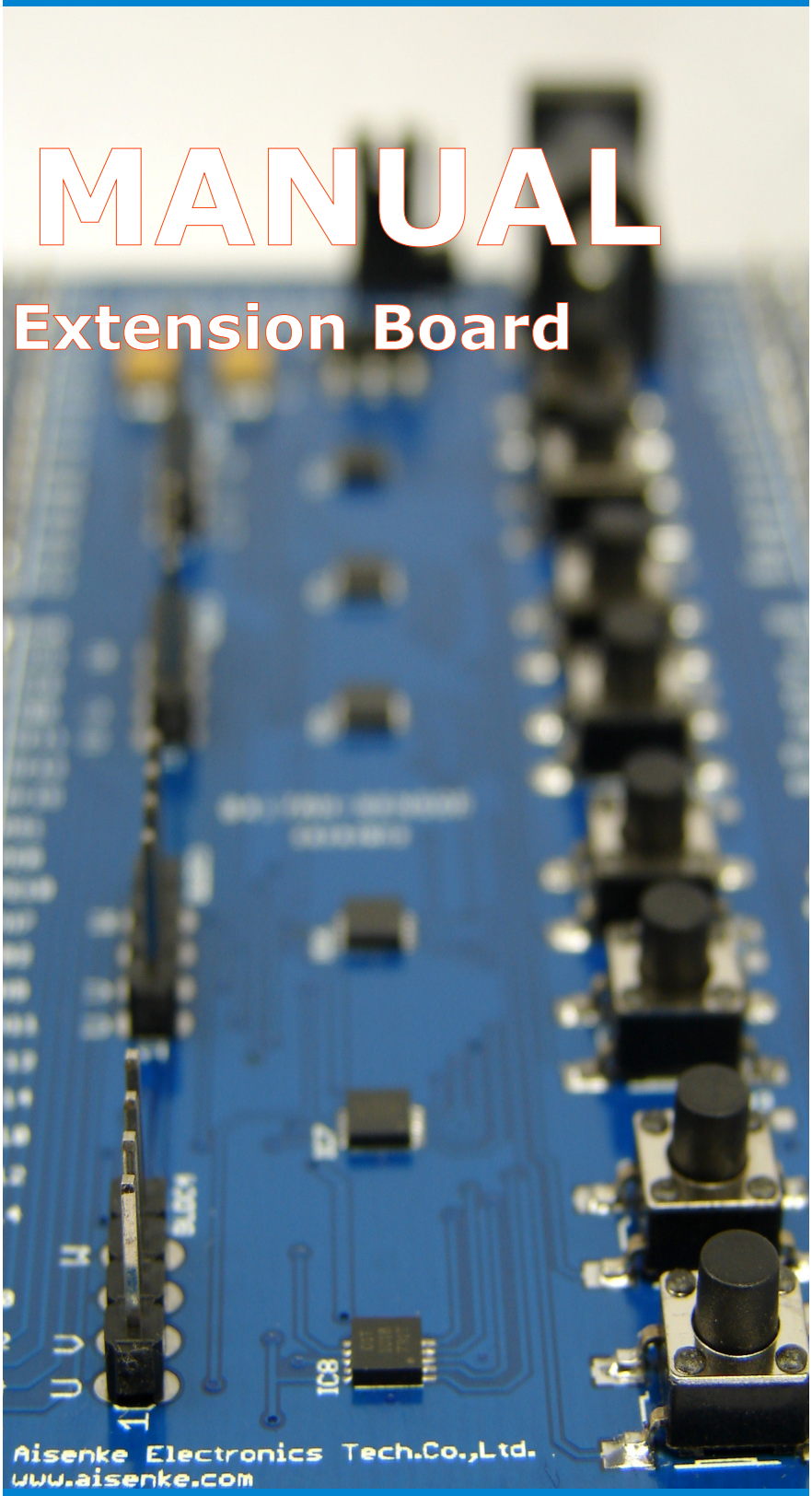
F4-DiscoverFree Kit Series

USER MANUAL

BLDC Motor Extension Board

Version R1.0

UM-EB-0001



Chongqing Aisenke Electronic Technology Co.,Ltd.

Phone/Fax: +86-2363743515
Email: info@aisenke.com

User Manual for BLDC Motor Extension Board R1.0

This user manual contains information on how to install and operate the BLDC motor extension board atop of F4-DiscoverFree kit.

First release, January 2013

Nodino Robotics is a registered trademark of Aisenke Electronic Technology Co.,Ltd. For more information about this project and F4-DiscoverFree, visit us online at http://www.aisenke.com

Table of Contents

1 Introduction....6
2 Getting Started....7
2.1 System Setup....7
2.2 Hardware References....7
2.3 Software Requirements....10
2.3.1 PC Software....10
2.3.1.1 PuTTY Setup on Windows....10
2.3.1.2 PuTTY Setup on Ubuntu Linux....12
2.3.1.3 PuTTY Setup on Mac OS X....14
2.3.2 BLDC Firmware....15
2.3.3 FTDI Driver....16
3 Hardware References....17
3.1 Jumper SV5....17
3.2 DC 5V Socket....18
3.3 BLDC Motor Interface....18
3.4 Push Buttons....19
4 Command References....20
4.1 Motor Properties....20
4.1.1 position command....20
4.1.2 position_reset command....21
4.1.3 anchor command....21
4.1.4 anchor_set command....21
4.2 Controller Properties....22
4.2.1 bemf command....22
4.2.2 bemf_on command....22
4.2.3 bemf_off command....22
4.2.4 button command....22
4.2.5 button_on command....23
4.2.6 button_off command....23
4.3 BLDC Movement....23
4.3.1 direction command....23
4.3.2 step_cw [n] command....23
4.3.3 step_ccw [n] command....24
4.3.4 cw command....24
4.3.5 ccw command....24
4.3.6 stop command....24
5 Schematic....25
6 Revision History....26

Illustration Index

Illustration 1: BLDC Motor Extension Board from top view..... 6
Illustration 2: Hardware setup with a computer or laptop..... 7
Illustration 3: BLDC-MEB top component description.....8
Illustration 4: BLDC-MEB bottom component description..... 9
Illustration 5: PuTTY Session setting on Windows..... 10
Illustration 6: PuTTY Terminal setting on Windows..... 11
Illustration 7: PuTTY Terminal Session is running on Windows.....11
Illustration 8: PUTTY Session setting on Linux..... 12
Illustration 9: PuTTY Terminal setting on Linux..... 13
Illustration 10: PuTTY Terminal Session is running on Linux.....13
Illustration 11: PuTTY Session setting on Mac OS X.....14
Illustration 12: PuTTY Terminal setting on Mac OS X.....15
Illustration 13: PuTTY Terminal Session is running on Mac OS X.....15
Illustration 14: Jumper pin at 2 and 3, separating external power to USB power.....17
Illustration 15: Jumper pin at 1 and 2, combining both external and USB power line.....17
Illustration 16: Power supply connected to BLDC-MEB.....18
Illustration 17: BLDC pins: U, V, N (not used) and W.....19
Illustration 18: Programmable CW and CCW buttons.....19
Illustration 19: BLDC help screen..... 20
Illustration 20: BLDC-MEB Schematic.....25

Index of Tables

Table 1: Revision history.....26

1 Introduction

Brushless DC (BLDC) Motor Extension Board (MEB) is powered by Allegro Microsystems' chips to drive four brushless DC motors at 3.3V with maximum amperage up to 400mA for each one. The driving method is square wave commutation with back EMF feedback reading, also known as sensor-less BLDC driving.

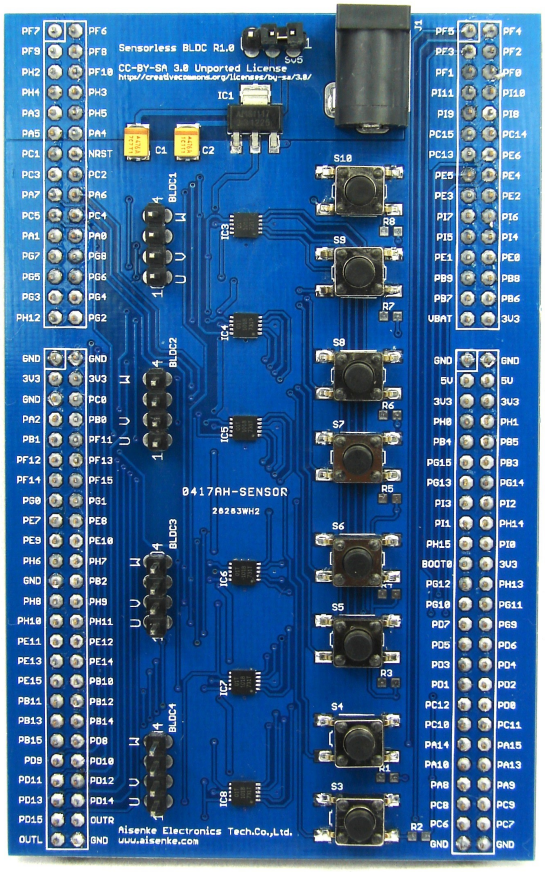


Illustration 1: BLDC Motor Extension Board from top view

The extension board merely includes motor driver chips, push buttons as inputs and voltage regulator, to be used together with an F4-DiscoverFree development kit. There is no firmware required on the BLDC-MEB. However a firmware update is necessary on the F4-DiscoverFree board.

Motor pins assignment are compatible with Namiki brushless DC motors. However, it is easy to adapt to other brushless DC motors since U, V and W connectors are provided on the board as generic single row headers with standard 2.54mm pitch.

2 Getting Started

2.1 System Setup

The system setup you need to prepare should look similar to illustration 2 containing F4-DiscoverFree with BLDC-MEB, a computer and one to four brushless motors connected:

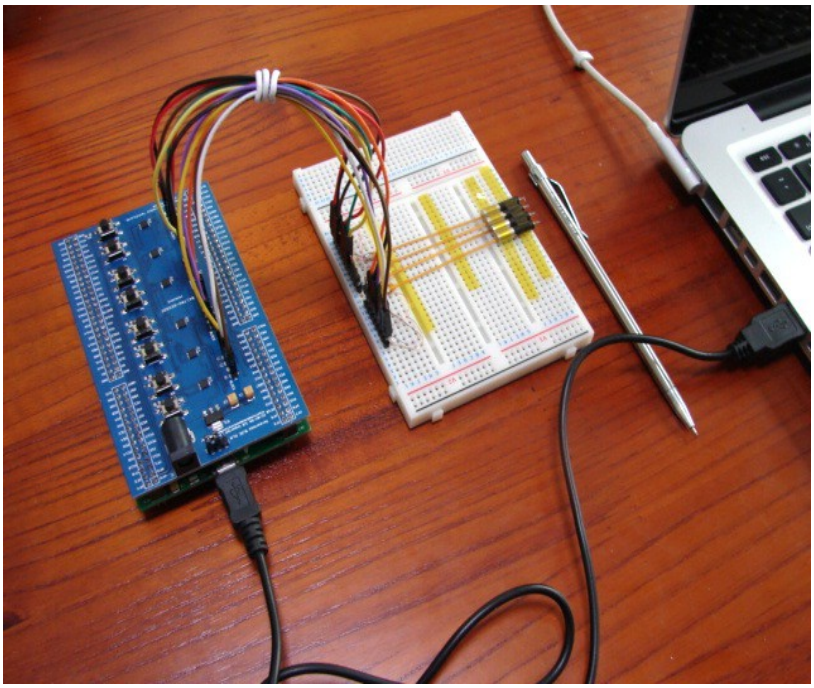


Illustration 2: Hardware setup with a computer or laptop

Detail hardware required to start developing application with BLDC-MEB are:

- F4-DiscoverFree R1.3 or later
- BLDC-MEB R1.0
- One BLDC motors with less than 400mA stall current
- One USB type A to mini B cable
- Computer running Windows, Linux or Mac OS X with one USB port
- Some wiring and soldering for motor interfacing

2.2 Hardware References

Detail hardware references can be found on page 17. Figure 3 and figure 4 briefly explain all interfaces available on top and bottom layers of BLDC-MEB:

- JP SV5 is a three-pins headers with a jumper to separate or combine the external power supply to the USB power line.

- DC 5V is an external power supply socket. Add an external power supply if the stall current of all motors combined exceed 400 mA.

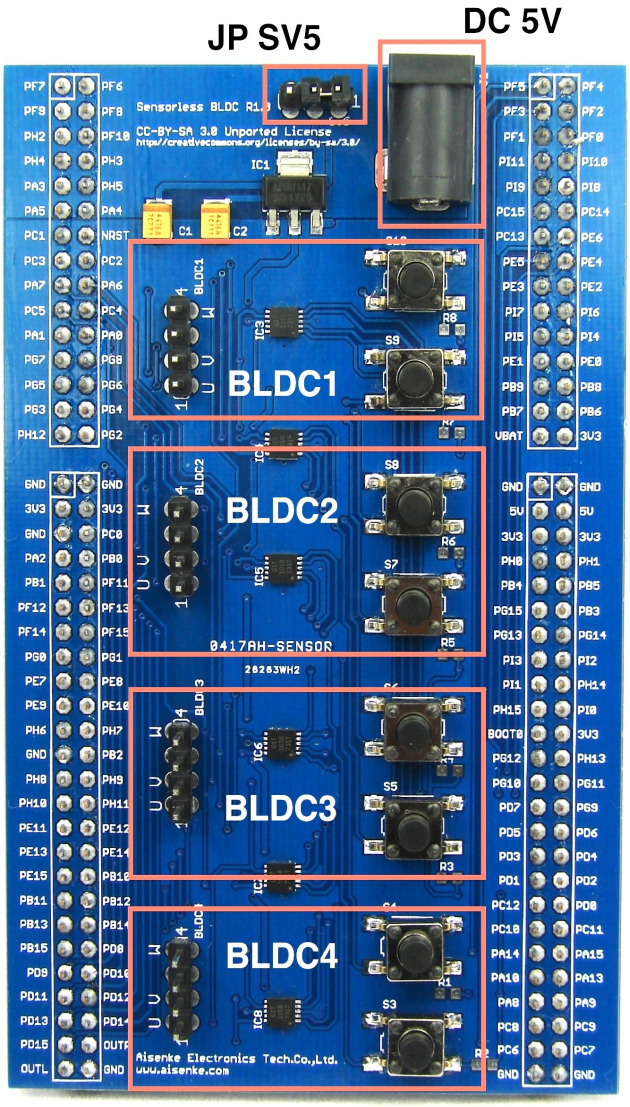


Illustration 3: BLDC-MEB top component description

- BLDC1 area contains a four-pins header as output to motor and two programmable push buttons for manual control.
- BLDC2 area contains a four-pins header as output to motor and two programmable push buttons for manual control.
- BLDC3 area contains a four-pins header as output to motor and two programmable push buttons for manual control.

- BLDC4 area contains a four-pins header as output to motor and two programmable push buttons for manual control.

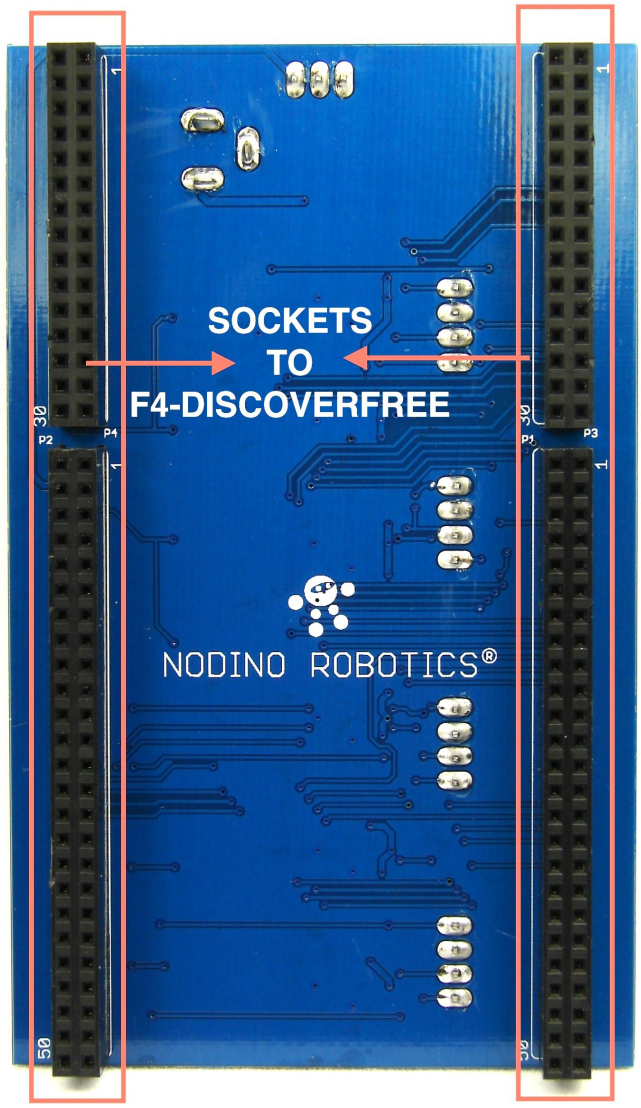


Illustration 4: BLDC-MEB bottom component description

- At the bottom layer, two lines containing each 2x15 and 2x25 sockets are board interconnection to F4-DiscoverFree board. Please note carefully that the BLDC-MEB is stacked up on the F4-DiscoverFree (NOT FROM THE BOTTOM).

2.3 Software Requirements

2.3.1 PC Software

There is no particular PC Software provided with BLDC-MEB kit. The only interface you need is a serial terminal. You may choose the terminal program like HyperTerminal, TeraTerm, PuTTY or picocom. The following subsections describe how to setup PuTTY on Windows, Linux and Mac OS X.

2.3.1.1 PuTTY Setup on Windows

Download the latest version of PuTTY from http://www.chiark.greenend.org.uk/~sgtatham/putty and put it on your Desktop. Double click to start a terminal session.

In the configuration window, choose the connection type as Serial with the speed of 115200, and the port number may vary depending on your Windows setup. Figure 5 is showing an example of PuTTY Session configuration on COM4.

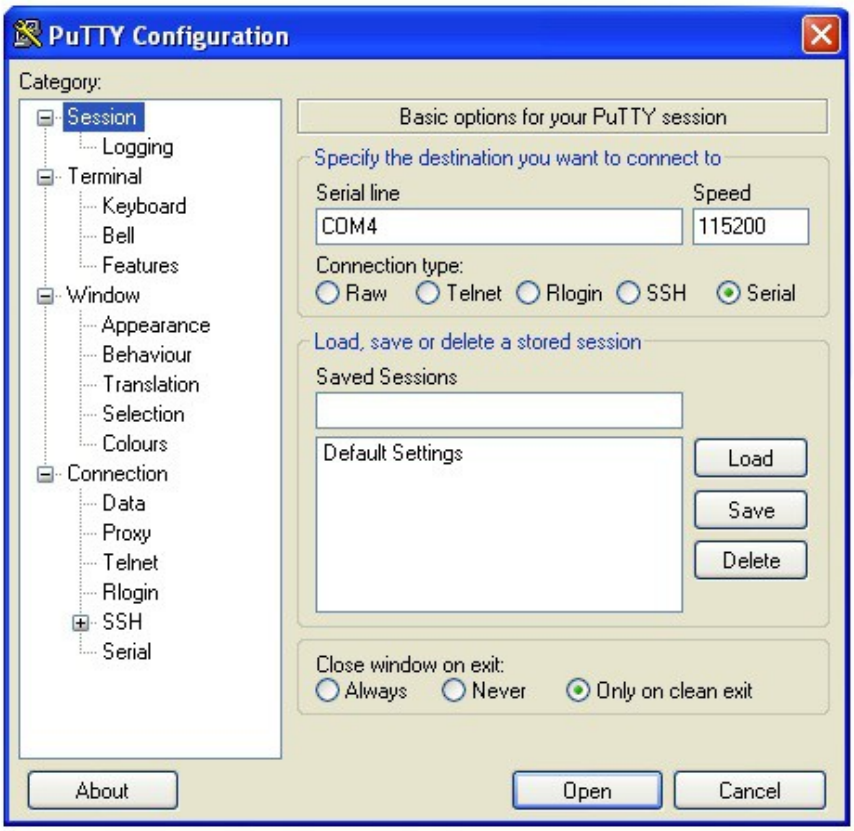


Illustration 5: PuTTY Session setting on Windows

Under category on the left side, select Terminal and make sure Implicit CR every LF is ticked. Leave everything else as their default value as illustrated in figure 6.

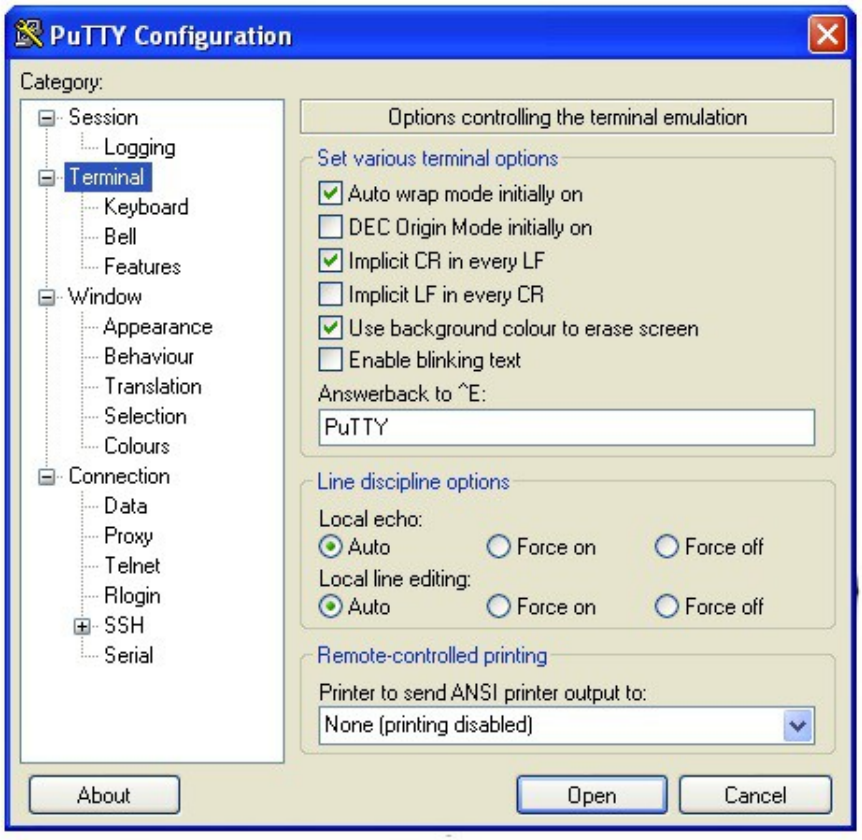


Illustration 6: PuTTY Terminal setting on Windows

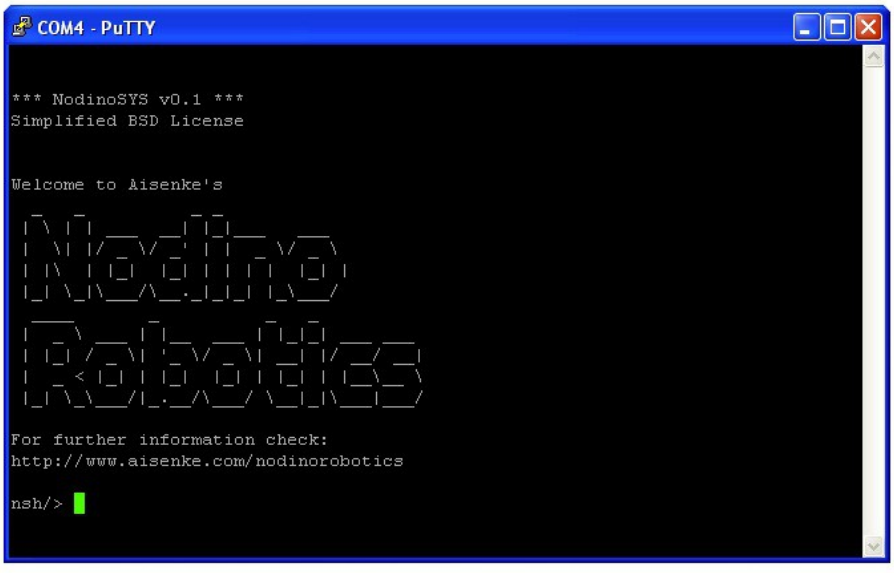


Illustration 7: PuTTY Terminal Session is running on Windows

After you are done with the configuration, click Open and the terminal session is running. You may type BLDC commands

through terminal.

2.3.1.2 PuTTY Setup on Ubuntu Linux

To install PuTTY on Ubuntu Linux, go to Synaptic Package Manager and search for PuTTY. Alternatively, you may also open a terminal and type:

```
sudo apt-get install putty && putty
```

You need to provide root access to install it.

In the configuration window, choose the connection type as Serial with the speed of 115200, and the device name/number may vary depending on your Linux setup. Figure 8 is showing an example of PuTTY Session configuration on /dev/ttyUSB1 device.

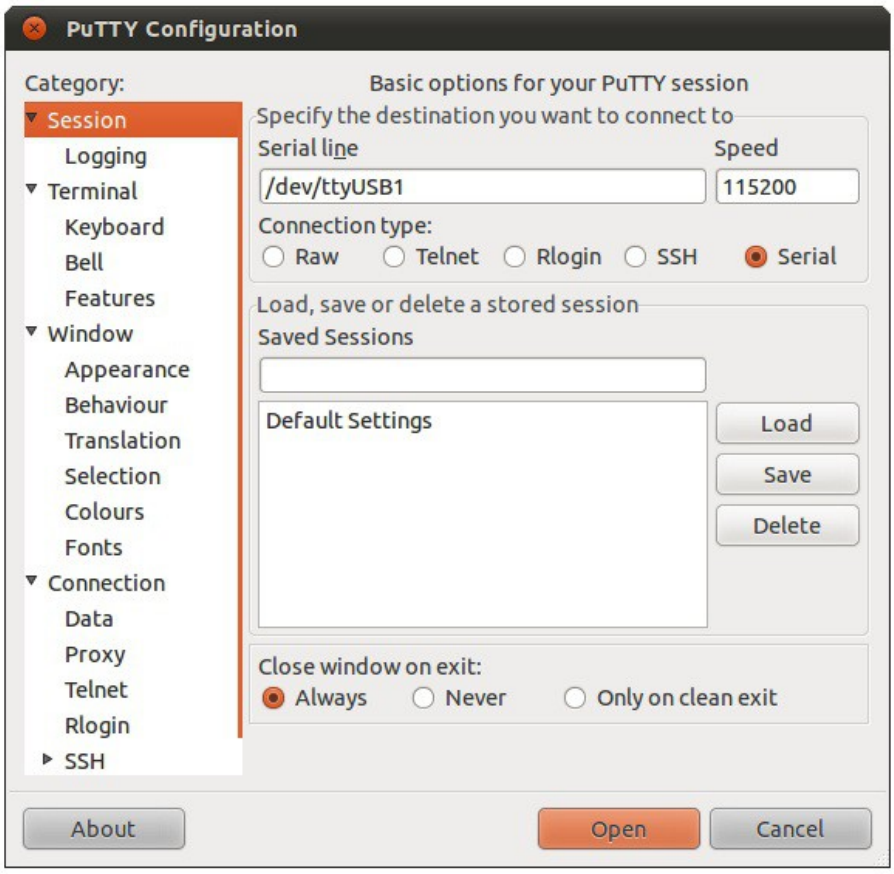


Illustration 8: PUTTY Session setting on Linux

Under category on the left side, select Terminal and make sure Implicit CR every LF is ticked. Leave everything else as their default value as illustrated in figure 9.

After you are done with the configuration, click Open and the terminal session is running. You may type BLDC commands through terminal.

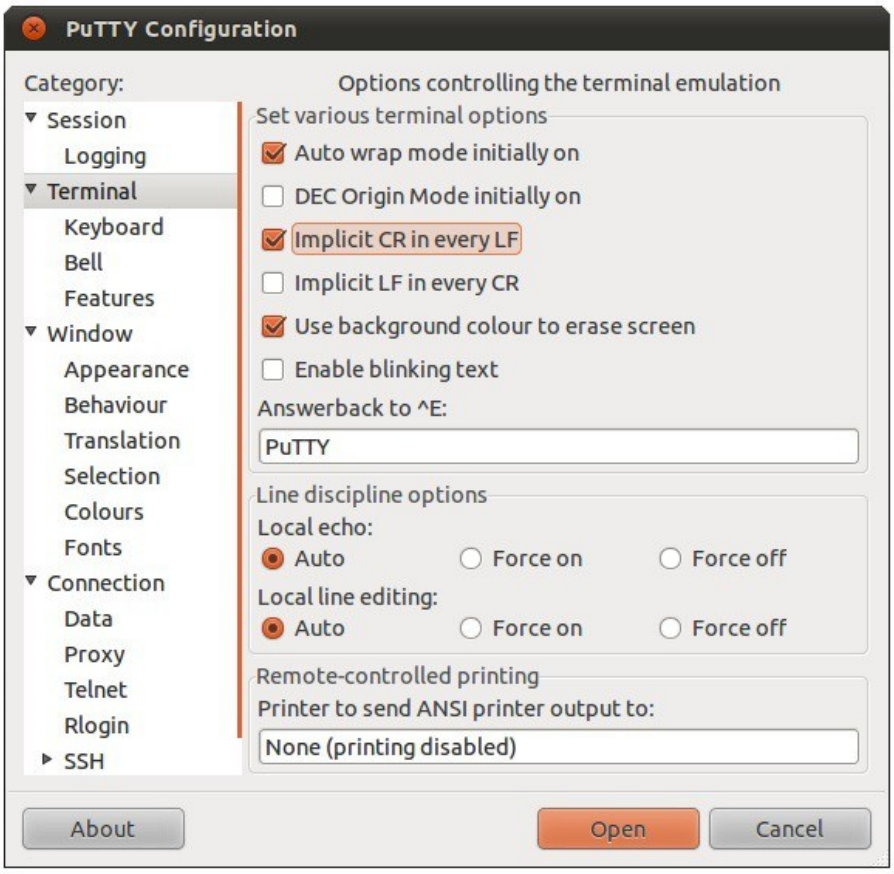


Illustration 9: PuTTY Terminal setting on Linux

Illustration 10: PuTTY Terminal Session is running on Linux

2.3.1.3 PuTTY Setup on Mac OS X

To install PuTTY on Mac OS X, you need to install macports or brew. Below is an example of installing PuTTY using macports:

```
sudo port install putty && putty
```

You need to provide root access to install it.

In the configuration window, choose the connection type as Serial with the speed of 115200, and the device name/number may vary depending on your Mac OS X setup. Figure 11 is showing an example of PuTTY Session configuration on /dev/tty.usbserial-00001004B device.

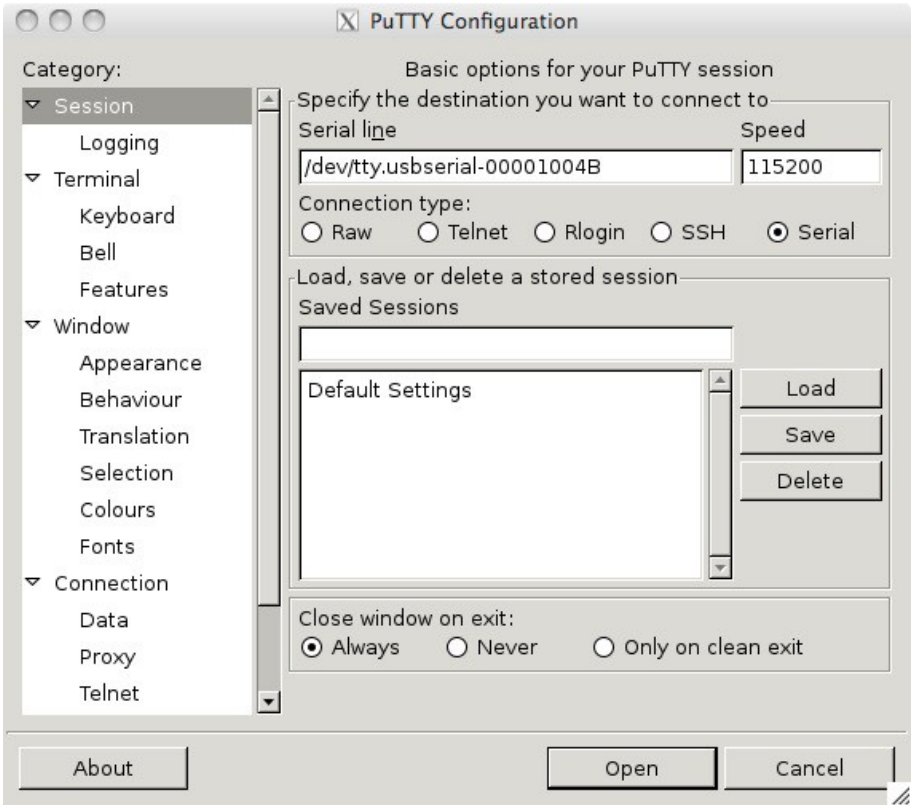


Illustration 11: PuTTY Session setting on Mac OS X

Under category on the left side, select Terminal and make sure Implicit CR every LF is ticked. Leave everything else as their default value as illustrated in figure 12.

After you are done with the configuration, click Open and the terminal session is running. You may type BLDC commands through terminal.

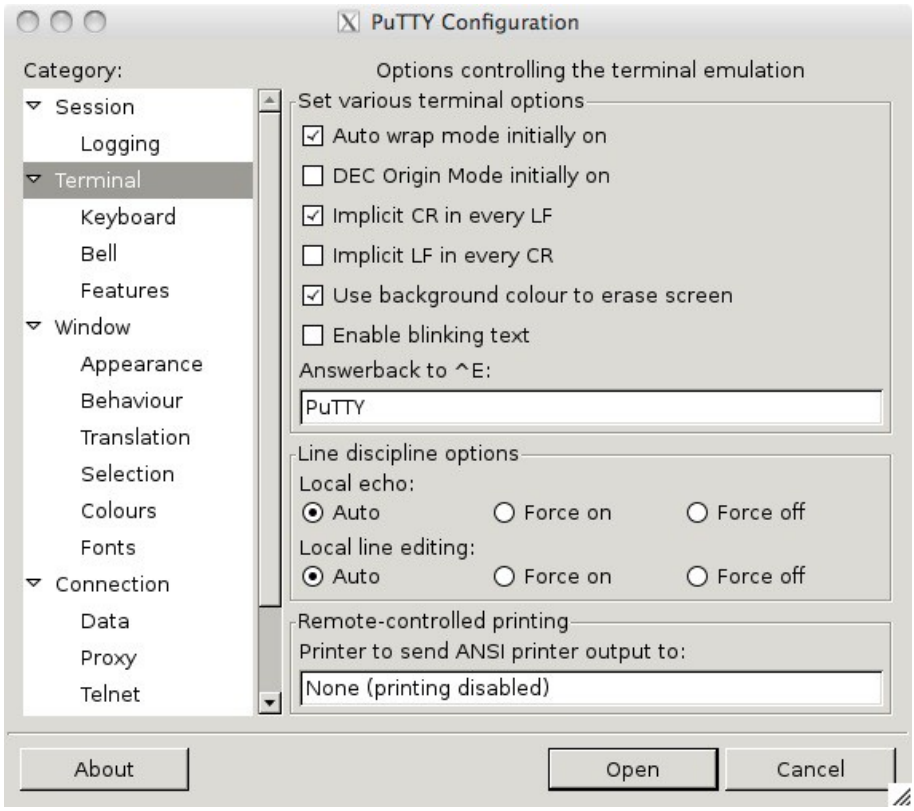


Illustration 12: PuTTY Terminal setting on Mac OS X

Illustration 13: PuTTY Terminal Session is running on Mac OS X

2.3.2 BLDC Firmware

If you purchase the BLDC-MEB bundled with an F4-DiscoverFree,

the BLDC firmware is already preinstalled within the F4-DiscoverFree. This firmware is ready to use with any serial terminal.

If you already owned an F4-DiscoverFree and purchase BLDC-MEB separately, then you need to download the firmware from our repository and upload it. You may also want to download the source code if you want to customize to your need. Please refer to F4-DiscoverFree User Manual to update/upload the new firmware and to develop custom application on F4-DiscoverFree.

2.3.3 FTDI Driver

Running serial terminal on Windows might need to download and install FTDI driver chip manually in case the Windows is not able to find it online and install the driver automatically. Please refer to F4-DiscoverFree User Manual on how to do it.

Mac OS X requires you to download and install the driver manually. Please refer to F4-DiscoverFree User Manual.

Linux kernel includes FTDI driver starting from kernel version 2.6.x and later. There is no need to install the driver from FTDI chip.

3 Hardware References

3.1 Jumper SV5

Jumper SV5 by default is put between pin 2 and 3 (or left open) as seen in figure 14. This configuration separates the 5V external power supply from the USB 5V line coming from F4-DiscoverFree board. This is the most recommended configuration to drive BLDC motors with high electrical current (more than 400 mA in total).

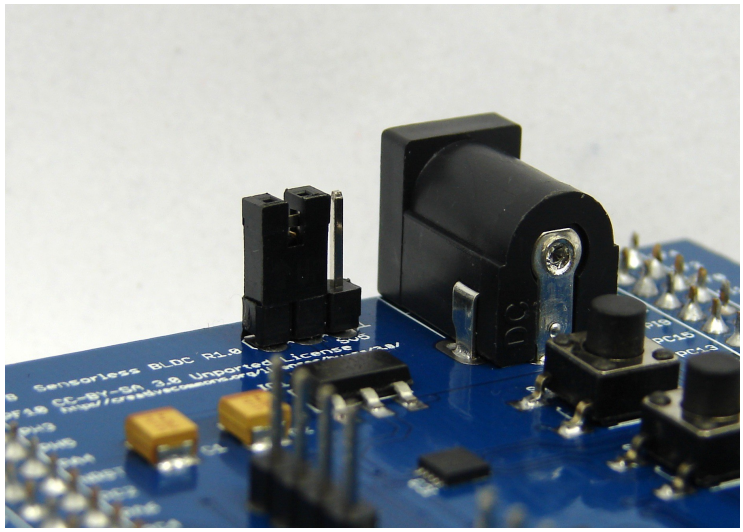


Illustration 14: Jumper pin at 2 and 3, separating external power to USB power

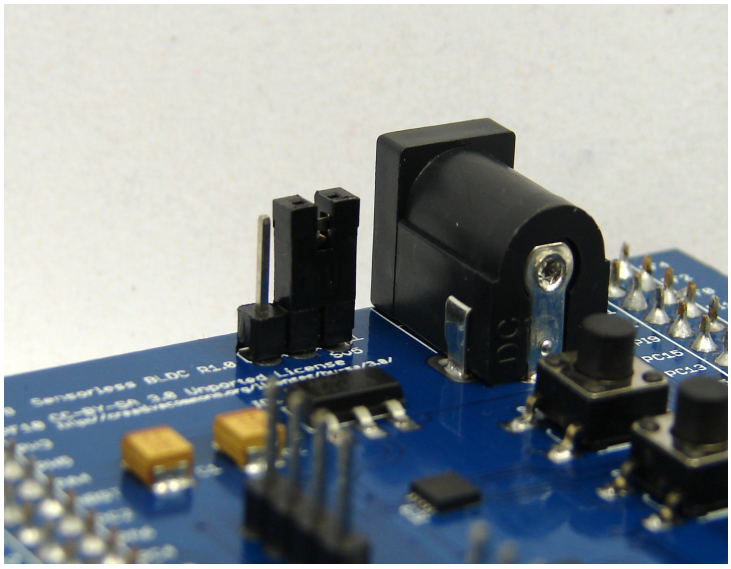


Illustration 15: Jumper pin at 1 and 2, combining both external and USB power line

In some cases, you might use micro BLDC motors with less than 400 mA current. This amount of current can be provided by the computer through USB port. If this is the preferable configuration, put the a jumper between pin 1 and 2 as seen on figure 15. This way, the BLDC-MEB and F4-DiscoverFree is supplied through the USB cable. Please do not withdraw more than 500 mA as specified in the USB specification.

Another example where you need this configuration, is when you use the board as a standalone controller utilizing only push buttons. Without the USB cable connected to a computer, you need to power the F4-DiscoverFree from the BLDC-MEB module. Put the jumper between 1 and 2 as seen on figure 15.

3.2 DC 5V Socket

The external power supply specification is DC regulated 5V able to deliver at least 2A of electrical current. The standard 2.1mm x 5.5mm male plug should fit in the socket provided on board.

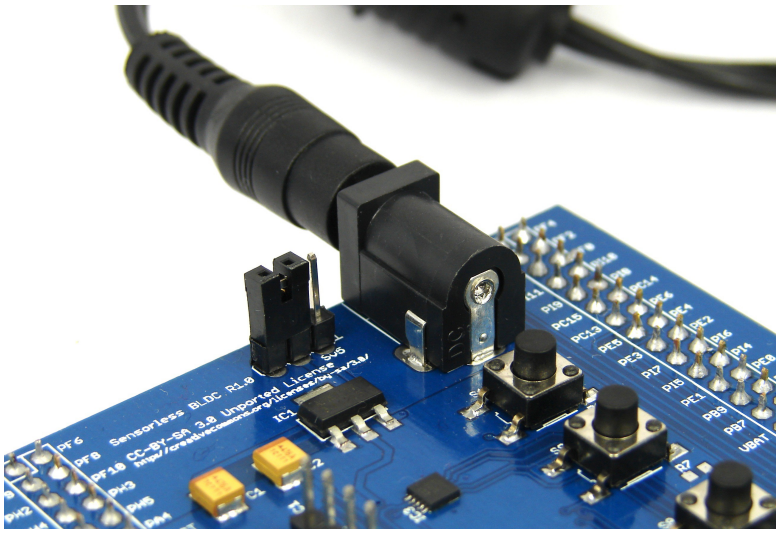


Illustration 16: Power supply connected to BLDC-MEB

Please refer to page 17 for the jumper setting in order to separate or combine between external power line to the USB power line.

3.3 BLDC Motor Interface

Standard 3-phase BLDC motors have at least 3 pins namely U, V and W. Some type of BLDC motors also provide a neutral pin N as seen in figure 17. BLDC-MEB uses only pin U, V and W to perform sensor-less driving and the neutral pin N is not connected on-board. You may connect it or leave it open on BLDC-MEB.

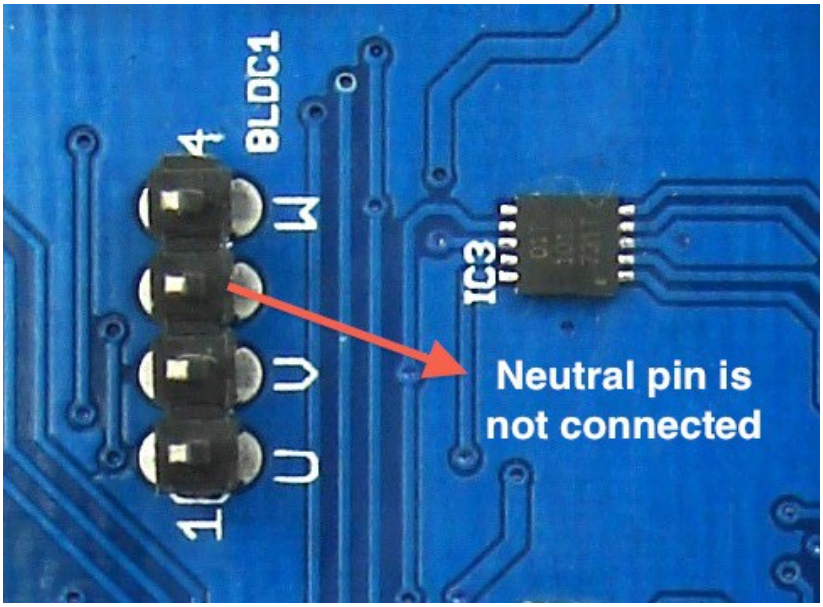


Illustration 17: BLDC pins: U, V, N (not used) and W

The header pins are of 2.54 mm pitch and any wiring may suffice to interface to your BLDC motors.

3.4 Push Buttons

The default firmware for each BLDC motor assigns a push button depicted in figure 18 at the top as a programmable clockwise button and the other as a programmable counter clockwise button. The term programmable means that each button can be enabled/disabled, toggling between manual or software control.

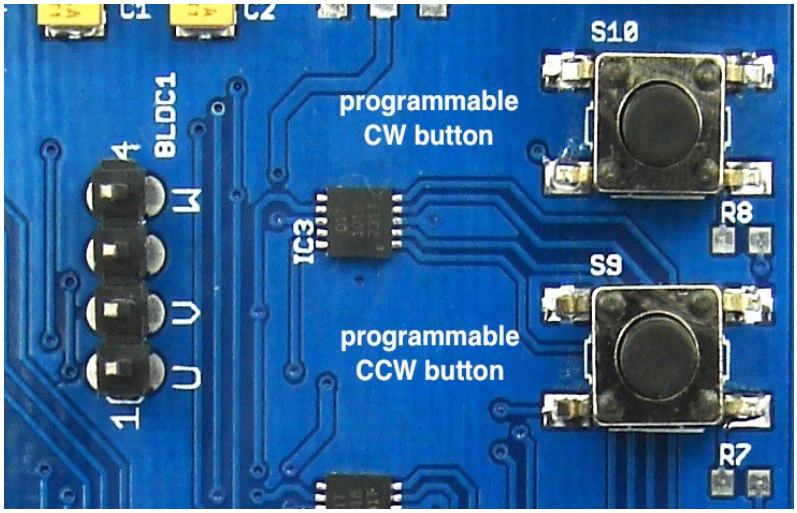


Illustration 18: Programmable CW and CCW buttons

4 Command References

The BLDC application v0.1 firmware introduces three categories of commands:

- Command to change brushless motor properties
- Command to change brushless controller properties
- Command to move brushless motors

To list all of commands available, type:

```
help bldc
```

You will get the command listing as seen on figure 19.

Illustration 19: BLDC help screen

4.1 Motor Properties

4.1.1 position command

Brushless motor's rotor position is held by position property. The position command is read-only and it returns the current rotor position. This command can be used while the motor is stationary or turning. Example of use:

```
bldc -p 0 position
> /dev/bldc0 position = 1234
```

4.1.2 position_reset command

This command resets the rotor position counter to zero. Example of use:

```
bldc -p 0 position_reset
bldc -p 0 position
> /dev/bldc0 position = 0
```

It is recommended to use this command only when the motor is stationary. Otherwise, position accuracy cannot be guaranteed.

4.1.3 anchor command

Anchor holds the value of rotor position to which the controller will bring to stop automatically when the current rotor position is equal to anchor value set by anchor_set command. Anchor command is read-only. Example of use:

```
bldc -p 0 anchor
> /dev/bldc0 anchor = 0
```

4.1.4 anchor_set command

This command sets a new anchor value at position [pos]. The value can be a negative or positive number. Example of use:

```
bldc -p 0 anchor_set 1234
bldc -p 0 anchor
> /dev/bldc0 anchor = 1234
```

Changing anchor value can be done while the motor is stationary or turning. However, please note:

- If the motor is stationary, changing anchor value will not bring the motor to move automatically toward any direction until *cw* or *ccw* command is executed.
- If the motor is turning toward CW direction, then setting anchor value to less than the current rotor position will never bring the motor to stop automatically.
- If the motor is turning toward CCW direction, then setting anchor value to greater than the current rotor position will never bring the motor to stop automatically.

The design of brushless controller is always comparing the value of rotor position and its anchor. If both values are the same and the motor is currently turning, the controller immediately brings the motor to a complete stop.

4.2 Controller Properties

4.2.1 bemf command

This command is read-only and it returns the current BEMF feedback controller mode enabled/disabled. Return value 1 = BEMF enabled; 0 = BEMF disabled. When BEMF feedback is disabled, the motor moves much slower but the rotor position accuracy can be guaranteed. Example os use:

```
bldc -p 0 bemf
> /dev/bldc0 bemf mode = 1
```

4.2.2 bemf_on command

This command is write-only and used to turn BEMF feedback controller mode to ON. Example of use:

```
bldc -p 0 bemf_on
bldc -p 0 bemf
> /dev/bldc0 bemf mode = 1
```

Please note that turning on BEMF feedback mode to ON while the motor is turning, will not guarantee rotor position accuracy since miscalculation may occur during mode transition. It is suggested that the motor is always stationary prior to changing BEMF feedback mode.

4.2.3 bemf_off command

This command is write-only and used to turn BEMF feedback controller mode to OFF. Example of use:

```
bldc -p 0 bemf_off
bldc -p 0 bemf
> /dev/bldc0 bemf mode = 0
```

Please note that turning off BEMF feedback mode to OFF while the motor is turning, will not take effect immediately. The mode is effective after the motor has stopped.

4.2.4 button command

This command is read-only and it returns the status of button mode enabled/disabled. Return value 1 = button enabled; 0 = button disabled. Example of use:

```
bldc -p 0 button
> /dev/bldc0 button mode = 1
```

4.2.5 button_on command

This command is write-only and used to turn button mode to ON. Example of use:

```
bldc -p 0 button_on
bldc -p 0 button
> /dev/bldc0 button mode = 1
```

When button is enabled, the motor control priority goes to push buttons. Command line can still be used, but limited to position read/write, anchor read/write, turning on/off BEMF mode and of course turning on/off button mode. The button mode is enabled by default, when the BLDC-MEB is powered up.

4.2.6 button_off command

This command is write-only and used to turn button mode to OFF. Example of use:

```
bldc -p 0 button_off
bldc -p 0 button
> /dev/bldc0 button mode = 0
```

When button is disabled, motor control is transferred from push button control to command line control. Buttons cannot be used until they are re-enabled.

4.3 BLDC Movement

4.3.1 direction command

This command is read-only and it returns the current motor direction. Return value 1 = turning CW; 0 = stop; -1 = turning CCW. Example of use:

```
bldc -p 0 direction
> /dev/bldc0 direction = 1
```

4.3.2 step_cw [n] command

This command turns the motor toward CW direction for [n] steps. Example of use:

```
bldc -p 0 step_cw 2024
```

No return result will be given whether the motor stops or currently is moving. You need to poll the motor status by executing *direction* command. The value n can be a negative or positive number.

4.3.3 step_ccw [n] command

This command turns the motor toward CCW direction for [n] steps. Example of use:

```
bldc -p 0 step_ccw 2024
```

No return result will be given whether the motor stops or currently is moving. You need to poll the motor status by executing *direction* command. The value n can be a negative or positive number.

4.3.4 cw command

This command infinitely turns the motor toward CW direction until stop command is executed or rotor position is equal to anchor value. Example of use:

```
bldc -p 0 cw
```

4.3.5 ccw command

This command infinitely turns the motor toward CCW direction until stop command is executed or rotor position is equal to anchor value. Example of use:

```
bldc -p 0 ccw
```

4.3.6 stop command

This command immediately stops the motor. A sudden electrical commutation stop will not guarantee that the rotor position remains accurate due to the inertia while it was turning. Use this command as necessary.

We suggest you (whenever possible) to substitute it by combining anchor_set to *cw* or *ccw* commands; *step_cw* or *step_ccw* which include built-in braking algorithm. Example of use:

```
bldc -p 0 stop
```

5 Schematic

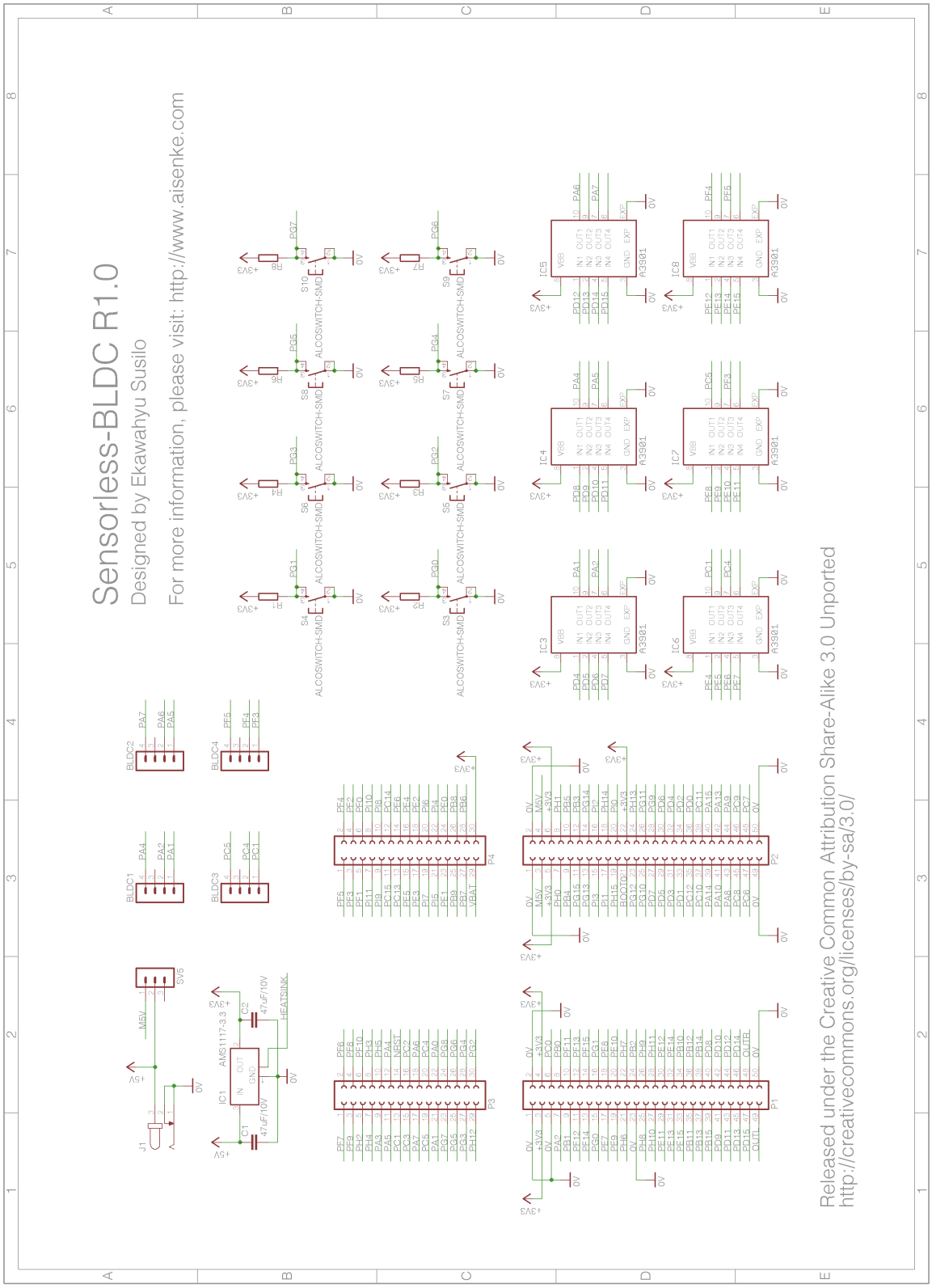


Illustration 20: BLDC-MEB Schematic

6 Revision History

Date	Revision	Changes
29 Jan 2013	1	Initial document release

Table 1: Revision history