

```
1
2 //-----
3 //
4 //                      CBEZ-STANDARD-D
5 //                      OCT Co.,Ltd.
6 //-----
7
8 #include <vcl.h>
9 #include <stdio.h>
10 #include <conio.h>
11
12 #pragma hdrstop
13
14 int OpenComPort(char *ComPort);
15 int CBEZTxRx(BYTE CMD, int *SIZE, BYTE *DATA);
16
17 int CheckCBEZ(void);
18 int StandbyCM(void);
19 int StandbyBV(void);
20 void CtrlLoop(void);
21
22
23 char str_COM2[4]="COM2";          // デフォルトのCOMポート番号
24
25 HANDLE hComm;                   // ComPortハンドル
26 char pszBuf[0x100];             /* データバッファ */
27
28 DWORD dwErrors; /* エラー情報 */
29 COMSTAT ComStat; /* デバイスの状態 */
30 DWORD dwCount; /* 受信データのバイト数 */
31 DWORD dwRead; /* ポートから読み出したバイト数 */
32 DWORD dwWritten; /* ポートへ書き込んだバイト数 */
33
34 DWORD lpErrors; /* エラー情報 */
35 COMSTAT lpStat; /* ポートの状態に関する情報 */
36
37
38 //-----
39 #pragma argsused
40 int main(int argc, char* argv[])
41 {
42     int result;
43     char *ComPort;
44
45     // COMポート指定
46     // コマンドライン引数に指定のポートをオープン
47     // 指定なき場合はCOM3
48
49     if ( argc>=2 )
50     {
51         ComPort=argv[1];
52     }
53     else
54         ComPort=str_COM2;
55
56     // COMポートオープン
57     result=OpenComPort(ComPort);
58     if ( result!=0 )
59     {
60         getch();
61         return(result);
62     }
63
64     // CBEZ存在確認
65     result=CheckCBEZ();
66     if ( result!=0 )
67     {
68         printf("CBEZ-Standard2基板が応答しません。(プログラム書込み異常の疑い) %n");
69         goto EXIT;
70     }
71
72     printf("CBEZ-Standard2基板 認識OK. %n");
73
74     // C/Mスタンバイ
75     result=StandbyCM();
76     if ( result!=0 )
```

```

77     {
78         printf("C/Mが応答しません。(基板不良orC/Mとの接続異常の疑い) %n");
79         goto EXIT;
80     }
81
82     // B/Vスタンバイ
83     result=StandbyBV();
84     if( result!=0 )
85     {
86         printf("B/Vが応答しません。(基板不良orC/Mとの接続異常の疑い) %n");
87         goto EXIT;
88     }
89
90     // 販売ループ
91     printf("初期化テストOK. %n");
92     printf("入出金、I/Oテストを行います。(C/Mの返金レバーでテスト終了) %n");
93     CtrlLoop();
94
95     printf("全てのプログラムテスト項目(目視確認項目を除く)にパスしました. %n");
96
97 EXIT:
98
99     // COMポートクローズ
100    CloseHandle(hComm);
101    printf("CBEZ-Standard2との接続を切断し、プログラムを終了します. %n");
102
103    getch();
104    return(0);
105 }
106 //-----
107 int OpenComPort(char * ComPort)
108 {
109     // COMポートオープン
110     hComm = CreateFile(
111         ComPort,          /* COMポート指定 */
112         GENERIC_READ | GENERIC_WRITE, /* アクセスモード */
113         0,                /* 共有モード */
114         NULL,             /* セキュリティ属性 */
115         OPEN_EXISTING,   /* 作成フラグ */
116         FILE_ATTRIBUTE_NORMAL, /* 属性 */
117         NULL              /* テンプレートのハンドル */
118     );
119
120     if(hComm == INVALID_HANDLE_VALUE)
121     {
122         printf("%s ポートがオープンできません. %n", ComPort);
123         printf("CBEZ-Standard2基板がWindowsに認識されていない。(基板不良orUSB接続異常の疑い) %n");
124         return(-1);
125     }
126
127     // COMポート初期化
128     DCB dcb;
129     GetCommState(hComm, &dcb); /* DCB を取得 */
130     // 38400bps, 8bit-char, Non-parity, StopBit1
131     dcb.BaudRate = 38400;
132     dcb.ByteSize = 8;
133     dcb.fParity = FALSE;
134     dcb.StopBits = ONESTOPBIT;
135     // hardware flow-control disable
136     dcb.fOutxCtsFlow = FALSE;
137     dcb.fRtsControl = RTS_CONTROL_DISABLE;
138     dcb.fOutxDsrFlow = FALSE;
139     dcb.fDtrControl = DTR_CONTROL_DISABLE;
140     dcb.fDsrSensitivity = FALSE;
141     // Xon/Xoff flow-control disable
142     dcb.fOutX = FALSE;
143     dcb.fInX = FALSE;
144     dcb.fAbortOnError = FALSE;
145     SetCommState(hComm, &dcb); /* DCB を設定 */
146
147     //COMポートタイムアウト定義
148     COMMTIMEOUTS pcto;
149     GetCommTimeouts(hComm, &pcto);
150     pcto.ReadIntervalTimeout=0;
151     pcto.ReadTotalTimeoutMultiplier=0;
152     pcto.ReadTotalTimeoutConstant=500;          // 応答タイムアウト500ms

```

```

153     pcto.WriteTotalTimeoutMultiplier=0;
154     pcto.WriteTotalTimeoutConstant=0;
155     SetCommTimeouts(hComm, &pcto);
156
157     return(0);
158 }
159 }
160 //-----
161 int CBEZTxRx(BYTE CMD, int *SIZE, BYTE *DATA)
162 {
163     BYTE TxData[0x100];
164     BYTE RxData[0x100];
165     BYTE TxSize=0;
166     BYTE RxSize=0;
167
168     BYTE bc, fcc, ack;
169     int i=0;
170
171     // 送信コマンドフレーム生成
172     TxData[0]=CMD;
173     TxData[1]=~CMD;
174     if( SIZE!=NULL )
175         bc=*SIZE;
176
177     if( SIZE!=NULL && DATA!=NULL && *SIZE!=0 )
178     {
179         // 送信データフレーム生成
180
181         TxData[2]=bc;
182         fcc=bc;
183         for(i=0; i<bc; i++)
184         {
185             TxData[3+i]=DATA[i];
186             fcc^=DATA[i];
187         }
188         TxData[3+i]=fcc;
189         TxSize=i+4;
190     }
191     else
192         TxSize=2;
193
194     // 受信バッファをダミーリード
195     ClearCommError(hComm, &dwErrors, &ComStat);
196     dwCount = ComStat.cbInQue;
197     if( dwCount>0 )
198         ReadFile(hComm, pszBuf, dwCount, &dwRead, NULL);
199
200     // コマンド発行
201     WriteFile(hComm, TxData, TxSize, &dwWritten, NULL);
202
203     // 応答確認
204     ReadFile(hComm, &ack, 1, &dwRead, NULL);
205     bc=0;
206     if( dwRead!=1 )
207         return(-1);    // 無応答
208
209     if( (CMD&0x07)==0x01 )
210     {
211         if( ack!=0x11 && ack!=0x33 )
212             return(ack);    // 応答データフレームなし
213     }
214     else
215     {
216         if( ack!=0x22 && ack!=0x33 )
217             return(ack);    // 応答データフレームなし
218     }
219
220     // 応答データフレーム受信
221     ReadFile(hComm, &RxSize, 1, &dwRead, NULL);
222     if( dwRead!=1 )
223         return(-2);    // 無応答
224     ReadFile(hComm, &RxData, RxSize+1, &dwRead, NULL);
225     if( dwRead!=(RxSize+1) )
226         return(-3);    // 応答異常
227
228     if( SIZE==NULL || DATA==NULL )

```

```
229         return(ack);    // 応答データフレーム無視
230
231
232     // FCCチェック
233     fcc=RxSize;
234     for (i=0;i<RxSize;i++)
235     {
236         DATA[i]=RxData[i];
237         fcc ^=RxData[i];
238     }
239     *SIZE=RxSize;
240     if ( fcc!=RxData[i] )
241         return(-4);    // FCC不一致
242
243     return(ack);
244 }
245 //-----
246 int CheckCBEZ(void)
247 {
248
249     int result;
250     int i;
251     int size=0;
252     BYTE data[0x100];
253
254     // CBEZへ スタンバイコマンド発行
255     result=CBEZTxRx( 0xF0, NULL, NULL );
256
257     if ( result!=0x11 )
258     {
259         printf("CBEZ-Standard2が見つかりません. %n");
260         return(-1);
261     }
262
263     // CBEZへ 一括要求コマンド発行
264     size=0;
265     result=CBEZTxRx( 0xF1, &size, data );
266
267     printf("CBEZ-Standard2と接続しました. %n");
268     printf("%s%n", &data[2]);
269     return(0);
270 }
271 //-----
272 //
273 int StandbyCM(void)
274 {
275
276     int result;
277
278     // C/Mへ スタンバイコマンド発行
279     result=CBEZTxRx( 0x60, NULL, NULL );
280
281     if ( result!=0x11 )
282     {
283         printf("C/Mが準備できていません. #1%n");
284         return(-1);
285     }
286
287     // C/Mへ 一括要求コマンド発行
288     result=CBEZTxRx( 0x61, NULL, NULL );
289
290     if ( result!=0x11 && result!=0x33 )
291     {
292         printf("C/Mが準備できていません. #2%n");
293         return(-1);
294     }
295
296     printf("C/Mの接続を確認しました. %n");
297     return(0);
298
299 }
300 //-----
301 //
302 int StandbyBV(void)
303 {
304
```

```

305     int result;
306
307     // B/Vへ スタンバイコマンド発行
308     result=CBEZTxRx( 0x58, NULL, NULL );
309
310     if( result!=0x11 )
311     {
312         printf("B/Vが準備できていません。%n");
313         return(-1);
314     }
315
316     // B/Vへ 一括要求コマンド発行
317     result=CBEZTxRx( 0x59, NULL, NULL );
318
319     if( result!=0x11 && result!=0x33 )
320     {
321         printf("B/Vが準備できていません。%n");
322         return(-1);
323     }
324
325     printf("B/Vの接続を確認しました。%n");
326     return(0);
327
328 }
329 }
330 //-----
331 void CtrlLoop(void)
332 {
333     int result,i;
334     int size=0;
335     BYTE data[0x100];
336     BYTE btn=0;
337     BYTE prev_btn=0;
338     int coinin=0;
339     int bill=0;
340     int credit=-1;
341     int secCnt=0;
342     int exitflag=0;
343
344
345     // C/M 硬貨受け入れ指示
346     data[0]=0x00;
347     data[1]=0x01;
348     size=2;
349     result=CBEZTxRx( 0x63, &size, data );
350
351     // B/V 紙幣受け入れ指示
352     data[0]=0x10;
353     data[1]=0x21;
354     data[2]=0x00;
355     size=3;
356     result=CBEZTxRx( 0x5B, &size, data );
357
358
359     do
360     {
361
362         // 状況確認 C/M
363         size=0;
364         result=CBEZTxRx( 0x62, &size, data );
365         if( result!=0x11 )
366         {
367             size=0;
368             result=CBEZTxRx( 0x61, &size, data );
369             coinin=(data[2]&0x0F)*10+(data[3]&0x0F)*50+(data[4]&0x0F)*100+(data[5]&0x0F)*500
370                 +(data[2]>>4)*100+(data[3]>>4)*500+(data[4]>>4)*1000+(data[5]>>4)*5000;
371
372             if( (data[20]&0x08) !=0 )
373                 exitflag=1;
374         }
375
376         // 状況確認 B/V
377         size=0;
378         result=CBEZTxRx( 0x5A, &size, data );
379         if( result!=0x11 )
380         {

```

```

381         size=0;
382         result=CBEZTxRx( 0x59, &size, data );
383         bill=data[8];
384     }
385
386     // 金額更新
387     if( credit!=(coinin+bill*1000) )
388     {
389         credit=(coinin+bill*1000);
390         printf("投入金額:¥%d¥n", credit);
391     }
392 }while( exitflag==0 );
393
394
395
396
397
398 // C/M 硬貨受け入れ禁止指示
399 data[0]=0x00;
400 data[1]=0x00;
401 size=2;
402 result=CBEZTxRx( 0x63, &size, data );
403
404 // B/V 紙幣受け入れ禁止指示
405 data[0]=0x10;
406 data[1]=0x00;
407 data[2]=0x00;
408 size=3;
409 result=CBEZTxRx( 0x5B, &size, data );
410
411 // B/V 後続金なしフラグチェック
412 do
413 {
414     size=0;
415     result=CBEZTxRx( 0x59, &size, data );
416 }while( (data[20]&0x04)==0 );
417
418 // C/M 払い出し可能フラグチェック
419 do
420 {
421     size=0;
422     result=CBEZTxRx( 0x61, &size, data );
423     coinin=data[2]*10+data[3]*50+data[4]*100+data[5]*500;
424 }while( (data[20]&0x04)==0 );
425
426
427 if( bill!=0 )
428 {
429     printf("B/V返金処理開始. ¥n");
430
431     // B/V 返金
432     data[0]=0x11;
433     data[1]=0x00;
434     data[2]=0x10;
435     data[3]=0x00;
436     data[4]=0x00;
437     size=5;
438     result=CBEZTxRx( 0x5B, &size, data );
439
440     // B/V 払い出し完了フラグチェック
441     do
442     {
443         // C/M ダミーアクセス
444         size=0;
445         result=CBEZTxRx( 0x62, &size, data );
446
447         // B/V
448         size=0;
449         result=CBEZTxRx( 0x59, &size, data );
450     }while( (data[20]&0x08)==0 );
451 }
452
453 if( coinin!=0 )
454 {
455     printf("C/M返金処理開始. ¥n");
456

```

```
457 // C/M 返金
458 data[0]=0x01;
459 data[1]=((coinin/10)%10)<<4;
460 data[2]=((coinin/1000)%10)<<4 | ((coinin/100)%10);
461 data[3]=((coinin/10000)%10);
462 size=4;
463 result=CBEZTxRx( 0x63, &size, data );
464
465 // C/M 払い出し完了フラグチェック
466 do
467 {
468     // B/Vダミーアクセス
469     size=0;
470     result=CBEZTxRx( 0x5A, &size, data );
471
472     // C/M
473     size=0;
474     result=CBEZTxRx( 0x61, &size, data );
475     coinin=data[2]*10+data[3]*50+data[4]*100+data[5]*500;
476 }while( (data[20]&0x10)==0 );
477 }
478
479 // B/V 決済(投入金クリア)
480 data[0]=0x10;
481 data[1]=0x02;
482 data[2]=0x00;
483 size=3;
484 result=CBEZTxRx( 0x5B, &size, data );
485
486 // B/Vクリア済みチェック
487 do
488 {
489     size=0;
490     result=CBEZTxRx( 0x59, &size, data );
491 }while( (data[20]&0x10)==0 );
492
493
494
495 // C/M 決済(投入金クリア)
496 data[0]=0x00;
497 data[1]=0x02;
498 size=2;
499 result=CBEZTxRx( 0x63, &size, data );
500
501 // C/M クリア済みチェック
502 do
503 {
504     size=0;
505     result=CBEZTxRx( 0x61, &size, data );
506 }while( (data[20]&0x20)==0 );
507
508
509 }
510 //-----
511 //
512 //-----
513
```