MAMBA™ Family
Application Development Kit

Devices Supported
HI-6135, HI-6136
HI-6137, HI-6138

June 2016

REVISION HISTORY

Revision	Date	Description of Change
AN-6138, Rev. New	07-20-15	Initial Release
Rev. A	08-18-15	Changed DIP switch defaults
Rev. B	06-01-16	Add schematic and BOM for Cortex™ M3 mother board. Update MAMBA schematic and BOM.

Introduction

The Holt Mamba™ Evaluation board demonstrates the broad feature set of Holt's MIL-STD-1553 Mamba™ family, consisting of:

HI-6135 Remote Terminal

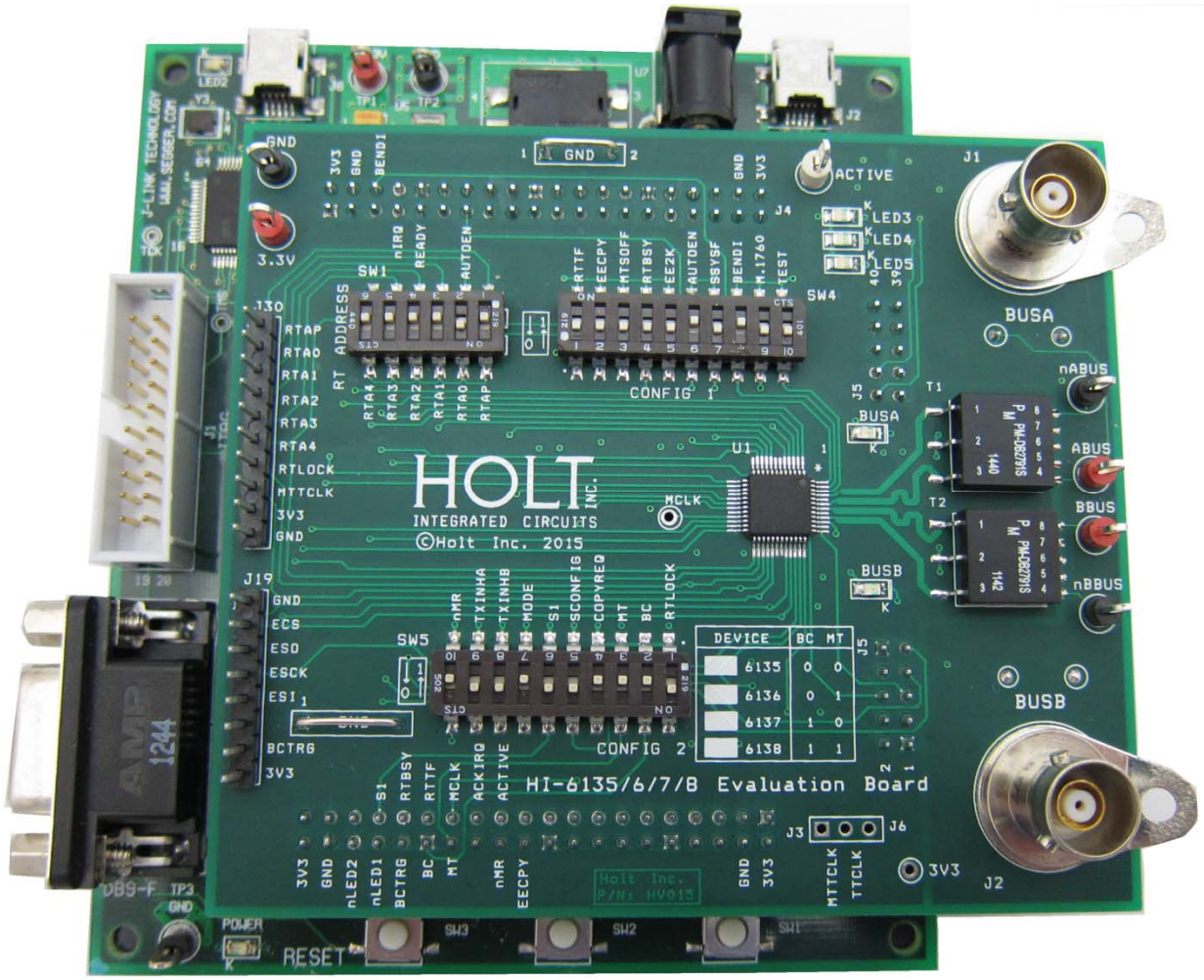
HI-6136 Remote Terminal and Monitor

HI-6137 Remote Terminal and Bus Controller

HI-6138 Remote Terminal, Bus Controller and Monitor

The Mamba™ family is a set of MIL-STD-1553B bus communication devices; containing protocol management 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. Note that in this Mamba™ guide the HI-6138 is used as the reference device because it contains all the available features, other devices have a subset of these features.

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.

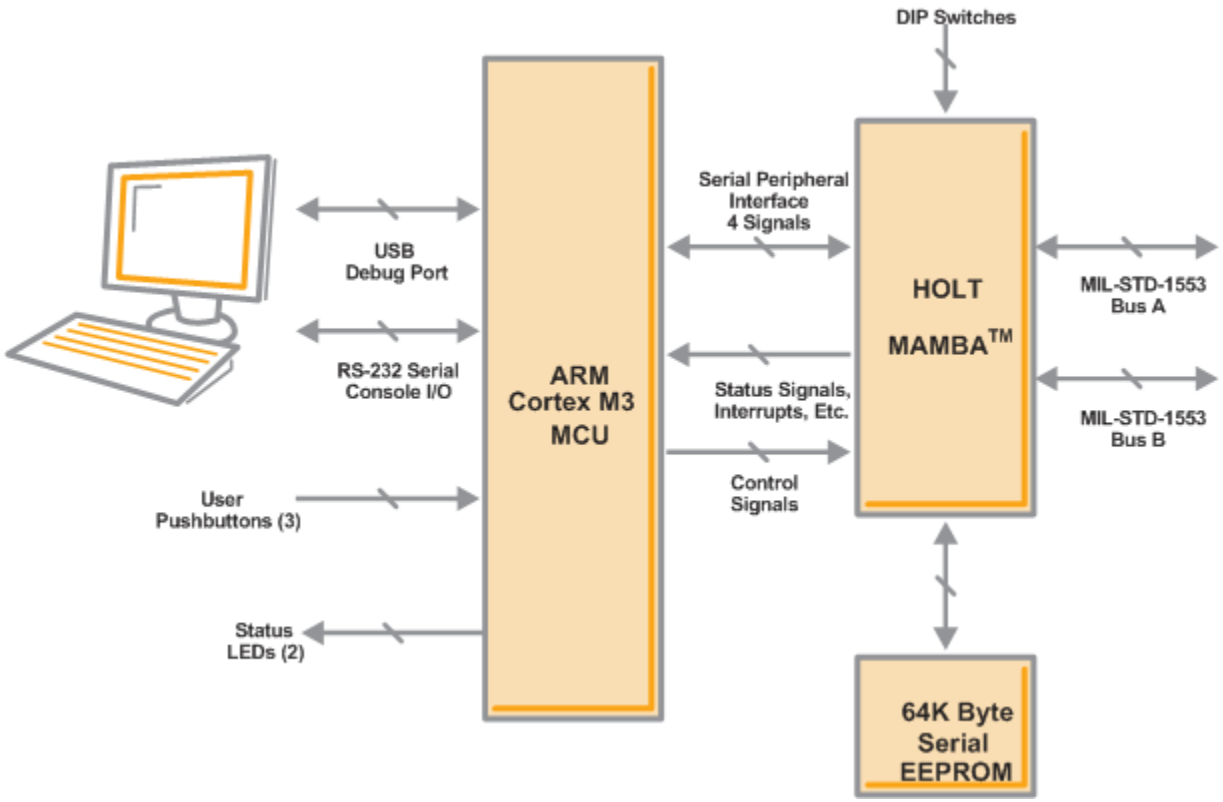


MAMBA™ Evaluation Board, mounted on the ARM Cortex MCU Board

Evaluation Kit Contents

- This User Guide.
- Holt Mamba™ Project Software and Documentation CD.
- Installation CD for IAR Systems Embedded Workbench® for ARM (32KB KickStart.).
- Plug-in 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 DUT board with Mamba™ 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 to 03
SW1, 1	OFF = 1	RTAP, RT address parity bit '1', must be odd parity or device will not work

Config 1

SWITCH	DEFAULT	DESCRIPTION
SW4, 1	OFF	RTTF – ON, sets terminal flag bit in RT status response
SW4, 2	OFF	EECPY – ON, makes copy of RAM and regs to EPROM
SW4, 3	OFF	MTSOFF – ON, disables memory test on power up
SW4, 4	OFF	RTBSY – ON, micro sets busy bit of status word
SW4, 5	OFF	EE2K – ON, EECPY only uses 2K words of EEPROM
SW4, 6	OFF	AUTOEN – ON , configuration boots from EEPROM
SW4, 7	OFF	SSYSF – ON , sets SSYF (system fail) bit in RT status
SW4, 8	ON	BENDI – ON , sets big endian for memory and register access
SW4, 9	OFF	MODE1760 – ON , Busy bit is set in Status word, directly after a hardware reset
SW4, 10	OFF	TEST – ON , sets test mode, not for customer use

CONFIG 2

SWITCH	DEFAULT	DESCRIPTION
SW5, 1	ON	RTLOCK– OFF locks RT address, so it cannot be overwritten
SW5, 2	OFF (6138, 6137) ON (6136,6135)	BC – OFF, enables BC mode through software
SW5, 3	OFF (6138, 6136) ON (6137,6135)	MT– OFF, enables Simple monitor mode through software
SW5, 4	ON	COPYREQ – OFF, software writes configuration to EEPROM after bootup
SW5, 5	ON	SCONFIG –Not Used
SW5, 6	ON	S1 – Not Used
SW5, 7	OFF	MODE – Device requires mode pin high
SW5, 8	ON	TXINHB – OFF, disables 1553 BUSB driver
SW5, 9	ON	TXINHA – OFF, disables 1553 BUSA driver
SW5, 10	OFF	nMR – ON will hold device in reset

Default Jumper Settings

JUMPER	POSITION	DESCRIPTION
JP2	ON	Link to connect the negative line of BUSA to the board ground.
JP3	ON	Link to connect the negative line of BUSB to the board ground.
JP8	ON	Link to connect 70Ω load resistor on BUSA.
JP9	ON	Link to connect 70Ω load resistor on BUSB.
JP10	OFF	Links 50MHz clock to micro board

Hardware Design Overview

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

The detachable DUT 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 configuration pins (and the Remote Terminal address setting pins) are controlled by DIP switches on the upper DUT 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) connects to the DUT. 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 DUT 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").

Initial Setting Up

The Holt Mamba™ Application Development Kit is designed to support all four devices from the family. The MAMBATM is used as the example because it will operate in all three modes; Remote Terminal (RT), Bus Controller (BC), SMT Bus Monitor (MT).

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.

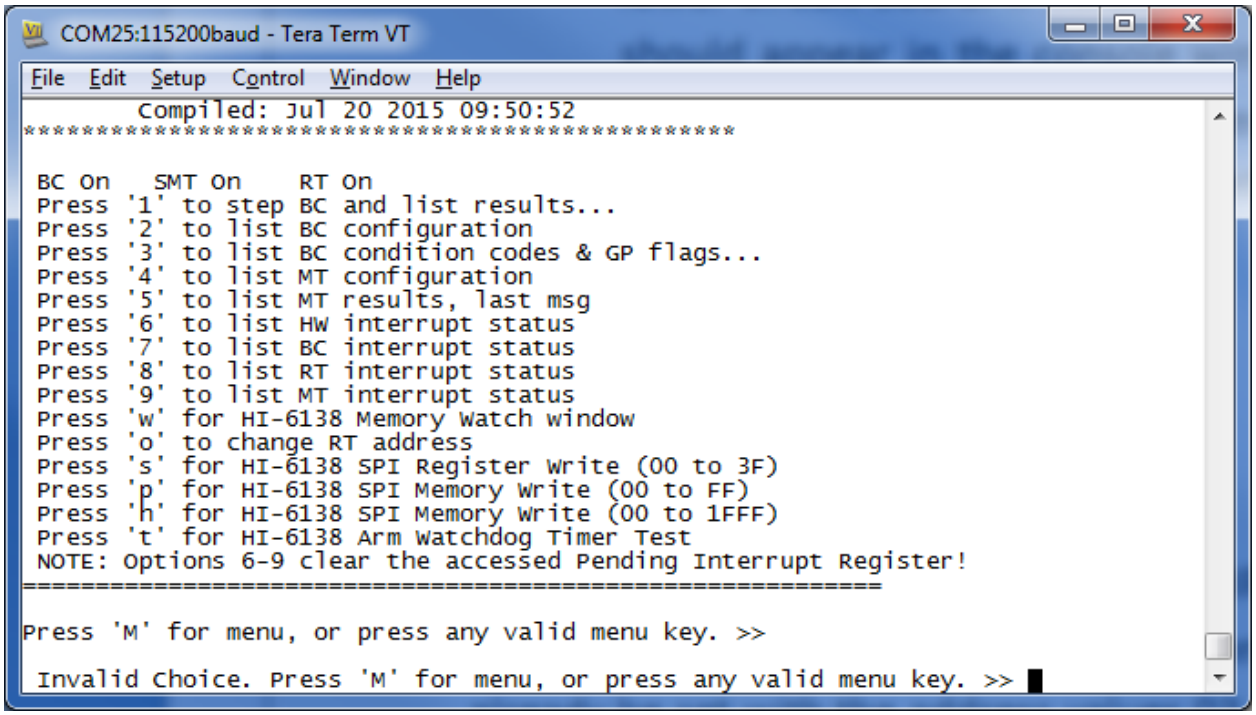
   If using Vista or Windows 7…

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 Plugin = 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 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**.



Testing Modes

The RT terminal address is set using DIP switches, before applying power. RT addresses 3 and 4 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.

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

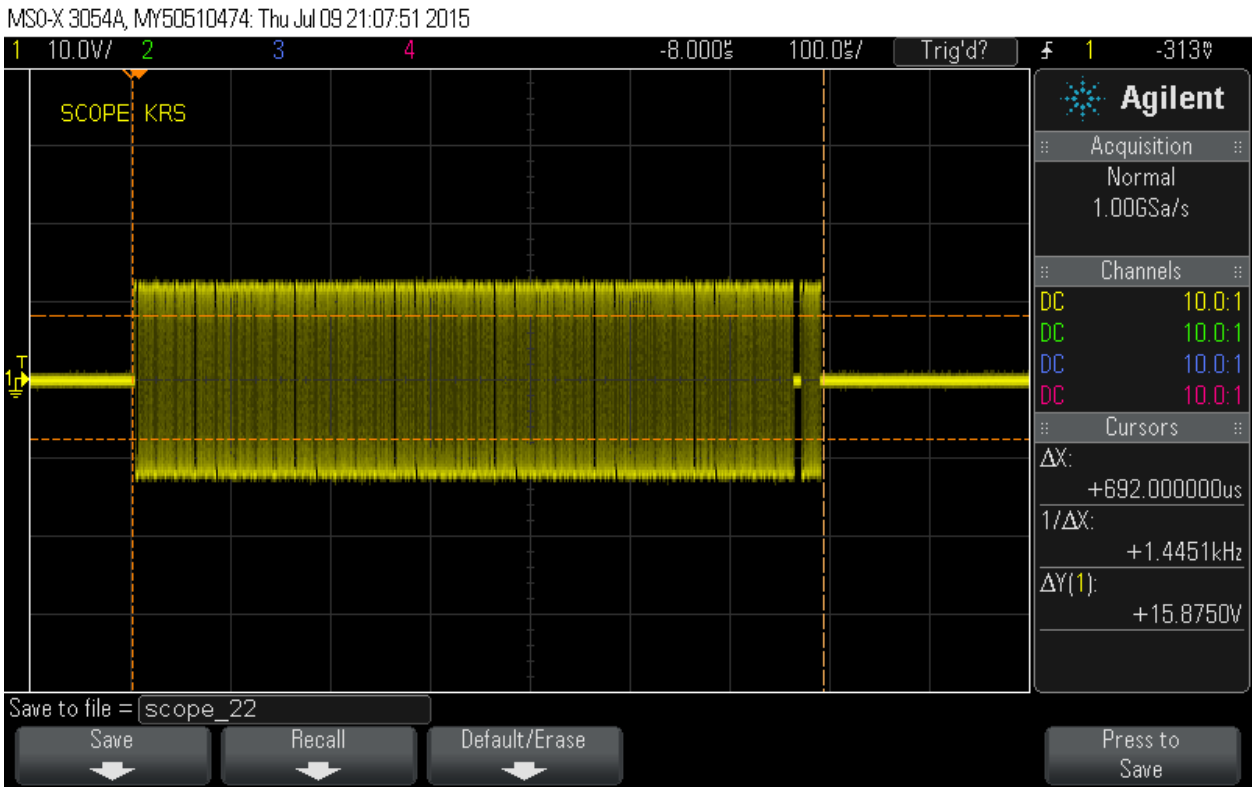
1. To observe bus activity, connect an oscilloscope to the red BUS A and red BUS B test points. The test point labeled ACTIVE is a convenient scope trigger signal.

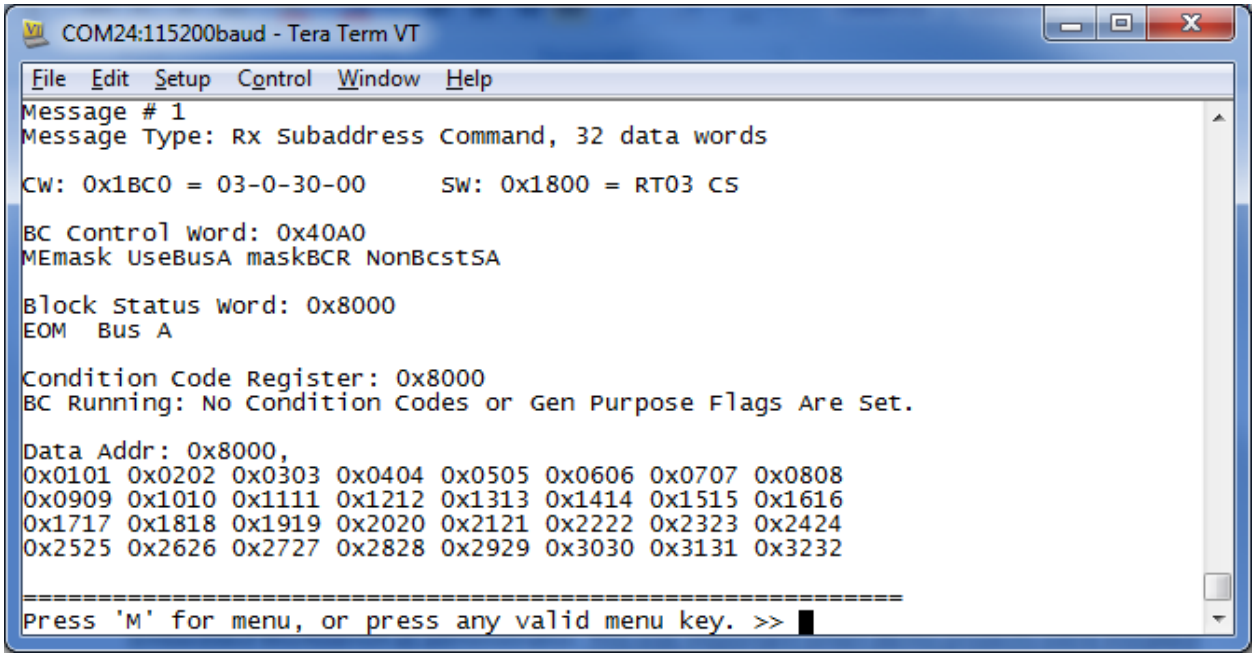
2. If not connected by cable to MIL-STD-1553 buses, a dummy 70Ω load for the buses is provided on the board by connecting solder jumper JP2 and JP3.

3. The Bus Controller is programmed to execute a repeating series of MIL-STD-1553 commands to Remote Terminal addresses 3 and 4. Each bus command is preceded by a BC "Wait for Trigger" op code. The MCU is preprogrammed to issue a trigger pulse to the BC each time the numeric "1" key is pressed on the computer keyboard. By turning off the console I/O option in 613x_initialization.h and recompiling the program, the Bus Controller command words can be triggered instead by using the SW1 button on the MCU board, but the TeraTerm console output will be disabled in this mode.

4. Each time the numeric "1" key is pressed on the computer keyboard, a new Bus Controller command is issued to RT address 3 or RT address 4 (or both RT addresses 3 and 4, in the case of RT-RT messages). Broadcast commands are also sent (these address all RT's and do not expect and RT response).

5. For RT testing, the board is shipped with the RT enabled, set address 03 then the device will respond to its own commands. The Remote Terminal responses can be observed, and the TeraTerm console screen reports new message results for each key press. An example of a response is shown below for message #1, the scope shot shows the BC sending 32 words of data to RT address 03, subaddress 30, after a short delay the RT responds with a Status word. The second picture shows the terminal screen after the same command.





RT Mode (HI-6135, 6136)

1. These devices do not contain the BC Mode that sends messages, to demonstrate the RT mode it is necessary to provide an external BC command. A separate board is required to generate the BC commands, for example the Holt 6130/31 Board can be used for this purpose.

2. Connect the external board to the BUSA or BUSB terminal of the ADK board. Make sure the RT address switches (SW1) are set to the correct address and the parity is odd. Transmit a command to the RT that requires a response (such Mode 2 command, 'Transmit Status Word') and observe the RT response on the BUS.

1760 Mode (all devices)

In this mode the device will respond with the busy bit set within 2ms of reset being removed. To test this feature the device should be powered up without the software running, this can be done by holding the micro in reset. If the nMR switch is toggled on the ADK (SW5/1) the device will respond to a BC command with the 'Busy' bit set.

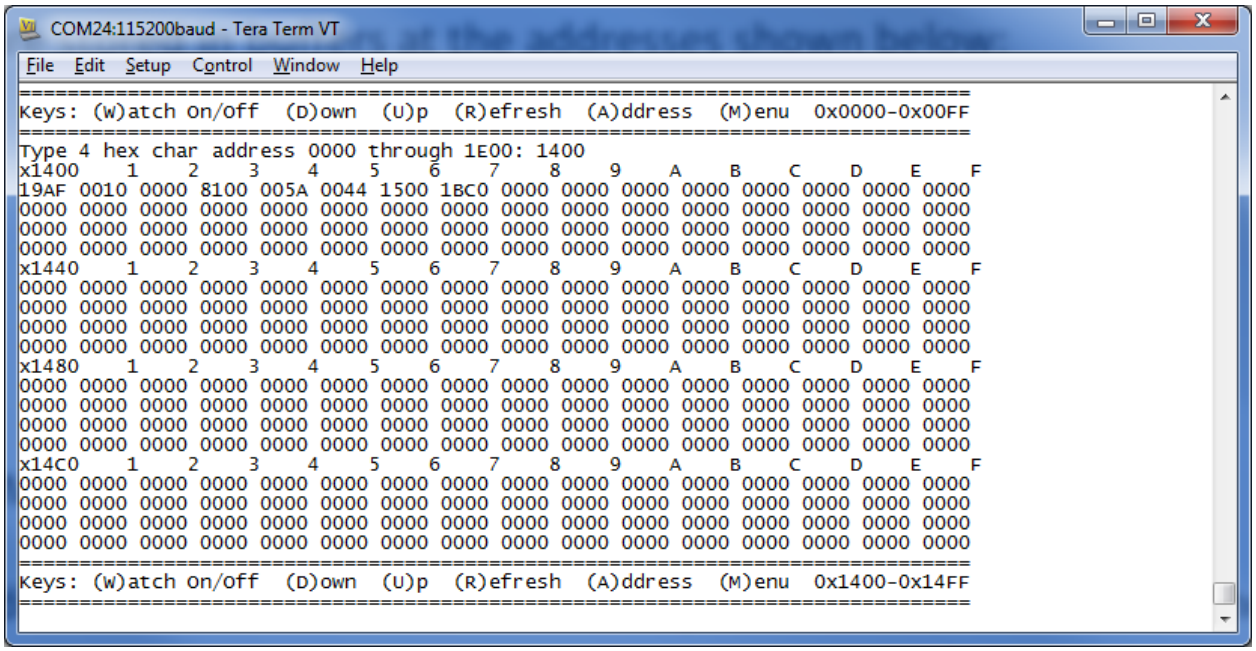
MT Mode (HI-6136, 6138)

In this mode the device will monitor and store all the commands and data sent on the Bus. The commands are stored in buffers at the addresses shown below:

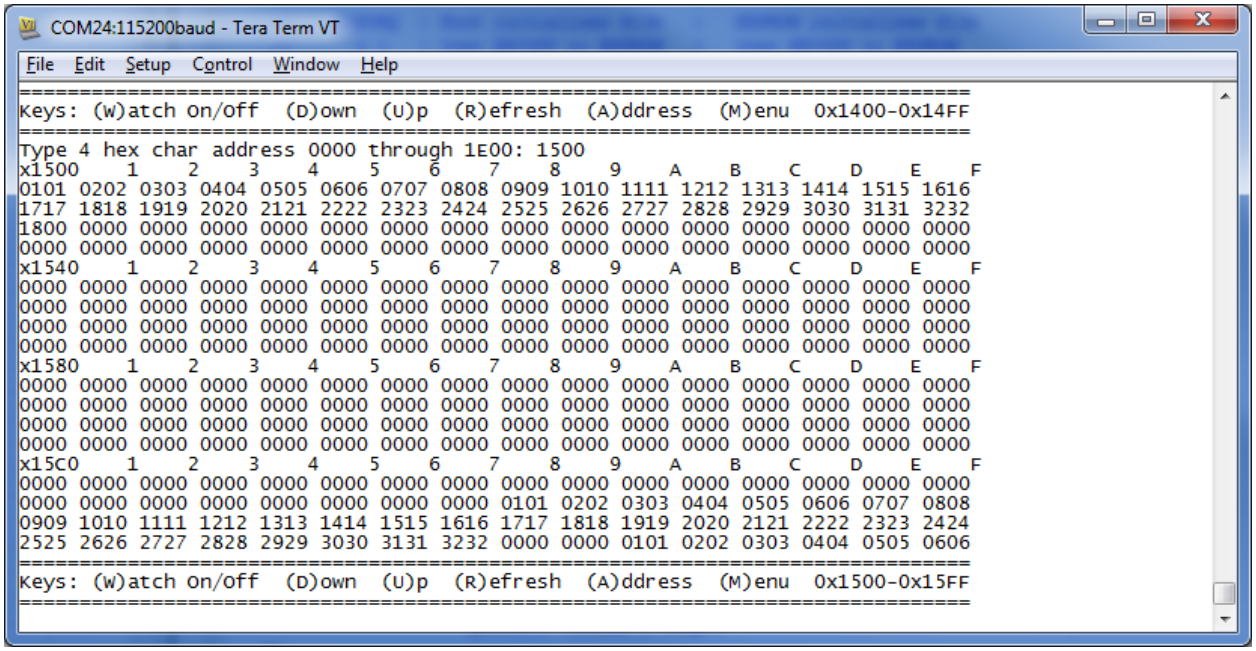
Command words	0x1400 to 0x14FF
Data words	0x1500 to 0x16FF

The above address content can be viewed using the 'w' command utility, then press 'a' for address and type 0x1400. The eight command words stored are shown in the top line.

An example of the eight commands words stored after Message #1 was transmitted is shown below for the SMT in 48 bit time tag mode:



Below are the 32 subaddress data words stored at address 0x1500 after the same Message #1 was transmitted:


COM24:115200baud - Tera Term VT
File Edit Setup Control Window Help
Keys: (W)atch On/Off (D)own (U)p (R)efresh (A)ddress (M)enu 0x1400-0x14FF
Type 4 hex char address 0000 through 1E00: 1500
x1500 1 2 3 4 5 6 7 8 9 A B C D E F
0101 0202 0303 0404 0505 0606 0707 0808 0909 1010 1111 1212 1313 1414 1515 1616
1717 1818 1919 2020 2121 2222 2323 2424 2525 2626 2727 2828 2929 3030 3131 3232
1800 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
x1540 1 2 3 4 5 6 7 8 9 A B C D E F
0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
x1580 1 2 3 4 5 6 7 8 9 A B C D E F
0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
x15C0 1 2 3 4 5 6 7 8 9 A B C D E F
0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000 0000 0000 0000 0000 0000 0101 0202 0303 0404 0505 0606 0707 0808
0909 1010 1111 1212 1313 1414 1515 1616 1717 1818 1919 2020 2121 2222 2323 2424
2525 2626 2727 2828 2929 3030 3131 3232 0000 0000 0101 0202 0303 0404 0505 0606
Keys: (W)atch On/Off (D)own (U)p (R)efresh (A)ddress (M)enu 0x1500-0x15FF

Using the C demo code

The following steps install and configure the IAR C compiler and describe how to load and modify demonstration projects using the Mamba™ Application Development Kit.

1. Using installation defaults, install IAR Embedded Workbench® for ARM (EWARM) onto your Windows computer. EWARM is a fully functional integrated development environment including project manager, editor, compiler, assembler, linker, librarian and debugger tools. It includes an optimizing C compiler, and supports a wide range of ARM devices and hardware debug systems. Ready-made device configuration files, flash loaders and example projects are included. The kit's installation CD is for the KickStart edition of IAR Embedded Workbench®. You must register at the IAR website to obtain a license file for Embedded Workbench®. The license is permanent; its only limitation is program size. If your compiled program exceeds 32K bytes, try disabling the console I/O option (in project file **`613x_initialization.h`**). This significantly reduces compiled program size without compromising MIL-STD-1553 functionality. Otherwise, a 30 day evaluation license for the unrestricted version of Embedded Workbench for ARM is available at IAR web site, www.iar.com. Refer to IAR's Released Notes in section "Important Information" for more installation information.

2. Debug requires an interface between the computer running IAR Embedded Workbench® and the Mamba™ Application Development Kit. Connect the small end of the provided USB cable to the Mamba™ evaluation board USB connector marked DEBUG. Connect the other end of the USB cable to a free computer USB port. The IAR C-SPY Debugger for ARM includes drivers for numerous target system interfaces, including built in "J-link On Board".

   The first time the evaluation board USB cable is connected to the computer, the Windows "Found New Hardware" message should appear for the J-Link device. After several seconds, Windows should load the appropriate driver and advise, "Your hardware is ready for use". If Windows fails to find the J-Link driver, direct it to look in the Drivers directory the IAR Embedded Workbench® installation CD.

   If difficulties arise when initiating a debug session at step 5, click **Project** then **Options**. In the window that opens, under **Category** = **Debugger** highlight **J-Link/J-Trace**. Click the tab labeled **Connection**, then verify Communications = USB and Interface = SWD.

3. Install IAR Embedded WorkBench® from the included IAR CD ROM. Refer to document "Holt Demo Project Installation for IAR Systems". Once installation is complete and at least one Atmel sample project is installed copy the Holt zipped projects somewhere on the PC and unzip the two projects.

4. The Holt CD provided contains an example MIL-STD-1553 project written in ANSI C. The entire project folder (or subdirectory) should be copied to a specific location on your computer's hard drive. Move the entire **_Holt** project folder into this location:
   C:\My Document\IAR embedded workbench\Atmel\atmel\at91sam3u-ek

5. Open IAR Embedded Workbench®. Click **File**, then **Open Workspace**, then navigate to the project subdirectory created in step 4. Select the project file with .eww extension, then click **Open**. (The next time Embedded Workbench® opens, this project will appear in the Recent Workspaces list when **File** is clicked.)

6. If problems occur with IAR installation or with using the IAR debugger, two Holt technical notes are provided to help resolve these issues included on the Holt CD ROM.

7. The MAMBATM project only uses unsigned integer variables. Turn off the nuisance compiler message that occurs when a variable's most significant bit toggles. The message looks like this:
   Remark[Pe068]: integer conversion resulted in a change of sign
   To disable this diagnostic message, click **Project** then click **Options**
   Category = C/C++ Compiler
   Tab = Diagnostics
   Suppress these diagnostics: add "Pe068" to list

8. The Holt IAR project includes six predefined combinations of RT, BC and MT, these can be selected from the workspace pull-down menu. The selection instructs the software to compile that particular combination for your application. The allowable combinations with depend on whether that Mode is available on your particular device. These combinations modify the preprocessor labels BC_ena, RT_ena, SMT_ena of the compiler.

9. The default combination BC_MT_RT enables the primary modes BC, MT and RT in the HI-613X device. These combinations are all flash based projects. RAM based projects are not supported due to the limited amount of RAM on the MCU. By design the Cortex™-M3 runs slower in RAM than in Flash so there is little need for a RAM based project. The six configurations and the corresponding preprocessor label values are provided in the table below:

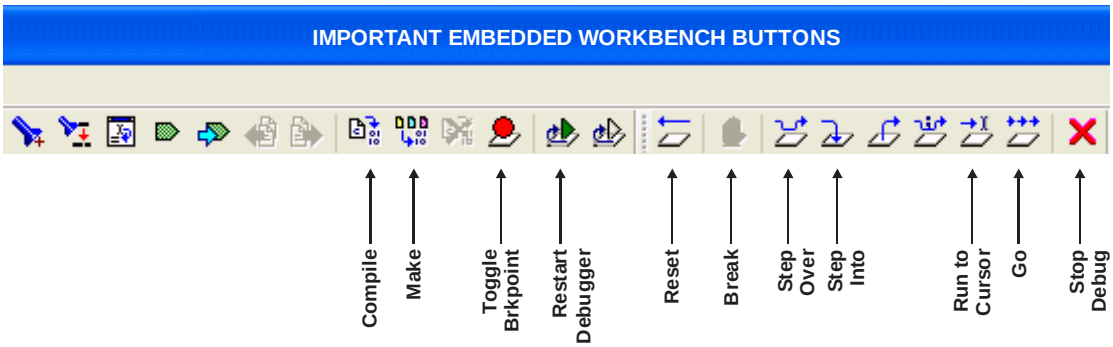
Configuration	BC_ena	RT_ena	SMT_ena	Compatibility
BC_MT_RT (default)	1	1	1	HI-6138
BC_ONLY	1	0	0	HI-6138, HI- 6137
BC_RT	1	0	0	HI-6138, HI- 6137
SMT_ONLY	0	1	1	HI-6138, HI- 6136
RT_ONLY	0	1	0	All devices
RT_SMT	0	1	1	HI-6138, HI- 6136

Other configurations are possible. 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.

11. Compile the project by clicking the **Make** button. See following illustration. If the Build messages window in IAR Embedded Workbench® indicates no errors or warnings, you can continue. If errors occurred, correct them and recompile the program.

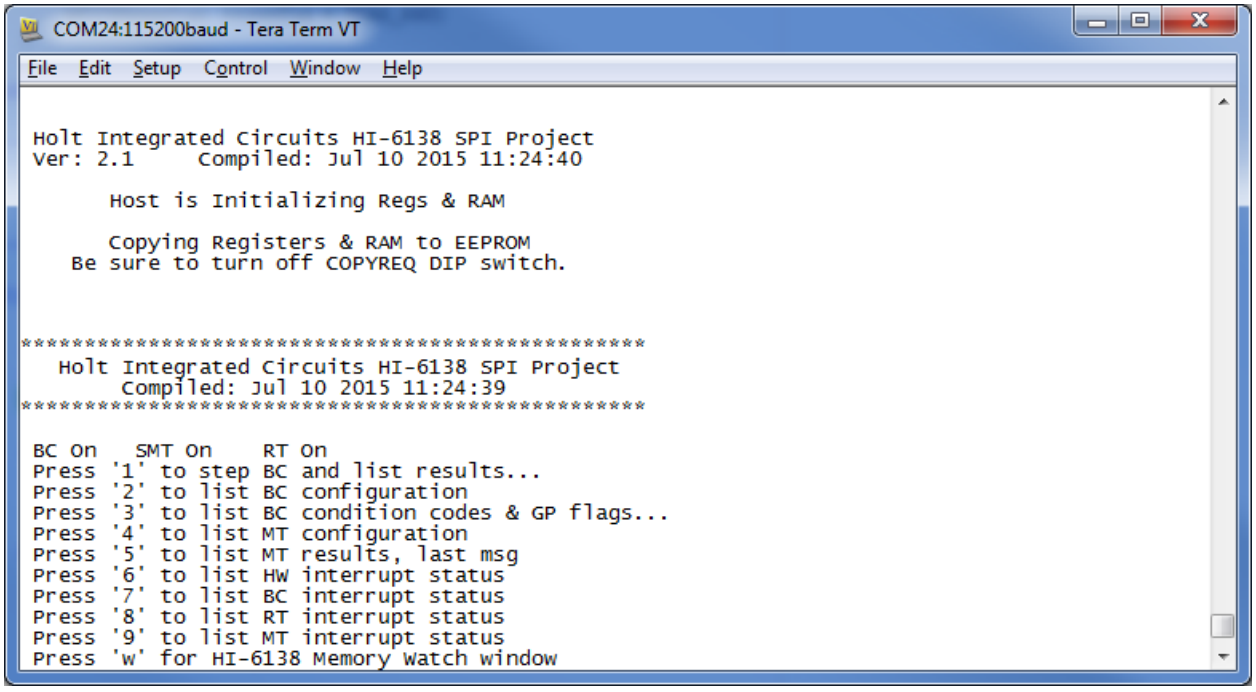
12. Initiate a debug session by clicking the **Restart Debugger** button. This downloads the compiled program into the MCU and readies the board for program execution. Click **Go** to start execution. Click **Break** (normally displayed during execution as a red upheld hand) to stop execution.

13. To observe bus activity, connect an oscilloscope to the red BUS A and red BUS B test points. The test point labeled ACTIVE is a convenient scope trigger signal. If not connected by cable to MIL-STD-1553 buses, provide a dummy loads for buses A and B by connecting a 70Ω 1 Watt resistor across each pair of red and black Bus test points.



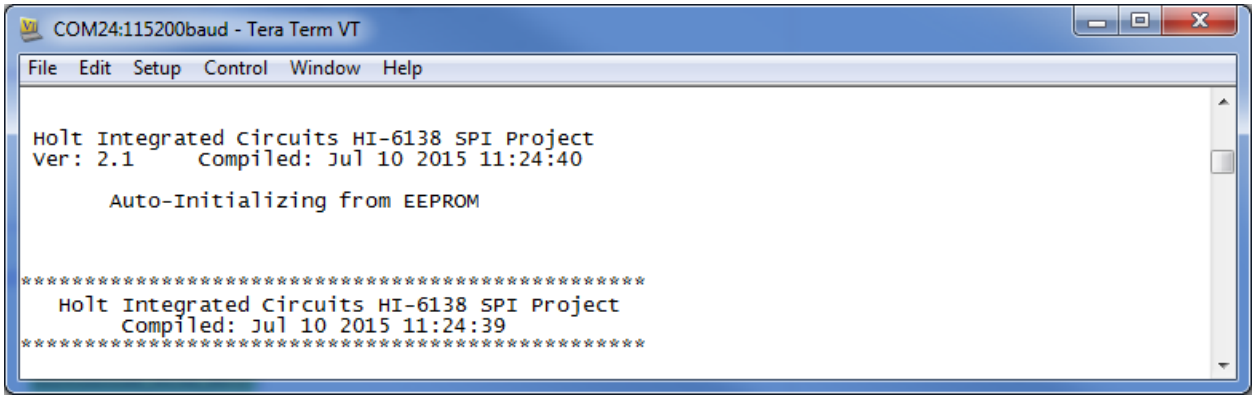
Auto-initialization from the EEPROM

1. To save a configuration to the serial EEPROM:
   - Before starting program execution, turn off the DIP switch labeled AUTOEN, directing the MCU to initialize the Mamba™ device, instead of using self-initialization from the EEPROM. Set the DIP switch labeled COPYREQ to a '1', this directs the MCU to initiate the EEPROM copy sequence after post-reset initialization of the device's registers and RAM is complete. A screen message like the one below should be displayed:



The red LED illuminates during the EEPROM copy process. If the amber LED lights then a loading error has occurred. Once the LEDs turn off, the DIP switch labeled COPYREQ should be set to a '0'.

2. Before powering back up set AUTOEN.Switch to a '1', turn off power and then turn back on, or do a master reset, the device should self-initialization from the EEPROM. A screen message like below should be displayed:



Project File List with Selected Descriptions

HEADER FILES WITHOUT CORRESPONDING C FILES

device_6138.h

ONLY USED FOR HI-6135/6/7/8 (SPI) PROJECTS

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

613x_initialization.h

Definitions for important configuration settings

613x_regs.h

Macros for register bits and bit fields

C FILES WITH CORRESPONDING HEADER FILES

Most of the function names are self-explanatory, some functions retain the names from the original 6131 they were written for, don't worry about this they work fine with the MAMBATM family.

main.c

`main();`

The primary program entry portal, `main()`demonstrates the initialization sequence used, whether or not self-initialization from EEPROM is enabled, for any combination of enabled terminals. After initialization is complete, function calls demonstrate powerful addressing methods for all RAM structures used by the enabled terminal modes.

board_613x.c

`board_613x.h contains ARM MCU i/o definitions`

`ConfigureGpio();` initializes ARM MCU general purpose I/O

```
reset_613x();
autoinit_check();
initialize_613x_shared();
init_timer();
Delay_us(num_us);
Delay_ms(num_ms);
Delay_x100ms(num);
Flash_Red_LED();
Flash_Green_LED();
error_trap(count);
enable_check();
write_init_eeprom();
```

board_6138.c

board_6138.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();
```

613x_BC.c

613x_BC.h has instruction list macros used on HI-6137/8

```
BC_bus_addressing_examples();(HI-6130 only)
initialize_bc_msg_blocks();
initialize_bc_instruction_list();
initialize_613x_BC();
bc_disable();
bc_enable();
bc_start();
bc_trigger();
bc_switch_tests();
```

For the demo, this function polls pushbutton SW1 and triggers next BC message when pressed

```
SW1_BC_Trigger();
SW2_BCtest ();
initialize_613x_BC();
```

613x_MT.c

613x_MT.h contains bus addressing structs (HI-6130 only)

```
initialize_613x_MT();
```

This function initializes either simple SMT monitor operation

613x_RT.c

613x_RT.h has descriptor table address macros used by HI-6135/6/7/8

```
initialize_613x_RT1();
RTAddr_okay(RTnum);
modify_RT_status_bits();
RTstatusUpdate();
```

This function updates RT status bits based on DIP switch settings

```
write_dummy_tx_data_RT1();
```

The last function initializes the transmit data buffers for the demo

console.c

Console functions used by all terminal modes:

```
ConfigureUsart1();
text_header();
chk_key_input();
list_hw_ints_console();
```

Console functions used by Bus Controller (BC) mode:

```
bc_last_msg_console();
list_bc_config ();
list_bc_ccgpf_reg();
list_bc_ints_console();
```

Console functions used by Remote Terminal RT1 and/or RT2:

```
list_rt_ints_console();
```

Console functions used by SMT or IMT bus monitor modes:

```
list_mt_config();
mt_last_msg_console();
list_mt_ints_console();
```

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

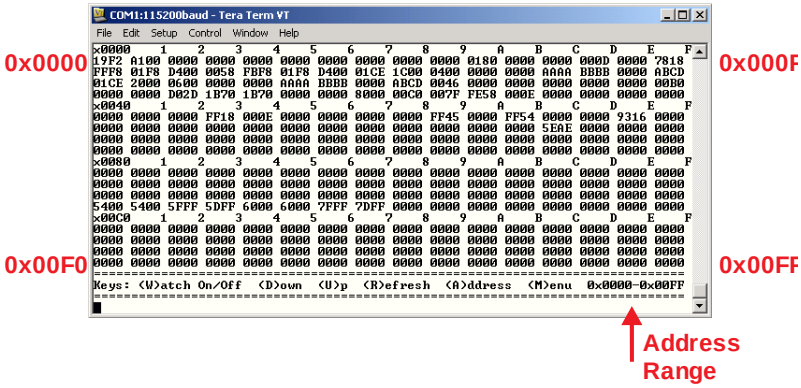
Manuak SPI write utilities

```
ascii2int();
```

Application Development Kit Notes

Mamba™ 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™ 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:



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 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 MAMBATM demonstration program covers for all devices in the Mamba™ family. The default set up has Bus Controller, Remote Terminal and Simple Monitor all enabled. Enabling or disabling any of these terminal functions is a two-step process: the software configuration in the top line of the workspace must match the hardware configuration DIP switches (BCENA and MTRUN) or a software error trap occurs.

Mamba™ SPI Interface

Mamba™ features a four wire Serial Peripheral Interface (SPI) to the host MCU or FPGA. It is offered in a plastic QFP, or 6 mm x 6 mm QFN package.

The Mamba™ 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, 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™ 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™ 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 this 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.

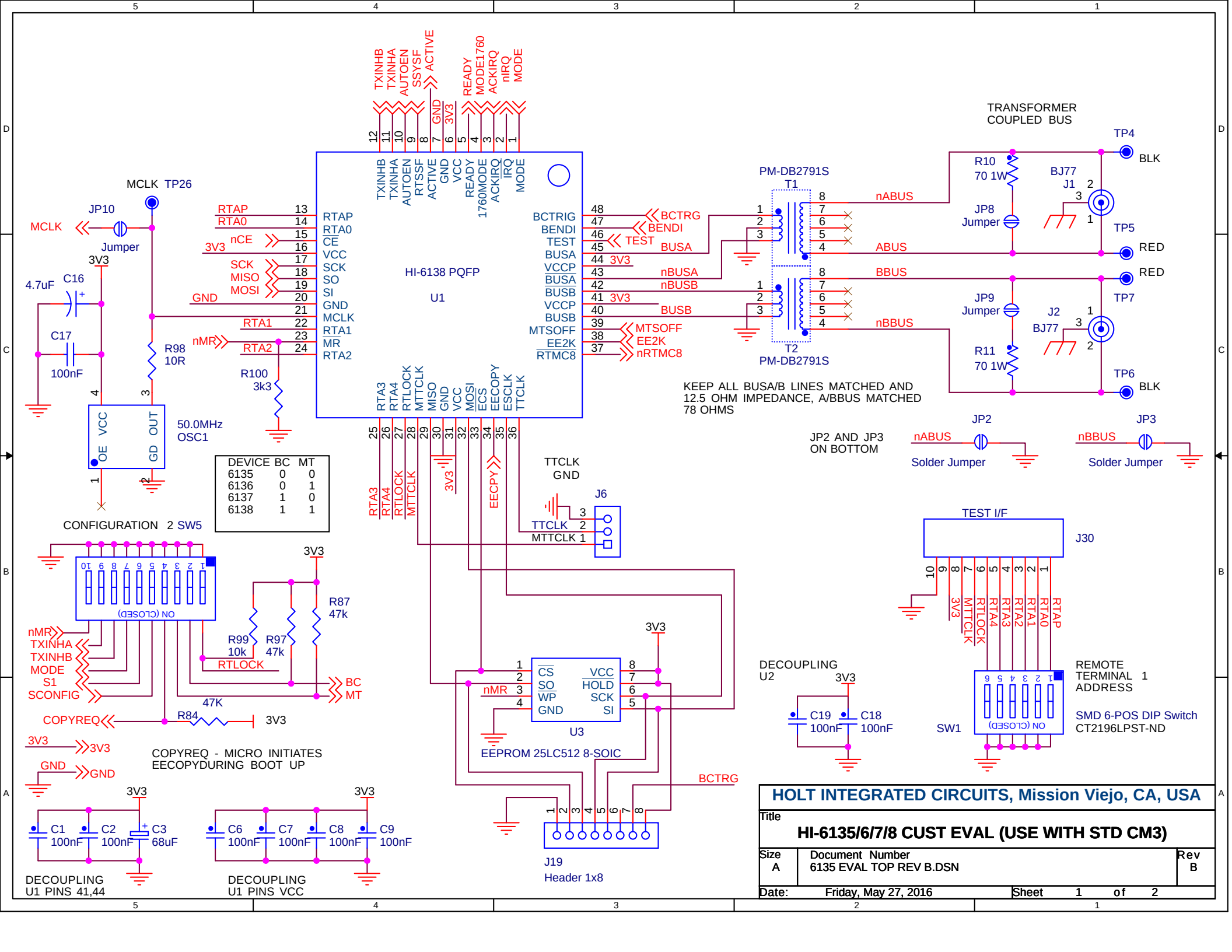
The example Mamba™ software successfully completes an interrupted multi-word sequence by using a "SPI interrupt occurred" flag, tested each time interrupts are momentarily disabled then re-enabled between SPI words. Upon detection of interrupt servicing between words, the program re-initializes the memory 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™, SPI access is incompatible with C structures for table addressing. Register/RAM inspection, watch windows are unavailable when using Mamba™. 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. Other utilities are also supplied for SPI writing to registers and memory addresses.

In summary…
With just 4 host interface signals for accessing RAM or registers, the Mamba™ SPI interface simplifies hardware design and saves board space.

Bill of Materials
Mamba Evaluation board
Rev. B

Item	Qty	Description	Reference	DigiKey	Mfr P/N
1	1	PCB, Bare, Eval Board	N/A	--------	
2	10	Capacitor, Cer 0.1uF 20% 50V Z5U 0805	C1,C2,C6,C7,C8,C9,C17,C18,C19,C20	399-1176-1-ND	Kemet C0805C104M5UACTU
3	1	Capacitor, Cer 4.7uF 10% 6.3V X5R 0805	C16	399-3134-1-ND	Kemet C0805C475K9PACTU
4	2	Capacitor 68uF 10% 6.3V Tant 400 mOhm SMD EIA 6032-28	C3,C26	495-1507-1-ND	T495C686K006ZTE400
5	2	Connector 3-Lug Concentric Triax Bayonet Jack, Panel Front Mount TRB (BJ77)	J1,J2	MilesTek 10-06570	Trompeter Electronics BJ77
6	2	Header, Male 2x20 0.1" Pitch, 0.230" Pins, 0.120" Tails	J3,J4	S2012E-20-ND	Sullins PEC20DAAN
7	2	Header, Male 2x5, 0.1" Pitch, 0.230" Pins, 0.120" Tails	J5A,J5B	S2012E-05-ND	Sullins PEC05DAAN
8	1	Header, 1x10, 0.1" pitch	J6	DO NOT STUFF	
9	1	Header, 1x8, 0.1" pitch	J19	DO NOT STUFF	
10	1	Header, 1x3, 0.1" pitch	J6	DO NOT STUFF	
11	5	Solder Jumper	JP2,JP3,JP8,JP9,JP10	DO NOT STUFF	
12	1	LED Yellow 0805	LED5	160-1175-1-ND	Lite On LTST-C170YKT
13	3	LED Green 0805	LED1 - LED3	160-1179-1-ND	LiteOn LTST-C170GKT
14	1	LED Red 0805	LED4	160-1178-1-ND	LiteOn LTST-C170EKT
15	1	Osc, 50MHz 25ppm 3.3V SMD 5x7mm	OSC1	535-10087-1-ND	Abracon ASV-50.000MHZ-E-T
16	2	Res 69.8 Ohm 1W 1% 2512 SMD	R10,R11	RHM69.8BBCT-ND	Rohm MCR100JZHF69R8
17	5	Resistor, 150 Ohm 5% 1/8W 0805	R8,R9,R12,R80,R96	P150KACT-ND	Panasonic ERJ-6GEY0R151V
18	1	Resistor, 10 Ohm 5% 1/8W 0805	R98	P10ACT-ND	Panasonic ERJ-6GEY0R100V
19	1	Resistor, 10K 5% 1/8W 0805	R99	P10KACT-ND	Panasonic ERJ-6GEYJ103V
20	5	Resistor, 47K 5% 1/8W 0805	R82,R83,R84,R87,R97	P47KACT-ND	Panasonic ERJ-6GEYJ473V
21	1	Resistor, 3.3k 5% 1/8W 0805	R100	P3.3KACT-ND	Panasonic ERJ-6GEYJ332V
22	1	DIP Switch 6-Position SMD	SW1	CT2196MST-ND	CTS 219-6MST
23	2	DIP Switch 10-Position SMD	SW4,SW5	CT21910MST-ND	CTS 219-10MST
24	2	Transformer MIL-STD-1553 Single, 1:2.50,	T1,T2	Holt PM-DB2791S	Holt / Premier Magnetics
25	3	Test Point, Red Insulator, 0.062" hole	(+)BusA, (+)BusB, 3V3	5010K-ND	Keystone 5010
26	3	Test Point, Black Insulator, 0.062" hole	(-)BusA, (-)BusB, GND	5011K-ND	Keystone 5011
27	1	Test Point, White Insulator, 0.062" hole	TP8 (Active)	5012K-ND	Keystone 5012
28	1	IC HI-6135/6/7/8 Holt 48-PQFP	U1	HOLT IC	Holt IC
29	1	IC, Serial EEPROM 512Kbit 20MHz SPI 8SOIC, Microchip	U3	25LC512-I/SN-ND	Microchip 25LC512-I/SN
30	1	Hookup Solid wire - 20AWG - Black - 4" Long per Board	For J1 and J2	C2028B-XX-ND	General Cable C2028A.12.01

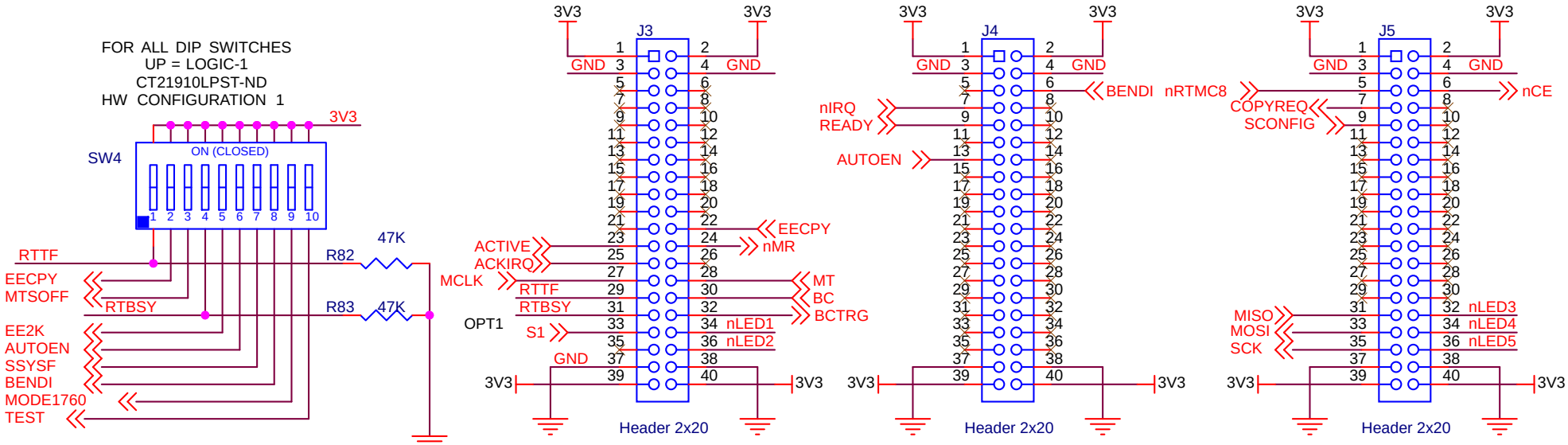


HOLT INTEGRATED CIRCUITS, Mission Viejo, CA, USA

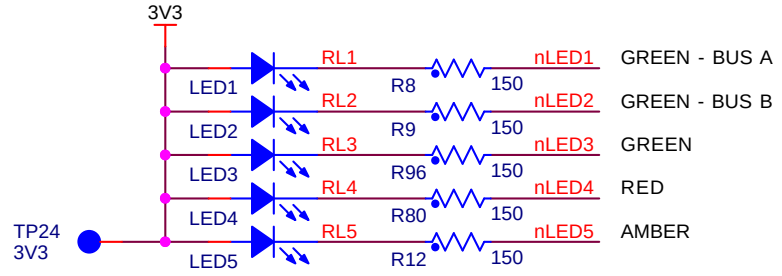
Title: **HI-6135/6/7/8 CUST EVAL (USE WITH STD CM3)**

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

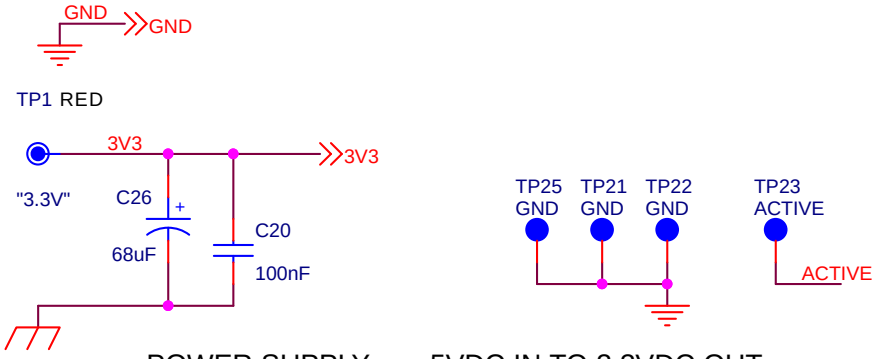
Date: Friday, May 27, 2016 Sheet 1 of 2



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

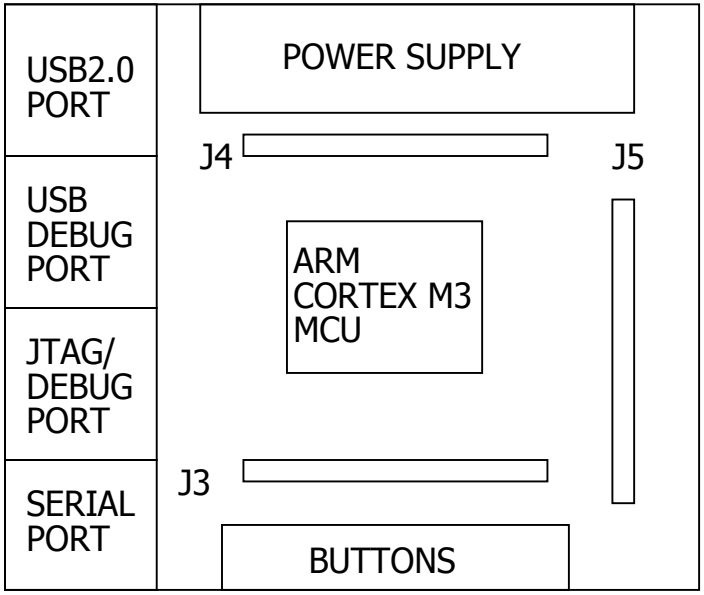


POWER SUPPLY -- 5VDC IN TO 3.3VDC OUT

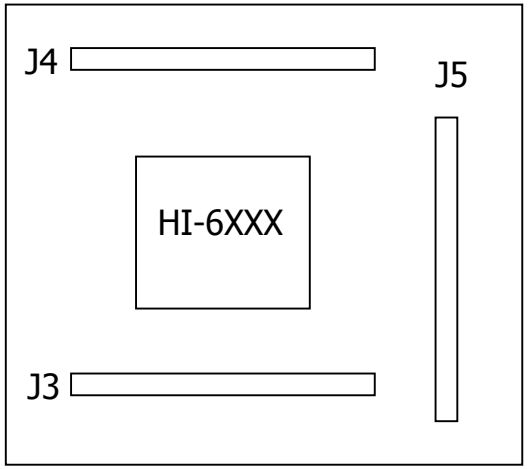
HOLT INTEGRATED CIRCUITS, Mission Viejo, CA, USA		
Title: HI-6135/6/7/8 CUST EVAL USE WITH STD CM3		
Size: A	Document Number: <Doc>	Rev: B
Date: Friday, May 27, 2016	Sheet 2 of 3	

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



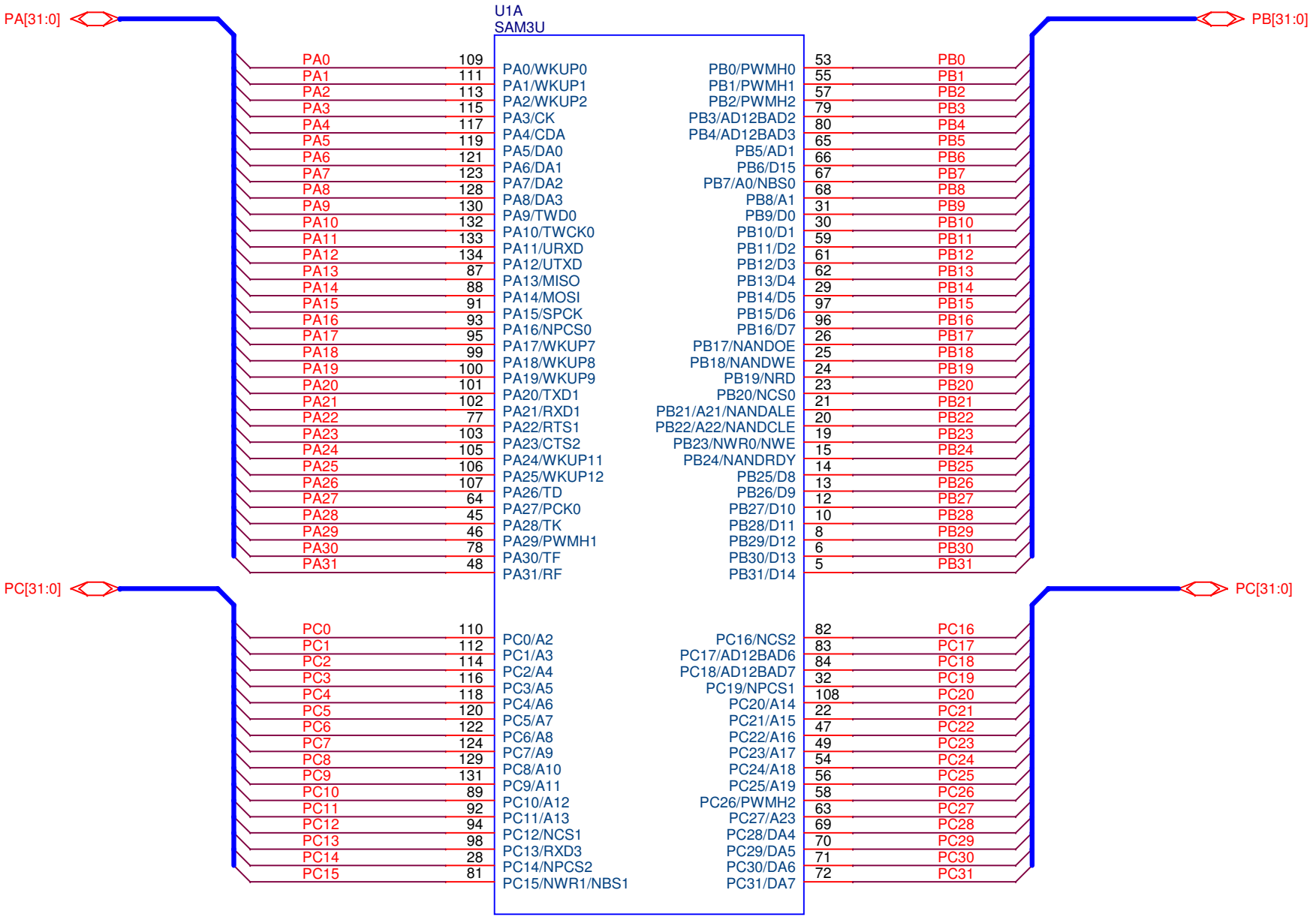
LOWER CIRCUIT BOARD



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

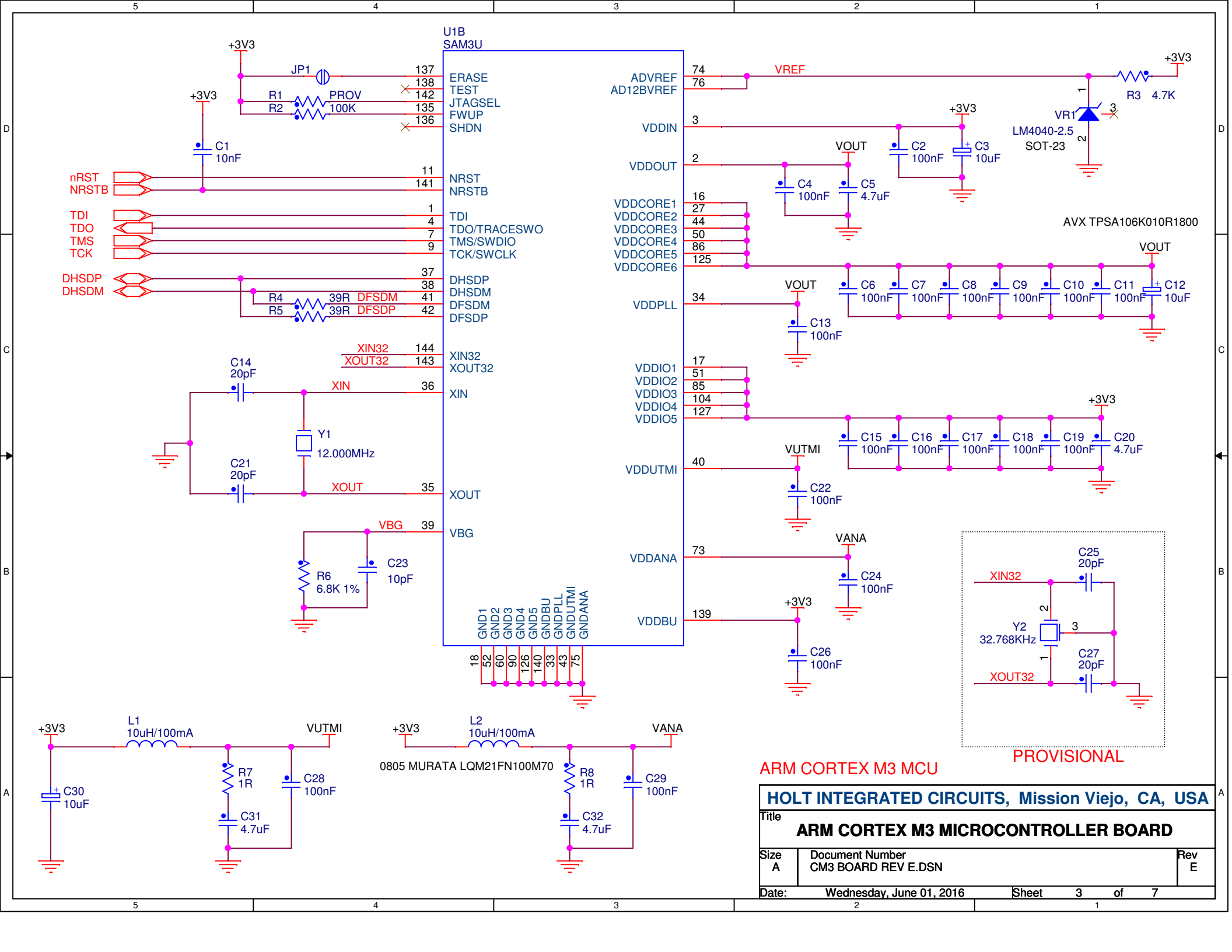
HOLT INTEGRATED CIRCUITS, Mission Viejo, CA, USA

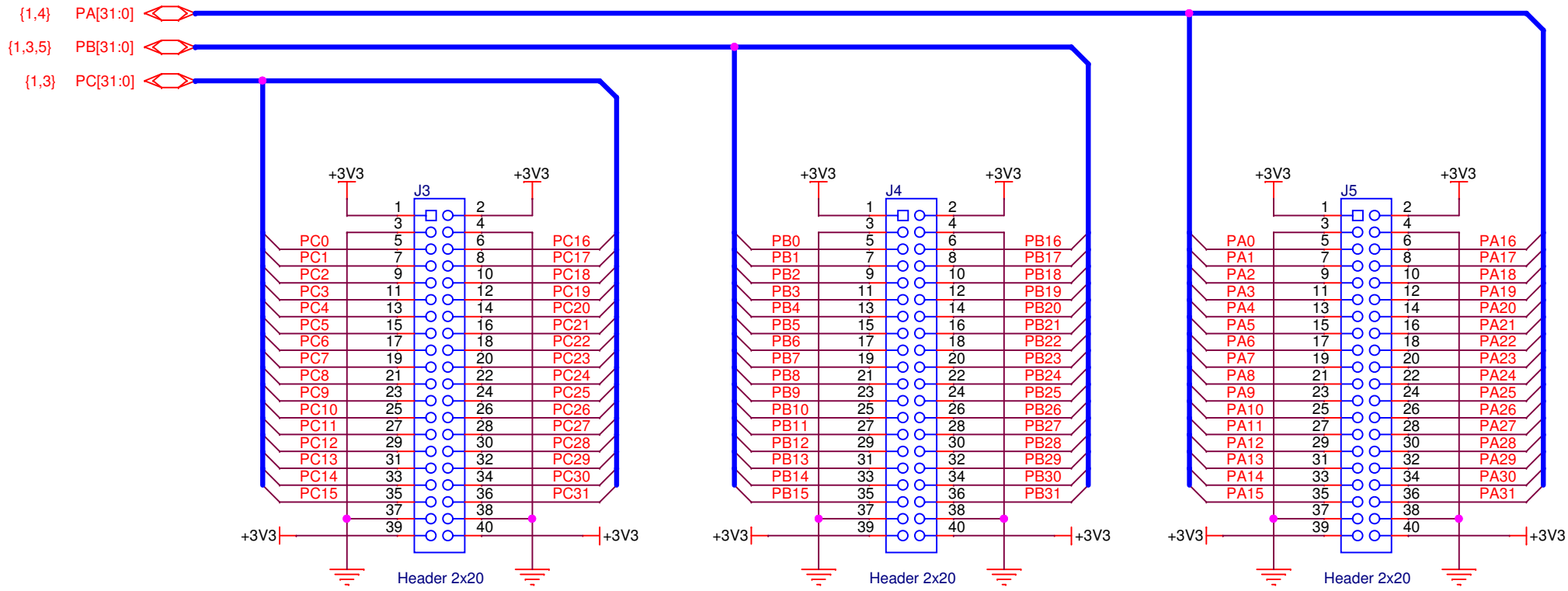
Title

ARM CORTEX M3 MICROCONTROLLER BOARD

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

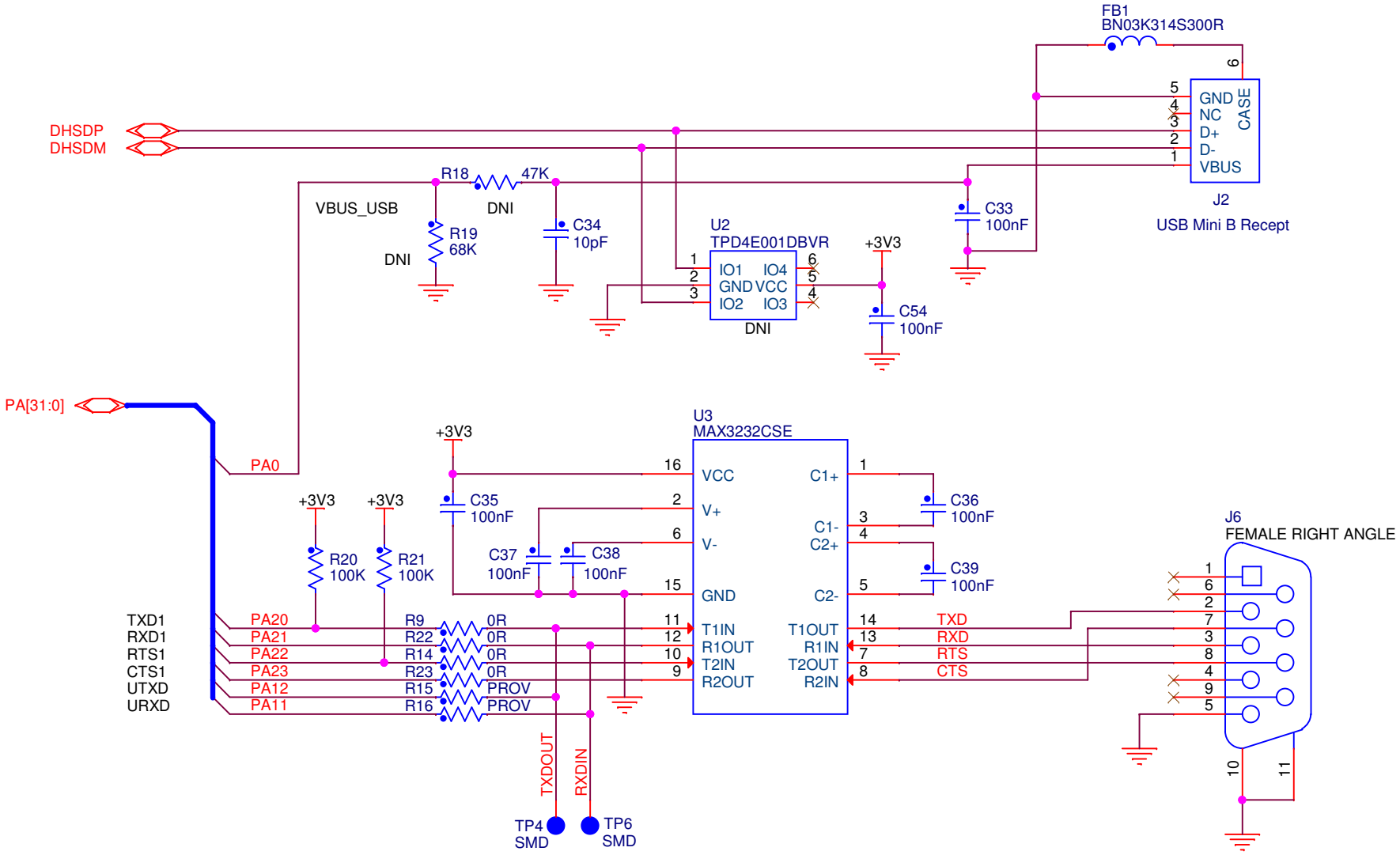
Date: Wednesday, June 01, 2016 Sheet 2 of 7


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



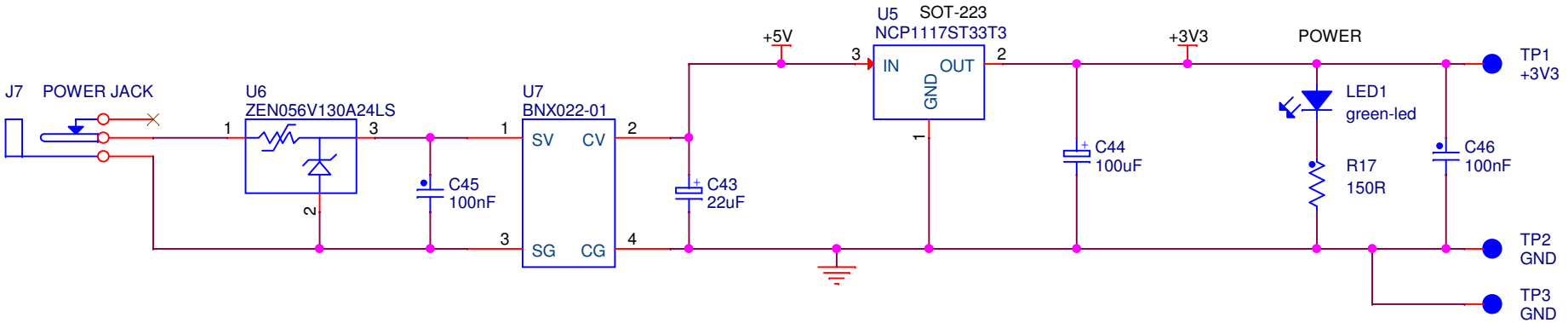
BOARD I/O HEADERS, BUTTONS

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

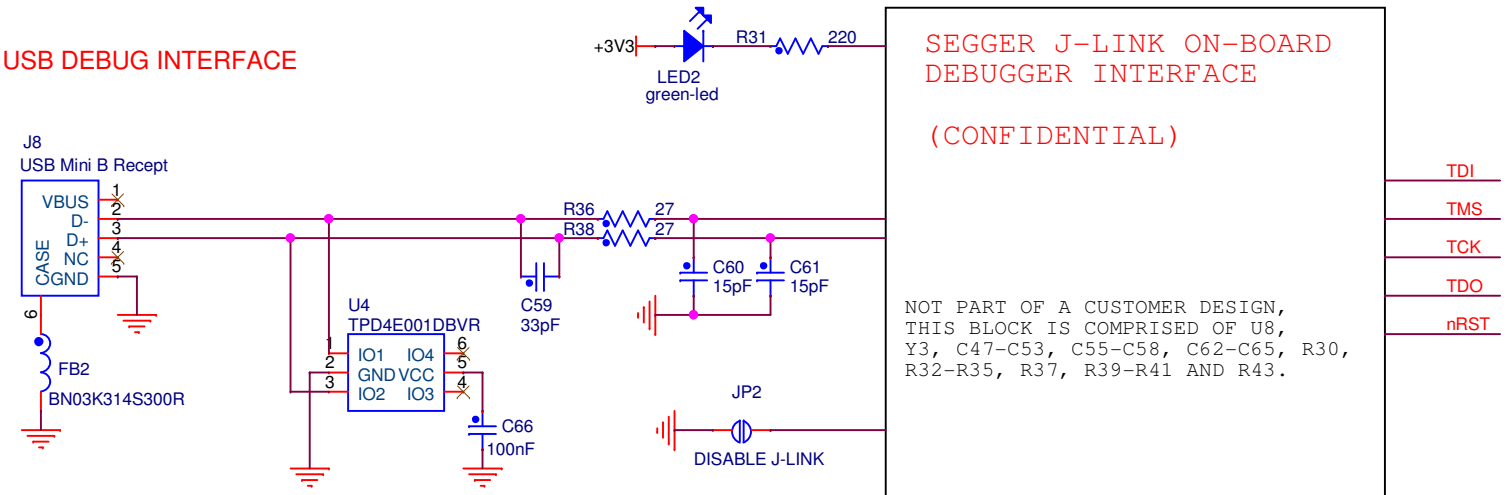
HOLT INTEGRATED CIRCUITS, Mission Viejo, CA, USA

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

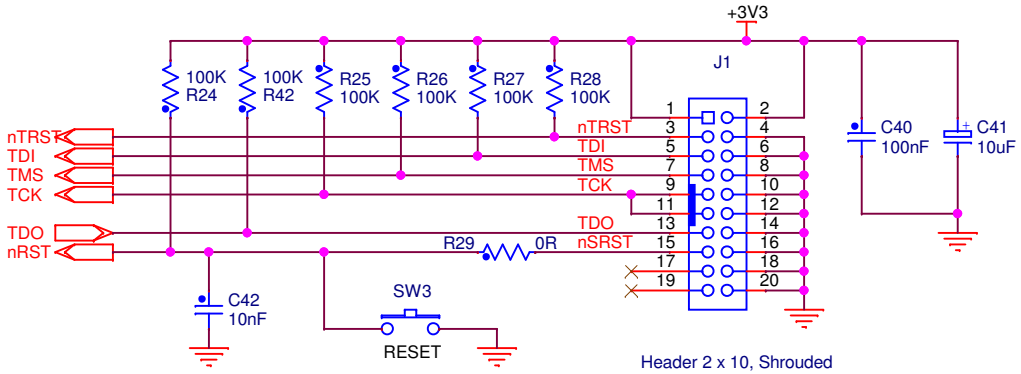
USB DEBUG INTERFACE



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	