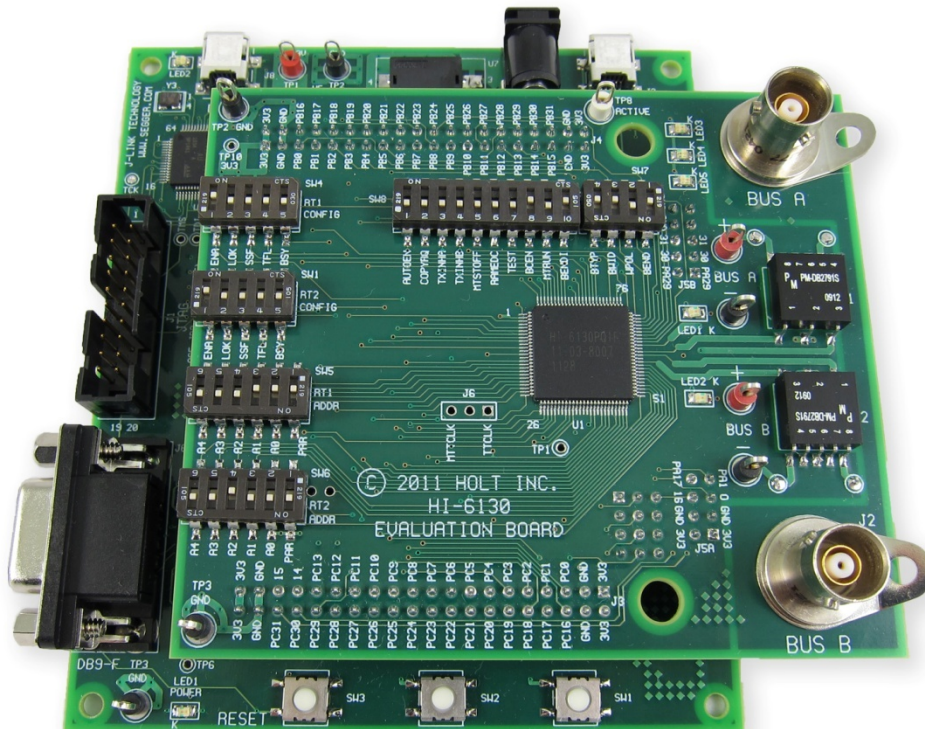


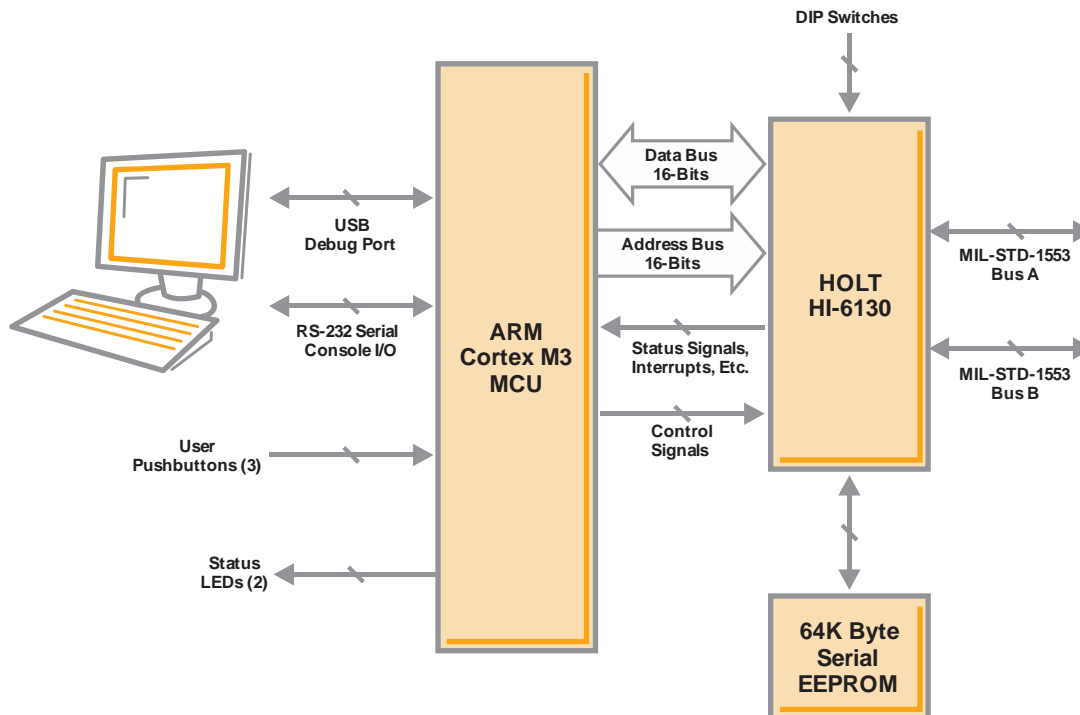
## Introduction

This application note is a software design guide for initializing and configuring the Holt MIL-STD-1553 BC/MT/RT Multi-Terminal device as a single Remote Terminal. This application note focuses on the software needed for RT1 initialization. The separate “AN-6130 Application Development Kit” document provides essential information on the HI-6130 evaluation board, software project and development tools. AN-6130 should be reviewed before reading this document. The AN-6130 application note is found in the evaluation kit CD-ROM, or can be found on the Holt website, <http://www.holtic.com>



## HI-6130 System Block Diagram

The HI-6130 integrates a BC, two RT's and two types of bus monitor (MT) on a single 3.3V IC with integrated dual redundant bus transceivers. The BC/RT/MT protocol state machines run independently of each other and can be enabled or disabled by the host-programmed configuration. The HI-6130 bus interface appears as a 32K word memory-mapped device to the host processor. The first 80 word addresses are reserved for system configuration registers; the remaining space is used by the device and host for Control Blocks, state machine flags, tables and subaddress (SA) message buffers. The HI-6130 memory map in Table 1 shows the partitions. The HI-6130 interfaces to an FPGA or host processor using 16 data lines, 16 address lines, /CS (chip select) and a handful of control signals for register/RAM read/write and several signals for external interrupts. The HI-6131 is also offered, with SPI host interface instead of parallel bus. Both devices require a 50MHz system clock. For detailed information on either device, refer to the HI-6130 data sheet.

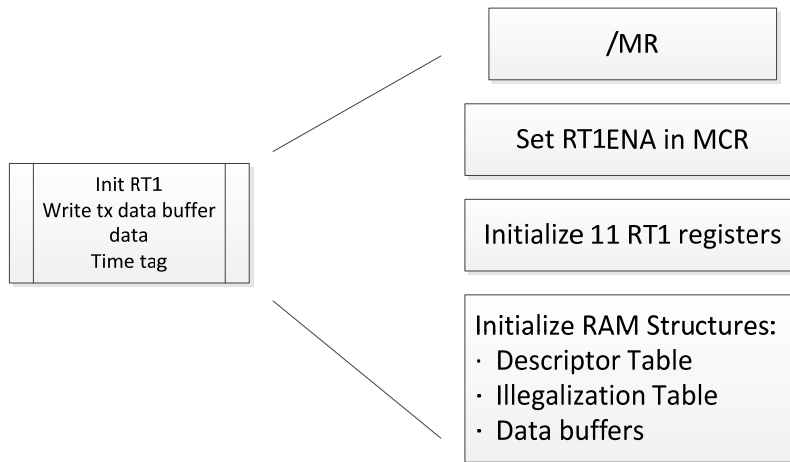


## HI-6130 RT Initialization

The host initializes the BC, MT or RT by configuring a set of system registers and shared RAM structures. This application note focuses on initializing Remote Terminal RT1, but the demo project accompanying this application note also provides easily selectable configurations used by the IAR Systems *Embedded Workbench*® for ARM C programmer's tool set rebuild the project to include any or all of the three 1553 terminal types. Any combination of 1553 terminal devices (BC, MT and up to two RTs) can be enabled for concurrent operation. The AN-6130 Application Development Kit document provides information on the project configurations and how to select them.

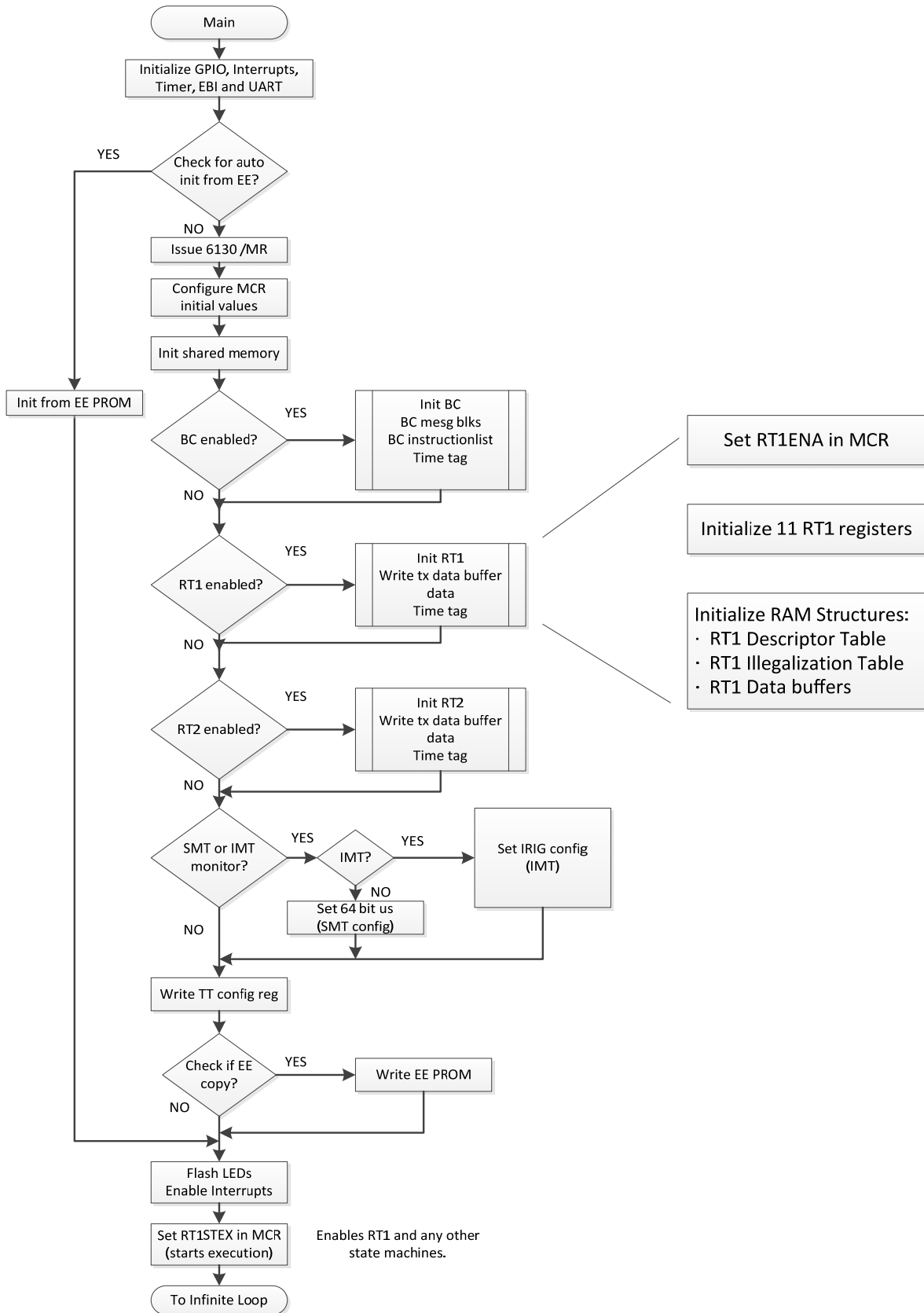
The following summarized steps are used to configure the HI-6130 device for RT1 only. This is a subset of the demo project that is shown on the `main.c` flow chart on the next page. A detailed 17-step sequence is described in the section, "HI-613x Demo RT1 Configuration".

- 1) The host issues a `/MR` pulse then waits for READY assertion (self-test completion),
- 2) The host sets RT1ENA (Remote Terminal 1 Enable) bit6 in the Master Control Register (MCR). The RT1ENA input pin should already be high as part of the board configuration.
- 3) The host sets eleven RT1 configuration registers, steps 2- 17 in the section, "HI-6130 Demo RT1 Configuration".
- 4) The host initializes shared RAM structures for RT1 Descriptor Table, RT1 command Illegalization Table and subaddress data buffers.
- 5) The host sets the RT1STEX in the Master Control Register (MCR) after all other initialization, starting RT1 terminal execution.



This initialization sequence describes the Simple RT demo program. These steps will vary according to actual use requirements. The `main.c` flow chart shows the initialization blocks for the `BC_SMT_RT` configuration enabling all state machines: BC, MT, and both RT's.

MAIN.C FLOW CHART



## Shared RAM Structures

To configure RAM structures for a single RT, the host initializes specific ranges in HI-6130 RAM to set up the RT1 Descriptor Table, RT1 Illegalization Table and may initialize transmit message data buffers used by the RT.

The RT1 Descriptor Table is comprised of four word Control Blocks. One block is used for each transmit or receive subaddress, and one block is used for each transmit or receive mode code command. Each descriptor Control Block consists of four words. The first word in each block is always the Control Word, while function for the remaining three words depends on the data buffer type selected in the Control Word.

By default, the RT1 Descriptor Table resides in the RAM address range 0x0400 to 0x05FF. The RT1 Descriptor Table can be relocated by writing the RT1 Descriptor Table Base Address register.

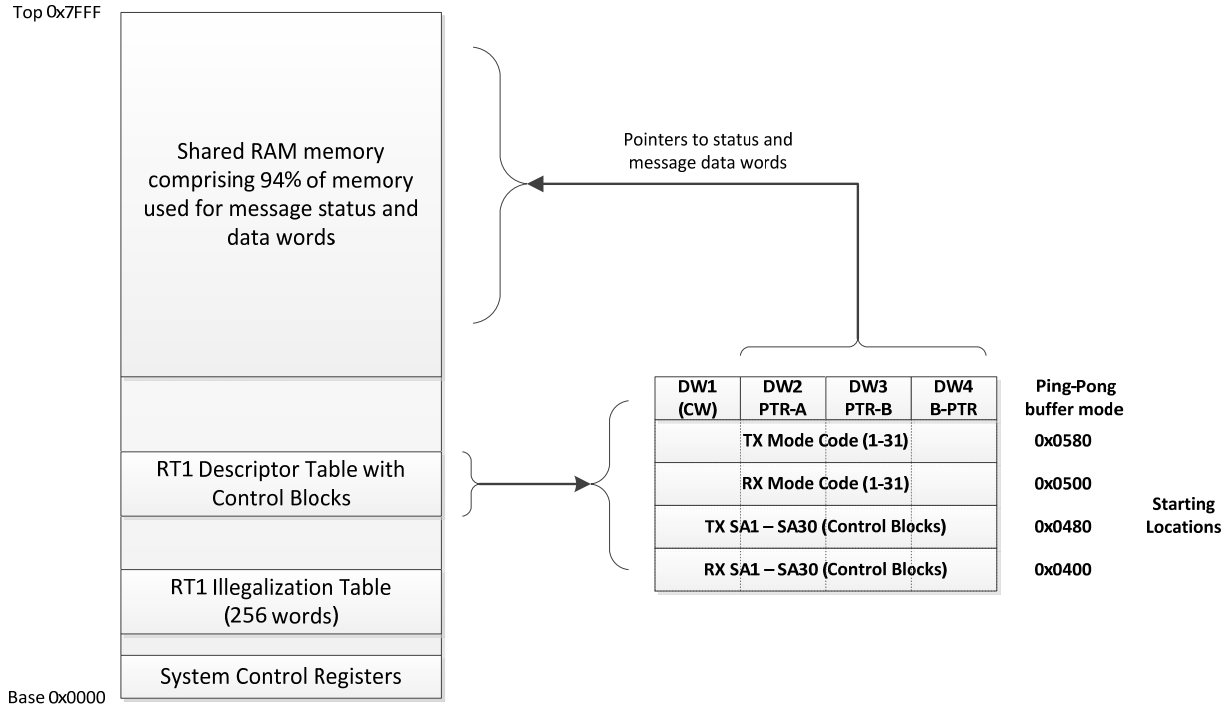
### Descriptor Table Overview

The RT Descriptor Table is partitioned into quadrants organized by message type, shown below. The default quadrant start addresses for RT1 are shown. Please refer to the HI-6130 data sheet for detailed information on using the RT Descriptor Table.

RT Transmit Mode Code Commands, Tx MC0 - MC31 (occurring when command word subaddress is 0 or 31)	Address Range: 0x580 - 0x5FE
RT Receive Mode Code Commands, Rx MC0 - MC31 (occurring when command word subaddress is 0 or 31)	Address Range: 0x500 – 0x57E
RT Transmit Subaddress Commands, Tx SA1 - SA30	Address Range: 0x480 – 0x4FE
RT Receive Subaddress Commands, Rx SA1 - SA30	Address Range: 0x400 – 0x47E

# AN6130 DG

This diagram shows the memory relationship between the RT1 data structures in shared RAM. The complete memory map is found in HI-6130 data sheet Figure 2 entitled, "Address Mapping for Registers and RAM".



For each transmit or receive subaddress (or mode code command), a specific descriptor Control Word selects the applied RAM data buffer scheme. The Control Word also enables interrupts and provides limited status information that is updated after message transaction. For example, when a subaddress descriptor block is accessed, the DBAC (Descriptor Block Accessed) bit is set; the host can poll the DBAC bit to determine whether the corresponding command was issued by the BC and transacted by the RT.

## One Control Block

Control Word (command type)	Descriptor Word 2 Data Pointer A	Descriptor Word 3 Data Pointer B	Descriptor Word 4 Broadcast Data Pointer
-----------------------------	----------------------------------	----------------------------------	--

## Receive Subaddress Control Word



- H Bit maintained by host
- D Bit maintained by device
- X Bit set by device, reset by host read cycle

## AN6130 DG

### RAM Data Buffer Schemes:

There are four message buffer schemes selectable by writing bits 2-0 in the descriptor Control Word. These bits can select double-buffered Ping-Pong mode, single-buffered Index mode, Circular Mode 1 or Circular Mode 2.

Function for Descriptor Words 2 and 3 is determined by the buffer scheme set by Control Word bits 2-0. The Broadcast Data Pointer is exclusively used for broadcast commands, so data from broadcast and non-broadcast receive commands can be stored in separate buffers, to comply with MIL-STD 1553B Notice II.

### Ping-Pong Mode Data Buffering

Ping-Pong mode uses two data pointers designated A and B. For successive messages to the subaddress, device RT logic alternately transmits data from (or stores received data into) these two starting locations. The descriptor Control Word DPB bit is controlled by device logic to indicate the pointer used for the next-occurring command to the subaddress. If DPB is high, the next message uses the buffer start address indicated by Data Pointer B. If DPB is low, the next message uses the Data Pointer A address.

#### Ping-Pong Control Block

Control Word (command type)	Descriptor Word 2 Data Pointer A	Descriptor Word 3 Data Pointer B	Descriptor Word 4 Broadcast Data Pointer
--------------------------------	-------------------------------------	-------------------------------------	---

### Index Mode Data Buffering

Indexed mode exclusively uses Data Pointer A for all non-broadcast messages. When Notice II storage option is enabled, data from broadcast receive messages is stored separately (at the address indicated by Descriptor Word 4). When the Notice II option is not enabled, broadcast and non-broadcast receive data is stored at the buffer address indicated by Data Pointer A.

The Index value in Descriptor Word 3 is initialized with an integer number of messages known as the Index. The initial Index value indicates a number of subaddress messages for which message data will be sequentially stored in the same buffer. When Index is initialized with 2 or more, the buffer can accumulate much more than 32 words for bulk data transfers. Upon error-free completion of each subaddress message, Data Pointer A advances by 2+N where N is the number of words transacted by the message. The 2 additional words are the stored Message Information and Time Tag words. The Index count decrements. When Index decrements from 1 to zero, an interrupt can be enabled (buffer full). The host should service the buffer then reinitialize the Index and Data Pointer A for continued reuse of the buffer. When the Index value is zero, subsequent messages repeatedly overwrite the last-used buffer range, reverting to Indexed Single Message Mode, described next.

#### Indexed Control Block

Control Word (command type)	Descriptor Word 2 Data Pointer A	Descriptor Word 3 Index count	Descriptor Word 4 Broadcast Data Pointer
--------------------------------	-------------------------------------	----------------------------------	---

## Indexed Single Message Buffer Mode

In Indexed Single Message Mode, the subaddress uses a fixed Data Pointer A start location for all messages. This mode is initialized by writing the fixed Data Pointer A address, and an Index of zero in Descriptor Word 3. Usually 34 words of buffer RAM are allocated, using 32 words for data and two words for the stored Message Information and Time Tag words. This is the most basic type of buffering scheme, but requires timely buffer servicing by the RT external host.

Indexed Single Message Mode may accidentally occur when normal multi-message Index is initialized with non-zero integer Index value, N, but the subaddress has already transacted N messages. Behavior reverts to Indexed Single Message mode if additional messages are received to the same subaddress before the Index N and original Data Pointer A value are reinitialized.

As long as Index N equals zero at completion of a new message, newly received data simply overwrites the address range from the previous message (for receive subaddresses) or the RT re-transmits data from the same address range as the previous message (for transmit subaddresses). With Index already zero, no post-message N-decrement occurs, and Data Pointer A continues to point at the starting buffer address as the last subaddress message.

## Circular Buffer Modes 1 and 2

The demo program configures subaddress SA3 for Circular Buffer Mode 1 and SA4 for Circular Buffer Mode 2. Refer to the HI-6130 data sheet for a detailed description of these modes.

## Command Illegalization Table

Optional “Illegal Command Detection” allows illegalization for unimplemented receive or transmit subaddresses (or word counts within subaddresses) as well as unimplemented, reserved or undefined Mode Code commands. The RT responds to illegal commands with a “Message Error” Status Word, and no data words are transacted.

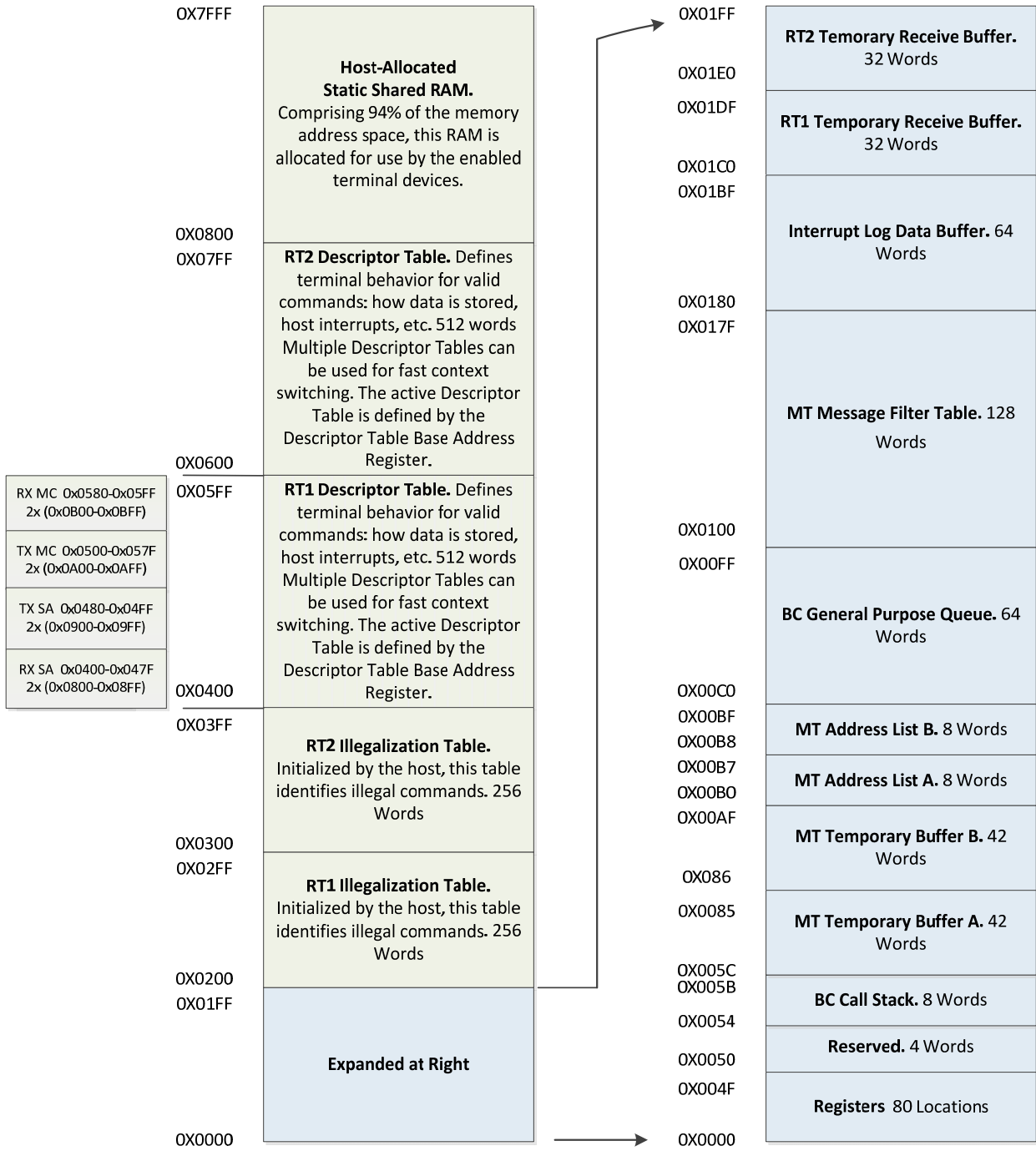
By setting appropriate bits in the command Illegalization Table, valid subaddress commands can be made illegal based on subaddress number, transmit/receive and the number of data words in the message. Mode code commands can be made illegal based on transmit/receive bit and mode code value.

To globally enable all messages (turning off illegal command detection) all memory is cleared in the default 256-word Illegalization Table address range. The default address range for RT1 is 0x0200–0x02FF while the default range for RT2 is 0x0300–0x03FF. After Master Reset rising edge, all HI-6130 RAM is reset to 0x0000, defaulting an enabled RT1 to “illegal command detection” OFF condition. See the HI-6130 data sheet for the format, bit fields and usage of this table. Step 15 shows how the host initializes the initialization table for this demo project.



# AN6130 DG

## HI-6130 MEMORY MAP



Address Mapping for Registers and RAM

## HI-6130 Demo RT1 Configuration

This is the host sequence to initialize the RT1 system registers shown in the flow chart block: “Init RT1”. For a complete definition of all the HI-6130 register bits, see the HI-6130 data sheet.

### Step 1: Master Reset

The first HI-6130 initialization step is applying Master Reset. The /MR pin should be pulsed low for at least 100ns. A 50MHz clock should already be applied and the RT1ENA input pin should be asserted beforehand. After Master Reset rising edge, the host awaits READY pin assertion before continuing to Step 2.

### Step 2: RT1 Master Configuration Register (0x0000)

The host writes the initial value to this register to enable on-chip 1553 terminal devices: BC, RT1, RT2 and MT. For this demo project, a final write to this register sets the RT1STEX bit high to begin RT1 terminal execution. This occurs in Step 17 after all other registers and RAM structures have been initialized.

### Step 3: RT1 CONFIG REG (0x0017).

The host writes to this register to configure the RT device and select various options. For normal operation, both RTINHA and RTINHB should be reset; otherwise transmission onto the 1553 buses will be inhibited. The remaining register bits control RT options; their use may depend on the end-system requirements. Some of these bits control how the RT handles Broadcast commands, optional Notice II broadcast data storage, automatic bus shutdown for “Shutdown Bus” mode codes MC4 and MC20, as well as several other mode command options.

### Step 4: Hardware Interrupt Enable Register (0x000F).

The host writes to this register to enable interrupts from several possible sources such as initialization errors (EECHK and RAMERR), self-test errors (Loopback Fail), RT Address Parity Fail, and time tag counter rollover or count match for BC, RT1, RT2 and MT time tag counters.

### Step 5: Hardware Pending Interrupt Register (0x0006).

These read-only registers align with the upper 12 bit positions of the Hardware Interrupt Enable Register. The lower three bits 2:0 simplify detection of RT, MT and BC interrupts. These bits can be read by the host to determine whether pending interrupts have occurred for the BC, MT or RT terminals, if enabled. When the host reads this register, it automatically clears to 0x0000.

### Step 6: Hardware Interrupt Output Enable Register (0x0013).

These registers align with the upper 12 bits of the Hardware Interrupt Enable register. These bits enable the associated interrupt to drive the /IRQ output pin.

### **Step 7: RT1 Descriptor Table Base Address Register (0x0019).**

This register sets the starting address for the RT1 Descriptor Table. This demo project uses the post-reset default value, 0x0400 seen in the device memory map. The software for the HI-6130 reflects a different address, the sum of the left-shifted 0x0400 value (0x0800) plus the address offset for the MCU chip select (0x60000000) that is connected to the HI-6130.

### **Step 8: RT1 MIL-STD-1553 Status Word Bits Register (0x001A).**

Bits 10 – 0 are reflected in the outgoing MIL-STD-1553 RT status word. For “Clear Status” the host restores the post-reset default value of this register, 0x0000. Register bits 4 and 10, BCR (Broadcast Command Received) and ME (Message Error) are maintained by the device. The status word RT Address bits 15-11 are automatically provided by the device when an RT Status Word is transmitted.

### **Step 9: RT1 TTAG Utility Register (0x004A).**

This register is maintained by the host and serves as a Time Tag holding register. When used with the RT Time Tag Counter Configuration Register, the value in this register is loaded into the Time Tag Counter (0x0049).

### **Step 10: RT1 Bus A Select Register (0x001C).**

The host loads this register to specify a "bus A select" value for mode code commands 20 and 21. The default value after power-on reset is 0xAAAA. This register is exclusively used for automated Transmit Shutdown and Shutdown Override compliance to the receive mode codes 20 and 21, when bit 4 is set in the RT Configuration Register. This is not exercised in this demo project.

### **Step 11: RT1 Bus B Select Register (0x001D).**

The host loads this register to specify a "bus B select" value for mode code commands 20 and 21. The default value after power-on reset is 0BBBBB. This register is exclusively used for automated Transmit Shutdown and Shutdown Override compliance to the receive mode codes 20 and 21, when bit 4 is set in the RT Configuration Register. This is not exercised in this demo project.

### **Step 12: RT1 BIT (Built-InTest) Word Register (0x001E).**

This register contains a set of predefined status bits and result bits for a range of host-managed RAM and loopback tests. Several bits are user-defined and may be written by the host to convey other application-specific status conditions. When bit 5 in the RT Configuration register is reset, this word is transmitted in response to “Transmit BIT Word” Mode Code 19. After power-on reset, this register reads 0x0000 unless bits 3:1 are set due to reset fault conditions.

### **Step 13: RT1 Alternate BIT Word Register (0x001F).**

The acronym BIT stands for “built-in test”. If the ALTBITW option bit 5 is set in the RT Configuration Register when a valid “Transmit BIT Word” mode command (MC19) is received,

## AN6130 DG

the current value in this register is transmitted as the mode data word in the terminal response. This option is used when the system specification calls for a BIT word defined differently from the preset BIT Word defined for register 0x001E. The demo program sets this register initially to an arbitrary value, 0xABCD. This register is used by the demo program which is described in the Mode Code section.

### Step 14: RT1 Descriptor Table

To enable message interrupts and specify RAM data buffers, each RT uses a Descriptor Table consisting of 128 consecutive "descriptor blocks", each comprised of four 16-bit words. There are four quadrants of 32 four-word blocks. The default starting location for the RT1 Descriptor Table is 0x0400 (0x0800) which was set by writing to the Descriptor Table Base Address Register in Step 6. Page 5 shows table partitioning onto four quadrants.

- 30 Receive SA commands, SA1-30 (SA0 & SA31 indicate mode commands)
- 30 Transmit SA commands, SA1-30 (SA0 & SA31 indicate mode commands)
- 32 Receive Mode Code commands, MC0-31
- 32 Transmit Mode Code commands, MC0-31

Lines 2 and 3 in the following table show the 4-word descriptor blocks for receive subaddresses SA1 and SA2, as used in this demo project. The first line of 4 addresses is a "place holder" for subaddress 0 words. In MIL-STD-1553, subaddresses SA0 and SA31 indicate mode code commands. Hence, the four SA0 word locations are not used; in this demo program, these words are written by the host with "0xDEAD" but it is not necessary to write these locations.

### Control Block Structure

SA	Control Word	Descriptor Word 2	Descriptor Word 3	Descriptor Word 4
0	0xDEAD	0xDEAD	0xDEAD	0xDEAD
1	0x4004	0x0800	0x0822	0x0844
2	0x8000	0x0A00	0x0020	0x0D10

The first location in the 4-word descriptor block for Receive SA1 is the Control Word 0x4004. Some Control Word bits are managed by the device, other bits are managed by the external host. Bits 2-0 in this Control Word establish the data buffer mode for the subaddress, either double-buffered Ping-Pong, single-buffered Index mode, or Circular Mode 1 or 2.

On the SA1 line, we see Receive SA1 uses Ping-Pong since PPEN is high and CIR1EN and CIR2EN are both reset. Receive SA1 is configured to generate an interrupt since the IWA bit14 is set high. In Ping-Pong mode, Descriptor Word 2 contains Data Pointer A and Word 3 contains Data Pointer B for storing or retrieving buffered SA1 data. When the Control Word DPB bit is 0, data for the next-occurring SA1 receive message will use Data Pointer A. When the DPB bit is 1, data for the next-occurring SA1 receive message uses Data Pointer B. The DPB bit toggles after each successfully-transacted message, so data storage for successive messages alternately uses the DPA and DPB buffer locations.

Notice that Receive subaddress SA2 is configured for Index buffer mode with an Index count of decimal 32 (0x20). The buffer should be sized to hold 32 times the single-message requirement, nominally 34 words each: a Message Information Word, a Time Tag Word and up to 32 data words. When the subaddress is configured for Index buffer mode, the Descriptor Word 2 is

## AN6130 DG

current buffer storage address (at initialization, set equal to the buffer start address) and Descriptor Word 3 contains the message count (Index) which decrements when error-free messages are transacted. The HI-6130 data sheet section 20.4 provides further details of Descriptor Words used by Indexed buffers. A review of this data sheet section explains other configuration bits. Each error-free message advances the DPA buffer pointer by  $2 + N$ , where  $N$  is the number of words in the message, and the Descriptor Word 3 Index count decrements by one. When Index decrements from 1 to zero, an interrupt can be enabled (buffer full). The host should service the buffer then reinitialize the Index and Data Pointer A for continued reuse of the buffer. When the Index value is zero, subsequent messages repeatedly overwrite the last-used buffer range, reverting to Indexed Single Message Mode.



Any subaddress command or Mode Code command may be programmed to generate an external interrupt on the /IRQ output pin if the IWA bit is set high in the associated message Control Word. In addition, the IWA bit must be set in both the RT Interrupt Enable Register and the RT Interrupt Output Enable Register.

### Step 15: RT1 Illegalization Table

Each RT uses a 256 word Illegalization Table in RAM which is fully cleared by Master Reset RAM self-test. When the table is fully clear, “illegal command detection” is disabled; the RT provides “in form” response to all valid commands, responding as commanded with Clear Status, unless message error occurs.

To enable/disable illegal command detection in this demo, the external MCU host reads C project file `613x_initialization.h` for the statement `#define ILLEGAL_CMD_DETECT (YES or NO)` to determine whether the default fully-cleared Illegalization Table should be initialized using values in an array provided in the demo project. The user can disable illegal command detection by changing the macro to “NO” then rebuilding the project. For a complete description of table bit fields, refer to the “Command Illegalization Table” section in the HI-6130 data sheet and comments in the `613x_RT.c` source file.

### Step 16: Transmit Subaddress data buffer initialization

The host writes arbitrary data words into assigned Transmit Subaddress RAM buffer locations specified by the descriptor table Control Blocks, described in Step 14. There are two 32-word arrays defined in the C program providing arbitrary values used for initializing these buffers. The buffer associated with the SA1 Transmit Subaddress is referenced later in this document, concerning demo program operation.

### Step 17: Enable the RT.

At step 2, the RT1ENA bit was already set high in the Master Configuration register, prior to RT RAM feature initialization. The last remaining step to start RT execution is setting the **RT1STEX** bit in the Master Control Register. This enables the RT to respond to valid messages from the BC. After the host completes these steps, the HI-613x responds to MIL-STD 1553 messages automatically without host interaction.

After a receive command is transacted, the RT stores 1 to 32 incoming data words at the buffer address indicated by the descriptor table Control Block corresponding to the receive subaddress. After data words are received, the RT transmits a Status Word back to the BC to complete the transfer. The expected number of words is determined by the BC message and is encoded in the MIL-STD-1553 command word.

When the RT receives a Transmit Subaddress command, the RT responds to the BC with a status word then contiguously transmits the requested 1 to 32 words from the data buffer indicated by the Data Pointer A or Data Pointer B depending on the buffer scheme selected in the descriptor table Control Block for the transmit subaddress

In this demo project, Transmit SA1 commands are configured for single-buffer Index Mode. When the RT receives a Transmit SA1 command from the BC, the transmit data always comes from the buffer start address indicated by Data Pointer A. Unless the host MCU overwrites the data in the Transmit SA1 buffer, the RT always transmits the same data to the BC for transmit SA1 messages.

In this demo project, Receive SA1 commands are configured for double-buffered Ping-Pong Mode. When the RT gets a Receive SA1 command from the BC, incoming data words are initially stored in a temporary receive buffer until all expected data words are received error-free. When all words are received, the RT responds back to the BC with its MIL-STD-1553 Status Word. Data is transferred from the temporary receive buffer to the active buffer indicated in the Control Word DPB bit: When DPB = 0, data words are stored in the buffer indicated by Data Pointer A. When DPB = 1, data words are stored in the buffer indicated by Data Pointer B.

### Host Data Interaction with message data

This section describes how the host services message data buffers: offloading data in receive message buffers and refreshing the data in transmit message data buffers.

### Simple RT Demo Program

The demo project can concurrently demonstrate any combination of three 1553 terminal types: BC, MT and RT1/RT2 by selecting a BC\_RT\_MT configuration from the IAR Systems *Embedded Workbench® for ARM C* programmer's toolset. For more information on these configurations and a quick tutorial on *Embedded Workbench for ARM*, please refer to the "AN-6130 Application Development Kit" document. Several project configurations are available and selectable from the project configuration menu. This section is focused on using RT1, using either the BC\_RT\_MT or RT\_ONLY project configuration. The RT configured in this project

## AN6130 DG

demonstrates several types of RT subaddress commands, mode code commands, and all four buffer schemes with host data interaction.

The project can be configured for polled or vectored Interrupts by changing a C program statement `#define INT YES` (or `NO`) in file `613x_initialization.h` and rebuilding the project. A separate flow chart is provided for each project configuration.

### Transmit and Receive Subaddress commands

The external host MCU services subaddress command RAM data buffers to complete the transaction of message data words. For transmit subaddress commands, the host writes data into buffers for transmission; for receive subaddress commands, the MCU offloads received data from the buffers. Demonstrating MCU buffer servicing for HI-6130 RT subaddresses, the host demo program uses the Receive and Transmit commands on subaddress 1 (SA1). The Transmit SA1 command is configured for single-buffer Index Mode. The descriptor table configuration for the Transmit SA1 control block is shown below. The HI-6130 RT decodes a transmit-data command and responds by transmitting its Status Word followed by the requested number of data words, fetched from the Tx SA1 buffer address indicated by Data Pointer A, (control block Descriptor Word 2). The buffered data for transmit should be written before the transmit command occurs. For the demo program, the host MCU initializes the Tx SA1 data buffer with 32 arbitrary words as part of post-reset configuration. Explained on page 18, the post-reset initialized data can be altered if the BC issues a Receive SA1 command with Data Word 1 value either a 0x0003 or 0x0004. Although the reserved Tx SA1 buffer size is 32 words, the actual number of words transacted ranges from 1 to 32, as encoded in the received Command Word from the BC. In the demo program, Control Word IWA bit14 is asserted, enabling "Interrupt When Accessed." Thus, the HI-613x generates an interrupt output on the /IRQ output pin each time a Transmit SA1 message occurs.

#### SA1 Transmit Control Block

Control Word (IWA, index buffer)	Descriptor Word 2 (Data Pointer A)	Descriptor Word 3 (single-index)	Descriptor Word 4 (Broadcast Data Pointer)
0x4000	0x0866	0x0000	0x08AA

Receive Subaddress 1 is configured for Ping-Pong mode with IWA interrupt; the descriptor table Control Block for Rx SA1 is shown below. A subaddress can have both Transmit and Receive commands; transmit and receive are usually assigned separate data buffer locations in RAM. The exception is subaddress 30, used for MIL-STD-1553 "data wraparound". Receive SA30 and Transmit SA30 messages share a common data buffer location in RAM. As a result, data received in Receive SA30 messages is transmitted back to the BC by following Transmit SA30 commands. Transmit and receive data pointers for Subaddress 30 are initialized to 0x08AE in Descriptor Word 2 for the Rx and Tx SA30 control blocks. The shared buffer reserves space for 32 words although the actual number of transacted data words is encoded in the Rx SA30 and Tx SA30 command words received from the Bus Controller.

#### Subaddress 1 Receive Control Block

Control Word (IWA, ping-pong)	Descriptor Word 2 (Data Pointer A)	Descriptor Word 3 (Data Pointer B)	Descriptor Word 4 (Broadcast Data Pointer)
0x4004	0x0800	0x0822	0x0844

## AN6130 DG

The demo program uses two API's found in C project file `613x_Driver.c` to read or write data from/to Transmit or Receive subaddress data buffers. These API's are only compatible with Ping-Pong or single-buffer Index modes.

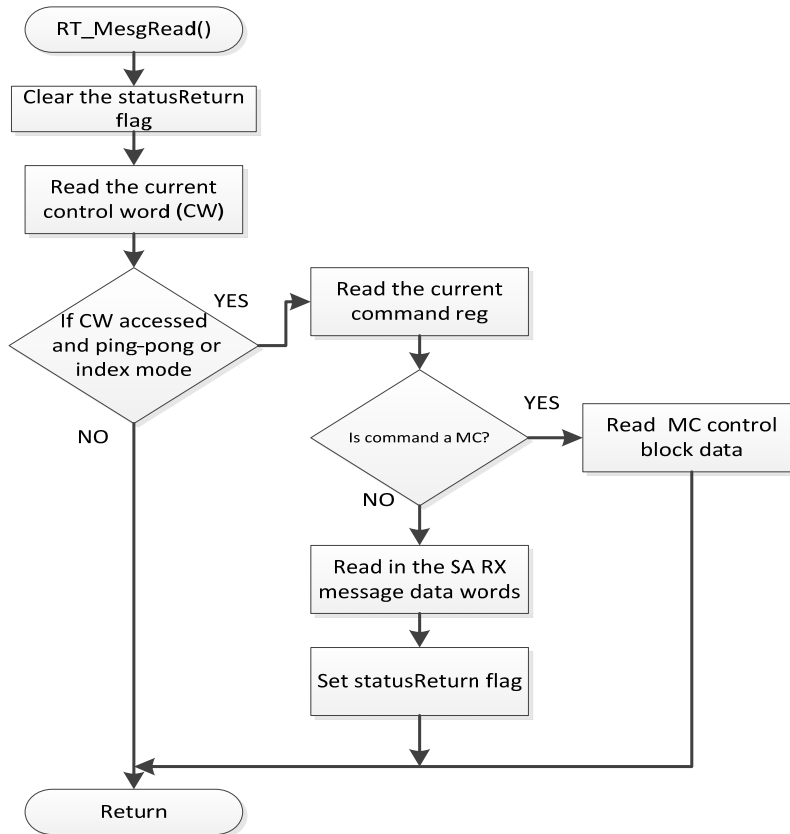
### **RX\_MesgRead ( )**

This API reads the last Receive Subaddress or Receive Mode Command transacted, and retrieves the message data word(s). The Command Word, Message Information Word (MIW), Time Tag word and 1 to 32 data words are stored in a 35 word array. The Current Control Word Address register indicates the location of the descriptor table Control Block; the Current Command Register provides the last-received Command Word from the BC. This function can be used in a polling loop or called from a message interrupt handler. Any Subaddress or Mode Command can be configured to generate an interrupt on the /IRQ output pin by setting the IWA bit in the associated descriptor table Control Word for the message.

The function Input parameters:

`BUFFMODE` - buffer mode (Ping-pong or Index Single Buffer)

`*buffer` - is a pointer to the array data is stored.







## Host Data Interaction

Host data interaction is demonstrated in the main loop of the program using the API functions `RX_MesgRead()` and `TX_MesgWrite()`. `RX_MesgRead()` detects Receive Subaddress or Receive Mode Command messages. If the demo project is configured for interrupts, `RX_MesgRead()` is called by the message interrupt service routine; all buffer servicing is handled in the interrupt handler and a flag is set. If the demo project is configured for polling, `RX_MesgRead()` is called directly in the main loop of the program in `main.c`. When a command is received by the RT, demo program displays message type and associated data words on the console if the computer has a serial port and is set up for terminal operation.

When the program is configured for interrupts, only messages having IWA bit 14 bit asserted in the Control Word are detected by the `RX_MesgRead()` function. Messages without the IWA bit-14 set will still be acknowledge by the RT with a proper Status Word, but the Simple RT console message will be absent. When the program is configured for polling, all messages will be detected and displayed on the console.

For Receive SA1 commands, the host transacts on the DW1 value by calling `executeSACommand()` and performs the actions listed below:

### Receive SA1 Commands with Data Word 1 action values

Host Action	DW1 Value
Flash Green LED	0x0001
Flash Amber LED	0x0002
Load TxSA1 Data Buffer with Data Set 1	0x0003
Load TxSA1 Data Buffer with Data Set 2	0x0004

After reset, the RT responds to Transmit SA1 commands with data initialized in the TxSA1 buffer by MCU initialization or auto-initialization from EEPROM. TxSA1 buffer data can be altered in response to a Receive SA1 command from the BC, based on the value on received Data Word 1:

### Transmit SA1 Commands

HI-6130 Device	Buffer Values from DW1 - DW32
Transmit Buffer data set at post-reset initialization	0x0101, 0x0202, 0x0303, 0x0404, 0x0505 ....0x3232
Transmit Buffer Set 1	0x1000, 0x1001, 0x1002, 0x1003, 0x1004 ....0x1001F
Transmit Buffer Set 2	0x2000, 0x2001, 0x2002, 0x2003, 0x2004 ....0x2001F

## AN6130 DG

In C project file 613x\_initialization.h if statement #define CONSOLE YES, then the following console output will be displayed by the Simple RT demo program.

### RT1\_ONLY configured project at power up:

```
Holt Integrated Circuits HI-6130 Simple Terminal Ver: 1.0
Compiled: Dec 6 2012 14:58:37
```

Host is Initializing Regs & RAM

```
*****
Holt Integrated Circuits HI-6130 Simple Terminal V 1.0
Compiled: Dec 6 2012 14:44:06
*****
```

```
BC Off MT Off RT1 On RT2 Off
Press '6' to list HW interrupt status...
Press '8' to list RT interrupt status...
NOTE: Options 6-9 clear the accessed Pending Interrupt Register!
=====
```

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

#### Simple RT TX Polled Message:

```
CW: 1C20 MIW: 0000 TimeTag: 9225 SA: 1
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
```

#### Simple RT RX Polled Message: (flashes the green LED)

```
CW: 1830 MIW: 0010 TimeTag: 20CC SA: 1
0003 0004 0005 0006 0007 0008 0009 000A 000B 000C 000D 000E 000F 0010 0011 0012
```

#### Simple RT TX Polled Message:

```
CW: 1C20 MIW: 0000 TimeTag: 29AF SA: 1
1000 1001 1002 1003 1004 1005 1006 1007 1008 1009 100A 100B 100C 100D 100E 100F
1010 1011 1012 1013 1014 1015 1016 1017 1018 1019 101A 101B 101C 101D 101E 101F
```

#### Mode Code-3

Simple RT TX Polled Message:

Mode Code: 03

Control Word: 0x1C03

Mode Command Control Word: 0x4000

Message Information Word: 0x0503

TimeTag: 0xDAB2

Initiate Self-Test by HOST Timer Alive \*\*\* Passed \*\*\*

## AN6130 DG

---

Messages with the program configured with message interrupts.

### Simple RT RX Interrupt Message with 32 Data Words:

Simple RT RX Interrupt Message:

```
CW: 1820          MIW: 0020          TimeTag: F8EF   SA: 1
0004 0001 0002 0003 0004 0005 0006 0007 0008 0009 000A 000B 000C 000D 000E 000F
0010 0011 0012 0013 0014 0015 0016 0017 0018 0019 001A 001B 001C 001D 001E 001F
```

### Simple RT TX Mode Code 19 Command Interrupt Message:

Simple RT TX Interrupt Message:

Mode Code: 19

Control Word: 0x1FF3

Mode Command Control Word: 0x4000

Message Information Word: 0x0013

TimeTag: 0x7253

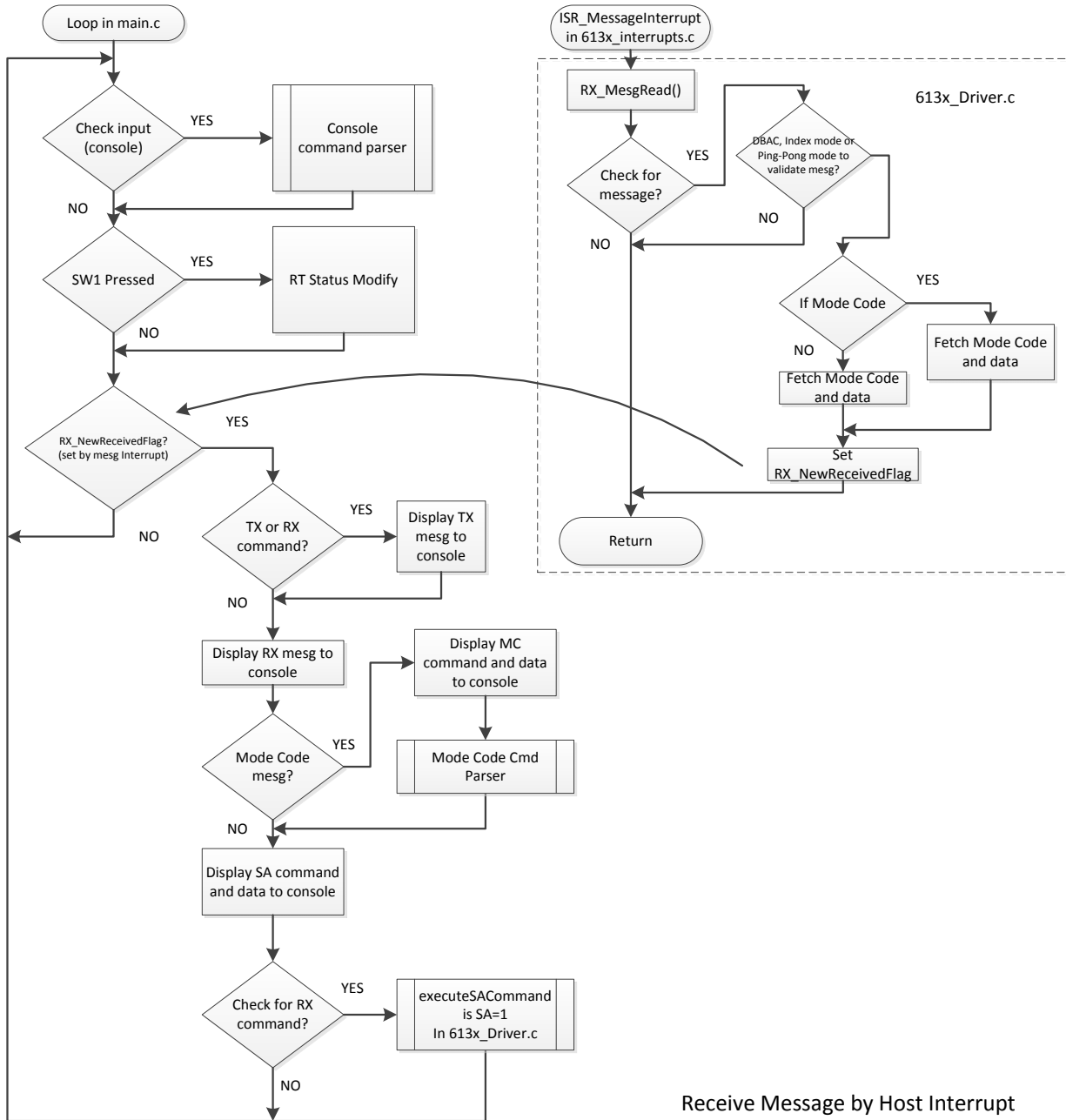
Mode Code Data Word: 0xABCD

If the project is rebuilt with a different configuration to include a BC such as the BC\_RT1 configuration, the BC displays other message information. The BC information is displayed before the Simple RT message data. The separate "AN-6130 Application Development Kit" document provides information on various project configurations and how to select them.

The following flow charts show the program flow for Polled Interrupt and Vectored Interrupt configurations of the Simple RT demo project.

# AN6130 DG

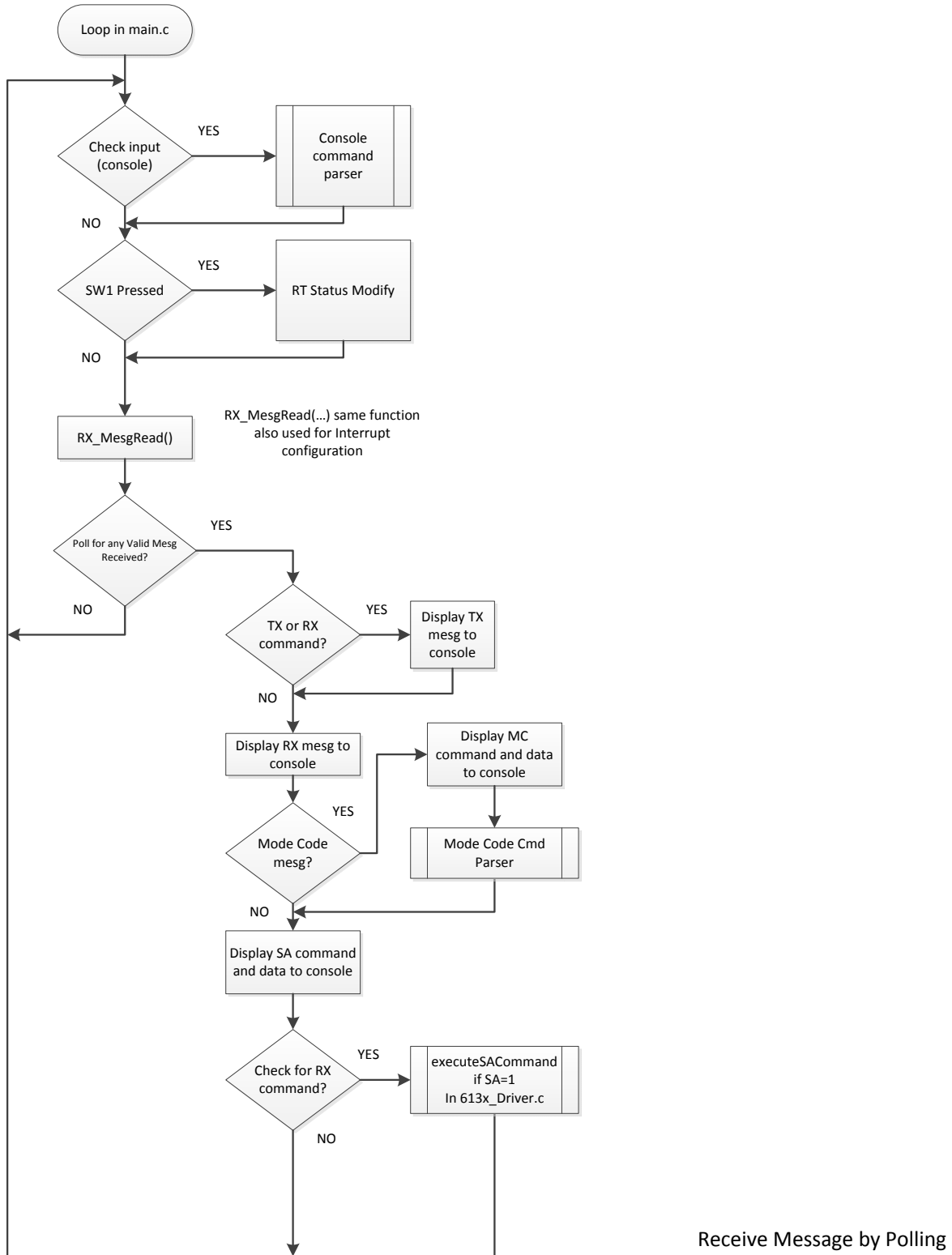
This flowchart shows the main loop and message retrieval by interrupts.



Receive Message by Host Interrupt

# AN6130 DG

This flowchart shows the main loop and message retrieval by polling.



### Mode Code Commands

The HI-6130 demo project configures the Descriptor Table and Illegalization Table to enable Mode Codes so the RT acknowledges and responds to a subset of the defined Mode Code commands. If illegal command detection is enabled (default), the unimplemented, reserved and undefined mode commands result in Message Error RT status.

The RT automatically responds to the BC for legal Mode Code commands. Example mode commands for which the RT transmits Status Word response and performs defined mode code action include: MC1 (synchronize without data), MC2 (transmit Status Word), MC4 (transmitter shutdown), MC5 (override transmitter shutdown), MC8 (reset remote terminal), MC17 (synchronize with data), MC18 (transmit last command), MC19 (transmit BIT word) and others. Some of the mode commands implemented in the demo program may require additional user programming on the host side to fully implement the intent of the MC or to tailor RT action to meet application-specific requirements.

#### RT Self-Test and Built-In Test (BIT) Word

The Mode Code 3 command (initiate self-test) is issued by the BC when remote terminal self-test is needed. The HI-613x has built-in features supporting self-test that is managed by the RT MCU program. The built-in self-test features, including RAM test and loopback tests, initiated by MCU writes to HI-6130 control registers. The hardware-conducted RAM test is thorough but “destructive,” overwriting preexisting register and RAM data, so may not be suitable for inclusion in MC3 mode commands. If RAM testing is needed for Mode Code 3 response, software-based tests conducted by the MCU program may meet requirements while restoring data as tests are performed. The implemented tests and how they are performed depends on end user system requirements. A complete test strategy probably consists of built-in tests and additional tests implemented by the host programming. The RT cannot respond to valid legal commands while self-test is underway. The RT may need full initialization afterwards to restore normal RT response, especially if destructive RAM tests are performed. Live periodic tests are also possible if the host implements simple, fast sequences. For example, the MCU program might periodically check several reserved RAM locations for read/write operation. Another possibility is a host-managed quick check to verify the HI-613x Time Tag counter is operational. A method to do this is found in the demo program.

After the completion of the built-in self-test, the RT1 Built-in Test Word Register 0x001E should be updated with test results. When built-in (RAM or loopback) test features are used, certain predetermined bits in register 0x001E are written by the device. Other bits in that register are freely defined for reporting status of other self-test routines executed in host programming. If ALTBITW bit 5 is **reset** in RT1 Configuration Register 0x0017, the RT automatically transmits its Status Word followed by the register 0x001E value in response to the MC19 command.

If ALTBITW bit 5 is **set** in RT1 Configuration Register 0x0017, the RT automatically transmits its Status Word followed by the value in the Alternate BIT Word register 0x001E value in response to the MC19 command. The alternate BIT word register is used when the system specification defines flag bits in conflict with the fixed register 0x001E bit assignments. The RT Alternate Built-in Test (BIT) Word register allows full flexibility in BIT Word definition. This register is used in the example below.

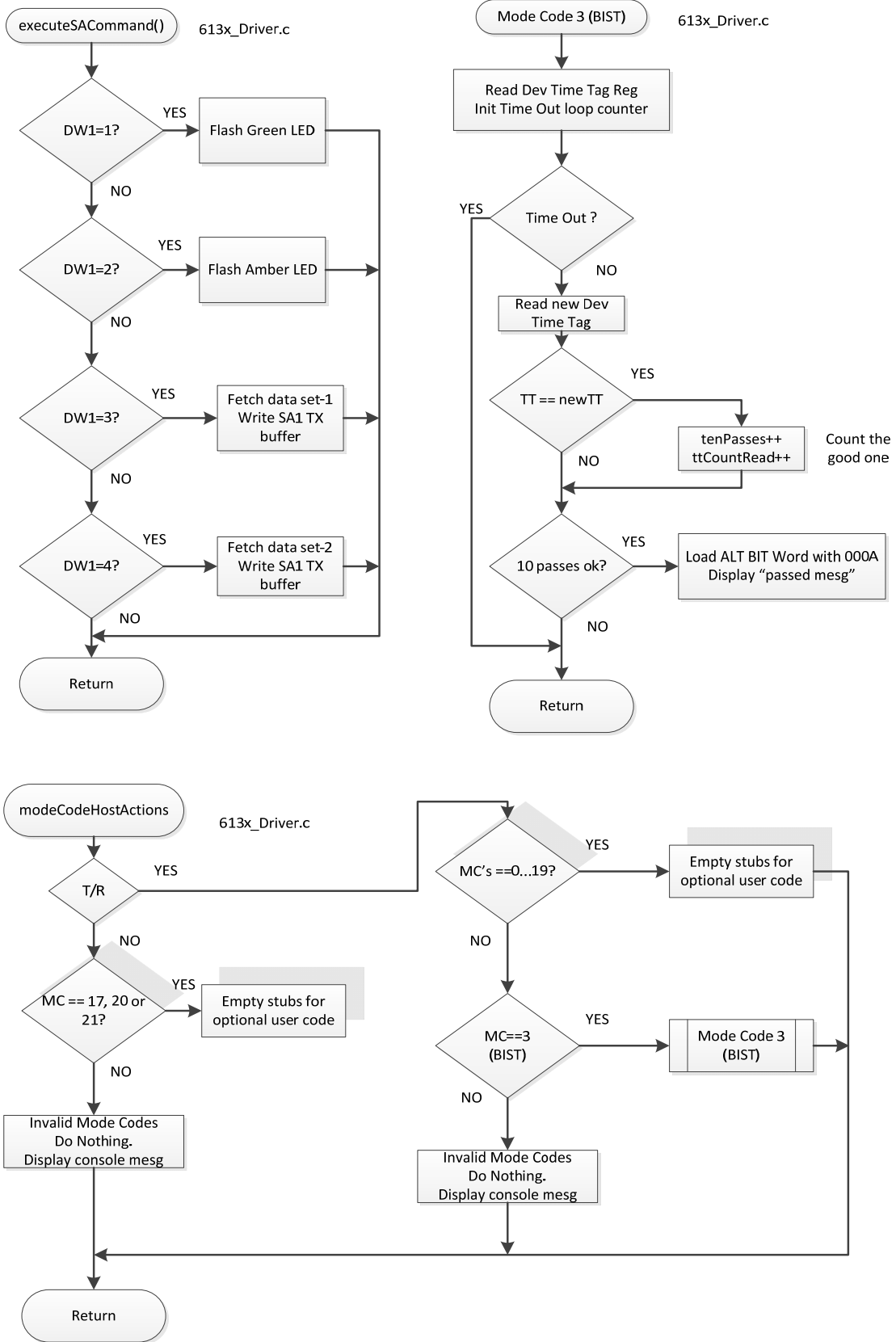
### Mode Code 3 (BIT) example

The Simple RT demo program provides a “built-in test” example using MC3 and MC19. The function `modeCodeHostActions()` is called when the RT receives Mode Code commands. This function provides program stubs permitting easy insertion of application-specific routines for mode code commands. A handler for Mode Code 3 (initiate self-test) is provided in the `modeCodeHostActions()` function. When MC3 is received, host software performs a test to verify that the HI-6130 Time Tag counter is counting properly. The host reads the value of the Time Tag counter, then rechecks the incrementing count for the next 10 consecutive values. If the test passes, the host program writes 0x000A into the Alternate BIT Word Register. If the console is enabled, a “Passed” message is also displayed on the terminal screen. If the test fails, the value loaded into the RT1 Alternate Built-In Test Word Register will be either 0x0000 or a count less than 10 (0x000A). Refer to the Mode Code flow chart.



# AN6130 DG

## Support functions



# AN6130 DG

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

	<b>Dec</b>	<b>Hex</b>
Descriptor Table Base	1024	0400
First Buffer Address	2048	0800

Descriptor Table	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		Data Bus Addr Hex HI-6130 Only		Structures Reserved MIW = Msg Info Word TT = TimeTag Word
			Start	End	Start	End	
Rx SA1 (data pointers A, B and Broadcast data pointer)	Ping-pong DPA DPB BDP	34 34 34	0800 0822 0844	0821 0843 0865	60001000 60001044 60001088	60001042 60001086 600010CA	MIW + TT + 32 words same same
Tx SA1 (data pointers A, B and Broadcast data pointer)	Ping-pong DPA DPB BDP	34 34 4	0866 0888 08AA	0887 08A9 08AD	600010CC 60001110 60001154	6000110E 60001152 6000115A	same same 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 DPB	1088 34	08D0 0D10	0D0F 0D31	600011A0 60001A20	60001A1E 60001A62	32 x ( MIW + TT + 32 words)
Tx SA2	Index-32 DPA DPB	1088 34	0D32 1172	1171 1175	60001A64 600022E4	600022E2 600022EA	MIW + TT + 2pad
Rx SA3	Circ1-32 DPA pad	1088 34	1176 15B6	15B5 15D5	600022EC 60002B6C	60002B6A 60002BAA	32 x ( MIW + TT + 32 words) pad for overrun
Tx SA3	Circ1-32 DPA pad	1088 34	15D6 1A16	1A15 1A35	60002BAC 6000342C	6000342A 6000346A	32 x ( MIW + TT + 32 words) 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 MIB msg max DPA	512 8192	1C00 1E00	1DFF 3DFF	600036A0 60003C00	60003BFE 60007BFE	256 X (MIW + TT)
assigned to BC	---	256	1E00	1EFF	60007C00	60007DFE	BC Msg Control Blocks
Unassigned RAM	---	256	3F00	3FFF	60007C00	60007FFF	

### Summary,

This RT design guide describes the key HI-6130 Remote Terminal RAM structures and configuration registers. It provides a step-by-step procedure for configuring the RT1 Remote Terminal. The accompanying Simple RT demo project demonstrates several functions that read Receive Subaddress message data and transact variable data to the BC in response to Transmit Subaddress messages. The Simple RT demo demonstrates customization of mode command actions, as well as use of polled or vectored interrupts.

**Conclusion:**

**REVISION HISTORY**

---

<b>P/N</b>	<b>Rev</b>	<b>Date</b>	<b>Description of Change</b>
AN-6130 DG	NEW	11/18/12	Initial Release

---