Acromag®

Series IP560 Industrial I/O Pack
CAN Bus Interface Module

USER'S MANUAL

ACROMAG INCORPORATED
30765 South Wixom Road
P.O. BOX 437
Wixom, MI 48393-7037 U.S.A.
Tel: (248) 295-0310
Fax: (248) 624-9234

8500-887-C12A021

The information contained in this manual is subject to change without notice. Acromag, Inc. makes no warranty of any kind with regard to this material, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. Further, Acromag, Inc. assumes no responsibility for any errors that may appear in this manual and makes no commitment to update, or keep current, the information contained in this manual. No part of this manual may be copied or reproduced in any form, without the prior written consent of Acromag, Inc.

IMPORTANT SAFETY CONSIDERATIONS

It is very important for the user to consider the possible adverse effects of power, wiring, component, sensor, or software failures in designing any type of control or monitoring system. This is especially important where economic property loss or human life is involved. It is important that the user employ satisfactory overall system design. It is agreed between the Buyer and Acromag, that this is the Buyer's responsibility.

1 GENERAL INFORMATION

The Industrial I/O Pack (IP) Series IP560 module is a 16-bit, high density, single size IP, with the capability to interface to two Controller Area Network (CAN) 2.0B channels. The IP560 utilizes state of the art Surface Mounted Technology (SMT) to achieve its high channel density. Four units may be mounted on a 6U VME or CompactPCI carrier board to provide up to 8 CAN channels per bus system slot or 10 CAN channels per PCI bus (PC) system slot. The IP560 offers a variety of features which make it an ideal choice for many industrial and scientific applications as described below.

Model	Operating Temperature Range
IP560	0 to 70°C
IP560E	-40 to 85°C
IP560-I	0 to 70°C
IP560E-I	-40 to 85°C

1.1 KEY IP560 FEATURES

- **Two CAN channels**
- **1000 Volt isolation (IP560(E)-I) channel to channel and channel to host**
- **ISO 11898 compliance for standard (11-bit) and extended (29-bit) arbitration IDs**
- **CAN 2.0B protocol compatibility (extended frame passive in PCA82C200 compatibility mode)**
- **NXP SJA1000 CAN controllers**
- **NXP TJA1043 high speed CAN transceivers**
- **Bit rates up to 1 Mbits/s**
- **PCA82C200 mode (BasicCAN mode is default)**
- **Extended receive buffer (64-byte FIFO)**
- **Supports 11-bit identifier as well as 29-bit identifier**
- **24 MHz clock frequency**
- **PeliCAN mode extensions:**
  – Error counters with read/write access
  – Programmable error warning limit
  – Last error code register
  – Error interrupt for each CAN-bus error
  – Arbitration lost interrupt with detailed bit position
  – Single-shot transmission (no re-transmission)
  – Listen only mode (no acknowledge, no active error flags)
  – Hot plugging support (software driven bit rate detection)
  – Acceptance filter extension (4-byte code, 4-byte mask)
  – Reception of ‘own’ messages (self reception request)
- **Undervoltage detection on V_{BAT}**
- **Listen-only mode for node diagnosis and failure containment**

- **TXD dominant clamping handler with diagnosis**
- **RXD recessive clamping handler with diagnosis**
- **TXD-to-RXD short-circuit handler with diagnosis**
- **Bus line short-circuit diagnosis**
- **Bus dominant clamping diagnosis**

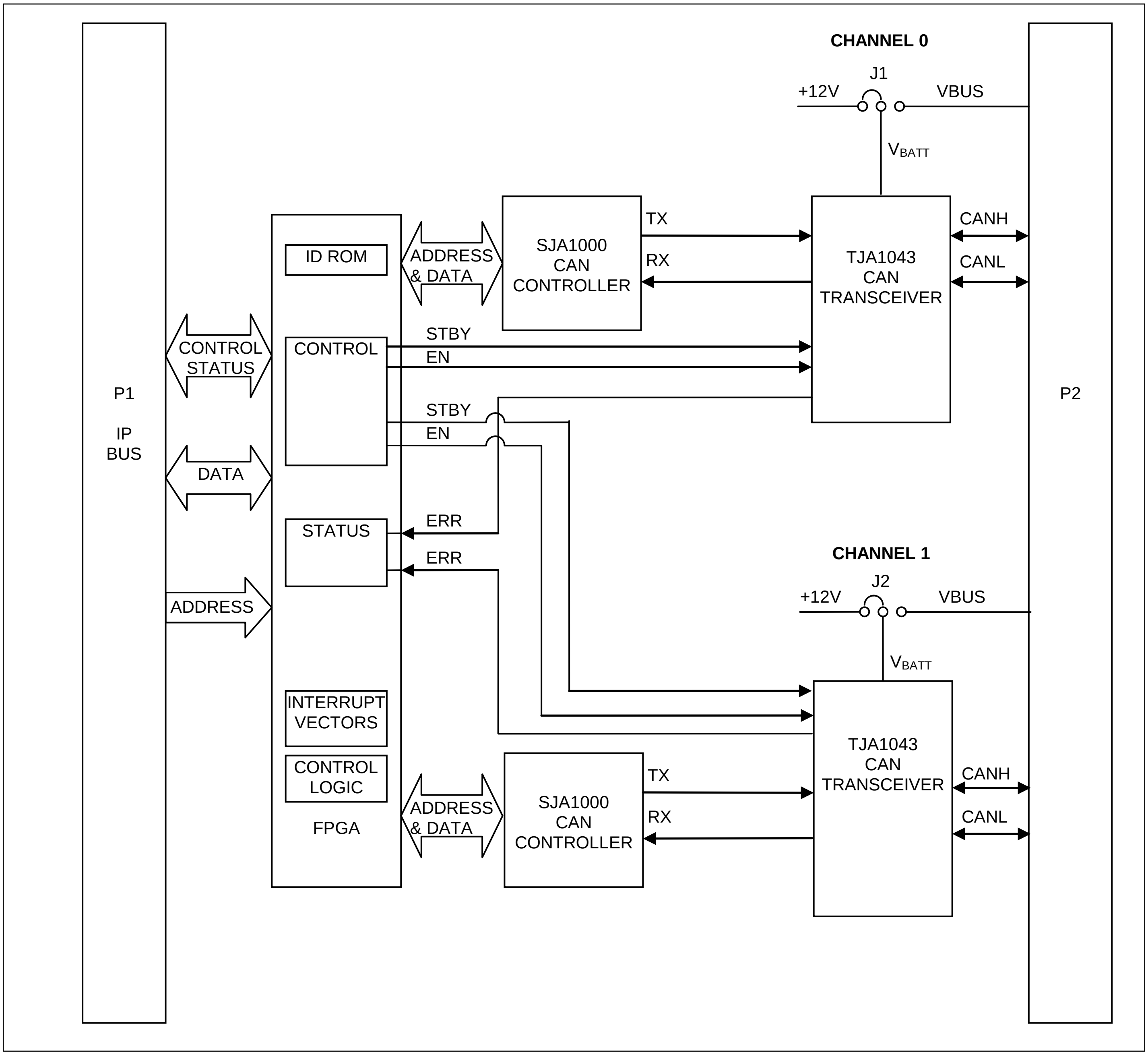


Figure 1-1 IP560(E) Block Diagram

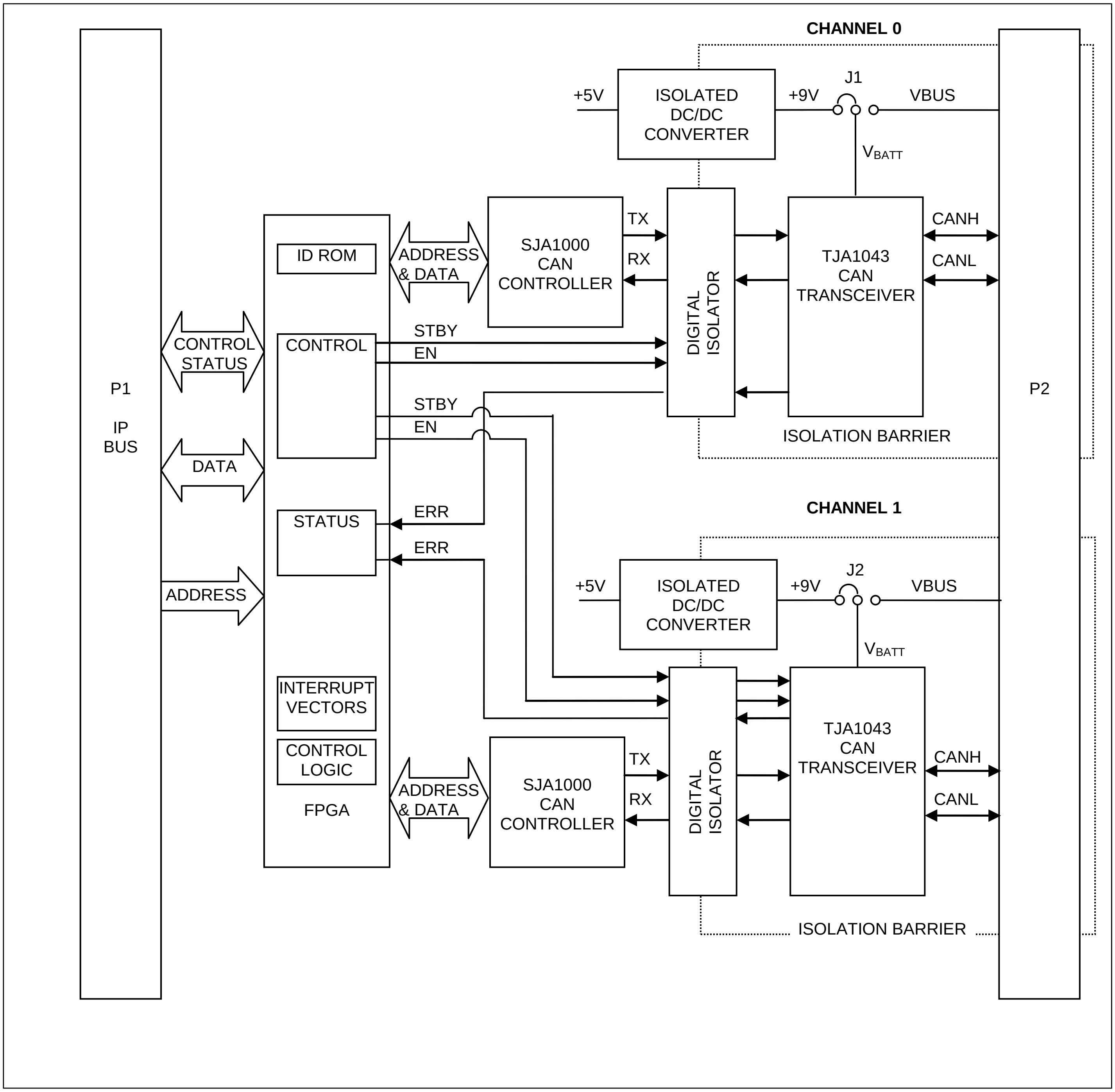


Figure 1-2 IP560(E)-I Block Diagram

1.2 INDUSTRIAL I/O PACK INTERFACE FEATURES

High density - Single-size, industry standard, IP module footprint. Four units mounted on a carrier board provide up to 8 channels in a single system slot. VMEbus and desktop PCI bus (PC) carriers are supported.

- **Local ID** - Each IP module has its own 8-bit ID signature which can be read via access to the ID space.
- **8-bit I/O** – Control and status register Read/Write is performed through data transfer cycles in the IP module I/O space. Access to the SJA1000 CAN controllers is performed in memory space.

1.3 INDUSTRIAL I/O PACK SOFTWARE LIBRARY

1.4 IP MODULE WINDOWS SOFTWARE

Acromag provides a software product (sold separately) to facilitate the development of Windows (2000/XP/Vista/7®) applications accessing Acromag Industry Pack models installed on Acromag PCI Carrier Cards. This software (Model IPSW-API-WIN) consists of low-level drivers and Windows 32 Dynamic Link Libraries (DLLs) that are compatible with a number of programming environments including Visual C++™, Visual Basic .NET® and others. The DLL functions provide a high-level interface to boards eliminating the need to perform low-level reads/writes of registers, and the writing of interrupt handlers.

1.4.1 IP MODULE VxWorks DRIVER SOFTWARE

Acromag provides a software product (sold separately) consisting of board VxWorks® software. This software (Model IPSW-API-VXW) is composed of VxWorks® (real time operating system) libraries for all Acromag IP modules and carriers including the AVME9670, AVME9600/9630, and APC8620A/21A . The software is implemented as a library of “C” functions which link with existing user code to make possible simple control of all Acromag PCI boards.

1.4.2 IP MODULE Linux SOFTWARE

Acromag provides a software product (sold separately) consisting of board Linux® software. This software (Model IPSW-API-LNX) is composed of Linux® (real time operating system) libraries for all Acromag IP modules and carriers including the APC8620A/21A and APC8640. The software supports X86 PCI bus only and is implemented as library of “C” functions. These functions link with existing user code to make possible simple control of all Acromag IP modules and carriers.

2 PREPARATION FOR USE

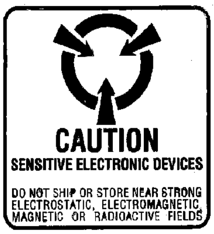
2.1 UNPACKING AND INSPECTION

Upon receipt of this product, inspect the shipping carton for evidence of mishandling during transit. If the shipping carton is badly damaged or water stained, request that the carrier's agent be present when the carton is opened. If the carrier's agent is absent when the carton is opened and the contents of the carton are damaged, keep the carton and packing material for the agent's inspection.

For repairs to a product damaged in shipment, refer to the Acromag Service Policy to obtain return instructions. It is suggested that salvageable shipping cartons and packing material be saved for future use in the event the product must be shipped.

This board is physically protected with packing material and electrically protected with an anti-static bag during shipment. However, it is recommended that the board be visually inspected for evidence of mishandling prior to applying power.

The board utilizes static sensitive components and should only be handled at a static-safe workstation.


CAUTION
SENSITIVE ELECTRONIC DEVICES
DO NOT SHIP OR STORE NEAR STRONG ELECTROSTATIC, ELECTROMAGNETIC, MAGNETIC OR RADIOACTIVE FIELDS

2.2 CARD CAGE CONSIDERATIONS

Refer to the specifications for loading and power requirements. Be sure that the system power supplies are able to accommodate the power requirements of the carrier board, plus the installed IP modules, within the voltage tolerances specified.

IMPORTANT: Adequate air circulation must be provided to prevent a temperature rise above the maximum operating temperature.

The dense packing of the IP modules to the carrier board restricts air flow within the card cage and is cause for concern. Adequate air circulation must be provided to prevent a temperature rise above the maximum operating temperature and to prolong the life of the electronics. If the installation is in an industrial environment and the board is exposed to environmental air, careful consideration should be given to air-filtering.

2.3 BOARD CONFIGURATION

The board may be configured differently, depending on the application. All possible jumper settings will be discussed in the following sections. The jumper locations are shown in the IP Mechanical Assembly Drawing located in the *Drawings* Section.

Remove power from the board when configuring hardware jumpers, installing IP modules, cables, termination panels, and field wiring. Refer to the IP560 Jumper Location drawing in section 7.1 and the following paragraphs for configuration and assembly instructions.

2.3.1 Power Monitor Hardware Jumper Configuration

The IP560 series modules provide the capability to monitor the CAN bus supply voltage (V_{BAT}) and report any under-voltage condition. This feature may be disabled by selecting an on-board power source to monitor. The selection of internal or external V_{BAT} power is accomplished via hardware jumpers J1 (channel 0) and J2 (channel 1) (see section 7.1). J1 and J2 select either the internal +12 Volt on-board supply or the external V_{BAT} supply sourced from the P2 connector. The configuration of the jumpers for the different supplies is shown in Table 2.1. "IN" means that the pins are shorted together with a shorting clip. "OUT" means that the clip has been removed. The jumper locations are shown in the IP560 Jumper Location drawing in section 7.1.

Table 2.1 Power Supply Selections (Pins of J1 and J2)

Power Supply Selection	**Jn (Pins 1&2)**	**Jn (Pins 3&4)**
+12 Volt on-board	IN	OUT
V_{BAT} (External, P2)	OUT	IN

2.3.2 Default Hardware Jumper Configuration

When the board is shipped from the factory, it is configured as follows:

- Internal +12 Volt power supplies are used for V_{BAT} monitoring

2.3.3 Software Configuration

Software configurable control registers are provided for control of transceiver mode and bus clock frequency. These control registers must be configured as desired before starting CAN communications. Refer to section 3 for programming details.

2.4 CONNECTORS

2.4.1 IP Field I/O Connector (P2)

P2 provides the field I/O interface connections for mating IP modules to the carrier board. P2 is a 50-pin female receptacle header which mates to the male connector of the carrier board. This provides excellent connection integrity and utilizes gold-plating in the mating area. Threaded metric M2 screws and spacers are supplied with the module to provide additional stability for harsh environments (see Mechanical Assembly Drawing 4501-434) near the end of this manual. The field and logic side connectors are keyed to avoid incorrect assembly.

P2 pin assignments are unique to each IP model (see Table 2.2) and normally correspond to the pin numbers of the field I/O interface connector on the carrier board (you should verify this for your carrier board).

Table 2.2 IP560 Field I/O Pin Connections (P2)

Pin Description	Number	Pin Description	Number
*N.C.	1	N.C.	26
VBAT_0	2	VBAT_1	27
N.C.	3	N.C.	28
N.C.	4	N.C.	29
GND_0	5	GND_1	30
CAN_H_0	6	CAN_H_1	31
CAN_L_0	7	CAN_L_1	32
GND_0	8	GND_1	33
N.C.	9	N.C.	34
N.C.	10	N.C.	35
N.C.	11	N.C.	36
N.C.	12	N.C.	37
N.C.	13	N.C.	38
N.C.	14	N.C.	39
N.C.	15	N.C.	40
N.C.	16	N.C.	41
N.C.	17	N.C.	42
N.C.	18	N.C.	43
N.C.	19	N.C.	44
N.C.	20	N.C.	45
N.C.	21	N.C.	46
N.C.	22	N.C.	47
N.C.	23	N.C.	48
N.C.	24	N.C.	49
N.C.	25	N.C.	50

***** Indicates pin not connected.

2.4.2 IP Logic Interface Connector (P1)

P1 of the IP module provides the logic interface to the mating connector on the carrier board. This connector is a 50-pin female receptacle header (AMP 173279-3 or equivalent) which mates to the male connector of the carrier board (AMP 173280-3 or equivalent). This provides excellent connection integrity and utilizes gold-plating in the mating area. Threaded metric M2 screws and spacers are supplied with the IP module to provide additional stability for harsh environments (see Drawing 4501-434 for assembly details). Field and logic side connectors are keyed to avoid incorrect assembly. The pin assignments of P1 are standard for all IP modules according to the Industrial I/O Pack Specification (see Table 2.3).

Table 2.3 Standard Logic Interface Connections (P1)

Pin Description	Number	Pin Description	Number
GND	1	GND	26
CLK	2	+5V	27
Reset*	3	R/W*	28
D00	4	IDSEL*	29
D01	5	***DMAReq0****	30
D02	6	MEMSEL*	31
D03	7	***DMAReq1****	32
D04	8	IntSel*	33
D05	9	***DMAck0****	34
D06	10	IOSEL*	35
D07	11	RESERVED	36
D08	12	A1	37
D09	13	***DMAEnd****	38
D10	14	A2	39
D11	15	***ERROR****	40
D12	16	A3	41
D13	17	INTReq0*	42
D14	18	A4	43
D15	19	INTReq1*	44
BS0*	20	A5	45
BS1*	21	***STROBE****	46
-12V	22	A6	47
+12V	23	ACK*	48
+5V	24	RESERVED	49
GND	25	GND	50

An Asterisk (*) is used to indicate an active-low signal.
BOLD ITALIC Logic Lines are NOT USED by this IP Model.

3 PROGRAMMING INFORMATION

3.1 IP IDENTIFICATION PROM

Each IP module contains identification (ID) information that resides in the ID space per the IP module specification. This area of memory contains 32 bytes of information at most. Both fixed and variable information may be present within the ID space. Fixed information includes the "IPAH" identifier, model number, and manufacturer's identification codes. Variable information includes unique information required for the module. The IP560 ID information does not contain any variable (e.g. unique calibration) information. ID space bytes are addressed using only the odd addresses in a 64 byte block (on the “Big Endian” VMEbus). Even addresses are used on the “Little Endian” PC ISA bus. The IP560 ID space contents are shown in Table 3.1. Note that the base-address for the IP module ID space (see your carrier board instructions) must be added to the addresses shown to properly access the ID space. Execution of an ID space read requires 1 wait state.

Table 3.1 IP560 ID Space Identification (ID) PROM

Hex Offset From ID PROM Base Address	ASCII Character Equivalent	Numeric Value (Hex)	Field Description
01	I	49	Format I ID ROM supports 8 or 32 MHz IP bus clock frequency
03	P	50	
05	A	41	
07	H	48	
09		A3	Acromag ID Code
0B			IP Model Code[1]
		50	IP560
		51	IP560-I
0D		00	Not Used (Revision)
0F		00	Reserved
11		00	Not Used (Driver ID Low Byte)
13		00	Not Used (Driver ID High Byte)
15		0C	Total Number of ID PROM Bytes
17			CRC
		6C	IP560
		0D	IP560-I
19 to 3F		yy	Not Used

Notes (Table 3.1):

1. The IP model number is represented by a two-digit code within the ID space (the IP560 model is represented by 50H, the IP560-I is represented by 51H).

3.2 I/O SPACE ADDRESS MAP

This board is addressable in the Industrial Pack I/O space to provide access to control registers, a status register, and interrupt vector registers.

The I/O space may be as large as 64, 16-bit words (128 bytes) using address lines A1 to A6, but the IP560 uses only a portion of this space. The I/O space address map for the IP560 is shown in Table 3.2. Note that the base address for the IP module I/O space (see your carrier board instructions) must be added to the addresses shown to properly access the I/O space.

Table 3.2 IP560 I/O Space Address (Hex) Memory Map[1,2]

Base Addr+	MSB D15 D08	LSB D07 D00	Base Addr+
00	Not Used	Control Register 0	01
02	Not Used	Control Register 1	03
04	Not Used	Status Register	05
06	Not Used	Interrupt Vector 0	07
08	Not Used	Interrupt Vector 1	09

Notes (Table 3.2):

1. All addresses that are "Not Used" will read as zero.
2. All Reads are one wait state. All writes are zero wait state.

3.2.1 Control Register 0

This read/write register is used to select the transceiver mode and enable/disable interrupts for each CAN channel.

The function of each of the control register bits is described in Table 3.3. A power-up or system reset sets all control register bits to 0.

Table 3.3 Control Register 0

BIT	FUNCTION
0	Not Used[1]
1	STB Channel 0[2] CAN transceiver standby Channel 0 0 = Standby 1 = Active
2	EN Channel 0[2] CAN transceiver enable Channel 0 0 = Disabled 1 = Enabled
3	Channel 0 interrupt enable 0 = Disabled 1 = Enabled
4	Not Used[1]
5	STB Channel 1[2] CAN transceiver standby Channel 1 0 = Standby 1 = Active
6	EN Channel 1[2] CAN transceiver enable Channel 1 0 = Disabled 1 = Enabled
7	Channel 1 interrupt enable 0 = Disabled 1 = Enabled

Notes (Table 3.3):

1. All bits labeled “Not Used” will return the last value written on a read access.
2. See section 3.7 for further description.

3.2.2 Control Register 1

This read/write register is used to select the IP bus clock frequency and enable the clock for the CAN controllers.

The function of each of the control register bits is described in Table 3.4. A power-up or system reset sets all control register bits to 0.

Table 3.4 Control Register 1

BIT	FUNCTION
0	IP bus clock 0
1	IP bus clock 1
2 - 7	Not Used[1]

Notes (Table 3.4):

1. All bits labeled “Not Used” will return the last value written on a read access.

IP bus clock: A phase locked loop is used to generate the module clock from the IP bus clock. The IP bus clock must remain constant after setting the appropriate bits in this register. Set the IP bus clock frequency on the carrier board prior to writing these bits. The module clock is not enabled until the IP bus clock frequency is specified by writing the following bit pattern:

Table 3.5 IP bus clock selection

IP bus clock 1	IP bus clock 0	FUNCTION
0	0	Module clock disabled
0	1	IP bus clock 8 MHz, enable module clock
1	X[1]	IP bus clock 32 MHz, enable module clock

Notes (Table 3.5):

1. All bits labeled “X” can be 0 or 1.

If the IP bus clock frequency on the carrier board must be changed after setting the IP bus clock selection bits in the IP560 Control Register 1, then the IP560 must be reset and the IP bus clock selection bits must be set for the new IP bus clock frequency.

3.2.3 Status Register

This read only register is used to indicate interrupt and transceiver status for each of the CAN channels.

The function of each of the status register bits is described in Table 3.6. A power-up or system reset sets all control register bits to 0.

Table 3.6 Status Register

BIT	FUNCTION
0	Interrupt Pending Channel 0 0 = No interrupt pending 1 = Interrupt Pending
1	Transceiver Error Channel 0 The error output pin from the TJA1043 CAN transceiver is connected to this status bit 0 = No error 1 = Error
2	Interrupt Pending Channel 1 0 = No interrupt pending 1 = Interrupt Pending
3	Transceiver Error Channel 1 The error output pin from the TJA1043 CAN transceiver is connected to this status bit 0 = No error 1 = Error
4	Module clock ready[2] 0 = not ready 1 = ready
5 - 7	Not Used[1]

Notes (Table 3.6):

1. All bits labeled “Not Used” will return zero on a read access.
2. Module clock ready may take up to 5 mS to indicate ready after setting the appropriate clock enable bit.

3.2.4 Interrupt Vector Register 0

The Interrupt Vector Register 0 can be written with an 8-bit interrupt vector. This vector is provided to the carrier and system bus upon an active INTSEL* cycle. The register contents are cleared upon reset.

Interrupt Vector Register 0							
MSB							LSB
07	06	05	04	03	02	01	00

Interrupt 0 is released by clearing the interrupt condition of the Channel 0 SJA1000 CAN controller.

3.2.5 Interrupt Vector Register 1

The Interrupt Vector Register 1 can be written with an 8-bit interrupt vector. This vector is provided to the carrier and system bus upon an active INTSEL* cycle. The register contents are cleared upon reset.

Interrupt Vector Register 1							
MSB							**LSB**
07	06	05	04	03	02	01	00

Interrupt 1 is released by clearing the interrupt condition of the Channel 1 SJA1000 CAN controller.

3.3 MEMORY SPACE ADDRESS MAP

This board is addressable in the Industrial Pack memory space to access the SJA1000 CAN controllers. The IP560 uses the lower 512 words of this memory space. The SJA1000 controller has two modes of operation: BasicCAN and PeliCAN. The memory maps differ depending on the mode selected. The memory space address map for the IP560 in BasicCAN mode is shown in Table 3.7, PeliCAN mode is shown in Table 3.16. Note that the base address for the IP module memory space (see your carrier board instructions) must be added to the addresses shown to properly access the memory space. 8-bit and 16-bit accesses to the registers in memory space are permitted, but the upper 8 bits of the 16-bit access are ignored on writes and read as zeros.

This memory map reflects byte accesses using the “Big Endian” byte ordering format. Big Endian is the convention used in the Motorola 68000 microprocessor family and is the VMEbus convention. In Big Endian, the lower-order byte is stored at odd-byte addresses. The Intel x86 family of microprocessors uses the opposite convention, or “Little Endian” byte ordering. Little Endian uses even-byte addresses to store the low-order byte. As such, installation of this module on a PC carrier board will require the use of the even address locations to access the lower 8-bits of data while on a VMEbus carrier use of odd address locations are required to access the lower 8-bit data.

3.4 Detailed description of the CAN controller

The SJA1000 supports two different modes of operation:

- BasicCAN mode
- PeliCAN mode; with extended features.

The mode of operation is selected with the CAN-mode bit located within the clock divider register (see section 3.7.3). The default mode upon reset is the BasicCAN mode.

3.4.1 CAN 2.0B

The SJA1000 is designed to support the full CAN 2.0B protocol specification, which means that the extended oscillator tolerance is implemented as well as the processing of extended frame messages. In BasicCAN mode it is possible to transmit and receive standard frame messages only (11-bit identifier). If extended frame messages (29-bit identifier) are detected on the CAN-bus, they are tolerated and an acknowledge is given if the message was correct, but there is no receive interrupt generated (i.e. the message is discarded).

3.4.2 DIFFERENCES BETWEEN BasicCAN AND PeliCAN MODE

In the PeliCAN mode the SJA1000 appears with a reorganized register mapping and additional features. In the PeliCAN mode the complete CAN 2.0B functionality is supported (29-bit identifier).

PeliCAN mode provides the following additional features:

- Reception and transmission of standard and extended frame format messages
- Receive FIFO (64-byte)
- Single/dual acceptance filter with mask and code register for standard and extended frame
- Error counters with read/write access
- Programmable error warning limit
- Last error code register
- Error interrupt for each CAN-bus error
- Arbitration lost interrupt with detailed bit position
- Single-shot transmission (no re-transmission on error or arbitration lost)
- Listen only mode (monitoring of the CAN-bus, no acknowledge, no error flags)
- Hot plugging supported (disturbance-free software driven bit rate detection)
- Disable CLKOUT by hardware.

3.5 BasicCAN mode

The address area of the SJA1000 consists of the control segment and the message buffers. The control segment is programmed during an initialization download in order to configure communication parameters (e.g. bit timing). Communication over the CAN-bus is also controlled via this segment by the processor.

A message, which should be transmitted, has to be written to the transmit buffer. After a successful reception the host processor may read the received message from the receive buffer and then release it for further use.

The exchange of status, control and command signals between the processor and the SJA1000 is performed in the control segment. The layout of this segment is shown in Table 3.7. After the initial download, the contents of the register’s acceptance code, acceptance mask, bus timing registers 0 and 1 and output control should not be changed. Therefore these registers may only be accessed when the reset request bit in the control register is set HIGH.

For register access, two different modes have to be distinguished:

- Reset mode
- Operating mode.

The reset mode (see Table 3.9, control register, bit Reset Request) is entered automatically after a hardware reset or when the controller enters the bus-off state (see Table 3.11, status register, bit Bus Status). The operating mode is activated by setting the reset request bit in the control register to zero.

Table 3.7 IP560 Memory Space Address (Hex) Memory Map BasicCAN see note 1

Hex Offset From Base Addr	MSB D15 D08	LSB D07 D00				Hex Offset From Base Addr
		OPERATING MODE		RESET MODE		
		READ	WRITE	READ	WRITE	
CHANNEL 0						
00	Not Used	control	control	control	control	01
02	Not Used	(FFH)	command	(FFH)	command	03
04	Not Used	status	-	status	-	05
06	Not Used	interrupt	-	interrupt	-	07
08	Not Used	(FFH)	-	acceptance code	acceptance code	09
0A	Not Used	(FFH)	-	acceptance mask	acceptance mask	0B
0C	Not Used	(FFH)	-	bus timing 0	bus timing 0	0D
0E	Not Used	(FFH)	-	bus timing 1	bus timing 1	0F
10	Not Used	(FFH)	-	output control	output control	11
12	Not Used	test[2]	test[2]	test[2]	test[2]	13
14	Not Used	TX identifier (10 to 3)	TX identifier (10 to 3)	(FFH)	-	15
16	Not Used	TX identifier (2 to 0), RTR and DLC	TX identifier (2 to 0), RTR and DLC	(FFH)	-	17
18	Not Used	TX buffer data byte 1	TX buffer data byte 1	(FFH)	-	19
1A	Not Used	TX buffer data byte 2	TX buffer data byte 2	(FFH)	-	1B
1C	Not Used	TX buffer data byte 3	TX buffer data byte 3	(FFH)	-	1D
1E	Not Used	TX buffer data byte 4	TX buffer data byte 4	(FFH)	-	1F
20	Not Used	TX buffer data byte 5	TX buffer data byte 5	(FFH)	-	21
22	Not Used	TX buffer data byte 6	TX buffer data byte 6	(FFH)	-	23
24	Not Used	TX buffer data byte 7	TX buffer data byte 7	(FFH)	-	25
26	Not Used	TX buffer data byte 8	TX buffer data byte 8	(FFH)	-	27
28	Not Used	RX identifier (10 to 3)	RX identifier (10 to 3)	RX identifier (10 to 3)	RX identifier (10 to 3)	29
2A	Not Used	RX identifier (2 to 0), RTR and DLC	RX identifier (2 to 0), RTR and DLC	RX identifier (2 to 0), RTR and DLC	RX identifier (2 to 0), RTR and DLC	2B
2C	Not Used	RX buffer data byte 1	RX buffer data byte 1	RX buffer data byte 1	RX buffer data byte 1	2D
2E	Not Used	RX buffer data byte 2	RX buffer data byte 2	RX buffer data byte 2	RX buffer data byte 2	2F
30	Not Used	RX buffer data byte 3	RX buffer data byte 3	RX buffer data byte 3	RX buffer data byte 3	31
32	Not Used	RX buffer data byte 4	RX buffer data byte 4	RX buffer data byte 4	RX buffer data byte 4	33
34	Not Used	RX buffer data byte 5	RX buffer data byte 5	RX buffer data byte 5	RX buffer data byte 5	35
36	Not Used	RX buffer data byte 6	RX buffer data byte 6	RX buffer data byte 6	RX buffer data byte 6	37
38	Not Used	RX buffer data byte 7	RX buffer data byte 7	RX buffer data byte 7	RX buffer data byte 7	39
3A	Not Used	RX buffer data byte 8	RX buffer data byte 8	RX buffer data byte 8	RX buffer data byte 8	3B
3C	Not Used	(FFH)	-	(FFH)	-	3D
3E	Not Used	Clock divider[3]	Clock divider[3]	Clock divider[3]	Clock divider[3]	3F

Hex Offset From Base Addr	MSB D15 D08	LSB D07 D00				Hex Offset From Base Addr
		OPERATING MODE		RESET MODE		
		READ	WRITE	READ	WRITE	
Channel 1						
200	Not Used	control	control	control	control	201
202	Not Used	(FFH)	command	(FFH)	command	203
204	Not Used	status	-	status	-	205
206	Not Used	interrupt	-	interrupt	-	207
208	Not Used	(FFH)	-	acceptance code	acceptance code	209
20A	Not Used	(FFH)	-	acceptance mask	acceptance mask	20B
20C	Not Used	(FFH)	-	bus timing 0	bus timing 0	20D
20E	Not Used	(FFH)	-	bus timing 1	bus timing 1	20F
210	Not Used	(FFH)	-	output control	output control	211
212	Not Used	test[2]	test[2]	test[2]	test[2]	213
214	Not Used	TX identifier (10 to 3)	TX identifier (10 to 3)	(FFH)	-	215
216	Not Used	TX identifier (2 to 0), RTR and DLC	TX identifier (2 to 0), RTR and DLC	(FFH)	-	217
218	Not Used	TX buffer data byte 1	TX buffer data byte 1	(FFH)	-	219
21A	Not Used	TX buffer data byte 2	TX buffer data byte 2	(FFH)	-	21B
21C	Not Used	TX buffer data byte 3	TX buffer data byte 3	(FFH)	-	21D
21E	Not Used	TX buffer data byte 4	TX buffer data byte 4	(FFH)	-	21F
220	Not Used	TX buffer data byte 5	TX buffer data byte 5	(FFH)	-	221
222	Not Used	TX buffer data byte 6	TX buffer data byte 6	(FFH)	-	223
224	Not Used	TX buffer data byte 7	TX buffer data byte 7	(FFH)	-	225
226	Not Used	TX buffer data byte 8	TX buffer data byte 8	(FFH)	-	227
228	Not Used	RX identifier (10 to 3)	RX identifier (10 to 3)	RX identifier (10 to 3)	RX identifier (10 to 3)	229
22A	Not Used	RX identifier (2 to 0), RTR and DLC	RX identifier (2 to 0), RTR and DLC	RX identifier (2 to 0), RTR and DLC	RX identifier (2 to 0), RTR and DLC	22B
22C	Not Used	RX buffer data byte 1	RX buffer data byte 1	RX buffer data byte 1	RX buffer data byte 1	22D
22E	Not Used	RX buffer data byte 2	RX buffer data byte 2	RX buffer data byte 2	RX buffer data byte 2	22F
230	Not Used	RX buffer data byte 3	RX buffer data byte 3	RX buffer data byte 3	RX buffer data byte 3	231
232	Not Used	RX buffer data byte 4	RX buffer data byte 4	RX buffer data byte 4	RX buffer data byte 4	233
234	Not Used	RX buffer data byte 5	RX buffer data byte 5	RX buffer data byte 5	RX buffer data byte 5	235
236	Not Used	RX buffer data byte 6	RX buffer data byte 6	RX buffer data byte 6	RX buffer data byte 6	237
238	Not Used	RX buffer data byte 7	RX buffer data byte 7	RX buffer data byte 7	RX buffer data byte 7	239
23A	Not Used	RX buffer data byte 8	RX buffer data byte 8	RX buffer data byte 8	RX buffer data byte 8	23B
23C	Not Used	(FFH)	-	(FFH)	-	23D
23E	Not Used	Clock divider[3]	Clock divider[3]	Clock divider[3]	Clock divider[3]	23F

Notes:(Table 3.7):

1. Registers are repeated within higher CAN address areas (the most significant bits of the lower 8-bit portion of the address are not decoded: address 4016 maps to address 0 and address 14016 maps to 10016)

2. Test register is used for production testing only. Using this register during normal operation may result in undesired device behavior.
3. Some bits are writeable in reset mode only (CAN mode, CBP, and clock off)

3.5.1 Reset Values

Detection of a ‘reset request’ results in aborting the current transmission/reception of a message and entering the reset mode. On the ‘1-to-0’ transition of the reset request bit, the CAN controller returns to the operating mode.

Table 3.8 Reset mode configuration; notes 1 and 2

REGISTER	BIT	SYMBOL	NAME	VALUE	
				RESET BY HARDWARE	SETTING BIT CR.0 BY SOFTWARE OR DUE TO BUS-OFF
Control	CR.7	-	reserved	0	0
	CR.6	-	reserved	X	X
	CR.5	-	reserved	1	1
	CR.4	OIE	Overrun Interrupt Enable	X	X
	CR.3	EIE	Error Interrupt Enable	X	X
	CR.2	TIE	Transmit Interrupt Enable	X	X
	CR.1	RIE	Receive Interrupt Enable	X	X
	CR.0	RR	Reset Request	1 (reset mode)	1 (reset mode)
Command	CMR.7	-	reserved	note 3	note 3
	CMR.6	-	reserved		
	CMR.5	-	reserved		
	CMR.4	GTS	Go To Sleep		
	CMR.3	CDO	Clear Data Overrun		
	CMR.2	RRB	Release Receive Buffer		
	CMR.1	AT	Abort Transmission		
	CMR.0	TR	Transmission Request		
Status	SR.7	BS	Bus Status	0 (bus-on)	X
	SR.6	ES	Error Status	0 (ok)	X
	SR.5	TS	Transmit Status	0 (idle)	0 (idle)
	SR.4	RS	Receive Status	0 (idle)	0 (idle)
	SR.3	TCS	Transmission Complete Status	1 (complete)	X
	SR.2	TBS	Transmit Buffer Status	1 (released)	1 (released)

REGISTER	**BIT**	**SYMBOL**	**NAME**	**VALUE**	
				RESET BY HARDWARE	**SETTING BIT CR.0 BY SOFTWARE OR DUE TO BUS-OFF**
	SR.1	DOS	Data Overrun Status	0 (absent)	0 (absent)
	SR.0	RBS	Receive Buffer Status	0 (empty)	0 (empty)
Interrupt	IR.7	-	reserved	1	1
	IR.6	-	reserved	1	1
	IR.5	-	reserved	1	1
	IR.4	WUI	Wake-Up Interrupt	0 (reset)	0 (reset)
	IR.3	DOI	Data Overrun Interrupt	0 (reset)	0 (reset)
	IR.2	EI	Error Interrupt	0 (reset)	X; note 4
	IR.1	TI	Transmit Interrupt	0 (reset)	O (reset)
	IR.0	RI	Receive Interrupt	0 (reset)	O (reset)
Acceptance code	AC.7 to 0	AC	Acceptance Code	X	X
Acceptance mask	AM.7 to 0	AM	Acceptance Mask	X	X

REGISTER	BIT	SYMBOL	NAME	VALUE	
				RESET BY HARDWARE	SETTING BIT CR.0 BY SOFTWARE OR DUE TO BUS-OFF
Bus timing 0	BTR0.7	SJW.1	Synchronization Jump Width 1	X	X
	BTR0.6	SJW.0	Synchronization Jump Width 0	X	X
	BTR0.5	BRP.5	Baud Rate Prescaler 5	X	X
	BTR0.4	BRP.4	Baud Rate Prescaler 4	X	X
	BTR0.3	BRP.3	Baud Rate Prescaler 3	X	X
	BTR0.2	BRP.2	Baud Rate Prescaler 2	X	X
	BTR0.1	BRP.1	Baud Rate Prescaler 1	X	X
	BTR0.0	BRP.0	Baud Rate Prescaler 0	X	X
Bus timing 1	BTR1.7	SAM	Sampling	X	X
	BTR1.6	TSEG2.2	Time Segment 2.2	X	X
	BTR1.5	TSEG2.1	Time Segment 2.1	X	X
	BTR1.4	TSEG2.0	Time Segment 2.0	X	X
	BTR1.3	TSEG1.3	Time Segment 1.3	X	X
	BTR1.2	TSEG1.2	Time Segment 1.2	X	X
	BTR1.1	TSEG1.1	Time Segment 1.1	X	X
	BTR1.0	TSEG1.0	Time Segment 1.0	X	X

REGISTER	BIT	SYMBOL	NAME	VALUE	
				RESET BY HARDWARE	SETTING BIT CR.0 BY SOFTWARE OR DUE TO BUS-OFF
Output control	OC.7	OCTP1	Output Control Transistor P1	X	X
	OC.6	OCTN1	Output Control Transistor N1	X	X
	OC.5	OCPOL1	Output Control Polarity 1	X	X
	OC.4	OCTP0	Output Control Transistor P0	X	X
	OC.3	OCTN0	Output Control Transistor N0	X	X
	OC.2	OCPOL0	Output Control Polarity 0	X	X
	OC.1	OCMODE1	Output Control Mode 1	X	X
	OC.0	OCMODE0	Output Control Mode 0	X	X
Transmit buffer	-	TXB	Transmit Buffer	X	X
Receive buffer	-	RXB	Receive Buffer	X	X
Clock divider	-	CDR	Clock Divider Register	00000000	X

Notes (Table 3.8):

1. X means that the value of these registers or bits is not influenced.
2. Remarks in brackets explain functional meaning.
3. Reading the command register will always reflect a binary ‘11111111’.
4. On bus-off the error interrupt is set, if enabled.
5. Internal read/write pointers of the RXFIFO are reset to their initial values. A subsequent read access to the RXB would show undefined data values (parts of old messages). If a message is transmitted, this message is written in parallel to the receive buffer but no receive interrupt is generated and the receive buffer area is not locked. So, even if the receive buffer is empty, the last transmitted message may be read from the receive buffer until it is overridden by the next received or transmitted message.
   Upon a hardware reset, the RXFIFO pointers are reset to the physical RAM address ‘0’. Setting CR.0 by software or due to the bus-off event will reset the RXFIFO pointers to the currently valid FIFO start address which is different from the RAM address ‘0’ after the first release receive buffer command.

3.5.2 Control Register (CR)

The contents of the control register are used to change the behavior of the CAN controller. Bits may be set or reset by the host CPU which uses the control register as a read/write memory.

Table 3.9 Bit interpretation of the control register (CR); base address + 0H

BIT	SYMBOL	NAME	VALUE	FUNCTION
CR.7	-	-	-	reserved; note 1
CR.6	-	-	-	reserved; note 2
CR.5	-	-	-	reserved; note 3
CR.4	OIE	Overrun Interrupt Enable	1	enabled; if the data overrun bit is set, the CPU receives an overrun interrupt signal (see also status register; Table 3.11)
			0	disabled; the CPU receives no overrun interrupt signal from the SJA1000
CR.3	EIE	Error Interrupt Enable	1	enabled; if the error or bus status change, the CPU receives an error interrupt signal (see also status register; Table 3.11)
			0	disabled; the CPU receives no error interrupt signal from the SJA1000
CR.2	TIE	Transmit Interrupt Enable	1	enabled; when a message has been successfully transmitted or the transmit buffer is accessible again, (e.g. after an abort transmission command) the SJA1000 transmits a transmit interrupt signal to the CPU
			0	disabled; the CPU receives no transmit interrupt signal from the SJA1000
CR.1	RIE	Receive Interrupt Enable	1	enabled; when a message has been received without errors, the SJA1000 transmits a receive interrupt signal to the CPU
			0	disabled; the CPU receives no transmit interrupt signal from the SJA1000
CR.0	RR	Reset Request; note 4	1	present; detection of a reset request results in aborting the current transmission/reception of a message and entering the reset mode
			0	absent; on the ‘1-to-0’ transition of the reset request bit, the SJA1000 returns to the operating mode

Notes (Table 3.9):

1. Any write access to the control register has to set this bit to logic 0 (reset value is logic 0).
2. In the PCA82C200 this bit was used to select the synchronization mode. Because this mode is no longer implemented, setting this bit has no influence on the CPU. Due to software compatibility setting this bit is allowed. This bit will not change

after hardware or software reset. In addition the value written by users software is reflected.

3. Reading this bit will always reflect a logic 1.
4. During a hardware reset or when the bus status bit is set to logic 1 (bus-off), the reset request bit is set to logic 1 (present). If this bit is accessed by software, a value change will become visible and takes effect first with the next positive edge of the internal clock which operates with 1⁄2 of the external oscillator frequency. During an external reset the CPU cannot set the reset request bit to logic 0 (absent). Therefore, after having set the reset request bit to logic 0, the CPU must check this bit to ensure that the external reset pin is not being held LOW. Changes of the reset request bit are synchronized with the internal divided clock. Reading the reset request bit reflects the synchronized status. After the reset request bit is set to logic 0 the SJA1000 will wait for:
   a) One occurrence of bus-free signal (11 recessive bits), if the preceding reset request has been caused by a hardware reset or a CPU-initiated reset.
   b) 128 occurrences of bus-free, if the preceding reset request has been caused by a CAN controller initiated bus-off, before re-entering the bus-on mode; it should be noted that several registers are modified if the reset request bit was set (see also Table 3.8).

3.5.3 Command Register (CMR)

A command bit initiates an action within the transfer layer of the SJA1000. The command register appears to the processor as a write only memory. If a read access is performed to this address the byte ‘11111111’ is returned.

Table 3.10 Bit interpretation of the command register (CMR); base address + 03H

BIT	SYMBOL	NAME	VALUE	FUNCTION
CMR.7	-	-	-	reserved
CMR.6	-	-	-	reserved
CMR.5	-	-	-	reserved
CMR.4	GTS	Go To Sleep; note 1	1	sleep; the SJA1000 enters sleep mode if no CAN interrupt is pending and there is no bus activity
			0	wake up; SJA1000 operates normal
CMR.3	CDO	Clear Data Overrun; note 2	1	clear; data overrun status bit is cleared
			0	no action
CMR.2	RRB	Release Receive Buffer; note 3	1	released; the receive buffer, representing the message memory space in the RXFIFO is released
			0	no action
CMR.1	AT	Abort Transmission; note 4	1	present; if not already in progress, a pending transmission request is cancelled
			0	absent; no action
CMR.0	TR	Transmission Request; note 5	1	present; a message will be transmitted
			0	absent; no action

Notes (Table 3.10):

1. The SJA1000 will enter sleep mode if the sleep bit is set to logic 1 (sleep); there is no bus activity and no interrupt is pending. Setting of GTS with at least one of the previously mentioned exceptions valid will result in a wake-up interrupt. The SJA1000 will wake up when one of the three previously mentioned conditions is negated: after ‘Go To Sleep’ is set LOW (wake-up), there is bus activity or INT is driven LOW (active). On wake-up, the oscillator is started and a wake-up interrupt is generated. A sleeping SJA1000 which wakes up due to bus activity will not be able to receive this message until it detects 11 consecutive recessive bits (bus-free sequence). It should be noted that setting of GTS is not possible in reset mode. After clearing of reset request, setting of GTS is possible first, when bus-free is detected again.
2. This command bit is used to clear the data overrun condition indicated by the data overrun status bit. As long as the data overrun status bit is set no further data overrun interrupt is generated. It is allowed to give the clear data overrun command at the same time as a release receive buffer command.
3. After reading the contents of the receive buffer, the CPU can release this memory space of the RXFIFO by setting the release receive buffer bit to logic 1. This may result in another message becoming immediately available within the receive buffer. This event will force another receive interrupt, if enabled. If there is no other message available no further receive interrupt is generated and the receive buffer status bit is cleared.
4. The abort transmission bit is used when the CPU requires the suspension of the previously requested transmission, e.g. to transmit a more urgent message before. A transmission already in progress is not stopped. In order to see if the original message had been either transmitted successfully or aborted, the transmission complete status bit should be checked. This should

be done after the transmit buffer status bit has been set to logic 1 (released) or a transmit interrupt has been generated.

5. If the transmission request was set to logic 1 in a previous command, it cannot be cancelled by setting the transmission request bit to logic 0. The requested transmission may be cancelled by setting the abort transmission bit to logic 1.

3.5.4 Status Register (SR)

The content of the status register reflects the status of the SJA1000. The status register appears to the processor as a read only memory.

Table 3.11 Bit interpretation of the status register (SR); base address + 05H

BIT	SYMBOL	NAME	VALUE	FUNCTION
SR.7	BS	Bus Status; note 1	1	bus-off; the SJA1000 is not involved in bus activities
			0	bus-on; the SJA1000 is involved in bus activities
SR.6	ES	Error Status; note 2	1	error; at least one of the error counters has reached or exceeded the CPU warning limit
			0	ok; both error counters are below the warning limit
SR.5	TS	Transmit Status; note 3	1	transmit; the SJA1000 is transmitting a message
			0	idle; no transmit message is in progress
SR.4	RS	Receive Status; note 3	1	receive; the SJA1000 is receiving a message
			0	idle; no receive message is in progress
SR.3	TCS	Transmission Complete Status; note 4	1	complete; the last requested transmission has been successfully completed
			0	incomplete; the previously requested transmission is not yet completed
SR.2	TBS	Transmit Buffer Status; note 5	1	released; the CPU may write a message into the transmit buffer
			0	locked; the CPU cannot access the transmit buffer; a message is waiting for transmission or is already in process
SR.1	DOS	Data Overrun Status; note 6	1	overrun; a message was lost because there was not enough space for that message in the RXFIFO
			0	absent; no data overrun has occurred since the last clear data overrun command was given
SR.0	RBS	Receive Buffer Status; note 7	1	full; one or more messages are available in the RXFIFO
			0	empty; no message is available

Notes (Table 3.11):

1. When the transmit error counter exceeds the limit of 255 [the bus status bit is set to logic 1 (bus-off)] the CAN controller will set the reset request bit to logic 1 (present) and an error interrupt is generated, if enabled. It will stay in this mode until the CPU clears the reset request bit. Once this is completed the CAN controller will wait the minimum protocol-defined time (128 occurrences of the bus-free signal). After that the bus status bit is cleared (bus-on), the error status bit is set to logic 0 (ok), the error counters are reset and an error interrupt is generated, if enabled.
2. Errors detected during reception or transmission will affect the error counters according to the CAN 2.0B protocol specification. The error status bit is set when at least one of the error counters has reached or exceeded the CPU warning limit of 96. An error interrupt is generated, if enabled.
3. If both the receive status and the transmit status bits are logic 0 (idle) the CAN-bus is idle.
4. The transmission complete status bit is set to logic 0 (incomplete) whenever the transmission request bit is set to logic 1. The transmission complete status bit will remain at logic 0 (incomplete) until a message is transmitted successfully.
5. If the CPU tries to write to the transmit buffer when the transmit buffer status bit is at logic 0 (locked), the written byte will not be accepted and will be lost without being indicated.
6. When a message that shall be received has passed the acceptance filter successfully (i.e. earliest after arbitration field), the CAN controller needs space in the RXFIFO to store the message descriptor. Accordingly there must be enough space for each data byte which has been received. If there is not enough space to store the message, that message will be dropped and the data overrun condition will be indicated to the CPU only, if this received message has no errors until the last but one bit of end of frame (message becomes valid).
7. After reading a message stored in the RXFIFO and releasing this memory space with the command release receive buffer, this bit is cleared. If there is another message available within the FIFO this bit is set again with the next bit quantum (tscl).

3.5.5 Interrupt Register (IR)

The interrupt register allows the identification of an interrupt source. When one or more bits of this register are set, the INT pin is activated (LOW). After this register is read by the processor, all bits are reset. This results in a floating level at INT. The interrupt register appears to the processor as a read only memory.

Table 3.12 Bit interpretation of the interrupt register (IR); base address + 07H

BIT	SYMBOL	NAME	VALUE	FUNCTION
IR.7	-	-	-	reserved; note 1
IR.6	-	-	-	reserved; note 1
IR.5	-	-	-	reserved; note 1
IR.4	WUI	Wake-Up Interrupt; note 2	1	set; this bit is set when the sleep mode is left
			0	reset; this bit is cleared by any read access of the CPU
IR.3	DOI	Data Overrun Interrupt; note 3	1	set; this bit is set on a ‘0-to-1’ transition of the data overrun status bit, when the data overrun interrupt enable is set to logic 1 (enabled)
			0	reset; this bit is cleared by any read access of the CPU
IR.2	EI	Error Interrupt	1	set; this bit is set on a change of either the error status or bus status bits if the error interrupt enable is set to logic 1 (enabled)
			0	reset; this bit is cleared by any read access of the CPU
IR.1	TI	Transmit Interrupt	1	set; this bit is set whenever the transmit buffer status changes from logic 0 to logic 1 (released) and transmit interrupt enable is set to logic 1 (enabled)
			0	reset; this bit is cleared by any read access of the CPU
IR.0	RI	Receive Interrupt; note 4	1	set; this bit is set while the receive FIFO is not empty and the receive interrupt enable bit is set to logic 1 (enabled)
			0	reset; this bit is cleared by any read access of the CPU

Notes (Table 3.12):

1. Reading this bit will always reflect a logic 1.
2. A wake-up interrupt is also generated if the CPU tries to set go to sleep while the CAN controller is involved in bus activities or a CAN interrupt is pending.

3. The overrun interrupt bit (if enabled) and the data overrun status bit are set at the same time.
4. The receive interrupt bit (if enabled) and the receive buffer status bit are set at the same time. It should be noted that the receive interrupt bit is cleared upon a read access, even if there is another message available within the FIFO. The moment the release receive buffer command is given and there is another message valid within the receive buffer, the receive interrupt is set again (if enabled) with the next tscl.

3.5.6 Transmit Buffer Layout

The global layout of the transmit buffer is shown in Table 3.13. The buffer serves to store a message from the processor to be transmitted by the SJA1000. It is subdivided into a descriptor and data field. The transmit buffer can be written to and read out by the processor in operating mode only. In reset mode a ‘FFH’ is reflected for all bytes.

Table 3.13 Layout of transmit buffer

CAN ADDR.	FIELD	NAME	BITS							
			7	6	5	4	3	2	1	0
15	descriptor	identifier byte 1	ID.10	ID.9	ID.8	ID.7	ID.6	ID.5	ID.4	ID.3
17		identifier byte 2	ID.2	ID.1	ID.0	RTR	DLC.3	DLC.2	DLC.1	DLC.0
19	data	TX data 1	transmit data byte 1							
1B		TX data 2	transmit data byte 2							
1D		TX data 3	transmit data byte 3							
1F		TX data 4	transmit data byte 4							
21		TX data 5	transmit data byte 5							
23		TX data 6	transmit data byte 6							
25		TX data 7	transmit data byte 7							
27		TX data 8	transmit data byte 8							

3.5.6.1 Identifier (ID)

The identifier consists of 11 bits (ID.10 to ID.0). ID.10 is the most significant bit, which is transmitted first on the bus during the arbitration process. The identifier acts as the message’s name. It is used in a receiver for acceptance filtering and also determining the bus access priority during the arbitration process. The lower the binary value of the identifier the higher the priority. This is due to a larger number of leading dominant bits during arbitration.

3.5.6.2 Remote Transmission Request (RTR)

If this bit is set, a remote frame will be transmitted via the bus. This means that no data bytes are included within this frame. Nevertheless, it is necessary to specify the correct data length code which depends on the corresponding data frame with the same identifier coding. If the RTR bit is not set, a data frame will be sent including the number of data bytes as specified by the data length code.

3.5.6.3 Data Length Code (DLC)

The number of bytes in the data field of a message is coded by the data length code. At the start of a remote frame transmission the data length code is not considered due to the RTR bit being at logic 1 (remote). This forces the number of transmitted/received data bytes to be logic 0. Nevertheless, the data length code must be specified correctly to avoid bus errors if two CAN controllers start a remote frame transmission with the same identifier simultaneously. The range of the data byte count is 0 to 8 bytes and is coded as follows:

DataByteCount = 8 x DLC.3 + 4 x DLC.2 + 2 x DLC.1 + DLC.0

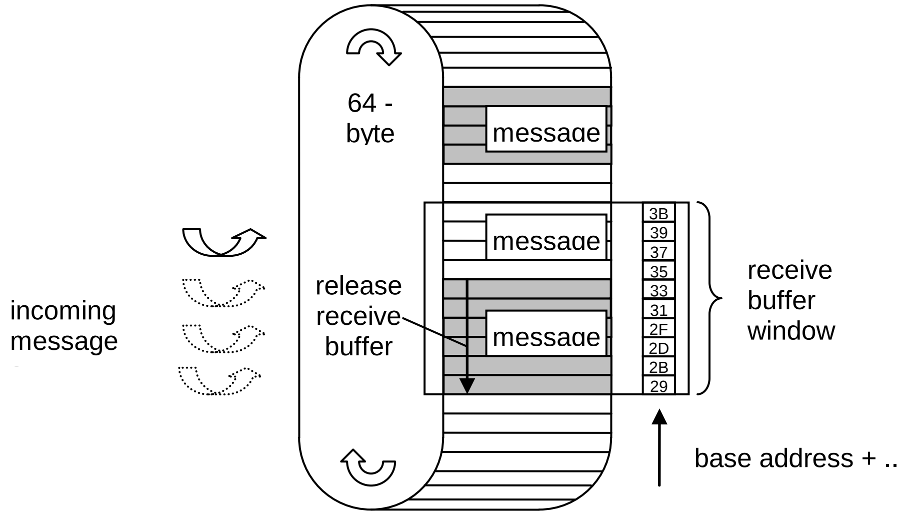
For reasons of compatibility no data length code >8 should be used. If a value >8 is selected, 8 bytes are transmitted in the data frame with the data length code specified in DLC.

3.5.6.4 Data field

The number of transferred data bytes is determined by the data length code. The first bit transmitted is the most significant bit of data byte 1 at address 12.

3.5.7 RECEIVE BUFFER

The global layout of the receive buffer is very similar to the transmit buffer described in Section 3.5.6. The receive buffer is the accessible part of the RXFIFO and is located in the range between base address + 29H and base address + 3BH. Identifier, remote transmission request bit and data length code have the same meaning and location as described in the transmit buffer but within the address range base address + 20H to base address + 29H. As illustrated in Figure 3-1 the RXFIFO has space for 64 message bytes in total. The number of messages that can be stored in the FIFO at any particular moment depends on the length of the individual messages. If there is not enough space for a new message within the RXFIFO, the CAN controller generates a data overrun condition. A message which is partly written into the RXFIFO, when the data overrun condition occurs, is deleted. This situation is indicated to the CPU via the status register and the data overrun interrupt, if enabled and the frame was received without any errors until the last but one bit of end of frame (RX message becomes valid).



Message 1 is now available in the receive buffer.

Figure 3-1 Example of the message storage within the RXFIFO

3.5.8 ACCEPTANCE FILTER

With the help of the acceptance filter the CAN controller is able to allow passing of received messages to the RXFIFO only when the identifier bits of the received message are equal to the predefined ones within the acceptance filter registers. The acceptance filter is defined by the acceptance code register (ACR; see Section 0) and the acceptance mask register (AMR; see Section 0).ACCEPTANCE CODE REGISTER (ACR)

Table 3.14 ACR bit allocation; base address + 09H

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
AC.7	AC.6	AC.5	AC.4	AC.3	AC.2	AC.1	AC.0

This register can be accessed (read/write), if the reset request bit is set HIGH (present). When a message is received which passes the acceptance test and there is receive buffer space left, then the respective descriptor and data field are sequentially stored in the RXFIFO. When the complete message has been correctly received the following occurs:

- The receive status bit is set HIGH (full)
- If the receive interrupt enable bit is set HIGH (enabled), the receive interrupt is set HIGH (set).

The acceptance code bits (AC.7 to AC.0) and the eight most significant bits of the message’s identifier (ID.10 to ID.3) must be equal to those bit positions which are marked relevant by the acceptance mask bits (AM.7 to AM.0). If the conditions as described in the following equation are fulfilled, acceptance is given:

(ID.10 to ID.3) ≡ (AC.7 to AC.0)] V (AM.7 to AM.0) ≡ 11111111

3.5.9 ACCEPTANCE MASK REGISTER (AMR)

Table 3.15 AMR bit allocation; base address + 0BH

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
AM.7	AM.6	AM.5	AM.4	AM.3	AM.2	AM.1	AM.0

This register can be accessed (read/write), if the reset request bit is set HIGH (present). The acceptance mask register qualifies which of the corresponding bits of the acceptance code are ‘relevant’ (AM.X = 0) or ‘don’t care’ (AM.X = 1) for acceptance filtering.

3.5.10 Other registers

The other registers are described in Section 3.7.

3.6 PeliCAN mode

3.6.1 PeliCAN ADDRESS LAYOUT

The CAN controller’s internal registers appear to the CPU as on-chip memory mapped peripheral registers. Because the CAN controller can operate in different modes (operating/reset; see also Section 3.6.3), one has to distinguish between different internal address definitions. Starting from base address + 41H the complete internal RAM (80-byte) is mapped to the CPU interface.

Table 3.16 IP560 Memory Space Address (Hex) Memory Map PeliCAN

Hex Offset From Base Addr	MSB D15 D08	LSB D07 D00						Hex Offset From Base Addr
		OPERATING MODE				RESET MODE		
		READ		WRITE		READ	WRITE	
CHANNEL 0								
00	Not Used	mode		mode		mode	mode	**01**
02	Not Used	(00H)		command		(00H)	command	**03**
04	Not Used	status		-		status	-	**05**
06	Not Used	interrupt		-		interrupt	-	**07**
08	Not Used	interrupt enable		interrupt enable		interrupt enable	interrupt enable	**09**
0A	Not Used	reserved (00H)		-		Reserved (00H)	-	**0B**
0C	Not Used	bus timing 0		-		bus timing 0	bus timing 0	**0D**
0E	Not Used	bus timing 1		-				**0F**
10	Not Used	output control		-		output control	-	**11**
12	Not Used	test[1]		test[1]		test[1]	test[1]	**13**
14	Not Used	reserved (00H)		-		reserved (00H)	-	**15**
16	Not Used	arbitration lost capture		-		arbitration lost capture	-	**17**
18	Not Used	error code capture		-		error code capture	-	**19**
1A	Not Used	error warning limit		-		error warning limit	error warning limit	**1B**
1C	Not Used	RX error counter		-		RX error counter	RX error counter	**1D**
1E	Not Used	TX error counter		-		TX error counter	TX error counter	**1F**
20	Not Used	RX frame information SFF[2]	RX frame information EFF[3]	TX frame information SFF[2]	TX frame information EFF[3]	acceptance code 0	acceptance code 0	**21**
22	Not Used	RX identifier 1	RX identifier 1	TX identifier 1	TX identifier 1	acceptance code 1	acceptance code 1	**23**
24	Not Used	RX identifier 2	RX identifier 2	TX identifier 2	TX identifier 2	acceptance code 2	acceptance code 2	**25**
26	Not Used	RX data 1	RX identifier 3	TX data 1	TX identifier 3	acceptance code 3	acceptance code 3	**27**
28	Not Used	RX data 2	RX identifier 4	TX data 2	TX identifier 4	acceptance mask 0	acceptance mask 0	**29**
2A	Not Used	RX data 3	RX data 1	TX data 3	TX data 1	acceptance	acceptance	**2B**

Hex Offset From Base Addr	MSB D15 D08	LSB D07 D00						Hex Offset From Base Addr
		OPERATING MODE				RESET MODE		
		READ		WRITE		READ	WRITE	
						mask 1	mask 1	
2C	Not Used	RX data 4	RX data 2	TX data 4	TX data 2	acceptance mask 2	acceptance mask 2	**2D**
2E	Not Used	RX data 5	RX data 3	TX data 5	TX data 3	acceptance mask 3	acceptance mask 3	**2F**
30	Not Used	RX data 6	RX data 4	TX data 6	TX data 4	reserved (00H)	-	**31**
32	Not Used	RX data 7	RX data 5	TX data 7	TX data 5	reserved (00H)	-	**33**
34	Not Used	RX data 8	RX data 6	TX data 8	TX data 6	reserved (00H)	-	**35**
36	Not Used	(FIFO RAM)[4]	RX data 7	-	TX data 7	reserved (00H)	-	**37**
38	Not Used	(FIFO RAM)[4]	RX data 8	-	TX data 8	reserved (00H)	-	**39**
3A	Not Used	RX message counter		-		RX message counter	-	**3B**
3C	Not Used	RX buffer start address		-		RX buffer start address	RX buffer start address	**3D**
3E	Not Used	Clock divider[5]		Clock divider[5]		Clock divider[5]	Clock divider[5]	**3F**
40	Not used	internal RAM address 0 (FIFO)		-		internal RAM address 0	internal RAM address 0	**41**
42	Not used	internal RAM address 1 (FIFO)		-		internal RAM address 1	internal RAM address 1	**43**
▼	▼	▼		▼		▼	▼	▼
BE	Not used	internal RAM address 63 (FIFO)		-		internal RAM address 63	internal RAM address 63	**BF**
C0	Not used	internal RAM address 64 (TX buffer)		-		internal RAM address 64	internal RAM address 64	**C1**
▼	▼	▼		▼		▼	▼	▼
D8	Not used	internal RAM address 76 (TX buffer)		-		internal RAM address 76	internal RAM address 76	**D9**
DA	Not used	internal RAM address 77 (free)		-		internal RAM address 77	internal RAM address 77	**DB**
DC	Not used	internal RAM address 78 (free)		-		internal RAM address 78	internal RAM address 78	**DD**

Hex Offset From Base Addr	MSB D15 D08	LSB D07 D00				Hex Offset From Base Addr
		OPERATING MODE		RESET MODE		
		READ	WRITE	READ	WRITE	
DE	Not used	internal RAM address 79 (free)	-	internal RAM address 79	internal RAM address 79	**DF**
E0	Not used	(00H)	-	(00H)	-	**E1**
▼	▼	▼	▼	▼	▼	▼
FE	Not used	(00H)	-	(00H)	-	**FF**
	-					

Hex Offset From Base Addr	MSB	LSB D07 D00						Hex Offset From Base Addr
		OPERATING MODE				RESET MODE		
	D15 D08	READ		WRITE		READ	WRITE	
Channel 1								
200	Not Used	mode		mode		mode	mode	**201**
202	Not Used	(00H)		command		(00H)	command	**203**
204	Not Used	status		-		status	-	**205**
206	Not Used	interrupt		-		interrupt	-	**207**
208	Not Used	interrupt enable		interrupt enable		interrupt enable	interrupt enable	**209**
20A	Not Used	reserved (00H)		-		Reserved (00H)	-	**20B**
20C	Not Used	bus timing 0		-		bus timing 0	bus timing 0	**20D**
20E	Not Used	bus timing 1		-				**20F**
210	Not Used	output control		-		output control	-	**211**
212	Not Used	test[1]		test[1]		test[1]	test[1]	**213**
214	Not Used	reserved (00H)		-		reserved (00H)	-	**215**
216	Not Used	arbitration lost capture		-		arbitration lost capture	-	**217**
218	Not Used	error code capture		-		error code capture	-	**219**
21A	Not Used	error warning limit		-		error warning limit	error warning limit	**21B**
21C	Not Used	RX error counter		-		RX error counter	RX error counter	**21D**
21E	Not Used	TX error counter		-		TX error counter	TX error counter	**21F**
220	Not Used	RX frame information SFF[2]	RX frame information EFF[3]	TX frame information SFF[2]	TX frame information EFF[3]	acceptance code 0	acceptance code 0	**221**
222	Not Used	RX identifier 1	RX identifier 1	TX identifier 1	TX identifier 1	acceptance code 1	acceptance code 1	**223**
224	Not Used	RX identifier 2	RX identifier 2	TX identifier 2	TX identifier 2	acceptance code 2	acceptance code 2	**225**
226	Not Used	RX data 1	RX identifier 3	TX data 1	TX identifier 3	acceptance code 3	acceptance code 3	**227**
228	Not Used	RX data 2	RX identifier 4	TX data 2	TX identifier 4	acceptance mask 0	acceptance mask 0	**229**
22A	Not Used	RX data 3	RX data 1	TX data 3	TX data 1	acceptance mask 1	acceptance mask 1	**22B**
22C	Not Used	RX data 4	RX data 2	TX data 4	TX data 2	acceptance mask 2	acceptance mask 2	**22D**

Hex Offset From Base Addr	MSB D15 D08	LSB D07 D00						Hex Offset From Base Addr
		OPERATING MODE				RESET MODE		
		READ		WRITE		READ	WRITE	
22E	Not Used	RX data 5	RX data 3	TX data 5	TX data 3	acceptance mask 3	acceptance mask 3	22F
230	Not Used	RX data 6	RX data 4	TX data 6	TX data 4	reserved (00H)	-	231
232	Not Used	RX data 7	RX data 5	TX data 7	TX data 5	reserved (00H)	-	233
234	Not Used	RX data 8	RX data 6	TX data 8	TX data 6	reserved (00H)	-	235
236	Not Used	(FIFO RAM)[4]	RX data 7	-	TX data 7	reserved (00H)	-	237
238	Not Used	(FIFO RAM)[4]	RX data 8	-	TX data 8	reserved (00H)	-	239
23A	Not Used	RX message counter		-		RX message counter	-	23B
23C	Not Used	RX buffer start address		-		RX buffer start address	RX buffer start address	23D
23E	Not Used	Clock divider[5]		Clock divider[5]		Clock divider[5]	Clock divider[5]	23F
240	Not used	internal RAM address 0 (FIFO)		-		internal RAM address 0	internal RAM address 0	241
242	Not used	internal RAM address 1 (FIFO)		-		internal RAM address 1	internal RAM address 1	243
▼	▼	▼		▼		▼	▼	▼
2BE	Not used	internal RAM address 63 (FIFO)		-		internal RAM address 63	internal RAM address 63	2BF
2C0	Not used	internal RAM address 64 (TX buffer)		-		internal RAM address 64	internal RAM address 64	2C1
▼	▼	▼		▼		▼	▼	▼
2D8	Not used	internal RAM address 76 (TX buffer)		-		internal RAM address 76	internal RAM address 76	2D9
2DA	Not used	internal RAM address 77 (free)		-		internal RAM address 77	internal RAM address 77	2DB
2DC	Not used	internal RAM address 78 (free)		-		internal RAM address 78	internal RAM address 78	2DD
2DE	Not used	internal RAM address 79 (free)		-		internal RAM address 79	internal RAM address 79	2DF

Hex Offset From Base Addr	MSB D15 D08	LSB D07 D00				Hex Offset From Base Addr
		OPERATING MODE		RESET MODE		
		READ	WRITE	READ	WRITE	
2E0	Not used	(00H)	-	(00H)	-	**2E1**
▼	▼	▼	▼	▼	▼	▼
2FE	Not used	(00H)	-	(00H)	-	**2FF**

Notes (Table 3.16):

1. Test register is used for production testing only. Using this register during normal operation may result in undesired behavior of the device.
2. SFF = Standard Frame Format.
3. EFF = Extended Frame Format.
4. These addresses allocations reflect the FIFO RAM space behind the current message. The contents are random after power-up and contain the beginning of the next message which is received after the current one. If no further message is received, parts of old messages may occur here.
5. Some bits are writeable in reset mode only (CAN mode, CBP, RXINTEN and clock off).

3.6.2 RESET VALUES

Detection of a set reset mode bit results in aborting the current transmission/reception of a message and entering the reset mode. On the ‘1-to-0’ transition of the reset mode bit, the CAN controller returns to the mode defined within the mode register.

Table 3.17 Reset mode configuration; notes 1 and 2

REGISTER	BIT	SYMBOL	NAME	VALUE	
				RESET BY HARDWARE	SETTING BIT CR.0 BY SOFTWARE OR DUE TO BUS-OFF
Mode	MOD.7 to 5	-	reserved	0 (reserved)	0 (reserved)
	MOD.4	SM	Sleep Mode	0 (wake-up)	0 (wake-up)
	MOD.3	AFM	Acceptance Filter Mode	0 (dual)	X
	MOD.2	STM	Self Test Mode	0 (normal)	X
	MOD.1	LOM	Listen Only Mode	0 (normal)	X
	MOD.0	RM	Reset Mode	1 (present)	1 (present)
Command	CMR.7 to 5	-	reserved	0 (reserved)	0 (reserved)
	CMR.4	SRR	Self Reception Request	0 (absent)	0 (absent)
	CMR.3	CDO	Clear Data Overrun	0 (no action)	0 (no action)
	CMR.2	RRB	Release Receive Buffer	0 (no action)	0 (no action)
	CMR.1	AT	Abort Transmission	0 (absent)	0 (absent)
	CMR.0	TR	Transmission Request	0 (absent)	0 (absent)
Status	SR.7	BS	Bus Status	0 (bus-on)	X
	SR.6	ES	Error Status	0 (ok)	X
	SR.5	TS	Transmit Status	1 (wait idle)	1 (wait idle)
	SR.4	RS	Receive Status	1 (wait idle)	1 (wait idle)
	SR.3	TCS	Transmission Complete Status	1 (complete)	X
	SR.2	TBS	Transmit Buffer Status	1 (released)	1 (released)
	SR.1	DOS	Data Overrun Status	0 (absent)	0 (absent)
	SR.0	RBS	Receive Buffer Status	0 (empty)	0 (empty)
Interrupt	IR.7	BEI	Bus Error Interrupt	0 (reset)	0 (reset)
	IR.6	ALI	Arbitration Lost Interrupt	0 (reset)	0 (reset)
	IR.5	EPI	Error Passive Interrupt	0 (reset)	0 (reset)
	IR.4	WUI	Wake-Up Interrupt	0 (reset)	0 (reset)
	IR.3	DOI	Data Overrun Interrupt	0 (reset)	0 (reset)
	IR.2	EI	Error Warning Interrupt	0 (reset)	X; note 3
	IR.1	TI	Transmit Interrupt	0 (reset)	0 (reset)
	IR.0	RI	Receive Interrupt	0 (reset)	0 (reset)
Interrupt	IER.7	BEIE	Bus Error	X	X

REGISTER	BIT	SYMBOL	NAME	VALUE	
				RESET BY HARDWARE	SETTING BIT CR.0 BY SOFTWARE OR DUE TO BUS-OFF
enable			Interrupt Enable		
	IER.6	ALIE	Arbitration Lost Interrupt Enable	X	X
	IER.5	EPIE	Error Passive Interrupt Enable	X	X
	IER.4	WUIE	Wake-Up Interrupt Enable	X	X
	IER.3	DOIE	Data Overrun Interrupt Enable	X	X
	IER.2	EIE	Error Warning Interrupt Enable	X	X
	IER.1	TIE	Transmit Interrupt Enable	X	X
	IER.0	RIE	Receive Interrupt Enable	X	X
Bus timing 0	BTR0.7	SJW.1	Synchronization Jump Width 1	X	X
	BTR0.6	SJW.0	Synchronization Jump Width 0	X	X
	BTR0.5	BRP.5	Baud Rate Prescaler 5	X	X
	BTR0.4	BRP.4	Baud Rate Prescaler 4	X	X
	BTR0.3	BRP.3	Baud Rate Prescaler 3	X	X
	BTR0.2	BRP.2	Baud Rate Prescaler 2	X	X
	BTR0.1	BRP.1	Baud Rate Prescaler 1	X	X
	BTR0.0	BRP.0	Baud Rate Prescaler 0	X	X
Bus timing 1	BTR1.7	SAM	Sampling	X	X
	BTR1.6	TSEG2.2	Time Segment 2.2	X	X
	BTR1.5	TSEG2.1	Time Segment 2.1	X	X
	BTR1.4	TSEG2.0	Time Segment 2.0	X	X
	BTR1.3	TSEG1.3	Time Segment 1.3	X	X
	BTR1.2	TSEG1.2	Time Segment 1.2	X	X
	BTR1.1	TSEG1.1	Time Segment 1.1	X	X
	BTR1.0	TSEG1.0	Time Segment 1.0	X	X
Output control	OC.7	OCTP1	Output Control Transistor P1	X	X

REGISTER	BIT	SYMBOL	NAME	VALUE	
				RESET BY HARDWARE	SETTING BIT CR.0 BY SOFTWARE OR DUE TO BUS-OFF
	OC.6	OCTN1	Output Control Transistor N1	X	X
	OC.5	OCPOL1	Output Control Polarity 1	X	X
	OC.4	OCTP0	Output Control Transistor P0	X	X
	OC.3	OCTN0	Output Control Transistor N0	X	X
	OC.2	OCPOL0	Output Control Polarity 0	X	X
	OC.1	OCMODE1	Output Control Mode 1	X	X
	OC.0	OCMODE0	Output Control Mode 0	X	X
Arbitration lost capture	-	ALC	Arbitration Lost Capture	0	X
Error code capture	-	ECC	Error Code Capture	0	X
Error warning limit	-	EWLR	Error Warning Limit Register	96	X
RX error counter	-	RXERR	Receive Error Counter	0 (reset)	X; note 4
TX error counter	-	TXERR	Transmit Error Counter	0 (reset)	X; note 4
TX buffer	-	TXB	Transmit Buffer	X	X
RX buffer	-	RXB	Receive Buffer	X; note 5	X; note 5
ACR 0 to 3	-	ACR0 to ACR3	Acceptance Code Registers	X	X
AMR 0 to 3	-	AMR0 to AMR3	Acceptance Mask Registers	X	X
RX message counter	-	RMC	RX Message Counter	0	0
RX buffer start address	-	RXERR	RX Buffer Start Address	00000000	X
Clock divider	-	CDR	Clock Divider Register	00000000	X

Notes (Table 3.17):

1. X means that the value of these registers or bits is not influenced.
2. Remarks in brackets explain functional meaning.
3. On bus-off the error warning interrupt is set, if enabled.
4. If the reset mode was entered due to a bus-off condition, the receive error counter is cleared and the transmit error counter is initialized to 127 to count-down the CAN-defined bus-off recovery time consisting of 128 occurrences of 11 consecutive recessive bits.
5. Internal read/write pointers of the RXFIFO are reset to their initial values. A subsequent read access to the RXB would show undefined data values (parts of old messages). If a message is transmitted, this message is written in parallel to the receive buffer. A receive interrupt is generated only if this

transmission was forced by the self reception request. So, even if the receive buffer is empty, the last transmitted message may be read from the receive buffer until it is overwritten by the next received or transmitted message. Upon a hardware reset, the XFIFO pointers are reset to the physical RAM address ‘0’. Setting CR.0 by software or due to the bus-off event will reset the RXFIFO pointers to the currently valid FIFO start address (RBSA register) which is different from the RAM address ‘0’ after the first release receive buffer command.

3.6.3 MODE REGISTER

The contents of the mode register are used to change the behavior of the CAN controller. Bits may be set or reset by the CPU which uses the control register as a read/write memory. Reserved bits are read as logic 0.

Table 3.18 Bit interpretation of the mode register (MOD); base address + 0H

BIT	SYMBOL	NAME	VALUE	FUNCTION
MOD.7	-	-	-	reserved
MOD.6	-	-	-	reserved
MOD.5	-	-	-	reserved
MOD.4	SM	Sleep Mode; note 1	1	sleep; the CAN controller enters sleep mode if no CAN interrupt is pending and if there is no bus activity
			0	wake-up; the CAN controller wakes up if sleeping
MOD.3	AFM	Acceptance Filter Mode; note 2	1	single; the single acceptance filter option is enabled (one filter with the length of 32 bit is active)
			0	dual; the dual acceptance filter option is enabled (two filters, each with the length of 16 bit are active)
MOD.2	STM	Self Test Mode; note 2	1	self test; in this mode a full node test is possible without any other active node on the bus using the self reception request command; the CAN controller will perform a successful transmission, even if there is no acknowledge received
			0	normal; an acknowledge is required for successful transmission
MOD.1	LOM	Listen Only Mode; notes 2 and 3	1	listen only; in this mode the CAN controller would give no acknowledge to the CAN-bus, even if a message is received successfully; the error counters are stopped at the current value
			0	normal
MOD.0	RM	Reset Mode; note 4	1	reset; detection of a set reset mode bit results in aborting the current transmission/reception of a message and entering the reset mode
			0	normal; on the ‘1-to-0’ transition of the reset mode bit, the CAN controller returns to the operating mode

Notes (Table 3.18):

1. The SJA1000 will enter sleep mode if the sleep mode bit is set to logic 1 (sleep); then there is no bus activity and no interrupt is pending. Setting of SM with at least one of the previously mentioned exceptions valid will result in a wake-up interrupt. The SJA1000 will wake up when one of the three previously mentioned conditions is negated: after SM is set LOW (wake-up), there is bus activity or INT is driven LOW (active). On wake-up, a wake-up interrupt is generated. A sleeping SJA1000 which wakes up due to bus activity will not be able to receive this message until it detects 11 consecutive recessive bits (bus-free sequence). It should be noted that setting of SM is not possible in reset mode. After clearing of reset mode, setting of SM is possible first, when bus-free is detected again.
2. A write access to the bits MOD.1 to MOD.3 is only possible if the reset mode is entered previously.
3. This mode of operation forces the CAN controller to be error passive. Message transmission is not possible. The listen only mode can be used e.g. for software driven bit rate detection and ‘hot plugging’. All other functions can be used like in normal mode.
4. During a hardware reset or when the bus status bit is set to logic 1 (bus-off), the reset mode bit is also set to logic 1 (present). If this bit is accessed by software, a value change will become visible and takes effect first with the next positive edge of the internal clock which operates at 12 MHz. Changes of the reset request bit are synchronized with the internal divided clock. Reading the reset request bit reflects the synchronized status. After the reset mode bit is set to logic 0 the CAN controller will wait for:
   a) One occurrence of bus-free signal (11 recessive bits), if the preceding reset has been caused by a hardware reset or a CPU-initiated reset.
   b) 128 occurrences of bus-free, if the preceding reset has been caused by a CAN controller initiated bus-off, before re-entering the bus-on mode.

3.6.4 Command Register (CMR)

A command bit initiates an action within the transfer layer of the CAN controller. This register is write only, all bits will return a logic 0 when being read. Between two commands at least one internal clock cycle is needed in order to proceed. The internal clock frequency is 12 MHz.

Table 3.19 Bit interpretation of the command register (CMR); base address + 03H

BIT	SYMBOL	NAME	VALUE	FUNCTION
CMR.7	-	-	-	-
CMR.6	-	-	-	-
CMR.5	-	-	-	-
CMR.4	SRR	Self Reception Request; notes 1 and 2	1	present; a message shall be transmitted and received simultaneously
			0	absent
CMR.3	CDO	Clear Data Overrun; note 3	1	clear; data overrun status bit is cleared
			0	no action
CMR.2	RRB	Release Receive Buffer; note 4	1	released; the receive buffer, representing the message memory space in the RXFIFO is released
			0	no action
CMR.1	AT	Abort Transmission; notes 5 and 2	1	present; if not already in progress, a pending transmission request is cancelled
			0	absent
CMR.0	TR	Transmission Request; notes 6 and 2	1	present; a message shall be transmitted
			0	absent

Notes (Table 3.19):

1. Upon self reception request a message is transmitted and simultaneously received if the acceptance filter is set to the corresponding identifier. A receive and a transmit interrupt will indicate correct self reception (see also self test mode in mode register).
2. Setting the command bits CMR.0 and CMR.1 simultaneously results in sending the transmit message once. No re-transmission will be performed in the event of an error or arbitration lost (single-shot transmission). Setting the command bits CMR.4 and CMR.1 simultaneously results in sending the transmit message once using the self reception feature. No re-transmission will be performed in the event of an error or arbitration lost. Setting the command bits CMR.0, CMR.1 and CMR.4 simultaneously results in sending the transmit message once as described for CMR.0 and CMR.1. The moment the transmit status bit is set within the status register, the internal transmission request bit is cleared automatically. Setting CMR.0 and CMR.4 simultaneously will ignore the set CMR.4 bit.
3. This command bit is used to clear the data overrun condition indicated by the data overrun status bit. As long as the data overrun status bit is set no further data overrun interrupt is generated.
4. After reading the contents of the receive buffer, the CPU can release this memory space in the RXFIFO by setting the release receive buffer bit to logic 1. This may result in another message becoming immediately available within the receive buffer. If there is no other message available, the receive interrupt bit is reset.

5. The abort transmission bit is used when the CPU requires the suspension of the previously requested transmission, e.g. to transmit a more urgent message before. A transmission already in progress is not stopped. In order to see if the original message has been either transmitted successfully or aborted, the transmission complete status bit should be checked. This should be done after the transmit buffer status bit has been set to logic 1 or a transmit interrupt has been generated. It should be noted that a transmit interrupt is generated even if the message was aborted because the transmit buffer status bit changes to ‘released’.
6. If the transmission request was set to logic 1 in a previous command, it cannot be cancelled by setting the transmission request bit to logic 0. The requested transmission may be cancelled by setting the abort transmission bit to logic 1.

3.6.5 STATUS REGISTER (SR)

The content of the status register reflects the status of the CAN controller. The status register appears to the CPU as a read only memory.

Table 3.20 Bit interpretation of the status register (SR); base address + 05H

BIT	SYMBOL	NAME	VALUE	FUNCTION
SR.7	BS	Bus Status; note 1	1	bus-off; the SJA1000 is not involved in bus activities
			0	bus-on; the SJA1000 is involved in bus activities
SR.6	ES	Error Status; note 2	1	error; at least one of the error counters has reached or exceeded the CPU warning limit defined by the Error Warning Limit Register (EWLR)
			0	ok; both error counters are below the warning limit
SR.5	TS	Transmit Status; note 3	1	transmit; the SJA1000 is transmitting a message
			0	idle
SR.4	RS	Receive Status; note 3	1	receive; the SJA1000 is receiving a message
			0	idle
SR.3	TCS	Transmission Complete Status; note 4	1	complete; the last requested transmission has been successfully completed
			0	incomplete; the previously requested transmission is not yet completed
SR.2	TBS	Transmit Buffer Status; note 5	1	released; the CPU may write a message into the transmit buffer
			0	locked; the CPU cannot access the transmit buffer; a message is waiting for transmission or is already in process
SR.1	DOS	Data Overrun Status; note 6	1	overrun; a message was lost because there was not enough space for that message in the RXFIFO
			0	absent; no data overrun has occurred since the last clear data overrun command was given
SR.0	RBS	Receive Buffer Status; note 7	1	full; one or more messages are available in the RXFIFO
			0	empty; no message is available

Notes (Table 3.20):

1. When the transmit error counter exceeds the limit of 255, the bus status bit is set to logic 1 (bus-off), the CAN controller will set the reset mode bit to logic 1 (present) and an error warning interrupt is generated, if enabled. The transmit error counter is set to 127 and the receive error counter is cleared. It will stay in this mode until the CPU clears the reset mode bit. Once this is completed the CAN controller will wait the minimum protocol-defined time (128 occurrences of the bus-free signal) counting down the transmit error counter. After that the bus status bit is cleared (bus-on), the error status bit is set to logic 0 (ok), the error counters are reset and an error warning interrupt is generated, if enabled. Reading the TX error counter during this time gives information about the status of the bus-off recovery.
2. Errors detected during reception or transmission will effect the error counters according to the CAN 2.0B protocol specification. The error status bit is set when at least one of the error counters has reached or exceeded the CPU warning limit (EWLR). An error warning interrupt is generated, if enabled. The default value of EWLR after hardware reset is 96.
3. If both the receive status and the transmit status bits are logic 0 (idle) the CAN-bus is idle. If both bits are set the controller is waiting to become idle again. After a hardware reset 11 consecutive recessive bits have to be detected until the idle status is reached. After bus-off this will take 128 of 11 consecutive recessive bits.
4. The transmission complete status bit is set to logic 0 (incomplete) whenever the transmission request bit or the self reception request bit is set to logic 1. The transmission complete status bit will remain at logic 0 until a message is transmitted successfully.
5. If the CPU tries to write to the transmit buffer when the transmit buffer status bit is logic 0 (locked), the written byte will not be accepted and will be lost without this being indicated.
6. When a message that is to be received has passed the acceptance filter successfully, the CAN controller needs space in the RXFIFO to store the message descriptor and for each data byte which has been received. If there is not enough space to store the message, that message is dropped and the data overrun condition is indicated to the CPU at the moment this message becomes valid. If this message is not completed successfully (e.g. due to an error), no overrun condition is indicated.
7. After reading all messages within the RXFIFO and releasing their memory space with the command release receive buffer this bit is cleared.

3.6.6 INTERRUPT REGISTER (IR)

The interrupt register allows the identification of an interrupt source. When one or more bits of this register are set, a CAN interrupt will be indicated to the CPU. After this register is read by the CPU all bits are reset except for the receive interrupt bit.

The interrupt register appears to the CPU as a read only memory.

Table 3.21 Bit interpretation of the interrupt register (IR); base address + 07H

BIT	SYMBOL	NAME	VALUE	FUNCTION
IR.7	BEI	Bus Error Interrupt	1	set; this bit is set when the CAN controller detects an error on the CAN-bus and the BEIE bit is set within the interrupt enable register
			0	reset
IR.6	ALI	Arbitration Lost Interrupt	1	set; this bit is set when the CAN controller lost the arbitration and becomes a receiver and the ALIE bit is set within the interrupt enable register
			0	reset
IR.5	EPI	Error Passive Interrupt	1	set; this bit is set whenever the CAN controller has reached the error passive status (at least one error counter exceeds the protocol-defined level of 127) or if the CAN controller is in the error passive status and enters the error active status again and the EPIE bit is set within the interrupt enable register
			0	reset
IR.4	WUI	Wake-Up Interrupt; note 1	1	set; this bit is set when the CAN controller is sleeping and bus activity is detected and the WUIE bit is set within the interrupt enable register
			0	reset
IR.3	DOI	Data Overrun Interrupt	1	set; this bit is set on a ‘0-to-1’ transition of the data overrun status bit and the DOIE bit is set within the interrupt enable register
			0	reset
IR.2	EI	Error Warning Interrupt	1	set; this bit is set on every change (set and clear) of either the error status or bus status bits and the EIE bit is set within the interrupt enable register

			0	reset
IR.1	TI	Transmit Interrupt	1	set; this bit is set whenever the transmit buffer status changes from ‘0-to-1’ (released) and the TIE bit is set within the interrupt enable register
			0	reset
IR.0	RI	Receive Interrupt; note 2	1	set; this bit is set while the receive FIFO is not empty and the RIE bit is set within the interrupt enable register
			0	reset; no more message is available within the RXFIFO

Notes (Table 3.21):

1. A wake-up interrupt is also generated, if the CPU tries to set the sleep bit while the CAN controller is involved in bus activities or a CAN interrupt is pending.
2. The behavior of this bit is equivalent to that of the receive buffer status bit with the exception, that RI depends on the corresponding interrupt enable bit (RIE). So the receive interrupt bit is not cleared upon a read access to the interrupt register. Giving the command ‘release receive buffer’ will clear RI temporarily. If there is another message available within the FIFO after the release command, RI is set again. Otherwise RI remains cleared.

3.6.7 INTERRUPT ENABLE REGISTER (IER)

The register allows to enable different types of interrupt sources which are indicated to the CPU.

The interrupt enable register appears to the CPU as a read/write memory.

Table 3.22 Bit interpretation of the interrupt enable register (IER); base address + 09H

BIT	SYMBOL	NAME	VALUE	FUNCTION
IER.7	BEIE	Bus Error Interrupt Enable	1	enabled; if an bus error has been detected, the CAN controller requests the respective interrupt
			0	disabled
IER.6	ALIE	Arbitration Lost Interrupt Enable	1	enabled; if the CAN controller has lost arbitration, the respective interrupt is requested
			0	disabled
IER.5	EPIE	Error Passive Interrupt Enable	1	enabled; if the error status of the CAN controller changes from error active to error passive or vice versa, the respective interrupt is requested
			0	disabled
IER.4	WUIE	Wake-Up Interrupt Enable	1	enabled; if the sleeping CAN controller wakes up, the respective interrupt is requested
			0	disabled
IER.3	DOIE	Data Overrun Interrupt Enable	1	enabled; if the data overrun status bit is set (see status register; Table 3.20), the CAN controller requests the respective interrupt
			0	disabled
IER.2	EIE	Error Warning Interrupt Enable	1	enabled; if the error or bus status change (see status register; Table 3.20), the CAN controller requests the respective interrupt
			0	disabled
IER.1	TIE	Transmit Interrupt Enable	1	enabled; when a message has been successfully transmitted or the transmit buffer is accessible again (e.g. after an abort transmission command), the CAN controller requests the respective interrupt
			0	disabled
IER.0	RIE	Receive Interrupt Enable; note 1	1	enabled; when the receive buffer status is ‘full’ the CAN controller requests the respective interrupt

			0	disabled

Note (Table 3.22):

1. The receive interrupt enable bit has direct influence to the receive interrupt bit and the external interrupt output INT. If RIE is cleared, the external INT pin will become HIGH immediately, if there is no other interrupt pending.

3.6.8 ARBITRATION LOST CAPTURE (ALC)

This register contains information about the bit position of losing arbitration. The arbitration lost capture register appears to the CPU as a read only memory. Reserved bits are read as logic 0.

Table 3.23 Bit interpretation of the arbitration lost capture register (ALC); base address + 17H

BIT	SYMBOL	NAME	VALUE	FUNCTION
ALC.7 to ALC.5	-	Reserved	For value and function see Table 3.24	
ALC.4	BITNO4	bit number 4		
ALC.3	BITNO3	bit number 3		
ALC.2	BITNO2	bit number 2		
ALC.1	BITNO1	bit number 1		
ALC.0	BITNO0	bit number 0		

On arbitration lost, the corresponding arbitration lost interrupt is forced, if enabled. At the same time, the current bit position of the bit stream processor is captured into the arbitration lost capture register. The content within this register is fixed until the users software has read out its contents once. The capture mechanism is then activated again.

The corresponding interrupt flag located in the interrupt register is cleared during the read access to the interrupt register. A new arbitration lost interrupt is not possible until the arbitration lost capture register is read out once.

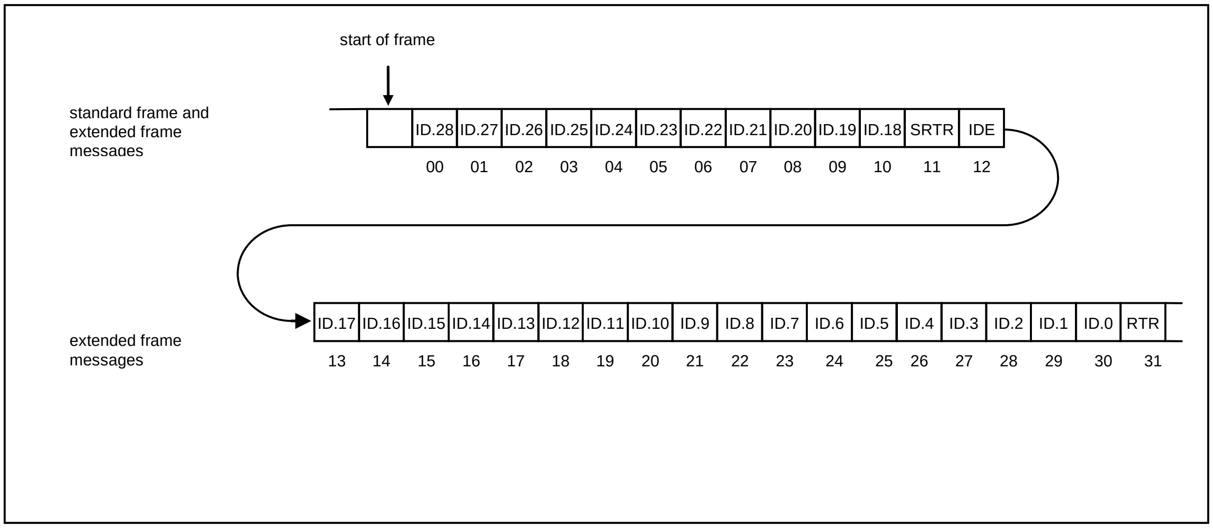

start of frame
standard frame and extended frame messages
ID.28 ID.27 ID.26 ID.25 ID.24 ID.23 ID.22 ID.21 ID.20 ID.19 ID.18 SRTR IDE
00 01 02 03 04 05 06 07 08 09 10 11 12
extended frame messages
ID.17 ID.16 ID.15 ID.14 ID.13 ID.12 ID.11 ID.10 ID.9 ID.8 ID.7 ID.6 ID.5 ID.4 ID.3 ID.2 ID.1 ID.0 RTR
13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31

Figure 3-2 Arbitration lost bit number interpretation

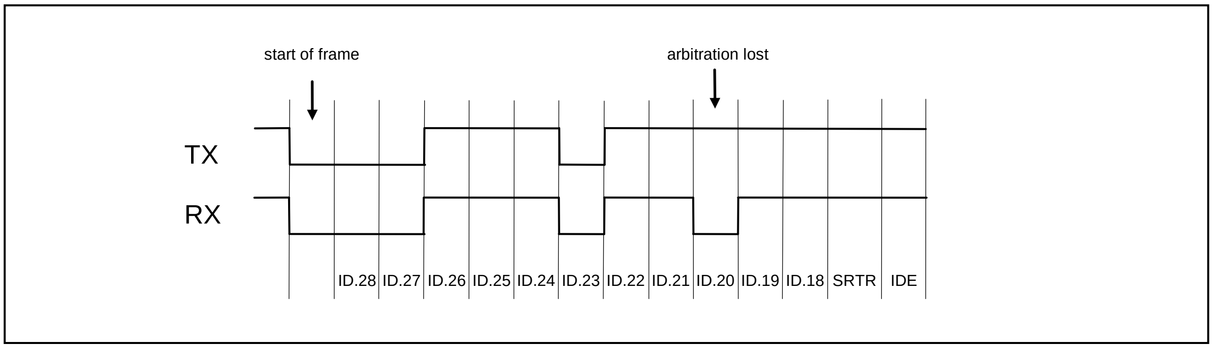

start of frame
arbitration lost
TX
RX
ID.28 ID.27 ID.26 ID.25 ID.24 ID.23 ID.22 ID.21 ID.20 ID.19 ID.18 SRTR IDE

Figure 3-3 Example of arbitration lost bit number interpretation; result: ALC = 08

Table 3.24 Function of bits 4 to 0 of the arbitration lost capture register

BITS[1]					DECIMAL VALUE	FUNCTION
ALC.4	ALC.3	ALC.2	ALC.1	ALC.0		
0	0	0	0	0	00	arbitration lost in bit 1 of identifier
0	0	0	0	1	01	arbitration lost in bit 2 of identifier
0	0	0	1	0	02	arbitration lost in bit 3 of identifier
0	0	0	1	1	03	arbitration lost in bit 4 of identifier
0	0	1	0	0	04	arbitration lost in bit 5 of identifier
0	0	1	0	1	05	arbitration lost in bit 6 of identifier
0	0	1	1	0	06	arbitration lost in bit 7 of identifier
0	0	1	1	1	07	arbitration lost in bit 8 of identifier
0	1	0	0	0	08	arbitration lost in bit 9 of identifier
0	1	0	0	1	09	arbitration lost in bit 10 of identifier
0	1	0	1	0	10	arbitration lost in bit SRTR; note 2
0	1	0	1	1	11	arbitration lost in bit IDE
0	1	1	0	0	12	arbitration lost in bit 12 of identifier; note 3
0	1	1	0	1	13	arbitration lost in bit 13 of identifier; note 3
0	1	1	1	0	14	arbitration lost in bit 14 of identifier; note 3
0	1	1	1	1	15	arbitration lost in bit 15 of identifier; note 3
1	0	0	0	0	16	arbitration lost in bit 16 of identifier; note 3
1	0	0	0	1	17	arbitration lost in bit 17 of identifier; note 3
1	0	0	1	0	18	arbitration lost in bit 18 of identifier; note 3
1	0	0	1	1	19	arbitration lost in bit 19 of identifier; note 3
1	0	1	0	0	20	arbitration lost in bit 20 of identifier; note 3
1	0	1	0	1	21	arbitration lost in bit 21 of identifier; note 3
1	0	1	1	0	22	arbitration lost in bit 22 of identifier; note 3
1	0	1	1	1	23	arbitration lost in bit 23 of identifier; note 3
1	1	0	0	0	24	arbitration lost in bit 24 of identifier; note 3
1	1	0	0	1	25	arbitration lost in bit 25 of identifier; note 3
1	1	0	1	0	26	arbitration lost in bit 26 of identifier; note 3
1	1	0	1	1	27	arbitration lost in bit 27 of identifier; note 3
1	1	1	0	0	28	arbitration lost in bit 28 of identifier; note 3
1	1	1	0	1	29	arbitration lost in bit 29 of identifier; note 3
1	1	1	1	0	30	arbitration lost in bit RTR; note 3
1	1	1	1	1	31	arbitration lost in bit 12 of identifier; note 3

Notes (Table 3.24):

1. Binary coded frame bit number where arbitration was lost.
2. Bit RTR for standard frame messages.
3. Extended frame messages only.

3.6.9 ERROR CODE CAPTURE REGISTER (ECC)

This register contains information about the type and location of errors on the bus. The error code capture register appears to the CPU as a read only memory.

Table 3.25 Bit interpretation of the error code capture register (ECC); base address + 19H

BIT	SYMBOL	NAME	VALUE	FUNCTION
ECC.7[1]	ERRC1	Error Code 1	-	-
ECC.6[1]	ERRC0	Error Code 0	-	-
ECC.5	DIR	Direction	1	RX; error occurred during reception
			0	TX; error occurred during transmission
ECC.4[2]	SEG4	Segment 4	-	-
ECC.3[2]	SEG3	Segment 3	-	-
ECC.2[2]	SEG2	Segment 2	-	-
ECC.1[2]	SEG1	Segment 1	-	-
ECC.0[2]	SEG0	Segment 0	-	-

Notes (Table 3.25):

1. For bit interpretation of bits ECC.7 and ECC.6 see Table 3.26.

2. For bit interpretation of bits ECC.4 to ECC.0 see Table 3.27.

Table 3.26 Bit interpretation of bits ECC.7 and ECC.6

BIT ECC.7	BIT ECC.6	FUNCTION
0	0	bit error
0	1	form error
1	0	stuff error
1	1	other type or error

Table 3.27 Bit interpretation of bits ECC.4 to ECC.0; note 1

BIT ECC.4	BIT ECC.3	BIT ECC.2	BIT ECC.1	BIT ECC.0	FUNCTION
0	0	0	1	1	start of frame
0	0	0	1	0	ID.28 to ID.21
0	0	1	0	0	ID.20 to ID.18
0	0	1	0	1	bit IDE
0	0	1	1	1	ID.17 to ID.13
0	1	1	1	1	ID.12 to ID.5
0	1	1	1	0	ID.4 to ID.0
0	1	1	0	0	bit RTR
0	1	1	0	1	reserved bit 1
0	1	0	0	1	reserved bit 0
0	1	0	1	1	data length code
0	1	0	1	0	data field
0	1	0	0	0	CRC sequence
1	1	0	0	0	CRC delimiter
1	1	0	0	1	acknowledge slot
1	1	0	1	1	acknowledge delimiter
1	1	0	1	0	end of frame
1	0	0	1	0	intermission
1	0	0	0	1	active error flag
1	0	1	1	0	passive error flag

1	0	0	1	1	tolerate dominant bits
1	0	1	1	1	error delimiter
1	1	1	1	1	overload flag

Note (Table 3.27):

1. Bit settings reflect the current frame segment to distinguish between different error events.

If a bus error occurs, the corresponding bus error interrupt is always forced, if enabled. At the same time, the current position of the bit stream processor is captured into the error code capture register. The content within this register is fixed until the users software has read out its content once. The capture mechanism is then activated again. The corresponding interrupt flag located in the interrupt register is cleared during the read access to the interrupt register. A new bus error interrupt is not possible until the capture register is read out once.

3.6.10 ERROR WARNING LIMIT REGISTER (EWLR)

The error warning limit can be defined within this register. The default value (after hardware reset) is 96. In reset mode this register appears to the CPU as a read/write memory. In operating mode it is read only. Note, that a content change of the EWLR is only possible, if the reset mode was entered previously. An error status change (see status register; Table 3.20) and an error warning interrupt forced by the new register content will not occur until the reset mode is cancelled again.

Table 3.28 **Bit interpretation of the error warning limit register (EWLR); base address + 1BH**

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
EWL.7	EWL.6	EWL.5	EWL.4	EWL.3	EWL.2	EWL.1	EWL.0

3.6.11 RX ERROR COUNTER REGISTER (RXERR)

The RX error counter register reflects the current value of the receive error counter. After a hardware reset this register is initialized to logic 0. In operating mode this register appears to the CPU as a read only memory. A write access to this register is possible only in reset mode.

If a bus-off event occurs, the RX error counter is initialized to logic 0. The time bus-off is valid, writing to this register has no effect.

Note, that a CPU-forced content change of the RX error counter is only possible, if the reset mode was entered previously. An error status change (see status register; Table 3.20), an error warning or an error passive interrupt forced by the new register content will not occur, until the reset mode is cancelled again.

Table 3.29 Bit interpretation of the RX error counter register (RXERR); base address + 1DH

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
RXERR.7	RXERR.6	RXERR.5	RXERR.4	RXERR.3	RXERR.2	RXERR.1	RXERR.0

3.6.12 TX ERROR COUNTER REGISTER (TXERR)

The TX error counter register reflects the current value of the transmit error counter.

In operating mode this register appears to the CPU as a read only memory. A write access to this register is possible only in reset mode. After a hardware reset this register is initialized to logic 0. If a bus-off event occurs, the TX error counter is initialized to 127 to count the minimum protocol-defined time (128 occurrences of the bus-free signal). Reading the TX error counter during this time gives information about the status of the bus-off recovery.

If bus-off is active, a write access to TXERR in the range from 0 to 254 clears the bus-off flag and the controller will wait for one occurrence of 11 consecutive recessive bits (bus-free) after the reset mode has been cleared.

Table 3.30 Bit interpretation of the TX error counter register (TXERR); base address + 1FH

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
TXERR.7	TXERR.6	TXERR.5	TXERR.4	TXERR.3	TXERR.2	TXERR.1	TXERR.0

Writing 255 to TXERR allows to initiate a CPU-driven bus-off event. It should be noted that a CPU-forced content change of the TX error counter is only possible, if the reset mode was entered previously. An error or bus status change (see status register; Table 3.20), an error warning or an error passive interrupt forced by the new register content will not occur until the reset mode is cancelled again. After leaving the reset mode, the new TX counter content is interpreted and the bus-off event is performed in the same way, as if it was forced by a bus error event. That means, that the reset mode is entered again, the TX error counter is initialized to 127, the RX counter is cleared and all concerned status and interrupt register bits are set.

Clearing of reset mode now will perform the protocol-defined bus-off recovery sequence (waiting for 128 occurrences of the bus-free signal).

If the reset mode is entered again before the end of bus-off recovery (TXERR > 0), bus-off keeps active and TXERR is frozen.

3.6.13 TRANSMIT BUFFER

The global layout of the transmit buffer is shown in Figure 3-4. One has to distinguish between the Standard Frame Format (SFF) and the Extended Frame Format (EFF) configuration. The transmit buffer allows the definition of one transmit message with up to eight data bytes.

3.6.13.1 Transmit buffer layout

The transmit buffer layout is subdivided into descriptor and data fields where the first byte of the descriptor field is the frame information byte (frame information). It describes the frame format (SFF or EFF), remote or data frame and the data length. Two identifier bytes for SFF or four bytes for EFF messages follow. The data field contains up to eight data bytes. The transmit buffer has a length of 13 bytes and is located in the CAN address range from 21 to 39. Note, that a direct access to the transmit buffer RAM is possible using the CAN address space from C1 to D9. This RAM area is reserved for the transmit buffer. The three following bytes may be used for general purposes (CAN address DB, DD and DF).

3.6.13.2 Descriptor field of the transmit buffer

The bit layout of the transmit buffer is represented in Table 3.31 to Table 3.33 for SFF and Table 3.34 to Table 3.38 for EFF. The given configuration is chosen to be compatible with the receive buffer layout (see insert reference).

base address +	21	TX frame information
	23	TX identifier 1
	25	TX identifier 2
	27	TX data byte 1
	29	TX data byte 2
	2B	TX data byte 3
	2D	TX data byte 4
	2F	TX data byte 5
	31	TX data byte 6
	33	TX data byte 7
	35	TX data byte 8
	37	unused
	39	unused

a. standard frame format

base address +	21	TX frame information
	23	TX identifier 1
	25	TX identifier 2
	27	TX identifier 3
	29	TX identifier 4
	2B	TX data byte 1
	2D	TX data byte 2
	2F	TX data byte 3
	31	TX data byte 4
	33	TX data byte 5
	35	TX data byte 6
	37	TX data byte 7
	39	TX data byte 8

a. Extended frame format

Figure 3-4 Transmit buffer layout for standard and extended frame format configurations

Table 3.31 TX frame information (SFF); base address + 21H

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
FF[1]	RTR[2]	X[3]	X[3]	DLC.3[4]	DLC.2[4]	DLC.1[4]	DLC.0[4]

Notes (Table 3.31):

1. Frame format.
2. Remote transmission request.
3. Don’t care; recommended to be compatible to receive buffer (0) in case of using the self reception facility (self test).
4. Data length code bit.

Table 3.32 TX identifier 1 (SFF); base address + 23H; note 1

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
ID.28	ID.27	ID.26	ID.25	ID.24	ID.23	ID.22	ID.21

Note (Table 3.32):

1. ID.X means identifier bit X.

Table 3.33 TX identifier 2 (SFF); base address + 25H; note 1

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
ID.20	ID.19	ID.18	X[2]	X[3]	X[3]	X[3]	X[3]

Notes (Table 3.33):

1. ID.X means identifier bit X.
2. Don’t care; recommended to be compatible to receive buffer (RTR) in case of using the self reception facility (self test).
3. Don’t care; recommended to be compatible to receive buffer (0) in case of using the self reception facility (self test).

Table 3.34 TX frame information (EFF); base address + 21H

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
FF[1]	RTR[2]	X[3]	X[3]	DLC.3[4]	DLC.2[4]	DLC.1[4]	DLC.0[4]

Notes (Table 3.34):

1. Frame format.
2. Remote transmission request.
3. Don't care; recommended to be compatible to receive buffer (0) in case of using the self reception facility (self test).
4. Data length code bit.

Table 3.35 TX identifier 1 (EFF); base address + 23H; note 1

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
ID.28	ID.27	ID.26	ID.25	ID.24	ID.23	ID.22	ID.21

Note (Table 3.35):

1. ID.X means identifier bit X.

Table 3.36 TX identifier 2 (EFF); base address + 25H; note 1

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
ID.20	ID.19	ID.18	ID.17	ID.16	ID.15	ID.14	ID.13

Note (Table 3.36):

1. ID.X means identifier bit X.

Table 3.37 TX identifier 3 (EFF); base address + 27H; note 1

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
ID.12	ID.11	ID.10	ID.9	ID.8	ID.7	ID.6	ID.5

Note (Table 3.37):

1. ID.X means identifier bit X.

Table 3.38 TX identifier 4 (EFF); base address + 29H; note 1

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
ID.4	ID.3	ID.2	ID.1	ID.0	X[2]	X[3]	X[3]

Note (Table 3.38):

1. ID.X means identifier bit X.
2. Don't care; recommended to be compatible to receive buffer (RTR) in case of using the self reception facility (self test).
3. Don't care; recommended to be compatible to receive buffer (0) in case of using the self reception facility (self test).

Table 3.39 Frame Format (FF) and Remote Transmission Request (RTR) bits

BIT	VALUE	FUNCTION
FF	1	EFF; extended frame format will be transmitted by the CAN controller
	0	SFF; standard frame format will be transmitted by the CAN controller
RTR	1	remote; remote frame will be transmitted by the CAN controller
	0	data; data frame will be transmitted by the CAN controller

3.6.13.3 Data Length Code (DLC)

The number of bytes in the data field of a message is coded by the data length code. At the start of a remote frame transmission the data length code is not considered due to the RTR bit being logic 1 (remote). This forces the number of transmitted/received data bytes to be 0. Nevertheless, the data length code must be specified correctly to avoid bus errors, if two CAN controllers start a remote frame transmission with the same identifier simultaneously.

The range of the data byte count is 0 to 8 bytes and is coded as follows:

DataByteCount = 8 x DLC.3 + 4 x DLC.2 + 2 x DLC.1 + DLC.0

For reasons of compatibility no data length code >8 should be used. If a value >8 is selected, 8 bytes are transmitted in the data frame with the Data Length Code specified in DLC.

3.6.13.4 Identifier (ID)

In Standard Frame Format (SFF) the identifier consists of 11 bits (ID.28 to ID.18) and in Extended Frame Format (EFF) messages the identifier consists of 29 bits (ID.28 to ID.0). ID.28 is the most significant bit, which is transmitted first on the bus during the arbitration process. The identifier acts as the message’s name, used in a receiver for acceptance filtering, and also determines the bus access priority during the arbitration process.

The lower the binary value of the identifier the higher the priority. This is due to the larger number of leading dominant bits during arbitration.

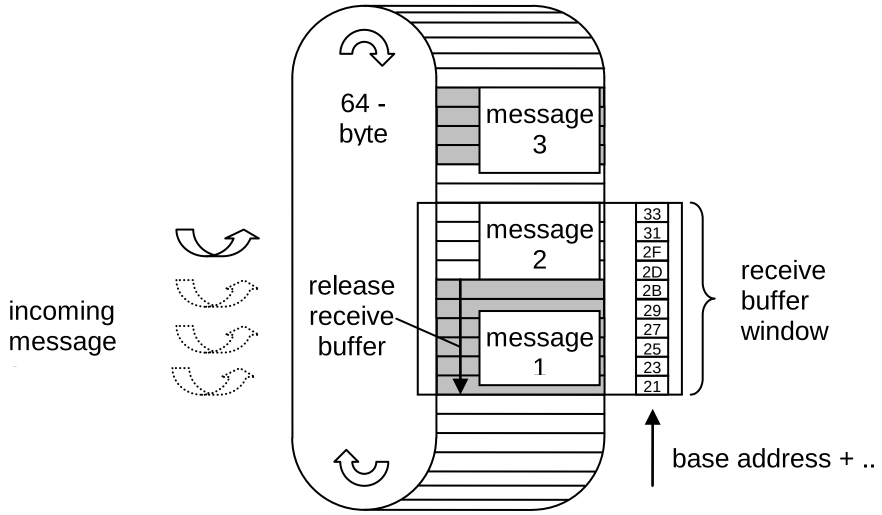
3.6.13.5 Data field

The number of transferred data bytes is defined by the data length code. The first bit transmitted is the most significant bit of data byte 1 at CAN address 19 (SFF) or CAN address 21 (EFF).

3.6.14 RECEIVE BUFFER

The global layout of the receive buffer is very similar to the transmit buffer described in the previous section. The receive buffer is the accessible part of the RXFIFO and is located in the range between CAN address 16 and 28. Each message is subdivided into a descriptor and a data field.

3.6.14.1 Descriptor field of the receive buffer



Message 1 is now available in the receive buffer.

Figure 3-5 Example of the message storage in the RXFIFO

The bit layout of the receive buffer is represented in Table 3.40 to Table 3.42 for SFF and Table 3.43 to Table 3.47 for EFF. The given configuration is chosen to be compatible with the transmit buffer layout (see Descriptor field of the transmit buffer 3.6.13.2).

Table 3.40 RX frame information (SFF); base address + 21H

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
FF[1]	RTR[2]	0	0	DLC.3[3]	DLC.2[3]	DLC.1[3]	DLC.0[3]

Notes (Table 3.40):

1. Frame format.
2. Remote transmission request.
3. Data length code bit.

Table 3.41 RX identifier 1 (SFF); base address + 23H; note 1

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
ID.28	ID.27	ID.26	ID.25	ID.24	ID.23	ID.22	ID.21

Note (Table 3.41):

1. ID.X means identifier bit X.

Table 3.42 RX identifier 2 (SFF); base address + 25H; note 1

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
ID.20	ID.19	ID.18	RTR[2]	0	0	0	0

Notes (Table 3.42):

1. ID.X means identifier bit X.
2. Remote transmission request.

Table 3.43 RX frame information (EFF); base address + 21H

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
FF[1]	RTR[2]	0	0	DLC.3[3]	DLC.2[3]	DLC.1[3]	DLC.0[3]

Notes (Table 3.43)

1. Frame format.
2. Remote transmission request.
3. Data length code bit.

Table 3.44 RX identifier 1 (EFF); base address + 23H; note 1

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
ID.28	ID.27	ID.26	ID.25	ID.24	ID.23	ID.22	ID.21

Note (Table 3.44):

1. ID.X means identifier bit X.

Table 3.45 RX identifier 2 (EFF); base address + 25H; note 1

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
ID.20	ID.19	ID.18	ID.17	ID.16	ID.15	ID.14	ID.13

Note (Table 3.45):

1. ID.X means identifier bit X.

Table 3.46 RX identifier 3 (EFF); base address + 27H; note 1

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
ID.12	ID.11	ID.10	ID.9	ID.8	ID.7	ID.6	ID.5

Note (Table 3.46):

1. ID.X means identifier bit X.

Table 3.47RX identifier 4 (EFF); base address + 29H; note 1

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
ID.4	ID.3	ID.2	ID.1	ID.0	RTR[2]	0	0

Notes (Table 3.47):

1. ID.X means identifier bit X.
2. Remote transmission request.

Remark: the received data length code located in the frame information byte represents the real sent data length code, which may be greater than 8 (depends on sender). Nevertheless the maximum number of received data bytes is 8. This should be taken into account by reading a message from the receive buffer.

As described in Figure 3-4 the RXFIFO has space for 64 message bytes in total. It depends on the data length how many messages can fit in it at one time. If there is not enough space for a new message within the RXFIFO, the CAN controller generates a data overrun condition the moment this message becomes valid and the acceptance test was positive. A message which is partly written into the RXFIFO, when the data overrun situation occurs, is deleted. This situation is indicated to the CPU via the status register and the data overrun interrupt, if enabled.

3.6.15 ACCEPTANCE FILTER

With the help of the acceptance filter the CAN controller is able to allow passing of received messages to the RXFIFO only when the identifier bits of the received message are equal to the predefined ones within the acceptance filter registers.

The acceptance filter is defined by the Acceptance Code Registers (ACRn) and the Acceptance Mask Registers (AMRn). The bit patterns of messages to be received are defined within the acceptance code registers. The corresponding acceptance mask registers allow to define certain bit positions to be ‘don’t care’.

Two different filter modes are selectable within the mode register (MOD.3, AFM; see section 3.6.3):

- Single filter mode (bit AFM is logic 1)
- Dual filter mode (bit AFM is logic 0).

3.6.15.1 Single filter configuration

In this filter configuration one long filter (4-bytes) could be defined. The bit correspondences between the filter bytes and the message bytes depend on the currently received frame format.

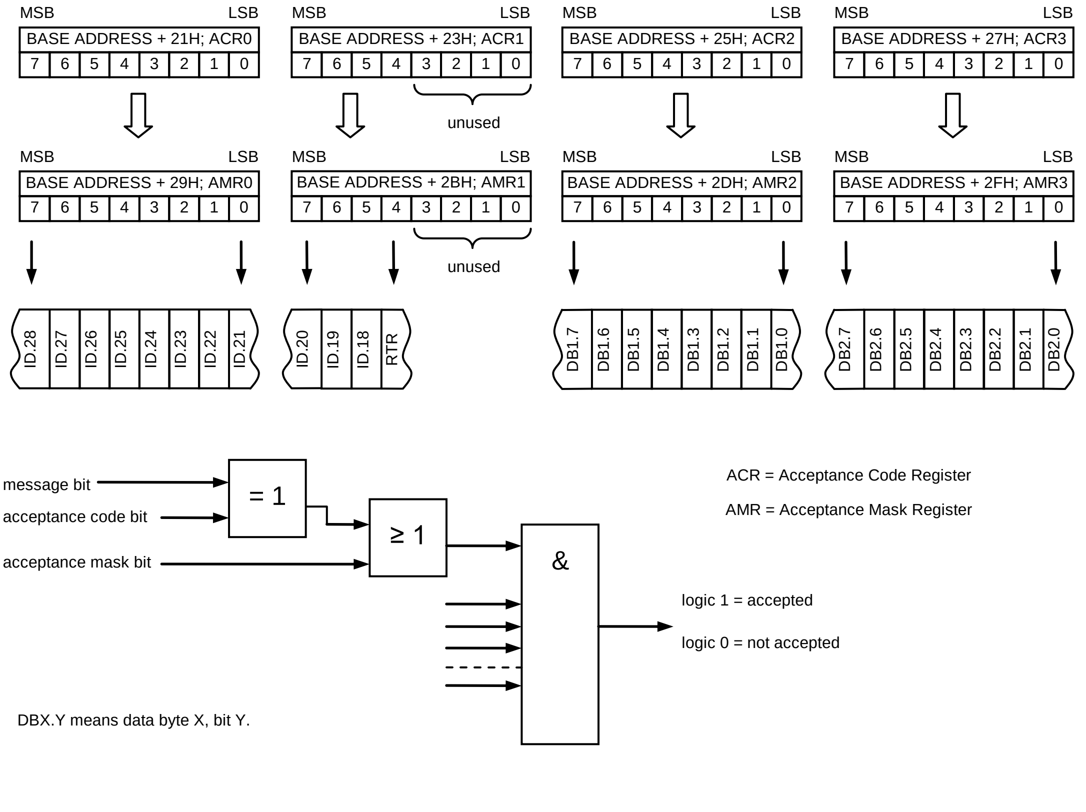


Figure 3-6 Single filter configuration, receiving standard frame messages

Standard frame: if a standard frame format message is received, the complete identifier including the RTR bit and the first two data bytes are used for acceptance filtering. Messages may also be accepted if there are no data bytes existing due to a set RTR bit or if there is none or only one data byte because of the corresponding data length code.

For a successful reception of a message, all single bit comparisons have to signal acceptance. Note, that the 4 least significant bits of AMR1 and ACR1 are not used. In order to be compatible with future products these bits should be programmed to be 'don't care' by setting AMR1.3, AMR1.2, AMR1.1 and AMR1.0 to logic 1.

Extended frame: if an extended frame format message is received, the complete identifier including the RTR bit is used for acceptance filtering.

For a successful reception of a message, all single bit comparisons have to signal acceptance.

It should be noted that the 2 least significant bits of AMR3 and ACR3 are not used. In order to be compatible with future products these bits should be programmed to be 'don't care' by setting AMR3.1 and AMR3.0 to logic 1.

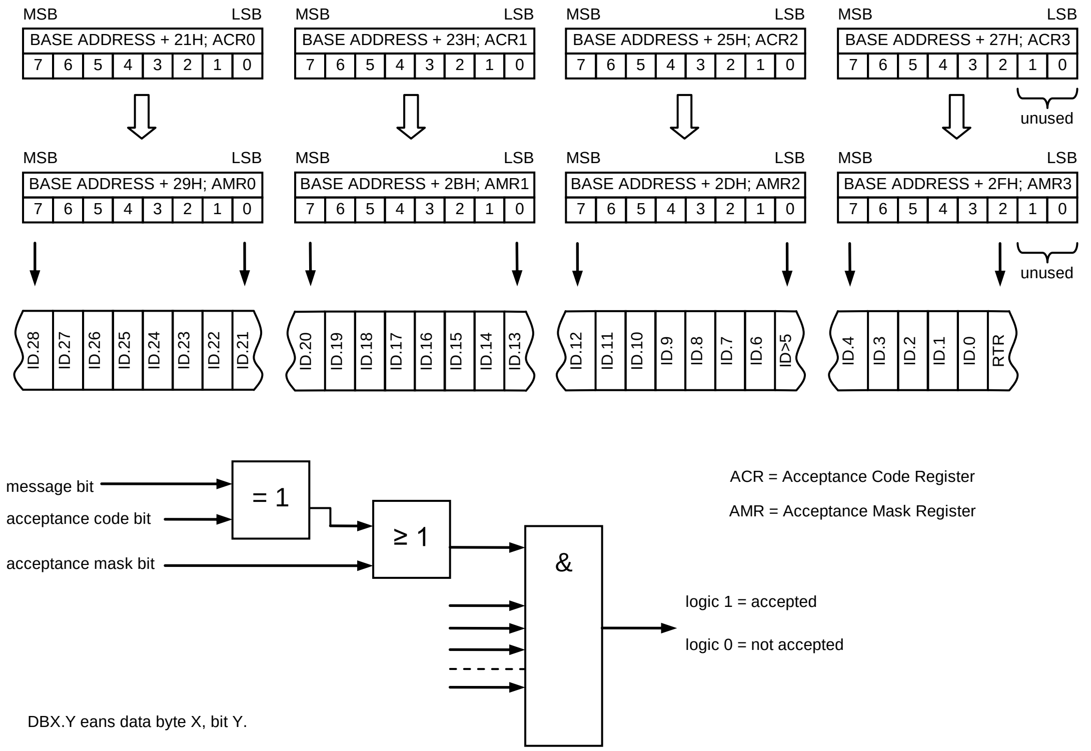


Figure 3-7 Single filter configuration, receiving extended frame messages.

3.6.15.2 Dual filter configuration

In this filter configuration two short filters can be defined. A received message is compared with both filters to decide, whether this message should be copied into the receive buffer or not. If at least one of the filters signals an acceptance, the received message becomes valid. The bit correspondences between the filter bytes and the message bytes depends on the currently received frame format.

Standard frame: if a standard frame message is received, the two defined filters are looking different. The first filter compares the complete standard identifier including the

RTR bit and the first data byte of the message. The second filter just compares the complete standard identifier including the RTR bit.

For a successful reception of a message, all single bit comparisons of at least one complete filter have to signal acceptance. In case of a set RTR bit or a data length code of logic 0 no data byte is existing. Nevertheless a message may pass filter 1, if the first part up to the RTR bit signals acceptance.

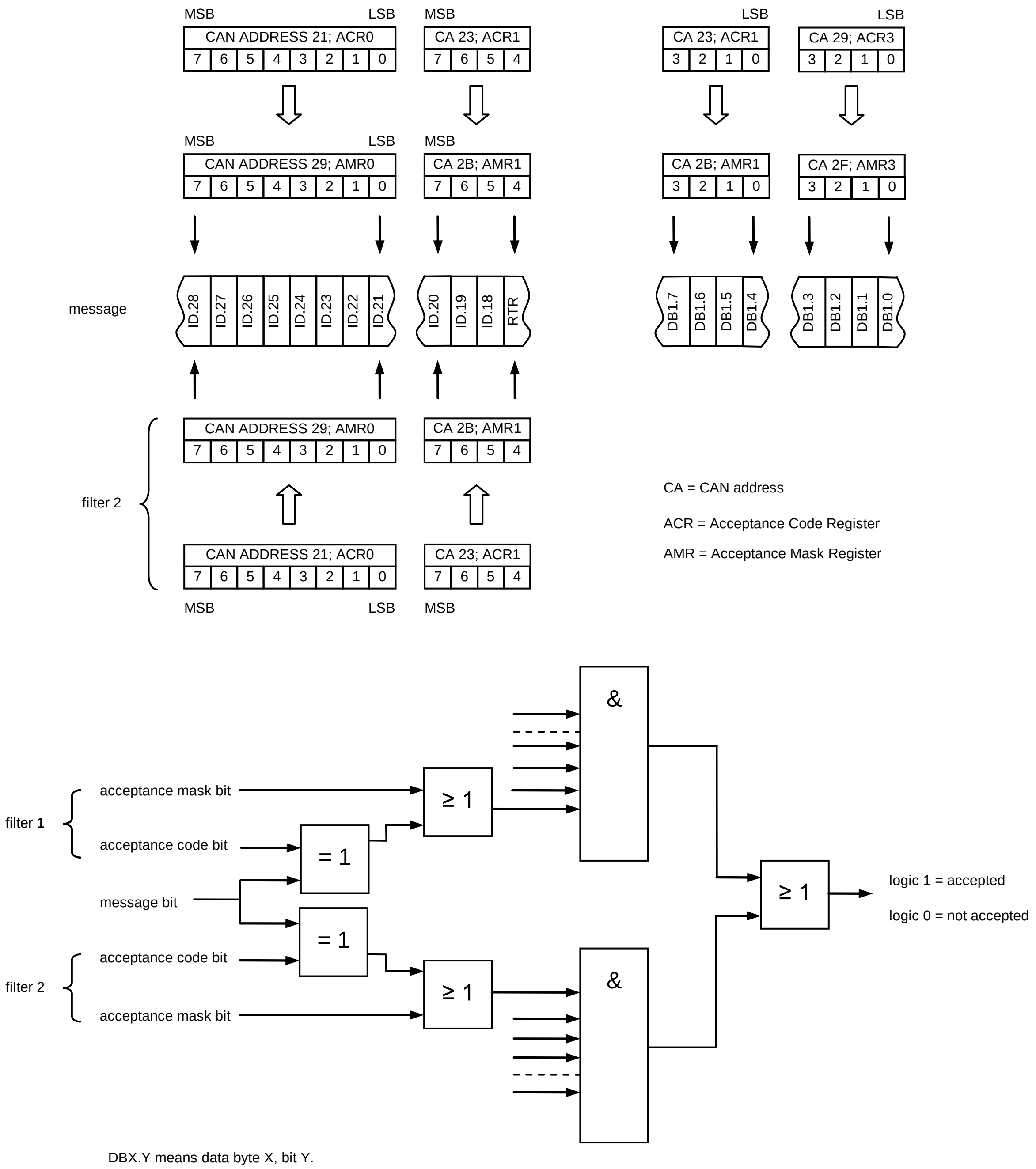


Figure 3-8 Dual filter configuration, receiving standard frame messages.

If no data byte filtering is required for filter 1, the four least significant bits of AMR1 and AMR3 have to be set to logic 1 (don’t care). Then both filters are working identically using the standard identifier range including the RTR bit.

Extended frame: if an extended frame message is received, the two defined filters are looking identically. Both filters are comparing the first two bytes of the extended identifier range only. For a successful reception of a message, all single bit comparisons of at least one complete filter have to indicate acceptance.

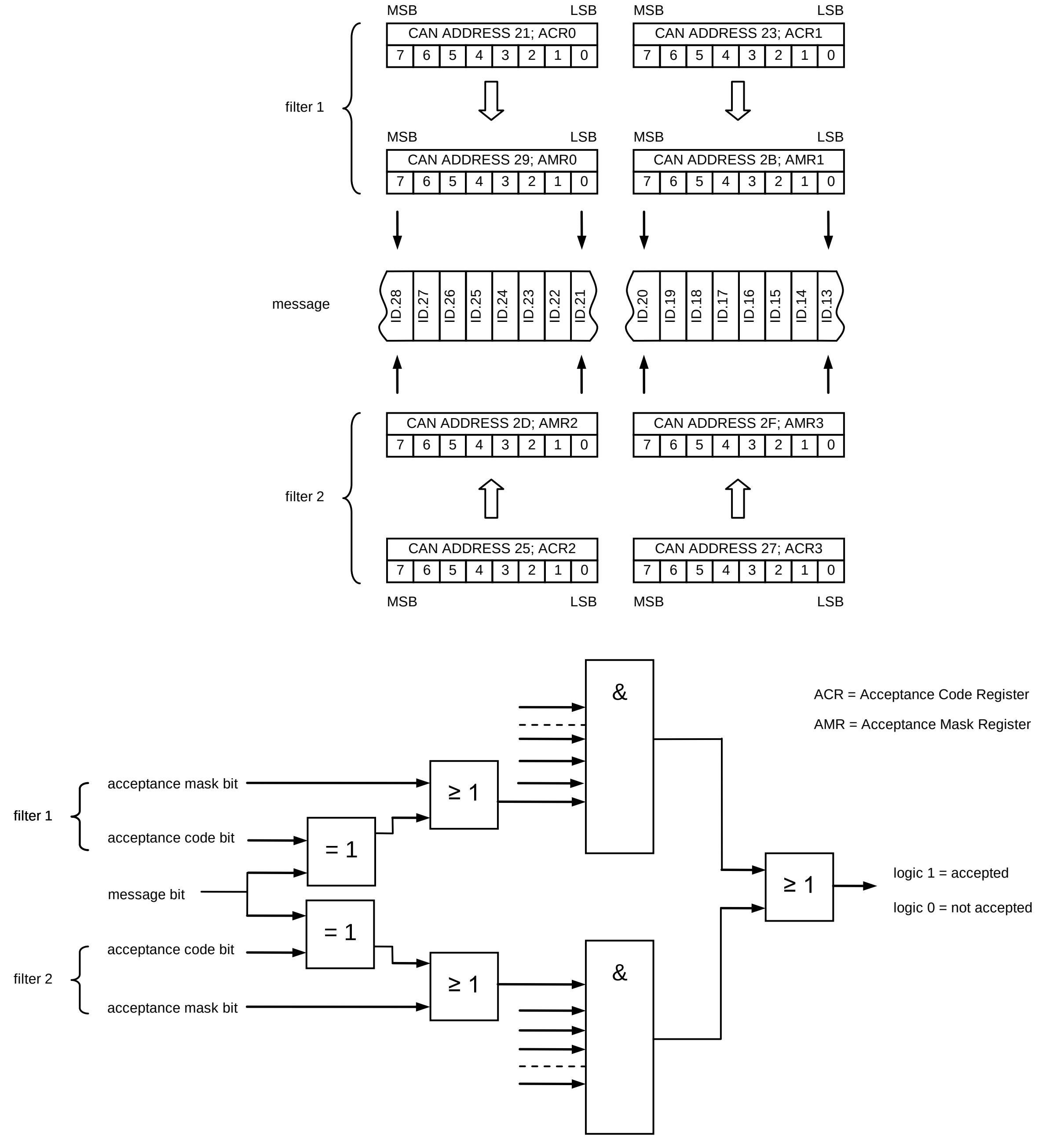


Figure 3-9 Dual filter configuration, receiving extended frame messages.

3.6.16 RX MESSAGE COUNTER (RMC)

The RMC register (base address + 3BH) reflects the number of messages available within the RXFIFO. The value is incremented with each receive event and decremented by the release receive buffer command. After any reset event, this register is cleared.

Table 3.48 Bit interpretation of the RX message counter (RMC); base address + 3BH

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
0[1]	0[1]	0[1]	RMC.4	RMC.3	RMC.2	RMC.1	RMC.0

Note (Table 3.48):

1. This bit cannot be written. During read-out of this register always a zero is given.

3.6.17 RX BUFFER START ADDRESS REGISTER (RBSA)

The RBSA register (base address + 3DH) reflects the currently valid internal RAM address, where the first byte of the received message, which is mapped to the receive buffer window, is stored. With the help of this information it is possible to interpret the internal RAM contents.

The internal RAM address area begins at base address + 41H and may be accessed by the CPU for reading and writing (writing in reset mode only).

Example: if RBSA is set to 24 (decimal), the current message visible in the receive buffer window (base address + 21H to base address + 39H) is stored within the internal RAM beginning at RAM address 24. Because the RAM is also mapped directly to the IP Memory space base address + 41H (equal to RAM address 0) this message may also be accessed using base address + 71H.

(IP Memory Space address = base address + RBSA * 2 + 41H > base address + 24 * 2 + 41H = base address + 71H).

If a message exceeds RAM address 63, it continues at RAM address 0.

The release receive buffer command is always given while there is at least one more message available within the FIFO. RBSA is updated to the beginning of the next message.

On hardware reset, this pointer is initialized to ‘00H’. Upon a software reset (setting of reset mode) this pointer keeps its old value, but the FIFO is cleared; this means that the RAM contents are not changed, but the next received (or transmitted) message will override the currently visible message within the receive buffer window.

The RX buffer start address register appears to the CPU as a read only memory in operating mode and as read/write memory in reset mode. It should be noted that a write access to RBSA takes effect first after the next positive edge of the internal clock frequency, which is half of the external oscillator frequency.

Table 3.49 Bit interpretation of the RX buffer start address register (RBSA); base address + 3DH

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
0[1]	0[1]	RBSA.5	RBSA.4	RBSA.3	RBSA.2	RBSA.1	RBSA.0

Note (Table 3.49):

1. This bit cannot be written. During read-out of this register always a zero is given.

3.7 Common registers

3.7.1 BUS TIMING REGISTER 0 (BTR0)

The contents of the bus timing register 0 defines the values of the Baud Rate Prescaler (BRP) and the Synchronization Jump Width (SJW). This register can be accessed (read/write) if the reset mode is active.

In operating mode this register is read only, if the PeliCAN mode is selected. In BasicCAN mode a ‘FFH’ is reflected.

Table 3.50 Bit interpretation of bus timing register 0 (BTR0); base address + 0DH

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
SJW.1	SJW.0	BRP.5	BRP.4	BRP.3	BRP.2	BRP.1	BRP.0

3.7.1.1 Baud Rate Prescaler (BRP)

The period of the CAN system clock tscl is programmable and determines the individual bit timing. The CAN system clock is calculated using the following equation:

tscl = 83.33 ns x (32 x BRP.5 + 16 x BRP.4 + 8 x BRP.3 + 4 x BRP.2 + 2 x BRP.1 + BRP.0 + 1)

3.7.1.2 Synchronization Jump Width (SJW)

To compensate for phase shifts between clock oscillators of different bus controllers, any bus controller must re-synchronize on any relevant signal edge of the current transmission. The synchronization jump width defines the maximum number of clock cycles a bit period may be shortened or lengthened by one re-synchronization:

tSJW = tscl x (2 x SJW.1 + SJW.0 + 1)

3.7.1.3 BUS TIMING REGISTER 1 (BTR1)

The contents of bus timing register 1 defines the length of the bit period, the location of the sample point and the number of samples to be taken at each sample point. This register can be accessed (read/write) if the reset mode is active.

In operating mode, this register is read only, if the PeliCAN mode is selected. In BasicCAN mode a ‘FFH’ is reflected.

Table 3.51 Bit interpretation of bus timing register 1 (BTR1); base address + 0FH

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
SAM	TSEG2.2	TSEG2.1	TSEG2.0	TSEG1.3	TSEG1.2	TSEG1.1	TSEG1.0

Table 3.52 Sampling (SAM)

BIT	VALUE	FUNCTION
SAM	1	triple; the bus is sampled three times; recommended for low/medium speed buses (class A and B) where filtering spikes on the bus line is beneficial
	0	single; the bus is sampled once; recommended for high speed buses (SAE class C)

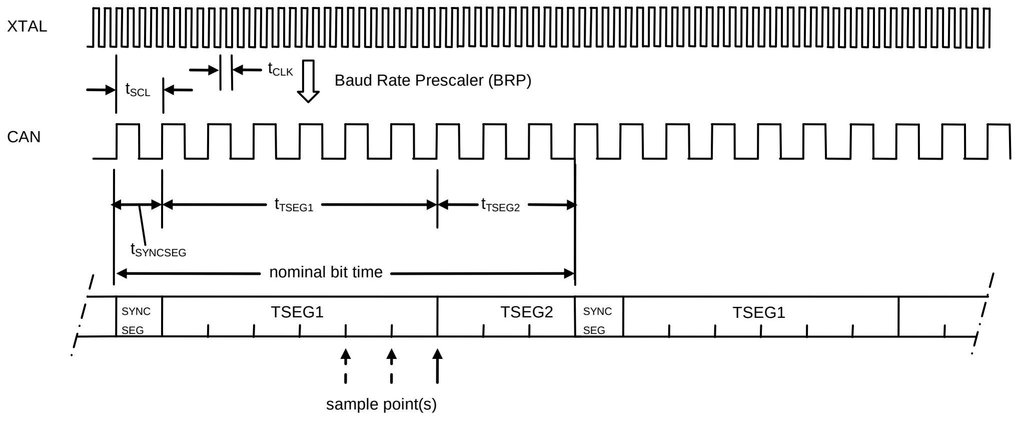
3.7.1.4 Time Segment 1 (TSEG1) and Time Segment 2 (TSEG2)

TSEG1 and TSEG2 determine the number of clock cycles per bit period and the location of the sample point, where:

tSYNCSEG = 1 x tscl

tTSEG1 = tscl x (8 x TSEG1.3 + 4 x TSEG1.2 + 2 x TSEG1.1 + TSEG1.0 + 1)

tTSEG2 = tscl x (4 x TSEG2.2 + 2 x TSEG2.1 + TSEG2.0 + 1)



Possible values are BRP = 000001, TSEG1 = 0101 and TSEG2 = 010.

Figure 3-10 General structure of a bit period

3.7.2 OUTPUT CONTROL REGISTER (OCR)

The output control register allows the set-up of different output driver configurations under software control. This register may be accessed read/write) if the reset mode is active. In operating mode, this register is read only, if the PeliCAN mode is selected. In BasicCAN mode a ‘FFH’ is reflected. This register must be written with the value ‘1AH’ for IP560 series module compatibility.

Table 3.53 Bit interpretation of the output control register (OCR); base address + 11H

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
OCTP1	OCTN1	OCPOL1	OCTP0	OCTN0	OCPOL0	OCMODE1	OCMODE0

If the SJA1000 is in the sleep mode a recessive level is output on the TX0 and TX1 pins with respect to the contents within the output control register. If the SJA1000 is in the reset state (reset request = HIGH) or the external reset pin RST is pulled LOW the outputs TX0 and TX1 are floating.

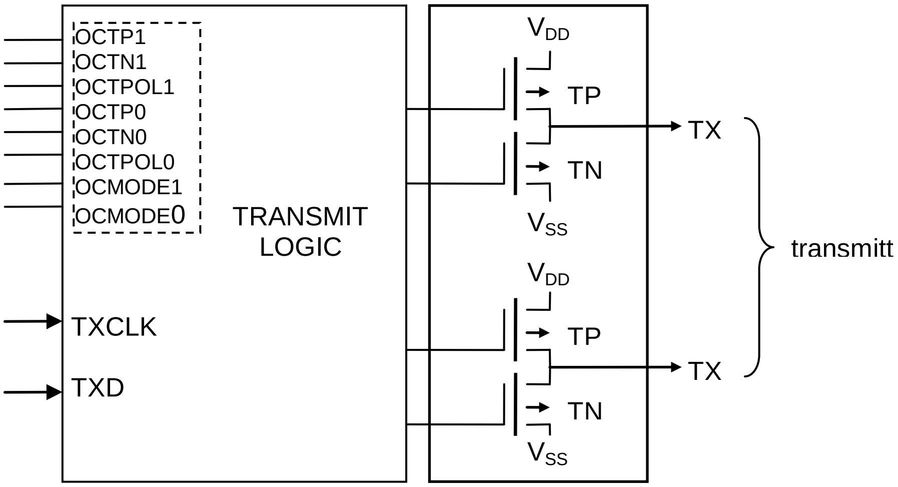


Figure 3-11 Transceiver input/output control logic.

The transmit output stage is able to operate in different modes. Table 3.54 shows the output control register settings.

Table 3.54 Interpretation of OCMODE bits

OCMODE1	OCMODE0	FUNCTION
0	0	bi-phase output mode; note 1
0	1	test output mode; note 2
1	0	normal output mode
1	1	clock output mode; note 3

Note (Table 3.54):

1. Do not use bi-phase mode. This mode is for transformer coupled applications. It is not appropriate for the IP560 series modules.
2. In test output mode, TXn will reflect the bit detected on RX pins with the next positive edge of the system clock. TN1, TN0, TP1 and TP0 are configured in accordance with the setting of OCR.
3. Since the TX1 output pin is not connected in the IP560 series modules, this mode is effectively the same as normal output mode.

3.7.2.1 Normal output mode

In normal output mode the bit sequence (TXD) is sent via TX0 and TX1. The voltage levels on the output driver pins TX0 and TX1 depend on both the driver characteristic programmed by OCTPx, OCTNx (float, pull-up, pull-down, push-pull) and the output polarity programmed by OCPOLx.

3.7.2.2 Clock output mode

For the TX0 pin this is the same as in normal output mode. However, the data stream to TX1 is replaced by the transmit clock (TXCLK). The rising edge of the transmit clock (non-inverted) marks the beginning of a bit period. The clock pulse width is 1 x tscl. The TX1 output pin is not connected in the IP560 series modules.

3.7.2.3 Bi-phase output mode

In contrast to the normal output mode the bit representation is time variant and toggled. If the bus controllers are galvanically decoupled from the bus line by a transformer, the bit stream is not allowed to contain a DC component. This is achieved by the following scheme.

During recessive bits all outputs are deactivated (floating). Dominant bits are sent with alternating levels on TX0 and TX1, i.e. the first dominant bit is sent on TX0, the second is sent on TX1, and the third one is sent on TX0 again, and so on. This mode is not appropriate for the IP560 series modules since they are not transformer coupled.

3.7.2.4 Test output mode

In test output mode the level connected to RX is reflected at TXn with the next positive edge of the system clock corresponding to the programmed polarity in the output control register. Table 3.55 shows the relationship between the bits of the output control register and the output pins TX0 and TX1.

Table 3.55 Output pin configuration; note 1

DRIVE	TXD	OCTPX	OCTNX	OCPOLX	TPX[2]	TNX[3]	TXX[4]
Float	X	0	0	X	off	off	float
Pull-down	0	0	1	0	off	on	LOW
	1	0	1	0	off	off	float
	0	0	1	1	off	off	float
	1	0	1	1	off	on	LOW
Pull-up	0	1	0	0	off	off	float
	1	1	0	0	on	off	HIGH
	0	1	0	1	on	off	HIGH
	1	1	0	1	off	off	float
Push-pull	0	1	1	0	off	on	LOW
	1	1	1	0	on	off	HIGH
	0	1	1	1	on	off	HIGH
	1	1	1	1	off	on	LOW

Notes (Table 3.55):

1. X = don’t care.
2. TPX is the on-chip output transistor X, connected to VDD.
3. TNX is the on-chip output transistor X, connected to VSS.
4. TXX is the serial output level on pin TX0 or TX1. It is required that the output level on the CAN-bus line is dominant when TXD = 0 and recessive when TXD = 1.

3.7.3 CLOCK DIVIDER REGISTER (CDR)

The clock divider register controls the CLKOUT frequency for the CPU and allows to deactivate the CLKOUT pin. Additionally a dedicated receive interrupt pulse on TX1, a receive comparator bypass and the selection between BasicCAN mode and PeliCAN mode is made here. The default state of the register after hardware reset is divide-by-2.

On software reset (reset request/reset mode) this register is not influenced.

The reserved bit (CDR.4) will always reflect a logic 0. The application software should always write a logic 0 to this bit in order to be compatible with future features, which may be 1-active using this bit.

Table 3.56 Bit interpretation of the clock divider register (CDR); base address + 3FH

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
CAN mode	CBP[1]	RXINTEN[2]	(0) [3]	clock off[4]	CD.2[5]	CD.1[5]	CD.0[5]

Note (Table 3.56):

1. Since the RX1 input on the IP560 series modules is connected to ground, this bit should be set to bypass the CAN input comparator.
2. The TX1 output is not connected in the IP560 series modules. Setting this bit has no effect on the operation of the module.
3. This bit cannot be written. During read-out of this register always a zero is given.
4. The CLKOUT pin is not connected in the IP560 series modules. This bit should be set to disable the clock output.
5. The CLKOUT pin is not connected in the IP560 series modules.

3.7.3.1 CD.2 to CD.0

The bits CD.2 to CD.0 are used to define the frequency at the external CLKOUT pin. Since the CLKOUT pin is not connected in the IP560 series modules, these bits have no effect on the operation of the module.

3.7.3.2 Clock off

Setting this bit allows the external CLKOUT pin of the SJA1000 to be disabled. A write access is possible only in reset mode. If this bit is set, CLKOUT is LOW during sleep mode, otherwise it is HIGH. Since the CLKOUT pin is not connected in the IP560 series modules, the Clock off bit should be set to disable the CLKOUT output.

3.7.3.3 RXINTEN

This bit allows the TX1 output to be used as a dedicated receive interrupt output. When a received message has passed the acceptance filter successfully, a receive interrupt pulse with the length of one bit time is always output at the TX1 pin (during the last bit of end of frame). The transmit output stage should operate in normal output mode. A write access is only possible in reset mode. The TX1 output is not connected in the IP560 series modules.

3.7.3.4 CBP

Setting of CDR.6 allows to bypass the CAN input comparator and is only possible in reset mode. This is useful in the event that the SJA1000 is connected to an external transceiver circuit. The internal delay of the SJA1000 is reduced, which will result in a longer maximum possible bus length. If CBP is set, only RX0 is active. The RX1 input is connected to ground in the IP560 series modules. The CBP bit should be set to bypass the input comparator.

3.7.3.5 CAN mode

CDR.7 defines the CAN mode. If CDR.7 is at logic 0 the CAN controller operates in BasicCAN mode. If set to logic 1 the CAN controller operates in PeliCAN mode. Write access is only possible in reset mode.

3.8 TRANSCEIVER CONTROL

The CAN physical layer is implemented with a NXP TJA1043 high speed CAN transceiver.

3.8.1 TRANSCEIVER OPERATING MODES

The TJA1043 can be operated in five modes, each with specific features. Control bits STB and EN in Control Register 0 (see 3.2.1) select the operating mode. Changing between modes also gives access to a number of diagnostics flags, available via pin ERR in the Status Register (see 3.2.3). The following sections describe the five operating modes. Table 3.57 shows the conditions for selecting these modes. Figure 3-12 illustrates the mode transitions when V_{BAT} is present.

Table 3.57 Operating mode selection

Control pins		**Internal flags**			**Operating mode**
STB[1]	EN	UV_{NOM}[2]	UV_{BAT}	Wake[3]	
From Normal, Listen-only, Standby, and Go-to-Sleep modes					
X[4]	X	Set	X	X	Sleep mode
0	X	cleared	set	X	Standby mode
1	X	cleared	X	set	Standby mode
1	0	cleared	X	cleared	Standby mode
1	1	cleared	X	cleared	Go-to-Sleep mode[5]
0	0	cleared	cleared	X	Listen-only mode
0	1	cleared	cleared	X	Normal mode
From Sleep mode					
X	X	Set	X	X	Sleep mode
0	X	cleared	set	X	Standby mode
1	X	cleared	X	set	Standby mode
1	X	cleared	X	cleared	Sleep mode
0	0	cleared	cleared	X	Listen-only mode
0	1	cleared	cleared	X	Normal mode

Notes: (Table 3.57)

1. A HIGH-TO_LOW transition on pin STB will clear the UV_{NOM} flag
2. Setting the UV_{NOM} flag will clear the WAKE flag.
3. Setting the Wake flag will clear the UV_{NOM} flag.
4. X = don't care.
5. After $t_{h(min)}$ in Go-to-Sleep mode, the transceiver will enter Sleep mode.

3.8.1.1 NORMAL MODE

Normal mode is the mode for normal bidirectional CAN communication. The receiver will convert the differential analog bus signal on pins CANH and CANL into digital data, available for output to the SJA1000 CAN controller. The transmitter will convert digital data from the SJA1000 CAN controller into a differential analog signal, available for output to the bus pins. The bus pins are biased at 2.5 VDC.

3.8.1.2 Pwon/listen-only mode

In pwon/listen-only mode the transmitter of the transceiver is disabled, effectively providing a transceiver listen-only behavior. The receiver will still convert the analog bus signal on pins CANH and CANL into digital data, available for output to the SJA1000 CAN controller. As in normal mode the bus pins are biased at 2.5 V_{DC}.

3.8.1.3 Standby mode

The Standby mode is the first-level power saving mode of the transceiver, offering reduced current consumption. In Standby mode the transceiver is not able to transmit or receive data and the low-power receiver is activated to monitor bus activity. The bus pins are biased at ground level.

3.8.1.4 Go-to-sleep command mode

The go-to-sleep command mode is the controlled route for entering Sleep mode. In go-to-sleep command mode the transceiver behaves as if in Standby mode, plus a go-to-sleep command is issued to the transceiver. After remaining in go-to-sleep command mode for the minimum hold time ($t_{h(min)}$), the transceiver will enter Sleep mode. The transceiver will not enter the Sleep mode if the state of bits STB and EN in Control Register 0 (see 3.2.1) is changed or the UV_{BAT}, pwon or wake-up flag is set before $t_{h(min)}$ has expired.

3.8.1.5 Sleep Mode

The Sleep mode is the second-level power saving mode of the transceiver. Sleep mode is entered via the go-to-sleep command mode, and also when the undervoltage detection time on either V_{CC} or $V_{I/O}$ elapses before that voltage level has recovered. In Sleep mode the transceiver still behaves as described for Standby mode. The current into pin V_{BAT} is reduced to a minimum. Waking up a node from Sleep mode is possible via the wake-up flag and (as long as the UV_{NOM} flag is not set) via bit STB in Control Register 0 (see 3.2.1).

3.8.2 Transceiver Internal Flags

The TJA1043 makes use of seven internal flags for its fail-safe fallback mode control and system diagnosis support. Table 3.58 shows the relation between flags and operating modes of the transceiver. Five of the internal flags can be made available to the host via the ERR bit in the Status Register (see 3.2.3). Table 3.58 shows the details on how to access these flags. The following sections describe the seven internal flags.

Table 3.58 Accessing TJA1043 internal flags via ERR bit in Status Register

Internal Flag	Flag is available at ERR[1] bit	Flag is cleared
UV_{NOM}	No	by setting the Pwon or Wake flags, by a LOW-to-HIGH transition on STB or when both $V_{I/O}$ and V_{BAT} have recovered.
UV_{BAT}	No	when V_{BAT} has recovered
pwon	in listen-only mode (coming from Standby mode, go-to-sleep command mode, or Sleep mode)	on entering normal mode
wake-up	in Standby mode, go-to-sleep command mode, and Sleep mode	on entering normal mode, or by setting the UV_{NOM} flag
wake-up source	in normal mode (before the fourth dominant to recessive edge on pin TXD[2])	on leaving normal mode
bus failure	in normal mode (after the fourth dominant to recessive edge on pin TXD[2])	on re-entering normal mode or by setting the Pwon flag
local failure	in listen-only mode (coming from normal mode)	on entering normal mode or when RXD is dominant while TXD is recessive (provided that all local failures are resolved) or by setting the Pwon flag

Notes: (Table 3.58)

1. Pin ERR_N is an active-LOW output, so a LOW-level indicates a set flag and a HIGH-level indicates cleared flag.
2. Allow for a TXD dominant time of at least 4 µs per dominant-recessive cycle.

3.8.2.1 UV_{NOM} flag

UV_{NOM} is the V_{CC} and V_{IO} undervoltage detection flag. The flag is set when the voltage on pin V_{CC} drops below Vuvd(V_{CC}) for longer than $t_{det(uv)}$, or when the voltage on pin V_{IO} drops below $V_{uvd(VIO)}$ for longer than $t_{det(uv)}$. When the UV_{NOM} flag is set, the transceiver enters Sleep mode to save power and to ensure the bus is not disturbed. Any wake-up request, setting the Pwon flag or a LOW-to-HIGH transition on STB will clear UV_{NOM} and the timers. UV_{NOM} will also be cleared if both V_{CC} and V_{IO} recover for longer than $t_{rec(uv)}$. The transceiver will then switch to the operating mode indicated by the logic levels on pins STB and EN (see Table 3.58).

3.8.2.2 UV_{BAT} flag

UVBAT is the VBAT undervoltage detection flag. The flag is set when the voltage on pin VBAT drops below VBAT(stb). When UVBAT is set, the transceiver will try to enter Standby mode to save power and not disturb the bus. UVBAT is cleared when the voltage on pin VBAT has recovered. The transceiver will then return to the operating mode determined by the logic state of bits STB and EN in Control Register 0 (see 3.2.1).

3.8.2.3 Pwon flag

Pwon is the V_{BAT} power-on flag. This flag is set when the voltage on pin V_{BAT} has recovered after it dropped below V_{BAT}(pwon), particularly after the transceiver was disconnected from the battery. By setting the pwon flag, the UV_{NOM} flag and timers are cleared and the transceiver cannot enter Sleep mode. In pwon/listen-only mode the pwon flag can be made available at bit ERR in the Status Register (see 3.2.3). The flag is cleared when the transceiver enters normal mode.

3.8.2.4 Wake-Up flag

The wake-up flag is set when the transceiver detects a local or a remote wake-up request. A local wake-up request is detected when a logic state change on pin V_{BAT} remains stable for at least twake. A remote wake-up request is detected when the bus remains in dominant state for at least t_{BUS}. The wake-up flag can only be set in Standby mode, go-to-sleep command mode or Sleep mode. Setting of the flag is blocked during the UV_{NOM} flag waiting time. By setting the wake-up flag, the UV_{NOM} flag and timers are cleared. The wake-up flag is immediately available on pins at bit ERR in the Status Register (see 3.2.3) (provided that $V_{I/O}$ and V_{CC} are present). The flag is cleared at power-on, or when the UV_{NOM} flag is set or the transceiver enters normal mode.

3.8.2.5 Wake-up source flag

Wake-up source recognition is provided via the wake-up source flag, which is set when the wake-up flag is set by a local wake-up request via pin V_{BAT}. The wake-up source flag can only be set after the pwon flag is cleared. In normal mode the wake-up source flag can be made available at bit ERR in the Status Register (see 3.2.3). The flag is cleared at power-on or when the transceiver leaves normal mode.

3.8.2.6 Bus failure flag

The bus failure flag is set if the transceiver detects a bus line short-circuit condition to V_{BAT}, V_{CC} or GND during four consecutive dominant-recessive cycles on pin TXD, when trying to drive the bus lines dominant. In normal mode the bus failure flag can be made available at bit ERR in the Status Register (see 3.2.3). The flag is cleared when the transceiver re-enters normal mode.

3.8.2.7 Local failure flag

In normal mode or pwon/listen-only mode the transceiver can recognize five different local failures, and will combine them into one local failure flag. The five local failures are: TXD dominant clamping, RXD recessive clamping, a TXD-to-RXD short circuit, bus dominant clamping, and overtemperature. The nature and detection of these local failures is described in 3.8.3. In pwon/listen-only mode the local failure flag can be made available at bit ERR in the Status Register (see 3.2.3). The flag is cleared when entering normal mode or when RXD is dominant while TXD is recessive, provided that all local failures are resolved.

3.8.3 Local Failures

The TJA1043 can detect five different local failure conditions. Any of these failures will set the local failure flag, and in most cases the transmitter of the transceiver will be disabled. The following sections give the details.

3.8.3.1 TXD dominant clamping detection

A permanent LOW level on pin TXD (due to a hardware or software application failure) would drive the CAN bus into a permanent dominant state, blocking all network communication. The TXD dominant time-out function prevents such a network lock-up by disabling the transmitter of the transceiver if pin TXD remains at a LOW level for longer than the TXD dominant time-out tdom(TXD). The tdom(TXD) timer defines the minimum possible bit rate of 40 kbit/s. The transmitter remains disabled until the local failure flag is cleared.

3.8.3.2 RXD recessive clamping detection

An RXD pin clamped to HIGH level will prevent the controller connected to this pin from recognizing a bus dominant state. So the controller can start messages at any time, which is likely to disturb all bus communication. RXD recessive clamping detection prevents this effect by disabling the transmitter when the bus is in dominant state without RXD reflecting this. The transmitter remains disabled until the local failure flag is cleared.

3.8.3.3 TXD to RXD short circuit detection

A short-circuit between pins RXD and TXD would keep the bus in a permanent dominant state once the bus is driven dominant, because the low-side driver of RXD is typically stronger than the high-side driver of the controller connected to TXD. The TXD-to-RXD short-circuit detection prevents such a network lock-up by disabling the transmitter. The transmitter remains disabled until the local failure flag is cleared.

3.8.3.4 Bus dominant clamping detection

A CAN bus short circuit (to V_{BAT}, V_{CC} or GND) or a failure in one of the other network nodes could result in a differential voltage on the bus high enough to represent a bus dominant state. Because a node will not start transmission if the bus is dominant, the normal bus failure detection will not detect this failure, but the bus dominant clamping detection will. The local failure flag is set if the dominant state on the bus persists for longer than $t_{dom(bus)}$. By checking this flag, the controller can determine if a clamped bus is blocking network communication. There is no need to disable the transmitter. Note that the local failure flag does not retain a bus dominant clamping failure, and is released as soon as the bus returns to recessive state.

3.8.3.5 Overtemperature detection

To protect the output drivers of the transceiver against overheating, the transmitter will be disabled if the virtual junction temperature exceeds the shutdown junction temperature $T_{j(sd)}$. The transmitter remains disabled until the local failure flag is cleared.

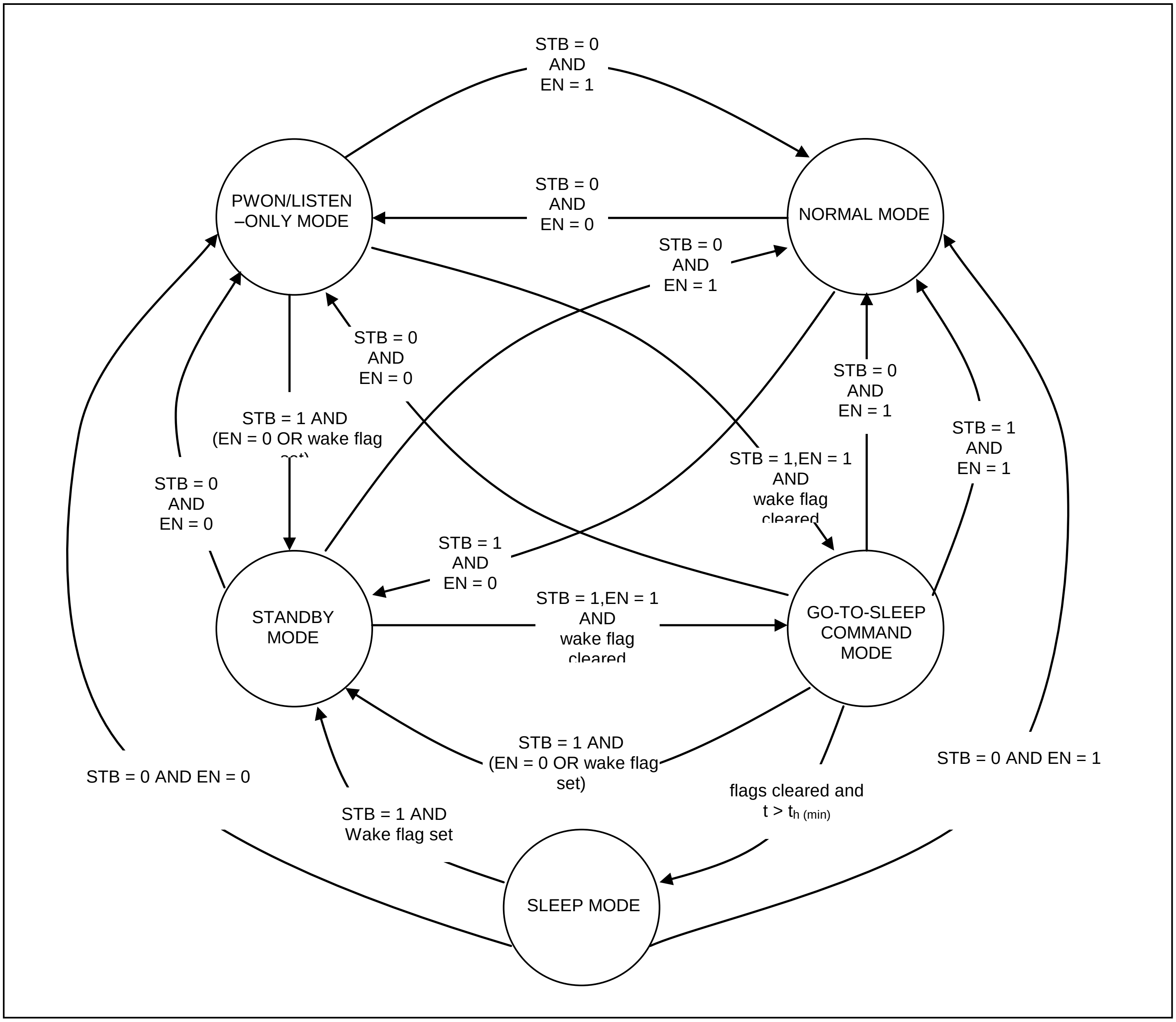


Figure 3-12 TJA1043 Mode transitions when valid V_{CC}, $V_{I/O}$, and V_{BAT} voltages are present

3.9 PROGRAMMING INTERRUPTS

Interrupts generated by the IP560 use interrupt request lines INTREQ0* (Interrupt Request 0) for CAN channel 0 and INTREQ1* (Interrupt Request 1) for CAN channel 1.

The IP560 Interrupt Vector register can be used as a pointer to an interrupt handling routine. The vector is an 8-bit value and can be used to point to any one of 256 possible locations to access the interrupt handling routine.

To enable a CAN channel to generate an interrupt, the interrupt enable bit in Control Register 0 (see 3.2.1) for the appropriate channel must be set. In addition, the interrupt enable bit for each of the potential interrupt sources must be set in either the control register (see 3.5.2) if operating in BasicCAN mode or the interrupt enable register (see 3.6.7) if operating in PeliCAN mode for the appropriate channel.

The INTREQ0* or INTREQ1* line is released when the interrupt register for the appropriate channel is read. If the 'receive' interrupt bit is set then a 'release receive buffer' command must be issued to the appropriate channel. (See 3.5.5 and 3.6.6).

4 THEORY OF OPERATION

This section contains information regarding the IP560 hardware. A description of the basic functionality of the circuitry used on the board is also provided. Refer to the IP560 and IP560-I block diagrams at the end of this manual as you review this material.

4.1 LOGIC/POWER INTERFACE

The logic interface to the carrier board is made through connector P1 (refer to Table 2.3). The P1 interface also provides +5V and +12V power to the module. Note that the DMA control, ERROR∗, and STROBE∗ signals are not used.

A Field Programmable Gate-Array (FPGA) installed on the IP Module provides an interface to the carrier board. The interface to the carrier board allows complete control of all IP560 functions.

4.2 IP INTERFACE LOGIC

IP interface logic of the IP560 is imbedded within the FPGA. This logic includes: address decoding, I/O and ID read/write control circuitry, and ID PROM implementation.

Address decoding of the nine IP address signals A(1:9) is implemented in the FPGA, in conjunction with the IP select signals, to identify access to the IP modules MEM, INT, ID or I/O spaces. In addition, the byte strobes BS0∗ and BS1∗ are decoded to identify low byte, high byte, or double byte data transfers.

The carrier to IP module interface implements access to both MEM, INT, ID and I/O space via 16 or 8-bit data transfers. Read only access to ID space provides the identification for the individual module (as given in Table 3.1) per the IP specification. Read and write access to the I/O space provides a means to control the IP560 and monitor status. Reads and writes to MEM space provide access to the SJA1000 CAN controllers.

The timing for access to the various memory spaces is shown in Table 4.1.

Table 4.1 Access Times

Address Space	Wait States	
	Read	Write
INT	1	0
ID	1	N/A
I/O	1	0
MEM (8 MHz IP clock)	3	2
MEM (32 MHz IP clock)	5	2

4.3 IP560 CONTROL LOGIC

All logic to provide access to the SJA1000 CAN controllers and TJA1043 CAN transceivers is imbedded in the module's FPGA. Once the IP560 has been configured, the control logic performs the following:

- Provides access to theSJA1000 CAN controllers
- Provides mode control (enable / standby) and errors status for the TJA1043 transceivers
- Provides the 24 MHz clock to the SJA1000 CAN controllers.
- Issues interrupt requests to the carrier.

4.4 ISOLATION (IP560(E)-I only)

The IP560(E)-I models provide isolation between the host and the CAN bus channels. The CAN bus channels are also isolated from each other. Each channel has its own isolated DC/DC converter to provide power for its TJA1043 transceiver and on-board V_{BAT} power. Digital signals that interface between the FPGA and SJA100 CAN controller and the TJA1043 transceiver are isolated by ADuM3211 digital isolators.

4.5 CAN BUS TERMINATION

The CAN bus signals are stub terminated using split termination with 1.3 K ohm termination resistors connected to a 4.7 nF capacitor to ground in order to provide DC stabilization of the common mode voltage.

5 SERVICE AND REPAIR

5.1 SERVICE AND REPAIR ASSISTANCE

Surface-Mounted Technology (SMT) boards are generally difficult to repair. It is highly recommended that a non-functioning board be returned to Acromag for repair. The board can be easily damaged unless special SMT repair and service tools are used. Further, Acromag has automated test equipment that thoroughly checks the performance of each board. When a board is first produced and when any repair is made, it is tested, placed in a burn-in room at an elevated temperature, and retested before shipment.

Please refer to Acromag's Service Policy Bulletin or contact Acromag for complete details on how to obtain parts and repair.

5.2 PRELIMINARY SERVICE PROCEDURE

Before beginning repair, be sure that all of the procedures in Section 2, Preparation For Use, have been followed. Also, refer to the documentation of your carrier/CPU board to verify that it is correctly configured. Replacement of the board with one that is known to work correctly is a good technique to isolate a faulty board. Use the unmodified example we provide.

CAUTION: POWER MUST BE TURNED OFF BEFORE
REMOVING OR INSERTING BOARDS

5.3 WHERE TO GET HELP

If you continue to have problems, your next step should be to visit the Acromag worldwide web site at http://www.acromag.com. Our web site contains the most up-to-date product and software information.

Choose “Bus Board Products” then go to the “Support” tab in the Acromag banner to access:

Application Notes

- Frequently Asked Questions (FAQ’s)
- Knowledge Base
- Tutorials
- Software Updates/Drivers

An email question can also be submitted from within the Knowledge Base or directly from the “Contact Us” tab.

Acromag’s application engineers can also be contacted directly for technical assistance via telephone or FAX through the numbers listed below. When needed, complete repair services are also available.

Phone: 248-295-0310
Fax: 248-624-9234
Email: solutions@acromag.com

6 SPECIFICATIONS

6.1 GENERAL SPECIFICATIONS

Operating Temperature..............-0 to +70°C
-40 to +85°C (E Version)
Relative Humidity ....5-95% non-condensing.
Storage Temperature.................-55°C to +150°C.

Physical Configuration................Single Industrial I/O Pack Module.
Length ....3.880 inches (98.5 mm).
Width.....................................1.780 inches (45.2 mm).
Board Thickness ....0.062 inches (1.59 mm).
Max Component Height ....0.314 inches (7.97 mm).

Connectors:
P1 (IP Logic Interface) ....50-pin female receptacle header
(AMP 173279-3 or equivalent).
P2 (Field I/O).........................50-pin female receptacle header
(AMP 173279-3 or equivalent).

IP560 Non-Isolated.....................Logic and field commons have a direct electrical connection.

IP560-I Isolated ....Logic and field commons are isolated from each other and channel 0 connections are isolated from channel 1. The Murata DC/DC converters (part number NTE0509MC) used on the IOS-560-I are tested at 1 kVDC for 1 second.

Radiated Field Immunity (RFI)....Designed to comply with IEC61000-4-3: 2006 Level A (10V/m, 80 to 1000MHz AM).

Surge Immunity..........................Not required for signal I/O per European Norm EN61000-6-1.
Electric Fast Transient
Immunity (EFT) ....Designed to comply with IEC61000-4-4: 2007 Level 3 (2kV at field input and output terminals)

Radiated Emissions ....Designed to comply with CISPR 16-2-3 class A

Electrostatic Discharge (ESD)....Designed to comply with IEC6100-4-2: 2001 Level 2 (4kV contact discharge, 4kV air discharge)

6.2 INDUSTRIAL I/O PACK COMPLIANCE

Specification...............................This module meets or exceeds
ANSI/VITA 4-1995 specifications.
Electrical/Mechanical Interface..Single-Size IP Module.

IP Data Transfer Cycle Types Supported:
Input/Output (IOSel*).............D16 or D08 read/write of data.
ID Read (IDSel*) ....32 x 8 ID PROM read on D0..D7 as D16 or D08.
Interrupt Select (INTSel*)......8-bits (D08) Interrupt Vector Register contents.
Memory (MEMSel*)...............D16 or D08 read/write of 8-bit data on D0..D7
Access Times (8 MHz Clock):
ID PROM Read ....1 wait state (375 ns cycle).
I/O Space Read.....................1 wait state (375 ns cycle).
I/O Space Write.....................0 wait state (250 ns cycle).
Interrupt Select Read ....1 wait state (375 ns cycle).

Memory Space Read ....3 wait states (750 ns cycle).
Memory Space Write ....2 wait states (625 ns cycle).
Access Times (32 MHz Clock):
ID PROM Read ....1 wait state (94 ns cycle).
I/O Space Read.....................1 wait state (94 ns cycle).
I/O Space Write.....................0 wait state (63 ns cycle).
Interrupt Select Read ....1 wait state (94 ns cycle).
Memory Space Read ....5 wait states (250 ns cycle)
Memory Space Write ....2 wait states (156 ns cycle).

Power-Up Initialization Time 200mS Max. (During this time the IP module will ignore all signals.)

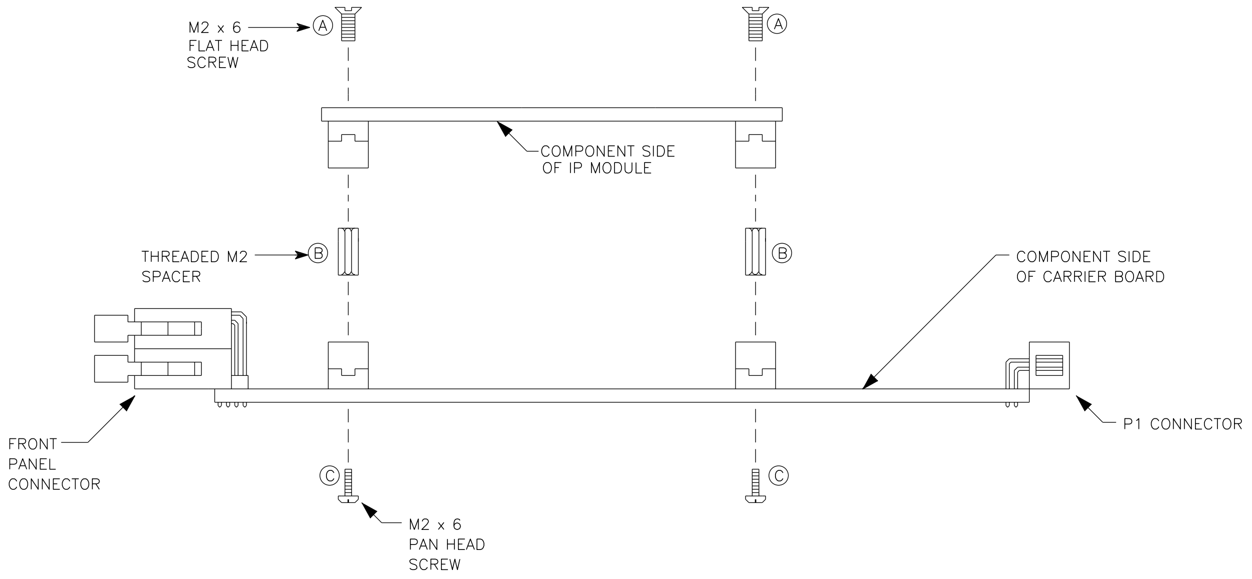
Table 6.1 Electrical Specifications

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
Bus lines pins (CAN_H_0, CAN_L_0, CAN_H_1, and CAN_L_1)						
$V_{O (DOM)}$	dominant output voltage	V_{TXD} = 0 V (dominant)				
		pins CAN_H_x	3	3.6	4.25	V
		Pins CAN_L_x	0.5	1.4	1.75	V
$V_{O (DOM)(m)}$	matching of dominant output voltage (V_{CC} - V_{CANH} - V_{CANL})		-0.1	-	+0.15	V
$V_{O (dif)(bus)}$	differential bus output voltage (V_{CANH} - V_{CANL})	V_{TXD} = 0 V (dominant); 45 Ω < RL < 65 Ω	1.5	-	3.0	V
		V_{TXD} = VI/O (recessive); no load	-50	-	-50	mV
$V_{O (reces)}$	recessive output voltage	normal or pwon/listen-only mode; V_{TXD} = $V_{I/O}$; no load	2	2.5	3	V
		Standby or Sleep mode; no load	-0.1	0	+0.1	V
$I_{O (sc)}$	short-circuit output current	V_{TXD} = 0 V (dominant)				
		pins CAN_H_x; V_{CANH} = 0 V	-45	-70	-95	mA
		Pins CAN_L_x; V_{CANL} = 40 V	45	70	95	mA
$I_{O (reces)}$	recessive output current	-27 V < V_{CAN} < 32 V	-2.5	-	+2.5	mA
$V_{diff(th)}$	differential receiver threshold voltage	normal or pwon/listen-only mode; see figure ? -12 V < V_{CANH} < 12 V -12 V < V_{CANL} < 12 V	0.5	0.7	0.9	V
		Standby or Sleep mode; -12 V < V_{CANH} < 12 V -12 V < V_{CANL} < 12 V	0.5	0.7	1.15	V
$V_{hys(dif)}$	differential receiver hysteresis voltage	normal or pwon/listen-only mode; see figure ? -12 V < V_{CANH} < 12 V -12 V < V_{CANL} < 12 V	50	70	100	mV
I_{LI}	input leakage current	VCC = 0 V; V_{CANH} = V_{CANL} = 5 V	100	170	250	µA
$R_{i(cm)}$	common mode input resistance		15	25	35	kΩ
$R_{i(cm)(m)}$	common mode input resistance matching		-3	0	+3	%
$R_{i(dif)}$	differential input resistance		25	50	75	kΩ
$C_{i(cm)}$	common mode input capacitance	V_{TXD} = V_{CC}; not tested	-	-	20	pF
$C_{i(dif)}$	differential input capacitance	V_{TXD} = V_{CC}; not tested	-	-	10	pF

$R_{SC(bus)}$	detectable short circuit resistance between bus lines and VBAT, VCC and GND	normal mode	0	-	50	Ω
Timing Characteristics						
$t_{dom(TXD)}$	TXD dominant time-out	V_{TXD} = 0 V	300	600	1000	µS
$t_{dom(bus)}$	bus dominant time-out	V_{dif} > 0.9 V	300	600	1000	µS
$t_{h(min)}$	minimum hold time of go-to-sleep command		20	35	50	µS
t_{bus}	dominant time for wake-up via bus	Standby or Sleep mode; V_{BAT} = 12 V	0.75	1.75	5	µS
twake	minimum wake-up time after receiving a falling or rising edge	Standby or Sleep mode; V_{BAT} = 12 V	5	25	50	µS
TJA1043 Transceiver Thermal Shutdown						
$T_{j(sd)}$	shutdown junction temperature		155	165	180	°C
Power Requirements						
	+5 Volts (±5%)	IP560(E) IP560(E)-I		92 230	110 275	mA
	+12 Volts (±5%)	IP560(E) IP560(E)-I		0.12	0.20	mA
	-12 Volts (±5%)				0	mA

7 APPENDIX

7.1 IP MODULE TO CARRIER BOARD ASSEMBLY



ASSEMBLY PROCEDURE:

1. THREADED SPACERS ARE PROVIDED FOR USE WITH AVME 9630/9660 CARRIER BOARDS (SHOWN). CHECK YOUR CARRIER BOARD TO DETERMINE ITS REQUIREMENTS. MOUNTING HARDWARE PROVIDED MAY NOT BE COMPATIBLE WITH ALL TYPES OF CARRIER BOARDS.
2. INSERT FLAT HEAD SCREWS (ITEM A) THROUGH SOLDER SIDE OF IP MODULE AND INTO HEX SPACERS (ITEM B) AND TIGHTEN (4 PLACES) UNTIL HEX SPACER IS COMPLETELY SEATED. THE RECOMMENDED TORQUE IS 0.226 NEWTON METER OR 2 INCH POUNDS. OVER TIGHTENING MAY DAMAGE CIRCUIT BOARD.
3. CAREFULLY ALIGN IP MODULE TO CARRIER BOARD AND PRESS TOGETHER UNTIL CONNECTORS AND SPACERS ARE SEATED.
4. INSERT PAN HEAD SCREWS (ITEM C) THROUGH SOLDER SIDE OF CARRIER BOARD AND INTO HEX SPACERS (ITEM B) AND TIGHTEN (4 PLACES). THE RECOMMENDED TORQUE IS 0.226 NEWTON METER OR 2 INCH POUNDS. OVER TIGHTENING MAY DAMAGE CIRCUIT BOARD.

IP MODULE TO CARRIER BOARD MECHANICAL ASSEMBLY 4501-434C

7.2 JUMPER LOCATIONS

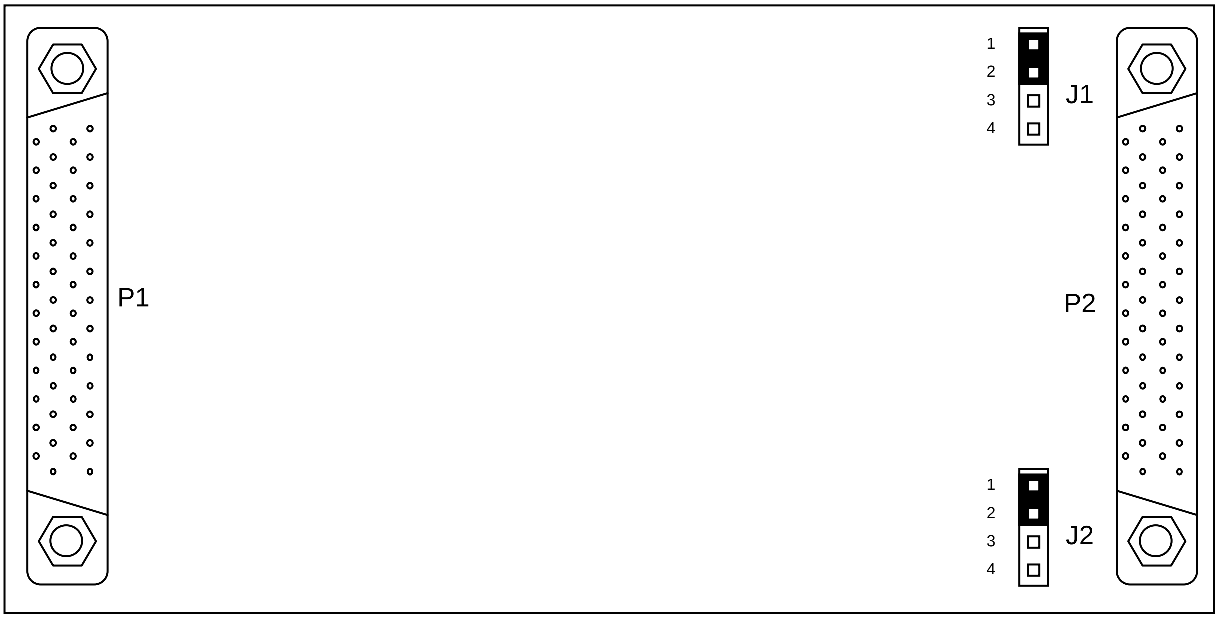


Figure 7-1 VBATT Source Selection

Table 7.1 Jumper Locations

Jumper Position	J1	J2
1 – 2 (factory default)	Channel 0 V_{BAT} connected to on-board power	Channel 1 V_{BAT} connected to on-board power
2 – 3	Do not install shunt in this position	Do not install shunt in this position
3 – 4	Channel 0 V_{BAT} connected to P2 pin 2 (V_{BAT})	Channel 1 V_{BAT} connected to P2 pin 27 (V_{BAT})