



MAMBA™ ファミリー
MIL-STD-1553 API アプリケーション開発キット

2016 年 6 月



株式会社ナセル

履歴

履歴	日付	変更内容
AN-6138API, Rev. New	2016 / 4 / 1	初版リリース
Rev.A	2016 / 6 / 7	BOM を更新 ボードの写真を更新

本文書は、Holt 社の『AN-6138API』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]

Holt MAMBA™ 評価ボードは、MIL-STD-1553 プロトコル・デバイスの MAMBA™ ファミリーで使用する場
合、Holt ハイレベル 1553API ライブラリの幅広い機能セットを示しています。

HI-6135 : RT

HI-6136 : RT および／または BM

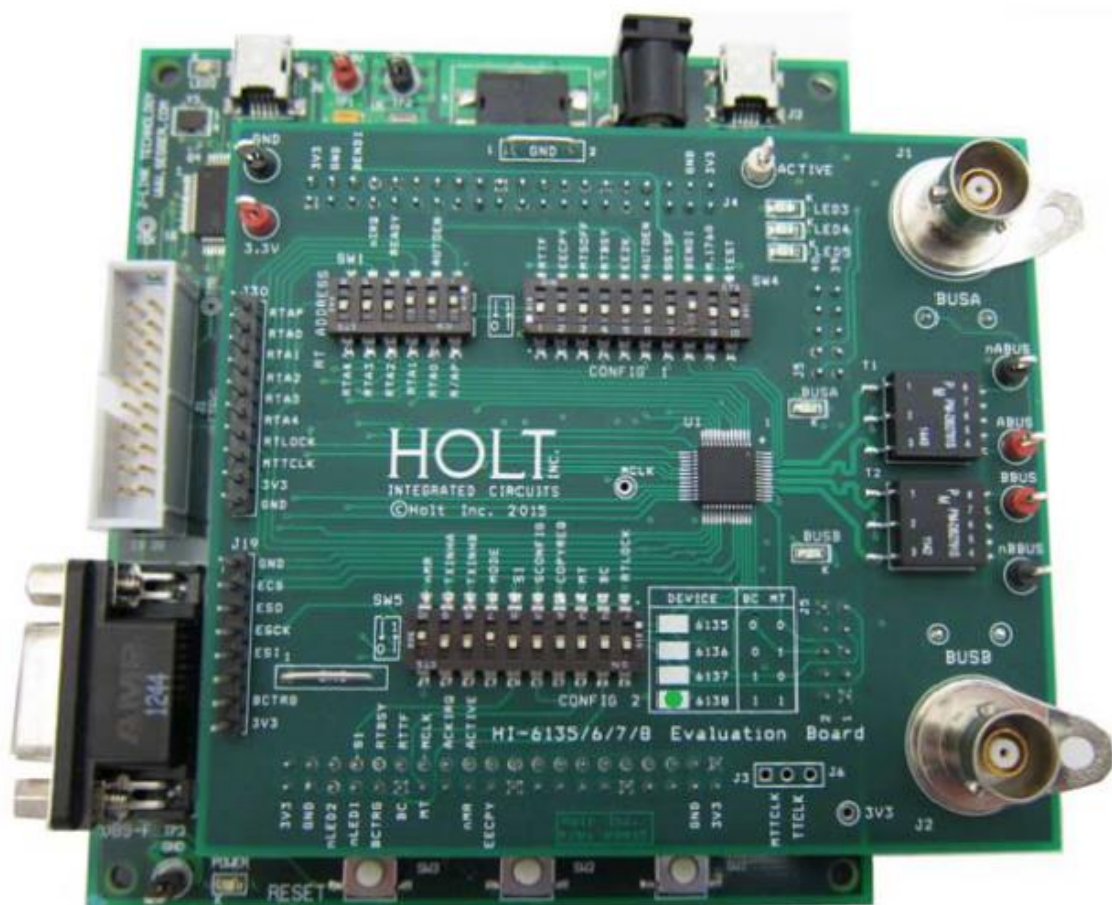
HI-6137 : RT および／または BC

HI-6138 : RT、BC、および／または BM (ADK HI-6138 使用)

MIL-STD-1553B バス通信デバイスの MAMBA™ ファミリーには極めて効率の良いプロトコル・ロジックと
物理バス・インターフェイス回路が含まれています。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、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 MAMBA™ ガイドのリファレンス・デバイスは HI-6138 です。この IC には利用可能なすべ
ての機能が含まれているためです；他の MAMBA™ デバイスには HI-6138 機能のサブセットが含まれてい
ます。

この API バージョンの MAMBA™ ADK は、Holt API ライブラリを示しています。非 API 組み込み制御（デ
モ・キット ADK-6135/6136/6137/6138）については、AN-6138.pdf を参照してください。

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。

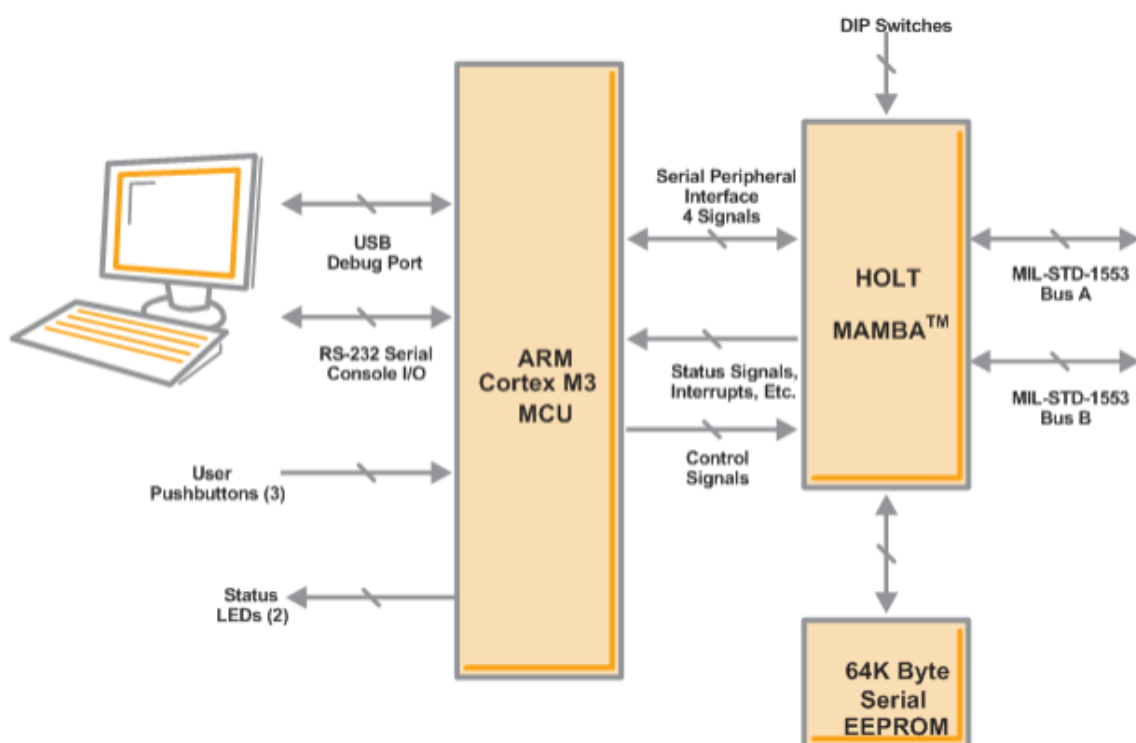


MAMBA™ 評価ボード、ARM Cortex M3 MCU ボードに搭載

Evaluation Kit Contents [開発キット構成]

- 本ユーザー・ガイド
- Holt MAMBA™ ソフトウェア・プロジェクトおよび、ドキュメント CD
 - ◇ HI-613x API LIB ランタイム・ライブラリ
 - ◇ HI-6138 API デモ・サンプル・プロジェクト
- IAR システムズの ARM 用 Embedded Workbench® Ver.7.1 または、それ以上。重要な手順については、インストール・ガイド「Holt HI-6138 API project installation guide」を参照してください。
- 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
- 2 ボード構成で、
 - 上段 MAMBA™ 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]



Default Switch Settings [デフォルト・スイッチ設定]

RT アドレス

スイッチ	ポジション	説明
SW1、6-2	00011 (OFF = 1)	RTA4:0 … RT アドレス設定。デフォルト設定 RT3
SW1、1	OFF = 1	RTAP … RT アドレス・パリティ・ビット。正しい奇数パリティを反映していなければ、RT 動作を開始することができません。デフォルト (RT3) は論理 1

コンフィグ 1

スイッチ	ポジション	説明
SW4、1	OFF (0)	RTTF – ON … RT ステータス・ワードのターミナル・フラグ・ビットを設定
SW4、2	OFF (0)	EECPY – ON … RAM とレジスタのコピーを EEPROM に作成 注 1
SW4、3	OFF (0)	MTSTOFF – ON … 電源投入時のメモリテスト無効
SW4、4	OFF (0)	RTBSY – ON … MCU が Busy Bit を RT ステータス・ワードにセット
SW4、5	OFF (0)	EE2K – ON … EECPY と AUTOEN は、ブートプロセス高速化のためにわずか 2K ワードの EEPROM を使用します。注 1
SW4、6	OFF (0)	AUTOEN – ON … 外部 EEPROM からのレジスタ/RAM 初期化を有効にします。それ以外の場合は MCU がデバイスを初期化します。注 1
SW4、7	OFF (0)	SSYSF – ON … RT ステータス・ワードにサブシステム・フェイル・ビットを設定します。
SW4、8	ON (1)	BENDI – ON … ビッグ・エンディアンのメモリ&レジスタ・アクセスを設定します。
SW4、9	OFF (0)	MODE1760 – ON … RT ステータス・ワードのビジー・ビットを、ハードウェア・マスタ・リセットの立ち上がりエッジですぐにセットします。
SW4、10	OFF (0)	TEST – ON … セルフテストを可能にします。デモには追加のソフトウェアが必要です。データシートの 21 項を参照してください。

コンフィグ 2

スイッチ	ポジション	説明
SW5、1	ON (1)	RTLCK - OFF (論理 1) : RT アドレスをロックするため、スイッチ SW1 からラッチされた外部 RT アドレスは、レジスタ書込みによって上書きされません。
SW5、2	OFF (1) (6138, 6137) ON (0) (6136, 6135)	BC - OFF このスイッチは、この API バージョンのデモ・プログラムでは使用されていません。ユーザーが再利用することができます。
SW5、3	OFF (1) (6138, 6137) ON (0) (6136, 6135)	MT - OFF このスイッチは、この API バージョンのデモ・プログラムでは使用されていません。ユーザーが再利用することができます。
SW5、4	ON (0)	COPYREQ - OFF (論理 1) : MCU プログラムはブートアップ後に RAM および、レジスタ設定を EEPROM に書き込みます。 注 1
SW5、5	ON (0)	SCONFIG : 未使用
SW5、6	ON (0)	S1 : 未使用
SW5、7	OFF (1)	MODE : デバイスはモード・ピンを High にする必要があります
SW5、8	ON (0)	TXINHB - OFF (論理 1) : 1553 BUSB ドライバを無効
SW5、9	ON (0)	TXINHA - OFF (論理 1) : 1553 BUSA ドライバを無効
SW5、10	OFF (1)	nMR - ON (論理 0) : デバイスをリセット状態に保持

注 1 : このデモ・ソフトウェアでは、API ランタイム・ライブラリが初期化を実行するため、EEPROM からのレジスタおよび、RAM の自動初期化は使用されません。このオプション機能の情報とサンプル・コードについては、Holt アプリケーション・ノート AN-6138 を参照してください。

1760 モード (全デバイス)

このモードでは、RT デバイスは、マスター・リセット・ピンの立ち上がりエッジの 2msec 以内に設定されたステータス・ワードの Busy ビットで応答します。この機能をテストするには、ソフトウェアを動作させずにデバイスを起動することができます (例えば、DIP スイッチを使用して MCU をリセット状態に保持するなど)。nMR スイッチが ADK (SW5/10) 上でトグルされている場合、デバイスは「Busy」Bit がセットされた BC コマンドに迅速に応答することができます。

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

ジャンパ	位置	説明
JP2	OPEN	BUSA のマイナス・ラインをボードのグラウンドに接続するためのリンク 注2
JP3	OPEN	BUSB のマイナス・ラインをボードのグラウンドに接続するためのリンク 注2
JP8	OPEN	BUSA に 70Ω の不可抵抗を接続するためのリンク 注3
JP9	OPEN	BUSB に 70Ω の不可抵抗を接続するためのリンク 注3
JP10	CLOSED	50MHz クロックを下段 MCU ボードにリンク

注2：バスの Negative をグラウンドに接続することは、厳密なベンチ・テストに便利な機能です。送受信される 1553 信号のほとんどの性能特性は、バス・スタブで差動線間測定（バス Positive – バス Negative）を使用して指定されます。これは上段ボードの右側にあるトランスに隣接する赤と黒の「BUS」テスト・ポイントに対応します。スコープの CH1 - CH2 の演算機能と組み合わせて赤と黒に接続された 2 つのオシロスコープのプロブを使用できますが、バス Positive に接続された単一のプロブは、バス Negative が接地されているときに表示されるものと同じ信号を提供します。これによりスコープ・プロブが他の目的で使用可能です。

バス Negative の接地は、製品設計には含めないでください。

注3：スタンドアロン・テスト（従来の MIL-STD-1553 バスに接続しない）の場合、ハードウェアが提供するドーター・カードの背面のオンボード 70Ω 終端抵抗を JP8 と JP9 で有効にします。これは厳密なテスト・ベンチに便利な機能で、外部の 1553 バス接続無しで BC と RT を同時に実行できることを実証します。HI-6138 には、BC、RT および、MT モードを同時に実行するための独立したステート・マシンが含まれています。デモンストレーションの目的で、BC および、RT は、バスモニタの有無にかかわらずメッセージを完全に処理できます。ADK ではそのように構成されています。

オンボード終端抵抗は、適切に終端された MIL-STD-1553 バスに接続するときには使用されません。製品設計には終端抵抗の規定を含めないでください

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 MAMBA™ 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。GPIO ピンとして設定された NPCS0 を使用する Atmel プロセッサの 4 線シリアル・ペリフェラル・インターフェイス (SPI マスタ 24MHz) は、MAMBA™ (スレーブ) に接続されます。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。REST 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。)

Holt API Host Memory Considerations [Holt API ホスト・メモリ構成注意事項]

Holt API は C `malloc()`関数を使用して、ARM Cortex M3 内蔵 SRAM から API ホスト・バッファのようにメモリを割り当てます。選択した Atmel ARM Cortex M3 で使用可能な SRAM の総量は 48Kbyte です。この SRAM は、スタックおよび、ヒープを含むすべての C コード静的動的変数で共有されます。このソフトウェア・デモには、48Kbyte で十分です。ユーザー SRAM 要件は、アプリケーションに必要な有効な 1553 ターミナル・モードおよび、バッファ・サイズの数によって異なります。

より多くのメモリが必要な場合は、より多くの SRAM を持つプロセッサを選択するか、外付け SRAM を使用してプロセッサを拡張します。Holt ADK-6130-2 ボードは、同じ Atmel ARM Cortex M3 プロセッサと 256Kbyte の外部 SRAM メモリを使用する例です。この例については、AN-6130-2A.pdf を参照してください。注: ADK-6130 はパラレル・バス・インターフェイス (SPI ではなく) を持つ Holt HI-6130 プロトコル IC を使用し、メモリ拡張も MCU 外部バス・インターフェイスを使用します。

より多くの内部 SRAM を持つ代替 MCU があります。例えば、NXP は 256Kbyte の内部 SRAM を持つ「Kinetic」ARM Cortex M4 マイクロコントローラを提供しています。NXP は Kinetic ARM Cortex M4 を SPI 専用 (SPI および、外部バス・インターフェイスと同様) 構成で提供しています。ここで使用しているボー

ド上の ARM Cortex M3 を 256Kbyte の内臓 SRAM で SPI のみの選択で置き換えることができます。

Initial Kit Set Up [キットの初期セット・アップ]

Holt の MAMBA™ アプリケーション開発キットは、MAMBA™ ファミリーの 4 つのデバイスすべてを示します。HI-6130 は、3 つのターミナル・モード全てで動作します：RT、BC、MT。別の MAMBA™ を使用する場合は、1 つまたは 2 つのターミナル・モードを無効にすることで簡単に適用できます。

1. ご使用の PC にはシリアル (COM) ポートと、TeraTerm のような「ターミナル・エミュレーション」プログラムが必要です。ほとんどのコンピュータには RS-232 COM ポートが無いため、ADK 付属のシリアル/USB アダプタが必要になります。これをコンピュータの USB ポートに接続し、9 ピン・コネクタを ADK ボードに接続します。
2. Windows 2000 または、WindowsXP を使用している場合、ハイパーターミナルをターミナル・エミュレーションに使用できます。「スタート」→「すべてのプログラム」の順にクリックして、「Windows アクセサリ」→「通信プログラム」の順にクリックして、ハイパーターミナルを開きます。ハイパーターミナルをクリックして実行します。次の段落をスキップしてください。

Windows7 または、Vista 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 Tera Term」をクリックすることで Tera Term 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。

Tera Term 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. 付属の 5VDC 電源を差し込み、ケーブルを下段のボードの電源入力ジャックに接続します。Tera Term が正しく動作し、正しく設定されている場合、コンソール・ウィンドウに以下のコマンド・メニューが表示されます。このメニューは、ボードの電源が投入されたとき、または RESET プッシュ・ボタンが押された後に表示されます。評価ボードとの正しい Tera Term 通信を確認した後、セット・アップをクリックして保存することにより、ターミナルセットアップを保存できます。

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

Host is Initializing Regs & RAM
Reset 613x waiting for READY
Reset 613x waiting for READY

*****
Holt Integrated Circuits Mamba SPI API Demo Project
Compiled: Feb 19 2016 10:07:17
*****

BC On   SMT On   RT On

Press 'R' to Display HI-613x Registers.
Press 'K' to Enable RTMT.
Press 'A' to run BC Async demo.
Press 'H' to send high priority BC message.
Press 'L' to send low priority BC message.
Press 'N' to run BC Major Minor Frame demo.
Press 'X' to stop BC transmissions.
Press 'S' to run SMT demo.
Press 'T' to display RT Traffic Toggle.
Press 'B' to run RT demo.
Press 'W' for Mamba Memory Watch window
Press '1' for Mamba SPI Register Write (00 to 3F)
Press '2' for Mamba SPI Memory Write (00 to FF)
Press '3' for Mamba SPI Memory Write (00 to 1FFF)

=====

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

```

「R」または「r」を押すと、HI-613x レジスタを表示します。

```

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

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

0x0000 MASTER_CONFIG_REG = 40
0x0001 STATUS_AND_RESET_REG = 8000
0x0002 RT_CURR_CMD_REG = 0
0x0003 RT_CURR_CNTRL_WRD = 0
0x0006 HDW_PENDING_INT_REG = 0
0x0007 BC_PENDING_INT_REG = 0
0x0008 SMT_IMT_PENDING_INT_REG = 0
0x0009 RT_RT_PENDING_INT_REG = 0
0x000a INT_COUNT_AND_LOG_ADDR_REG = 180
0x000f HDW_INT_ENABLE_REG = 6018
0x0010 BC_INT_ENABLE_REG = 0
0x0011 SMT_IMT_INT_ENABLE_REG = 0
0x0012 RT_RT_INT_ENABLE_REG = 408
0x0013 HDW_INT_OUTPUT_ENABLE_REG = 6018
0x0014 BC_INT_OUTPUT_ENABLE_REG = 0
0x0015 SMT_IMT_INT_OUTPUT_ENABLE_REG = 0
0x0016 RT_RT_INT_OUTPUT_ENABLE_REG = 408
0x0017 RT_CONFIG_REG = 80
0x0018 RT_OP_STATUS_REG = 1c00
0x0019 RT_DESC_TBL_BASE_ADDR_REG = 400
0x001a RT_1553_STATUS_BITS_REG = 0
0x001b RT_MSG_INFO_WD_ADDR_REG = 0
0x001c RT_BUSA_SELECT_REG = 0
0x001d RT_BUSB_SELECT_REG = 0
0x001e RT_BIT_WORD_REG = 0
0x001f RT_ALT_BIT_WORD_REG = 0
0x0029 SMT_IMT_CONFIG_REG = 1
0x002a IMT_MAX_MSG_OUNT = 0
0x002b IMT_MAX_1553_WORDS = 0
0x002c IMT_MAX_PKT_TIME = 0
0x002e IMT_CHANNEL_ID = 0
0x002f SMT_IMT_START_ADDR_LIST_POINTER = b0
0x0030 SMT_IMT_NEXT_MSG_STACK_ADDR_REG = 0
0x0031 SMT_IMT_LAST_MSG_STACK_ADDR_REG = 0

```

RT ターミナル・アドレスは、電源投入前に DIP スイッチを使用して設定します。RT アドレス 3 および、1 は、あらかじめプログラムされたバスコントローラ・メッセージ・レパトリリーによって利用されます。6 桁の DIP スイッチは、アドレス値 03 と奇数パリティで既に設定されているはずで

従来の MIL-STD-1553 バスにケーブルを接続していない場合は、半田ジャンパ JP8 と JP9 を接続することにより、バスにダミーの 70Ω 負荷抵抗がボード上に提供されます。

デモ機能の一般的な構造

Holt API デモンストレーション・プログラムは **democ.c** モジュールに含まれています。Holt API ランタイム・ライブラリは、実行可能オブジェクトコードとしてライブラリ・ファイル **MAMBA API LIB.a** に含まれています。**democ.c** ファイルには、コンソール・メニューから実行されるデモをサポートするデモ初期化 API 関数呼び出しが含まれており、BC、RT および、MT を初期化します。**main.c** のメイン・ループから呼び出された **console.c** でキーの押下が検出され、**democ.c** のデモ機能が実行されます。

コマンド「A」と「N」の送信 BC コマンドは、オシロスコープで表示できます。オプションで「K」および、「T」コマンド・シーケンスを使用して、コンソールにメッセージ・トラフィック・データを表示できます。これらのデモでは、Holt API を使用して BC 非同期メッセージ、メジャー／マイナー・フレーム、低優先度、高優先度メッセージを生成する方法を示します。外部の MIL-STD-1553 テスト装置でこれらのメッセージを表示するか、オシロスコープで表示してください。

この演習では、内部 BC を使用してメッセージを送信しているので、メッセージ・トラフィックがコンソールに表示されます。内部 BC、RT、SMT は同じバス・ピンを共有しているため、RT および、SMT モニタ・ターミナルは、BC メッセージを受信します。外部 BC がすでにバスカプラを介してバスジャックに接続されている場合は、接続したままにしておくことはできますが、内部 BC 送信と競合する外部 BC 送信はすべて無効にしてください。外部バスが接続されていない場合、BUSA と BUSB の間にオンボードの 70Ω ダミー・バス負荷抵抗を接続するためにジャンパ JP8 と JP9 をショートしておきます (6 ページ参照)。

BC and RT Mode (HI-6138 and HI-6137) [BC および、RT モード (HI-6138 および、HI-6137)]

1. コマンド「B」を押して RT を有効にします
2. コマンド「K」を押して RTMT デモを有効にします
3. コマンド「T」または「スペース」を押して、コンソールに RT トラフィックを表示します。コマンド「T」は交互にオンとオフを切り換えて、コンソールに表示される RT トラフィックを有効または無効にします。コマンド「T」の使用は、コマンド「K」の事前実行に依存します。
4. コマンド「A」を押して、BC 送信メッセージを開始します
メッセージが画面に急速に表示されます。「スペース」を押してコンソール出力を停止します。コンソールは停止し、以下の画面のように表示されます。「スペース」をもう一度押して、コンソール出力を再開します。「スペース」を使用しても BC の送信が停止したり、RT または MT メッセージの受信が妨げられることはありません。コンソール出力のみを停止します。

```

MSG #0682.  TIME = 00119460us   BUS A   TYPE0: BC to RT
             CMD1 1822 --> 03-R-01-02
             DATA 0005 0002
             STA1 1800

MSG #0683.  TIME = 00119508us   BUS A   TYPE2: RT to RT
             CMD1 182A --> 03-R-01-10
             CMD2 0C2A --> 01-T-01-10
ERROR: NORES

MSG #0684.  TIME = 00119572us   BUS A   TYPE2: RT to RT
             CMD1 182A --> 03-R-01-10
             CMD2 0C2A --> 01-T-01-10
ERROR: NORES

MSG #0685.  TIME = 00119638us   BUS B   TYPE2: RT to RT
             CMD1 182A --> 03-R-01-10
             CMD2 0C2A --> 01-T-01-10
ERROR: NORES

MSG #0686.  TIME = 00087800us   BUS A   TYPE0: BC to RT
             CMD1 1822 --> 03-R-01-02
             DATA 0005 0002
             STA1 1800

MSG #0687.  TIME = 00087848us   BUS A   TYPE2: RT to RT
             CMD1 182A --> 03-R-01-10
             CMD2 0C2A --> 01-T-01-10
ERROR: NORES

```

BC コマンドの一部は、RT to RT です。RT3 への BC コマンドではエラーは表示されませんが、RT1 および、RT3 への RT コマンドからは、アドレス 1 に RT が無いので「ERROR : NORES」と表示されます。

このデモでは、バス A の緑色 LED が急速に点滅します（しかし、連続的に点灯しているように見えます）。

5. BC 低優先度非同期メッセージ挿入

コマンド「L」は、スケジュールされた BC メッセージ・リストに優先順位の低いメッセージを挿入します。優先度の低い挿入メッセージは、挿入が要求されたときに処理中の BC マイナー・フレームの完了時に発生します。まず、「B」を押して RT を有効にし、「A」を押すと BC 送信が有効になります。バス B で追加のメッセージを送信するには、「L」を押します。バス B は、スコープで見やすくするために使用され、バス B LED が点滅します。RT が有効になっていない場合は、再試行メッセージがバス B に表示されます；挿入された 3 つのメッセージを確認することが困難になります。これは、電源投入または、リセット後に 1 回だけ機能します。

これらの 3 つのメッセージのスクリーンショットは Ballard USB UA1133 テスタによってキャプチャされたものです。

Rec #	Time	Message	Bus	Error	Data 4x8	Chan	Swd Bits	Warning
0	T=000:00:0... dT=000:00:0...	Cwd1=0822 (01,R,01,02) <DATA WORDS> Swd1=0800	B		01: DEAD BEEF	1		
1	T=000:00:0... dT=000:00:0...	Cwd1=0C2F (01,T,01,15) <DATA WORDS> Swd1=0800	B		01: BBBB 0202 1414 ... 05: 0505 0606 0707 ... 09: 0909 1010 1111 ... 13: 1313 1414 1515	1		
2	T=000:00:0... dT=000:00:0...	Cwd1=0825 (01,R,01,05) <DATA WORDS> Swd1=0800	B		01: CAFE CODE 0303 ... 05: 0505	1		

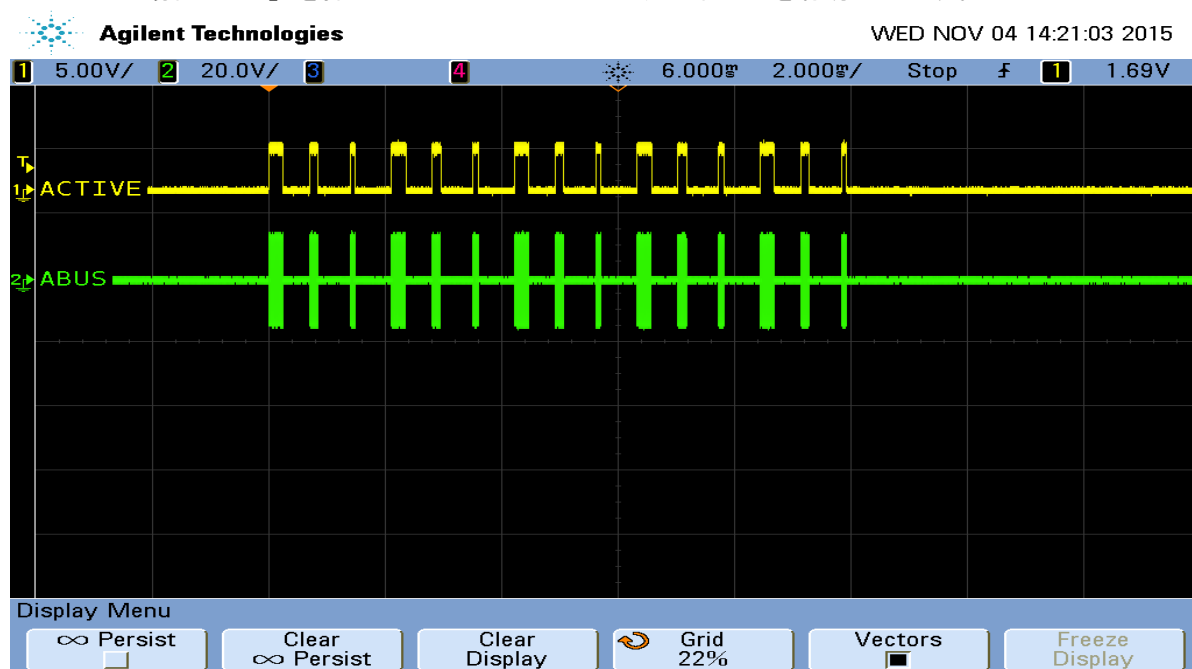
6. BC 高優先度非同期メッセージの挿入

以前の BC 低優先度メッセージの例と同様の手段を実行しますが、今回は「H」を押して高優先度の単一のメッセージを挿入します。このコマンドは繰り返し実行され、各コマンドでバス B LED が点滅します。挿入された高優先度メッセージは、挿入が要求されたときの処理中メッセージ完了時に発生します。

T=000:00:0... dT=000:00:0...	Cwd1=0822 (01,R,01,02) <DATA WORDS> Swd1=0800	B		01: DEAD BEEF	1		
---------------------------------	---	---	--	---------------	---	--	--

7. コマンド「E」 Enumerate Card は、将来の使用のために予約されています。
8. RESET からメッセージ・トラフィックを表示するために、「K」および、「T」シーケンスを有効にする前に BC が開始されている場合、最初のメッセージにエラーが含まれている可能性があります。これは、正常です；これは、RT と MT が進行中のメッセージの中間ストリームで有効になっているために発生します。
9. コマンド「N」は、RT アドレス 3 に 15 個のコマンドを送信します。「B」を押すと RT が有効になります。

「N」を押すと、以下のようにバスに表示される BC 送信（15 個のメッセージが送信されます）が実行されます。オプションでコンソール上のメッセージ・トラフィックを表示するには、まだ有効になっていない場合は「T」を押して RT メッセージ・トラフィックを有効にします。



Ballard USB 1553 モニタを使用した、コマンド「N」(15 メッセージ) トラフィック・キャプチャ

Rec #	Time	Message	Bus	Error	Data 4x8	Chan	Swd Bits	Warning
0	T=000:00:00:00.1583950 dT=000:00:00:00.000000	Cwd1=1C2A (03,T,01,10) <DATA WORDS> Swd1=1800	A		01: 1000 1001 1002 1003 05: 1004 1005 1006 1007 09: 1008 1009	0		
1	T=000:00:00:00.15831208 dT=000:00:00:00.000258	Cwd1=1825 (03,R,01,05) <DATA WORDS> Swd1=1800	A		01: AAAA 0202 0303 0404 05: 0505	0		
2	T=000:00:00:00.15831466 dT=000:00:00:00.000257	Cwd1=1822 (03,R,01,02) <DATA WORDS> Swd1=1800	A		01: BBBB 0202	0		
3	T=000:00:00:00.1584128 dT=-000:00:00:00.000337	Cwd1=1C2A (03,T,01,10) <DATA WORDS> Swd1=1800	A		01: 1000 1001 1002 1003 05: 1004 1005 1006 1007 09: 1008 1009	0		
4	T=000:00:00:00.1584446 dT=000:00:00:00.000318	Cwd1=1825 (03,R,01,05) <DATA WORDS> Swd1=1800	A		01: AAAA 0202 0303 0404 05: 0505	0		
5	T=000:00:00:00.1584704 dT=000:00:00:00.000258	Cwd1=1822 (03,R,01,02) <DATA WORDS> Swd1=1800	A		01: BBBB 0202	0		
6	T=000:00:00:00.1584967 dT=000:00:00:00.000262	Cwd1=1C2A (03,T,01,10) <DATA WORDS> Swd1=1800	A		01: 1000 1001 1002 1003 05: 1004 1005 1006 1007 09: 1008 1009	0		
7	T=000:00:00:00.15841224 dT=000:00:00:00.000256	Cwd1=1825 (03,R,01,05) <DATA WORDS> Swd1=1800	A		01: AAAA 0202 0303 0404 05: 0505	0		
8	T=000:00:00:00.15841542 dT=000:00:00:00.000318	Cwd1=1822 (03,R,01,02) <DATA WORDS> Swd1=1800	A		01: BBBB 0202	0		
9	T=000:00:00:00.1585205 dT=-000:00:00:00.000337	Cwd1=1C2A (03,T,01,10) <DATA WORDS> Swd1=1800	A		01: 1000 1001 1002 1003 05: 1004 1005 1006 1007 09: 1008 1009	0		
10	T=000:00:00:00.1585463 dT=000:00:00:00.000258	Cwd1=1825 (03,R,01,05) <DATA WORDS> Swd1=1800	A		01: AAAA 0202 0303 0404 05: 0505	0		
11	T=000:00:00:00.1585721 dT=000:00:00:00.000257	Cwd1=1822 (03,R,01,02) <DATA WORDS> Swd1=1800	A		01: BBBB 0202	0		
12	T=000:00:00:00.15851043 dT=000:00:00:00.000321	Cwd1=1C2A (03,T,01,10) <DATA WORDS> Swd1=1800	A		01: 1000 1001 1002 1003 05: 1004 1005 1006 1007 09: 1008 1009	0		
13	T=000:00:00:00.15851301 dT=000:00:00:00.000258	Cwd1=1825 (03,R,01,05) <DATA WORDS> Swd1=1800	A		01: AAAA 0202 0303 0404 05: 0505	0		
14	T=000:00:00:00.15851565 dT=000:00:00:00.000264	Cwd1=1822 (03,R,01,02) <DATA WORDS> Swd1=1800	A		01: BBBB 0202	0		

有効にされていない RT に BC メッセージが送信されると「RT no response」(NORES) エラーが表示されます。

MSG #0170. TIME = 00086918us BUS A TYPE2: RT to RT
CMD1 182A --> 03-R-01-10
CMD2 0C2A --> 01-T-01-10

ERROR: NORES

RT Mode (HI-6135, 6136) using external BC [外部 BC を使用する RT モード (HI-6135、HI-6136)]

デモ・ボードにメッセージを送信するために、外部 BC テスタ (Ballard USB 1553 など) を使用する。

- 従来の 1553 バスを使用して外部 BC を接続する場合は、ケーブルを使用してデモ・ボードの円形トライアキシャル・バス・ジャックを A および、B バス・ネットワーク上のバスカプラ・ポートに接続します。この場合、オンボード・ダミー・バス負荷の 70Ω 抵抗を切断する必要があります。6 ページの JP8 と JP9 を参照してください。

バスカプラが容易に利用できない場合は、ボード上のダミー・バス負荷 70Ω 抵抗 (6 ページ参照) を有効にし、BC テスタ・ケーブルをバス A および、B のデモ・ボードのトライアキシャル・バス・ジャックに直接接続して、テスト・ベンチを実行できます。

- RESET ボタンを押して、次に「B」を押して RT を有効にし、「K」と「T」を押してコンソールの RT トラフィックを有効にします。以下に示すメッセージに類似した SA=1 および、32 のデータワードを有する BC to RT メッセージを構成します。

- コンソールは。トランザクションの後、BC が送信したメッセージを表示する必要があります。

```

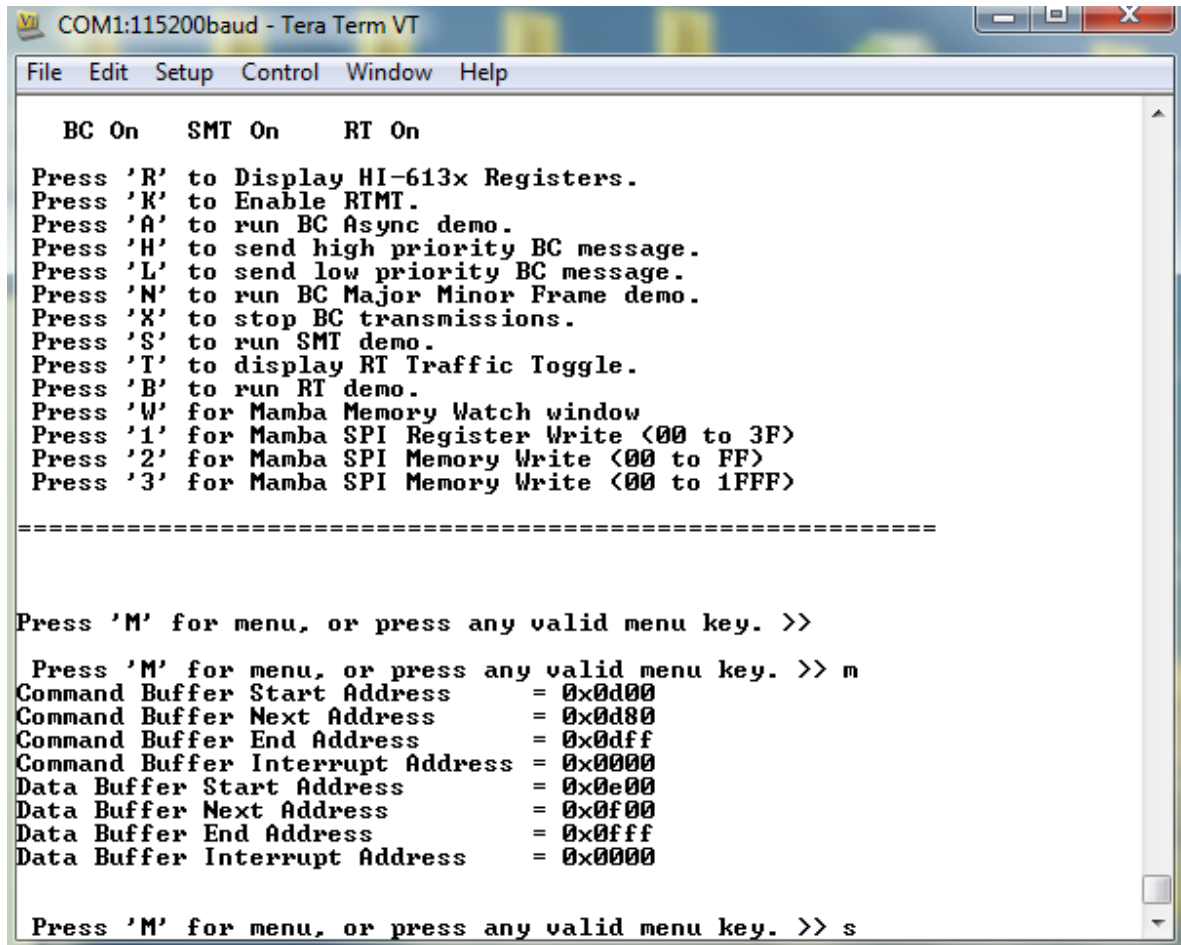
Press 'M' for menu, or press any valid menu key. >> t
MSG #0000.  TIME = 00010098us    BUS A    TYPE0: BC to RT
            CMD1 1820 --> 03-R-01-00
            DATA 0000 0002 0004 0006 0008 000A 000C 000E
                  0010 0012 0014 0016 0018 001A 001C 001E
                  0020 0022 0024 0026 0028 002A 002C 002E
                  0030 0032 0034 0036 0038 003A 003C 003E

            STA1 1800
  
```

- MIL-STD-1553 の典型的な高速レートで繰り返しメッセージを送信する場合、コンソールの RT トラフィックは、コンソールの 115,200bps ボーレート制限と、C プログラムのコンソール・ユーザー・インターフェイスでの実行が遅い *printf* 関数多用のためペースを維持できない可能性があります。メッセージ内容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。すべてのメッセージは適切に処理され、有効な RT と MT によってキャプチャされますが、一部のメッセージはコンソールに表示されません。

MT Mode (HI-6136, 6138) [MT モード (HI-6136、HI6138)]

「S」を押すと、SMT 簡易モニタが有効になります。他のターミナルは必要ありません；ボード・リセット後に「S」を押すと使用することができます。アドレスのリストには、コマンド・スタックとデータ・スタック・バッファの開始アドレスと終了アドレスが表示されます。いくつかのメッセージをモニタに送信した後、このコマンドを使用してアドレスを表示し、メモリ・ウォッチ・ウィンドウを使用してメモリ内のコマンドとデータを表示します。SMT は、メッセージの詳細を提供する「K」コマンドでも初期化されます。「T」RT トラフィック機能は、フォーマットされた RT メッセージ・データの表示をコンソール上で切り換えます。



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

  BC On   SMT On   RT On

Press 'R' to Display HI-613x Registers.
Press 'K' to Enable RTMT.
Press 'A' to run BC Async demo.
Press 'H' to send high priority BC message.
Press 'L' to send low priority BC message.
Press 'N' to run BC Major Minor Frame demo.
Press 'X' to stop BC transmissions.
Press 'S' to run SMT demo.
Press 'T' to display RT Traffic Toggle.
Press 'B' to run RT demo.
Press 'W' for Mamba Memory Watch window
Press '1' for Mamba SPI Register Write (00 to 3F)
Press '2' for Mamba SPI Memory Write (00 to FF)
Press '3' for Mamba SPI Memory Write (00 to 1FFF)

-----

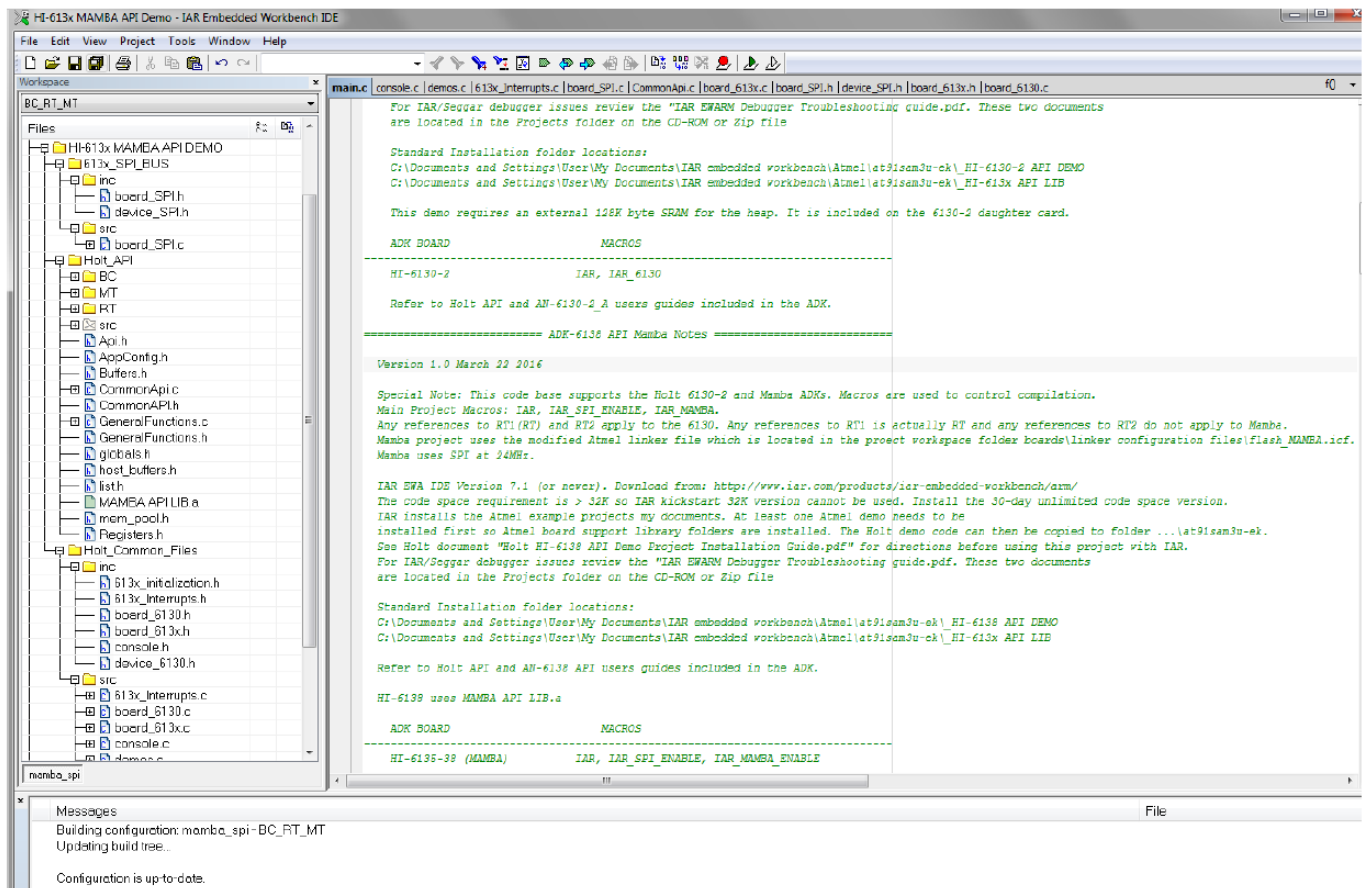
Press 'M' for menu, or press any valid menu key. >>

  Press 'M' for menu, or press any valid menu key. >> m
Command Buffer Start Address   = 0x0d00
Command Buffer Next Address    = 0x0d80
Command Buffer End Address     = 0x0dff
Command Buffer Interrupt Address = 0x0000
Data Buffer Start Address      = 0x0e00
Data Buffer Next Address       = 0x0f00
Data Buffer End Address        = 0x0fff
Data Buffer Interrupt Address   = 0x0000

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

Getting Started with the Holt API demo software project and installing IAR Systems Embedded Workbench for ARM Compiler [Holt API デモ・ソフトウェア・プロジェクトの開始と ARM コンパイラ用 IAR Systems Embedded Workbench]

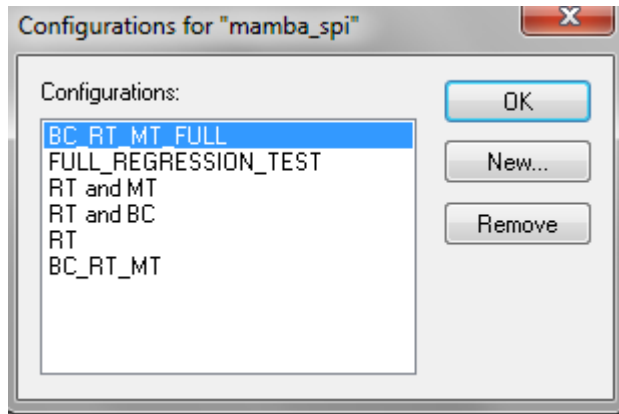
1. Atmel ボード・ライブラリ・ファイルとデモ・プロジェクト・フォルダがすべて適切な場所に作成されるように、Holt のデモ・プロジェクトを追加する前に ARM 用 IAR Systems のコンパイラ、Workbench[®] (EWARM) をインストールする必要があります。Holt の CD-ROM 上のプロジェクト・フォルダにある「Holt HI-6138 API Demo Project Install Guide」に従ってください。次の手順に進む前に、IAR をインストールし、2 つの Holt プロジェクト・フォルダを、ガイドに従って適切なフォルダの場所に配置しなければなりません。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「File / Open / Workspace」をクリックして、プロジェクト・フォルダの場所に移動し、「Holt HI-6138 API Demo Project」から、「HI-6138 API Demo.eww」を選択して「Open」ボタンをクリックします。
3. 以下に示すように、左側に IAR ワークスペース・ウィンドウが表示されます。ワークスペースのディレクトリ・ペインが無い場合は、「View」プルダウン・メニューから、「Workspace」を選択します。任意のウィンドウを調整するか、フォルダ・グループを開いて好みのファイルを表示します。
4. **main.c** ファイルをダブルクリックすると、以下の画面キャプチャと同様に、テキスト・エディタ・ペインに表示されます。
5. プロジェクトが最初に解凍され、適切なフォルダにインストールされると、「Rebuild All」（プロジェクトのプルダウン・メニューから）を実行する必要があります。
6. IAR の開始、プロジェクト管理および、その他のガイドは、IAR Workbench[®]ヘルプ・プルダウン・メニューから利用できます。



Mode Holt Project Configurations [モード Holt プロジェクト構成]

IAR プロジェクト設定は、プリプロセッサ・マクロを使用してデモ・ソフトウェアを再構成するために使用されます。プロジェクト・オプションの C/C++プリプロセッサ・タブで定義されたマクロを使用して、ワークスペース・プルダウン・メニューから設定を選択できます。3 つのマクロ、**IAR_SPI_ENABLE**、**IAR_MAMBA_ENABLE** は MAMBA デモ・プロジェクト用にコンパイルされるコード・セクションを制御します。他のマクロは、コードの BC、RT、または SMT セクションを有効にするために使用されます。Holt は、複数のプロジェクトに共通のコード・ベースを使用します。つまり、このプロジェクトでは使用されていない機能がありますが、C および、H ファイルに残っています。IAR コンパイラはマクロを使用して、ビルドおよび、リンクするコードのセクションを決定します。HI-6130 または HI-6131 へのいくつかの参照が予想されます。場合によっては、一部の関数が単に使用されていないか、関数内でマクロが適切なコードを選択するために使用される場合があります。

構成リストは、表示されているリストと多少異なる場合があります。

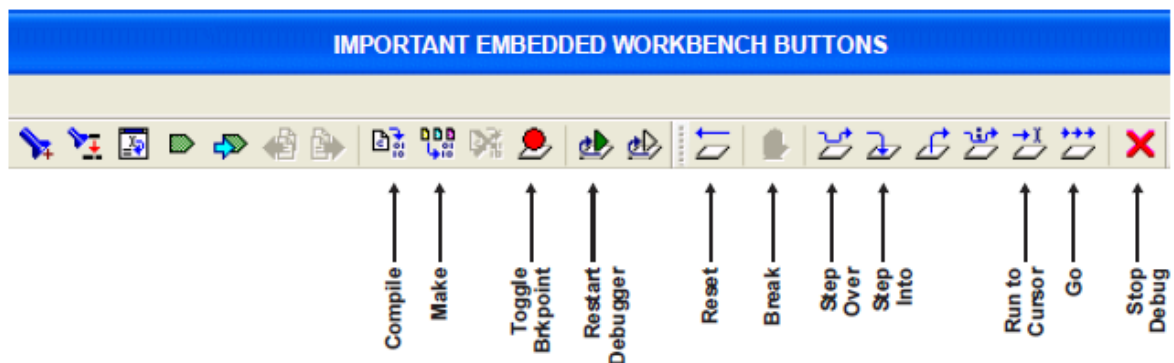


「FULL」のプロジェクト構成では、Holt API ライブラリのソース・コードが必要です。API ライブラリのソース・コードは標準の ADK では提供されていません。完全なソース・コードは、Holt のソフトウェア使用許諾契約書 (SLA) 締結後に使用できます。それ以外の場合、標準の ADK デモは完全に機能し、API プロジェクトは API C ソース・コードなしで構築できます。オプションの API C ソース・コードの SLA については、Holt セールスにお問い合わせください。

「**FULL REGRESSION_TEST**」は Holt テスト専用であり、ユーザーが削除することができます。

新しい構成を作成する簡単な方帆は、「Project/Edit Configurations」を選択し「New」を選択することです。このダイアログ・ボックスでは、新しい構成名を使用して既存の構成に基づき新しい構成を行うことができます。新しい構成を選択し、必要に応じてプリプロセッサのラベルを編集し、新しい設定を保存します。新しい構成がプルダウン・メニューに表示されます。プロジェクト・ファイル **613x_initialization.h** は、タイムタグの分解能、コンソール I/O のオン・オフなどその他の重要なプロジェクト設定を構成します。

IAR Embedded ワークスペースは、一般的に使用されるタスクを、図のように上部にショートカット・ボタンとして提供します。



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

関数名のほとんどは自明のもので、一部の関数は元も 6131 または 6130 の名前を保持していますが、MAMBA™ ファミリーでうまく動作するかどうか心配しないでください。

device_SPI.h

HI-6135/6/7/8 レジスタ・アドレスのマクロ定義

board_SPI.c, board_SPI.h

SPI コンフィグレーション、SPI リード/ライトおよび、メモリ・ウォッチ機能

613x_initialization.h

重要な構成設定の定義

613x_Interrupts.h

API 使用のための割り込みに関するプロトタイプの提供

main.c

プライマリ・プログラム・エントリ・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。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。デモの初期化と実行には、このデモ・キットの Holt ハイレベル API が使用されています。

board_613x.c, board_613x.h

ARM MCU SPI の I/O 定義と SPI コマンドのマクロ定義が含まれます

SPlopcode(opcode) ;

Write_6131LowReg(reg_number, data, irq_mgmt) ;

Read_6131LowReg(reg_number, irq_mgmt) ;

Write_6131_1word(data, irq_mgmt) ;

Read_6131_1word(irq_mgmt) ;

Write_6131(write_data[], inc_pointer_first, irq_mgmt) ;

Read_6131(number_of_words, irq_mgmt) ;

Write_6131_Buffer(write_data[], inc_pointer_first, irq_mgmt) ;

Read_6131_Buffer(number_of_words, inc_pointer_first, irq_mgmt) ;

Read_Current_Control_Word(rt_num, irq_mgmt) ;

getMAPAddr() ;

enaMAP(map_num) ;

Read_Current_Control_Word(rt_num, irq_mgmt) ;

Read_RT1_Control_Word(txrx, samc, number, irq_mgmt) ;

Read_RT2_Control_Word(txrx, samc, number, irq_mgmt) ;

ReadWord_Adv4(irq_mgmt) ;

Read_Last_Interrupt(irq_mgmt) ;613x_initialization.h

Fill_6131RAM_Offset() ;

Fill_6131RAM(addr, num_words, fill_value) ;

Memory_watch(address) ;

Configure_ARM_MCU_SPI() ;

```
Spi_register_write();  
Spi_memory_write_ff();  
Spi_memory_write_1fff();
```

board_613x.c

```
ConfigureGpio(); ... ARM MCU の汎用 I/O 初期化  
reset_613x();  
init_timer();  
Delay_us(num_us);  
Delay_ms(num_ms);  
Delay_x100ms(num);  
Flash_Red_LED();  
Flash_Green_LED();
```

board_613x.h;

マクロによって制御される ARM MCU I/O 定義が含まれます。ピンと SPI などの多くは MAMBA と HI-6131 の間で共通しています。これは、どちらも SPI を使用するためです。

BC.h

BC.hには、API用のマクロ、C構造体、プロトタイプがあります

MT.h

MT.hには、API用のマクロ、C構造体、プロトタイプがあります

RT.h, RTMT.h

API用のマクロ、C構造体、プロトタイプがあります

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 :

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

BC モードで使用されるコンソール機能 :

```
bcAsync();  
MajorMinorframe();
```

RT モードで使用されるコンソール機能 :

RTDemo();

Rt_mt();

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 :

SMTDemo();

冗長な文字列を「*printf*」してプログラムサイズを縮小する原始的なコンソール機能 :

print_null(), *print_sp1sp()*, *print_b1sp()*, *print_b0sp()*;

print_ddd(), *print_dd0n()*, *print_dd1n()*;

print_menuprompt(), *print_line()*;

Memory_watch() 関数によって呼び出されるコンソール関数

マニュアル SPI 書込みユーティリティ

ascii2int();

demos.c

コンソール・メニューから呼び出されるすべてのデモ機能を提供します

DisplayDecodedMsg(); … 「T」コマンドが使用されたときに 1553 形式のデータをコンソールに表示するために使用されます。

displayRTTraffic(); … メインから呼び出され、*DisplayDecodedMsg()* を呼び出します

board_lowlevel.c

これは Atmel ボード・ファイルで、フラッシュ・メモリの待機状態を 3SW に変更します

Flash_MAMBA.icf – プロジェクト・リンク・ファイル

MAMBA API LIB.a – プリコンパイルされた Holt API ライブラリ

Holt API Library files

API ヘッダ・ファイルは、ワークスペース Holt_API / sc サブフォルダに含まれ、グループ化されていますが、標準 API キットには Holt API ソース・ファイルは含まれていません。

MAMBA™ SPI Interface [MAMBA™ SPI インターフェイス]

MAMBA™ はホスト MCU または FPGA への 4 線シリアル・ペリフェラル・インターフェイス (SPI) を備えています。このデバイスは、プラスチック QFP または、6mm×6mm QFN パッケージで提供されます。API を使用する場合、以下に示す割り込みの懸念事項は、API ランタイム・ライブラリで既に解決されています。

MAMBA™ データ転送速度は、MCU SPI インターフェイスによって提供される SPI クロック周波数に依存します。SPI が最大 SCK 周波数 40MHz (このデモでは 24MHz を使用) でクロックされると、各 16Bit ワードは 400ns で転送され、SPI オペコードの実行に関するオーバー・ヘッドが加算されます。メモリ・アドレス・ポインタ (レジスタ) は、読出または書込み操作が開始される前に、MCU または FPGA によって初期化されます。その後、8Bit の SPI オペコードを使用してリード/ライト動作が開始され、MCU または FPGA によって MAMBA™ にシリアル・シフトされます。その後ホストは、連続する RAM またはレジスタ・アドレスを読み書きするために、16 クロックの倍数で SCK クロックを継続します。クロックが継続されている限り、連続したアドレスが読み書きされます。マルチワード・アクセス中に割り込みがイネーブルされていると、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 SPI を補足して割り込みを処理すると、未完了のマルチワード転送が中断される可能性があります。適切なソフトウェア設計が無ければ、ハードウェアは割り込みが発生したことを知らず、メモリ・アドレス・ポインタは中断されたマルチワード転送の次の位置の RAM またはレジスタ・アドレスを含んでも含まなくてもよいため、割り込みからの簡単な復帰により、マルチワード転送が壊れます。

MAMBA™ SPI 転送中は、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、オペコードを送信する前に割り込みを無効にし、マルチワード転送で最後のワードをリード/ライト後に割り込みを再び有効にします。これらが受け入れがたい割り込み遅延を引き起こす場合は、慎重なソフトウェア設計が必要です。適切な注意を払うことで、割り込みを一時的に再び有効にしてから、SPI ワード間ですぐに無効にすることができます。割り込みが再有効されると、割り込みインターバル中に発生した保留中の割り込みがすぐに認識されます。保留中の割り込みサービス・ルーチンが実行されます；割り込みからの復帰は次の「割り込み禁止」ステートメントにジャンプして実行します。

MAMBA™ のソフトウェア例では、割り込みが一時的に無効にされてから SPI ワード間で再度有効になるたびにテストされた「SPI 割り込み発生」フラグを使用して、中断されたマルチワード・シーケンスを正常に完了します。ワード間の割り込み処理を検出すると、プログラムは次のワードのメモリ・アドレス・ポインタを再初期化し、次の元のオペコードを再発行して中断したマルチワード転送を再開します。3 つ以上のレベルの割り込みをネストするのは難しいです。

MAMBA™ を使用する場合、SPI アクセスはテーブル・アドレスの C 構造体と互換性がありません。MAMBA™ を使用しているときは、レジスタ/RAM 検査、ウォッチ・ウィンドウは使用できません。ユーティリティ機能の代わりに、HI-6135/6/7/8 デモ・プログラムの *Memory_watch()* 関数のように、アドレス範囲を読み取るためには C で書かれていなければならず、コンソール I/O またはその他の表示手段を用いて表示データをフォーマットします。もちろん *Memory_watch()* が必要な場合は、アプリケーションプログラムが表示機能呼び出すために実行されている必要があります。上級ユーザー向けには、レジスタやメモリ・アドレスへの SPI 書込み用に他のユーティリティも用意されています。

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

MAMBA™ は、SPI を搭載したマイクロコントローラとの互換性を考慮して設計されています。RAM とレ

レジスタの位置は、8Bit SPI コマンドの助けを借りてリード/ライトされます。大部分のリード/ライト操作は、4つのメモリ・アドレス・ポインタ (MAP) のうちの1つを使用して、次にアクセスされる場所のアドレスを指定します。マルチワード転送を高速化するために、有効にされたメモリ・アドレス・ポインタは、各リード/ライトが実行されると自動的に次のアドレスにインクリメントされます。レジスタ・アドレス 0~15 十進は、メモリ・アドレス・ポインタを使用せずに直接読み取ることができます。レジスタ・アドレス 0~63 十進は、メモリ・アドレス・ポインタを使用せずに直接書き込むことができます。

デバッグ時に、このツールは MAMBA™ SPI ツールでは機能しないため、IAR Embedded Workbench® デバッガから見ることはできないので、レジスタまたは、RAM の値を監視するのに役立ちます。デモ・プログラムは、*Memory_watch()* という C 関数を使用して SPI を介して同様の機能を提供します。この関数呼び出しは、コンソール I/O が有効な場合にのみ機能します。提供されたメモリ・アドレス・パラメータで始まる 256 の連続するレジスタまたは RAM 値を表示します。メモリ・アドレス空間 0~0x1FFF 全体は、256 ワード単位でアクセス可能です。デモ・プログラムはキーボード入力をポーリングし、実行する必要があります。コンソール・メニューの「W」コマンドが入力されると、0x0000~0x00FF メモリ・アドレス空間が表示されます：

```

Press 'M' for menu, or press any valid menu key. >>
 0      1      2      3      4      5      6      7      8      9      A      B      C      D      E      F
Adr 0000: 0040 8000 0000 0000 0000 0000 0000 0000 0000 0000 0180 000B 0000 0000 0000 6018
Adr 0010: 0000 0000 0408 6018 0000 0000 0408 0080 1C00 0400 0000 0000 0000 0000 0000 0000
Adr 0020: 0000 0000 0600 0000 0000 0000 0000 0000 0000 0000 0001 0000 0000 0000 0000 0000
Adr 0030: 0000 0000 0000 0000 0000 0000 0000 0000 0000 00C0 0022 3D54 0028 0000 0000 0000
Adr 0040: 0000 0000 0000 5941 0000 0000 0000 0000 0000 0000 5ED8 0000 0000 0000 3100 0000
Adr 0050: 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
Adr 0060: 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
Adr 0070: 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
 0      1      2      3      4      5      6      7      8      9      A      B      C      D      E      F
Adr 0080: 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
Adr 0090: 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
Adr 00A0: 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
Adr 00B0: 0D00 0D80 0DF0 0000 0E00 0F00 0FFF 0000 1200 1280 12FF 0000 1000 1100 11FF 0000
Adr 00C0: 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
Adr 00D0: 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
Adr 00E0: 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
Adr 00F0: 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
=====
Keys: <W>atch On/Off <D>own <U>p <R>efresh <A>ddress <M>enu 0x0000-0x00FF
=====
Press 'M' for menu, or press any valid menu key. >> w

```

画面下部のサブメニューには、使用可能なメモリ・ウォッチ・オプションがリストされています。「D」(DOWN コマンド) を押すと、表示されているアドレス範囲が 0x0100~0x01FF に変わります。上記の画面から「U」(UP コマンド) を押すと、デバイスのアドレス空間がラップアラウンドされ、表示されているアドレス範囲が、0x1F00~0x1FFF に変更されます。UP または DOWN コマンドを繰り返すと、アドレス範囲内を移動します。「R」を押すと現在選択されているアドレス範囲がリフレッシュされ、「A」(ADDRESS コマンド) を押すと 4 つの 16 進文字を入力して、メモリ・ウォッチの開始アドレスを選択できます。「W」(WATCH)、「M」(MENU) を押すと、メモリ・ウォッチ表示がトグルし、5 ページのメニューに戻ります。

Memory_watch() が実行されると、表示されている各位置が再スキャンされることに注意してください。いくつかのレジスタ RAM 構造体ビットは、各読み出しが発生した後に自動的にリセットされます。これには、保留割り込みレジスタのビットと RAM の RT デスクリプタ・テーブル・コントロール・ワードの DBAC データ・ブロック・アドレス・ビットが含まれます。これらのために、メモリ・ウォッチ・ウィンドウは、関数が実行されたときに有効な値を反映します。

Tera Term を使用したコンソール I/O オプションには、ペンディング割り込みレジスタのステータスを読んで表示するいくつかのメニュー・オプションがあります。ペンディング割り込みビットは、読み出しが発生した後に自動的にリセットされることに注意してください。これらのレジスタの場合、メモリ・ウォッ

チ・ウィンドウは、実行が停止したときに有効な値が反映されます。

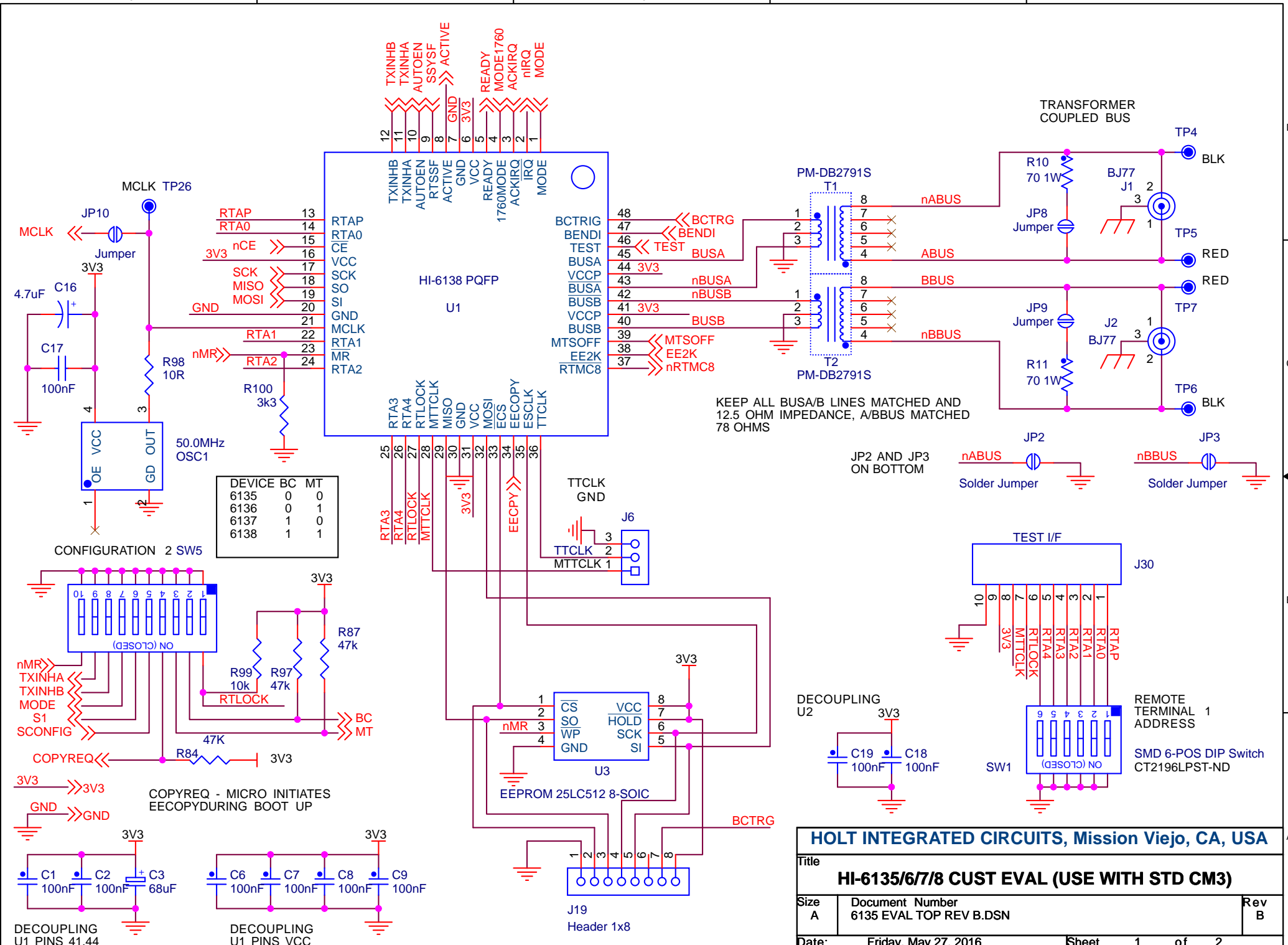
MAMBA™ デモ・プログラムは、MAMBA™ ファミリーのすべてのデバイスを対象としています。デフォルト設定では BC、RT、SMT すべてが有効になっています。

まとめ

MAMBA™ SPI インターフェイスは、RAM またはレジスタにアクセスするためのわずか 4 つのホスト・インターフェイス信号を使用して、MIL-STD-1553 ターミナルのハードウェア設計を簡素化し、同様の 16Bit パラレル・アドレスおよび、データ・バス・インターフェイスに相当するボード・スペースを節約します。Holt 1553 API ランタイム・ライブラリは、プロジェクト開発を大幅に加速するハイレベルの C 関数呼び出しを提供することで、MAMBA™ プログラミングを簡素化します。

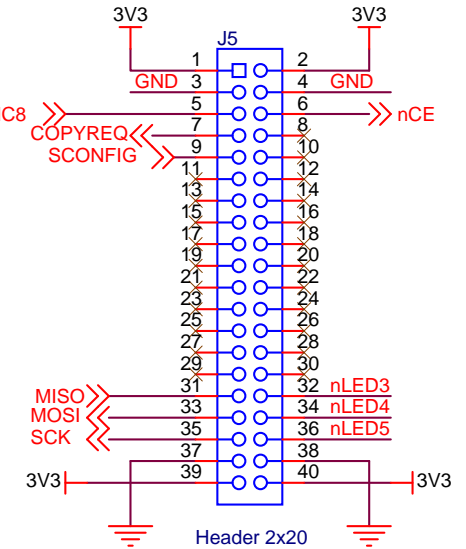
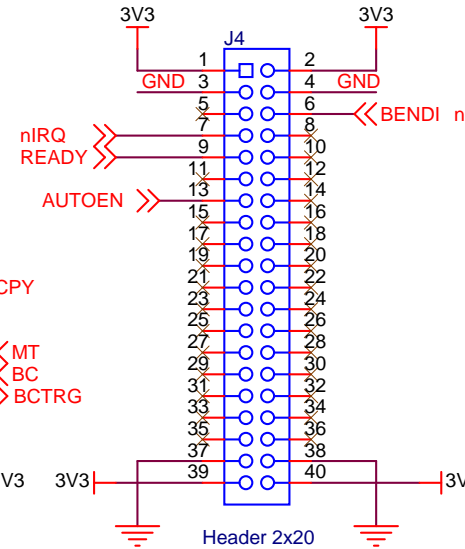
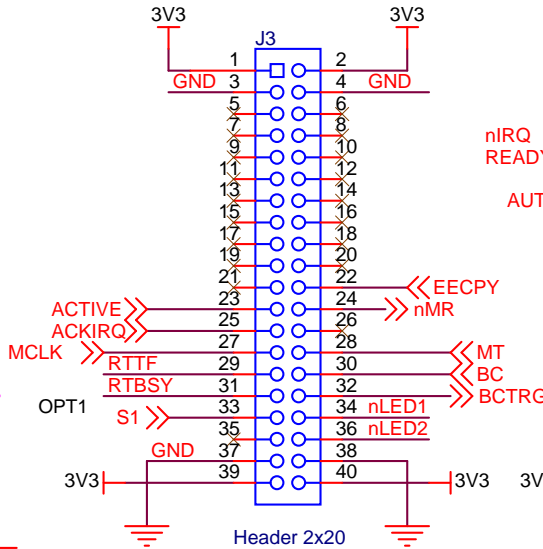
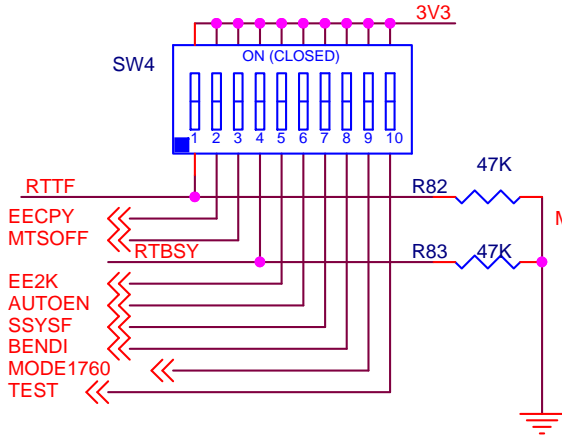
Bill of Materials
Mamba Evaluation board
Rev. B

Item	Qty	Description	Reference	DigiKey	Mfr P/N
1	1	PCB, Bare, Eval Board	N/A	-----	
2	10	Capacitor, Cer 0.1uF 20% 50V Z5U 0805	C1,C2,C6,C7,C8,C9,C17,C18,C19,C20	399-1176-1-ND	Kemet C0805C104M5UACTU
3	1	Capacitor, Cer 4.7uF 10% 6.3V X5R 0805	C16	399-3134-1-ND	Kemet C0805C475K9PACTU
4	2	Capacitor 68uF 10% 6.3V Tant 400 mOhm SMD EIA 6032-28	C3,C26	495-1507-1-ND	T495C686K006ZTE400
5	2	Connector 3-Lug Concentric Triax Bayonet Jack, Panel Front Mount TRB (BJ77)	J1,J2	MilesTek 10-06570	Trompeter Electronics BJ77
6	2	Header, Male 2x20 0.1" Pitch, 0.230" Pins, 0.120" Tails	J3,J4	S2012E-20-ND	Sullins PEC20DAAN
7	2	Header, Male 2x5, 0.1" Pitch, 0.230" Pins, 0.120" Tails	J5A,J5B	S2012E-05-ND	Sullins PEC05DAAN
8	1	Header, 1x10, 0.1" pitch	J6	DO NOT STUFF	
9	1	Header, 1x8, 0.1" pitch	J19	DO NOT STUFF	
10	1	Header, 1x3, 0.1" pitch	J6	DO NOT STUFF	
11	5	Solder Jumper	JP2,JP3,JP8,JP9,JP10	DO NOT STUFF	
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 25ppm 3.3V SMD 5x7mm	OSC1	535-10087-1-ND	Abracon ASV-50.000MHZ-E-T
16	2	Res 69.8 Ohm 1W 1% 2512 SMD	R10,R11	RHM69.8BBCT-ND	Rohm MCR100JZHF69R8
17	5	Resistor, 150 Ohm 5% 1/8W 0805	R8,R9,R12,R80,R96	P150KACT-ND	Panasonic ERJ-6GEY0R151V
18	1	Resistor, 10 Ohm 5% 1/8W 0805	R98	P10ACT-ND	Panasonic ERJ-6GEY0R100V
19	1	Resistor, 10K 5% 1/8W 0805	R99	P10KACT-ND	Panasonic ERJ-6GEYJ103V
20	5	Resistor, 47K 5% 1/8W 0805	R82,R83,R84,R87,R97	P47KACT-ND	Panasonic ERJ-6GEYJ473V
21	1	Resistor, 3.3k 5% 1/8W 0805	R100	P3.3KACT-ND	Panasonic ERJ-6GEYJ332V
22	1	DIP Switch 6-Position SMD	SW1	CT2196MST-ND	CTS 219-6MST
23	2	DIP Switch 10-Position SMD	SW4,SW5	CT21910MST-ND	CTS 219-10MST
24	2	Transformer MIL-STD-1553 Single, 1:2.50,	T1,T2	Holt PM-DB2791S	Holt / Premier Magnetics
25	3	Test Point, Red Insulator, 0.062" hole	(+)BusA, (+)BusB, 3V3	5010K-ND	Keystone 5010
26	3	Test Point, Black Insulator, 0.062" hole	(-)BusA, (-)BusB, GND	5011K-ND	Keystone 5011
27	1	Test Point, White Insulator, 0.062" hole	TP8 (Active)	5012K-ND	Keystone 5012
28	1	IC HI-6135/6/7/8 Holt 48-PQFP	U1	HOLT IC	Holt IC
29	1	IC, Serial EEPROM 512Kbit 20MHz SPI 8SOIC, Microchip	U3	25LC512-I/SN-ND	Microchip 25LC512-I/SN
30	1	Hookup Solid wire - 20AWG - Black - 4" Long per Board	For J1 and J2	C2028B-XX-ND	General Cable C2028A.12.01

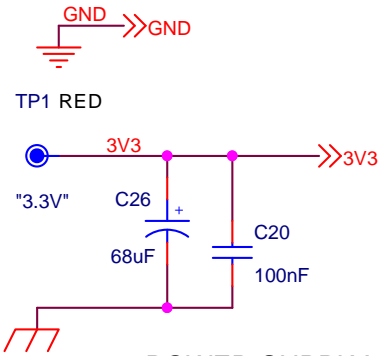


HOLT INTEGRATED CIRCUITS, Mission Viejo, CA, USA		
Title		
HI-6135/6/7/8 CUST EVAL (USE WITH STD CM3)		
Size	Document Number	Rev
A	6135 EVAL TOP REV B.DSN	B
Date:	Friday, May 27, 2016	Sheet 1 of 2

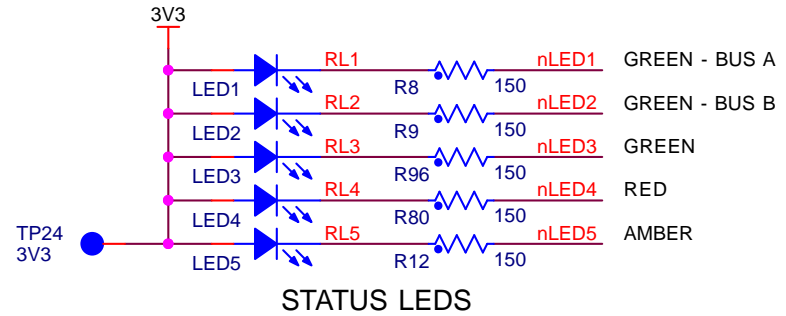
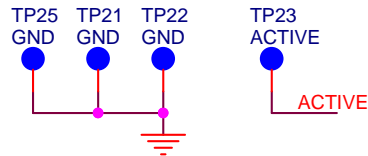
FOR ALL DIP SWITCHES
UP = LOGIC-1
CT21910LPST-ND
HW CONFIGURATION 1



RTTF - SETS RT TERMINAL FLAG
EECPY - MAKES COPY OF RAM AND REGS TO EPROM
MTSSOFF - DISABLE MEMORY TEST ON POWER UP
RTBSY - SETS BUSY BIT OF STATUS WORD
EE2K - SET HIGH EEPROM SAVES ONLY LOWER 2K WORDS
AUTOEN - LOADS FROM EEPROM ON POWER UP
SSSYF - SET SSSYF (SYSTEM FAIL) BIT IN RT STATUS WORD
BENDI SET HI: SPI USES BIG ENDIAN FORMAT
MODE1760 - POWERS UP WITH BUSY BIT SET
TEST - ENABLES TEST MODE, SEE DATA SHEET



POWER SUPPLY -- 5VDC IN TO 3.3VDC OUT



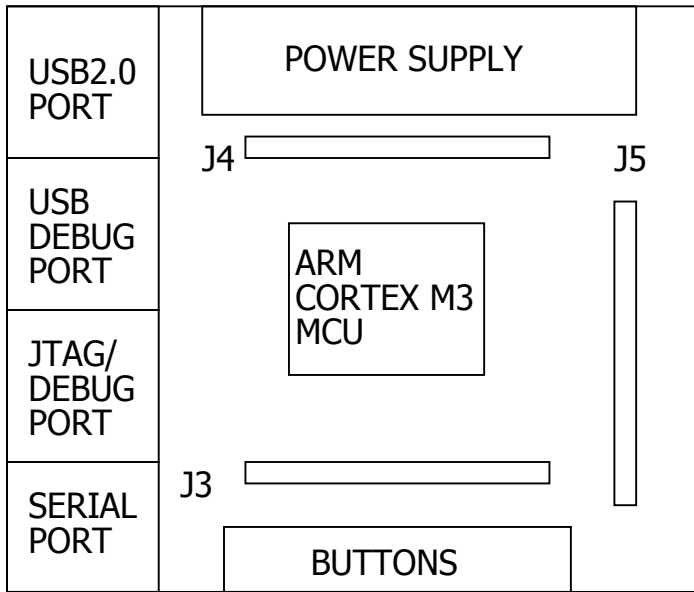
STATUS LEDES

HOLT INTEGRATED CIRCUITS, Mission Viejo, CA, USA

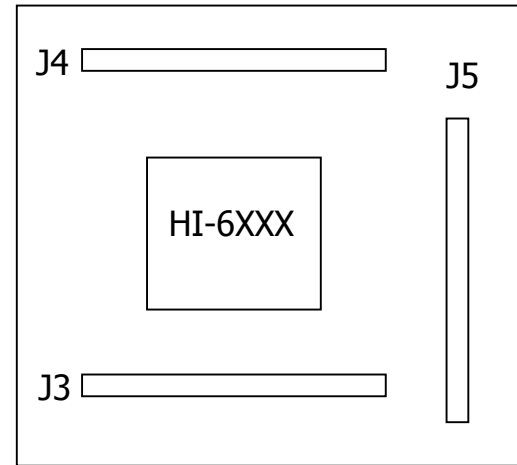
Title		
HI-6135/6/7/8 CUST EVAL USE WITH STD CM3		
Size	Document Number	Rev
A	<Doc>	B
Date:	Friday, May 27, 2016	Sheet 2 of 3

Bill of Materials
ARM Cortex M3 MCU Board
Rev. E

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% NPO C0G 0V 0603	C23,C34	490-1403-1-ND	Murata GRM1885C1H100JA01D
6	4	Capacitor, Ceramic 20pF 5% NPO 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	1	Connector, Receptacle USB Mini B Rt-Angle PCB Mount	J2	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	1	LED Green 0805	LED1	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	1	4-Ch TVS ESD Protection SOT23-6	U2	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	5	Rubber Foot, Bump on Black Hemisphere, .312 X.200 H	Place at 4 corners and center	SJ5746-0-ND	3M SJ61A1
47	1	Capacitor, Ceramic 100nF, -20% / +80% 25V Y5V 0603	C66	490-1575-1-ND	Murata GRM188F51E104ZA01D
48	1	Capacitor, Ceramic 33pF, 5% 50V C0G 0603	C59	490-1415-1-ND	Murata GRM1885C1H330JA01D
49	2	Capacitor, Ceramic 15pF, 5% 50V C0G 0603	C60,C61	490-1407-1-ND	Murata GRM1885C1H150JA01D
54	1	Ferrite Bead, 220 Ohm @ 100MHz 300mA DC 0805	FB2	732-1602-1-ND	Wurth 742792034
55	1	Solder Jumper	JP2	SOLDER OPEN	
56	1	Connector, Receptacle USB Mini B Rt-Angle PCB Mount	J8	H2959CT-ND	Hirose UX60-MB-5ST
57	1	LED Green 0805	LED2	160-1179-1-ND	LiteOn LTST-C170GKT
59	1	Resistor, 220 ohm 5% 1/10W 0603	R31	P220GCT-ND	Panasonic ERJ-3GEYJ221V
63	2	Resistor, 27 ohm 5% 1/10W 0603	R36,R38	P27GCT-ND	Panasonic ERJ-3GEYJ270V
66	1	4-Ch TVS ESD Protection SOT23-6	U4	296-28203-1-ND	TI TPD4E001DBVR



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 E.DSN	E
Date:	Wednesday, June 01, 2016	Sheet 1 of 7

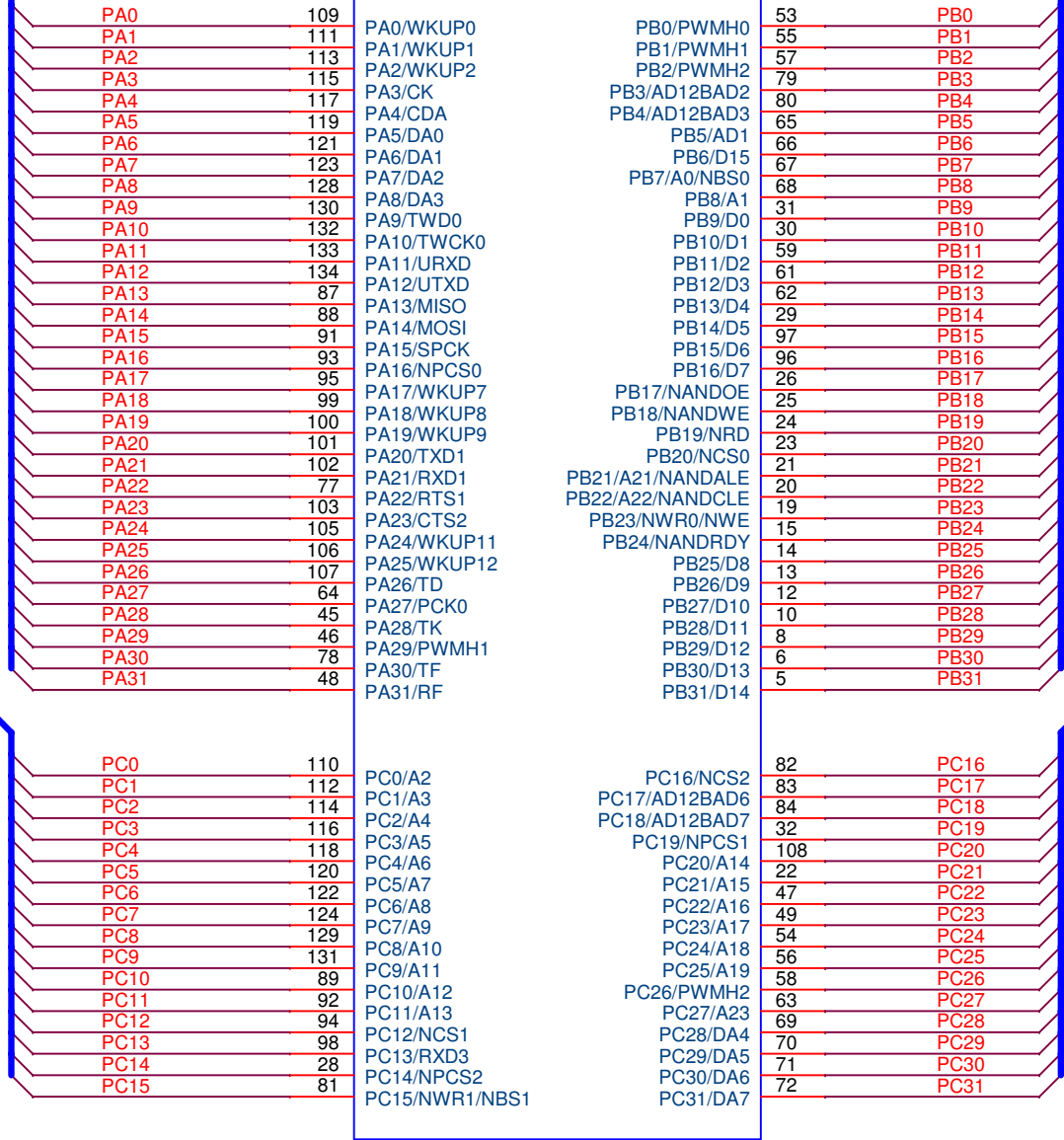
PA[31:0]

PB[31:0]

PC[31:0]

PC[31:0]

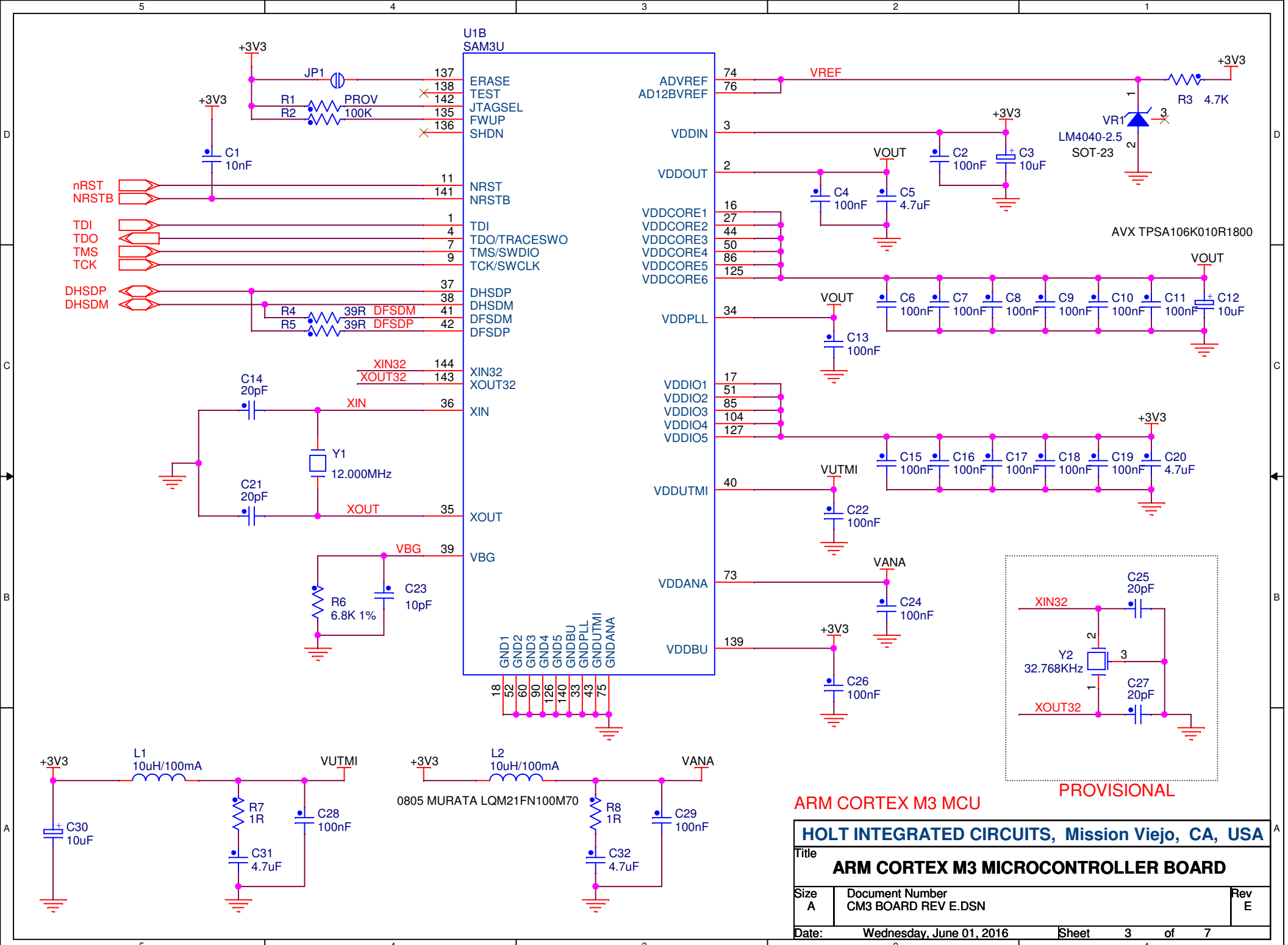
U1A
SAM3U



ARM CORTEX M3 PIO

HOLT INTEGRATED CIRCUITS, Mission Viejo, CA, USA

Title		
ARM CORTEX M3 MICROCONTROLLER BOARD		
Size	Document Number	Rev
A	CM3 BOARD REV E.DSN	E
Date:	Wednesday, June 01, 2016	Sheet 2 of 7



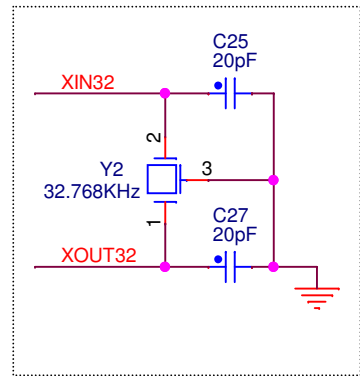
ARM CORTEX M3 MCU

HOLT INTEGRATED CIRCUITS, Mission Viejo, CA, USA

Title
ARM CORTEX M3 MICROCONTROLLER BOARD

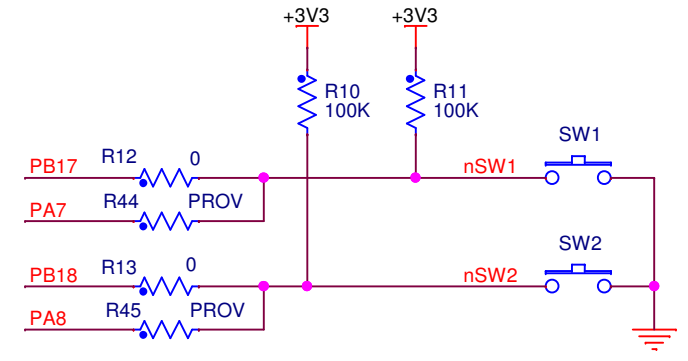
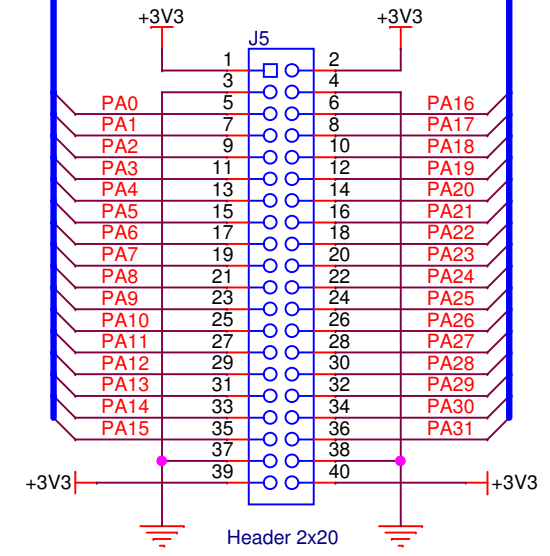
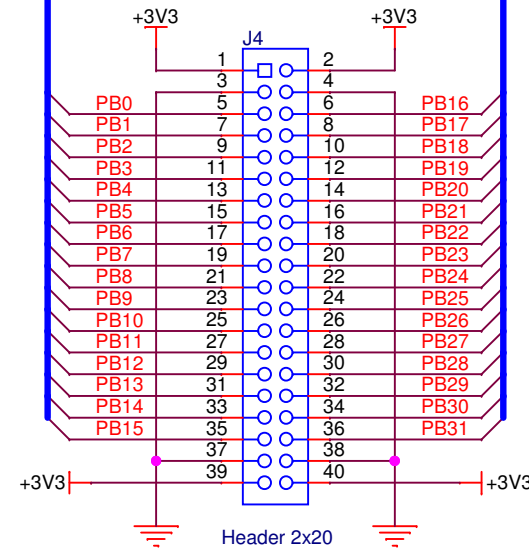
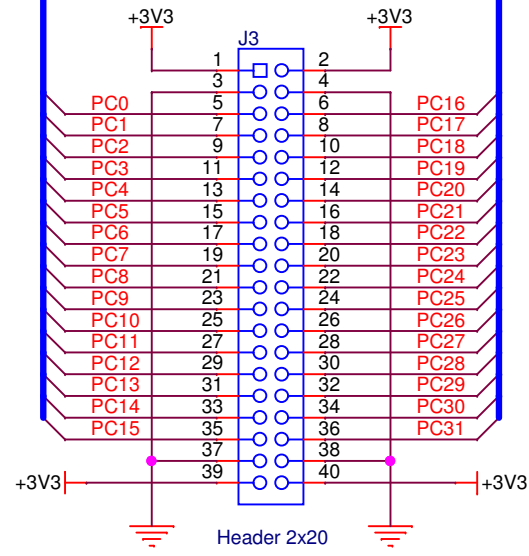
Size A	Document Number CM3 BOARD REV E.DSN	Rev E
-----------	--	----------

Date: Wednesday, June 01, 2016 Sheet 3 of 7



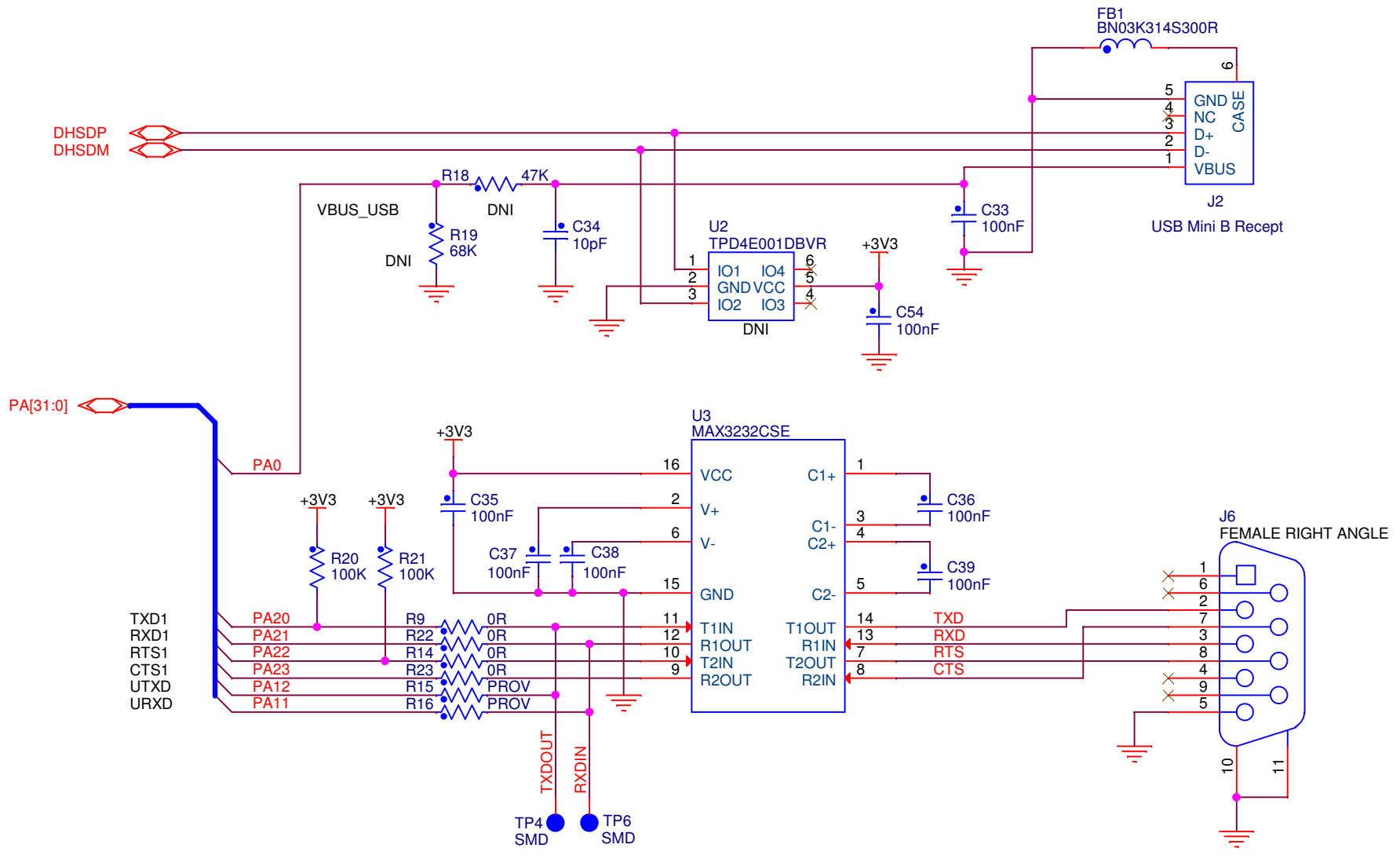
PROVISIONAL

{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	Document Number	Rev
A	CM3 BOARD REV E.DSN	E
Date:	Wednesday, June 01, 2016	Sheet 4 of 7

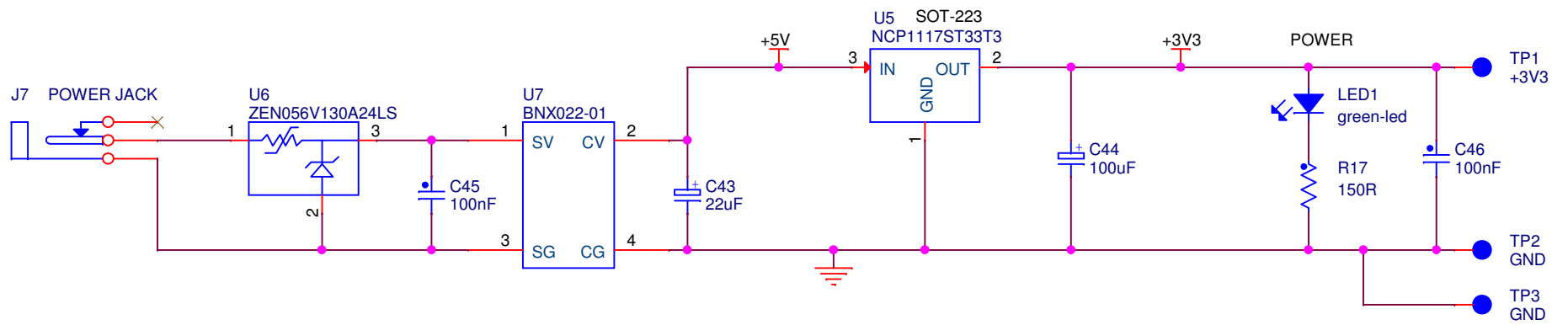


USB & RS-232 SERIAL

HOLT INTEGRATED CIRCUITS, Mission Viejo, CA, USA

Title		
ARM CORTEX M3 MICROCONTROLLER BOARD		
Size	Document Number	Rev
A	CM3 BOARD REV E.DSN	E

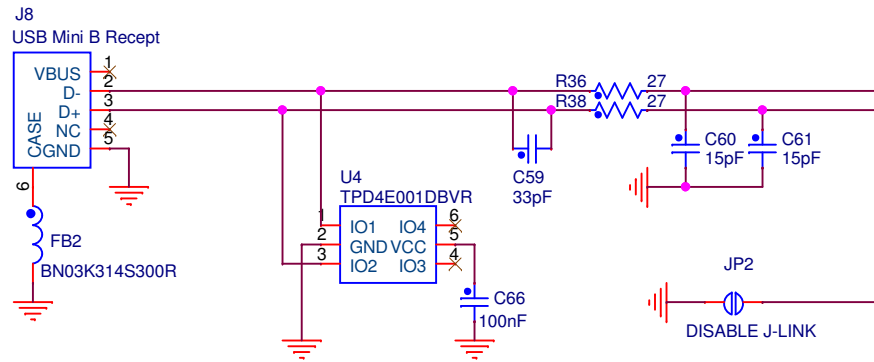
Date: Wednesday, June 01, 2016 Sheet 5 of 7



POWER SUPPLY

HOLT INTEGRATED CIRCUITS, Mission Viejo, CA, USA		
Title		
ARM CORTEX M3 MICROCONTROLLER BOARD		
Size	Document Number	Rev
A	CM3 BOARD REV E.DSN	E
Date:	Wednesday, June 01, 2016	Sheet 6 of 7

USB DEBUG INTERFACE



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

(CONFIDENTIAL)

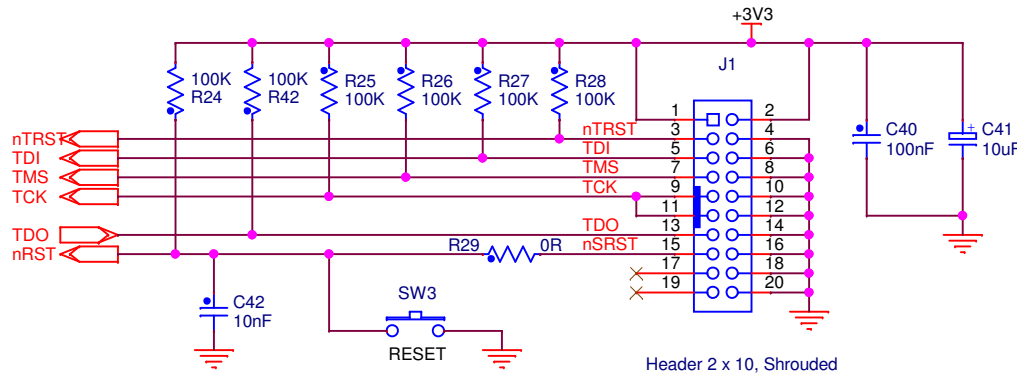
NOT PART OF A CUSTOMER DESIGN,
THIS BLOCK IS COMPRISED OF U8,
Y3, C47-C53, C55-C58, C62-C65, R30,
R32-R35, R37, R39-R41 AND R43.

- 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
Custom	CM3 BOARD REV E.DSN	E
Date:	Wednesday, June 01, 2016	Sheet 7 of 7