YAMAR *Datasheet*

DCAN500
CAN Over Powerline Transceiver

1. Overview

The DCAN500 device for CAN-BUS communication over noisy power lines (DC-CAN), supports CAN A/B protocols at CAN host bitrate up 1Mbit/s. Multiple CAN networks may share a common DC or AC powerline where each network uses a different carrier frequency. The device avoids complex cabling, saves weight, and simplifies installation. Sleep mode allows low power consumption when the device is not used. A QFN32 5x5 mm package provides a small PCB footprint ✓(RoHS)

The DCAN500 is beneficial for many applications using the CAN protocol but needs to eliminate the CAN bus wires in Aerospace, Automotive, Industrial, and more.

Applications

- Multiple vehicle CAN networks sharing the same powerline
- Aerospace CAN networks
- Sensors/actuators buses
- Robotics control network
- Truck-Trailer redundant CAN communication

Features

- Noise robust CAN A/B transceiver over DC powerline.
- CAN Host bitrates - 83.3kbit/s, 125kbit/s, 250kbit/s, 500kbit/s, and 1Mbit/s
- Multiple networks may operate over a single powerline.
- 251 selectable carrier frequencies (5MHz to 30MHz).
- Built-in CAN power line arbitration mechanism.
- Channel interference detection.
- Communicates over a wide range of DC voltages.
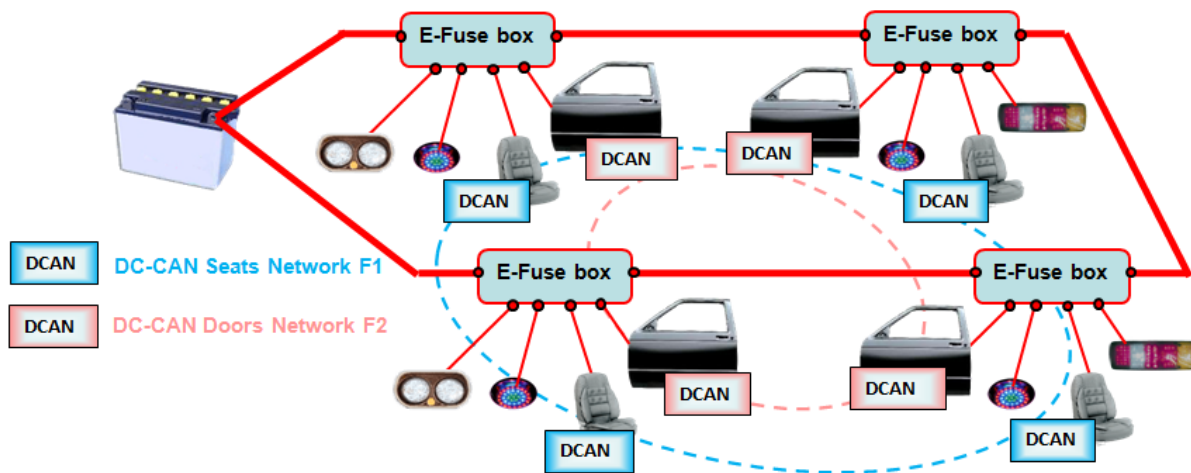- Sleep modes for low power consumption.

Figure 1- DCAN500 CAN networks sharing single powerline

TABLE OF CONTENT

2. Description

2.1 The DCAN500 network

The DCAN500 device transmits and receives CAN message from/to CAN controller (ECU) over DC and AC power lines at CAN host bitrates up to 1Mbit/s. The device operates as part of a power line (DC-BUS) communication network consisting of multiple DCAN500 devices. The CAN messages are error protected and phase-modulated by a sine wave at a user predefined carrier frequency and transmitted over the powerline as CAN frames.

All network topologies (e.g. Star, ring, line, tree, etc.) are applicable, as long as the received signal level at the RXI pin is above the minimal RXI_{lev} level (see 6).

Multiple DCAN500 networks can operate over a single powerline, whereas each network communicates over a different carrier frequency (channel).

2.2 Channel parameters

Carrier frequency: 251 selectable frequencies between 5MHz - 30MHz with 100 kHz spacing.
Powerline voltage: Any, with proper powerline coupling interfacing (see 2.5.8)
Cable length: Depends on the powerline loads AC signal-attenuation (100m is practicable)
Cable type: Any cable.

2.3 DCAN500 Architecture

Figure 2 depicts the DCAN500 blocks.
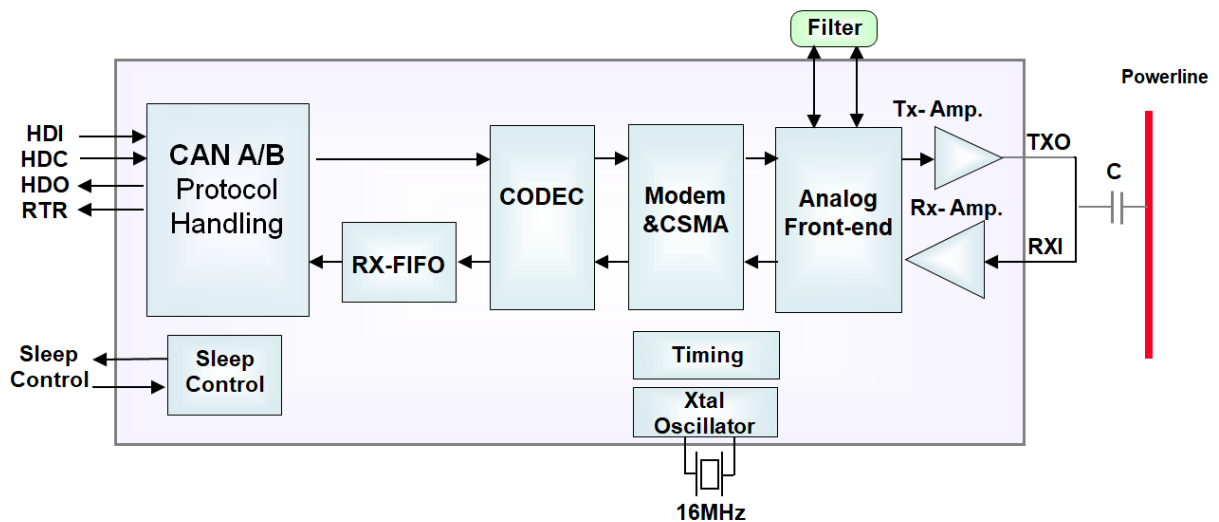
Figure 2 - DCAN500 block diagram

The DCAN500 main building blocks:

- **Protocol handling -** Interprets ECU's CAN A/B protocol.
- **Rx FIFO -** Buffers CAN frames between DCAN500 device and its ECU.
- **CODEC -** Encodes/decodes the CAN message.
- **Modem -** Phase modulates and demodulates the data to and from powerline CAN frames.
- **CSMA/CA –** CAN frame Carrier sense and an arbitration mechanism.
- **Sleep -** Ensures low power consumption during Sleep mode.

2.4   Configuration and Function

2.4.1   Pinout diagram
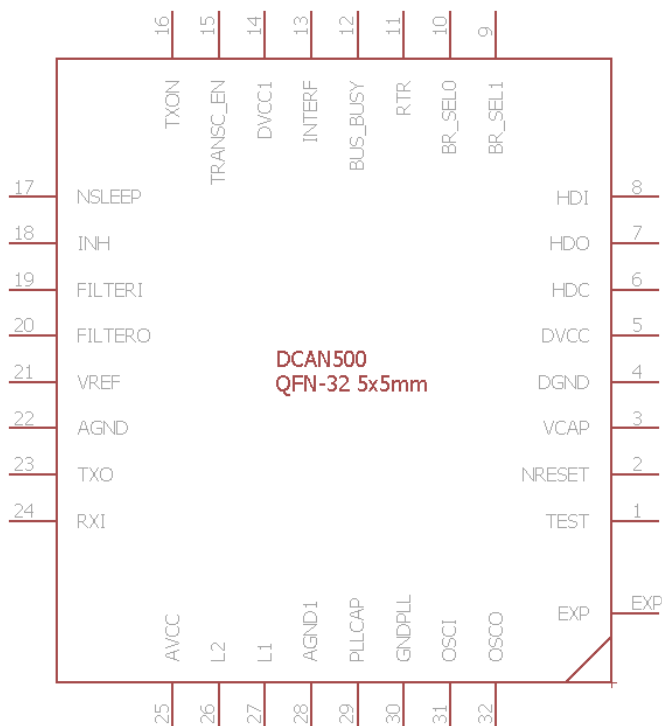
Figure 3 - DCAN500 pinout diagram in QFN32 5x5 mm package

2.4.2   Signals and Pinout description

Table 1- Pinout description

Name	Pin #	Pin type	Internal PU/PD	Description
HDO	7	Output 12mA		Outputs the received CAN data from the powerline or from internal registers to the ECU.
HDI	8	Digital input	PU	CAN message data input from the ECU to the powerline or the internal registers.
HDC	6	Digital input	PU	ECU data/command input enables read and writes from/to DCAN500 internal control registers (see section 5.12)
TEST	1	Digital Input	PD	Test pin should be connected to GND.
NRESET	2	Digital Input	PU	Reset, active low.
NSLEEP	17	Digital Input		Sleep mode control input (see section 4.3). Should be pull-up to 3.3V when not in use.
BR_SEL1	9	Digital Input		ECU bit rate selection pins (sampled once after each power-up/reset event). Setting of 1Mbit/s requires access to DCAN500 internal register **(see 10)** {{BITRATE_TABLE}}
BR_SEL0	10	Digital Input		
RTR	11	Digital Output 8mA		Ready to Receive (Optional feature) - When high, ECU can transfer CAN messages through the HDI pin. When low, ECU should pause its data bytes transfer, allowing extraction of RX-FIFO stored CAN frame to HDO pin.

Bitrate table (within BR_SEL1/BR_SEL0 Description):

BR_SEL[1:0]	CAN Bitrate [kbit/s]
'00'	83.33
'01'	125
'10'	250
'11'	500

Name	Pin #	Pin type	Internal PU/PD	Description
TRANSC_EN	15	Digital Input	PU	Enables usage of external CAN transceiver. Normally when HDI and HDO pins are connected directly to an ECU, connect this pin to GND. When HDI and HDO pins are connected to a CAN transceiver (i.e. interfacing a CAN-BUS), this pin can be left floated (see 3.1).
INH	18	Digital Output 8mA		When high, the DCAN500 is in Normal mode When low, the DCAN500 is in Sleep mode
INTERF	13	Digital Output 12mA		High while an interference signal is being detected in the operating carrier frequency.
BUS_BUSY	12	Digital Output 8mA		High when DCAN500 is transmitting over the powerline or during a reception from the powerline. This function can be used to monitor the status of the powerline channel and act accordingly (wait for the completion of reception or transmission).
TXON	16	Output 12mA		TX_ON output - High during transmission onto the powerline.
TXO	23	Analog Output Max 66 mA		**Powerline Transmit signal out** See table below
RXI	24	Analog Input		Powerline receive Input
VREF	21	Analog Output		VCC/2 analog output reference to a filtering capacitor. Place 1uF between VREF to AGND. The VREF is used as a virtual ground for the external analog circuitry.
FILTERI	19	Analog, Bi-directional		External filter I/O
FILTERO	20	Analog, Bi-directional		External filter I/O
OSCO	32	Analog output		16MHz Crystal Output
OSCI	31	Analog Input		16MHz Crystal Input
L1	27	Analog Input		External inductor L1 (maximal pin capacitance 1pF), see 2.5.4.
L2	26	Analog Input		External inductors L2 (optional), see2.5.4.
AVCC	25	Power		Analog 3.3V supply
AGND	22,28	Power		Analog ground
VCAP	3	Power		1.8V core supply output to a filtering capacitor. Place 4.7uF between VCAP and DGND.
DGND	4	Power		Digital Ground
DVCC	5,14	Power		Digital 3.3V supply
GNDPLL	30	Power		Analog Ground
PLLCAP	29	Power		PLL 1.8V output to a filtering capacitor. Place 1uF between PLLCAP and GNDPLL.
EXP	33	Power		Exposed pad, should be connected to DGND.

Powerline Transmit signal out (TXO):

TXON State	REG_1[3]	TX level [V-p-p]	Impedance [Ω]
High	'0'	1	18
	'1' (Default)	2	
Low		High Z	5.3k [2]

[1]Series output impedance
[2]Input impedance referenced to VREF

PD –Internal Pull-down resistor 50K Ω +/-%30
PU –Internal Pull-up resistor 50K Ω +/-%30

2.5   Implementation

2.5.1   Reference schematic

Figure 4 depicts a reference DCAN500 schematic.
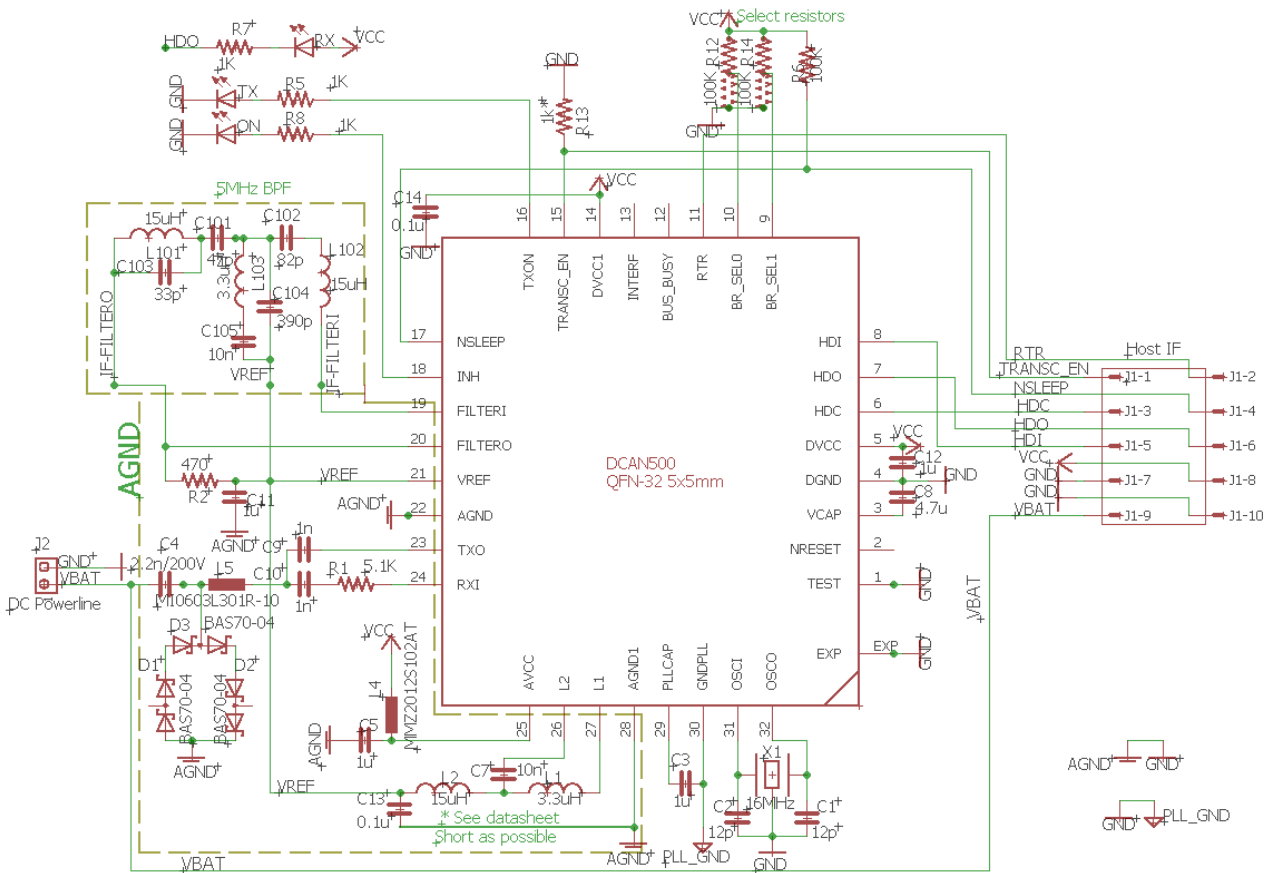
Figure 4- DCAN500 reference schematic

2.5.2   External filter (BPF)

The DCAN500 operates using an external 5MHz bandpass filter. The minimum allowable bandwidth of the filters is +/-700 kHz @ 3dB. Narrower bandwidth limits the maximal bitrate.
Figure 5 depicts a recommended 5MHz discrete passive filter.
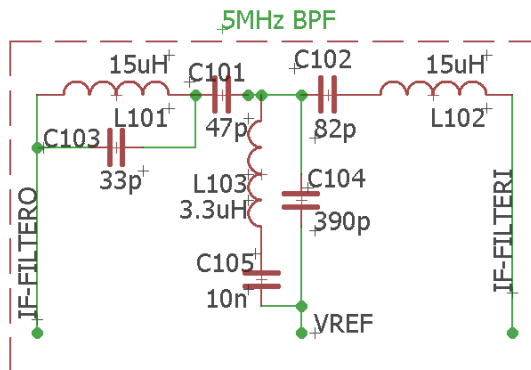
Figure 5 - 5MHz bandpass filter

2.5.3 External Crystal

The device operates with a low cost, small size 16MHz crystal connected between OSCI and OSCO pins. Each of these pins should be connected to the DGND via a load capacitor. The load capacitors values should be determined according to the crystal manufacturer's recommendations and the actual PCB layout. The PCB traces should be as short as possible. The overall crystal frequency tolerance should not exceed ± 50ppm.

2.5.3.1 Recommended Crystals

- o NDK - NX2520SA-16MHz, SMD, 2.5x2 mm
- o NDK - NX3225SA/GB-16MHz, SMD, 3.2x2.5mm
- o NDK - NX2016GC-16MHz, SMD, 2.0x1.6mm
- o ECS - ECS-160-12-37B-CTN-TR, SMD, 2.0x1.6mm

2.5.3.2 16MHz clock from an external source

The device can operate from an external 16HMz clock source that meets the requirements above.
Figure 6 depicts an external 16MHz clock connection to the device.
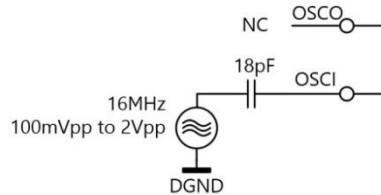
Figure 6 - External 16MHz clock connection

2.5.4 L1 and L2 inductors

The DCAN500 requires one or two inductors for its operation, depending on the desired operating frequency. High Q inductors (above 30) should be selected.

 - For full in-band operation, 5MHz - 30MHz:
- ➢ L1 - 3.3uH
- ➢ L2 - 15uH with 10nF series capacitor between L2 pin and L2 inductor.

 - For low in-band operation, 5MHz -12MHz:
- ➢ L1 - 18uH
- ➢ L2 - NC

 - For high in-band operation, 12MHz - 30MHz:
- ➢ L1 - 3.3uH
- ➢ L2 - NC

Figure 7 depicts the in-band operation inductors' connection to pins L1 and L2.
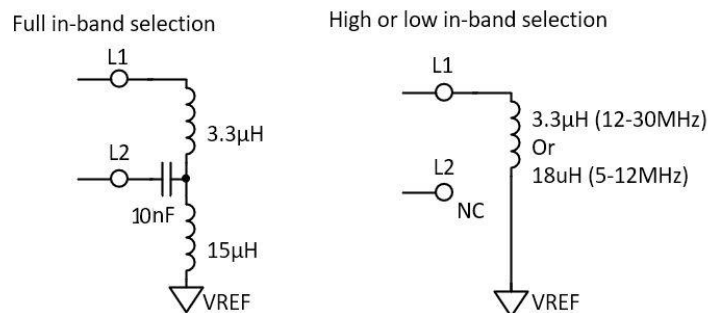
Figure 7 - L1 and L2 inductors connections

2.5.4.1 Recommended L1 & L2 inductors

Table 2 describes the recommended L1 and L2 inductors.

Table 2 - Recommended L1 and L2 manufacturers

Inductor	ABRACON	VISHAY	TDK
L1=3.3uH	815-AIML-0805-3R3K-T	ILSB0805ER3R3K	MLF2012A3R3JT000
L2=15uH	815-AIML-0805-150K-T	ILSB0805ER150K	MLF2012C150KT000
L1=18uH	815-AIML-0805-180K-T	ILSB0805ER180K	MLF2012C180KT000

2.5.5 Optional EMC chip-bead (L5)

For enhanced mitigation of high harmonics above 30MHz conducted over the powerline, it is recommended to add L5 in series to the coupling capacitor C4 (see Figure 4).

Table 3 describes the recommended EMC chip-beads.

Table 3 - Recommended L5 (optional)

LAIRD	MI0603L301R-10
LAIRD	HZ0603A222R-10
TDK	MMZ1608Q

2.5.6 Ceramic capacitors

Low ESR capacitors will provide better performance. X5R and X7R capacitors are recommended, especially for Vcap (C8) and PLLCAP (C3).

2.5.7 TXO output level and drive control

The TXO pin output level and drive capability to the powerline are controlled by REG_*1[3]*, as described in Table 4.

Table 4 - TXO signal level

TXON State	REG_1[3]	TX level [V-p-p]
High	'0'	1
	'1' (Default)	2
Low (Rx)		High Z

Setting the TXO output drive capability is made by configuring REG_*1[0]*, as described in Table 5.

Table 5- TXO output drive control

TXON State	REG_1[0]	Output drive [A]	Impedance [Ω]
High	'0' (Default)	33mA	18 [1]
	'1'	66mA	
Low (Rx)		Disabled	5.3k [2]

[1]Series output impedance
[2]Input impedance referenced to VREF

2.5.8 Powerline coupling

The DCAN500 is coupled to the powerline through a single coupling capacitor C$_{coupling}$ that blocks the DC, typically 2.2nF. The capacitor voltage rating depends on the powerline voltage and its expected impulses.
For high voltage powerline applications (e.g. EV battery monitoring system or solar panels), it is required to add proper galvanic isolation.

2.5.9 External protection network

It is recommended to add an external diode protection network before C$_{coupling}$ to the powerline to protect the device from high powerline impulses (above 2 V-P-P). The protection network consists of three Schottky diodes serially connected (for both polarities), with low capacitance (< 10pF) and fast response time (e.g. BAS70-04).

2.5.10   Recommended connection to power-supply

The DCAN500 carrier signal may attenuate strongly due to power-supplies internal filtering capacitors. It is recommended to add an inductor (>22uH) or ferrite bead (>100Ω @ 5MHz-30MHz) in series to the power supply connection to the DC powerline to avoid carrier signal attenuation.

Figure 8 depicts a typical DCAN500 and its 3.3V power-supply connection to DC powerline.
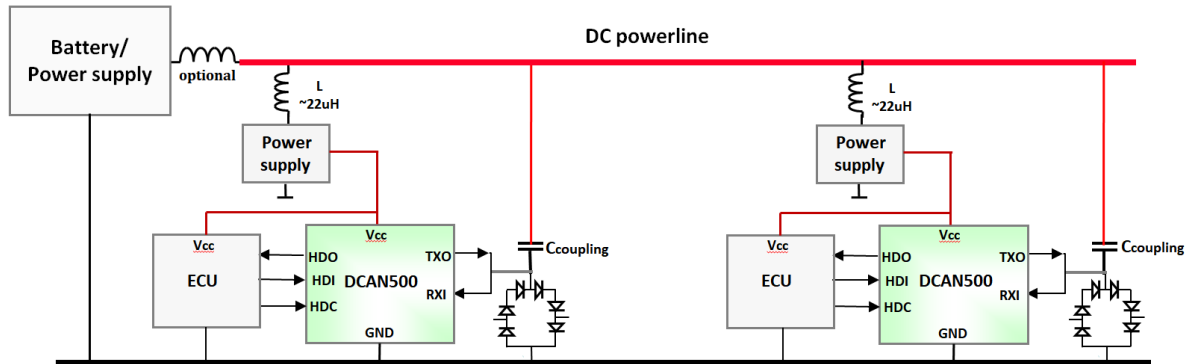
Figure 8– DCAN500 and its 3.3V power-supply connection to the powerline.

---

3.   DCAN500 Operation

3.1   Interfacing to CAN ECU

The DCAN500 interfaces directly to ECU CAN-TX and CAN-RX pins, (3.3V logic) (TRANSC_EN pin must be tied to GND). Optionally, it is possible to interface to traditional CAN-BUS through CANH and CANL signals using an external CAN transceiver (TRANSC_EN pin should be left floated). However, do not use a mixed network containing both traditional CAN-BUS nodes and DCAN500 nodes.

The CAN communication protocol uses five pins as described in Table 6.

Table 6 - CAN interface pins

HDI	Data Input from the ECU.
HDC	Data/Command select input. When pulled down, the DCAN500 enters command mode, enabling access to DCAN500 control registers.
HDO	Data output to the ECU
RTR	Ready to Receive output (optional feature). Indicating that the device is ready to receive new data bytes from the ECU (i.e. RX-FIFO is not almost full). Used to control the data flow between the ECU and the DCAN500 (see 3.5)
TRANSC_EN	External CAN Transceiver enables pin input. When interfacing an ECU this pin must be tied to GND. HDI is loopback to the HDO. When interfacing a CAN transceiver this pin can be left floating. No loopback between HDO and HDI.

Figure 9 depicts a typical DCAN500 to ECU interface connection.
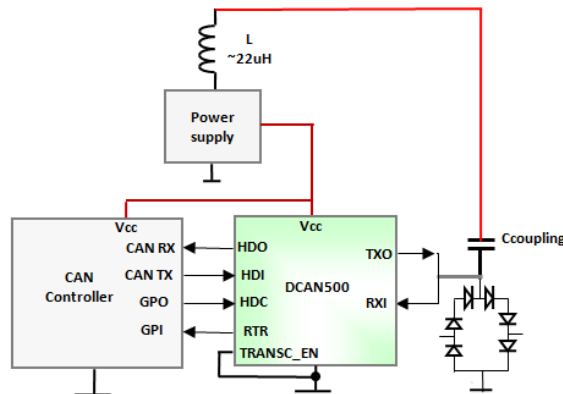
Figure 9 - DCAN500 to ECU interface

Figure 10 depicts DCAN500 to CAN module connection via CAN transceiver.
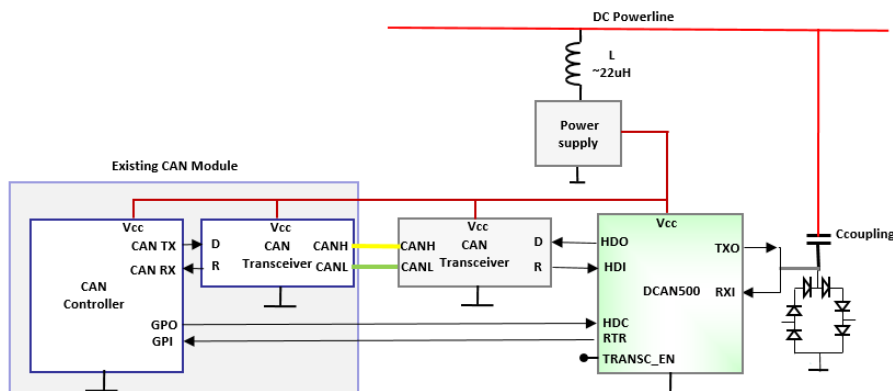
Figure 10 - Optional DCAN500 to ECU interface via CAN transceiver

3.2 DCAN500 Messages

3.2.1 Message structure

The DCAN500 is a powerline CAN gateway device. Each CAN message transferred by the ECU (*CAN-Message*) is constructed into a single DCAN500 CAN frame transmitted over a powerline (*CAN-Frame*).

A *CAN-Frame* is constructed from a *Start-frame* consisting of arbitration and preamble patterns followed by packet/s of data bytes (at least 1 packet), and terminated with a *Frame-End,* indicating the last packet of the frame. An Error Correction Code [ECC] protects each data packet.

A *CAN-Frame* may be longer than *CAN-Message*. Therefore, the device has an automatic provision ('*DUMMY-MSG'*) to prevent bus contention due to the short delay between two CAN messages (see 3.2.2.1).

- *Start - frame* pattern length: **170 μs minimum**
- Packet length: **180 μs**

Recommendation: The delay between two consecutive ECU CAN-Messages ≥ 400μs (see Figure 12).
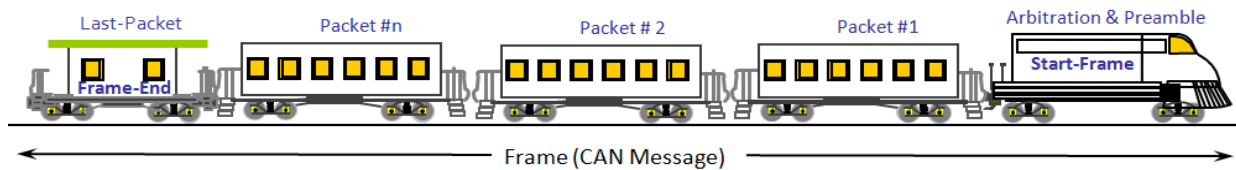
Figure 11 - CAN Frame structure
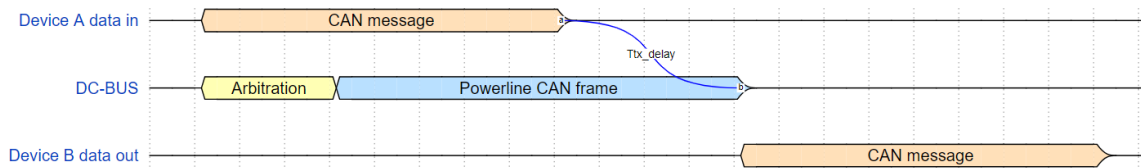
Figure 12 – Typical TX-RX CAN message flow

3.2.2 Transmit flow

Upon receiving a CAN message from its ECU, the device generates a *Start-frame* pattern transmission over the powerline. The *Start-frame* consists of unique powerline arbitration based on the ECU's CAN message 11 LSB ID.

When DCAN500 wins the arbitration over the powerline, it will respond to the ECU with ACK at the CAN ACK slot, and transmit a CAN frame to all attached DCAN500 nodes over the powerline.

If DCAN500 loses the arbitration over the powerline due to other device transmissions, it will generate a NACK at the ACK slot, abort transmission, receive the *CAN-Frame* from the powerline and transfer it to its ECU.

It is up to the ECU (upper layer) to determine whether to re-transmit the *CAN-Message* or discard it.

3.2.2.1 DCAN500 '*DUMMY-MSG'* description (CAN massage ID zero and DLC zero are preserved)

A *DUMMY-MSG* is a "zero" CAN message, consists of ID = 0, and DLC = 0. Such a message always wins the local CAN arbitration with its ECU, generating an artificial delay in the ECU TX flow.

The *DUMMY-MSG* is generated automatically by the DCAN500 to the HDO pin when it detects that the ECU starts a new CAN message transfer while the previous CAN message is still being transmitted over the powerline. The ECU loses the arbitration and waits for the duration of the *DUMMY-MSG* and becomes a receiver node without increasing its error counters.

Also, when the ECU starts a new *CAN-Message* while the DCAN500 is in the middle of CAN frame reception from the powerline, the DCAN500 will initiate transfer a *DUMMY-MSG*, until completion of powerline *CAN Frame* reception.

To conclude, the DCAN500 will always produce *DUMMY-MSGs* as long as the powerline is not idle while ECU is starting a new CAN message transfer.

Requirements:

Avoid using CAN message with ID = 0 as part of CAN messages for DUMMY-MSG proper operation.

Avoid using CAN message with DLC = 0 as part of CAN messages for DUMMY-MSG proper operation.

ECU may use the DCAN500 pin 12 BUS_BUSY output indication to monitor the powerline BUS activity.

3.2.3   Receive flow

Upon detecting a powerline *CAN-Frame*, the frame is automatically decoded into a *CAN-Message*, inserted to the RX-FIFO, and then transferred to ECU via HDO pin.

In case the ECU starts transferring *CAN-Message* while the DCAN500 starts transferring its received powerline *CAN-Frame* to the ECU, a local arbitration between ECU and the DCAN500 begins (same as performed in CAN-BUS). If the ECU wins the local arbitration, the DCAN500 will stop the transfer of the powerline *CAN-Frame* and will get the ECU's CAN message for transmission over the powerline. Then, the DCAN500 will automatically re-transmit the powerline CAN message to its ECU, until successful transfer.

When the ECU loses the local arbitration to the DCAN500, the powerline *CAN-Frame* will be fully transferred to the ECU. Then, ECU may re-transmit its CAN message.

Recommendation: Use the RTR pin to poll the powerline status and pause the ECU's CAN-Message transfer, allowing extraction of the stored CAN-Frame/s from the RX-FIFO (see 3.5).

3.2.4   TX-RX flow examples

3.2.4.1   Example 1 - ECU consecutive *CAN-Messages* transmission

The CAN protocol defines the 3-bit minimal time between two consecutive transmissions. Since the powerline *CAN-Frame* may longer than an ECU's message, therefore the ECU must wait until the powerline is idle.

In this example the DCAN500 issues *DUMMY-MSG* (empty message) to its ECU to delay the transfer of a new message until completion of the ongoing CAN powerline transmission. Figure 13 depicts two CAN message A and B with a delay of less than 400µs. Although CAN message A is still transmitted over the powerline. The ECU starts a new message B. Upon detection of the SOF of the new message B, the DCAN500 immediately initiates a DUMMY-MSG. When the *DUMMY-MSG* is completed (ACK by ECU), the powerline is idle again and the ECU re-transmits the *CAN-Message* B by the DCAN500 over the powerline.
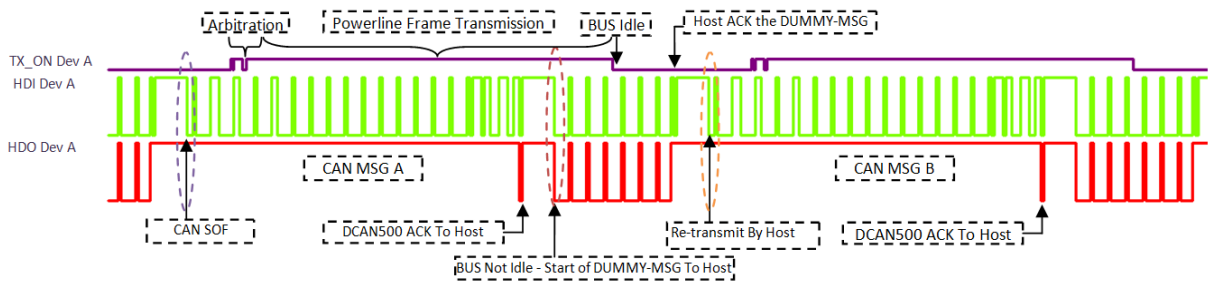
Figure 13 - ECU consecutive CAN message transmission example

Example 2 - ECU generates new SOF while receiving powerline CAN frame from other DCAN500 nodes.

This case is divided into 2 sub-cases:

Case A -Powerline CAN message was fully decoded and is ready to be transferred to ECU.

In this case, the DCAN500 starts transferring the powerline *CAN-Frame* to the ECU.
If the ECU starts at the same time transferring its *CAN-Message*, a local arbitration will take place between the DCAN500 and the ECU (same as performed in CAN-BUS). The message with the higher ID priority will win.
In case the DCAN500 loses the local arbitration, it will be become an RX station, transmitting the ECU *CAN-Message* over the powerline. Then, the DCAN500 will retransmit its waiting powerline *CAN-Frame* to ECU again.

Case B - Powerline CAN message is detected but not fully decoded before ECU new SOF.

In this case, the DCAN500 has detected a new powerline *CAN-Frame* before the start of a new *CAN-Message* from the ECU. The DCAN500 issues a *DUMMY-MSG* until the powerline is idle again. The ECU loses the local arbitration and becomes a receiver. The DCAN500 repeats to issue *DUMMY-MSG* as long as the powerline is not idle.

Figure 14 depicts an example of Case A and Case B.

The sequence starts when CAN message B is transmitted to the DCAN500 while the powerline is Idle. After a while, ECU DEV A tries to transmit its CAN message A. Due to the fact the DCAN500 has already started detecting DEV B's powerline *CAN-Frame*, it will issue *DUMMY-MSG* to DEV A ECU until the powerline is Idle again.
Then, the DEV B message is ready to be sent. DCAN500 DEV A will start local arbitration with its ECU. ECU DEV A wins local arbitration and its *CAN-Message* is sent over the powerline. Then, the DCAN500 re-transmit DEV B MSG to its ECU successfully. ECU DEV B receives successfully DEV A CAN MSG.
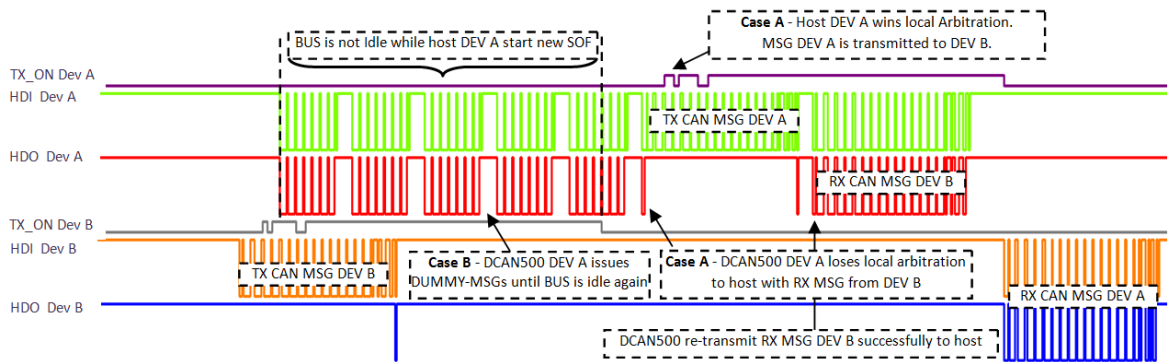
Figure 14 - ECU new SOF while receiving powerline CAN frame from other DCAN500 node examples

3.3 Device Configuration

3.3.1 ECU Bitrate selection

The DCAN500 supports CAN A/B bitrates of 83.3kbit/s, 125kbit/s, 250kbit/s, 500kbit/s, and 1Mbit/s (see section (10.
Requirements:
When using CAN host bitrate of 500kbit/s, CAN message DLC ≥ 3 is required.
When using CAN host bitrate of 1Mbit/s, CAN message DLC ≥ 7 is required.

Bitrate selection by pins BR_SEL[1:0] as described in Table 7.

Table 7 - CAN bitrate selection

BR_SEL[1:0]	ECU bitrate [kbit/s]
'00'	83.3
'01'	125
'10'	250
'11'	500

The bitrate selection is sampled once after the power-up/reset event, and remains until the next power-up/reset event.

Setting of ECU bitrate of 1Mbit/s requires access to the DCAN500 internal register.
Please refer to **section 10** for more details.

3.3.2 Carrier frequency configuration

The carrier frequency can be defined between 5MHz to 30MHz with a spacing of 100 kHz (Total of 251 selectable carriers). The active carrier frequency selection is made by configuring REG_*2* (see 0). Upon completion of configuration, the DCAN500 will update its operating carrier frequency within 1msec. During this period, the

DCAN500 is kept in *Soft-Reset* and will not communicate with its ECU nor detect new *CAN-Frames* from the powerline.

Recommendation: When setting multiple DCAN500 networks to operate over a single powerline, select carrier frequencies spaced more than 1.5MHz from each other.
The carrier-selected value is calculated as given in Equation 1 (3).
Equation 1

$$REG\_2 = (Carrier\ Freq.\ [MHz] - 5) * 10 \tag{1}$$

EXAMPLE 1

- ❖ When setting the frequency to 14.1MHz:
  REG_2 = (14.1 - 5) * 10 = **0x5B**

EXAMPLE 2

- ❖ When Setting to 5MHz:
  REG_2 = (5 - 5) * 10 = **0x00**

3.4 RX-FIFO handling (optional feature)

The DCAN500 has an internal mechanism to buffer between received *CAN-Frames* and its transfer to the ECU. The mechanism contains RX-FIFO with maximal 1024 bytes and RTR interrupt output pin indicating when RX-FIFO is filled up to a configured threshold.

3.4.1 RX-FIFO RTR thresholds configuration

Users may define RX-FIFO interrupt (RTR pin) thresholds according to the network expected payload. The interrupt is invoked on RTR pin 11.

The RX- FIFO RTR thresholds control registers are described in Table 8.

Table 8 - RX-FIFOs RTR threshold control

FIFO Threshold Description	Default threshold [Data bytes]	Related Control registers	Comments
RX-FIFO-almost-full[9:0]	256	REG_6[1:0], REG_5[7:0]	Define how many data bytes are inserted to RX-FIFO before the interrupt raise. Indicates the ECU to start reading the received frame. See Example 1 in 3.4.1.1

3.4.1.1 Example 1 - *RX-FIFO-almost-full* threshold setting

Configuration *RX-FIFO-almost-full* threshold to 0x384 results in interrupt triggering when there are at least 900 data bytes are stored in Rx-FIFO:
REG_6 = 0x03
REG_5 = 0x84
For more details, please refer to the RTR handling in 3.5.

3.4.2 RX-FIFO reset control (Soft-Reset event)

ECU may reset the RX-FIFO stored data by activating a DCAN500 *Soft-reset* event.

A *Soft-reset* is activated while the HDC pin 6 is low.

During *Soft-reset*, the DCAN500 performs only write and read to/from DCAN500 control registers. Neither transmission nor reception to/from the powerline is available. The RX-FIFO is kept in reset.

3.5 RTR pin handling (optional feature)

The Ready to Receive (RTR, pin 11) output allows the ECU to control *CAN-Message* transfers from DCAN500.
When RTR is high, ECU can transfer new *CAN-Message* via HDI pin. When low, ECU should pause its transfer, allowing the DCAN500 to extract stored powerline *CAN-Frames* from the RX-FIFO.
The RTR output state is subject to *RX-FIFO-almost-full* threshold configuration (see Table 8), allowing RX-FIFO overflow protection.
Recommendation: In a high payload network, sample the RTR state before a new ECU CAN-Message transfer.

3.6   DCAN500 UUID

Each DCAN500 device is hard-coded with a 48 bit universally unique identifier (UUID[47:0]).
The UUID is stored in REG_59 to REG_5E and can be retrieved using the READ-REG commands (see 5.6 to 5.11).

3.7   Typical set-up and operation

1. Interface HDI, HDO, HDC, and RTR pins.
2. Set BR_SEL[1:0] pin (see 3.3) or set 1Mbit/s (see)
3. Set TRANSC_EN pin according to HDI and HDO pins interface (see 3.1).
4. Select a carrier frequency (default 13MHz) (see Section 3.3.2 Carrier frequency configuration).
5. Transmit CAN message via HDI pin to the powerline (with RTR status polling).
6. Receive CAN message from the powerline via HDO pin.

4.   Power Operation Modes

The DCAN500 has three power operation modes; Normal, (Sleep) Standby, and Sleep.

4.1   Normal mode

In Normal mode, the DCAN500 is either in RX mode, listening for a powerline CAN frame, or in TX mode, transmitting a CAN frame over the powerline.

4.2   Standby mode

The DCAN500 enters Standby mode upon wake-up from Sleep mode, while the NSLEEP pin is still low. The DCAN500 is kept in *Soft-Reset*, whereas communication with the ECU is suspended until the NSLEEP pin is set High.

4.3   Sleep modes (power-saving)

The DCAN500 has four Sleep modes for best power consumption/performance during Sleep. During this mode, only a small amount of hardware is operational mainly to detect wake-up messages (*WUM*) from the powerline and returning to Normal mode operation.

Table 9 describes DCAN500 sleep modes.

Table 9- Sleep modes description

Sleep mode	Description	Typical Power consumption [A]	Performance
Enhanced sleep (SLP1)	The device wakes-up every 32ms to sense the powerline for WUM detection.	120μ	Wake-up detection with-in 64mSec. Best in a noisy environment.
Fast wake-up (SLP2)	The device continuously monitors the powerline for WUM detection.	1000μ	Fast wake-up detection with-in 250uSec.
Low-power (SLP3)	The device wakes-up every 32ms to sense the powerline for WUM detection.	85μ	Wake-up Detection with-in 64mSec.
Deep Sleep (SLP4)	The device does NOT wake-up to sense for bus activity, staying in deep sleep. Wake-up only locally by the ECU.	65μ	No bus wake-up detection.

The Sleep modes use four interface pins as described in Table 10.

Table 10- Sleep interface pins

NSLEEP	Digital input	High - Normal mode is active. Low - Sleep /Standby mode is active.
INH	Digital output	Output indication to Inhibit ECU. High - Normal mode is active. Low - Sleep mode is active.
BUS_BUSY	Digital output	Asserted high while wake-up message is being detected/transmitted over the powerline.

HDC	Digital input	**Normal mode**- ECU Command mode / chip select. **Sleep mode** - ECU wakes-up the DCAN500 locally by toggling the HDC high-low-high. The DCAN500 then exits the Sleep mode to Standby mode (NSLEEP still asserted low), or Normal mode (NSLEEP is high).

4.3.1 Wake-up message (WUM)

When *Auto-WUM* is enabled (REG*[3]='1'*), upon the rise of the NSLEEP pin, the DCAN500 transmits a broadcast *WUM* over the powerline, to wake-up all network-connected devices.
ECU can configure the length of the WUM as described in Table 11.

Table 11 - Wake-up message length configuration

REG_3[2]	Wake-up message length
0	*SLP2* - 250usec / *SLP1, SLP3* - 75msec
1	*SLP2* - 1.5msec / *SLP1, SLP3* - 150msec

During WUM transmission, the BUS_BUSY pin is asserted high until WUM transmission is completed, indicating to the ECU the wake-up process status. ECU shall wait for the BUS_BUSY drop, before initiating new bytes transfer.

4.3.2 Entering Sleep mode

During Sleep mode, the device is kept in a Soft-reset state and will not transfer data bytes from the ECU nor receive data frames from the powerline. When the device enters Sleep mode, the INH pin is asserted low.
There are two ways to enter Sleep mode;

4.3.2.1 Enter Sleep by NSLEEP

By asserting the NSLEEP pin low, the DCAN500 will enter Sleep mode.

4.3.2.2 Enter Sleep by register setting

By setting REG_*3[7]* high, the DCAN500 will enter Sleep mode, and reset automatically REG_*3[7]* to low.

4.3.3 Exiting Sleep mode

There are three ways to exit Sleep mode. When exiting Sleep mode, the INH pin is raised and the device switches to Standby or Normal mode.

4.3.3.1 Exit Sleep by *WUM* detection

Upon detection of a *WUM*, the device immediately exits Sleep mode, INH pin rises and the device enters Standby mode.
In case the NSLEEP pin is low, the device remains in Standby mode, where the device is kept in *Soft-reset*.
In case the NSLEEP pin is high, the device immediately switches to Normal mode.
During WUM reception, the BUS_BUSY pin is asserted high until WUM reception is completed, indicating the ECU on the wake-up process status. ECU shall wait for BUS_BUSY to drop, before initiating new bytes transfer.

4.3.3.2 Exit sleep by NSLEEP pin

Upon detection of NSLEEP pin rise, the device immediately exits Sleep mode, INH pin rises, and enters Normal mode. When *Auto-WUM is* enabled, a WUM is transmitted over the powerline (see 4.3.1).

4.3.3.3 Exit Sleep by toggling HDC

Upon detection of HDC pin toggle high-low-high, the device immediately exits Sleep mode, INH pin rises, and enters Standby mode.
In case the NSLEEP pin is still low, the device remains in Standby mode, where the device is kept in *Soft-reset*.
In case the NSLEEP pin is high, the device immediately switches to Normal mode.
In this case, the WUM will NOT be transmitted over the powerline.
ECU shall use the HDC pin to exit Sleep mode when the NSLEEP pin is not connected.

4.3.4 Sleep modes description

ECU can select between four Sleep modes (see 5.3).

4.3.4.1 Enhanced Sleep mode (*SLP1*)

By setting REG_*3[1:0]* = *'00'*, the enhanced Sleep mode (*SLP1*) is selected.
When entering*SLP1*, the device wakes-up every 32ms periodically to monitor (sense period) for activity on the powerline. If a WUM is detected, the device exit Sleep modes as described in section 4.3.3.1, otherwise the device return to Sleep mode until the next sense period, and so on...

4.3.4.2 Fast wake-up Sleep mode (*SLP2*)

By setting REG_*3[1:0]* = *'01'*, the Fast wake-up Sleep mode (*SLP2*) is selected. The device continuously monitors the powerline for WUM detection. It allows fast WUM detection within 250usec. When WUM is detected, the device exit Sleep mode as described in section 4.3.3.1.

4.3.4.3 Low-power Sleep mode (*SLP3*)

By setting REG_*3[1:0]* = *'10'*, the low-power mode (*SLP3*) is selected. The device wakes-up every 32msec periodically to monitor (sense period) for activity on the powerline. If a WUM is detected, the device exit Sleep modes as described in section 4.3.3.1, otherwise the device return to Sleep mode until the next sense period.

4.3.4.4 Deep Sleep mode (*SLP4*)

By setting REG_*3[1:0]* = *'11'*, the Deep Sleep mode (*SLP4*) is selected. The device will NOT wake-up to monitor (sense) the powerline for activity, rather than stay in deep sleep, whereas all its analog resources are shut down to maintain the lowest power consumption.
The device can exit Deep Sleep mode locally only, either by the NSLEEP or by HDC pins (see 4.3.3.2 and 4.3.3.3).

4.3.5 Sleep modes Examples

4.3.5.1 Sleep Example 1 - Enter by NSLEEP, Exit Sleep mode by *NSLEEP & WUM*

Figure 15 depicts entering sleep by NSLEEP and exit sleep by NSLEEP pin (Node A) and WUM detection (Node B).
In this example, the ECU wakes-up device Node A by raising the NSLEEP pin. Upon pull-up the NSLEEP pin, the INH pin is raised and a WUM is transmitted over the powerline (*Auto-WUM* is enabled) to wake-up Node B.
While transmitting the WUM, device Node A asserts BUS_BUSY pin high. After completion of WUM transmission, the HDO is raised again (can be used as signal/interrupt to ECU). At the Node B side, during its sensing period (e.g. *SLP1),* the WUM is detected, and the INH rises while switching to Standby mode. Node B BUS_BUSY pin is asserted high for the reaming duration of WUM reception. Then, ECU Node B raises the NSLEEP pin and the device switches to Normal mode.
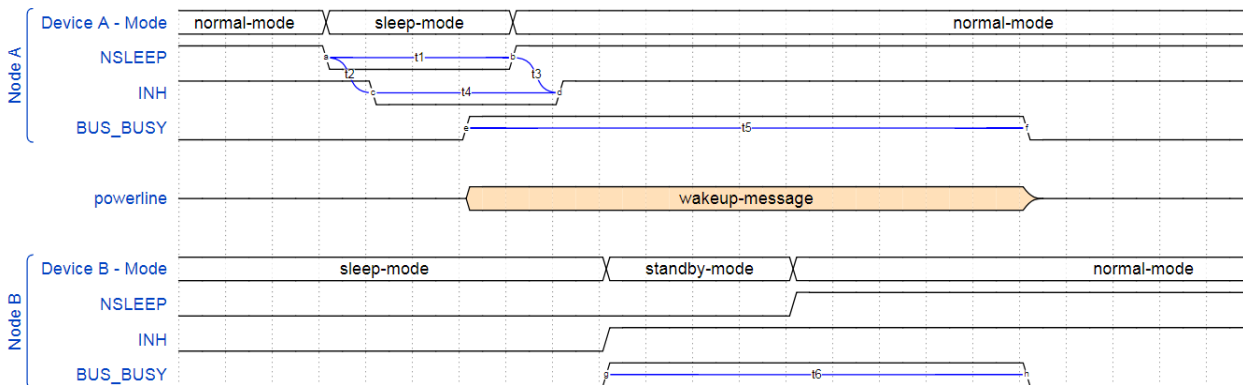
Figure 15 - Enter sleep by NSLEEP, Exit sleep by *NSLEEP& WUM*

4.3.5.2 Sleep Example 2 - Enter sleep by control register bit, exit sleep by HDC

Figure 16 depicts entering sleep by setting REG_*3[7]* high and exiting Sleep mode by toggling the HDC pin.
In this example, ECU configured REG_*3[7]* high using Command mode, the device enters Sleep mode, and INH pin drops. After a while, ECU toggle HDC pin low to high, and the device exits Sleep mode without transmitting the WUM, raising the INH pin and switching to Normal mode again.
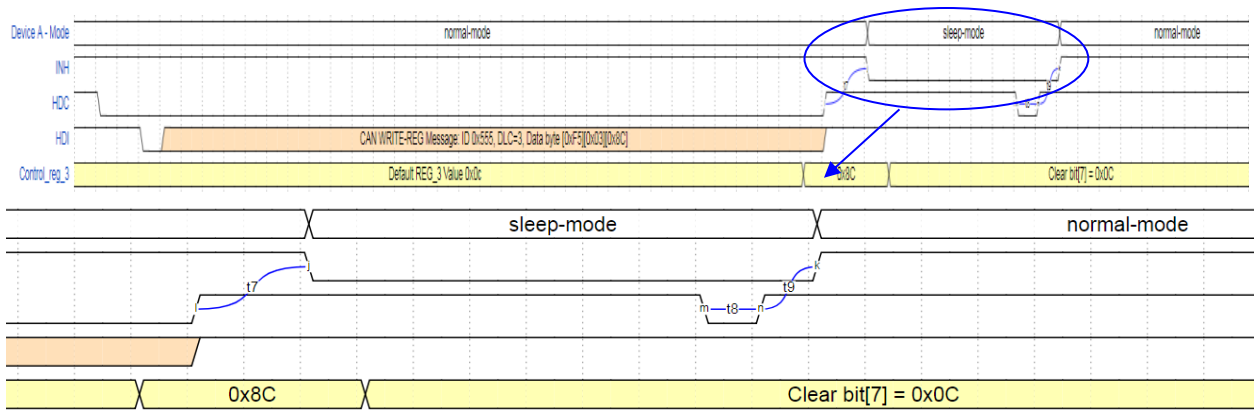
Figure 16 - Enter sleep by control register bit, Exit sleep by *HDC*

5. DCAN500 Registers

The DCAN500 contains internal registers for configuration and status checks. Each of these registers is accessible by the ECU for *Read* and *Write* operations. The access method to these registers is described in section 5.12. This section elaborates on the registers and their default values after power-up/reset.

Table 12 - Registers summary table

Register name	Addr.	Description
REG_1 - 'Device Control 1'	0x01	Transmit level control
REG_2 - 'Frequency Select'	0x02	Carrier frequency selection
REG_3 - 'Sleep & IO Control'	0x03	Sleep modes and IO pins
REG_5 - 'RX-FIFO Threshold 1'	0x05	*RX-FIFO-almost-full* threshold lower nibble
REG_6 -'RX-FIFO Threshold 2', *Rx-FIFO overflow* indication	0x06	*RX-FIFO-almost-full* threshold higher nibble, *Rx-FIFO overflow* error indication.
REG_59 - DCAN500 UUID[47:40]	0x59	Read only - UUID[47:40]
REG_5A - DCAN500 UUID[39:32]	0x5A	Read only - UUID[39:32]
REG_5B - DCAN500 UUID[31:24]	0x5B	Read only - UUID[31:24]
REG_5C - DCAN500 UUID[23:16]	0x5C	Read only - UUID[23:16]
REG_5D - DCAN500 UUID[15:8]	0x5D	Read only - UUID[15:8]
REG_5E - DCAN500 UUID[7:0]	0x5E	Read only - UUID[7:0]

5.1 REG_1 - 'Device Control 1' (Address 0x01)

| Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
[1]	[1]	[1]	[1]	R/W [1]	[0]	[0]	R/W [0]
1	1	1	1	TX signal level	0	0	Enable TXO high power

Bit [0]   - Enable TXO high power. Set this bit to enable maximal TXO drive of 66mA, clear this bit for maximal TXO drive of 33mA (see section 2.5.7).
Bit [1]   - '0'
Bit [2]   - '0'
Bit [3]   - TX signal level control at TXO pin: '0' - 1Vpp, '1'- 2Vpp (see section2.5.7).
Bit [7:4]   - '1111'
R - Readable bit,   W - Writeable bit   [x] - Value on power-up. '1' - bit is set; '0' - bit is cleared

5.2 REG_2 - 'Frequency Select' (Address 0x02)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
R/W [0]	R/W [1]	R/W [0]	R/W [1]	R/W [0]	R/W [0]	R/W [0]	R/W [0]
Carrier Frequency Configuration							

Bits [7:0]   - Carrier Frequency configuration for in-band operation. Default configuration is 13MHz (See section 3.3.2 - 3.3.2Carrier frequency configuration).

5.3   REG_3 - 'Sleep & IO Control' (Address 0x03)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
W [0]	[0]	[1]	[0]	R/W [1]	R/W [1]	R/W [0]	R/W [0]
Enter Sleep mode	0	1	0	Auto WUM	Long WUM	Sleep modes selection	

Bit [1:0]   - '00' - Enhanced Sleep mode [*SLP1*], '01' -Fast wake-up Sleep mode[*SLP2*], '10' - low-power sleep mode [*SLP3*], '11' - Deep Sleep mode [*SLP4*] (see section4.3).
Bit [2]   - Control powerline wake-up message duration (see Table 11).
Bit [3]   -Auto wake-up message (WUM): '0' disables transmission of WUM after wakeup from NSLEEP pin.
Bit [4]   - '0'
Bit [5]   - '1'
Bit [6]   - '0'
Bit [7]   -Enter Sleep mode reg. Instead of entering Sleep mode through the NSLEEP pin, the user can activate the Sleep mode selected in bits[1:0], by setting bit[7]. After entering Sleep mode, bit[7] is automatically cleared to '0'.

R - Readable bit,   W - Writeable bit   [x] - Value on power-up. '1' - bit is set; '0' - bit is cleared

5.4   REG_5 - 'RX-FIFO Almost full Threshold 1' (Address 0x05)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
W/R[0]	R/W [0]	R/W [0]	R/W [0]	R/W [0]	R/W [0]	R/W [0]	R/W [0]
RX-FIFO-almost-full[7:0] *threshold*							

Bits [7:0]   - *RX-FIFO-almost-full* [7:0], eight LSB of *RX-FIFO-almost-full*[9:0] threshold bits[9:8] are configured in REG_6[1:0].

RX-FIFO-almost-full[9:0] threshold - default set to 256 data bytes

5.5   REG_6 - RX-FIFO Almost full Threshold 2' and overflow indication (Address 0x06)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
[0]	[0]	R	[0]	[0]	[0]	R/W [0]	R/W [1]
0	0	*Rx-FIFO* overflow	0	0	0	*Rx-FIFO-almost-full[9:8] threshold*	

Bits [1:0] -   *RX-FIFO-almost-full*[9:8],*Two MSB of RX-FIFO-almost-full*[9:0] threshold.
Bits [4:2] -   '000'.
Bits [5]   -   Read-only - *Rx-FIFO* overflow error indication.
Bits [7:6] -   '00'.
R - Readable bit,   W - Writeable bit   [x] - Value on power-up. '1' - bit is set; '0' - bit is cleared

5.6   REG_59 – UUID[47:40] (Address 0x59)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
R	R	R	R	R	R	R	R
UUID[47:40]							

Bits [7:0] - UUID[47:40]
R - Readable bit,   W - Writeable bit   [x] - Value on power-up. '1' - bit is set; '0' - bit is cleared

5.7   REG_5A – UUID[39:32] (Address 0x5A)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
R	R	R	R	R	R	R	R
UUID[39:32]							

Bits [7:0] - UUID[39:32]
R - Readable bit,   W - Writeable bit   [x] - Value on power-up. '1' - bit is set; '0' - bit is cleared

5.8   REG_5B – UUID[31:24] (Address 0x5B)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
R	R	R	R	R	R	R	R
UUID[31:24]							

Bits [7:0] - UUID[31:24]

R   - Readable bit,   W - Writeable bit   [x] - Value on power-up. '1' - bit is set; '0' - bit is cleared

5.9   REG_5C – UUID[23:16] (Address 0x5C)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
R	R	R	R	R	R	R	R
UUID[23:16]							

Bits [7:0] - UUID[23:16]

R   - Readable bit,   W - Writeable bit   [x] - Value on power-up. '1' - bit is set; '0' - bit is cleared

5.10   REG_5D – UUID[15:8] (Address 0x5D)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
R	R	R	R	R	R	R	R
UUID[15:8]							

Bits [7:0] - UUID[15:8]

R   - Readable bit,   W - Writeable bit   [x] - Value on power-up. '1' - bit is set; '0' - bit is cleared

5.11   REG_5E – UUID[7:0] (Address 0x5E)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
R	R	R	R	R	R	R	R
UUID[7:0]							

Bits [7:0] - UUID[7:0]

R   - Readable bit,   W - Writeable bit   [x] - Value on power-up. '1' - bit is set; '0' - bit is cleared

5.12   Registers configuration (Command mode)

The Command mode allows the ECU to access the DCAN500 internal registers for write and read operations. Controlling these registers is done by lowering the HDC pin and sending a pre-defined CAN-Message to the registers as described below. During Command mode, the DCAN500 is in *Soft-Reset* state, RX-FIFO is reset and all data in the FIFO is erased and the device cannot send or receive a message to/from the powerline.

5.12.1   WRITE-REG command

A Write-REG command is constructed from CAN A message, with ID = 0x555 and DLC = 3, as described in Table 13.

Table 13 - WRITE-REG command structure

1st data Byte	2nd data Byte	3rd data Byte
0xF5	Control register address	Data to write

The 1st CAN data byte is the write command byte.

The 2nd CAN data byte is the designated control register address to write to.

The 3rd CAN data byte is the data byte value to write.

The new configured value is kept until the next power-up/reset event.

For example, writing 0x34 to REG_2 (address 0x02) preformed as follows (depicted in Figure 17):

1. Lower the HDC pin (Enter Command mode).

2. Wait at least 100nsec

3. Transfer CAN A message with ID = 0x555 and 3 data bytes: [0xF5][0x02][0x34]

4. The value 0x34 is written to REG_2.

5. Wait for at least 100ns.

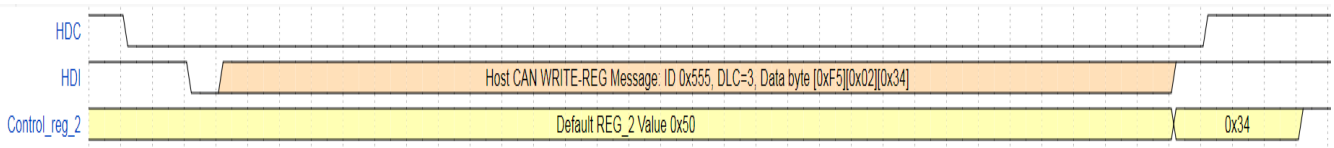6. Raise the HDC pin (Exit Command mode to Normal mode).

Figure 17 - WRITE-REG example

5.12.2   READ-REG command

A READ-REG command is constructed from a CAN A message, with ID = 0x555 and DLC = 2 as described in Table 14.

Table 14 - READ-REG command structure

1st data Byte	2nd data Byte
0xFD	Control register address

1st CAN data byte is the Read command byte.

2nd CAN data byte is the designated register address to read from.

Upon receiving a complete READ-REG CAN message, the DCAN500 will respond with feedback CAN message with ID 0x00, and DLC =1, consists of the read register value.

For example, reading from REG_3 (address 0x03) is performed as follows (depicted in Figure 18).

1. Lower the HDC pin (Enter Command mode).

2. Wait at least 100nsec

3. Transfer CAN A message with ID = 0x555 and 2 bytes: [0xFD][0x03]

4. Wait for the DCAN500 response CAN A message with ID=0x000, and one data byte that holds the value of REG_3.

5. Wait for at least 100ns.

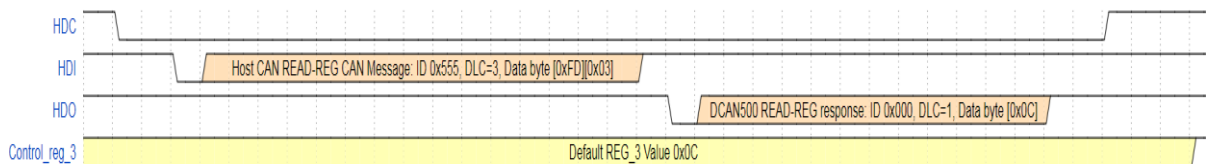6. Raise the HDC pin (Exit Command mode to Normal mode).

Figure 18 - READ-REG example

6.   Specifications

Table 15 - Absolute maximal rating

Parameter	Symbol	Comments	Min.	Typ.	Max.	Unit
Input voltage, DC	V_{im}		-0.6	3.3	3.9	V
Output voltage, DC	V_{om}		-0.6	3.3	3.9	V
Ambient temperature	T_{am}		-40		125	°C
Storage temperature	T_{sm}		-55		150	°C

Table 16 - Recommended operation conditions

Parameter	Symbol	Comments	Min.	Typ.	Max.	Unit
Supply Voltage	V_{DVCC} V_{AVCC}		3.0	3.3	3.6	V
Supply Voltage ripple	V_{CC_RIP} A_{VCC_RIP}	Max 2.5MHz, waveform type of triangular		50m		V-p-p
Ambient operating temperature range	T_A		-40		105	°C
Minimum high-level input voltage	V_{IH}		2			V
Maximum low-level input voltage	V_{IL}				0.8	V
Minimum high-level output voltage	V_{OH}		2.4			V
Maximum low-level output voltage	V_{OL}				0.4	V
Maximal output current	I_{out}	see Table 1				
Maximum input current	I_{IN}		-1		1	µA

Table 17 - Device characteristics

Parameter	Symbol	Comments	Min.	Typ.	Max.	Unit
External components requierments						
Powerline coupling capacitor	$C_{coupling}$	Capacitor rate should be selected with respect to powerline voltage		2.2		nF
Protection diodes capacitance	D_{protec}			10		pF
Capacitor at VCAP	V_{cap}		1	4.7		µF
Capacitor at PLLCAP	PLL_{cap}		1			µF
Capacitor at VREF	$VREF_{cap}$		1			µF
Inductor at L1	L1	see 2.5.4		3.3 / 18		µH
Inductor at L2	L2			15		µH
L1 pin input capacitance					1	pF
Crystal frequency	$Xtal_{freq}$	see2.5.3		16		MHz
Crystal frequency tolerance	$Xtal_{ppm}$				50	±ppm
AC signals characteristics						
Tx signal at TXO	TXO_{lev_1}	TXON high (transmission is active) see 2.5.7		1		V-p-p
	TXO_{lev_2}			2		V-p-p
TXO input impedance	TXO_{In}	TXON low (transmission is not active)	5.3k			Ω
TXO output impedance	TXO_{out}	TXON high (transmission is active)		18		Ω
TXO driving strength	I_{TXO}	TXON high (transmission is active)	33		66	mA
Rx signal at RXI	RXI_{lev}		10m		3.3	V-p-p
RXI input impedance	RXI_{In}		5.1k			Ω
Carrier Frequency in-band (channels selection)	F_c	Selectionresolutionis 100kHz, a total of 251	5		30	MHz

Parameter	Symbol	Comments	Min.	Typ.	Max.	Unit
		carrier frequencies, see 3.3.2				
Adjacent channels spacing	F_{adj}	The space between two adjacent channels operating over the same powerline.	1.5			MHz
Timing requirements of the CAN interface						
CAN bitrate	CAN_{br}	ECU CAN bitrate. see Table 7 and section 10.	83.3		1000 [1]	kbit/s
ECU TX delay	T_{tx_delay}	The minimal delay between two consecutive ECU CAN messages. see 3.2.2.1.	400			µs
RX-FIFO size		see 3.4			1024	Byte
Timing of device operation modes						
Power-cycle/ hard-reset	T_{init}	Initialization time after power-cycle or hard-reset event.		2		ms
Carrier frequency setting	T_{freq_cng}	Carreri frequnecy change process time		1		ms
Current Consumtption @ 3.3V						
Normal TX mode – low power	I_{Tx_lp}	TXON high (transmission is active)		80		mA
Normal TX mode – high power	I_{Tx_hp}			100		mA
Normal RX mode	I_{RX}	TXON low (transmission is not active)		50		mA
Enhanced sleep (SLP1)	I_{slp1}	See 4.3		120		µA
Fast wake-up (SLP2)	I_{slp2}	See 4.3		1000		µA
Low-power (SLP3)	I_{slp3}	See 4.3		85		µA
Deep Sleep (SLP4)	I_{slp4}	See 4.3		65		µA

1 CAN message DLC ≥ 3 is required for 500Kbit/s and DLC ≥ 7 for 1Mbit/s.

7. DCAN500 PCB layout recommendation

Figure 19 below are examples of the two-layer PCB layout. The separate analog ground is surrounding the analog parts of the design.


Figure 19 – Grounds layout reference

- ✓ Analog ground layer and GND PLL should be connected to the digital ground near the Expose pad.
- ✓ VCC and DGND layout traces should be as wide as possible. Connect a 0.1uF capacitor between each VCC and DGND pins, as close as possible to the pins.
- ✓ It is recommended to keep the traces connecting the 3.3V power supply to VCC pins as short as possible with wide PCB traces.
- ✓ Connect L1, L2, C13, C3, C5, C7, C8, C11, and C12 as close as possible to their pins.
- ✓ Connect R1 as close as possible to the RXI pin.
- ✓ Connect all filtering caps as close as possible to their pins.
- ✓ Connect crystal and its capacitors close to OSCI and OSCO pins. Keep DGND plan around them.

8. Package, Mechanical
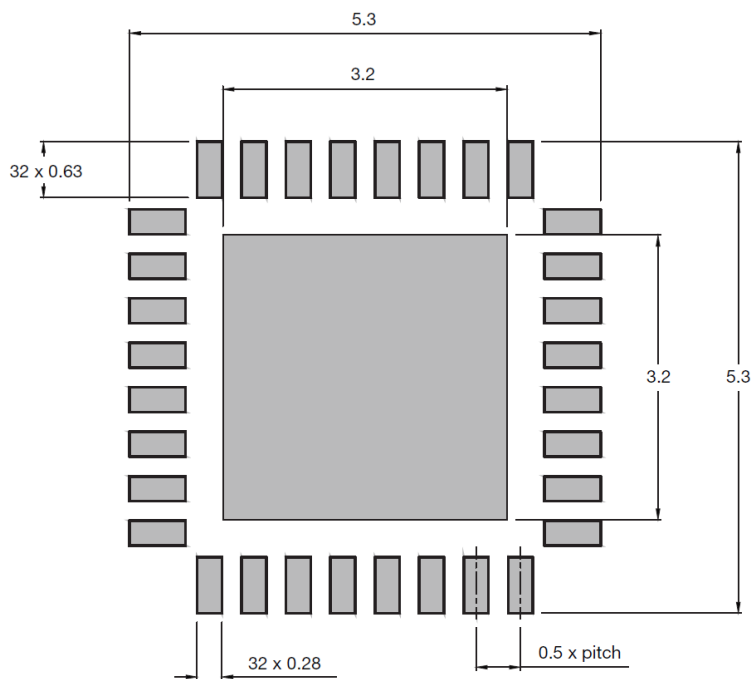
The device package is QFN 32 5mm x 5mm.

8.1 Mechanical Drawing
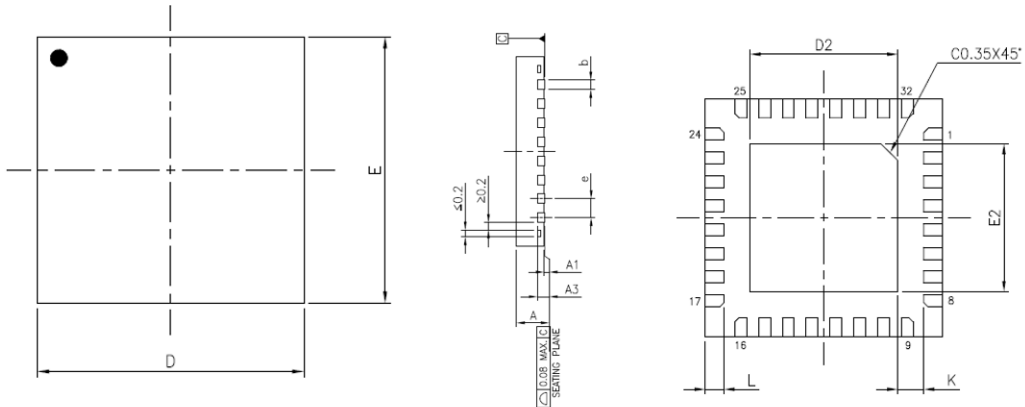
	D2			E2		
	MIN.	NOM.	MAX.	MIN.	NOM.	MAX.
	3.15	3.20	3.25	3.15	3.20	3.25

SYMBOLS	MIN.	NOM.	MAX.
A	0.70	0.75	0.80
A1	0.00	0.02	0.05
A3	0.203 REF.		
b	0.18	0.25	0.30
D	5.00 BSC		
E	5.00 BSC		
e	0.50 BSC		
L	0.35	0.40	0.45
K	0.20	—	—

8.2 PCB drawing

8.3 Soldering profile

Soldering reflow profile is according to IPC/JEDEC J-STD-020 (MSL3).

- ➢ The peak temperature (TP) is 260°C.
- ➢ Holding time is between 60 sec to 120 sec between TH min 150°C to TH max 200°C.
- ➢ Liquidus temperature (TL) is 217 °C. Liquidus time is between 60 sec to 150 sec.
- ➢ TL to TP max ramp-up is 3°C/sec.
- ➢ TP to TL max cooldown rate is 6°C/sec.
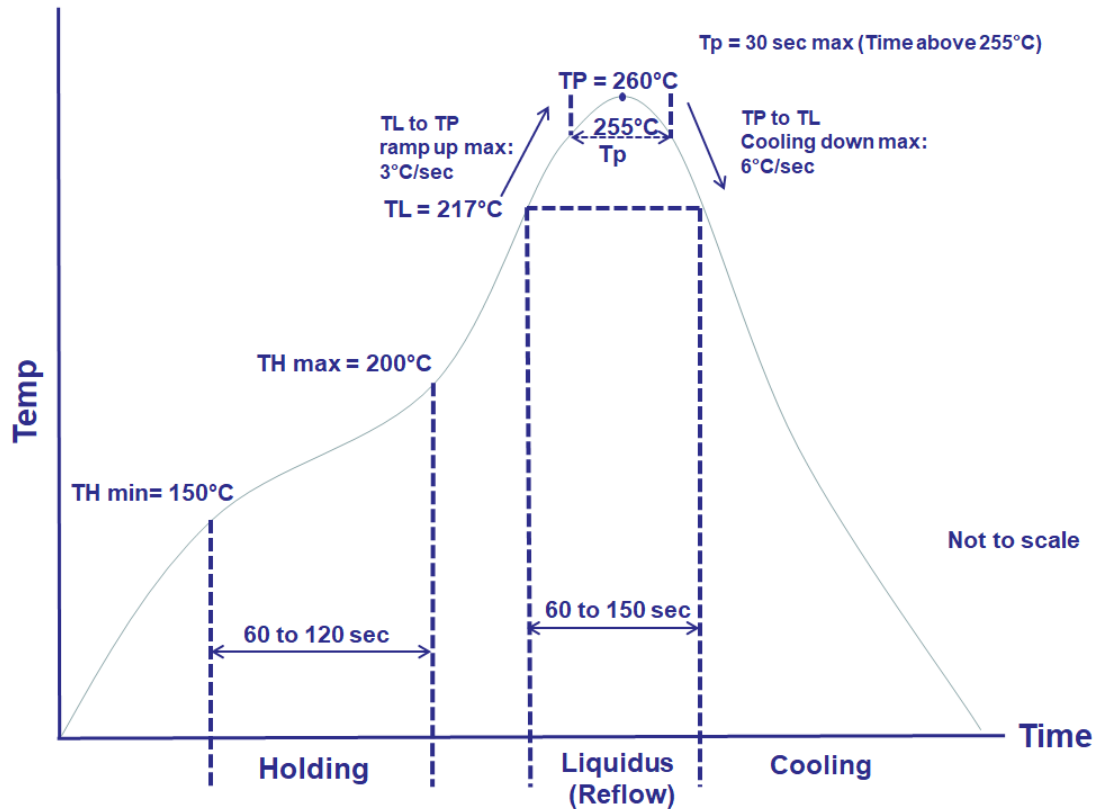- ➢ Max time above 255°C (Tp) is 30 sec.

Figure 20 - Representation of IPC/JEDEC J-STD-020 (MSL3) profile
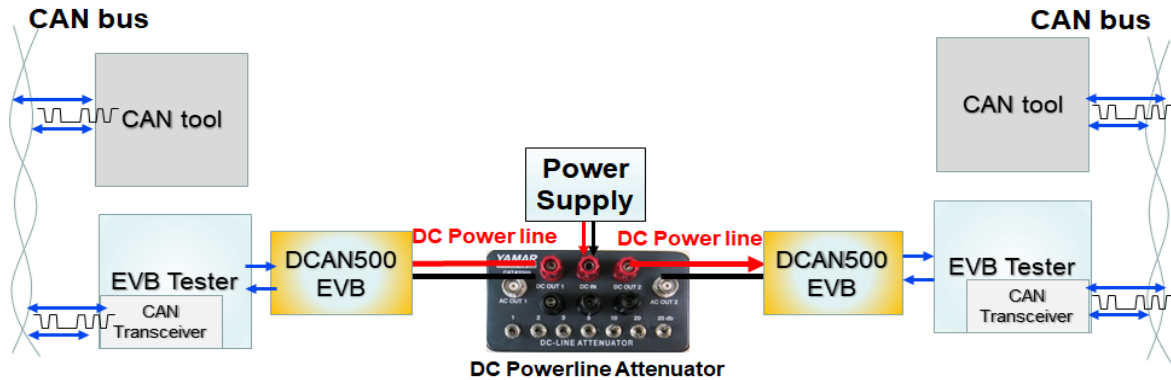
9. Test Environment

Figure 21 depicts the DC-BUS Test environment that allows testing the DCAN500 devices in the emulated lab DC powerline environment.
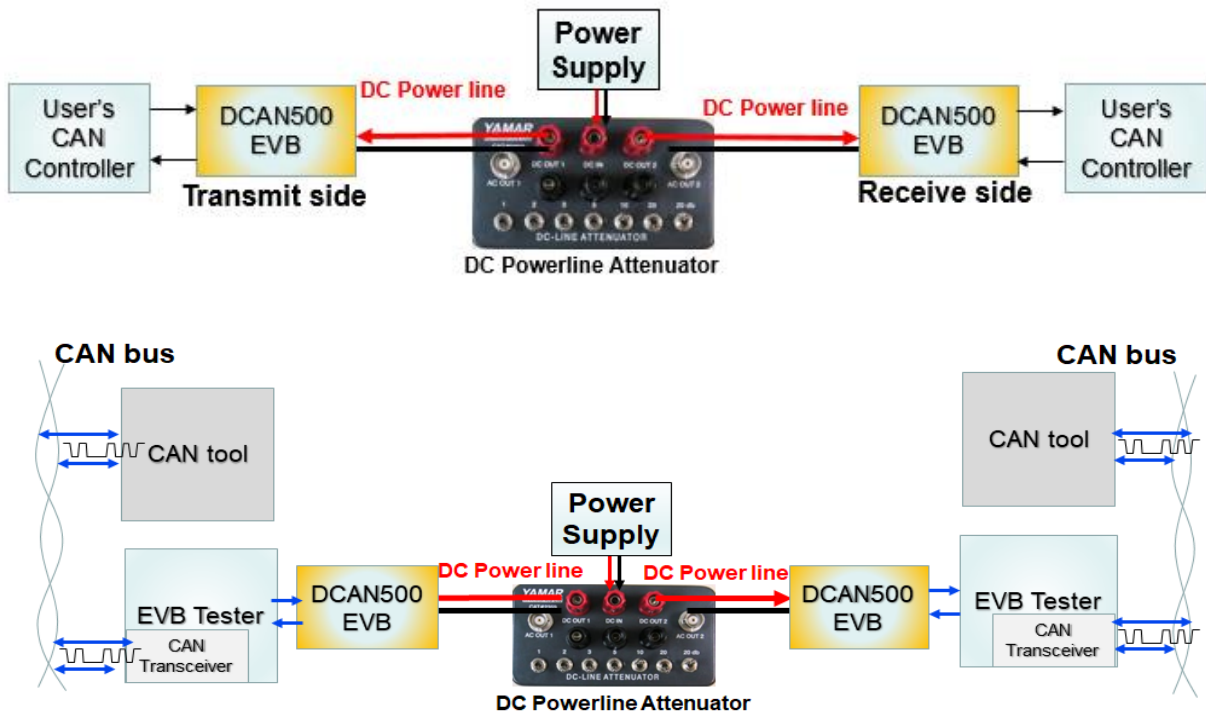

Figure 21 - DC-BUS Test environment

This test environment consists of two DCAN500 evaluation boards (EVB), two EVB Tester boards (optional), DC powerline Attenuator (optional), and a user's ECUs.

Users can interface directly from a CAN controller with the DCAN500. At the transmitting side, the ECU generates CAN test messages. At the receiver side, the DCAN500 EVB receives the test message, transfer it to the other ECU, and analyze the received messages for errors.

Adding the two EVB Tester boards, allows also the stand-alone mode (auto TX-RX test), on-board DCAN500 configuration switches, and onboard CAN transceiver which can interface to an existing CAN-BUS.

See EVB Tester board manual for more details.

The DC-powerline attenuator is used to test the communication in variable attenuation levels (0-61dB), emulating a DC powerline environment. When powering the ECU directly to the powerline, it is recommended to add an inductor (> 22uH) in serial to its power supply to avoid strong attenuation due to the power supply input filtering capacitors.

ANNEX A

This Annex describe how to operate the DCAN500 with 1Mbit/s CAN bitrate using DCAN500 bitrate configuration registers.

10. DCAN500 Bitrate Configuration Registers

Each of these registers is accessible by the ECU for *Read* and *Write* operations. The access method to these registers is described in section 5.12. This section elaborates on the registers and their default values after power-up/reset.

Table 18 – Advanced Registers summary table

Register name	Addr.	Description
REG_9 - 'CAN bit time conf_0'	0x09	CAN bit time SEG 1 lower nibble
REG_B - 'CAN bit time conf_1'	0x0B	CAN bit time SEG 1 higher nibbles
REG_C - 'CAN bit time conf_2'	0x0C	CAN bit time SEG 2 lower nibble
REG_E - 'CAN bit time conf_3'	0x0E	CAN bit time SEG 2 higher nibbles

10.1   REG_9 – 'CAN bit time conf_0' (Address 0x09)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
CAN bit time SEG 1 lower nibble[7:0]							

Bits [7:0] -   CAN bit time SEG 1 lower nibble. Default configuration is according to BR_SEL[1:0] bits (see 3.3).
R   - Readable bit,   W - Writeable bit   [x] - Value on power-up. '1' - bit is set; '0' - bit is cleared

10.2   REG_B – 'CAN bit time conf_1' (Address 0x0B)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	CAN bit time SEG 1 higher nibble[3:0]			

Bits [3:0] -   CAN bit time SEG 1 higher nibble. Default configuration is according to BR_SEL[1:0] bits (see 3.3).
Bits [7:4] -   '0000'
R   - Readable bit,   W - Writeable bit   [x] - Value on power-up. '1' - bit is set; '0' - bit is cleared

10.3   REG_C – 'CAN bit time conf_2' (Address 0x0C)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
CAN bit time SEG 2 lower nibble[7:0]							

Bits [7:0] -   CAN bit time SEG 2 lower nibble. Default configuration is according to BR_SEL[1:0] bits (see 3.3).
R   - Readable bit,   W - Writeable bit   [x] - Value on power-up. '1' - bit is set; '0' - bit is cleared

10.4   REG_E – 'CAN bit time conf_3' (Address 0x0E)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	CAN bit time SEG 2 higher nibble[3:0]			

Bits [3:0] -   CAN bit time SEG 2 higher nibble. Default configuration is according to BR_SEL[1:0] bits (see 3.3).
Bits [7:4] -   '0000'
R   - Readable bit,   W - Writeable bit   [x] - Value on power-up. '1' - bit is set; '0' - bit is cleared

11.   Device Configuration

11.1   Bit rate configuration

The host can configure the bitrate using two configuration methods;

11.1.1   Bit rate select hardware pins

The selected CAN bitrate is configured through dedicated bitrate select hardware pins.
The DCAN500 samples these pins once upon reset or power cycle event. The configured bit rate remains until the next reset or power cycle event.

Table 19 - CAN bit rate selection using HW pins

Bitrate_sel [1:0]	Bitrate [Kbps]	Sample point
00	83.3	80%
01	125	80%
10	250	80%
11	500	80%

11.1.2   Set CAN BIT internal control registers in data command mode

The DCAN500 bit time combined from two segments:
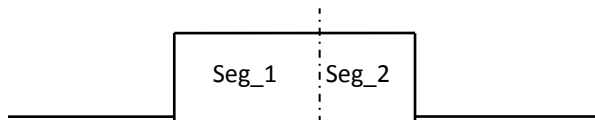
Where: DCAN500_bit_time = seg_1 + seg_2

The sample point defined as the point of time between seg_1 and seg_2.
The DCAN500 defines the bit time and those two segments using four internal registers that can be configured in data command mode.

Seg_1 = {REG_B[3:0],REG_9[7:0]}
Seg_2 = {REG_E[3:0],REG_C[7:0]}

Where:

Seg_1 + seg_2 = Num_of_samples = $\dfrac{\left(\frac{1}{bit\ rate}\right)}{12.5n}$

Example 1
Host CAN Bitrate = 83.3kbit/s
Sample point = 80%

Bit time [samples] = Seg_1 + seg_2 = $\dfrac{\left(\frac{1}{83.3k}\right)}{12.5n}$ = 960.

For 80% sample point:
*Seg_1 = 960 * 0.8 = 768 = 0x300*
Seg_2 = 960 - seg_1 = 192 = 0x0C0

Configuration will be:
REG_9 = 0x00
REG_B = 0x03
REG_C = 0xC0
REG_E = 0x00

Example 2
Host CAN Bitrate = 500kbit/s
Sample point = 85%

Bit time [samples] = Seg_1 + seg_2 = $\frac{\left(\frac{1}{500k}\right)}{12.5n}$ = 160

For 85% sample point:
*Seg_1 = 160 * 0.85 = 136 = 0x088*
Seg_2 = 160 - seg_1 = 24 = 0x018

Configuration will be:
REG_9 = 0x88
REG_B = 0x00
REG_C = 0x18
REG_E = 0x00

Example 3
Host CAN Bitrate = 1Mbit/s
Sample point = 80%

Bit time [samples] = Seg_1 + seg_2 = $\frac{\left(\frac{1}{1000k}\right)}{12.5n}$ = 80

For 80% sample point:
*Seg_1 = 80 * 0.8 = 64 = 0x040*
Seg_2 = 80 - seg_1 = 16 = 0x010

Seg_1 = {REG_B[3:0],REG_9[7:0]}
Seg_2 = {REG_E[3:0],REG_C[7:0]}

Configuration will be:
REG_9 = 0x40
REG_B = 0x00
REG_C = 0x10
REG_E = 0x00

11.1.3 DCAN500 Minimal message data bytes (DLC) requirements

Table 20 describes the minimal required data bytes to be sent per CAN message according to bitrate selection pair

Table 20 – Minimal data bytes requirements per CAN message bitrate selection

CAN bitrate	Min. DLC
83.3Kbit/s	---
125Kbit/s	---
250Kbit/s	---
500Kbit/s	3
1000Kbit/s	7

12. DCAN500 CAN frame Latency

The DCAN500 is a powerline CAN gateway device. Each CAN message transferred by the ECU (*CAN-Message*) is constructed into a single DCAN500 CAN frame transmitted over a powerline (*CAN-Frame*).

A *CAN-Frame* is constructed from a *Start-frame* consisting of arbitration and preamble patterns followed by packet/s of data bytes (at least 1 packet), and terminated with a *Frame-End,* indicating the last packet of the frame. An Error Correction Code [ECC] protects each data packet.

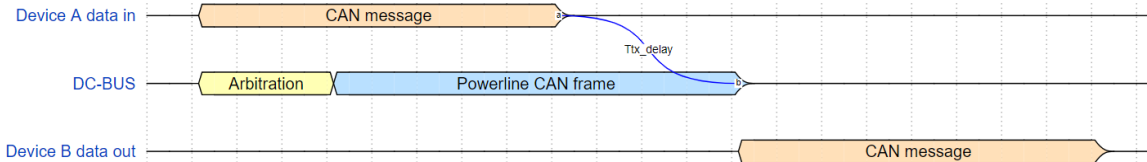A *CAN-Frame* may be longer than *CAN-Message* as depicted in Figure 12.

Figure 22 – Typical TX-RX CAN message flow

The overall delay (latency) from TX side CAN message SOF to RX side CAN message is subject to the powerline CAN frame length.

For example, Figure 23 depicts the transmission of CAN message @ 1Mbit/s with DLC =8, which results in fixed latency of ~340 µsec.
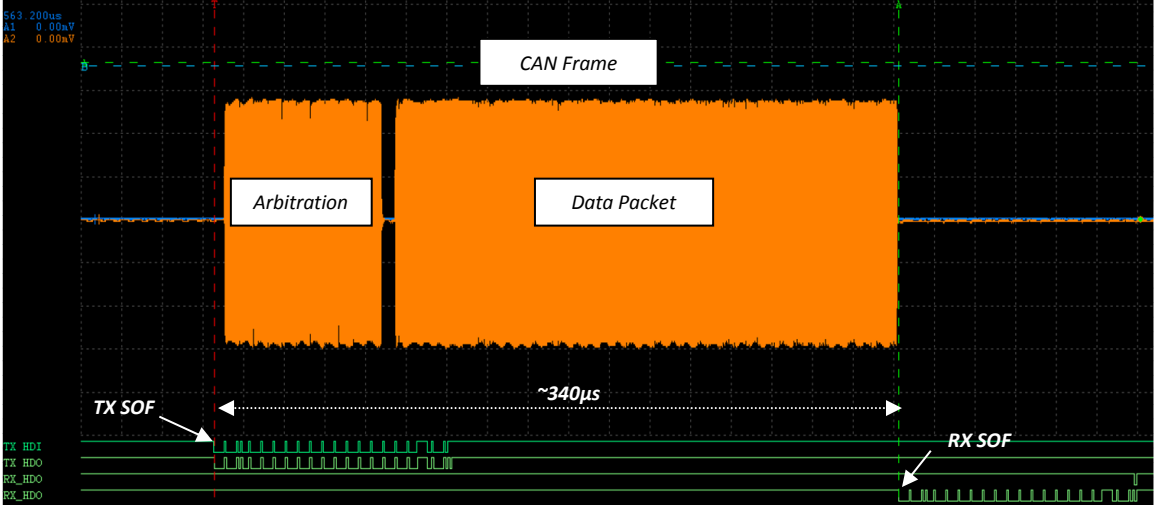

Figure 23 – Example of CAN A TX, DLC =8 @ 1Mbit/s CAN bitrate

Revision History

Rev.	Date	Description
0.72	10/05/2019	Initial preliminary revision.
0.73	22/7/2019	Updated 5MHz filter.
0.74	2/8/2019	Rearrange paragraphs.
0.75	05/09/2019	Update schematic, Table 16.
0.76	23/09/2019	Editing.
0.77	02/10/2019	Update Table 2 and Figure 7.
0.78	14/11/2019	Update Figure 4 and NSLEEP pin description.
0.79	19/01/2020	Update Table 2.
0.80	18/02/2020	Update clause 2.5.3.1. Add UUID clause 3.6. Update clause 5 with UUID REGs.
0.81	01/08/2020	Update 2.5.3.1, 2.5.5, and Table 16.
0.82	11/01/2021	Editing.
0.83	16/01/2022	Update Table 2.
0.84	18/04/23	Update DCAN500 interface with ECU bitrate of 1Mbit/s option (see 10)