DT0056
Design Tip

Planning for easy troubleshooting with your L647x, L648x, or PowerStep01 design

By Dennis Nolan

Main components	
L6470, L6472	Fully integrated microstepping motor driver with motion engine and SPI
L6474	Fully integrated microstepping motor driver
L6480, L6482	Fully integrated microstepping motor controller with motion engine and SPI
powerSTEP01	System-in-package integrating microstepping controller and 10 A power MOSFETs

Purpose and benefits

This document describes a simple design technique that allows easy connection of the prototype board to a known development tool to allow easy diagnostic and debug of the motor driver hardware.

Description

The L647x, L648x, and PowerStep01 stepper motor drivers all share a similar command interface and a common mission: that of driving stepper motors with a high degree of performance and precision. Given that commonality, the path that the typical designer takes to develop a product using these devices also tends to be similar. In a typical product development, the designer will usually start out by obtaining the board or boards that constitute the ST evaluation system for the device, downloading the SPINFamily evaluation tool Graphical User Interface (GUI), and then installing it on a PC. With the evaluation system connected to the PC and the motor, and the motor mechanically coupled into the mechanical load, the designer may use the GUI to load and examine all of the device configuration registers and operate the motor in either speed or position control modes. In this manner, the designer can fine tune the device configuration to match the motor and the mechanical load being driven, as well as evaluate the fitness of the drive and motor for the mechanical task at hand.

Once the designer is satisfied with the operation of their motor drive design using the ST evaluation system, the next step will usually be to design a prototype circuit board incorporating the stepper driver device and most likely a microcontroller with an SPI port to control the device. The designer also takes on the task of writing the microcontroller code to control the device. Usually, many lines of code are written, a few obvious hardware issues are dispatched, and the system is powered up in great anticipation. Unfortunately, the designer often finds that the motor drive system does not work, or does not work as it did in the evaluation phase. This is the nature of product development.

Now comes the troubleshooting stage. Is it a hardware problem? Is it a software problem? Is it a problem with the SPI communication link between the microcontroller and the device? In the evaluation stage, performance was satisfactory so we know that, if all aspects of the design were working as planned, performance would be correct.

The question now: HOW TO TROUBLESHOOT THIS COMPLEX SYSTEM?

Consider the following suggestions to prepare the troubleshooting system:

1. When the prototype is designed, insert a 1.0K series resistor in series with the 4 SPI communication lines: SDI, SDO, CK, and CS*. The resistors will not affect normal operation and can be very small as they dissipate negligible power
2. Include a test point for each signal on the device side of the resistor.
3. You will also need an ST STEVAL-PCC009V2 USB-to-SPI bridge board. This board provides an SPI communications port at the 10 pin dual in-line header J1. The pinout of the SPI connector is as follows:
    a. J1-1...GND
    b. J1-5...SDO
    c. J1-6...CK
    d. J1-7...SDI
    e. J1-8...CS*
4. Prepare a test cable to connect the STEVAL-PCC009V2 to the prototype board. The test cable must be constructed so that it will mate with J1 on the STEVAL-PCC009V2 and should terminate with the four signal wires (and GND) listed above with the individual wires separated (these can be "peeled" apart from a typical ribbon cable) and terminated with a convenient "grabber", such as an EZ-hook or micro-hook.

You can now use this cable to transfer control of the device "on-the fly" between your on-board microcontroller and the evaluation GUI in order to facilitate troubleshooting.

Using the troubleshooting system

1. Connect the bridge board (PCC009V2) to your PC and to the test cable.
2. Connect the four grabbers and ground to their respective test points. (You did remember to give yourself a good ground test point, didn't you?).
3. Start up the GUI as you have done before and select the part, but do not connect.
4. Power-up your prototype board.
5. Initiate connection with the GUI. At this point, we have "hijacked" the device back to control with the GUI. Proceed to load in your known good register configuration file. (You did remember to save the known good configuration file, didn't you).
6. You should now be able to run the motor and mechanical system as you did when using the ST evaluation board. If there are any problems with operation, this points to a hardware problem on the prototype. Having the complete capabilities of the GUI available should make troubleshooting the problem much easier. If all has tested well, move on to the next stage.
7. Carefully remove the four signal grabbers (leave the GND grabber). Do not shut down the GUI. Modify the microcontroller program so that the full initialization routine to configure the device registers executes, but the program then falls into an endless loop.
8. Power-up the prototype board so that the program will run and the registers will be initialized (and then the endless loop). Allow enough time for the initialization to execute, then carefully re-connect the four signal grabbers. Go to the register display page of the GUI and do a full register read-back (green icon in upper left screen corner).
9. Now you can carefully examine the contents of all the device registers. This data should match the intentions of your register initialization as written in the program. If not, having a known good way of reading back and checking the content of these registers should facilitate troubleshooting the problem in the code.

Support material

Documentation
Datasheet: L6470, Fully integrated microstepping motor driver with motion engine and SPI
Datasheet: L6472, Fully integrated microstepping motor driver with motion engine and SPI
Datasheet: L6474, Fully integrated microstepping motor driver
Datasheet: L6480, Fully integrated microstepping motor controller with motion engine and SPI
Datasheet: L6482, Fully integrated microstepping motor controller with motion engine and SPI
Datasheet: powerSTEP01, System-in-package integrating microstepping controller and 10 A power MOSFETs

Revision history

Date	Version	Changes
15-Feb-2016	1	Initial release
26-Aug-2016	2	Updated product associations

IMPORTANT NOTICE – PLEASE READ CAREFULLY

STMicroelectronics NV and its subsidiaries (“ST”) reserve the right to make changes, corrections, enhancements, modifications, and improvements to ST products and/or to this document at any time without notice. Purchasers should obtain the latest relevant information on ST products before placing orders. ST products are sold pursuant to ST’s terms and conditions of sale in place at the time of order acknowledgement.

Purchasers are solely responsible for the choice, selection, and use of ST products and ST assumes no liability for application assistance or the design of Purchasers’ products.

No license, express or implied, to any intellectual property right is granted by ST herein.

Resale of ST products with provisions different from the information set forth herein shall void any warranty granted by ST for such product.

ST and the ST logo are trademarks of ST. All other product or service names are the property of their respective owners.

Information in this document supersedes and replaces information previously supplied in any prior versions of this document.