
RTA-OSEK

ORTI ユーザーズガイド

著作権について

© 2001 - 2007 LiveDevices Ltd. All rights reserved.

Version : RTA-OSEK V5.0.2

本書のいかなる部分も、あらかじめ LiveDevices Ltd. の書面による許可を得ないで複製することは禁じられています。本書に記載されているソフトウェアは、ライセンスに基づいて供給されるもので、使用およびコピーはライセンスの条項に基づく場合のみ認められます。

免責事項

本書の内容は、予告なく変更されることがあり、LiveDevices のいかなる部門からのお約束を表明するものでもありません。本書の情報は正確であると想定されておりますが、LiveDevices はいかなる誤りや記載漏れに対する責任も負わないものとします。

いかなる場合も、LiveDevices とその社員、受託業者あるいは本書の著者は、あらゆる性質または種類の損失利益、手数料、または出費に対する、特別、直接的、あるいは間接的な損害、損失、犠牲、負債、料金、要求、請求についての責任も負わないものとします。

商標

RTA-OSEK、Realogy および LiveDevices は LiveDevices Ltd. の商標です。

OSEK/VDX は Siemens AG の商標です。

Windows および MS-DOS は Microsoft Corp. の商標です。

他のすべての製品名も、各社の商標または登録商標です。

目次

1	本書について	5
1.1	本書の対象ユーザー	5
1.2	表記上の規約	5
2	はじめに	6
2.1	互換性	6
3	ORTI と RTA-OSEK の概要	7
3.1	開発手順	7
3.2	ORTI がアプリケーションに与える影響	8
3.3	出力値の信頼性	9
3.4	ブレイク位置による信頼性の違い	9
4	ORTI オブジェクトとその属性	10
4.1	概要	10
4.2	オブジェクトと属性	11
4.2.1	OS	11
4.2.2	タスク	11
4.2.3	カテゴリ 2 の ISR	12
4.2.4	カテゴリ 1 の ISR	12
4.2.5	カウンタ	13
4.2.6	アラーム	13
4.2.7	メッセージコンテナ	13
4.2.8	COM	14
5	お問い合わせ先	15

1 本書について

本書では、ORTI 対応デバッガと RTA-OSEK コンポーネント間のインターフェースについて説明します。

1.1 本書の対象ユーザー

本書では、RTA-OSEK コンポーネントを使用するプログラムを C で作成したことがあるソフトウェア開発者を対象としています。プログラムの挙動についての情報を得るために ORTI 対応デバッガを使用します。他の RTA マニュアルの内容を理解していることも前提となります。

1.2 表記上の規約

重要：このように表記されている注記には、ユーザーが知っておく必要のある重要な情報が記載されています。内容をよく読み、記載されているすべての指示に必ず従ってください。

移植性：このように表記されている注記では、RTA-OSEK コンポーネントが実行されるプロセッサ上で実行できるコードを作成する場合に知っておく必要がある事柄について説明しています。

本書では、プログラムコード、ヘッダファイル名、データ型名、C 関数、および RTA-OSEK コンポーネントの API 関数名はすべてクーリエ体 (courier) で表記されています。またプログラマに公開されているオブジェクトの名前も、やはりクーリエ体で表記されます。たとえば、Task1 という名前のタスクのタスクハンドルは、`Task1` と表記されます。

GUI エレメントとのインタラクションについての記述では、エレメントのキャプションは**ボールド体 (bold)** で表記されています。また、メニューなどの階層的なナビゲーションは矢印でレベルを区切り、たとえば、「メニューコマンド **Edit → Select All** を選択します。」、または「メニューから **Edit → Select All** を選択します。」のように表記されています。

また PDF 文書において、索引、および他の部分を参照する箇所（例：「第 3 章を参照してください」の部分）については、その参照先へのリンクが設けられているので、必要な参照箇所を素早く見つけることができます。

2 はじめに

ORTI は「OSEK Run Time Interface」の略語で、OSEK オペレーティングシステムの内部処理とデバッグの間のインターフェースを円滑化するために設計されたものです。デバッグが、実行中のオペレーティングシステム内でオブジェクトと変数を見つけてそれらの値を表示できるように、必要な情報が簡易な言語で提供されます。

ORTI 言語は汎用的な設計になっているので、OSEK 以外のオペレーティングシステムにも利用できます。ORTI のサポート機能は RTA-OSEK コンポーネントに組み込まれています。このため、RTA でビルドされたアプリケーションで使用される主要なオペレーティングシステム変数の値を、アプリケーション実行中に監視できます。

本書には、デバッグ用の ORTI 情報を生成する方法や、RTA-OSEK コンポーネントがアプリケーションの情報を ORTI 情報として出力する内容について記述されています。実行時における ORTI 情報の表示方法についての詳細は、ご使用のデバッグのマニュアルを参照してください。

2.1 互換性

ORTI は長期にわたり何度も更新されているため、それに対応するデバッグの種類も多種多様ですが、RTA-OSEK はこれらのデバッグを幅広くサポートしています。今後も新しいデバッグやアップデートされた既存のデバッグに順次対応していく方針ですが、もし現在ご使用中の ORTI 対応デバッグがサポートされていない場合は、ETAS のサポート窓口までお問い合わせください。

3 ORTI と RTA-OSEK の概要

3.1 開発手順

以下に、ORTI をユーザープログラムに組み込む方法を、順を追って説明します。

1. RTA-OSEK の GUI を使用して、アプリケーションにデバッガサポートを組み込む旨を指定します。これを行うには、ナビゲーションバーから **Target** グループを選択し、さらに **Debugger** サブグループを選択します。図 3-1 のように操作してください。

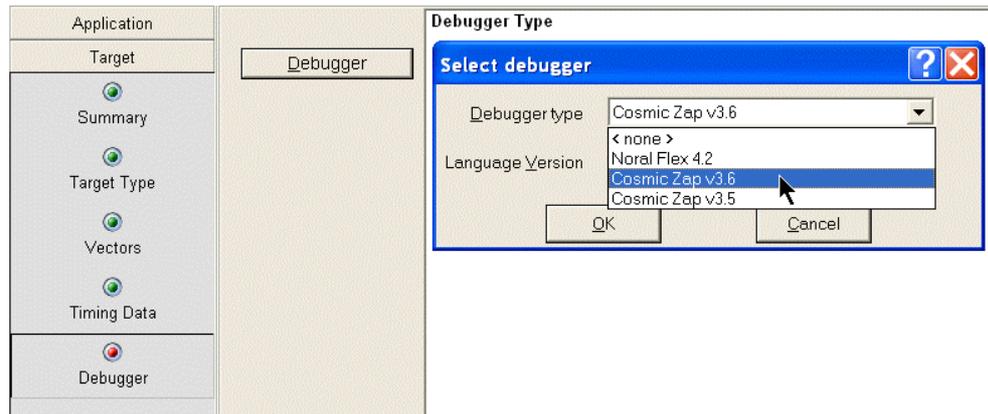


図 3-1 デバッガタイプを選択する

2. ワークスペースにある **Debugger** ボタンをクリックすると、**Select Debugger** ダイアログが表示されます。ドロップダウンリストを使用して **Debugger Type** (デバッガタイプ) と ORTI の **Language Version** (言語バージョン) を選択してから **OK** ボタンをクリックします。この例を図 3-2 に示します。

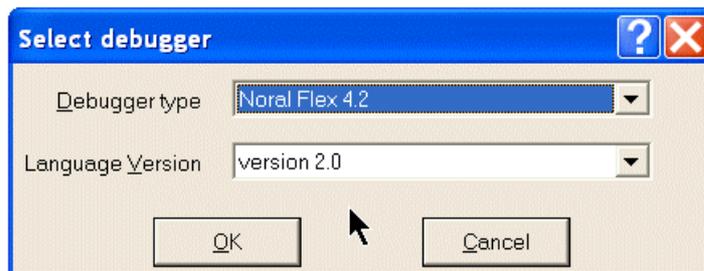


図 3-2 デバッガを選択する

3. アプリケーションのビルドを行います。<projectname>.ort というファイルに ORTI コードが挿入されます。図 3-3 に示すように、詳細な情報がワークスペースに表示されます。

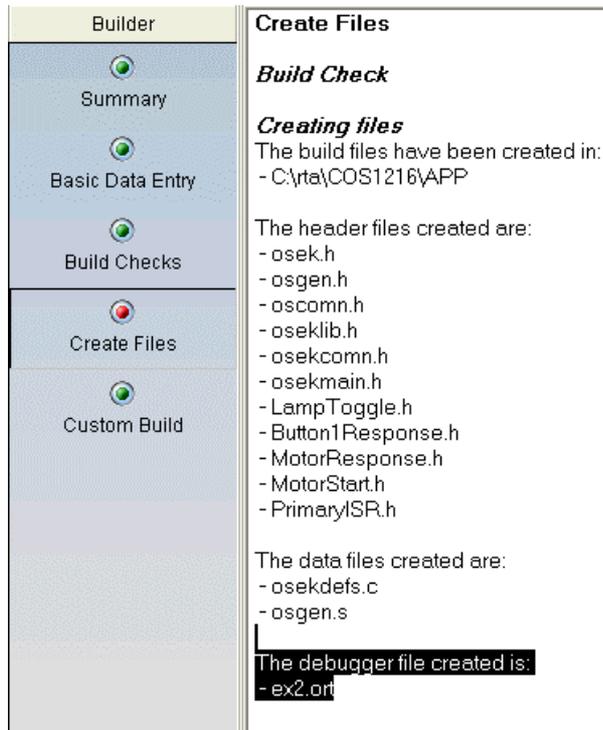


図 3-3 RTA-OSEK の GUI を使用してアプリケーションをビルドする

4. デバッガを起動し、アプリケーションをロードして ORTI ファイルをロードします。この手順については、デバッガのマニュアルに記載されています。
デバッガは ORTI ファイルに格納されている情報を表示します。この表示形式はデバッガにより異なります。

3.2 ORTI がアプリケーションに与える影響

ORTI は、実行中のアプリケーションのメモリから読み取った値を利用するので、ORTI が存在することによってアプリケーションの処理が影響を受ける可能性があります。ORTI は以下の 4 種類のデータについて、それぞれ異なる方法で値を取得します。

1. ORTI ファイル内の定数値
アプリケーションの実行中に変化しない値です。実行中のアプリケーションにはまったく影響を与えません。
2. アプリケーションの通常の処理の中で生成される値
ORTI が組み込まれていなくても、もともと存在している変数から読み取られる値です。これらの読み取りも、アプリケーションに余分な影響を与えることはありません。
3. Extended ビルドによって生成される、ORTI 専用の値
このような変数の読み取りは、アプリケーションに非常にわずかなオーバーヘッドをもたらします。
4. すべてのビルドモードで生成される、ORTI 専用の定数
このような値の読み取りは、アプリケーションにわずかなオーバーヘッドをもたらします。デバッガサポートを使用するように設定されていると、デバッガに情報を伝えるために必要な場合に限りこれらの定数が生成されます。最終的なリリースにおいては、デバッガサポートを無効にしておくことをお奨めします。

3.3 出力値の信頼性

ORTIにより報告される値の多くは、アプリケーションのメモリに格納されている値がそのまま読み取られたものです。そのため、システムが完全に初期化される前に ORTI を使用してシステムを検査すると、初期化前の不正な値が示されることとなります。RTA-OSEK の初期化が完了するのは、StartOS() 呼び出しによって最初のタスク（アイドルタスクを含みます）またはカテゴリ 2 の ISR の処理が開始された時点です。

また、変数の寿命内において、その変数の値がレジスタ内にキャッシュされている時間の割合が長い場合、特にレジスタの多いプロセッサでは注意が必要です。なぜなら ORTI は、所定のメモリロケーションに保存されている変数の値しか読み取ることができないためです。つまり、レジスタを使用した処理においてレジスタにコピーされた値が更新された場合、ORTI によって読み取られるメモリの値は、古いものになってしまう可能性があります。

3.4 ブ레이크位置による信頼性の違い

プログラムが以下のような位置に設定されたブ레이크ポイントで停止した場合、ORTI は正しい値を出力します。

- OSEK API 関数の外側で実行される、タスク、およびカテゴリ 2 の ISR のコード内

以下のようなブ레이크ポイントで停止した場合は、ORTI は無効な値を出力する可能性があります。

- OSEK API 関数内
- カテゴリ 1 の割込みハンドラにより実行されるコード内

ORTI が使用する OSEK データは、その一部が更新された状態になっている可能性があり、その場合は無効な値が出力されてしまいますが、通常、デバッガはプロセッサが停止した関数名を表示するので、プログラムが OSEK 呼び出し内で停止したかどうかを判断することができます。

ただし、3 つ以上の割込み優先度レベルが使用されている場合は、OSEK 呼び出しの途中でカテゴリ 1 割込みが発生してしまう可能性があります。このため、プログラムがカテゴリ 1 割込みハンドラ内のブ레이크ポイントで停止した場合は、デバッガのスタクトレース機能を使用して、割り込まれた関数の名前を明らかにする必要があります。ORTI の出力値は、OSEK API 関数内でカテゴリ 1 割込みが発生していない場合に限り、有効です。

4 ORTI オブジェクトとその属性

4.1 概要

本章では、RTA-OSEK に定義されている ORTI オブジェクトとそれらの属性について説明します。ORTI オブジェクトは、RTA-OSEK 内の情報をカプセル化したもので、OS、TASK、ALARM などのタイプがあります。1つのアプリケーション内には、これらの各オブジェクトのインスタンスが1つも含まれない場合もあれば、それぞれユニークな名前を持つ複数のインスタンスが含まれる場合もあります。各オブジェクトは複数の属性を持ち、各属性は1つの値をとります。たとえば、OS オブジェクトには RUNNINGTASK という属性があり、この属性は現在実行中のタスクを示します。

以下に、各オブジェクトとそのすべての属性について、一覧表で示します。表は、以下のような形式となっています。

オブジェクト：タイプ	オブジェクトの説明	
属性	説明	ビルドステータス
属性名， “ORTI ファイル内での記述形式”	属性の説明	All = すべて S = Standard (標準) T = Timing (タイミング) E = Extended (拡張)

各オブジェクトには、それぞれ異なる名前の付いた複数の属性が定義されていて、表の各行に、各属性の名前とそれについての簡単な説明が記載されています。

属性の列には、ORTI ファイルに記述される属性の名称が引用符で囲んで示されています。デバッガの種類によって、属性名を表示するものと ORTI ファイル内の属性名称を表示するものがあります。vs_ というプレフィックスは、その属性が RTA-OSEK をサポートするために追加されたものであり、標準的な ORTI 属性ではないことを示しています。

説明の列には、その属性が意味する内容が説明されています。

ステータスの列は、その属性がどのビルドステータス (Standard、Timing、Extended、または3つすべて) で生成されるかを示しています。

4.2 オブジェクトと属性

4.2.1 OS

オブジェクト: OS		OS オブジェクトは常に1つだけ存在し、"RTKOS" という名前を持ちます。	
属性	説明	ステータス	
RUNNINGTASK, "Running task"	現在実行中の TASK (タスク) の名前です。ISR がタスクに割り込んだ場合、その ISR の実行中においてもこのタスク名は変わりません。	All	
RUNNINGTASKPRIORITY, "Running task priority"	実行中のタスクの優先度を、OIL ファイルに定義されている名前で表わします。RUNNINGTASKPRIORITY は、タスクと ISR の共有リソースのロック状態を示すものではありません。	All	
RUNNINGISR2, "Running cat 2 ISR"	実行中のカテゴリ 2 ISR がある場合はその名前です。カテゴリ 2 ISR が実行中でない場合は NO_ISR と表示されます。	T,E	
vs_INTERRUPTPRIORITY, "Interrupt priority"	現在の割り込み優先度レベルです (ユーザーレベルで実行している場合は 0 です)。この属性は、一部のターゲットでのみサポートされています。	ターゲットごとに異なる	
SERVICETRACE, "OS Services Watch"	サービスルーチン (RTA-OSEK コンポーネントの API 関数) の入口または出口、およびルーチン名を示します。この属性を特別な totrace 属性として認識し、付加的な診断サポートを提供できるデバッガもあります。それ以外のデバッガでは、どの API 関数が最後に開始 (または終了) したかが表示されます。	E	
LASTERROR, "Last OSEK error"	最後に発生したエラーの名前を示します。初期値は E_OK です。	E	
CURRENTAPPMODE, "Current AppMode"	OIL ファイルに定義されている名前を用いた、現在のアプリケーションモードです。OIL ファイル内に定義されていない値になっていると、"unknown AppMode" という値が表示されます。	All	
vs_SPEED, "Processor speed"	OIL ファイルに定義されているプロセッサの処理速度です。	All	
vs_STOPWATCHSPEED, "Stopwatch speed"	OIL ファイルに定義されているストップウォッチ速度です。	All	

4.2.2 タスク

オブジェクト: TASK		コンフィギュレーションファイル内のタスク宣言に対応して生成されません。	
属性	説明	ステータス	
STATE, "State"	タスクステートです。値は SUSPENDED、RUNNING、READY、WAITING のいずれかです。	All	
vs_BASEPRIORITY, "Base priority"	タスクのベース優先度を示します。ベース優先度とは、OIL ファイルに定義されているタスクの優先度です。	All	
vs_DISPATCHPRIORITY, "Dispatch priority"	タスクのディスパッチ優先度を示します。ディスパッチ優先度は、タスクの実行開始の優先度です。内部リソースが使用される場合や、ノンプリエンプタブルタスクの場合は、ディスパッチ優先度をベース優先度よりも高くすることができます。	All	

オブジェクト : TASK	コンフィギュレーションファイル内のタスク宣言に対応して生成されます。	
属性	説明	ステータス
vs_ACTIVATIONS, "Maximum activations"	タスクを重複して起動できる最大回数です。	All
vs_TYPE, "Conformance type"	タスクのコンフォーマンスクラスを示します。取りうる値は BCC1、BCC2、ECC1、ECC2 です。	All
vs_RESOURCES, "Resources"	タスクが使用するリソースのリストです。	All
vs_EVENTS, "Events"	タスクが待つことのできるイベントのリストです。	All
vs_TERMINATION, "Termination"	タスクのターミネションタイプです。HEAVY または LIGHT です。	All
vs_SCHEDULE, "Preemptability"	このタスクがプリエンパブルか、またはノンプリエンパブルであるか示します。	All
vs_USESFP, "Floating point"	このタスクが浮動小数点演算を使用するかどうかを示します。浮動小数点演算を使用する場合は TRUE、使用しない場合は FALSE です。	All

4.2.3 カテゴリ 2 の ISR

オブジェクト : ISR2	コンフィギュレーションファイル内のカテゴリ 2 の ISR 宣言に対応して生成されます。	
属性	説明	ステータス
vs_PRIORITY, "Priority"	ISR の優先度を示します。	All
vs_VECTOR, "Vector"	ISR が関連付けられているベクタを示します。	All
vs_USESFP, "Floating point"	この ISR が浮動小数点演算を使用するかどうかを示します。浮動小数点演算を使用する場合は TRUE です。	All
vs_RESOURCES, "Resources"	ISR が使用するリソースのリストです。	All
vs_BUFFERING, "Buffering"	ISR のバッファリングタイプを示します。これは、RTA-OSEK の GUI で表示されるバッファリングと一致します。	All

4.2.4 カテゴリ 1 の ISR

オブジェクト : ISR1	コンフィギュレーションファイル内のカテゴリ 1 の ISR 宣言に対応して生成されます。	
属性	説明	ステータス
vs_PRIORITY, "Priority"	ISR の優先度を示します。	All
vs_VECTOR, "Vector"	ISR が関係付けられているベクタを示します。	All
vs_BUFFERING, "Buffering"	ISR のバッファリングタイプを示します。これは、RTA-OSEK の GUI で表示されるバッファリングと一致します。	All

4.2.5 カウンタ

オブジェクト : COUNTER	コンフィギュレーションファイル内のカウンタ宣言に対応して生成され ます。	
属性	説明	ステータス
vs_COUNT, "Count"	現在のカウンタ値を示します。	All
vs_TICKRATE, "Tick rate"	予想チックレートを示します。	All
vs_DRIVER, "Driver"	カウンタをドライブする (カウンタの値を更新する) ISR またはタスクを示します。	All
vs_MAXALLOWEDVALUE, "MaxAllowedValue"	カウンタの最大値を示します。	All
vs_MINCYCLE, "MinCycle"	カウンタの最小サイクル値を示します。	All
vs_TICKSPERBASE, "TicksPerBase"	カウンタの ticksperbase 値を示します。	All

4.2.6 アラーム

オブジェクト : ALARM	コンフィギュレーションファイル内のアラーム宣言に対応して生成され ます。	
属性	説明	ステータス
ALARMTIME, "Alarm Time"	次にアラームカウンタが満了する時間を示します。現 在のカウンタ値を得るには、COUNTER オブジェクト 内の Count の値を参照してください。	All
CYCLETIME, "Cycle Time"	サイクリックアラームの周期を示します。シングル ショットアラームの場合、この値は0になります。	All
ACTION, "Action"	アラームカウンタ満了時に実行するアクションです。 以下のようなアクションを指定できます。 <ul style="list-style-type: none"> タスクの起動 イベントの設定 コールバック関数の実行 RTA-OSEK コンポーネントの場合、複数のアクション を実行することができます。そのためにはアクション をリストとして記述します。	All
STATE, "Alarm state"	アラームカウンタが行われているかどうかを示しま す。RUNNING または STOPPED という値を取ります。	All
"COUNTER", "Counter"	このアラームが割り当てられているカウンタの名前を 示します。	All

4.2.7 メッセージコンテナ

オブジェクト : MESSAGECONTAINER	コンフィギュレーションファイル内のメッセージ受信アクセサ宣言に対応 して生成されます。	
属性	説明	ステータス
MSGNAME, "Message Name"	メッセージの名前	All
vs_CDATATYPE, "C type"	メッセージの C データ型を示します。	All
MSGTYPE, "Message Type"	メッセージタイプを示します。QUEUED または UNQUEUED という値を取ります。	All

オブジェクト : MESSAGECONTAINER	コンフィギュレーションファイル内のメッセージ受信アクセサ宣言に対応して生成されます。	
属性	説明	ステータス
"QUEUE SIZE", "Queue size"	キューイングされるメッセージ用のメッセージキューのサイズを示します。キューイングされないメッセージの場合、この値は0です。	All
ACTION, "Action performed when message is received"	メッセージ受信時に実行されるアクションを示します。以下の各アクションを、それぞれ最大1つずつ含めることができます。 <ul style="list-style-type: none"> • タスクの起動 • イベントのセット • 関数の呼び出し • フラグのセット 	All
SENDER, "Sender"	メッセージを送信するタスクの名前です。	All
vs_SENDCOPY, "Send copy"	送信アクセサがメッセージデータのコピーを使用するかどうかを示します。WithCopyの場合はTRUE、WithoutCopyの場合はFALSEにしてください。	All
RECEIVER, "Receiver"	メッセージを受信するタスクの名前です。	All
vs_RECEIVECOPY, "Receive copy"	受信アクセサがメッセージデータのコピーを使用するかどうかを示します。WithCopyの場合はTRUE、WithoutCopyの場合はFALSEにしてください。	All

4.2.8 COM

オブジェクト : COM	COM オブジェクトは常に1つだけ存在し、"RTACOM" という名前を持ちます。	
属性	説明	ステータス
vs_RUNNING, "Running"	COM が稼働しているかどうかを示します。稼働している場合はTRUE、そうでなければFALSEです。	E

5 お問い合わせ先

製品サポートに関しては、各 ETAS 支社までお問い合わせください。

ヨーロッパ (フランス、ベルギー、ルクセンブルグ、イギリスを除く)

ETAS GmbH

Borsigstrasse 14	Phone:	+49 711 8 96 61-0
70469 Stuttgart	Fax:	+49 711 8 96 61-105
Germany	E-mail:	sales@etas.de
	WWW:	WWW.ETAS.COM

フランス、ベルギー、ルクセンブルグ

ETAS S.A.S

1, place des Etats-Unis	Phone:	+33 1 56 70 00 50
SILIC 310	Fax:	+33 1 56 70 00 51
94588 Rungis Cedex	E-mail:	sales@etas.fr
France	WWW:	www.etas.com

イギリス

ETAS Ltd.

Studio 3, Waterside Court	Phone:	+44 1283 - 54 65 12
Third Avenue, Centrum 100	Fax:	+44 1283 - 54 87 67
Burton-upon-Trent	E-mail:	sales@etas-uk.net
Staffordshire DE14 2WQ	WWW:	www.etas.com
UK		

USA

ETAS Inc.

3021 Miller Road	Phone:	+1 (888) ETAS INC
Ann Arbor, MI 48103	Fax:	+1 (734) 997-9449
USA	E-mail:	sales@etas.us
	WWW:	www.etas.com

日本

イータス株式会社

〒 220-6217	Phone:	(045) 222-0900
神奈川県横浜市西区	Fax:	(045) 222-0956
みなとみらい 2-3-5	E-mail:	sales@etas.co.jp
クイーンズタワー C 17F	WWW:	www.etas.com

韓国

ETAS Korea Co., Ltd.

4F, 705 Bldg. 70-5	Phone:	+82 2 57 47-016
Yangjae-dong, Seocho-gu	Fax:	+82 2 57 47-120
Seoul 137-889	E-mail:	sales@etas.co.kr
Korea	WWW:	www.etas.com

中国

ETAS (Shanghai) Co., Ltd.

2404, Bank of China Tower	Phone:	+86 21 5037 2220
200 Yincheng Road Central	Fax:	+86 21 5037 2221
Shanghai 200120	E-mail	sales.cn@etasgroup.com
P.R. China	WWW:	www.etas.com

インド

ETAS Automotive India Pvt. Ltd.

No. 690, Gold Hill Square, 12F	Phone:	+91 80 4191 2585
Hosur Road, Bommanahalli	Fax:	+91 80 4191 2586
Bangalore, 560 068	E-mail	sales.in@etasgroup.com
India	WWW:	www.etas.com