Application Note

How Selective Wake Enables Partial Networking

Richard Hubbard

Transceiver Interface

ABSTRACT

This application note discusses selective wake-up as per ISO 11898-2:2016 and how it enables partial networking (PN). The application note includes how PN is used, how to configure TI devices, and two set-up examples.

Table of Contents

List of Figures

List of Tables

Trademarks

All trademarks are the property of their respective owners.

1 Introduction – Partial Networking

Controller Area Network (CAN) is a communication architecture used for reliable communication in environments that are considered to be noisy, one example being automobiles. Over time, CAN has evolved to include classic high-speed CAN as well as CAN flexible data rate (CAN FD). Classic high-speed CAN supports data rates up to 1 Mbps and although CAN FD is backwards compatible with classic high-speed CAN, it also supports data rates up to 5 Mbps and higher. CAN FD supports up to 64 bytes and a bit rate switching mode, which allows faster than 1 Mbps data transmission.

In an automotive CAN network, power is always a concern for situations when the vehicle is off or when only a few electrical control units (ECUs) need to be on during normal operation. The development of PN addresses this concern. PN is a method used to save power when both homogeneous (all ECUs capable of PN) or mixed (not all ECUs capable of PN) CAN and CAN FD networks are used. This process is accomplished by allowing bus communication, but sleeping nodes only wake on a specific CAN message or frame. PN was developed by the International Standard Organization (ISO) and is based upon the latest version of the ISO 11898-2:2016 standard.

This application note provides information using an example application. The application note focuses on what selective wake, also known as partial networking (PN), is and how it is used in an application. The register set from the TCAN1145-Q1 and TCAN1146-Q1 is provided for ease of understanding and finishes with two examples on how to program a device for PN.

2 Partial Networking Application

Figure 2-1 represents a simplified mixed CAN FD network. ECU3 and ECU6 are partial networking capable ECUs. The rest of the ECUs are not PN capable. ECU1 and ECU8 represent the termination locations. CAN bus represents the CANH and CANL signals. PN enabled ECUs described in the following sections relate to ECU3 and ECU6 in this figure.

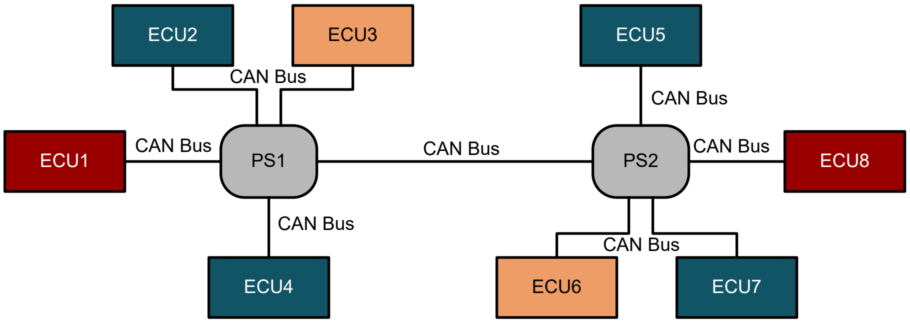


Figure 2-1. Simplified CAN FD Network

To understand partial networking in an application the PN capable CAN transceiver needs to be understood. Modes of operation and the impact of partial networking will be presented, followed by a deeper investigation of sleep mode and how utilizing PN creates two different levels of sleep. To understand how PN works, a brief explanation on a CAN frame is provided which will be followed up with a more detailed explanation. Lastly, information will be provided on how a mixed network similar to the simplified one provided will behave.

2.1 Modes of Operation and Partial Networking

CAN has three main modes of operation that have a direct impact on the system power: normal mode, standby mode, and sleep mode. Normal mode is the highest power mode where CAN or CAN FD communication takes place. Standby mode is the transitional mode from sleep, where the ECU wakes up from sleep and the transceiver waits for the ECU system processor to place the CAN transceiver into normal mode. It's important to note that standby and sleep modes do not allow data from the CAN network to be passed through to the CAN transceiver RXD pin. Lastly, sleep mode is the lowest power mode where only the CAN transceiver's low-power receiver is active in the ECU.

When the ECUs are placed into sleep mode (low-power mode), a wake-up pattern (WUP) starts the wake up process for every node on the network. WUP is a filtered dominant pulse, followed by a filtered recessive pulse, and then by another filtered dominant pulse. WUP acts as a basic noise filtering mechanism that differentiates CAN noise from actual CAN traffic. When a CAN frame is sent over CAN, every node recognizes a WUP and transitions out of sleep mode to a higher power mode such as standby or normal modes and all CAN nodes consume current at higher levels.

When an ECU supports PN, an additional step in the wake process is required to leave sleep mode. A CAN node that supports partial networking breaks sleep mode into two states, deep sleep (CAN bus GND bias) and sleep (CAN bus 2.5 V bias). While the PN-enabled CAN transceiver is in either of these sleep states, no CAN frames are passed through to the RXD pin.

2.2 Sleep Mode and Partial Networking

CAN ECUs are generally seen in three states: sleep mode (low current consumption, ≤ 100 µA), standby mode (higher current consumption, 10s to 100s mA), and normal mode (highest current consumption, 100s mA to several amps). Since each node uses significantly less current when it is asleep, the longer ECU is in sleep mode, the more power that can be saved. PN breaks sleep mode into two different levels, see Figure 2-2, allowing overall network current consumption to be reduced. When an ECU is in sleep mode, it is waiting to wake up, which can be accomplished by a WUP signal.

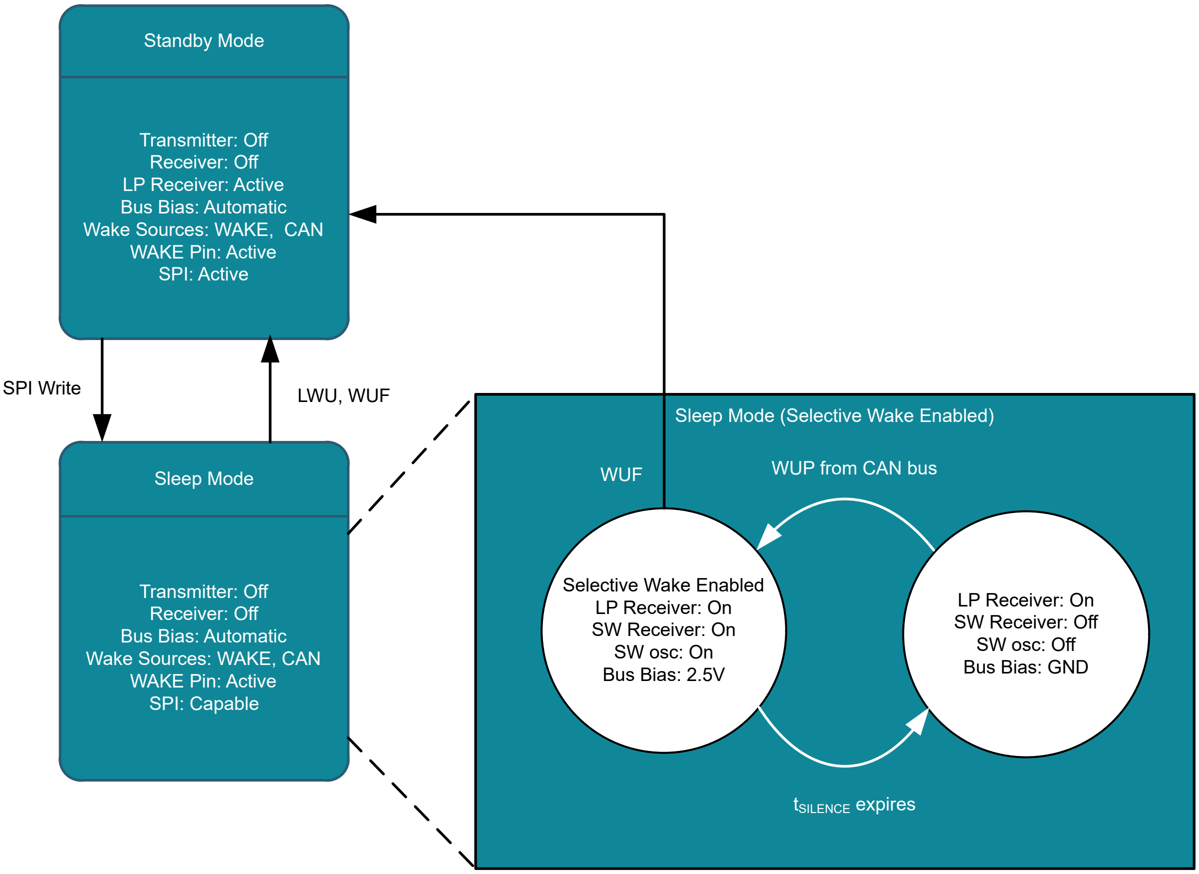


Figure 2-2. Partial Network in Sleep Mode

When an ECU receives a WUP, non-PN capable ECUs will wake up and transition to standby mode, whereas PN enabled ECUs will transitions to a higher sleep state, but not into standby mode. In this higher sleep state, the CAN bus bias changes from ground to 2.5 V while listening for a valid wake-up frame (WUF). This bias change is necessary for the WUF receiver to decode the CAN frames and no bus traffic is passed through to the CAN transceiver RXD pin. While the ECU is in this higher sleep state waiting for a valid WUF, only a subset of the node is awake and is consuming ~ 500 μA and not the 10s to 100s of mA that a non-PN node would be consuming.

2.3 Wake-Up Frame

Before the enabled PN CAN transceiver correctly receives and interprets the Wake-Up Frame (WUF), the receiver must synchronize to the CAN data rate. To save cost, the local PN transceiver uses an internal oscillator that can have a variance of ± 3% compared to the CAN controllers on the network. To properly decode CAN frames, the device must first synchronize its internal clock to the clock of the controllers sending CAN frames. This is accomplished by analyzing CAN frames and comparing the length of received bits versus its own expected length, and then making bit length adjustments. This can take several CAN frames before the PN CAN transceiver locking step is complete.

The ISO 11898-2:2016 CAN standard allows up to four CAN frames at 500 kbps or 8 CAN frames at 1 Mbps for the transceiver to lock on to the data rate, not including the first CAN frame that contains the WUP, which starts after t_{Bias}. Correct decoding of these frames is not required, and does not cause the internal frame error counter to increment. The transceiver can receive a valid WUF after it successfully syncs to the CAN data rate. See Section 3 for clarification on the wake-up frame.

During this process, CAN bus data does not get passed through to the transceiver RXD pin. Once the PN-enabled ECU receives this valid WUF, it fully wakes up and transitions to standby mode. Other nodes that did not wake from this WUF must continue to remain in Sleep mode, decoding CAN frames and looking for their matching WUF pattern. If the transceiver does not receive a valid WUF, and the ISO defined $t_{SILENCE}$ timer expires (timeout for bus inactivity), the ECU will re-enter deep sleep and bias the bus back to ground. Keep in mind that if other nodes are communicating, the CAN network will stay biased to 2.5 V and will no longer be in the deep sleep state but is still consuming less current than ECUs in standby or normal mode.

2.4 Classical High-speed CAN, CAN FD, and PN

The ISO 11898-2:2016 standard covers the physical layer for high-speed CAN, CAN FD and PN. PN uses classical high-speed CAN for waking up an ECU but does not preclude the CAN network from using CAN FD during normal CAN traffic. CAN FD communication would typically generate a CAN error when used on a classic CAN node; the standard overcame this by using the CAN FD Format indicator (FDF) bit in the CAN packet. The FDF bit set to 1b indicates that the packet is CAN FD. This bit allows a PN-enabled CAN transceiver to determine if the frames are CAN FD.

PN-capable CAN transceivers have the ability to decide whether to indicate an error or to ignore the CAN FD frames by programming a SW_FD_PASSIVE bit. When this bit is set to 1b, CAN FD frames are ignored, which is known as CAN FD passive.

The benefit for using classical high-speed CAN for WUF is that a PN-enabled CAN transceiver does not need a precise internal clock. If a CAN FD frame was to be used, the CAN transceiver needs a 0.5% tolerant crystal to decode the WUF which adds system level cost.

2.5 Mixed Network Information

Since partial networking Wake Up Frames (WUF) are only recognized as part of a Classical CAN frame, it is not necessary to fully decode CAN FD frames. Also, the faster data rates and switching of data rates within a CAN FD frame, can prove to be too difficult to decode without using a highly accurate external clock source. However, if a CAN frame is not decoded correctly, it will accumulate an error counter that can overflow, and cause an unwanted wake-up. To avoid this problem of CAN FD frames leading to unwanted decode errors, an FD Passive mode is provided.

This key feature of Partial networking capable CAN transceivers is that they can operate seamlessly within a network of CAN FD transceivers when FD Passive mode is enabled by setting the SW_FD_PASSIVE bit, and by providing the data rate ratio of CAN FD:CAN with the FD_DR setting. When CAN FD frames are detected, as indicated by FDF=1, the rest of the frame can be safely ignored so as to not cause an increment to the error counter. CAN bus noise that is smaller than CAN FD bit times will be filtered out, while active FD data can be detected, so that the end of the CAN FD frame can be properly recognized. In this way, a WUF based on Classical CAN may appear at any time and adjacent to CAN-FD frames.

CAN FD passive is only enabled in sleep mode so PN enabled transceivers will ignore CAN FD frames while the transceiver is in sleep mode. These PN enabled transceivers will wake when a valid WUP followed by a valid WUF is received and then will transition to standby mode. The ECU controller changes the PN enabled CAN transceiver to normal mode which allows CAN FD frames to be transmitted and received. The CAN FD PN passive mode should not be disabled. When the go-to-sleep command is issued, the PN enabled transceiver can receive a WUF while transitioning from normal to sleep mode, allowing it to decode the WUF correctly and wake up if valid. If these are disabled, they would have to be re-enabled before placing the PN capable transceiver into sleep mode. What could cause PN to be disabled?

In a typical application, PN would be enabled at power-up and left on to cover the mode transition cases mentioned as well as the transition from normal to sleep mode. If a WUP takes place during this transition, the device already has PN enabled and will be looking for a valid WUF. For the case that sleep mode is entered due to a fault, PN can end up disabled. In this case a WUP will wake up the device even though PN is still configured (SWCFG=1) but it is off due to SW_EN=0. Only the SW_EN bit has to be reset to 1 to completely re-enable PN.

As an example, if a UVIO fault or TSD fault occurs and Failsafe mode is disabled, the device will enter Sleep mode and set SW_EN=0. Since PN is now disabled, a WUP will act as a wake event and transition the device to Standby. As another example, in sleep mode, decoding errors that lead to FRAME_OVF error would set SW_EN=0 and cause the device to transition to standby mode. So, after a fault and before placing the PN capable transceiver to sleep the SW_EN should be set to 1.

3 CAN Frames and Wake-up Frames

To comprehend how selective wake works, a basic understanding of the CAN and CAN FD frame needs to be understood.

3.1 CAN Frame Structure

To understand how Wake Up Frames (WUF) are used in CAN partial networking, it is important to first understand the structure of a CAN frame. In Figure 3-1, a high signal represents the CAN bus is recessive (logic 1), while a low signal represents the CAN bus is dominant (logic 0).

There are two formats used which are based upon the number of ID bits. As shown in Figure 3-1 there is an 11-bit ID format known as Classic Base Frame Format (CBFF) and a 29-bit ID format know as Classic Extended Frame Format (CEFF). The following is a list of the names and description:

- IDLE – When the CAN bus is in an idle state, the bus will be recessive, or ‘1.’
- SOF – Start of Frame, a dominant bit, or ‘0.’
- ID[10:0], the Base ID in CBFF, or EXT_ID[28:18], the first 11 bits of the Extended ID in CEFF. These share the same configuration bits.
- EXT_ID[17:0], the remaining 18 bits of the Extended ID in CEFF
- RTR – Remote Transmission Request. A ‘1’ indicates a remote frame (will have a zero length DLC). A ‘0’ indicates a data frame. For a WUF, this bit must be ‘0.’
- IDE – Identifier. A ‘1’ indicates CEFF (extended ID), while a ‘0’ indicates CBFF (base ID)
- FDF – FD Format indicator which states whether the frame is classic CAN,(0), or CAN FD (1). For a WUF, this bit must be ‘0.’ Note that for CAN-FD frames, the structure following this FDF bit is different from either Classical CAN frame, and is not described here.
- DLC – Four-bit Data Length Code states how many data bytes are in the frame. Values of 8-15 will be interpreted as 8 bytes
- SRR – Substitute RTR. Will always be a ‘1.’
- r1 – Reserved bit. Will always be a ‘0.’
- CRC – 15-bit Cyclic Redundancy Check which is used to determine the integrity of the information
- Del – Delimiter bits. Will always be ‘1.’
- ACK – Acknowledge. The sender of the CAN frame will transmit a ‘1,’ while all of the receivers of the CAN frame will transmit a ‘0’ if the CRC is correct, or a ‘1’ if the CRC is incorrect

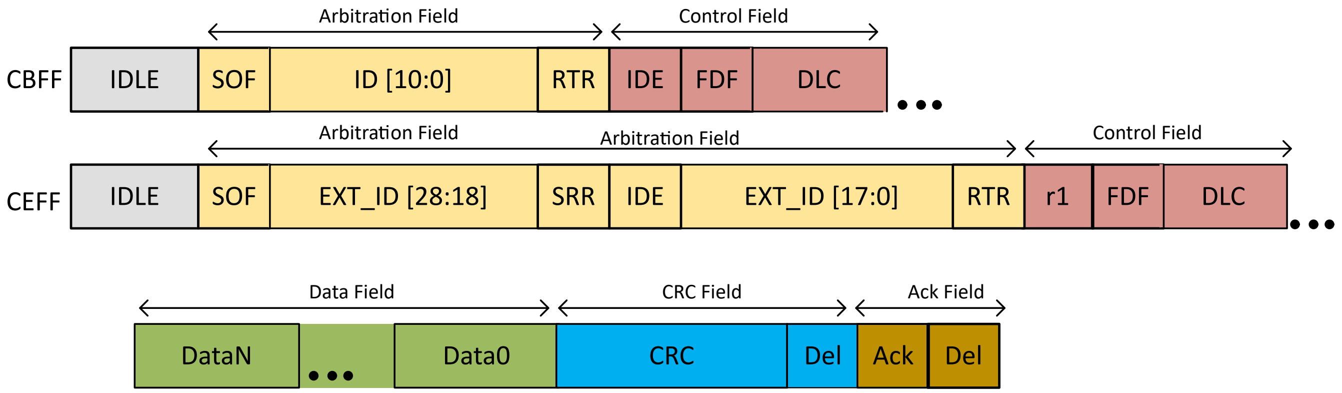


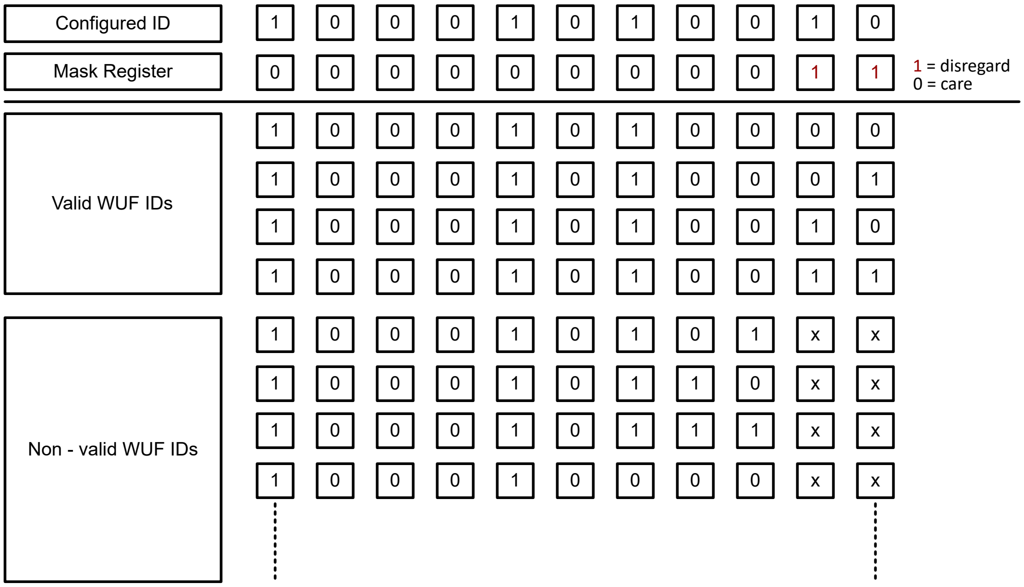
Figure 3-1. CAN Frame Structure

3.2 Wake-up Frames

The ISO 11898-2 standard allows several configuration options for a CAN frame to be identified by the receiving node as a Wake-Up Frame (WUF). Several components of the CAN frame must meet the WUF criteria that is provisioned into the receiving node.

3.2.1 ID Field Match

The first criteria for a WUF is that the ID Field must match. This can be either an 11-bit Base ID for CBFF, or the 29-bit Extended ID for CEFF. A configuration register (IDE) will define which one to use (0=Base ID, 1=Extended ID). A matching ID does not have to match every bit of the programmed ID field. Some bits may be masked, so that either a 0 or a 1 in that bit position is considered a match. For instance, the following 11-bit Base ID, the ID Mask field, and the received Base ID. In the ID Mask field a 0 means the value must match and a 1 means the value is disregarded.



Configured ID	1	0	0	0	1	0	1	0	0	1	0	
Mask Register	0	0	0	0	0	0	0	0	0	1	1	1 = disregard 0 = care
Valid WUF IDs	1	0	0	0	1	0	1	0	0	0	0	
	1	0	0	0	1	0	1	0	0	0	1	
	1	0	0	0	1	0	1	0	0	1	0	
	1	0	0	0	1	0	1	0	0	1	1	
Non - valid WUF IDs	1	0	0	0	1	0	1	0	1	x	x	
	1	0	0	0	1	0	1	1	0	x	x	
	1	0	0	0	1	0	1	1	1	x	x	
	1	0	0	0	1	0	0	0	0	x	x	

Figure 3-2. ID Field

In this example, the configured IDE bit would be 0 (11-bit ID field), the *Configured ID* is set by the register ID[10:0], and the *Mask Register* would be set by register ID_MASK[10:0]. In the ID_MASK field, a 1 is a mask, or disregard, and a 0 is a care. So, for the 11-bits, all *care* bits must match the programmed ID[n] bit, while all *disregard care* bits are accepted as a match. If all bits in the ID field are matching, then the ID filed is accepted as a WUF ID match.

If the configuration register DATA_MASK_EN=0, then only the ID field has to match. The rest of the CAN frame must still be decoded and have a correct CRC field, followed by a recessive bit (the CRC delimiter) to be considered a valid WUF.

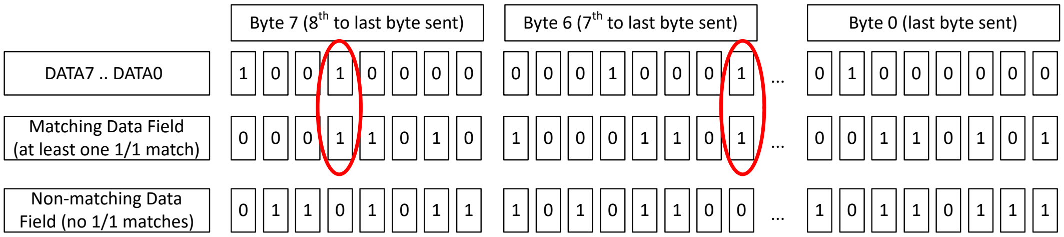
3.2.2 Data Length Code (DLC) Match

If DATA_MASK_EN=1, then the DLC and the Data Field must also match in order to be a valid WUF. The DLC field is a 4-bit value and can range from 0-15. This must exactly match the DLC field provisioned into the DLC register.

3.2.3 Data Match

The Data field match requires that at least one bit, located in any bit position and in any byte, must match as a 1 in both the received field and the register programmed field. If the DLC code is ‘b0000, such that there are no Data bytes, then the Data field match is considered a WUF Data match.

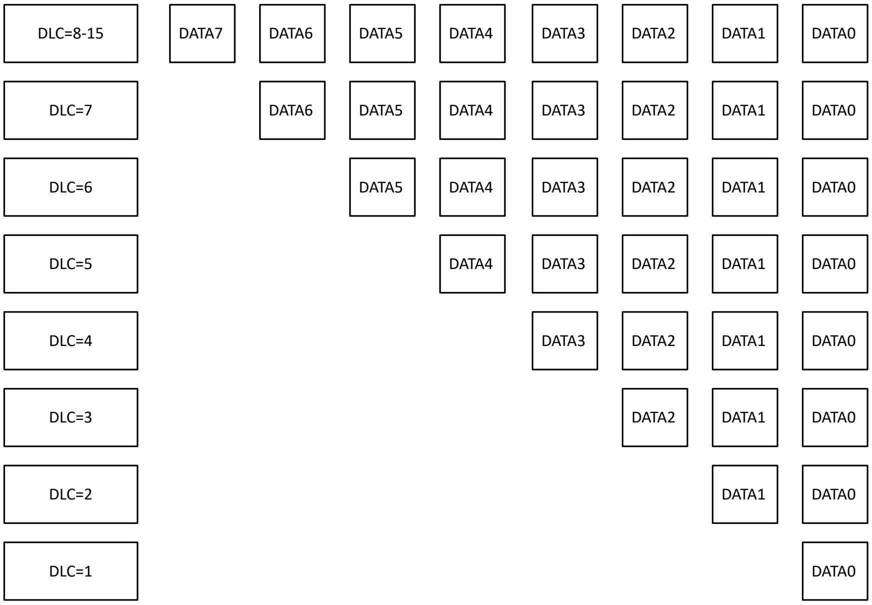
The order of the data bytes stored in the DATA7-DATA0 register fields are designated in a descending order, such that the last byte of the frame is always considered Byte 0, to correspond with the register DATA0.



	Byte 7 (8th to last byte sent)	Byte 6 (7th to last byte sent)		Byte 0 (last byte sent)
DATA7 .. DATA0	1 0 0 1 0 0 0 0	0 0 0 1 0 0 0 1	...	0 1 0 0 0 0 0 0
Matching Data Field (at least one 1/1 match)	0 0 0 1 1 0 1 0	1 0 0 0 1 1 0 1	...	0 0 1 1 0 1 0 1
Non-matching Data Field (no 1/1 matches)	0 1 1 0 1 0 1 1	1 0 1 0 1 1 0 0	...	1 0 1 1 0 1 1 1

Figure 3-3. Data Matching

For this example, with DLC=8, there are two bits that have matched a ‘1’ in both the received data byte and the configured DATAn byte. Since only one match is required, this is a valid WUF Data field.



DLC=8-15	DATA7	DATA6	DATA5	DATA4	DATA3	DATA2	DATA1	DATA0
DLC=7		DATA6	DATA5	DATA4	DATA3	DATA2	DATA1	DATA0
DLC=6			DATA5	DATA4	DATA3	DATA2	DATA1	DATA0
DLC=5				DATA4	DATA3	DATA2	DATA1	DATA0
DLC=4					DATA3	DATA2	DATA1	DATA0
DLC=3						DATA2	DATA1	DATA0
DLC=2							DATA1	DATA0
DLC=1								DATA0

Figure 3-4. DLC Value and Data Byte

Figure 3-4 shows the mapping of how data bytes are interpreted for different DLCs. The order of transmitted bytes is from left to right. For a DLC code of 4, for example, the incoming bytes received would be checked against the provisioned bytes in the order of Data3, Data2, Data1, and Data0.

3.2.4 CRC Match

As data is being received, a 15-bit CRC is being calculated for all bits, other than stuff bits, up to the CRC field. This calculated CRC is then compared against the CRC field given in the frame. If the CRC matches, and the bit following the CRC is a ‘1,’ then the CRC is considered to be valid.

3.2.5 Acknowledge Match

Finally, the last two bits checked for the WUF are part of the Acknowledge (Ack) Field. The value at the Ack is not checked, as all nodes could be in Sleep mode at this point and are thus unable send the Ack response of ‘0.’ The bit following Ack is the Ack Delimiter, and it must also be a ‘1’ to be considered a valid WUF.

3.3 Error Counter

PN capable transceivers have an error counter to track issues. The error counter will increment for the following reasons:

- If any CAN frame is not decoded properly, which can include a ‘1’ where a ‘0’ is expected, a ‘0’ where a ‘1’ is expected
- A run length of 6 consecutive bits of the same polarity occurs
- If the received CRC does not match the calculated CRC
- A CAN-FD frame is recognized and SW_FD_PASSIVE=0b

When the CAN frame is decoded properly, either WUF or non-WUF, the error counter will decrement by 1, down to 0. If this error counter exceeds a designated threshold (selected by the register FRAME_CNT_THRESHOLD, default of 31), an overflow will occur, the part will wake-up if in sleep and transition to standby mode. The device will set an interrupt FRAME_OVF. These errors indicate that CAN frames are not being decoded properly, so an actual WUF can not be decoded. As such, when the device wakes from an error counter overflow, selective wake is disabled (SW_EN=0). It is recommend to set SW_EN=1 again.

3.4 Selective Wake FD Passive

The TCAN114x-Q1 can be configured to either ignore CAN-FD frames (no change to the error counter) by setting SW_FD_PASSIVE=1 to enable FD Passive mode, or treat them as error frames (increment the error counter) by setting SW_FD_PASSIVE=0. Most applications will prefer to ignore CAN-FD frames and use the FD Passive feature. As the CAN-FD frame format is different than a Classical CAN frame format following the FDF bit, the internal frame decoder will ignore the remaining part of the CAN-FD frame by waiting for the next Bus Idle, indicated by a run of six recessive (1) bits.

4 Selective Wake Registers

This section provides the typical registers used for selective wake configuration and status. These are from the TCAN1145-Q1 and TCAN1146-Q1 data sheets.

Table 4-1 shows register address 10h: MODE_CNTRL. Register 8’h10[7] is the selective wake enable bit, SW_EN.

Table 4-1. MODE_CNTRL Address 10h

Bit	Field	Type	Reset	Description
7	SW_EN	R/W	0b	Selective wake enable for TCAN1145-Q1 and TCAN1146-Q1 otherwise reserved
6-0	N/A			0b = Disabled 1b = Enabled

Table 4-2 through Table 4-5 are registers 30h-33h:SW_ID1 – SW_ID4.

Table 4-2. SW_ID1 Register Address = 30h

Bit	Field	Type	Reset	Description
7-0	Ext_ID_17:10	R/W	0b	Extended ID bits 17:10

Table 4-3. SW_ID2 Register Address = 31h

Bit	Field	Type	Reset	Description
7-0	Ext_ID_9:2	R/W	0b	Extended ID bits 9:2

Table 4-4. SW_ID3 Register Address = 32h

Bit	Field	Type	Reset	Description
7-6	Ext_ID_1:0	R/W	0b	Extended ID bits 1:0
5	IDE	R/W	0b	Extended ID field 0b = Standard ID (11-bits) 1b = Extended ID (29-bits)
4-0	ID_10:6 EXT_ID_28:24	R/W	0b	ID [10:6] and Extended ID[28:24]

Table 4-5. SW_ID4 Register Address = 33h

Bit	Field	Type	Reset	Description
7-2	ID_5:0 EXT_ID_23:18	R/W	0b	ID [5:0] and Extended ID [23:18]
1-0	RESERVED	R	0b	Reserved

Table 4-6 through Table 4-10 are registers 34h-38h:SW_ID_MASK 1 – SW_ID_MASK4 and SW_ID_DLC_MASK.

- Used to set the mask ID for the devices you would like to communicate with
- Register 38h also contains DLC bits and DATA_MASK_EN

Table 4-6. SW_ID_MASK1 Register Address = 34h

Bit	Field	Type	Reset	Description
7-2	Reserved	R	0b	Reserved
1-0	EXT_ID_MASK_17:16	R/W	0b	Extended ID Mask 17:16

Table 4-7. SW_ID_MASK2 Register Address = 35h

Bit	Field	Type	Reset	Description
7-0	EXT_ID_MASK_15:8	R/W	0b	Extended ID Mask 15:8

Table 4-8. SW_ID_MASK3 Register Address = 36h

Bit	Field	Type	Reset	Description
7-0	EXT_ID_MASK_7:0	R/W	0b	Extended ID Mask 7:0

Table 4-9. SW_ID_MASK4 Register Address = 37h

Bit	Field	Type	Reset	Description
7-0	ID_MASK_10:3 EXT_ID _MASK_28:21	R/W	0b	ID Mask 10:3 and Extended ID Mask 28:21 (Base ID)

Table 4-10. SW_ID_MASK_DLC Register Address = 38h

Bit	Field	Type	Reset	Description
7-5	SW_ID_Mask_5	R/W	0b	ID Mask 2:0 and Extended ID Mask 20:18 (Base ID)
4-1	DLC	R/W	0b	DLC [3:0]
0	DATA_MASK_EN	R/W	0b	Data mask enable 0b = DLC field and Data field are not compared and assumed valid. Remote frames are allowed. 1b = DLC field must match DLC [3:0] register and data field bytes are compared with DATAx registers for a matching 1. Remote frames are ignored

Table 4-11 provides the registers that is used for the data. This is registers 39h-40h

Table 4-11. DATA_y Register Address = 39h + formula

Offset = 39h + (y × 1h); where y = 0h to 7h for TCAN1145-Q1 and TCAN1146-Q1

Bit	Field	Type	Reset	Description
7-0	DATAx	R/W	00h	CAN data byte x

Table 4-12 through Table 4-16 provides registers 44h – 47h:SW_CONFIG_1 – SW_CONFIG_4

- These registers are used to configure selective wake.

Table 4-12. SW_CONFIG_1 Register Address = 44h

Bit	Field	Type	Reset	Description
7	SW_FD_PASSIVE	R/W	0b	Selective Wake FD Passive: this bit modifies the behavior of the error counter when CAN with flexible data rate frames are seen. 0b = CAN with flexible data rate frame will be counted as an error frame 1b = CAN with flexible data rate frame are ignored (passive)
6-4	CAN_DR	R/W	101b	CAN bus data rate 0b = 50 kbps
				1b = 100 kbps 10b = 125 kbps 11b = 250 kbps 100b = Reserved
				101b = 500 kbps
				110b = Reserved
				111b = 1 Mbps
3-2	FD_DR	R/W	0b	CAN bus FD data rate ratio verses CAN data rate
				0b = CAN FD <= 4x CAN data rate
				1b = CAN FD => 5x and <= 10x CAN data rate
				10b = Reserved
				11b = Reserved
1-0	RESERVED	R	0b	Reserved

Table 4-13. SW_CONFIG_3 Register Address = 45h

Register 45h: SW_CONFIG_3 is the frame error counter

Bit	Field	Type	Reset	Description
7-0	FRAME_CNTx	RH	00h	Frame Error Counter: this error counter is incremented by 1 for every received frame error detected (stuff bit, CRC or CRC delimiter form error). The counter is decremented by 1 for every correctly received CAN frame assuming the counter is not zero. In case the device is set for passive on CAN with flexible data rate frames, any frame detected as a CAN FD frame will have no impact on the frame error counter (no increment or decrement). If the frame counter reaches FRAME_CNT_THRESHOLD [7:0] value the next increment will overflow the counter, setting FRAME_OVF flag. The counter is reset by the following: enabling the frame detection or t**SILENCE** detection.

Table 4-14. SW_CONFIG_3 Register Address = 46h

Register 46h: SW_CONFIG_3 sets the frame error counter threshold for register 45h

Bit	Field	Type	Reset	Description
7-0	FRAME_CNT_THRESHOLD	R/W	1 Fh	Frame Error Counter Threshold: these bits set the point at which the error counter reaches its maximum and on the next error frame will overflow and set the FRAME_OVF flag. Default is 31 so the 32nd error will set the overflow flag

Table 4-15. SW_CONFIG_4 Register Address = 47h

Register 47h: SW_CONFIG_4 is used to make sure CAN frames are being decoded correctly.

- Bit 7 can be written to, informing the device that all the selective wake registers have been configured.

Bit	Field	Type	Reset	Description
7	SWCFG	RH/W	0b	Select wake configuration complete 0b = SW registers not configured 1b = SW registers configured (make this the last step in configuring and turning on selective wake) NOTE: Writing to any of these wake configuration registers (30h - 44h, 46h) clears the SWCFG bit.
6	CAN_SYNC_FD	RH	0b	The device is properly decoding CAN FD frames if frame detection is enabled. This flag is updated after every received frame. By polling this flag, the system may determine if the device is properly decoding CAN FD frames, up to but not including the data field. This flag is self-clearing.
5	CAN_SYNC	RH	0b	Synchronized to CAN data: this flag indicates the device is properly decoding CAN frames if frame detection is enabled. This flag is updated after every received frame. By polling this flag, the system may determine if the device is properly decoding CAN frames. This flag is self-clearing.
4-0	RESERVED	R	0b	Reserved

Registers 51h and 53h: INT_1 and INT_3 are interrupt registers that contain interrupts when selective wake errors take place.

Table 4-16. INT_1 Register Address = 51h

Bit	Field	Type	Reset	Description
7-4	N/A			
3	FRAME_OVF	R/W1C	0b	Frame error counter overflow
2-0	N/A			

Table 4-17. NT_3 Register Address = 53h

Bit	Field	Type	Reset	Description
7	N/A			
6	SWERR	RH	0b	Logical OR of (SW_EN=1 and NOT(SWCFG)) and FRAME_OVF. Selective Wake may not be enabled while SWERR is set
5-0	N/A			

Registers 56h and 58h: INT_ENABLE_1 and INT_ENABLE_3 are interrupt mask registers that contain the interrupts mask if the interrupt is to be masked out

Table 4-18. INT_ ENABLE_1 Register Address = 56h

Bit	Field	Type	Reset	Description
7-4	N/A			
3	FRAME_OVF_ENABLE	R/W	1b	Frame error counter overflow enable
2-0				

Table 4-19. INT_ ENABLE _3 Register Address = 58h

Bit	Field	Type	Reset	Description
7	N/A			
6	SWERR_ENABLE	R/W	0b	Selective wake error enable
5-0	N/A	R/W	1b	

5 Configuring Partial Networking

Partial networking allows the user to configure the node to wake up to a specific CAN message ID, and data mask (if configured to check data). An example for both of these are provided.

5.1 Valid CAN Message ID Example

For this example, it is assumed that the WUF’s (Wake Up Frame) ID will be a standard ID (11-bit) of 0×123. The steps below outline the required procedure to configure the device for this behavior, and then put it to sleep. Note that when selective wake is enabled, only the specific or desired WUF can wake the device, unless there is some type of error threshold that is hit (such as a decoding error).

Table 5-1. Valid CAN Message ID Example

Parameter	Example Value
Desired WUF ID	Standard ID (11-bit) 0x123
ID Mask	0x000 (ID must match exactly, 1 = Disregard, 0 = must match)
Bus arbitration rate/CAN FD rate	500 kbps/2 Mbps
Ignore CAN FD	True
WUF Data Payload	Disregard

Breaking down the example parameter values, the WUF ID is 0×123. This is the specific ID that will need to match the received message’s ID to wake. The ID mask describes which bits in the ID must match. It is a disregard or must match bit. This means that when a bit in the ID mask is set to 1, then the specified bit will NOT be checked for a match. In order to exactly match a specific ID, the ID mask should be configured as 0s. The CAN FD bus in this example has an arbitration and data rate of 500 kbps and 2 Mbps respectively. Since this theoretical bus is a CAN FD bus, we want to allow regular CAN FD traffic to occur while some nodes are asleep. The SW_PASSIVE bit is used to describe how the partial networking IP handles reception of a CAN FD frame. The device can be configured to treat it like an error, and with enough errors the device will wake up and set an error flag.

This example has the ID Mask set to all 0s, so the ID must match exactly.

The device can also be configured to simply ignore any CAN FD frames, which does not affect waking the part up, or setting any errors. In this example, we do not care about the data payload. This means that any classic CAN message that has a standard ID of 0×123 and any payload will wake the device.

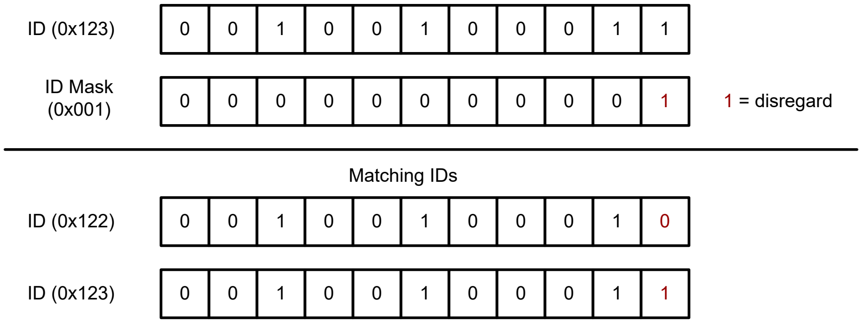

ID (0x123) 0 0 1 0 0 1 0 0 0 1 1
ID Mask (0x001) 0 0 0 0 0 0 0 0 0 0 1
1 = disregard
Matching IDs
ID (0x122) 0 0 1 0 0 1 0 0 0 1 0
ID (0x123) 0 0 1 0 0 1 0 0 0 1 1

Figure 5-1. Example 1: ID Matching

Note

The wake-up frame must be a classic CAN message (FDF bit = 0), but this does not require that the bus operate in only a classic CAN mode. CAN FD can be used for regular data transmission, but the WUF must be a classic CAN message.

It is critical to note that the order that registers are written to is important. Specifically, all of the selective wake registers should be configured before setting the selective wake enable bits. The reason is that any write to a selective wake configuration register will clear the SWCFG bit (0×47). If SWCFG is not set, then selective wake will NOT be enabled, even if the primary selective wake enable is set. It is best practice to ensure that this register is the last one written to during configuration. Table 5-2 shows every SPI transaction that should occur to configure the device for this example WUF configuration. The register address, hexadecimal data, and the raw hexadecimal of the SPI message is given.

Table 5-2. Valid CAN Message ID Programming

Step	Description	Register	Register (HEX)	Data (HEX)	Raw HEX
0	Disable selective wake	SW_CONFIG_4	0×47	0×00	0×8F, 0×00
1	Write desired WUF ID (0×123)	SW_ID1	0×30	0×00	0×61, 0×00
2	Write desired WUF ID (0×123)	SW_ID2	0×31	0×00	0×63, 0×00
3	Write desired WUF ID (0×123)	SW_ID3	0×32	0×04	0×65, 0×04
4	Write desired WUF ID (0×123)	SW_ID4	0×33	0×8C	0×67, 0×8C
5	Write ID mask to (0) SW_ID_MASK1	SW_ID_MASK1	0×34	0×00	0×69, 0×00
6	Write ID mask to (0) SW_ID_MASK2	SW_ID_MASK2	0×35	0×00	0×6B, 0×00
7	Write ID mask to (0) SW_ID_MASK3	SW_ID_MASK3	0×36	0×00	0×6D, 0×00
8	Write ID mask to (0) SW_ID_MASK4	SW_ID_MASK4	0×37	0×00	0×6F, 0×00
9	Write ID mask (0) and DATA_MASK_EN (0)	SW_ID_MASK_DLC	0×38	0×00	0×71, 0×00
10	Configure arbitration speed, FD:CAN ratio, and SW_FD_PASSIVE (1)	SW_CONFIG_1	0×44	0×D0	0×89, 0×D0
11	Set frame overflow to 255 (maximum number of errors before wake up)	SW_CONFIG_3	0×46	0×FE	0×8D, 0×FE
12	Set SWCFG bit (configuration is complete)	SW_CONFIG_4	0×47	0×80	0×8F,0×80
13	Enable selective wake and go to sleep	MODE_CNTRL	0×10	0×81	0×21, 0×81

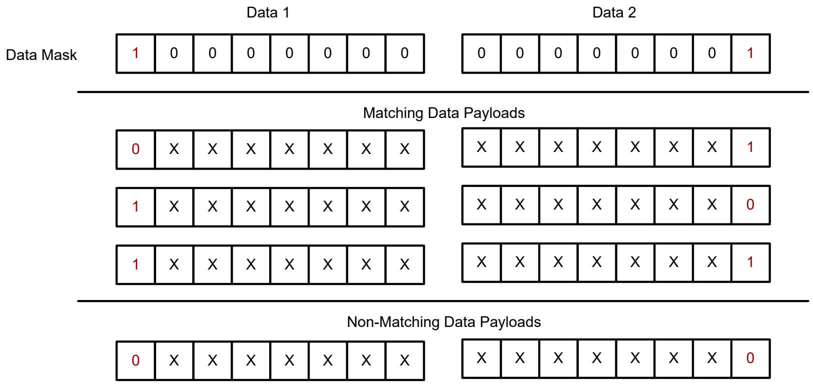
5.2 Valid Data Example

In this example, the CAN ID is less important as the ID mask bit 0 is marked as disregard. The data is what must be matched. The steps below outline the required procedure to configure the device for this behavior, and then transition the device to sleep. Note that when selective wake is enabled, only the specific or desired WUF can wake the device, unless there is some type of error threshold that is hit (such as a decoding error).

Table 5-3. Valid Data Example

Parameter	Example Value
Desired WUF ID	Standard ID (11-bit) 0x122/123
ID Mask	0×001 (Match all bits except bit 0, 1 = Disregard, 0 = must match)
Bus arbitration rate/CAN FD rate	500 kbps/2 Mbps
Ignore CAN FD	True
WUF DLC	2 Byte (DLC: 0×2)
WUF Data bits to wake up to	Byte 0[1], Byte 1[7]

Breaking down the example parameter values, the WUF ID is 0×123 or 0×122. The value written into the register does not matter which of the two is entered, since the ID mask will be used to mark bit 0 as disregard. By setting bit 0 of the ID mask to a 1, we tell the device to ignore this bit, which means it will accept a 0×122 or 0×123 ID. The CAN FD bus in this example has an arbitration and data rate of 500 kbps and 2 Mbps respectively. Since this theoretical bus is a CAN FD bus, we want to allow regular CAN FD traffic to occur while some nodes are asleep, so SW_PASSIVE is set to 1b.



	Data 1	Data 2
Data Mask	1 0 0 0 0 0 0 0	0 0 0 0 0 0 0 1
Matching Data Payloads	0 X X X X X X X	X X X X X X X 1
	1 X X X X X X X	X X X X X X X 0
	1 X X X X X X X	X X X X X X X 1
Non-Matching Data Payloads	0 X X X X X X X	X X X X X X X 0

Figure 5-2. Valid Data Example: Data Payload Verification

A change for this example is to do some verification against the data payload. Note that data payload checking will require an exact match for the DLC, but the way a WUF data bit is checked is bit-wise OR only. For example, the bits that are set to 1 are checked against the payload and will be considered a match if any single bit (or more) match. Likewise, if a WUF data mask is set to 0×FF, then any value with a bit equal to 1 will be matched. If a WUF data mask is set to 0×00, then that particular byte is essentially ignored, and cannot be matched, not even if the received data is 0×00. This OR is applied across all configured data bytes (set by the DLC field). Table 5-4 provides an example of how to program the device for this behavior.

Table 5-4. Valid Data Programming

Step	Description	Register	Register (Hex)	Data (Hex)	Raw Hex
0	Disable selective wake	SW_CONFIG_4	0×47	0×00	0×8F, 0×00
1	Write desired WUF ID (0×123)	SW_ID1	0×30	0×00	0×61, 0×00
2	Write desired WUF ID (0×123)	SW_ID2	0×31	0×00	0×63, 0×00
3	Write desired WUF ID (0×123)	SW_ID3	0×32	0×04	0×65, 0×04
4	Write desired WUF ID (0×123)	SW_ID4	0×33	0×8C	0×67, 0×8C
5	Write ID mask to (0×001) SW_ID_MASK1	SW_ID_MASK1	0×34	0×00	0×69, 0×00
6	Write ID mask to (0×001) SW_ID_MASK2	SW_ID_MASK2	0×35	0×00	0×6B, 0×00
7	Write ID mask to (0×001) SW_ID_MASK3	SW_ID_MASK3	0×36	0×00	0×6D, 0×00
8	Write ID mask to (0×001) SW_ID_MASK4	SW_ID_MASK4	0×37	0×00	0×6F, 0×00
9	Write ID mask (0×001), DLC (2), and DATA_MASK_EN (1)	SW_ID_MASK_DLC	0×38	0×25	0×71, 0×25
10	Write Data 0 (0×01)	DATA_0	0×39	0×01	0×73, 0×01
11	Write Data 1 (0×80)	DATA_1	0×40	0×80	0×75, 0×80
12	Configure arbitration speed, FD:CAN ratio, and SW_FD_PASSIVE (1)	SW_CONFIG_1	0×44	0×D0	0×89, 0×D0
13	Set frame overflow to 255 (maximum number of errors before wake up)	SW_CONFIG_3	0×46	0×FE	0×8D, 0×FE
14	Set SWCFG bit (configuration is complete)	SW_CONFIG_4	0×47	0×80	0×8F,0×80
15	Enable selective wake and go to sleep	MODE_CNTRL	0×10	0×81	0×21, 0×81

6 Summary

Partial networking is an extremely important function within CAN systems. It allows a reduction of power consumption, which has a direct effect on vehicle emissions. Since there are many applications for partial networking, it is important that any engineer working with CAN is aware of and understands partial networking. This application report serves as an informative guide to any engineer learning partial networking and implementing it using the TCAN1145-Q1 and TCAN1146-Q1.

The benefit of partial networking is to reduce the overall power consumption in the network and is ideal for mixed networks. Because each ECU uses less power while in sleep mode, it benefits the network to only enable needed nodes while the rest are in a low power sleep mode. This is especially important in an automotive setting. Internal combustion engine vehicles use an alternator to recharge their battery and if the vehicle is drawing more power than the alternator can supply, the battery is discharged. This is important when the engine is not running, because sleep mode limits the current and reduces the drain on the battery. The reduced power consumption due to partial networking can have a direct correlation to emissions in vehicles and by reducing the current draw of a car.

7 References

- Texas Instruments, TCAN114x-Q1 Enhanced CAN FD Transceiver with Partial Networking data sheet.

8 Revision History

Changes from Revision A (April 2021) to Revision B (June 2022) **Page**

- Updated the publication to include information about CAN Frame structure and what each bit represents 2
- Updated the publication to include clarification on partial networking and how it works, clarification on how the SW Mask work, error counter, and the various acceptable method for a wake-up frame 2
- Updated to add information on how CAN FD passive works 2
- Added two examples 2

IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATA SHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES “AS IS” AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, regulatory or other requirements.

These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you will fully indemnify TI and its representatives against, any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

TI’s products are provided subject to TI’s Terms of Sale or other applicable terms available either on ti.com or provided in conjunction with such TI products. TI’s provision of these resources does not expand or otherwise alter TI’s applicable warranties or warranty disclaimers for TI products.

TI objects to and rejects any additional or different terms you may have proposed.

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265