ADK-2130: HI-2130 API Application Development Kit

April 2017

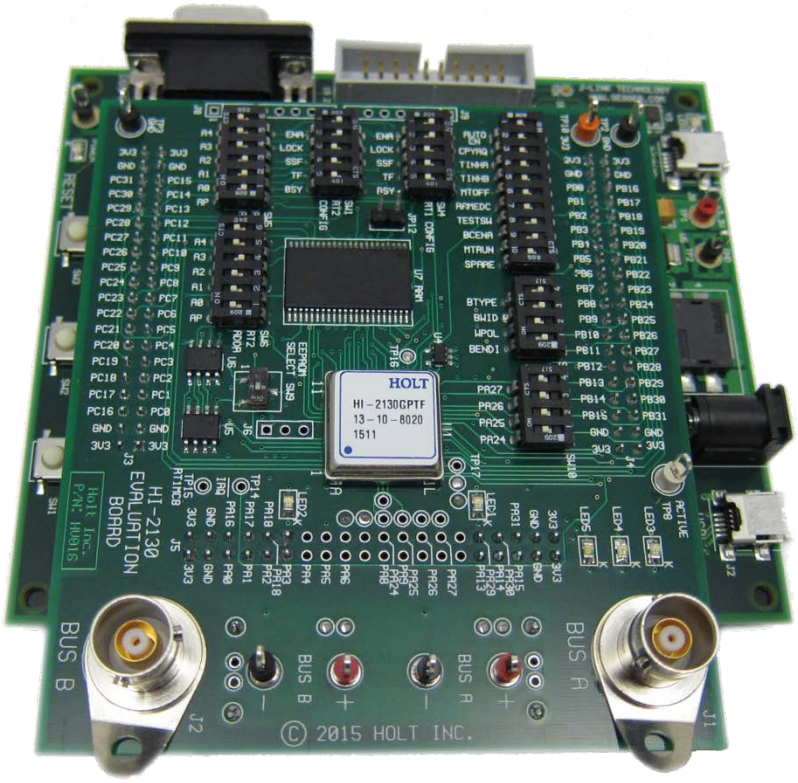
REVISION HISTORY

Revision	Date	Description of Change
AN-2130, Rev. New	02/02/16	Initial Release
Rev. A	04/05/17	Change to new document format. Update BOM. Update ARM Cortex M3 board schematic.

Introduction

The Holt HI-2130 Application Development Kit (ADK) demonstrates the broad feature set of the HI-2130 Multi Terminal IC for MIL-STD-1553. The 2-board assembly and C project reference design provide a ready-to-run evaluation platform demonstrating concurrent operation for any combination of Bus Controller, Bus Monitor and one or two Remote Terminals. For convenience, this kit includes IAR Systems *Embedded Workbench® for ARM*, and a fully integrated debug interface for the ARM Cortex M3 microcontroller.

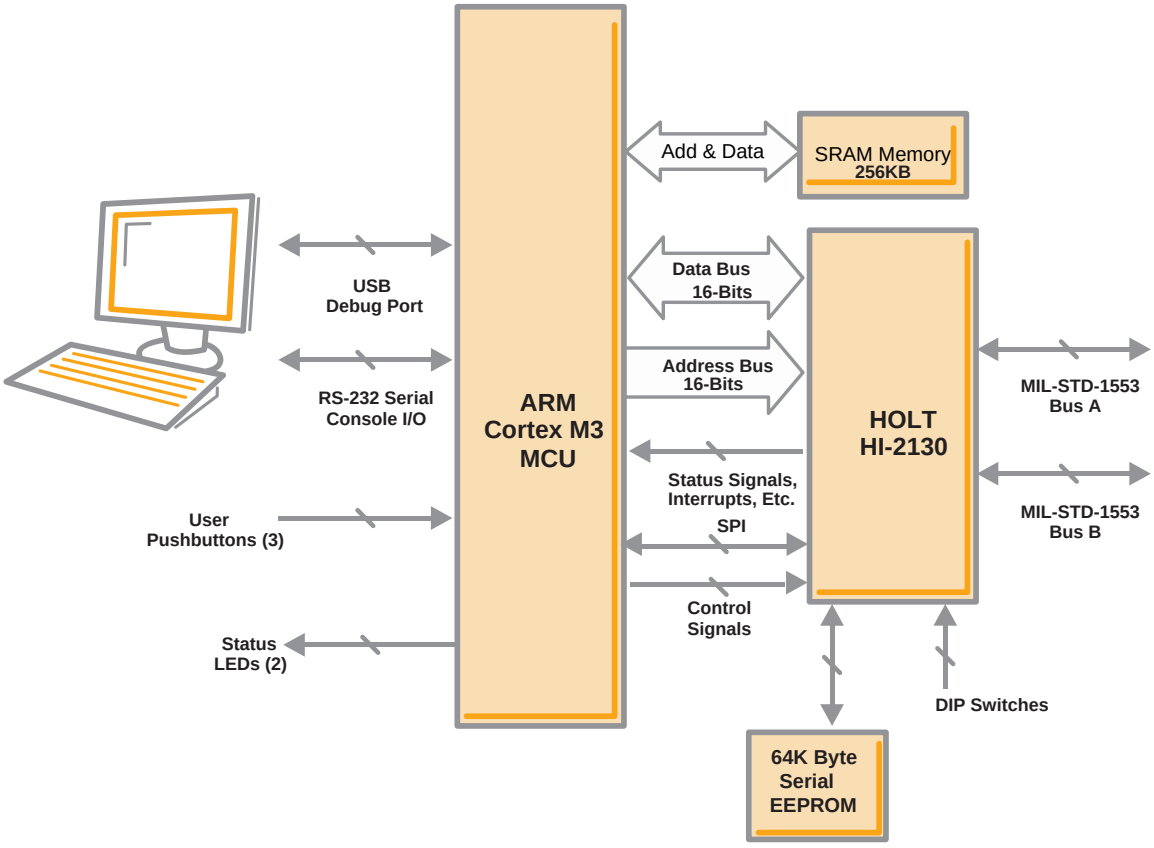
This guide describes how to set up and run the board. Additional support material and all required project software are found in the included Holt CD-ROM. A version of the demonstration software is already programmed in the microcontroller's internal flash; the board is operational right out of the box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



Evaluation Kit Contents

- This User Guide.
- Holt HI-2130 Holt Combo Project Software and Documentation CD.
- Installation CD for IAR Systems *Embedded Workbench® for ARM*, version 7.1 or greater.
- 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.
- The two board HI-2130 ADK is comprised of…
    - Upper HI-2130 board with integrated dual transformer-coupled MIL-STD-1553 bus interfaces. Numerous DIP switches configure board operation. Compared to the standard HI-6130 ADK, external 256KB static RAM memory is added for future software expansion.
    - Lower MCU board with ARM Cortex M3 16/32-bit microprocessor, integrated USB debug interface and regulated 3.3VDC power supply.

Hardware Block Diagram



Hardware Design Overview

The end of this guide provides separate schematic diagrams and bills of material for the upper and lower circuit boards. The HI-2130 is comprised of a HI-6130 protocol IC with both external 16-bit bus and SPI serial interfaces, and two internal MIL-STD-1553 isolation transformers. References in this document or in the project source code may refer to either HI-6130 or HI-2130 interchangeably. The HI-2130 data sheet shows solder ball or pin assignments and dimensions for the PGA and BGA packages; all other technical information is covered by the HI-6130 data sheet.

In contrast to the standard HI-6130 ADK, the upper board has additional 256K byte SRAM. This RAM is not utilized in the current reference project but will be utilized by future versions of the demo program for the purpose of demonstrating Holt's API's. Another difference is that MIL-STD-1553 isolation transformers are internal to the HI-2130, unlike the HI-6130 (or HI-6131) which requires external transformers on the PCB.

The HI-2130 upper daughter board can be separated from the provided ARM MCU board for connection to an alternate user-supplied microprocessor or FPGA board. The inter-board headers are located on 0.1" (2.54 mm) grid for compatibility with generic prototyping boards. All host interface signals go through the inter-board headers. Numerous HI-2130 configuration and Remote Terminal address setting pins are directly controlled by DIP switches on the upper HI-2130 board; these signals are not available to the MCU on the inter-board headers.

The lower ARM Cortex M3 board is based on the flash-programmable Atmel AT91SAM3U-EK microprocessor. Two MCU-to-1553 interfaces are provided on the HI-2130 ADK: a 16-bit parallel bus interface and SPI interface. A typical design uses just one of these interfaces although the HI-2130 interface can be switched "on the fly". The demonstration program is configurable for 16-bit parallel bus, serial SPI or both. See the section under EBI debugging in SPI mode, later in this document. 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, SW1 and SW2 are available for software interaction on the lower ARM MCU board. A RESET SW3 pushbutton resets the ARM microprocessor, which in turn controls the HI-2130 Master Reset signal. On the (upper) daughter card, SW10 provides four spare DIP switches that are available to the user.

There are five LEDs on the daughter card that show status. At power up, they momentarily flash sequentially to demonstrate operability. The green Bus A and Bus B LEDs flash when a message is received on the corresponding bus, if the "RT Traffic" program feature is enabled from the console 'T' command. This option is OFF by default. LEDs 3, 4 and 5 are reused by different sections of the program. The green and red LEDs show AUTOEN/EECOPY request status. See section on AUTOEN/EECOPY operation, later in this document. The green LED turns ON briefly during the message interrupt handler, which is a convenient place to measure the interrupt service time by probing the right side of the LED with a scope probe. The red LED pulses low during the Delay_ms() function. These LEDs are easily repurposed.

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 standard USB cable for connecting the board’s debug interface to your computer. (For users already owning an ARM debug interface with ribbon-cable connector, an ARM-standard 2x10 debug connector provides debug connectivity. In this case, jumper JP2 on the bottom of the lower board should be soldered closed to disable “J-Link On Board”.)

The ACTIVE (TP8) test point is asserted by the HI-2130 after initialization while a 1553 message is being transacted by an enabled BC and/or RT, or optionally recorded when the SMT monitor is running. ACTIVE is a useful scope trigger signal that is asserted at start-of-message and resets after end-of-message register/interrupt updates.

The four SPI host interface signals SCK, MISO, MOSI and slave select nCE may be monitored on J9, an 8-pin header for probing or hooking up a logic analyzer. When starting a new design using an unfamiliar processor or FPGA, a logic analyzer is highly recommended to help verify SPI timing. A USB logic analyzer such as the Saleae *Logic Pro 16* works well for this purpose. The SPI nCE pin (used as SPI slave select) is directly controlled by a GPIO pin in the SPI low-level driver code. This is required because nCE must be continuously asserted low for the full duration of opcode and N number of data word transfers without interruption.

External Bus or SPI Host-1553 Interface

There are two ways to interface the MCU to the HI-2130: 16-bit parallel interface or 4-signal SPI port. The maximum SCK clock rate on the host SPI port is 20MHz. In applications with high 1553 bus utilization or requiring fastest host access to transacted message data, the 16-bit parallel bus interface is recommended. This demo program demonstrates both methods by setting the value for C macro `HOST_BUS_INTERFACE` appropriately in the `613x_initialization.h` header file, then recompiling the program and downloading it to the board. Follow the Quick Start Guide below before installing IAR or the demo code. The demo board will be programmed with the SPI version.

Because SPI clocks data into and out of the HI-2130 serially (and because SPI uses op codes and a memory address pointer to define SPI transactions), SPI data transfers are slower than comparable parallel bus (EBI) transfers. A further disadvantage of SPI is that the IAR *Embedded Workbench* debugger cannot display HI-2130 register and RAM memory contents in a standard debug “memory” or “memory Watch” window. When the HI-2130 is used in EBI mode, it appears like SRAM; the debugger can easily examine memory-mapped registers and memory data structures using a “memory ” window. For SPI-only operation, the serial console provides a convenient debugging tool. Strategic placement of `printf()` statements are used in the program to print out values on the console. To mitigate SPI register/memory display limitations when debugging, special preprogrammed console commands are provided which display important registers and control or data blocks via the console. Customized `printf()` sequences can be implemented by the user.

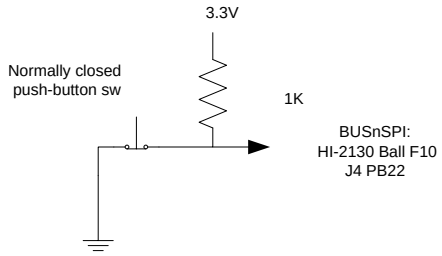
Optional EBI parallel debugging technique for SPI designs

For SPI host interface designs, if an EBI host parallel bus is temporarily available during software development, it is possible to use the bus interface to refresh a debugger “memory” or “memory Watch” window on demand when program execution is stopped:

Assume the SPI host interface is used by default. When execution is stopped and the programmer wishes to refresh a “memory ” window (or open a new one), the BUSnSPI pin is driven high temporarily while a single line of program code is single-stepped. This action causes the IAR debugger to access registers or memory, and refresh the debug “memory ” window.

For this to work, a working 16-bit address and data bus must be connected to the HI-2130. The macro `BOTH_EBI_SPI` must be set to 1 and macro `HOST_BUS_INTERFACE` must be set to 0 in the `613x_initialization.h` header file before compiling. This initializes the EBI bus and SPI port, and makes the BUSnSPI pin an active input instead of an output of the Atmel MCU GPIO.

A pull-up resistor (1K) should be connected to the HI-2130 BUSnSPI pin (solder ball F10) from J4 PB22 to 3.3V. The HI-2130 has an internal weak pull-down resistor (~50KΩ) so a 1K will be sufficient to pull up the signal when the push-button switch is opened.



When debugger “memory” window refresh is needed, stop program execution or use a breakpoint that is not within a SPI transaction sequence. Temporarily open the switch (pin goes high) then single-step one or more lines of code with the debugger. The debug “memory” window is refreshed after each single-step. The BUSnSPI switch is then closed (going low) before resuming full speed (RUN) execution using SPI.

When debug is complete, be sure to reset C macro `BOTH_EBI_SPI` to 0 and macro `HOST_BUS_INTERFACE` to 0 in the `613x_initialization.h` header file before compiling.

See more examples of this technique later.

A Quick Demonstration

The Holt HI-2130 Application Development Kit is pre-programmed to concurrently operate as a Bus Controller, SMT Bus Monitor and two independent Remote Terminals (RT and RT2). Terminal addresses for the two RTs are preset using DIP switches, before applying power. For demo purposes, the RT1 DIP switch should be set to RT address 3 (0-0-0-1-1-parity 1), RT2 should be set to RT address 4 (0-0-1-0-0-

parity 0). These values work with the demonstration program's preprogrammed Bus Controller message repertoire. The two 6-position DIP switches should already be set with these address values, plus odd parity. The user's guide, source code and software documentation sometimes refer to RT1 as just RT. All other DIP switches are set to defaults positions shown in the board photo.

1. Follow this quick demonstration of the demo board **before** installing IAR Systems *Embedded Workbench® for ARM* (EWARM) and Holt demo project folders to ensure the board operates stand-alone. The demo program is already programmed in flash memory in the Atmel ARM Cortex M3 processor so it can be demonstrated right out of the box. Install IAR EWARM and the Holt demo projects later, **when instructed to do so**.

2. The demonstration program uses the console (serial port to the PC) to provide a command menu and display message traffic information. Use console I/O with a computer serial (COM) port and a "terminal emulation" program like *TeraTerm*. Most desktop computers have a COM port; many notebook computers do not have a COM port; these require a serial-to-USB adapter (not provided by the Holt ADK). When the HI-2130 is designed with the SPI interface, the console is very useful as a debugging aid since debuggers cannot directly access HI-2130 registers or RAM for display in a "memory" window.

   Windows 7 is recommended. The installation instructions refer to *Windows* 7 directory names.

   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
Using the provided DB-9 serial cable, connect the MCU board to the computer serial (COM) port.

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

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

5. To observe bus activity on BUS A, connect an oscilloscope to red test point labeled BUS A + and connect the scope ground to the BUS A -. To observe bus activity on BUS B connect another probe to the same points on BUS B. The ACTIVE test point is a convenient scope trigger and it goes high at start of message and goes low at end of message.

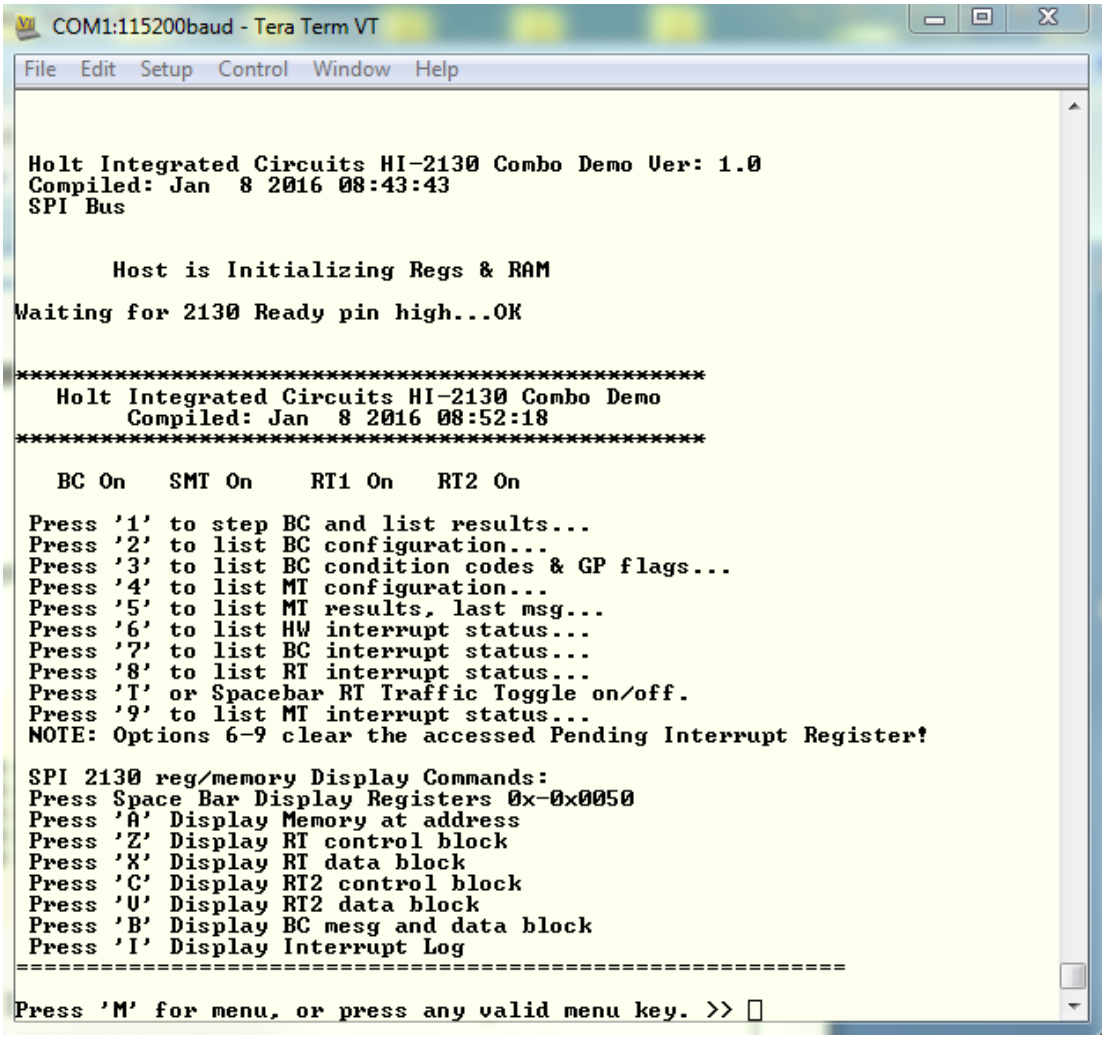
6. If not connected by cable to actual MIL-STD-1553 buses, provide resistive dummy loads for buses A and B by connecting a 75 Ω 1/2 Watt (or any value between 70-80 Ω) resistor across each pair of red and black Bus test points. (For this demonstration, half-Watt resistors are adequate because transmit duty cycle is sufficiently low. When using the on-chip HI-2130 to generate BC messages directed to on-chip RTs, use external 75 Ω resistor loads. When using a bus coupler to connect to actual MIL-STD-1553 buses, do not use the 75 Ω dummy load resistors.

7. Set SW8 configuration DIP switches labeled AUTOEN (sw1) and COPYREQ (sw2) off (down position).

8. Plug-in the provided 5V DC power supply and connect the cable to the power input jack on the lower circuit board. At power up, the LEDs will flash on sequentially to demonstrate operability.

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

Dates and times shown below will differ from your console screen.



Press space bar on the PC keyboard to display the HI-2130 system registers. Commands can be entered upper or lower case.

```
 2130 Registers:
          0    1    2    3    4    5    6    7    8    9    A    B    C    D    E    F
Adr 0000: 19F0 8100 0000 0000 0000 0000 0000 0000 0000 0000 0180 527A 0000 000D 0000 7818
Adr 0010: FFF8 0000 D458 0058 FBF8 0000 D5A8 00EF 1E00 0400 0000 0000 AAAA BBBB 0000 ABCD
Adr 0020: 01CE 2200 0600 0000 0000 AAAA BBBB 0000 ABCD 0800 0000 0000 0000 0000 0000 00B0
Adr 0030: 0000 0000 D02D 1B70 1B70 0000 0000 8000 00C0 000F 0000 0000 0000 0000 0000 0000
Adr 0040: 0000 0000 0000 C092 004B 0000 0000 0000 0000 C0BF 0000 C0CD 0000 0000 0000 0000
Adr 0050: 0000
>>
```

Register 0 (0000) is the Master Control register and Register 1 is the Master Status and Reset Register. Notice the MSB is high (8) in word address 0x0001, this is the READY bit indicating the HI-2130 is ready for host access. Widen the Tera-Term window enough to view the full 16 registers across for best viewing.

Press ‘M’ again to view the menu of available commands.

When the program is configured for SPI mode, an additional set of commands are available that display important HI-2130 control block and data block memory sections. These commands are Z, X, C, V, B and I. Command A prompts for a 16-bit address then displays 256 words starting at that address.

Essential to RT operation are the control blocks for RT messages and corresponding data buffers.

Press Z to view RT1’s control block memory. This shows the beginning of the initialized RT1 control block structures. Subaddress SA1 control block has four words starting at address 0x0404.

Address 0x0404 = 0x4104 – SA1 control word (shows IWA interrupt, ping-pong buffer mode)
Address 0x0405 = 0x0800 – Buffer A pointer address
Address 0x0406 = 0x0822 – Buffer B pointer address
Address 0x0407 = 0x0844 - Broadcast Data pointer address

Review the HI-2130 data for a complete description of these data structures.

```
 RT1 Control Blocks
          0    1    2    3    4    5    6    7    8    9    A    B    C    D    E    F
Adr 0400: D0AD DEAD DEAD DEAD 4104 0800 0822 0844 C000 0A00 0000 0D10 8001 1176 1176 15B6
Adr 0410: 8042 1E00 1E00 1C00 8000 0C00 0005 1A36 0000 1A36 0000 1A36 0000 1A36 0000 1A36
Adr 0420: 0000 1A36 0000 1A36 0000 1A36 0000 1A36 0000 1A36 0000 1A36 0000 1A36 0000 1A36
Adr 0430: 0000 1A36 0000 1A36 0000 1A36 0000 1A36 0000 1A36 0000 1A36 0000 1A36 0000 1A36
Adr 0440: 0000 1A36 0000 1A36 0000 1A36 0000 1A36 0000 1A36 0000 1A36 0000 1A36 0000 1A36
Adr 0450: 0000 1A36 0000 1A36 0000 1A36 0000 1A36 0000 1A36 0000 1A36 0000 1A36 0000 1A36
Adr 0460: 0000 1A36 0000 1A36 0000 1A36 0000 1A36 0000 1A36 0000 1A36 0000 1A36 0000 1A36
Adr 0470: 0000 1A36 0000 1A36 0000 1A36 0000 1A36 C000 08AE 0000 08AE D0AD DEAD DEAD DEAD
          0    1    2    3    4    5    6    7    8    9    A    B    C    D    E    F
Adr 0480: D0AD DEAD DEAD DEAD 4000 0866 0000 08AA 8000 0D32 0020 1172 8001 15D6 15D6 1A16
Adr 0490: 8052 1E00 1E00 1C00 0000 1A58 0000 1A58 0000 1A58 0000 1A58 0000 1A58 0000 1A58
Adr 04A0: 0000 1A58 0000 1A58 0000 1A58 0000 1A58 0000 1A58 0000 1A58 0000 1A58 0000 1A58
Adr 04B0: 0000 1A58 0000 1A58 0000 1A58 0000 1A58 0000 1A58 0000 1A58 0000 1A58 0000 1A58
Adr 04C0: 0000 1A58 0000 1A58 0000 1A58 0000 1A58 0000 1A58 0000 1A58 0000 1A58 0000 1A58
Adr 04D0: 0000 1A58 0000 1A58 0000 1A58 0000 1A58 0000 1A58 0000 1A58 0000 1A58 0000 1A58
Adr 04E0: 0000 1A58 0000 1A58 0000 1A58 0000 1A58 0000 1A58 0000 1A58 0000 1A58 0000 1A58
Adr 04F0: 0000 1A58 0000 1A58 0000 1A58 0000 1A58 C000 08AE 0000 08AE D0AD DEAD DEAD DEAD

 RT1 MC blocks
          0    1    2    3    4    5    6    7    8    9    A    B    C    D    E    F
Adr 0500: 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
Adr 0510: 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
Adr 0520: 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
Adr 0530: 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
Adr 0540: 0000 0000 0000 0000 4000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
Adr 0550: 4000 0000 0000 0000 4000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
Adr 0560: 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
Adr 0570: 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 0580: 4000 0000 0000 0000 4000 0000 0000 0000 4000 0000 0000 0000 4000 0000 0000 0000
Adr 0590: 4000 0000 0000 0000 4000 0000 0000 0000 4000 0000 0000 0000 4000 0000 0000 0000
Adr 05A0: 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
Adr 05B0: 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
Adr 05C0: 4000 0000 0000 0000 0000 0000 0000 0000 4000 0000 0000 0000 4000 0000 0000 0000
Adr 05D0: 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
Adr 05E0: 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
Adr 05F0: 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
================================================================================
Keys:   (D)own  (U)p  (R)efresh  (A)ddress  (M)enu  0x0500-0x05FF
================================================================================

>>
```

A sub menu at the bottom allows moving up or down in memory space.

Press X to display RT1 data buffer area:

```
 RT1 Data blocks
            0    1    2    3    4    5    6    7    8    9    A    B    C    D    E    F
Adr 0800: 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
Adr 0810: 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
Adr 0820: 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
Adr 0830: 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
Adr 0840: 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
Adr 0850: 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
Adr 0860: 0000 0000 0000 0000 0000 0000 0000 0000 0101 0202 0303 0404 0505 0606 0707 0808
Adr 0870: 0909 1010 1111 1212 1313 1414 1515 1616 1717 1818 1919 2020 2121 2222 2323 2424
            0    1    2    3    4    5    6    7    8    9    A    B    C    D    E    F
Adr 0880: 2525 2626 2727 2828 2929 3030 3131 3232 0000 0000 F001 F002 F003 F004 F005 F006
Adr 0890: F007 F008 F009 F00A F00B F00C F00D F00E F00F F010 F011 F012 F013 F014 F015 F016
Adr 08A0: F017 F018 F019 F01A F01B F01C F01D F01E F01F F020 0000 0000 0000 0000 0000 0000
Adr 08B0: 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
Adr 08C0: 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
Adr 08D0: 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
Adr 08E0: 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
Adr 08F0: 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 0900: 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
Adr 0910: 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
Adr 0920: 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
Adr 0930: 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
Adr 0940: 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
Adr 0950: 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
Adr 0960: 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
Adr 0970: 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 0980: 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
Adr 0990: 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
Adr 09A0: 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
Adr 09B0: 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
Adr 09C0: 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
Adr 09D0: 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
Adr 09E0: 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
Adr 09F0: 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
================================================================================
Keys:   (D)own  (U)p  (R)efresh  (A)ddress  (M)enu  0x0900-0x09FF
================================================================================
```

General structure of demo functions

The Holt HI-2130 Combo Demo program calls functions in main.c to initialize the terminals and display the console menu before entering a loop waiting for console commands. By default, all terminals are initialized and enabled so MIL-STD 1553 messages are transacted automatically.

The BC portion of the demo is configured to transmit from a 10 message list, one message is transmitted each time the '1' key is pressed. After the tenth message is transacted, the message list repeats. The following messages are used:

(See file 613x_bc . c for the array structures showing how these messages are constructed)

```
Msg Block 1  = Subaddress Rx   Command 03-1-30-00 (loopback subaddress) Bus A
Msg Block 2  = Subaddress Tx   Command 03-1-30-00 (loopback subaddress) Bus B
Msg Block 3  = Subaddress Rx   Command 03-0-01-00 (SA1)                 Bus A
Msg Block 4  = Subaddress BRx  Command 31-0-05-11                       Bus B
Msg Block 5  = Subaddress BRx  Command 31-0-30-00 (loopback subaddress) Bus A
Msg Block 6  = Mode Code  Tx   Command 03-1-31-02 (tx mode code 2)      Bus B
Msg Block 7  = Mode Code  Tx   Command 03-1-31-18 (tx mode code 18)     Bus A
Msg Block 8  = Mode Code  Rx   Command 03-0-31-21 (rx mode code 21)     Bus B
RT-RT Msg Block 1 = RT-RT msg  Commands 04-0-30-02 03-1-05-02           Bus A
RT-RT Msg Block 2 = BRT-RT msg Commands 31-0-30-11 04-1-05-11           Bus B
```

A simple demonstration uses the internal BC to transmit these messages. See the results in RT buffer memory or use the 'T' command to display RT traffic on the console. All that is needed is to connect a 75Ω 1/2 Watt resistor across the BUSA and nBUSA test points to terminate the bus. If a tri-axial bus terminator is available, plug that into the BUSA cable jack (and optionally BUSB).

Demo steps:

Connect termination resistor(s).
Press T to enable traffic console display.
Press 1 to command the BC to transmit a message and see the results on the console.

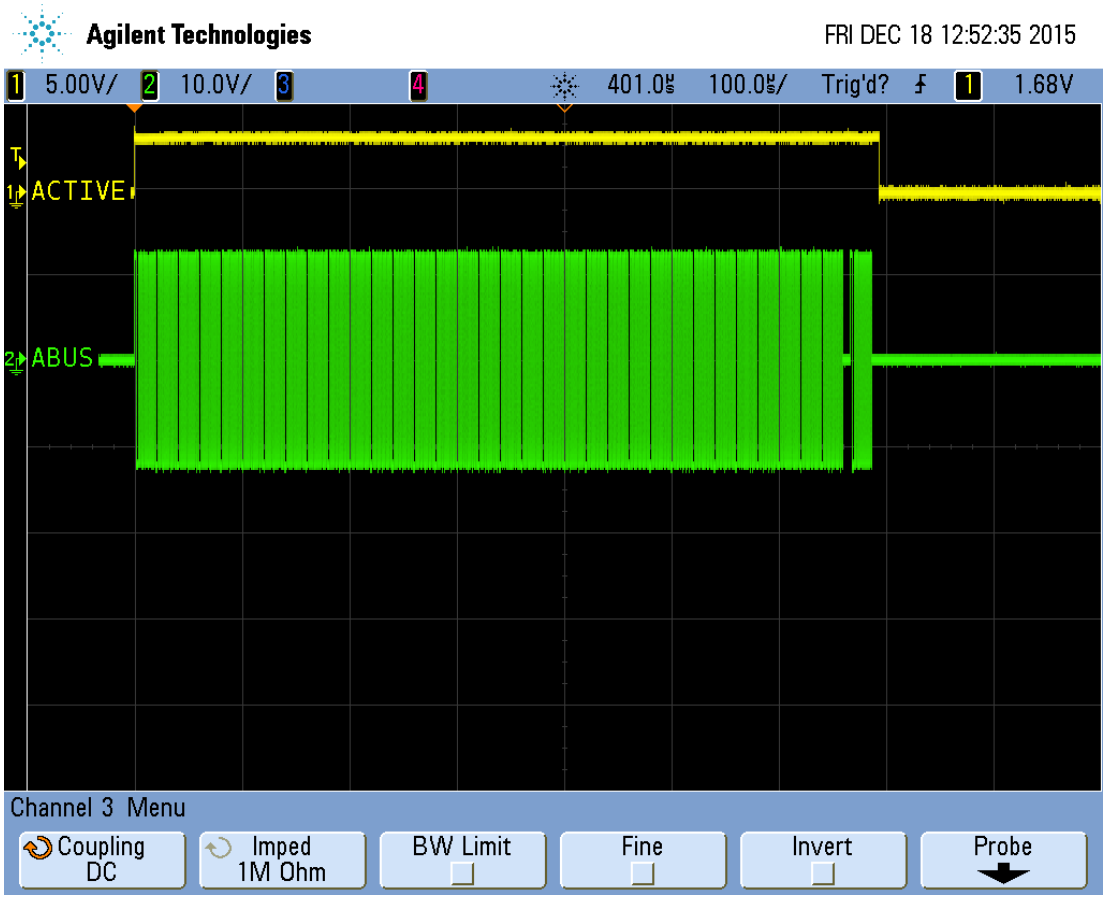
There are two portions of this display. The top section shows BC message details and the last section starting with "MSG #0001" is the data the RT message interrupt captures. The new message is detected in the main loop and displayed.

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

Results From Last Message Issued by BC
============================================================
BC Message # 1
Message Type: Rx Subaddress Command, 32 data words

CW: 0x1BC0 = 03-0-30-00     SW: 0x1800 = RT03 CS

BC Control Word: 0x4180
MEmask RetryEna UseBusA NonBcstSA

Block Status Word: 0x8000
EOM  Bus A

Condition Code Register: 0x8000
BC Running: No Condition Codes or Gen Purpose Flags Are Set.

Data Addr: 0x8000,
0x3022 0x0202 0x0303 0x0404 0x0505 0x0606 0x0707 0x0808
0x0909 0x1010 0x1111 0x1212 0x1313 0x1414 0x1515 0x1616
0x1717 0x1818 0x1919 0x2020 0x2121 0x2222 0x2323 0x2424
0x2525 0x2626 0x2727 0x2828 0x2929 0x3030 0x3131 0x3232

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

MSG #0001.  TIME = 00000000us    Bus A
            CMD 0x1BC0 --> 03-R-30-00    BC to RT
            MIW 0x0000
            DATA 3022  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
```

Using the T command is useful for viewing RT message reception and works well when using the internal BC to transmit messages by pressing the "1" key. When the T command is used to enable the console traffic, the Bus A or Bus B green LED will flash when a message is detected, saved and displayed.

Message #1 appears as shown below on an oscilloscope. See step #5 above for scope setup. Notice the small gap before the last word. This is the RT status word being transmitted back to the BC. Because a terminator is used on the bus connector, the voltage level when the BC transmits is the same as when the RT transmits its response. When an external BC tester is used to transmit commands through a MIL-STD 1553 bus coupler, the BC and RT signal amplitudes will be different. The BC signal amplitude

(originating off board) will be approximately 25% of the amplitude of the local RT board (where the scope is connected) due to normal end-to-end impedance transformation and attenuation across current limiting resistors in the bus coupler.



Some of the BC internal messages will not appear on the ABUS because they are transacted on BBUS. To see those messages, put the scope probe on Bus B.

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

When using a BC to initiate 1553 messages through a properly-terminated external bus to an RT or SMT on the HI-2130 ADK board, remove the external 75 Ω resistors (if present) and connect the BC test equipment to the demo board circular tri-axial bus jacks using MIL-STD-1553 cables and bus couplers. If a bus coupler is not readily available, connect BC tester directly to the demo board tri-axial jack, but in this case use the 75 Ω termination resistor.

Construct a BC test message set to RT address 3, RX message, SA1, and any number of data words.

Transmit the message to the demo board (ensure the T command was used to enable the console traffic) and the console should display a message similar to this:

```
MSG #0001.  TIME = 00000000us    Bus A
            CMD 0x1820 --> 03-R-01-00    BC to RT
            MIW 0x0000
            DATA 0003  0201  0202  0203  0204  0205  0206  0207
                 0208  0209  020A  020B  020C  020D  020E  020F
                 0210  0211  0212  0213  0214  0215  0216  0217
                 0218  0219  021A  021B  021C  021D  021E  021F
```

Here's the console display when an external BC transmits Mode Code 19:

```
MSG #0003.  TIME = 00124328us    Bus A
            CMD 0x1C13 --> 03-T-00-19   RT to BC
            MIW 0x0013
            Mode Code 19
            Mode Data 0xABCD
```

Subaddress 30 is typically reserved for 1553 data wrap-around (loop back). This is accomplished by setting the data pointers for SA30 Tx messages and SA30 Rx messages to the same memory address. When a SA30 receive command is transacted, the RT stores received data in the allocated buffer area. When that RT later receives a SA30 transmit command, it responds by transmitting status with the same data previously received from the BC. Thus the received and transmitted SA30 data should match.

```
MSG #0005.  TIME = 00124328us    Bus A
            CMD 0x1BC0 --> 03-R-30-00    BC to RT
            MIW 0x0013
            DATA 3001  1001  2002  3003  4004  5005  6006  7007
                 8008  9009  A00A  B00B  C00C  D00D  E00E  F00F
                 0010  1011  2012  3013  4014  5015  6016  7017
                 8018  9019  A01A  B01B  C01C  D01D  E01E  F01F

MSG #0006.  TIME = 00124328us    Bus A
            CMD 0x1FC0 --> 03-T-30-00   RT to BC
            MIW 0x0013
            DATA 3001  1001  2002  3003  4004  5005  6006  7007
                 8008  9009  A00A  B00B  C00C  D00D  E00E  F00F
                 0010  1011  2012  3013  4014  5015  6016  7017
                 8018  9019  A01A  B01B  C01C  D01D  E01E  F01F
```

When transacting 1553 messages with high repetition rate, decoded RT traffic displayed by the console may not keep pace due to limitations of the slow console 115,200 baud rate. Depending on the message

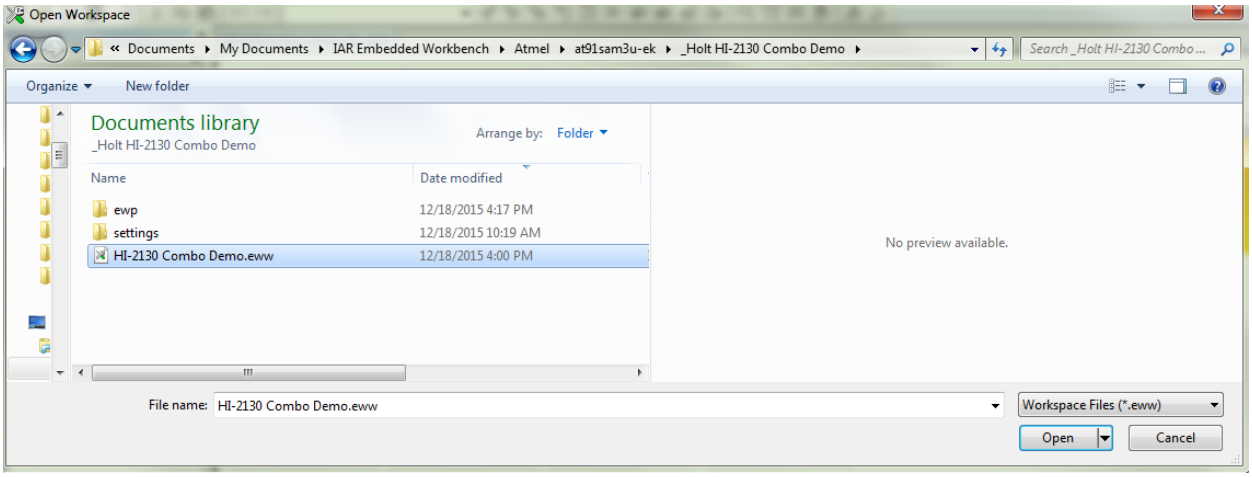
size and repetition rate, some messages may not show on the console. Rest assured that the HI-2130 RT (or SMT monitor) is properly transacting (or recording) all valid message data. Some messages are simply not being decoded and shown on the console.

This concludes the out of box demonstration. Proceed to the next section to learn how to install IAR Systems *Embedded Workbench® for ARM* development tools and the Holt demonstration software. This will enable modifications to the software.

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

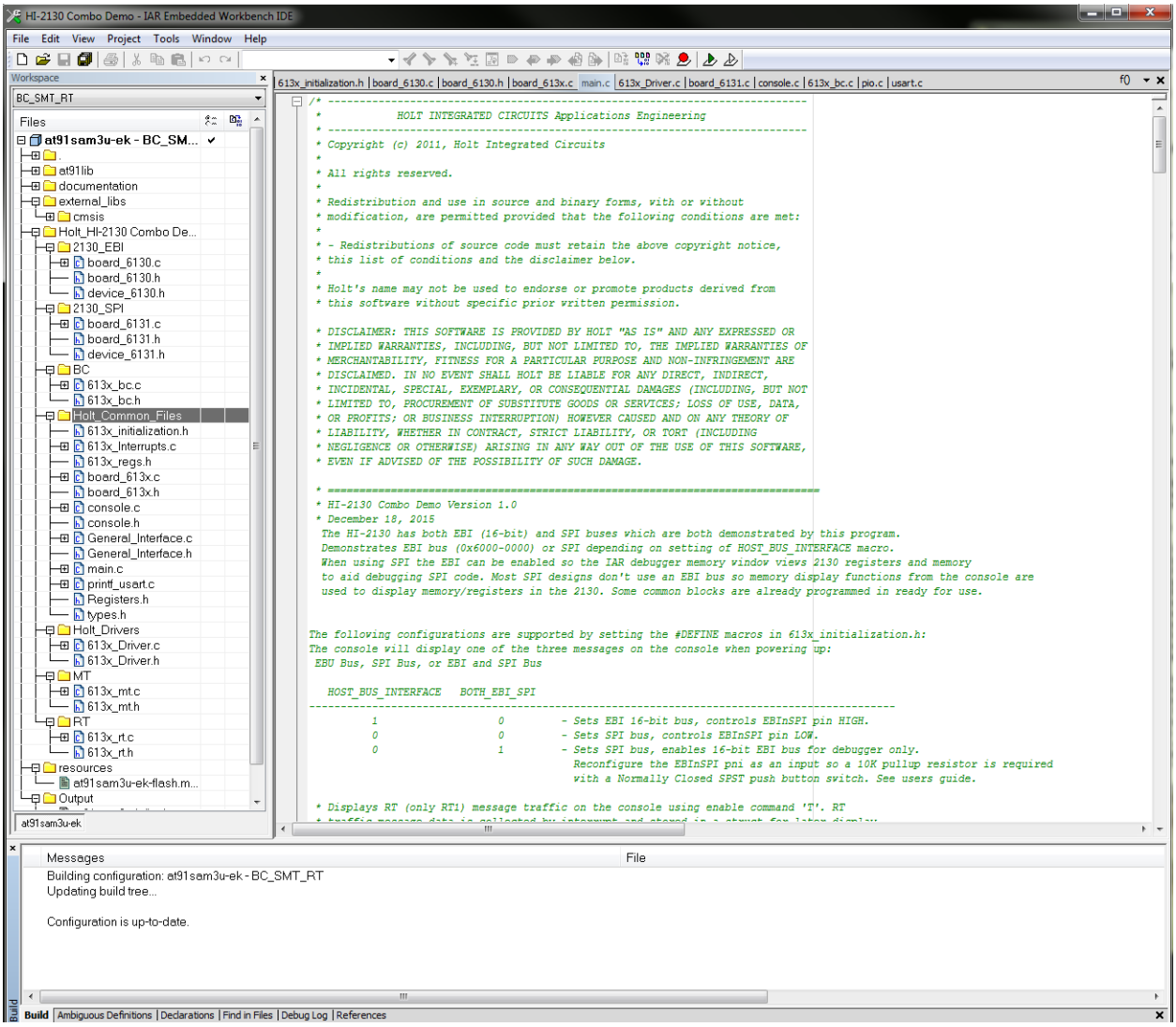
1. BEFORE copying the Holt demo project to your computer, you must install the IAR Systems compiler, *Embedded Workbench for ARM* (*EWARM*). The install sequence is critical to ensure that Atmel ARM library files and Holt demo project folder are created in their proper locations. Follow the *Holt HI-2130 Combo Demo Project Installation Guide* found in the Project folder on the Holt CD-ROM. Before proceeding to the next step, IAR EWARM must be installed and the Holt project folder must be in its proper location 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-2130 Combo Demo Project from the IAR File pull-down menu, click on File/Open/Workspace and navigate to the project folder location and select “HI-2130 Combo Demo.eww” and click the Open button.



3. A 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 in the project Workspace hierarchy panel on the left side. The source file should open in the EWARM text editor pane, similar to this…



There are pre-configured 1553 terminal selections (IAR project configurations) chosen by a pull-down menu above the Files directory tree; the default is BC_RT_MT shown at the top. Some configurations exclude file groups and files from the compiler/linker build. For example if the “BC Only” configuration is selected, folders and files not needed are dimmed indicating the project will not use those files on the next build. Modifying or creating new configurations is

easy from the Project pull-down menu. See IAR IDE project documentation from the Help menu for more information on project configurations. The default is shown below.

5. Debug requires an interface between the computer running IAR *Embedded Workbench®* and the HI-2130 Application Development Kit. Connect the small end of the provided USB cable to the HI-2130 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". Before the debug cable is plugged in, Debugger LED2 should flash repeatedly until the cable is connected to the PC. If LED2 does not flash see the Holt Technical Note – IAR EWARM Debugger Troubleshooting guide included in the project folder.

   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 11, 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.

6. Optionally turn off the nuisance compiler message that occurs when a variable's most significant bit toggles. Some users prefer to see all warnings so in this case nothing is required. Some of the Atmel board files produces these warnings, the message looks like this:

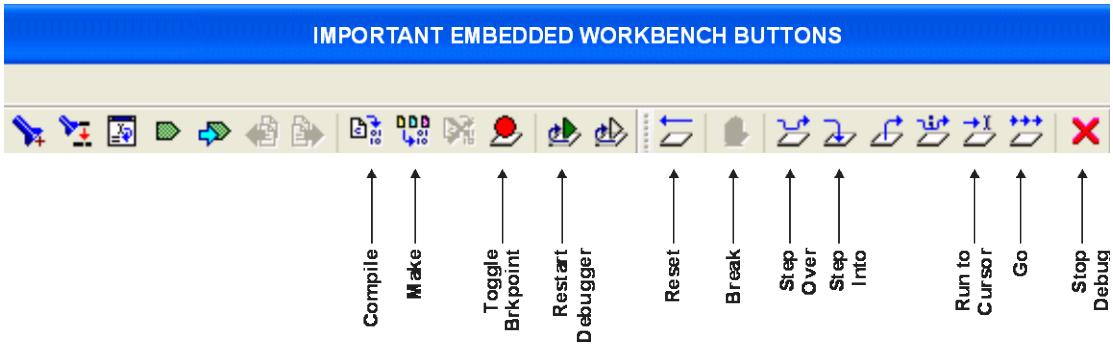
   `Remark[Pe068]: integer conversion resulted in a change of sign`
   To optionally disable this diagnostic message, click **Project** then click **Options**
   Category = C/C++ Compiler
   Tab = Diagnostics
   `Suppress these diagnostics:` add "Pe068" to list

7. The default configuration BC_SMT_RT enables the primary modes BC, MT, RT and RT2. This is the default configuration programmed into the demo board. This enables and demonstrates the Bus Controller, RT (RT1) and RT2 Remote Terminals and a Monitor. These configurations are all flash based projects. RAM based projects are not supported due to the limited amount of RAM

on the MCU. By design the ARM Cortex-M3 runs slower in RAM than in Flash so there is little need for a RAM based project.

8. Compile the project by clicking the **Make (or Rebuild All)** button. See following illustration. If the Build messages bottom pane in IAR *Embedded Workbench®* indicates no errors or warnings, you can continue. If errors occurred, correct them and recompile the program. Sometimes a "clean" build should be performed from the Project pull down window if strange compiler or linker issues occur.

9. 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.



When execution starts the program menu will be displayed on the console and the LEDs will flash sequentially that was described in the quick start section

10. 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.

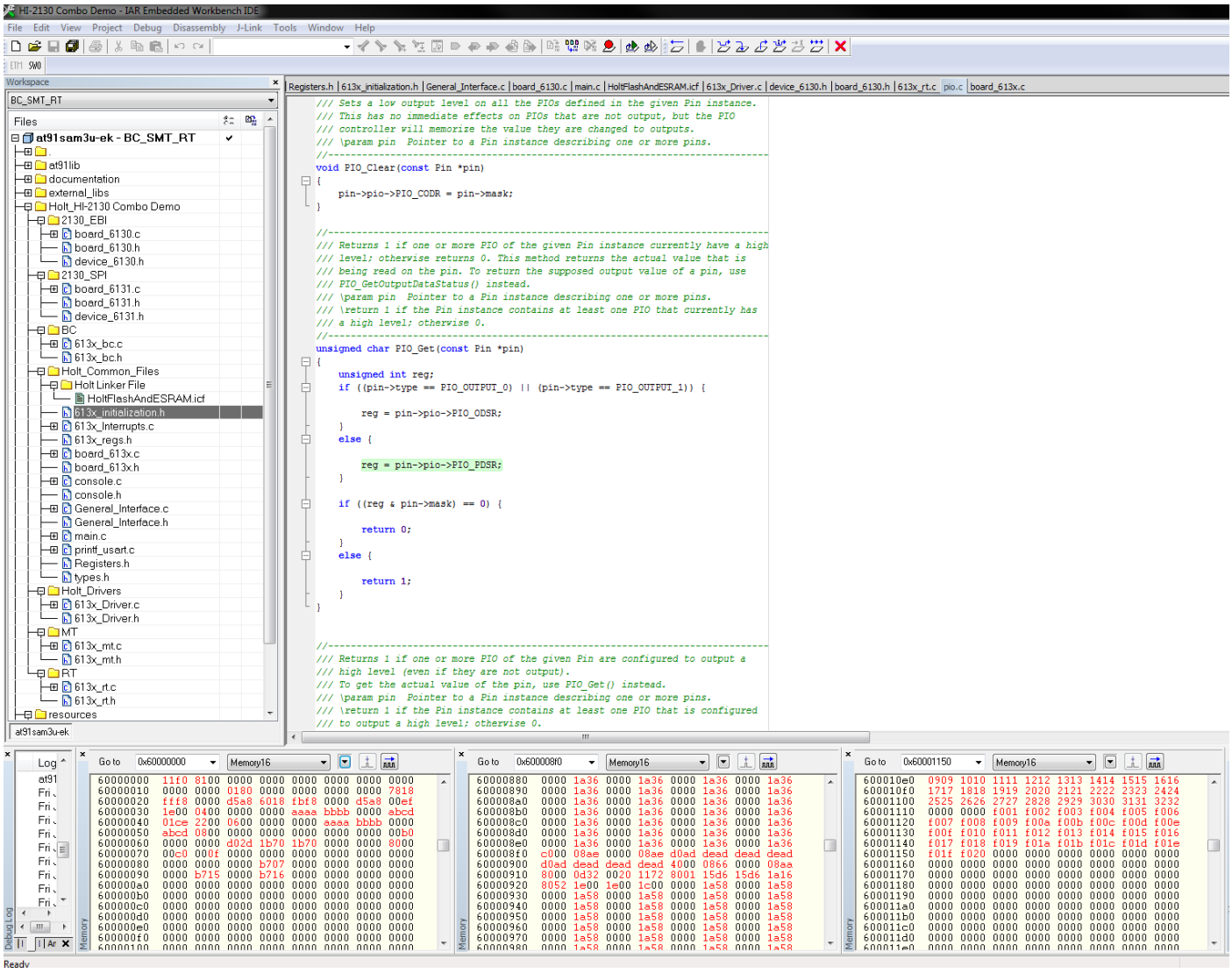
11. It's possible to view multiple debugger windows including several memory windows when using the 16-bit parallel interface (EBI mode). Because the HI-2130 is a memory mapped device all the system register space and control block areas of the HI-2130 can be viewed making debugging code easier. This will not work in SPI so use the SPI console commands to display memory instead. A breakpoint was set and executed below.

12. To program the auto-initialization serial EEPROM with a new configuration:

    - Before starting program execution, turn off the DIP switch labeled AUTOEN, directing the MCU to initialize the HI-2130 instead of using self-initialization from the EEPROM. Turn on the DIP switch labeled COPYREQ, directing the MCU to initiate the EEPROM copy sequence after post-reset initialization of HI-2130 registers and RAM is complete.

- When execution starts, the red LED illuminates during the EEPROM copy process. When it turns off, turn on the DIP switch labeled AUTOEN to reactivate self-initialization. Turn off the COPYREQ DIP switch, preventing EEPROM rewrite at each reset.

13. This screen was captured after recompiling the program for EBI mode (no SPI) by first changing the setting the macro HOST_BUS_INTERFACE =1. Download the program but before clicking the Run button on the IAR IDE open three memory windows from the View pull down menu and position and size them similarly shown below. To view HI-2130 registers space, RT1 control blocks and data buffers, select the Memory pull-down menu, then select Memory16; in the small adjacent pull-down select 2x units. In the first window, enter base address 0x60000000 in the “Go to” box so the HI-2130 register space is displayed. Set the second window to view SA30 RX control block enter 0x600008F0. Click the RUN button. After the console screen is displayed, click the Break button to stop the program, the screen should update as shown. IAR debugger window addresses are byte addresses.



RX SA1 though SA30 control blocks (4 words each) span address 0x60000808 through 0x600008F0. To calculate the IDE byte address for a particular SA control block add the upper offset address 0x6000 to the particular base address (0x0800 for RX, 0x900 for TX) for RT1. Multiply the SA value by eight and add that to the base address.

2130 offset, base address for RT1 RX, SA1
0x6000-0800 + 1 x 8 = 0x6000-0808

For example the byte address for RT1, RX SA1 is shown in the first row below:

2130 base address	Control Block Address (for RT1 **RX**)	SA	IDE Full Address
0x6000-0000	0x0800	1	0x60000808
0x6000-0000	0x0800	2	0x60000810
0x6000-0000	0x0800	30	0x600008F0

2130 base address	Control Block Address (for RT1 **TX**)	SA	IDE Full Address
0x6000-0000	0x0900	1	0x60000908
0x6000-0000	0x0900	2	0x60000910
0x6000-0000	0x0900	30	0x600009F0

To calculate the data buffer address for these RT control blocks examine Word 2 or Word 3 (depending on the control word type) in the four word control block. For example the RT1 RX SA30 control block contains these words at address 0x600008F0:

RT1 RX SA30 = 0x600008F0

Descriptor Word 1 Control Word	Descriptor Word 2 Data Pointer A	Descriptor Word 3 Data Pointer B	Descriptor Word 4 Broadcast Data Pointer
0xC000	0x8AE	0x0000	0x08AE

The Control Word is defined as a Single Message Mode with the data pointer address set to 0x08AE. With Single Message Mode, Word 2 is the data pointer and Word 3 is set to 0x0000. See section 19.4 Descriptor Table in the data sheet for a full description of the control word types and buffer schemes. Since bit-14 (IWA) is set high in this Control Word a message interrupt will occur when this message is transacted by the RT (see section 19.4.1 of the data sheet). When the demo program is compiled with the C macro “INT” set to 0 (default), RT message data is captured by polling RT_MesgRead(). When “INT” set to 1 , RT_MesgRead() is called in the interrupt function. In both cases the DBAC bit in the corresponding message control word is examined to determine if the Descriptor Block was accessed for the current message (message just received) in order to store the message data into the C struct. When the program is configured for message interrupts (INT=1) then

only RT messages with the IWA bit-14 set high in the RT control word in the descriptor table will generate a message interrupt and displayed on the console using the T command. These words are initialized in the device descriptor table by function initialize_613x_RT1() using array descr_table_RT1[512].

To calculate the Data Pointer address, multiple 0x8AE x 2 = 0x115C. The first word at 0x115C is the MIW (message information word), followed by the (TT) Time-Tag word followed by up to 32 words beginning at address 0x1160. These addresses are 16-bit word address so to get the full address add the 0x6000 offset to these addresses (0x60001160). For further explanation of the 2130 control block addresses for BC, RT, RT2 and SMT refer to the HI-6130 data sheet. A 6130(2130) memory map is provided on page 22 which shows the starting addresses for the control blocks and data buffer areas, the data sheet provides more information on all the control block and data structure addresses for all the terminals.

SPI Interface

In SPI mode, the IAR debugger is not capable of showing device memory (registers/control block/buffer data) so use the special pre configured commands: space bar, Z, X,C,V,B and I to display the corresponding control blocks and data buffer to the console. Custom commands can be added by the user to display data at any address. These commands display data as word addresses. See console.c for these example commands.

```
SPI 2130 reg/memory Display Commands:
Press Space Bar Display Registers 0x-0x0050
Press 'A' Display Memory at address
Press 'Z' Display RT control block
Press 'X' Display RT data block
Press 'C' Display RT2 control block
Press 'V' Display RT2 data block
Press 'B' Display BC mesg and data block
Press 'I' Display Interrupt Log
```

```
2130 Registers:
          0    1    2    3    4    5    6    7    8    9    A    B    C    D    E    F
Adr 0000: 19F0 8100 0000 0000 0000 0000 0000 0000 0000 0000 0180 527A 0000 000D 0000 7818
Adr 0010: FFF8 0000 D458 0058 FBF8 0000 D5A8 00EF 1E00 0400 0000 0000 AAAA BBBB 0000 ABCD
Adr 0020: 01CE 2200 0600 0000 0000 AAAA BBBB 0000 ABCD 0800 0000 0000 0000 0000 0000 00B0
Adr 0030: 0000 0000 D02D 1B70 1B70 0000 0000 8000 00C0 000F 0000 0000 0000 0000 0000 0000
Adr 0040: 0000 0000 0000 C092 004B 0000 0000 0000 0000 C0BF 0000 C0CD 0000 0000 0000 0000
Adr 0050: 0000
>>
```

Key Project Files with Selected Descriptions

HEADER FILES WITHOUT CORRESPONDING C FILES

613x_initialization.h

Definitions for important configuration settings

HOST_BUS_INTERFACE

1: Set program compilation for EBI parallel bus.

0: Set program compilation for SPI bus.

BOTH_EBI_SPI (only valid with HOST_BUS_INTERFACE = 0)

0: Normal

1: When HOST_BUS_INTERFACE = 0, allows EBI debugger memory window usage.

device_6130.h

ARM MCU external bus interface definitions and structures for register addressing the HI- 2130.

device_6131.h

ARM MCU SPI interface definitions and structures for register addressing the HI- 2130.

C FILES WITH CORRESPONDING HEADER FILES

main.c

The primary program entry portal, `main()` demonstrates the initialization sequence and enters an endless loop waiting for console commands entered from the PC keyboard using the terminal program.

board_6130/31.c

This function initializes ARM MCU external bus interface for the HI-6130 and the 128K external SRAM.

console.c

This function looks for console key presses and executes corresponding demo commands.

613x_bc.c

This module includes functions to initialize and support BC operation. When used the BCENA DIP switch must be set high(ON).

613x_mt.c

This module includes functions to initialize and support SMT operation. When used the MTRUN DIP switch must be set high(ON).

613x_rt.c

This module includes functions to initialize and support RT(AKA RT1) and RT2 operation. When used the RT1 and RT2 ENA DIP switches must be set high(ON).

613x_Interrupts.c

The primary purpose is to initialize and process RT message interrupts to capture RT message data.

613x_Driver.c

Includes functions to capture RT message data (by interrupt or polling) and stores the data in a C message structure. When the console 'T' command is used a flag is set and RT message data is displayed on the console when messages are detected in the main loop.

```
MSG #0001.  TIME = 00000000us    Bus A
            CMD 0x1820 --> 03-R-01-00    BC to RT
            MIW 0x0000
            DATA 0003  0201  0202  0203  0204  0205  0206  0207
                 0208  0209  020A  020B  020C  020D  020E  020F
                 0210  0211  0212  0213  0214  0215  0216  0217
                 0218  0219  021A  021B  021C  021D  021E  021F
```

Application Development Kit Notes

The HI-2130 data sheet is included on the kit CD-ROM. For reference, the HI-6130 data sheet is also found there. The latest revision for these documents can be found at www.holtic.com.

Primary project configuration settings are found in the header file **`613x_initialization.h.`**

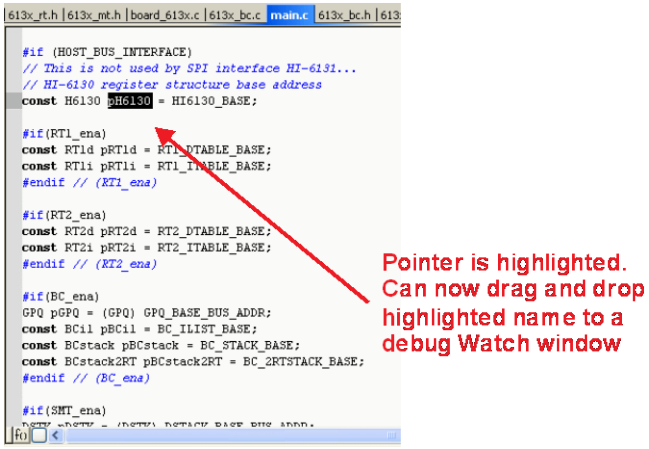
The HI-2130 was designed for compatibility with microcontrollers having an external parallel 16-bit (EBI) or SPI interface. RAM and register locations appear in the memory address space of the ARM Cortex M3 microprocessor on the MCU board when using EBI. The utilized MCU chip select output (connected to the HI-2130 chip enable input pin) accesses a memory region starting at MCU bus address 0x60000000. Byte addressing is used. RAM and register operations transact 16-bit values, so all access addresses are even. To use byte addressing, HI-2130 RAM or register addresses are doubled and then added to the MCU chip select base address. Therefore, HI-2130 Register 0 is accessed at MCU bus address 0x60000000. Register 1 is accessed at bus address 0x60000002, and Register 5 is accessed at address 0x6000000A.

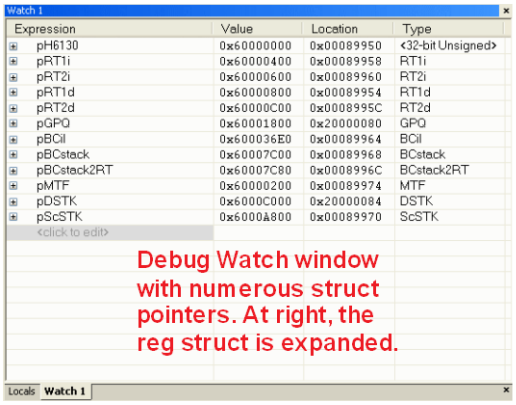
The evaluation board program provides comprehensive bus addressing examples for the fixed-address and relocatable RAM structures used by each of the MIL-STD-1553 terminal modes: BC, RT or monitor.

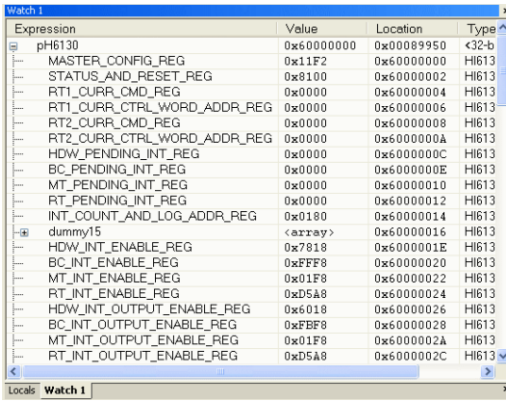
When using the Debugger, a Memory window may be helpful for observing register or RAM values, updated each time program execution stops. Be mindful that each displayed location is rescanned when execution stops. Some register or RAM structure bits automatically reset after read occurs. This includes bits in the Pending Interrupt registers, and DBAC Data Block Accessed bits for RT Descriptor Table Control Words in RAM. For these, the memory window reflects the value in effect when execution stopped.

When using the Debugger, a memory window may be helpful for observing values contained in the various defined RAM structures for enabled MIL-STD-1553 modes, updated each time program execution stops. When debugging, IAR Embedded Workbench® allows up to 4 Watch windows, so separate Watch tabs can be set up for BC, RT1, RT2, etc. Memory Watch windows allow viewing symbol data by variable name and are useful for viewing global variables. For each structure of interest, double-click the struct pointer name to highlight it, then drag and drop the highlighted pointer name into an

open Watch window, where it can be examined. Be mindful that some RAM structure bits automatically reset after read occurs. This includes the DBAC Data Block Accessed bits for RT Descriptor Table Control Words. For these, the Memory Watch window reflects the value in effect when execution stopped.

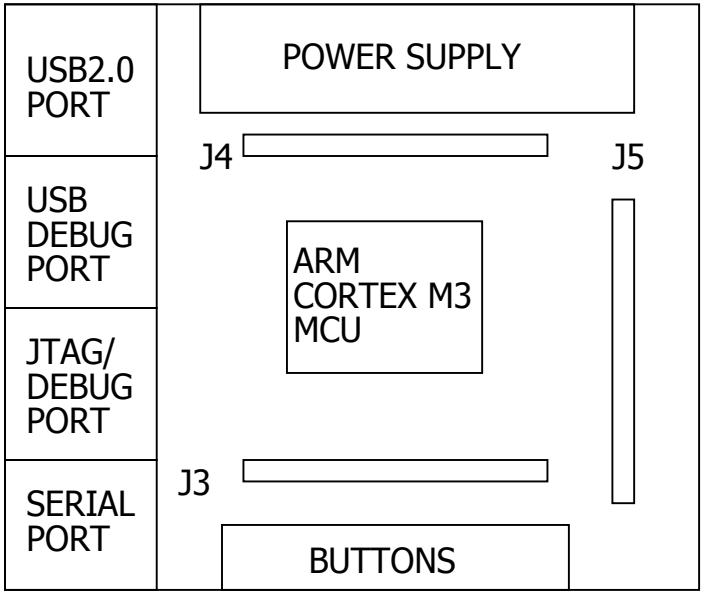




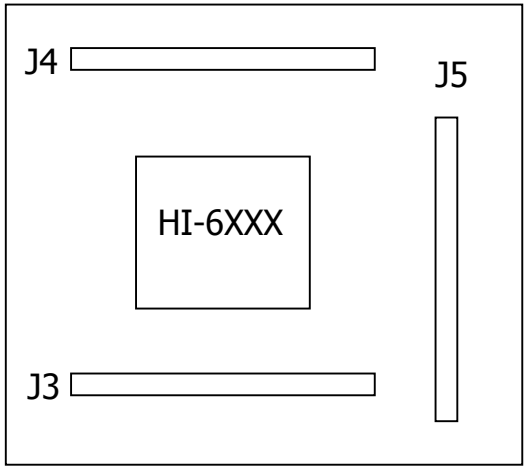


Evaluation board schematic diagrams and bills of material are provided on the following pages

At the end of the document, memory maps show how RAM is allocated.



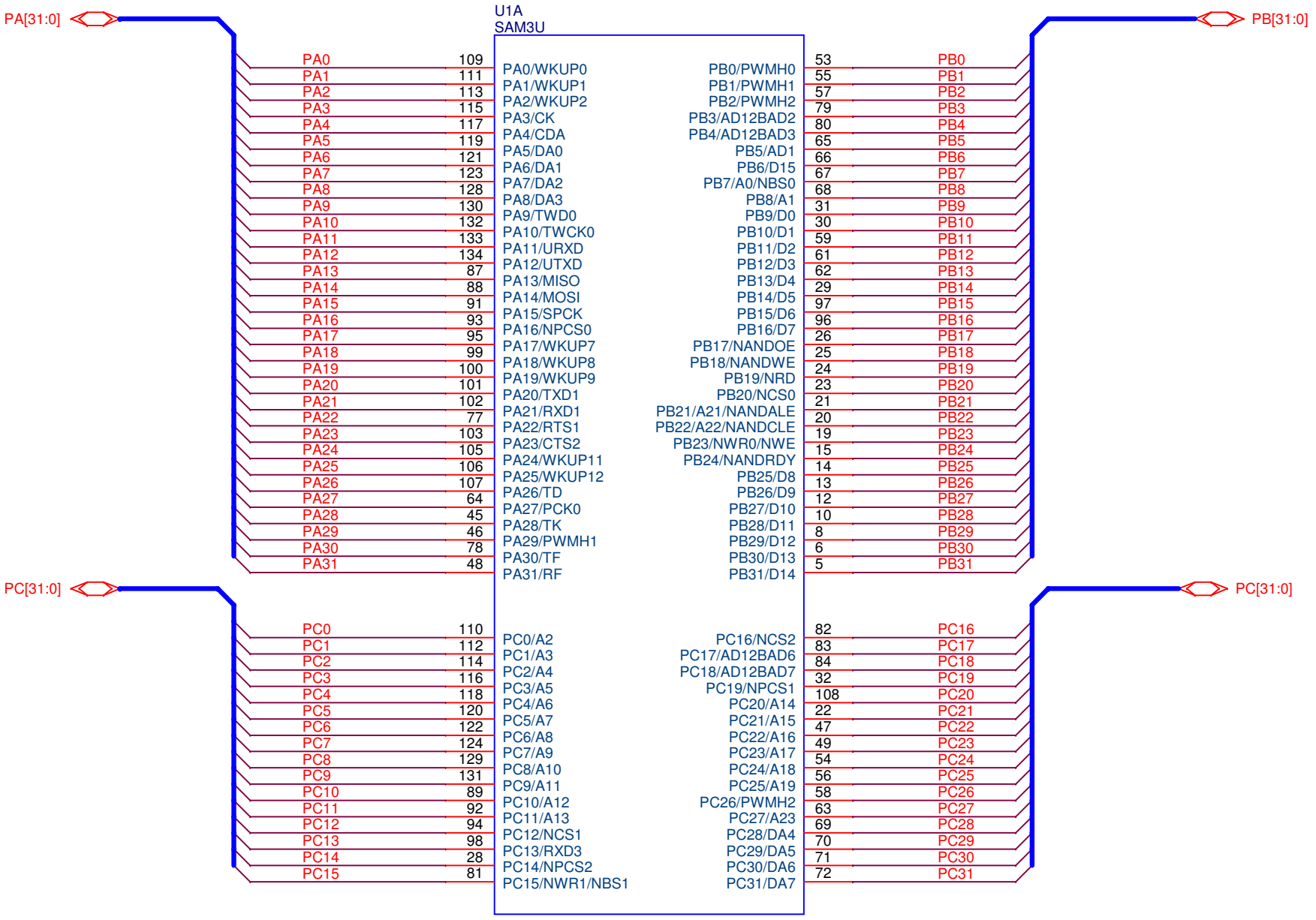
LOWER CIRCUIT BOARD



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 F.DSN	Rev: F
Date: Tuesday, April 11, 2017	Sheet 1 of 7	



ARM CORTEX M3 PIO

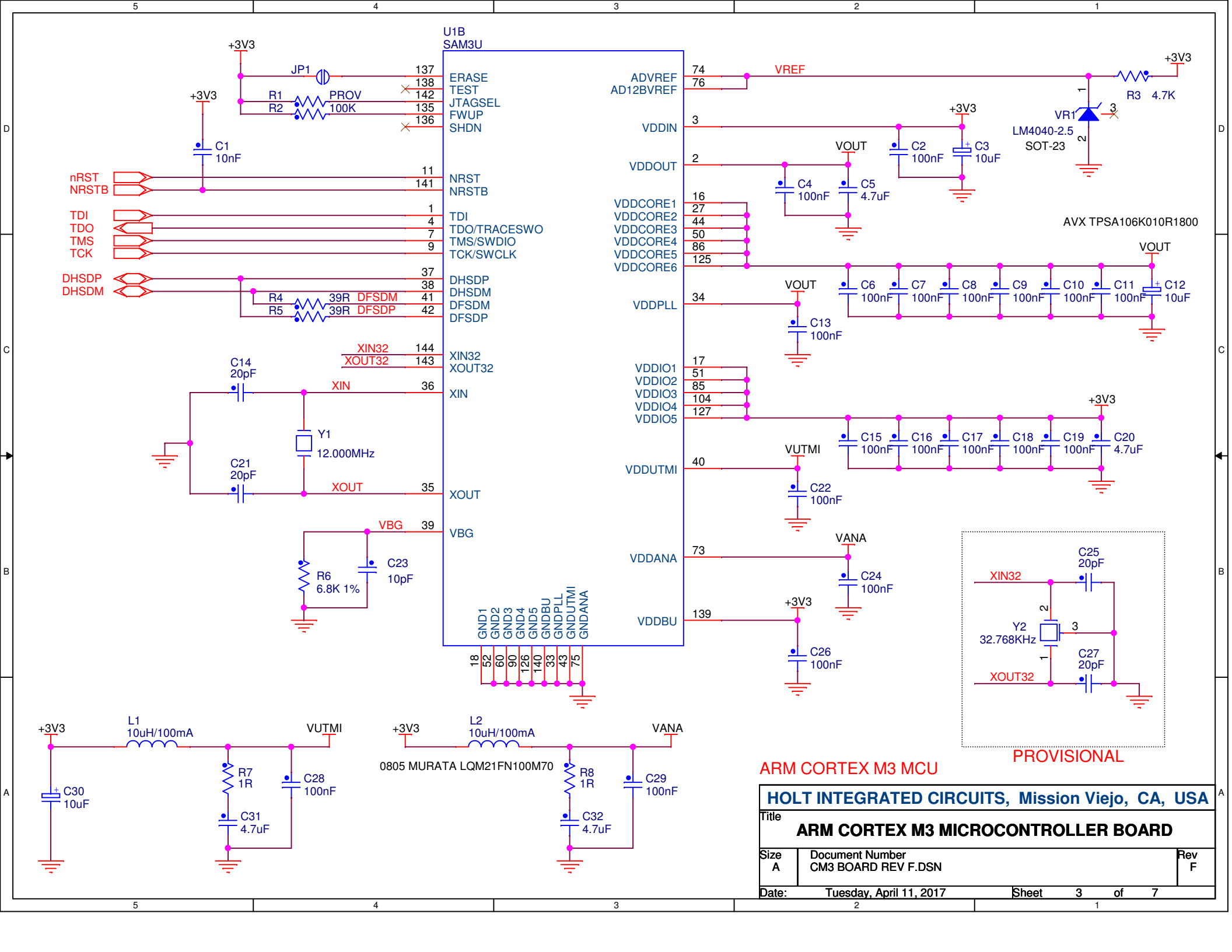
HOLT INTEGRATED CIRCUITS, Mission Viejo, CA, USA

Title

ARM CORTEX M3 MICROCONTROLLER BOARD

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

Date: Tuesday, April 11, 2017 Sheet 2 of 7



PROVISIONAL

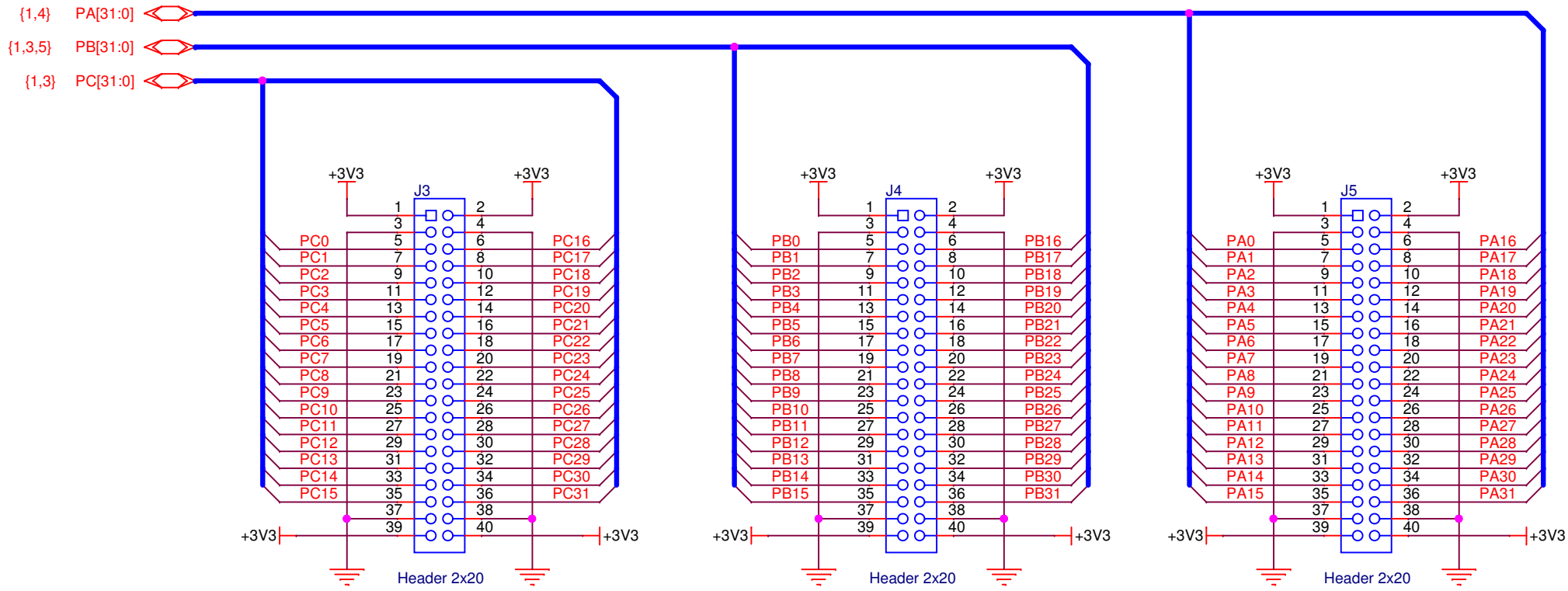
ARM CORTEX M3 MCU

HOLT INTEGRATED CIRCUITS, Mission Viejo, CA, USA

Title
ARM CORTEX M3 MICROCONTROLLER BOARD

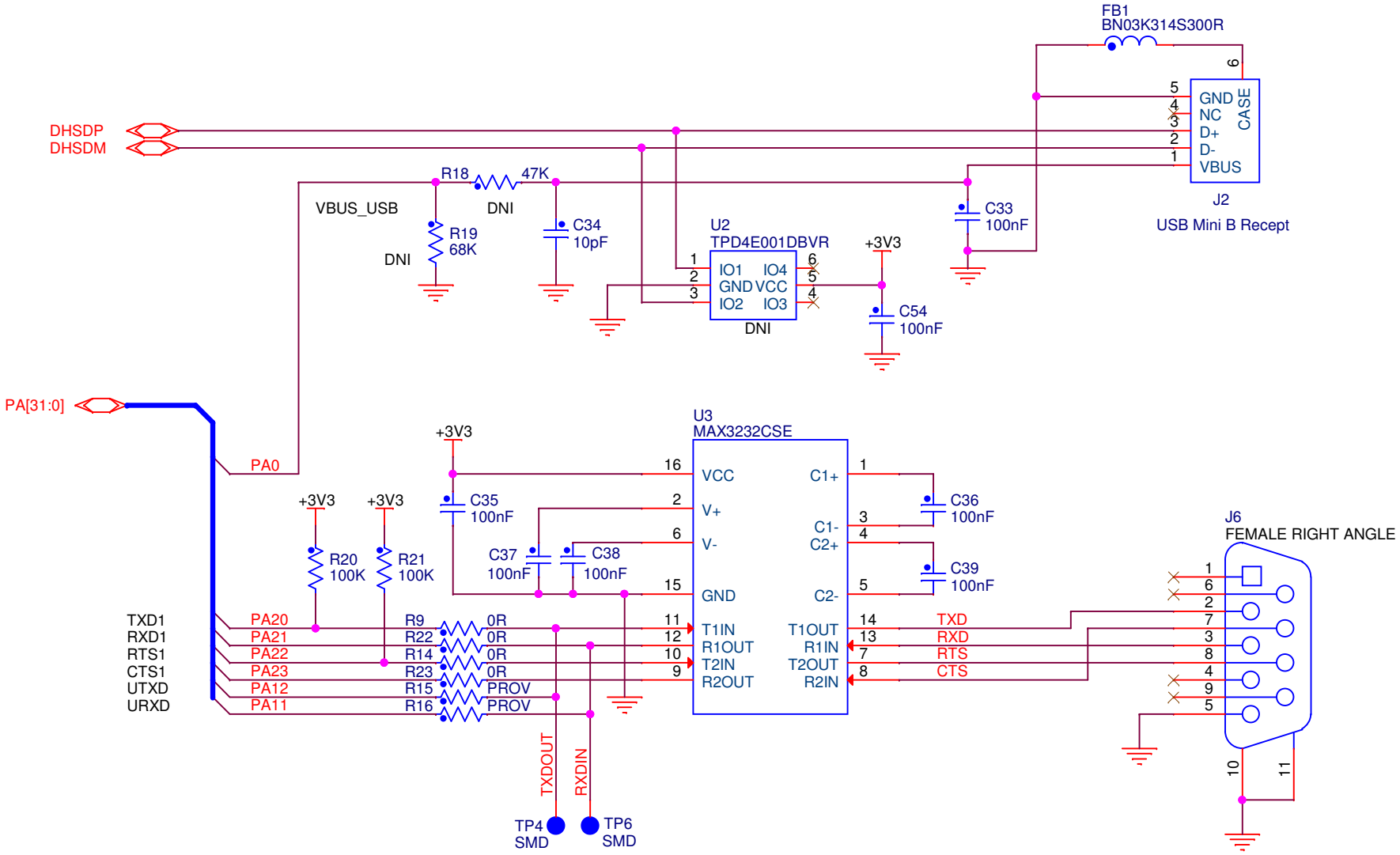
Size	Document Number	Rev
A	CM3 BOARD REV F.DSN	F

Date: Tuesday, April 11, 2017 Sheet 3 of 7



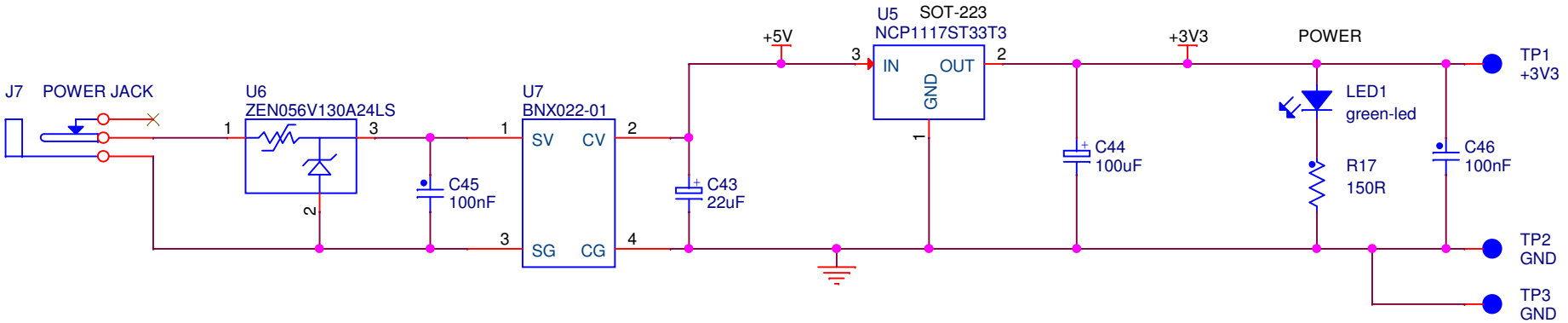
BOARD I/O HEADERS, BUTTONS

HOLT INTEGRATED CIRCUITS, Mission Viejo, CA, USA		
Title: ARM CORTEX M3 MICROCONTROLLER BOARD		
Size: A	Document Number: CM3 BOARD REV F.DSN	Rev: F
Date: Tuesday, April 11, 2017	Sheet 4 of 7	



USB & RS-232 SERIAL

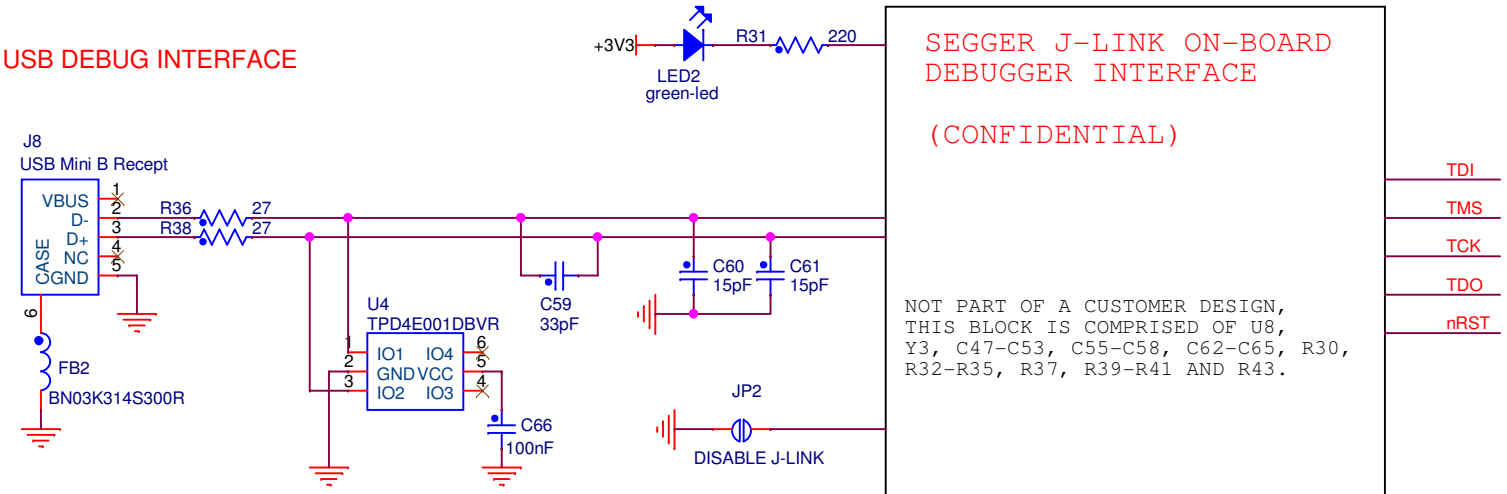
HOLT INTEGRATED CIRCUITS, Mission Viejo, CA, USA		
Title: ARM CORTEX M3 MICROCONTROLLER BOARD		
Size: A	Document Number: CM3 BOARD REV F.DSN	Rev: F
Date: Tuesday, April 11, 2017	Sheet 5 of 7	



POWER SUPPLY

HOLT INTEGRATED CIRCUITS, Mission Viejo, CA, USA		
Title: ARM CORTEX M3 MICROCONTROLLER BOARD		
Size: A	Document Number: CM3 BOARD REV F.DSN	Rev: F
Date: Tuesday, April 11, 2017	Sheet 6 of 7	

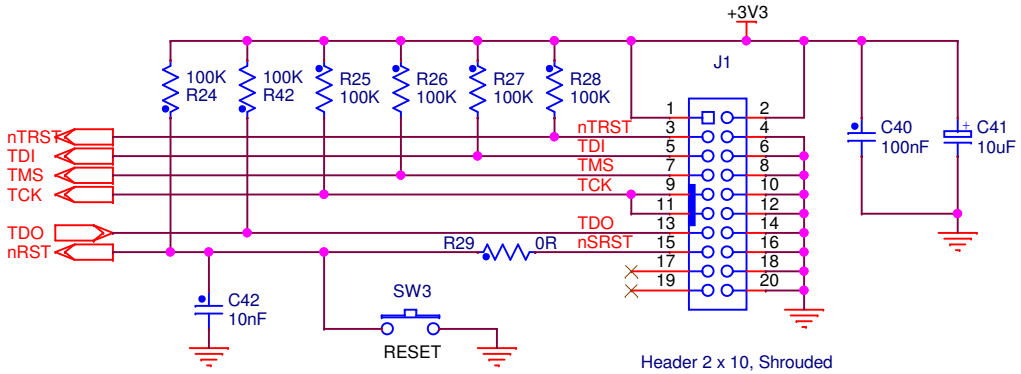
USB DEBUG INTERFACE



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

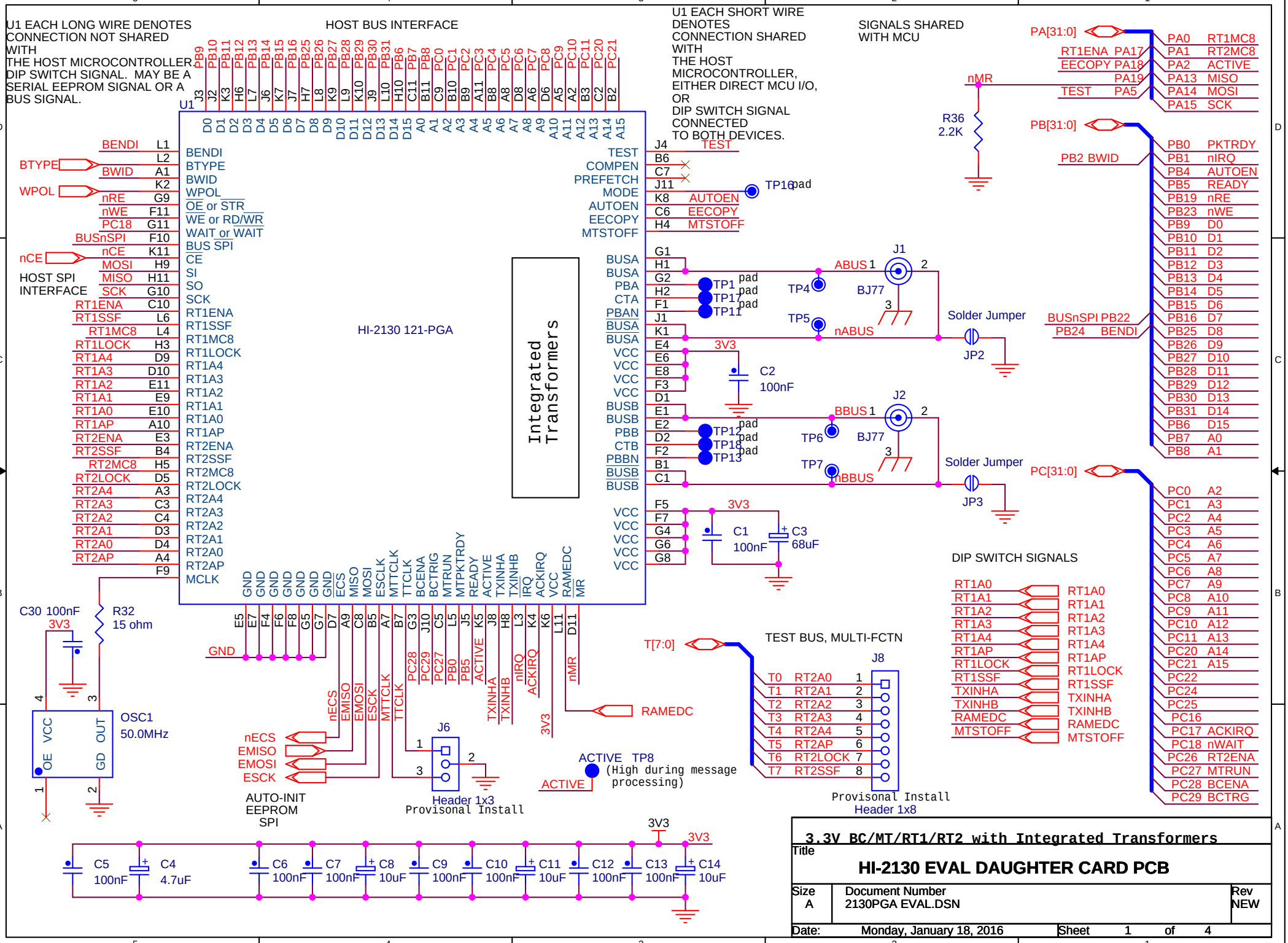


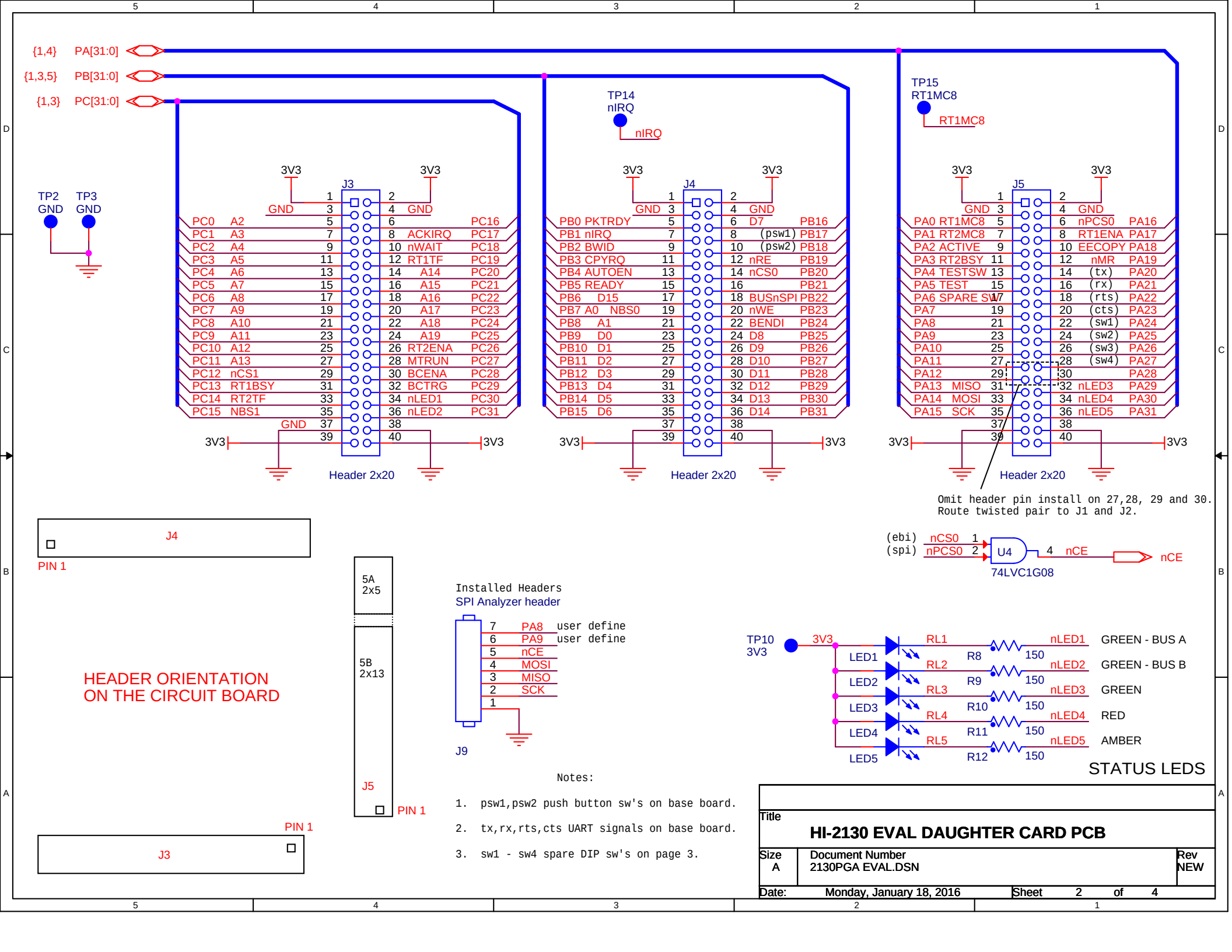
HOLT INTEGRATED CIRCUITS, Mission Viejo, CA, USA		
Title: ARM CORTEX M3 MICROCONTROLLER BOARD		
Size	Document Number: CM3 BOARD REV F.DSN	Rev: F
Date: Tuesday, April 11, 2017	Sheet 7 of 7	

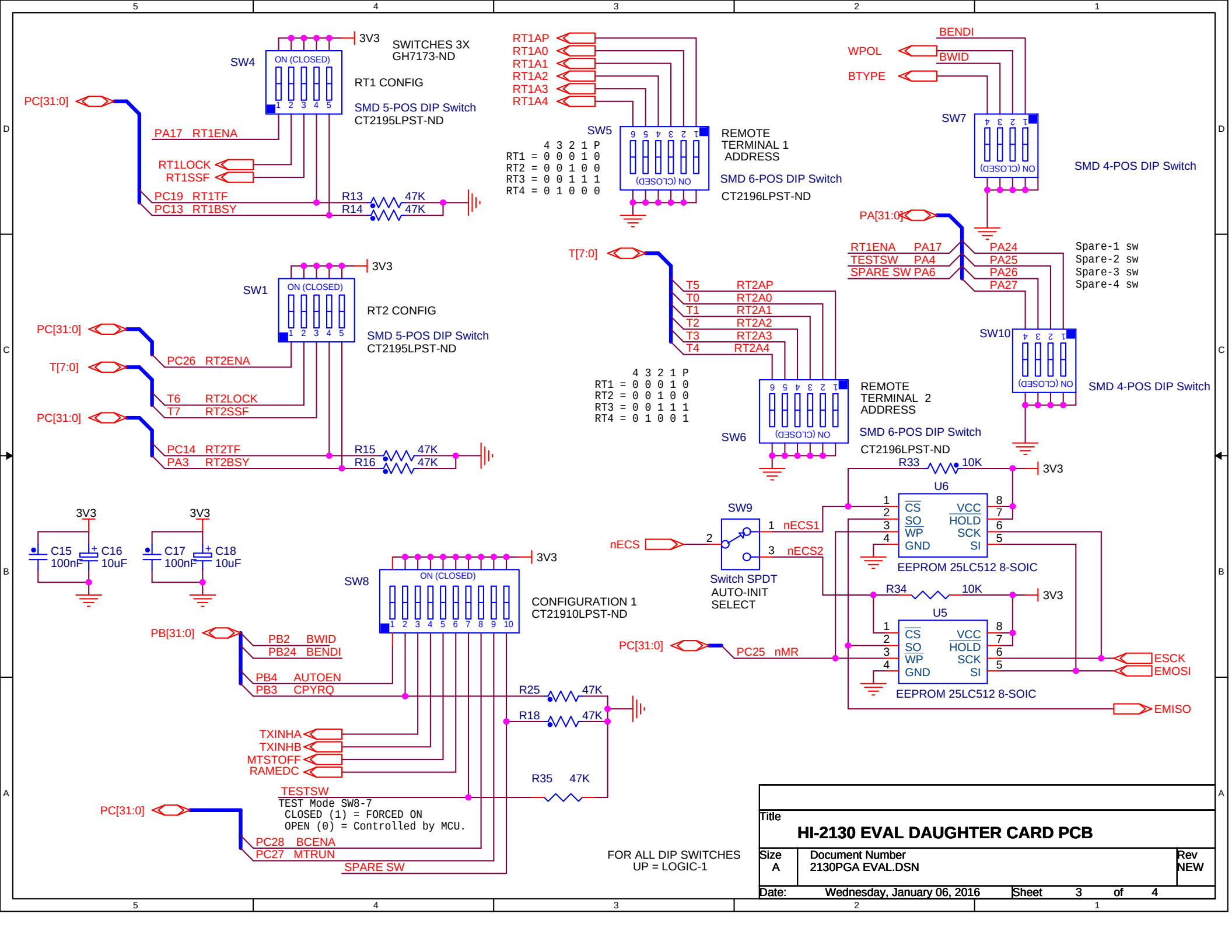
PCB P/N: HV028
April 10, 2017

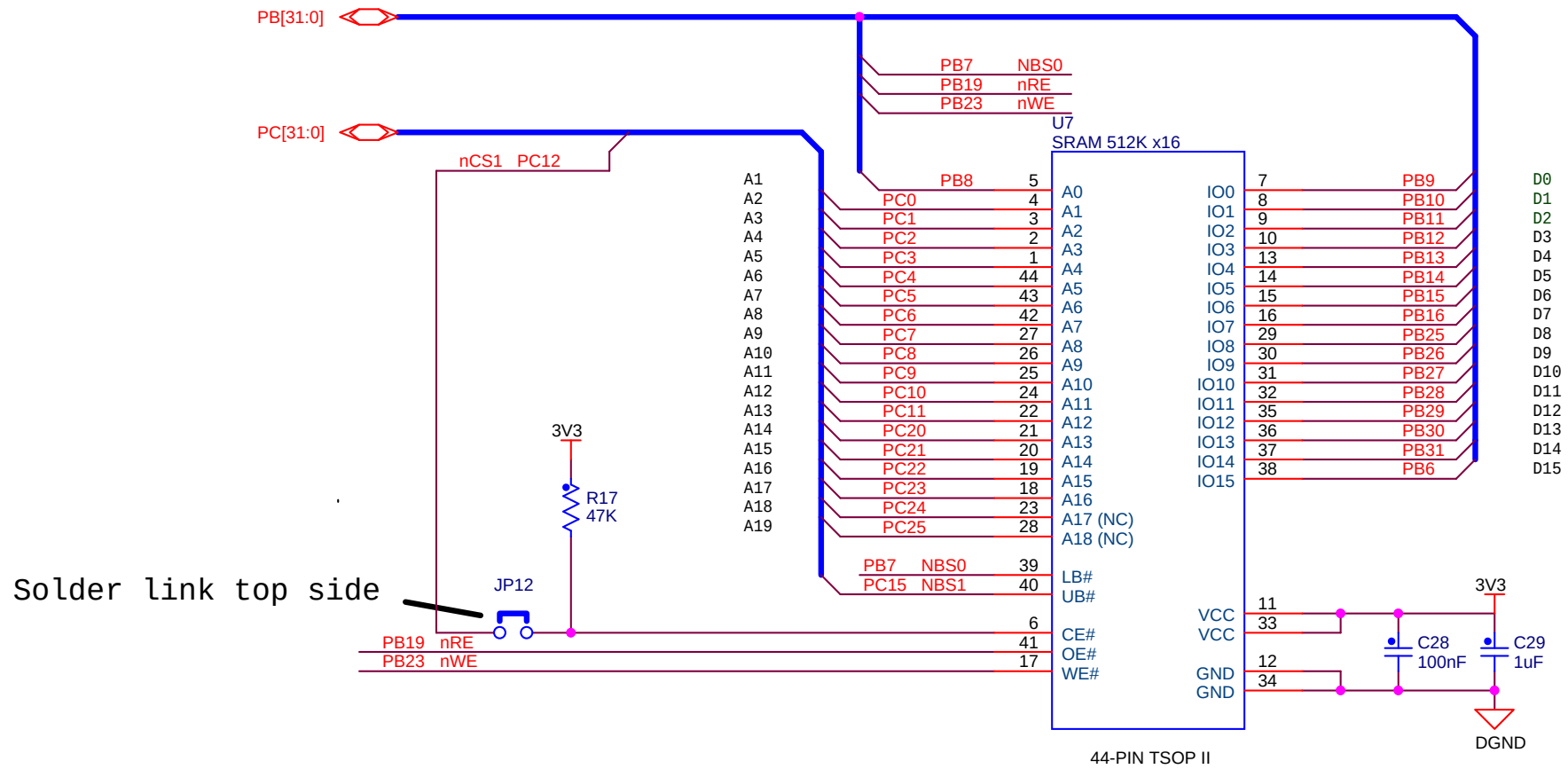
Bill of Materials
ARM Cortex M3 MCU Board
Rev. F

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	2	Connector, Receptacle USB Mini B Rt-Angle PCB Mount	J2,J8	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	2	LED Green 0805	LED1,LED2	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	2	4-Ch TVS ESD Protection SOT23-6	U2,U4	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	1	Capacitor, Ceramic 100nF, -20% / +80% 25V Y5V 0603	C66	490-1575-1-ND	Murata GRM188F51E104ZA01D
43	1	Capacitor, Ceramic 33pF, 5% 50V C0G 0603	C59	490-1415-1-ND	Murata GRM1885C1H330JA01D
44	2	Capacitor, Ceramic 15pF, 5% 50V C0G 0603	C60,C61	490-1407-1-ND	Murata GRM1885C1H150JA01D
45	1	Ferrite Bead, 220 Ohm @ 100MHz 300mA DC 0805	FB2	732-1602-1-ND	Wurth 742792034
46	1	Solder Jumper	JP2	SOLDER OPEN	
47	1	Resistor, 220 ohm 5% 1/10W 0603	R31	P220GCT-ND	Panasonic ERJ-3GEYJ221V
48	2	Resistor, 27 ohm 5% 1/10W 0603	R36,R38	P27GCT-ND	Panasonic ERJ-3GEYJ270V
49	5	Rubber Foot, Bumpon Black Hemisphere, .312 X.200 H	Place at 4 corners and center	SJ5746-0-ND	3M SJ61A1


U1 EACH LONG WIRE DENOTES CONNECTION NOT SHARED WITH THE HOST MICROCONTROLLER DIP SWITCH SIGNAL. MAY BE A SERIAL EEPROM SIGNAL OR A BUS SIGNAL.
HOST BUS INTERFACE
U1 EACH SHORT WIRE DENOTES CONNECTION SHARED WITH THE HOST MICROCONTROLLER, EITHER DIRECT MCU I/O, OR DIP SWITCH SIGNAL CONNECTED TO BOTH DEVICES.
SIGNALS SHARED WITH MCU
HI-2130 121-PGA
Integrated Transformers
HOST SPI INTERFACE
BENDI
BTYPE
BWID
WPOL
nRE
nWE
BUSnSPI
nCE
MOSI
MISO
SCK
TEST
COMPEN
PREFETCH
MODE
AUTOEN
EECOPY
MTSTOFF
TP16 pad
TP1 pad
TP17 pad
TP11 pad
TP4
TP5
ABUS
nABUS
BJ77
J1
J2
JP2
JP3
Solder Jumper
TP12 pad
TP18 pad
TP13 pad
TP6
TP7
BBUS
nBBUS
C1 100nF
C2 100nF
C3 68uF
3V3
R36 2.2K
nMR
PA[31:0]
PB[31:0]
PC[31:0]
DIP SWITCH SIGNALS
TEST BUS, MULTI-FCTN
T[7:0]
J8
Provisional Install Header 1x8
OSC1 50.0MHz
C30 100nF
R32 15 ohm
nECS
EMISO
EMOSI
ESCK
AUTO-INIT EEPROM SPI
J6
Header 1x3 Provisional Install
RAMEDC
ACTIVE TP8 (High during message processing)
C4 4.7uF
C5 100nF
C6 100nF
C7 100nF
C8 10uF
C9 100nF
C10 100nF
C11 10uF
C12 100nF
C13 100nF
C14 10uF
3.3V BC/MT/RT1/RT2 with Integrated Transformers
HI-2130 EVAL DAUGHTER CARD PCB
Size A
Document Number 2130PGA EVAL.DSN
Rev NEW
Date: Monday, January 18, 2016
Sheet 1 of 4


{1,4} PA[31:0]
{1,3,5} PB[31:0]
{1,3} PC[31:0]
TP2 GND
TP3 GND
J3
3V3
GND
PC0 A2 PC16
PC1 A3 8 ACKIRQ PC17
PC2 A4 10 nWAIT PC18
PC3 A5 12 RT1TF PC19
PC4 A6 14 A14 PC20
PC5 A7 16 A15 PC21
PC6 A8 18 A16 PC22
PC7 A9 20 A17 PC23
PC8 A10 22 A18 PC24
PC9 A11 24 A19 PC25
PC10 A12 26 RT2ENA PC26
PC11 A13 28 MTRUN PC27
PC12 nCS1 30 BCENA PC28
PC13 RT1BSY 32 BCTRG PC29
PC14 RT2TF 34 nLED1 PC30
PC15 NBS1 36 nLED2 PC31
Header 2x20
TP14 nIRQ
nIRQ
J4
PB0 PKTRDY 6 D7 PB16
PB1 nIRQ 8 (psw1) PB17
PB2 BWID 10 (psw2) PB18
PB3 CPYRQ 12 nRE PB19
PB4 AUTOEN 14 nCS0 PB20
PB5 READY 16 PB21
PB6 D15 18 BUSnSPI PB22
PB7 A0 NBS0 20 nWE PB23
PB8 A1 22 BENDI PB24
PB9 D0 24 D8 PB25
PB10 D1 26 D9 PB26
PB11 D2 28 D10 PB27
PB12 D3 30 D11 PB28
PB13 D4 32 D12 PB29
PB14 D5 34 D13 PB30
PB15 D6 36 D14 PB31
TP15 RT1MC8
RT1MC8
J5
PA0 RT1MC8 6 nPCS0 PA16
PA1 RT2MC8 8 RT1ENA PA17
PA2 ACTIVE 10 EECOPY PA18
PA3 RT2BSY 12 nMR PA19
PA4 TESTSW 14 (tx) PA20
PA5 TEST 16 (rx) PA21
PA6 SPARE SW 18 (rts) PA22
PA7 20 (cts) PA23
PA8 22 (sw1) PA24
PA9 24 (sw2) PA25
PA10 26 (sw3) PA26
PA11 28 (sw4) PA27
PA12 30 PA28
PA13 MISO 32 nLED3 PA29
PA14 MOSI 34 nLED4 PA30
PA15 SCK 36 nLED5 PA31
Omit header pin install on 27,28, 29 and 30.
Route twisted pair to J1 and J2.
(ebi) nCS0 1
(spi) nPCS0 2
U4
74LVC1G08
4 nCE
nCE
J4
PIN 1
5A 2x5
5B 2x13
J5
PIN 1
HEADER ORIENTATION ON THE CIRCUIT BOARD
PIN 1
J3
Installed Headers
SPI Analyzer header
7 PA8 user define
6 PA9 user define
5 nCE
4 MOSI
3 MISO
2 SCK
1
J9
TP10 3V3
3V3
LED1 RL1 R8 150 nLED1 GREEN - BUS A
LED2 RL2 R9 150 nLED2 GREEN - BUS B
LED3 RL3 R10 150 nLED3 GREEN
LED4 RL4 R11 150 nLED4 RED
LED5 RL5 R12 150 nLED5 AMBER
STATUS LEDS
Notes:
1. psw1,psw2 push button sw's on base board.
2. tx,rx,rts,cts UART signals on base board.
3. sw1 - sw4 spare DIP sw's on page 3.
Title
HI-2130 EVAL DAUGHTER CARD PCB
Size A
Document Number 2130PGA EVAL.DSN
Rev NEW
Date: Monday, January 18, 2016
Sheet 2 of 4


SW4
ON (CLOSED)
1 2 3 4 5
3V3
SWITCHES 3X
GH7173-ND
RT1 CONFIG
SMD 5-POS DIP Switch
CT2195LPST-ND
PC[31:0]
PA17 RT1ENA
RT1LOCK
RT1SSF
PC19 RT1TF
PC13 RT1BSY
R13 47K
R14 47K
RT1AP
RT1A0
RT1A1
RT1A2
RT1A3
RT1A4
SW5
4 3 2 1 P
RT1 = 0 0 0 1 0
RT2 = 0 0 1 0 0
RT3 = 0 0 1 1 1
RT4 = 0 1 0 0 0
REMOTE
TERMINAL 1
ADDRESS
SMD 6-POS DIP Switch
CT2196LPST-ND
BENDI
WPOL
BWID
BTYPE
SW7
SMD 4-POS DIP Switch
PA[31:0]
RT1ENA PA17
TESTSW PA4
SPARE SW PA6
PA24
PA25
PA26
PA27
Spare-1 sw
Spare-2 sw
Spare-3 sw
Spare-4 sw
SW10
SMD 4-POS DIP Switch
SW1
RT2 CONFIG
SMD 5-POS DIP Switch
CT2195LPST-ND
PC26 RT2ENA
T6 RT2LOCK
T7 RT2SSF
PC14 RT2TF
PA3 RT2BSY
R15 47K
R16 47K
T[7:0]
T5 RT2AP
T0 RT2A0
T1 RT2A1
T2 RT2A2
T3 RT2A3
T4 RT2A4
4 3 2 1 P
RT1 = 0 0 0 1 0
RT2 = 0 0 1 0 0
RT3 = 0 0 1 1 1
RT4 = 0 1 0 0 1
SW6
REMOTE
TERMINAL 2
ADDRESS
SMD 6-POS DIP Switch
CT2196LPST-ND
R33 10K
U6
CS
SO
WP
GND
VCC
HOLD
SCK
SI
EEPROM 25LC512 8-SOIC
SW9
nECS
nECS1
nECS2
Switch SPDT
AUTO-INIT
SELECT
R34 10K
U5
EEPROM 25LC512 8-SOIC
PC25 nMR
ESCK
EMOSI
EMISO
C15 100nF
C16 10uF
C17 100nF
C18 10uF
SW8
CONFIGURATION 1
CT21910LPST-ND
PB[31:0]
PB2 BWID
PB24 BENDI
PB4 AUTOEN
PB3 CPYRQ
R25 47K
R18 47K
R35 47K
TXINHA
TXINHB
MTSTOFF
RAMEDC
TESTSW
TEST Mode SW8-7
CLOSED (1) = FORCED ON
OPEN (0) = Controlled by MCU.
PC28 BCENA
PC27 MTRUN
SPARE SW
FOR ALL DIP SWITCHES
UP = LOGIC-1
Title
HI-2130 EVAL DAUGHTER CARD PCB
Size
A
Document Number
2130PGA EVAL.DSN
Rev
NEW
Date:
Wednesday, January 06, 2016
Sheet
3
of
4



Title: HI-2130 EVAL PCB (USE WITH STD ARM CM3 LOWER PCB)		
Size: A	Document Number: 2130PGA EVAL.DSN	Rev: NEW
Date: Monday, January 18, 2016	Sheet 4 of 4	

Holt P/N: HV016
Rev. A

Bill of Material
HI-2130 PGA
Evaluation Board

Item	Qty	Description	Reference	DigiKey	Mfr P/N
1	1	PCB, Bare, Eval Board	N/A	--------	JetTech # 38233
2	14	Capacitor, Ceramic 0.1uF 20% 50V Z5U 0805	C1,C2,C5,C6,C7,C9,C10,C12, C13,C15,C17,C30,C28,C29	399-1176-1-ND	Kemet C0805C104M5UACTU
3	1	Capacitor, Ceramic 4.7uF 10% 6.3V X5R 0805	C4	399-3134-1-ND	Kemet C0805C475K9PACTU
4	5	Capacitor, Ceramic 10uF 10% 6.3V X5R 0805	C8,C11,C14,C16,C18	399-3138-1-ND	Kemet C0805C106K9PACTU
5	1	Capacitor 68uF 10% 6.3V Tant 400 mOhm SMD 6032	C3	399-10513-1-ND	Kemet T495C686K006ATE400
6	2	Connector 3-Lug Concentric Triax Bayonet Jack, Panel Front Mount TRB (BJ77)	J1,J2 - See Note	MilesTek 10-06570	Trompeter Electronics BJ77 Use 0.469" Round Hole
7	3	Header, Male 2x20 0.1" Pitch	J3,J4,J5	S2012E-20-ND	Sullins PEC20DAAN
8	1	Header, 1x3, 0.1" pitch	J6	DO NOT STUFF	
9	1	Header, 1x7, 0.1" pitch	J8,J9	DO NOT STUFF	
10	1	IC, AND Gate 1 Chan SOT-23-5	U4	296-22345-1-ND	TI SN74LVC1G08MDBVREP
11	1	IC SRAM 2Mbit 45ns 44TSOP	U7	428-2068-ND	Cypress CY62136EV30LL-45ZSXI
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, 20ppm, 3.3V, SMD 5x7mm	OSC1	535-9330-1-ND	Abracon ASV-50.000MHZ-EJ-T
16	1	Resistor, 15 5% 1/8W 0805	R32	P150ACT-ND	Panasonic ERJ-6GEYJ150V
17	5	Resistor, 150 5% 1/8W 0805	R8,R9,R10,R11,R12	P150ACT-ND	Panasonic ERJ-6GEYJ151V
18	1	Resistor, 2.2K 5% 1/8W 0805	R36	P2.2KACT-ND	Panasonic ERJ-6GEYJ222V
19	2	Resistor, 10K 5% 1/8W 0805	R33, R34	P10KACT-ND	Panasonic ERJ-6GEYJ103V
20	8	Resistor, 47K 5% 1/8W 0805	R13,R14,R15 R16,R17,R18,R25,R35	P47KACT-ND	Panasonic ERJ-6GEYJ473V
21	2	DIP Switch 4-Position ThrHole	SW7, SW10	CT2094LPST-ND	CTS 209-4LPST
22	2	DIP Switch 5-Position ThrHole	SW1,SW4	CT2095LPST-ND	CTS 209-5LPST
23	2	DIP Switch 6-Position ThrHole	SW5,SW6	CT2096LPST-ND	CTS 209-6LPST
24	1	DIP Switch 10-Position ThrHole	SW8	CT20910LPST-ND	CTS 209-10LPST
25	1	Slide Switch SPDT SMD	SW9	563-1022-1-ND	Copal CJS-1200TB
26	2	Test Point, Red, 0.062" hole	TP4(Bus+),TP6(Bus+)	36-5010-ND	Keystone 5010
27	1	Test Point, Orange, 0.062" hole	TP10(3V3)	36-5013-ND	Keystone 5013
28	4	Test Point, Black, 0.062" hole	TP2(Gnd),TP3(Gnd), TP5(Bus-), TP7(Bus-)	36-5011-ND	Keystone 5011
29	1	Test Point, White, 0.062" hole	TP8 (Active)	36-5012-ND	Keystone 5012
30	1	IC HI-2130 Holt CTPF (121 PGA)	U1	Holt	Holt IC
31	2	IC, EEPROM 512Kbit 20MHz 8-SOIC	U5, U6	25LC512T-I/SNCT-ND	Microchip 25LC512T-I/SN
32	1	Hookup Solid wire - 20AWG - Black - 4" Long per Board	For J1 and J2	C2028B-XX-ND	General Cable C2028A.12.01

REMOTE TERMINAL RT1 MEMORY MAP FOR HI-6130 AND HI-6131 APPLICATION DEVELOPMENT BOARD PROGRAM

	dec	hex
Descriptor Table Base Addr	1024	0400
First Buffer Address	2048	0800

Descriptor Table Sector	Device Internal Addr same as HI-6131 Addr		Data Bus Addr Hex HI-6130 Only	
	Start	*End*	*Start*	*End*
Receive Subaddresses	0400	047F	60000800	600008FE
Transmit Subaddresses	0480	04FF	60000900	600009FE
Receive Mode Codes	0500	057F	60000A00	60000AFE
Transmit Mode Codes	0580	05FF	60000B00	60000BFE

Buffer Assignments for Receive and Transmit Subaddresses

Receive (Rx) Subaddress or Transmit (Tx) Subaddress	Buffer Method and Data Pointer(s)		Buffer Size *Words*	Device Internal Addr same as HI-6131 Addr *Start*	*End*	Data Bus Addr Hex HI-6130 Only *Start*	*End*	Structures Reserved *MIW = Msg Info Word* *TT = TimeTag Word*
Rx SA1	ping-pong	DPA	34	0800	0821	60001000	60001042	MIW + TT + 32 words
(data pointers A, B and		DPB	34	0822	0843	60001044	60001086	same
broadcast data pointer)		BDP	34	0844	0865	60001088	600010CA	same
Tx SA1	ping-pong	DPA	34	0866	0887	600010CC	6000110E	same
(data pointers A, B and		DPB	34	0888	08A9	60001110	60001152	same
broadcast data pointer)		BDP	4	08AA	08AD	60001154	6000115A	MIW + TT + 2 pad
Rx SA30 and Tx SA30 for data wrap-around	index-0	DPA	34	08AE	08CF	6000115C	6000119E	MIW + TT + 32 words
Rx SA2	index-32	DPA	1088	08D0	0D0F	600011A0	60001A1E	32 x (MIW + TT + 32 words)
		BDP	34	0D10	0D31	60001A20	60001A62	
Tx SA2	index-32	DPA	1088	0D32	1171	60001A64	600022E2	
		BDP	4	1172	1175	600022E4	600022EA	MIW + TT + 2 pad
Rx SA3	circ1-32	DPA	1088	1176	15B5	600022EC	60002B6A	32 x (MIW + TT + 32 words)
		pad	32	15B6	15D5	60002B6C	60002BAA	pad for overrun
Tx SA3	circ1-32	DPA	1088	15D6	1A15	60002BAC	6000342A	32 x (MIW + TT + 32 words)
		pad	32	1A16	1A35	6000342C	6000346A	pad for overrun
shared buffer: all unimplemented Rx subaddresses	index-0	DPA	34	1A36	1A57	6000346C	600034AE	MIW + TT + 32 words
shared buffer: all unimplemented Tx subaddresses	index-0	DPA	34	1A58	1A79	600034B0	600034F2	MIW + TT + 32 words
RAM assigned below (MCs)	---	---	142	1A7A	1B07	600034F4	6000360E	
unassigned RAM	---	---	72	1B08	1B4F	60003610	6000369E	
assigned to BC	---	---	176	1B50	1BFF	600036A0	600037FE	BC Mode Command Data BC Instruction List
Rx & Tx SA4	circ-2 256 msg max	MIB	512	1C00	1DFF	600036A0	60003BFE	256 x (MIW + TT)
		DPA	8192	1E00	3DFF	60003C00	60007BFE	
assigned to BC	---	---	256	3E00	3EFF	60007C00	60007DFE	BC Msg Control Blocks
unassigned RAM	---	---	256	3F00	3FFF	60007C00	60007FFE	

Shared Buffer Assignments for Undefined and Reserved Mode Code Commands

These RAM buffer allocations for mode code commands only apply when the application does not use the SMCP option.

Undefined & Reserved Receive (Rx) Mode Codes Transmit (Tx) Mode Codes	Buffer Method and Data Pointer(s)	Buffer Size *Words*	Device Internal Addr same as HI-6131 Addr *Start*	*End*	Data Bus Addr Hex HI-6130 Only *Start*	*End*	Structures Reserved *MIW = Msg Info Word* *TT = TimeTag Word*
shared buffer: undefined Rx MC0 - MC15	index-0 DPA	4	1A7A	1A7D	600034F4	600034FA	MIW + TT, 0 data, 2 pad
shared buffer: undefined Rx MC16, undefined Rx MC18 - MC19, reserved Rx MC22 - MC31	index-0 DPA	4	1A7E	1A81	600034FC	60003502	MIW + TT, 1 data, 1 pad
shared buffer: undefined Tx MC9 - MC15	index-0 DPA	4	1A82	1A85	00803504	6000350A	MIW + TT, 0 data, 2 pad
shared buffer: undefined Tx MC17, undefined Tx MC20 - MC21, reserved Tx MC22 - MC31	index-0 DPA	4	1A86	1A89	6000350C	60003512	MIW + TT, 1 data, 1 pad

Buffer Assignments for Defined Transmit Mode Code Commands MC0 - MC8 (No Data Word)

These RAM buffer allocations for mode code commands only apply when the application does not use the SMCP option.

Defined Transmit (Tx) Mode Code Commands, No Data	Buffer Method and Data Pointer(s)		Buffer Size *Words*	Device Internal Addr same as HI-6131 Addr *Start*	*End*	Data Bus Addr Hex HI-6130 Only *Start*	*End*	Structures Reserved *MIW = Msg Info Word* *TT = TimeTag Word*
Tx MC0	ping-pong	DPA	2	1A8A	1A8B	60003514	60003516	MIW + TT
		DPB	2	1A8C	1A8D	60003518	6000351A	same
		BDP	2	1A8E	1A8F	6000351C	6000351E	same
Tx MC1	ping-pong	DPA	2	1A90	1A91	60003520	60003522	MIW + TT
		DPB	2	1A92	1A93	60003524	60003526	same
		BDP	2	1A94	1A95	60003528	6000352A	same
Tx MC2	ping-pong	DPA	2	1A96	1A97	6000352C	6000352E	MIW + TT
		DPB	2	1A98	1A99	60003530	60003532	same
		BDP	2	1A9A	1A9B	60003534	60003536	same
Tx MC3	ping-pong	DPA	2	1A9C	1A9D	60003538	6000353A	MIW + TT
		DPB	2	1A9E	1A9F	6000353C	6000353E	same
		BDP	2	1AA0	1AA1	60003540	60003542	same
Tx MC4	ping-pong	DPA	2	1AA2	1AA3	60003544	60003546	MIW + TT
		DPB	2	1AA4	1AA5	60003548	6000354A	same
		BDP	2	1AA6	1AA7	6000354C	6000354E	same
Tx MC5	ping-pong	DPA	2	1AA8	1AA9	60003550	60003552	MIW + TT
		DPB	2	1AAA	1AAB	60003554	60003556	same
		BDP	2	1AAC	1AAD	60003558	6000355A	same
Tx MC6	ping-pong	DPA	2	1AAE	1AAF	6000355C	6000355E	MIW + TT
		DPB	2	1AB0	1AB1	60003560	60003562	same
		BDP	2	1AB2	1AB3	60003564	60003566	same
Tx MC7	ping-pong	DPA	2	1AB4	1AB5	60003568	6000356A	MIW + TT
		DPB	2	1AB6	1AB7	6000356C	6000356E	same
		BDP	2	1AB8	1AB9	60003570	60003572	same
Tx MC8	ping-pong	DPA	2	1ABA	1ABB	60003574	60003576	MIW + TT
		DPB	2	1ABC	1ABD	60003578	6000357A	same
		BDP	2	1ABE	1ABF	6000357C	6000357E	same

Buffer Assignments for Defined Transmit Mode Code Commands MC16, MC18 and MC19 (1 Data Word)

These RAM buffer allocations for mode code commands only apply when the application does not use the SMCP option.

Defined Transmit (Tx) Mode Code Commands with Data Word	Buffer Method and Data Pointer(s)		Buffer Size *Words*	Device Internal Addr same as HI-6131 Addr *Start*	*End*	Data Bus Addr Hex HI-6130 Only *Start*	*End*	Structures Reserved *MIW = Msg Info Word* *TT = TimeTag Word*
Tx MC16	ping-pong	DPA	4	1AC0	1AC3	60003580	60003586	MIW + TT, 1 data, 1 pad
		DPB	4	1AC4	1AC7	60003588	6000358E	same
		BDP	4	1AC8	1ACB	60003590	60003596	same
Tx MC18	ping-pong	DPA	4	1ACC	1ACF	60003598	6000359E	MIW + TT, 1 data, 1 pad
		DPB	4	1AD0	1AD3	600035A0	600035A6	same
		BDP	4	1AD4	1AD7	600035A8	600035AE	same
Tx MC19	ping-pong	DPA	4	1AD8	1ADB	600035B0	600035B6	MIW + TT, 1 data, 1 pad
		DPB	4	1ADC	1ADF	600035B8	600035BE	same
		BDP	4	1AE0	1AE3	600035C0	600035C6	same

Buffer Assignments for Defined Receive Mode Code Commands MC17, MC20 and MC21 (1 Data Word)

These RAM buffer allocations for mode code commands only apply when the application does not use the SMCP option.

Defined Receive (Rx) Mode Code Commands with Data Word	Buffer Method and Data Pointer(s)		Buffer Size *Words*	Device Internal Addr same as HI-6131 Addr *Start*	*End*	Data Bus Addr Hex HI-6130 Only *Start*	*End*	Structures Reserved *MIW = Msg Info Word* *TT = TimeTag Word*
Rx MC17	ping-pong	DPA	4	1AE4	1AE7	600035C8	600035CE	MIW + TT, 1 data, 1 pad
		DPB	4	1AE8	1AEB	600035D0	600035D6	same
		BDP	4	1AEC	1AEF	600035D8	600035DE	same
Rx MC20	ping-pong	DPA	4	1AF0	1AF3	600035E0	600035E6	MIW + TT, 1 data, 1 pad
		DPB	4	1AF4	1AF7	600035E8	600035EE	same
		BDP	4	1AF8	1AFB	600035F0	600035F6	same
Rx MC21	ping-pong	DPA	4	1AFC	1AFF	600035F8	600035FE	MIW + TT, 1 data, 1 pad
		DPB	4	1B00	1B03	60003600	60003606	same
		BDP	4	1B04	1B07	60003608	6000360E	same

Notes:

1. All addresses shown are expressed as hexadecimal values.

2. Addressing for HI-6131 uses device internal addresses. Bus addressing for HI-6130 is offset by chip select base address 0x60000000 and microprocessor uses byte addressing so all address offsets are doubled. (The LSB becomes upper/lower byte select for each word.)

3. Memory allocations are shared for undefined and reserved mode code commands, and unimplemented subaddress commands, These commands are grouped by like requirements, and share common RAM resources (bit bucket).

4. For messages needing an odd number of words, an extra "pad" word is added so the next buffer begins at an even address.

5. Subaddresses using circular buffer Mode 1 are followed by a 32-word overrun buffer, in case a 32 data word receive command arrives with just one location remaining before "buffer full" attainment.

REMOTE TERMINAL RT2 MEMORY MAP FOR HI-6130 AND HI-6131 APPLICATION DEVELOPMENT BOARD PROGRAM

	dec	hex
Descriptor Table Base Addr	1536	0600
First Buffer Address	16384	4000

Descriptor Table Sector	Device Internal Addr same as HI-6131 Addr		Data Bus Addr Hex HI-6130 Only	
	Start	*End*	*Start*	*End*
Receive Subaddresses	0600	067F	60000C00	60000CFE
Transmit Subaddresses	0680	06FF	60000D00	60000DFE
Receive Mode Codes	0700	077F	60000E00	60000EFE
Transmit Mode Codes	0780	07FF	60000F00	60000FFE

Buffer Assignments for Receive and Transmit Subaddresses

Receive (Rx) Subaddress or Transmit (Tx) Subaddress	Buffer Method and Data Pointer(s)		Buffer Size *Words*	Device Internal Addr same as HI-6131 Addr *Start*	*End*	Data Bus Addr Hex HI-6130 Only *Start*	*End*	Structures Reserved *MIW = Msg Info Word* *TT = TimeTag Word*
Rx SA1	ping-pong	DPA	34	4000	4021	60008000	60008042	MIW + TT + 32 words
(data pointers A, B and		DPB	34	4022	4043	60008044	60008086	same
broadcast data pointer)		BDP	34	4044	4065	60008088	600080CA	same
Tx SA1	ping-pong	DPA	34	4066	4087	600080CC	6000810E	same
(data pointers A, B and		DPB	34	4088	40A9	60008110	60008152	same
broadcast data pointer)		BDP	4	40AA	40AD	60008154	6000815A	MIW + TT + 2 pad
Rx SA30 and Tx SA30 for data wrap-around	index-0	DPA	34	40AE	40CF	6000815C	6000819E	MIW + TT + 32 words
Rx SA2	index-32	DPA	1088	40D0	450F	600081A0	60008A1E	32 x (MIW + TT + 32 words)
		BDP	34	4510	4531	60008A20	60008A62	
Tx SA2	index-32	DPA	1088	4532	4971	60008A64	600092E2	
		BDP	4	4972	4975	600092E4	600092EA	MIW + TT + 2 pad
Rx SA3	circ1-32	DPA	1088	4976	4DB5	600092EC	60009B6A	32 x (MIW + TT + 32 words)
		pad	32	4DB6	4DD5	60009B6C	60009BAA	pad for overrun
Tx SA3	circ1-32	DPA	1088	4DD6	5215	60009BAC	6000A42A	32 x (MIW + TT + 32 words)
		pad	32	5216	5235	6000A42C	6000A46A	pad for overrun
shared buffer: all unimplemented Rx subaddresses	index-0	DPA	34	5236	5257	6000A46C	6000A4AE	MIW + TT + 32 words
shared buffer: all unimplemented Tx subaddresses	index-0	DPA	34	5258	5279	6000A4B0	6000A4F2	MIW + TT + 32 words
RAM assigned below (MCs)	---	---	142	527A	5307	6000A4F4	6000A60E	
assigned to BC	---	---	248	5308	53FF	6000A610	6000A7FE	BC Msg Data Buffers (excl mode commands)
SA4 not used by RT2								
assigned to MT	---	---	11264	5400	7FFF	6000A800	6000FFFE	IMT Stack or SMT Stacks

Shared Buffer Assignments for Undefined and Reserved Mode Code Commands

These RAM buffer allocations for mode code commands only apply when the application does not use the SMCP option.

Undefined & Reserved Receive (Rx) Mode Codes Transmit (Tx) Mode Codes	Buffer Method and Data Pointer(s)	Buffer Size *Words*	Device Internal Addr same as HI-6131 Addr *Start*	*End*	Data Bus Addr Hex HI-6130 Only *Start*	*End*	Structures Reserved *MIW = Msg Info Word* *TT = TimeTag Word*
shared buffer: undefined Rx MC0 - MC15	index-0 DPA	4	527A	527D	6000A4F4	6000A4FA	MIW + TT, 0 data, 2 pad
shared buffer: undefined Rx MC16, undefined Rx MC18 - MC19, reserved Rx MC22 - MC31	index-0 DPA	4	527E	5281	6000A4FC	6000A502	MIW + TT, 1 data, 1 pad
shared buffer: undefined Tx MC9 - MC15	index-0 DPA	4	5282	5285	6000A504	6000A50A	MIW + TT, 0 data, 2 pad
shared buffer: undefined Tx MC17, undefined Tx MC20 - MC21, reserved Tx MC22 - MC31	index-0 DPA	4	5286	5289	6000A50C	6000A512	MIW + TT, 1 data, 1 pad

Buffer Assignments for Defined Transmit Mode Code Commands MC0 - MC8 (No Data Word)

These RAM buffer allocations for mode code commands only apply when the application does not use the SMCP option.

Defined Transmit (Tx) Mode Code Commands, No Data	Buffer Method and Data Pointer(s)	Buffer Size *Words*	Device Internal Addr same as HI-6131 Addr *Start*	*End*	Data Bus Addr Hex HI-6130 Only *Start*	*End*	Structures Reserved *MIW = Msg Info Word* *TT = TimeTag Word*
Tx MC0	ping-pong DPA	2	528A	528B	6000A514	6000A516	MIW + TT
	DPB	2	528C	528D	6000A518	6000A51A	same
	BDP	2	528E	528F	6000A51C	6000A51E	same
Tx MC1	ping-pong DPA	2	5290	5291	6000A520	6000A522	MIW + TT
	DPB	2	5292	5293	6000A524	6000A526	same
	BDP	2	5294	5295	6000A528	6000A52A	same
Tx MC2	ping-pong DPA	2	5296	5297	6000A52C	6000A52E	MIW + TT
	DPB	2	5298	5299	6000A530	6000A532	same
	BDP	2	529A	529B	6000A534	6000A536	same
Tx MC3	ping-pong DPA	2	529C	529D	6000A538	6000A53A	MIW + TT
	DPB	2	529E	529F	6000A53C	6000A53E	same
	BDP	2	52A0	52A1	6000A540	6000A542	same
Tx MC4	ping-pong DPA	2	52A2	52A3	6000A544	6000A546	MIW + TT
	DPB	2	52A4	52A5	6000A548	6000A54A	same
	BDP	2	52A6	52A7	6000A54C	6000A54E	same
Tx MC5	ping-pong DPA	2	52A8	52A9	6000A550	6000A552	MIW + TT
	DPB	2	52AA	52AB	6000A554	6000A556	same
	BDP	2	52AC	52AD	6000A558	6000A55A	same
Tx MC6	ping-pong DPA	2	52AE	52AF	6000A55C	6000A55E	MIW + TT
	DPB	2	52B0	52B1	6000A560	6000A562	same
	BDP	2	52B2	52B3	6000A564	6000A566	same
Tx MC7	ping-pong DPA	2	52B4	52B5	6000A568	6000A56A	MIW + TT
	DPB	2	52B6	52B7	6000A56C	6000A56E	same
	BDP	2	52B8	52B9	6000A570	6000A572	same
Tx MC8	ping-pong DPA	2	52BA	52BB	6000A574	6000A576	MIW + TT
	DPB	2	52BC	52BD	6000A578	6000A57A	same
	BDP	2	52BE	52BF	6000A57C	6000A57E	same

Buffer Assignments for Defined Transmit Mode Code Commands MC16, MC18 and MC19 (1 Data Word)

These RAM buffer allocations for mode code commands only apply when the application does not use the SMCP option.

Defined Transmit (Tx) Mode Code Commands with Data Word	Buffer Method and Data Pointer(s)		Buffer Size *Words*	Device Internal Addr same as HI-6131 Addr *Start*	*End*	Data Bus Addr Hex HI-6130 Only *Start*	*End*	Structures Reserved *MIW = Msg Info Word* *TT = TimeTag Word*
Tx MC16	ping-pong	DPA	4	52C0	52C3	6000A580	6000A586	MIW + TT, 1 data, 1 pad
		DPB	4	52C4	52C7	6000A588	6000A58E	same
		BDP	4	52C8	52CB	6000A590	6000A596	same
Tx MC18	ping-pong	DPA	4	52CC	52CF	6000A598	6000A59E	MIW + TT, 1 data, 1 pad
		DPB	4	52D0	52D3	6000A5A0	6000A5A6	same
		BDP	4	52D4	52D7	6000A5A8	6000A5AE	same
Tx MC19	ping-pong	DPA	4	52D8	52DB	6000A5B0	6000A5B6	MIW + TT, 1 data, 1 pad
		DPB	4	52DC	52DF	6000A5B8	6000A5BE	same
		BDP	4	52E0	52E3	6000A5C0	6000A5C6	same

Buffer Assignments for Defined Receive Mode Code Commands MC17, MC20 and MC21 (1 Data Word)

These RAM buffer allocations for mode code commands only apply when the application does not use the SMCP option.

Defined Receive (Rx) Mode Code Commands with Data Word	Buffer Method and Data Pointer(s)		Buffer Size *Words*	Device Internal Addr same as HI-6131 Addr *Start*	*End*	Data Bus Addr Hex HI-6130 Only *Start*	*End*	Structures Reserved *MIW = Msg Info Word* *TT = TimeTag Word*
Rx MC17	ping-pong	DPA	4	52E4	52E7	6000A5C8	6000A5CE	MIW + TT, 1 data, 1 pad
		DPB	4	52E8	52EB	6000A5D0	6000A5D6	same
		BDP	4	52EC	52EF	6000A5D8	6000A5DE	same
Rx MC20	ping-pong	DPA	4	52F0	52F3	6000A5E0	6000A5E6	MIW + TT, 1 data, 1 pad
		DPB	4	52F4	52F7	6000A5E8	6000A5EE	same
		BDP	4	52F8	52FB	6000A5F0	6000A5F6	same
Rx MC21	ping-pong	DPA	4	52FC	52FF	6000A5F8	6000A5FE	MIW + TT, 1 data, 1 pad
		DPB	4	5300	5303	6000A600	6000A606	same
		BDP	4	5304	5307	6000A608	6000A60E	same

Notes:

1. All addresses shown are expressed as hexadecimal values.

2. Addressing for HI-6131 uses device internal addresses. Bus addressing for HI-6130 is offset by chip select base address 0x60000000 and microprocessor uses byte addressing so all address offsets are doubled. (The LSB becomes upper/lower byte select for each word.)

3. Memory allocations are shared for undefined and reserved mode code commands, and unimplemented subaddress commands, These commands are grouped by like requirements, and share common RAM resources (bit bucket).

4. For messages needing an odd number of words, an extra "pad" word is added so the next buffer begins at an even address.

5. Subaddresses using circular buffer Mode 1 are followed by a 32-word overrun buffer, in case a 32 data word receive command arrives with just one location remaining before "buffer full" attainment.

BUS CONTROLLER MEMORY MAP FOR HI-6130 AND HI-6131 APPLICATION DEVELOPMENT BOARD PROGRAM

BC Message Blocks
used in application development kit program

Block Number	Command Type	# Block Words	Block Start Addr	Block End Addr	HI-6130 Bus Addr
1	Tx SA *	8	3E00	3E07	60007C00
2	Tx SA *	8	3E08	3E0F	60007C10
3	Rx SA	8	3E10	3E17	60007C20
4	B Rx SA	8	3E18	3E1F	60007C30
5	B Rx SA	8	3E20	3E27	60007C40
6	Tx MC2 ND	8	3E28	3E2F	60007C50
7	Tx MC18 D	8	3E30	3E37	60007C60
8	Rx MC21 D	8	3E38	3E3F	60007C70
RTRT1	RTRT	16	3E40	3E4F	60007C80
RTRT2	B RTRT	16	3E50	3E5F	60007CA0
	available for expansion through end addr...				
		160		3EFF	60007DFE

Corresponding BC Message Data Buffers
used in application development kit program

Number of Words	Buffer Start Addr	Buffer End Addr	HI-6130 Bus Addr
32	5308	5327	6000A610
32	5308	5327	6000A610
32	5328	5347	6000A650
32	5348	5367	6000A690
32	5368	5387	6000A6D0
0	no data	no data	no data
1	1B62	----	600036C6
1	1B55	----	600036AC
32	5388	53A7	6000A710
32	53A8	53C7	6000A750
available for expansion through end addr...			
56		53FF	6000A7FE

* These 2 message blocks are Transmit Subaddress commands to the same subaddress, so use same Tx buffer.

BC Fixed Mode Command Data Word Storage
used in application development kit program

	Mode Code Cmd	# Data Words	Mode Cmd Data Addr		HI-6130 Bus Addr
Receive Mode Code Commands with Data	RxMC 16	1	1B50		600036A0
	RxMC 17	1	1B51		600036A2
	RxMC 18	1	1B52		600036A4
	RxMC 19	1	1B53		600036A6
	RxMC 20	1	1B54		600036A8
	RxMC 21	1	1B55		600036AA
	RxMC 22	1	1B56		600036AC
	RxMC 23	1	1B57		600036AE
	RxMC 24	1	1B58		600036B0
	RxMC 25	1	1B59		600036B2
	RxMC 26	1	1B5A		600036B4
	RxMC 27	1	1B5B		600036B6
	RxMC 28	1	1B5C		600036B8
	RxMC 29	1	1B5D		600036BA
	RxMC 30	1	1B5E		600036BC
	RxMC 31	1	1B5F		600036BE
Transmit Mode Code Commands with Data	TxMC 16	1	1B60		600036C0
	TxMC 17	1	1B61		600036C2
	TxMC 18	1	1B62		600036C4
	TxMC 19	1	1B63		600036C6
	TxMC 20	1	1B64		600036C8
	TxMC 21	1	1B65		600036CA
	TxMC 22	1	1B66		600036CC
	TxMC 23	1	1B67		600036CE
	TxMC 24	1	1B68		600036D0
	TxMC 25	1	1B69		600036D2
	TxMC 26	1	1B6A		600036D4
	TxMC 27	1	1B6B		600036D6
	TxMC 28	1	1B6C		600036D8
	TxMC 29	1	1B6D		600036DA
	TxMC 30	1	1B6E		600036DC
	TxMC 31	1	1B6F		600036DE

BC Instruction List Addresses
used in application development kit program

Op Code #	Op Code Addr	Msg Block called	HI-6130 Bus Addr
0	1B70	op WTG	600036E0
2	1B72	1	600036E4
4	1B74	op WTG	600036E8
6	1B76	2	600036EC
8	1B78	op WTG	600036F0
10	1B7A	3	600036F4
12	1B7C	op WTG	600036F8
14	1B7E	4	600036FC
16	1B80	op WTG	60003700
18	1B82	5	60003704
20	1B84	op WTG	60003708
22	1B86	6	6000370C
24	1B88	op WTG	60003710
26	1B8A	7	60003714
28	1B8C	op WTG	60003718
30	1B8E	8	6000371C
32	1B90	op WTG	60003720
34	1B92	RTRT1	60003724
36	1B94	op WTG	60003728
38	1B96	RTRT2	6000372C
40	1B98	op WTG	60003730
42	1B9A	2	60003734
44	1B9C	op JMP	60003738
46	1B9E		6000373C
48	1BA0	*Execute op codes can call Message Blocks in any order!*	60003740
50	1BA2		60003744
52	1BA4		60003748
54	1BA6		6000374C
56	1BA8		60003750
58	1BAA		60003754
60	1BAC		60003758
62	1BAE		6000375C
available for expansion through end addr...			
142	1BFE		600037FC

Notes:
1. Command Types: SA = Subaddress cmd, MC = Mode Code cmd, ND = no data, D = with data, B = broadcast.
2. All 4-digit hexadecimal addresses refer to the internal IC address, equal to the address used by HI-6131 SPI.
3. The HI-6130 Bus Address = ARM MCU chip select base addr 0x60000000 + 2 x (feature's IC address)

MISCELLANEOUS RAM STRUCTURES NOT ALREADY LISTED

RAM Structure	Start Address	End Address	Number of Words
Interrupt Log Buffer	0x0180	0x01BF	64
Bus Controller General Purpose Queue	0x00C0	0x00FF	64
Bus Controller Call Stack	0x0054	0X005B	8
RT1 Temporary Receive Buffer	0x01C0	0x01DF	32
RT2 Temporary Receive Buffer	0x01E0	0x01FF	32
RT1 Command Ilegalization Table	0x0200	0x02FF	256
RT2 Command Ilegalization Table	0x0300	0x03FF	256
SMT or IMT Message Filter Table	0x0100	0x017F	128
SMT or IMT Address List	0x00B0	0x00B7	8
SMT Command Stack	0x5400	0x5FFF	3072
SMT Data Stack	0x6000	0x7FFF	24577
IMT Combined Stack	0x5400	0x6400	6400