizerDll_tsPXI_TRIG_3,
  - afDigitizerDll_tsPXI_TRIG_4, afDigitizerDll_tsPXI_TRIG_5,
  - afDigitizerDll_tsPXI_TRIG_6, afDigitizerDll_tsPXI_TRIG_7,
  - afDigitizerDll_tsPXI_STAR,
  - afDigitizerDll_tsPXI_LBL_0, afDigitizerDll_tsPXI_LBL_1,
  - afDigitizerDll_tsPXI_LBL_2, afDigitizerDll_tsPXI_LBL_3,
  - afDigitizerDll_tsPXI_LBL_4, afDigitizerDll_tsPXI_LBL_5,
  - afDigitizerDll_tsPXI_LBL_6, afDigitizerDll_tsPXI_LBL_7,
  - afDigitizerDll_tsPXI_LBL_8, afDigitizerDll_tsPXI_LBL_9,
  - afDigitizerDll_tsPXI_LBL_10, afDigitizerDll_tsPXI_LBL_11,
  - afDigitizerDll_tsPXI_LBL_12,
  - afDigitizerDll_tsLVDS_MARKER_0, afDigitizerDll_tsLVDS_MARKER_1,
  - afDigitizerDll_tsLVDS_MARKER_2, afDigitizerDll_tsLVDS_MARKER_3,
  - afDigitizerDll_tsLVDS_MARKER_4,
  - afDigitizerDll_tsLVDS_AUX_0, afDigitizerDll_tsLVDS_AUX_1,
  - afDigitizerDll_tsLVDS_AUX_2, afDigitizerDll_tsLVDS_AUX_3,
  - afDigitizerDll_tsLVDS_AUX_4, afDigitizerDll_tsLVDS_SPARE_0,
  - afDigitizerDll_tsSW_TRIG,
  - afDigitizerDll_tsINT_TIMER, afDigitizerDll_tsINT_TRIG,
  - afDigitizerDll_tsFRONT_SMB.
- `edgeGatePolarity`: Polarity of PXI trigger signal used to start a measurement. One of the following:
  - afDigitizerDll_egpPositive
  - afDigitizerDll_egpNegative.
timeout

Timeout duration. Execution continues if trigger has not occurred by this time.

Usage:

```c
float Ires[1024], Qres[1024]; // Result arrays

TevAXRF_MeasureSetup(AXRF_CH2, -10.0 dBm, 3.4 GHz); // Setup measurement

// Setup and arm the digitizer trigger
TevAXRF_MeasureIQTriggerArm(AXRF_CH2, 1024,
  afDigitizerDll_tsPXI_TRIG_1,
  afDigitizerDll_egpPositive,
  500 mS);

// Trigger condition occurs ➔ Digitizer captures signal

// The measure API waits for the trigger to occur or
// for the timeout time to elapse before continuing.

// I and Q data is calculated and returned
TevAXRF_MeasureArrayIQ(AXRF_CH2, 1024, Ires, Qres);
```
TevAXRF_Measure

Description: Perform a standard AXRF_SCALAR measurement of the RF signal on the specified AXRF_CHANNEL and return the dBm signal level of the measured signal. The AXRF_CHANNEL is set up with the TevAXRF_MeasureSetup() API.

**Note:** default digitizer sample size for a scalar measurement is 2048. Sampling frequency is fixed at 250 MHz.

Syntax:
```c
double TevAXRF_Measure(
    AXRF_CHANNEL channel
);
```

Parameters:
- `channel` An AXRF channel. One of the following:
  - AXRF_CH1, AXRF_CH2, AXRF_CH3, AXRF_CH4, AXRF_CH5, AXRF_CH6, AXRF_CH7, AXRF_CH8.

Return variable This API returns the result of the measurement as a double.

Usage:
```c
double result;

// Setup the measurement
TevAXRF_MeasureSetup(AXRF_CH2, -10.0 dBm, 3.4 GHz);

// Make the measurement
result= TevAXRF_Measure(AXRF_CH2);
```
TevAXRF_MeasureMultiChannel

Description: Perform a standard AXRF_SCALAR measurement of the RF signal on the specified AXRF_CHANNELs and return the dBm signal level of each measured signal. The AXRF_CHANNELs are setup with the TevAXRF_MeasureSetupMultiChannel() API.

The RF Mux module is used to switch between the channels to make each measurement.

**Note:** only one channel on each DRPM can be in the channel array.

Default digitizer sample size for a scalar measurement is **2048**.

Sampling frequency is fixed at **250 MHz**.

Syntax:
```c
TevAXRF_MeasureMultiChannel(
    AXRF_CHANNEL channelArray[],
    int numChannels,
    double resultsArray[]
);
```

Parameters:
- **channelArray** An array of AXRF channels to set up for measurement:
  - AXRF_CH1, AXRF_CH2, AXRF_CH3, AXRF_CH4,
  - AXRF_CH5, AXRF_CH6, AXRF_CH7, AXRF_CH8.
- **numChannels** The number of channels in the channel array:
  - 1 to 4.
- **resultsArray** Array to return the result(s).

Usage:
```c
AXRF_CHANNEL myMeasChans[2];
double     results[2];

// Setup
TevAXRF_MeasureSetupMultiChannel(myMeasChans, 2, -10.0dBm,
    3.4GHz);

// Measure
TevAXRF_MeasureMultiChannel(myMeasChans, 2, results);
```
TevAXRF_MeasureIQ

Description: Digitize signal, transform IQ data, measure the power on the specified AXRF_CHANNEL and return the dBm signal level at the previously specified measure frequency. The measurement parameters must have been previously set up with TevAXRF_MeasureSetupIQ().

Syntax: TevAXRF_MeasureIQ(
    AXRF_CHANNEL channel,
    double *power
);

Parameters:

- **channel** An AXRF channel:
  - AXRF_CH1, AXRF_CH2, AXRF_CH3, AXRF_CH4,
  - AXRF_CH5, AXRF_CH6, AXRF_CH7, AXRF_CH8.

- **power** Variable to hold the measurement result.

Usage:
```c
double result;

// Setup the measurement
TevAXRF_MeasureSetupIQ(AXRF_CH2, -10.0 dBm, 3.4 GHz, 200 MHz, 100 KHz, 10 Mhz);

// Make the measurement
TevAXRF_MeasureIQ(AXRF_CH2, &result);
```
**TevAXRF_MeasureIQMultiChannel**

**Description:** This is the multi-channel version of `TevAXRF_MeasureIQ()`.

Digitize signal, transform IQ data, measure the power on the specified `AXRF_CHANNEL`s and return the dBm signal level at the previously specified measure frequency. The measurement parameters must have been previously set up with `TevAXRF_MeasureSetupIQMultiChannel()`. The RF Mux module is used to switch between the channels to make each measurement.

*Note:* only one channel on each DRPM can be in the channel array.

**Syntax:**

```c
TevAXRF_MeasureIQMultiChannel(
    AXRF_CHANNEL channelArray[],
    int numChannels,
    double resultsArray[]
);
```

**Parameters:**

- `channelArray`: An array of AXRF channels to measure:
  - `AXRF_CH1`, `AXRF_CH2`, `AXRF_CH3`, `AXRF_CH4`,
  - `AXRF_CH5`, `AXRF_CH6`, `AXRF_CH7`, `AXRF_CH8`.
- `numChannels`: The number of channels in the channel array:
  - 1 to 4.
- `resultsArray`: The array to return the result(s).

**Usage:**

```c
AXRF_CHANNEL myMeasChans[2];
double      results[2];

// Setup the measurements
TevAXRF_MeasureIQSetupMultiChannel(myMeasChans, 2, -10.0 dBm, 3.4 GHz, 200 MHz, 100 KHz, 10 Mhz);

// Make the measurements
TevAXRF_MeasureIQMultiChannel(myMeasChans, 2, results);
```
TevAXRF_GetPowerAtFreqOffset

Description: Get the power at the closest frequency bin specified by:
\[ F = \text{measureFrequency} + \text{freqOffset} \]

Note: this API is used after an IQ measurement has been made.

Syntax: TevAXRF_GetPowerAtFreqOffset(
    double freqOffset,
    double *power
);

Parameters:
- freqOffset: Frequency offset from the measureFrequency previously set up.
- *power: Variable to hold the measurement results.

Usage:

double result, offset;

    // Setup the measurement
    TevAXRF_MeasureSetupIQ(AXRF_CH2, -10.0 dBm, 3.4 GHz, 
                             200 MHz, 100 KHz, 10 MHz);

    // Make the IQ measurement
    TevAXRF_MeasureIQ(AXRF_CH2, &result);

    // Get power at 3.5 GHz
    TevAXRF_GetPowerAtFreqOffset(100 Mhz, &offset);
TevAXRF_GetPowerAtFreqOffsetMultiChannel

Description: This is the multi-channel version of TevAXRF_GetPowerAtFreqOffset().

Get the power at the closest frequency bin specified by:
F = measureFrequency + freqOffset
for the number of channels in the previously specified channel list.

Note: this API is used after an IQ measurement has been made.

Syntax: TevAXRF_GetPowerAtFreqOffsetMultiChannel(
    int numChannels,
    double freqOffset,
    double *power
);

Parameters: numChannels the number of channels in the channel array:
1 to 4
freqOffset Frequency offset from the measureFrequency previously set up.
*power Variable to hold the measurement results.

Usage:
AXRF_CHANNEL myMeasChans[2];
double results[2], offsets[2];

    // Setup
    TevAXRF_MeasureIQSetupMultiChannel(myMeasChans, 2,
        -10.0 dBm, 3.4 GHz, 200 MHz, 100 KHz, 10 Mhz);

    //Measure
    TevAXRF_MeasureIQMultiChannel(myMeasChans, 2, results);

    //Get power at 3.5 GHz
    TevAXRF_GetPowerAtFreqOffsetMultiChannel(2, 100 Mhz, offsets);
**TevAXRF_GetNumberOfIQTraceDataPoints**

**Description:** Use of this API assumes previous execution of `TevAXRF_MeasureSetupIQ()` and `TevAXRF_MeasureIQ()`. As part of this sequence an FFT of a certain size and having a certain frequency resolution was calculated.

As part of that measurement setup, a `measurementSpan` of frequencies was also specified. This API returns the number of data points (FFT bins) corresponding to that frequency range. This value is used to allocate a results array for use by `TevAXRF_GetIQTraceData()`.

*Note:* this API returns the *number* of data points, not the actual data.

**Syntax:**

```c
Int TevAXRF_GetNumberOfIQTraceDataPoints(void)
```

**Parameters:**
- **Return variable**
  - This API returns the number of data points corresponding to a frequency range.

**Usage:**

```c
double result;
int numPoints;

// Setup the measurement
TevAXRF_MeasureSetupIQ(AXRF_CH2, -10.0 dBm, 3.4 GHz, 200 MHz, 100 KHz, 10 Mhz);

// Make the IQ measurement
TevAXRF_MeasureIQ(AXRF_CH2, &result);

// Get the number of data points corresponding to 10MHz around 3.4GHz
numPoints= TevAXRF_GetNumberOfIQTraceDataPoints();
```
TevAXRF_GetNumberOfIQTraceDataPointsChannelIndex

Description: This is the multi-channel version of TevAXRF_GetNumberOfIQTraceDataPoints(). Use of this API assumes previous execution of TevAXRF_MeasureSetupIQMultiChannel() and TevAXRF_MeasureIQMultiChannel(). As part of this sequence an FFT of a certain size and having a certain frequency resolution was calculated.

As part of that measurement setup, a measurementSpan of frequencies was also specified. This API returns the number of data points (FFT bins) corresponding to that frequency range. This value is used to allocate a results array for use by TevAXRF_GetIQTraceDataChannelIndex().

Note: this API returns the number of data points, not the actual data.

Syntax: Int TevAXRF_GetNumberOfIQTraceDataPointsChannelIndex(int channelIndex)

Parameters: channelIndex The index value identifying one IQ data set from a multi-channel data set:
0 to 3.

Return Variable This API returns the number of data points corresponding to a frequency range.

Usage: AXRF_CHANNEL myMeasChans[2];
double results[2];
int int numPoints;

// Setup
TevAXRF_MeasureIQSetupMultiChannel(myMeasChans, 2, -10.0 dBm, 3.4 GHz, 200 MHz, 100 KHz, 10 Mhz);

// Measure
TevAXRF_MeasureIQMultiChannel(myMeasChans, 2, results);

// Get the number of data points corresponding to 10MHz // around 3.4GHz for the second channel's data set numPoints= TevAXRF_GetNumberOfIQTraceDataPointsChannelIndex(1);
TevAXRF_GetIQTraceData

Description: Use of this API assumes previous execution of TevAXRF_MeasureSetupIQ() and TevAXRF_MeasureIQ(). As part of this sequence an FFT of a certain size and having a certain frequency resolution was calculated. As part of that measurement setup, a measurementSpan of frequencies was also specified.

This API returns the data (FFT bins) corresponding to that frequency range.

The number of data points returned is determined by TevAXRF_GetNumberOfIQTraceDataPoints().

Result arrays must be sized appropriately.

Note: there is no bounds checking for the results array. You are responsible for ensuring that arrays are large enough to hold all returned data.

Syntax:

TevAXRF_GetIQTraceData(
    double *xData,
    double *yData);

Parameters:

*xData Array to hold the list of discrete frequencies being returned.

*yData Array to hold the calculated power at each of the corresponding frequencies.

Usage:

double result;
int numPoints;
double *freqData= 0;       // reserve array pointer - freq data
double *powerData= 0;      // reserve array pointer - power data

// Setup the measurement
TevAXRF_MeasureSetupIQ(AXRF_CH2, -10.0 dBm, 3.4 GHz, 200 MHz, 100 KHz, 10 MHz);

// Make the IQ measurement
TevAXRF_MeasureIQ(AXRF_CH2, &result);

// Get the number of data points corresponding to 10MHz around 3.4GHz
numPoints= TevAXRF_GetNumberOfIQTraceDataPoints();

// Allocate data array space
freqData= new (double[numPoints]);
powerData= new (double[numPoints]);

// Get trace data
TevAXRF_GetIQTraceData(freqData, powerData);
**TevAXRF_GetIQTraceDataChannelIndex**

**Description:** This is the multi-channel version of TevAXRF_GetIQTraceData().

Use of this API assumes previous execution of TevAXRF_MeasureSetupIQMultiChannel() and TevAXRF_MeasureIQMultiChannel(). As part of this sequence an FFT of a certain size and having a certain frequency resolution was calculated. As part of that measurement setup, a measurementSpan of frequencies was also specified.

This API returns the data (FFT bins) corresponding to that frequency range. Since only one set of data is returned, an additional parameter (channelIndex) is needed to specify which of the multi-channel data sets is returned.

The number of data points returned is determined by TevAXRF_GetNumberOfIQTraceDataPointsChannelIndex(). Result arrays must be sized appropriately.

**Note:** there is no bounds checking for the results array. You are responsible for ensuring that arrays are large enough to hold all returned data.

**Syntax:**

```c
TevAXRF_GetIQTraceDataChannelIndex(
    int channelIndex,
    double *xData,
    double *yData
);
```

**Parameters:**

- `channelIndex`  The index value identifying one IQ data set from a multi-channel data set:
  0 to 3.
- `*xData` Array to hold the list of discrete frequencies being returned.
- `*yData` Array to hold the calculated power at each of the corresponding frequencies.

**Usage:**

```c
AXRF_CHANNEL myMeasChans[2];
double results[2];
int numPoints;
double *freqData= 0; // reserve array pointer – freq data
double *powerData= 0; // reserve array pointer – power data

// Setup the measurement
TevAXRF_MeasureIQSetupMultiChannel(myMeasChans, 2,
    -10.0 dBm, 3.4 GHz, 200 MHz, 100 KHz, 10 Mhz);

// Measure
TevAXRF_MeasureIQMultiChannel(myMeasChans, 2, results);

// Get the number of data points corresponding to 10MHz
// around 3.4GHz for the second channel's data set
numPoints= TevAXRF_GetNumberOfIQTraceDataPointsChannelIndex(1);

// Allocate data array space
freqData= new (double[numPoints]);
powerData= new (double[numPoints]);

// Get trace data for second data set
TevAXRF_GetIQTraceDataChannelIndex(1, freqData, powerData);
```
TevAXRF_MeasureArray

Description: Perform a standard AXRF_SCALAR measurement of the RF signal on the specified AXRF_CHANNEL and return the dBm signal level of the measured signal.

Digitize the RF signal on the specified AXRF_CHANNEL.

Return the result of a standard scalar measurement on the digitized data.

Return either the time domain or the frequency domain data in the dataArray.

The default number of data points returned is:
- AXRF_TIME_DOMAIN 2048
- AXRF_FREQ_DOMAIN 1024

These values can be changed using TevAXRF_SetMeasureSamples() API.

Syntax:
```c
double TevAXRF_MeasureArray(
    AXRF_CHANNEL channel,
    double dataArray[],
    AXRF_ARRAY_TYPE type
);
```

Parameters:
- **channel** An AXRF Channel:
  - AXRF_CH1, AXRF_CH2, AXRF_CH3, AXRF_CH4,
  - AXRF_CH5, AXRF_CH6, AXRF_CH7, AXRF_CH8.
- **dataArray** The array to return the result(s).
- **type** The type of measurement data to be returned:
  - AXRF_TIME_DOMAIN time domain data
  - AXRF_FREQ_DOMAIN frequency domain data.

Usage:
```c
double result, digArray[1024];

// Setup the measurement
TevAXRF_MeasureSetup(AXRF_CH2, -10.0 dBm, 3.4 GHz);

// Make the measurement, return freq domain data
result = TevAXRF_MeasureArray(AXRF_CH2, &digArray, AXRF_FREQ_DOMAIN);
```
**TevAXRF_MeasureArrayMultiChannel**

**Description:** Perform a standard AXRF_SCALAR measurement of the RF signal on the specified AXRF_CHANNELS. Return either the time domain or the frequency domain data in the dataArray.

Unlike the single channel version, the scalar measurement value is not returned — only the array of data.

The default number of data points returned is:

- **AXRF_TIME_DOMAIN**: 2048
- **AXRF_FREQ_DOMAIN**: 1024

These values can be changed using **TevAXRF_SetMeasureSamples()**.

The size of the dataArray must be sufficient to hold (# samples returned) x numChannels measured. For example: for the default case, measuring on four AXRF channels and returning frequency domain data requires that dataArray is at least 4096 elements. Data is stored in measure channel order — channel 1 data is stored in elements 0–1023, channel 2 data is stored in elements 1024–2047, etc.

**Syntax:**

```c
TevAXRF_MeasureArrayMultiChannel(
    AXRF_CHANNEL channelArray[],
    int numChannels,
    double dataArray[],
    AXRF_ARRAY_TYPE type
);
```

**Parameters:**

- **channelArray** An array of AXRF channels to measure:
  - AXRF_CH1, AXRF_CH2, AXRF_CH3, AXRF_CH4,
  - AXRF_CH5, AXRF_CH6, AXRF_CH7, AXRF_CH8.

- **numChannels** The number of channels in the channel array
  1 to 4

- **dataArray** The array to return the result(s).

- **type** The type of measurement data to be returned:
  - **AXRF_TIME_DOMAIN**: time domain data
  - **AXRF_FREQ_DOMAIN**: frequency domain data.

**Usage:**

```c
AXRF_CHANNEL myMeasChans[2];
double measData[2048];  // array to hold returned data
TevAXRF_MeasureArrayMultiChannel(myMeasChans, 2, measData, AXRF_FREQ_DOMAIN);
```
**TevAXRF_MeasureArrayIQ**

Description: Digitize the RF signal on the specified AXRF_CHANNEL and return the raw IQ data in the `iDataArray` and `qDataArray`. This raw IQ Data can then be de-modulated and analyzed using the Cobham measurement libraries.

*Note:* the default sampling frequency for IQ data is 250 MHz. The sampling frequency for an IQ measurement can be adjusted using `TevAXRF_SetIQSampleFrequency()`.

**Syntax:**

```c
TevAXRF_MeasureArrayIQ(
    AXRF_CHANNEL channel,
    int numberOfPoints,
    float iDataArray[],
    float qDataArray[]
);
```

**Parameters:**

- `channel` An AXRF channel. One of the following:
  - `AXRF_CH1`
  - `AXRF_CH2`
  - `AXRF_CH3`
  - `AXRF_CH4`
  - `AXRF_CH5`
  - `AXRF_CH6`
  - `AXRF_CH7`
  - `AXRF_CH8`.

- `numberOfPoints` The number of points to capture.

- `iDataArray` I data results. Legal float pointer.

- `qDataArray` Q data results. Legal float pointer.

**Usage:**

```c
// Result arrays
float Ires[1024], Qres[1024];

// Setup measurement
TevAXRF_MeasureSetup(AXRF_CH2, -10.0 dBm, 3.4 GHz);

// Make measurement
TevAXRF_MeasureArrayIQ(AXRF_CH2, 1024, Ires, Qres);
```
TevAXRF_MeasureArrayIQwithKey

Description: Same as TevAXRF_MeasureArrayIQ() but also returns the Cobham analysis library tag and key. The tag and key must be used to enable the use of the individual analysis functions within the Cobham analysis library.

Note: the default sampling frequency for IQ data is 250 MHz. The sampling frequency for an IQ measurement can be adjusted using TevAXRF_SetIQSampleFrequency().

Syntax:
```
TevAXRF_MeasureArrayIQwithKey(
    AXRF_CHANNEL channel,
    int numberOfPoints,
    float iDataArray[],
    float qDataArray[],
    double *tag,
    double *key
);
```

Parameters:
- **channel**
  - An AXRF channel:
    - AXRF_CH1, AXRF_CH2, AXRF_CH3, AXRF_CH4,
    - AXRF_CH5, AXRF_CH6, AXRF_CH7, AXRF_CH8.

- **numberOfPoints**
  - The number of points to capture.

- **iDataArray**
  - I data results.
  - Legal float pointer.

- **qDataArray**
  - Q data results.
  - Legal float pointer.

- **tag**
  - analysis library authentication tag (returned).

- **key**
  - analysis library authentication key (returned).

Usage:
```
float *iWlan, *qWlan; // Result array pointers
int sampleSize;
double LevelCalp;
double tag1, key1;

// Dynamically allocate result array size based on sampleSize
iWlan= new(float[sampleSize]);
qWlan= new(float[sampleSize]);

TevAXRF_MeasureSetup(AXRF_CH2, -10.0 dBm, 3.4 GHz); // Setup measurement

// Make measurement
TevAXRF_MeasureArrayIQwithKey(
    AXRF_CH2, sampleSize, iWlan, qWlan, &tag1, &key1);

// Set center frequency in analysis library for primary segment
afWlanDll_Spectrum_FreqAxis_Centre_Set(nlD, RfinFreq1);
TevAXRF_GetMeasureFactor(AXRF_CH2, &LevelCalp);
afWlanDll_RfLevelCal_Set(nlD, LevelCalp-3);

// Compute spectrum for primary segment – Aeroflex analysis library
afWanDll_AnalyseIQ(
    nlD, afWlanMeasSpectrum, iWlan, qWlan, sampleSize, 0,
    sampleSize, tag1, key1);
```
**TevAXRF_MeasureReturnLoss**

**Description:** Perform a return loss measurement at the frequency and amplitude specified by the return loss setup function on the port selected. The format parameter determines what format is used to report the measurement results.

<table>
<thead>
<tr>
<th>Format</th>
<th>Results. Real</th>
<th>Results. Imaginary</th>
</tr>
</thead>
<tbody>
<tr>
<td>AXRF_VSWR</td>
<td>(1+magnitude)/(1−magnitude)</td>
<td>N/A</td>
</tr>
<tr>
<td>AXRF_RLDB</td>
<td>−20 * log10(magnitude)</td>
<td>Phase (degrees)</td>
</tr>
<tr>
<td>AXRF_REFL</td>
<td>Real component</td>
<td>Imaginary component</td>
</tr>
</tbody>
</table>

*Note:* digitizer default sample size is 2048. Sampling frequency is fixed at 250 MHz.

**Syntax:**
```
TevAXRF_MeasureReturnLoss(
    AXRF_CHANNEL channel,
    AXRF_RL_FORMAT format,
    NIComplexNumber *results
);
```

**Parameters:**
- **channel** An AXRF channel. One of the following: AXRF_CH1, AXRF_CH2, AXRF_CH3, AXRF_CH4, AXRF_CH5, AXRF_CH6, AXRF_CH7, AXRF_CH8.
- **format** The data format of the results. One of the following: AXRF_VSWR, AXRF_RLDB, AXRF_REFL.
- **results** The returned results as a complex pair.

**Usage:**
```
NIComplexNumber returnLoss;

// Setup
TevAXRF_MeasureReturnLossSetup(AXRF_CH1, 0.5dBm, 3.4GHz);

// Measure
TevAXRF_MeasureReturnLoss(AXRF_CH1, AXRF_VSWR, &returnLoss);
```
**TevAXRF_MeasureReturnLossMultiChannel**

**Description:** Perform a return loss measurement at the frequency and amplitude specified by the return loss setup function on the specified ports.

Only one channel from each channel pair (corresponding to a DRPM) can be specified in the `channelArray`. This is checked by the software driver.

The `format` parameter determines what format is used to report the measurement results.

<table>
<thead>
<tr>
<th>Format</th>
<th>Results. Real</th>
<th>Results. Imaginary</th>
</tr>
</thead>
<tbody>
<tr>
<td>AXRF_VSWR</td>
<td>(1+magnitude)/(1−magnitude)</td>
<td>N/A</td>
</tr>
<tr>
<td>AXRF_RLDB</td>
<td>−20 * log10(magnitude)</td>
<td>Phase (degrees)</td>
</tr>
<tr>
<td>AXRF_REFL</td>
<td>Real component</td>
<td>Imaginary component</td>
</tr>
</tbody>
</table>

Measurements are performed on the channels sequentially, in the order specified by `channelArray`. Results are stored in the `rlResults` array in the same order.

**Note:** digitizer default sample size is 2048. Sampling frequency is fixed at 250 MHz.

**Syntax:**

```c
TevAXRF_MeasureReturnLossMultiChannel(
    AXRF_CHANNEL channelArray[],
    int numChannels,
    AXRF_RL_FORMAT format,
    NIComplexNumber *rlResults
);
```

**Parameters:**
- `channelArray` An array of AXRF channels to measure:
  - AXRF_CH1, AXRF_CH2, AXRF_CH3, AXRF_CH4, AXRF_CH5, AXRF_CH6, AXRF_CH7, AXRF_CH8.
- `numChannels` The number of channels in the channel array:
  - 1 to 4.
- `format` The data format of the results. One of the following:
  - AXRF_VSWR
  - AXRF_RLDB
  - AXRF_REFL.
- `rlResults` The returned results as an array of complex pairs.

**Usage:**

```c
AXRF_CHANNEL myMeasChans[2];
NIComplexNumber returnLoss[2];

// Setup and Measure
TevAXRF_MeasureReturnLossSetupMultiChannel(myMeasChans, 2, 0.5dBm, 3.4GHz);
TevAXRF_MeasureReturnLossMultiChannel(myMeasChans, 2, AXRF_VSWR, returnLoss);
```
TevAXRF_MeasureReturnLossArray

Description: The TevAXRF_MeasureReturnLossArray() routine is the same as the TevAXRF_MeasureReturnLoss() API but also returns the raw measurement data.

The `rlArray` contains the results of two digitizer measurements:
- `array[0]` to `array[2047]` forward signal digitizer samples
- `array[2048]` to `array[4095]` reflected signal digitizer samples.

**Note:** digitizer default sample size is 2048.
Sampling frequency is fixed at 250 MHz.

Syntax: TevAXRF_MeasureReturnLossArray(
    AXRF_CHANNEL channel,
    AXRF_RL_FORMAT format,
    NIComplexNumber *rlResults
    double rlArray[]
);

Parameters:
- **channel** An AXRF channel. One of the following:
  - AXRF_CH1, AXRF_CH2, AXRF_CH3, AXRF_CH4,
  - AXRF_CH5, AXRF_CH6, AXRF_CH7, AXRF_CH8.
- **format** The data format of the results. One of the following:
  - AXRF_VSWR
  - AXRF_RLDB
  - AXRF_REFL.
- **rlResults** NIComplexNumber
- **rlArray** Data array to return the raw data in.
  Should be 2048*2 points.

Usage:
```
    // Measurement result
    NIComplexNumber returnLoss;

    // Raw digitizer data
    double rawData[2048*2];

    // Setup and measure
    TevAXRF_MeasureReturnLossSetup(AXRF_CH1, 0.5dBm, 3.4GHz);
    TevAXRF_MeasureReturnLossArray(AXRF_CH1, AXRF_VSWR, &returnLoss, rawData);
```
**TevAXRF_MeasureReturnLossArrayMultiChannel**

**Description:** The same as the TevAXRF_MeasureReturnLossMultiChannel() API but also returns the raw measurement data for all specified channels.

Only one channel from each channel pair (corresponding to a DRPM) can be specified in the `channelArray`. This is checked by the software driver.

Measurements are performed on the channels sequentially, in the order specified by `channelArray`. Results are stored in the `rlResults` array in the same order. `rlArray` contains the results of two digitizer measurements for each channel measured, in the order specified by `channelArray`:

<table>
<thead>
<tr>
<th>rlArray element range</th>
<th>Signal direction</th>
<th>Channel</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>2047</td>
<td>Forward</td>
</tr>
<tr>
<td>2048</td>
<td>4095</td>
<td>Reflected</td>
</tr>
<tr>
<td>4096</td>
<td>6143</td>
<td>Forward</td>
</tr>
<tr>
<td>6144</td>
<td>8191</td>
<td>Reflected</td>
</tr>
</tbody>
</table>

**Note:** digitizer default sample size is 2048. Sampling frequency is fixed at 250 MHz.

**Syntax:**

```c
TevAXRF_MeasureReturnLossArrayMultiChannel(
    AXRF_CHANNEL channelArray[],
    int numChannels,
    AXRF_RL_FORMAT format,
    NIComplexNumber *rlResults
    double rlArray[]);
```

**Parameters:**

- **channelArray** An array of AXRF channels to measure:
  - AXRF_CH1, AXRF_CH2, AXRF_CH3, AXRF_CH4, AXRF_CH5, AXRF_CH6, AXRF_CH7, AXRF_CH8.

- **numChannels** The number of channels in the channel array:
  - 1 to 4.

- **format** The data format of the results. One of the following:
  - AXRF_VSWR
  - AXRF_RLDB
  - AXRF_REFL.

- **rlresults** The returned results as an array of complex pairs.

- **rlArray** Data array to return the raw data in.

  Must be large enough to hold all digitizer data.
  - (2048*2) points * (number of channels measured).

**Usage:**

```c
AXRF_CHANNEL myMeasChans[2];
NIComplexNumber returnLoss[2];
double rawData[2048*2*2];

// Setup and measure
TevAXRF_MeasureReturnLossSetupMultiChannel(myMeasChans, 2, 0.5dBm, 3.4GHz);
TevAXRF_MeasureReturnLossArrayMultiChannel(
    myMeasChans, 2, AXRF_VSWR, returnLoss, rawData);
```
**TevAXRF_MeasureSparameters**

**Description:** Perform an S-parameter measurement using the ports defined by TevAXRF_MeasureSparametersSetupMultiChannel(). The results are returned in a four-element array of complex numbers as follows, dependent on the format:

<table>
<thead>
<tr>
<th>S-param</th>
<th>Array element</th>
<th>AXRF_RECT</th>
<th>AXRF_LOG_POLAR</th>
<th>AXRF_POLAR</th>
</tr>
</thead>
<tbody>
<tr>
<td>S11</td>
<td>results [0].real</td>
<td>Real</td>
<td>Log(Magnitude)</td>
<td>Magnitude</td>
</tr>
<tr>
<td>S11</td>
<td>results [0].imaginary</td>
<td>Imaginary</td>
<td>Phase</td>
<td></td>
</tr>
<tr>
<td>S21</td>
<td>results [1].real</td>
<td>Real</td>
<td>Log(Magnitude)</td>
<td>Magnitude</td>
</tr>
<tr>
<td>S21</td>
<td>results [1].imaginary</td>
<td>Imaginary</td>
<td>Phase</td>
<td></td>
</tr>
<tr>
<td>S12</td>
<td>results [2].real</td>
<td>Real</td>
<td>Log(Magnitude)</td>
<td>Magnitude</td>
</tr>
<tr>
<td>S12</td>
<td>results [2].imaginary</td>
<td>Imaginary</td>
<td>Phase</td>
<td></td>
</tr>
<tr>
<td>S22</td>
<td>results [3].real</td>
<td>Real</td>
<td>Log(Magnitude)</td>
<td>Magnitude</td>
</tr>
<tr>
<td>S22</td>
<td>results [3].imaginary</td>
<td>Imaginary</td>
<td>Phase</td>
<td></td>
</tr>
</tbody>
</table>

**Note:** Digitizer default sample size is 2048. Sampling frequency is fixed at 250 MHz.

**Syntax:**
```
TevAXRF_MeasureSparameters(
    AXRF_CHANNEL channel,
    AXRF_SPARAM_FORMAT format,
    NIComplexNumber results[]
);
```

**Parameters:**
- **channel** An AXRF channel:
  - AXRF_CHANNEL channel,
  - AXRF_CH1, AXRF_CH2, AXRF_CH3, AXRF_CH4,
  - AXRF_CH5, AXRF_CH6, AXRF_CH7, AXRF_CH8
- **format** The data format of the results. One of the following:
  - AXRF_POLAR
  - AXRF_LOG_POLAR
  - AXRF_RECT
- **results** Results array.
  - Array must be NIComplexNumber, size 4 elements.

**Usage:**
```
NIComplexNumber sPparams[4];
// Setup
TevAXRF_MeasureSparametersSetup(AXRF_CH1, -10dBm, 0dBm, 3.4GHz);
// Measure
TevAXRF_MeasureSparameters(AXRF_CH1, AXRF_RECT, sParams);
```
**TevAXRF_MeasureSparametersMultiChannel**

Description: Perform an S-parameter measurement using the ports defined by `TevAXRF_MeasureSparametersSetup()MultiChannel()`. The results are returned in an array of complex numbers, four elements per measured channel as follows, dependent on the format:

<table>
<thead>
<tr>
<th>S-param</th>
<th>Array element</th>
<th>AXRF_RECT</th>
<th>AXRF_LOG_POLAR</th>
<th>AXRF_POLAR</th>
</tr>
</thead>
<tbody>
<tr>
<td>S11</td>
<td>results [0].real</td>
<td>Real</td>
<td>Log(Magnitude)</td>
<td>Magnitude</td>
</tr>
<tr>
<td>S11</td>
<td>results [0].imaginary</td>
<td>Imaginary</td>
<td>Phase</td>
<td>Phase</td>
</tr>
<tr>
<td>S21</td>
<td>results [1].real</td>
<td>Real</td>
<td>Log(Magnitude)</td>
<td>Magnitude</td>
</tr>
<tr>
<td>S21</td>
<td>results [1].imaginary</td>
<td>Imaginary</td>
<td>Phase</td>
<td>Phase</td>
</tr>
<tr>
<td>S12</td>
<td>results [2].real</td>
<td>Real</td>
<td>Log(Magnitude)</td>
<td>Magnitude</td>
</tr>
<tr>
<td>S12</td>
<td>results [2].imaginary</td>
<td>Imaginary</td>
<td>Phase</td>
<td>Phase</td>
</tr>
<tr>
<td>S22</td>
<td>results [3].real</td>
<td>Real</td>
<td>Log(Magnitude)</td>
<td>Magnitude</td>
</tr>
<tr>
<td>S22</td>
<td>results [3].imaginary</td>
<td>Imaginary</td>
<td>Phase</td>
<td>Phase</td>
</tr>
</tbody>
</table>

*Note:* digitizer default sample size is **2048**. Sampling frequency is fixed at **250 MHz**.

**Syntax:**
```c
TevAXRF_MeasureSparametersMultiChannel(
    AXRF_CHANNEL channelArray[],
    int numChannels,
    AXRF_SPARAM_FORMAT format,
    NIComplexNumber spResults[]
);
```

**Parameters:**
- `channelArray`: An array of AXRF channels to measure:
  - AXRF_CH1, AXRF_CH2, AXRF_CH3, AXRF_CH4, AXRF_CH5, AXRF_CH6, AXRF_CH7, AXRF_CH8.
- `numChannels`: The number of channels in the channel array:
  - 1 to 4.
- `format`: The data format of the results. One of the following:
  - AXRF_POLAR
  - AXRF_LOG_POLAR
  - AXRF_RECT.
- `spResults`: Results array.
  - Array must be type `NIComplexNumber`.
  - Array size = (4 elements) X (number of measured channels).

**Usage:**
```c
AXRF_CHANNEL myMeasChans[2];
NIComplexNumber sParams[4*2];

// Setup
TevAXRF_MeasureSparametersSetupMultiChannel(
    myMeasChans, 2, -10dBm, 0dBm, 3.4GHz);

// Measure
TevAXRF_MeasureSparametersMultiChannel(myMeasChans, 2, AXRF_RECT, sParams);
```
**TevAXRF_MeasureParametersArray**

**Description:** The TevAXRF_MeasureParametersArray() routine is the same as the TevAXRF_MeasureParameters() API but will also return the raw digitizer measurement data.

The **rawDataArray** must be sized to contain the results of six digitizer measurements:

<table>
<thead>
<tr>
<th>Port</th>
<th>Measurement</th>
<th>Port</th>
<th>Measurement</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>forward</td>
<td>2</td>
<td>reflected</td>
</tr>
<tr>
<td>3</td>
<td>through</td>
<td>4</td>
<td>forward</td>
</tr>
<tr>
<td>5</td>
<td>reflected</td>
<td>6</td>
<td>through</td>
</tr>
</tbody>
</table>

**Note:** digitizer default sample size is **2048**. Sampling frequency is fixed at **250 MHz**.

**Syntax:**
```c
TevAXRF_MeasureParametersArray(
    AXRF_CHANNEL channel,
    AXRF_SPARAM_FORMAT format,
    NIComplexNumber results[],
    double rawDataArray[]
);
```

**Parameters:**
- **channel** An AXRF channel:
  - AXRF_CH1, AXRF_CH2, AXRF_CH3, AXRF_CH4,
  - AXRF_CH5, AXRF_CH6, AXRF_CH7, AXRF_CH8.
- **format** The data format of the results. One of the following:
  - AXRF_POLAR
  - AXRF_LOG_POLAR
  - AXRF_RECT.
- **results** NIComplexNumber Array.
- **rawDataArray** Data array to return the raw data in. Should be **2048*6** points.

**Usage:**
```c
NIComplexNumber sParams[4];
double rawData[2048*6];

// Setup
TevAXRF_MeasureParametersSetup(AXRF_CH1, -10dBm, 0dBm, 3.4GHz);

// Measure
TevAXRF_MeasureParameters(AXRF_CH1, AXRF_RECT, &sParams, rawData);
```
TevAXRF_MeasureSparametersArrayMultiChannel

Description: The same as the TevAXRF_MeasureSparametersMultiChannel() API but also returns the raw digitizer measurement data for all measured channels. The rawDataArray must be sized to contain the results of six digitizer measurements for all measured channels:

<table>
<thead>
<tr>
<th>DRPM 1</th>
<th>Port 1</th>
<th>1. forward</th>
<th>2. reflected</th>
<th>3. through</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Port 2</td>
<td>4. forward</td>
<td>5. reflected</td>
<td>6. through</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DRPM 2</th>
<th>Port 1</th>
<th>1. forward</th>
<th>2. reflected</th>
<th>3. through</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Port 2</td>
<td>4. forward</td>
<td>5. reflected</td>
<td>6. through</td>
</tr>
</tbody>
</table>

Syntax: TevAXRF_MeasureSparametersArrayMultiChannel(
           AXRF_CHANNEL channelArray[],
           int numChannels,
           AXRF_SPARAM_FORMAT format,
           NIComplexNumber spResults[]
           double rawDataArray[]
       );

Parameters: channelArray An array of AXRF channels to measure:
             AXRF_CH1, AXRF_CH2, AXRF_CH3, AXRF_CH4,
             AXRF_CH5, AXRF_CH6, AXRF_CH7, AXRF_CH8.

numChannels The number of channels in the channel array:
              1 to 4.

format The data format of the results. One of the following:
         AXRF_POLAR
         AXRF_LOG_POLAR
         AXRF_RECT.

results NIComplexNumber Array.

rawDataArray Data array to return the raw data in.
               Should be 2048*6 points * (number of measured channels).

Usage:
AXRF_CHANNEL myMeasChans[2];
NIComplexNumber sParams[4*2];
double rawData[2048*6*2];

// Setup
TevAXRF_MeasureSparametersSetupMultiChannel(
       myMeasChans, 2, -10dBm, 0dBm, 3.4GHz);

// Measure
TevAXRF_MeasureSparametersArrayMultiChannel(
       myMeasChans, 2, AXRF_RECT, &sParams, &rawData);
TevAXRF_MeasureNoiseFigureYfactor

Description: A noise figure measurement should be set up using TevAXRF_MeasureNoiseFigureSetup() prior to calling this API. The number of samples digitized for each measurement is set with that API.

Determining the noise figure requires two separate measurements:
1. With the noise source turned ON, the signal at measureChannel is digitized.
2. The noise source is turned OFF, the signal at measureChannel is digitized.

Multiple measurements can be taken and averaged. The number of measurements is set by aveCount. The noise figure and noise gain are then calculated from these two measurements.

Syntax: TevAXRF_MeasureNoiseFigureYfactor(
    AXRF_CHANNEL sourceChannel,
    AXRF_CHANNEL measureChannel,
    int aveCount,
    double *noiseFigure,
    double *noiseGain
);

Parameters: sourceChannel  An AXRF channel. One of the following:
    AXRF_CH1, AXRF_CH2, AXRF_CH3, AXRF_CH4,
    AXRF_CH5, AXRF_CH6, AXRF_CH7, AXRF_CH8.

measureChannel  An AXRF channel. One of the following:
    AXRF_CH1, AXRF_CH2, AXRF_CH3, AXRF_CH4,
    AXRF_CH5, AXRF_CH6, AXRF_CH7, AXRF_CH8.

aveCount  Number of measurements to be taken and averaged:
    1 to …

noiseFigure  Variable to hold the noise figure result.

noiseGain  Variable to hold the noise gain result.

Usage: double noiseFigure, noiseGain;

TevAXRF_MeasureNoiseFigureSetup(AXRF_CH1, AXRF_CH2, 1.007e9, 32768);
TevAXRF_MeasureNoiseFigureYfactor(AXRF_CH1, AXRF_CH2, 1,
    &noiseFig, &noiseGain);
AXRF measurement syntax

**TevAXRF_MeasureNoisePower**

**Description:** Measure the RMS noise power by calculating the sum-of-squares of the captured data samples. Individual measurements can be averaged to provide a more repeatable result. Each ‘measurement’ captures a specified number of samples. Multiple measurements can be averaged by specifying an `aveCount` > 1.

**Syntax:**
```plaintext
Double TevAXRF_MeasureNoisePower(
    AXRF_CHANNEL channel,
    int samples,
    int aveCount
);
```

**Parameters:**
- **channel**: An AXRF channel. One of the following:
  - AXRF_CH1, AXRF_CH2, AXRF_CH3, AXRF_CH4,
  - AXRF_CH5, AXRF_CH6, AXRF_CH7, AXRF_CH8.
- **samples**: The number of samples to capture for each noise power measurement.
- **aveCount**: The number of measurements to average together for the final result.

**Usage:**
```plaintext
double measuredNoise;
measuredNoise = TevAXRF_MeasureNoisePower(AXRF_CH1, 1000, 8);
```

**TevAXRF_GetRawArray**

**Description:** Provides access to the raw data captured by the Digitizer as part of an IF measurement. The `numberOfPoints` are specified to be returned as integer values to the `dataArray`. This API is a wrapped version of `afDigitizerDll_Capture_IF_CaptMem()`.

**Syntax:**
```plaintext
TevAXRF_GetRawArray(
    AXRF_CHANNEL channel,
    short dataArray[],
    int numberOfPoints
);
```

**Parameters:**
- **channel**: An AXRF channel. One of the following:
  - AXRF_CH1, AXRF_CH2, AXRF_CH3, AXRF_CH4,
  - AXRF_CH5, AXRF_CH6, AXRF_CH7, AXRF_CH8.
- **dataArray**: the array to return the results
- **numberOfPoints**: number of sample points to return
  - 32 - 268434432

**Usage:**
```plaintext
short rawData[4096];
TevAXRF_GetRawArray(AXRF_CH1, rawData, 4096);
```
AXRF advanced syntax

There are a number of APIs that provide additional, lower-level access to the AXRF hardware and software. These advanced APIs fall into several categories:

- Get instrument ‘handles’ to allow direct programming of Cobham instruments
- Lower-level programming of the RFSM and DRPMs for noise measurements
- Set the RF Digitizer’s sample size and sample frequency
- Get internal calibration factors for the source and measurement paths.

TevAXRF_CheckLocked

Description: Check if the AXRF Signal Generator(s) and Digitizer are locked to the master 10 MHz clock.

Syntax: int TevAXRF_CheckLocked(
        int subSystem
    );

Parameters: subSystem 0-based chassis identifier
0  first (primary) PXI cage
1  second (optional) PXI cage

return locked status
0= All instruments LOCKED
1= Digitizer NOT LOCKED
2= Signal GeneratorA NOT LOCKED
4= Signal GeneratorB NOT LOCKED.

Usage: int lockCheck;

    lockCheck = TevAXRF_CheckLocked(0);
**TevAXRF_IsCalibrationValid**

**Description:** Read back the state of all AXRF calibrations. The returned error code value shows the status of the 16 conditions.

**Syntax:**

```c
TevAXRF_IsCalibrationValid(
    int subSystem,
    AXRF_CAL_MODE mode,
    UINT *errorCode
);
```

**Parameters:**

- `subSystem`: 0-based chassis identifier
  - 0: first (primary) PXI cage
  - 1: second (optional) PXI cage

- `mode`: Calibration mode to check
  - AXRF_CAL_SCALAR
  - AXRF_CAL_VECTOR
  - AXRF_CAL_NOISE

- `errorCode`: An AXRF calibration error code is returned to this variable.
  - 0: Valid Calibration

Each calibration error code is a unique bit in the error value, so all failing codes can be represented in a single value.

<table>
<thead>
<tr>
<th>Error code</th>
<th>Error meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x08000</td>
<td>Noise Calibration File invalid</td>
</tr>
<tr>
<td>0x04000</td>
<td>Vector Calibration File invalid</td>
</tr>
<tr>
<td>0x02000</td>
<td>Scalar Calibration File invalid</td>
</tr>
<tr>
<td>0x01000</td>
<td>Calibration out of date</td>
</tr>
<tr>
<td>0x00800</td>
<td>SigGen B LO in system not same as in calibration file</td>
</tr>
<tr>
<td>0x00400</td>
<td>SigGen B in system not same as in calibration file</td>
</tr>
<tr>
<td>0x00200</td>
<td>SigGen A LO in system not same as in calibration file</td>
</tr>
<tr>
<td>0x00100</td>
<td>SigGen A in system not same as in calibration file</td>
</tr>
<tr>
<td>0x00080</td>
<td>Digitizer LO in system not same as in calibration file</td>
</tr>
<tr>
<td>0x00040</td>
<td>Digitizer in system not same as in calibration file</td>
</tr>
<tr>
<td>0x00020</td>
<td>RFM in system not same as in calibration file</td>
</tr>
<tr>
<td>0x00010</td>
<td>RFSM in system not same as in calibration file</td>
</tr>
<tr>
<td>0x00008</td>
<td>DRPM #4 in system not same as in calibration file</td>
</tr>
<tr>
<td>0x00004</td>
<td>DRPM #3 in system not same as in calibration file</td>
</tr>
<tr>
<td>0x00002</td>
<td>DRPM #2 in system not same as in calibration file</td>
</tr>
<tr>
<td>0x00001</td>
<td>DRPM #1 in system not same as in calibration file</td>
</tr>
</tbody>
</table>

**Usage:**

```c
UNIT calErrorStatus;
TevAXRF_IsCalibrationValid(0, AXRF_CAL_SCALAR, &calErrorStatus);
```
TevAXRF_GetDigitizerHandle

Description: Return the handle to the Cobham RF digitizer in the AXRF RF Subsystem. This handle can then be used with the Cobham digitizer software drivers to provide direct control of the instrument’s full feature set.

Syntax:
TevAXRF_GetDigitizerHandle(
    int subSystem,
    afDigitizerInstance_t *DigitizerId
);

Parameters:
subSystem Subsystem number (0-based) used to identify an AXRF instance.
DigitizerId The software handle that uniquely identifies the Cobham digitizer.

Usage:
afDigitizerInstance_t digHandle;
TevAXRF_GetDigitizerHandle(0, &digHandle);

TevAXRF_GetSigGenHandle

Description: Return the handle to the Cobham signal generator in the AXRF RF Subsystem. This handle can then be used with the Cobham source software drivers to provide direct control of the instrument’s full feature set.

Syntax:
TevAXRF_GetSigGenHandle(
    int subSystem,
    afSigGenInstance_t *SigGenId
);

Parameters:
subSystem Subsystem number (0-based) used to identify an AXRF instance.
DigitizerId The software handle that uniquely identifies the Cobham digitizer.

Usage:
afSigGenInstance_t sigGenHandle;
TevAXRF_GetSigGenHandle(0, &sigGenHandle);
**TevAXRF_GetSigGenBHandle**

**Description:** Same as `TevAXRF_GetSigGenHandle()` but for the optional second Signal Generator.

Return the handle to the Cobham signal generator in the AXRF RF Subsystem. This handle can then be used with the Cobham source software drivers to provide direct control of the instrument’s full feature set.

**Syntax:**

```c
TevAXRF_GetSigGenBHandle(
    int subSystem,
    afSigGenInstance_t *SigGenId
);
```

**Parameters:**

- `subSystem` 0-based chassis identifier
- `DigitizerId` The software handle that uniquely identifies the Cobham digitizer.

**Usage:**

```c
afSigGenInstance_t sigGenBHandle;
TevAXRF_GetSigGenBHandle(0, &sigGenBHandle);
```

**TevAXRF_ConnectNoiseSource**

**Description:** Ensures that the entire path from the noise source (located on the RFSM) to the selected output port (located on a DRPM) is set up correctly to source a noise signal.

This setup includes:

- Select the noise source instead of RF Source A.
- Noise source gate is UNCHANGED.
- RFSM source path relay (one of four) is CLOSED. Other three relays are OPEN.
- RFSM source path amplifier (one of four) is ON. Other three amplifiers are OFF.
- RFSM source path attenuator (one of four) is programmed. Other three attenuators = 30 dB.
- DRPM relays for selected port are configured as an output.

*Note: only the states of DRPM relays on the specified port are set. DRPM relays for all other ports are UNCHANGED.*

**Syntax:**

```c
TevAXRF_ConnectNoiseSource(
    AXRF_CHANNEL channel,
    double attenuation
);
```

**Parameters:**

- `channel` An AXRF channel: `AXRF_CH1`, `AXRF_CH2`, `AXRF_CH3`, `AXRF_CH4`, `AXRF_CH5`, `AXRF_CH6`, `AXRF_CH7`, `AXRF_CH8`.
- `attenuation` Attenuation for the selected output path on the RFSM: 0 dB to 31 dB, 0.5 dB resolution.

**Usage:**

```c
TevAXRF_ConnectNoiseSource(AXRF_CH1, 30dB);
```
TevAXRF_DisconnectNoiseSource

Description: Ensures that the noise source (located on the RFSM) is disconnected from the output port (located on a DRPM).

This setup includes:

- Noise source is gated OFF.
- RFSM source path relay (one of four) is OPEN.
- RFSM source path amplifier (one of four) is OFF.
- RFSM source path attenuator is set to 30 dB.
- DRPM relays for selected port are set to ‘ISOLATE’ mode.

Syntax: `{tevaxrf DisconnectNoiseSource(AXRF_CHANNEL channel);}

Parameters: channel An AXRF channel:

AXRF_CH1, AXRF_CH2, AXRF_CH3, AXRF_CH4,
AXRF_CH5, AXRF_CH6, AXRF_CH7, AXRF_CH8.

Usage: TevAXRF_DisconnectNoiseSource(AXRF_CH1);

TevAXRF_SetNoiseSource

Description: Enable/disable the AXRF noise source, physically located on the RFSM (RF Splitter Module). This function operates independently of whether the noise source is connected to a sourcing path.

Syntax: `{tevaxrf SetNoiseSource(AXRF_CHANNEL channel, BOOL state);}

Parameters: channel An AXRF channel:

AXRF_CH1, AXRF_CH2, AXRF_CH3, AXRF_CH4,
AXRF_CH5, AXRF_CH6, AXRF_CH7, AXRF_CH8.

state State of the noise source:

FALSE — noise source is off
TRUE — noise source is on.

Usage: TevAXRF_SetNoiseSource(AXRF_CH1, TRUE);
**TevAXRF_SetMeasureSamples**

**Description:** Set the number of samples to be captured by the RF digitizer each time it is used as part of the following types of measurement APIs:

- Scalar
- Return Loss
- S-Parameter.

This value should be a power of 2. The default sample size is 2048.

**Syntax:**

```c
TevAXRF_SetMeasureSamples(unsigned int samples);
```

**Parameters:**
- `samples` Number of samples to be captured whenever the digitizer is used (power of 2).

**Usage:**

```
TevAXRF_SetMeasureSamples(4096);
```

**TevAXRF_SetIQSampleFrequency**

**Description:** Set the sample frequency to be used by the RF digitizer when making IQ measurements. The default sample frequency is 250 MHz.

**Syntax:**

```c
TevAXRF_SetIQSampleFrequency(double sampleFrequency);
```

**Parameters:**
- `sampleFrequency` RF digitizer sample frequency for IQ measurements.
  - Must be in the range 7630 kHz to 250 MHz.

**Usage:**

```
TevAXRF_SetIQSampleFrequency(225 MHz);
```

**TevAXRF_SetNoiseFigureSampleFrequency**

**Description:** Set the sample frequency to be used by the RF digitizer when making Noise Figure measurements. The default sample frequency is 8 MHz.

**Syntax:**

```c
TevAXRF_SetNoiseFigureSampleFrequency(double sampleFrequency);
```

**Parameters:**
- `sampleFrequency` RF digitizer sample frequency for Noise Figure measurements.
  - Practical range: 4 MHz to 16 MHz.

**Usage:**

```
TevAXRF_SetNoiseFigureSampleFrequency(5 MHz);
```
**TevAXRF_GetSourceFactor**

**Description:** Return the current RF calibration factor for the specified port. The cal factor for the frequency and level of the last TevAXRF_Source() routine executed is returned.

**Syntax:**
```c
TevAXRF_GetSourceFactor(
    AXRF_CHANNEL channel,
    double *sourceFactor
);
```

**Parameters:**
- `channel` An AXRF channel:
  - AXRF_CH1, AXRF_CH2, AXRF_CH3, AXRF_CH4,
  - AXRF_CH5, AXRF_CH6, AXRF_CH7, AXRF_CH8.
- `sourceFactor` The current source cal factor.

**Usage:**
```c
double srcFactor;
TevAXRF_GetSourceFactor(AXRF_CH1, &srcFactor);
```

**TevAXRF_GetMeasureFactor**

**Description:** Return the current RF calibration factor for the specified port. The cal factor for the frequency and level of the last TevAXRF_MeasureSetup() routine executed is returned. This can then be used to calibrate measurements.

**Syntax:**
```c
TevAXRF_GetMeasureFactor(
    AXRF_CHANNEL channel,
    double *measureFactor
);
```

**Parameters:**
- `channel` An AXRF channel:
  - AXRF_CH1, AXRF_CH2, AXRF_CH3, AXRF_CH4,
  - AXRF_CH5, AXRF_CH6, AXRF_CH7, AXRF_CH8.
- `measureFactor` The current source cal factor.

**Usage:**
```c
double measFactor;
TevAXRF_GetMeasureFactor(AXRF_CH2, &measFactor);
```
**TevAXRF_GetNoiseFactor**

**Description:** As part of the AXRF calibration process, a set of noise calibration factors is created for each signal path (that is, the noise source connected to each DRPM port). These calibration factors (ENR, \( P_{on} \), \( P_{off} \)) are dependent on the AXRF channel and the center frequency. This API reads the set of cal factors selected and returns them as three separate variables. The calibration factors can then be used to manually calibrate noise measurements.

*Note:* All noise calibration is performed with RFSM source attenuators programmed to 30 dB.

**Syntax:**
```
TevAXRF_GetNoiseFactor(
    AXRF_CHANNEL channel,
    double frequency,
    double *enrFactor,
    double *Pon,
    double *Poff
);
```

**Parameters:**
- `channel`: An AXRF channel: `AXRF_CH1`, `AXRF_CH2`, `AXRF_CH3`, `AXRF_CH4`, `AXRF_CH5`, `AXRF_CH6`, `AXRF_CH7`, `AXRF_CH8`.
- `frequency`: Center frequency associated with the calibration data of interest. Range 50 MHz to 6 GHz. The specified frequency should be one that was previously calibrated.
- `enrFactor`: Excess Noise Ratio (ENR). Pointer to a return value variable.
- `Pon`: Measured power with the Noise Source on. Pointer to a return value variable.
- `Poff`: Measured power with the Noise Source off. Pointer to a return value variable.

**Usage:**
```
double enrFactor, Pon, Poff;

TevAXRF_GetMeasureFactor(AXRF_CH1, 2.4GHz, &enrFactor, &Pon, &Poff);
```
TevAXRF_GetConfiguration

**Description:** Returns the AXRF Hardware Configuration data structure that contains all the information about the AXRF configuration. See data structure details below (and in TevAXRF.h):

```
AXRFHardware{
    int SubSystems;
    int SystemType;
    char ScalarCalDate[30];
    char ScalarCalTime[30];
    char VectorCalDate[30];
    char VectorCalTime[30];
    char NoiseCalDate[30];
    char NoiseCalTime[30];
    char ScalarAutoCalSerialNumber[30];
    char VectorAutoCalSerialNumber[30];
    char NoiseAutoCalSerialNumber[30];

    int DrpmSlots[4];
    int DrpmChassis[4];
    char DrpmPxiIDs[4][30];
    ViSession DrpmSessions[4];
    char DrpmSerialNumbers[4][30];

    int MuxTypes[2];
    int MuxSlots[2];
    int MuxChassis[2];
    char MuxPxiIDs[2][30];
    ViSession MuxSessions[2];
    char MuxSerialNumbers[4][30];

    int SourceGenTypes[2];
    int SourceGenSlots[2];
    int SourceGenChassis[2];
    char SourceGenPxiIDs[2][30];
    int SourceGenLoSlots[2];
    int SourceGenLoChassis[2];
    char SourceGenLoPxiIDs[2][30];
    ViSession SourceGenSessions[2]; // NI RF Source Generator
    afSigGenInstance_t SourceGenIds[2]; // Cobham RF Source Generator

    int SourceGenRefClock;
    int SourceGenExtClock[2];
    char SourceGenSerialNumbers[2][30];
    char SourceGenLoSerialNumbers[2][30];

    int DigitizerType;
    int DigitizerSlots[2];
    int DigitizerChassis[2];
    char DigitizerPxiIDs[2][30];
    ViSession DigitizerSessions[2]; // NI Digitizer

    int DigitizerLoSlot;
    int DigitizerLoChassis;
    char DigitizerLoPxiID[30];
    ViSession DigitizerLoSession; // NI RF Generator
    char DigitizerSerialNumbers[2][30];
    char DigitizerLoSerialNumber[30];
    afDigitizerInstance_t DigitizerId; // Cobham RF Digitizer
```
Syntax:

```c
int DigitizerRefClock;
int DigitizerExtClock;
}
```

Parameters:

- `subSystem` 0-based chassis identifier
  - 0 first (primary) PXI cage.
  - 1 second (optional) PXI cage.
- `config` a data structure to hold the requested configuration details.

Usage:

```c
AXRFHardware myConfig;

TevAXRF_GetConfiguration(0, &myConfig);
```
AXRF programming examples

RF source & measure example

This programming example sets the RF port AXRF_CH1 into the source mode, sourcing a 1 GHz signal at −10 dBm. The AXRF_CH5 port is set up to measure a 1.2 GHz signal and then the measurement is made and returns the dBm level of the signal into variable result.

```c
double result;
TevAXRF_Source(AXRF_CH1, -10.0, 1000e6);
TevAXRF_MeasureSetup(AXRF_CH5, 0.0, 1.2e9);
TevAXRF_Measure(AXRF_CH5, &result);
```

RF time and frequency domain examples

If the time domain or frequency domain data is required then the following code can be used to obtain the desired information.

```c
TevAXRF_MeasureSetup(AXRF_CH5, 0.0, 1.2e9);
TevAXRF_MeasureArray(AXRF_CH5, measureArray, AXRF_TIME_DOMAIN);
```

In this example the hardware is set up the same as in the previous example except that the RF measurement type is set to AXRF_TIME_DOMAIN. The time domain data of the RF signal is returned in measureArray. If frequency domain information is required, the RF measurement type should be AXRF_FREQ_DOMAIN.

```c
TevAXRF_MeasureSetup(AXRF_CH5, 0.0, 1.2e9, AXRF_FREQ_DOMAIN);
```
AXRF control software

AXRF control description

The AXRF control software is a centralized tool to configure, calibrate, and manually control the AXRF RF Subsystem:

- The Configuration tab is used to define the slot positions of modules in the AXRF RF Subsystem.
- The Control tab is used to control the AXRF RF Subsystem while it sources and measures RF signals.
- The Calibration tab is used to calibrate the AXRF RF Subsystem using an external calibration standard.
- The Checker tab is used to run and review subsystem diagnostics.

The functionality of the AXRF Control Panel is explained in detail in the following sections.

Installation

The Control Panel is installed during the driver installation process

Access the Control Panel from the Windows Start menu under:

All Programs\TEV\AXRF\AXRFControl.exe

Or open the AXRFControl.exe in:

Program Files\TEV\AXRF Subsystem.

The start-up splash screen is displayed:

![Start-up screen](image)

Figure 2  AXRF Start-up
Control panel

The AXRF Control Panel (Figure 3) provides a graphical user interface to the AXRF RF Subsystem. The underlying control is provided via the standard user-level API calls.

Figure 3  AXRF Control Panel
Configuration tab

The Configuration tab (Figure 4) is used to configure the AXRF RF Subsystem and initialize the hardware.

The Resources list displays all the available PXI resources in the system.

Each AXRF resource in the resource display must be selected and configured by clicking the button next to the appropriate AXRF resource location.

The AXRF system type must first be set to the correct system type:

- AXRF0 — no DRPMs
- AXRF2 — two-channel system (one DRPM, AXRF Option 16)
- AXRF4 — four-channel system (two DRPMs, AXRF Option 17)
- AXRF6 — reserved
- AXRF8 — eight-channel system (four DRPMs, AXRF Option 19)

![Configuration tab](image)

**Figure 4 Configuration tab**

**Source Generator**

Select the Source Generator resource from the Resources list and click the button to configure the Source Generator.

Select the Source LO resource from the Resources list and click the button to configure the Source LO.

If the Source Generator is going to supply the reference 10 MHz clock, click the Reference Clock box.
Digitizer

Select the Digitizer resource from the Resources list and click the button to configure the RF digitizer.

Select the Digitizer LO resource from the Resources list and click the button to configure the Digitizer Source LO.

If the digitizer is going to supply the reference 10 MHz clock, click the box.

Port Modules

The Dual Port Modules are configured in the order of RF channels: CH1 & 2 in the top location followed by CH3 & 4, CH5 & 6, and finally CH 7 & 8 in the bottom location.

Splitter and Mux Module

The Splitter and Mux Modules are configured here.
Control tab

The Control tab (Figure 5) is used to control the AXRF RF Subsystem. RF signals can be sourced and measured. Scalar (IF and IQ), Return Loss, S-Parameter and Noise measurements can be set up and executed.

![Control tab](image)

**Figure 5  Control tab**

**Mode**

The **Mode** drop-down list is used to select the high-level operating mode of the AXRF RF Subsystem.

![Mode](image)

**Source Channel**

The **Source Channel** drop-down list is used to select the RF Channel that is to be used to Source an RF signal. The **Level** and **Frequency** controls set the RF frequency and level to be generated. The **Source** button sources the RF signal.

![Source Channel](image)
Measure Channel

The Measure Channel drop-down list is used to select the RF channel that is to be used to measure an RF signal. The Level and Frequency controls set the RF frequency and level to be measured.

The Measure button measures the RF signal.

The Track Source box makes the Measure Level and Frequency track the Source Level and Frequency.

The Measure Meter button allows for an external power meter to be used to measure an RF signal.

Scalar measurements

The screen shot below shows a basic instrument setup and results of a scalar measurement using the standard IF measurement technique (Data switch= IF).

Note that this measurement uses Fsample= 250 MHZ and Sample Size= 2048.

Figure 6 Scalar measurement
The repeatability of any measurement setup can be analyzed using the Repeatability tab. A measurement loop can be specified — the number of measurements and the pre-measurement wait time. Results can be viewed as a histogram and via statistical parameters (for example, mean, std. deviation):

![Figure 7 Scalar measurement — repeatability](image)

**IQ measurements**

Scalar measurements can also be made using IQ data. This technique provides flexibility that is unavailable using just IF data. Setting `Data=IQ` enables an additional setup of measurement parameters.

These parameters control the Cobham digitizer via the lower-level Cobham function `afSpectrumDll()`. A critical part of the digitizer's setup determines the minimum FFT size needed to achieve the specified RBW.

\[
\text{MinimumFFTsize} = 2^\left(\frac{\log_2(\text{sampleFrequency} / \text{RBW} \times ENB \text{bins})}{\log_2(\text{sampleFrequency} / \text{RBW})}\right)
\]

where \( ENB\text{bins} = 2.5 \)

The digitizer can capture data in the range of 256 to 131072 points which corresponds to FFT sizes from 128 to 65536. In the following example, \( F_{\text{sample}} = 125 \text{ MHz} \), RBW= 10 KHz resulting in a minimum FFT size of 32K (data size = 65536).

![Figure 8 Scalar measurement using IQ data](image)
Increasing the RBW to 100 kHz reduces the required FFT size to 4096 (data size = 8192):

Figure 9  Scalar measurement using IQ data, increased RBW

Changing the Frequency Span from 10 MHz to 50 MHz does not change the required FFT size. It only changes the range (span) of frequencies (around the center frequency) that are displayed:

Figure 10  Scalar measurement using IQ data, changed frequency span
Decreasing RBW to 5 kHz increases the required FFT size. This reduces the effective power per frequency bin, making it easier to isolate low-level spurs:

![Figure 11 Scalar measurement using IQ data, decreased RBW](image1)

IQ mode measurements provide several benefits over the basic scalar measurement. The additional flexibility of being able to adjust the digitizer sampling frequency, effective resolution bandwidth (RBW) and measurement frequency span are very useful in making accurate, low-power CW measurements for harmonics, etc. However, smaller RBWs are generally achieved by using larger sample sizes, which in turn require longer FFT calculation time.

The AXRF Control Panel provides an interactive way to experiment with the IQ mode parameters and to design RF measurements to meet specific needs. Once the proper parameters values are determined, they can be transferred to the API calls in the test program.

**Return Loss measurements**

The Return Loss measurement results show the two digitizer signal data captures (direct, reflected):

![Figure 12 Return loss measurement](image2)
Return Loss measurement repeatability:

*Figure 13  Return loss measurement — repeatability*

**S-parameter measurements**

S-parameter measurement results show the six digitizer signal data captures that are required to calculate the S-parameter results:

*Figure 14  S-parameter measurement*
Repeatability of S-parameter measurements:

![Repeatability of S-parameter measurements](image1)

*Figure 15 S-parameter measurement — repeatability*

**Noise measurements**

Noise Gain and Noise Figure measurements:

![Noise Gain and Noise Figure measurements](image2)

*Figure 16 Noise measurement*
Noise Gain/Figure repeatability:

Figure 17 Noise measurement — repeatability
Calibration tab

The Calibration tab (Figure 18) is used to calibrate the AXRF RF Subsystem. Default setup assumes the use of an AXRF AutoCal Unit.

Figure 18 Calibration tab
Calibration tab

### File
- **Load Frequency List**
- **Save Frequency List As...**

### Frequencies
- **Add Frequencies to List**
- **Delete Selected Frequencies from List**
- **Save Calibration Frequencies As...**

---

**Load Frequency List**
Load a pre-existing list of frequencies (.frq).

**Save Frequency List As...**
Save a frequency list to a new file name (.frq).
Useful when keeping separate frequency calibration lists for each test program.

**Add Frequencies to List**
Create a list of calibration frequencies. Add new calibration frequencies to an existing list.

This item pops up a sub-window that allows the addition of a single frequency or range of frequencies:

**Delete Selected Frequencies from List**
Select one or more frequencies from the existing list. Select this item to remove those frequencies.

**Save Calibration Frequencies As...**
Save a frequency list to a new file name (.cal)

The .cal file type is used in conjunction with the command line execution of calibration routines.
The Channel List and Frequency List controls are used to select which RF channels are to be used, and at what frequencies the calibration is to be performed.

Frequencies can be added to or deleted from the list by using the options in the Frequencies menu.

Before launching any calibrations, the AutoCal Unit’s power meter must be initialized by clicking on the Initialize Power Meter button. The initialization checks for a valid standards data file directory and for the presence of the AutoCal Unit and its internal power meter via USB communication. If a calibration activity is attempted before this initialization, the initialization runs automatically first.

Scalar calibration is performed using the AutoCal Unit’s power meter and its Thru connections between DRPM port channel pairs. Vector Calibration is performed using the AutoCal Unit’s Short, Open, Load and Thru standards.

Scalar calibration is started by clicking the Scalar Calibration button. Source calibration is performed by the AutoCal Unit connecting its power meter to each RF channel. The source calibration factor is calculated and saved for each selected frequency. A source calibration factor for each frequency is plotted. Next, the measure calibration is performed as the AutoCal Unit connects each RF channel pair (1&2, 3&4, 5&6, 7&8) together using its internal Thru connection. Measure calibration factors are calculated for all frequencies and for each measure range. These calibration factors are plotted and saved to the calibration file: AXRFScalarCalFactors.cal in the AXRF \caldata directory.

Vector calibration is started by clicking the Vector Calibration button. The AutoCal Unit connects its internal Short, Open, Load and Thru networks as needed to perform all necessary S-parameter measurements. The 12-term error factors are calculated and saved to the calibration file: AXRFVectorErrorTerms.cal in the AXRF \caldata directory.

If the Measure Only option is selected, only the measure calibration is executed. This assumes that valid source calibration factors already exist.
If the Append Factors option is selected, the calibration factors are appended to the factors list. Normally, all calibration factors are cleared before a new calibration is performed.

The AutoCal Unit supports noise figure calibration. Clicking the Noise Figure Cal button performs the calibration and creates a calibration file: AXRFNoiseFactors.cal in the AXRF \caldata directory

Calibration can be verified by clicking these buttons. By default, channel pairs are connected together for source and measure verification. If the Use Power Meter option is selected, source calibration is verified with the AutoCal Unit power meter.

Verification is performed at all frequencies in the Frequency List. Verification is performed at the signal level specified in the Validate Level parameter box.

Cumulative calibration results are plotted here:

Cal Channel/Frequency/Result provides a real time update of the results of the calibration in progress.

Clear Output clears the plot window data.

The calibration test sequence is logged here:

A tabular listing of the calibration steps provides a detailed view of the calibration in progress.
Basic calibration procedure

1. Configure the PXI chassis **trigger buses** so that the DRPM trigger signals are routed to the RF digitizer. For the default AXRF configuration (see AXRF RF Subsystem configuration and cabling), Trigger Bus 2 signals must be routed to Trigger Bus 3.

2. Configure the trigger buses of the **Cobham 18-slot** chassis using the NI-MAX PXI setup tool as shown below:

   ![Figure 19 18-slot chassis trigger configuration](image)

   **Figure 19 18-slot chassis trigger configuration**

   This configures each trigger line as ‘Away from Bus 2’.

3. Connect each AXRF port (two per DRPM card) to the AutoCal Unit using an SMA-to-SMA cable.

   Refer to Figure 20 for the following steps:

4. Review/adjust the list of channels to be calibrated. Normally, all channels should be selected.

5. Review/adjust the list of frequencies to be calibrated. This frequency list is saved as part of the Control Panel user interface.

6. Run scalar calibration. Click the **Scalar Calibration** button.

7. Run vector calibration (optional — required when making S-parameter measurements). Click the **Vector Calibration** button.
8 Run noise figure calibration (optional — required when making noise measurements). Click the Noise Figure Cal button.

Figure 20 Calibration procedure
Checker tab

AXRF checker

The Checker tab (Figure 21) provides access to an AXRF RF Subsystem checker. This checker verifies proper cabling between the PXI cards and performs checks of the basic, overall functionality of the subsystem.

![Checker tab](image)

**Figure 21  Checker tab**

Hardware setup for Checker

Before running the AXRF Checker, disconnect any load boards or other circuitry connected to the external channel cables. Cables can remain connected.

The checker cannot differentiate a loose connector from a properly tightened connector. A loose connection creates an unreliable, intermittent signal path which can lead to poor performance.

Before running the checker, double-check that all cable connectors are properly seated and torqued.

Checker color code

A standard color code indicates a status for each of the interconnect cables in the subsystem:

- **BLUE** the cable is presently undiscovered.
  This is the default state for all cables before the checker is run.
- **GREEN** the cable is in place.
- **RED** the cable is missing or incorrectly connected.
The progress/results of an AXRF checker run are displayed in the two panes of the Checker tab. The right pane shows a subsystem diagram. The colors of the individual cables provide a visual status of the checker results:

![Checker results — no failures](image)

The left pane shows a log of subsystem information collected followed by a status of the test sequence performed by the checker:

- Inventory of the subsystem boards — board type, slot location, serial number:
  - DRPMs
  - RSM
  -RFM
  - RF source
  - RF digitizer
  - RF digitizer 10 MHz External Reference Lock
  - RF source output cable
  - RF digitizer input cable
  - DRPM cables
  - etc.
If a test fails, the related cables are shown in **RED** on the subsystem diagram. The failure is logged in the left pane, along with suggested cable and board checks:

![Checker results — failure](image)

**Figure 23 Checker results — failure**

**Cobham self-test**

Cobham provides a comprehensive self-test that can be run on the combined front- and back-end of the subsystem. Running the self-test software (available at [www.cobham.com/wireless](http://www.cobham.com/wireless)) is straightforward.

This entails connecting the RF source (3025C) directly to the RF digitizer (3035C). The self-test provides a means to help isolate problems within the subsystem.

- If the self-test fails, there is a problem with the RF source and/or the RF digitizer. Since the subsystem depends on these two main components, a problem here would show as a catastrophic AXRF RF Subsystem failure.
- If the self-test passes, but the AXRF checker still fails, the problem is probably due to one of the other modules, or the cables.

Changes of cabling must be made before the Cobham self-test is run. The changes create a direct connection from the RF source output to the RF digitizer input:

- Locate the cable that connects RF source RF OUT to RSM IN A. Disconnect the SMA cable from the RF source RF OUT connector.
- Locate the cable that connects RFM COM to RF digitizer RF IN. Disconnect the SMA cable from the RF digitizer RF IN connector.
- Source a separate SMA/SMA cable Connect the cable between RF source RF OUT and RF digitizer RF IN.
Launching/running the self-test software

The Cobham self-test application (Figure 24) is launched from:

Start / All Programs / Aeroflex / PXI SelfTest /

The initial screen shows the required cable configuration. Following the cable changes above should put the subsystem in the proper configuration.

Clicking NEXT starts the self-test process. The first step checks for the required instruments (RF source and RF digitizer) as well as the correct cabling.

**IMPORTANT**

Once the Cobham self-test has been run, re-running the AXRF checker and/or running the AXRF RF Subsystem requires that the cabling be returned to the standard AXRF configuration. This involves reversing the steps made previously:

- Remove the SMA/SMA cable between RF source RF OUT and RF digitizer RF IN.
- Locate the cable connected to RSM IN A.
  Reconnect the open end of this cable to the RF source RF OUT connector.
- Locate the cable connected to RFM COM.
  Reconnect the open end of this cable to the RF digitizer RF IN connector.

Re-run the AXRF checker to ensure proper cabling and subsystem operation.
AutoCal Unit

The AXRF ACAL-8 AutoCal Unit calibrates up to eight ports of the AXRF RF Subsystem without user intervention. It contains:

- a power meter for scalar calibration
- Short, Open, Load and Thru (SOLT) standards for vector calibration
- a noise source for noise figure calibration.

The AXRF AutoCal Unit is the default mechanism for performing subsystem calibration. It is controlled from the Control Panel.

There are two versions of the AXRF AutoCal Unit: a benchtop box in a case with carrying handle, and a loadboard version.

Benchtop box

The eight channels (ports) of the AXRF RF Subsystem are connected by cable to the eight SMA connectors of the AutoCal Unit.

**Note:** the calibration plane is the end of the cables connected to the AutoCal Unit.

**CAUTION**

DO NOT exceed +20 dBm input level. Higher levels will damage the internal power meter.

Power to the AutoCal Unit is provided by an external +48 VDC power adapter (included).

Connect the power adapter to the +48 VDC input connector on the front of the AutoCal Unit.
The benchtop AutoCal Unit requires two separate USB connections to the test computer:

One USB connection controls the AutoCal Unit’s circuitry...

...the second USB connection controls the independent, internal power meter.

Loadboard

This loadboard version of the AutoCal Unit can be used in standalone mode or it can be mated to an AXIe Series testhead.

See the Note on page xi concerning correct use of this unit, which is not supplied with a protective enclosure.

Loadboard used in standalone mode

The loadboard version of the AutoCal Unit can be used in a standalone mode. When the loadboard is not attached to an AXIe testhead, the mating surface exposes all the additional necessary connectors.

Figure 26 AutoCal Unit — loadboard version

- Eight RF channel SMA connectors
- +48 VDC external supply connector
- USB connector for control of the AutoCal circuitry.

When the loadboard is used in standalone mode, cabling and connections are the same as for the benchtop AutoCal Unit.
Loadboard mated to testhead

The loadboard is mated to the testhead and secured with the two butterfly screws.

The eight channels (ports) of the AXIe chassis are connected automatically to the loadboard through the RF blind-mate connector assembly on the mating surface of the loadboard.

Power (+48 VDC) to the loadboard is provided directly from the AXIe chassis. Access to chassis power is provided through a connector on the mating surface of the loadboard. No external power adapter is required.

**CAUTION**

DO NOT connect the external +48 VDC supply to the loadboard before attaching it to the testhead. Doing so could damage the loadboard.

DO NOT exceed +20 dBm input level. Higher levels will damage the internal power meter.

Control of the loadboard's circuitry is provided through the AXIe chassis as well.

One external USB connection is required to control the independent internal power meter.

Standards data and serial number

The standards data for the AutoCal Unit (either version) is stored on disk. The folder containing the standards data must be named:

AutoCal_Data_TEVxxxx where TEVxxxx is the serial number of the AutoCal Unit.

The serial number is stored in the EEPROM of the unit.

This folder must be in the CalData directory. The path of the caldata directory is read from the registry with the key:

For Windows 32-bit:
HKEY_LOCAL_MACHINE\SOFTWARE\TEV\AXRF\CalPath

For Windows 64-bit:
HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\TEV\AXRF\CalPath

The default path added to the registry by the AXRF software installer is:

For Windows 32-bit:
C:\Program Files\TEV\AXRF Subsystem\caldata

For Windows 64-bit:
C:\Program Files(X86)\TEV\AXRF Subsystem\caldata

Standards data is supplied on a USB stick with a new AutoCal Unit, or when a unit is returned from factory calibration. Copy the files from the USB stick to the standards data folder before using the AutoCal Unit.
Calibration command line options

A command line interface to the Control Panel has been added to allow calibration and diagnostics to be run from the command line. The available commands are:

- Scalar <Focused Frequency File> -[ACL | -NOACL]
- Vector <Focused Frequency File> -[ACL | -NOACL]
- Noise <Focused Frequency File> -[ACL | -NOACL]
- Checker

<Focused Frequency File> contains a list of frequencies to run during calibration.

Include the full directory path to a file with a .cal extension.

The file is created by using the Save Calibration Frequencies As... item under the Frequencies dropdown menu.

-ACL AutoCal Unit will be used
-NOACL AutoCal Unit will not be used.

Calibration files

Scalar calibration

Scalar calibration files calibration files are in a folder in the same directory as the executable:

Folder name: CalData\CalData Folder name: CalData\CalData
File names: AXRFScalarFactors.cal AXRFVectorErrorTerms.cal

Vector calibration

AXRF vector calibration requires calibration data associated with the vector calibration kit (the SHORT, OPEN, LOAD and THRU standards). Vector calibration files are in a folder in the same directory as the executable:

Folder name: CalData\VectorCalKitData

Vector calibration kit file names:

Open_1.s1p Open_2.s1p Open_3.s1p Open_4.s1p
Open_5.s1p Open_6.s1p Open_7.s1p Open_8.s1p
Short_1.s1p Short_2.s1p Short_3.s1p Short_4.s1p
Short_5.s1p Short_6.s1p Short_7.s1p Short_8.s1p
Load_1.s1p Load_2.s1p Load_3.s1p Load_4.s1p
Load_5.s1p Load_6.s1p Load_7.s1p Load_7.s1p
Thru_12.s2p Thru_34.s2p Thru_56.s2p Thru_78.s2p

The calibration files are formatted as s1p and s2p files created by an HP8720D Network Analyzer.

When a new calibration is executed, the old calibration files are saved to a folder in the CalData directory:

AXRF_ArchivedCalFactors

Each archived file is saved with the AutoCal Unit’s serial number and archive date attached to the file name:

AC_TEV0002_05-31-2014_13_58_37_AXRFScalarCalFactors.ca
Hardware description

AXRF RF Subsystem functional description

The AXRF RF Subsystem contains up to four Dual RF Port Modules (DRPM). Each DRPM contains two bi-directional ports that can either source or measure RF signals. In addition, each of these ports is capable of making noise figure and S-parameter measurements. The AXRF RF Subsystem has the capability of having up to eight bi-directional RF ports.

The AXRF RF Subsystem consists of the following:

- Up to two RF source signal generators
- RF Splitter Module — RFSM
- RF Mux Module — RFM
- One, two or four Dual RF Port Modules — DRPM
- RF digitizer

The AXRF RF Subsystem (Figure 27) contains a source splitter module that splits and connects the RF signal from the source generator(s) to one or more of the four DRPMs. The AXRF RF Subsystem also contains an RF mux module that connects the RF output of any of the DRPMs to the RF digitizer for RF measurements.
RF Splitter Module — RFSM

A functional block diagram of the source splitter module is shown in Figure 28. The RF Splitter Module (RFSM) provides path switching for the RF source generator(s) to one or more of the four DRPMs in the AXRF RF Subsystem. The RF source signal comes from the generator (AXRF Option 11 or Option 13) and is connected to IN A (Source A). A second RF signal source (AXRF Option 12) can be connected to IN B (Source B). The RFSM allows either signal source to be selected individually or both linearly combined to produce a two-tone signal. The user can also select between Source A and an internal noise source.

The selected RF signal(s) can then be switched to any of the four output ports. Each output leg contains adjustable RF attenuators to equalize the signal levels or independently adjust the output levels, which are then used to drive the DRPM modules.

See Current consumption for the current consumed by the RFSM.
Dual RF Port Module — DRPM

The DRPM functional block diagram is shown in Figure 29.

The RF source signal from one of the RFSM outputs is connected to the RF IN connector of the DRPM. The DRPM has two RF I/O ports; PORT A and PORT B. These can be configured to either source or measure RF signal signals. The DUT (device under test) is connected to these ports. If for example, PORT A is being used as a stimulus to the DUT, the RF signal from the RFSM is amplified and switched to this port. The RF signal levels present at either of the forward and reverse directional coupler ports or the "through" path of the signal present at PORT B can be switched to the LO IN/RF OUT port for measurement. This allows the AXRF software to make all of the measurements necessary for 1- and 2-port S-parameters, insertion loss and return loss. The measured signals are passed through a 0 to 31 dB attenuator for input ranging. The signal outputs from the LO IN/RF OUT ports of up to four DRPMs are selected by the RFM module and measured by the RF digitizer (AXRF Option 09 or 10).

The DRPM contains additional circuitry (Lo/Hi mixers and associated circuit) that can down-convert the selected measurement signal. If this is used, an external local oscillator signal needs to be supplied to the LO IN/RF OUT port. The down-converted IF signal is then available at the IF OUT port.

See Current consumption for the current consumed by the DRPM.
RF Mux Module — RFM

The AXRF RF Mux Module is shown in Figure 30. The RF Mux Module is used to switch the RF output of one of the four DRPMs to the input of the RF digitizer.

See Current consumption for the current consumed by the RFM.

RF digitizer

There is one RF digitizer in the AXRF RF Subsystem. The RF output of each AXRF DRPM can be connected to the RF digitizer through the RF Mux (RFM).

The standard RF digitizer used by AXRF RF Subsystem is the Cobham 3035C (Option 09, 6 GHz). The Cobham 3036 is available as Option 10 (13 GHz).

RF source generators

The AXRF RF Subsystem can be configured with up to two RF source generators installed in the PXI chassis. They are referred to as the Source A and Source B generators.

The standard RF source used by AXRF RF Subsystem is the Cobham 3025C (Option 11). A higher-power source is available: the Cobham 3026C (Option 13, not available with Option 11). A second source (Option 12) can be fitted when Option 11 or Option 13 is fitted.
Current consumption

The following current measurements table shows the typical values of current drawn from the 3.3 V, 5 V, and ±12 V PXI supply rails for each of the AXRF modules (RFSM, DRPM, and RFM). For the RFSM and DRPM, the currents are listed for a number of modes of operation. The last condition for each module listed gives the maximum current for settings that would be expected by the AXRF system.

The accuracy of the current measurements is ±10 % for the 3.3 V, 5 V and 12 V supplies and ±15 % or a minimum of 100 mA for the −12 V rail.

Current measurements

<table>
<thead>
<tr>
<th>Module type</th>
<th>Supply rail</th>
<th>Module set-up</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>3.3 V</td>
<td>5 V</td>
</tr>
<tr>
<td>RFSM</td>
<td>0.35</td>
<td>0.33</td>
</tr>
<tr>
<td></td>
<td>0.35</td>
<td>0.78</td>
</tr>
<tr>
<td></td>
<td>0.35</td>
<td>0.78</td>
</tr>
<tr>
<td></td>
<td>0.35</td>
<td>0.78</td>
</tr>
<tr>
<td>DRPM</td>
<td>0.35</td>
<td>0.02</td>
</tr>
<tr>
<td></td>
<td>0.35</td>
<td>0.02</td>
</tr>
<tr>
<td></td>
<td>0.35</td>
<td>0.02</td>
</tr>
<tr>
<td>RFM</td>
<td>0.32</td>
<td>0.01</td>
</tr>
</tbody>
</table>

*Note:* values are typical current drawn in Amps from each supply rail unless otherwise stated.
Maintenance

Cooling

Cooling fans in the AXRF RF Subsystem chassis draw air across the installed modules from inlets around the chassis. To maintain correct cooling, ensure that inlets and outlets are unobstructed during operation. There are no user-accessible fan filters.

WARNING

Moving the AXRF RF Subsystem

The combined weight of the AXRF RF Subsystem in its maximum typical populated configuration (10 PXI modules plus controller) may be more than 18 kg (40 lb), which exceeds the recommended maximum weight for manual handling by a single person. In this case, use two people to lift the instrument. Use handle(s) when provided. Be careful when moving the chassis to avoid any possible injury.

Cleaning the exterior

Make sure that the system is turned off before cleaning the exterior of the instrument. Wipe the exterior with a clean cloth, starting from areas that easily accumulate dust or dirt such as the area in and around the air inlet apertures for the AXRF RF Subsystem chassis and power supply. Take care not to damage exposed components or wiring on the loadboard version of the AutoCal Unit.

Power requirements

Make sure that the power cord is in good condition before plugging it into the system. It is important to check the reliability of the power source. Instrument voltage and frequency requirements are given in Connecting to supply. Do not connect instruments to an already overloaded circuit.

Service or assistance

Contact us should you require any service or assistance. See the support contact details at the end of this document or at www.cobham.com/wireless.
AEROFLEX LIMITED
SOFTWARE LICENCE AND WARRANTY

This document is an Agreement between the user of this Licensed Software, the Licensee, and Aeroflex Limited (‘Aeroflex’), the Licensor. By installing or commencing to use the Licensed Software you accept the terms of this Agreement. If you do not agree to the terms of this Agreement do not use the Licensed Software.

1. DEFINITIONS

The following expressions will have the meanings set out below for the purposes of this Agreement:

Add-In Application SoftwareLicensed Software that may be loaded separately from time to time into the Designated Equipment to improve or modify its functionality

Computer Application SoftwareLicensed Software supplied to run on a standard PC or workstation

Designated Equipment means either:
the single piece of equipment or system supplied by Aeroflex upon which the Licensed Software is installed; or
a computer that is connected to a single piece of equipment or system supplied by Aeroflex upon which computer the Licensed Software is installed

Downloaded Software any software downloaded from an Aeroflex web site

Embedded Software Licensed Software that forms part of the Designated Equipment supplied by Aeroflex and without which the Equipment cannot function

Licence Fee means either the fee paid or other consideration given to Aeroflex for the use of the Licensed Software on the Designated Equipment

Licensed Software all and any programs, listings, flow charts and instructions in whole or in part including Add-in, Computer Application, Downloaded and Embedded Software supplied to work with Designated Equipment

Licensee the organization or individual that is the user of the Licensed Software

PXI Software Licensed Software specific to Aeroflex’s 3000 Series PXI product range

2. LICENCE FEE

The Licensee shall pay the Licence Fee to Aeroflex in accordance with the terms of the contract between the Licensee and Aeroflex.

3. TERM

This Agreement shall be effective from the date of receipt or download (where applicable) of the Licensed Software by the Licensee and shall continue in force until terminated under the provisions of Clause 8.

4. LICENCE

4.1 The following rights and restrictions in this Article 4 apply to all Licensed Software unless otherwise expressly stated in other Articles of this Agreement.

4.2 Unless and until terminated, this Licence confers upon the Licensee the non-transferable and non-exclusive right to use the Licensed Software on the Designated Equipment.

4.3 Neither the Licensed Software nor any information provided by Aeroflex to the Licensee nor any licence and rights granted hereunder, may be sold, leased, assigned, sublicensed, electronically distributed, timeshared or otherwise transferred, in whole or in part by the Licensee other than as specified in this Licence without the prior written consent of Aeroflex. Such consent may be withheld at Aeroflex’s sole discretion.

4.4 The Licensee may not use the Licensed Software on other than the Designated Equipment, unless written permission is first obtained from Aeroflex and until the appropriate additional Licence Fee has been paid to Aeroflex.

4.5 The Licensee may not amend or alter the Licensed Software and shall have no right or licence other than that stipulated herein.

4.6 Except as specifically permitted elsewhere in this Agreement the Licensee may make not more than two copies of the Licensed Software (but not the Authoring and Language Manuals) in machine-readable form for operational security and shall ensure that all such copies include Aeroflex’s copyright notice, together with any features which disclose the name of the Licensed Software and the Licensee. Furthermore, the Licensee shall not permit the Licensed Software or any part to be disclosed in any form to any third party and shall maintain the Licensed Software in secure premises to prevent any unauthorized disclosure. The Licensee shall notify Aeroflex immediately if the Licensee has knowledge that any unlicensed party possesses the Licensed Software. The Licensee’s obligation to maintain confidentiality shall cease when the Licensed Software and all copies have been destroyed or returned. The copyright in the Licensed Software shall remain with Aeroflex. The Licensee will permit Aeroflex at all reasonable times to audit the use of the Licensed Software.
4.7 The Licensee will not disassemble or reverse engineer the Licensed Software, nor sub-license, lease, rent or part with possession or otherwise transfer the whole or any part of the Licensed Software.

5 ADDITIONAL LICENCE RIGHTS SPECIFIC TO PXI SOFTWARE

5.1 Definitions for PXI Software

The following expressions will have the meanings set out below for the purposes of the supplementary rights granted in this Article.

**PXI Drivers**
All 3000 Series PXI module device drivers including embedded firmware that are installed at runtime

**PXI Executable Applications**
All executable applications supplied with each 3000 Series PXI module including:
- PXI Studio
- Soft Front Panels (manual operation graphical user interfaces)
- Utilities including: RF Investigator, PXI Version Information and Self Test

**PXI Spectrum Analysis Library**
The spectrum analysis measurement suite library .dll software supplied with each 3000 Series PXI module

**PXI Optional Application Library**
Individual measurement suite available from a range of optional .dll application libraries

5.2 PXI Drivers, PXI Executable Applications and PXI Spectrum Analysis Library Licence Rights

Subject to the licence granted in Article 4 hereof notwithstanding the limitations on number of copies in Clause 4.5 hereof, the Licensee is entitled to make and distribute as many copies of the PXI Drivers and PXI Executable Applications as necessary for use with 3000 Series PXI modules acquired by the Licensee from Aeroflex or its authorized distributor or reseller provided that the Licensee may not sell or charge a fee for the PXI Drivers and PXI Executable Applications.

5.3 PXI Optional Application Library Licence Rights

Subject to the licence granted in Article 4 hereof notwithstanding the limitations on number of copies in Clause 4.5 hereof, the Licensee is entitled to distribute as many copies of any PXI Optional Application Library as necessary for use with 3000 Series PXI modules acquired by the Licensee from Aeroflex or its authorized distributor or reseller provided that:

5.3.1 copies of the applicable PXI Optional Application Library are used solely with 3000 Series PXI modules which the customer has purchased with the corresponding option or part number for the applicable PXI Optional Application Library; and

5.3.2 the Licensee may not sell or charge a fee for the PXI Optional Application Library.

6 WARRANTY

6.1 Aeroflex certifies that the Licensed Software supplied by Aeroflex will at the time of delivery function substantially in accordance with the applicable Software Product Descriptions, Data Sheets or Product Specifications published by Aeroflex.

6.2 The warranty period (unless an extended warranty for Embedded Software has been purchased) from date of delivery in respect of each type of Licensed Software is:

<table>
<thead>
<tr>
<th>Type of Software</th>
<th>Warranty Period</th>
</tr>
</thead>
<tbody>
<tr>
<td>PXI Drivers</td>
<td>24 months</td>
</tr>
<tr>
<td>Embedded Software</td>
<td>12 months</td>
</tr>
<tr>
<td>Add-In Application Software</td>
<td>90 days</td>
</tr>
<tr>
<td>Computer Application Software</td>
<td>90 days</td>
</tr>
<tr>
<td>Downloaded Software</td>
<td>No warranty</td>
</tr>
</tbody>
</table>

6.3 If during the appropriate Warranty Period the Licensed Software does not conform substantially to the Software Product Descriptions, Data Sheets or Product Specifications Aeroflex will provide:

6.3.1 In the case of Embedded Software and at Aeroflex’s discretion either a fix for the problem or an effective and efficient work-around.

6.3.2 In the case of Add-In Application Software and Computer Application Software and at Aeroflex’s discretion replacement of the software or a fix for the problem or an effective and efficient work-around.

6.4 Aeroflex does not warrant that the operation of any Licensed Software will be uninterrupted or error free.

6.5 The above Warranty does not apply to:

6.5.1 Defects resulting from software not supplied by Aeroflex, from unauthorized modification or misuse or from operation outside of the specification.

6.5.2 Third party produced proprietary software (‘Third Party Software’) which Aeroflex may deliver with its products (where such Third Party Software carries a more limited warranty than the above warranty). In such case Aeroflex will provide a remedy for any non-conformance of the Third Party Software commensurate with the third party’s warranty to Aeroflex (if any).

6.6 The remedies offered above are sole and exclusive remedies and to the extent permitted by applicable law are in lieu of any implied conditions, guarantees or warranties whatsoever and whether statutory or otherwise as to the Licensed Software all of which are hereby expressly excluded.
7. INDEMNITY

7.1 Aeroflex shall defend, at its expense, any action brought against the Licensee alleging that the Licensed Software infringes any patent, registered design, trademark or copyright, and shall pay all Licensee's costs and damages finally awarded up to an aggregate equivalent to the Licence Fee provided the Licensee shall not have done or permitted to be done anything which may have been or become any such infringement and shall have exercised reasonable care in protecting the same failing which the Licensee shall indemnify Aeroflex against all claims costs and damages incurred and that Aeroflex is given prompt written notice of such claim and given information, reasonable assistance and sole authority to defend or settle such claim on behalf of the Licensee. In the defense or settlement of any such claim, Aeroflex may obtain for the Licensee the right to continue using the Licensed Software or replace it or modify it so that it becomes non-infringing.

7.2 Aeroflex shall not be liable if the alleged infringement:

7.2.1 is based upon the use of the Licensed Software in combination with other software not furnished by Aeroflex, or
7.2.2 is based upon the use of the Licensed Software alone or in combination with other software in equipment not functionally identical to the Designated Equipment, or
7.2.3 arises as a result of Aeroflex having followed a properly authorized design or instruction of the Licensee, or
7.2.4 arises out of the use of the Licensed Software in a country other than the one disclosed to Aeroflex as the intended country of use of the Licensed Software at the commencement of this Agreement.

7.3 Aeroflex shall not be liable to the Licensee for any loss of use or for loss of profits or of contracts arising directly or indirectly out of any such infringement of patent, registered design, trademark or copyright.

7.4 Other than as may be covered by the indemnity provisions of Clauses 7.1, 7.2 and 7.3 Aeroflex shall not, under any circumstances, be liable to the Licensee for any direct, indirect, special, consequential and/or contingent loss or damage (and such loss or damage shall include without limitation loss of use or profit, loss of revenue, loss of product, liquidated damages or penalties, economic loss, delay in operations, loss of contracts or loss of business) whether or not the same are foreseeable and whether arising out of breach of contract, tort, statutory duty or otherwise. The total liability of Aeroflex and its employees, in contract, tort, or otherwise (including negligence, warranty, indemnity, and strict liability) howsoever arising out of this Licence shall be limited to the total amount of the Licence Fee and total support fees actually paid to Aeroflex by the Licensee.

8. TERMINATION

8.1 Notwithstanding anything herein to the contrary, this Licence shall forthwith determine if the Licensee:

8.1.1 As an individual has a Receiving Order made against him or is adjudicated bankrupt or compounds with creditors or as a corporate body, compounds with creditors or has a winding-up order made against it or
8.1.2 Parts with possession of the Designated Equipment.

8.2 This Licence may be terminated by notice in writing to the Licensee if the Licensee shall be in breach of any of its obligations hereunder and continue in such breach for a period of 21 days after notice thereof has been served on the Licensee.

8.3 On termination of this Agreement for any reason, Aeroflex may require the Licensee to return to Aeroflex all copies of the Licensed Software in the custody of the Licensee and the Licensee shall, at its own cost and expense, comply with such requirement within 14 days and shall, at the same time, certify to Aeroflex in writing that all copies of the Licensed Software in whatever form have been obliterated from the Designated Equipment.

9. THIRD PARTY LICENCES

9.1 The Licensed Software or part thereof may be the proprietary property of third party licensors. In such an event such third party licensors (as may be referenced on the software media, or any on screen message on start up of the software or on the order acknowledgement) and/or Aeroflex may directly enforce the terms of this Agreement and may terminate the Agreement if the Licensee is in breach of the conditions contained herein.

9.2 If any third party software supplied with the Licensed Software is supplied with, or contains or displays the third party’s own licence terms then the Licensee shall abide by such third party licence terms (for the purpose of this Article the term “third party” shall include other companies within the Aeroflex group of companies).

10. EXPORT REGULATIONS

The Licensee undertakes that where necessary the Licensee will conform with all relevant export regulations imposed by the Governments of the United Kingdom and/or the United State of America.

11. U.S. GOVERNMENT RESTRICTED RIGHTS

The Licensed Software and documentation are provided with RESTRICTED RIGHTS. Use, duplication, or disclosure by the Government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFAR 252.227-7013 or subparagraphs (c)(1) and (2) of the Commercial Computer Software-Restricted Rights at 48 CFR 52.227-19, as applicable.

12. NOTICES

Any notice to be given by the Licensee to Aeroflex shall be addressed to:
Aeroflex Limited, Longacres House, Six Hills Way, Stevenage, SG1 2AN, UK.

13. LAW AND JURISDICTION

This Agreement shall be governed by the laws of England and shall be subject to the exclusive jurisdiction of the English courts. This agreement constitutes the whole agreement between the parties and may be changed only by a written agreement signed by both parties.