Mosquito BLDC Controller User Guide

Version 1.0

Revision 1.0.0.0

April 22, 2022

Revision History

Revision ID	Date	Owner	Details
1.0.0.0	04/22/2022	VALENTINE	Initial draft

Table of Contents

1 Introduction to Mosquito® User Guide

1.1 Purpose

This guide provides step-by-step instructions on how to install and configure the Mosquito BLDC Board.

1.2 Product Version and Revision

Mosquito® is based on a modular architecture, comprising of three basic modules that can be used in conjunction to meet the customer's specific requirements. This guide provides instructions for Mosquito®, Version 1.0, Revision 1.0.0.0.

1.3 Intended Users

This guide is primarily intended for:

- Design and development electrical engineers – responsible for developing very low power BLDC motor systems.
- DYI hobbyists – responsible for experimenting with BLDS motors and interfacing with other software applications.

2 Mosquito® Installation Instructions

2.1 Overview of Hardware Components

There are three different systems integrated on the board:

1. Integrated Motor Driver DRV8313
2. STM32G031 microcontroller
3. Step-down 3.3v buck converter

2.2 Integrated Motor Driver DRV8313

This section provides detailed specifications for the motor driver.

2.2.1 Assumptions and prerequisites:

The following knowledge is assumed:

1. Basic understanding of PMSM motor design
2. Basic understanding of power electronics and specifically MOSFETs
3. Basic understanding of PMSM motor control

2.2.2 DRV8313

DRV8313 is a 2.5-A Triple 1/2-H Bridge Driver. The DRV8313 provides three individually controllable half-H-bridge drivers. The device is intended to drive a three-phase brushless-DC motor, although it can also be used to drive solenoids or other loads. Each output driver channel consists of N-channel power MOSFETs configured in a 1/2-H-bridge configuration. Each 1/2-H-bridge driver has a dedicated ground terminal, which allows independent external current sensing. An uncommitted comparator is integrated into the DRV8313, which allows for the construction of current-limit circuitry or other functions. Internal protection functions are provided for undervoltage, charge pump faults, overcurrent, short circuits, and overtemperature. Fault conditions are indicated by the nFAULT pin.

2.2.3 Features

For an in-depth description of the driver, please refer to the official Texas Instruments documentation at:

https://www.ti.com/lit/ds/symlink/drv8313.pdf

2.2.3.1 Triple 1/2-H Bridge Driver IC

– 3-Phase brushless DC Motors
– Solenoid and Brushed DC Motors

• High Current-Drive Capability: 2.5-A Peak
• Low MOSFET ON-Resistance
• Independent 1/2-H-Bridge Control
• Uncommitted Comparator Can Be Used for

2.2.3.2 **Current Limit or Other Functions**

- Built-In 3.3-V 10-mA LDO Regulator
- 8-V to 60-V Operating Supply-Voltage Range
- Sleep Mode for Standby Operation
- Small Package and Footprint
  - 28-Pin HTSSOP (PowerPAD™ Package)
  - 36-Pin VQFN

2.2.3.3 **Applications**

- Camera Gimbals
- HVAC Motors
- Office Automation Machines
- Factory Automation and Robotics

2.3 MCU STM32G031

This section provides detailed specifications for the microcontroller.

2.3.1 Assumptions and prerequisites:

The following knowledge is assumed:

1. Basic understanding of microcontroller design
2. Basic understanding of microcontroller programming

2.3.2 DRV8313

The STM32G031x4/x6/x8 mainstream microcontrollers are based on high-performance Arm® Cortex®-M0+ 32-bit RISC core operating at up to 64 MHz frequency. Offering a high level of integration, they are suitable for a wide range of applications in consumer, industrial and appliance domains and ready for the Internet of Things (IoT) solutions. The devices incorporate a memory protection unit (MPU), high-speed embedded memories (8 Kbytes of SRAM and up to 64 Kbytes of Flash program memory with read protection, write protection, proprietary code protection, and securable area), DMA, an extensive range of system functions, enhanced I/Os, and peripherals. The devices offer standard communication interfaces (two I2Cs, two SPIs / one I2S, and two USARTs), one 12-bit ADC (2.5 MSps) with up to 19 channels, an internal voltage reference buffer, a low-power RTC, an advanced control PWM timer running at up to double the CPU frequency, four general purpose 16-bit timers, a 32-bit general-purpose timer, two low-power 16-bit timers, two watchdog timers, and a SysTick timer. The devices operate within ambient temperatures from -40 to 125°C and with supply voltages from 1.7 V to 3.6 V. Optimized dynamic consumption combined with a comprehensive set of power-saving modes, low-power timers and low-power UART, allows the design of low-power applications. VBAT direct battery input allows keeping RTC and backup registers powered. The devices come in packages with 8 to 48 pins.

2.3.3 Features

For an in-depth description of the microcontroller, please refer to the official ST Micro documentation at:

https://www.st.com/resource/en/datasheet/stm32g031j6.pdf

• Core: Arm® 32-bit Cortex®-M0+ CPU,frequency up to 64 MHz
• -40°C to 85°C/105°C/125°C operating temperature
• Memories
– Up to 64 Kbytes of Flash memory with protection and securable area
– 8 Kbytes of SRAM with HW parity check
• CRC calculation unit
• Reset and power management
– Voltage range: 1.7 V to 3.6 V
– Power-on/Power-down reset (POR/PDR)
– Programmable Brownout reset (BOR)
– Programmable voltage detector (PVD)
– Low-power modes:
Sleep, Stop, Standby, Shutdown
– VBAT supply for RTC and backup registers
• Clock management
– 4 to 48 MHz crystal oscillator
– 32 kHz crystal oscillator with calibration
– Internal 16 MHz RC with PLL option (±1 %)
– Internal 32 kHz RC oscillator (±5 %)
• Up to 44 fast I/Os
– All mappable on external interrupt vectors
– Multiple 5 V-tolerant I/Os
• 5-channel DMA controller with flexible mapping
• 12-bit, 0.4 µs ADC (up to 16 ext. channels)
– Up to 16-bit with hardware oversampling
– Conversion range: 0 to 3.6V
• 11 timers (one 128 MHz capable): 16-bit for advanced motor control, one 32-bit and four 16-bit general-purpose, two low-power 16-bit, two watchdogs, SysTick timer
• Calendar RTC with alarm and periodic wakeup from Stop/Standby/Shutdown
• Communication interfaces
– Two I2C-bus interfaces supporting Fastmode Plus (1 Mbit/s) with extra current sink, one supporting SMBus/PMBus and wakeup from Stop mode
– Two USARTs with master/slave synchronous SPI; one supporting ISO7816 interface, LIN, IrDA capability, auto baud rate detection and wakeup feature
– One low-power UART
– Two SPIs (32 Mbit/s) with 4- to 16-bit programmable bitframe, one multiplexed with I2S interface
• Development support: serial wire debug (SWD)
• 96-bit unique ID

2.4 Integrated Step-Down Buck Converter LMR14006

This section provides detailed specifications for the buck converter.

2.4.1 Assumptions and prerequisites:

The following knowledge is assumed:

1. Basic understanding of DC-DC power modules and converter design
2. Basic understanding of power electronics and specifically MOSFETs

2.4.2 LMR14006

The LMR14006 is a PWM DC/DC buck (step-down) regulator. With a wide input range of 4 V-40 V, it is suitable for a wide range of application from industrial to automotive for power conditioning from an unregulated source. The regulator's standby current is 28 µA in sleep mode, which is suitable for battery operating systems. An ultra low 1 µA current can further prolong battery life in shutdown mode. Operating frequency is fixed at 1.1 MHz (X version) and 2.1 MHz (Y version) allowing the use of small external components while still being able to have low output ripple voltage. Soft-start and compensation circuits are implemented internally, and these allow the device to be used with minimized external components. The LMR14006 is optimized for up to 600 mA load current which has a 0.765 V typical feedback voltage. The device has built-in protection features such as pulse by pulse current limit, thermal sensing and shutdown due to excessive power dissipation. The LMR14006 is available in a low profile TSOT-6L package.

2.4.3 Features

For an in-depth description of the driver, please refer to the official Texas Instruments documentation at:

https://www.ti.com/lit/ds/symlink/lmr14006.pdf

- Ultra Low 28 µA Standby Current in Sleep Mode
- Input Voltage Range 4 V to 40 V
- 1 µA Shutdown Current
- High Duty Cycle Operation Supported
- Output Current up to 600 mA
- MHz and 2.1 MHz Switching Frequency
- Internal Compensation
- High Voltage Enable Input
- Internal Soft Start
- Over Current Protection
- Over Temperature Protection
- Small Overall Solution Size (TSOT-6L Package)

2.4.4 Applications

- Industrial Distributed Power Systems
- Automotive
- Battery Powered Equipment
- Portable Handheld Instruments
- Portable Media Players

2.5 Integrated BLDC Motor Driver Mosquito®

This section provides detailed specifications for the integrated BLDC motor driver.

2.5.1 Assumptions and prerequisites:

The following knowledge is assumed:

1. Basic understanding of PMSM motor design
2. Basic understanding of power electronics and specifically MOSFETs
3. Basic understanding of PMSM motor control
4. Basic understanding of microcontroller programming

2.5.2 Mosquito®

The Mosquito® is a fully self-contained, miniature 2.5-A Triple 1/2-H Bridge Driver integrated on a board with an STM32G031 microcontroller, a 3.3v, 500mA step-down buck converted and a set of capacitive snubbers and bulk buffer capacitors. The board provides three individually controllable half-H-bridge drivers. The Mosquito® device is intended to drive a three-phase brushless-DC motor, although it can also be used to drive solenoids or other loads. Each output driver channel consists of N-channel power MOSFETs configured in a 1/2-H-bridge configuration. Internal protection functions are provided for undervoltage, charge pump faults, overcurrent, short circuits, and overtemperature. Fault conditions are indicated by the nFAULT pin.

The Mosquito® has full protection against undervoltage, overcurrent, and overtemperature events. If at any time the voltage on the VM pin falls below the undervoltage threshold voltage (VUVLO), all FETs in the H-bridge will be disabled, the charge pump will be disabled, the internal logic is reset, and the nFAULT pin will be driven low. Operation will resume when VM rises above the UVLO threshold. The nFAULT pin will be released after operation has resumed.

If the die temperature exceeds safe limits, all FETs in the H-bridge will be disabled and the nFAULT pin will be driven low. Once the die temperature has fallen to a safe level operation will automatically resume. The nFAULT pin will be released after operation has resumed.

An analog current limit circuit on each FET limits the current through the FET by removing the gate drive. If this analog current limit persists for longer than tOCP, the device disables the channel experiencing the overcurrent and drives the nFAULT pin low. The driver remains off until either assertion of nRESET or the cycling of VM power.

Overcurrent conditions on both high- and low-side devices, that is, a short to ground, supply, or across the motor winding, all result in an overcurrent shutdown.

The Mosquito® is active unless the nSLEEP pin is brought logic low. In sleep mode the charge pump is disabled, the output FETs are disabled Hi-Z, and the V3P3 regulator is disabled. The driver is brought out of sleep mode automatically if nSLEEP is brought logic high.

The nRESET pin, when driven low, resets any faults. It also disables the output drivers while it is active. The device ignores all inputs while nRESET is active. Note that there is an internal power-up-reset circuit, so that driving nRESET at power up is not required. Driving nSLEEP low puts the device into a low-power sleep state. Entering this state disables the output drivers, stops the gate-drive charge pump, resets all internal logic (including faults), and stops all internal clocks. In this state, the device ignores all inputs until nSLEEP returns inactive-high. When returning from sleep mode, some time (approximately 1 ms) must pass before the Mosquito® driver becomes fully operational. The V3P3 regulator remains operational in sleep mode.

2.5.3 Operational Modes

Table 1. Logic States

INx	ENx	OUTx
X	0	Z
0	1	L
1	1	H

Table 2. Fault Condition Summary

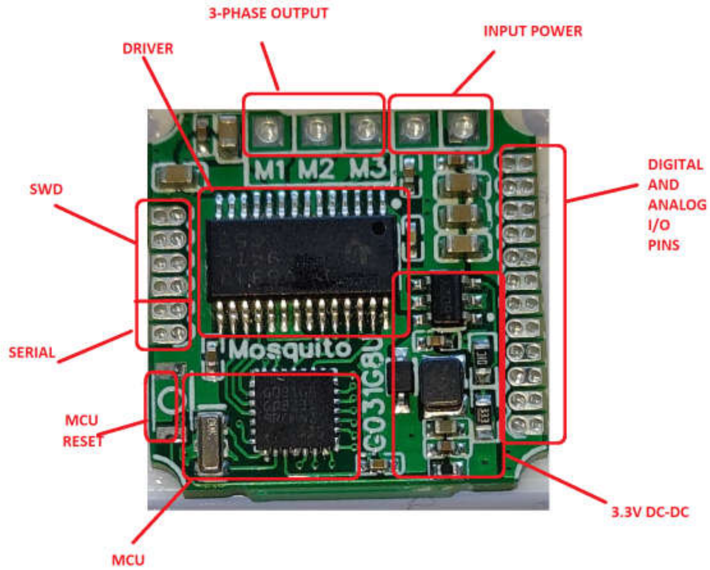
FAULT	CONDITION	ERROR REPORT	H-BRIDGE	CHARGE PUMP	V3P3	RECOVERY
VM undervoltage (UVLO)	$VM < V_{UVLO}$ (max 8 V)	nFAULT	Disabled	Disabled	Operating	$VM > V_{UVLO}$ (max 8 V)
Thermal Shutdown (TSD)	$T_J > T_{TSD}$ (min 150°C)	nFAULT	Disabled	Operating	Operating	$T_J < T_{TSD} - T_{HYS}$ (T_{HYS} typ 35°C)
Overcurrent (OCP)	$I_{OUT} > I_{OCP}$ (min 3 A)	nFAULT	Disabled	Operating	Operating	nRESET

Table 3. Functional Modes Summary

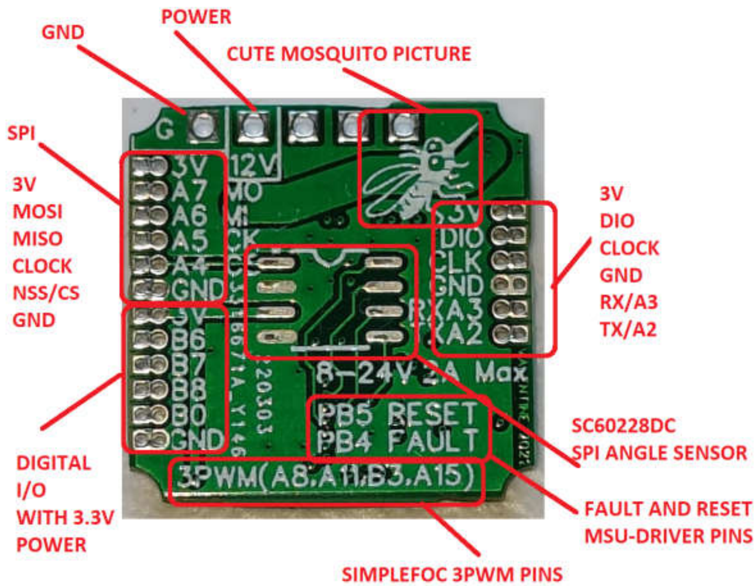
FAULT	CONDITION	H-BRIDGE	CHARGE PUMP	V3P3
Operating	8 V < VM < 60 V nSLEEP pin = 1	Operating	Operating	Operating
Sleep mode	8 V < VM < 60 V nSLEEP pin = 0	Disabled	Disabled	Disabled
Fault encountered	VM undervoltage (UVLO)	Disabled	Disabled	Operating
	Overcurrent (OCP)	Disabled	Operating	Operating
	Thermal shutdown (TSD)	Disabled	Operating	Operating

2.5.4 Board Pinouts

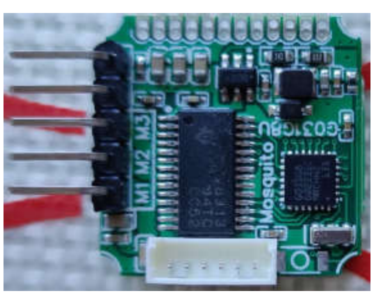
Front of the board



Back of the board



Board with pins and an SWD / Serial connector soldered on



2.5.5 Board Programming

The board is programmed using an ST-Link Debugger and Programmer hardware, STM32CubeProgrammer software, and an Arduino IDE. These are the parts of the hardware needed to program the Mosquito:

Mosquito board with SWD and serial connector (custom soldering required)

ST-Link Debugger

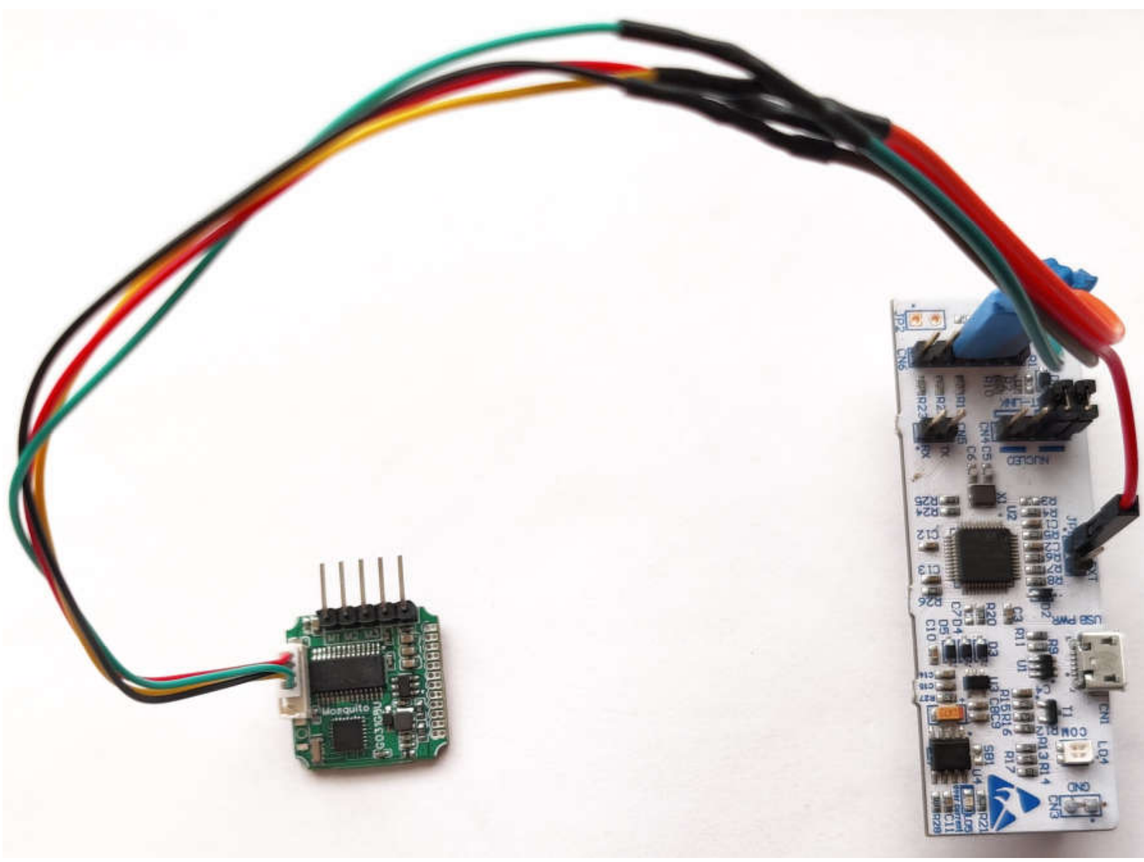
Computer with installed Arduino IDE

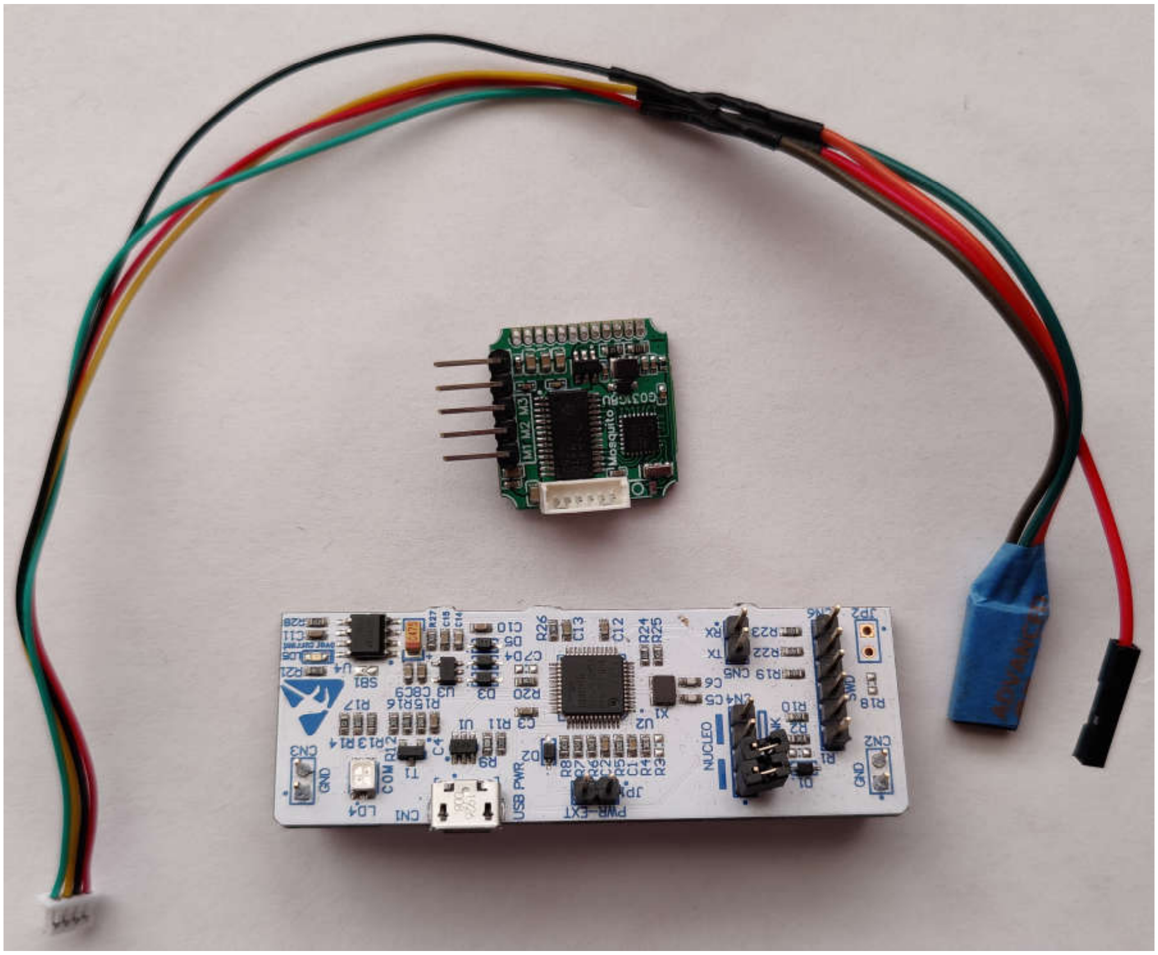
Computer with installed STM32CubeProgrammer

Micro-USB Cable

ST-Link to Mosquito cable (custom soldering required)

Picture of the core hardware components

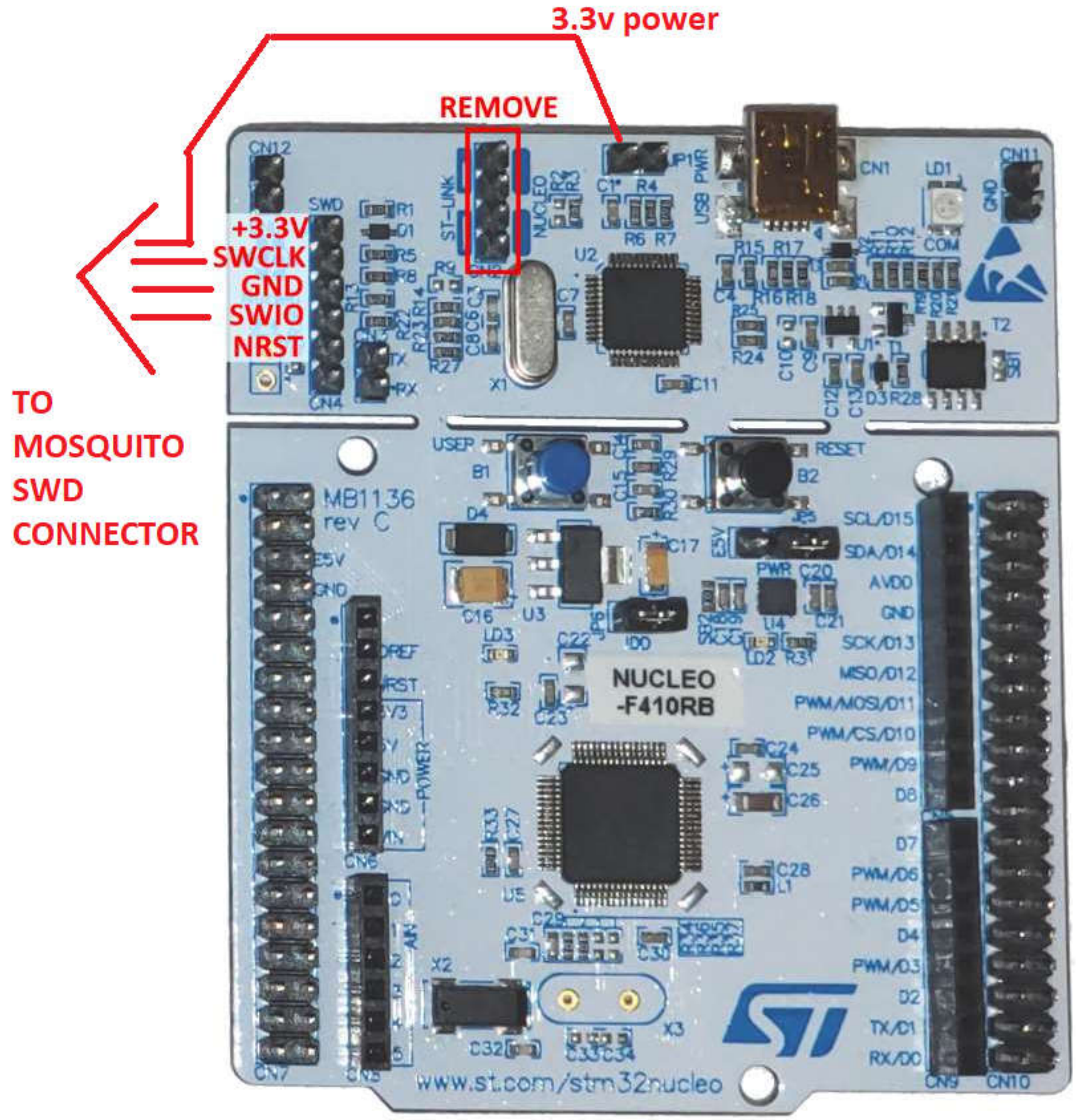




Connect the cable as shown, matching the exact pins and directions.

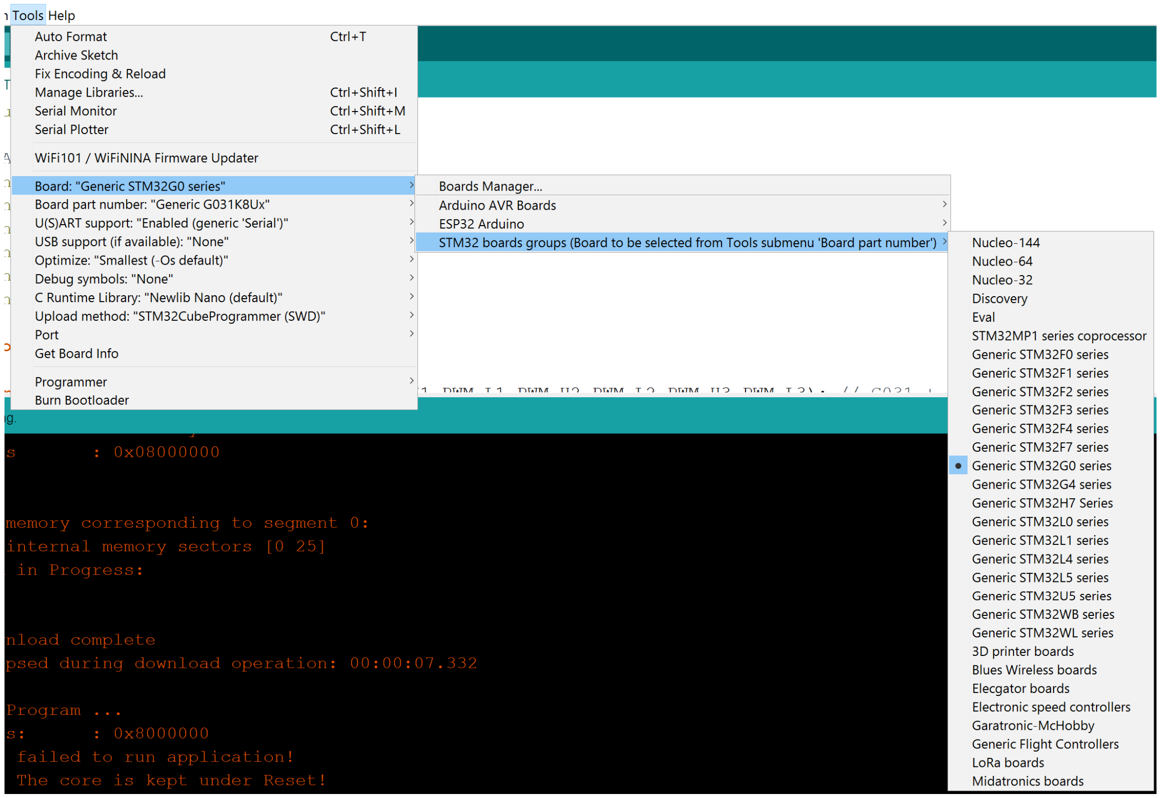
Custom soldering is required.

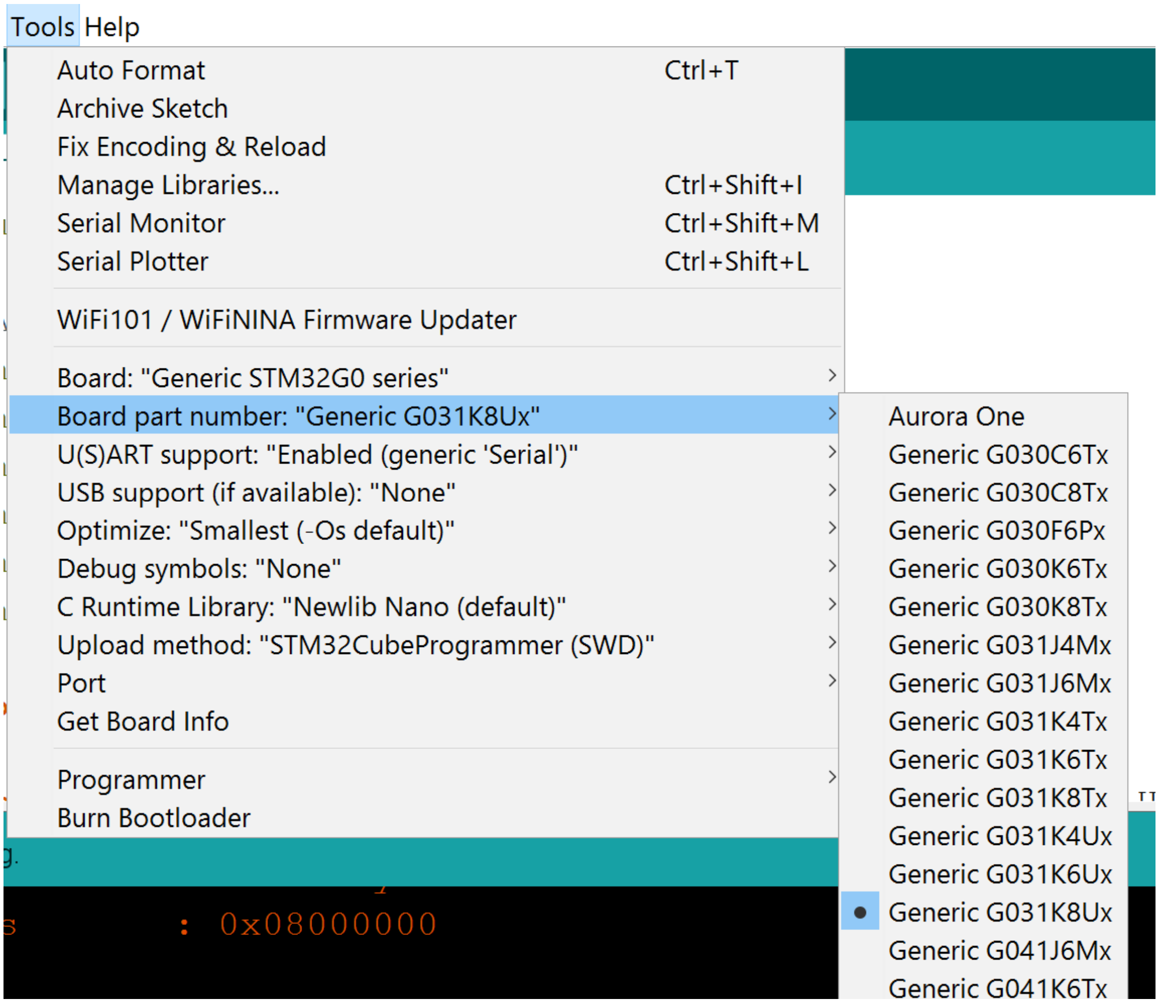
Connect only the 3.3v power from the outside pin, the SWCLK to clock, GND to GND, and SWDIO to DIO. Ignore the other pins. Do not use the 3.3v and NRST on the SWD ST-Link, leave unconnected.



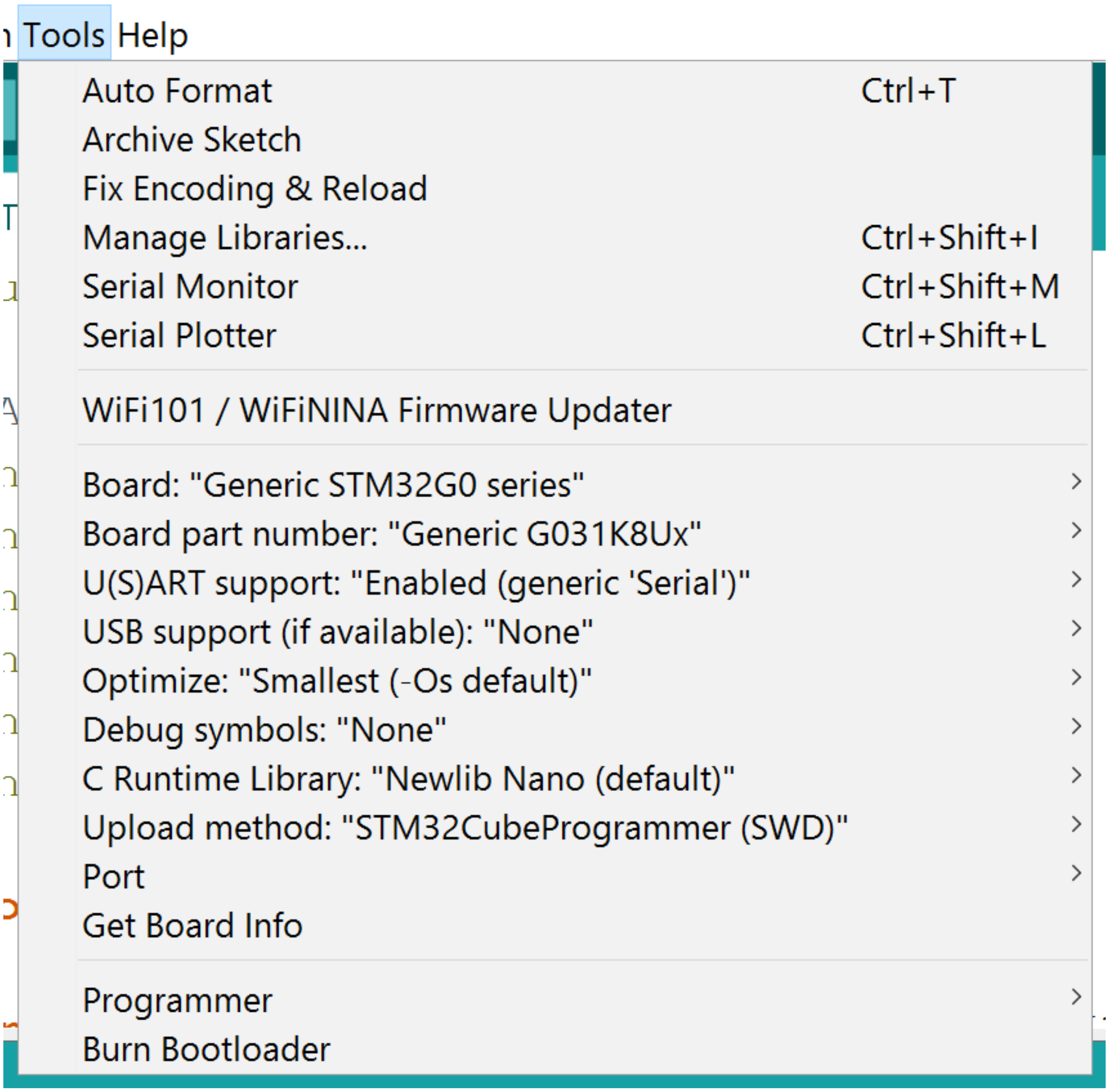
Now you are ready to program the Mosquito.

Open Arduino IDE, load the STM boards. Please refer online on how and where to load the STM boards in Arduino IDE. Also, install the STM Cube Programmer on the same machine. Once the boards are loaded, select the following MCU combination


Tools Help
Auto Format Ctrl+T
Archive Sketch
Fix Encoding & Reload
Manage Libraries... Ctrl+Shift+I
Serial Monitor Ctrl+Shift+M
Serial Plotter Ctrl+Shift+L
WiFi101 / WiFiNINA Firmware Updater
Board: "Generic STM32G0 series"
Board part number: "Generic G031K8Ux"
U(S)ART support: "Enabled (generic 'Serial')"
USB support (if available): "None"
Optimize: "Smallest (-Os default)"
Debug symbols: "None"
C Runtime Library: "Newlib Nano (default)"
Upload method: "STM32CubeProgrammer (SWD)"
Port
Get Board Info
Programmer
Burn Bootloader
Boards Manager...
Arduino AVR Boards
ESP32 Arduino
STM32 boards groups (Board to be selected from Tools submenu 'Board part number')
Nucleo-144
Nucleo-64
Nucleo-32
Discovery
Eval
STM32MP1 series coprocessor
Generic STM32F0 series
Generic STM32F1 series
Generic STM32F2 series
Generic STM32F3 series
Generic STM32F4 series
Generic STM32F7 series
Generic STM32G0 series
Generic STM32G4 series
Generic STM32H7 Series
Generic STM32L0 series
Generic STM32L1 series
Generic STM32L4 series
Generic STM32L5 series
Generic STM32U5 series
Generic STM32WB series
Generic STM32WL series
3D printer boards
Blues Wireless boards
Elecgator boards
Electronic speed controllers
Garatronic-McHobby
Generic Flight Controllers
LoRa boards
Midatronics boards
s : 0x08000000
memory corresponding to segment 0:
internal memory sectors [0 25]
in Progress:
nload complete
psed during download operation: 00:00:07.332
Program ...
s: : 0x8000000
failed to run application!
The core is kept under Reset!



This is the final board and part number and other settings you need to have



Once done, connect the entire setup: The Mosquito board, ST Link, USB to the computer, load the following program, and flash the Mosquito.

```
////////////////////////
// MOSQUITO BOARD
// SELECT STM32G031K8Ux

#include <SimpleFOC.h>

// TX/RX SERIAL PINS
#define serial1_tx_pin PA2
#define serial1_rx_pin PA3

// DRIVER PINS
#define PWM1 PA8
#define PWM2 PA11
```

```
#define PWM3 PB3
#define DRVENABLE PA15
#define DRVFAULT PB4
#define DRVRESET PB5

BLDCMotor motor = BLDCMotor(6);

BLDCDriver3PWM driver = BLDCDriver3PWM(PWM1, PWM2, PWM3, DRVENABLE); //
STM32G031K8Ux + MOSQUITO

float target_velocity = 5.0;

//loop delay debug print command variables
uint32_t txDlyPrint = 100;  // milli-seconds print delay counter
uint32_t lastprint = 0;     // temp delay counter variable

HardwareSerial Serial1(serial1_rx_pin, serial1_tx_pin);

void setup() {

  pinMode(DRVFAULT, INPUT); // Fault pin

  pinMode(DRVRESET, OUTPUT); // Reset pin

  digitalWrite(DRVRESET, HIGH); // Default low

  Serial1.begin(115200);
  Serial1.println("Motor ready.");
  _delay(500);

  driver.voltage_power_supply = 12;

  driver.pwm_frequency = 15000;

  driver.init();

  motor.linkDriver(&driver);

  motor.voltage_limit = 4;   // [V]
  motor.velocity_limit = 1000; // [rad/s] cca 50rpm

  motor.controller = MotionControlType::velocity_openloop;

  motor.init();

}

void loop() {

  if (millis() / txDlyPrint != lastprint)    // send every txDlyPrint
  {
```

```
    lastprint = millis() / txDlyPrint;

    // display something to the terminal
    //Serial1.print("something");
    //Serial1.println("");

  }

  motor.move(target_velocity);
}
///////////////////
```

The board will flash. Connect 12V power and a gimbal motor and the motor will slowly spin.