HI-3110H, HI-3111H, HI-3112H

Avionics CAN Controller with Integrated Transceiver and High Operating Temperature

March 2021

GENERAL DESCRIPTION

The HI-3110H is a standalone Controller Area Network (CAN) controller with built in transceiver optimized for use in high temperature avionics applications. The device is capable of operating at extended temperature ranges of -55°C to 175°C in the 18-pin SOIC plastic package and -55°C to 200°C in the 20-pin ceramic CERDIP package. It provides a complete, integrated, cost-effective solution for high temperature avionics applications implementing the CAN 2.0B specification. It can be configured to comply with both the ARINC 825 (General Standardization of CAN Bus Protocol for Airborne Use) and CANaerospace standards. The HI-3110H is capable of transmitting and receiving standard data frames, extended data frames and remote frames. The internal transceiver allows direct connection to the CAN bus without using external components and coupled with the host Serial Peripheral Interface (SPI), results in minimal board space.

The HI-3110H provides the optimum solution for high temperature avionics applications where minimum host (MCU) overhead is required, filtering unwanted messages using a maskable identifier filter and storing up to 8 messages in the receive FIFO. A flexible interrupt scheme allows real time servicing of the FIFO by the host, if required. Transmissions are handled using an 8 message transmit FIFO. A Transmit Enable pin can be used by the host to initiate a transmission. The device also provides monitor or listen-only mode, low power sleep mode, loopback mode for self-test and a re-transmission disable capability (necessary to implement TTCAN protocol).

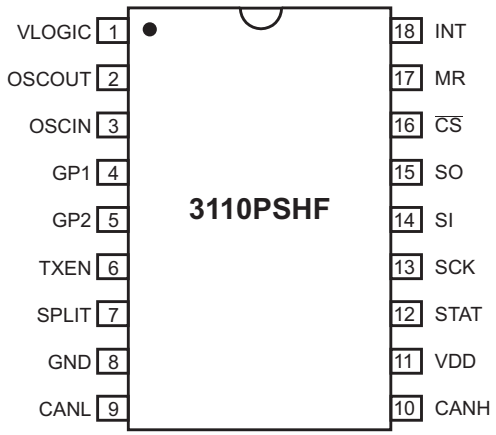
The HI-3111H is a digital only version of the HI-3110H (no transceiver). This version provides a “protocol only” solution for customers who wish to use an external transceiver and may be used in situations where the customer requires galvanic isolation between the bus and digital protocol logic. The HI-3112H provides an option of a CLKOUT pin instead of a SPLIT pin, which may be used as the main system clock or as a clock input for other devices in the system.

The HI-3110H design has been independently validated by C&S group, GmbH, an ISO/IEC 17025 accredited test house. A copy of the test report is available from Holt on request.

FEATURES

- Implements CAN version 2.0B with programmable bit rate up to 1Mbit/sec. ISO 11898-5 compliant.
- Extended Temperature Ranges of -55°C to 175°C (plastic SOIC-18 package) and -55°C to 200°C (ceramic CERDIP-20 package).
- Configurable to support ARINC 825 and CANaerospace Standards.
- Filtering on ID and first two data bytes for both Standard and Extended Identifiers.
- Permanent dominant timeout protection.
- Serial Peripheral Interface (SPI) (20MHz).
- Standard, Extended and Remote frames supported.
- 8 maskable identifier filters.
- Loopback mode for self-test.
- Monitor (Listen-only) and Low Power Sleep Modes with automatic wake-up possible.
- 8-message Transmit and Receive FIFOs.
- Internal 16-bit free running counter for time tagging of transmitted or received messages.
- Short Circuit Protection of -58V to + 58V on CAN_H, CAN_L and SPLIT pins (ISO 11898-5).
- Re-transmission disable capability.
- Transmit Enable pin.

PIN CONFIGURATION (Top View)



18-Pin Plastic SOIC - WB Package

BLOCK DIAGRAM

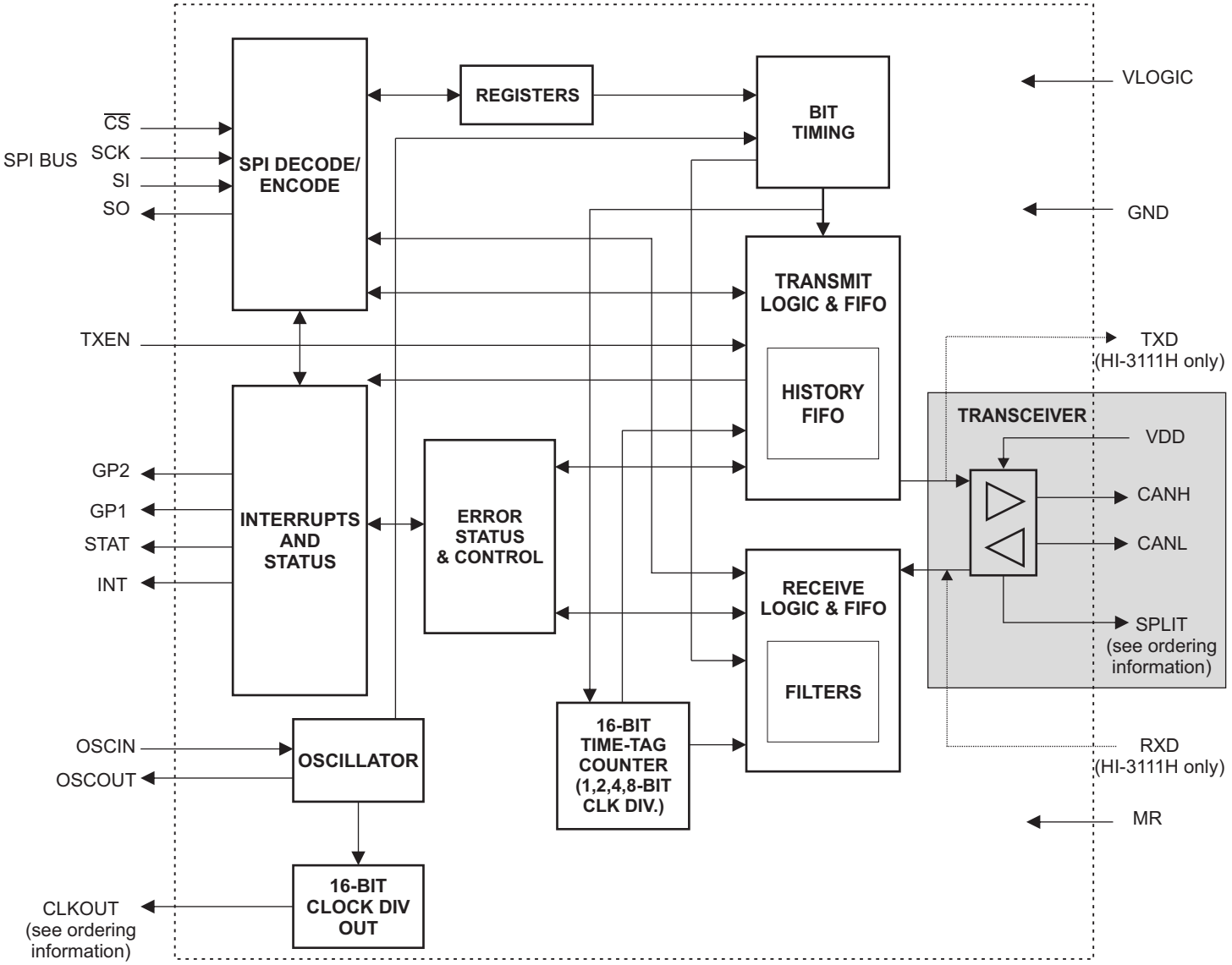


Figure 1. HI-3110H Block Diagram

PRIMARY FUNCTIONS OF HI-3110H LOGIC BLOCKS

SPI PROTOCOL BLOCK
Handles data transfers between the host and the chip

REGISTERS BLOCK
Stores configuration data

BIT TIMING BLOCK
Sets the data strobe and bit period

TRANSMIT BLOCK
Manages transmission protocol
8 message FIFO
Confirmation and time stamp of each message sent is available in the History FIFO

RECEIVER BLOCK
Manages reception protocol
8 message FIFO with optional filters
Forwards message data and optional time stamp to the host

ERROR BLOCK
Detects and records errors for protocol management

STATUS AND INTERRUPT
Provides hardware and software options for managing communications

OSCILLATOR
Configuration chooses either the crystal oscillator or and external clock

TRANSCEIVER
Analog interface connects directly to the CAN bus

PIN DESCRIPTIONS

SCK	INPUT	SPI Clock. Data is shifted into or out of the SPI interface using SCK	50K ohm pull-down
$\overline{CS}$	INPUT	Chip Select. Data is shifted into SI and out of SO when $\overline{CS}$ is low.	50K ohm pull-up
SI	INPUT	SPI interface serial data input	50K ohm pull-down
SO	OUTPUT	SPI interface serial data output	
INT	OUTPUT	Active high. Programmable interrupt output	
STAT	OUTPUT	Active high. Programmable status output.	
TXEN	INPUT	Active high. Transmit Enable pin. When the TXEN pin is asserted, any message in the Transmit FIFO will be automatically loaded to the Transmit buffer and sent if the bus is available. This pin is logically ORed with the TXEN and TX1M bits in the CTRL1 register. When the TXEN pin is reset, messages loaded to the FIFO will not be sent until TXEN or TX1M bits are set in the CTRL1 register.	100K ohm pull-down
OSCIN	INPUT	Crystal input. A parallel resonant crystal can be connected between OSCIN and OSCOUT. If an external clock is used, it should be connected to the OSCIN pin and the OSCOUT pin should be left floating. The internal oscillator should be shut off by setting the OSCOFF bit in the CTRL1 register.	
OSCOUT	OUTPUT	Crystal output. If an external clock is used, this pin should be left floating and disabled by setting the OSCOFF bit in the CTRL1 register.	
GP1	OUTPUT	General purpose pin 1, which can be programmed to reflect the values of interrupt and status flag bits.	
GP2	OUTPUT	General purpose pin 2, which can be programmed to reflect the values of interrupt and status flag bits.	
CLKOUT	OUTPUT	Clock output pin with programmable frequency divider.	
SPLIT	OUTPUT	VDD/2 output bias (Powered off in Sleep Mode and when the common mode bias is greater than 25V).	
CANH	BUS I/O	CAN bus line high.	
CANL	BUS I/O	CAN bus line low.	
TXD	OUTPUT	Transmit Data Out. Connect to TXD input pin on CAN transceiver (e.g. HI-3000H).	HI-3111H only
RXD	INPUT	Receive Data In. Connect to RXD output pin on CAN transceiver (e.g. HI-3000H).	HI-3111H only 5V Logic Tolerant
MR	INPUT	Active High. Device Master Reset input pin. Asserting this pin resets all registers and memory buffers to their default state at start-up.	50K ohm pull-down
VDD	POWER	5V supply voltage input.	
VLOGIC	POWER	3.3V supply voltage input. This supply is used to drive the host digital logic I/O. It can either be connected directly to VDD (+5V) or a +3.3V supply.	
GND	POWER	Supply voltage ground.	

FUNCTIONAL OVERVIEW

The HI-3110 is the first single chip product to integrate both the CAN (Controller Area Network) protocol and analog interface transceiver on a single IC. The protocol conforms to CAN version 2.0B and is compliant with ISO 11898-1:2003(E) specification. The transceiver is compliant with ISO 11898-5 specification.

Configuration options include an internal Loopback mode that does not disturb the bus, a Monitor only mode, and a Sleep mode that includes an option to either wake up automatically when data is present on the bus, or by host command. The following sections describe some of the key features.

SPI and REGISTERS

To minimize the footprint, a 20 MHz standard four wire SPI (Serial Peripheral Interface) is provided to manage the flow of data between the host microcontroller and the HI-3110. Complete messages are loaded and retrieved with single SPI op codes. On the receive side, SPI op code options may be used to retrieve the whole message or just the data. An option to include a time tag or no time tag may also be specified. On the transmit side, each message can be assigned an identifier which allows monitoring of the Transmit History FIFO to confirm the successful completion of a transmission along with the time stamp. In addition the transmitter logic automatically assembles the message frame based on the data presented.

BIT TIMING

Bit timing is controlled with standard CAN options. These include control of the Resychronization Jump Width (SJW), Prop delay Phase Seg 1 (TSeg1), Phase Seg 2 (TSeg2), the number of samples, and the derivation of Tq from the system clock using a prescaler. The maximum bit rate is 1 MBit/sec. Upon reset, the chip automatically enters Initialization mode which allows programming of the Bit Timing before entering Normal mode.

TRANSMITTER

The transmitter state machine automatically handles all CAN 2.0B protocol requirements. Messages for transmission are first loaded into a FIFO and transmission may start upon availability of data in the FIFO. Assertion of the TXEN pin or configuration bits in Control Register 1 allow either continuous transmission until the FIFO is empty or only one message from the FIFO at a time. One shot (no retry) transmission may also be enabled by setting the OSM and TX1M bits. SPI op codes are provided to clear the Transmit FIFO and to abort transmission.

RECEIVER

The receiver state machine automatically handles all CAN 2.0B protocol requirements. The receiver supports eight sets of filters and masks and each allows filtering of a full CAN ID (extended or not) and two bytes of data. Even when filtering is enabled, message data is always accessible as received via the Temporary Receive Buffer, and retrievable by SPI op codes 0x42 and 0x44.

If the Filter/Mask option is set (FILTON bit in Control Register 1), only messages that match one of the 8 stored data patterns are passed into the FIFO. Note that the Mask option allows certain bits of the programmed filter bits to be “don't care.” If the Filter/Mask option is not set, then all valid messages are passed to the FIFO. When the FIFO is full (8 completed messages received), the next received message will overwrite the latest message stored in the FIFO.

ERROR CONTROL

Errors are detected per ISO 11898-1:2003(E) and detections are counted and used by the protocol state machines. Active, Passive, and Bus Off conditions are managed per the CAN standard. A configuration bit is provided to allow automatic recovery from Bus Off.

STATUS and INTERRUPTS

The Message Status Register, MESSTAT, provides information about the current state of the receiver and transmitter operation. In addition, the Interrupt Flag Register, INTF, monitors 8 operational conditions, any or all of which may be directed to the INT pin by enabling bits in the Interrupt Enable Register, INTE. Similarly, the Status Flag Register, STATF, bits reflect the status of selected FIFO and Error properties. Any or all of these conditions may be directed to the STAT pin by setting the enable bits in the Status Flag Enable Register, STATFE.

To provide additional hardwired flag options, the GP1 and GP2 pins may also be programmed to reflect any of the Interrupt or Status Flag bits.

OSCILLATOR and TIME TAG

A configuration bit allows a choice for the source of the system clock. Either the on-board crystal oscillator may be selected or an external clock may be provided at the OSCIN pin.

On product versions with the CLKOUT pin, a programmable division of the system clock is provided. The clock source for the 16 bit Time Tag Counter is derived from a separate programmable division of the system clock. SPI op codes provide for reading and resetting the Time Tag Counter.

TRANSCEIVER

The HI-3110 contains an integrated transceiver operating from 5V and the line driver is capable of maintaining a detectable signal for bus lengths well in excess of recommended CAN 2.0B standards. The digital logic and IO can be powered from 3.3V or 5V.

PROTECTION FEATURES

The BUS and SPLIT pins are protected against ESD to over 4KV (HBM) and from shorts between -58V to +58V continuous, as specified in ISO 11898-5.

In addition, a Permanent Dominant Timeout protection is implemented by means of an independent counter monitoring the dominant transmission state and automatically shutting off the transmission if it exceeds typically 2ms.

MODES OF OPERATION

The HI-3110 supports five modes of operation, namely, Initialization Mode, Normal Mode, Loopback Mode, Monitor Mode and Sleep Mode.

INITIALIZATION MODE

Initialization mode is used to configure the device before normal operation. **Bit timing registers and acceptance filters and masks can only be modified in this mode.** Initialization mode is the default mode following RESET and can also be activated by programming the MODE<2:0> bits to <1xx> in the CTRL0 register. Switching to Initialization mode resets the receiver and transmitter. During initialization mode, the error counters are held reset.

NORMAL MODE

Normal mode is the standard operating mode of the HI-3110. In this mode, the HI-3110 can transmit, receive and acknowledge messages from the CAN bus, handling all aspects of the CAN protocol. Normal mode is activated by programming the MODE<2:0> bits to <000> in the CTRL0 register.

LOOPBACK MODE

Loopback mode is used for self-test. The transceiver digital input is fed back to the receiver without being transmitted to the bus. Messages are transmitted from the transmit FIFO in the usual way and received by the receive FIFO as if they were received from a remote node on the bus.

Acceptance filters can be set up to accept or reject specific messages into the FIFO and all interrupt flags are set as required in the usual way. While in this mode, any bus activity is ignored. Loopback is activated by programming the MODE<2:0> bits to <001> in the CTRL0 register.

MONITOR MODE

Monitor mode (also known as listen-only or silent mode) allows the HI-3110 to monitor all bus activity without disturbing the bus. No messages or dominant bits (such as ACK or active error frame bits) are transmitted to the bus while in this mode. Also, the error counters are reset and deactivated. Messages from the bus are received in the same way as Normal Mode and messages that are not acknowledged by another node on the bus are ignored i.e. any frame containing an error will be ignored. Acceptance filters can be set up to reject or accept specific messages into the FIFO and all interrupt flags are set as required in the usual way. Monitor mode is activated by programming the MODE<2:0> bits to <010> in the CTRL0 register.

SLEEP MODE

The HI-3110 can be placed in a low power sleep mode if there is no bus activity and the transmit FIFO is empty. In this mode, the internal oscillator and all analog circuitry (transceiver) are off, drawing typically less than 20μA. Note that the SPI bus is active during sleep mode, so it is possible for the host to communicate with the HI-3110 while it is asleep (e.g. load transmit FIFOs). Sleep mode is exited by selecting an alternative mode of operation, or automatic wake up following bus activity can be enabled by setting the WAKEUP bit in the CTRL0 register - in this case a low power receiver monitors the bus for a detectable dominant bit.. The device will wake up in Monitor Mode. Note that it will take a finite time for the oscillator and analog circuitry to come back on line. Since the internal oscillator takes a finite time to wake up, the message which caused the wake-up may not be stored.

Sleep mode is activated by programming the MODE<2:0> bits to <011> in the CTRL0 register. However, the actual mode change will only occur whenever the CAN bus is quiet. If the chip is transmitting, the mode change is delayed until the transmission is complete. If there is bus activity, the mode change is delayed until the receiver protocol control detects an inter-message gap.

CAN PROTOCOL OVERVIEW

The HI-3110 supports Standard, Extended and Remote Frames, as defined in the CAN specification IS0 11898-1:2003(E) (also known as CAN 2.0B).

BIT ENCODING

CAN frames are encoded according to the Non-Return-To-Zero (NRZ) method with bit stuffing. NRZ means that the

generated bit level is constant during the total bit time and consecutive bits do not return to a neutral or rest condition. This means that a bit stream of “1s” or “0”s appears continuous on the bus. A logic “0” is called a dominant bit and a logic “1” is called a recessive bit.

Bit stuffing is used to ensure frequent enough transitions occur to achieve synchronization. Every time a transmitter detects five consecutive bits of the same polarity in the bit stream to be transmitted, it inserts a bit of opposite polarity into the actual transmitted bit stream.

This bit stuffing rule applies to the Start-of-Frame field, arbitration field, control field, data field and CRC sequence. The CRC delimiter, ACK field and End-Of-Frame fields are of fixed form and not stuffed (see below for definition of these fields). Furthermore, Error frames and Overload frames are also of fixed form and not stuffed.

An example of how the bits in a stuffed bit stream might look is shown below.

00101011111**O**0000**I**1100000**I**11000

0 = dominant bit, **O** = dominant stuffed bit.
1 = recessive bit, **I** = recessive stuffed bit.

MESSAGE FRAMES

STANDARD DATA FRAME

The standard data frame is shown in figure 2. The frame starts with a Start-of-Frame (SOF) bit. This is a dominant bit that identifies the start of the data frame on the bus.

The SOF is followed by the 12-bit arbitration field. The arbitration field consists of an 11-bit identifer, ID28 - ID18, and the Remote Transmission Request (RTR) bit. The RTR bit is used to distinguish between a data frame (RTR bit dominant, logic 0) and a remote frame (RTR bit recessive, logic 1).

Following the arbitration field is the 6-bit control field. The first bit of the control field is the Identifier Extension flag bit (IDE). This is used to distinguish between standard and extended identifiers and must be dominant (logic 0) for standard data frames. The next bit, r0, is specified by the CAN protocol as a reserved bit for future expansion. This bit must be transmitted dominant, but receivers must be capable of receiving either a dominant or recessive bit. The final 4 bits of the control field make up the data length code (DLC). The binary value of this 4-bit field specifies the number of data bytes in the data payload (0 - 8 bytes). **Note:** All binary combinations greater than or equal to <1 0 0 0> specify 8 bytes of data.

After the control field is the data field, which contains a data payload equal to the number of bytes specified by the DLC (see note above).

The data field is followed by the 16-bit Cyclic Redundancy Check (CRC) field. This is used to check transmission errors by computing a 15-bit CRC sequence from the previous bit stream (SOF, arbitration field, control field and data field, excluding stuff bits). The last bit in the CRC field is the CRC delimiter bit (always recessive).

After the CRC field is the Acknowledge Field (ACK Field). The first bit is the ACK Slot bit. A transmitting node sends a recessive bit (logic 1) during the ACK slot. Any node which receives the message error-free acknowledges the reception by placing a dominant bit (logic 0) in the ACK slot, over-writing the recessive bit of the transmitter. The final bit in the ACK field is a recessive ACK delimiter bit. Therefore, the dominant ACK slot bit is surrounded on each side by a recessive bit.

Each data frame is delimited by an End-Of-Frame field (EOF). The EOF consists of seven recessive bits.

Following the EOF, there is a gap to the next frame called the Interframe Space (IFS) . The IFS consists of two bit fields, Intermission and Bus-Idle. The Intermission consists of three recessive bits, however the following notes apply:
a) detection of a dominant bit on the bus at the third slot is interpreted as a SOF,
b) detection of a dominant bit in either the first or second slots results in generation of an overload frame (see below).

The bus idle period is of arbitrary length and consists of recessive bits. A dominant bit detected during this period is interpreted as a SOF.

EXTENDED DATA FRAME

The extended data frame is shown in figure 3. In this frame format, SOF is followed by a 32-bit arbitration field consisting of a 29-bit identifier, ID28 - ID0. The first 11 most significant bits of the ID are know as the base identifier. This is followed by the Substitute Remote Request (SRR) bit, which is defined as recessive. Following the SRR bit is the IDE bit, which is defined as recessive for extended data frames. Note that the SRR bit is in the same slot as the RTR bit of the standard frame and the IDE bits are also in corresponding slots. This means if standard and extended identifier data frames with identical base identifiers are transmitted simultaneously, the standard identifier data frame will win arbitration (see Bitwise Arbitration section below).

The SRR and IDE bits are followed by the remaining 18 bits of the identifier (extended ID) and the last bit of the arbitration field is the RTR bit. The RTR bit has the same function as in the standard frame format.

Following the arbitration field is the 6-bit control field. The first two bits, r1 and r0, are specified by the CAN protocol as reserved bits for future expansion. Both these bits must be transmitted dominant, but receivers must be able to receive all combinations of dominant or recessive bits. The final 4 bits of the control field is the data length code (DLC). The

binary value of this 4-bit field specifies the number of data bytes in the data payload (0 - 8 bytes). **Note:** All binary combinations greater than or equal to <1 0 0 0> specify 8 bytes of data.

The remaining fields of the extended data frame (Data field, CRC field, acknowledge field, EOF field and IFS field) are constructed in the same way as the standard frame format.

REMOTE FRAME

The remote frame is shown in figure 4. The function of remote frames is to allow a receiver which periodically receives certain types of data to request that data from the transmitting source. The identifier of the remote frame must be identical to the identifier of the requested transmitting node's data frame and the data length code (DLC) should be equal to the DLC of the requested data. **Simultaneous transmission of remote frames with the same identifier and different DLCs will lead to unresolvable collisions on the bus.** For this reason, ARINC 825 strongly discourages the use of remote frames.

The format of a remote frame is identical to the format of the corresponding data frame except the remote frame has no data payload. Remote frames and data frames are distinguished by a recessive RTR bit in the remote frame. This means if a receiver sends a remote frame and the sending node transmits at the same time, the sending node (with a dominant RTR bit) will win arbitration and the requesting node will receive the desired data immediately.

ERROR FRAME

The error frame is shown in figure 5. Any node detecting an error generates an error frame. The error frame consists of two fields, the error flag field and the error delimiter. The type of error flag field depends on the error status of the node, error-active or error-passive (see below). An error-active node generates an active error flag and an error-passive node generates a passive error flag.

Active Error Flag: An active error flag consists of 6 consecutive dominant bits. This condition violates the rule of bit-stuffing and causes all other nodes on the bus to generate error flags, known as echo error flags. Therefore, the error flag field will consist of the superposition of different error flags sent by individual nodes, resulting in a minimum of 6 and maximum of 12 consecutive dominant bits. The error flag field is followed by the error delimiter, consisting of 8 recessive bits.

Passive Error Flag: A passive error flag consists of 6 recessive bits. This is followed by the 8 recessive bits of the error delimiter. Therefore, an error frame sent by an error-passive node consists of 14 consecutive recessive bits. Since this will not disturb the bus, a transmitting node will continue to transmit unless it detects the error itself, or another error-active node detects the error.

Notes: If the passive error flag is generated by a receiver, it cannot prevail over any other activity on the bus. Therefore, it must wait for 6 consecutive bits of equal polarity before completing the error flag. If the passive error flag is generated by a transmitter, the bit stuffing rule is violated and it will cause other nodes to generate error flags. Two exceptions to this rule are

a) the passive error flag starts during arbitration and another node prevails and begins transmitting, and

b) the error flag starts less than 6 bits before the end of the CRC sequence and the last bits of the CRC sequence all happen to be recessive.

OVERLOAD FRAME

The overload frame is shown in figure 6. It has the same format as the active error frame, consisting of an overload flag field and an overload delimiter. The overload flag consists of 6 consecutive dominant bits. This condition violates the rule of bit-stuffing and causes all other nodes on the bus to generate echo flags, similar to the active error flag echos. Therefore, the overload flag field will consist of the superposition of different overload flags sent by individual nodes, resulting in a minimum of 6 and maximum of 12 consecutive dominant bits. The overload flag is followed by the overload delimiter, consisting of 8 recessive bits.

An overload frame, unlike an error frame, can only be generated during the interframe space. There are two types of overload frame:

1) Reactive Overload Frame, resulting from

a) detection of a dominant bit during the first or second bit of intermission,

b) detection of a dominant bit at the last (seventh) bit of EOF in received frames, or

c) detection of a dominant bit at the last (eighth) bit of an error delimiter or overload delimiter.

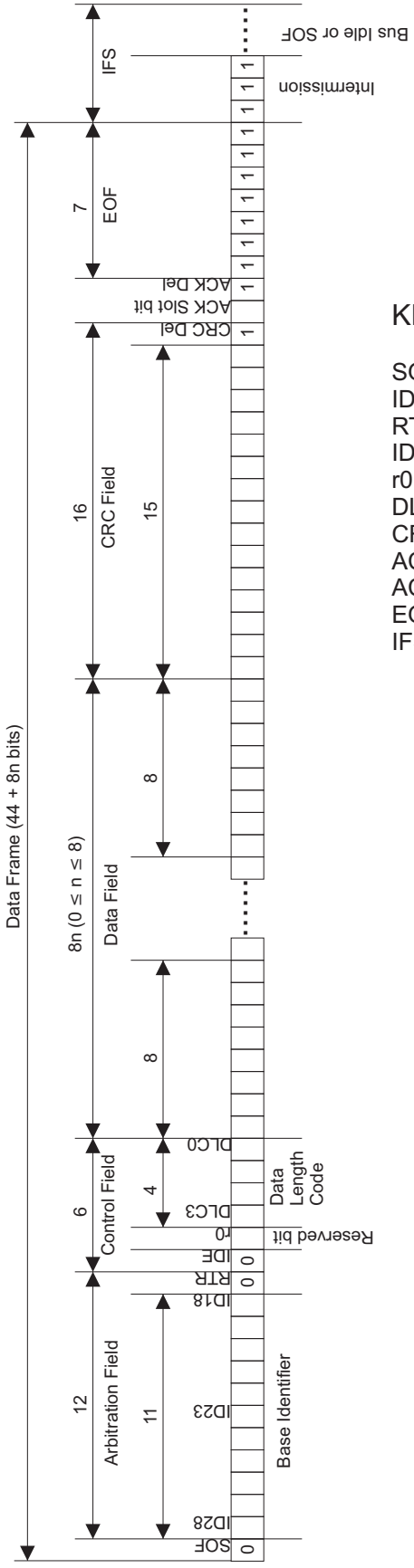
The reactive overload frame is started one bit after detecting any of the above dominant bit conditions.

2) Requested Overload Frame. A node which is unable to begin reception of the next message due to internal conditions may request a delay by transmitting a maximum of two consecutive overload frames. The requested overload frame must be started at the first bit of an expected intermission.

Note 1): The HI-3110 will never initiate an overload frame unless reacting to one of the conditions in case 1) above.

Note 2): Initiation of overload frames is prohibited by ARINC 825 since they increase the network loading.

Standard Data Frame

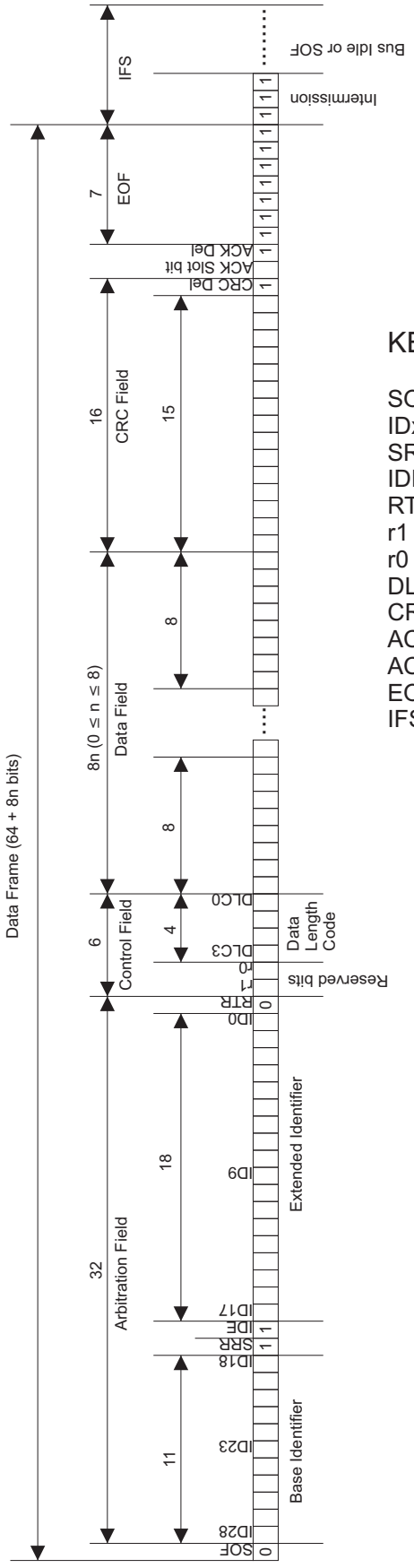


KEY.

SOF	Start-Of-Frame
IDxx	Identifier bit xx
RTR	Remote Transmission Request bit
IDE	Identifier Extension Flag
r0	Reserved bit 0
DLCx	Data Length Code bit x
CRC Del	Cyclic Redundancy Check Delimiter
ACK Slot bit	Acknowledge Slot bit
ACK Del	Acknowledge Delimiter
EOF	End-Of-Frame
IFS	Interframe space

Figure 2. Standard Frame Format.

Extended Data Frame



KEY.

SOF	Start-Of-Frame
IDxx	Identifier bit xx
SRR	Substitute Remote Request bit
IDE	Identifier Extension Flag
RTR	Remote Transmission Request bit
r1	Reserved bit 1
r0	Reserved bit 0
DLCx	Data Length Code bit x
CRC Del	Cyclic Redundancy Check Delimiter
ACK Slot bit	Acknowledge Slot bit
ACK Del	Acknowledge Delimiter
EOF	End-Of-Frame
IFS	Interframe space

Figure 3. Extended Frame Format.

Remote Frame

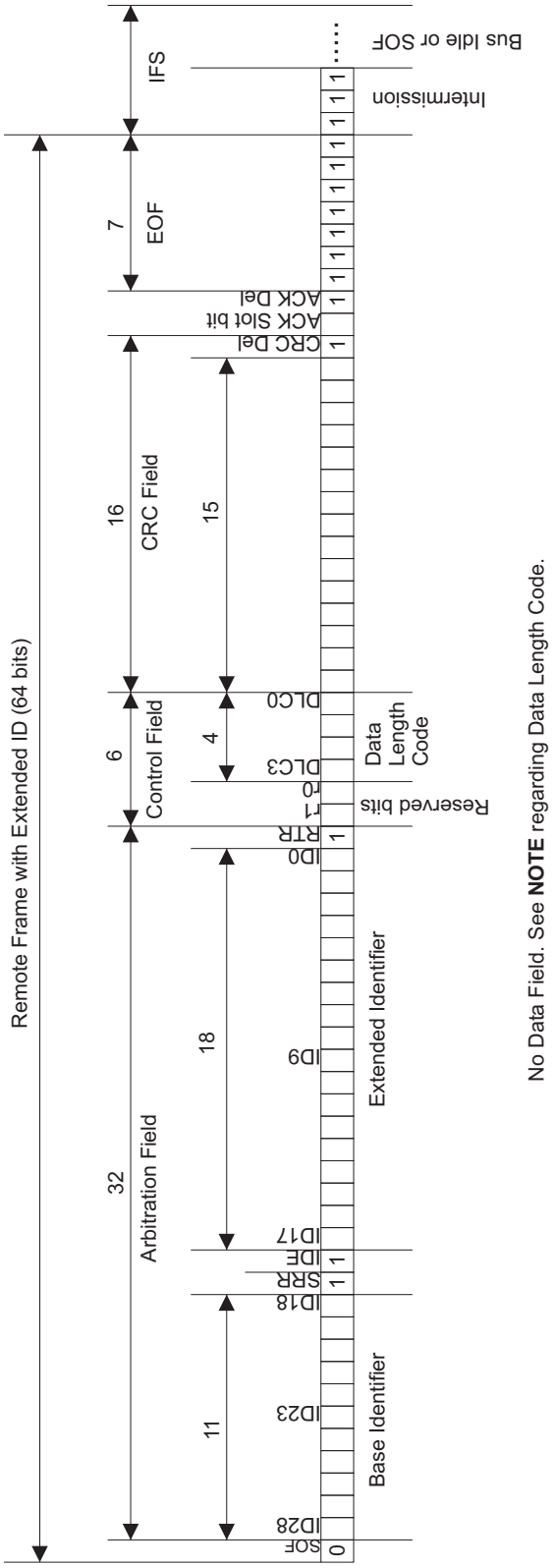


Figure 4. Remote Frame Format (Extended Identifier).

Error Frame

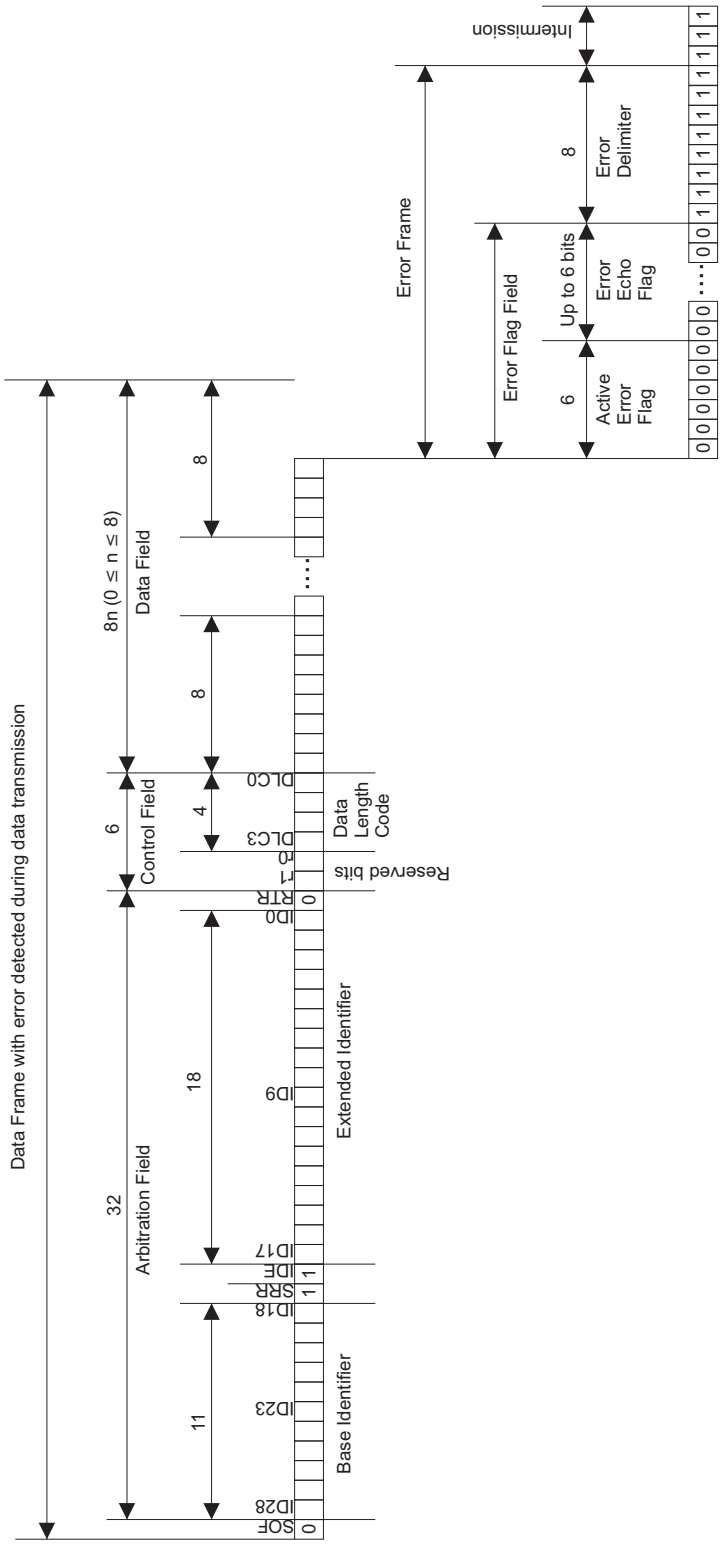


Figure 5. Error Frame Format.

Overload Frame

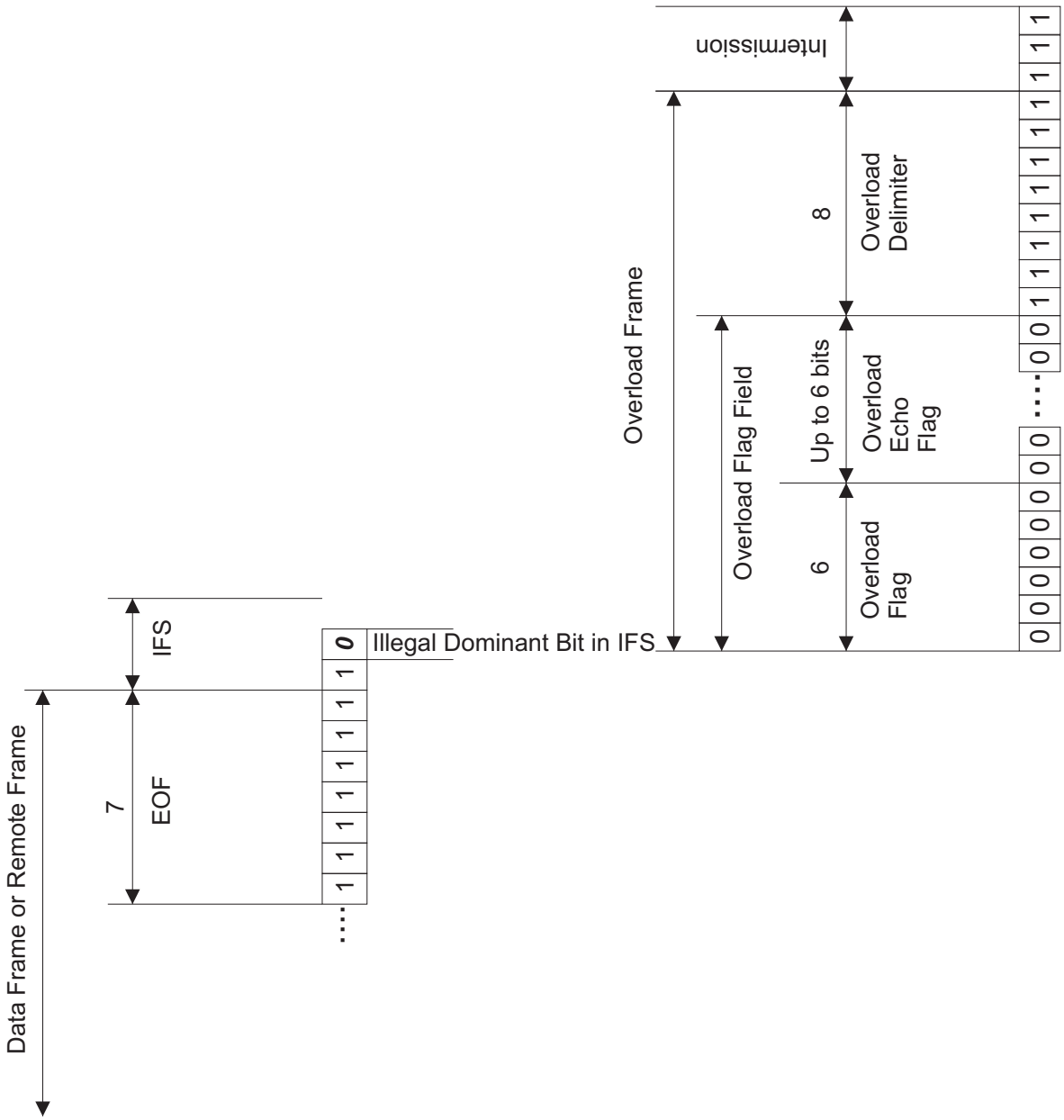


Figure 6. Overload Frame Format

BIT WISE ARBITRATION

The CAN standard resolves data contention on the bus using a scheme called Carrier Sense Multiple Access/Collision Detection-Carrier Resolution (CSMA/CD-CR).

Carrier Sense: Each node waits for a period without bus activity (bus idle state) before attempting transmission.
Multiple Access: Every node on the bus has equal access to the bus for transmitting.
Collision Detection: Collisions occur if two nodes attempt to transmit at the same time.
Collision Resolution: Collisions are resolved by bitwise arbitration. Highest priority messages (lowest binary identifiers) are sent first without delay and lower-priority messages are automatically re-transmitted later. A dominant bit (logic 0) has priority over a recessive bit (logic 1).

Bitwise arbitration works by comparing each node's transmitted data bit by bit. All nodes are synchronized by adjusting individual bit times as a function of bit time quanta (see section on bit timing). Synchronization takes place on recessive to dominant edges. A Hard Synchronization at the start of each frame and subsequent re-synchronizations during a message frame ensures corresponding bits match in time during a given transmission cycle. When a node transmits a recessive bit on the bus, but detects a dominant bit on it's receiver, it realizes arbitration is lost and it immediately ceases transmission and becomes a receiver. It will then wait for the next bus idle state and attempt to re-transmit. Eventually, lower priority messages will gain access to the bus. Figure 7 shows an example of how this works for a frame with a standard identifier.

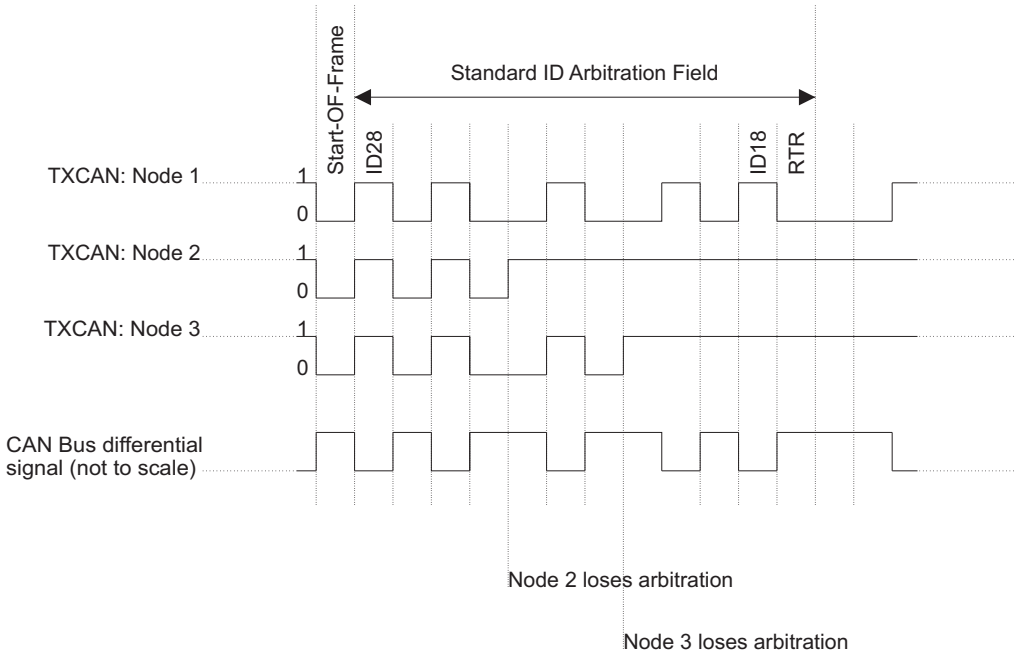


Figure 7. Bitwise Arbitration.

BIT TIMING

The CAN protocol supports a broad range of bit rates, from a few kHz up to 1MHz (**Note:** the minimum bit rate of the HI-3110 is limited to 40kHz by the permanent dominant timeout protection of the transceiver). Every node on the network has it's own clock generator (typically a quartz oscillator), however the bit rate must obviously be the same for every node on the bus. Therefore, each CAN node must be configurable to generate the nominal bit rate as a function of it's own oscillator frequency, f_{OSC}. This is done by generating a time quanta (TQ) clock, whose period t_{TQ} is related to the oscillator frequency by a Baud Rate Prescaler value, BRP as follows:

$$t_{TQ} = 2 \bullet BRP/f_{OSC} \qquad (1)$$

The TQ clock is used to construct the bit time in terms of time quanta, such that one time quantum, Tq, equals one TQ clock period, t_{TQ}, as shown in figure 8 below.
The CAN system nominal bit rate (BR) is defined in terms of the nominal bit time, t_b, as

$$BR = 1/t_b \qquad (2)$$

Therefore, the nominal bit rate is related to the TQ clock period by the following relationship

$$BR = 1/(t_{TQ} \text{ x (number of time quanta per bit))} \qquad (3)$$

The CAN standard divides the bit time into four segments, namely, synchronization segment (Sync Seg), propagation time segment (Prop Seg), phase buffer segment 1 (Phase Seg1) and phase buffer segment 2 (Phase Seg2). This is illustrated in figure 8. The HI-3110 fixes the Sync Seg at 1Tq. Prop Seg and Phase Seg1 are treated as one time segment, TSeg1, which is programmable from 2Tq to 16Tq. Phase Seg2 is a second time segment, TSeg2, which is programmable from 2Tq to 8Tq (**Note:** Not all combinations are valid, see below for examples).

Synchronization Segment (Sync Seg)

The Sync Seg is the first segment of the bit time and is used to synchronize the various nodes on the bus. A bit edge is expected to occur within the Sync Seg.

Propagation Time Segment (Prog Seg)

The Prog Seg is used to compensate for physical delays on the bus, which include signal propagation delay time on the bus and internal node delay times. For two nodes A and B communicating on the bus, Prog Seg must be greater than or equal to the sum of both nodes internal delays plus twice the bus line propagation delay between the two nodes.

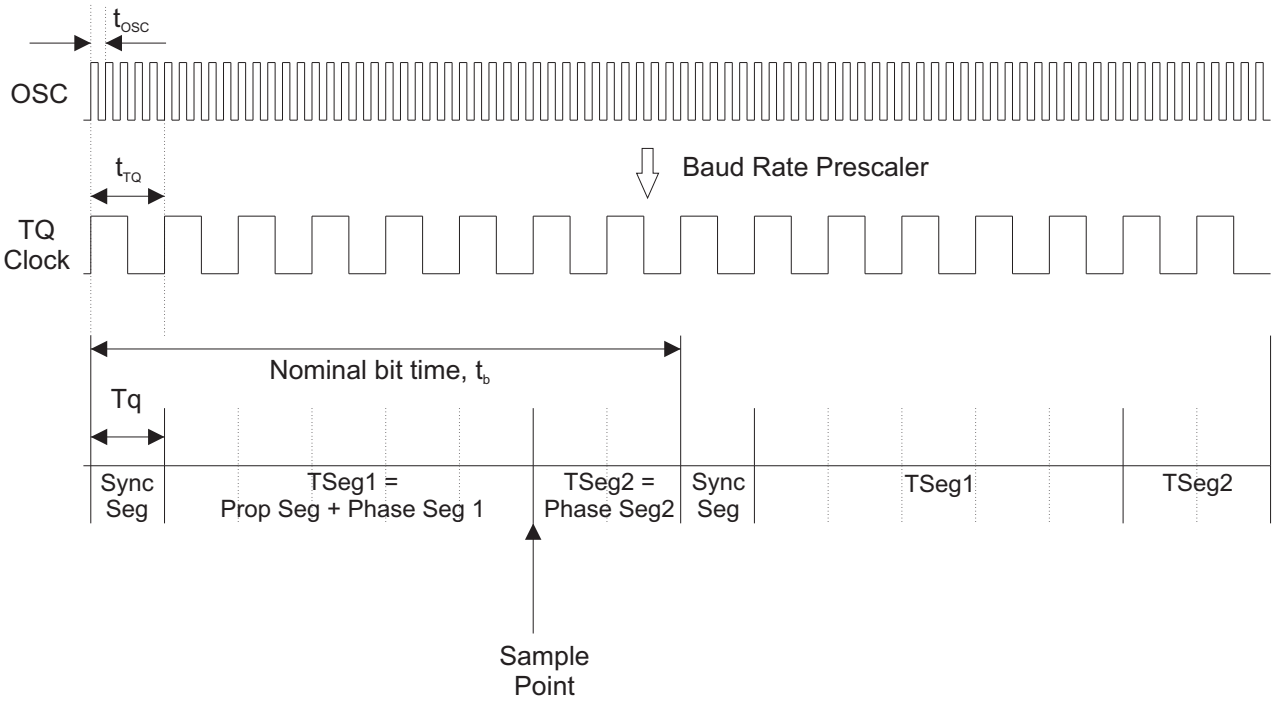


Figure 8. CAN Bit Time

Phase Buffer Segment 1 and Phase Buffer Segment 2 (Phase Seg1 and Phase Seg2)

The phase buffer segments are used to compensate for phase errors on the bus. Phase Seg1 can be lengthened or Phase Seg2 can be shortened duringthe re-synchronization bit period automatically by the HI-3110 so that the bit time can be adjusted to account for phase errors. The upper limit by which the lengthening (or shortening) can occur is set by the **re-synchronization jump width (SJW),** explained in more detail below.

Sample Point

The sample point is the point in the bit time at which the bit logic level is interpreted. It is located at the end of Phase Seg1. The HI-3110 also allows three sample points to be taken. In this case, two other sample points are taken prior to the end of Phase Seg1 (at one-half TQ intervals) and the value of the bit is determined by a majority decision. Three sample points are typically only used at low bit rates. **Note:** ARINC 825 states that there shall be only one sample per bit, taken at the end of Phase Seg1.

The time required for the logic to determine the bit level of a sampled bit is known as the **information processing time (IPT)**. According to the standard, IPT can be up to 2Tq. Since Phase Seg2 occurs after the sample point, Phase Seg2 must be greater than or equal to the worst case IPT (2Tq).

Phase Errors (e)

If a bit edge occurs within the Sync Seg as expected, there is no phase error (e = 0). However, if an edge occurs outside Sync Seg, a phase error is deemed to have occurred. If the edge occurs after Sync Seg (edge occurs “late”), the phase error is positive (e > 0), whereas if the edge occurs before Sync Seg (edge occurs “early”), the phase error is negative (e < 0).

Synchronization

Synchronization is carried out only on recessive-to-dominant bit edges and is used to ensure the bit times of all nodes on the bus are synchronized. This is necessary for arbitration and message acknowledgment to function properly. Only one synchronization can occur per bit time.

Hard synchronization forces the bit edge to lie within the Sync Seg, regardless of the phase error. Hard synchronization only occurs on reception of the start of a frame.

Re-synchronization results in the shortening or lengthening of the bit time such that the position of the sample point is shifted with respect to the edge causing the re-synchronization. **For e > 0**, Phase Seg 1 is lengthened by the magnitude of the phase error, up to a maximum of SJW. **For e < 0**, Phase Seg 2 is shortened by the magnitude of the phase error, up to a maximum of SJW.

Examples

1) CAN bit rate (BR) = 125kHz, f_{osc} = 12MHz.
Assume sample point (at end of TSeg1) will occur at 75% of bit time. Hence, for Sync Seg = 1Tq, TSeg1 = 5 Tq and TSeg2 = 2Tq. Therefore, total bit time will be 8Tq. Chose SJW = 1Tq.
For 125kHz, the bit time needs to be 1/125kHz = 8µs. Hence, 1Tq = 1µs. Using equation (1) => BRP = 6.

2) CAN bit rate (BR) = 1MHz, f_{osc} = 32MHz.
Assume sample point (at end of TSeg1) will occur at 75% of bit time. For Sync Seg = 1Tq, then TSeg1 = 11Tq and TSeg2 = 4Tq. Therefore, total bit time will be 16Tq. Chose SJW = 1Tq.
For 1MHz, the bit time needs to be 1/1MHz = 1µs. Hence, 1Tq = 62.5ns. Using equation (1) => BRP = 1.

Note: Choosing the sample point at 75% of the bit time is a requirement of ARINC 825. The oscillator frequency must be chosen such that a valid value of BRP (integer) can generate the TQ clock (e.g. in example 2 above, using a lower oscillator frequency than 32MHz results in BRP < 1).

REGISTERS

This section describes the HI-3110 registers. All register bits are active high. Unless otherwise indicated, all registers are reset in software to the logic zero condition after Master Reset. For all registers, bit 7 is the most significant:

REGISTER	R/W	DESCRIPTION	SPI WRITE OP-CODE	SPI READ OP-CODE
CTRL0	R/W	Control Register 0	0x14	0xD2
CTRL1	R/W	Control Register 1	0x16	0xD4
BTR0	R/W	Bit Timing Register 0	0x18	0xD6
BTR1	R/W	Bit Timing Register 1	0x1A	0xD8
TEC	R/W	Transmit Error Counter Register	0x26	0xEC
REC	R/W	Receive Error Counter Register	0x24	0xEA
MESSTAT	R	Message Status Register	N/A	0xDA
ERR	R	Error Register	N/A	0xDC
INTF	R	Interrupt Flag Register	N/A	0xDE
INTE	R/W	Interrupt Enable Register	0x1C	OxE4
STATF	R	Status Flag Register	N/A	0xE2
STATFE	R/W	Status Flag Enable Register	0x1E	0xE6
GPINE	R/W	General Purpose Pins Enable Register	0x22	0xE8
TIMERUB	R	Free-Running Timer Upper Byte Register	N/A	0xFA*
TIMERLB	R	Free-Running Timer Lower Byte Register	N/A	0xFA*

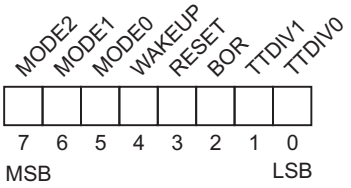
Note: Free-running counter registers, TIMERUB:TIMERLB are read with a single SPI Op-code (0xFA) as a 16-bit value in two SPI data bytes.

Power-On-Reset

Following power-on, the HI-3110 will automatically perform a Master Reset and return all registers to the default state. Following reset, the device will default to Initialization Mode to allow programming of Control and Bit Timing Registers (see following sections).

CONTROL REGISTER 0: CTRL0

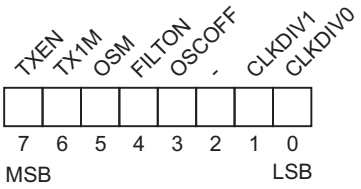
(Write, SPI Op-code 0x14)
(Read, SPI Op-code 0xD2)



Bit	Name	R/W	Default	Description
7-5	MODE2:0	R/W	1,0,0	Mode select bits <2:0>. These bits select the mode of operation as follows. 000: Normal Mode. Normal CAN operation. 001: Loopback Mode. The transceiver digital input is fed back to the receiver without disturbing the bus. This mode can be used for test purposes, allowing the HI-3110 to receive its own messages. 010: Monitor Mode. The HI-3110 can be set up to monitor bus activity without transmitting to the bus (no ACK bits or error frames are sent in this mode). Receive filters can be programmed in Initialization Mode to buffer selected messages. 011: Sleep Mode. The HI-3110 can be placed in a low power sleep mode if there is no bus activity and the transmit FIFO is empty. Sleep mode is exited by selecting an alternative mode of operation, or automatic wake up following bus activity can be enabled by setting the WAKEUP bit. The device will wake up in Monitor mode. 1xx: Initialization Mode. The device **must** be in this mode for bit timing and filter set-up. This is the default following reset. The host exits initialization mode by selecting an alternative mode of operation.
4	WAKEUP	R/W	0	Wake-Up Enable. When this bit is set, the HI-3110 will automatically wake up from Sleep Mode to Monitor Mode when it detects activity on the bus. 1 = Automatic wake-up enabled. When the device wakes up from Sleep Mode, the WAKEUP bit will be set in the Interrupt Flag Register, INTF. A hardware interrupt can be generated at the INT pin by setting the WAKEUPIE bit in the Interrupt Enable Register. 0 = Automatic wake-up not enabled. In this case, wake-up from Sleep Mode is initiated by the host by selecting another mode of operation. When WAKEUP = 0, all bus activity is ignored.
3	RESET	R/W	0	Setting this bit causes HI-3110 reset to occur. Following reset, the CTRL0 register should be written with < 1 x x x 0 x x x >. This will clear the RESET bit and also avoid unpredictable behavior by ensuring the part is programmed to Initialization Mode, ready for set-up. A reset may also be performed by setting the MR pin or issuing the “MR” SPI command, 0x56. 1 = Master Reset (same as MR pin = 1). 0 = Normal Operation (same as MR pin = 0).
2	BOR	R/W	0	Bus-off Reset. When this bit is set, automatic bus-off recovery is initiated following 128 occurrences of 11 consecutive recessive bits on the bus. The HI-3110 will become error-active with both its error counters set to zero and resume operation in Normal Mode. 1 = Automatic bus-off recovery. 0 = The host is responsible for bus-off recovery (default).
1-0	TDIV1:0	R/W	0,0	Time Tag Clock Division Bits <1:0>. See TIMERUB and TIMERLB register descriptions. 00 = No division (counts every bit clock). 01 = Divide by 2 (counts every 2 bit clocks). 10 = Divide by 4 (counts every 4 bit clocks). 11 = Divide by 8 (counts every 8 bit clocks).

CONTROL REGISTER 1: CTRL1

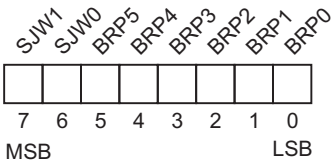
(Write, SPI Op-code 0x16)
(Read, SPI Op-code 0xD4)



Bit	Name	R/W	Default	Description
7	TXEN	R/W	0	Transmission Enable. This bit is logically ORed with the TXEN pin. When this bit is asserted, each message in the FIFO will be sequentially loaded to the transmit buffer and sent if the bus is available. If this bit is not set, a transmission can be enabled by either the TXEN pin or the TX1M bit. If the TXEN pin is pulled low during a transmission, the current message being transmitted will be completed. Any additional messages in the FIFO will not be transmitted. 1 = Enable transmission and send any messages in FIFO (until empty if TXEN is held set). 0 = Wait for transmission enable or TX1M bit set before sending next message in FIFO.
6	TX1M	R/W	0	Enable transmission of only next message. This bit is applicable only if TXEN = 0. It is reset automatically upon completion of a successful transmission or by initiation of transmission if the OSM bit is set. 1 = Enable transmission of only next message in FIFO when TXEN = 0. 0 = Wait for transmission enable or TX1M bit set before sending next message in FIFO.
5	OSM	R/W	0	One-Shot Mode Enable. OSM is intended to be used ONLY with the TX1M bit. If OSM is enabled and TX1M is set, the controller transmits only once and does not attempt re-transmission upon loss of arbitration or error. This feature is necessary to support the implementation of fixed time slots in the Time-Triggered CAN standard (TTCAN). Note**: Un-transmitted messages will remain in the FIFO. If a new message is required on the next transmission cycle, the user must first clear the FIFO with SPI command 0x54 and then reload the new message.** 1 = Enable one-shot mode. 0 = Messages will re-transmit according to CAN protocol.
4	FILTON	R/W	0	Filter on enable. This bit is set to turn on the HI-3110 CAN ID filtering mechanism. The default after reset is FILTON = 0, meaning filtering is turned off and every valid CAN message is accepted into the receive FIFO. **Note: The device must be in initialization mode in order to program the acceptance filters and masks.** 1 = Enable CAN ID filtering. 0 = No CAN ID filtering (every valid message accepted into receive FIFO).
3	OSCOFF	R/W	0	Oscillator off. This bit should be set to a one if an external clock is used. In this case the external clock is connected to the OSCIN pin and OSCOUT should be left floating. 1 = Shuts off external OSCOUT pin. 0 = OSCOUT pin enabled.
2	Not used	R/W	0	
1-0	CLKDIV1:0	R/W	00	External CLKOUT division bits <1:0> 00: Divide by 1. 01: Divide by 2. 10: Divide by 4. 11: Divide by 8.

BIT TIMING REGISTER 0: BTR0

(Write, SPI Op-code 0x18)
(Read, SPI Op-code 0xD6)

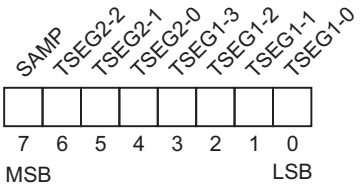


BTR0 defines the value of the Re-synchronization Jump Width (SJW) and the Baud Rate Prescaler (BRP). This register can be read anytime and written only in init mode (MODE<2:0> bits set to <1xx> in the CTRL0 register).

Bit	Name	R/W	Default	Description
7-6	SJW1:0	R/W	0	Re-synchronization Jump Width bits <1:0>. These bits are used to compensate for phase shifts between different clock oscillators on the bus. They define the maximum number of time quanta (Tq) a bit can be shortened or lengthened to allow a node achieve re-synchronization to the edge of an incoming signal. Note that one time quantum (Tq) is the single unit of time within a bit time (see Bit Timing section). **SJW bits <1:0>** 00: SJW = 1 Tq 01: SJW = 2 Tq 10: SJW = 3 Tq 11: SJW = 4 Tq **Note:** ARINC 825 states that the Re-synchronization Jump Width shall be 1 Tq
5-0	BRP5:0	R/W	0	Baud Rate Prescaler bits <5:0>. The baud rate prescaler relates the system oscillator frequency, f_{OSC}, to the CAN bit time as described in the bit timing section. **BRP bits <5:0>** 000000: BRP = 1 000001: BRP = 2 000010: BRP = 3 000011: BRP = 4 . etc. . 111111: BRP = 64

BIT TIMING REGISTER 1: BTR1

(Write, SPI Op-code 0x1A)
(Read, SPI Op-code 0xD8)

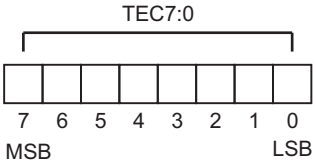


BTR1 configures the CAN protocol bit timing segments in terms of time quanta (Tq) and sets the number of sampling points. This register can be read anytime and written only in init mode (MODE<2:0> bits set to <1xx> in the CTRL0 register).

Bit	Name	R/W	Default	Description
7	SAMP	R/W	0	Samples per bit. This bit configures how many samples are taken per bit. 1 = three samples per bit. 0 = one sample per bit. **Notes:** ARINC 825 states that there shall be only one sample per bit. Furthermore, it is recommended to sample only once at higher CAN bit rates. Bit sampling occurs at the end of Phase Seg1.
6-4	TSEG2-2:0	R/W	0	Time Segment 2 bits <2:0>. Tseg2 = Phase Seg2 of the CAN protocol bit timing specification. Bits TSEG2-2:0 specify the number of time quanta in Phase Seg2. **Note:** Not all combinations are valid, since Phase Seg 2 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
3-0	TSEG1-3:0	R/W	0	Time Segment 1 bits <3:0>. Tseg1 = Prop Seg + Phase Seg1 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 Seg 2. **The CAN protocol states that the minimum number of Tq in a bit time shall be 8.** **TSEG1 bits <3:0>** 0000: Not valid 0001: Tseg1 = 2 Tq clock cycles 0010: Tseg1 = 3 Tq clock cycles 0011: Tseg1 = 4 Tq clock cycles 0100: Tseg1 = 5 Tq clock cycles . . . 1111: Tseg1 = 16 Tq clock cycles **Notes:** 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.

TRANSMIT ERROR COUNTER REGISTER: TEC

(Write, SPI Op-code 0x26)
(Read, SPI Op-code 0xEC)



The TEC register reflects the current value of the CAN Transmit Error Counter. This register can be written by SPI command for test purposes.

Bit	Name	R/W	Default	Description
7-0	TEC7:0	R/W	0x00	Transmit Error Counter bits <7:0>.

0 ≤ TEC ≤ 95: Error active status.

96≤ TEC ≤ 127: Error active status. Error warning flag, ERRW, set in STATF register. This may be used to generate a hardware interrupt if ERRWIE bit is set in STATFE register.

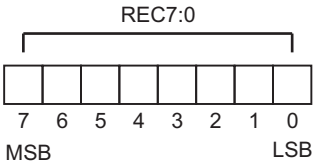
128 ≤ TEC≤ 255: Error passive status. Transmit error passive flag, TXERRP, set in ERR register. ERRP also set in STATF register. This may be used to generate a hardware interrupt if ERRPIE bit is set in STATFE register.

TEC > 255: Bus-off status. Bus-off flags, BUSOFF, set in ERR and STATF registers. The latter may be used to generate a hardware interrupt if BUSOFFIE bit is set in STATFE register.

The HI-3110 will, after entering bus-off state, automatically recover to error active status without host intervention if the BOR bit is set in control register CTRL0 and 128 x 11 consecutive recessive bits are detected on the bus. If the BOR bit is not set, bus-off recovery is managed by the host.

RECEIVE ERROR COUNTER REGISTER: REC

(Write, SPI Op-code 0x24)
(Read, SPI Op-code 0xEA)



The REC register reflects the current value of the CAN Receive Error Counter. This register can be written by SPI command for test purposes.

Bit	Name	R/W	Default	Description
7-0	REC7:0	R/W	0x00	Receiver Error Counter bits <7:0>.

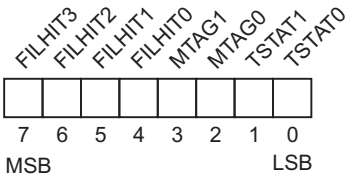
0 ≤ REC ≤ 95: Error active status.

96 ≤ REC ≤ 127: Error active status. Error warning flag, ERRW, set in STATF register. This may be used to generate a hardware interrupt if ERRWIE bit is set in STATFE register.

128 ≤ REC ≤ 255: Error passive status. Receive error passive flag, RXERRP, set in ERR register. ERRP also set in STATF register. This may be used to generate a hardware interrupt if ERRPIE bit is set in STATFE register.

MESSAGE STATUS REGISTER: MESSTAT

(Read only)
(Read, SPI Op-code 0xDA)

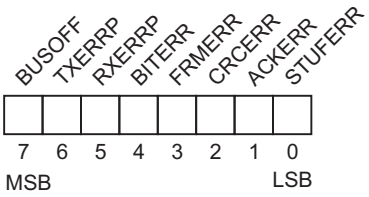


This register reflects transmission status and also which filters were responsible for filtering valid received messages. It is read-only.

Bit	Name	R/W	Default	Description
7-4	FILHIT3:0	R	0	Filter hit bits <3:0>. These bit combinations indicate which filters were responsible for filtering received messages . 0000: No filter matches the received message. 1000: Filter 0 matches, but filters 1, 2, 3, 4, 5, 6 or 7 may also match. 1001: Filter 1 matches, filter 0 does not, but filters 2, 3, 4, 5, 6 or 7 may also match. 1010: Filter 2 matches, filters 0 and 1 do not, but filters 3, 4, 5, 6 or 7 may also match. 1011: Filter 3 matches, filters 0 through 2 do not, but filters 4, 5, 6 or 7 may also match. 1100: Filter 4 matches, filters 0 through 3 do not, but filters 5, 6, or 7 may also match. 1101: Filter 5 matches, filters 0 through 4 do not, but filters 6 or 7 may also match. 1110: Filter 6 matches, filters 0 through 5 do not, but filter 7 may also match. 1111: Filter 7 matches, all other filters do not. **Note:** Filter checking is carried out by the internal logic in order of increasing filter number. Once a filter matches, no further checking takes place. Therefore, in the case of more than one filter matching for a given message, the lowest filter will be given priority and the FILHIT3:0 bits will reflect this value.
3-2	MTAG1:0	R	0	Message Tag bits <1:0>. These bits will reflect the last two bits of the host assigned message tag of the last successful transmission.
1-0	TSTAT1:0	R	0	Transmission Status bits <1:0>. These bits reflect the transmission status. 00: Transmission not enabled (TXEN = 0). 01: Transmission is enabled (TXEN = 1), but FIFO is empty. 10: Waiting to transmit or re-transmit. 11: Transmitter is currently transmitting a message or a flag.

ERROR REGISTER: ERR

(Read only)
(Read, SPI Op-code 0xDC)

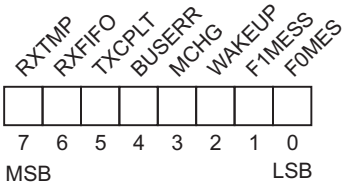


The ERR register indicates CAN bus status and protocol errors. It is read only. All bits default to 0 at power up and maintain their current status following reset. Bits 4:0 are reset following a host read.

Bit	Name	R/W	Default	Description
7	BUSOFF	R	0	Bus-off status indicator. This bit is set when TEC > 255. Node is in bus off condition. The bit is reset by HI-3110 when a successful bus recovery sequence is detected (128 x 11 consecutive recessive bits). d the FILHIT3:0 bits will reflect this value.
6	TXERRP	R	0	Transmit Error Passive status indicator. This bit is set when 128 ≤ TEC ≤ 255.
5	RXERRP	R	0	Receive Error Passive status indicator. This bit is set when 128 ≤ REC ≤ 255.
4	BITERR	R	0	Bit Error. A bit error was detected in a transmitted frame (the bit observed on the bus was opposite to what was expected).
3	FRMERR	R	0	Form Error. A Form error was detected in a receive frame.
2	CRCERR	R	0	CRC Error. A CRC error was detected in a receive frame.
1	ACKERR	R	0	Acknowledgement Error. An ACK error was detected in a receive frame.
0	STUFERR	R	0	Stuff Error. A bit stuffing error was detected in a received frame.

INTERRUPT FLAG REGISTER: INTF

(Read only)
(Read, SPI Op-code 0xDE)

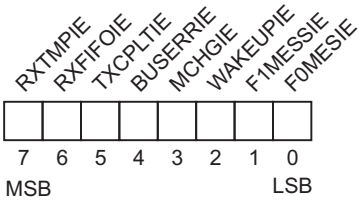


The Interrupt Flag Register INTF bits will be set by HI-3110 when the corresponding related events described below occur. If individual bits in the Interrupt Enable Register INTE are set, the INT pin will be latched high when any of the corresponding INTF bits are set. This alerts the host that one of the conditions below has occurred. Reading this register will clear all bits and reset the INT pin. The value of individual bits in the INTF register may also be reflected on the GP1 and GP2 pins by setting the correct bit combinations in the General Purpose Pins Enable Register GPINE (see section General Purpose Pins Enable Register).

Bit	Name	R/W	Default	Description
7	RXTMP	R	0	Message received in temporary receive buffer (unfiltered). This bit is set when a valid message is received in the temporary receive buffer.
6	RXFIFO	R	0	Message received in FIFO (filtered). This bit is set when a valid message passes the filter criteria and is passed from the temporary receive buffer to the receive FIFO.
5	TXCPLT	R	0	Successful transmission complete. This bit is set when a message is successfully transmitted.
4	BUSERR	R	0	Bus Error. This bit is set when a bus error occurs. Bits 4:0 in the ERR register can be read to determine the source of the error.
3	MCHG	R	0	Mode Change bit. This bit is set when the mode of operation is changed. Any pending transmissions in the transmit FIFO will be completed (FIFO will be emptied) before the mode change occurs.
2	WAKEUP	R	0	Wake-Up detected. This bit is set when the HI-3110 wakes up from Sleep Mode in response to bus activity.
1	F1MESS	R	0	Filter 1 passed a valid message. This bit is set when receive filter one passes a valid message. FILHIT3:0 bits will also be set to <1001> in the Message Status Register, MESSTAT.
0	F0MESS	R	0	Filter 0 passed a valid message. This bit is set when receive filter zero passes a valid message. FILHIT3:0 bits will also be set to <1000> in the Message Status Register, MESSTAT.

INTERRUPT ENABLE REGISTER: INTE

(Write SPI Op-code 0x1C)
(Read, SPI Op-code 0xE4)

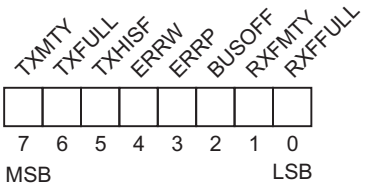


Setting bits in the Interrupt Enable Register causes a hardware interrupt to be generated at the INT pin when the corresponding bits in the Interrupt Flag Register are set by HI-3110 as a result of the related events described below.

Bit	Name	R/W	Default	Description
7	RXTMPIE	R/W	0	Enable interrupt when a message is received in the temporary receive buffer (unfiltered). Setting this bit causes a hardware interrupt to be generated at the INT pin when the RXTMP bit is set in the INTF register.
6	RXFIFOIE	R/W	0	Enable interrupt when a message is received in the receive FIFO (filtered). Setting this bit causes a hardware interrupt to be generated at the INT pin when the RXFIFO bit is set in the INTF register.
5	TXCPLTIE	R/W	0	Enable interrupt when a successful transmission is complete. Setting this bit causes a hardware interrupt to be generated at the INT pin when the TXCPLT bit is set in the INTF register.
4	BUSERRIE	R/W	0	Enable interrupt when a bus error occurs. Setting this bit causes a hardware interrupt to be generated at the INT pin when the BUSERR bit is set in the INTF register.
3	MCHGIE	R/W	0	Enable interrupt when a mode change occurs. Setting this bit causes a hardware interrupt to be generated at the INT pin when the MCHG bit is set in the INTF register.
2	WAKEUPIE	R/W	0	Enable interrupt when HI-3110 wakes up from Sleep Mode. Setting this bit causes a hardware interrupt to be generated at the INT pin when the WAKEUP bit is set in the INTF register.
1	F1MESSIE	R/W	0	Enable interrupt when filter one passes a message. Setting this bit causes a hardware interrupt to be generated at the INT pin when the F1MESS bit is set in the INTF register.
0	F0MESSIE	R/W	0	Enable interrupt when filter zero passes a message. Setting this bit causes a hardware interrupt to be generated at the INT pin when the F0MESS bit is set in the INTF register.

STATUS FLAG REGISTER: STATF

(Read-only)
(Read, SPI Op-code 0xE2)

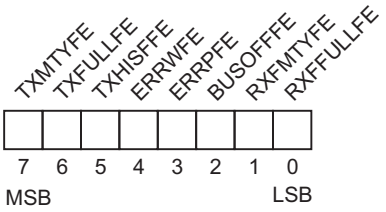


The Status Flag Register STATF bits will be set by HI-3110 when the corresponding related events described below occur. Unlike the Interrupt Flag Register, reading this register will NOT clear all bits. These bits are reset automatically by HI-3110 when the described status for each bit changes (e.g. if TXMTY is set and a message is loaded to the transmit FIFO, TXMTY will be automatically cleared by HI-3110). If individual bits in the Status Flag Enable Register STATFE are set, the STAT pin will pulse high when any of the enabled STATF bits are set. The value of individual bits in the STATF register may also be reflected on the GP1 and GP2 pins by setting the correct bit combinations in the General Purpose Pins Enable Register GPINE.

Bit	Name	R/W	Default	Description
7	TXMTY	R	1	Transmit FIFO is empty. This bit is set when the transmit FIFO is empty. This is the default following Reset.
6	TXFULL	R	0	Transmit FIFO is full. This bit is set when the transmit FIFO is full.
5	TXHISF	R	0	Transmit history FIFO full. This bit is set when the transmit history FIFO is full. Up to eight messages can be stored in the transmit history FIFO (Note: a user generated message tag and a time tag are stored with each message).
4	ERRW	R	0	Error warning flag. This bit is set when 96 ≤ (TEC or REC) ≤ 127.
3	ERRP	R	0	Error passive indicator. This bit is set when the device enters error passive mode.
2	BUSOFF	R	0	BUSOFF indicator. This bit is set when the device enters bus-off state.
1	RXFMTY	R	1	Receive FIFO empty. This bit is set when the receive FIFO is empty. This is the default following Reset.
0	RXFFULL	R	0	Receive FIFO full. This bit is set when the receive FIFO is full.

STATUS FLAG ENABLE REGISTER: STATE

(Write, SPI Po-code 0x1E)
(Read, SPI Op-code 0xE6)

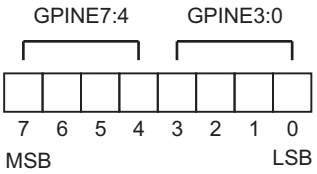


Setting bits in the Status Flag Enable Register causes the STAT pin to go high when any of the corresponding bits in the Status Flag Register are set by HI-3110 as a result of the related events described below.

Bit	Name	R/W	Default	Description
7	TXMTYFE	R/W	0	Transmit FIFO empty status flag enable. Setting this bit causes the status of the TXMTY bit in the Status Flag Register to be output on the STAT pin.
6	TXFULLFE	R/W	0	Transmit FIFO full status flag enable. Setting this bit causes the status of the TXFULL bit in the Status Flag Register to be output on the STAT pin.
5	TXHISFFE	R/W	0	Transmit History FIFO full status flag enable. Setting this bit causes the status of the TXHISF bit in the Status Flag Register to be output on the STAT pin.
4	ERRWFE	R/W	0	Error warning status flag enable (96 ≤ (TEC or REC) ≤ 127). Setting this bit causes the status of the ERRW bit in the Status Flag Register to be output on the STAT pin.
3	ERRPFE	R/W	0	Error Passive status flag enable. Setting this bit causes the status of the ERRP bit in the Status Flag Register to be output on the STAT pin.
2	BUSOFFFE	R/W	0	Bus-off status flag enable. Setting this bit causes the status of the BUSOFF bit in the Status Flag Register to be output on the STAT pin.
1	RXFMTYFE	R/W	0	Receive FIFO empty status flag enable. Setting this bit causes the status of the RXFMTY bit in the Status Flag Register to be output on the STAT pin.
0	RXFFULLFE	R/W	0	Receive FIFO full status flag enable. Setting this bit causes the status of the RXFFULL bit in the Status Flag Register to be output on the STAT pin.

GENERAL PURPOSE PINS ENABLE REGISTER: GPINE

(Write, SPI Op-code 0x22)
(Read, SPI Op-code 0xE8)



Setting bits in the General Purpose Pins Enable Register allows the user to reflect the values of the Interrupt Flag bits in the INTF Register or the Status Flag bits in the STATF Register on the GP1 andGP2 pins.

Bit	Name	R/W	Default	Description
7-4	GPINE7:43:0	R/W	0000	Reflect status of interrupt flag bits in INTF or status flag bits in STATF on GP2 pin as follows:

0000: GP2 pin is asserted when F0MESS bit is set in register INTF.
0001: GP2 pin is asserted when F1MESS bit is set in register INTF.
0010: GP2 pin is asserted when WAKEUP bit is set in register INTF.
0011: GP2 pin is asserted when MCHG bit is set in register INTF.
0100: GP2 pin is asserted when BUSERR bit is set in register INTF.
0101: GP2 pin is asserted when TXCPLT bit is set in register INTF.
0110: GP2 pin is asserted when RXFIFO bit is set in register INTF.
0111: GP2 pin is asserted when RXTMP bit is set in register INTF.

1000: GP2 pin is asserted when RXFFULL bit is set in register STATF.
1001: GP2 pin is asserted when RXFMPTY bit is set in register STATF.
1010: GP2 pin is asserted when BUSOFF bit is set in register STATF.
1011: GP2 pin is asserted when ERRP bit is set in register STATF.
1100: GP2 pin is asserted when ERRW bit is set in register STATF.
1101: GP2 pin is asserted when TXHISF bit is set in register STATF.
1110: GP2 pin is asserted when TXFULL bit is set in register STATF.
1111: GP2 pin is asserted when TXMPTY bit is set in register STATF.

3-0	GPINE3:0	R/W	0000	Reflect status of interrupt flag bits in INTF or status flag bits in STATF on GP1 pin as follow:

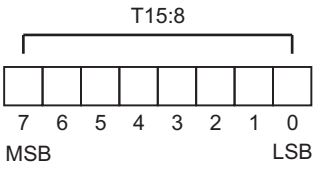
0000: GP1 pin is asserted when F0MESS bit is set in register INTF.
0001: GP1 pin is asserted when F1MESS bit is set in register INTF.
0010: GP1 pin is asserted when WAKEUP bit is set in register INTF.
0011: GP1 pin is asserted when MCHG bit is set in register INTF.
0100: GP1 pin is asserted when BUSERR bit is set in register INTF.
0101: GP1 pin is asserted when TXCPLT bit is set in register INTF.
0110: GP1 pin is asserted when RXFIFO bit is set in register INTF.
0111: GP1 pin is asserted when RXTMP bit is set in register INTF.

1000: GP1 pin is asserted when RXFFULL bit is set in register STATF.
1001: GP1 pin is asserted when RXFMPTY bit is set in register STATF.
1010: GP1 pin is asserted when BUSOFF bit is set in register STATF.
1011: GP1 pin is asserted when ERRP bit is set in register STATF.
1100: GP1 pin is asserted when ERRW bit is set in register STATF.
1101: GP1 pin is asserted when TXHISF bit is set in register STATF.
1110: GP1 pin is asserted when TXFULL bit is set in register STATF.
1111: GP1 pin is asserted when TXMPTY bit is set in register STATF.

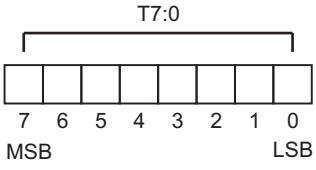
FREE-RUNNING TIMER

(Read-only)
(Read, SPI Op-code 0xFA)

UPPER BYTE: TIMERUB



LOWER BYTE: TIMERLB



Bit	Name	R/W	Default	Description
7-0	T15:8	R/W	0x00	Free Running Timer Upper Byte, bits <15:8>. The 16-bit free-running timer starts counting from zero following RESET. The timer counts linearly up to 0xFFFF and then wraps around to 0x0000. **Note: A timer overrun is not flagged by the HI-3110.** The timer is clocked by the HI-3110 bit clock and continuously increments by the bit rate (default). The user may program the timer to 2, 4 or 8 bit clocks by setting the TTDIV1:0 bits in Control Register 0, CTRL0. The timer may be reset by the host after any interrupt condition (e.g. reading the receive buffer, transmit FIFO empty, etc), by issuing an SPI instruction (see Table 1, SPI Instruction Set).
7-0	T7:0	R/W	0x00	Free Running Timer Lower Byte, bits <7:0>. **NOTE:** A single SPI Op-code (0xFA) reads the16-bit timer value in two SPI data bytes, TIMERUB and TIMERLB. TIMERUB is read first.

SERIAL PERIPHERAL INTERFACE

SERIAL PERIPHERAL INTERFACE (SPI) BASICS

The HI-3110 uses an SPI synchronous serial interface for host access to internal registers and data FIFOs. 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 read / write cycles are completely self-timed.

The SPI (Serial Peripheral Interface) protocol specifies master and slave operation; the HI-3110 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-3110 operates in mode 0 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 (CPHA = 0, CPOL = 0). Be sure to set the host SPI logic for mode 0.

As seen in Figure 9, SPI Mode 0 holds SCK in the low state when idle. 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-3110 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-3110 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), per Figure 9 below. However the HI-3110 operates half duplex, maintaining high impedance on the SO output, except when actually transmitting serial data. When the HI-3110 is sending data on SO during read operations, activity on its SI input is ignored. Figures 10 and 11 show actual behavior for the HI-3110 SO output.

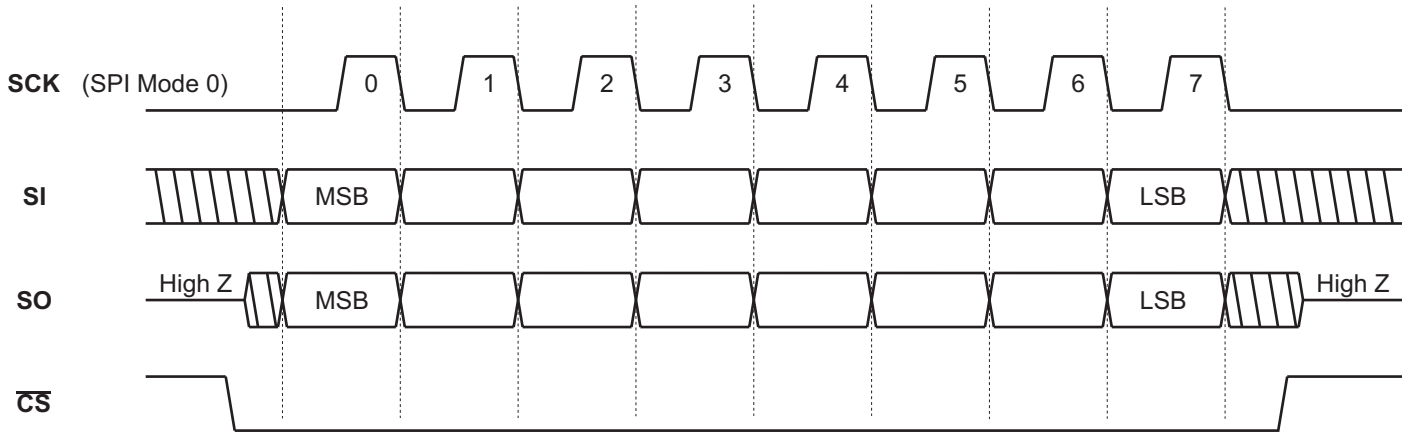


Figure 9. Generalized Single-Byte Transfer Using SPI Protocol Mode 0

HOST SERIAL PERIPHERAL INTERFACE, cont.

HI-3110 SPI COMMANDS

For the HI-3110, 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-3110 command byte reception is half-duplex, the host discards the dummy byte it receives while serially transmitting the command byte.

Figures 10 and 11 show read and write timing as it appears for a single-byte and dual-byte register operation. 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 byte read or write cycles may be performed by transferring more than one byte before $\overline{CS}$ is negated. Tables 2 - 5 define the required number of bytes for each instruction.

Note: SPI Instruction op-codes not shown in Tables 2 - 5 are “reserved” and must not be used. Further, these op-codes will not provide meaningful data in response to read commands.

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

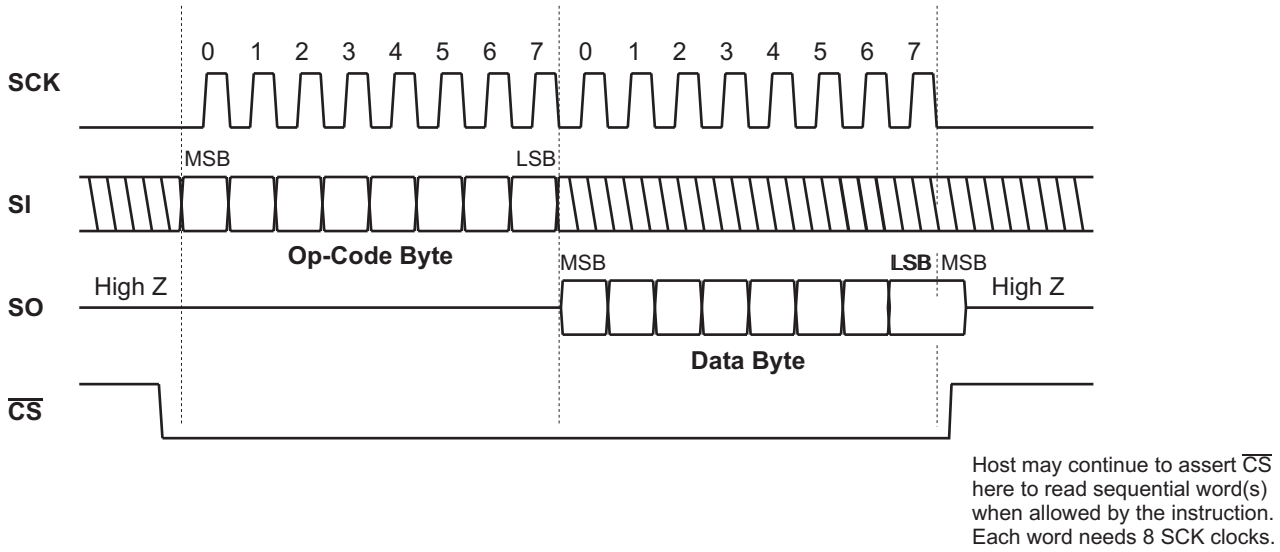


Figure 10. Single-Byte Read From a Register

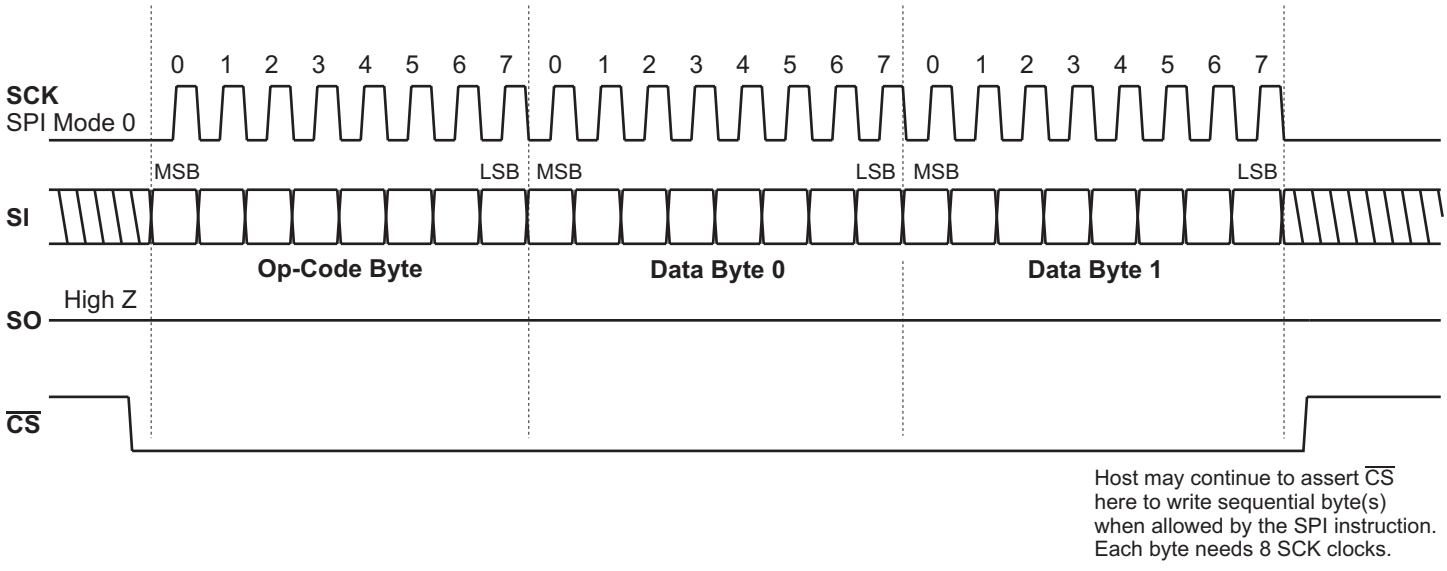


Figure 11. 2-Byte Write example

Table 1. SPI Instruction Set

SPI Instruction	Command Hex	Data Field	Bit Content
READ Commands			
Read Temporary Receive Buffer with Time Tag	0x42	15 bytes	(see Table 5, bytes 2-16)
Read Temporary Receive Buffer without Time Tag	0x44	13 bytes	(see Table 5, bytes 4-16)
Read Next FIFO Message with Time Tag	0x46	16 bytes	(see Table 5, bytes 1-16)
Read Next FIFO Message without Time Tag	0x48	14 bytes	(see Table 5, bytes 1 & 4-16)
Read Next FIFO Message Data only with Time Tag	0x4A	11 bytes	(see Table 5, bytes 1-3 & 9-16)
Read Next FIFO Message Data only without Time Tag	0x4C	9 bytes	(see Table 5, bytes 1 & 9-16)
Read Filter 0 ID	0xA2	6 bytes	(see Table 4a)
Read Filter 1 ID	0xA4	6 bytes	(see Table 4a)
Read Filter 2 ID	0xA6	6 bytes	(see Table 4a)
Read Filter 3 ID	0xA8	6 bytes	(see Table 4a)
Read Filter 4 ID	0xAA	6 bytes	(see Table 4a)
Read Filter 5 ID	0xAC	6 bytes	(see Table 4a)
Read Filter 6 ID	0xAE	6 bytes	(see Table 4a)
Read Filter 7 ID	0xB2	6 bytes	(see Table 4a)
Read Mask 0 ID	0xB4	6 bytes	(see Table 4b)
Read Mask 1 ID	0xB6	6 bytes	(see Table 4b)
Read Mask 2 ID	0xB8	6 bytes	(see Table 4b)
Read Mask 3 ID	0xBA	6 bytes	(see Table 4b)
Read Mask 4 ID	0xBC	6 bytes	(see Table 4b)
Read Mask 5 ID	0xBE	6 bytes	(see Table 4b)
Read Mask 6 ID	0xC2	6 bytes	(see Table 4b)
Read Mask 7 ID	0xC4	6 bytes	(see Table 4b)
Read Control Register 0	0xD2	1 byte	(see CTRL0 Register Definition)
Read Control Register 1	0xD4	1 byte	(see CTRL1 Register Definition)
Read Bit Timing Register 0	0xD6	1 byte	(see BTR0 Register Definition)
Read Bit Timing Register 1	0xD8	1 byte	(see BTR1 Register Definition)
Read Message Status Register	0xDA	1 byte	(see MESSTAT Register Definition)
Read Error Register	0xDC	1 byte	(see ERR Register Definition)
Read Interrupt Flag Register	0xDE	1 byte	(see INTF Register Definition)
Read Status Flag Register	0xE2	1 byte	(see STATF Register Definition)
Read Interrupt Enable Register	0xE4	1 byte	(see INTE Register Definition)
Read Status Flag Enable Register	0xE6	1 byte	(see STATFE Register Definition)

Table 1. SPI Instruction Set (cont.)

SPI Instruction	Command Hex	Data Field	Bit Content
WRITE Commands			
Write Transmit FIFO	0x12	4 to 12 bytes for Std frame or 6 to 14 bytes for Ext frame	
Write Control Register 0	0x14	1 byte	(see CTRL0 Register Definition)
Write Control Register 1	0x16	1 byte	(see CTRL1 Register Definition)
Write Bit Timing Register 0	0x18	1 byte	(see BTR0 Register Definition)
Write Bit Timing Register 1	0x1A	1 byte	(see BTR1 Register Definition)
Write Interrupt Enable Register	0x1C	1 byte	(see INTE Register Definition)
Write Status Flag Enable Register	0x1E	1 byte	(see STATFE Register Definition)
Write General Purpose Pins Enable Register	0x22	1 byte	(see GPINE Register Definition)
Write REC Register (Test only)	0x24	1 byte	(see REC Register Definition)
Write TEC Register (Test only)	0x26	1 byte	(see TEC Register Definition)
Write Filter 0 ID	0x62	6 bytes	(see Table 4a)
Write Filter 1 ID	0x64	6 bytes	(see Table 4a)
Write Filter 2 ID	0x66	6 bytes	(see Table 4a)
Write Filter 3 ID	0x68	6 bytes	(see Table 4a)
Write Filter 4 ID	0x6A	6 bytes	(see Table 4a)
Write Filter 5 ID	0x6C	6 bytes	(see Table 4a)
Write Filter 6 ID	0x6E	6 bytes	(see Table 4a)
Write Filter 7 ID	0x72	6 bytes	(see Table 4a)
Write Mask 0 ID	0x74	6 bytes	(see Table 4b)
Write Mask 1 ID	0x76	6 bytes	(see Table 4b)
Write Mask 2 ID	0x78	6 bytes	(see Table 4b)
Write Mask 3 ID	0x7A	6 bytes	(see Table 4b)
Write Mask 4 ID	0x7C	6 bytes	(see Table 4b)
Write Mask 5 ID	0x7E	6 bytes	(see Table 4b)
Write Mask 6 ID	0x82	6 bytes	(see Table 4b)
Write Mask 7 ID	0x84	6 bytes	(see Table 4b)

Table 1. SPI Instruction Set (cont.)

SPI Instruction	Command Hex	Data Field	Bit Content
READ Commands (ctd)			
Read General Purpose Pins Enable Register	0xE8	1 byte	(see GPINE Register Definition)
Read REC Register	0xEA	1 byte	(see REC Register Definition)
Read TEC Register	0xEC	1 byte	(see TEC Register Definition)
Read Transmit History FIFO	0xEE	3 bytes	(see Table 3)
Reserved (Factory test purposes only)	0x4E	13 bytes	
Reserved (Factory test purposes only)	0xF4	1 byte	
Reserved (Factory test purposes only)	0xF6	1 byte	
Read Bus-Off Recessive Count	0xF8	1 byte	
Read Time Tag	0xFA	2 bytes	(see TIMERUB and TIMERLB Register Definitions)
RESET Commands			
Abort Transmission (stops current transmission and resets TXEN & TX1M)	0x52	None	(see CTRL1 Register Definition)
Reset (Clear) Transmit FIFO (resets TXEN & TX1M, but completes current transmission)	0x54	None	
Master Reset	0x56	None	
Reset Time Tag Counter	0x58	None	(see TIMERUB and TIMERLB Register Definitions)
Reset Receive FIFO	0x5A	None	

HI-3110 Transmit Buffer

The HI-3110 transmit buffer consists of an eight message FIFO which allows transmission of up to eight messages. Messages are loaded to the transmit FIFO via SPI instruction. Similarly, an SPI instruction resets (clears) the transmit FIFO.

Transmission from the FIFO is enabled by asserting the TXEN pin or setting the TXEN bit in CTRL1. If TXEN transmission is enabled, all loaded messages are automatically sent if the bus is available. If TXEN is not enabled, messages will not be sent until TXEN is set. If TXEN is reset, a single message may also be sent by setting the TX1M bit in CTRL1.

MESSAGE TRANSMISSION SEQUENCE

A simplified transmission flow is illustrated in Figure 9. The next message to be transmitted (or current message trying to gain access to the bus) is loaded from the FIFO to the Transmit Buffer. This will happen automatically if TXEN or TX1M are set in CTRL1.

If the bus is available, the message is sent. The transmission can be aborted at any time using the Abort Transmission SPI command (see Table 1). Care should be exercised when using the command as it may cause other nodes on the bus to generate error frames if the message is aborted prior to completing transmission. The current transmission sequence can also be paused by resetting the TXEN bit in CTRL1 (or pulling TXEN pin low). In this case, the current message will be completed and any remaining messages in the transmit FIFO will not be transmitted.
If the current message transmission goes ahead, two things can happen:
a) The message is successful, and the transmit buffer is now ready to receive the next message from the transmit FIFO. The transmit history FIFO is updated (see below).
b) The message is not successful due to lost arbitration or message error.

i. Lost Arbitration: If arbitration is lost, the current message stays in the Transmit Buffer for re-transmission.

ii. Message error: Flag BUSERR is set in the Interrupt Flag Register. An error frame is sent and an optional hardware interrupt may also be generated at the INT pin if enabled in the Interrupt Enable Register (bit BUSERRIE = 1). If there is an error, the current message stays in the Transmit Buffer for automatic re-transmission in accordance with the CAN protocol.

NOTE: If OSM is set, re-transmission will NOT be attempted upon loss of arbitration or message error. The existing message will remain in the FIFO. If the user requires a new message on the next transmission cycle, the FIFO must be cleared using SPI command 0x54 and re-loaded with the new message.

LOADING THE TRANSMIT FIFO VIA SPI

The transmit FIFO holds up to 8 messages and is loaded via SPI instruction (see Table 1). The data format for the SPI instruction is illustrated in Table 2. The host simply needs to issue the SPI write command 0x12 followed by the SPI data field as described in Table 2. For standard frames, the SPI data field has the format shown in Table 2(a). For extended frames, the SPI data field has the format shown in Table 2(b). The HI-3110 will automatically interpret Standard or Extended frames by decoding the IDE bit. The HI-3110 also decodes the data length code (DLC) and ignores data bytes greater than the DLC value (**Note:** a DLC of greater than 8 is automatically assumed to be equal to 8). The user has the option of assigning a unique message tag to each message which can be used later to identify successfully transmitted messages from the transmit history FIFO. One frame at a time can be loaded to the transmit FIFO using SPI command 0x12. The byte format should be as shown in Table 2 and the $\overline{CS}$ pin should remain low during the entire SPI sequence.

TRANSMIT HISTORY FIFO

The Transmit History FIFO can optionally be used by the host to keep a record of up to eight successfully transmitted messages. A user-assigned message tag and a time tag are stored for each message. The data format is shown in Table 3. The time tag is assigned from the value of the free running counter upon receipt of an ACK bit. The transmit history FIFO is cleared when read by SPI command 0xEE (see Table 3).

SINGLE FRAME TRANSMISSION

For single frame transmission, use TX1M, bit 6, Register CTRL1. When using TX1M for single frame transmission, TXEN should be held low. **Pulsing the TXEN pin for single frame transmission is not recommended.**

If single frame transmission is preferred using the TXEN pin, the user should load only ONE message to the FIFO and hold TXEN high to transmit in the normal way. Following a successful transmission, TXEN should be reset and the FIFO re-loaded with the next message. A successful transmission may be confirmed by polling the Transmit History FIFO or by monitoring the Transmit FIFO empty flag, TXMTY, in the STATF register.

HI-3110 Transmit Message Flow Diagram

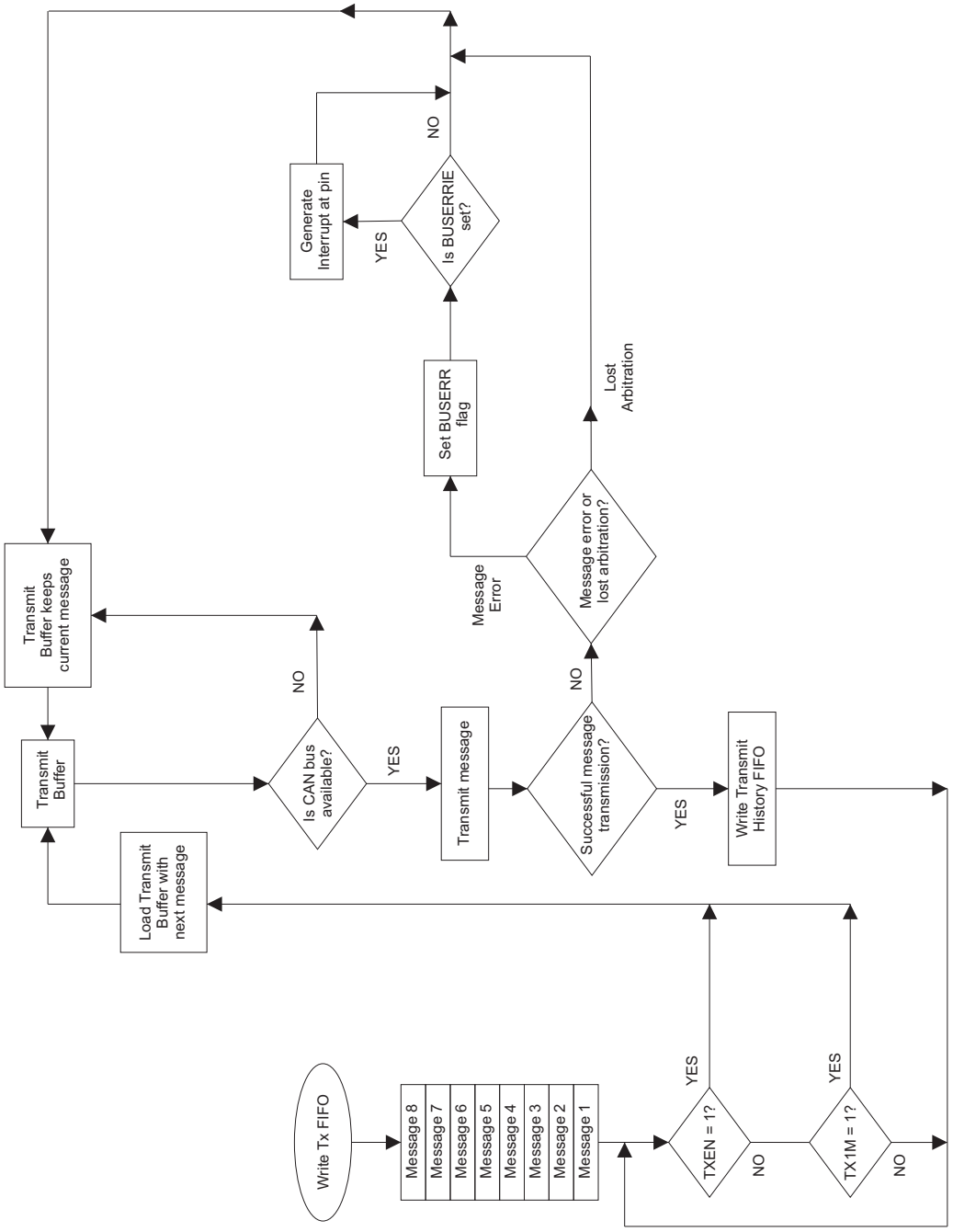
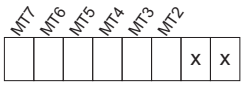
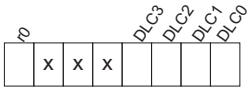


Figure 12. Simplified Transmission Flow Diagram

Table 2. SPI Transmit Data Format

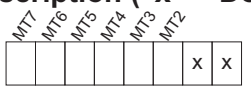
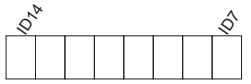
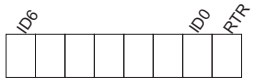
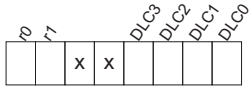
a) Standard Frame Transmit

Byte	Byte Content	Bit Description (”x” = Don’t care)
1	Message Tag*	
2	ID28 to ID21	
3	ID20 to ID18, RTR, IDE	
4	r0, DLC3 to DLC0	
5	*Data Byte 1	
6	Data Byte 2	.
7	Data Byte 3	.
8	Data Byte 4	.
9	Data Byte 5	.
10	Data Byte 6	.
11	Data Byte 7	.
12	Data Byte 8	

*** Note:** The Message Tag is a host-assigned identifier that is stored along with a time tag in the Transmit History FIFO. It can be used by the host to log successfully transmitted messages at a later time.

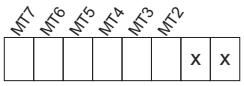
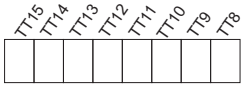
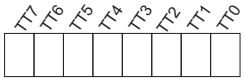
Table 2. SPI Transmit Data Format

b) Extended Frame Transmit

Byte	Content	Bit Description (”x” = Don’t care)
1	Message Tag*	
2	ID28 to ID21	
3	ID20 to ID18, SRR, IDE, ID17 to ID15	
4	ID14 to ID7	
5	ID6 to ID0, RTR	
6	r0, r1, DLC3 to DLC0	
7	Data Byte 1	
8	Data Byte 2	
9	Data Byte 3	
10	Data Byte 4	
11	Data Byte 5	
12	Data Byte 6	
13	Data Byte 7	
14	Data Byte 8	

*** Note:** The Message Tag is a host-assigned identifier that is stored along with a time tag in the Transmit History FIFO. It can be used by the host to log successfully transmitted messages at a later time.

Table 3. Transmit History FIFO Data Format

Byte	Content	Bit Description ("x" = Don't care)
1	Message Tag	
2	Time Tag Upper Byte	
3	Time Tag Lower Byte	

HI-3110 Receive Buffers and Frame Acceptance Filters

RECEIVE BUFFERS

The HI-3110 has an extremely flexible receive buffer and ID filter scheme. **Acceptance filters and masks may only be programmed when the device is in Initialization Mode.** The basic concept is shown in figure 13. All valid received messages (both standard or extended frames) are stored in the temporary receive buffer before passing through the filter bank. The host can read the temporary receive buffer using SPI commands 0x42 or 0x44 (see table 1). The filter bank must be enabled by setting the FILTON bit in Control Register, CTRL1. The default after reset is FILTON = 0, which disables the filter bank and stores every valid message received in the FIFO. With filtering enabled, it is possible to filter up to eight extended identifiers plus the first two associated data bytes. Any filtered messages will be passed to the receive FIFO. Up to eight messages can be stored in the receive FIFO. All valid received messages have a 16-bit time tag appended following transmission of the ACK bit. The user can decide to retrieve the time tag or not via dedicated SPI instructions (see Table 1).

FILTER AND MASK ID FORMAT

The HI-3110 allows filtering of up to 8 unique extended frames with the first two data bytes. Filtering is enabled by setting the FILTON bit in Control Register 1, CTRL1. It the FILTON bit is not set, then filtering is globally disabled and all CAN IDs are accepted.

There is a specific SPI instruction for loading and reading each filter and mask. The format is the same for both standard and extended IDs and is shown in Table 4. Bits specific to Extended IDs should be written as zeros for Standard IDs. The filter mechanism works by comparing the filter ID to the CAN message ID. If the corresponding bit in the mask ID is logic one, then the CAN ID bit must match the filter ID for acceptance to occur. If a mask ID bit is logic zero, then acceptance will occur regardless of the value of the CAN ID bit. In this case, the filter bits are don't care.

Following reset, all eight filter and mask registers should be loaded before enabling the FILTON bit. **Note that following reset, filter and mask bits are not reset**, therefore the FILTON bit may be set to enable filtering using the pre-reset mask and filter values.

READING THE RECEIVE BUFFERS VIA SPI

Table 1 summarizes the SPI instructions for reading the receive buffers. The host has a choice of retrieving a message with a 16-bit time tag or not. The receive data format is shown in Table 5. The first data byte identifies whether the frame was standard or extended format and the FILHIT2:0 bits identify which filter passed the message; <000> to <111>. If more than one filter passed the message, the lowest value will be given priority and be identified by the FILHIT2:0 bits. As can be seen in Table 5, bits specific to extended frames will be read as zeros for standard frames. If the received data does not contain an 8 byte payload (8 data bytes), the HI-3110 will pad the remaining data bytes with zeros. The host should keep $\overline{CS}$ low for the duration of the SPI sequence.

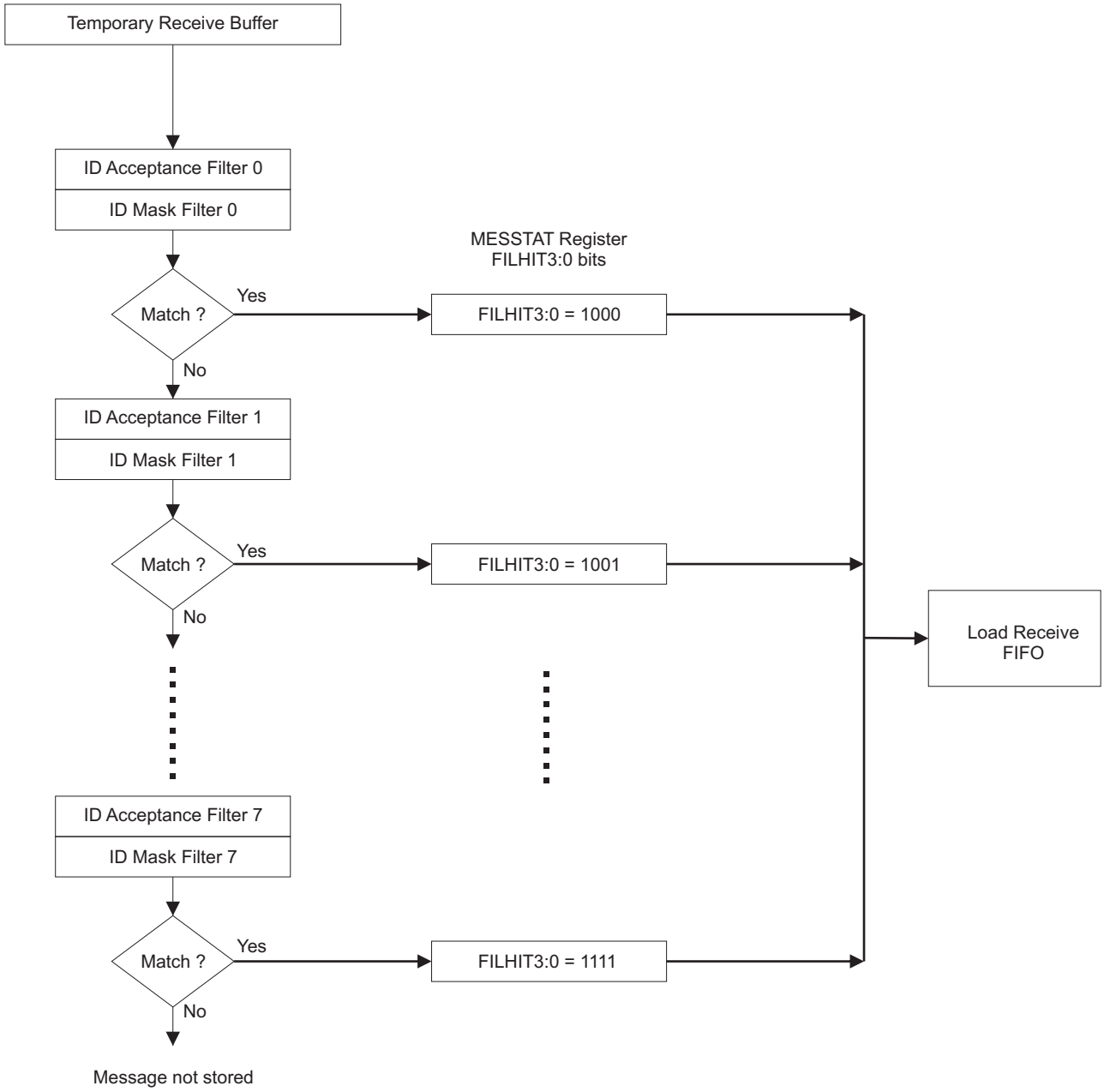
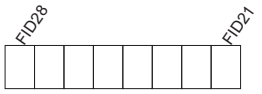
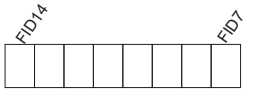
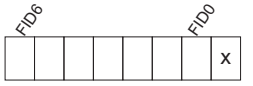
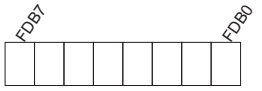


Figure 13. Receive Buffer Structure

Table 4. SPI Filter and Mask ID Format

a) Filter ID Format

Byte	Content	Bit Description ("x" = Don't care)
1	ID28 to ID21	
2	ID20 to ID18, RTR, IDE, ID17 to ID15 (Note: ID17 to ID15 should be written as zeros for Standard ID)	
3	ID14 to ID7 (Note: Written as zeros for Standard ID)	
4	ID6 to ID0 (Note: Written as zeros for Standard ID)	
5	Data Byte 1	
6	Data Byte 2	

b) Mask ID Format

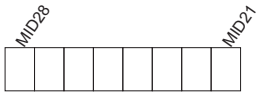
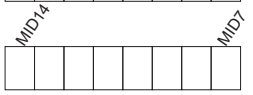
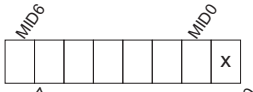
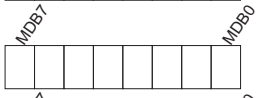
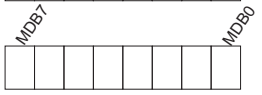
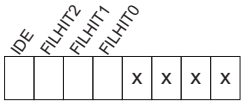
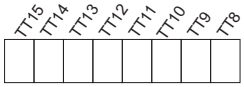
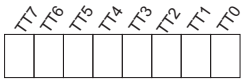
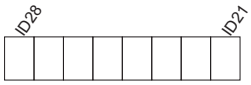
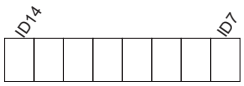
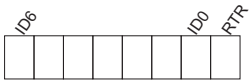
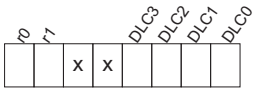
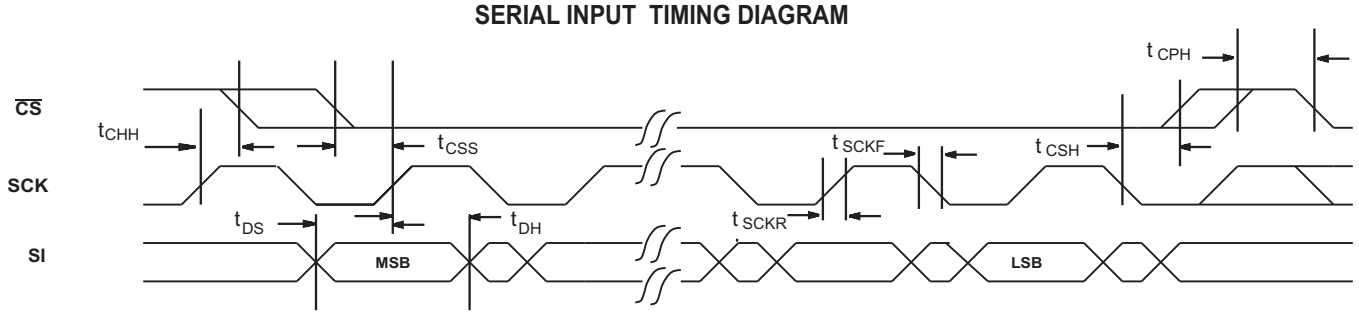
Byte	Content	Bit Description ("x" = Don't care)
1	ID28 to ID21	
2	ID20 to ID18, RTR, IDE, ID17 to ID15 (Note: ID17 to ID15 should be written as zeros for Standard ID)	
3	ID14 to ID7 (Note: Written as zeros for Standard ID)	
4	ID6 to ID0 (Note: Written as zeros for Standard ID)	
5	Data Byte 1	
6	Data Byte 2	

Table 5. SPI Receive Data Format

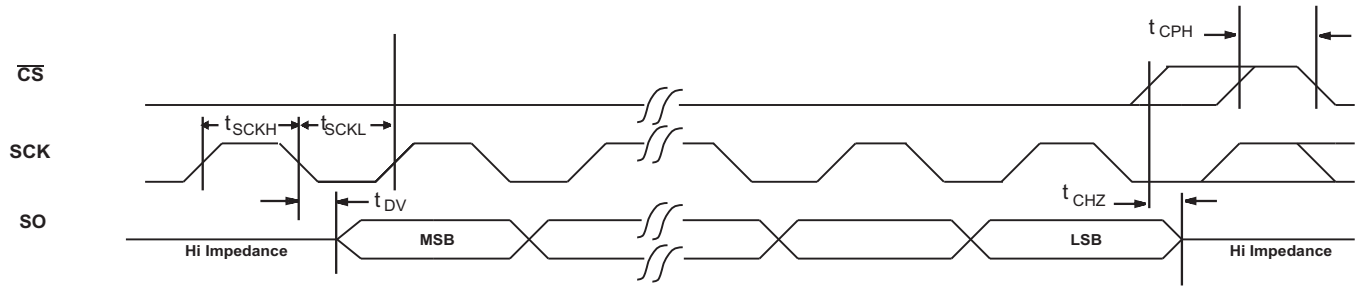
Byte	Content	Bit Description (”x” = Don’t care)
1	Message Tag (Note: This byte does not apply to the Temporary Receive Buffer)	
2	Time Tag Upper Byte (Note: This byte only applies to time tag SPI instructions; see Table 1)	
3	Time Tag Lower Byte (Note: This byte only applies to time tag SPI instructions; see Table 1)	
4	ID28 to ID21	
5	ID20 to ID18, SRR, IDE, ID17 to ID15 (Note: SRR and ID17 to ID15 are read as zeros for Std Frames)	
6	ID14 to ID7 (Read as zeros for Std Frames)	
7	ID6 to ID0, RTR (Note: ID6 to ID0 are read as zeros for Std Frames)	
8	r0, r1, DLC3 to DLC0 (Note: r1 is read as zero for Std Frames)	
9	Data Byte 1	
10	Data Byte 2	
11	Data Byte 3	
12	Data Byte 4	
13	Data Byte 5	
14	Data Byte 6	
15	Data Byte 7	
16	Data Byte 8	

TIMING DIAGRAMS

SERIAL INPUT TIMING DIAGRAM



SERIAL OUTPUT TIMING DIAGRAM



ABSOLUTE MAXIMUM RATINGS

(Voltages referenced to Gnd = 0V)

Supply Voltages: VDD:.....7V VLOGIC:.....7V	Electrostatic Discharge (ESD)[1], pins CANH, CANL .....+/- 6kV all other pins .....+/- 4kV
DC Voltages at CANH, CANL:.....-58V to +58V DC Voltages at all other pins:.....-0.5V to V_{DD} +0.5V	Storage Temperature Range: .....-65°C to +150°C
Soldering Temperature: (Ceramic).....60 sec. at +300°C (Plastic - leads).....10 sec. at +280°C (Plastic - body) .....+260°C Max.	Operating Temperature Range: (Plastic).....-55°C to +175°C (Ceramic).....-55°C to +200°C

NOTES:
1. Human Body Model (HBM).

Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. These are stress ratings only. Functional operation of the device at these or any other conditions above those indicated in the operational sections of the specifications is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

DC ELECTRICAL CHARACTERISTICS

V_{LOGIC} = 3.3V or 5V, V_{DD} = 5V. Operating temperature range (unless otherwise noted).

PARAMETER	SYMBOL	CONDITIONS	LIMITS			UNIT
			MIN	TYP	MAX	
Supply						
Supply Voltage	V_{LOGIC} V_{DD}		3.15 4.75		5.25 5.25	V V
Supply Current		All inputs/outputs open except the oscillator configured for 24MHz and with 60 Ohm resistors at CANH and CANL to SPLIT				
	I_{LOGIC}	V_{LOGIC} current: Operating Sleep		0.6 5.0	5 25	mA µA
	I_{DD}	V_{DD} current: Operating Sleep		6.3 15	15 50	mA µA
Logic Inputs						
High-Level Input Voltage	V_{IH}		0.7V_{LOGIC}			V
Low-Level Input Voltage	V_{IL}				0.3V_{LOGIC}	V
Input source current	I_{IH}	V_{IL} = 0V, V_{LOGIC} = 5.0V $\overline{CS}$ (pull down) All other inputs		100	200 1.5	µA µA
Input sink current	I_{IL}	V_{IH} = 5.0V, V_{LOGIC} = 5.0V $\overline{CS}$ SCK, SI, MR (pull up) TXEN (pull up)	-1.5 -200 -100	 -100 -50		µA µA µA
Pull-up current ($\overline{CS}$)	I_{PU}	V_{PU} = V_{LOGIC}	-200		-30	µA
Pull-down current (SI, SCK, MR pins)	I_{PD}	V_{PD} = GND	30		200	µA
(TXEN)	I_{PD}	V_{PD} = GND	15		100	µA
Logic Outputs						
High-Level Output Voltage	V_{OH}	I_{OH} = -1mA	0.9V_{LOGIC}			V
Low-Level Output Voltage	V_{OL}	I_{OL} = 1mA			0.1V_{LOGIC}	V
Output sink current	I_{OL}	V_{OUT} = 0.4V	1.6			mA
Output source current	I_{OH}	V_{OUT} = V_{DD} - 0.4V			-1	mA
SO Output Leakage	I_{SOL}	V_{OUT} = V_{LOGIC} Tri-state condition			10	µA
Oscillator Pins						
OSCIN Feedback Resistor	R_{OSC}	Control Register 1, bit 3 = 0 Input at V_{LOGIC} or 0V Control Register 1, bit 3 = 1 Input to V_{LOGIC} only (pull down)	4 4	5 5	6 6	MΩ MΩ
OSCOUT Drive Current	I_{OSC}	V_{LOGIC} = 3.3V, V_{IN} = 3.3V, V_{OUT} = 0.5V V_{LOGIC} = 3.3V, V_{IN} = 0V, V_{OUT} = 2.8V	0.85	1.1 -0.7	 -0.45	mA mA

DC ELECTRICAL CHARACTERISTICS (ctd.)

V_{LOGIC} = 3.3V or 5V, V_{DD} = 5V. Operating temperature range (unless otherwise noted).

PARAMETER	SYMBOL	CONDITIONS	LIMITS			UNIT
			MIN	TYP	MAX	
Bus Lines (pins CANH, CANL)						
CANH dominant output voltage CANL dominant output voltage	VO(CANH) VO(CANL)	See Fig 14. RL = 60 Ω	3 0.5	3.6 1.4	4.25 1.75	V V
Matching of dominant output voltage, VCC − VCANH − VCANL	VOM	See Fig. 14. RL = 60 Ω	− 100	0	150	mV
Dominant differential output voltage Recessive differential output voltage	VDIFF(d)(o) VDIFF(r)(o)	45 Ω < RL < 65 Ω, see Fig. 15 No Load	1.5 − 50	0	3 50	V mV
Recessive output voltage	VCANH(r), VCANL(r)	No Load, see Fig. 14	2	0.5VDD	3	V
Dominant differential input voltage (receiver) Recessive differential input voltage (receiver) Differential input hysteresis (receiver)	VDIFF(d)(i) VDIFF(r)(i) VDIFF(hys)	− 12 V < VCANH, VCANL < + 12 V − 12 V < VCANH, VCANL < + 12 V − 12 V < VCANH, VCANL < + 12 V See Fig. 18	0.9	 0 70	 0.5	V V mV
Input leakage current	ICANH, ICANL	VDD = 0 V (unpowered node)	− 200		+ 200	µA
Short circuit output current	IO(sc)	pin CANH, VCANH = -58 V pin CANL, VCANL = +58 V See Fig. 16	− 200		 200	mA mA
Common mode input resistance Deviation between common mode input resistance	RIN(CM) RIN(CM)(m)	− 12 V < VCANH, VCANL < + 12 V VCANH = VCANL	15 − 3	25	45 + 3	kΩ %
Differential input resistance	RIN(DIFF)	− 12 V < VCANH, VCANL < + 12 V	25	50	100	kΩ
SPLIT pin output voltage	Vsplit	Normal Mode See Fig. 17	2.4	2.5	2.6	V
Common mode input capacitance[1] (1Mbit/s data rate) Differential input capacitance[1] (1Mbit/s data rate)	$C_{IN(CM)}$ $C_{DIFF(CM)}$			20 10		pF pF

AC ELECTRICAL CHARACTERISTICS

V_{LOGIC} = V_{DD} = 5V. Operating temperature range (unless otherwise noted).

PARAMETER	SYMBOL	LIMITS			UNITS
		MIN	TYP	MAX	
SPI Interface Timing					
SCK clock frequency	f_{SCK}			20	MHz
SCK clock period	t_{CYC}	50			ns
$\overline{CS}$ active after last SCK rising edge	t_{CHH}	5			ns
$\overline{CS}$ setup time to first SCK rising edge	t_{CSS}	10			ns
$\overline{CS}$ hold time after last SCK falling edge	t_{CSH}	25			ns
$\overline{CS}$ inactive between SPI instructions	t_{CPH}	25			ns
SPI SI Data set-up time to SCK rising edge	t_{DS}	5			ns
SPI SI Data hold time after SCK rising edge	t_{DH}	5			ns
SCK rise time	t_{SCKR}			2	ns
SCK fall ime	t_{SCKF}			2	ns
SCK pulse width high	t_{SCKH}	25			ns
SCK pulse width low	t_{SCKL}	25			ns
SO valid after SCK falling edge	t_{DV}			25	ns
SO high-impedance after $\overline{CS}$ inactive	t_{CHZ}			25	ns
CAN Bus Data Rate					
Bit time	t_{Bit}	1		25	µs
Bit rate	f_{Bit}	40		1000	kHz
Time Out					
Permanent dominant time-out	$t_{dom(TXD)}$	0.3	2	6	ms
External Clock					
Maximum external clock frequency on OSCIN pin	$f_{OSC(max)}$			40	MHz

AC ELECTRICAL CHARACTERISTICS

V_{LOGIC} = 3.3V, V_{DD} = 5V. Operating temperature range (unless otherwise noted).

PARAMETER	SYMBOL	LIMITS			UNITS
		MIN	TYP	MAX	
SPI Interface Timing					
SCK clock frequency	f_{SCK}			16.66	MHz
SCK clock period	t_{CYC}	66			ns
$\overline{CS}$ active after last SCK rising edge	t_{CHH}	5			ns
$\overline{CS}$ setup time to first SCK rising edge	t_{CSS}	10			ns
$\overline{CS}$ hold time after last SCK falling edge	t_{CSH}	25			ns
$\overline{CS}$ inactive between SPI instructions	t_{CPH}	25			ns
SPI SI Data set-up time to SCK rising edge	t_{DS}	5			ns
SPI SI Data hold time after SCK rising edge	t_{DH}	5			ns
SCK rise time	t_{SCKR}			2	ns
SCK fall ime	t_{SCKF}			2	ns
SCK pulse width high	t_{SCKH}	25			ns
SCK pulse width low	t_{SCKL}	25			ns
SO valid after SCK falling edge	t_{DV}			30	ns
SO high-impedance after $\overline{CS}$ inactive	t_{CHZ}			25	ns
CAN Bus Data Rate					
Bit time	t_{Bit}	1		25	µs
Bit rate	f_{Bit}	40		1000	kHz
Time Out					
Permanent dominant time-out	$t_{dom(TXD)}$	0.3	2	6	ms
External Clock					
Maximum external clock frequency on OSCIN pin	$f_{OSC(max)}$			40	MHz

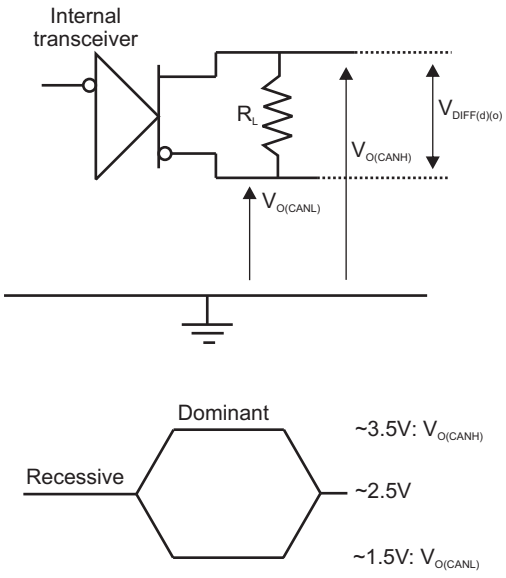


Figure 14. CAN Bus Driver Circuit

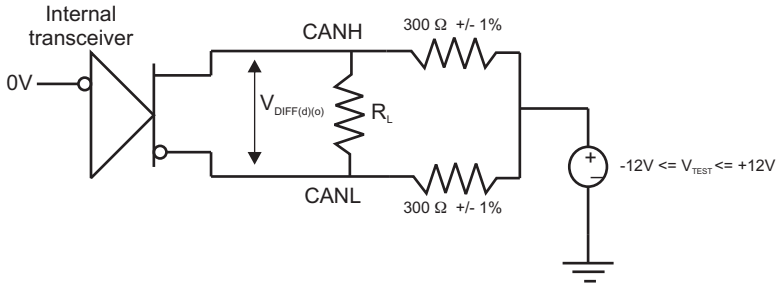


Figure 15. CAN Bus Driver (Dominant) Test Circuit

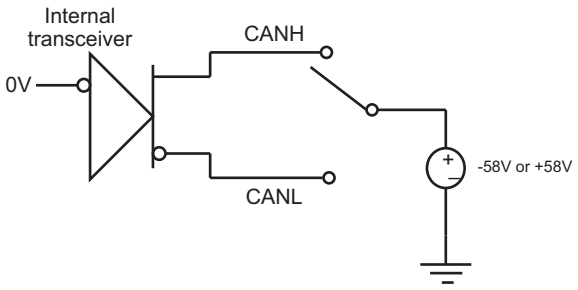


Figure 16. CAN Bus Driver Short-Circuit Test

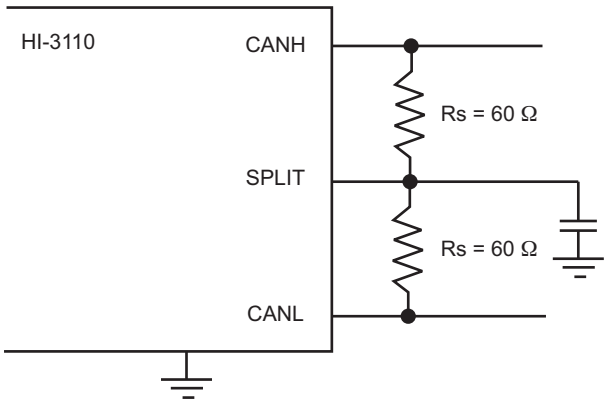


Figure 17. Split-Termination Connection

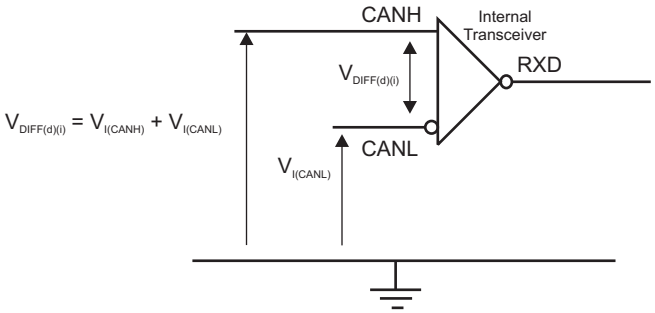
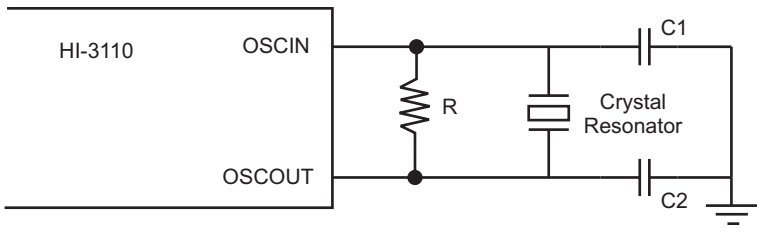


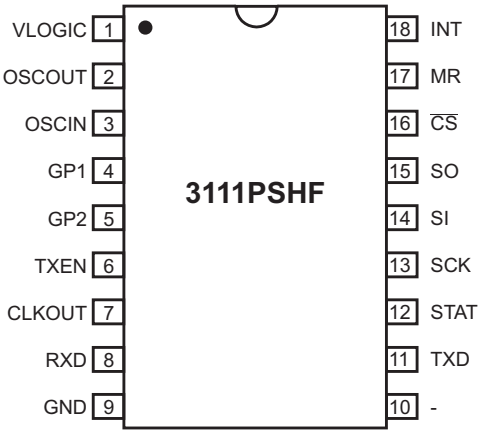
Figure 18. CAN Bus Receiver Common Mode Voltage Test



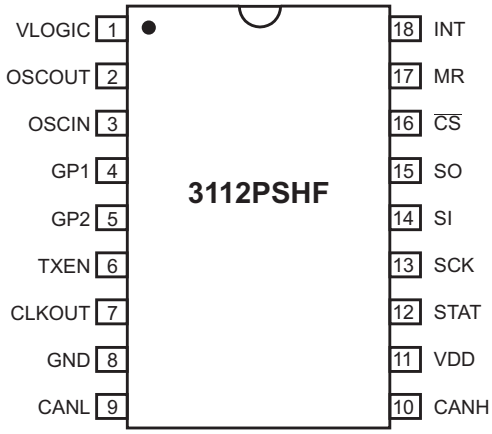
R = 1.5 MΩ, C1 = C2 = 10pF typ.

Figure 19. Suggested Crystal Oscillator Circuit

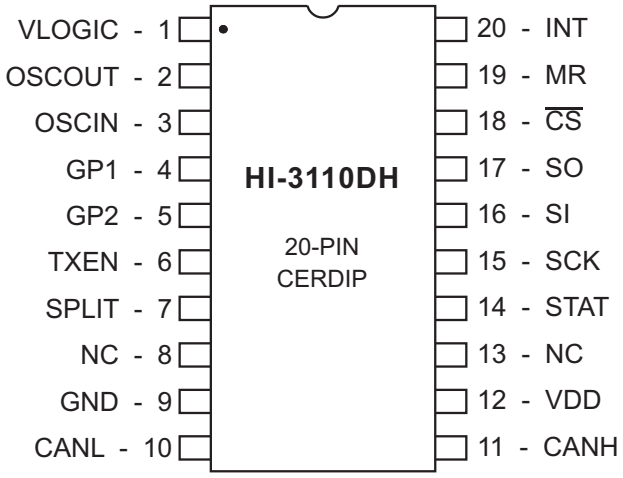
ADDITIONAL PACKAGE CONFIGURATIONS AND PINOUTS



18-Pin Plastic SOIC - WB Package



18-Pin Plastic SOIC - WB Package



20-Pin ceramic CERDIP

(NC = Not Connected)

ORDERING INFORMATION

HI - 311x PS H F

PART NUMBER	LEAD FINISH
F	100% Matte Tin (Pb-free, RoHS compliant)

PART NUMBER	PACKAGE DESCRIPTION
PS	18 PIN PLASTIC WIDE BODY SOIC (18HW): -55°C to +175°C

PART NUMBER	DESCRIPTION
3110	Integrated transceiver with SPLIT pin option
3111	Digital-only option, no transceiver
3112	Integrated transceiver with CLKOUT pin option

HI - 3110 D H

PART NUMBER	PACKAGE DESCRIPTION	LEAD FINISH
D	20 PIN CERDIP (20D): -55°C to +200°C	Tin / Lead (Sn / Pb) Solder

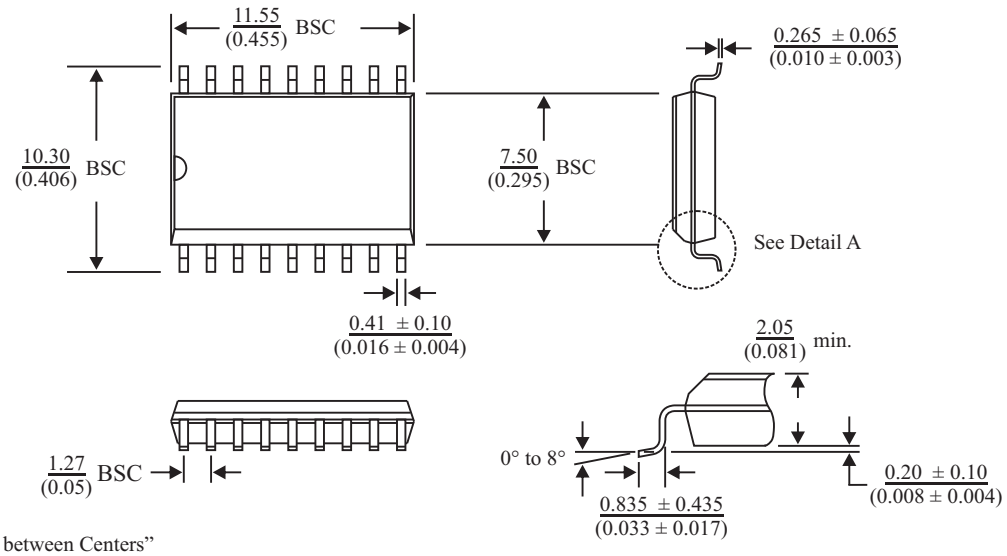
REVISION HISTORY

P/N	Rev	Date	Description of Change
HI-3110H	New	03/30/16	Initial Release.
	A	02/02/17	Include pin descriptions for HI-3111H variant. Update SPI clock frequency parameters for operation at VLOGIC = 3.3V and VLOGIC = 5V. Update block diagram to correct Time Tag Counter block connections.
	B	11/18/2021	Clarify operation of Rx FIFO when FIFO is full.
	C	03/17/2022	Add 200°C operating temperature capability and ceramic CERDIP-20 package.

18-PIN PLASTIC SMALL OUTLINE (SOIC) - WB
(Wide Body)

millimeters (inches)

Package Type: 18HW

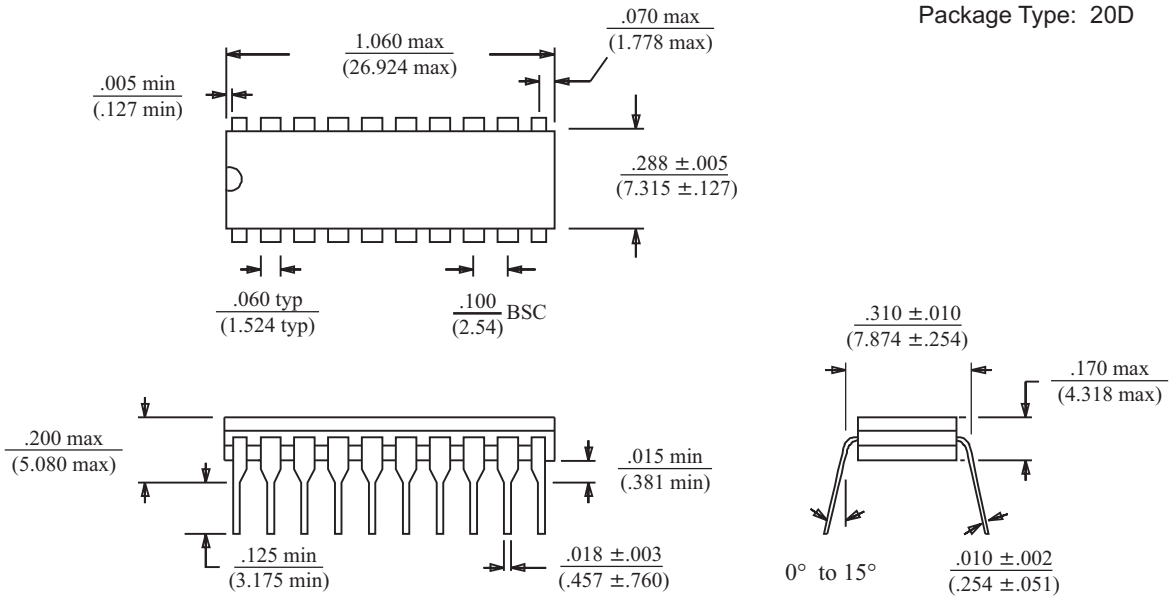


BSC = "Basic Spacing between Centers" is theoretical true position dimension and has no tolerance. (JEDEC Standard 95)

20-PIN CERDIP

inches (millimeters)

Package Type: 20D



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