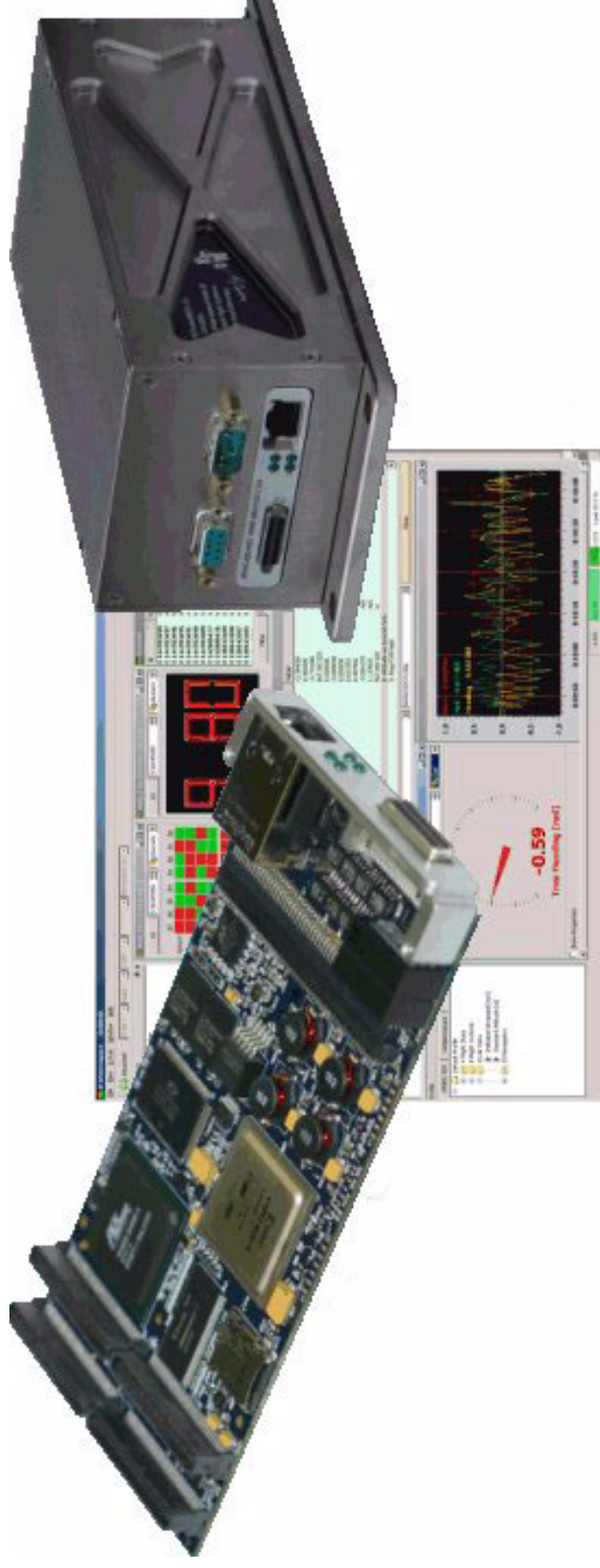


# PMC825 Ethernet/UDP/IP Interface



**MICHAEL  
STOCK**  
FLIGHT SYSTEMS

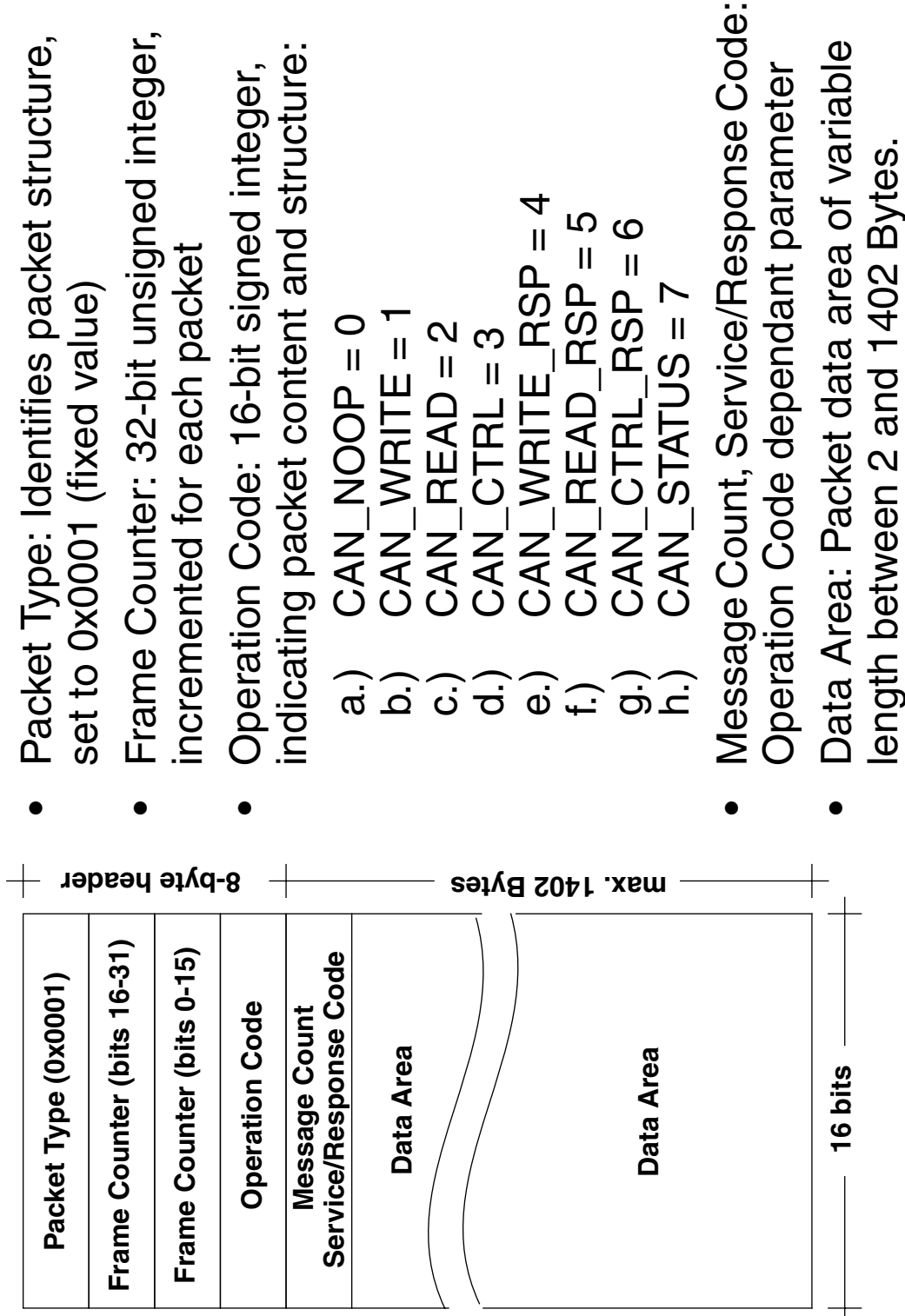


---

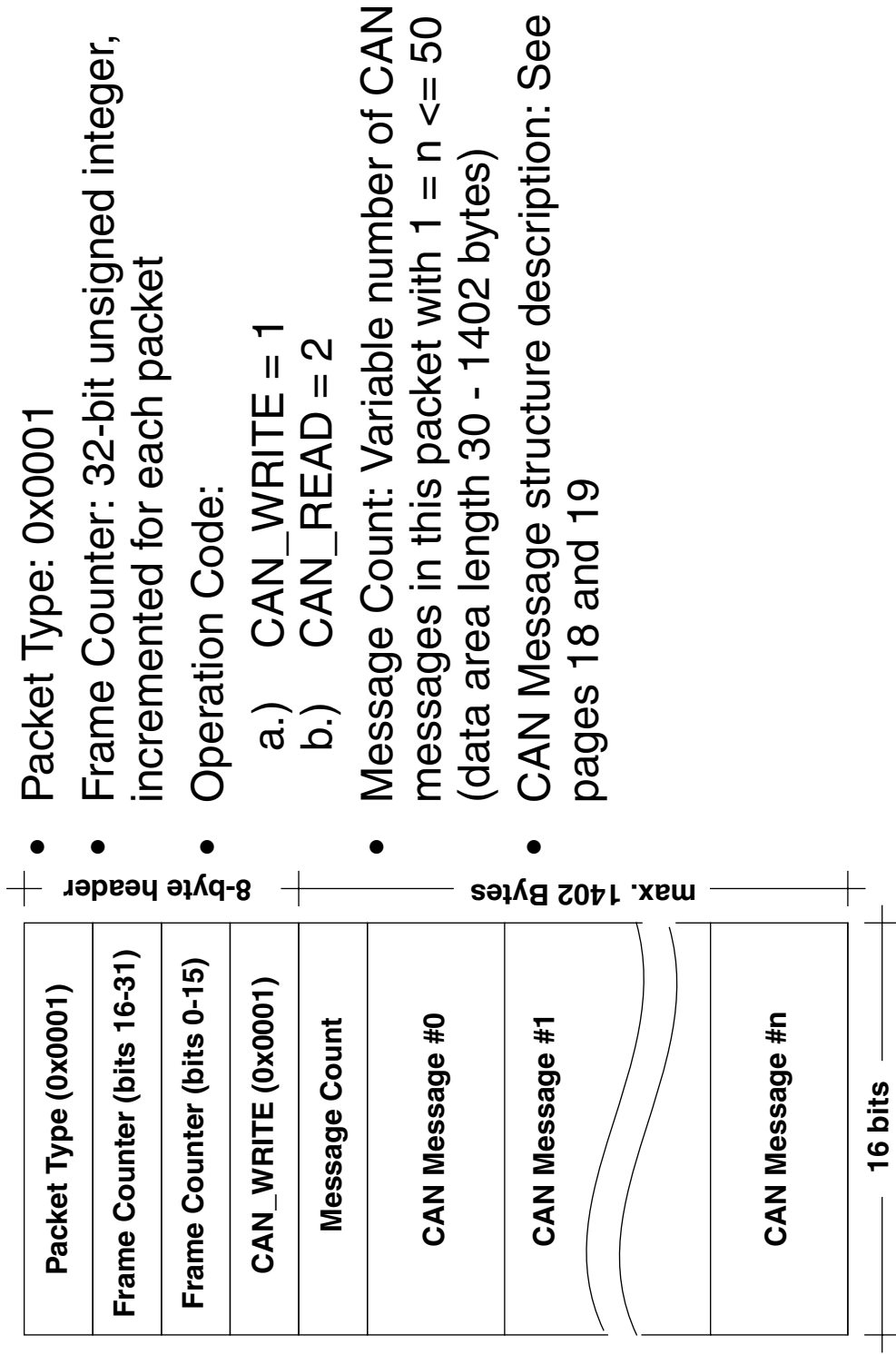
## Ethernet Interface Features

- Full-duplex network communication module
- Integrated MAC and PHY
- Configurable 10/100/1000 BaseT Ethernet
- RJ-45 connector with link and activity LEDs
- UDP/IP support with configurable MAC addresses, IP addresses and port numbers
- DHCP client functionality
- Each CAN channel can be assigned individual IP addresses and port numbers for host communication
- Human readable and editable configuration data file stored on microSD/microSDHC card

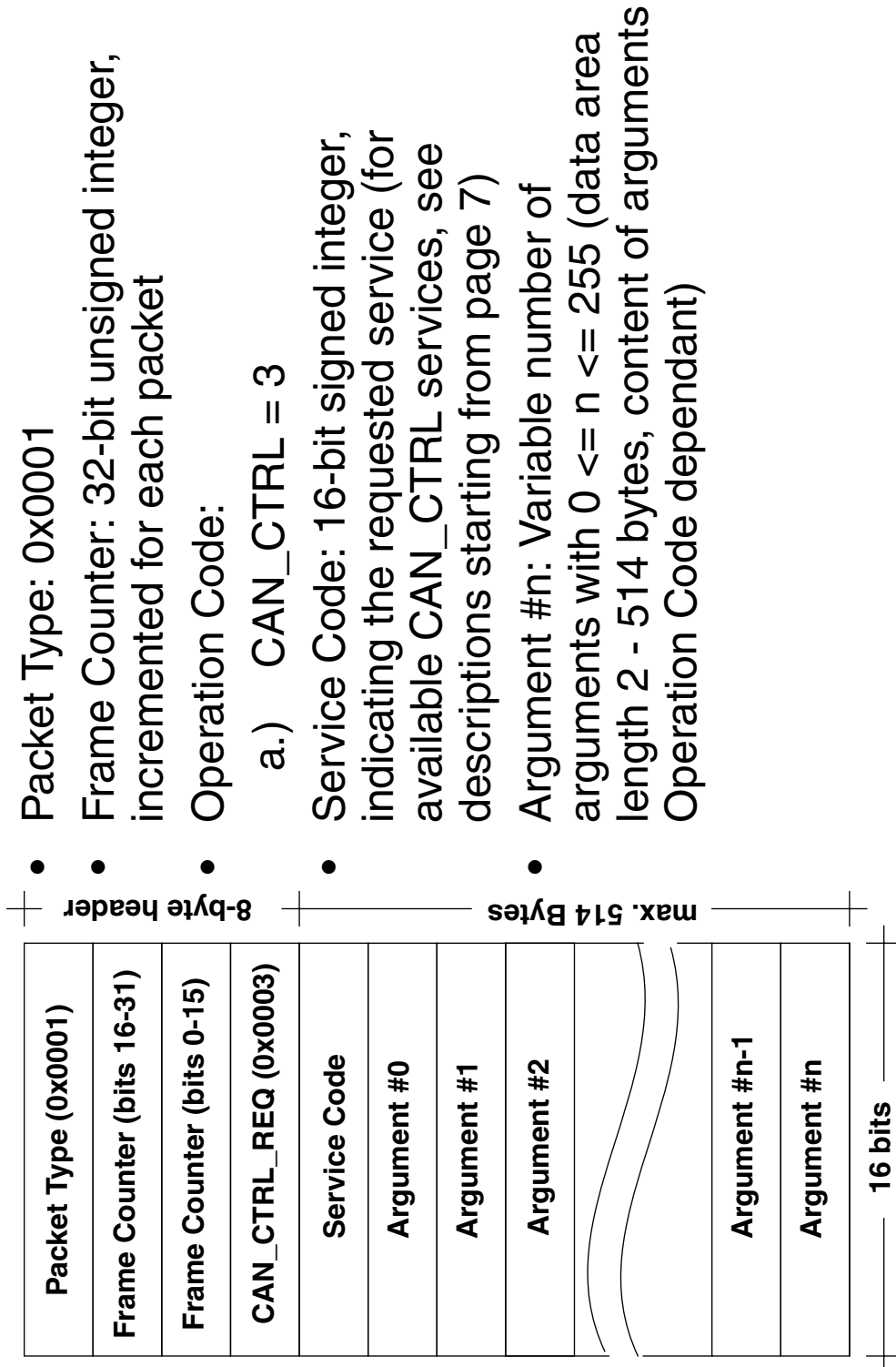
## General UDP Packet Structure



## CAN\_WRITE/CAN\_READ Packet Structure

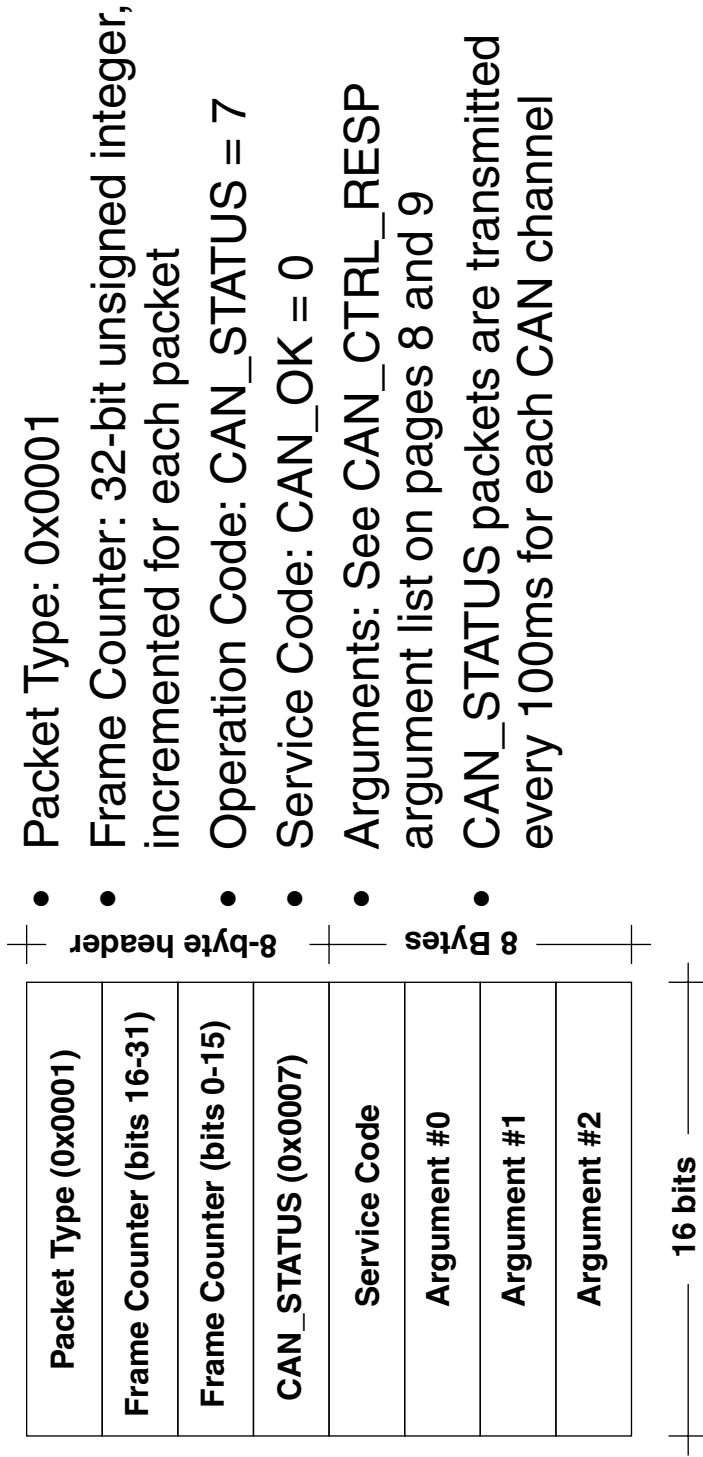


# CAN\_CTRL Packet Structure



- Packet Type: 0x0001
- Frame Counter: 32-bit unsigned integer, incremented for each packet
- Operation Code:
  - a.) CAN\_CTRL = 3
- Service Code: 16-bit signed integer, indicating the requested service (for available CAN\_CTRL services, see descriptions starting from page 7)
- Argument #n: Variable number of arguments with  $0 \leq n \leq 255$  (data area length 2 - 514 bytes, content of arguments Operation Code dependant)

## CAN\_STATUS Packet Structure



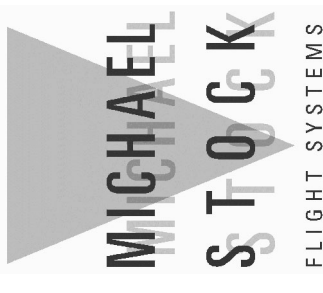
- Packet Type: 0x0001
  - Frame Counter: 32-bit unsigned integer, incremented for each packet
  - Operation Code: CAN\_STATUS = 7
  - Service Code: CAN\_OK = 0
  - Arguments: See CAN\_CTRL\_RESP argument list on pages 8 and 9
  - CAN\_STATUS packets are transmitted every 100ms for each CAN channel
- A CAN\_STATUS packet is also transmitted for CAN channel 0 as a broadcast packet for each PMC825 which obtained its IP address through DHCP (once per second)

# CAN\_CTRL Service (1)

## INIT\_CAN\_CHIP

Service	Service Code	Description	CAN_CTRL_REQ Argument List	CAN_CTRL_RSP Argument List	Arg #
INIT_CAN_CHIP	10	<p>Initializes the C_CAN IP Core Bit Timing Register to set the CAN baud rate and synchronization jump width.</p> <p>Allows also to configure the C_CAN to totally release the bus (Bus-off), for silent mode (listen only, no ACK bit generation, no CAN transmissions) and for loopback mode (internal loopback for transmitted CAN messages) as described on pages 22-24.</p> <p>If Bus-Off Mode is selected, all other settings issued through INIT_CAN_CHIP have no effect.</p> <p>CPM mode settings can be OR'd together (see page 25)</p>	<p>Bit Timing Register Setting based on 32MHz CAN_CLK (see page 19)</p> <p>Sample settings for various CAN baud rates based on 32MHz CAN_CLK are:</p> <p>0x1403 = 1 Mbit/s            0x1407 = 500 kbit/s            0x140F = 250 kbit/s            0x141F = 125 kbit/s            0x142F = 83.3 kbit/s</p> <p>For Silent/Loopback Modes, see pages 22-24</p>	-	0
			0x0001 = Silent Mode	-	1
			0x0001 = Loopback Mode	-	2
			0x0001 = Bus-Off Mode	-	3
			0x0001 = CPM On 0x0002 = CPM Pause 0x0004 = CPM Cyclic	-	4

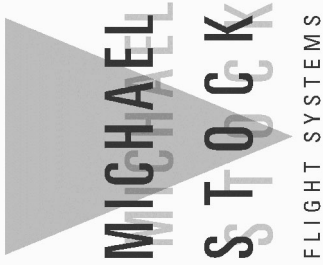
# CAN\_CTRL Service (2) GET\_CAN\_STATUS



Service	Service Code	Description	CAN_CTRL_REQ Argument List	CAN_CTRL_RSP Argument List	Arg #
GET_CAN_STATUS	11	Returns the content of the C_CAN IP Core Status register, the Error Counters and the Bit Timing register, the CPM mode and the transmitted/received bit count for the last 100ms	-	Status Register content (see page 20)  For register content description, refer to Bosch C_CAN IP Core User's Manual Revision 1.2, section 3.2.2	0
			-	Error Counter content (see page 21)  For register content description, refer to Bosch C_CAN IP Core User's Manual Revision 1.2, section 3.2.3	1
			-	Bit Timing Register Setting based on 32MHz CAN_CLK (see page 19)  For register content description, refer to Bosch C_CAN IP Core User's Manual Revision 1.2, section 3.2.4	2

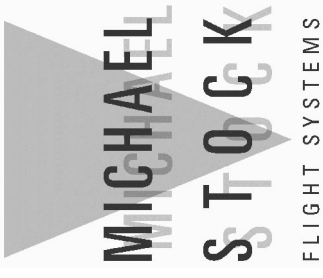


# CAN\_CTRL Service (2) GET\_CAN\_STATUS (contd.)



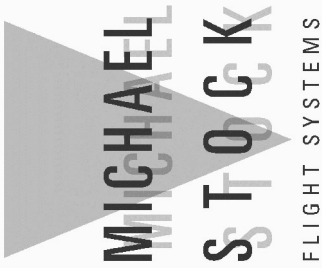
Service	Service Code	Description	CAN_CTRL_REQ Argument List	CAN_CTRL_RSP Argument List	Arg #
GET_CAN_STATUS	11	Returns the content of the C_CAN IP Core Status register, the Error Counters and the Bit Timing register, the CPM mode and the transmitted/received bit count for the last 100ms	-	CPM Mode/Status (see page 26)	3
			-	Number of transmitted CAN bits during the last 100ms	4-5
			-	Number of received CAN bits during the last 100ms	6-7
			-	Number of transmitted CAN messages (last 100ms)	8-9
			-	Number of received CAN messages (last 100ms)	10-11
			-	Current Board Temperature in Degrees Celsius	12
			-	Current FPGA Temperature in Degrees Celsius	13
			-	Current PMC825 IP address	14-15
			-	Module Name (see page 15)	16-31

# CAN\_CTRL Service (2) GET\_CAN\_STATUS (contd.)



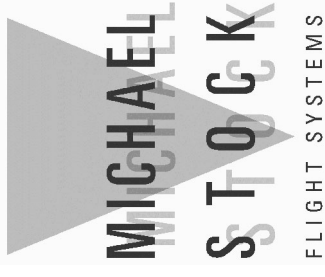
Service	Service Code	Description	CAN_CTRL_REQ Argument List	CAN_CTRL_RSP Argument List	Arg #
GET_CAN_STATUS	11	Returns the content of the C_CAN IP Core Status register, the Error Counters and the Bit Timing register, the CPM mode and the transmitted/received bit count for the last 100ms	-	CPM Buffer Fill Status (# of CAN messages in the buffer waiting for transmission)	32

# CAN\_CTRL Service (3) RESET\_TIME\_STAMP



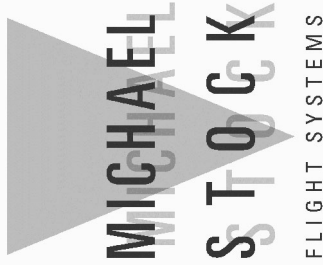
Service	Service Code	Description	CAN_CTRL_REQ Argument List	CAN_CTRL_RSP Argument List	Arg #
RESET_TIME_STAMP	12	<p>Sets the 64-bit CAN time stamp to "0"</p> <p>CAN timers may be OR'd together</p>	<p>0x0001 = CAN timer #0            0x0002 = CAN timer #1            0x0004 = CAN timer #2            0x0008 = CAN timer #3            0x0010 = CAN timer #4            0x0020 = CAN timer #5            0x0040 = CAN timer #6            0x0080 = CAN timer #7</p>	-	0

# CAN\_CTRL Service (4) CONTROL\_CPM



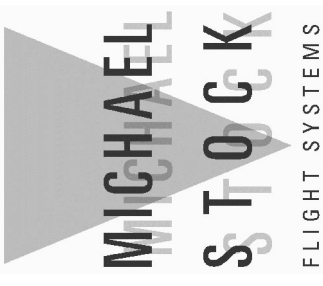
Service	Service Code	Description	CAN_CTRL_REQ Argument List	CAN_CTRL_RSP Argument List	Arg #
CONTROL_CPM	13	Controls the CPM CPM mode settings may be OR'd together (see page 26)	0x0001 = CPM On 0x0002 = CPM Pause 0x0004 = CPM Cyclic	-	0

# CAN\_CTRL Service (5) GET\_TEMPERATURES



Service	Service Code	Description	CAN_CTRL_REQ Argument List	CAN_CTRL_RSP Argument List	Arg #
GET_TEMPERATURES	14	Gets the current board/FPGA temperatures	-	Current Board Temperature in Degrees Celsius	0
			-	Current FPGA Temperature in Degrees Celsius	1

# CAN\_CTRL Service (6) CONFIG\_IP\_INTERFACE



Service	Service Code	Description	CAN_CTRL_REQ Argument List	CAN_CTRL_RSP Argument List	Arg #
CONFIG_IP_INTERFACE	100	Configures the PMC825 IP interface based on the PMC825 ASCII configuration file format	PMC825 configuration file content (max. 512 ASCII characters)  Supported tags: MAX IPx LPx RPx LSx	-	0-255

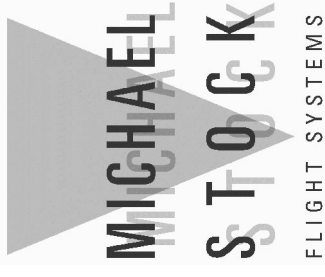
**Note:** IP Interfaces can be set to broadcast by specifying the broadcast MAC address and the broadcast IP address. For a Class C network address, this is:

MAX=FF:FF:FF:FF:FF:FF  
IPx=xxx.xxx.xxx.255

(where “xxx” reflects the corresponding network settings)

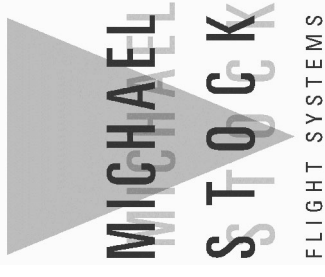
Automatic IP address assignment via DHCP is supported. In this case, the corresponding IPx-Tag has to be set to:  
  
IPx=DHCP

# CAN\_CTRL Service (7) GET\_MODULE\_INFO



Service	Service Code	Description	CAN_CTRL_REQ Argument List	CAN_CTRL_RSP Argument List	Arg #
GET_MODULE_INFO	101	Returns the PMC825 module information	-	Firmware Loop Count	0-1
			-	Module Type	2-3
			-	FPGA Revision	4-5
			-	Hardware Revision	6-7
			-	Firmware Revision	8-9
			-	Build Date String	10-11
			-	Built-In Test Result	12-13
			-	Module Name (32 Bytes)	14-29

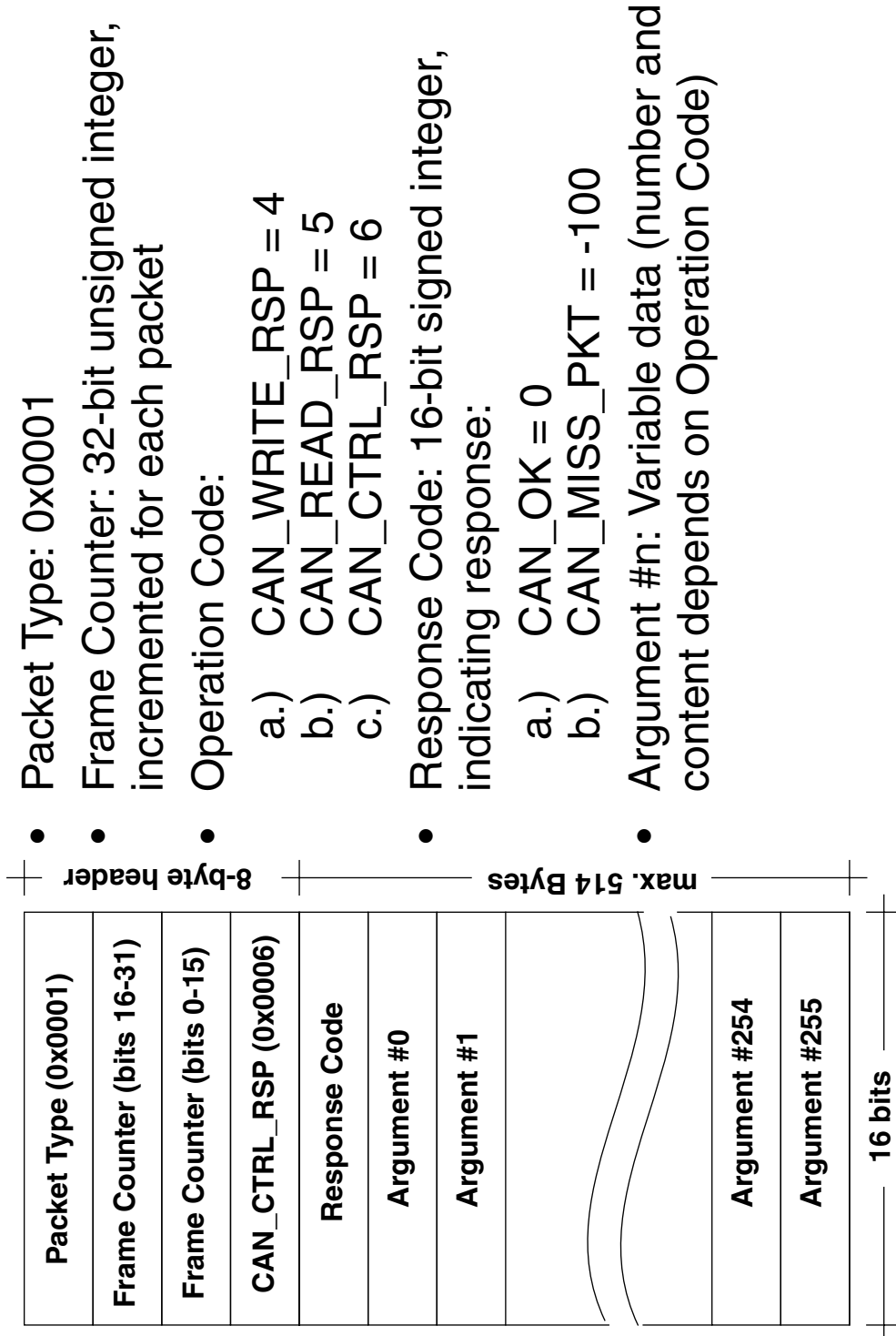
# CAN\_CTRL Service (8) SET\_MODULE\_NAME



Service	Service Code	Description	CAN_CTRL_REQ Argument List	CAN_CTRL_RSP Argument List	Arg #
SET_MODULE_NAME	102	Sets the PMC825 module name string	Module Name (32 Bytes)	-	0-15

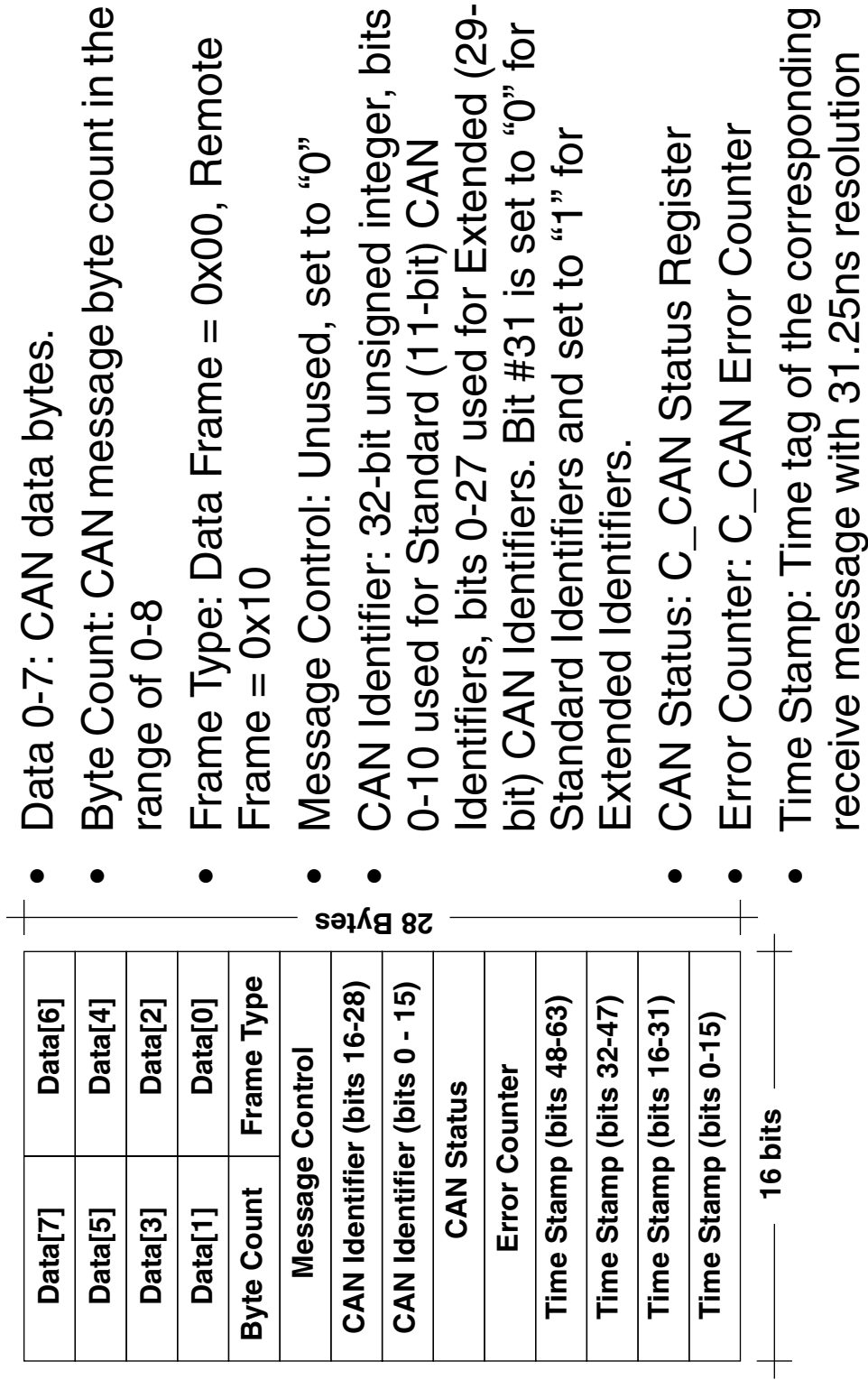


# CAN\_XXXX\_RSP Packet Structure



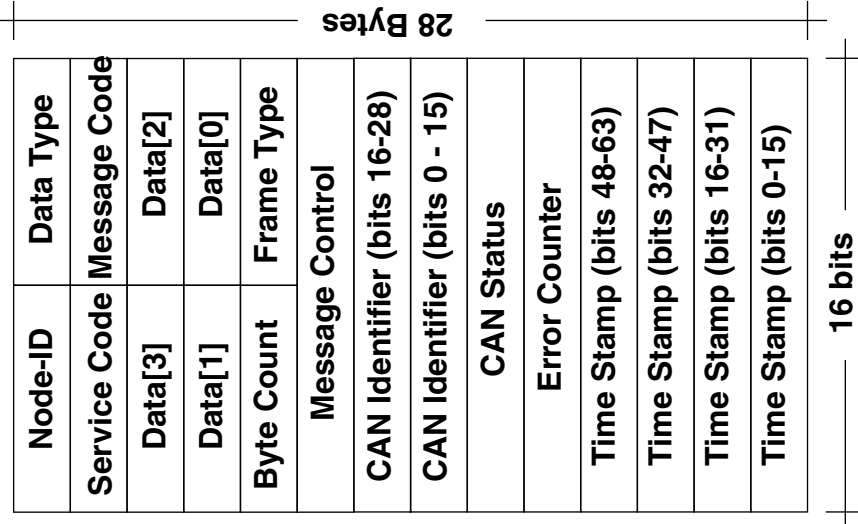
- Packet Type: 0x0001
- Frame Counter: 32-bit unsigned integer, incremented for each packet
- Operation Code:
  - a.) CAN\_WRITE\_RSP = 4
  - b.) CAN\_READ\_RSP = 5
  - c.) CAN\_CTRL\_RSP = 6
- Response Code: 16-bit signed integer, indicating response:
  - a.) CAN\_OK = 0
  - b.) CAN\_MISS\_PKT = -100
- Argument #n: Variable data (number and content depends on Operation Code)

# CAN Message Structure

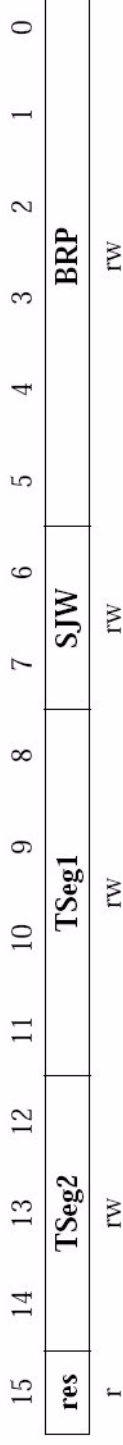


# CANaerospace Message Structure

- Identical to CAN Message Structure but with specific meaning of Data Bytes 4-7 according to CANaerospace specification



## C\_CAN Bit Timing Register



- TSEG1: Time segment before sample point (1-15)
- TSEG2: Time segment after sample point (0-7)
- SJW: Resynchronization Jump Width (0-3)
- BRP: Baud Rate Prescaler (0-63). Note that the CAN time quantum generation is based on 32 MHz C\_CAN clock for the PMC825
- res: Unused, settings have no effect

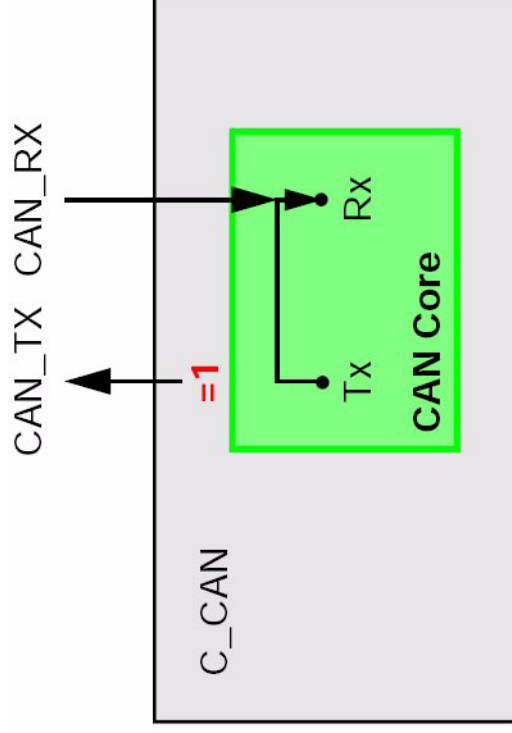


## C\_CAN Error Counter

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RP	REC6-0										TEC7-0				
I	I										I				

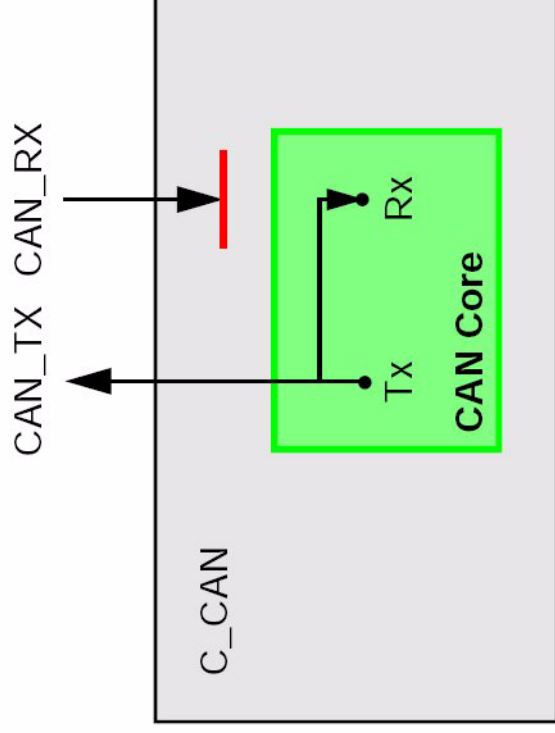
- RP: Set if the C\_CAN core is in the Receive Error Passive mode
- REC6-0: Current Value of the Receive Error Counter
- TEC7-0: Current Value of the Transmit Error Counter

## C\_CAN Silent Mode



- In Silent Mode, the C\_CAN is able to receive valid data frames and valid remote frames, but it sends only recessive bits on the CAN bus and it cannot start a transmission.
- If the CAN Core is required to send a dominant bit (ACK bit, overload flag, active error flag), the bit is rerouted internally so that the CAN Core monitors this dominant bit, although the CAN bus may remain in recessive state.
- The Silent Mode can be used to analyse the traffic on a CAN bus without affecting it by the transmission of dominant bits (Acknowledge Bits, Error Frames).

## C\_CAN Loopback Mode

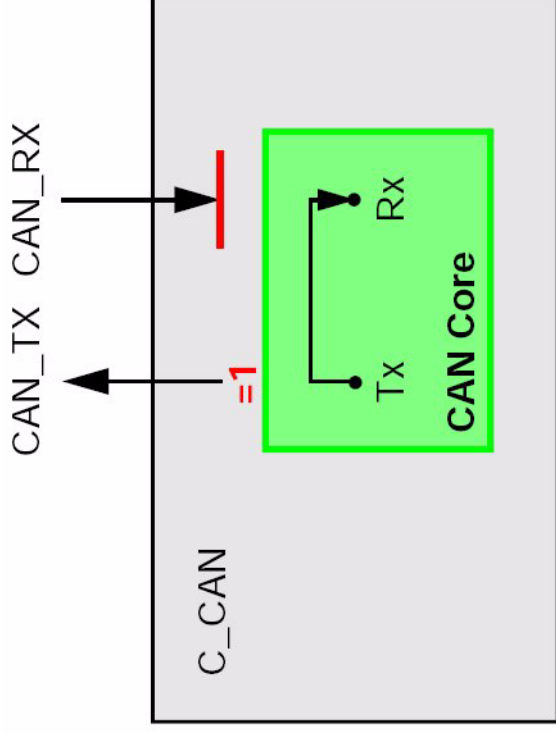


- In Loop Back Mode, the CAN Core treats its own transmitted messages as received messages and stores them (if they pass acceptance filtering) into a Receive Buffer.
- This mode is provided for self-test functions. To be independent from external stimulation, the CAN Core ignores acknowledge errors (recessive bit sampled in the acknowledge slot of a data/remote frame) in Loop Back Mode.
- In this mode the CAN Core performs an internal feedback from its Tx output to its Rx input. The actual value of the CAN\_RX input pin is disregarded by the CAN Core.
- The transmitted messages can be monitored at the CAN\_TX pin.

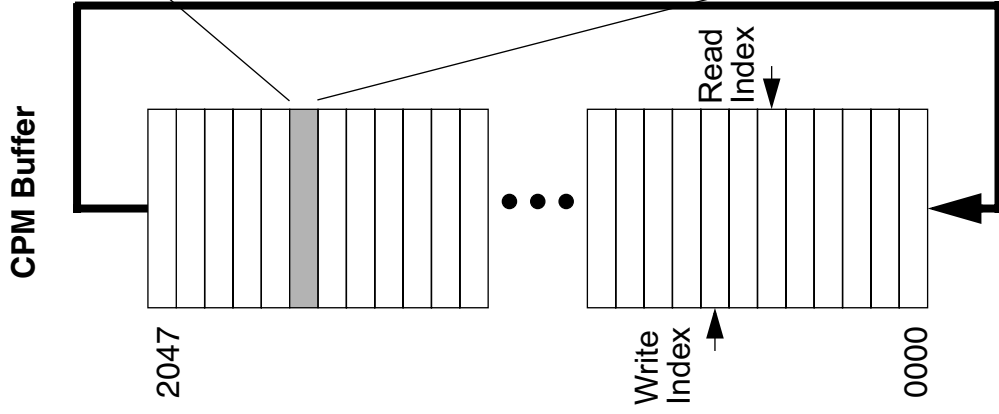


## C\_CAN Silent + Loopback Mode

- The combination of Loop Back Mode and Silent Mode can be used for a “Hot Selftest” .
- In this mode, the C\_CAN can be tested without affecting a running CAN system connected to the pins CAN\_TX and CAN\_RX.
- The CAN\_RX pin is disconnected from the CAN Core and the CAN\_TX pin is held recessive.



# CAN Playback Module (CPM)



Data[7]	Data[6]
Data[5]	Data[4]
Data[3]	Data[2]
Data[1]	Data[0]
Byte Count	Frame Type
Message Control	
CAN Identifier (bits 16-28)	
CAN Identifier (bits 0 - 15)	
CAN Status	
Error Counter	
Time Stamp (bits 48-63)	
Time Stamp (bits 32-47)	
Time Stamp (bits 16-31)	
Time Stamp (bits 0-15)	

16 bits

## CPM Playback Operation Mode

- Host writes x messages to buffer, then sets Write Index to top of queue
- Host commands CPM Enable
- CPM transmits messages starting at 0 until reaching Write Index, then stops
- If the CPM sees that Write Index is moved, it continues until reaching it
- If the CPM reaches end of buffer, it rolls over to zero and continues
- Host continuously writes messages to buffer and sets Write Index to top of queue
- Host may command CPM Disable at any time

## CPM Cyclic Operation Mode

- Host writes x messages to buffer, then sets Write Index to top of queue
- Host commands CPM Cyclic and CPM Enable
- CPM transmits messages starting at 0 until reaching Write Index, then rolls over to zero
- After rolling over, the CPM uses its previous start timer value and reruns
- Host may command CPM Disable at any time

**CPM Pause may be commanded at any time**

# CAN Playback Module (CPM) Mode/Status Register

