

R&S®TSMW Interface & Programming Manual R&S®TSMW-K1 Software Manual



1503.3776.32 – 07

The Software Manual describes the following R&S®TSMW Interface options:

- R&S®TSMW-K1 – Digital I/Q SW Option (1503.3960.02)

Dear Customer,

The R&S® TSMW-K1 makes use of open source software packages. The most important of them are listed below together with their corresponding open source license. The verbatim license texts are provided on the user documentation CD-ROM (included in delivery).

Package	Link	License
OpenSSL	http://www.openssl.org/	OpenSSL / SSLeay

The R&S® TSMW-K1 includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit ("http://www.openssl.org/"). It includes cryptographic software written by Eric Young (eay@cryptsoft.com) and software written by Tim Hudson (tjh@cryptsoft.com).

Rohde&Schwarz would like to thank the open source community for their valuable contribution to embedded computing.

© 2011 Rohde & Schwarz GmbH & Co. KG

81671 Munich, Germany

Printed in Germany – Subject to change – Data without tolerance limits is not binding.

R&S® is a registered trademark of Rohde & Schwarz GmbH & Co. KG.

Trade names are trademarks of the owners.

The following abbreviations are used throughout this manual:

R&S® TSMW-K1 is abbreviated as R&S TSMW-K1.

R&S® TSMW-B1 is abbreviated as R&S TSMW-B1.

Radio Network Analyzer R&S®TSMW is abbreviated as R&S TSMW.

MATLAB® is abbreviated as MATLAB.

Basic Safety Instructions

Always read through and comply with the following safety instructions!

All plants and locations of the Rohde & Schwarz group of companies make every effort to keep the safety standards of our products up to date and to offer our customers the highest possible degree of safety. Our products and the auxiliary equipment they require are designed, built and tested in accordance with the safety standards that apply in each case. Compliance with these standards is continuously monitored by our quality assurance system. The product described here has been designed, built and tested in accordance with the attached EC Certificate of Conformity and has left the manufacturer's plant in a condition fully complying with safety standards. To maintain this condition and to ensure safe operation, you must observe all instructions and warnings provided in this manual. If you have any questions regarding these safety instructions, the Rohde & Schwarz group of companies will be happy to answer them.

Furthermore, it is your responsibility to use the product in an appropriate manner. This product is designed for use solely in industrial and laboratory environments or, if expressly permitted, also in the field and must not be used in any way that may cause personal injury or property damage. You are responsible if the product is used for any intention other than its designated purpose or in disregard of the manufacturer's instructions. The manufacturer shall assume no responsibility for such use of the product.

The product is used for its designated purpose if it is used in accordance with its product documentation and within its performance limits (see data sheet, documentation, the following safety instructions). Using the product requires technical skills and a basic knowledge of English. It is therefore essential that only skilled and specialized staff or thoroughly trained personnel with the required skills be allowed to use the product. If personal safety gear is required for using Rohde & Schwarz products, this will be indicated at the appropriate place in the product documentation. Keep the basic safety instructions and the product documentation in a safe place and pass them on to the subsequent users.

Observing the safety instructions will help prevent personal injury or damage of any kind caused by dangerous situations. Therefore, carefully read through and adhere to the following safety instructions before and when using the product. It is also absolutely essential to observe the additional safety instructions on personal safety, for example, that appear in relevant parts of the product documentation. In these safety instructions, the word "product" refers to all merchandise sold and distributed by the Rohde & Schwarz group of companies, including instruments, systems and all accessories.

Symbols and safety labels

							
Notice, general danger location Observe product documentation	Caution when handling heavy equipment	Danger of electric shock	Warning! Hot surface	PE terminal	Ground	Ground terminal	Be careful when handling electrostatic sensitive devices

					
ON/OFF supply voltage	Standby indication	Direct current (DC)	Alternating current (AC)	Direct/alternating current (DC/AC)	Device fully protected by double (reinforced) insulation

Tags and their meaning

The following signal words are used in the product documentation in order to warn the reader about risks and dangers.



indicates a hazardous situation which, if not avoided, will result in death or serious injury.



indicates a hazardous situation which, if not avoided, could result in death or serious injury.



indicates a hazardous situation which, if not avoided, could result in minor or moderate injury.



indicates the possibility of incorrect operation which can result in damage to the product.

In the product documentation, the word ATTENTION is used synonymously.

These tags are in accordance with the standard definition for civil applications in the European Economic Area. Definitions that deviate from the standard definition may also exist in other economic areas or military applications. It is therefore essential to make sure that the tags described here are always used only in connection with the related product documentation and the related product. The use of tags in connection with unrelated products or documentation can result in misinterpretation and in personal injury or material damage.

Operating states and operating positions

The product may be operated only under the operating conditions and in the positions specified by the manufacturer, without the product's ventilation being obstructed. If the manufacturer's specifications are not observed, this can result in electric shock, fire and/or serious personal injury or death. Applicable local or national safety regulations and rules for the prevention of accidents must be observed in all work performed.

1. Unless otherwise specified, the following requirements apply to Rohde & Schwarz products: predefined operating position is always with the housing floor facing down, IP protection 2X, pollution severity 2, overvoltage category 2, use only indoors, max. operating altitude 2000 m above sea level, max. transport altitude 4500 m above sea level. A tolerance of $\pm 10\%$ shall apply to the nominal voltage and $\pm 5\%$ to the nominal frequency.
2. Do not place the product on surfaces, vehicles, cabinets or tables that for reasons of weight or stability are unsuitable for this purpose. Always follow the manufacturer's installation instructions when installing the product and fastening it to objects or structures (e.g. walls and shelves). An installation that is not carried out as described in the product documentation could result in personal injury or death.
3. Do not place the product on heat-generating devices such as radiators or fan heaters. The ambient temperature must not exceed the maximum temperature specified in the product documentation or in the data sheet. Product overheating can cause electric shock, fire and/or serious personal injury or death.

Electrical safety

If the information on electrical safety is not observed either at all to the extent necessary, electric shock, fire and/or serious personal injury or death may occur.

1. Prior to switching on the product, always ensure that the nominal voltage setting on the product matches the nominal voltage of the AC supply network. If a different voltage is to be set, the power fuse of the product may have to be changed accordingly.
2. In the case of products of safety class I with movable power cord and connector, operation is permitted only on sockets with an earthing contact and protective earth connection.
3. Intentionally breaking the protective earth connection either in the feed line or in the product itself is not permitted. Doing so can result in the danger of an electric shock from the product. If extension cords or connector strips are implemented, they must be checked on a regular basis to ensure that they are safe to use.
4. If the product does not have a power switch for disconnection from the AC supply network, the plug of the connecting cable is regarded as the disconnecting device. In such cases, always ensure that the power plug is easily reachable and accessible at all times (corresponding to the length of connecting cable, approx. 2 m). Functional or electronic switches are not suitable for providing disconnection from the AC supply network. If products without power switches are integrated into racks or systems, a disconnecting device must be provided at the system level.
5. Never use the product if the power cable is damaged. Check the power cable on a regular basis to ensure that it is in proper operating condition. By taking appropriate safety measures and carefully laying the power cable, you can ensure that the cable will not be damaged and that no one can be hurt by, for example, tripping over the cable or suffering an electric shock.
6. The product may be operated only from TN/TT supply networks fused with max. 16 A (higher fuse only after consulting with the Rohde & Schwarz group of companies).
7. Do not insert the plug into sockets that are dusty or dirty. Insert the plug firmly and all the way into the socket. Otherwise, sparks that result in fire and/or injuries may occur.
8. Do not overload any sockets, extension cords or connector strips; doing so can cause fire or electric shocks.
9. For measurements in circuits with voltages $V_{rms} > 30$ V, suitable measures (e.g. appropriate measuring equipment, fusing, current limiting, electrical separation, insulation) should be taken to avoid any hazards.
10. Ensure that the connections with information technology equipment, e.g. PCs or other industrial computers, comply with the IEC60950-1/EN60950-1 or IEC61010-1/EN 61010-1 standards that apply in each case.
11. Unless expressly permitted, never remove the cover or any part of the housing while the product is in operation. Doing so will expose circuits and components and can lead to injuries, fire or damage to the product.
12. If a product is to be permanently installed, the connection between the PE terminal on site and the product's PE conductor must be made first before any other connection is made. The product may be installed and connected only by a licensed electrician.
13. For permanently installed equipment without built-in fuses, circuit breakers or similar protective devices, the supply circuit must be fused in such a way that anyone who has access to the product, as well as the product itself, is adequately protected from injury or damage.

Basic Safety Instructions

14. Use suitable overvoltage protection to ensure that no overvoltage (such as that caused by a bolt of lightning) can reach the product. Otherwise, the person operating the product will be exposed to the danger of an electric shock.
15. Any object that is not designed to be placed in the openings of the housing must not be used for this purpose. Doing so can cause short circuits inside the product and/or electric shocks, fire or injuries.
16. Unless specified otherwise, products are not liquid-proof (see also section "Operating states and operating positions", item 1. Therefore, the equipment must be protected against penetration by liquids. If the necessary precautions are not taken, the user may suffer electric shock or the product itself may be damaged, which can also lead to personal injury.
17. Never use the product under conditions in which condensation has formed or can form in or on the product, e.g. if the product has been moved from a cold to a warm environment. Penetration by water increases the risk of electric shock.
18. Prior to cleaning the product, disconnect it completely from the power supply (e.g. AC supply network or battery). Use a soft, non-linting cloth to clean the product. Never use chemical cleaning agents such as alcohol, acetone or diluents for cellulose lacquers.

Operation

1. Operating the products requires special training and intense concentration. Make sure that persons who use the products are physically, mentally and emotionally fit enough to do so; otherwise, injuries or material damage may occur. It is the responsibility of the employer/operator to select suitable personnel for operating the products.
2. Before you move or transport the product, read and observe the section titled "Transport".
3. As with all industrially manufactured goods, the use of substances that induce an allergic reaction (allergens) such as nickel cannot be generally excluded. If you develop an allergic reaction (such as a skin rash, frequent sneezing, red eyes or respiratory difficulties) when using a Rohde & Schwarz product, consult a physician immediately to determine the cause and to prevent health problems or stress.
4. Before you start processing the product mechanically and/or thermally, or before you take it apart, be sure to read and pay special attention to the section titled "Waste disposal", item 1.
5. Depending on the function, certain products such as RF radio equipment can produce an elevated level of electromagnetic radiation. Considering that unborn babies require increased protection, pregnant women must be protected by appropriate measures. Persons with pacemakers may also be exposed to risks from electromagnetic radiation. The employer/operator must evaluate workplaces where there is a special risk of exposure to radiation and, if necessary, take measures to avert the potential danger.
6. Should a fire occur, the product may release hazardous substances (gases, fluids, etc.) that can cause health problems. Therefore, suitable measures must be taken, e.g. protective masks and protective clothing must be worn.
7. If a laser product (e.g. a CD/DVD drive) is integrated into a Rohde & Schwarz product, absolutely no other settings or functions may be used as described in the product documentation. The objective is to prevent personal injury (e.g. due to laser beams).

Basic Safety Instructions

Repair and service

1. The product may be opened only by authorized, specially trained personnel. Before any work is performed on the product or before the product is opened, it must be disconnected from the AC supply network. Otherwise, personnel will be exposed to the risk of an electric shock.
2. Adjustments, replacement of parts, maintenance and repair may be performed only by electrical experts authorized by Rohde & Schwarz. Only original parts may be used for replacing parts relevant to safety (e.g. power switches, power transformers, fuses). A safety test must always be performed after parts relevant to safety have been replaced (visual inspection, PE conductor test, insulation resistance measurement, leakage current measurement, functional test). This helps ensure the continued safety of the product.

Batteries and rechargeable batteries/cells

If the information regarding batteries and rechargeable batteries/cells is not observed either at all or to the extent necessary, product users may be exposed to the risk of explosions, fire and/or serious personal injury, and, in some cases, death. Batteries and rechargeable batteries with alkaline electrolytes (e.g. lithium cells) must be handled in accordance with the EN 62133 standard.

1. Cells must not be taken apart or crushed.
2. Cells or batteries must not be exposed to heat or fire. Storage in direct sunlight must be avoided. Keep cells and batteries clean and dry. Clean soiled connectors using a dry, clean cloth.
3. Cells or batteries must not be short-circuited. Cells or batteries must not be stored in a box or in a drawer where they can short-circuit each other, or where they can be short-circuited by other conductive materials. Cells and batteries must not be removed from their original packaging until they are ready to be used.
4. Keep cells and batteries out of the hands of children. If a cell or a battery has been swallowed, seek medical aid immediately.
5. Cells and batteries must not be exposed to any mechanical shocks that are stronger than permitted.
6. If a cell develops a leak, the fluid must not be allowed to come into contact with the skin or eyes. If contact occurs, wash the affected area with plenty of water and seek medical aid.
7. Improperly replacing or charging cells or batteries that contain alkaline electrolytes (e.g. lithium cells) can cause explosions. Replace cells or batteries only with the matching Rohde & Schwarz type (see parts list) in order to ensure the safety of the product.
8. Cells and batteries must be recycled and kept separate from residual waste. Rechargeable batteries and normal batteries that contain lead, mercury or cadmium are hazardous waste. Observe the national regulations regarding waste disposal and recycling.

Transport

1. The product may be very heavy. Therefore, the product must be handled with care. In some cases, the user may require a suitable means of lifting or moving the product (e.g. with a lift-truck) to avoid back or other physical injuries.

2. Handles on the products are designed exclusively to enable personnel to transport the product. It is therefore not permissible to use handles to fasten the product to or on transport equipment such as cranes, fork lifts, wagons, etc. The user is responsible for securely fastening the products to or on the means of transport or lifting. Observe the safety regulations of the manufacturer of the means of transport or lifting. Noncompliance can result in personal injury or material damage.
3. If you use the product in a vehicle, it is the sole responsibility of the driver to drive the vehicle safely and properly. The manufacturer assumes no responsibility for accidents or collisions. Never use the product in a moving vehicle if doing so could distract the driver of the vehicle. Adequately secure the product in the vehicle to prevent injuries or other damage in the event of an accident.

Waste disposal

1. If products or their components are mechanically and/or thermally processed in a manner that goes beyond their intended use, hazardous substances (heavy-metal dust such as lead, beryllium, nickel) may be released. For this reason, the product may only be disassembled by specially trained personnel. Improper disassembly may be hazardous to your health. National waste disposal regulations must be observed.
2. If handling the product releases hazardous substances or fuels that must be disposed of in a special way, e.g. coolants or engine oils that must be replenished regularly, the safety instructions of the manufacturer of the hazardous substances or fuels and the applicable regional waste disposal regulations must be observed. Also observe the relevant safety instructions in the product documentation. The improper disposal of hazardous substances or fuels can cause health problems and lead to environmental damage.

Informaciones elementales de seguridad

Es imprescindible leer y observar las siguientes instrucciones e informaciones de seguridad!

El principio del grupo de empresas Rohde & Schwarz consiste en tener nuestros productos siempre al día con los estándares de seguridad y de ofrecer a nuestros clientes el máximo grado de seguridad. Nuestros productos y todos los equipos adicionales son siempre fabricados y examinados según las normas de seguridad vigentes. Nuestro sistema de garantía de calidad controla constantemente que sean cumplidas estas normas. El presente producto ha sido fabricado y examinado según el certificado de conformidad adjunto de la UE y ha salido de nuestra planta en estado impecable según los estándares técnicos de seguridad. Para poder preservar este estado y garantizar un funcionamiento libre de peligros, el usuario deberá atenerse a todas las indicaciones, informaciones de seguridad y notas de alerta. El grupo de empresas Rohde & Schwarz está siempre a su disposición en caso de que tengan preguntas referentes a estas informaciones de seguridad.

Además queda en la responsabilidad del usuario utilizar el producto en la forma debida. Este producto está destinado exclusivamente al uso en la industria y el laboratorio o, si ha sido expresamente autorizado, para aplicaciones de campo y de ninguna manera deberá ser utilizado de modo que alguna persona/cosa pueda sufrir daño. El uso del producto fuera de sus fines definidos o sin tener en cuenta las instrucciones del fabricante queda en la responsabilidad del usuario. El fabricante no se hace en ninguna forma responsable de consecuencias a causa del mal uso del producto.

Informaciones elementales de seguridad

Se parte del uso correcto del producto para los fines definidos si el producto es utilizado conforme a las indicaciones de la correspondiente documentación del producto y dentro del margen de rendimiento definido (ver hoja de datos, documentación, informaciones de seguridad que siguen). El uso del producto hace necesarios conocimientos técnicos y ciertos conocimientos del idioma inglés. Por eso se debe tener en cuenta que el producto solo pueda ser operado por personal especializado o personas instruidas en profundidad con las capacidades correspondientes. Si fuera necesaria indumentaria de seguridad para el uso de productos de Rohde & Schwarz, encontraría la información debida en la documentación del producto en el capítulo correspondiente. Guarde bien las informaciones de seguridad elementales, así como la documentación del producto, y entréguelas a usuarios posteriores.

Tener en cuenta las informaciones de seguridad sirve para evitar en lo posible lesiones o daños por peligros de toda clase. Por eso es imprescindible leer detalladamente y comprender por completo las siguientes informaciones de seguridad antes de usar el producto, y respetarlas durante el uso del producto. Deberán tenerse en cuenta todas las demás informaciones de seguridad, como p. ej. las referentes a la protección de personas, que encontrarán en el capítulo correspondiente de la documentación del producto y que también son de obligado cumplimiento. En las presentes informaciones de seguridad se recogen todos los objetos que distribuye el grupo de empresas Rohde & Schwarz bajo la denominación de "producto", entre ellos también aparatos, instalaciones así como toda clase de accesorios.

Símbolos y definiciones de seguridad

							
Aviso: punto de peligro general Observar la documentación del producto	Atención en el manejo de dispositivos de peso elevado	Peligro de choque eléctrico	Advertencia: superficie caliente	Conexión a conductor de protección	Conexión a tierra	Conexión a masa	Aviso: Cuidado en el manejo de dispositivos sensibles a la electrostática (ESD)

					
Tensión de alimentación de PUESTA EN MARCHA / PARADA	Indicación de estado de espera (Standby)	Corriente continua (DC)	Corriente alterna (AC)	Corriente continua / Corriente alterna (DC/AC)	El aparato está protegido en su totalidad por un aislamiento doble (reforzado)

Palabras de señal y su significado

En la documentación del producto se utilizan las siguientes palabras de señal con el fin de advertir contra riesgos y peligros.



PELIGRO identifica un peligro inminente con riesgo elevado que provocará muerte o lesiones graves si no se evita.



ADVERTENCIA identifica un posible peligro con riesgo medio de provocar muerte o lesiones (graves) si no se evita.



ATENCIÓN identifica un peligro con riesgo reducido de provocar lesiones leves o moderadas si no se evita.



AVISO indica la posibilidad de utilizar mal el producto y, como consecuencia, dañarlo.

En la documentación del producto se emplea de forma sinónima el término CUIDADO.

Las palabras de señal corresponden a la definición habitual para aplicaciones civiles en el área económica europea. Pueden existir definiciones diferentes a esta definición en otras áreas económicas o en aplicaciones militares. Por eso se deberá tener en cuenta que las palabras de señal aquí descritas sean utilizadas siempre solamente en combinación con la correspondiente documentación del producto y solamente en combinación con el producto correspondiente. La utilización de las palabras de señal en combinación con productos o documentaciones que no les correspondan puede llevar a interpretaciones equivocadas y tener por consecuencia daños en personas u objetos.

Estados operativos y posiciones de funcionamiento

El producto solamente debe ser utilizado según lo indicado por el fabricante respecto a los estados operativos y posiciones de funcionamiento sin que se obstruya la ventilación. Si no se siguen las indicaciones del fabricante, pueden producirse choques eléctricos, incendios y/o lesiones graves con posible consecuencia de muerte. En todos los trabajos deberán ser tenidas en cuenta las normas nacionales y locales de seguridad del trabajo y de prevención de accidentes.

1. Si no se convino de otra manera, es para los productos Rohde & Schwarz válido lo que sigue: como posición de funcionamiento se define por principio la posición con el suelo de la caja para abajo, modo de protección IP 2X, grado de suciedad 2, categoría de sobrecarga eléctrica 2, uso solamente en estancias interiores, utilización hasta 2000 m sobre el nivel del mar, transporte hasta 4500 m sobre el nivel del mar. Se aplicará una tolerancia de $\pm 10\%$ sobre el voltaje nominal y de $\pm 5\%$ sobre la frecuencia nominal.
2. No sitúe el producto encima de superficies, vehículos, estantes o mesas, que por sus características de peso o de estabilidad no sean aptos para él. Siga siempre las instrucciones de instalación del fabricante cuando instale y asegure el producto en objetos o estructuras (p. ej. paredes y estantes). Si se realiza la instalación de modo distinto al indicado en la documentación del producto, pueden causarse lesiones o incluso la muerte.
3. No ponga el producto sobre aparatos que generen calor (p. ej. radiadores o calefactores). La temperatura ambiente no debe superar la temperatura máxima especificada en la documentación del producto o en la hoja de datos. En caso de sobrecalentamiento del producto, pueden producirse choques eléctricos, incendios y/o lesiones graves con posible consecuencia de muerte.

Seguridad eléctrica

Si no se siguen (o se siguen de modo insuficiente) las indicaciones del fabricante en cuanto a seguridad eléctrica, pueden producirse choques eléctricos, incendios y/o lesiones graves con posible consecuencia de muerte.

1. Antes de la puesta en marcha del producto se deberá comprobar siempre que la tensión preseleccionada en el producto coincida con la de la red de alimentación eléctrica. Si es necesario modificar el ajuste de tensión, también se deberán cambiar en caso dado los fusibles correspondientes del producto.
2. Los productos de la clase de protección I con alimentación móvil y enchufe individual solamente podrán enchufarse a tomas de corriente con contacto de seguridad y con conductor de protección conectado.
3. Queda prohibida la interrupción intencionada del conductor de protección, tanto en la toma de corriente como en el mismo producto. La interrupción puede tener como consecuencia el riesgo de que el producto sea fuente de choques eléctricos. Si se utilizan cables alargadores o regletas de enchufe, deberá garantizarse la realización de un examen regular de los mismos en cuanto a su estado técnico de seguridad.
4. Si el producto no está equipado con un interruptor para desconectarlo de la red, se deberá considerar el enchufe del cable de conexión como interruptor. En estos casos se deberá asegurar que el enchufe siempre sea de fácil acceso (de acuerdo con la longitud del cable de conexión, aproximadamente 2 m). Los interruptores de función o electrónicos no son aptos para el corte de la red eléctrica. Si los productos sin interruptor están integrados en bastidores o instalaciones, se deberá colocar el interruptor en el nivel de la instalación.
5. No utilice nunca el producto si está dañado el cable de conexión a red. Compruebe regularmente el correcto estado de los cables de conexión a red. Asegúrese, mediante las medidas de protección y de instalación adecuadas, de que el cable de conexión a red no pueda ser dañado o de que nadie pueda ser dañado por él, p. ej. al tropezar o por un choque eléctrico.
6. Solamente está permitido el funcionamiento en redes de alimentación TN/TT aseguradas con fusibles de 16 A como máximo (utilización de fusibles de mayor amperaje solo previa consulta con el grupo de empresas Rohde & Schwarz).
7. Nunca conecte el enchufe en tomas de corriente sucias o llenas de polvo. Introduzca el enchufe por completo y fuertemente en la toma de corriente. La no observación de estas medidas puede provocar chispas, fuego y/o lesiones.
8. No sobrecargue las tomas de corriente, los cables alargadores o las regletas de enchufe ya que esto podría causar fuego o choques eléctricos.
9. En las mediciones en circuitos de corriente con una tensión $U_{\text{eff}} > 30 \text{ V}$ se deberán tomar las medidas apropiadas para impedir cualquier peligro (p. ej. medios de medición adecuados, seguros, limitación de tensión, corte protector, aislamiento etc.).
10. Para la conexión con dispositivos informáticos como un PC o un ordenador industrial, debe comprobarse que éstos cumplan los estándares IEC60950-1/EN60950-1 o IEC61010-1/EN 61010-1 válidos en cada caso.
11. A menos que esté permitido expresamente, no retire nunca la tapa ni componentes de la carcasa mientras el producto esté en servicio. Esto pone a descubierto los cables y componentes eléctricos y puede causar lesiones, fuego o daños en el producto.

12. Si un producto se instala en un lugar fijo, se deberá primero conectar el conductor de protección fijo con el conductor de protección del producto antes de hacer cualquier otra conexión. La instalación y la conexión deberán ser efectuadas por un electricista especializado.
13. En el caso de dispositivos fijos que no estén provistos de fusibles, interruptor automático ni otros mecanismos de seguridad similares, el circuito de alimentación debe estar protegido de modo que todas las personas que puedan acceder al producto, así como el producto mismo, estén a salvo de posibles daños.
14. Todo producto debe estar protegido contra sobretensión (debida p. ej. a una caída del rayo) mediante los correspondientes sistemas de protección. Si no, el personal que lo utilice quedará expuesto al peligro de choque eléctrico.
15. No debe introducirse en los orificios de la caja del aparato ningún objeto que no esté destinado a ello. Esto puede producir cortocircuitos en el producto y/o puede causar choques eléctricos, fuego o lesiones.
16. Salvo indicación contraria, los productos no están impermeabilizados (ver también el capítulo "Estados operativos y posiciones de funcionamiento", punto 1). Por eso es necesario tomar las medidas necesarias para evitar la entrada de líquidos. En caso contrario, existe peligro de choque eléctrico para el usuario o de daños en el producto, que también pueden redundar en peligro para las personas.
17. No utilice el producto en condiciones en las que pueda producirse o ya se hayan producido condensaciones sobre el producto o en el interior de éste, como p. ej. al desplazarlo de un lugar frío a otro caliente. La entrada de agua aumenta el riesgo de choque eléctrico.
18. Antes de la limpieza, desconecte por completo el producto de la alimentación de tensión (p. ej. red de alimentación o batería). Realice la limpieza de los aparatos con un paño suave, que no se deshilache. No utilice bajo ningún concepto productos de limpieza químicos como alcohol, acetona o diluyentes para lacas nitrocelulósicas.

Funcionamiento

1. El uso del producto requiere instrucciones especiales y una alta concentración durante el manejo. Debe asegurarse que las personas que manejen el producto estén a la altura de los requerimientos necesarios en cuanto a aptitudes físicas, psíquicas y emocionales, ya que de otra manera no se pueden excluir lesiones o daños de objetos. El empresario u operador es responsable de seleccionar el personal usuario apto para el manejo del producto.
2. Antes de desplazar o transportar el producto, lea y tenga en cuenta el capítulo "Transporte".
3. Como con todo producto de fabricación industrial no puede quedar excluida en general la posibilidad de que se produzcan alergias provocadas por algunos materiales empleados, los llamados alérgenos (p. ej. el níquel). Si durante el manejo de productos Rohde & Schwarz se producen reacciones alérgicas, como p. ej. irritaciones cutáneas, estornudos continuos, enrojecimiento de la conjuntiva o dificultades respiratorias, debe avisarse inmediatamente a un médico para investigar las causas y evitar cualquier molestia o daño a la salud.
4. Antes de la manipulación mecánica y/o térmica o el desmontaje del producto, debe tenerse en cuenta imprescindiblemente el capítulo "Eliminación", punto 1.

5. Ciertos productos, como p. ej. las instalaciones de radiocomunicación RF, pueden a causa de su función natural, emitir una radiación electromagnética aumentada. Deben tomarse todas las medidas necesarias para la protección de las mujeres embarazadas. También las personas con marcapasos pueden correr peligro a causa de la radiación electromagnética. El empresario/operador tiene la obligación de evaluar y señalizar las áreas de trabajo en las que exista un riesgo elevado de exposición a radiaciones.
6. Tenga en cuenta que en caso de incendio pueden desprenderse del producto sustancias tóxicas (gases, líquidos etc.) que pueden generar daños a la salud. Por eso, en caso de incendio deben usarse medidas adecuadas, como p. ej. máscaras antigás e indumentaria de protección.
7. En caso de que un producto Rohde & Schwarz contenga un producto láser (p. ej. un lector de CD/DVD), no debe usarse ninguna otra configuración o función aparte de las descritas en la documentación del producto, a fin de evitar lesiones (p. ej. debidas a irradiación láser).

Reparación y mantenimiento

1. El producto solamente debe ser abierto por personal especializado con autorización para ello. Antes de manipular el producto o abrirlo, es obligatorio desconectarlo de la tensión de alimentación, para evitar toda posibilidad de choque eléctrico.
2. El ajuste, el cambio de partes, el mantenimiento y la reparación deberán ser efectuadas solamente por electricistas autorizados por Rohde & Schwarz. Si se reponen partes con importancia para los aspectos de seguridad (p. ej. el enchufe, los transformadores o los fusibles), solamente podrán ser sustituidos por partes originales. Después de cada cambio de partes relevantes para la seguridad deberá realizarse un control de seguridad (control a primera vista, control del conductor de protección, medición de resistencia de aislamiento, medición de la corriente de fuga, control de funcionamiento). Con esto queda garantizada la seguridad del producto.

Baterías y acumuladores o celdas

Si no se siguen (o se siguen de modo insuficiente) las indicaciones en cuanto a las baterías y acumuladores o celdas, pueden producirse explosiones, incendios y/o lesiones graves con posible consecuencia de muerte. El manejo de baterías y acumuladores con electrolitos alcalinos (p. ej. celdas de litio) debe seguir el estándar EN 62133.

1. No deben desmontarse, abrirse ni triturarse las celdas.
2. Las celdas o baterías no deben someterse a calor ni fuego. Debe evitarse el almacenamiento a la luz directa del sol. Las celdas y baterías deben mantenerse limpias y secas. Limpiar las conexiones sucias con un paño seco y limpio.
3. Las celdas o baterías no deben cortocircuitarse. Es peligroso almacenar las celdas o baterías en estuches o cajones en cuyo interior puedan cortocircuitarse por contacto recíproco o por contacto con otros materiales conductores. No deben extraerse las celdas o baterías de sus embalajes originales hasta el momento en que vayan a utilizarse.
4. Mantener baterías y celdas fuera del alcance de los niños. En caso de ingestión de una celda o batería, avisar inmediatamente a un médico.
5. Las celdas o baterías no deben someterse a impactos mecánicos fuertes indebidos.

6. En caso de falta de estanqueidad de una celda, el líquido vertido no debe entrar en contacto con la piel ni los ojos. Si se produce contacto, lavar con agua abundante la zona afectada y avisar a un médico.
7. En caso de cambio o recarga inadecuados, las celdas o baterías que contienen electrolitos alcalinos (p. ej. las celdas de litio) pueden explotar. Para garantizar la seguridad del producto, las celdas o baterías solo deben ser sustituidas por el tipo Rohde & Schwarz correspondiente (ver lista de recambios).
8. Las baterías y celdas deben reciclarse y no deben tirarse a la basura doméstica. Las baterías o acumuladores que contienen plomo, mercurio o cadmio deben tratarse como residuos especiales. Respete en esta relación las normas nacionales de eliminación y reciclaje.

Transporte

1. El producto puede tener un peso elevado. Por eso es necesario desplazarlo o transportarlo con precaución y, si es necesario, usando un sistema de elevación adecuado (p. ej. una carretilla elevadora), a fin de evitar lesiones en la espalda u otros daños personales.
2. Las asas instaladas en los productos sirven solamente de ayuda para el transporte del producto por personas. Por eso no está permitido utilizar las asas para la sujeción en o sobre medios de transporte como p. ej. grúas, carretillas elevadoras de horquilla, carros etc. Es responsabilidad suya fijar los productos de manera segura a los medios de transporte o elevación. Para evitar daños personales o daños en el producto, siga las instrucciones de seguridad del fabricante del medio de transporte o elevación utilizado.
3. Si se utiliza el producto dentro de un vehículo, recae de manera exclusiva en el conductor la responsabilidad de conducir el vehículo de manera segura y adecuada. El fabricante no asumirá ninguna responsabilidad por accidentes o colisiones. No utilice nunca el producto dentro de un vehículo en movimiento si esto pudiera distraer al conductor. Asegure el producto dentro del vehículo debidamente para evitar, en caso de un accidente, lesiones u otra clase de daños.

Eliminación

1. Si se trabaja de manera mecánica y/o térmica cualquier producto o componente más allá del funcionamiento previsto, pueden liberarse sustancias peligrosas (polveros con contenido de metales pesados como p. ej. plomo, berilio o níquel). Por eso el producto solo debe ser desmontado por personal especializado con formación adecuada. Un desmontaje inadecuado puede ocasionar daños para la salud. Se deben tener en cuenta las directivas nacionales referentes a la eliminación de residuos.
2. En caso de que durante el trato del producto se formen sustancias peligrosas o combustibles que deban tratarse como residuos especiales (p. ej. refrigerantes o aceites de motor con intervalos de cambio definidos), deben tenerse en cuenta las indicaciones de seguridad del fabricante de dichas sustancias y las normas regionales de eliminación de residuos. Tenga en cuenta también en caso necesario las indicaciones de seguridad especiales contenidas en la documentación del producto. La eliminación incorrecta de sustancias peligrosas o combustibles puede causar daños a la salud o daños al medio ambiente.

Qualitätszertifikat

Certificate of quality

Certificat de qualité

Certified Quality System
ISO 9001

Certified Environmental System
ISO 14001

Sehr geehrter Kunde,

Sie haben sich für den Kauf eines Rohde&Schwarz-Produktes entschieden. Hiermit erhalten Sie ein nach modernsten Fertigungsmethoden hergestelltes Produkt. Es wurde nach den Regeln unseres Qualitätsmanagementsystems entwickelt, gefertigt und geprüft. Das Rohde&Schwarz-Qualitätsmanagementsystem ist u.a. nach ISO9001 und ISO 14001 zertifiziert.

Der Umwelt verpflichtet

- Energie-effiziente, RoHS-konforme Produkte
- Kontinuierliche Weiterentwicklung nachhaltiger Umweltkonzepte
- ISO 14001-zertifiziertes Umweltmanagementsystem

Dear Customer,

You have decided to buy a Rohde&Schwarz product. You are thus assured of receiving a product that is manufactured using the most modern methods available. This product was developed, manufactured and tested in compliance with our quality management system standards. The Rohde&Schwarz quality management system is certified according to standards such as ISO9001 and ISO 14001.

Environmental commitment

- Energy-efficient products
- Continuous improvement in environmental sustainability
- ISO 14001-certified environmental management system

Cher client,

Vous avez choisi d'acheter un produit Rohde&Schwarz. Vous disposez donc d'un produit fabriqué d'après les méthodes les plus avancées. Le développement, la fabrication et les tests respectent nos normes de gestion qualité. Le système de gestion qualité de Rohde&Schwarz a été homologué, entre autres, conformément aux normes ISO9001 et ISO 14001.

Engagement écologique

- Produits à efficience énergétique
- Amélioration continue de la durabilité environnementale
- Système de gestion de l'environnement certifié selon ISO 14001

Customer Support

Technical support – where and when you need it

For quick, expert help with any Rohde & Schwarz equipment, contact one of our Customer Support Centers. A team of highly qualified engineers provides telephone support and will work with you to find a solution to your query on any aspect of the operation, programming or applications of Rohde & Schwarz equipment.

Up-to-date information and upgrades

To keep your instrument up-to-date and to be informed about new application notes related to your instrument, please send an e-mail to the Customer Support Center stating your instrument and your wish. We will take care that you will get the right information.

Europe, Africa, Middle East

Phone +49 89 4129 12345
customersupport@rohde-schwarz.com

North America

Phone 1-888-TEST-RSA (1-888-837-8772)
customer.support@rsa.rohde-schwarz.com

Latin America

Phone +1-410-910-7988
customersupport.la@rohde-schwarz.com

Asia/Pacific

Phone +65 65 13 04 88
customersupport.asia@rohde-schwarz.com



Table of Contents

Documentation Overview	7
Conventions Used in the Documentation	8
1 Introduction	9
1.1 What's new	11
1.1.1 This release	11
1.1.2 Previous release	11
1.2 System Requirements	11
1.2.1 Additional Requirements for I/Q Streaming over LAN	12
1.2.1.1 Formula	12
1.2.1.2 Example I/Q Streaming	12
2 Install	13
2.1 Installation of the R&S TSMW-K1 software	13
2.2 Set the path in MATLAB to the R&S TSMW-K1 folder	15
3 Uninstall.....	16
4 R&S TSMW signal processing block diagram.....	17
5 Features.....	20
5.1 Measurement Types.....	20
5.2 Resource management for concurrent measurement requests	21
5.3 Trigger	22
5.4 GPS	22
6 Function Overview.....	23
6.1 General.....	23
6.2 Measurement.....	24
6.2.1 Functions	24
6.2.2 Templates.....	25
6.3 Seamless streaming of I/Q data in real time	25
6.3.1 Functions	26
6.3.2 Templates.....	27
6.4 GPS	27

6.5	Digital IQ Interface	28
7	Structures	29
7.1	Structure: MeasCtrl	30
7.1.1	MeasCtrl values for general measurement control	31
7.1.2	MeasCtrl values for streaming control	32
7.1.3	MeasCtrl values for capacity request for RF channels	33
7.1.4	MeasCtrl values for schedule control.....	34
7.1.5	MeasCtrl values for periodic measurement control	35
7.1.6	MeasCtrl values for trigger control.....	37
7.1.7	MeasCtrl values for RF channel 1 and RF channel 2.....	38
7.2	Structure: FilterSpec	40
7.3	Structure: TSMWOptions	41
7.4	Structure: TSMWIQResult	42
7.5	Structure: StreamStatus	43
7.6	Structure: StreamInfo	44
8	R&S TSMW I/Q Interface Application	45
8.1	Main Window “TSMW Matlab IQ Interface”	45
8.1.1	GUI – connection and measurement setting	46
8.1.2	GUI – output functions	50
8.2	Streaming Dialog	52
8.3	Filter Design Dialog	54
8.4	GPS Server Configuration Dialog	56
9	How to	59
9.1	Use the R&S TSMW MATLAB IQ Interface Application (compiled)	59
9.2	Use the R&S TSMW MATLAB IQ Interface Application (not compiled)	59
9.3	Use native C++ Interface	60
9.4	Connect to the R&S TSMW	60
9.5	Create a resampling filter	60
9.6	Use predefined filters	63
9.7	Modify the measurement values	63
9.8	Perform a continuous measurement	64
9.9	Perform a single measurement	64

9.10	Record streaming measurement data	65
9.11	View streaming measurement data.....	66
9.12	Export single data.....	66
9.13	Export streamed data.	67
9.14	Release the connection to the R&S TSMW IQ interface	68
9.15	Perform streaming measurements (C++ API)	68
10	Troubleshooting.....	69
10.1	Working with a second R&S TSMW	69
10.2	Problems during LAN connection to the R&S TSMW.....	69
10.3	Skipped I/Q data blocks during I/Q streaming.....	70
	Appendix	71
A	C++ Header Files.....	71
B	M-Functions.....	71
B.1	MeasCtrlTemplate.....	71
B.2	MeasCtrlTemplate_RF1	72
B.3	MeasCtrlTemplate_RF2	72
B.4	MeasCtrlTemplate_RF1Stream.....	73
B.5	MeasCtrlTemplate_RF2Stream.....	73
B.6	TSMWConnect.....	74
B.7	TSMWExtReEnable	75
B.8	TSMWGetFIRParam.....	75
B.9	TSMWGetIQTime	76
B.10	TSMWGetLastError	77
B.11	TSMWGetStreamError.....	77
B.12	TSMWGetVersion	78
B.13	TSMWGetVersionText	78
B.14	TSMWGPSEnable	79
B.15	TSMWGPSGetNMEALine	80
B.16	TSMWGPSGetServerStatus.....	80
B.17	TSMWGPSSendNMEACmd.....	81
B.18	TSMWGPSSendUBX_CFG_INF.....	82
B.19	TSMWGPSSendUBX_CFG_MSG.....	82
B.20	TSMWGPSSendUBX_CFG_PRT_AllIn_NMEAOut	83

B.21	TSMWGPSSendUBX_CFG_RATE	83
B.22	TSMWGPSSendUBXCmd(TSMWID, UBXCmd).....	84
B.23	TSMWGPSStartTcpServer.....	84
B.24	TSMWGPSStopServer	85
B.25	TSMWGPSSync	86
B.26	TSMWInitInterface	86
B.27	TSMWIQCloseStreamFile.....	87
B.28	TSMWIQDataAvailable	87
B.29	TSMWIQGetData.....	88
B.30	TSMWIQGetDataInt16.....	90
B.31	TSMWIQGetDataInt32.....	93
B.32	TSMWIQGetDataSingle.....	94
B.33	TSMWIQGetDataDouble	95
B.34	TSMWIQGetDataSingleIv	96
B.35	TSMWIQGetDataDoubleIv.....	97
B.36	TSMWIQGetStreamSingle.....	98
B.37	TSMWIQGetStreamDouble	99
B.38	TSMWIQGetStreamSingleIv	100
B.39	TSMWIQGetStreamDoubleIv.....	101
B.40	TSMWIQGetResultParam	102
B.41	TSMWIQMeasure	102
B.42	TSMWIQOpenStreamFile	104
B.43	TSMWIQPeriodCtrl	105
B.44	TSMWIQSetup.....	105
B.45	TSMWIQStream.....	106
B.46	TSMWIQStreamStatus.....	108
B.47	TSMWIQStopStreaming.....	109
B.48	TSMWIQGetStreamDouble	110
B.49	TSMWIQGetStreamSingleIv	110
B.50	TSMWIQGetStreamDoubleIv.....	110
B.51	TSMWIQGetResultParam	110
B.52	TSMWIQMeasure	110
B.53	TSMWIQOpenStreamFile	110

- B.54 TSMWIQPeriodCtrl 110
- B.55 TSMWIQSetup 110
- B.56 TSMWReleaseInterface 111
- B.57 TSMWResourceRequest 112
- B.58 TSMWSetDAC 113
- B.59 TSMWShutdown 113
- B.60 TSMWTGEnable 114
- B.61 TSMWTrigger 114
- B.62 TSMWINFOIFEnable 115
- B.63 TSMWINFOIFGetStatus 116

- Index 118**

Documentation Overview

This manual describes the R&S TSMW-K1 programming interfaces for R&S TSMW.

For details about the R&S ViCom programming interface refer to the operating manual "R&S ViCom Description for R&S TSMx Receivers" (1505.1329.42).

The R&S TSMW is a high-power platform for optimizing all conventional mobile radio networks. Two highly sensitive 20 MHz frontends for center frequencies from 30 MHz to 6 GHz, a dual-channel preselector and an FPGA-based software-defined architecture offer unsurpassed performance while providing maximum flexibility and future-proofness. In addition to functioning as a mobile radio scanner, the R&S TSMW makes also an ideal digital I/Q receiver for customer specific applications.

In chapter R&S TSMW signal processing block diagram on p. 17 you can find a block diagram of R&S TSMW.

Related Documentation

- For further information about the R&S TSMW and its functionality see the Operation Manual of R&S TSMW Radio Network Analyzer.
- For information of the general MATLAB functions see the online help in MATLAB.

Conventions Used in the Documentation

The following conventions are used throughout the R&S TSMW Interface Programming Manual:

Typographical conventions

Convention	Description
"Graphical user interface elements"	All names of graphical user interface elements both on the screen and on the front and rear panels, such as dialog boxes, softkeys, menus, options, buttons etc., are enclosed by quotation marks.
"KEYS"	Key names are written in capital letters and enclosed by quotation marks.
<i>Input</i>	Input to be entered by the user is displayed in italics.
File names, commands, program code	File names, commands, coding samples and screen output are distinguished by their font.
"Links"	Links that you can click are displayed in blue font.
"References"	References to other parts of the documentation are enclosed by quotation marks.

1 Introduction

An integral part of the R&S TSMW is its R&S Digital I/Q SW Option (R&S TSMW-K1). The R&S TSMW-K1 option is a data interface application. It allows the R&S TSMW I/Q to be controlled for making data measurements (for later analysis) as well as to measure I/Q data and stream it over the R&S TSMW-B1 interface to another R&S instrument as input (e.g. R&S IQR, R&S Signal Analyzer, R&S Signal Generator).

The R&S TSMW-K1 option provides a flexible MATLAB API as well as an equivalent C++ API for performing and controlling the measurements.

This enables you, for example, not only to design and analyze receiver algorithms in MATLAB, but also to port them to C++ for a real time version. Or you can even perform technology-independent channel measurements, which can be used to simulate realistic fading scenarios in the lab.

For transmission of measured I/Q data the R&S TSMW offers two hardware interfaces that can be accessed via R&S TSMW-K1 option:

- **Gigabit LAN Interface**

Measured I/Q data from periodic or single measurements are transmitted over the Gigabit LAN interface, since the bandwidth of an Ethernet connection is mostly sufficient for the transmission of I/Q data to the destination. In limited extent also I/Q data from a stream measurements could be transmitted over the Gigabit LAN interface. Please notice that the R&S TSMW could generate a higher bit rate than a Gigabit LAN supports, see also Additional Requirements for I/Q Streaming on p. 12.

- **R&S TSMW-B1 Interface**

The R&S TSMW hardware option R&S Digital I/Q Interface (R&S TSMW-B1) is provided for streaming measurements.

The R&S Digital I/Q Interface is a proprietary R&S standard and allows measured signals to be transferred easily between different R&S instruments. Furthermore, this interface offers a higher bandwidth than a Gigabit LAN connection and therefore makes it possible to use the R&S TSMW with higher sampling and transfer rates.

Hence, in order to use the full sampling rate for streaming measurements of I/Q data it is recommended to transfer the I/Q data over the R&S TSMW-B1 interface.

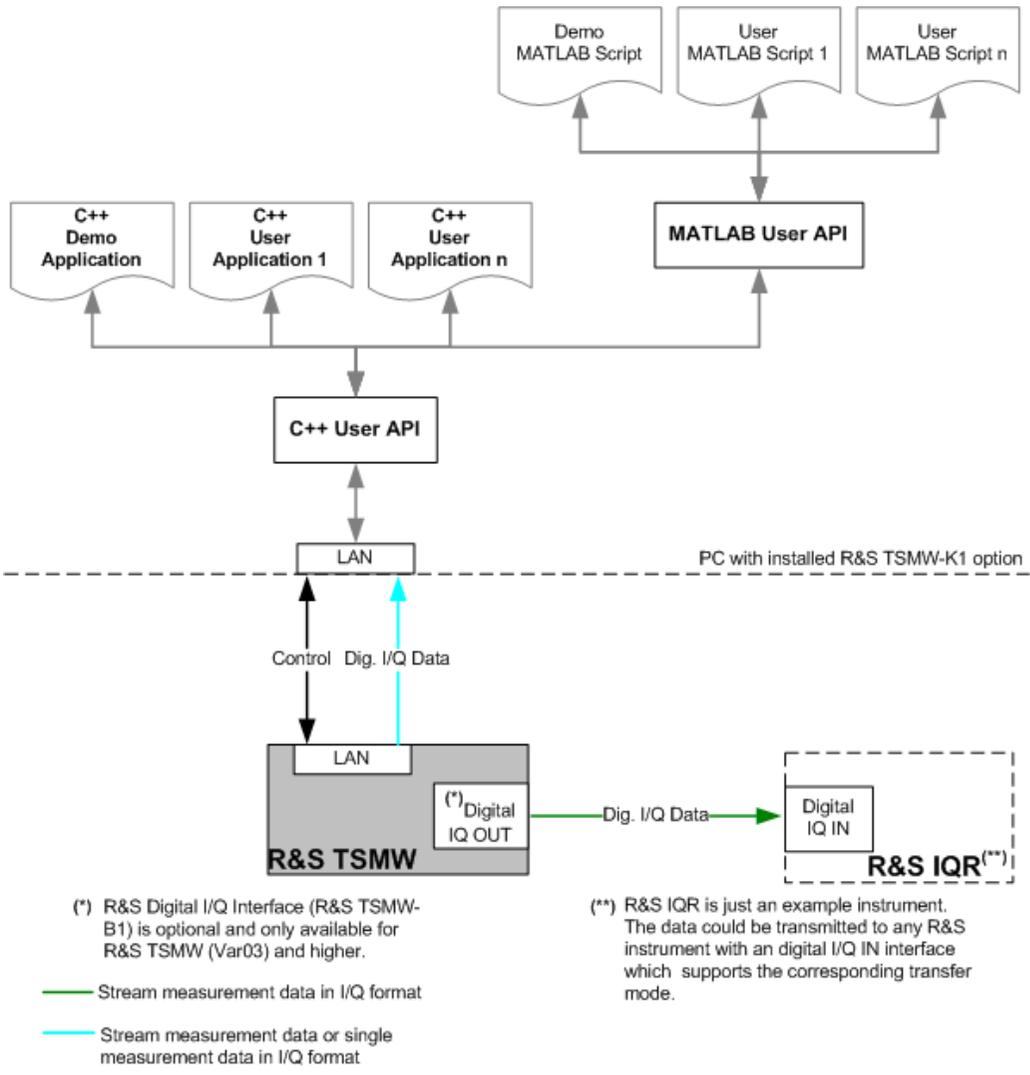


Figure 1-1: Impact of R&S TSMW-K1 for R&S TSMW

Several MATLAB and C++ example scripts are available on CD-ROM and on installation folder <TSMW-K1 installation directory>\Examples. They demonstrate example implementations of R&S TSMW-K1 functionalities.

1.1 What's new

1.1.1 This release

The new features in this version are:

- Digital I/Q Interface extended to support up to 8 time multiplexed channels
- Updated GUIs:
 - Main window (TSMWIQInterfaceDemo)
 - Streaming window (TSMWIQStreamingTool)
- New MATLAB and C-Functions

For an overview of all relevant MATLAB functions refer to the Appendix M-Functions on page 71.

For more details about new features, see the release notes of the current R&S TSMW-K1 software version.

1.1.2 Previous release

The new feature in the previous release was:

- TCP server implemented to transmit GPS NMEA data over LAN.

Changes related to this feature can be found in the chapters 6.4 for the API, and 8.4 for the GUI.

1.2 System Requirements

- 2GB RAM
- Gigabit Ethernet Adapter supporting 9kb Jumbo Frames
- CD-ROM drive (for installation only)
- MS Windows XP, Service Pack 2
- R&S TSMW
- MATLAB Software (including Filter Design Toolbox) Version R2007b (mandatory when using MATLAB source files, not necessary for compiled MATLAB demo application or when using C++ API)

1.2.1 Additional Requirements for I/Q Streaming over LAN

Use of the R&S TSMW feature "I/Q streaming" places additional requirements on the computer equipment used. The following items are essential:

- PCI-Express Gigabit Ethernet Adapter
- High-speed hard disk with >40MB/s sustained transfer rate (write) - for calculating the actual disc speed required, see the equation below.

These requirements are due to the large amount of data created during I/Q streaming.

Based on the above facts it is useful to calculate the I/Q data rate and to define the minimum system requirements for the computer equipment.

1.2.1.1 Formula

$$[\text{Bitrate TSMW}] = [\text{Sample rate in MS/s}] * 2 * (X + Y) * Z * \text{OVal}$$

X	Number of selected channels on RF 1.
Y	Number of selected channels on RF 2.
Z	Number of bits used for the data transfer. Possible values are 8 Bit, 12 Bit, 16 Bit and 20 Bit.
OVAL	Overhead value, approximately 1,1.

1.2.1.2 Example I/Q Streaming

The following parameters are used for an I/Q raw-data streaming measurement with a R&S TSMW via Gigabit LAN interface:

X = 1	On RF1 two sub channels are selected.
Y = 2	On RF2 four sub channels are selected.
Z = 20 Bit	Data transfer format is 20 Bit.
OVAL = 1,1	Overhead value 10%.
Sample rate = 3 [MS/s]	Sample rate 3 [MS/s]

Estimated data transfer rate:

$$\begin{aligned}
 & 3000000 \text{ Samples/s} * 2 * (1 + 2) * 20 \text{ Bit/Samples} * 1,1 \\
 & = 396000000 \text{ Bit/s} \\
 & = 49,5 \text{ MByte/s}
 \end{aligned}$$

This means that 49,5 MB per second has to be transferred from the R&S TSMW to the computer over a Gigabit Ethernet card (and saved on the hard disk) during this I/Q data streaming measurement.

2 Install

2.1 Installation of the R&S TSMW-K1 software

The installation of the software is controlled by a Setup Wizard and requires a full version of MATLAB R2007/R2008. If the installed MATLAB is not a full version it is necessary to install the MATLAB Component Runtime (MCR). A compatible MCR version is available on the CD-ROM.



Please keep in mind that the MCR simply allows the compiled MATLAB applications to be run. Direct access to the corresponding R&S TSMW MATLAB functions is only possible with a full MATLAB installation.

Procedure:

1. Insert the R&S TSMW CD into the CD-ROM drive.
The CD start menu opens automatically.
2. Select the menu item "Applications"-> "TSMW-K1 Installation".
The "Setup Wizard" dialog starts immediately.

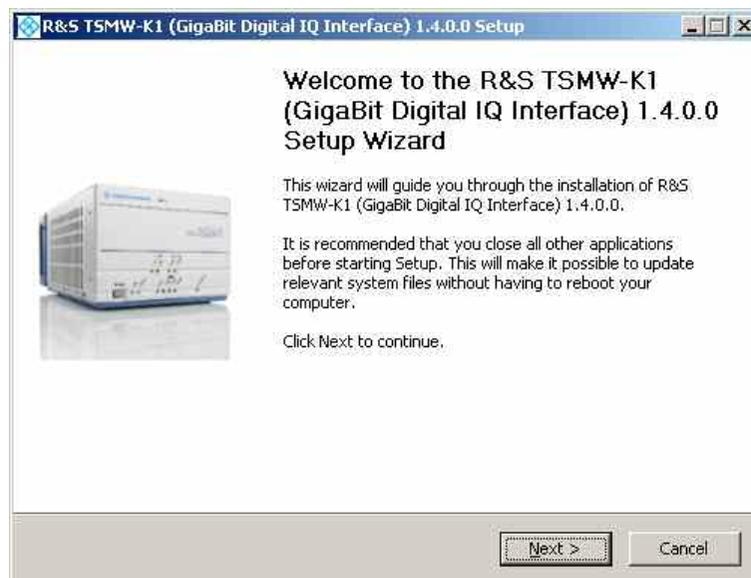


Figure 2-1: Start dialog of the Setup Wizard

3. Click the "Next" button.
The "License Agreement" window opens.
4. To continue the installation you have to agree with the End User License Agreement. Therefore click the "I Agree" button.
The "Choose Components" window opens.

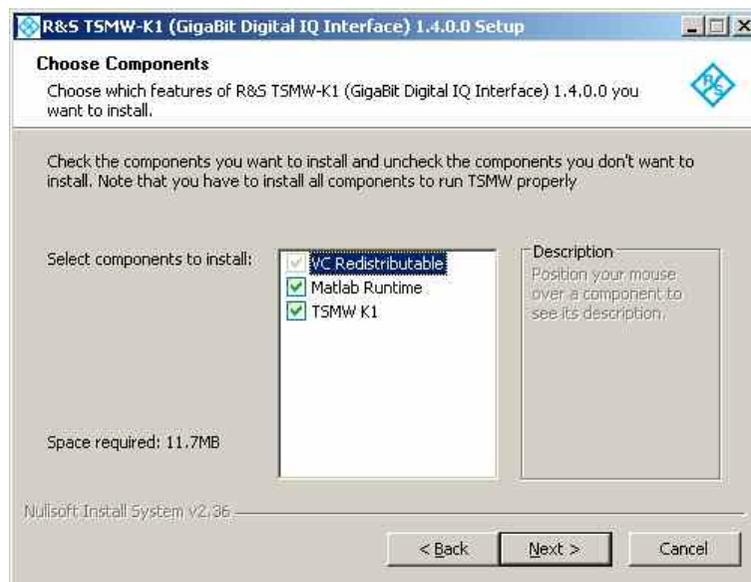


Figure 2-2: Choose Component window

5. Check the desired components to install:
 - a. Check the “TSMW-K1” component to install the R&S TSMW-K1 application. (Mandatory).
 - b. Check the “Matlab Runtime” component if no full version of MATLAB R2007/R2008 is installed (optional).
6. Click the “Next” button to continue.
The “Choose Install Location” window opens.

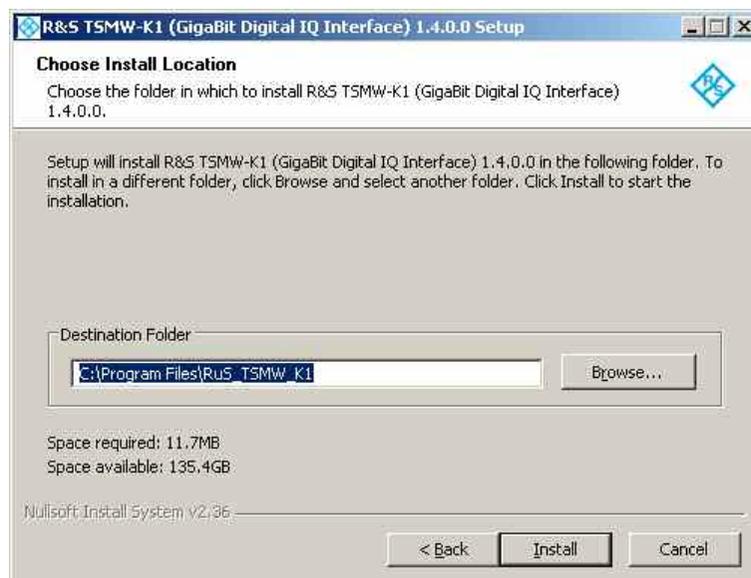


Figure 2-3: Choose Install Location window

7. It is recommended to install all files to the MATLAB toolbox folder. The exact path depends on the installation of MATLAB.
I. e.: C:\Program Files\MATLAB\R2007a\bin\toolbox\RuS_TSMW_K1
Click the “Install” button to start the installation.

Set the path in MATLAB to the R&S TSMW-K1 folder

8. The installation of the R&S TSMW-K1, VC Redistributable starts automatically.
9. During the installation a dialog window opens for the Redistributable Package. Follow the instruction and continue the installation.
10. If the component "Matlab Runtime" was selected the MATLAB Component Runtime (MCR) installer starts automatically.
For detailed information how to install the MCR see [Working with the MCR \(http://www.mathworks.com/access/helpdesk/help/toolbox/compiler/f12-999353.html\)](http://www.mathworks.com/access/helpdesk/help/toolbox/compiler/f12-999353.html).
11. At the end of the installation process an installation complete window appears. Click the "Finish" button.

The installation is finished.

In the menu "Start" -> "Programs" -> "Rohde&Schwarz" exists a new menu folder for the R&S TSMW-K1 software with all tools and information about it.

2.2 Set the path in MATLAB to the R&S TSMW-K1 folder

It is required to add the installation path of the R&S TSMW-K1 to the MATLAB environment in order to have access to the R&S TSMW-K1 MATLAB functions.

Procedure:

1. Select in MATLAB on the menu bar "File" -> "Set Path...".
The "Set Path" window opens.
2. Click the "Add Folder..." button.
The "Browse for Folder" window opens.
3. Add the directory where the MATLAB wrapper functions are installed. I.e.:
`C:\Program Files\MATLAB\R2007a\bin\toolbox\RuS_TSMW_K1\Matlab`
4. Add the directory where the R&S TSMW IQ Interface and related libraries are installed. I.e.:
`C:\Program Files\MATLAB\R2007a\bin\toolbox\RuS_TSMW_K1\lib`
5. Click the "Save" button.

The R&S TSMW-K1 functions are now available in MATLAB.

Set the path in MATLAB to the R&S TSMW-K1 folder

3 Uninstall

The R&S TSMW-K1 application offers a wizard to guide you through the uninstall process of the software.

Procedure:

1. Select
"Start"->"Programs"->"Rohde&Schwarz"->"TSMW-K1 <ver. no.>"->"Uninstall <ver. no.>".
The uninstall wizard window opens.

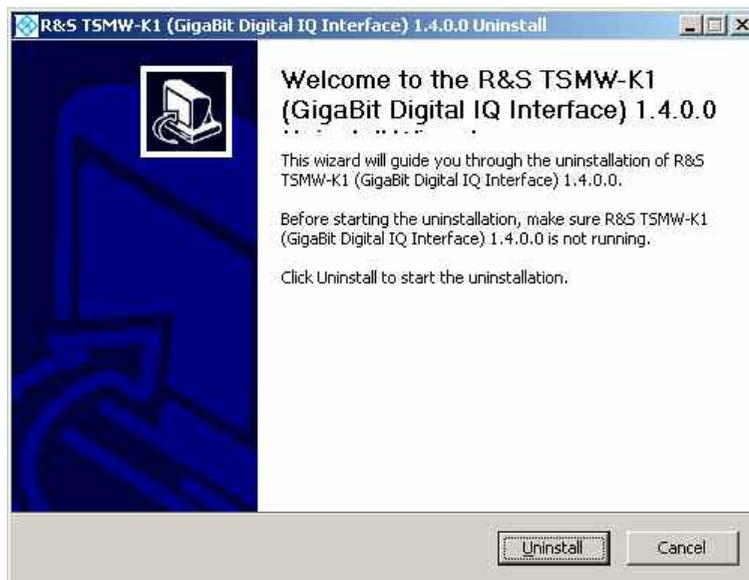


Figure 3-1: Uninstall wizard

2. Follow the introduction of the wizard to uninstall the application.

If no errors occur, the application is successfully uninstalled.

Set the path in MATLAB to the R&S TSMW-K1 folder

4 R&S TSMW signal processing block diagram

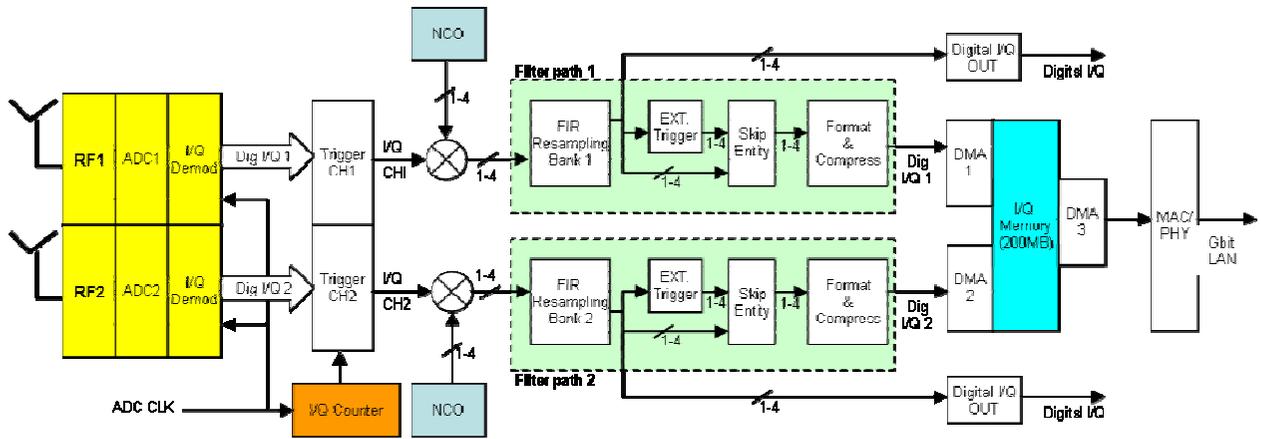


Figure 4-1: R&S TSMW signal processing block diagram

RF1 RF2	Control the frequency setting for frontend 1 and 2. RF1 and RF2 are configurable via the following m-structure fields: MeasCtrl.MeasCtrl.AttStrategy, MeasCtrl.MeasCtrl.Splitter, MeasCtrl.ChannelCtrl1.Frequency, MeasCtrl.ChannelCtrl2.Frequency, MeasCtrl.ChannelCtrl1.Attenuation, MeasCtrl.ChannelCtrl2.Attenuation, MeasCtrl.ChannelCtrl1.Preamplifier, MeasCtrl.ChannelCtrl2.Preamplifier, MeasCtrl.ChannelCtrl1.CalibInput, MeasCtrl.ChannelCtrl2.CalibInput, TSMWOptions.Frontends, TSMWOptions.AMPS_CH1, TSMWOptions.AMPS_CH2, TSMWOptions.Mode For further information about the fields refer to chapter Structure: MeasCtrl on page 30, Structure: TSMWOptionson page 41.
I/Q Demodulation 1 I/Q Demodulation 2	Controls the RF signal processing technique.
I/Q CLK	Native R&S TSMW I/Q sample clock of 21.9444MHz.
I/Q Counter	I/Q Counter counts the I/Q samples starting at the time when a connection to the R&S TSMW has been established.
Trigger CH1 Trigger CH2	Controls the start time of a measurement for channel 1 and 2. The specified I/Q start time is the same for all sub channels of channel 1 or 2, respectively. Measurements on different frontends can be started either independently or simultaneously. When using a resampling-filter, automatically the measurement is started GroupDelay samples before the specified I/Q sample time. GroupDelay is a parameter of the filter specification.

Set the path in MATLAB to the R&S TSMW-K1 folder

	<p>Trigger CH1 and CH2 are configurable via parameter <code>StartTimes</code> in m-function <code>TSMWIQMeasure</code>.</p>
NCO	<p>Numerical Control Oscillator. Controls the frequency shift for the 1 to 4 sub channels for frontend 1 and 2. The following FIR Resampling Bank will perform low pass filtering and fractional resampling for each sub channel using the user defined resampling filter.</p> <p>Note: The filter and resampling factor is the same for all sub channels but can be different for each frontend.</p> <p>NCO is configurable via the following m-structure fields: <code>MeasCtrl.ChannelCtrl1.NoOfChannels,</code> <code>MeasCtrl.ChannelCtrl2.NoOfChannels,</code> <code>MeasCtrl.ChannelCtrl1.FreqShift[],</code> <code>MeasCtrl.ChannelCtrl2.FreqShift[].</code></p> <p>For further information about the fields refer to chapter Structure: <code>MeasCtrl</code> on page 30.</p>
FIR Resampling Bank 1 FIR Resampling Bank 2	<p>Controls the used filter type for frontend 1 and 2.</p> <p>Note: The corresponding filter specification has to be transferred to the R&S TSMW via the m-function <code>TSMWIQSetup</code>.</p> <p>A filter specification has the following m-structure fields: <code>FilterSpec.FilterID,</code> <code>FilterSpec.NoOfCoeffs,</code> <code>FilterSpec.OversplFact,</code> <code>FilterSpec.FilterCorrdB,</code> <code>ResultShiftNumber,</code> <code>FilterSpec.NDown</code></p> <p>Which filter is active on FIR Resampling Bank 1 and 2 are configurable via the m-structure fields: <code>MeasCtrl.MeasCtrl.FilterType,</code> <code>MeasCtrl.MeasCtrl.FilterID.</code></p> <p>For further information about the fields refer to chapter Structure: <code>MeasCtrl</code> on page 30 and Structure: <code>FilterSpec</code> on page 40.</p>
Ext. Trigger 1 Ext. Trigger 2	<p>The external trigger units activate or deactivate the I/Q data measurement when the positive or negative edge of the trigger signal was reached.</p> <p>External Trigger 1 and 2 are configurable via m-function <code>TSMWTrigger</code>.</p> <p>For Trigger 1 and 2 own m-structure fields exist: <code>MeasCtrl.TriggerCtrl.Cmd,</code> <code>MeasCtrl.TriggerCtrl.Mode,</code> <code>MeasCtrl.TriggerCtrl.Falling,</code> <code>MeasCtrl.TriggerCtrl.TriggerLine,</code> <code>MeasCtrl.TriggerCtrl.MeasRequestID,</code> <code>MeasCtrl.TriggerCtrl.Att[2],</code> <code>MeasCtrl.TriggerCtrl.Preampl[2].</code></p> <p>For further information about the fields refer to chapter Structure: <code>MeasCtrl</code> on page 30.</p>
Skip Entity 1 Skip Entity 2	<p>Allow to skip a variable number of samples for each sub channel after the measurement on the corresponding frontend has started before I/Q samples are recorded on this sub channel.</p> <p>Skip Entity 1 and 2 are configurable via the following m-structure fields: <code>MeasCtrl.ChannelCtrl1.ChannelDelay[],</code> <code>MeasCtrl.ChannelCtrl2.ChannelDelay[].</code></p>

Set the path in MATLAB to the R&S TSMW-K1 folder

	For further information about the fields refer to chapter Structure: MeasCtrl on page 30.
Format & Compress	<p>Controls the I/Q data compression. I/Q data are converted into a block floating point format. 64 I/Q samples get always the same exponent. This allows an optimum usage of the I/Q bit width during transmission to the host PC.</p> <p>The I/Q data width (without exponent) used for transmission to the host PC is specified by <code>MeasCtrl.MeasCtrl.DataFormat</code>.</p> <p>For further information about the field refer to chapter Structure: MeasCtrl on page 30.</p>
DMA 1, DMA 2, DMA 3	Direct Memory Access
I/Q Memory	I/Q data memory.
MAC/Phy	Specifies the MAC-Address and the Physical-Address of the R&S TSMW.
Digital I/Q OUT	<p>Allows to stream the digital I/Q data over the optional hardware option R&S Digital I/Q Interface (R&S TSMW-B1). The I/Q data could be streamed either over the LAN or the R&S Digital I/Q Interface.</p> <p>Note:</p> <p>The R&S Digital I/Q Interface is only available for R&S TSMW (variant 03) and higher. Digital I/Q data is provided for all 4 channels on each frontend.</p> <p>Digital I/Q OUT is configurable via the following m-structure fields:</p> <p><code>MeasCtrl.ChannelCtrl1.DigIqOnOff</code> and <code>MeasCtrl.ChannelCtrl2.DigIqOnOff</code></p> <p>For further information about the fields refer to chapter Structure: MeasCtrl on page 30.</p>

Table 4-1: Correlation between R&S TSMW signal processing and m-structures, m-functions.

5 Features

5.1 Measurement Types

The R&S TSMW-K1 option supports different I/Q measurement types:

Types	Description
Single measurement	<p>Performs a single I/Q data measurement.</p> <p>Periodic measurement</p> <p>Optionally, single measurements can be performed as periodic measurements. A periodic measurement performs a single measurement at regular (equal) intervals. When the fields of periodic measurement in m-structure <code>MeasCtrl</code> exist, a single measurement is automatically a periodic measurement. For further information about these fields refer to chapter <code>MeasCtrl</code> values for trigger control on page 37.</p> <p>For further information see the short description of the m-functions Measurement on page 24 and the demo. application on page 45.</p>
Stream measurement	<p>Performs continuous measurements. The I/Q data can be streamed either over the Gigabit LAN Interface or (if available) over the R&S Digital I/Q Interface (R&S TSMW-B1).</p> <p>When streaming over the Gigabit LAN Interface, the measured I/Q data is written into one file until the maximum size is reached. Two modes are distinguish:</p> <p>"Online" streaming</p> <p>The stream data is processed online. Hence, stream data is only buffered in memory (up to the given stream buffer size) until it is acquired via one of the <code>TSMWIQGetStreamXXX</code> functions. As soon as the corresponding data has been acquired, the corresponding memory is automatically released.</p> <p>"Offline" streaming</p> <p>The data is automatically written to a stream data file. After the streaming has finished, the stream data file can be opened and read.</p> <p>For more details refer to the m-functions Seamless streaming of I/Q data in real time on page 25 and see the description of the demo. application on page 45.</p>

5.2 Resource management for concurrent measurement requests

The R&S TSMW-K1 supports a scalable resource management for I/Q data measurement tasks.

The basic idea is that concurrent measurement requests can be assigned different portions of the available receiver time. This is particularly important for applications where independent and concurrent measurement tasks perform measurements using the R&S TSMW.

In order to deal with this problem, it is possible to allocate receiver resources (i.e. receiver time for frontend 1 or 2) and assign measurements to these resources. Having done this, the R&S TSMW internal measurement scheduler considers the current usage of each acquired receiver when deciding whether a measurement shall be performed or not.



Using resource management only makes sense if concurrent measurement requests exist. This means situations occur in which a large number of measurement requests are started that cannot be served at once and therefore have to be delayed. If start times are specified for these measurement requests, it makes sense (if possible) to specify several alternative start times in order to allow the scheduler to start the corresponding measurement at a later time. If a start time cannot be met and no alternative start times are available, the measurement request can not be actioned.

Example:

Three receivers are resourced as follows:

- Resource 1: 30% of frontend 1
- Resource 2: 50% of frontend 2
- Resource 3: 70% of frontend 1 and 50% of frontend 2

This is done with the resource request functions, see chapter Measurement on p. 24.

When a measurement is started, the resource request fields within the MATLAB structure `MeasCtrl` are used to assign each measurement to the right resource by specifying the corresponding resource ID. For details see chapter `MeasCtrl` values for capacity request for RF channels on p. 33).

The R&S TSMW scheduler considers the usage of the specified resource when deciding which of the concurrent measurement requests to serve first.

Scheduling algorithm for measurement tasks

If too many competing measurement tasks request the same resource, the system schedules all waiting measurement tasks automatically. The scheduling algorithm manages the access to the resource according to the priority number (m-structure field `MeasCtrl.MeasCtrl.Priority`). It is possible to give the measurement task a priority value between 0 and 15. 15 is the priority with the highest preferences for resource access. The given priority numbers increase dynamically according to the waiting time (observation time).

If a measurement task does not get at a priority value at creation time, the system will attach the lowest possible priority value to it => 0.

5.3 Trigger

R&S TSMW supports up to two external trigger sources. The R&S TSMW-K1 API allows configuring positive/negative edge of the trigger signal for measurements.

5.4 GPS

An integrated SuperSense GPS receiver with 16 channels and a refresh rate of 4 Hz also allows the R&S TSMW to be used in areas with weak GPS signals. The GPS receiver sends data in NMEA and UBX format. Therefore it is possible to send control commands for NMEA and UBX to the R&S. This is possible via the MATLAB and C++ functions of R&S TSMW-K1 API.

6 Function Overview

Each function returns an error code: 0 if successful. Otherwise a number is returned which defines the error code. To get the corresponding error text calls the function `TSMWGetLastError` or for a streaming measurement `TSMWgetStreamError`. The functions return the error information of the occurred error and reset the error code variable again to zero.



Variable ErrorCode

Please keep in mind that only one `ErrorCode` variable is used for all functions. Therefore only the current returned error code is available.

6.1 General

The following functions allow to control and access the I/Q interface of the R&S TSMW. Furthermore trigger settings and error handling is managed.

Function MATLAB (*.m file)	Function C++ (header files)	Description
<code>TSMWInitInterface</code>	<code>TSMWInitInterface_c</code>	Load and initialize the R&S TSMW IQ interface library <code>TSMWInterface.dll</code> .
<code>TSMWReleaseInterface</code>	<code>TSMWReleaseInterface_c</code>	Unload the R&S TSMW MATLAB interface library <code>TSMWInterface.dll</code> and disconnect from R&S TSMW(s).
<code>TSMWConnect</code>	<code>TSMWConnect_c</code>	Establish a connection to the R&S TSMW with the given IP address and given options.
<code>TSMWGetLastError</code>	<code>TSMWGetLastError_c</code>	Return the error message of the occurred error.
<code>TSMWShutdown</code>	<code>TSMWShutdown_c</code>	Shutdown the R&S TSMW.
<code>TSMWGetIQTime</code>	<code>TSMWGetIQTime_c</code>	Get current I/Q time of given R&S TSMW.
<code>TSMWTrigger</code>	<code>TSMWTrigger_c</code>	Set/get the R&S TSMW trigger output/ input.
<code>TSMWGetVersion</code>	<code>TSMWGetVersion_c</code>	Get R&S TSMW I/Q interface version.
<code>TSMWGetVersionText</code>	<code>TSMWGetVersionText_c</code>	Get R&S TSMW I/Q interface version as text.

Table 6-1: Overview of the MATLAB and C++ I/Q interface functions.

6.2 Measurement

These functions configure, setup and control the measurements. Furthermore, it is possible to manage the scalable resources of the two RF to the complete measurement tasks and organize the transfer of the I/Q data to the pc for further handling.

6.2.1 Functions

Function MATLAB (*.m file)	Function C++ (header files)	Description
TSMWIQSetup	TSMWIQSetup_c	Send a filter specification to the R&S TSMW.
TSMWIQMeasure	TSMWIQMeasure_c	Start a measurement with the given parameters.
	TSMWIQMeasureExt_c	Start an extended measurement.
	TSMWIQMeasureTrig_c	Start a triggered measurement.
-	TSMWIQTriggerCtrl_c	Change parameters of a triggered measurement.
TSMWResourceRequest	TSMWIQResourceRequest_c	Request receiver resources.
TSMWIQPeriodCtrl	TSMWIQPeriodCtrl_c	Change parameters of a running periodic measurement request.
TSMWIQDataAvailable	TSMWIQDataAvailable_c	Check how many measurement results from R&S TSMW are available.
TSMWGetIQResultParam	TSMWGetIQResultParam_c	Get measurement result parameters.
TSMWIQGetData	TSMWIQGetDataInt16_c	Retrieves I/Q data as complex double array.
TSMWIQGetDataInt16	TSMWIQGetDataInt16_c	Return the measurement data in 16 bit integer array format.
TSMWIQGetDataInt32	TSMWIQGetDataInt32_c	Return the measurement data in 32 bit integer array format.
TSMWIQGetDataSingle	TSMWIQGetDataSingle_c	Return the measurement data in single precision floating point array format.
TSMWIQGetDataDouble	TSMWIQGetDataDouble_c	Return the measurement data in double precision floating point array format.
TSMWIQGetDataSingleIlv	TSMWIQGetDataSingleIlv_c	Return the measurement data in interleaved single precision floating point array format.
TSMWIQGetDataDoubleIlv	TSMWIQGetDataDoubleIlv_c	Return the measurement data in interleaved double precision floating point array format.
TSMWGetFIRParam	-	Calculate FIR parameters for a given downsampling factor.

Table 6-2: Overview of the MATLAB and C++ I/Q interface functions.

6.2.2 Templates

Function MATLAB (*.m file)	Function C++ (header files)	Description
MeasCtrlTemplate	-	Create a template for the MeasCtrl structure for double channel (MIMO) measurements with default values.
MeasCtrlTemplate_RF1	-	Create a template for the MeasCtrl structure for RF channel 1 measurements.
MeasCtrlTemplate_RF2	-	Create a template for the MeasCtrl structure for RF channel 2 measurements.

Table 6-3: Overview of the MATLAB and C++ I/Q interface functions.

6.3 Seamless streaming of I/Q data in real time

R&S TSMW makes it possible to perform technology-independent channel measurements. These measurements can be used to simulate realistic fading scenarios in a laboratory environment. To do this, I/Q data recorded by the R&S TSMW can, for example, be replayed directly on a signal generator from Rohde & Schwarz. This data, which has been recorded under real conditions, can therefore be replayed again and again in the laboratory environment. The seamless streaming of I/Q data can be carried out with a maximum bandwidth of 20 MHz. The duration of an I/Q recording is determined solely by the size of the hard disk.

The measured I/Q data is transmitted in real time. Both "online" and "offline" streaming is possible. Streaming can in general be started with the same parameters as normal (block) measurements. However, bandwidth (data rate) limitations have to be considered for successful streaming without lost data blocks.

If the measured I/Q data is streamed only over the R&S TSMW-B1 interface, it is recommended to use the "online" streaming mode.

- For measured I/Q data from a stream measurement it is recommended to transmit the I/Q data over the R&S TSMW-B1 interface. The R&S TSMW-B1 interface allows to transmit more I/Q data per second than it is possible for a Gigabit LAN connection.
- Otherwise measured I/Q data from a periodic or a single measurements is usually transmitted over the Gigabit LAN interface, since the bandwidth of an ethernet connection is mostly sufficient for the transmission of I/Q data to the destination.



Parallel use of the Gigabit LAN interface and R&S TSMW-B1 interface for stream measurements

It is possible to use the frontend 1 to send streaming data over the R&S TSMW-B1 option and in parallel to use the frontend 2 to send streaming data over the Gigabit LAN interface.

6.3.1 Functions

Function MATLAB (*.m file)	Function C++ (header files)	Description
TSMWIQStream	TSMWIQStream_c	Start streaming measurement for a given stream ID.
TSMWTGEnable	TSMWTGEnable_c	Enable the tracking generator (test) output.
TSMWSetDAC	TSMWSetDAC_c	Set the reference oscillator DAC value.
TSMWIQStreamStatus	TSMWIQStreamStatus_c	Return the I/Q streaming status.
TSMWGetStreamError	TSMWGetStreamError_c	Return the error message of the last error occurring on the streaming measurement. This function should be called always after function TSMWIQStreamStatus, if an error was detected.
TSMWIQStopStreaming	TSMWIQStopStreaming_c	Stop streaming for a given stream ID.
TSMWIQOpenStreamFile	TSMWIQOpenStreamFile_c	Open the streaming file.
TSMWIQCloseStreamFile	TSMWIQCloseStreamFile_c	Close the streaming file.
TSMWIQGetStreamSingle	TSMWIQGetStreamSingle_c	Return the streaming data block in single format.
TSMWIQGetStreamDouble	TSMWIQGetStreamDouble_c	Return the streaming data block in double format.
TSMWIQGetStreamSingleIv	TSMWIQGetStreamSingleIv_c	Return the streaming data block in interleave single format.
TSMWIQGetStreamDoubleIv	TSMWIQGetStreamDoubleIv_c	Return the streaming data block in interleave double format.

Table 6-4: Overview of the MATLAB and C++ I/Q interface functions.

6.3.2 Templates

Function MATLAB (* .m file)	Function C++ (header files)	Description
MeasCtrlTemplate_RF1Stream	-	Create a template for the MeasCtrl structure for RF channel 1 streaming measurement.
MeasCtrlTemplate_RF2Stream	-	Create a template for the MeasCtrl structure for RF channel 2 streaming measurement.

Table 6-5: Overview of the MATLAB and C++ I/Q interface functions.

6.4 GPS

These functions are used to configure and receive GPS data from R&S TSMW. NMEA and UBX messages are supported.

Function MATLAB (* .m file)	Function C++ (header files)	Description
TSMWGPSSync	TSMWGPSSync_c	Enable or disable the GPS synchronization.
	TSMWGPSClearBuffer_c	Clear the GPS NMEA buffer.
TSMWGetIQTime	TSMWGetIQTime_c	Get current IQ time of given R&S TSMW.
TSMWGPSEnable	TSMWGPSEnable_c	Enable or disable GPS data subscription from R&S TSMW.
TSMWGPSGetNMEAline	TSMWGPSGetNMEAline_c	Get GPS NMEA text line.
TSMWGPSSendNMEACmd	TSMWGPSSendNMEACmd_c	Send NMEA command string to GPS.
TSMWExtRefEnable	TSMWExtRefEnable_c	Enable external reference.
TSMWGPSSendUBXCmd	TSMWGPSSendUBXCmd_c	Send UBX message to GPS.
-	TSMWGPSGetUBXmsg_c	Get next available GPS UBX message
TSMWGPSGetServerStatus	TSMWGPSGetServerStatus_c	Current status of the GPS server.
TSMWGPSSendUBX_CFG_INF		Send UBX command CFG_INF to TSMW.
TSMWGPSSendUBX_CFG_MSG		Send UBX command CFG_MSG to TSMW.
TSMWGPSSendUBX_CFG_PRT _AllIn_NMEAOut		Send UBX command CFG_PRT to TSMW.
TSMWGPSSendUBX_CFG_RATE		Send UBX command CFG_RATE to the TSMW

Function MATLAB (* .m file)	Function C++ (header files)	Description
TSMWGPSStartTcpServer	TSMWGPSStartTcpServer_c	Start server forwarding NMEA messages
TSMWGPSStopServer	TSMWGPSStopServer_c	Stop server forwarding NMEA messages

Table 6-6: Overview of the MATLAB and C++ I/Q interface functions.

6.5 Digital IQ Interface

These functions are used to configure and receive messages from the R&S TSMW.

Function MATLAB (* .m file)	Function C++ (header files)	Description
TSMWINFOIFEnable	TSMWINFOIFEnable_c	Function to subscribe/unsubscribe to the Digital I/Q Interface status messages.
TSMWINFOIFGetStatus	TSMWINFOIFGetStatus_c	Function to read the latest status message from the Digital I/Q Interface.

Table 6-7: Overview of the MATLAB and C++ I/Q interface functions.

7 Structures

For easier working with the R&S TSMW MATLAB I/Q interface API, several parameter structures are defined. The following sub chapters explain the structures with their fields. Additionally the corresponding C++ structure is mention. Most C++ parameters correspond to the available MATLAB fields.



Example of MATLAB structures

Examples are available in the example folder

```
<TSMW-K1 installation directory>\Examples\Matlab.
```

7.1 Structure: MeasCtrl

Description:

The structure contains measurement control values. It is divisible into the following parts:

- **General measurement control field**
Defines general measurement control values.
- **Stream control field**
Defines the streaming options.
- **Resource request field**
Defines the requested bandwidth of the RF.
- **Schedule control field**
Defines the schedule options.
- **Period control field**
Defines the settings for periodic measurement.
- **Trigger control field**
Defines the trigger options.
- **RF channel 1 control field**
Defines specific RF channel 1 control values.
- **RF channel 2 control field**
Defines specific RF channel 2 control values.

For easier handling in MATLAB these four components are put together in one structure `MeasCtrl`.

In C++ these measurement control structures are defined as structure types (`typedef struct`):

- **General measurement values**
`struct TSMW_IQIF_MEAS_CTRL`
- **Stream control**
`struct TSMW_IQIF_STREAM_CTRL`
- **Independent RF channel control**
`struct TSMW_IQIF_CH_CTRL`
- **Configure resource request**
`struct TSMW_IQIF_RESOURCE_PARAM`
- **Schedule control**
`struct TSMW_IQIF_SCHEDULE_CTRL`
- **Periodic measurement control**
`struct TSMW_IQIF_PERIOD_CTRL`
- **Trigger control**
`struct TSMW_IQIF_TRIG_CTRL`

7.1.1 MeasCtrl values for general measurement control

Description:

These fields allow configuring the general measurement control settings on the R&S TSMW. For faultless use the values has to be set and submit over the Digital I/Q interface to the R&S TSMW(s).

Field(s):

<code>MeasCtrl.NoOfSamples</code>	Specifies the number of I/Q samples to measure.
<code>MeasCtrl.FilterType</code>	Specifies the used filter type: 0: Use predefined filters 1: Use user defined filters. (They have to be sent to the R&S TSMW before).
<code>MeasCtrl.FilterID</code>	If a predefined filter is used this field define the filter number.
<code>MeasCtrl.DataFormat</code>	Specifies I/Q data compression format for block wise compression. 0: 2x 8 Bit per complex sample 1: 2x 12 Bit per complex sample 2: 2x 16 Bit per complex sample 3: 2x 20 Bit per complex sample
<code>MeasCtrl.AttStrategy</code>	Specifies the attenuation strategy. 0: Manual attenuation. 1: Auto attenuation. If activated and the signal strength is to strong: 1. The preamplifier is deactivated. 2. If the signal strength is still to strong, the attenuation will be increased in 5dB steps till the signal strength is ok. Note: After each reconfiguration step the I/Q measurement is restarted. Note: Only available for block measurements.
<code>MeasCtrl.Splitter</code>	Specifies the RF channel 1 splitter. Splitting the signal after preselector to both frontends. 0: Disable splitter 1: Enable splitter
<code>MeasCtrl.Priority</code>	Specifies the priority of the measurement. Valid priority levels are 0..15.

C++ structure:

```
typedef struct TSMW_IQIF_MEAS_CTRL
{
    unsigned long NoOfSamples; unsigned char FilterType;
    unsigned char FilterID; unsigned char DataFormat;
    unsigned char AttStrategy; unsigned char Splitter;
    unsigned short Priority;
} TSMW_IQIF_MEAS_CTRL_t;
```

7.1.2 MeasCtrl values for streaming control

Description:

These fields configure the settings of the streaming file.

Field(s):

<code>StreamCtrl.StreamID</code>	Specifies the stream ID which shall be used to control this stream. Value range: 0..15
<code>StreamCtrl.StreamBufferSize</code>	Specifies the buffer size for streaming in MBs. At least 200 MB is recommended.
<code>StreamCtrl.MaxStreamSize</code>	Specifies the maximum streaming size in MB.

Note:

The stream data is automatically splitted into several files such that a maximum file size of 2GB is not exceeded.

C++ structure:

```
typedef struct TSMW_IQIF_STREAM_CTRL
{
    unsigned long StreamID; unsigned long StreamBufferSize;
    unsigned long MaxStreamSize;
} TSMW_IQIF_STREAM_CTRL_t;
```

7.1.3 MeasCtrl values for capacity request for RF channels

Description:

These fields contains information about the requested receiver capacity and the desired observation time window.

Field(s):

Frontends	Specifies which frontend to use. 1: Frontend 1 2: Frontend 2 3: Both frontends
Capacity	Requested receiver capacity in % . The actual percentage a receiver resource will get is given by $\frac{\text{(Capacity of the receiver)}}{\text{(Sum of capacity of all resources)}}$ The sum capacity can be larger than 100%.
ObservationTime	Observation time window in ms. Only measurements that were taken in the last <ObservationTime> ms will count for the current resource usage

C++ structure:

```
typedef struct TSMW_IQIF_RESOURCE_PARAM
{
    unsigned char Frontends; unsigned char Capacity;
    unsigned long ObservationTime;
}TSMW_IQIF_RESOURCE_PARAM_t;
```

7.1.4 MeasCtrl values for schedule control

Description:

The structure describes the schedule option for the scalable bandwidth management for the receivers. It has to be specified if the measurement request shall use a specific receiver resource.



Schedule structure does not exist

If no structure exist no receiver resource class will be used. Measurements that do not use a specific receiver resource class will not be considered for receiver capacity limitation!

Field(s):

SchedCtrl.ResourceID	Resource ID of the schedule task.
SchedCtrl.SchedulerTimeConst	Scheduler time constant for non-periodic measurements in ms. The priority of a non-periodic measurement is given by $\frac{\langle \text{Priority in meas.req.} \rangle + \langle \text{Waiting time in ms} \rangle}{\langle \text{SchedulerTimeConst} \rangle}$ This means that the greater the constant is the longer is the waiting time of the standard measurement request.

C++ structure:

```
typedef struct TSMW_IQIF_SCHEDULE_CTRL
{
    unsigned long ResourceID; unsigned long SchedulerTimeConst;
}TSMW_IQIF_SCHEDULE_CTRL_t;
```

7.1.5 MeasCtrl values for periodic measurement control

Description:

The structure describes the control structure for starting periodic measurement requests. It has to be specified if the measurement request has to be a periodic measurement request. Otherwise it is a standard measurement.

Field(s):

Cmd	<p>Command:</p> <p>0: When used within m-function TSMWIQMeasure: Start periodical measurement request.</p> <p>Otherwise: Add credits to periodical measurement request without changing parameters.</p> <p>1: Clear all credits of periodical measurement request. De-activate the measurement request.</p> <p>2: Cancel periodical measurement request.</p> <p>3: Change parameters.</p> <p>4: Adjust time slip.</p> <p>5: Change attenuator and preamble settings.</p>
MeasRequestID	<p>Only used if Cmd == 3 (change parameters of the measurement request).</p> <p>Specifies the measurement request ID of periodical measurement request.</p>
IQPeriod	<p>Time period in I/Q samples. The minimum time period is 1ms -> 22000 I/Q samples.</p>
IQMinDistance	<p>Only used if no start time is given.</p> <p>Minimum distance between consecutive measurements in I/Q samples.</p>
GridTimeus	<p>Only used if start time is given.</p> <p>Time grid in microseconds on which measurements shall be performed. In that case, measurements are only performed on a time grid given by $StartTime + n * GridTime$.</p> <p>Note: Internally GridTimeus is converted from microseconds to I/Q-samples.</p> <p>The StartTime is always given in I/Q-samples.</p>
IQTimeSlip	<p>Only used if Cmd == 4 (Adjust time slip).</p> <p>Defines the grid at which a measurement will be started is adjusted by the IQTimeSlip. Time slip is only available for periodic measurement requests with a start time.</p> <p>Note: The time slip is given in I/Q samples before resampling.</p>

NoOfMeasurements	<p>Only used if Cmd == 0</p> <p>Specifies the initial number of measurements to deliver until new "measurement credits".</p>
TransmitWindow	<p>Specifies the transmit window. A transmit window defines the number of measurements that are performed until the first measurements are transmitted to the host. This prohibits the R&S TSMW of making more measurements than can be transmitted to the PC cause by i. e. limited ethernet capacity.</p> <p>Value range 2..16.</p>
Att[2]	<p>Only used if Cmd == 5 (Change attenuator and preample settings)</p> <p>Contain the new attenuator setting for each channel.</p>
Preamp[2]	<p>Only used if Cmd == 5 (Change attenuator and preamble settings)</p> <p>Contain the new preample setting for each channel.</p>

C++ structure:

```
typedef struct TSMW_IQIF_PERIOD_CTRL
{
    unsigned long Cmd; unsigned long MeasRequestID;
    unsigned __int64 IQPeriod; unsigned __int64 IQMinDistance;
    unsigned __int64 GridTimeus; short IQTimeSlip;
    unsigned short NoOfMeasurements; unsigned char TransmitWindow;
    unsigned char Att[2]; unsigned char Preamp[2];
}TSMW_IQIF_PERIOD_CTRL_t;
```

7.1.6 MeasCtrl values for trigger control

Description:

The structure describes the control structure for trigger configuration.

Field(s):

<code>TriggerCtrl.Cmd</code>	<p>Command:</p> <ul style="list-style-type: none"> 0: When used within m-function TSMWIQMeasure: Start triggering for a measurement. 1: Stop triggered measurement. 2: Change parameters.
<code>TriggerCtrl.Mode</code>	Specifies the trigger mode. At the moment only 0 is available.
<code>TriggerCtrl.Falling</code>	<p>Specifies the trigger edge:</p> <ul style="list-style-type: none"> 0: Rising 1: Falling
<code>TriggerCtrl.TriggerLine</code>	<ul style="list-style-type: none"> 1: Use trigger input 1 2: Use trigger input 2 3: Use trigger on both inputs.
<code>TriggerCtrl.MeasRequestID</code>	<p>Only used if <code>Cmd == 2</code> (change parameters)</p> <p>Specifies the measurement request ID of periodical measurement request</p>
<code>TriggerCtrl.Att[2]</code>	<p>Only used if <code>Cmd == 2</code> (change parameters)</p> <p>Change the current attenuator setting for each channel.</p>
<code>TriggerCtrl.Preampl[2]</code>	<p>Only used if <code>Cmd == 2</code> (change parameters)</p> <p>Change preample setting for each channel.</p>

C++ structure:

```
typedef struct TSMW_IQIF_TRIG_CTRL
{
    unsigned long Cmd; unsigned char Mode; unsigned char Falling;
    unsigned char TriggerLine; unsigned long MeasRequestID;
    unsigned char Att[2]; unsigned char Preampl[2];
}TSMW_IQIF_TRIG_CTRL_t;
```

7.1.7 MeasCtrl values for RF channel 1 and RF channel 2

Description:

Each RF channel has its own fields. These fields contain measurement control values for the corresponding RF channel.

Field(s):



The following explanation shows only the values for the RF channel 1 control fields. They correspond to the RF channel 2. Therefore replace the prefix "ChannelCtrl1"(RF channel 1) to "ChannelCtrl2" (RF channel 2).

<code>ChannelCtrl1.Frequency</code>	Center frequency (Hz)
<code>ChannelCtrl1.UseOtherFrontend</code>	Reserved for future use, has to be set to zero
<code>ChannelCtrl1.NoOfChannels</code>	Specifies the number of channels that shall be used (1..4). The received bandwidth of the selected frontend is split up into the corresponding number of channels. I. e.: If 3 channels are used, the downsampling factor has to be at least 3.
<code>ChannelCtrl1.Attenuation</code>	Specifies the attenuation to use if the parameter is set. In auto attenuation mode this is the initial (suggested) attenuation to use for the measurement. Note: Only the values 0,5,10 and 15 dB are calibrated.
<code>ChannelCtrl1.Preamplifier</code>	Defines whether the preamp. shall be enabled or not. In auto attenuation mode, this is the initial (suggested) setting of the preamp. 0: Disable 1: Enable
<code>ChannelCtrl1.CalibInput</code>	Defines whether the calibration input shall fit to the used frontend. 0: Disable 1: Enable Note: The calibration amplifier has to be enabled, too.
<code>ChannelCtrl1.FreqShift[]</code>	Defines the frequency shift of each frequency shift from center frequency for each channel
<code>ChannelCtrl1.ChannelDelay[]</code>	Specifies a separate delay in I/Q samples after filtering and downsampling for each channel.
<code>ChannelCtrl1.BlockSize</code>	Reserved for future use, has to be zero.
<code>ChannelCtrl1.BlockSkip</code>	Reserved for future use, has to be zero.

ChannelCtrl1.DigIqOnOff

Configuration for digital I/Q output. The measured digital I/Q data is streamed over the R&S Digital I/Q Interface.

0: Digital I/Q output disabled

1: Data from Front End mapped on Digital I/Q output

C++ structure:

```
typedef struct TSMW_IQIF_CH_CTRL
{
    unsigned __int64 Frequency; unsigned char UseOtherFrontend;
    unsigned char NoOfChannels; char Attenuation;
    unsigned char Preamp; unsigned char CalibInput;
    double FreqShift[4]; long ChannelDelay[4];
    long BlockSize; long BlockSkip; unsigned char DigIqOnOff;
} TSMW_IQIF_CH_CTRL_t;
```

7.2 Structure: FilterSpec

Description:

The structure describes the filter parameter structure.



The down sampling factor has to be greater than the no. of sub channels. The maximum down sampling factor is 100.

Therefore following requirement has to be fulfilled:

$$(\text{No. of Sub Channels}) < (\text{Downsampling Factor}) < \left(\frac{\text{Sampling Rate}}{\text{No. of Sub Channels}} \right)$$

Field(s):

FilterID	Specifies the filter number.
NoOfCoeffs	Specifies the number of filter coefficients.
OversplFact	Specifies the filter coefficient oversampling factor.
FilterCorrdB	Specifies the negative average filter gain in the pass band in dB. The result will be corrected by this value. Note: This value is the remaining filter gain after considering a factor of $2^{(-\text{ResultShiftNumber})}$.
ResultShiftNumber	Specifies the number of bits the result is shifted. Compensates for the filter gain.
GroupDelay	Specifies the group delay of the current filter in taps. A measurement will automatically started "GroupDelay" samples before the given start time (if a start time was given). Default value is 0.
NDown	Specifies the fractional down sampling factor. Note: Since the maximum number of filter coefficients and the oversampling factor depends on the downsampling factor, it is essential that these values match, for correct filtering/downsampling. Note: The down sampling factor has to be greater than the no. of sub channels. The maximum down sampling factor is 100.

C++ structure:

```
typedef struct TSMW_IQIF_FILTER_PARAM
{
    unsigned short FilterID; unsigned short NoOfCoeffs;
    unsigned char OvsplFact; double FilterCorrdB;
    unsigned char ResultShiftNumber; unsigned short GroupDelay;
    double Ndown;
} TSMW_IQIF_FILTER_PARAM_t;
```

7.3 Structure: TSMWOptions

Description:

The structure describes the Digital I/Q interface options for the R&S TSMW.

Field(s):

Frontends	Specifies which frontend to enable 1: Enable frontend 1 2: Enable frontend 2 3: Enable both frontends
AMPS_CH1	Enable the frequency band for RF channel 1. Set to (2 ³² -1) for all bands (amplifiers).
AMPS_CH2	Enable the frequency band for RF channel 2. Set to (2 ³² -1) for all bands (amplifiers).
Mode	At the moment only the value zero is supported.

C++ structure:

```
typedef struct TSMW_IQIF_MODE
{
    unsigned char Frontends; unsigned long AMPS_CH1;
    unsigned long AMPS_CH2; unsigned char Mode;
}TSMW_IQIF_MODE_t;
```

7.4 Structure: TSMWIQResult

Description:

The structure describes the IQ result parameter structure.

Field(s):

MeasRequestID	Specifies the measurement request ID.
Reserved	Unused
StartTimeIQ	Specifies the I/Q-Counter value at which measurement was started. Note: Each individual channel will have a different start time, according to the specified ChannelDelay parameter in the measurement request structure.
StartTimeHost	Specifies the corresponding host time.
Offset	Offset in I/Q-Samples from measurement start (only for streaming meas.)
NoOfSamples	Number of samples that were copied for each sub channel
Reserved2	Unused
Fsample	Specifies the sampling rate (sample/s) that was actually used. Mostly the used sampling rate for a specific filter differs from the requested sample rate. This is caused by rounding errors. I. e.: For a specific filter the requested sample rate is 10Ms/s. But in this case the actually used sample rate for the filter is 10.00000772597512 MS/s. This information is available via field Fsample.
Attenuation.1	Specifies the used attenuation in dB for frontend 1.
Attenuation.2	Specifies the used attenuation in dB for frontend 2.
Preamp.1	Specifies for frontend 1: 0: Preamp was off 1: Preamp was on
Preamp.2	Specifies for frontend 2: 0: Preamp was off 1: Preamp was on

C++ structure:

```
typedef struct TSMW_IQ_RESULT
{
    unsigned long MeasRequestID; unsigned long Reserved;
    unsigned long long StartTimeIQ; unsigned long long StartTimeHost;
    unsigned long long Offset; unsigned long NoOfSamples;
    unsigned long Reserved2; unsigned long Attenuation[2];
    unsigned long Preamp[2];
} TSMW_IQ_RESULT_t;
```

7.5 Structure: StreamStatus

Description:

The structure describes the current status of the streaming measurement.

Field(s):

Status	Streaming status 0: Not activated 1: Running 2: Finished 3: Stopped (because of an error) 4: Stopped because maximum file size reached
BlockSize	Number of samples per streaming block.
NoOfBlocks	Number of received blocks.
NoOfSkippedBlocks	Number of skipped blocks.

C++ structure:

```
typedef struct TSMW_IQIF_STREAM_STATUS
{
    unsigned char Status; unsigned long BlockSize;
    unsigned long NoOfBlocks; unsigned long NoOfSkippedBlocks;
} TSMW_IQIF_STREAM_STATUS_t;
```

7.6 Structure: StreamInfo

Description:

The structure shows information about date and time about a streaming measurement file.

Field(s):

Year	The year when the stream was started.
Month	The month when the stream was started.
Date	The date of the day when the stream was started.
Hour	The hour when the stream was started.
Minute	The minute when the stream was started.
Second	The second when the stream was started.
Description[256]	In this field the user can write a short description about the streaming file. This information is saved with the streaming file.
NoOfFiles	Number of files for this streaming measurement. Every time 2GB is reached a new files with increased index number will be created (i. e. myfile_000, myfile_001, etc.)
StreamSize	Total stream size in MB.
BlockSize	Number of samples per streaming block.
NoOfBlocks	Number of received blocks.
NoOfSkippedBlocks	Number of skipped blocks.

C++ structure:

```
typedef struct TSMW_IQIF_STREAM_INFO
{
    unsigned long Month; unsigned long Date; unsigned long Hour;
    unsigned long Minute; unsigned long Second;
    char Description[256]; unsigned __int64 NoOfFiles;
    unsigned long StreamSize; unsigned long BlockSize;
    unsigned long NoOfBlocks; unsigned long NoOfSkippedBlocks;
} TSMW_IQIF_STREAM_INFO_t;
```

8 R&S TSMW I/Q Interface Application



Use R&S TSMW MATLAB IQ Interface functions for user specific MATLAB scripts

The "R&S TSMW IQ Interface" application uses a couple of functions which could be adapted for user specific MATLAB scripts. Therefore refer to the following chapters.

The "TSMW MATLAB I/Q Interface" application is used as GUI interface for an easy handling the measurement control. Furthermore it can be used as reference example how using the R&S TSMW-K1 API. On the basis of this application the functionality of the R&S TSMW-K1 API is explained.

Likewise, the application may be utilized as the basis for customized applications in order to quickly obtain the measurement results needed.

The application makes it possible to define the center frequency, the sampling rate and the measurement filter. In this context, the measurement filter is used primarily to define the bandwidth to be measured. Individual measurements or continuous measurements may be carried out. The user can also define which of the two R&S TSMW receivers ("frontends") is to be used. For professional applications, it is furthermore possible to set the attenuation values, the activation of the preamplifier, and the data format of the digital I/Q data.



Source code of the R&S TSMW MATLAB IQ Interface application

The source code of the `TSMWInterfaceDemo.m` is available on the installed folder.

8.1 Main Window "TSMW Matlab IQ Interface"

This section covers the main window of the application which consists of different parts and includes: connection to the R&S TSMW, configuration of the measurement algorithm and displaying the current status of the Digital I/Q Interface.

Every button is mapped to one or more R&S TSMW-K1 MATLAB functions and shows an example implementation.

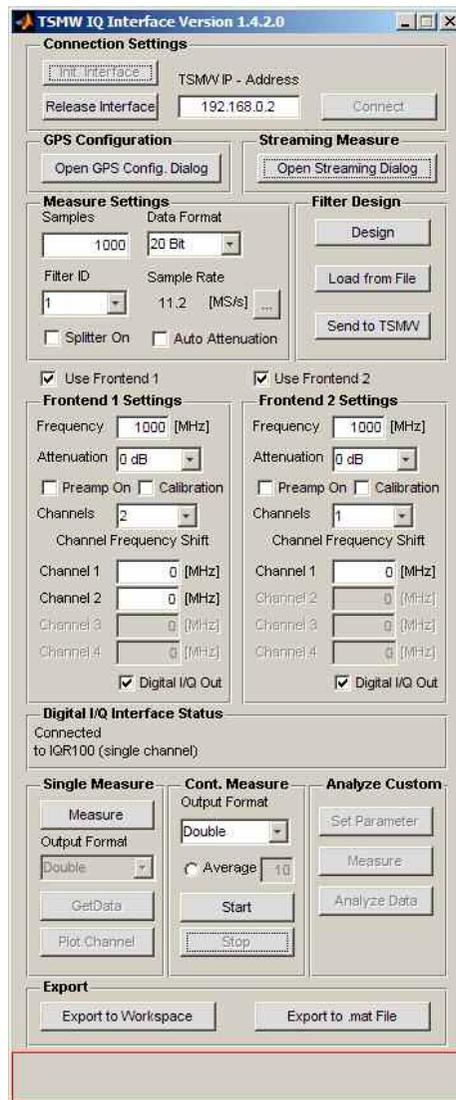


Figure 8-1: Main window of TSMW IQ Interface demo application

8.1.1 GUI – connection and measurement setting

The following tables list the functionality and the corresponding MATLAB functions of the application.

Connection Settings

Name	Description
Init. Interface	Load the R&S TSMW IQ interface library <code>TSMWInterface.dll</code> . M-function: <code>TSMWInitInterface</code>
ReleaseInterface	Unload the external R&S TSMW IQ interface library and disconnect from TSMW(s). M-function: <code>TSMWReleaseInterface</code>

Name	Description
TSMW IP-Address	Enter the R&S TSMW IP address. Default value is 192.168.0.2. M-parameter: IPAddress
Connect	Establish a connection to the R&S TSMW with the given IP address. M-function: TSMWConnect

GPS Configuration

Name	Description
Open GPS Config. Dialog	Open a dialog window which controls the GPS server settings. For details see GPS Server Configuration Dialog on page 56.

Streaming Measure

Name	Description
Open Streaming Dialog	Open a dialog window which controls the streaming settings for I/Q data streaming measurements. For details see Streaming Dialog On page 52,

Measure Settings

Correspond to MeasCtrl.MeasCtrl. For details see Structure: MeasCtrl on page 30.

Name	Description
Samples	Specify the number of I/Q samples to measure. Field: MeasCtrl.MeasCtrl.NoOfSamples
Data Format	Specify I/Q data compression format for block wise compression. Field: MeasCtrl.MeasCtrl.DataFormat Note: Reduce the data transfer rate To reduce the amount of data accumulated, the R&S TSMW-K1 interface offers to set the bit resolutions individual to 20 Bit, 16 Bit, 12 Bit or 8 Bit. Thus the user can reduce the transmission rate on the LAN interface.
Filter ID	Select a filter to use. 0 : Select to use a predefined filter. ≥ 1: Select to use a user defined filter. Note: A user defined filter has to be sent to the R&S TSMW before it is available in the drop down list. Fields: MeasCtrl.MeasCtrl.FilterID, MeasCtrl.MeasCtrl.FilterType
...	Display the transfer function of the selected filter.
Splitter On	Specify the RF channel 1 splitter. Splitting the signal after preselector to both frontends. Activated: Enable splitterStructure MeasCtrlTemplate Not activated: Disable splitter Field: MeasCtrl.MeasCtrl.Splitter

Name	Description
Auto Attenuation (only for block measurements)	Specify the attenuation strategy. Activated: Auto attenuation. If activated and the signal strength is too strong: <ol style="list-style-type: none"> 1. The preamplifier is deactivated. 2. If the signal strength is still too strong, the attenuation will be increased in 5dB steps till the signal strength is ok. <p>Note: After each reconfiguration step the I/Q measurement is restarted.</p> <p>Field: <code>MeasCtrl.MeasCtrl.AttStrategy</code></p>

Filter Design

Name	Description
Design	Open a dialog window to create and modify filters for I/Q data measurements. For details see Digital I/Q Interface Status As described in chapter 8.1 Main Window "TSMW Matlab IQ Interface", various messages can appear in this window during a measurement. Please see the section on Digital I/Q Interface Status in chapter 8.1 for a description of the messages. Filter Design Dialog on page 54.
Load from file	Loads a saved filter specification.
Send to TSMW	Transmit the last design filter to the R&S TSMW. Once a filter specification is transferred, it is available on the drop down menu "FilterID". M-function: <code>TSMWIQSetup</code>

Frontend 1 Settings/ Frontend 2 Settings

Correspond to `MeasCtrl.ChannelCtrl1` / `MeasCtrl.ChannelCtrl2`.

Name	Description
Use Frontend 1	If checked, a measurement will be performed on frontend 1. This means that only the field <code>ChannelCtrl1</code> is present in the <code>MeasCtrl</code> structure
Use Frontend 2	If checked, a measurement will be performed on frontend 2. This means that only the field <code>ChannelCtrl2</code> is present in the <code>MeasCtrl</code> structure.

Name	Description
Frequency (in MHz)	Defines the center frequency. Field: <code>MeasCtrl.ChannelCtrl1.Frequency</code> , <code>MeasCtrl.ChannelCtrl2.Frequency</code>
Attenuation	Defines the attenuation to use. In the auto attenuation mode this is the initial (suggested) attenuation to use for the measurement. Field: <code>MeasCtrl.ChannelCtrl1.Attenuation</code> , <code>MeasCtrl.ChannelCtrl2.Attenuation</code> ,

Name	Description
Preamp. On	Defines whether the preamp. shall be enabled or not. In auto attenuation mode, this is the initial (suggested) setting of the preamp. Field: <code>MeasCtrl.ChannelCtrl1.Preamp,</code> <code>MeasCtrl.ChannelCtrl2.Preamp</code>
Calibration	Defines whether the calibration input shall fit to the used frontend. Field: <code>MeasCtrl.ChannelCtrl1.CalibInput,</code> <code>MeasCtrl.ChannelCtrl2.CalibInput.</code>
Channels	Defines the number of channels that shall be used (1..4). The received bandwidth of the selected frontend is split up into the corresponding number of channels. Field: <code>MeasCtrl.ChannelCtrl1.NoOfChannels,</code> <code>MeasCtrl.ChannelCtrl2.NoOfChannels</code>
Channels Frequency Shift 1 to 4 (in MHz)	Defines a separate delay in I/Q samples after filtering and downsampling for each channel. Field: <code>MeasCtrl.ChannelCtrl1.FreqShift1 to 4,</code> <code>MeasCtrl.ChannelCtrl2.FreqShift1 to 4</code>
Digital I/Q Out	Defines whether the measured I/Q data is streamed either over the R&S Digital I/Q Interface or the GigaBit LAN interface. Field: <code>MeasCtrl.ChannelCtrl1.DigIqOnOff,</code> <code>MeasCtrl.ChannelCtrl2.DigIqOnOff</code>



Digital I/Q Out checkbox for Frontend 2

In previous versions the checkbox to enable Digital I/Q Interface output on the Frontend 2 was greyed out as the function was not available. As you can see in Figure 8-1: Main window of TSMW IQ Interface demo application, the checkbox is now also available on Frontend 2.



Checkbox is valid for all channels

If the Digital I/Q Out checkbox is ticked, the I/Q data of all the channels configured on the associated Frontend will be sent to the Digital I/Q Interface output.

Single Measure

Performs a single I/Q data measurement.

Name	Description
Measure	Start a new measurement with above defined settings. M-function: <code>TSMWIQMeasure</code>

Name	Description
Output Format	Define the unpack format for the measurement result. M-functions: TSMWIIQGetDataInt16, TSMWIIQGetDataInt32, TSMWIIQGetDataSingle, TSMWIIQGetDataDouble
GetData	Transfer the measured data from R&S TSMW to the local computer.
Plot Channel	Show results.

Cont. Measure

Repeats the number of single measurement without deleting the I/Q data of the last measurement. This is particularly of interest when using the configurations "Average" or "Max Hold" to take previously recorded measurements into account for averaging / maximum search.

Name	Description
Output Format	Define the unpack format for the measurement result. M-functions: TSMWIIQGetDataInt16, TSMWIIQGetDataInt32, TSMWIIQGetDataSingle, TSMWIIQGetDataDouble
Average	If checked an average spectrum is calculated by averaging over the specified number of measurements.
Start	Start a continuous measurement immediately. M-function: TSMWIIQMeasure
Stop	Stop a continuous measurement.

Analyze Custom

For all elements a corresponding dummy function in file `TSMWIIQInterfaceDemo.m` exists.

Name	Description
Set Parameter	M-function in file <code>TSMWIIQInterfaceDemo.m</code> can be customized.
Measure	M-function in file <code>TSMWIIQInterfaceDemo.m</code> can be customized
Analyze Data	M-function in file <code>TSMWIIQInterfaceDemo.m</code> can be customized.

Export

Name	Description
Export to Worksp.	Export the measured data to the MATLAB Workspace.
Export to .mat	Export the settings to a *.mat File.

8.1.2 GUI – output functions

During a measurement the application provides:

- information on the status of the Digital I/Q interface

- A graphical representation of the measurement data

Digital I/Q Interface Status

During a measurement, various messages can appear in this panel on the GUI.

Message	Description
"Connection error"	Written with a bold fontweight in a dark red color, if a protocol error has occurred in the Digital I/Q Interface. The error message will persist 5 seconds so the users may read it. In most cases, a watchdog will be triggered to force a reinitialisation of the interface. [Note: this is the only message with a persistence and not written in normal fontweight, in a black color]
"Connection in progress to %s"	While the handshaking procedure with the other device connected is running, and no error occurred. "%s" will be replaced by "Unknown" at the beginning of the procedure, and by the real name of the device connected once it is known [e.g. in the screenshots: SMBV100A or IQR100].
"Connected to %s1 (%s2)"	When the digital I/Q interface has been successfully configured. "%s1" is the name of the device connected, "%s2" the transfer mode negotiated with 2 possible strings: - "single channel": only 1 channel can be transmitted on the Digital I/Q Interface. In this mode, either the first channel of FrontEnd1 (also known as Channel 1) or the first channel of Frontend2, if FrontEnd1 is disabled, will be transmitted. - "multi channels": more than 1 channel can be transmitted on the Digital I/Q Interface.
"Unknown/undefined state"	This message should never occur. [This message may only happen with mismatching FW / SW versions]

Plot Window

For viewing the measurement data a new window appears when a single measurement is performed over the LAN interface, see Figure 8-2: Example plot "Frontend 2"

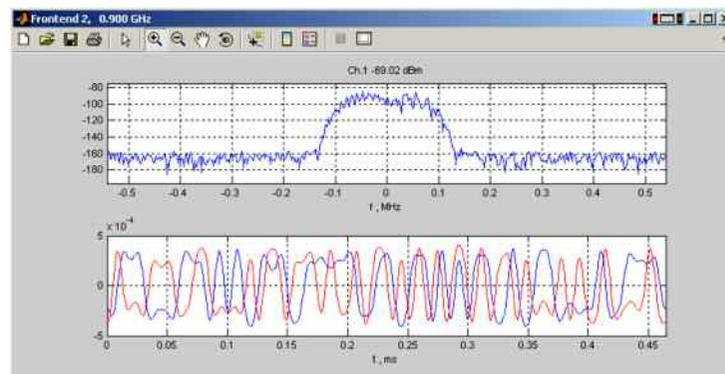


Figure 8-2: Example plot "Frontend 2"

8.2 Streaming Dialog

The "TSMW I/Q Streaming Tool" dialog window controls the streaming settings for I/Q data streaming measurements. It is possible to set and view the parameters for streaming measurements. Furthermore a streaming measurement is controlled and analyzed over this window.

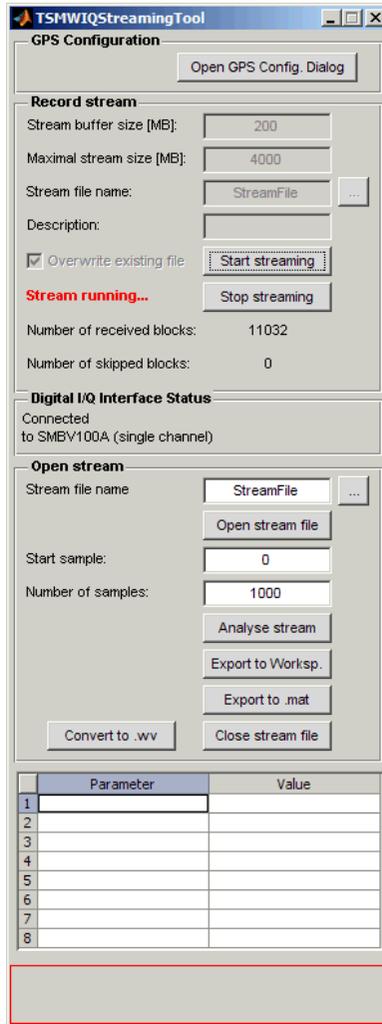


Figure 8-3: "TSMW IQ Streaming Tool" dialog window

The following table describes the functionality and the corresponding MATLAB functions of the dialog window.

GPS Configuration

Name	Description
Open GPS Config. Dialog	Open a dialog window which controls the GPS server settings. For details see GPS Server Configuration Dialog on page 56.

Record stream

Name	Description
Stream buffer size [MB]	Specifies the stream buffer. Choose a value at least 200MB. Default value is 200MB. Field: MeasCtrl.StreamCtrl.StreamBufferSize
Maximal stream size [MB]	Specifies the maximum streaming size in MB. Field: MeasCtrl.StreamCtrl.MaxStreamSize Note: The stream data is automatically splitted into several files such that a maximum file size of 2GB is not exceeded.
Stream file name	Specifies the file where the streaming data has to be saved. M-function: TSMWIQStream, parameter: FileName
Description	In this field the user can write a short description about the streaming file. This information is saved with the streaming file. Field: StreamInfo.Description
Overwrite existing file	Specifies whether an already existing streaming data file will be overwritten. M-function: TSMWIQStream, parameter: CreateIfExists
Start streaming	Start a streaming measurement and save the I/Q data in the specified file. M-function: TSMWIQStream
Stop streaming	Stop the current streaming measurement. M-function: TSMWIQStopStreaming

Open stream

Name	Description
Stream file name	Specifies the streaming file to view. M-function: TSMWIQOpenStreamFile, parameter: FileName
Open stream file	Open the streaming file. Furthermore information about parameter values is shown inside the dialog window. M-function: TSMWIQOpenStreamFile
Start sample	Index of I/Q sample where to start.
Number of samples	Enter the number of samples for analysis
Analyze stream	Starts the analysis of the streaming data. (Plot spectrum and I/Q components of selected segment of the recorded I/Q stream.)
Export to .Worksp.	Export the stream data to the MATLAB Workspace.
Export to .mat File	Export the stream data to *.mat file format.
Convert to .wav	Convert streamed data to R&S Signal Generator waveform and list format in order to allow RF-replay on R&S Signal Generators. Therefore the corresponding signal is normalized prior to conversion in order to achieve best replay performance. The necessary sig. generator level for regeneration at exact the same signal level is written into the comment section of the resulting *.wav file.
Close stream file	Close the open stream file. M-function: TSMWIQCloseStreamFile

Digital I/Q Interface Status

As described in chapter 8.1 Main Window “TSMW Matlab IQ Interface”, various messages can appear in this window during a measurement.

Please see the section on Digital I/Q Interface Status in chapter 8.1 for a description of the messages.

8.3 Filter Design Dialog

The “TSMW Filter Design Tool” dialog window creates and modifies filters for I/Q data measurements. The designed filters can be saved as .mat file for loading it back into the Demo application. Furthermore, the designed filters can be saved as C++ header file and could be included it into a C++ user application.

The drop down menu offers several filter design algorithms with predefined values.

The filter designer base on the MATLAB Filter Design functions.

It is possible to check the new designed filter over the “View transfers function” functionality. The black vertical line in the "Filter Response" window corresponds to $\text{Sampling Rate} / 2$, i.e. the maximum frequency without aliasing.

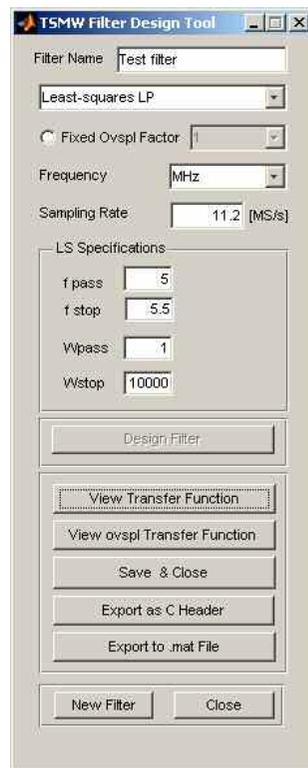


Figure 8-4: “TSMW Filter Design Tool” dialog window

The following table describes the functionality and the corresponding MATLAB functions of the “TSMW Filter Design Tool” dialog window.

Name	Description
Filter Name	Defines a filter name.
-/-	Select the desired filter type.
Fixed Ovspl Factor	If checked a filter coefficient oversampling factor can be specified Field: <code>FilterSpec.OvsplFact</code>
Design Filter	A new filter with the entered values will be created.
View Transfer Function	The function checks the designed filter.
Name	Description
View ovspl Transfer	The function checks the designed filter.
Save & Close	Save the designed filter and return to the main window of the MATLAB Demo application. The designed filter is available for measure purpose on the current instance of the MATLAB Demo application.
Export as C Header	Export the designed filter to a C Header file.
Export to .mat File	Export the designed filter to an M-file.
New Filter	Reset the values and start to design a new filter.
Close	Close the "TSMW Filter Design Tool" window without saving and return to the main window.

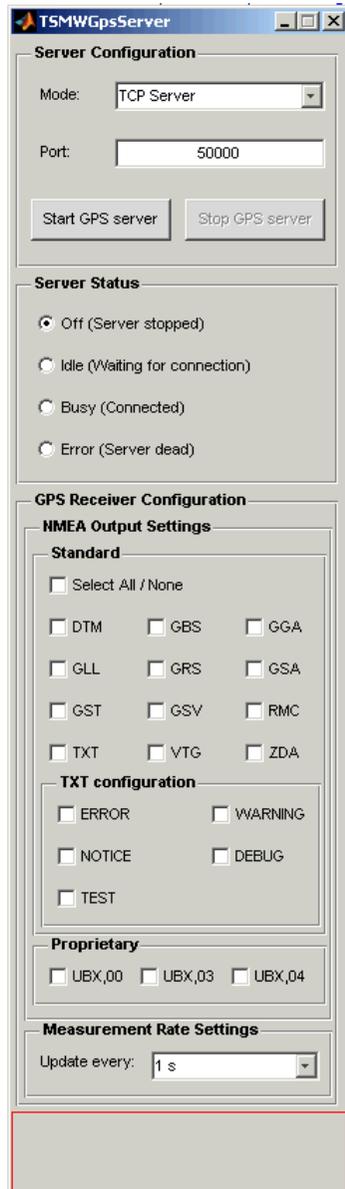
8.4 GPS Server Configuration Dialog

The “TSMW GPS Server” dialog window controls the transmission of NMEA 0183 sentences over the LAN interface. It is possible to select the sentences to transmit and the update rate of the GPS receiver chip.

The dialog window may be closed after configuring the GPS Server.



The GPS Server runs on the PC connected to the R&S TSMW.



Server Configuration

Name	Description
Mode	currently only TCP Server supported.
Port	listening port of the server, valid range 0 to 65535, recommended range 49152 to 65535
Start server buttons	Start the server M-function: TSMWGPSStartTcpServer
Stop server buttons	Stop the server M-function: TSMWGPSStopServer

Server Status

Status of the server, updated every second.

GPS Receiver Configuration**NMEA Output Settings****Standard**

Name	Description
Select All / None	Select / deselect all the standard NMEA messages
DTM, GBS, GGA, ..., TXT, VTG, ZDA	Standard NMEA 0183 messages, configured by the register UBX-CFG-MSG. M-function: TSMWGPSSendUBX_CFG_MSG
TXT configuration	Select the type of messages to transmit over the TXT message: ERROR, WARNING, NOTICE, DEBUG, TEST, configured by the register UBX-CFG-INF. M-function: TSMWGPSSendUBX_CFG_INF

Proprietary

Name	Description
UBX,00	Proprietary NMEA message from <i>U-Blox</i> , carrying longitude/latitude position data, configured by the register UBX-CFG-MSG. M-function: TSMWGPSSendUBX_CFG_MSG
UBX,03	Proprietary NMEA message from <i>U-Blox</i> , carrying satellite status, configured by the register UBX-CFG-MSG. M-function: TSMWGPSSendUBX_CFG_MSG
UBX,04	Proprietary NMEA message from <i>U-Blox</i> , carrying time of day and clock information, configured by the register UBX-CFG-MSG. M-function: TSMWGPSSendUBX_CFG_MSG

Measurement Rate Settings

Set the GPS measurements rate, possible values: 200 ms, 250 ms, 500 ms, 1 s, 2 s, 4 s, 8 s, configured by the register UBX-CFG-RATE.

M-function: TSMWGPSSendUBX_CFG_RATE

Data format generated by the server

```
<message counter><TAB><IQ time><TAB><NMEA sentence><CR><LF>
```

with

<message counter>	A 32bits unsigned integer value incremented for each NMEA message received from the GPS receiver.
<IQ time>	A 64bits unsigned integer value containing the current value of the TSMW IQ counter.
<NMEA sentence>	A NMEA sentence sent by the GPS receiver.
<TAB>	A horizontal tab character.
<CR>	A carriage return character.
<LF>	A line feed character.

This format can be saved in a file with network tools like PuTTY (<http://www.chiark.greenend.org.uk/~sgtatham/putty/>) (raw format mode) or netcat, and is compatible with the CSV format used by spreadsheet editors.

Example

```
Plink.exe -P <port> -raw <ip-address> > <filename>
```

with

<port>: port used by the GPS server, 50000 by default

<ip-address>: ip address of the PC running the GPS server, 127.0.0.1 can be used to save the file locally on the PC running the GPS server

<filename>: name of the file which is logging the data, e.g. C:\nmea.log

Use the R&S TSMW MATLAB IQ Interface Application (compiled)

9 How to...

This chapter describes typical use cases for the R&S TSMW MATLAB IQ Interface application.

9.1 Use the R&S TSMW MATLAB IQ Interface Application (compiled)

Prerequisite:

- The R&S TSMW-K1 software is installed on the PC.
For detailed information how to install refer to chapter Install on page 13.
- The MCR environment is installed.
- The R&S TSMW is successful connected to the PC.

Procedure:

- ▶ Select "Start"->"Programs"->"Rohde&Schwarz"->"TSMW-K1 <ver. no.>" -> "TSMW MatlabDemo IQ".



Figure 9-1: Start demo application

The main window of the R&S TSMW MATLAB IQ Interface application opens.

9.2 Use the R&S TSMW MATLAB IQ Interface Application (not compiled)

Procedure:

1. Start MATLAB.
2. On the command window type : `"TSMWInterfaceDemo"`

The R&S TSMW Matlab IQ Interface application starts and the main window of the application open.

9.3 Use native C++ Interface

Procedure:

- ▶ Copy all files from

C:\Program Files\MATLAB\R2007a\bin\toolbox\RS_TSMWIQInterface\lib
including subfolders to your C++ user application folder.

9.4 Connect to the R&S TSMW

Prerequisite:

- A LAN connection between the computer and the R&S TSMW is required. How to link the software to R&S TSMW refer to the corresponding R&S TSMW operation manual.

Procedure:

1. Start the R&S TSMW MATLAB IQ Interface application.
The R&S TSMW MATLAB IQ Interface script window opens.
2. Click on the "Init. Interface" button.
An entry field for the R&S TSMW IP address appears.
3. Enter the IP address of the R&S TSMW device.
4. Click the "Connect" button.

A connection to the R&S TSMW is established.

9.5 Create a resampling filter

The following example demonstrates how to use the TSMW Filter Design Tool to design a user specific resampling filter in order to get a user specific sampling rates. In this example, the desired signal is a WiMAX signal using a 1024-FFT and 10MHz bandwidth, specified in IEEE 802.16. The Sampling rate is specified with 11.2 MS/s, which means that after resampling to 11.2MS/s the baseband signal spans +/-5.6 MHz. The WiMAX standard for a 1024-FFT signal, defines a guard interval of 92 sub carriers on the left and 91 on the right. That means the signal of interest is between $\pm (512 - 91) * 11.2 / 1024 \text{ MHz} = 4.6047 \text{ MHz}$. Hence a sensible choice for the pass band frequency is 4.6MHz. To sufficiently suppress interferences caused by adjacent channels, the stop band edge is chosen to be the edge of the baseband signal, i.e., 5.6 MHz.



Sampling Rate

The minimum supported sampling rate is 0.22MS/s. However, best filtering results will be achieved with sampling rates above 1MS/s

Prerequisites:

- The R&S TSMW MATLAB IQ Interface application is running.
- A connection to a R&S TSMW is established.

Procedure:

1. On the “TSMW MATLAB IQ Interface” window click the "Design" button. The Filter Design GUI opens.
2. From the “-select Filtertype-“ button choose “Least squares Lowpass”. Now a parameter input section appears, where the input parameters for the specific design can be defined.

Note:

A filter design using the “Modified Equiripple Lowpass” or “Raised Cosine Lowpass” function is also possible, but with the restrictions on low- , and passband edges, the “Least squares Lowpass” will usually give the best design.

3. Enter the parameters:

For this example as follow (see figure):

- a. Set the Sampling rate to 11.2 [MS/s],
- b. Enter the passband frequency (f pass)
- c. Enter the stopband frequency (f stop).
- d. The parameters wpass and wstop are the weighting factors for the pass band and stop band.

Decreasing the weighting factor for the stop band will decrease the stop band attenuation, increasing it will increase it and cost a higher pass band ripple.

Sensible values for the stop band weighting factor are 1000 to 10000. The pass band weighting factor will usually be 1.

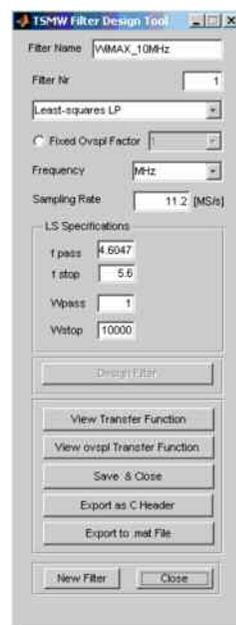


Figure 9-2: Design filter for WiMAX measurements

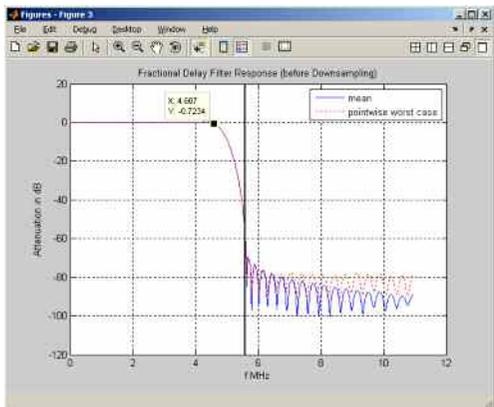
4. If the parameters are selected, as depicted in figure and step 3, press the “Design

Filter” button to start the filter designer.
A new filter is created and new buttons appears with further actions on the designed filter.

View Transfer Function

To check the designed filter, click “View Transfer Function”

The transfer function plot appears. It shows the filter response of the fractional filter before downsampling. The black line in the transfer function plot shows 1/2 of the sampling frequency, which is the maximum input frequency that does not produce aliasing products. In the figure below the marker is set to the passband edge.

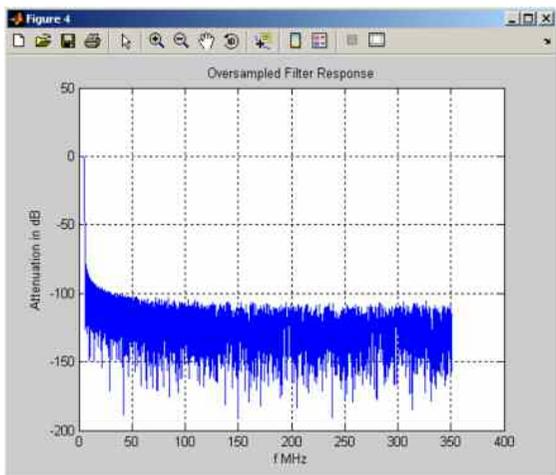


View ovspl Transfer Function

To check the designed filter, click “View ovspl Transfer Function”

With this plot you can verify if the designed filter fulfils the requirements.

The plot shows the impulse response of the oversampled transfer function as depicted in the picture below.



Save & Close

Pass the designed filter to the current instance of R&S TSMW MATLAB IQ Interface application.

Export as C Header

Save and export the designed filter to a C Header file. Thus it is possible to load the designed filter into C++ applications, afterwards.

Export to .mat File

Save and export the designed filter to a *.mat file. Thus it is possible to load the designed filter into the R&S TSMW IQ Interface Demo Application, afterwards.

A new filter is created.



Submit filter

To make a filter selectable you have to submit the filter explicitly to R&S TSMW. How to submit the filter see Use predefined filters on page 63.

9.6 Use predefined filters

Prerequisites:

- The Demo GUI application is running.
- A connection to the R&S TSMW is established.

Procedure:

1. Click the "Load from File" to load a filter or design a new filter over the "Design" button. To design a new filter see procedure Create a resampling filter.
2. Click the "Send to TSMW" button.

The filter is transferred to the R&S TSMW and selectable over the "Filter ID" drop down menu for use.

9.7 Modify the measurement values

Prerequisites:

- The R&S TSMW MATLAB IQ Interface application is running and a connection to the R&S TSMW is established.

Procedure:

1. Edit the general values in the group box "Measure Settings". Please refer to chapter Structure: MeasCtrl on page 30 for an explanation of the parameters.
2. Check one check box "Use Frontend 1" or "Use Frontend 2" to activate the corresponding group box "Frontend 1 Settings" or "Frontend 2 Settings".
3. Edit the values in the activated group box "Frontend 1 Settings" or "Frontend 2 Settings". Please refer to chapter Structure: MeasCtrl on page 30 for an explanation of the parameters.
 - a. Before a filter can be selected in the "Filter ID." drop down menu, it has to be designed. How to design a filter and make it available are explained in procedure Create a resampling filter and Use predefined filters on page 63.

The measurement settings are modified.
Now a measurement can be performed.

9.8 Perform a continuous measurement

Prerequisites:

- The R&S TSMW MATLAB IQ Interface application is running and a connection to the R&S TSMW is established.
- The measurement settings are defined.

Procedure:

1. It is possible to customize the average result of a number of measurements.
 - a. Therefore click on the "Average Result" check box.
The "No. of. Meas." entry field will be activated.
 - b. Enter the desired number of measurements for the generation of average results.
2. Select from the drop down list box "Output Format" the desired output width.
3. Click the "Start" button to start the continuous measurement.
The measurement starts and the data are available on the application.
4. Click the "Stop" button to stop the measurement.

A measurement was performed.

9.9 Perform a single measurement

Prerequisites:

- The R&S TSMW MATLAB IQ Interface application is running and a connection to the R&S TSMW is established.
- The measurement settings are defined.

Procedure:

1. Click the "Measure" button.
The R&S TSMW starts a measurement. Until now no measurement data are transferred to the MATLAB application.
2. Select from the drop down list box "Output Format" the desired output width.
3. Click the "GetData" button to transfer the measurement result to the application.
4. Click the "Plot Channel" button to visualize the measurement.

A measurement was performed.

9.10 Record streaming measurement data

Prerequisites:

- The R&S TSMW MATLAB IQ Interface application is running and a connection to the R&S TSMW is established.
- The measurement settings in the main window are defined.
- The possible maximum bandwidth is known.
- To stream measured I/Q data over the R&S Digital I/Q Interface (R&S TSMW-B1), activate the checkbox "Digital I/Q Out". Otherwise the I/Q data is streamed over the GigaBit LAN interface.

Note:

The selected output path (Digital I/Q or Gigabit LAN) is applied to all the channels configured on the Frontend. If more than 1 channel is configured to be sent over the R&S Digital I/Q Interface and the target device only supports 1 channel, only the 1st channel of the configuration set will be transmitted to the target device. This channel will be either Channel 1 of FrontEnd 1 (if it is configured) or Channel 1 of FrontEnd 2, if FrontEnd 1 is not configured or not using the R&S Digital I/Q Interface.

Therefore calculate the maximum I/Q data rate and compare it to the supported maximum bandwidth of the used computer equipment. For more details see Additional Requirements for I/Q Streaming on page 12.

Procedure:

1. Click the "Open Streaming Dialog" button.
The TSMW IQ Streaming Tool dialog window opens.

Note:

The parameters for the streaming measurement will be taken from the main TSMW IQ Interface GUI at the time when the streaming dialog opens. This means that in order to start a new stream measurement, you have to close the streaming dialog, change the measurement parameters and reopen the streaming dialog again.

2. Inside the "Record stream" group fill-out the following fields:
 - a. Enter the stream buffer size. It has to be at least 200 MB.
 - b. Enter the maximum stream size. The maximum value is 4000 MB.
 - c. Enter a stream file name
 - d. Enter a short description.

Note:

If the measured I/Q data are streamed over the R&S TSMW-B1 interface the fields are not necessary to be filled out and therefore grayed out (not editable).

3. Click "Start streaming" to start the measurement.
The streaming measurement is started.
4. Click "Stop streaming" to stop recording.

If the measured I/Q data were streamed over the Gigabit LAN interface the data are saved in a file for further analysis.

If the measured I/Q data were streamed over the R&S TSMW-B1 interface the data are handled on the destination instrument. Thus the I/Q data are now completely independent from the R&S TSMW-K1 software.

9.11 View streaming measurement data

Prerequisites:

- The R&S TSMW MATLAB IQ Interface application is running.
- Recorded data of a streaming measurement exists.

Procedure:

1. Click the "Open Streaming Dialog" button.
The TSMWIQStreamingTool dialog window opens.
2. Inside the "Open stream" group:
 - a. Enter the name of the streaming file into the field "Stream file name".
 - b. Click "Open stream file".
Information about the streaming file appears at the end of the window.
 - c. Enter the number of start samples into the field "Number of start sample".
 - d. Enter the number of samples into the field "Number of samples".
 - e. Click "Analyze stream"
A plot appears to analyze the data.
 - f. Click "Close stream file".
The streaming file is closed.

9.12 Export single data

Prerequisites:

- The R&S TSMW MATLAB IQ Interface application is running.
- A single measurement or a continuous measurement was performed so that a measured data block is loaded for analysis on the application.
- A MATLAB Workspace is open (only required for "Export to Workspace").

Procedures:

Export to MATLAB Workspace

- ▶ On the main window of the application click "Export to the Workspace".

The measured data block is exported to the current open MATLAB Workspace.

Export to .mat file

- ▶ On the main window of the application click "Export to the Workspace".

The measured data block is exported to the current open MATLAB Workspace.

9.13 Export streamed data.

Prerequisites:

- The R&S TSMW MATLAB IQ Interface application is running.
- Recorded data of a streaming measurement exists.
- A MATLAB Workspace is open (only required for "Export to Workspace")

Procedure:

Export to MATLAB Workspace

1. Click the "Open Streaming Dialog" button.
The TSMWIQStreamingTool dialog window opens.
2. Inside the "Open stream" group:
 - a. Enter the name of the streaming file for export into the field "Stream file name".
 - b. Click "Open stream file".
Information about the streaming file appears at the end of the window.
 - c. Click "Export to Worksp.".
The requested number of samples (parameters "Start sample" and "Number of samples") are exported into the current running MATLAB Workspace.
 - d. Click "Close stream file".
The streaming file is closed.

Export to .mat file

1. Click the "Open Streaming Dialog" button.
The TSMWIQStreamingTool dialog window opens.
2. Inside the "Open stream" group:
 - a. Enter the name of the streaming file for export into the field "Stream file name".
 - b. Click "Open stream file".
Information about the streaming file appears at the end of the window.
 - c. Click "Export to *mat".
The requested number of samples (parameters "Start sample" and "Number of samples") are exported into a *.mat file.
 - d. Click "Close stream file".
The streaming file is closed.

9.14 Release the connection to the R&S TSMW IQ interface

Prerequisites:

- The R&S TSMW MATLAB IQ Interface script is running and a connection to the R&S TSMW is established.
- All measurements are stopped.

Procedure:

- ▶ Click the "ReleaseInterface" button.

The connection to the R&S TSMW is released.

9.15 Perform streaming measurements (C++ API)

Examples how to perform streaming measurements with the C++ API are available on the CD-ROM and inside the folder

<R&S TSMW-K1 installation directory>\Examples\CPP.

C++ example file	Description
TestStreaming	Performs online streaming. That mean during streaming the measured data are directly accessible for further processing.
TestStreamingFile	Performs streaming measurements into a file. After finishing the measurement, the file with the streamed data is accessible for further processing.

10 Troubleshooting

10.1 Working with a second R&S TSMW

Description

After replacing the R&S TSMW with another one the computer do not recognize the new connected R&S TSMW.

Possible reason:

The Address Resolution Protocol Cache (ARP-Cache) on the computer is not updated.

The ARP-Cache of the operating system has not updated the IP-to-Physical address translation tables with the physical address of the new connected R&S TSMW. The table still keeps the physical address of the R&S TSMW which was connected before.

Proposal for solution

On the computer open a command prompt and enter following command:

```
arp -d
```

The command deletes all entries in the IP-to-Physical address translation tables used by ARP.



Automatically update of the ARP-Cache

Usually the ARP-Cache will be refreshed by the operating system within 15 minutes.

10.2 Problems during LAN connection to the R&S TSMW

Description

Over the ethernet card no connection is possible between R&S TSMW and a computer.

Possible reason

The computer firewall blocks the connection.

Proposal for solution

On the computer deactivate the firewall rules for the ethernet card which is connected to the R&S TSMW.

If the problem still exists, check all installation settings of the R&S TSMW. Also see the R&S TSMW operation manual chapter 11 "Troubleshooting" for more details about troubleshooting the LAN connection.

10.3 Skipped I/Q data blocks during I/Q streaming

Description

The field "Number of skipped blocks" shows an increasable high value of skipped I/Q data blocks.

Possible reason

The available hardware is too slow for the I/Q data streaming.

That means that during the transfer of the data from R&S TSMW to computer I/Q data blocks are skipped.

Proposal for solution

- Check the used hardware if it fits the system requirements. For details about the system requirements see chapter 12 Additional Requirements for I/Q Streaming on page 12.
- Calculate the I/Q data transfer rate needed and compare with your hardware capabilities.
- Decrease the bit resolution used for starting the measurement.
- Decrease the sampling rate.
- Decrease the number of activated channels/frontends.

Appendix

A C++ Header Files

The current C++ header `TSMWIQInterfaceTypes.h` and `TSMWIQInterfaceFunc.h` are available on the installation folder `<TSMW-K1 installation directory>\inc`.

B M-Functions

All R&S TSMW-K1 functions for the R&S TSMW MATLAB IQ interface are explained in the following sub chapters. The corresponding C++ functions with the correct command syntax are also shown. Most MATLAB parameters correspond to the C++ parameters.

B.1 MeasCtrlTemplate

Description:

This function creates a template for the `MeasCtrl` structure for double channel (MIMO) measurements with default values.

Command syntax:

```
[MeasCtrl] = MeasCtrlTemplate();
```

Parameter(s):

-/-

Return value(s):

`MeasCtrl`

Specifies a `MeasCtrl` structure with default values for double channel (MIMO) measurements. This means that the general measurement control sub-structure, the sub-structure for RF channel 1 as well as the sub-structure for RF channel 2 parameters will be set.

For a detailed description of the structure refer to chapter Structure: `MeasCtrl` on page 30.

Corresponding C++ command:

-/-

B.2 MeasCtrlTemplate_RF1

Description:

The function creates a template for the `MeasCtrl` structure for RF channel 1 measurements.

Command syntax:

```
[MeasCtrl] = MeasCtrlTemplate_RF1 ();
```

Parameter(s):

-/-

Return value(s):

<code>MeasCtrl</code>	Specifies a <code>MeasCtrl</code> structure with default values for RF channel 1 measurements. This means that the general measurement control sub-structure and the sub-structure for RF channel 1 parameters will be set. For a detailed description of the structure refer to chapter Structure: <code>MeasCtrl</code> on page 30.
-----------------------	---

Corresponding C++ command:

-/-

B.3 MeasCtrlTemplate_RF2

Description:

The function creates a template for the `MeasCtrl` structure for RF channel 2 measurements.

Command syntax:

```
[MeasCtrl] = MeasCtrlTemplate_RF2 ();
```

Parameter(s):

-/-

Return value(s):

<code>MeasCtrl</code>	Specifies a <code>MeasCtrl</code> structure with default values for RF channel 2 measurements. This means that the general measurement control sub-structure and the sub-structure for RF channel 2 parameters will be set. For a detailed description of the structure refer to chapter Structure: <code>MeasCtrl</code> on page 30.
-----------------------	---

Corresponding C++ command:

-/-

B.4 MeasCtrlTemplate_RF1Stream

Description:

The function creates a template for the `MeasCtrl` structure for RF channel 1 streaming measurements.

**Resampling filter specification**

In order to make use of this template, an appropriate resampling filter specification has to be transferred to the R&S TSMW before a streaming measurement is started. For filter design you might use the provided R&S TSMW Filter Design Tool.

Command syntax:

```
[MeasCtrl] = MeasCtrlTemplate_RF1Stream();
```

Parameter(s):

-/-

Return value(s):

`MeasCtrl`

Specifies a `MeasCtrl` structure with default values for RF channel 1 streaming measurements. This means that the general measurement control sub-structure and the sub-structure for RF channel 1 parameters will be set.

For a detailed description of the structure refer to chapter chapter Structure: `MeasCtrl` on page 30.

Corresponding C++ command:

-/-

B.5 MeasCtrlTemplate_RF2Stream

Description:

The function creates a template for the `MeasCtrl` structure for RF channel 2 streaming measurements.

Command syntax:

```
[MeasCtrl] = MeasCtrlTemplate_RF2Stream();
```

Parameter(s):

-/-

Return value(s):

MeasCtrl

Specifies a `MeasCtrl` structure with default values for RF channel 2 streaming measurements. This means that the general measurement control sub-structure and the sub-structure for RF channel 2 parameters will be set.

For a detailed description of the structure refer chapter Structure: `MeasCtrl` on page 30.

Corresponding C++ command:

-/-

B.6 TSMWConnect**Description:**

Establish a connection to the R&S TSMW with the given IP address and given options.

Command syntax:

```
[ErrorCode, TSMWID] = TSMWConnect(IPAddress, TSMWOptions);
```

Parameter(s):

IPAddress

IP address of the R&S TSMW.

TSMWOptions

Specifies R&S TSMW operating options. If omitted, default options will be used. That means both frontends and all amplifiers are enabled.

Note:

When the R&S TSMW is powered, the current firmware version always turns the frontend on.

Return value(s):

ErrorCode

0 if successful. Otherwise the error code is returned.

TSMWID

The ID represents the R&S TSMW connection.

Corresponding C++ command:

```
int ErrorCode = TSMWConnect_c(char* IPAddress,
                             TSMW_IQIF_MODE_t *pTSMWMode,
                             unsigned short *pTSMWID);
```

B.7 TSMWExtReEnable**Description:**

The function enables or disables usage of external reference clock.

**Connect external reference clock before calling the function**

If no external reference clock is connected and the function `TSMWExtRefEnable.m` is executed, measurements on RF 2 will not work.

Therefore the external reference has already connected before the function `TSMWExtRefEnable.m` is called.

Command syntax:

```
[ErrorCode] = TSMWExtRefEnable(TSMWID, Enable);
```

Parameter(s):

TSMWID	Defines the used R&S TSMW.
Enable	1: Enable usage of external reference clock 0: Disable

Return value(s):

ErrorCode	0 if successful. Otherwise the error code is returned.
-----------	--

Corresponding C++ command:

```
int ErrorCode = TSMWTGEnable_c(unsigned short TSMWID,
                               int Enable);
```

B.8 TSWMGetFIRParam**Description:**

The function assists for custom filter design. It calculates the maximum number of FIR coefficients and an appropriate filter oversampling factor given a specific down sampling factor. The number of FIR filter tabs is given by `NoOfCoeffs/CoeffOver`.

Command syntax:

```
[NoOfCoeffs, OvsplFact] = TSMWGetFIRParam(Ndown, OvsplFact);
```

Parameter(s):

Ndown	Defines the fractional down sampling factor. Given the desired sampling rate f_s , the down sampling factor is defined as $f_s/(395e6/18)$. I.e.: The desired sampling rate divided by the native R&S TSMW sampling rate.
OvsplFact	If given, the specified oversampling factor is used. If not, an appropriate oversampling factor is used. It has to be 1, 2, 4, 8, 16 or 32. This parameter is optional.

Return value(s):

NoOfCoeffs	Maximum number of FIR filters coefficients. The number of FIR filter tabs is given by $NoOfCoeffs / OvsplFact$.
OvsplFact	Selected oversampling factor.

Corresponding C++ command:

```
-/-
```

B.9 TSMWGetIQTime**Description:**

The function retrieves an estimate of the current I/Q time at the R&S TSMW. The I/Q counter starts as soon as a connection to the R&S TSMW is established. The counter counts the I/Q-samples in the native sampling rate of 395/18 MS/s. It has a valid bit width of 48 bit.

The I/Q-time specifies the time that is used to start measurements.

Command syntax:

```
[ErrorCode, IQTime] = TSMWGetIQTime(TSMWID);
```

Parameter(s):

TSMWID	Defines the instrument.
--------	-------------------------

Return value(s):

ErrorCode	0 if successful. Otherwise the error code is returned.
-----------	--

IQTime Specifies the current I/Q time.

Corresponding C++ command:

```
int ErrorCode = TSMWGetIQTime_c(unsigned short TSMWID,
                                unsigned __int64 *pIQTime);
```

B.10 TSMWGetLastError

Description:

Each function returns the error code "0" if successful. Otherwise, they return a number which defines the error code. To get the corresponding error text, it calls the function `TSMWGetLastError`. The function returns the error information of the occurred error and resets the error code variable again to zero.



Variable ErrorCode

Please keep in mind that only one `ErrorCode` variable is used for all functions. Therefore only the current returned error code is available.

Command syntax:

```
[ErrorText, ErrorCode] = TSMWGetLastError;
```

Parameter(s):

-/-

Return value(s):

<code>ErrorCode</code>	0 if successful. Otherwise the error code is returned.
<code>ErrorText</code>	If no error message exists the value zero is returned

Corresponding C++ command:

```
char* ErrorText = TSMWGetLastError_c( int *pErrorCode);
```

B.11 TSMWGetStreamError

Description:

The function returns the error message corresponding to the status of the streaming measurement. This function should be called when streaming measurement is stopped due to an error.

Command syntax:

```
[ErrorText] = TSMWGetStreamError(StreamID);
```

Parameter(s):

StreamID	ID of the corresponding stream
----------	--------------------------------

Return value(s):

ErrorText	If no error message exists an empty string is returned.
-----------	---

Corresponding C++ command:

```
char* ErrorText = TSMWGetStreamError_c(
    unsigned char StreamID);
```

B.12 TSMWGetVersion**Description:**

The function returns the version number of the R&S TSMW IQ interface.

Command syntax:

```
[VersionNo] = TSMWGetVersion;
```

Parameter(s):

-/-

Return value(s):

VersionNo	Returns the version number in a 32-bit integer format. It represents the following 4-byte version code: MAJOR.MINOR.PATCH.QFE.
-----------	--

Corresponding C++ command:

```
int VersionNo = TSMWGetVersion_c();
```

B.13 TSMWGetVersionText**Description:**

The function returns the version text of the R&S TSMW IQ interface.

Command syntax:

```
[VersionText] = TSMWGetVersionText;
```

Parameter(s):

```
-/-
```

Return value(s):

`VersionText` Returns the version text of the R&S TSMW IQ Interface.

Corresponding C++ command:

```
char* VersionText = TSMWGetVersionText_c();
```

B.14 TSMWGPSEnable

Description:

The function enables or disables GPS data subscription from R&S TSMW. NMEA and UBX GPS data lines will be received from the R&S TSMW and are stored in a ring buffer with a predefined size of 20 lines. If the buffer size is exceeded, the oldest received NMEA string or UBX message is overwritten. The NMEA and UBX buffers are independent.

Command syntax:

```
[ErrorCode] = TSMWGPSEnable(TSMWID, Enable);
```

Parameter(s):

<code>TSMWID</code>	Specifies the instrument.
<code>Enable</code>	1: Enable GPS data subscription. 0: Disable GPS data subscription.

Return value(s):

`ErrorCode` 0 if successful. Otherwise the error code is returned.

Corresponding C++ command:

```
int ErrorCode = TSMWGPSEnable_c(
                                unsigned short TSMWID,
                                int NoOfBufferedLines,
                                int Enable);
```

B.15 TSMWGPSGetNMEAline**Description:**

The function gets the oldest available NMEA text in buffer.

**Configure NMEA buffer**

The number of buffered NMEA text lines can be parameterized by TSMWGPSEnable_c. The MATLAB wrapper function TSMWGPSEnable uses a fixed buffer size of 20.

Command syntax:

```
[NMEAText] = TSMWGPSGetNMEAline();
```

Parameter(s):

-/-

Return value(s):

NMEAText Oldest available NMEA text in buffer.

Corresponding C++ command:

```
char NMEAText = TSMWGPSGetNMEAline_c();
```

B.16 TSMWGPSGetServerStatus**Description:**

Get the current status of the GPS server for the TSMW with the given TSMW-ID.

Command syntax:

```
[ErrorTxt, StatusCode, PortNumber] =
TSMWGPSGetServerStatus(TSMWID)
```

Parameter(s):

TSMWID ID of the TSMW.

Return value(s):

ErrorText Error message, an empty string is returned if no error occurred.

StatusCode 0: the server is not running.
1: the server is started and waiting for a client.
2: the server is connected to a client
3: an unrecoverable error occurred.

PortNumber Port number used by the server.

Corresponding C++ command:

```
char* ErrorText = TSMWGPSGetServerStatus_c(unsigned short
TSMWID, unsigned char *pStatusCode, unsigned short
*pPortNumber)
```

B.17 TSMWGPSSendNMEACmd

Description:

The function sends NMEA command to the R&S TSMW with the given TSMW ID.

Command syntax:

```
[ErrorCode] = TSMWGPSSendNMEACmd(TSMWID, NMEACmd);
```

Parameter(s):

TSMWID Specifies the instrument.

NMEACmd NMEA text command.

Return value(s):

ErrorCode 0 if successful. Otherwise the error code is returned.

Corresponding C++ command:

```
int ErrorCode = TSMWGPSSendNMEACmd_c(
int TSMWID, char* pNMEAString);
```

B.18 TSMWGPSSendUBX_CFG_INF**Description:**

Send UBX command CFG_INF to the TSMW with the given TSMW-ID. Returns zero if successful, otherwise error code.

Command syntax:

```
[ErrorCode] = TSMWGPSSendUBX_CFG_INF(TSMWID, TxtError,
TxtWarning, TxtNotice, TxtDebug, TxtTest)
```

Parameter(s):

TSMWID	ID of the TSMW
TxtError	0: error txt messages disabled, otherwise enabled.
TxtWarning	0: warning txt messages disabled, otherwise enabled
TxtNotice	0: notice txt messages disabled, otherwise enabled.
TxtDebug	0: debug txt messages disabled, otherwise enabled.
TxtTest	0: test txt messages disabled, otherwise enabled.

Return value(s):

ErrorCode	0 if no error occurs.
-----------	-----------------------

Corresponding C++ command:

```
-/-
```

B.19 TSMWGPSSendUBX_CFG_MSG**Description:**

Send UBX command CFG_MSG to the TSMW with the given TSMW-ID. Returns zero if successful, otherwise error code.

Command syntax:

```
[ErrorCode] = TSMWGPSSendUBX_CFG_MSG(TSMWID, Mnemonic, Enable)
```

Parameter(s):

TSMWID	ID of the TSMW
Mnemonic	Mnemonic of the message type. Mnemonic supported: 'UBX,00', 'UBX,03', 'UBX,04', 'DTM', 'GBS', 'GGA', 'GLL', 'GRS', 'GSA', 'GST', 'GSV', 'RMC', 'TXT', 'VTG', 'ZDA'

Enable 0: disable message, otherwise enable message.

Return value(s):

ErrorCode 0 if no error occurs.

Corresponding C++ command:

-/-

B.20 TSMWGPSSendUBX_CFG_PRT_AllIn_NMEAOut**Description:**

Send UBX command CFG_PRT to the TSMW with the given TSMW-ID. The configuration of the GPS receiver is forced to support all protocol as input, and only NMEA as output on UART 1. Returns zero if successful, otherwise error code.

Command syntax:

```
[ErrorCode] = TSMWGPSSendUBX_CFG_PRT_AllIn_NMEAOut (TSMWID)
```

Parameter(s):

TSMWID ID of the TSMW

Return value(s):

ErrorCode 0 if no error occurs.

Corresponding C++ command:

-/-

B.21 TSMWGPSSendUBX_CFG_RATE**Description:**

Send UBX command CFG_RATE to the TSMW with the given TSMW-ID. Reference time is forced to UTC. Returns zero if successful, otherwise error code.

Command syntax:

```
[ErrorCode] = TSMWGPSSendUBX_CFG_RATE (TSMWID,
UpdateRateString)
```

Parameter(s):

TSMWID ID of the TSMW.

UpdateRateString GPS measurement output rate, supported values:
'200 ms', '250 ms', '500 ms', '1 s', '2 s', '4 s', '8 s'.

Return value(s):

ErrorCode 0 if no error occurs.

Corresponding C++ command:

-/-

B.22 TSMWGPSSendUBXCmd(TSMWID, UBXCmd)**Description:**

Send UBX command to the TSMW with the given TSMW-ID.
Returns zero if successful, otherwise error code.

Command syntax:

```
[ErrorCode] = TSMWGPSSendUBXCmd(TSMWID, UBXCmd)
```

Parameter(s):

TSMWID	ID of the TSMW.
UBXCmd	UBX command, converted into an array of 2 hexadecimal characters. E.g.: to set CFG_DAT to WGS84: ['B5';'62';'06';'06';'02'; '00';'00';'0E';'4A'].

Return value(s):

ErrorCode 0 if no error occurs.

For further information see TSMW Matlab IQ Interface Documentary.

Corresponding C++ command:

```
int ErrorCode = TSMWGPSSendUBXCmd_c(int TSMWID, void*
pUBXCmd, unsigned int UBXMsgLength)
```

B.23 TSMWGPSStartTcpServer**Description:**

Start server forwarding NMEA messages from the TSMW with the given TSMW-ID over LAN. Returns zero if successful, otherwise error code.

Command syntax:

```
[ErrorCode] = TSMWGPSSstartTcpServer(TSMWID, IpPort)
```

Parameter(s):

TSMWID	ID of the TSMW.
IpPort	Port used by the TCP server.

Return value(s):

ErrorCode 0 if no error occurs.

For further information see TSMW Matlab IQ Interface Documentary.

Corresponding C++ command:

```
Int ErrorCode = TSMWGPSSstartTcpServer_c(unsigned short  
TSMWID, unsigned short IpPort)
```

B.24 TSMWGPSSstopServer

Description:

Stop server forwarding NMEA messages from the TSMW with the given TSMW-ID over LAN. Returns zero if successful, otherwise error code.

Command syntax:

```
[ErrorCode] = TSMWGPSSstopServer(TSMWID)
```

Parameter(s):

TSMWID	ID of the TSMW.
--------	-----------------

For further information see TSMW Matlab IQ Interface Documentary.

Return value(s):

ErrorCode 0 if no error occurs.

Corresponding C++ command:

```
Int ErrorCode = TSMWGPSSstopServer_c(unsigned short TSMWID)
```

B.25 TSMWGPSSync

Description:

The function enables or disables GPS synchronization. GPS synchronization is enabled by default.

Command syntax:

```
[ErrorCode] = TSMWGPSSync (TSMWID, Enable);
```

Parameter(s):

TSMWID		Specifies the TSMW ID.
Enable	1	Enable GPS synchronization.
	0	Disable GPS synchronization.

Return value(s):

ErrorCode 0 if successful. Otherwise the error code is returned.

Corresponding C++ command:

```
int ErrorCode = TSMWGPSSync_c(unsigned short TSMWID,
                                int Enable);
```

B.26 TSMWInitInterface

Description:

Load the R&S TSMW IQ interface library `TSMWInterface.dll`. The function `TSMWInitInterface` has to be called once before any other action can be performed on the TSMW.

Command syntax:

```
[ErrorText, ErrorCode] = TSMWInitInterface;
```

Parameter(s):

-/-

Return value(s):

ErrorText If no error occurs an empty string is returned. Otherwise the message text of the error is returned.

ErrorCode 0 if successful. Otherwise the error code is returned.

Corresponding C++ command:

```
int ErrorCode = TSMWInitInterface_c();
```

B.27 TSMWIQCloseStreamFile**Description:**

The function closes the stream data file.

Command syntax:

```
[ErrorCode] = TSMWIQCloseStreamFile(StreamID);
```

Parameter(s):

StreamID Stream ID this file was associated with.

Return value(s):

ErrorCode 0 if successful. Otherwise the error code is returned.

Corresponding C++ command:

```
int ErrorCode = TSMWIQStopStreaming_c (
    unsigned short TSMWID,
    unsigned char StreamID,
    TSMW_IQIF_STREAM_STATUS_t *pStreamStatus);
```

B.28 TSMWIQDataAvailable**Description:**

The function returns the number of the available measurement results. The value zero is returned when no data is available.

Command syntax:

```
[ErrorCode, NoOfDataBlocks] = TSMWIQDataAvailable;
```

Parameter(s):

-/-

Return value(s):

ErrorCode 0 if successful. Otherwise the error code is returned.

NoOfDataBlocks Specifies the number of available data blocks.

Corresponding C++ command:

```
int ErrorCode = TSMWIQDataAvailable_c(long *pNoOfIQResults);
```

B.29 TSMWIQGetData**Description:**

This function retrieves I/Q data in complex double array format.

**Delete measurement result**

The function will delete the corresponding measurement result on the R&S TSMW.

Command syntax:

```
[ErrorCode, IQResultParam,
IQData, Overflow,
Calibrated] = TSMWIQGetData (TSMWID, MeasRequestID, TimeOut,
NoOfSamples, NoOfChannels);
```

Parameter(s):

TSMWID	Defines the used R&S TSMW.
MeasRequestID	Specifies the measurement request ID of the measurement result to wait for. If 0, get parameters of next available measurement data block (with lowest measurement request ID).
TimeOut	Defines the time in milliseconds to wait for the measurement response result.
NoOfSamples	Defines the number of expected samples. Only the specified number of samples will be copied from the measurement result.
	Note: The no. of requested samples has to be a multiple of 8.
NoOfChannels	Specifies the number of expected channels. This has to be equal to the total number of channels the corresponding measurement request covers. I.e.: Measuring only on frontend 1 with only 1 sub-channel means NoOfChannels = 1. Measuring on frontend 1 and 2 with 1 sub-channel active on frontend 1 and 2 sub-channels on frontend 2 mean NoOfChannels = 3.

Return value(s):

ErrorCode	0 if successful. Otherwise the error code is returned.
IQResultParam	Measurement result parameters. For information about the fields refer to chapter Structure: TSMW_IQ_Result on page 42.
IQData	Complex I/Q data array (already scaled with reference level 1dBm).
Overflow	Array of 1 x NoOfChannels The array contains for each channel the number of overflows that have occurred during the measurement
Calibrated	Array of 1 x NoOfChannels The array contains for each channel the calibration information. 0 if corresponding measurement setting is not calibrated. Otherwise 1.

Corresponding C++ command:

The parameters `Reserved1` and `Reserved2` have to be zero.

```
int ErrorCode = TSMW_IQ_GetDataDouble_c(
    unsigned short TSMWID,
    unsigned int MeasRequestID,
    unsigned int TimeOut,
    TSMW_IQ_RESULT *pIQResult,
    short* pReal, short* pImag, short* pScaling,
    unsigned long* pOvfl,
    unsigned int *pCalibrated,
    unsigned int NoOfSamples,
    unsigned int NoOfChannels,
    int Reserved1,int Reserved2);
```



Additional information for specific C++ return values:

pReal: Pointer to a short array with reserved elements for NoOfSamples x NoOfChannels. The first NoOfSamples data samples correspond to the first sub channel of the first frontend. The next NoOfSamples data samples correspond to the next sub channel (if applicable) of the first frontend etc..

pImag: Quadrature data array. Ordering as in pReal.

pScaling: Pointer to a short array with NoOfChannels reserved elements. The user has to reserve the space for the array, before. The array receives for each channel the reference level of the data in 1/100 dBm.

pOvfl: Pointer to a short array with NoOfChannels reserved elements. The user has to reserve the space for the array, before. The array receives for each channel the information if an overflow has occurred during the measurement. 0 if no overflow occurs. Otherwise 1.

pCalibration: Pointer to a short array with NoOfChannels reserved elements. The array receives for each channel the calibration information. 0 if corresponding measurement setting is not calibrated. Otherwise 1.

B.30 TSMWIQGetDataInt16

Description:

This function unpacks the measurement result into 16 bit integer array format.



Delete measurement result

The function will delete the corresponding measurement result.

Command syntax:

```
[ErrorCode,
IQResultParam,
IData, QData, Scaling,
Overflow, Calibrated] = TSMWIQGetDataInt16(TSMWID,
MeasRequestID, Timeout,
NoOfSamples, NoOfChannels);
```

Parameter(s):

TSMWID	Defines the used R&S TSMW.
MeasRequestID	Specifies the measurement request ID of the measurement result to wait for. If 0, get parameters of next available measurement data block (with lowest measurement request ID).
Timeout	Defines the time in milliseconds to wait for the measurement response result.
NoOfSamples	Defines the number of expected samples. Only the specified

number of samples will be copied from the measurement result.

Note:

For the functions TSMWIQGetDataInt16 and TSMWIQGetDataInt32, the number of requested samples has to be a multiple of 8. This is not the case for functions the deliver single or double precision data (e.g. TSMWIQGetDataDouble, TSMWIQGetStreamDouble, ...).

NoOfChannels Specifies the number of expected channels. This has to be equal to the total number of channels the corresponding measurement request covers.
I.e.: Measuring only on frontend 1 with only 1 sub-channel means NoOfChannels = 1. Measuring on frontend 1 and 2 with 1 sub-channel active on frontend 1 and 2 sub-channels on frontend 2 mean NoOfChannels = 3.

Return value(s):

ErrorCode	0 if successful. Otherwise the error code is returned.
IQResultParam	Measurement result parameters. For information about the fields refer to chapter Structure: TSMWIQResult on page 42 defined.
IData	NoOfSamples x NoOfChannels in-phase data array
QData	Quadrature data array. Ordering as in IData.
Scaling	Array of 1 x NoOfChannels The array contains for each channel the reference level of the data in 1/100 dBm.
Overflow	Array of 1 x NoOfChannels The array contains for each channel the number of overflows that have occurred during the measurement.
Calibrated	Array of 1 x NoOfChannels The array contains for each channel the calibration information. 0 if corresponding measurement setting is not calibrated. Otherwise 1.

Corresponding C++ command:



The parameters Reserved1 and Reserved2 have to be zero.

```
int ErrorCode = TSMWIQGetDataInt16_c(
    unsigned short TSMWID,
    unsigned int MeasRequestID,
    unsigned int TimeOut,
    TSMW_IQIF_RESULT_t *pIQResult,
    short* pReal, short* pImag, short* pScaling,
```

```
unsigned long* pOvfl,  
unsigned int *pCalibrated,  
unsigned int NoOfSamples,  
unsigned int NoOfChannels,  
int Reserved1, int Reserved2);
```

**Additional information for specific C++ return values:**

pReal: Pointer to a short array with reserved elements for `NoOfSamples x NoOfChannels`. The first `NoOfSamples` data samples correspond to the first sub channel of the first frontend. The next `NoOfSamples` data samples correspond to the next sub channel (if applicable) of the first frontend etc..

pImag: Quadrature data array. Ordering as in `pReal`.

pScaling: Pointer to a short array with `NoOfChannels` reserved elements. The user has to reserve the space for the array, before. The array receives for each channel the reference level of the data in 1/100 dBm.

pOvfl: Pointer to a short array with `NoOfChannels` reserved elements. The user has to reserve the space for the array, before. The array receives for each channel the information if an overflow has occurred during the measurement. 0 if no overflow occurs. Otherwise 1.

pCalibration: Pointer to a short array with `NoOfChannels` reserved elements. The array receives for each channel the calibration information. 0 if corresponding measurement setting is not calibrated. Otherwise 1.

B.31 TSMWIQGetDataInt32

Description:

This function unpacks the measurement result into 32 bit integer array format.



Delete measurement result

The function will delete the corresponding measurement result.

See also TSMWIQGetDataInt16.

Command syntax:

```
[ErrorCode,
IQResultParam,
IData,QData,Scaling,
Overflow,Calibrated] = TSMWIQGetDataInt32(TSMWID,
                                           MeasRequestID,Timeout,
                                           NoOfSamples,NoOfChannels);
```

Parameter(s):

See TSMWIQGetDataInt16.

Return value(s):

See TSMWIQGetDataInt16.

Corresponding C++ command:



The parameters `Reserved1` and `Reserved2` have to be zero.

```
int ErrorCode = TSMWIQGetDataInt32_c(
    unsigned short TSMWID,
    unsigned int MeasRequestID,
    unsigned int Timeout,
    TSMW_IQ_RESULT_t *pIQResult,
    short* pReal, short* pImag,short* pScaling,
    unsigned long* pOvfl,
    unsigned int *pCalibrated,
    unsigned int NoOfSamples,
    unsigned int NoOfChannels,
    int Reserved1, int Reserved2);
```

B.32 TSMWIQGetDataSingle

Description:

This function unpacks the measurement result into single precision floating point array format.



Delete measurement result

The function will delete the corresponding measurement result.

See also TSMWIQGetDataInt16.

Command syntax:

```
[ErrorCode,
IQResultParam,
IData, QData, Scaling,
Overflow, Calibrated] = TSMWIQGetDataSingle (TSMWID,
                                             MeasRequestID, Timeout,
                                             NoOfSamples, NoOfChannels);
```

Parameter(s):

See TSMWIQGetDataInt16.

Return value(s):

See TSMWIQGetDataInt16.

Corresponding C++ command:



The parameters `Reserved1` and `Reserved2` have to be zero.

```
int ErrorCode = TSMWIQGetDataSingle_c(
    unsigned short TSMWID,
    unsigned int MeasRequestID,
    unsigned int Timeout,
    TSMW_IQ_RESULT_t *pIQResult,
    short* pReal, short* pImag, short* pScaling,
    unsigned long* pOvfl,
    unsigned int *pCalibrated,
    unsigned int NoOfSamples,
    unsigned int NoOfChannels,
    int Reserved1, int Reserved2);
```

B.33 TSMWIQGetDataDouble

Description:

This function unpacks the streaming measurement result into double precision floating point array format.

**Delete measurement result**

The function will delete the corresponding measurement result.

See also TSMWIQGetDataInt16.

Command syntax:

```
[ErrorCode, IQResultParam,  
IData, QData, Scaling,  
Overflow, Calibrated] = TSMWIQGetDataDouble (TSMWID,  
MeasRequestID, TimeOut,  
NoOfSamples, NoOfChannels);
```

Parameter(s):

See TSMWIQGetDataInt16.

Return value(s):

See TSMWIQGetDataInt16.

Corresponding C++ command:

The parameters `Reserved1` and `Reserved2` have to be zero.

```
int ErrorCode = TSMWIQGetDataDouble_c(  
    unsigned short TSMWID,  
    unsigned int MeasRequestID,  
    unsigned int TimeOut,  
    TSMW_IQ_RESULT *pIQResult,  
    short* pReal, short* pImag, short* pScaling,  
    unsigned long* pOvfl,  
    unsigned int *pCalibrated,  
    unsigned int NoOfSamples,  
    unsigned int NoOfChannels,  
    int Reserved1, int Reserved2);
```

B.34 TSMWIQGetDataSingleIlv

Description:

This function unpacks the measurement result into interleaved single precision floating point array format.



Delete measurement result

The function will delete the corresponding measurement result.

See also TSMWIQGetDataInt16.

Command syntax:

```
[ErrorCode, IQResultParam,
IQData, Scaling, Overflow,
Calibrated] = TSMWIQGetDataSingle(TSMWID,
MeasRequestID, TimeOut,
NoOfSamples, NoOfChannels);
```

Parameter(s):

See TSMWIQGetDataInt16

Return value(s):

ErrorCode	See TSMWIQGetDataInt16.
IQResultParam	See TSMWIQGetDataInt16.
IQData	Complex I/Q data array.
Scaling	See TSMWIQGetDataInt16.
Overflow	See TSMWIQGetDataInt16.
Calibrated	See TSMWIQGetDataInt16.

Corresponding C++ command:



The parameters `Reserved1` and `Reserved2` have to be zero.

```
int ErrorCode = TSMWIQGetDataSingleIlv_c(
    unsigned short TSMWID,
    unsigned int MeasRequestID,
    unsigned int TimeOut,
    TSMW_IQ_RESULT_t *pIQResult,
    float* pCplxData, short* pScaling,
    unsigned long* pOvfl,
    unsigned int *pCalibrated,
    unsigned int NoOfSamples,
    unsigned int NoOfChannels,
    int Reserved1, int Reserved2);
```

B.35 TSMWIQGetDataDoubleIlv

Description:

This function unpacks the measurement result into interleaved double precision floating point array format. The corresponding measurement result will be deleted.



Delete measurement result

The function will delete the corresponding measurement result.

See also TSMWIQGetDataInt16.

Command syntax:

```
[ErrorCode, IQResultParam,
IQData, Scaling, Overflow,
Calibrated] = TSMWIQGetDataDouble (TSMWID, MeasRequestID,
                                     Timeout, NoOfSamples, NoOfChannels);
```

Parameter(s):

See TSMWIQGetDataInt16.

Return value(s):

ErrorCode	See TSMWIQGetDataInt16.
IQResultParam	See TSMWIQGetDataInt16.
IQData	See TSMWIQGetDataSingleIlv.
Scaling	See TSMWIQGetDataInt16.
Overflow	See TSMWIQGetDataInt16.
Calibrated	See TSMWIQGetDataInt16.

Corresponding C++ command:



The parameters `Reserved1` and `Reserved2` have to be zero.

```
int ErrorCode = TSMWIQGetDataDoubleIlv_c(
    unsigned short TSMWID, unsigned int MeasRequestID,
    unsigned int Timeout, TSMW_IQ_RESULT_t *pIQResult,
    double* pCplxData, short* pScaling,
    unsigned long* pOvfl, unsigned int *pCalibrated,
    unsigned int NoOfSamples,
    unsigned int NoOfChannels,
    int Reserved1, int Reserved2);
```

B.36 TSMWIQGetStreamSingle

Description:

This function unpacks the streaming measurement result into single precision floating point array format. The function works for both "online" streaming and "offline" streaming.

"Online" streaming means processing stream data on the fly, hence no automatic writing to a stream data file. During "online" streaming the memory for the stream data is released automatically.

"Offline" streaming means that data is recorded on the disc in a stream data file. After the streaming has been finished, the stream data file can be opened and data can be read from the file using this function.

Command syntax:

```
[ErrorCode,
 IQResultParam,
 IData, QData, Scaling, Overflow,
 Calibrated] = TSMWIQGetDataStreamSingle(StreamID, Timeout
 Offset, NoOfSamples, NoOfChannels);
```

Parameter(s):

StreamID	Stream ID that was specified with function Fehler! Ungültiger Eigenverweis auf Textmarke. on page 98.
TimeOut	Defines the time in milliseconds to wait for the measurement response result.
Offset	Sample offset where to get I/Q data. Only used for offline streaming (reading from a stream data file).
NoOfSamples	Defines the number of expected samples. This has to be equal to the number of samples specified in the measurement request (TSMWIQStream).
NoOfChannels	Specifies the number of expected channels. This has to be equal to the total number of channels the corresponding measurement request covers. I.e.: Measuring only on frontend 1 with only 1 sub-channel means NoOfChannels = 1. Measuring on frontend 1 and 2 with 1 sub-channel active on frontend 1 and 2 sub-channels on frontend 2 mean NoOfChannels = 3.

Return value(s):

See TSMWIQGetDataInt16.

Corresponding C++ command:

```
int ErrorCode = TSMWIQGetStreamSingle_c(
    unsigned char StreamID,
    unsigned int Timeout,
    TSMW_IQ_RESULT_t *pIQResult,
    float* pReal, float* pImag,
    short* pScaling, unsigned long* pOvfl,
    unsigned int *pCalibrated,
    unsigned __int64 Offset
    unsigned int NoOfSamples,
    unsigned int NoOfChannels);
```

B.37 TSMWIQGetStreamDouble**Description:**

This function unpacks the streaming measurement result into double precision floating point array format. The function works for both "online" streaming and "offline" streaming. For more details about "online" and "offline" streaming see description of TSMWIQGetStreamSingle on p. 98.

Command syntax:

```
[ErrorCode, IQResultParam,
IQData, Scaling, Overflow
Calibrated]= TSMWIQGetStreamDouble(StreamID, Timeout,
    Offset, NoOfSamples, NoOfChannels);
```

Parameter(s):

See TSMWIQGetStreamSingle.

Return value(s):

See TSMWIQGetStreamSingle.

Corresponding C++ command:

```
int ErrorCode = TSMWIQGetStreamDouble_c(
    unsigned char StreamID,
    unsigned int Timeout,
    TSMW_IQ_RESULT_t *pIQResult,
    double* pReal,
    double* pImag,
    short* pScaling,
    unsigned long* pOvfl,
    unsigned int *pCalibrated,
    unsigned __int64 Offset
    unsigned int NoOfSamples,
    unsigned int NoOfChannels);
```

B.38 TSMWIQGetStreamSingleIv

Description:

This function unpacks the streaming measurement result into interleaved single precision floating point array format.

The function works for both "online" streaming and "offline" streaming. For more details about "online" and "offline" streaming see description of TSMWIQGetStreamSingle on p. 98.

Command syntax:

```
[ErrorCode, IQResultParam,
IQData, Scaling, Overflow,
Calibrated]=TSMWIQGetDataStreamSingle(StreamID, TimeOut,
Offset, NoOfSamples, NoOfChannels);
```

Parameter(s):

StreamID	Stream ID that was specified with function Fehler! Ungültiger Eigenverweis auf Textmarke. on page 98.
TimeOut	Max. time in ms to wait for the measurement data.
Offset	Sample offset where to get I/Q data. Only used for offline streaming (reading from a stream data file).
NoOfSamples	Defines the number of expected samples. This has to be equal to the number of samples specified in the measurement request (TSMWIQStream).
NoOfChannels	Specifies the number of expected channels. This has to be equal to the total number of channels the corresponding measurement request covers. I.e.: Measuring only on frontend 1 with only 1 sub-channel means NoOfChannels = 1. Measuring on frontend 1 and 2 with 1 sub-channel active on frontend 1 and 2 sub-channels on frontend 2 mean NoOfChannels = 3.

Return value(s):

ErrorCode	See TSMWIQGetDataInt16.
IQResultParam	See TSMWIQGetDataInt16.
IQData	Complex I/Q data array.
Scaling	See TSMWIQGetDataInt16.
Overflow	See TSMWIQGetDataInt16.
Calibrated	See TSMWIQGetDataInt16.

Corresponding C++ command:

```
int ErrorCode = TSMWIQGetStreamSingleIlv_c(
    unsigned char StreamID,
    unsigned int Timeout,
    TSMW_IQ_RESULT_t *pIQResult,
    float* pCplxData,
    short* pScaling,
    unsigned long* pOvfl,
    unsigned int *pCalibrated,
    unsigned __int64 Offset,
    unsigned int NoOfSamples,
    unsigned int NoOfChannels);
```

B.39 TSMWIQGetStreamDoubleIlv**Description:**

This function unpacks the streaming measurement result into interleaved double precision floating point array format.

The function works for both "online" streaming and "offline" streaming. For more details about "online" and "offline" streaming see description of TSMWIQGetStreamSingle on page 98.

Command syntax:

```
[ErrorCode, IQResultParam,
IQData, Scaling, Overflow,
Calibrated]= TSMWIQGetStreamDouble(StreamID, Timeout, Offset,
    NoOfSamples, NoOfChannels);
```

Parameter(s):

See TSMWIQGetStreamSingle.

Return value(s):

See TSMWIQGetStreamSingle.

Corresponding C++ command:

```
int ErrorCode = TSMWIQGetStreamDoubleIlv_c(
    unsigned char StreamID,
    unsigned int Timeout,
    TSMW_IQ_RESULT *pIQResult,
    double* pCplxData, short* pScaling,
    unsigned long* pOvfl,
    unsigned int *pCalibrated,
    unsigned __int64 Offset,
    unsigned int NoOfSamples,
    unsigned int NoOfChannels);
```

B.40 TSMWIQGetResultParam

Description:

This function returns the parameters of a measurement result.



Do not delete measurement result

The function will not delete the measurement result. This means that it can be called before one of the TSMWIQGetDataXXX functions is called. The "Get" functions, however, will delete the measurement result.

Command syntax:

```
[ErrorCode, IQResultParam] = TSMWIQDGetResultParam(
    MeasRequestID, Timeout);
```

Parameter(s):

MeasRequestID	Specifies the measurement request ID of the measurement result to wait for. If 0, get parameters of next available measurement data block (with lowest measurement request ID).
Timeout	Defines the time in milliseconds to wait for the measurement result.

Return value(s):

ErrorCode	0 if successful. Otherwise the error code is returned.
IQResultParam	Measurement result parameters. For information about the fields refer to chapter Structure: TSMWIQResult on page 42.

Corresponding C++ command:

```
int ErrorCode = TSMWIQGetResultParam_c(
    unsigned int MeasRequestID,
    unsigned int Timeout,
    TSMW_IQIF_RESULT_t *pIQResult);
```

B.41 TSMWIQMeasure

Description:

The function starts a new measurement with given parameters. The measurement parameters are defined in the structure `MeasCtrl`.

Measurements are scheduled on the R&S TSMW. The measurement specification includes a priority and might also include (several) start times at which the measurement can be started. The R&S TSMW schedules automatically the measurement request with the highest priority and the earliest start time. Start times given by the StartTimes parameter are exclusive, i.e. only a single measurement is scheduled but any of the given start times can be used. If no start times are given, the measurement is scheduled as early as possible.

Extend measurement

Via the measurement control structure it is possible to start single measurement or streaming measurement. Furthermore following attributes of the measurement can be set:

- schedules the measurement
- defines the measurement as periodic task
- define a trigger when to start and stop the measurement.

Command syntax:

```
[ErrorCode, MeasRequestID] = TSMWMeasure(TSMWID,
                                         StartTimes, MeasCtrl);
```

Parameter(s):

TSMWID	Specifies the R&S TSMW to use for measurement.
StartTimes	<p>Defines one or more start time(s) at which the measurement can be started. It is a multiple of 1/(395e6 / 18) seconds (approximately 45ns) and relative to start-up of the R&S TSMW.</p> <p>If the parameter is empty, the measurement starts as soon as possible.</p> <p>For the current I/Q time on the R&S TSMW an estimate can be obtained by calling TSMWGetIQTime.</p>
MeasCtrl	<p>Measurement control structure. A template of the measurement control structure is provided by the MATLAB scripts "MeasCtrlTemplate", "MeasCtrlTemplate_RF1" and "MeasCtrlTemplate_RF2".</p> <p>If the field "ChannelCtrl1" is present in the MeasCtrl structure, a measurement on RF-channel 1 is performed. If the field "ChannelCtrl2" is present, a measurement on RF-channel 2 is performed. A measurement on both channels is performed if both fields are present.</p> <p>For further information about the fields refer to chapter Structure: MeasCtrl on page 30 and provided MATLAB example application.</p>

Return value(s):

ErrorCode	0 if successful. Otherwise the error code is returned.
MeasRequestID	ID for this measurement request.

Corresponding C++ command:

```

int ErrorCode = TSMWIQMeasure_c(
    unsigned short TSMWID,
    unsigned long *pMeasRequestID,
    unsigned __int64 *pStartTimes,
    long NoOfStartTimes,
    TSMW_IQIF_MEAS_CTRL_t *pMEAS_CTRL,
    TSMW_IQIF_CH_CTRL_t *pCHANNEL_CTRL1,
    TSMW_IQIF_CH_CTRL_t *pCHANNEL_CTRL2 );

int ErrorCode = TSMWIQMeasureExt_c (
    unsigned short TSMWID,
    unsigned long *pMeasRequestID,
    unsigned __int64 *pStartTimes,
    long NoOfStartTimes,
    TSMW_IQIF_MEAS_CTRL_t *pMEAS_CTRL,
    TSMW_IQIF_CH_CTRL_t *pCHANNEL_CTRL1,
    TSMW_IQIF_CH_CTRL_t *pCHANNEL_CTRL2,
    TSMW_IQIF_SCHEDULE_CTRL_t *pSchedParam,
    TSMW_IQIF_PERIOD_CTRL_t *pPeriodParam );

int ErrorCode = TSMWIQMeasureTrig_c (
    unsigned short TSMWID,
    unsigned long *pMeasRequestID,
    unsigned long long *pStartTimes,
    long NoOfStartTimes,
    TSMW_IQIF_MEAS_CTRL *pMEAS_CTRL,
    TSMW_IQIF_CH_CTRL *pCHANNEL_CTRL1,
    TSMW_IQIF_CH_CTRL *pCHANNEL_CTRL2,
    TSMW_IQIF_TRIG_CTRL_t *pTriggerParam );

```

B.42 TSMWIQOpenStreamFile**Description:**

The function opens a stream data file.

Command syntax:

```
[ErrorCode] = TSMWIQOpenStreamFile (FileName, StreamID);
```

Parameter(s):

FileName	Specifies the stream data file to open
StreamID	Stream ID this file was associated with.

Return value(s):

ErrorCode	0 if successful. Otherwise the error code is returned.
-----------	--

Corresponding C++ command:

```

int ErrorCode = TSMWIQOpenStreamFile_c(
    char *pFileName, unsigned char StreamID,

```

```
TSMW_IQIF_STREAM_INFO_t *pStreamInfo,
TSMW_IQIF_MEAS_CTRL_t *pMeasCtrl,
TSMW_IQIF_CH_CTRL_t *pChannelCtrl1,
TSMW_IQIF_CH_CTRL_t *pChannelCtrl2,
TSMW_IQIF_FILTER_PARAM_t *pFilterSpec,
long *pCoeff, unsigned long NoOfCoeffs);
```

B.43 TSMWIQPeriodCtrl

Description:

The function set and modify the parameters of a periodic measurement request.

Command syntax:

```
[ErrorCode] = TSMWIQPeriodCtrl(TSMWID, PeriodCtrl);
```

Parameter(s):

TSMWID	Specifies the R&S TSMW to use.
PeriodCtrl	Control structure of the periodic measurement requests. For further information about the fields refer to chapter Structure: MeasCtrl on page 30.

Return value(s):

ErrorCode 0 if successful. Otherwise the error code is returned.

Corresponding C++ command:

```
int ErrorCode = TSMWIQPeriodCtrl_c (
    unsigned short TSMWID,
    TSMW_IQIF_PERIOD_CTRL_t *pParam);
```

B.44 TSMWIQSetup

Description:

Transmit a filter specification to the R&S TSMW. The field `FilterID` in the structure `FilterSpec` contains the ID of the filter.



Overwrite an existing filter with the same ID

To overwrite an existing filter with a new transmitted filter with the same ID, the previous filter specification has to be used with the same application with the same IP connection to the R&S TSMW.



Resampling filter

In order to perform resampling on the R&S TSMW, an appropriate resampling filter has to be transmitted to the R&S TSMW. For filter design you might use the provided R&S TSMW Filter Design Tool.

Command syntax:

```
[ErrorCode] = TSMWSetup(TSMWID, FilterSpec);
```

Parameter(s):

<code>TSMWID</code>	Specifies the R&S TSMW ID to use for measurement.
<code>FilterSpec</code>	Filter specification structure. For a detailed description of the structure refer to chapter Structure: FilterSpec on page 40 and to Digital I/Q Interface Status As described in chapter 8.1 Main Window "TSMW Matlab IQ Interface", various messages can appear in this window during a measurement. Please see the section on Digital I/Q Interface Status in chapter 8.1 for a description of the messages. Filter Design Dialog on page 54.

Return value(s):

<code>ErrorCode</code>	0 if successful. Otherwise the error code is returned.
------------------------	--

Corresponding C++ command:

```
[ErrorCode] = TSMWIQSetup_c(unsigned short TSMWID,  
                             TSMW_IQIF_FILTER_PARAM_t *pFilterParam,  
                             long *plCoeff);
```

B.45 TSMWIQStream

Description:

The function starts a streaming measurement with given parameters. The measurement parameters are defined in the structure `MeasCtrl`.

Command syntax:

```
[ErrorCode] = TSMWIQStream(TSMWID,  
                           MeasCtrl, FileName, CreateIfExists);
```

Parameter(s):

TSMWID	Specifies the instrument to use for measurement.
MeasData	<p>Measurement control structure. A template of the measurement control structure is provided by the MATLAB scripts "MeasCtrlTemplate", "MeasCtrlTemplate_RF1" and "MeasCtrlTemplate_RF2".</p> <p>If the field "ChannelCtrl1" is present in the MeasCtrl structure, a streaming measurement on RF-channel 1 is performed. If the field "ChannelCtrl2" is present, a streaming measurement on RF-channel 2 is performed. A streaming measurement on both channels is performed if both fields are present.</p> <p>For further information about the fields refer to chapter Structure: MeasCtrl on page 30 and provided MATLAB example scripts.</p>
FileName	<p>Specifies the file where the streaming data has to be saved. If no file name is given, online-processing of the streaming data is enabled.</p> <p>Note: Online-processing of streaming data includes unpacking the compressed stream data which needs additional processor performance and increases the amount of data. Highest streaming performance can be achieved when streaming to hard disk.</p>
CreateIf Exists	<p>Specifies whether an already existing streaming data file will be overwritten.</p> <p>1: If a stream data file with the given file name already exists, it will be overwritten, if not a new file will be created.</p> <p>0: If a stream data file with the given file name already exists, the function returns an error code.</p>

Return value(s):

ErrorCode	0 if successful. Otherwise the error code is returned.
-----------	--

Corresponding C++ command:

```
int ErrorCode = TSMWIQStream_c(
    unsigned short TSMWID,
    TSMW_IQIF_MEAS_CTRL *pMEAS_CTRL,
    TSMW_IQIF_CH_CTRL *pCHANNEL_CTRL1,
    TSMW_IQIF_CH_CTRL *pCHANNEL_CTRL2,
    TSMW_IQIF_STREAM_CTRL_t *pStreamCtrl,
    char *pFileName, char *pDescription,
    unsigned int Flags);
```

B.46 TSMWIQStreamStatus

Description:

The function returns current streaming status.

Command syntax:

```
ErrorCode, StreamStatus] = TSMWIQStreamStatus (StreamID);
```

Parameter(s):

StreamID Stream ID this file was associated with.

Return value(s):

ErrorCode 0 if successful. Otherwise the error code is returned.

StreamStatus Streaming status structure.

For further information about the fields refer to chapter
Structure: StreamStatus on page 43 and provided MATLAB
example scripts

Corresponding C++ command:

```
int ErrorCode = TSMWIQStopStreaming_c (  
    unsigned short TSMWID,  
    unsigned char StreamID,  
    TSMW_IQIF_STREAM_STATUS_t *pStreamStatus);
```

B.47 TSMWIQStopStreaming

Description:

The function stops a streaming measurement.

Command syntax:

```
[ErrorCode, StreamStatus] = TSMWIQStopStreaming(  
TSMWID, StreamID);
```

Parameter(s):

TSMWID ID of R&S TSMW to use for streaming measurement.

StreamID

Stream ID that was specified with function TSMWIQGetStreamSingle

Description:

This function unpacks the streaming measurement result into single precision floating point array format. The function works for both "online" streaming and "offline" streaming.

"Online" streaming means processing stream data on the fly, hence no automatic writing to a stream data file. During "online" streaming the memory for the stream data is released automatically.

"Offline" streaming means that data is recorded on the disc in a stream data file. After the streaming has been finished, the stream data file can be opened and data can be read from the file using this function.

Command syntax:

```
[ErrorCode,
IQResultParam,
IData,QData,Scaling,Overflow,
Calibrated] = TSMWIQGetDataStreamSingle(StreamID,
Offset,NoOfSamples,NoOfC
```

Parameter(s):

StreamID	Stream ID that was specified with function Fehler Ungültiger Eigenverweis auf Textmarke. on page 110
TimeOut	Defines the time in milliseconds to wait for the measurement response result.
Offset	Sample offset where to get I/Q data. Only used for "offline" streaming (reading from a stream data file).
NoOfSamples	Defines the number of expected samples. This has to be equal to the number of samples specified in the measurement request (TSMWIQStream).
NoOfChannels	Specifies the number of expected channels. This has to be equal to the total number of channels the corresponding measurement request covers. I.e.: Measuring only on frontend 1 with only 1 sub-channel active means NoOfChannels = 1. Measuring on frontend 1 with 1 sub-channel active on frontend 1 and 2 sub-channels on frontend 2 mean NoOfChannels = 3.

Return value(s):

See TSMWIQGetDataInt16.

Corresponding C++ command:

```
int ErrorCode = TSMWIQGetStreamSingle_c(
    unsigned char StreamID,
    unsigned int TimeOut,
    TSMW_IQ_RESULT_t *pIQResult,
    float* pReal, float* pImag,
    short* pScaling, unsigned long*
    unsigned int *pCalibrated,
    unsigned __int64 Offset
```

Return value(s):

<code>ErrorCode</code>	0 if successful. Otherwise the error code is returned.
<code>StreamStatus</code>	Streaming status after streaming has stopped.

For further information about the fields refer to chapter Structure: StreamStatus on page 43 and provided MATLAB example scripts.

Corresponding C++ command:

```
int ErrorCode = TSMWIQStopStreaming_c (
    unsigned short TSMWID,
    unsigned char StreamID,
    TSMW_IQIF_STREAM_STATUS_t *pStreamStatus);
```

B.56 TSMWReleaseInterface**Description:**

Unload the external R&S TSMW IQ interface library and disconnect from R&S TSMW(s).

**MATLAB Interface Library**

To secure your MATLAB application from crash you have to call the function TSMWReleaseInterface before you can close the application.

Command syntax:

```
[ErrorCode] = TSMWReleaseInterface;
```

Parameter(s):

```
-/-
```

Return value(s):

<code>ErrorCode</code>	0 if successful. Otherwise the error code is returned.
------------------------	--

Corresponding C++ command:

```
int ErrorCode = TSMWReleaseInterface_c();
```

B.57 TSMWResourceRequest

Description:

The function requests a receiver resource for a measurement task. Therefore enter the resourceID of the resource. The value will be entered automatically into the m-structure field `MeasCtrlSchedCtrl`.

If too many requests are sent to the same resource, the system schedule all waiting measurement tasks automatically. The schedule algorithm manages the access to the resource according to the priority number (m-structure field `MeasCtrl.MeasCtrl.Priority`).

The given priority number increase dynamically according to the waiting time (observation time).

If a measurement task does not have a priority value at creation time, the system will attach the lowest possible priority value to it => 0.

Command syntax:

```
[ErrorCode, ResourceID] = TSMWResourceRequest(TSMWID,
                                             IsRequest, ResourceID, ResourceParam);
```

Parameter(s):

<code>TSMWID</code>	Specifies the R&S TSMW to use.
<code>IsRequest</code>	1: Request receiver resource. 0: Release receiver resource.
<code>ResourceID</code>	Only relevant for <code>IsRequest == 0</code> . Resource ID of receiver resource that shall be released.
<code>ResourceParam</code>	Resource parameter structure. For further information about the fields refer to chapter Structure: <code>MeasCtrl</code> on page 30.

Return value(s):

<code>ErrorCode</code>	0 if successful. Otherwise the error code is returned.
<code>ResourceID</code>	Only relevant for <code>IsRequest == 0</code> Returns the resource ID if resource was granted.

Corresponding C++ command:

```
int ErrorCode = TSMWIQResourceRequest_c (
    unsigned short TSMWID,
    unsigned long IsRequest,
    unsigned long *pResourceID,
    TSMW_IQIF_RESOURCE_PARAM_t *pParam);
```

B.58 TSMWSetDAC

Description:

The function set the R&S TSMW references oscillator DAC (Digital-to-Analog Converter) value for fine adjustment of the internal reference oscillator.



GPS compatibility

This function only works when GPS synchronization is disabled.

Command syntax:

```
[ErrorCode] = TSMWSetDAC (TSMWID, Value);
```

Parameter(s):

TSMWID	Specifies the R&S TSMW which shall be used for streaming.
Value	Specifies DAC value. Value range: 0..1023. Reasonable values are in the range 200..800.

Return value(s):

ErrorCode	0 if successful. Otherwise the error code is returned.
-----------	--

Corresponding C++ command:

```
int ErrorCode = TSMWSetDAC_c(unsigned short TSMWID,  
                             short Value);
```

B.59 TSMWShutdown

Description:

The function shuts the specified R&S TSMW down.

Command syntax:

```
[ErrorCode] = TSMWShutdown (TSMWID);
```

Parameter(s):

TSMWID	Defines the R&S TSMW to shut down.
--------	------------------------------------

Return value(s):

ErrorCode	0 if successful. Otherwise the error code is returned.
-----------	--

Corresponding C++ command:

```
int ErrorCode = TSMWShutdown_c(unsigned short TSMWID);
```

B.60 TSMWTGEnable**Description:**

The function enables the tracking (test) generator output.

Command syntax:

```
[ErrorCode] = TSMWTGEnable(TSMWID, Enable);
```

Parameter(s):

TSMWID	Defines the used R&S TSMW.
Enable	1: Enable test output 0: Disable test output

Return value(s):

ErrorCode 0 if successful. Otherwise the error code is returned.

Corresponding C++ command:

```
int ErrorCode = TSMWTGEnable_c(unsigned short TSMWID,  
                               unsigned int Frontends);
```

B.61 TSMWTrigger**Description:**

The function set and read R&S TSMW trigger I/O data.

Command syntax:

```
[ErrorCode, TriggerIn1,  
   TriggerIn2] = TSMWTrigger(TSMWID, TriggerLine, Out,  
                             TriggerValue1, TriggerValue2);
```

Parameter(s):

TSMWID	Specifies the instrument ID to use for measurement.
TriggerLine	Set or get data for different trigger lines: 1: For trigger line 1. 2: For trigger line 2.

	3: For trigger line 1 and 2.
Out	Set the corresponding trigger line: 0: Set to 'input'. 1: Set to 'output'.
TriggerValue1	If trigger line 1 is used as output 0: set to low 1: set to high.
TriggerValue2	If trigger line 2 is used as output 0: set to low 1: set to high.

Return value(s):

ErrorCode	0 if successful. Otherwise the error code is returned.
TriggerIn1	Returns the current value of trigger line 1 (only if it is if used as input.) 0: low 1: high
TriggerIn2	Returns the current value of trigger line 1 (only if it is if used as input.) 0: low 1: high.

Corresponding C++ command:

```
int ErrorCode = TSMWTrigger_c(unsigned short TSMWID,
                              int TriggerLine,int Out,int *pTrig1Value,
                              int *pTrig2Value);
```

B.62 TSMWINFOIFEnable**Description:**

Function to subscribe/unsubscribe to the Digital I/Q Interface status messages.

Command syntax:

```
[ErrorCode]= TSMWINFOIFEnable(TSMWID, Enable)
```

Parameter(s):

TSMWID	Specifies the instrument ID to use for measurement.
Enable	1: Enable subscription

0: Disable subscription

Return value(s):

ErrorCode 0 if successful. Otherwise the error code is returned.

Corresponding C++ command:

```
int ErrorCode = TSMWINFOIFEnable_c (unsigned short TSMWID,
                                     int Enable);
```

B.63 TSMWINFOIFGetStatus

Description:

Function to read the latest status message from the Digital I/Q Interface.

Command syntax:

```
[ StatusTxt
  , ConnectionState
  , RxMultiChanSupport
  , MultiChanMode      ] = TSMWINFOIFGetStatus()
```

Parameter(s):

Return value(s):

StatusTxt	Status string containing the information received from the device connected
ConnectionState	-2: Unknown/undefined mode. -1: Connection error. 0: Cable not connected. 1: Connection in progress. 2: Reserved for later use. 3: Connection successful.
RxMultiChanSupport	0: Target device supports only 1 channel or multiple channel mode is not compatible with TSMW. 1: Target device supports a compatible multiple channel mode.
MultiChanMode	0: Digital I/Q Interface is operating in single channel mode. 1: Digital I/Q Interface is operating in multiple channels mode.

Corresponding C++ command:

```
char* StatusTxt = TSMWINFOIFGetStatus_c (int
*pConnectionState, int *pRxChanMuxSupport, int
*pChanMuxMode);
```


Index

ARP-Cache.....	67	Offline streaming.....	96, 108
Block diagram R&S TSMW	15	Online streaming.....	96, 108
Calculate bit rate.....	10	R&S TSMW	5
Connection Settings (GUI)	44	R&S TSMW-B1.....	7
Continuous Measure (GUI)	48	R&S TSMW-K1.....	7
Conventions.....	6	Receiver Management.....	19
Create filter	52	Related documentation	5
Customize GUI	46, 48, 49, 52, 104, 108	resampling filter	58
Demo application	43	Scheduling.....	19
Example application		Signal processing	15
C++	8, 66	Single Measure (GUI)	47
MATLAB	8	Source code	
Example TSMW	43	example files.....	8
Export		MATLAB files	43
*.mat.....	64, 65	Streaming Measure (GUI)	45
*.wav	51	Streaming measurement	
MATLAB	64, 65	Analyze data.....	64
Measured data (GUI)	48	GUI.....	50
Streamed data (GUI).....	51	Offline.....	96, 108
Filter Design	52	Online.....	96, 108
Filter Design (GUI).....	46	Record.....	63
Functions	21	Structures	27
GPS.....	20	System requirements	9
GPS Configuration.....	45	Trigger	20
Installation path.....	12	TSMW IQ Interface window	43
Interface.....	7	TSMW IQ StreamingTool	50
MATLAB Component Runtime	11	Uninstall.....	14
MATLAB functions	21	Use cases.....	57
MATLAB path environment	13	VC Redistributable.....	13
MCR	11	Waveformat	51
M-functions	21		