I²C to CAN Bus Bridge IC

HT42B216-x

Revision: V1.00 Date: August 02, 2024

www.holtek.com

Table of Contents

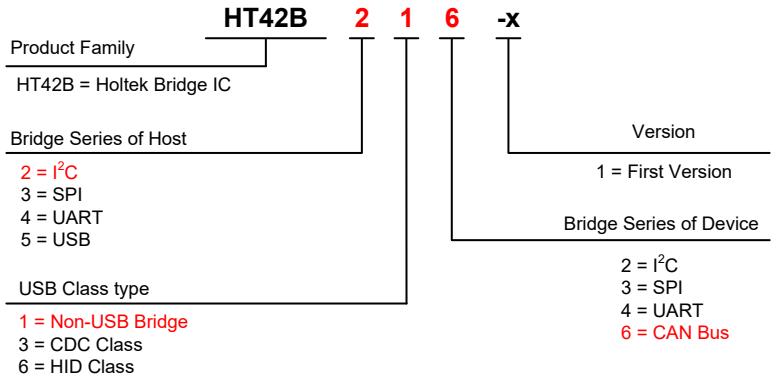
Features

- Operating voltage (V_{DD}): 3.0V~5.5V
- Power down and wake-up functions to reduce power consumption
- Fully integrated 8MHz internal oscillator requires no external components
- Serial Peripheral Interface – I²C
  - Provide Slave mode
  - Address can be set by IA0/IA1
  - I²C clock up to 400kHz
- CAN Bus Controller
  - Compatible with ISO11898-1
  - Support both formats CAN 2.0A and CAN 2.0B
  - Support the bit rates ranging from 5kbps to 1Mbps
  - Payloads up to 8 bytes per CAN frame
- Package type: 16-pin NSOP

General Description

The HT42B216-x is an I²C to CAN Bus Bridge controller with fully integrated I²C and CAN Bus interface functions, which can implement communication and data conversion between I²C and CAN Bus. This enables the MCU to easily communicate with the CAN Bus using the I²C interface. At the same time, it also provides an easily extensible CAN interface to meet a wider range of application requirements.

HOLTEK Bridge IC Naming Rules



Selection Table

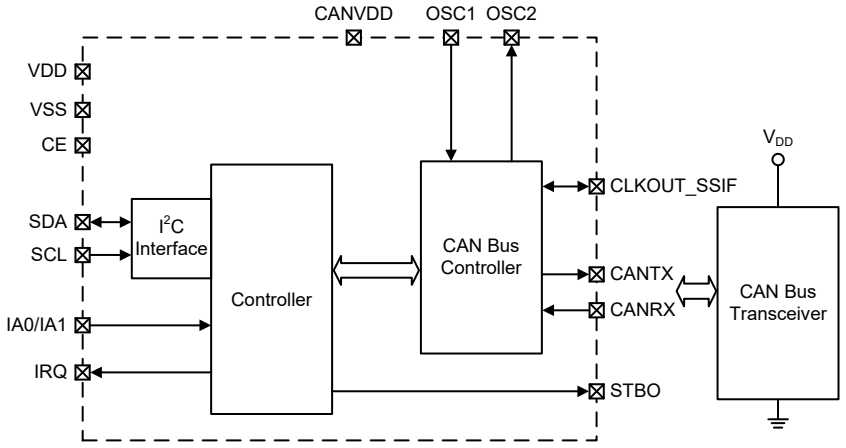
• **CAN Bus Bridge Selection Table**

Part No.	Description	V_{DD}	Interface Data Rate	FIFO/Buffer	CAN Bus Data Rate	Package
HT42B216-x	I²C to CAN Bus Bridge	3.0V~5.5V	Up to 400kHz	TX: 28 bytes RX: 28 bytes	Up to 1Mbps	16NSOP
HT42B316-x	SPI to CAN Bus Bridge		Up to 12MHz	TX: 28 bytes RX: 28 bytes		16NSOP
HT42B416-x	UART to CAN Bus Bridge		Up to 115.2kbps Baud	TX: 28 bytes RX: 28 bytes		16NSOP

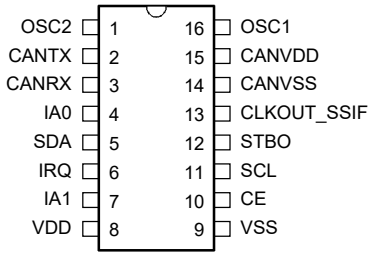
• **USB Bridge Selection Table**

Part No.	Description	V_{DD}	USB	Virtual COM	HID	FIFO/Buffer	Interface Data Rate	I/O V_{DD}	Package
HT42B536-x	USB to CAN Bus Bridge	3.3V~5.5V	Full-Speed	√	—	TX: 32 bytes RX: 64 bytes	CAN Bus Up to 1Mbps	—	16NSOP
HT42B532-x	USB to I²C Bridge			√	—	TX: 62 bytes RX: 62 bytes	Up to 400kHz	√	8SOP 10MSOP
HT42B533-x	USB to SPI Bridge			√	—	TX: 128 bytes RX: 128 bytes	Up to 8MHz	√	10MSOP 16NSOP
HT42B534-x	USB to UART Bridge			√	—	TX: 128 bytes RX: 128 bytes	Up to 3Mbps Baud	√	8SOP 10SOP/MSOP 16NSOP
HT42B564-x	USB (HID) to UART Bridge			—	√	TX: 32 bytes RX: 32 bytes	Up to 115.2kbps Baud	√	10SOP

Block Diagram



Pin Assignment



HT42B216-x
16 NSOP-A

Pin Description

Pin Name	Type	Description
SCL	I	I²C SCL line
SDA	I/O	I²C SDA line
CE	I	Chip Enable
IA0	I	I²C Address Select A0
IA1	I	I²C Address Select A1
IRQ	O	Interrupt Request Output IRQ pin output low, indicating there is data to be transmitted
STBO	O	Mode indication STBO pin output low, indicating normal mode STBO pin output high, indicating standby mode
CANTX	O	Transmit output pin to CAN bus
CANRX	I	Receive input pin from CAN bus
CLKOUT_SSIF	O	Clock output pin with CAN Bus CLK; it should connect a 510K resistor to ground
OSC1	I	CAN Bus Controller Oscillator input
OSC2	O	CAN Bus Controller Oscillator output
CANVDD	PWR	CAN Bus Controller positive power supply
CANVSS	PWR	CAN Bus Controller negative power supply
VDD	PWR	USB Bus positive power supply
VSS	PWR	Negative power supply, ground

Absolute Maximum Ratings

Supply Voltage ..... V_{SS}-0.3V to 6.0V

Input Voltage ..... V_{SS}-0.3V to V_{DD}+0.3V

Storage Temperature..... -60°C to 150°C

Operating Temperature..... -40°C to 105°C

I_{OL} Total 80mA

I_{OH} Total ..... -80mA

Total Power Dissipation 500mW

Note: These are stress ratings only. Stresses exceeding the range specified under "Absolute Maximum Ratings" may cause substantial damage to the device. Functional operation of the device at other conditions beyond those listed in the specification is not implied and prolonged exposure to extreme conditions may affect device reliability.

D.C. Characteristics

Ta=-45°C~105°C

Symbol	Parameter	Test Conditions		Min.	Typ.	Max.	Unit
		V_{DD}	Conditions				
V_{DD}	Operating Voltage	—	V_{DD}=CANV_{DD}	3.0	—	5.5	V
I_{DD}	Operating Current	5V	No load	—	5	12	mA
I_{STB}	Standby Mode	5V	Sleep mode, no load, CAN Bus sleep	—	5	10	μA
V_{IL}	Input Low Voltage	—	—	0	—	0.2V_{DD}	V
V_{IH}	Input High Voltage	—	—	0.8V_{DD}	—	V_{DD}	V
I_{OL}	Sink Current for I/O Ports	3.3V	V_{OL}=0.1V_{DD}	4	8	—	mA
		5V		10	20	—	mA
I_{OH}	Source Current for I/O Ports	3.3V	V_{OH}=0.9V_{DD}	-2	-4	—	mA
		5V		-5	-10	—	mA
I_{LEAK}	Input Leakage Current	3.3V	V_{IN}=V_{DD} or V_{IN}=V_{SS}	—	—	±1	μA
		5V		—	—	±1	μA

A.C. Characteristics

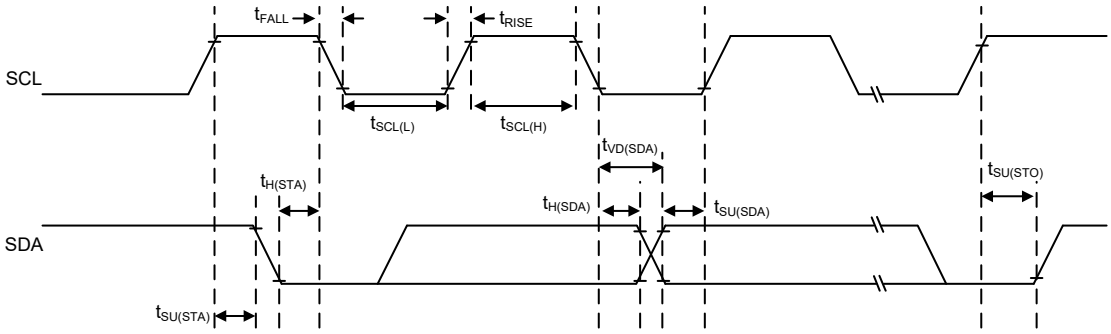
Ta=-45°C~105°C

Symbol	Parameter	Test Conditions		Min.	Typ.	Max.	Unit
		V_{DD}	Conditions				
f_{HIRC}	High Speed Internal RC Oscillator Frequency	3.3V~5.5V	—	-2%	8	+2%	MHz
f_{CAN}	CAN Bus System Clock (OSC)	3.3V~5.5V	—	—	16	—	MHz
t_{RSTD}	System Reset Delay Time	—	Power-on reset	25	50	100	ms
t_{CELOW}	Minimum Chip Enable Low Pulse Width	—	—	120	—	—	μs

I²C Electrical Characteristics

Ta=25°C

Symbol	Parameter	Test Conditions		Min.	Typ.	Max.	Unit
		V_{DD}	Conditions				
f_{SCL}	SCL Clock Frequency	3V/5V	—	—	—	400	kHz
$t_{SCL(H)}$	SCL Clock High Time	3V/5V	—	0.9	—	—	μs
$t_{SCL(L)}$	SCL Clock Low Time	3V/5V	—	0.9	—	—	μs
t_{FALL}	SCL and SDA Fall Time	3V/5V	—	—	—	0.34	μs
t_{RISE}	SCL and SDA Rise Time	3V/5V	—	—	—	0.34	μs
$t_{SU(SDA)}$	SDA Data Setup Time	3V/5V	—	0.1	—	—	μs
$t_{H(SDA)}$	SDA Data Hold Time	3V/5V	—	0.1	—	—	μs
$t_{VD(SDA)}$	SDA Data Valid Time	3V/5V	—	—	—	0.6	μs
$t_{SU(STA)}$	START Condition Setup Time	3V/5V	—	0.6	—	—	μs
$t_{H(STA)}$	START Condition Hold Time	3V/5V	—	0.6	—	—	μs
$t_{SU(STO)}$	STOP Condition Setup Time	3V/5V	—	0.6	—	—	μs

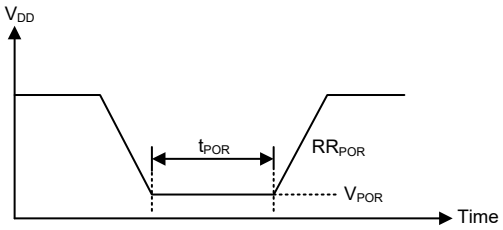


I²C Timing Diagram

Power-on Reset Characteristics

Ta=-45°C~105°C

Symbol	Parameter	Test Conditions		Min.	Typ.	Max.	Unit
		V_{DD}	Conditions				
V_{POR}	V_{DD} Start Voltage to Ensure Power-on Reset	—	—	—	—	100	mV
RR_{POR}	V_{DD} Rising Rate to Ensure Power-on Reset	—	—	0.035	—	—	V/ms
t_{POR}	Minimum Time for V_{DD} Stays at V_{POR} to Ensure Power-on Reset	—	—	1	—	—	ms



CE Pin Description

The HT42B216-x provides a CE pin, which is controlled by the Host MCU to enable the device. The HT42B216-x can operate normally when the CE is high, and the HT42B216-x is in the Reset state when the CE is low. The time duration for CE to be low should not be less than t_{CELOW}.

CAN Bus Interface

The HT42B216-x contains a CAN Bus control unit. For the connection to the physical layer additional transceiver hardware is required. Two pins of CANTX and CANRX interface to the CAN Bus Transceiver. The CAN Bus Controller supports the CAN 2.0 Part A and B protocol specifications and compatible with the ISO11898-1 standards. It is capable of transmitting and receiving standard and extended messages. It also capable of both acceptance filtering and message handler.

The CAN Bus control unit is connected to the external 16MHz high speed crystal oscillator via the OSC1 and OSC2 pins. In addition, the CLKOUT_SSIF pin needs to be connected to a 510K resistor to ground to ensure that the CAN Bus control unit operates normally.

I²C Interface

The I²C serial interface is a two-line interface, a serial data line, SDA, and serial clock line, SCL. It is a two-line low speed serial interface for synchronous serial data transfer. The I²C interface has the advantages of only two lines for communication, relatively simple communication protocol and the ability to accommodate multiple devices on the same bus.

The HT42B216-x integrated I²C interface contains the following features:

- Always as I²C slave
- There are four sets of addresses to select from through IA0 and IA1
- 28-byte FIFO receive buffer
- 28-byte FIFO transmission buffer
- SDA pin can wake up from Standby (MCU sends 1-byte I²C data)

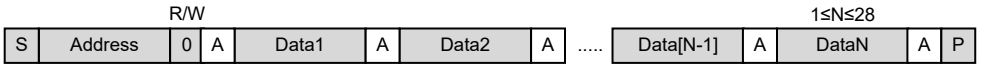
I²C slave address definition:

IA1 Pin Level	IA0 Pin Level	7-Bit I²C Slave Address (10101A_1A_0)
0	0	1010100
0	1	1010101
1	0	1010110
1	1	1010111

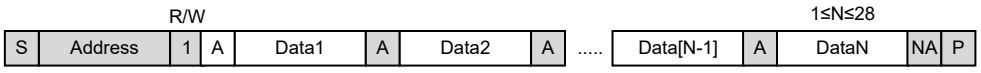
I²C Timing

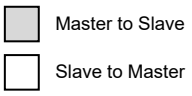
The following diagram shows the I²C transmission sequence. A single reception or transmission can be up to 28 bytes. The IRQ pin is provided to indicate whether there is data to be transmitted. The IRQ pin will be pulled high before Data[N-1] is sent to inform the master that this is the last byte.

Slave Receive N Bytes from Master



Slave Transfer N Bytes to Master





S = Start
A = Acknowledgment
NA = Negative-Acknowledgment
P = Stop

HT42B216-x Protocol Description

The HT42B216-x protocol combines ASCII code and hexadecimal code to configure CAN communication parameters and convert between I²C data and CAN messages.

Command Set

Data transmission starts with ASCII and ends with the Carriage Return character, CR (0x0D). In the following subsections, the hexadecimal code is represented by a backslash \.

Command	Response	Function
Open/Close the CAN Bus Device		
C[CR]	[CR]	Close the CAN bus device if it is opened
O[CR]	[CR]	Open the CAN bus device in Normal mode
l[CR]	[CR]	Open the CAN bus device in Loopback mode
L[CR]	[CR]	Open the CAN bus device in Listen mode
Setting CAN Bitrate (Standard)		
S\00[CR]	[CR]	Set the CAN bus bitrate to 5K
S\01[CR]	[CR]	Set the CAN bus bitrate to 10K
S\02[CR]	[CR]	Set the CAN bus bitrate to 20K
S\03[CR]	[CR]	Set the CAN bus bitrate to 50K
S\04[CR]	[CR]	Set the CAN bus bitrate to 100K
S\05[CR]	[CR]	Set the CAN bus bitrate to 125K
S\06[CR]	[CR]	Set the CAN bus bitrate to 250K
S\07[CR]	[CR]	Set the CAN bus bitrate to 500K
S\08[CR]	[CR]	Set the CAN bus bitrate to 800K
S\09[CR]	[CR]	Set the CAN bus bitrate to 1M
Transmitting a CAN Frame		
t\0i\ii\l\dd\dd\...\dd[CR]	z[CR]	Transmits a standard CAN frame (11-bit) over the CAN bus
T\ii\ii\ii\ii\l\dd\dd\...\dd[CR]	Z[CR]	Transmits an extended CAN frame (29-bit) over the CAN bus
Transmitting a Remote Request CAN Frame		
r\0i\ii\l[CR]	z[CR]	Transmits a standard remote request (11-bit) over the CAN bus
R\ii\ii\ii\ii\l[CR]	Z[CR]	Transmits an extended remote request (29-bit) over the CAN bus
Setting Acceptance Mask		
m\ii\ii[CR]	[CR]	Set acceptance filter mask for standard CAN frame (11-bit) identifier
m\ii\ii\ii\ii[CR]	[CR]	Set acceptance filter mask for extended CAN frame (29-bit) identifier
Setting Acceptance Code		
M\ii\ii[CR]	[CR]	Set acceptance filter code for standard CAN frame (11-bit) identifier
M\ii\ii\ii\ii[CR]	[CR]	Set acceptance filter code for extended CAN frame (29-bit) identifier
Getting Status Flags		
F[CR]	F\xx[CR]	Get CAN bus status
Getting the CAN Bridge VID		
V[CR]	V\04\D9[CR]	Get the CAN Bridge VID
Getting the CAN Bridge PID		
N[CR]	N\mm\nn[CR]	Get the CAN Bridge PID
Getting Version Information		
v[CR]	v\yy\zz[CR]	Get the current firmware version
CAN Bridge Enters Standby Mode		
STBY[CR]	[CR]	The CAN Bridge enters standby mode
Reset the CAN Bridge		
RST[CR]	[CR]	Reset
CAN Busoff Recovery		
RCY[CR]	[CR]	Set the CAN busoff recovery sequence

Command List

Example:

Set bitrate = 500kbps, normal mode and send standard CAN frame (ID = 7DFh, DLC=4, Data = 11 22 3344).

Command	Response	Function
C[CR]	[CR]	Close the CAN bus device
S\07[CR]	[CR]	Set bitrate to 500kbps
O[CR]	[CR]	Open in normal mode
t\07\DF\04\11\22\33\44[CR]	z[CR]	Send standard CAN message

Example:

Set bitrate = 250kbps, normal mode and receive extended CAN frame (ID = 543h).

Command	Response	Function
C[CR]	[CR]	Close the CAN bus device
S\06[CR]	[CR]	Set bitrate to 250kbps
O[CR]	[CR]	Open in normal mode
m\1F\FF\FF\FF[CR]	[CR]	Set acceptance filter mask for extended CAN frame identifier
M\00\00\05\43[CR]	[CR]	Set acceptance filter code for extended CAN frame identifier

Open/Close the CAN Bus Device

Command	Response	Function
C[CR]	[CR]	Close the CAN bus device if it is opened
O[CR]	[CR]	Open the CAN bus device in Normal mode
l[CR]	[CR]	Open the CAN bus device in Loopback mode
L[CR]	[CR]	Open the CAN bus device in Listen mode

Setting CAN Bitrate (Standard)

Command	Response	Function
S\00[CR]	[CR]	Set the CAN bus bitrate to 5K
S\01[CR]	[CR]	Set the CAN bus bitrate to 10K
S\02[CR]	[CR]	Set the CAN bus bitrate to 20K
S\03[CR]	[CR]	Set the CAN bus bitrate to 50K
S\04[CR]	[CR]	Set the CAN bus bitrate to 100K
S\05[CR]	[CR]	Set the CAN bus bitrate to 125K
S\06[CR]	[CR]	Set the CAN bus bitrate to 250K
S\07[CR]	[CR]	Set the CAN bus bitrate to 500K
S\08[CR]	[CR]	Set the CAN bus bitrate to 800K
S\09[CR]	[CR]	Set the CAN bus bitrate to 1M

Example:

Set CAN bitrate=500kbps and in Normal mode.

>S\07[CR]O[CR]

Set CAN bitrate=500kbps and in Loopback mode.

>S\07[CR]l[CR]

Set CAN bitrate=500kbps and in Listen mode.

>S\07[CR]L[CR]

Transmitting a Standard CAN Frame

Command	Response	Function
t\0i\ii\l\dd\dd\...\dd[CR]	z[CR]	Transmits a standard CAN frame (11-bit) over the CAN bus

\0i\ii: Standard 11-bit CAN frame identifier (000h~7FFh), MSB → LSB

\l: Data length 0~8 bytes (00h~08h)

\dd: Data byte value (00h~FFh)

Example:

Send a standard CAN frame with ID = 7DFh, DLC = 4, Data = 11 22 33 44.

> t\07\DF\04\11\22\33\44[CR]

Transmitting an Extended CAN Frame

Command	Response	Function
T\ii\ii\ii\ii\l\dd\dd\...\dd[CR]	Z[CR]	Transmits an extended CAN frame (29-bit) over the CAN bus

\ii\ii\ii\ii: Extended 29-bit CAN frame identifier (00000000h~1FFFFFFFh), MSB → LSB

\l: Data length 0~8 bytes (00h~08h)

\dd: Data byte value (00h~FFh)

Example:

Send an extended CAN frame with ID = 18DB33F1h, DLC = 8, Data = 11 22 33 44 55 66 77 88.

> T\18\DB\33\F1\08\11\22\33\44\55\66\77\88[CR]

Transmitting a Standard Remote Request CAN Frame

Command	Response	Function
r\0i\ii\l[CR]	z[CR]	Transmits a standard remote request (11-bit) over the CAN bus

\0i\ii: Standard remote request 11-bit CAN frame identifier (000h~7FFh), MSB → LSB

\l: Data length 0~8 bytes (00h~08h)

Example:

Send a standard remote Request CAN frame with ID = 7DFh, DLC = 4 and request 4 data bytes.

> r\07\DF\04[CR]

Transmitting an Extended Remote Request CAN Frame

Command	Response	Function
R\ii\ii\ii\ii\l[CR]	Z[CR]	Transmits an extended remote request (29-bit) over the CAN bus

\ii\ii\ii\ii: Extended remote request 29-bit CAN frame identifier (00000000h~1FFFFFFFh), MSB → LSB

\l: Data length 0~8 bytes (00h~08h)

Example:

Send an extended remote Request CAN frame with ID = 18DB33F1h, DLC = 8 and request 8 data bytes.

> R\18\DB\33\F1\08[CR]

Setting Acceptance Mask and code

Command	Response	Function
m\ii\ii[CR]	[CR]	Set acceptance filter mask for standard CAN frame (11-bit) identifier
m\ii\ii\ii\ii[CR]	[CR]	Set acceptance filter mask for extended CAN frame (29-bit) identifier
M\ii\ii[CR]	[CR]	Set acceptance filter code for standard CAN frame (11-bit) identifier
M\ii\ii\ii\ii[CR]	[CR]	Set acceptance filter code for extended CAN frame (29-bit) identifier

\0i\ii: Set acceptance filter mask/code for standard data CAN frame (11-bit) identifier. (0000h~07FFh)

\8i\ii: Set acceptance filter mask/code for standard remote CAN frame (11-bit) identifier. (8000h~87FFh)

\ii\ii\ii\ii: Set acceptance filter mask/code for extended data CAN frame (29-bit) identifier. (00000000h~1FFFFFFFh)

\ii\ii\ii\ii: Set acceptance filter mask/code for extended remote CAN frame (29-bit) identifier. (80000000h~9FFFFFFFh)

Receive CAN frame filter message maximum number = 30.

Note: 1. m\ii\ii & M\ii\ii or m\ii\ii\ii\ii & M\ii\ii\ii\ii, where ii is 0, has no-filter function, and can receive all IDs on the CAN bus, the previous relevant receive filter settings are invalid.

2. In the no-filter mode, data frames and remote frames can be received, where Data frame maximum number = 24, Remote frame maximum number = 6.

Example:

Set receive standard CAN frame with ID = 7DFh & extended CAN frame with ID = 18DB33F1h.

> m\07\FF[CR]M\07\DF[CR]m\1F\FF\FF\FF[CR]M\18\DB\33\F1[CR]

Set receive all standard or extended CAN frame.

> m\00\00\00\00[CR]M\00\00\00\00[CR]

Set receive standard remote CAN frame with ID = 7DFh & extended remote CAN frame with

ID = 18DB33F1h.

> m\87\FF[CR]M\87\DF[CR]m\9F\FF\FF\FF[CR]M\98\DB\33\F1[CR]

Getting Status Flags

Command	Response	Function
F[CR]	F\xx[CR]	Get CAN bus status

\xx: CAN bus status.

CAN Bus Status	Value	Description
CAN_TX_OK	6	Transmitted a CAN frame successfully
CAN_BOFF	-8	The CAN module is in busoff state
CAN Bus Last Error Code (Reference ISO11898-1)		
CAN_RERR_CRC	-10	CRC Error
CAN_TERR_BIT1	-11	Bit1 Error
CAN_TERR_BIT0	-12	Bit0 Error
CAN_TERR_ACK	-13	Ack Error
CAN_RERR_FORM	-14	Form Error
CAN_RERR_STUFF	-15	Stuff Error
CAN_TRX_SUCCEED	-16	No Error

Example:

Transmitted a CAN frame successfully.

>F\06[CR]

Getting the CAN Bridge VID

Command	Response	Function
V[CR]	V\04\D9[CR]	Get the CAN Bridge Vendor ID

\04\D9: Vendor ID (Hex, High → Low byte)

Example:

VID = 0x04D9

> V\04\D9[CR]

Getting the CAN Bridge PID

Command	Response	Function
N[CR]	N\mm\nn[CR]	Get the CAN Bridge Product ID

\mm\nn: Product ID (Hex, High → Low byte)

I²C to CAN Bridge HT42B216-x PID = 0xB216

Example:

> N\B2\16[CR]

Getting Version Information

Command	Response	Function
v[CR]	v\yy\zz[CR]	Get the current firmware version

\yy\zz: Firmware version (Hex, High → Low byte)

Example:

Version 1.00

> v\01\00[CR]

CAN Bridge Enters Standby Mode

Command	Response	Function
STBY[CR]	[CR]	CAN Bridge enters standby mode

If the CAN Bridge is required to exit Standby mode, the MCU should send 1-byte I²C data.

Example:

The CAN Bridge enters the Standby mode to save power.

> STBY[CR]

Reset the CAN Bridge

Command	Response	Function
RST[CR]	[CR]	Reset

Example:

> RST[CR]

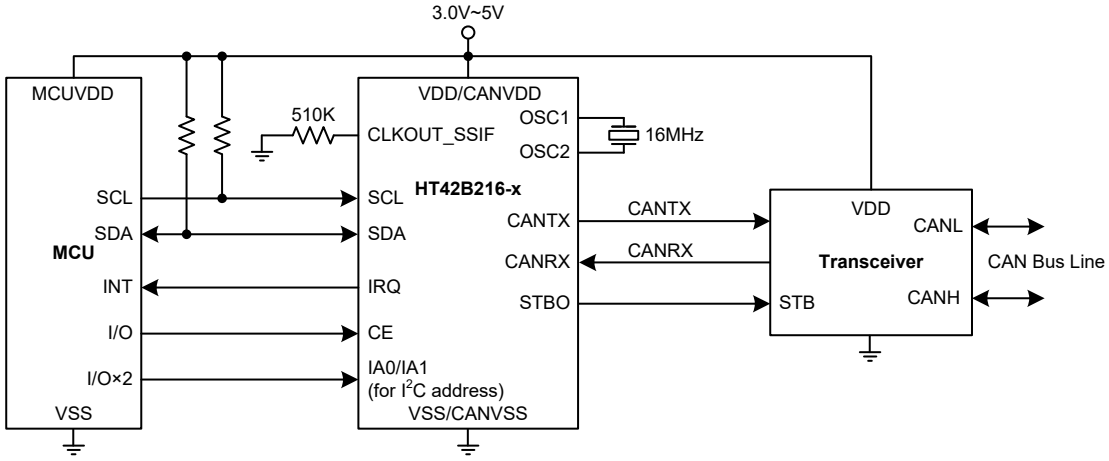
CAN Busoff Recovery

Command	Response	Function
RCY[CR]	[CR]	Set the CAN busoff recovery sequence

Example:

> RCY[CR]

Application Circuits



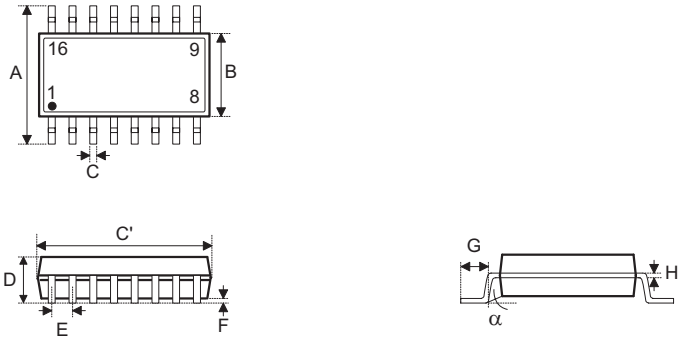
Package Information

Note that the package information provided here is for consultation purposes only. As this information may be updated at regular intervals users are reminded to consult the Holtek website for the latest version of the Package/Carton Information.

Additional supplementary information with regard to packaging is listed below. Click on the relevant section to be transferred to the relevant website page.

- Package Information (include Outline Dimensions, Product Tape and Reel Specifications)
- The Operation Instruction of Packing Materials
- Carton information

16-pin NSOP (150mil) Outline Dimensions



Symbol	Dimensions in inch		
	Min.	Nom.	Max.
A		0.236 BSC	
B		0.154 BSC	
C	0.012	—	0.020
C’		0.390 BSC	
D	—	—	0.069
E		0.050 BSC	
F	0.004	—	0.010
G	0.016	—	0.050
H	0.004	—	0.010
α	0°	—	8°

Symbol	Dimensions in mm		
	Min.	Nom.	Max.
A		6.00 BSC	
B		3.90 BSC	
C	0.31	—	0.51
C’		9.90 BSC	
D	—	—	1.75
E		1.27 BSC	
F	0.10	—	0.25
G	0.40	—	1.27
H	0.10	—	0.25
α	0°	—	8°