HI-6131 API
Application Development Kit

December 2018

REVISION HISTORY

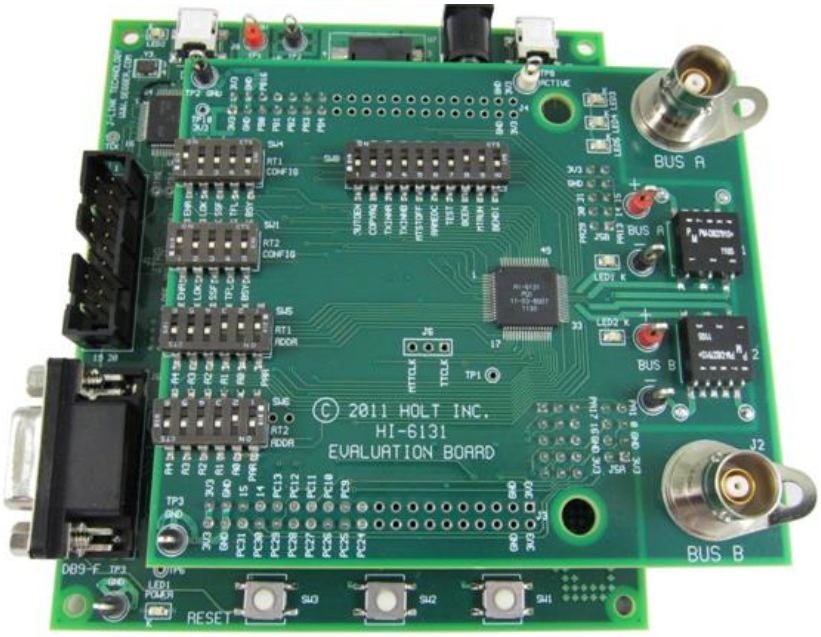
Revision	Date	Description of Change
AN-6131API, Rev. New	06-07-16	Initial Release
AN-6131API, Rev. A	12-12-18	Rename API Library file to coincide with software update. HI-613x API LIB.a

Introduction

The Holt ADK-6131API Application Development Kit demonstrates the broad feature set of the HI-6131 Multi Terminal IC for MIL-STD-1553. The 2-board assembly and C project software reference design provide a ready-to-run evaluation platform demonstrating concurrent operation for any combination of Bus Controller, Bus Monitor and one or two Remote Terminals. For convenience, this kit includes IAR Systems Embedded Workbench® for ARM, and a fully integrated debug interface for the ARM Cortex M3 microcontroller.

This guide describes how to set up and run the board. Additional support material and all required project software are found in the included Holt CD-ROM. A version of the demonstration software is already programmed into the microcontroller flash; the board is operational right out of the box without installing or running the provided software development tools.

The ADK-6131API ADK demonstrates the Holt API library on the same hardware as the ADK-6131 development kit. The ADK-6131 kit development uses low level 6131 drivers and demonstration code written in ANSI 'C' without high level APIs. See document AN-6131.pdf for more information on this kit.

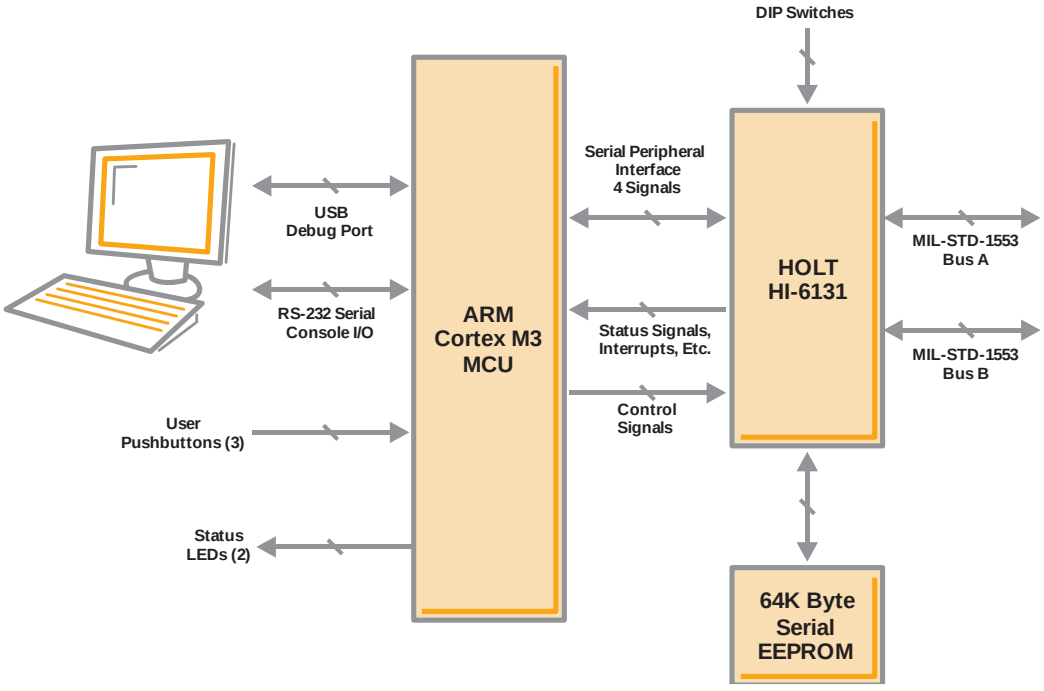


This guide describes how to set up and run the board. Additional support material and all required project software are found in the included Holt CD-ROM. A version of the demonstration software is already programmed into the microcontroller flash; the board is operational right out of the box without installing or running the provided software development tools.

Evaluation Kit Contents

- This User Guide.
- Holt Demo and Holt API Software Projects and Documentation CD.
    - HI-613x API LIB runtime library
    - HI-6131 API Demo example project
- Installation CD for IAR Systems *Embedded Workbench® for ARM*, version 7.1 or greater. See installation guide "Holt HI-6131 API project installation guide" for important instructions.
- Plug-in 5V DC power supply and USB debug interface cable.
- RS-232 serial cable, DB-9M to DB-9F for console I/O using a connected computer.
- 2-board assembly comprised of:
    - Upper TARGET board with HI-6131 device and dual transformer-coupled MIL-STD-1553 bus interfaces. Numerous DIP switches configure board operation.
    - Lower MCU board with ARM Cortex M3 16-/32-bit microprocessor, debug interface and regulated 3.3VDC power supply

Hardware Block Diagram



Hardware Design Overview

Separate schematics and BOM's documents are included on the Holt CD-ROM.

The detachable HI-6131 board can be separated from the provided MCU board for connection to a user-supplied alternate microprocessor or FPGA board. The inter-board headers are located on 0.1" (2.54 mm) grid for compatibility with generic prototyping boards. All host interface signals go through the inter-board headers. Numerous HI-6131 configuration pins (and the Remote Terminal address setting pins) are controlled by DIP switches on the upper HI-6131 board; these signals are not available on the inter-board headers.

The lower ARM Cortex M3 board is based on the flash-programmable Atmel AT91SAM3U-EK microprocessor. A 4-signal Serial Peripheral Interface (SPI) clocked at 16MHz (20MHz is possible) connects to the HI-6131. A UART-based serial port provides RS-232 console I/O (optional). An uncommitted USB 2.0 port is available for future expansion. Two pushbuttons are available for software interaction. A RESET pushbutton resets the ARM microprocessor, which in turn controls the HI-6131 Master Reset signal.

The ARM Cortex M3 board includes "J-Link On Board" debug interface, licensed from www.segger.com, providing out-of-box readiness without having to buy a costly JTAG debug cable. The kit includes a simple USB cable for connecting the board's debug interface to your computer. (For users already owning an ARM debug interface with ribbon-cable connector, an ARM-standard 2x10 debug connector provides debug connectivity. In this case, jumper JP2 on the bottom of the lower board should be soldered closed to disable "J-Link On Board".)

The 64K serial EEPROM is not demonstrated on this kit. For information on the AUTO initialization feature of the HI-6131 using the EPPROM see document AN-6131.pdf for the non Holt API version of the demo kit software.

Holt API Host Memory Considerations

Holt API uses C malloc() functions to allocate memory from ARM Cortex M3 internal SRAM for API host buffers. The total amount of SRAM available on the selected Atmel ARM Cortex M3 MCU is 48K bytes. This SRAM is shared for all C code static, dynamic variables including the stack and heap. The 48K bytes are adequate for the demonstrations in this software. User SRAM requirements depend on number of enabled 1553 terminal modes and buffer size(s) needed for the application.

If more memory is required, choose a processor with more SRAM or augment the processor with external SRAM. The Holt ADK-6130-2 board is an example that uses the same Atmel ARM Cortex M3 processor with an additional 256K byte external SRAM memory. See AN-6130-2.pdf for this example. Note: The ADK-6130-2 uses a Holt HI-6130 protocol IC with parallel bus interface (not SPI) and the SRAM memory expansion also uses the MCU external bus interface.

There are alternative MCUs with more internal SRAM. For example NXP offers “Kinetis” ARM Cortex M4 microcontrollers with up to 256K bytes internal SRAM. NXP offers the Kinetis ARM Cortex M4 in SPI-only (as well as SPI and External Bus Interface) configuration. A SPI-only selection could replace the ARM Cortex M3 on the board we are using here, with up to 256K bytes internal SRAM.

Initial Kit Set Up

The Application Development Kit is pre-programmed to concurrently operate as a Bus Controller, SMT Bus Monitor and two independent Remote Terminals. Terminal addresses for the two RTs are preset using DIP switches, before applying power. The two 6-position DIP switches should already be set with these address values, plus odd parity. The RT1 DIP switches should be set to RT address 3 (0-0-0-1-1-parity 1), RT2 should be set to RT address 1 (0-0-0-0-1-parity 0), to work with the demonstration program utilized by the preprogrammed Bus Controller message repertoire. The two 6-position DIP switches should already be set with these address values, plus odd parity and match the board photo on page 2. The user’s guide, source code and software documentation sometimes refers RT1 as just RT.

1. The PC will need a serial (COM) port and a “terminal emulation” program like TeraTerm. Most computers no longer have RS232 com ports so will require a serial-to-USB adapter, supplied with the ADK. Connect this to the computers USB port and the 9 pin connector to the ADK board.
2. If using Windows 2000 or Windows XP, you can use HyperTerminal for terminal emulation. Open HyperTerminal by clicking **Start** then **All Programs;** click the Windows **Accessories** then **Communications** program group. Double-click HyperTerminal to run it. Skip the next paragraph.

   If using Vista or Windows 7…
   HyperTerminal is not included with these versions of Windows. Install the free open-source terminal emulation program, *TeraTerm 4.71*, by running the provided teraterm-4.71.exe installer program from the Holt CD. Accept the license agreement stating redistribution is permitted provided that copyright notice is retained. The notice can be displayed from the TeraTerm window by clicking **Help** then clicking **About TeraTerm**. Continuing to install…
   - Accept the default install destination and click **Next**.
   - At the Select Components screen, unselect all options except Additional Plug-in = TTXResizeMenu and click **Next**.
   - Select the installed language, then click **Next**.
   - Accept the default Start Menu folder, then click **Next**.
   - Select any desired shortcuts, then click **Next**.
   - At the Ready to Install screen, click **Install**.

Run the TeraTerm program. At the **New Connection** screen, select **(x)Serial** and choose the selected COM port. Click **Setup** then **Serial Port** to open the serial port setup window. Choose these settings: Baud Rate: 115200, Data: 8 bits, Parity: none, Stop: 1 bit, Flow Control: none.

3. After configuring the Console port, connect the DB-9 serial cable to the PC.

4. Do not connect the included USB debugger cable between the DEBUG port on the lower (MCU) board and the PC until instructed to, later.

5. To observe bus activity, connect an oscilloscope to the red test points labeled BUS A and BUS B. The test point labeled ACTIVE is a convenient scope trigger signal that goes high at start of message and goes low at message completion.

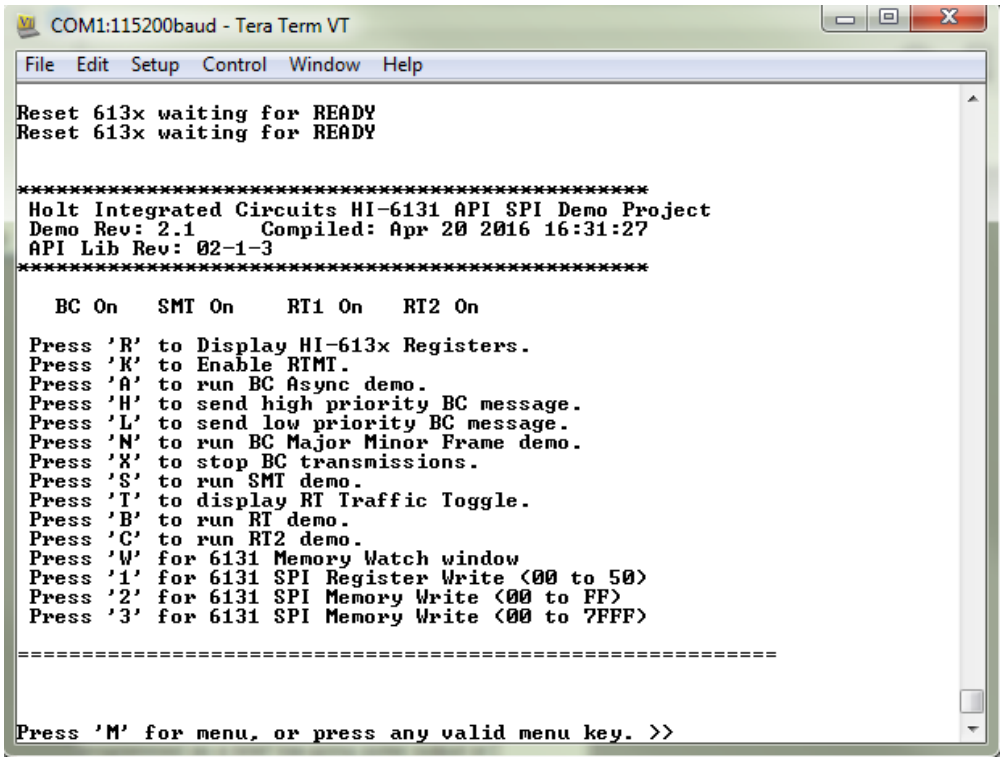
6. If not connected by cable to actual MIL-STD-1553 buses, provide resistive dummy loads for buses A and B by connecting a 75 Ω 1/2 Watt (or any value between 70-80 Ω) resistor across each pair of red and black Bus test points. (For this demonstration, half-Watt resistors are adequate because duty cycle is sufficiently low. When using the on-chip HI-6131 to generate BC messages directed to on-chip RTs, use external 75 Ω resistor loads. When using a bus coupler to connect to actual MIL-STD-1553 buses, do not use the 75 Ω dummy load resistors.

7. Set SW8 configuration DIP switches labeled AUTOEN and COPYREQ off (down position). These switches are not used by this version of the demo program so they can be repurposed by the user.

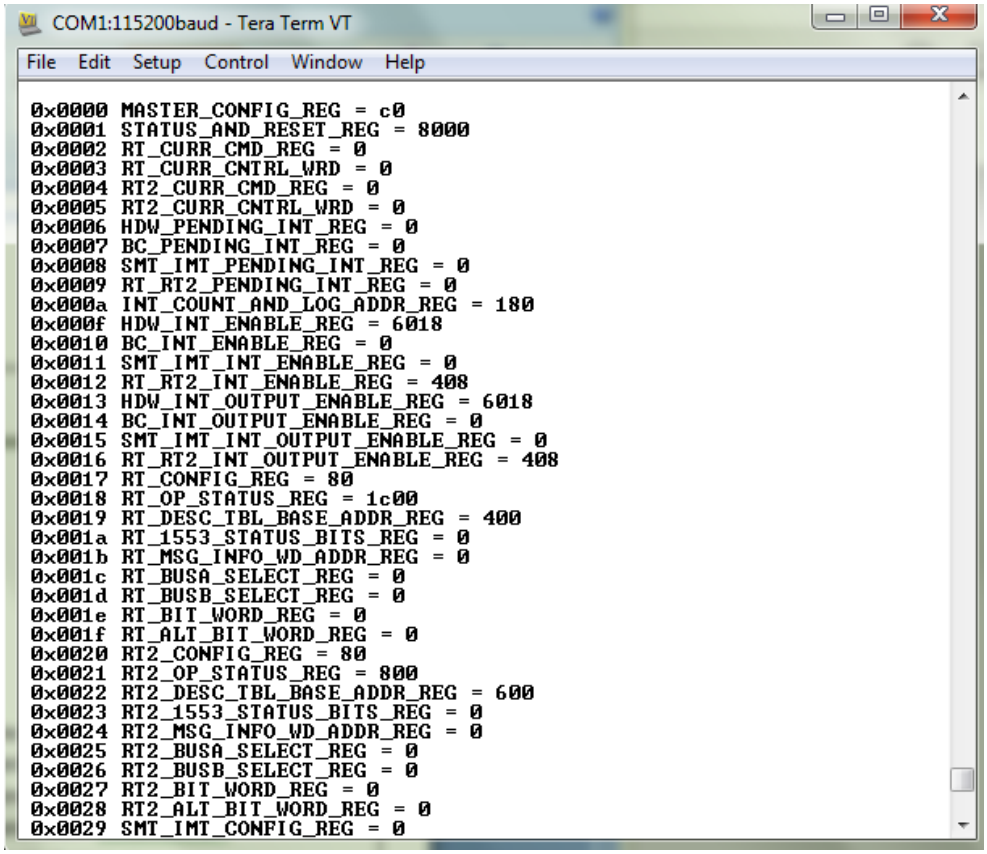
8. Plug-in the provided 5V DC power supply and connect the cable to the power input jack on the lower circuit board. If TeraTerm is running and configured correctly, the command menu below should appear in the console window.

9. This menu appears and three green LEDs will flash momentarily whenever the board power is applied, or when the RESET pushbutton is pressed. After verifying correct *TeraTerm* communication with the evaluation board, the terminal set up can be saved by clicking **Setup** then **Save Setup**.

The dates and times shown will differ from the screen captures shown below.



Press 'R' or 'r' to display the HI-613x registers.



General Structure of Demo Functions

The Holt API demonstration program is contained in module `demos.c`. The Holt API runtime library is contained in the library file "`HI-613x API LIB.a`" as executable object code. File `demos.c` contains the demo initialization API function calls supporting demonstrations executed from the console menu to initialize the BC, RT,RT2 and monitor terminals. Key presses are detected in `console.c` which is called from the main loop in `main.c` and executes demo functions in `demos.c`.

Commands 'A' and 'N' transmit BC commands can be viewed on an oscilloscope and optionally display the message traffic data on the console using the 'K' and 'T' command sequence. These demos demonstrate how Holt API's are used to generate BC Asynchronous messages, Major/Minor frames, low priority and high priority messages. View these messages with external MIL-STD-1553 test equipment or view them with an oscilloscope.

This exercise uses the internal BC to transmit messages, so message traffic data is displayed on the console. Since the internal BC, RTs and SMT share the same bus pins, the RT's and SMT monitor terminals receive the BC messages. If an external BC is already connected to the bus jack though a bus coupler, it is okay to leave it connected, but disable any external BC transmissions that will conflict with the on-chip BC transmissions.

For the following demonstrations all four terminals in the HI-6131 API demo program are enabled and initialized by console commands.

BC and RT MT Mode.

This demonstrates BC, RT and MT all together and displays the traffic data on the console.

1. Press command 'B' to enable the RT.
2. Press command 'K' to enable the RTMT demo.
3. Press command 'T' (or spacebar) to display RT traffic on the console. Command 'T' toggles on and off alternately to enable or disable the RT traffic shown on the console. Using the 'T' command relies on prior execution of command 'K'.
4. Press command 'A' to start the BC transmitting messages.

   Messages will display rapidly on the screen, Press the space bar to stop the console output. The console should freeze and look similar to the screen below. Press space bar again to restart the console output. Using the space bar does not stop BC transmission or prevent RT or MT message reception; it only stops console output.

```
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
```

Some of the BC commands are RT to RT. The BC commands to RT3 only shows no errors but RT to RT commands to RT1 and RT3 will show " ERROR: NORES" (no response) since there's no RT at address 1.

To see these same messages without the "ERROR:NORES" enable RT2 by pressing the 'C' command and repeat the messages. RT2 will now receive and acknowledge the messages to RT2 which is set to RT address 1 by DIP switches.

The Bus A green LED flashes rapidly (but appears continuously lit) with this demo.

5. BC Low Priority Asynchronous Message Insertion

Command 'L' inserts a low priority message into the scheduled BC message list. Low priority Inserted messages occur upon completion of any BC minor frame in-process when insertion is requested. First, enable the RT by pressing 'B' then Press 'A' to enable the BC transmission. Press 'L' to transmit three extra messages on Bus B. Bus B is used to make it easier to see on the scope and the Bus B LED should flash. If the RT is not enabled, retry messages appear on Bus B; this makes it difficult to see the three inserted messages. This will only work once after a power up or RESET.

The screen shot of these three messages are shown below captured by a Ballard USB UA1133 tester.

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 High Priority Asynchronous Message Insertion.

Follow the same steps as the previous BC low priority message example but this time Press ‘H’ to insert a single high priority message. This command is repeatable and the Bus B LED will flash with each command. Inserted High Priority messages occur upon completion of any in-process message when insertion is requested.

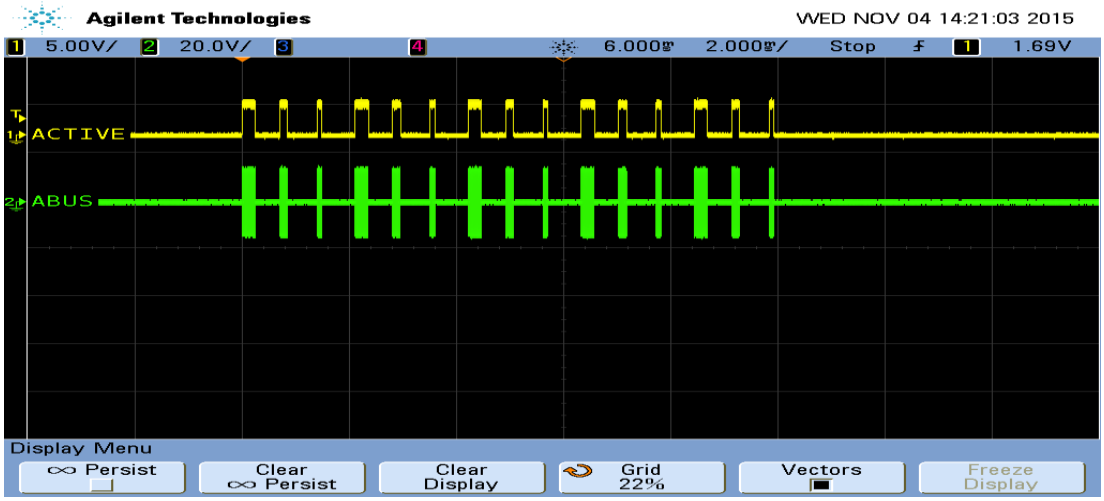
	T=000:00:0... dT=000:00:0...	Cwd1=0822 (01,R,01,02) <DATA WORDS> Swd1=0800	B		01: DEAD BEEF	1		

7. Command ‘E’ Enumerate Card is reserved for future use.

8. From a RESET, if the BC is started before enabling the ‘K’ and ‘T’ sequence to display message traffic, the first message may contain an error. This is normal; this occurs because the RT and MT are enabled midstream of a message in progress.

9. The ‘N’ command transmits fifteen commands to RT address 3. Press ‘B’ to enable the RT.

Press ‘N’ to execute the BC transmissions (15 messages are transmitted) which will appear on the bus as shown below. To optionally see the message traffic on the console, enable the RT message traffic by pressing ‘T’ if it hasn’t already been enabled.



Command ‘N’ (15 message) Traffic capture using a Ballard USB 1553 monitor.

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		

When a BC message is transmitted to a RT that is not enabled, “RT no response” (NORES) error is indicated.

```
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 using external BC

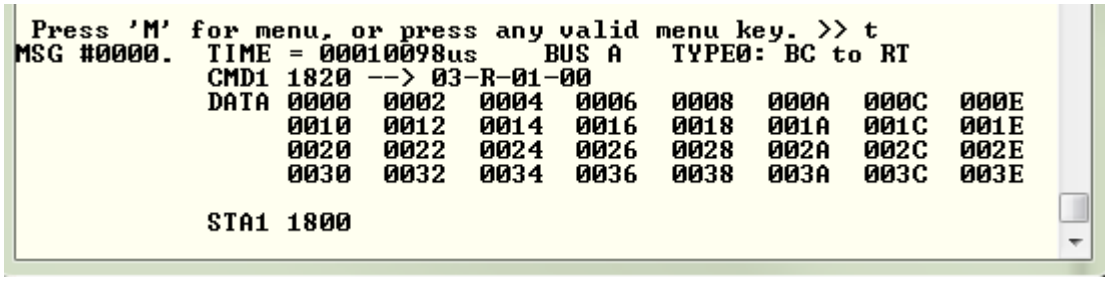
Using an external BC tester (such as Ballard USB 1553) to transmit messages to the demo board.

1. When an external BC is connected using conventional 1553 buses, use cables to connect the demo board circular tri-axial bus jacks to bus coupler ports on the A and B bus networks. In this case any external bus load 70 Ω resistors should be disconnected.

   If bus couplers are not readily available, bench testing can be done by connecting external dummy bus load 70 Ω resistors and connecting BC tester cables directly to the demo board tri-axial jacks for buses A and B.

2. Press the RESET button and then Press 'B' to enable the RT (RT1) then Press 'K' and 'T' to activate the RT traffic on the console. Compose a BC to RT message with SA=1 and 32 data words similar to the message shown below using external BC tester equipment.

3. The console should show the message transmitted by the BC, after the transaction.



4. When transmitting repeating messages at a high rate typical of MIL-STD 1553, the RT Traffic shown on the console may not keep pace due to limitations of the console 115,200 baud rate and the prolific use of slow-to-execute `printf` function calls in the C program's console user interface. Depending on the message content and repetition rate, some messages may not show on the console. All messages are transacted properly and captured by enabled RT and MT, some messages simply will not be shown on the console.

MT Mode

Press ‘S’ to enable SMT simple monitor. No other terminal is required; ‘S’ can be used after board reset.

A list of addresses shows the Command Stack and Data Stack buffer start address and end address. After sending some messages to the monitor use this command to display the addresses and use the Memory Watch window to view the Command and Data in memory. The SMT is also initialized with the ‘K’ command that provides message details. The ‘T’ RT Traffic feature toggles display of formatted RT message data on the console. The data shown below is the default values after a board reset.

```
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      = 0x3700
Command Buffer Next Address       = 0x3780
Command Buffer End Address        = 0x37ff
Command Buffer Interrupt Address  = 0x0000
Data Buffer Start Address         = 0x3800
Data Buffer Next Address          = 0x3900
Data Buffer End Address           = 0x39ff
Data Buffer Interrupt Address     = 0x0000


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

After one message transmitted by an external BC as shown will produce the following results.

```
MSG #0000.  TIME = 00093130us     BUS A   TYPE0: BC to RT
            CMD1 1830 --> 03-R-01-16
            DATA BBBB  0202  1414  0404  0505  0606  0707  0808
                 0909  1010  0000  0000  0000  0000  0000  0000

            STA1 1800

Command Buffer Start Address      = 0x3700
Command Buffer Next Address       = 0x3784
Command Buffer End Address        = 0x37ff
Command Buffer Interrupt Address  = 0x0000
Data Buffer Start Address         = 0x3800
Data Buffer Next Address          = 0x3911
Data Buffer End Address           = 0x39ff
Data Buffer Interrupt Address     = 0x0000
```

Getting Started with the Holt API demo software project and installing IAR Systems *Embedded Workbench for ARM* Compiler

1. Installed IAR Systems *Embedded Workbench for ARM* (*EWARM*) compiler is required BEFORE adding the Holt demo projects, so all Atmel board library files and the demo project folder are created in the proper location. Follow *the "Holt HI-6131 API Demo Project Installation Guide"* found in the Project folder on the Holt CD-ROM. Before proceeding to the next steps IAR must be installed and the two Holt project folders must be in the proper folder locations, according to that guide. **Instructions beyond this point assume you have completed the above installation tasks.**

2. Launch IAR *Embedded Workbench* from the Windows Start menu. A blank screen should appear. Open the Holt HI-6131 API Demo Project from the IAR File pull-down menu, click on File/Open/Workspace and navigate to the project folder location and select "HI-6131 API Demo.eww" and click the Open button.

3. An IAR Workspace window should appear on the left side as shown below. If the Workspace directory pane is missing, select "Workspace" from the View pull-down menu. Make any window adjustments or open any of the folder groups to view included files to suit your preferences.

4. Double click the `main.c` file, it should appear in the text editor pane, similar to the screen capture shown on the next page.

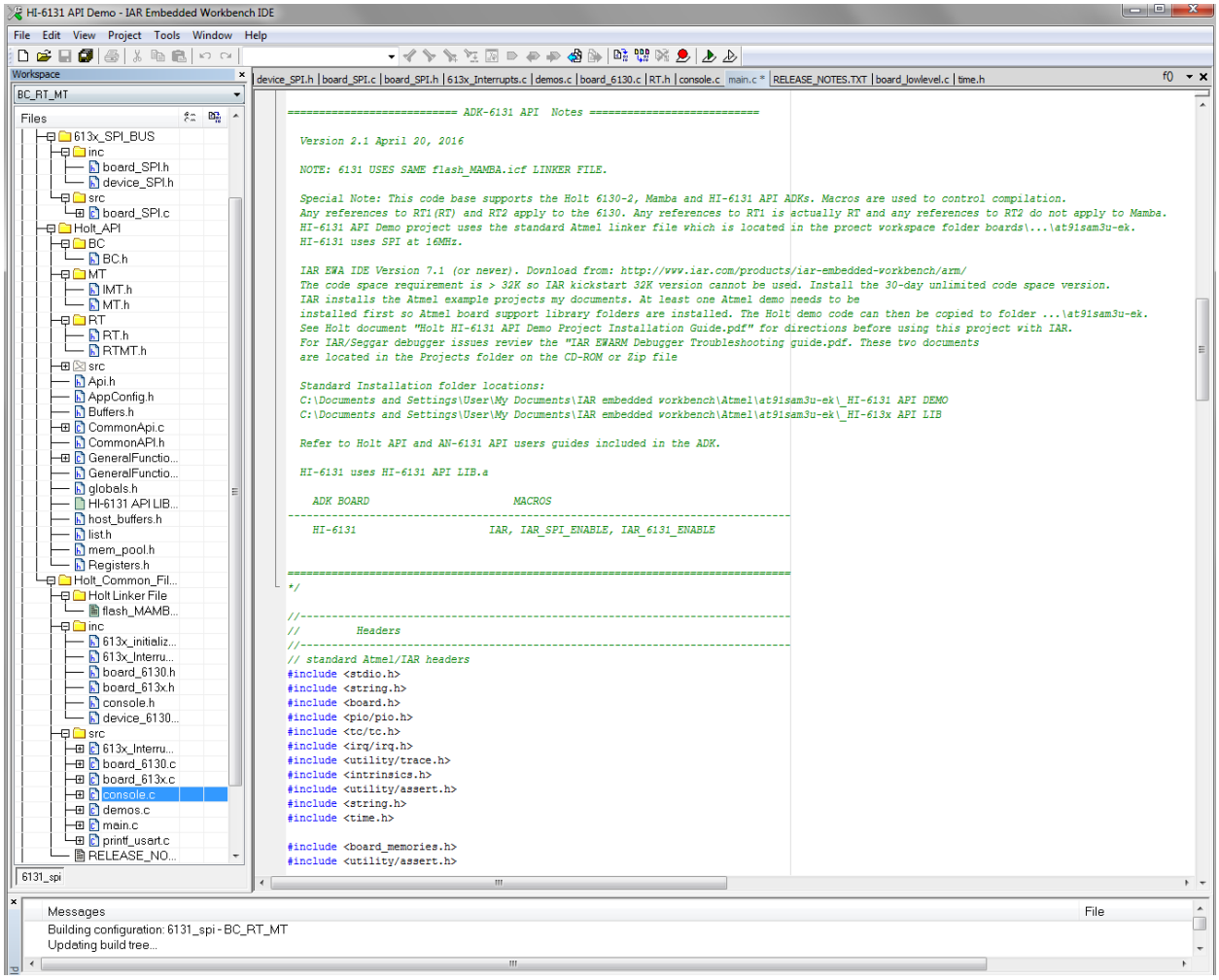
5. The first time a project is unzipped and installed in the appropriated folder a Rebuild All should be performed (from Project pull down menu).

6. IAR getting started, project management and other guides are available from the IAR Workbench Help pull down menu.

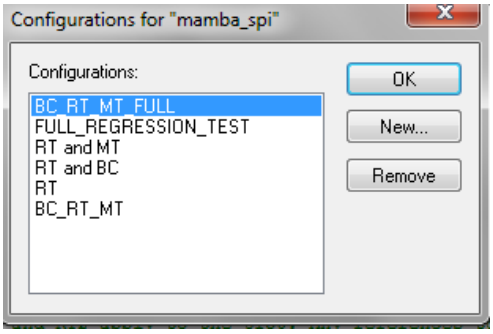
Mode Holt Project Configurations

IAR project configurations are used to reconfigure the demo software using preprocessor macros. Configurations are selectable from the Workspace pull-down menu using macros defined in the project options C/C++ preprocessor tab. The three macros **IAR**, **IAR_SPI_ENABLE** and **IAR_HI-6131_ENABLE** control which sections of code are compiled for the demo project. Other macros are used to enable BC, RT or SMT sections of code. Holt uses a common code base for multiple projects. This means there are functions not used in this project but are left in the C and H files. The IAR compiler uses the macros to select sections of code to build and link. Some references to HI-6130 or HI-6138 MAMBA™ are to be expected. In some cases some functions are simply not used or macros are used inside a function to select appropriate code.

Main Project demo screen (move and size the windows according to preference):



The configuration list may vary slightly from the list shown.

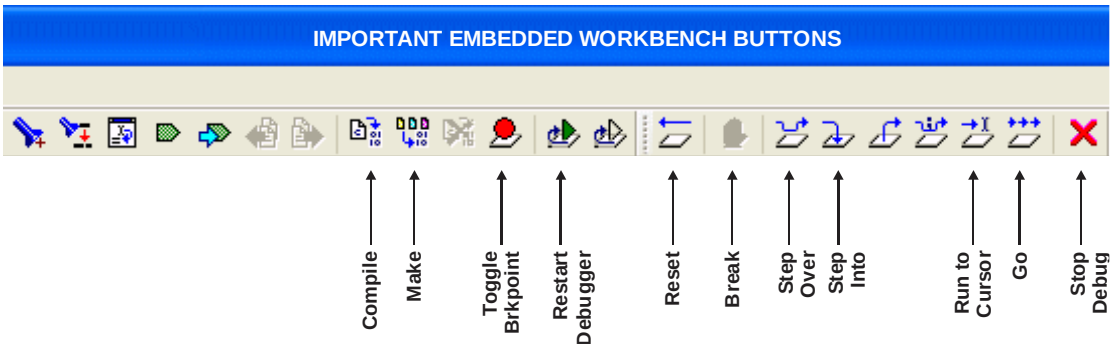


Project configurations with “FULL” required the Holt API library source code. The API library source code is not provided in the standard ADK. The full API source code is available with a signed Holt software license agreement (SLA). The standard ADK demos are fully functional otherwise, and API projects can be built without the API C source code. Contact Holt sales for the SLA for the optional API C source code.

The “FULL REGRESSION_TEST” is for Holt testing only and can be deleted by the user.

A simple way to create a new configuration is to select Project/Edit Configurations and then select New. The dialog box will allow a new configuration based on an existing configuration with a new configuration name. Select the new configuration and edit the preprocessor labels as desired then save the new configuration. The new configuration will now appear in the pull down menu. Project file **`613x_initialization.h`** configures other critical project settings, including the time tag resolution and console I/O on-off.

The IAR embedded workspace provides may of the commonly used tasks as short-cut buttons on the top as shown.



Project File List with Selected Descriptions

Most of the function names are self-explanatory, some functions retain the names from the original 6130-2 or MAMBA™ (HI-6138) project they were written for, don't worry about this they work fine with the HI-6131 API demo.

device_SPI.h

Macro definitions for register addressing.

board_SPI.c, board_SPI.h

SPI configuration, SPI read/write and Memory watch functions. SPI frequency selection.

613x_initialization.h

Definitions for important configuration settings.

613x_Interrupts.h

Provides interrupt related prototypes for API use.

main.c

The primary program entry portal, `main()` demonstrates initialization sequence used of enabled terminals. After initialization is complete, function calls demonstrate powerful addressing methods for all RAM structures used by the enabled terminal modes. Demo initialization and execution uses Holt high-level API in this demo kit.

board_613x.c, board_613x.h

```
Contains ARM MCU SPI i/o definitions and macro definitions for
SPI commands
SPIopcode(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) ;
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();      initializes ARM MCU general purpose I/O
reset_613x();
init_timer();
Delay_us(num_us);
Delay_ms(num_ms);
Delay_x100ms(num);
Flash_Red_LED();
Flash_Green_LED();
```

board_613x.h;

Contains ARM MCU i/o definitions controlled by macros. The SPI pins are common between MAMBA™ and HI-6131 since both use a SPI interface.

BC.h

BC.h has macros, C structures and prototypes for API use.

MT.h

MT.h has macros, C structures and prototypes for API use.

RT.h, RTMT.h

Has macros, C structures and prototypes for API use.

console.c

Console functions used by all terminal modes:

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

Console functions used by Bus Controller (BC) mode:

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

Console functions used by Remote Terminal RT:

```
RTDemo();
Rt_mt();
```

Console functions used by SMT or IMT bus monitor modes:

```
SMTDemo();
```

Primitive console functions that "printf" redundant char strings to reduce program size:

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

Console function called by the Memory_watch() function

Manual SPI write utilities

```
ascii2int();
```

demos.c

Provides all demo functions that are called by the console menu.

DisplayDecodedMsg(); used to display 1553 formatted data to the console when the 'T' command is used.

displayRTTraffic(); called from main; calls DisplayDecodedMsg()

board_lowlevel.c

This is an Atmel board file that changes the flash memory wait states to 3WS.

Flash_MAMBA.icf - Project Linker File SAME one used for MAMBA™ for this HI-6131 API demo project.

HI-613x API LIB.a – Precompiled Holt API library.

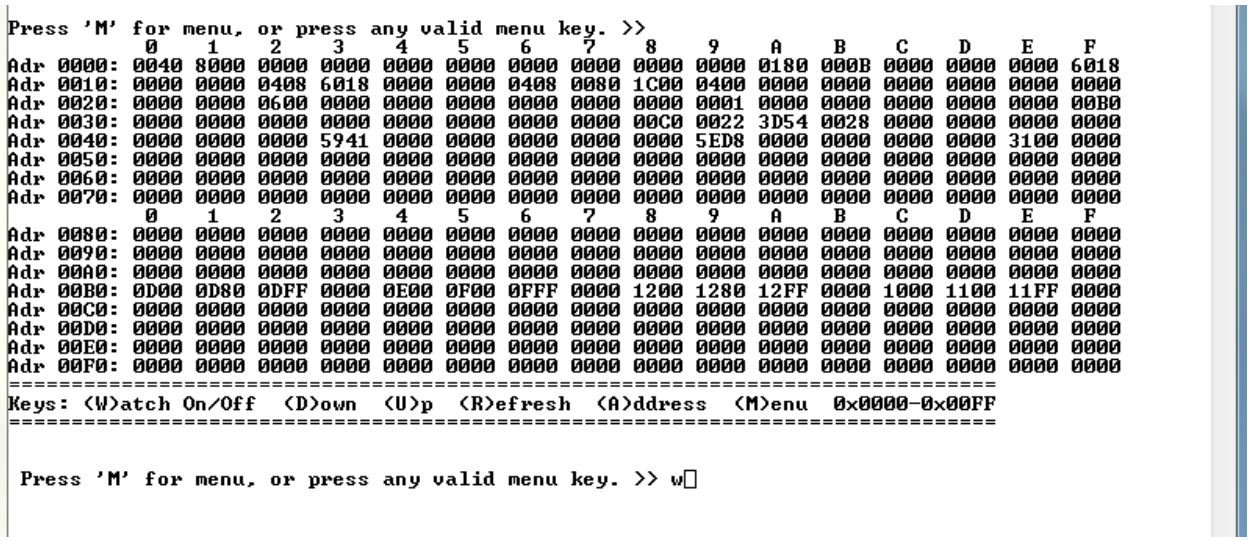
Holt API Library files

The standard API kit does not include the Holt API source files although the API header files are included and grouped in the workspace Holt_API/src sub-folder.

HI-6131 SPI Interface

The HI-6131 was designed for compatibility with microcontrollers having a Serial Peripheral Interface (SPI). RAM and register locations are written or read with the help of 8-bit SPI commands. Most read or write operations use one of four Memory Address Pointers (MAPs) to designate the address of the next location accessed. To speed up a multiword transfers, the enabled Memory Address Pointer automatically increments to the next address after each read or write is performed. Register addresses 0-15 decimal can be read directly, without using a memory address pointer. Register addresses 0-63 decimal can be written directly without using a memory address pointer.

When debugging, a memory watch utility may be helpful for observing register or RAM values. However SPI-accessed memory cannot be viewed from an IAR Embedded Workbench debug memory display window. The demonstration program provides memory watch capability via SPI, by using a C function called Memory_watch(). This function call only works when Console I/O is enabled. It displays 256 consecutive register or RAM values, starting with the provided memory address parameter. The entire memory address space 0 to 0x7FFF is accessible in 256 word increments. The demonstration program polls for keyboard input, and must be running. When the console menu “W” command is entered, the memory address space from 0x0000 to 0x00FF is displayed:



Be mindful that each displayed location is rescanned when Memory_watch( ) executes. Some register or RAM structure bits automatically reset after read occurs. This includes bits in the Pending Interrupt registers, and DBAC Data Block Accessed bits for RT Descriptor Table Control Words in RAM. For these, the Memory Watch window reflects the value in effect when the function executed.

The console I/O option using TeraTerm includes several menu options that read and display Pending Interrupt register status. Remember that Pending Interrupt bits automatically reset after read occurs. For these registers, the Memory Watch window reflects the value in effect when execution stopped.

The HI-6131ADK demonstration program is set up with Bus Controller, Remote Terminal 1, Remote Terminal 2 and Simple Monitor all enabled. Enabling or disabling any of these terminal functions is a two-step process: the software configuration (controlled C/C++ preprocessor macros) must match the hardware configuration DIP switches (BCENA, RT1ENA, RT2ENA and MTRUN) or a software error trap occurs. The preprocessor macros settings are set in the Project pull-down menu/Options/C-C++/Preprocessor tab.

The sub-menu at the bottom of the screen lists available Memory Watch options. Pressing ”D” (DOWN command) changes the displayed address range to 0x0100-0x01FF. Pressing ”U” (UP command) from the above screen wraps around the device address space, changing the displayed address range to 0x1F00-0x1FFF. Repeating UP or DOWN commands moves through the address range. Pressing “R” refreshes the currently selected address range, while pressing “A” (ADDRESS command) allows you to enter four hexadecimal characters to select any Memory Watch start address. Pressing “W” (WATCH) or “M” (MENU) toggles off Memory Watch display, restoring the menu shown on page 5.

For advanced users, other utilities available on the console (commands ‘1’, ‘2’ and ‘3) menu are also supplied for SPI writing to registers and memory addresses.

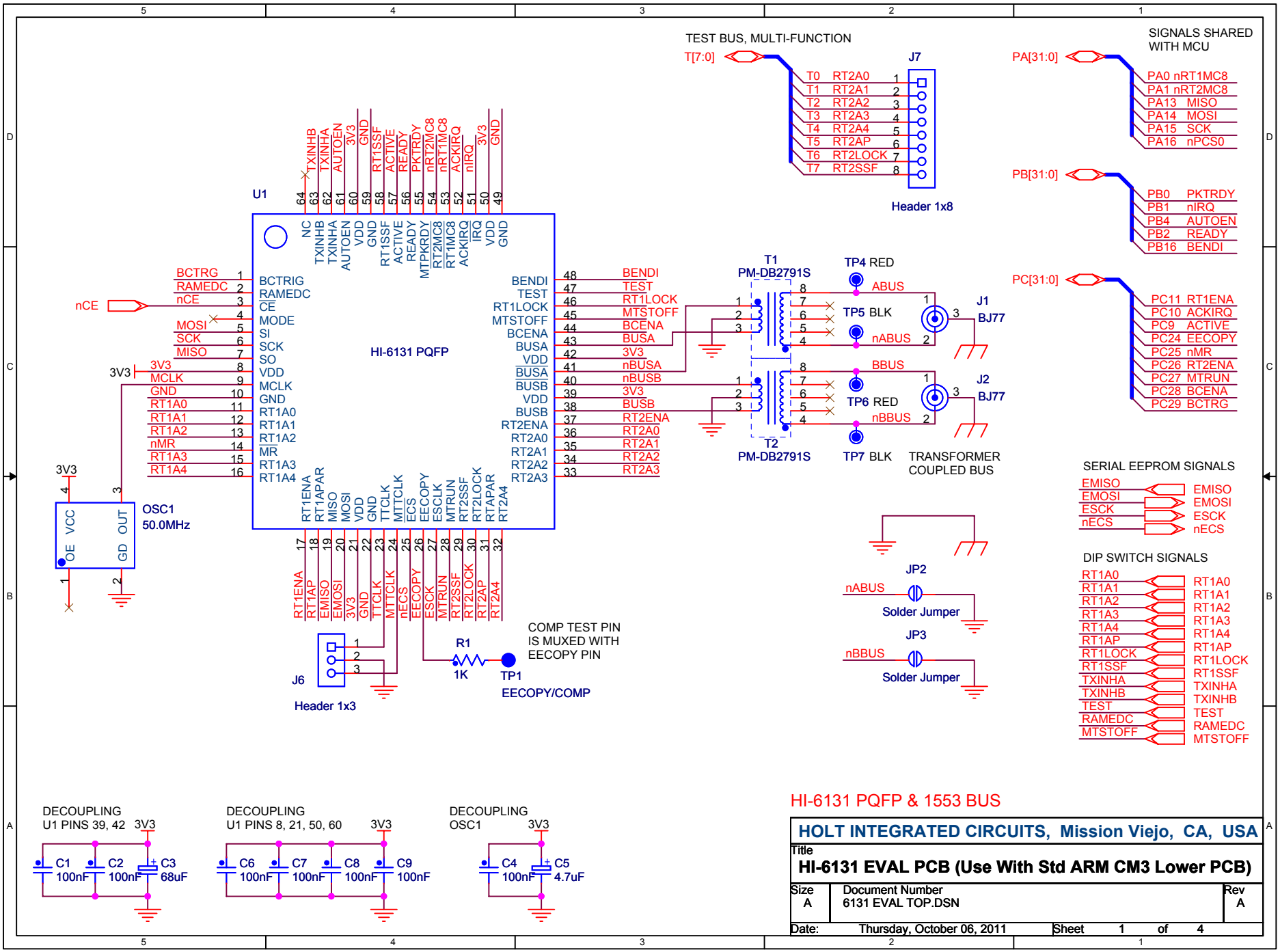
Summary

With just 4 host interface signals for accessing RAM or registers, the HI-6131 SPI interface simplifies MIL-STD-1553 terminal hardware design and saves considerable board space over a similar 16-bit parallel address and data bus interface.

The Holt 1553 API Runtime Library simplifies HI-6131 programming by providing high level C function calls that greatly accelerate project development.

Bill of Materials **HI-6131 Evaluation Board, Top** **Revised: July 19, 2011**

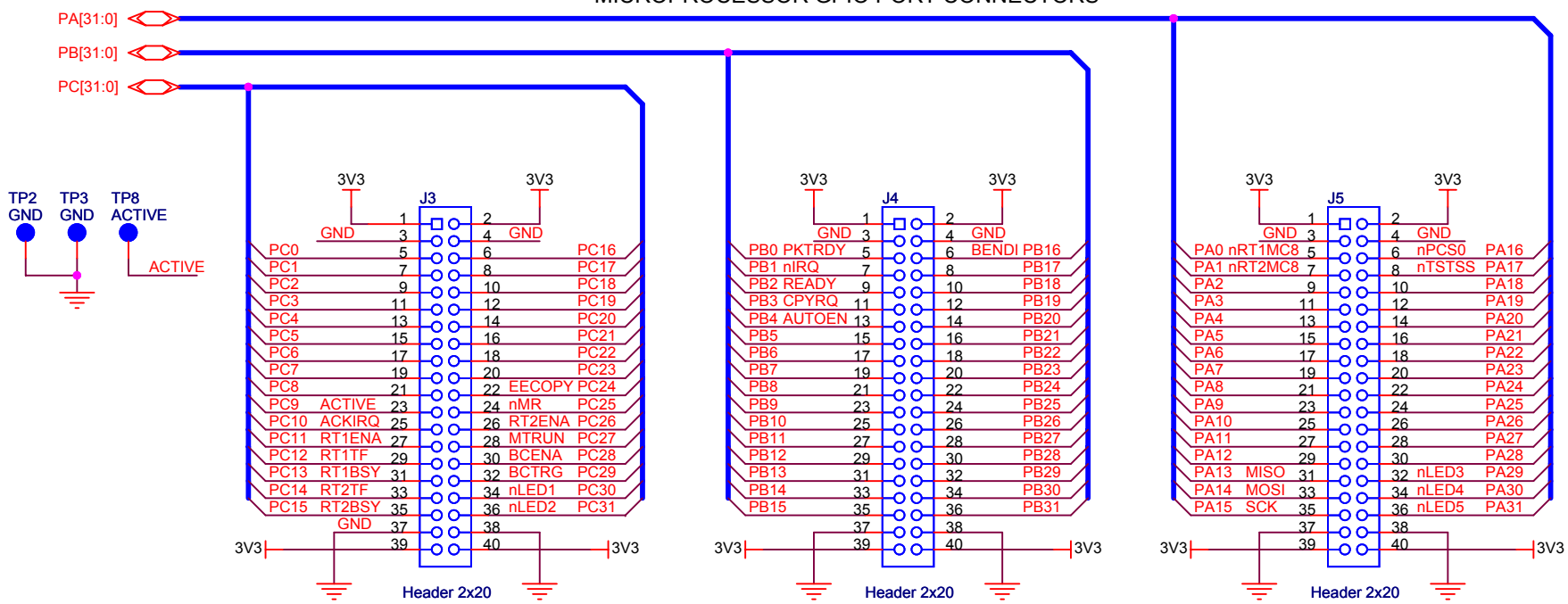
Item	*Qty*	*Description*	*Reference*	*DigiKey*	*Mfr P/N*
1	1	PCB, Bare, Eval Board	N/A	--------	--------
2	9	Capacitor, Ceramic 0.1uF 20% 50V Z5U 0805	C1,C2,C5, C6,C7,C8, C9,C10,C11	399-1176-1-ND	Kemet C0805C104M5UACTU
3	1	Capacitor, Ceramic 4.7uF 10% 6.3V X5R 0805	C5	399-3134-1-ND	Kemet C0805C475K9PACTU
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	Trompeter Electronics BJ77 Use 0.469" Round Hole
6	1	Header, Male 2x10 0.1" Pitch, 0.230" Pins, 0.120" Tails	J3	S2012E-10-ND	Sullins
7	1	Header, Male 2x7 0.1" Pitch, 0.230" Pins, 0.120" Tails	J4	S2012E-07-ND	Sullins
8	1	Header, Male 2x4 0.1" Pitch, 0.230" Pins, 0.120" Tails	J5A	S2012E-04-ND	Sullins
9	1	Header, Male 2x5 0.1" Pitch, 0.230" Pins, 0.120" Tails	J5B	S2012E-05-ND	Sullins
10	------	Header, 1x3, 0.1" pitch	J6	DO NOT STUFF	--------
11	------	Header, 1x8, 0.1" pitch	J7	DO NOT STUFF	--------
12	1	Header, 5x3, 0.1" pitch	JP1	Samtec	Samtec TSW-105-07-T-T
13	5	Jumper, shorting, w/ grip, 0.1"	JP1	S9341-ND	Sullins NPC02SXON-RC
14		Solder Jumper	JP2,JP3	DO NOT SOLDER	--------
15	1	LED Yellow 0805	LED5	160-1175-1-ND	Lite On LTST-C170YKT
16	3	LED Green 0805	LED1 - LED3	160-1179-1-ND	LiteOn LTST-C170GKT
17	1	LED Red 0805	LED4	160-1176-1-ND	LiteOn LTST-C170CKT
18	1	Osc, 50.00MHz 25ppm 3.3V SMD 5mm x 7mm	OSC1	CTX328LVCT-ND	CTX CB3LV-3I-64M0000-T
19	5	Resistor, 150 5% 1/8W 0805	R8,R9,R10, R11,R12	P150ACT-ND	Any
20	1	Resistor, 1.0K 5% 1/8W 0805	R1	P1.0KACT-ND	Any
21	1	Resistor, 2.2K 5% 1/8W 0805	R17	P2.2KACT-ND	Any
22	2	Resistor, 10K 5% 1/8W 0805	R30,R31	P10KACT-ND	Any
23	6	Resistor, 47K 5% 1/8W 0805	R13,R14,R15 R16,R18,R19	P47KACT-ND	Any
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	Premier Magnetics PM-DB2791S
29	------	Test Point, pad w/ plated hole	TP1,TP9	------	--------
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-6131 Holt 64-PQFP	U1	------	--------
34	2	IC, Serial EEPROM 512Kbit 20MHz SPI 8-SOIC, Microchip	U2, U3	25LC512T-I/SNCT-ND	Microchip 25LC512T-I/SN

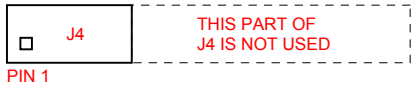


HI-6131 PQFP & 1553 BUS

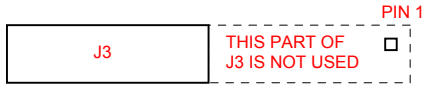
HOLT INTEGRATED CIRCUITS, Mission Viejo, CA, USA		
Title: HI-6131 EVAL PCB (Use With Std ARM CM3 Lower PCB)		
Size: A	Document Number: 6131 EVAL TOP.DSN	Rev: A
Date: Thursday, October 06, 2011	Sheet 1 of 4	

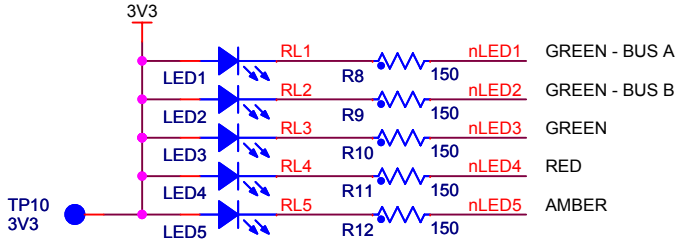
MICROPROCESSOR GPIO PORT CONNECTORS





HEADER ORIENTATION
ON THE CIRCUIT BOARD

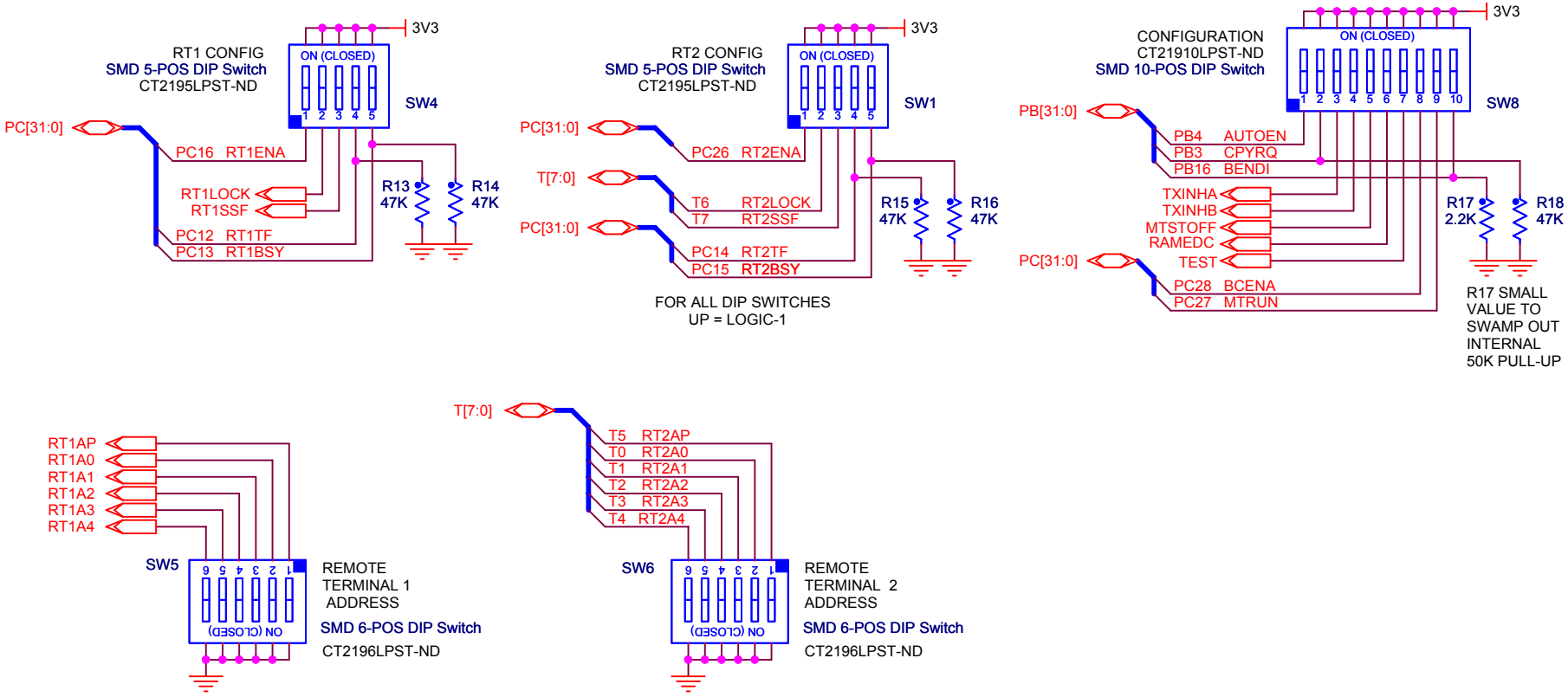




STATUS LEDS

MCU I/O HEADERS & LEDS

HOLT INTEGRATED CIRCUITS, Mission Viejo, CA, USA		
Title: HI-6131 EVAL PCB (Use With Std ARM CM3 Lower PCB)		
Size: A	Document Number: 6131 EVAL TOP.DSN	Rev: A
Date: Thursday, October 06, 2011	Sheet 2 of 4	



ALL DIP SWITCH SIGNALS WITH A PULL-UP OR A PULL-DOWN GO TO THE MCU.

DIP SWITCHES & SERIAL EEPROM

HOLT INTEGRATED CIRCUITS, Mission Viejo, CA, USA

Title: **HI-6131 EVAL PCB (Use With Std ARM CM3 Lower PCB)**

Size	Document Number	Rev
A	6131 EVAL TOP.DSN	A

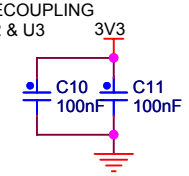
Date: Thursday, October 06, 2011 Sheet 3 of 4

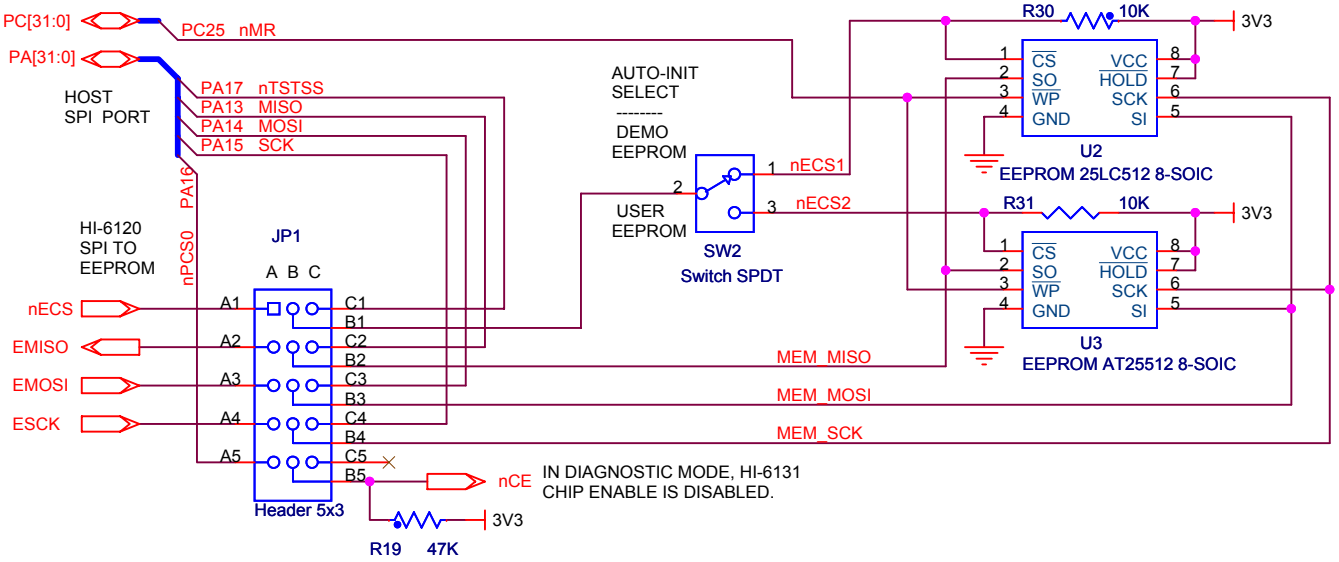
DUAL EEPROM CIRCUIT FOR EVALUATION BOARD ONLY. SW2 SELECTS EEPROM.
JP1 JUMPER ALSO PROVIDES MCU READ/WRITE ACCESS TO SELECTED EEPROM.

JUMPERS NORMALLY SPAN COLUMNS A-B FOR HI-613X CONTROL OF SERIAL EEPROM.

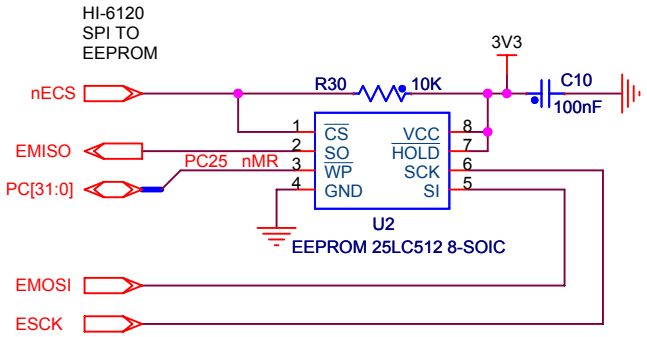
FOR DIAGNOSTIC TESTS, JUMPERS SPAN COLUMNS B-C SO THE MCU SPI CAN READ/WRITE THE SERIAL EEPROM. THE C PROGRAM CONTROLS TEST SLAVE SELECT SIGNAL, nTSTSS.

DECOUPLING U2 & U3





TYPICAL APPLICATION REPLACES ABOVE EEPROM CIRCUIT WITH THIS SIMPLE CONFIGURATION



HI-6131 PQFP & 1553 BUS

HOLT INTEGRATED CIRCUITS, Mission Viejo, CA, USA

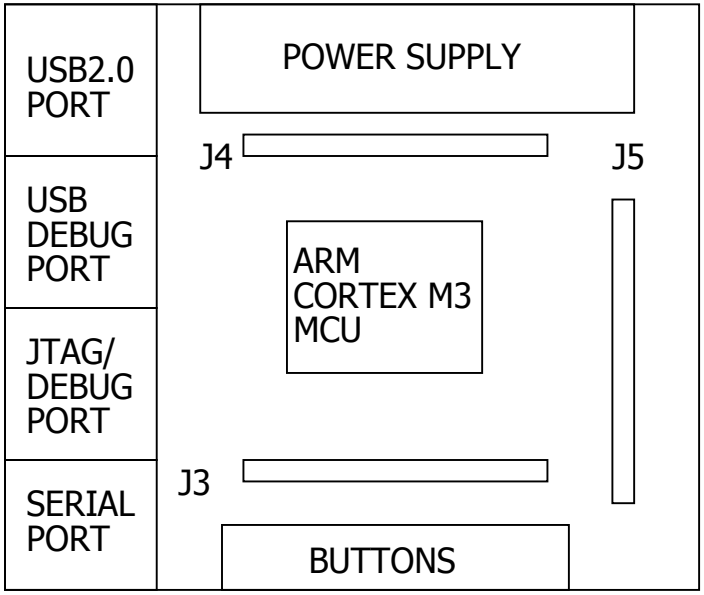
Title: **HI-6131 EVAL PCB (Use With Std ARM CM3 Lower PCB)**

Size	Document Number	Rev
A	6131 EVAL TOP.DSN	A

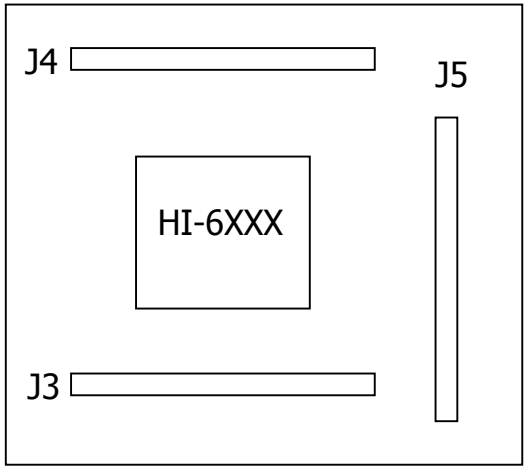
Date: Thursday, October 06, 2011 Sheet 4 of 4

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% NP0 C0G 0V 0603	C23,C34	490-1403-1-ND	Murata GRM1885C1H100JA01D
6	4	Capacitor, Ceramic 20pF 5% NP0 C0G 0V 0603	C14,C21,C25, C27	490-1410-1-ND	Murata GRM1885C1H200JA01D
7	29	Capacitor, Ceramic 100nF 10% 25V Y5V 0603	C2,C4,C6-C11, C13,C15-C19,C22,C24,C26,C28,C29,C33, C35-C40,C45-46,C54	490-1575-1-ND	Murata GRM188F51E104ZA01D
8	4	Capacitor, Tantalum 4.7uF 10% 10V Low ESR SMD 1206	C5,C20,C31, C32	478-2391-1-ND	AVX TPSA475K010R1400
9	4	Capacitor, Tantalum 10uF 10% 10V Low ESR SMD 1206	C3,C12,C30,C41	478-3317-1-ND	AVX TPSA106K010R1800
10	1	Capacitor 22uF 10% 6.3V Tantalum Low ESR SMD C	C43	399-10521-1-ND	Kemet T495C226K006ATE380
11	1	Capacitor 100uF 10% 6.3V Tantalum Low ESR SMD C	C44	495-1509-1-ND	Kemet T495C107K006ZTE150
12	1	Header, Male Shrouded 2x10, 0.1" Pitch	J1	HRP20H-ND	Assmann AWHW20G-0202-T
13	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, Bumpon 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



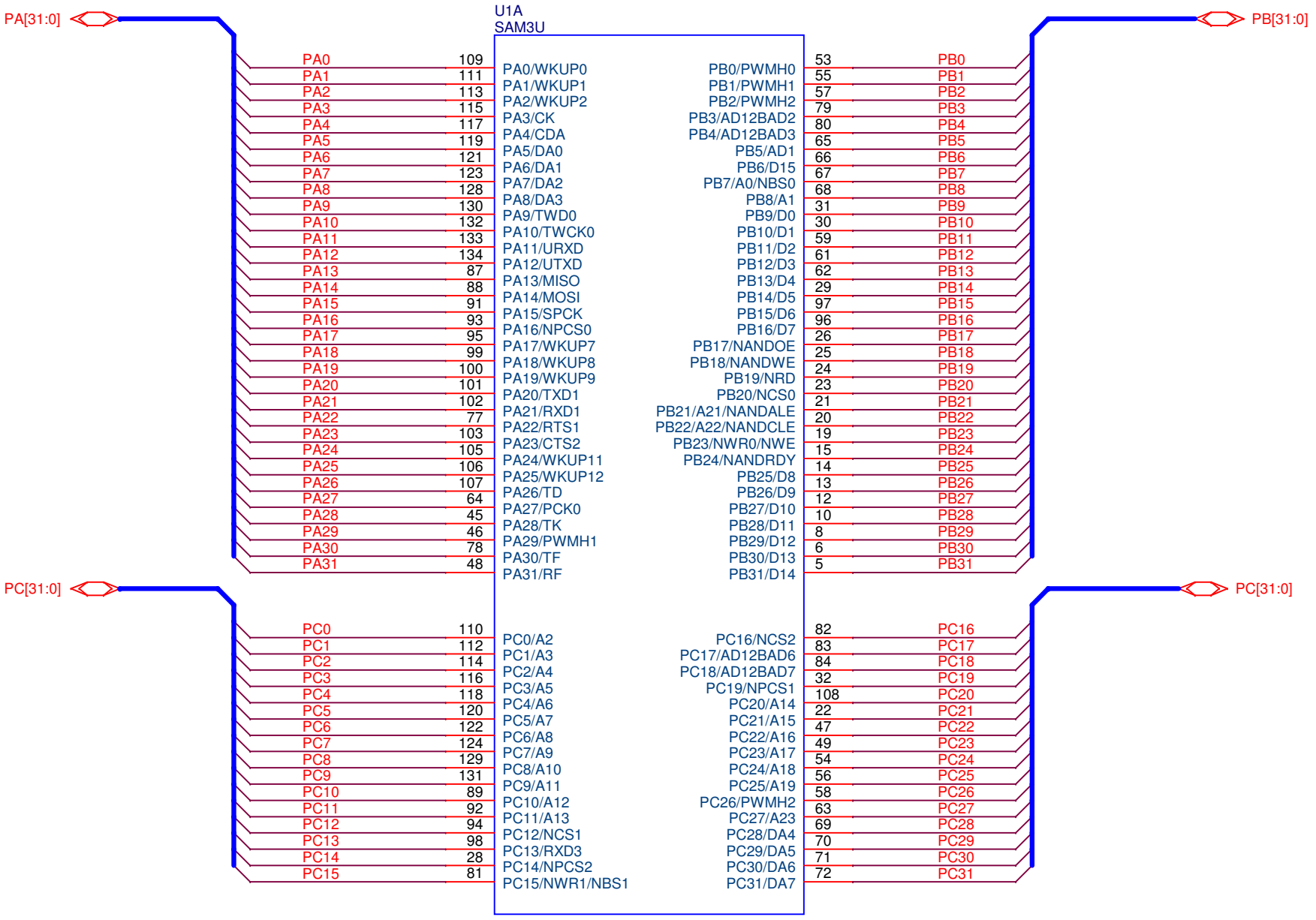
LOWER CIRCUIT BOARD



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: A	Document Number: CM3 BOARD REV E.DSN	Rev: E
Date: Wednesday, June 01, 2016	Sheet 1 of 7	



ARM CORTEX M3 PIO

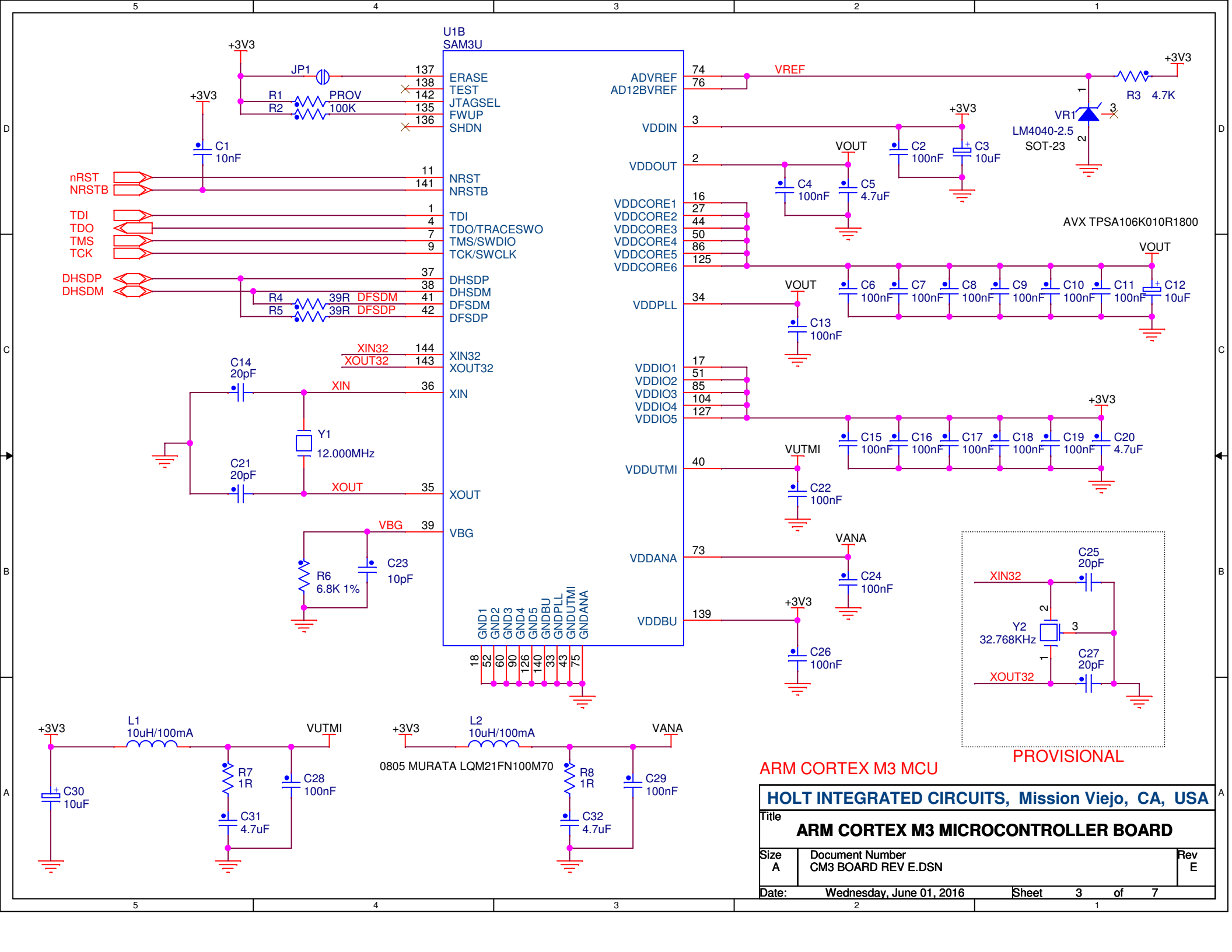
HOLT INTEGRATED CIRCUITS, Mission Viejo, CA, USA

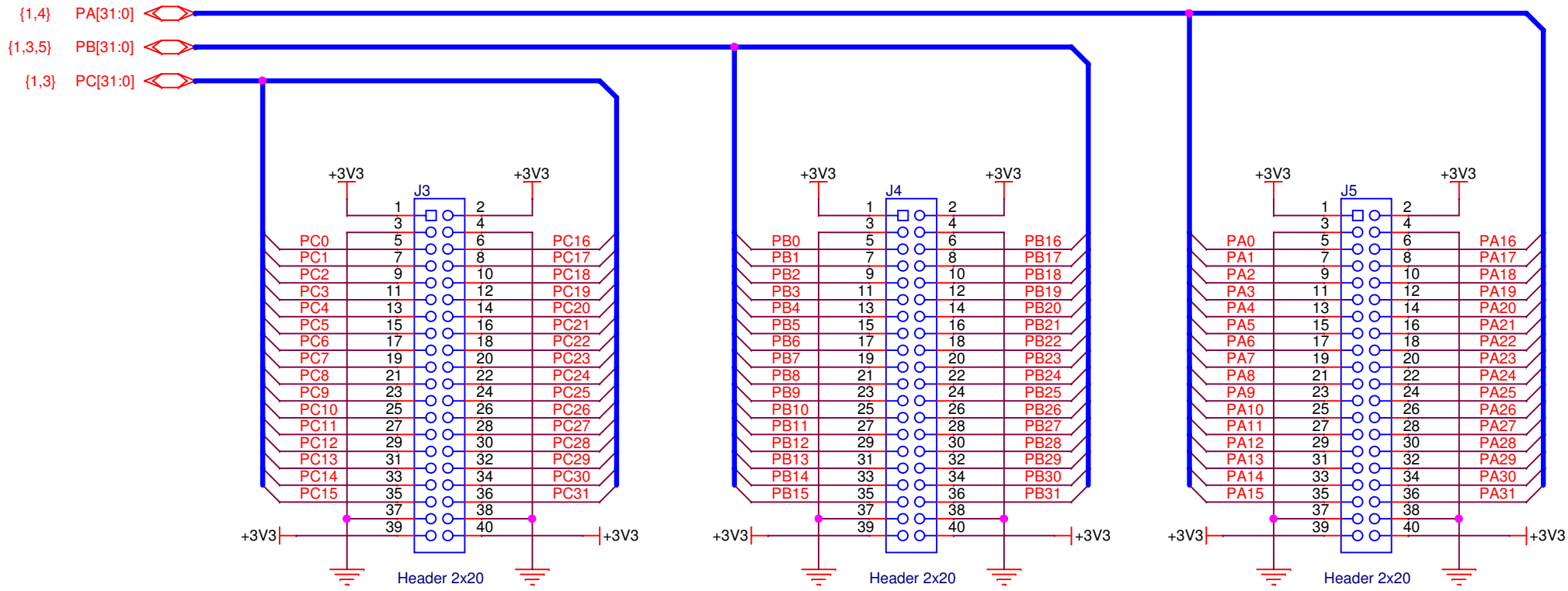
Title

ARM CORTEX M3 MICROCONTROLLER BOARD

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

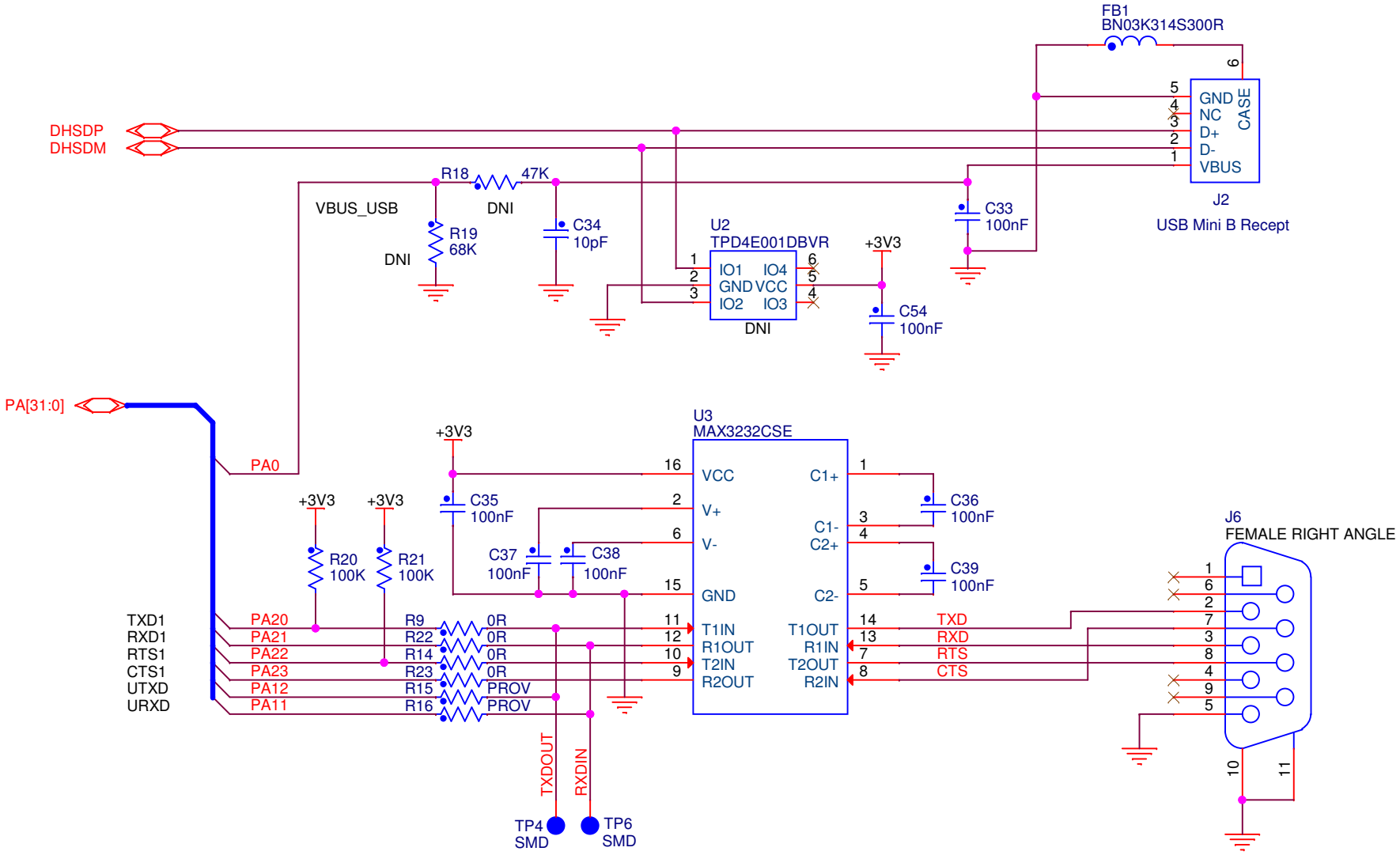
Date: Wednesday, June 01, 2016 Sheet 2 of 7


U1B
SAM3U
+3V3
JP1
R1
R2
PROV
100K
C1
10nF
nRST
NRSTB
TDI
TDO
TMS
TCK
DHSDP
DHSDM
R4
R5
39R
DFSDM
DFSDP
XIN32
XOUT32
C14
20pF
XIN
Y1
12.000MHz
C21
20pF
XOUT
VBG
R6
6.8K 1%
C23
10pF
ERASE
TEST
JTAGSEL
FWUP
SHDN
NRST
NRSTB
TDI
TDO/TRACESWO
TMS/SWDIO
TCK/SWCLK
DHSDP
DHSDM
DFSDM
DFSDP
XIN32
XOUT32
XIN
XOUT
VBG
GND1
GND2
GND3
GND4
GND5
GNDBU
GNDPLL
GNDUTMI
GNDANA
ADVREF
AD12BVREF
VDDIN
VDDOUT
VDDCORE1
VDDCORE2
VDDCORE3
VDDCORE4
VDDCORE5
VDDCORE6
VDDPLL
VDDIO1
VDDIO2
VDDIO3
VDDIO4
VDDIO5
VDDUTMI
VDDANA
VDDBU
VREF
R3 4.7K
VR1
LM4040-2.5
SOT-23
VOUT
C2
100nF
C3
10uF
C4
100nF
C5
4.7uF
AVX TPSA106K010R1800
C6
100nF
C7
100nF
C8
100nF
C9
100nF
C10
100nF
C11
100nF
C12
10uF
C13
100nF
C15
100nF
C16
100nF
C17
100nF
C18
100nF
C19
100nF
C20
4.7uF
VUTMI
C22
100nF
VANA
C24
100nF
C26
100nF
C25
20pF
Y2
32.768KHz
C27
20pF
L1
10uH/100mA
C30
10uF
R7
1R
C28
100nF
C31
4.7uF
L2
10uH/100mA
0805 MURATA LQM21FN100M70
R8
1R
C29
100nF
C32
4.7uF
PROVISIONAL
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



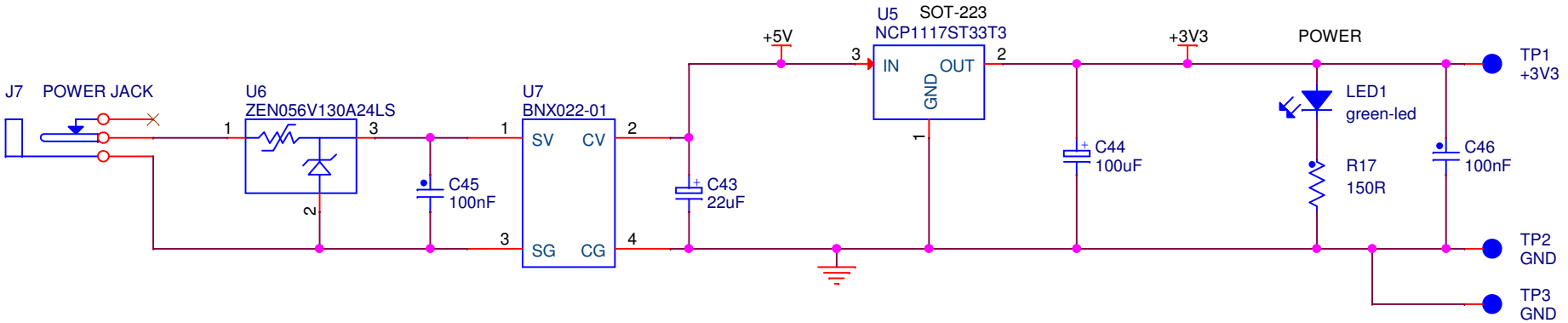
BOARD I/O HEADERS, BUTTONS

HOLT INTEGRATED CIRCUITS, Mission Viejo, CA, USA		
Title: ARM CORTEX M3 MICROCONTROLLER BOARD		
Size: A	Document Number: CM3 BOARD REV E.DSN	Rev: 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: A	Document Number: CM3 BOARD REV E.DSN	Rev: E
Date: Wednesday, June 01, 2016	Sheet 5 of 7	



POWER SUPPLY

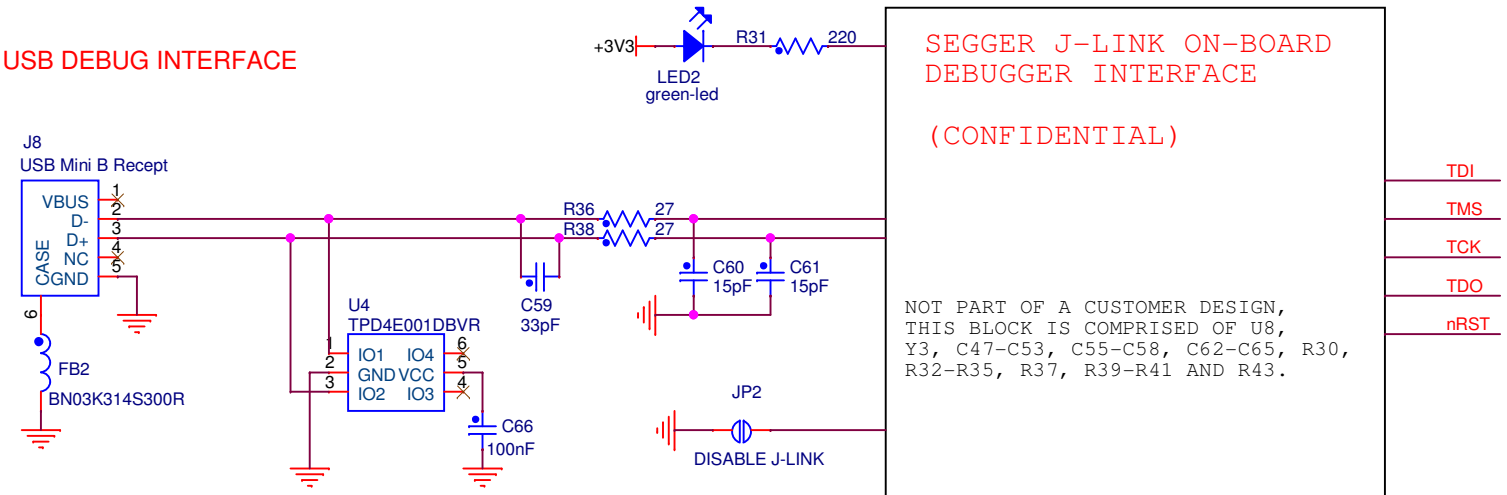
HOLT INTEGRATED CIRCUITS, Mission Viejo, CA, USA

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

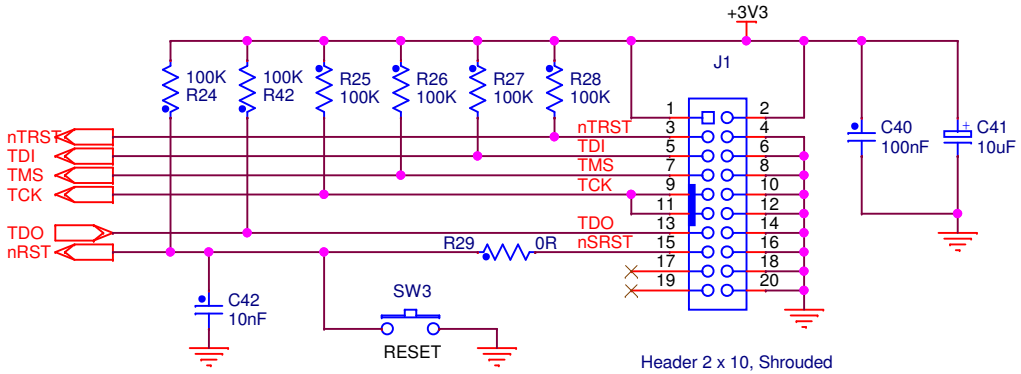
USB DEBUG INTERFACE



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