



# AN-6130 開発キット

2015 年 2 月



株式会社ナセル

**履歴**

履歴	日付	変更内容
AN-6130 New	2011 / 9 / 15	初版リリース
AN-6130 Rev.A	2011 / 10 / 6	RAM メモリ・マップ追加 回路図と部品表の更新
AN-6130 Rev.B	2012 / 6 / 15	文書改訂、ソフトウェア・プロジェクトを更新。この新しいプロジェクトは HI-6130 (HI-6131 ではなく) のみをサポートしますが、IAR IDE のフォルダと構成を使用した整理されたプロジェクトを提供
AN-6130 Rev.C	2013 / 1 / 15	Windows Vista および Windows 7 ユーザーのインストール手順の更新
AN-6130 Rev.D	2015 / 4 / 15	デモ・プログラム・ソフトウェアを IAR との互換性のためにアップグレードし、デフォルトのデモとして Simple RT を使用

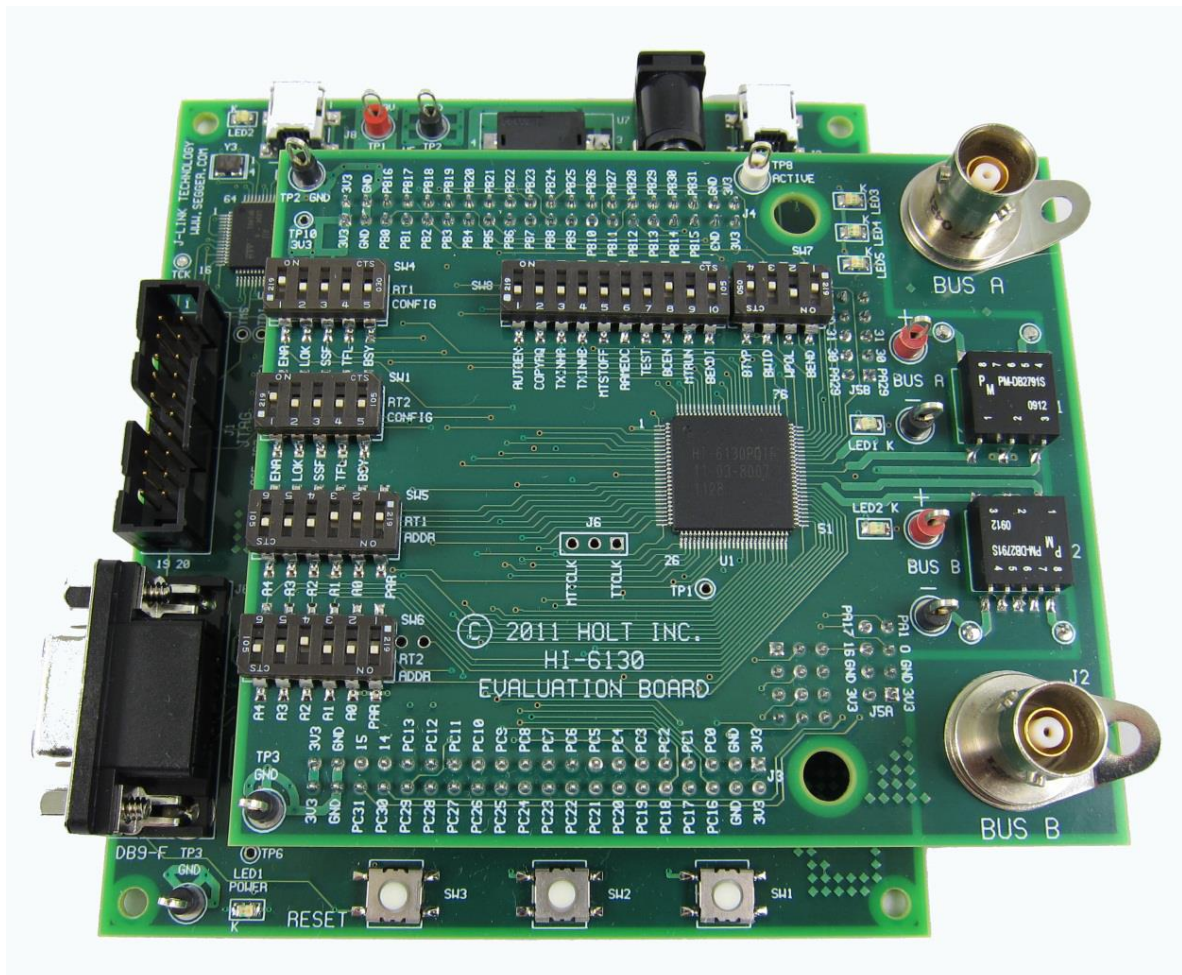
本文書は、Holt 社の『AN-6130』の和訳になります。

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-6130 アプリケーション開発キットは、MIL-STD-1553 用の HI-6130 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・アセンブリと 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、2×RT 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、このキットは、IAR システムズの ARM 用 Embedded Workbench®を、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のインターフェイスが含まれています。

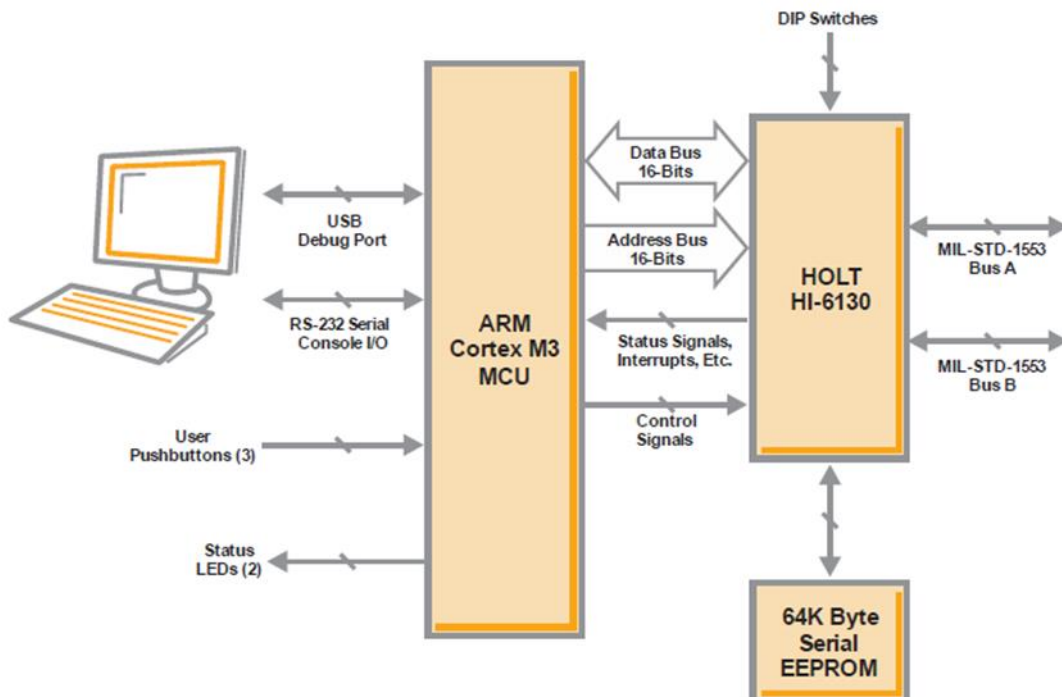
このガイドでは、ボードの設定および実行方法について説明します。資料と必要なすべてのプロジェクト・ソフトウェアが、Holt の CD-ROM に含まれます。デモ・ソフトウェア・バージョンがすでにマイコンのフラッシュ・メモリにプログラムされています；提供されるソフトウェア開発ツールをインストールまたは実行する必要無く、ボードは箱から出してすぐに動作可能です。



## Evaluation Kit Contents [開発キット構成]

- 本ユーザー・ガイド
- Holt HI-6130 ソフトウェア・プロジェクトおよび、ドキュメント CD
- IAR システムズの ARM 用 Embedded Workbench® (32KB サイズ限定版) インストール CD
- 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
- 2 ボード構成で、
  - 上段 HI-6130 デバイス、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。
  - 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]

上段のターゲット・ボードと下段の MCU ボード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。

取り外し可能なターゲット・ボードは、ユーザー提供の代替マイクロプロセッサまたは FPGA ボードに接続するために、付属の MCU ボードから分離できます。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-6130 コンフィグレーション・ピン (リモート・ターミナル・アドレス設定端子など) は、上段のターゲット・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 ボードは、フラッシュ・プログラム可能な、Atmel 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・インターフェイスにより、HI-6130 に接続しています。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・ボタンは、ソフトウェアとの対話のために利用可能です。RESET プッシュ・ボタンは、ターゲット・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。

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 2×10 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、下段ボードの底面のジャンパ 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。)

## New In This Version [このバージョンの新機能]

デモ・プログラム、Holt Hi-6130-40 EBI Demo 2.0 および、Holt HI-6130 Simple RT V1.0 は、それぞれ 2.1 および 1.01 にアップデートされています。どちらもプログラムも、IAR の推奨する「マイドキュメント」パスから実行するように変更されました。以前の IAR プロジェクトは、

「C:\Program Files\IAR Systems\...\at91sam3u - ek」にインストールされていました。これらのデモ・プロジェクトをコンパイルするには、IAR EWARM バージョン 7.1 を使用しました。通常 IAR の新しいバージョンでは、古いプロジェクトを開く際に問題はありません。新しい Holt プロジェクトには、IAR EWARM 7.1 以上を推奨します。古いプロジェクトを「マイドキュメント」パスを使用するように変換するには、付属の CD-ROM に収録されている「Holt Technical Note- IAR Demo Project Installation」を参照してください。

新しいデモ・プログラム、HI-6130 Simple RT は、Atmel Cortex M3 マイクロコントローラにフラッシュされているデフォルトのデモ・プログラムです。このプログラムは、HI-6130 のより多くの機能を実証し、RT トラフィック・データをコンソールに表示します。Simple RT デモは、AN-6130DG デザインガイドで完全に説明されており RT1 を初期化する方法を段階的に説明しています。他のデモ・プログラムは、Holt の CD-ROM に収録されています。

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

Holt HI-6130 アプリケーション開発キットは、BC、SMT バスモニタ、2×独立 RT 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。2×RT 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。RT アドレス 3 と 4 が BC メッセージ・レパトリリーとしてプログラムされています。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。

1. バスのアクティビティを観察するには、BUS A と BUS B の赤色のテスト・ピンにオシロスコープを接続してください。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す。
2. 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 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）
3. あらかじめプログラムされたデモでは、MCU ボードのシリアル・ポートとターミナル・エミュレーション・プログラムを実行する Windows コンピュータ間でコンソール I/O を提供します。付属の DB9 シリアル・ケーブルを使用して、Windows コンピュータ上のシリアル・ポートと MCU ボードを接続します。Win XP または、Win2000 の場合、Windows アクセサリから、ハイパーターミナルを開きます。Vista または Windows7 を使用する場合、Tera Term をインストールし起動します。ターミナルの通信設定を、ボーレート：115,200bps、パリティ：なし、データビット：8、ストップビット：1、フロー制御：OFF、に設定します。正しく設定されている場合、電源投入時または RESET プッシュ・ボタンが押されたときにテキストヘッダが表示されます。
4. 5VDC 電源アダプタを、下段の基板の電源入力ジャックに接続します。コマンド・メニューが、ハイパーターミナル（TeraTerm）ウィンドウ上に表示されます。
5. バスコントローラは、リモートターミナル 3 と 4 に対して、MIL-STD-1553 コマンドを繰り返し連続で実行するようにプログラムされています。各バス・コマンドは、「Wait for Trigger」OP コードが付いています。MCU は、数字の「1」がコンピュータのキーボードで押されるたびに、BC トリガ・パルスを発行するようにプログラムされています。（プログラムを再コンパイルし、コンソール I/O オプションを外すことにより、バスコントローラは、MCU ボード上の SW1 ボタンを押すことで同じことが行えますが、ハイパーターミナル（Tera Term）コンソール出力は無効になります。）
6. 数字の「1」キーがコンピュータのボードで押されるたびに、新しいバスコントローラ・コマンドは、RT アドレス 3 と 4 に発行されます（または、RT-RT メッセージの場合、RT アドレス 3 と 4 に対応）。リモートターミナルの応答が観測され、ハイパーターミナル（Tera Term）コンソール・ウィンドウには、各キーを押すための新しいメッセージの結果を報告することができます。

7. このデモ・プログラムの新機能では、デモの Simple RT 部分で RT トラフィック・データがコンソールに表示されます。RT1 から SA1 への受信コマンドは、最初のデータ・ワードの内容に応じて、ボード上の緑色または橙色の LED を点滅させます。詳細、使用方法、データ・トラフィック・コンソールの出力例については、AN-6130DG を参照してください。

## Getting Started With Full Evaluation [フル評価の開始]

次の手順は、HI-6130 アプリケーション開発キットを使用して、インストール、C コンパイラの設定、そして、変更したデモ・プロジェクトのロード方法について説明します。

1. デフォルト・インストールを使用して、ご使用の Windows コンピュータに ARM 用 IAR Embedded Workbench<sup>®</sup>をインストールしてください。ARM 用 Embedded Workbench<sup>®</sup>は、プロジェクト・マネージャ、エディタ、コンパイラ、アセンブラ、リンカ、ライブラリ、デバッグツールを含む統合開発環境です。それは、最適化された C コンパイラを含み、ARM デバイスとハードウェア・デバッグ・システムの広範囲をサポートしています。既製のデバイスのコンフィグレーション・ファイル、フラッシュ・ローダーとサンプル・プロジェクトが含まれています。キットのインストール CD は、IAR Embedded Workbench<sup>®</sup> の KickStart 版です。Embedded Workbench<sup>®</sup>用のライセンス・ファイルを取得するには、IAR の Web サイトで登録する必要があります。ライセンスは永久です；その唯一の制限は、プログラム・サイズです。コンパイルされたプログラムが 32K バイトを超える場合は、コンソール I/O オプション（プロジェクト・ファイルの 613x\_config.h）を無効にしてみてください。これにより、MIL-STD-1553 の機能を損なうことなく、コンパイルされたプログラム・サイズを小さくすることができます。そうで無い場合は、ARM 用の Embedded Workbench<sup>®</sup>の無制限版の 30 日間の評価ライセンスを IAR の Web サイト [www.iar.com](http://www.iar.com) から入手可能です。インストールの詳細については、IAR の「リリース・ノート」の「重要な情報」を参照してください。

**注意**：Windows Vista および 7 は、UAC(user Access Control)と呼ばれる、新しいセキュリティ・メカニズムを持ちます。UAC を無効にすると、Vista/7 はより XP のように動作します。これは、IAR Embedded Workbench<sup>®</sup>を管理者権限で実行する必要があります。これは、デモ・プロジェクトが「Program Files ¥ ... ¥」にインストールされている場合に重要でしたが、デモ・プログラムが「マイドキュメント」パスにインストールされている場合は必要ありません。これは参考用です。

UAC を無効にするには、つぎの手順を実行します。

- 1) コントロールパネル → ユーザーアカウントと家族のための安全設定
  - 2) ユーザーアカウント → ユーザーアカウント制御設定の変更
  - 3) 制御設定を → 通知しない
  - 4) 再起動
2. コンソール I/O（初期評価のために推奨）を使用する場合、コンピュータは、シリアル（COM）ポートと、Tera Term などのターミナル・エミュレーション・プログラムが必要です。ほとんどのデスクトップ・コンピュータでは、COM ポートを持ちますが、多くのノートブック・コンピュータでは、COM ポートを持っていません。そのため、シリアル-USB アダプタが別途必要になります。

Windows2000 または XP を使用している場合は、ターミナル・エミュレーション用のハイパーターミナルを使用できます。「スタート」→「すべてのプログラム」→「アクセサリ」を開き、ハイパーターミナルをクリックし、起動します。次の段落はスキップします。

Vista や Windows7 を使用している場合

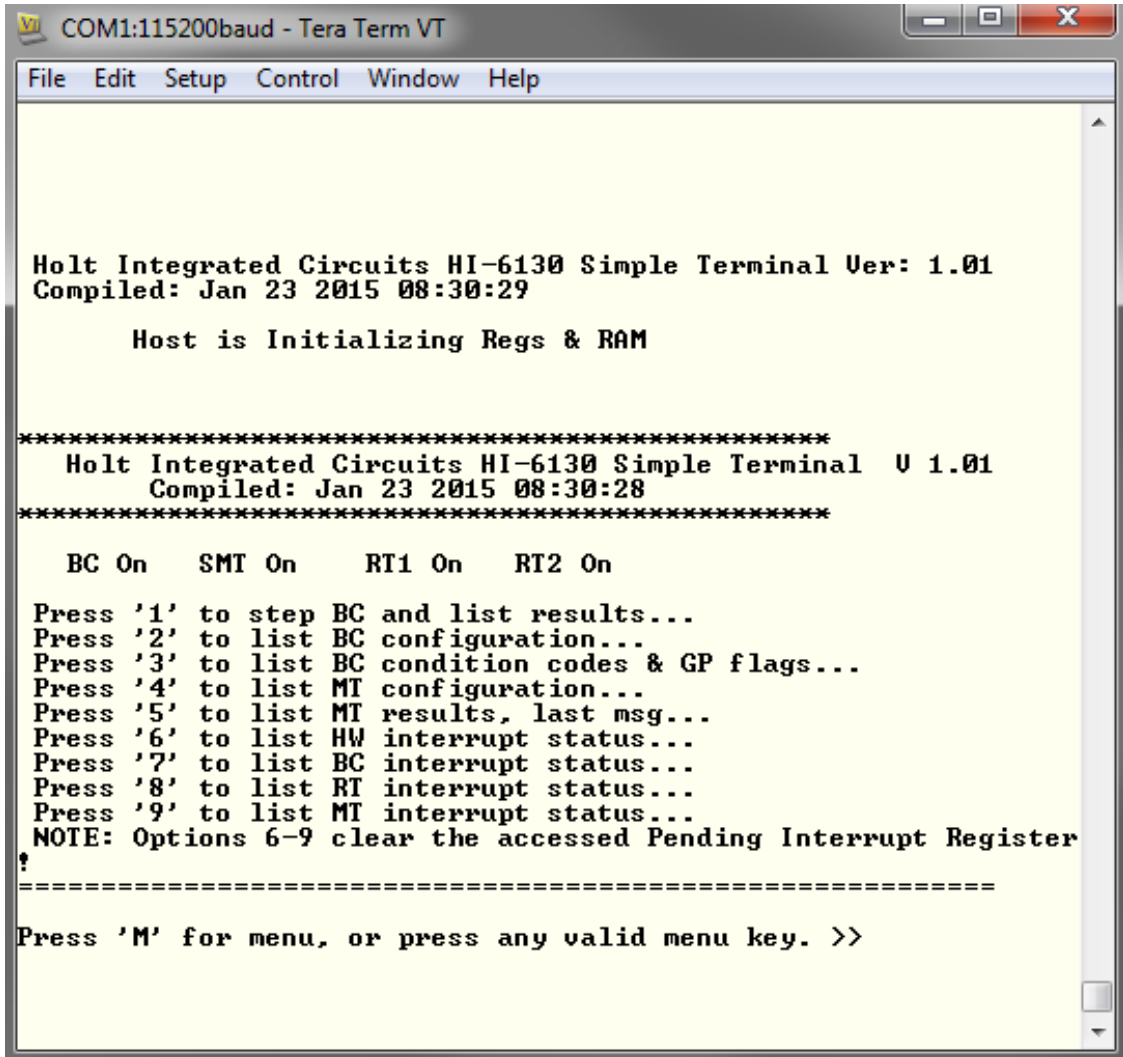
ハイパーターミナルは、これらのバージョンの Windows には含まれていません。無料のオープンソース・ターミナル・エミュレーション・プログラム TeraTerm4.71 を、Holt 提供の CD 内にある、「teraterm-4.71.exe」インストール・プログラムを実行して、インストールしてください。再配布は、著作権表示が保持されることを条件に許可された旨のライセンス契約を受け入れます。通知は、「Help」→「About TeraTerm」をクリックすることで TeraTerm ウィンドウから表示することができます。インストールを継続します・・・

- 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. 下段の回路基板上の電源入力ジャックに、5VDC アダプタケーブルを接続します。TeraTerm が実行され、正しく設定されている場合、以下のコマンド・メニューがコンソール・ウィンドウに表示されます。このメニューは、ボードの電源が入るたび、RESET ボタンが押されるたびに表示されます。評価ボードとの正しい TeraTerm で通信を確認した後、ターミナルの設定は、「設定」→「設定の保存」をクリックして保存することができます。





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

Holt Integrated Circuits HI-6130 Simple Terminal Ver: 1.01
Compiled: Jan 23 2015 08:30:29

Host is Initializing Regs & RAM

*****
Holt Integrated Circuits HI-6130 Simple Terminal U 1.01
Compiled: Jan 23 2015 08:30:28
*****

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 '9' to list MT interrupt status...
NOTE: Options 6-9 clear the accessed Pending Interrupt Register
?
=====
Press 'M' for menu, or press any valid menu key. >>
```

以下は、前のメニューからキーボードで「1」を3回押した後のコンソール画面を示しています。BCはCW=0x0BC0（コマンド・ワード）を送信し、送信されたデータを示します。Simple RTは、このコマンドとデータを受信し、トラフィック・メッセージ・タイプ、MIW、TimeTag、サブアドレス、および受信したデータ・ワードの数を表示します。HI-6130 BUSA 出力コネクタは、コマンドとデータを確実に受信するために、75Ω 終端抵抗で終端するか、Simple RT用のMIL-STD-1553バスカプラに接続する必要があります。

```

COM1:115200baud - Tera Term VT
File Edit Setup Control Window Help
Press 'M' for menu, or press any valid menu key. >>
Simple RT TX Polled Message:
CW: 1FC0 MIW: 0020 TimeTag: 29E9 SA: 30
0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
>
Results From Last Message Issued by BC
=====
Message Type: Rx Subaddress Command, 32 data words
CW: 0x1BC0 = 03-0-30-00 SW: 0x1800 = RT03 CS
BC Control Word: 0x40A0
MEmask UseBusA maskBCR 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, Bus Addr: 0x6000A650
Data:
0x0101 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. >>
Simple RT RX Polled Message:
CW: 1BC0 MIW: 0000 TimeTag: 750B SA: 30
0101 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
>

```

- デバッグでは、IAR Embedded Workbench<sup>®</sup>および、HI-6130 アプリケーション開発キットを実行するコンピュータ間のインターフェイスが必要です。HI-6130 評価ボード USB コネクタの「DEBUG」マークがある側に、付属の USB ケーブルを接続します。もう一方をコンピュータの USB ポートに接続します。ARM 用 IAR C-SPY デバッガは、「オンボード J-link」に組込まれる、多数のターゲット・システムのインターフェイス用ドライバが含まれています。

評価ボードの USB ケーブルが初めてコンピュータに接続されると、Windows は「新しいハードウェア検出」メッセージを表示し、J-Link デバイスを検出します。数秒後、Windows は、適切なドライバをロードして、「ハードウェアを使用する準備ができました」のメッセージを表示します。Windows が J-Link のドライバを見つけるのに失敗した場合は、IAR Embedded Workbench<sup>®</sup>のインストール CD の Driver ディレクトリを検索するようにしてください。

手順 11 のデバックセッションを開始するときに、問題が発生した場合、「プロジェクト」→「オプション」をクリックします。表示されたウィンドウ内の、カテゴリ: デバッガハイライト J-Link/J-Trace で、コミュニケーション=USB、インターフェイス=SWD を確認してください。

5. 提供された Holt CD には、ANSI C で書かれた MIL-STD-1553 のサンプル・プロジェクトが 1 つ以上含まれています。CD の PROJECTS ディレクトリを調べて、どのプロジェクトが自分の要件に最も適しているかを判断します。プロジェクト・フォルダ全体（またはサブ・ディレクトリ）は、コンピュータのハードドライブ上の特定の場所にコピーする必要があります。
6. 付属の IAR CD-ROM から IAR Embedded Workbench<sup>®</sup>をインストールします。ドキュメント「Holt Demo Project Installation for IAR Systems」を参照してください。インストールが完了し、少なくとも 1 つの Atmel サンプル・プロジェクトがインストールされたら、Holt zip プロジェクトを PC のどこかにコピーし、2 つのプロジェクトを解凍します。
7. \_Holt プロジェクト・フォルダ全体をこの場所に移動します :  
 C:¥My Document¥IAR embedded workbench¥Atmel¥atmel¥at91sam3u-ek  
以下の古い場所には設置しないでください :  
 C:¥Program Files¥IAR Systems¥Embedded Workbench 6.xx¥arm¥examples¥Atmel¥at91sam3u-ek
8. IAR Embedded Workbench<sup>®</sup>を開きます。[File] - [Open Workspace]をクリックし、手順 5 で作成したプロジェクト・サブディレクトリに移動します。拡張子.EWW のプロジェクト・ファイルを選択し、[Open]をクリックします。（次回 Embedded Workbench<sup>®</sup>を開くと、ファイルがクリックされると、このプロジェクトが最近のワークスペース・リストに表示されます）。
9. IAR インストールまたは IAR デバッガを使用して問題が発生した場合、Holt CD ROM に含まれるこれらの問題の解決に役立つ 2 つの Holt テクニカル・ノートが提供されています。
10. HI-6130 プロジェクトは、符号なし整数変数のみを使用します。変数の最上位ビットがトグルするときに発生する迷惑なコンパイラ・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 :  
 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
11. Holt の IAR プロジェクトはワークスペースのプルダウン・メニューから選択することができ、7 つの事前定義されたコンフィグレーションを含みます。コンフィグレーションはプリプロセッサ・ラベル、BC\_ena、RT1\_ena、RT2\_ena、SMT\_ena、IMT\_ena を変更します。これらはビルドをカスタマイズするためにコンパイラによってコンパイル時に使用されるラベルです。

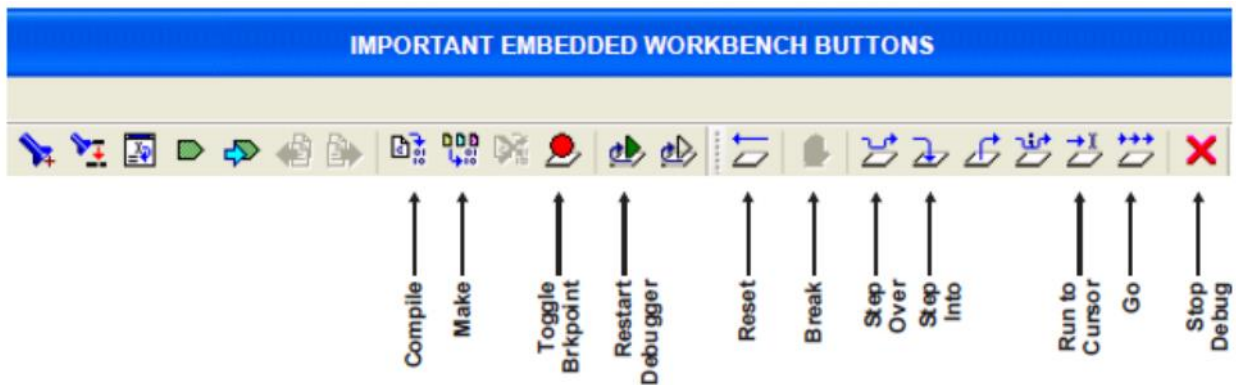
デフォルト設定の BC\_MT\_RT は HI-613x デバイスのプライマリ・モード BC、MT、RT1、RT2 を可能にします。これはデモボードにプログラムされたデフォルトの設定です。これは、BC、RT1、RT2、MT を有効にし、デモンストレーションを行います。これらの設定はすべてフラッシュ・ベースのプロジェクトです。RAM ベースのプロジェクトは MCU の RAM 制限により対応していません。設計上 Cortex-M3 は RAM ベースのプロジェクトに比べフラッシュでは動作速度が遅いです。(By design the Cortex™ - M3 runs slower in RAM than in Flash so there is little need for a RAM based project.)。7 つコンフィグレーションと対応するプリプロセッサ・ラベル値を以下の表に示します。

構成	BC_ena	RT1_ena	RT2_ena	SMT_ena	IMT_ena
BC_MT_RT (default)	1	1	1	1	0
BC_ONLY	1	0	0	0	0
SMT_ONLY	0	0	0	1	0
RT_ONLY	0	1	1	0	0
IMT_ONLY	0	0	0	0	1
RT_SMT	0	1	1	1	0
RT_IMT	0	1	1	0	1

他のコンフィグレーションは、SMT\_ena と IMT\_ena の両方(1に設定)を有効にすることはできませんので、モニタを一度に一種類しか選択できないこと意外は可能です。新しいコンフィグレーションを作成する簡単な方法は、Project/Edit を選択肢、New を選択することです。ダイアログ・ボックスに、新しいコンフィグレーション名をもつ既存の設定に基づいた、新しい設定が可能になります。新しいコンフィグレーションを設定して、必要に応じてプリプロセッサのラベルを編集します。新しい設定は、プルダウン・メニューに表示されます。

デフォルトから設定を変更する場合、ボード上の4つの端子のDIPスイッチに対応した、BCENA、MTRUN、RT1ENA、RT2ENA とが一致するように設定する必要があります。これらのいずれかが一致しない場合、ボードに最初に電源が投入されたときにコンソールにエラーが表示されます。

12. プロジェクト・ファイル「613x\_initialization.h」はタイムタグ分解能およびコンソール I/O の ON/OFF を含む重要なプロジェクト設定を構成します。
13. 「Make」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。次の図を参照してください。IAR Embedded Workbench®の「Message」ウィンドウに、エラーまたは警告が無かった場合、作業を続行できます。エラーが発生した場合は、それらを修正し、プログラムを再コンパイルします。zip 形式の Holt プロジェクトは、\*.O および\*.lst ファイルを除外することができます。これによってファイルサイズが大幅に縮小され、電子メールで提供することがはるかに簡単になります。
14. 「Restart Debugger」ボタンをクリックして、デバックセッションを開始します。これは、プログラム実行のためのコンパイルされたプログラムを MCU と readies ボードにダウンロードします。実行を開始するには、「GO」をクリックします。



15. 新しい設定を自動初期化のシリアル EEPROM にプログラムする :

- 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する代わりに、AUTOEN ラベルの付いた DIP スイッチを OFF にし、HI-6130 の初期化を MCU からするように設定します。COPYREQ ラベルの付いた DIP スイッチを ON にし、HI-6130 のレジスタおよび RAM のポストラセット初期化後、EEPROM コピー・シーケンスを開始するために MCU を設定します。
- 実行が開始されると、赤色の LED が EEPROM のコピー処理中に点灯します。それが OFF になったら、AUTOEN ラベルの付いた DIP スイッチを ON にして、自動初期化を有効にします。リセット時に、EEPROM 書き換えを防止する、COPYREQ DIP スイッチを OFF にします。

16. バスのアクティビティを観察するには、赤色の BUS A と赤色の BUS B のテスト・ピンにオシロスコープを接続してください。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す。MIL-STD-1553 バスにケーブルが接続されていない場合、バスの赤と黒のテスト・ポイントの各ペア間に、70Ω 1W の抵抗を接続することで、A、B のためのダミー負荷を提供しています。

## Project File List with Selected Descriptions [プロジェクト・ファイルリスト]

### HEADER FILES WITHOUT CORRESPONDING C FILES

#### **613x\_initialization.h**

重要なコンフィグレーション設定の定義

#### **device\_6130.h**

ARM MCU 外部バス・インターフェイスの定義

レジスタ・アドレッシング用構造体(HI-6130 のみ)

#### **613x\_regs.h**

レジスタ・ビットおよびビット・フィールドのマクロ

### C FILES WITH CORRESPONDING HEADER FILES

Most of the function names are self - explanatory.

#### **main.c**

---

*main(void);*

プライマリ・プログラム・エン트리・ポータル *main()*は、有効にされたターミナルの初期化シーケンスを示しています。EEPROM からの自己初期化を有効にするかどうかにかかわらず、有効化されたターミナルの任意の組み合わせに対して、使用されるシーケンスを示します。初期化が完了した後、関数呼び出しは、有効なターミナル・モードで使用されるすべての RAM 構造に対して強力なアドレッシング方法を提供します。

#### ***board\_613x.c***

*ConfigureGpio(void);*

この関数は、ARM MCU 汎用 I/O を初期化します

*reset\_613x(void);*

*char autoinit\_check(void);*

*initialize\_613x\_shared(void);*

*init\_timer(void);*

*Delay\_us(unsigned short int num\_us);*

*Delay\_ms(unsigned short int num\_ms);*

*Delay\_Nx100ms(char num);*

*error\_trap(char count);*

#### ***board\_6130.c***

*Configure\_6130\_EBI(void);*

この関数は、ARM MCU 外部バス・インターフェイスを初期化します

#### ***613x\_BC.c***

*void BC\_bus\_addressing\_examples(void);*

*initialize\_bc\_msg\_blocks(void);*

*initialize\_bc\_instruction\_list(void);*

*initialize\_613x\_BC(void);*

*bc\_disable(void);*

*bc\_enable(void);*

*bc\_start(void);*

*bc\_trigger(void);*

*SW1\_BC\_Trig\_Test(void);*

デモの場合、この関数はプッシュ・ボタン SW1 をポーリングし、次の BC メッセージをトリガします

#### ***613x\_MT.c***

*initialize\_613x\_MT(void);*

この関数は、Simple または IRIG-106 (SMT または IMT) モニタ動作を初期化します

#### ***613x\_RT.c***

*initialize\_613x\_RT1(void);*

*initialize\_613x\_RT2(void);*

*RTAddr\_okay(char RTnum);*

*modify\_RT\_status\_bits(void);*

*write\_dummy\_tx\_data\_RT1(void);*

---

```
write_dummy_tx_data_RT2(void);
```

最後の2つの関数は、デモのRT1とRT2の送信データ・バッファを初期化します

### **console.c**

全てのターミナル・モードで使用されるコンソール機能 :

```
void ConfigureUsart1(void);
```

```
void text_header(void);
```

```
void chk_key_input(void);
```

```
unsigned short int list_hw_ints_console(void);
```

BCモードで使用されるコンソール機能 :

```
void bc_last_msg_console(void);
```

```
void list_bc_config(void);
```

```
void list_bc_ccgpf_reg(void);
```

```
unsigned short int list_bc_ints_console(void);
```

RT1 および/または RT2 で使用されるコンソール機能 :

```
unsigned short int list_rt_ints_console(void);
```

SMT または IMT バスモニタ・モードで使用されるコンソール機能 :

```
void list_mt_config(void);
```

```
void mt_last_msg_console(void);
```

```
unsigned short int list_mt_ints_console(void);
```

冗長な文字列を「printf」してプログラム・サイズを縮小するプリミティブ・コンソール機能 :

```
void print_null(void);
```

```
void print_sp1sp(void);
```

```
void print_b1sp(void);
```

```
void print_b0sp(void);
```

```
void print_dddn(void);
```

```
void print_dd0n(void);
```

```
void print_dd1n(void);
```

```
void print_menuprompt(void);
```

```
void print_line(void);
```

## **Application Development Kit Notes [アプリケーション開発キット注意点]**

最新のHI-6130データシートは [www.holt.com](http://www.holt.com) にあります。主要なプロジェクト構成設定は、613x\_initialization.h ファイルに記載されています。

HI-6130 は、外部バスコントローラ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ました。RAM とレジスタの位置は、MCU ボード上の ARM Cortex 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。利用する MCU チップ・セレクト出力（HI-6130 チップイネーブル入力端子に接続されている）は、MCU のバスアドレス 0x60000000 から始まるメモリ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 およびレジスタ処理は 16 ビット値の取引なので、すべてのアドレスバスは偶数です。バイトアドレッシングを使用するには、HI-6130 の RAM またはレジスタ・アドレスは倍増し、MCU のチップ・セレクトのベースアドレスに追加されます。したがって、HI-6130 レジスタ 0 は、MCU のバスアドレス 0x60000000 でアクセスされます。レジスタ 1 は、バスアドレス 0x60000002 で、レジスタ 5 は、バスアドレス 0x6000000A でアクセスされます。

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され、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、読み込み発生後リセットされます。これは、ペンディング割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ます。

Address	Value	Value	Value	Value	Value	Value	Value	Value	Value
5fffffff0									
60000000	11f2	8100	0000	0000	0000	0000	0000	0000	0000
600000010	0000	0000	0180	0000	0000	0000	0000	0000	7818
600000020	fff8	01f8	d5a8	6018	fbf8	01f8	d5a8	01ce	
600000030	1c00	0400	0000	0000	aaaa	bbbb	0000	abcd	
600000040	01ce	2000	0600	0000	0000	aaaa	bbbb	0000	
600000050	abcd	4377	0000	0000	0000	0000	0000	00b0	
600000060	0000	0000	d02d	1b70	1b70	0000	0000	8000	
600000070	00c0	007f	1203	5bf8	0000	0000	0000	0000	
600000080	0000	0000	0000	1221	5bf8	0000	0000	0000	
600000090	0000	1244	0000	1245	0000	0000	0000	0000	
6000000a0	0000	0000	0000	0000	0000	0000	0000	0000	
6000000b0	0000	0000	0000	0000	0000	0000	0000	0000	
6000000c0	0000	0000	0000	0000	0000	0000	0000	0000	
6000000d0	0000	0000	0000	0000	0000	0000	0000	0000	
6000000e0	0000	0000	0000	0000	0000	0000	0000	0000	
6000000f0	0000	0000	0000	0000	0000	0000	0000	0000	

レジスタ空間を示すデバ  
グメモリウィンドウ  
赤色は、値の変化を表します

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され、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がされている 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。読み込みが発生した後、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ます。



```

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 BCil pBCil = 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;
ScSTK pScSTK = (ScSTK) ScSTK_BASE_BUS_ADDR;
#endif // (SMT_ena)

```

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
pBCil	0x600036E0	0x00089964	BCil
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、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-bit Unsigned>
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	0xFFFF	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	0xFFB8	0x60000028	HI613
MT_INT_OUTPUT_ENABLE_REG	0x01F8	0x6000002A	HI613
RT_INT_OUTPUT_ENABLE_REG	0xD5A8	0x6000002C	HI613

コンソール I/O オプションは、TeraTerm を使用して、ペンディング割込みレジスタステータスを読み取り表示する、いくつかのメニュー・オプションを含みます。ペンディング割込みビットは自動的に読み込みが発生した後にリセットされることに注意してください。これらのレジスタは、メモリ・ウォッチ・ウィンドウは、実行が停止したことで値を反映しています。

HI-6130 回路基板は、自動初期化用の 2 つの別々の EEPROM を持っています。上段のボードのスライドスイッチ SW2 で、出荷時にプログラム済みの DEMO EEPROM もしくは、未プログラムの USER EEPROM が選択可能です。SW2 を使用してユーザー EEPROM を選択することで、DEMO EEPROM のデモ・プログラムを上書きすることなくプロジェクトのビルドプログラムを試すことができます。

IRIG-106 (IMT) バスモニタをデバッグするときは、「最大レコーディング時間」の上限に達することによるパケットの終了処理を防ぐために、タイムタグロックを無効にすることが有効です。例えば、BC は、新しい MIL-STD-1553 コマンドを発行します。デバッガメモリまたは、ウォッチ・ウィンドウを使用すると、次のコマンドを発行する前に BC にトリガをかけ、保存されたメッセージデータを調べられます。タイムタグロックが動作している状態で、ユーザーは、最大パケットレコーディング時間を越えることなく保存されたメッセージのデータを調べる方法はありません。タイムタグロックを無効にすると、一連のメッセージが個別に発行され、パケットファイナライズせずに確認することができます。

## Choosing HI-6130 or HI-6131 [HI-6130 と HI-6131 の選択]

### トレードオフ

ホスト MCU または FPGA に対し、HI-6130 は 16Bit パラレル・バス・インターフェイスを持ちます。それは、100 ピン・プラスチック QFP パッケージで、16×16mm です。HI-6130 の信号インターフェイスは、バス制御信号とチップ・セレクトを含め合計 36 ピンです。MCU または FPGA の外部バス・インターフェイスは、ピン数、パッケージ・サイズ、基板回路上での信号のルーティング・エリアが増加します。

一方、ホスト MCU または FPGA に対し、HI-6131 は 4 線式シリアル・ペリフェラル・インターフェイス (SPI) を持ちます。それは 12×12 mm のプラスチック QFP、または 9×9 mm のチップ・スケール・パッケージで提供されます。接続に 32 より少ない信号数を有するので、回路基板面積は、HI-6130 に必要な面積よりも小さいです。デバイスの選定に単に信号ルーティングのためのピン数や、回路基板面積を優先する場合は、HI-6131 は明らかに勝者ですが、パフォーマンスの違いを考慮せずにあわてて結論を出すことはありません。

HI-6130 バス・インターフェイスは、150ns のサイクル・タイムで、一度に一つのワードをリード/ライトします。リード/ライトはランダム・アクセスであり、任意の順序で発生する可能性があります。シーケンシャル・アドレスから、マルチ・ワード・リード・シーケンスをスピードアップするために、HI-6130 は、シングル・ワード・リード・サイクル後に次の RAM またはレジスタ位置を自動的にプリフェッチします。シーケンシャル・アドレスを読み込むとき、デバイスは、最初のデータ位置をフェッチ中に WAIT をアサートし、その後、すべてのシーケンシャル・アドレス (プリフェッチ) リードをします。WAIT をアサートせずに実行した方が早くなります。

HI-6131 のデータ転送速度は、MCU が SPI インターフェイスに供給する SPI クロック周波数に依存します。SPI は、SCK 周波数が最大 20MHz でクロックされている場合、各ワードは 800ns で転送され、さらに、SPI オプションコードによりオーバーヘッドが追加されます。多くの MCU の最大 SCK 周波数は、15MHz で、その結果、16Bit ワードの転送時間で 1,067ns、さらに、OP コード実行のオーバーヘッドが追加されます。

HI-6130 のメモリ・アドレス・ポインタ (レジスタ) は、リード/ライト動作が始まる前に、MCU または FPGA により初期化されます。リード/ライト動作は、8Bit の SPI OP コードを使用して開始され、MCU または FPGA によりシリアルに HI-6130 SPI にシフトインされます。ホストは、連続した RAM またはレジスタ・アドレスに連続したリード/ライトするために 16 クロック倍数で SCK クロックを継続します。クロックが続く限り連続したアドレスがリード/ライトされます。割り込みがこの処理の間に有効になっている場合、潜在的な問題が発生します。プログラムの割り込みハンドラが割り込みを処理するために SPI バスを補足した場合、潜在的に未完成のマルチ・ワード転送が中断されます。適切なソフトウェア設計が無ければ、壊れたマルチ・ワード転送における、単純な復帰割り込みの結果は、ハードウェアは割り込みが発生したことを知りません、そして、中断されたマルチ・ワード転送のための次の RAM またはレジスタ・アドレスを含むメモリ・アドレス・ポインタが無い場合があります。

HI-6131 の SPI 転送中に、割り込みを無効にする必要があります。最も単純な実装では、OP コードを送信する前に割り込みを無効にし、マルチ・ワード転送の最後のワードをリード/ライトした後で、割り込みを再度有効にします。これが許容できない割り込みの遅延時間が発生する場合、ソフトウェア設計

にはいくつかの注意点が必要です。適切な予防措置として、SPI ワードの間、割込みは一時的に再度有効にし、すぐに無効にすることができます。割込みが再度有効にされるとき、割込みオフ間隔中に発生した保留中の割り込みは直ちに認識されます。ペンディング割込みサービス・ルーチンが実行され、割込みから復帰はジャンプし、以下の割込み禁止の文を実行します。

HI-6130 と HI-6131 のどちらを選択するか最後の問題は、IAR Embedded Workbench<sup>®</sup>を使用する時のデバッグの容易さです。HI-6130 のレジスタと RAM は、メモリ・マップされており、デバッグのメモリウィンドウは、プログラムの実行が停止するたびに自動的に更新して、アドレス範囲を表示できます。更新されたデータ値は、赤色で表示されます。HI-6130 設計例では、C 言語の構造体を使用して BC、RT、モニタ、RAM テーブルやその他の構造のアドレッシングを。簡素化するため、ポインタ参照の C 構造体を定義しています。定義された C 構造体は、プログラムの実行が停止するたびに更新される、レジスタや RAM テーブル・データを観測するための、デバッガ・ウォッチ・ウィンドウを使用可能にします。

HI-6131 を使用する場合、SPI アクセス・テーブルに対応するための C 言語の構造体と互換性はありません。HI-6131 を使用するとき、HI-6130 デバッガのウォッチ・ウィンドウに相当するレジスタ／RAM の検査ツールは使用できません。ユーティリティ機能は、アドレス範囲を読取るために C 言語で記述する必要があり、そして、コンソール I/O または、他の表示手段を使って表示する必要があります。

#### まとめ

RAM やレジスタにアクセスするためにわずか 4 本のホスト・インターフェイス信号で行えるので、HI-6131 の SPI インターフェイスは、ハードウェア設計を簡素化できます。この利点は、割込みハンドラ・ソフトウェアを注意深く設計する必要があることと、デバッグ時のパッケージ化されたレジスタ／RAM の検査ツールの欠如により、相殺されています。対照的に、HI-6130 バス・インターフェイスは、より高速なランダム・アクセス、プログラミングの容易さと強力なデバッガツールが用意されています。これらの利点は、RAM やレジスタをアクセスするための 36 インターフェイス信号を接続する必要性によって相殺されています。HI-6131 と比較すると、2 つの追加 16 ビット幅のバス（アドレスおよびデータ）は、HI-6130 とホスト MCU 間で接続する必要があります。

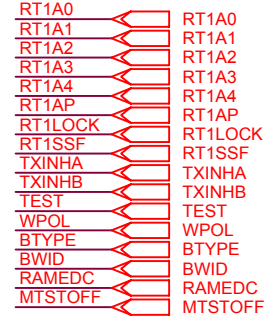
**評価ボードのブロック図および、部品表を以下のページに示します。**

**文書の最後に、RAM がどのように割り当てられているかのメモリ・マップを示します。**

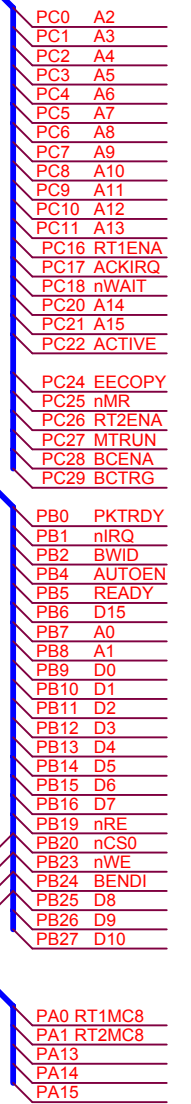
U1 EACH SHORT WIRE DENOTES CONNECTION SHARED WITH THE HOST MICROCONTROLLER, EITHER DIRECT MCU I/O, OR DIP SWITCH SIGNAL CONNECTED TO BOTH DEVICES.

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

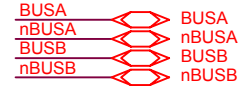
DIP SWITCH SIGNALS



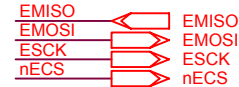
SIGNALS SHARED WITH MCU



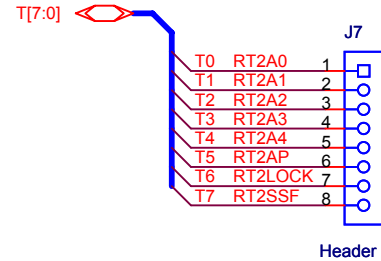
1553 BUS XFMR SIGNALS



SERIAL EEPROM SIGNALS



TEST BUS, MULTI-FCTN



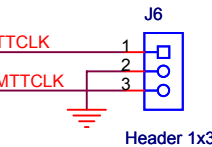
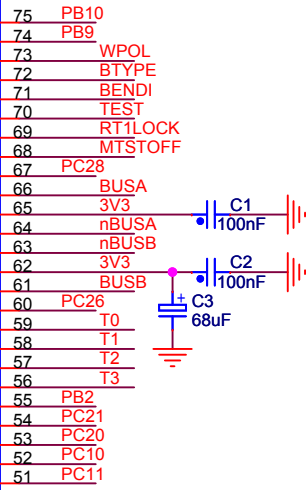
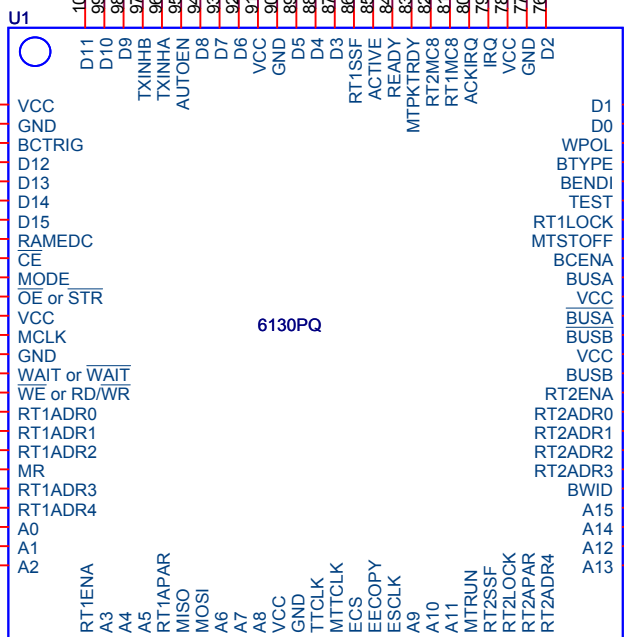
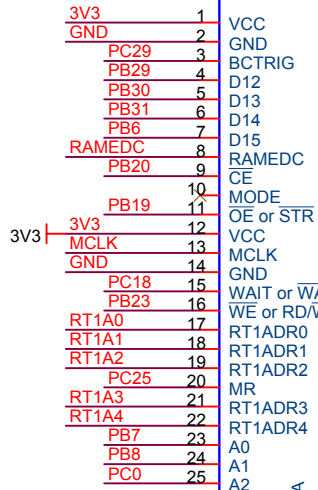
HI-6130 PQFF

HOLT INTEGRATED CIRCUITS, Mission Viejo, CA, USA

Title  
**HI-6130 EVAL PCB (Use With Std ARM CM3 Lower PCB)**

Size A Document Number 6130 EVAL TOP.DSN Rev A

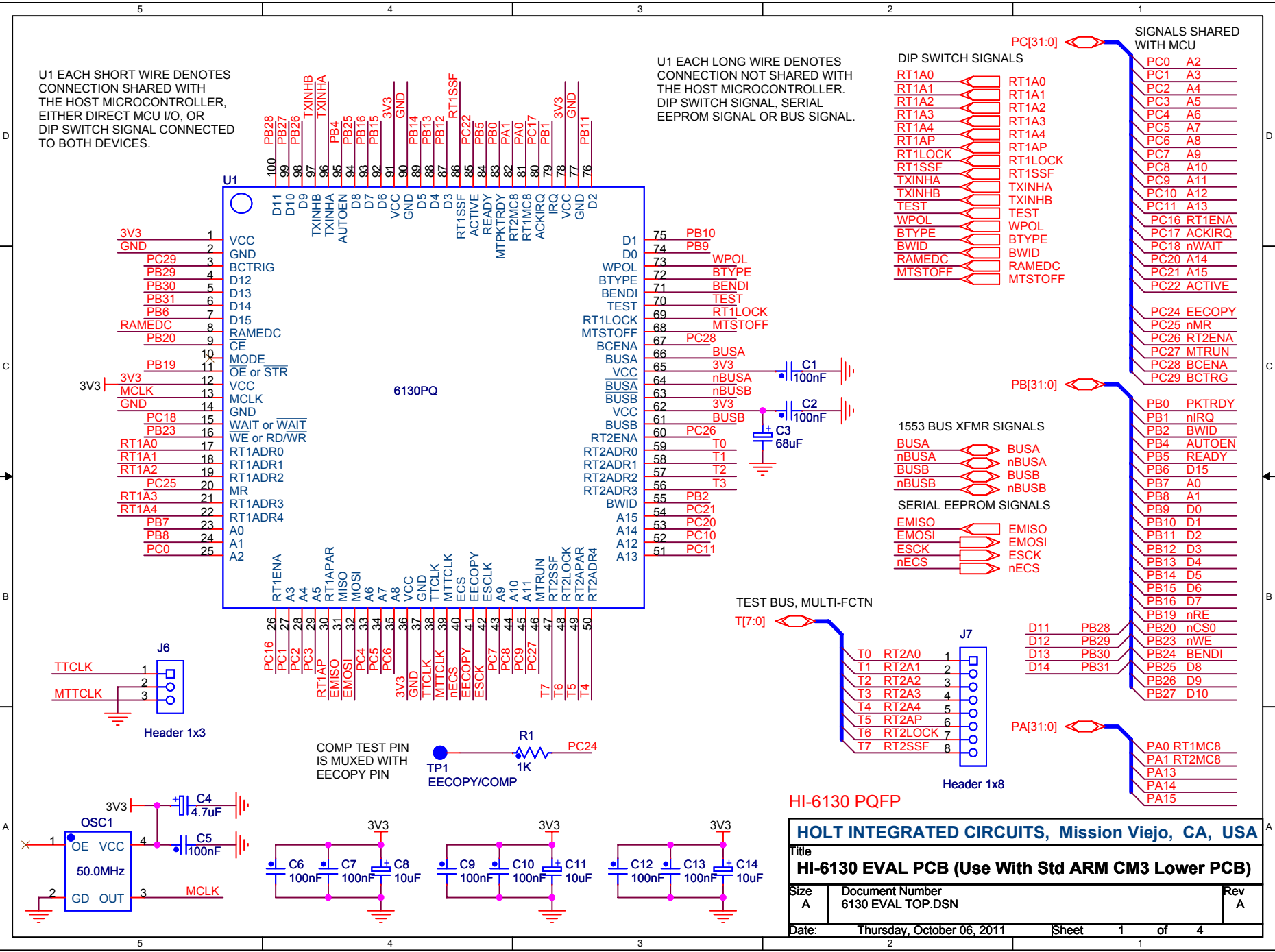
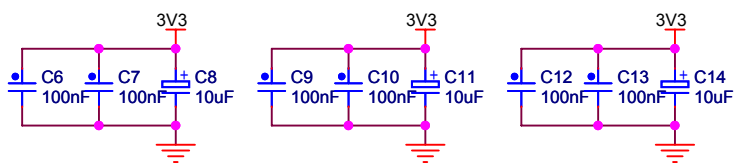
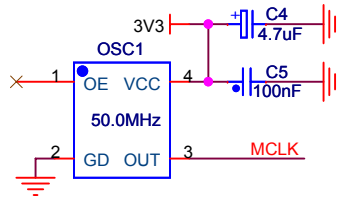
Date: Thursday, October 06, 2011 Sheet 1 of 4

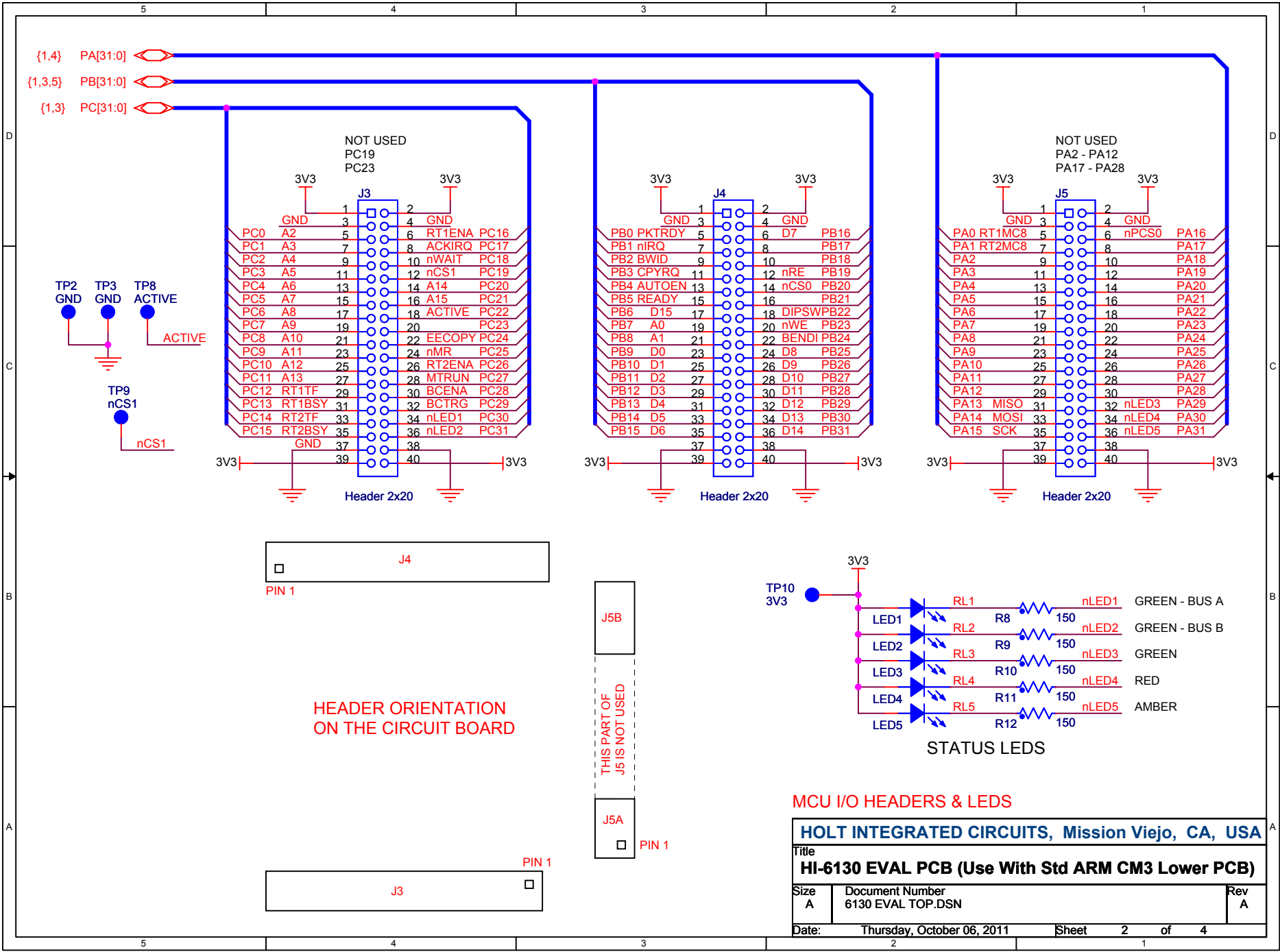


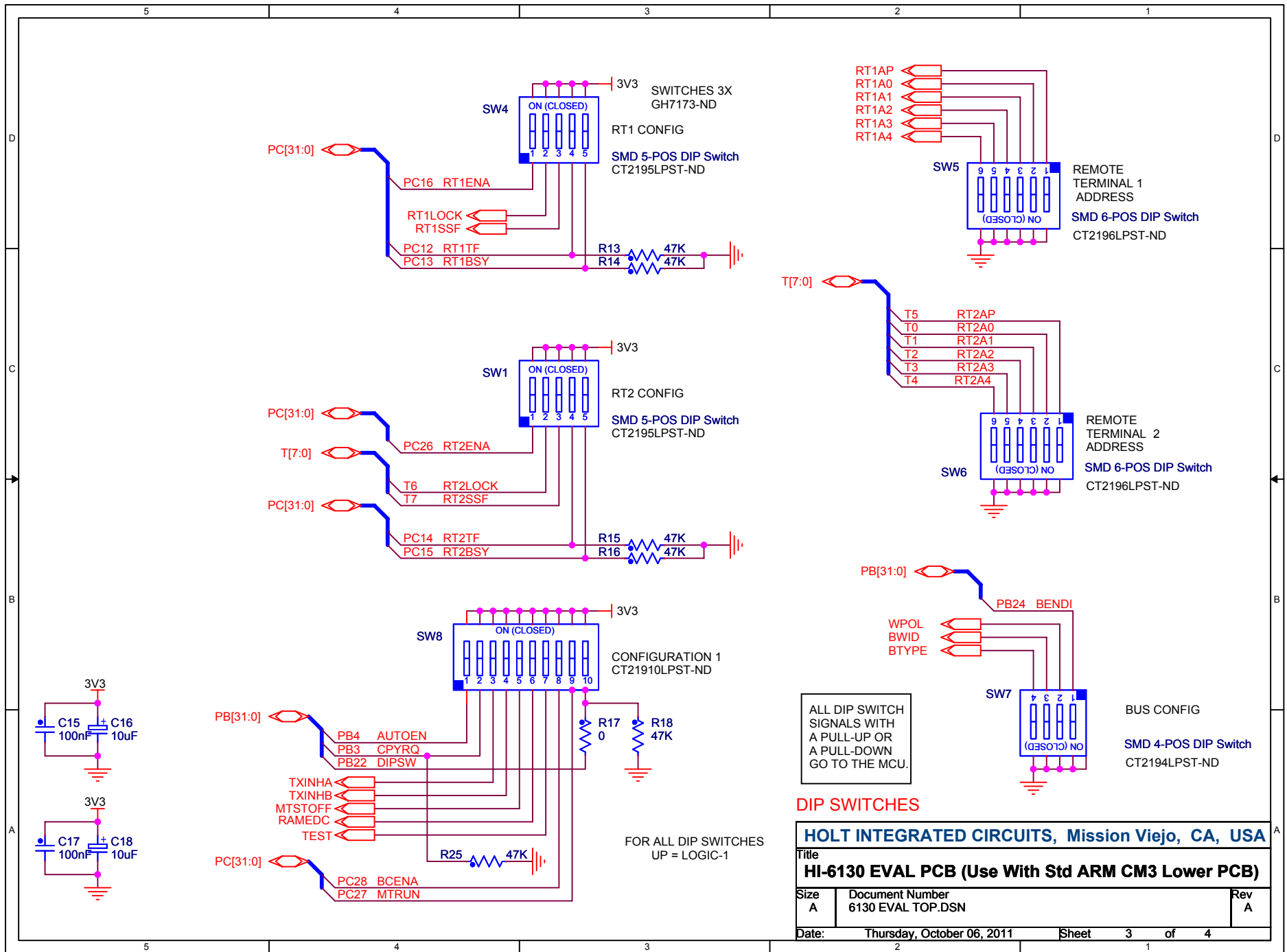
Header 1x3



COMP TEST PIN IS MUXED WITH EECOPY PIN



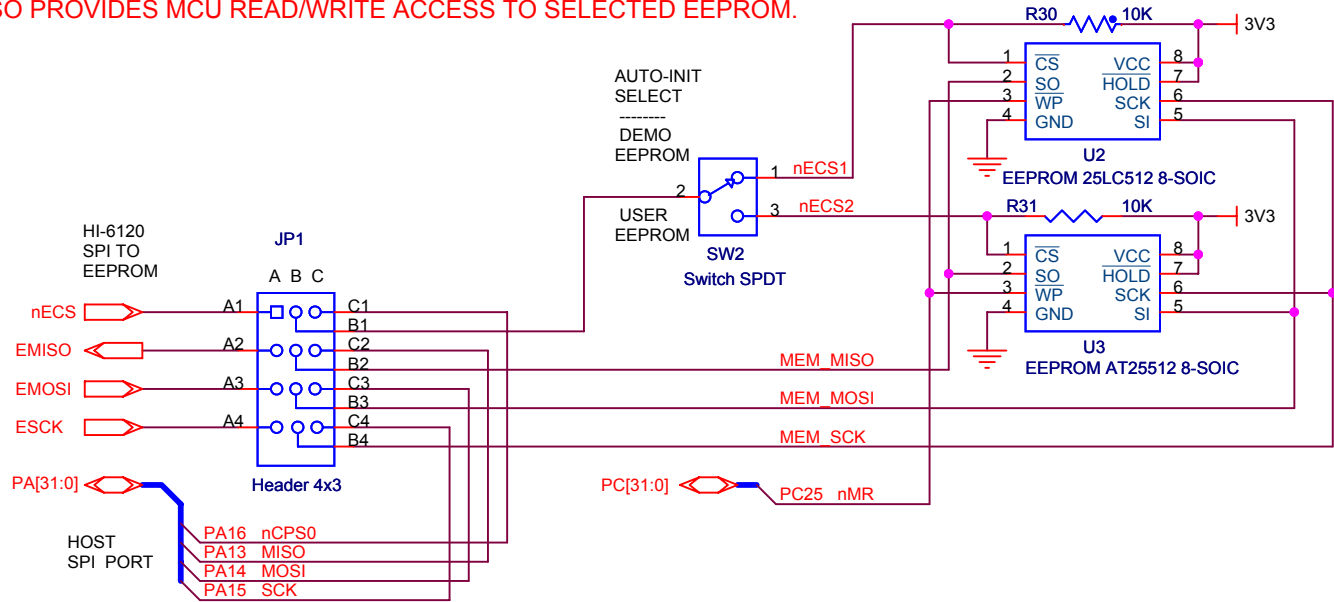




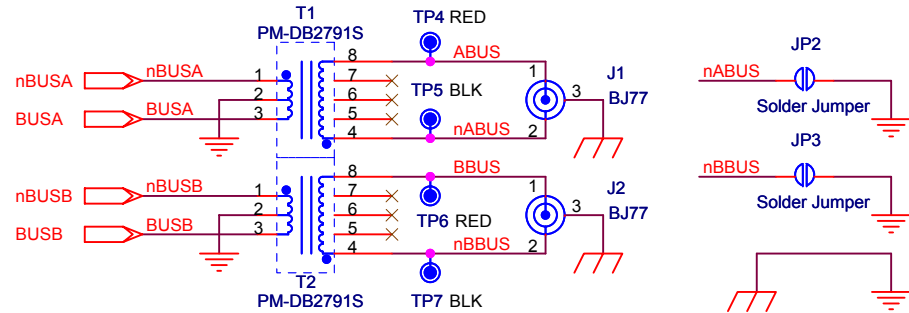
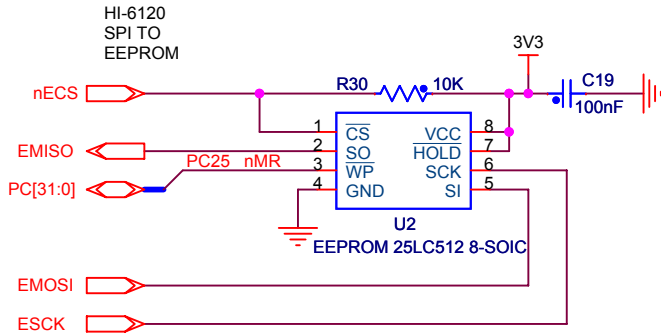
DUAL EEPROM CIRCUIT FOR EVALUATION BOARD ONLY. SW2 SELECTS EEPROM.  
 JP1 JUMPER ALSO PROVIDES MCU READ/WRITE ACCESS TO SELECTED EEPROM.

JUMPERS NORMALLY SPAN COLUMNS A-B FOR HI-6120 SPI TO EEPROM

FOR DIAGNOSTIC TESTS, JUMPERS SPAN COLUMNS B-C SO THE MCU CAN READ/WRITE THE SERIAL EEPROM.



TYPICAL APPLICATION REPLACES ABOVE EEPROM CIRCUIT WITH THIS SIMPLE CONFIGURATION



EEPROM, 1553 BUS

HOLT INTEGRATED CIRCUITS, Mission Viejo, CA, USA

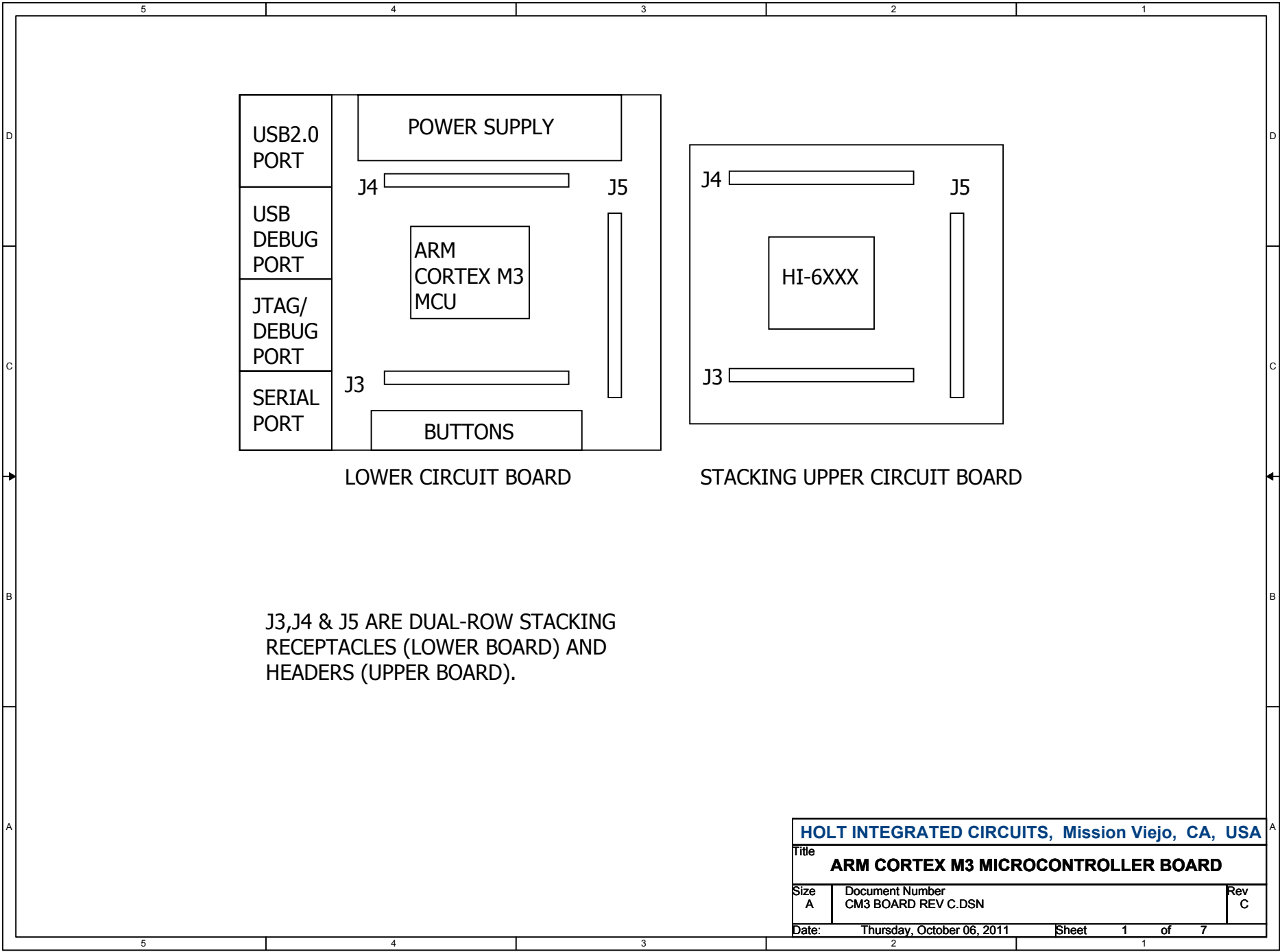
Title  
**HI-6130 EVAL PCB (Use With Std ARM CM3 Lower PCB)**

Size A	Document Number 6130 EVAL TOP.DSN	Rev A
-----------	--------------------------------------	----------

Date: Thursday, October 06, 2011 Sheet 4 of 4

Bill of Materials		HI-6130 Evaluation Board, Top			Revised: 9 Sept 2011
Item	Qty	Description	Reference	DigiKey	Mfr P/N
1	1	PCB, Bare, Eval Board	N/A	-----	-----
1	11	Capacitor, Ceramic 0.1uF 20% 50V Z5U 0805	C1,C2,C5, C6,C7,C9, C10,C12,C13, C15,C17	399-1176-1-ND	Kemet C0805C104M5UACTU
2	1	Capacitor, Ceramic 4.7uF 10% 6.3V X5R 0805	C4	399-3134-1-ND	Kemet C0805C475K9PACTU
3	5	Capacitor, Ceramic 10uF 10% 6.3V X5R 0805	C8,C11,C14, C16,C18	399-3138-1-ND	Kemet C0805C106K9PACTU
4	1	Capacitor 68uF 10% 6.3V Tantalum 400 mOhm SMD EIA 6032-28	C3	495-1507-1-ND	Kemet B45197A1686K309
5	2	Connector 3-Lug Concentric Triax Bayonet Jack, Panel Front Mount TRB (BJ77)	J1,J2	MilesTek 10-06570	Trompete Electronics BJ77
6	2	Header, Male 2x20 0.1" Pitch, 0.230" Pins, 0.120" Tails	J3,J4	S2012E-20-ND	Sullins
7	1	Header, Male 2x4 0.1" Pitch, 0.230" Pins, 0.120" Tails	J5A	S2012E-04-ND	Sullins
8	1	Header, Male 2x5 0.1" Pitch, 0.230" Pins, 0.120" Tails	J5B	S2012E-05-ND	Sullins
9	-----	Header, 1x3, 0.1" pitch	J6	DO NOT STUFF	-----
10	-----	Header, 1x8, 0.1" pitch	J7	DO NOT STUFF	-----
11	1	Header, 4x3, 0.1" pitch	JP1	Samtec	Samtec TSW-104-07-T-T
12	4	Jumper, shorting, w/ grip, 0.1"	JP1	S9341-ND	Sullins NPC02SXON-RC
13		Solder Jumper	JP2,JP3	DO NOT SOLDER	-----
14	1	LED Yellow 0805	LED5	160-1175-1-ND	Lite On LTST-C170YKT
15	3	LED Green 0805	LED1 - LED3	160-1179-1-ND	LiteOn LTST-C170GKT
16	1	LED Red 0805	LED4	160-1176-1-ND	LiteOn LTST-C170CKT
17	1	Osc, 50.00MHz 25ppm 3.3V SMD 5mm x 7mm	OSC1	CTX328LVCT-ND	CTX CB3LV-3I-64M0000-T
18	2	Resistor, 0 Ohm 0805	R17,R32	311-0ARCT-ND	Panasonic ERJ-6GEY0R00V
19	5	Resistor, 150 5% 1/8W 0805	R8,R9,R10, R11,R12	P150ACT-ND	Panasonic ERJ-6GEYJ151V
20	1	Resistor, 1.0K 5% 1/8W 0805	R1	P1.0KACT-ND	Panasonic ERJ-6GEYJ102V
21	2	Resistor, 10K 5% 1/8W 0805	R30,R31	P10KACT-ND	Panasonic ERJ-6GEYJ103V
22	6	Resistor, 47K 5% 1/8W 0805	R13,R14,R15 R16,R18,R25	P47KACT-ND	Panasonic ERJ-6GEYJ473V
23	1	DIP Switch 4-Position SMD	SW7	CT2194LPST-ND	CTS 219-4LPST
24	2	DIP Switch 5-Position SMD	SW1,SW4	CT2195LPST-ND	CTS 219-5LPST
25	2	DIP Switch 6-Position SMD	SW5,SW6	CT2196LPST-ND	CTS 219-6LPST
26	1	DIP Switch 10-Position SMD	SW8	CT21910LPST-ND	CTS 219-10LPST
27	1	Slide Switch SPDT SMD	SW2	563-1022-1-ND	Copal CJS-1200TB
28	2	Transformer MIL-STD-1553 Single, 1:2.50, PM-DB2791S	T1,T2	Holt PM-DB2791S	Holt / Premier Magnetics PM-DB2791S
29	-----	Test Point, pad w/ plated hole	TP1,TP9,TP10	-----	-----
30	2	Test Point, Red Insulator, 0.062" hole	TP4,TP6	5010K-ND	Keystone 5010
31	3	Test Point, Black Insulator, 0.062" hole	TP2,TP3,TP5, TP7	5011K-KD	Keystone 5011
32	1	Test Point, White Insulator, 0.062" hole	TP8	5012K-KD	Keystone 5012
33	1	IC HI-6130 Holt 100-PQFP	U1	-----	-----
34	2	IC, Serial EEPROM 512Kbit 20MHz SPI 8-SOIC, Microchip	U2, U3	25LC512T-I/SNCT-ND	Microchip 25LC512T-I/SN

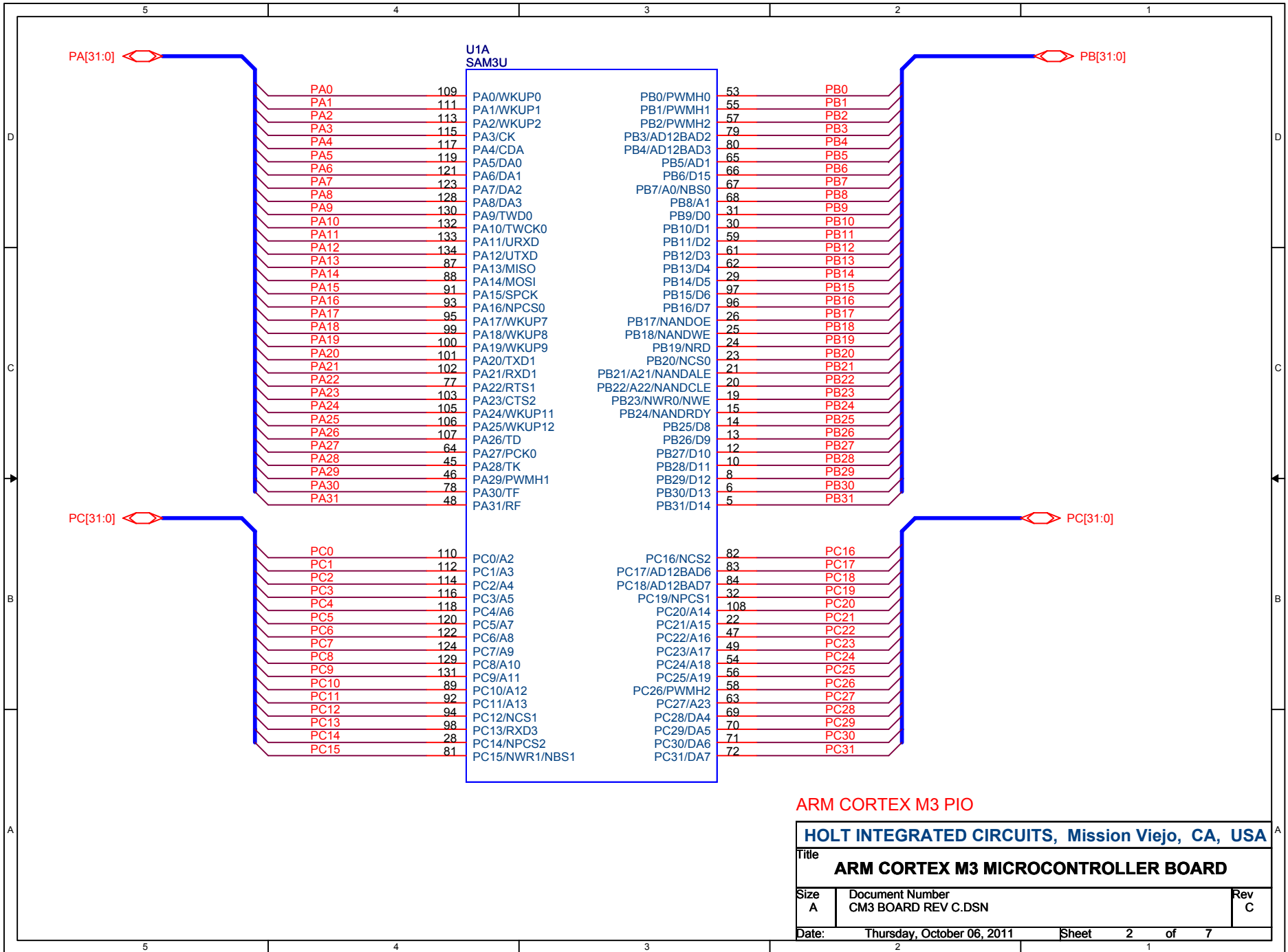




J3,J4 & J5 ARE DUAL-ROW STACKING RECEPTACLES (LOWER BOARD) AND HEADERS (UPPER BOARD).

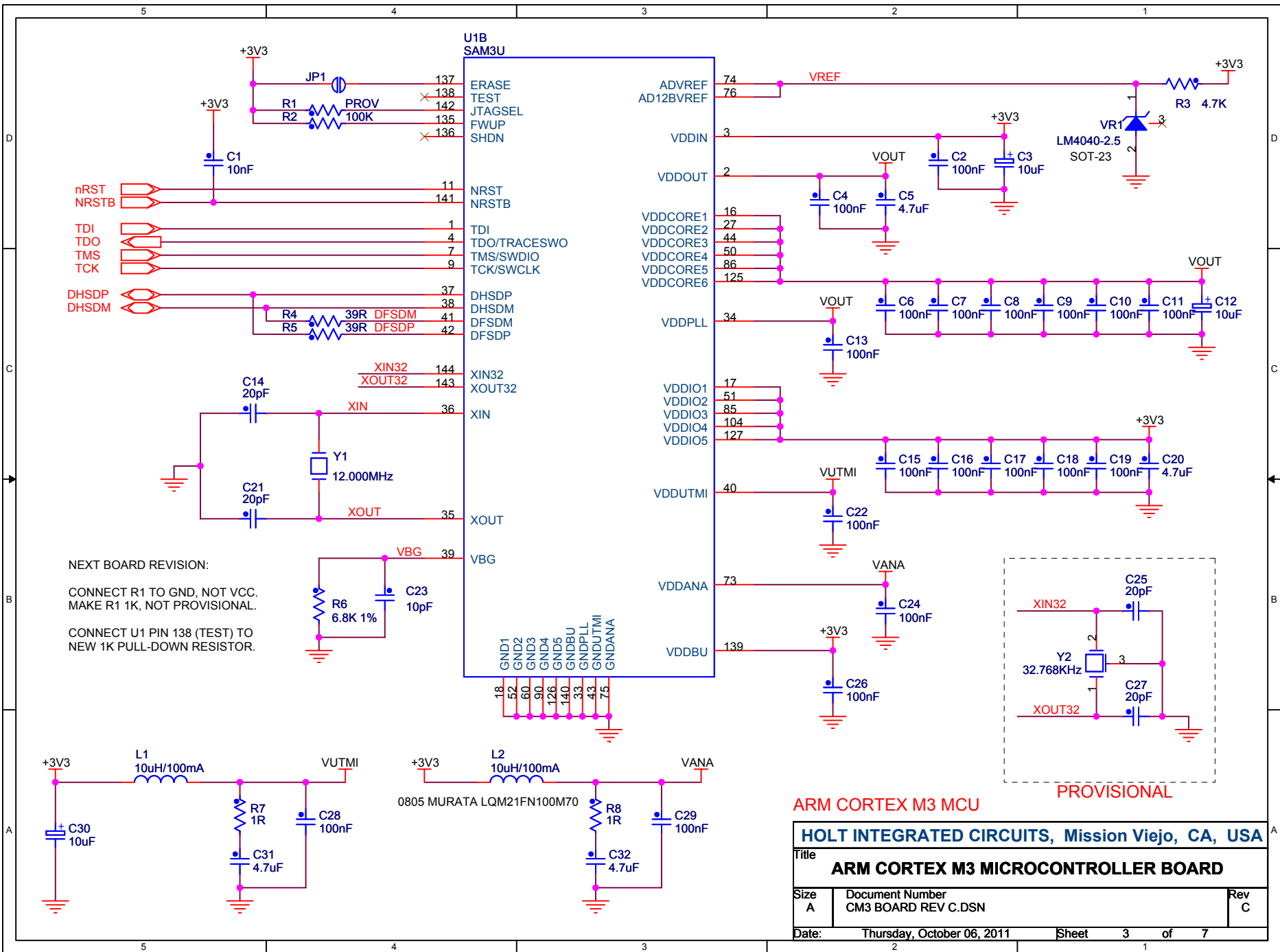
<b>HOLT INTEGRATED CIRCUITS, Mission Viejo, CA, USA</b>		
Title		
<b>ARM CORTEX M3 MICROCONTROLLER BOARD</b>		
Size	Document Number	Rev
A	CM3 BOARD REV C.DSN	C
Date:	Thursday, October 06, 2011	Sheet 1 of 7

BUTTONS

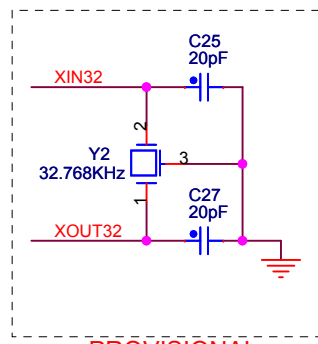


**ARM CORTEX M3 PIO**

<b>HOLT INTEGRATED CIRCUITS, Mission Viejo, CA, USA</b>		
Title <b>ARM CORTEX M3 MICROCONTROLLER BOARD</b>		
Size A	Document Number CM3 BOARD REV C.DSN	Rev C
Date:	Thursday, October 06, 2011	Sheet 2 of 7

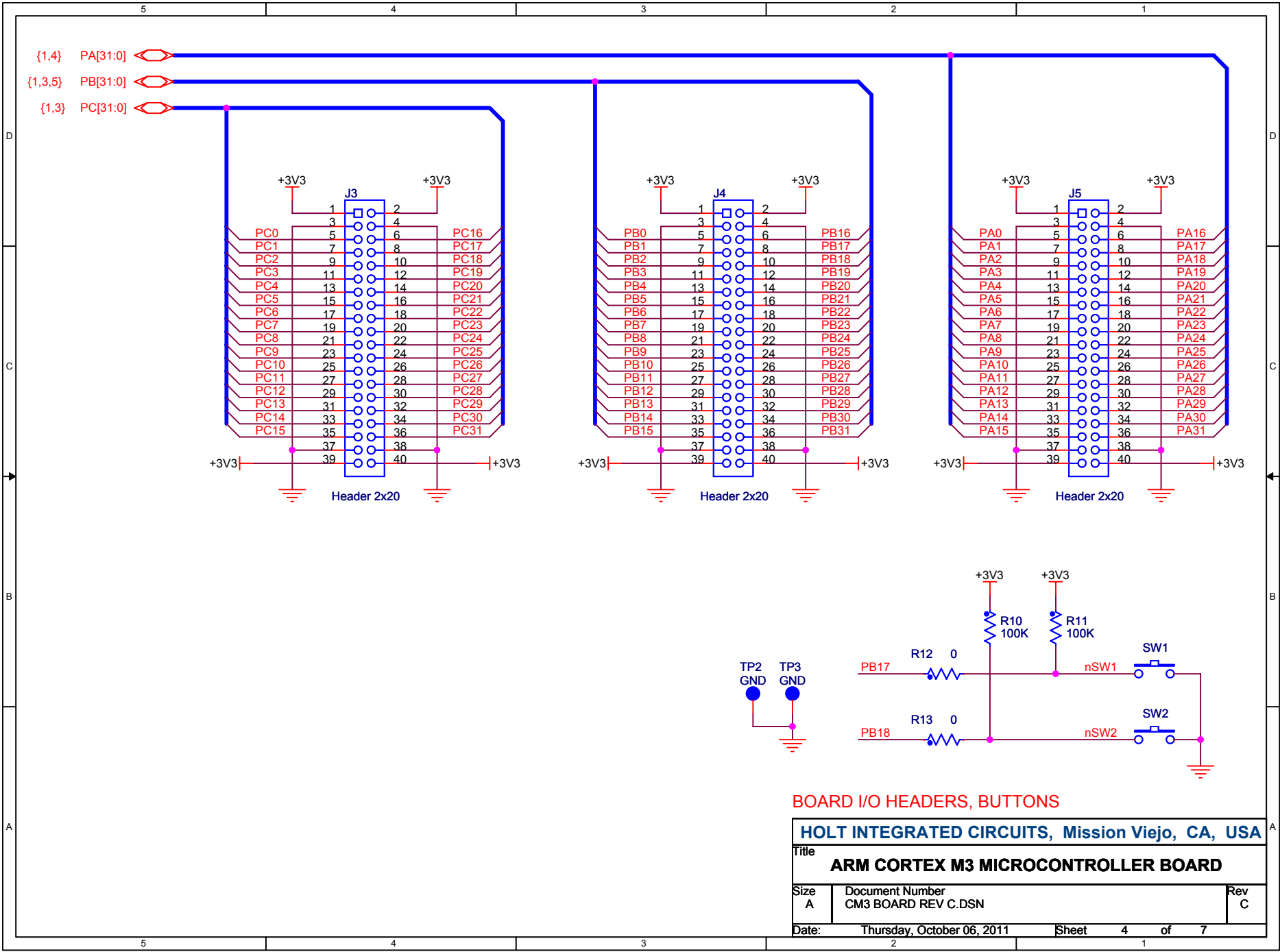


NEXT BOARD REVISION:  
CONNECT R1 TO GND, NOT VCC.  
MAKE R1 1K, NOT PROVISIONAL.  
CONNECT U1 PIN 138 (TEST) TO  
NEW 1K PULL-DOWN RESISTOR.



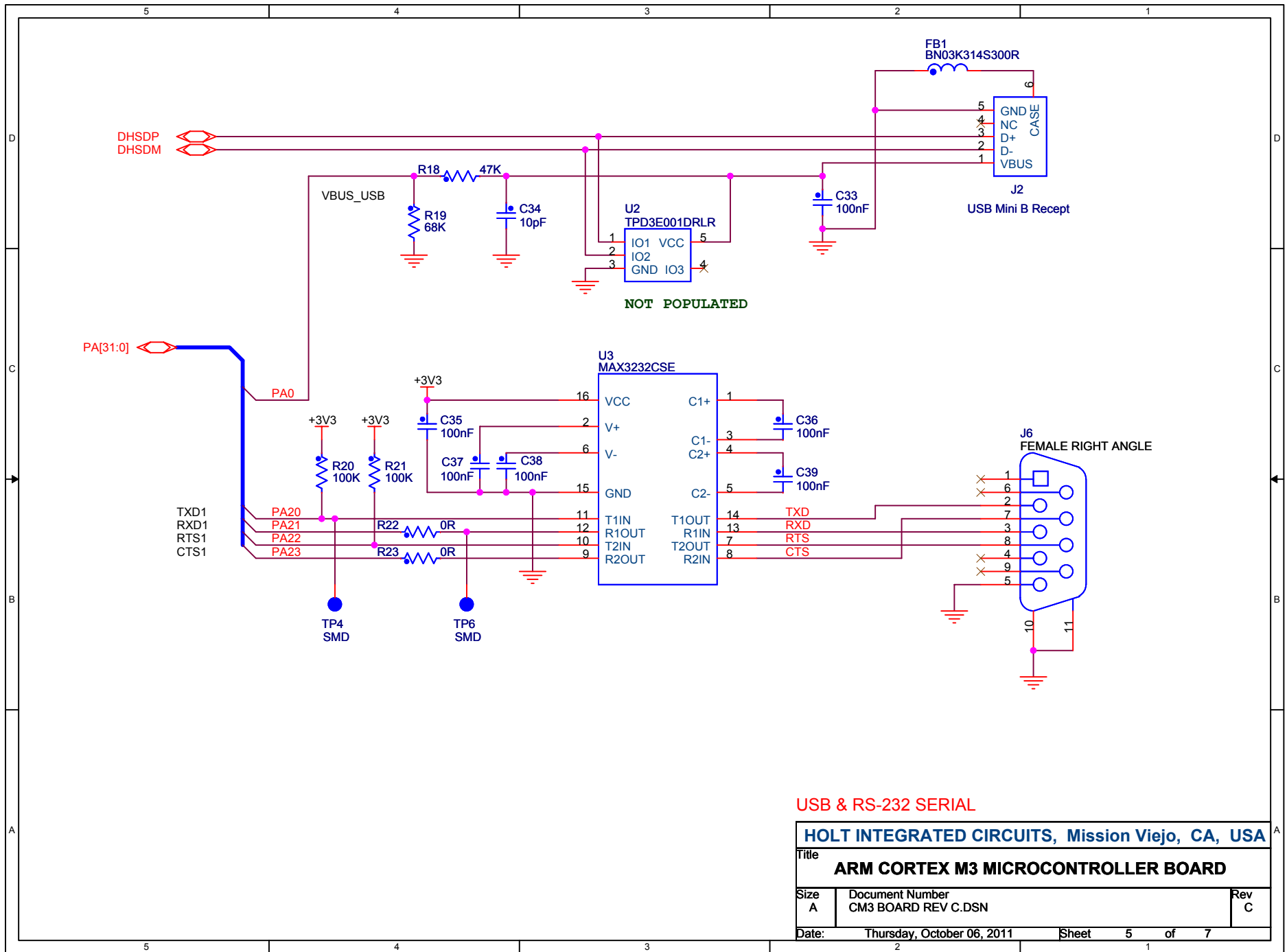
ARM CORTEX M3 MCU

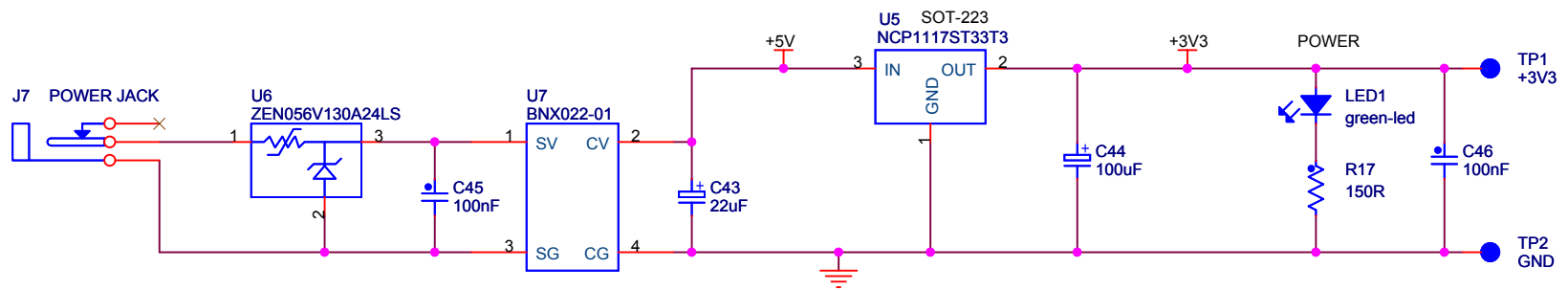
HOLT INTEGRATED CIRCUITS, Mission Viejo, CA, USA		
Title <b>ARM CORTEX M3 MICROCONTROLLER BOARD</b>		
Size A	Document Number CM3 BOARD REV C.DSN	Rev C
Date:	Thursday, October 06, 2011	Sheet 3 of 7



**BOARD I/O HEADERS, BUTTONS**

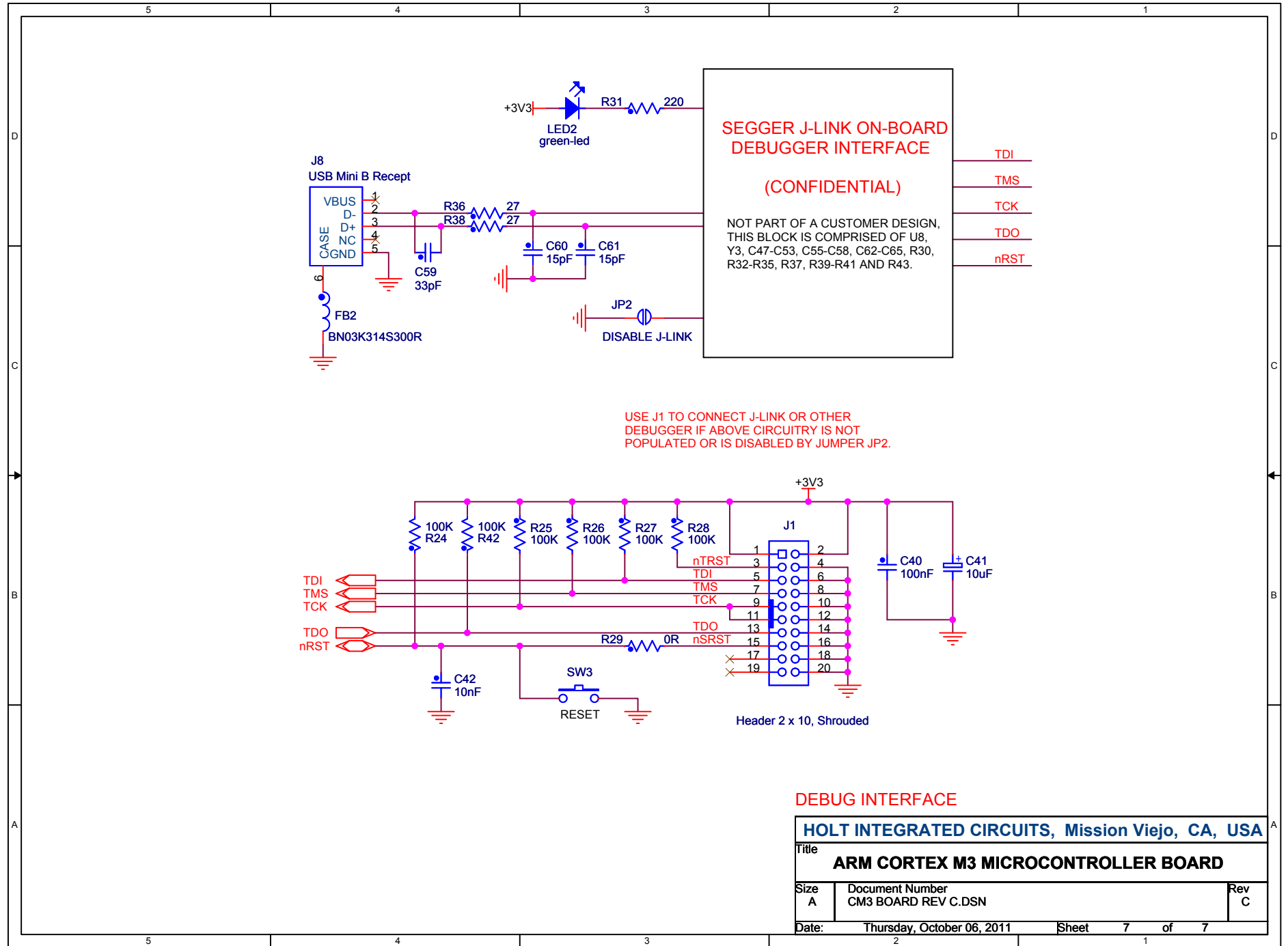
<b>HOLT INTEGRATED CIRCUITS, Mission Viejo, CA, USA</b>		
<b>Title</b> <b>ARM CORTEX M3 MICROCONTROLLER BOARD</b>		
<b>Size</b> A	<b>Document Number</b> CM3 BOARD REV C.DSN	<b>Rev</b> C
<b>Date:</b> Thursday, October 06, 2011	<b>Sheet</b> 4	<b>of</b> 7





**POWER SUPPLY**

HOLT INTEGRATED CIRCUITS, Mission Viejo, CA, USA		
Title <b>ARM CORTEX M3 MICROCONTROLLER BOARD</b>		
Size A	Document Number CM3 BOARD REV C.DSN	Rev C
Date:	Thursday, October 06, 2011	Sheet 6 of 7



Bill of Materials		ARM Cortex M3 MCU Board			Revised: 9 Sept 2011
Item	Qty	Description	Reference	DigiKey	Mfr P/N
1	1	PCB, Bare, Evaluation Board, revision B or C	N/A	-----	
2	1	Ferrite Bead, 220 Ohm @ 100MHz 300mA DC 0805	FB1	732-1602-1-ND	Wurth 742792034
3	2	Capacitor, Ceramic 10nF 10% 50V X7R 0805	C1,C42	399-1158-1-ND	Kemet C0805C103K5RACTU
4	2	Capacitor, Ceramic 10pF 10% NP0 C0G 0V 0805	C23,C34	478-3731-1-ND	AVX 080551A100KAT2A
5	4	Capacitor, Ceramic 20pF 5% NP0 C0G 0V 0805	C14,C21,C25, C27	478-3735-1-ND	AVX 080551A200JAT2A
6	28	Capacitor, Ceramic 100nF 20% 50V Z5U 0805	C2,C4,C6-C11, C13,C15-C19, C22,C24,C26, C28,C29,C33, C35-C40,C45-46	399-1176-1-ND	Kemet C0805C104M5UACTU
7	4	Capacitor, Tantalum 4.7uF 10% 10V Low ESR SMD 1206	C5,C20,C31, C32	478-2391-11-ND	AVX TPSA475K010R1400
8	4	Capacitor, Tantalum 10uF 10% 10V Low ESR SMD 1206	C3,C12,C30,C41	478-3317-1-ND	AVX TPSA106K010R1800
9	1	Capacitor 22uF 10% 6.3V Tantalum Low ESR SMD C	C43	495-1504-1-ND	Kemet B45197A1226K309
10		Capacitor 100uF 10% 6.3V Tantalum Low ESR SMD C	C44	495-1509-1-ND	Kemet B45197A1107K309
11	1	Header, Male Shrouded 2x10 0.1" Pitch	J1	MHB20K-ND	3M 2520-6002UB
12	1	Connector, Receptacle USB Mini B Rt-Angle PCB Mount	J2	H2959CT-ND	Hirose UX60-MB-5ST
13	1	Connector DB9F, Right-Angle PCB Short Body, Board Lock	J6	182-109FE-ND	NorComp 182-009-213R-561
14	1	Jack, DC Power, 2.5mm ID x 2.1mm pin	J7	CP-102AH-ND	CUI PJ-102AH
15	2	Receptacle, Female 2x20 0.1" Pitch, 8.5mm Height, 3.2mm Solder Tails	J3,J4	S6104-ND	Sullins PPTC202LFBN-RC
16	1	Receptacle, Female 2x4 0.1" Pitch, 8.5mm Height, 3.2mm Solder Tails	J5A (J5 lower end, close to Bus B)	S7072-ND	Sullins PPTC042LFBN-RC
17	1	Receptacle, Female 2x5 0.1" Pitch, 8.5mm Height, 3.2mm Solder Tails	J5B (J5 upper end, close to Bus A)	S6105-ND	Sullins PPTC052LFBN-RC
18	1	Solder Jumper	JP1	SOLDER CLOSED	
19	2	Inductor, 10uH, 100mA 0805	L1,L2	490-4029-1-ND	Murata LQM21FN100M70L
20	1	LED Green 0805	LED1	160-1179-2-ND	LiteOn LTST-C170GKT
21	0	Resistor, Prov 1/8W 0805	R1	DO NOT STUFF	
22	5	Resistor, 0 ohm 1/8W 0805	R12,R13,R22, R23,R29	311-0ARCT-ND	Panasonic ERJ-6GEY0R00V
23	2	Resistor, 1.0 5% 1/8W 0805	R7,R8	P1.0ACT-ND	Panasonic ERJ-6GEYJ1R0V
24	2	Resistor, 39 5% 1/8W 0805	R4,R5	P39ACT-ND	Panasonic ERJ-6GEYJ390V
25	1	Resistor, 150 5% 1/8W 0805	R17	P150ACT-ND	Panasonic ERJ-6GEYJ151V
26	1	Resistor, 4.7K 5% 1/8W 0805	R3	P4.7KACT-ND	Panasonic ERJ-6GEYJ472V
27	1	Resistor, 6.8K 5% 1/8W 0805	R6	P6.8KACT-ND	Panasonic ERJ-6GEYJ682V
28	1	Resistor, 47K 5% 1/8W 0805	R18	P47KACT-ND	Panasonic ERJ-6GEYJ473V
29	1	Resistor, 68K 5% 1/8W 0805	R19	P68KACT-ND	Panasonic ERJ-6GEYJ683V
30	10	Resistor,100K 5% 1/8W 0805	R2,R10,R11, R20,R21,R24, R25,R26,R27, R28	P100KACT-ND	Panasonic ERJ-6GEYJ104V
31	3	Pushbutton	SW1,SW2,SW3	P10886SCT-ND	Panasonic EVQ-QWS02W
32	2	Test Point, Black Insulator, 0.062" hole	TP2,TP3	5011K-KD	Keystone 5011



33	1	Test Point, Orange Insulator, 0.062" hole	TP1	5008K-ND	Keystone 5008
34	1	Test Point, Yellow Insulator, 0.062" hole	TP4	5009K-ND	Keystone 5009
35	2	Test Point, Hole / Pad Only	TP5,TP6		
36	1	IC, MCU 32-Bit 256KB Flash, 144-LQFP	U1	ATSAM3U4EA-AU-ND	Atmel ATSAM3U4EA-AU
37	1	IC, ESD Protection Array 3-Channel SOT-5	U2	296-21885-1-ND	Texas Inst TPD3E001DRLR
38	1	IC, RS232 Driver/Receiver 3.0 to 5.5VDC 16-SOIC (3.9mm wide)	U3	296-19752-1-ND	Texas Inst MAX3232EIDR
39	1	IC, Single Inverter 74LVC1G04 SC70-05	U4	296-11600-1-ND	Texas Inst SN74LVC1G04DCKR
40	1	IC Voltage Regulator 3.3V 1A LDO, SOT-223	U5	497-1228-1-ND	ST Micro LD1117AS33TR
41	1	PolyZen 5.6V PPTC protected Zener SMD	U6	ZEN056V130A24LSCT-ND	Tyco ZEN056V130A24LS
42	1	Filter, EMI 35dB 10A 1MHz-1GHz SMD	U7	490-5052-1-ND	Murata BNX022-01L
43	1	IC Voltage Ref 2.5V 1% Micropower SOT-23	VR1	576-1047-1-ND	Micrel LM4040DYM3-2.5
44	1	Crystal 12.00MHz, 50ppm 20pF, HC-49US leaded	Y1	631-1105-ND	Fox FOXSLF/120-20
45	1	Crystal, 32768 Hz 12.5pF cylinder leaded	Y2	535-9033-1-ND	Abrakon AB26TRB-32.768KHZ-T
<b>J-Link On-Board Circuitry...</b>					
46	10	Capacitor, Ceramic 100nF -20% / +80% 25V Y5V 0603	C48-C53, C55-C58	490-1575-1-ND	Murata GRM188F51E104ZA01D
47	1	Capacitor, Ceramic 33pF 5% 50V C0G 0603	C59	490-1415-1-ND	Murata GRM1885C1H330JA01D
48	2	Capacitor, Ceramic 15pF 5% 50V C0G 0603	C60,C61	490-1407-1-ND	Murata GRM1885C1H150JA01D
49	2	Capacitor, Ceramic 10pF 5% 50V C0G 0603	C62,C63	490-1403-1-ND	Murata GRM1885C1H100JA01D
50	1	Capacitor, Ceramic 1nF 20% 50V X7R 0603	C64	490-1495-1-ND	Murata GRM188R71H102MA01D
51	1	Capacitor, Ceramic 10nF 10% 50V X7R 0603	C65	490-1512-1-ND	Murata GRM188R71H103KA01D
52	1	Capacitor, Ceramic 4.7uF -20% / +80% 6.3V 0603	C47	587-1313-1-ND	Taiyo Yuden JMK212F475ZD-T
53	1	Ferrite Bead, 220 Ohm @ 100MHz 300mA DC 0805	FB2	SAME AS FB2 ABOVE	
54	1	Solder Jumper	JP2	LEAVE OPEN	
55	1	Connector, Receptacle USB Mini B Rt-Angle PCB Mount	J8	SAME AS J2 ABOVE	
56	1	LED Green 0805	LED2	SAME AS LED1 ABOVE	
57	1	Resistor, 0 ohm 1/10W 0603	R30	P0.0GCT-ND	Panasonic ERJ-3GEY0R00V
58	1	Resistor, 220 ohm 5% 1/10W 0603	R31	P220GCT-ND	Panasonic ERJ-3GEYJ221V
59	2	Resistor, 1.5K ohm 5% 1/10W 0603	R32,R41	P1.5KGCT-ND	Panasonic ERJ-3GEYJ152V
60	1	Resistor, 47K ohm 5% 1/10W 0603	R33	P47KGCT-ND	Panasonic ERJ-3GEYJ473V
61	1	Resistor, 100 ohm 5% 1/10W 0603	R34,R35,R37 R39	P100GCT-ND	Panasonic ERJ-3GEYJ101V
62	1	Resistor, 27 ohm 5% 1/10W 0603	R36,R38	P27GCT-ND	Panasonic ERJ-3GEYJ270V
63	1	Resistor, 300 ohm 5% 1/10W 0603	R40	P300GCT-ND	Panasonic ERJ-3GEYJ301V
64	1	IC AT91SAM7S64 64-PQFP programmed by Segger	U8	from Segger	
65	1	Crystal 18.432MHz, 30ppm 10pF, SMD 3.2x2.5 mm	Y3	535-10909-1-ND	Abrakon ABM8G-18.432MHZ-4Y-T3

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) Tx SA1 (data pointers A, B and broadcast data pointer)	ping-pong	DPA	34	0800	0821	60001000	60001042	MIW + TT + 32 words same
		DPB	34	0822	0843	60001044	60001086	
	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 for overrun
		pad	32	15B6	15D5	60002B6C	60002BAA	
Tx SA3	circ1-32	DPA	1088	15D6	1A15	60002BAC	6000342A	32 x (MIW + TT + 32 words) pad for overrun
		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 same
		DPB	34	4022	4043	60008044	60008086	
		BDP	34	4044	4065	60008088	600080CA	
	ping-pong	DPA	34	4066	4087	600080CC	6000810E	same same MIW + TT + 2 pad
		DPB	34	4088	40A9	60008110	60008152	
		BDP	4	40AA	40AD	60008154	6000815A	
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	index-32	DPA	1088	4532	4971	60008A64	600092E2	MIW + TT + 2 pad
		BDP	4	4972	4975	600092E4	600092EA	
Rx SA3	circ1-32	DPA	1088	4976	4DB5	600092EC	60009B6A	32 x (MIW + TT + 32 words) pad for overrun
		pad	32	4DB6	4DD5	60009B6C	60009BAA	
Tx SA3	circ1-32	DPA	1088	4DD6	5215	60009BAC	6000A42A	32 x (MIW + TT + 32 words) pad for overrun
		pad	32	5216	5235	6000A42C	6000A46A	
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**

<b>RAM Structure</b>	<b>Start Address</b>	<b>End Address</b>	<b>Number of Words</b>
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