1. CAN Bus

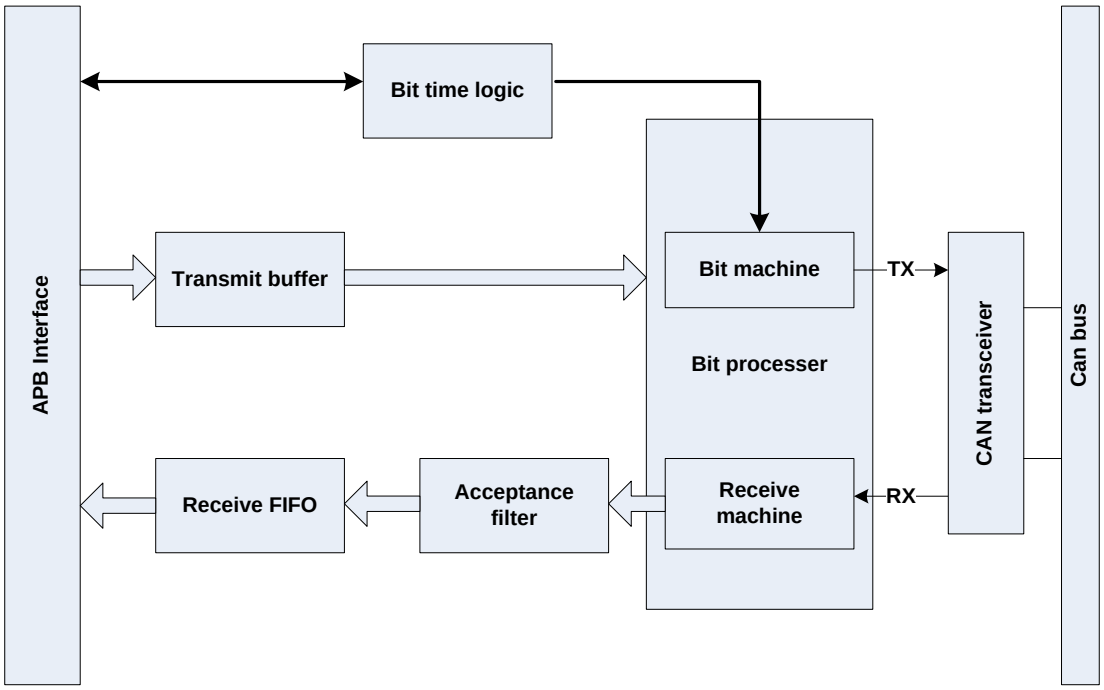
1.1. CAN Overview

The CAN module is a controller for the Controller Area Network (CAN) used in automotive and general industrial environments. It implements the CAN 2.0A/B protocol as defined in the BOSCH CAN bus specification 2.0.

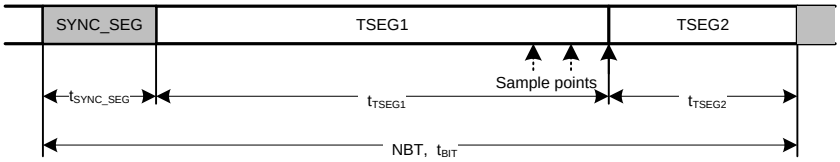
The CAN controller includes the following features:

- Supports industry-standard AMBA Peripheral Bus (APB) and it is fully compliant with the AMBA Specification, Revision 2.0
- Supports APB 32-bit bus width operation
- Supports the CAN 2.0A and 2.0B protocol specification
- Programmable data rate up to 1Mbps
- 64 byte receive buffers
- Support one-shot transmission option
- Supports two configurable filter modes
- Supports listen only mode
- Supports self-test mode

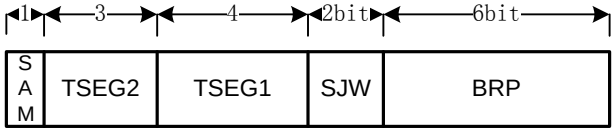
1.2. CAN System Block Diagram



1.3. CAN Bit Time Configuration



CAN bit timing segment



NBT x BPR = f_{base} / f_{canbus}, f_{base} = f_{osc} / 2 = 1 / (2 x t_{clk}), (NBT = 8~25 recommended)

TQ = 2 x t_{clk} x (32 x BRP.5 + 16 x BRP.4 + 8 x BRP.3 + 4 x BRP.2 + 2 x BRP.1 + BRP.0 + 1)

t_{clk} = 1/f_{osc}

$t_{syncseg}$ =1 x TQ

t_{tseg1} =TQ x (8 x TSEG1.3 + 4 x TSEG1.2 + 2 x TSEG1.1 + TSEG1.0 + 1)

t_{tseg2}=TQ x (4 x TSEG2.2 + 2 x TSEG2.1 + TSEG2.0 + 1)

1.4. CAN Controller Register List

Module Name	**Base Address**	
CAN controller	0x01C2BC00	

Register Name	**Offset**	**Description**
CAN_MSEL	0x0000	CAN mode select register
CAN_CMD	0x0004	CAN command register
CAN_STA	0x0008	CAN status register
CAN_INT	0x000C	CAN interrupt register
CAN_INTEN	0x0010	CAN interrupt enable register
CAN_BUSTIME	0x0014	CAN bus timing register
CAN_TEWL	0x0018	CAN TX error warning limit register
CAN_ERRC	0x001c	CAN error counter register
CAN_RMCNT	0x0020	CAN receive message counter register
CAN_RBUF_SADDR	0x0024	CAN receive buffer start address register
CAN_TRBUF0	0x0040	CAN TX/RX message buffer 0 register
CAN_TRBUF1	0x0044	CAN TX/RX message buffer 0 register
CAN_TRBUF2	0x0048	CAN TX/RX message buffer 0 register
CAN_TRBUF3	0x004c	CAN TX/RX message buffer 0 register

CAN_TRBUF4	0x0050	CAN TX/RX message buffer 0 register
CAN_TRBUF5	0x0054	CAN TX/RX message buffer 0 register
CAN_TRBUF6	0x0058	CAN TX/RX message buffer 0 register
CAN_TRBUF7	0x005c	CAN TX/RX message buffer 0 register
CAN_TRBUF8	0x0060	CAN TX/RX message buffer 0 register
CAN_TRBUF9	0x0064	CAN TX/RX message buffer 0 register
CAN_TRBUF10	0x0068	CAN TX/RX message buffer 0 register
CAN_TRBUF11	0x006c	CAN TX/RX message buffer 0 register
CAN_TRBUF12	0x0070	CAN TX/RX message buffer 0 register
CAN_ACPC	0x0040	CAN acceptance code 0 register
CAN_ACPM	0x0044	CAN acceptance mask 0 register
CAN_RBUF_RBACK	0x180~0x1b0	CAN transmit buffer for read back register

1.5. CAN Controller Register Description

1.5.1. CAN Mode Select Register

Offset: 0x0000			**Register Name: CAN_MOD_SEL** **Default Value: 0x0000_0001**
Bit	**R/W**	**Default**	**Description**
31:5	/	/	/
4	R/W	0	**SLEEP_SEL** **Sleep Mode** 1 - Sleep. The controller enters its Sleep Mode provided no CAN interrupt is pending and there is no bus activity. 0 - Wake-up (normal operation). If sleeping, the controller wakes up. *(This bit can only be written in Reset Mode)*
3	R/W	0	**ACP_FLT_MOD_SEL** **Acceptance Filter Mode Select** 1 - Single Filter. Receive data MOD.3 AFM Acceptance Filter Mode1 filtered using one 4-byte filter

			0 - Dual Filter. Receive data filtered using two shorter filters.
2	R/W	0	**LB_MOD** **Loopback Mode** 1 - Self Test enabled. 0 - Normal operation. An acknowledgement is required for successful transmission.
1	R/W	0	**LST_ONLY** **Listen Only Mode** 1 - Listen Only enabled. 0 - Normal operation. The error counters are stopped at the current value.
0	R/W	1	**RST** **Reset Mode** 1 – Reset mode selected. 0 - Normal operation. The controller returns to Operating Mode

1.5.2. CAN Command Register

Offset: 0x0004			**Register Name: CAN_CMD_REG** **Default Value: 0x0000_0000**
Bit	**R/W**	**Default**	**Description**
31:6	/	/	/
5	W	0	**BUS_OFF** **Bus off Request** Set this bit to 1 to initial a CPU-driven BUS OFF event.
4	W	0	**SELF_REQ** **Self Reception Request** Set this bit to 1 to make a message to be transmitted and received simultaneously
3	W	0	**CLR_OR_FLAG** **Clear Data Overrun Flag** Set this bit to 1 to clear the data overrun flag signaled by the data overrun status bit.
2	W	0	**REL_RX_BUF** **Release Rx Buffer** Set this bit to 1 to release receive buffer
1	W	0	**ABT_REQ** **Abort Request** Set this bit to 1 to request to abort the current message transmission
0	W	0	**TRANS_REQ** **Transmission Request** Set this bit to 1 to request to transmit a message

1.5.3. CAN Status Register

Offset: 0x0008			Register Name: CAN_STA_REG Default Value: 0x0000_003c
Bit	**R/W**	**Default**	**Description**
31:24	/	/	/
23:22	R	0	**ERR_CODE** **Error Capture Error Code** 0 - Bit error 1 - Form error 2 - Stuff error 3 - Some other type of error
21	R	0	**ERR_DIR** **Error Capture Direction** 1 - the error occurred during reception 0 - the error occurred during transmission
20:16	R	0	**ERR_SEG_CODE** **Error Capture Segment Code** 00011 - Start of frame 00010 - ID.28 to ID.21 00110 - ID.20 to ID.18 00100 - SRTR bit 00101 - IDE bit 00111 - ID.17 to ID.13 01111 - ID.12 to ID.5 01110 - ID.4 to ID.0 01100 - RTR bit 01101 - Reserved bit 1 01001 - Reserved bit 0 01011 - Data Length Code 01010 - Data Field 01000 - CRC sequence 11000 - CRC delimiter 11001 - Acknowledge 11011 - Acknowledge delimiter 11010 - End of frame 10010 - Intermission 10001 - Active error flag 10110 - Passive error flag 10011 - Tolerate dominant bits 10111 - Error delimiter 11100 - Overload flag
15:13	-	-	/
12:8	R	0	**ARB_LOST**

			Arbitration Lost Capture [0~10] – Arbitration lost in bit[0~10](1st~11th bit of ID, ID.28~ID.18). 11 - Arbitration lost in bit[11](SRTR bit). 12 - Arbitration lost in bit[12](IDE bit). [13~30] – Arbitration lost in bit[13th~30th](12th~29th bit of ID, ID.17~ID.0). 31 - Arbitration lost in bit[31](RTR bit).
7	R	0	**BUS_STA** **Bus Status** 1 - The controller is in 'Bus Off' state and is not involved in bus activities 0 - The controller is involved in bus activities
6	R	0	**ERR_STA** **Error Status** 1 - At least one of the error counters has reached or exceeded the CPU warning limit defined by the Error Warning Limit Register (EWL). 0 - Both error counters are below the warning limit
5	R	1	**TX_STA** **Transmit Status** 1 – controller is in the process of transmitting a message 0 – nothing is currently being Transmitted
4	R	1	**RX_STA** **Receive Status** 1 – controller is in the process of receiving a message 0 – nothing is currently being received
3	R	1	**TX_OVER** **Transmission Complete** 1 – The last requested transmission has been successfully completed 0 - The last requested transmission has not been completed
2	R	1	**TX_RDY** **Tx Buffer Ready** 1 – Tx buffer ready. 0 – Tx buffer not ready.
1	R	0	**DATA_OR** **Data overrun** 1 – data buffer overrun 0 – data buffer not overrun
0	R	0	**RX_RDY** **Rx Buffer Ready** 1 – Rx buffer is not empty. 0 – Rx buffer is empty.

1.5.4. CAN Interrupt Register

Offset: 0x000c			Register Name: CAN_INT_REG Default Value: 0x0000_0000
Bit	**R/W**	**Default**	**Description**
31:8	/	/	/
7	R	0	**BERR** **Bus Error Interrupt** Set when the controller detects an bit error on the CAN bus *This is a wirte-1-to-clear bit.*
6	R	0	**ARB_LOST** **Arbitration Lost Interrupt** Set when the controller loses arbitration and becomes a receiver *This is a wirte-1-to-clear bit.*
5	R	0	**ERR_PASSIVE** **Error Passive Interrupt** Set when the controller re-enters error active state after being in error passive state or when at least one error counter exceeds the protocol-defined level of 127 *This is a wirte-1-to-clear bit.*
4	R	0	**WAKEUP** **Wake-Up Interrupt** Set when bus activity is detected while the CAN controller is sleeping *This is a read-to-clear bit.*
3	R	0	**DATA_OR** **Data Overrun Interrupt** Set on a '0-to-1' transition of the Data Overrun Status bit *This is a wirte-1-to-clear bit.*
2	R	0	**ERR** **Error Warning Interrupt** Set on every change (set or clear) of either the Bus Status or Error Status bits (SR.7,SR.6) *This is a wirte-1-to-clear bit.*
1	R	0	**TX_FLAG** **Transmit Interrupt Flag** Set whenever the Transmit Buffer Status (SR.2) changes from '0-to-1' (released) *This is a wirte-1-to-clear bit.*
0	R	0	**RX_FLAG** **Receive Interrupt Flag** Set whenever the Receive Buffer contains one or more messages. Cleared when the release Receive Buffer command (CMR. 2) is

			issued, provided there is no further data to read in the Receive Buffer. *This is a wirte-1-to-clear bit.*

1.5.5. CAN Interrupt Enable Register

Offset: 0x0010			**Register Name: CAN_INTE_REG** **Default Value: 0x0000_0000**
Bit	**R/W**	**Default**	**Description**
31:8	/	/	/
7	R/W	0	**BERR_EN** **Bus Error Interrupt Enable** 1 – bus error interrupt enable 0 – bus error interrupt disable
6	R/W	0	**ARB_LOST_EN** **Arbitration Lost Interrupt Enable** 1 – arbitration lost interrupt enable 0 – arbitration lost interrupt disable
5	R/W	0	**ERR_PASSIVE_EN** **Error Passive Interrupt Enable** 1 – error passive interrupt enable 0 – error passive interrupt disable
4	R/W	0	**WAKEUP_EN** **Wake-Up Interrupt Enable** 1 – wake up interrupt enable 0 – wake up interrupt disable
3	R/W	0	**OR_EN** **Data Overrun Interrupt Enable** 1 – data overrun interrupt enable 0 – data overrun interrupt disable
2	R/W	1	**ERR_WRN_EN** **Error Warning Interrupt Enable** 1 - Error Warning Interrupt Enable 0 - Error Warning Interrupt Disable
1	R/W	1	**TX_EN** **Transmit Interrupt Enable** 1 – transmit interrupt enable 0 – transmit interrupt disable
0	R/W	0	**RX_EN** **Receive Interrupt Enable** 1 – receive interrupt enable 0 - receive interrupt disable

1.5.6. CAN Bus Timing Register

Offset: 0x0014			Register Name: CAN_BUS_TIME Default Value: 0x0000_0000
Bit	**R/W**	**Default**	**Description**
31:24	/	/	/
23	R/W	0	**SAM** **Sample Point Control** 0 – Bus line is sampled once at the sample point 1 – Bus line is sampled three times at the sample point *This bit is only writable in reset mode.*
[22:20]	R/W	0	**PHSEG2** **Phase Segment 2** [0..7] – [1..8] Tq clock cycle(s) *These bits are only writable in reset mode.*
[19:16]	R/W	0	**PHSEG1** **Phase Segment 1** [0..15] – [1..16] Tq clock cycle(s) *These bits are only writable in reset mode.*
[15:14]	R/W	0	**SJW** **Synchronization Jump Width** The SJW defines the maximum number of Tq clock cycles a bit can be shortened or lengthened to achieve resynchronization to data transitions on the bus. 0 – 1 Tq clock cycle 1 – 2 Tq clock cycles 2 – 3 Tq clock cycles 3 – 4 Tq clock cycles *These bits are only writable in reset mode.*
[13:10]	/	/	/
[9:0]	R/W	0	**TQ_BRP** **Time Quanta Baud Rate Prescaler** These bits determine the time quanta (Tq) clock which is used to build up the individual bit timing. *These bits are only writable in reset mode.*

1.5.7. CAN TX Error Warning Limit Register

Offset: 0x0018			Register Name: CAN_EWL_REG Default Value: 0x0000_0060
Bit	**R/W**	**Default**	**Description**
31:8	/	/	/
7:0	R/W	0x60	ERR_WRN_LMT Error Warning Limit These bits define the number of errors after which an Error

			Warning Interrupt should be generated (if enabled). *These bits are only writable in reset mode.*

1.5.8. CAN Error Counter Register

Offset: 0x001c			**Register Name: CAN_REC_REG** **Default Value: 0x0000_0000**
Bit	**R/W**	**Default**	**Description**
31:24	/	/	/
23:16	R/W	0	**RX_ERR_CNT** **Receive Error Counter** These bits record the current value of receive counter *These bits are only writable in reset mode.*
15:8	/	/	/
7:0	R/W	0	**TX_ERR_CNT** **Transmit Error Counter** These bits record the current value of transmit counter *These bits are only writable in reset mode.*

1.5.9. CAN Receive Message Counter

Offset: 0x0020			**Register Name: CAN_RMSGC_REG** **Default Value: 0x0000_0000**
Bit	**R/W**	**Default**	**Description**
31:8	/	/	/
7:0	R	0	RX_MSG_CNT CAN receive message counter

1.5.10. CAN Receive Buffer Start Address Register

Offset: 0x0024			**Register Name: CAN_RSADDR_REG** **Default Value: 0x0000_0000**
Bit	**R/W**	**Default**	**Description**
31:6	/	/	/
5:0	R/W	0	RX_BUFF_SADDR CAN receive buffer start address pointer *These bits are only writable in reset mode.*

1.5.11. CAN TX/RX Message Buffer 0 Register (transfer mode)

Offset: 0x0040			**Register Name: CAN_TXBUF0** **Default Value: 0x0000_0000**
Bit	**R/W**	**Default**	**Description**
31:8	/	/	/
7	R/W	0	**EFF**

			Extend frame flag 1 – Extend frame 0 – Standard frame
6	R/W	0	**RTR** Remote Transmit 1 – remote frame 0 – normal frame
5:4	/	/	/
3:0	R/W	0	**Data Length** Date length of message requested to send

1.5.12. CAN TX/RX Message Buffer 1 Register (transfer mode)

Offset: 0x0044			**Register Name: CAN_TXBUF1** **Default Value: 0x0000_0000**
Bit	**R/W**	**Default**	**Description**
31:8	/	/	/
7:0	R/W	0	ID[28:21]

1.5.13. CAN TX/RX Message Buffer 2 Register (transfer mode)

Offset: 0x0048			**Register Name: CAN_TXBUF2** **Default Value: 0x0000_0000**
Bit	**R/W**	**Default**	**Description**
31:8	/	/	/
7:5	R/W	0	SID Standard ID SEF-ID[20:18] / EFF-ID[20:18]
4:0	R/W	0	EID Extended ID EFF- ID[17:13]

1.5.14. CAN TX/RX Message Buffer 3 Register (transfer mode)

Offset: 0x004c			**Register Name: CAN_TXBUF3** **Default Value: 0x0000_0000**
Bit	**R/W**	**Default**	**Description**
31:8	/	/	/
7:0	R/W	0	SDATA1_EID SFF - TX data byte 1 / EFF- ID[12:5]

1.5.15. CAN TX/RX Message Buffer 4 Register (transfer mode)

Offset: 0x0050	**Register Name: CAN_TXBUF4** **Default Value: 0x0000_0000**

Bit	R/W	Default	Description
31:8	/	/	/
7:3	R/W	0	SDATA2_EID SFF-TX data byte2[7:3] / EFF-ID[4:0]
2:0	R/W	0	SDATA2 SFF-TX data byte2[2:0]

1.5.16. CAN TX/RX Message Buffer 5 Register (transfer mode)

Offset: 0x0054			Register Name: CAN_TXBUF5 Default Value: 0x0000_0000
Bit	**R/W**	**Default**	**Description**
31:8	/	/	/
7:0	R/W	0	SDATA3_EDATA1 SFF-TX data byte 3 / EFF- TX data byte 1

1.5.17. CAN TX/RX Message Buffer 6 Register (transfer mode)

Offset: 0x0058			Register Name: CAN_TXBUF6 Default Value: 0x0000_0000
Bit	**R/W**	**Default**	**Description**
31:8	/	/	/
7:0	R/W	0	SDATA4_EDATA2 SFF-TX data byte 4 / EFF- TX data byte 2

1.5.18. CAN TX/RX Message Buffer 7 Register (transfer mode)

Offset: 0x005c			Register Name: CAN_TXBUF7 Default Value: 0x0000_0000
Bit	**R/W**	**Default**	**Description**
31:8	/	/	/
7:0	R/W	0	SDATA5_EDATA3 SFF-TX data byte 5 / EFF- TX data byte 3

1.5.19. CAN TX/RX Message Buffer 8 Register (transfer mode)

Offset: 0x0060			Register Name: CAN_TXBUF8 Default Value: 0x0000_0000
Bit	**R/W**	**Default**	**Description**
31:8	/	/	/
7:0	R/W	0	SDATA6_EDATA4 SFF-TX data byte 6 / EFF- TX data byte 4

1.5.20. CAN TX/RX Message Buffer 9 Register (transfer mode)

Offset: 0x0064	Register Name: CAN_TXBUF9

			Default Value: 0x0000_0000
Bit	**R/W**	**Default**	**Description**
31:8	/	/	/
7:0	R/W	0	SDATA7_EDATA5 SFF-TX data byte 7 / EFF- TX data byte 5

1.5.21. CAN TX/RX Message Buffer 10 Register (transfer mode)

Offset: 0x0068			Register Name: CAN_TXBUF10 Default Value: 0x0000_0000
Bit	**R/W**	**Default**	**Description**
31:8	/	/	/
7:0	R/W	0	SDATA8_EDATA6 SFF-TX data byte 8 / EFF- TX data byte 6

1.5.22. CAN TX/RX Message Buffer 11 Register (transfer mode)

Offset: 0x006c			Register Name: CAN_TXBUF11 Default Value: 0x0000_0000
Bit	**R/W**	**Default**	**Description**
31:8	/	/	/
7:0	R/W	0	EDATA7 EFF- TX data byte 7

1.5.23. CAN TX/RX Message Buffer 12 Register (transfer mode)

Offset: 0x0070			Register Name: CAN_TXBUF12 Default Value: 0x0000_0000
Bit	**R/W**	**Default**	**Description**
31:8	/	/	/
7:0	R/W	0	EDATA8 EFF- TX data byte 8

1.5.24. CAN Acceptance Code Register (reset mode)

Offset: 0x0040			Register Name: CAN_AC0_REG Default Value: 0x0000_0000
Bit	**R/W**	**Default**	**Description**
31:0	R/W	0	CAN_ACP_CODE CAN acceptance code byte[3:0]

1.5.25. CAN Acceptance Mask Register (reset mode)

Offset: 0x0044			Register Name: CAN_AM0_REG Default Value: 0x0000_0000
Bit	**R/W**	**Default**	**Description**

31:8	/	/	reserved
7:0	R/W	0	CAN_ACP_MSK CAN acceptance mask byte[3:0]

1.5.26. CAN TX Message Buffer for Read Register

Offset: 0x0180~0x01b0			**Register Name: CAN_WBUF_RD** **Default Value: 0x0000_0000**
Bit	**R/W**	**Default**	**Description**
31:8	/	/	/
7:0	R	0	TBUF_RD_BACK TX message buffer for read back. *Each register is 32-bit width register, but only the lower 8 bits are valid to access. All higher 24 bits will return 0 when be read.*