

### RX600シリーズ CAN-UARTブリッジ

R01AN0721JU0210  
Rev.2.10  
2013.09.18

#### 要旨

このアプリケーションノートとともにダウンロードできる MCU のファームウェアは、非同期シリアルデバイス (UART) と CAN の間のインタフェースが RX MCU によっていかに簡単に実現できるかを示しています。

RX600 シリーズのような CAN を備えたルネサスの MCU は、非同期シリアルポート (UART) 周辺回路も内蔵しています。このアプリケーションノートと付属しているソースコードでは、RX MCU を PC の RS232 と CAN ネットワークの間の両方向のブリッジとして利用する例を示します。

PC 端末から入力されたデータは RS232 シリアル接続で RX MCU に送られます。これらのデータは RX MCU の CAN-UART ブリッジソフトウェアで解析され、CAN データパケットにエンコードされた上で、RX MCU から CAN バス上に送り出されます。逆に、CAN バスからのコマンドやデータパケットは RX MCU で受信され、デコードされた後に、RS232 シリアルリンクを使用して PC 端末に渡されます。

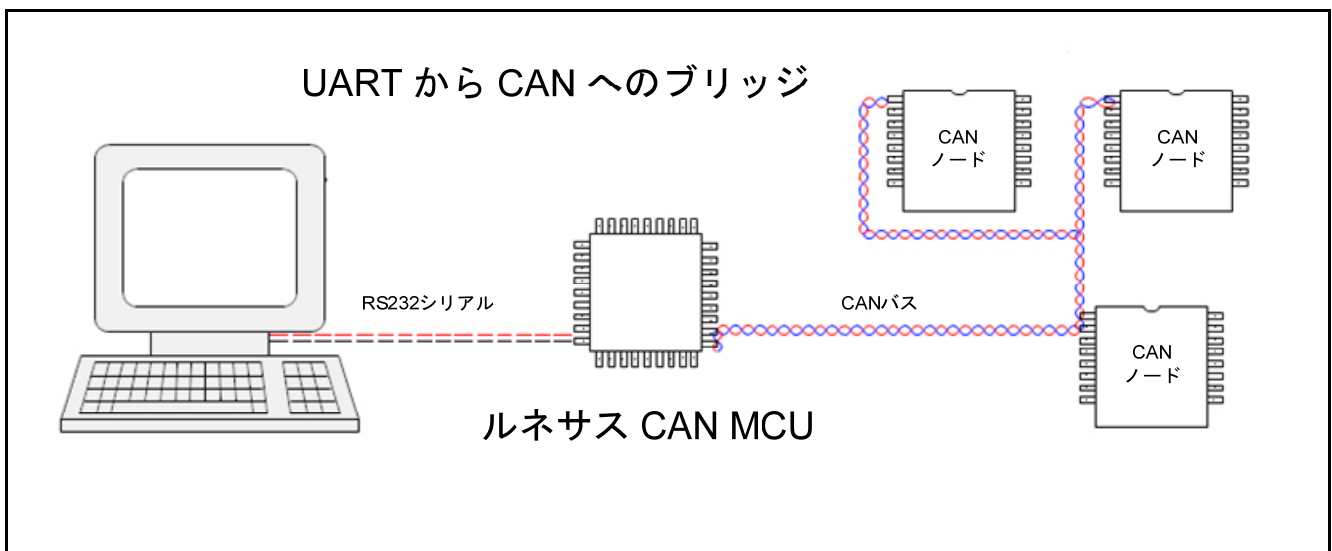


図1 UART / CAN ブリッジ

PC と CAN バス間の CAN-UART ゲートウェイとしての RX MCU。CAN バスとの間で両方向のデータ受け渡しが可能で、これを PC 端末上に表示することができます。

#### 対象デバイス

CAN 機能をもつ RX600 シリーズの MCU。アプリケーションノートのソースコードは多種の RX RSK に対応しています。

本アプリケーションノートを他のマイコンへ適用する場合、そのマイコンの仕様にあわせて変更し、十分評価してください。

#### 関連ドキュメント

CAN ドライバの使用については、CAN API の仕様とアプリケーションノート R01AN0339EU を参照してください。

また、ご使用になるボードの回路図をご覧ください。現状では次のボードに対応しています。

RSK-RX63N、RSK-RX630、RSK-RX62N、RSK-RX62T および RSK-RX62G。

## 目次

1. CAN-UART ゲートウェイ .....	3
1.1 CAN バスからのデータのバッファリング .....	4
2. サポートされているルネサス開発ボード .....	5
3. CAN-UART ブリッジを実行するために必要なもの .....	5
4. プロジェクトの構築 .....	6
4.1 ツール .....	6
4.2 CAN API のコンフィグレーション .....	6
5. 操作 .....	7
5.1 ボードの接続 .....	7
5.2 シリアルポートの設定 .....	7
5.3 シリアルターミナル上のコマンド構文 .....	7
5.4 ボード上の LED 表示 .....	8
6. System CAN バスアナライザの利用 .....	9

## 1. CAN-UART ゲートウェイ

このアプリケーションノートは、CAN 機能を備えたルネサス RX600 シリーズ MCU をシリアル UART から CAN バスへのブリッジとして使用する場合について説明します。デモプログラムが UART 周辺回路と CAN モジュールをもつ RX MCU が 2 つのインタフェース間の通信のゲートウェイとして働く場合の有効な実例として提供されています。

CAN-UART ブリッジアプリケーションは CAN バスを監視し CAN バス上のすべてのデータをシリアルポート (UART/RS232) に送出する「CAN スニファ」として動作します。このシリアルデータは PC 上でハイパーターミナルのような一般的なターミナルプログラムを使用して表示させることができます。また PC ターミナルプログラムからブリッジを経由して CAN バスにメッセージを送ることもできます。

```

COM1:115200baud - Tera Term VT
File Edit Setup Control Window Help

Renesas Electronics Inc.

RX63N CAN-UART Bridge. Send and receive data to/from the CAN bus.
UART set to BAUDRATE, defined in uart.c. 8-bit data, no parity, 1 stop bit.

Press 'z' character to stop this test.
0123456789;:<=>?@ABCDEFGHIJKLMNopQRSTUVWXYZ[\]^_`abcdefghijklmnopqrstuvwxyz 0123456789;:<
>?@ABCDEFGHIJKLMNopQRSTUVWXYZ[\]^_`abcdefghijklmnopqrstuvwxyz 0123456789;:<=>?@ABCDEFGHI
LMNOPQRSTUVWXYZ[\]^_`abcdefghijklmnopqrstuvwxyz
'z' character received. Stopped test.

CAN-UART Bridge active.

Send UART data to CAN with syntax <iXXXdB1B2B3B4B5B6B7B8z>, where XXX are ID-nibbles (hex),
B1, B2,... are databytes (hex) and z means 'Send!'. For example; to send a CAN frame with ID
7A and data=010203A1FF, write <i7A0d010203A1FFz> to serial terminal.NOTE: CAPs must be used
for hex digits A-F.

-----
Data from CAN format:  ID=XXX Data=B1B2B3...
-----
ID=501 Data=00540000
ID=001 Data=103100
ID=005 Data=03FFFFFFF00
ID=080 Data=0000005108
ID=005 Data=03FFFFFFF00
ID=001 Data=103100
ID=501 Data=00540000
ID=005 Data=03FFFFFFF00
ID=7F0 Data=FF00
ID=001 Data=103100
ID=7A0 Data=800000000000400
ID=005 Data=03FFFFFFF00
ID=001 Data=103100
ID=080 Data=0000005108
ID=005 Data=03FFFFFFF00
ID=501 Data=00540000
ID=001 Data=103100
ID=005 Data=03FFFFFFF00
ID=001 Data=103100
ID=005 Data=03FFFFFFF00
ID=7F0 Data=FF00
ID=501 Data=00540000
ID=7A0 Data=800000000000400
ID=080 Data=0000005108
ID=001 Data=103100
ID=005 Data=03FFFFFFF00
ID=005 Data=03FFFFFFF00
ID=001 Data=103100
ID=501 Data=00540000

```

図1.1 PC 端末のスクリーンショット

ここでは、RSK につながっている CAN バスからの CAN データ が CAN-UART ブリッジによって送り込まれます。ユーザは iXXXdB1B2B3B4B5B6B7B8z の形式で端末に入力し、データを CAN バスに送り出すことができます。XXX は 11 ビットの CAN ID で、B1B2...はデータバイトを示しています。XXX と B1B2...は 16 進数の数字 0-F です。A-F は大文字で入力する必要があります。

## 1.1 CANバスからのデータのバッファリング

通常、CANバスはUARTと比べてはるかに高速のボーレートで動作しています。実例のコードではCANバスが500kボーで動作しているのに対しUARTは115.2kボーで動作しています。また、CANのデータ送信は連続ではなくバースト状に発生する傾向があります。このためCANの送信データがこれを転送するUARTの容量を超え、結果としてデータフレームが失われる可能性があります。CANバスとUARTの間のボーレートの速度差を補うために、受信されたCANデータは自動的に蓄えられます。CANデータフレームをRAMに蓄えることで、MCUからデータを送り出すUARTの時間が確保されます。バッファがオーバーフローしたときにはメッセージがシリアルデータの流れの中に現れますので、受信されるCANデータのためにバッファサイズを大きくします。

## 2. サポートされているルネサス開発ボード

表2.1

ルネサス MCU	開発ボード	開発ボードの注文番号
RX62N	RSK-62N	R0K5562N0S000BE
RX62N	YRDK62N	YRDKRX62N
RX62T	RSK-62T	R0K5562T0S000BE
RX630	RSK-630	R0K505630C011BR
RX63N	RSK-63N	R0K50563NC00BR

このデモンストレーションは表 2.1にあるルネサス RX 開発ボードのいずれでも実行することができます。

r\_can\_api.h、r\_can\_api.c、および config\_r\_can\_rapi.h で構成されているルネサス CAN API(ダウンロードパッケージ R01AN0339 にある application programming interface)では CAN ハードウェアを操作するためのドライバやプロトコルレベルのルーチンがすべて提供されていますので、これを利用することで CAN アプリケーションの開発が単純化されます。R\_CAN\_API の利用方法の詳細は、ダウンロードパッケージ中の CAN API アプリケーションノートを参照してください。config\_r\_can\_rapi.h のみ変更し、r\_can\_api.c は変更しない点に注意してください。いつでも最新のルネサス CAN API ドライバパッケージ( R01AN0339 )をダウンロードし、上記の 3 個のファイルと交換することができます。

## 3. CAN-UARTブリッジを実行するために必要なもの

このアプリケーションノートとソースコードのほか、HEW、RX ツールチェイン(コンパイラとリンカ)および E1 もしくは E20 に対応するデバッガをルネサスのウェブサイトからダウンロードしてください。

以下は、用意すべきものの一覧表です。

- ルネサス統合開発環境(HEW: High-performance Embedded Workshop)プログラミング環境、ツールチェイン、およびデバッガ
- CAN\_UART\_Bridge\_RX6xx プロジェクト HEW ワークスペースパッケージ
- CAN API アプリケーションノート: R01AN0339EU0110
- RX6xx MCU 開発ボード(表 2.1)
- E1 もしくは E20 エミュレータ
- DB9 コネクタを有する RS232 ケーブル
- Systec CAN バスアナライザ GW-200、32040xx、もしくは同等品(オプション)
- PC ターミナルアプリケーション(HyperTerminal、TeraTerm など)

## 4. プロジェクトの構築

### 4.1 ツール

ご使用になるルネサス RX600 シリーズ開発ボードと E1 もしくは E20 エミュレータの接続については、各開発ボードのインストレーションガイドを参照してください。

CAN-UARTブリッジを実行可能な状態に構築するには、CAN\_UART\_Bridge\_RX6xx.hws ファイルをダブルクリックして、対象となるボードの HEW ワークスペースプロジェクトを開きます。

また、HEW 開発ツールソフトウェアを理解するためには、HEW のユーザマニュアルの PDF ファイルを参照してください。

### 4.2 CAN API のコンフィグレーション

表 4.1は CAN-UARTブリッジプロジェクトのワークスペースに一般に含まれているファイルの一覧表です。

表4.1

ソースファイル	ヘッダファイル
can_uart_bridge.c	config_r_can_rapi.h
r_can_api.c	can_uart_bridge.h
dbstc.c	r_can_api.h
hwsetup.c	iodef.h
intprg.c	lcd.h
lcd.c	rskRX6xxdef.h
resetprg.c	sbrk.h
sbrk.c	stacksct.h
uart.c	typedef.h
vecttbl.c	uart.h
	vect.h

can\_uart\_bridge.c ソースファイルにはアプリケーションのメインルーチンが含まれています。

config\_r\_can\_rapi.h は CAN ドライバファイル r\_can\_api.h および r\_can\_api.c の内容をユーザが変更する必要がないよう設計されています。ユーザ独自のハードウェアで CAN API を利用する際には、config\_r\_can\_rapi.h 内で設定を変更する必要があるかもしれません。たとえば、他のボードへのポーティングまたは CAN ボーレート設定の変更の場合に、CAN 割り込み優先度や使用するトランシーバー端子を変更します。デフォルトでは CAN バスのデータ速度は 500 kbit/s に設定されています。



CAN-UARTブリッジを実行するには、デモアプリケーションが CAN 割り込みを使用するように設定しなければなりません。

CAN API の使用方法の詳細に関してはルネサス CAN API アプリケーションノート (R01AN0339EU) をご覧ください。

## 5. 操作

CAN-UARTブリッジは、ターゲットMCUを経由してUARTにCANデータを送り、たとえば、PCのターミナルプログラム上に表示します。また、PC端末からCANバス上にデータを送ることも可能です。CAN-UARTブリッジを実行するには、次の手順に従ってください。

### 5.1 ボードの接続

1. E1もしくはE20デバッガ/エミュレータをRSKボードに接続し、アプリケーションをダウンロードします。RSKをお持ちであれば、HEWプロジェクトを開き、コードをダウンロードもしくはデバッグする方法がクイックスタートガイドに示されています。
2. ユーザPCのRS232COMポートからCAN-UARTブリッジを実行しているMCUにUARTケーブルを接続します。ご使用のCAN-UARTブリッジボードにDSUB-9コネクタが搭載されていないときには、ボードの回路図を参照してUARTのRXとTXピンを見つけ、直接接続してください。
3. CAN-UARTブリッジボードのCANバスをCANネットワークに接続します。CANバスでは、両端にそれぞれ120オームの抵抗が必要であることに注意してください。ツイストペアの2線間の全体の抵抗値は60オームになります。(ノードには抵抗が含まれていないのが理想ですが、これはオフィスのキュービクルやテストベンチよりも接続距離の長い、規模の大きなネットワークに適用されるため抵抗があります。)

### 5.2 シリアルポートの設定

1. ファームウェアにおけるUARTのデフォルト設定は8データビット、1ストップビット、パリティなし、フローコントロールなし、です。
2. 実際の速度を知るためにBAUDRATEを確認してください。これはuart.cファイルで定義されており、幾つかのモデルのボードでは起動時にLCD上に表示されます。
3. ご使用のターミナルアプリケーションのシリアル通信ポートの設定をCAN-UARTブリッジプログラムの設定と一致させます。

### 5.3 シリアルターミナル上のコマンド構文

シリアル接続が正しく行われると、PC端末上には文字の列が流れるように表示されます。このループを止めるには、PCのキーボードからzを入力し、CAN-UARTブリッジモードに入ります。

ブリッジを経由してCANバスにデータを送る際には次の構文を使用します。

iXXXdB1B2B3B4B5B6B7B8z

- ここで:
- iは次に16進数の3桁の数字で標準CANIDが入力されることを示します(文字iを上記の位置に入力しなければなりません)。
  - dは次に0ないし8バイトの16進数で示されるペイロードデータが入力されることを示します(文字dを上記の位置に入力しなければなりません)。
  - zは入力されたデータを直ちに送信することを指示します。

注: 16進数の値A-Fを入力する際には、必ず大文字を使用してください。

たとえば、CANID 0x07A0とデータバイト 0x09 0x0A 0x0B 0xCCをもつフレームを送るには、UARTターミナルでi7A0d090A0BCCzのようにCANメッセージを入力します。

## 5.4 ボード上の LED 表示

プログラムは動作中に幾つかの LED を点灯し、状態を表示します。RSK ボードに搭載された LED は次のように状態を示しますが、実際の LED との対応はボードの種類によって異なっていることがあります。

- LED0 (多くのボードでは緑): UART がデータ受信中
- LED1 (多くのボードでは橙): UART がデータ送信中
- LED2 (多くのボードでは赤): CAN がデータフレーム受信中
- LED3 (多くのボードでは赤): CAN がデータフレーム送信中
- 全 LED 点灯: CAN はエラー状態

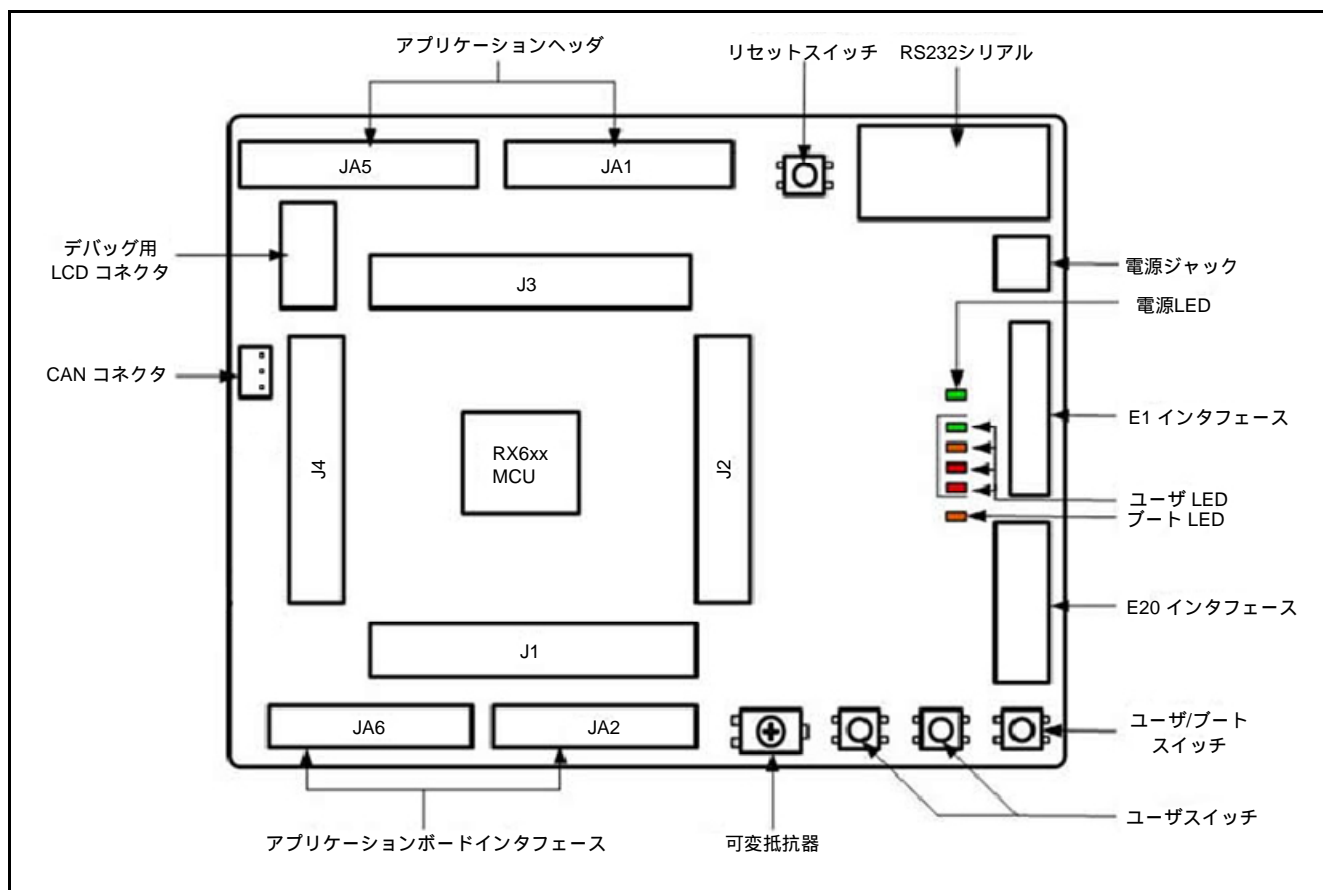


図5.1 RX 600 シリーズ RSK 開発ボードの典型的な部品配置 (右側に LED インジケータを配置)



## 6. Systec CAN バスアナライザの利用

CAN-UARTブリッジアプリケーションを手軽に評価するためには、Systec CANmodulのようなCANバスアナライザを使用します。これはルネサスCAN開発キットとともに提供されているSystecの「CANスニファ」ですが、CAN接続によるフラッシュメモリの書き換えのアプリケーションノートとファームウェア(R01AN0235EU\_RX)で使用されています。

Systec CANmodulは個別にドイツのSystec-Electronics社から購入することもできます。スニファはPCのUSBホストおよびCANバスネットワークに接続されます。CAN側では、SystecはCANバスのノードとして動作します。PC側では、Systecモジュールを経由してGUIモニタソフトウェアがデータを受信し、またCANバスにデータフレームを送出することもできます。

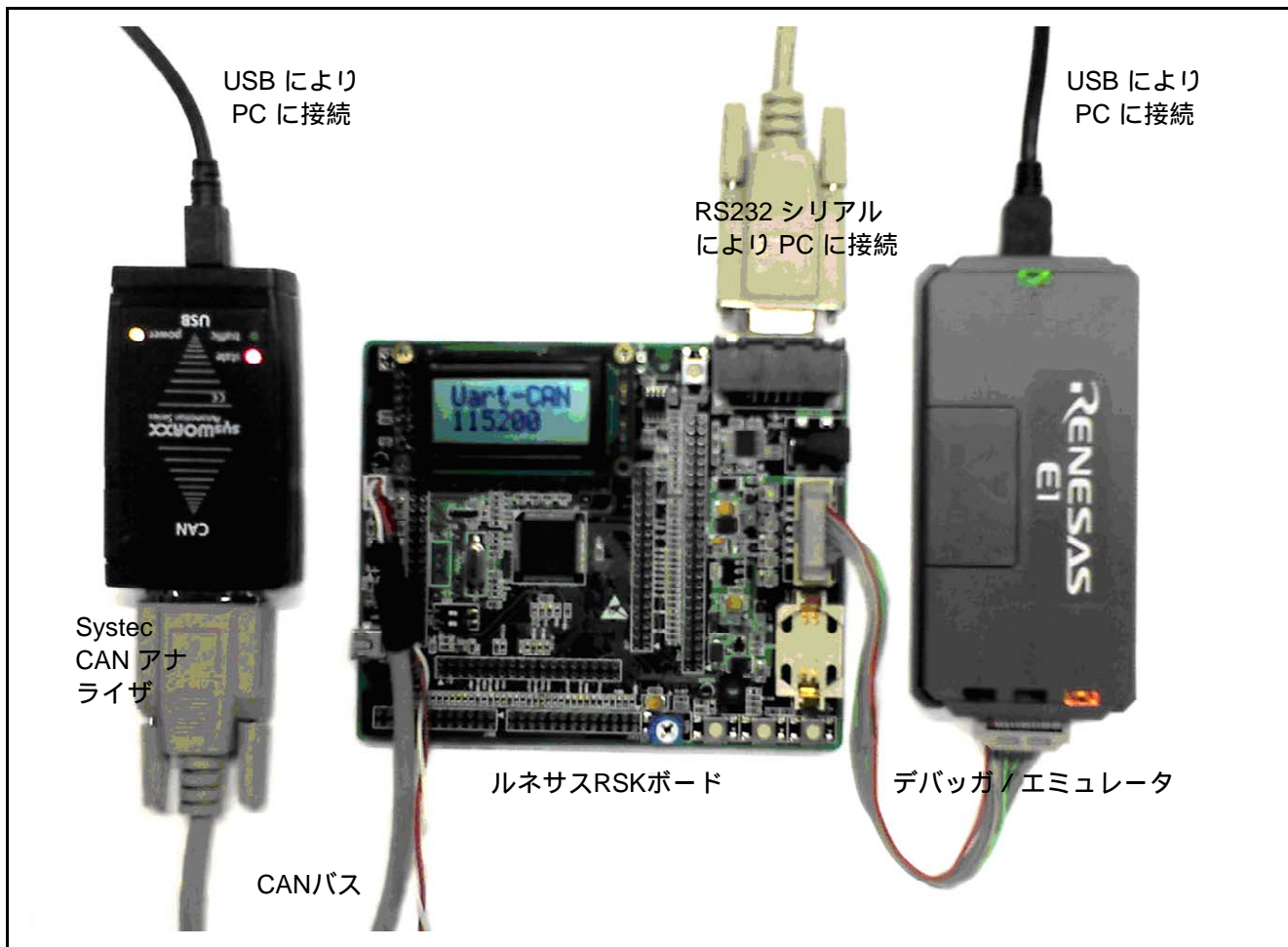


図6.1 CAN UARTブリッジのテストのためのRSK RX630の構成

ホームページとサポート窓口

ルネサス エレクトロニクスホームページ

<http://japan.renesas.com>

お問い合わせ先

<http://japan.renesas.com/contact/>

改訂記録	RX600 シリーズ アプリケーションノート CAN-UART ブリッジ
------	--------------------------------------

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2010.01.30	—	初版発行
2.00	2012.01.15	1	PC 端末のスクリーンショットの図の追加。 63N ワークスペースの追加。
2.10	2013.09.18	1	62G ワークスペースの追加。

すべての商標および登録商標は、それぞれの所有者に帰属します。

## 製品ご使用上の注意事項

ここでは、マイコン製品全体に適用する「使用上の注意事項」について説明します。個別の使用上の注意事項については、本文を参照してください。なお、本マニュアルの本文と異なる記載がある場合は、本文の記載が優先するものとします。

### 1. 未使用端子の処理

【注意】未使用端子は、本文の「未使用端子の処理」に従って処理してください。

CMOS製品の入力端子のインピーダンスは、一般に、ハイインピーダンスとなっています。未使用端子を開放状態で動作させると、誘導現象により、LSI周辺のノイズが印加され、LSI内部で貫通電流が流れたり、入力信号と認識されて誤動作を起こす恐れがあります。未使用端子は、本文「未使用端子の処理」で説明する指示に従い処理してください。

### 2. 電源投入時の処置

【注意】電源投入時は、製品の状態は不定です。

電源投入時には、LSIの内部回路の状態は不確定であり、レジスタの設定や各端子の状態は不定です。外部リセット端子でリセットする製品の場合、電源投入からリセットが有効になるまでの期間、端子の状態は保証できません。

同様に、内蔵パワーオンリセット機能を使用してリセットする製品の場合、電源投入からリセットのかかる一定電圧に達するまでの期間、端子の状態は保証できません。

### 3. リザーブアドレスのアクセス禁止

【注意】リザーブアドレスのアクセスを禁止します。

アドレス領域には、将来の機能拡張用に割り付けられているリザーブアドレスがあります。これらのアドレスをアクセスしたときの動作については、保証できませんので、アクセスしないようにしてください。

### 4. クロックについて

【注意】リセット時は、クロックが安定した後、リセットを解除してください。

プログラム実行中のクロック切り替え時は、切り替え先クロックが安定した後に切り替えてください。リセット時、外部発振子（または外部発振回路）を用いたクロックで動作を開始するシステムでは、クロックが十分安定した後、リセットを解除してください。また、プログラムの途中で外部発振子（または外部発振回路）を用いたクロックに切り替える場合は、切り替え先のクロックが十分安定してから切り替えてください。

### 5. 製品間の相違について

【注意】型名の異なる製品に変更する場合は、事前に問題ないことをご確認下さい。

同じグループのマイコンでも型名が違っていると、内部メモリ、レイアウトパターンの相違などにより、特性が異なる場合があります。型名の異なる製品に変更する場合は、製品型名ごとにシステム評価試験を実施してください。

## ご注意書き

1. 本資料に記載された回路、ソフトウェアおよびこれらに関連する情報は、半導体製品の動作例、応用例を説明するものです。お客様の機器・システムの設計において、回路、ソフトウェアおよびこれらに関連する情報を使用する場合には、お客様の責任において行ってください。これらの使用に起因して、お客様または第三者に生じた損害に関し、当社は、一切その責任を負いません。
2. 本資料に記載されている情報は、正確を期すため慎重に作成したのですが、誤りがないことを保証するものではありません。万一、本資料に記載されている情報の誤りに起因する損害がお客様に生じた場合においても、当社は、一切その責任を負いません。
3. 本資料に記載された製品データ、図、表、プログラム、アルゴリズム、応用回路例等の情報の使用に起因して発生した第三者の特許権、著作権その他の知的財産権に対する侵害に関し、当社は、何らの責任を負うものではありません。当社は、本資料に基づき当社または第三者の特許権、著作権その他の知的財産権を何ら許諾するものではありません。
4. 当社製品を改造、改変、複製等しないでください。かかる改造、改変、複製等により生じた損害に関し、当社は、一切その責任を負いません。
5. 当社は、当社製品の品質水準を「標準水準」および「高品質水準」に分類しており、各品質水準は、以下に示す用途に製品が使用されることを意図しております。  
標準水準： コンピュータ、OA機器、通信機器、計測機器、AV機器、  
家電、工作機械、パーソナル機器、産業用ロボット等  
高品質水準： 輸送機器（自動車、電車、船舶等）、交通用信号機器、  
防災・防犯装置、各種安全装置等  
当社製品は、直接生命・身体に危害を及ぼす可能性のある機器・システム（生命維持装置、人体に埋め込み使用するもの等）、もしくは多大な物的損害を発生させるおそれのある機器・システム（原子力制御システム、軍事機器等）に使用されることを意図しておらず、使用することはできません。たとえ、意図しない用途に当社製品を使用したことによりお客様または第三者に損害が生じても、当社は一切その責任を負いません。なお、ご不明点がある場合は、当社営業にお問い合わせください。
6. 当社製品をご使用の際は、当社が指定する最大定格、動作電源電圧範囲、放熱特性、実装条件その他の保証範囲内でご使用ください。当社保証範囲を超えて当社製品をご使用された場合の故障および事故につきましては、当社は、一切その責任を負いません。
7. 当社は、当社製品の品質および信頼性の向上に努めていますが、半導体製品はある確率で故障が発生したり、使用条件によっては誤動作したりする場合があります。また、当社製品は耐放射線設計については行っておりません。当社製品の故障または誤動作が生じた場合も、人身事故、火災事故、社会的損害等を生じさせないよう、お客様の責任において、冗長設計、延焼対策設計、誤動作防止設計等の安全設計およびエージング処理等、お客様の機器・システムとしての出荷保証を行ってください。特に、マイコンソフトウェアは、単独での検証は困難なため、お客様の機器・システムとしての安全検証をお客様の責任で行ってください。
8. 当社製品の環境適合性等の詳細につきましては、製品個別に必ず当社営業窓口までお問合せください。ご使用に際しては、特定の物質の含有・使用を規制するRoHS指令等、適用される環境関連法令を十分調査のうえ、かかる法令に適合するようご使用ください。お客様がかかる法令を遵守しないことにより生じた損害に関して、当社は、一切その責任を負いません。
9. 本資料に記載されている当社製品および技術を国内外の法令および規則により製造・使用・販売を禁止されている機器・システムに使用することはできません。また、当社製品および技術を大量破壊兵器の開発等の目的、軍事利用の目的その他軍事用途に使用しないでください。当社製品または技術を輸出する場合は、「外国為替及び外国貿易法」その他輸出関連法令を遵守し、かかる法令の定めるところにより必要な手続を行ってください。
10. お客様の転売等により、本ご注意書き記載の諸条件に抵触して当社製品が使用され、その使用から損害が生じた場合、当社は何らの責任も負わず、お客様にご負担して頂きますのでご了承ください。
11. 本資料の全部または一部を当社の文書による事前の承諾を得ることなく転載または複製することを禁じます。

注1. 本資料において使用されている「当社」とは、ルネサス エレクトロニクス株式会社およびルネサス エレクトロニクス株式会社とその総株主の議決権の過半数を直接または間接に保有する会社をいいます。

注2. 本資料において使用されている「当社製品」とは、注1において定義された当社の開発、製造製品をいいます。



ルネサスエレクトロニクス株式会社

■営業お問合せ窓口

<http://www.renesas.com>

\*営業お問合せ窓口の住所・電話番号は変更になることがあります。最新情報につきましては、弊社ホームページをご覧ください。

ルネサス エレクトロニクス販売株式会社 〒100-0004 千代田区大手町2-6-2（日本ビル）

(03)5201-5307

■技術的なお問合せおよび資料のご請求は下記へどうぞ。  
総合お問合せ窓口：<http://japan.renesas.com/contact/>