

ADK-622x3
ユーザー・ガイド

ADK-62203 BC/RT/MT 64K RAM

ADK-62213 BC/RT/MT 4K RAM



株式会社ナセル

REVISION HISTORY [履歴]

リビジョン	日付	変更内容
AN-62203 Rev. New	2019年5月21日	初版リリース
Rev. A	2019年11月1日	4K デバイスとプロセッサ・オプション追加
Rev. B	2021年2月23日	「Holt モードのプロジェクト設定」のスクリーンショットを更新。 「プロジェクト・ファイル・リストと選択された説明」で使用されるコンソール関数を更新。
Rev. C	2021年10月25日	Holt プロジェクト・ライブラリ (API) のスクリーンショット、コマンドの説明、スコーププロットを更新

Introduction [はじめに]

Holt の HI-62203/62213 評価ボードは、2つのアプリケーション開発キット（ADK-62203、および、ADK-62213）をサポートし、Holt の MIL-STD-1553 HI-622x3 ファミリの幅広い機能セットを実証しています。

- ADK-62203 は、Holt 社の HI-62203 RT（リモート・ターミナル）、BC（バス・コントローラ）、MT（モニタ）デバイスを使用し、64K Word RAM を搭載
- ADK-62213 は、Holt 社の HI-62213 RT、BC、MT デバイスを使用し、4K Word RAM を搭載

H-622x3 ファミリは、プロトコル管理と物理バス・インターフェイス回路を含む MIL-STD-1553B バス通信デバイスのセットである。2ボード・アセンブリと C プロジェクト・リファレンス・デザインは、BC、MT、RT の動作を実証する、すぐに実行できる評価プラットフォームを提供します。このキットには、IAR Systems Embedded Workbench® for ARM、および、ARM Cortex M3 マイクロ・コントローラ用の完全統合デバッグ・インターフェイスが含まれ、利便性を高めています。本書は、アプリケーション開発キット ADK-62203 と ADK-62213 の両方をサポートしています。

このガイドでは、ボードのセットアップと実行方法について説明します。その他のサポート資料と必要なプロジェクト・ソフトウェアは、付属の Holt USB ドライブに収録されています。デモ・ソフトウェアのバージョンは、すでにマイコン・フラッシュにプログラムされており、付属のソフトウェア開発ツールをインストール、または、実行しなくても、ボードは箱から出してすぐに動作可能です。

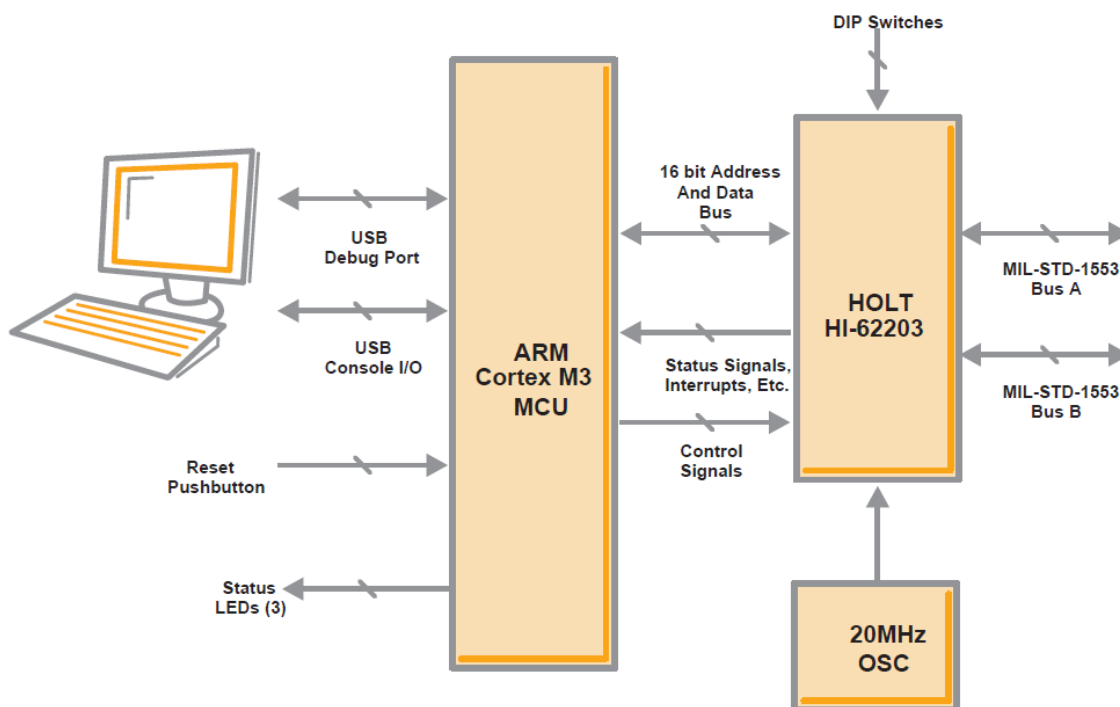


図 1 HI-62203 評価ボード、ARM Cortex MCU ボード上に設置

Evaluation Kit Contents [評価キットの内容]

- 本ユーザー・ガイド
- USB ドライブの HOLT HI-622x3TM プロジェクト・ソフトウェアとドキュメント
- IAR Systems Embedded Workbench® for ARM のインストーラー (USB ドライブに付属のファイル「Holt HI-622x3 API Demo Project Installation Guide」参照)
- 2×USB インターフェイス・ケーブル
- 2 枚のボードで構成
- HI-62203TM デバイスとデュアル・トランス結合 MIL-STD-1553 バス・インターフェイスを備えた上部 DUT ボード
- 多数の DIP スイッチでボード動作を設定
- ARM Cortex M3 16/32 Bit マイクロ・プロセッサ、デバッグ・インターフェイス、3.3VDC 安定化電源を搭載した下部 MCU ボード

Hardware Block Diagram [ハードウェア・ブロック図]



Default Switch Settings (HI-62203 board)

CONFIG 2 (SW1)

スイッチ	デフォルト	説明
SW1、1	OFF	nRTB: ON、nRTBOOT ピン = 0、1760 モード in 4K device OFF、A12/nRTBOOT = 1 (open)
SW1、2	OFF	LOG1: ON、A13/Logic 「1」 pin = 0 OFF、A13/Logic 「1」 pin open
SW1、3	OFF	CS_S0: ON、CLK_SEL_0 ピン = 0 OFF、CLK_SEL_0 ピン = 1 (open)
SW1、4	OFF	CS_S1: ON、CLK_SEL_1 ピン = 0 OFF、CLK_SEL_1 ピン = 1 (open)
SW1、5	ON	BC_D: ON、BC 無効化されない OFF、BC 無効化

RT ADDRESS (SW2)

スイッチ	デフォルト	説明
SW2、6-2	00011 (ON = 0)	RT アドレスを設定します。デフォルトは 03 に設定
SW2、1	OFF	OFF = RT アドレスパリティビット「1」、奇数パリティでなければデバイスは動作しません

CONFIG 1 (SW3)

スイッチ	デフォルト	説明
SW3、1	OFF	TRG_SEL: 未使用
SW3、2	ON	TXINHA: OFF、BUSA での送信を禁止
SW3、3	ON	TXINHB: OFF、BUSB での送信を禁止
SW3、4	OFF	nSGLE_END: OFF、XCVR は差動データを使用 ON、XCVR はシングルエンド・データを使用
SW3、5	OFF	nSSFLAG/EXT_TRIG: OFF、1553 SSFLAG bit が設定されていない ON、SSFLAG bit が設定されている Note: 外部トリガを使用する場合、SW3、5 を OFF にする必要があります
SW3、6	OFF	RSTBITEN: OFF、リセット時に内部セルフテストが有効 ON、内部セルフテスト無効
SW3、7	ON	TRAN/nBUFF: OFF、トランスペアレント・モード ON、バッファモード
SW3、8	OFF	UPADDREN: OFF、内部レジスタ CLK_SEL のビットを使用 ON、CLK_SEL[1:0] ピンを使用
SW3、9	OFF	POL_SEL: OFF、RD/nWR を使用 ON、逆極性の nRD/WR を使用
SW3、10	OFF	MSCLR: ON、ハードウェア・リセット

Default Jumper Settings [デフォルト・ジャンパ設定]

HI-62203 Board [HI-62203 ボード]

ジャンパー

ジャンパー	位置	説明
JP1	OFF	ARM 基板へのクロック送信用リンク（通常は使用しません）
JP2	OFF	BUSA マイナスラインを接地
JP3	OFF	BUSB マイナスラインを接地
JP4	ON	ロジックと XCVR を別々に供給する場合はオープンにしてください（TP9 &10）
JP5	OFF	BUSA の 70Ω 負荷抵抗に接続
JP6	OFF	BUSB の 70Ω 負荷抵抗に接続
J3	OFF	XTAL 発信器を無効にし、外部クロックを J4 に接続することができません
J7	ON	これらは、トランシーバーのデジタル信号を伝送しており、外部トランシーバーを使用する場合のみ切断してください

Test Points [テスト・ポイント]

テスト・ポイント	説明
TP1	HI-62203 入力クロック
TP2	nSSFLAG 出力、または、外部トリガ入力
TP3	Positive 接続 1553 Bus A
TP4	Negative 接続 1553 Bus A
TP5	nINCMD、「0」は HI-62203 アクティビティを示す（デフォルト） nMCRST、モード・コード 8 リセット出力（有効な場合）
TP6	Positive 接続 1553 Bus B
TP7	Negative 接続 1553 Bus B
TP8	TAG クロックの入力
TP9/TP10	HI-62203、または、HI-62213 用 3.3V 電源供給 （ARM ボードから供給される）
TP11/12	グラウンド接続

ARM Board [ARM ボード]

ジャンパー

ジャンパー	位置	説明
JP1	OFF	モード・コード 8 でボードをリセットするためのリンク
JP2	ON、または、OPEN	NonZero Wait タイプのインターフェイスを使用するためのリンク
JP3	OFF	Zero Wait タイプのインターフェイスを使用するためのリンク
JP4	OFF	使用しません
J1	OFF	外部 ARM クロックのリンク

J6	OFF	USB 5V からの電源供給を可能にするためのリンク。ベンチ電源を使用する場合は、これが切断されていることを確認すること。
----	-----	---

LEDs

LED #	DESCRIPTION
LED1	ソフトウェア定義 LED
LED2	ソフトウェア定義 LED
LED3	ソフトウェア定義 LED

Hardware Design Overview [ハードウェア設計概要]

DUT 基板は着脱式で、付属の MCU 基板から分離して、ユーザー提供の代替マイクロ・プロセッサや FPGA 基板に接続することが可能です。基板間ヘッダーは、一般的なプロトタイピング・ボードと互換性を持たせるために、0.1 インチ (2.54 mm) グリッドに配置されています。すべてのホスト・インターフェイス信号は、ボード間ヘッダーを経由します。RT アドレス設定ピンを含むいくつかの設定ピンは、上部 DUT ボード上の 2 つのディップ・スイッチで制御されており、これらの信号はボード間ヘッダーでは使用できません。

下部の ARM Cortex M3 ボードは、フラッシュ・プログラミング可能な Atmel AT91SAM3U-EK マイクロ・プロセッサをベースにしています。ARM からの 16 Bit データ/アドレス・バスは、DUT に接続されます。USB シリアル・ポートがコンソール I/O を提供します。RESET プッシュ・ボタンは、ARM マイクロ・プロセッサをリセットし、DUT マスター・リセット信号を制御します。

ARM Cortex M3 ボードには、www.segger.com からライセンスを受けた「J-Link On Board」デバッグ・インターフェイスが含まれており、高価な JTAG デバッグ・ケーブルを購入することなく、すぐに使えるようになっています。このキットには、ボードのデバッグ・インターフェイスをコンピュータに接続するためのシンプルな USB ケーブルが含まれています。

HI-62203 Host Interface [HI-62203 ホスト・インターフェイス]

HI-62203 は 16 Bit のパラレル・データ・バスを持ち、64K×16 ワードの SRAM アドレス空間を持ちます (4K×16 Bit SRAM を持つバージョン HI-62213 もあります)。どちらも 312 ピン PBGA パッケージで提供されます。

HI-62203 のデータ転送速度は、4 種類のクロック周波数のうち、どの周波数を選択するかによって異なります。このボードには 20MHz の XTAL 発振器モジュールが付属しているため、デフォルトではソフトウェアが 50MHz の動作を設定することになります。しかし、SMA コネクタ J4 から外部クロックを入力することができますので、その場合はジャンパー J7 を接続する必要があります。デバイスは、20、16、12、10MHz のクロックで動作しますが、適切なレジスタ設定をレジスタ 0x18 に設定する必要があります。

Control Switch [コントロール・スイッチ]

SW2 には HI-62203 の動作に影響を与える 6 つの制御機能があります。これらは設定のセクションで説明されていますが、続ける前にデフォルトの位置にあることを確認してください。

RT address set up [RT アドレスの設定]

RT ターミナルのアドレスは、電源を入れる前にディップ・スイッチで設定します。6 ポジション・ディップ・スイッチ SW2 には、すでにアドレス値 03+奇数パリティが設定されているはずで

1553 Bus Interface [1553 バス・インターフェイス]

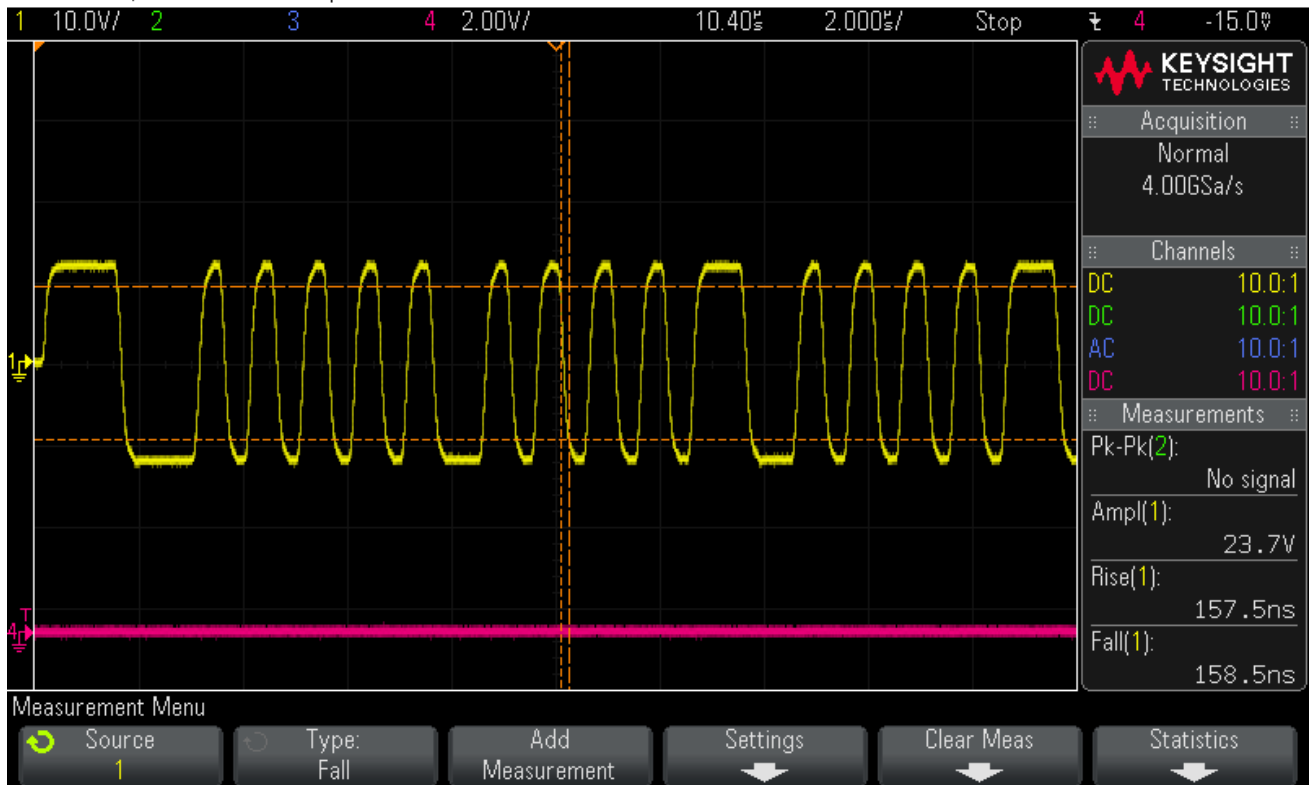
注 1 : Bus Negative をグランドに接続するのは、あくまでベンチ・テストの便宜上のものです。1553 信号の送受信のほとんどの性能特性は、バス・スタブでの差動ライン間測定 (Bus Positive - Bus Negative) を使用して指定されます。これは、上部回路基板の右側、トランスに隣接する赤と黒の「BUS」テスト・ポイントに対応します。赤と黒に接続した 2 本のオシロスコープのプロープは、スコープの CH1-CH2 演算機能と組み合わせて使うことができますが、Bus Positive に接続した 1 本のプロープで、Bus Negative が接地されたときに同じ信号を表示することができます。これにより、スコープのプロープを他の用途に使用することができます。nINCMD (TP5) 信号は、次ページからのプロットのマゼンタのトレースに示すように、スコープをトリガするために使用することができますが、この信号は 1553 の活動中にローになります。

警告 : Bus Negative を接地するための規定を製品設計に含めないでください。

注 2 : スタンドアロン・テスト (従来の MIL-STD-1553 バスに接続しない) のために、ハードウェアはオンボードの 70 Ω 終端抵抗を提供します。これは、外部 1553 バス接続なしで BC と RT のデモをサポートするベンチ・テストの便宜的な機能に過ぎません。RT/MT モードを使用する場合、RT はバス・モニタの有無にかかわらず、完全にメッセージを処理することができます。

MIL-STD-1553 で適切に終端されたバスに接続する場合、オンボードの終端抵抗は使用されません。終端抵抗は設計に含めないでください。

MSO-X 3054A, MY50510474: Thu Sep 20 10:10:15 2018



BusA 1553 出力と nINCMD 信号、BC モードでバス・コマンドを生成します。

MSO-X 3054A, MY50510474: Mon Sep 24 13:29:40 2018



BusA 1553 出力と nINCMD 信号 (RT モード、TxData コマンドに応答)。

Initial Setting Up [初期設定]

アプリケーション開発キット ADK-62203 は、QFP80 ピン・パッケージの Holt HI-62203 をサポートするために設計されています。HI-62203 は 64K の 16 Bit SRAM を持ち、RT、BC、SMT バス・モニタ (MT) の 3 つのモードすべてで動作します。同様に、アプリケーション開発キット ADK-62213 は、4K の 16 Bit SRAM を搭載した Holt 社の HI-62213 デバイスをサポートしています。本ドキュメントは、両キットのセットアップと使用方法をサポートします。

Windows 7、10 ...

Holt の USB から付属の teraterm-4.71.exe インストーラ・プログラムを実行し、無料のオープン・ソース・ターミナル・エミュレーション・プログラム、TeraTerm 4.71 をインストールしてください。著作権表示を残すことを条件に再配布を許可する旨の使用許諾書に同意してください。著作権表示は、TeraTerm のウィンドウで Help をクリックし、About TeraTerm をクリックすることで表示させることができます。インストールを続行する

デフォルトのインストール先を承諾し、「Next」をクリックします。

Select Components 画面で、Additional Plugin = TTXResizeMenu 以外の選択を解除し、「Next」をクリックします。

インストールする言語を選択し、「Next」をクリックします。

スタート・メニュー・フォルダがデフォルトで表示されるので、これを受け入れて、「Next」をクリックします。

必要なショートカットを選択し、「Next」をクリックします。

Ready to Install 画面で、「Install」をクリックします。

TeraTerm プログラムを実行します。New Connection 画面で、(x)Serial を選択し、選択した COM ポートを選択します。Setup、Serial Port の順にクリックすると、シリアル・ポートの設定画面が表示されます。以下の設定を行います。ボーレート：115200、データ：8 Bit、パリティ：なし、ストップ：1 Bit、フロー・コントロール：なし。

電源は DC5V のベンチ電源と 5V の USB 電源の両方が使用可能です。USB を使用する場合は、ARM ボード下部のジャンパー J6 を閉じてください。ただし、多くの PC では、USB が送信中のボードに供給するのに十分な電力を持っていないことに注意してください。ベンチ電源を使用する場合は、J6 がオープンであることを確認してください。TeraTerm が起動し、正しく設定されていれば、コンソール・ウィンドウに以下のようなコマンド・メニューが表示されるはずです。このメニューは、電源投入時や RESET ボタンが押された時に表示されます。TeraTerm が評価ボードと正しく通信していることを確認したら、「Setup」⇒「Save Setup」でターミナルの設定を保存することができます。

RT アドレスは、DIP スイッチ SW2 で設定します。RT アドレス 3 は、BC のメッセージ・レパトリで使われます。6 ポジションの DIP スイッチには、アドレス値 03 と奇数パリティが既に設定されているはずです。このボードには 20MHz の XTAL オシレーターが供給されており、ソフトウェアはデフォルトで 20MHz で動作するように設定されています。もし、異なるクロック周波数が必要な場合は、コネクタ J4 から外部クロックを入力します。その前にコネクタ J3 をジャンプして、オンボードのオシレーターをディセーブルにする必要があります。

Demonstration functions [デモ機能]

General Structure of Demo Functions [デモ関数の一般的な構造]

Holt API のデモ・プログラムは、src (ソース) フォルダ内のソース・ファイルから実行されます。main.c ファイルは console.c を呼び出し、demos.c と demos_supplemental.c のデモ関数を実行します。Holt API ランタイム・ライブラリは、実行可能なオブジェクト・コードとして HoltPortableLib というライブラリに含まれています。キー押下は console.c で検出します。HoltPortableLib のソース・ファイル・パッケージは、Holt の営業に問い合わせることで入手できます。

Main Menu and utilities [メイン・メニューとユーティリティー]

リセット、または、電源投入後、以下のメニューが表示されますが、表示される日時は以下に示す画面キャプチャと異なる場合があります。

```

COM5:115200baud - Tera Term VT
File Edit Setup Control Window Help

Non-Holt mode set
16 bit mode set
ARM Clock:      50 Mhz
ARM RAM Range:  4096 words

Holt device reset.
Initializing for BC mode.
Setting metrics enabled.
Setting Holt Device clock frequency: 20MHz.

*****
Holt Integrated Circuits HI-6220 Project: 62213
Demonstration Rev 2.0 Compiled: Oct 15 2021 09:55:58
Holt Portable Library version:  4.0.0
Device-core version:           1.8.5
Configured demonstrations:     BC RT MT RMT (+Supplemental)
*****

R: display the 62213 registers
D: dump memory in the 62213
P: toggle polling mode of demonstrations

A: run the BC asynchronous demo
H: send a high priority BC message
L: send all low priority BC message(s)
N: run the BC Major-Minor-Frame demo
X: stop BC transmissions
S: run the SMT demo
B: run the RT demo
K: enable RMT operation
T: toggle RT or MT traffic display

Supplemental BC demonstrations
0: BC1 - simple BC sends RX MC17 to RT1

Supplemental RT demonstrations
1: RT1 - simple RT SA1 separate RX/TX buffers
2: RT2 - simple RT SA 30 RX/TX using the same buffer
3: RT3 - simple RT SA 30 increments TX data
4: RT4 - simple RT responds to MC17 using ISQ and interrupts

W: to reset the 62213
M: to show menu

62213 ->

```

「w」を押して HI-620x3 をリセットし、「r」を押して HI-620x3 のレジスタを表示すると、以下のような表示になります。

```

COM5:115200baud - Tera Term VT
File Edit Setup Control Window Help
0: BC1 - simple BC sends RX MC17 to RT1

Supplemental RT demonstrations
1: RT1 - simple RT SA1 separate RX/TX buffers
2: RT2 - simple RT SA 30 RX/TX using the same buffer
3: RT3 - simple RT SA 30 increments TX data
4: RT4 - simple RT responds to MC17 using ISQ and interrupts

W: to reset the 62213
M: to show menu

62213 ->
Register Value Description
00 0000 Interrupt Enable Reg #1
01 0000 Configuration Reg #1
02 0000 Configuration Reg #2
03 0000 Start-Reset/BC-RT Cmd Stk Pointer Reg/Enh BC ILP
04 0000 BC Control Word/RT SubAddr Ctrl Word
05 3223 Time Tag Reg
06 0000 Interrupt Status Reg #1
07 0000 Configuration Reg #3
08 0000 Configuration Reg #4
09 0005 Configuration Reg #5
0A 0000 RT/MT Data Stk Address Reg
0B 0000 BC Frame Time Remaining Reg
0C 0000 BC Message Time Remaining Reg
0D 0000 Basic BC Frame Time/Enh BC ILP/RT Last Cmd/MT Trigger
0E 0000 RT Status Word Reg
0F 0000 RT BIT Word Reg
10 0000 Test Reg #0
11 0000 Test Reg #1
12 0000 Test Reg #2
13 0000 Test Reg #3
14 0000 Test Reg #4
15 0000 Test Reg #5
16 0000 Test Reg #6
17 0000 Test Reg #7
18 c002 Configuration Reg #6
19 0000 Configuration Reg #7
1B 0000 BC Condition Code/BC General Purpose Flag Reg
1C a800 BIT Test Status Reg
1D 0000 Interrupt Enable Reg #2
1E 0000 Interrupt Status Reg #2
1F 0000 BC GPQ Ptr/RT-MT Int Status Queue Ptr

62213 ->

```

Reg#5 のデータはタイムタグ・カウンタで、継続的に変化します。Reg#9 は RT アドレス・レジスタで、コンテンツ 0005 は RT アドレス 3 が設定されていることを表しています。Reg#1C はセルフテスト・レジスタで、a800 は電源投入時のプロトコル・テストが実行され、パスしたことを示します。RSBITEN ピンは、リセット時にこのテストを実行するために High でなければならぬことに注意してください。

「d」コマンドは、メモリをダンプし、デバイスの全アドレス範囲を検査することができます。このコマンドには、以下のような独自のコマンド・メニューがあります。

```

COM5:115200baud - Tera Term VT
File Edit Setup Control Window Help
Running Major Minor Frame Bus A...
62213 ->
Adr 0000: 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 FFFF 0014 0000 0000 0000 0000
Adr 0010: 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
Adr 0020: 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
Adr 0030: 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
Adr 0040: 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
Adr 0050: 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
Adr 0060: 1111 1101 1202 1303 1404 1505 1606 1707 1808 1909 1A0A 1B0B 1C0C 1D0D 1E0E 1F0F
Adr 0070: 10F0 10E0 10D0 10C0 10B0 10A0 1090 1080 1070 1060 1050 1040 1030 1020 1010 1001
Adr 0080: 3333 3101 3202 3303 3404 3505 3606 3707 3808 3909 3A0A 3B0B 3C0C 3D0D 3E0E 3F0F
Adr 0090: 30F0 30E0 30D0 30C0 30B0 30A0 3090 3080 3070 3060 3050 3040 3030 3020 3010 3003
Adr 00A0: 4444 4101 4202 4303 4404 4505 4606 4707 4808 4909 4A0A 4B0B 4C0C 4D0D 4E0E 4F0F
Adr 00B0: 40F0 40E0 40D0 40C0 40B0 40A0 4090 4080 4070 4060 4050 4040 4030 4020 4010 4004
Adr 00C0: 0080 1BC0 0060 02BC 09F7 9200 1001 0000 0080 1FC0 0080 02BC 09FB 9200 1FC0 0000
Adr 00D0: 0000 1FC0 00A0 02BC 0A27 B200 1FC0 0000 394F FFFF BD4F 0000 054F 00C0 054F 00C8
Adr 00E0: 054F 00D0 C94F FFF8 C94F 0001 994F 0008 A94F FF9B 0940 00E8 114F 0000 1D4F 0000
Adr 00F0: 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
=====
Keys: <D>own <U>p <R>efresh <A>ddress <M>enu 0x0000-0x00FF
=====
Adr 0100: 0000 0000 0000 0000 0000 0000 0000 0000 8D4F 00D8 8D4F 00D8 8D4F 00D8 8D4F 00D8
Adr 0110: 8D4F 00D8 C94F FFFF C94F 0014 0957 0108 1D4F 0000 1D4F 0000 0000 0000 0000 0000
Adr 0120: 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
Adr 0130: 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
Adr 0140: 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
Adr 0150: 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
Adr 0160: 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
Adr 0170: 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
Adr 0180: 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
Adr 0190: 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
Adr 01A0: 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
Adr 01B0: 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
Adr 01C0: 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
Adr 01D0: 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
Adr 01E0: 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
Adr 01F0: 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
=====
Keys: <D>own <U>p <R>efresh <A>ddress <M>enu 0x0100-0x01FF
=====

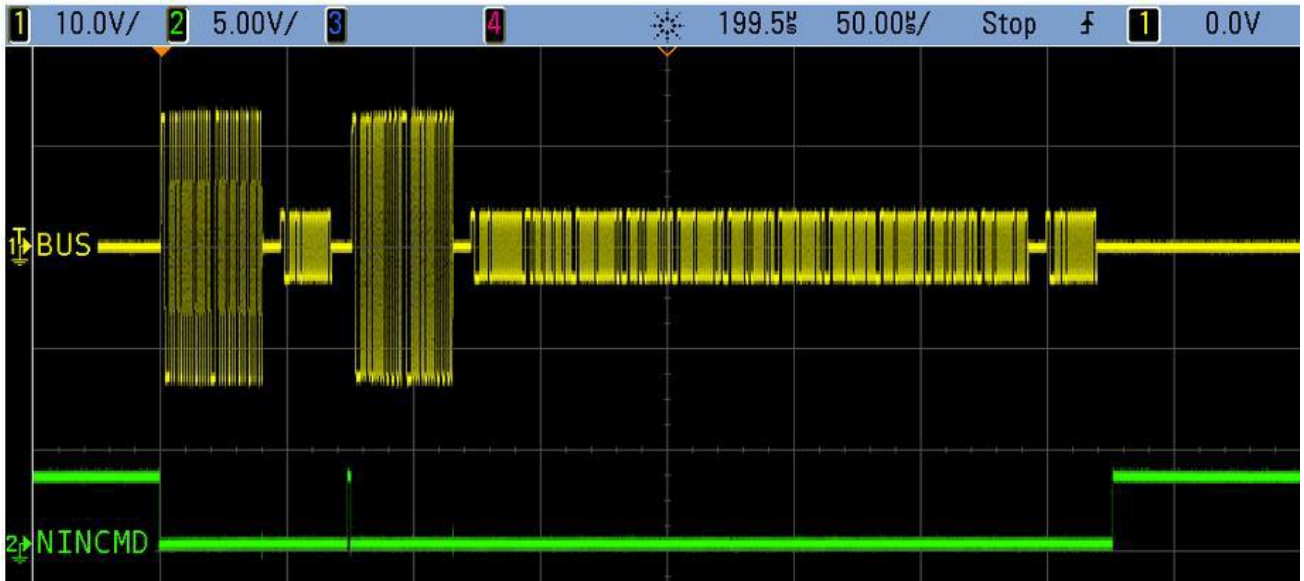
```

BC Mode overview [BC モード概要]

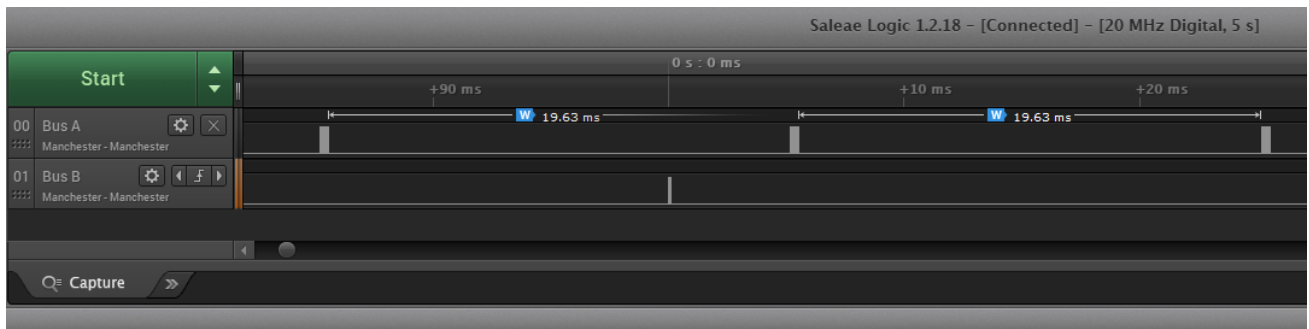
コマンド「a」は、BC が連続したフレームを送信するように設定し、高、および、低優先度の非同期メッセージを準備します。コマンド「a」の後にコマンド「h」を発行すると、時間が許す限りフレーム間で高優先度の非同期メッセージが送信されます。コマンド「a」の後にコマンド「l」を発行すると、時間が許す限り、フレーム間に低優先度の非同期メッセージが送信されます。「n」コマンドは、BUSA と BUSB の両方を使用して Major/Minor フレームを送信します。これらのデモでは、HoIt API を使用して BC 非同期メッセージ、Major/Minor フレーム、低優先度メッセージ、高優先度メッセージを生成する方法を、オシロスコープ、または、1553 パス・モニタで確認することができます。

BC Mode (using an external RT) [BC モード (外部 RT を使用)]

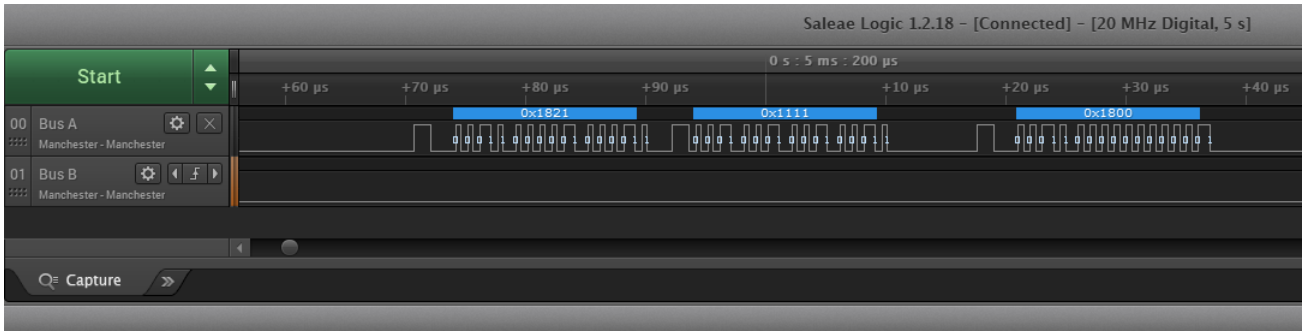
1. コンソール・コマンド「a」は、RT3 への BC コマンドと RT1 への BC コマンドの下のバス・トレースを示しています。両方の RT が応答していることは、低振幅のデータとステータス・ワードが表示されていることで明らかです。コマンド「a」は、BC が連続フレームを送信するように設定し、高、および、低優先度の非同期メッセージを準備します。同じ波形を見るには、2 つの RT をバスに接続し、1 つの RT をアドレス 3、もう 1 つの RT をアドレス 1 に接続する必要があります。



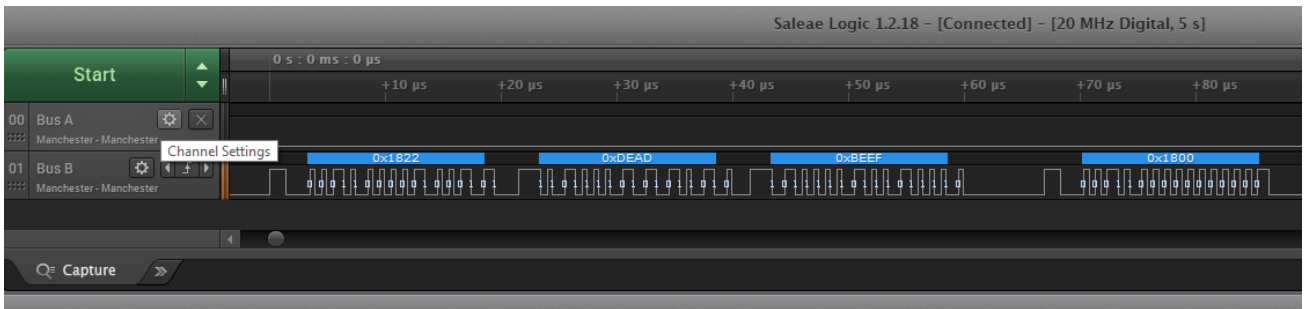
2. コマンド「a」の後にコマンド「h」を発行すると、時間が許す限り、フレーム間で高優先度の非同期メッセージを送信します。コマンド「a」の後にコマンド「l」を発行すると、時間が許す限り、フレームの終わりに低優先度の非同期メッセージが送信されます。
3. RT を BUSA に接続し、RT アドレスを 3 に設定し、オシロスコープやモニタでバスをモニタすると、以下のような応答があり、RT はクリアなステータス・ワードで応答するはずですが、BC は 2 つのコマンドを送ります。RX は RT3/SA1 へ 1 ワード、RT to RT は RT1/SA1 から RT3/SA1 へ 10 ワードです。このフレームが繰り返されている間に、「h」コマンドが発行され、高優先度の非同期メッセージが送信されます。以下のロジック・アナライザ画像は、この転送の様子を示しています。
以下の最初の画像は、メッセージ周期が～20mSec で、約 0mSec で BusA の周期的メッセージの間に BusB の高優先度メッセージが確認できます。



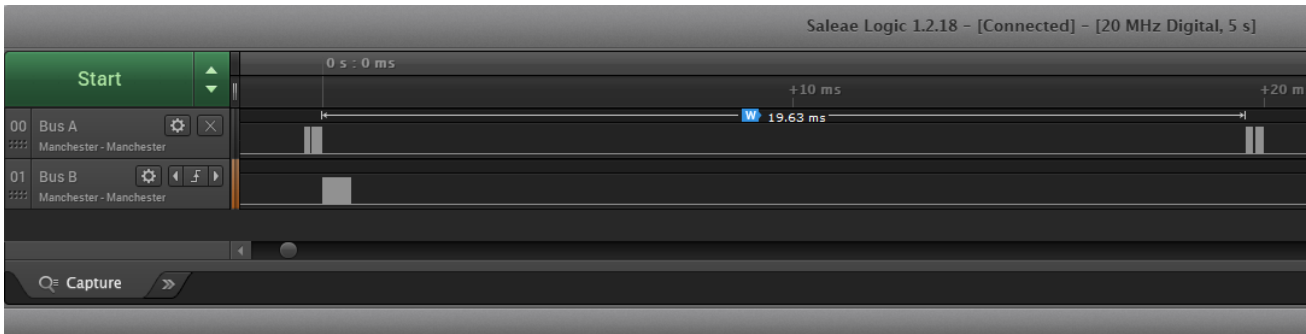
次の画像は、BusA の定期的な RX RT1 ワード・メッセージと RT ステータス・レスポンスの詳細を示しています。



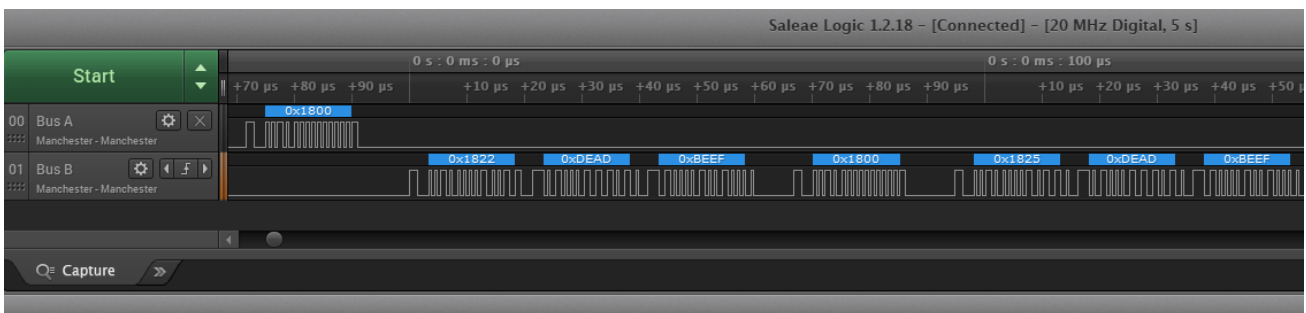
次の画像は、BusB の高優先度メッセージと RT ステータス応答を示しています。

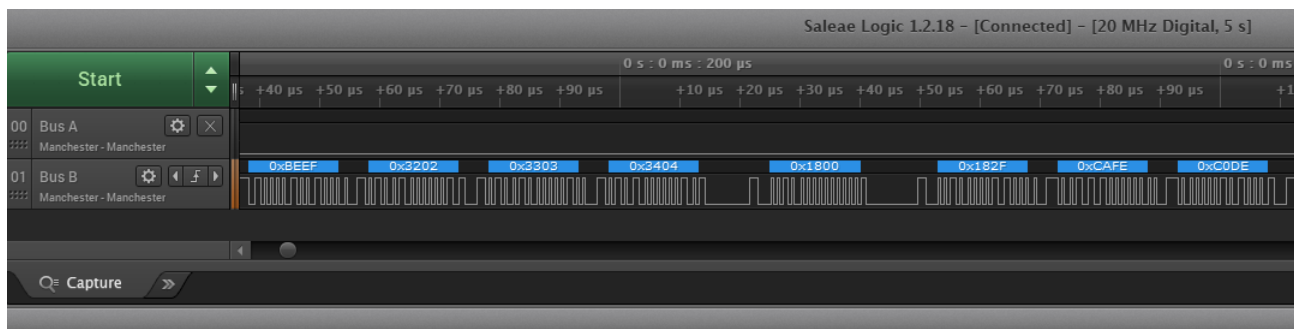


次の画像は、定期的なフレームの終わりにある低優先度メッセージです。



低優先度メッセージの一部を拡大したものです。





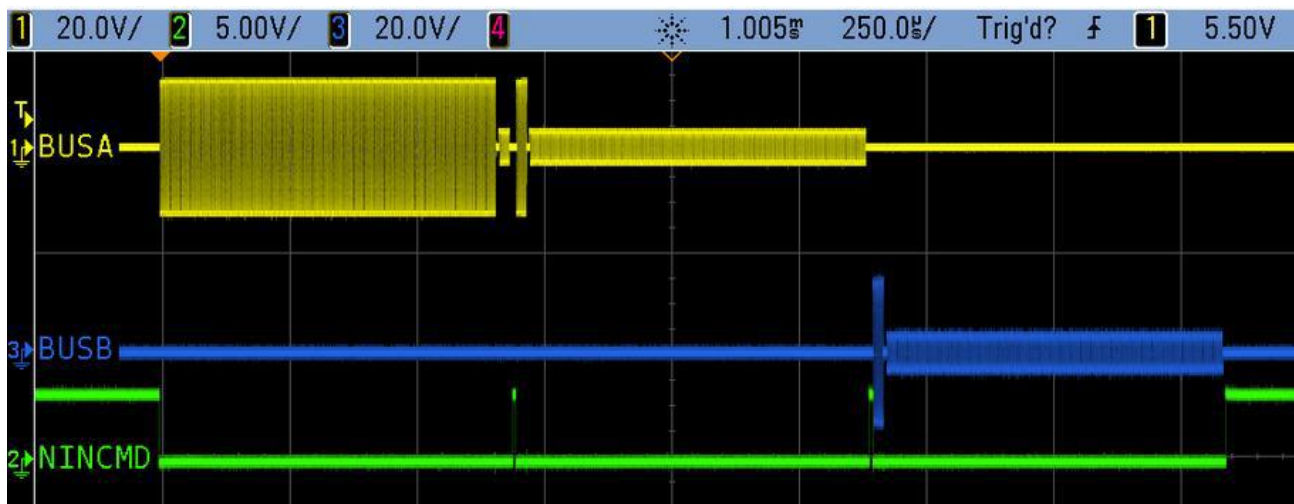
4. 「n」 コマンドは、BUSA と BUSB を使用して、5 つの Minor フレームの Major フレームを 3 つのメッセージで送信します。Major/Minor フレームのオシロスコープとロジック・アナライザの画像は以下の通りです。

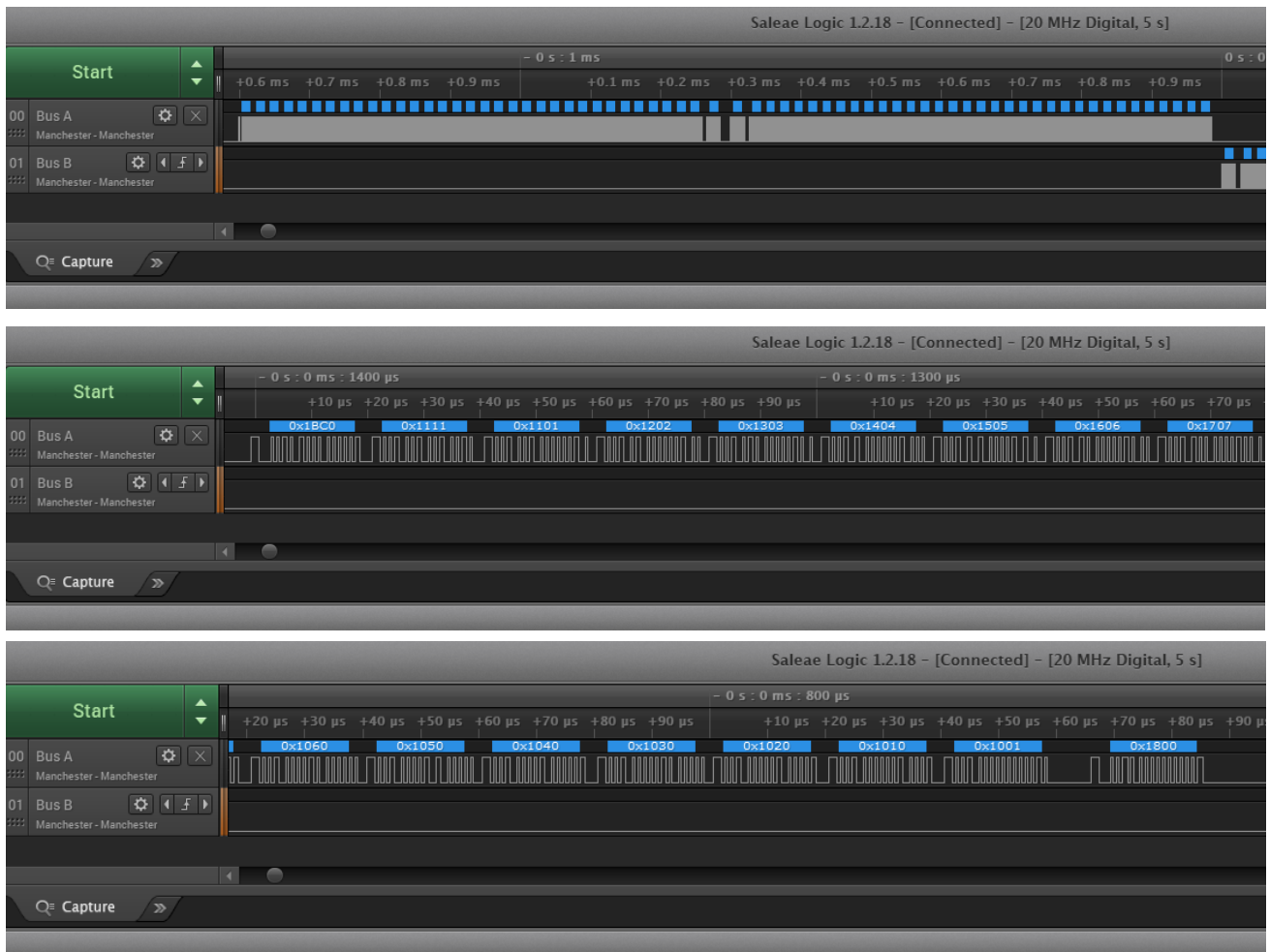
メッセージ×5 セット、合計 15 メッセージが送信されます。

メッセージ 1 バス A : 03-R-30-00 (RT アドレス 3, Rx, SA30, 32Words)

メッセージ 2 バス A : 03-T-30-00 (RT アドレス 3, Tx, SA30, 32Words)

メッセージ 3 バス B : 03-T-30-00 (RT アドレス 3, Tx, SA30, 32Words)





0x1800 ワードは、RT アドレス 3 がクリア状態で応答しています。

5. Major/Minor フレームのメッセージ・データは `demo.c` プログラムに含まれており、簡単に編集することができます。出荷時は RT アドレス 3 が使用されています。

RT モード、外部 BC 使用時

外部 BC テスター（バラード社製 USB1553 など）を使用して、デモ・ボードにメッセージを送信します。

1. 外部 BC を従来の 1553 バスで接続する場合、デモ・ボードの円形 3 軸バス・ジャックと A、B バス・ネットワークのバス・カプラ・ポートをケーブルで接続します。この場合、搭載されているダミー・バス負荷 70Ω 抵抗は外してください。バス・カプラが入手困難な場合は、搭載しているダミー・バス負荷 70Ω 抵抗 (R4、5) を有効にし、デモ・ボードのバス A、B 用 3 軸ジャックに BC テスター・ケーブルを直接接続してベンチ・テストが可能です。
2. 「w」コマンドでデバイスをリセットし、「b」コマンドで RT モードを設定します。これにより、DIP スイッチ SW1 で設定したアドレスで、HI-620x3 が RT ターミナルに設定されます。以下の画面が表示されます。トラフィックのトグルコマンド「t」により、トラフィックをリアルタイムに表示することができます。

```

COM5:115200baud - Tera Term VT
File Edit Setup Control Window Help
B: run the RT demo
K: enable RIMT operation
T: toggle RT or MT traffic display

Supplemental BC demonstrations
0: BC1 - simple BC sends RX MC17 to RT1

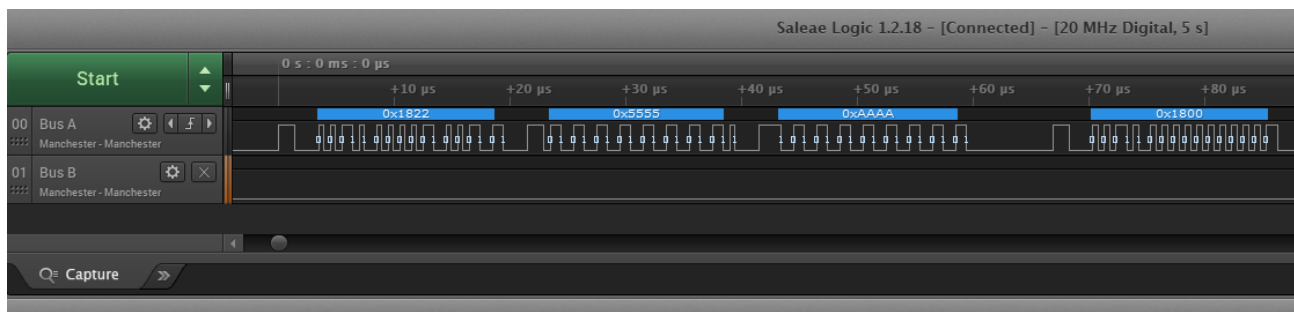
Supplemental RT demonstrations
1: RT1 - simple RT SA1 separate RX/TX buffers
2: RT2 - simple RT SA 30 RX/TX using the same buffer
3: RT3 - simple RT SA 30 increments TX data
4: RT4 - simple RT responds to MC17 using ISQ and interrupts

W: to reset the 62213
M: to show menu

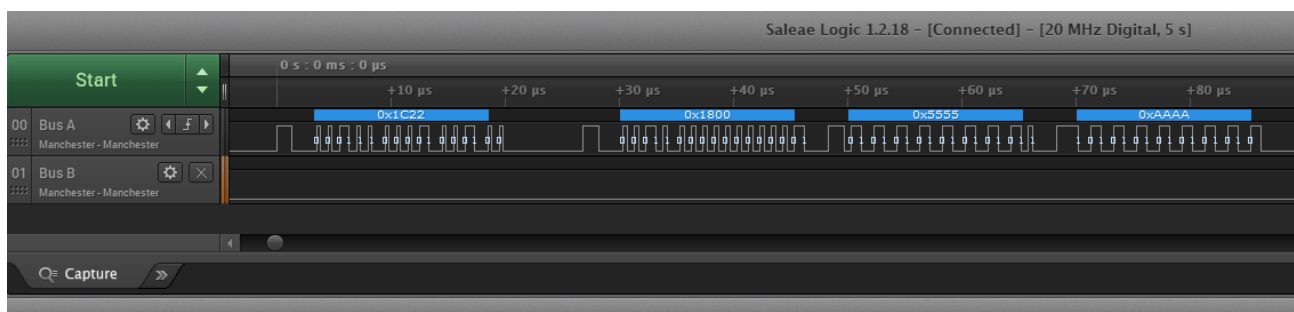
62213 ->
RtDemo
Using RT Address 3
62213 ->
Traffic Enabled
62213 ->

```

- RT はシングル・バッファ・モードに設定されています。この RT の設定は、RT SA1 からのデータの書き込みと読み出しをサポートし、ほとんどのモード・コードに対応しています。詳細は、Holts API ソフトウェアのマニュアルを参照してください。
- 1553 テスターで 03-R-01-02 Rcv コマンド (0x1822) をデータ 0x5555 と 0xaaaa で送信します。バス・データは以下のようなはずです。



- 次に 1553 テスターで 03-T-01-02 Xmt コマンド (0x1c22) を送信してください。以下のような Bus データになるはずです。



- コンソールにも以下のようにトランザクションが表示されるはずです。

```

COM5:115200baud - Tera Term VT
File Edit Setup Control Window Help
M: to show menu
62213 ->
RtDemo
Using RT Address 3
62213 ->
Traffic Enabled
62213 ->
MSG #0001. TIME = 00002042us   BUS A   TYPE0: BC to RT
CMD1 1822 --> 03-R-01-02
DATA 5555 AAAA
MSG #0002. TIME = 00046016us   BUS A   TYPE1: RT to BC
CMD1 1C22 --> 03-T-01-02
DATA 5555 AAAA

```

RT/MT モード、外部 BC 使用時

外部 BC テスター（パラード社製 USB1553 など）を使用して、デモ・ボードにメッセージを送信します。

1. 「w」コマンドでデバイスをリセットし、「k」コマンドで RT/MT モードにします。これにより、デバイスは RT ターミナルとモニタに設定されます。以下の画面が表示されます。

```

COM5:115200baud - Tera Term VT
File Edit Setup Control Window Help
B: run the RT demo
K: enable RIMT operation
T: toggle RT or MT traffic display

Supplemental BC demonstrations
0: BC1 - simple BC sends RX MC17 to RT1

Supplemental RT demonstrations
1: RT1 - simple RT SA1 separate RX/TX buffers
2: RT2 - simple RT SA 30 RX/TX using the same buffer
3: RT3 - simple RT SA 30 increments TX data
4: RT4 - simple RT responds to MC17 using ISQ and interrupts

W: to reset the 62213
M: to show menu

62213 ->
RIMT Demo
Using RT Address 3
62213 ->
Traffic Enabled
62213 ->

```

2. RT はシングル・バッファ・モードに設定されています。RT は上記の RT モードと同様に設定されていますが、さらにバス上のトラフィックを監視し、ターミナル画面にデータ・ログを報告します。1553 テスターから以下のコマンドを送信します。

- a. 03-R-01-02 Rcv コマンド (0x1822)、データ 0x5555、0xaaaa
- b. 03-T-01-02 Xmt コマンド (0x1c22)
- c. 04-T-01-02 Xmt コマンド (0x2422)

3. 画面には、以下のようなデータ・ログが表示されます。

```

COM5:115200baud - Tera Term VT
File Edit Setup Control Window Help
62213 ->
-RT Message-
MSG #0003. TIME = 00109036us   BUS A   TYPE0: BC to RT
CMD1 1822 --> 03-R-01-02
DATA 5555  AAAA
-RT Message-
MSG #0004. TIME = 00079670us   BUS A   TYPE1: RT to BC
CMD1 1C22 --> 03-T-01-02
DATA 5555  AAAA
-MT Message-
MSG #0005. TIME = 00124830us   BUS A   TYPE1: RT to BC
CMD1 2422 --> 04-T-01-02
ERROR: NORES

```

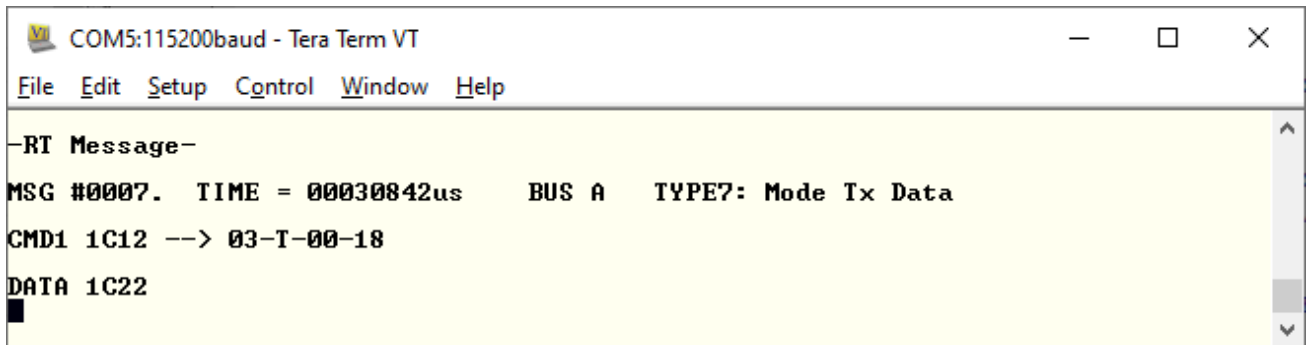
MSG #0001 と MSG #0002 は、BC コマンドに応答する RT です。MSG #0003 は MT からのものです。デモ・セットアップで RT4 が設定されていないため、BC からのバス・トラフィックを表示していますが、応答はありません。

表示されるメッセージの中で、以下のものがあります：

- 1 行目には、MSG#、タイムスタンプ、使用中の BUS、メッセージの種類が表示されます。
- 2 行目には、コマンドの 16 進コードとコマンドのサブテキストの略語が表示されます。
- 3 行目には、コマンドのデータ内容がフォーマットで表示されます：

RT ADD - Tx/Rx タイプ - サブアドレス - ワード長

4. モード・コード 18 (Transmit last command) を送信すると、上記のコマンド 03-R-01-02 が以下のように記録されますが、メッセージの Hex 内容 (0x1c22) だけが表示されることに注意してください。



```

COM5:115200baud - Tera Term VT
File Edit Setup Control Window Help
-RT Message-
MSG #0007. TIME = 00030842us   BUS A   TYPE7: Mode Tx Data
CMD1 1C12 --> 03-T-00-18
DATA 1C22
█

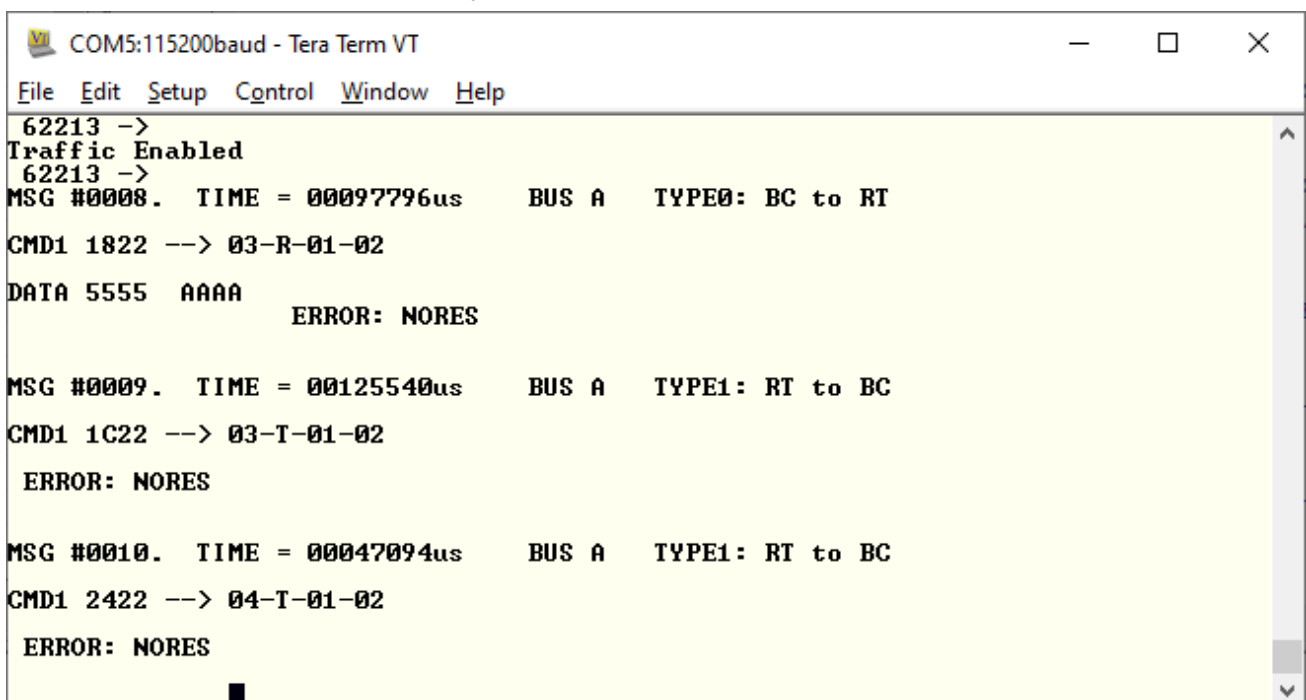
```

MT モード、外部 BC 使用時

外部 BC テスター（バラード社製 USB1553 など）を使用して、デモ・ボードにメッセージを送信します。

1. モニター・モードは RT/MT モードと非常によく似ていますが、RT が有効になっていないため、モニタの機能は上記の RT/MT の項と同じになります。
2. 「w」コマンドでデバイスをリセットし、「s」コマンドで MT モードに設定します。これにより、デバイスはモニタのみに設定されます。1553 テスターから以下のコマンドを送信します：
 - a. 03-R-01-02 Rcv コマンド (0x1822)、データは 0x5555 と 0xaaaa。
 - b. 03-T-01-02 Xmt コマンド (0x1c22)
 - c. 04-T-01-02 Xmt コマンド (0x2422)

以下の画面が表示されます。RT3 へのコマンドに対して、RT 有効時のようにデバイスが応答していないことに注意してください。



```

COM5:115200baud - Tera Term VT
File Edit Setup Control Window Help
62213 ->
Traffic Enabled
62213 ->
MSG #0008. TIME = 00097796us   BUS A   TYPE0: BC to RT
CMD1 1822 --> 03-R-01-02
DATA 5555 AAAA
      ERROR: NORES

MSG #0009. TIME = 00125540us   BUS A   TYPE1: RT to BC
CMD1 1C22 --> 03-T-01-02
      ERROR: NORES

MSG #0010. TIME = 00047094us   BUS A   TYPE1: RT to BC
CMD1 2422 --> 04-T-01-02
      ERROR: NORES
█

```

補助的な機能

その他、HoIt デバイスや API 関数の学習補助として、いくつかの関数がソース・コードの例として含まれています。helloRT のソース・コードでは、デバイスの初期化、メッセージ・ブロックとデータ・ブロックの設定、および合法化の設定方法を説明しています。helloBC1 関数は、モード・コードを送信するために BC を構成する方法を説明します。関数は次のとおりです：

helloRT1 - SA1 を使用する単純な RT で、RX、および、TX バッファを分離することを実証します。

helloRT2 - SA30 を使用したシンプルな RT で、共通の RX、および、TX バッファを使用することを実証します。RT に送信されたデータをエコー・バックするために便利です。

helloRT3 - helloRT2 と同じですが、すべての TX コマンドの後、バッファ内の 32 ワードのそれぞれを増分します。

helloRT4 - モードコード割り込みのためのユーザー割り込みサービス・ルーチンと同様に、割り込みステータス・キューを使用して RX MC17 を実装しています。

helloBC1 - 上記の helloRT4 で使用される RX MC17 を送信するための BC を構成します。

Getting Started with the Holt API demo software project and installing [Holt API デモ・ソフトのプロジェクトとインストールを開始する]

IAR Systems Embedded Workbench for ARM Compiler [IAR システムズ・エンベデッドワークベンチ for ARM コンパイラ]

IAR Systems Embedded Workbench for ARM Compiler をインストールした Holt API デモ・ソフトウェア・プロジェクトのスタートアップ。

IAR Systems Embedded Workbench for ARM (EWARM) コンパイラは、Holt デモ・プロジェクトを追加する前にインストールする必要があります。すべての Atmel ボード・ライブラリ・ファイルとデモ・プロジェクト・フォルダが適切な場所に作成されます。Holt USB DRIVE の Project フォルダにある「Holt HI-622x3 API Demo Project Installation Guide」に従ってください。次のステップに進む前に、そのガイドに従って IAR をインストールし、2 つの Holt プロジェクトフォルダを適切なフォルダ位置に配置する必要があります。ここからの説明は、上記のインストール作業が完了したものとします。

Windows のスタート・メニューから IAR Embedded Workbench を起動します。空白の画面が表示されるはずですが、IAR のファイル・プルダウン・メニューから Holt HI-622x3 API Demo Project を開き、ファイル／オープン／ワークスペースをクリックし、プロジェクト・フォルダの場所に移動して“HI-62xx.eww”を選択してオープンボタンをクリックしてください。

下図のように、左側に IAR ワークスペース・ウィンドウが表示されるはずですが、もし、ワークスペースのディレクトリ・ペインがない場合は、表示プルダウン・メニューから「Workspace」を選択してください。このウィンドウの中にあるフォルダ・グループを開いて、お好みのファイルを表示させてください。

メイン・ファイルをダブルクリックすると、以下のようなテキスト・エディタ・ペインが表示されます。

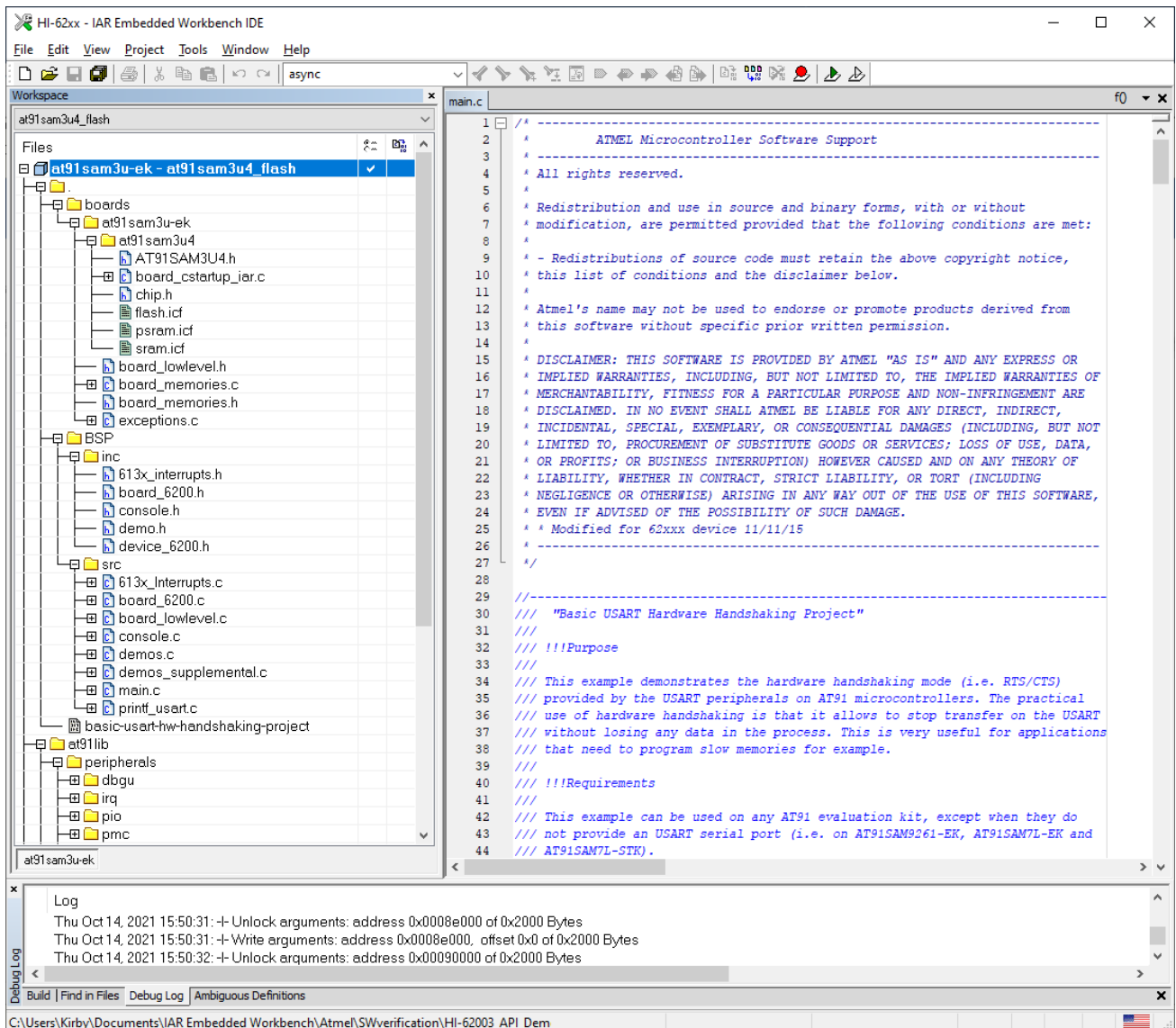
プロジェクトを初めて解凍し、適切なフォルダにインストールするときは、「Rebuild All」（プロジェクトのプルダウン・メニューから）を実行する必要があります。

IAR を開始する、プロジェクト管理などのガイドは、IAR Workbench のヘルプのプルダウン・メニューから入手することができます。

Mode Holt Project Configuration [モード Holt プロジェクト・コンフィギュレーション]

IAR プロジェクト・コンフィギュレーションは、プリプロセッサ・マクロを使用して、デモ・ソフトを再コンフィギュレーションするために使用されます。Holt は複数のプロジェクトで共通のコード・ベースを使用しています。つまり、このプロジェクトでは使用しないが、G ファイルや H ファイルに残された関数があります。IAR コンパイラはマクロを使用して、コードのどのセクションをビルドしてリンクするかを決定します。HI-6130 や HI-6131 への参照は、予想されることです。いくつかのケースでは、関数が単に使用されないか、適切なコードを選択するために関数内部でマクロが使用されま

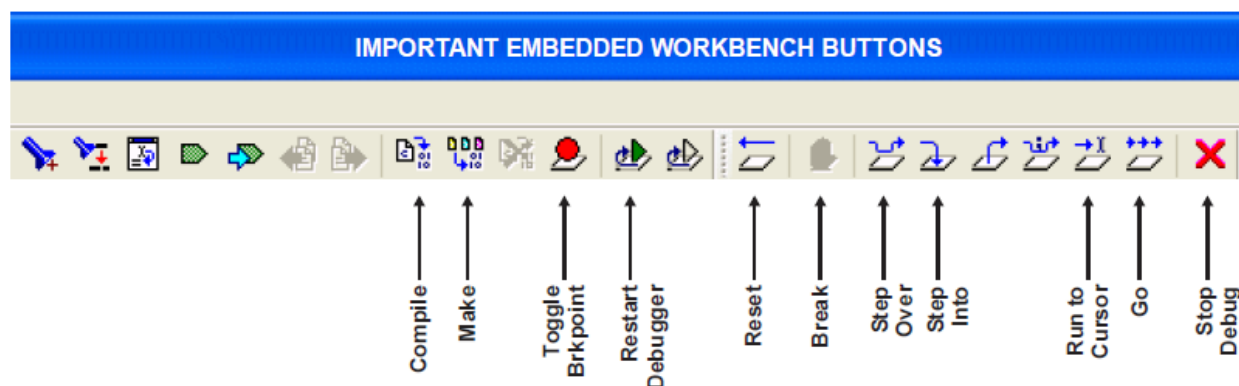
す。



プリプロセッサのオプションを選択するには、画面左上の at91sam--- の行の上にマウスを置き、右クリックします。下図のように「オプション」→「C/C++コンパイラオプション」→「プリプロセッサ」を選択します：

API ライブラリのソース・コードは、標準 ADK では提供されません。それ以外の標準 ADK デモは完全に機能しており、API プロジェクトは API C ソース・コードなしでビルドできます。

IAR embedded workspace は、よく使うタスクの多くを、図のように上部にショートカットボタンとして提供しています：



アプリケーション開発キットの注意事項

HI-62203TM は、16 Bit、または、8 Bit のデータバス・インターフェイスを持つマイクロ・コントローラと互換性を持つように設計されています。

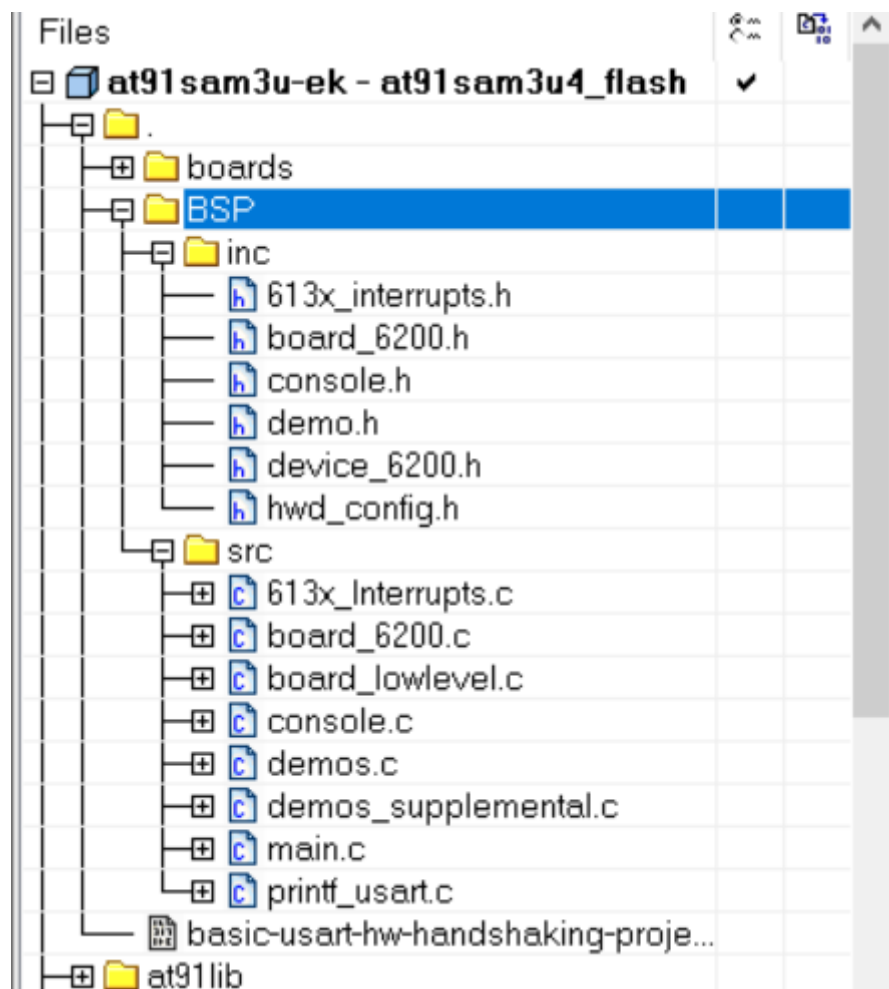
デバイスは、RAM、または、レジスタ・ロケーションのアドレス空間に分割されており、アドレス・ピン A17 に接続されている nMEM/REG 制御ピンを使用して選択します。メニュー・コマンド「d」を実行すると、0x00 から 0x01F までのすべてのレジスタ・アドレスが読み出されます。

ARM デバッガを使用する場合、レジスタのアドレス空間は、ARM アドレス 0x60000000 から始まり、RAM は 0x60020000 です。

TeraTerm を使用したコンソール I/O オプションには、保留中の割り込みレジスタの状態を含むレジスタの内容を読み出し、表示するメニュー・オプションが用意されています。保留中の割り込みビットは、読み出しが行われると自動的にリセットされることを忘れないでください。

Project File List with Selected Descriptions [プロジェクト・ファイル一覧と選択された説明]

デモ・プロジェクト・ファイルは、IAR のワークスペースに以下のように配置されています：



関数名のほとんどは説明不要ですが、いくつかの関数は、オリジナルの HI-6131、または、6130 デバイス用に書かれた名前を残していますが、HI-620x3 ファミリでは正しく機能します。

Board_6200.c

HI-620x3 の ARM クロック・レートとインターフェイスの設定に関する定義を扱います。

Board_lowlevel.c

ARM PLL の設定など、ハードウェアの設定を行います。

613x_interrupts.c

割り込み処理ルーチンやコンフィギュレーション設定

main.c

プログラムの入り口となる main() では、ターミナルを有効にするための初期化シーケンスが示されます。初期化が完了すると、関数呼び出しにより、有効なターミナル・モードで使用されるすべての RAM

構造に対する強力なアドレス指定方法が示されます。デモの初期化および実行には、このデモ・キットの `Holt` ハイレベル API を使用します。

ConfigureGpio(); ARM MCU 汎用 I/O を初期化します。

console.c

すべてのターミナル・モードで使用されるコンソール関数です：

```
ConfigureUsart1();
Show_menu();
chk_key_input();
list_all_regs();
```

BC モードで使用されるコンソール関数。

```
BcAsync();
MajorMinorframe();
```

```
RT で使用するコンソール関数：      RtDemo();
MT で使用するコンソール関数：      mtDemo();
MT/RT で使用するコンソール関数：    RtMtDemo();
```

MT モードで使用するコンソール関数：

プログラム・サイズを小さくするために、冗長な文字列を「printf」する原始的なコンソール関数です。

```
print_null(), print_sp1sp(), print_b1sp(), print_b0sp();
print_dddn(), print_dd0n(), print_dd1n();
print_menuprompt(), print_line();
```

`DisplayDecodedMsg()`：「t」モニタ・コマンド使用時に、1553 フォーマットのデータをコンソールに表示するために使用される。

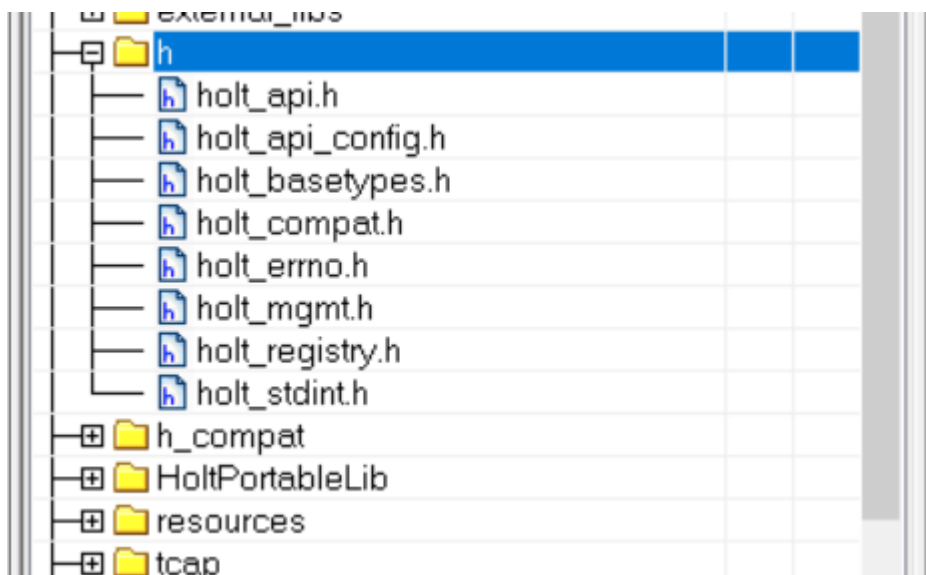
`displayRTTraffic()`：main から呼び出され、`DisplayDecodedMsg()` を呼び出す。

`flash.icf` - プロジェクト・リンカー・ファイル

Holt API Library files [Holt API ライブラリ・ファイル]

HoltPortableLib.a – プリコンパイルされた Holt API ライブラリです。
このファイルの拡張子は .a で、プリコンパイルされた Holt API ライブラリ・ファイルです。
注意：ファイル名は後のリビジョン番号を持つ可能性があります。

標準 API キットには Holt API ソース・ファイルは含まれていませんが、API ヘッダー・ファイルは含まれており、以下の workspace/h サブ・フォルダにグループ化されています。



基板ファイルと回路図のリンク

[BOM、および、回路図ファイル \(ARM マザーボード\) : EV-620x3CM3](#)

[BOM、および、回路図ファイル \(ドーターカード\) : EV-622x3DC](#)