# Zikodrive ZDBL Series UART Operating Manual V1.1

## For use with firmware versions 1.4 and above (Updated 8/10/19)

Zikodrive Motor Controllers is a trading name of Round Bank Engineering Ltd
Unit 4B Top Land Business Park, Hebden Bridge, West Yorkshire, HX7 5RW
www.zikodrive.com - +44 (0) 333 1237130

# Contents

# Introduction

## *Overview*

This manual is intended to be used with the Zikodrive ZDBL SERIES UART compatible Motor controllers running standard firmware (1.4) and includes the following UART Commands that can also be used over RS232 or RS485 using the relevant Break Out Boards:

- Setting up standard running features such as Speeds, Output Enable, Direction, braking and running current.
- Setting up the start-up parameters including Hold torque, Hold time, Starting torque and the starting speed.
- Setting up the motor running parameters such as the Phase Advance Angle, Rotor Positional Proportional Gain, Rotor Positional Integral Gain, PWM frequency, etc.…
- PI Closed-loop control settings
- Additional Mode settings

## *Specifications*

- Up to 35Vdc.
- Up to 20A.
- Sensorless BLDC trapezoidal/Block commutation control
- Modular design connects to many Break Out Boards that can provide options such as 420mA, 0-10Vdc, Optically Isolated inputs, etc...
- Over-temperature protection and stall detection.
- 16bit high-speed Microcontroller controlled.

## Important NOTE:

It is critically important to not exceed the recommended input voltages as this may cause unrepairable damage to the Zikodrive ZD SERIES UART Motor Driver. The max voltage recommended for any digital or analogue inputs is 5Vdc. DO NOT APPLY Higher Voltage PWM as this will damage the board.

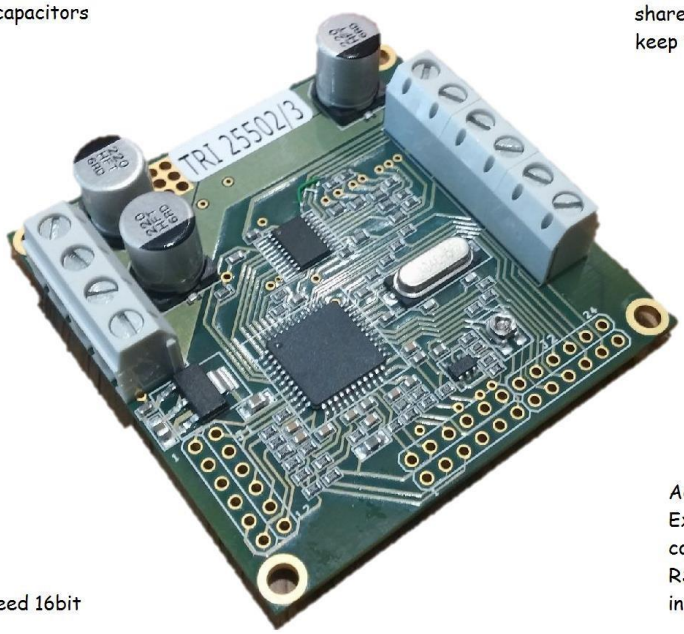
All wires must be kept as short as possible to reduce EMC emissions from the motor phase cables and to reduce noise on both the analogue and digital inputs. If you require more information regarding good wiring practice please contact Zikodrive enquiries@Zikodrive.com.

# *Components*

## Board View

Large Very low ESR capacitors

Connections doubled up to evenly share high current loads, as well as keep the total board hight Very low.




Additional IO that can support External break out boards or connectors. Options include RS232/RS485 CanBUS, Isolated inputs etc...

Powerfull high speed 16bit microcontroller

Thick, heavy copper traces to reduce power loss and improve heat dissapation.

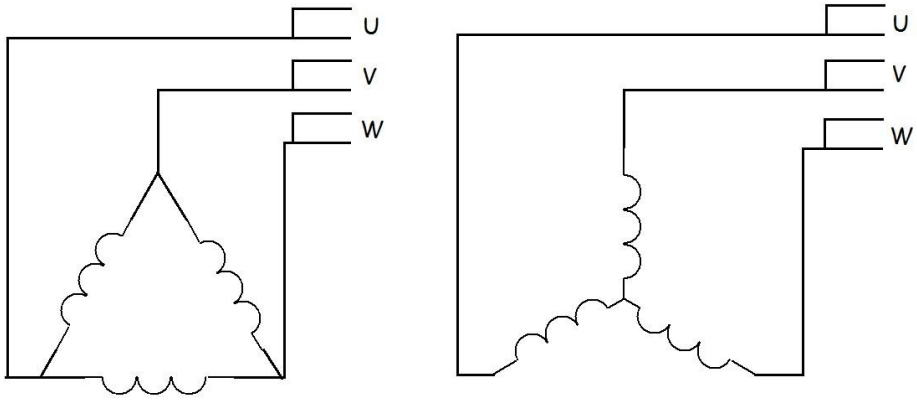
Reverse Poilarity Protected


Power Mosfets mounted to the bottom for easy heat sinking and heat dissapation.

# Setting Up

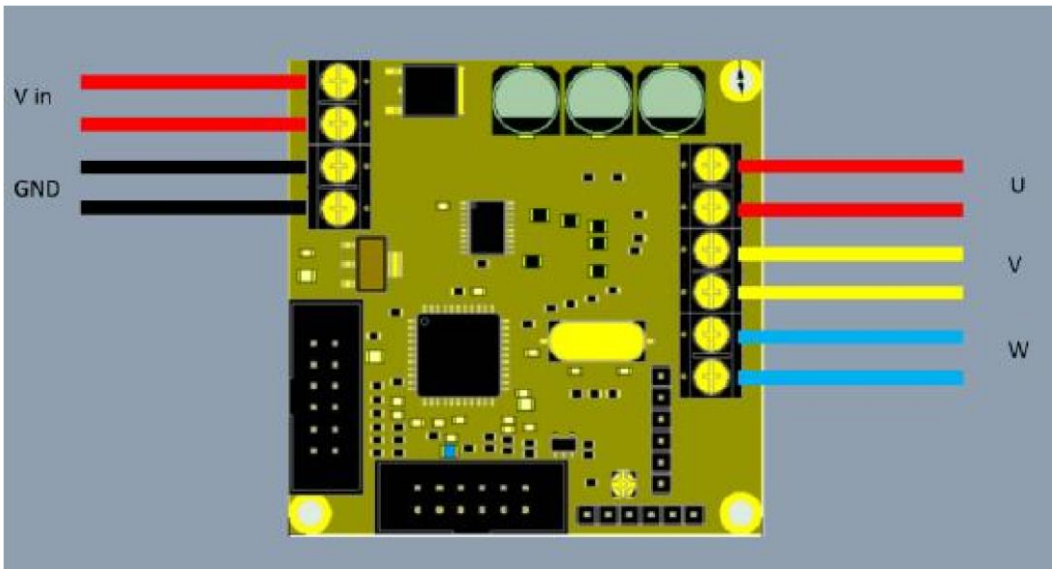
Wiring Motor Phase Wires

Wire up the Motor phase wiring according to the motor manufacturers datasheet. The exact order the wires are inserted is not critically important. If the wiring order is incorrect the motor direction will change.

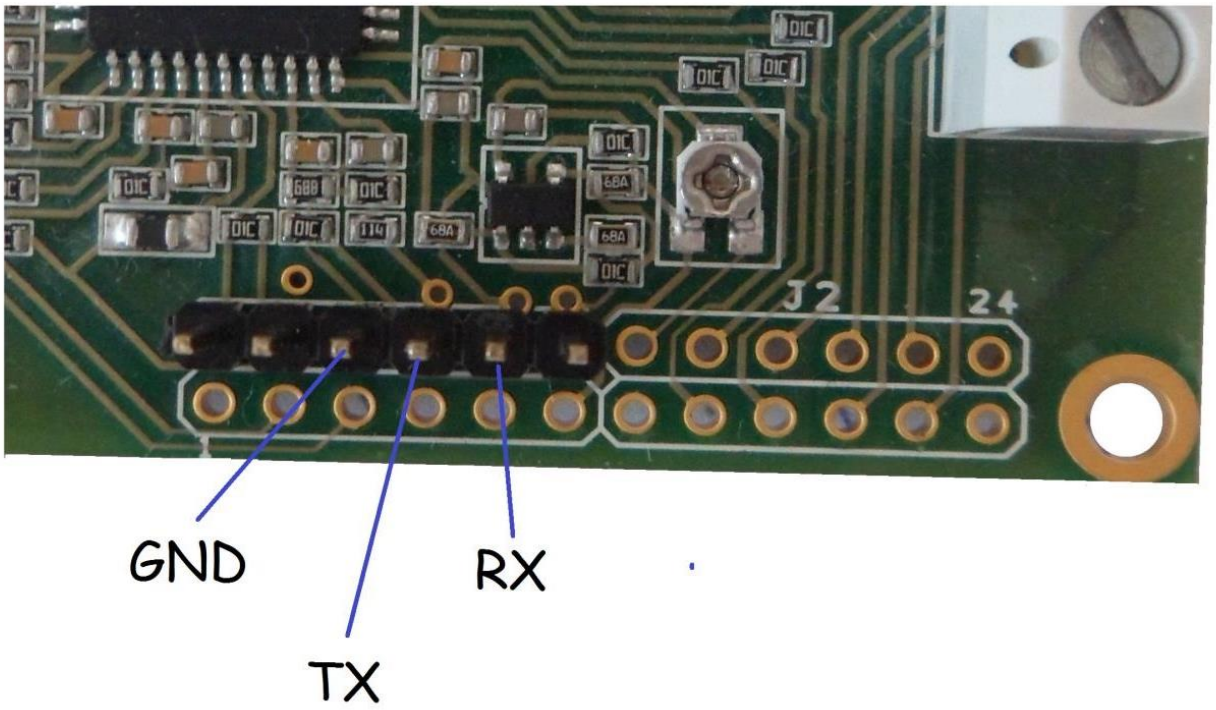



Power Wires

12-35Vdc must be applied to the board on at least two terminals of the terminal Block Marked GND & Vin (on the rear of the board). Although the board is protected against reverse polarity, care should be taken to ensure the wires are not reversed.
Note – Please ensure that the DC power supply has the proper current supply capability. Care should be taken to ensure the DC supply is clean (keep wires as short as possible, and shield where required).



UART Connections




GND is the common digital GND for the PCB. The RX and the TX pins must be connected to the master Controller. I.e., TX (transmitting) from the Master controller must be connected to the RX (receiving) pin on the ZDBL board. The TX (transmitting) from the ZD SERIES UART must be connected to the RX (receiving) of the Master Controller.

Connections for any serial Break Out Board (RS485 & RS232)

Connecting to the Break Out Boards are simple.

| | | |
|---|---|---|
| A | = | Non-inverting receiver input and non-inverting driver output |
| B | = | Inverting receiver input and inverting driver output |
| GND | = | Ground |

Note - If shielded cables are used between the Master Controller and the Zikodrive please ensure only one end of the shielding is grounded to ensure no Ground loops.

# Programming Manual

## *The UART Protocol*

The protocol used is non-standard and derived by the Zikodrive engineers to provide fast and efficient data transfer.
The Data is transmitted and received in 8bit binary Bytes. To easily transition between binary, decimal, and hexadecimal you can either use the programmer mode on the PC calculator.

Please note we use an 8bit UART protocol. This means that to detect data on the UART the system checks to see if the data byte is < 0xFF. If the data on any bytes within the command equal 0xFF the whole command will not be identified!

## *Protocol Setup*

| | | |
|---|---|---|
| BAUD RATE | = | 19200 |
| START BITS | = | One |
| STOP BITS | = | One |
| PARITY | = | None |

## *Recommended hardware*

TTL-232R-3V3

## *Recommended Software*

Realterm Serial/TCP Terminal that can be *downloaded here* from SourceForge
There is ongoing development for Zikodrive's own software ZDSoft to incorporate the ZDBL series of controllers in the near future.

## *The Zikodrive Protocol*

All commands are initiated with a two-byte header, one-byte address code, one-byte register address, a data byte and then followed with a CheckSum byte.
To ensure the correct data is being transferred, the protocol will send either a confirm (0x06 followed by the accepted Data) or a reject (0x15 followed by the anticipated checksum value) for a bad command. If the ZD SERIES UART has accepted the command, the confirmation will then be sent to the Master Controller and then process the given command. If the command was rejected, all data will be disregarded and a new command must be sent.

Example 1 – Basic Send

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| From Master | 0x7A | 0x64 | 0x01 | 0x02 | 0x00 | 0xFC | |
| From Zikodrive | | | | | | | 0x60 0x00 |

| | | |
|---|---|---|
| 0x7A & 0x64 | = | Header, ASCII code ZD |
| 0x01 | = | Zikodrive Address, Set as 0x01 as default |
| 0x02 | = | Register Address, Enable outputs |
| 0xFC | = | CheckSum |

## *CheckSum formula*

Each command sent to the Zikodrive controller must be followed with a CheckSum byte or the data will be rejected.

CHECKSUM = ((ADDRESS + REGISTER + DATA) XOR 0xFF) & 0xFF;

1. The Address, the required register, and the data bytes are added together.
2. A command is run through an XOR with 0xFF
3. A command is then run through & with 0xFF

We have available an EXCEL spreadsheet to help calculate the CheckSum byte if required.
Please note that the Header and the CheckSum bytes are not included in the CheckSum Formula.

## *Shortlist of Commands*

| Register | Name | Length | Default |
|---|---|---|---|
| RUN | | | |
| 0x01 | Control | 8bits | 0x00 |
| 0x02 | Enable Outputs | 1bit | 0x1 |
| 0x03 | Direction | 1bit | 0x0 |
| 0x04 | Brake | 1bit | 0x0 |
| 0x05 | Running Current | 4bits | 0x3 |
| STARTUP | | | |
| 0x10 | Hold torque | 4bits | 0x2 |
| 0x11 | Hold Time | 4bits | 0x2 |
| 0x12 | Start Torque | 4bits | 0x2 |
| 0x13 | Start Speed | 4bits | 0x8 |
| Motor Run | | | |
| 0x20 | Phase Advance | 4bits | 0x8 |
| 0x21 | POS Prop GAIN | 4bits | 0x7 |
| 0x22 | POS Intg GAIN | 4bits | 0x8 |
| 0x23 | Recirculation | 2bits | 0x0 |
| 0x24 | Blank Time | 4Bits | 0xC |
| 0x25 | Decay | 1bit | 0x0 |
| 0x26 | PWM Frequency | 5bits | 0x13 |
| Closed Loop | | | |
| 0x30 | Speed Prop Gain | 4bits | 0x7 |
| 0x31 | Speed Intg Gain | 4bits | 0x6 |
| 0x32 | Max Speed | 3bits | 0x4 |
| 0x33 | Overspeed Limit | 2bits | 0x2 |
| SETTINGS | | | |
| 0x40 | Mode | 2bits | 0x0 |
| 0x41 | Restart(loss) | 1bit | 0x1 |
| 0x42 | Stop on Fail | 1bit | 0x0 |
| 0x43 | Control | 2bit | 0x0 |

## *RUN Registers*

These registers are used to control the motor in real-time or let the controller only how to control the motor once UART has been disconnected.

Control Register (0x01)

The control is used to control the motor. The function of this control register depends on which mode the driver is set to run in. More information on the various modes can be found in the MODE register of this manual.

When the UART is not connected, either an external potentiometer/analogue voltage/current loop/PWM, etc. can be used as the control signal.

Example 2 - Control

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| From Master | 0x7A | 0x64 | 0x01 | 0x01 | 0x80 | 0x7D | | |
| From Zikodrive | | | | | | | 0x06 | 0x80 |

0x7A & 0x64 = Header, ASCII code ZD

0x01 = Zikodrive Address, Set as 0x01 as default

0x01 = Register Address, Control register

0x80 = Data

0x7D = CheckSum

## Enable Outputs (0x02)

If this value is set to high, all the output transistors will be disabled. This register is only a single bit large so it can only be set as a “1” run as normal or a “0” disable outputs. This register can also be accessed through the external IO of the Driver.

Example 3 - Enable outputs

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| From Master | 0x7A | 0x64 | 0x01 | 0x02 | 0x00 | 0xFC | | |
| From Zikodrive | | | | | | | 0x06 | 0x00 |

0x7A & 0x64 = Header, ASCII code ZD

0x01 = Zikodrive Address, Set as 0x01 as default

0x02 = Register Address, Enable Register

0x00 = Ensure Outputs enabled

0xFC = CheckSum

## Change Direction (0x03)

The direction register is used to set the direction of the motor. As the direction is also changed with the wiring of the three motor phases the direction bit will rotate the motor in the opposite direction.

CAUTION – Please ensure the motor is stopped before changing the direction. If the motor is not fully stopped it may possible to damage the driver due to induced reverse voltages.

Example 4 - Change Direction

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| From Master | 0x7A | 0x64 | 0x01 | 0x03 | 0x01 | 0xFA | | |
| From Zikodrive | | | | | | | 0x06 | 0x01 |

Brake (0x04)

When the Brake bit is enabled, the motor's phases will be shorted together when a "0" control signal is provided. If the brake is disabled the motor will freewheel to a stop.

Example 5 - Brake

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| From Master | 0x7A | 0x64 | 0x01 | 0x04 | 0x01 | 0xF9 | | |
| From Zikodrive | | | | | | | 0x06 | 0x01 |

Running Current (0x05)

The running current is the max current calculated in the motor phase (Not the average total current draw of the motor). The Running current is only used once the motor has entered the full sensorless mode. Each incremented value will increase the phase current by 1.25A.

Motor Phase current = Register Value x 1.25A

Example 6 - Change running current

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| From Master | 0x7A | 0x64 | 0x01 | 0x05 | 0x03 | 0xF9 | | |
| From Zikodrive | | | | | | | 0x06 | 0x03 |

5A max phase current.

## *StartUp Registers*

These registers are used to set the StartUp parameters of the motor. If these values are not set correctly to the motor specifications, the motor may be very unstable during StartUp, or may not start at all.

Firstly, to start a sensorless BLDC motor, the rotor needs to be aligned into a known position. This is done by putting a current through two motor phases for a time long enough for the rotor to align. Once the rotor has been aligned the motor will go into a forced commutation cycle where the driver will accelerate the rotor around until the rotor speed is fast enough that the Back EMF created on the spare motor phase is large enough to be detected by the driver. Only once this speed is reached will the driver exit out of forced commutation mode and enter full BLDC senseless commutation.

Alignment Torque Register (0x10)

This value changes the amount of current in the two motor phases for the rotor alignment. The larger values will allow rotors with higher inertia associated with them to be aligned correctly, however care must be taken on smaller motors as too high currents may damage the motor.

The current is calculated as a percentage of the max current of the motor coils and can be calculated if required.

Motor Phase resistance = 0.7Ohms (Example Motor)
Motor supply voltage = 24Vdc

Max motor phase current = 24/0.7 = 34.3A

The alignment torque has a register 4bits long which gives us 16 possible percentage values (starting with “0”). So, each increment = 34.3/16 = 2.14A (6.25%)

Example 7 - Alignment Torque

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| From Master | 0x7A | 0x64 | 0x01 | 0x10 | 0x05 | 0xE9 | | |
| From Zikodrive | | | | | | | 0x06 | 0x05 |

Hex 0x05 = decimal 5 giving us 5x6.25% = 31.25% = 10.7A

NOTE – this value may seem high, however, the full current may not enter the motor phases due to the motor inductance and the hold time. I.e., V=L*di/dt where V = supply voltage, L = motor inductance, di = change in current and dt = change in time. This formula can be re-arranged to find the max current value.

Alignment Time Register (0x11)

This value changes the amount of time the Hold Torque is applied to the two motor phases for the rotor alignment. The larger values will allow enough time for rotors with higher inertia associated with them to be aligned correctly, however, care must be taken on smaller motors as too high currents for too long time periods may damage the motor.

The Hold time can be set from 0ms and increase in increments of 8ms up to a total of 120ms.

Example 8 - Alignment Time

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| From Master | 0x7A | 0x64 | 0x01 | 0x11 | 0x05 | 0xE8 | | |
| From Zikodrive | | | | | | | 0x06 | 0x05 |

40ms allowed for rotor alignment

Forced Acceleration Torque (0x12)

This value changes the amount of current applied to the two motor phases for motor acceleration. The larger values will allow higher torque which will allow higher inertia loads to be accelerated. Care must be taken on smaller motors as too high currents for too long time periods may damage the motor.

The current is calculated as a percentage of the max current of the motor coils and can be calculated if required.

Motor Phase resistance = 0.7Ohms (Example Motor)
Motor supply voltage = 24Vdc

Max motor phase current = 24/0.7 = 34.3A

The alignment torque has a register 4bits long which gives us 16 possible percentage values (starting with “0”). So, each increment = 34.3/16 = 2.14A (6.25%)

Example 9 - Forced Acceleration Torque

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| From Master | 0x7A | 0x64 | 0x01 | 0x12 | 0x05 | 0xE7 | | |
| From Zikodrive | | | | | | | 0x06 | 0x05 |

Hex 0x05 = decimal 5 giving us 5x6.25% = 31.25% = 10.7A

Start Speed (0x13)

This is the speed (defined electrical cycle frequency) that the controller will accelerate the motor up to before switching to full closed-loop senseless control.

Start Speed (electrical frequency) = (Data + 1) * 2Hz

Start speed RPM = (Electrical frequency * 60) / Motor Pole pairs

Please note – the Min speed of the controller is set to 25% of the start speed.

Example 10 - Start Speed

4 pole motor (2 pole pairs)

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| From Master | 0x7A | 0x64 | 0x01 | 0x13 | 0x05 | 0xE6 | | |
| From Zikodrive | | | | | | | 0x06 | 0x05 |

Hex 0x05 = decimal 5
Start Speed (electrical frequency) = (5 + 1) * 2Hz = 12Hz
Start Speed (RPM) = (12 * 60) / 2 = 360RPM

Min motor speed = 360 * 0.25 = 90RPM

## *Motor Run Registers*

These registers are used to set the normal running parameters of the motor once full closed-loop sensorless BLDC mode has been entered. If these values are not set correctly to the motor specifications, the motor may be very unstable, have reduced torque, or not even work at all.

The driver uses a 6-step program to commutate the motor phases. The timing between each electrical commutation cycle is calculated and determined through several parameters to achieve an optimal dynamic response of the motor.

Phase Advance (0x20)

When the Phase advance is set to 0°, the controller will commutate the motor exactly in the middle of two back EMF positions. This is typically undesirable as by the time the controller reads the signal and reacts to it, the rotor has already progressed. The phase advance must be set at an optimal angle to ensure maximum torque at all speeds required by the application.

Example 11 - Phase Advance

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| From Master | 0x7A | 0x64 | 0x01 | 0x20 | 0x05 | 0xD9 | | |
| From Zikodrive | | | | | | | 0x06 | 0x05 |

Each value (starting from "0") will advance the angle by 1.875°

Position Proportional Gain (0x21)

Within a system where the speed is varying, the position of the commutation will vary as well. The ZDBL range of controllers has an integrated PI control system for a fast-dynamic response to these applications. If the value of the Proportional term on this PI controller is set too high, the controller overcompensates the required commutation position which can result in reduced torque and/or motor stall. If the value is set too low, the controller will be unable to keep up with speed variations and again produce low torque or stall.

The $K_{CP}$ (Positional Proportional Gain term) can be calculated as follows:

$K_{CP} = 2^{(2n-7)}$

This allows for a $K_{CP}$ range between 1/128 up to 256

Example 12 - Position Proportional Gain

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| From Master | 0x7A | 0x64 | 0x01 | 0x21 | 0x05 | 0xD8 | | |
| From Zikodrive | | | | | | | 0x06 | 0x05 |

Position Integral Gain (0x22)

The second part to the PI positional controller for the commutation.

The $K_{CI}$ (Positional Integral Gain term) can be calculated as follows:

$K_{CI} = 2^{(2n-7)}$

This allows for a $K_{CI}$ range between 1/128 up to 256

Example 13 - Position Integral Gain

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| From Master | 0x7A | 0x64 | 0x01 | 0x22 | 0x05 | 0xD7 | | |
| From Zikodrive | | | | | | | 0x06 | 0x05 |

Recirculation option (0x23)

Synchronous rectification is performed by the controller to control the current flows through the three-phase bridge rectifier. the circulation option allows you to choose a high side or a low side recirculation of the current, use both, or switch off completely (operating with fast decay of the current).

Changing the recirculation mode can improve, or worsen Back EMF virtual zero-point crossing reliability depending on the duty cycle and motor supply voltage.

| Value | Mode |
|---|---|
| 0x00 | Auto |
| 0x01 | High Side |
| 0x02 | Low Side |
| 0x03 | Off – No synchronous rectification |

Example 14 - Recirculation

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| From Master | 0x7A | 0x64 | 0x01 | 0x23 | 0x00 | 0xDB | | |
| From Zikodrive | | | | | | | 0x06 | 0x00 |

Blank Time (0x24)

Current spikes are created due to the clamp diodes integrated within the mosfets, as well as additional switching transients can cause false current trips/measurements. To overcome this issue a blank time is used to only measure the current after these transients/spikes have dispersed.

The blank time is increased from “0ns” in increments of 400ns. I.e.:

Blank Time = Data*400ns

Example 15 - Blank Time

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| From Master | 0x7A | 0x64 | 0x01 | 0x24 | 0x0A | 0xD0 | | |
| From Zikodrive | | | | | | | 0x06 | 0x0A |

0x0A = 10
10 * 400ns = 4ms

Fast Decay on start (0x25)

The percentage fast decay of the current which is used during the start. The fast decay applies a reverse voltage to the motor phase winding and may reduce the out of phase Back EMF voltage.

0x00 = 12.5%
0x01 = 25%

Example 16 - Fast Decay

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| From Master | 0x7A | 0x64 | 0x01 | 0x25 | 0x00 | 0xD9 | | |
| From Zikodrive | | | | | | | 0x06 | 0x00 |

Fast Decay set to 12.5%

PWM Frequency (0x26)

This is the PWM frequency used to drive the three-phase power inverter. Faster frequencies allow for a better motor response, higher motor efficiency, lower torque ripple. However, the higher frequency switches the mosfets more often which accounts for higher switching losses making the Mosfets run hotter and less efficient. Typically set around 20Khz so the frequency is out of the audible range (lower frequencies can resonate the motor) and low enough to not cause excessive mosfets inefficiencies. Typically motors with a higher inductance (slower Motors) can be run with lower frequencies, and motors with lower inductance (high-speed motors) will require higher frequencies.

PWM Frequency = 1/ ((Data * 1.6 µs) +20 µs)

Example 17 - PWM frequency

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| From Master | 0x7A | 0x64 | 0x01 | 0x26 | 0x13 | 0xC5 | | |
| From Zikodrive | | | | | | | 0x06 | 0x13 |

0x13 = 19
19*1.6 µs = 30.4 µs
30.4 µs + 20 µs = 50.4 µs
1/50.4 µs = 19.84Khz

## *Closed Loop Speed Control Registers*

These registers set the closed Loop PI Speed control parameters of the motor once full closed-loop sensorless BLDC mode has been entered. Please see The Mode Register is used to select between the type of closed-loop control required. The closed-loop speed control must be entered the Mode Register.

The Speed control system measures the electrical cycle frequency to determine speed control. The control register input is used to set the desired speed for these registers.

Speed Proportional Gain (0x30)

Used to control the Proportional segment ($K_{SP}$) of the PI control loop. I.e., the electrical cycle difference (error) is multiplied by this data value ($K_{SP}$) and added to the integral term on each feedback loop. Higher values will cause very large changes where smaller values cause much more subtle changes which cause the motor to take longer to accelerate/decelerate to the control value.

$K_{SP} = 2_{(n-7)}$

Example 18 - Speed Proportional Gain

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| From Master | 0x7A | 0x64 | 0x01 | 0x30 | 0x08 | 0xC6 | | |
| From Zikodrive | | | | | | | 0x06 | 0x08 |

$K_{SP}$ Value of 2

Speed Integral Gain (0x31)

Used to control the Integral segment ($K_{SI}$) of the PI control loop. I.e., the electrical cycle difference (error) is integrated by this data value ($K_{SI}$) and added to the Proportional term on each feedback loop. Higher values will cause very small changes where higher values cause much more subtle changes which cause the motor to take longer to accelerate/decelerate to the control value.

$K_{SI} = 2_{(n-7)}$

Example 19 - Speed Integral Gain

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| From Master | 0x7A | 0x64 | 0x01 | 0x31 | 0x08 | 0xC5 | | |
| From Zikodrive | | | | | | | 0x06 | 0x08 |

$K_{SI}$ Value of 2
Max Speed (0x32)

The control register input is set to a percentage of this maximum speed (in electrical frequency). To set the Max speed you must know the number of motor poles and the no-load speed of the motor at the motor input voltage. If this value is set too low, the motor will never reach its Max speed, if the value is set too high, control resolution is lost as the motor will run at its max speed through a larger proportion of the control register.

Max speed $_{(\text{electrical frequency})} = 0.1 * (2_{(\text{DATA} +8)} - 1)$ Hz

Max Speed $_{(\text{RPM})}$ = (Electrical frequency * 60) / No. of motor pole pairs

Note – the number of pole pairs can be found on the datasheet of the motor. Alternatively, you can count the number of poles by either looking inside the motor and counting the coils, this value gives you the number of poles so you will need to divide by 2 to get the number of pairs. If this is not possible you can apply a very low voltage through two motor phases, rotate the shaft by hand and count the number of positions the rotor settles in through a full revolution, this is the number of pole pairs.

Example 20 - Max Speed

Motor with 4 poles (2 pole pairs) and no-load speed of 6000RPM.

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| From Master | 0x7A | 0x64 | 0x01 | 0x32 | 0x03 | 0xC9 | | |
| From Zikodrive | | | | | | | 0x06 | 0x03 |

0x03 = 3
$2^{(3+8)}$ = 2048
2048-1 = 2047

Electrical frequency = 2047*0.1 = 204.7Hz

Max Speed RPM = (204.7 * 60) / 2 = 6141RPM
Overspeed ratio (0x33)

If the electrical frequency exceeds this value, the controller indicates a loss of synchronisation.

0x00 = 100% Max Speed
0x01 = 125% Max Speed
0x02 = 150% Max Speed
0x03 = 200% Max Speed

Note, the low-speed limit is 25% of the Start Speed register value

Example 21 - Overspeed

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| From Master | 0x7A | 0x64 | 0x01 | 0x33 | 0x02 | 0xC9 | | |
| From Zikodrive | | | | | | | 0x06 | 0x02 |

Overspeed = 150% of max speed. If max speed = 6141RPM, overspeed threshold = 9211.5RPM

## *Settings Registers*

Basic settings for various features.

Mode (0x40)

This register allows you to select the control mode of the motor by switching between open-loop speed control, closed loos speed control and closed-loop current control.

0x00 = All closed loop systems off
0x01 = Not used on current hardware – do NOT select this option
0x02 = Closed loop current control
0x03 = Closed loop speed control

Example 22 - Speed Proportional Gain

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| From Master | 0x7A | 0x64 | 0x01 | 0x40 | 0x03 | 0xBB | |
| From Zikodrive | | | | | | | 0x06 |

Closed-loop speed control selected

Restart (0x41)

When the motor stalls or a loss of synchronisation is detected (electrical frequency smaller than low-speed limit or greater than Overspeed limit), this option allows the controller to automatically attempt to re-start or to remain off. To manually restart take the control register below the low-speed limit and then back above it again.

The restart is a single bit, 0x00 for no restart, 0x01 for restart.

Note, the low-speed limit is 25% of the Start Speed register

Example 23 - Restart

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| From Master | 0x7A | 0x64 | 0x01 | 0x41 | 0x00 | 0xBD | | |
| From Zikodrive | | | | | | | 0x06 | 0x00 |

No restart on the loss of synchronisation.

Stop on Fail (0x42)

If the controller acknowledges a fault, it can either ignore the fault and continue running or stop the motor until the control signal resets.

The stop on fail is a single bit, 0x00 for no stop on fail 0x01 for a stop on fail.

Fail examples:
Over temperature
Loss of synchronisation
Under voltage
Mosfets over current

Example 24 - Stop on fail

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| From Master | 0x7A | 0x64 | 0x01 | 0x42 | 0x00 | 0xBC | | |
| From Zikodrive | | | | | | | 0x06 | 0x00 |

No stop on fail

Control Method (0x43)

Selects the control register input

0x00 = Serial only
0x01 = Potentiometer fitted on driver
0x02 = External analogue voltage supplied

Example 25 - Control Method

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| From Master | 0x7A | 0x64 | 0x01 | 0x43 | 0x00 | 0xBB | | |
| From Zikodrive | | | | | | | 0x06 | 0x00 |

Change Address (0x44)

This will change the existing address to the new required address until the next power cycle. The address can be set between 0x00 and 0xFE

# Fault Codes

To ensure the controller is working optimally, there are a number of fault codes sent through as bytes back via the UART Bus. There are some Auto response fault codes, as well as some diagnostic codes.

On startup, two bytes are sent to test communication via UART 0xAA 0x7A

## *Auto Fault codes*

These are the codes that will be sent immediately after the fault occurs:

0xF0 - Requested register does not exist
0xF1 - A fault is detected, this is followed by two additional fault bytes.
0x15 - Incorrect Checksum, the correct checksum value immediately follows this byte.

Please note, if any byte sent to the controller is equal to 0xFF, the byte will be ignored.

It is possible to check if there are any latched faults by using an address anywhere between 0x50 and 0x5E. When using this register, no data or checksum is required (response is sent directly following the address).

The two fault bytes that will either follow 0xF1 or by the fault register are read as binary bits.

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| NA | NA | NA | NA | A | B | C | NA | D | NA | E | E | E | E | E | E |

NA - Not applicable, the data in these positions should be ignored
A - High-Temperature Warning
B - Overtemperature shutdown
C - Motor has lost synchronisation
D - Low voltage on the Power supply
E - Overcurrent fault

# Quick Commands

## *Run (0x0*)*

Control (0x01)

| | |
|---|---|
| Control 0 | 0x7A 0x64 0x01 0x01 0x00 0xFD |
| Control 1 | 0x7A 0x64 0x01 0x01 0x30 0xCD |
| Control 2 | 0x7A 0x64 0x01 0x01 0x80 0x7D |
| Control 3 | 0x7A 0x64 0x01 0x01 0x0F 0xFE |

Enable (0x02)

| | |
|---|---|
| Enable 0 | 0x7A 0x64 0x01 0x02 0x00 0xFC |
| Enable 1 | 0x7A 0x64 0x01 0x02 0x01 0xFB |

Direction (0x03)

| | |
|---|---|
| Direction 0 | 0x7A 0x64 0x01 0x03 0x00 0xFB |
| Direction 1 | 0x7A 0x64 0x01 0x03 0x01 0xFA |

Brake (0x04)

| | |
|---|---|
| Brake 0 | 0x7A 0x64 0x01 0x04 0x00 0xFA |
| Brake 1 | 0x7A 0x64 0x01 0x04 0x01 0xF9 |

Running Current (0x05)

| | |
|---|---|
| 1.25A | 0x7A 0x64 0x01 0x05 0x00 0xF9 |
| 2.5A | 0x7A 0x64 0x01 0x05 0x01 0xF8 |
| 3.75A | 0x7A 0x64 0x01 0x05 0x02 0xF7 |
| 5A | 0x7A 0x64 0x01 0x05 0x03 0xF6 |

## *Start Up (0x1*)*

Alignment Torque (0x10)

| | |
|---|---|
| 6.25% | 0x7A 0x64 0x01 0x10 0x00 0xEE |
| 31.25% | 0x7A 0x64 0x01 0x10 0x05 0xE9 |
| 62.5% | 0x7A 0x64 0x01 0x10 0x0A 0xE4 |
| 100% | 0x7A 0x64 0x01 0x10 0x0F 0xDF |

Alignment Time (0x11)

| | |
|---|---|
| 0ms | 0x7A 0x64 0x01 0x11 0x00 0xED |
| 40ms | 0x7A 0x64 0x01 0x11 0x05 0xE8 |
| 80ms | 0x7A 0x64 0x01 0x11 0x0A 0xE3 |
| 120ms | 0x7A 0x64 0x01 0x11 0x0F 0xDE |

Forced Acceleration Torque (0x12)

| | |
|---|---|
| 6.25% | 0x7A 0x64 0x01 0x12 0x00 0xEC |
| 31.25% | 0x7A 0x64 0x01 0x12 0x05 0xE7 |
| 62.5% | 0x7A 0x64 0x01 0x12 0x0A 0xE2 |
| 100% | 0x7A 0x64 0x01 0x12 0x0F 0xDD |

Start Speed (0x13)

| | |
|---|---|
| 2Hz | 0x7A 0x64 0x01 0x13 0x00 0xEB |
| 12Hz | 0x7A 0x64 0x01 0x13 0x05 0xE6 |
| 22Hz | 0x7A 0x64 0x01 0x13 0x0A 0xE1 |
| 32Hz | 0x7A 0x64 0x01 0x13 0x0F 0xDC |

## *Motor Run (0x2*)*

Phase Advance (0x20)

| | |
|---|---|
| 0° | 0x7A 0x64 0x01 0x20 0x00 0xDE |
| 9.375 ° | 0x7A 0x64 0x01 0x20 0x05 0xD9 |
| 18.75 ° | 0x7A 0x64 0x01 0x20 0x0A 0xD4 |
| 28.125 ° | 0x7A 0x64 0x01 0x20 0x0F 0xCF |

Position Proportional Gain (0x21)

| | |
|---|---|
| $K_{CP}$ = 1/128 | 0x7A 0x64 0x01 0x21 0x00 0xDD |
| $K_{CP}$ = 1/4 | 0x7A 0x64 0x01 0x21 0x05 0xD8 |
| $K_{CP}$ = 8 | 0x7A 0x64 0x01 0x21 0x0A 0xD3 |
| $K_{CP}$ = 256 | 0x7A 0x64 0x01 0x21 0x0F 0xCE |

Position Integral Gain (0x22)

| | |
|---|---|
| $K_{CI}$ = 1/128 | 0x7A 0x64 0x01 0x22 0x00 0xDC |
| $K_{CI}$ = 1/4 | 0x7A 0x64 0x01 0x22 0x05 0xD7 |
| $K_{CI}$ = 8 | 0x7A 0x64 0x01 0x22 0x0A 0xD2 |
| $K_{CI}$ = 256 | 0x7A 0x64 0x01 0x22 0x0F 0xCD |

Recirculation (0x23)

| | |
|---|---|
| Auto | 0x7A 0x64 0x01 0x23 0x00 0xDB |
| High | 0x7A 0x64 0x01 0x23 0x01 0xDA |
| Low | 0x7A 0x64 0x01 0x23 0x02 0xD9 |
| Off | 0x7A 0x64 0x01 0x23 0x03 0xD8 |

Blank Time (0x24)

| | |
|---|---|
| 0ns | 0x7A 0x64 0x01 0x24 0x00 0xDA |
| 2ms | 0x7A 0x64 0x01 0x24 0x05 0xD5 |
| 4ms | 0x7A 0x64 0x01 0x24 0x0A 0xD0 |
| 6ms | 0x7A 0x64 0x01 0x24 0x0F 0xCB |

Percentage Fast Decay (0x25)

| | |
|---|---|
| 12.5% | 0x7A 0x64 0x01 0x25 0x00 0xD9 |
| 25% | 0x7A 0x64 0x01 0x25 0x01 0xD8 |

PWM Frequency (0x26)

| | |
|---|---|
| 50Khz | 0x7A 0x64 0x01 0x26 0x00 0xD8 |
| 35.7Khz | 0x7A 0x64 0x01 0x26 0x05 0xD3 |
| 27.78Khz | 0x7A 0x64 0x01 0x26 0x0A 0xCE |
| 22.73Khz | 0x7A 0x64 0x01 0x26 0x0F 0xC9 |
| 14.37Khz | 0x7A 0x64 0x01 0x26 0x1F 0xB9 |

## *Closed Loop Control (0x3*)*

Speed Proportional Gain (0x30)

| | |
|---|---|
| $K_{SP}$ = 1/128 | 0x7A 0x64 0x01 0x30 0x00 0xCE |
| $K_{SP}$ = 1/4 | 0x7A 0x64 0x01 0x30 0x05 0xC9 |
| $K_{SP}$ = 8 | 0x7A 0x64 0x01 0x30 0x0A 0xC4 |
| $K_{SP}$ = 256 | 0x7A 0x64 0x01 0x30 0x0F 0xBF |

Speed Integral Gain (0x31)

| | |
|---|---|
| $K_{SP}$ = 1/128 | 0x7A 0x64 0x01 0x31 0x00 0xCD |
| $K_{SP}$ = 1/4 | 0x7A 0x64 0x01 0x31 0x05 0xC8 |
| $K_{SP}$ = 8 | 0x7A 0x64 0x01 0x31 0x0A 0xC3 |
| $K_{SP}$ = 256 | 0x7A 0x64 0x01 0x31 0x0F 0xBE |

Max Speed (0x32)

| | |
|---|---|
| $F_{MX}$ = 25.5Hz | 0x7A 0x64 0x01 0x32 0x00 0xCC |
| $F_{MX}$ = 204.7Hz | 0x7A 0x64 0x01 0x32 0x03 0xC9 |
| $F_{MX}$ = 3276.7Hz | 0x7A 0x64 0x01 0x32 0x07 0xC5 |

OverSpeed Limit Ratio (0x33)

| | |
|---|---|
| 100% | 0x7A 0x64 0x01 0x33 0x00 0xCB |
| 125% | 0x7A 0x64 0x01 0x33 0x01 0xCA |
| 150% | 0x7A 0x64 0x01 0x33 0x02 0xC9 |
| 200% | 0x7A 0x64 0x01 0x33 0x03 0xC8 |

## *Settings (0x4*)*

Mode (0x40)

| | |
|---|---|
| Indirect Speed (off) | 0x7A 0x64 0x01 0x40 0x00 0xBE |
| Direct Speed | 0x7A 0x64 0x01 0x40 0x01 0xBD (not used, do not select) |
| Closed Loop Current | 0x7A 0x64 0x01 0x40 0x02 0xBC |
| Closed Loop Speed | 0x7A 0x64 0x01 0x40 0x03 0xBB |

Restart on loss of Synchronisation (0x41)

| | |
|---|---|
| No Restart | 0x7A 0x64 0x01 0x41 0x00 0xBD |
| Restart | 0x7A 0x64 0x01 0x41 0x01 0xBC |

Stop on Fail (0x42)

| | |
|---|---|
| No Stop | 0x7A 0x64 0x01 0x42 0x00 0xBC |
| Stop | 0x7A 0x64 0x01 0x42 0x01 0xBB |

Control Method (0x43)

| | |
|---|---|
| Serial Only | 0x7A 0x64 0x01 0x43 0x00 0xBB |
| OnBoard POT | 0x7A 0x64 0x01 0x43 0x01 0xBA |
| External Analogue Input | 0x7A 0x64 0x01 0x43 0x02 0xB9 |

Change Address (0x44)

| | |
|---|---|
| New Address = 0x03 | 0x7A 0x64 0x01 0x44 0x03 0xB7 |