ORCA™ MOTORS

Reference Manual 220115

This document is written for the following versions of the ORCA motors firmware:

- 6.3.4
- 7.1.5

Download the latest version of this reference manual at https://irisdynamics.com/downloads

CONTENTS

Revision History

Version	Date	Author	Reason
0.0	January, 2022	kh	Initial Draft
1.0	April, 2022	sj	Include GUI and Kinematic sections
1.2	August, 2022	kh	Review and update
1.3	September, 2022	sj	Update for 6.x.5 changes.
1.4	February, 2023	Sj	Update for 6.1.6 changes.
1.5	April, 2023	rm	Formatting, memory map update
1.6	May, 2023	rm, ab, sj	Update GUI, add Haptics section, change section order
1.7	August, 2023	kh, sj	Expand on control register
1.8	December, 2023	sj, rm	Update for 6.1.8 changes, autozeroing, negative positions, safety warnings
1.9	April 2024	sj	Update for 6.2.8 patch changes.
1.10	June, 2024	sj	Update with IOSH Autozeroing Information
1.11	Jan, 2025	sj	Update with PWM Information. 6.3.0 Changes
1.12	Mar, 2025	sj	Update with 6.3.1, 7.1.1 firmware version numbers. Add COM port latency troubleshooting.
1.13	May, 2025	sj, jg	Improve CTRL_REG_1 documentation, add Modbus RS485 mode information, update memory map, and updated motor setup image
1.14	June, 2025	sj	Update with firmware 6.3.3, 7.1.4. Adjust auto zero error information.
1.15	July, 2025	sj	Update with firmware 6.3.4, 7.1.5 information.
1.16	July, 2025	jg	Updated with tutorials.
1.17	Aug, 2025	jg	Updated splitter notation.

Introduction

This document describes the functions and operation of ORCA motors having integrated drivers and an integrated sensor suite.

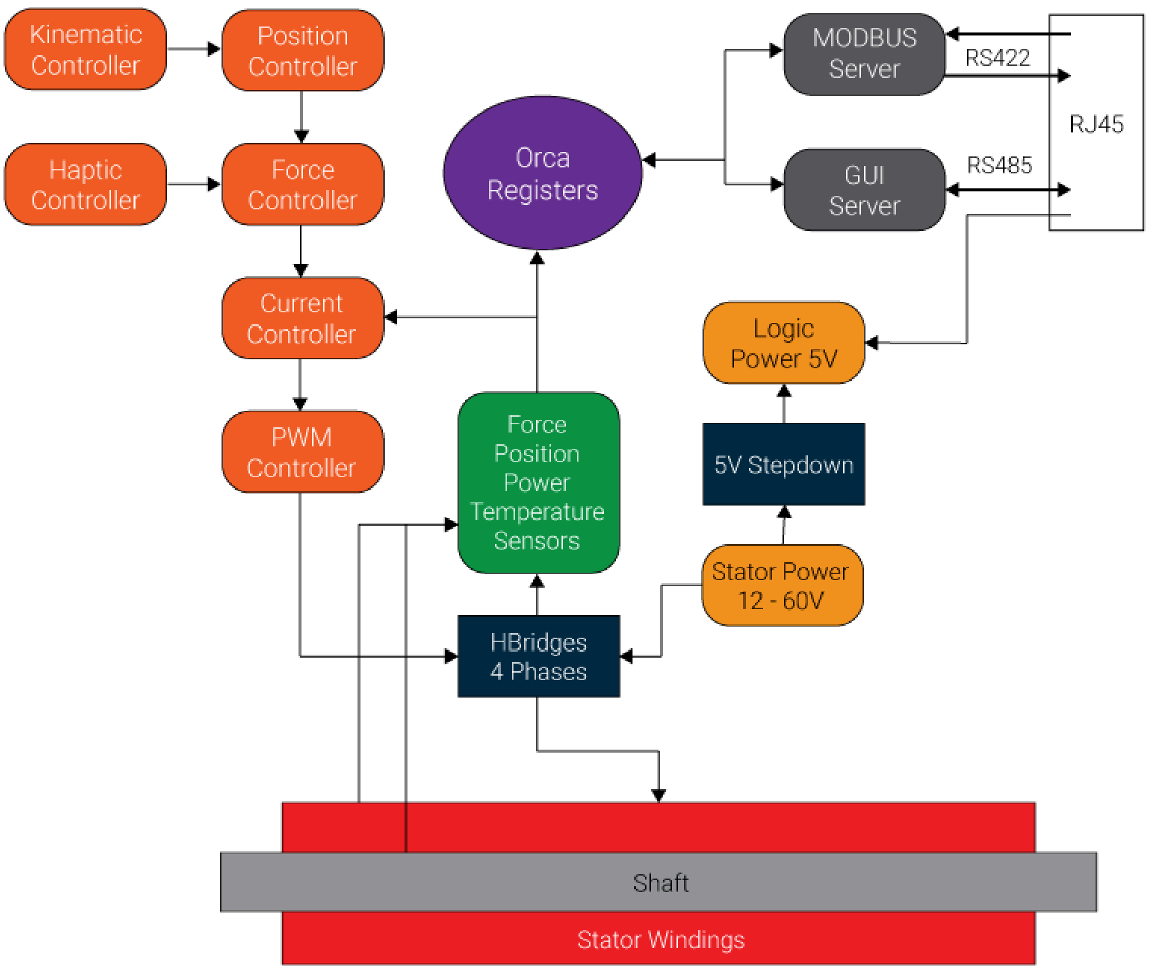


Figure 1: Block Diagram

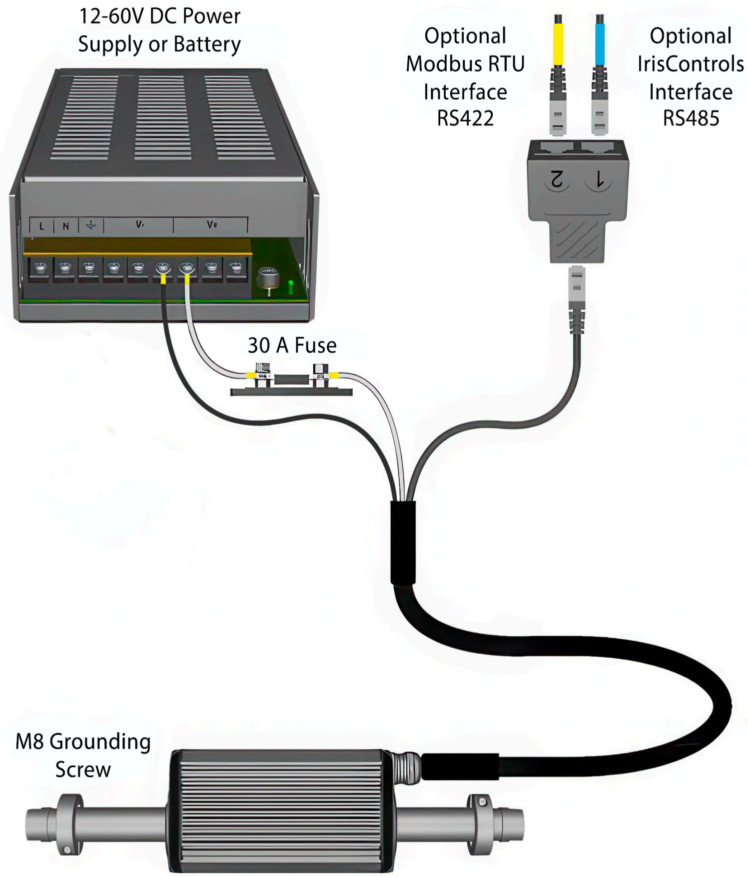


*Figure 2: ORCA Motor's Setup. *Note, the splitter may also be marked port A and B; ports 1 / A and 2 / B are equivalent.*

Safety Warnings

⚠WARNING⚠ The motor's shaft contains STRONG magnets. Keep away from ferrous metals and sensitive electronics.

⚠WARNING⚠ Ensure the motor is mounted in a safe location to avoid danger or damage to people or property.

⚠WARNING⚠ Pinching hazard between shaft collars and stator body.

⚠WARNING⚠ On start up the current motor position is zero. Absolute positions are not kept through a power cycle. (A feature is available for autozeroing on start up).

⚠WARNING⚠ In the event of loss of power the shaft moves freely within the stator.

Powering the Motor

ORCA motors are powered by DC voltage. The ORCA motors variant name specifies the voltage at which the motor is most efficient, however all ORCA motors can be powered by 12 – 60 VDC.

Flying lead power connections are provided for each motor. ORCA motors are configured to be able to draw significant currents and proper attachment of the power lines to a suitable power source is important.

Suitable power sources for ORCA motors are AC-to-DC converters, batteries of various chemistry, capacitor banks, or other sources of DC voltages. Engineering support from Iris Dynamics on the selection of a suitable power source is available.

Fuse

A 30-amp fast-blow fuse must be installed between the power supply and the motor. This fuse will be destroyed if negative voltage is applied to the motor.

Ground

In all cases, the black power lead is "Ground" and should be connected to the negative terminal of the power supply or battery.

It is important that the power supply Ground, and the Ground of any electronics attached to the device's communication cable is kept at the same level. Usually this is done by making sure the power supplies of the motor and external controller (if different) are connected to Earth at both their chassis and negative output terminal.

Chassis

As an important safety measure, the chassis of the motor must be securely connected to Earth. Proper grounding of the chassis is also important for proper operation of the motor.

The motor enclosure and cable shields are connected but these are not connected to the Ground circuit of the on-board electronics. Therefore, a separate connection from the Chassis to Earth must be made which can be accommodated by threading a conductive bolt into any of the provided thread patterns on the enclosure.

Communication Interfaces

Communication cables are terminated with an RJ45 connector. The 5V and GND lines on the connector can be used to power the motor's logic so that communications and diagnosis can continue when the motor's power source is switched off or disconnected.

Pin	Signal	Notes
1	RX_2+ / PWM	Modbus / PWM Control input.
2	RX_2-	
3	TX_2+	
4	RX/TX_1+	IrisControls™
5	RX/TX_1-	
6	TX_2-	Modbus
7	+5V	Can be used to power logic, or indicate logic power status
8	GND	Must be connected to the communication hub ground.

There are two serial communication interfaces available which enable a Windows-based GUI (via IrisControls), firmware updates, and a Modbus RTU communication stream. An RJ45 splitter can be used to allow both communication streams simultaneously. Depending on the model of splitter, the ports may be labelled as 1 and 2 or as A and B. Ports 1 and A, are equivalent, as are ports 2 and B.

Splitter Port	Signaling	Protocols
1 / A	RS485 Half Duplex	IrisControls (GUI) IrisBootloader
2 / B	RS422 Full Duplex 4-Wire*	Modbus RTU

* See [UG230323] ORCA motors Modbus over Half-Duplex RS485 (available at irisdynamics.com) for information on setting up a 2-wire half-duplex connection on this port.

Methods of Motor Control

Modbus RTU Serial Interface

ORCA motors feature a 'field-bus' serial communication interface which allows configuration, control, and monitoring. Features of the motors are offered by exposing data fields (registers) which can be written to and read from by sending and receiving characters over the serial interface. Serial communications are implemented using a subset of the Modbus RTU specification, with additional functionality to support a high-speed stream of commands and feedback.

See the ORCA Memory Map for a list of register addresses. A PLC, PC, or other Modbus client can be used to command the motor. Third party programs with built in Modbus libraries or serial libraries such as Labview, or Matlab can also be used. There is an IrisSDK for Windows available with C++ libraries that abstract the Modbus communications for use in custom software solutions.

The Modbus RTU User Manual, along with manuals for specific client interfaces, links to git repository with source code are available for download at irisdynamics.com/downloads.

IrisControls Software

ORCA motors feature an optional graphical user interface called IrisControls which can be used to monitor details and configure settings. This interface provides an easy way to visually tune the internal PID position controller, set up motion profiles, add performance restrictions, and capture information while connected. See the Getting Started With IrisControls section of this document for full details.

IO SmartHub *(optional and sold separately)*

The IO SmartHub provides control of ORCA motors in Force, Position, and Kinematic Modes through simple digital and analog inputs. Real-time force and position data are fed back from the motor and provided as analog outputs. The IO SmartHub attaches to the motor's data cable (RJ45) and allows easier integration with existing industrial control methods such as PLCs with 4-20 mA current loop outputs. Find more information in the IO SmartHub User Guide at irisdynamics.com/downloads.

Modes of Operation

The motor operates in five main modes of operation: Sleep Mode (1), Force Mode (2), Position Mode (3), Haptic Mode (4), and Kinematic Mode (5). Force Mode (2) and Position Mode (3) are meant to be used with a Modbus communication stream and cannot be entered directly through the IrisControls GUI.

Sleep Mode (1)

In this mode, power drivers are not active and windings in the stator are all shorted (producing a braking effect). Force and position commands are ignored.
When Sleep Mode (1) is commanded (even when Sleep Mode is currently active), any active errors will be re-evaluated and cleared where appropriate.

Force Mode (2)

The force controller will control the power delivered to the motor to achieve the forces written to the FORCE_CMD register. Note: this register is double-wide (32 bits) and read as a signed 32 bit integer.
Once in Force Mode (2), communications must be made regularly over the Modbus interface to avoid a timeout error. In the event of communications failure or other motor errors, the motor will cease all power draw and produce zero force output.

Position Mode (3)

The position controller will calculate forces for and command the force controller according to the configured PID tuning, the setpoint, and the current shaft position.
Before entering Position Mode (3), the user should make sure the shaft has been zeroed to a known position, either manually or through the autozeroing feature.
Once in Position Mode (3), communications must be made regularly over the Modbus interface to avoid a timeout error. In the event of communications failure or other motor errors, the motor will cease all power draw and put the windings into a passive braking effect.

Haptic Mode (4)

In Haptic Mode, the force controller works similarly to Force Mode, however, force commands will be dictated by the haptic controller rather than a Modbus stream. The haptic controller comprises several haptic force effects that can be configured and enabled. The sum of forces resulting from all enabled haptic effects is used as the force target. A Modbus stream can be used to update the constant force effect, or updating other effect configurations.

Kinematic Mode (5)

In Kinematic Mode, the position controller works similarly to when in Position Mode (3), however, position targets will be dictated by the kinematic controller rather than a Modbus stream.
The kinematic controller outputs motion profiles to move to a shaft position over a specified time, while respecting the chosen kinematic constraints. The Modbus interface can be used in this mode to configure motion profiles and trigger motions.

Pulse Width Mode (11)

In Pulse Width Mode, the position controller works similarly to when in Position Mode (3), however, position targets will be dictated by the width of incoming pulses. In this mode, the motor can effectively be controlled like an RC servo motor.

Use of this mode requires additional hardware described in the Pulse Width Control section.

Auto Zeroing Mode (55)

This mode is available to put the motor into a routine that will automatically retract the shaft until reaching a hard stop, and set the zero position to that location. Once the zeroing routine is complete the motor will move into the specified Exit Mode.

More information on the Auto Zeroing feature can be found in the Auto Zeroing section.

Getting Started with IrisControls

Required Software

ORCA motors can be configured through an integrated graphical user interface (GUI). To connect to an ORCA motor's GUI, first download the latest version of the IrisControls software on a Windows PC.

The latest release is always available at www.irisdynamics.com/downloads under the 'Software' section.

Once downloaded, unzip the folder, run the Setup_IrisControls_vX.X.X-... application and allow the installer to complete setup.

IrisControls Software without an ORCA motor Connected

Required Hardware

Besides the motor itself, connecting to the IrisControls application requires two additional pieces of hardware detailed below.

ORCA motor

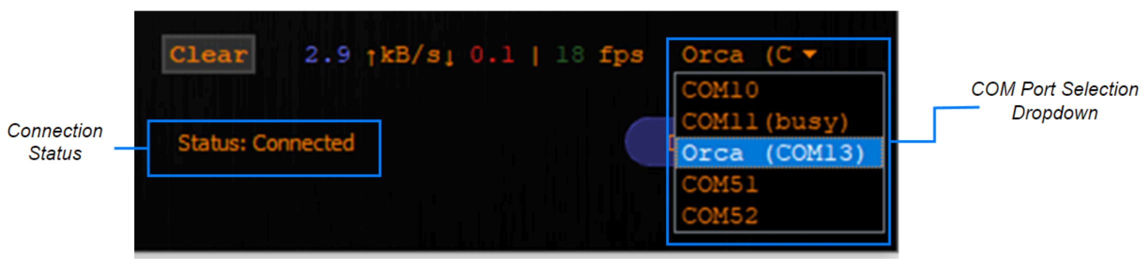
RS485 to USB cable

RJ45 Splitter

Connect the RJ45 communication cable from the ORCA motors to the single port side of the RJ45 splitter. Connect the RJ45 connector end of the blue RS485 cable to the splitter input labeled 1 or A. Finally, connect the USB end of the blue RS485 cable to the Windows PC running IrisControls.

Connecting to IrisControls

Ensure that the motor is connected to a PC with an RS485 to USB cable. Open the IrisControls application. Using the COM selection dropdown menu at the bottom of the window, select either ORCA (if it is present) or the COM port of the USB to RS485 connection. If neither option is present, ensure that the hardware is connected correctly as specified.



IrisControls Connection Information

Once the correct device is selected, press the connect button to begin attempting to connect to the motor. If successful, the ORCA GUI should launch its homepage.

Troubleshooting Connection

Correct Splitter Port

If the COM port does not show up in the drop down menu, ensure that the blue RS485 to USB cable is plugged into port 1 or A of the RJ Splitter and that the motor's data cable is plugged into the single side of the splitter.

FTDI Drivers

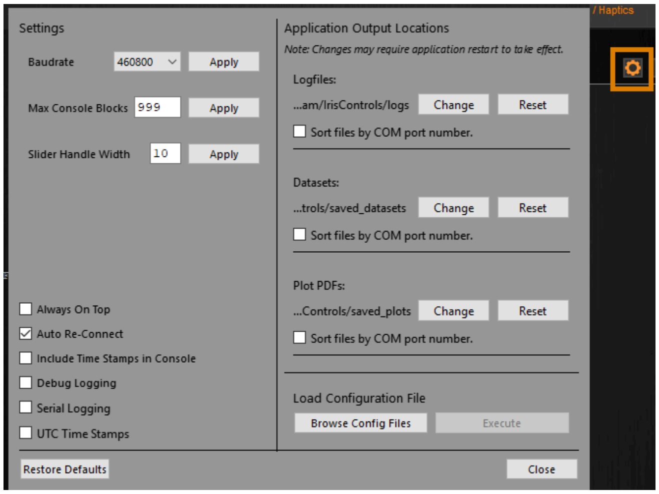
If the COM port does not show up in the drop down menu, it may be due to not having the FTDI Virtual COM Port drivers installed. Most newer Windows operating systems have these drivers included, but they can also be found at https://ftdichip.com/drivers/vcp-drivers/.

Too Many Rows / Columns Requested Error

This error can arise from a combination of a screen's scaling and the size of the display. The first option to resolve this is to turn down the screen scaling (windows key -> 'scaling' -> "Change the size of text, apps, and other items").

GUI Baudrate

If the motor does not connect, toggle the connect button off and press the gear icon in the top right of the IrisControls window and select a baudrate of 460800 from the dropdown menu, and press apply.



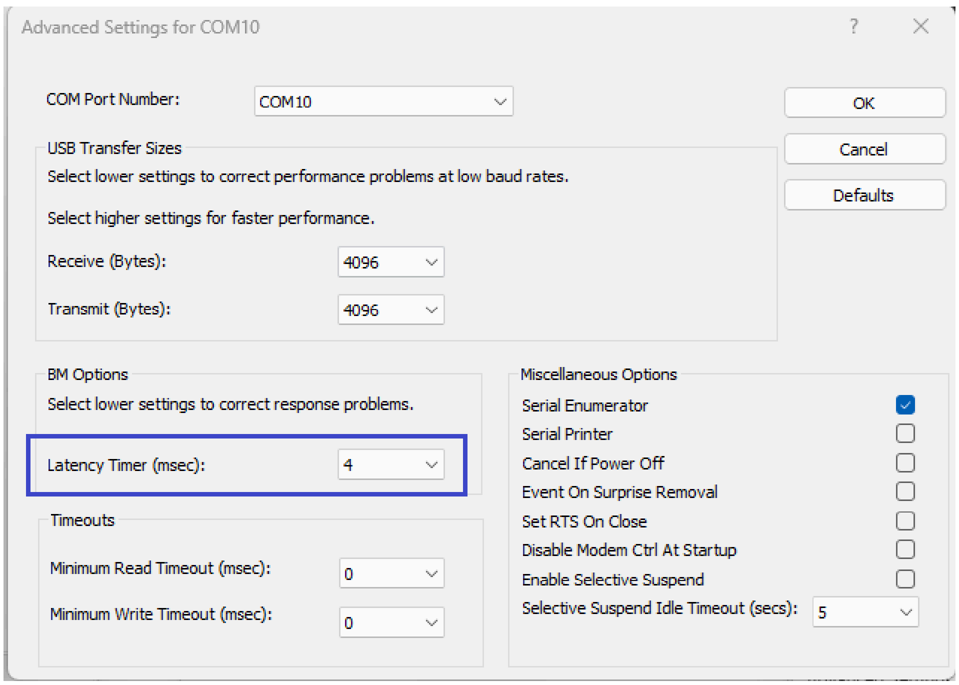
IrisControls Settings Dialog

Low Framerate

If the GUI is refreshing at a low framerate (less than 20 Hz), this can be caused by the COM port latency value. On Windows, open the device manager, and navigate to the Ports (COM &

LPT) section. Set the COM port latency for the RS485 cable to 4 ms under **Properties > Port Settings > Advanced**.

Note: Setting the latency below 4 ms is **<u>not</u>** recommended, as it can cause communication issues.



COM Port Latency Setting

Navigating the GUI

Navigation of the GUI is done using the page buttons above the console. The buttons can also be pressed with numerical keyboard shortcuts, listed beside the page name. Pressing each button will open a different page of the GUI within the page content area. The default page upon connection is the home page, shown below. Several other indicators, such as power draw and mode of operation will remain visible on the GUI regardless of the page selection.

Home Page

ORCA GUI with Home Page Open

The home page is the default page opened upon connecting to an ORCA motor. A screenshot of the home page is shown in Figure 9. This page displays the firmware version, hardware version, firmware build date, and the motor serial number. There are no interactive elements specific to this page.

Position / Kinematics Page

The position page provides a real time graph displaying the target position, measured position, and target force of the motor. Below the graph, the currently set parameters of the position controller are displayed, including proportional gain, integral gain, derivative gain, and maximum force output. These values can be tuned and are described in further detail in the PID Position Controller section.

IrisControls Position Page

Position and Force Graph

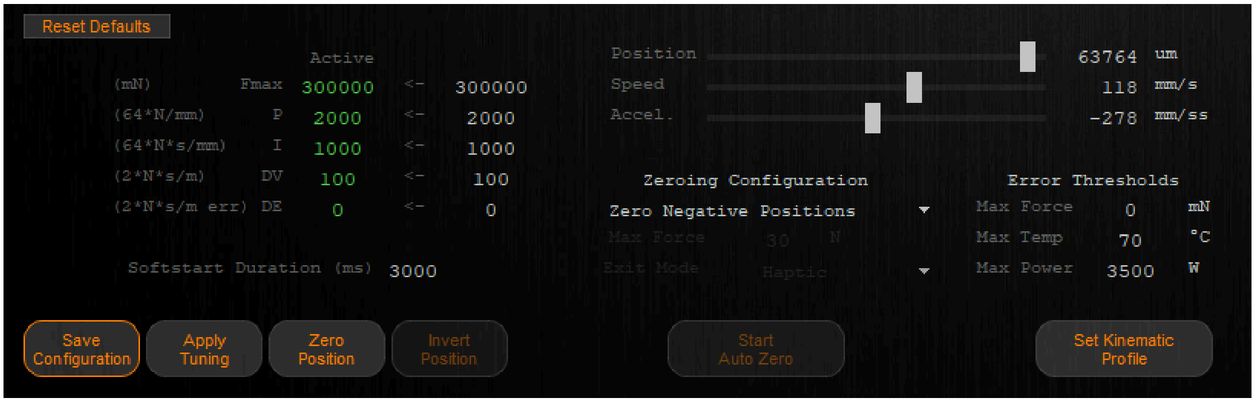
This graph will plot the position controller's target position, the commanded force used to reach that position, the measured shaft position, and the individual actions of each component of the PID controller. The left y-axis is position, and the right y-axis is force.

The green "Target" line will show the active commanded position. (Modbus or Kinematic Controller).

The blue "Position" line indicates the motor's sensed position.

The other lines indicate the overall force being commanded to the motor and the contributions from each PID parameter to that force.

Tuning Panel



IrisControls Tuning Panel

PID Tuning - Max Force / Gain Adjustment

The maximum force output and position controller gains can be edited here.
The maximum force (Fmax) determines the upper limit of force that the position controller can command.

The number on the left of the arrow is the actual value that is in effect and is not editable. This value will only be populated when the position controller is active either in Position Mode (3) or Kinematic Mode (5)

The number on the right side of the arrow is the target gain value. This value can be manually edited and applied with the Apply Tuning button and saved with the Save Tuning button.

Softstart Duration configures the duration that forces generated by the position controller will linearly ramp up over when entering the mode.

Sensor Sliders

Three sliders show the current values for position, speed and acceleration.

Error Thresholds

Allows configuration of the force, temperature and power thresholds that trigger their respective errors when exceeded.

Save Tuning Button

This button will save the gain target values and the maximum force value to permanent memory to allow them to persist through a power cycle.

Apply Tuning Button

To update the gains to their target values while operating in Position Mode (3), press the update gains button.

Invert Position Button

Change direction of positive movement. This should only be done while the motor is in Sleep Mode (1) to avoid unexpected movement.

Zero Position Button

Set the motor's zero position to the current location. Zero position does not persist through motor power cycles.

Zeroing Configuration

There are four schemes for zeroing the motor's position that may be selected here.

- **Zero Negative Positions (Default)**
- **Manual Zeroing**
- **Auto Zero Enabled**
- **Auto Zero on Boot**
- **IOSH Auto Zeroing**

These settings are described in detail in the Position Zeroing Mode section.

Auto Zeroing Max Force

The maximum force exerted during the auto zeroing routine can be configured here, in Newtons.

Auto Zeroing Speed

The speed at which the shaft will move during the auto zeroing routine can be configured here, in millimeters per second.

Start Auto Zero Button

This button only is available if one of the auto zero configurations are selected. These configurations can be selected from the drop down above the button. This will trigger the Auto Zeroing routine and then put the motor into its specified exit mode.

Set Kinematic Profile Button

This button will switch the interface to the kinematic panel to configure kinematic motion profiles.

Application Note:

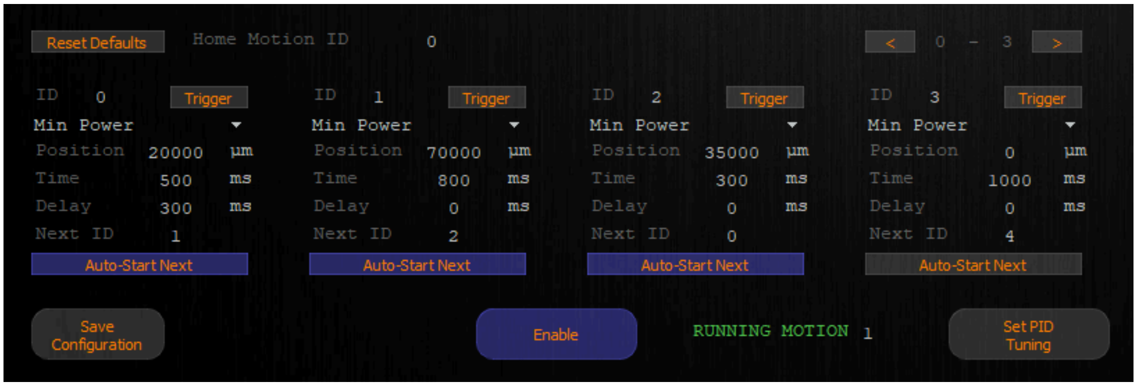
Configuring and Using Auto Zeroing From the IrisControls GUI

To use the Auto Zeroing feature from the GUI:

1. Navigate to the Position / Kinematics page. If the Kinematic controls are showing, press the Set PID Tuning button to switch to the tuning panel.
2. Select a valid zeroing configuration from the Zeroing Configuration dropdown. If an invalid zeroing configuration is selected, the Start Auto Zeroing button will be disabled. Valid settings include:
    - Auto Zero Enabled
    - Auto Zero on Boot
    - IOSH Auto Zeroing
3. Configure the maximum allowable force to be exerted during the Auto Zeroing routine, in Newtons. This is done through the Max Force input. Note that this force may be exerted for a short period of time (<200 ms) when the shaft reaches its hard stop. It's important to set this parameter high enough such that the motor is still able to move its load.
4. Configure the speed at which the shaft should move during the Auto Zeroing routine, in millimeters per second. This is done through the Shaft Speed input.
5. Select the mode of operation the motor should transition into after finishing Auto Zeroing. This is done through the Exit Mode dropdown. Upon successful completion of Auto Zeroing, the motor will immediately enter the specified mode.
6. To initiate Auto Zeroing from the GUI, press the Start Auto Zero button. This button is only available if a valid mode is selected as described in step 1.

Kinematic Panel

The kinematic page provides an interface to the kinematic controller feature of the ORCA motor. For more information on the capabilities of the kinematic controller, refer to the Kinematic Controller section of this manual.



IrisControls Kinematic Panel

Reset Defaults Button

Resets current kinematic settings to the factory default demo motion profile.

Home Motion ID

This is the ID of the motion that will start when the kinematic controller is enabled, either via the GUI, Modbus, or the IO SmartHub.

Configuration Page Buttons

The 32 possible motions can be paged through using the left and right arrow buttons at the top right of the page. The current range of motion IDs being configured is shown between the page buttons.

Motion Configuration Boxes

The motion configuration boxes below the main graph represent the configuration of four motions at a time. The motion ID can be seen at the top of each box.

Save Configuration Button

Saves the global settings and motion ID configurations to permanent memory.

Enable Button

This button toggles the motor between Kinematic Mode (5) and Sleep Mode (1).

State Indicator

The state indicator will read CONTROLLER IDLE if the kinematic controller is not currently executing a motion. It will read RUNNING if a motion is in progress, with a number to the right indicating which motion ID is running.

Configuring a Motion

1. Use the page buttons to navigate to the page with the desired motion ID.
2. In the configuration box with the desired motion ID, enter the motion parameters (position, time, type, delay, next ID, and auto-start). Press the enter key after inputting a value in one of the text boxes.
3. Enable the kinematic controller to test the configured motion settings.

Press the Save Configuration button to save changes to permanent memory.

Trigger Buttons

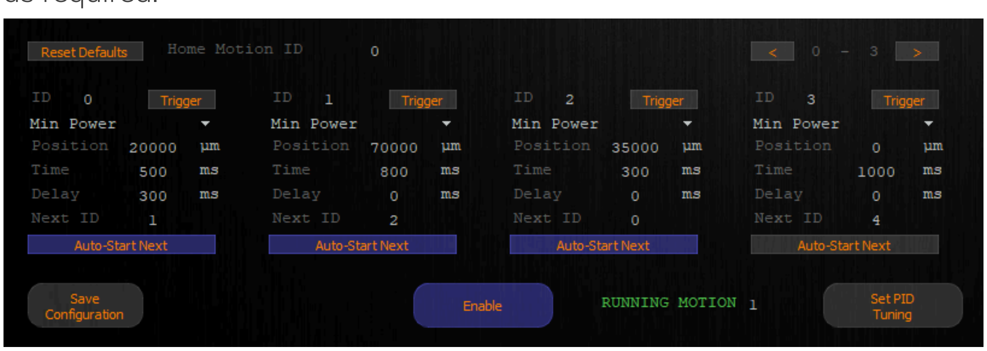
A trigger button is available for each motion ID which will start that motion and any sequential motions (Must be in Kinematic Mode (5) for trigger to have effect).

Application Note:

Continuous Looping Sequence

It is possible to set up a single motion profile that will start on motor boot up and loop indefinitely without any external control, i.e., motor with power supply only, data cable not used.

This can be done by setting a sequence of motions that all ‘Auto-start Next’ and having the last motion’s ‘Next ID’ be the first in the sequence. The ‘Home Motion ID’ should be set to the first motion in the sequence. When Kinematic Mode (5) is entered, each motion will be performed sequentially as defined by the ‘Next ID’ parameter. Delays between motions can be added as required.



Kinematic Looping Sequence Configuration

Force / Haptics Page

The haptics page provides an interface for the haptics features of the ORCA motor.

IrisControls Force / Haptics Page

Force Graph

This plot displays the force sensed by the motor and the force commanded by all haptic effects. The plot can also show the individual forces commanded for up to three haptic effects. The force range on the plot is configurable using the Range input.

Enable Haptics Button

Toggling this button will switch the motor into Haptics Mode (4) and will begin applying all enabled haptics effects. Toggling it again will return the motor to Sleep Mode (1).

Save Configuration Button

Clicking this button will save the haptic effects currently displayed on the GUI to the long-term memory on the motor.

Target and Sensed Force Sliders

These sliders cannot be interacted with, they display the current target (or commanded) force along with the sensed force.

Individual Haptic Effect Panels

Each of the other panels on this page provide an interface for adjusting the various haptics effects available to the motor. Each of the data fields in these panels can be modified to adjust the effect. Clicking the enable button contained in a panel will enable that individual haptic effect. The save configuration button will save any changes made here to permanent memory. As many effects as desired can be layered on each other.
The following is a list of the effects with a brief description. For more detailed information on each effect and their configurations, see the Haptic Controller section.

Constant

Constant force in either the positive or negative direction, regardless of shaft position. This input can be low pass filtered with an adjustable strength to accommodate updates from slower communication streams.

Spring

Force that linearly increases with distance from the configured center position. Adjustable in strength, saturation force, dead-zone, and coupling direction.

Damper

Force that linearly increases with speed to oppose the current direction of travel.

Inertia

Force that linearly increases with acceleration to maintain motion of the motor in the direction it is moving.

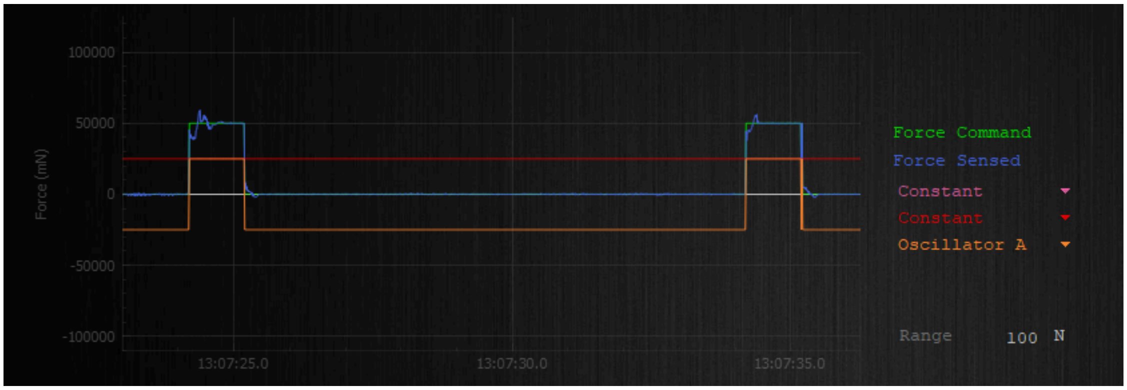
Oscillator

Force waveform generator with adjustable frequency, amplitude, and wave type.

Application Notes:

50 N force for 1 second followed by 9 seconds of rest.

To accomplish this type of setup the ‘Oscillator’ and ‘Constant’ effects are used in combination. Using a 25 N amplitude for the oscillator will give a 50 N spread between the maximum and minimum values. Using a ‘Constant’ effect with 25 N will add an offset to the oscillator. Instead of alternating between +25 N and -25 N, the force will alternate between 0 N and 50 N. Setting the Frequency to 1 dHz will have the waveform repeat every 10 seconds (which is the total time ‘high’ and ‘low’). Setting the Duty to 10 will give 1 second high and 9 seconds low.



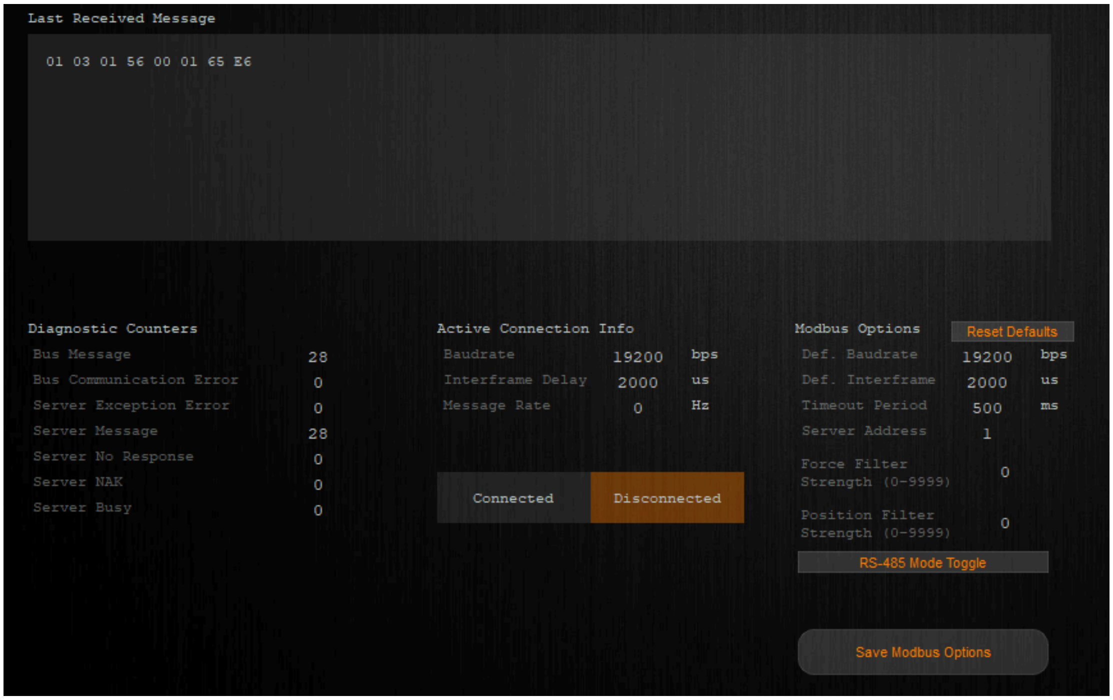
Force Graph when 25 N Constant Force and 25 N Amplitude Pulse Oscillator with 10% Duty Cycle

Testing of High Force

Set up a ‘Spring’ effect with a large dead zone and high Gain to create virtual hard stops. Then use a ‘Constant’ effect to the desired force. This will allow for the demonstration of high forces within a range of the shaft without pushing against hard stops.

Modbus Page

ORCA motors support the Modbus RTU serial communication protocol as a control method. This GUI page allows the user to view the status of the Modbus communication.



IrisControls Modbus Page

Last Received Modbus Message

All bytes of the last received Modbus message are displayed on this page. If the message was successful, bytes will be displayed in white, if there is an error with the message; incorrect length, CRC byte missing/wrong, unsupported function code, wrong address etc, the bytes text will be red.

Diagnostic Counters

Modbus diagnostic counters are shown below the last received message, on the left-hand side of the page. See Modbus RTU specification for more information on each counter.

Active Connection Info

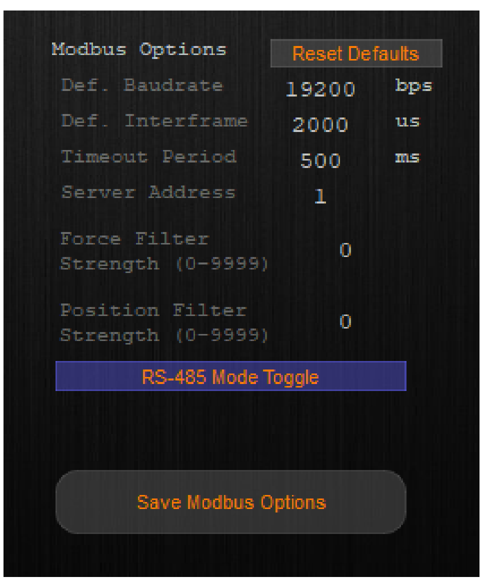
Connection information is shown below the last received message, in the center of the page. Information shown includes current baudrate, rate of successful messages and connection status. If a high speed negotiation has taken place successfully (i.e., baudrate and interface delay have been negotiated other than the default using function code 65 - manage high speed stream) the Connected label will be highlighted orange. If communication is taking place without connection negotiation, Disconnected will be highlighted.

Modbus Options

Default Modbus behaviour can be configured here to match any Modbus client setup or to increase messages framerate without requiring negotiation.
Default baudrates up to 1000000 bps (1 Mbaud) can be configured. Interframe delay can be decreased as low as 0. Timeout period which will dictate when a message timeout error is triggered can also be configured here.

Force and position input filters are available to smooth out force or position target commands if Modbus communication rate is slow. A value of 0 will result in no filtering whereas a value of 9999 will provide maximum filtering. Note that due to the nature of the filter, values between 9000 and 9999 are most likely to be useful during high speed communication.

The RS485 mode setting enables half-duplex or two-wire RS485 communication. Communication using half-duplex allows data to be sent and received by both wires, one direction at a time. The RS485 mode setting can be enabled through clicking the RS-485 Mode Toggle button within Iris Controls or by writing a value of 1 to the MB_485_MODE register (address 174). Further information regarding this setting is also available in [UG230323] ORCA motors Modbus over Half-Duplex RS485, available at irisdynamics.com.



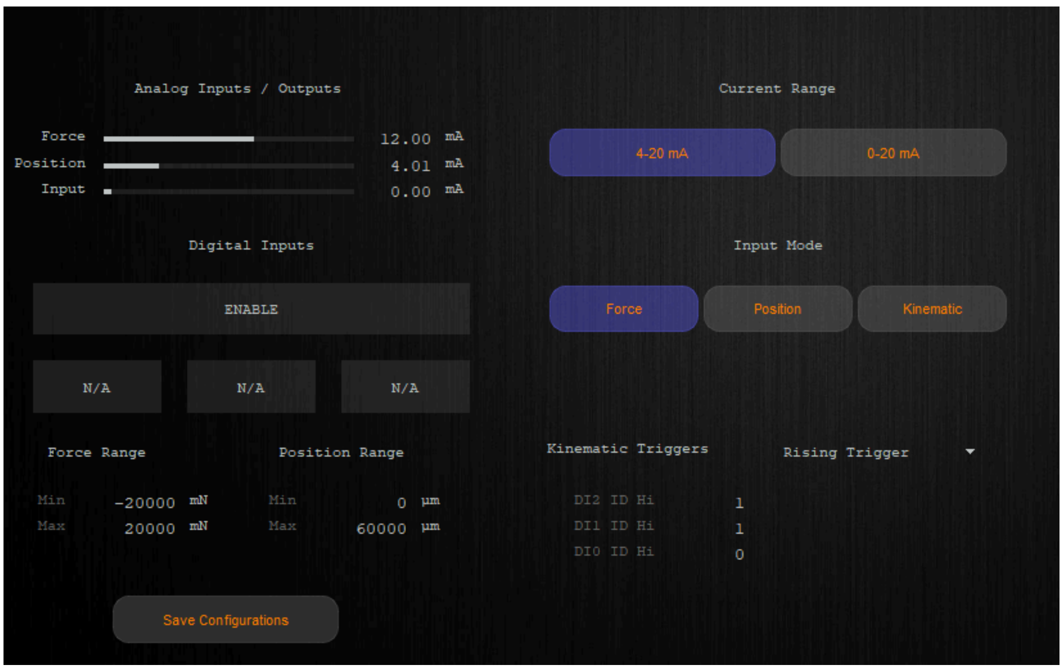
RS-485 Setting Enabled Within Iris Controls

Save Modbus Options Button

Pressing the will put new configurations into effect immediately and will be saved to permanent memory.

IO SmartHub Page

This panel contains the status and configuration options for the IO SmartHub. To find more information on how to configure and use the IO SmartHub, please see the IO SmartHub User Manual available at irisdynamics.com/downloads.

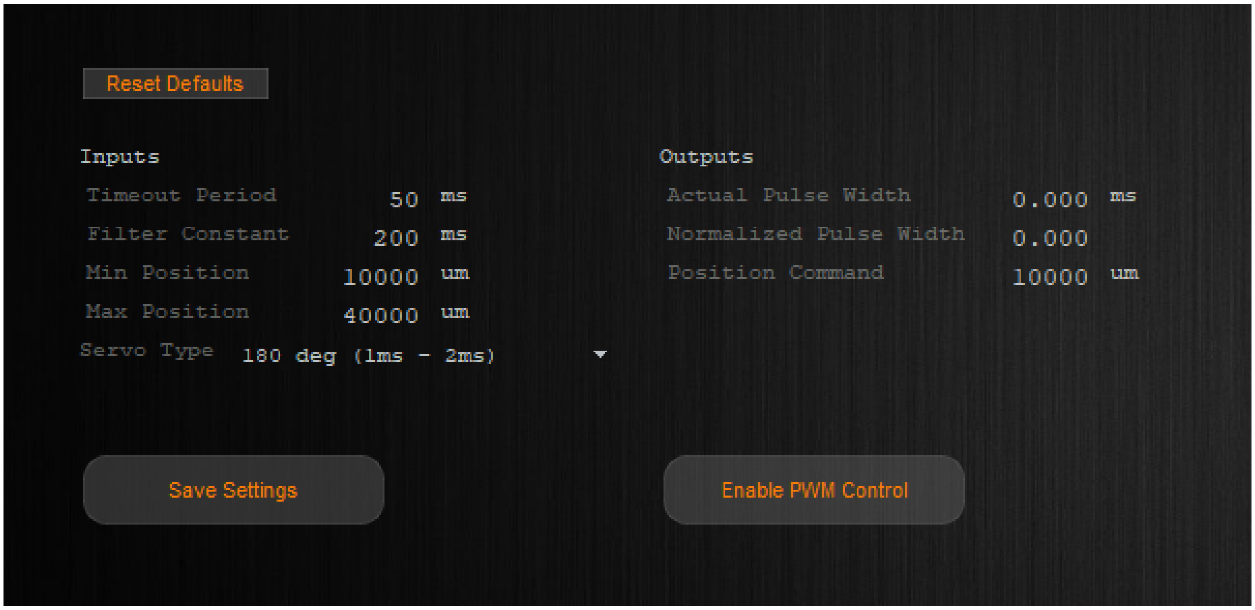


IrisControls IO SmartHub Page

Pulse Width Page

Note: This page is not accessible through the page buttons. If access is required, the console command “pwm” will open this page. See Pulse Width Control for more information.

This panel contains configurations and status outputs for pulse width control mode.



Buttons

Save Settings

Saves the currently selected input settings to permanent memory.

Enable PWM Control

Enters the Pulse Width mode (11)

Inputs

Timeout Period

Selects the amount of time between pulses before a Comms Timeout (2048) error is thrown.

Filter Constant

Time constant for a low pass filter applied to the calculated position command. Higher values have the effect of smoothing the position control, at the cost of slower response times. This is helpful when the pulse width signal is low frequency.

Min Position

Motor position target when the minimum pulse width is detected.

Max Position

Motor position target when the maximum pulse width is detected.

Servo Type

Dropdown menu to select two pulse width control schemes:

- 180 Degree Servo - Pulse range of 1.0ms to 2.0ms
- 270 Degree Servo - Pulse range of 0.5ms to 2.5ms

Outputs

Actual Pulse Width

The last actual detected pulse width on the input. Only updated when in Pulse Width Mode (11).

Normalized Pulse Width

The last pulse width detected on the input, normalized to the selected pulse width range. Only updated when in Pulse Width Mode (11).

Position Command

The live position command that is being sent to the position controller from the pulse width input. Only updated when in Pulse Width Mode (11).

IrisControls Logging

While connected to IrisControls, ORCA motors provide periodic logging of various sensor data. Logging will start automatically upon connection with IrisControls. Log data will be saved to a file named "Orca_[serial number]_data_log.txt" in the the logs folder located by default in the IrisControls folder in the Users directory. The save location can be changed in the IrisControls Settings menu which is accessed via the gear icon.

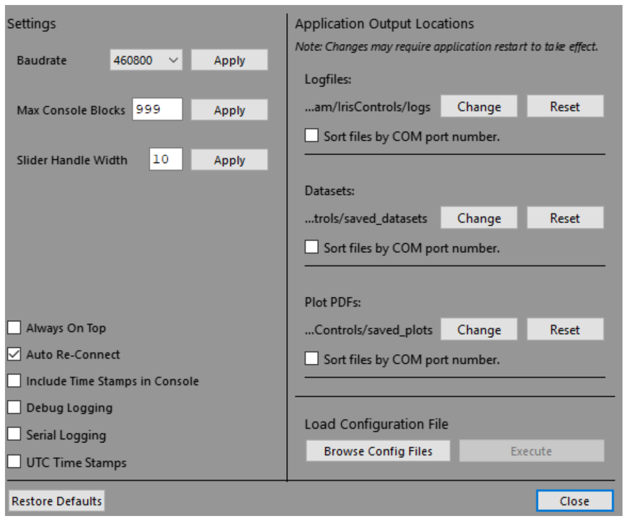


Figure 18: Changing Log File Location Through Settings

The frequency of data logging can be changed by typing "log [time ms]" into the IrisControls console.

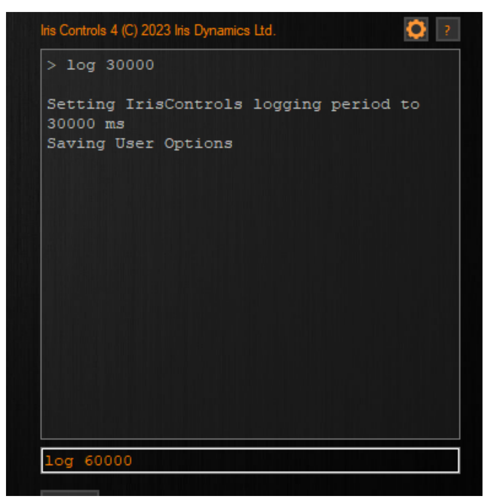


Figure 19: Changing Logging Period Through Console

ORCA Registers

System data like sensor readings, user settings, and controller setpoints are stored in and can be accessed from a collection of registers. Registers can be read and, in some cases, modified by the Modbus or GUI interface.

For the list and description of available registers on ORCA motors, see ORCA Memory Map.

For information on accessing registers from the Modbus interface, see the ORCA motors Modbus User Guide.

Control Registers

ORCA registers labeled CTRL_REG_n are registers that are used to initiate various processes in the ORCA. Some control registers use arrays of bits to have multiple functions available in each register. Below are tables describing the bit positions and functions for each control register.

After a control register has parsed an input and performed the function, the read value of the register will return to 0.

CTRL_REG_0

Control register 0 is used for basic system functions.

Bit Position	Write Value	Name	Description
0	1	reset	Full reset of the ORCA.
1	2	clear errors	Clear all active and latched errors.
2	4	zero position	Rest zero position to current shaft position.
3	8	invert position	Change the direction of positive movement.

CTRL_REG_1

Control register 1 is used to apply newly-written PID gains to position and current controllers.

Bit Position	Write Value	Name	Description
0 - 9	-	RESERVED	
10	1024	position controller gain set flag	Apply PID gains to the position controller.
11	2048	current controller gain set flag	Apply PID gains to the current controller.
12 - 15	-	RESERVED	

CTRL_REG_2

Control register 2 is used for functions that save register data to nonvolatile memory.

Bit Position	Write Value	Name	Description
0 - 4	-	RESERVED	
5	32	tuning save	Save the tuning section of registers to flash memory.
6	64	user opt save	Save the user options section of registers to flash memory.
7	128	motion config save	Save the kinematic section of registers to flash memory.
8	256	IOSH save	Save the IO SmartHub section of registers to flash memory.
9	512	haptic config save	Save the haptic section of registers to flash memory.

CTRL_REG_3

Control register 3 is used to change the mode of operation of the ORCA.

Bit Position	Name	Description
0-15	mode	The desired mode number can be written to this register. If the number is a valid mode, the ORCA will change modes. Each mode is detailed in the Modes of Operation section.

CTRL_REG_4

Control register 4 is used to set defaults for sections of registers that are saved to permanent memory. Note that the sections must still be saved through control register 2 to keep these values through power cycles.

Bit Position	Write Value	Name	Description
1	2	tuning defaults	Set the tuning section of registers to their default values.
2	4	motor user options defaults	Set the motor settings in the user options section of registers to their default values. [1]
3	8	Modbus user options defaults	Set the Modbus settings in the user options section of registers to their default values. [1]
4	16	kinematic defaults	Set the kinematic section of registers to their default values.
5	32	haptic defaults	Set the haptic section of registers to their default values.
6	64	IOSH defaults	Set the IO SmartHub section of registers to their default values.
7	128	PWM defaults	Set the pulse width control section of registers to their default values. [1]

Notes:

1. The user options section of registers contains motor settings, Modbus settings and PWM settings in the same block of flash memory. Therefore they are saved together with 1 bit in CTRL_REG_2, but can have their defaults restored separately.

Sensors

Several solid-state, contactless sensors are embedded in each ORCA motor.

Sensor	Register	Units	Required Voltage
Force	FORCE, FORCE_H (double wide signed 32-bit integer)	Millinewtons	>10 V
Position	SHAFT_POS_UM, SHAFT_POSITION_H (double wide signed 32-bit integer)	Micrometres	Any
Power	POWER	Watts	>10 V
Board Temperature	BOARD_TEMP	Degrees Celsius	Any
Coil Temperature	COIL_TEMP	Degrees Celsius	>10 V

Force Sensor

Forces resulting from current in the motor windings, whether caused by the motor drivers, or by currents induced due to shaft movement, is calculated, and reported with low latency and high bandwidth.

Position Sensor

The position of the shaft is measured and reported with low latency and high bandwidth. The reported position can be set to zero by writing to the “Zero Position” bit of “Control Register 0.” The ORCA motor will take its current shaft position as zero at startup every time the motor is power cycled as the absolute position will be lost if the motor loses 5V.

Depending on the configuration, the motor will either continuously zero the position value when moving in the negative direction or negative values can be valid. The default direction of positive travel is the shaft moving away from the cable side. The positive position direction can also be inverted by writing to the invert position bit flag in Control Register 0.

The shaft position is obtained by measuring and integrating the ‘shaft alignment,’ or the position of the shaft’s magnets with respect to the stator.

The shaft alignment is absolute in nature (persists throughout power cycles); however, it repeats every shaft period interval. This interval is listed on the motor’s datasheet as “Shaft Magnetic Period.”

The position sensor is active in all modes of operation.

Power Sensor

Power being consumed or generated is measured and reported with low latency and high bandwidth. This power is a measure of the heat being generated by the stators and is not necessarily an accurate reflection of the power drawn from the supply, especially when the shaft is moving quickly. The power sensor reflects the rate at which the stator will be increasing in temperature.

Temperature Sensors

The temperature of the power driver and the stator windings are measured and reported to the ORCA registers.
The temperature of the electronics can be obtained by reading the BOARD_TEMP register.
The stator coil temperature can be obtained by reading the COIL_TEMP register. This value represents the temperature of the windings and is a calculated estimate based on thermocouple reading and power draw.
Each of the temperature readings has an individually configurable maximum value before an error is thrown. See User Configurations.

Controllers

The integrated logic included in ORCA motors carries out several feedforward and feedback controllers with very low latency loops that provide high performance motion and force control without a lot of tuning and setup required.

Force Controller

Internal control loops control the phase currents in the motor to achieve a commanded force, compensating for shaft position, speed, winding temperature, and supply voltage. There are no user tuning parameters for the force controller, and calibration is typically done during manufacturing.

The force controller acts on commands from the position controller when the motor is in Position Mode (3) or Kinematic Mode (5), acts on commands from Modbus when in Force Mode (2), and commands from the haptic controller when in Haptic Mode (4).

Position Controller

A common use of linear motors, especially in kinematic robotic applications, is to move to and hold position in the presence of dynamic disturbances. The integrated PID position controller accomplishes this by using the position sensor as feedback and generating a force command which will realize the position setpoint effectively. Position setpoint means the position target commanded via Modbus or from the kinematic controller.

When tuning the PID position controller in a new system, it is always recommended to start with as low a force saturation level as possible.

General Tuning Guide

The easiest way to tune the position controller is using the ORCA GUI. The "Position" tab allows easy editing of the gains, enabling and disabling of the controller, and a real-time view of the target position, the actual position, and the controller force output.

There is no shortage of PID tuning guides available on the internet and tuning of the integrated PID position controller follows most traditional PID controllers. The reader is encouraged to research the term "PID position controller" if the concepts are unfamiliar prior to using this feature.

Specific optimal tuning will depend on the nature of the load and any disturbances experienced. Default tuning values are based on standard shaft lengths and default kinematic motion profile. Different speed profiles and load may affect tuning requirements.

Force Saturation Level

Units: millinewtons (mN)

This setting limits the maximum force the position controller will generate in either direction.

Proportional Action

Units: millinewtons-per-micrometer (mN / µm)

The 'P' in PID is "proportional gain" and will act like a spring; the further the shaft is from its setpoint, the more force the controller will apply. Higher P-gain results in a stronger spring effect.
ORCA motors have fast feedback loops which enable relatively high proportional gains. In systems where some steady-state error may be acceptable, it is possible to achieve good motion control using only the proportional action.
In precise systems where small errors are unacceptable, Integral gain should be added after Proportional gain to remove steady state errors.
The proportional gain register value is scaled up by a factor of 64 for increased resolution.

Integral Action

Units: millinewton-seconds-per-micrometer (mN·s / µm)

Integral action takes any small errors in position and accumulates action to correct them over time. For example, if an error of 1 millimeter is present for 1 second, an I-gain of 1000 will generate 1 Newton of corrective force. A second later this force will be 2 N. This will continue until enough force is accumulated to remove the 1 mm of error.
Integral gain is inherently unstable and must be used in combination with proportional gain to prevent oscillation.
Higher I-gain will remove steady state errors faster but could cause instability, especially when the position setpoint is moved rapidly, or sharp disturbances (*e.g.,* kicking the shaft) are encountered.
The integral gain register value is scaled up by a factor of 64 for increased resolution.

Derivative Action (Velocity)

Units: millinewton-millimeters-per-second (mN·mm / s)

In this context, derivative action is a force acting against the speed of the shaft. The action feels like a damper on the shaft and can prevent it from reaching high speeds.

This action is often not required, especially when fast transient response is desired. However, D-gain will prevent the shaft from moving too fast and can improve stability and reduce overshoot.
It is safe to use derivative action alone if a brake force is required. The derivative gain register value is scaled up by a factor of 2 for increased resolution.

Derivative Action (Error)

Units: millinewton-millimeters-per-second (mN·mm / s)

As an alternative to derivative action based on velocity, a derivative action based on error is also available. The end effect of both derivative actions is quite similar and can be used in similar situations.
The derivative action acts proportional to the change in error between the setpoint and measured position.
The derivative gain register value is scaled up by a factor of 2 for increased resolution.

Auto Zeroing

The auto zeroing algorithm is intended to bring the shaft to the fully retracted position until it reaches a hard stop, and zero the position reading at that point. This process can be initiated by commanding the ORCA motor to enter Auto Zero Mode (55). Auto Zeroing can also be initiated through the GUI, or configured to start when the ORCA motor boots up.

Configuration

Auto Zeroing mode must be enabled through selecting a valid value in the ZERO_MODE register to use the Auto Zeroing routine. The Auto Zeroing process has configuration options that may need to be tweaked for different applications. All relevant registers and their values are described in the Register Descriptions table. If configuration through the IrisControls GUI is preferred, the Configuring and Using Auto Zeroing From the IrisControls GUI provides an example.

Zero Mode

The valid ZERO_MODE register settings for Auto Zeroing are

- **Auto Zero Enabled**
  Auto Zeroing mode is enabled, but is only started through the “Start Auto Zero” button or writing to Control Register 3 to place the motor in Auto Zeroing mode.

- **Auto Zero on Boot**
  Auto Zeroing mode is enabled and on startup/reboot the motor will automatically perform the Auto Zeroing routine and then move into the specified exit mode.
- **IOSH Auto Zeroing**
  When a connected IO SmartHub's enable signal is asserted, the Auto Zeroing routine will be performed before entering the configured IO SmartHub control mode.

Maximum Force

- The absolute maximum force allowed to move the shaft during auto zeroing.
- Units: Newtons
- Configurable through the AUTO_ZERO_FORCE_N register or the position page on the IrisControls GUI.

Shaft Speed

- The speed that the shaft will move at during the Auto Zeroing routine, in millimeters per second.

Exit Mode

- The mode of operation which should be entered when Auto Zeroing is successfully completed.
- Must be one of the following ORCA modes of operation or will default to mode 1 (Sleep).
  - Sleep Mode (1)
  - Force Mode (2)
  - Position Mode (3)
  - Haptic Mode (4)
  - Kinematic Mode (5)
  - Pulse Width Mode (11)
- Configurable through the AUTO_ZERO_EXIT_MODE register or the dropdown on the position page on the IrisControls GUI.

Enable on Startup

- The auto zeroing algorithm can be configured to run on motor startup through the ZERO_MODE register, where Auto Zero on Boot (3) must be the selected option.

Register Descriptions

Modbus Register Name	Description	Valid Inputs
ZERO_MODE	Motor zeroing setting.	Negative Zeroing (0) Manual Zeroing (1) Auto Zero Enabled (2) Auto Zero on Boot (3) IOSH Auto Zeroing (4)
AUTO_ZERO_FORCE_N	The maximum force the auto-zeroing algorithm will use to move the motor. In Newtons.	0-800
AUTO_ZERO_EXIT_MODE	Mode to enter after completing auto-zeroing.	Sleep (1) Force (2) Position (3) Haptic (4) Kinematic (5) Pulse Width (11)
AUTO_ZERO_SPEED_MMPS	Shaft speed during Auto Zeroing routine.	1-500

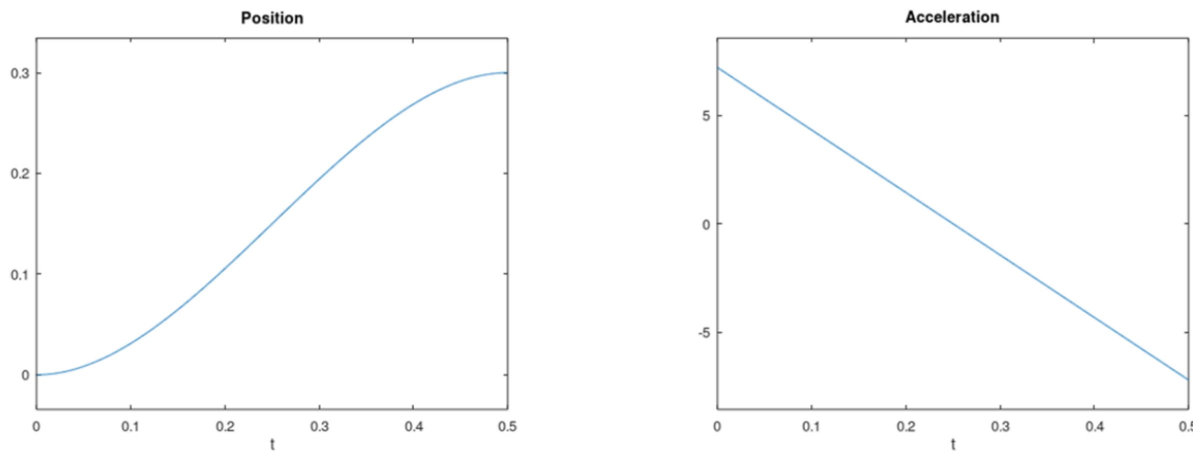
All configuration options can be saved by saving the user options section of flash memory through CTRL_REG_2.

Kinematic Controller

ORCA motors are equipped with a kinematic controller that provides configurable motion profiles which allow movement to a shaft position over a specified time, while respecting the chosen kinematic constraints. The motion profiles are fed to the position controller as position targets.

Types of kinematic motions available on the motor include:

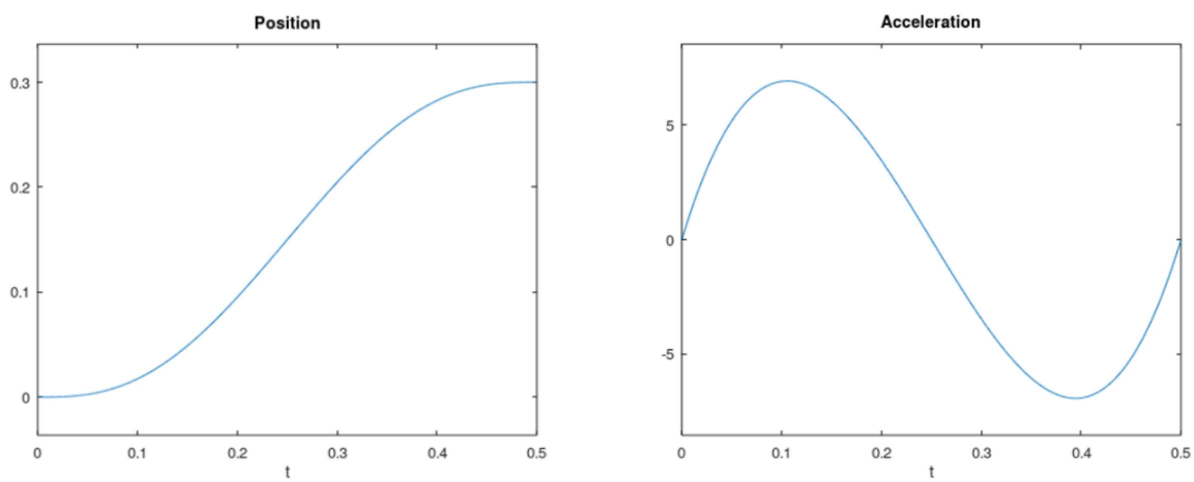
- Minimum power (linear acceleration).



Position vs Time with Linear Acceleration *Acceleration vs Time with Linear Acceleration*

Figure 20: Position and Acceleration of Type 0 Kinematic Motion

- Maximum smoothness (minimum jerk).



Position vs Time with Minimum Jerk *Acceleration vs Time with Minimum Jerk*

Figure 21: Position and Acceleration of Type 1 kinematic motion

Up to 32 motions can be saved to a single ORCA motor. Motions are configured either from the Kinematic GUI page, or through direct writes to the ORCA memory map. Motions can be initiated either by Modbus messages, or from digital triggers through the IO SmartHub.

Software Triggering

Motions can be initiated by writing the desired motion ID number to the KIN_SW_TRIG[MOTION_ID] field. This register will be set to 0x1000 initially and will return to 0x1000 after processing a software trigger request. Software triggers of motions with a number higher than 32 will be ignored.

Register	Bits 15-6	Bits 4-0
KIN_SW_TRIG	Reserved	MOTION_ID

Configuring a Motion

Individual kinematic motions consist of 5 variables:

- Motion target position (μm).
- Motion time period (ms).
- Motion type (Minimum power or maximum smoothness).
- Auto-start delay (Delay in ms before triggering next motion if auto-start is enabled).
- Next ID (motion ID that will be queued after this motion finishes).
- Auto-start next (Start next ID motion after this one).

The kinematic GUI page provides an interface for programming each of the motions, or the motions can be written directly to the memory map. A single motion configuration takes up 6 consecutive registers in the memory map, totaling 192 registers allocated for the motions (KIN_MOTION_0 to KIN_MOTION_31). The layout of each set of these registers is shown in table 4.

Offset from KIN_MOTION_#	Description
0	Position Target (Low 16 bits)
1	Position Target (High 16 bits)
2	Settling Time (Low 16 bits)
3	Settling Time (High 16 bits)
4	Auto-start Delay
5	Next ID, Type, and Auto-Start Next

The motion type and auto-start next options occupy the same register with the following structure.

Register	Bits 15-8	Bits 7-3	Bit 2-1	Bit 0

KIN_MOTION_# + 5	Reserved	Next ID	Type 00 = min. power 01 = min. jerk	Auto-Start Next

The Type field is interpreted as a 2-bit number indicating motion type.
The Auto-Start Next bit is 1 when the chain feature is enabled and 0 when it is disabled.

Kinematic Status

The KINEMATIC_STATUS registers is used to indicate the state of the kinematic controller to indicate the active motion ID and whether a motion is in progress or finished,

Register	Bit 15	Bits 14-0
KINEMATIC_STATUS	Running Flag	Active ID

Haptic Controller

ORCA motors are equipped with a haptic controller that provides a series of force effects which can be combined to create force targets based on shaft position, speed, and acceleration, in addition to periodic force options. The sum of the haptic effects is sent as the force target to the force controller.

Effects

The status of each effect is contained in the HAPTIC_STATUS register. Each bit position represents whether a single effect is enabled or disabled. A 1 represents an active effect that will produce force, and a 0 represents a disabled effect. Effects may be configured as well as toggled on and off while the motor is in Haptic Mode for dynamic effects. Note that the motor must be in Haptic Mode for any of the effects to produce force.

Register	Bits 15-8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
HAPTIC_STATUS	-	Osc B	Osc A	Inertia	Damper	Spring C	Spring B	Spring A	Constant

Below is a description of each type of effect and their configurations.

- Constant: This effect applies a constant force to the haptic controller. This is also a useful effect to stream updates through Modbus commands to implement unique, dynamic effects.
  - Force (mN): Constant force.
  - Filter Strength (0-10000): Strength of low pass filter applied to the constant force input.
- Spring: The spring effects apply a force that linearly increases as the shaft is moved away from the center location.
  - Gain (N/mm): Rate at which the force will increase proportional to the change in position.
  - Center (μm): Location of spring center.
  - Dead Zone (mm): Indicates a distance around either side of the center in which the spring effect doesn't apply.
  - Saturation (N): The maximum force that will be output by this spring effect, if set to 0 no force limit will exist.
  - Coupling (+/-): Type of spring behaviour.
    - +/-: Spring will act in both directions.
    - +: Spring forces only applied when moving the shaft in the positive direction beyond the center.

        - -: Spring forces only applied when moving the shaft in the negative direction beyond the center.

- Damper: The damper effect applies a force to reduce the speed of the motor's movement.
    - Gain (N·s/mm): Rate at which the force increases proportional to the speed of the motor.
- Inertia: The inertia effect applies a force that attempts to reduce the acceleration of the motor. This makes it both more difficult to get the shaft in motion, but also keep the shaft in motion once moved.
    - Gain (N·s^2/mm): Rate at which the force increases proportional to the acceleration of the motor.

- Oscillator: The oscillator effects apply periodic forces in the shape of various waveforms.
    - Amplitude (N): The peak force of the waveform, oscillators are centered on 0 N of force so the force will oscillate between negative amplitude and positive amplitude values. Note: adding a constant force effect with an oscillator will change the force around which the waveform oscillates.
    - Frequency (dHz): Frequency of the oscillation in decihertz. (1 dHz = 0.1 Hz)
    - Duty (%): This field only applies to 'Pulse' waveforms, the duty indicates what percentage of the waveform should be spent in a 'high' state (applying positive force) with the remaining percent being in the 'low' state.
    - Type: The waveform pattern can be selected from the dropdown.
        - Pulse: Square wave that alternates between the positive amplitude value and the negative amplitude value. The duty cycle is used to determine the portion of the period each direction of force is active.
        - Sine: Sinusoidal waveform.
        - Triangle: Linearly increases force, then reverses and linearly decreases force.
        - Sawtooth: Linearly increases from negative amplitude value to positive amplitude value over entire period, then sharply go back to negative amplitude value

Configuring Effects

Effects are individually configured through sets of registers shown in a table below. Note: the units for each effect configuration, as well as the scaling factors applied to each gain.

Effect	Parameter	Register
Constant Force	Force (mN)	CONSTANT_FORCE_MN
		CONSTANT_FORCE_MN_H
	Filter Strength (0-10000)	CONST_FORCE_FILTER
Spring n	Gain (256·N/mm)	Sn_GAIN_N_MM
	Center (mm)	Sn_CENTER_UM
		Sn_CENTER_UM_H
	Coupling	Sn_COUPLING
	Dead Zone (mm)	Sn_DEAD_ZONE_MM
	Saturation (N)	Sn_FORCE_SAT_N
Damper	Gain (4·N/mm)	D0_GAIN_NS_MM
Inertia	Gain (64·N·s^2/mm)	I0_GAIN_NS2_MM
Oscillator n	Amplitude (N)	On_GAIN_N
	Frequency (dHz)	On_FREQ_DHZ
	Duty (%)	On_DUTY
	Type	On_TYPE

Pulse Width Control

With the addition of a PWM (Pulse Width Modulation) Interface board (available for purchase as a motor accessory), the motor position can be controlled with a typical RC servo pulse width signal. While typical servos control their position between 0 and 180 degrees, the motor will control its position between any two configurable values.

With Auto Zeroing configured to run on power up and to go directly into Pulse Width Mode, the motor operates analogously to a servo motor.

Note that Modbus communications will be entirely disabled in Pulse Width Mode (11).

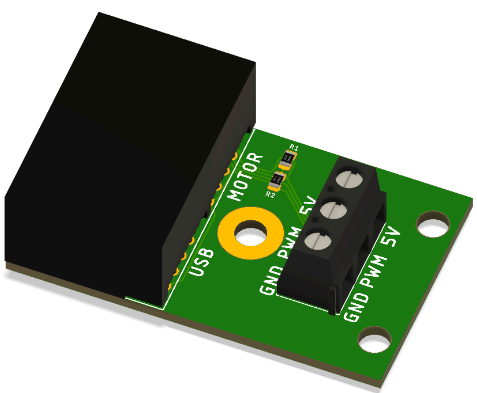
Supported Input Signal Parameters

Supported Input Frequency: 40 Hz to 400 Hz
Supported Pulse Widths:

- 1.0ms to 2.0ms
- 0.5ms to 2.5ms

Hardware Interface

In Pulse Width mode, the Modbus interface is repurposed for PWM input. To facilitate this, Iris Dynamics offers an Interface PCB to easily connect PWM signals directly to the motors data cable.



PWM Input

Input Voltage Range: 0 to 5.5 V
Input Voltage Low: -0.1 to 0.5 V
Input Voltage High: 4.5 to 5.1 V
Input Current (@5 V): +2 mA
Input Current (@0 V): -2 mA
ESD Rating: IEC 61000-4-2 Level 4

5V Input

The 5V input is optional and can be used to provide emergency logic power to the motor, in case the motor's power supply is disabled. If the motor's power supply is turned off or disconnected and there is a source of 5 V applied to the PWM interface, the logic will continue to function. This is beneficial as the motor will continue to track its position and not require a zeroing routine when power is reapplied.

Maximum Voltage Input: **5.2 V**
Minimum Voltage Input to retain logic: **4.0 V**

Ground

This ground input on the interface must be connected to the PWM controller ground. If 5V is applied to the motor for backup logic power, the 5V source and the PWM controller should all connect to the same ground as closely as possible.

Motor Connector

The motor data cable should be plugged directly into the RJ45 port labeled "MOTOR" on the interface PCB. Note that no splitter from the ORCA Starter kit should be connected, the interface PCB essentially replaces the splitter.

IrisControls Connection

The RJ45 port labeled "USB" on the interface board can accept the USB-to-RS485 used to connect a motor to IrisControls via USB. This may be used simultaneously with the PWM input, or may be omitted.

Configuring Pulse Width Control

There are 5 configurable parameters for Pulse Width mode, which can be set directly through Modbus registers, or the GUI.

- **Timeout Period (ms):** The amount of time between pulses before a Comms Timeout (2048) error is thrown.
- **Filter Time Constant (ms):** Low-pass filter time constant. Used to smooth the position output from low-frequency pulse inputs.
- **Minimum Position (um):** Position target limit when the minimum pulse width is detected.
- **Maximum Position (um):** Position target limit when the maximum pulse width is detected.
- **Servo Control Type:** Sets the expected pulse width to 180 degree or 270 degree servo mode. (1.0 ms to 2.0ms and 0.5 ms to 2.5 ms respectively).

Below is a list of the registers used to control those parameters. Values can be configured by writing to these registers, and saving to permanent memory by setting the CONTROL_REG_2 [usr_opt_save_flag].

Register	Description
PWM_TIMEOUT_MS	PWM timeout period in milliseconds.
PWM_TIME_CONST_MS	Low-pass filter time constant in milliseconds.
PWM_MIN_POS	Minimum position in micrometers. Lower 2 bytes.
PWM_MIN_POS_H	Minimum position in micrometers. Upper 2 bytes.
PWM_MAX_POS	Maximum position in micrometers. Lower 2 bytes.
PWM_MAX_POS_H	Maximum position in micrometers. Upper 2 bytes.

PWM_SERVO_TYPE	Servo control type. Valid values are:
	• 0
	  ○ 180 Degree Mode (1.0ms - 2.0ms)
	• 1
	  ○ 270 Degree Mode (0.5ms - 2.5ms)

Automatic Startup

In conjunction with the Auto Zeroing mode, pulse width mode can be configured to automatically begin on motor power up. Set the Auto Zero exit mode to Pulse Width, and enable the Auto Zero on Boot option. Save the settings to your motor.

On startup, the motor will then perform an Auto Zero routine, and enter Pulse Width mode on completion.

Timeout Behaviour

If a time longer than the period configured in PWM_TIMEOUT_MS occurs between incoming pulses, the PWM mode will enter a timeout state. The motor will exhibit a Comms Timeout error (2048) and put the position controller into damping mode. The motor will stay in this damping only configuration until another pulse arrives, at which point the error will be cleared and the position controller will initiate a softstart and return to the command position.

User Configurations

Some properties of the motor can be configured to suit the application. In all cases, this configuration is done by writing to various registers and, when satisfied, saving those settings to permanent memory by writing the CONTROL_REG_2 [usr_opt_save_flag].

Motor Performance Limits

Parameter	User Register	Default	Units
Maximum Board Temperature	USER_MAX_TEMP	70	Degrees Celsius
Maximum Coil Temperature	USER_MAX_COIL_TEMP	120	Degrees Celsius
Temperature Hysteresis	TEMP_ERR_HYSTERESIS	5	Degrees Celsius
Maximum Force	USER_MAX_FORCE	0	Millinewtons
Maximum Power	USER_MAX_POWER	0	Watts
Communication Timeout	USER_COMMS_TIMEOUT	500	Milliseconds

Maximum Board Temperature

The maximum board temperature at which point the Temperature Exceeded error (64) will be thrown, disabling the motor force output. The default (and maximum value) is 70, but this can be lowered by writing a number between 1 and 69 to the USER_MAX_COIL_TEMP register.

Maximum Coil Temperature

The maximum coil temperature at which point the Temperature Exceeded error (64) will be thrown, disabling the motor force output. The default (and maximum value) is 120, but this can be lowered by writing a number between 1 and 119 to the USER_MAX_COIL_TEMP register.

Temperature Hysteresis

Board and coil temperatures must fall below their limits minus this value before Temperature Exceeded errors (64) can be cleared. By default, this threshold is 5 degrees. For example:

- USER_MAX_TEMP = 50
- USER_MAX_ COIL_TEMP = 100
- TEMP_ERR_HYSTERESIS = 5
- Temperature Exceeded error (64) will not be cleared until:
    - BOARD_TEMP < 50 - 5 = **45**
    - COIL_TEMP < 100 - 5 = **95**

This way, the hysteresis introduces a temperature gap that reduces the chance of thermally stressing the motor.

Maximum Force

The motor has no default maximum force threshold, but one can be set which will limit forces beyond this level, and throw a Force Exceeded error (128). Setting the register to zero disables this threshold.

Maximum Power

If the power burned in the motor exceeds this or the device's default threshold, the drivers will be disabled, preventing power draw. Setting this register to zero or higher than the default setting will result in the motor only powering down when unsafe levels are reached. See the Power Exceeded section for more information.

Communication Timeout

By default, Force Mode (2), Position Mode (3), and Haptic Mode (4) will timeout when serial communications are interrupted for longer than the timeout period. This timeout period can be adjusted with this user setting.

Input and Output Filters

Input and output filters are available for both force and position values. These are infinite impulse response (IIR) filters. In this case 0 represents no filter and 9999 is maximum filtering.

User Register	Description	Valid Range
FORCE_FILT	Filter on motor's sensed force.	0-9999
POS_FILT	Filter on motor's sensed position.	0-9999
MB_FORCE_FILT	Filter on target force values received over Modbus.	0-9999
MB_POS_FILT	Filter on target position values received over Modbus.	0-9999

Modbus Communications

The configurations that govern Modbus communication can be adjusted. This can be used to match a certain Modbus client configuration or to speed up message frame rate.

Parameter	User Register	Default Value	Valid Range
Baudrate	USR_MB_BAUD_LO USR_MB_BAUD_HI	19200	9600 - 1000000
Interframe Delay	USR_MB_DELAY	2000 μs	0 - 10000
Server Address	USR_MB_ADDR	1	1 - 65535

Position Zeroing Mode

The value in the ZERO_MODE register determines the method of position zeroing the motor will use.

Mode Name	Description	Register Value
Zero Negative Positions	Moving the shaft into the negative position will reset the zero position to that current position.	0
Manual Zeroing	Position will only be zeroed at start up or when zero position is commanded through the GUI button or over Modbus.	1
Auto Zero Enabled	This mode acts the same as Manual Zeroing, but also enables the use of the Auto Zeroing routine.	2
Auto Zero on Boot	This mode acts the same as Auto Zero Enabled, but the Auto Zeroing routine is initiated once automatically when the motors boots up for the first time.	3
IOSH Auto Zeroing	This acts like Auto Zero Enabled, but allows the IOSH enable signal to start the Auto Zeroing routine.	4

More information on the Auto Zeroing feature can be found in the Auto Zeroing section.

Errors

The motor will generate error codes when a user setting, or a device limit is reached or exceeded. Depending on the error, certain features will not be available until the error is cleared.
Some errors, like temperature and power, which trigger when a parameter exceeds a device maximum can be configured using motor registers to trigger earlier as the application requires.
Other errors, like force, will never trigger unless the registers are configured.

Active and Latched Error Registers

Errors which are currently in effect are reflected in the ERROR_0 register. Any time an error is asserted, its flag will also appear in the ERROR_1 register and will remain asserted until cleared by a write to the CONTROL_REG_0 [clear_errors_flag]. For example, if a temperature error becomes active, it will show up in both registers. Once the temperature falls below the error level and the motor is brought to Sleep Mode (1), the error will be cleared from ERROR_0 but not from ERROR_1.

Error	Mask	Trigger Level Registers	Modules disabled	Cleared By
Configuration Errors	1 (0x0001)	-	Position, Force	Reset
Force Clipping	32 (0x0020)	-	-	Automatically
Temperature Exceeded	64 (0x0040)	USER_MAX_TEMP USER_MAX_COIL_TEMP	Position, Force	Sleep Mode
Force Exceeded	128 (0x0080)	USER_MAX_FORCE	-	Automatically
Power Exceeded	256 (0x0100)	USER_MAX_POWER	Position, Force	Sleep Mode
Shaft Image Failed	512 (0x0200)	-	Position, Force	Sleep Mode + Insert Shaft
Voltage Invalid	1024 (0x0400)	-	Position, Force	Providing a valid voltage source
Comms Timeout	2048 (0x0800)	USER_COMMS_TIMEOUT	Position*, Force	Sleep Mode
Auto Zero Failed	8192 (0x2000)	-	Position*	Sleep Mode

*These errors do not completely disable control in position mode, but force the tuning to be purely damped for safe recovery. The DV gain value that will be applied can be set through the SAFETY_DGAIN register.

Configuration Errors (1)

These errors indicate calibrations or settings have not been done or have been made invalid. These errors can be cleared by running the appropriate calibration routine, followed by committing the valid results to permanent memory.

Force Clipping (32)

When the force controller is commanded to output forces that would require a phase to exceed its rated current, the Force Clipping error is asserted. This error has no effect on operation except to inform the user that linear force output has been compromised. The error self clears as soon as the condition is removed.

Temperature Exceeded (64)

When the temperature of the stator windings or of the motor driver exceeds the device or user-set maximum, a Temperature Exceeded error is asserted. This error is cleared when entering Sleep Mode (1) or when the CONTROL_REG_0 [clear_errors_flag] is set ONLY if the board and coil temperatures are both below their configured maximums. A hysteresis option is also available to prevent temperature errors from being cleared until they fall below the configured limits by a number of degrees. See Maximum Temperature.

Force Exceeded (128)

When the measured force output of the motor exceeds the user-set force limit, a Force Exceeded error is asserted. This error is cleared when the force falls below the measured threshold.

The force controller will attempt to control the force output to minimize force output beyond the user setting.

Power Exceeded (256)

When the power burned in the stator exceeds the device or user-set maximum value, a Power Exceeded error is asserted. This error is cleared by commanding the motor into Sleep Mode (1).

If this error is experienced, either the maximum power user setting can be increased, or the maximum force user setting should be decreased.
If the position controller (*i.e., Position* Mode (3)) is causing this error, the saturation level can also be decreased to prevent this error.

Shaft Image Failed (512)

The shaft image is a calculation of the shaft's alignment with the windings based on magnetic field measurements. If this image is detected to be invalid, the shaft might not be inserted, it might be an invalid shaft for the device, or the device may require calibration. This error is cleared by resolving the underlying issue and commanding the motor into Sleep Mode (1).

Voltage Invalid (1024)

When the supply voltage of the motor is measured to be less than 10V, a Voltage Invalid error is asserted. This error will disable the current sensors, resulting in force and power calculations being disabled as well. The driver temperature sensor is also invalid in this configuration, so it is disabled. This error is cleared by providing a supply voltage of greater than 10V.

Communications Timeout (2048)

When in Force Mode (2), Position Mode (3), or Haptic Mode (4), a steady stream of communications must be successfully received to avoid this error. If the timeout specified in the USER_COMMS_TIMEOUT register elapsed between successful messages, the Comms Timeout error is asserted. When this error is active, the motor will only produce a damping force.
In Pulse Width Mode (11), if the motor does not receive a new pulse for the amount of time set in PWM_TIMEOUT_MS, this error will also be set. The error will be cleared when another pulse is received.
Users can adjust the communications timeout by writing a non-zero value to the USER_COMMS_TIMEOUT register. This register has units of milliseconds (ms). This can also be configured on the Modbus Page of the GUI.

Auto Zero Failed (8192)

If the auto zeroing process is interrupted in any way, this error will occur. Most commonly, this means another error occurred. Otherwise, the auto zeroing process can time out if the shaft is detected to be still moving after a long enough period of time. When this error is active, the motor will only produce a damping force.

APPENDIX : ORCA Memory Map

All registers can be assumed to be unsigned 16 bit integers unless otherwise specified.

Address	Name	Width	Description
0	CTRL_REG_0	1	Control Register 0
1	CTRL_REG_1	1	Control Register 1
2	CTRL_REG_2	1	Control Register 2
3	CTRL_REG_3	1	Control Register 3
4	CTRL_REG_4	1	Control Register 4
5 - 8	Reserved		
9	KIN_SW_TRIGGER	1	Software trigger for initiating kinematic movements over Modbus.
10 - 27	Reserved		
28	FORCE_CMD	1	Commanded actuator output force in millinewtons. Lower 2 bytes.
29	FORCE_CMD_H	1	Commanded actuator output force in millinewtons. Upper 2 bytes.
30	POS_CMD	1	Commanded actuator position in micrometers. Lower 2 bytes.
31	POS_CMD_H	1	Commanded actuator position in micrometers. Upper 2 bytes.
32 - 128	Reserved		
129	CC_PGAIN	1	Current controller proportional gain.
130	CC_IGAIN	1	Current controller integral gain.

131	CC_FGAIN	1	Current controller forward gain.
132	CC_MAX_DUTY	1	Current controller maximum duty cycle.
133	PC_PGAIN	1	Position controller proportional gain.
134	PC_IGAIN	1	Position controller integral gain.
135	PC_DVGAIN	1	Position controller velocity gain.
136	PC_DEGAIN	1	Position controller error derivative gain.
137	PC_FSATU	1	Position controller maximum force output. Lower 2 bytes.
138	PC_FSATU_H	1	Position controller maximum force output. Upper 2 bytes.
139	USER_MAX_TEMP	1	User configurable maximum motor temperature before over temperature error in degrees Celsius.
140	USER_MAX_FORCE	1	User configurable maximum force output in millinewtons. Lower 2 bytes.
141	USER_MAX_FORCE_H	1	User configurable maximum force output in millinewtons. Upper 2 bytes.
142	USER_MAX_POWER	1	User configurable maximum power burn in watts.
143	SAFETY_DGAIN	1	Speed damping gain value used when communications are interrupted.
144 - 146	Reserved		

147	USER_MAX_COIL_TEMP	1	User configurable maximum coil temperature before over temperature error in degrees Celsius.
148	TEMP_ERR_HYSTERESIS	1	User configurable temp error clearing hysteresis. Errors cannot be cleared until temps are < limit - hysteresis.
149	Reserved		
150	PC_SOFTSTART_PERIOD	1	Time in ms over which the position controller max force output will ramp from zero any time a mode of operation in which the position controller used is entered
151	Reserved		
152	POS_SIGN	1	Determines which direction of shaft movement is the zero direction.
153 - 161	Reserved		
162	LOG_PERIOD	1	Period between data log entries.
163	USER_COMMS_TIMEOUT	1	Time between successful force or position commands before a communications error occurs. In milliseconds.
164	USR_MB_BAUD_LO	1	Default Modbus baudrate low 16 bits. Leaving this register at 0 will use the system default of 19200 bps.
165	USR_MB_BAUD_HI	1	Default Modbus baudrate high 16 bits. Leaving this register at 0 will use the system default of 19200 bps.

166	FORCE_FILT	1	Force output IIR filter alpha value. Maps 0-65535 to alpha values of 0 to 1.
167	POS_FILT	1	Position output IIR filter alpha value. Maps 0-65535 to alpha values of 0 to 1.
168	USR_MB_DELAY	1	Default Modbus interframe delay in microseconds. Default value is 2000 us.
169	USR_MB_ADDR	1	Start up Modbus address. Default value is 1
170	Reserved		
171	ZERO_MODE	1	Selects the type of zeroing the motor should use.
172	AUTO_ZERO_FORCE_N	1	The maximum force the auto-zeroing algorithm will use to move the motor. In Newtons.
173	AUTO_ZERO_EXIT_MODE	1	Mode to enter after completing auto-zeroing.
174	MB_RS485_MODE	1	Controls whether RS485 mode is enabled for the Modbus interface.
175	MB_FORCE_FILTER	1	Input filter value for Modbus Force inputs. 0-65535 where 65535 is no filtering.
176	MB_POS_FILTER	1	Input filter value for Modbus Position inputs. 0-65535 where 65535 is no filtering.
177	AUTO_ZERO_SPEED_MMPS	1	Speed that the Auto Zeroing mode should move the shaft at.

178	PWM_TIMEOUT_MS	1	If the PWM input stays high or low for longer than this time in milliseconds, the motor will enter sleep mode.
179	PWM_TIME_CONST_MS	1	Filter time constant in milliseconds applied to position commands from the PWM input.
180	PWM_MIN_POS	1	Max motor extension at or below the minimum pulse width. Lower 2 Bytes
181	PWM_MIN_POS_H	1	Max motor extension at or below the minimum pulse width. Upper 2 Bytes
182	PWM_MAX_POS	1	Max motor extension at or above the maximum pulse width. Lower 2 Bytes
183	PWM_MAX_POS_H	1	Max motor extension at or above the maximum pulse width. Upper 2 Bytes
184	PWM_SERVO_TYPE	1	PWM Servo Type . 180 Degree Servo uses 1ms to 2ms pulses. 270 Degree Servo uses 0.5ms to 2.5ms pulses.
185 - 296	Reserved		
297	H0_QUALITY	1	
298	H1_QUALITY	1	
299	H2_QUALITY	1	
300	H3_QUALITY	1	
301	H4_QUALITY	1	
302	H5_QUALITY	1	

303	H6_QUALITY	1	
304	H7_QUALITY	1	
305 - 312	Reserved		
313	GUI_PERIOD	1	Period between IrisControls GUI communications in milliseconds.
314 - 316	Reserved		
317	MODE_OF_OPERATION	1	Active mode the actuator is currently running in.
318	CALIBRATION_STATUS	1	A value other than zero indicates a calibration routine is in process.
319	KINEMATIC_STATUS	1	Indicates the state of the kinematic controller, and which motion is currently being performed.
320 - 335	Reserved		
336	BOARD_TEMP	1	Temperature of the motor driver in degrees Celsius.
337	Reserved		
338	VDD_FINAL	1	Motor supply voltage in volts.
339 - 341	Reserved		
342	SHAFT_POS_UM	1	Shaft absolute position in micrometers. Lower 2 bytes.
343	SHAFT_POSITION_H	1	Shaft absolute position in micrometers. Upper 2 bytes.

344	SHAFT_SPEED_MMPS	1	Shaft speed in millimeters per second. Lower 2 bytes.
345	SHAFT_SHEED_H	1	Shaft speed in millimeters per second. Upper 2 bytes.
346	SHAFT_ACCEL_MMPSS	1	Shaft acceleration in millimeters per second per second. Lower 2 bytes.
347	SHAFT_ACCEL_H	1	Shaft acceleration in millimeters per second per second. Upper 2 bytes.
348	FORCE	1	Sensed actuator output force in millinewtons. Lower 2 bytes.
349	FORCE_H	1	Sensed actuator output force in millinewtons. Upper 2 bytes.
350	POWER	1	Sensed actuator output power in watts.
351	HBA_CURRENT	1	Sensed phase current for Hbridge A, in milliamps.
352	HBB_CURRENT	1	Sensed phase current for Hbridge B, in milliamps.
353	HBC_CURRENT	1	Sensed phase current for Hbridge C, in milliamps.
354	HBD_CURRENT	1	Sensed phase current for Hbridge D, in milliamps.
355	AVG_POWER	1	Average sensed actuator output power in watts.
356	COIL_TEMP	1	Estimated coil temperature in degrees Celsius.
357	Reserved		

358	SPEED_UMPS	1	Shaft speed in micrometers per second. Lower 2 bytes.
359	SPEED_UMPS_H	1	Shaft speed in micrometers per second. Upper 2 bytes.
360 - 405	Reserved		
406	SERIAL_NUMBER_LOW	1	Actuator serial number. Lower 2 bytes.
407	SERIAL_NUMBER_HIGH	1	Actuator serial number. Upper 2 bytes.
408	MAJOR_VERSION	1	Firmware major version.
409	RELEASE_STATE	1	Firmware minor version.
410	REVISION_NUMBER	1	Firmware revision number.
411	COMMIT_ID_LO	1	Firmware commit ID lower 2 bytes.
412	COMMIT_ID_HI	1	Firmware commit ID upper 2 bytes.
413 - 417	Reserved		
418	STATOR_CONFIG	1	Physical stator configuration type.
419 - 430	Reserved		
431	WARNING	1	Active warning flags.
432	ERROR_0	1	Currently active error flags. Only reflects error conditions that have not been cleared.
433	ERROR_1	1	Latched error flags. Reflects all errors that have occurred since reset.

434 - 463	Reserved		
464	MB_CNT0	1	Return bus message count. Refer to Modbus specification.
465	MB_CNT1	1	Return bus communication error. Refer to Modbus specification.
466	MB_CNT2	1	Return server exception error count. Refer to Modbus specification.
467	MB_CNT3	1	Return server message count. Refer to Modbus specification.
468	MB_CNT4	1	Return server no response count. Refer to Modbus specification.
469	MB_CNT5	1	Return server NAK count. Refer to Modbus specification.
470	MB_CNT6	1	Return server busy count. Refer to Modbus specification.
471	MB_CNT7	1	Return bus character overrun count. Refer to Modbus specification.
472	MB_CNT8	1	Rx line error.
473	MB_CNT9	1	Ignoring state error.
474	MB_CNT10	1	Unexpected interchar.
475	MB_CNT11	1	Unexpected interframe.
476	MB_CNT12	1	Timeout sequence error.
477	MB_CNT13	1	Unexpected emission.
478	MB_CNT14	1	Unexpected reception.
479 - 481	Reserved		

482	MB_BAUD	1	Modbus serial baudrate. Lower 2 bytes.
483	MB_BAUD_H	1	Modbus serial baudrate. Upper 2 bytes.
484	MB_IF_DELAY	1	Modbus interframe delay in microseconds.
485	MB_ADDRESS	1	Modbus server address.
486 - 495	Reserved		
496	MESSAGE_0_SIZE	1	Size of last received Modbus message in bytes.
497	MESSAGE_0	128	-
626 - 640	Reserved		
641	HAPTIC_STATUS	1	Enabled state of effects
642	CONSTANT_FORCE_MN	1	Value of constant force effect in milliwnewtons, low 2 bytes
643	CONSTANT_FORCE_MN_H	1	Value of constant force effect in milliwnewtons, high 2 bytes
644	S0_GAIN_N_MM	1	Strength of spring force
645	S0_CENTER_UM	1	Location of spring center, low 2 bytes
646	S0_CENTER_UM_H	1	Location of spring center, high 2 bytes
647	S0_COUPLING	1	Coupling type, 0 (Both), 1 (Positive), 2 (Negative),
648	S0_DEAD_ZONE_MM	1	Zone from center where no spring effect exists

649	S0_FORCE_SAT_N	1	Maximum force that the spring can output
650	S1_GAIN_N_MM	1	Strength of spring force
651	S1_CENTER_UM	1	Location of spring center, low 2 bytes
652	S1_CENTER_UM_H	1	Location of spring center, high 2 bytes
653	S1_COUPLING	1	Coupling type, 0 (Both), 1 (Positive), 2 (Negative),
654	S1_DEAD_ZONE_MM	1	Zone from center where no spring effect exists
655	S1_FORCE_SAT_N	1	Maximum force that the spring can output
656	S2_GAIN_N_MM	1	Strength of spring force
657	S2_CENTER_UM	1	Location of spring center, low 2 bytes
658	S2_CENTER_UM_H	1	Location of spring center, high 2 bytes
659	S2_COUPLING	1	Coupling type, 0 (Both), 1 (Positive), 2 (Negative),
660	S2_DEAD_ZONE_MM	1	Zone from center where no spring effect exists
661	S2_FORCE_SAT_N	1	Maximum force that the spring can output
662	D0_GAIN_NS_MM	1	Strength of damping force
663	I0_GAIN_NS2_MM	1	Strength of inertia force
664	O0_GAIN_N	1	Amplitude of periodic effect
665	O0_TYPE	1	Type of periodic effect 0 (square), 1 (sine), 2 (triangle), 3 (sawtooth)

666	O0_FREQ_DHZ	1	Period of oscillation
667	O0_DUTY	1	Pulse width modulation of signal as a % of duty cycle max value
668	O1_GAIN_N	1	Amplitude of periodic effect
669	O1_TYPE	1	Type of periodic effect 0 (square), 1 (sine), 2 (triangle), 3 (sawtooth)
670	O1_FREQ_DHZ	1	Frequency of periodic effect
671	O1_DUTY	1	Pulse width modulation of signal as a % of duty cycle max value
672	CONST_FORCE_FILTER	1	Amount of filtering on constant force inputs.
673	HAPTIC_SOFTSTART	1	Amount of time in milliseconds for force to ramp up upon enabling haptic mode.
674 - 755	Reserved		
756	ILOOP_DIN	1	Status of Ceagle digital inputs. Each input represented by a bit.
757	ILOOP_OUT_CH1	1	4-20 mA output channel 1.
758	ILOOP_OUT_CH2	1	4-20 mA output channel 2.
759	ILOOP_IN	1	4-20 mA input.
760	Reserved		
761	ILOOP_CONFIG	1	Configuration for 4-20mA control.
762	ILOOP_FORCE_MIN	1	Force that maps to 4 mA. Low 2 bytes. In millinewtons.
763	ILOOP_FORCE_MIN_HI	1	Force that maps to 4 mA. High 2 bytes. In millinewtons.

764	ILOOP_FORCE_MAX	1	Force that maps to 20 mA. Low 2 bytes. In millinewtons.
765	ILOOP_FORCE_MAX_HI	1	Force that maps to 20 mA. High 2 bytes. In millinewtons.
766	ILOOP_POS_MIN	1	Position that maps to 4 mA. Low 2 bytes. In micrometers.
767	ILOOP_POS_MIN_HI	1	Position that maps to 4 mA. High 2 bytes. In micrometers.
768	ILOOP_POS_MAX	1	Position that maps to 20 mA. Low 2 bytes. In micrometers.
769	ILOOP_POS_MAX_HI	1	Position that maps to 20 mA. High 2 bytes. In micrometers.
770	ILOOP_KIN_TYPE	1	Type of trigger behaviour
771	ILOOP_D0_HIGH_ID	1	Kinematic motion id value for rising edge digital 0
772	ILOOP_D0_LOW_ID	1	Kinematic motion id value for falling edge digital 0
773	ILOOP_D1_HIGH_ID	1	Kinematic motion id value for rising edge digital 1
774	ILOOP_D1_LOW_ID	1	Kinematic motion id value for falling edge digital 1
775	ILOOP_D2_HIGH_ID	1	Kinematic motion id value for rising edge digital 2
776	ILOOP_D2_LOW_ID	1	Kinematic motion id value for falling edge digital 2
777 - 779	Reserved		
780	KIN_MOTION_0	6	

786	KIN_MOTION_1	6	
792	KIN_MOTION_2	6	
798	KIN_MOTION_3	6	
804	KIN_MOTION_4	6	
810	KIN_MOTION_5	6	
816	KIN_MOTION_6	6	
822	KIN_MOTION_7	6	
828	KIN_MOTION_8	6	
834	KIN_MOTION_9	6	
840	KIN_MOTION_10	6	
846	KIN_MOTION_11	6	
852	KIN_MOTION_12	6	
858	KIN_MOTION_13	6	
864	KIN_MOTION_14	6	
870	KIN_MOTION_15	6	
876	KIN_MOTION_16	6	
882	KIN_MOTION_17	6	
888	KIN_MOTION_18	6	
894	KIN_MOTION_19	6	
900	KIN_MOTION_20	6	
906	KIN_MOTION_21	6	
912	KIN_MOTION_22	6	

918	KIN_MOTION_23	6	
924	KIN_MOTION_24	6	
930	KIN_MOTION_25	6	
936	KIN_MOTION_26	6	
942	KIN_MOTION_27	6	
948	KIN_MOTION_28	6	
954	KIN_MOTION_29	6	
960	KIN_MOTION_30	6	
966	KIN_MOTION_31	6	
972	KIN_HOME_ID	1	ID of kinematic motion triggered when Kinematic mode enabled or when Home signal asserved from Analog interface