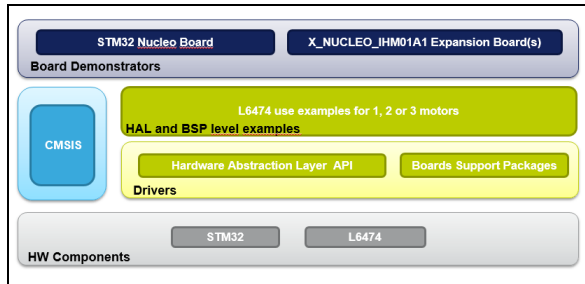


STM32Cube software expansion for the X-NUCLEO-IHM01A1 expansion board

Data brief



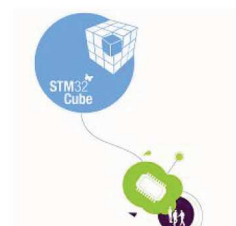
Features

- Driver layer for complete management of the L6474 microstepping motor driver integrated in the X-NUCLEO-IHM01A1 expansion board
- Examples for controlling up to three stepper motors
- Easy portability across different MCU families, thanks to STM32Cube
- Free, user-friendly license terms

Description

The X-CUBE-SPN1 software package is an expansion for STM32Cube, associated with the X-NUCLEO-IHM01A1 expansion board. It is compatible with the Nucleo STM32F401, the Nucleo STM32F030 and the Nucleo STM32L053 when they are equipped with one or more (up to 3) X-NUCLEO-IHM01A1 boards.

The source code of this package is based on STM32Cube to ease portability across different STM32 MCU families.



1 Detailed description

The X-CUBE-SPN1 software package allows complete management of the L6474 fully integrated microstepping motor driver, by providing complete APIs. It offers the following features:

- L6474 registers read, write
- Nucleo and expansion board configuration (GPIOs, PWMs, IRQs, etc.)
- Speed profile configuration
- Motion commands
- FLAG interrupts handling (alarms reporting)
- Microstepping handling
- Daisy chaining handling

By starting the L6474 driver, the user specifies the number of L6474 chips which are connected to the Nucleo board (i.e. the number of X-NUCLEO-IHM01A1 expansion boards). Once set, the number of devices must not be changed.

Depending on the devices number, the driver:

- sets up the required GPIOs to handle the motor directions and the FLAG interrupt
- initializes the PWMs that will act as step clock generators
- initializes the speed profile (acceleration, deceleration, min and max speed) of each device by using the parameters of the file "easyspin_target_config.h"
- starts the SPI driver to communicate with the L6474 chips
- releases the reset of each of the L6474 chips
- disables the power bridge and clear the status flags of the L6474 chips
- loads the registers of each L6474 chips with the predefined values from "easyspin_target_config.h"

Once the initialization is complete, the user can modify the L6474 registers and speed profile configurations as desired. Most of the functions of the driver take a device ID (from 0 to 2) as input parameter. It gives the user the possibility to specify which of the device configurations to modify.

The user can also write a callback function and attach it to the Flag interrupt handler, depending of the actions to be performed when an alarm is reported (read the flags, clear and read the flags, etc.)

Then, the user can request to move one or several motors (using the same principle of device ID). This request can be:

- to move for a given number of steps in a specified direction
- to go to a specific position
- to run until reception of a new instruction

On reception of this request, the driver enables the PWM which is used as the step clock of the corresponding L6474.

At each pulse period, the motor will perform one step and an ISR (interrupt service routine) is triggered on the microcontroller side.

This ISR is used by the FW to count the number of performed steps and to update the speed. Indeed, the motor starts moving by using the minimum speed parameter. At each step, the speed is increased using the acceleration parameter.

If the target position is far enough, the motor will perform a trapezoidal move:

- acceleration phase using the device acceleration parameter
- steady phase where the motor turns at max speed
- deceleration phase using the device deceleration parameter
- stop at the targeted position

Otherwise, if the target position does not allow it to reach maximum speed, the motor will perform a triangular move:

- acceleration phase using the device acceleration parameter
- deceleration phase using the device deceleration parameter
- stop at the targeted position

A moving command can be stopped at any moment:

- either by a soft stop or softHiz which progressively decreases the speed using the deceleration parameter. Once the minimum speed is reached, the motor is stopped.
- Or by a hard stop or hardHiz command which immediately stops the motor.

When the motor is stopped using the softHiz or hardHiz command, the power bridge is automatically disabled.

To avoid sending a new command to a device before the completion of the previous one, the driver offers a EasySpin_WaitWhileActive() command which locks the program execution until the motor stops moving.

The driver also offers the possibility to change the step mode (from full step to 1/16 microstep mode) for a given device. When the step mode is changed, the current position (ABS_POSITION register) is automatically reset, but it is up to the user to update the speed profile (max and min speed, acceleration deceleration).

2 Revision history

Table 1. Document revision history

Date	Revision	Changes
07-Nov-2014	1	Initial release.

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.

© 2014 STMicroelectronics – All rights reserved

