HOLT INTEGRATED CIRCUITS INC.

AN-520

May 2008

MIL-STD-1553 Remote Terminal Design Example Applications Note

INTRODUCTION

The HI-6110 is a versatile message processor for MIL-STD-1553 applications. It can be configured as a Bus Controller, Remote Terminal or Bus Monitor, with or without assigned RT address.

This application note describes a Remote Terminal design example using the HI-6110 and a simple, commercial microcontroller. This example is available from Holt Integrated Circuits as an evaluation board. AN-520 provides an overview of hardware and software principles for a working design. Schematic diagram and program flowcharts are included. This can be used for HI-6110 evaluation, and may be a suitable design basis for some applications. The evaluation board has the following features:

- Two 1553 bus interfaces, transformer- or direct-coupled.
- Signal headers for connecting a logic analyzer
- Status LEDs
- Buttons for board reset and test sequencing
- DIP switch for selecting RT address, 1553 operating mode
- Serial port for an optional computer interface
- Microcontroller JTAG header for software debug
- Self-contained 3.3V power supply

Figure 1 is a block diagram of the evaluation board. Some features shown apply only to this demonstration circuit.

REMOTE TERMINAL OPERATION

For RT mode, a DIP switch sets the 5-bit Remote Terminal address and parity. At reset, the HI-6110 checks its address inputs for valid parity. If parity is wrong, an error is signaled to the host controller, and the device will not function until address and parity are corrected and reset is applied again.

Each bus has a dedicated 1553 decoder. Overlapping valid commands on both buses are decoded properly. All 1553 words received are checked for "word validity" – proper sync, 16 data bits plus correct parity, and proper Manchester II encoding.

"Command validity" is also checked. Commands addressed to other remote terminals are ignored. When a received command word address matches the preset HI-6110 RT address (or indicates broadcast command) the command is validated. The HI-6110 asserts a RCVA or RCVB output to signal which bus received the valid command. When RCV is asserted, the host reads the HI-6110 Message Register to determine one of twelve message types.

When a valid non-broadcast command is received, the HI-6110 automatically begins Status Word response after the command is received in full. The device Status Word Register is combined with the preset RT address inputs to fully specify the status word response.

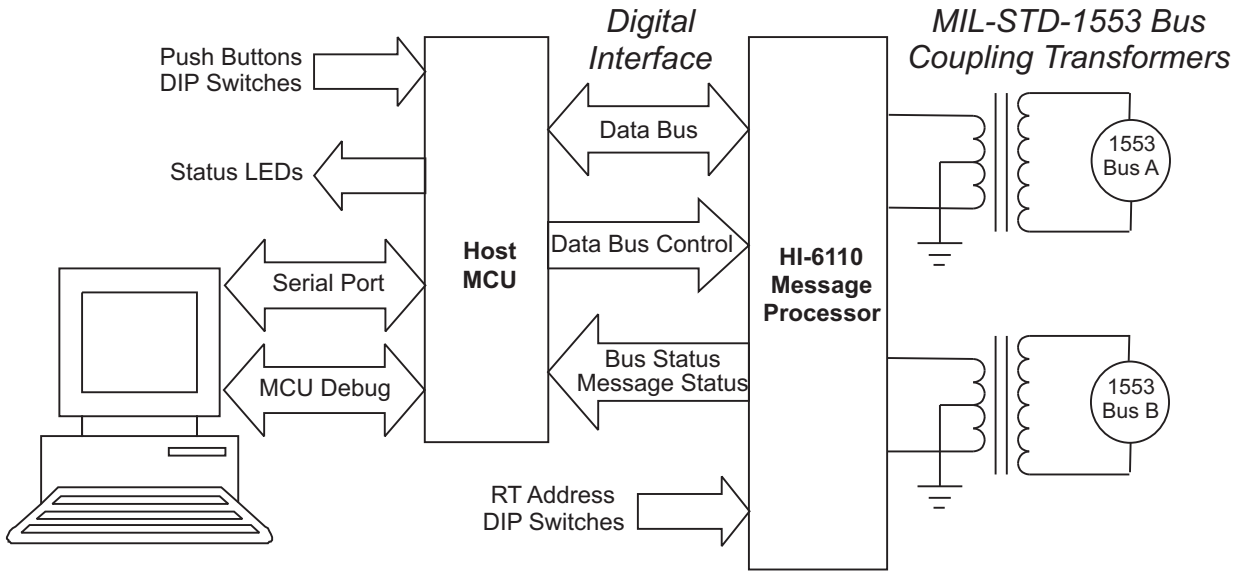


Figure 1. Typical Remote Terminal Block Diagram

For valid transmit commands, status word transmission begins automatically, about 6uS after the Command Word is completed. During the 20uS Status Word transmit interval, the host begins loading the Transmit Data FIFO or Transmit Mode Data Register.

For valid, non-broadcast receive commands, status word transmission is automatically handled. The HI-6110 stores received Data Words in its 32-word receive data FIFO, and resets its FFEMPTY (receive FIFO empty) status flag when the first data word is received. When low, the FFEMPTY flag indicates at least one unread data word is available in the FIFO. Successive FIFO fetches will read data words in the order received. When the last word is read, FFEMPTY goes high and stays high.

If a valid receive command is followed by data word error, the HI-6110 asserts its ERROR output. The host then reads the device Error Register to determine the error type. The host can write to the HI-6110 Status Word Register to set the "message error" (ME) bit. Changes to the Status Word Register must be written up to 1uS after mid-parity of status word transmission.

By handling low-level 1553 protocol requirements, the HI-6110 reduces host real-time programming demands. The host handles high-level tasks like loading data word FIFOs, interpreting received messages, and interaction with the Remote Terminal subsystem.

DEBUG FEATURES

Two 40-pin logic analyzer signal headers provide connections for all HI-6110 signals. This includes isolation networks for direct connection of Agilent logic analyzer 40-pin pod connectors without using the "flying lead" cables. The Agilent E9340A PC-hosted logic analyzer is a suitable example.

Direct signal headers permit connection of other logic analyzers, using conventional "flying lead" signal connectors.

The host microcontroller is a Zilog Z8F6422 8-bit MCU with flash program storage. During software development, an inexpensive JTAG "Smart Cable" connects the board's 6-pin debug header to a computer USB or serial port. The "Smart Cable" comes with a software development package called "Zilog Developer Studio II for Z8 Encore". We recommend that you register the purchased cable at www.zilog.com so you can download the current version of the program. The "ZDS II" program includes an assembler, C-compiler, linker, simulator, and complete source-level debug interface. Equipped with the ZDS II program and cable, the computer can perform source-level software debug, including single stepping, breakpoints, examining registers, etc. This is also used for programming the Flash program memory in the MCU.

Software for the serial port connected to the host microcontroller is not implemented. The microcontroller has an integral UART connected to the RS-232 level shifter and DB-9 connector. Program changes would permit use of this communication port.

The schematic diagram for the board is provided at the end of this application note. Assembled evaluation boards for the design are available from Holt Integrated Circuits.

RT HOST SOFTWARE

The example program written for this design is available as a separate PDF document, and as a text file directly compatible with the text editor in the Zilog ZDS II design software. The following summarizes the program structure:

1. Upon power-up or forced reset, the Z8 microcontroller initializes itself. It then initializes the HI-6110 by setting the mode-determining input pins, and writing to its Control register.

In the HI-6110 Control Register, only one bus can be designated as "active" while the other bus is considered "inactive". HI-6110 transmissions can only occur on the active bus.

2. The register interface to the HI-6110 is "bit-banged" in this example. With the /CS chip select low, the 4-bit Register Address and Read-/Write control signal is output. For write sequences, the data is output onto the bus by the MCU. For read sequences, the MCU bus port pins are put into high-impedance input mode. The /STR strobe is pulsed low to execute the read or write operation. Write data words are latched into the selected HI-6110 register on the rising edge of /STR strobe.

3. After initialization, the host enters Standby mode, where it awaits one of five interrupts from the HI-6110: RCVA, RCVB, RCVCMDA, RCVCMDB or ERROR.

4. A receive interrupt occurs when a valid command is decoded. The host determines the receiving bus based on which input the interrupt occurs. If the valid command is received on the inactive bus, the host switches active/inactive buses by writing to the HI-6110 Control register. The host then fetches the contents of the Message register and branches to the message decode routine. Appropriate action is determined, based on received message type. The example program shows how the message is decoded.

The example program sends arbitrary data words when responding to transmit commands, loading the transmit FIFO upon receiving the command. All mode code commands required for validation are implemented, plus "Transmit Last Command".

5. Upon completing the appropriate RT response for a valid message, the HI-6110 signals one of two results to the host: VALMESS indicates a good result, while ERROR is self-explanatory.

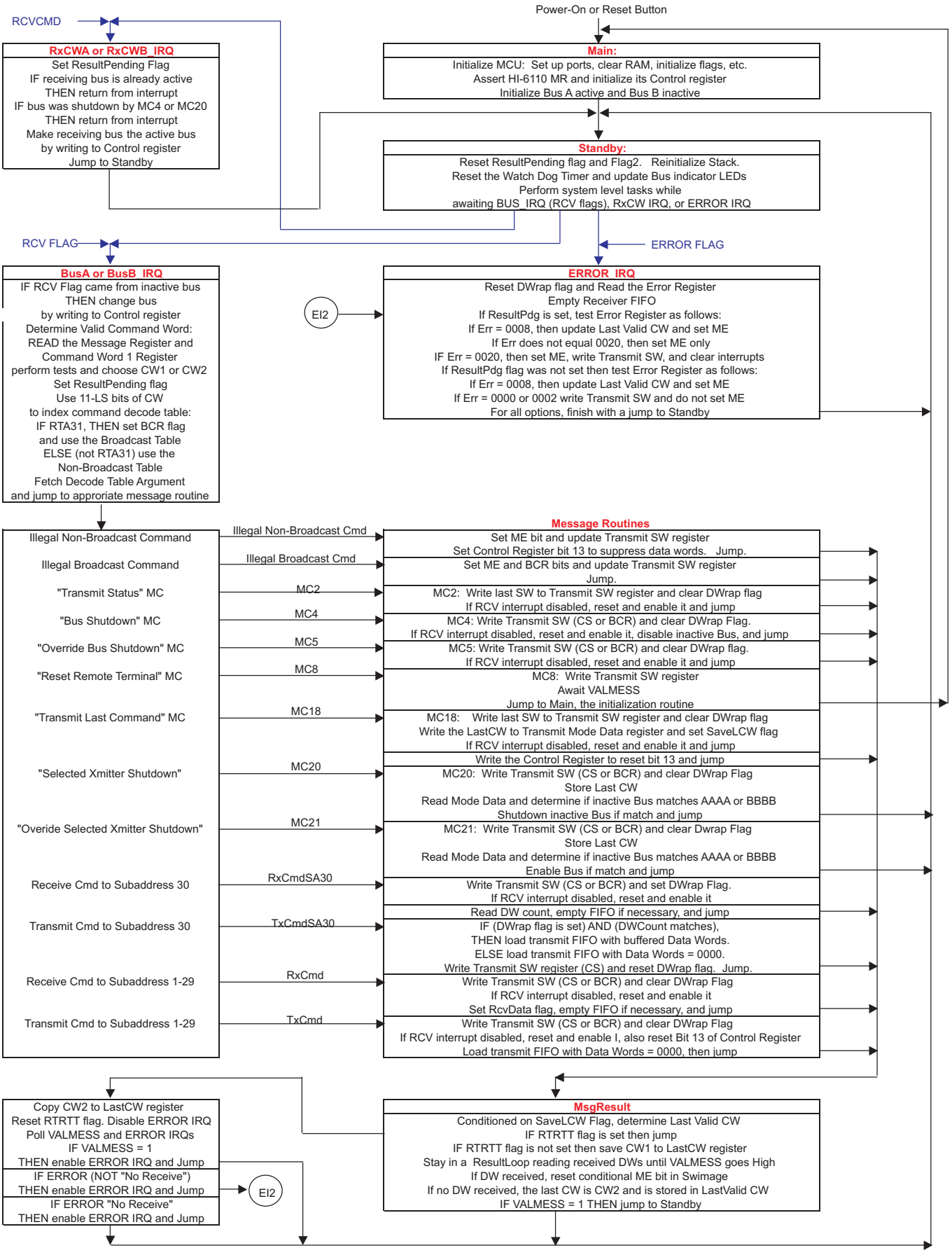
6. An error interrupt can occur after a receive interrupt. It can also occur before or without an accompanying receive interrupt. When unaccompanied by a receive interrupt, an error indicates a valid command was decoded concurrently with the recognized error. The flow chart on the next page shows the program logic for handling RCVA, RCVB and ERROR interrupts consistent with the requirements for remote terminal operation per MIL-HDBK-1553.

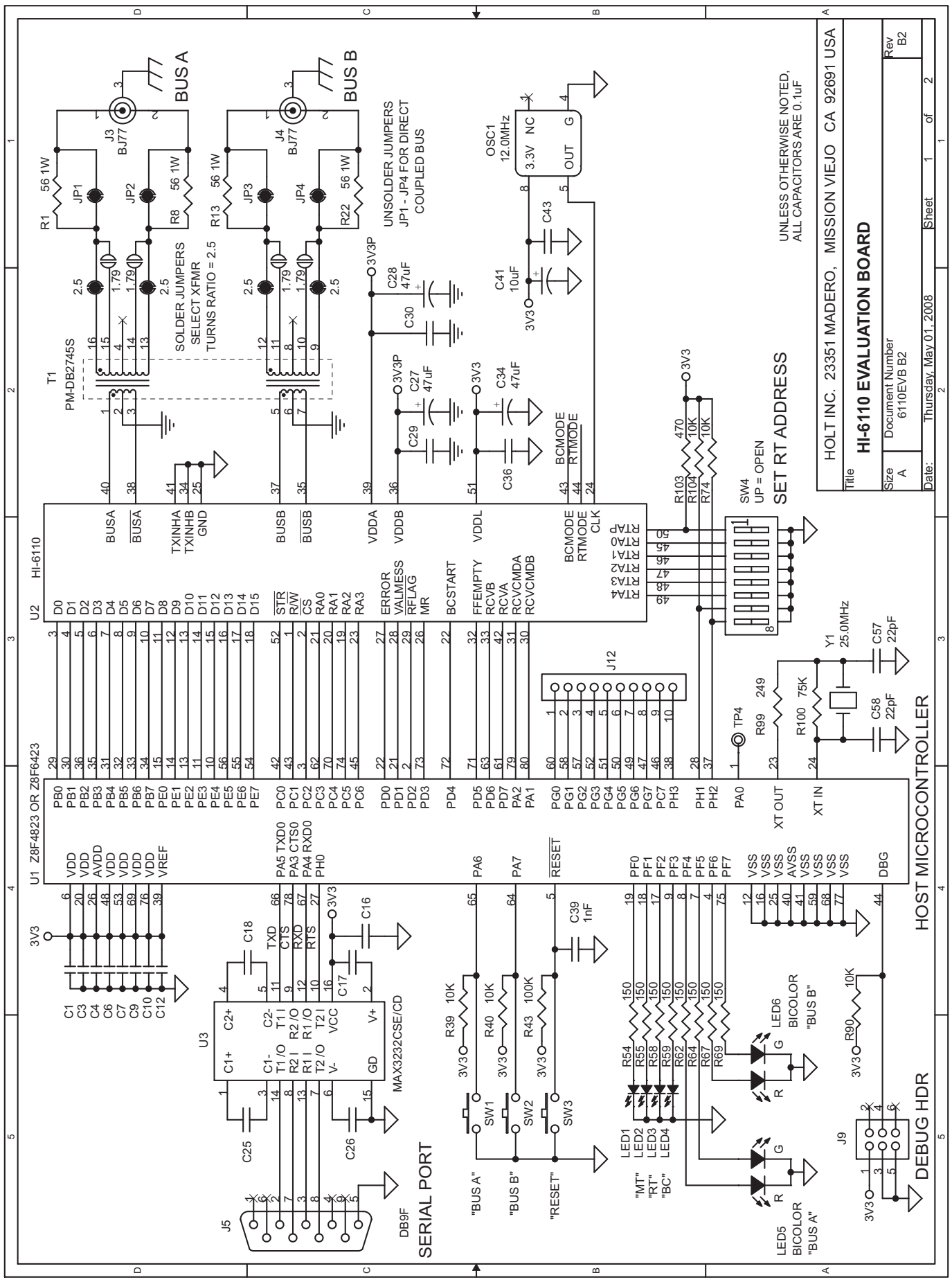
In MIL-HDBK-1553, the sections entitled "RT Response to Commands" and "Error Injection" test a wide range of illegal, undefined or otherwise inappropriate 1553 message sequences. Frequently, the remote terminal is given as many as three or four acceptable forms of response to a given error condition. In many cases, HI-6110 inherent behavior comprises an acceptable response without host assistance. In some cases, the example program overrides inherent behavior to demonstrate illegal command detection, or "illegalization".

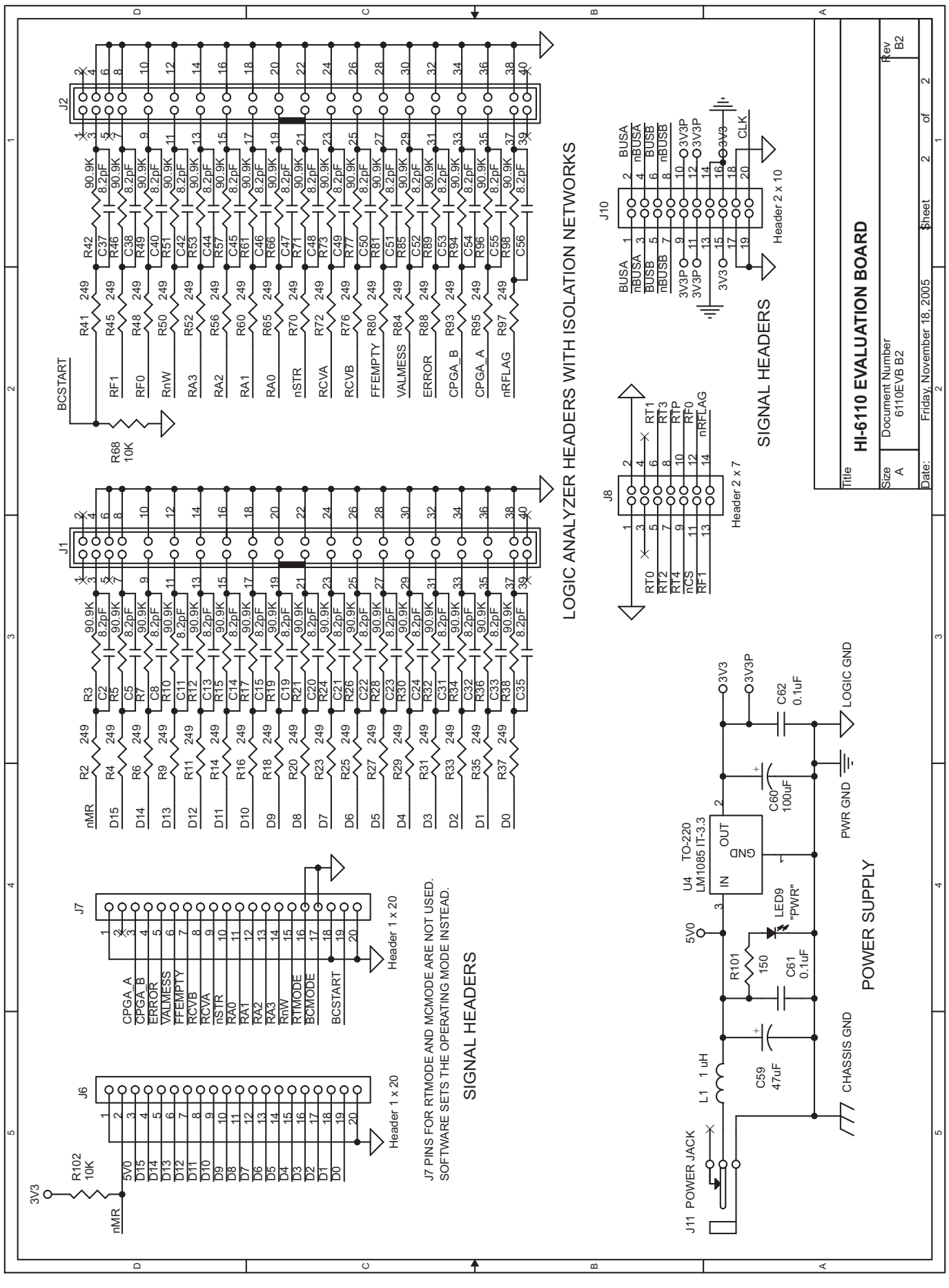
Many of the alternate response forms are easily implemented by modifying the host program.

HI-6110 PROGRAM FLOW CHART

HI-6110 INTERRUPT HANDLING FOR RT VALIDATION




HOLT INC. 23351 MADERO, MISSION VIEJO CA 92691 USA
HI-6110 EVALUATION BOARD
Document Number 6110EVB B2
Rev B2
Sheet 1 of 2
Thursday, May 01, 2008
UNLESS OTHERWISE NOTED, ALL CAPACITORS ARE 0.1uF
BUS A
BUS B
T1 PM-DB2745S
SOLDER JUMPERS SELECT XFMR TURNS RATIO = 2.5
UNSOLDER JUMPERS JP1 - JP4 FOR DIRECT COUPLED BUS
OSC1 12.0MHz
U2 HI-6110
U1 Z8F4823 OR Z8F6423
U3 MAX3232CSE/CD
SET RT ADDRESS
SERIAL PORT
HOST MICROCONTROLLER
DEBUG HDR
DB9F
"BUS A"
"BUS B"
"RESET"
"MT"
"RT"
"BC"
BICOLOR "BUS A"
BICOLOR "BUS B"


LOGIC ANALYZER HEADERS WITH ISOLATION NETWORKS
SIGNAL HEADERS
POWER SUPPLY
SIGNAL HEADERS
J7 PINS FOR RTMODE AND MCMODE ARE NOT USED.
SOFTWARE SETS THE OPERATING MODE INSTEAD.
HI-6110 EVALUATION BOARD
Document Number 6110EVB B2
Rev B2
Size A
Date: Friday, November 18, 2005
Sheet 2 of 2
Header 2 x 10
Header 2 x 7
Header 1 x 20
Header 1 x 20
U4 LM1085 IT-3.3 TO-220
J11 POWER JACK
L1 1 uH
CHASSIS GND
PWR GND
LOGIC GND

REVISION HISTORY

Revision	Date	Page	Description of Change
AN-520, Rev. E	05/01/08	1	Date was “May 2006”, is “May 2008”. Footer was “Rev. D”, is “Rev. K”.
		4	Added connections for ratio select jumpers on transformer T1. Added note regarding jumpers Jp1- JP4 and direct coupled operation.
		6	Added “Revision History”