TI Designs: TIDA-01406

Energy Efficient and Isolated CANopen Interface for PLC and Communication Modules

Description

CAN and CANopen are legacy fieldbus protocols that are used in many industrial applications including factory automation and control. Whenever high voltage can damage the end equipment, there is need for isolation. Modern smart factories with many automation nodes also benefit in reducing total power consumption when every device uses less energy. The TIDA-01406 is a TI Design for energy efficient and isolated CANopen interfaces for PLC and communication modules. It is based on a BeagleBone Black cape, which can be easily tested with the BeagleBone Black development board.

Resources

TIDA-01406	Design Folder
ISO1050	Product Folder
SN6501	Product Folder
TPD2E007	Product Folder
BeagleBone Black Development Board	Tool Folder

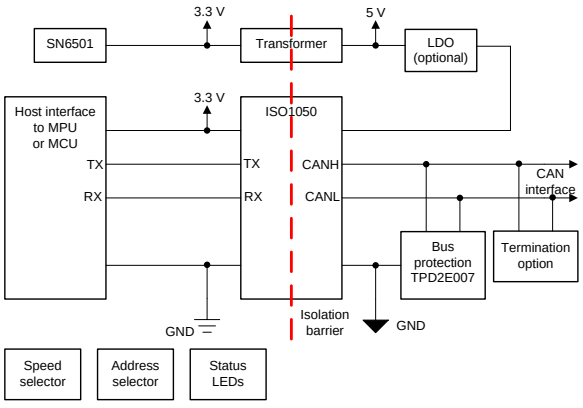
TI E2E™ Community — ASK Our E2E Experts

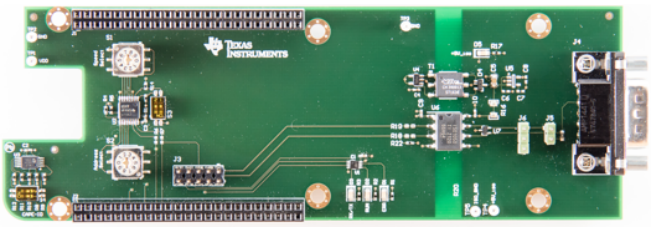
Features

- CAN Transceiver Reference Design With ISO1050 and SN6501
- Small Form Factor Design
- Isolation of ISO1050:
  - 2.5-kV Isolation With DUB Package
- BeagleBone Black Cape Form Factor
- Expansion Interface for BeagleBone Black and for Generic Processor Boards
- CAN Bus Address and CAN Bus Speed Rotary Switches
- CAN Status LEDs

Applications

- Communication Modules
- PLC





An IMPORTANT NOTICE at the end of this TI reference design addresses authorized use, intellectual property matters and other important disclaimers and information.

1 System Description

A Controller Area Network (CAN) bus is an interface for automotive and vehicle applications developed in 1986 by Bosch GmbH. Due to its low cost, the CAN bus has been adopted to be used in factory and process automation to control the production lines. Newly developed products still support the CAN interface to support legacy and factory deployed remote input/output devices, motor drives, sensors, and actuators.

CANopen is a communication protocol and device profile over CAN bus. CANopen is released by the organization CAN in Automation (CiA) and the CANopen protocol is specified in CiA 301.

Each CAN node consists of the following functional blocks:

- Processor unit: This is a microcontroller or microprocessor. The processor unit runs the CAN stack, CANopen protocol stack, and application software layer above the CANopen protocol stack. The CAN stack has an application programmers interface, which interfaces to the CANopen protocol stack and to customer applications.
- CAN controller: This is a peripheral block inside the microcontroller or microprocessor. The CAN controller is responsible for transmitting and receiving CAN messages.
- CAN transceiver: This is an external device, separate from the microcontroller or microprocessor, that converts the data stream from CAN bus levels to CAN controller levels. The transceiver has the isolation barrier added to protect the CAN controller from interference and destructive voltages on the CAN bus.

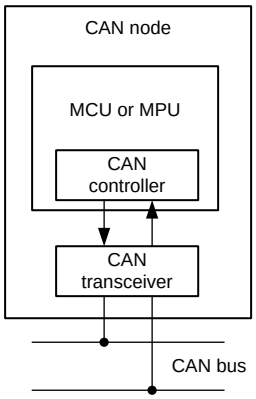


Figure 1. CAN Node

This TI Design integrates the isolated CAN transceiver on a BeagleBone Black cape form factor. This enables customers to evaluate and validate TI's ISO1050 CAN transceiver using an existing software infrastructure available through the BeagleBone open source community (CAN tools, CANopen Stack). In addition to the cape form factor, the TIDA-01406 design supports a generic host interface pin header to allow customers to connect the cape form factor board to any microprocessor or microcontroller development board. This TI Design provides additional hardware support for two rotary input switches to select CAN bus speed and CAN device identity (ID). In addition it supports three CAN status LEDs.

1.1 Key System Specifications

Table 1. Key System Specifications

PARAMETER	SPECIFICATIONS
Isolated CAN transceiver	See the ISO1050 datasheet[1]
Dominant-time-out circuit	See the ISO1050 datasheet
BeagleBone Black cape form factor	Supports BeagleBone Black device tree overlay (DTO) and device driver for CAN controller, I^2C bus, and GPIO interface.
Energy efficient transformer driver	See the SN6501 datasheet[2]
ESD and surge protection	See the TPD2E007 datasheet[3]

2 System Overview

2.1 Block Diagram

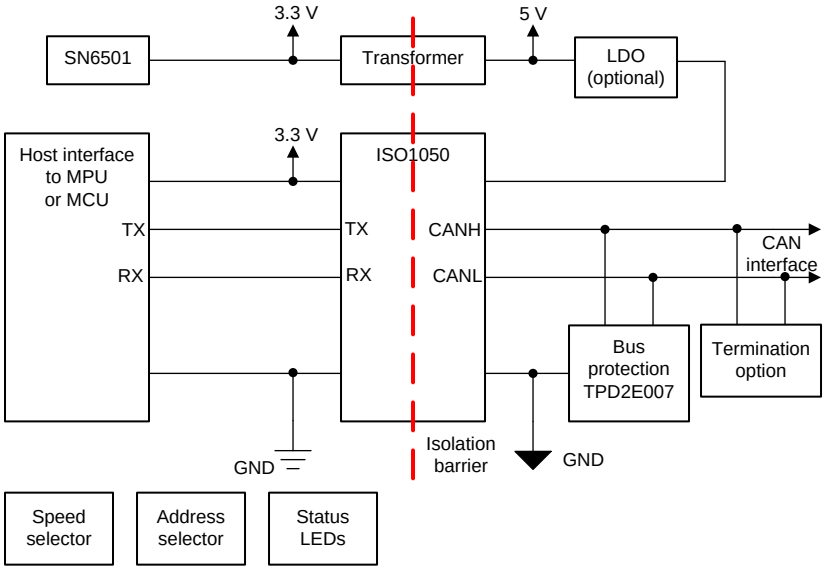


Figure 2. CANopen Block Diagram

2.2 Design Considerations

2.2.1 ISO1050 and SN6501

The ISO1050 device is available in two different packages: DW and DUB. The CANopen cape uses the DUB package with a lower package pin count and up to 2500-V_{RMS} of isolation. The isolated voltage supply is generated by the SN6501 transformer driver. The default PCB assembly uses the isolated 5 V after the diodes to power the ISO1050. However, it is possible through an assembly option to use an LDO for filtering the supply voltage for the ISO1050 device.

The CANopen cape has the option to insert a termination resistor of 120 Ω through a wire jumper into the CAN bus. CAN termination is needed at the first and last CAN devices in the CAN bus line topology. The CANopen cape also supports CAN bus protection devices such as the TPD2E007. The CANopen cape supports a 3-pin connector to the access the CAN_H, CAN_L, and isolated GND signals.

2.2.2 Rotary Switch Interface

The CANopen cape supports two dedicated rotary switches to set the CAN bus speed and the CAN device address. The rotary switches are read out through an 8-bit port expander, which is accessed through the I^2C1 interface. Each rotary switch has 10 positions and those are translated into 4 bits of output data. The output data of the rotary switches are combined into a single 8-bit data word, which is read by the I^2C port expander TPS9534.

The 8-bit output data is translated as follows:

- Bit 0 to 3: CAN speed value selected with position switch
- Bit 4 to 7: CAN address value selected with position switch

Note that the rotary switch output data are controlled by the user application to set CAN bus speed and CAN device address. The CANopen stack does not process this information.

2.2.3 Status LED Interface

The CANopen cape supports three CAN status LEDs. Each LED is controlled by a dedicated processor GPIO. The CANopen cape supports the following LEDs:

- ERR: Indicates an CAN bus error
- RUN: Indicates the CAN bus is in operation
- RX/TX: Indicates the CAN bus is receiving or transmitting

Note that all three LEDs are controlled by the user application through GPIO access. The CANopen stack does not control the LEDs directly.

2.2.4 BeagleBone Black Cape Identity EEPROM

The Linux operating system on the BeagleBone Black uses an ID EEPROM to detect attached capes. Based on the detected cape ID, the Linux system loads the respective device tree overlay (DTO) for pin muxing and initialing the cape drivers. The cape detection is performed through the I^2C2 interface.

2.2.5 BeagleBone Black Cape Expansion Connector Assignment

Connector J8 on the CANopen cape connects to connector P8 on the BeagleBone Black. Connector J9 on the CANopen cape connects to connector P9 on the BeagleBone Black. Table 2 specifies the pins and signals used by the CANopen cape:

Table 2. CANopen Cape Interface to BeagleBone Black

CONNECTOR	PIN	MODE	DESCRIPTION
8	1, 2	GND	Ground
9	1, 2, 43, 44, 45, 46	GND	Ground
9	3, 4	VDD_3V3	3.3-V voltage supply from BBB
9	12	GPIO1[28] (Mode 7)	CAN LED 3: RX/TX
9	15	GPIO1[16] (Mode 7)	CAN LED 1: ERR
9	17	I2C1_SCL (Mode 2)	I^2C interface to position switch port expander
9	18	I2C1_SDA (Mode 2)	I^2C interface to position switch port expander
9	19	I2C2_SCL (Mode 3)	I^2C interface to cape ID EEPROM
9	20	I2C2_SDA (Mode 3)	I^2C interface to cape ID EEPROM
9	23	GPIO1[17] (Mode 7)	CAN LED 2: RUN
9	24	DCAN1_RX (Mode 2)	CAN1 receive
9	26	DCAN1_TX (Mode 2)	CAN1 transmit

2.3 Highlighted Products

2.3.1 ISO1050 Isolated CAN Transceiver

- Meets the requirements of ISO11898-2
- 5000-V_{RMS} isolation (ISO1050DW)
- 2500-V_{RMS} isolation (ISO1050DUB)
- Fail-safe outputs
- Low loop delay: 150 ns (typical), 210 ns (maximum)
- 50-kV/μs typical transient immunity
- Bus-fault protection of –27 to 40 V
- Driver (TXD) dominant timeout function
- I/O voltage range supports 3.3-V and 5-V microprocessors
- VDE approval per DIN V VDE V 0884-10 (VDE V0884-10): 2006-12 and DIN EN 61010-1
- UL 1577 approved
- CSA approved for IEC 60950-1, IEC 61010-1, IEC 60601-1 3rd Ed (Medical), and Component Acceptance Notice 5A
- TUV 5-kV_{RMS} reinforced insulation approval for EN/UL/CSA 60950-1 (ISO1050DW only)
- CQC reinforced insulation per GB4843.1-2011 (ISO1050DW only)

2.3.2 SN6501 Transformer Driver for Isolated Power Supplies

- Push-pull driver for small transformers
- Single 3.3-V or 5-V supply
- High primary-side current drive:
  - 5-V supply: 350 mA (max)
  - 3.3-V supply: 150 mA (max)
- Low ripple on rectified output permits small output capacitors
- Small 5-pin SOT-23 package

2.3.3 TPD2E007 2-Channel ESD Protection Array for AC-Coupled/Negative-Rail Data Interfaces

- Industrial interfaces (RS-232, RS-485, RS-422, LVDS, CAN)
- IEC 61000-4-2 Level 4 ESD protection:
  - ±8-kV IEC 61000-4-2 contact discharge
  - ±15-kV IEC 61000-4-2 air-gap discharge
- IEC 61000-4-5 surge protection:
  - 4.5-A peak pulse current (8/20-µs pulse)
- IO capacitance 15 pF (max)
- Low 50-nA leakage current
- Space-saving PicoStar™ and SOT package

2.4 BeagleBone Black Development Board

BeagleBone Black is a low-cost, open source, community-supported development platform for ARM® Cortex®-A8 processor developers. Boot Linux in under 10-seconds and get started on the Sitara™ AM335x ARM Cortex-A8 processor development in less than 5 minutes with just a single USB cable. BeagleBone Black ships with the Debian GNU/Linux in onboard FLASH to start evaluation and development. Many other Linux distributions and operating systems are also supported on BeagleBone Black including:

- Debian
- Ubuntu
- Android™
- Fedora

BeagleBone Black's capabilities can be extended using plug-in boards called "capes" that can be plugged into BeagleBone Black's two 46-pin dual-row expansion headers.

3 Hardware, Software, Testing Requirements, and Test Results

3.1 Required Hardware and Software

3.1.1 Hardware

The following hardware items are required:

- BeagleBone Black
- TIDA-01406 Isolated CAN cape

3.1.2 Software

3.1.2.1 Update BeagleBone Black Linux Image

Be sure to update the Linux Kernel and file system to the latest software, especially if the BeagleBone Black board has not been used in a while. Follow the detailed instruction on how to download and flash the onboard eMMC on the BeagleBone Black here.

3.1.2.2 Proxy Server Configuration

When connecting the BeagleBone Black board to a network that is behind a firewall, configure the proxy server that allows access to the Internet. Setting up a proxy server allows one to download all the tools that are needed to get the TIDA-001406 functional. This proxy server setup description is useful. Alternatively, contact a company network administrator for help with setting up the proxy server.

3.1.2.3 EEPROM Tool and EEPROM Cape Programming

The TIDA-01406 cape design needs a generated EEPROM binary file. Find the EEPROM binary file generation tool here with a detailed EEPROM tutorial here.

1. Clone the BBCape_EEPROM repository and build the BBCape_EEPROM executable:

```
root@beaglebone:~# git clone https://github.com/picoflamingo/BBCape_EEPROM
root@beaglebone:~# cd BBCape_EEPROM
root@beaglebone:~# make
```

   The program is menu driven and allows you to create the EEPROM data.

2. Follow the description from the EEPROM generation tool website to generate an EEPROM binary with the following parameters:
   - Cape Name (32 bytes): isolated CAN interface
   - Cape Version (4 bytes): 00A0
   - Cape Manufacturer (16 bytes): TI
   - Part Number (16 bytes): BB-CAN1
   - Serial Number (12 bytes): TIDA-01406
3. Store the EEPROM binary in a file called cape.bin.
4. Use the cat command to write the file cape.bin into the EEPROM onboard the TIDA-01406 design.

```
cat cape.bin >
/sys/bus/i2c/devices/2-0057/eeprom
```

5. Reboot the BeagleBone Black board after programming the EEPROM. Use dmesg to see if Linux has detected the TIDA-01406 cape and search for the following line:

```
2.532120] bone_capemgr bone_capemgr: slot #3: dtbo
                  'BB-CAN1-00A0.dtbo' loaded; overlay id #0
```

3.1.2.4 Install SocketCAN Utilities

Clone the SocketCAN utilities repository, then build and install the SocketCAN utilities executables. The Can-utils website with further instructions can be found here.

```
root@beaglebone:~# git clone https://github.com/linux-can/can-utils.git
root@beaglebone:~# cd can-utils
root@beaglebone:~# ./autogen.sh
root@beaglebone:~# ./configure
root@beaglebone:~# make
root@beaglebone:~# make install
```

3.1.2.5 Install CANopenSocket Stack

Clone the CANopenSocket stack by following the instructions on the CANopen Socket website.

3.2 Testing and Results

3.2.1 Test Setup

The test setup consists of two BeagleBone Black with ISO CAN capes and a test PC with a serial console or terminal to connect to the Linux system on the BeagleBone Black boards.

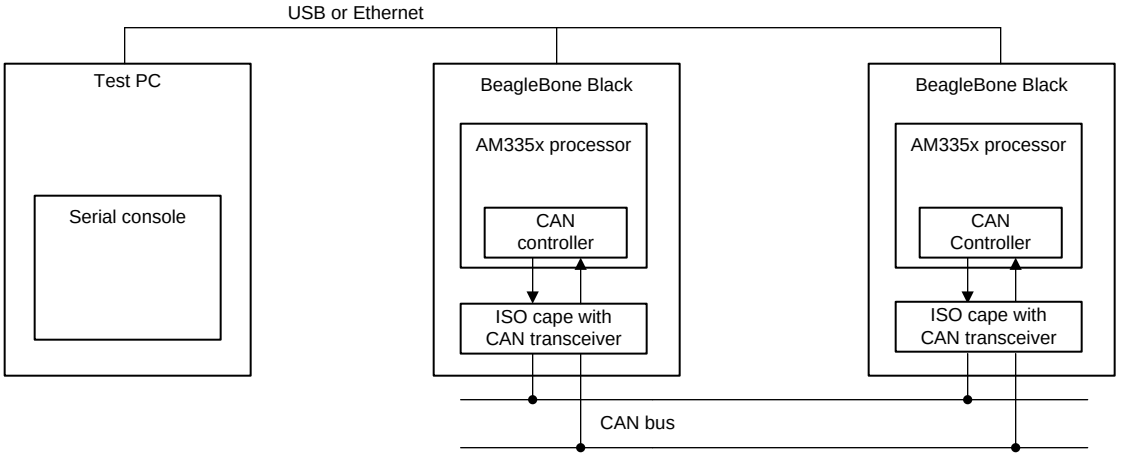


Figure 3. Test Setup

3.2.1.1 Cape EEPROM

Display the content of the TIDA-01406 cape EEPROM:

```
root@beaglebone:~# cat /sys/bus/i2c/devices/2-0057/eeprom | hexdump -C
00000000 aa 55 33 ee 41 31 69 73 6f 6c 61 74 65 64 20 43 |.U3.A1isolated C|
00000010 41 4e 20 69 6e 74 65 72 66 61 63 65 00 00 00 00 |AN interface....|
00000020 00 00 00 00 00 00 30 30 41 30 54 49 00 00 00 00 |......00A0TI....|
00000030 00 00 00 00 00 00 00 00 00 00 42 42 2d 43 41 4e |..........BB-CAN|
00000040 31 00 00 00 00 00 00 00 00 00 30 30 54 49 44 41 |1.........00TIDA|
00000050 30 31 34 30 36 00 00 00 00 00 00 00 00 00 00 00 |01406...........|
00000060 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |................| *
000000f0 00 00 00 00 ff ff ff ff ff ff ff ff ff ff ff ff |................|
00000100 ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff |................| *
00008000
```

3.2.1.2 CAN Interface

The SocketCAN utilities are used to test CAN protocol communication.

1. First, the CAN interface needs to get configured for both BBB#1 and BBB#2 boards. The CAN interface is configured with 125kbit/s with the following command:

   ```
   root@beaglebone:~# ip link set can0 up type can bitrate 125000
   ```

2. Enable CAN protocol monitor on the second BBB#2.

   ```
   BBB#2 --> root@beaglebone:~# candump can0
   ```

3. Now generate a CAN frame on the first BBB#1:

   ```
   BBB#1 --> root@beaglebone:~# cangen can0 -D 11223344DEADBEEF -L 8 -g 1
   ```

The CAN monitor on second BBB#2 will display the received CAN frame:

```
BBB#2
can0 731 [8] 11 22 33 44 DE AD BE EF
can0 588 [8] 11 22 33 44 DE AD BE EF
can0 1A4 [8] 11 22 33 44 DE AD BE EF
can0 50D [8] 11 22 33 44 DE AD BE EF
can0 3E2 [8] 11 22 33 44 DE AD BE EF
can0 464 [8] 11 22 33 44 DE AD BE EF
can0 784 [8] 11 22 33 44 DE AD BE EF
can0 5C6 [8] 11 22 33 44 DE AD BE EF
```

To verify the CANopenSocket stack functionality, follow the "Getting started with CANopen Socket" instructions on the CANopenSocket website.

3.2.2 CAN Speed and Device ID Position Switch

Enable the I2C1 interface:

```
root@beaglebone:~# cd /sys/devices/platform/bone_capemg
root@beaglebone:/sys/devices/platform/bone_capemgr# echo BB-I2C1 > slots
```

Send configuration parameter to the port expander:

```
root@beaglebone:/sys/devices/platform/bone_capemgr# i2cset -y 1 0x20 0
```

Read out the port expander data:

```
root@beaglebone:/sys/devices/platform/bone_capemgr# i2cget -y 1 0x20
```

A value is return which represents the values of the two position switched:

- Example: 0x91
- CAN speed set to 1.
- CAN address set to 9

3.2.3 CAN Status LED

To control GPIO48 (ERR LED):

```
root@beaglebone:/sys/class/gpio# echo 48 > export
root@beaglebone:/sys/class/gpio# cd gpio48
root@beaglebone:/sys/class/gpio/gpio48# echo "out" > direction
root@beaglebone:/sys/class/gpio/gpio48# echo "1" > value
root@beaglebone:/sys/class/gpio/gpio48# echo 0 > value
root@beaglebone:/sys/class/gpio/gpio48# cd ..
```

To control GPIO49 (RX/TX LED):

```
root@beaglebone:/sys/class/gpio# echo 49 > export
root@beaglebone:/sys/class/gpio# cd gpio49
root@beaglebone:/sys/class/gpio/gpio49# echo "out" > direction
root@beaglebone:/sys/class/gpio/gpio49# echo "1" > value
root@beaglebone:/sys/class/gpio/gpio49# echo 0 > value
root@beaglebone:/sys/class/gpio/gpio49# cd ..
```

To control GPIO60 (RX/TX LED):

```
root@beaglebone:/sys/class/gpio# echo 60 > export
root@beaglebone:/sys/class/gpio# cd gpio60
root@beaglebone:/sys/class/gpio/gpio60# echo "out" > direction
root@beaglebone:/sys/class/gpio/gpio60# echo "1" > value
root@beaglebone:/sys/class/gpio/gpio60# echo 0 > value
root@beaglebone:/sys/class/gpio/gpio60# cd ..
```

3.2.4 Test Results

3.2.4.1 CAN Protocol Exchange

The following figures show the test results for CAN_H and CAN_L signals at different CAN speeds.

For Figure 4:

- Full CAN frame
- CAN speed: 125 kHz
- Channel 1: CAN_H
- Channel 2: CAN_L

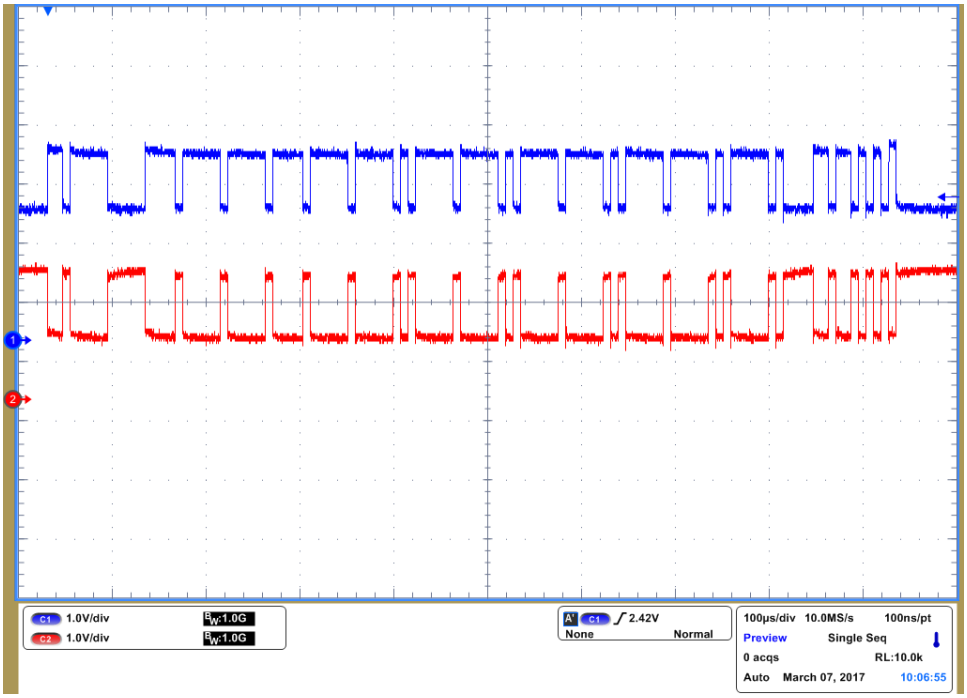


Figure 4. 125-kHz Frame

For Figure 5:

- Zoom in CAN frame
- CAN speed: 125 kHz
- Channel 1: CAN_H
- Channel 2: CAN_L

Figure 5. 125-kHz Zoom

For Figure 6:

- Full CAN frame
- CAN speed: 250 kHz
- Channel 1: CAN_H
- Channel 2: CAN_L

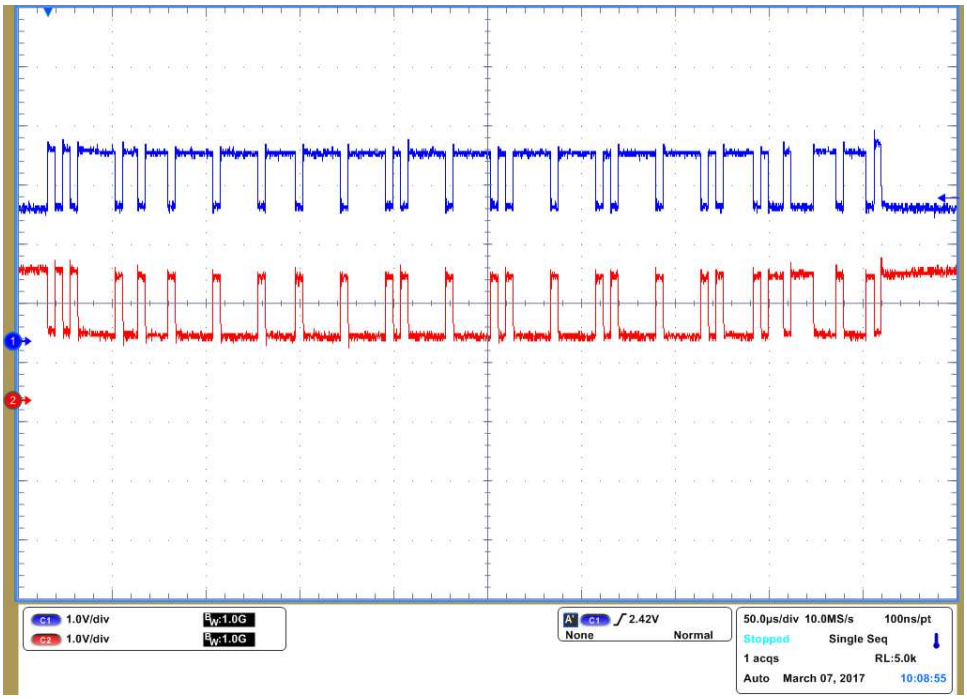


Figure 6. 250-kHz Frame

For Figure 7:

- Zoom in CAN frame
- CAN speed: 250 kHz
- Channel 1: CAN_H
- Channel 2: CAN_L

Figure 7. 250-kHz Zoom

For Figure 8:

- Full CAN frame
- CAN speed: 500 kHz
- Channel 1: CAN_H
- Channel 2: CAN_L

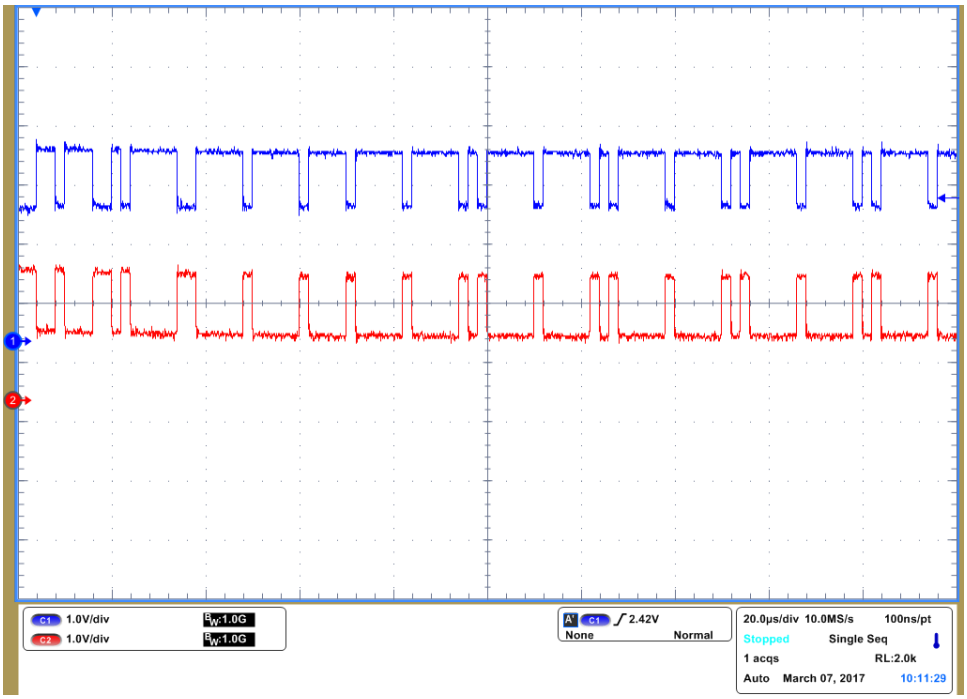


Figure 8. 500-kHz Frame

For Figure 9:

- Zoom in CAN frame
- CAN speed: 500 kHz
- Channel 1: CAN_H
- Channel 2: CAN_L

Figure 9. 500-kHz Zoom

For Figure 10:

- Full CAN frame
- CAN speed: 1 MHz
- Channel 1: CAN_H
- Channel 2: CAN_L

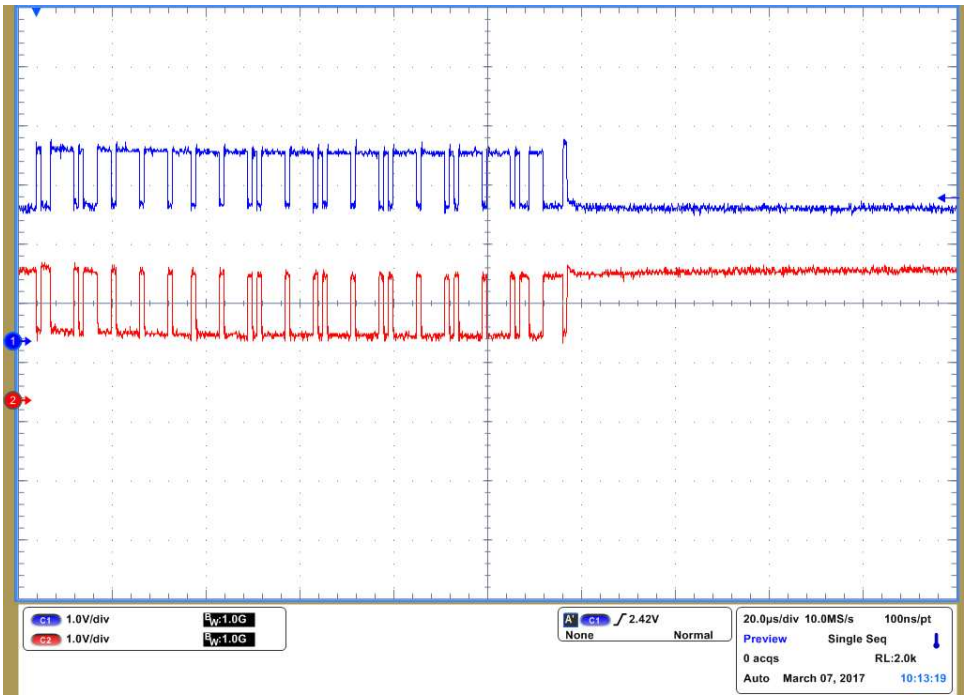


Figure 10. 1-MHz Frame

For Figure 11:

- Zoom in CAN frame
- CAN speed: 1 MHz
- Channel 1: CAN_H
- Channel 2: CAN_L

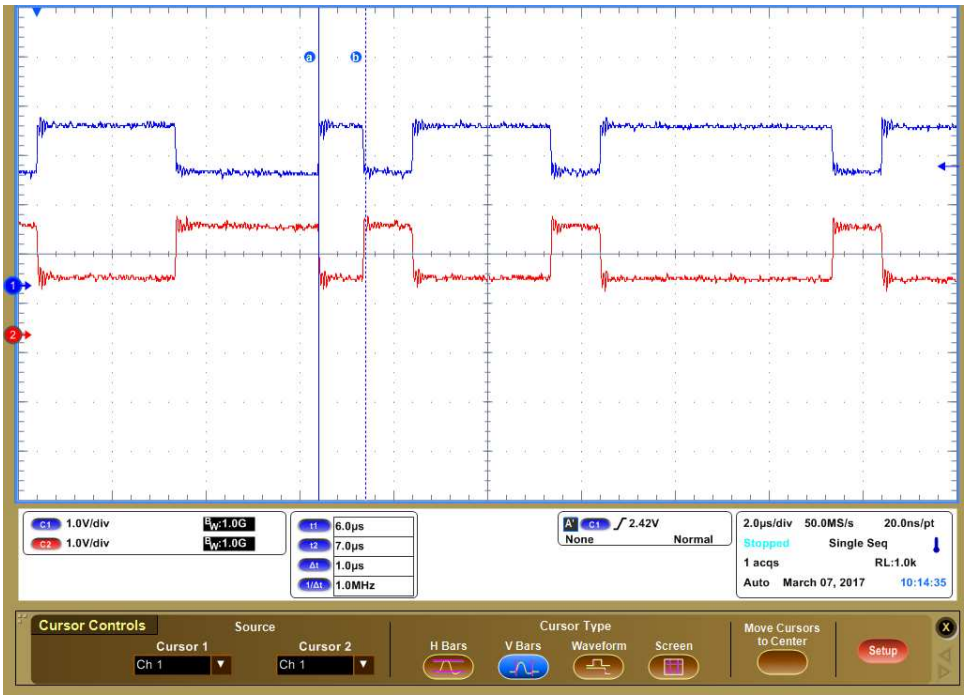


Figure 11. 1-MHz Zoom

3.2.4.2 Isolated Power

Figure 12 shows the isolated 5-V voltage supply during frame transmission:

- Channel 1: CAN_H
- Channel 2: CAN_L
- Channel 3: Isolated 5 V

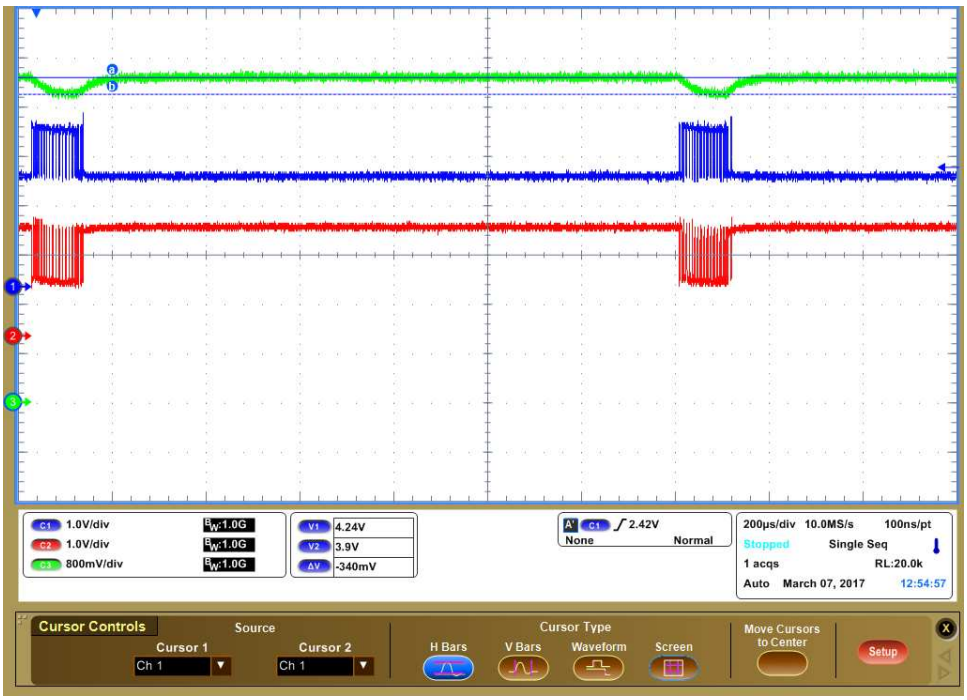


Figure 12. ISO Supply

3.2.4.3 Position Switch Readout

Figure 13 shows the test results for I²C interface:

- Channel 1: I2C1 SCL
- Channel 2: I2C1 SDA

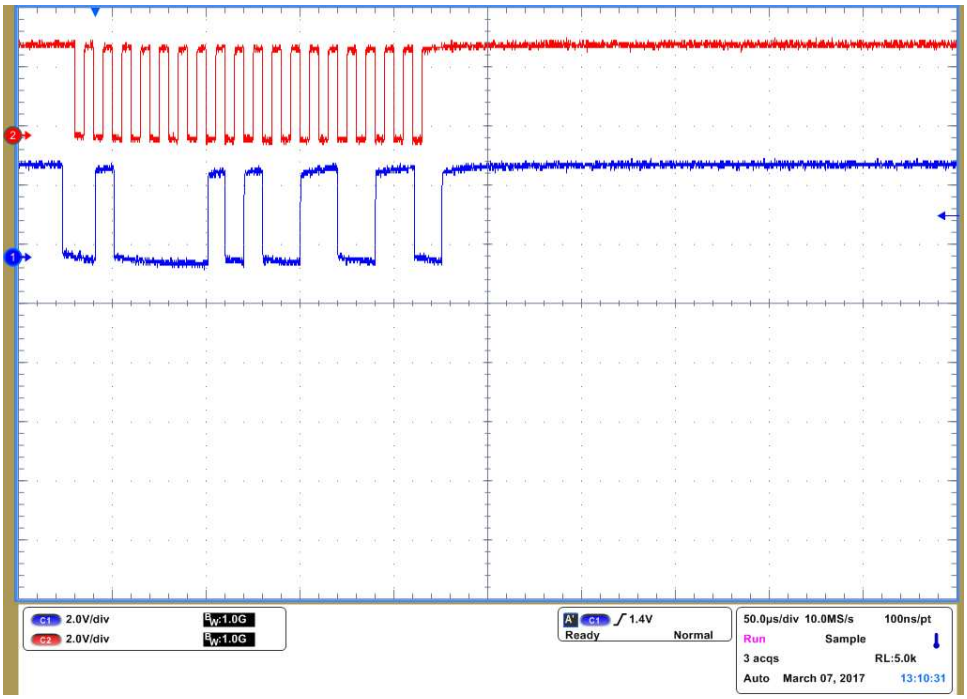


Figure 13. I²C Communication

4 Design Files

NOTE: Version E2 of the Altium CAD project, Gerber files, and PCB plot have an updated PCB component footprint for the J1 connector (SUB-D9).

4.1 Schematics

To download the schematics, see the design files at TIDA-01406.

4.2 Bill of Materials

To download the bill of materials (BOM), see the design files at TIDA-01406.

4.3 PCB Layout Recommendations

4.3.1 Layout Prints

To download the layer plots, see the design files at TIDA-01406.

4.4 Altium Project

To download the Altium project files, see the design files at TIDA-01406.

4.5 Gerber Files

To download the Gerber files, see the design files at TIDA-01406.

4.6 Assembly Drawings

To download the assembly drawings, see the design files at TIDA-01406.

5 Software Files

To download the software files, see the design files at TIDA-01406.

6 Related Documentation

1. Texas Instruments, *ISO1050 Isolated CAN Transceiver*, ISO1050 Datasheet (SLLS983)
2. Texas Instruments, *SN6501 Transformer Driver for Isolated Power Supplies*, SN6501 Datasheet (SLLSEA0)
3. Texas Instruments, *TPD2E007 2-Channel ESD Protection Array for AC-Coupled/Negative-Rail Data Interfaces*, TPD2E007 Datasheet (SLVS796)

6.1 Trademarks

7 About the Author

THOMAS MAUER is a system engineer in the Factory Automation and Control Team at Texas Instruments Freising. He is responsible for developing reference design solutions for the industrial segment. Thomas brings his extensive experience in industrial communications like Industrial Ethernet and fieldbuses and industrial applications to this role. Thomas earned his degree in electrical engineering (Dipl. Ing. (FH)) at the University of Applied Sciences in Wiesbaden, Germany

Revision History

NOTE: Page numbers for previous revisions may differ from page numbers in the current version.

Changes from Original (March 2017) to A Revision	Page
• Added note on Version E2 of the Altium CAD project, Gerber files, and PCB plot	22

IMPORTANT NOTICE FOR TI DESIGN INFORMATION AND RESOURCES

Texas Instruments Incorporated ('TI") technical, application or other design advice, services or information, including, but not limited to, reference designs and materials relating to evaluation modules, (collectively, "TI Resources") are intended to assist designers who are developing applications that incorporate TI products; by downloading, accessing or using any particular TI Resource in any way, you (individually or, if you are acting on behalf of a company, your company) agree to use it solely for this purpose and subject to the terms of this Notice.

TI's provision of TI Resources does not expand or otherwise alter TI's applicable published warranties or warranty disclaimers for TI products, and no additional obligations or liabilities arise from TI providing such TI Resources. TI reserves the right to make corrections, enhancements, improvements and other changes to its TI Resources.

You understand and agree that you remain responsible for using your independent analysis, evaluation and judgment in designing your applications and that you have full and exclusive responsibility to assure the safety of your applications and compliance of your applications (and of all TI products used in or for your applications) with all applicable regulations, laws and other applicable requirements. You represent that, with respect to your applications, you have all the necessary expertise to create and implement safeguards that (1) anticipate dangerous consequences of failures, (2) monitor failures and their consequences, and (3) lessen the likelihood of failures that might cause harm and take appropriate actions. You agree that prior to using or distributing any applications that include TI products, you will thoroughly test such applications and the functionality of such TI products as used in such applications. TI has not conducted any testing other than that specifically described in the published documentation for a particular TI Resource.

You are authorized to use, copy and modify any individual TI Resource only in connection with the development of applications that include the TI product(s) identified in such TI Resource. NO OTHER LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE TO ANY OTHER TI INTELLECTUAL PROPERTY RIGHT, AND NO LICENSE TO ANY TECHNOLOGY OR INTELLECTUAL PROPERTY RIGHT OF TI OR ANY THIRD PARTY IS GRANTED HEREIN, including but not limited to any patent right, copyright, mask work right, or other intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information regarding or referencing third-party products or services does not constitute a license to use such products or services, or a warranty or endorsement thereof. Use of TI Resources may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

TI RESOURCES ARE PROVIDED "AS IS" AND WITH ALL FAULTS. TI DISCLAIMS ALL OTHER WARRANTIES OR REPRESENTATIONS, EXPRESS OR IMPLIED, REGARDING TI RESOURCES OR USE THEREOF, INCLUDING BUT NOT LIMITED TO ACCURACY OR COMPLETENESS, TITLE, ANY EPIDEMIC FAILURE WARRANTY AND ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT OF ANY THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

TI SHALL NOT BE LIABLE FOR AND SHALL NOT DEFEND OR INDEMNIFY YOU AGAINST ANY CLAIM, INCLUDING BUT NOT LIMITED TO ANY INFRINGEMENT CLAIM THAT RELATES TO OR IS BASED ON ANY COMBINATION OF PRODUCTS EVEN IF DESCRIBED IN TI RESOURCES OR OTHERWISE. IN NO EVENT SHALL TI BE LIABLE FOR ANY ACTUAL, DIRECT, SPECIAL, COLLATERAL, INDIRECT, PUNITIVE, INCIDENTAL, CONSEQUENTIAL OR EXEMPLARY DAMAGES IN CONNECTION WITH OR ARISING OUT OF TI RESOURCES OR USE THEREOF, AND REGARDLESS OF WHETHER TI HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

You agree to fully indemnify TI and its representatives against any damages, costs, losses, and/or liabilities arising out of your non-compliance with the terms and provisions of this Notice.

This Notice applies to TI Resources. Additional terms apply to the use and purchase of certain types of materials, TI products and services. These include; without limitation, TI's standard terms for semiconductor products http://www.ti.com/sc/docs/stdterms.htm), evaluation modules, and samples (http://www.ti.com/sc/docs/sampterms.htm).

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