



## **UART Guide for the ZDBL15UART Series of Sensorless Brushless DC Motor Drivers (Brushless ESCs)**

## Contents

Introduction	3
Wiring Motor Phase Wires	3
Required connecting cable to PC	4
Software Download	4
Software setup and test	4
Opening the Port	5
Setting up the display	7
Testing the software and FDTI cable	8
Connecting to the ZDBL15UART	9
Programming Manual	10
Connecting the Power	17
Terms of Sale	18

## Introduction

First of all, thank you for your decision to purchase the ZDBL15UART Sensorless Brushless DC (BLDC) Motor Controller. The ZDBL15UART is a powerful sensorless brushless DC (BLDC) motor controller which allows you to directly optimise and control all of the features of the ZDBL15UART via UART control.

The controller is suitable for use with a wide range of brushless DC motors but care must be taken to optimise each particular controller to the specific motor. We ask that you take care when reading this manual and ensure that you follow the instructions carefully.

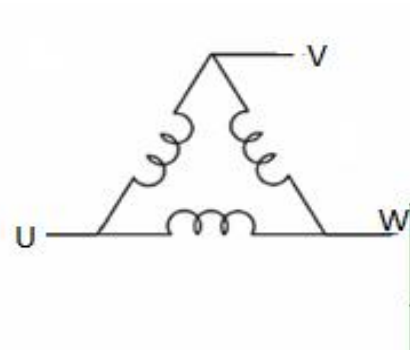
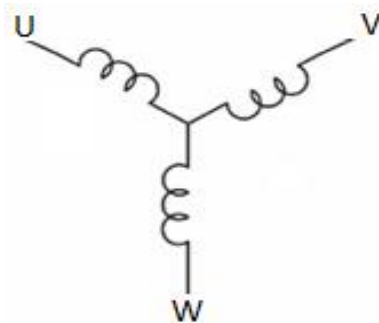
If you have any questions or require assistance with any aspect of the controller setup please contact our support team via [www.zikodrive.com](http://www.zikodrive.com) and they will be happy to assist you.

## Wiring Motor Phase Wires

The motor phase wires sequence on a sensorless BLDC motor is not critical, below is an example.

**Note** – if the orientation of two motor phase wires are switched, the motor direction will be reversed.

Terminal	Wire
U	Yellow
V	Red
W	Black



## Required connecting cable to PC

The Zikodrive ZDBL15UART requires a 0-5V TTL UART signal for the basic UART protocol to work. Any cables that meets this specification is acceptable however Zikodrive Motor Controllers recommend the FTDI TTL-232RG.

The wires labels required are:

GND

RX

TX

The remaining wires (if any) should be taped back and kept out of the way. It is the responsibility of the user to ensure connections between these three wires and the Zikodrive ZDBL15UART box header.

The Drivers for the cable should auto install, if they do not they can be found on the FDTI website (<http://www.ftdichip.com/FTDrivers.htm>).

## Software download

Any terminal software that has the capability of transmitting and receiving serial data in either a binary or a Hexadecimal format is acceptable. However throughout this guide we will be utilising a free software called Realterm. The software download can be found here ([http://sourceforge.net/project/showfiles.php?group\\_id=67297](http://sourceforge.net/project/showfiles.php?group_id=67297)) and additional information on the software can be found here ([http://realterm.sourceforge.net/index.html#downloads\\_Download](http://realterm.sourceforge.net/index.html#downloads_Download)).

## Software set up & Test

Once installation has completed and all the drivers for the FTDI cable have successfully been installed, the Realterm software can be started. The first objective is to ensure your port is open.

Before we connect up to the Zikodrive ZDBL15UART driver, a quick transmit and receive test can be completed by shorting our the RX and the TX pins (yellow and orange on the FTDI TTL-232RG cable).

Click on the "SEND" TAB just under the black screen.

In the top Text box we can now write a test Hexadecimal serial string. For example:

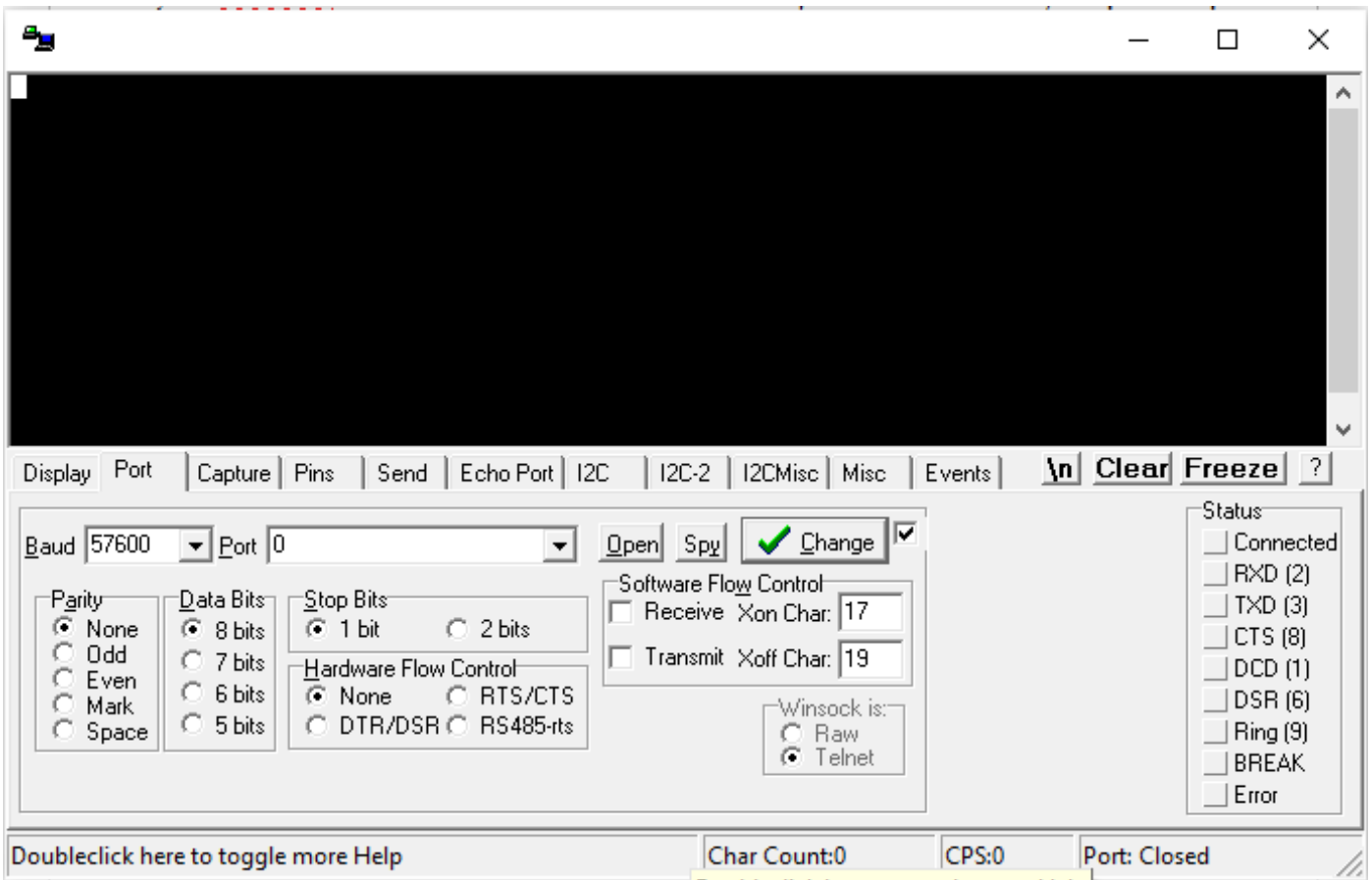
```
0x7A 0x64      //Ascii code for zd//
```

Always ensure a space is left between each byte.

If all the setup is correct and the RX and TX wires are shorted together, once you press the "SEND NUMBERS" button you should see 7A64 displayed on the top right hand corner of the black screen as in the screen shot below:

## Opening the Port

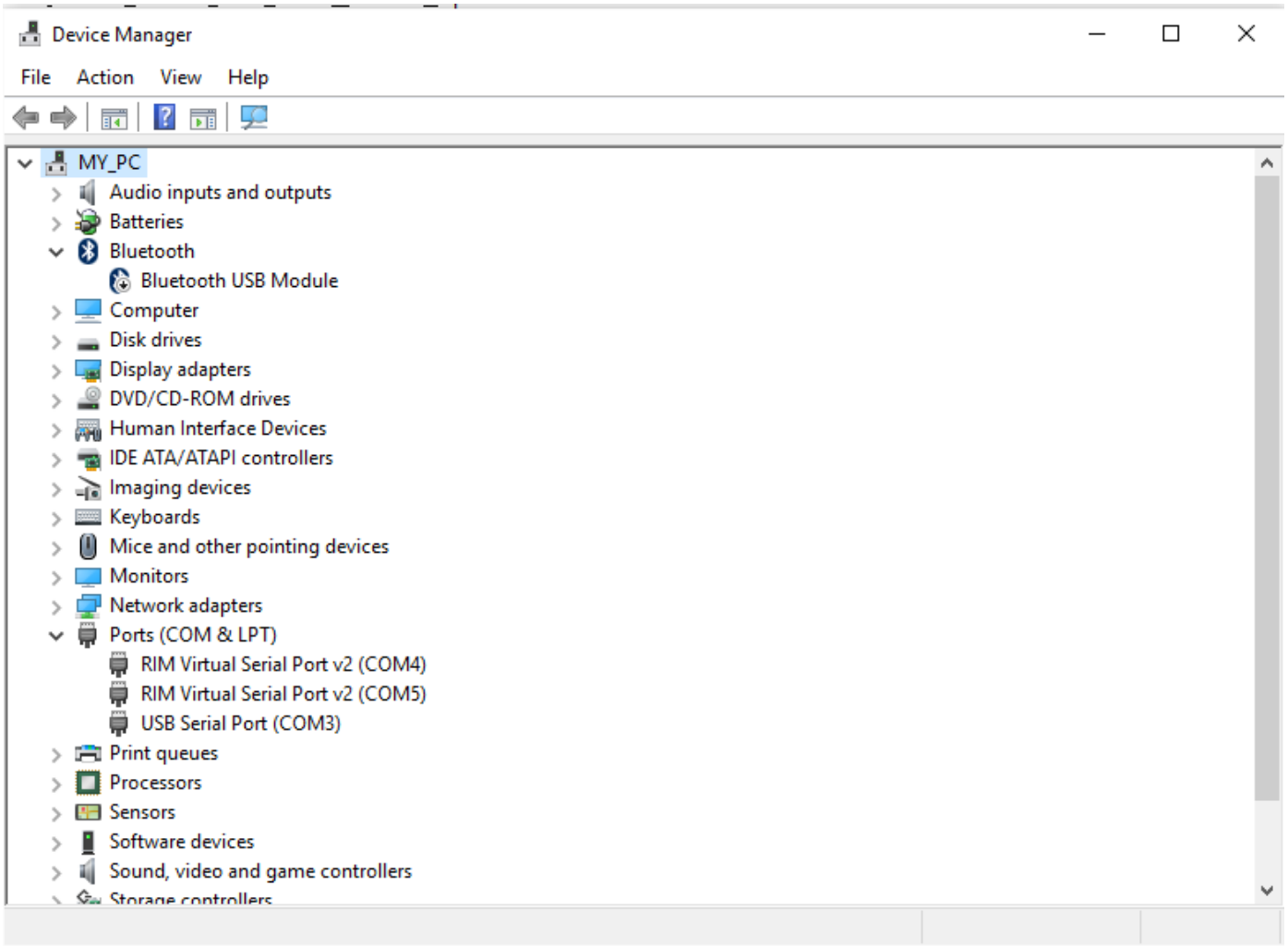
Click the Port TAB just below the black screen, you should see the screenshot shown below:



Please make the following changes:

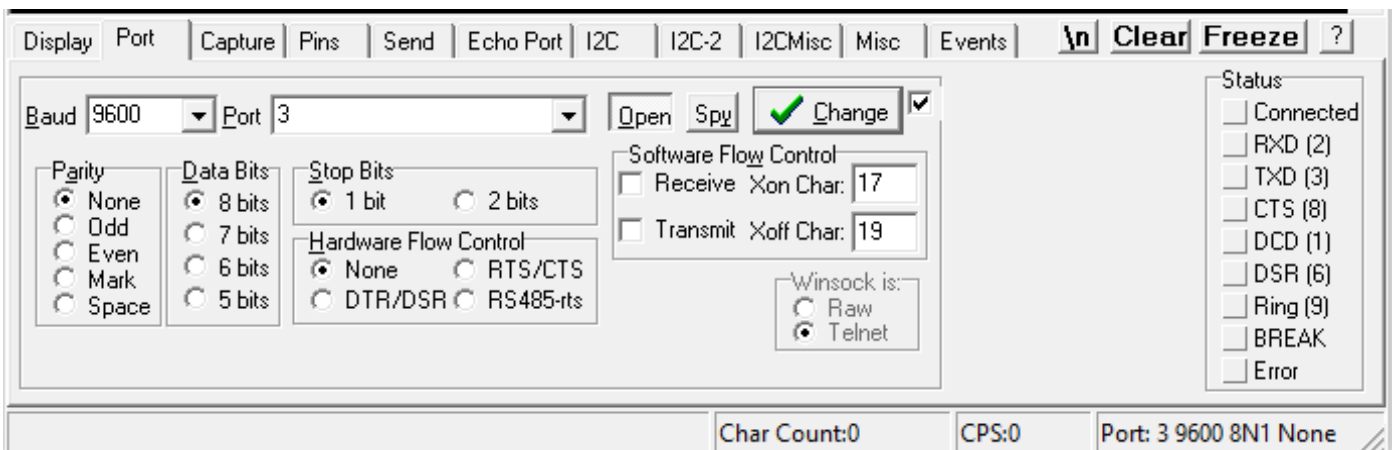
Change the Baud rate to 9600

Select your Port, now this depends on which USB port you plugged your FTDI cable into. If your drivers were successfully installed rescan all the ports. If the Port is still not found you will need to access the Ports via your PC's device manager to find the correct one. You can see on the screenshot below that this one is connected to COM3.



Leave the other options as default (Parity – None; Data bits – 8; Stop bits – 1; Hardware flow control – None; and leave software flow control boxes unchecked).

Once all these options are correct, press the “Open” button to open the com port. Once done you should see a screen like this:



Please note in the bottom right hand corner, the port saying that it is open along with all its parameters “3 9600 \*N1 None”.

## Setting up the display

To ensure that we can read the code returned from the Zikodrive ZDBL15UART driver back to us, we need to ensure that the display is showing the in the correct format. To do this, click on the “DISPLAY” TAB on the far left and make the following changes:

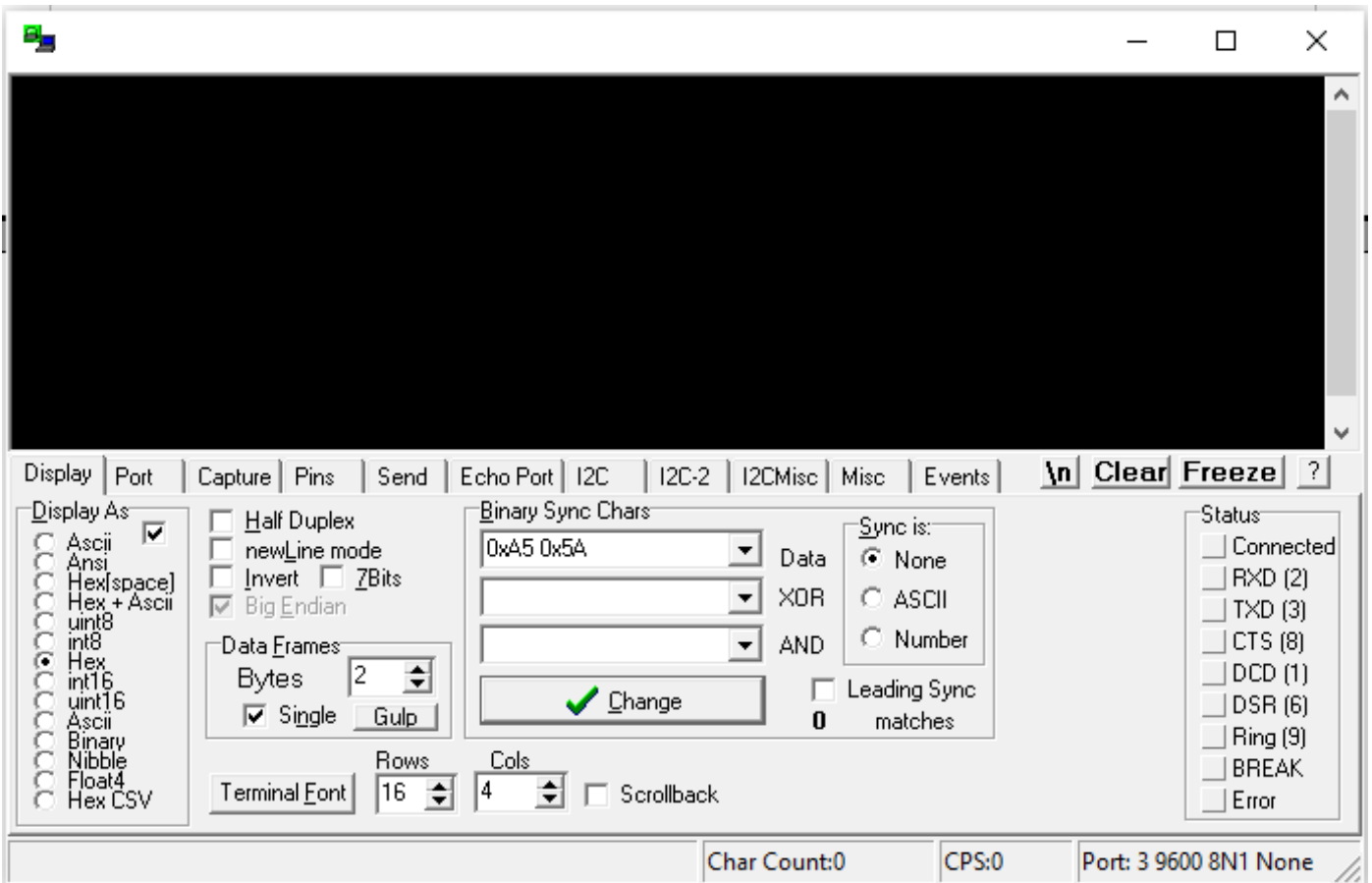
Display as “Hex”

Data Frames Bytes = 2 (it will not let you have one for some reason)

Check the box saying “single”

On the Binary Sync Charts the drop down menu choose the “0xA5 0x5A” option.

Your screen should now look like this:



Once the port ins Open and all the settings have been updated, we can move onto testing the software and FTDI cable.

## Testing the software and FDTI cable

Before we connect up to the Zikodrive ZDBL15UART driver, a quick transmit and receive test can be completed by shorting our the RX and the TX pins (yellow and orange on the FTDI TTL-232RG cable).

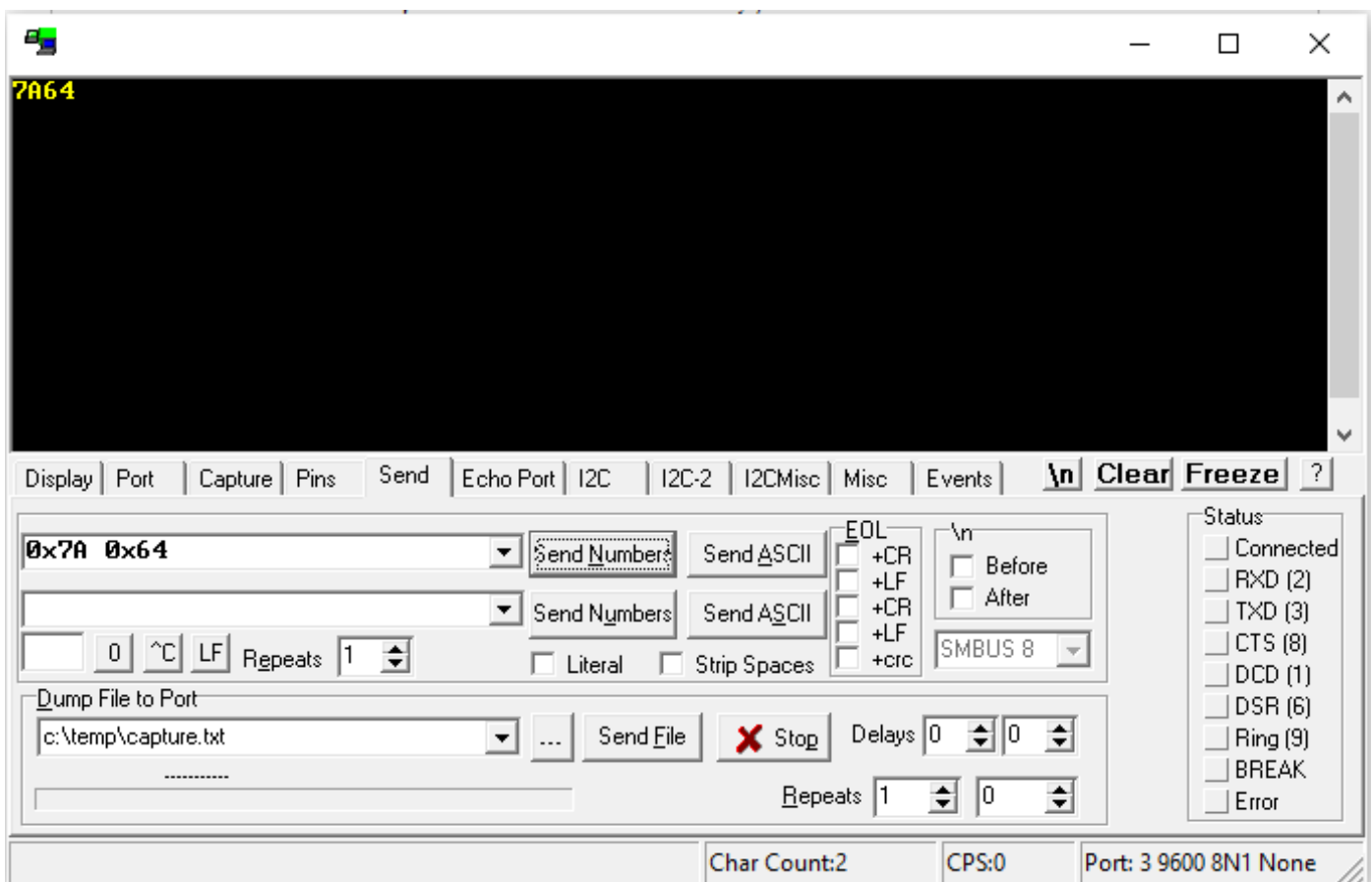
Click on the "SEND" TAB just under the black screen.

In the top Text box we can now write a test Hexadecimal serial string. For example:

```
0x7A 0x64 //Ascii code for zd//
```

Always ensure a space is left between each byte.

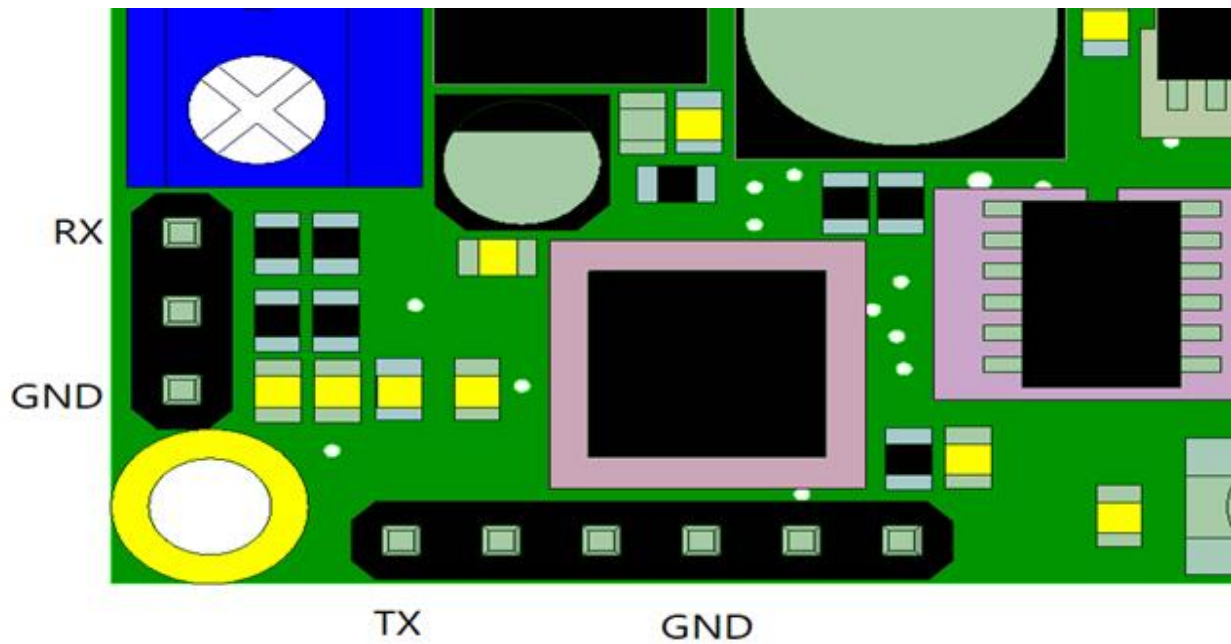
If all the setup is correct and the RX and TX wires are shorted together, once you press the "SEND NUMBERS" button you should see 7A64 displayed on the top right hand corner of the black screen as in the screen shot below:





## Connecting to the ZDBL15UART

The UART setup uses signal voltages up to 5V through the RX and TX lines on pins shown above, two GND pins are also provided (however only 1 is required)



Baud Rate	9600
Parity	None
Start Bits	One
Stop Bits	One

In order to restrict any data corruption a fixed coded header (0x7A, 0x64 ASCII codes zd) is used to commence all communication with all ZikoDrive devices.

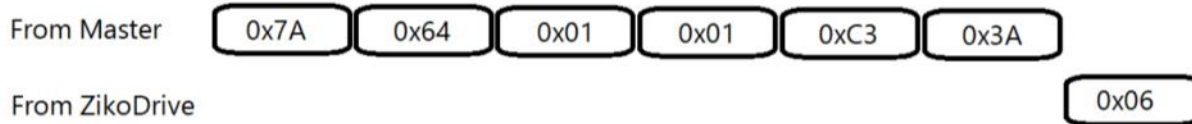
## Programming Manual

This manual pre-supposes that you have some experience in this type of programming. For those who do not have this please visit [www.zikodrive.com](http://www.zikodrive.com) and contact our team who will be happy to help.

Device Address	Register	Register Name	Notes	DATA Size (bits)	Default
0xXX	0x01	Speed	Speed data used for setting motor running speed	8	0xFD
	0x02	R_Current	Change Running Current	8	0x03
	0x03	FB_DIR	Feedback on/off & change direction	8	0x01
	0x04	Setup_1	Setup Configuration 1 (Proportional Gain, Starting torque & Start Speed)	2x8	0x9718
	0x05	Setup_2	Setup Configuration 2 (Integral Gain, Max FD Speed & Phase Advance)	2x8	0xB6C2
	0xAA	Read_All	Read all Programmed Parameters from Ziko-Drive	7x8	NA
	0xAB	Write_All	Write all Parameters to the ZikoDrive	7x8	0x03FD0197 18B6C2

**DATA PACKETS:** The data is processed in 8bit bytes MSB first. Once a full set of bytes has been received for the required register, the device will then send a acknowledgement byte back to confirm command and including any errors (eg, incorrect checksum).

## Example of typical data packet



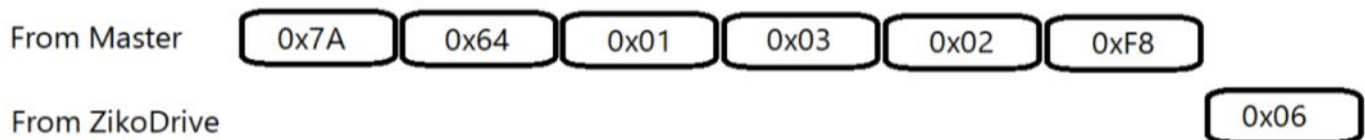
From Master

0X7A & 0x64 = Ascii Letters zd header to commence communication  
 0x01 = ZikoDrive Device Address  
 0x01 = Required write register (Change Speed)  
 0xC3 = Write new speed  
 0x3A = CheckSum

From ZikoDrive

0x06 = Confirmation Speed set OK

## Example 2 of typical data packet



From Master

0X7A & 0x64 = Ascii Letters zd header to commence communication  
 0x01 = ZikoDrive Device Address  
 0x03 = Required write register (Change Direction & Feedback)  
 0x02 = Change Direction to CCW, Turn Feedback off  
 0xF8 = CheckSum

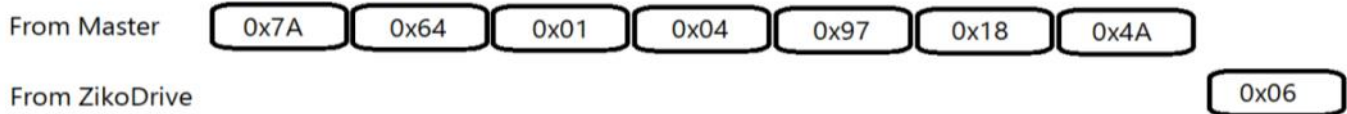
From ZikoDrive

0x06 = Confirm Speed & Direction set OK

## Programming Guide

Address	The addresses are used when multiple devices are used on the same network (typically used on RS485 hardware). All devices have a default 0x01 programmed when dispatched from the factory. This address can be changed using the ZikoDrive programmer board (max Address 0xFF).		
Speed (0x01)	Sets the speed of the motor as an 8bit byte that can be seen as a percentage of the motor speed. For Example, if a HEX value of 0xC3 (decimal 195) the speed set will be $(195/255)*100 = 76.5\%$ .		
	If the ZikoDrive controller is operation in Open Loop with no feedback set, the actual motor speed will vary due to load exerted on the motor.		
	It must be noted that due to the sensorless drive configuration the starting speed must exceed a minimum value. This minimum value changes depending on different motor parameters such as number of motor poles, resistance and inductance. If a lower speed is required it is suggested to start the motor at a higher speed and then decrease the speed once the startup sequence has been completed.		
R_Current (0x02)	The R_Current or running current value limits the maximum current available to the motor during normal running conditions. This value should be ideally set as close as possible to the motor Nominal Current (Max continuous current) of the motor. Making this value too low will restrict the motor torque, however will reduce heat. Setting this value too high could risk overheating the motor and the ZikoDrive Controller.		
FB_DIR (0x03)	FB_DIR register allows you to change the direction of the motor and switch the constant speed feedback on or off. This setting can be changed while the motor is running, however it is advised to set the motor speed to "0" before changing this register.		
	FB_DIR	Notes	
	0x00	Motor Drives Clockwise* with all feedback turned off	
	0x01	Motor Drives Clockwise* with PI feedback turned on	
	0x02	Motor Drives Counter clockwise* with feedback turned off.	
	0x03	Motor Drives Counter clockwise* with PI Feedback turned on.	
	* The direction of the motor may also be determined by the wiring of the motor phased to the Controller. This may change from different motor manufacturers		

Setup_1 (0x04)	Setup_1 register contains the following sub registers:				
	Header [15:12] (0x9) required to start write				
	Proportional Gain control [11:8] for when PI feedback is switched on				
	Starting Torque [7:4] to control the current during starting up forced commutation				
	Start Speed [3:0] for setting the speed during forced commutation				
	<b>Register Position</b>	<b>Register Name</b>	<b>Notes</b>	<b>DATA Size (bits)</b>	<b>Default</b>
	[15:12]	Header	Required to write to this register	4	0x9
	[11:8]	Prop_Gain	Proportional gain control for PI feedback loop	4	0x7
	[7:4]	Start_Torque	Amount of current used to align and force commutate the motor	4	0x1
	[3:0]	Start_Speed	Forced commutation transition speed	4	0x8

**EXAMPLE**

From Master

0x7A & 0x64 = Ascii Letters zd header to commence communication  
 0x01 = ZikoDrive Device Address  
 0x04 = Required write register (Change Setup\_1)  
 0x97 = Set Proportional multiplier  
 0x18 = Set Start Torque and Start Speed  
 0x4A = CheckSum

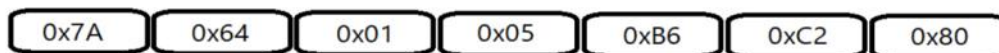
From ZikoDrive

0x06 = Confirm Setup\_1 set OK

Header [15:12]	Always required to write to this register				
Prop_Gain [11:8]	Proportional multiplier part of the PI feedback loop. This value is only used with the closed loop feedback in register FB_DIR (0x03) is set to either 0x01 or 0x03. The proportional gain multiplier essentially multiplies the error (required speed set by speed register (0x01) – Actual motor speed). I.e, if this value is high very large changes in the speed will occur getting smaller and smaller as they approach the target speed, this will obtain the requires speed much faster, however may result in instability and oscillation around the target speed. If this value is set low, the system will be much more stable however as the error speed increments are much smaller to begin with it takes longer to obtain the target speed.				
Start_Torque [7:4]	As the BL15 motor driver utilises sensorless drive commutation, in order for the driver to align the motor poles to ensure a smooth startup the BL15 sets a PWM voltage between two motor phases. For a smooth start up this value needs to be tuned based on motor parameters (rotor inertia) and shaft load.				
Start Speed [3:0]	Once the motor has aligned itself, the BL15 will start to step the motor around (much like a stepper motor) which is called the forced commutation sequence. A sensorless driver required the back EMF from the motor to be read, the faster the motor is rotation the larger the back EMF. The Start speed sets the point at which the motor will transition between the forced commutation mode and sensorless commutation. Values too low will not allow the motor to start (the Bemf can not be sufficiently read) and values too high may result is an unstable start up and even loss in synchronisation (stall)				
Setup_2 (0x05)	Setup_2 register contains the following sub registers:				
	Header [15:12] (0xB) required to start write				
	Integral Gain control [11:8] for when PI feedback is switched on				
	Max_Speed [6:4] used when PI Feedback is switched on				
	Phase_Adv [3:0] for setting up commutation angle				
	<b>Register Position</b>	<b>Register Name</b>	<b>Notes</b>	<b>DATA Size (bits)</b>	<b>Default</b>
	[15:12]	Header	Required to write to this register	4	0xB
	[11:8]	Int_Gain	Integral gain control for PI feedback loop	4	0x6
	[6:4]	Max_Speed	top threshold to which the PI feedback accepts the Max Speed	3	0xC
	[3:0]	Phase_Adv	Forced commutation transition speed	4	0x2
	Note, [7] is not stated in the above settings.				

**EXAMPLE**

From Master



From ZikoDrive

0x06

From Master

0X7A & 0x64 = Ascii Letters zd header to commence communication  
 0x01 = ZikoDrive Device Address  
 0x05 = Required write register (Change Setup\_2)  
 0xB6 = Set Integral Term  
 0xC2 = Set Max Speed and Phase Advance  
 0x80 = CheckSum

From ZikoDrive

0x06 = Confirm Setup\_2 Programr

Header [15:12]	Always required to write to this register		
Int_Gain [11:8]	Integral multiplier part of the PI feedback loop. This value is only used with the closed loop feedback in register FB_DIR (0x03) is set to either 0x01 or 0x03. The Integral gain multiplier essentially multiplies the error (required speed set by speed register (0x01) – Actual motor speed) along with the time (duration) of the error. I.e, if this value is high very large changes in the speed will occur getting smaller and smaller as they approach the target speed, this will obtain the requires speed much faster, however may result in instability and oscillation around the target speed. If this value is set low, the system will be much more stable however as the error speed increments are much smaller to begin with it takes longer to obtain the target speed.		
Max_Speed [6:4]	This value needs to be determined from the motor parameters ie, number of pole pairs and no load speed. Below is a formula to help calculate the required Value:		
		$F_{el} = (2^{(8+n)} - 1) * 0.1$	
		$RPM = (F_{el} * 60)/PP$	
	Where:		
	F <sub>el</sub> =	Electrical Frequency	
	n =	Value entered into register	
	RPM =	Speed in revolution per minute	
	PP =	Number of pole pairs	
	The max speed must be set as close as possible, yet slightly higher than the no load speed of the motor.		
Phase_Adv [3:0]	This value is adjusted if the electrical commutation of the motor is deemed to be out of phase with the magnetic field of the rotor. Signs of this error may be a drop in expected motor torque.		
Read_All (0xAA)	Register is used to read all the settings pre-stored in the driver.		

### EXAMPLE



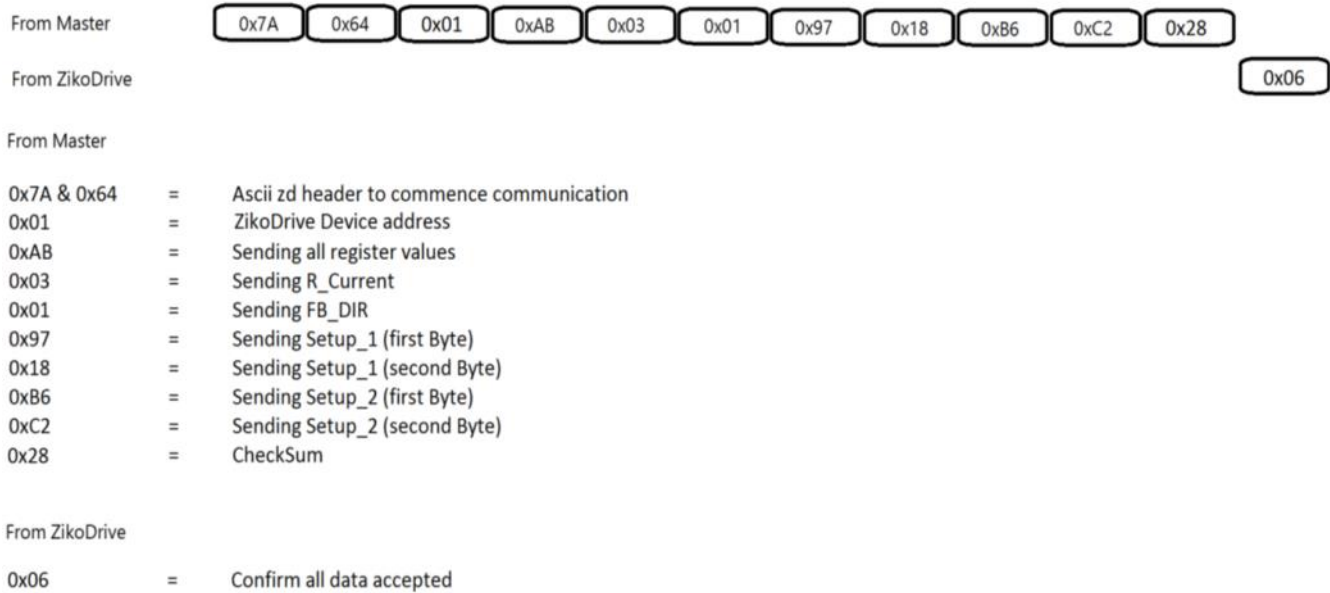
From Master

0x7A & 0x64 = Ascii Letters zd header to commence communication  
 0x01 = ZikoDrive Device Address  
 0xAA = Read all Registers  
 0x52 = CheckSum

From ZikoDrive

0x7A & 0x64 = Ascii zd header to commence communication  
 0x16 = Default Master address for return communication  
 0xAB = Sending all register values  
 0x03 = Sending R\_Current  
 0x01 = Sending FB\_DIR  
 0x97 = Sending Setup\_1 (first Byte)  
 0x18 = Sending Setup\_1 (second Byte)  
 0xB6 = Sending Setup\_2 (first Byte)  
 0xC2 = Sending Setup\_2 (second Byte)  
 0x12 = CheckSum

Write\_All (0xAB) - Register is used to Write all the settings into the ZikoDrive in one go (used mainly for OEM's)

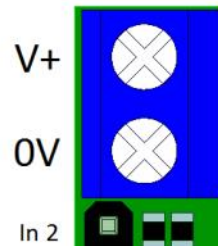


Checksum Formula	Checksum is used on all communications to the ZikoDrive controller to ensure no data is missing or corrupted. Below is a simple formulae used for the checksum
	Checksum = ((DATA0 + DATA1 + DATA2 + DATA3) XOR 0xFF) & 0xFE
Unacceptable data	If the data is rejected, a failure command of 0x15 is sent to the Master instead of the default 0x06.



## Connecting the Power

8-30Vdc must be applied to the board next to the motor phase wires on the 2 way terminal Block. The Ground position is marked with 0V and the DC voltage marked with a +V.



Once all these connections are complete, the Motor can be powered up.

## Terms of Sale

All Zikodrive Motor Controllers are designed to be a component incorporated within equipment manufactured by our customers and are not suitable for use by an end user. As such, none of our motor controllers are CE marked. It is entirely the buyer's responsibility to ensure that all Zikodrive Motor Controllers and other related products meet the required specification and safety requirements for applications in which they are used.

The use of Zikodrive products in safety critical applications is entirely at the buyers risk, and the buyer agrees to defend, indemnify and hold harmless Round Bank Engineering Ltd from any and all damages, claims, suits, or expenses resulting from such use. Round Bank Engineering Ltd. is not responsible for injury or damage of any kind, including but not limited to, injury, death, damage, property damage/loss or any other type of loss which may arise in whole or in part from the use of Zikodrive Motor Controllers.

## **RETURNS**

Subject to the terms provided herein, all returns for exchange, refund or credit must be made within fourteen (14) days from the date of delivery. All questions or comments relating to returns should be sent directly to [enquiries@zikodrive.com](mailto:enquiries@zikodrive.com) or by calling us direct on +44 (0) 333 123 7130.

In order to qualify for an exchange, refund or credit, the product must be in re-sellable condition, which shall be determined at Zikodrive Motor Controller's sole discretion. Factors affecting the re-sellable condition of a controller/motor/accessory include, but are not limited to: obvious signs of use or abuse; customer negligence; excessive wear and tear and/or damaged/missing product or parts.

Where it is determined, at Zikodrive Motor Controller's sole discretion, that the controller/accessory is, in fact, re-sellable, Customer will be given a full refund or credit for the product, less the cost of shipping. Customer should expect a refund within 30 business days of Zikodrive Motor Controller's receipt of the controller/accessory.

CUSTOM PARTS AND PRODUCTS DO NOT QUALIFY FOR REFUND OR CREDIT.

## **WARRANTY**

Zikodrive Motor Controllers warrants to the original purchaser that any part of its controller/accessory purchased will be free of defects in workmanship and parts for a period of twelve (12) months from the date of delivery (hereinafter "Warranty Period"). During the Warranty Period, Zikodrive Motor Controllers will, at its option: (1) provide replacement parts necessary to repair the product; (2) replace the product with a comparable product; or (3) refund the amount Customer paid for the product upon its return.

Replacement parts or products will be new or serviceably used, comparable in function and performance to the original part or product, and warranted for the longer of thirty days for the US or the remainder of the warranty period. Any additional purchases or upgrades will not extend this warranty. This product warranty covers normal use only.

This product warranty does not cover damage caused during shipment and any damage caused by: actions that are beyond Zikodrive Motor Controller's control, including (but not limited to) impacts, fluids, fire, flood, wind, earthquake, lightning or similar disaster, war, lockout, epidemic, destruction of production facilities, riot, insurrection, or material unavailability; unauthorized modifications, attachments or peripherals; improper use, environment, installation or electrical supply; improper maintenance; any other misuse, abuse or mishandling.

EXCEPT FOR THE WARRANTIES EXPRESSED IN THIS AGREEMENT, ZIKODRIVE MOTOR CONTROLLERS DISCLAIMS ALL OTHER WARRANTIES, EITHER EXPRESS OR IMPLIED, INCLUDING IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE, OTHER THAN THOSE WARRANTIES IMPLIED BY AND INCAPABLE OF EXCLUSION, RESTRICTION OR MODIFICATION UNDER THE APPLICABLE LAW. THE TERM OF ANY IMPLIED WARRANTIES THAT CANNOT BE DISCLAIMED ARE LIMITED TO THE TERM OF THIS AGREEMENT.