UM1691
User manual

EVAL6470H-DISC: fully integrated stepper motor driver based on the L6470 and STM32™

Introduction

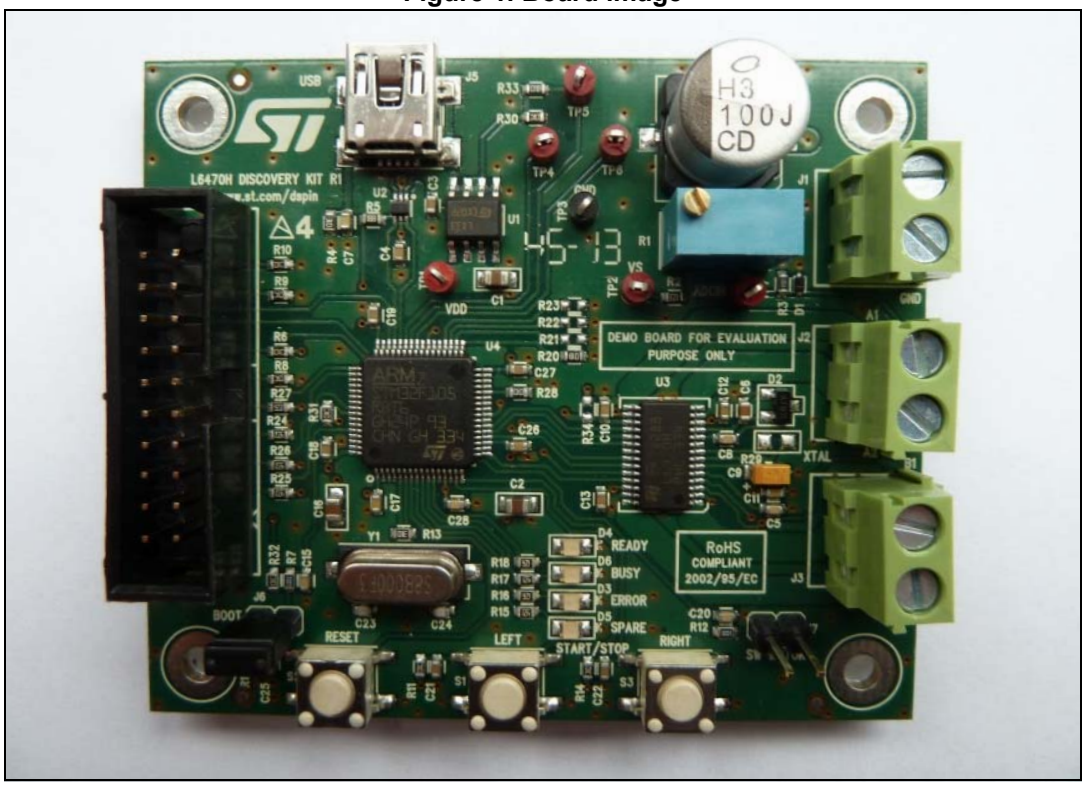
The EVAL6470H-DISC can be used together with the STM32™ firmware library V1.0 and constitutes a complete motor control evaluation and a development platform.

It is a demonstration board for motor control applications in the range of 8 V to 45 V of DC bus voltage using the STM32F105RB microcontroller with an internal 128 kB Flash size and a 64 kB internal RAM and the L6470 fully integrated solution suitable for driving two-phase bipolar stepper motors up to 1/128 microstepping.

The L6470 integrates a dual DMOS full bridge with all of the power switches equipped with an accurate on-chip current sensing circuitry suitable for non dissipative current control and overcurrent protection.

With dedicated hardware evaluation features, the EVAL6470H-DISC board is designed to help developers evaluate the device and develop their own applications.

Figure 1. Board image



Contents

1 Main features 4

1.1 Target applications 4

1.2 Description 4

2 Electrical characteristics of the board 5

3 Schematics, layout and bill of material 6

4 General description 11

4.1 Power supply 11

4.2 L6470 stepper motor driver 12

4.2.1 Charge pump 13

4.2.2 Voltage mode driving 14

4.2.3 Overcurrent and stall detection thresholds 14

4.2.4 Speed profile 15

4.3 STM32F105RB microcontroller 15

4.4 Firmware loading 17

4.4.1 DfuSe installation 17

4.4.2 Generate a DFU file from a HEX file 17

4.4.3 Board settings 19

4.4.4 DFU loading 20

4.5 Using the EVAL6470H-DISC with firmware for the GUI 24

4.5.1 Sanity check of the board with firmware for the GUI 24

4.5.2 Parameters exportation from the GUI to the FW library 33

4.6 Using the EVAL6470H-DISC with the firmware library 34

4.6.1 FW library package contents 34

4.6.2 FW L6470 library description 34

4.6.3 Demonstration sequence description 35

4.6.4 Create a project using the FW library package 39

5 References 40

6 Revision history 40

List of figures

Figure 1. Board image . . . 1
Figure 2. Schematic (microcontroller supply part) . . . 6
Figure 3. Schematic (microcontroller part) . . . 7
Figure 4. Schematic (motor driver part) . . . 8
Figure 5. Layout top layer and bottom layer . . . 8
Figure 6. Microcontroller supply section . . . 11
Figure 7. L6470 block diagram . . . 12
Figure 8. Charge pump circuitry . . . 13
Figure 9. STM32F105xx and STM32F107xx block diagram . . . 16
Figure 10. DFU file manager (action) . . . 17
Figure 11. DFU file manager (generation) . . . 18
Figure 12. Board settings . . . 19
Figure 13. DFU loading . . . 20
Figure 14. DFU file (open) . . . 21
Figure 15. DFU file (upgrade) . . . 22
Figure 16. DFU file (confirmation) . . . 22
Figure 17. DFU file (download OK) . . . 23
Figure 18. Starting board (boot mode) . . . 24
Figure 19. Starting board (motor power supply) . . . 25
Figure 20. Starting board (USB connection) . . . 26
Figure 21. Starting board (error case) . . . 27
Figure 22. Starting board (board OK) . . . 28
Figure 23. Starting board (display board ID) . . . 29
Figure 24. Starting board (action LEFT key) . . . 30
Figure 25. Starting board (action RIGHT key) . . . 31
Figure 26. Starting board (action RESET key) . . . 32
Figure 27. Parameters exportation . . . 33
Figure 28. Parameters exportation (save file) . . . 33

1 Main features

The characteristics of the EVAL6470H-DISC board are the following:

- DC voltage range from 8 V to 45 V
- Maximum load phase current at 3 $A_{r.m.s.}$
- Footprint for external resonator or crystal
- Control interface through trimmer - user keys and switch motor input
- Control through LED indicators
- Interface control by USB and debug outputs
- Compatible with SPINFamily evaluation tool
- Autonomous board due to an embedded firmware
- Up to 1/128 microstepping
- Optimized layout on 2-layer board - low cost and high thermal performance

1.1 Target applications

The demonstration board is designed to fit all typical stepper motor applications - it is an autonomous board due an embedded firmware.

1.2 Description

The EVAL6470_DISC is downloaded with an embedded firmware.

The possibility with this board is double:

- Adapt the settings with your specific setup (motor - voltage) by using the dedicated PC application.

  The user is able to save inside the board the specific parameters depending on a user setup.
- Using the board (with user parameters) directly on a site - without a connected PC.

2 Electrical characteristics of the board

Table 1. Electrical specifications

Name	Value
Supply voltage (V_S)	8 to 45 V
Maximum output current (each phase)	3 $A_{r.m.s}$
Logic supply voltage (V_{REG})	3 V (L6470 regulator supply)
Logic interface voltage (V_{DD})	3.3 V (USB supply)
Low level logic inputs voltage	0 V
High level logic input voltage	V_{DD}
Stepping	Up to 1/128 microstepping
Operating temperature	0 to 85 °C

3 Schematics, layout and bill of material

Figure 2. Schematic (microcontroller supply part)

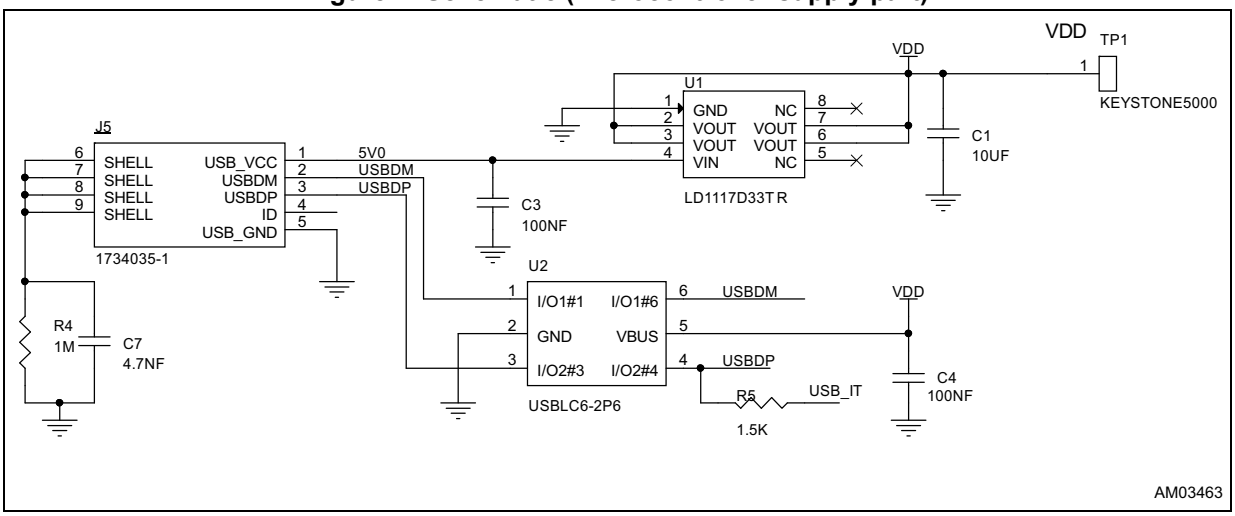


Figure 3. Schematic (microcontroller part)

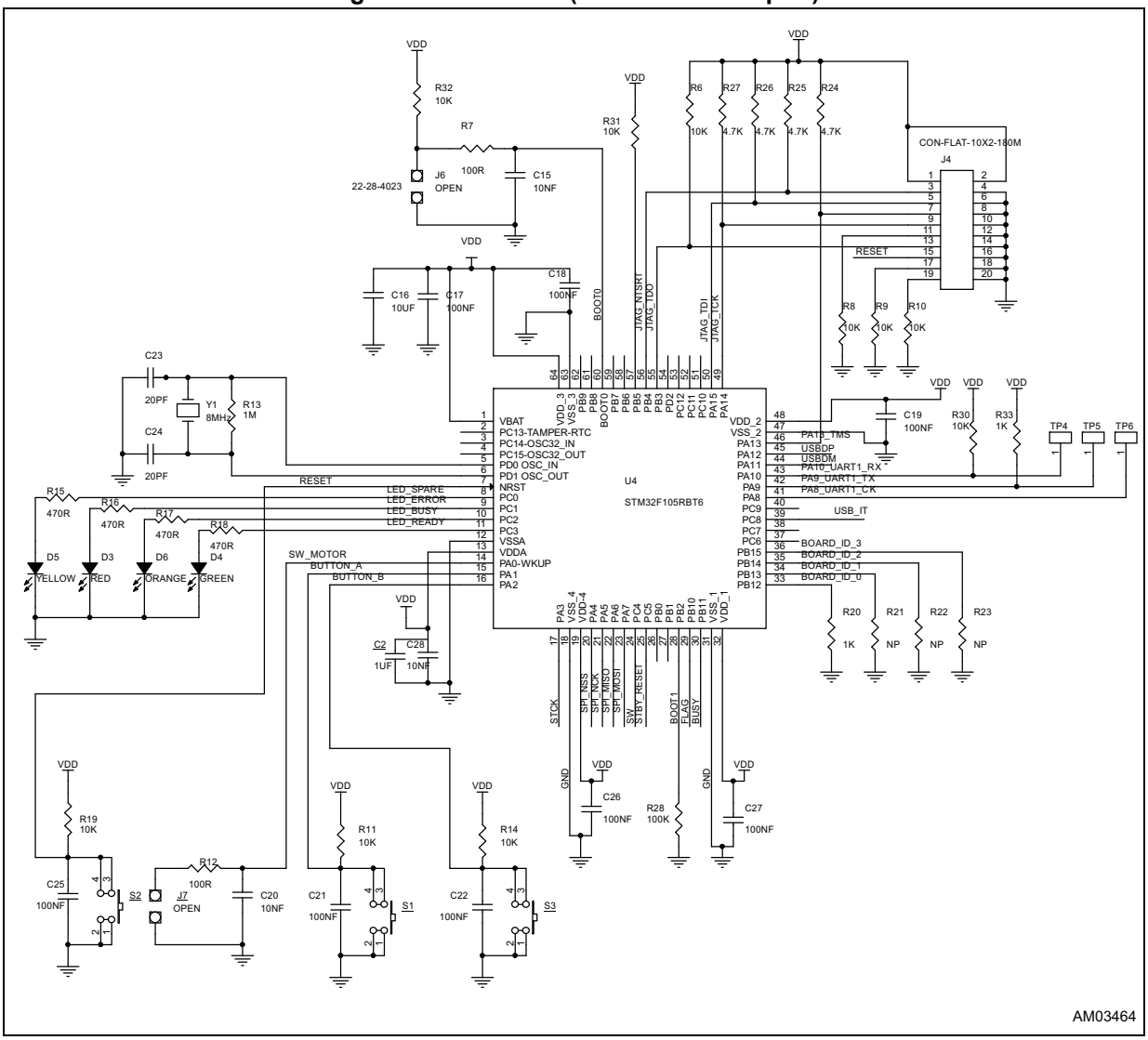


Figure 4. Schematic (motor driver part)

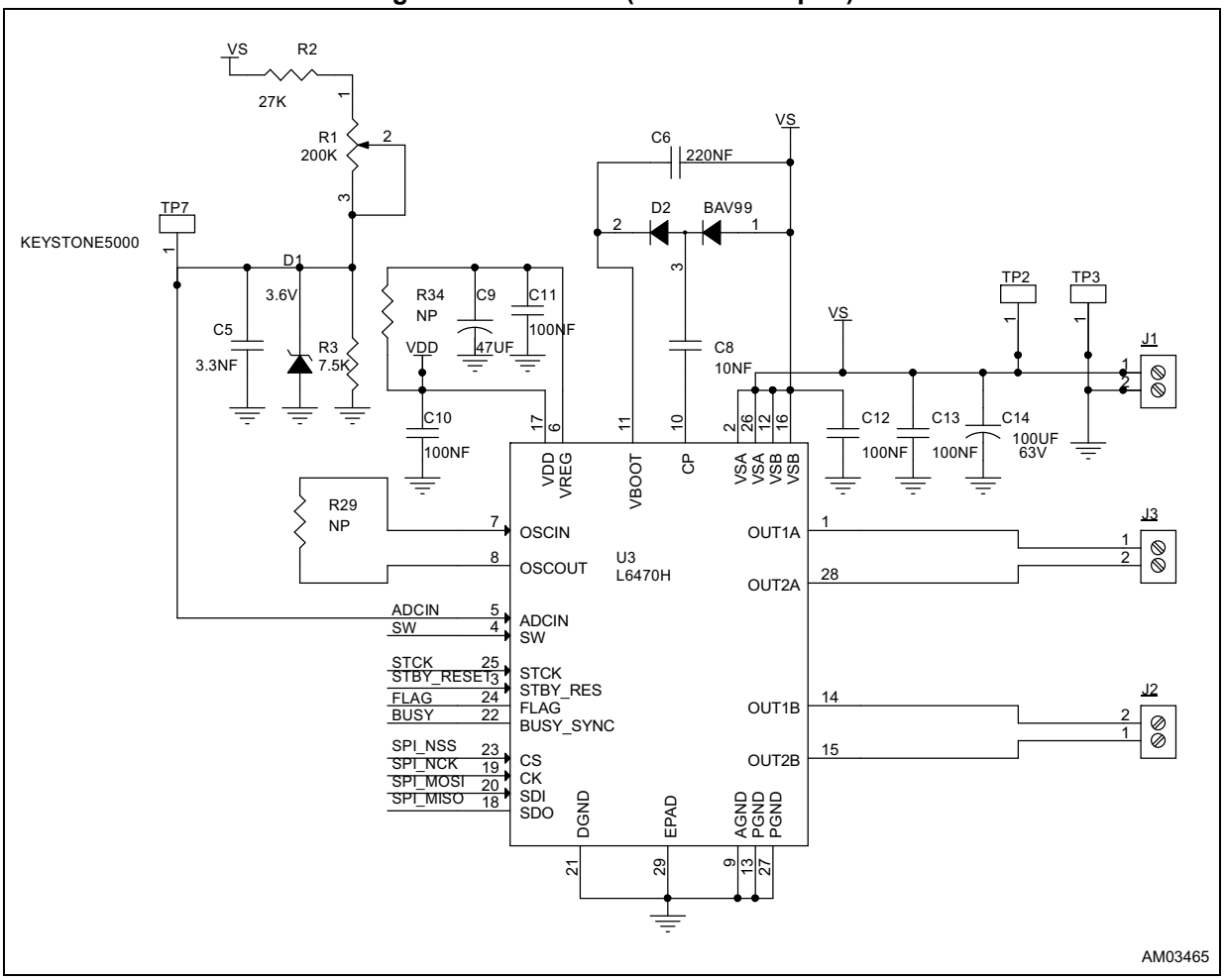


Figure 5. Layout top layer and bottom layer

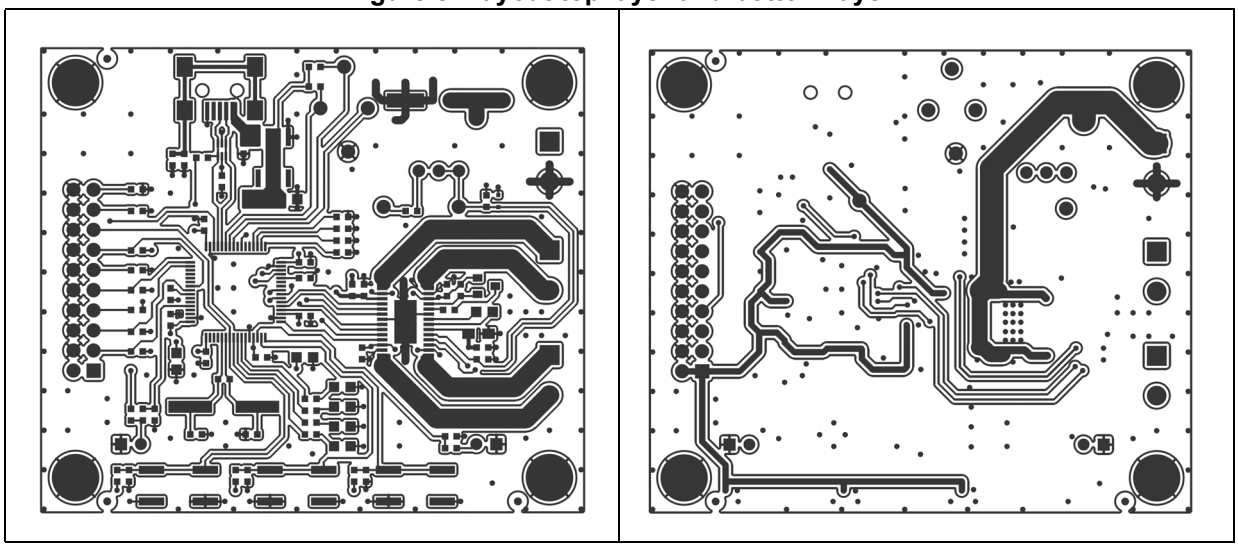


Table 2. Bill of material

Item	Quantity	Reference	Value	Package
C1, C16	2	Cap. cer. 10 µF 10 V X7R 0805	10 µF	0805
C2	1	Cap. cer. 1 µF 10 V X7R 0805	1 µF	0805
C3, C4, C10 - C13, C17 - C19, C21, C22, C25 - C27	14	Cap. cer. 100 nF 50 V X7R 0603	100 nF	0603
C5	1	Cap. cer. 3.3 nF 50 V X7R 0603	3.3 nF	0603
C6	1	Cap. cer. 220 nF 35 V X7R 0603	220 nF	0603
C7	1	Cap. cer. 4.7 nF 50 V X7R 0603	4.7 nF	0603
C8, C15, C20, C28	4	Cap. cer. 10 nF 50 V X7R 0603	10 nF	0603
C9	1	Cap. tant. 47 µF 6.3 V 10% PACK-A	47 µF	3216
C14	1	Cap. elec. 100 µF 63 V	100 µF	CAPES-R10HXX
C23, C24	2	CAP. CER. 20pF 50V COG 0603	20 pF	0603
D1	1	Zener regulator	3.6 V	SOD 523
D2	1	Double diode - high speed switching diode	BAV99	SOT23
D3	1	LED red - 0805 -2 mcd - 621 nm	Red	0805
D4	1	LED green - 0805 -6 mcd - 569 nm	Green	0805
D5	1	LED yellow - 0805 -6 mcd - 588 nm	Yellow	0805
D6	1	LED orange - 0805 -2 mcd - 602 nm	Orange	0805
FIX1 - FIX4	4	Hole	3 mn	-
J1 - J3	3	Screw connector 2 poles MKDSN 1.5/2 - 5.08	MKDSN 1.5/2 - 5.08	MKDSN 1.5/2 - 5.08
J4	1	JTAG CON-FLAT-10 x 2 - 180 M	CON-FLAT-10 x 2 - 180 M	CON-FLAT-10 x 2 -180 M
J5	1	USB_B_MINI_AMP_1734035-1	CN-USB	CMS Mini USB
J6, J7	2	JUMP254P-M-2	OPEN	STRIP 2 x 2.54
MIRE1 - MIRE3	3	OPTICAL_TARGET	OPTICAL_TARGET	Diam. 1 mn
R1	1	Trimmer 200 KΩ	200 KΩ	Trimm. 100 x 50 x 110
R2	1	27 KΩ 5% 1/10 W	27 KΩ	0603
R3	1	Res. 7.5 KΩ 5% 1/10 W 0603 SMD	7.5 KΩ	0603
R4, R13	2	Res. 1 MΩ 1/10 W 5% 0603 SMD	1 MΩ	0603
R5	1	Res. 1.5 KΩ 1/10 W 5% 0603 SMD	1.5 KΩ	0603
R6, R8 - R11, R14, R19, R30 - R32	10	Res. 10 KΩ 5% 1/10 W 0603 SMD	10 KΩ	0603

Table 2. Bill of material

Item	Quantity	Reference	Value	Package
R7, R12	2	Res. 100 Ω 5% 1/10 W	100 Ω	0603
R15 - R18	4	Res. 470 Ω 5% 1/10 W 0603	470 Ω	0603
R20, R33	2	Res. 1 KΩ 5% 1/10 W 0603 SMD	1 KΩ	0603
R21 - R23, R34	4	Res. NP 0603	NP	0603
R24 - R27	4	Res. 4.7 KΩ 5% 1/10 W 0603 SMD	4.7 KΩ	0603
R28	1	Res. 100 KΩ 5% 1/10 W 0603 SMD	100 KΩ	0603
R29	1	Res. NP 0805	NP	0805
S1 - S3	3	Switch button SMD	EVQQ2D03W	CMS 6.5 x 6 x 3.1
TP1, TP2, TP4 - TP7	6	Test point red	KEYSTONE-5000	TH
TP3	1	Test point black	KEYSTONE-5001	TH
U1	1	IC REG 1300MA LN 3.3 V	LD1117D33TR	SO8
U2	1	USBLC6-2P6	USBLC6-2P6	SOT 666
U3	1	L6470 microstepping motor driven	L6470	HTSSOP28
U4	1	IC, MCU, RISC, 72 MHz, 3.6 V, 32-bit, 64-pin, LQFP	STM32F105RBT6	LQFP64 10 x 10
Y1	1	XTAL 8 MHz-30 PPM-20 pF	8 MHz	HC49/US-SM

4 General description

4.1 Power supply

The EVAL6470H-DISC board is designed to be powered via:

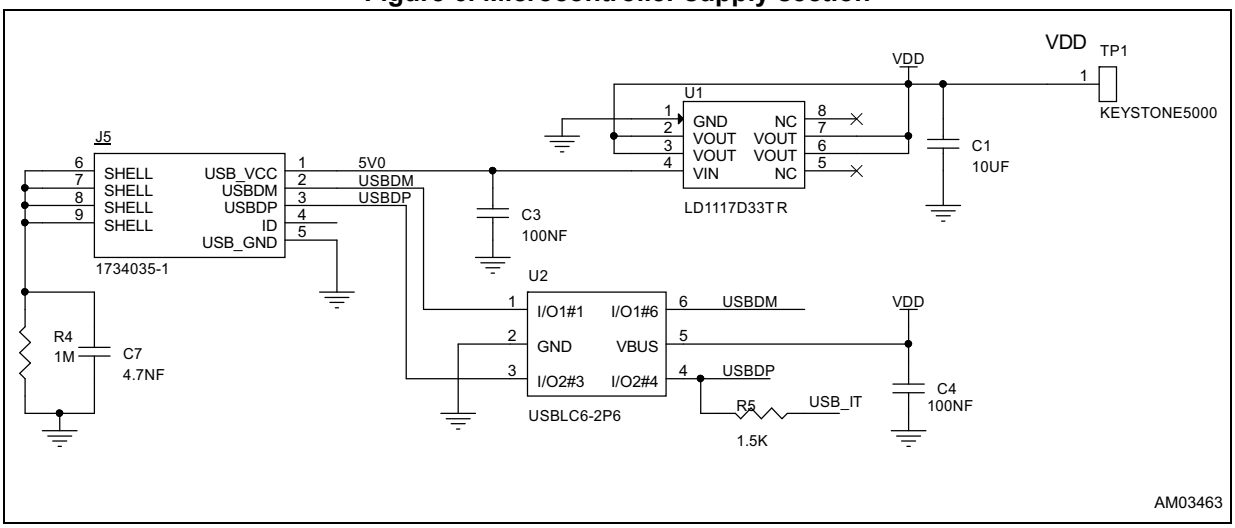
- Connector J1: power of the motor and also motor control driver.
- USB connector J5: power of the microcontroller and logic control.

The USB cable supplies the digital part through a dedicated LDO (U1) providing 3.3 V.

The motor power must be set according to the voltage required by the user motor.

Note: ***Both the supply sources (USB connector and J1 connector) must be present to make the board operative.***

Figure 6. Microcontroller supply section



4.2 L6470 stepper motor driver

The L6470 is an advanced fully integrated solution suitable for driving two-phase bipolar stepper motors with microstepping. It integrates a dual low $R_{DS(on)}$ DMOS full bridge.

Features

- Operating voltage: 8 - 45 V
- 7.0 A out peak current (3.0 $A_{r.m.s.}$)
- Low $R_{DS(on)}$ power MOSFETs
- Programmable speed profile and positioning
- Programmable power MOS slew rate
- Up to 1/128 microstepping
- Sensorless stall detection
- SPI interface
- Low quiescent and standby currents
- Programmable non dissipative overcurrent
- Two levels of overtemperature protection

Figure 7. L6470 block diagram

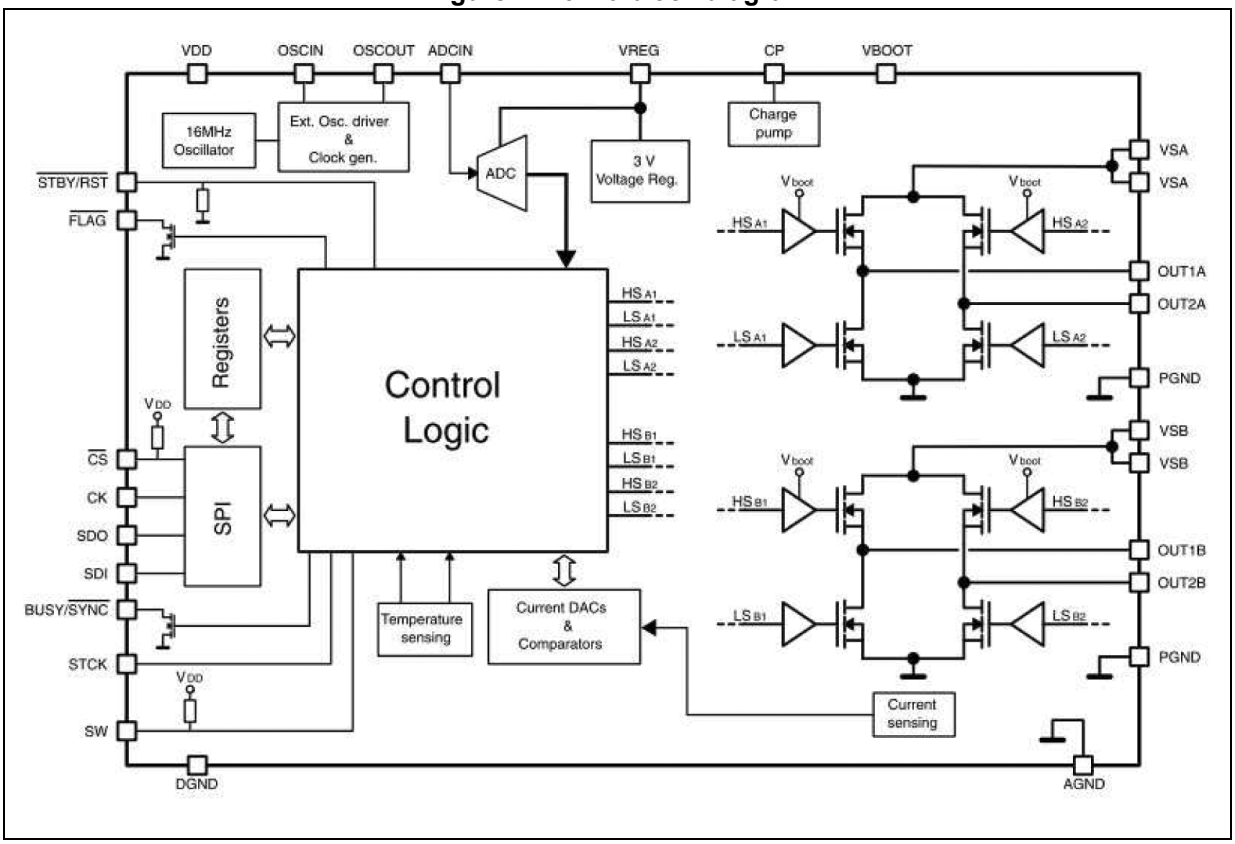


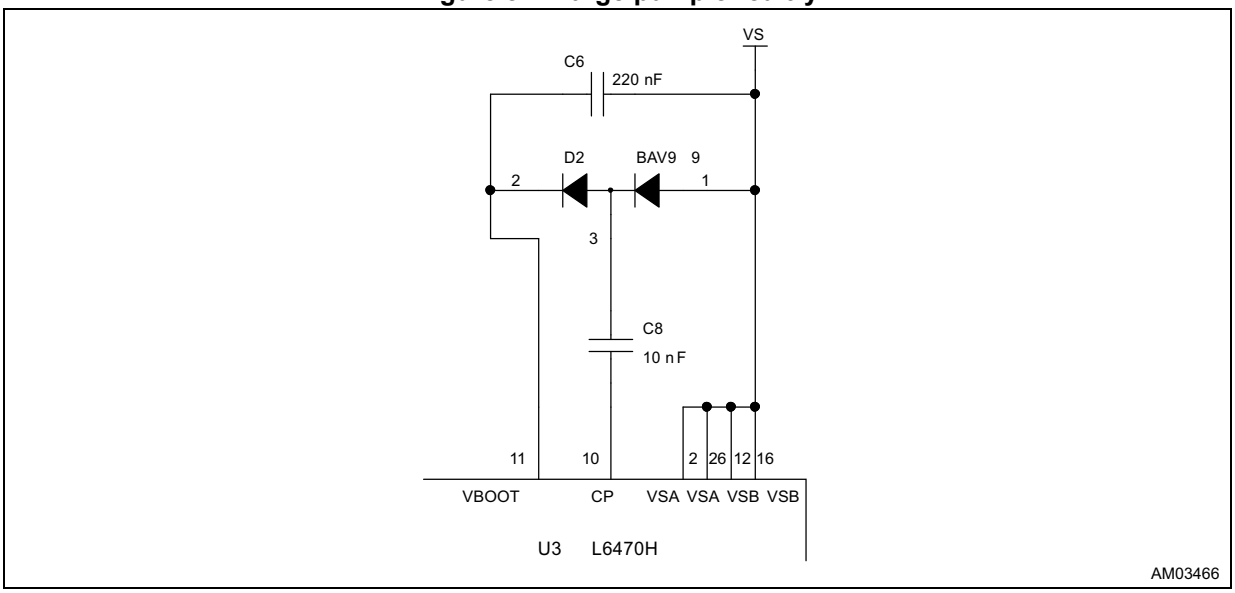
Table 3. L6470 recommended operating conditions

Symbol	Parameter	Test condition	Value	Unit
V_{DD}	Logic Interface supply voltage		3.3	V
V_S	Motor supply voltage	$V_{SA} = V_{SB} = V_S$	8 ÷ 45	V
V_{OUT_diff}	Differential between voltage V_{SA}, OUTI1A, OUT2A, PGND and V_{SB}, OUT1B, OUT2B, PGND pins	$V_{SA} = V_{SB} = VS$	Up to 45	V
$V_{REG\ in}$	Logic supply voltage	V_{REG} internal	3	V
V_{ADC}	Integrated ADC input voltage range (ADCIN pin)		0 ÷ V_{REG}	V

4.2.1 Charge pump

The L6470 device uses an internal charge pump for driving correctly the integrated MOSFETs, a voltage higher than the motor power supply. The charge pump is obtained through an oscillator and few external components.

Figure 8. Charge pump circuitry



4.2.2 Voltage mode driving

The configuration parameters of the voltage mode driving can be obtained through the BEMF compensation tool embedded into the SPINFamily software.

A wrong setup of these parameters could cause several issues, in particular:

- The phase current decreases with the speed and the motor will stall.
- The wrong voltage is applied to the motor and the system is very noisy.
- The phase current reaches the overcurrent limit.

The BEMF compensation form uses the application parameters as inputs in order to evaluate the proper device setup.

The required inputs are:

- Supply voltage.
- Target phase current (r.m.s. value) at different motion conditions (acceleration, deceleration, constant speed and holding).
- Target operating speed (maximum speed).
- Motor characteristics.

The motor characteristics are: electrical constant (Ke), phase inductance and resistance. The inductance and the resistance of the phase are given in the motor datasheet. The Ke is rarely given in the specification and must be measured.

In the help section of the SPINFamily software a step by step procedure is explained. The same procedure can also be found in the application note “AN4144: Voltage mode control operation and parameter optimization” on www.st.com.

Click on the “evaluate” button to get the suggested setup for the voltage mode driving. Then click on “write” button to copy the data in the registers of the device.

4.2.3 Overcurrent and stall detection thresholds

The overcurrent protection and the stall detection are implemented measuring the current flowing into each integrated MOSFET.

The overcurrent protection threshold should be set just above the current rating of the motor:

$$I_{OCDth} > I_{max,r.m.s.} \times \sqrt{2}$$

For example: if the maximum phase current of the motor is 2 $A_{r.m.s.}$, the overcurrent protection should be set to about 3 A.

Warning: Important - it is strongly discouraged to disable the overcurrent shutdown. It may result in critical failures.

The stall detection threshold should be just above the operating peak current of the application. During the preliminary stages of evaluation, it can be set to the maximum value.

4.2.4 Speed profile

The max. speed parameter is the maximum speed the motor will run. By default, it is about 1000 step/s. That means, if you send a command to run at 2000 step/s, the motor speed is limited at 1000 step/s.

This is an important safety feature in the final application, but not necessarily useful to evaluate the device performances. Setting the parameter to high values (e.g. 6000 step/s) allows evaluating the maximum speed which can be achieved by the application under test through the speed tracking command (Run), but it probably limits the possibility to use positioning commands (Move, GoTo, etc.).

The Full-step speed parameter indicates the speed at which the system switches from microstepping to full step operation.

In voltage mode driving devices it is always recommended to operate in microstepping and not to switch to full step. Hence, this parameter should be greater than the maximum speed.

4.3 STM32F105RB microcontroller

The STM32F105xx incorporates the high-performance ARM® Cortex™-M3 32-bit RISC core operating at a 72 MHz frequency, high-speed embedded memories (a Flash memory up to 256 Kbytes and an SRAM 64 Kbytes), and an extensive range of enhanced I/O and peripherals connected to two APB buses. All devices offer two 12-bit ADCs, four general-purpose 16-bit timers plus a PWM timer, as well as standard and advanced communication interfaces: up to two I^2Cs, three SPIs, two I^2Ss, five USARTs, a USB OTG FS and two CANs.

The STM32F105xx device operates in the -40 to +105 °C temperature range, from a 2.0 to 3.6 V power supply. A comprehensive set of power saving mode allows the design of low-power applications.

The STM32F105xx offers devices in three different package types: from 64 pins to 100 pins. Depending on the device chosen, different sets of peripherals are included.

These features make the STM32F105xx and STM32F107xx connectivity line microcontroller family suitable for a wide range of applications such as motor drives and application control, medical and handheld equipment, industrial applications, PLCs, inverters, printers and scanners, alarm systems, video intercom, and HVAC and home audio equipment.

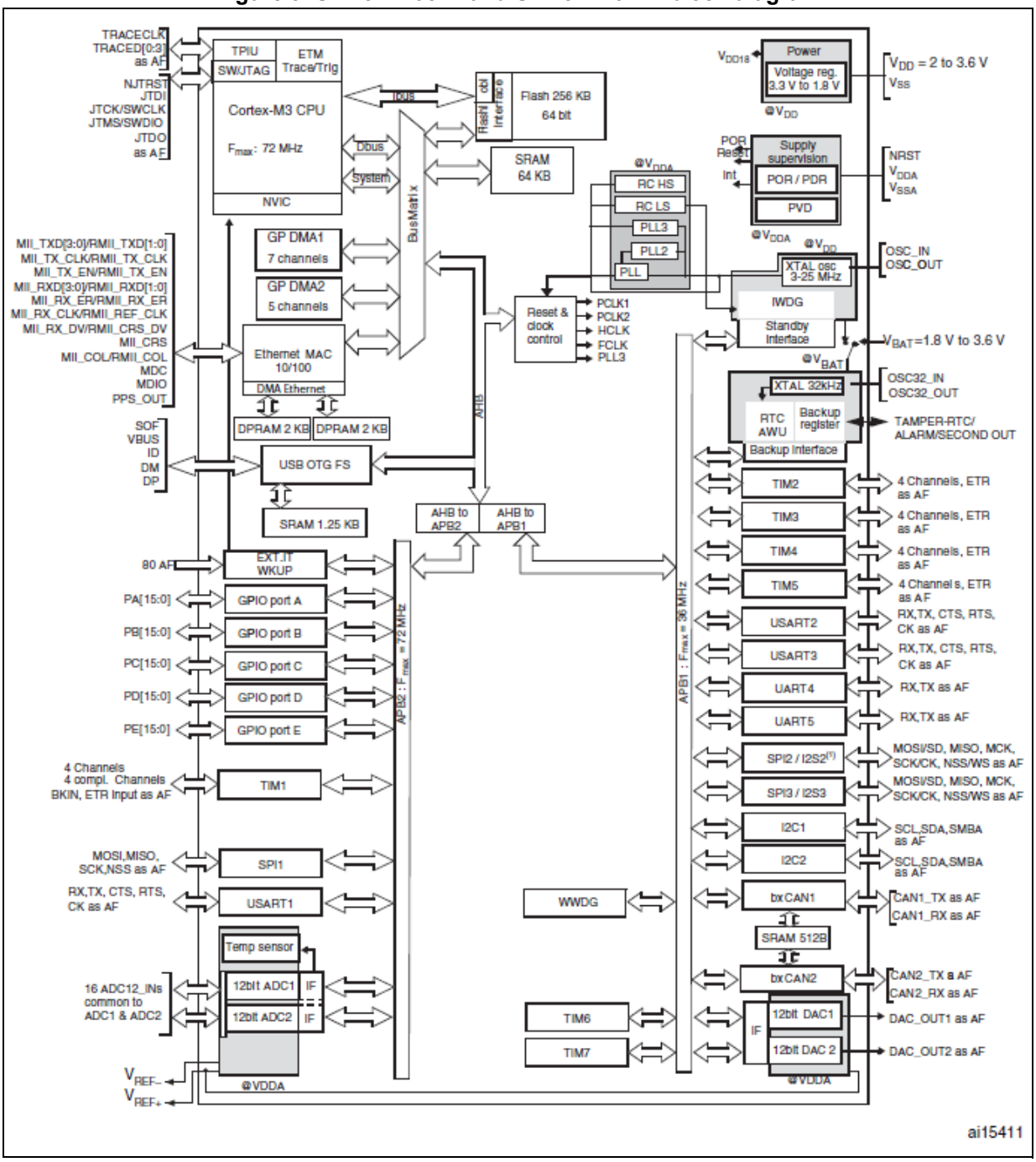
Please refer to the STM32F105xx datasheet for an overview of the complete range of peripherals proposed in this family.

Please refer to the STM32F105xx reference manual to get more information on the microcontroller operation.

The STM32F105RBT6 has a 64-pin LQFP package with a 128-KByte Flash memory and operates in the -40 to +85°C temperature range.

Figure 9 shows the general block diagram of the STM32F105xx and STM32F107xx family.

Figure 9. STM32F105xx and STM32F107xx block diagram



4.4 Firmware loading

This section describes how to load firmware to the board by using the DfuSe demonstration software.

4.4.1 DfuSe installation

You need first to download the DfuSe demonstration software from: www.st.com.

The DfuSe tool is referenced under the development suite STSW-STM32080.

Once downloaded, run the setup.exe file.

More details on the DfuSe are given in the UM0412 user manual.

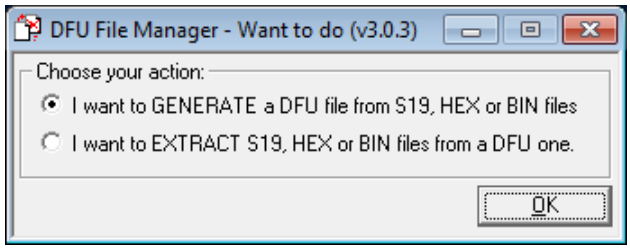
4.4.2 Generate a DFU file from a HEX file

If the file you want to download to the discovery board is not a DFU file but a HEX file, you will need first to convert it.

In this purpose:

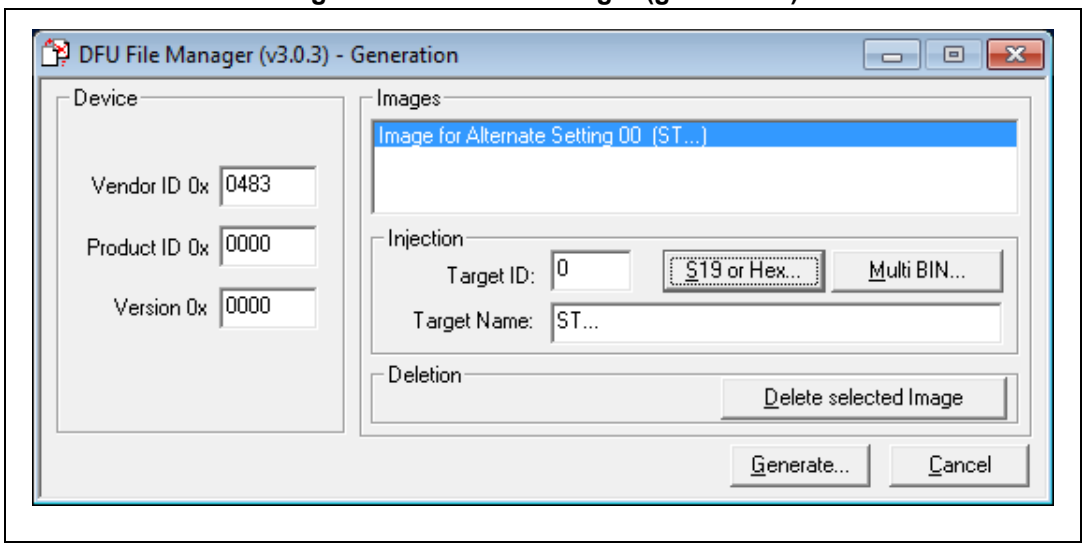
- Start the DFU file manager (V3.0.3 or greater) which has been installed with the DfuSe.
- Choose “I want to GENERATE a DFU file from S19, HEX or BIN files”.

Figure 10. DFU file manager (action)



- Click “S19 or HEX”… button.
- Select in the open dialog box the file of type “hex Files”, select the HEX file and click “OK”.
- Click on the “Generate…” button.
- Give a name to the *.DFU file and click on the “Save” button.

Figure 11. DFU file manager (generation)



4.4.3 Board settings

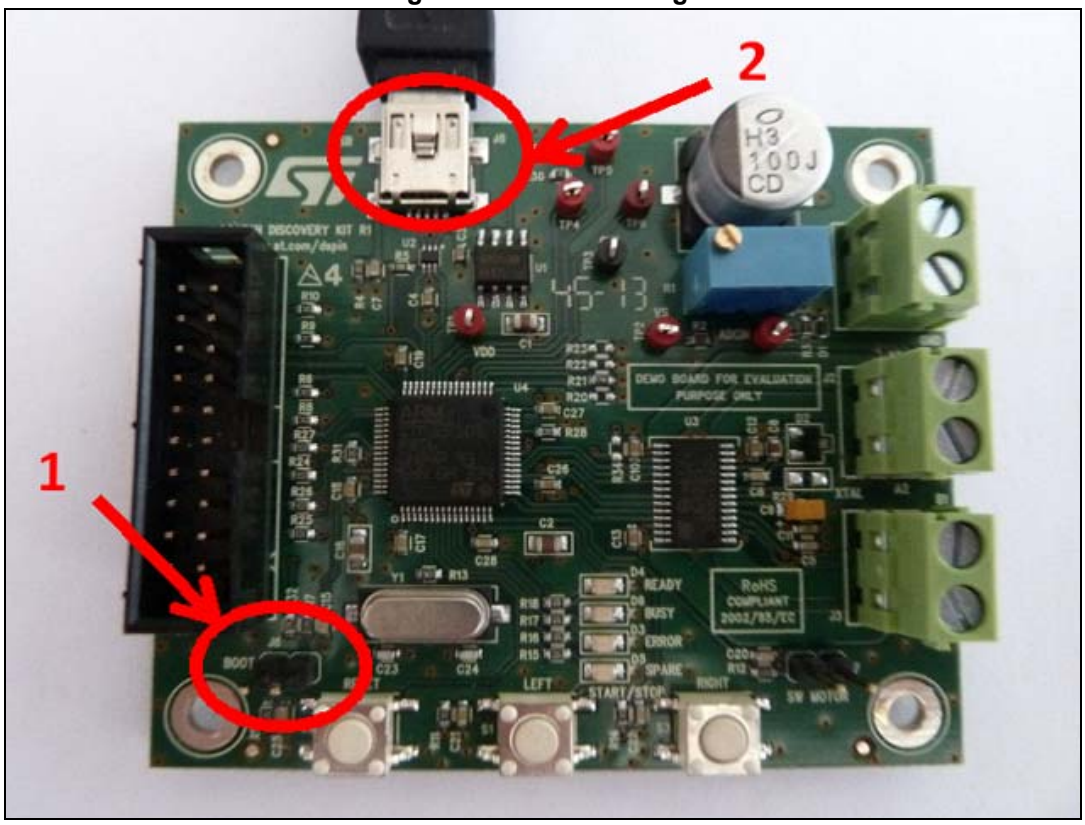
To be able to download firmware, the discovery board should be started in the “DFU” mode.

In this purpose:

1. Remove the jumper from the BOOT pins.
2. Plug a USB cable between the discovery board and the PC.

It does not matter if the VS connector is plugged or not to a supply voltage.

Figure 12. Board settings

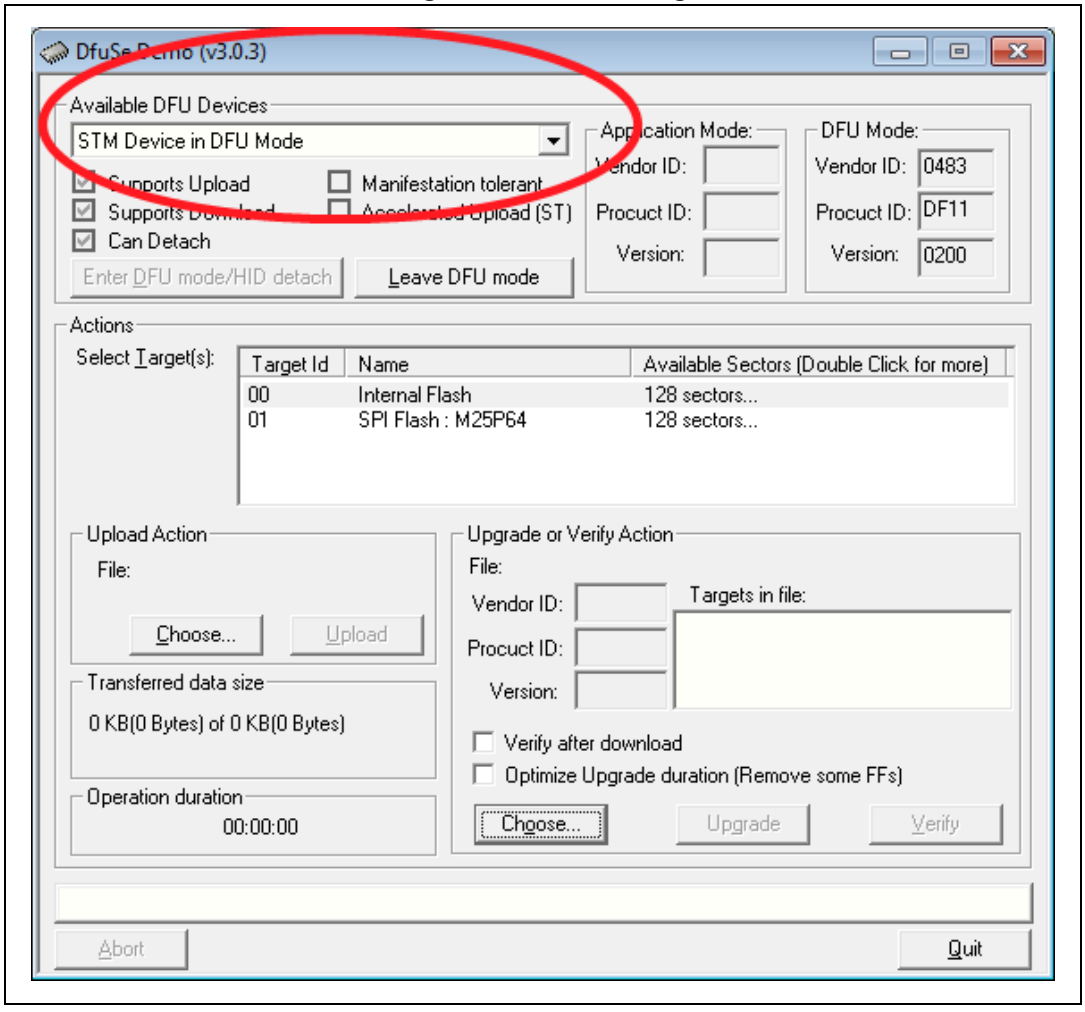


4.4.4 DFU loading

At this step, you are now ready to perform the firmware upgrade.

1. Start the “DfuSeDemo.exe”.
2. You must have an “STM Device in DFU Mode” in the list of the “Available DFU Devices”. Else, it means that your board is not correctly configured or not connected to the PC.

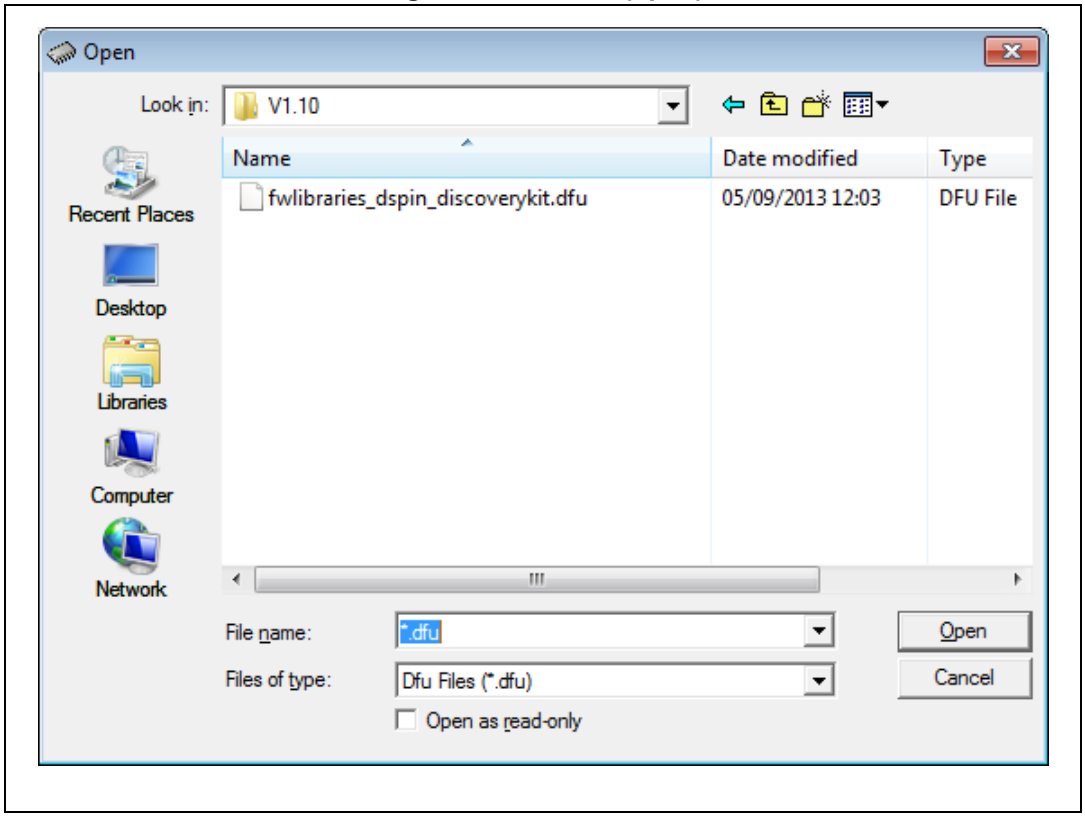
Figure 13. DFU loading



3. In the “Upgrade or Verify Action” group, click on the “Choose…” button.

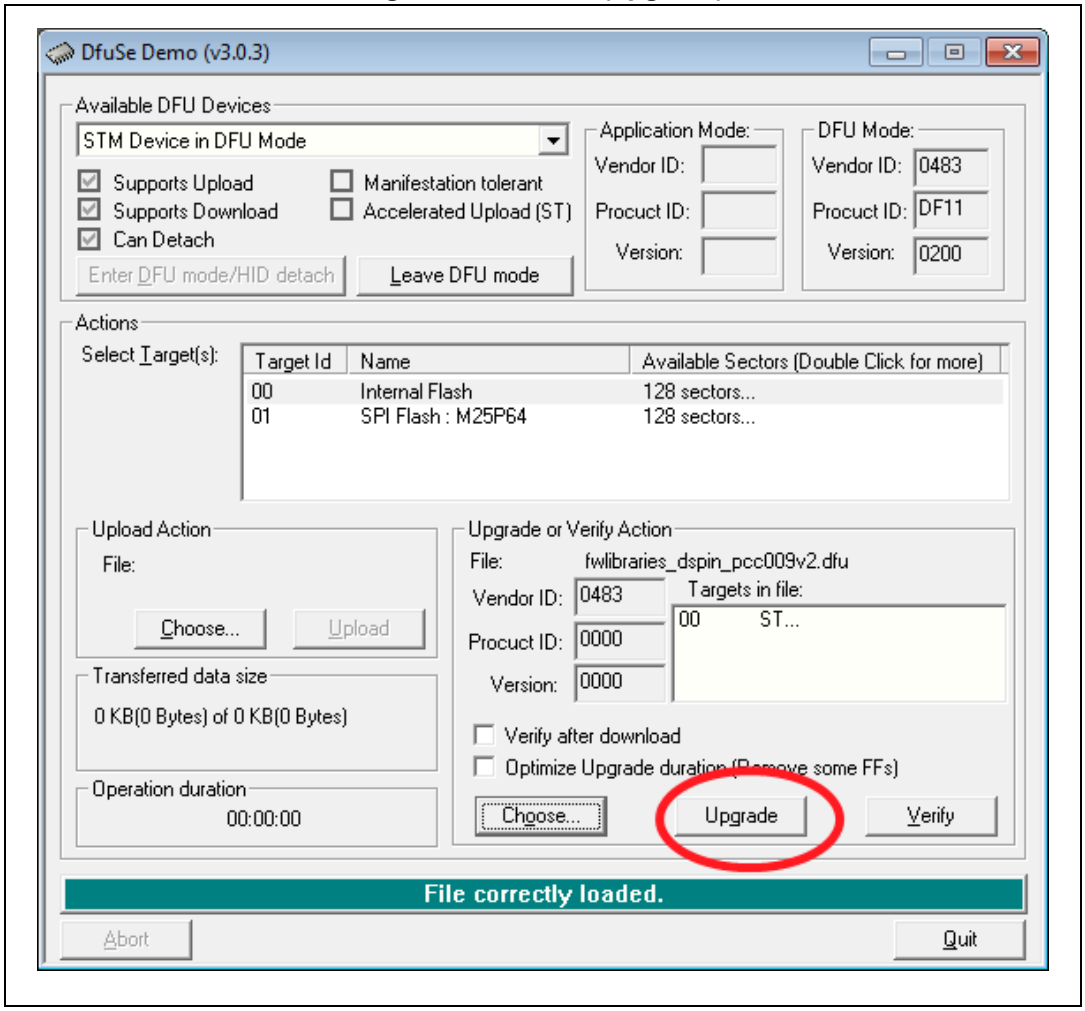
4. Select the *.dfu file of your choice in the open dialog box and click on the “Open” button.

Figure 14. DFU file (open)



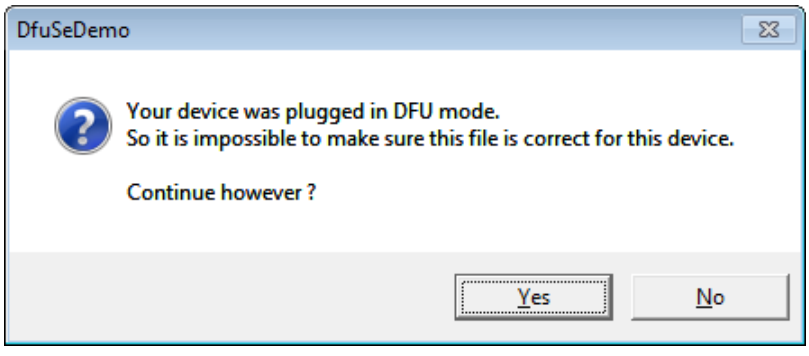
5. Click on the “Upgrade” button.

Figure 15. DFU file (upgrade)



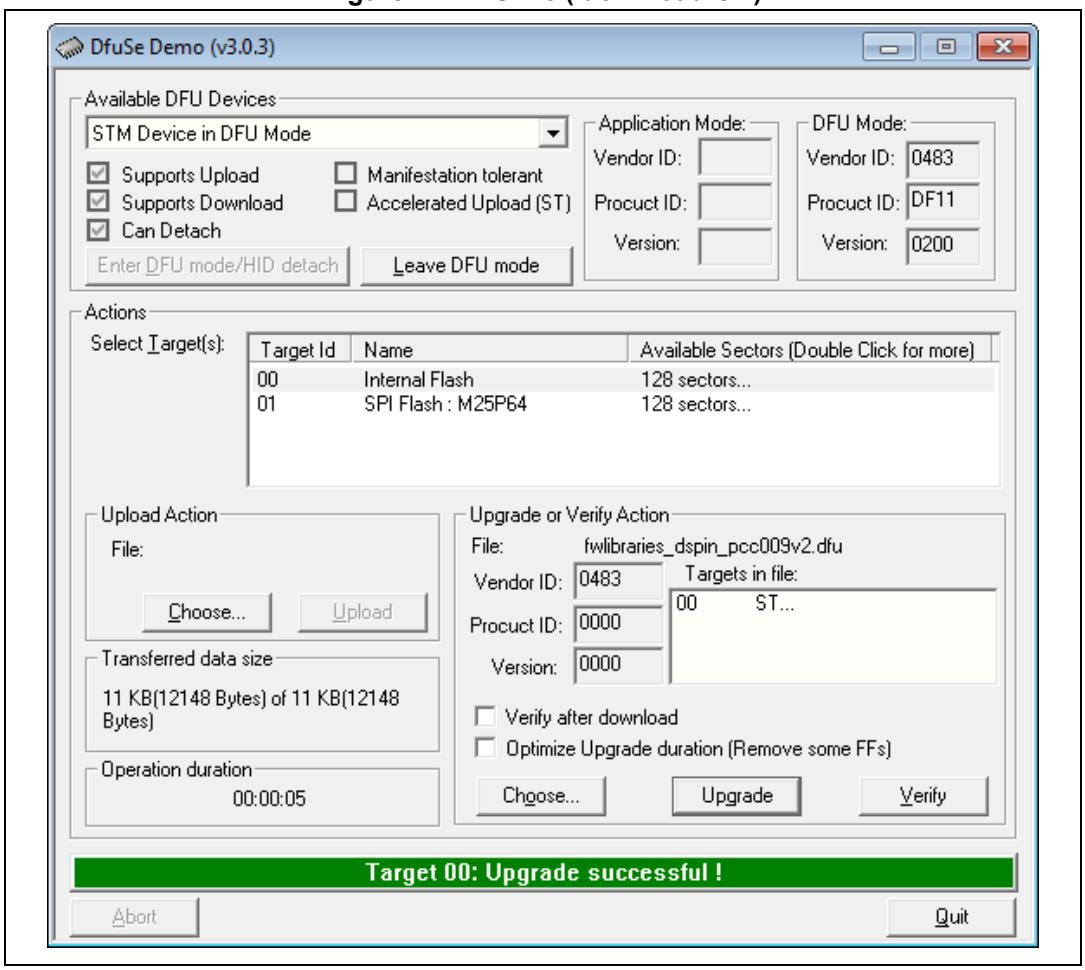
6. If this dialog box appears, click “Yes”.

Figure 16. DFU file (confirmation)



7. Once the download is performed, you should have:

Figure 17. DFU file (download OK)



8. Do no forget to put the jumper back on the BOOT pins in order to restart the discovery board to the normal mode!

4.5 Using the EVAL6470H-DISC with firmware for the GUI

By default the discovery board is loaded with firmware offering the capability to connect the board with the SPINFamily evaluation tool. This GUI provides direct access to all L6470 registers and allows sending application commands.

4.5.1 Sanity check of the board with firmware for the GUI

At the startup of the FWGUI, a sanity check is performed to confirm the discovery board is working correctly. The status is returned via the board LEDs.

To have a correct execution of the sanity check, please follow the steps below:

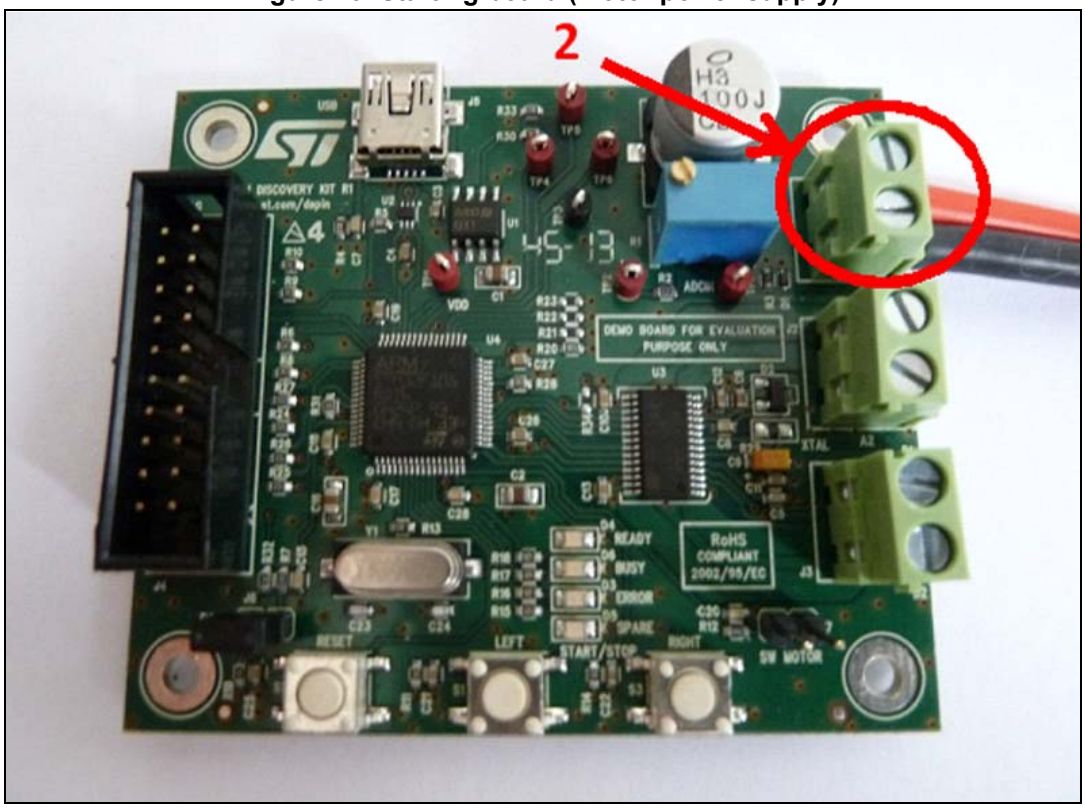
1. Place a jumper on the BOOT pins (bottom left corner of the board):

Figure 18. Starting board (boot mode)



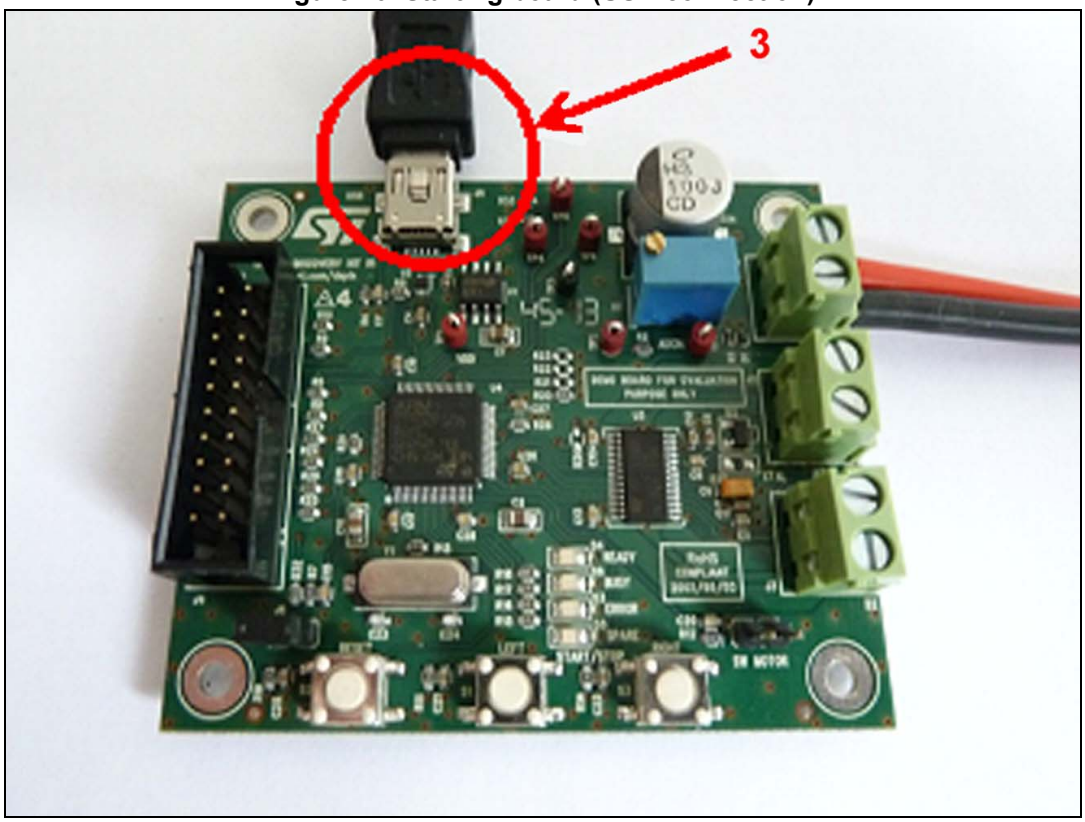
2. Connect the board to a 5 V - 45 V DC power supply:

Figure 19. Starting board (motor power supply)



3. Plug a USB cable (which must at least provide a power supply).

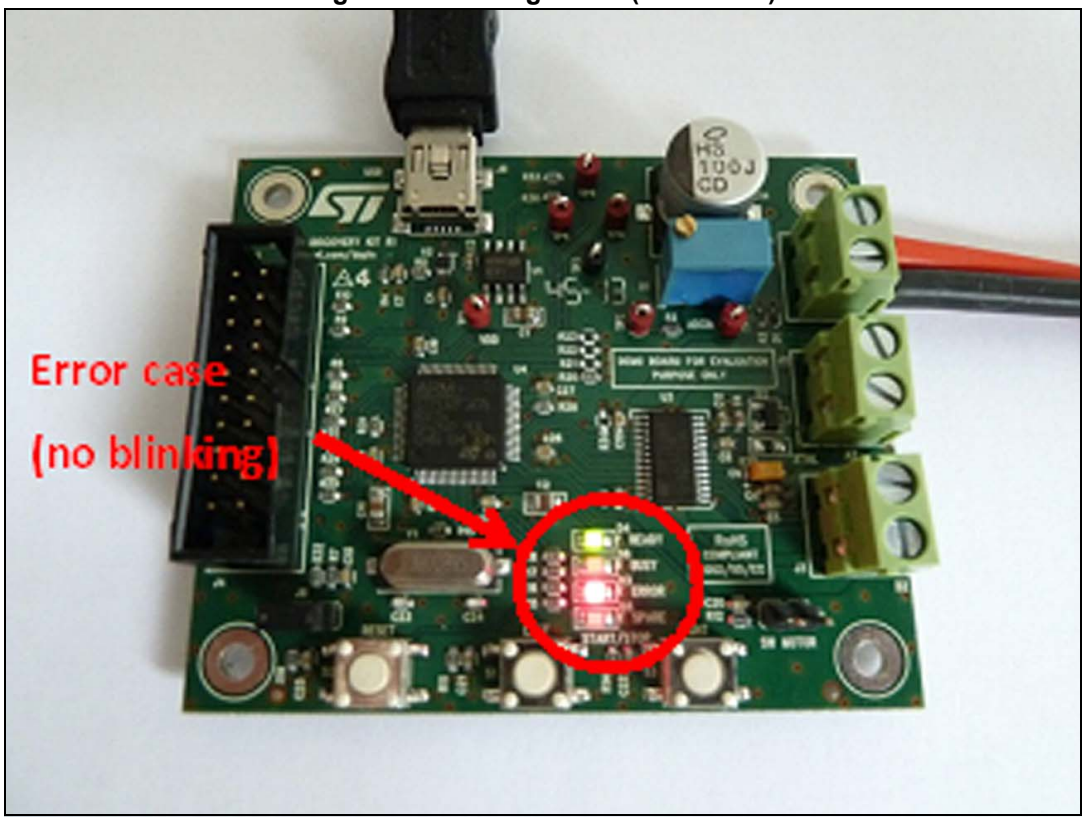
Figure 20. Starting board (USB connection)



4. The board should switch on automatically.

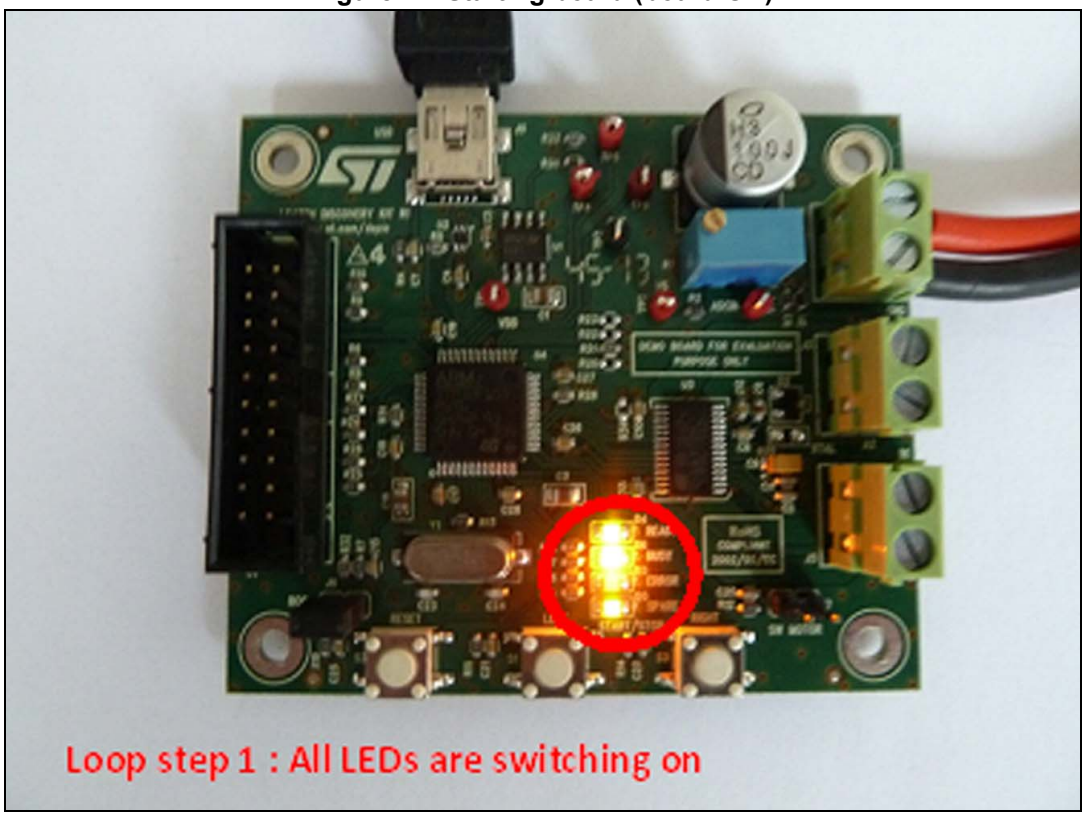
5. At this step:
   a) **If a problem is detected**, the “ready” LED (green) and the “error” LED (red) will switch on without blinking. This means that:
   – either the board ID is not recognized by the FW (bad FW versions used)
   – or there is a problem with the SPI (no connection between the MCU and the L6470 via the SPI)
   – or there is no 5 V - 45 V DC power supply.

Figure 21. Starting board (error case)



b) **If no problem is detected**, the LEDs will start an infinite two-step loop:
In the first step, the four LEDs will switch on one after the other by starting by the green one and ending by the yellow one.

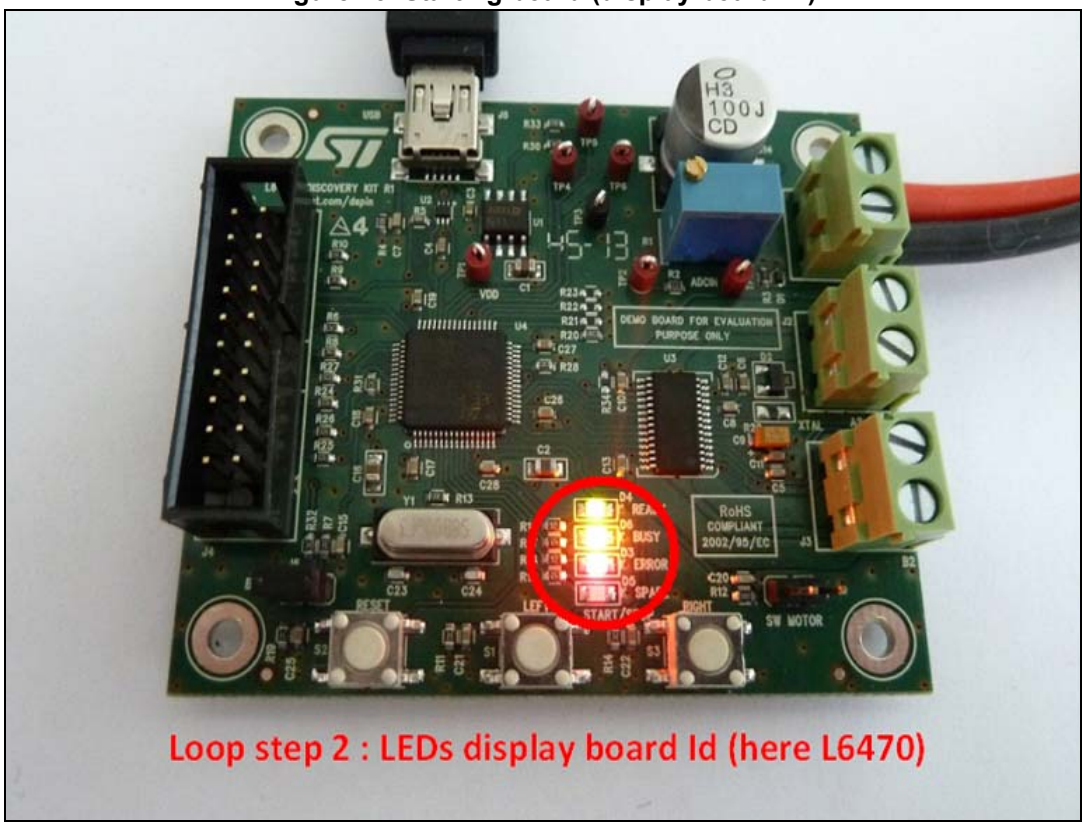
Figure 22. Starting board (board OK)



In the second step, only the LEDs which correspond to the board ID are switched on all at the same time.

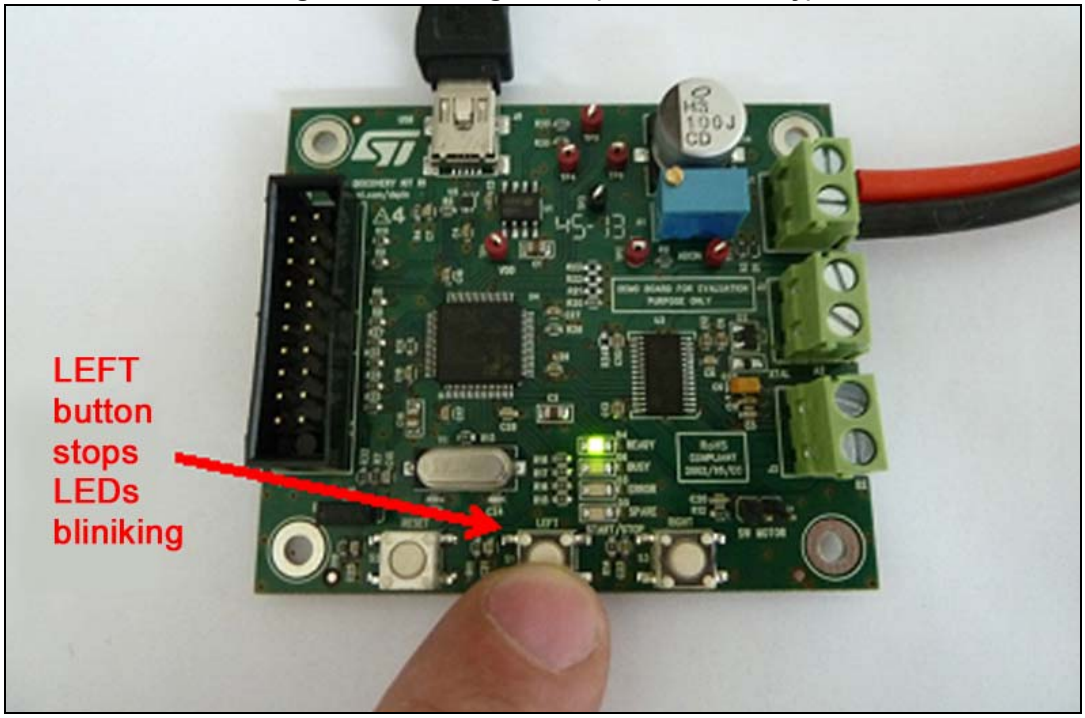
For the L6470 device, there are green, orange, red LEDs.

Figure 23. Starting board (display board ID)



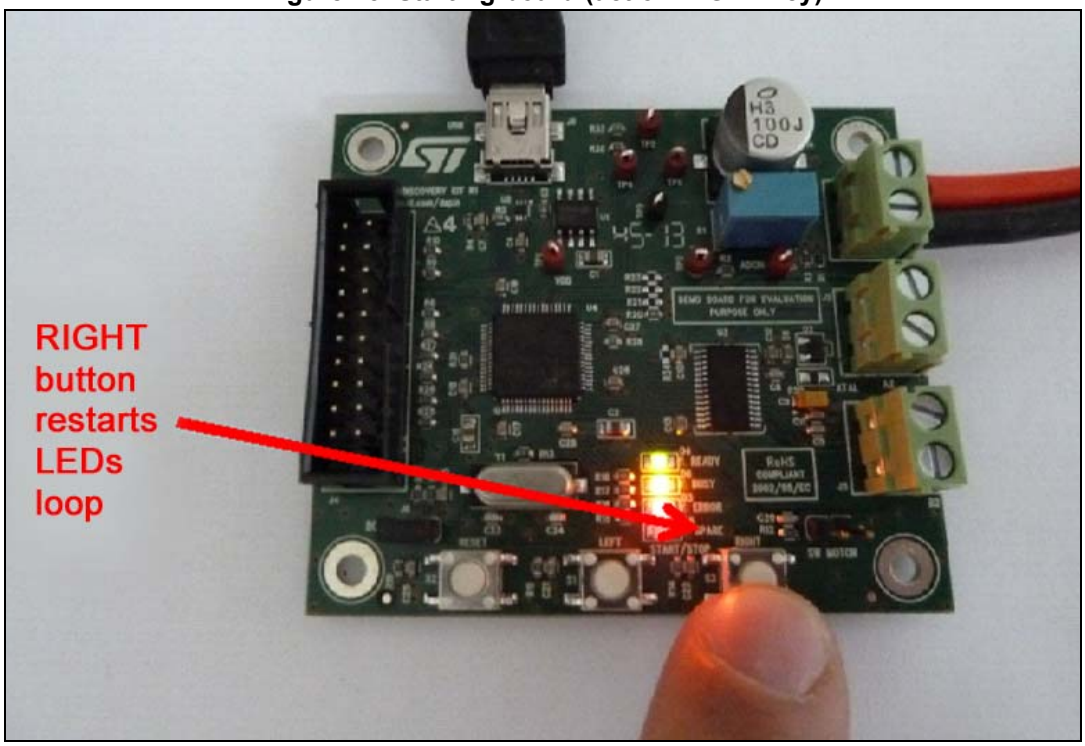
6. Press the LEFT button and check the two-step loop stops after a few seconds. Only green LED remains switched on.

Figure 24. Starting board (action LEFT key)



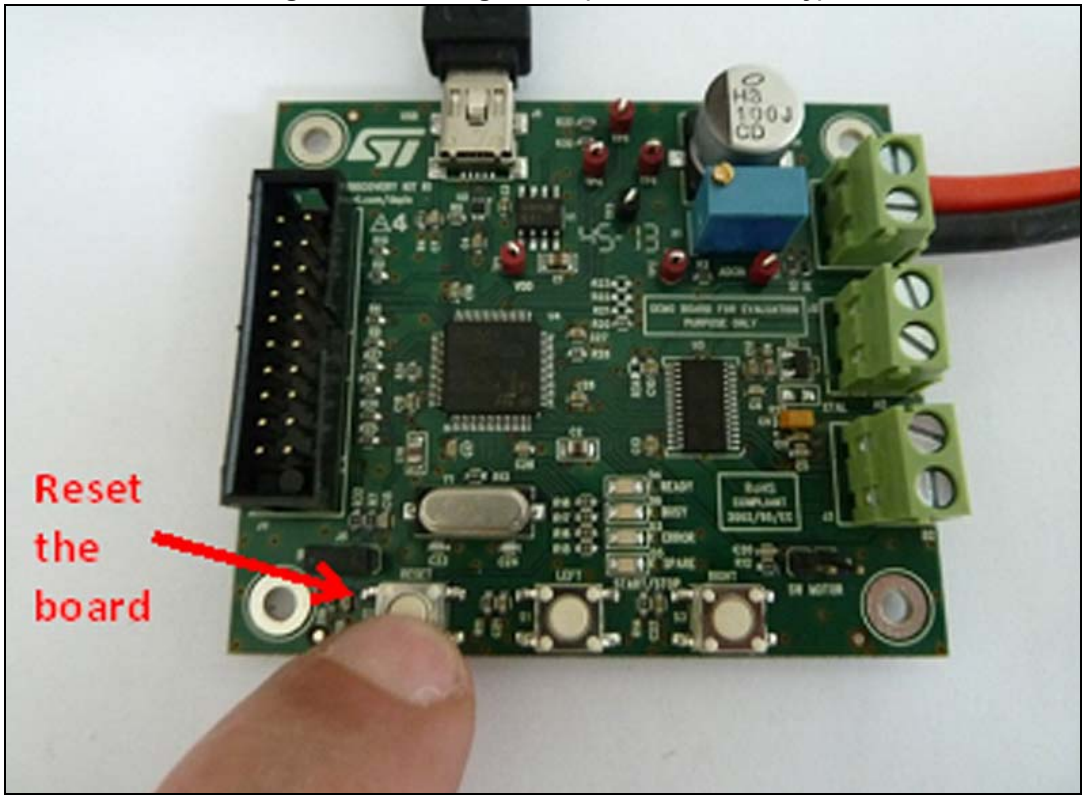
7. Press the RIGHT button and check the two-step loops restarts.

Figure 25. Starting board (action RIGHT key)



8. Press the RESET button and check that the LEDs restart there two-step loop after the board reset.

Figure 26. Starting board (action RESET key)



Note: Please note that once you have connected the discovery board to the GUI, the LEDs meaning is different. You then need to restart the board to perform a new auto-check and to have a valid status from the LEDs.

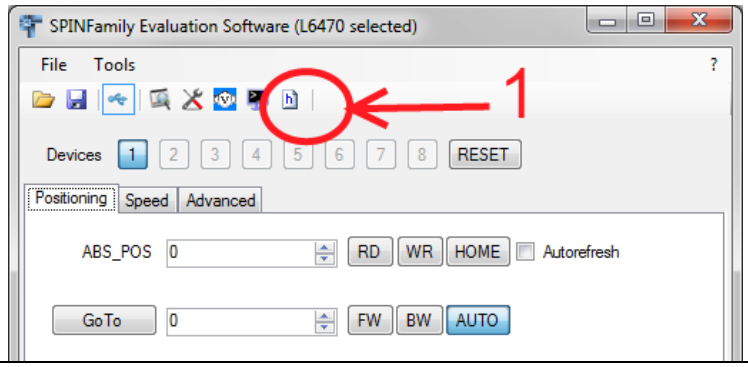
4.5.2 Parameters exportation from the GUI to the FW library

Once you have customized the L6470 parameters with the GUI, you can export them to a header file in order to use it with the FW library.

In this purpose:

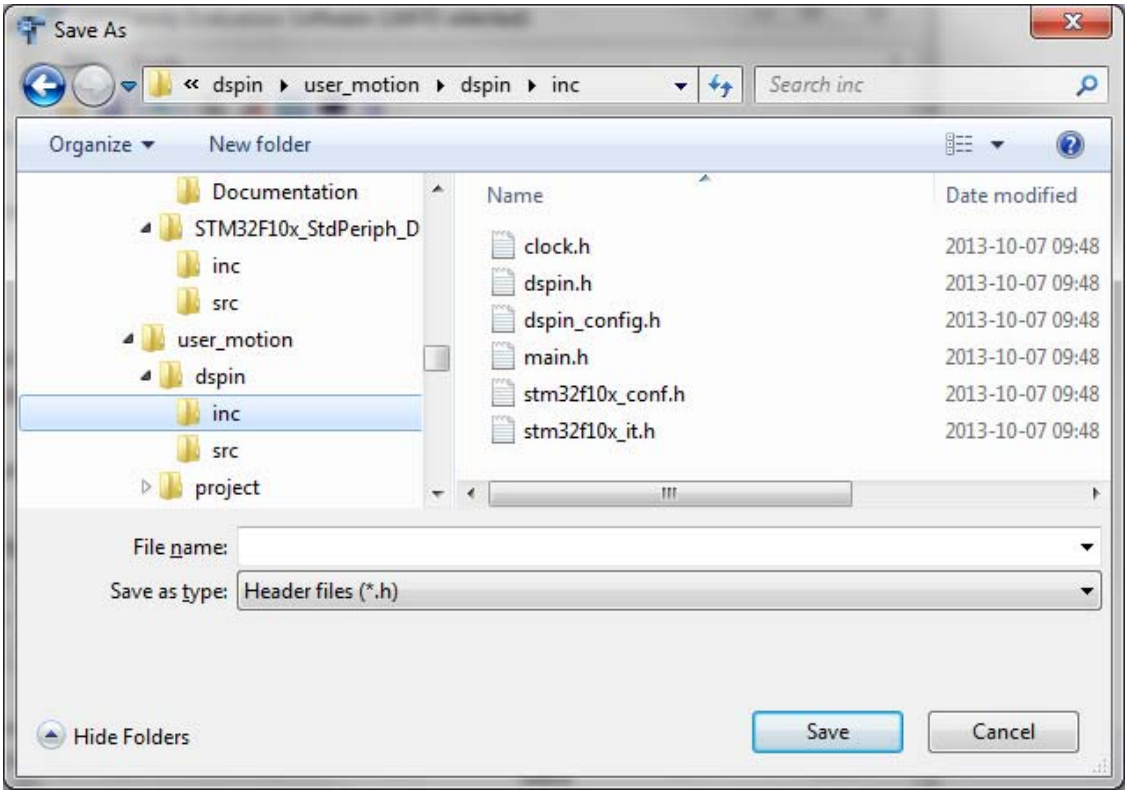
1. Press the “header file” button on the main window of the GUI.

Figure 27. Parameters exportation



2. Replace the existing “dspin_config.h” of your current FW library by the new one.

Figure 28. Parameters exportation (save file)



3. Then you only need to recompile your project as usual to use the exported parameters.

4.6 Using the EVAL6470H-DISC with the firmware library

The L6470 firmware library is supplied as an IAR workspace with a source include and project files. If you are using an IAR design environment, you just need to load the “dspin.eww” file and use the fwlibrairies_dspin_discoverykit project which is already active.

If you are using a different design environment, you will find instructions in this user manual to build a new project on your preferred IDE.

The L6470 firmware library is also supplied as an executable file in a HEX and in a DFU format. It can be loaded into the EVAL6470H-DISC board as explained in *Section 4.4 on page 17*.

4.6.1 FW library package contents

- FW L6470 library
  - Described in *Section 4.6.2*.
- CMSIS library
  - stm32f10x/CMSIS Library used by the L6470 FW library
- STM32F10x standard peripherals library drivers
  - /stm32f10x/STM32F10x_StdPeriph_Driver Library used by the L6470 FW library
- IAR workspace files
  - /user_motion/project/ewarm6/fwlibraries/dspin/dspin.eww
    - a workspace file
  - /user_motion/project/ewarm6/fwlibraries/dspin/settings/dspin.wsdt
    - a workspace settings file
  - /user_motion/project/ewarm6/fwlibraries/dspin/discoverykit
    - a directory containing the L6470 discovery board project files and subdirectories
  - /user_motion/project/ewarm6/fwlibraries/dspin/discoverykit/Debug/Exe/
    - a directory containing the *.HEX and *.DFU executable files
  - /user_motion/project/ewarm6/fwlibraries/dspin/pcc009v2
    - a directory containing the PCC009V2 board project files and subdirectories.

4.6.2 FW L6470 library description

The FW L6470 library has the following features:

- Register read, write and check
- Register values conversion
- Device configuration
- Motion commands
- FLAG and BUSY interrupts management
- Button interrupts management
- Step “Clock mode” management
- Initialization routine using “GoUntil” and “ReleaseSW” commands
- “Daisy Chain” mode

The FW L6470 library has been tested on the L6470H DISCOVERY board R1. The main program contains a commented demonstration sequence which uses all the FW L6470 library supported features. This sequence is interactive and to proceed to the end some user action is necessary.

For the “GoUntil” feature demonstration, when the LED SPARE is blinking for the first time, the user shall close the SW MOTOR jumper J8. This triggers a switch turn on event at the L6470 SW pin.

For the “ReleaseSW” feature demonstration, when the LED SPARE is blinking again, the user shall open the SW MOTOR jumper J8. This triggers a L6470 SW pin release.

At the end of the demonstration sequence, the GPIO connected to LEFT and RIGHT buttons are configured to trigger interrupts on the microcontroller:

On a LEFT button press, the microcontroller starts the motor at quarter of max. speed if it is stopped or doubles the motor speed if it is already running.

On a LEFT button press, the microcontroller disables the power bridges after a smooth stop if the motor is running at minimum speed or halves the motor speed if the motor is running above minimum speed.

Even if the “Daisy Chain” mode is supported in the L6470 FW library, the L6470 Discovery board HW does not allow it. Please contact ST support if you want more information on “Daisy Chain” mode testing or implementation.

The FW L6470 library consists of the following files:

- user_motion/dspin/inc/stm32f10x_conf.h Library configuration file
- user_motion/dspin/src/stm32f10x_it.cInterrupt handlers
- user_motion/dspin/inc/stm32f10x_it.hHeader for stm32f10x_it.c
- user_motion/dspin/inc/clock.h System clock setup related header
- user_motion/dspin/src/clock.c System clock source file
- user_motion/dspin/inc/dspin.h L6470 definitions header
- user_motion/dspin/src/dspin.c L6470 routines source file
- user_motion/dspin/inc/dspin_config.hdSPIN configuration parameters (this file can be generated by the GUI)
- user_motion/dspin/inc/main.h Main header file
- user_motion/dspin/src/main.c Main program
- user_motion/dspin/inc/pre_include.hFirst header file included by the preprocessor
- user_motion/dspin/readme.txt Information on the files

4.6.3 Demonstration sequence description

The demonstration sequence description is for one L6470 device without daisy chaining. The application commands used are noted after the pipe character. The application commands traffic can be seen on the SPI interface, pins 18, 19, 20 and 23 which are respectively SDO, CK, SDI and CS.

Initialization and configuration

1. Initialize peripherals used by L6470
2. *LED_READY (green) is light up, other LEDs are switched off.*
3. LED check sequence:
   a) *LED_ERROR (red)* is switched ON
   b) *LED_BUSY (orange)* is switched ON
   c) *LED_SPARE (yellow)* is switched ON
   d) *LED_ERROR (red), LED_BUSY (orange)* and *LED_SPARE (yellow)* are switched OFF.
4. Resets and puts the L6470 into standby mode. The L6470 STBY_RESET pin goes low for a few hundreds of μs and then goes high.
5. Program all L6470 registers | **SetParam (PARAM, VALUE)**.

Flag signal management

6. Read status register content | **GetStatus**
7. Interrupt configuration for FLAG signal
8. Motor runs at constant speed of 400 steps/s forward direction | Run (DIR, SPD)
9. Tentative to write to the current motor absolute position register while the motor is running, *as a consequence the LED_ERROR (red) is lighted up* | **SetParam (PARAM, VALUE)**
10. Get Status to clear FLAG due to non-performable command, as a consequence the LED_ERROR (RED) *is switched off* | **GetStatus**
11. Motor stops smoothly | **SoftStop**
12. Wait until not busy - busy pin test (L6470 pin 22).

Busy signal management

13. Interrupt configuration for BUSY signal
14. Motor moves by 100,000 steps in reverse direction | **Move (DIR, N_STEP)**
15. *During busy time the LED_BUSY (orange) is switched ON*
16. The L6470 disables the power bridges after a deceleration to zero phase | **SoftHiZ**
17. *LED_BUSY (orange) is switched OFF.*

The LED_BUSY after the point 7 and the LED_ERROR after the point 13 are tied respectively to the BUSY/SYNC pin and the FLAG pin. So for example they are lighted up when motor accelerates or decelerates. This is happening in the remaining part of the demonstration although not mentioned.

Various application command examples

18. Motor moves by 60,000 steps forward | **Move (DIR, N_STEP)**
19. Wait until not busy - busy pin test (L6470 pin 22).
20. Send L6470 command setting hold duty cycle to 0.5%, send L6470 command changing run duty cycle to 5% | **SetParam (PARAM, VALUE)**
21. Motor runs at constant speed of 50 steps/s in reverse direction | **Run (DIR, SPD)**
22. Motor softly stops after a few seconds | **SoftStop**
23. RESET KVAL_HOLD to initial value, RESET KVAL_RUN to initial value | **SetParam (PARAM, VALUE)**
24. Wait until not busy - busy status check in Status register | **GetStatus**
25. Motor moves by 100,000 steps forward | **Move (DIR, N_STEP)**
26. Wait until not busy - busy status check in Status register | **GetStatus**
27. Test of the Flag pin (L6470 pin 24) by polling, wait in endless cycle if problem is detected.
28. Motor moves to its home position | **GoHome**
29. Wait until not busy - busy pin test (L6470 pin 22).
30. Motor goes to absolute position 65535 through the shortest path | **GoTo (ABS_POS)**
31. Wait until not busy - busy pin test (L6470 pin 22).
32. Motor goes in forward direction to absolute position 131071 | **GoTo_DIR (DIR, ABS_POS)**
33. Wait until not busy - busy pin test (L6470 pin 22).
34. Read run duty cycle (dSPIN_KVAL_RUN) parameter from L6470, read intersect speed (dSPIN_INT_SPD) parameter from L6470 | **GetParam(PARAM)**
35. Read Status register content | **GetStatus**
36. Read absolute position (dSPIN_ABS_POS) parameter from L6470 | **GetParam(PARAM)**
37. Reset position counter, actually | **ResetPos**
38. Read absolute position (dSPIN_ABS_POS) parameter from L6470 | **GetParam(PARAM)**
39. L6470 disable power stage (High Impedance) immediately, as a consequence the motor supply current drops | **HardHiZ.**

“Go Until” example

40. Interrupt configuration for SW MOTOR
41. Motor motion in forward direction at speed 400 steps/s until the user puts a jumper on J8, *the LED_SPARE (yellow) toggles until the user puts a jumper on J8* | **GoUntil (ACT, DIR, SPD)**
42. Motor stops
43. *The LED_SPARE (yellow) is switched off.*
44. Wait until not busy - busy pin test (L6470 pin 22)
45. Motor moves by 50,000 steps reverse | Move **(DIR, N_STEP)**
46. Motor moves to the position saved by the GoUntil command into the MARK register, so 50000 steps forward | **GoMark**
47. Wait until not busy - busy pin test (L6470 pin 22)
48. Nothing happens during a few seconds.

Release SW example

49. Motor motion in reverse direction at minimum speed until the user removes the jumper on J8, *the LED_SPARE (yellow) toggles until the user removes the jumper on J8* | **ReleaseSW (ACT, DIR)**
50. *The LED_SPARE (yellow) is switched off.*
51. Motor moves by 100,000 steps forward | **Move (DIR, N_STEP)**
52. Wait until not busy - busy pin test (L6470 pin 22)
53. Motor goes to home position set by the ReleaseSW command | **GoHome**
54. Wait until not busy - busy pin test (L6470 pin 22).

Step clock mode example

55. Get Status to clear FLAG due to switch turn-on event (falling edge on the SW pin) | **GetStatus**
56. Motor runs in step clock mode at 2000 steps/s in forward direction for a few seconds while a 2 kHz clock signal from the MCU is applied to the L6470 STCK pin (25) | **StepClock (DIR)**
57. The above cited clock is stopped.

Buttons interrupt example

58. Buttons interrupt configuration:
    a) Button_A
       - Starts the motor at quarter of max speed if it is stopped | **Run (DIR, SPD)**
       - Doubles the motor speed if it is already running | **Run (DIR, SPD)**
    b) Button_B
       - Disables the power bridges after a smooth stop if the motor is running at minimum speed | **SoftHiZ**
       - Halves the motor speed if the motor is running above minimum speed | **Run (DIR, SPD)**.

4.6.4 Create a project using the FW library package

Using your preferred IDE, create a new project.

In project options, properties or settings,

- Select for the device, the ST STM32F105xB.
- Use the CMSIS library.

Edit the preprocessor defined symbols and add:

- STM32F10X_CL
- USE_STDPERIPH_DRIVER
- ST_DSPIN_6470H_DISCOVERY

Edit the preprocessor include directories and add:

- $PROJ_DIR$\…\…\…\…\…\…\stm32f10x\CMSIS\CM3\DeviceSupport\ST\STM32F10x
- $PROJ_DIR$\…\…\…\…\…\…\stm32f10x\STM32F10x_StdPeriph_Driver\inc
- $PROJ_DIR$\…\…\…\…\…\dspin\inc

Where $PROJ_DIR$ is a variable containing the path to the project directory.

Add the required library source files:

- startup_stm32f10x_cl.s
- system_stm32f10x.c
- misc.c
- stm32f10x_exti.c
- stm32f10x_flash.c
- stm32f10x_gpio.c
- stm32f10x_rcc.c
- stm32f10x_spi.c
- stm32f10x_tim.c

For the debugger, for example, select the ST-LINK and configure it to run to main, to verify download, to use the Flash loader and to override the default *.board file with the FlashSTM32F105xB.board.

5 References

This user manual provides information on the hardware features and use of the EVAL6470H-DISC board along with the demonstration firmware and software. For additional information, refer to the following:

1. STM32F105xx/STM32F107xx datasheet (CD00220364)
2. STM32F101xx, STM32F102xx, STM32F103xx, STM32F105xx and STM32F107xx advanced ARM®-based 32-bit MCUs reference manual (RM0008)
3. Voltage mode control operation and parameter optimization application note (AN4144)
4. L6470 and L6472: fully integrated stepper motor drivers (AN4241)
5. L647x, L648x and powerSTEP01 family communication protocol (AN4290).

6 Revision history

Table 4. Document revision history

Date	Revision	Changes
29-Nov-2013	1	Initial release.
17-Apr-2015	2	Updated *Section : Introduction on page 1* (replaced “L6470H” by “L6470”). Removed Figure 5. Layout (silk screen) from page 9. Replaced “dSPIN” by “L6470” and/or “motor” in the whole document. Updated *Table 2 on page 9* and *Section 4.2.1 on page 13* (replaced “L6470H” by “L6470”). Updated *Table 3 on page 13* (minor modifications). Updated *Section 4.5.1 on page 24* (replaced “FWPSPIN” by “FWGUI”). Updated text above *Figure 24 on page 30* (removed L6472 device step). Updated *Section 4.5.2 on page 33* and *Section 4.6.2 on page 34* (removed L6472 device). Updated *20.* in *Section : Various application command examples on page 37* (minor modifications). Updated *1.*, *2.*, *4.*, and *5.* in *Section 5 on page 40*. Minor modifications throughout document.

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.

© 2015 STMicroelectronics – All rights reserved