Chipsmall Limited consists of a professional team with an average of over 10 year of expertise in the distribution of electronic components. Based in Hongkong, we have already established firm and mutual-benefit business relationships with customers from,Europe,America and south Asia,supplying obsolete and hard-to-find components to meet their specific needs.

With the principle of “Quality Parts,Customers Priority,Honest Operation,and Considerate Service”,our business mainly focus on the distribution of electronic components. Line cards we deal with include Microchip,ALPS,ROHM,Xilinx,Pulse,ON,Everlight and Freescale. Main products comprise IC,Modules,Potentiometer,IC Socket,Relay,Connector.Our parts cover such applications as commercial,industrial, and automotives areas.

We are looking forward to setting up business relationship with you and hope to provide you with the best service and solution. Let us make a better world for our industry!

Contact us

Tel: +86-755-8981 8866 Fax: +86-755-8427 6832
Email & Skype: info@chipsmall.com Web: www.chipsmall.com
Address: A1208, Overseas Decoration Building, #122 Zhenhua RD., Futian, Shenzhen, China

CY8CKIT-017 CAN/LIN Expansion Board Kit Guide

Doc. # 001-57814 Rev. *D

Cypress Semiconductor
198 Champion Court
San Jose, CA 95134-1709
Phone (USA): 800.858.1810
Phone (Intnl): 408.943.2600
http://www.cypress.com

Copyrights

Flash Code Protection

Cypress products meet the specifications contained in their particular Cypress PSoC Data Sheets. Cypress believes that its family of PSoC products is one of the most secure families of its kind on the market today, regardless of how they are used. There may be methods, unknown to Cypress, that can breach the code protection features. Any of these methods, to our knowledge, would be dishonest and possibly illegal. Neither Cypress nor any other semiconductor manufacturer can guarantee the security of their code. Code protection does not mean that we are guaranteeing the product as "unbreakable."

Cypress is willing to work with the customer who is concerned about the integrity of their code. Code protection is constantly evolving. We at Cypress are committed to continuously improving the code protection features of our products.

Contents

Safety Information

Regulatory Compliance

The CY8CKIT-017 is intended for use as a development platform for hardware or software in a laboratory environment. The board is an open system design, which does not include a shielded enclosure. This may cause interference to other electrical or electronic devices in close proximity.

In a domestic environment, this product may cause radio interference. In this case, the user may be required to take adequate prevention measures. Also, the board should not be used near any medical equipment or RF devices.

Attaching additional wiring to this product or modifying the product operation from the factory default may affect its performance and cause interference with other apparatus in the immediate vicinity. If such interference is detected, suitable mitigating measures should be taken.

The CY8CKIT-017 as shipped from the factory has been verified to meet with requirements of CE as a Class A product.

The CY8CKIT-017 contains electrostatic discharge (ESD) sensitive devices. Electrostatic charges readily accumulate on the human body and any equipment, and can discharge without detection. Permanent damage may occur on devices subjected to high-energy discharges. Proper ESD precautions are recommended to avoid performance degradation or loss of functionality. Store unused CY8CKIT-017 boards in the protective shipping package.

End-of-Life/Product Recycling

The end of life for this kit is five years from the date of manufacture, mentioned on the back of the box. Contact your nearest recycler for information on how to disposition the kit.

General Safety Instructions

ESD Protection

ESD can damage boards and associated components. Cypress recommends that you perform procedures only at an ESD workstation. If one is not available, use appropriate ESD protection by wearing an antistatic wrist strap attached to chassis ground (any unpainted metal surface) on your board when handling parts.

Handling Boards

CY8CKIT-017 boards are sensitive to ESD. Hold the board only by its edges. After removing the board from its box, place it on a grounded, static free surface. Use a conductive foam pad if available. Do not slide board over any surface.

1. Introduction

The CY8CKIT-017 CAN/LIN Expansion Board Kit (EBK) is an expansion board that is used with the CY8CKIT-001 PSoC® Development Kit (DVK) or the CY8CKIT-030 PSoC 3 Development Kit (DVK). It enables you to evaluate the Controller Area Network (CAN) and Local Interconnect Network (LIN) slave communication capability of PSoC 3 and PSoC 5 devices. You can design your own projects with an easy-to-use CAN and LIN slave components in Cypress's PSoC Creator™ software, or by altering code examples provided with this kit.

1.1 Kit Contents

This kit contains:

- CAN/LIN expansion board
- Quick start guide
- Kit CD

Inspect the contents of the kit; if anything is missing, visit http://www.cypress.com/?rID=40215 or contact Cypress Technical Support.

1.2 Kit Compatibility

This kit contains an expansion board and requires other Cypress kits to use it. It is designed to add CAN and LIN capabilities to the CY8CKIT-001 PSoC Development Kit (DVK). This DVK supports PSoC 1, PSoC 3, and PSoC 5 families. However, it may be necessary to obtain or purchase additional processor modules for the CY8CKIT-001 to develop applications for a particular PSoC device family.

This kit is also compatible with the CY8CKIT-030 PSoC 3 Development Kit. The EBK can be attached to port E of the CY8CKIT-030 DVK. A CY8CKIT-030 kit can generally be substituted for a CY8CKIT-001 kit when using the CY8CKIT-017 kit. Therefore, any information regarding the CY8CKIT-001 kit in this document generally applies to the CY8CKIT-030 kit as well.

The CY8CKIT-017 can also interface with the CY3280-22x45 Universal CapSense Controller (UCC) kit for CY8C2xx45 PSoC 1 devices. This EBK can add LIN capabilities to the UCC kit. However, it does not add CAN capabilities to this kit, because PSoC 1 devices do not have CAN hardware.

CAN and LIN are communication protocols and require more than one CAN or LIN node to set up communication between nodes. Therefore, it is recommended to have two CY8CKIT-001 DVKs and two CY8CKIT-017 EBKs. This enables you to set up CAN or LIN communication between two CAN or LIN nodes. An alternative recommendation is to have a CAN or LIN bus emulator or analyzer. This enables you to emulate a CAN or LIN node to communicate with a PSoC CAN or LIN controller. See sections Using a CAN Bus Analyzer Tool on page 22 and Using a LIN Bus Analyzer Tool on page 22 for more details on using a CAN analyzer or LIN analyzer tool with this kit.

For detailed information about the differences between PSoC 1, PSoC 3, and PSoC 5 devices, go to http://www.cypress.com/psoc. For more information about Cypress' kits, go to the Cypress Store at http://www.cypress.com/shop.

1.3 PSoC Creator

Cypress' PSoC Creator software is a state-of-the-art, easy-to-use software Integrated Development Environment (IDE). It introduces a hardware and software co-design environment based on classical schematic entry and revolutionary embedded design methods.

With PSoC Creator, you can:

- Create and share user defined, custom peripherals using hierarchical schematic design.
- Automatically place and route select components and integrate simple glue logic that is normally present in discrete muxes.
- Trade-off hardware and software design considerations allowing you to focus on what matters and get to market faster.

PSoC Creator also enables you to tap into an entire tools ecosystem with integrated compiler tool chains, RTOS solutions, and production programmers to support both PSoC 3 and PSoC 5.

1.4 Getting Started

To get started, see the Kit Operation chapter on page 13 for a description of the kit operation and how to program the PSoC 3 device. Code examples are used to explain how to use the CAN/LIN expansion board with the CY8CKIT-001 DVK. The Hardware chapter on page 23 provides details of the expansion board hardware. The Code Examples chapter on page 31 guides you to create simple code examples. The Appendix chapter on page 61 provides the schematics and bill of materials (BOM) associated with the expansion board.

1.5 Additional Learning Resources

Visit http://www.cypress.com for additional learning resources in the form of datasheets, technical reference manuals, and application notes.

Application Note AN52701 describes the implementation of CAN bus communication between two PSoC devices. It explains how to send and receive CAN messages and handle error messages.

1.6 Technical Support

If you have any technical questions or issues related to this kit, call Cypress Customer Support

+1 (800) 541-4736 Ext. 8 (in the USA),

+1 (408) 943-2600 Ext. 8 (International),

or visit http://www.cypress.com/go/support

1.7 Document History

Revision	Release Date	Description of Change
**	01/27/2010	Initial version of the guide
*A	02/12/2010	CDT based Updates
*B	02/14/2011	Added hyperlinks in Introduction chapter on page 7. Many minor text edits. Updated to include information about the kit's CD and the CY8CKIT-030 kit.
*C	12/16/2011	Updated images for PSoC Creator version (2.0)
*D	09/11/2012	Added a code example for the LIN component and included updates related to LIN throughout the document. Added Safety Information chapter on page 5 and Regulatory Compliance Information on page 62.

1.8 Document Conventions

Convention	Usage
Courier New	Displays file locations, user entered text, and source code: C:\ ...cd\icc\
Italics	Displays file names and reference documentation: Read about the *sourcefile.hex* file in the *PSoC Designer User Guide*.
[**Bracketed, Bold**]	Displays keyboard commands in procedures: [**Enter**] or [**Ctrl**] [**C**]
File > Open	Represents menu paths: File > Open > New Project
Bold	Displays commands, menu paths, and icon names in procedures: Click the **File** icon and then click **Open**.
Times New Roman	Displays an equation: 2 + 2 = 4
Text in gray boxes	Describes Cautions or unique functionality of the product.

2. Installation

2.1 Prerequisite Software

The CY8CKIT-017 CAN/LIN EBK requires the PSoC Programmer and PSoC Creator software programs to be installed before the kit can be used. If you want to use this EBK to develop applications for PSoC 1 devices, then use the PSoC Designer software. You can install these programs from the CY8CKIT-001 kit CD, this kit's CD, or by downloading them from the following locations:

- PSoC Programmer: http://www.cypress.com/psocprogrammer
- PSoC Creator: http://www.cypress.com/creator
- PSoC Designer: http://www.cypress.com/psocdesigner

2.2 Software Installation

2.2.1 Installation from CD

Follow these steps to install the CY8CKIT-017 CAN/LIN EBK software from the kit's CD:

1. Insert the kit CD into your computer.
2. On the autorun screen that appears, choose the **Install the kit contents from CD** option to install the kit contents from the CD.

   Note If the Autorun screen does not appear, go to **My Computer**, open the drive containing the kit CD. Click **CAN-LIN EBK > Run CANLINEBKSetup.exe**.
3. Follow all on-screen prompts to proceed with the installation. When installing the kit software, the installer checks if the prerequisite software is installed in your system. These include PSoC Creator, PSoC Programmer, Windows Installer, .NET, Acrobat Reader, and Keil Complier. If these applications are not installed, the installer prompts you to download and/or install them.

2.2.2 Installation from Internet

Follow these steps to install the CY8CKIT-017 CAN/LIN EBK software from the internet (this is done to ensure that the latest software is installed):

1. Insert the kit CD into your computer.
2. Choose the **Install the latest kit contents from web** option on the auto run screen. This directs you to http://www.cypress.com/?rID=40215, where you can download the latest installer.
3. Download the installer executable file.
4. Run the installer executable file after it is downloaded.
5. Follow all on-screen prompts to proceed with the installation. When installing the kit software, the installer checks if the prerequisite software is installed in your system. These include PSoC Creator, PSoC Programmer, Windows Installer, .NET, Acrobat Reader, and Keil Complier. If these applications are not installed, the installer prompts you to download and/or install them.

Note All kit contents are found on the kit CD. Also, after the installation is complete, the kit contents are found at the following location:

```
C:\Program Files\Cypress\CY8CKIT-017 CAN-LIN Expansion Board Kit\
```

2.3 Software Uninstallation

Follow these steps to uninstall the CY8CKIT-017 CAN/LIN EBK software:

1. Open the Cypress Update Manager program. This is a program that is installed along with other Cypress software.
2. Click the **Uninstall** button associated with the CY8CKIT-017 kit software.
3. Follow the on-screen prompts to uninstall the software.

The software can also be uninstalled by using the **Add**/**Remove Programs** tool included with Windows.

2.4 Hardware Installation

Follow these steps to install the hardware:

1. The EBK board must be physically attached to port A of the CY8CKIT-001 DVK.

   Note This document explains how to use this EBK with port A of the CY8CKIT-001 DVK. You can also attach this board to port B or port C of CY8CKIT-001 DVK and port E of CY8CKIT-030 DVK. You need to modify pin assignments of the example projects to use with other ports. See Port Options with CY8CKIT-001 DVK on page 28 for pin assignment details and limitations of using with CY8CKIT-001 DVK ports. See Table 5-1 and Table 5-2 for pin assignment details with port E of CY8CKIT-030 DVK.
2. A MiniProg3 device programmer must be connected to your computer to program this kit's code examples into the PSoC devices. Follow the instructions in the MiniProg3 kit documentation to connect it to your computer.
3. For the CAN example testing:
   a. Connect the two CAN/LIN EBK boards together with a male-to-male, 9-pin, RS-232 cable with "straight-through" connections, as shown in Figure 3-8 on page 17, or
   b. Connect a CAN analyzer to P2 of the EBK.
4. For the LIN example test, connect VCC, LIN bus, and GND of the LIN analyzer to connector P5 of the EBK.

3. Kit Operation

3.1 Introduction

The CY8CKIT-017 CAN/LIN EBK includes two CAN code examples and one LIN code example. The CAN code examples demonstrate two-way CAN communication between two PSoC 3 CAN controller nodes on a CAN bus. These examples must be downloaded into two separate PSoC 3 devices.

- **Code Example 1: CAN_Example_1**

  This project demonstrates sending and receiving of CAN messages. The project sends an 8-bit value on the CAN bus and also receives an 8-bit value from the CAN bus.

- **Code Example 2: CAN_Example_2**

  This project is identical to the 'CAN_Example_1' project, except the CAN message IDs are reversed. Therefore, this project implements a CAN node that can communicate with a PSoC 3 that is programmed with the 'CAN_Example_1' project. See Code Examples on page 31 for more information.

  Note Most of the information related to CAN code examples describes kit operation when two CYC8KIT-001 DVKs and two CY8CKIT-017 EBKs are available. If only one CY8CKIT-001 DVK and one CY8CKIT-017 EBK are available, see Using a CAN Bus Analyzer Tool on page 22 for information on alternative ways of using this kit.

- **Code Example 3: LIN_Example**

  This project demonstrates the LIN slave functionality of PSoC 3. It receives an unconditional frame having eight bytes of data from the LIN bus with frame ID 0x10. The eight bytes are: Byte 1 is the scalar signal of 7-bit length and Byte 2 to 8 are the byte array signal of 7-byte length. This is written to another unconditional frame. The LIN master can read back the data by sending the frame ID 0x11.

  Because the LIN master component is not yet available with PSoC 3, a LIN analyzer is required to test this project. See Using a LIN Bus Analyzer Tool on page 22 for more information.

3.2 Programming PSoC 3 Device

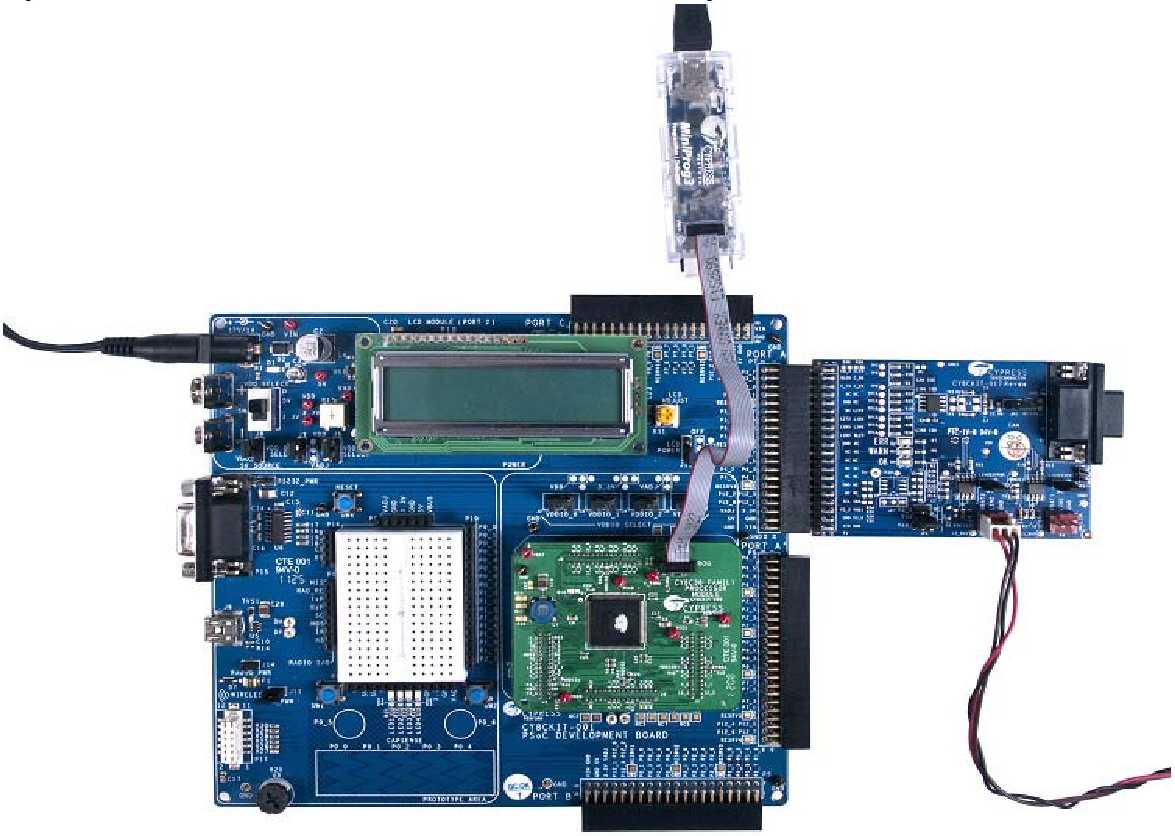
The code examples are provided on the Start Page of PSoC Creator after the CY8CKIT-017 kit contents are installed. This section provides details on programming the PSoC 3 device.

To program the 'CAN_Example_1' project to the PSoC 3 silicon, follow these steps:

1. Place the PSoC 3 processor module on the CY8CKIT-001 DVK.
2. Power the DVK using either battery connections or a wall power unit.
3. Connect the MiniProg3 JTAG cable to the JTAG connector, both on MiniProg3 and the PSoC 3 processor module. Connect the MiniProg3 to a host PC USB high speed port using a USB cable.

The connections for steps 1 to 3 are shown in Figure 3-1.

Figure 3-1. PSoC 3 Processor Module, Power, and MiniProg3 Connection with CY8CKIT-001 DVK



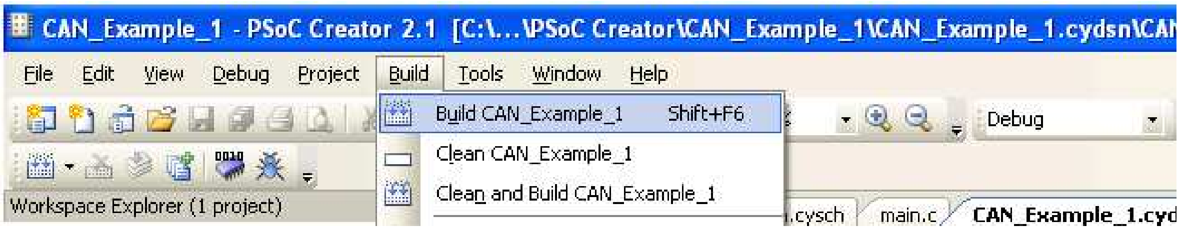
Note See the *PSoC Development Kit Board Guide* for details on connecting and programming PSoC devices.

4. Click on the code example, *CAN_Example_1* located in **Examples and Kits** on the Start Page of PSoC Creator.

Figure 3-2. PSoC Creator Start Page

5. Create a folder in the desired location and click **OK**. The project opens in PSoC Creator and is saved in that folder.
6. Build the project by selecting the **Build** option.

Figure 3-3. Build Project

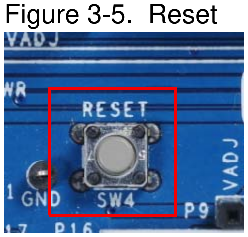


7. Click the **Program** icon.

Figure 3-4. Program Option

8. The project is programmed successfully.
9. Reset the device by pressing the SW4 switch on the DVK; see Figure 3-5.

Figure 3-5. Reset



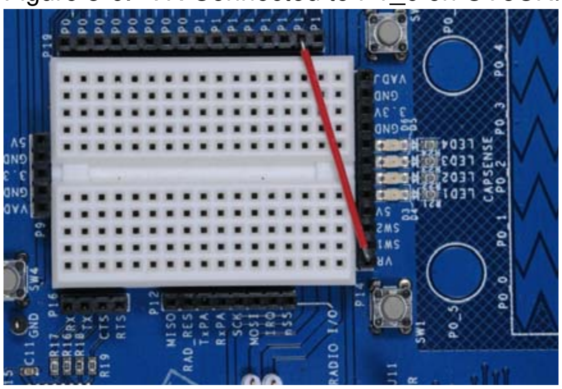
10. Follow steps 1 through 9 to program other code examples, CAN_Example_2 and LIN_Example, on PSoC 3.

3.3 Hardware Connections

3.3.1 CAN Communication Hardware Setup

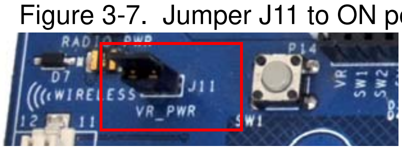
1. Connect the CAN/LIN expansion board to port A of CY8CKIT-001 DVK, as shown in Figure 3-1.
2. Connect a second CY8CKIT-017 expansion board to port A of a second CY8CKIT-001 DVK, as shown in Figure 3-1.
3. On both CY8CKIT-001 DVK boards, connect the analog input from the potentiometer (VR slot in CY8CKIT-001 DVK) to the P1_6 on the DVK, as shown in Figure 3-6.

Figure 3-6. VR Connected to P1_6 on CY8CKIT-001 DVK



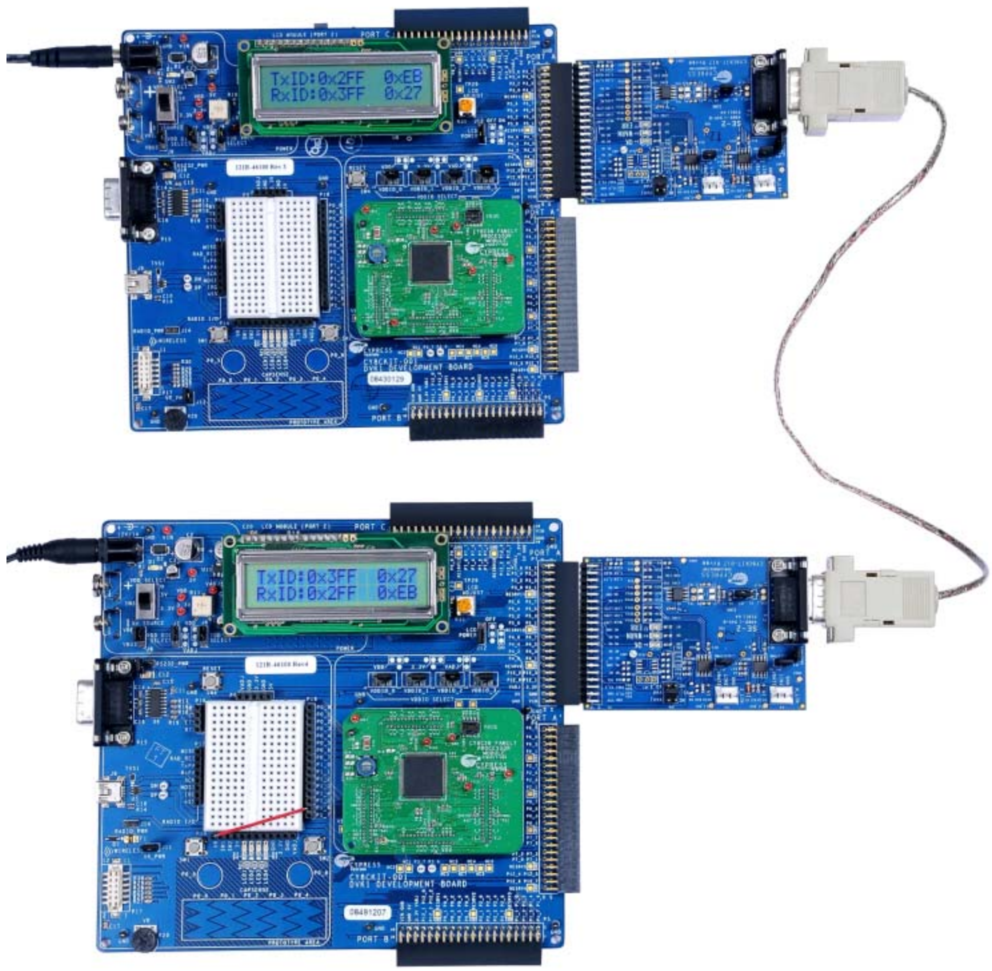
4. On both DVK boards, power the VR by setting the jumper J11 to ON position.

Figure 3-7. Jumper J11 to ON position on CY8CKIT-001 DVK



5. The remaining jumper settings on both DVKs should be in the default state. See the *PSoC Development Kit Board Guide* for the default setting of jumpers.
6. Connect the two CAN/LIN EBK boards together with a male-to-male, 9-pin, RS-232 cable with "straight-through" connections, as shown in Figure 3-8. Connect the cable to the CAN DB9 connector (P2) on both EBK boards.

Figure 3-8. Connected CAN/LIN EBK Boards

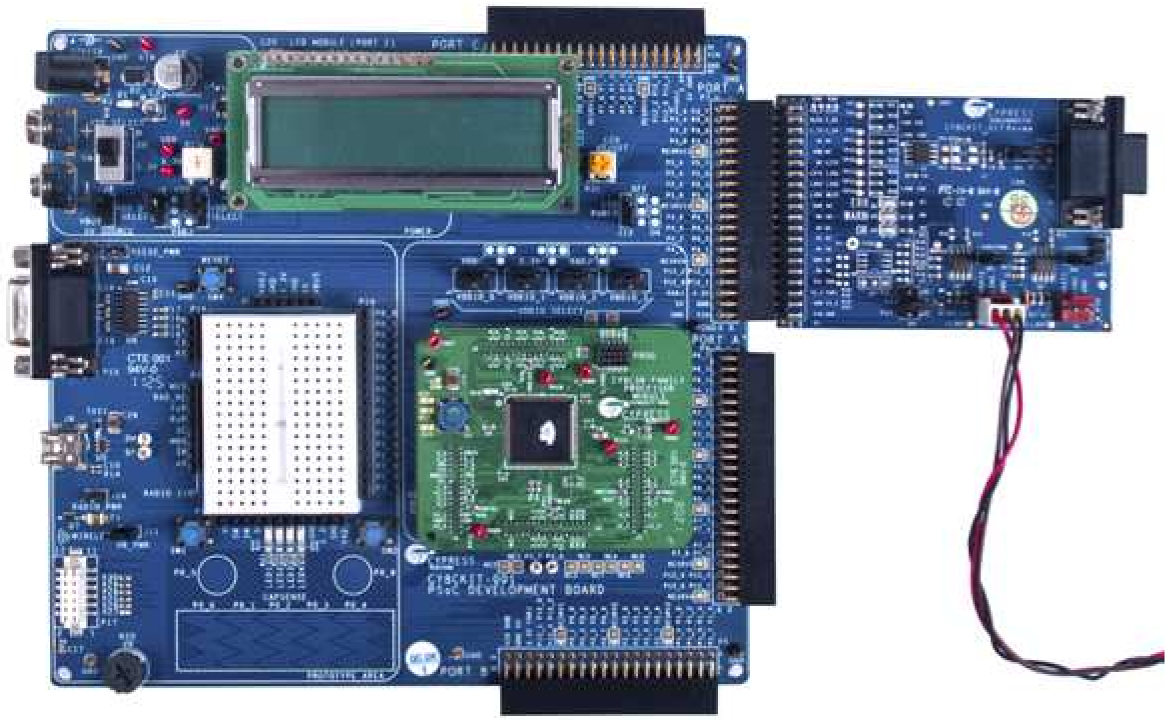


7. Power up one DVK board with the 12-V power supply. Then, power up the other DVK with a 12-V power supply. The second DVK must be powered up within approximately 5 seconds of powering the first DVK.

3.3.2 LIN Communication Hardware Setup

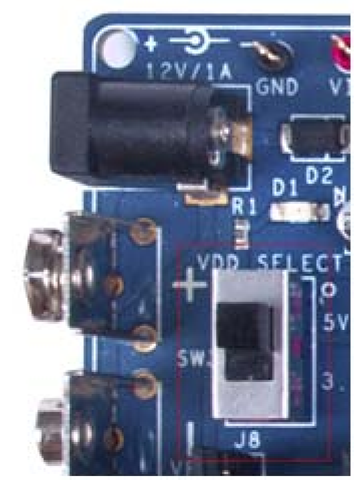
1. Connect the CAN/LIN expansion board to port A of CY8CKIT-001 DVK, as shown in Figure 3-9.

Figure 3-9. EBK to Port A of CY8CKIT-001 DVK



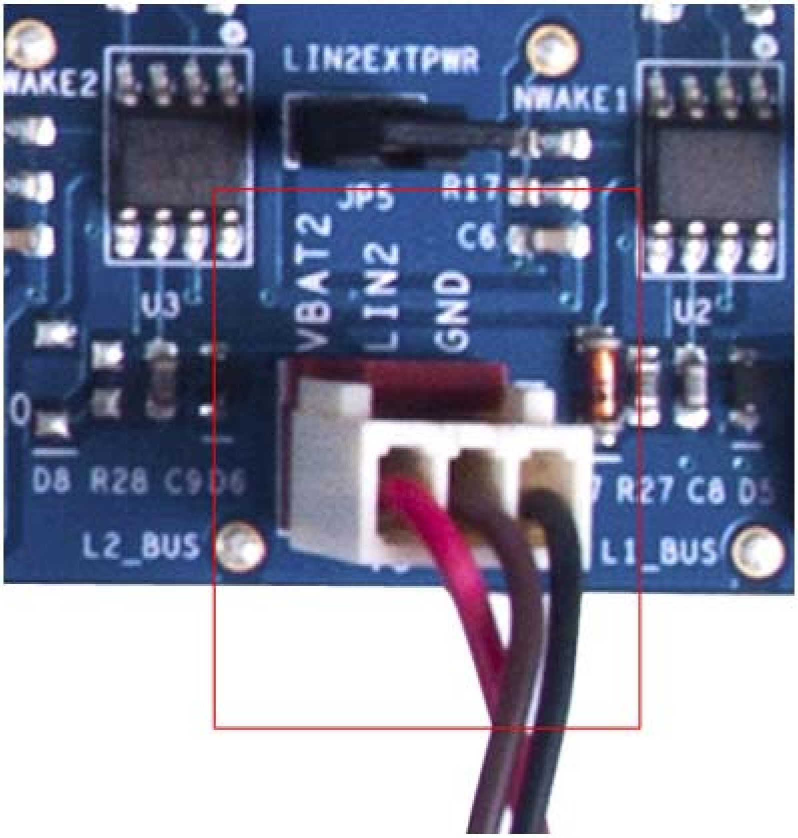
2. Set VDD SELECT switch (SW3) on the DVK to the 5 V position. The remaining jumper settings on the DVK must be set to or left at the default state.

Figure 3-10. VDD Select



3. Connect Vbat, LIN bus, and GND of the LIN analyzer/LIN master to S_LIN (P5) connector of EBK. See LIN Bus Connectors on page 26 for details of LIN bus connectors.

Figure 3-11. Connect to S_LIN



4. Connect a 12-V power supply adapter, which is supplied along with the CY8CKIT-001 DVK to the power jack of the DVK.

Note If you are using the CY8CKIT-030 PSoC 3 DVK, connect the CY8CKIT-017 EBK to port E.

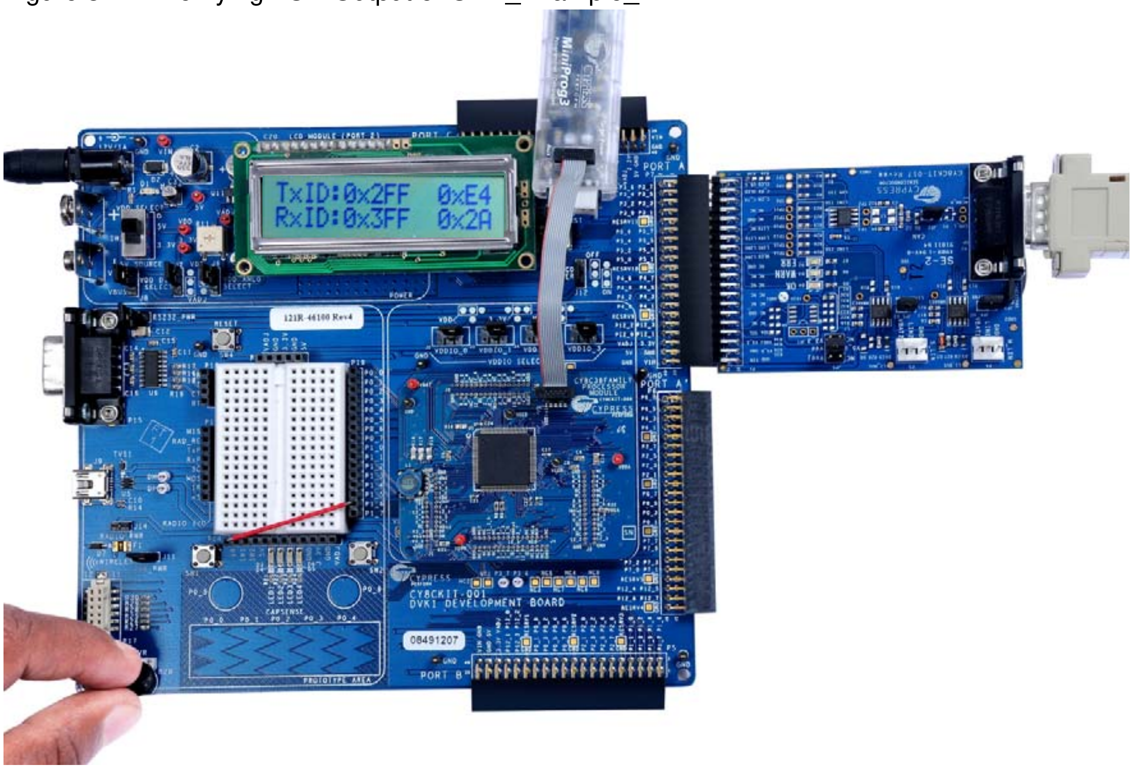
3.4 Verify Functionality

3.4.1 CAN Communication

Vary the VR (potentiometer on either DVK) and note the change in status displayed on the LCD of either DVK.

Note The CAN communication may not work correctly if the PSoC devices are not using an accurate, external clock source. See CAN Bus Clock Accuracy on page 24 for details on oscillator requirements.

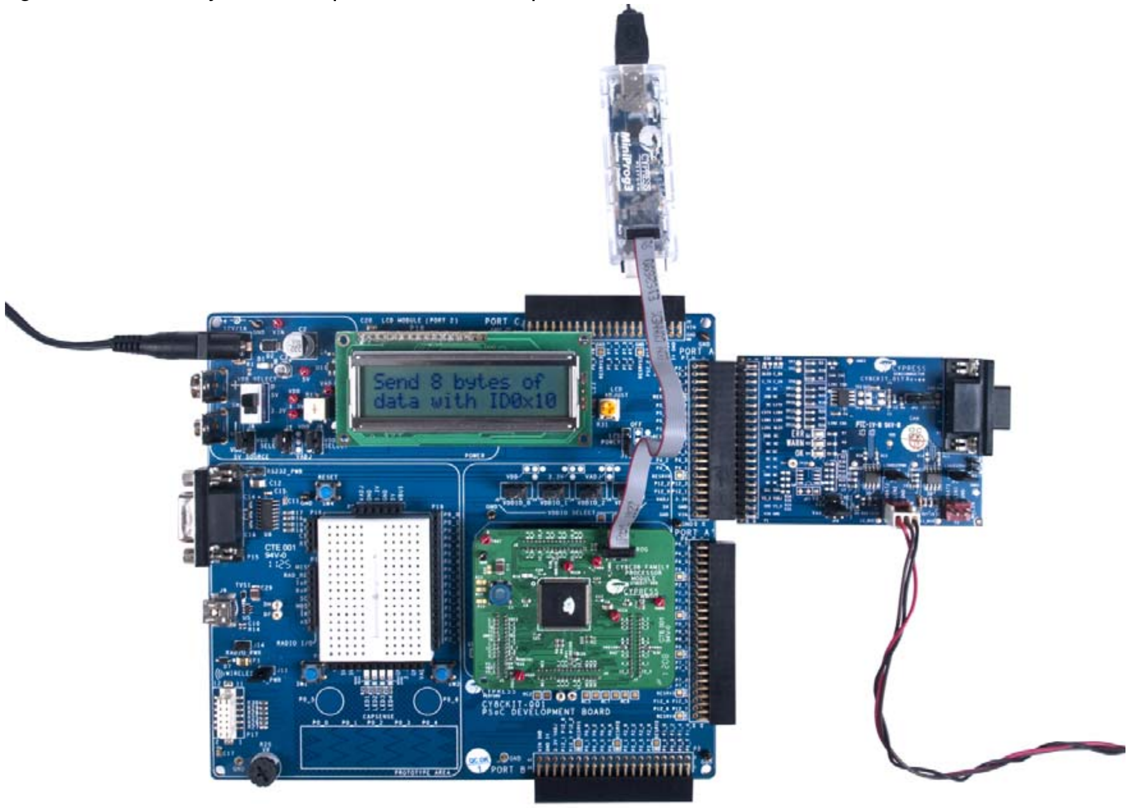
Figure 3-12. Verifying LCD Output of CAN_Example_1



3.4.2 LIN Communication

1. When the LIN_Example project starts, the LIN Slave component is initialized. If the startup is successful, the 'LINS Initialize Successful' message is displayed on the LCD, followed by 'Send the 8 bytes of data with ID 0x10'.

Figure 3-13. Verify LCD Output of LIN_Example



2. Configure the LIN analyzer/master with checksum set to **Enhanced** and baud rate set to **19200**.
3. Send an unconditional frame having eight bytes of data (Byte 1 is the scalar signal of 7-bit length and Byte 2 to 8 is byte array signal of 7-byte length) with a frame ID of 0x10 from the LIN analyzer/master. The frame ID and values of data received are displayed after ‘Recd-Data’ on the LCD. This data will be displayed for 12 seconds.

Figure 3-14. Received Data Display

4. After 12 seconds, the LCD display changes to ‘Send ID 0x11 to read back data’. Send a frame with an ID of 0x11 from the LIN analyzer/master.
5. When the frame with ID 0x11 is received, the data is sent to the LIN master. The frame ID and values of sent data is displayed on the LCD for 12 seconds.

Figure 3-15. Sent Data Display

6. After 12 seconds, the LCD display again switches to "Send the 8 bytes of data with ID 0x10". Repeat steps 3 to 5 to check LIN slave communication with a different set of data.

LIN Slave Communication Diagram

Protected Identifier	Data1	Data2	Data3	Data4	Data5	Data6	Data7	Data8	Checksum (Enhanced)
0x50	0x01	0x02	0x03	0x04	0x05	0x06	0x07	0x08	0x8B
0x11	0x01	0x02	0x03	0x04	0x05	0x06	0x07	0x08	0xCA
0x50	0x0A	0x0B	0x0C	0x0D	0x0E	0x0F	0x1A	0x1B	0x2F
0x11	0x0A	0x0B	0x0C	0x0D	0x0E	0x0F	0x1A	0x1B	0x6E

Legend:

Data published by LIN Master
Data published by LIN Slave

3.5 Using a CAN Bus Analyzer Tool

This kit functions most effectively when two CY8CKIT-001 DVKs and two CY8CKIT-017 EBKs are available. However, it is also possible to replace one CY8CKIT-001 DVK and one CY8CKIT-017 EBK with a CAN bus analyzer or emulator tool. It is even possible to use any other CAN node to communicate with this kit.

If you use a CAN bus analyzer or emulator tool to communicate with this kit, then the tool must be set up to send and receive CAN messages (at proper intervals) with proper length and message ID and at a proper baud rate. See the Code Examples chapter on page 31 for more details on the CAN controller configuration of this kit's code examples.

If you use any other CAN node to communicate with this kit, then you may need to modify the firmware to allow communication. You can modify the code examples, firmware, or settings of the other CAN node.

3.6 Using a LIN Bus Analyzer Tool

The LIN_Example project demonstrates functionality of the LIN slave device, so the LIN master device must be also used. The LIN bus analyzer or emulator tool can be used as the LIN master device. It is also possible to use any other LIN master node to communicate with this example project. If you use a LIN bus analyzer or emulator tool, then the tool must be set up to send and receive frames with proper length and ID and at a proper baud rate. See the Code Examples chapter on page 31 for more details on the LIN slave component configuration of this kit example project.

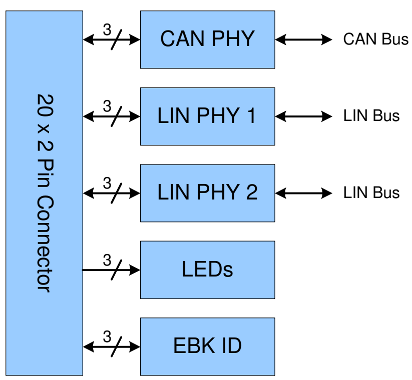
4. Hardware

4.1 System Block Diagram

The CAN/LIN EBK hardware consists of the following functional blocks:

- CAN transceiver circuit (TJA1050)
- LIN transceiver circuit in master configuration (TJA1020)
- LIN transceiver circuit in slave configuration (TJA1020)
- Three indicator LEDs
- EBK identification circuit (not populated)
- 40-pin (2x20) connector (Sullins Connector Solutions, S2111E-20-ND)

Figure 4-1. EBK Block Diagram



The primary functional blocks of this EBK are the three transceiver circuits: one CAN transceiver circuit and two LIN transceiver circuits. Each of these transceiver circuits enables a digital CMOS PSoC device to interface with a physical CAN or LIN bus, respectively. Without these transceiver circuits, it is impossible for CMOS devices to communicate with other CAN or LIN nodes on a CAN or LIN bus.

The EBK ID functional block consists of two circuits that are not populated, and can therefore be ignored. These circuits exist to provide forward compatibility with automatic EBK identification features that do not yet exist.

The LEDs functional block consists of three, active-low LEDs that can provide indications. These LEDs are driven by PSoC pins.

The 40-pin (2x20) connector connects the configured PSoC I/O pins to the various circuits on the expansion board.

4.2 CAN Physical Layer Transceiver Circuit

The CAN physical layer transceiver circuit on the expansion board uses a TJA1050 CAN transceiver device. This device translates differential CAN bus signals to and from digital CMOS signals with standard TTL voltage levels.

Three signals are used between this circuit and the PSoC controller. These signals are CAN_RX, CAN_TX, and CAN_EN. Any data signals on the CAN bus are driven at the CAN_RX signal. Any data signals driven on the CAN_TX signal is driven onto the CAN bus. The CAN_EN signal enables and disables the CAN transceiver.

A pull-up resistor footprint (R1) is provided on the CAN_EN net. This can be populated if it is desired to have a pull-up resistor on the CAN_EN net.

See the TJA1050 device datasheet for details on each of these three pins of the CAN transceiver.

4.2.1 CAN Bus Clock Accuracy

For accurate CAN communication, a CAN controller device must typically have a clock source with a frequency tolerance of 0.5% or less. Therefore, the PSoC device used as the CAN controller must meet this requirement. If the native internal oscillator tolerance of the PSoC device is greater than 0.5%, then some external clock source that is more accurate must be used. For example, an external oscillator or an external crystal can be used with the PSoC to improve the accuracy of the clocks in the PSoC device.

Footprints for oscillator and crystal devices (and any necessary passive components) are provided on the CY8CKIT-001 DVK board and PSoC processor module boards. Newer versions of these boards already have these footprints populated with components. If you have an older PSoC processor module that does not have a populated crystal circuit, you can populate it yourself or contact Cypress Technical Support for assistance.

See Design Wide Resources on page 48 for details on the clock configuration of this kit's code examples.

4.2.2 CAN Bus Connector

The following table shows the pinout of the CAN DB9 connector (P2) on the expansion board.

Table 4-1. CAN Connector Pinout

Pin	Signal
1	NC
2	CAN_L
3	GND
4	NC
5	NC
6	GND
7	CAN_H
8	NC
9	NC (VIN)

By default, pin 9 of the CAN connector is left floating. However, if the CANEXTPWR jumper (JP3) is populated, pin 9 of the CAN connector is connected to the VIN power rail of the DVK and EBK. This is useful if you want to power up other CAN nodes through the CAN connector or if you want to power up the DVK and EBK from the power supply of some other CAN node.

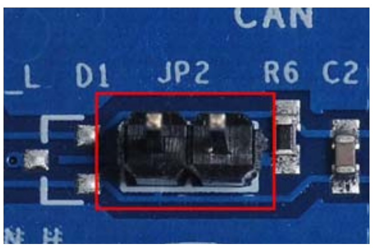
Warning: Take extra care when populating the CANEXTPWR jumper. The DVK or EBK can be damaged if the DVK and EBK have their own power supply powering the VIN rail and a different power supply is connected through pin 9 of the CAN connector.

4.2.3 CAN Bus Termination

The CAN specification requires that CAN nodes located physically at the end of the CAN bus must terminate the CAN differential signals with 120 Ω. The EBK features a 120-Ω termination resistor (R6) that can be enabled or disabled by using jumper JP2. If JP2 is populated, the termination resistor is active. If JP2 is not populated, the termination resistor is not active.

The termination resistor is active (JP2 is populated) by default.

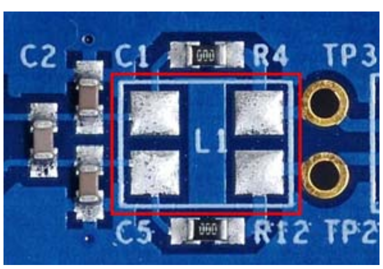
Figure 4-2. JP2 Jumper



4.2.4 Choke Footprint

A footprint for a common-mode signal suppression choke is available on the expansion board, but it is not populated. This footprint can be populated with a B82789C0 (or equivalent) choke to suppress common-mode signals on the CAN bus. If a choke component is mounted on the L1 footprint, resistors R4 and R12 must be removed from the board. The choke component has no effect if these two resistors are not removed.

Figure 4-3. CAN Choke Footprint



4.2.5 CAN Circuit Isolation

The CAN circuit can be completely isolated from the rest of the CY8CKIT-001 DVK by removing resistors R3, R5, and R10. Each of these resistors is 0 Ω. This is useful if the CAN circuit is not needed, but other circuits on the EBK are. In this case, isolating the CAN circuit ensures that the CAN circuit does not interfere with any other circuits that share the same pins. The footprints for R3, R5, and R10 are designed so that they can be reconnected easily with a ‘solder jumper’ instead of repopulating the footprints with a 0-Ω resistor.