Motor Control Platform Quick Start Guide

IN THIS DOCUMENT

1 Introduction

The XMOS Motor Control Platform simplifies the development of applications requiring multi-axis field orientated control of motors, fieldbus and industrial Ethernet.
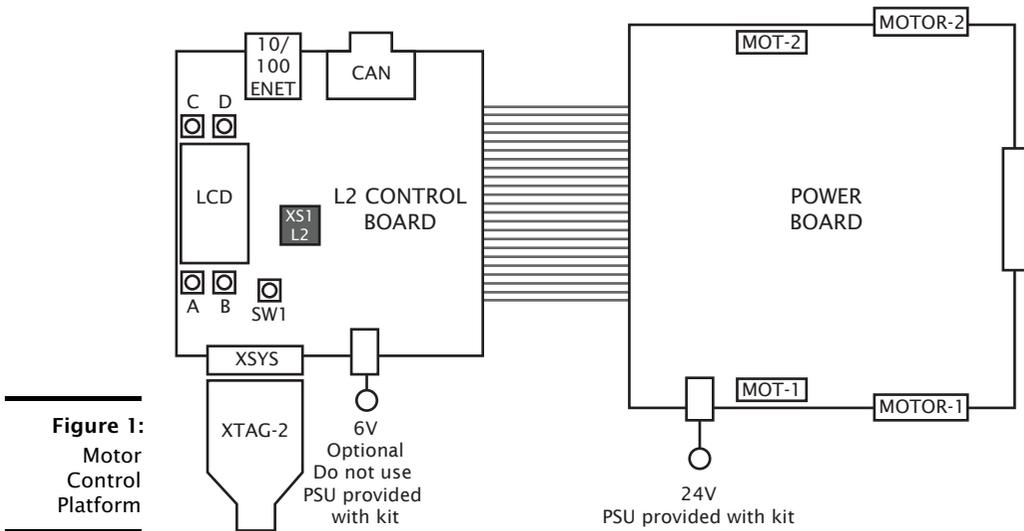
Figure 1:
Motor
Control
Platform

The kit contains the following hardware:

▶ A control board comprising a 500MHz XS1-L2 processor, Ethernet interface, CAN interface, a discrete 2-channel 12-bit sample-and-hold ADC, LCD display and XSYS interface

▶ A power board comprising 2 motor connectors, 2 connectors for QEI sensors, 6x24V 5A per channel inverters and a 0V zero-crossing detector (up to 1 MHz)

▶ 50-way ribbon cable for connecting the control board and power board

▶ XTAG-2 debug adapter

▶ 24V power supply

The board firmware includes a dual-axis field oriented control application and simple commutation application for BLDC motors. A single core can drive the current loop for a 3-phase BLDC motor at up to 125kHz, with a PWM resolution of 4ns. The firmware is built from the following software components, which can be used in a wide range of motor control applications:

▶ 10/100MBit Ethernet interface

▶ 0.5Mbit CAN interface

▶ 2 Hall sensor inputs

▶ 3-phase complementary symmetrical PWM with dead time insertion

Multiple chips can be connected to scale to a nearly unlimited number of coordinated axes for robotics and other applications, or the control board can be used by itself as a generic L2 development platform.

The kit is supported by the following tools:

▶ Demo GUI based on LABView run-time, which allows you to control the demo applications from a host PC using the Ethernet or CAN interface.

▶ XMOS Development Tools, which provide everything you need to develop your own applications, including an IDE, real-time software scope and timing analyzer.

2 Set up the hardware to run the firmware demo

The Motor Control Platform comes with a dual-axis FOC application programmed into flash memory on the control card. To set up the hardware and run this demo, follow these steps:

1. Connect the control board interface to the power board interface using the 50-wire ribbon cable.

2. Connect the 8-wire cable on one of the motors to the MOTOR-1 connector, and connect the 4-wire cable to the MOT-1 connector. Connect the second motor to the MOTOR-2 and MOT-2 connectors the same way.

3. Connect the 24V supply to the power board, and use a power lead with an IEC 320-C13 connector (also known as a "Kettle Lead", not provided) to connect the power supply to a mains outlet.

Do **not** connect the 24V power supply to the control board. The control board takes a 6V power supply and will be damaged by 24V.

On power-up, the motors should start spinning at a demand speed of 1000RPM. The LCD shows the speed of each motor, and the demand speed of both. You can use buttons A and B to alter the demand speed for the system in steps of 100RPM.

2.1 Control the application using a GUI interface

XMOS has developed a demo GUI application that allows you to control the board from a host PC using either the Ethernet or CAN interface. The GUI application is available for Windows and requires the LabView 8.1 runtime environment to be installed on your PC.
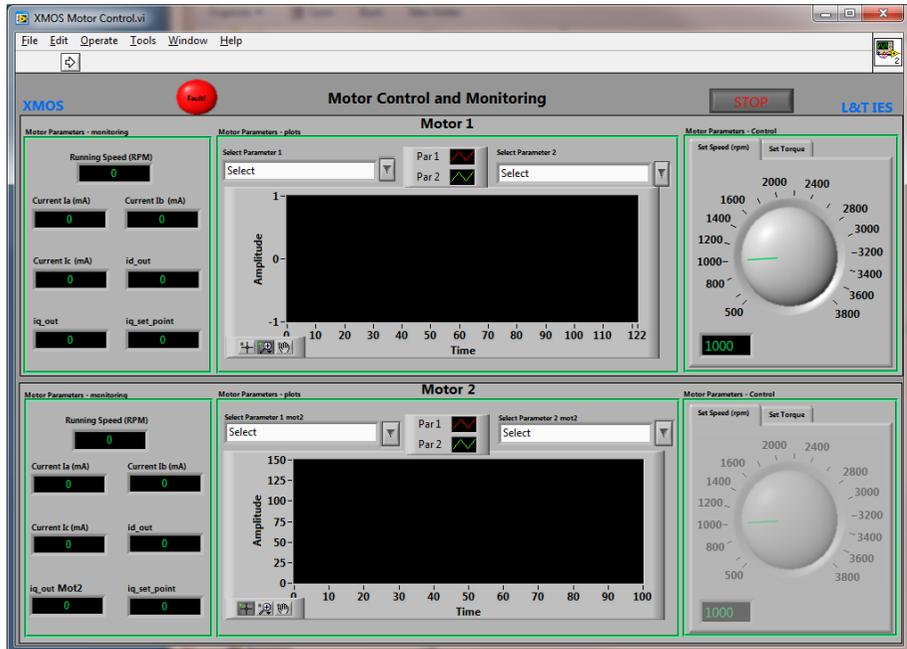
Figure 2:
Demo GUI
application

The GUI interface can be downloaded from:

▶ http://www.xmos.com/partnum/XM-001564-SM

The LabView 8.1 runtime environment can be downloaded from:

▶ http://joule.ni.com/nidu/cds/view/p/id/861/lang/en

To run the GUI, unzip the download archive to an empty directory and run the file MotorControl.exe.

On launching the GUI, a dialog pops up asking you to select CAN or Ethernet. If you select Ethernet, you are then asked to provide the IP address of the board. The default firmware uses IP address 169.254.0.1 (a link local IP address).

To use the CAN interface, you must first configure the firmware to use the CAN interface (see §3.2). LabView supports the Kvaser Leaf Light HS USB to CAN dongle.

3   Configure the firmware demo

The firmware demo is provided as a source code archive. To configure, you should modify the source code for the dual-axis FOC application, build the project and load it onto your hardware using the XMOS Development Tools.

The motor control firmware is available from:

▸ http://www.xmos.com/partnum/XM-000011-SW

The XMOS Development Tools are available from:

▸ http://www.xmos.com/tools

For instructions on installing the tools and XTAG-2 driver, and on starting up the tools, see X5713 and X8084.

3.1   Create a demo application

You can create a demo application either in the XMOS Development Environment (XDE) or on the command-line. XMOS recommends making a copy of the original application so that you can easily revert to the default firmware settings in the future.

XDE **Create an application using the XDE**

1. Choose **File ▸ Import**.

2. Double-click on the **General** option, select **Existing Projects into Workspace** and click **Next**.

3. In the **Import** dialog box, click **Browse** (next to the **Select archive file** text box). In the dialog that appears, browse to the directory in which you downloaded the firmware archive, select it (`.zip` extension) and click **Open**.

4. Click **Finish**.

   The XDE imports a set of projects into your workspace.

5. In the **Project Explorer**, click the folder `sw_motor_control` to expand it.

6. Right-click on the sub-folder `app_dsc_demo` and select **Copy**, then right-click the folder `sw_motor_control` and select **Paste**. In the dialog that appears, enter a name for your application and click **OK**.

7. Double-click on the file `sw_motor_control/Makefile` to open it in an editor, and ensure that your application is checked as enabled in the build.

XMOS

Create an application on the command line

1. Unzip the firmware archive.

2. Change to the directory `sw_motor_control` and copy the directory `app_dsc_demo` to a new directory. For example, in Linux type the following command:

   ▶ `cp -fr app_dsc_demo app_my_demo`

3. Edit the file `sw_motor_control/Makefile` and add the name of your application to the `BUILD_SUBDIRS` environment variable.

3.2 Configure your application settings

Application settings are configured by modifying the source code.

▶ **Change the TCP/IP address**

By default the Ethernet and TCP/IP interface has a statically allocated IP address of 169.254.0.1 (a link local IP address) and a net mask of 255.255.0.0. To change, in your application directory open the file `src/main.xc` and search for the function `init_tcp_server` which contains these values; modify as required.

▶ **Switch from Ethernet to CAN control**

By default the application is controlled by the buttons around the LCD and the Ethernet interface. To use CAN instead, in your application directory open the source file `src/dsc_config.h`, enable the macro `USE_CAN` and disable the macro `USE_ETH`.

The file `src/dsc_config.h` contains other compile-time configuration options. These options are described in more detail in the motor control software guide (see X7328).

3.3 Build and run your application

Once you have configured your application, you must build it into an executable binary and load this binary onto your hardware. To build and run, follow these steps:

Once you have configured your application, you must build it into an executable binary and load this binary onto your hardware.

To use the XDE

1. Select your application in the **Project Explorer** and click **Build**.

   The XDE builds the firmware, displaying progress in the **Console**. If there are no errors, the XDE adds the compiled binary to the application folder `bin/Debug`.

2. Ensure that your XMOS XTAG-2 debug adaptor is connected to the XSYS connector on the control board, and use a USB cable (not provided) to connect the adapter to your PC.

3. Choose **Run ▶ Run Configurations**.

XMOS®

4. In the left panel, double-click **XCore Application**.

   The XDE creates a new configuration and displays the default settings in the right panel.

5. In **Name**, enter a name such as Demo App.

6. The XDE tries to identify the target project and executable for you. To select one yourself, click **Browse** to the right of the **Project** text box and select your project in the **Project Selection** dialog box. Then click **Search Project** and select the executable file in the **Program Selection** dialog box.

7. Ensure that the **hardware** option is selected, and in the **Target** drop-down list select your target board.

8. Click **Run**.

   The XDE loads your executable, displaying any output generated by your program in the **Console**.

To use the command-line tools

1. Change to your application directory and enter the following command:

   ▶ xmake all

   The tools build your application. If there are no errors, the tools create a binary in the sub-folder bin/Debug.

2. Ensure that your XMOS XTAG-2 debug adaptor is connected to the XSYS connector on the control board, and use a USB cable (not provided) to connect the adapter to your PC.

3. To run, enter the following command:

   ▶ xrun bin/Debug/*binary*.xe

4 Configure the hardware

The hardware can be configured by modifying the jumper settings on the control board and power board.

4.1 Control board jumper settings

The control board jumper settings are shown in Figure 3. Jumper J2 controls the power source for the board. The board can be powered from either the power board or from a separate 6V power supply. For the default firmware to run correctly, jumpers J33 and J34 must be set as shown.

Figure 3:
Control board jumper settings
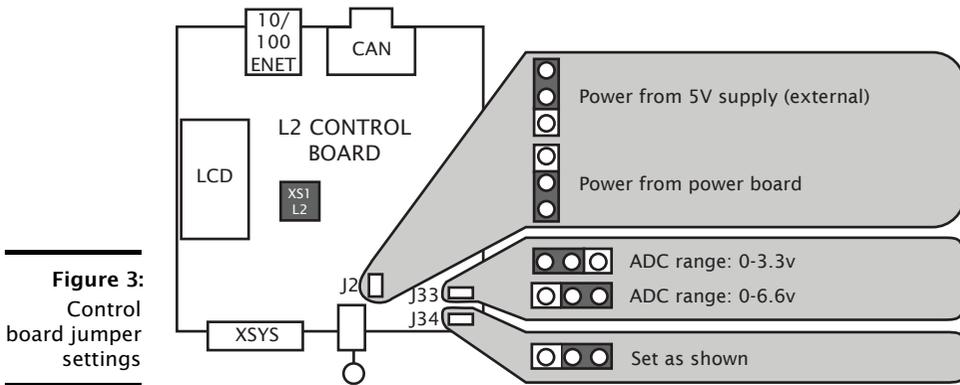
4.2   Power board jumper settings

The power board jumper settings are shown in Figure 4. Jumper J6 controls the watchdog protection mode. If enabled, the watchdog circuit is directly connected to the processor, otherwise you must hold button SW1 on the control board to enable the watchdog connection. This latter configuration is useful for testing new software algorithms: hold down the button for normal operation, and if an error occurs causing risk of damage to the motors or power board, release the button to prevent the FETs from being further engerized.

Figure 4:
Power board jumper settings
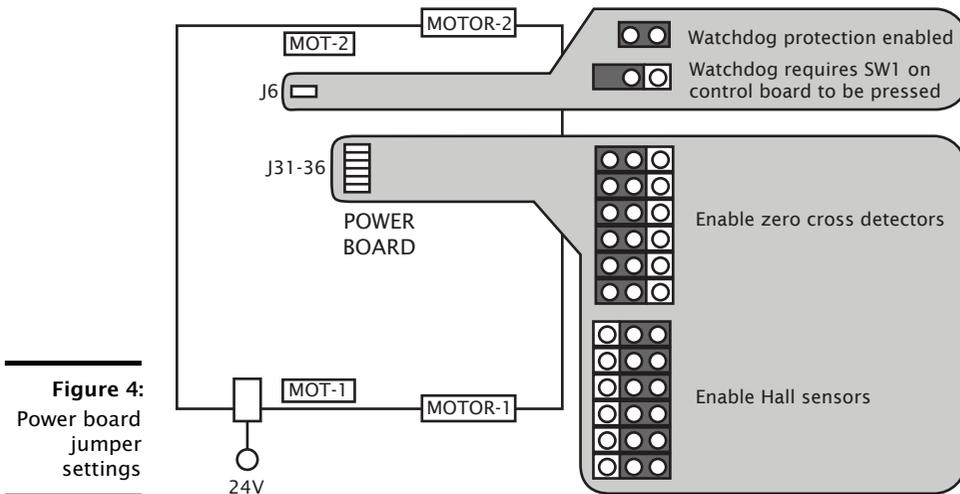
Jumpers J31 to J36 are used to enable either the hall sensors or zero-cross detectors. Note that the default application firmware does not use the zero-cross detectors.

5 Motor connectors

If one of the 5-wire quadrature cables becomes disconnected from its motor, care must be taken when reconnecting it to ensure that the alignment flanges on the cable match those on the connector **before** inserting, as shown in Figure 5. Inserting the cable incorrectly may permanently damage your hardware.

Figure 5:
5-wire quadrature encoder connection