







分类专栏		
	linux	4篇
	ethercat	13篇
	其他	1篇
	c 语言基础	4篇
	stm32	5篇
	C++	2篇

```


108     }
109     printf("Domain1: State %u.\n", ds.wc_state);
110     domain1_state = ds;
111 }
112
113 void check_master_state(void)
114 {
115     ec_master_state_t ms;
116     ecrt_master_state(master, &ms);
117     if (ms.slaves_responding != master_state.slaves_responding)
118     {
119         printf("%u slave(s).\n", ms.slaves_responding);
120     }
121     if (ms.al_states != master_state.al_states)
122     {
123         printf("AL states: 0x%02X.\n", ms.al_states);
124     }
125     if (ms.link_up != master_state.link_up)
126     {
127         printf("Link is %s.\n", ms.link_up ? "up" : "down");
128     }
129     master_state = ms;
130 }
131
132
133
134 void check_slave_config_states(void)
135 {
136     ec_slave_config_state_t s;
137     ecrt_slave_config_state(sc, &s);
138     if (s.al_state != sc_state.al_state)
139     {
140         printf("slave: State 0x%02X.\n", s.al_state);
141     }
142     if (s.online != sc_state.online)
143     {
144         printf("slave: %s.\n", s.online ? "online" : "offline");
145     }
146     if (s.operational != sc_state.operational)
147     {
148         printf("slave: %soperational.\n", s.operational ? "" : "Not ");
149     }
150     sc_state = s;
151 }
152
153
154
155 void cyclic_task()
156 {
157     static uint16_t command=0x004f; //用来帮助判断状态字的值
158     uint16_t status;
159     /*Receive process data*/
160     ecrt_master_receive(master);
161     ecrt_domain_process(domain1);
162     /*Check process data state(optional)*/
163     check_domain1_state();
164
165     /*Check for master state
166     check_master_state();
167     //Check for slave configuration state(s)
168     check_slave_config_states();
169     //Read state*/
170     status = EC_READ_U16(domain1_pd + offset.status_word); //读取状态字
171
172     //DS402 CANOpen over EtherCAT status machine
173     if( (status & command) == 0x0040 )
174     {
175         EC_WRITE_U16(domain1_pd + offset.ctrl_word, 0x0006 );
176         EC_WRITE_S8(domain1_pd + offset.operation_mode, PROFILE_VELOCITY);
177     //设置控制模式
178         command = 0x006f;
179     }
180
181     else if( (status & command) == 0x0021)
182     {
183         EC_WRITE_U16(domain1_pd + offset.ctrl_word, 0x0007 );
184         command = 0x006f;
185     }
186
187     else if( (status & command) == 0x0023)
188     {
189         EC_WRITE_U16(domain1_pd + offset.ctrl_word, 0x000f );
190         command = 0x006f;
191     }
192     //operation enabled
193
194     else if( (status & command) == 0x0027)
195     {
196         EC_WRITE_S32(domain1_pd + offset.target_velocity, TARGET_VELOCITY);
197         EC_WRITE_U16(domain1_pd + offset.ctrl_word, 0x001f );
198     }
199
200     /*Send process data*/
201     ecrt_domain_queue(domain1);
202     ecrt_master_send(master);
203 }
204
205
206
207 int main(int argc, char **argv)
208 {
209     printf("Requesting master...\n");
210     master = ecrt_request_master(0);
211     if (!master)
212     {
213         exit(EXIT_FAILURE);
214     }
215
216     domain1 = ecrt_master_create_domain(master);
217     if (!domain1)
218     {
219         exit(EXIT_FAILURE);
220     }
221     if (! (sc = ecrt_master_slave_config(master, DM3E, VID_PID)))
222     {
223         fprintf(stderr, "Failed to get slave configuration for slave!\n");
224         exit(EXIT_FAILURE);
225     }
226     printf("Configuring PDOs...\n");
227     if (ecrt_slave_config_pdos(sc, EC_END, device_syncs))
228     {
229         fprintf(stderr, "Failed to configure slave PDOs!\n");
230         exit(EXIT_FAILURE);

```




```
231 }
232 else
233 {
234     printf("**Success to configuring slave PDOs*\n");
235 }
236
237 if (ecrt_domain_reg_pdo_entry_list(domain1, domain1_regs))
238 {
239     fprintf(stderr, "PDO entry registration failed!\n");
240     exit(EXIT_FAILURE);
241 }
242
243 printf("Activating master...\n");
244 if (ecrt_master_activate(master)) {
245     exit(EXIT_FAILURE);
246 }
247 else
248 {
249     printf("**Master activated*\n");
250 }
251 if (!(domain1_pd = ecrt_domain_data(domain1))) {
252     exit(EXIT_FAILURE);
253 }
254
255 printf("**It's working now*\n");
256
257 while (1)
258 {
259     usleep(100000/TASK_FREQUENCY);
260     cyclic_task();
261 }
262 return EXIT_SUCCESS;
263 }
264
```

igh ethercat控制台达电机(ds402)	10-28
目前ethercat主站资料比较少，大部分主站代码收费，开源igh驱动电机应用在国内网站几乎没有资料，好不容易找到控制台达ethercat驱动器代码，本人...	
igh ethercat 台达电机控制代码	10-28
ethercat主站资料较少，相关应用就更少了。好不容易找到一段控制台达电机的代码，亲自测试过能驱动点击标准的ds402协议	



优质评论可以帮助作者获得更高权重



评论

DieterZeh: 兄弟 igh控制电机的时候不需要涉及到实时线程的东西吗 这样直接控制电机 他的通讯周期是多少呢 1年前 回复 ***

IGH EtherCAT [\[博主\]](#) 回复: usleep(100000/TASK_FREQUENCY);这句代码控制着主站的发送周期，当然，这并没有实时保证，所以有时候伺服会出现报错 1年前 回复 ***

中华田园巨龙: 我一加上check_slave_config_states(这个函数就报错：段错误。删掉以后就正常运行。 1年前 回复 ***

中华田园巨龙 回复 IGH EtherCAT: 找到错误了，是我变量定义问题! 年前 回复 ***

IGH EtherCAT [\[博主\]](#) 回复: 其他的变量名字或者位置又没有改什么，段错误一般就是指针指向错误位置了 1年前 回复 ***

megfli: 62行后是空白？？ 1年前 回复 ***

megfli 回复: 然后电机也没动，状态字一直是0 1年前 回复 ***

megfli 回复 IGH EtherCAT: 额，之前用手机版看得，63行后代码就不显示了，电脑版可以看到。我在这使用该代码，遇到些问题，可以请教你们么？具体运行完后终端显示如下信息，该怎么改？

```
1 Activating master...
2 *Master activated*
3 *It's working now*
4 1 slave(s).
5 AL states: 0x02.
6 Link is up.
7 A6B: State 0x02.
8 A6B: online.
9 A6B: State 0x01.
10 A6B: State 0x02.
```

1年前 回复 ***

IGH EtherCAT [\[博主\]](#) 回复: 你是说63行花括号里面什么都没有还有什么？ 1年前 回复 ***

EtherCAT源代码分析(1)_IGH EtherCAT的博客	8-17
我感觉ecrt_request_master()函数除了增强鲁棒性的判断语句和操作语句.假设无任何错误.此函数做的操作是调用了ecrt_open_master(master_index),ecrt...	
EtherCAT igh主站控制埃斯顿伺服(csp模式)_c1n512的博客	9-21
完整代码算了、干脆直接贴代码了。最近都在搞EtherCAT主站、从站的应用.过段时间再分享一些项目中基础的东西。驱动伺服主要还是参考厂商提供的Et...	
igh ethercat应用层代码分析	qq_43530144的博客 1247
igh ethercat应用层代码分析 igh ethercat可以自己编写应用层代码与从站进行数据交互，本身主站给了例子，在example文件夹下，根据自己平台选择看...	
将Ethercat主站IGH移植到Linux应用层	huoxian00的专栏 1064
设计目的：IGH平台是完全全的Linux内核层代码，这样设计的目的是为了保证Ethercat主站的实时性。但是Linux即使加了实时补丁也只能实现软实时效...	
EtherCAT igh主站控制3个台达asdaa2伺服转圈圈_c1n512...	7-21
设置伺服DC.我写了篇文章有谈到这个函数EtherCAT igh源码的ecrt_slave_config_dc(函数)的理解。 https://blog.csdn.net/c1n512/article/details/1033852...	
igh ethercat主站文档(中文翻译上)_IGH EtherCAT的博客	8-15
这些程序不是EtherCAT主代码的一部分,而是必须由用户生成或编写。应用程序可以通过应用程序接口请求主服务器(请参见第3章)。如果成功,则可以控制...	
Ethercat系列--IGH DC模式测试电机程序(Preempt-RT)	中华田园巨龙 1566
先上代码 #include <errno.h> #include <signal.h> #include <stdio.h> #include <string.h> #include <sys/resource.h> #include <sys/time.h> #include <sy...	
igh EtherCAT Master 的使用	qq_46136275的博客 740
igh EtherCAT Master 的使用EtherCAT Mater 的启动ethercat masterethercat slavesethercat cstruc二级目录三级目录 EtherCAT Mater 的启动 输入命令...	
EtherCAT igh主站控制松下伺服(csp模式)	c1n512的博客 1759
完整代码 #include <errno.h> #include <signal.h> #include <stdio.h> #include <string.h> #include <sys/resource.h> #include <sys/time.h> #include <sy...	
Ethercat igh 文档	11-25
Ethercat igh 文档	
EtherCAT IGH 安装手册	10-28
IGH安装手册，beckhoff 数字量输入输出控制实例解析。	
igh配置EtherCAT流程	EtherCAT 6168
igh设置EtherCAT数据流程 在启动igh协议栈后，协议栈会自动的把EtherCAT从站初始化到PREOP模式，从PREOP状态迁移到SAFEOP状态，需要设置P...	
ethercat igh.zip	06-29
内含两个版本的ethercat igh master，其中一个支持4.x的内核。 怎么使用可以参考《ubuntu14.04、xenomai3.1、ethercat igh构建主站》和《ubuntu14.0...	
基于EtherCAT实时通信的电机驱动控制	10-16
实时以太网EtherCAT凭借着高性能、低成本、应用简易等优点在现代控制领域得到了广泛的应用和迅速的发展。为了将EtherCAT快速应用到电机驱...	
ethercat电机mfc简单连接程序	12-28
ethercat电机mfc简单连接程序，在windows操作系统上，建立控制电机工程的程序，可以自主的设计控制程序，代码只提供了连接电机部分代码，简单清晰	
SylixOS-igh系统平台搭建	weixin_28952781的博客 398
1.参考手册 下面7个文件存放在本地服务器\10.9.1.113.研发部9.ExchangeFolder\郑磊\路径下 《RealEvo-IDE使用手册》 《SylixOS应用开发手册》...	
基于AM4377的EtherCAT主站控制stm32从站	c1n512的博客 2629

weixin_45086497的博客

636

最新发布

在CODESYS中通过EtherCAT总线驱动单个电机

在CODESYS中通过EtherCAT总线驱动单个电机 本文讲述了手上有个支持EtherCAT总线的驱动器和步进电机的情况下，如何通过CODESYS添加Ether...

基于LinuxCNC下的EtherCAT的伺服电机控制《二》

思泽之思

3133

第二阶段，伺服电机动起来了，接下来的连接HAL，ini xml 文件，实现界面控制，不再是命令行 继续更新中

©2020 CSDN 皮肤主题: 1024 设计师: 白松林 返回首页

关于我们 招贤纳士 广告服务 开发助手 400-660-0108 kefu@csdn.net 在线客服 工作时间 8:30-22:00

公安备案号11010502030143 京ICP备19004658号 京网文〔2020〕1039-165号 经营性网站备案信息 北京互联网违法和不良信息举报中心 网络110报警服务 中国互联网举报中心 家长监护 Chrome商店下载 ©1999-2021北京创新乐知网络技术有限公司 版权与免责声明 版权申诉 出版物许可证 营业执照

IGH EtherCAT

关注

1

9

22

专栏目录

微信扫码

0/0

举报