



```
93 SDO 0x1009, "Hardware version"
94 0x1009:00, r-r-r-r, string, 32 bit, "Hardware version"
95 SDO 0x100a, "Software version"
96 0x100a:00, r-r-r-r, string, 88 bit, "Software version"
97 SDO 0x1010, "Store parameters"
98 0x1010:00, r-r-r-r, uint8, 8 bit, "Software version"
99 0x1010:01, rwrwrw, uint32, 32 bit, "Save all parameters"
100 SDO 0x1018, "Identity"
101 0x1018:00, r-r-r-r, uint8, 8 bit, "Identity"
102 0x1018:01, r-r-r-r, uint32, 32 bit, "Vendor ID"
103 0x1018:02, r-r-r-r, uint32, 32 bit, "Product code"
104 0x1018:03, r-r-r-r, uint32, 32 bit, "Revision"
105 0x1018:04, r-r-r-r, uint32, 32 bit, "Serial number"
106
107 5. 查看主机信息
108 ethercat master
109 Master0
110 Phase: Idle
111 Active: no
112 Slaves: 1
113 Ethernet devices:
114 Main: e4:f3:f5:c6:4e:b5 (attached)
115 Link: UP
116 Tx frames: 44056
117 Tx bytes: 3942496
118 Rx frames: 44055
119 Rx bytes: 3942436
120 Tx errors: 0
121 Tx frame rate [1/s]: 100 100 100
122 Tx rate [KByte/s]: 5.9 5.8 6.0
123 Rx frame rate [1/s]: 100 100 100
124 Rx rate [KByte/s]: 5.9 5.8 6.0
125 Common:
126 Tx frames: 44056
127 Tx bytes: 3942496
128 Rx frames: 44055
129 Rx bytes: 3942436
130 Lost frames: 0
131 Tx frame rate [1/s]: 100 100 100
132 Tx rate [KByte/s]: 5.9 5.8 6.0
133 Rx frame rate [1/s]: 100 100 4340
134 Rx rate [KByte/s]: 5.9 5.9 6.0
135 Loss rate [1/s]: 0 -0 0
136 Frame loss [%]: 0.0 -0.0 0.0
137 Distributed clocks:
138 Reference clock: Slave 0
139 Application time: 0
140 2000-01-01 00:00:00.000000000
141
142 6.更改别名
143 ethercat alias 0x0001
144 ethercat slaves
145 0 1:0 PREOP + R ADVANCED MODEL with EtherCAT(P0002813G01)
146 别名一旦更改可以保存在驱动器E2PROM中,配置好后可以下次直接使用,多轴控制的时候可以配置不同的别名信息区分不同的轴
147
148 7. 操作模式设置 (CoE INDEX:SUBINDEX 0x0600:0x00 )
149 设置
150 ethercat download -a 0 -p 0 0x6060 0x00 0x01
151 读取
152 ethercat upload -a 0 -p 0 0x6061 0x00
153
154 8. 从机状态设置
155 ethercat state -a 0 -p 0 OP
156
157 9. 自由运行模式(手册P221)
158 设置自由运行模式(FreeRun mode)
159 root@linaro-ubuntu-desktop:/home/linaro/ftp/wuliang# ethercat download -a 0 -p 0 -t uint8 0x1c32 0x01 0x00
160
161 10. DC模式不能进入OP模式的可能原因?
162 DC模式一般用于实时性较高的场合,主站实时性要求非常高,一般高实时性场合Slaves要求时间抖动不超过100us
163
164 11. 通过ethercatit电机力矩
165
166 11.1 设置自由模式
167 ethercat download -a 0 -p 0 -t uint8 0x1c32 0x01 0x00
168 设置力矩模式:
169 ethercat download -a 0 -p 0 -t uint8 0x6060 0x00 0x04
170
171 11.2 设置驱动器成'OP'模式
172 ethercat states -a 0 -p 0 OP
173
174 11.3 伺服ON
175 root@linaro-ubuntu-desktop:/home/output/bin# ethercat download -a 0 -p 0 -t uint8 0x6040 0x00 0x06
176 root@linaro-ubuntu-desktop:/home/output/bin# ethercat download -a 0 -p 0 -t uint8 0x6040 0x00 0x07
177 root@linaro-ubuntu-desktop:/home/output/bin# ethercat download -a 0 -p 0 -t uint8 0x6040 0x00 0x0f
178
179 11.4 设置目标力矩电机开始转动(山羊电机手册P270)
180 以正力矩转动:
181 ethercat download -a 0 -p 0 -t uint16 0x6071 0x00 0x001c
182 以负力矩转动:
183 ethercat download -a 0 -p 0 -t uint16 0x6071 0x00 0xffe4
184
185 11.5 停止电机转动
186 ethercat download -a 0 -p 0 -t uint8 0x6040 0x00 0x0f
187 ethercat download -a 0 -p 0 -t uint8 0x6040 0x00 0x06
188
189 12.设置例句模式问题
190 OP模式下进行一些对象字典设置会有如下报错:
191 root@linaro-ubuntu-desktop:/home/linaro/ftp/ethercat-1.5.2/test# ethercat download -a 0 -p 0 -t uint8 0x6060 0x00 0x00
192 EtherCAT ERROR 0-0: Reception of CoE download response failed: No response.
193 EtherCAT ERROR 0-0: Failed to process SDO request.
194 Failed to download SDO: Input/output error
195 原因是OP模式不能更改一些寄存器的值所以会包上述错误
196
197 13.非'OP'模式力矩测试问题
198 经测试发现非'OP'模式下也可以设置电机力矩控制
199
200 14. 数字输出OUT1,OUT2控制
201
202 Ethercat控制OUT1输出
203 ethercat download -a 0 -p 0 -t uint32 0x60fe 0x01 0x00010000
204 Ethercat控制OUT2输出
205 ethercat download -a 0 -p 0 -t uint32 0x60fe 0x01 0x00020000
206 Ethercat同时控制OUT1,OUT2输出
207 ethercat download -a 0 -p 0 -t uint32 0x60fe 0x01 0x00030000
208
209 OUT1输出ON(刹车松开)
210 ethercat download -a 0 -p 0 -t uint8 0x20f9 0x02 0x42
211 OUT1输出OFF((刹车抱死))
212 ethercat download -a 0 -p 0 -t uint8 0x20f9 0x02 0x43
213 OUT2输出ON
214 ethercat download -a 0 -p 0 -t uint8 0x20f9 0x02 0x44
215 OUT2输出OFF
```

```
217 ethercat download -a 0 -p 0 -t uint8 0x20f9 0x02 0x45
218
219 14.
220 OUTPUT----RxPdo 映射对象字典原始数据读取
221 ethercat upload -a 0 -p 0 -t uint16 0x6040 0x00
222 ethercat upload -a 0 -p 0 -t uint32 0x607a 0x00
223 ethercat upload -a 0 -p 0 -t uint32 0x6081 0x00
224 ethercat upload -a 0 -p 0 -t uint32 0x6083 0x00
225 ethercat upload -a 0 -p 0 -t uint32 0x6084 0x00
226 ethercat upload -a 0 -p 0 -t uint32 0x60ff 0x00
227 ethercat upload -a 0 -p 0 -t uint16 0x6071 0x00
228 ethercat upload -a 0 -p 0 -t uint16 0x60b8 0x00
229 ethercat upload -a 0 -p 0 -t uint32 0x60fe 0x01
230
231
232 INPUT----TxPdo 映射对象字典原始数据读取
233 ethercat upload -a 0 -p 0 -t uint16 0x6041 0x00
234 ethercat upload -a 0 -p 0 -t uint16 0x2100 0x00
235 ethercat upload -a 0 -p 0 -t uint32 0x6064 0x00
236 ethercat upload -a 0 -p 0 -t uint32 0x606c 0x00
237 ethercat upload -a 0 -p 0 -t uint16 0x6077 0x00
238 ethercat upload -a 0 -p 0 -t uint32 0x60f4 0x00
239 ethercat upload -a 0 -p 0 -t uint16 0x60b9 0x00
240 ethercat upload -a 0 -p 0 -t uint32 0x60ba 0x00
241 ethercat upload -a 0 -p 0 -t uint32 0x60bb 0x00
242 ethercat upload -a 0 -p 0 -t uint32 0x60fd 0x00
243 ethercat upload -a 0 -p 0 -t uint8 0x1001 0x00
244 ethercat upload -a 0 -p 0 -t uint8 0x6061 0x00
245
246 15. 发生报警时根据手册P420页内容逐步排查原因
247 15.1 任务处理异常
248
249 报警代码      描述
250 F1              任务处理异常
251 0x603f:0x8700
252 0x2101/0x2102
253 针对问题15.1的解决办法:
254 (1)要先配0x6040:0x00=0x06,让伺服驱动处于ShutDown状态 山洋电机手册 P229
255 (2)没有进行pdo映射的对象字典要通过sdo方式先进行配置
256 (3)sdo配置对象字典要在主站激活前进行
257 (4)所有配置完成,主机激活完成通过PDO方式配置0x6040:0x00=0x07,使能主机
258
259 *****
260 泰科电机调试
261 (1)获取VENDOR_ID(手册P212查看索引及子索引)
262 ethercat upload -a 1 -p 0 0x1018 0x01
263 0x000001b9 441
264 (2)获取PRODUCT_ID(手册P212查看索引及子索引)
265 ethercat upload -a 1 -p 0 0x1018 0x02
266 0x00000002 2
267
268 -----
269 TwinCat 配置泰科伺服
270 1. 要在配置模式进行对象字典的配置
271
272 伺服使能状态字对比
273 root@linaro-ubuntu-desktop:/home/linaro/ftp/ethercat-1.5.2/test# ethercat download -a 255 -p 0 -t uint8 0x6061
274 root@linaro-ubuntu-desktop:/home/linaro/ftp/ethercat-1.5.2/test# ethercat download -a 255 -p 0 -t uint16 0x60
275 root@linaro-ubuntu-desktop:/home/linaro/ftp/ethercat-1.5.2/test# ethercat upload -a 255 -p 0 -t uint16 0x6041
276 0x0631 1585
277 root@linaro-ubuntu-desktop:/home/linaro/ftp/ethercat-1.5.2/test# ethercat download -a 255 -p 0 -t uint16 0x60
278 root@linaro-ubuntu-desktop:/home/linaro/ftp/ethercat-1.5.2/test# ethercat upload -a 255 -p 0 -t uint16 0x6041
279 0x0633 1587
280 root@linaro-ubuntu-desktop:/home/linaro/ftp/ethercat-1.5.2/test# ethercat download -a 255 -p 0 -t uint16 0x60
281 root@linaro-ubuntu-desktop:/home/linaro/ftp/ethercat-1.5.2/test# ethercat upload -a 255 -p 0 -t uint16 0x6041
282 0x0637 1591
283
284 -----
285 重启后:
286 root@linaro-ubuntu-desktop:~# ethercat upload -a 255 -p 0 -t uint16 0x6041 0x00
287 0x0670 1648
288 通过调试软件使能JOG后:
289 root@linaro-ubuntu-desktop:~# ethercat upload -a 255 -p 0 -t uint16 0x6041 0x00
290 0x0537 1335
291 命令行使能后:
292 ethercat upload -a 255 -p 0 -t uint16 0x6041 0x00
293 0x0737 1847
294
295
296 0x6076 额定输出力矩值
297 最大力矩 = 额定输出力矩值 * 3
298 泰科模组Ethercat调试
299
300
301 可以正常以力矩模式控制泰科电机的命令序列
302 0.设置自由模式
303 ethercat download -a 255 -p 0 -t uint16 0x1c32 0x01 0x00
304 ethercat states -a 255 -p 0 OP
305
306 1.设置力矩模式
307 ethercat download -a 255 -p 0 -t uint8 0x6060 0x00 0x04
308
309 2.设置力矩上升斜率
310 ethercat download -a 255 -p 0 -t int32 0x6087 0x00 1000
311
312 3.伺服ON
313 ethercat download -a 255 -p 0 -t uint16 0x6040 0x00 0x06
314 ethercat download -a 255 -p 0 -t uint16 0x6040 0x00 0x07
315 ethercat download -a 255 -p 0 -t uint16 0x6040 0x00 0x0f
316
317 4. 设置输出目标力矩,电机旋转
318 ethercat download -a 255 -p 0 -t int16 0x6071 0x00 100
319
320 5.输出力矩0电机
321 ethercat download -a 255 -p 0 -t int16 0x6071 0x00 0
322
323
324 关于OP模式的几点要点
325 1.进行PDO映射的对象字典要在OP模式下更改才会生效,否则可能即使更改成功了也会不起作用
326 2.没有进行PDO映射的对象字典在OP模式下读写可能会发生错误
327
328
329 对象字典使用PDO还是SDO方式进行操作?
330 SDO一般用于配置从站信息,PDO方式一般用于同从站进行实时数据交互,SDO方式主从站需要交互确认机制因而费时。
331
332
333
334
335 EthercatIO板调试
336
337 root@linaro-ubuntu-desktop:~# ethercat pdos -a 0 -p 0
338 SM0: PhysAddr 0x1000, DefaultSize 128, ControlRegister 0x26, Enable 1
```

```
339 SM1: PhysAddr 0x1000, DefaultSize 128, ControlRegister 0x22, Enable 1
340 SM2: PhysAddr 0x1100, DefaultSize 2, ControlRegister 0x64, Enable 1
341 RxPDO 0x1001 "DO RxPDO-Map"
342     PDO entry 0x7010:01, 1 bit, "LED 1"
343     PDO entry 0x7010:02, 1 bit, "LED 2"
344     PDO entry 0x7010:03, 1 bit, "LED 3"
345     PDO entry 0x7010:04, 1 bit, "LED 4"
346     PDO entry 0x7010:05, 1 bit, "LED 5"
347     PDO entry 0x7010:06, 1 bit, "LED 6"
348     PDO entry 0x7010:07, 1 bit, "LED 7"
349     PDO entry 0x7010:08, 1 bit, "LED 8"
350     PDO entry 0x0000:00, 8 bit, "Gap"
351 SM3: PhysAddr 0x1400, DefaultSize 6, ControlRegister 0x20, Enable 1
352 TxPDO 0x1400 "DI TxPDO-Map"
353     PDO entry 0x6000:01, 1 bit, "Switch 1"
354     PDO entry 0x6000:02, 1 bit, "Switch 2"
355     PDO entry 0x6000:03, 1 bit, "Switch 3"
356     PDO entry 0x6000:04, 1 bit, "Switch 4"
357     PDO entry 0x6000:05, 1 bit, "Switch 5"
358     PDO entry 0x6000:06, 1 bit, "Switch 6"
359     PDO entry 0x6000:07, 1 bit, "Switch 7"
360     PDO entry 0x6000:08, 1 bit, "Switch 8"
361     PDO entry 0x0000:00, 8 bit, "Gap"
362 TxPDO 0x1402 "AI TxPDO-Map"
363     PDO entry 0x6020:01, 1 bit, "Underrange"
364     PDO entry 0x6020:02, 1 bit, "Overrange"
365     PDO entry 0x6020:03, 2 bit, "Limit 1"
366     PDO entry 0x6020:05, 2 bit, "Limit 2"
367     PDO entry 0x0000:00, 8 bit, "Gap"
368     PDO entry 0x1802:07, 1 bit, "TxPDOState"
369     PDO entry 0x1802:09, 1 bit, "TxPDO Toggle"
370     PDO entry 0x6020:11, 16 bit, "Analog input"
371     PDO entry 0x6020:12, 16 bit, "Temp"
372     PDO entry 0x6020:13, 16 bit, "Hum"
373
374 ~~~~~
375 PDO方式控制问题的解决:
376 对象字典0x0000:0x00 为填充对象, 不能在ecrt_domain_reg_pdo_entry_list注册, 也就是不能出现在ec_pdo_entry_reg_t board0_
377 但是为了匹配队形E2PROM中PDO映射关系要在ec_pdo_entry_info_t slave_0_pdo_entries[]定义
378
379 PDO数据命令行显示:
380 ethercat data -a 0 -p 0 | hd
381 ethercat domains -a 0 -p 0 -v //更详细会显示哪个PDO管理器
382
383 const static ec_pdo_entry_reg_t board0_domain_regs[] = {
384     /*Leds*/
385     {
386         IO_BOARD_0_ALIAS,           //别名
387         IO_BOARD_0_POSITION,        //位置
388         IO_BOARD_0_VENDORID,        //厂商id
389         IO_BOARD_0_PRODUCTID,       //产品id
390         0x7010,                     //索引
391         0x01, //子索引
392         &off_bytes_board0_led1_val, //PDO入口在process_data字节偏移量
393         &off_bits_board0_led1_val
394     },
395     {
396         IO_BOARD_0_ALIAS,           //别名
397         IO_BOARD_0_POSITION,        //位置
398         IO_BOARD_0_VENDORID,        //厂商id
399         IO_BOARD_0_PRODUCTID,       //产品id
400         0x7010,                     //索引
401         0x02, //子索引
402         &off_bytes_board0_led2_val, //PDO入口在process_data字节偏移量
403         &off_bits_board0_led2_val
404     },
405     {
406         IO_BOARD_0_ALIAS,           //别名
407         IO_BOARD_0_POSITION,        //位置
408         IO_BOARD_0_VENDORID,        //厂商id
409         IO_BOARD_0_PRODUCTID,       //产品id
410         0x7010,                     //索引
411         0x03, //子索引
412         &off_bytes_board0_led3_val, //PDO入口在process_data字节偏移量
413         &off_bits_board0_led3_val
414     },
415     {
416         IO_BOARD_0_ALIAS,           //别名
417         IO_BOARD_0_POSITION,        //位置
418         IO_BOARD_0_VENDORID,        //厂商id
419         IO_BOARD_0_PRODUCTID,       //产品id
420         0x7010,                     //索引
421         0x04, //子索引
422         &off_bytes_board0_led4_val, //PDO入口在process_data字节偏移量
423         &off_bits_board0_led4_val
424     },
425     {
426         IO_BOARD_0_ALIAS,           //别名
427         IO_BOARD_0_POSITION,        //位置
428         IO_BOARD_0_VENDORID,        //厂商id
429         IO_BOARD_0_PRODUCTID,       //产品id
430         0x7010,                     //索引
431         0x05, //子索引
432         &off_bytes_board0_led5_val, //PDO入口在process_data字节偏移量
433         &off_bits_board0_led5_val
434     },
435     {
436         IO_BOARD_0_ALIAS,           //别名
437         IO_BOARD_0_POSITION,        //位置
438         IO_BOARD_0_VENDORID,        //厂商id
439         IO_BOARD_0_PRODUCTID,       //产品id
440         0x7010,                     //索引
441         0x06, //子索引
442         &off_bytes_board0_led6_val, //PDO入口在process_data字节偏移量
443         &off_bits_board0_led6_val
444     },
445     {
446         IO_BOARD_0_ALIAS,           //别名
447         IO_BOARD_0_POSITION,        //位置
448         IO_BOARD_0_VENDORID,        //厂商id
449         IO_BOARD_0_PRODUCTID,       //产品id
450         0x7010,                     //索引
451         0x07, //子索引
452         &off_bytes_board0_led7_val, //PDO入口在process_data字节偏移量
453         &off_bits_board0_led7_val
454     },
455     {
456         IO_BOARD_0_ALIAS,           //别名
457         IO_BOARD_0_POSITION,        //位置
458         IO_BOARD_0_VENDORID,        //厂商id
459         IO_BOARD_0_PRODUCTID,       //产品id
460         0x7010,                     //索引
461         0x08, //子索引
```

```
462      &off_bytes_board0_led8_val,          //PDO入口在process_data字节偏移量
463      &off_bits_board0_led8_val
464  },
465
466  //一定要注意, 对象字典0x0000:0x00 为填充对象, 不能注册
467  /*
468  {
469      IO_BOARD_0_ALIAS,          //别名
470      IO_BOARD_0_POSITION,      //位置
471      IO_BOARD_0_VENDORID,      //厂商id
472      IO_BOARD_0_PRODUCTID,     //产品id
473      0x0000,                   //索引
474      0x00, //子索引
475      &off_bytes_board0_ledgap,   //PDO入口在process_data字节偏移量
476      &off_bits_board0_ledgap
477  },
478  */
479
480
481  /*switchs*/
482  {
483      IO_BOARD_0_ALIAS,          //别名
484      IO_BOARD_0_POSITION,      //位置
485      IO_BOARD_0_VENDORID,      //厂商id
486      IO_BOARD_0_PRODUCTID,     //产品id
487      0x6000,                   //索引
488      0x01, //子索引
489      &off_bytes_board0_switch1_status, //PDO入口在process_data字节偏移量
490      &off_bits_board0_switch1_status
491  },
492  {
493      IO_BOARD_0_ALIAS,          //别名
494      IO_BOARD_0_POSITION,      //位置
495      IO_BOARD_0_VENDORID,      //厂商id
496      IO_BOARD_0_PRODUCTID,     //产品id
497      0x6000,                   //索引
498      0x02, //子索引
499      &off_bytes_board0_switch2_status, //PDO入口在process_data字节偏移量
500      &off_bits_board0_switch2_status
501  },
502  {
503      IO_BOARD_0_ALIAS,          //别名
504      IO_BOARD_0_POSITION,      //位置
505      IO_BOARD_0_VENDORID,      //厂商id
506      IO_BOARD_0_PRODUCTID,     //产品id
507      0x6000,                   //索引
508      0x03, //子索引
509      &off_bytes_board0_switch3_status, //PDO入口在process_data字节偏移量
510      &off_bits_board0_switch3_status
511  },
512  {
513      IO_BOARD_0_ALIAS,          //别名
514      IO_BOARD_0_POSITION,      //位置
515      IO_BOARD_0_VENDORID,      //厂商id
516      IO_BOARD_0_PRODUCTID,     //产品id
517      0x6000,                   //索引
518      0x04, //子索引
519      &off_bytes_board0_switch4_status, //PDO入口在process_data字节偏移量
520      &off_bits_board0_switch4_status
521  },
522  {
523      IO_BOARD_0_ALIAS,          //别名
524      IO_BOARD_0_POSITION,      //位置
525      IO_BOARD_0_VENDORID,      //厂商id
526      IO_BOARD_0_PRODUCTID,     //产品id
527      0x6000,                   //索引
528      0x05, //子索引
529      &off_bytes_board0_switch5_status, //PDO入口在process_data字节偏移量
530      &off_bits_board0_switch5_status
531  },
532  {
533      IO_BOARD_0_ALIAS,          //别名
534      IO_BOARD_0_POSITION,      //位置
535      IO_BOARD_0_VENDORID,      //厂商id
536      IO_BOARD_0_PRODUCTID,     //产品id
537      0x6000,                   //索引
538      0x06, //子索引
539      &off_bytes_board0_switch6_status, //PDO入口在process_data字节偏移量
540      &off_bits_board0_switch6_status
541  },
542  {
543      IO_BOARD_0_ALIAS,          //别名
544      IO_BOARD_0_POSITION,      //位置
545      IO_BOARD_0_VENDORID,      //厂商id
546      IO_BOARD_0_PRODUCTID,     //产品id
547      0x6000,                   //索引
548      0x07, //子索引
549      &off_bytes_board0_switch7_status, //PDO入口在process_data字节偏移量
550      &off_bits_board0_switch7_status
551  },
552  {
553      IO_BOARD_0_ALIAS,          //别名
554      IO_BOARD_0_POSITION,      //位置
555      IO_BOARD_0_VENDORID,      //厂商id
556      IO_BOARD_0_PRODUCTID,     //产品id
557      0x6000,                   //索引
558      0x08, //子索引
559      &off_bytes_board0_switch8_status, //PDO入口在process_data字节偏移量
560      &off_bits_board0_switch8_status
561  },
562
563  //一定要注意, 对象字典0x0000:0x00 为填充对象, 不能注册
564  /*
565  {
566      IO_BOARD_0_ALIAS,          //别名
567      IO_BOARD_0_POSITION,      //位置
568      IO_BOARD_0_VENDORID,      //厂商id
569      IO_BOARD_0_PRODUCTID,     //产品id
570      0x0000,                   //索引
571      0x00, //子索引
572      &off_bytes_board0_switchgap, //PDO入口在process_data字节偏移量
573      &off_bits_board0_switchgap
574  },
575  */
576
577  {
578      IO_BOARD_0_ALIAS,          //别名
579      IO_BOARD_0_POSITION,      //位置
580      IO_BOARD_0_VENDORID,      //厂商id
581      IO_BOARD_0_PRODUCTID,     //产品id
582      0x6020,                   //索引
583      0x01, //子索引
584      &off_bytes_board0_underrange_status, //PDO入口在process_data字节偏移量
```

```
585     &off_bits_board0_underrange_status
586 },
587 {
588     IO_BOARD_0_ALIAS,          //别名
589     IO_BOARD_0_POSITION,       //位置
590     IO_BOARD_0_VENDORID,       //厂商id
591     IO_BOARD_0_PRODUCTID,      //产品id
592     0x6020,                    //索引
593     0x02, //子索引
594     &off_bytes_board0_underrange_status, //PDO入口在process_data字节偏移量
595     &off_bits_board0_underrange_status
596 },
597 {
598     IO_BOARD_0_ALIAS,          //别名
599     IO_BOARD_0_POSITION,       //位置
600     IO_BOARD_0_VENDORID,       //厂商id
601     IO_BOARD_0_PRODUCTID,      //产品id
602     0x6020,                    //索引
603     0x03, //子索引
604     &off_bytes_board0_limit1_status, //PDO入口在process_data字节偏移量
605     &off_bits_board0_limit1_status
606 },
607
608 {
609     IO_BOARD_0_ALIAS,          //别名
610     IO_BOARD_0_POSITION,       //位置
611     IO_BOARD_0_VENDORID,       //厂商id
612     IO_BOARD_0_PRODUCTID,      //产品id
613     0x6020,                    //索引
614     0x05, //子索引
615     &off_bytes_board0_limit2_status, //PDO入口在process_data字节偏移量
616     &off_bits_board0_limit2_status
617 },
618
619 //一定要注意,对象字典0x0000:0x00 为填充对象,不能注册
620 /*
621 {
622     IO_BOARD_0_ALIAS,          //别名
623     IO_BOARD_0_POSITION,       //位置
624     IO_BOARD_0_VENDORID,       //厂商id
625     IO_BOARD_0_PRODUCTID,      //产品id
626     0x0000,                    //索引
627     0x00, //子索引
628     &off_bytes_board0_limitgap, //PDO入口在process_data字节偏移量
629     &off_bits_board0_limitgap
630 },
631 */
632
633 {
634     IO_BOARD_0_ALIAS,          //别名
635     IO_BOARD_0_POSITION,       //位置
636     IO_BOARD_0_VENDORID,       //厂商id
637     IO_BOARD_0_PRODUCTID,      //产品id
638     0x1802,                    //索引
639     0x07, //子索引
640     &off_bytes_board0_TxPDOState_status, //PDO入口在process_data字节偏移量
641     &off_bits_board0_TxPDOState_status
642 },
643
644
645 {
646     IO_BOARD_0_ALIAS,          //别名
647     IO_BOARD_0_POSITION,       //位置
648     IO_BOARD_0_VENDORID,       //厂商id
649     IO_BOARD_0_PRODUCTID,      //产品id
650     0x1802,                    //索引
651     0x09, //子索引
652     &off_bytes_board0_TxPDO_Toggle_status, //PDO入口在process_data字节偏移量
653     &off_bits_board0_TxPDO_Toggle_status
654 },
655
656 {
657     IO_BOARD_0_ALIAS,          //别名
658     IO_BOARD_0_POSITION,       //位置
659     IO_BOARD_0_VENDORID,       //厂商id
660     IO_BOARD_0_PRODUCTID,      //产品id
661     0x6020,                    //索引
662     0x11, //子索引
663     &off_bytes_board0_Analog_input_status, //PDO入口在process_data字节偏移量
664     &off_bits_board0_Analog_input_status
665 },
666
667
668 {
669     IO_BOARD_0_ALIAS,          //别名
670     IO_BOARD_0_POSITION,       //位置
671     IO_BOARD_0_VENDORID,       //厂商id
672     IO_BOARD_0_PRODUCTID,      //产品id
673     0x6020,                    //索引
674     0x12, //子索引
675     &off_bytes_board0_Temp_val, //PDO入口在process_data字节偏移量
676     &off_bits_board0_Temp_val
677 },
678
679 {
680     IO_BOARD_0_ALIAS,          //别名
681     IO_BOARD_0_POSITION,       //位置
682     IO_BOARD_0_VENDORID,       //厂商id
683     IO_BOARD_0_PRODUCTID,      //产品id
684     0x6020,                    //索引
685     0x13, //子索引
686     &off_bytes_board0_Hum_val, //PDO入口在process_data字节偏移量
687     &off_bits_board0_Hum_val
688 },
689 {
690 }
691 };
692
693 ec_pdo_entry_info_t slave_0_pdo_entries[] = {
694     {0x7010, 0x01, 1}, /* LED 1 */
695     {0x7010, 0x02, 1}, /* LED 2 */
696     {0x7010, 0x03, 1}, /* LED 3 */
697     {0x7010, 0x04, 1}, /* LED 4 */
698     {0x7010, 0x05, 1}, /* LED 5 */
699     {0x7010, 0x06, 1}, /* LED 6 */
700     {0x7010, 0x07, 1}, /* LED 7 */
701     {0x7010, 0x08, 1}, /* LED 8 */
702     {0x0000, 0x00, 8}, /* Gap */ //一定要注意,对象字典0x0000:0x00 为填充对象,不能在ecrt_domain_reg_pdo_entry_list中
703     {0x6000, 0x01, 1}, /* Switch 1 */
704     {0x6000, 0x02, 1}, /* Switch 2 */
705     {0x6000, 0x03, 1}, /* Switch 3 */
706     {0x6000, 0x04, 1}, /* Switch 4 */
707     {0x6000, 0x05, 1}, /* Switch 5 */
708 }
```

```
700 {0x6000, 0x00, 1}, /* Switch 6 */
701 {0x6000, 0x07, 1}, /* Switch 7 */
702 {0x6000, 0x08, 1}, /* Switch 8 */
703 {0x0000, 0x00, 8}, /* Gap */ //一定要注意, 对象字典0x0000:0x00 为填充对象, 不能在ecrt_domain_reg_pdo_entry_list中
704 {0x6010, 0x01, 1}, /* Underrange */
705 {0x6020, 0x02, 1}, /* Overrange */
706 {0x6020, 0x03, 2}, /* Limit 1 */
707 {0x6020, 0x05, 2}, /* Limit 2 */
708 {0x0000, 0x00, 8}, /* Gap */ //一定要注意, 对象字典0x0000:0x00 为填充对象, 不能在ecrt_domain_reg_pdo_entry_list中
709 {0x1802, 0x07, 1}, /* TxPDOState */
710 {0x1802, 0x09, 1}, /* TxPDO Toggle */
711 {0x6010, 0x11, 16}, /* Analog input */
712 {0x6020, 0x12, 16}, /* Temp */
713 {0x6020, 0x13, 16}, /* Hum */
714 };
715
716 -----
717 root@linaro-ubuntu-desktop:~# ethercat sdos
718 SDO 0x1000, "Device type"
719   0x1000:00, r-r-r-r-, uint32, 32 bit, "Device type"
720 SDO 0x1001, "Error register"
721   0x1001:00, r-r-r-r-, uint8, 8 bit, "Error register"
722 SDO 0x1008, "Device name"
723   0x1008:00, r-r-r-r-, string, 80 bit, "Device name"
724 SDO 0x1009, "Hardware version"
725   0x1009:00, r-r-r-r-, string, 32 bit, "Hardware version"
726 SDO 0x100a, "Software version"
727   0x100a:00, r-r-r-r-, string, 32 bit, "Software version"
728 SDO 0x1018, "Identity"
729   0x1018:00, r-r-r-r-, uint8, 8 bit, "SubIndex 000"
730   0x1018:01, r-r-r-r-, uint32, 32 bit, "Vendor ID"
731   0x1018:02, r-r-r-r-, uint32, 32 bit, "Product code"
732   0x1018:03, r-r-r-r-, uint32, 32 bit, "Revision"
733   0x1018:04, r-r-r-r-, uint32, 32 bit, "Serial number"
734 SDO 0x10f1, "Error Settings"
735   0x10f1:00, r-r-r-r-, uint8, 8 bit, "SubIndex 000"
736   0x10f1:01, r-r-r-r-w, uint32, 32 bit, "Local Error Reaction"
737   0x10f1:02, r-r-r-r-w, uint16, 16 bit, "Sync Error Counter Limit"
738 SDO 0x1601, "DO RxPDO-Map"
739   0x1601:00, r-r-r-r-, uint8, 8 bit, "SubIndex 000"
740   0x1601:01, r-r-r-r-, uint32, 32 bit, "SubIndex 001"
741   0x1601:02, r-r-r-r-, uint32, 32 bit, "SubIndex 002"
742   0x1601:03, r-r-r-r-, uint32, 32 bit, "SubIndex 003"
743   0x1601:04, r-r-r-r-, uint32, 32 bit, "SubIndex 004"
744   0x1601:05, r-r-r-r-, uint32, 32 bit, "SubIndex 005"
745   0x1601:06, r-r-r-r-, uint32, 32 bit, "SubIndex 006"
746   0x1601:07, r-r-r-r-, uint32, 32 bit, "SubIndex 007"
747   0x1601:08, r-r-r-r-, uint32, 32 bit, "SubIndex 008"
748   0x1601:09, r-r-r-r-, uint32, 32 bit, "SubIndex 009"
749 SDO 0x1802, "TxPDO Parameter"
750   0x1802:00, r-r-r-r-, uint8, 8 bit, "SubIndex 000"
751   0x1802:01, -----, type 0000, 0 bit, "SubIndex 001"
752   0x1802:02, -----, type 0000, 0 bit, "SubIndex 002"
753   0x1802:03, -----, type 0000, 0 bit, "SubIndex 003"
754   0x1802:04, -----, type 0000, 0 bit, "SubIndex 004"
755   0x1802:05, -----, type 0000, 0 bit, "SubIndex 005"
756   0x1802:06, r-r-r-r-, octet_string, 0 bit, "Exclude TxPDOs"
757   0x1802:07, r-r-r-r-, bool, 1 bit, "TxPDOState"
758   0x1802:08, -----, type 0000, 0 bit, "SubIndex 008"
759   0x1802:09, r-r-r-r-, bool, 1 bit, "TxPDO Toggle"
760 SDO 0x1a00, "DI TxPDO-Map"
761   0x1a00:00, r-r-r-r-, uint8, 8 bit, "SubIndex 000"
762   0x1a00:01, r-r-r-r-, uint32, 32 bit, "SubIndex 001"
763   0x1a00:02, r-r-r-r-, uint32, 32 bit, "SubIndex 002"
764   0x1a00:03, r-r-r-r-, uint32, 32 bit, "SubIndex 003"
765   0x1a00:04, r-r-r-r-, uint32, 32 bit, "SubIndex 004"
766   0x1a00:05, r-r-r-r-, uint32, 32 bit, "SubIndex 005"
767   0x1a00:06, r-r-r-r-, uint32, 32 bit, "SubIndex 006"
768   0x1a00:07, r-r-r-r-, uint32, 32 bit, "SubIndex 007"
769   0x1a00:08, r-r-r-r-, uint32, 32 bit, "SubIndex 008"
770   0x1a00:09, r-r-r-r-, uint32, 32 bit, "SubIndex 009"
771 SDO 0x1a02, "AI TxPDO-Map"
772   0x1a02:00, r-r-r-r-, uint8, 8 bit, "SubIndex 000"
773   0x1a02:01, r-r-r-r-, uint32, 32 bit, "SubIndex 001"
774   0x1a02:02, r-r-r-r-, uint32, 32 bit, "SubIndex 002"
775   0x1a02:03, r-r-r-r-, uint32, 32 bit, "SubIndex 003"
776   0x1a02:04, r-r-r-r-, uint32, 32 bit, "SubIndex 004"
777   0x1a02:05, r-r-r-r-, uint32, 32 bit, "SubIndex 005"
778   0x1a02:06, r-r-r-r-, uint32, 32 bit, "SubIndex 006"
779   0x1a02:07, r-r-r-r-, uint32, 32 bit, "SubIndex 007"
780   0x1a02:08, r-r-r-r-, uint32, 32 bit, "SubIndex 008"
781   0x1a02:09, r-r-r-r-, uint32, 32 bit, "SubIndex 009"
782   0x1a02:0a, r-r-r-r-, uint32, 32 bit, "SubIndex 010"
783 SDO 0x1c00, "Sync manager type"
784   0x1c00:00, r-r-r-r-, uint8, 8 bit, "SubIndex 000"
785   0x1c00:01, r-r-r-r-, uint8, 8 bit, "SubIndex 001"
786   0x1c00:02, r-r-r-r-, uint8, 8 bit, "SubIndex 002"
787   0x1c00:03, r-r-r-r-, uint8, 8 bit, "SubIndex 003"
788   0x1c00:04, r-r-r-r-, uint8, 8 bit, "SubIndex 004"
789 SDO 0x1c12, "RxPDO assign"
790   0x1c12:00, r-r-r-r-, uint8, 8 bit, "SubIndex 000"
791   0x1c12:01, r-r-r-r-, uint16, 16 bit, "SubIndex 001"
792 SDO 0x1c13, "TxPDO assign"
793   0x1c13:00, r-r-r-r-, uint8, 8 bit, "SubIndex 000"
794   0x1c13:01, r-r-r-r-, uint16, 16 bit, "SubIndex 001"
795   0x1c13:02, r-r-r-r-, uint16, 16 bit, "SubIndex 002"
796 SDO 0x1c32, "SM output parameter"
797   0x1c32:00, r-r-r-r-, uint8, 8 bit, "SubIndex 000"
798   0x1c32:01, r-r-r-r-w, uint16, 16 bit, "Synchronization Type"
799   0x1c32:02, r-r-r-r-, uint32, 32 bit, "Cycle Time"
800   0x1c32:03, -----, type 0000, 32 bit, "SubIndex 003"
801   0x1c32:04, r-r-r-r-, uint16, 16 bit, "Synchronization Types supported"
802   0x1c32:05, r-r-r-r-, uint32, 32 bit, "Minimum Cycle Time"
803   0x1c32:06, r-r-r-r-, uint32, 32 bit, "Calc and Copy Time"
804   0x1c32:07, -----, type 0000, 32 bit, "SubIndex 007"
805   0x1c32:08, r-r-r-r-w, uint16, 16 bit, "Get Cycle Time"
806   0x1c32:09, r-r-r-r-, uint32, 32 bit, "Delay Time"
807   0x1c32:0a, r-r-r-r-w, uint32, 32 bit, "Sync0 Cycle Time"
808   0x1c32:0b, r-r-r-r-, uint16, 16 bit, "SM-Event Missed"
809   0x1c32:0c, r-r-r-r-, uint16, 16 bit, "Cycle Time Too Small"
810   0x1c32:0d, -----, type 0000, 16 bit, "Shift Time Too Short"
811   0x1c32:0e, -----, type 0000, 16 bit, "SubIndex 014"
812   0x1c32:0f, -----, type 0000, 32 bit, "SubIndex 015"
813   0x1c32:10, -----, type 0000, 32 bit, "SubIndex 016"
814   0x1c32:11, -----, type 0000, 32 bit, "SubIndex 017"
815   0x1c32:12, -----, type 0000, 32 bit, "SubIndex 018"
816   0x1c32:13, -----, type 0000, 0 bit, "SubIndex 019"
817   0x1c32:14, -----, type 0000, 0 bit, "SubIndex 020"
818   0x1c32:15, -----, type 0000, 0 bit, "SubIndex 021"
819   0x1c32:16, -----, type 0000, 0 bit, "SubIndex 022"
820   0x1c32:17, -----, type 0000, 0 bit, "SubIndex 023"
821   0x1c32:18, -----, type 0000, 0 bit, "SubIndex 024"
822   0x1c32:19, -----, type 0000, 0 bit, "SubIndex 025"
823   0x1c32:1a, -----, type 0000, 0 bit, "SubIndex 026"
```

```
831 0x1c32:1b, -----, type 0000, 0 bit, "SubIndex 027"
832 0x1c32:1c, -----, type 0000, 0 bit, "SubIndex 028"
833 0x1c32:1d, -----, type 0000, 0 bit, "SubIndex 029"
834 0x1c32:1e, -----, type 0000, 0 bit, "SubIndex 030"
835 0x1c32:1f, -----, type 0000, 0 bit, "SubIndex 031"
836 0x1c32:20, r-r-r-, bool, 1 bit, "Sync Error"
837 SDO 0x1c33, "SM input parameter"
838 0x1c33:00, r-r-r-, uint8, 8 bit, "SubIndex 000"
839 0x1c33:01, rwr-r-, uint16, 16 bit, "Synchronization Type"
840 0x1c33:02, r-r-r-, uint32, 32 bit, "Cycle Time"
841 0x1c33:03, -----, type 0000, 32 bit, "SubIndex 003"
842 0x1c33:04, r-r-r-, uint16, 16 bit, "Synchronization Types supported"
843 0x1c33:05, r-r-r-, uint32, 32 bit, "Minimum Cycle Time"
844 0x1c33:06, r-r-r-, uint32, 32 bit, "Calc and Copy Time"
845 0x1c33:07, -----, type 0000, 32 bit, "SubIndex 007"
846 0x1c33:08, rwrwrw, uint16, 16 bit, "Get Cycle Time"
847 0x1c33:09, r-r-r-, uint32, 32 bit, "Delay Time"
848 0x1c33:0a, rwrwrw, uint32, 32 bit, "Sync0 Cycle Time"
849 0x1c33:0b, r-r-r-, uint16, 16 bit, "SM-Event Missed"
850 0x1c33:0c, r-r-r-, uint16, 16 bit, "Cycle Time Too Small"
851 0x1c33:0d, -----, type 0000, 16 bit, "Shift Time Too Short"
852 0x1c33:0e, -----, type 0000, 16 bit, "SubIndex 014"
853 0x1c33:0f, -----, type 0000, 32 bit, "SubIndex 015"
854 0x1c33:10, -----, type 0000, 32 bit, "SubIndex 016"
855 0x1c33:11, -----, type 0000, 32 bit, "SubIndex 017"
856 0x1c33:12, -----, type 0000, 32 bit, "SubIndex 018"
857 0x1c33:13, -----, type 0000, 0 bit, "SubIndex 019"
858 0x1c33:14, -----, type 0000, 0 bit, "SubIndex 020"
859 0x1c33:15, -----, type 0000, 0 bit, "SubIndex 021"
860 0x1c33:16, -----, type 0000, 0 bit, "SubIndex 022"
861 0x1c33:17, -----, type 0000, 0 bit, "SubIndex 023"
862 0x1c33:18, -----, type 0000, 0 bit, "SubIndex 024"
863 0x1c33:19, -----, type 0000, 0 bit, "SubIndex 025"
864 0x1c33:1a, -----, type 0000, 0 bit, "SubIndex 026"
865 0x1c33:1b, -----, type 0000, 0 bit, "SubIndex 027"
866 0x1c33:1c, -----, type 0000, 0 bit, "SubIndex 028"
867 0x1c33:1d, -----, type 0000, 0 bit, "SubIndex 029"
868 0x1c33:1e, -----, type 0000, 0 bit, "SubIndex 030"
869 0x1c33:1f, -----, type 0000, 0 bit, "SubIndex 031"
870 0x1c33:20, r-r-r-, bool, 1 bit, "Sync Error"
871 SDO 0x6000, "DI Inputs"
872 0x6000:00, r-r-r-, uint8, 8 bit, "SubIndex 000"
873 0x6000:01, r-r-r-, bool, 1 bit, "Switch 1"
874 0x6000:02, r-r-r-, bool, 1 bit, "Switch 2"
875 0x6000:03, r-r-r-, bool, 1 bit, "Switch 3"
876 0x6000:04, r-r-r-, bool, 1 bit, "Switch 4"
877 0x6000:05, r-r-r-, bool, 1 bit, "Switch 5"
878 0x6000:06, r-r-r-, bool, 1 bit, "Switch 6"
879 0x6000:07, r-r-r-, bool, 1 bit, "Switch 7"
880 0x6000:08, r-r-r-, bool, 1 bit, "Switch 8"
881 SDO 0x6020, "AI Inputs"
882 0x6020:00, r-r-r-, uint8, 8 bit, "SubIndex 000"
883 0x6020:01, r-r-r-, bool, 1 bit, "Underrange"
884 0x6020:02, r-r-r-, bool, 1 bit, "Overrange"
885 0x6020:03, r-r-r-, type 0031, 2 bit, "Limit 1"
886 0x6020:04, -----, type 0000, 0 bit, "SubIndex 004"
887 0x6020:05, r-r-r-, type 0031, 2 bit, "Limit 2"
888 0x6020:06, -----, type 0000, 2 bit, "SubIndex 006"
889 0x6020:07, -----, type 0000, 6 bit, "SubIndex 007"
890 0x6020:08, -----, type 0000, 0 bit, "SubIndex 008"
891 0x6020:09, -----, type 0000, 0 bit, "SubIndex 009"
892 0x6020:0a, -----, type 0000, 0 bit, "SubIndex 010"
893 0x6020:0b, -----, type 0000, 0 bit, "SubIndex 011"
894 0x6020:0c, -----, type 0000, 0 bit, "SubIndex 012"
895 0x6020:0d, -----, type 0000, 0 bit, "SubIndex 013"
896 0x6020:0e, -----, type 0000, 0 bit, "SubIndex 014"
897 0x6020:0f, r-r-r-, bool, 1 bit, "TxPDO State"
898 0x6020:10, r-r-r-, bool, 1 bit, "TxPDO Toggle"
899 0x6020:11, r-r-r-, int16, 16 bit, "Analog input"
900 0x6020:12, r-r-r-, int16, 16 bit, "Temp"
901 0x6020:13, r-r-r-, int16, 16 bit, "Hum"
902 SDO 0x7010, "DO Outputs"
903 0x7010:00, rwrwrw, uint8, 8 bit, "SubIndex 000"
904 0x7010:01, rwrwrw, bool, 1 bit, "LED 1"
905 0x7010:02, rwrwrw, bool, 1 bit, "LED 2"
906 0x7010:03, rwrwrw, bool, 1 bit, "LED 3"
907 0x7010:04, rwrwrw, bool, 1 bit, "LED 4"
908 0x7010:05, rwrwrw, bool, 1 bit, "LED 5"
909 0x7010:06, rwrwrw, bool, 1 bit, "LED 6"
910 0x7010:07, rwrwrw, bool, 1 bit, "LED 7"
911 0x7010:08, rwrwrw, bool, 1 bit, "LED 8"
912 SDO 0x8020, "AI Settings"
913 0x8020:00, r-r-r-, uint8, 8 bit, "SubIndex 000"
914 0x8020:01, rwrwrw, bool, 1 bit, "Enable user scale"
915 0x8020:02, rwrwrw, type 0000, 3 bit, "Presentation"
916 0x8020:03, -----, type 0000, 0 bit, "SubIndex 003"
917 0x8020:04, -----, type 0000, 0 bit, "SubIndex 004"
918 0x8020:05, -----, type 0000, 2 bit, "SubIndex 005"
919 0x8020:06, -----, type 0000, 0 bit, "SubIndex 006"
920 0x8020:07, rwrwrw, bool, 1 bit, "Enable limit 1"
921 0x8020:08, rwrwrw, bool, 1 bit, "Enable limit 2"
922 0x8020:09, -----, type 0000, 8 bit, "SubIndex 009"
923 0x8020:0a, -----, type 0000, 0 bit, "SubIndex 010"
924 0x8020:0b, -----, type 0000, 0 bit, "SubIndex 011"
925 0x8020:0c, -----, type 0000, 0 bit, "SubIndex 012"
926 0x8020:0d, -----, type 0000, 0 bit, "SubIndex 013"
927 0x8020:0e, -----, type 0000, 0 bit, "SubIndex 014"
928 0x8020:0f, -----, type 0000, 0 bit, "SubIndex 015"
929 0x8020:10, -----, type 0000, 0 bit, "SubIndex 016"
930 0x8020:11, rwrwrw, int16, 16 bit, "Offset"
931 0x8020:12, rwrwrw, int32, 32 bit, "Gain"
932 0x8020:13, rwrwrw, int16, 16 bit, "Limit 1"
933 0x8020:14, rwrwrw, int16, 16 bit, "Limit 2"
934 SDO 0xf000, "Modular device profile"
935 0xf000:00, r-r-r-, uint8, 8 bit, "SubIndex 000"
936 0xf000:01, r-r-r-, uint16, 16 bit, "Module index distance"
937 0xf000:02, r-r-r-, uint16, 16 bit, "Maximum number of modules"
938 SDO 0xf010, "Module profile list"
939 0xf010:00, r-r-r-, uint8, 8 bit, "SubIndex 000"
940 0xf010:01, r-r-r-, uint32, 32 bit, "SubIndex 001"
941 0xf010:02, r-r-r-, uint32, 32 bit, "SubIndex 002"
942 0xf010:03, r-r-r-, uint32, 32 bit, "SubIndex 003"
943
944
945 root@linaro-ubuntu-desktop:~# ethercat cstruct -a 0 -p 0
946 /* Master 0, Slave 0, "TempHum1"
947  * Vendor ID:      0x00000017
948  * Product code:   0x26483056
949  * Revision number: 0x00020111
950  */
951
952 ec_pdo_entry_info_t slave_0_pdo_entries[] = {
953     {0x7010, 0x01, 1}, /* LED 1 */

```



```
954 {0x7010, 0x02, 1}, /* LED 2 */
955 {0x7010, 0x03, 1}, /* LED 3 */
956 {0x7010, 0x04, 1}, /* LED 4 */
957 {0x7010, 0x05, 1}, /* LED 5 */
958 {0x7010, 0x06, 1}, /* LED 6 */
959 {0x7010, 0x07, 1}, /* LED 7 */
960 {0x7010, 0x08, 1}, /* LED 8 */
961 {0x0000, 0x00, 0}, /* Gap */
962 {0x6000, 0x01, 1}, /* Switch 1 */
963 {0x6000, 0x02, 1}, /* Switch 2 */
964 {0x6000, 0x03, 1}, /* Switch 3 */
965 {0x6000, 0x04, 1}, /* Switch 4 */
966 {0x6000, 0x05, 1}, /* Switch 5 */
967 {0x6000, 0x06, 1}, /* Switch 6 */
968 {0x6000, 0x07, 1}, /* Switch 7 */
969 {0x6000, 0x08, 1}, /* Switch 8 */
970 {0x0000, 0x00, 0}, /* Gap */
971 {0x6020, 0x01, 1}, /* Underrange */
972 {0x6020, 0x02, 1}, /* Overrange */
973 {0x6020, 0x03, 2}, /* Limit 1 */
974 {0x6020, 0x05, 2}, /* Limit 2 */
975 {0x0000, 0x00, 0}, /* Gap */
976 {0x1802, 0x07, 1}, /* TxPDOState */
977 {0x1802, 0x09, 1}, /* TxPDO Toggle */
978 {0x6020, 0x11, 16}, /* Analog input */
979 {0x6020, 0x12, 16}, /* Temp */
980 {0x6020, 0x13, 16}, /* Hum */
981 };
982
983 ec_pdo_info_t slave_0_pdos[] = {
984 {0x1601, 9, slave_0_pdo_entries + 0}, /* DO RxPDO-Map */
985 {0x1a00, 9, slave_0_pdo_entries + 9}, /* DI TxPDO-Map */
986 {0x1a02, 10, slave_0_pdo_entries + 18}, /* AI TxPDO-Map */
987 };
988
989 ec_sync_info_t slave_0_syncs[] = {
990 {0, EC_DIR_OUTPUT, 0, NULL, EC_WD_DISABLE},
991 {1, EC_DIR_INPUT, 0, NULL, EC_WD_DISABLE},
992 {2, EC_DIR_OUTPUT, 1, slave_0_pdos + 0, EC_WD_ENABLE},
993 {3, EC_DIR_INPUT, 2, slave_0_pdos + 1, EC_WD_DISABLE},
994 {0xff}
995 };
996
997 *****
998 wireshark 抓包分析
999 过滤器
1000 ecat
1001 ecat协议
1002
1003 ecat and ecat.cmd==0x0b
1004 ecat and ecat.cmd==0x0c
1005 ecat协议 且cmd=0x0b(LWR)
1006
1007 ecat_mailbox.coe.sdoreq
1008 过滤SDO请求
```

## 1. libethercat\std 目录

```
1 | 编译Igh生成的ethercat库
```

## 2. libethercat\include

```
1 | Igh Master Ethercat库包含的头文件
```

## 3. scripts/

```
1 | 自动生成从站信息的相关脚本文件，要生成从站信息，将该文件夹复制到开发板运行GenerateSlavesConfig.sh.
2 | 默认生成ec_common_configs_define.h和ec_common_configs_define.c文件，这两个文件会在libethercat\ec_common\ecat_common
```

## 4. libethercat\ec\_common

### 4.0

```
1 | ec_common_configs_define.h
2 | ec_common_configs_define.c
3 | 由脚本scripts/GenerateSlavesConfig.sh自动生成的从站信息，包括从站PDO,SDO设置
4 | 等，更具体的可以参照scripts/README.txt
```

### 4.1

```
1 | ecat_common.h
2 | ecat_common.c
3 | 基于libethercat\std 中的库的二次封装库，简化了PDO,SDO等操作
```

### 4.2

```
1 | ecat_common_intermediate_interface.h
2 | ecat_common_intermediate_interface.c
3 | 与具体厂商相关的接口库，基于ecat_common.c中接口的实现，不同设备修改这两个文件中的接口进行适配。
4 | 目前该文件实现的山羊电机接口。主要实现的PDO接口的电机状态查询，上电开机，关机，操作模式设置，力矩设置等接口，不同电机的类似
5 | 封装接口的关键是对象字典的操作，根据手册设置对象字典即可，标准的ethercat接口伺服电机一般来说对象字典定义基本是一样的，可能
```

## 5. 标准Igh Master接口的使用例子

### mytest目录

```
1 | mytest/test_torque_sanyo_ioc1
2 | 基于ioc1接口的sanyo电机测试例子
3 | mytest/test_torque_tec_ioc1
4 | 基于ioc1接口的泰科电机电机测试例子
5 | mytest/test_torque_tec_lib
6 | 基于libethercat\std库接口的电机测试例子
7 | mytest/test_torque2
8 | 基于libethercat\std库接口的电机测试例子2
9 | mytest/test_io_board
10 | ET1100,IO板的测试例子
11 | 基于标准的Igh 库及ioc1接口可以参照该文件夹例子
```

## 6.二次封装接口库ecat\_common\_intermediate\_interface.c的使用说明

### 6.1使用步骤

```
1 | (1)将编译生成的Igh库文件替换libethercat\std文件， 文件名可能要改成，或者不改也行，自己写Makefile时匹配库名称就行
```

```
2 (2)将 scripts/ 目录复制到开发运行GeneratesSlavesConfig.sh脚本，将脚本生成的ec_common_configs_define.h ec_common_conf
3 (3)基于二次封装接口的电机一般操作步骤
4 参照demo.c中的ethercat_init() 进行从站初始化
5 初始化后就可以调用ecat_common_intermediate_interface.c中接口对从站进行操作
6 比如
7 interpolation_2_ecat_set_slave_pwr_on()接口使能电机，
8 interpolation_2_ecat_set_slave_pwr_off() 关闭电机
9 interpolation_2_ecat_set_slave_target_pos() 设置目标位置，对应位置模式操作
10 其它接口作用见参照具体实现
```

## 7.关于轴操作的几点说明

```
1 (1) 每个轴对应一个从站，由alias，position确定，一般来说从站不多时alias=0固定不变，对不同轴根据positon确定。
2 例如使能和关闭不同的轴
3 int interpolation_2_ecat_set_slave_pwr_on(MasterSpecifiedInfo_T *master_specified_info, int slave_pos);
4 int interpolation_2_ecat_set_slave_pwr_off(MasterSpecifiedInfo_T *master_specified_info, int slave_pos);
5 中slave_pos参数就对应不同的轴，slave_pos=0,axis1 slave_pos=1,axis2...
6 (2) 设置不同轴的操作模式，位置，力矩，速度模式
7 int interpolation_2_ecat_set_slave_operation_model(MasterSpecifiedInfo_T *master_specified_info, int slave_
```

## Igh Master 1.5.2 源码

ethercat-1.5.2.zip

## Igh Master 1.5.2 ethercat-1.5.2/源码编译安装说明文件.txt

```
1 EtherCAT 1.5.2 编译及使用说明.txt
2 1.交叉编译源码
3 源码中有内核模块编译，指令定模块目录
4 根据实际情况修改build_ethercat-1.5.2.sh中以下几个变量：
5 output_dir='output' #编译输出目录
6 module_install_dir='module_install' #内核模块安装目录
7 kernel_source_dir='/mnt/fs_ext/lmx6/linux-3.0.35' #内核源码目录
8 lib_modules_kernel_prompt='3.0.35-2666-gbdde708' #内核版本号
9 host=arm-fsl-linux-gnueabi #交叉编译链前缀
10 注意要先编译内核，然后编译Ethercat因为Ethercat依赖于内核通用网卡模块
11 以root用户运行 ./build_ethercat-1.5.2.sh #编译具体根据提示选择y/n就可以了
12 -----
13 手动安装Ethercat到ARM开发板：
14 2.内核模块安装
15 2.1
16 将output文件下通lib_modules_kernel_prompt='3.0.35-2666-gbdde708'(build_ethercat-1.5.2.sh 中定义)放入开发板/lib/
17 2.2
18 开发板中执行"depmod"命令
19 3. output里面的其它文件放入开发板对应位置
20 4. 设置参数
21 修改/etc/sysconfig/ethercat
22 MASTER0_DEVICE="e4:f3:f5:c6:41:b6" #与ethercat绑定的Mac地址
23 DEVICE_MODULES="generic" #通用网卡就填generic，其余支持网卡换成模块名字就行
24 5.添加udev规则
25 echo KERNEL=="\EtherCAT[0-9]*\", MODE=\\"0664\\" > /etc/udev/rules.d/99-EtherCAT.rules
26 6. 启动服务
27 /etc/init.d/ethercat restart
28 出现以下信息表明移植成功
29 Shutting down EtherCAT master 1.5.2 done
30 Starting EtherCAT master 1.5.2 ec_generic: Binding socket to interface 3 (eth0).
31 done
32 7. 应用层测试
33 root@linaro-ubuntu-desktop:~# ethercat
34 Please specify a command!
35
36 Usage: ethercat <COMMAND> [OPTIONS] [ARGUMENTS]
37
38 Commands (can be abbreviated):
39 alias Write alias addresses.
40 config Show slave configurations.
41 cstruct Generate slave PDO information in C language.
42 data Output binary domain process data.
43 debug Set the master's debug level.
44 domains Show configured domains.
45 download Write an SDO entry to a slave.
46 eoe Display Ethernet over EtherCAT statistics.
47 foe_read Read a file from a slave via FoE.
48 foe_write Store a file on a slave via FoE.
49 graph Output the bus topology as a graph.
50 master Show master and Ethernet device information.
51 pdos List Sync managers, PDO assignment and mapping.
52 reg_read Output a slave's register contents.
53 reg_write Write data to a slave's registers.
54 rescan Rescan the bus.
55 sdos List SDO dictionaries.
56 sii_read Output a slave's SII contents.
57 sii_write Write SII contents to a slave.
58 slaves Display slaves on the bus.
59 soe_read Read an SoE IDN from a slave.
60 soe_write Write an SoE IDN to a slave.
61 states Request application-layer states.
62 upload Read an SDO entry from a slave.
63 version Show version information.
64 xml Generate slave information XML.
65
66 Global options:
67 --master -m <master> Comma separated list of masters
68 to select, ranges are allowed.
69 Examples: '1,3', '5-7,9', '-3'.
70 Default: '-' (all).
71 --force -f Force a command.
72 --quiet -q Output less information.
73 --verbose -v Output more information.
74 --help -h Show this help.
75
76 Numerical values can be specified either with decimal (no
77 prefix), octal (prefix '0') or hexadecimal (prefix '0x') base.
78
79 Call 'ethercat <COMMAND> --help' for command-specific help.
80 以上内容为基本的使用，进一步使用要结合ethercat说明文档和电机说明。
81 -----
82 自动安装Ethercat到ARM开发板：
83 将output目录复制到开发板然后运行output目录下install_to_arm.sh脚本
84 执行6，7步测试自动安装是否成功
85 -----
86 问题汇总：
87 (1)Starting EtherCAT master 1.5.2 EtherCAT ERROR: MAC address may not be empty.
88 FATAL: Error inserting ec_master (/lib/modules/3.0.35-2666-gbdde708-gbdf2583/mnt/hgfs/win_linux_share/linux/dm
89 failed
90 Mac地址为空，将对应的Mac地址MASTER0_DEVICE="e4:f3:f5:c6:41:b6"
91 (2)编译Ethercat 一定要保证所用编译器与内核与硬件平台一致，否则会出现内核模块不能使用或者其它未知问题
92 (3)内核源码要先编译通过，否则Ethercat模块将不能编译通过，因为Ethercat模块依赖于内核本身驱动模块。
```

ethercat-1.5.2源码  
ethercat是目前使用较多的工业以太网解决方案。主站硬件无关，可以在通用的平台上面移植。ethercat源码，研究工业以太网不可少的研究对象

08-19

ethercat-1.5.2.pdf  
ethercat-1.5.2.pdf里面讲述ethercat总线相关协议，以及讲述如何修改网卡驱动程序来支持ethercat协议。

01-12

优质评论可以帮助作者获得更高权重

评论

leecheni 这个twin cat可以让伺服进入OP模式，但是IGH的ethercat states op 不行，这个怎么解决呢？ 6 月前 回复 \*\*

xws245925587 回复 wllw7176: PDO我没修改，我用的电机原始的PDO映射，没做修改；后面我测试时自己写了一些代码，设置为free run模式，看了报错的调试信息，确实是pdo映射的问题，修改了后就可以了，谢谢大佬 1 月前 回复 \*\*

wllw7176 [博主] 回复 xws245925587: 你不能完全按照我的命令照搬，每个点击的控制pdo入口可能不一致，你要按照你的电机手册来配置 1 月前 回复 \*\*

xws245925587 回复 wllw7176: 使用SOEM主站可以与伺服通信，控制电机转动，igh一直没搞通 1 月前 回复 \*\*

xws245925587 回复 wllw7176: 执行完上面几条指令后，使用ethercat slaves查询显示：0 0:0 PREOP E xxx(设备名)，大佬知道这是什么原因吗？ 1 月前 回复 \*\*

wllw7176 [博主] 回复 xws245925587: 10. DC模式不能进入OP模式的可能原因？DC模式一般用于实时性较高的场合，主站实时性要求非常高，一般高实时性场合Slaves要求时间抖动不超过100us 11. 通过ethercat让电机力矩 11.1 设置自由模式 ethercat download -a 0 -p 0 -l uint8 0x1c32 0x01 0x00 设置力矩模式 ethercat download -a 0 -p 0 -l uint8 0x6060 0x00 0x04 11.2 设置驱动器成OP模式 ethercat states -a 0 -p 0 OP 看下是不是我笔记里面说的这个原因非DC模式要设置freerun才能进入op 1 月前 回复 \*\*

leecheni 回复 xws245925587: 没有解决，我这几天一直在用清能德创的伺服试。还是不行。 1 月前 回复 \*\*

xws245925587 回复： 你解决了吗，我安装后测试也是一直在init和preop两个状态来回切换 1 月前 回复 \*\*

Nature8965 回复： 要从init状态开始切换，最后切换到OP 5 月前 回复 \*\*

EtherCAT从站开发入门 - ethercat\_i7的博客 - CSDN博客  
来自: 卢小喵的学习笔记 EtherCAT主站SOEM源码解析---同步管理器SM配置 ...来自: ethercat\_i7的博客 下载 ethercat-1.5.2源码 - 杨波试试 08-...IGH Eth...

11-20

编译SOEM(Simle Open EtherCAT Master)-windows篇 - 云...  
EtherCAT Master 1.5.2. 主站驱动程序,涉及到从站配置,pdo映射,sdo等 下载 EtherCAT Master Sample Code - - 01-06 基于Beckhoff EL9800平台的主站代...

11-8

EtherCAT1.5.2官方文档翻译 第七章  
EtherCAT1.5.2 翻译 7.2 用户空间库 本地应用程序接口 (参见第3章) 驻留在内核空间中，因此只能从内核访问。 为了使应用程序接口在用户空间程序中...

absinjun的博客 2557

Lan9252-FPGA调试笔记  
1.接口完成以后先去读，0x64寄存器。如果值为0x87654321，则时序okay；否，则修改时序 2.访问Lan9252接口的时候 Byte1 Byte2 Byte3 Byte4 数据线...

albr10546的博客 664

Linux中EtherCAT主站执行过程\_若海人生的专栏  
在上层申请主站,完成各种主从站的配置项之后,通过ecrt\_master\_activate函数调用激活整个EtherCAT总线,ecrt\_master\_activate函数最主要的工作还是完...

9-9

EtherCAT的从站配置 - qq\_22530197的博客 - CSDN博客  
2."EtherCAT的从站配置" 对于EtherCAT最首先要知道的不是他的具体的通信原理,通信方式,协议是什么样子的,首先要知道他的用途。 1)现场总线工业以太...

11-1

ethercat1.5.2官方文档+中文翻译  
IGH ethercat 官方手册，包含特性、安装说明、接口说明、etherca工具说明等等

11-24

ethercat源码 v1.5.2  
ethercat是目前使用较多的工业以太网解决方案。主站硬件无关，可以在通用的平台上面移植。ethercat源码

02-11

EtherCAT之Lan9252调试笔记 - 世界各处去跑马 - CSDN博客  
Xilinx zynq EtherCAT LAN9252 - A的博客 236 参考 开源EtherCAT主站 SOEM和igh的比较 Igh Master 1.5.2 ...STM32F407VGT6+LAN8720 调试笔记 - 翼...

11-23

ethercat主站源码 1.5.2  
适合linux平台下的ethercat主站开发

01-11

elasticsearch-1.5.2  
ElasticSearch是一个基于Lucene的搜索服务器。它提供了一个分布式多用户能力的全文搜索引擎，基于RESTful web接口。Elasticsearch是用Java开发的...

07-06

Ethercat-Ighmaster bt  
1.基于Igh-master1.5.2的多种EtherCAT IO 模块及伺服电机的操作例子 2.基于Ethercat-Ighmaster二次开发库简化你的开发流程 所有源码及文档托管在码云...

02-13

dev调试时无法进入下一步\_Linux开源实时EtherCAT主站操作系统安装和调试笔记 最新发布  
话要从去年找工作的时候说起。秋招收了俩offer，一个自动驾驶领域一个机器人领域，跟我说的工作内容都是“实时操作系统的移植”方面的实现工作。好...

weixin\_39715997的博客 123

4-3-基于IGH EtherCAT实现控制LED开发例程使用手册  
4-3-基于IGH EtherCAT实现控制LED开发例程使用手册,总线开发技术的手册，是一个开发的全过程案例

05-05

Linuxcnc-Ighethercat 调试心得  
Linuxcnc-Ighetherca调试心得 一、前言 如果想通过linuxcnc和Ighethercat来实现对伺服电机的控制。那就要搞明白你需要经过哪几个步骤才能实现伺服控...

混合编程 2943

IgH-EtherCAT Master  
EtherCAT的主站开发是基于EtherCAT机器人控制系统的开发中非常重要的环节。EtherLab的the IgH EtherCAT® Master是目前常见的一种开源的主站。Ig...

10-28

EtherCAT使用手册  
EtherCAT使用手册，分为若干章节：EtherCAT简介，安装与连接，EtherCAT-EC信息，基于分布时钟的网络同步等

07-26

ubuntu下编译Igb ethercat驱动时报错ec\_master not found的解决  
在ubuntu下编译Igb ethercat 时，按照文档install的顺序安装完了之后，在执行 sudo /etc/init.d/ethercat start后，提示错误 Failat:ec\_master not found fail...

1093

文章(学习笔记)  
文章目录1. IgH Master 1.5.2 Documentation1.1 特性1.2 Architecture1.2.1 Master Module1.2.2 Master Phases1.2.3 Process Data1.3 Application Interfa...

pw19997的博客 2420

IgH EtherCAT Master 源码编译安装 热门推荐  
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:1...

scyhk的博客 1万+

Ethercat解析（五）之基础答疑  
(1) 为什么要将EtherCAT主站整合在内核空间？ 答：一方面考虑到内核空间相比于用户空间函数的延迟时间短且内核空间可直接操控硬件；另一方面，由...

absinjun的博客 2628

©2020 CSDN 皮肤主题: 数字20 设计师:CSDN官方博客 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

wllw7176 关注

0 9 18

专栏目录