Device Requirements for CAN Transceiver Tests

Version 1.5

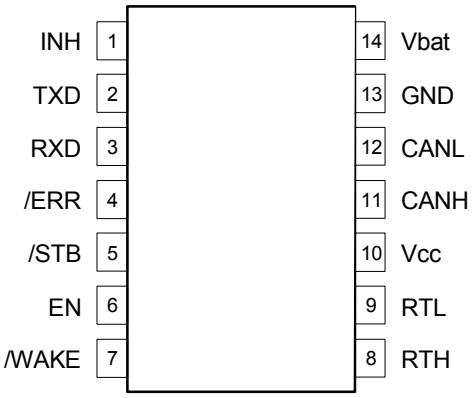
Contents

Hardware Requirements for Transceiver Tests

Supported Interfaces

We offer two sockets for low speed or high speed transceiver device. Please find below the possible interfaces and pinning.

14 pin DIL socket:



Note:

No further software is required for stand alone devices.

For emulated devices you have to provide us at least with STB and EN timings.

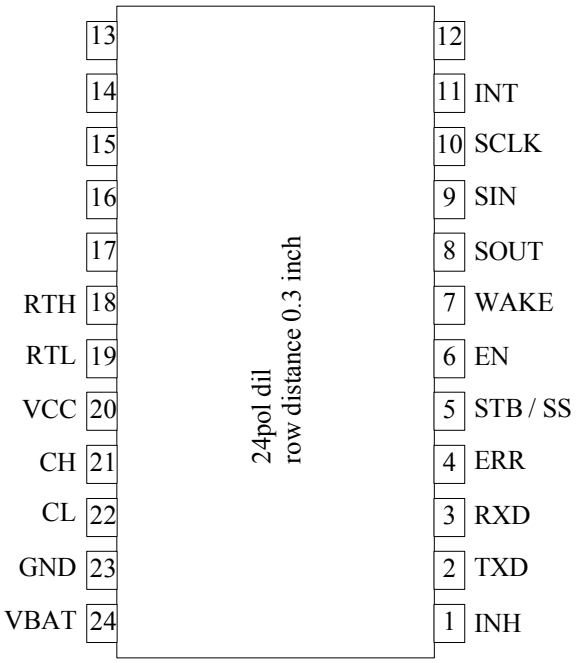
Devices with 8 pins or different package must be placed on a 14 pin DIL adapter and routed to the related pins by customer.

Missing INH, ERR, EN and STB can be left open.

Inverted signals for INH, ERR, EN, STB or WAKE will be adapted by test software.

SBC device with or without SPI control and devices more than 14 pins.

24 pin DIL socket:



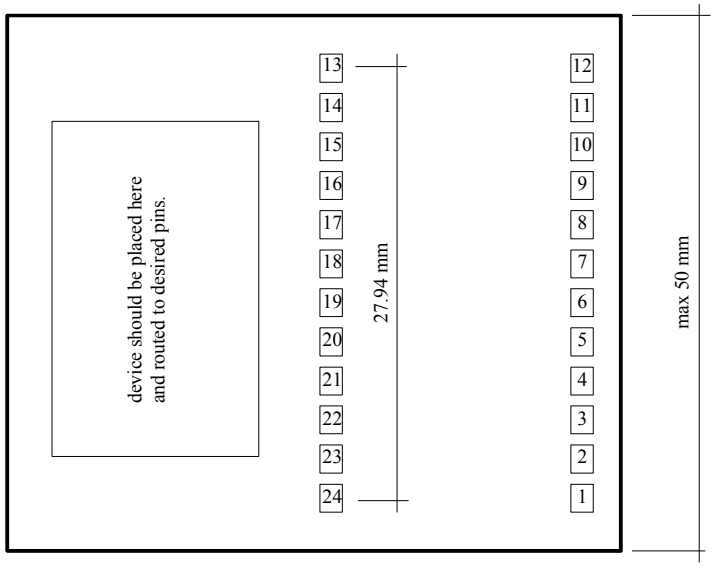
SPI communication with HC12 is provided by pins SIN, SOUT, SS and SCLK.

SPI requires a special software driver, which is to be delivered by customer. See Software requirements.

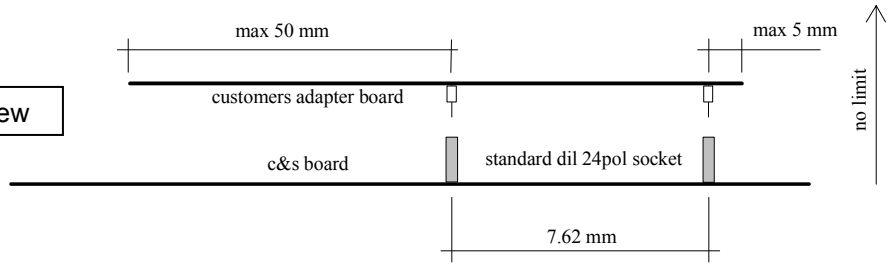
HighSpeed devices: Split Pin can be linked to RTH or RTL

SBCs generally: - remaining Pin out can be linked to other not used “stand alone” pins
- if Vcc supply is not necessary, Vcc should be feft open

Adapter top view



Adapter side view



Software Requirements for Transceiver Tests

SPI driver for SBC devices

Devices controlled by SPI commands require an additional software implementation. This driver has to be delivered by the customer.

The driver has to be written in c-code matching the cosmic-cross-compiler v4.21. Source-code has to be delivered for eventual necessary adaptations.

The test software expects following functions combined in TRX_API.H file.

Transceiver Application Programming Interface

Function description:

```
/* definition of constants for transceiver */
#define TRX_NORMAL                  0x01
#define TRX_SLEEP                   0x00
#define TRX_RECEIVEONLY             0x02
#define TRX_STANDBY                 0x03
#define TRX_GOTO_SLEEP              0x04

#define TRX_FCT_ERROR               0xFF

#define TRX_INH_STATUS_ON           0x01
#define TRX_INH_STATUS_FLOATING     0x00
```

```
/*****************************************************************************************
 *  unsigned char TRX_API_Init (void)
 *
 *  FILENAME:    TRX_API.H
 *
 *  PARAMETERS:        --
 *
 *  DESCRIPTION: use this function to initialize your driver
 *
 *
 *  RETURNS:     Error value
 *               TRX_FCT_ERROR    -> error during execution
 *               0 no error
 *               > 0 error e.g. error code !!!!!!!!!!values to be defined !!!!!!!!!!!!
 *
 *****************************************************************************************/
 unsigned char TRX_API_Init (void);

/*****************************************************************************************
 *  unsigned char TRX_GetError (void)
 *
 *  FILENAME:   TRX_API.H
 *
 *  PARAMETERS:        --
 *
 *  DESCRIPTION: read the error information from the transceiver
 *
 *
 *  RETURNS:     Error value
 *               TRX_FCT_ERROR    -> error during execution
 *               0 no error
 *               > 0 error e.g. error code !!!!!!!!!!values to be defined !!!!!!!!!!!!
 *
 *****************************************************************************************/
 unsigned char TRX_GetError (void);
```

```
/*****************************************************************************************
** unsigned char TRX_SetOperationMode (unsigned char mode)
*
*  FILENAME:   TRX_API.H
*
*  PARAMETERS:        mode to be set
*               TRX_NORMAL       -> set TRX to normal mode
*               TRX_GOTO_SLEEP   -> set TRX to sleep mode
*               TRX_STANDBY      -> set TRX to standby mode
*               TRX_RECEIVEONLY  -> set TRX to receive only mode
*               other values will be ignored
*
*
*  DESCRIPTION: set the transceiver in the provided mode
*
*  RETURNS:     function results
*               TRX_FCT_ERROR    -> error during execution
*               0 mode is set
*               1 wrong parameter or mode could not be set
*               all other values not valid
*
*****************************************************************************************/
unsigned char TRX_SetOperationMode (unsigned char mode);


/*****************************************************************************************
*  unsigned char TRX_GetOperationMode (void)
*
*  FILENAME:   TRX_API.H
*
*  PARAMETERS:        ---
*
*
*  DESCRIPTION: get the transceiver operation mode
*
*  RETURNS:           transceiver operation mode
*               TRX_NORMAL       -> TRX is in normal mode
*               TRX_SLEEP        -> TRX is in sleep mode
*               TRX_STANDBY      -> TRX is in standby mode
*               TRX_RECEIVEONLY  -> TRX is in receive only mode
*               TRX_FCT_ERROR    -> error during execution
*               other values are not valid
*
*
*****************************************************************************************/
unsigned char TRX_GetOperationMode (void);


 /*****************************************************************************************
*  unsigned char TRX_GetINHState (void)
*
*  FILENAME:   TRX_API.H
*
*  PARAMETERS:        --
*
*  DESCRIPTION: read the status of the INH bit
*
*
*  RETURNS:     Error value
*               TRX_FCT_ERROR            -> error during execution of function
*               TRX_INH_STATUS_ON        -> INH == 1 -> voltage regulator on
*               TRX_INH_STATUS_FLOATING  -> INH == 0 -> voltage regulator on
*               > 0 error e.g. error code !!!!!!!!!!values to be defined !!!!!!!!!!!
*
*
*****************************************************************************************/
unsigned char TRX_GetINHState (void);
```

```
/*****************************************************************************************
 *  unsigned char TRX_WD (void)
 *
 *  FILENAME:    TRX_API.H
 *
 *  PARAMETERS:  --
 *
 *  DESCRIPTION: kill watchdog by sending spi commands within desired time
 *
 *
 *  RETURNS:     Error value
 *               TRX_FCT_ERROR            -> error during execution of function
 *               > 0 error e.g. error code !!!!!!!!!!values to be defined !!!!!!!!!!!!
 *
 *
 *****************************************************************************************/
unsigned char TRX_WD (void);
```

Integration of driver into test software

The function TRX_API_INIT is called in main routine before entering while(1){....}.

Following functions:

- TRX_GetError()
- TRX_SetOperationMode()
- TRX_GetOperationMode()
- TRX_GetINHState()

are called after user requests for it in pc gui.

The function TRX_WD() is called in a timer interrupt function within every 500µs. Use this timer called function to satisfy your device watch dog with spi commands if needed.

Misc

µC

The used µC for this test system is the motorola hc12dg128 with 2 can interfaces running with 8Mhz crystal.

Devices

For network testing we need 50 transceiver for lowspeed- and 25 transceiver for highspeed- testing.

Address

Please send your devices to below address:

C&S group GmbH
Am Exer 19c
D-38302 Wolfenbuettel
Germany

do not feel hesitate to ask more details.

Tel: +49 5331 90-555 0
Fax: +49 5331 90-555 110

E-Mail: info@cs-group.de

Web: www.cs-group.de

Literature

Documentation for the hc12dg128 µC is available at

http://www.freescale.com/

Documentation for TJA1054 and TJA1041 is available at

http://www.nxp.com/

Documentation for Cosmic-Cross-Compiler is available at

http://www.cosmic-software.com