



ADK-2130

HI-2130 API アプリケーション開発キット

2017 年 4 月



株式会社ナセル

履歴

履歴	日付	変更内容
AN-2130, Rev. New	2016 / 2 / 2	初版リリース
Rev.A	2017 / 4 / 5	新しい文書形式に変更 BOM を更新 ARM Cortex M3 ボードの回路図を更新

本文書は、Holt 社の『AN-2130』の和訳になります。

分かりにくい表現や誤訳がある場合は、Holt 社発行の英語版文書を参照してください。

Introduction [はじめに]

HI-2130 アプリケーション開発キットは (ADK)、MIL-STD-1553 用の HI-2130 マルチ・ターミナル IC の広範囲な機能セットを示しています。2 ボードから構成されるキットと C プロジェクトのリファレンス・デザインは、すぐに実行できる BC、MT、2xRT の任意の組合せの同時動作を評価可能なプラットフォームを提供します。便宜上、このキットには、IAR システムズの ARM 用 Embedded Workbench[®]、そして ARM Cortex M3 マイクロコントローラ用の完全に統合されたデバッグ・インターフェイスが含まれています。

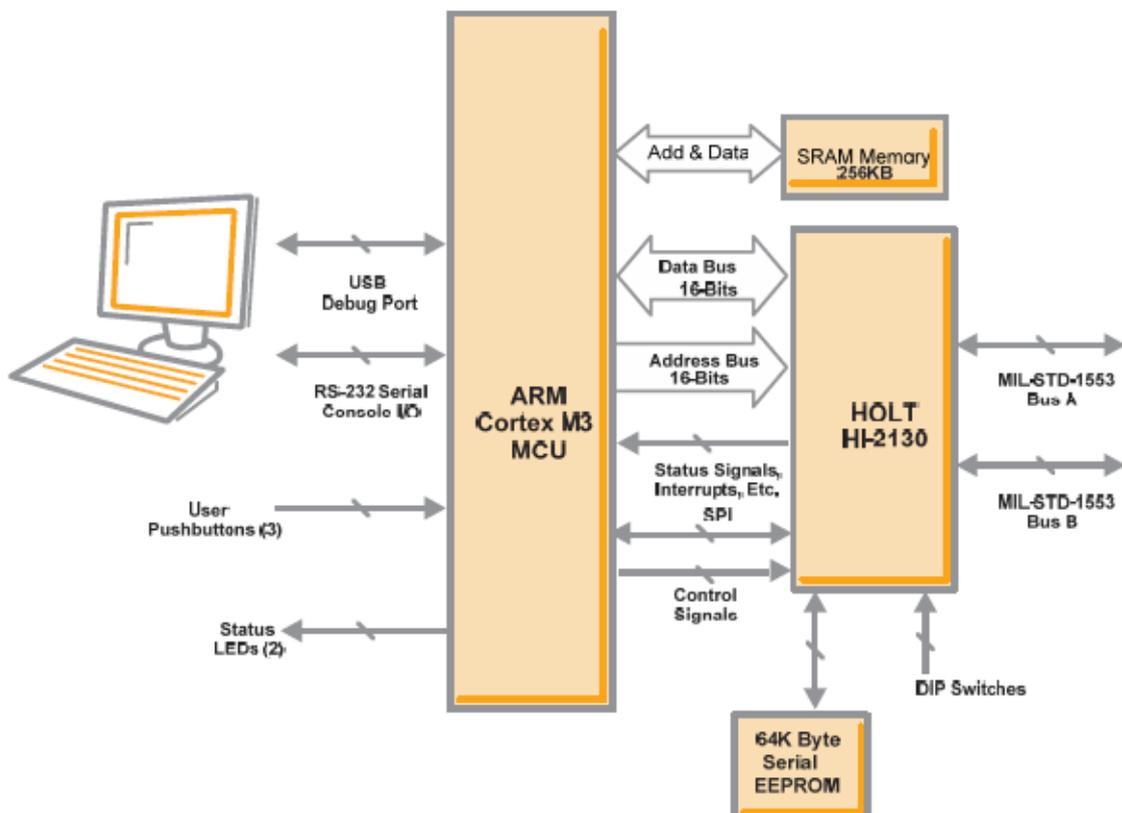
このガイドでは、ボードの設定および実行する方法について説明します。資料と必要なすべてのプロジェクト・ソフトウェアは、Holt の CD-ROM に含まれます。デモ・ソフトウェア・バージョンがすでにマイコンのフラッシュ・メモリにプログラムされています；提供されるソフトウェア開発ツールをインストールまたは実行する必要無く、ボードは箱から出してすぐに動作可能です。



Evaluation Kit Contents [開発キット構成]

- 本ユーザー・ガイド
- Holt HI-2130 プロジェクト・ソフトウェア、ドキュメント CD
- IAR システムズの ARM 用 Embedded Workbench[®] Ver.7.1 または、それ以上
- 5V DC 電源アダプタ
- デバッグ用 USB ケーブル
- RS-232 シリアル・ケーブル、DB9（オス）-DB9（メス）タイプ、PC 接続コンソール I/O 用
- HI-2130 ADK は、2 ボード構成で、
 - 上段 HI-2130 ボード：内蔵デュアル・トランスフォーマ・カップリング MIL-STD-1553 バス・インターフェイス。DIP スイッチにより、ボードの動作設定を行います。標準的な HI-6130 ADK と比較し、外部に 256KB の SRAM が将来のソフトウェア拡張のために追加されています。
 - 下段 MCU ボード：ARM Cortex M3 16/32Bit マイクロプロセッサ、デバッグ・インターフェイスおよび 3.3VDC レギュレート電源供給

Hardware Block Diagram [ハードウェア・ブロック図]



Hardware Design Overview [ハードウェア設計概要]

回路基板の部品表と回路図は、本ガイドの末尾を参照してください。HI-2130 は、外部 16Bit バスおよび、SPI シリアル・インターフェイスの両方を持ち、2 つの内蔵 MIL-STD-1553 絶縁トランスおよび、HI-6130 プロトコル IC で構成されています。HI-6130 または、HI-2130 の互換性については、この文書を参照するか、プロジェクトのソース・コード内を参照してください。HI-2130 データシートでは、半田ボールまたは、ピン・アサインおよび、PGA および BGA の寸法を示しています；全ての技術情報は HI-6130 のデータシートで包括されています。

標準の HI-6130 ADK と比べ、上段基板は 256KB の SRAM が追加されています。この RAM は、現在の標準のデモ・プログラムでは利用されていませんが、Holt の API を使用する目的で、将来のデモ・プログラムのバージョンで利用されることとなります。もう 1 つの違いは、MIL-STD-1553 絶縁トランスが HI-2130 に内蔵されていることです。HI-6130 (HI-6131) では、PCB 上に外部トランスを必要とします。

HI-2130 の上段のドーター・ボードは、ARM MCU ボードから取り外して分離可能なので、別のマイクロプロセッサや FPGA ボードへ接続することができます。ボード間のヘッダーは、0.1 インチ (2.54mm) ピッチの汎用的なコネクタで接続されています。すべてのホスト・インターフェイス信号は、ボード間のヘッダーを介して行われます。多数の HI-2130 コンフィグレーション・ピンおよびリモート・ターミナル・アドレス設定端子は、上段の HI-2130 ボードの DIP スイッチによって設定されます；これらの信号は、ボード間のヘッダー上の MCU では使用できません。

下段の ARM Cortex M3 基板は、フラッシュ・プログラム可能な、Atmel の AT91SAM3U-EK マイクロプロセッサを使用しています。16Bit のパラレル・バス・インターフェイスおよび SPI インターフェイスが、MCU と 1553 インターフェイスのために HI-2130 に ADK 上に設けられています。HI-2130 インターフェイスは「その場で」切換えることができますが、通常的设计ではどちらか 1 つの MCU インターフェイスを使用します。デモ・プログラムでは、16Bit パラレル・バス、シリアル SPI の両方が設定可能です。この文書の後半では、SPI モードでの EBI デバックの章を参照してください。UART ベースのシリアル・ポートは、RS-232 コンソール I/O (オプション) を提供します。USB2.0 ポートは、将来の拡張用に使用可能です。2 つのプッシュ・ボタン SW1、SW2 は、ARM MCU ボード上のソフトウェアとの対話のために利用可能です。REST プッシュ・ボタン SW3 は、HI-2130 マスター・リセット信号を制御し、ARM マイクロプロセッサをリセットします。上段のドーター・カードでは、SW10 にユーザーが利用可能な予備の 4 つの DIP スイッチが用意されています。

ドーター・カードにはステータス表示用の 5 つの LED があります。電源投入時それらは、動作を実証するために順番に点滅します。「RT トラフィック」プログラム起動がコンソールの「T」コマンドによって有効になっている場合、対応するバス上でメッセージが受信されたときに、Bus A および Bus B LED が点滅します。このオプションはデフォルトで OFF になっています。LED 3、4、5 はプログラムの異なるセクションで再利用されています。緑と赤の LED は AUTOEN/EECOPY 要求のステータスを表示します。この文書の後半の、AUTOEN/EECOPY 操作に関する章を参照してください。緑 LED はメッセージ割込みハンドラ中に ON になり、LED の右側にプローブを接続することによって割込みサービス時間を測定するのに便利な場所です。赤 LED は `Delay_ms()` 関数中に点滅します。これらの LED は容易に再利用されます。

ARM Cortex M3 ボードは、www.segger.com からライセンスされた「オンボード J-Link」デバッグ・インターフェイスが含まれています。高価な JTAG デバッグ・ケーブルを購入することなく箱から出してすぐに使用することができます。キットには、ご使用のコンピュータに、ボードのデバッグ・インターフェイスを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（ユーザーがすでに ARM デバッグ・インターフェイス用のリボン・ケーブル・コネクタを所有している場合、ARM 標準の 2x10 のデバッグ・コネクタでデバッグ接続が可能です。この場合、下側基板の底面のジャンパ JP2 をショートさせ、「オンボード J-Link」を無効にする必要があります。）。

有効な BC および/または RT によって 1553 メッセージが処理されている間に、ACTIVE (TP8) テスト・ポイントは初期化後に HI-2130 によってアサートされ、SMT モニタの実行時にオプションで記録されます。ACTIVE は有用なスコープ・トリガ信号です。それは、メッセージ開始およびメッセージ終了のレジスタ/割込み更新後にリセットされます。

4 つの SPI ホスト・インターフェイス信号 SCK、MISO、MOSI、スレーブ・セレクト nCE は、プローブやロジック・アナライザをフックするために J9 上の 8 ピンヘッダを監視することができます。不慣れなプロセッサや FPGA を使用して新しい設計を開始すると、ロジック・アナライザは SPI タイミングを確認するのに役立つのでお勧めします。こういった目的のために、Saleae Logic Pro 16 などの USB ロジック・アナライザが有用です。SPI nCE ピン (SPI スレーブセレクト) は SPI ローレベル・ドライバ・コード内で、直接 GPIO によって制御されます。nCE は継続的に中断することなく、OP コードと N 個データワード転送のすべての期間 LOW にアサートされなければなりません。

External Bus or SPI Host-1553 Interface [外部バスまたは SPI ホスト 1553 インターフェイス]

HI-2130 に MCU をインターフェイスする方法が 2 つあります: 16Bit パラレル・インターフェイスまたは、4 線 SPI ポートです。ホストの SPI ポートの最大 SCK クロック・レートは 20MHz です。高負荷の 1553 バス使用率または、メッセージ・データ取引の高速ホスト・アクセスを必要とするアプリケーションでは、16Bit パラレル・バスの使用を推奨します。デモ・プログラムでは、**613x_initialization.h** ヘッダー・ファイル内の C マクロ **HOST_BUS_INTERFACE** の値を設定し、プログラムを再コンパイルし、ボードにダウンロードすることで両方の設定が可能です。IAR インストールまたはデモ・コードを実行する前に以下のクイック・スタート・ガイドに従ってください。デモ・ボードは SPI バージョンでプログラムされています。

SPI クロック・データの入力/出力は、HI-2130 にシリアルで行われるので（および、SPI は OP コードを使用し、メモリ・アドレス・ポインタにより SPI トランザクションを定義しているため）、SPI データ転送は同等のパラレル・バス (EBI) の転送よりも遅いです。SPI の更なる欠点は、IAR Embedded Workbench® のデバッグの標準的なデバッグ「メモリ」または「メモリ・ウォッチ」ウィンドウに HI-2130 のレジスタおよび RAM メモリの内容を表示することができないという点です。HI-2130 を EBI モードで使用すると、SRAM のように見えます；デバッグは、「メモリ」ウィンドウを使用して、簡単にメモリ・マップ・レジスタとメモリ・データ構造を調べることができます。SPI のみの動作の場合、シリアル・コンソールは便利なデバッグ・ツールを提供します。**printf()** 文を配置して、コンソールから値を出力します。デバッグ時に SPI レジスタ/メモリ表示制限を軽減するために、特別に事前プログラムされたコンソール・コマンドによって、コンソール経由で重要なレジスタおよび、コントロール、データ・ブロックの表示を提供します。カスタマイズされた **printf()** のシーケンスは、ユーザーによって実行することができます。

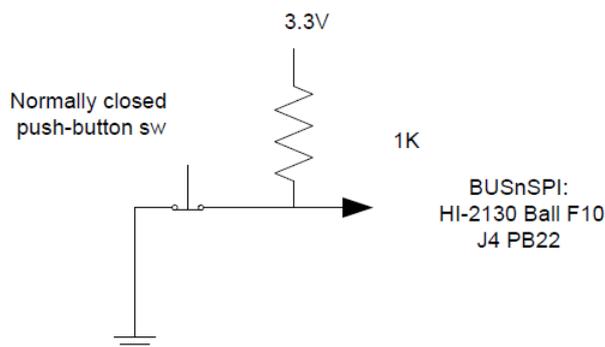
Optional EBI parallel debugging technique for SPI designs [SPI 設計時における、オプション EBI パラレル・デバッグ方法]

SPI ホスト・インターフェイス設計のため、EBI ホスト・パラレル・バスが、ソフト開発のために一時的に利用可能である場合、プログラムの実行が停止しているときに、必要に応じてデバッガ「メモリ」または、「メモリ・ウォッチ」ウィンドウを更新するために、バス・インターフェイスを使用することが可能です：

SPI ホスト・インターフェイスがデフォルトで使用されていると仮定します。実行が停止され、プログラマが「メモリ」ウィンドウをリフレッシュしたい（または新しいものを開く）場合、BUSnSPI ピンは、一時的に High になり、一方でプログラム・コードの一行はシングル・ステップされます。このアクションによって IAR デバッガが、レジスタやメモリへのアクセス、デバッグ「メモリ」ウィンドウを更新します。

これが機能するためには、16Bit アドレスおよびデータ・バスが HI-2130 に接続されている必要があります。**613x_initialization.h** ヘッダー・ファイルのマクロ、**HOST_BUS_INTERFACE** を 0 に設定する必要があり、**BOTH_EBI_SPI** を 1 に設定する必要があります。この EBI バスおよび SPI ポートの初期化、および Atmel MCU GPIO を出力にする代わりに、BUSnSPI ピンをアクティブ入力にします。

プルアップ抵抗（1K）は、J4 PB22 から 3.3V に、HI-2130 BUSnSPI ピン（半田ボール F10）に接続する必要があります。HI2130 は内部に弱いプルダウン・レジスタ（～50KΩ）を持っているので、1K はプッシュ・ボタンがオープンになった時の信号をプルアップするのに十分です。



デバッガ「メモリ」ウィンドウのリフレッシュが必要な場合は、プログラムの実行を停止または、SPI トランザクション・シーケンス内にブレークポイントを使用します。一時的にスイッチを開き（ピンが High になり）、デバッガで 1 行または複数行のコードをシングル・ステップ実行します。デバッグの「メモリ」ウィンドウは、1 ステップごとにリフレッシュされます。SPI を使用してフルスピード（RUN）実行を再開する前に、BUSnSPI スイッチを閉じます（Low にします）。

デバッグが完了したら、コンパイルする前に、**613x_initialization.h** ヘッダー・ファイルで **BOTH_EBI_SPI** を 0 に、**HOST_BUS_INTERFACE** を 0 に設定します。

この方法の更なる例は次項を参照してください。

A Quick Demonstration [クイック・デモンストレーション]

Holt HI-2130 アプリケーション開発キットは、BC、SMT バス・モニタ、2x独立 RT（RT1 と RT2）として動作するようにあらかじめプログラムされています。2xRT のターミナル・アドレスは、電源を投入する

前に DIP スイッチ (上段基板の) を使用して設定します。デモの目的のために、RT1 の DIP スイッチは RT アドレス 3 (0-0-0-1-1-1-パリティ 1)、RT2 は RT アドレス 4 (0-0-1-0-0-パリティ 0) に設定する必要があります。これらの値は、事前にプログラムされたバス・コントローラのメッセージ・レパートリーによって動作します。2 つの 6 ポジション DIP スイッチは、これらのアドレス+奇数パリティに設定されています。ユーザー・ガイド、ソース・コード、およびソフトウェア・ドキュメントでは RT1 を RT として参照してください。他のすべての DIP スイッチは、ボードの写真に示すデフォルト位置に設定されています。

1. ボードがスタンドアロンで動作することを保障するために、ARM 用 IAR Systems Embedded Workbench® (EWARM) および Holt デモ・プロジェクト・フォルダをインストールする前に、デモ・ボードのクイック・デモンストレーションを行ってください。それは、箱から出してすぐに検証できるように、デモ・プログラムが Atmel の ARM Cortex M3 プロセッサ内のフラッシュ・メモリにプログラムされています。IAR EWARM および Holt デモ・プロジェクトは、**後で指示されたときにインストールします。**
2. デモンストレーション・プログラムは、コマンド・メニューと表示メッセージ・トラフィック情報を提供するために、コンソール (PC へのシリアル・ポート) を使用します。コンソール I/O (初期評価時は推奨) を使用する場合、ご使用のコンピュータはシリアル (COM) ポートと TeraTerm などの「ターミナル・エミュレーション」ソフトを必要とします。ほとんどのデスクトップ・コンピュータは COM ポートを持ちますが、多くのノート PC は COM ポートがありませんので、USB-シリアル・アダプタが必要となります (Holt からは提供しません)。デバッガが直接「メモリ」ウィンドウに表示するための HI-2130 のレジスタや RAM にアクセスすることはできませんのでコンソールはデバッグ支援として非常に有用です。

Windows7 を推奨します。インストール手順は、Windows7 のディレクトリ名を参照しています。

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 *TeraTerm4.71* を、Holt 提供の CD 内にある、「*teraterm-4.71.exe*」インストール・プログラムを実行して、インストールしてください。再配布は、著作権表示が保持されることを条件に許可された旨のライセンス契約を受け入れます。通知は、「*Help*」→「*About TeraTerm*」をクリックすることで TeraTerm ウィンドウから表示することができます。インストールを続けます・・・

- デフォルトのインストール先をそのまま使用し、「*Next*」をクリックします。
- コンポーネント選択画面で、追加のプラグイン=TTXResizeMenu を除いたすべてのオプションを選択解除し、「*Next*」をクリックします。
- インストール言語を選択し、「*Next*」をクリックします。
- デフォルト・スタート・メニュー・フォルダを許可し、「*Next*」をクリックします。
- ショートカット作成を選択し、「*Next*」をクリックします。
- インストール画面で「*Install*」をクリックします。

TeraTerm プログラムを実行します。「**新しい接続**」画面で (x) シリアルを選択し、COM ポートを選択します。シリアル・ポート設定ウィンドウを開くために、「設定」→「シリアル・ポート」をクリックします。設定を変更します、ボーレート: 115200、データ: 8 bit、パリティ: none、ストップ: 1 bit、フロー制御: none。付属の DB9 シリアル・ケーブルを使用して、MCU ボードとコンピュータのシリアル (COM) ポートを接続します。

3. コンソール・ポートの設定後、DB9 シリアル・ケーブルを PC へ接続します。

-
4. 後で指示があるまで、下段（MCU）ボード上のデバッグ・ポートと PC の間に付属の USB デバッガのケーブルを接続しないでください。
 5. BUS A のバスの動作を観察するには、BUS A+とラベルされた赤のテスト・ポイントおよび、BUS A-のグランドにオシロスコープを接続します。BUS B のバスの動作を観察するには、同じようにプローブを BUS B の同じ点に接続します。ACTIVE というラベルのテスト・ポイントは、便利なスコープのトリガ信号で、メッセージ開始時に High になりメッセージ終了時に Low になります。
 6. MIL-STD-1553 バスにケーブルで接続されていない場合、BUS A と BUS B の、それぞれの黒と赤のテスト・ポイント間に、 70Ω （ $70\sim 80\Omega$ の任意の値） $1/2W$ 抵抗を接続することにより、ダミー負荷を提供します（このデモンストレーションでは、デューティ・サイクルが十分に小さいので、 $1/2W$ で十分です）。オンチップの RT に向け BC メッセージを生成するために、オンチップの HI-2130 を使用する場合は、外部 75Ω の負荷抵抗を使用します。実際の MIL-STD-1553 バスに接続するためのバス・カップラを使用する場合は、 75Ω ダミー負荷抵抗を使用しないでください。
 7. AUTOEN (SW1) および、COPYREQ (SW2) とラベルされた SW8 の DIP スイッチを OFF（ダウン・ポジション）に設定します。
 8. 5V の電源を接続し、下段の回路基板の電源ジャックにケーブルを接続します。電源投入時 LED は信頼性を実証するために、順次点滅します。
 9. メニューが、ボードに電源が供給されたときまたは、RESET プッシュ・ボタンが押されたときに表示されます。評価ボードで正しい TeraTerm の通信を検証した後、ターミナル設定は **Setup->Save Setup** をクリックすることで保存できます。

以下の日付と時刻は、コンソール画面とは異なります。

```

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

Holt Integrated Circuits HI-2130 Combo Demo Ver: 1.0
Compiled: Jan  8 2016 08:43:43
SPI Bus

      Host is Initializing Regs & RAM
Waiting for 2130 Ready pin high...OK

*****
Holt Integrated Circuits HI-2130 Combo Demo
Compiled: Jan  8 2016 08:52:18
*****

      BC On   SMT On   RT1 On   RT2 On

Press '1' to step BC and list results...
Press '2' to list BC configuration...
Press '3' to list BC condition codes & GP flags...
Press '4' to list MT configuration...
Press '5' to list MT results, last msg...
Press '6' to list HW interrupt status...
Press '7' to list BC interrupt status...
Press '8' to list RT interrupt status...
Press 'I' or Spacebar RT Traffic Toggle on/off.
Press '9' to list MT interrupt status...
NOTE: Options 6-9 clear the accessed Pending Interrupt Register!

SPI 2130 reg/memory Display Commands:
Press Space Bar Display Registers 0x-0x0050
Press 'A' Display Memory at address
Press 'Z' Display RT control block
Press 'X' Display RT data block
Press 'C' Display RT2 control block
Press 'U' Display RT2 data block
Press 'B' Display BC msg and data block
Press 'I' Display Interrupt Log
=====
Press 'M' for menu, or press any valid menu key. >> □

```

PC キーボードのスペース・キーを押すと、HI-2130 システム・レジスタを表示します。コマンドは大文字または小文字が使用可能です。

```

2130 Registers:
  0   1   2   3   4   5   6   7   8   9   A   B   C   D   E   F
Adr 0000: 19F0 8100 0000 0000 0000 0000 0000 0000 0000 0180 527A 0000 0000 0000 7818
Adr 0010: FFF8 0000 D458 0058 FBF8 0000 D5A8 00EF 1E00 0400 0000 0000 AAAA BBBB 0000 ABCD
Adr 0020: 01CE 2200 0600 0000 0000 AAAA BBBB 0000 ABCD 0800 0000 0000 0000 0000 0000
Adr 0030: 0000 0000 D02D 1B70 1B70 0000 0000 8000 00C0 000F 0000 0000 0000 0000 0000
Adr 0040: 0000 0000 0000 C092 004B 0000 0000 0000 0000 C0BF 0000 C0CD 0000 0000 0000
Adr 0050: 0000
>> □

```

レジスタ 0 (0000) は、マスター・コントロール・レジスタ、そして、レジスタ 1 はマスター・ステータスおよびリセット・レジスタです。ワード・アドレス 0x0001 の MSB が High (8) であることに注意してください。これは、HI-2130 がホスト・アクセスの準備ができていることを示す READY ビットです。Tera Term ウィンドウを十分に広げて、全部の 16 個のレジスタが見やすいようにしてください。

「M」を再度押すと、有効なコマンドのメニューを再度表示します。

プログラムが SPI モードで設定されたとき、追加のコマンドは、重要な HI-2130 コントロール・ブロックおよびデータ・ブロック・メモリ・セクションの表示に利用可能です。これらのコマンドは Z、X、C、V、B、I です。コマンド「A」を入力すると、16Bit アドレスは、そのアドレスから始まる 256 ワードが表示されます。

RT の動作に必要なのは、RT メッセージおよび対応するデータ・バッファの制御ブロックです。

「Z」を押すと RT1 のコントロール・ブロック・メモリを表示します。これは、初期化された RT1 コントロール・ブロック構造体の始まりを示しています。

サブアドレス SA1 コントロール・ブロックは、アドレス 0x0404 から始まる 4 ワードを持っています。

アドレス 0x0404 = 0x4104 – SA1 コントロール・ワード (IWA 割り込み、ピンポン・バッファ・モード)

アドレス 0x0405 = 0x0800 – バッファ A ポインタ・アドレス

アドレス 0x0406 = 0x0822 – バッファ A ポインタ・アドレス

アドレス 0x0407 = 0x0844 – ブロードキャスト・データ・ポインタ・アドレス

これらのデータ構造の完全な説明に関しては、HI-2130 のデータシートを参照してください。

RT1 Control Blocks																
Adr	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Adr 0400:	D0AD	DEAD	DEAD	DEAD	4104	0800	0822	0844	C000	0A00	0000	0D10	8001	1176	1176	15B6
Adr 0410:	8042	1E00	1E00	1C00	8000	0C00	0005	1A36	0000	1A36	0000	1A36	0000	1A36	0000	1A36
Adr 0420:	0000	1A36														
Adr 0430:	0000	1A36														
Adr 0440:	0000	1A36														
Adr 0450:	0000	1A36														
Adr 0460:	0000	1A36														
Adr 0470:	0000	1A36	0000	1A36	0000	1A36	0000	1A36	C000	08AE	0000	08AE	D0AD	DEAD	DEAD	DEAD
Adr 0480:	D0AD	DEAD	DEAD	DEAD	4000	0866	0000	08AA	8000	0D32	0020	1172	8001	15D6	15D6	1A16
Adr 0490:	8052	1E00	1E00	1C00	0000	1A58										
Adr 04A0:	0000	1A58														
Adr 04B0:	0000	1A58														
Adr 04C0:	0000	1A58														
Adr 04D0:	0000	1A58														
Adr 04E0:	0000	1A58														
Adr 04F0:	0000	1A58	0000	1A58	0000	1A58	0000	1A58	C000	08AE	0000	08AE	D0AD	DEAD	DEAD	DEAD
RT1 MC blocks																
Adr	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Adr 0500:	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
Adr 0510:	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
Adr 0520:	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
Adr 0530:	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
Adr 0540:	0000	0000	0000	0000	4000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
Adr 0550:	4000	0000	0000	0000	4000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
Adr 0560:	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
Adr 0570:	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
Adr 0580:	4000	0000	0000	0000	4000	0000	0000	0000	4000	0000	0000	0000	4000	0000	0000	0000
Adr 0590:	4000	0000	0000	0000	4000	0000	0000	0000	4000	0000	0000	0000	4000	0000	0000	0000
Adr 05A0:	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
Adr 05B0:	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
Adr 05C0:	4000	0000	0000	0000	0000	0000	0000	0000	4000	0000	0000	0000	4000	0000	0000	0000
Adr 05D0:	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
Adr 05E0:	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
Adr 05F0:	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 0x0500-0x05FF

>> ■

下部にあるサブ・メニューは、メモリ空間内を上下に移動することができます。

RT1 データ・バッファ領域を表示するには、「X」を押します：

RT1 Data blocks																
Adr	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Adr 0800:	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
Adr 0810:	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
Adr 0820:	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
Adr 0830:	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
Adr 0840:	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
Adr 0850:	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
Adr 0860:	0000	0000	0000	0000	0000	0000	0000	0000	0101	0202	0303	0404	0505	0606	0707	0808
Adr 0870:	0909	1010	1111	1212	1313	1414	1515	1616	1717	1818	1919	2020	2121	2222	2323	2424
Adr	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Adr 0880:	2525	2626	2727	2828	2929	3030	3131	3232	0000	0000	F001	F002	F003	F004	F005	F006
Adr 0890:	F007	F008	F009	F00A	F00B	F00C	F00D	F00E	F00F	F010	F011	F012	F013	F014	F015	F016
Adr 08A0:	F017	F018	F019	F01A	F01B	F01C	F01D	F01E	F01F	F020	0000	0000	0000	0000	0000	0000
Adr 08B0:	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
Adr 08C0:	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
Adr 08D0:	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
Adr 08E0:	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
Adr 08F0:	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
Adr	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Adr 0900:	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
Adr 0910:	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
Adr 0920:	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
Adr 0930:	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
Adr 0940:	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
Adr 0950:	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
Adr 0960:	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
Adr 0970:	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
Adr	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Adr 0980:	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
Adr 0990:	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
Adr 09A0:	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
Adr 09B0:	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
Adr 09C0:	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
Adr 09D0:	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
Adr 09E0:	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
Adr 09F0:	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 0x0900-0x09FF
 =====

General structure of demo functions [デモ機能の一般的な構造]

Holt HI-2130 コンボ・デモ・プログラムは、コンソール・コマンドを待っているループに入る前に、*main.c* の関数を呼び出してターミナルを初期化し、コンソール・メニューを表示します。デフォルトでは、すべてのターミナルが初期化され、有効化されているため、MIL-STD-1553 メッセージは自動的に処理されます。

デモの BC 部分は、10 メッセージ・リストから送信するように構成され、「1」キーが押されるたびに 1 つのメッセージが送信されます。10 番目のメッセージが処理されると、メッセージ・リストが繰り返されます。次のメッセージが使用されます：

(これらのメッセージ構築方法を示す配列構造については、*613x_bc.c* を参照してください。)

Msg Block 1	= SA Rx コマンド 03-1-30-00 (ループバック・サブアドレス)	バス A
Msg Block 2	= SA Tx コマンド 03-1-30-00 (ループバック・サブアドレス)	バス B
Msg Block 3	= SA Rx コマンド 03-0-01-00 (SA1)	バス A
Msg Block 4	= SA BRx コマンド 31-0-05-11	バス B
Msg Block 5	= SA BRx コマンド 31-0-30-00 (ループバック・サブアドレス)	バス A
Msg Block 6	= Mode Code Tx コマンド 03-1-31-02 (Tx モード・コード 2)	バス B
Msg Block 7	= Mode Code Tx コマンド 03-1-31-18 (Tx モード・コード 18)	バス A
Msg Block 8	= Mode Code Rx コマンド 03-0-31-21 (Rx モード・コード 21)	バス B
RT-RT Msg Block 1	= RT-RT メッセージ・コマンド 04-0-30-02 03-1-05-02	バス A
RT-RT Msg Block 2	= RT-RT メッセージ・コマンド 31-0-30-11 04-1-05-11	バス B

シンプルなデモンストレーションでは、内部 BC を使用してこれらのメッセージを送信します。結果を RT バッファ・メモリに表示するか、「T」コマンドを使用して RT トラフィックをコンソールに表示します。BUSA と nBUSA のテスト・ポイント間に 75Ω 1/2W 抵抗を接続してバスを終端するだけです。バス・ターミネータが使用可能な場合は、BUSA ケーブル・ジャック（およびオプション BUSB）に接続します。

デモ手順：

終端抵抗を接続

「T」を押すとトラフィック・コンソール表示が有効になります。

BC メッセージを送信し、結果をコンソールで確認するには、「1」を押します。

この表示には 2 つの部分があります。上の部分は BC メッセージの詳細を示し、「MSG #0001」で始まる最後の部分は RT メッセージ割り込みが取り込むデータです。新しいメッセージがメイン・ループで検出され、表示されます。

```

Press 'M' for menu, or press any valid menu key. >>
Traffic Enabled

Results From Last Message Issued by BC
=====
BC Message # 1
Message Type: Rx Subaddress Command, 32 data words
CW: 0x1BC0 = 03-0-30-00      SW: 0x1800 = RT03 CS

BC Control Word: 0x4180
MEMask RetryEna UseBusA NonBcstSA

Block Status Word: 0x8000
EOM Bus A

Condition Code Register: 0x8000
BC Running: No Condition Codes or Gen Purpose Flags Are Set.

Data Addr: 0x8000,
0x3022 0x0202 0x0303 0x0404 0x0505 0x0606 0x0707 0x0808
0x0909 0x1010 0x1111 0x1212 0x1313 0x1414 0x1515 0x1616
0x1717 0x1818 0x1919 0x2020 0x2121 0x2222 0x2323 0x2424
0x2525 0x2626 0x2727 0x2828 0x2929 0x3030 0x3131 0x3232

=====
Press 'M' for menu, or press any valid menu key. >>

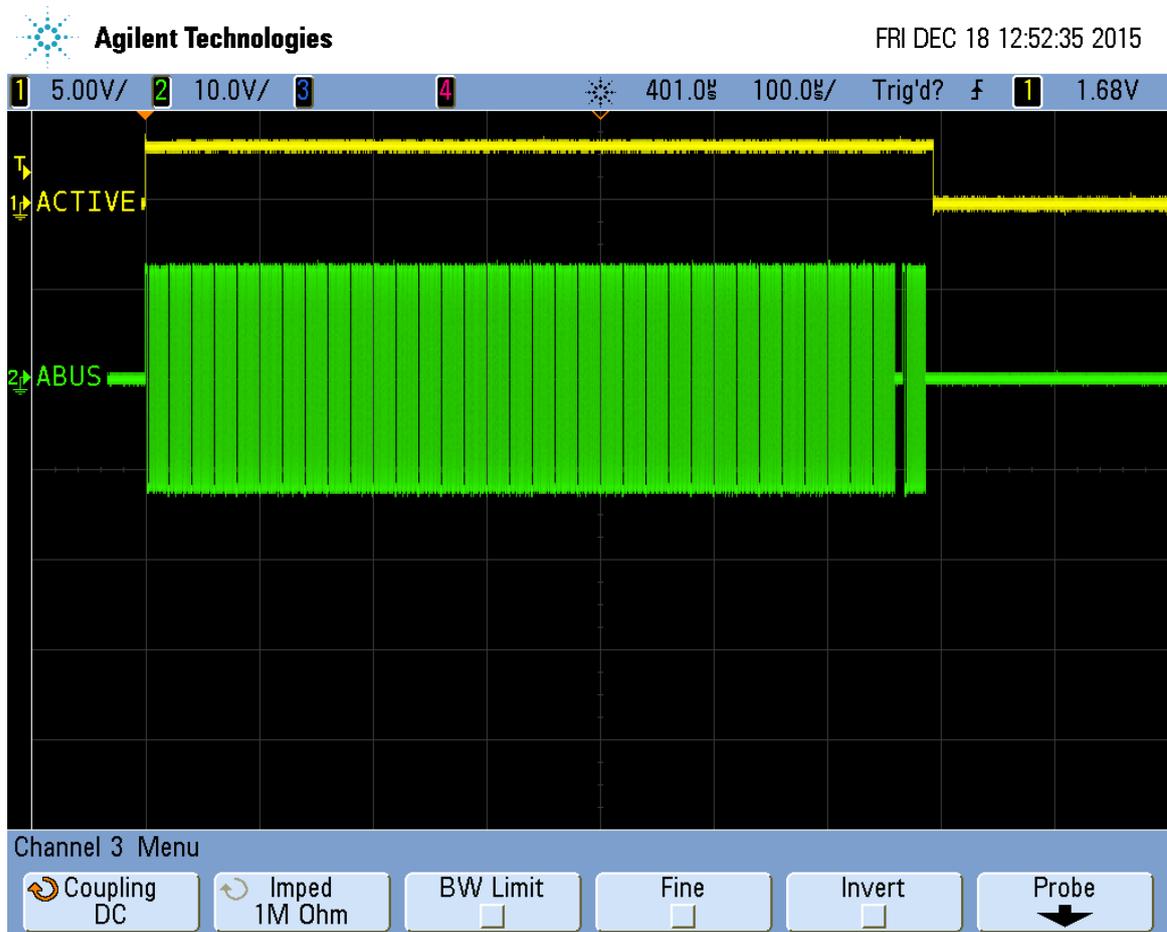
MSG #0001.  TIME = 00000000us   Bus A
            CMD 0x1BC0 --> 03-R-30-00   BC to RT
            MIW 0x0000
            DATA 3022 0202 0303 0404 0505 0606 0707 0808
                   0909 1010 1111 1212 1313 1414 1515 1616
                   1717 1818 1919 2020 2121 2222 2323 2424
                   2525 2626 2727 2828 2929 3030 3131 3232
            □

```

「T」コマンドを使用すると、RT メッセージ受信を表示するのに便利で、「1」キーを押して内部 BC を使用してメッセージを送信するとうまく動作します。「T」コマンドを使用してコンソール・トラフィックを有効にすると、メッセージ検出、保存、表示のときに、バス A またはバス B の緑色 LED が点滅します。

MSG #0001 は、オシロスコープ波形で以下のように表示されます。スコープの設定については、上記の手順 5 を参照してください。最後のワードの前に小さなギャップがあることに注意してください。これは BC に返信される RT ステータス・ワードです。ターミネータはバス・コネクタ上で使用されるため、BC が送信するときの電圧レベルは、RT が応答を送信するときと同じです。外部 BC テスタを使用して MIL-STD-1553 バス・カプラを介してコマンドを送信すると、BC 信号と RT 信号の振幅は異なります。BC 信号の振幅（ボードから発生した）は、ローカル RT ボード（スコープの接続されてい

る場所)の振幅の約 25%になります。これは、エンドツーエンドのインピーダンス変換と、バス・カプラの電流制限抵抗を介した減衰のためです。



BC 内部メッセージの一部は、それらが BBUS で処理されるため、ABUS には表示されません。これらのメッセージを表示するためには、スコープ・プローブをバス B に接続します。

デモ・ボードにメッセージを送信するために外部 BC を使用します。

BC を使用して HI-2130 ADK ボード上の RT または SMT へ適切に終端された外部バスを介して 1553 メッセージを開始する場合、外部 75Ω 抵抗（存在する場合）を取り外し、MIL-STD-1553 ケーブルとバス・カプラを使用して BC 試験装置をデモ・ボードのトライアキシャル・バス・ジャックに接続します。バス・カプラをすぐに使用できない場合は、BC テスタをデモ・ボードのトライアキシャル・ジャックに直接接続しますが、この場合は 75Ω の終端抵抗を使用してください。

RT アドレス 3、Rx メッセージ、SA1 および任意の数のデータワードに設定された BC テストメッセージを作成します。

デモ・ボードにメッセージを送信すると（コンソールのトラフィックを有効にするために、「T」コマンドが使用されていることを確認してください）、コンソールに次のようなメッセージが表示されます：

```
MSG #0001.  TIME = 00000000us  Bus A
             CMD 0x1820 --> 03-R-01-00  BC to RT
             MIW 0x0000
             DATA 0003  0201  0202  0203  0204  0205  0206  0207
                   0208  0209  020A  020B  020C  020D  020E  020F
                   0210  0211  0212  0213  0214  0215  0216  0217
                   0218  0219  021A  021B  021C  021D  021E  021F
```

外部 BC がモード・コード 19 を送信するときのコンソール表示は次のとおりです：

```
MSG #0003.  TIME = 00124328us  Bus A
             CMD 0x1C13 --> 03-T-00-19  RT to BC
             MIW 0x0013
             Mode Code 19
             Mode Data 0xABCD
```

SA30 は、通常 1553 データ・ラップアラウンド（ループバック）用に予約されています。これは SA30 Tx メッセージおよび、SA30 Rx メッセージのデータ・ポインタを同じメモリ・アドレスに設定することによって達成されます。SA30 受信コマンドが処理されると、RT は受信データを割り当てられたバッファ領域に格納します。その RT が後で SA30 送信コマンドを受信すると、BC から前に受信した同じデータでステータスを送信することによって応答します。したがって、受信され送信された SA30 データは一致していなければなりません。

```
MSG #0005.  TIME = 00124328us  Bus A
             CMD 0x1BC0 --> 03-R-30-00  BC to RT
             MIW 0x0013
             DATA 3001  1001  2002  3003  4004  5005  6006  7007
                   8008  9009  A00A  B00B  C00C  D00D  E00E  F00F
                   0010  1011  2012  3013  4014  5015  6016  7017
                   8018  9019  A01A  B01B  C01C  D01D  E01E  F01F

MSG #0006.  TIME = 00124328us  Bus A
             CMD 0x1FC0 --> 03-T-30-00  RT to BC
             MIW 0x0013
             DATA 3001  1001  2002  3003  4004  5005  6006  7007
                   8008  9009  A00A  B00B  C00C  D00D  E00E  F00F
                   0010  1011  2012  3013  4014  5015  6016  7017
                   8018  9019  A01A  B01B  C01C  D01D  E01E  F01F
```

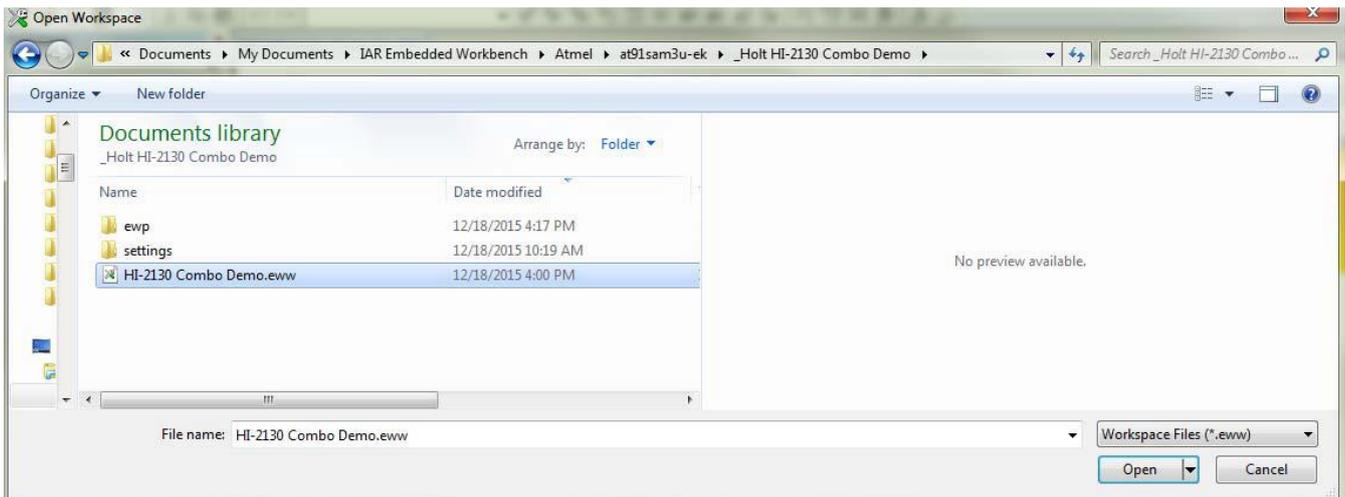
早いリポート周期で 1553 メッセージを処理する場合、コンソールによって表示されたデコードされた RT トラフィックは、コンソールの 115,200bps ボーレート制限のためペースを維持できない可能性があります

あります。メッセージ・サイズとリピート周期によっては、コンソールにメッセージが表示されないことがあります。HI-2130 RT（または SMT モニタ）がすべての有効なメッセージ・データを適切に処理（または記録）していることを確認してください。一部のメッセージはデコードされず、コンソールに表示されます。

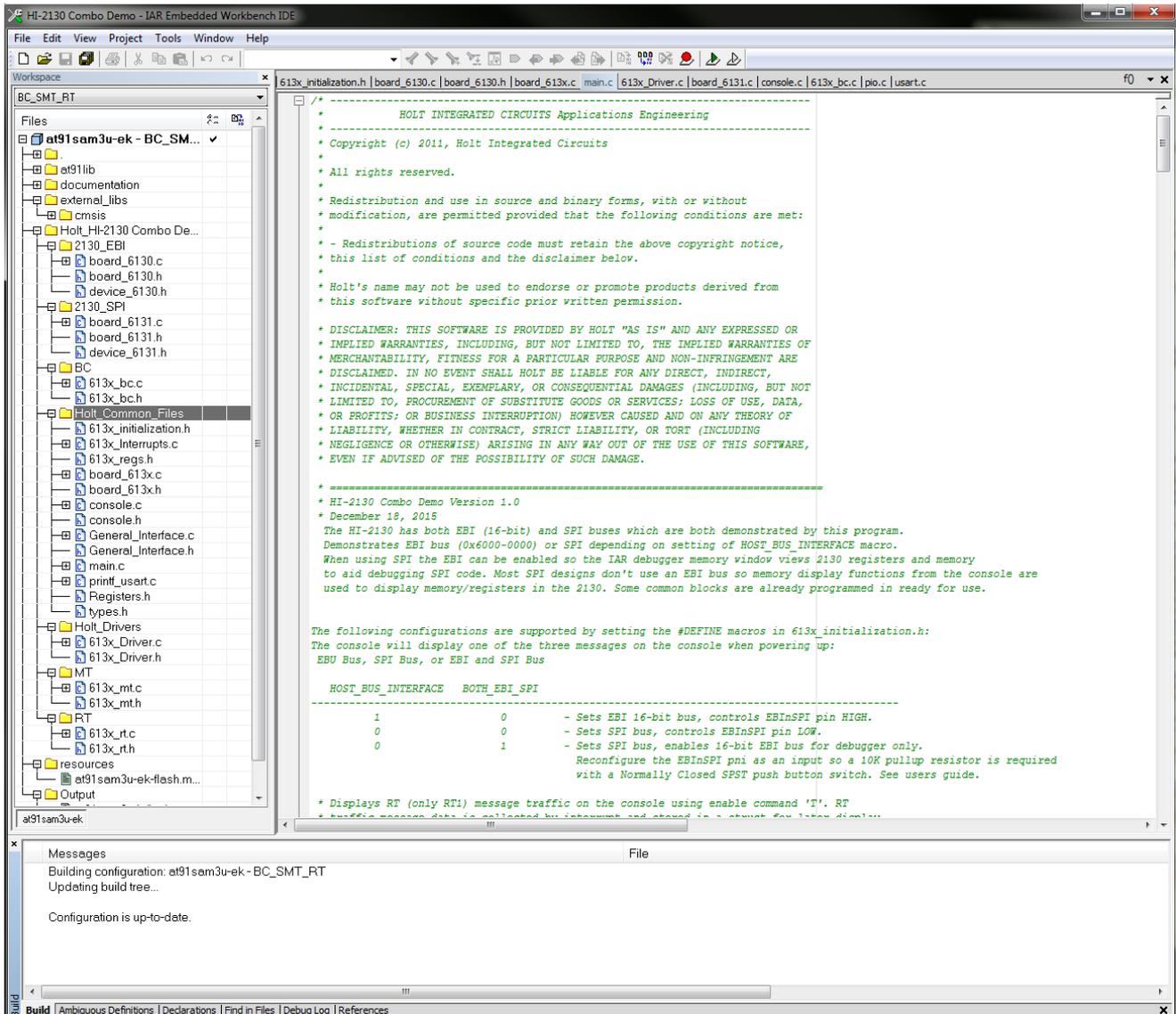
これで、デモは完了です。次の項目に進んで、ARM 用 IAR Systems Embedded Workbench[®]をインストールする方法と、Holt デモンストレーション・ソフトウェアをインストールする方法について説明します。これによりソフトウェアの変更が可能になります。

Installing IAR Systems Embedded Workbench for ARM Compiler and Getting Started with the Holt API demo software project [ARM コンパイラ用 IAR Systems Embedded Workbench のインストールおよび Holt API デモ・ソフトウェア・プロジェクトの開始]

1. ご使用のコンピュータに Holt のデモ・プロジェクトをコピーする前に ARM 用 IAR Systems のコンパイラ、Workbench (EWARM) をインストールする必要があります。インストール・シーケンスは、Atmel の ARM ライブラリと Holt のデモ・プロジェクト・フォルダが、適切な場所に作成されていることを確認することが重要です。Holt の CD-ROM 上のプロジェクト・フォルダにある Holt HI-2130 コンボ・デモ・プロジェクトのインストール・ガイドに従ってください。次の手順に進む前に、IAR EWARM をインストールし、Holt プロジェクト・フォルダを適切な位置に配置しなければなりません。これ以降の説明は、インストール作業が完了していることを前提としています。
2. 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ます。IAR ファイルのプルダウン・メニューから Holt HI-2130 コンボ・デモ・プロジェクトを開きます。File -> Open -> Workspace をクリックし、プロジェクト・フォルダの場所に移動し、「HI-2130 Combo Demo.eww」を選択し「Open」ボタンをクリックします。



- 以下に示すようにワークスペース・ウィンドウが左側に表示されます。ワークスペース・ディレクトリ・ペインが見つからない場合は、「View」プルダウン・メニューから「Workspace」を選択します。ウィンドウの調整を行うまたは、所望のインクルード・ファイルを表示するには、フォルダのグループのいずれかを開きます。
- 左側のワークスペース階層パネルで **main.c** をダブル・クリックします。ソース・ファイルが次のように EWARM テキスト・エディタ・ペインで開きます。



ファイル・ディレクトリ・ツリー上にあるプルダウン・メニューで選択された事前定義済みの 1553 ターミナルの選択があります：デフォルトでは BC_RT_MT が表示されます。構成によっては、コンパイラ／リンカ・ビルドからファイル・グループおよびファイルを除外します。例えば、「BC のみ」の設定が選択されている場合、必要のないフォルダやファイルは、次のビルドのこれらを使用しないことを表すために、グレーアウトされています。更新または、新しいコンフィグレーションを作成すると、プロジェクトのプルダウンから選択可能です。プロジェクトのコンフィグレーションの詳細については、Help メニューから IAR IDE プロジェクトのドキュメントを参照してください。デフォルトは以下の通りです。



5. デバッグには IAR Embedded Workbench[®]を実行しているコンピュータと HI-2130 アプリケーション開発キットとの間のインターフェイスを必要とします。DEBUG とマークされた HI-2130 評価ボードの USB コネクタに付属の USB ケーブルの小さい方を接続します。もう一方の USB コネクタをコンピュータの USB ポートに接続します。ARM 用 IAR C-SPY デバッガには、多数のターゲット・システム・インターフェイス用のビルドイン「J-link オンボード」のドライバが、が含まれています。デバッグ・ケーブルが PC に接続されるまで、デバッガ LED2 が繰り返し点滅しなければなりません。LED2 が点滅しない場合、プロジェクト・フォルダに含まれる、Holt のテクニカル・ノート、IAR EWARM デバッガ・トラブル・シューティング・ガイドを参照してください。

評価ボードの USB ケーブルが最初に接続されたとき、Windows は J-Link デバイスのために「新しいハードウェアが検出されました」のメッセージを表示します。数秒後、Windows が適切なドライバをロードし、「使用できる準備ができました」のメッセージが表示されます。Windows が J-Link ドライバを見つけることができなかった場合、IAR Embedded Workbench[®]インストール CD の Driver ディレクトリからインストールします。

手順 11 でデバッグ・セッションを開始するときに困難が生じた場合、Project→Options をクリックします。ウィンドウが表示され、Category = Debugger で J-Link/J-Trace をハイライトにします。Connection タブを選択し、Communications = USB and Interface = SWD をクリックします。

6. 必要に応じて、変数の最上位ビット判定で警告を発生する、コンパイラ・メッセージをオフにします。メッセージは次のようになります。すべての警告を表示したいユーザーの場合は、何も設定する必要はありません。Atmel のボード・ファイルのいくつかはこれらの警告を生成します、メッセージは次のようになります：

Remark[Pe068]: integer conversion resulted in a change of sign

To disable this diagnostic message, click Project then click Options

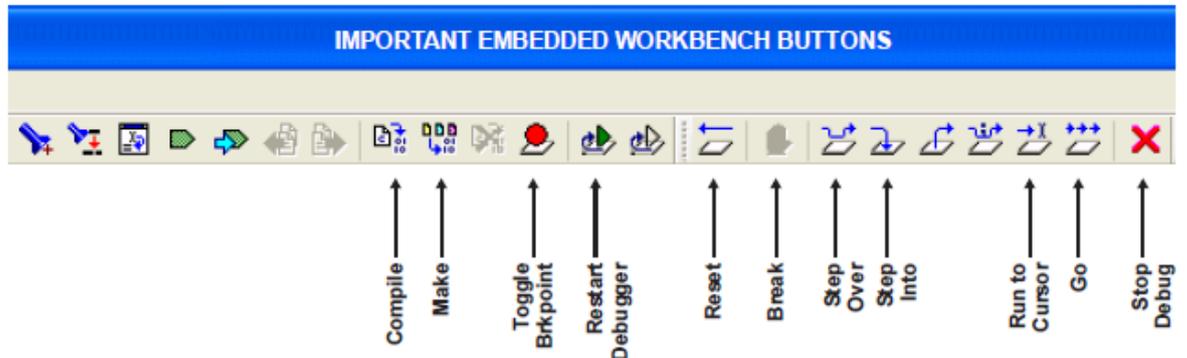
Category = C/C++ Compiler

Tab = Diagnostics

Suppress these diagnostics: add "Pe068" to list

7. デフォルトのコンフィグレーション BC_SMT_RT では、プライマリ・モード BC、MT、2×RT が有効になっています。これはデモ・ボードにプログラムされているデフォルトの設定です。これは、BC、RT1、RT2、MT が有効で、デモンストレーションされます。これらの設定は全てフラッシュ・ベースのプロジェクトです。RAM ベースのプロジェクトは MCU 上の RAM 制限によりサポートされていません。設計上、ARM Cortex-M3 は Flash よりも RAM の方が遅く実行されるため、RAM ベースのプロジェクトはほとんど必要ありません。

8. **Make (または Rebuild All)** ボタンをクリックして、プロジェクトをコンパイルします。イラストを参照してください。IAR Embedded Workbench®ビルド・メッセージ下のペインに、エラーや警告が無い場合続行することができます。エラーが発生した場合は、それらを訂正しプログラムを再コンパイルします。奇妙なコンパイラやリンク問題が発生した場合、プルダウン・ウィンドウの「**Clean**」ビルドを実行する必要があります。「**Restart Debugger**」ボタンをクリックしてデバッグ・セッションを開始します。
9. これは、MCUにコンパイルされたプログラムをダウンロードし、プログラムを実行できるように設定します。「**Go**」をクリックし実行を開始します。実行を停止する（通常は赤い手のマークで表示されます）には、「**Break**」をクリックします。



実行が開始されると、コンソールにプログラム・メニューが表示され、クイック・スタートの項目に記載されている LED が順番に点滅します。

10. バスの動作を観察するには、オシロスコープを使用し赤い BUS A テスト・ポイントと赤い BUS B テスト・ポイントに接続します。ACTIVE と表示されたテスト・ポイントは、便利なスコープ・トリガです。
11. 16Bit 平行・インターフェイス (EBI モード) を使用する場合、複数のメモリ・ウィンドウを含む複数のデバッガ・ウィンドウを使用することができます。HI-2130 はメモリマップ・デバイスなので、HI-2130 のすべてのシステム・レジスタ空間とコントロール・ブロック領域を表示することができ、デバッグ・コードを簡単に作成できます。これは SPI では機能しないので、代わりに SPI コンソール・コマンドを使用してメモリを表示してください。以下はブレークポイントを設定して実行した例です。
12. 新しい構成で、自動初期化シリアル EEPROM をプログラムするには :
- プログラムの実行を開始する前に、EEPROM からの自己初期化を使用せずに、HI-2130 を初期化するように MCU に指示する AUTOEN ラベルの DIP スイッチを OFF にしてください。COPYREQ ラベルの DIP スイッチを ON にして、HI-2130 レジスタのポストリセット初期化と RAM の完了後に MCU に EEPROM コピー・シーケンスを開始するように指示します。

Rx SA1~SA30 コントロール・ブロック（各 4 ワード）は、アドレス・ブロック 0x60000808~0x600008F0 に跨ります。特定の SA コントロール・ブロックの IDE バイト・アドレスを計算するには、RT1 の特定のベース・アドレス（Rx は 0x0800、Tx は 0x0900）に上位オフセット・アドレス 0x6000 を追加します。SA の値に 8 を掛け、それをベース・アドレスに加えてください。

$$2130 \text{ オフセット、RT1 Rx ベース・アドレス、SA1} \\ 0x6000 - 0800 + 1 \times 8 = 0x6000 - 0808$$

例として、RT1、RX SA1 のバイト・アドレスが下の 1 行目に示されています。

2130 ベース・アドレス	コントロール・ブロック・アドレス (for RT1 RX)	SA	IDE フル・アドレス
0x6000 - 0000	0x0800	1	0x60000808
0x6000 - 0000	0x0800	2	0x60000810
0x6000 - 0000	0x0800	30	0x600008F0

2130 ベース・アドレス	コントロール・ブロック・アドレス (for RT1 TX)	SA	IDE フル・アドレス
0x6000 - 0000	0x0900	1	0x60000908
0x6000 - 0000	0x0900	2	0x60000910
0x6000 - 0000	0x0900	30	0x600009F0

これらの RT コントロール・ブロックのデータ・バッファ・アドレスを計算するには、4 ワード・コントロール・ブロック内のワード 2 またはワード 3（コントロール・ワード・タイプに応じて）を調べます。例えば、RT1 Rx SA30 コントロール・ブロックには、アドレス 0x600008F0 に次のワードが含まれています：

RT1 RX SA30 = 0x600008F0

デスクリプタ・ワード 1 コントロール・ワード	デスクリプタ・ワード 2 データ・ポインタ A	デスクリプタ・ワード 3 データ・ポインタ B	デスクリプタ・ワード 4 ブロードキャスト・ データ・ポインタ
0xC000	0x8AE	0x0000	0x08AE

コントロール・ワードは、データ・ポインタ・アドレスが 0x08AE に設定されたシングル・メッセージ・モードとして定義されます。シングル・メッセージ・モードでは、ワード 2 はデータ・ポインタであり、ワード 3 は 0x0000 に設定されています。コントロール・ワードのタイプと、バッファ・スキームの詳細については、データシートの「19.4 Descriptor Table」の表を参照してください。このコントロール・ワードで Bit14 (IWA) が High に設定されているため、このメッセージが RT によって処理されるとメッセージ割り込みが発生します（データシートの 19.4.1 を参照）。C マクロ「INT」を 0（デフォルト）に設定してデモ・プログラムをコンパイルすると、RT メッセージ・データはポーリング *RT_MesgRead()* によって取得されます。「INT」を 1 に設定すると、割り込み関数で *RT_MesgRead()* が呼び出されます。どちらの場合も、メッセージ・データを C 構造体に格納するために、対応するメッセージ・コントロール・ワード内の DBAC Bit が検査され、現在のメッセージ（受信したばかりのメッセージ）に対してデスクリプタ・ブ

ックがアクセスされたかどうかを判別します。プログラムがメッセージ割り込み (INT=1) に設定されると、デスクリプタ・テーブルの RT コントロール・ワードに IWA Bit14 がセットされた RT メッセージのみがメッセージ割り込みを生成し、T コマンドを使用してコンソールに表示されます。これらのワードは、配列 `descr_table_RT1[512]` を使用して、関数 `initialize_613x_RT1()` によってデバイス・デスクリプタ・テーブルで初期化されます。

データ・ポインタ・アドレスを計算するには、複数の $0x8AE \times 2 = 0x115C$ 。 $0x115C$ の最初のワードは MIW (メッセージ情報ワード) で、次にタイムタグ (TT) ワードの後に、アドレス $0x1160$ で始まる最大 32 ワードが続きます。これらのアドレスは 16Bit のワード・アドレスであるため、完全なアドレスを得るにはこれらのアドレス ($0x60001160$) に $0x6000$ オフセットを追加します。BC、RT、RT2 および SMT の 2130 コントロール・ブロック・アドレスの詳細については、HI-6130 データシートを参照してください。コントロール・ブロックとデータ・バッファ領域の開始アドレスを示す 6130 (2130) メモリ・マップが 22 ページに用意されていますが、データシートにはすべてのターミナルのすべてのコントロール・ブロックとデータ構造アドレスに関する詳細情報が提供されています。

SPI Interface [SPI インターフェイス]

SPI モードでは、IAR デバッガはデバイス・メモリ (レジスタ/コントロール・ブロック/バッファ・データ) を表示できないため、特別な事前設定コマンドを使用します: スペース、Z、X、C、V、B、I を押して、対応するコントロール・ブロックのデータ・バッファをコンソールに表示します。ユーザーが任意のアドレスにデータを表示するためのカスタム・コマンドを追加することができます。これらのコマンドは、データをワード・アドレスとして表示します。これらのコマンドの例については、`console.c` を参照してください。

```
SPI 2130 reg/memory Display Commands:
Press Space Bar Display Registers 0x-0x0050
Press 'A' Display Memory at address
Press 'Z' Display RT control block
Press 'X' Display RT data block
Press 'C' Display RT2 control block
Press 'U' Display RT2 data block
Press 'B' Display BC mesg and data block
Press 'I' Display Interrupt Log
```

```
2130 Registers:
  0 1 2 3 4 5 6 7 8 9 A B C D E F
Adr 0000: 19F0 8100 0000 0000 0000 0000 0000 0000 0000 0180 527A 0000 000D 0000 7818
Adr 0010: FFF8 0000 D458 0058 FBF8 0000 D5A8 00EF 1E00 0400 0000 0000 AAAA BBBB 0000 ABCD
Adr 0020: 01CE 2200 0600 0000 0000 AAAA BBBB 0000 ABCD 0800 0000 0000 0000 0000 0000
Adr 0030: 0000 0000 D02D 1B70 1B70 0000 0000 8000 00C0 000F 0000 0000 0000 0000 0000
Adr 0040: 0000 0000 0000 C092 004B 0000 0000 0000 0000 C0BF 0000 C0CD 0000 0000 0000 0000
Adr 0050: 0000
>> □
```

Key Project Files with Selected Descriptions [キー・プロジェクト・ファイル選択項説明]

HEADER FILES WITHOUT CORRESPONDING C FILES

613x_initialization.h

重要な構成設定の定義

HOST_BUS_INTERFACE

1 : プログラム・コンパイル設定を EBI 平行バスにします

0 : プログラム・コンパイル設定を SPI バスにします

BOTH_EBI_SPI (only valid with HOST_BUS_INTERFACE = 0)

0 : 通常

1 : HOST_BUS_INTERFACE = 0 の場合、EBI デバッガのメモリ・ウィンドウの使用を許可

device_6130.h

HI-2130 をアドレス指定する ARM MCU の外部バス・インターフェイス定義および構造体

device_6131.h

HI-2130 をアドレス指定する ARM MCU の外部 SPI インターフェイス定義および構造体

C FILES WITH CORRESPONDING HEADER FILES

main.c

プライマリ・プログラム・エントリ・ポータルである、**main()** は、初期化シーケンスを示し、ターミナル・プログラムを使用して PC キーボードから入力されたコンソール・コマンドを待つ無限ループに入ります。

board_6130/31.c

この関数は、HI-6130 と 256K 外付け SRAM の ARM MCU 外部バス・インターフェイスを初期化します。

console.c

この関数は、コンソールキーの押下を検出し、対応するデモ・コマンドを実行します。

613x_bc.c

このモジュールには、BC 動作の初期化および、サポートするための関数が含まれています。BCENA DIP スイッチを使用する場合は、High (ON) に設定する必要があります。

613x_mt.c

このモジュールには、SMT 動作を初期化しサポートするための関数が含まれています。使用する場合、MTRUN DIP スイッチを High (ON) に設定する必要があります。

613x_rt.c

このモジュールには、RT (AKA RT1) および、RT2 動作の初期化およびサポートするための関数が含まれています。RT1 および RT2 ENA DIP スイッチを使用する場合は High (ON) に設定する必要があります。

613x_Interrupts.c

主目的は、RT メッセージの割り込みを初期化して、処理して、RT メッセージをキャプチャすることです。

613x_Driver.c

RT メッセージ・データをキャプチャする機能（割り込みまたは、ポーリングによる）を含み、データを C メッセージ構造体に格納します。コンソール「T」コマンドが使用されると、メッセージがメイン・ループで検出されると、フラグが設定され、RT メッセージ・データがコンソールに表示されます。

```
MSG #0001.  TIME = 00000000us    Bus A
             CMD 0x1820 --> 03-R-01-00    BC to RT
             MIW 0x0000
             DATA 0003  0201  0202  0203  0204  0205  0206  0207
                   0208  0209  020A  020B  020C  020D  020E  020F
                   0210  0211  0212  0213  0214  0215  0216  0217
                   0218  0219  021A  021B  021C  021D  021E  021F
```

Application Development Kit Notes [アプリケーション開発キット注記]

HI-2130 データシートはキットの CD-ROM に含まれています。参照として、HI-6130 データシートも含まれています。最新のドキュメントは www.holt.com にあります。

主要なプロジェクト・コンフィグレーション設定は、**613x_initialization.h** ファイルに記載されています。

HI-2130 は、外部パラレル 16Bit (EBI) または、SPI インターフェイスを持つマイクロコントローラに適合するように設計されています。RAM およびレジスタ EBI を使用した場合、RAM およびレジスタの場所は、MCU ボード上の ARM Coretex M3 マイクロプロセッサのメモリ・アドレス空間に表示されます。MCU チップ・セレクト出力 (HI-2130 チップ・イネーブル入力端子に接続されている) は、MCU バス・アドレス 0x60000000 から始まる領域にアクセスします。バイト・アドレッシングが使用されます。RAM およびレジスタの動作は 16Bit 値を取引しますので、全てのアクセス・アドレスは偶数です。バイト・アドレスを使用するには、HI-2130 RAM またはレジスタ・アドレスは 2 倍になり、MCU チップセレクト・ベース・アドレスに加算されます。したがって、HI-2130 レジスタ 0 は、MCU バス・アドレス 0x60000000 にアクセスされます。レジスタ 1 はバス・アドレス 0x60000002、レジスタ 5 はアドレス 0x6000000A でアクセスできます。

評価ボードのプログラムは、MIL-STD-1553 ターミナル・モードのそれぞれ (BC、MT、RT) を使用し、固定アドレスと再配置可能な RAM 構造のための包括的なバス・アドレス指定の例を示します。

デバッガを使用する場合、プログラムの実行がストップするたびにごとに更新するメモリ・ウィンドウは、レジスタや RAM 値を観測するのに役に立ちます。実行停止時にそれぞれの位置の表示が再スキャンされていることに留意してください。一部のレジスタや RAM 構造のビットは、読込発生後リセットされます。これは、ペンディング割り込みレジスタのビットと、RAM 内の RT ディスクリプタ・テーブル・コントロール・ワードのための DBAC データ・ブロック・アドレス・ブロックが含まれています。これらの場合は、メモリ・ウォッチ・ウィンドウでは、実行が停止したことで値を反映している。

デバッガを使用する場合、プログラムの実行がストップするたびにごとに更新する、ウォッチ・ウィンドウでは、MIL-STD-1553 モードのための様々な定義されている RAM 構造に含まれる値を観測するのに役立ちます。デバッグするときは、IAR Embedded Workbench®では 4 つのウォッチ・ウィンドウまで可能なので、BC、RT1、RT2 などのために別々のウォッチタブを設定することができます。興味がある各構造に対して、構造体のポインタ名をハイライトするためにダブル・クリックし、ハイライトされたポインタ名をオープン・ウォッチ・ウィンドウにドラッグ&ドロップすることで、調べることができます。読込が発生

した後、いくつかの RAM 構造ビットは自動的にリセットすることに留意してください。これは、RT ディスクリプタ・テーブル・コントロール・ワードのための DBAC データ・ブロック・アドレス・ブロックが含まれています。これらの場合は、メモリ・ウォッチ・ウィンドウでは、実行が停止したことで値を反映している。

```

613x_rt.h | 613x_mt.h | board_613x.c | 613x_bc.c | main.c | 613x_bc.h | 613x
...
#if (HOST_BUS_INTERFACE)
// This is not used by SPI interface HI-6131...
// HI-6130 register structure base address
const H6130 pH6130 = HI6130_BASE;

#if(RT1_ena)
const RT1d pRT1d = RT1_DTABLE_BASE;
const RT1i pRT1i = RT1_ITABLE_BASE;
#endif // (RT1_ena)

#if(RT2_ena)
const RT2d pRT2d = RT2_DTABLE_BASE;
const RT2i pRT2i = RT2_ITABLE_BASE;
#endif // (RT2_ena)

#if(BC_ena)
GPQ pGPQ = (GPQ) GPQ_BASE_BUS_ADDR;
const BCi1 pBCi1 = BC_ILIST_BASE;
const BCstack pBCstack = BC_STACK_BASE;
const BCstack2RT pBCstack2RT = BC_2RTSTACK_BASE;
#endif // (BC_ena)

#if(SMT_ena)
DSTK pDSTK = (DSTK) DSTK_BASE_BUS_ADDR;

```

ハイライトされた変数をウォッチ・ウィンドウにドラッグ&ドロップできます。

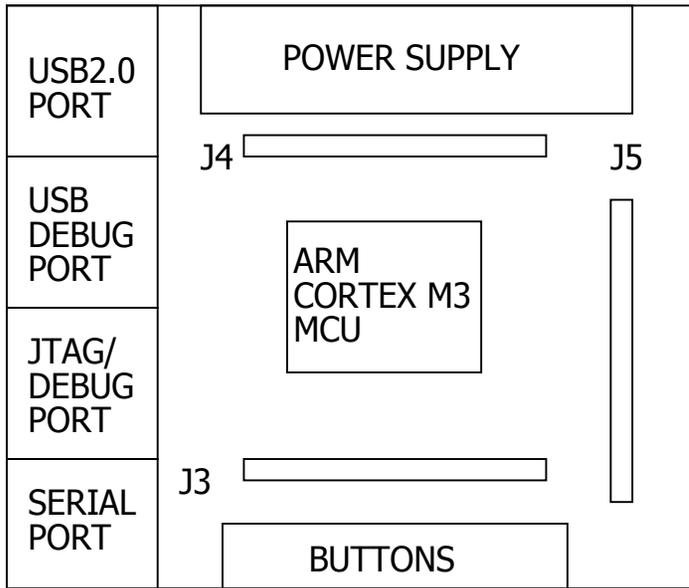
Expression	Value	Location	Type
pH6130	0x60000000	0x00089950	<32-bit Unsigned>
pRT1i	0x60000400	0x00089958	RT1i
pRT2i	0x60000600	0x00089960	RT2i
pRT1d	0x60000800	0x00089954	RT1d
pRT2d	0x60000C00	0x0008995C	RT2d
pGPQ	0x60001800	0x20000080	GPQ
pBCi1	0x600036E0	0x00089964	BCi1
pBCstack	0x60007C00	0x00089968	BCstack
pBCstack2RT	0x60007C80	0x0008996C	BCstack2RT
pMTF	0x60000200	0x00089974	MTF
pDSTK	0x6000C000	0x20000084	DSTK
pScSTK	0x6000A800	0x00089970	ScSTK

多数の構造体のポインタを使用してウォッチ・ウィンドウをデバッグします。右側は REG 構造体が拡張表示されたものです。

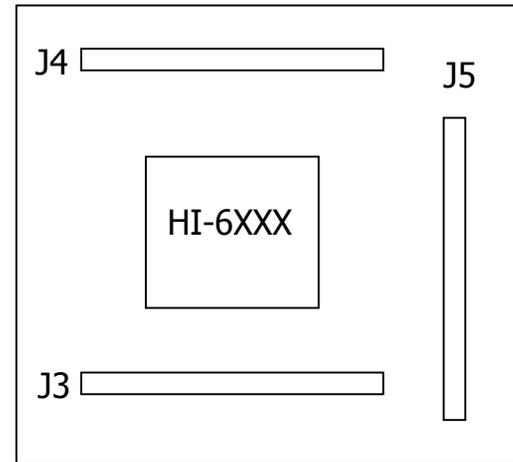
Expression	Value	Location	Type
pH6130	0x60000000	0x00089950	<32-b
MASTER_CONFIG_REG	0x11F2	0x60000000	HI613
STATUS_AND_RESET_REG	0x8100	0x60000002	HI613
RT1_CURR_CMD_REG	0x0000	0x60000004	HI613
RT1_CURR_CTRL_WORD_ADDR_REG	0x0000	0x60000006	HI613
RT2_CURR_CMD_REG	0x0000	0x60000008	HI613
RT2_CURR_CTRL_WORD_ADDR_REG	0x0000	0x6000000A	HI613
HDW_PENDING_INT_REG	0x0000	0x6000000C	HI613
BC_PENDING_INT_REG	0x0000	0x6000000E	HI613
MT_PENDING_INT_REG	0x0000	0x60000010	HI613
RT_PENDING_INT_REG	0x0000	0x60000012	HI613
INT_COUNT_AND_LOG_ADDR_REG	0x0180	0x60000014	HI613
dummy15	<array>	0x60000016	HI613
HDW_INT_ENABLE_REG	0x7818	0x6000001E	HI613
BC_INT_ENABLE_REG	0xFFF8	0x60000020	HI613
MT_INT_ENABLE_REG	0x01F8	0x60000022	HI613
RT_INT_ENABLE_REG	0xD5A8	0x60000024	HI613
HDW_INT_OUTPUT_ENABLE_REG	0x6018	0x60000026	HI613
BC_INT_OUTPUT_ENABLE_REG	0xFBF8	0x60000028	HI613
MT_INT_OUTPUT_ENABLE_REG	0x01F8	0x6000002A	HI613
RT_INT_OUTPUT_ENABLE_REG	0xD5A8	0x6000002C	HI613

評価ボードの回路図と部品表は次ページにあります。

ドキュメントの最後のメモリ・マップは RAM の割り当て方法を示しています。



LOWER CIRCUIT BOARD



STACKING UPPER CIRCUIT BOARD

J3,J4 & J5 ARE DUAL-ROW STACKING RECEPTACLES (LOWER BOARD) AND HEADERS (UPPER BOARD).

HOLT INTEGRATED CIRCUITS, Mission Viejo, CA, USA		
Title		
ARM CORTEX M3 MICROCONTROLLER BOARD		
Size	Document Number	Rev
A	CM3 BOARD REV F.DSN	F
Date:	Tuesday, April 11, 2017	Sheet 1 of 7

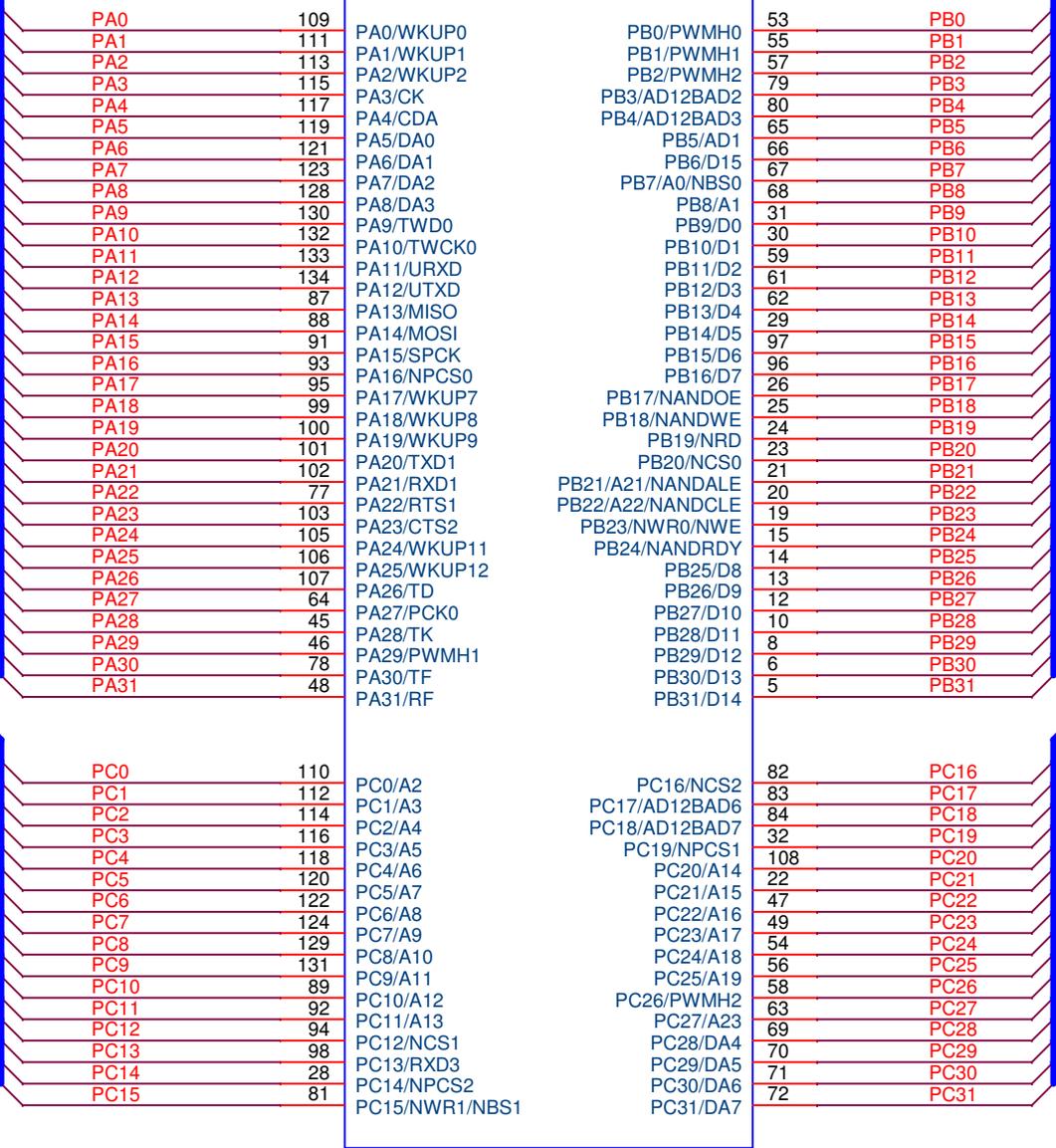
PA[31:0]

PB[31:0]

PC[31:0]

PC[31:0]

U1A
SAM3U



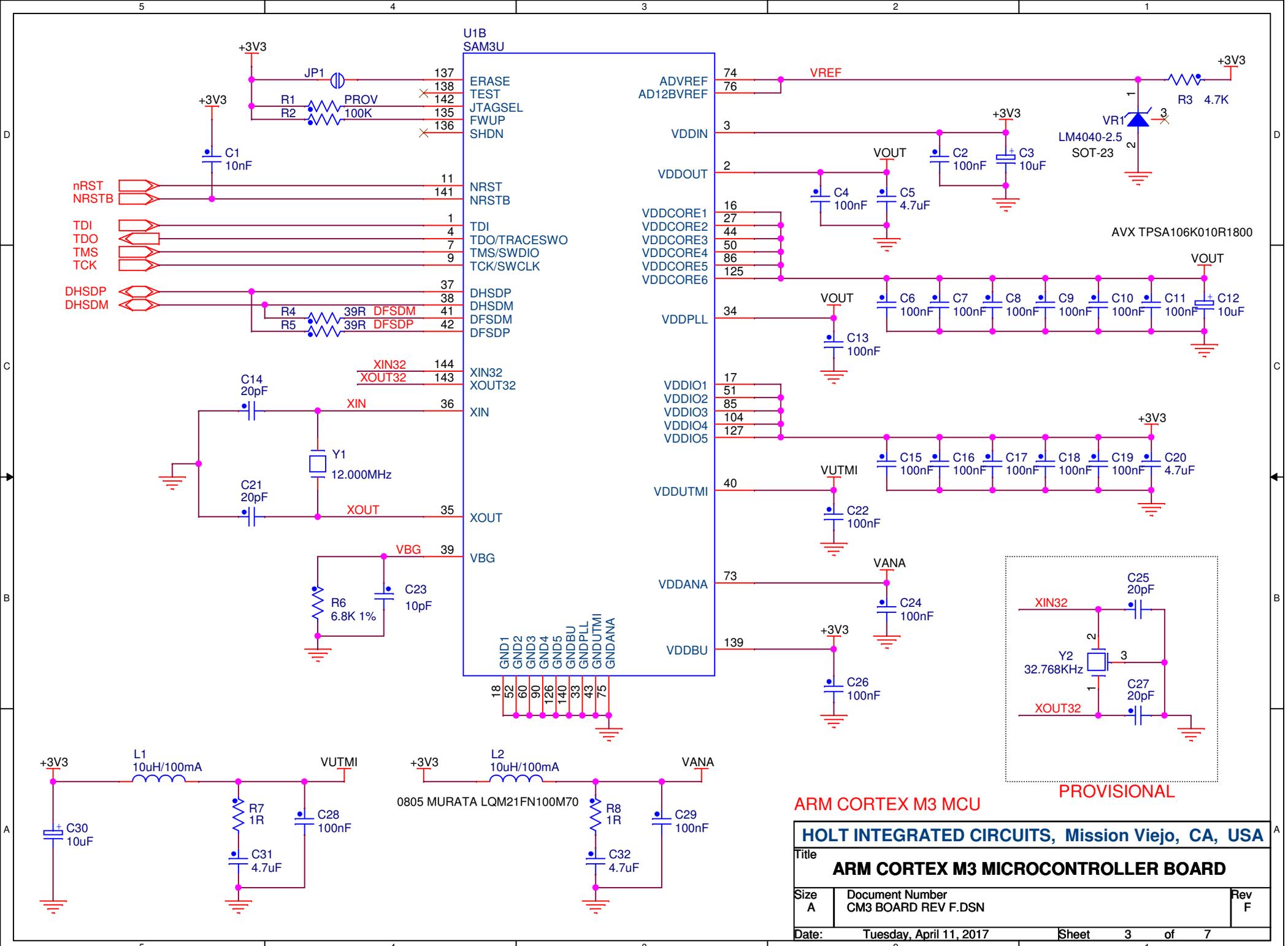
ARM CORTEX M3 PIO

HOLT INTEGRATED CIRCUITS, Mission Viejo, CA, USA

Title
ARM CORTEX M3 MICROCONTROLLER BOARD

Size A	Document Number CM3 BOARD REV F.DSN	Rev F
-----------	--	----------

Date: Tuesday, April 11, 2017 Sheet 2 of 7



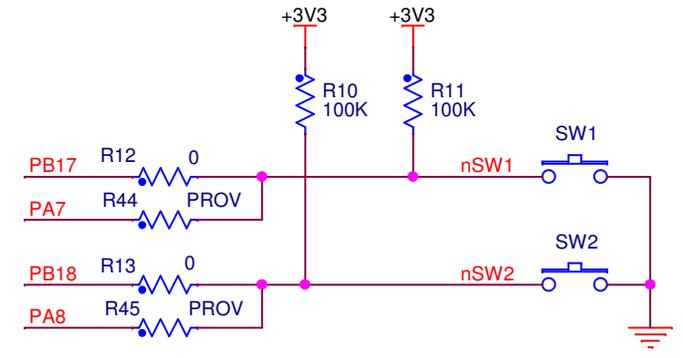
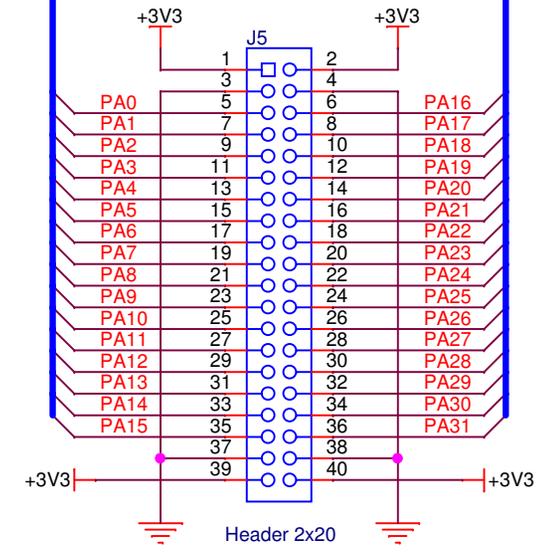
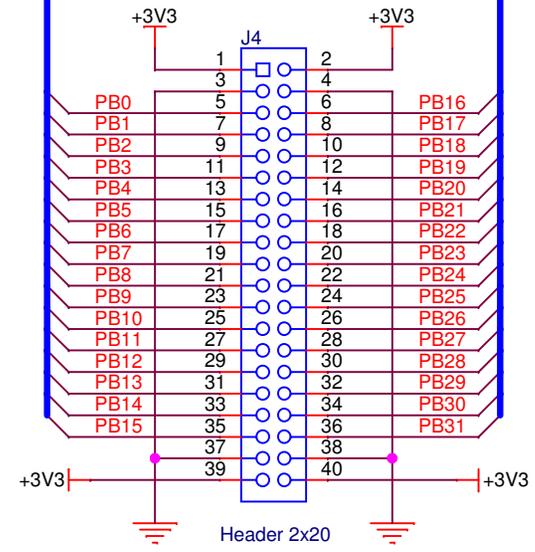
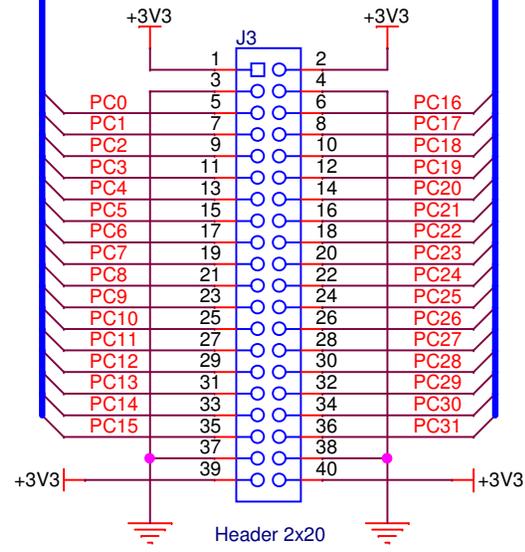
ARM CORTEX M3 MCU

PROVISIONAL

HOLT INTEGRATED CIRCUITS, Mission Viejo, CA, USA

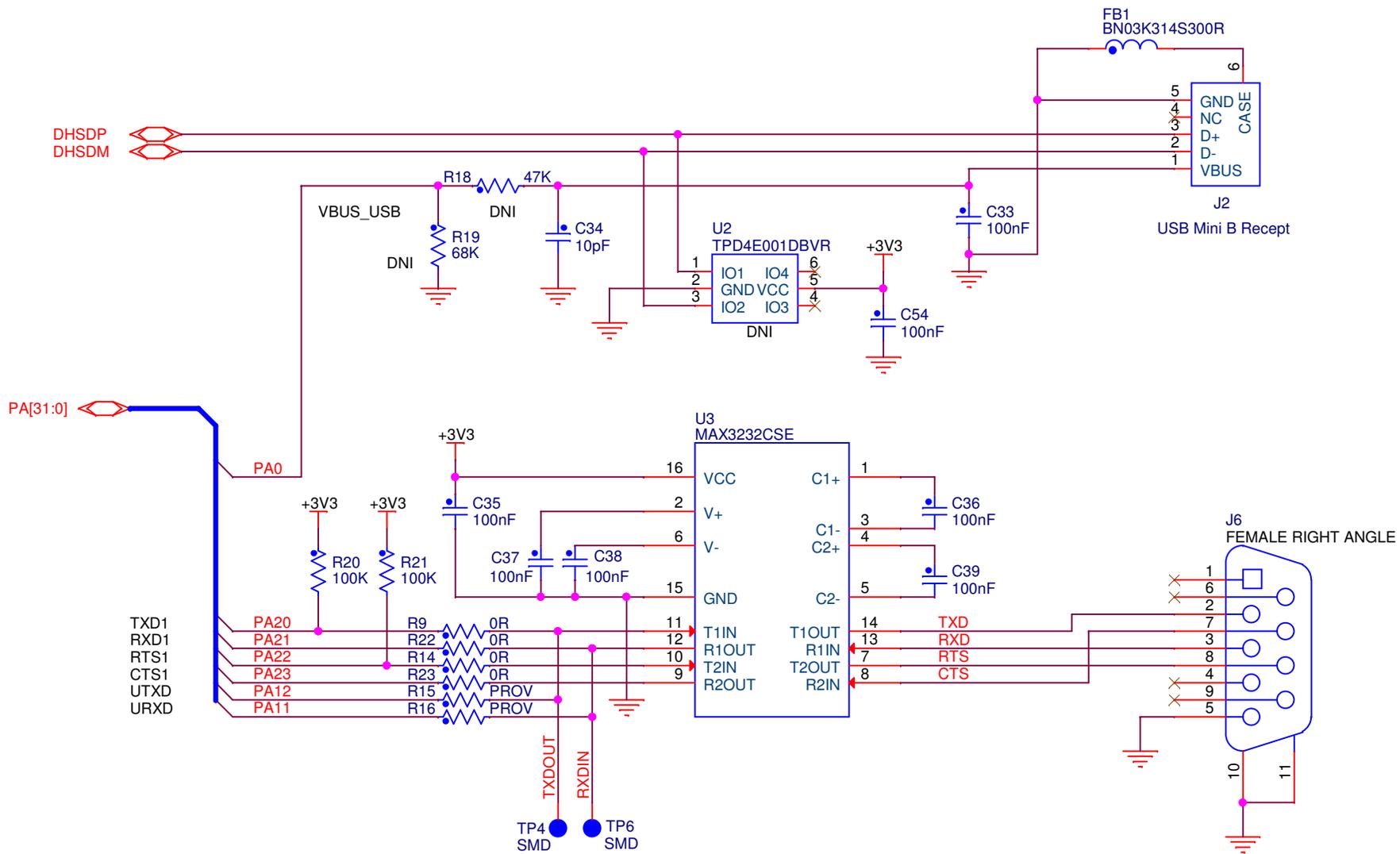
Title		
ARM CORTEX M3 MICROCONTROLLER BOARD		
Size	Document Number	Rev
A	CM3 BOARD REV F.DSN	F
Date:	Tuesday, April 11, 2017	Sheet 3 of 7

{1,4} PA[31:0]
 {1,3,5} PB[31:0]
 {1,3} PC[31:0]



BOARD I/O HEADERS, BUTTONS

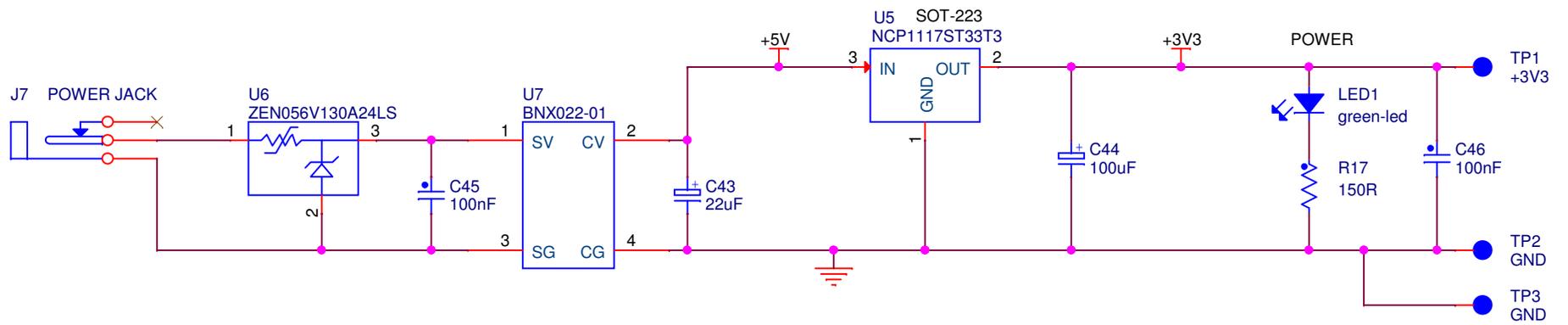
HOLT INTEGRATED CIRCUITS, Mission Viejo, CA, USA		
Title ARM CORTEX M3 MICROCONTROLLER BOARD		
Size A	Document Number CM3 BOARD REV F.DSN	Rev F
Date: Tuesday, April 11, 2017		Sheet 4 of 7



USB & RS-232 SERIAL

HOLT INTEGRATED CIRCUITS, Mission Viejo, CA, USA

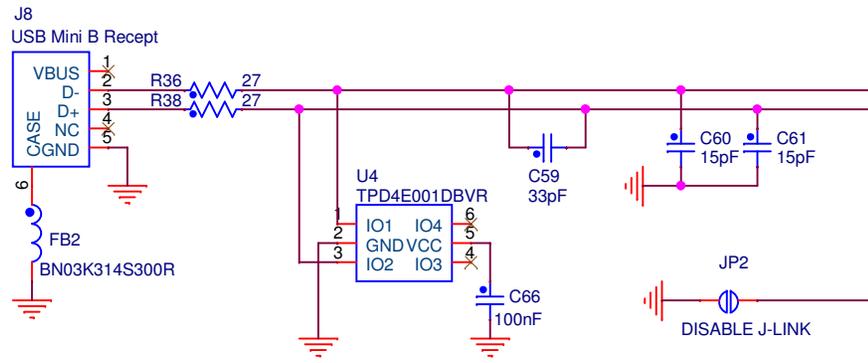
Title		
ARM CORTEX M3 MICROCONTROLLER BOARD		
Size	Document Number	Rev
A	CM3 BOARD REV F.DSN	F
Date:	Tuesday, April 11, 2017	Sheet 5 of 7



POWER SUPPLY

HOLT INTEGRATED CIRCUITS, Mission Viejo, CA, USA		
Title		
ARM CORTEX M3 MICROCONTROLLER BOARD		
Size	Document Number	Rev
A	CM3 BOARD REV F.DSN	F
Date:	Tuesday, April 11, 2017	Sheet 6 of 7

USB DEBUG INTERFACE



**SEGGER J-LINK ON-BOARD
DEBUGGER INTERFACE**

(CONFIDENTIAL)

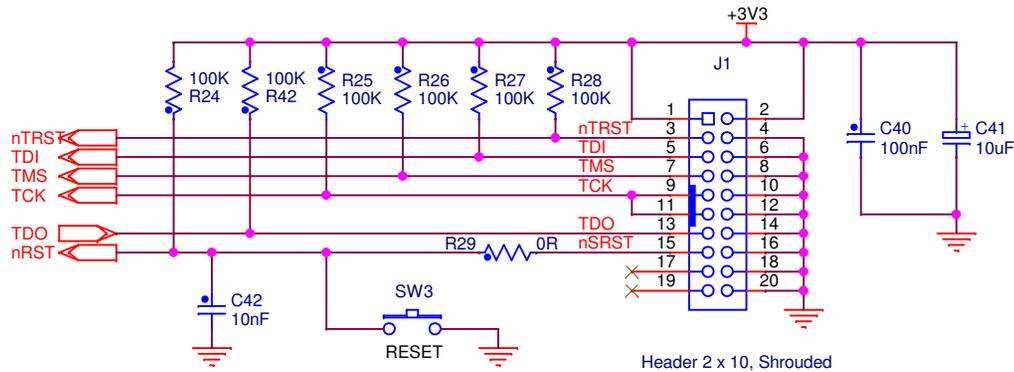
NOT PART OF A CUSTOMER DESIGN,
THIS BLOCK IS COMPRISED OF U8,
Y3, C47-C53, C55-C58, C62-C65, R30,
R32-R35, R37, R39-R41 AND R43.

- TDI
- TMS
- TCK
- TDO
- nRST

**DEBUGGER INTERFACE COPIED
FROM ATMEL ARM CORTEX M3**

USE THIS TO CONNECT J-LINK IF ABOVE
CIRCUITRY IS NOT POPULATED OR WHEN
IT IS DISABLED BY JUMPER JP2.

**PARALLEL
DEBUG INTERFACE**



HOLT INTEGRATED CIRCUITS, Mission Viejo, CA, USA

Title		
ARM CORTEX M3 MICROCONTROLLER BOARD		
Size	Document Number	Rev
	CM3 BOARD REV F.DSN	F
Date:	Tuesday, April 11, 2017	Sheet 7 of 7

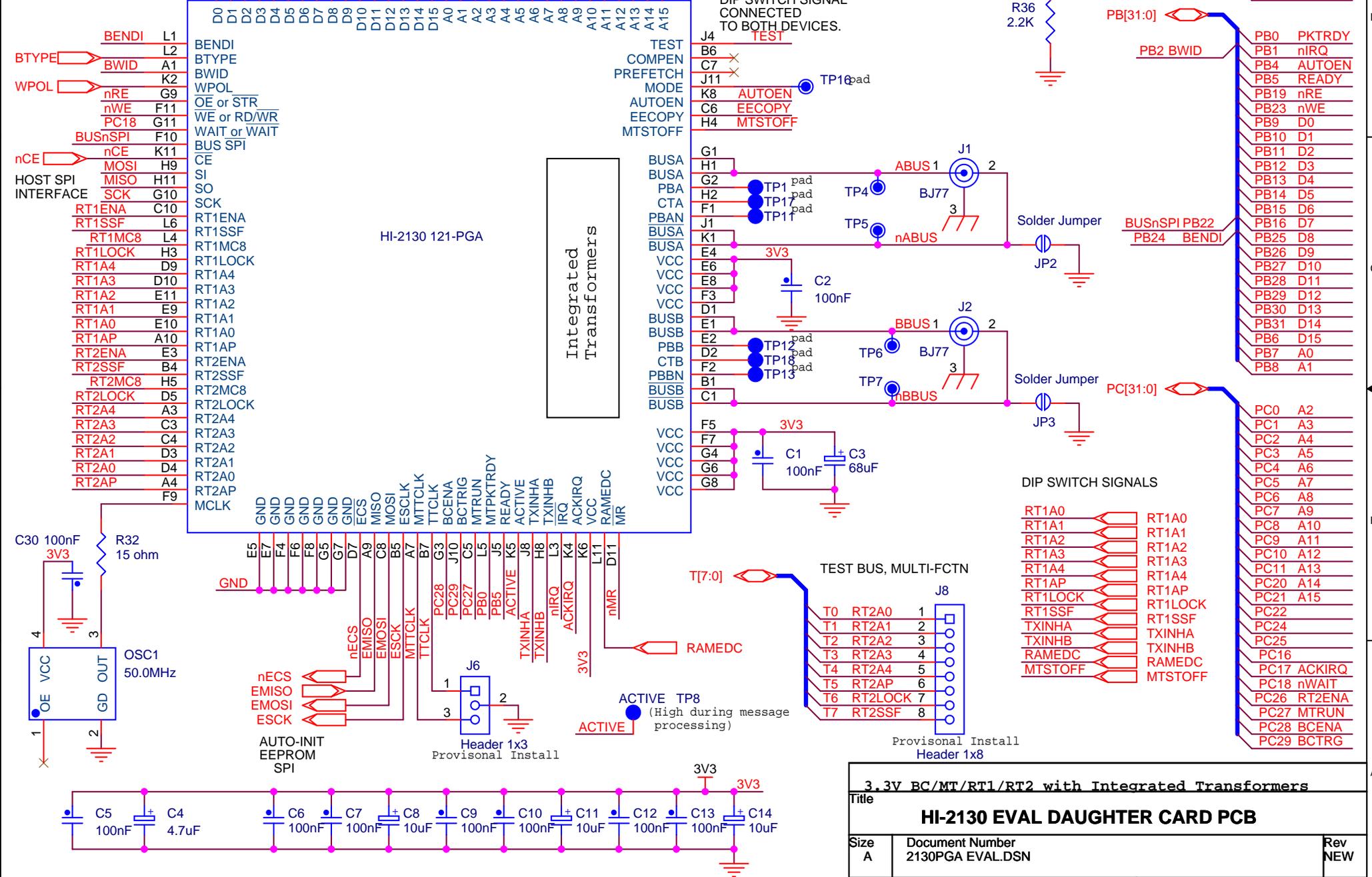
Item	Qty	Description	Reference	DigiKey	Mfr P/N
1					
2	1	PCB, Bare, Evaluation Board	N/A	-----	
3	1	Ferrite Bead, 220 Ohm @ 100MHz 300mA DC 0805	FB1	732-1602-1-ND	Wurth 742792034
4	2	Capacitor, Ceramic 10nF 10% 50V X7R 0603	C1,C42	490-1512-1-ND	Murata GRM188R71H103KA01D
5	2	Capacitor, Ceramic 10pF 10% NP0 C0G 0V 0603	C23,C34	490-1403-1-ND	Murata GRM1885C1H100JA01D
6	4	Capacitor, Ceramic 20pF 5% NP0 C0G 0V 0603	C14,C21,C25, C27	490-1410-1-ND	Murata GRM1885C1H200JA01D
7	29	Capacitor, Ceramic 100nF 10% 25V Y5V 0603	C2,C4,C6-C11,C13,C15-C19,C22, C24,C26,C28, C29, C33,C35-C40,C45-46,C54	490-1575-1-ND	Murata GRM188F51E104ZA01D
8	4	Capacitor, Tantalum 4.7uF 10% 10V Low ESR SMD 1206	C5,C20,C31, C32	478-2391-1-ND	AVX TPSA475K010R1400
9	4	Capacitor, Tantalum 10uF 10% 10V Low ESR SMD 1206	C3,C12,C30,C41	478-3317-1-ND	AVX TPSA106K010R1800
10	1	Capacitor 22uF 10% 6.3V Tantalum Low ESR SMD C	C43	399-10521-1-ND	Kemet T495C226K006ATE380
11	1	Capacitor 100uF 10% 6.3V Tantalum Low ESR SMD C	C44	495-1509-1-ND	Kemet T495C107K006ZTE150
12	1	Header, Male Shrouded 2x10, 0.1" Pitch	J1	HRP20H-ND	Assmann AWHW20G-0202-T
13	2	Connector, Receptacle USB Mini B Rt-Angle PCB Mount	J2,J8	H2959CT-ND	Hirose UX60-MB-5ST
14	1	Connector DB9F, Right-Angle PCB Short Body, Board Lock	J6	AE10924-ND	Assman A-DF-09-A/KG-T4S
15	1	Jack, DC Power, 2.5mm ID x 2.1mm pin	J7	CP-102AH-ND	Cui PJ-102AH
16	3	Receptacle, Female 2x20, 0.1" Pitch, 8.5mm Height, 3.2mm Solder Tails	J3,J4,J5	S6104-ND	Sullins PPTC202LFBN-RC
17	1	Solder Jumper	JP1	SOLDER OPEN	
18	2	Inductor, 10uH,100mA 0805	L1,L2	490-4029-1-ND	Murata LQM21FN100M70L
19	2	LED Green 0805	LED1,LED2	160-1179-1-ND	LiteOn LTST-C170GKT
20	0	Resistor, Prov 1/8W 0805	R1,R15,R16, R44,R45	DO NOT STUFF	
21	7	Resistor, 0 ohm 1/8W 0805	R9,R12,R13, R14,R22,R23, R29	P0.0ACT-ND	Panasonic ERJ-6GEY0R00V
22	2	Resistor, 1.0 5% 1/8W 0805	R7,R8	P1.0ACT-ND	Panasonic ERJ-6GEYJ1R0V
23	2	Resistor, 39 5% 1/8W 0805	R4,R5	P39ACT-ND	Panasonic ERJ-6GEYJ390V
24	1	Resistor, 150 5% 1/8W 0805	R17	P150ACT-ND	Panasonic ERJ-6GEYJ151V
25	1	Resistor, 4.7K 5% 1/8W 0805	R3	P4.7KACT-ND	Panasonic ERJ-6GEYJ472V
26	1	Resistor, 6.8K 5% 1/8W 0805	R6	P6.8KACT-ND	Panasonic ERJ-6GEYJ682V
27	0	Resistor, 47K 5% 1/8W 0805	R18	DO NOT STUFF	Panasonic ERJ-6GEYJ473V
28	0	Resistor, 68K 5% 1/8W 0805	R19	DO NOT STUFF	Panasonic ERJ-6GEYJ683V
29	11	Resistor,100K 5% 1/8W 0805	R2,R10,R11, R20,R21,R24, R25,R26,R27, R28,R42	P100KACT-ND	Panasonic ERJ-6GEYJ104V
30	3	Switch Tactile SPST 6 x 6 mm SMT	SW1,SW2,SW3	P12932SCT-ND	Panasonic EVQ-Q2B03W
31	2	Test Point, Black Insulator, 0.062" hole	TP2,TP3	5011K-ND	Keystone 5011
32	1	Test Point, Red Insulator, 0.062" hole	TP1	5010K-ND	Keystone 5010
33	1	IC, MCU 32-Bit 256KB Flash, 144-LQFP	U1	ATSAM3U4EA-AU-ND	Atmel ATSAM3U4EA-AU
34	2	4-Ch TVS ESD Protection SOT23-6	U2,U4	296-28203-1-ND	TI TPD4E001DBVR
35	1	IC, RS232 Driver/Receiver 3.0 to 5.5VDC 16-SOIC (3.9mm wide)	U3	296-19752-1-ND	Texas Inst MAX3232EIDR
36	1	IC Voltage Regulator 3.3V 1A LDO, SOT-223	U5	497-1228-1-ND	ST Micro LD1117AS33TR
37	1	PolyZen 5.6V PPTC protected Zener SMD	U6	ZEN056V130A24LSCT-ND	TE ZEN056V130A24LS
38	1	Filter, EMI 35dB 10A 1MHz-1GHz SMD	U7	490-5052-1-ND	Murata BNX022-01L
39	1	IC Voltage Ref 2.5V 1% Micropower SOT-23	VR1	576-1047-1-ND	Micrel LM4040DYM3-2.5
40	1	Crystal 12.00MHz, 50ppm 20pF, HC-49US leaded	Y1	631-1105-ND	Fox FOXSLF/120-20
41	1	Crystal, 32768 Hz 12.5pF cylinder leaded	Y2	535-9033-1-ND	Abracon AB26TRB-32.768KHZ-T
42	1	Capacitor, Ceramic 100nF, -20% / +80% 25V Y5V 0603	C66	490-1575-1-ND	Murata GRM188F51E104ZA01D
43	1	Capacitor, Ceramic 33pF, 5% 50V C0G 0603	C59	490-1415-1-ND	Murata GRM1885C1H330JA01D
44	2	Capacitor, Ceramic 15pF, 5% 50V C0G 0603	C60,C61	490-1407-1-ND	Murata GRM1885C1H150JA01D
45	1	Ferrite Bead, 220 Ohm @ 100MHz 300mA DC 0805	FB2	732-1602-1-ND	Wurth 742792034
46	1	Solder Jumper	JP2	SOLDER OPEN	
47	1	Resistor, 220 ohm 5% 1/10W 0603	R31	P220GCT-ND	Panasonic ERJ-3GEYJ221V
48	2	Resistor, 27 ohm 5% 1/10W 0603	R36,R38	P27GCT-ND	Panasonic ERJ-3GEYJ270V
49	5	Rubber Foot, Bumpon Black Hemisphere, .312 X.200 H	Place at 4 corners and center	SJ5746-0-ND	3M SJ61A1

U1 EACH LONG WIRE DENOTES CONNECTION NOT SHARED WITH THE HOST MICROCONTROLLER DIP SWITCH SIGNAL. MAY BE A SERIAL EEPROM SIGNAL OR A BUS SIGNAL.

HOST BUS INTERFACE

U1 EACH SHORT WIRE DENOTES CONNECTION SHARED WITH THE HOST MICROCONTROLLER, EITHER DIRECT MCU I/O, OR DIP SWITCH SIGNAL CONNECTED TO BOTH DEVICES.

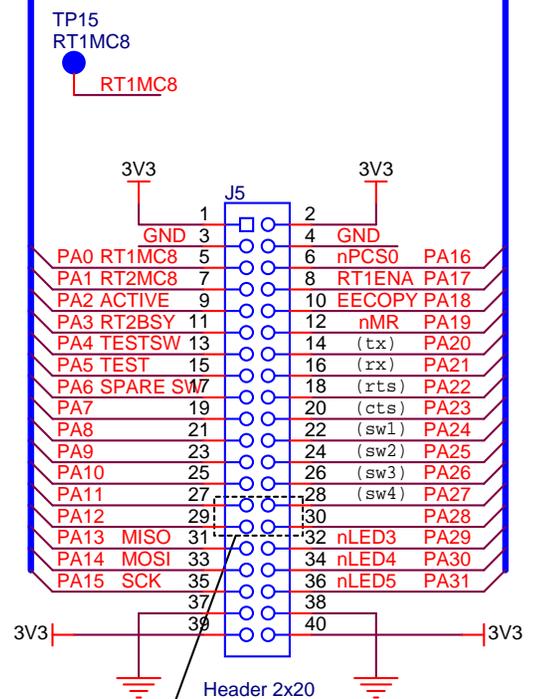
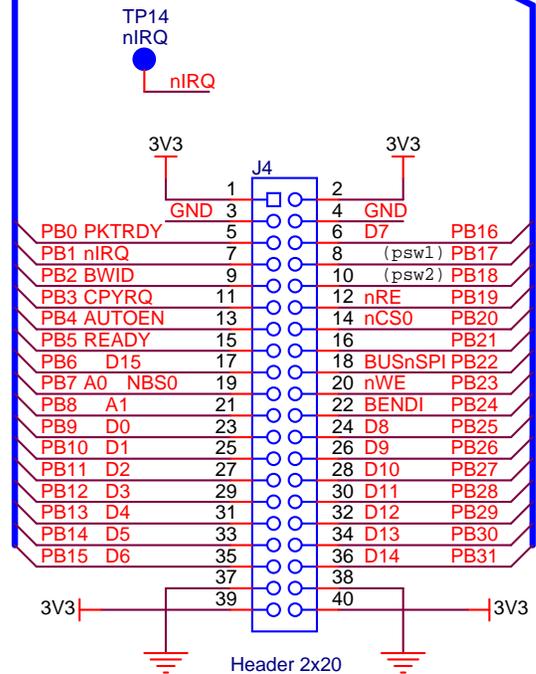
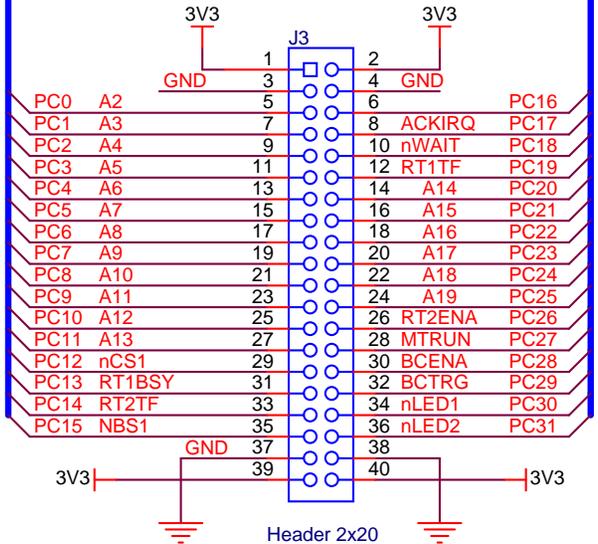
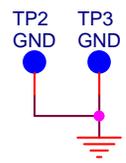
SIGNALS SHARED WITH MCU



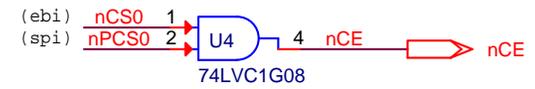
- DIP SWITCH SIGNALS
- RT1A0
 - RT1A1
 - RT1A2
 - RT1A3
 - RT1A4
 - RT1AP
 - RT1LOCK
 - RT1SSF
 - TXINHA
 - TXINHB
 - RAMEDC
 - MTSTOFF
 - RT1A0
 - RT1A1
 - RT1A2
 - RT1A3
 - RT1A4
 - RT1AP
 - RT1LOCK
 - RT1SSF
 - TXINHA
 - TXINHB
 - RAMEDC
 - MTSTOFF

3.3V BC/MT/RT1/RT2 with Integrated Transformers		
Title		
HI-2130 EVAL DAUGHTER CARD PCB		
Size A	Document Number 2130PGA EVAL.DSN	Rev NEW
Date:	Monday, January 18, 2016	Sheet 1 of 4

{1,4} PA[31:0]
 {1,3,5} PB[31:0]
 {1,3} PC[31:0]



Omit header pin install on 27,28, 29 and 30.
 Route twisted pair to J1 and J2.



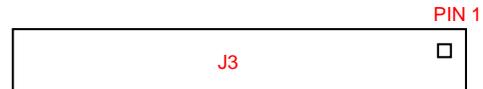
PIN 1



J5

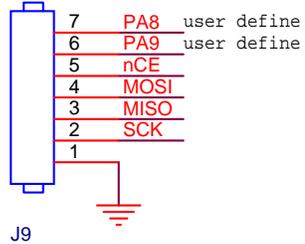
PIN 1

HEADER ORIENTATION ON THE CIRCUIT BOARD



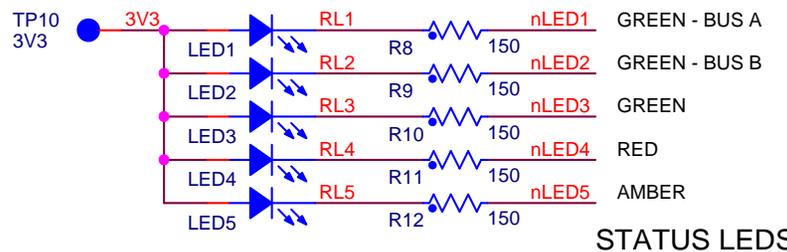
PIN 1

**Installed Headers
 SPI Analyzer header**



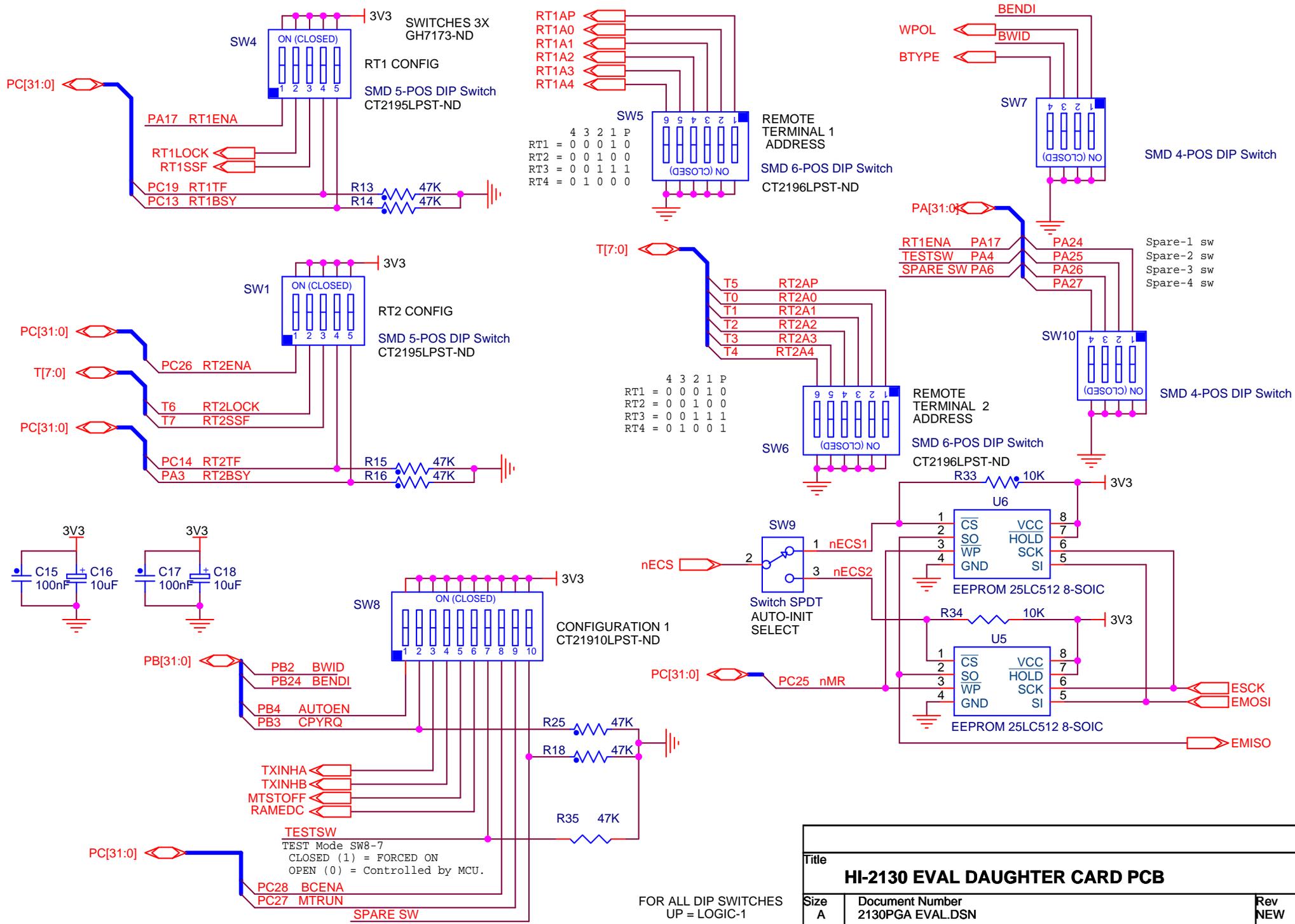
Notes:

1. psw1,psw2 push button sw's on base board.
2. tx,rx,rts,cts UART signals on base board.
3. sw1 - sw4 spare DIP sw's on page 3.



STATUS LEADS

Title		
HI-2130 EVAL DAUGHTER CARD PCB		
Size	Document Number	Rev
A	2130PGA EVAL.DSN	NEW
Date:	Monday, January 18, 2016	Sheet 2 of 4



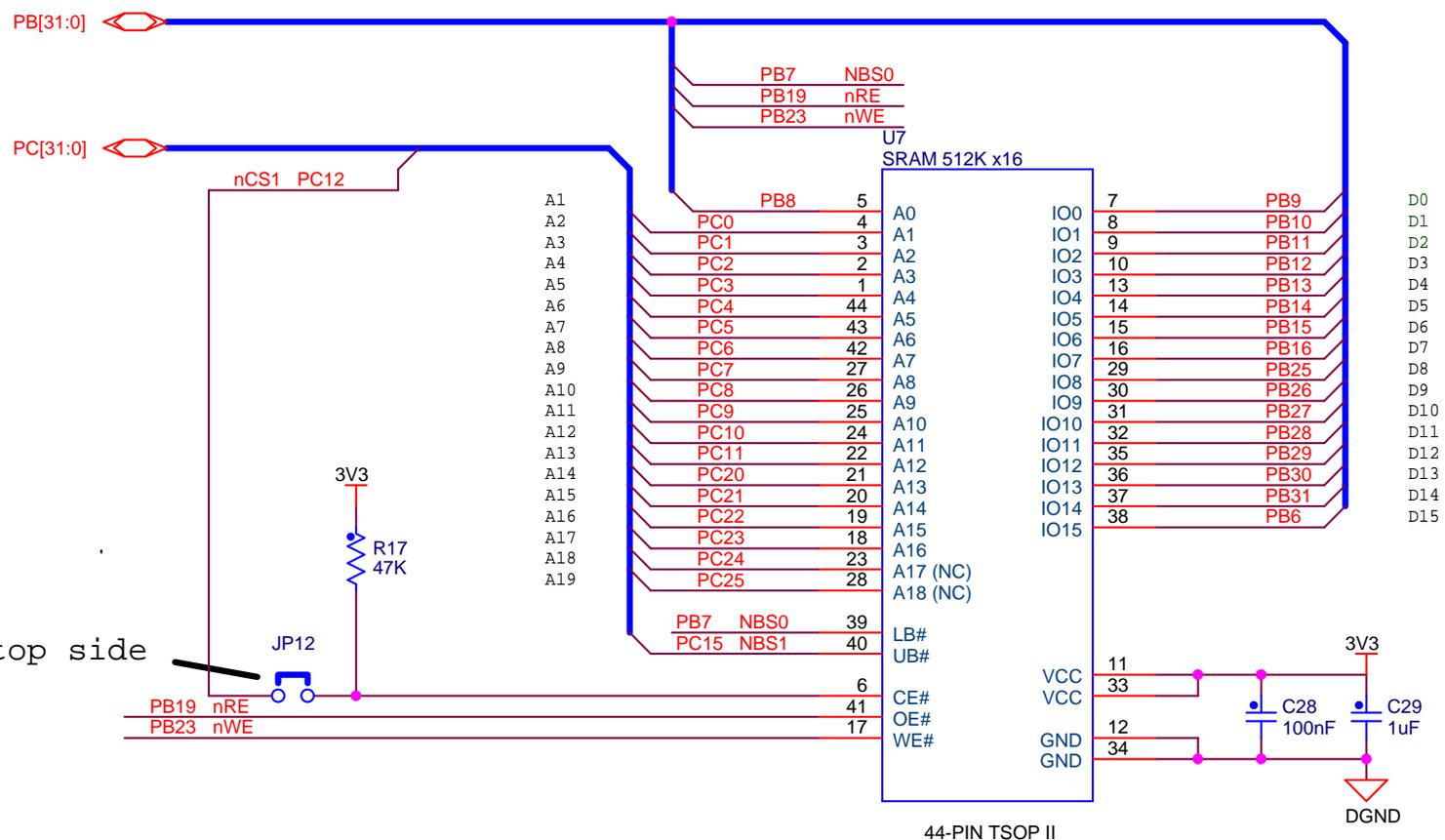
RT1AP
RT1A0
RT1A1
RT1A2
RT1A3
RT1A4

4 3 2 1 P
RT1 = 0 0 0 1 0
RT2 = 0 0 1 0 0
RT3 = 0 0 1 1 1
RT4 = 0 1 0 0 0

T5
T0
T1
T2
T3
T4

4 3 2 1 P
RT1 = 0 0 0 1 0
RT2 = 0 0 1 0 0
RT3 = 0 0 1 1 1
RT4 = 0 1 0 0 1

Spare-1 sw
Spare-2 sw
Spare-3 sw
Spare-4 sw



Solder link top side

2M 128Kx16 CY62136FV30 ADDRESS BUS A16:0
 4M 256Kx16 CY62147EV30 ADDRESS BUS A17:0
 8M 512Kx16 CY62157EV30 ADDRESS BUS A18:0

U7 SRAM may not be used on all demos. Refer to ADK users guide.



Title		
HI-2130 EVAL PCB (USE WITH STD ARM CM3 LOWER PCB)		
Size	Document Number	Rev
A	2130PGA EVAL.DSN	NEW
Date:	Monday, January 18, 2016	Sheet 4 of 4

Item	Qty	Description	Reference	DigiKey	Mfr P/N
1	1	PCB, Bare, Eval Board	N/A	-----	JetTech # 38233
2	14	Capacitor, Ceramic 0.1uF 20% 50V Z5U 0805	C1,C2,C5,C6,C7,C9,C10,C12, C13,C15,C17,C30,C28,C29	399-1176-1-ND	Kemet C0805C104M5UACTU
3	1	Capacitor, Ceramic 4.7uF 10% 6.3V X5R 0805	C4	399-3134-1-ND	Kemet C0805C475K9PACTU
4	5	Capacitor, Ceramic 10uF 10% 6.3V X5R 0805	C8,C11,C14,C16,C18	399-3138-1-ND	Kemet C0805C106K9PACTU
5	1	Capacitor 68uF 10% 6.3V Tant 400 mOhm SMD 6032	C3	399-10513-1-ND	Kemet T495C686K006ATE400
6	2	Connector 3-Lug Concentric Triax Bayonet Jack, Panel Front Mount TRB (BJ77)	J1,J2 - See Note	MilesTek 10-06570	Trompeter Electronics BJ77 Use 0.469" Round Hole
7	3	Header, Male 2x20 0.1" Pitch	J3,J4,J5	S2012E-20-ND	Sullins PEC20DAAN
8	1	Header, 1x3, 0.1" pitch	J6	DO NOT STUFF	
9	1	Header, 1x7, 0.1" pitch	J8,J9	DO NOT STUFF	
10	1	IC, AND Gate 1 Chan SOT-23-5	U4	296-22345-1-ND	TI SN74LVC1G08MDBVREP
11	1	IC SRAM 2Mbit 45ns 44TSOP	U7	428-2068-ND	Cypress CY62136EV30LL- 45ZSXI
12	1	LED Yellow 0805	LED5	160-1175-1-ND	Lite On LTST-C170YKT
13	3	LED Green 0805	LED1 - LED3	160-1179-1-ND	LiteOn LTST-C170GKT
14	1	LED Red 0805	LED4	160-1178-1-ND	LiteOn LTST-C170EKT
15	1	Osc, 50MHz, 20ppm, 3.3V, SMD 5x7mm	OSC1	535-9330-1-ND	Abrakon ASV-50.000MHZ-EJ-T
16	1	Resistor, 15 5% 1/8W 0805	R32	P150ACT-ND	Panasonic ERJ-6GEYJ150V
17	5	Resistor, 150 5% 1/8W 0805	R8,R9,R10,R11,R12	P150ACT-ND	Panasonic ERJ-6GEYJ151V
18	1	Resistor, 2.2K 5% 1/8W 0805	R36	P2.2KACT-ND	Panasonic ERJ-6GEYJ222V
19	2	Resistor, 10K 5% 1/8W 0805	R33, R34	P10KACT-ND	Panasonic ERJ-6GEYJ103V
20	8	Resistor, 47K 5% 1/8W 0805	R13,R14,R15 R16,R17,R18,R25,R35	P47KACT-ND	Panasonic ERJ-6GEYJ473V
21	2	DIP Switch 4-Position ThrHole	SW7, SW10	CT2094LPST-ND	CTS 209-4LPST
22	2	DIP Switch 5-Position ThrHole	SW1,SW4	CT2095LPST-ND	CTS 209-5LPST
23	2	DIP Switch 6-Position ThrHole	SW5,SW6	CT2096LPST-ND	CTS 209-6LPST
24	1	DIP Switch 10-Position ThrHole	SW8	CT20910LPST-ND	CTS 209-10LPST
25	1	Slide Switch SPDT SMD	SW9	563-1022-1-ND	Copal CJS-1200TB
26	2	Test Point, Red, 0.062" hole	TP4(Bus+),TP6(Bus+)	36-5010-ND	Keystone 5010
27	1	Test Point, Orange, 0.062" hole	TP10(3V3)	36-5013-ND	Keystone 5013
28	4	Test Point, Black, 0.062" hole	TP2(Gnd),TP3(Gnd), TP5(Bus-) , TP7(Bus-)	36-5011-ND	Keystone 5011
29	1	Test Point, White, 0.062" hole	TP8 (Active)	36-5012-ND	Keystone 5012
30	1	IC HI-2130 Holt CTPF (121 PGA)	U1	Holt	Holt IC
31	2	IC, EEPROM 512Kbit 20MHz 8- SOIC	U5, U6	25LC512T-I/SNCT-ND	Microchip 25LC512T-I/SN
32	1	Hookup Solid wire - 20AWG - Black - 4" Long per Board	For J1 and J2	C2028B-XX-ND	General Cable C2028A.12.01

REMOTE TERMINAL RT1 MEMORY MAP FOR HI-6130 AND HI-6131 APPLICATION DEVELOPMENT BOARD PROGRAM

	dec	hex
Descriptor Table Base Addr	1024	0400
First Buffer Address	2048	0800

Descriptor Table Sector	Device Internal Addr same as HI-6131 Addr		Data Bus Addr Hex HI-6130 Only	
	Start	End	Start	End
Receive Subaddresses	0400	047F	60000800	600008FE
Transmit Subaddresses	0480	04FF	60000900	600009FE
Receive Mode Codes	0500	057F	60000A00	60000AFE
Transmit Mode Codes	0580	05FF	60000B00	60000BFE

Buffer Assignments for Receive and Transmit Subaddresses

Receive (Rx) Subaddress or Transmit (Tx) Subaddress	Buffer Method and Data Pointer(s)		Buffer Size Words	Device Internal Addr same as HI-6131 Addr		Data Bus Addr Hex HI-6130 Only		Structures Reserved <i>MIW = Msg Info Word</i> <i>TT = TimeTag Word</i>
				Start	End	Start	End	
Rx SA1 (data pointers A, B and broadcast data pointer)	ping-pong	DPA	34	0800	0821	60001000	60001042	MIW + TT + 32 words
		DPB	34	0822	0843	60001044	60001086	same
Tx SA1 (data pointers A, B and broadcast data pointer)	ping-pong	BDP	34	0844	0865	60001088	600010CA	same
		DPA	34	0866	0887	600010CC	6000110E	same
		DPB	34	0888	08A9	60001110	60001152	same
		BDP	4	08AA	08AD	60001154	6000115A	MIW + TT + 2 pad
Rx SA30 and Tx SA30 for data wrap-around	index-0	DPA	34	08AE	08CF	6000115C	6000119E	MIW + TT + 32 words
Rx SA2	index-32	DPA	1088	08D0	0D0F	600011A0	60001A1E	32 x (MIW + TT + 32 words)
		BDP	34	0D10	0D31	60001A20	60001A62	
Tx SA2	index-32	DPA	1088	0D32	1171	60001A64	600022E2	MIW + TT + 2 pad
		BDP	4	1172	1175	600022E4	600022EA	
Rx SA3	circ1-32	DPA	1088	1176	15B5	600022EC	60002B6A	32 x (MIW + TT + 32 words)
		pad	32	15B6	15D5	60002B6C	60002BAA	
Tx SA3	circ1-32	DPA	1088	15D6	1A15	60002BAC	6000342A	32 x (MIW + TT + 32 words)
		pad	32	1A16	1A35	6000342C	6000346A	
shared buffer: all unimplemented Rx subaddresses	index-0	DPA	34	1A36	1A57	6000346C	600034AE	MIW + TT + 32 words
shared buffer: all unimplemented Tx subaddresses	index-0	DPA	34	1A58	1A79	600034B0	600034F2	MIW + TT + 32 words
RAM assigned below (MCs)	---	---	142	1A7A	1B07	600034F4	6000360E	
unassigned RAM	---	---	72	1B08	1B4F	60003610	6000369E	
assigned to BC	---	---	176	1B50	1BFF	600036A0	600037FE	BC Mode Command Data BC Instruction List
Rx & Tx SA4	circ-2 256 msg max	MIB	512	1C00	1DFF	600036A0	60003BFE	256 x (MIW + TT)
		DPA	8192	1E00	3DFF	60003C00	60007BFE	
assigned to BC	---	---	256	3E00	3EFF	60007C00	60007DFE	BC Msg Control Blocks
unassigned RAM	---	---	256	3F00	3FFF	60007C00	60007FFE	

Shared Buffer Assignments for Undefined and Reserved Mode Code Commands

These RAM buffer allocations for mode code commands only apply when the application does not use the SMCP option.

Undefined & Reserved Receive (Rx) Mode Codes Transmit (Tx) Mode Codes	Buffer Method and Data Pointer(s)		Buffer Size Words	Device Internal Addr same as HI-6131 Addr		Data Bus Addr Hex HI-6130 Only		Structures Reserved <i>MIW = Msg Info Word</i> <i>TT = TimeTag Word</i>
				Start	End	Start	End	
shared buffer: undefined Rx MC0 - MC15	index-0	DPA	4	1A7A	1A7D	600034F4	600034FA	MIW + TT, 0 data, 2 pad
shared buffer: undefined Rx MC16, undefined Rx MC18 - MC19, reserved Rx MC22 - MC31	index-0	DPA	4	1A7E	1A81	600034FC	60003502	MIW + TT, 1 data, 1 pad
shared buffer: undefined Tx MC9 - MC15	index-0	DPA	4	1A82	1A85	00803504	6000350A	MIW + TT, 0 data, 2 pad
shared buffer: undefined Tx MC17, undefined Tx MC20 - MC21, reserved Tx MC22 - MC31	index-0	DPA	4	1A86	1A89	6000350C	60003512	MIW + TT, 1 data, 1 pad

Buffer Assignments for Defined Transmit Mode Code Commands MC0 - MC8 (No Data Word)

These RAM buffer allocations for mode code commands only apply when the application does not use the SMCP option.

Defined Transmit (Tx) Mode Code Commands, No Data	Buffer Method and Data Pointer(s)		Buffer Size Words	Device Internal Addr same as HI-6131 Addr		Data Bus Addr Hex HI-6130 Only		Structures Reserved <i>MIW = Msg Info Word</i> <i>TT = TimeTag Word</i>
				Start	End	Start	End	
Tx MC0	ping-pong	DPA	2	1A8A	1A8B	60003514	60003516	MIW + TT
		DPB	2	1A8C	1A8D	60003518	6000351A	same
		BDP	2	1A8E	1A8F	6000351C	6000351E	same
Tx MC1	ping-pong	DPA	2	1A90	1A91	60003520	60003522	MIW + TT
		DPB	2	1A92	1A93	60003524	60003526	same
		BDP	2	1A94	1A95	60003528	6000352A	same
Tx MC2	ping-pong	DPA	2	1A96	1A97	6000352C	6000352E	MIW + TT
		DPB	2	1A98	1A99	60003530	60003532	same
		BDP	2	1A9A	1A9B	60003534	60003536	same
Tx MC3	ping-pong	DPA	2	1A9C	1A9D	60003538	6000353A	MIW + TT
		DPB	2	1A9E	1A9F	6000353C	6000353E	same
		BDP	2	1AA0	1AA1	60003540	60003542	same
Tx MC4	ping-pong	DPA	2	1AA2	1AA3	60003544	60003546	MIW + TT
		DPB	2	1AA4	1AA5	60003548	6000354A	same
		BDP	2	1AA6	1AA7	6000354C	6000354E	same
Tx MC5	ping-pong	DPA	2	1AA8	1AA9	60003550	60003552	MIW + TT
		DPB	2	1AAA	1AAB	60003554	60003556	same
		BDP	2	1AAC	1AAD	60003558	6000355A	same
Tx MC6	ping-pong	DPA	2	1AAE	1AAF	6000355C	6000355E	MIW + TT
		DPB	2	1AB0	1AB1	60003560	60003562	same
		BDP	2	1AB2	1AB3	60003564	60003566	same
Tx MC7	ping-pong	DPA	2	1AB4	1AB5	60003568	6000356A	MIW + TT
		DPB	2	1AB6	1AB7	6000356C	6000356E	same
		BDP	2	1AB8	1AB9	60003570	60003572	same
Tx MC8	ping-pong	DPA	2	1ABA	1ABB	60003574	60003576	MIW + TT
		DPB	2	1ABC	1ABD	60003578	6000357A	same
		BDP	2	1ABE	1ABF	6000357C	6000357E	same

Buffer Assignments for Defined Transmit Mode Code Commands MC16, MC18 and MC19 (1 Data Word)
These RAM buffer allocations for mode code commands only apply when the application does not use the SMCP option.

Defined Transmit (Tx) Mode Code Commands with Data Word	Buffer Method and Data Pointer(s)	Buffer Size Words	Device Internal Addr same as HI-6131 Addr		Data Bus Addr Hex HI-6130 Only		Structures Reserved MIW = Msg Info Word TT = TimeTag Word	
			Start	End	Start	End		
Tx MC16	ping-pong	DPA	4	1AC0	1AC3	60003580	60003586	MIW + TT, 1 data, 1 pad
		DPB	4	1AC4	1AC7	60003588	6000358E	same
		BDP	4	1AC8	1ACB	60003590	60003596	same
Tx MC18	ping-pong	DPA	4	1ACC	1ACF	60003598	6000359E	MIW + TT, 1 data, 1 pad
		DPB	4	1AD0	1AD3	600035A0	600035A6	same
		BDP	4	1AD4	1AD7	600035A8	600035AE	same
Tx MC19	ping-pong	DPA	4	1AD8	1ADB	600035B0	600035B6	MIW + TT, 1 data, 1 pad
		DPB	4	1ADC	1ADF	600035B8	600035BE	same
		BDP	4	1AE0	1AE3	600035C0	600035C6	same

Buffer Assignments for Defined Receive Mode Code Commands MC17, MC20 and MC21 (1 Data Word)
These RAM buffer allocations for mode code commands only apply when the application does not use the SMCP option.

Defined Receive (Rx) Mode Code Commands with Data Word	Buffer Method and Data Pointer(s)	Buffer Size Words	Device Internal Addr same as HI-6131 Addr		Data Bus Addr Hex HI-6130 Only		Structures Reserved MIW = Msg Info Word TT = TimeTag Word	
			Start	End	Start	End		
Rx MC17	ping-pong	DPA	4	1AE4	1AE7	600035C8	600035CE	MIW + TT, 1 data, 1 pad
		DPB	4	1AE8	1AEB	600035D0	600035D6	same
		BDP	4	1AEC	1AEF	600035D8	600035DE	same
Rx MC20	ping-pong	DPA	4	1AF0	1AF3	600035E0	600035E6	MIW + TT, 1 data, 1 pad
		DPB	4	1AF4	1AF7	600035E8	600035EE	same
		BDP	4	1AF8	1AFB	600035F0	600035F6	same
Rx MC21	ping-pong	DPA	4	1AFC	1AFF	600035F8	600035FE	MIW + TT, 1 data, 1 pad
		DPB	4	1B00	1B03	60003600	60003606	same
		BDP	4	1B04	1B07	60003608	6000360E	same

6130 Demo Memory Map.xls

Notes:

1. All addresses shown are expressed as hexadecimal values.
2. Addressing for HI-6131 uses device internal addresses. Bus addressing for HI-6130 is offset by chip select base address 0x60000000 and microprocessor uses byte addressing so all address offsets are doubled. (The LSB becomes upper/lower byte select for each word.)
3. Memory allocations are shared for undefined and reserved mode code commands, and unimplemented subaddress commands. These commands are grouped by like requirements, and share common RAM resources (bit bucket).
4. For messages needing an odd number of words, an extra "pad" word is added so the next buffer begins at an even address.
5. Subaddresses using circular buffer Mode 1 are followed by a 32-word overrun buffer, in case a 32 data word receive command arrives with just one location remaining before "buffer full" attainment.

REMOTE TERMINAL RT2 MEMORY MAP FOR HI-6130 AND HI-6131 APPLICATION DEVELOPMENT BOARD PROGRAM

	dec	hex
Descriptor Table Base Addr	1536	0600
First Buffer Address	16384	4000

Descriptor Table Sector	Device Internal Addr same as HI-6131 Addr		Data Bus Addr Hex HI-6130 Only	
	Start	End	Start	End
Receive Subaddresses	0600	067F	60000C00	60000CFE
Transmit Subaddresses	0680	06FF	60000D00	60000DFE
Receive Mode Codes	0700	077F	60000E00	60000EFE
Transmit Mode Codes	0780	07FF	60000F00	60000FFE

Buffer Assignments for Receive and Transmit Subaddresses

Receive (Rx) Subaddress or Transmit (Tx) Subaddress	Buffer Method and Data Pointer(s)		Buffer Size Words	Device Internal Addr same as HI-6131 Addr		Data Bus Addr Hex HI-6130 Only		Structures Reserved <i>MIW = Msg Info Word</i> <i>TT = TimeTag Word</i>	
				Start	End	Start	End		
Rx SA1 (data pointers A, B and broadcast data pointer) Tx SA1 (data pointers A, B and broadcast data pointer)	ping-pong	DPA	34	4000	4021	60008000	60008042	MIW + TT + 32 words	
		DPB	34	4022	4043	60008044	60008086	same	
		BDP	34	4044	4065	60008088	600080CA	same	
	ping-pong	DPA	34	4066	4087	600080CC	6000810E	same	
		DPB	34	4088	40A9	60008110	60008152	same	
		BDP	4	40AA	40AD	60008154	6000815A	MIW + TT + 2 pad	
Rx SA30 and Tx SA30 for data wrap-around	index-0	DPA	34	40AE	40CF	6000815C	6000819E	MIW + TT + 32 words	
Rx SA2	index-32	DPA	1088	40D0	450F	600081A0	60008A1E	32 x (MIW + TT + 32 words)	
		BDP	34	4510	4531	60008A20	60008A62		
Tx SA2		DPA	1088	4532	4971	60008A64	600092E2		
		BDP	4	4972	4975	600092E4	600092EA	MIW + TT + 2 pad	
Rx SA3	circ1-32	DPA	1088	4976	4DB5	600092EC	60009B6A	32 x (MIW + TT + 32 words)	
		pad	32	4DB6	4DD5	60009B6C	60009BAA		pad for overrun
Tx SA3		DPA	1088	4DD6	5215	60009BAC	6000A42A		32 x (MIW + TT + 32 words)
	pad	32	5216	5235	6000A42C	6000A46A	pad for overrun		
shared buffer: all unimplemented Rx subaddresses	index-0	DPA	34	5236	5257	6000A46C	6000A4AE	MIW + TT + 32 words	
shared buffer: all unimplemented Tx subaddresses	index-0	DPA	34	5258	5279	6000A4B0	6000A4F2	MIW + TT + 32 words	
RAM assigned below (MCs)	---	---	142	527A	5307	6000A4F4	6000A60E		
assigned to BC	---	---	248	5308	53FF	6000A610	6000A7FE	BC Msg Data Buffers (excl mode commands)	
SA4 not used by RT2	---	---	11264	5400	7FFF	6000A800	6000FFFE	IMT Stack or SMT Stacks	

Shared Buffer Assignments for Undefined and Reserved Mode Code Commands

These RAM buffer allocations for mode code commands only apply when the application does not use the SMCP option.

Undefined & Reserved Receive (Rx) Mode Codes Transmit (Tx) Mode Codes	Buffer Method and Data Pointer(s)		Buffer Size Words	Device Internal Addr same as HI-6131 Addr		Data Bus Addr Hex HI-6130 Only		Structures Reserved MIW = Msg Info Word TT = TimeTag Word
				Start	End	Start	End	
shared buffer: undefined Rx MC0 - MC15	index-0	DPA	4	527A	527D	6000A4F4	6000A4FA	MIW + TT, 0 data, 2 pad
shared buffer: undefined Rx MC16, undefined Rx MC18 - MC19, reserved Rx MC22 - MC31	index-0	DPA	4	527E	5281	6000A4FC	6000A502	MIW + TT, 1 data, 1 pad
shared buffer: undefined Tx MC9 - MC15	index-0	DPA	4	5282	5285	6000A504	6000A50A	MIW + TT, 0 data, 2 pad
shared buffer: undefined Tx MC17, undefined Tx MC20 - MC21, reserved Tx MC22 - MC31	index-0	DPA	4	5286	5289	6000A50C	6000A512	MIW + TT, 1 data, 1 pad

Buffer Assignments for Defined Transmit Mode Code Commands MC0 - MC8 (No Data Word)

These RAM buffer allocations for mode code commands only apply when the application does not use the SMCP option.

Defined Transmit (Tx) Mode Code Commands, No Data	Buffer Method and Data Pointer(s)		Buffer Size Words	Device Internal Addr same as HI-6131 Addr		Data Bus Addr Hex HI-6130 Only		Structures Reserved MIW = Msg Info Word TT = TimeTag Word
				Start	End	Start	End	
Tx MC0	ping-pong	DPA	2	528A	528B	6000A514	6000A516	MIW + TT
		DPB	2	528C	528D	6000A518	6000A51A	same
		BDP	2	528E	528F	6000A51C	6000A51E	same
Tx MC1	ping-pong	DPA	2	5290	5291	6000A520	6000A522	MIW + TT
		DPB	2	5292	5293	6000A524	6000A526	same
		BDP	2	5294	5295	6000A528	6000A52A	same
Tx MC2	ping-pong	DPA	2	5296	5297	6000A52C	6000A52E	MIW + TT
		DPB	2	5298	5299	6000A530	6000A532	same
		BDP	2	529A	529B	6000A534	6000A536	same
Tx MC3	ping-pong	DPA	2	529C	529D	6000A538	6000A53A	MIW + TT
		DPB	2	529E	529F	6000A53C	6000A53E	same
		BDP	2	52A0	52A1	6000A540	6000A542	same
Tx MC4	ping-pong	DPA	2	52A2	52A3	6000A544	6000A546	MIW + TT
		DPB	2	52A4	52A5	6000A548	6000A54A	same
		BDP	2	52A6	52A7	6000A54C	6000A54E	same
Tx MC5	ping-pong	DPA	2	52A8	52A9	6000A550	6000A552	MIW + TT
		DPB	2	52AA	52AB	6000A554	6000A556	same
		BDP	2	52AC	52AD	6000A558	6000A55A	same
Tx MC6	ping-pong	DPA	2	52AE	52AF	6000A55C	6000A55E	MIW + TT
		DPB	2	52B0	52B1	6000A560	6000A562	same
		BDP	2	52B2	52B3	6000A564	6000A566	same
Tx MC7	ping-pong	DPA	2	52B4	52B5	6000A568	6000A56A	MIW + TT
		DPB	2	52B6	52B7	6000A56C	6000A56E	same
		BDP	2	52B8	52B9	6000A570	6000A572	same
Tx MC8	ping-pong	DPA	2	52BA	52BB	6000A574	6000A576	MIW + TT
		DPB	2	52BC	52BD	6000A578	6000A57A	same
		BDP	2	52BE	52BF	6000A57C	6000A57E	same

Buffer Assignments for Defined Transmit Mode Code Commands MC16, MC18 and MC19 (1 Data Word)
These RAM buffer allocations for mode code commands only apply when the application does not use the SMCP option.

Defined Transmit (Tx) Mode Code Commands with Data Word	Buffer Method and Data Pointer(s)		Buffer Size Words	Device Internal Addr same as HI-6131 Addr		Data Bus Addr Hex HI-6130 Only		Structures Reserved MIW = Msg Info Word TT = TimeTag Word
				Start	End	Start	End	
Tx MC16	ping-pong	DPA	4	52C0	52C3	6000A580	6000A586	MIW + TT, 1 data, 1 pad
		DPB	4	52C4	52C7	6000A588	6000A58E	same
		BDP	4	52C8	52CB	6000A590	6000A596	same
Tx MC18	ping-pong	DPA	4	52CC	52CF	6000A598	6000A59E	MIW + TT, 1 data, 1 pad
		DPB	4	52D0	52D3	6000A5A0	6000A5A6	same
		BDP	4	52D4	52D7	6000A5A8	6000A5AE	same
Tx MC19	ping-pong	DPA	4	52D8	52DB	6000A5B0	6000A5B6	MIW + TT, 1 data, 1 pad
		DPB	4	52DC	52DF	6000A5B8	6000A5BE	same
		BDP	4	52E0	52E3	6000A5C0	6000A5C6	same

Buffer Assignments for Defined Receive Mode Code Commands MC17, MC20 and MC21 (1 Data Word)
These RAM buffer allocations for mode code commands only apply when the application does not use the SMCP option.

Defined Receive (Rx) Mode Code Commands with Data Word	Buffer Method and Data Pointer(s)		Buffer Size Words	Device Internal Addr same as HI-6131 Addr		Data Bus Addr Hex HI-6130 Only		Structures Reserved MIW = Msg Info Word TT = TimeTag Word
				Start	End	Start	End	
Rx MC17	ping-pong	DPA	4	52E4	52E7	6000A5C8	6000A5CE	MIW + TT, 1 data, 1 pad
		DPB	4	52E8	52EB	6000A5D0	6000A5D6	same
		BDP	4	52EC	52EF	6000A5D8	6000A5DE	same
Rx MC20	ping-pong	DPA	4	52F0	52F3	6000A5E0	6000A5E6	MIW + TT, 1 data, 1 pad
		DPB	4	52F4	52F7	6000A5E8	6000A5EE	same
		BDP	4	52F8	52FB	6000A5F0	6000A5F6	same
Rx MC21	ping-pong	DPA	4	52FC	52FF	6000A5F8	6000A5FE	MIW + TT, 1 data, 1 pad
		DPB	4	5300	5303	6000A600	6000A606	same
		BDP	4	5304	5307	6000A608	6000A60E	same

6130 Demo Memory Map.xls

Notes:

- All addresses shown are expressed as hexadecimal values.
- Addressing for HI-6131 uses device internal addresses. Bus addressing for HI-6130 is offset by chip select base address 0x60000000 and microprocessor uses byte addressing so all address offsets are doubled. (The LSB becomes upper/lower byte select for each word.)
- Memory allocations are shared for undefined and reserved mode code commands, and unimplemented subaddress commands. These commands are grouped by like requirements, and share common RAM resources (bit bucket).
- For messages needing an odd number of words, an extra "pad" word is added so the next buffer begins at an even address.
- Subaddresses using circular buffer Mode 1 are followed by a 32-word overrun buffer, in case a 32 data word receive command arrives with just one location remaining before "buffer full" attainment.

BUS CONTROLLER MEMORY MAP FOR HI-6130 AND HI-6131 APPLICATION DEVELOPMENT BOARD PROGRAM

BC Message Blocks

used in application development kit program

Block Number	Command Type	# Block Words	Block Start Addr	Block End Addr	HI-6130 Bus Addr
1	Tx SA *	8	3E00	3E07	60007C00
2	Tx SA *	8	3E08	3E0F	60007C10
3	Rx SA	8	3E10	3E17	60007C20
4	B Rx SA	8	3E18	3E1F	60007C30
5	B Rx SA	8	3E20	3E27	60007C40
6	Tx MC2 ND	8	3E28	3E2F	60007C50
7	Tx MC18 D	8	3E30	3E37	60007C60
8	Rx MC21 D	8	3E38	3E3F	60007C70
RTRT1	RTRT	16	3E40	3E4F	60007C80
RTRT2	B RTRT	16	3E50	3E5F	60007CA0
<i>available for expansion through end addr...</i>					
		160		3EFF	60007DFE

Corresponding BC Message Data Buffers

used in application development kit program

Number of Words	Buffer Start Addr	Buffer End Addr	HI-6130 Bus Addr
32	5308	5327	6000A610
32	5308	5327	6000A610
32	5328	5347	6000A650
32	5348	5367	6000A690
32	5368	5387	6000A6D0
0	no data	no data	no data
1	1B62	----	600036C6
1	1B55	----	600036AC
32	5388	53A7	6000A710
32	53A8	53C7	6000A750
<i>available for expansion through end addr...</i>			
56		53FF	6000A7FE

* These 2 message blocks are Transmit Subaddress commands to the same subaddress, so use same Tx buffer.

BC Fixed Mode Command Data Word Storage

used in application development kit program

	Mode Code Cmd	# Data Words	Mode Cmd Data Addr	HI-6130 Bus Addr
Receive Mode Code Commands with Data	RxMC 16	1	1B50	600036A0
	RxMC 17	1	1B51	600036A2
	RxMC 18	1	1B52	600036A4
	RxMC 19	1	1B53	600036A6
	RxMC 20	1	1B54	600036A8
	RxMC 21	1	1B55	600036AA
	RxMC 22	1	1B56	600036AC
	RxMC 23	1	1B57	600036AE
	RxMC 24	1	1B58	600036B0
	RxMC 25	1	1B59	600036B2
	RxMC 26	1	1B5A	600036B4
	RxMC 27	1	1B5B	600036B6
	RxMC 28	1	1B5C	600036B8
	RxMC 29	1	1B5D	600036BA
	RxMC 30	1	1B5E	600036BC
	RxMC 31	1	1B5F	600036BE
Transmit Mode Code Commands with Data	TxMC 16	1	1B60	600036C0
	TxMC 17	1	1B61	600036C2
	TxMC 18	1	1B62	600036C4
	TxMC 19	1	1B63	600036C6
	TxMC 20	1	1B64	600036C8
	TxMC 21	1	1B65	600036CA
	TxMC 22	1	1B66	600036CC
	TxMC 23	1	1B67	600036CE
	TxMC 24	1	1B68	600036D0
	TxMC 25	1	1B69	600036D2
	TxMC 26	1	1B6A	600036D4
	TxMC 27	1	1B6B	600036D6
	TxMC 28	1	1B6C	600036D8
	TxMC 29	1	1B6D	600036DA
	TxMC 30	1	1B6E	600036DC
	TxMC 31	1	1B6F	600036DE

BC Instruction List Addresses

used in application development kit program

Op Code #	Op Code Addr	Msg Block called	HI-6130 Bus Addr
0	1B70	op WTG	600036E0
2	1B72	1	600036E4
4	1B74	op WTG	600036E8
6	1B76	2	600036EC
8	1B78	op WTG	600036F0
10	1B7A	3	600036F4
12	1B7C	op WTG	600036F8
14	1B7E	4	600036FC
16	1B80	op WTG	60003700
18	1B82	5	60003704
20	1B84	op WTG	60003708
22	1B86	6	6000370C
24	1B88	op WTG	60003710
26	1B8A	7	60003714
28	1B8C	op WTG	60003718
30	1B8E	8	6000371C
32	1B90	op WTG	60003720
34	1B92	RTRT1	60003724
36	1B94	op WTG	60003728
38	1B96	RTRT2	6000372C
40	1B98	op WTG	60003730
42	1B9A	2	60003734
44	1B9C	op JMP	60003738
46	1B9E	Execute op codes can call Message Blocks in any order!	6000373C
48	1BA0		60003740
50	1BA2		60003744
52	1BA4		60003748
54	1BA6		6000374C
56	1BA8		60003750
58	1BAA		60003754
60	1BAC		60003758
62	1BAE		6000375C
<i>available for expansion through end addr...</i>			
142	1BFE		600037FC

Notes:

1. Command Types: SA = Subaddress cmd, MC = Mode Code cmd, ND = no data, D = with data, B = broadcast.
2. All 4-digit hexadecimal addresses refer to the internal IC address, equal to the address used by HI-6131 SPI.
3. The HI-6130 Bus Address = ARM MCU chip select base addr 0x60000000 + 2 x (feature's IC address)

MISCELLANEOUS RAM STRUCTURES NOT ALREADY LISTED

RAM Structure	Start Address	End Address	Number of Words
Interrupt Log Buffer	0x0180	0x01BF	64
Bus Controller General Purpose Queue	0x00C0	0x00FF	64
Bus Controller Call Stack	0x0054	0X005B	8
RT1 Temporary Receive Buffer	0x01C0	0x01DF	32
RT2 Temporary Receive Buffer	0x01E0	0x01FF	32
RT1 Command Illegalization Table	0x0200	0x02FF	256
RT2 Command Illegalization Table	0x0300	0x03FF	256
SMT or IMT Message Filter Table	0x0100	0x017F	128
SMT or IMT Address List	0x00B0	0x00B7	8
SMT Command Stack	0x5400	0x5FFF	3072
SMT Data Stack	0x6000	0x7FFF	24577
IMT Combined Stack	0x5400	0x6400	6400