

```

1  #include "all03.h"
2
3  #define PROGNAME      "TEST PROGRAM  V1.00"
4
5  int PROMA3 = 1;
6
7  /*=====*/
8  void showvals( void )
9  {
10     struct text_info ti;
11
12     gettextinfo( &ti );
13
14     textattr( ( BLUE << 4 ) + WHITE ); clrscrn( 0,40, 9,79 );
15     _window( 3,40, 9,79 );
16
17     _window( 6,69, 9,79 );
18     locate( 7,71 ); cprintf( "COUNTER" );
19     locate( 8,71 ); cprintf( " 00000 " );
20
21     locate( 4,54 ); cprintf( "PARAMETERS" );
22     locate( 5,50 ); cprintf( "I/O addr.: %04X", idport );
23
24     locate( 0,40 ); cprintf( "*VCC :%4.1fV", VCC / 10.0 );
25     locate( 1,40 ); cprintf( "*VHH :%4.1fV", VHH / 10.0 );
26     locate( 2,40 ); cprintf( "*VOP :%4.1fV", VOP / 10.0 );
27
28     locate( 0,62 ); cprintf( "*GND PIN:%s",
29         ( getport( OTHERENID, 0 ) & 1 ) ? "P11&30" : "P20" );
30
31     textattr( ti.attribute );
32     gotoxy( ti.curx, ti.cury );
33 }
34
35 /*=====*/
36 void power_menu( void )
37 {
38     textattr( ( CYAN << 4 ) + WHITE ); _window( 10,40, 23,79 );
39
40     locate(11,41 ); cprintf( "Measure the following points by DVM, " );
41     locate(12,41 ); cprintf( "values have to be within the specified" );
42     if( PROMA3 ) {
43         locate(13,41 ); cprintf( "limits.          Ground point is P10 of CN3" );
44
45         locate(15,41 ); cprintf( "      CN3          MIN.      TYP.      MAX. " );
46         locate(16,41 ); cprintf( "P1,P20          4.8V      5.0V      5.2V " );
47         locate(17,41 ); cprintf( "P3,P18          10.8V     12.0V     13.2V " );
48         locate(18,41 ); cprintf( "(Coll.TR1)      27.5V     30.5V     33.5V " );
49         locate(19,41 ); cprintf( "(Coll.TR2)      17.3V     19.2V     21.2V " );
50         locate(20,41 ); cprintf( "U26 P3(723C)    7.2V      8.0V      8.8V " );
51     } else {
52         locate(13,41 ); cprintf( "limits.          Ground point is P1 of J2" );
53
54         locate(15,41 ); cprintf( "      J2          MIN.      TYP.      MAX. " );
55         locate(16,41 ); cprintf( "P2,P39          4.8V      5.0V      5.2V " );
56         locate(17,41 ); cprintf( "P3,P38          10.8V     12.0V     13.2V " );
57         locate(18,41 ); cprintf( "(Coll.Q5)       27.5V     30.5V     33.5V " );
58         locate(19,41 ); cprintf( "(Coll.Q1)       17.3V     19.2V     21.2V " );
59         locate(20,41 ); cprintf( "U38 P3(723C)    7.2V      8.0V      8.8V " );
60     }
61
62     locate(22,41 ); cprintf( "Press any key back to main menu" );
63
64     textattr( ( BLUE << 4 ) + WHITE );
65     locate(10,43 ); cprintf( " POWER SUPPLY TEST " );
66
67     getch();
68 }
69
70 /*=====*/
71 void adj_menu( void )
72 {

```

```

73     textattr( ( CYAN << 4 ) + WHITE ); _window( 10,40, 23,79 );
74
75     locate(11,41 ); cprintf( "Measure and adjust the following      " );
76     locate(12,41 ); cprintf( "points on connector J2 by DVM, the      " );
77     locate(13,41 ); cprintf( "values have to be in the specified      " );
78
79     if( PROMA3 ) {
80         locate(14,41 ); cprintf( "limits.          Ground point is P10 of CN4 " );
81
82         locate(16,41 ); cprintf( "  CN4          MIN.    TYP.    MAX.    ADJ. " );
83         locate(17,41 ); cprintf( "P9,P12        7.9V    8.0V    8.1V    VR3  " );
84         locate(18,41 ); cprintf( "P8,P13        15.2V   15.3V   15.4V   VR2  " );
85         locate(19,41 ); cprintf( "P7,P14        25.4V   25.5V   25.6V   VR1  " );
86     } else {
87         locate(14,41 ); cprintf( "limits.          Ground point is P20 of J2 " );
88
89         locate(16,41 ); cprintf( "  J2          MIN.    TYP.    MAX.    ADJ. " );
90         locate(17,41 ); cprintf( "P17,P24       7.9V    8.0V    8.1V    VR3  " );
91         locate(18,41 ); cprintf( "P18,P23       15.2V   15.3V   15.4V   VR2  " );
92         locate(19,41 ); cprintf( "P19,P22       25.4V   25.5V   25.6V   VR1  " );
93     }
94
95     locate(22,41 ); cprintf( "Press any key back to main menu" );
96
97     setdac_raw( VOPID, 255 );          /* VOP = 25.50V nom. */
98     setdac_raw( VHHID, 153 );          /* VHH = 15.30V nom. */
99     setdac_raw( VCCID, 80 );           /* VCC = 8.00V nom. */
100    showvals();
101
102    textattr( ( BLUE << 4 ) + WHITE );
103    locate(10,43 ); cprintf( " VOLTAGE ADJUSTMENT " );
104
105    getch();
106 }
107
108 /*=====*/
109 void gndtest( void )
110 {
111     int ch, step = 0;
112
113     textattr( ( CYAN << 4 ) + WHITE ); _window( 10,40, 23,79 );
114
115     locate(11,41 ); cprintf( "There are 3 GND pins on test-socket      " );
116     locate(12,41 ); cprintf( "that have to be tested by DVM, the      " );
117     locate(13,41 ); cprintf( "values have to be within the specified" );
118     if( PROMA3 ) {
119         locate(14,41 ); cprintf( "limits.          Ground point is P10 of CN4" );
120     } else {
121         locate(14,41 ); cprintf( "limits.          Ground point is P20 of J2" );
122     }
123     locate(16,41 ); cprintf( "  Test-socket    MIN.    TYP.    MAX. " );
124     locate(17,41 ); cprintf( " STEP           0.0V    0.1V    0.2V " );
125     locate(18,41 ); cprintf( " all other pins  4.9V    5.0V    5.1V " );
126
127     locate(20,41 ); cprintf( "Press 'S' key to change testing step      " );
128     locate(21,41 ); cprintf( "Press <CR> back to main menu              " );
129
130     textattr( ( BLUE << 4 ) + WHITE );
131     locate(10,43 ); cprintf( " GND PIN TEST " );
132
133     setdac( VCCID, 50 );
134     showvals();
135
136     textattr( ( CYAN << 4 ) + WHITE );
137
138     while( TRUE )
139     {
140         if( step == 0 )
141         {
142             setother( 0, 0 );
143             locate( 17, 47 );
144             cprintf( "1: P20          " );

```

```

145     }
146     else
147     {
148         setother( 0, 1 );
149         locate( 17, 47 );
150         cprintf( "2: P11,P30" );
151     }
152
153     locate( 21,70 );
154     step ^= 1;
155
156     while( (ch=getch()) == 0 )
157         getch();
158
159     if( ch==CR || ch==LF || ch==ESC )
160         break;
161 }
162 }
163
164 /*=====*/
165 void vcctest( void )
166 {
167     int ch, step=0;
168     static char pins[] = { 1,5,7,9,26,27,28,29,30,31,32,34,36,40, 0 };
169
170     textattr( ( CYAN << 4 ) + WHITE ); _window( 10,40, 23,79 );
171
172     locate(11,41); cprintf( "There are 14 VCC pins on testsocket" );
173     locate(12,41); cprintf( "that must be tested by DVM, the values" );
174     locate(13,41); cprintf( "have to be within the specified limits" );
175     locate(14,41); cprintf( "Ground point is P20 of test-socket" );
176     locate(15,41); cprintf( "(except for P20: Ground is P11,P30)" );
177
178     locate(17,41 ); cprintf( "      Test-socket      MIN.      TYP.      MAX." );
179     locate(18,41 ); cprintf( " STEP                4.9V      5.0V      5.1V" );
180
181     locate(20,41); cprintf( "Press 'S' key to change testing step" );
182     locate(21,41); cprintf( "Press <CR> back to main menu" );
183
184     textattr( ( BLUE << 4 ) + WHITE );
185     locate(10,43); cprintf( " VCC PIN TEST " );
186
187     textattr( ( CYAN << 4 ) + WHITE );
188
189     setdac( VCCID, 50 ); /* vcc 5V output */
190     showvals();
191
192     while( TRUE )
193     {
194         setpin( pins[step], TTLID, 1 );
195         setpin( pins[step], VCCENID, 1 );
196
197         locate( 18, 47 ); cprintf( "%2d: P%d ", step+1, pins[step] );
198         locate( 21, 70 );
199
200         while( (ch=getch()) == 0 )
201             getch();
202
203         initio();
204
205         if( ch==CR || ch==LF || ch==ESC )
206             break;
207
208         if( ( ch=toupper(ch) ) == 'S' )
209             if( pins[++step] == 0 )
210                 step = 0;
211     }
212 }
213
214 /*=====*/
215 void vhhstest( void )
216 {

```

```

217     int ch, step = 0;
218
219     textattr( ( CYAN << 4 ) + WHITE ); _window( 10,40, 23,79 );
220
221     locate(11,41 ); cprintf( "There are 23 VHH pins on test-socket" );
222     locate(12,41 ); cprintf( "that must be tested by DVM, the values" );
223     locate(13,41 ); cprintf( "have to be within the specified limits" );
224     locate(14,41 ); cprintf( "Ground point is P20 of test-socket" );
225     locate(15,41 ); cprintf( "(except for P20: Ground is P11,P30)" );
226
227     locate(17,41 ); cprintf( "      Test-socket      MIN.      TYP.      MAX." );
228     locate(18,41 ); cprintf( " STEP 1:              11.7V  12.0V  12.3V" );
229     locate(19,41 ); cprintf( " all other Pins      4.9V   5.0V   5.1V" );
230
231     locate(21,41 ); cprintf( "Press 'S' key to change testing step" );
232     locate(22,41 ); cprintf( "Press <CR> back to main menu" );
233
234     textattr( ( BLUE << 4 ) + WHITE );
235     locate( 10,43 ); cprintf( " VHH PIN TEST " );
236
237     textattr( ( CYAN << 4 ) + WHITE );
238
239     setdac( VCCID, 50 );                /* VCC 5V output */
240     setdac( VHHID, 120 );               /* VHH 12V output */
241     showvals();
242
243     while( TRUE )
244     {
245         setother( 0, ( step == 20-9 ) ? 1:0 );        /* Pin20 -> Pin30=GND */
246         setpin( step + 9, VHHENID, 1 );
247
248         locate( 18, 47 ); cprintf( "%2d: P%d ", step+1, step+9 );
249         locate( 22, 70 );
250
251         while( (ch=getch()) == 0 )
252             getch();
253
254         initio();
255
256         if( ch==CR || ch==LF || ch==ESC )
257             break;
258
259         if( ( ch = toupper(ch) ) == 'S' )
260             if( ++step == 24 )
261                 step = 0;
262     }
263 }
264
265 /*=====*/
266 void vopptest( void )
267 {
268     int ch, step=0;
269     char pin[29] = {1,5,6,7,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,
270                    26,27,28,29,30,31,32,36};
271
272     textattr( ( CYAN << 4 ) + WHITE ); _window( 10,40, 23,79 );
273
274     locate(11,41 ); cprintf( "There are %d VOP pins on test-socket", PROMA3 ? 29:28 );
275     locate(12,41 ); cprintf( "that must be tested by DVM, the values" );
276     locate(13,41 ); cprintf( "have to be within the specified limits" );
277     locate(14,41 ); cprintf( "Ground point is P20 of test-socket" );
278     locate(15,41 ); cprintf( "(except for P20: Ground is P11,P30)" );
279
280     locate(17,41 ); cprintf( "      Test-socket      MIN.      TYP.      MAX." );
281     locate(18,41 ); cprintf( " STEP 1:              20.5V  21.0V  21.5V" );
282     locate(19,41 ); cprintf( " all other Pins      4.9V   5.0V   5.1V" );
283
284     locate(21,41 ); cprintf( "Press 'S' key to change testing step" );
285     locate(22,41 ); cprintf( "Press <CR> back to main menu" );
286
287     textattr( ( BLUE << 4 ) + WHITE );
288     locate( 10,43 ); cprintf( " VOP PIN TEST " );

```

```

289
290     textattr( ( CYAN << 4 ) + WHITE );
291
292     setdac( VCCID, 50 );          /* VCC 5V out put */
293     setdac( VOPIID, 210 );        /* VOP 21V output */
294     showvals();
295
296     while( TRUE )
297     {
298         setother( 0, ( pin[step] == 20 ) ? 1:0 );    /* Pin20 -> Pin30=GND */
299         setpin( pin[step], VOPENID, 1 );
300
301         locate( 18, 47 ); cprintf( "%2d: P%d ",
302                                     step + ( ( PROMA3 | ( step < 2 ) ) ? 1:0 ),
303                                     pin[step] );
304         locate( 22, 70 );
305
306         while( (ch=getch()) == 0 )
307             getch();
308
309         initio();
310
311         if( ch==CR || ch==LF || ch==ESC )
312             break;
313
314         if( toupper(ch) == 'S' )
315         {
316             if( ++step == 29 )
317                 step = 0;
318             else if( step == 2 && !PROMA3 )
319                 ++step;
320         }
321     }
322 }
323
324 /*=====*/
325 void iotest( void )
326 {
327     int vcc = 25, ch, done = 0, step=1, i;
328
329     for( i = 0; i < 5; ++i )
330         setport( TTLID, i, 0x00 );    /* all pins to TTL low */
331
332     textattr( ( CYAN << 4 ) + WHITE ); clrscrn( 0,0, 24,79 );
333
334     textattr( ( BLUE << 4 ) + WHITE );
335     locate( 0,3 ); cprintf( " I/O PIN TEST ");
336     textattr( ( CYAN << 4 ) + WHITE );
337
338     locate( 2,1 ); cprintf( "Measure all 40 I/O pins on test-socket." );
339     locate( 3,1 ); cprintf( "The values have to be within the specified" );
340     locate( 4,1 ); cprintf( "limits. Ground point is P20 of test-socket." );
341     locate( 5,1 ); cprintf( "(except for P20: Ground is P11,P30)" );
342
343     locate( 8,1 ); cprintf( "Measure test-socket pin for" );
344     locate( 9,1 ); cprintf( "          1 : 4.9V to 5.1V" );
345     locate(10,1 ); cprintf( "          0 : 0.0V to 0.2V" );
346
347     locate(12,1 ); cprintf( "O: software output to test-socket." );
348     locate(13,1 ); cprintf( "I: readback from test-socket pin." );
349     locate(14,1 ); cprintf( "    If there is a difference between" );
350     locate(15,1 ); cprintf( "    OUTPUT and INPUT, the test fails !" );
351
352     locate(17,1 ); cprintf( "Press 'S' key to change testing step" );
353     locate(18,1 ); cprintf( "Press 'V' key to change test voltage" );
354     locate(19,1 ); cprintf( "Press <CR> back to main menu" );
355
356     _window( 2,56, 23,65 );
357
358     for( i = 0; i < 40; ++i )
359     {
360         if( i < 20 )

```

```

361         locate( 3 + i, 57 ), cprintf( "%d", i+1 );
362     else
363         locate( 42 - i, 63 ), cprintf( "%d", i+1 );
364 }
365
366 locate( 1,52 ); cprintf( "I O          O I" );
367
368 locate( 2,60 );
369 putchar( 194 ); putchar( 194 );
370
371 while( !done )
372 {
373     setdac( VCCID, vcc );          /* VCC output */
374
375     locate( 9,15 );
376     cprintf( " %d.%dV to %d.%dV",
377         ( 95 * vcc ) / 1000, ( ( 95 * vcc ) / 100 ) % 10,
378         ( 105 * vcc ) / 1000, ( ( 105 * vcc ) / 100 ) % 10 );
379
380     setother( 0, ( step == 20 ) ? 1:0 );    /* Pin20 -> Pin30=GND */
381     setpin( step, TTLID, 1 );
382
383     for( i = 1; i <= 40; ++i )
384     {
385         if( i < 21 )
386         {
387             locate( 2 + i, 52 );
388             cprintf( "%d %d", getpin(i), i == step );
389         }
390         else
391         {
392             locate( 43 - i, 67 );
393             cprintf( "%d %d", i == step, getpin(i) );
394         }
395     }
396
397     if( step < 21 )
398         locate( 2 + step, 57 );
399     else
400         locate( 43 - step, 64 );
401
402     while( ( ch = toupper( getch() ) ) == 0 )
403         getch();
404
405     setpin( step, TTLID, 0 );
406
407     switch( ch )
408     {
409     case CR:
410     case LF:
411     case ESC: done = 1; break;
412
413     case 'S': if( ++step == 41 ) step = 1; break;
414     case 'V':
415         switch( vcc )
416         {
417         case 25: vcc = 33; break;
418         case 33: vcc = 50; break;
419         case 50: vcc = 65; break;
420         case 65: vcc = 25; break;
421         }
422     }
423 }
424 }
425
426 /*=====*/
427 void xtratest( void )
428 {
429     int ch, step;
430
431     textattr( ( CYAN << 4 ) + WHITE ); clrscr( 0,0, 24,79 );
432

```

```

433     textattr( ( BLUE << 4 ) + WHITE );
434     locate( 0,3 ); cprintf( " EXTRA PIN TEST " );
435     textattr( ( CYAN << 4 ) + WHITE );
436
437     locate( 1,1 ); cprintf( "Check values being within limits.          " );
438     locate( 2,1 ); cprintf( "Ground point is P20 of test-socket          " );
439
440     locate( 4,1 ); cprintf( "          Test-socket  MIN.    TYP.    MAX." );
441     locate( 5,1 ); cprintf( " STEP 1      P32      11.7V  12.0V  12.3V" );
442     locate( 6,1 ); cprintf( " STEP 2      P31      11.7V  12.0V  12.3V" );
443     locate( 7,1 ); cprintf( " STEP 3      P30      11.7V  12.0V  12.3V" );
444     locate( 8,1 ); cprintf( " STEP 4      P29      11.7V  12.0V  12.3V" );
445     locate( 9,1 ); cprintf( " STEP 5      P28      11.7V  12.0V  12.3V" );
446     locate(10,1 ); cprintf( " STEP 6      P1       20.5V  21.0V  21.5V" );
447     locate(11,1 ); cprintf( " STEP 7      P5       20.5V  21.0V  21.5V" );
448     locate(12,1 ); cprintf( " STEP 8      P7       20.5V  21.0V  21.5V" );
449     locate(13,1 ); cprintf( " STEP 9      P7       17.0V  17.5V  18.0V" );
450     locate(14,1 ); cprintf( " STEP 10     P25      17.0V  17.5V  18.0V" );
451     if( PROMA3 ) {
452         locate(15,1 ); cprintf( " STEP 11     P2              2.00 MHz OSC  " );
453         locate(16,1 ); cprintf( " STEP 12     P19             2.00 MHz OSC  " );
454     } else {
455         locate(15,1 ); cprintf( " STEP 11     P2              2.21 MHz OSC  " );
456         locate(16,1 ); cprintf( " STEP 12     P19             4.43 MHz OSC  " );
457     }
458
459     locate(18,1 ); cprintf( "all other pins are VCC (4.9V - 5.1V)" );
460     locate(19,1 ); cprintf( "P3,P18 are HiZ in step 11,12 resp.  " );
461
462     locate(21,1 ); cprintf( "Press 'S' key to change testing step" );
463     locate(22,1 ); cprintf( "Press <CR> back to main menu" );
464
465     step = 0;
466
467     locate( 5,1 );
468
469     setdac( VCCID,  50 );          /* VCC 5V output */
470     setdac( VHHID, 120 );          /* VHH 12V output */
471     setdac( VOPID, 210 );          /* VOP 21V output */
472
473     while( TRUE )
474     {
475         showvals();
476
477         switch( step )
478         {
479             case 0: setpin( 32, VHHENCID, 1 ); break;
480             case 1: setpin( 31, VHHENCID, 1 ); break;
481             case 2: setpin( 30, VHHENCID, 1 ); break;
482             case 3: setpin( 29, VHHENCID, 1 ); break;
483             case 4: setpin( 28, VHHENCID, 1 ); break;
484             case 5: setpin(  1, VOPENID, 1 ); break;
485             case 6: setpin(  5, VOPENID, 1 ); break;
486             case 7: setpin(  7, VOPENID, 1 ); break;
487             case 8: setport( OTHERENID, 0, 0x02 ); break; // VOP-2.4V @ 7
488             case 9: setport( OTHERENID, 0, 0x04 ); break; // VOP-2.4V @ 25
489             case 10: setport( OTHERENID, 0, 0x08 ); break; // 2MHz @ 2/3
490             case 11: setport( OTHERENID, 0, 0x20 ); break; // 4MHz @ 19/18
491         }
492
493         locate( 5+step,1 );
494
495         while( (ch=getch()) == 0 )
496             getch();
497
498         initio();
499
500         if( ch==CR || ch==LF || ch==ESC )
501             break;
502
503         if( ( ch=toupper(ch) ) == 'S' )
504             if( ++step == 12 )

```

```

505         step = 0;
506     }
507 }
508
509 /*=====*/
510 void muxtest( void )
511 {
512     int done = 0, step = 0;
513
514     textattr( ( CYAN << 4 ) + WHITE ); clrscrn( 0,0, 24,79 );
515
516     textattr( ( BLUE << 4 ) + WHITE );
517     locate( 0,3 ); cprintf( " MULTIPLEXER TEST " );
518     textattr( ( CYAN << 4 ) + WHITE );
519
520     locate( 1,1 ); cprintf( " STEP          LATCH   REMARKS" );
521     locate( 2,1 ); cprintf( "-----" );
522     locate( 3,1 ); cprintf( " STEP 1      TTL[0] (Pin 1-8)" );
523     locate( 4,1 ); cprintf( " STEP 2      TTL[1] (Pin 9-16)" );
524     locate( 5,1 ); cprintf( " STEP 3      TTL[2] (Pin 17-24)" );
525     locate( 6,1 ); cprintf( " STEP 4      TTL[3] (Pin 25-32)" );
526     locate( 7,1 ); cprintf( " STEP 5      TTL[4] (Pin 33-40)" );
527     locate( 8,1 ); cprintf( " STEP 6      VOP[0] (Pin 1,5,7)" );
528     locate( 9,1 ); cprintf( " STEP 7      VOP[1] (Pin 9-16)" );
529     locate(10,1 ); cprintf( " STEP 8      VOP[2] (Pin 17-24)" );
530     locate(11,1 ); cprintf( " STEP 9      VOP[3] (Pin 25-32)" );
531     locate(12,1 ); cprintf( " STEP 10     VOP[4]" ); // PROMA-3 only !!
532     locate(13,1 ); cprintf( " STEP 11     VHH[1] (Pin 9-16)" );
533     locate(14,1 ); cprintf( " STEP 12     VHH[2] (Pin 17-24)" );
534     locate(15,1 ); cprintf( " STEP 13     VHH[3] (Pin 25-32)" );
535     locate(16,1 ); cprintf( " STEP 14     VHHC[0] (Pin 28-32)" );
536     locate(17,1 ); cprintf( " STEP 15     VCC[0] (Pin 28-32,34,36,40)" );
537     locate(18,1 ); cprintf( " STEP 16     VCC[1] (Pin 1,5,7,9,26,27)" );
538     locate(19,1 ); cprintf( " STEP 17     OTHER" );
539     locate(20,1 ); cprintf( " STEP 18     USER" );
540
541
542     locate(23,1 ); cprintf( "Press 'S' key to change testing step" );
543     locate(24,1 ); cprintf( "Press <CR> back to main menu" );
544
545     while( !done )
546     {
547         locate( 3+step, 1 );
548
549         switch( step )
550         {
551             case 0: setport( TTLID, 0, 0 ); break;
552             case 1: setport( TTLID, 1, 0 ); break;
553             case 2: setport( TTLID, 2, 0 ); break;
554             case 3: setport( TTLID, 3, 0 ); break;
555             case 4: setport( TTLID, 4, 0 ); break;
556
557             case 5: setport( VOPENID, 0, 0 ); break;
558             case 6: setport( VOPENID, 1, 0 ); break;
559             case 7: setport( VOPENID, 2, 0 ); break;
560             case 8: setport( VOPENID, 3, 0 ); break;
561             case 9: setport( VOPENID, 4, 0 ); break;
562
563             case 10: setport( VHHENID, 1, 0 ); break;
564             case 11: setport( VHHENID, 2, 0 ); break;
565             case 12: setport( VHHENID, 3, 0 ); break;
566
567             case 13: setport( VHHENCID, 0, 0 ); break;
568
569             case 14: setport( VCCENID, 0, 0 ); break;
570             case 15: setport( VCCENID, 1, 0 ); break;
571
572             case 16: setport( OTHERENID, 0, 0 ); break;
573
574             case 17: setport( USERBITS, 0, 0 ); break;
575         }
576     }

```

```

577         if( kbhit() )
578         {
579             switch( toupper( getch() ) )
580             {
581                 case 0: getch(); break;
582
583                 case CR:
584                 case LF:
585                 case ESC: done = 1; break;
586
587                 case 'S':
588                     if( ++step == 21 )
589                         step = 0;
590                     break;
591             }
592         }
593     }
594 }
595
596 /*=====*/
597 void bittest( void )
598 {
599     int done = 0, step = 0;
600
601     textattr( ( CYAN << 4 ) + WHITE ); clrscrn( 0,0, 24,79 );
602
603     textattr( ( BLUE << 4 ) + WHITE );
604     locate( 0,3 ); cprintf( " BIT POSITION TEST " );
605     textattr( ( CYAN << 4 ) + WHITE );
606
607     locate( 1,1 ); cprintf( " STEP          OTHER-BIT" );
608     locate( 2,1 ); cprintf( "-----" );
609     locate( 3,1 ); cprintf( " STEP %d    OTHER.%d (Pin )", step+1, step );
610
611
612     locate(23,1 ); cprintf( "Press 'S' key to change testing step" );
613     locate(24,1 ); cprintf( "Press <CR> back to main menu" );
614
615     while( !done )
616     {
617         setport( OTHERENID, 0, 1<<step );
618
619         if( kbhit() )
620         {
621             switch( toupper( getch() ) )
622             {
623                 case 0: getch(); break;
624
625                 case CR:
626                 case LF:
627                 case ESC: done = 1; break;
628
629                 case 'S':
630                     if( ++step == 8 )
631                         step = 0;
632                     break;
633             }
634         }
635     }
636 }
637
638 /*=====*/
639 int main( void )
640 {
641     int ch = 0, loop = 1;
642
643     /*-----*/
644     /* main program realy start here */
645     /*-----*/
646     while( TRUE )
647     {
648         init_hw();

```

```

649     initdacs();
650
651     textattr( ( LIGHTGRAY << 4 ) | YELLOW ); clrscrn( 0,0, 24,79 );
652
653     locate( 0,0 ); cprintf( "Universal      Programmer" );
654     locate( 1,0 );
655     if( PROMA3 == 1 )    cprintf( "MODEL: PROMA-3 " );
656     else                cprintf( "MODEL: PC Based" );
657     locate( 2,0 ); cprintf( PROGNAME );
658
659     showvals();
660
661     textattr( ( CYAN << 4 ) | WHITE ); clrscrn( 3,0, 23,38 );
662
663     locate( 3,0 ); cprintf( "----- Main Menu -----" );
664     locate( 4,0 ); cprintf( "A. Change I/O base address          " );
665     locate( 5,0 ); cprintf( "T. Change programmer type          " );
666     locate( 6,0 ); cprintf( "                                " );
667     locate( 7,0 ); cprintf( "    Bottom board test program :    " );
668     locate( 8,0 ); cprintf( "                                " );
669     locate( 9,0 ); cprintf( "P. Power supply      V. Voltage adjust " );
670     locate(10,0 ); cprintf( "                                " );
671     locate(11,0 ); cprintf( "    Test-socket test program :    " );
672     locate(12,0 ); cprintf( "                                " );
673     locate(13,0 ); cprintf( "G. GND pin test      C. VCC pin test  " );
674     locate(14,0 ); cprintf( "H. VHH pin test      O. VOP pin test  " );
675     locate(15,0 ); cprintf( "I. I/O pin test      X. Extra pin test " );
676     locate(16,0 ); cprintf( "                                " );
677     locate(17,0 ); cprintf( "D. Demuxer test      B. OTHER Bit Test " );
678     locate(18,0 ); cprintf( "                                " );
679     locate(19,0 ); cprintf( "Q. Quit              " );
680     locate(20,0 ); cprintf( "-----" );
681     locate(21,0 ); cprintf( "WARNING : Don't place any device    " );
682     locate(22,0 ); cprintf( "            into the test-socket !    " );
683
684     for( loop = 1; loop; )
685     {
686         textattr( ( BLUE << 4 ) + WHITE ); clrscrn( 24,0, 24,38 );
687
688         locate( 24,0 ); cprintf( "Select function ?" );
689
690         for(;;)
691         {
692             if( ( ch = getch() ) != 0 )
693                 break;
694             getch(); /* neglect extended code */
695         }
696
697         switch( ch = toupper(ch) )
698         {
699             case 'A': set_io_adr(); loop = 0; break; /* change i/o address
700
701             case 'P': power_menu(); break;
702             case 'V': adj_menu(); break;
703             case 'T': PROMA3 ^= 1; loop = 0; break;
704
705             case 'I': iotest(); break;
706             case 'X': xtratest(); break;
707
708             case 'D': muxtest(); break;
709             case 'B': bittest(); break;
710
711             case 'G': gndtest(); break;
712             case 'C': vcctest(); break;
713             case 'H': vhhctest(); break;
714             case 'O': voptest(); break;
715         }
716
717         initio();
718         initdacs();
719
720         if( ch == 'I' || ch == 'X' )

```

```
721         break;
722
723     if( ch == 'Q' )
724         return( 0 );
725
726     textattr( ( LIGHTGRAY << 4 ) | YELLOW ); clrscrn( 10,40, 24,79 );
727     }
728 }
729 }
730
```