HOLT INC.
INTEGRATED CIRCUITS

MAMBA^{TM} Family
MIL-STD-1553 API
Application Development Kit

Devices Supported

HI-6135, HI-6136

HI-6137, HI-6138

June 2016

REVISION HISTORY

Revision	Date	Description of Change
AN-6138API, Rev. New	04-01-16	Initial release
Rev. A	06-07-16	Update board schematics and BOMs. Update board photo.

Introduction

The Holt MAMBA^{TM} evaluation board demonstrates the broad feature set of Holt's high level 1553 API library when used with the MAMBA^{TM} family of MIL-STD-1553 protocol devices…

HI-6135 Remote Terminal

HI-6136 Remote Terminal and/or Bus Monitor

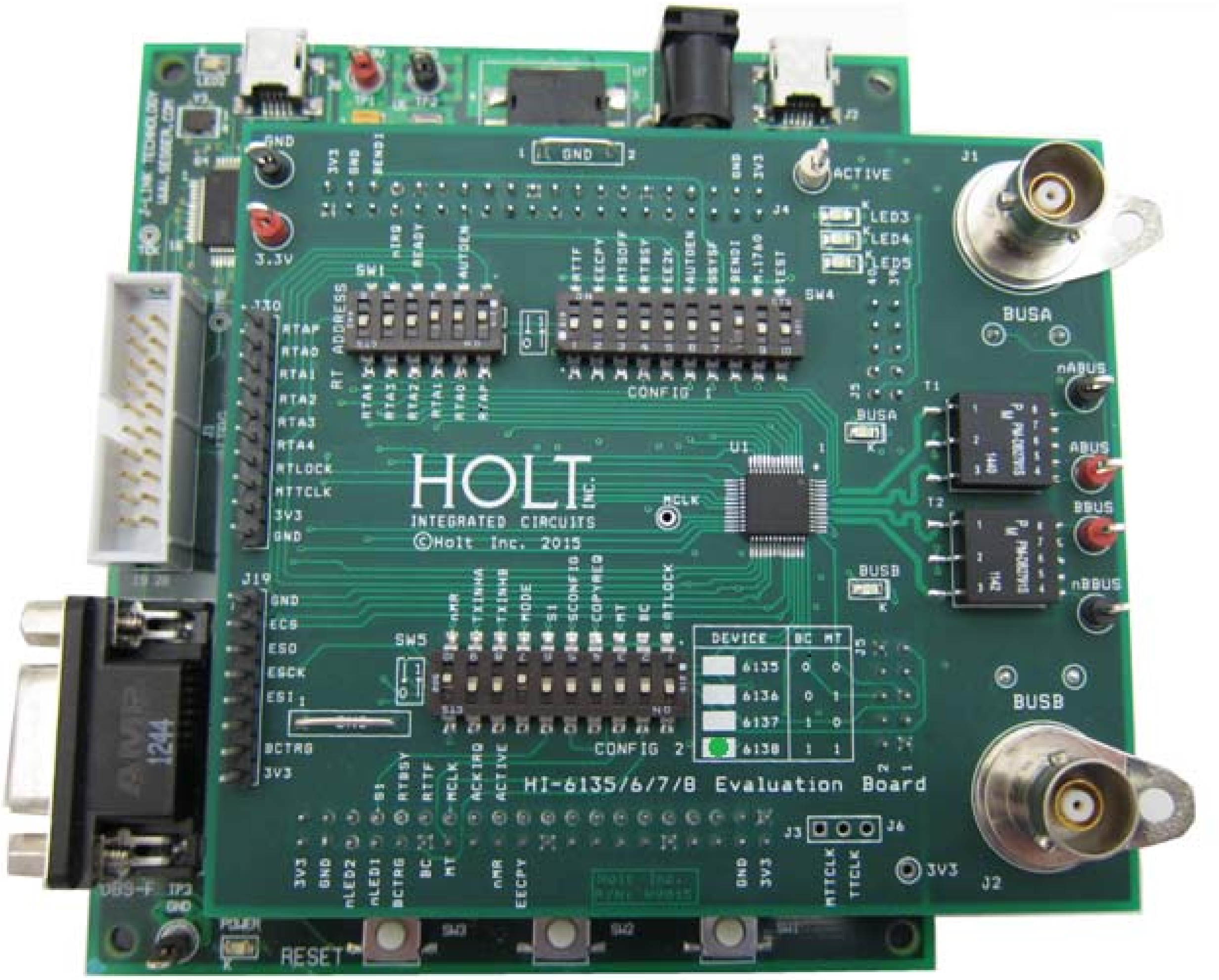
HI-6137 Remote Terminal and/or Bus Controller

HI-6138 Remote Terminal, Bus Controller and/or Bus Monitor (ADK uses a HI-6138)

The MAMBA^{TM} family of MIL-STD-1553B bus communication devices contains highly effective protocol logic and physical bus interface circuitry. The 2-board assembly and C project reference design provides a ready-to-run evaluation platform demonstrating concurrent operation for any combination of Bus Controller, Bus Monitor and Remote Terminal. For convenience, this kit includes IAR Systems *Embedded Workbench® for ARM*, and a fully integrated debug interface for the ARM Cortex M3 microcontroller. The reference device for this MAMBA^{TM} guide is the HI-6138 because this IC contains all available features; the other MAMBA^{TM} devices contain a subset of HI-6138 features.

This API version of the MAMBA™ ADK demonstrates the Holt API library. For information on non-API embedded control (demo kits ADK-6135/6136/6137/6138) refer to document AN-6138.pdf.

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.

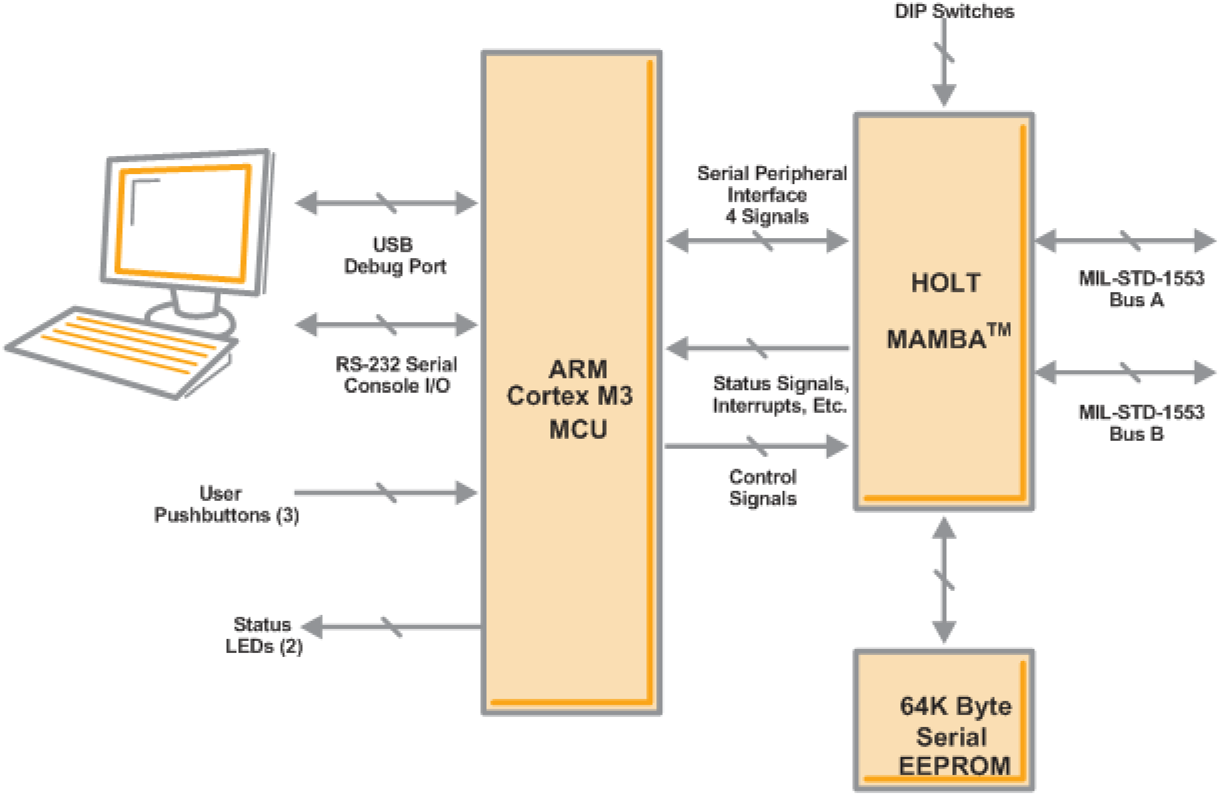


MAMBA^{TM} Evaluation Board, mounted on the ARM Cortex M3 MCU Board

Evaluation Kit Contents

- This User Guide.
- Holt MAMBA^{TM} Software Projects and Documentation CD.
    - HI-613x API LIB runtime library
    - HI-6138 API Demo example project
- Installation CD for IAR Systems *Embedded Workbench® for ARM*, version 7.1 or greater. See installation guide “Holt HI-6138 API project installation guide” for important instructions.
- Plug-in 5V DC power supply.
- 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 MAMBA^{TM} 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



Default Switch Settings

RT Address

Switch	POSITION	DESCRIPTION
SW1, 6-2	00011 (OFF = 1)	RTA4:0, sets the RT address, default is set for RT3
SW1, 1	OFF = 1	RTAP, RT address parity bit, must reflect correct odd parity or RT operation cannot be started. Default (for RT3) is logic-1

Config 1

SWITCH	DEFAULT	DESCRIPTION
SW4, 1	OFF (0)	RTTF – ON sets Terminal Flag bit in RT Status Word
SW4, 2	OFF (0)	EECPY – ON, makes copy of RAM and regs to EPROM **Note 1**
SW4, 3	OFF (0)	MTSTOFF – ON, disables memory test on power up
SW4, 4	OFF (0)	RTBSY – ON, MCU sets Busy bit in RT Status Word
SW4, 5	OFF (0)	EE2K – ON: EECPY and AUTOEN use just 2K words of EEPROM for faster boot process **Note 1**
SW4, 6	OFF (0)	AUTOEN – ON enables register/RAM initialization from external EEPROM, else MCU initializes device **Note 1**
SW4, 7	OFF (0)	SSYSF – ON sets Subsystem Fail bit in RT Status Word
SW4, 8	ON (1)	BENDI – ON sets big endian memory & register access
SW4, 9	OFF (0)	MODE1760 – ON sets Busy bit in RT Status Word immediately at hardware Master Reset rising edge
SW4, 10	OFF (0)	TEST – ON enables self-test. Demonstration requires additional software. See data sheet section 21

Config 2

SWITCH	DEFAULT	DESCRIPTION
SW5, 1	ON (1)	RTLOCK– OFF (logic-1) locks RT address, so latched external RT address from switch SW1 cannot be overwritten by register write
SW5, 2	OFF (1) (6138, 6137) ON (0) (6136,6135)	BC – OFF. This switch is not used by this API version of demo program. Can be repurposed by user.
SW5, 3	OFF (1) (6138, 6136) ON (0) (6137,6135)	MT– OFF. This switch is not used by this API version of demo program. Can be repurposed by user.
SW5, 4	ON (0)	COPYREQ – OFF (logic-1) MCU program writes RAM and register configuration to EEPROM after boot-up **Note 1**
SW5, 5	ON (0)	SCONFIG –Not Used
SW5, 6	ON (0)	S1 – Not Used
SW5, 7	OFF (1)	MODE – Device requires mode pin high
SW5, 8	ON (0)	TXINHB – OFF (logic-1) disables 1553 BUSB driver
SW5, 9	ON (0)	TXINHA – OFF (logic-1) disables 1553 BUSA driver
SW5, 10	OFF (1)	nMR – ON (logic-0) holds device in reset

Note 1: Auto-initialization of registers and RAM from EEPROM is not used in this demo software because the API runtime library performs initialization. See Holt application note AN-6138 for information and example code for this optional feature.

1760 Mode (all devices)

In this mode, the RT device responds with the Status Word’s Busy bit set within 2ms of Master Reset pin rising edge. To test this feature, the device can be powered up without the software running (for example by using DIP switch to hold the MCU in reset). If the nMR switch is toggled on the ADK (SW5/10) the device can quickly respond to a BC command with the ‘Busy’ bit set.

Default Jumper Settings

JUMPER	POSITION	DESCRIPTION
JP2	OPEN	Link to connect the negative line of BUSA to the board ground. Note 2
JP3	OPEN	Link to connect the negative line of BUSB to the board ground. Note 2
JP8	OPEN	Link to connect 70Ω load resistor on BUSA. Note 3
JP9	OPEN	Link to connect 70Ω load resistor on BUSB. Note 3
JP10	CLOSED	Links 50 MHz clock to lower MCU board

Note 2: Connecting Bus Negative to ground is strictly a bench test convenience feature. Most performance characteristics of transmitted and received 1553 signals are specified using differential line-to-line measurements at the bus stub, Bus Positive minus Bus Negative. This corresponds to the red and black “BUS” test points adjacent to the transformers on the right side of the upper circuit board. While two oscilloscope probes connected to red and black may be used in conjunction with scope’s Ch1-Ch2 math function, a single probe connected to Bus Positive provides the same signal display when Bus Negative is grounded. This frees up scope probes for other purposes.

Do not include a provision for grounding Bus Negative in your production design.

Note 3: For stand-alone testing (without connection to a conventional MIL-STD 1553 bus) the hardware provides on-board 70Ω termination resistors on the back side of the daughter card, enabled by JP8 and JP9. This is strictly a bench test convenience feature that supports demonstration of concurrent BC and RT without external 1553 bus connections. The HI-6138 contains independent state machines for concurrent BC, RT and monitor terminal modes. For demonstration purposes the integrated BC and RT can fully transact messages, with or without operational bus monitor. The ADK is configured to do so.

On-board termination resistors are not used when connecting to a properly terminated MIL-STD-1553 bus. Do not include a provision for termination resistors in your production design.

Hardware Design Overview

Refer to the end of this guide for separate schematic diagrams and bills of material for the upper device TARGET board and lower MCU board.

The detachable TARGET 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 connectors and boards. All host interface signals go through the inter-board headers. Numerous MAMBA™ device configuration pins (such as Remote Terminal address setting inputs) are controlled by DIP switches on the upper TARGET board; these signals are not available on the inter-board headers.

The lower ARM Cortex M3 MCU board is based on the flash-programmable Atmel AT91SAM3U-EK microprocessor. The Atmel processor 4-signal Serial Peripheral Interface (SPI Master 24MHz) using NPCS0 configured as a GPIO pin connects to MAMBA™ (Slave). A UART-based serial port provides RS-232 console I/O (optional). An uncommitted USB 2.0 port is available for future or customer expansion. Two pushbuttons are available for software interaction. A RESET pushbutton resets the ARM microprocessor, which in turn controls the TARGET 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").

Holt API Host Memory Considerations

Holt APIs use 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 demos 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-6130-2A.pdf for this example. Note: That ADK uses a Holt HI-6130 protocol IC with parallel bus interface (not SPI) and the 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 Holt MAMBA^{TM} Application Development Kit demonstrates all four devices in the MAMBA^{TM} family. The HI-6138 is used because it operates in all three terminal modes; Remote Terminal (RT), Bus Controller (BC), SMT Bus Monitor (MT). If using a different MAMBA^{TM} variant, it is a simple matter to disable one or two unused terminal modes.

1. Your 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.

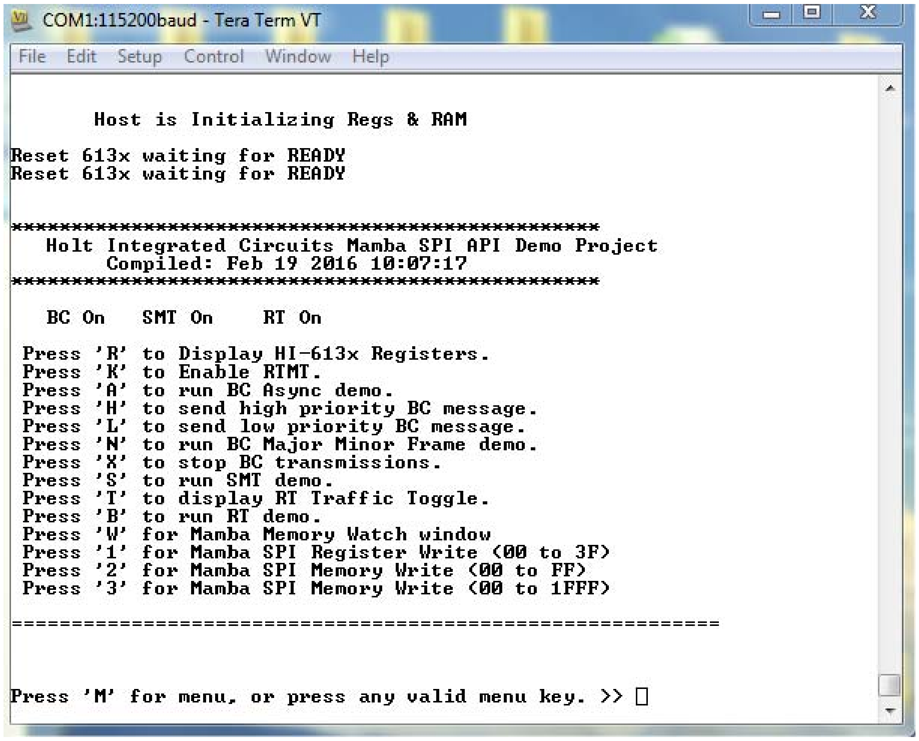
   If using Vista or Windows 7…
   HyperTerminal is not included with these versions of Windows. Install the free open-source terminal emulation program, *TeraTerm 4.71*, by running the provided teraterm-4.71.exe installer program from the Holt CD. Accept the license agreement stating redistribution is 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.

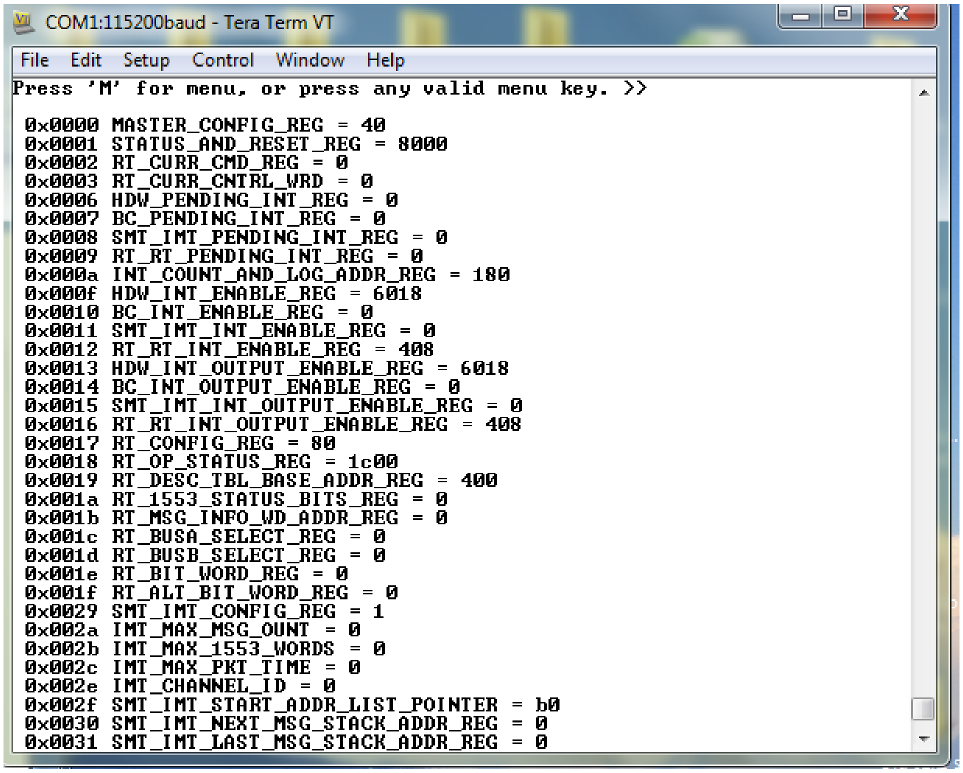
3. Plug-in the provided 5V DC power supply and connect the cable to the power input jack on the lower circuit board. If TeraTerm is running and configured correctly, the command menu below should appear in the console window. This menu appears whenever board power is applied, or after the RESET pushbutton 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.


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. >> 
```

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


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
```

The RT terminal address is set using DIP switches, before applying power. RT addresses 3 and 1 are utilized by the preprogrammed Bus Controller message repertoire. The 6-position DIP switch should already be set with the address values 03, plus odd parity.

If not connected by cable to conventional MIL-STD-1553 buses, a dummy 70Ω load for the buses is provided on the board by connecting solder jumpers JP8 and JP9.

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 `MAMBA 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 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 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. Without an external buses connected, close jumpers JP8 and JP9 (as described on page 6) to connect on-board 70Ω dummy bus load resistors across BUS A and BUS B.

BC and RT Mode (HI-6138 and HI-6137)

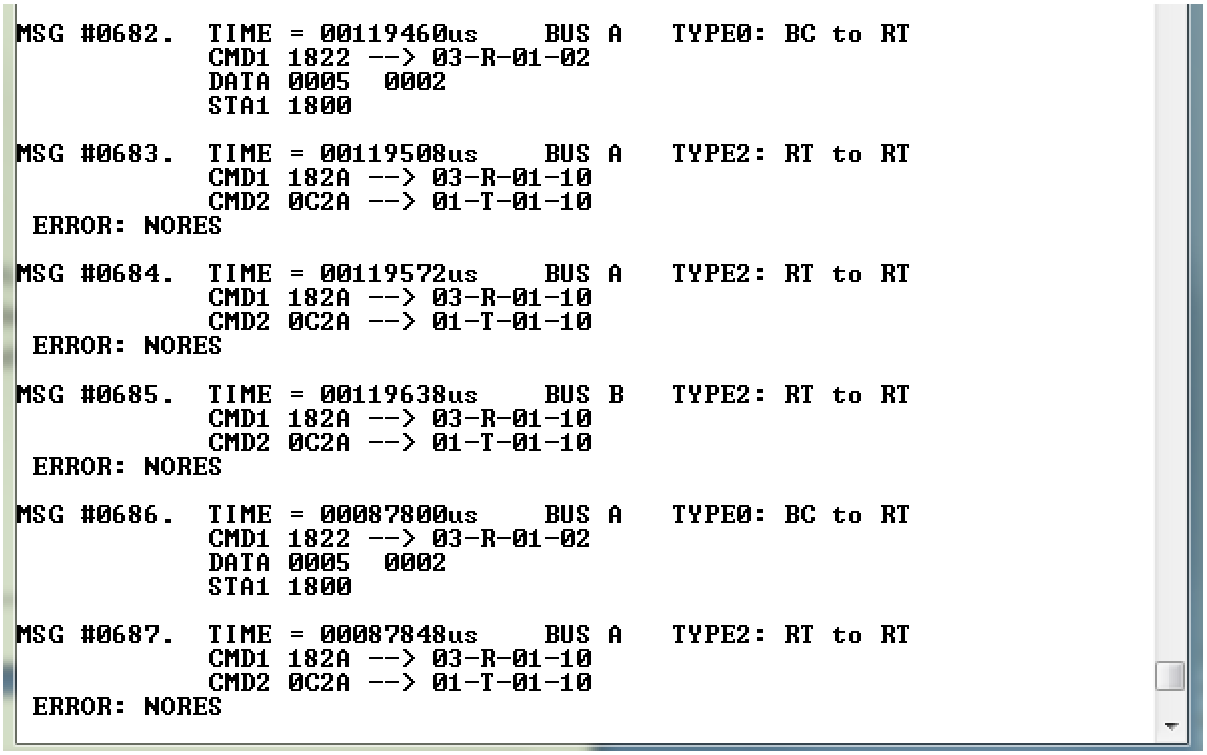
1. Press command 'B' to enable the RT.

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
```

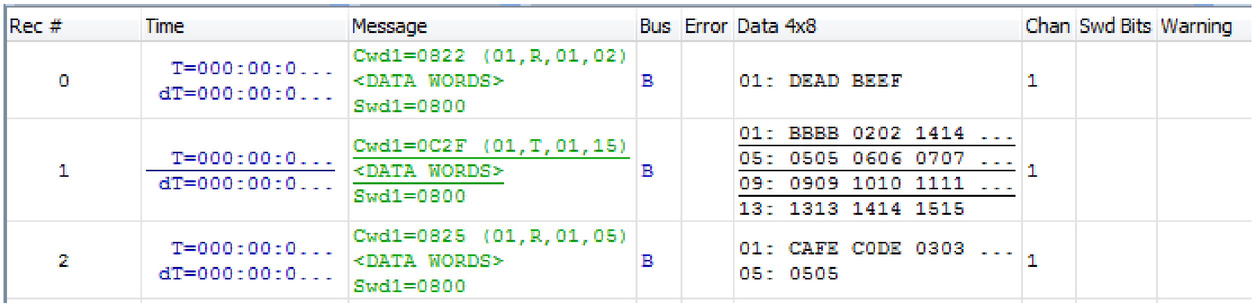
Some of the BC commands are RT to RT. The BC commands to RT3 only shows no errors but RT to RT commands to RT1 and RT3 will show “ ERROR: NORES” since there’s no RT at address 1.

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.

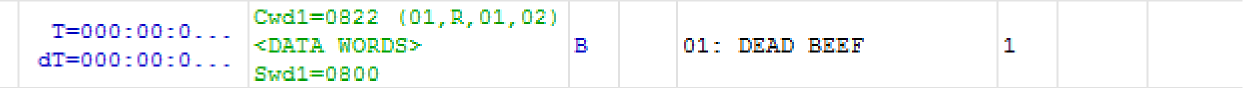
The screen shot of these three messages are shown below captured by a Ballard USB UA1133 tester.



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.



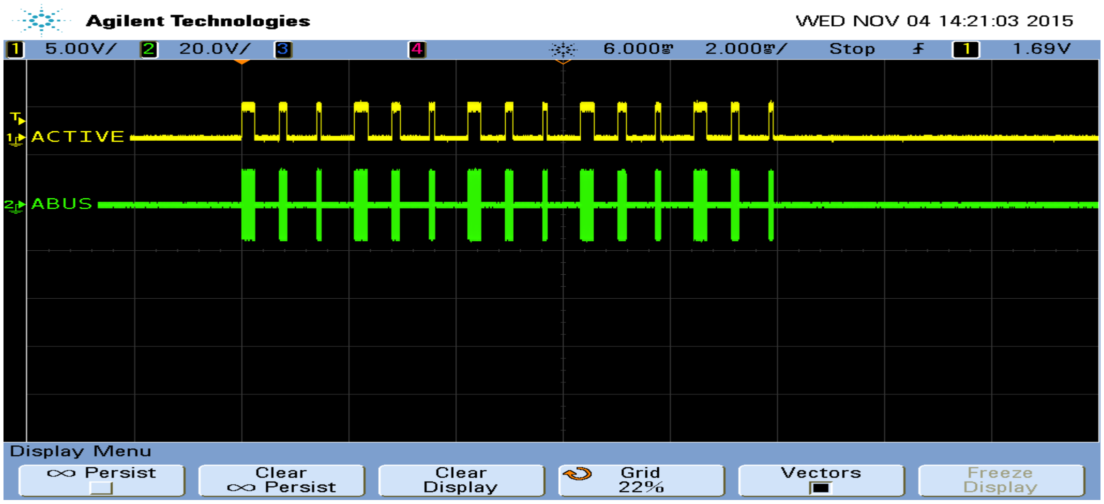
	T=000:00:0... dT=000:00:0...	Cwd1=0822 (01,R,01,02) <DATA WORDS> Swd1=0800	B		01: DEAD BEEF	1		

7. Command 'E' Enumerate Card is reserved for future use.

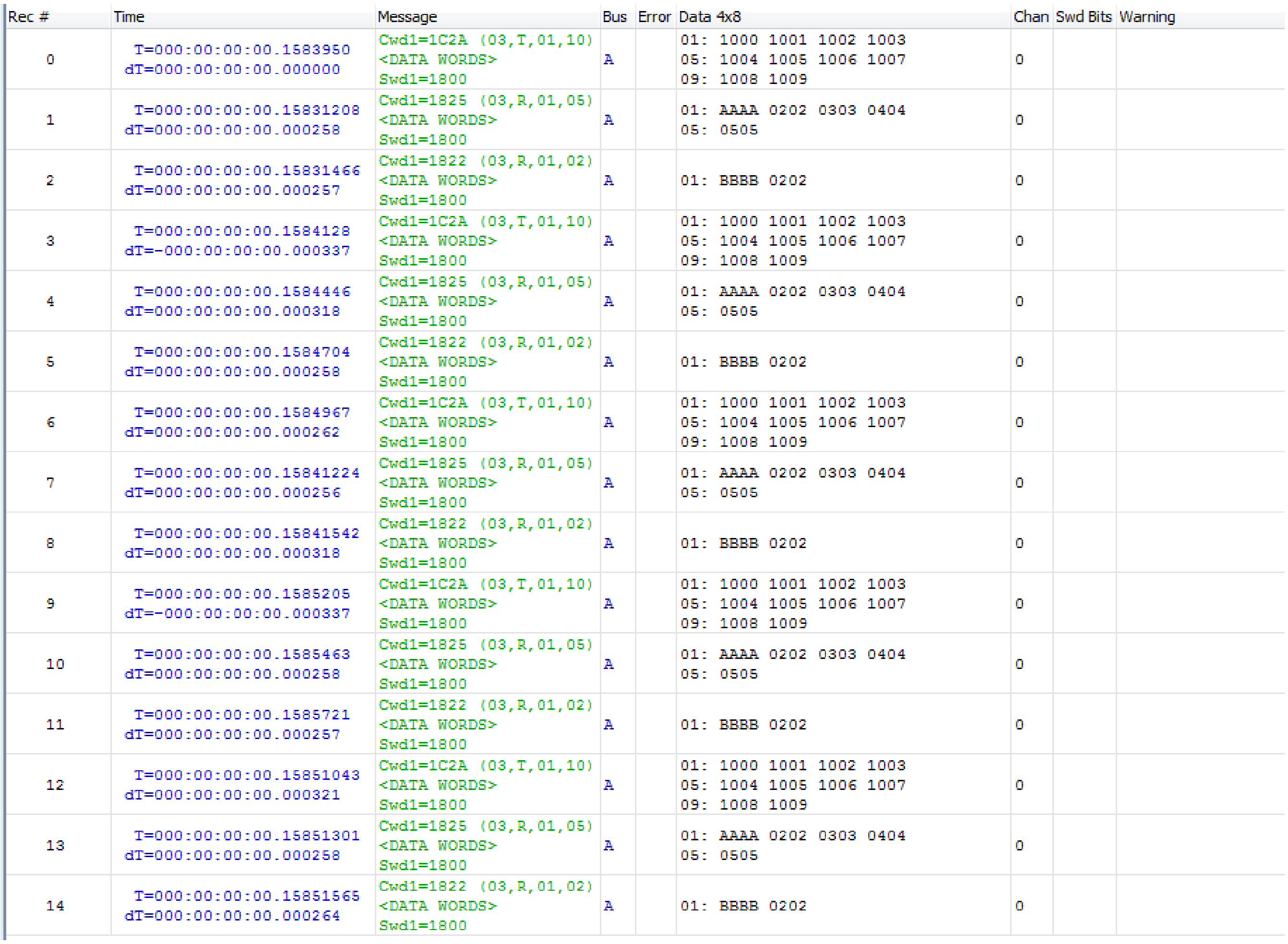
8. From a RESET, if the BC is started before enabling the 'K' and 'T' sequence to display message traffic, the first message may contain an error. This is normal; this occurs because the RT and MT are enabled midstream of a message in progress.

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 (HI-6135, 6136) using external BC

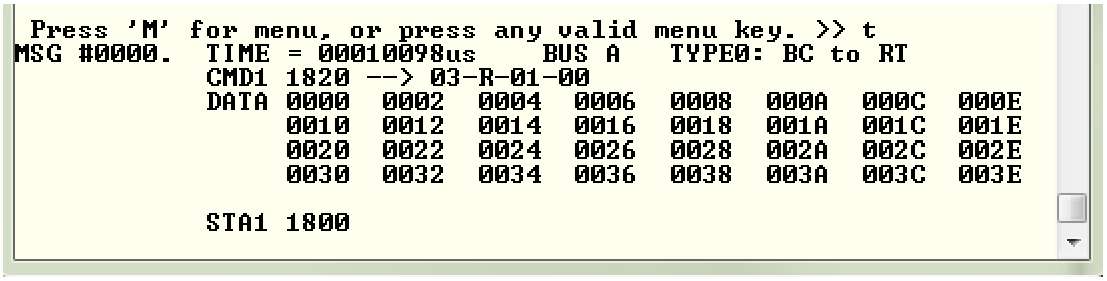
Using an external BC tester (such as Ballard USB 1553) to transmit messages to the demo board.

10. 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, the on-board dummy bus load 70 Ω resistors should be disconnected. See JP8 and JP9 described on page 6.

If bus couplers are not readily available, bench testing can be done by enabling the on-board dummy bus load 70 Ω resistors (see page 6) and connecting BC tester cables directly to the demo board tri-axial jacks for buses A and B.

11. Press the RESET button and then Press ‘B’ to enable the RT 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.

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



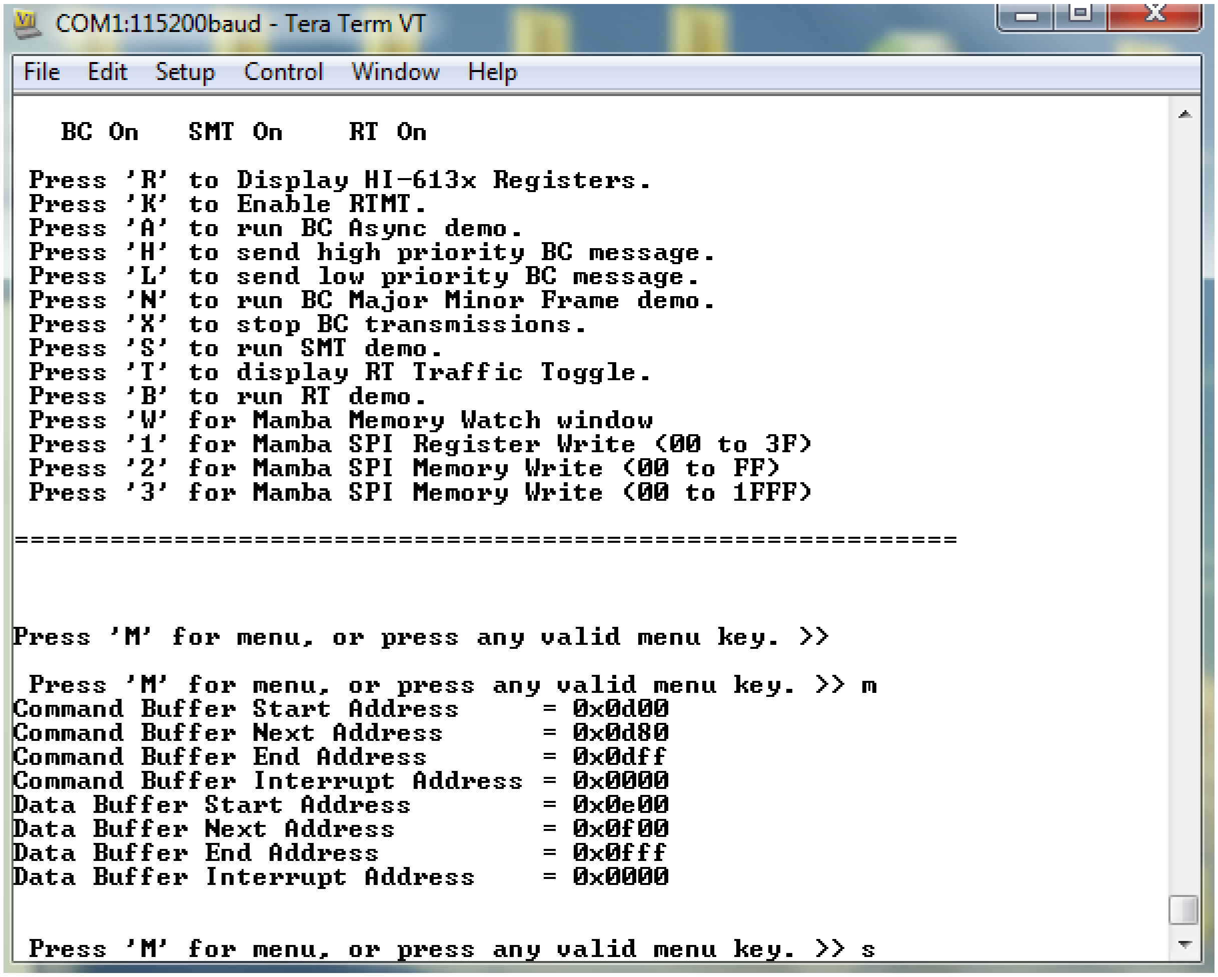
```
 Press 'M' for menu, or press any valid menu key. >> t
MSG #0000.   TIME = 00010098us    BUS A   TYPE0: BC to RT
             CMD1 1820 --> 03-R-01-00
             DATA 0000  0002  0004  0006  0008  000A  000C  000E
                  0010  0012  0014  0016  0018  001A  001C  001E
               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

             STA1 1800
```

13. 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 (HI-6136, 6138)

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.


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

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-6138 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.**

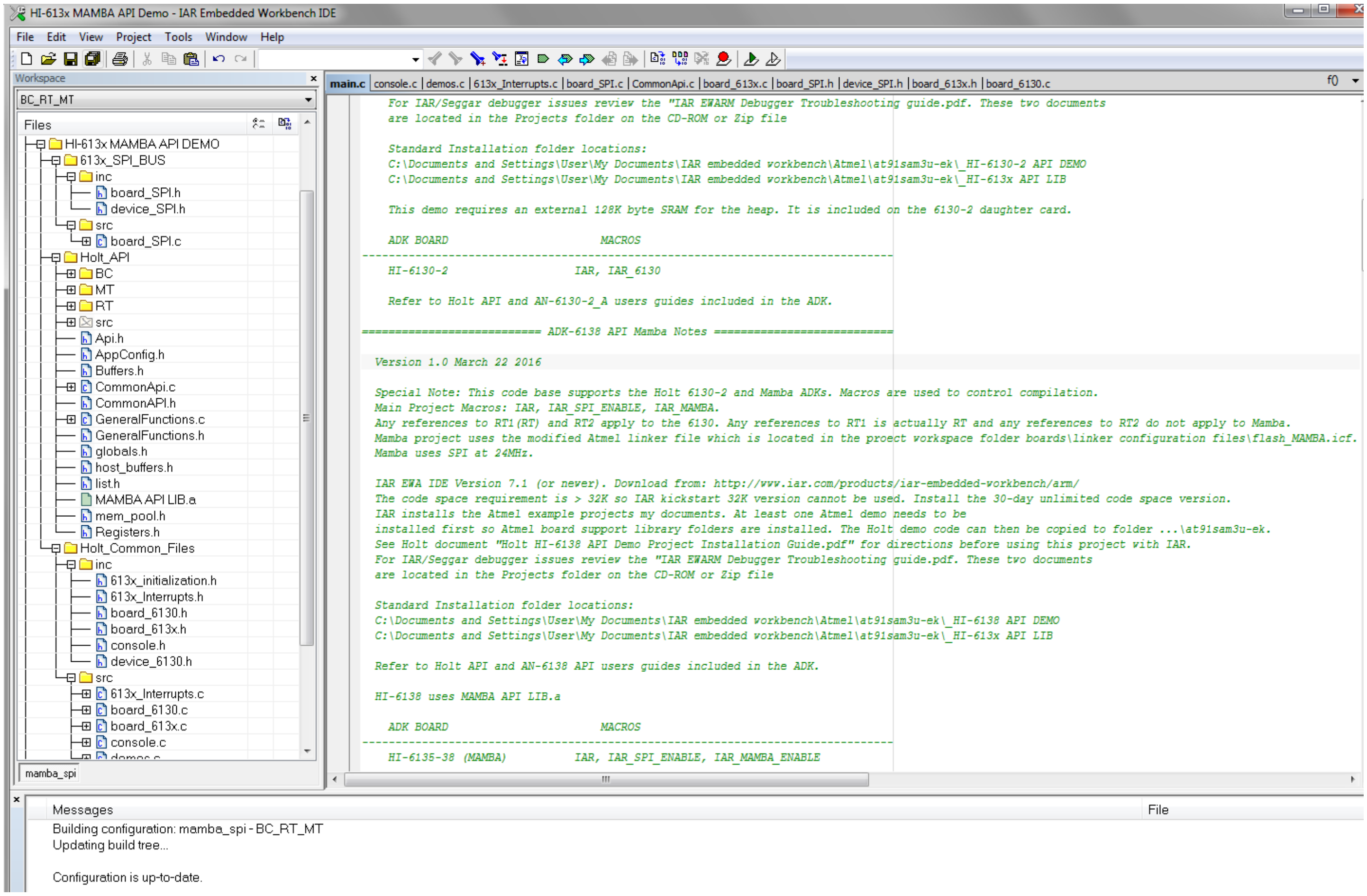
2. Launch IAR *Embedded Workbench* from the Windows Start menu. A blank screen should appear. Open the Holt HI-6138 API Demo Project from the IAR File pull-down menu, click on File/Open/Workspace and navigate to the project folder location and select "HI-6138 API Demo.eww" and click the Open button.

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 below.

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.



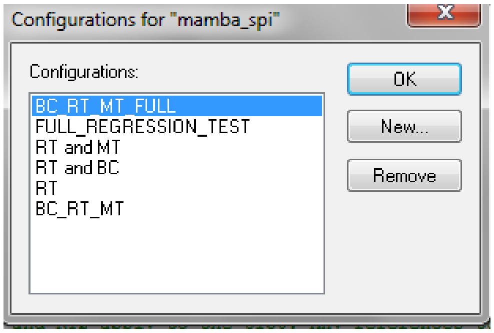
```
    For IAR/Seggar debugger issues review the "IAR EWARM Debugger Troubleshooting guide.pdf. These two documents
    are located in the Projects folder on the CD-ROM or Zip file

    Standard Installation folder locations:
    C:\Documents and Settings\User\My Documents\IAR embedded workbench\Atmel\at91sam3u-ek\_HI-6130-2 API DEMO
    C:\Documents and Settings\User\My Documents\IAR embedded workbench\Atmel\at91sam3u-ek\_HI-613x API LIB

    This demo requires an external 128K byte SRAM for the heap. It is included on the 6130-2 daughter card.

    ADK BOARD                    MACROS
--------------------------------------------------------------------------------
    HI-6130-2                  IAR, IAR_6130

    Refer to Holt API and AN-6130-2_A users guides included in the ADK.

=========================== ADK-6138 API Mamba Notes ===========================

  Version 1.0 March 22 2016

  Special Note: This code base supports the Holt 6130-2 and Mamba ADKs. Macros are used to control compilation.
  Main Project Macros: IAR, IAR_SPI_ENABLE, IAR_MAMBA.
  Any references to RT1(RT) and RT2 apply to the 6130. Any references to RT1 is actually RT and any references to RT2 do not apply to Mamba.
  Mamba project uses the modified Atmel linker file which is located in the proect workspace folder boards\linker configuration files\flash_MAMBA.icf.
  Mamba uses SPI at 24MHz.

  IAR EWA IDE Version 7.1 (or newer). Download from: http://www.iar.com/products/iar-embedded-workbench/arm/
  The code space requirement is > 32K so IAR kickstart 32K version cannot be used. Install the 30-day unlimited code space version.
  IAR installs the Atmel example projects my documents. At least one Atmel demo needs to be
  installed first so Atmel board support library folders are installed. The Holt demo code can then be copied to folder ...\at91sam3u-ek.
  See Holt document "Holt HI-6138 API Demo Project Installation Guide.pdf" for directions before using this project with IAR.
  For IAR/Seggar debugger issues review the "IAR EWARM Debugger Troubleshooting guide.pdf. These two documents
  are located in the Projects folder on the CD-ROM or Zip file

  Standard Installation folder locations:
  C:\Documents and Settings\User\My Documents\IAR embedded workbench\Atmel\at91sam3u-ek\_HI-6138 API DEMO
  C:\Documents and Settings\User\My Documents\IAR embedded workbench\Atmel\at91sam3u-ek\_HI-613x API LIB

  Refer to Holt API and AN-6138 API users guides included in the ADK.

  HI-6138 uses MAMBA API LIB.a

    ADK BOARD                    MACROS
--------------------------------------------------------------------------------
    HI-6135-38 (MAMBA)         IAR, IAR_SPI_ENABLE, IAR_MAMBA_ENABLE
```

```
Building configuration: mamba_spi - BC_RT_MT
Updating build tree...

Configuration is up-to-date.
```

Mode Holt Project Configurations

IAR project configurations are used to re configure 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_MAMBA_ENABLE** control which sections of code are compiled for the MAMBA 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 decide which sections of code to build and link. Some references to HI-6130 or HI-6131 are to be expected. In some cases some functions are simply not used or macros are used inside a function to select appropriate code.

The configuration list may vary slightly from the list shown.



Configurations for "mamba_spi"

Configurations:

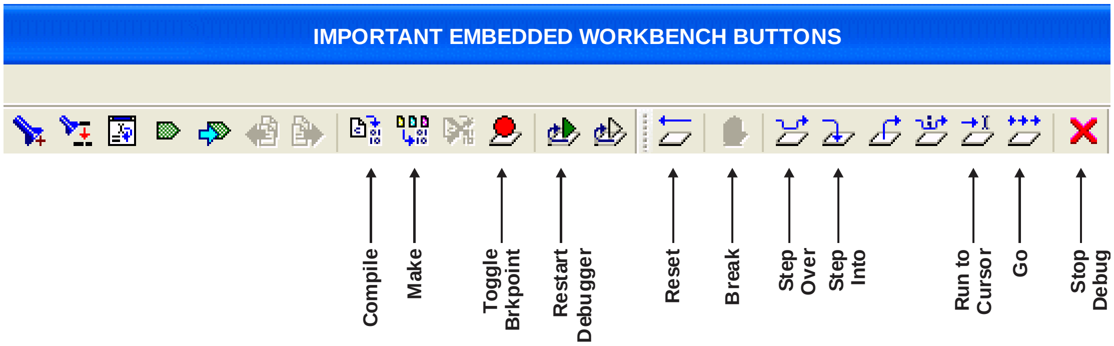
- BC_RT_MT_FULL
- FULL_REGRESSION_TEST
- RT and MT
- RT and BC
- RT
- BC_RT_MT

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 6131 or 6130 they were written for, don't worry about this they work fine with the MAMBA™ family.

device_SPI.h

Macro definitions for HI-6135/6/7/8 register addressing.

board_SPI.c, board_SPI.h

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

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. Many of the pins, SPI etc. are common between MAMBA and HI-6131 since both use SPI.

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.

MAMBA 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.

MAMBA^{TM} SPI Interface

MAMBA^{TM} features a four wire Serial Peripheral Interface (SPI) to the host MCU or FPGA. The device is offered in a plastic QFP, or 6 mm x 6 mm QFN package. When using the API, the interrupt concerns expressed below are already addressed in the API runtime library.

The MAMBA^{TM} data transfer speed depends on the SPI clock frequency provided by the MCU SPI interface. When the SPI is clocked at the maximum SCK frequency, 40 MHz (this demo uses 24MHz), each 16 bit word is transferred in 400 ns, plus the overhead associated with SPI op code execution. A Memory Address Pointer (register) is initialized by the MCU or FPGA before a read or write operation begins. The read/write operation is then initiated using an 8-bit SPI op code, serially shifted into MAMBA^{TM} by the MCU or FPGA. The host then continues clocking SCK in 16-clock multiples to read or write successive RAM or register addresses. As long as clocking continues, successive addresses are read or written. Potential problems occur when interrupts are enabled during a multi-word access. If the program's interrupt handler seizes the SPI bus to service the interrupt, it potentially disrupts an unfinished multi-word transfer. Without proper software design, a simple return-from-interrupt results in a broken multi-word transfer because the hardware doesn't know that an interrupt occurred and the Memory Address Pointer may or may not contain the RAM or register address for the next location in the interrupted multi-word transfer.

During MAMBA^{TM} SPI transfers, interrupts must be disabled. The simplest implementation disables interrupts before sending the op code, it then re-enables interrupts after reading or writing the last word in the multi-word transfer. If these causes unacceptable interrupt latency, some careful software design is needed. With suitable precautions, interrupts can be momentarily re-enabled then immediately disabled between SPI words. A pending interrupt that occurred during the interrupt-off interval will be immediately recognized when interrupts are re-enabled. The pending interrupt's service routine will execute; the return-from-interrupt will jump to and execute the following "disable interrupt" statement.

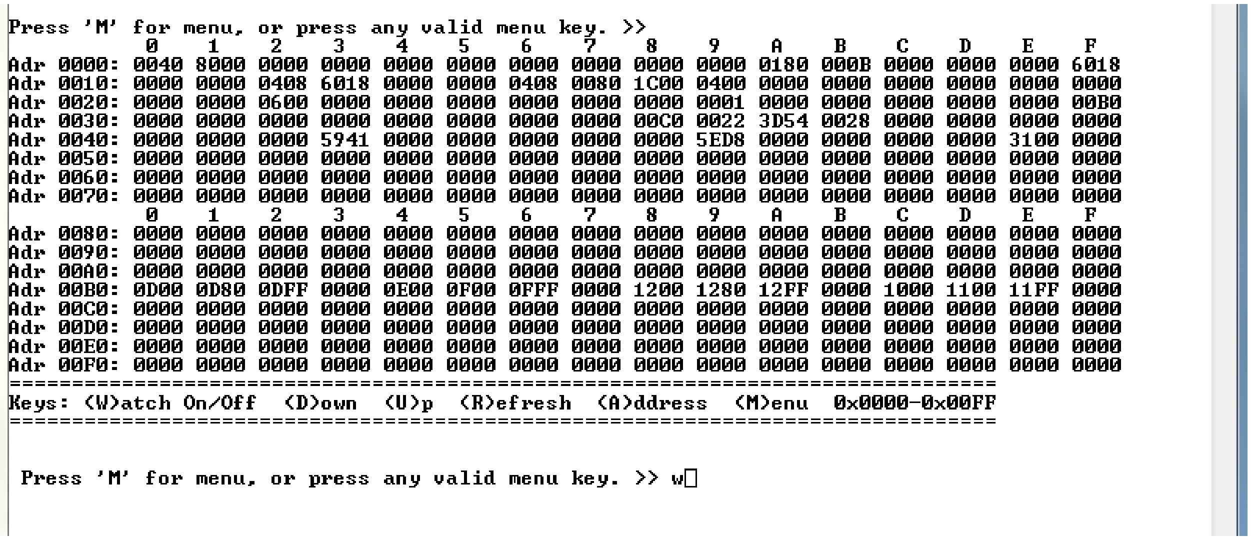
The example MAMBA^{TM} software successfully completes an interrupted multi-word sequence by using a "SPI interrupt occurred" flag, tested each time interrupts are momentarily disabled then re-enabled between SPI words. Upon detection of interrupt servicing between words, the program re-initializes the memory address pointer for the next word, then re-issues the original op code to resume the interrupted multi-word transfer. Nesting interrupts 3 or more levels would be challenging.

When using MAMBA^{TM}, SPI access is incompatible with C structures for table addressing. Register/RAM inspection, watch windows are unavailable when using MAMBA^{TM}. Instead a utility function, like the HI-6135/6/7/8 demo program's Memory_watch() function, must be written in C to read a range of addresses, and format the data for display using console I/O or some other display means. Of course the application program must be running to call the display function when Memory_watch() is needed. For advanced users, other utilities are also supplied for SPI writing to registers and memory addresses.

Application Development Kit Notes

MAMBA^{TM} 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 since these cannot be viewed from the IAR Embedded Workbench debugger, this tool will not work with the MAMBA^{TM} SPI interface. The demonstration program provides similar capability via SPI, by using a C function called Memory_watch(). This function call only works when the 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 0x1FFF 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:



```
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 0001 0000 0000 0000 0000 0000 00B0
Adr 0030: 0000 0000 0000 0000 0000 0000 0000 0000 00C0 0022 3D54 0028 0000 0000 0000 0000
Adr 0040: 0000 0000 0000 5941 0000 0000 0000 0000 0000 5ED8 0000 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 0DFF 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
```

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.

Be mindful that each displayed location is rescanned when Memory_watch( ) executes. Some register or RAM structure bits automatically reset after each 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 MAMBA™ demonstration program covers for all devices in the MAMBA™ family. The default set up has Bus Controller, Remote Terminal and Simple Monitor all enabled.

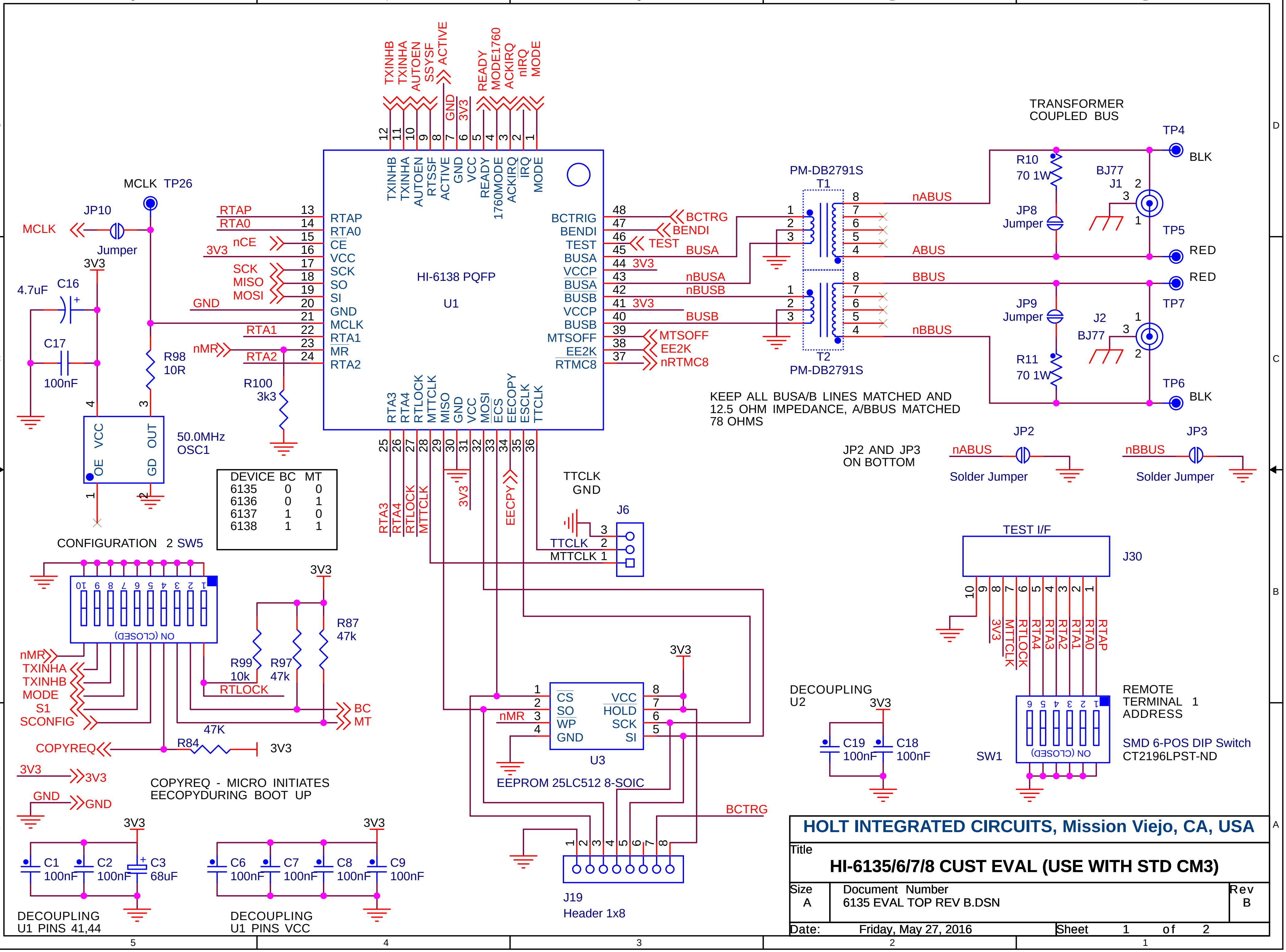
Summary

With just 4 host interface signals for accessing RAM or registers, the MAMBA™ 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 MAMBA™ programming by providing high level C function calls that greatly accelerate project development.

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



DEVICE	BC	MT
6135	0	0
6136	0	1
6137	1	0
6138	1	1

KEEP ALL BUSA/B LINES MATCHED AND
12.5 OHM IMPEDANCE, A/BBUS MATCHED
78 OHMS

JP2 AND JP3
ON BOTTOM

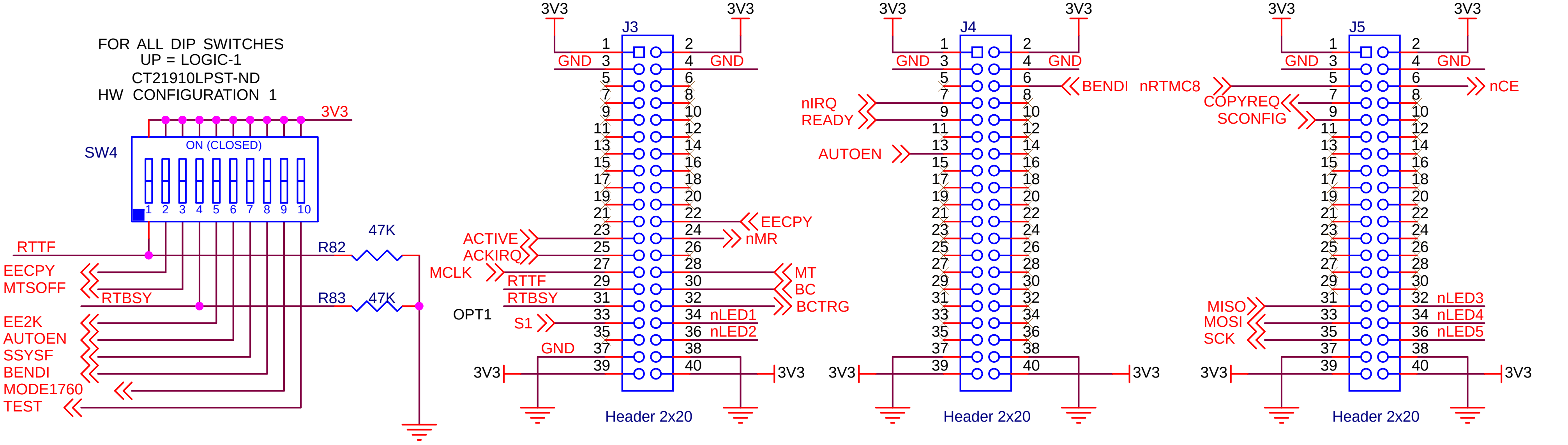
COPYREQ - MICRO INITIATES
EECOPYDURING BOOT UP

HOLT INTEGRATED CIRCUITS, Mission Viejo, CA, USA

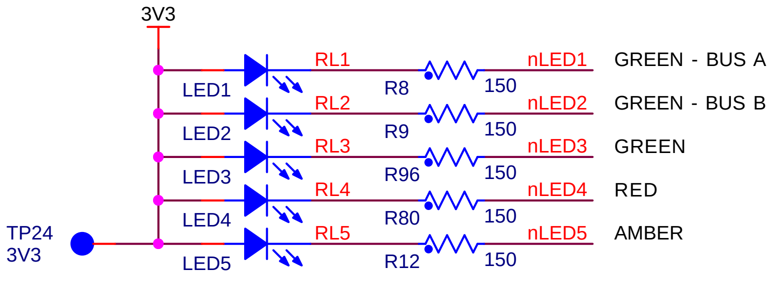
Title
HI-6135/6/7/8 CUST EVAL (USE WITH STD CM3)

Size	Document Number	Rev
A	6135 EVAL TOP REV B.DSN	B

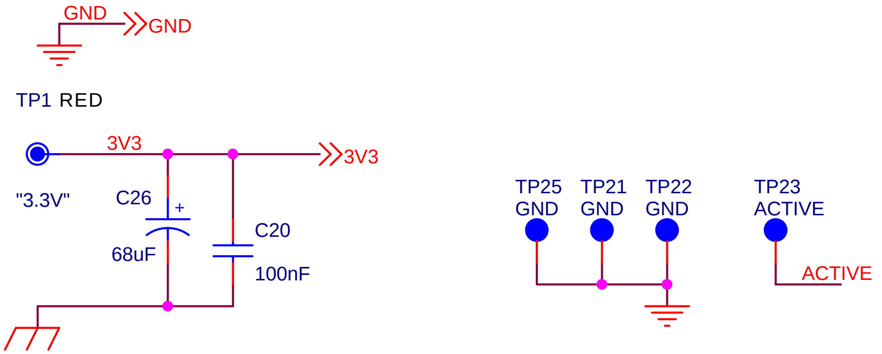
Date: Friday, May 27, 2016 Sheet 1 of 2



RTTF - SETS RT TERMINAL FLAG
EECPY - MAKES COPY OF RAM AND REGS TO EPROM
MTSOFF - 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
SSYSF - SET SSYF (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



STATUS LEDS



POWER SUPPLY -- 5VDC IN TO 3.3VDC OUT

HOLT INTEGRATED CIRCUITS, Mission Viejo, CA, USA

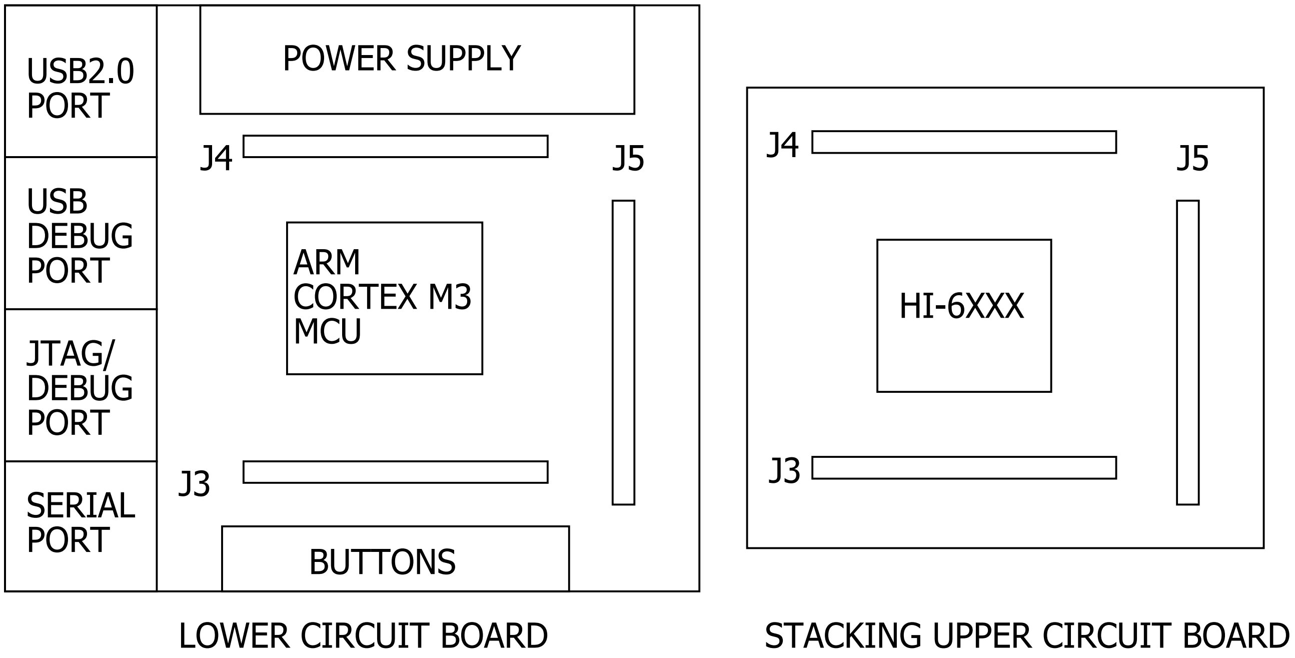
Title
HI-6135/6/7/8 CUST EVAL USE WITH STD CM3

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% 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

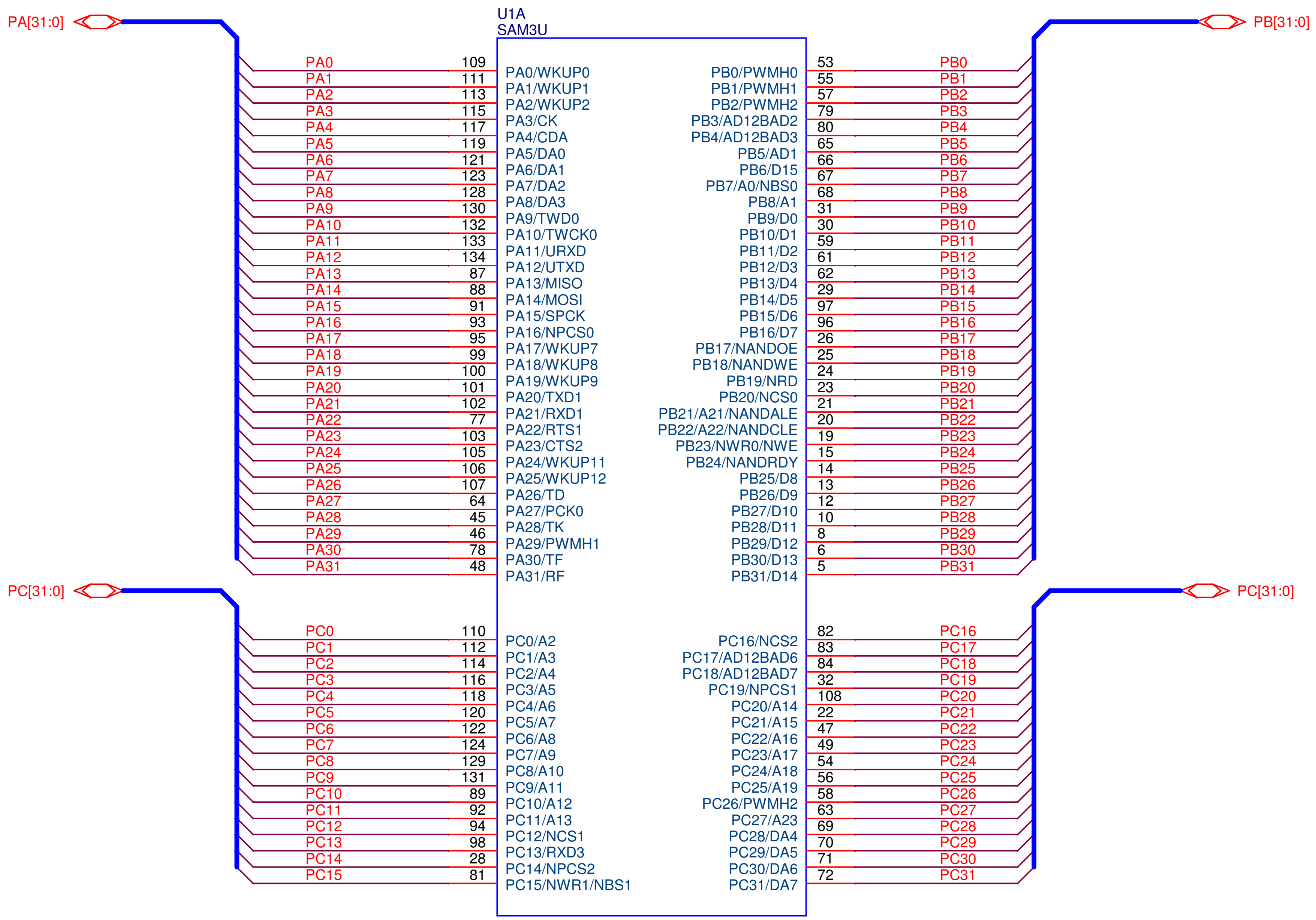


LOWER CIRCUIT BOARD

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	



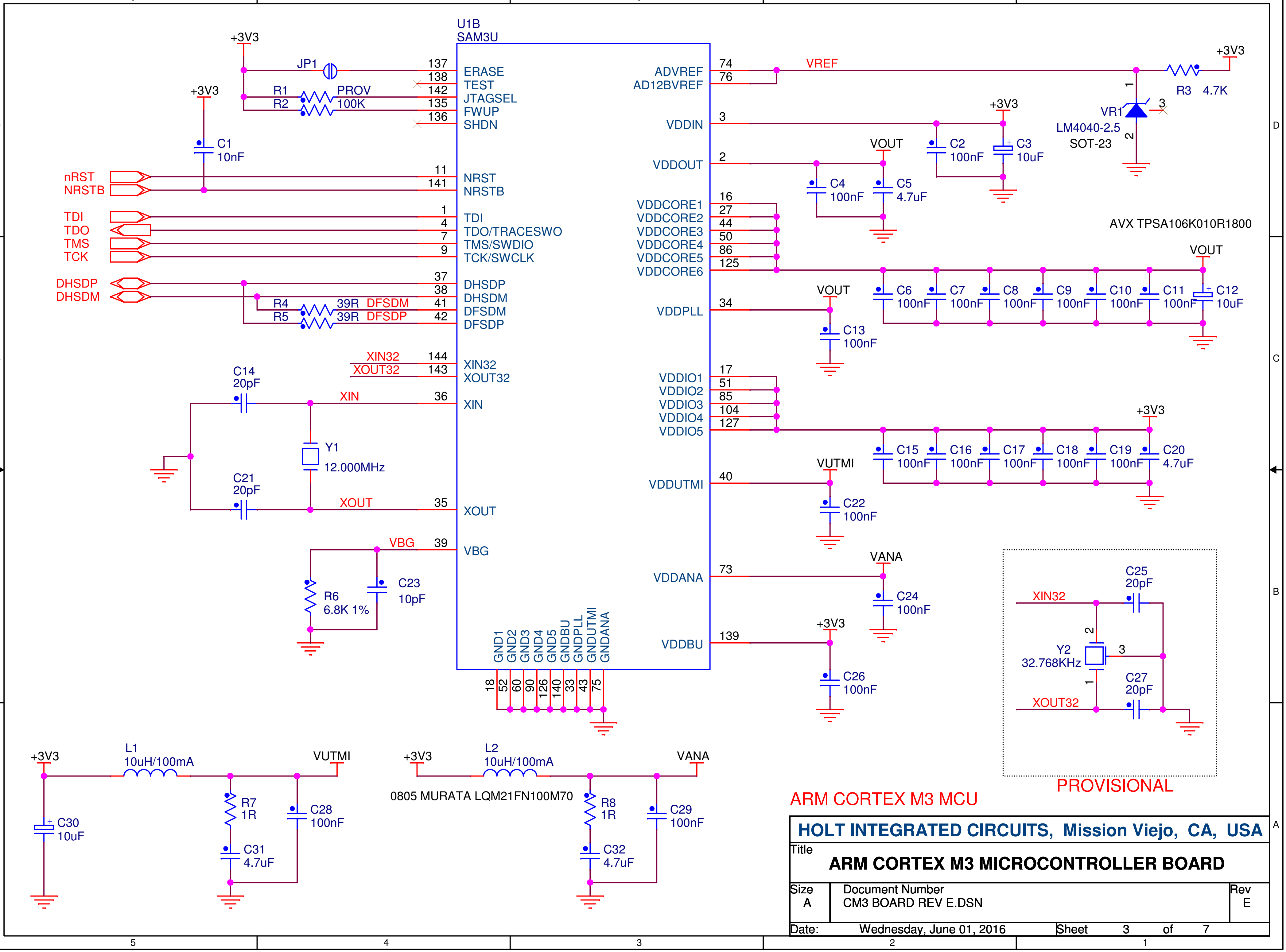
ARM CORTEX M3 PIO

HOLT INTEGRATED CIRCUITS, Mission Viejo, CA, USA

Title
ARM CORTEX M3 MICROCONTROLLER BOARD

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

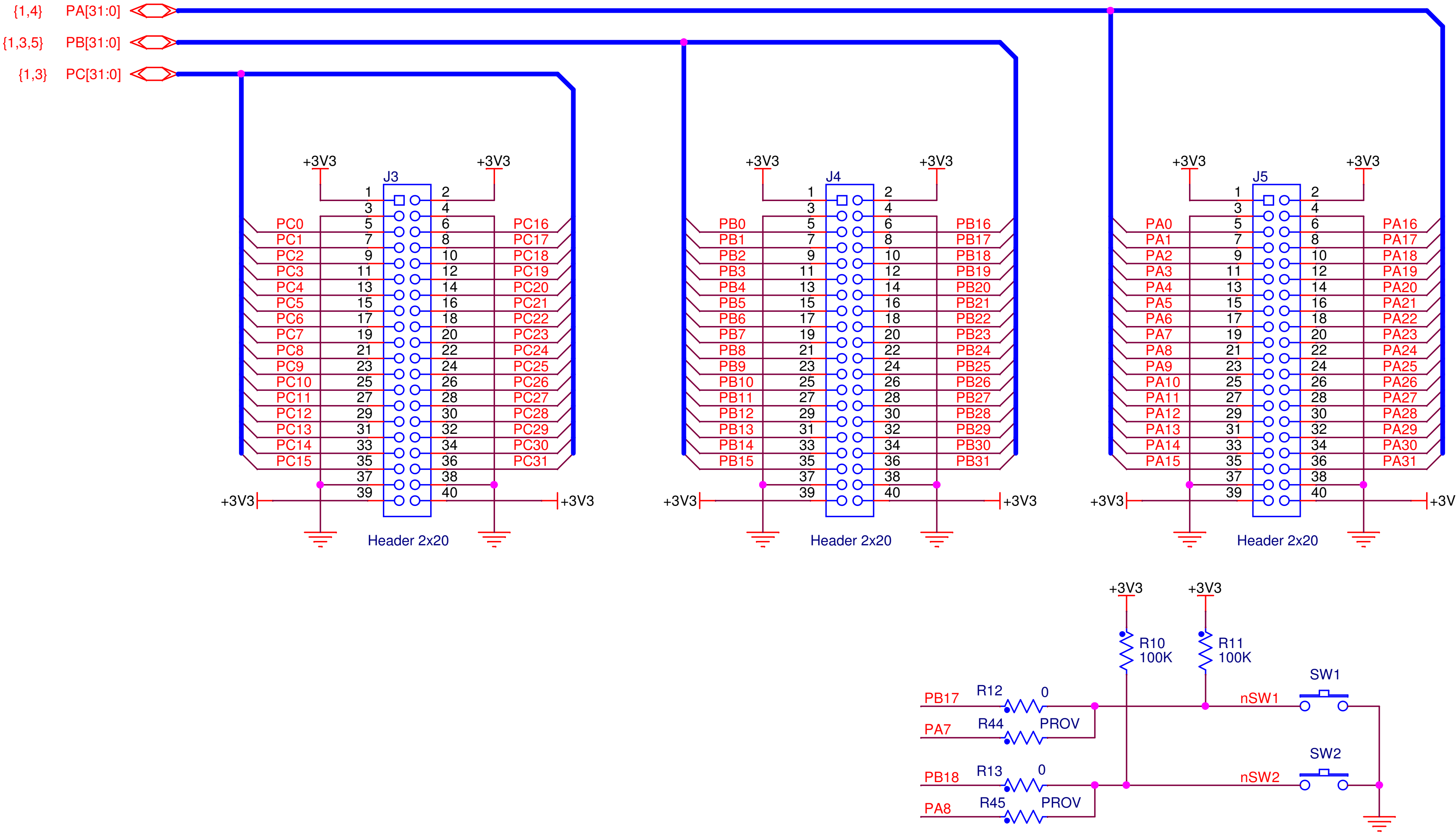
Date: Wednesday, June 01, 2016 Sheet 2 of 7



ARM CORTEX M3 MCU

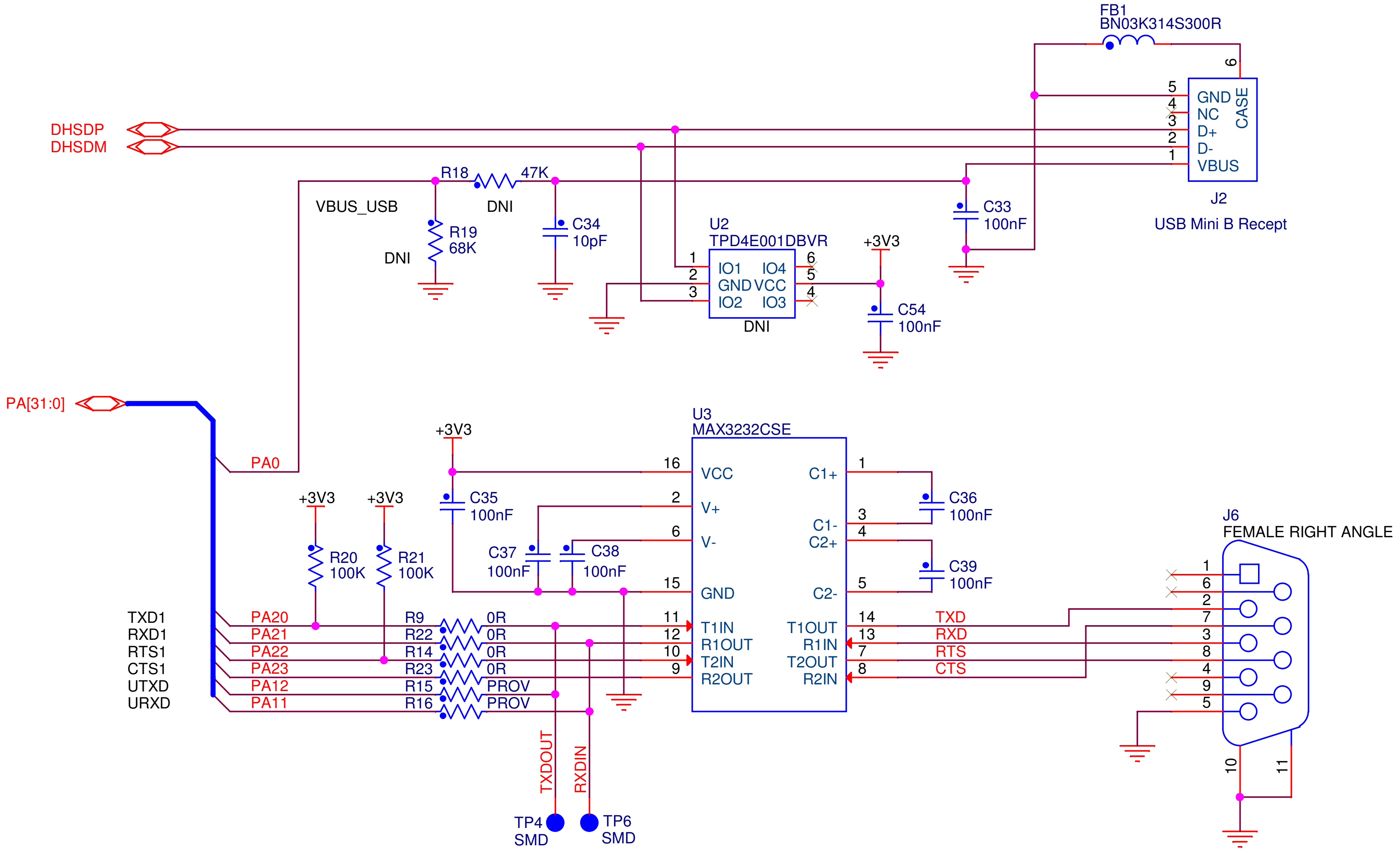
HOLT INTEGRATED CIRCUITS, Mission Viejo, CA, USA

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	


{1,4} PA[31:0]
{1,3,5} PB[31:0]
{1,3} PC[31:0]
J3
J4
J5
Header 2x20
Header 2x20
Header 2x20
+3V3
GND
PC0 PC1 PC2 PC3 PC4 PC5 PC6 PC7 PC8 PC9 PC10 PC11 PC12 PC13 PC14 PC15
PC16 PC17 PC18 PC19 PC20 PC21 PC22 PC23 PC24 PC25 PC26 PC27 PC28 PC29 PC30 PC31
PB0 PB1 PB2 PB3 PB4 PB5 PB6 PB7 PB8 PB9 PB10 PB11 PB12 PB13 PB14 PB15
PB16 PB17 PB18 PB19 PB20 PB21 PB22 PB23 PB24 PB25 PB26 PB27 PB28 PB29 PB30 PB31
PA0 PA1 PA2 PA3 PA4 PA5 PA6 PA7 PA8 PA9 PA10 PA11 PA12 PA13 PA14 PA15
PA16 PA17 PA18 PA19 PA20 PA21 PA22 PA23 PA24 PA25 PA26 PA27 PA28 PA29 PA30 PA31
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29 31 33 35 37 39
2 4 6 8 10 12 14 16 18 20 22 24 26 28 30 32 34 36 38 40
R10 100K
R11 100K
R12 0
R44 PROV
R13 0
R45 PROV
PB17
PA7
PB18
PA8
nSW1
nSW2
SW1
SW2

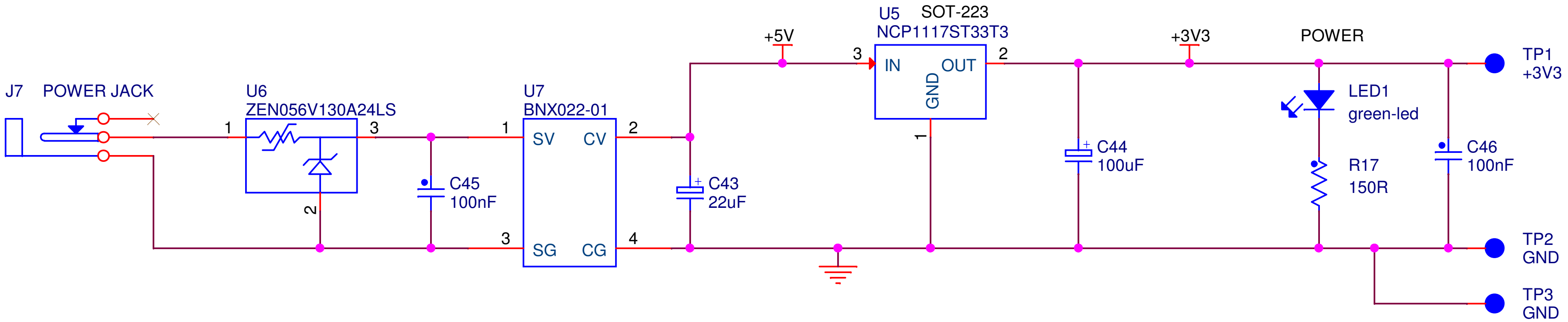
BOARD I/O HEADERS, BUTTONS

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

SEGGER J-LINK ON-BOARD DEBUGGER INTERFACE

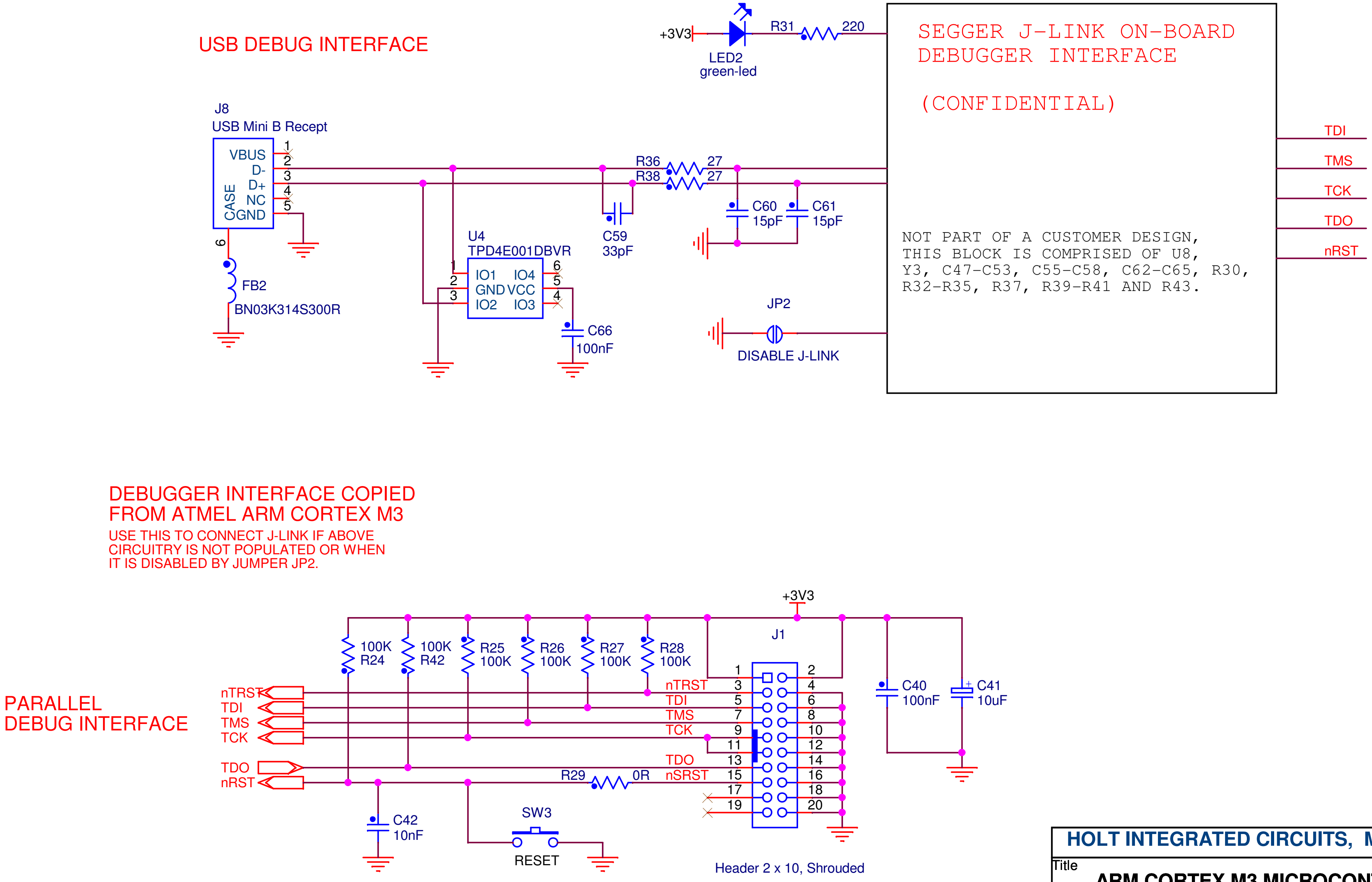
(CONFIDENTIAL)

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.

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