

```

1  #include "all03.h"
2
3  typedef struct {
4      char offs;
5      byte mask;
6  } PINDEF;
7
8  static PINDEF TTLPINS[40] = {          /* all pins allowed */
9      { 0, 0x01 }, /* Pin 1 */
10     { 0, 0x02 }, /* Pin 2 */
11     { 0, 0x04 }, /* Pin 3 */
12     { 0, 0x08 }, /* Pin 4 */
13     { 0, 0x10 }, /* Pin 5 */
14     { 0, 0x20 }, /* Pin 6 */
15     { 0, 0x40 }, /* Pin 7 */
16     { 0, 0x80 }, /* Pin 8 */
17
18     { 1, 0x01 }, /* Pin 9 */
19     { 1, 0x02 }, /* Pin 10 */
20     { 1, 0x04 }, /* Pin 11 */
21     { 1, 0x08 }, /* Pin 12 */
22     { 1, 0x10 }, /* Pin 13 */
23     { 1, 0x20 }, /* Pin 14 */
24     { 1, 0x40 }, /* Pin 15 */
25     { 1, 0x80 }, /* Pin 16 */
26
27     { 2, 0x01 }, /* Pin 17 */
28     { 2, 0x02 }, /* Pin 18 */
29     { 2, 0x04 }, /* Pin 19 */
30     { 2, 0x08 }, /* Pin 20 */
31     { 2, 0x10 }, /* Pin 21 */
32     { 2, 0x20 }, /* Pin 22 */
33     { 2, 0x40 }, /* Pin 23 */
34     { 2, 0x80 }, /* Pin 24 */
35
36     { 3, 0x01 }, /* Pin 25 */
37     { 3, 0x02 }, /* Pin 26 */
38     { 3, 0x04 }, /* Pin 27 */
39     { 3, 0x08 }, /* Pin 28 */
40     { 3, 0x10 }, /* Pin 29 */
41     { 3, 0x20 }, /* Pin 30 */
42     { 3, 0x40 }, /* Pin 31 */
43     { 3, 0x80 }, /* Pin 32 */
44
45     { 4, 0x01 }, /* Pin 33 */
46     { 4, 0x02 }, /* Pin 34 */
47     { 4, 0x04 }, /* Pin 35 */
48     { 4, 0x08 }, /* Pin 36 */
49     { 4, 0x10 }, /* Pin 37 */
50     { 4, 0x20 }, /* Pin 38 */
51     { 4, 0x40 }, /* Pin 39 */
52     { 4, 0x80 }, /* Pin 40 */
53 };
54
55 static PINDEF VOPPINS[40] = {          /* 1, 5, 7, 9-32, 36 */
56     { 0, 0x01 }, /* Pin 1 */
57     { -1, 0x02 }, /* Pin 2 */
58     { -1, 0x04 }, /* Pin 3 */
59     { -1, 0x08 }, /* Pin 4 */
60     { 0, 0x10 }, /* Pin 5 */
61     { -1, 0x20 }, /* Pin 6 */
62     { 0, 0x40 }, /* Pin 7 */
63     { -1, 0x80 }, /* Pin 8 */
64
65     { 1, 0x01 }, /* Pin 9 */
66     { 1, 0x02 }, /* Pin 10 */
67     { 1, 0x04 }, /* Pin 11 */
68     { 1, 0x08 }, /* Pin 12 */
69     { 1, 0x10 }, /* Pin 13 */
70     { 1, 0x20 }, /* Pin 14 */
71     { 1, 0x40 }, /* Pin 15 */
72     { 1, 0x80 }, /* Pin 16 */

```

```

73
74     { 2, 0x01 },          /* Pin 17 */
75     { 2, 0x02 },          /* Pin 18 */
76     { 2, 0x04 },          /* Pin 19 */
77     { 2, 0x08 },          /* Pin 20 */
78     { 2, 0x10 },          /* Pin 21 */
79     { 2, 0x20 },          /* Pin 22 */
80     { 2, 0x40 },          /* Pin 23 */
81     { 2, 0x80 },          /* Pin 24 */
82
83     { 3, 0x01 },          /* Pin 25 */
84     { 3, 0x02 },          /* Pin 26 */
85     { 3, 0x04 },          /* Pin 27 */
86     { 3, 0x08 },          /* Pin 28 */
87     { 3, 0x10 },          /* Pin 29 */
88     { 3, 0x20 },          /* Pin 30 */
89     { 3, 0x40 },          /* Pin 31 */
90     { 3, 0x80 },          /* Pin 32 */
91
92     {-1, 0x01 },          /* Pin 33 */
93     {-1, 0x02 },          /* Pin 34 */
94     {-1, 0x04 },          /* Pin 35 */
95     { 4, 0x08 },          /* Pin 36 PROMA-3 only ? */
96     {-1, 0x10 },          /* Pin 37 */
97     {-1, 0x20 },          /* Pin 38 */
98     {-1, 0x40 },          /* Pin 39 */
99     {-1, 0x80 }          /* Pin 40 */
100 };
101
102 static PINDEF VHHPINS[40] = {          /* 9-32 */
103     {-1, 0x01 },          /* Pin 1 */
104     {-1, 0x02 },          /* Pin 2 */
105     {-1, 0x04 },          /* Pin 3 */
106     {-1, 0x08 },          /* Pin 4 */
107     {-1, 0x10 },          /* Pin 5 */
108     {-1, 0x20 },          /* Pin 6 */
109     {-1, 0x40 },          /* Pin 7 */
110     {-1, 0x80 },          /* Pin 8 */
111
112     { 1, 0x01 },          /* Pin 9 */
113     { 1, 0x02 },          /* Pin 10 */
114     { 1, 0x04 },          /* Pin 11 */
115     { 1, 0x08 },          /* Pin 12 */
116     { 1, 0x10 },          /* Pin 13 */
117     { 1, 0x20 },          /* Pin 14 */
118     { 1, 0x40 },          /* Pin 15 */
119     { 1, 0x80 },          /* Pin 16 */
120
121     { 2, 0x01 },          /* Pin 17 */
122     { 2, 0x02 },          /* Pin 18 */
123     { 2, 0x04 },          /* Pin 19 */
124     { 2, 0x08 },          /* Pin 20 */
125     { 2, 0x10 },          /* Pin 21 */
126     { 2, 0x20 },          /* Pin 22 */
127     { 2, 0x40 },          /* Pin 23 */
128     { 2, 0x80 },          /* Pin 24 */
129
130     { 3, 0x01 },          /* Pin 25 */
131     { 3, 0x02 },          /* Pin 26 */
132     { 3, 0x04 },          /* Pin 27 */
133     { 3, 0x08 },          /* Pin 28 */
134     { 3, 0x10 },          /* Pin 29 */
135     { 3, 0x20 },          /* Pin 30 */
136     { 3, 0x40 },          /* Pin 31 */
137     { 3, 0x80 },          /* Pin 32 */
138
139     {-1, 0x01 },          /* Pin 33 */
140     {-1, 0x02 },          /* Pin 34 */
141     {-1, 0x04 },          /* Pin 35 */
142     {-1, 0x08 },          /* Pin 36 */
143     {-1, 0x10 },          /* Pin 37 */
144     {-1, 0x20 },          /* Pin 38 */

```

```

145     {-1, 0x40 },          /* Pin 39 */
146     {-1, 0x80 }          /* Pin 40 */
147 };
148
149 static PINDEF VHHCPINS[40] = {          /* 28-32 */
150     {-1, 0x00 },          /* Pin 1 */
151     {-1, 0x00 },          /* Pin 2 */
152     {-1, 0x00 },          /* Pin 3 */
153     {-1, 0x00 },          /* Pin 4 */
154     {-1, 0x00 },          /* Pin 5 */
155     {-1, 0x00 },          /* Pin 6 */
156     {-1, 0x00 },          /* Pin 7 */
157     {-1, 0x00 },          /* Pin 8 */
158
159     {-1, 0x00 },          /* Pin 9 */
160     {-1, 0x00 },          /* Pin 10 */
161     {-1, 0x00 },          /* Pin 11 */
162     {-1, 0x00 },          /* Pin 12 */
163     {-1, 0x00 },          /* Pin 13 */
164     {-1, 0x00 },          /* Pin 14 */
165     {-1, 0x00 },          /* Pin 15 */
166     {-1, 0x00 },          /* Pin 16 */
167
168     {-1, 0x00 },          /* Pin 17 */
169     {-1, 0x00 },          /* Pin 18 */
170     {-1, 0x00 },          /* Pin 19 */
171     {-1, 0x00 },          /* Pin 20 */
172     {-1, 0x00 },          /* Pin 21 */
173     {-1, 0x00 },          /* Pin 22 */
174     {-1, 0x00 },          /* Pin 23 */
175     {-1, 0x00 },          /* Pin 24 */
176
177     {-1, 0x00 },          /* Pin 25 */
178     {-1, 0x00 },          /* Pin 26 */
179     {-1, 0x00 },          /* Pin 27 */
180     { 0, 0x80 },          /* Pin 28 */
181     { 0, 0x40 },          /* Pin 29 */
182     { 0, 0x20 },          /* Pin 30 */
183     { 0, 0x10 },          /* Pin 31 */
184     { 0, 0x08 },          /* Pin 32 */
185
186     {-1, 0x00 },          /* Pin 33 */
187     {-1, 0x00 },          /* Pin 34 */
188     {-1, 0x00 },          /* Pin 35 */
189     {-1, 0x00 },          /* Pin 36 */
190     {-1, 0x00 },          /* Pin 37 */
191     {-1, 0x00 },          /* Pin 38 */
192     {-1, 0x00 },          /* Pin 39 */
193     {-1, 0x00 }          /* Pin 40 */
194 };
195
196 static PINDEF VCCPINS[40] = {          /* 1, 5, 7, 9, 26-32, 34, 36, 40 */
197     { 1, 0x20 },          /* Pin 1 */
198     {-1, 0x00 },          /* Pin 2 */
199     {-1, 0x00 },          /* Pin 3 */
200     {-1, 0x00 },          /* Pin 4 */
201     { 1, 0x10 },          /* Pin 5 */
202     {-1, 0x00 },          /* Pin 6 */
203     { 1, 0x08 },          /* Pin 7 */
204     {-1, 0x80 },          /* Pin 8 */
205
206     { 1, 0x04 },          /* Pin 9 */
207     {-1, 0x00 },          /* Pin 10 */
208     {-1, 0x00 },          /* Pin 11 */
209     {-1, 0x00 },          /* Pin 12 */
210     {-1, 0x00 },          /* Pin 13 */
211     {-1, 0x00 },          /* Pin 14 */
212     {-1, 0x00 },          /* Pin 15 */
213     {-1, 0x00 },          /* Pin 16 */
214
215     {-1, 0x00 },          /* Pin 17 */
216     {-1, 0x00 },          /* Pin 18 */

```

```

217     {-1, 0x00 },          /* Pin 19 */
218     {-1, 0x00 },          /* Pin 20 */
219     {-1, 0x00 },          /* Pin 21 */
220     {-1, 0x00 },          /* Pin 22 */
221     {-1, 0x00 },          /* Pin 23 */
222     {-1, 0x00 },          /* Pin 24 */
223
224     {-1, 0x00 },          /* Pin 25 */
225     { 1, 0x02 },          /* Pin 26 */
226     { 1, 0x01 },          /* Pin 27 */
227     { 0, 0x80 },          /* Pin 28 */
228     { 0, 0x40 },          /* Pin 29 */
229     { 0, 0x20 },          /* Pin 30 */
230     { 0, 0x10 },          /* Pin 31 */
231     { 0, 0x08 },          /* Pin 32 */
232
233     {-1, 0x00 },          /* Pin 33 */
234     { 0, 0x04 },          /* Pin 34 */
235     {-1, 0x00 },          /* Pin 35 */
236     { 0, 0x02 },          /* Pin 36 */
237     {-1, 0x00 },          /* Pin 37 */
238     {-1, 0x00 },          /* Pin 38 */
239     {-1, 0x00 },          /* Pin 39 */
240     { 0, 0x01 }           /* Pin 40 */
241 };
242
243 int VCC,VHH,VOP;
244
245 /* remember last states */
246
247 static byte ttlen[5], vopen[5], vhh[5];
248 static byte vccen[2], vhhenc[2];
249 static byte otheren;
250
251 /*-----*/
252 void setport( int id, int off, int data )
253 {
254     outp( idport, id+off );
255
256     switch( id )
257     {
258     case TTLID:    ttlen[ off ] = data; break;
259     case VOPENID:  vopen[ off ] = data; break;
260     case VHHENID:  vhh[ off ] = data; break;
261     case VHHENCID: vhhenc[off ] = data; break;
262     case VCCENID:  vccen[ off ] = data; break;
263     case OTHERENID: otheren      = data; break;
264     }
265
266     outp( idport+2, data );
267 }
268
269 /*-----*/
270 int getport( int id, int off )
271 {
272     switch( id )
273     {
274     case VOPENID:    return vopen[ off ];
275     case VHHENID:    return vhh[ off ];
276     case VHHENCID:   return vhhenc[off ];
277     case VCCENID:    return vccen[ off ];
278     case OTHERENID:  return otheren;
279     default:         outp( idport, id+off ); return inp( idport+2 );
280     }
281 }
282
283 int getpin( int pin )
284 {
285     if( --pin < 0 || pin > 39 )
286         return 0;
287     return ( getport( TTLID, pin / 8 ) >> ( pin % 8 ) ) & 1;
288 }

```

```

289
290 void setdac( int which, int data )
291 {
292     outp( idport, which );
293
294     switch( which )
295     {
296     case VOPID: outp( idport+2,      (VOP = data)      + VOPDROP ); break;
297     case VHHID: outp( idport+2, 10 * (VHH = data) / 6 + VHHDROP ); break;
298     case VCCID: outp( idport+2, 10 * (VCC = data) / 4 + VCCDROP ); break;
299     }
300 }
301
302 void setdac_raw( int which, int data ) // for direct measurement
303 {
304     outp( idport, which );
305
306     switch( which )
307     {
308     case VOPID: outp( idport+2,      (VOP = data)      ); break;
309     case VHHID: outp( idport+2, 10 * (VHH = data) / 6 ); break;
310     case VCCID: outp( idport+2, 10 * (VCC = data) / 4 ); break;
311     }
312 }
313
314 void setother( int bit, int state )
315 {
316     if( state )
317         setport( OTHERENID, 0, otheren | (1<<bit) );
318     else
319         setport( OTHERENID, 0, otheren & ~(1<<bit) );
320 }
321
322 int setpin( int pin, int voltage, int state )
323 {
324     int off;
325
326     if( --pin < 0 || pin > 39 )
327         return 0;
328
329     /* if we try to set a pin to a current source, first make sure to set TTL to H,
330      * and that all other sources for this pin are switched off before ! */
331     if( voltage != TTLID && state != 0 )
332     {
333         off = TTLPINS[pin].offs;
334         setport( TTLID, off, ttlen[off] | TTLPINS[pin].mask );
335
336         if( voltage != VOPENID && ( off = VOPPINS[pin].offs ) != -1 )
337             setport( VOPENID, off, vopen[off] & ~VOPPINS[pin].mask );
338         if( voltage != VHHENID && ( off = VHPINS[pin].offs ) != -1 )
339             setport( VHHENID, off, vhh[off] & ~VHPINS[pin].mask );
340         if( voltage != VHHENCID && ( off = VHHCPINS[pin].offs ) != -1 )
341             setport( VHHENCID, off, vhhenc[off] & ~VHHCPINS[pin].mask );
342         if( voltage != VCCENID && ( off = VCCPINS[pin].offs ) != -1 )
343             setport( VCCENID, off, vccen[off] & ~VCCPINS[pin].mask );
344     }
345
346     /* if we set TTL to L, make sure all current sources are switched off before */
347     if( voltage == TTLID && state == 0 )
348     {
349         if( ( off = VOPPINS[pin].offs ) != -1 )
350             setport( VOPENID, off, vopen[off] & ~VOPPINS[pin].mask );
351         if( ( off = VHPINS[pin].offs ) != -1 )
352             setport( VHHENID, off, vhh[off] & ~VHPINS[pin].mask );
353         if( ( off = VHHCPINS[pin].offs ) != -1 )
354             setport( VHHENCID, off, vhhenc[off] & ~VHHCPINS[pin].mask );
355         if( ( off = VCCPINS[pin].offs ) != -1 )
356             setport( VCCENID, off, vccen[off] & ~VCCPINS[pin].mask );
357     }
358
359     switch( voltage )
360     {

```

```

361 case TTLID:
362     off = TTLPINS[pin].offs;    /* no need to check, all pins allowed... */
363
364     if( state )
365         setport( TTLID, off, ttlen[off] | TTLPINS[pin].mask );
366     else
367         setport( TTLID, off, ttlen[off] & ~TTLPINS[pin].mask );
368     break;
369
370 case VOPENID:
371     if( ( off = VOPPINS[pin].offs ) == -1 )
372         return 0;
373
374     if( state )
375         setport( VOPENID, off, vopen[off] | VOPPINS[pin].mask );
376     else
377         setport( VOPENID, off, vopen[off] & ~VOPPINS[pin].mask );
378     break;
379
380 case VHHENID:
381     if( ( off = VHHPINS[pin].offs ) == -1 )
382         return 0;
383
384     if( state )
385         setport( VHHENID, off, vhhenc[off] | VHHPINS[pin].mask );
386     else
387         setport( VHHENID, off, vhhenc[off] & ~VHHPINS[pin].mask );
388     break;
389
390 case VHHENCID:
391     if( ( off = VHHCPINS[pin].offs ) == -1 )
392         return 0;
393
394     if( state )
395         setport( VHHENCID, off, vhhenc[off] | VHHCPINS[pin].mask );
396     else
397         setport( VHHENCID, off, vhhenc[off] & ~VHHCPINS[pin].mask );
398     break;
399
400 case VCCENID:
401     if( ( off = VCCPINS[pin].offs ) == -1 )
402         return 0;
403
404     if( state )
405         setport( VCCENID, off, vccen[off] | VCCPINS[pin].mask );
406     else
407         setport( VCCENID, off, vccen[off] & ~VCCPINS[pin].mask );
408     break;
409 }
410 return 1;
411 }
412

```