参考資料

TI Designs: TIDA-01406
PLCおよび通信モジュール用、エネルギー効率の高い絶縁型CANopenインターフェイス

概要

CANおよびCANopenはレガシーのフィールドバス・プロトコルで、ファクトリ・オートメーションや制御など、多くの産業用アプリケーションに使用されます。高電圧が最終機器に損傷を与える可能性がある場合、絶縁が必要になります。今日のインテリジェントな工場では多くの自動化ノードがあるため、すべてのデバイスのエネルギー消費が少なくなれば、総消費電力が削減されるという利点もあります。TIDA-01406は、PLCおよび通信モジュール用の、エネルギー効率の高い絶縁型CANopenインターフェイスのTI Designです。このデザインはBeagleBone Black Capeをベースとしており、BeagleBone Black開発ボードで簡単にテストできます。

リソース

TIDA-01406	デザイン・フォルダ
ISO1050	プロダクト・フォルダ
SN6501	プロダクト・フォルダ
TPD2E007	プロダクト・フォルダ
BeagleBone Black 開発ボード	ツール・フォルダ

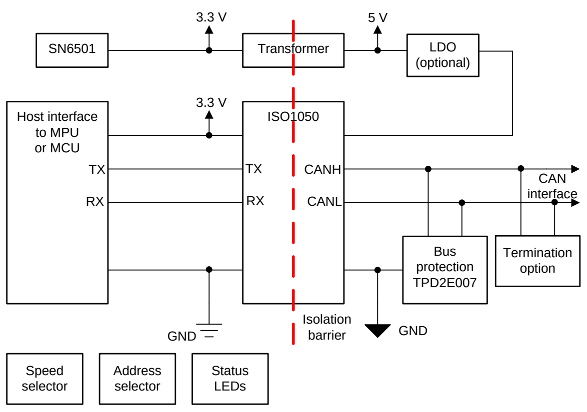
TI E2E™ Community — E2Eエキスパートに質問

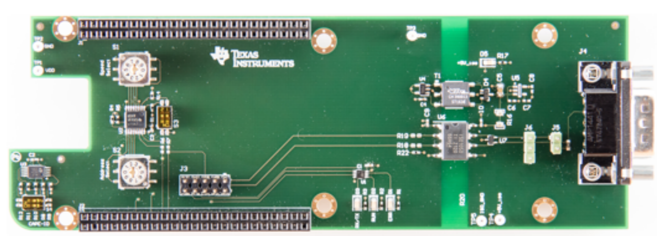
特長

- ISO1050およびSN6501によるCANトランシーバのリファレンス・デザイン
- 小さなフォーム・ファクタのデザイン
- ISO1050の絶縁:
  - DUBパッケージによる2.5kV絶縁
- BeagleBone Black Capeのフォーム・ファクタ
- BeagleBone Blackおよび汎用プロセッサ・ボード用の拡張インターフェイス
- CANバス・アドレスおよびCANバス速度のロータリー・スイッチ
- CANステータスLED

アプリケーション

- 通信モジュール
- PLC





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.

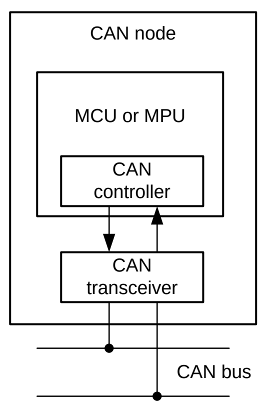


図 1. CAN Node

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

表 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

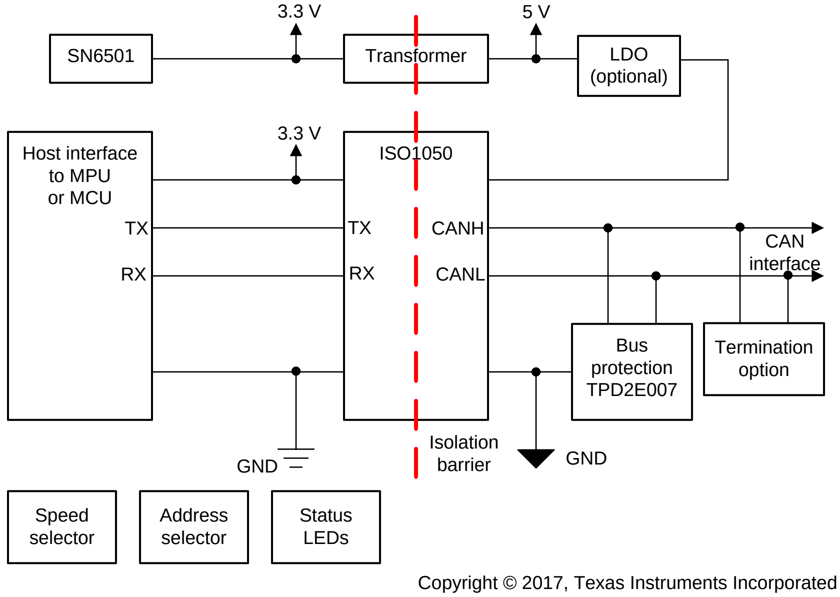


図 2. CANopen Block Diagram

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. 表 2 specifies the pins and signals used by the CANopen cape:

表 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.

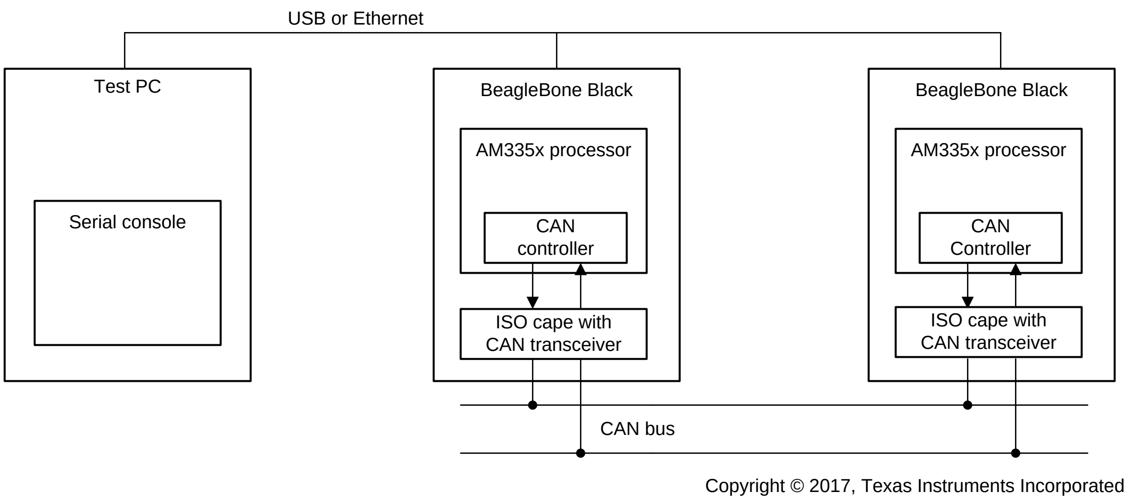


図 3. Test Setup

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 図 4:

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

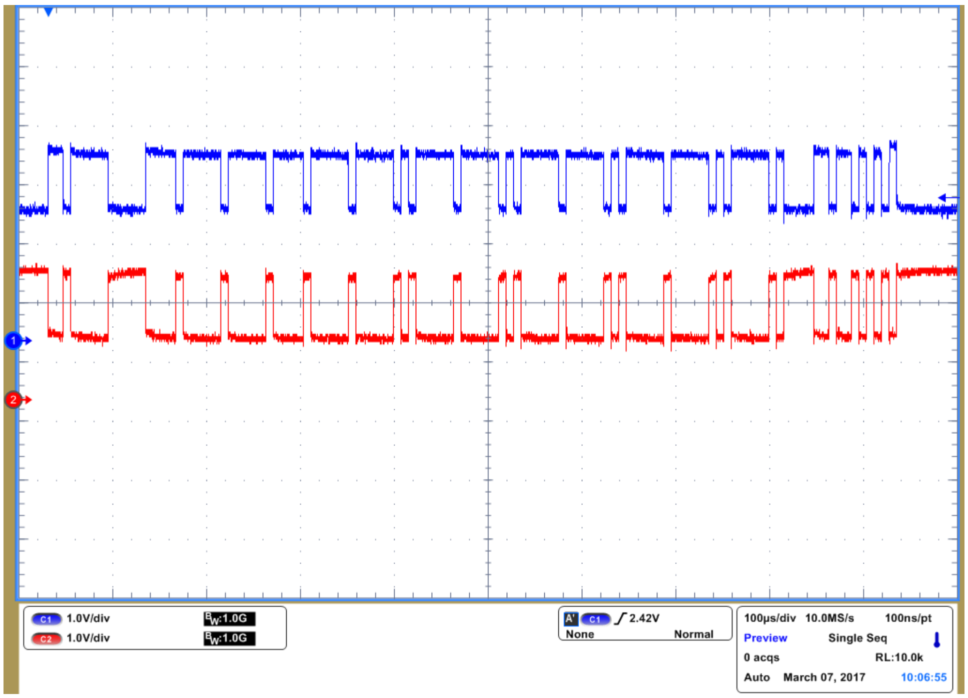


図 4. 125-kHz Frame

For 図 5:

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

図 5. 125-kHz Zoom

For 図 6:

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

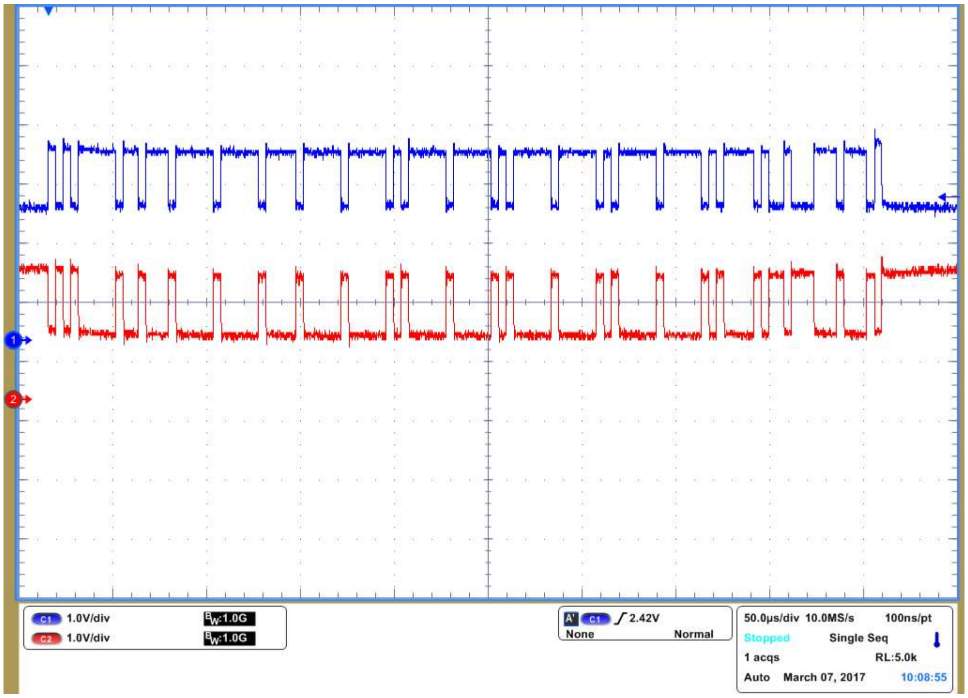


図 6. 250-kHz Frame

For 図 7:

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

図 7. 250-kHz Zoom

For 図 8:

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

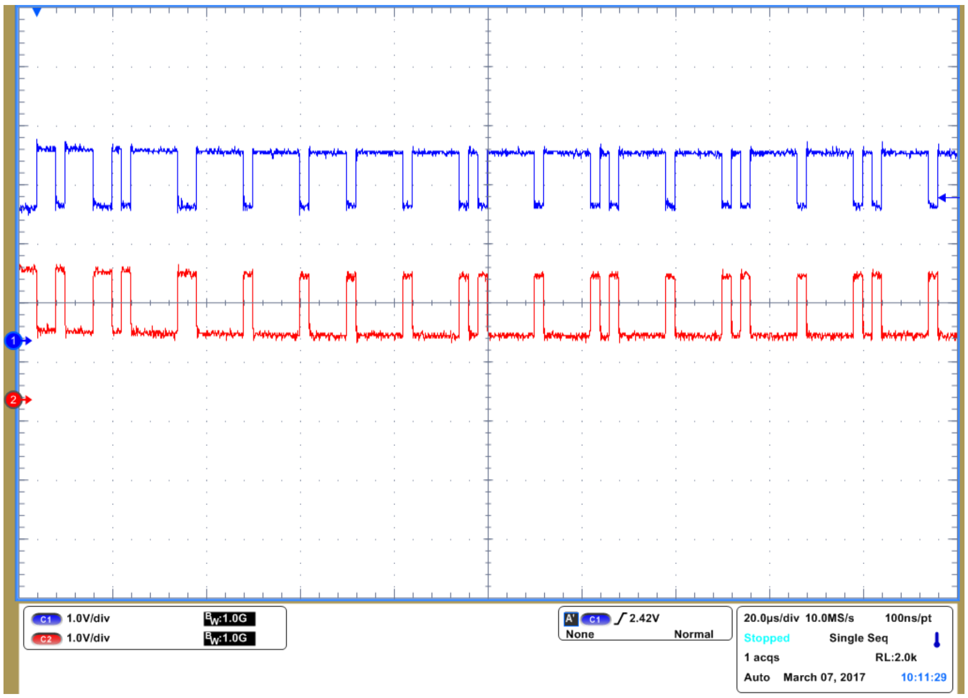


図 8. 500-kHz Frame

For 図 9:

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

図 9. 500-kHz Zoom

For 図 10:

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

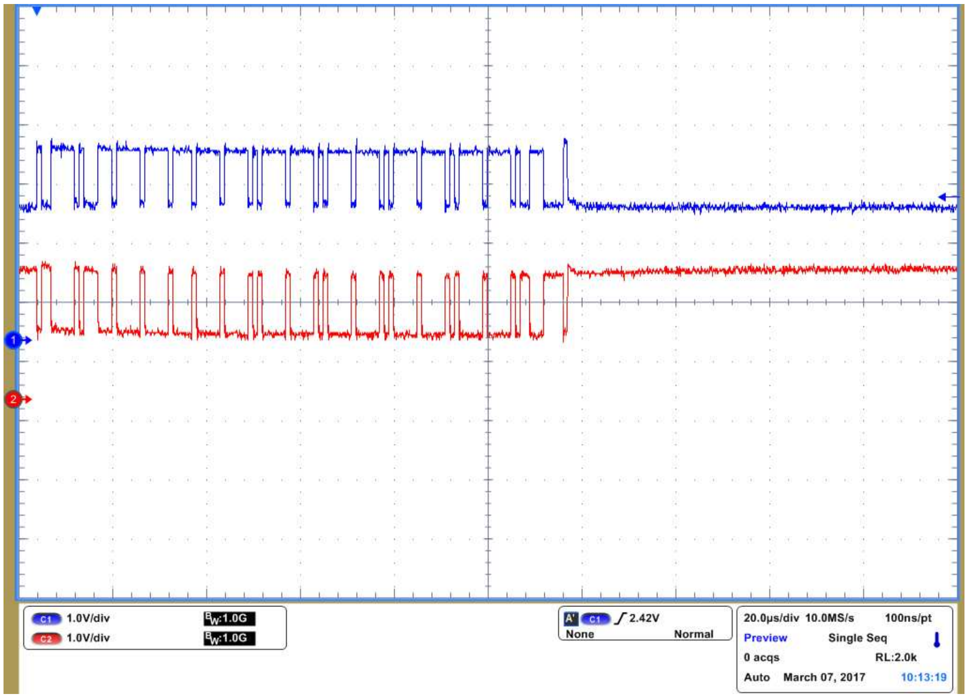


図 10. 1-MHz Frame

For 図 11:

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

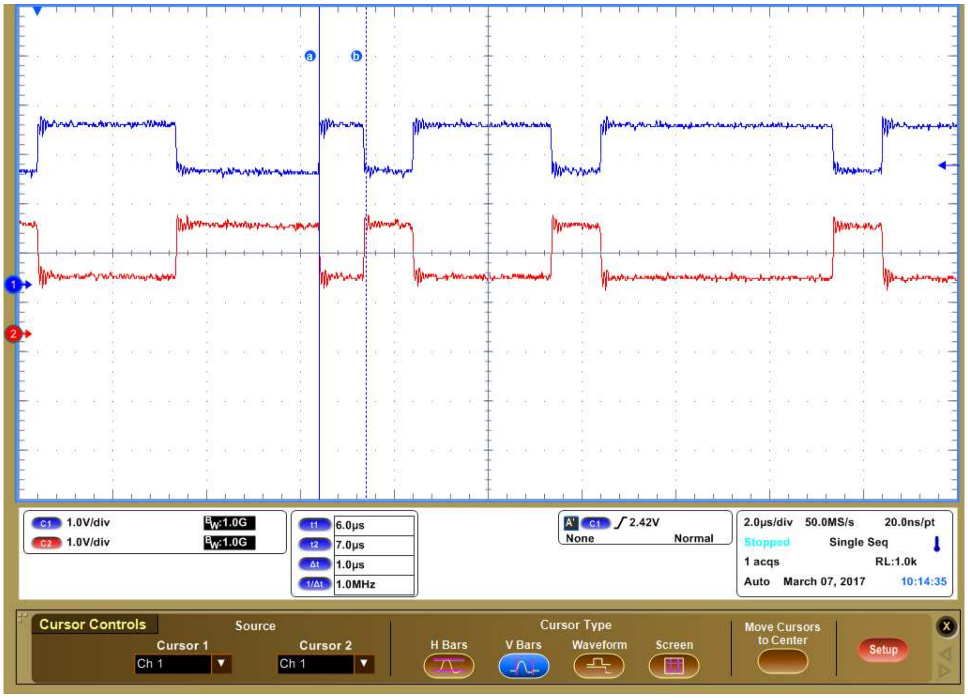


図 11. 1-MHz Zoom

3.2.4.2 Isolated Power

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

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

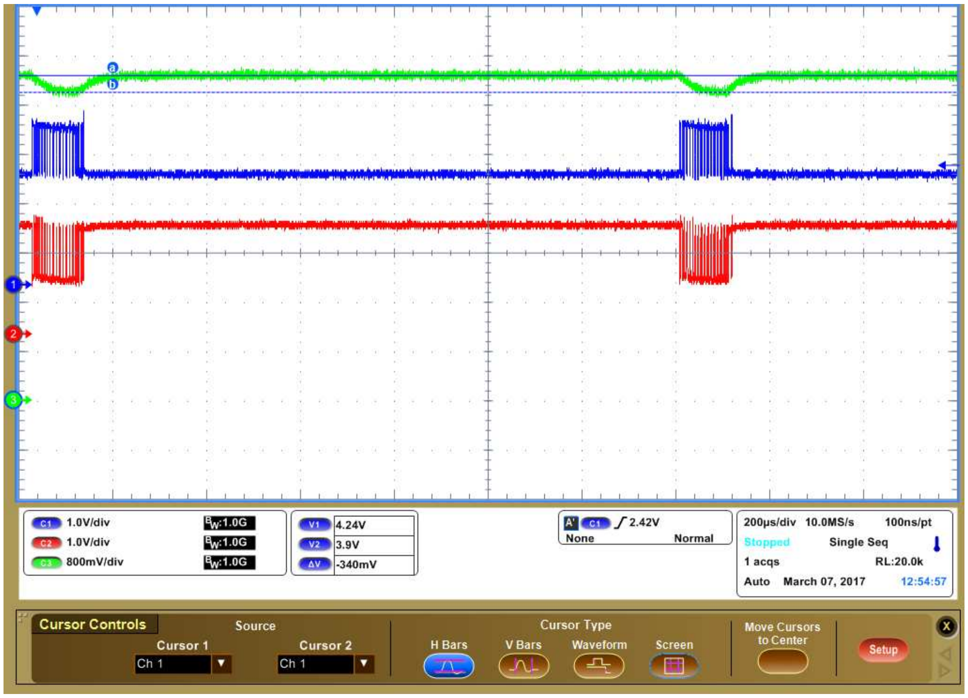


図 12. ISO Supply

3.2.4.3 Position Switch Readout

図 13 shows the test results for I²C interface:

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

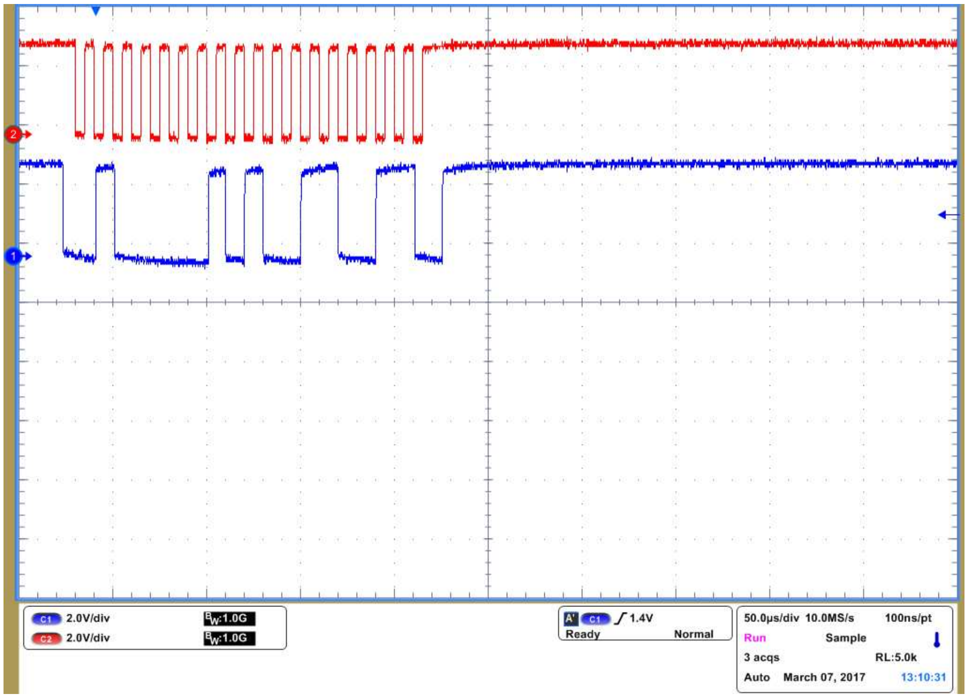


図 13. I²C Communication

4 Design Files

> **注:** 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 商標

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

改訂履歴

資料番号末尾の英字は改訂を表しています。その改訂履歴は英語版に準じています。

2017年**3**月発行のものから更新	**Page**
• note on Version E2 of the Altium CAD project, Gerber files, and PCB plot 追加	22

TIの設計情報およびリソースに関する重要な注意事項

Texas Instruments Incorporated ("TI")の技術、アプリケーションその他設計に関する助言、サービスまたは情報は、TI製品を組み込んだアプリケーションを開発する設計者に役立つことを目的として提供するものです。これにはリファァレンス設計や、評価モジュールに関係する資料が含まれますが、これらに限られません。以下、これらを総称して「TIリソース」と呼びます。いかなる方法であっても、TIリソースのいずれかをダウンロード、アクセス、または使用した場合、お客様(個人、または会社を代表している場合にはお客様の会社)は、これらのリソースをここに記載された目的にのみ使用し、この注意事項の条項に従うことに合意したものとします。

TIによるTIリソースの提供は、TI製品に対する該当の発行済み保証事項または免責事項を拡張またはいかなる形でも変更するものではなく、これらのTIリソースを提供することによって、TIにはいかなる追加義務も責任も発生しないものとします。TIは、自社のTIリソースに訂正、拡張、改良、およびその他の変更を加える権利を留保します。

お客様は、自らのアプリケーションの設計において、ご自身が独自に分析、評価、判断を行う責任がお客様にあり、お客様のアプリケーション(および、お客様のアプリケーションに使用されるすべてのTI製品)の安全性、および該当するすべての規制、法、その他適用される要件への遵守を保証するすべての責任をお客様のみが負うことを理解し、合意するものとします。お客様は、自身のアプリケーションに関して、(1) 故障による危険な結果を予測し、(2) 障害とその結果を監視し、および、(3) 損害を引き起こす障害の可能性を減らし、適切な対策を行う目的での、安全策を開発し実装するために必要な、すべての技術を保持していることを表明するものとします。お客様は、TI製品を含むアプリケーションを使用または配布する前に、それらのアプリケーション、およびアプリケーションに使用されているTI製品の機能性を完全にテストすることに合意するものとします。TIは、特定のTIリソース用に発行されたドキュメントで明示的に記載されているもの以外のテストを実行していません。

お客様は、個別のTIリソースにつき、当該TIリソースに記載されているTI製品を含むアプリケーションの開発に関連する目的でのみ、使用、コピー、変更することが許可されています。明示的または黙示的を問わず、禁反言の法理その他どのような理由でも、他のTIの知的所有権に対するその他のライセンスは付与されません。また、TIまたは他のいかなる第三者のテクノロジまたは知的所有権についても、いかなるライセンスも付与されるものではありません。付与されないものには、TI製品またはサービスが使用される組み合わせ、機械、プロセスに関連する特許権、著作権、回路配置利用権、その他の知的所有権が含まれますが、これらに限られません。第三者の製品やサービスに関する、またはそれらを参照する情報は、そのような製品またはサービスを利用するライセンスを構成するものではなく、それらに対する保証または推奨を意味するものでもありません。TIリソースを使用するため、第三者の特許または他の知的所有権に基づく第三者からのライセンス、あるいはTIの特許または他の知的所有権に基づくTIからのライセンスが必要な場合があります。

TIのリソースは、それに含まれるあらゆる欠陥も含めて、「現状のまま」提供されます。TIは、TIリソースまたはその仕様に関して、明示的か暗黙的かにかかわらず、他のいかなる保証または表明も行いません。これには、正確性または完全性、権原、続発性の障害に関する保証、および商品性、特定目的への適合性、第三者の知的所有権の非侵害に対する黙示の保証が含まれますが、これらに限られません。

TIは、いかなる苦情に対しても、お客様への弁護または補償を行う義務はなく、行わないものとします。これには、任意の製品の組み合わせに関連する、またはそれらに基づく侵害の請求も含まれますが、これらに限られず、またその事実についてTIリソースまたは他の場所に記載されているか否かを問わないものとします。いかなる場合も、TIリソースまたはその使用に関連して、またはそれらにより発生した、実際的、直接的、特別、付随的、間接的、懲罰的、偶発的、または、結果的な損害について、そのような損害の可能性についてTIが知らされていたかどうかにかかわらず、TIは責任を負わないものとします。

お客様は、この注意事項の条件および条項に従わなかったために発生した、いかなる損害、コスト、損失、責任からも、TIおよびその代表者を完全に免責するものとします。

この注意事項はTIリソースに適用されます。特定の種類の資料、TI製品、およびサービスの使用および購入については、追加条項が適用されます。これには、半導体製品(http://www.ti.com/sc/docs/stdterms.htm)、評価モジュール、およびサンプル(http://www.ti.com/sc/docs/sampterms.htm)についてのTIの標準条項が含まれますが、これらに限られません。