

Beliven  
码农9年 暂无认证

6  
原创

317  
积分

私信

关注

搜博主文章

### 最新评论

IGH (EtherCAT Master Linux on PC)编译...  
此姓名已被抢注: 也是一样的, 没解决, 所以还是用了低版本

IGH (EtherCAT Master Linux on PC)编译...  
Beliven: 注意看示例, ( ) 是表示说明的, 不能用在来表示configure命令的参数, 重...

IGH (EtherCAT Master Linux on PC)编译...  
YJC呀: 您好, 我是内核4.9.38+xenomai3.1 .make modules的时候也是报错, 报关于...

IGH (EtherCAT Master Linux on PC)编译...  
头不壳还能学: 博主, 我打了3.18.59-rt65内核补丁, 但是在usr/src/linux-header都...

IGH (EtherCAT Master Linux on PC)编译...  
此姓名已被抢注: 你好, 请教一下, 就是运行example/xenomai这个例子, 运行一段...

### 您愿意向朋友推荐“博客详情页”吗？

强烈不推荐 不推荐 一般般 推荐 强烈推荐

### 最新文章

使用gdb定位for循环中的段错误的一种方法

visio中形状如何链接到形状

IGH on Beaglebone Black(AM3358) with Xenomai

2020年 1篇

2019年 1篇

2018年 4篇

## 松下(Panasonic)伺服EtherCAT(IGH EtehrCAT Master)通信注意事项

原创 Beliven 2018-04-12 16:23:16 17869 收藏 40

分类专栏: EtherCAT 文章标签: 松下 伺服 IGH EtherCAT

EtherCAT 专栏收录该内容

本文使用的松下伺服为A6B系列, EtherCAT主站搭建方法为IGH EtherCAT Master第三方协议栈。

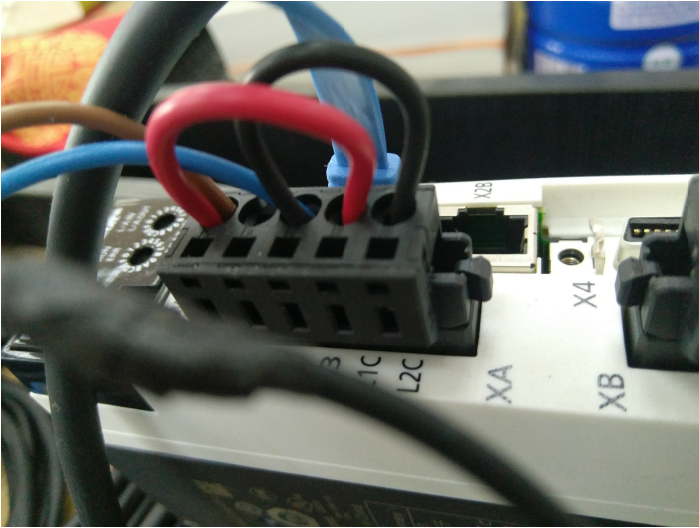
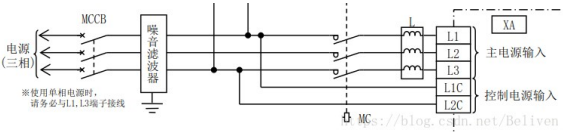
### ● 电源接线

很多人拿到新的设备不知道怎么接电源线, 这里给一个简洁的说明。

普通交流电接L1和L3 (不用区分火线和零线), 然后L1C接到L1, L2C接到L3;

三相电接L1、L2和L3, 然后L1C接到L1, L2C接到L3;

当然还要接地。



### ● 试运行

用miniUSB线缆连接到电脑, 电脑上事先装好PANATERM软件, 识别到设备之后即可进行USB通信。这时候如果进入试运行, 然后伺服使能, 再伺服开启, 发现并不能打开伺服设备, 提示你错误代码38.1:

38	1	驱动禁止输入保护 2	Pr5.04「驱动禁止输入设定」=0 或者 1 状态下, EtherCAT 通信断开状态并且 POT/NOT 其中任何一个开启的状态下, 通过USB 通信(PANATERM)接收动作指令(试运行、FFT 等)。相反, 通过USB通信(PANATERM)在动作指令下动作中开启POT/NOT。	• 确认接入正方向/负方向驱动禁止输入的开关、电线、电源是否异常。特别是要确认控制用信号电源(DC12~24 V)启动是否延迟。
----	---	------------	---	--

进入参数设置界面, 将参数Pr4.01(SI2)改为0x010101h (十进制为65793), 4.02(SI3)改为0x020202h (十进制为131586) 即可, 然后写入到EEROM, 重新上电之后即可以进行试运行。

### ● EtherCAT试运行

这里给出IGH的测试代码, 模式为pv, 需要注意的是, 速度的单位是指令/s, 23位编码器的电机转一圈的脉冲是8388608, 那么如果将目的速度设置为8388608则表示赋给的速度为1r/s; 如果将设置为i r/s, 那么速度值应该为i\*8388608。

程序需要的信息都可以从松下电机官网下载的ESI文件中获取, 包括VID、PID以及寄存器信息。

```
1  /******  
2  * Example of Panasonic A6B servo drive setup with Igh EtherCAT Master library  
3  * that starts the servo drive in profile velocity mode.  
4  *  
5  * Licence: the same than Igh EtherCAT Master library  
6  *  
7  * ABSTRACT:  
8  *  
9  * This small program shows how to configure a Panasonic A6B servo drive  
10 * with Igh EtherCAT Master library. It illustrates the flexible PDO mapping  
11 * and the DS402 command register programming in order to start the servo drive  
12 * in profile velocity mode.  
13 *  
14 * Because the Panasonic A6B servo drive follows the DS402 standard, several  
15 * sections of this example may apply also to other servo drive models.  
16 *  
17 * The code is based on a previous example from Igh EtherCAT Master  
18 *  
19 * Author: Zhou Yonghong  
20 * Date: 2018/04/04  
21 * Compile with:  
22 *  
23 gcc testbyesm.c -Wall -I /opt/etherlab/include -l ethercat -L /opt/etherlab/lib -o testbyesm  
24 *  
25 *****/  
26  
27 #include <errno.h>  
28 #include <signal.h>  
29 #include <stdio.h>  
30 #include <string.h>  
31 #include <sys/resource.h>  
32 #include <sys/time.h>  
33 #include <sys/types.h>  
34 #include <unistd.h>  
35 #include <sys/mman.h>  
36  
37 /******  
38  
39 #include "ecrt.h"  
40  
41 /******  
42  
43 /*Application Parameters*/  
44 #define TASK_FREQUENCY 100 /*Hz*/  
45 #define TIMEOUT_CLEAR_ERROR (1*TASK_FREQUENCY) /*clearing error timeout*/  
46 #define TARGET_VELOCITY 8388608 /*target velocity*/  
47 #define PROFILE_VELOCITY 3 /*Operation mode for 0x6060:0*/  
48
```

### 分类专栏

0 订阅 3 篇文章

订阅专栏

```

49  /******
50
51  /*EtherCAT*/
52  static ec_master_t *master = NULL;
53  static ec_master_state_t master_state = {};
54
55  static ec_domain_t *domain1 = NULL;
56  static ec_domain_state_t domain1_state = {};
57
58  static ec_slave_config_t *sc_A6B;
59  static ec_slave_config_state_t sc_A6B_state = {};
60
61  /******
62
63  /*Process Data*/
64  static uint8_t *domain1_pd = NULL;
65
66  #define A6BSlavePos      0,0          /*EtherCAT address on the bus*/
67
68  #define Panasonic        0x000066F,0x60380004 /*Vendor ID, product code*/
69
70  /*Offsets for PDO entries*/
71  static struct{
72      unsigned int operation_mode;
73      unsigned int ctrl_word;
74      unsigned int target_velocity;
75      unsigned int status_word;
76      unsigned int mode_display;
77      unsigned int current_velocity;
78      // signed int current_velocity;
79  }offset;
80
81  const static ec_pdo_entry_reg_t domain1_regs[] = {
82      {A6BSlavePos, Panasonic, 0x6040, 0, &offset.ctrl_word},
83      {A6BSlavePos, Panasonic, 0x6060, 0, &offset.operation_mode},
84      {A6BSlavePos, Panasonic, 0x60FF, 0, &offset.target_velocity},
85      {A6BSlavePos, Panasonic, 0x6041, 0, &offset.status_word},
86      {A6BSlavePos, Panasonic, 0x6061, 0, &offset.mode_display},
87      {A6BSlavePos, Panasonic, 0x606C, 0, &offset.current_velocity},
88      {}
89  };
90
91  static unsigned int counter = 0;
92  //static int state = -500;
93
94  /******
95  /*Config PDOS*/
96  static ec_pdo_entry_info_t A6B_pdo_entries[] = {
97      /*RxPdo 0x1600*/
98      {0x6040, 0x00, 16},
99      {0x6060, 0x00, 8},
100     {0x60FF, 0x00, 32},
101     /*TxPdo 0x1A00*/
102     {0x6041, 0x00, 16},
103     {0x6061, 0x00, 8},
104     {0x606C, 0x00, 32}
105 };
106
107 static ec_pdo_info_t A6B_pdos[] = {
108     /*RxPdo
109     {0x1600, 3, A6B_pdo_entries + 0},
110     /*TxPdo
111     {0x1A00, 3, A6B_pdo_entries + 3}
112 };
113
114 static ec_sync_info_t A6B_syncs[] = {
115     { 0, EC_DIR_OUTPUT, 0, NULL, EC_WD_DISABLE },
116     { 1, EC_DIR_INPUT, 0, NULL, EC_WD_DISABLE },
117     { 2, EC_DIR_OUTPUT, 1, A6B_pdos + 0, EC_WD_DISABLE },
118     { 3, EC_DIR_INPUT, 1, A6B_pdos + 1, EC_WD_DISABLE },
119     { 0xFF}
120 };
121
122 /******
123
124 /******
125
126
127 void check_domain1_state(void)
128 {
129     ec_domain_state_t ds;
130     ecrt_domain_state(domain1, &ds);
131     if (ds.working_counter != domain1_state.working_counter)
132     {
133         printf("Domain1: WC %u.\n", ds.working_counter);
134     }
135     if (ds.wc_state != domain1_state.wc_state)
136     {
137         printf("Domain1: State %u.\n", ds.wc_state);
138     }
139     domain1_state = ds;
140 }
141
142 void check_master_state(void)
143 {
144     ec_master_state_t ms;
145     ecrt_master_state(master, &ms);
146     if (ms.slaves_responding != master_state.slaves_responding)
147     {
148         printf("%u slave(s).\n", ms.slaves_responding);
149     }
150     if (ms.al_states != master_state.al_states)
151     {
152         printf("AL states: 0x%02X.\n", ms.al_states);
153     }
154     if (ms.link_up != master_state.link_up)
155     {
156         printf("Link is %s.\n", ms.link_up ? "up" : "down");
157     }
158     master_state = ms;
159 }
160
161
162 /******
163
164 void check_slave_config_states(void)
165 {
166     ec_slave_config_state_t s;
167     ecrt_slave_config_state(sc_A6B, &s);
168     if (s.al_state != sc_A6B_state.al_state)
169     {
170         printf("A6B: State 0x%02X.\n", s.al_state);
171     }
172     if (s.online != sc_A6B_state.online)
173     {
174         printf("A6B: %s.\n", s.online ? "online" : "offline");
175     }
176     if (s.operational != sc_A6B_state.operational)
177     {
178         printf("A6B: %soperational.\n", s.operational ? "" : "Not ");
179     }
180     sc_A6B_state = s;
181 }
182
183 /******

```

```

184 void cyclic_task()
185 {
186     static unsigned int timeout_error = 0;
187     static uint16_t command=0x004F;
188     static int32_t target_velocity = TARGET_VELOCITY;
189
190     uint16_t    status;
191     int8_t      opmode;
192     int32_t     current_velocity;
193     // int32_t    command_value;
194
195     /*Receive process data*/
196     ecrt_master_receive(master);
197     ecrt_domain_process(domain1);
198     /*Check process data state(optional)*/
199     check_domain1_state();
200
201     counter = TASK_FREQUENCY;
202     //Check for master state
203     check_master_state();
204     //Check for slave configuration state(s)
205     check_slave_config_states();
206     /*Read inputs*/
207     status = EC_READ_U16(domain1_pd + offset.status_word);
208     opmode = EC_READ_U8(domain1_pd + offset.mode_display);
209     current_velocity = EC_READ_S32(domain1_pd + offset.current_velocity);
210
211     printf("A68:  act velocity = %d , cmd = 0x%x , status = 0x%x , opmode = 0x%x\n",
212           (current_velocity*60)/target_velocity, command, status, opmode);
213
214
215
216     //DS402 CANOpen over EtherCAT status machine
217     if( (status & command) == 0x0040 )
218     {
219         EC_WRITE_U16(domain1_pd + offset.ctrl_word, 0x0006 );
220         EC_WRITE_S8(domain1_pd + offset.operation_mode, PROFILE_VELOCITY);
221         command = 0x006F;
222     }
223
224     else if( (status & command) == 0x0021)
225     {
226         EC_WRITE_U16(domain1_pd + offset.ctrl_word, 0x0007 );
227         command = 0x006F;
228     }
229
230     else if( (status & command) == 0x0023)
231     {
232         EC_WRITE_U16(domain1_pd + offset.ctrl_word, 0x000f );
233         EC_WRITE_S32(domain1_pd + offset.target_velocity, target_velocity);
234         command = 0x006F;
235     }
236     //operation enabled
237
238     else if( (status & command) == 0x0027)
239     {
240         EC_WRITE_U16(domain1_pd + offset.ctrl_word, 0x001f);
241     }
242
243     /*Send process data*/
244     ecrt_domain_queue(domain1);
245     ecrt_master_send(master);
246 }
247
248
249
250
251 int main(int argc, char **argv)
252 {
253     master = ecrt_request_master(0);
254     if (!master)
255     {
256         exit(EXIT_FAILURE);
257     }
258     domain1 = ecrt_master_create_domain(master);
259
260     if (!domain1)
261     {
262         exit(EXIT_FAILURE);
263     }
264     if (!(sc_A68 = ecrt_master_slave_config(master, A68SlavePos, Panasonic)))
265     {
266         fprintf(stderr, "Failed to get slave configuration for A68!\n");
267         exit(EXIT_FAILURE);
268     }
269     // printf("alias=%d,vid=%d,watchdog_divider=%d,sc_A68->alias,sc_A68->position,sc_A68->watchdog_divider);
270
271     printf("Configuring PDOs...\n");
272     if (ecrt_slave_config_pdos(sc_A68, EC_END, A68_syncs))
273     {
274         fprintf(stderr, "Failed to configure A68 PDOs!\n");
275         exit(EXIT_FAILURE);
276     }
277     else
278     {
279         printf("**Success to configuring A68 PDOs*\n");
280     }
281
282     if (ecrt_domain_reg_pdo_entry_list(domain1, domain1_regs))
283     {
284         fprintf(stderr, "PDO entry registration failed!\n");
285         exit(EXIT_FAILURE);
286     }
287     else
288     {
289         printf("**Success to configuring A68 PDO entry*\n");
290         printf("operation_mode=%d, ctrl_word=%d, target_velocity=%d, status_word=%d, mode_display=%d, current_velocity=%d, offset.operation_mode,offset.ctrl_word,offset.target_velocity,offset.status_word,offset.mode_display\n",
291               offset.operation_mode,offset.ctrl_word,offset.target_velocity,offset.status_word,offset.mode_display,offset.current_velocity);
292     }
293
294     printf("Activating master...\n");
295     if (ecrt_master_activate(master)) {
296         exit(EXIT_FAILURE);
297     }
298     else
299     {
300         printf("**Master activated*\n");
301     }
302     if (!(domain1_pd = ecrt_domain_data(domain1))) {
303         exit(EXIT_FAILURE);
304     }
305
306     printf("**It's working now*\n");
307
308     while (1)
309     {
310         usleep(100000/TASK_FREQUENCY);
311         cyclic_task();
312     }
313     ecrt_master_deactive(master);
314     return EXIT_SUCCESS;
315 }

```

热

本站控制3个伺服的EtherCAT报文分析

目前有多种用于提供实时功能的大网方案：例如，通过较高级的协议层禁止CSMA/CD存取过程，并使用时间片轮询过程来取代它。其它方案使用专用...

松下EtherCAT驱动器EtherCAT描述文件

此xml是松下电机的配置文件，通过此配置文件，能让我们很快的连接EtherCAT主站，搭建运动控制系统，同时XML也是必不可少的从站伺服的描述文件，没...

09-14

12-11

评论

优化评论可以帮助您获得更高权重

master\_qxt: 我参照你的文章写了代码，发现linux系统日志里报'slaves dont support changing pdo mapping'的信息，代码可以激活主站，用ec\_write\_u16函数将控制字命令写入0x0f,但是电机不动了，用手也可以晃动，功能是什么原因呢 2年前 回复 ···

nsygc: 你好 可以简单咨询一下吗 2年前 回复 ···

liuzq: 你好BeIiven,可以给出QQ号么? 2年前 回复 ···

BeIiven(博主) 回复: 我毕业没搞这个了(´o`)忘得差不多啦 2年前 回复 ···

8-17

12-11

正文

正运动技术快速入门 | 篇十六:正运动控制器EtherCAT总线快速入门

正运动技术与大家分享了运动控制器的固件升级、ZBasic程序开发、ZPLC程序开发、与触摸屏通讯和输入/输出IO的应用、运动控制器数据与存储...

松下ethercat文件\_欧姆龙NXPLC和松下A6 EtherCAT伺服配置

首先你去使用的设备官网下载你使用的EtherCAT从设备的ESI文件(一般买的时候供应商会直接提供以下以松下伺服为例在Sysmac Studio中添加好你的P...

weixin\_39817139的博客 · 1651

EtherCAT Igh主站控制3个伺服 (csp模式)

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 <sys...

ch512的博客 · 1749

松下伺服上下限位信号EtherCAT读取

图1 EtherCAT报文中的数字输入信号 图2 下限位 图3 上位 图4 没有撞时 由此图4的最后两位可见上下限位置的触点

bdawn的专栏 · 1451

EtherCAT Igh主站控制3个台达asdaa2伺服转圈

EtherCAT Igh主站控制3个台达asdaa2伺服转圈,包括伺服使能、控制3个电机转圈、每秒读取电机的实际位置、运行90秒自动退出程序。。。需要...

12-03

松下伺服A6BF驱动器EtherCAT描述文件XML

此xml是松下A6BF的配置文件,通过此配置文件,能让我们很快的连接EtherCAT主站,搭建运动控制系统。

11-20

松下伺服MODBUS通信手册

松下伺服MODBUS通信手册,通过 Modbus 通信,可进行读写参数、读取伺服驱动器内部信息等操作。此外,通过将 Pr.6 28 [特殊功能选择] 设置为 1...

12-19

EtherCAT Igh主站控制埃斯顿伺服 (csp模式)

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...

ch512的博客 · 2686

Igh EtherCAT Master for Linux

Master EtherCAT applications do not need any custom device and can be run in a normal PC, laptop or device with an Ethernet connector. EtherCAT S...

11-27

Igh EtherCAT Master 安装记录

前言 IGH EtherCAT Master 的1.5.2版本中没有 bootstrap 文件,如果只运行 configure 文件后会报错,所以在安装时选取了最近的stable-1.5. IGH源码下...

qq\_28882933的博客 · 63

Igh EtherCAT Master 的使用

Igh EtherCAT Master 的使用EtherCAT Mater 的启动 ethercat master ethercat slaves ethercat cstruct 二级目录三级目录 EtherCAT Mater 的启动 输入命令...

qq\_46136278的博客 · 736

Linux下IGH EtherCAT 安装

引言 简单IGH ethercat master 安装 1, 准备工作 (1) 一个Linux系统,在虚拟机里面也可以,不过如果在虚拟机里面需要一些其他的设置,这个最后再说...

qq\_43530144的博客 · 2727

Igh EtherCAT Master 源码编译安装

在Linux环境中安装EtherCAT Master源码时,输入命令: # make modules\_install 出现下面的错误: /home/ethercat/ethercat-1.5.2/master/ethernet.c: L...

scyhk的博客 · 175+

松下EtherCAT伺服资料

介绍了松下EtherCAT伺服与Beckhoff如何通讯,如何操作松下伺服

06-08

倍福TwinCAT(贝福Beckhoff)应用教程12.1 TwinCAT控制松下伺服 连接和试运行

首先是用松下伺服自带的软件可以测试运行(驱动器,电机都连接好,然后用USB线连接到松下伺服驱动器的X1口),打开测试软件会自动提示连接到伺...

acetaohai123123的博客 · 17+

EtherCAT PDO Mapping Doc(松下)

松下的EtherCAT驱动器使用手册,以及关于如何使用EtherCAT配置PDO等内容

02-22

EtherCAT使用与解析-关于PDO映射

关于PDO映射这部分,最好先参考https://blog.csdn.net/lswdcy/article/details/105441470先了解一下配置相关的东西; 首先这里已经假设你已经了解完...

lswdcy的博客 · 4998

©2020 CSDN 皮肤主题: 大白 设计师:CSDN官方博主 返回首页