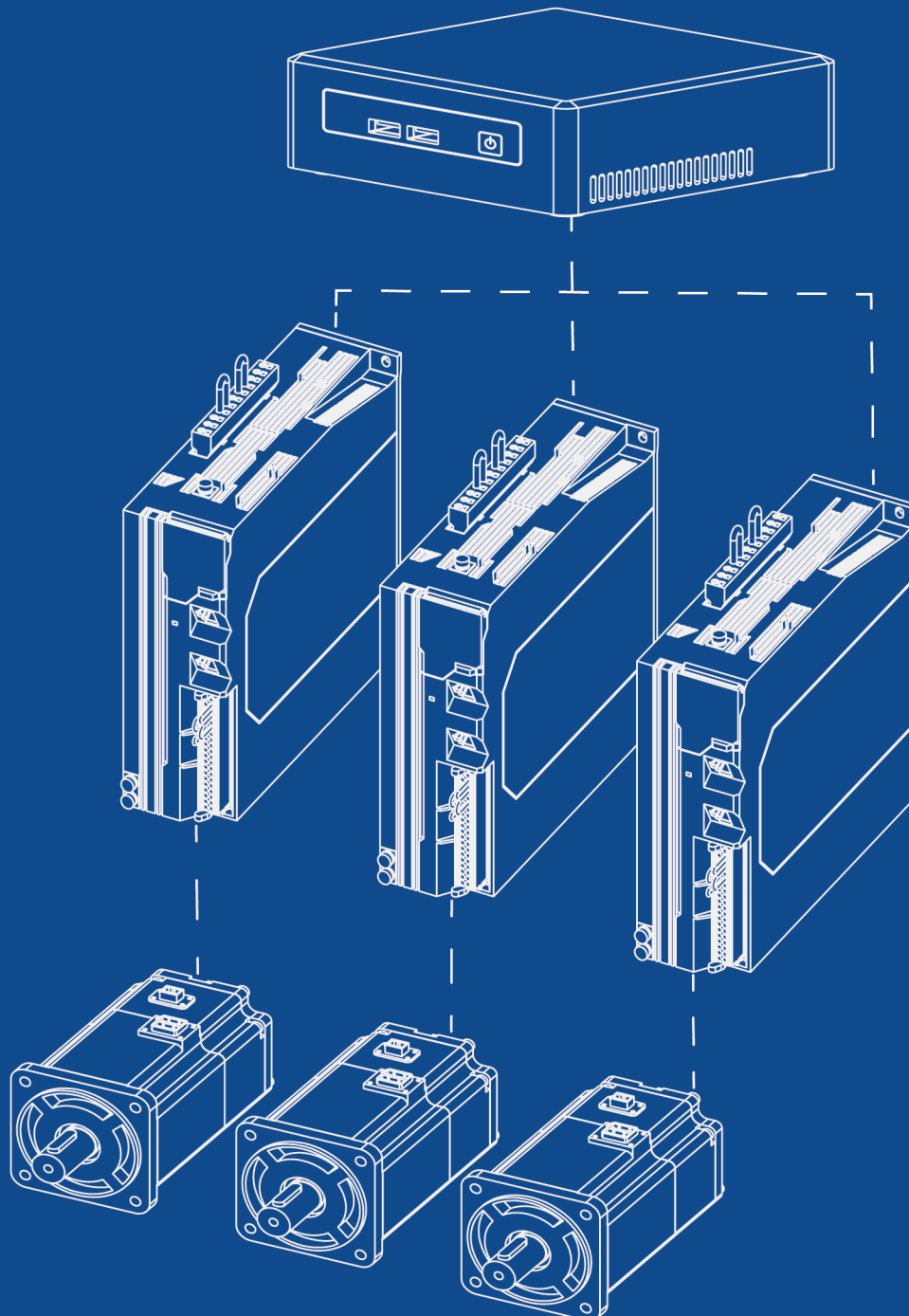




EtherCAT Master API Documentation

Precision. Scalability. Control



Getting Started with Adeeb Robotics EtherCAT Master Library

Thank you for evaluating the Adeeb Robotics EtherCAT Master library.

This library provides support for PDO and SDO EtherCAT communication via easy to use C api's.

Library also supports DC (Distributed Clocks) sync, EtherCAT state transitions, and CoE object reads & writes.

Requirements

Before using this library, ensure the following:

1. All connected slaves are EtherCAT devices implementing the CiA 402 motion control profile.
2. The TxPDO (slave-to-master) mapping includes at minimum:
 - a. Status Word: (0x6041)
 - b. Position Actual Value: (0x6064)
3. The RxPDO (master-to-slave) mapping includes at minimum:
 - a. Control Word: (0x6040)
 - b. Target Position: (0x607A)

Initial Setup

Ethernet Interface Configuration

1. Update the `interface_name` field in the `ethernet` section of the `app_config.yml` file to match the Ethernet adapter used for EtherCAT communication.

Note: If you're unsure of the interface name, simply run the library and it will display a list of available network interfaces.

Cycle Time Configuration

Set your desired EtherCAT cycle time in the `app_config.yml` file.

ESI Files

Add the EtherCAT Slave Information (ESI) files for your devices to the `ESI/` directory.

Building & Running

Linux / macOS

To rebuild and run the example application:

```
./rebuild_and_rerun_app.sh
```

To run the example application only:

```
sudo ./ethercat_master_example <path_to_config_file>
```

Windows

- To rebuild and run the example application (use the x64 Native Tools Command Prompt for Visual Studio 2022 or later):

```
rebuild_and_rerun_app.bat
```

- To run the example code only:

```
ethercat_master_example <path_to_config_file>
```

Note: The default configuration file is `app_config.yml`.

- To list available network interfaces, run:

```
ifconfig
```

Required Privileges

On Unix-based systems, root privileges (`sudo`) are required as the library performs real-time optimizations and direct access to the network interface. You can persist `sudo` privileges by running:

```
sudo -s
```

Additional Information

- Slave Indexing: Slaves' position indexed in network order starting from zero.
- Sample Application:

The `main.c` file contains a basic application that powers on a servo motor and jogs it at a slow constant speed. It serves as a reference for using the EtherCAT API.

⚠ Caution ⚠: The sample application will immediately power on the first motor and move it slowly. Ensure your hardware is safe and ready before running.

This library includes several OS- and system-level optimizations to help achieve high-performance real-time control.

Configuration File

Ethernet

Field	Type	Description
<code>interface_name</code>	String	Ethernet interface name
<code>prioritize_usr_space_eth_drv</code>	Boolean	If true, use ethernet driver optimization if possible. Only supported on select hardware running Linux, See Appendix B: Ethernet Driver Optimization

EtherCAT

Field	Type	Description	Unit
<code>esi_file_path</code>	String	ESI file path	N/A
<code>prioritize_esi_config</code>	Boolean	If true, prioritize obtaining slave configuration from ESI file; otherwise use slave EEPROM SSI data	N/A
<code>cycle_time</code>	32-bit unsigned integer	EtherCAT cycle time, only multiples of 250us is supported	us
<code>shift_time</code>	32-bit unsigned integer	DC master frame send to sync0 offset	% of <code>cycle_time</code>
<code>rx_frame_timeout</code>	32-bit unsigned integer	Receive frame timeout, only supported on Linux and MacOS	us

API's to be only called from within realtime cyclic thread

Note: All the API's below are non blocking except for: `thrd_rt_pre_opt_c()`, `sleep_tx_frm_rd_cyc_fbk()` and `wr_cyc_cmd_prc_async_dc_sync_rx_frm()`

API	Description	Parameters	Parameters Data Type	Returns
<code>thrd_rt_pre_opt_c()</code>	Thread real time optimization, to be	N/A	N/A	N/A

API	Description	Parameters	Parameters Data Type	Returns
	called only once before the start of the real time thread			
<code>sleep_tx_frm_rd_cyc_fbk()</code>	Sleep the required amount of time and transmit EtherCAT frame	N/A	N/A	Boolean
<code>wr_cyc_cmd_prc_async_dc_sync_rx_frm()</code>	Process asynchronous EtherCAT commands such as coe, performce DC sync and receive ethercat frame	N/A	N/A	N/A
<code>rd_slv_pos_act()</code>	Read slave actual position	<code>slv_idx</code>	unsigned integer	32-bit signed integer
<code>wr_slv_trg_pos()</code>	Write slave target position	<code>slv_idx</code>	unsigned integer	N/A
		<code>pos</code>	32-bit integer	
<code>rd_slv_sta_wrd()</code>	Read slave status word for warning and alarm monitoring	<code>slv_idx</code>	unsigned integer	16-bit unsigned integer
<code>rd_slv_cnt_wrd()</code>	Read slave control word	<code>slv_idx</code>	unsigned integer	16-bit unsigned integer
<code>wr_slv_cnt_wrd()</code>	Write slave control word i.e for clearing faults and alarms	<code>slv_idx</code>	unsigned integer	N/A
		<code>cnt_wrd</code>	16-bit integer	
<code>rd_slv_servo_on_current()</code>	Read current slave servo status (ON or OFF)	<code>slv_idx</code>	unsigned integer	Boolean
<code>wr_slv_servo_on_desired()</code>	Write desired slave servo status (ON or OFF), note this causes changes to the control word	<code>slv_idx</code>	unsigned integer	N/A
		<code>servo_on_desired</code>	Boolean	

API's to be only called from within the main thread.

Note: All the API's below are blocking except for `get_number_of_slaves()`

API	Description	Parameters	Parameters Data Type	Returns
<code>get_ecat_master_ver()</code>	Get EtherCAT master library version	<code>ver</code>	8-bit unsigned integer pointer	<code>rslt</code> 16-bit unsigned integer <i>See Appendix A: ECAT result error code.</i>
		<code>ver_len</code>	Unsigned integer	
<code>sys_rt_pre_opt_c()</code>	System real time optimization, to be called only once at the begining of main thread	N/A	N/A	N/A

<code>init_network()</code>	Scans the network for slaves, and places them in preop ethercat state	N/A	N/A	16-bit unsigned integer
<code>get_number_of_slaves()</code>	Gets number of slaves. Must be called after <code>init_network</code> api	N/A	N/A	16-bit unsigned integer
<code>req_preop()</code>	Place all slaves in pre-operational EtherCAT state	N/A	N/A	16-bit unsigned integer
<code>req_op()</code>	Place all slaves in operational EtherCAT state	N/A	N/A	16-bit unsigned integer
<code>coe_rd_u8()</code>	coe u8 read operation	<code>slv_idx</code>	unsigned integer	<code>rslt</code> 16-bit unsigned integer <i>See Appendix A: ECAT result error code.</i>
		<code>index</code>	16-bit unsigned integer	
		<code>sub_index</code>	8-bit unsigned integer	
		<code>data</code>	8-bit unsigned integer pointer	
<code>coe_wr_u8()</code>	coe u8 write operation	<code>slv_idx</code>	unsigned integer	<code>rslt</code> 16-bit unsigned integer <i>See Appendix A: ECAT result error code.</i>
		<code>index</code>	16-bit unsigned integer	
		<code>sub_index</code>	8-bit unsigned integer	
		<code>data</code>	8-bit unsigned integer	
<code>coe_rd_u16()</code>	coe u16 read operation	<code>slv_idx</code>	unsigned integer	<code>rslt</code> 16-bit unsigned integer <i>See Appendix A: ECAT result error code.</i>
		<code>index</code>	16-bit unsigned integer	
		<code>sub_index</code>	8-bit unsigned integer	
		<code>data</code>	16-bit unsigned integer pointer	
<code>coe_wr_u16()</code>	coe u16 write operation	<code>slv_idx</code>	unsigned integer	<code>rslt</code> 16-bit unsigned integer <i>See Appendix A: ECAT result error code.</i>
		<code>index</code>	16-bit unsigned integer	
		<code>sub_index</code>	8-bit unsigned integer	
		<code>data</code>	16-bit unsigned integer	
<code>coe_rd_u32()</code>	coe u32 read operation	<code>slv_idx</code>	unsigned integer	<code>rslt</code> 16-bit unsigned integer <i>See Appendix A: ECAT result error code.</i>
		<code>index</code>	16-bit unsigned integer	
		<code>sub_index</code>	8-bit unsigned integer	
		<code>data</code>	32-bit unsigned integer pointer	
<code>coe_wr_u32()</code>	coe u32 write operation	<code>slv_idx</code>	unsigned integer	<code>rslt</code> 16-bit unsigned integer <i>See Appendix A: ECAT result error code.</i>
		<code>index</code>	16-bit unsigned integer	
		<code>sub_index</code>	8-bit unsigned integer	
		<code>data</code>	32-bit unsigned integer	
<code>set_cia402_op_mode()</code>	Set slave to cyclic position mode	<code>slv_idx</code>	unsigned integer	<code>rslt</code> 16-bit unsigned integer <i>See Appendix A: ECAT result error code.</i>
		<code>op_mode</code>	8-bit unsigned integer	

Appendix A: ECAT result error codes

Value	Description
0x0000	Success
0x0001	Version string length insufficient
0x0002	Invalid slave index
0x0003	Uninitialized slave index
0x0004	Uninitialized station address
0x0005	Working counter not incremented
0x0006	Frame receive time out
0x0007	Frame command mismatch
0x0008	ESI file not found
0x0009	SM data cat section not found in SSI
0x000A	Register response mismatch
0x000B	SSI maximum byte length reached
0x000C	Max retries reached, while polling CoE SM1 response mbox ready status
0x000D	CoE mbox emergency
0x000E	CoE mbox abort code
0x000F	CiA operational mode maximum retries reached
0x0010	EtherCAT command mismatch
0x0FFF	Unspecified EtherCAT error

Appendix B: Ethernet Driver Optimization

Maximum possible update rate of 4000 Hz (i.e 250 us cycle time) only supported on PC's with realtime ethernet driver support. Specifically, this would be a PC with Linux OS and Intel ethernet LAN adapter using the igc Intel driver i.e one of the following ethernet PCIe cards from Intel (I225_LM, I225_V, I225_I, I220_V, I225_K, I225_K2, I226_K, I225_LMVP, I226_LMVP, I225_IT, I226_LM, I226_V, I226_IT, I221_V)