Technical Article

Industrial Communications: CAN You Do IEC 61131-3 PLC Programming with BeagleBone?

Thomas Mauer

Apr 21, 2017

Factory automation and control applications used the Controller Area Network (CAN) and CANopen protocols as field buses long before the invention of industrial Ethernet. When used as a communication field bus, these protocols connect the programmable logic controller (PLC) to digital input/outputs (I/Os), motor drives, sensors and actuators.

Bosch developed CAN in 1983. Its initial use case was the communication bus inside of an automobile; it enabled peripherals like the brake system, ignition control and motor to communicate with one another. CAN's low-cost and robust communication physical layer led to its adoption on the factory floor; CANopen is a specific enhancement of the CAN communication protocol for factory automation. CAN and CANopen are still deployed in many production lines; many manufacturers add the CAN interface to their equipment for compatibility with deployed equipment.

In March, while at the Embedded World trade show in Nuremberg, Germany, I learned that it is very simple to enable industrial International Electrotechnical Commission (IEC) 61131-3 PLC programming on a BeagleBone Black (BBB) using TI's Sitara™ AM335x processor with an CAN and CANopen communication "cape" or add-on board.

Getting an easy, fast start with the CAN interface and IEC 61131-3 PLC programming is essential for any developer. I find that a BBB is a great platform with which to get started because it runs with Linux, and the CAN tools and CANopen stack are simple to install and well documented. Still, you will have to write the software application that interfaces to the CAN bus.

At Embedded World I visited the 3S booth, the manufacturer of Controller Development System (CODESYS), an industrial IEC 61131-3 PLC programming software program. 3S offers a free evaluation license of CODESYS in their online shop that supports the CAN interface. The evaluation version operates for two hours on the BBB; you can reboot the BBB to restart the two-hour timer. This evaluation version is an ideal starting point to learn about the CAN interface, BBB and the IEC 61131-3 PLC programming language.

So what kind of hardware and software do you need?

You need a BBB, which you can purchase from many electronic component distributors. You also need a CAN interface cape for the BBB, which you can build on your own from the printed circuit board (PCB) design file in TI's Energy Efficient and Isolated CANopen Interface Reference Design. The CAN cape includes rotary switches and status light-emitting diodes (LEDs).

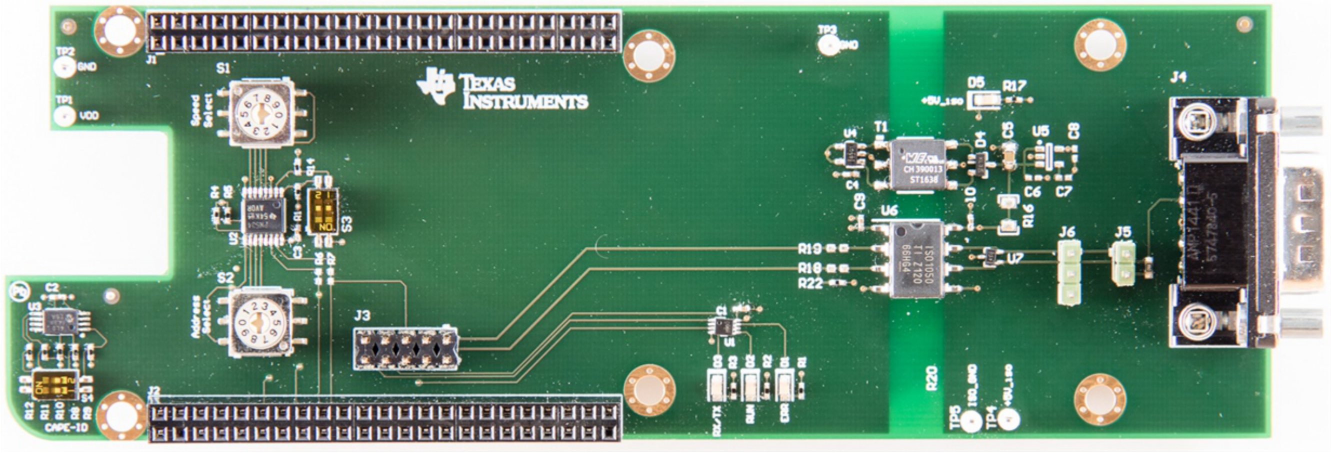


Figure 1. Isolated CAN Reference Design PCB Used as a Transceiver Cape for BBB

For software installation, follow the instructions in the design guide of the reference design, which walks you through updating the BBB Linux image, configuring the electrically erasable programmable read-only memory (EEPROM) ID on the CAN cape and testing basic CAN communication.

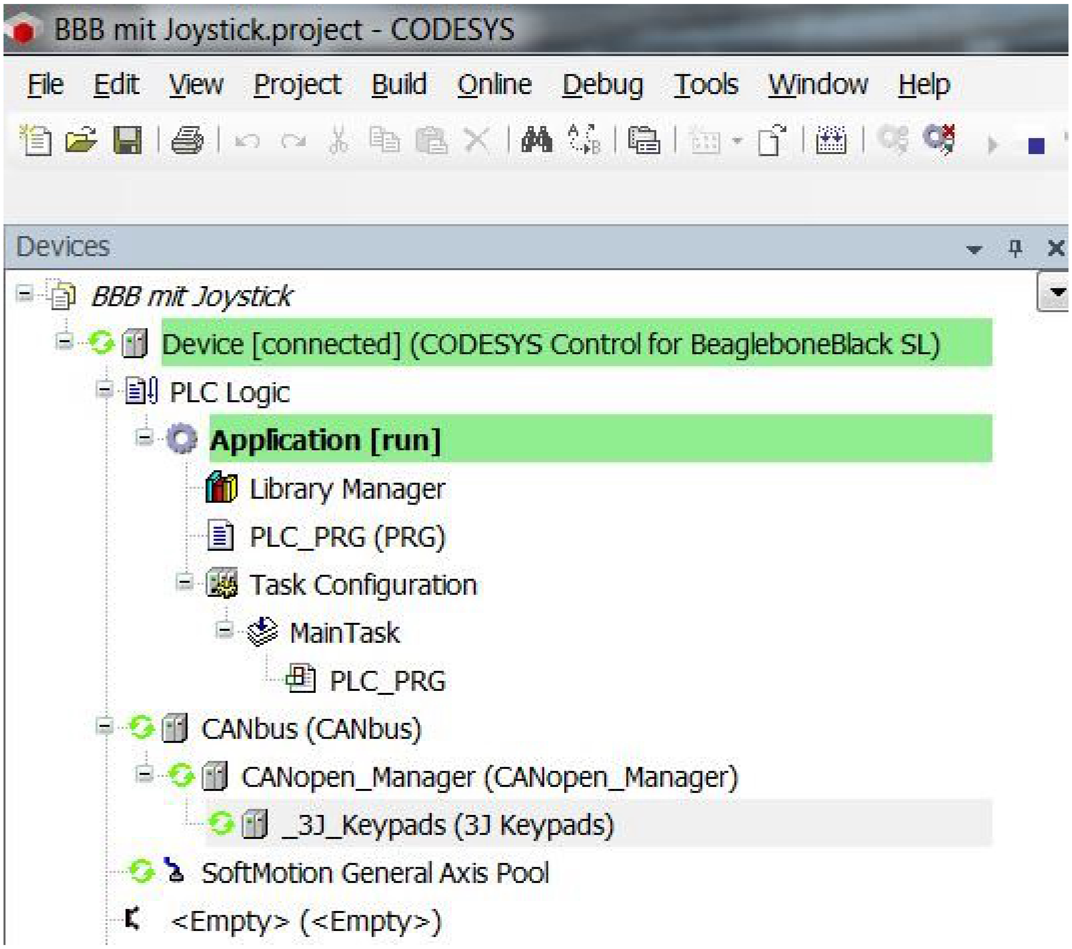
Next, install the CODESYS runtime software on your BBB. The software package is available on the 3S website. Follow the step-by-step description in the CODESYS for BBB getting started guide.

After that, you will want to make sure that the CAN drivers are loaded at Linux boot time so that the CODESYS runtime software will work with the CAN cape. To do this, edit the file /etc/modules and add the kernel modules can, can-dev, can-raw and c_can_platform. After the modification, the file should look like this:

```
# /Etc/modules: Kernel Modules to Load at Boot Time.

#

# This File Contains the Names of Kernel Modules That Should Be Loaded

# At Boot Time, One per Line. Lines Beginning with "#" Are Ignored.

Can

Can-dev

Can-raw

C_can_platform
```

Now reboot the BBB to start with CODESYS and IEC 61131-3 PLC programming of the CAN interface.

To test the setup, I used a keypad with a CANopen interface; Figure 2 shows my CODESYS configuration. The green marks indicate that the CAN protocol is operating OK. From here, I can start writing an industrial application by programming in IEC 61131-3 PLC language.



- BBB mit Joystick
  - Device [connected] (CODESYS Control for BeagleboneBlack SL)
    - PLC Logic
      - Application [run]
        - Library Manager
        - PLC_PRG (PRG)
        - Task Configuration
          - MainTask
            - PLC_PRG
    - CANbus (CANbus)
      - CANopen_Manager (CANopen_Manager)
        - _3J_Keypads (3J Keypads)
    - SoftMotion General Axis Pool
    - <Empty> (<Empty>)

Figure 2. CODESYS Device Configuration

This was a great learning experience for me, as it was very easy to get the reference design's isolated CAN cape working on a BBB. Enabling industrial IEC 61131-3 PLC programming through CODESYS is a great way to get started with structured text (ST) language, which one of the programming languages for IEC 61131-3. I hope that you find this description useful if you need to get started with the CAN interface.

Additional Resources

- Follow the industrial Ethernet blog post series if you are interested in field bus and industrial Ethernet communication.
- Click here to view our latest Energy Efficient and Isolated CANopen Interface reference design.

IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATA SHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES “AS IS” AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, regulatory or other requirements.

These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you will fully indemnify TI and its representatives against, any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

TI’s products are provided subject to TI’s Terms of Sale or other applicable terms available either on ti.com or provided in conjunction with such TI products. TI’s provision of these resources does not expand or otherwise alter TI’s applicable warranties or warranty disclaimers for TI products.

TI objects to and rejects any additional or different terms you may have proposed.

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