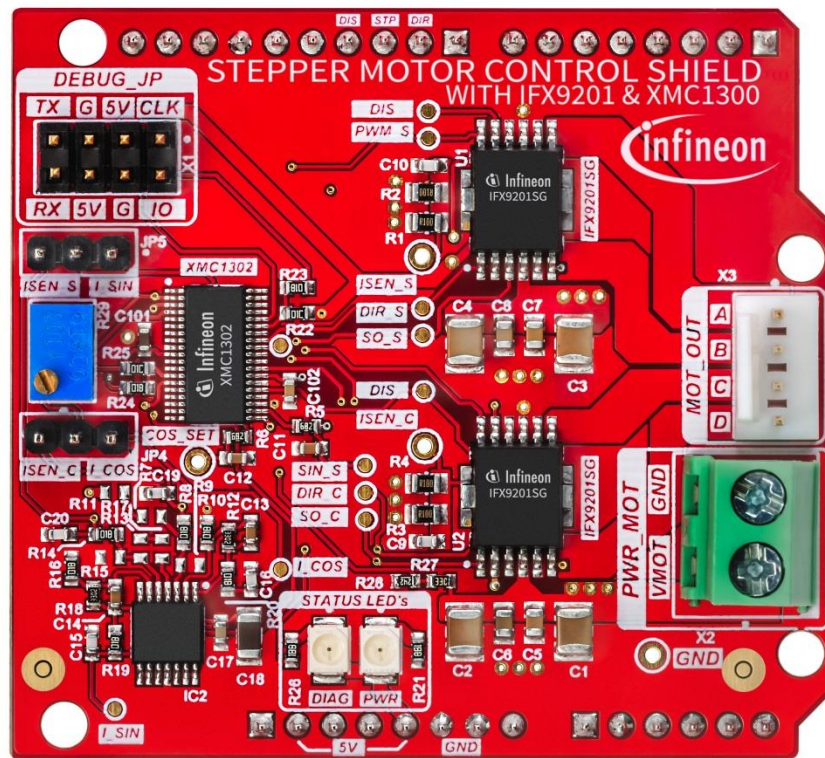


# Stepper Motor Control Shield

with IFX9201 and XMC1300



## About this document

### Scope and purpose

This board user manual provides a basic introduction to the Stepper Motor Control Shield.

The Stepper Motor Control Shield is a complete system solution for driving stepper motors with a continuous current capability of 2-3 A per coil (sine and cosine). It contains a 32 bit microcontroller with control logic for stepper motors and two integrated H-Bridges capable of driving up to 6 A peak, the IFX9201SG.

It is intended to use as a shield with an XMC 1100 Boot Kit or XMC4700 Relax Kit for 5V shields but can also be controlled by other methods.

### Intended audience

This board user manual is intended for anyone using the Stepper Motor Control Shield.

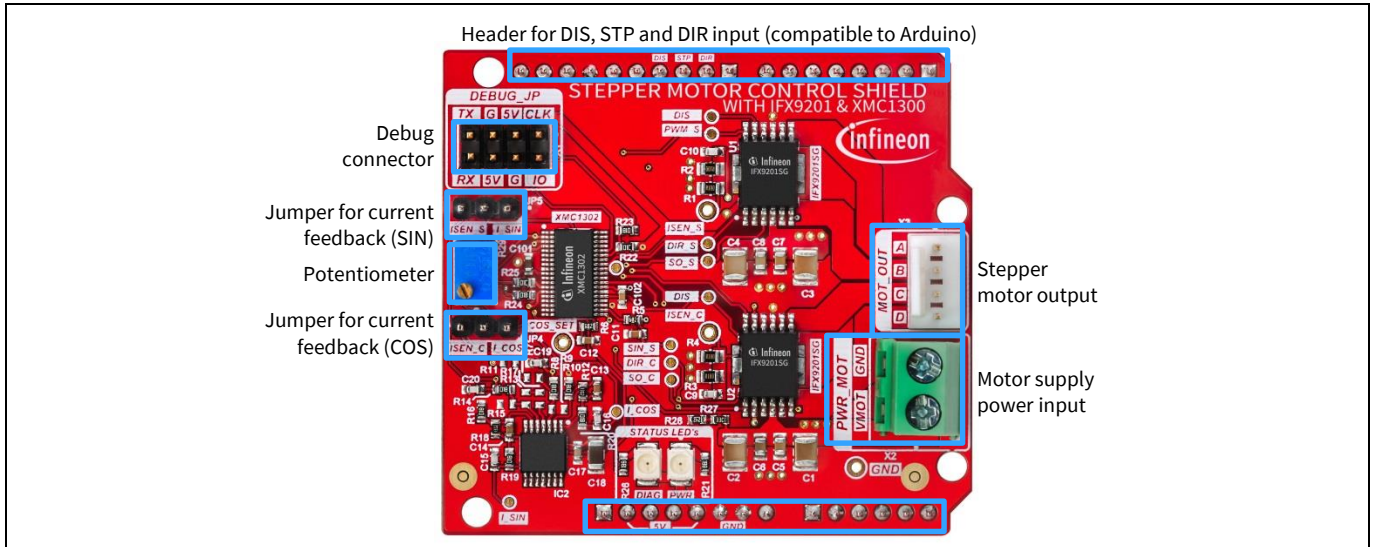
## Table of Contents

<b>About this document.....</b>	<b>1</b>
<b>Table of Contents .....</b>	<b>2</b>
<b>1 Overview .....</b>	<b>3</b>
<b>2 Introduction to stepper motor control .....</b>	<b>4</b>
2.1 Full step control.....	4
2.2 Half step control.....	5
2.3 Micro step control .....	5
<b>3 Getting Started.....</b>	<b>6</b>
3.1 Power Supply.....	6
3.2 Selecting a stepper motor.....	6
3.3 Preconditions .....	6
3.4 Connecting the shield .....	7
<b>4 Operation .....</b>	<b>9</b>
4.1 Control via XMC4700 Relax Kit for 5V shields .....	9
4.1.1 Preconditions .....	9
4.1.1.1 Hardware.....	9
4.1.1.2 PC setup.....	9
4.1.2 Test of the board .....	9
4.1.3 Configuration of the board .....	10
4.2 Control by other means .....	11
<b>5 Hardware Description .....</b>	<b>12</b>
5.1 Schematics .....	12
5.2 Layout .....	17
5.3 Order information .....	18

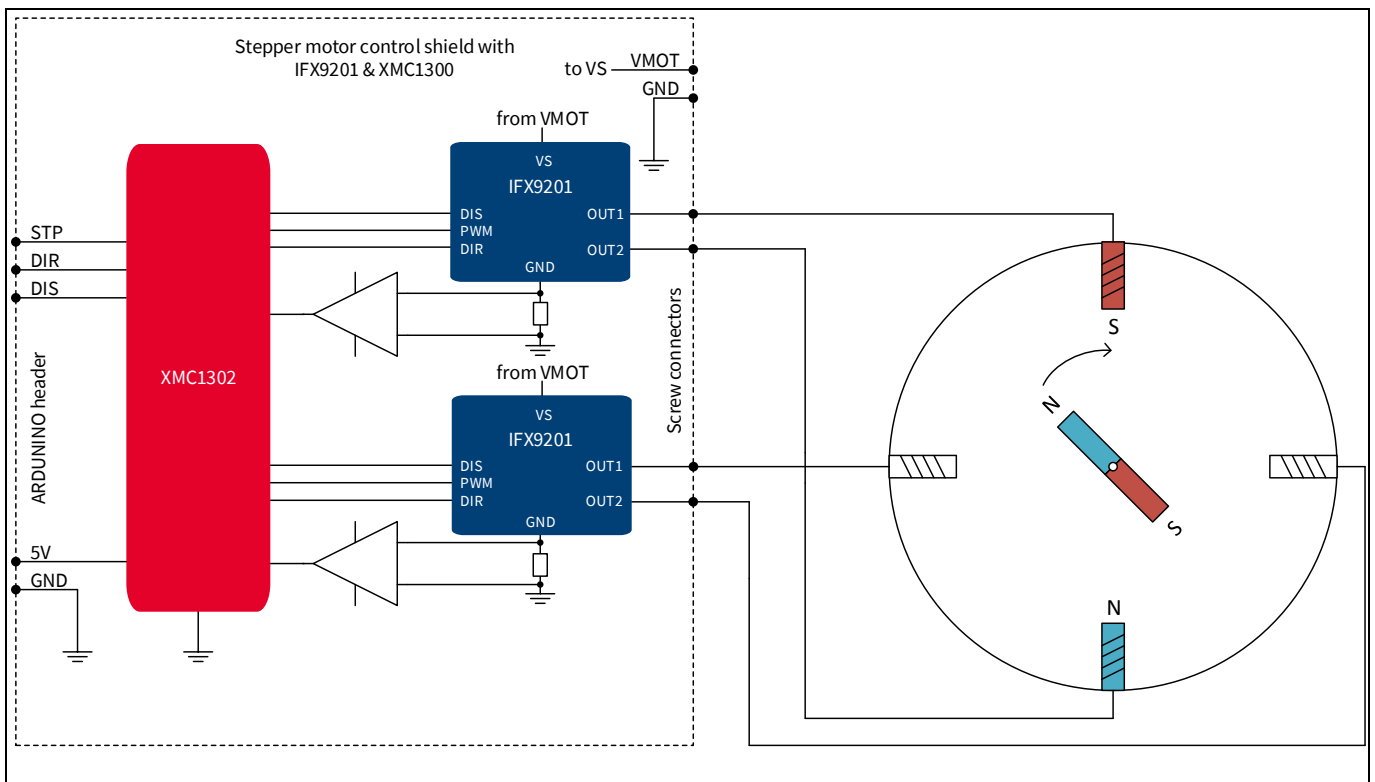
## Overview

### 1 Overview

The key components Stepper Motor Control Shield are shown in Figure 1.



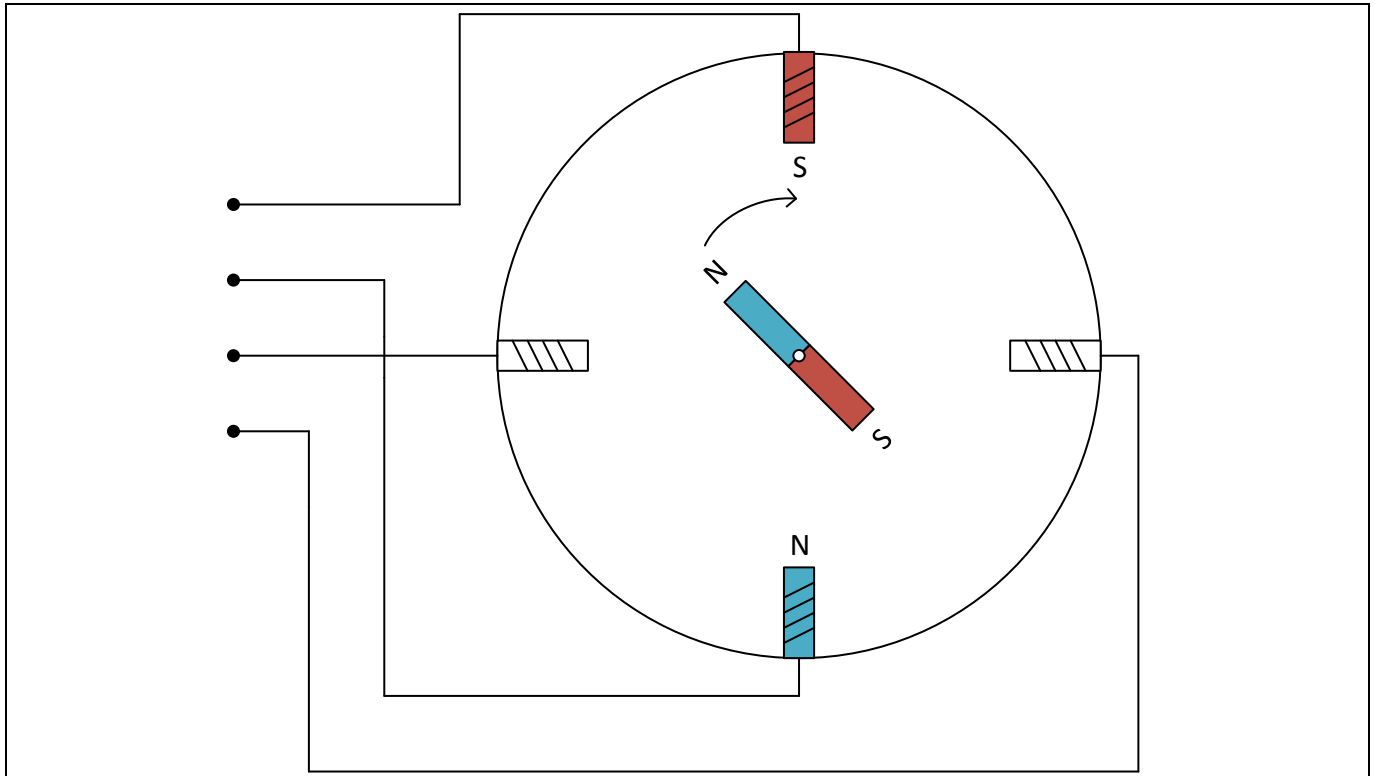
**Figure 1** Top view Stepper Motor Control Shield



**Figure 2** Block diagram Stepper Motor Control Shield

## 2 Introduction to stepper motor control

A bipolar stepper motor consists of two (four / six / ...) coils mounted inside the stator and a rotor which is a permanent magnet with toothed poles. The position of the motor is controlled by the magnetic field caused by the stator coils. The coils are mounted in a 90° distance, so they are called sine and cosine coil. See Figure 3.



**Figure 3 Simplified stepper motor**

The advantage is simple control; there is no feedback loop needed as the change of position can be calculated based on the steps done (the initial position must be known beforehand to determine the final position). In addition the holding torque is available if the motor is standing still.

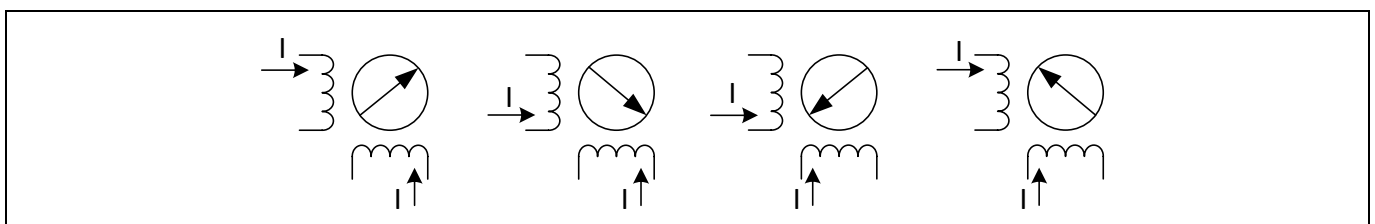
Typical applications are 3D printing, robotics and home automation.

There are different methods of control for stepper motors. These will be described in the following sections.

### 2.1 Full step control

Full step control offers a maximum holding torque and simple control. Typical stepper motors allow e.g. 200 steps per rotation which gives a sufficient resolution suitable for many applications already.

The example in Figure 4 shows the principle of full step control.

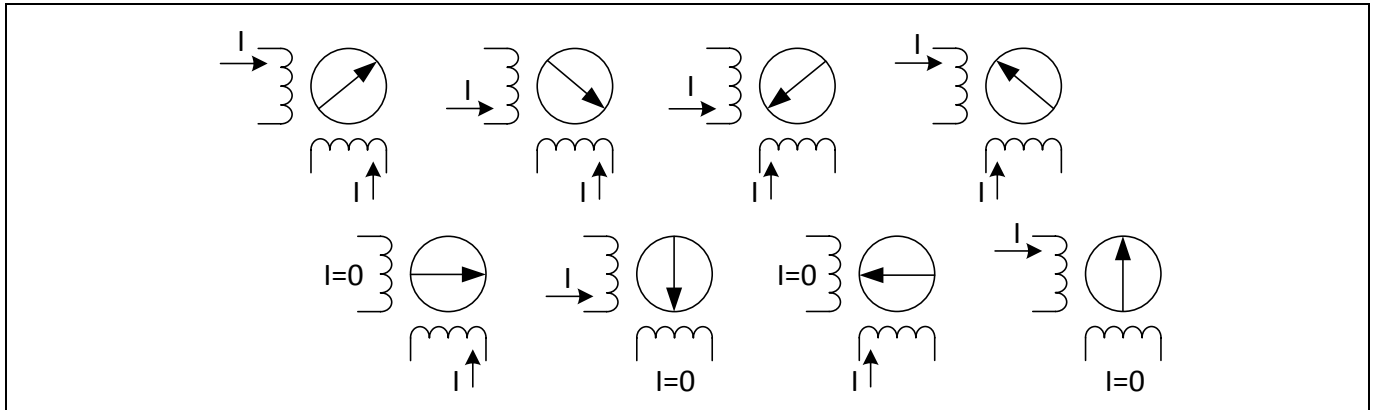


**Figure 4 Full step control**

**Introduction to stepper motor control**

**2.2 Half step control**

Full step control always uses current through all coils. To increase the number of steps by a simple method, there is the possibility to only supply one of the two coils (“wave drive”). The combination of full step and wave drive is called half step control. See Figure 5.

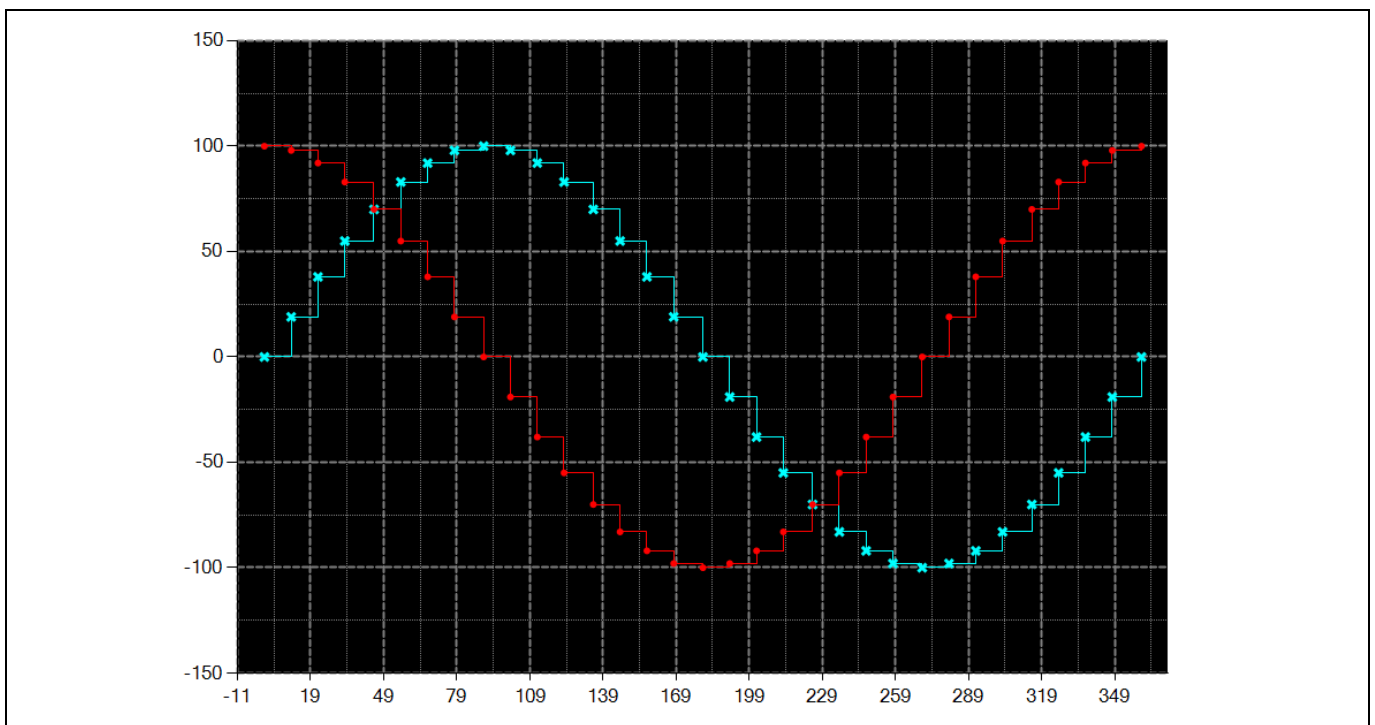


**Figure 5 Half step control**

The disadvantage is the holding torque being lower than in full step mode.

**2.3 Micro step control**

For better precision, there is the possibility to use micro steps by controlling the current through the coils in a more precise way. The Stepper Motor Control Shield supports micro step control as well. The number of steps can be configured starting with 8. The maximum is typically limited by the capability of the motor, the Stepper Motor Control Shield allows up to 128 microsteps. An example of the current profiles for one “full” step is shown in Figure 6.



**Figure 6 Micro step control, sin: blue, cos: red, example for 32 steps, percent of current over one “full”step(360°)**

Getting Started

### 3 Getting Started

#### 3.1 Power Supply

For providing the power to drive a DC motor the Stepper Motor Control Shield needs an external power supply connected to VMOT and GND and a 5 V supply for the micro processor. VMOT is typically set to 12 or 24 V.

#### 3.2 Selecting a stepper motor

The IFX9201 can drive loads with peak currents of up to 6 A. The achievable continuous drive current is lower and depends on supply voltage, switching frequency and the cooling conditions. Realistic continuous drive currents for this kit are in the range of 2 A per coil. Many stepper motors for applications such as 3D printers, toys and robotics fall in this range.

The nominal voltage of the stepper motor is derived from the rated current and the coil resistance. The nominal voltage of the stepper motor should be lower than the supply voltage.

**Attention:** *Stepper motors may not be connected directly to the supply voltage. The Stepper Motor Control Shield will control the current and reduce the voltage at the output by PWM.*

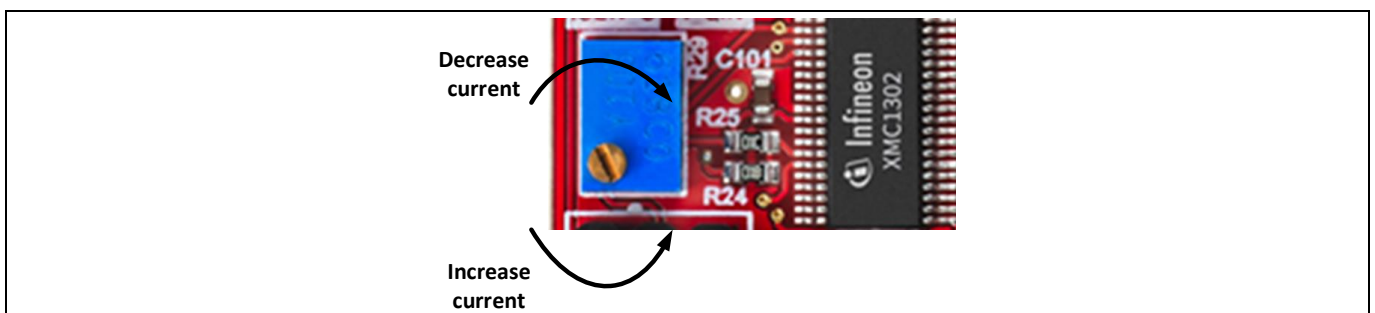
The shield can only be used for bipolar stepper motors (i.e. with a 4-wire connection).

#### 3.3 Preconditions

Before starting, make sure the current is limited for the motor. In the default firmware (as delivered), the current is limited to 1 A per coil.

To further decrease the maximum current, the potentiometer can be used, see Figure 7.

**Attention:** *For getting started, it is recommended to limit the current to a very low value and later increase the current. This can be done by turning the potentiometer clockwise as far as possible (minimum current). Any counterclockwise rotation will then increase the current up to the configured maximum.*



**Figure 7** Potentiometer setting

There are two jumpers on the board, JP4 and JP5. Per default, these shall be connected between the central pin and I\_SIN/I\_COS, see Figure 8. This means the operational amplifier is being used for measuring the current through the IFX9201 devices. It is recommended to keep this setting.

**Getting Started**



**Figure 8** Jumper setting: Connect to I\_SIN and I\_COS

### **3.4 Connecting the shield**

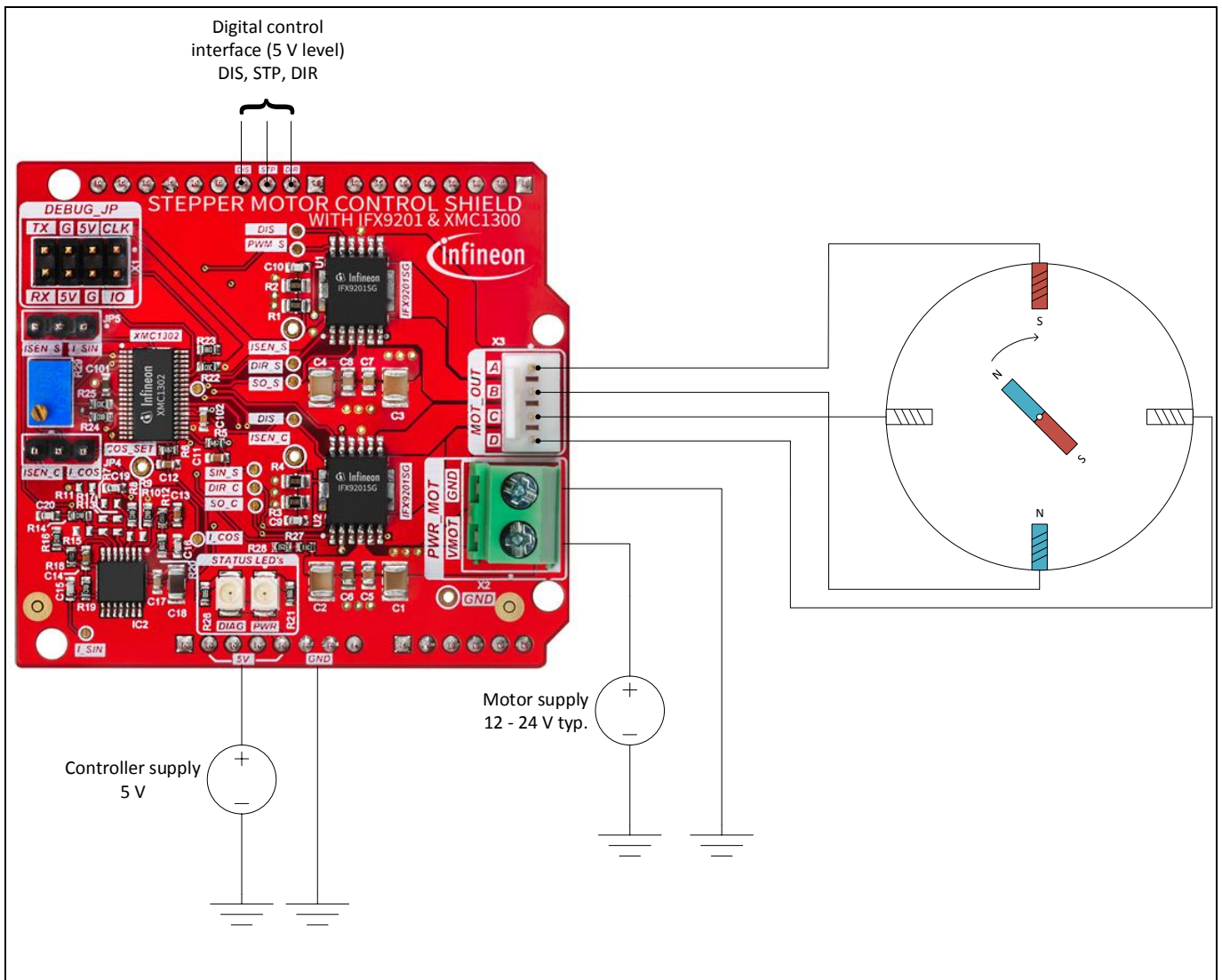
The Stepper Motor Control Shield can be used standalone or in combination with a microcontroller board with an ARDUINO™ header. An XMC4700 Relax Kit for 5V Shields is recommended and will be used in this description. However, there are also other options.

The main connections are shown in Figure 9.

# Stepper Motor Control Shield with IFX9201 and XMC1300



## Getting Started



**Figure 9** Connecting Motor, Supply and control signals

XMC4700 Relax Kit for 5V Shields will provide the 5 V supply and digital control interface via the ARDUINO™ header. The signals are described in Table 1.

**Table 1** Digital control interface

Pin	Signal	Options
DIS	Enable input	Low = disabled High = enabled
STP	Step input	Rising edge will generate one step of the motor
DIR	Direction input	Low = counterclockwise High = clockwise



**Operation**

## 4 Operation

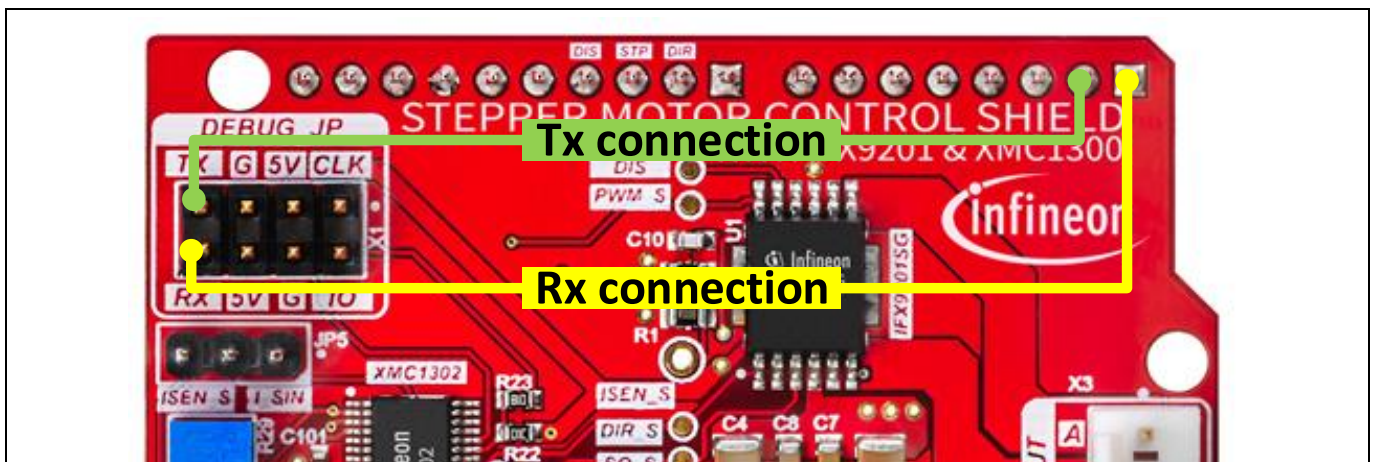
As mentioned above, there is software available for download at [www.infineon.com/ifx9201sg-stepper-motor-shield](http://www.infineon.com/ifx9201sg-stepper-motor-shield). It will control the Stepper Motor Control Shield by an XMC4700 Relax Kit for 5V shields. In addition, the firmware on the XMC1302 used for stepper control can be downloaded and modified.

### 4.1 Control via XMC4700 Relax Kit for 5V shields

#### 4.1.1 Preconditions

##### 4.1.1.1 Hardware

The Stepper Motor Control Shield has been intended as prototype and needs some hardware modification if it shall be configured by the XMC4700 Relax Kit. It needs a connection between the Rx and Tx pins of the debug interface to the Rx/Tx pins of the ARDUINO™ header. This is illustrated in Figure 10. Please make these connections by a wire.



**Figure 10** Modification for communication to XMC4700 Relax Kit

##### 4.1.1.2 PC setup

Make sure the J-Link driver is installed properly. This is for example included in the installation of DAVE™ (see <http://www.infineon.com/dave>). For more details on J-Link please visit [www.segger.com](http://www.segger.com).

For more information on the installation and USB connection, please refer to the board manual of the XMC4700 Relax Kit for 5V shields available under [www.infineon.com/cms/de/product/evaluation-boards/kit\\_xmc47\\_relax\\_5v\\_ad\\_v1](http://www.infineon.com/cms/de/product/evaluation-boards/kit_xmc47_relax_5v_ad_v1).

##### 4.1.2 Test of the board

The software uses the ARDUINO IDE. It allows doing a basic test so that the stepper motor will be turning. It controls the pins DIS, STP and DIR according to Table 1. Just configure the number per revolutions for the motor and make sure before running the test code that the current is limited by the potentiometer to a value lower than the motor's maximum current as described in section 3.3.

In addition, there is a print to a Terminal interface. Select the virtual COM port of the XMC4700 Relax Kit and a baud rate of 9600 (optional, the motor will rotate without the Terminal connection).

**Operation**

**4.1.3 Configuration of the board**

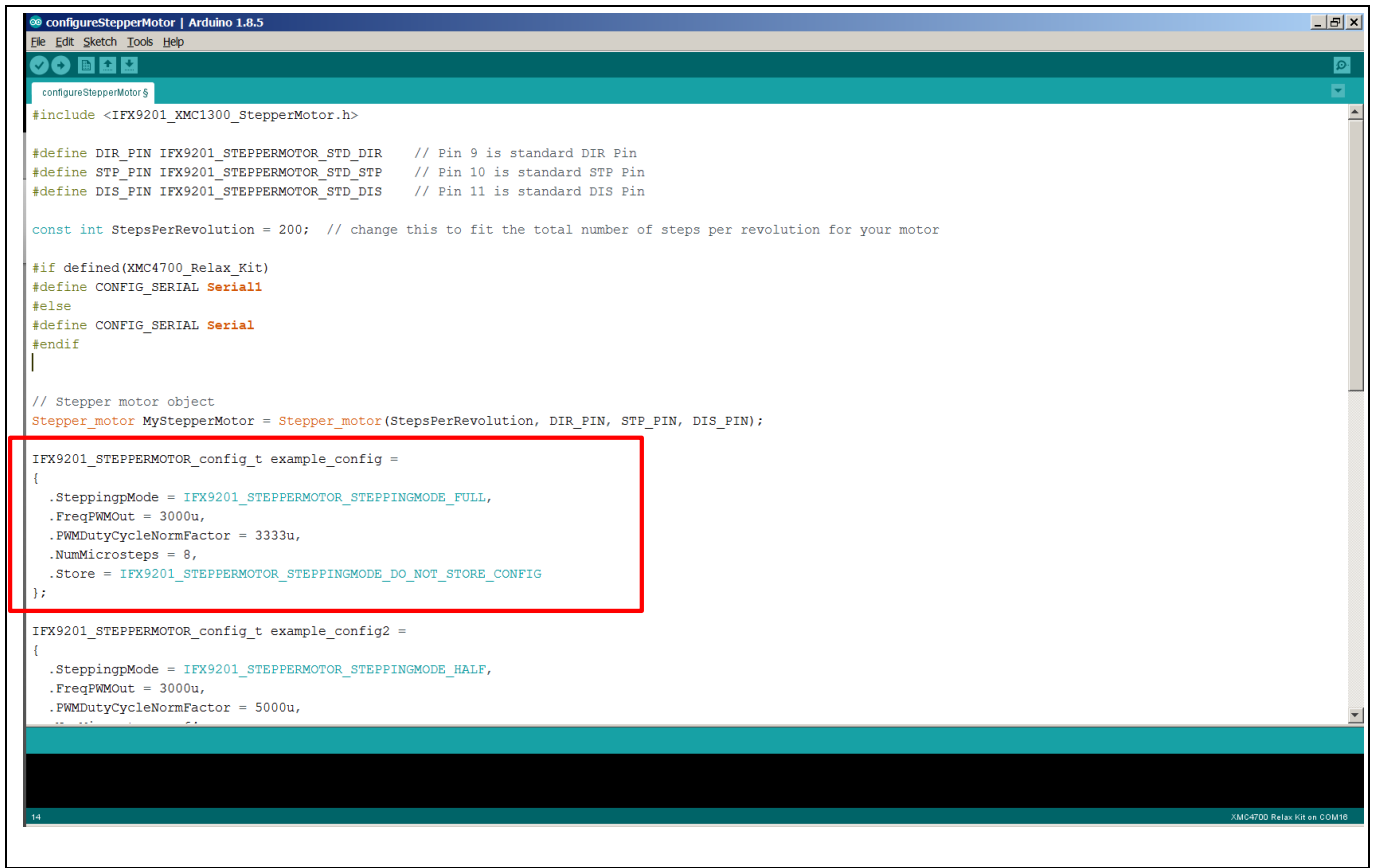
The software uses the ARDUINO IDE. It allows to configure the following parameters according to Table 2. The delivery condition is listed in the “Default” column.

**Table 2 Configuration of the Stepper Motor Control Shield**

Parameter	Options	Default
SteppingpMode	IFX9201_STEPPERMOTOR_STEPPINGMODE_FULL, IFX9201_STEPPERMOTOR_STEPPINGMODE_HALF, IFX9201_STEPPERMOTOR_STEPPINGMODE_MICROSTEP	Full step
FreqPWMOut	PWM output frequency. For higher currents stay at some kHz, for low current up to 20 kHz is possible.  <i>Note: The effective frequency may be different in order to limit the current.</i>	3000 [Hz]
PWMDutyCycleNormFactor	Maximum current (potentiometer at 100%). 10000 relates to 3 A current.	3333 (1 A)
NumMicrosteps	8, 12, 16, 20, 24,... 128, only valid if SteppingpMode = IFX9201_STEPPERMOTOR_STEPPINGMODE_MICROSTEP	n/a
Store	IFX9201_STEPPERMOTOR_STEPPINGMODE_DO_NOT_STORE_CONFIG; IFX9201_STEPPERMOTOR_STEPPINGMODE_STORE_CONFIG. If set, this will update the XMC1300 on the Stepper Motor Control Shield so that after a power cycle the values are kept.  <i>Note: The memory of the XMC1300 has a limited amount of writing cycles. Therefore, it is recommended to find a working setup and test it before finally storing the configuration.</i>	n/a

The new configuration can be entered into the IFX9201\_STEPPERMOTOR\_config\_t example\_config, see Figure 11.

## Operation



```
configureStepperMotor | Arduino 1.8.5
File Edit Sketch Tools Help

configureStepperMotor $
#include <IFX9201_XMC1300_StepperMotor.h>

#define DIR_PIN IFX9201_STEPPERMOTOR_STD_DIR // Pin 9 is standard DIR Pin
#define STP_PIN IFX9201_STEPPERMOTOR_STD_STP // Pin 10 is standard STP Pin
#define DIS_PIN IFX9201_STEPPERMOTOR_STD_DIS // Pin 11 is standard DIS Pin

const int StepsPerRevolution = 200; // change this to fit the total number of steps per revolution for your motor

#if defined(XMC4700_Relax_Kit)
#define CONFIG_SERIAL Serial1
#else
#define CONFIG_SERIAL Serial
#endif

// Stepper motor object
Stepper_motor MyStepperMotor = Stepper_motor(StepsPerRevolution, DIR_PIN, STP_PIN, DIS_PIN);

IFX9201_STEPPERMOTOR_config_t example_config =
{
    .SteppingMode = IFX9201_STEPPERMOTOR_STEPPINGMODE_FULL,
    .FreqPWMOut = 3000u,
    .PWMDutyCycleNormFactor = 3333u,
    .NumMicrosteps = 8,
    .Store = IFX9201_STEPPERMOTOR_STEPPINGMODE_DO_NOT_STORE_CONFIG
};

IFX9201_STEPPERMOTOR_config_t example_config2 =
{
    .SteppingMode = IFX9201_STEPPERMOTOR_STEPPINGMODE_HALF,
    .FreqPWMOut = 3000u,
    .PWMDutyCycleNormFactor = 5000u,
    .NumMicrosteps = 8,
    .Store = IFX9201_STEPPERMOTOR_STEPPINGMODE_DO_NOT_STORE_CONFIG
};
```

Figure 11 Configuration parameters

## 4.2 Control by other means

If the default configuration is sufficient (full step, 1 A max. current), only the signals DIS, STP and DIR need to be applied.

If the configuration options provided are not sufficient, there is as well a possibility to modify the code on the XMC1302. Therefore, a debugger, such as the XMC™ Link, is needed. It can be connected to the debug connector. The firmware is available as beta version under [www.infineon.com/ifx9201sg-stepper-motor-shield](http://www.infineon.com/ifx9201sg-stepper-motor-shield).

Hardware Description

## 5 Hardware Description

### 5.1 Schematics

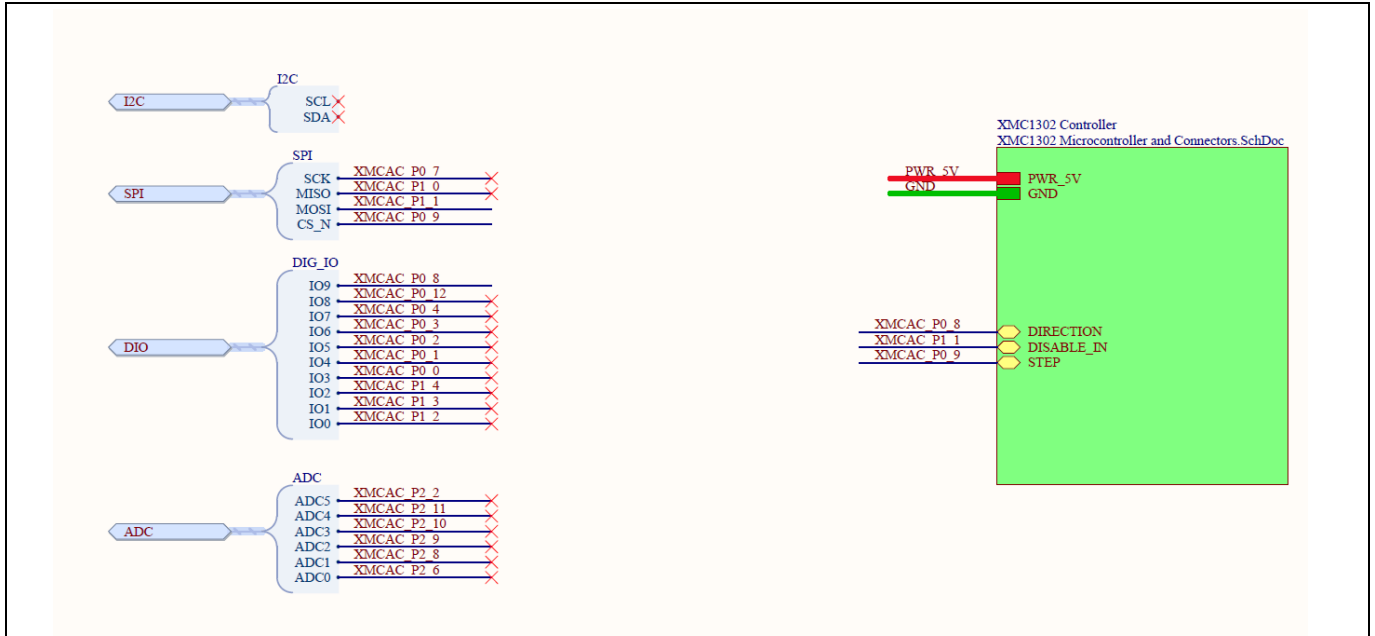


Figure 12 XMC1302 connection overview

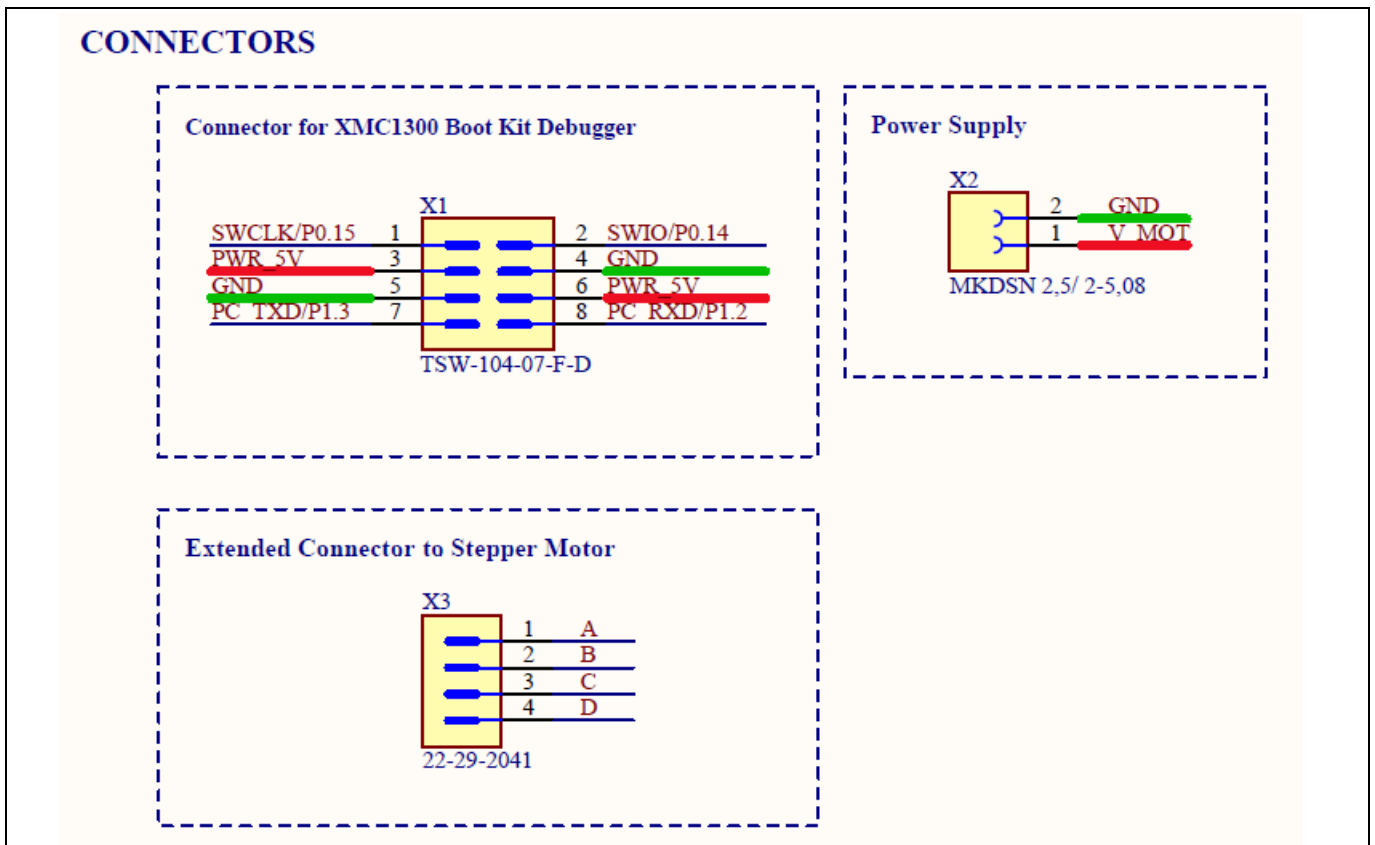


Figure 13 Connectors

## Hardware Description

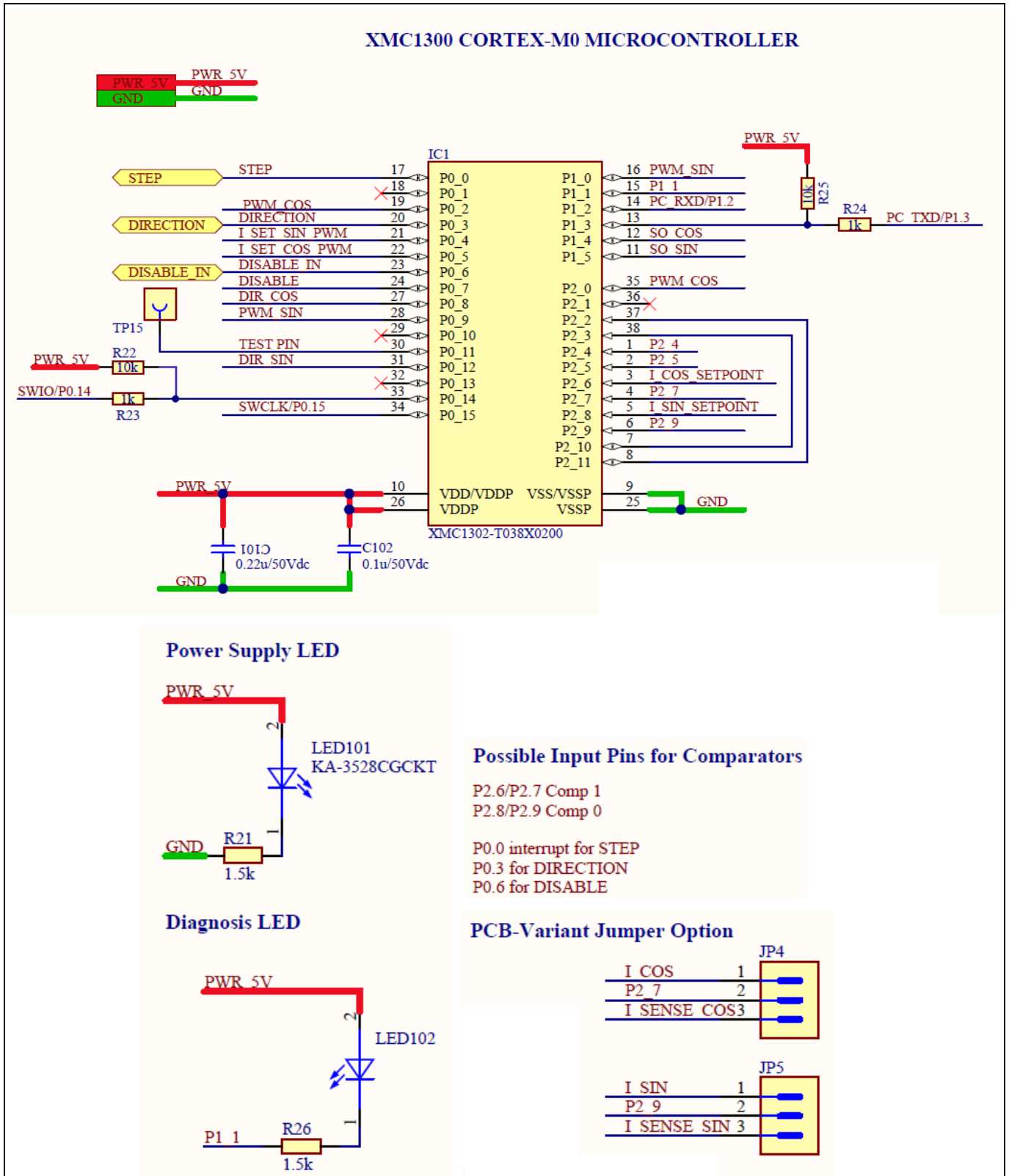
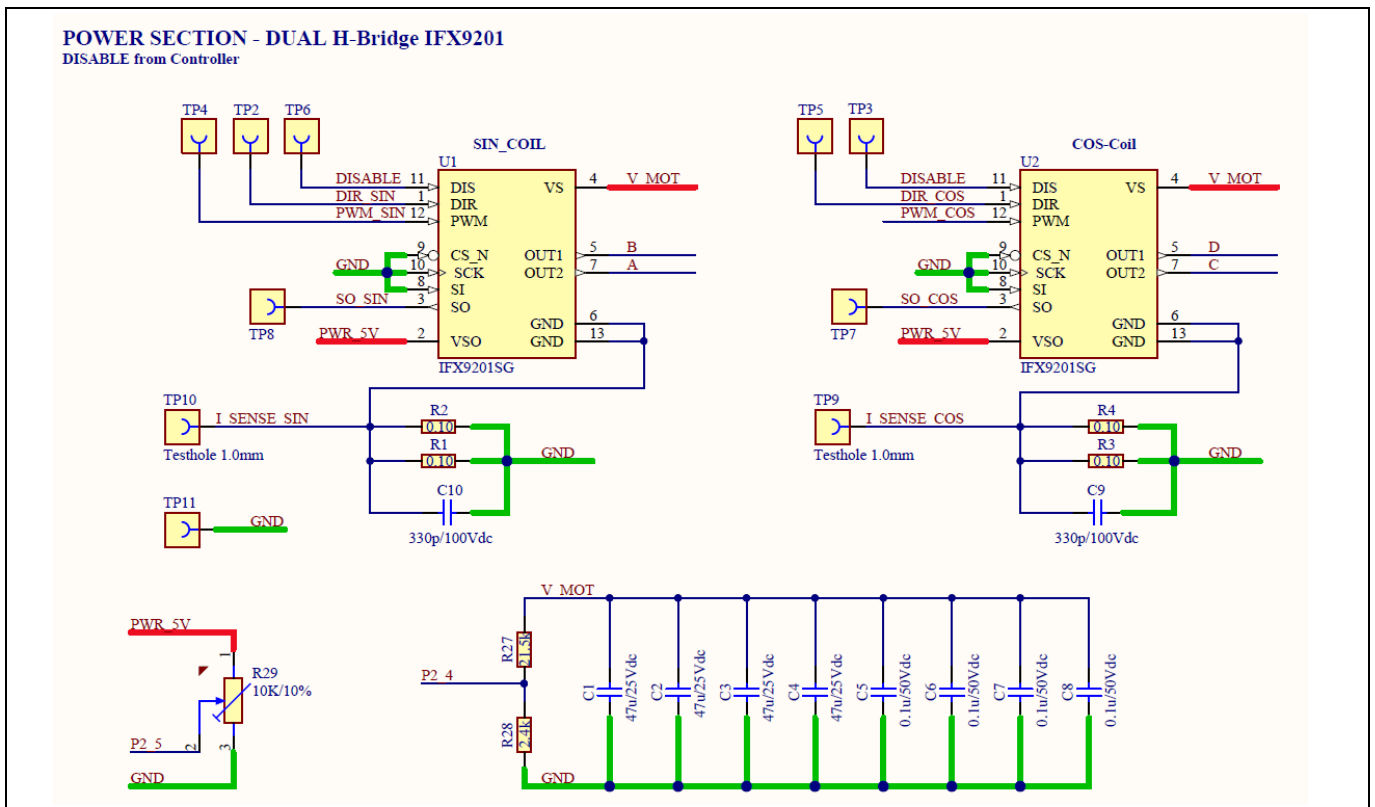


Figure 14 Microcontroller

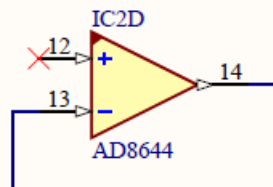
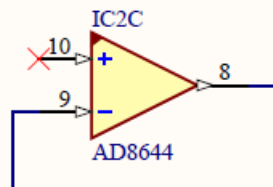
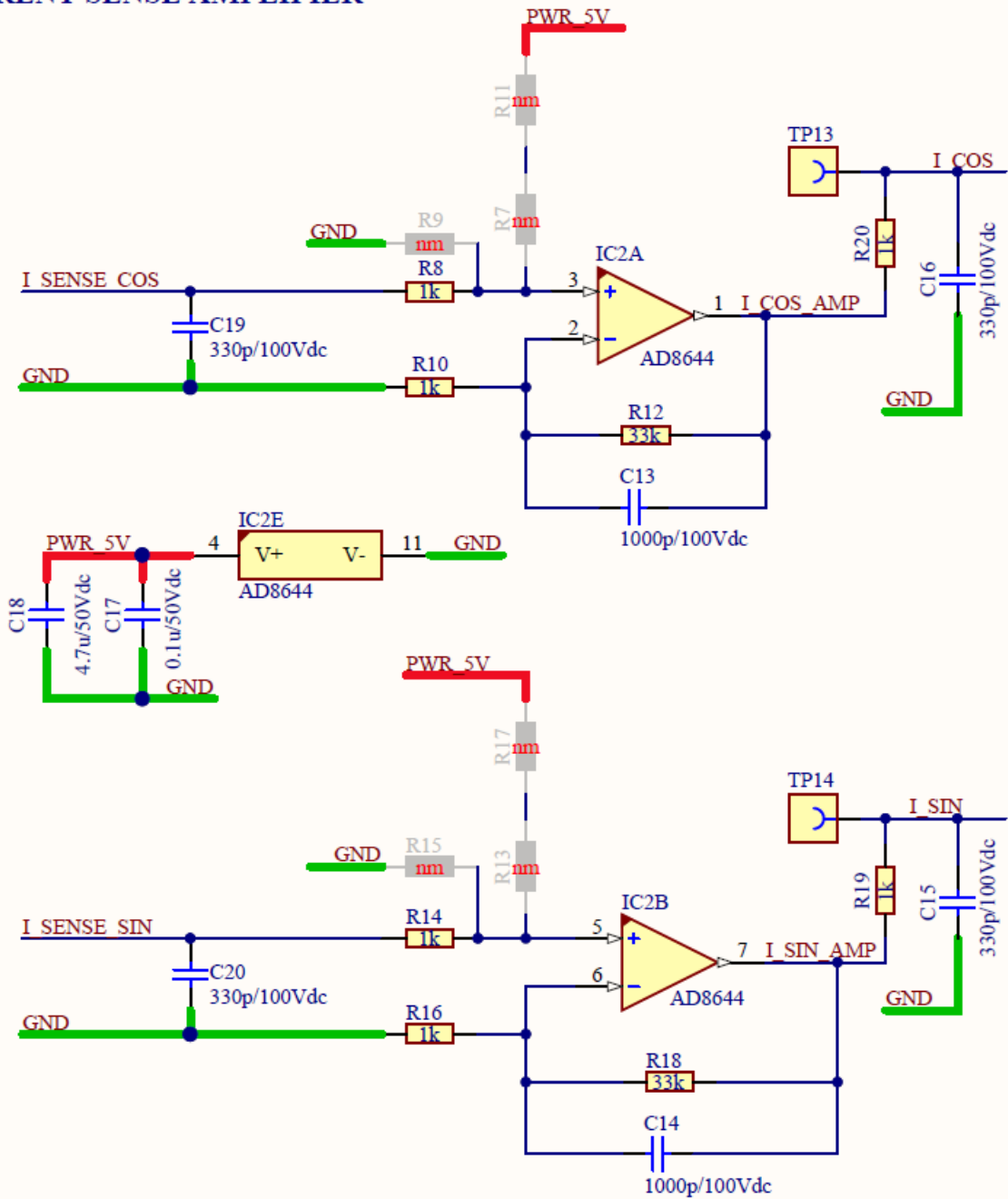
## Hardware Description



**Figure 15 Power part**

Hardware Description

CURRENT SENSE AMPLIFIER



Hardware Description

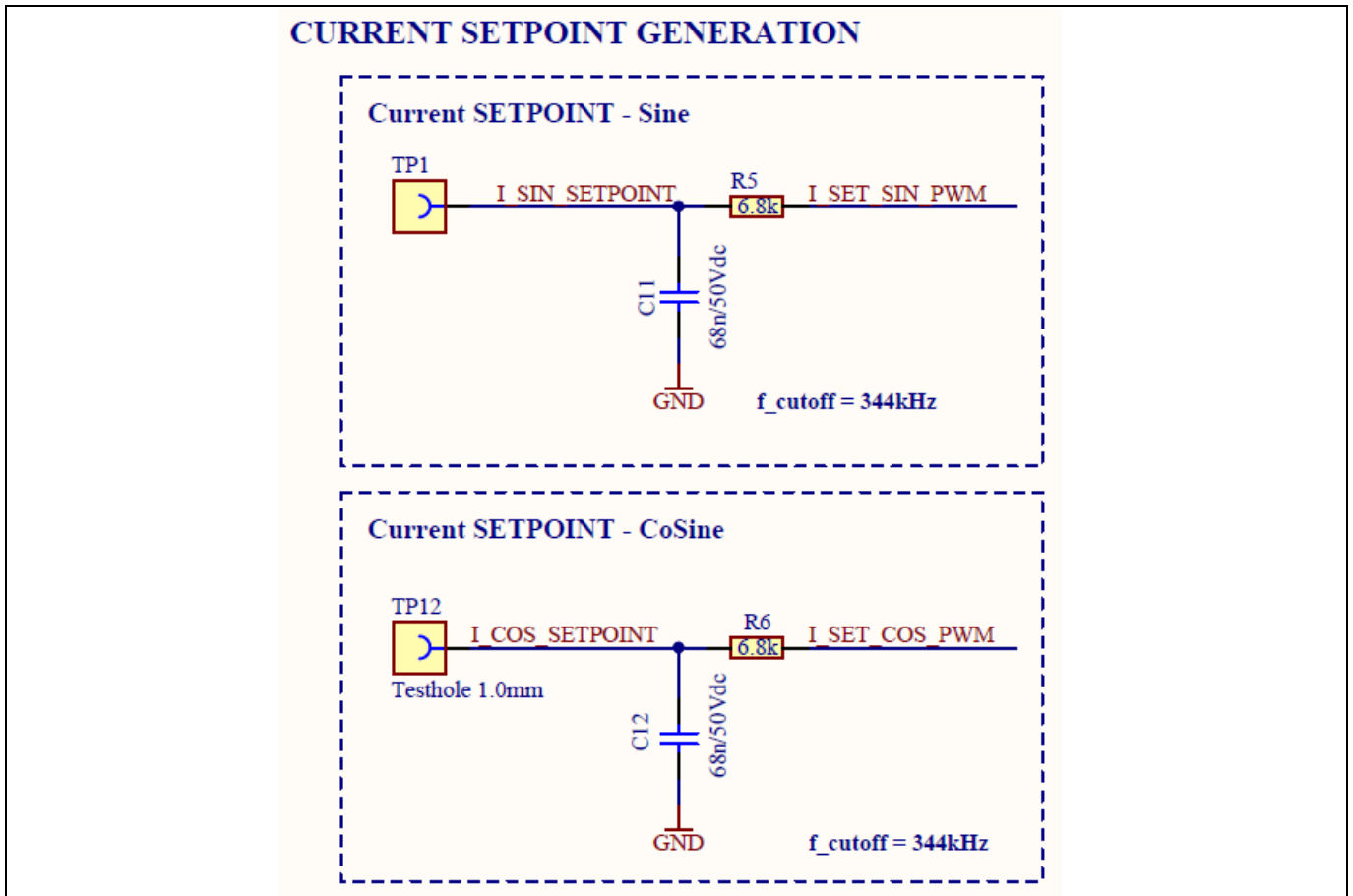


Figure 16 Current sensing circuitry



## 5.2 Layout

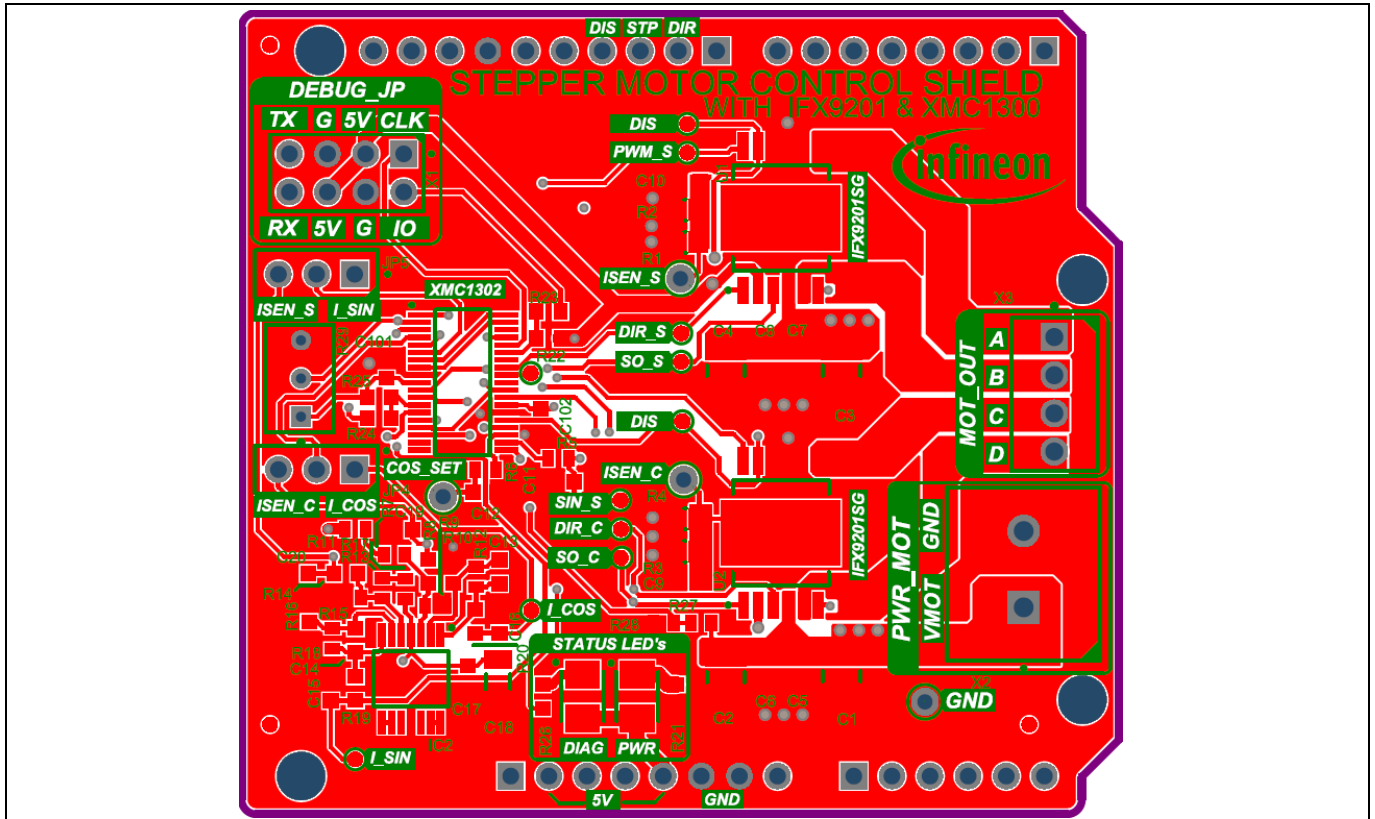


Figure 17 Layout top layer

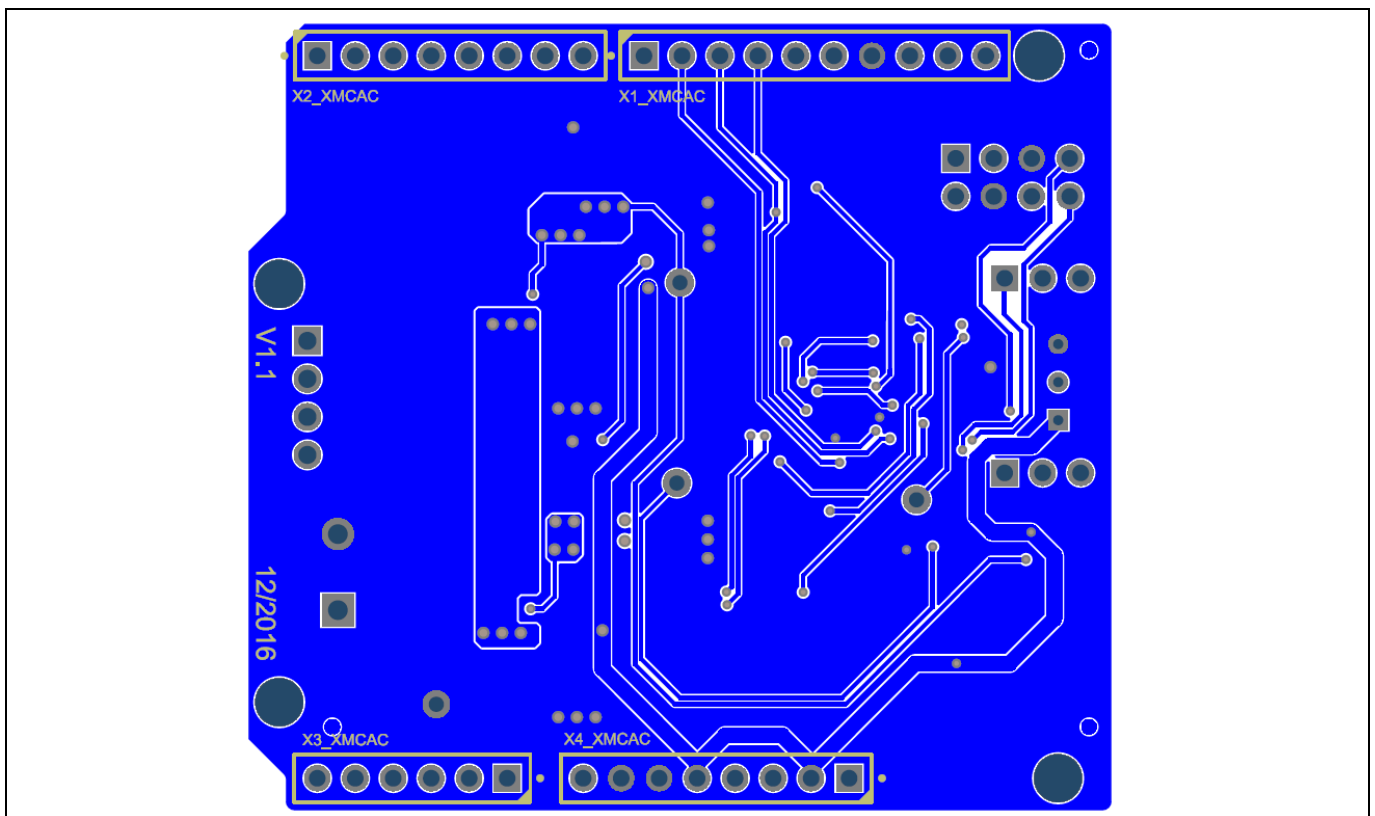


Figure 18 Layout bottom layer

## **5.3 Order information**

OPN: KITXMC1300IFX9201TOBO1