

# ERA-GLONASS コンフォーマンス・ テスト

## Application Note

### Products:

- R&S®CMW500
- R&S®CMW-KA094
- R&S®CMW-KA095
- R&S®SMBV100A



緊急ロード・アシスタンス (ERA-GLONASS) は、道路上で生じた事故またはその他の緊急事態の際の応答時間を短縮させることを目的としたロシアで提供が開始されるサービスです。本アプリケーションノートでは、ERA-GLONASS の背景にある技術を簡単に説明し、広帯域無線機テスタ R&S®CMW500 とベクトル・シグナル信号発生器 R&S®SMBV100A を使用したソリューションを紹介します。ERA-GLONASS のテストソフトウェアは、GSM または WCDMA 無線通信規格を使用したテストの実行を迅速かつ簡単に実施することができます。

### Note:

eCall コンフォーマンス・テストにつきましては、アプリケーションノート 1MA241 に記載されています。アプリケーションノートの最新版は以下の HP アドレスよりご参照ください。([www.rohde-schwarz.com/appnote/1MA251](http://www.rohde-schwarz.com/appnote/1MA251))

ERA\_GLONASS — 1MA251\_0j

## 目次

1	はじめに.....	3
2	ERA-GLONASS とは.....	6
3	IVS コンフォーマンス・テスト.....	14
4	Appendix.....	50
5	Rohde & Schwarz.....	57

# 1 はじめに

ERA-GLONASS は、道路上で生じた事故またはその他緊急事態の際の応答時間を短縮させることを目的としたロシアで提供が開始されるサービスです。ロシア政府は、今後すべての新しい自動車に ERA モジュールを搭載させる法律を制定しました。ERA は、欧州仕様の eCall システムと同様の仕組みを採用しています。つまり、同じ原則とプロトコルを使用しています。一方で SMS のような MSD の転送用冗長チャネルを提供します。例えば、車両管理や有料道路システムそしてデジタル・タコグラフ(デジタル自動回転速度計)のようなサービス提供のための設計が施されています。



ERA-GLONASS は事故を防ぐことはできませんが、例えばエアバックが展開した時などに、音声通話発信が自動的に緊急番号 112 へダイヤルされます。それと同時に、車両の現在位置を含む重要な情報が仕様で定められたデータ形式で送信されます。車載システム (IVS) はデータを収集し、公衆安全応答ポイント (PSAP) に GSM や WCDMA の音声網を経由しデータを送信します。

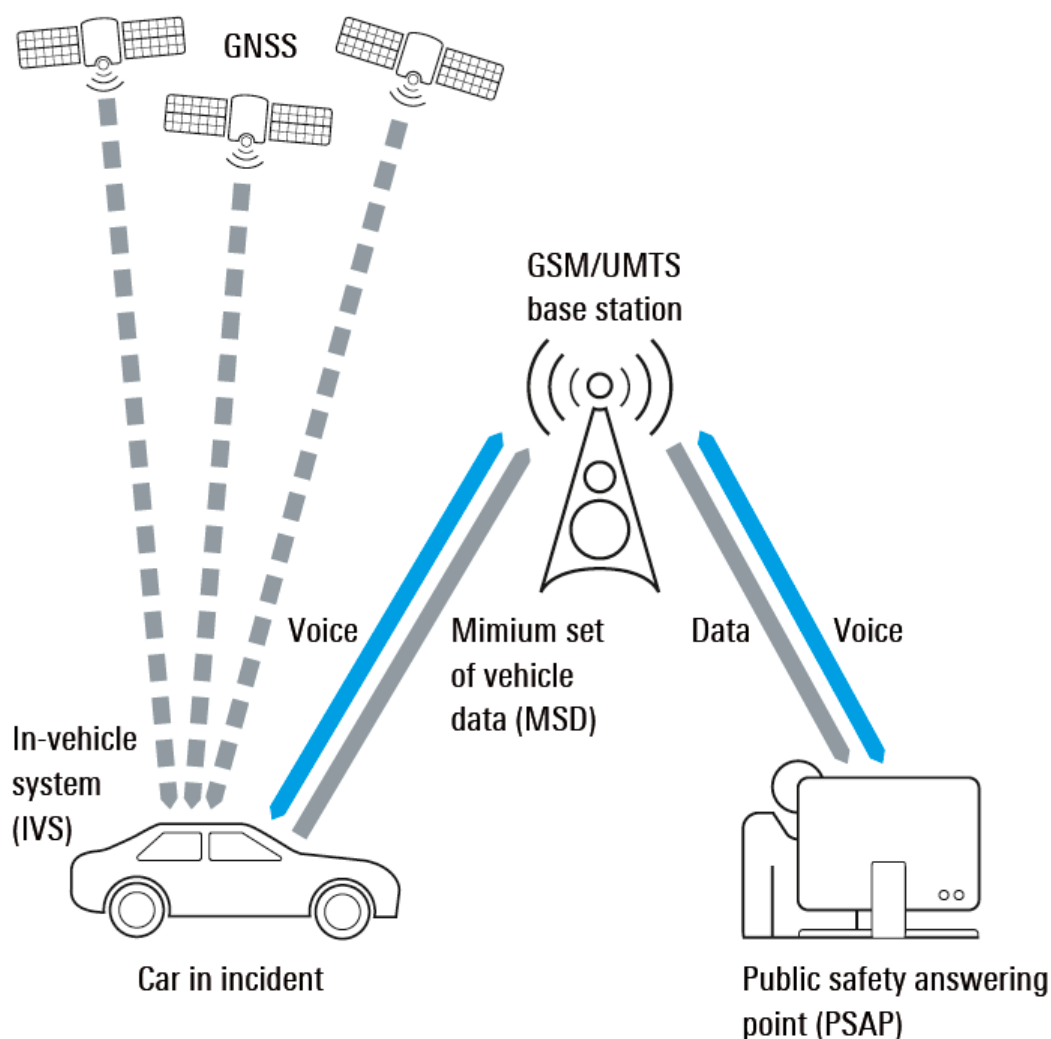


図 1-1: ERA-GLONASS システム: 緊急事態時に車両は、GSM または WCDMA の回線網を使用し緊急サービスへ通知を行いません。車両は自動的に位置情報を含む重要なデータを転送します。その後、例えば追加情報等を確認するため緊急サービスのオペレーターは直接乗員と通話を行うことができます。

本アプリケーションノートは ERA-GLONASS の背景にある原理とローデ・シュワルツが提供するテスト・ソリューションを用いた IVS のコンFORMANCE・テストの説明について記述されています。

ソフトウェアとテスト・ソリューションは、ユーザーが実際に必要なテスト項目を実施する際に使用しやすい仕様になっています。

- PSAP のシミュレーションと GSM または WCDMA を経由した ERA-GLONASS 緊急通話
- 時間測定と最小のデータ・セット (MSD) のデコード
- 相互間のプロトコル・メッセージの表示
- 無線通信や GNSS の仕様の深い理解を必要としない
- 使用者を選ばない、扱いやすい操作性
- GSM や WCDMA に関する各種パラメータを変更することが可能
- 再現性のある測定結果を実施可能

- 緊急電話番号 112 を使用した、実際の緊急通話が可能

本アプリケーションノートに記載されているローデ・シュワルツテスト機器に関する略語を以下に示します：

- 広帯域ワイドバンド無線機テスト R&S®CMW500 を CMW と表記します。
- ベクトル・シグナル信号発生器 R&S®SMBV100A を SMBV と表記します。

## 2 ERA-GLONASS とは

### 2.1 ERA-GLONASS: システムとコンセプト

ERA-GLONASS とは、道路上の事故または他の緊急事態が生じた際の応答時間を削減することを目的としたロシアで提供が始まるサービスです。ロシア政府は、今後すべての新しい自動車モデルに ERA モジュールの搭載を義務づける法律を制定しました。ERA は、欧州規格である eCall システムと同様の仕組みを採用しています。つまり、同じ原則とプロトコルを使用しています。一方で SMS のような冗長チャネルまたは、追加のデータ・サービスのような追加機能を提供します。

表 2-1: ERA/eCall の特徴と相違点

Feature	eCall	ERA-GLONASS
Radio Access Networks	GSM (2G) / UMTS (3G) (optional)	GSM (2G) / UMTS (3G)
GNSS	GPS	GLONASS (mandatory), GPS
In-Band Modem	yes	yes ( same to pan-European eCall)
secondary (redundant) channel	no	SMS
MSD	MSD	MSD (like the MSD in eCall, but optional additional fields)
additional (packet) data channel	no	yes for location-based services like <ul style="list-style-type: none"> <li>• fleet management</li> <li>• insurance telematics</li> <li>• toll collection</li> <li>• recovery of stolen vehicles</li> <li>• digital tachygraphy</li> </ul>

ERA-GLONASS は事故を防ぐことはできませんが、例えばエアバックが展開した時などに、音声通話発信が自動的に緊急番号 112 へダイヤルされます。それと同時に、車両の現在位置を含む重要な情報が仕様で定められたデータ形式で送信されます。自動ダイヤル機能に加え、ERA は他の緊急事態が生じた際に手動で緊急通知することもできます。

概要については、1, 「はじめに」 (3 ページ) を参照ください。

車載システム (IVS) は、すべての自動車に重要な要素として搭載されています (図 2-1)。車載システム (IVS) は、すべての自動車に重要な要素として搭載されています (図 2-1)。IVS の主要コンポーネントは、公衆安全応答ポイント (PSAP) に携帯電話ネットワークを介して最小のデータセット (MSD) の転送を可能にするため、現在地と携帯モジュール (GSM, WCDMA) を認識するための GNSS (GNSS: 通例 GPS を有する) レシーバーである。

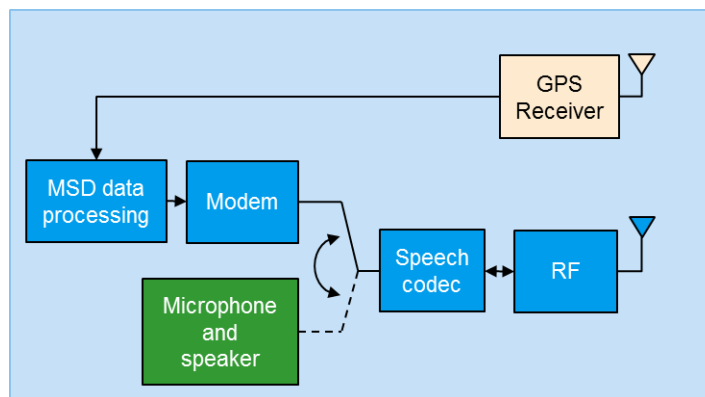


図 2-1: ERA-GLONASS の主要コンポーネント: IVS は重要なデータを準備し、無線インターフェース上で音声チャンネル介しデータを転送します。

公衆安全応答ポイント (PSAP の例: 警察、消防、EMT) は音声通話に応答すると同時にデータを受信し、適切な応答を開始します (救急車の派遣の有無等)。そして、音声接続を介して追加情報を確認します。

IVS は、標準化された MSD として PSAP にデータを送信します。

表 2-2: MSD のデータ内容: 最重要データとして、車両の位置 (ブロック 7...11)。さらに ERA には、診断結果とクラッシュ情報が含まれます。

Block	Name	Description
1	Format version	MSD format version: set to 1 for the current format EN15722:2011
2	Message identifier	Session specific counter; starts at 1 and is incremented with every retransmission
3	Control	Conveys the following information: Automatic or manual activation test call (TRUE/FALSE) Position can be trusted (TRUE/FALSE) Vehicle class
4	Vehicle ID	VIN (vehicle identification number) according to ISO3779
5	Propulsion type (energy storage)	Gasoline, diesel, hydrogen, electric, etc. (TRUE/FALSE)
6	Timestamp	Timestamp of incident event
7	Vehicle location	Latitude and longitude (ISO 6709) in milliarseconds
8	Vehicle direction	Deviation from the direction to the magnetic north pole in 2 degrees steps
9	Recent vehicle location n-1	Change in latitude and longitude compared to the last MSD transmission
10	Recent vehicle location n-2	Change in latitude and longitude compared to the last but one MSD transmission

11	No. of passengers	Minimum known number of fastened seatbelts; omitted if no information is available
12	Optional additional data	Optional information for the emergency rescue service (103 bytes, ASN.1 encoded); may also point to an address, where this information is located

### 帯域内モデム：主要 MSD 転送

MSD は 140 バイト、28 CRC ビットから構成され、長さは合計で 1148 ビットになります。前方誤り訂正のための符号化の後、MSD の長さは 1380 ビットになります。MSD は接頭辞同期フレームと 3 のデータの部分から構成されます。個々の部品の間、信号がミュートされます (図 2-2)。

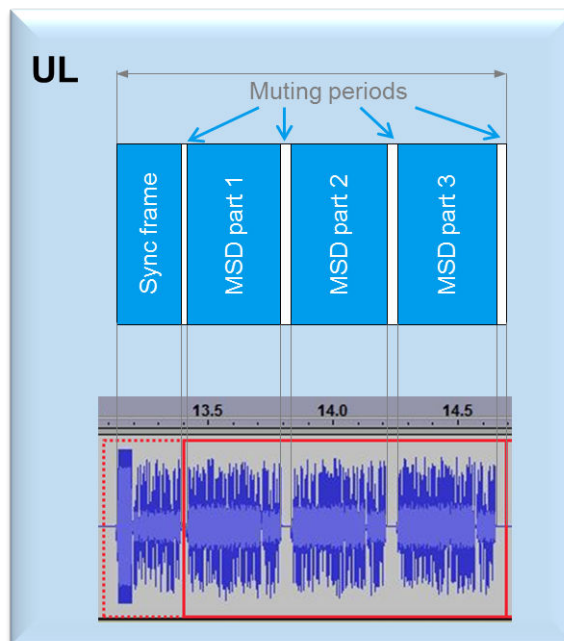


図 2-2: MSD は同期フレームと 3 の MSD 部 (アップリンクデータ部) で構成されています。MSD 間には、信号がミュートされます。オーディオファイル (WAV 形式) からの抜粋例を、図の下部に示します。

MSD データは、変調されたバイポーラパルス位置 (BPPM) です。基本パルス ( $p_{UL}(n)$ ) は、正または負のいずれかの方向 (図 2-3) のいずれかでオフセットされています。合計では、3 ビットのシンボルにパックされ、そして 8 つの異なるシンボルが利用可能です。高速かつ堅牢な 2 つのモードがあります。堅牢モードにおいて、シンボルは高速モードに比べて倍の間隔で送信されます。これは、チャネル状態が悪い状況において、簡単に受信機がシンボルを認識できるようになります。

表 2-3 に、個々のシンボルのパラメータを示します。図 2-4 図 2-4 に、時間経過による変調の例を示します。



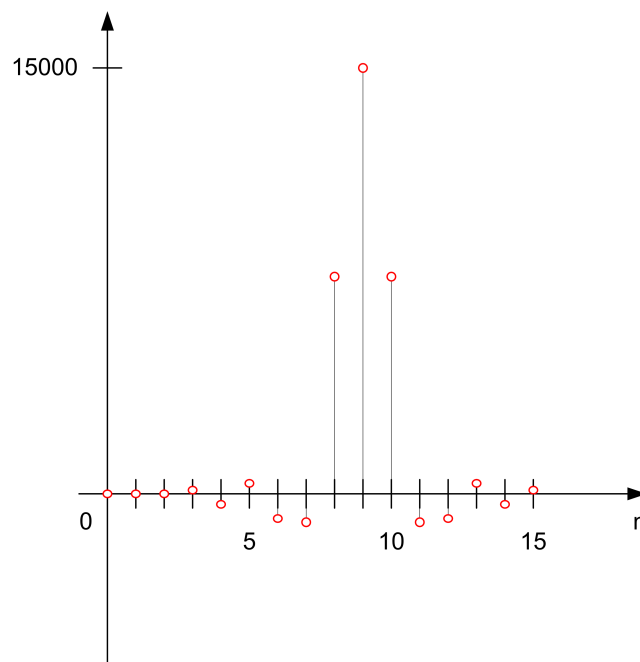


図 2-3: 基本 BPPM パルス ( $p_{UL}(n)$ ) (例として、“fast mode”使用時を示します。)

表 2-3: [2]に沿ったシンボル変調マッピング ( $w_{UL}(n) = q(p(n) \rightarrow k)$ )

Symbol		Uplink waveform (fast mode: $n = 0, 1, \dots, 15$ )		Uplink waveform (robust mode: $n = 0, 1, \dots, 31$ )	
a	b	sign q	cyclic shift k	sign q	cyclic shift k
0	000	1	0	1	0
1	001	1	4	1	8
2	010	1	8	1	16
3	011	1	12	1	24
4	100	-1	12	-1	24
5	101	-1	8	-1	16
6	110	-1	4	-1	8
7	111	-1	0	-1	0

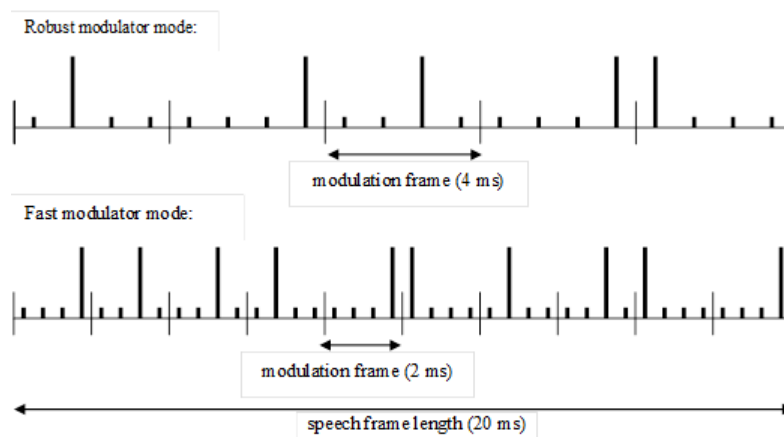


図 2-4: 1 フレーム内でのパルスの概要。上部の図は“Robust mode”、下部の図は“fast mode”を示しています。[2]

### 二次チャネル：SMS

バックアップの方法として、ERA-GLONASS では SMS が MSD に送信されます。MSD の受信確率を向上させるための二次的な方法です。音声接続が正常に確立されていますが、正しい MSD 受信がされていない場合、PSAP により SMS の送信が求められます。音声チャネルが正常に確立されていない場合、IVS においても同様に SMS の送信が求められます。

### 追加データチャネル

基本サービス・データ・チャネルは、追加のロケーションベースのサービスのために使用することができます（表 2-1 を参照）。ここで既に（E）GPRS のような GSM または HSPA（+）のような WCDMA が提供するパケットデータチャネルが使用されている可能性があります。

## 2.2 シーケンス

時間経過による通信状態を説明した一般的なシナリオです。

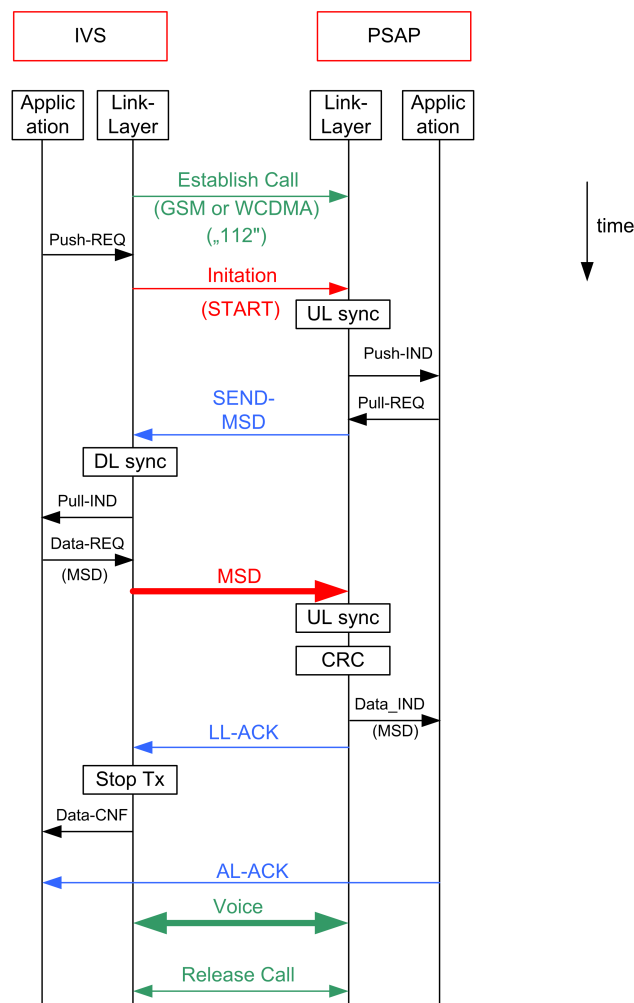


図 2-5: 一般的な ERA-GLONASS シーケンス。MSD の送信は反復プロセスです。PSAP が MSD を理解していない場合、NACK を送信し、IVS が異なる冗長バージョンで改めて MSD を送信します (RV0...7)。MSD を正常に受信した場合は、PSAP は、ACK を送信します。



図 2-6: オーディオ・ファイル内での通話シーケンス (上部: IVS、下部: PSAP)

1. 緊急事態が生じた際に、IVS は自動的に GSM 網または WCDMA 網を經由して PSAP との接続 (緊急通話) が確立されます。連続的に (最大 5 時間) START メッセージが送られ続けます。
2. PSAP が通話を受信し START 信号を認識すると直ぐに、IVS に SEND-MSD コマンドが送信されます。

3. 一度 IVS が SEND-MSD コマンドのデコードに成功すると、冗長バージョン RV0 (“MSD”)を持つ実際の MSD に続き 1 つの SYNC フレームが送信されます。
4. ( [図 2-6](#) を参照) : 例として、PSAP が MSD RV0 を理解せず、NACK (“MSD”)を返します。
5. ( [図 2-6](#) を参照) : IVS は NACK を認識し、RV1 を持つ MSD を PSAP (“MSD”)に再度送信します。
6. PSAP が MSD RV1 (CRC check)のデコードに成功し、フィードバックとして LL-ACK を送信します。IVS は MSD の送信を停止します。
7. PSAP は eCall の完了と MSD が理解されたというデータを示す AL-ACK を送信します。その後、音声接続に切り替わります。PSAP オペレーターは乗員 (例 : 家族)と通話することができます。最後に GSM または WCDMA の接続を終了します (“Release Call”)。

## 2.3 規格

ERA-GLONASS は、“Technical Regulating and Metrology” (GOST) を統括するロシア連邦庁により、複数の規格番号が定義されています (このリストは完全ではないことに注意してください)。

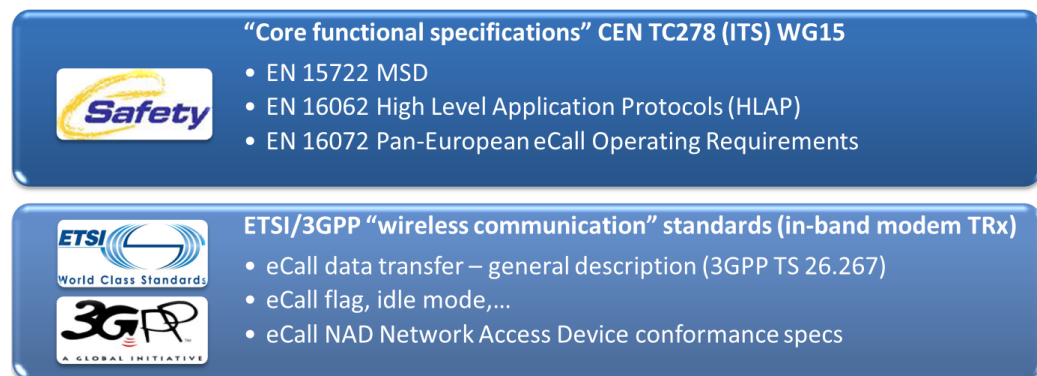
### 一般仕様

- GOST R 54619-2011: 車載緊急通話システムから緊急対応システム基盤へ送信されるデータ・プロトコル。
- GOST R 54620-2011: グローバル・ナビゲーション衛星システム、交通事故緊急対応システム、車載緊急通報システムに関する一般的な技術要件。
- GOST R 54721-2011: グローバル・ナビゲーション衛星システム、交通事故緊急対応システムに関する基本サービスの説明。
- GOST R 55524-2013 グローバル・ナビゲーション衛星システム、交通事故緊急対応システムに関する用語と定義。

### コンプライアンス・テスト

- GOST R 55530-2013: グローバル・ナビゲーション衛星システム、交通事故緊急対応システム。機能的な車載緊急通報システムの試験方法及びデータ転送プロトコル (機能的試験方法とデータ送信プロトコルを含む)。
- GOST R 55531-2013 : グローバル・ナビゲーション衛星システム、交通事故緊急対応システム。車両内のスピーカー品質の要件に関する車載緊急通報システムの適合性試験方法。
- GOST R 55533-2013 : グローバル・ナビゲーション衛星システム、交通事故緊急対応システム。車載緊急通報システムの無線通信モジュールのための試験方法。
- GOST R 55534-2013 グローバル・ナビゲーション衛星システム、交通事故緊急対応システム。車載緊急通報システムのナビゲーションモジュール試験方法。

ERA-GLONASS は eCall と同様のシステムです。 [図 2-7](#) に eCall で定められている規格を示します。



**“Core functional specifications” CEN TC278 (ITS) WG15**

- EN 15722 MSD
- EN 16062 High Level Application Protocols (HLAP)
- EN 16072 Pan-European eCall Operating Requirements

**ETSI/3GPP “wireless communication” standards (in-band modem TRx)**

- eCall data transfer – general description (3GPP TS 26.267)
- eCall flag, idle mode,...
- eCall NAD Network Access Device conformance specs

図 2-7: 最も重要な規格の概要

ローデ・シュワルツのソリューションは、ここに提示された規格の要件を全て満たしています。さらに、エンド・ツー・エンドの適合性試験においては、次の章で説明する、GOST R の 55530 - 2013 に準拠していることを保証します [1]。

## 3 IVS コンフォーマンス・テスト

### 3.1 ラボでテストを実施する理由

ERA-GLONASS デバイスのテストは様々な段階で必要になります。:

- モジュール（ボード）
  - － 開発/設計検証
  - － 生産
  - － 保守
- デバイス（IVS）
  - － 開発/設計検証
  - － 相互接続性
  - － コンフォーマンス・テスト
  - － 受入試験
  - － 生産
  - － 修理/保守
- システム（車両）
  - － 放射性能
  - － 車体仕様
  - － 相互接続性
  - － 脱感作
  - － アウトドア/インドア

本書に示すソリューションは、モジュールおよびデバイス試験について記載しています。チャンネルやレベルのような GSM または WCDMA ネットワークの設定を直接変更することが可能です。再現性のある測定を実施することができます。シールドされた環境での緊急電話番号 112 を使用した、実際の緊急通報をシミュレートすることも可能です。ネットワークオペレータが策定するような制御ができない設定項目の効果を避けることができます。

### 3.2 試験セットアップ

本書に記載する試験ソリューションは、GOST R 55530-2013 [1]に準拠する ERA-GLONASS のコンフォーマンス試験について記載します。図 3-1 に試験セットアップを示します。

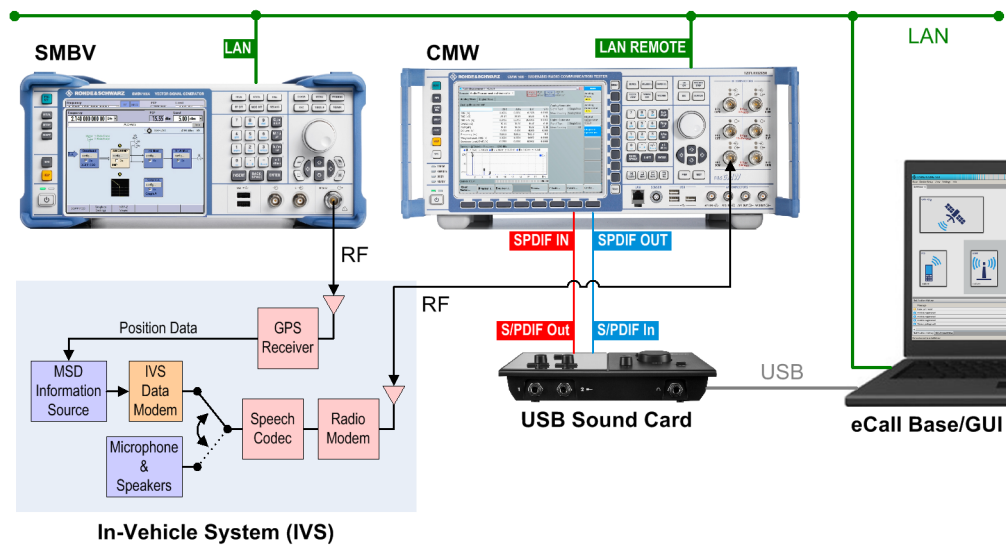
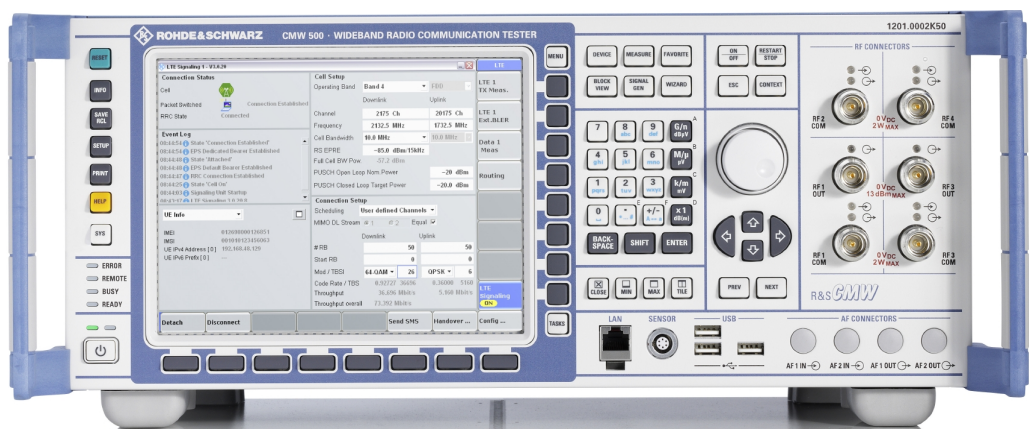


図 3-1: IVS コンフォーマンス試験のセットアップ

CMWは携帯電話ネットワークをシミュレートし、GSM (2G)またはWCDMA (3G)セルを再現します。IVS への RF 接続を介して、GSM または WCDMA シグナリング(例：通話セットアップ)ならびに音声接続が行なわれます。SMBV は、IVS の位置測定に使用される GLONASS または GPS をシミュレートした GNSS 信号を提供します。PSAP をシミュレートする外部 PC に CMW-KA095 (要 CMW-KA094) ソフトウェアをインストールし、CMW 上で ERA-GLONASS に必要なパラメータをセットします。CMW および SMBV は完全にリモート制御することができます。CMW および SMBV の操作知識は必要ありません。GSM または WCDMA 接続を介して転送されたオーディオ信号は、外部のサウンドカードから (へ) デジタル/ SPDIF インターフェースを介してルーティングされます。ERA-GLONASS プロトコルと MSD の送信はまた、CMW と IVS の間の音声接続を介して処理されます。エンドツールの適合性試験は、外部 PC とテスト中の IVS にインストールされている PSAP シミュレータとの間で実行されます。

### 3.2.1 ワイドバンド無線機テスト CMW



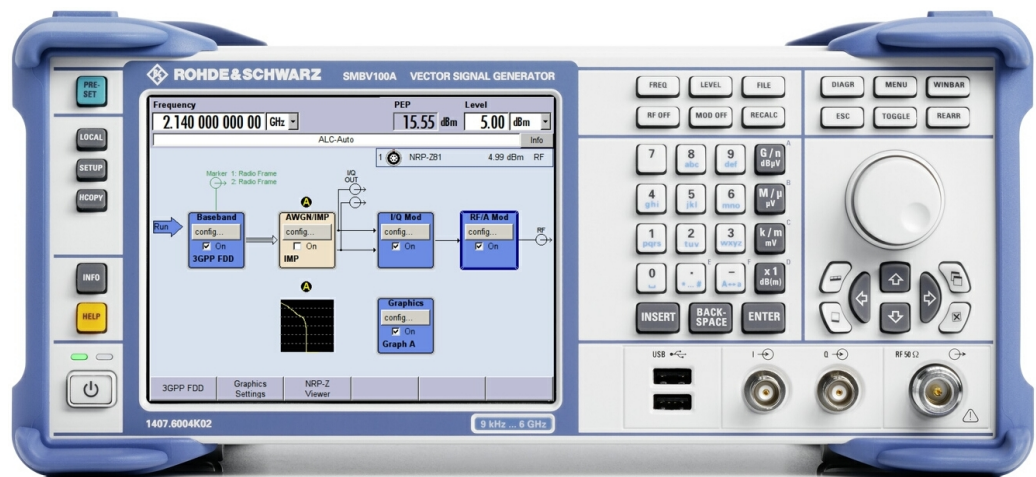
CMW は、移動無線および無線接続のような各種無線通信アプリケーションのためのオール・イン・ワン試験ソリューションです。CMW は下記の無線規格に対応しています。

- 2G
  - GSM, EGPRS, EGPRS2, EGDE Evolution, VAMOS
- 3G
  - W-CDMA with HSDPA, HSUPA, HSPA+
  - TD-SCDMA
  - CDMA2000, 1xEV-DO Rev A/B
- 4G
  - LTE (FDD/TDD), LTE-A (MIMO を含む)
- 無線接続
  - Bluetooth
  - WLAN
  - WiMAX

CMW は RF 試験やプロトコル試験を含む、物理層からエンド・ツー・エンドに至る、すべての OSI 層を試験することができます。

追加の packets データチャネルは、ERA-GLONASS ソフトウェア KA095 を使用することなく CMW 単体で試験することができます。

### 3.2.2 ベクトル・シグナル・ジェネレーター SMBV



SMBV は様々な無線通信規格だけでなく、他の無線規格に対応しています。ERA-GLONASS のための GNSS 信号を生成することができます。

GNSS 特性の概要：

- GPS, Glonass, Galileo, BeiDou, QZSS に対応
- 最大 24 個の衛星をシミュレーションが可能
- シミュレーション時間に制限無く自動衛星ハンドオーバーに対応
- 衛星の現在位置と（アクティブまたは非アクティブ）状態が、スカイ・ビュー部に表示されます。



- 高密度の都市などで観測したマルチ・パス・シナリオ (LOS + エコー)
- 移動シナリオは、移動中の車で観測されるようなユーザー定義の軌道に沿った受信機の動きを模擬します。(ウェイポイント形式は、KML、NMEA ファイルなどに対応)
- マップ表示部は、受信機の軌跡を観測することができ、受信機の現在位置を示します。

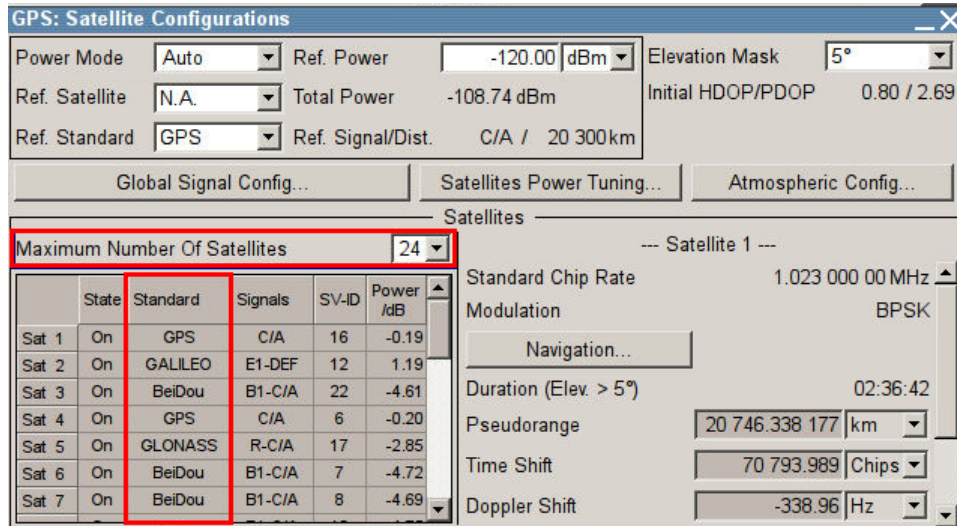


図 3-2: SMBV 上での衛星構成の一例

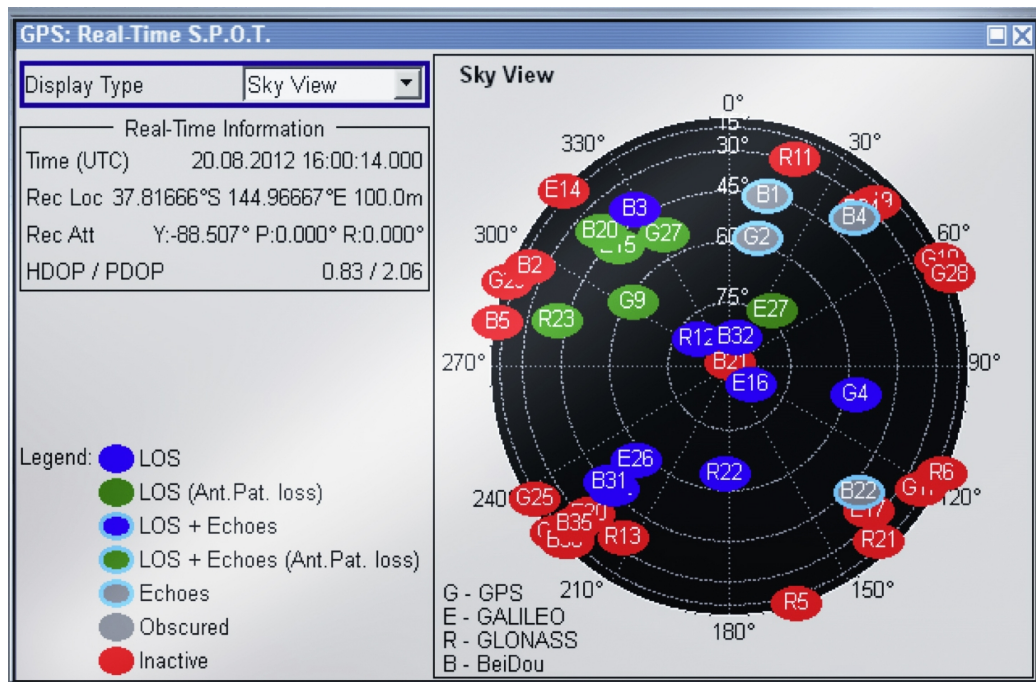


図 3-3: 地平線上の衛星の分布を示したスカイ・ビュー表示

GNSS に必要なオプション

SMBV に必要なオプション (eCall/ERA-GLONASS) :

表 3-1: GNSS シナリオ

Scenario	Option SMBV	Configuration	Satellites
GPS - City	K44	static, selectable city	fixed satellite constellation
GLONASS - City	K94	static, selectable city	fixed satellite constellation
GPS - Atlanta	K65	static, Atlanta location	fixed satellite constellation
GPS - Melbourne	K65	static, Melbourne location	fixed satellite constellation
GPS - Melbourne moving	K65	moving, Melbourne location	fixed satellite constellation
GPS - Atlanta Individual Power	K65	static, Atlanta location	variable satellite constellation
GPS - Melbourne Individual Power	K65	static, Melbourne location	variable satellite constellation
GPS - Melbournemoving, Individual Power	K65	moving, Melbourne location	variable satellite constellation

### 3.2.3 KA095 PSAP シミュレーション・ソフトウェア (ERA-GLONASS 試験ソフトウェア)



CMW-KA095 ソフトウェアは外部 PC 上で動作し、PSAP シミュレーションとして機能します。また、LAN 等を使用することで CMW と SMBV を外部制御することができます。ソフトウェアは、エンド・ツー・エンドのコンフォーマンス試験を実行します。MSD はデコードされ、測定結果を表示します。

- GSM または WCDMA を用いた ERA-GLONASS のための PSAP シミュレーション
- MSD 送信時間
- 通話が確立するまでの時間
- PSAP からの最初のトリガーがかかるまでの時間

- すべての冗長バージョンおよびアップリンク・データ部分について GEN EN 15722:2011 と GOST R 54620-2011 A1 に沿って MSD をデコード
- IVS からデコードされていない信号のオプション記録
- PUSH と SYNC の詳細な表示
  - タイミング
  - カウント
- GOST R 54619-2011 に基づく 200 以上の SMS コマンドを有する SMS プロトコル
- オプション：固定位置での GPS / GLONASS シミュレーション/ 移動する GPS シナリオ

### 3.3 ERA-GLONASS 測定：基本オペレーション

ERA-GLONASS の一般的な設定については、次の章に記載します。試験システムの準備および IVS からシミュレートする PSAP へ MSD を転送する ERA-GLONASS についての手順を示します。

図 3-1 に使用する試験セットアップを示します。

IVS は RF ケーブルを介し GSM または WCDMA のセルをシミュレートする CMW に接続します。この接続を介し、IVS は、GSM の緊急呼を確立し、CMW に MSD を転送します。GNSS 信号は RF ケーブルを介し、IVS へ送られます。

GSM または WCDMA 接続を介し転送された音声信号は、外部サウンドカードへ、あるいは、からデジタル SPDIF インターフェースを介してルーティングされます。

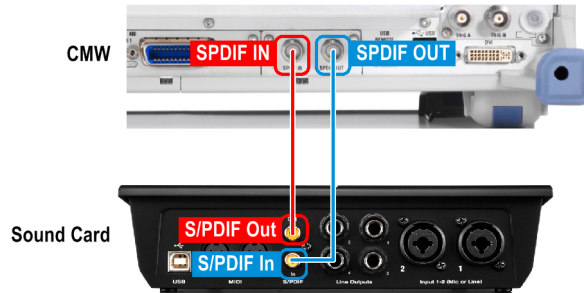
ERA-GLONASS のアプリケーションのベースと GUI は、PC にインストールされています。アプリケーションは PSAP をシミュレートし、LAN 経由で CMW の制御をします。そして、USB 接続を介してサウンドカードにアクセスします。

#### 3.3.1 試験セットアップの準備

以下のステップは試験のセットアップ、ERA-GLONASS アプリケーションの実行方法について記載します。

1. 試験セットアップの準備：
  - a) PC を LAN に接続します。
  - b) CMW の背面にある“LAN REMOTE”コネクタに LAN を接続します。
  - c) SMBV の背面にある“LAN”コネクタに LAN を接続します。
  - d) IVS の GSM RF コネクタを CMW の前面にある“RF 1 COM”に接続します。
  - e) IVS の GPS RF コネクタを SMBV の前面にある“RF”に接続します。

f) サウンドカードの背面パネルを CMW の背面パネルに接続します。



ケーブルはサウンドカード（GWM-Z94）に同梱されています。

g) USB ケーブルを使用し、サウンドカードを PC に接続します。

2. PC の準備をします。必要な手順は、CMW-KA095 ユーザーマニュアル “Preparing the Test System for Use” に詳述されています。

以下のリストは、概要を説明します：

- ERA-GRONASS アプリケーション用ソフトウェアをインストールします。
- NET Framework 4.0 をインストールします。
- NET Framework4.0 の開発をサポートする VISA ライブラリをインストールします。
- サウンドカードのドライバをインストールし、サウンドカードを設定します。
- PC のサウンド・デバイス・セッティングの設定をします。
- ライセンスを有するスマートカードを取り付けることによって、例えば、ライセンス CMW-KA095（CMW-KA094）を使用できるようにします。

### 3.3.2 ERA-GLONASS 試験ソフトウェア（KA095）の起動と設定

次の手順では、eCall のテストソフトウェアの起動、試験セットアップの設定の方法を記載します。

手順は、デフォルト設定から、必要な手順まで説明します。すでに設定を変更した場合は、アプリケーションを起動する前に、次の構成ファイルの名前を変更または削除することでデフォルト値に設定をリセットします

(%APPDATA%\Rohde-Schwarz\CMW-KA09x\_GUI\<version>\user.config)。

eCall 試験ソフトウェアは 2 つの部分から構成されます：

1. ERA-GLONASS テストソフトウェアのベースを起動します：  
 “Windows Start” menu > “All Programs” > “R&S CMW-KA09x” > “KA09x Application Base”

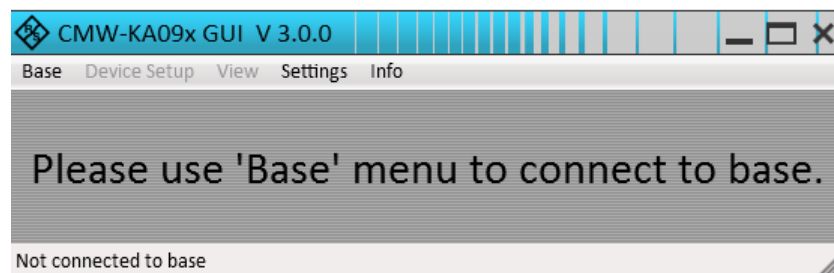
コンソールウィンドウが開きます。成功している場合のスタートは “Startup done!” と示されます。

```

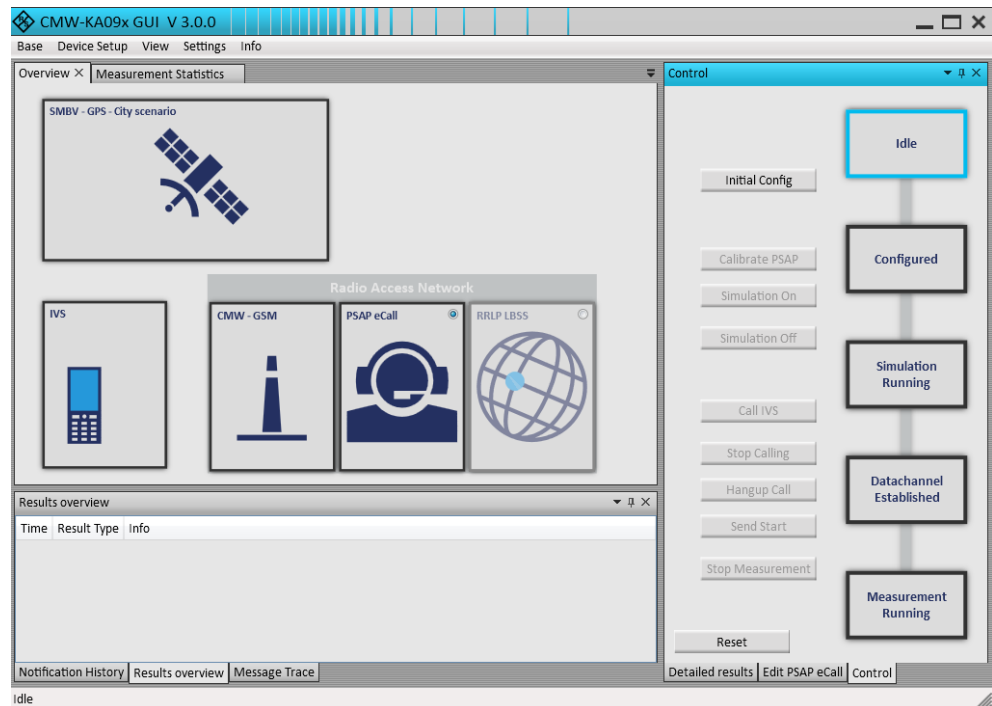
KA09x Appl Software Base
[DEBUG @ Boot/Boot_0] 'OptionKeyStandAlone Dynamic Link Library' - U2.2.0.0 - 20
14 Feb 4 - 09:52:32 - 26ba808b95d7ac48ecb515c79d5d1a64
[DEBUG @ Boot/Boot_0] 'OptionkeyWrapper' - U2.1.27.1 - 2013 Jul 9 - 08:57:49 - e
c2d05b923255c2046a4e994fefe8cac
[DEBUG @ Boot/Boot_0] 'GPSTk.dll' - 2014 Aug 5 - 12:24:20 - d20688bffb6b3e844eac
7849bf2573fe
[DEBUG @ Boot/Boot_0] 'Internet Explorer Compatibility Shims' - U8.00.7600.16385
(win7_rtm.090713-1255) - 2009 Jul 14 - 03:06:22 - 6464f37ad3d894de34376b46db9e6
269
[DEBUG @ Boot/Boot_0] 'RS-AGPSHandler.dll' - 2014 Aug 5 - 12:25:10 - 9ba612ea98d
dae6961d7d0826010bbe3
[DEBUG @ Boot/Boot_0] 'Wrapper.dll' - 2014 Aug 5 - 12:25:26 - 6c8e1b5bd8fb7c7bff
c181370f0fac31
[DEBUG @ Boot/Boot_0] 'Group Policy Client' - U6.1.7600.16385 (win7_rtm.090713-1
255) - 2009 Jul 14 - 03:05:34 - 8ba3c04702bf8f927ab36ae8313ca4ee
[DEBUG @ Boot/Boot_0] 'Libregex: searches for and matches text strings' - U2.7.2
853.21853 - 2007 Oct 24 - 12:10:55 - 547c43567ab8c08eb30f6c6bach479a3
[DEBUG @ SoapInterface/Boot_0] SOAP connection state INIT -> INIT
[INFO @ SoapInterface/Boot_0] Base soap server is listeninig on: http://127.0.0.
1:8085/baseService
[INFO @ Boot/Boot_0] Startup done!
[INFO @ Boot/Boot_0] Will use config file: C:\Program Files (x86)\Rohde-Schwarz\
CMW-KA09x\config\KA09x_config.xml
[DEBUG @ Boot/Boot_0] EventStatus.description = 'All ok!'

```

- ERA-GLONASS 試験ソフトウェア GUI を起動します：  
 “Windows Start” menu > “All Programs” > “R&S CMW-KA09x” > “KA09x Appl Software GUI”  
 The “CMW-KA09x GUI”ウィンドウが開きます。

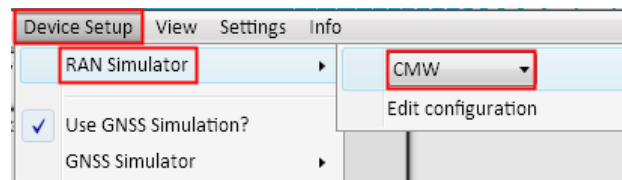


- GUI をベースに接続します：  
 “Base” menu > “Connect”  
 正常に接続すると以下のような GUI が表示されます。

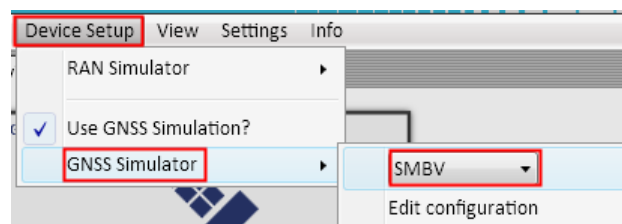


ステータスバーは、基本状態として“Idle”を示します

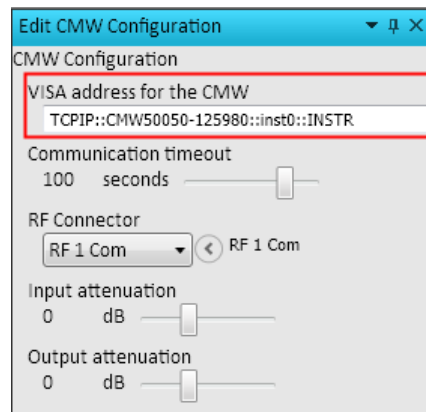
- RAN シミュレータを CMW にします：  
“Device Setup” menu > “RAN Simulator” > “CMW”



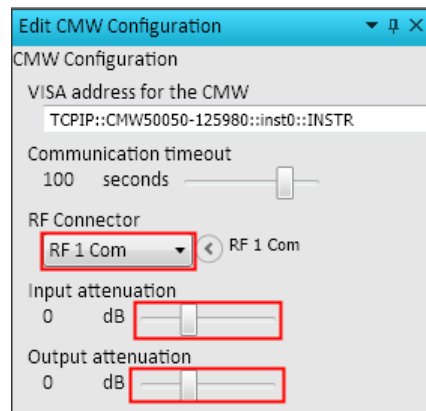
- GNSS シミュレータを SMBV にします：  
“Device Setup” menu > “GNSS Simulator” > “SMBV”



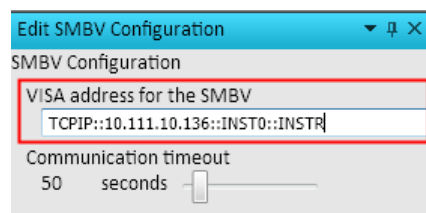
- CMW の設定を表示します：  
“Device Setup” menu > “RAN Simulator” > “Edit configuration”
- VISA アドレス・ストリングまたは CMW の TCP IP アドレスを入力します。



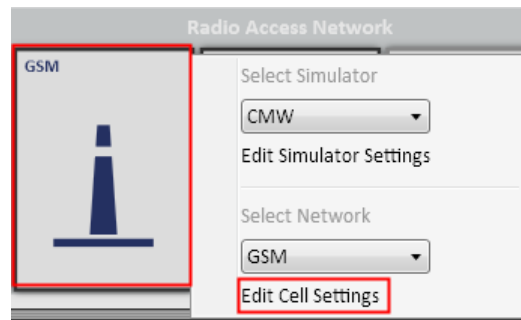
8. CMWに必要な他の項目の設定を構成します：
- GSM 信号に使用する RF コネクタを選択します。
  - RF ケーブルの減衰量に応じて、入力と出力の減衰量を設定します。



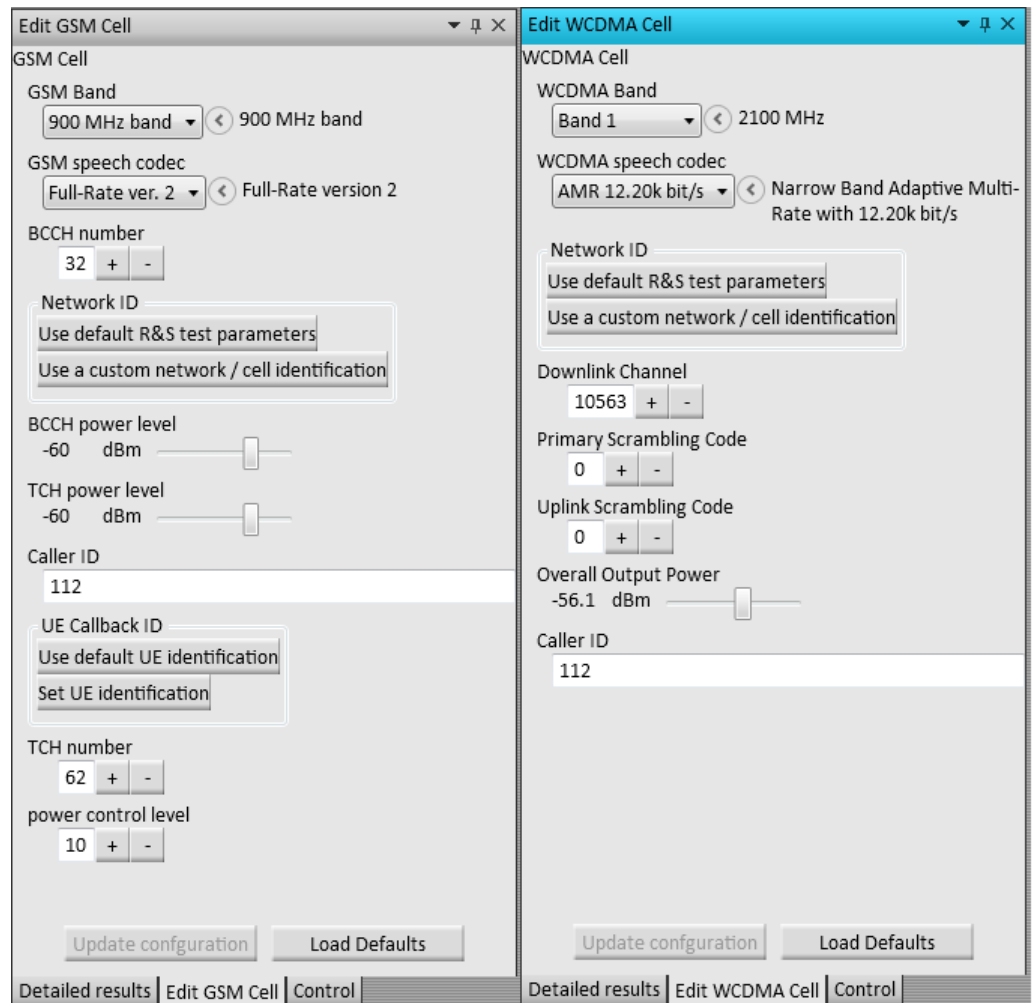
9. SMBV の設定を表示させます：  
 “Device Setup” menu > “GNSS Simulator” > “Edit configuration”
10. VISA アドレス・ストリングまたは SMBV の TCP/IP アドレスを入力します。



11. 左側にある無線アクセスネットワーク内のベースステーション上で右クリックします。ネットワーク GSM または WCDMA の選択をし、“Edit Cell Settings”をクリックします。

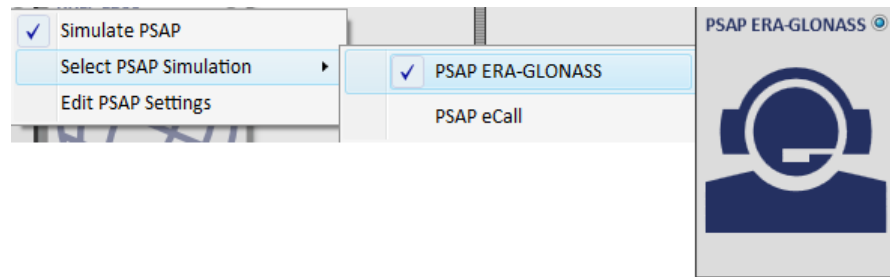


セルの設定（GSM または WCDMA）が右側に表示されます。

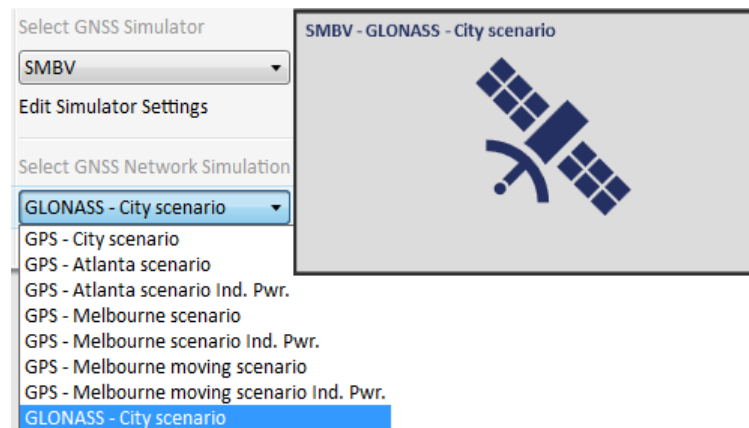


12. セルの設定が IVS に互換性があることを確認します。  
必要に応じて、例えば、帯域、チャンネル番号および電力レベルの設定を変更します。  
初期設定後に設定を変更した場合、変更を適用するために下部にある“Update Configuration”をクリックします。
13. 左側の無線アクセスネットワーク内にある PSAP エリアを右クリックします。ここで PSAP シミュレータ “ERA-GLONASS” を選択します。





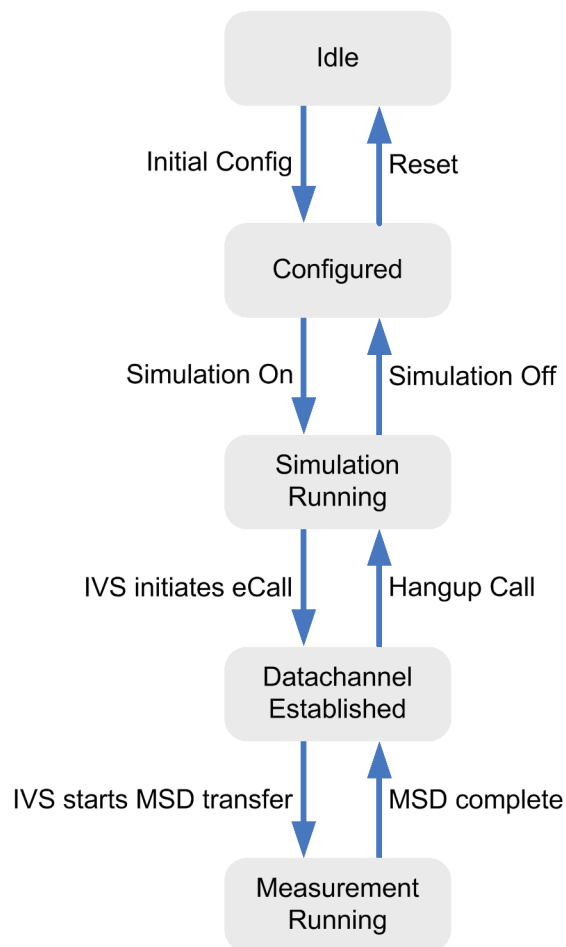
14. GLONASS を使用するには、左側の SMx エリアを右クリックします。ここで、“GLONASS - City Scenario”を選択します。



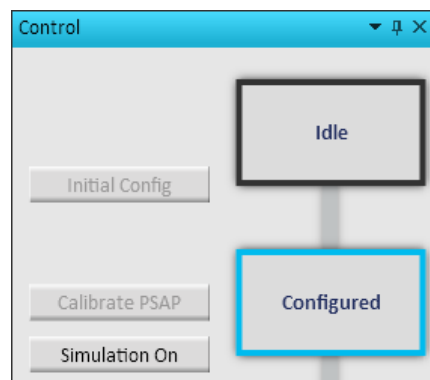
### 3.3.3 ERA-GLONASS 通話の実行

この項の手順では、GSM 緊急通話の設定、IVS から eCall アプリケーションへの MSD の送信および MSD の解析について記載します。

次の図は、ERA-GLONASS アプリケーションの状態を示しています。初期状態は、“Idle”状態になっています。そして、“Measurement Running”状態を経由し、“Datachannel Established”状態になります。

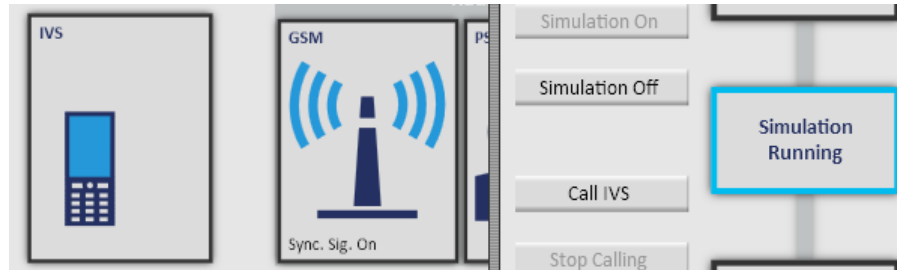


1. 右側の、“Control”ビューを選択します。
2. “Initial Config”をクリックします。  
アプリケーションは、設定を有効にし、CMW と SMBV の設定を行いません。起動後に初めてこのアクションを実行する場合、時間がかかることがあります。  
初期設定が終わると、“Control”ビュー内の状態が“Idle”から“Configured”に変わります。

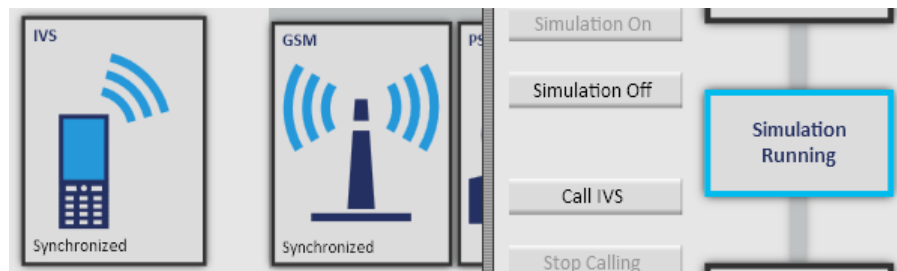


3. “Simulation On”をクリックします。

PSAP、GNSS および GSM シミュレーションが開始します。GSM セル信号が起動します。(時間がかかる場合があります)  
 問題なくシミュレーションが開始されると“Control”ビュー内の状態が“Configured”から“Simulation Running”に変わります。  
 GSM エリアには、アクティブなアンテナが表示されます。



- IVS を起動するとレジストされます。“Notification History”ビュー内では、問題なくレジストされたかを表示します。IVS エリアと GSM エリアのラベルには、IVS がレジスト(同期)されたことを示します。



- IVS 上で ERA-GLONASS 通話が開始されます。“Overview”タブと“Control”ビュー内で通話の進捗を確認できます。次の表には、ERA-GLONASS コールの状態が示されています。

IVS State	GSM State	PSAP State	Control State	Explanation
Synchronized	Synchronized	PSAP initialized <sup>1)</sup>	Simulation Running	IVS registered and synchronized
Connecting	Connecting			Call setup in progress
Call Established	Call Established	Ready for ERA-GLONASS call	Data Channel Established	Data channel between IVS and PSAP established, no MSD transmission
		Sending Start	Measurement Running	PSAP requests MSD from IVS
		Receiving 1st MSD part		PSAP receives first MSD part
		Sending NACK		MSD part received
		Sending LL-ACK		MSD reassembled and sent to higher layers
		Sending HL-ACK		MSD decoded at application layer
Ready for ERA-GLONASS call	Data Channel Established	Call still established, no MSD transmission Sine tone transmission to the IVS		
<sup>1)</sup> For the first ERA-GLONASS call, no state is indicated				

6. 左上部の、“Results Overview”を選択します。
7. “Decoded MSD message”の行をダブルクリックします。

Results overview		
Time	Result Type	Info
17:08:38.0	GSM results	Dialed number 'emergency call'
17:08:42.0	Raw MSD	Redundancy Version = 0 - 0x01 0x01 0x34 0x41 0x41 0x41 0x4
17:08:42.0	Decoded MSD message	Redundancy version 0 of MSD according to EN 15722 2011, de

Notification History | Results overview | Message Trace

“Detailed Results”ビューが自動的に右側に表示されます。MSD メッセージの内容が一覧表示されます。ここに、ERA-GLONASS の追加データが示されています。

Detailed results	
Decoded MSD message according to EN 15722 2011 + GOST R 54620 2011 A1 data	
Timestamp: 2014.07.22 - 15:44:56.3	
<ul style="list-style-type: none"> <li>▲ Additional Data               <ul style="list-style-type: none"> <li>Crash Severity ASI15 = 123</li> <li>▲ Diagnostic Result                   <ul style="list-style-type: none"> <li>Microphone Connection Failure (optional / not present)</li> <li>Microphone Failure (optional / not present)</li> <li>Right Speaker Failure (optional / not present)</li> <li>Left Speaker Failure (optional / not present)</li> <li>Speakers Failure (optional / not present)</li> <li>Ignition Line Failure (optional / not present)</li> <li>Upper Intake Manifold Failure (optional / not present)</li> <li>Status Indicator Failure (optional / not present)</li> <li><b>Battery Failure = False</b></li> <li>Battery Voltage Low (optional / not present)</li> <li>Crash Sensor Failure (optional / not present)</li> <li>Firmware Image Corruption (optional / not present)</li> <li>Communication Module Interface Failure (optional / not present)</li> <li>Gnss Receiver Failure (optional / not present)</li> <li>Raim Problem (optional / not present)</li> <li><b>Gnss Antenna Failure = True</b></li> <li>Communication Module Failure (optional / not present)</li> <li>Events Memory Overflow (optional / not present)</li> <li><b>Crash Profile Memory Overflow = True</b></li> <li>Other Critical Failures (optional / not present)</li> <li>Other Non-Critical Failures (optional / not present)</li> </ul> </li> <li>▲ Crash Info                   <ul style="list-style-type: none"> <li>Crash Front (optional / not present)</li> <li>Crash Left Side (optional / not present)</li> <li><b>Crash Right Side = True</b></li> <li>Crash Rear (optional / not present)</li> <li><b>Crash With Rollover = True</b></li> <li>Crash Side (optional / not present)</li> <li><b>Crash Front Or Side = True</b></li> <li>Crash Other Type (optional / not present)</li> </ul> </li> <li>▲ MSD                   <ul style="list-style-type: none"> <li>ID = 1</li> <li>▲ Minimum Set Of Data</li> </ul> </li> </ul> </li> </ul>	

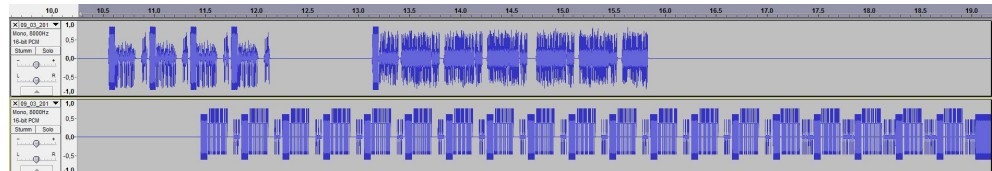
8. ERA-GLONASS 試験ソフトウェアには、各通話のための2つの wav ファイル、IVS からのアップリンク信号および PSAP からのダウンリンク信号が保存されています。

ファイルは、例とし

て%PROGRAMFILES(X86)%\Rohde-Schwarz\CMW-KA09x\Complementaryにある、eCall ベース・アプリケーションのインストール・ディレクトリに保存されています。

ファイルの内容を確認するために、外部オーディオ・エディタでファイルを開きます。

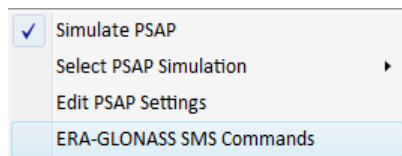
以下の例は、上部に IVS、下部に PSAP を示しています。より詳細な説明については、「帯域内モデム：主要 MSD 転送」(8 ページ)を参照してください。



### 3.3.4 ERA-GLONASS が有する SMS コマンド

ERA-GLONASS は、SMS を介して PSAP と IVS 間で通信することが可能です。例えば、コンフィギュレーション (チェンジ/クエリ・パラメータ) または SMS を経由した MSD 転送を可能にします。

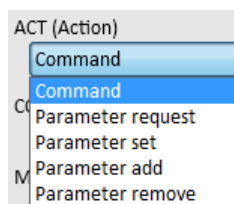
左側の“Radio Access Network”上で PSAP エリアを右クリックします。ここで、“ERA-GLONASS の SMS コマンド”を選択します。



設定は、右下のタブにあります“ERA-GLONASS SMS commands”に表示されます。

#### SMS を介した MSD

SMS 経由で MSD を送信する IVS を制御するために、PSAP は“ACT”設定を“Command”にする必要があります。



CCD (Command Code)	
	EGTS_ECALL_MSD_REQ
M	EGTS_RAW_DATA
	EGTS_TEST_MODE
	EGTS_CONFIG_RESET
T	EGTS_SET_AUTH_CODE
	EGTS_RESTART
	EGTS_ECALL_REQ
	EGTS_ECALL_MSD_REQ
	EGTS_ECALL_DEREGISTRATION

図 3-4: コマンドコード一覧

Transport channel	
IVS DISCRETION	◀ The IVS may choose the channel
IVS DISCRETION	
VOICE	
SMS	

コマンドコード (CCD) “EGTS\_ECALL\_MSD\_REQ”は IVS に MSD 送信を指示します。トランスポートチャンネルは、使用する IVS に依存します (例えば“SMS”の設定)。

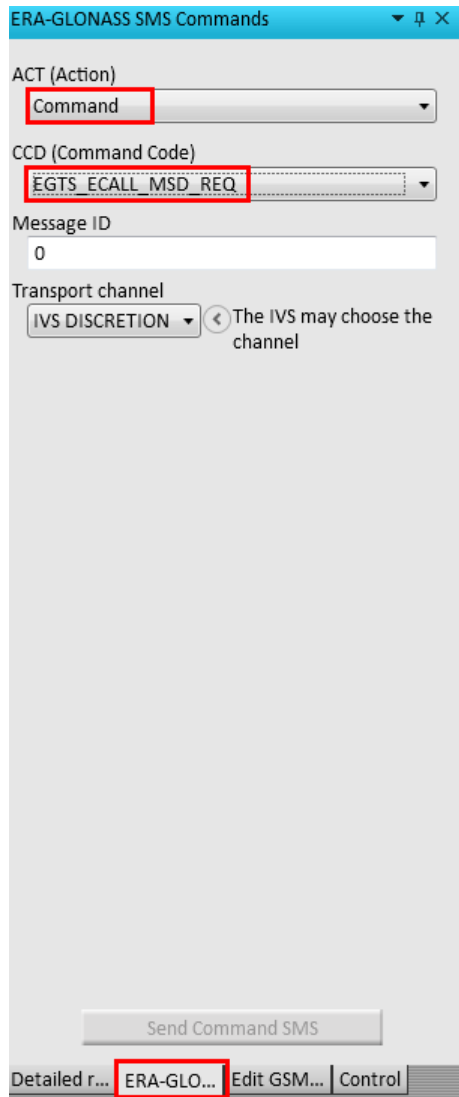


図 3-5: SMS コマンドの設定は、SMS 経由で MSD を送信するために IVS を促す。トランスポートチャンネルは IVS と異なり、IVS に依存します。

SMS が送信される前に、IVS をレジスト (“synchronized”) する必要がありますのでご注意ください。

IVS はその後、MSD で SMS を送信することで応えるものとします。図 3-6 は、メッセージトレースの項目を示しており、図 3-7 は MSD でデコードされた SM を示しています。

14:32:30.5		GSM results			Dialed number '+4953170222602'
14:32:30.6		Decoded ERA-GLONASS M			Decoded MSD successful
14:32:30.6	ERA-GLONAS	Incoming SMS	UE/IVS	PSAP	EGTS_SR_RAW_MSD_DATA (Successful)

図 3-6: メッセージトレース内の MSD で受信する SMS

```

Detailed results
ERA GLONASS SMS according to GOST R 54619 2011 Revision 1
Timestamp: 2015.02.19 - 14:32:30.6

  ▲ pdu (TransportProtocolStructure)
    PRV = GOST_R_54619_2011_Rev_1
    SKID = 0
    PRF = Prefix00
    ENA = NOT_ENCRYPTED
    CMP = NOT_COMPRESSED
    PR = HIGHEST
    HE = 0
    PID = 0
    routeInfo (optional / not present)
  ▲ SFRD (EGTS_PT_APPDATA)
    ▲ SDR [0] = ServiceDataRecord
      RN = 0
      ▲ RFL (RecordFlags)
        SSOD = IVS
        RSOD = TELEMATIC
        RPP = 0
        OID (optional / not present)
        EVID (optional / not present)
        TM (optional / not present)
        SST = EGTS_ECALL_SERVICE
        RST = EGTS_ECALL_SERVICE
      ▲ RD [0] = EGTS_SR_RAW_MSD_DATA
        FM = GOST_R_54620
        MSD = Byte[31] - [0x01, 0x00, 0x11, ...]

    decodingInformation = ""
    decodingStatus = Successful

```

図 3-7: MSD でデコードされた SMS

### SMS 経由におけるパラメータのアクション

ERA は、SMS 経由でパラメータを処理するために異なるアクションを定義し、PSAP は "ACT" でパラメータのアクションを設定する必要があります。例として PSAP は '1' に IVS の UNIT\_ID を設定します。

ACT (Action)	
	Command
	Command
C	Parameter request
	Parameter set
M	Parameter add
	Parameter remove



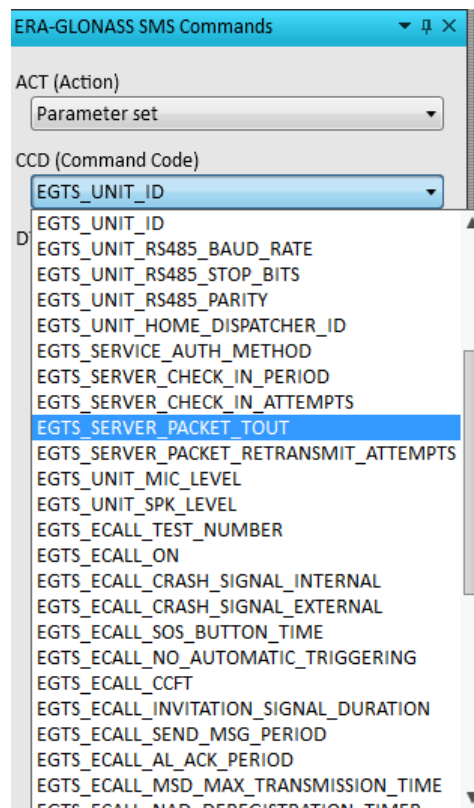


図 3-8: コマンドコードによるパラメータの設定

コマンドコード (CCD) "EGTS\_UNIT\_ID"はユニット ID (この場合は '1') を設定するよう IVS に指示します。

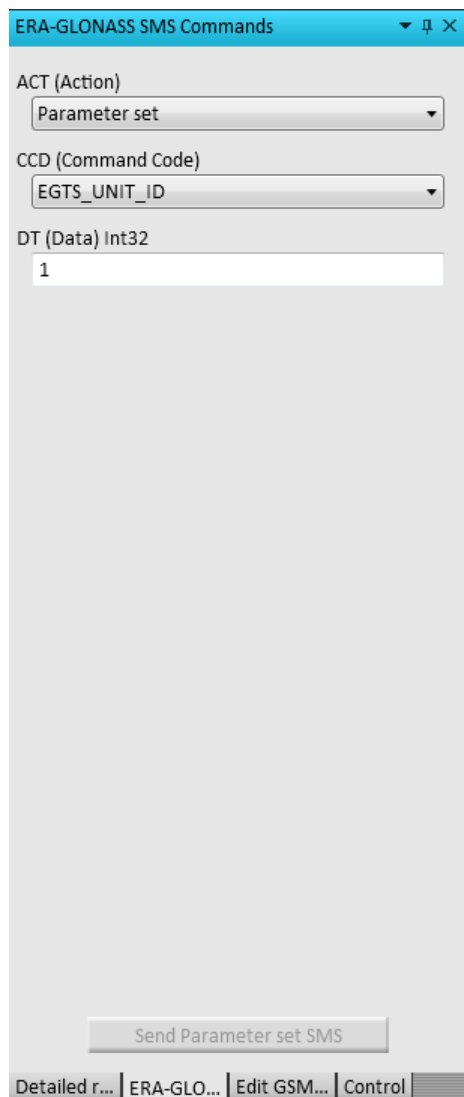


図 3-9: SMS コマンドの設定は、ID パラメータを設定します。

IVS は、SMS が送信される前に、IVS をレジスト (“synchronized”) する必要がありますのでご注意ください。

### 3.3.5 トラブルシューティング

以下のトラブルシューティングは、前述した章の手順に関連しています。

**GUI とベースの接続に問題があります。**

接続が失敗した場合は、エラーメッセージをコンソールウィンドウで確認してください。典型的なエラー：

- “No license found” CMW-KA095 ライセンスを有するスマートカードが PC に接続されていることを確認してください。スマートカードの内容は、“Windows”の[スター

ト]メニューにある“R&S License Key Manager”から確認することができます。接続を試みる前に、ライセンスキーマネージャを閉じてください。

- “Client register with <URL> failed”: “Base”メニュー内の URL が正しいことを確認してください。デフォルトの設定は、ベースと GUI が同じ PC にインストールされ、ポート 8085 が空いている状態になっております。

### 初期設定エラー

“Idle”状態から“Configured”状態への移行が失敗した場合は、“Notification History”ビューを確認してください。典型的なエラー：

- “Base rejected CMW... viOpen failed...”: CWM へのリモート制御接続エラー
  - LAN 経由で CWM と接続されていることを確認してください。例えば、CMW に PC から ping 要求を送信し確認します。
  - VISA アドレスを確認してください。
- “GSM bch number invalid for band...”または類似した表示の場合：GSM セルの設定を確認してください。

### レジストリ・エラー

GSM または WCDMA セルへの IVS のレジストリが失敗した場合は、以下の点を確認してください。

- IVS が有効になっていますか？
- IVS と CMW 間の RF ケーブに問題はありませんか？
- CMW で設定した、正しい RF コネクタを使用していますか？  
CMW の設定については、ページ上に表示されます。
- GSM または WCDMA セルの設定は、IVS に互換性がありますか？セルの設定については、ページ上に表示されます。

### MSD の送信エラー

PSAP に MSD の送信が失敗した場合は、トラブルシューティングに関する次のヒントを考慮してください。

- オーバービュー内で IVS と GSM の状態を確認します。状態は“Call Established”になっていますか？それとも“Call setup”エラーになっていますか？
- 通話が確立されると、オーバービュー内の PSAP の状態を確認してください。どの状態まで達成しますか？ MSD を受信しているが、復号することができない状態ですか？または、MSD を受信していない状態ですか？
- MSD を受信しているが、復号化できない場合は、MSD データの中身を確認してください。“Results Overview”内にある“Raw MSD”をダブルクリックしてください。
- 詳しいヒントにつきましては、“Message Trace”ビューをチェックしてください。やり取りを行なった eCall プロトコル・メッセージが表示されます。または“Log”ビューをチェックしてください。表示されていない場合は、“View” menu > “Show” > “Log”の手順でアクティブにしてください。

### 3.4 仕様に準拠した測定

IVS のためのコンフォーマンス・テストは GOST R 55530 - 2013 [1]、第 6 章 “Methods used in tests of conformity to functional requirements” に指定されています。これらのテストは、表 3-2 に記載されています。

表 3-2: コンフォーマンス・テスト

Number	Name	PSAP Settings	Check / Pass Condition
6.1	Checking MSD transfer in automatic mode		
6.1.1	Checking MSD transfer using in-band modem	Default	Check MSD contents: “Automatic activation = True”, “Test call = False”
6.1.2	Checking MSD transfer SMS	Use ERA-GLONASS SMS command: “Action= command”, “Command code = EGTS_ECALL_MSD_REQ”, “Transport channel = SMS”	Check MSD contents: “Automatic activation = True”, “Test call = False”
6.2	Checking MSD transfer in manual mode		
6.2.1	Checking MSD transfer using in-band modem	Default	Check MSD contents: “Automatic activation = False”, “Test call = False”
6.2.2	Checking MSD transfer SMS	Use ERA-GLONASS SMS command: “Action= command”, “Command code = EGTS_ECALL_MSD_REQ”, “Transport channel = SMS”	Check MSD contents: “Automatic activation = False”, “Test call = False”
6.3	Checking that transferred MSD contains last known vehicle location as for detection time of RTA event	Default	Check MSD contents: “last known vehicle location determined by GNSS (parameter vehicle_location)”
6.4	Checking that transferred MSD contains expected last known vehicle location as for detection time of RTA event	Default	Check MSD contents: no information on the last known vehicle location but “expected last known vehicle location (parameter Recent_vehicle_location_n-1 )”
6.5	Checking that transferred MSD contains valid vehicle location data	Default	Check MSD contents: “information on geographic location of the vehicle (coordinates determined by GNSS)”
6.6	Checking that transferred MSD contains vehicle movement direction data	Default	Check MSD contents: “information on geographic location of the vehicle (coordinates determined by GNSS), information on movement direction of the vehicle (parameter VehicleDirection)”

Number	Name	PSAP Settings	Check / Pass Condition
6.7	Checking that loud voice communication is possible during emergency calls	Default	check duplex voice communication during call
6.8	Checking IVS status indicators (for IVS in auxiliary equipment configuration)	not available / out of scope	
6.9	Checking IVS operation in Test mode	Default	Check MSD contents: "ECALL_TEST_NUMBER"
6.10	Checking IVS operation in "Service Station" mode (for IVS in auxiliary equipment configuration)	Default	Check MSD contents
6.11	Checking IVS operation in "Software Downloading" mode (for IVS in auxiliary equipment configuration)	not available / out of scope	
6.12	Checking UIM (for IVS in auxiliary equipment configuration)	not available / out of scope	
6.13	Checking internal memory of IVS	not available / out of scope	
6.14	Checking operation of IVS backup battery and power supply	not available / out of scope	
6.15	Checking IVS registration in network	Default	Cell Status: Registered ("synchronized")
6.16	Checking electric power supply and power consumption requirements (for IVS in auxiliary equipment configuration)	not available / Out of Scope	

表 3-3: コンフォーマンス・テスト

Number	Name	PSAP Settings	Check / Pass Condition
6.17	Checking transfer of SMS command for setting SMS transmission number when SMS is used as a redundant data channel	Use ERA-GLONASS SMS command: "Action= Parameter set", "Command code = EGTS_UNIT_ID", set wanted ID (1) and specify SMS number	check new SMS number via diagnostic software
6.18	Checking transfer of SMS command for setting emergency call number used in tests	Use ERA-GLONASS SMS command: "Action= Parameter set", "Command code = EGTS_UNIT_ID", set wanted ID (1) and specify test call number	check new test call number via diagnostic software
6.19	Checking transfer of MSD in manual activation mode using in-band modem	default	Check MSD contents: "Automatic activation = False"
6.20	Checking transfer of MSD in manual activation mode using SMS	default, choose SMS in diagnostic SW	Check SMS MSD contents: "Automatic activation = False"
6.21	Checking transfer of SMS command for initiation of test emergency call	Use ERA-GLONASS SMS command: "Action= command", "Command code = EGTS_ECALL_MSD_REQ", "Transport channel = VOICE"	Check MSD contents: "Automatic activation = False", "Test call = True"

Number	Name	PSAP Settings	Check / Pass Condition
6.22	Checking transfer of MSD with test call attribute using in-band modem	default	Check MSD contents: "Automatic activation = False", "Test call = True"
6.23	Checking transfer of MSD with test call attribute using SMS	Use ERA-GLONASS SMS command: "Action= command", "Command code = EGTS_ECALL_MSD_REQ", "Transport channel = SMS"	Check SMS MSD contents: "Automatic activation = False", "Test call = True"
6.24	Checking transfer of SMS command for repeated MSD transfer in regard to RTA event recorded earlier	Use ERA-GLONASS SMS command 2 times: "Action= command", "Command code = EGTS_ECALL_MSD_REQ", "Transport channel = SMS"	receive 2 MSD's: Check MSD contents are teh same except: location and time
6.25	Checking packet transmission of firmware data specific to a given IVS type (for IVS in auxiliary equipment configuration)'	not available / Out of Scope	
6.26	Checking that emergency call button is protected from accidental pressing	not available / Out of Scope	
6.27	Checking that backlighting of emergency call button is available	not available / Out of Scope	

### 3.5 自動テスト：リモートコントロール

CMW-KA09x ソフトウェアは、主要部分、および手動制御（KA09x GUI アプリケーションソフトウェア）用の GUI である KA09x ベースのアプリケーション・ソフトウェアが付属しています。GUI は、SOAP を介してベースを制御します。自動テストを実行するには、開発したプログラムにより SOAP を介したベースの制御または、ローデ・シュワルツが提供する CMWrun によって実行することができます。

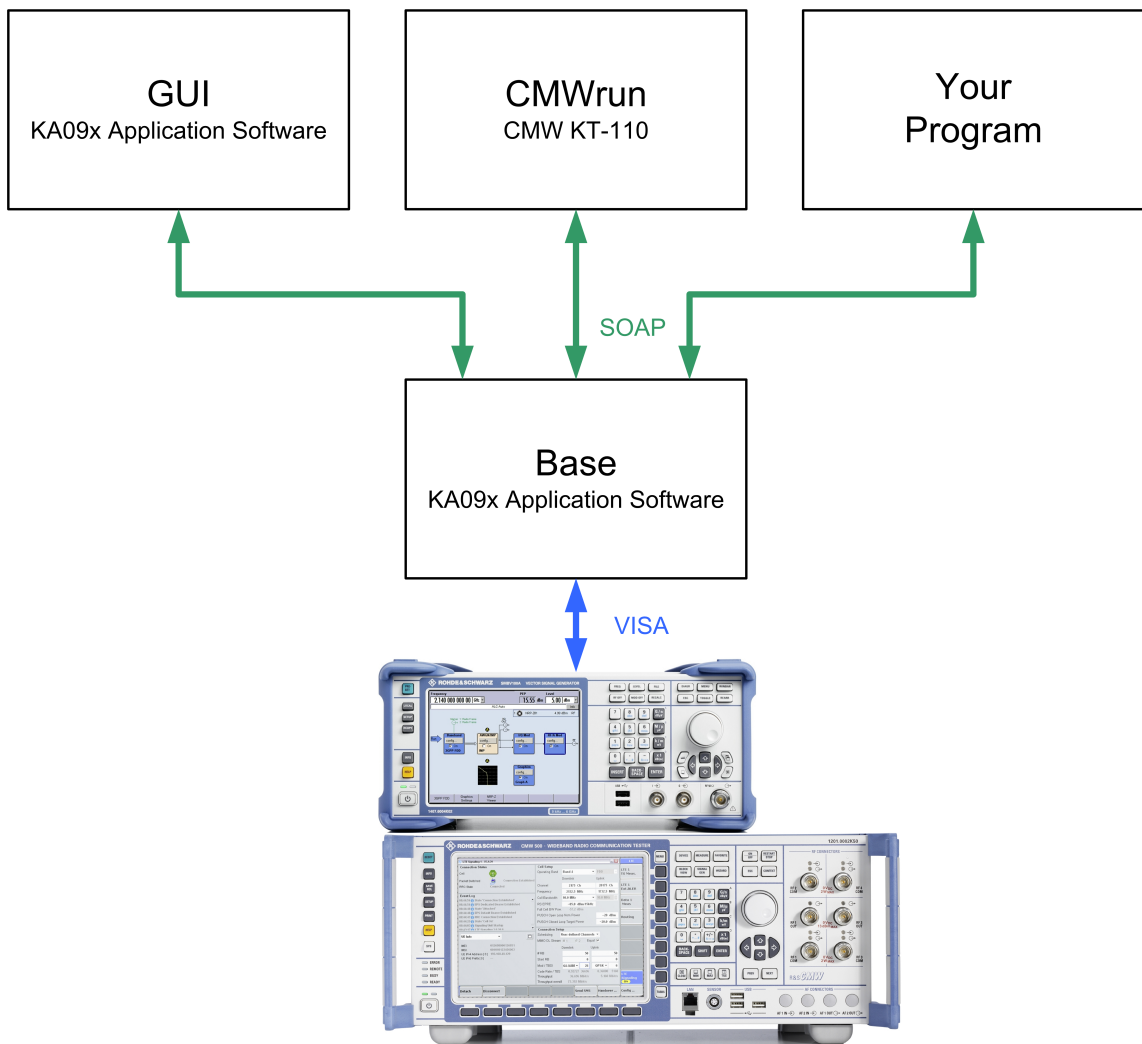


図 3-10: SOAP を介してベースを制御する 3 つの方法があります。手動制御のための GUI が KA09x 一部としてインストールされている BASE が提供されています。自動計測に関しては、CMWrun を使用するか、独自のプログラムを使用することで可能になります。

### 3.5.1 CMWrun(オプション：KT-110)

CMWrun は CMW がサポートする各種規格を、リモートで制御するためのテストシーケンスを構成する自動化ソフトウェアです。ソフトウェアエンジンは、テスト DLL (プラグインのアセンブリ) の実行に基づいています。このアーキテクチャは、特定のリモートプログラミングの知識がなくてもテストシーケンスを簡単に設定することが可能です。また、各種規格の CMWrun パッケージオプションを使用することで、テスト項目のパラメータを制限なく設定でき、柔軟にシーケンスを組むことができます。テスト終了時に、テスト結果を確認することができます。また、CSV、TXT、XML や PDF で試験報告書を出力することもできます。

リモート制御のための CMWrun のオプションである KT110 は、CEN/TS 16454:2012 および GOST 55530 準拠したコンフォーマンス・テストに必要な ERA-GLONASS の設定を簡単にこなうことができます。また、各種 IVS タイプの取り扱いが可能で、pass/fail テス

トプロトコルを容易に確認することが可能です。さらに、KT110 ソフトウェアは AT コマンドを用いることでテスト・デバイスの自動化を行なうこともできます。

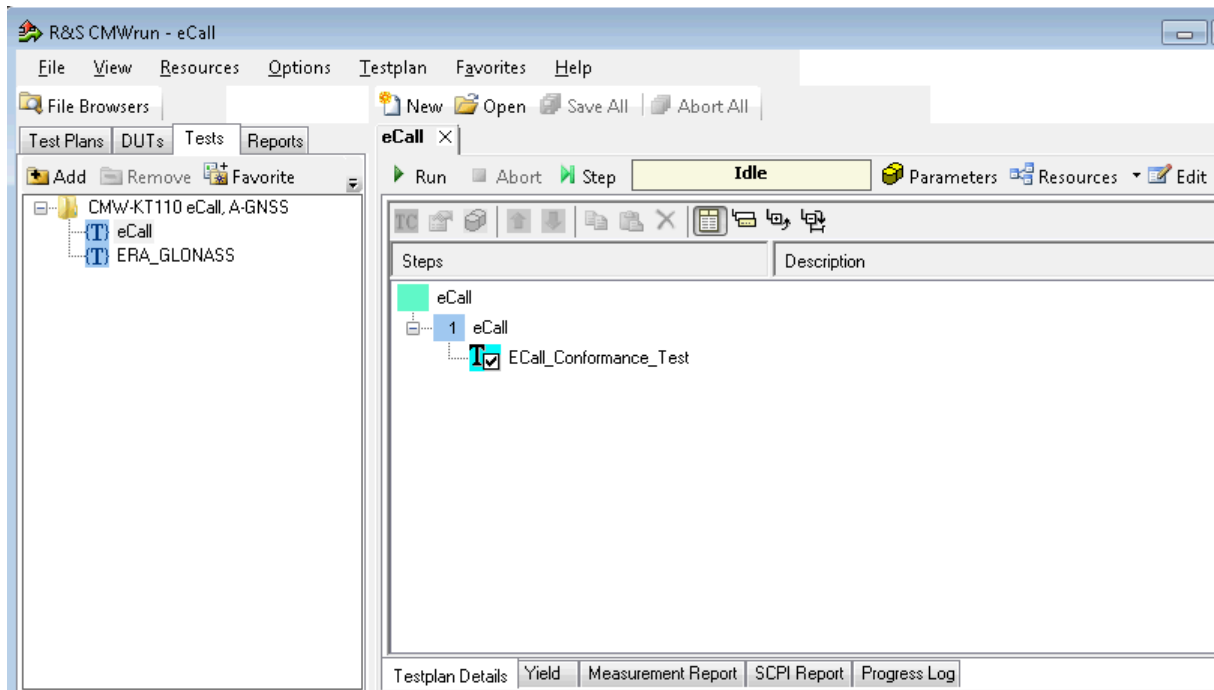


図 3-11: CMWrun のメインビュー (オプション: KT-110 は eCall 及び ERA-GLONASS 両方に対応)

### 3.5.2 SOAP を介したベースのリモートコントロール

自動テストシステムの ERA-GLONASS ソリューションを使用するには、リモート KA09x ベースのアプリケーション・ソフトウェアをコントロールする必要があります。SOAP は、情報やメッセージを交換するためのオープンな W3C 標準プロトコルです。リモートコントロール用の基本ソフトウェアは、API を提供します。下記リンクを参照ください。  
 %APPDATA%\Rohde-Schwarz\CMW-KA09x\DOC\CMW-KA09x\_API\_en.chm  
 コンフィギュレーション・データ (例: GSM チャンネル番号) を送信し、データ (例: 送信された MSD) を受信することができます。

デフォルトの uri をもつ "KA09x\_Base.exe" は既に実行されている必要がありますのでご注意ください。また、マニュアル、第 3 章を参照してください[3]

KA09x ソフトウェアをインストールしていただくことで、C# (.NET) で記述された例が下記リンク先にプロジェクトとして保存されます。

%APPDATA%\Rohde-Schwarz\CMW-KA09x\Complementary\KA09x\_Example1



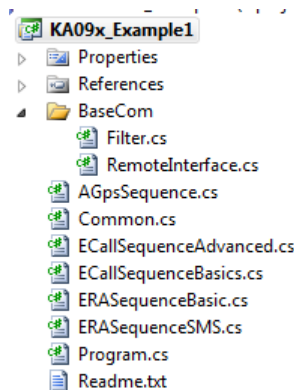


図 3-12: サンプルプロジェクト内のファイル

プロジェクトは、以下のファイルで構成されています：

- **Program**：メインプログラム・ファイルです。ベースに接続し、eCall または ERA、Basic または Advanced のいずれかを呼び出します。
- **Common**：CMW または SMBV のための VISA アドレスの基本的な設定を提供します。
- **BaseCom/RemoteInterface**：ベースに、SOAP 経由で基本的な通信（接続およびメッセージ）を処理します。
- **BaseCom/Filter**：ベースからクライアントへの着信イベントを処理します。たとえば、ベースのステートマシンの変化を待つことができます。
- special examples:
  - **ERASequenceBasic**：基本的な ERA-GLONASS 通話（詳細は下記参照）
  - **ERASequenceSMS**：ERA-GLONASS の SMS の例（詳細は下記参照）
  - **ECallSequenceBasic**：ベーシック eCall
  - **ECallSequenceAdvanced**：アドバンスド eCall

### リモートインタフェース

クラスは、ベースとの通信のためのクライアント・インタフェースを実装しています。また、.NET が提供する SOAP を使用しています。ベースから着信イベント/メッセージの制御、接続/切断の実行そしてベースに'keep alive'が送信されます。詳細は [4.2.1, 「RemoteInterface」](#)（50 ページ）を参照ください。

### フィルター

クラスは、ベースとの通信のためのクライアント・インターフェースを実装しています。また、.NET が提供する SOAP 実装し使用しています。ベースから着信イベント/メッセージの制御、接続/切断の実行そしてベースに'keep alive'が送信されます。詳細は [図 3-13](#) を参照ください。方法につきましては、EraSequenceBasic を参照ください。

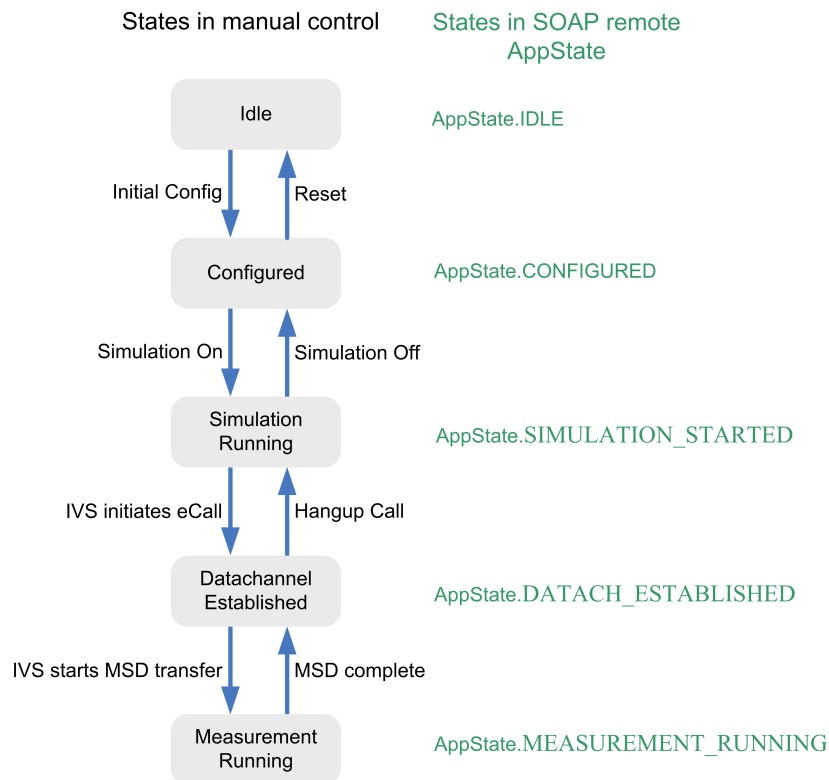


図 3-13: ベースの異なる状態。リモートコントロールでは、*appstate* を照会することができます。すべての状態の変化は、制御ソフトウェアで処理できるイベント 'NewState' を作成します。

## コモン

コモン・クラスは、すべての例に共通し使用される機器の基本的な設定を提供します。また、VISA アドレスまたは使用されるコネクタのように、使用される RAN または GNSS の設定とは無関係です。

機器やシミュレーションを行なう際に、特別な ID が定義されています：

```

// The IDs are used to address the individual systems
// like RANs, devices, services, etc...
public const UInt16 RAN_SIM_ID = 1;
public const UInt16 RAN_ID = 2;
public const UInt16 PSAP_ID = 3;
public const UInt16 GNSS_SIM_ID = 4;
public const UInt16 GNSS_ID = 5;
  
```

CMW の構成例。通話がシーケンスに沿って行われます。

```

public static void useCmw(Filter f, RemoteInterface r)
{
    System.Console.WriteLine("Will send CfgCmw");
    EventStatus ret = r.toBase.sendEvent(new CfgCmw()
    {
        targetId = RAN_SIM_ID,
        rfConnector = CmwRfConnector.RF1C,
        visaName = "TCPIP::CMW50050-123456::INST0::INSTR"
    });
    CfgResult res = f.waitFor<CfgResult>();
    throwIfUnsuccessfull(res);
}

```

1. 属性 ID、コネクタ、VISA 名を有するベースに 'CfgCmw' イベントを送信します。
2. 結果を待ちます。(event = 'CfgResult')

### Program

メインプログラムを示します。SOAP を介しベースに接続します。そして、所望するシーケンスを選択します。

```

class Program
{
    /// <summary>
    /// To use this example a 'KA09x_Base.exe' with the default uri (see below) must be running.
    /// </summary>
    /// <param name="args"></param>
    static void Main(string[] args)
    {
        RemoteInterface r = new RemoteInterface();
        Filter f = new Filter(r);
        Ka09xComponentVersion myVersion = new Ka09xComponentVersion()
        {
            major = 1,
            minor = 0,
            patchLevel = 0
        };
    };
}

```

最初に、リモート・インターフェースの型の新しいオブジェクト名 'r' を、フィルタータイプの新しいオブジェクト名 'f' を作成します。

```

try
{
    Console.WriteLine("Will connect now ...");
    r.connect(KA09x_SoapV3.SoapInterfaceConsts.DEFAULT_BASE_SERVICE_URL,
        KA09x_SoapV3.SoapInterfaceConsts.DEFAULT_CLIENT_SERVICE_URL,
        myVersion, "KA09x_Example1");

    // here you can define which sequence to use
    Console.WriteLine("Connected, will do sequence ...");

    //ECallSequenceBasics.doSequence(f, r);

    //ECallSequenceAdvanced.doSequence(f,r); // for a more advanced scenario

    ERASequenceBasic.doSequence(f,r); // an ERA-GLONASS basic

    Console.WriteLine("Finished, bye!");
}
}

```

'r' の 'connect' を介しベースに接続します。ベースのデフォルト URL が使用されます。所望のシーケンスが呼び出されます。(このケースでは、"ERASequenceBasic" が呼び出されます。)

## ERASequenceBasic

シーケンスは ERA の例を実行します：

- ベースのリセット

```
// Reset first to get to defined state
System.Console.WriteLine("Will reset the base ...");
ret = r.toBase.sendEvent(new ResetBase());
Common.throwIfUnsuccessful(ret);
f.waitFor<BaseDidReset>(TimeSpan.FromSeconds(10));
System.Console.WriteLine("Done base back in idle");
```

1. 'ResetBase' イベントの送信
2. リセットは 10 秒のタイムアウトで終了するまで待ちます。(event = 'BaseDidReset')

- 基本構成は 'InitialConfig' で実行されます。

```
// Sets up the most basic configuration
initialConfig(f, r);
```

- セットアップの構成(使用する機器)

```
System.Console.WriteLine("Will send CfgSetup");
ret = r.toBase.sendEvent(new CfgSetup()
{
    ranDevs = new RanSimulator[]
    {
        new RanSimulator()
        {
            simId = Common.RAN_SIM_ID,
            simulatedSystems = new RanSystem[]
            {
                new RanSystem()
                {
                    idRan = Common.RAN_ID,
                    simulatePsap = true,
                    idPsap = Common.PSAP_ID
                }
            }
        }
    },
    gnssDevs = new GnssSimulator[]
    {
        new GnssSimulator()
        {
            simId = Common.GNSS_SIM_ID,
            simulatedSystems = new GnsSystem[]
            {
                new GnsSystem()
                {
                    idGnss = Common.GNSS_ID,
                }
            }
        }
    }
});
Common.throwIfUnsuccessful(ret);
res = f.waitFor<CfgResult>(TimeSpan.FromSeconds(30));
Common.throwIfUnsuccessful(res);
```

1. イベント 'CfgSetup' を 'common' で定義された属性を持つベースに送ります。
  2. 30 秒間結果を待ちます。(event = 'CfgResult')
- どの CMW と SMBV を使用しているか ('common' で定義された)

```
// Note: Configure GNSS before RAN
Common.useSmbv(f, r);

// Configure the base to use the CMW, if you want to use the CMU comment the next line
Common.useCmw(f, r);
```

– GNSS シナリオ構成 (GLONASS: City Scenario Moscow)

```
System.Console.WriteLine("Will send CfgGlonassCity");
ret = r.toBase.sendEvent(new CfgGlonassCity()
{
    targetId = Common.GNSS_ID,
    location = PredefGnssReceiverCityLocation.MOSCOW,
});
res = f.waitFor<CfgResult>();
Common.throwIfUnsuccessful(res);
```

1. イベント 'CfgGlonassCity' を 'Moscow' で定義された属性を持つベースに送ります。
2. 結果を待ちます。(event = 'CfgResult')

– GSM セル設定の構成

```
// Now configure the GSM cell
System.Console.WriteLine("Will send CfgGsm");
ret = r.toBase.sendEvent(new CfgGsm()
{
    targetId = Common.RAN_ID,
    band = GsmBand.GSM900,
    bcchNum = 64,
    tchNum = 32,
});
res = f.waitFor<CfgResult>();
Common.throwIfUnsuccessful(res);
```

1. イベント 'CfgGSM' を 'GsmBand and Channels' で定義された属性を持つベースに送ります。
2. 結果を待ちます。(event = 'CfgResult')

– PSAP 設定の構成

```
// Now configure the PSAP
System.Console.WriteLine("Will send CfgPsap");
ret = r.toBase.sendEvent(new CfgEraGlonassPsap()
{
    targetId = Common.PSAP_ID,
    alAck = ALAckBehavior.POSITIVE,
    scenario = PsapModemTestScenario.ALLRV_REQ,
    storeRecordedSignal = true,
    playToneAfterMs = true,
    forceRobustModType = false
});
res = f.waitFor<CfgResult>();
Common.throwIfUnsuccessful(res);
```

1. イベント 'CfgEraGlonassPsap' を異なる属性を持つベースに送ります。
  2. 結果を待ちます。(event = 'CfgResult')
- (すべての使用される機器のライセンスのチェックと設定を含む) すべての設定を実行し、すべてが完了するまで待ちます。

```
// Transition to configured
System.Console.WriteLine("Will send DoConfiguration");
ret = r.toBase.sendEvent(new DoConfiguration());
Common.ThrowIfUnsuccessful(ret);
nState = f.waitFor<NewState>(TimeSpan.FromMinutes(2)); // initial setup can take quite long
if (nState.currentState != AppState.CONFIGURED)
    throw new Exception("DoConfiguration failed!");
```

1. InitialConfig で構成した後、イベント 'DoConfiguration' を送ります。
2. 2分のタイムアウトを持つ状態遷移を待ちます (event = 'NewState')
3. 現在の状態が 'CONFIGURED' であるかを確認し、そうでない場合は、何かが間違っています。

- シミュレーションの開始

```
// Start the simulation
System.Console.WriteLine("Will send SimulationOn to start simulation");
ret = r.toBase.sendEvent(new SimulationOn());
Common.ThrowIfUnsuccessful(ret);
nState = f.waitFor<NewState>();
if (nState.currentState != AppState.SIMULATION_STARTED)
    throw new Exception("SimulationOn failed!");
```

1. イベント 'SimulationOn' を ON にすることでシミュレーションを開始します。
2. 状態遷移を待ちます (event = 'NewState', default timeout)
3. 現在の状態が 'SIMULATION\_STARTED' であるかを確認し、そうでない場合は、何かが間違っています。

- ERA-GLONASS コールを待ちます。

```
// Wait for call
System.Console.WriteLine("Waiting for call ...");
nState = f.waitFor<NewState>(TimeSpan.FromMinutes(30));
if (nState.currentState != AppState.DATACH_ESTABLISHED)
    throw new Exception("Call somehow failed!");
```

1. 送信するために、クライアントからのイベントは、通話を IVS によって初期化する必要はありません。
2. 状態遷移を待ちます (event = 'NewState', default timeout)
3. 現在の状態が 'DATACH\_ESTABLISHED' であるかを確認し、そうでない場合は、何かが間違っています。

- MSD の受信

MSD の受信は、次の 3 つの部分で行われます。:

- MSD の送信の開始を待ちます。

```
// Wait for msd start
System.Console.WriteLine("Waiting for MSD tranfere ...");
nState = f.waitFor<NewState>(TimeSpan.FromMinutes(1));
if (nState.currentState != AppState.MEASUREMENT_RUNNING)
    throw new Exception("Sending MSD somehow failed!");
```

1. 状態遷移を待ちます (event = 'NewState', default timeout)
2. 現在の状態が 'MEASUREMENT\_RUNNING' (MSD 転送が進行中です) であるかを確認し、そうでない場合は、何かが間違っています。

- 全ての MSD が正常に受信されるまで待ちます ('waitForSuccessfulMSD')

```

static RawMsVoice waitForSuccessfulMsD(Filter f)
{
    RawMsVoice rawMsD;

    do
    {
        System.Console.WriteLine("Waiting for raw MSD result ...");
        rawMsD = f.waitFor<RawMsVoice>(TimeSpan.FromSeconds(10));
        System.Console.WriteLine("RawMsD: modulation={0}, RV={1}",
            rawMsD.modulationType, rawMsD.redundancyVersion);

        if (rawMsD.msD != null)
            return rawMsD;

        System.Console.WriteLine("RawMsD: transmission failed");
    }
    while (rawMsD.redundancyVersion < 8);

    throw new Exception("MsD transmission failed after last redundancy version");
}

```

1. IVS は、8 最大倍の異なる冗長バージョンに MSD を送信できます。DO-while ループを使用し、プログラムによって処理されます。
2. 10 秒間、状態が 'RawMsVoice' になるのを待ちます。
3. PSAP が正常に MSD を受信することができる場合は、'rawMsD.msD' は NULL とは異なり、メインプログラムに戻ります。そうでない場合は、ループの状態になります。

#### – デコードされた MSD 部出力

```

System.Console.WriteLine("Waiting for decoded MSD result ...");
DecodedEraGlonassMsDv1 decMsD = f.waitFor<DecodedEraGlonassMsDv1>(); // This expected the IVS to send an
//DecodedECallMsDv1 decMsD = f.waitFor<DecodedECallMsDv1>(); // This expected the IVS to send an MSD ver
System.Console.WriteLine("DecodedMSD: id={0} isowmi='{1}' AddData = '{2}'",
    decMsD.message.id,
    decMsD.message.msD.msDStructure.vehicleIdentificationNumber.isowmi,
    decMsD.message.msD.optionalAdditionalData);

```

1. イベント 'DecodedEraGlonassMsDv1' を待ちます。
2. デコードされた MSD を出力します。(V1 期待され、IVS に依存)

#### ● ACKs の待ち

```

// Wait for LL+HL/AL Ack to be fully sent
System.Console.WriteLine("Waiting for LL+AL/HL ...");
nState = f.waitFor<NewState>(TimeSpan.FromSeconds(10));
if (nState.currentState != AppState.DATACH_ESTABLISHED)
    throw new Exception("Call somehow failed!");

```

1. PSAP が MSD の復号に成功した後、ACK's を IVS に送信しクライアントからのイベントが、送信しないようにします。
2. 状態遷移を待ちます。(event = 'NewState', timeout 10 seconds)
3. 現在の状態が 'DATACH\_ESTABLISHED' であるかを確認し、そうでない場合は、何かが間違っています。

#### ERA シーケンス SMS

このシーケンスは、MSD の要求に ERA の SMS の例を実行します。

最初のステップは "ERASquenceBasic" のように同じであることに注意してください：

- ベースの初期化
- 基本的な構成は "InitialConfig" で行われます。

- (すべての使用される機器のライセンスのチェックと設定を含む) すべての設定を実行し、すべてが完了するまで待ちます。
- シミュレーションの開始

SMS の追加の手順があります。

- ネットワークへの IVS の登録を待ちます。

```
// Please swith on the IVS
System.Console.WriteLine("=====");
System.Console.WriteLine("Please switch on your IVS");
System.Console.WriteLine("=====");

// Wait for register
System.Console.WriteLine("Waiting for registration ...");
do {
    CMWGSMSState = f.waitFor<GsmCellSignalStateCmw>(TimeSpan.FromMinutes(1)); // ask for the Status of th

} while (CMWGSMSState.currentDetailState != GsmCellSigStateCmw.SYNC);
```

1. do...while ループでは、セルの状態が照会されます。(イベント "GsmCellSignalStateCmw" を待ちます。)

2. ステータスが 'SYNC' である場合、ループを終了します。

- IVS に PSAP から SMS コマンドを定義します。

```
// define the SMS command
EraGlonassSmsCommand msdReq = new KA09x_SoapV3.EraGlonass.EraGlonassSmsCommand()
{
    targetId = Common.PSAP_ID,
    data = new EGTS_ECALL_MSD_REQ()
    {
        MID = EGTS_ECALL_MSD_REQ.NEW_MESSAGE,
        TRANSPORT = Transport.SMS
    }
};
```

- 'EGTS\_ECAL\_MSD\_REQ' とトランスポートチャンネル 'SMS' を指定してコマンドを送信します。(「SMS を介した MSD」 (29 ページ) 参照)

- SMS の着信を待ちます。

```
System.Console.WriteLine("Waiting for Incoming SMS...");
IncomingSms IncSMS = null;
try
{
    // wait for an incoming SMS for 1 minute
    IncSMS = f.waitFor<IncomingSms>(TimeSpan.FromMinutes(1));
}
catch (TimeoutException e)
{
    // if timeout expires, send SMS command again
    r.toBase.sendEvent(msdReq);
    IncSMS = f.waitFor<IncomingSms>(TimeSpan.FromMinutes(1));
}
```

1. SMS の着信を待ちます。(一定時間イベント "IncomingSms" を待ちます)

2. タイムアウトが満了した場合、PSAP は再び SMS コマンドを送信する必要があり、再び受信する SMS を待ちます。(一定時間イベント 'IncomingSms' 待ちます)

- SMS の着信を表示します。(実際の SMS)



```
// show the incoming SMS (raw data)
System.Console.WriteLine("IncomingSMS: Information={0} Status = '{1}'",
    IncSMS.decodingInformation,
    IncSMS.decodingStatus);
```

- SMS のデコード

```
// check the decoding: it can be fully successful or partly successful
if ((IncSMS.decodingStatus == DecodingInfo.Successful) || (IncSMS.decodingStatus == DecodingInfo.WithProblem))
{
    // see also Table 3 and 14 of GOST 54619
    // check if the Service Frame Data (SFRD) is present
    if (IncSMS.pdu.SFRD.present) {
        EGTS_PT_APPDATA appData = IncSMS.pdu.SFRD.optFieldData as EGTS_PT_APPDATA;

        // check if the Service Data RECORD (SDR) is present
        if (appData != null && appData.SDR.present)
        {
            // check if there are data ind Record Data (RD)
            if (appData.SDR.optFieldData[0].RD.Length != 0) {

                // we believe that the MSD is in RD (see table 14)
                EGTS_SR_RAW_MSD_DATA subrecordRawMSD =
                    appData.SDR.optFieldData[0].RD[0] as EGTS_SR_RAW_MSD_DATA;

                if (subrecordRawMSD != null) {

                    // This expected the IVS to send an MSD version 1, this solely depends on the IVS
                    System.Console.WriteLine("Waiting for decoded MSD result ...");
                    DecodedEraGlonassMsdV1 decMsd = f.waitFor<DecodedEraGlonassMsdV1>();

                    System.Console.WriteLine("DecodedMSD: id={0} isowmi='{1}' AddData = '{2}'",
                        decMsd.message.id,
                        decMsd.message.msd.msdStructure.vehicleIdentificationNumber.isowmi,
                        decMsd.message.msd.optionalAdditionalData);
                    int Latitude = decMsd.message.msd.msdStructure.vehicleLocation.positionLatitude;
                    int Longitude = decMsd.message.msd.msdStructure.vehicleLocation.positionLongitude;
                    System.Console.WriteLine("Longitude = {0}, Latitude = {1}", Latitude, Longitude);

                }
            }
        }
    }
}
```

1. 復号ステータスが 'successful' または 'with problems' であり、両方とも OK です。（'with problems' は OK 意味するが、CRC が正しくありません）
2. SFRD 部分が存在するか確認します。（この部分はオプションです、[5]の表 3 を参照ください。）
3. SDR の一部が存在するかどうかを確認します。いくつかの SDR から構成されます。（SDR's はオプションです。[5]の表 3 を参照ください。）
4. SDR が存在する場合、1 つ以上（必須）の RD's を含みます。
5. RD が存在する場合、最初に MSD を期待するデータの一部を含みます (RD[0])
6. イベント "DecodedEraGlonassMsdV1" を待ちます。
7. デコードされた MSD を出力します。（V1 が期待されているが、IVS に依存します）。データは "msdStructure" に格納されています。一例として位置（経度と緯度）が示されています。

## 4 Appendix

### 4.1 用語集

AL-ACK	ACKnowledgment on the Application Layer
BPPM	Bipolar Pulse Position Modulation
CRC	Cyclic Redundancy Check
DL	Downlink
ERA	Emergency Road Assistance
GNSS	Global Navigation Satellite System
GLONASS	Globalnaya navigatsionnaya sputnikovaya sistema
GPS	Global Positioning System
GSM	Global System for Mobile Communications
IVS	In-Vehicle System
LL-ACK	ACKnowledgment on the Link Layer
MSD	Minimum Set of Data
PSAP	Public Safety Answering Point
RV	Redundancy Version
UL	Uplink
WCDMA	Wideband Code Division Multiple Access

### 4.2 SOAP クライアントの実装

リモート制御の例は、.NET で提供される SOAP クライアント (ページ 40)、[3.5.2, 「SOAP を介したベースのリモートコントロール」](#) (40 ページ) に記載されています。クラス "RemoteInterface" に実装されています。クラスフィルタは、着信イベントを簡素化します。どちらも、この章で詳しく説明します。

#### 4.2.1 RemoteInterface

このクラスは、ベースと通信するためのクライアント・インタフェースを実装しています。これは、ベースから着信イベント/メッセージを処理し、接続/切断を行います。

主なメンバーは、Windows 通信基盤 (WCF) によって提供される SOAP サーバーに由来します：

```
// Basically that is our soap server for callbacks from base
private ServiceHost host = null;

// Proxy factory and proxy object to send events to base ...
private ChannelFactory<IToBase32> eventSenderFactory = null;
private IToBase32 eventSender = null;
```

図 4-1: 'host' は DOT.NET 経由で SOAP 実装の主要メンバーです。タイプ 'IToBase32' の eventSender は、ベースに送信するイベントを処理します。

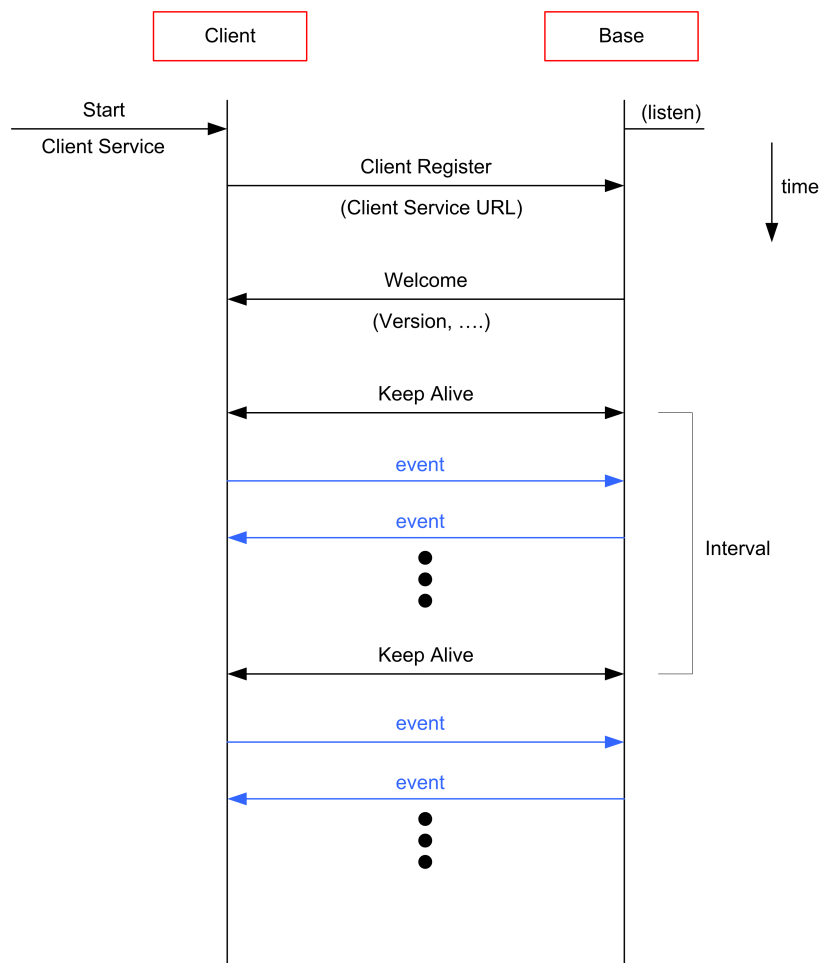


図 4-2: SOAP での基本的な通信

1. ベースはアイドル状態にあり、着信イベントを待ちます。
2. まず 'thew' のクライアント側は、そのサービス ('StartClientService') を開始します。
3. クライアントは自分の URL をベースに登録する必要があります。
4. ベースは、'Welcome' と回答します。
5. ベースとクライアント、どちらも特定の間隔で 'KeepAlive' を送信する必要があります。それ以外の場合は、基本適に接続を切断します。

6. 両方とも、イベントを送信することができます。

最初の4つの基本的な手順（ステップ2は、...5）は、“RemoteInterface”の機能“Connect”で処理されます。

```
public void connect(string baseServiceUrl, string myServiceUrl, Ka09xComponentVersion myVersion, string myName)
{
    if (connected)
        throw new InvalidOperationException("We are already connected");

    // First clear the incoming event queue
    clearIncomingEventQueues();

    // Open our client service for callbacks / messages / events from base
    openMySoapServer(myServiceUrl);

    // Build the proxy class for calls / messages / events to the base soap server
    buildToBaseSoapProxy(baseServiceUrl);

    // Finally register with the base and tell it to connect to our soap server
    registerWithBase(myServiceUrl, myVersion, myName);

    // We are connected, but to stay connected, we need to send keepalives
    startKeepAlive();
}
```

図 4-3: 'Connect' の方法

- 最初の着信イベントのキューは、クリアされます（“clearIncomingEventQueue”）。そのため、クライアントは空のキューで始まります。
- メソッド“openMySoapServer”は、クライアントサービスを開始します。（ステップ2）
- メソッド“buildToBaseSoapProxy”が、ベースに通信するためのチャンネルを開きます。
- ベースでの登録は、“registerWithBase”（ステップ3）を介して行われます。それは自動的にベースによって送信された“Welcome”のメッセージを待ちます。（ステップ4）
- “Welcome”のメッセージ後、クライアントは“KeepAlive”メッセージ（メソッド“startKeepAlive”）を送信するために開始します。これは別のスレッドで一定の間隔で自動的行われます。

```
void startKeepAlive()
{
    lock (keepAliveTimerLock)
    {
        Console.WriteLine("Will start sending keepalive");
        if (keepAliveTimer != null)
            return;

        keepAliveTimer = new System.Threading.Timer(sendKeepalive, null, 0,
            KA09x_SoapV3.SoapInterfaceConsts.KEEP_ALIVE_INTERVAL_IN_MS);
    }
}
```

ベースによって送信されたイベントは、メソッド“handleIncomingEvent”で処理されません。:

```
EventStatus handleIncommingEvent(EventBase eventData)
{
    try
    {
        Console.WriteLine("Rec: " + eventData.GetType() + " ");
        // Every event we receive, we just put into the queue
        incommingEventQueue.Add(eventData);
        Console.WriteLine("Ok");

        // We just always say everything was ok
        // the base is not really interested what we do with events;
        // it's enough to say we received it
        return okEventStatus();
    }
    catch(Exception e)
    {
        Console.WriteLine("Err '{0}'", e.Message);
        return new EventStatus()
        {
            successfullyDelivered = false,
            description = e.Message
        };
    }
}
```

図 4-4: 着信イベントは、単にキューに追加されています。

すべての着信イベントは、ちょうどこのように、シリアル化され、キューに追加されます。それらの制御(照会)は、クラス“Filter”で行ないます。

## 4.2.2 Filter

クラスフィルタは、特定の着信イベントを待つことができます。ベースがクライアントに送信するイベントは、非同期または予期せずに起こるかもしれません。クラスは、特定のイベントを待機させるメソッド“WaitFor”を提供します。例として、基本ソフトウェアのステートマシンをトラッキングするさいに、非常に便利です。

メンバー“backlog”は、処理されるのを待っているイベントのリストです：

```
const int MAX_BACKLOG_SIZE = 100;

List<EventBase> backlog = new List<EventBase>();
```

図 4-5: backlog と呼ばれるイベントのリスト

メソッド“waitFor”は、希望のイベントのための“backlog”を最初にチェックします。リストアップされていない場合は、それが受信されるまで待機します。

```

public EventBase waitFor(Type eventTypeToWaitFor, TimeSpan timeout)
{
    // first check the backlog for the wanted event
    for (int i = 0; i < backlog.Count; ++i)
    {
        EventBase oldEvent = backlog.ElementAt(i);
        if ( eventTypeToWaitFor.IsAssignableFrom( oldEvent.GetType()))
        {
            // that was the event we were waiting for
            backlog.RemoveAt(i);
            Console.WriteLine("waitFor(" + eventTypeToWaitFor + ") -> got it from backlog");
            return oldEvent;
        }
    }

    // The event we are waiting for, is not in the backlog,
    // so we have to block on the remote interface with
    // remote.getNextEvent() until we received it.
    while (true)
    {
        Console.WriteLine("waitFor(" + eventTypeToWaitFor + ") ...");
        EventBase newEvent = remote.receiveNextEvent(timeout); // very long timeout ...
        if (eventTypeToWaitFor.IsAssignableFrom( newEvent.GetType() ))
        {
            // that was the event we were waiting for
            return newEvent;
        }

        // we got an event, but it was not the type we are waiting for ...
        // ... so put into backlog for later processing
        backlog.Add(newEvent);
        if (backlog.Count > MAX_BACKLOG_SIZE)
        {
            // backlog too long, remove oldest entry
            Console.WriteLine("Backlog full, will discard " + backlog.ElementAt(0).GetType().Name);
            backlog.RemoveAt(0);
        }
    }
}

```

図 4-6: メソッド‘waitFor’

より便利な方法を提供するために、一般的な方法は、ラッパーとして提供されます。

```

public EVENT_TYPE waitFor<EVENT_TYPE>(TimeSpan timeout)
    where EVENT_TYPE: EventBase
{
    return (EVENT_TYPE)waitFor(typeof(EVENT_TYPE), timeout);
}

/// <see cref="EVENT_TYPE waitFor<EVENT_TYPE>(TimeSpan timeout)"/>
public EVENT_TYPE waitFor<EVENT_TYPE>()
    where EVENT_TYPE : EventBase
{
    return (EVENT_TYPE)waitFor(typeof(EVENT_TYPE), DEFAULT_TIMEOUT);
}

```

図 4-7: 一般的なラッパー‘waitFor’

### 4.3 参考文献

[1] CEN: CEN/TS 16454:2012: Intelligent transport systems – ESafety – ECall end to end conformance testing, December 2012

[2] 3GPP: TS26.267 :Technical Specification Group Services and System Aspects; eCall Data Transfer; In-band modem solution; General description V12.0.0., December 2012

[3] User Manual: Test Software for eCall, CMW-KA094, Rohde & Schwarz, 2013

[4] GOST R 55530-2013: Road Accident Emergency Response System: Functional test methods of IVS and data transfer protocols

[5] GOST R 54619-2011: Road Accident Emergency Response System: Protocol of Data Transmission from In-Vehicle Emergency Call System/Device to Emergency Response System Infrastructure

## 4.4 追加情報

このホワイトペーパーに関するご意見やご提案をお送りください

TM-Applications@rohde-schwarz.com

## 4.5 オータ情報

CMW500 Radio Communication Tester		
Base Unit	CMW500	
CMW500 Mainframe 03	CMW-PS503	1208.7154.02
Front Panel with Display H600B	CMW-S600B	1201.0102.03
BB Flexible Link H550B	CMW-S550B	1202.4801.03
RF Frontend (Basic) H590A	CMW-S590A	1202.5108.02
Signaling Unit Universal B200A	CMW-B200A	1202.6104.02
GSM Signaling option	CMW-B210A	1202.6204.02
Signaling Unit Wideband, B300B	CMW-B300B	1202.6304.03
Audio Analyzer/Generator H400B	CMW-B400B	1207.8457.02
Speech codec H405A	CMW-B405A	1207.8257.02
GSM GPRS EDGE Rel. 6, Basic signaling	CMW-KS200	1203.0600.02
WCDMA R99, basic signaling	CMW-KS400	1203.0751.02
Hardware set for eCall applications (acc.)	CMW-Z94	1208.6906.02

<b>CMW500 Software Licensing</b>		
CMWPC Smartcard	CMWPC	1201.0002K90
USB Smartcard for CMWPC	CMW-S089A	1202.7900.02
eCall test software	CMW-KA094	1208.4703.02
ERA-GLONASS test software	CMW-KA095	1208.8844.02

<b>CMWrun</b>		
CMWrun eCall	CMW-KT110	1208.7431.02

<b>SMBV100A Vector Signal Generator</b>		
Base Unit	SMBV100A	1407.6004.02
Baseband Generator	SMBV-B10	1407.8607.02
Hard Disk (removable)	SMBV-B92	1407.9403.02
RF Path	SMBV-B103	1407.9603.02
GPS	SMBV-K44	1415.8060.02
Assisted GPS	SMBV-K65	1415.8560.02
GLONASS	SMBV-K94	1415.8677.02
Assisted GLONASS	SMBV-K95	1419.2521.02
Galileo	SMBV-K66	1415.8590.02
Beidou	SMBV-K107	1419.2709.02
QZSS	SMBV-K105	1419.2350.02
GNSS Enhanced	SMBV-K92	1415.8583.02
GNSS Extension to 12 Satellites	SMBV-K91	1415.8577.02
GNSS Extension to 24 Satellites	SMBV-K96	1415.8790.02



## 5 Rohde & Schwarz

Rohde & Schwarz is an independent group of companies specializing in electronics. It is a leading supplier of solutions in the fields of test and measurement, broadcasting, radiomonitoring and radiolocation, as well as secure communications. Established more than 80 years ago, Rohde & Schwarz has a global presence and a dedicated service network in over 70 countries. Company headquarters are in Munich, Germany.

### Sustainable product design

- Environmental compatibility and eco-footprint
- Energy efficiency and low emissions
- Longevity and optimized total cost of ownership



### Regional contact

- Europe, Africa, Middle East  
Phone +49 89 4129 12345  
[customersupport@rohde-schwarz.com](mailto:customersupport@rohde-schwarz.com)
- North America  
Phone 1-888-TEST-RSA (1-888-837-8772)  
[customer.support@rsa.rohde-schwarz.com](mailto:customer.support@rsa.rohde-schwarz.com)
- Latin America  
Phone +1-410-910-7988  
[customersupport.la@rohde-schwarz.com](mailto:customersupport.la@rohde-schwarz.com)
- Asia/Pacific  
Phone +65 65 13 04 88  
[customersupport.asia@rohde-schwarz.com](mailto:customersupport.asia@rohde-schwarz.com)
- China  
Phone +86-800-810-8228 / +86-400-650-5896  
[customersupport.china@rohde-schwarz.com](mailto:customersupport.china@rohde-schwarz.com)

### Headquarters

Rohde & Schwarz GmbH & Co. KG

Mühlendorfstraße 15 | D - 81671 München

+ 49 89 4129 - 0 | Fax + 49 89 4129 - 13777

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

This application note and the supplied programs may only be used subject to the conditions of use set forth in the download area of the Rohde & Schwarz website.

R&S® is a registered trademark of Rohde & Schwarz GmbH & Co. KG. Trade names are trademarks of the owners.