HI-3200, HI-3201

AVIONICS DATA MANAGEMENT ENGINE / ARINC 429 - CAN BUS BRIDGE

October 2017

GENERAL DESCRIPTION

The HI-3200 from Holt Integrated Circuits is a single chip CMOS data management IC capable of managing, storing and forwarding avionics data messages between eight ARINC 429 receive channels, four ARINC 429 transmit channels and a single CAN / ARINC 825 data bus.

The ARINC 429 and CAN buses may be operated independently, allowing a host CPU to send and receive data on multiple buses, or the HI-3200 can be programmed to automatically re-format, re-label, re-packetize and re-transmit data from ARINC 429 receive buses to ARINC 429 transmit buses, as well as from ARINC 429 to CAN or CAN to ARINC 429.

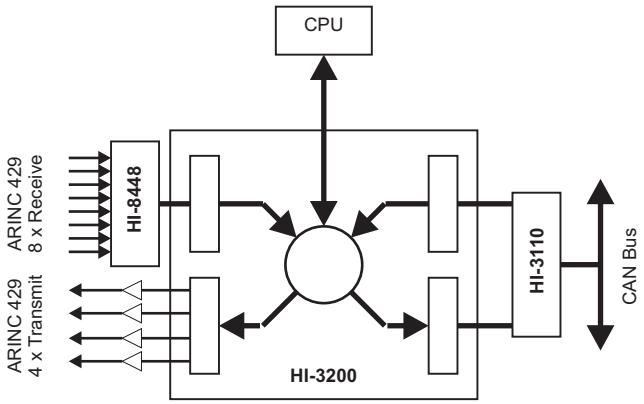
A 32K x 8 on-board memory allows received data to be logically organized and automatically updated as new ARINC 429 labels or CAN frames are received.

An auto-initialization feature allows configuration information to be up-loaded from an external EEPROM on reset to facilitate rapid start-up or operation without a host CPU.

The HI-3200 interfaces directly with Holt's HI-8448 octal ARINC 429 receiver IC, HI-8596 or HI-8592 ARINC 429 line drivers and HI-3110 integrated CAN controller / transceiver.

The HI-3201 is identical to the HI-3200 except it comes in an 80-pin PQFP package with eight instead of two ARINC 429 bit monitor pins.

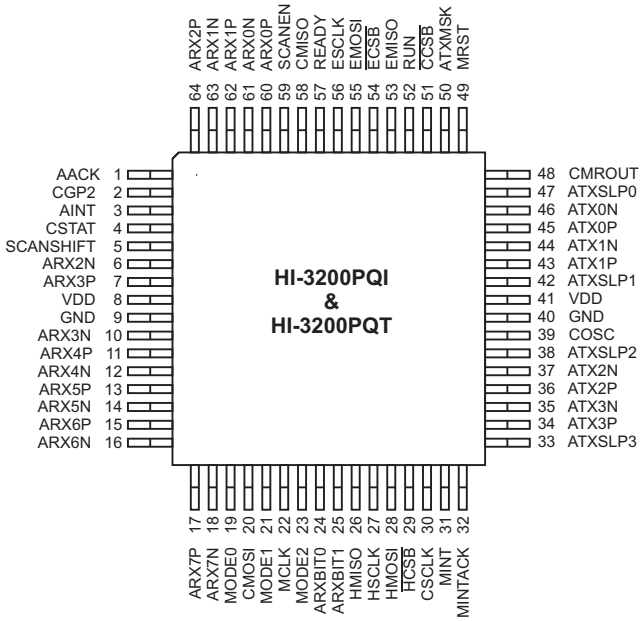
APPLICATION



FEATURES

- Eight ARINC 429 Receive channels
- Four ARINC 429 Transmit channels
- CAN Bus / ARINC 825 Interface
- 32KB on chip user-configurable data storage memory
- Programmable received data filtering for ARINC 429 and CAN buses
- Programmable transmission schedulers for periodic ARINC 429 and CAN message broadcasting
- Flexible protocol bridge ARINC 429 to CAN and CAN to ARINC 429
- SPI Host CPU interface
- Auto-initialization feature allows power-on configuration or independent operation without CPU

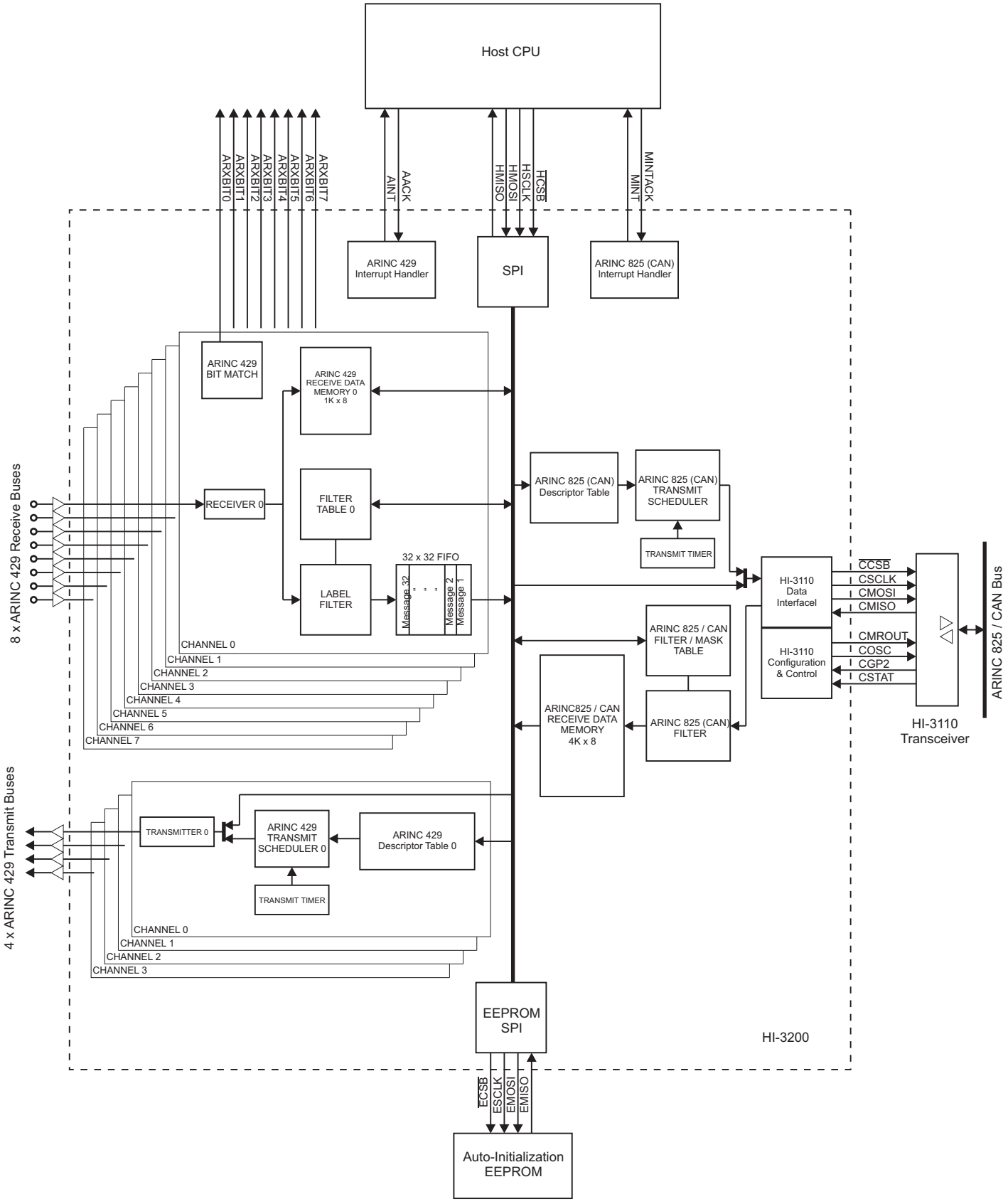
PIN CONFIGURATION



64 - Pin Plastic Quad Flat Pack (PQFP)

(See ordering information for additional pin configurations)

BLOCK DIAGRAM



APPLICATION OVERVIEW

The HI-3200 is a flexible device for managing ARINC 429 and ARINC 825 communications and data storage in many avionics applications. The device architecture centers around a 32K x 8 static RAM used for data storage, data filtering tables and table-driven transmission schedulers. Once configured, the device can operate autonomously without a host CPU, negating the need for software development or DO-178 certification. Configuration data may be uploaded into the device from an external EEPROM, following system reset.

The device supports up to eight ARINC 429 receive channels. Received data is stored in on-chip RAM organized by channel number and label. The data table continually updates as new labels arrive. Programmable interrupts and filters alert the host subsystem to labels of interest.

Each ARINC 429 receive channel also includes a 32 message deep FIFO allowing selected label data to be queued for subsequent host access.

The HI-3200 includes four independent ARINC 429 transmit channels. Transmission may be controlled entirely by an external CPU, or autonomously by programming one or more of the four on-chip ARINC 429 transmit schedulers. These allow periodic transmission to occur without CPU. Source data for transmission may be selected from RAM based tables of constants and / or from the received channel data. Powerful options exist for constructing ARINC 429 labels as well as controlling their timing and conditional transmission.

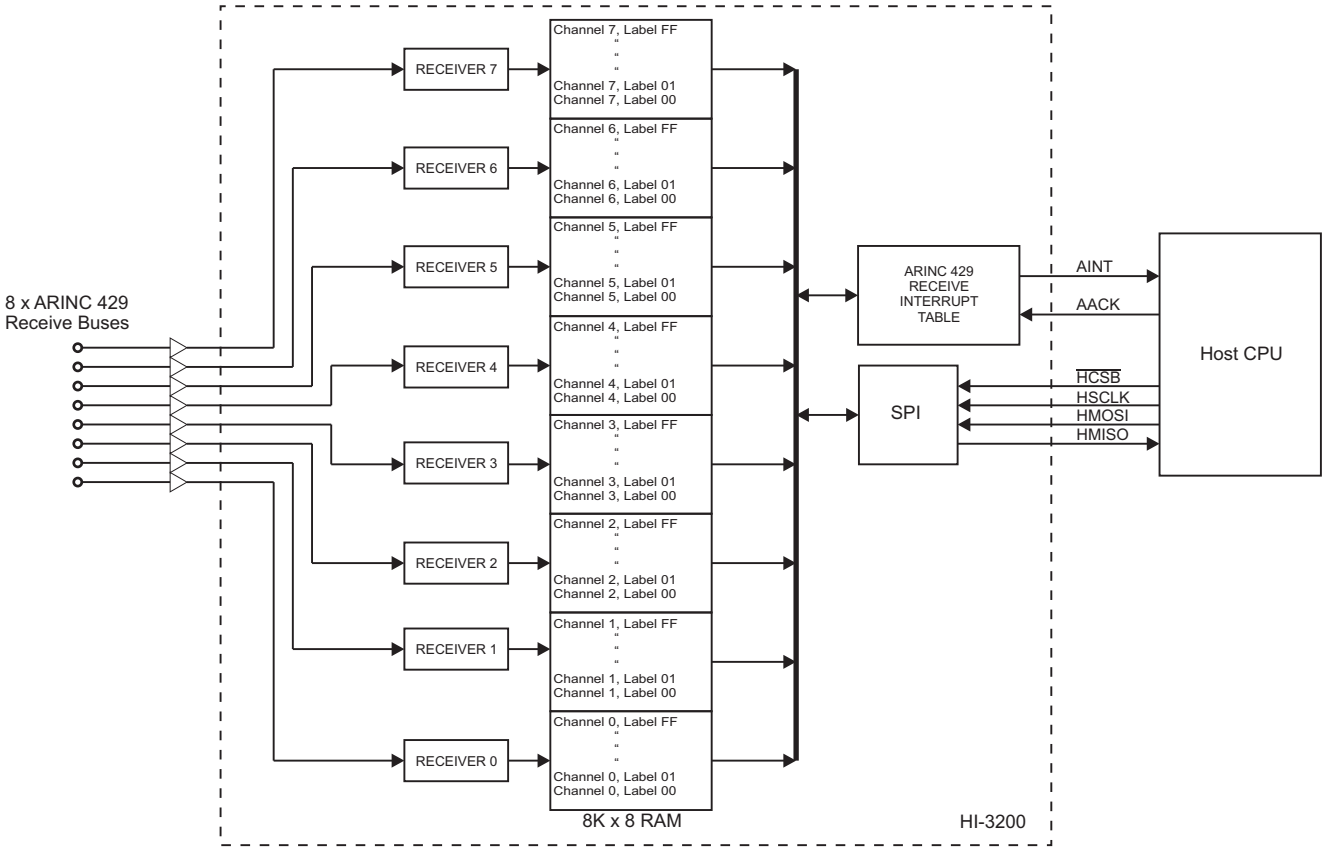
Even when running under the control of schedulers, the host CPU may insert new labels for transmission at will.

The HI-3200 also supports ARINC 825 (CAN) communication. An external HI-3110 CAN controller automatically handles the CAN bus protocol and physical interface. The HI-3200 configures the HI-3110 at system initialization and manages all traffic to and from the CAN bus.

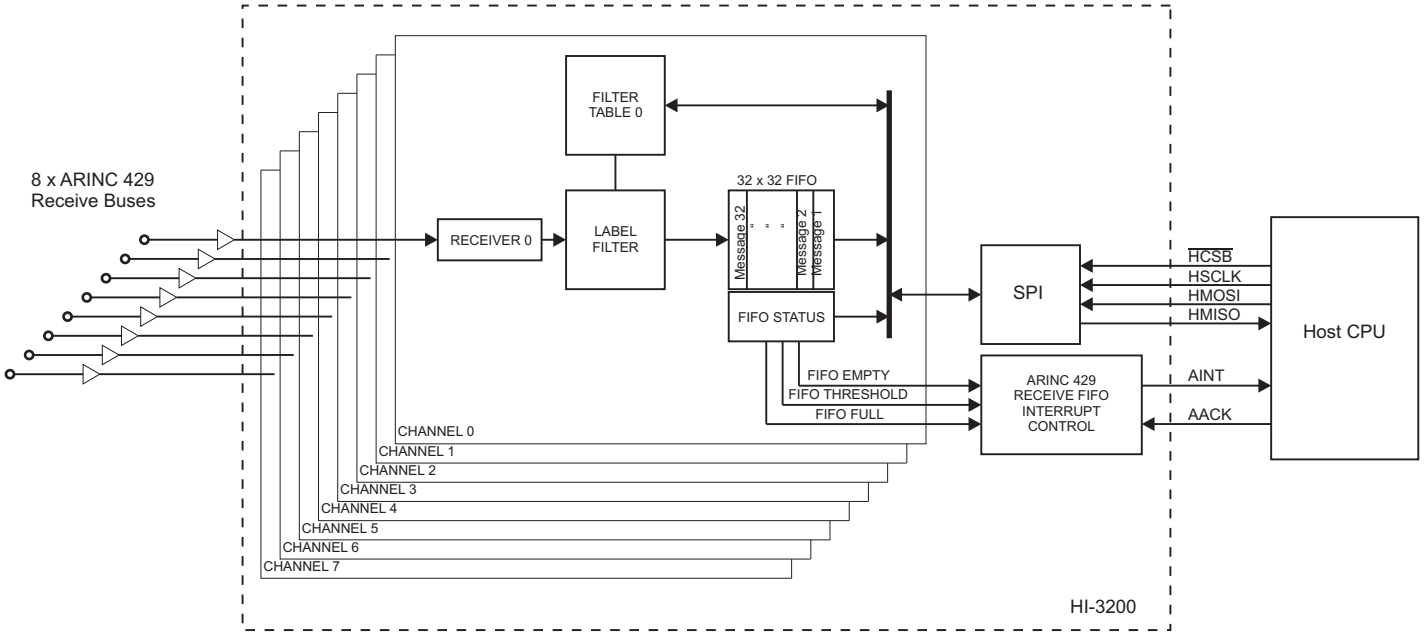
As with ARINC 429, ARINC 825 received data may be filtered and stored in on-chip RAM, organized by ID field filters. ARINC 825 frame transmission may be directly controlled by a host CPU or by an on-chip transmission scheduler. CAN frames may be built and conditionally transmitted using the scheduler's flexible instruction set. Source data for CAN frames can be from CPU, stored constants or from received ARINC 429 data tables.

The following examples show eight possible configurations of how the HI-3200 may be used:

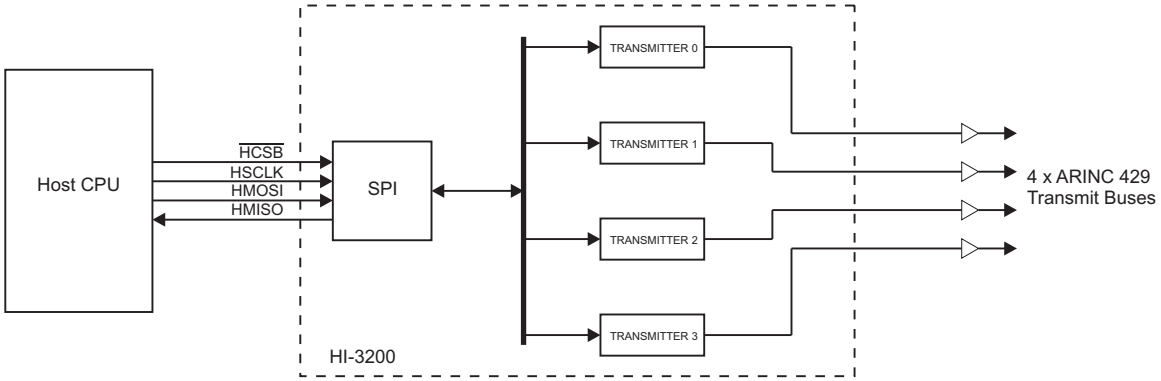
Example 1. ARINC 429 Data reception using on-chip RAM



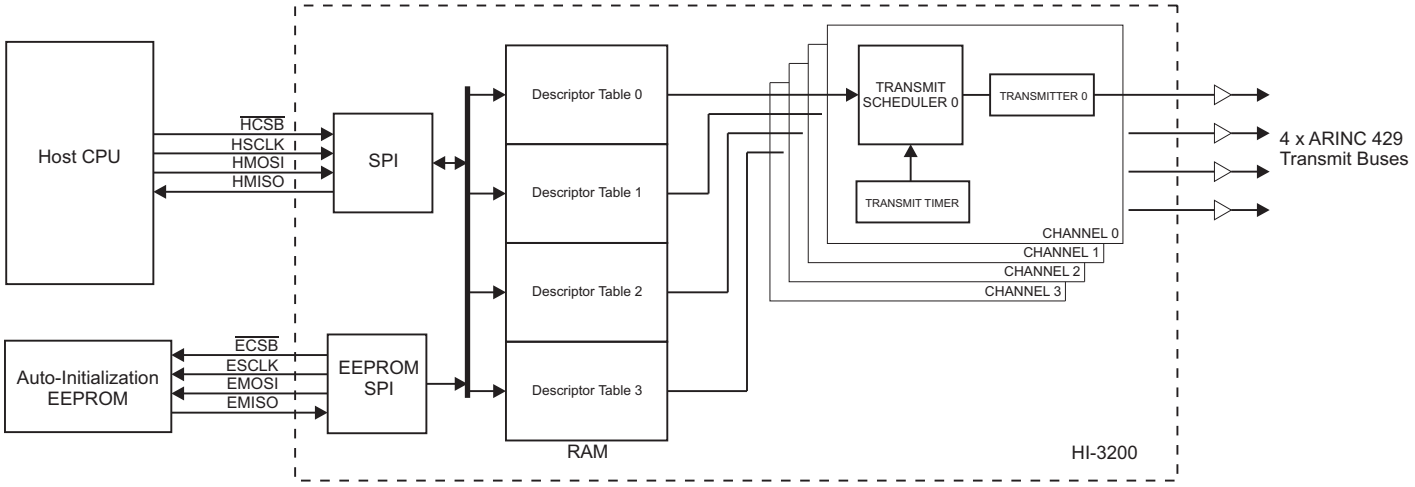
Example 2. ARINC 429 Data reception using on-chip filters and FIFOs



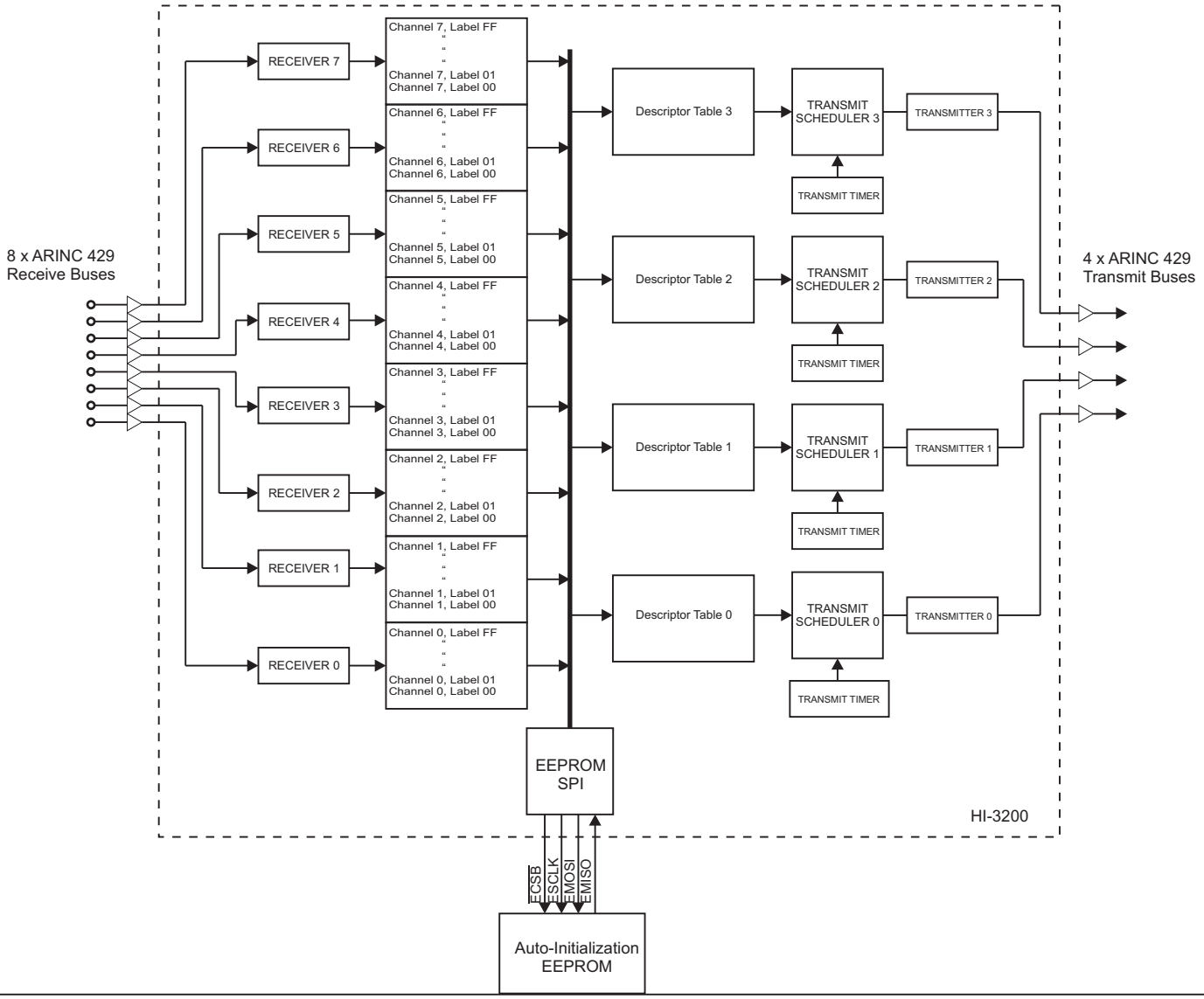
Example 3. ARINC 429 Data transmission directly from CPU



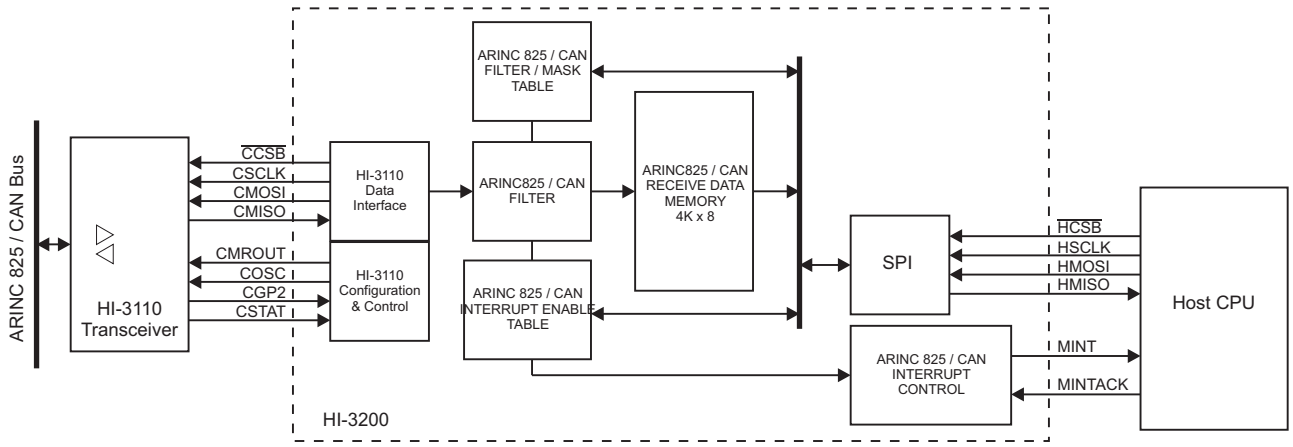
Example 4. ARINC 429 Data transmission using on-chip schedulers



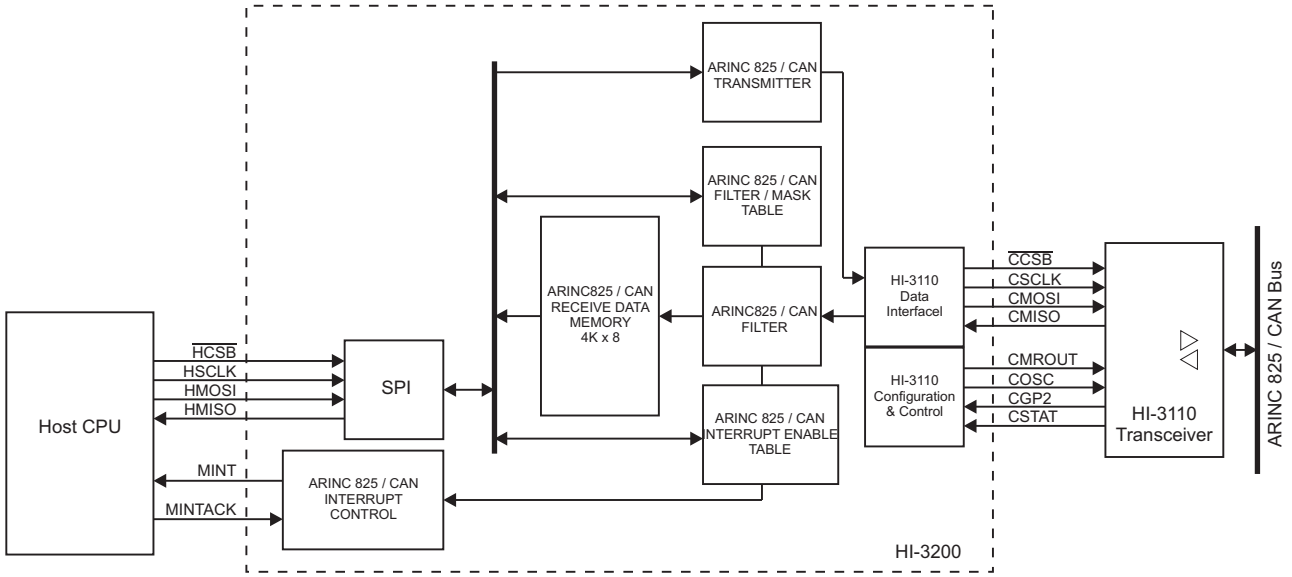
Example 5. Autonomous ARINC 429 Data Concentrator / Repeater



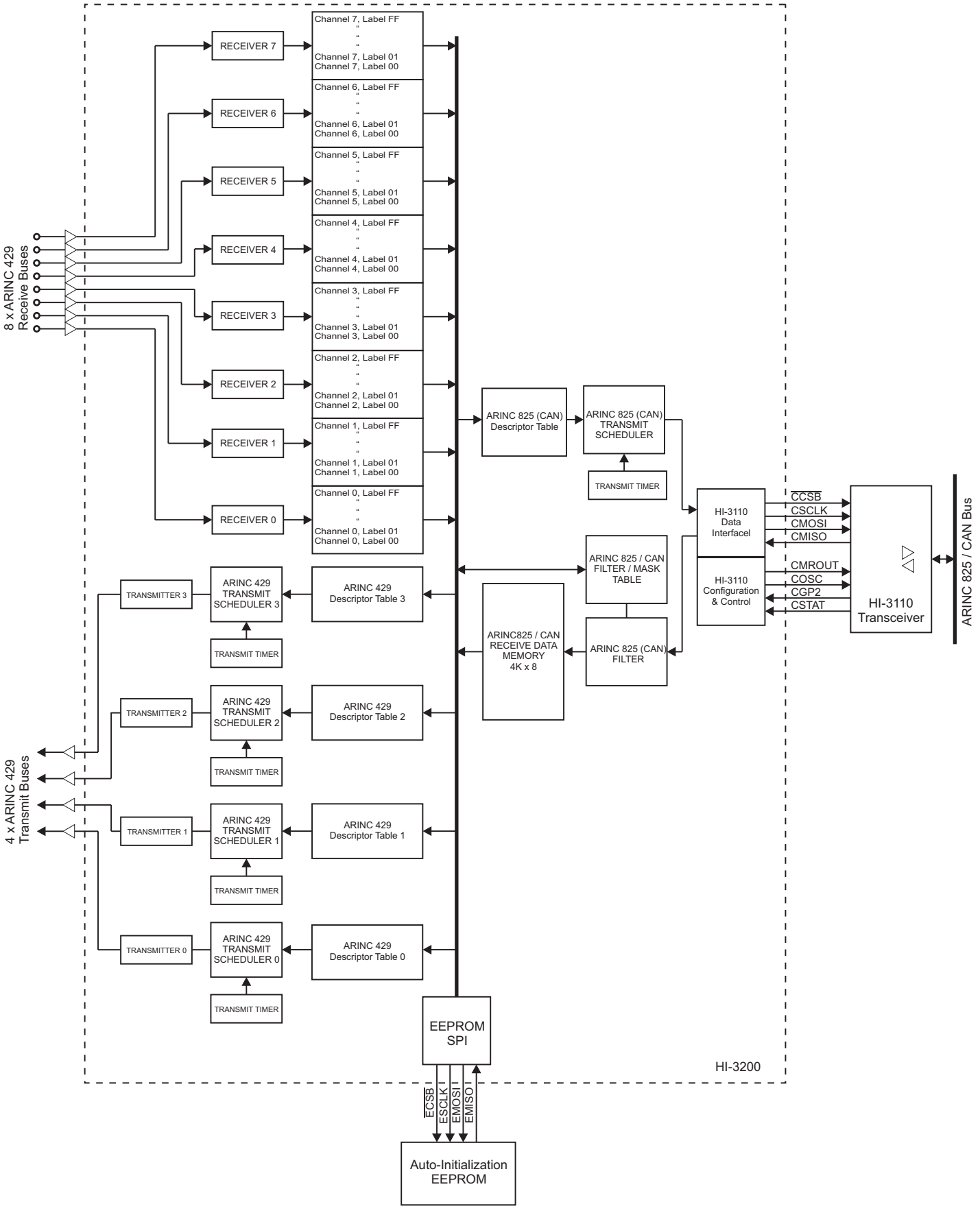
Example 6. ARINC 825 (CAN) bus Monitor / Receiver



Example 7. ARINC 825 (CAN) Terminal / Data Manager



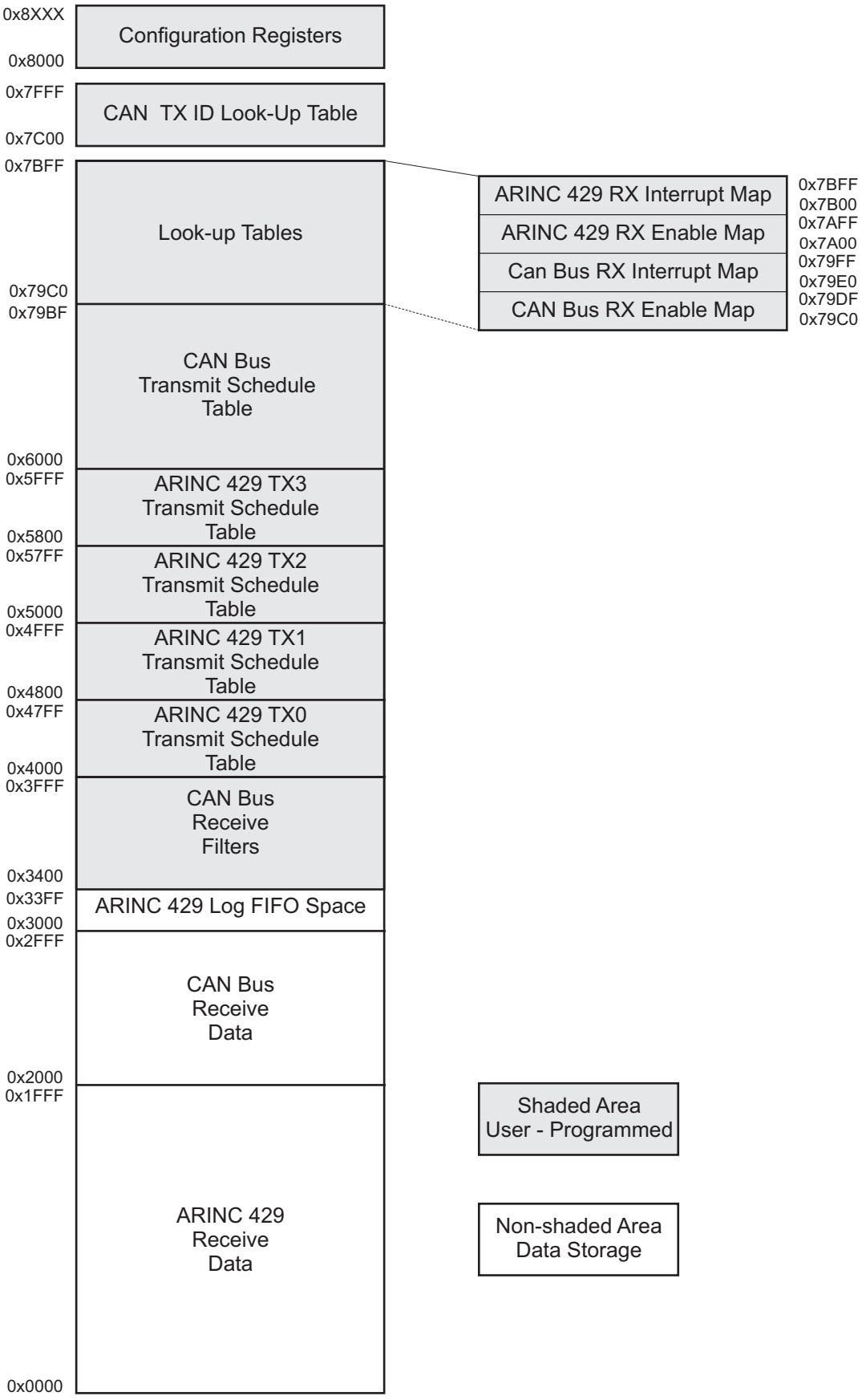
Example 8. ARINC 429 - ARINC 825 (CAN) Autonomous Bridge



PIN DESCRIPTIONS

SIGNAL	FUNCTION	DESCRIPTION	PULL UP/DOWN
AACK	INPUT	ARINC 429 Receiver Interrupt Acknowledge (Active High)	Pull-Down
AINT	OUTPUT	ARINC 429 Receiver Interrupt (Active High)	
ARX0N	INPUT	ARINC 429 Rx negative data input for channel 0	
ARX0P	INPUT	ARINC 429 Rx positive data input for channel 0	
ARX1N	INPUT	ARINC 429 Rx negative data input for channel 1	
ARX1P	INPUT	ARINC 429 Rx positive data input for channel 1	
ARX2N	INPUT	ARINC 429 Rx negative data input for channel 2	
ARX2P	INPUT	ARINC 429 Rx positive data input for channel 2	
ARX3N	INPUT	ARINC 429 Rx negative data input for channel 3	
ARX3P	INPUT	ARINC 429 Rx positive data input for channel 3	
ARX4N	INPUT	ARINC 429 Rx negative data input for channel 4	
ARX4P	INPUT	ARINC 429 Rx positive data input for channel 4	
ARX5N	INPUT	ARINC 429 Rx negative data input for channel 5	
ARX5P	INPUT	ARINC 429 Rx positive data input for channel 5	
ARX6N	INPUT	ARINC 429 Rx negative data input for channel 6	
ARX6P	INPUT	ARINC 429 Rx positive data input for channel 6	
ARX7N	INPUT	ARINC 429 Rx negative data input for channel 7	
ARX7P	INPUT	ARINC 429 Rx positive data input for channel 7	
ARXBIT0-1	OUTPUTS	ARINC 429 received payload bit monitor pins 0 and 1	
ARXBIT2-7	OUTPUTS	ARINC 429 received payload bit monitor pins 2 through 7 (HI-3201 only)	
ATX0N	OUTPUT	ARINC 429 Tx channel 0 negative data output to line driver	
ATX0P	OUTPUT	ARINC 429 Tx channel 0 positive data output to line driver	
ATX1N	OUTPUT	ARINC 429 Tx channel 1 negative data output to line driver	
ATX1P	OUTPUT	ARINC 429 Tx channel 1 positive data output to line driver	
ATX2N	OUTPUT	ARINC 429 Tx channel 2 negative data output to line driver	
ATX2P	OUTPUT	ARINC 429 Tx channel 2 positive data output to line driver	
ATX3N	OUTPUT	ARINC 429 Tx channel 3 negative data output to line driver	
ATX3P	OUTPUT	ARINC 429 Tx channel 3 positive data output to line driver	
ATXMSK	INPUT	Turn off ARINC 429 Transmit pins (Holds TXnA/B pins zero)	Pull-Down
ATXSLP0	OUTPUT	ARINC 429 Tx channel 0 data rate select output. 1 = high speed, 0 = low speed	
ATXSLP1	OUTPUT	ARINC 429 Tx channel 1 data rate select output. 1 = high speed, 0 = low speed	
ATXSLP2	OUTPUT	ARINC 429 Tx channel 2 data rate select output. 1 = high speed, 0 = low speed	
ATXSLP3	OUTPUT	ARINC 429 Tx channel 3 data rate select output. 1 = high speed, 0 = low speed	
$\overline{\text{CCSB}}$	OUTPUT	SPI chip select for HI-3110 CAN Controller / Transceiver	
CGP2	INPUT	Signal from HI-3110 CAN Controller / Transceiver indicating CAN Rx data is available	Pull-Down
CMISO	INPUT	SPI serial data input from HI-3110 CAN Controller / Transceiver	Pull-Down
CMOSI	OUTPUT	SPI serial data output to HI-3110 CAN Controller / Transceiver	
CMROUT	OUTPUT	Master Reset signal to HI-3110 CAN Controller / Transceiver	
COSC	OUTPUT	24 MHz CAN reference clock output to HI-3110 CAN Controller / Transceiver	
CSCLK	OUTPUT	SPI clock for HI-3110 CAN Controller / Transceiver	
CSTAT	INPUT	Signal from HI-3110 CAN Controller / Transceiver indicating CAN Tx FIFO is full	Pull-Down
$\overline{\text{ECSB}}$	OUTPUT	SPI chip select for auto-initialization EEPROM	
EMISO	INPUT	SPI serial data input from auto-inialization EEPROM	Pull-Down
EMOSI	OUTPUT	SPI serial data output to auto-initialization EEPROM	
ESCLK	OUTPUT	SPI clock for auto-initialization EEPROM (8MHz max.)	
GND	POWER	Chip 0V supply	
$\overline{\text{HCSB}}$	INPUT	Chip select. Data is shifted into HMOSI and out of HMISO when HCSB is low	Pull-Up
HMISO	OUTPUT	Host CPU SPI interface serial data output	
HMOSI	INPUT	Host CPU SPI interface serial data input	Pull-Down
HSCLK	INPUT	SPI Clock. Data is shifted into or out of the SPI interface using HSCLK	Pull-Down
MCLK	INPUT	Master 48 MHz (+/- 0.1%) reference clock for ARINC 429 and CAN bus bit timing	Pull-Down
MINT	OUTPUT	Programmable event interrupt output (Active High)	
MINTACK	INPUT	Programmable event interrupt acknowledge (Active High)	Pull-Down
MODE2:0	INPUTS	MODE2 through MODE0 define HI-3200 start-up and initialization mode	Pull-Downs
MRST	INPUT	Master Reset to HI-3200 Active High. See RESET AND START-UP section	Pull-Down
PROG (MODE0)	INPUT	Multiplexed with MODE0 pin, PROG initiates HI-3200 EEPROM Auto-Initialization	Pull-Down
READY	OUTPUT	READY goes high when post-RESET initialization is complete	
RUN	INPUT	Master enable signal for ARINC 429 and CAN transmit schedulers	Pull-Down
SCANEN	RESERVED	Connect to GND	
SCANSHIFT	RESERVED	Connect to GND	
VDD	POWER	3.3V power supply	

HI-3200 MEMORY MAP



HI-3200 REGISTER MAP

ADDRESS	R/W	REGISTER	MNEMONIC	DESCRIPTION
0x8000	R*	ARINC 429 Rx PENDING INTERRUPT	APIR	Defines channel(s) with pending Interrupt
0x8001	R	ARINC 429 Rx INTERRUPT ADDRESS 0	AIAR0	ARINC 429 Interrupt Vector channel 0
0x8002	R	ARINC 429 Rx INTERRUPT ADDRESS 1	AIAR1	ARINC 429 Interrupt Vector channel 1
0x8003	R	ARINC 429 Rx INTERRUPT ADDRESS 2	AIAR2	ARINC 429 Interrupt Vector channel 2
0x8004	R	ARINC 429 Rx INTERRUPT ADDRESS 3	AIAR3	ARINC 429 Interrupt Vector channel 3
0x8005	R	ARINC 429 Rx INTERRUPT ADDRESS 4	AIAR4	ARINC 429 Interrupt Vector channel 4
0x8006	R	ARINC 429 Rx INTERRUPT ADDRESS 5	AIAR5	ARINC 429 Interrupt Vector channel 5
0x8007	R	ARINC 429 Rx INTERRUPT ADDRESS 6	AIAR6	ARINC 429 Interrupt Vector channel 6
0x8008	R	ARINC 429 Rx INTERRUPT ADDRESS 7	AIAR7	ARINC 429 Interrupt Vector channel 7
0x8009	-	RESERVED		
0x800A	R*	PENDING INTERRUPT REGISTER	PIR	Indicates Interrupt type
0x800B	R	INTERRUPT ADDRESS REGISTER	CIAR	CAN bus Interrupt vector
0x800C	R	MUXED FIFO FLAGS	AMFF	ARINC 429 Multiplexed FIFO flags
0x800D	R	ARINC 429 TX READY BITS	ATRB	ARINC 429 Transmitter Ready flags
0x800E	R	MASTER STATUS REGISTER	MSR	Indicates HI-3200 current status
0x800F	R/W	MASTER CONTROL REGISTER	MCR	HI-3200 global configuration
0x8010	R/W	ARINC 429 RX CONTROL REGISTER 0	ARXC0	Configures ARINC 429 receive channel 0
0x8011	R/W	ARINC 429 RX CONTROL REGISTER 1	ARXC1	Configures ARINC 429 receive channel 1
0x8012	R/W	ARINC 429 RX CONTROL REGISTER 2	ARXC2	Configures ARINC 429 receive channel 2
0x8013	R/W	ARINC 429 RX CONTROL REGISTER 3	ARXC3	Configures ARINC 429 receive channel 3
0x8014	R/W	ARINC 429 RX CONTROL REGISTER 4	ARXC4	Configures ARINC 429 receive channel 4
0x8015	R/W	ARINC 429 RX CONTROL REGISTER 5	ARXC5	Configures ARINC 429 receive channel 5
0x8016	R/W	ARINC 429 RX CONTROL REGISTER 6	ARXC6	Configures ARINC 429 receive channel 6
0x8017	R/W	ARINC 429 RX CONTROL REGISTER 7	ARXC7	Configures ARINC 429 receive channel 7
0x8018	R/W	ARINC 429 TX CONTROL REGISTER 0	ATXC0	Configures ARINC 429 transmit channel 0
0x8019	R/W	ARINC 429 TX CONTROL REGISTER 1	ATXC1	Configures ARINC 429 transmit channel 1
0x801A	R/W	ARINC 429 TX CONTROL REGISTER 2	ATXC2	Configures ARINC 429 transmit channel 2
0x801B	R/W	ARINC 429 TX CONTROL REGISTER 3	ATXC3	Configures ARINC 429 transmit channel 3
0x801C	R/W	ARINC 429 TX REPETITION RATE 0	ATXRR0	Sets sequence repeat time for ARINC TX0
0x801D	R/W	ARINC 429 TX REPETITION RATE 1	ATXRR1	Sets sequence repeat time for ARINC TX1
0x801E	R/W	ARINC 429 TX REPETITION RATE 2	ATXRR2	Sets sequence repeat time for ARINC TX2
0x801F	R/W	ARINC 429 TX REPETITION RATE 3	ATXRR3	Sets sequence repeat time for ARINC TX3
0x8020	R/W	ARINC 429 Rx INTERRUPT MASK	AIMR	Enables Interrupts on AINT pin
0x8021	R/W	ARINC 429 Rx FIFO THRESHOLD VALUE	AFTV	Sets flag value for ARINC 429 Receive FIFO
0x8022	R/W	ARINC 429 LOOPBACK	ALOOP	Sets loop-back self-test mode
0x8029	R	ARINC 429 Rx FIFO FULL FLAG	AFFF	Indicates which FIFOs are full
0x802A	R	ARINC 429 Rx FIFO THRESHOLD FLAG	AFTF	Indicates which FIFOs hold > (thresh) messages
0x802B	R	ARINC 429 Rx FIFO NOT EMPTY FLAG	FFNE	Indicates which receive FIFOs hold data
0x802C	R	ARINC 429 TX SEQUENCE POINTER 0	ATXSP0	Current address of ARINC transmit sequence 0
0x802D	R	ARINC 429 TX SEQUENCE POINTER 1	ATXSP1	Current address of ARINC transmit sequence 1
0x802E	R	ARINC 429 TX SEQUENCE POINTER 2	ATXSP2	Current address of ARINC transmit sequence 2
0x802F	R	ARINC 429 TX SEQUENCE POINTER 3	ATXSP3	Current address of ARINC transmit sequence 3
0x8030	R/W	CAN BUS BIT TIMING REGISTER 0	CANBTR0	Sets bit timing parameters for CAN bus
0x8031	R/W	CAN BUS BIT TIMING REGISTER 1	CANBTR1	Sets bit timing parameters for CAN bus
0x8032	R/W	CAN TRANSMIT CONTROL REGISTER	CANTXC	Controls CAN bus transmit scheduler
0x8033	R/W	CAN TX REPETITION RATE	CANTXRR	Sets sequence repeat time for CAN transmitter
0x8034	R/W	PENDING INTERRUPT ENABLE REGISTER	IMR	Enables Interrupts on MINT pin
0x8035	R/W	ARINC 429 TX READY INT ENABLE	ATRIE	Enables ARINC 429 TX Ready Interrupts
0x803E	R	CAN TX SEQUENCE POINTER MSB	CANTXSPH	High order CAN transmit sequence counter
0x803F	R	CAN TX SEQUENCE POINTER LSB	CANTXSPL	Low order CAN transmit sequence counter

Fast Access Registers

Memory mapped register access only

* Register is cleared when read (auto clear)

ADDRESS	R/W	REGISTER	MNEMONIC	DESCRIPTION
0x805F	R/W	PINS ARXBIT[7:0]	ARXBIT	Values of pins ARXBIT[7:0]
0x8060	R/W	PIN ARXBIT0 CONFIG REG 1	ARX0CR1	ARINC 429 bit Monitor 0 channel & bit select
0x8061	R/W	PIN ARXBIT0 CONFIG REG 2	ARX0CR2	ARINC 429 bit Monitor 0 label select
0x8062	R/W	PIN ARXBIT1 CONFIG REG 1	ARX1CR1	ARINC 429 bit Monitor 1 channel & bit select
0x8063	R/W	PIN ARXBIT1 CONFIG REG 2	ARX1CR2	ARINC 429 bit Monitor 1 label select
0x8064	R/W	PIN ARXBIT2 CONFIG REG 1	ARX2CR1	ARINC 429 bit Monitor 2 channel & bit select
0x8065	R/W	PIN ARXBIT2 CONFIG REG 2	ARX2CR2	ARINC 429 bit Monitor 2 label select
0x8066	R/W	PIN ARXBIT3 CONFIG REG 1	ARX3CR1	ARINC 429 bit Monitor 3 channel & bit select
0x8067	R/W	PIN ARXBIT3 CONFIG REG 2	ARX3CR2	ARINC 429 bit Monitor 3 label select
0x8068	R/W	PIN ARXBIT4 CONFIG REG 1	ARX4CR1	ARINC 429 bit Monitor 4 channel & bit select
0x8069	R/W	PIN ARXBIT4 CONFIG REG 2	ARX4CR2	ARINC 429 bit Monitor 4 label select
0x806A	R/W	PIN ARXBIT5 CONFIG REG 1	ARX5CR1	ARINC 429 bit Monitor 5 channel & bit select
0x806B	R/W	PIN ARXBIT5 CONFIG REG 2	ARX5CR2	ARINC 429 bit Monitor 5 label select
0x806C	R/W	PIN ARXBIT6 CONFIG REG 1	ARX6CR1	ARINC 429 bit Monitor 6 channel & bit select
0x806D	R/W	PIN ARXBIT6 CONFIG REG 2	ARX6CR2	ARINC 429 bit Monitor 6 label select
0x806E	R/W	PIN ARXBIT7 CONFIG REG 1	ARX7CR1	ARINC 429 bit Monitor 7 channel & bit select
0x806F	R/W	PIN ARXBIT7 CONFIG REG 2	ARX7CR2	ARINC 429 bit Monitor 7 label select
0x8070	R/W	BIST CONTROL/STATUS	BISTS	Built-In Self-Test (bits 1,0 are read only)
0x8071	R	BIST FAIL ADDRESS [7:0]	BISTFL	Low-order failing BIST memory address
0x8072	R	BIST FAIL ADDRESS [12:8]	BISTFH	High-order failing BIST memory address
0x8073	R	AUTO-INIT FAIL LS ADDRESS [7:0]	AIFL	Auto-initialization fail address (low-byte)
0x8074	R	AUTO-INIT FAIL MS ADDRESS [15:8]	AIFH	Auto-initialization fail address (high byte)

HI-3200 SYSTEM CONFIGURATION

Starting at memory address 0x8000, the HI-3200 contains a set of registers that are used to configure the HI-3200 device and, if used, its associated HI-3110 integrated CAN controller / transceiver.

The user needs only to program the HI-3200 configuration registers to completely define the full system operation.

Configuration information for the HI-3110 is automatically transferred from the HI-3200 to the HI-3110 immediately after the RUN input is asserted.

An SPI by-pass mode allows the user to directly access the HI-3110, but it is highly recommended that this is used solely for design debugging purposes and is locked out in the final design implementation. By-pass mode is enabled by setting the state of the MODE2:0 pins during reset. See the Reset and Start-Up Configuration section for more details.

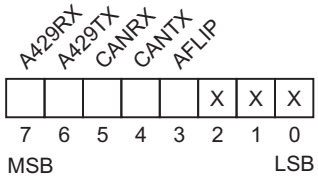
The configuration registers are divided into four categories, as follows;

1. HI-3200 global configuration
2. ARINC 429 Receive channel configuration
3. ARINC 429 Transmit channel configuration
4. CAN Bus bit timing configuration

HI-3200 Global Configuration

The following registers define the HI-3200 top-level configuration:

MASTER CONTROL REGISTER
(Address 0x800F)



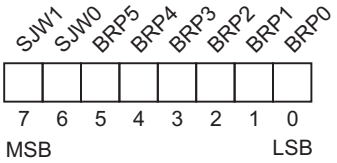
Bit	Name	R/W	Default	Description
7	A429RX	R/W	0	This bit must be set to a “1” to allow the HI-3200 to receive ARINC 429 data on any of the eight channels. If set to a zero, the HI-3200 will not respond to any ARINC 429 receive bus, regardless of the state of the ARINC 429 Receive channel Control Registers.
6	A429TX	R/W	0	This bit must be set to a “1” to allow the HI-3200 to transmit ARINC 429 data on any of the four channels. If set to a zero, the HI-3200 will not output ARINC 429 data and the ARINC 429 transmit sequencers will remain in their reset state.
5	CANRX	R/W	0	This bit must be set to a “1” to allow the HI-3200 to receive CAN Frames from the HI-3110 controller. If set to a zero, the HI-3200 will not respond to any received CAN frames, regardless of the state of the CAN Bus Control Register.
4	CANTX	R/W	0	This bit must be set to a “1” to allow the HI-3200 to transmit CAN frames. If set to a zero, the HI-3200 will not output CAN frames and the CAN transmit sequencer will remain in its reset state.
3	AFLIP	R/W	0	When set to a “1”, this bit switches the bit order of the ARINC 429 label byte in both receive and transmit channels.
2	-	R/W	0	Not Used
1	-	R/W	0	Not Used
0	-	R/W	0	Not Used

CAN Bus Timing Configuration

Two registers must be programmed to define the CAN bus data rate and bit sampling segment times. This information is transferred directly to the HI-3110 CAN controller's BTR0 and BTR1 registers following the rising edge of the RUN input.

The HI-3110 OSCIN clock frequency must be set to achieve the desired bit rate. The HI-3200 COSC output signal provides a convenient 24MHz clock source for the HI-3110. For a full description of CAN Bus timing requirements, please refer to the Holt HI-3110 data sheet.

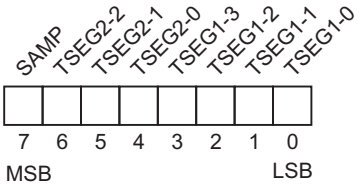
CANBTR0
(Address 0x8030)



CANBTR0 defines the value of the Re-synchronization Jump Width (SJW) and the Baud Rate Prescaler (BRP).

Bit	Name	R/W	Default	Description
7:6	SJW1:0	R/W	0	These bits are used to compensate for phase shifts between different oscillators on the CAN bus. They define the maximum number of time quanta (Tq) a bit can be shortened or lengthened to allow the node to achieve re-synchronization to the edge of an incoming signal. Note that the time quantum (Tq) is the single unit of time within a bit time.
5:0	BRP5:0	R/W	0	The baud rate prescaler relates the HI-3110 OSCIN clock frequency, fosc, to the CAN bit time as described in the HI-3110 data sheet. BRP bits <5:0> 000000: BRP=1 000001: BRP=2 000010: BRP=3 000011: BRP=4 etc. 111111: BRP=64

CANBTR1
(Address 0x8031)



CANBTR1 configures the CAN protocol bit timing segments in terms of time quanta (Tq) and sets the number of sampling points.

Bit	Name	R/W	Default	Description
7	SAMP	R/W	0	This bit configures how many samples are taken per bit. 1 = three samples/bit, 0 = one sample/bit. Bit sampling occurs at the end of Phase Seg 1. Note: ARINC825 states that there shall be only one sample per bit
5:0	TSEG2-2:0	R/W	0	Time segment 2 length. Tseg2 = Phase Seg2 of the CAN bit timing specification. Bits TSEG2-2:0 specify the number of time quanta in Phase Seg2. Note: Not all combinations are valid, since Phase Seg2 must be greater than SJW. TSEG2 bits <2:0> 000: Not valid 001: TSeg2 = 2 Tq clock cycles 010: TSeg2 = 3 Tq clock cycles etc. 111: TSeg2 = 8 Tq clock cycles

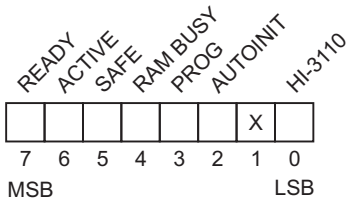
CANBTR1 cont.

Bit	Name	R/W	Default	Description
5:0	TSEG1-3:0	R/W	0	Time segment 1 length. Tseg1 = Prop Seg + Phase Seg 1 of the CAN protocol bit timing specification. Bits TSEG1-3:0 specify the number of time quanta in Prop Seg + Phase Seg1. Note: Not all combinations are valid since Prop Seg = Phase Seg1 >= Phase Seg2. **The CAN protocol states that the minimum number of Tq in a bit time shall be 8.** Note ARINC 825 states that the sample point shall not be less than 75% of the bit time. In this case, TSeg1 should be a minimum of 5Tq for Phase Seg2 (TSeg2) = 2Tq and SJW = 1Tq. TSEG2 bits <2:0> 0000: Not valid 0001: TSeg1 = 2 Tq clock cycles 0010: TSeg1 = 3 Tq clock cycles etc. 1111: TSeg1 = 16 Tq clock cycles

HI-3200 Operational Status Information

The Master Status Register may be read at any time to determine the current operational state of the HI-3200:

MASTER STATUS REGISTER
(Address 0x800E)



Bit	Name	R/W	Default	Description
7	READY	R	0	This bit is high, when the READY output pin is high, indicating that the part is able to accept and respond to host CPU SPI commands
6	ACTIVE	R	0	This bit is high after RUN is asserted and the HI-3200 is in normal operating mode.
5	SAFE	R	0	This bit goes high when the part enters safe mode as a result of a Built-in Self-test fail or auto-initialization fail.
4	RAM BUSY	R	0	This is high during the time the RAM Integrity Check is running and RAM is clearing
3	PROG	R	0	Indicates that the HI-3200 is currently in the EEPROM programming cycle. Note that READY stays low until the cycle is complete.
2	AUTOINIT	R	0	The HI-3200 is currently loading internal memory, registers and look-up tables from the Auto-initialization EEPROM
1	-	R	0	Not used
0	HI-3110	R	0	The HI-3200 has detected the presence of an HI-3110 device connected to the CAN SPI port. **Note:** Only valid when RUN = 1 and CANTX and/or CANRX are enabled. After HI-3110 initialization this bit is not updated in Mode 6 or 7.

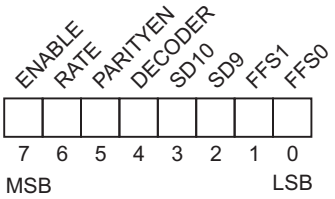
ARINC 429 RECEIVE OPERATION

The HI-3200 can receive ARINC 429 messages from up to eight ARINC 429 receive buses. External analog line receivers handle the physical layer connection

ARINC 429 Receive Channel Configuration

Each of the eight possible ARINC 429 Receive channels is configured using its own Control Register. Register address 0x8010 controls ARINC 429 Receive channel #0, register address 0x8011 controls channel #1 and so on. ARINC 429 Receive Control Registers may be read at any time, but can only be written when the device is in the IDLE state (RUN input = “0”, READY output = “1”).

ARINC 429 RX CONTROL REGISTER 0 - 7
(Address 0x8010 - 0x8017)



Bit	Name	R/W	Default	Description
7	ENABLE	R/W	0	This bit must be set to a “1” to enable ARINC 429 data reception on this channel.
6	$\overline{HI}$ / LO	R/W	0	Selects the ARINC 429 bit rate for the ARINC 429 receive channel. A “0” selects high-speed (100Kb/s) and a “1” selects low-speed (12.5Kb/s).
5	PARITYEN	R/W	0	When this bit is a one, the 32nd received ARINC bit is overwritten with a parity flag. The flag bit is set to a zero when the received ARINC word, including its parity bit has an odd number of ones. When PARITYEN is a zero, all 32-bits are received without parity checking.
4	DECODER	R/W	0	When DECODER is a “1”, bits 9 and 10 of ARINC 429 words received on this channel must match the SD9 and SD10 bits in the register. ARINC words received that do not match the SD conditions are ignored.
3	SD10	R/W	0	If DECODER is set to a “1”, then this bit must match the received ARINC word bit 10 for the word to be accepted.
2	SD9	R/W	0	If DECODER is set to a “1”, then this bit must match the received ARINC word bit 9 for the word to be accepted.
1-0	FFS1:0	R/W	0	FFS1 and FFS0 define when this channel’s FIFO Flag is set, as shown below. The FIFO Flag will be set in the ARINC 429 Muxed FIFO Flags Register (0x800C). Bit 0 is receive channel 0 FIFO Flag, bit 1 is receive channel 1 FIFO Flag, bit 2 is receive channel 2 FIFO Flag, etc.

FFS1	FFS0	FLAG set condition
0	0	FLAG never set
0	1	Set FLAG if FIFO NOT EMPTY bit = “1”
1	0	Set FLAG if FIFO > Threshold value
1	1	Set FLAG is FIFO FULL bit “1”

ARINC 429 Received Data Management

The HI-3200 supports eight ARINC 429 receive buses using on-chip receivers to handle the protocol validation. The eight ARINC 429 RX Control Registers, ARXC0 - 7, define the characteristics of each receive channel.

The ARINC 429 receive function of the HI-3200 is activated by setting the A429RX bit in the Master Control Register.

When an ARINC 429 message is received by the HI-3200 on any bus, it is checked for protocol compliance. Messages with incorrect encoding are rejected.

The HI-3200 contains an 8K byte memory for storing ARINC 429 received data. The memory is organized by channel number and ARINC 429 label value. Four bytes of memory are dedicated to each channel / label to store the 32-word ARINC 429 message.

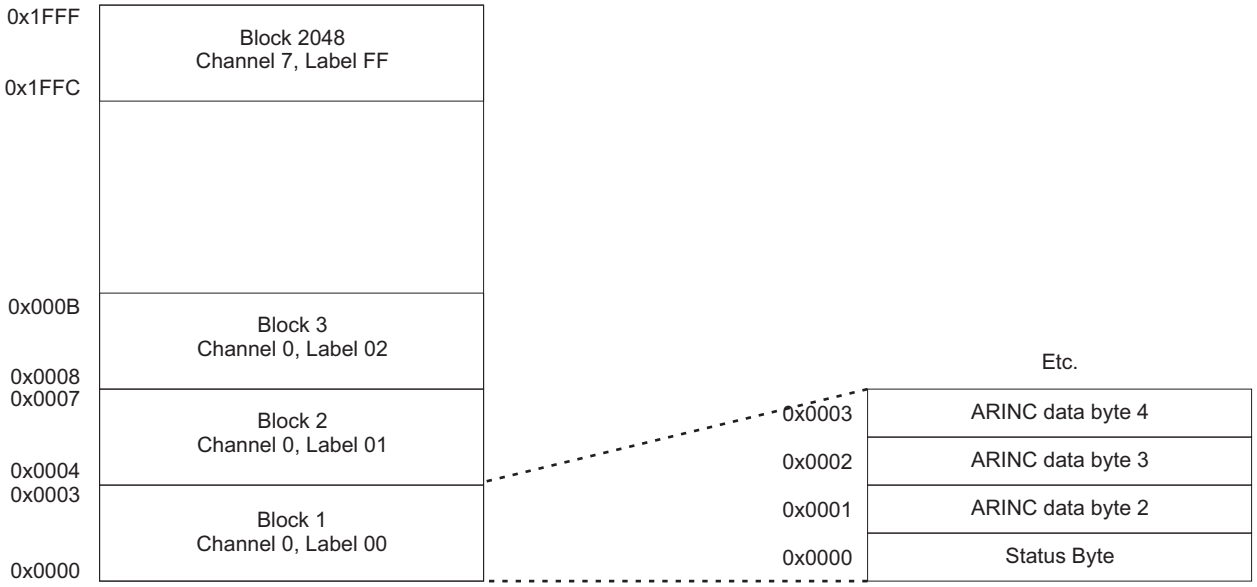
A look-up table is used to enable an interrupt on receipt of a new ARINC 429 message. Look-up table bit positions preloaded with a "1" will cause an Interrupt to be generated.

When a message is received that triggers an Interrupt, that channel's Interrupt bit is set in the ARINC 429 Receive Pending Interrupt Register. If this bit is unmasked in the ARINC 429 Receive Interrupt Mask Register, the AINT output pin is asserted. The label number of the ARINC 429 message causing the interrupt is loaded into that channel's ARINC 429 Receive Interrupt Address Register (AIAR0 - AIAR7).

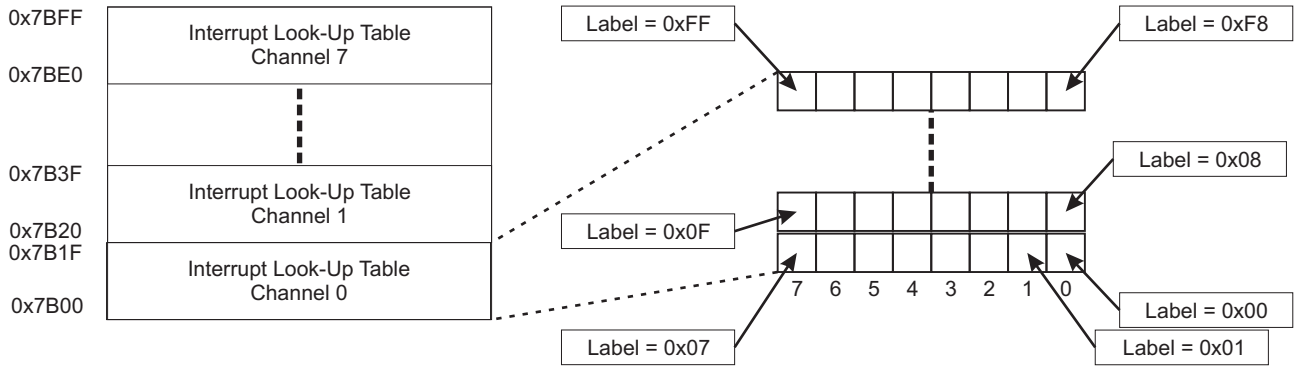
Because the ARINC Receive Memory is organized by label value, it is not necessary to store the received label value (first eight bits of the ARINC message) in the memory. Instead, the first byte is used to store a status byte.

The six active bits of the status byte are set to "1" when a new ARINC word is stored in the memory. These bits flag the ARINC word as new when the location is interrogated by the host CPU, any of the four ARINC 429 transmit schedulers or the CAN Bus transmit scheduler.

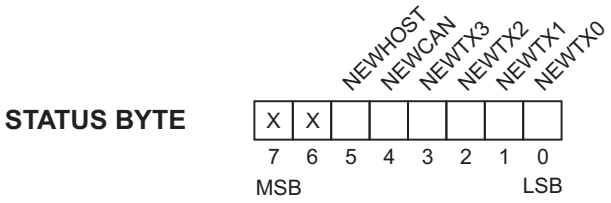
ARINC 429 Received Data Memory Organization



ARINC 429 Received Data Interrupt Look-Up Table



ARINC 429 Received Data Status Byte Definition



Bit	Name	R/W	Default	Description
7	-	R/W	0	Not used
6	-	R/W	0	Not used
5	NEWHOST	R/W	0	This bit is set when a new ARINC 429 word is received and stored in this block. It is reset when the host CPU executes SPI instruction 0xC0 - 0xFF to read the block.
4	NEWCAN	R/W	0	This bit is set when a new ARINC 429 word is received and stored in this block.It is reset when the CAN Transmit scheduler reads any bytes from the block.
3	NEWTX3	R/W	0	This bit is set when a new ARINC 429 word is received and stored in this block. It is reset when the ARINC 429 Transmit scheduler #3 reads any bytes from the block.
2	NEWTX2	R/W	0	This bit is set when a new ARINC 429 word is received and stored in this block. It is reset when the ARINC 429 Transmit scheduler #2 reads any bytes from the block.
1	NEWTX1	R/W	0	This bit is set when a new ARINC 429 word is received and stored in this block. It is reset when the ARINC 429 Transmit scheduler #1 reads any bytes from the block.
0	NEWTX0	R/W	0	This bit is set when a new ARINC 429 word is received and stored in this block. It is reset when the ARINC 429 Transmit scheduler #0 reads any bytes from the block.

ARINC 429 Received Data Log FIFO

A 1K x 8 block of memory located between 0x3000 and 0x33FF is reserved for a set of eight ARINC 429 received data FIFOs. There is one FIFO for each ARINC 429 received data channel. Each FIFO can hold up to 32 ARINC 429 32-bit messages.

A look-up table driven filter defines which ARINC 429 messages are stored in each FIFO. The look-up table is initialized by the user with a "1" for each bit position corresponding to a selected channel / label combination. The look-up table is located at memory address 0x7A00.

When a new ARINC 429 message is received that meets the programmed conditions for acceptance (Enable look-up table bit = "1"), it is written into the channel's Receive Data FIFO. The contents of the FIFO may be read by the host CPU using dedicated FIFO read SPI Instructions.

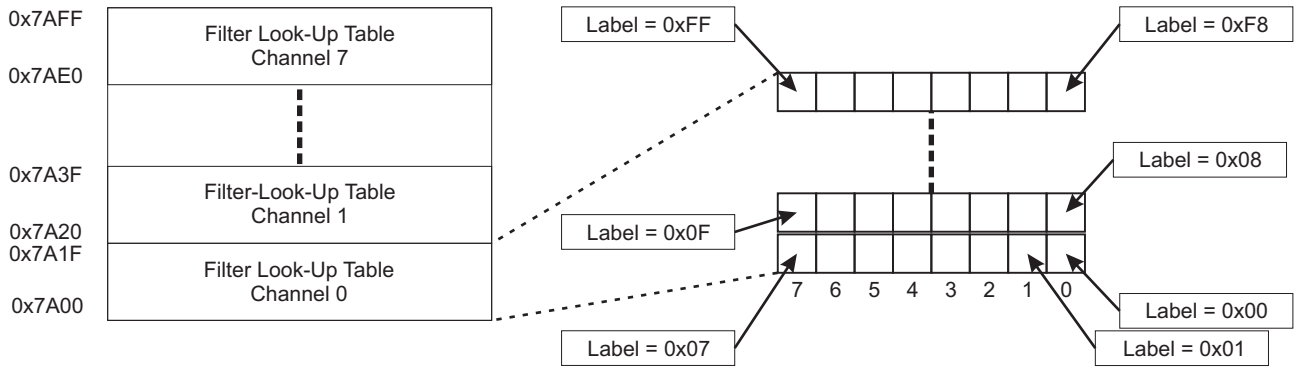
The status of each channel's FIFOs is monitored by three FIFO status registers: FIFO NOT EMPTY, FIFO THRESHOLD, and FIFO FULL. One bit of each register reflects the current status of each FIFO.

The FIFOs are empty following reset. All three status registers are cleared. When an ARINC 429 message is written to a FIFO, its FIFO NOT EMPTY bit is set to a "1". When the FIFO contains more than the user-defined number of messages as programmed in the ARINC FIFO THRESHOLD VALUE register, its FIFO THRESHOLD bit is set. If the FIFO is allowed to accumulate 32 messages, its FIFO FULL bit is set. Once a FIFO is full, subsequent messages continue to be written to the FIFO, and the oldest message is lost.

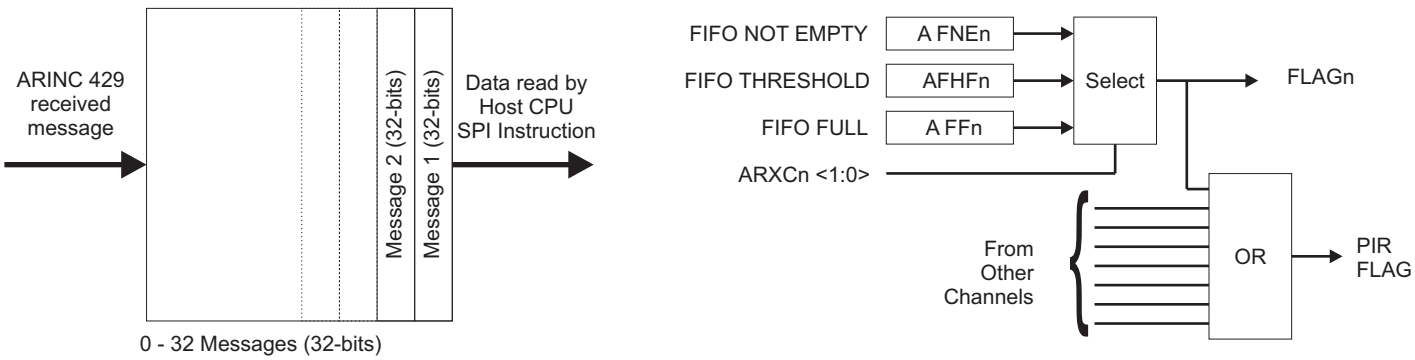
The user may generate an Interrupt by enabling one of the three FIFO status register bits to assert the FLAG bit in the Pending Interrupt Register. ARINC 429 Control Register bits 1:0 select the condition to trigger the FLAG interrupt.

The FIFO feature is particularly useful if the application wishes to accumulate sequential ARINC 429 messages of the same label value before reading them. The regular ARINC 429 receive data memory will, of course, overwrite messages of the same label value if a new message is received before the host CPU extracts the data.

ARINC 429 Received Data Enable Look-Up Table

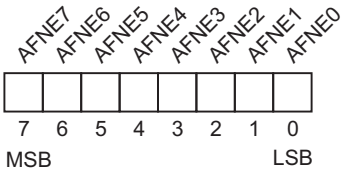


ARINC 429 Received Data FIFO (x8)



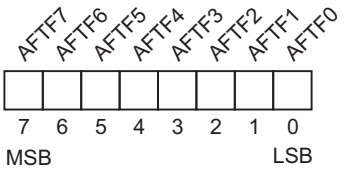
ARINC 429 Received Data FIFO Status Registers

FIFO NOT EMPTY REGISTER
(Address 0x802B)



Bit	Name	R/W	Default	Description
7	AFNE7	R	0	This bit is set to "1" if FIFO #7 contains at least one ARINC 429 message
6	AFNE6	R	0	This bit is set to "1" if FIFO #6 contains at least one ARINC 429 message
5	AFNE5	R	0	This bit is set to "1" if FIFO #5 contains at least one ARINC 429 message
4	AFNE4	R	0	This bit is set to "1" if FIFO #4 contains at least one ARINC 429 message
3	AFNE3	R	0	This bit is set to "1" if FIFO #3 contains at least one ARINC 429 message
2	AFNE2	R	0	This bit is set to "1" if FIFO #2 contains at least one ARINC 429 message
1	AFNE1	R	0	This bit is set to "1" if FIFO #1 contains at least one ARINC 429 message
0	AFNE0	R	0	This bit is set to "1" if FIFO #0 contains at least one ARINC 429 message

FIFO THRESHOLD REGISTER
(Address 0x802A)

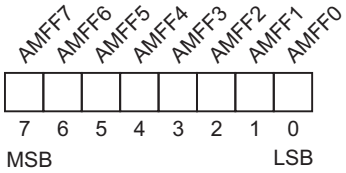


Bit	Name	R/W	Default	Description
7	AFTF7	R	0	This bit is set to "1" if FIFO #7 contains > threshold number of ARINC 429 messages
6	AFTF6	R	0	This bit is set to "1" if FIFO #6 contains > threshold number of ARINC 429 messages
5	AFTF5	R	0	This bit is set to "1" if FIFO #5 contains > threshold number of ARINC 429 messages
4	AFTF4	R	0	This bit is set to "1" if FIFO #4 contains > threshold number of ARINC 429 messages
3	AFTF3	R	0	This bit is set to "1" if FIFO #3 contains > threshold number of ARINC 429 messages
2	AFTF2	R	0	This bit is set to "1" if FIFO #2 contains > threshold number of ARINC 429 messages
1	AFTF1	R	0	This bit is set to "1" if FIFO #1 contains > threshold number of ARINC 429 messages
0	AFTF0	R	0	This bit is set to "1" if FIFO #0 contains > threshold number of ARINC 429 messages

FIFO FULL REGISTER
(Address 0x8029)

Bit	Name	R/W	Default	Description
7	AFFF7	R/W	0	This bit is set to "1" if FIFO #7 contains 32 ARINC 429 messages
6	AFFF6	R/W	0	This bit is set to "1" if FIFO #6 contains 32 ARINC 429 messages
5	AFFF5	R/W	0	This bit is set to "1" if FIFO #5 contains 32 ARINC 429 messages
4	AFFF4	R/W	0	This bit is set to "1" if FIFO #4 contains 32 ARINC 429 messages
3	AFFF3	R/W	0	This bit is set to "1" if FIFO #3 contains 32 ARINC 429 messages
2	AFFF2	R/W	0	This bit is set to "1" if FIFO #2 contains 32 ARINC 429 messages
1	AFFF1	R/W	0	This bit is set to "1" if FIFO #1 contains 32 ARINC 429 messages
0	AFFF0	R/W	0	This bit is set to "1" if FIFO #0 contains 32 ARINC 429 messages

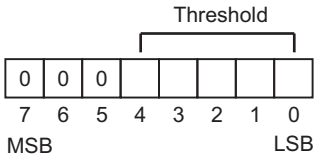
ARINC 429 Muxed FIFO Flags Register
(Address 0x800C)



Bit	Name	R/W	Default	Description
7	AMFF7	R	0	This bit gives the FIFO Flag Status for ARINC 429 Receive Channel 7. The FIFO Flag Status is set according to the value of the FFS[1:0] bits in the ARINC 429 Rx Control Register 7
6	AMFF6	R	0	This bit gives the FIFO Flag Status for ARINC 429 Receive Channel 6. The FIFO Flag Status is set according to the value of the FFS[1:0] bits in the ARINC 429 Rx Control Register 6
5	AMFF5	R	0	This bit gives the FIFO Flag Status for ARINC 429 Receive Channel 5. The FIFO Flag Status is set according to the value of the FFS[1:0] bits in the ARINC 429 Rx Control Register 5
4	AMFF4	R	0	This bit gives the FIFO Flag Status for ARINC 429 Receive Channel 4. The FIFO Flag Status is set according to the value of the FFS[1:0] bits in the ARINC 429 Rx Control Register 4
3	AMFF3	R	0	This bit gives the FIFO Flag Status for ARINC 429 Receive Channel 3. The FIFO Flag Status is set according to the value of the FFS[1:0] bits in the ARINC 429 Rx Control Register 3
2	AMFF2	R	0	This bit gives the FIFO Flag Status for ARINC 429 Receive Channel 2. The FIFO Flag Status is set according to the value of the FFS[1:0] bits in the ARINC 429 Rx Control Register 2
1	AMFF1	R	0	This bit gives the FIFO Flag Status for ARINC 429 Receive Channel 1. The FIFO Flag Status is set according to the value of the FFS[1:0] bits in the ARINC 429 Rx Control Register 1
0	AMFF0	R	0	This bit gives the FIFO Flag Status for ARINC 429 Receive Channel 0. The FIFO Flag Status is set according to the value of the FFS[1:0] bits in the ARINC 429 Rx Control Register 0

ARINC 429 FIFO Threshold Value Register

FIFO THRESHOLD VALUE
(Address 0x8021)



Default = 0x10

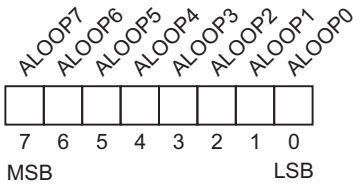
Threshold Value	Description
00000	Threshold flag is set if at least 1 message is in FIFO (Same as FIFO NOT EMPTY FLAG)
00001	Threshold flag is set if more than one message are in the FIFO
00010	Threshold flag is set if more than two messages are in the FIFO
00011	Threshold flag is set if more than three messages are in the FIFO
⋮	
10000	Threshold flag is set if more than sixteen messages are in the FIFO (default)
⋮	
11111	Threshold flag is set if 32 messages are in the FIFO (Same as FIFO FULL FLAG)

ARINC 429 Loop-back Self-Test

The HI-3200 includes an ARINC 429 loop-back feature, which allows users to exercise the ARINC 429 transmit and receive channels for self-test purposes. The ARINC 429 Loop-Back register, ALOOP defines which receiver channels are in loop-back mode. When a “1” is programmed in the ALOOP bit position for a receiver, then its ARINC 429 bus connection to the external pins is broken and instead the input is connected to one of the four ARINC 429 transmit channels. Transmit channel 0 is connected to receive channel 0 and 1, transmit channel 1 is connected to receive channels 2 and 3, and so on.

When in loop-back mode, incoming ARINC 429 messages are ignored by the HI-3200. When running in loop-back mode the ARINC 429 transmit pins may be disabled by pulling the TXMSK input high. This prevents test messages from being output to the external ARINC 429 transmit buses.

ARINC 429 LOOPBACK
(Address 0x8022)



Bit	Name	R/W	Default	Description
7	ALOOP7	R/W	0	This bit is set to "1" to loop-back transmit channel 3 to receiver 7
6	ALOOP6	R/W	0	This bit is set to "1" to loop-back transmit channel 3 to receiver 6
5	ALOOP5	R/W	0	This bit is set to "1" to loop-back transmit channel 2 to receiver 5
4	ALOOP4	R/W	0	This bit is set to "1" to loop-back transmit channel 2 to receiver 4
3	ALOOP3	R/W	0	This bit is set to "1" to loop-back transmit channel 1 to receiver 3
2	ALOOP2	R/W	0	This bit is set to "1" to loop-back transmit channel 1 to receiver 2
1	ALOOP1	R/W	0	This bit is set to "1" to loop-back transmit channel 0 to receiver 1
0	ALOOP0	R/W	0	This bit is set to "1" to loop-back transmit channel 0 to receiver 0

ARINC 429 Bit ordering

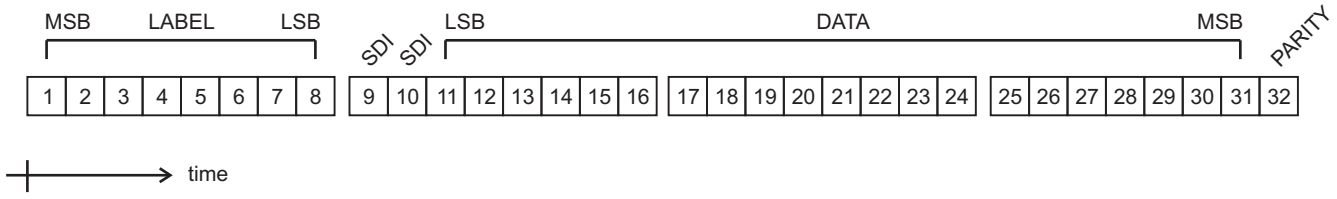
ARINC 429 messages consist of a 32-bit sequence as shown below. The first eight bits that appear on the ARINC 429 bus are the label byte. The next twenty three bits comprise a data field which presents data in a variety of formats defined in the ARINC 429 specification. The last bit transmitted is an odd parity bit.

The HI-3200 stores the received message as four bytes. The bytes are stored in memory in little-endian order. That is to say, the label byte (or status byte) is stored at the lowest memory address, the byte representing received bits 9 through 16 is stored at the next address, the byte representing bits 17 through 24 at the next address and the byte representing bits 25 though 32 at the highest address.

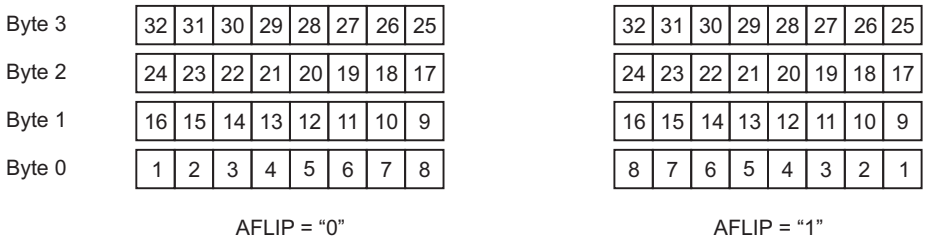
The ARINC 429 specifies the MSB of the label as ARINC bit 1. Conversely, the data field MSB is bit 31. So the bit significance of the label byte and data fields are opposite.

The HI-3200 may be programmed to "flip" the bit ordering of the label byte as soon as it is received and immediately prior to transmission. This is accomplished by setting the AFLIP bit to a "1" in the Master Control Register. Note that once the label byte has been flipped, the HI-3200 handles the flipped data byte "post-flip" for the purpose of label interrupt matching, filtering and storage.

ARINC 429 Message as received / transmitted on the ARINC 429 serial bus



ARINC 429 Message as stored in HI-3200 memory



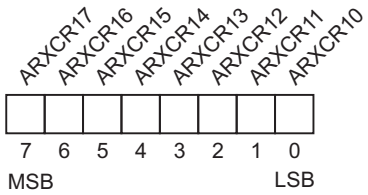
ARINC 429 Bit Monitor Pins

The HI-3200 has the capability of externally monitoring any ARINC 429 received payload bit through the pins ARXBIT[7:0] (**Note:** The HI-3200 provides external monitoring of two bits through pins ARXBIT1 and ARXBIT0, whereas the HI-3201 provides external monitoring of eight bits through pins ARXBIT7 to ARXBIT0). When the appropriate ARINC 429 receiver is enabled and the target label is received, the monitored bit value will be reflected on the pin. This allows the user to monitor any ARINC 429 received payload bit without performing any host SPI reads. The following registers configure the functionality of these monitor pins. Note that all these control register bits are reset to zero.

PINS ARXBIT[7:0] REGISTER (Address 0x805F)

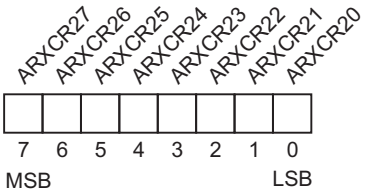
Bit	Name	R/W	Default	Description
7:0	ARXBIT[7:0]	R/W	0	These bits reflect the value of the corresponding pins ARXBIT[7:0]. After reset, all values are zero. When a monitored ARINC 429 bit changes, this register is updated with the value, which is reflected on the corresponding pin. The purpose of this register is to allow the user to preset the ARXBIT values after chip reset.

PIN ARXBIT0 CONFIGURATION REGISTER 1 (Address 0x8060)



Bit	Name	R/W	Default	Description
7:5	ARXCR1[7:5]	R/W	0	These bits select which receive channel (0 through 7) will have bits monitored and reflected on the pin ARXBIT0.
4:0	ARXCR1[4:0]	R/W	0	These bits select which bit (8 through 31) of the ARINC payload will be reflected on the pin ARXBIT0. The receiver is specified by bits ARX0CR1[7:5] and the target label is specified by pin ARXBIT0 Configuration Register 2 described below. Note that bits 0 through 7 of the ARINC payload are not monitored and selecting these bits results in no effect.

PIN ARXBIT0 CONFIGURATION REGISTER 2 (Address 0x8061)



Bit	Name	R/W	Default	Description
7:0	ARXCR2[7:0]	R/W	0	These bits select which label (0 through 255) will have bits monitored and reflected on the pin ARXBIT0. The receive channel and specific bits monitored are specified in ARXBIT0 Configuration Register 1 described above.

PINS ARXBIT1 Through ARXBIT7 CONFIGURATION REGISTERS (Addresses 0x8062 to 0x806F)

Each pin ARXBIT1 through ARXBIT7 are also specified by a pair of configuration registers similar to ARXBIT0 described above. Functionality is exactly the same. The register addresses for each pin specification are listed in the Register Map section (see page 11). Note that HI-3200 provides external monitoring of two bits through pins ARXBIT1 and ARXBIT0, whereas the HI-3201 provides external monitoring of eight bits through pins ARXBIT7 to ARXBIT0.

CAN BUS RECEIVE OPERATION

The HI-3200 can receive CAN frames from a single CAN bus using an external HI-3110 IC to handle the CAN bus protocol and physical layer connection

CAN Received Data Management

The HI-3200 interfaces to a CAN bus using an external HI-3110 CAN Controller / Transceiver IC. Communication between the HI-3200 and HI-3110 is handled by a dedicated high speed serial SPI link. Configuration parameters are automatically downloaded to the HI-3110 following a positive edge of the RUN input signal.

The HI-3110 handles all aspects of the CAN protocol as well as the physical layer interface to the CAN bus.

Received CAN frames are passed to the HI-3200 for filtering and storage. Each incoming frame's ID and first two data bytes is compared against a bank of up to 256 user-defined acceptance filters. If the frame meets the filter's acceptance criteria, it is stored in the CAN Received Data Memory. Each acceptance filter consists of a 6-byte match register and 6-byte mask. A frame is accepted if all unmasked bits match the corresponding bits in the frame. The frame is then written into the CAN received data memory location corresponding to the filter number.

Mask bits are defined as 1 = care and 0 = don't care. So for any given CAN frame bit to generate a filter pass condition, the received frame bit must equal the filter bit if the mask bit is "1", or any value if the mask bit is "0".

The number of filters active at any time is specified by writing a "1" into the corresponding bit position of the CAN Filter Enable look-up table. Only those filters identified as active in the table will be used to determine whether the frame is accepted.

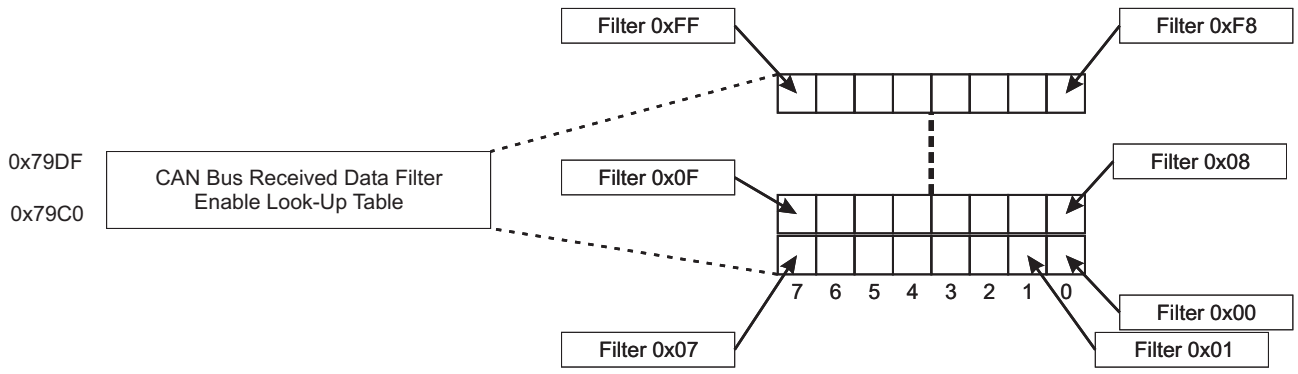
A CAN Receive Interrupt look-up table mirrors the active filter table. When a frame is accepted, and its corresponding Interrupt enable look-up bit is a one, then the CAN Interrupt bit is set in the Pending Interrupt Register. If the Interrupt Mask Register does not mask the CAN Interrupt, then the INT output signal will be asserted on frame reception and the CAN filter number will be loaded into the Interrupt Address Register.

It is possible that a received CAN frame may meet the acceptance criteria for more than one filter. In this case, only the first (lowest filter number) filter is used to qualify and store the incoming CAN frame.

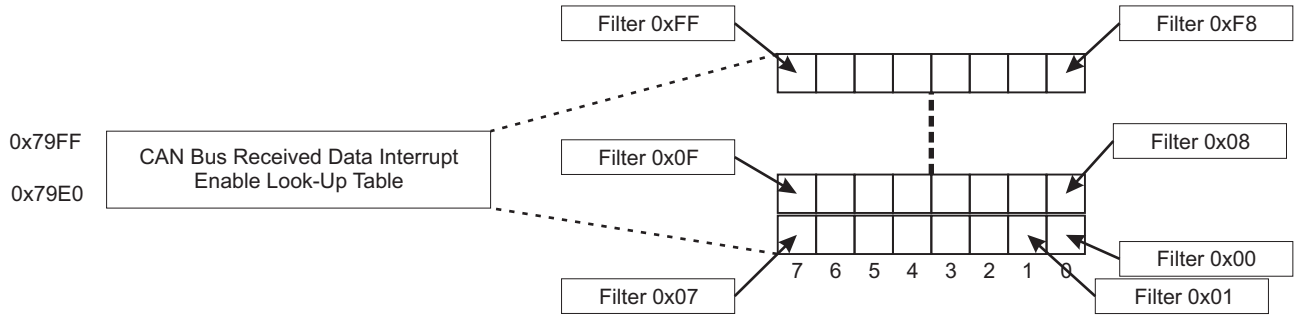
The CAN Filter Enable look-up table, CAN Interrupt Enable look-up table, Filter and Mask definition tables must be loaded prior to turning on the CAN receiver, either via the host CPU interface or from the auto-initialization EEPROM.

CAN frames are stored as sixteen-byte blocks as shown in the following diagram. Each block starts with a CAN frame Status Byte.

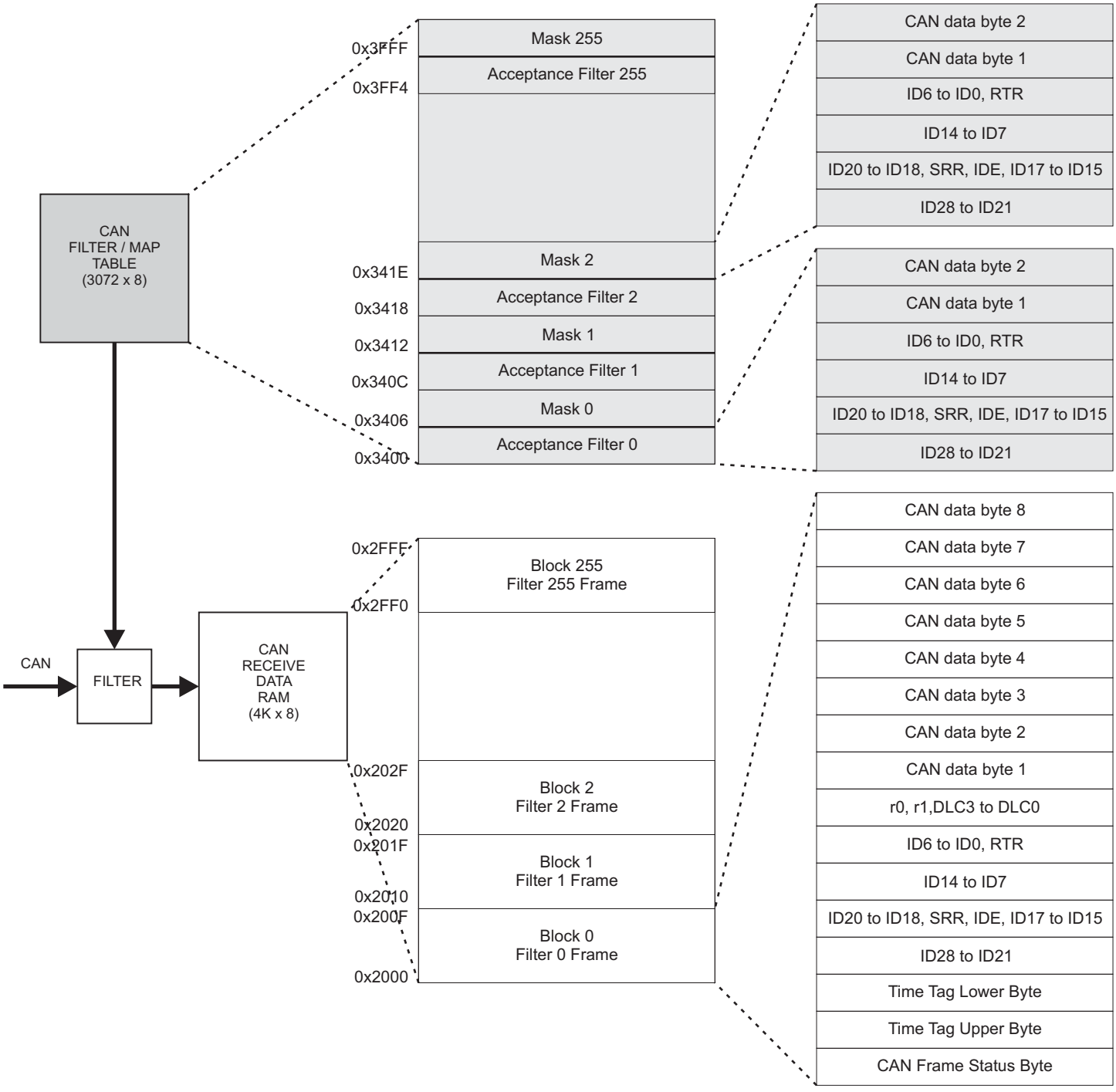
CAN Bus Received Data Filter Enable Look-Up Table



CAN Bus Received Data Interrupt Enable Look-Up Table

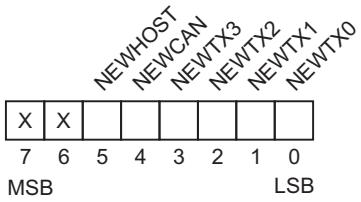


CAN Received Data Filter, Filter Mask and Data Storage Organization



CAN Bus Received Data Status Byte Definition

CAN RECEIVED DATA BLOCK STATUS BYTE



Bit	Name	R/W	Default	Description
7	-	R/W	0	Not used
6	-	R/W	0	Not used
5	NEWHOST	R/W	0	This bit is set when a new CAN frame is received and stored in this block. It is reset when the host CPU executes SPI instruction 0x9C to read the block.
4	NEWCAN	R/W	0	This bit is set when a new CAN frame is received and stored in this block. It is reset when the CAN Transmit scheduler reads any bytes from the block.
3	NEWTX3	R/W	0	This bit is set when a new CAN frame is received and stored in this block. It is reset when the ARINC 429 Transmit scheduler #3 reads any bytes from the block.
2	NEWTX2	R/W	0	This bit is set when a new CAN frame is received and stored in this block. It is reset when the ARINC 429 Transmit scheduler #2 reads any bytes from the block.
1	NEWTX1	R/W	0	This bit is set when a new CAN frame is received and stored in this block. It is reset when the ARINC 429 Transmit scheduler #1 reads any bytes from the block.
0	NEWTX0	R/W	0	This bit is set when a new CAN frame is received and stored in this block. It is reset when the ARINC 429 Transmit scheduler #0 reads any bytes from the block.

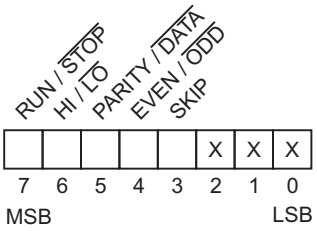
ARINC 429 TRANSMIT OPERATION

The HI-3200 has four on-board ARINC 429 transmit channels which directly drive ARINC 429 differential line drivers such as the Holt HI-8570. ARINC 429 words may be written to the transmitters either directly, using an SPI instruction, or be generated automatically using the four ARINC 429 message schedulers.

ARINC 429 Transmit Channel Configuration

Each of the four available ARINC 429 Transmit channels is configured using its own register. Register address 0x8018 controls ARINC 429 Transmit channel #0, register address 0x8019 controls channel #1 and so on. The ATXCn registers may be written or read at any time.

ARINC 429 TX CONTROL REGISTER 0 - 3
(Address 0x8018 - 0x801B)



Bit	Name	R/W	Default	Description
7	RUN / $\overline{\text{STOP}}$	R/W	0	When zero, transmission from this ARINC 429 transmit channel is suspended after the currently transmitting label is sent. When this bit is taken high, transmission starts at the beginning of the descriptor table for this channel.
6	$\overline{\text{HI}}$ / LO	R/W	0	Selects the transmission rate for the ARINC 429 transmit channel. A “0” selects high-speed (100Kb/s) and a “1” selects low-speed (12.5Kb/s).
5	PARITY / $\overline{\text{DATA}}$	R/W	0	When this bit is a one, the 32nd transmitted ARINC bit is overwritten with a parity flag. When this bit is a zero, all 32-bits are transmitted as data.
4	EVEN / $\overline{\text{ODD}}$	R/W	0	When PARITY / DATA is a “1”, this bit defines whether th 32nd transmitted bit is set for Even or Odd Parity. A “1” selects even parity and a “0” selects odd parity.
3	SKIP	R/W	0	When set a “1’ instructs the transmit sequencer to wait for the next Repetition Rate Counter rollover before beginning a new transmission cycle. A “0” causes an immediate restart of the cycle following completion of the prior cycle.
2	-	R/W	0	Not Used
1	-	R/W	0	Not Used
0	-	R/W	0	Not Used

ARINC 429 Transmit Scheduler

Each of the four ARINC 429 transmit channels has its own transmit controller. The controller is user-programmed to output ARINC labels in a predefined order and repetition rate. A sequence of up to 256 ARINC labels may be transmitted before repeating the sequence.

A descriptor table with up to 256 entries (descriptors) is compiled by the user to define the sequence of ARINC 429 messages transmitted on each channel. When the RUN/$\overline{\text{STOP}}$ bit in the ARINC TX Control Register is asserted, the controller compiles the first 32-bit ARINC word from the instructions given by the first descriptor and then transmits it. A Transmit Sequence Pointer then increments to the next descriptor in the table and the process is repeated for Descriptor number 2.

ARINC 429 messages continue to be compiled and transmitted until the last descriptor in the table. The end of the table is marked by a special descriptor if not all 256 entries are needed. The Sequence Pointer is then reset to zero.

A Repetition Rate Counter is used to time the start of the next transmission cycle.

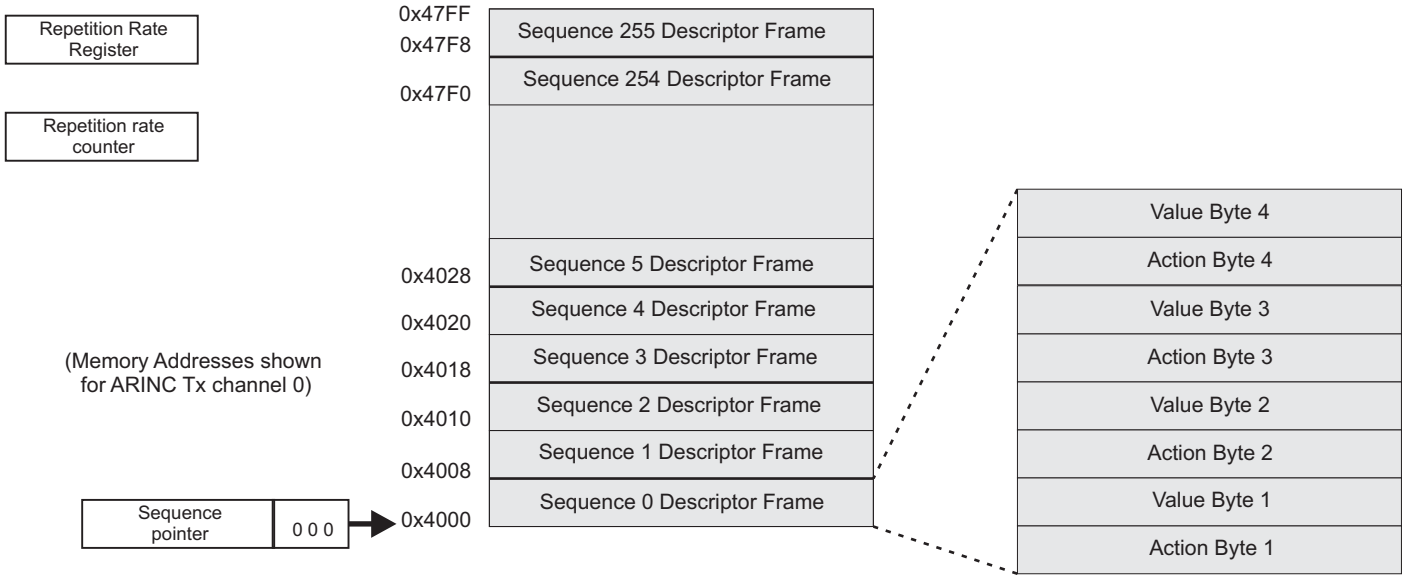
The user is responsible for construction of the descriptor table and for setting the Repetition Rate prior to asserting RUN/STOP. Facilities exist for immediate cycle repetition and for single-cycle operation.

The byte content of each ARINC 429 message transmitted is user defined by the descriptor contents. Data bytes may be sourced from the host CPU / auto-initialization EEPROM (immediate data) or from the ARINC 429 receive memory (ARINC indexed) or CAN bus receiver memory (CAN indexed). This allows received ARINC data to be re-transmitted on another bus with or without filtering, label byte re-assignment or data modification. It allows data received on the CAN bus to be re-formatted and re-labelled for automatic ARINC 429 transmission. It also allows data from multiple ARINC 429 receive buses and the CAN bus to be re-packetized into new ARINC 429 transmitted messages.

Conditional transmission control allows sequenced words to be skipped if no new data is available.

Each ARINC 429 transmit channel is independently configured with its own ARINC 429 TX Control Register, ATXCR0-3, as previously described.

ARINC 429 Transmit Descriptor table

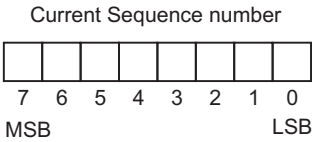


The value of each ARINC 429 label transmitted in the sequence is defined by its eight-byte descriptor. The descriptor consists of one “Action byte” and one “Value” byte for each of the four bytes that make up the ARINC 429 transmitted label.

The four pairs of Action and Value bytes describe where the data for each byte may be found. Different op-codes allow the data source to be host CPU populated fixed values, or values from specific locations within the ARINC 429 receive memory or CAN bus receive memory. Action byte 1 also has one additional op-code to facilitate sequence flow control.

The construction of Action and Value bytes are described in the next section.

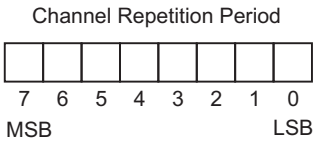
TRANSMIT SEQUENCE POINTERS 0 -3
(Address 0x802C - 0x802F)



The transmit sequence pointer is set to zero on Master Reset. Once the Control Register RUN / $\overline{\text{STOP}}$ bit goes high, sequence execution begins at sequence count zero. After the first word is sent, the pointer is incremented by one descriptor (counts descriptor frames).

This continues until the programmed sequence is complete. The sequence pointer is then reset to the beginning of the descriptor table and program execution begins as soon the channel repetition rate counter time elapses.

REPETITION RATE REGISTER
(Address 0x801C - 0x801F)



The Repetition rate register value defines the time interval between successive starts of the programmed transmit sequence for each ARINC 429 transmit channel. The value is set in binary, with the LSB representing 10 ms. Repetition rate time periods may therefore be set from 0 ms to 2.55 seconds

If the repetition rate is shorter than the minimum time needed to transmit all ARINC 429 words in the sequence (but not zero), the transmit sequence will begin again immediately if the Control Register SKIP bit is a zero. If the SKIP bit is a one, the sequencer will wait until the next rollover of the Repetition Rate Counter before starting a new cycle.

One-time Sequence Transmission

When the Repetition Rate Register contains zero (default), sequence transmission occurs just once, upon 0-to-1 transition of the RUN/STOP bit in the Transmit Control Register. One-time execution of the sequencer is useful when transmitting ARINC 429 words directly from the host CPU. One or more immediate-mode descriptors can be written into the sequence table and transmitted, then transmit values can be refreshed for the next cycle.

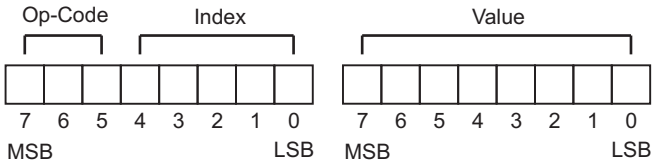
There are three ways to retrigger one-time transmission after its previous completion:

Method 1: Reset the RUN/STOP bit in the Transmit Control Register to zero. Sequentially write 0xFFFF then 0x0000 into the Repetition Rate Register. Initiate one-time transmission by setting the RUN/STOP bit in the Transmit Control Register; retransmission occurs immediately. This method can be used without affecting the other three transmitters.

Method 2: With the RUN/STOP bit in the Transmit Control Register already set to logic-1 by the previous one-time sequence transmission, toggle the state of the RUN input pin low and then high; retransmission occurs immediately.

Method 3: With the RUN/STOP bit in the Transmit Control Register already set to logic-1 by the previous one-time sequence transmission, toggle the state of A429TX bit 6 in the Master Control Register low and then high; retransmission occurs immediately.

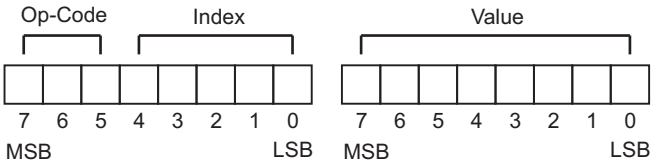
ARINC 429 BYTE 1 DESCRIPTOR



ACTION BYTE **VALUE BYTE**

Op-Code	Index	Value	Description
000	XXXXX	XXXXXXXX	End of sequence. When op-code 000 is encountered by the sequencer before it reaches sequence number 255, the sequencer resets to zero and begins the next transmission cycle starting at descriptor number 0 as soon as the repitition rate counter rolls over. Note that the descriptor table is cleared following Master Reset, so no ARINC 429 transmissions are possible until the sequence table has been configured.
001	XXXXX	XXXXXXXX	No-operation. This descriptor is ignored and the sequencer increments to the next descriptor in the sequence. This opcode may be used to temporarily suspend transmission of a particular message in the sequence, without having to modify the remaining bit fields of the descriptor or reloading the entire descriptor block in order to delete the entry.
010	XXXXX	LLLLLLLL	Immediate data. The value contained in the descriptor value data byte is loaded into byte 1 (the ARINC 429 “label” byte) of the ARINC 429 label to be transmitted.
011	CCCXX	LLLLLLLL	Immediate data conditional. The NEWTXn bit corresponding to the ARINC Data RAM location defined by channel “CCC” and label block “LLLLLLLL” is read. LLLLLLLL is used as Byte 1 if NEWTXn is set for this **or any other conditional opcode** within this descriptor frame. If NEWTXn = 0 for all conditional op codes (within this descriptor frame) then no transmission occurs for this frame and the sequencer increments to the next descriptor frame. The NEWTXn bit for the referenced ARINC RAM block is reset.
100	CCCBB	LLLLLLLL	Indexed data. The value of ARINC Data RAM location defined by channel “CCC”, label block “LLLLLLLL” and byte number “BB” is loaded into byte 1 of the ARINC 429 label to be transmitted.
101	CCCBB	LLLLLLLL	Indexed data conditional. The NEWTXn bit corresponding to the ARINC Data RAM location defined by channel “CCC”, label block “LLLLLLLL” and byte number “BB” is read. The corresponding byte is used as Byte 1 if NEWTXn is set for this **or any other conditional opcode** within this descriptor frame. If NEWTXn = 0 for all conditional op codes (within this descriptor frame) then no transmission occurs for this frame and the sequencer increments to the next descriptor frame. The NEWTXn bit for the referenced ARINC RAM block is reset.
110	XBBBB	NNNNNNNN	CAN byte. The value of the byte at CAN RAM address filter block “NNNNNNNN”, byte number “BBBB” is loaded into the ARINC buffer byte 1 position to be transmitted.
111	XBBBB	NNNNNNNN	CAN byte conditional. The NEWTXn bit corresponding to the CAN RAM address filter block “NNNNNNNN”, byte number “BBBB” is read. The corresponding byte is used as Byte 1 if NEWTXn is set for this **or any other conditional opcode** within this descriptor frame. If NEWTXn = 0 for all conditional op codes (within this descriptor frame) then ARINC word construction is terminated, no transmission occurs for this frame and the sequencer increments to the next descriptor frame. The NEWTXn bit for the referenced CAN RAM address filter block is reset.

ARINC 429 BYTES 2 - 4 DESCRIPTOR



ACTION BYTE **VALUE BYTE**

Op-Code	Index	Value	Description
000	XXXXX	XXXXXXXX	No-Op op-code. ARINC 429 word construction will be terminated and the sequencer will move on to the next descriptor in the table.
001	XXXXX	XXXXXXXX	No-Op op-code. ARINC 429 word construction will be terminated and the sequencer will move on to the next descriptor in the table.
010	XXXXX	LLLLLLLL	Immediate data. The value contained in the descriptor value data byte is loaded into this byte position of the ARINC 429 32-bit message to be transmitted.
011	CCCXX	LLLLLLLL	Immediate data conditional. The NEWTXn bit corresponding to the ARINC Data RAM location defined by channel “CCC” and label block “LLLLLLLL” is read. LLLLLLLL is used if NEWTXn is set for this **or any other conditional opcode** within this descriptor frame. If NEWTXn = 0 for all conditional op codes (within this descriptor frame) then no transmission occurs for this frame and the sequencer increments to the next descriptor frame. The NEWTXn bit for the referenced ARINC RAM block is reset.
100	CCCBB	LLLLLLLL	Indexed data. The value of ARINC Data RAM location defined by channel “CCC”, label block “LLLLLLLL” and byte number “BB” is loaded into this byte position of the ARINC 429 label to be transmitted.
101	CCCBB	LLLLLLLL	Indexed data conditional. The NEWTXn bit corresponding to the ARINC Data RAM location defined by channel “CCC”, label block “LLLLLLLL” and byte number “BB” is read. The corresponding byte is used if NEWTXn is set for this **or any other conditional opcode** within this descriptor frame. If NEWTXn = 0 for all conditional op codes (within this descriptor frame) then no transmission occurs for this frame and the sequencer increments to the next descriptor frame. The NEWTXn bit for the referenced ARINC RAM block is reset.
110	XBBBB	NNNNNNNN	CAN byte. The value of the byte at CAN RAM address filter block “NNNNNNNN”, byte number “BBBB” is loaded into the ARINC buffer byte position to be transmitted.
111	XBBBB	NNNNNNNN	CAN byte conditional. The NEWTXn bit corresponding to the CAN RAM address filter block “NNNNNNNN”, byte number “BBBB” is read. The corresponding byte is used if NEWTXn is set for this **or any other conditional opcode** within this descriptor frame. If NEWTXn = 0 for all conditional op codes (within this descriptor frame) then ARINC word construction is terminated, no transmission occurs for this frame and the sequencer increments to the next descriptor frame. The NEWTXn bit for the referenced CAN RAM address filter block is reset.

ARINC 429 Immediate Transmit Option

The Host CPU may instruct the HI-3200 to transmit an ARINC 429 message immediately using a special SPI command. The SPI command selects the transmit channel and provides the four bytes of data to be sent as a 32-bit ARINC 429 message.

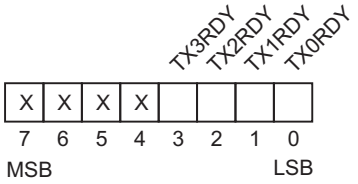
If the transmit channel's sequencer is not running (ATCR bit RUN/$\overline{\text{STOP}}$ = "0"), or the sequencer is waiting for the repetition rate counter to rollover, then the new ARINC 429 message is transmitted without delay.

If the transmit sequencer for the selected channel is active, then the new message is transmitted as soon as the current message has been sent. The sequencer then resumes operation at the next location in the queue.

Both the RUN input and the Master Control Register A429TX bit must be high to enable any ARINC 429 transmission.

Table 1 lists the host CPU SPI instruction format.

ARINC 429 TX READY BITS REGISTER (Address 0x800D)



Bit	Name	R/W	Default	Description
7	-	R	0	Not Used
6	-	R	0	Not Used
5	-	R	0	Not Used
4	-	R	0	Not Used
3	TX3RDY	R	1	This bit is set when Transmit Buffer 3 is empty and ready to receive the next 32-bit word for transmission. The bit is reset when Transmit Buffer 3 is not empty.
2	TX2RDY	R	1	This bit is set when Transmit Buffer 2 is empty and ready to receive the next 32-bit word for transmission. The bit is reset when Transmit Buffer 2 is not empty.
1	TX1RDY	R	1	This bit is set when Transmit Buffer 1 is empty and ready to receive the next 32-bit word for transmission. The bit is reset when Transmit Buffer 1 is not empty.
0	TX0RDY	R	1	This bit is set when Transmit Buffer 0 is empty and ready to receive the next 32-bit word for transmission. The bit is reset when Transmit Buffer 0 is not empty.

NOTE: Immediate Transmit is double-buffered so TXxRDY will remain high after a single message is loaded. If another message is loaded prior to completion of the first message, RXxRDY will go low, indicating Not Ready status.

CAN BUS TRANSMIT OPERATION

The HI-3200 is able to transmit CAN frames via an external HI-3110 CAN controller / transceiver IC. CAN frames may be loaded for immediate transmission from the host CPU, or in a pre-programmed sequence using the integrated CAN frame scheduler.

CAN BUS Transmit Scheduler

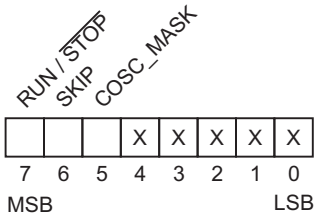
CAN frames to be transmitted are constructed and launched from the CAN Bus transmit scheduler. The scheduler is user programmed using a descriptor table to output CAN frames in a predefined order and repetition rate. To make best use of available memory space, three different types (Type 1-3) of descriptor tables entry formats are available. The user may mix descriptor types in the table.

The CAN Indentifier ID and data byte content of each frame transmitted is user defined and may be sourced from the host CPU / auto-initialization EEPROM (immediate data) or bytes from the ARINC 429 receive memory (ARINC indexed) or CAN bus receiver memory (CAN indexed). This allows received ARINC and / or CAN bus data to be re-formated and re-transmited on the CAN bus. Conditional transmission control allows sequenced words to be skipped if no new data is available. CAN frames are constructed and transmitted until an end of sequence marker or RAM location 0x79BF is reached. Note that if 0x79BF is reached before the frame is completely constructed, that frame will be discarded.

Note that because the CAN bus bandwidth is shared between all terminals on the bus, sufficient bandwidth to transmit the entire programmed sequence of frames may not be available in the time slot programmed. In such circumstances the user may choose to repeat the sequence immediately upon completion, or wait until the next multiple of the programmed repetition rate elapses.

The CAN sequencer operation is controlled by the CAN Transmit control Register:

CAN TRANSMIT CONTROL REGISTER (Address 0x8032)

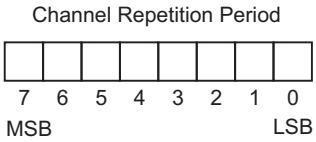


Bit	Name	R/W	Default	Description
7	RUN / $\overline{\text{STOP}}$	R/W	0	When zero, transmission from the CAN Bus transmit channel is suspended after the currently transmitting frame is sent. When this bit is taken high, transmission starts at the beginning of the descriptor table.
6	SKIP	R/W	0	When set to “1” instructs the transmit sequencer to wait for the next Repetition Rate Counter rollover before beginning a new transmission cycle. A “0” causes an immediate restart of the cycle following completion of the prior cycle.
5	COSC_MASK	R/W	0	When set to “1” this bit masks off the COSC pin.
4	-	R/W	0	Not Used
3	-	R/W	0	Not Used
2	-	R/W	0	Not Used
1	-	R/W	0	Not Used
0	-	R/W	0	Not Used

CAN TRANSMIT SEQUENCE POINTER (Address 0x803E/F)

The transmit sequence pointer is set to 0x6000 on Master Reset. Once the RUN / $\overline{\text{STOP}}$ bit goes high, sequence execution begins at sequence count zero (Memory Address 0x6000). After the first word is sent, the pointer is incremented to the address of the next descriptor in the sequence table. This continues until the programmed sequence is complete. The sequence pointer is then reset to zero and program execution begins as soon the CAN repetition rate counter time elapses.

CAN REPETITION RATE REGISTER (Address 0x8033)

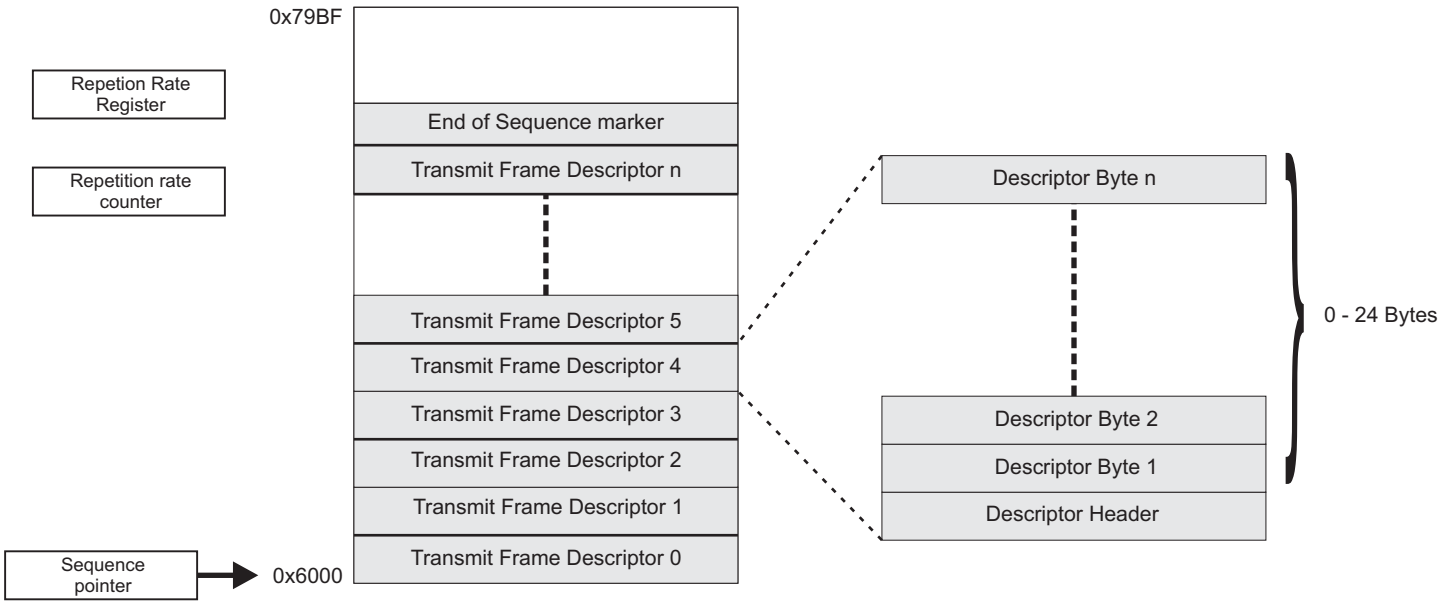


The Repetition rate register value defines the time interval between successive starts of the programmed CAN transmit sequence. The value is set in binary, with the LSB representing 10 ms.

If the repetition rate is shorter than the minimum time needed to transmit all CAN frames in the sequence (but not zero), the transmit sequence will begin again immediately if the CAN Control Register SKIP bit is a zero. If the SKIP bit is a one, the sequencer will wait until the next rollover of the Repetition Rate Counter before starting a new cycle.

When the Repetition Rate counter is programmed to zero (default), the transmit sequence shall execute one time only. A zero - to - one transition of the RUN/STOP bit will cause the transmit sequence to start. One-time execution of the sequencer is useful when transmitting CAN frames directly from the host CPU. One or more immediate-mode (Type 1) descriptors can be written into the sequence table, transmitted, and then refreshed for the next cycle.

CAN Bus Transmit Descriptor Table

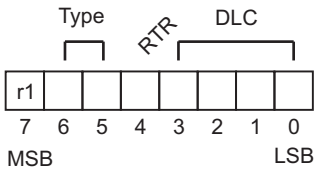


The contents of each CAN frame transmitted in the sequence is defined by its Frame Descriptor. Each descriptor consists of one “Descriptor Header Byte” and from 0 to 24 additional descriptor bytes depending upon, the descriptor type (type 1 - 3) and the data payload length. A special descriptor header marks the end of the descriptor table.

The three different descriptor types allow the user to make best use of the available descriptor table memory space while still allowing complete flexibility in defining frame content and data source. The Descriptor header byte is of a common format for all three descriptor types.

The CAN Transmit sequence pointer uses information in the header byte to determine the length of the descriptor and thus the address of the next descriptor block in the sequence table.

CAN TRANSMIT DESCRIPTOR HEADER BYTE



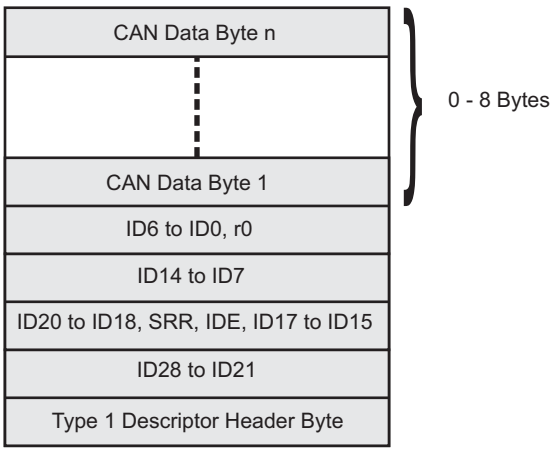
Type	RTR	DLC	Description
00	X	XXXX	End of sequence marker.
01	0	NNNN	Type 1 descriptor CAN Data frame. Data Length Code (DLC) indicates the number of data bytes in the CAN frame 0 - 8. DLC values >8 always have eight data bytes. The number of descriptor bytes D is given by D=1+(4+N),where N is the DLC value or 8 if DLC>8.
01	1	XXXX	Type 1 descriptor CAN Remote frame. The number of bytes in the descriptor is 5.
10	0	NNNN	Type 2 descriptor CAN Data frame. The number of descriptor bytes is D=2+(2 x N), where N is the DLC value or 8 if DLC>8.
10	1	XXXX	Type 2 descriptor CAN Remote Frame. The number of descriptor bytes is 2.
11	0	NNNN	Type 3 descriptor CAN Data frame. The number of descriptor bytes is D=9+(2 x N), where N is the DLC value or 8 if DLC>8.
11	1	XXXX	Type 3 descriptor CAN Remote frame. The number of descriptor bytes is 9.

NOTE: Bit 7 should be set to the value of the reserved bit “r1” of the CAN frame.

Type 1 CAN Transmit Descriptor Frame Format

Type 1 CAN transmit descriptors are used when transmitting frames using CAN Identifier and Data payload values defined explicitly by the HI-3200 Host CPU or Auto-initialization EEPROM.

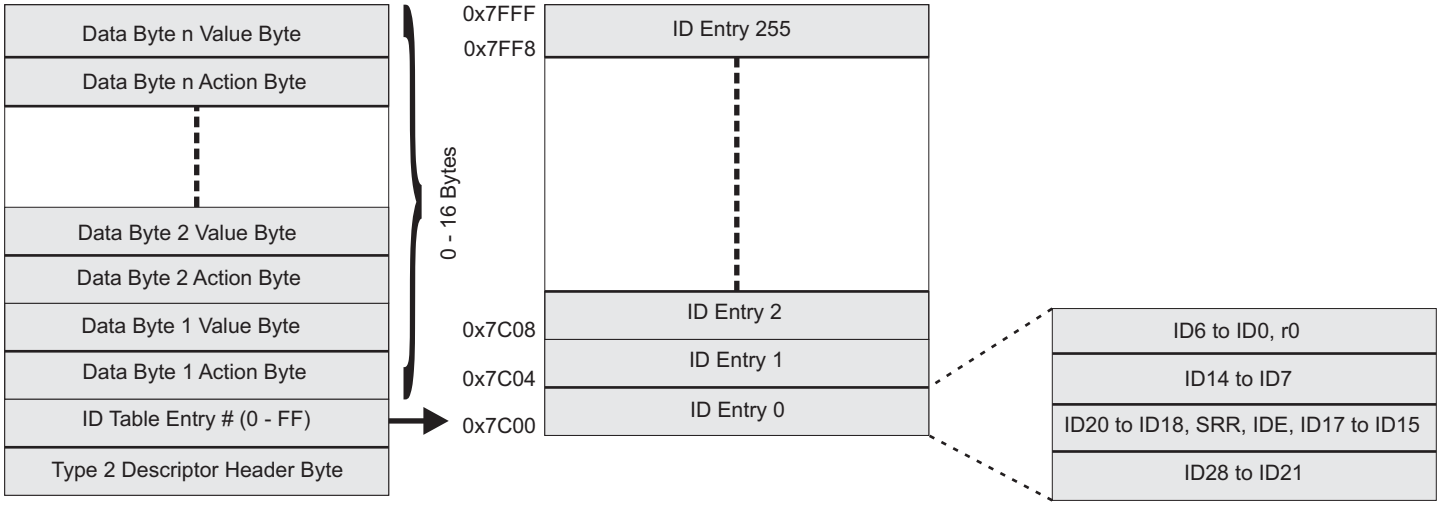
The descriptor format is as follows:



Type 2 CAN Transmit Descriptor Frame Format

Type 2 CAN transmit descriptors use a pre-loaded 256 entry common look-up table to define the CAN ID field of the transmitted frame.

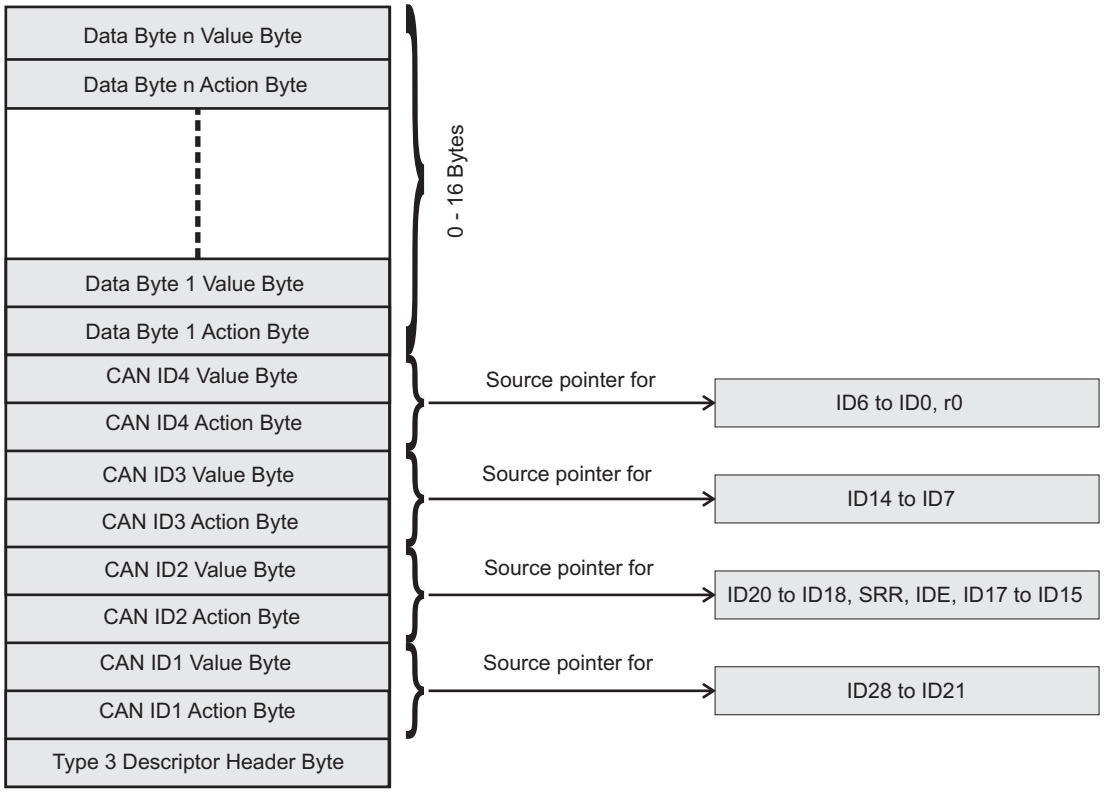
CAN Data Byte values may be directly loaded from the host CPU / Auto-initialization EEPROM, or are read from the ARINC 429 Received Data RAM or CAN Bus Received Data RAM as indexed by the two data source descriptor bytes (op-code byte and index byte).



Type 2 Descriptor Block **Type 2 ID Look-up Table** **Type 2 ID Table Entry**

Type 3 CAN Transmit Descriptor Frame Format

Type 3 CAN transmit descriptors are used when transmitting frames using CAN Identifier and Data payload extracted from the ARINC 429 Received Data RAM, CAN Bus Received Data RAM, or host CPU / Auto-initialization EEPROM loaded values.

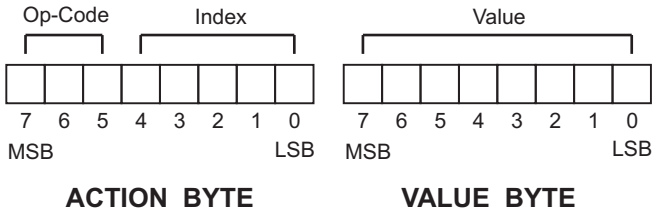


Type 3 Descriptor Block

Type 2 and 3 CAN Transmit Descriptor Opcode and Index Bytes

Type 2 and 3 CAN Transmit Discriptor Op-Code and Index byte pairs specify the source of data for their corresponding CAN frame ID bits and data bytes. Their format and operation is similar to the ARINC 429 descriptor byte pairs:

CAN TRANSMIT OPCODE - INDEX BYTE PAIR



Op-Code	Index	Value	Description
0X0	XXXXX	LLLLLLLL	Immediate data. The value contained in the descriptor value data byte is loaded into this byte position of the CAN Frame to be transmitted.
0X1	CCCXX	LLLLLLLL	Immediate data conditional. The NEWCAN bit corresponding to the ARINC Data RAM location defined by channel "CCC" and label block "LLLLLLLL" is read. LLLLLLLL is used if NEWCAN is set for this **or any other conditional opcode** within this descriptor frame. If NEWCAN = 0 for all conditional op codes (within this descriptor frame) then no transmission occurs for this frame and the sequencer increments to the next descriptor frame. The NEWCAN bit for the referenced ARINC RAM block is reset.
100	CCCBB	LLLLLLLL	Indexed data. The value of ARINC Data RAM location defined by channel "CCC", label block "LLLLLLLL" and byte number "BB" is loaded into this byte position of the CAN frame to be transmitted.
101	CCCBB	LLLLLLLL	Indexed data conditional. The NEWCAN bit corresponding to the ARINC Data RAM location defined by channel "CCC", label block "LLLLLLLL" and byte number "BB" is read. The corresponding byte is used if NEWCAN is set for this **or any other conditional opcode** within this descriptor frame. If NEWCAN = 0 for all conditional op codes (within this descriptor frame) then no transmission occurs for this frame and the sequencer increments to the next descriptor frame. The NEWCAN bit for the referenced ARINC RAM block is reset.
110	XBBBB	NNNNNNNN	CAN byte. The value of the byte at CAN RAM address filter block "NNNNNNNN", byte number "BBBB" is loaded into the CAN transmit buffer byte position to be transmitted.
111	XBBBB	NNNNNNNN	CAN byte conditional. The NEWCAN bit corresponding to the CAN RAM address filter block "NNNNNNNN", byte number "BBBB" is read. The corresponding byte is used if NEWCAN is set for this **or any other conditional opcode** within this descriptor frame. If NEWCAN = 0 for all conditional op codes (within this descriptor frame) then ARINC word construction is terminated, no transmission occurs for this frame and the sequencer increments to the next descriptor frame. The NEWCAN bit for the referenced CAN RAM address filter block is reset.

CAN Bus Immediate Transmit Option

The Host CPU may instruct the HI-3200 to transmit an CAN frame immediately using a special SPI command. The SPI command describes the CAN frame with 5 to 13 SPI data bytes. The bit format of the data bytes is exactly the same as a Type 1 transmit descriptor, except the Type Field of the header byte is “don’t care”.

If the CAN transmit sequencer is not running (CTCR bit RUN/$\overline{\text{STOP}}$ = “0”), or the sequencer is waiting for the repetition rate counter to rollover, then the new CAN frame is transmitted without delay.

If the CAN transmit sequencer is active, then the new frame is transmitted to HI-3110 as soon as the current frame has been sent. The sequencer then resumes operation at the next location in the queue.

Both the RUN input pin and the Master Control Register CANTX bit must be high to enable any CAN frame transmission.

Table 1 lists the host CPU SPI instruction format.

RESET AND START-UP OPERATION

After power-on, the HI-3200 is in an undefined state. The MRST pin must be taken high to begin device initialization. The MRST pin may be asserted at any time. Taking MRST high immediately stops all execution and sets the READY output low indicating that the part is in the reset state.

On the falling edge of MRST the HI-3200 samples the state of the MODE2-0 input pins. This is the only occasion these inputs are sampled. The state of the MODE pins determines one of eight possible initialization sequences (Mode 0 through Mode 7). MODE[2:0] = 000 sets Mode 0, MODE[2:0] = 001 sets Mode 1, MODE[2:0] = 111 sets Mode 7, etc. See Figure 1, Reset and Start-up Operation Flow Chart. These eight initialization modes allow the user to customize the start-up configuration of the device.

The total time from MRST falling edge to READY pin high depends on the total number of MCLK cycles (48MHz clock input) and is summarized in Table 2 for each mode.

For Modes 2 and 3, the RBFFAIL bit-7 should be cleared to 0 by writing to the BIST Control/Status register (Address 0x8070) BEFORE taking MRST high.

To use BIST self-tests, the part needs to be in Mode 4 with the RUN pin low.

Once the initialization is complete, the device enters the Idle State when the ready pin goes high. In Idle State, the host CPU may communicate with the HI-3200 memory and registers using the host CPU SPI link. Note that when Mode 6 is selected, the host CPU SPI by-passes the HI-3200 and instead communicates directly with the HI-3110 CAN controller, if used. When in the Idle State, The HI-3200 does not transmit or receive any messages on either the ARINC 429 buses or the CAN bus.

To begin data bus operation, the user must transition the RUN input from a low to high state. Immediately following the rising edge of RUN, the HI-3200 configures the HI-3110 CAN controller, if used, according to the MODE selection. The part then enters the Active State and bus message processing begins.

During initialization, various device configuration tasks are performed according to the Mode selection set at the MODE2:0 input pins. The available options are described below in sub-sections 1-6.

TABLE 2. Start-up Time for each Mode

MODE[2:0] PINS	MODE	MCLK Cycles	Start-up Time*
000	0	8,203	1.71µs
001	1	948,390	19.76ms
010	2	73,744	1.54ms
011	3	1,013,931	21.0ms
100	4	9	188.0ns
101	5	3,339	69.6µs
110	6	3,339	69.6µs
111	7	3,339	69.6µs

* The time from falling edge of MRST to READY pin high.

1. RAM Integrity Check

In Modes 2 and 3, the HI-3200 performs a RAM integrity check. A read/write check is performed on the entire RAM space. An incrementing pattern is written to sequential RAM locations then this pattern is read and verified. Each RAM location is re-written with the 1s complement of its current contents then this pattern is read and verified. The incrementing pattern followed by its 1s complement ensures that each RAM location can store both a 1 and 0 state. If the RAM integrity check fails, the MINT pin is asserted and the Pending Interrupt Register RAMFAIL bit is set. The part enters the “Safe” state, in which the HI-3200 is able to accept and respond to Host CPU SPI Instructions, but cannot enter Normal Operating mode until the MRST input is taken high to repeat the initialization sequence. The RAMFAIL Interrupt is not maskable.

2. Clear Data Memory

In Modes 0, 1, 2, 3, 5, 6, and 7, the HI-3200 automatically clears all memory locations in the address range 0x0000 to 0x33FF. This is the space reserved for ARINC 429 and CAN message data. Configuration tables and HI-3200 registers are not affected.

3. Initialize Registers and Clear all memory

In addition to clearing data memory (0x0000 to 0x33FF), Modes 0, 1, 2, and 3 also clear all configuration and look-up tables (0x3400 to 0x7FFF) as well as setting all registers (0x8000 to 0x807F) to their default states. All registers default to zero unless otherwise noted.

4. Auto-Initialize from EEPROM

The contents of the Auto-Initialization EEPROM are copied into the HI-3200 memory and registers via the EEPROM SPI interface. The part verifies the integrity of the data transfer from the EEPROM by running through a byte-by-byte compare routine and a checksum validation. If a compare error is detected, the AUTOERR bit is set in the Pending Interrupt Register, the MINT output is asserted, the location of the error is captured in the AUTO-INIT FAIL ADDRESS registers 0x8073 (Auto-Init Fail LS address) and 0x8074 (Auto-Init Fail MS address) and the part enters the Safe state. If a checksum error is detected, the CHKERR bit is set in the Pending Interrupt Register, the MINT output is asserted and the part enters the Safe state. The AUTOERR and the CHKERR interrupts are not maskable.

Once initialization is complete, the part enters the Idle state. The host CPU may read and write HI-3200 internal memory and registers in Modes 0, 1, 2, 3, 4, 5, and 7. If not using the auto-initizarion feature, the host CPU should configure the device at this time.

5 Enable SPI By-Pass

In Mode 6, the host CPU SPI interface completely by-passes the HI-3200, and all communication is directed to the HI-3110 SPI bus such that the user may directly access registers within the HI-3110. This “by-pass’ mode is intended as an aid to debugging only and is not recommended in the final system design implementation.

By-pass mode is exited at the first rising edge of the RUN pin. Further toggling of the RUN pin will not re-engage the by-pass mode. Since the host does not have access to internal HI-3200 registers prior to RUN going high, the user must first initialize these registers prior to entering Mode 6.

6. HI-3110 Initialize

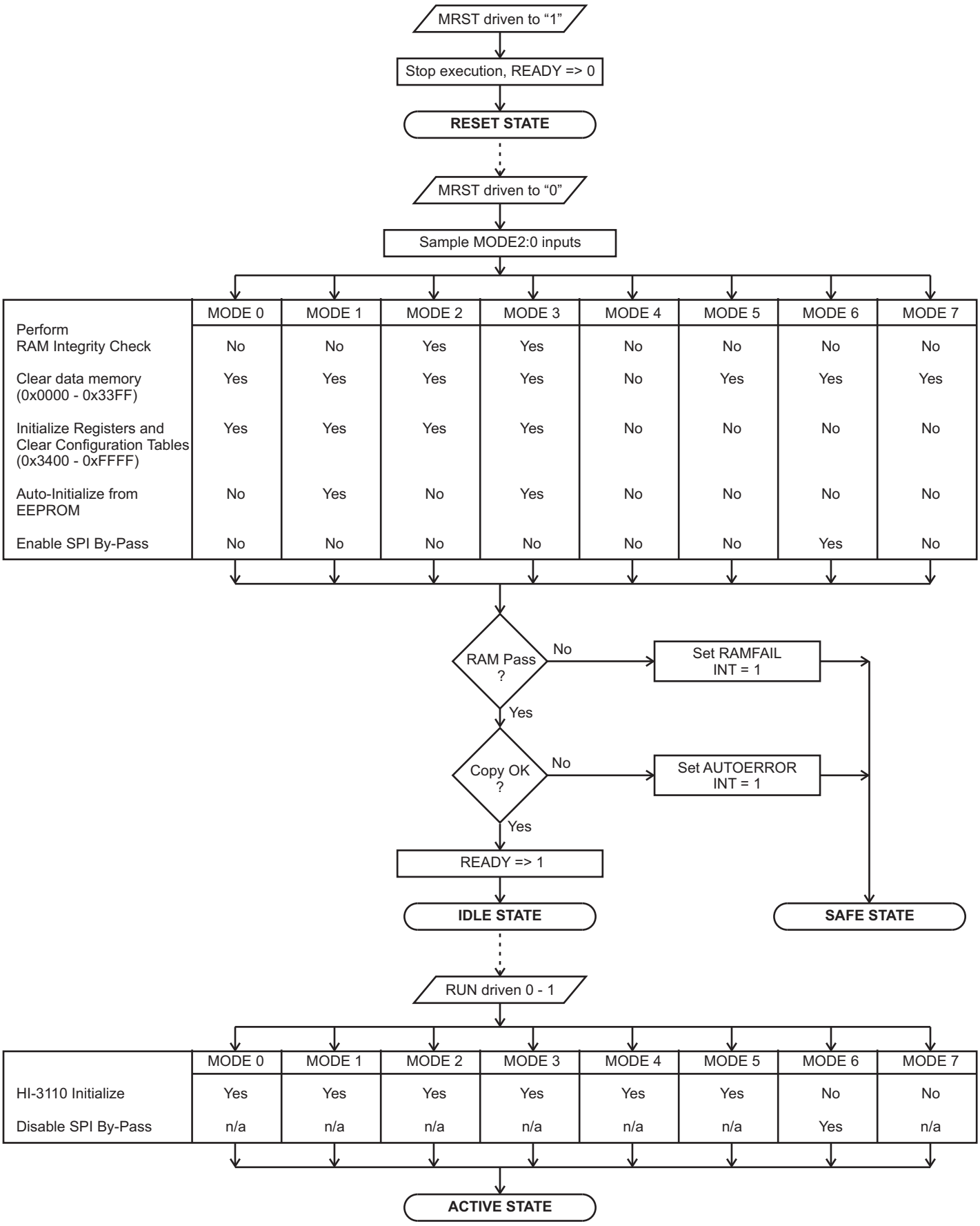
In Modes 0 through 5, as soon as the RUN input is transitioned from low to high, the contents of the CAN bus configuration registers (CANBTR0, CANBTR1) is transferred to the HI-3110 and other HI-3110 reset and initialization tasks are performed.

The following values are written to the HI-3110 registers in the sequence outlined below.

Write 0x88 to HI-3110 CTRL1 Register
Write Register 0x8030 to HI-3110 BTR0 Register
Write Register 0x8031 to HI-3110 BTR1 Register
Write 0x40 to HI-3110 STATE Register
Write 0x65 to HI-3110 GPINE Register
Write 0x07 to HI-3110 CTRL0 Register

The HI-3200 communicates with the HI-3110 over its dedicated SPI bus.

Following HI-3110 initialization, the HI-3200 enters the ACTIVE state and bus message processing begins.



	MODE 0	MODE 1	MODE 2	MODE 3	MODE 4	MODE 5	MODE 6	MODE 7
Perform RAM Integrity Check	No	No	Yes	Yes	No	No	No	No
Clear data memory (0x0000 - 0x33FF)	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes
Initialize Registers and Clear Configuration Tables (0x3400 - 0xFFFF)	Yes	Yes	Yes	Yes	No	No	No	No
Auto-Initialize from EEPROM	No	Yes	No	Yes	No	No	No	No
Enable SPI By-Pass	No	No	No	No	No	No	Yes	No

	MODE 0	MODE 1	MODE 2	MODE 3	MODE 4	MODE 5	MODE 6	MODE 7
HI-3110 Initialize	Yes	Yes	Yes	Yes	Yes	Yes	No	No
Disable SPI By-Pass	n/a	n/a	n/a	n/a	n/a	n/a	Yes	n/a

Figure 1. Reset and Start-up Operation flow chart.

INTERRUPT HANDLING

The HI-3200 includes a simple, user-selectable Interrupt Handler. Two types of Interrupt are possible - Message Event Driven (ARINC 429 or CAN Bus), and Fault Driven.

ARINC 429 Receive Interrupts

As described earlier, the user can elect to generate an interrupt upon receipt of an ARINC 429 message on any combination of the eight available channels and for any of the possible 256 label byte (ARINC message bits 1-8) values. Interrupts are enabled when the ARINC 429 Rx Interrupt look-up bit is a “1”.

When a message arrives that is flagged to generate an Interrupt, that channel’s bit is set in the ARINC 429 Receiver Pending Interrupt Register APIR. The ARINC 429 Interrupt Address Register (AIAR) for that channel is updated with the ARINC 429 8-bit label value.

For example, if ARINC Receive channel 7 is enabled for Interrupts when messages with ARINC label 0xD4 arrive, then on receipt of such a message, APIR bit 7 is set to a “1” and the value 0xD4 is written to AIAR7.

If the corresponding bit in the ARINC 429 Receive Interrupt Mask Register is a “1” the AINT interrupt output will go high and stay high until the AACK input pin is driven high. Driving AACK high, causes the AINT pin to return to zero.

A special Indexed SPI read instruction is available to allow the host to efficiently retrieve ARINC 429 messages which have Interrupts Enabled (see SPI instruction set section).

Note that if AACK is tied high permanently, the AINT pin will go high for approximately 1 us before returning to zero. A host CPU read of the APIR register reads the current value and resets APIR to 0x00.

CAN Bus Interrupts

An interrupt is generated on receipt of a CAN frame whose corresponding Filter and Interrupt look-up table bits are a “1”. The CANRX bit is set in the Pending Interrupt Register PIR, and the filter number which accepted the CAN frame is written to the CAN Interrupt Address Register CIAR. For example, if filter # 0xA1 accepts the frame, the value 0xA1 is written to CIAR.

The Interrupt output, MINT, is asserted (high) if the Interrupt Mask Register CANRX bit is a “1”.

A special SPI Instruction allows the user to extract the received frame information from the CAN receive memory without having to first determine its sixteen-bit absolute address. The CIAR value is used as a relative address pointer.

Fault Interrupts

There are four fault Interrupt bits in the PIR. Fault Interrupts are not maskable, and their Interrupt Mask bits are fixed at a “1”.

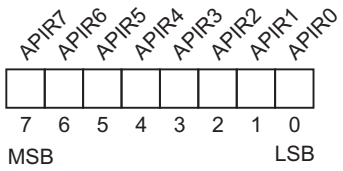
COPYERR is set when the HI-3200 detects a mismatch between RAM and EEPROM after attempting to program the Auto-initialization EEPROM.

AUTOERR is set when the Auto-Initialization EEPROM read verification cycle detects a mismatch between the on-chip memory and EEPROM following auto-initialization.

CHKERR is set when an auto-initialization checksum error is detected.

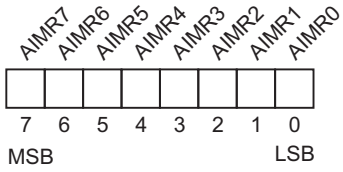
The RAMFAIL bit is set if the Built-In Self Test sequence fails.

ARINC 429 RECEIVE PENDING INTERRUPT REGISTER (Address 0x8000)



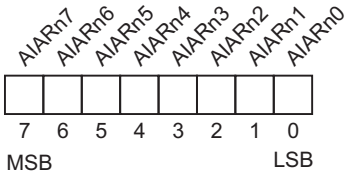
Bit	Name	R/W	Default	Description
7:0	APIR[7:0]	R	0	When a message is received on a given channel that triggers an interrupt, that channel's corresponding bit is set, e.g. if a message received on Rx channel 5 triggers an interrupt, the bit APIR5 will be set. If this bit is unmasked in the ARINC 429 Receive Interrupt Mask Register (see below), the AINT output pin is asserted. **Note:** This register is automatically cleared when read.

ARINC 429 RECEIVE INTERRUPT MASK REGISTER (Address 0x8020)



Bit	Name	R/W	Default	Description
7:0	AIMR[7:0]	R/W	0	Each bit in this register, AIMR[7:0], is a mask for a corresponding bit in the ARINC 429 Receive Pending Interrupt Register (APIR[7:0], described above). Writing a "1" to an AIMR bit results in assertion of the AINT output pin when the corresponding APIR bit is set (ARINC 429 message received). Writing a "0" to an AIMR bit will mask the corresponding APIR bit in the ARINC 429 Receive Pending Interrupt Register, resulting in non-assertion of the AINT pin when an ARINC 429 message is received.

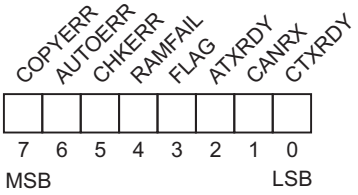
ARINC 429 RECEIVE INTERRUPT ADDRESS REGISTERS 0 - 7 (Address 0x8001 - 0x8008)



n = 0 - 7

Bit	Name	R/W	Default	Description
7:0	AIAR0[7:0]	R	0	The label number of the ARINC 429 message causing an interrupt is loaded into this register. Each channel has a corresponding ARINC 429 Receive Interrupt Address Register (AIAR0 - AIAR7).

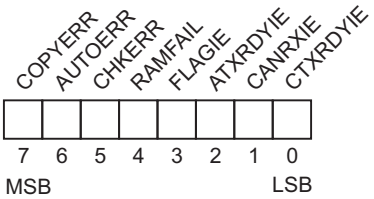
PENDING INTERRUPT REGISTER (Address 0x800A)



The MINT will be asserted when any of the bits in this register are set.

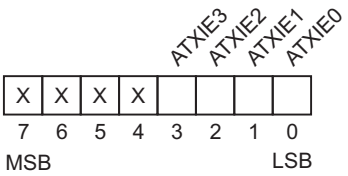
Bit	Name	R/W	Default	Description
7	COPYERR	R	0	EE copy error. RAM - EEPROM mismatch
6	AUTOERR	R	0	Auto-inititailization RAM read error
5	CHKERR	R	0	Auto-initialization checksum fail
4	RAMFAIL	R	0	Power-On Reset RAM Integrity Check fail
3	FLAG	R	0	Logical OR of ARINC 429 Receive FIFO FLAG signals
2	ATXRDY	R	0	ARINC 429 Host TX ready. Used with Host SPI op-code 100101TT (see Table 1). Interrupt when any of the four ARINC 429 transmitters are ready for the next 32-bit word from the host
1	CANRX	R	0	CAN Bus received frame Interrupt
0	CTXRDY	R	0	CAN Host Tx ready. Used with host SPI opcode 10010000. Interrupt when ready for next CAN frame from host.

PENDING INTERRUPT ENABLE REGISTER (Address 0x8034)



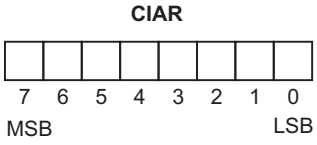
Bit	Name	R/W	Default	Description
7	COPYERR	R	1	COPYERR is not maskable
6	AUTOERR	R	1	AUTOERR is not maskable
5	CHKERR	R	1	CHKERR is not maskable
4	RAMFAIL	R	1	RAMFAIL is not maskable
3	FLAGIE	R/W	0	MINT pin is asserted if this bit is a “1” and the Pending Interrupt Register FLAG bit is set
2	ATXRDYIE	R/W	0	MINT pin is asserted if this bit is a “1” and the Pending Interrupt Register ATXRDY bit is set
1	CANRXIE	R/W	0	MINT pin is asserted if this bit is a “1” and the Pending Interrupt Register CANRX bit is set
0	CTXRDYIE	R/W	0	MINT pin is asserted if this bit is a “1” and the Pending Interrupt Register CTXRDY bit is set

ARINC 429 TX READY INT ENABLE (Address 0x8035)



Bit	Name	R/W	Default	Description
7	-	R/W	0	Not Used
6	-	R/W	0	Not Used
5	-	R/W	0	Not Used
4	-	R/W	0	Not Used
3	ATXIE3	R/W	0	Setting this bit generates an interrupt when ARINC 429 Transmitter 3 is ready to receive the next 32-bit word from the host for transmission. The ATXRDY bit in the Pending Interrupt Register will be set and the MINT pin will be asserted if the ATXRDYIE bit is set in the Pending Interrupt Enable Register.
2	ATXIE2	R/W	0	Setting this bit generates an interrupt when ARINC 429 Transmitter 2 is ready to receive the next 32-bit word from the host for transmission. The ATXRDY bit in the Pending Interrupt Register will be set and the MINT pin will be asserted if the ATXRDYIE bit is set in the Pending Interrupt Enable Register.
1	ATXIE1	R/W	0	Setting this bit generates an interrupt when ARINC 429 Transmitter 1 is ready to receive the next 32-bit word from the host for transmission. The ATXRDY bit in the Pending Interrupt Register will be set and the MINT pin will be asserted if the ATXRDYIE bit is set in the Pending Interrupt Enable Register.
0	ATXIE0	R/W	0	Setting this bit generates an interrupt when ARINC 429 Transmitter 0 is ready to receive the next 32-bit word from the host for transmission. The ATXRDY bit in the Pending Interrupt Register will be set and the MINT pin will be asserted if the ATXRDYIE bit is set in the Pending Interrupt Enable Register.

CAN INTERRUPT ADDRESS REGISTER (Address 0x800B)

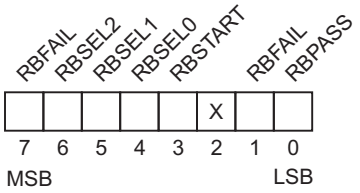


RAM BUILT-IN SELF-TEST

The HI-3200 offers a built-in self-test (BIST) feature which can be used to check RAM integrity. The BIST Control/Status Register is used to control the BIST function. All tests are destructive, overwriting data present before test commencement.

NOTE: To use BIST self-tests, the part needs to be in Mode 4 with the RUN pin low.

BIST CONTROL/STATUS REGISTER (Address 0x8070)



This register controls RAM built-in self-test. Bits 0,1 are Read Only.

BIST Control Register bits provide a means for the host to perform RAM self-test at other times. Register bits 6:4 select RAM test type. Then bit 3 starts the selected RAM test, and bits 1:0 report a fail/pass result after test completion.

Bit No. *Mnemonic* *Interrupt Type*

7 RBFFAIL RAM BIST Force Failure.
When this bit is asserted, RAM test failure is forced to verify that RAM BIST logic is functional.
For Modes 2 and 3, the RBFFAIL bit-7 should be cleared to 0 by writing to the BIST Control/Status register (Address 0x8070) BEFORE taking MRST high.

6:4 RBSEL2-0 RAM BIST Select Bits 2-0.
This 3-bit field selects the RAM BIST test mode applied when the RBSTART bit is set:

RBSEL2:0	SELECTED RAM TEST
000	Idle
001	Pattern Test, described below
010	Write 0x00 to RAM address range 0x0000 - 0x7FFF
011	Read and verify 0x00 over RAM address range 0x0000 - 0x7FFF
100	Write 0xFF to RAM address range 0x0000 - 0x7FFF
101	Read and verify 0xFF over RAM address range 0x0000 - 0x7FFF
110	Inc / Dec Test performs only steps 5 - 8 of the Pattern Test below
111	Idle

Description of the RAM BIST "PATTERN" test selected when register bits RBSEL2:0 = 001:

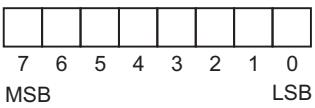
1. Write 0x00 to all RAM locations, 0x0000 through 0x7FFF
2. Repeat the following sequence for each RAM location from 0x0000 through 0x7FFF:
   a. Read and verify 0x00
   b. Write then read and verify 0x55
   c. Write then read and verify 0xAA
   d. Write then read and verify 0x33
   e. Write then read and verify 0xCC
   f. Write then read and verify 0x0F
   g. Write then read and verify 0xF0
   h. Write then read and verify 0x00
   I. Write then read and verify 0xFF
   j. Write 0x00 then increment RAM address and go to step (a)
3. Write 0xFF to all RAM locations, 0x0000 through 0x7FFF
4. Repeat the following sequence for each RAM location from 0x0000 through 0x7FFF:

a. Read and verify 0xFF
b. Write then read and verify 0x55
c. Write then read and verify 0xAA
d. Write then read and verify 0x33
e. Write then read and verify 0xCC
f. Write then read and verify 0x0F
g. Write then read and verify 0xF0
h. Write then read and verify 0x00
I. Write then read and verify 0xFF
j. Write 0xFF then increment RAM address and go to step (a)

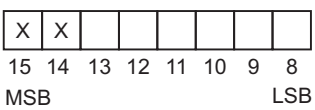
5. Write an incrementing pattern into sequential RAM locations from 0x0000 to 0x7FFF
6. Read each memory location from 0x0000 to 0x7FFF and verify the contents
7. Write 1s complement of each cell's current contents, into each RAM location (same addr range)
8. Read each memory location and verify the contents

3	RBSTRT	RAM BIST Start. Writing logic 1 to this bit initiates the RAM BIST test selected by register bits RBSEL2:0. The RBSTRT bit can only be set in MODE2:0 = 0x04. This bit is automatically cleared upon test completion. Register bits 1:0 indicate fail / pass test result.
2	---------	Not Used.
1	RBFAIL	RAM BIST Fail. Device logic asserts this bit when failure occurs while performing the selected RAM test. This bit is automatically cleared when RBSTRT bit 3 is set. When BIST failure occurs, a clue to the failing RAM address can be read at register addresses 0x8071 and 0x8072. For speed, the RAM BIST concurrently tests four consecutive RAM addresses in parallel. If a test failure occurs, register addresses 0x8071 and 0x8072 can be used to determine the four RAM addresses tested.
0	RBPASS	RAM BIST Pass. Device logic asserts this bit when the selected RAM test completes without error. This bit is automatically cleared when RBSTRT bit 3 is set.

LOWER BIST FAIL ADDRESS REGISTER
(Address 0x8071)



UPPER BIST FAIL ADDRESS REGISTER
(Address 0x8072)



HOST SERIAL PERIPHERAL INTERFACE

In the HI-3200, internal RAM and registers occupy a (32K + 128) x 8 address space. The lowest 32K addresses access RAM locations and the remaining addresses access registers. Timing is identical for register operations and RAM operations via the serial peripheral interface, and read and write operations have likewise identical timing.

Host access is only allowed when the part is READY or in SAFE mode. **NOTE:** writes will be blocked and reads will return the Master Status Register value until either of these modes occur.

Serial Peripheral Interface (SPI) Basics

The HI-3200 uses an SPI synchronous serial interface for host access to registers and RAM. Host serial communication is enabled through the Chip Select ($\overline{CS}$) pin, and is accessed via a three-wire interface consisting of Serial Data Input (SI) from the host, Serial Data Output (SO) to the host and Serial Clock (SCK). All programming cycles are completely self-timed, and no erase cycle is required before write.

The SPI (Serial Peripheral Interface) protocol specifies master and slave operation; the HI-3200 Host CPU interface operates as an SPI slave.

The SPI protocol defines two parameters, CPOL (clock polarity) and CPHA (clock phase). The possible CPOL-CPHA combinations define four possible "SPI Modes." Without describing details of the SPI modes, the HI-3200 operates in the two modes where input data for each device (master and slave) is clocked on the rising edge of SCK, and output data for each device changes on the falling edge. These are known as SPI Mode 0 (CPHA = 0, CPOL = 0) and SPI Mode 3 (CPHA = 1, CPOL = 1). Be sure to set the host SPI logic for one of these modes.

As seen in Figure 2, the difference between SPI Modes 0 and 3 is the idle state for the SCK signal. There is no configuration setting in the HI-3200 to select SPI Mode 0 or Mode 3 because compatibility is automatic. Beyond this point, the HI-3200 data sheet only shows the SPI Mode 0 SCK signal in timing diagrams.

The SPI protocol transfers serial data as 8-bit bytes. Once $\overline{CS}$ chip select is asserted, the next 8 rising edges on SCK latch input data into the master and slave devices, starting with each byte's most-significant bit. The HI-3200 SPI can be clocked at 20 MHz.

Multiple bytes may be transferred when the host holds $\overline{CS}$ low after the first byte transferred, and continues to clock SCK in multiples of 8 clocks. A rising edge on $\overline{CS}$ chip select terminates the serial transfer and reinitializes the HI-3200 SPI for the next transfer. If $\overline{CS}$ goes high before a full byte is clocked by SCK, the incomplete byte clocked into the device SI pin is discarded.

In the general case, both master and slave simultaneously send and receive serial data (full duplex) as shown in Figure 2 below. When the HI-3200 is sending data on SO during read operations, activity on its SI input is ignored. Figures 3 and 4 show actual behavior for the HI-3200 SO output.

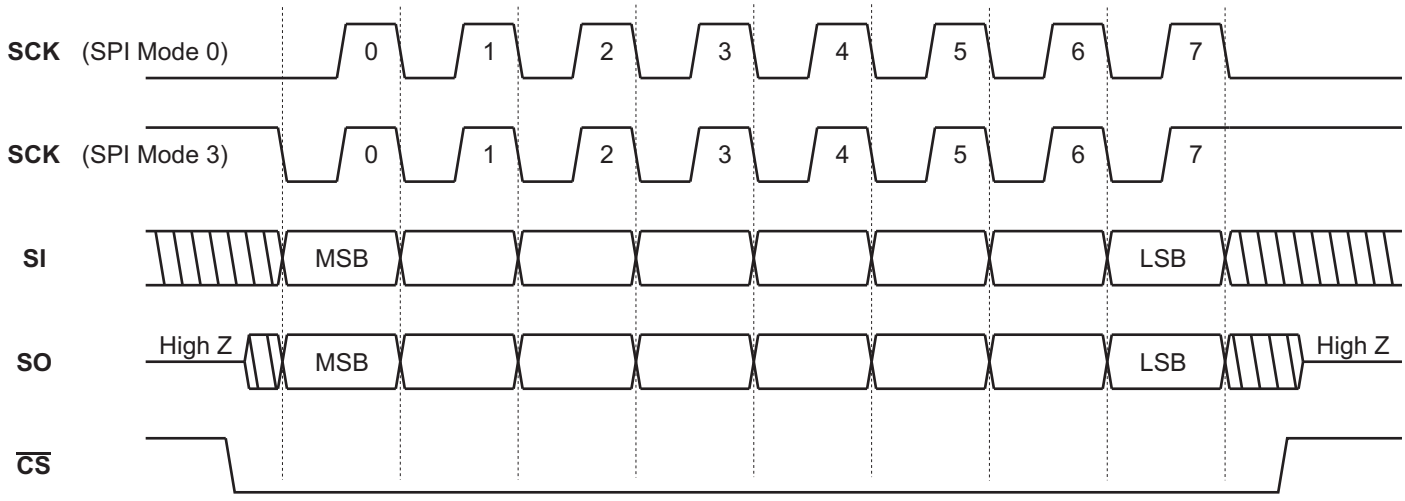


FIGURE 2. Generalized Single-Byte Transfer Using SPI Protocol, SCK is Shown for SPI Modes 0 and 3

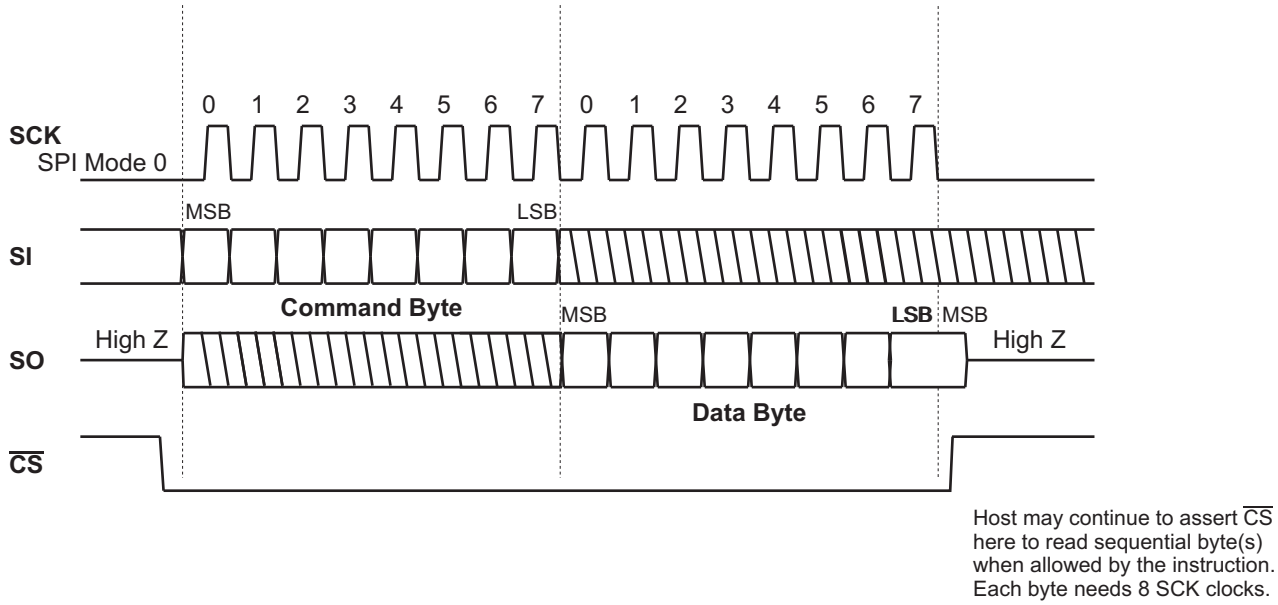


FIGURE 3. Single-Byte Read From RAM or a Register

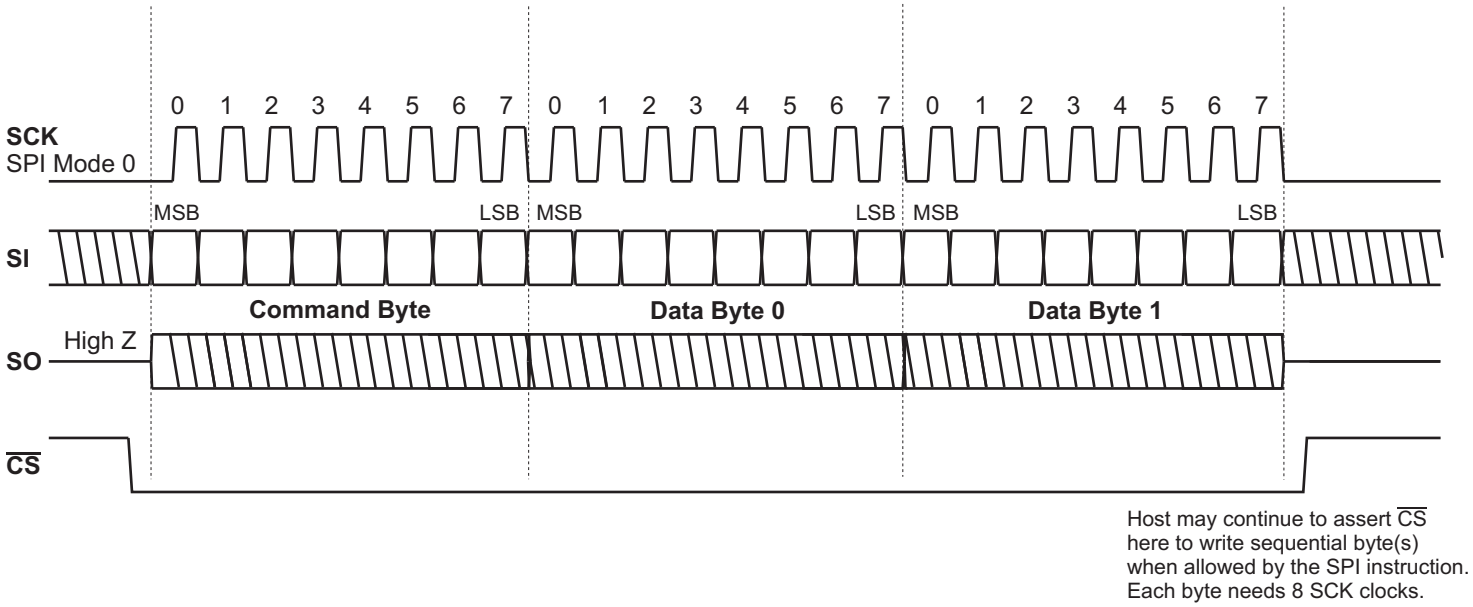


FIGURE 4. 2-Byte Write To RAM or a Register Pair

HI-3200 SPI COMMANDS

For the HI-3200, each SPI read or write operation begins with an 8-bit command byte transferred from the host to the device after assertion of $\overline{CS}$. Since HI-3200 command byte reception is half-duplex, the host discards the dummy byte it receives while serially transmitting the command byte.

Fast Access Commands for Registers 0-15

The SPI command set includes directly-addressed read and write commands for registers 0 through 15 (Memory Address 0x8000 to 0x800F). The 8-bit pattern for these commands has the general form

0-W-R-R-R-R-0-0

where RRRR is the 4-bit register number, and W signifies Write when 1, or Read when 0.

Figures 3 and 4 show read and write timing as it appears for fast-access register operations. The command byte is immediately followed by a data byte comprising the 8-bit data word read or written. For a single register read or write, $\overline{CS}$ is negated after the data byte is transferred.

Multiple register read or write cycles may be performed by transferring more than one byte before $\overline{CS}$ is negated. Multiple register access occur in address order starting with the register specified in the SPI instruction.

Note: Register locations not shown in table 1 are "reserved" and cannot be written using any SPI command. Further, these register addresses will not provide meaningful data in response to read commands.

RAM and Register Indirect Addressing

Refer to the HI-3200 SPI command set shown in Table 1. SPI commands other than fast-access use an address pointer to indicate the address for read or write transactions. This sixteen-bit memory address pointer (MAP) must be initialized before any non-fast-access read or write operation. Two dedicated SPI instructions are used to write and read the MAP. SPI Instruction 0x8C followed by two data bytes is used to write MAP. SPI instruction 0x88 reads two data bytes from MAP. The first byte is the most significant eight bytes of the address. For example, SPI sequence 0x8C, 0x12, 0x34 write the value 0x1234 into the MAP.

Two SPI instructions read and write data bytes to memory or registers using the MAP as an address pointer. Single or multi-byte reads and writes may be performed. MAP is incremented after each byte access.

Two command bytes cannot be "chained"; $\overline{CS}$ must be negated after the command, then reasserted for the following Read or Write command.

Note: When the primary or fast-access address pointer is used for auto-incrementing multi-word read/write and reaches the top of the memory address range (0x7FFF), or the top of the register address range (0xFFFF) attempts to read further bytes will result the terminal address (0x7FFF or 0xFFFF) being output again. The host should avoid this situation.

Two single-byte SPI commands use the current address pointer value in MAP without first loading or otherwise modifying it:

Command	Read Operation
0x80	read location addressed by pointer value

Command	Write Operation
0x84	write location addressed by pointer value

Either of these commands can be used to read or write a single location, or may be used when starting a multi-byte read or write by using the pointer's auto-increment feature.

Special Purpose Commands

Several other HI-3200 SPI commands load or otherwise modify the memory address pointer before initiating a read or write process. These commands are designed to allow speedy access to messages received on the ARINC 429 or CAN buses.

Using a single-byte SPI command, the address pointer can be directly loaded with the memory address for the last received ARINC 429 message which triggered an interrupt.

Op Code 110RRR00
The HI-3200 will retrieve the current ARINC Receive Interrupt Vector for a given channel (RRR), calculate the memory address for the first word of the corresponding receive memory data block and write it to the Memory Address pointer (MAP). Read the location addressed by the new pointer value.

This command can be used to read just the most recent ARINC 429 Receive Status Byte, or may be used to start a four-byte read because memory pointer auto-increment occurs after the Status Byte is read.

Op Code 111RRR00
The HI-3200 will retrieve the current ARINC Receive Interrupt Vector for a given channel (RRR), calculate the memory address for the first word of the corresponding receive memory data block and write it to the Memory Address Pointer (MAP). Output the value of the Receive Interrupt Vector (ARINC 429 label byte).

This command can be used to read just the most recent ARINC 429 label value received, or may be used to start a four-byte read to output the entire four-byte ARINC

message, because memory pointer auto-increment occurs after the label byte is output.

Op Code 0x9C
This command can be used to read just the most recent CAN frame Information byte received, or may be used to start a sixteen-byte read to output the entire sixteen-byte received CAN frame memory block, because memory pointer auto-increment occurs after the first byte is output.

Op Code 0x90
Writes a CAN frame to the CAN transmit scheduler for immediate transmission.

Op Code 100101TT
Writes an ARINC 429 message to ARINC 429 transmit scheduler TT for immediate transmission, where TT represents the channel number.

TABLE 1. DEFINED INSTRUCTIONS

OP CODE Binary	OP CODE Hex	Auto Increment	Number of Data Bytes	DESCRIPTION
00RRRR00	0x00 - 0x3C	Yes	1++	Fast Register Read from register RRRR
01RRRR00	0x40 - 0x7C	Yes	1++	Fast Register Write to register RRRR
10000000	0x80	Yes	1++	Read memory at address MAP
10000100	0x84	Yes	1++	Write memory at address MAP
10001000	0x88	No	2	Read MAP
10001100	0x8C	No	2	Write MAP
10010000	0x90	No	5 - 13	Transmit CAN Frame
100101TT	0x94 - 0x97	No	4	Transmit ARINC 429 message on transmit bus TT
10011100	0x9C	No	16	Read CAN Frame at filter block <CIAR>
101RRR00	0xA0 - 0xBC	Yes	4, 8, 12...	Read ARINC 429 FIFO # RRR. Reads exactly four bytes
110RRR00	0xC0 - 0xDC	No	4	Read ARINC block at receive channel RRR, label <AIARn>
111RRR00	0xE0 - 0xFC	No	4	Read ARINC message at receive channel RRR, label <AIARn>

FAST-ACCESS SPI COMMANDS FOR REGISTERS 0-15
Command Bits 5:2 Convey the 4-Bit Register Address

COMMAND BITS 7 6 5 4 3 2 1 0	HEX BYTE	FAST-ACCESS READ
0 0 0 0 0 0 0 0	0x00	Read APIR
0 0 0 0 0 1 0 0	0x04	Read AIAR0
0 0 0 0 1 0 0 0	0x08	Read AIAR1
0 0 0 0 1 1 0 0	0x0C	Read AIAR2
0 0 0 1 0 0 0 0	0x10	Read AIAR3
0 0 0 1 0 1 0 0	0x14	Read AIAR4
0 0 0 1 1 0 0 0	0x18	Read AIAR5
0 0 0 1 1 1 0 0	0x1C	Read AIAR6
0 0 1 0 0 0 0 0	0x20	Read AIAR7
0 0 1 0 0 1 0 0	0x24	Reserved
0 0 1 0 1 0 0 0	0x28	Read PIR
0 0 1 0 1 1 0 0	0x2C	Read CIAR
0 0 1 1 0 0 0 0	0x30	Read AMFF
0 0 1 1 0 1 0 0	0x34	Read ATRB
0 0 1 1 1 0 0 0	0x38	Read MSR
0 0 1 1 1 1 0 0	0x3C	Read MCR

COMMAND BITS 7 6 5 4 3 2 1 0	HEX BYTE	FAST-ACCESS WRITE
0 1 0 0 0 0 0 0	0x40	N/A (Read only)
0 1 0 0 0 1 0 0	0x44	N/A (Read only)
0 1 0 0 1 0 0 0	0x48	N/A (Read only)
0 1 0 0 1 1 0 0	0x4C	N/A (Read only)
0 1 0 1 0 0 0 0	0x50	N/A (Read only)
0 1 0 1 0 1 0 0	0x54	N/A (Read only)
0 1 0 1 1 0 0 0	0x58	N/A (Read only)
0 1 0 1 1 1 0 0	0x5C	N/A (Read only)
0 1 1 0 0 0 0 0	0x60	N/A (Read only)
0 1 1 0 0 1 0 0	0x64	Reserved
0 1 1 0 1 0 0 0	0x68	N/A (Read only)
0 1 1 0 1 1 0 0	0x6C	N/A (Read only)
0 1 1 1 0 0 0 0	0x70	N/A (Read only)
0 1 1 1 0 1 0 0	0x74	N/A (Read only)
0 1 1 1 1 0 0 0	0x78	N/A (Read only)
0 1 1 1 1 1 0 0	0x7C	Write MCR

HOST SPI BY-PASS

When the HI-3200 is reset and initialization Mode 6 is selected (by setting MODE2:0 inputs to “110”), the SPI by-pass function is enabled when the device is in the idle state. SPI By-Pass allows the host CPU to communicate directly with the HI-3110 CAN controller via its dedicated SPI interface, by-passing the HI-3200. SPI command sequences from the host CPU may then directly interrogate or program the attached HI-3110.

In normal operation, the HI-3200 handles all HI-3110 control functions. SPI By-Pass is intended only as a system debugging aid.

For a full description of the HI-3110 and its SPI instruction set, please refer to the latest revision of the HI-3110 data sheet.

SPI By-Pass mode is disabled as soon as the RUN input is taken high and the device enters the ACTIVE state. Any further toggling of the RUN pin will not re-initiate SPI by-pass mode.

PROGRAMMING THE AUTO-INITIALIZATION EEPROM.

Following reset, the HI-3200 may be completely configured by automatically copying the contents of an external EEPROM into HI-3200 memory and registers. An SPI enabled 64KByte EEPROM is used for this purpose. The EEPROM memory space is mapped to the HI-3200 as shown in the diagram below.

All configuration memory blocks are copied. The ARINC 429 Received Data Memory contents, ARINC 429 Receive log FIFO contents, and CAN Bus Received Data Memory contents are not copied to or from the EEPROM.

The HI-3200 can be used to program the Auto-Initialization EEPROM. When the HI-3200 is in its IDLE state (RUN input = “0”), a three step sequence must be performed to begin the EEPROM programming cycle:

1. Write data value 0x5A to HI-3200 memory address 0x8FFF.

2. Write data value 0xA5 to HI-3200 memory address 0x8FFF.

3. Apply a positive pulse to the PROG input pin of at least 1 ms.

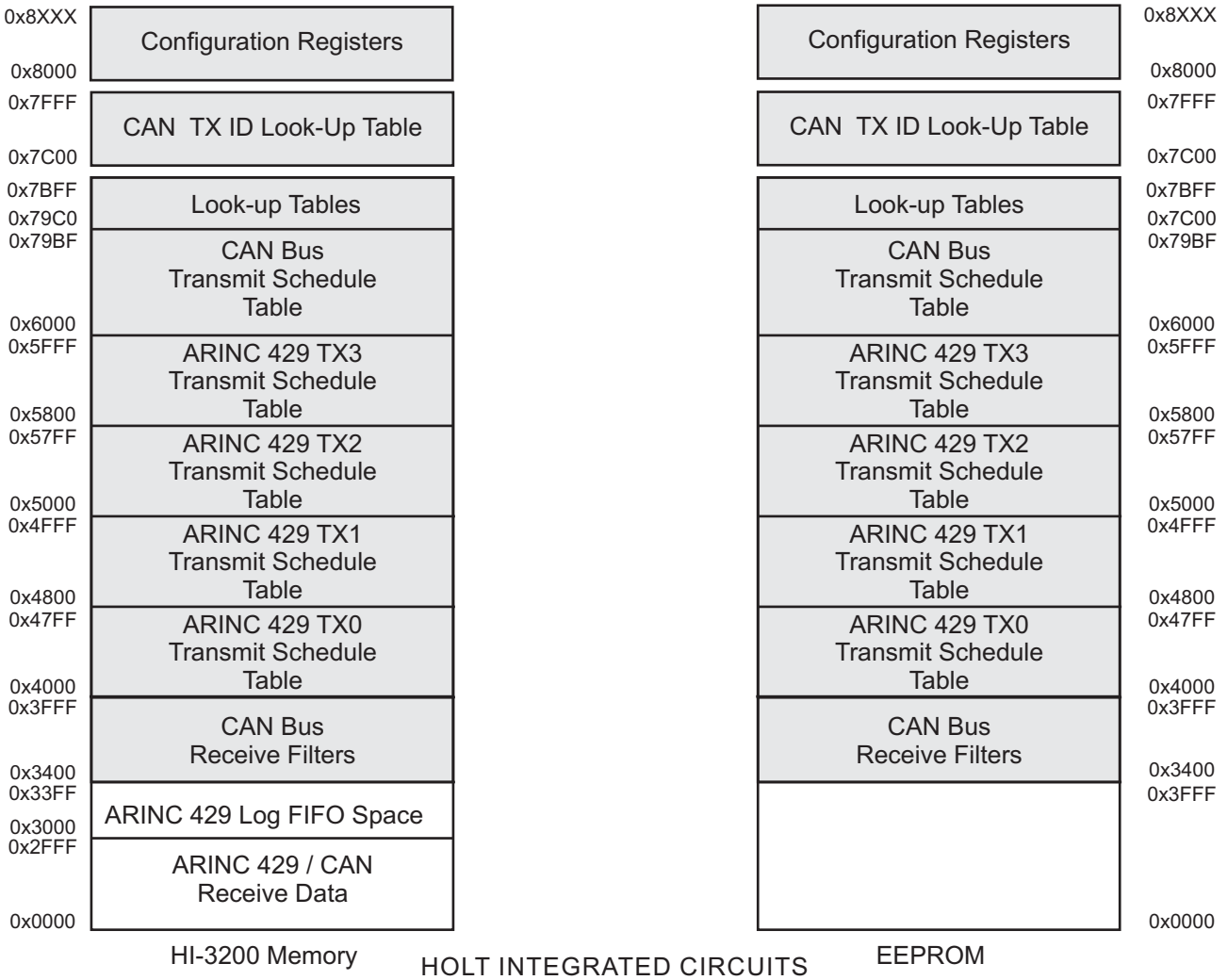
If the three-step sequence is interrupted by any intervening host activity between steps 1 and 2, or 2 and 3, or if the PROG pulse is less than 1 ms, the programming cycle will not start and the device remains in the IDLE state.

Taking the PROG pin low initiates the cycle. The READY pin goes low, and the contents of the HI-3200 memory and registers are copied to the EEPROM. When copying is complete, the HI-3200 executes a byte-by-byte comparison of the EEPROM and its own register / memory contents. If the verification completes successfully, the READY pin goes high.

A 2’s complement of the checksum is also written to the EEPROM at location 0x807F. The total read back checksum should be zero. The following locations are excluded from the checksum because they are either read-only or unused locations: 0x8000 thru 0x800e, 0x8023 thru 0x802f, 0x8036 thru 0x805e, 0x8070 thru 0x807e.

If the comparison of the EEPROM contents and HI-3200 memory / register contents results in a discrepancy, the HI-3200 enters the SAFE state, the PROGERR bit is set in the Pending Error Register and the INT output is asserted.

The user must clear the PROGERR issue before normal operation can resume.



ABSOLUTE MAXIMUM RATINGS

Supply voltage (VDD)	-0.3 V to +5.0 V
Logic input voltage range	-0.3 V DC to +3.6 V
Power dissipation at 25°C	1.0 W
Reflow Solder Temperature	260°C
Junction Temperature	175°C
Storage Temperature	-65°C to +150°C

RECOMMENDED CONDITIONS

Operating Supply Voltage

VDD..... 3.3 VDC ±5%

Operating Temperature Range

Industrial ..... -40°C to +85°C
Extended ..... -55°C to +125°C

NOTE: Stresses above absolute maximum ratings or outside recommended operating conditions may cause permanent damage to the device. These are stress ratings only. Operation at the limits is not recommended.

DC ELECTRICAL CHARACTERISTICS

VDD = 3.3 V, GND = 0V, T_A = Operating Temperature Range (unless otherwise specified).

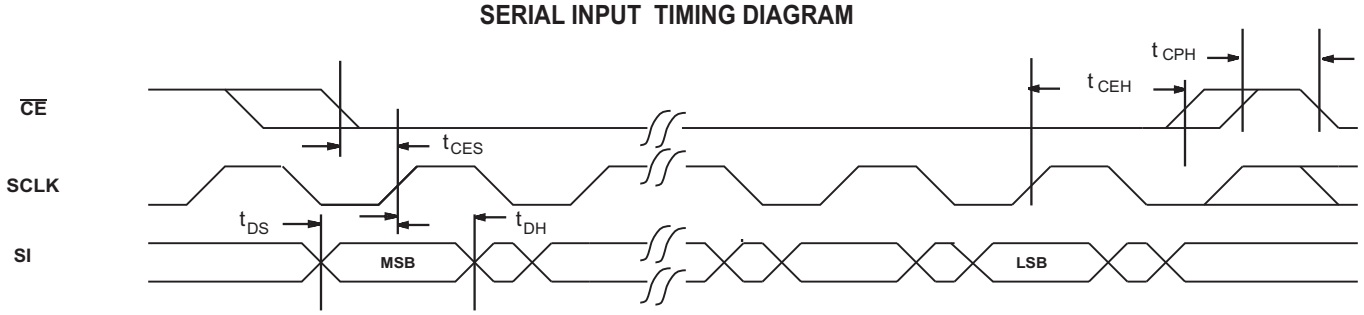
PARAMETER	SYMBOL	CONDITION	MIN	TYP	MAX	UNITS
Operating Voltage	VDD		3.15	3.30	3.45	V
Supply Current	IDD				50	mA
Min. Input Voltage (HI)	V_{IH}	Digital inputs	70%			VDD
Max. Input Voltage (LO)	V_{IL}	Digital inputs			30%	VDD
Digital Inputs Pull-Up / Pull-Down Current	I_{PUD}	Digital inputs and data bus		30	100	µA
Min. Output Current (HI)	I_{OH}	V_{OH} = 0.8 VDD VDD = 3.15 - 3.45 V	-6.0			mA
Max. Output Current (LO)	I_{OL}	V_{OL} = 0.4 V VDD = 3.15 - 3.45 V			6.0	mA
Min. Output Voltage (HI)	V_{OH}	I_{OUT} = -1.0mA, Digital outputs	90%			VDD
Max. Output Voltage (LO)	V_{OL}	I_{OUT} = 1.0mA, Digital outputs			10%	VDD

AC ELECTRICAL CHARACTERISTICS

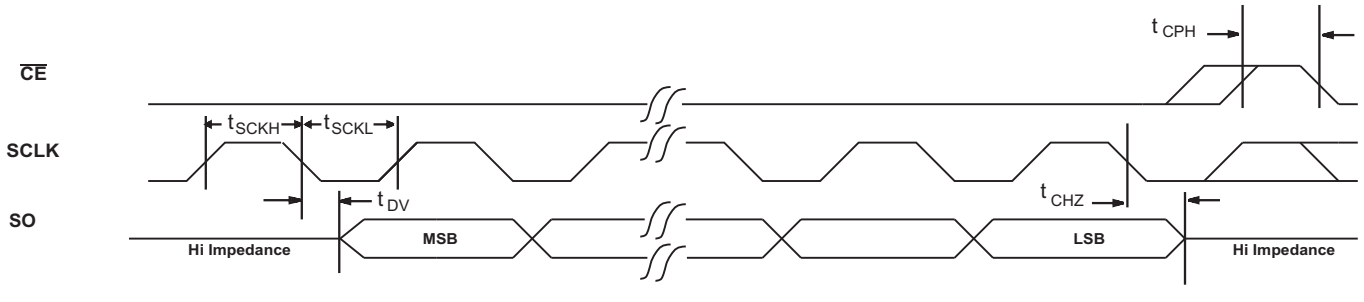
VDD = 3.3 V, GND = 0V, T_A = Operating Temperature Range (unless otherwise specified).

PARAMETER	SYMBOL	LIMITS			UNITS
		MIN	TYP	MAX	
SPI Host Bus Interface					
SCK clock period	t_{CYC}	50			ns
$\overline{CE}$ set-up time to first SCK rising edge	t_{CES}	25			ns
$\overline{CE}$ hold time after last SCK falling edge	t_{CEH}	25			ns
$\overline{CE}$ inactive between SPI instructions	t_{CPH}	100			ns
SPI SI Data set-up time to SCK rising edge	t_{DS}	10			ns
SPI SI Data hold time after SCK rising edge	t_{DH}	10			ns
SCK high time	t_{SCKH}	25			ns
SCK low time	t_{SCKL}	25			ns
SO valid after SCK falling edge	t_{DV}			20	ns
SO high-impedance after $\overline{CE}$ inactive	t_{CHZ}			75	ns

SERIAL INPUT TIMING DIAGRAM

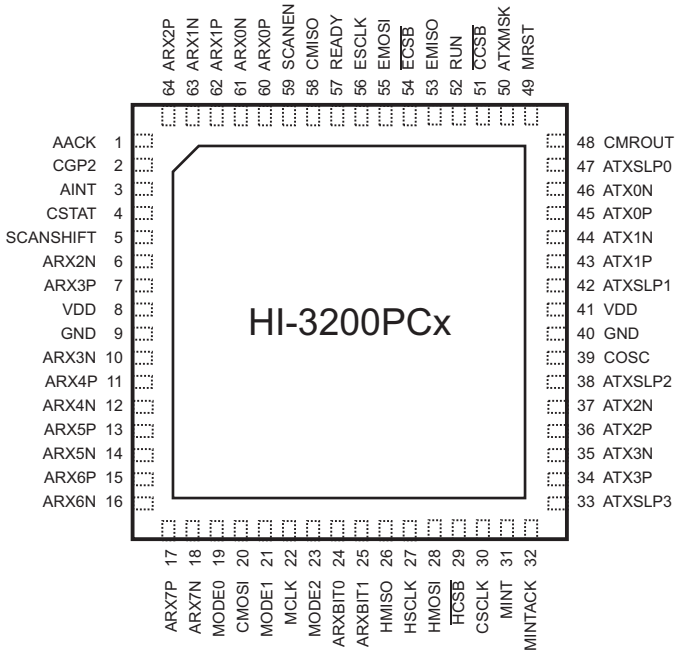


SERIAL OUTPUT TIMING DIAGRAM



PIN CONFIGURATION FOR HI-3200, 64-PIN QFN PACKAGE

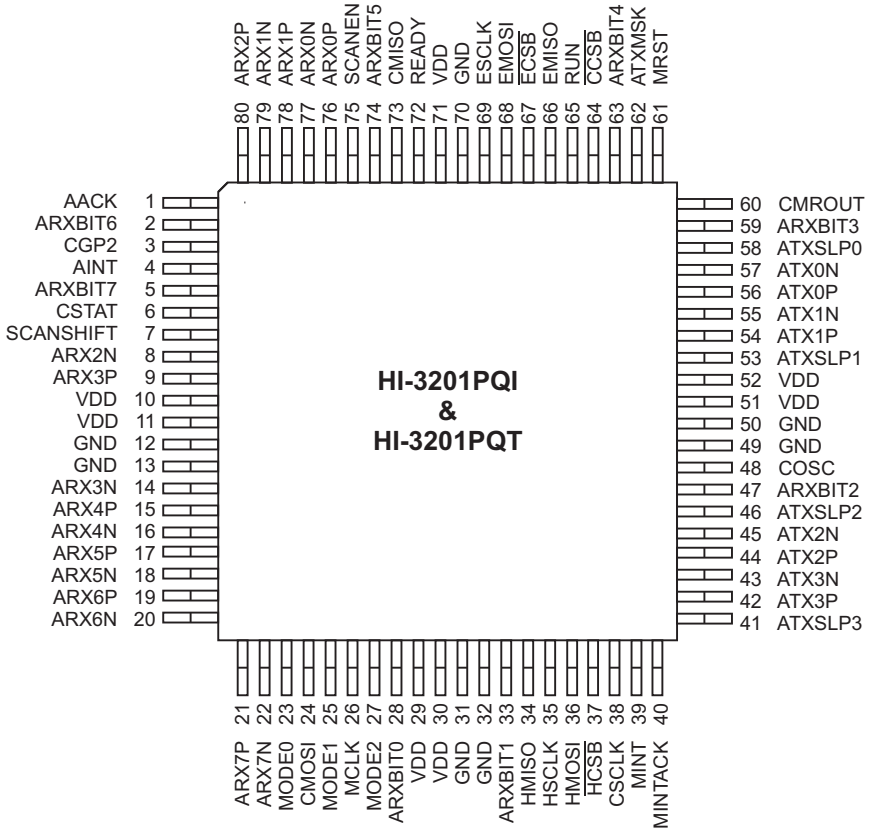
Notes
1 . All VDD and GND pins must be connected.
2. See data sheet page 1 for HI-3200, 64-Pin PQFP Package Configuration.



TOP VIEW

PIN CONFIGURATION FOR HI-3201, 80-PIN PQFP PACKAGE

Notes
1 . All VDD and GND pins must be connected.
2. See data sheet page 1 for HI-3200, 64-Pin PQFP Package Configuration.



ORDERING INFORMATION

HI-3200xx x x

PART NUMBER	PACKAGE DESCRIPTION
Blank	Tin / Lead (Sn / Pb) Solder
F	100% Matte Tin (Pb-free RoHS compliant)

PART NUMBER	TEMPERATURE RANGE	FLOW	BURN IN
I	-40°C TO +85°C	I	No
T	-55°C TO +125°C	T	No
M	-55°C TO +125°C	M	Yes

PART NUMBER	PACKAGE DESCRIPTION
PQ	64 THIN PIN PLASTIC QUAD FLAT PACK TQFP (64PQTS)
PC	64-PIN PLASTIC CHIP-SCALE PACKAGE QFN (64PCS)

HI-3201PQ x F

PART NUMBER	PACKAGE DESCRIPTION
F	100% Matte Tin (Pb-free RoHS compliant)

PART NUMBER	TEMPERATURE RANGE	FLOW	BURN IN
I	-40°C TO +85°C	I	No
T	-55°C TO +125°C	T	No
M	-55°C TO +125°C	M	Yes

PART NUMBER	PACKAGE DESCRIPTION
PQ	80 THIN PIN PLASTIC QUAD FLAT PACK TQFP (80PTQS)

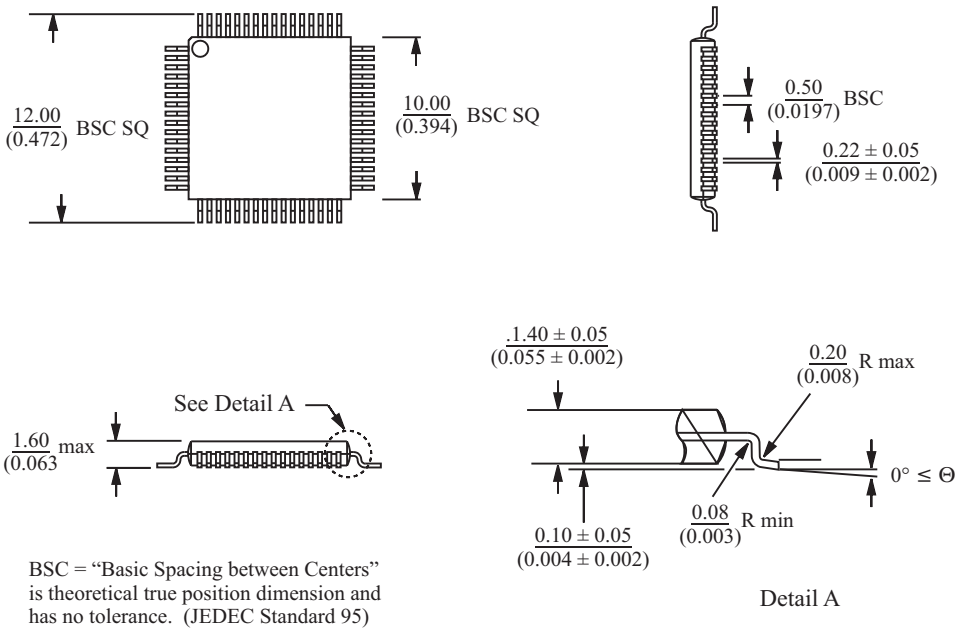
REVISION HISTORY

Document	Rev.	Date	Description of Change
DS3200	New	5/4/11	Initial Release.
	A	1/25/12	Corrected typo in ordering information.
	B	4/16/12	Corrected fonts in numerous block diagrams.
	C	5/7/13	Added description for register 0x8035, “ARINC 429 TX Ready INT Enable”. Updated QFN package dimensions.
	D	8/13/13	Updated Programming for Auto-Initialization EEPROM section. Update Reflow Solder Temperature.
	E	10/21/13	Added description of ARINC 429 Receive Pending Interrupt Register, ARINC 429 Receive Interrupt Mask Register and ARINC 429 Receive Interrupt Address Registers.
	F	11/4/13	Reverse direction of signals on EEPROM in Block Diagram and Examples 4, 5 and 8. Add “bar” (active low) to CCSB, ECSB and HCSB signals in Pin Descriptions. Updated PQFP-64, QFN-64 and PQFP-80 package drawings.
	G	01/29/14	Clarified description of Repetition Rate Register.
	H	08/04/14	Add notes to clear RBFFAIL bit before applying RESET for modes 2 and 3. Add description of ARINC 429 Muxed FIFO Flags Register and ARINC 429 Tx Ready Bits Register.
	I	11/04/15	Clarify RUN pin needs to be low to use BIST in Mode 4.
	J	09/06/16	Add tolerance for MCLK input. Correct label for RESET pin (should be MRST pin). Clarify Active High status for AACK, AINT, MINT and MINTACK pins. Add table showing Start-up time for each mode (The time from falling edge of MRST to READY pin high).
	K	10/19/16	Specify Pull-up/Pull-down for digital inputs in Pin Description Table. Note EEPROM SPI clock is 8 MHZ max. Specify max current for digital inputs. Specify maximum output current for digital outputs.
	L	06/19/17	Clarify function of TXxRDY bits and use of BIST.
	M	10/01/17	Update 80-pin PQFP package. Incorrect number of pins shown.

64 PIN PLASTIC QUAD FLAT PACK (PQFP)

millimeters (inches)

Package Type: 64PQTS

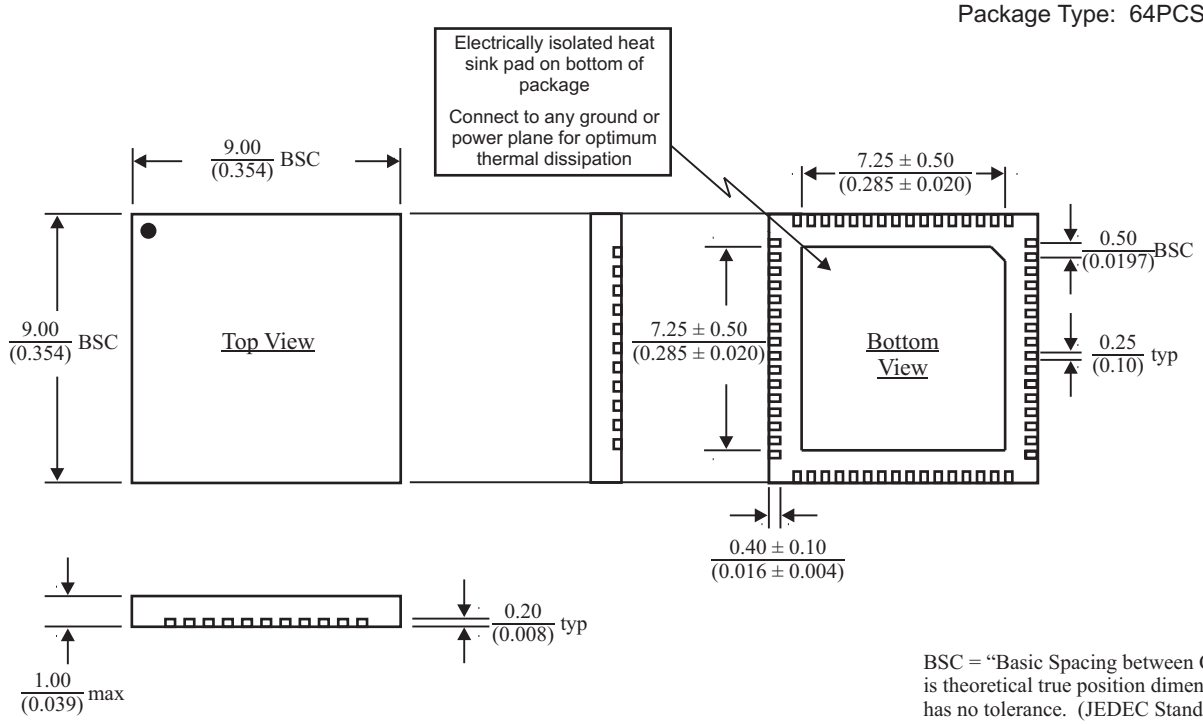


BSC = "Basic Spacing between Centers" is theoretical true position dimension and has no tolerance. (JEDEC Standard 95)

64-PIN PLASTIC CHIP-SCALE PACKAGE (QFN)

millimeters (inches)

Package Type: 64PCS

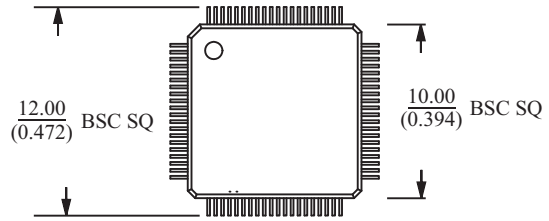


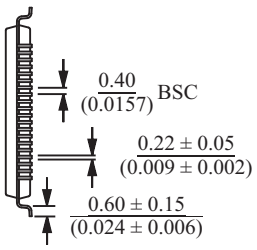
BSC = "Basic Spacing between Centers" is theoretical true position dimension and has no tolerance. (JEDEC Standard 95)

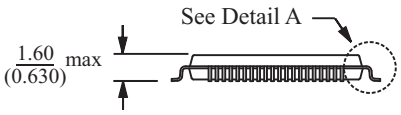
80 PIN PLASTIC QUAD FLAT PACK (PQFP)

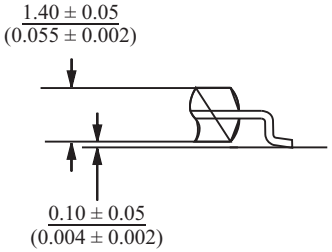
millimeters (inches)

Package Type: 80PTQS









Detail A

BSC = "Basic Spacing between Centers" is theoretical true position dimension and has no tolerance. (JEDEC Standard 95)