```
 1: /*************************************************************************
 2: * ATxmega library for port and pin manipulation - Version 1.0            *
 3: * © Copyright 2012-2019 Filippo Pardini - filippo@robotica.eng.br        *
 4: *                                                                        *
 5: * This program is free software: you can redistribute it and/or modify it   *
 6: * under the terms of the GNU Lesser General Public License as published by  *
 7: * the Free Software Foundation, either version 3 of the License, or any     *
 8: * later version.                                                         *
 9: *                                                                        *
10: * This program is distributed in the hope that it will be useful,           *
11: * but WITHOUT ANY WARRANTY; without even the implied warranty of            *
12: * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the              *
13: * GNU Lesser General Public License for more details.                       *
14: *                                                                        *
15: * You should have received a copy of the GNU Lesser General Public License  *
16: * along with this program. If not, see <http://www.gnu.org/licenses/>.      *
17: *************************************************************************/
18:
19: #include <io.h>
20: #include <iobits.h>
21:
22: void prta_DIR(unsigned char prt, unsigned char io)
23: {
24:     //Defines the pins direction of the prta port
25:     //io => each bit defines the pin direction (0 => input; 1 => output)
26:
27:     switch(prt)
28:     {
29:         case 'A':
30:             PORTA.DIR = io;
31:             break;
32:          case 'B':
33:             PORTB.DIR = io;
34:             break;
35:          case 'C':
36:             PORTC.DIR = io;
37:             break;
38:          case 'D':
39:             PORTD.DIR = io;
40:             break;
41:          case 'E':
42:             PORTE.DIR = io;
```

```
43:             break;
44:          case 'F':
45:             PORTF.DIR = io;
46:             break;
47:          case 'H':
48:             PORTH.DIR = io;
49:             break;
50:          case 'J':
51:             PORTJ.DIR = io;
52:             break;
53:          case 'K':
54:             PORTK.DIR = io;
55:             break;
56:          case 'Q':
57:             PORTQ.DIR = io;
58:             break;
59:          case 'R':
60:             PORTR.DIR = io;
61:             break;
62:     }
63: }
64:
65: //___________________________________________________________________________________________________________________________
66:
67: void prta_OUT(unsigned char prt, unsigned char byt)
68: {
69:     //Places the byt byte on the prta port
70:
71:     switch(prt)
72:     {
73:         case 'A':
74:             PORTA.OUT = byt;
75:             break;
76:          case 'B':
77:             PORTB.OUT = byt;
78:             break;
79:          case 'C':
80:             PORTC.OUT = byt;
81:             break;
82:          case 'D':
83:             PORTD.OUT = byt;
84:             break;
```

```
 85:           case 'E':
 86:              PORTE.OUT = byt;
 87:              break;
 88:           case 'F':
 89:              PORTF.OUT = byt;
 90:              break;
 91:           case 'H':
 92:              PORTH.OUT = byt;
 93:              break;
 94:           case 'J':
 95:              PORTJ.OUT = byt;
 96:              break;
 97:           case 'K':
 98:              PORTK.OUT = byt;
 99:              break;
100:           case 'Q':
101:              PORTQ.OUT = byt;
102:              break;
103:           case 'R':
104:              PORTR.OUT = byt;
105:              break;
106:     }
107: }
108:
109: //_____________________________________________________________________________________________________________________________
110:
111: unsigned char prta_IN(unsigned char prt)
112: {
113:     //Receives a byte from port prta
114:
115:     switch(prt)
116:     {
117:         case 'A':
118:              return PORTA.IN;
119:          case 'B':
120:              return PORTB.IN;
121:          case 'C':
122:              return PORTC.IN;
123:          case 'D':
124:              return PORTD.IN;
125:          case 'E':
126:              return PORTE.IN;
```

```
127:          case 'F':
128:             return PORTF.IN;
129:          case 'H':
130:             return PORTH.IN;
131:          case 'J':
132:             return PORTJ.IN;
133:          case 'K':
134:             return PORTK.IN;
135:          case 'Q':
136:             return PORTQ.IN;
137:          case 'R':
138:             return PORTR.IN;
139:      }
140: }
141:
142: //_____________________________________________________________________________________________________________________
143:
144: void pin_INPUT(char *door)
145: {
146:     //Defines the port pin *door as input
147:
148:     unsigned char n,pta;
149:
150:     pta = door[0];              //Port
151:     n = 1 << (door[1] - 48);    //Pin
152:
153:     n ^= 0xFF;
154:     switch(pta)
155:     {
156:      case 'A':
157:         PORTA.DIR &= n;
158:         break;
159:      case 'B':
160:         PORTB.DIR &= n;
161:         break;
162:      case 'C':
163:         PORTC.DIR &= n;
164:         break;
165:      case 'D':
166:         PORTD.DIR &= n;
167:         break;
168:      case 'E':
```

```
169:         PORTE.DIR &= n;
170:         break;
171:      case 'F':
172:         PORTF.DIR &= n;
173:         break;
174:      case 'H':
175:         PORTH.DIR &= n;
176:         break;
177:      case 'J':
178:         PORTJ.DIR &= n;
179:         break;
180:      case 'K':
181:         PORTK.DIR &= n;
182:         break;
183:      case 'Q':
184:         PORTQ.DIR &= n;
185:         break;
186:      case 'R':
187:         PORTR.DIR &= n;
188:         break;
189:     }
190: }
191:
192: //_____________________________________________________________________________________________________________________
193:
194: void pin_OUTPUT(char *door)
195: {
196:     //Defines the port pin *door output
197:
198:     unsigned char n,pta;
199:
200:     pta = door[0];              //Port
201:     n = 1 << (door[1] - 48);    //Pin
202:
203:     switch(pta)
204:     {
205:      case 'A':
206:         PORTA.DIR |= n;
207:         break;
208:      case 'B':
209:         PORTB.DIR |= n;
210:         break;
```

```
211:      case 'C':
212:         PORTC.DIR |= n;
213:         break;
214:      case 'D':
215:         PORTD.DIR |= n;
216:         break;
217:      case 'E':
218:         PORTE.DIR |= n;
219:         break;
220:      case 'F':
221:         PORTF.DIR |= n;
222:         break;
223:      case 'H':
224:         PORTH.DIR |= n;
225:         break;
226:      case 'J':
227:         PORTJ.DIR |= n;
228:         break;
229:      case 'K':
230:         PORTK.DIR |= n;
231:         break;
232:      case 'Q':
233:         PORTQ.DIR |= n;
234:         break;
235:      case 'R':
236:         PORTR.DIR |= n;
237:         break;
238:     }
239: }
240:
241: //_____________________________________________________________________________________________________________________
242:
243: void putpin(char *door,unsigned char v)
244: {
245:     //Sets the port pin *door as high (v=1) or low (v=0)
246:
247:     unsigned char n,pta;
248:
249:     pta = door[0];      //Port
250:     n = door[1] - 48;   //Pin
251:     if (v == 0)
252:     {
```

```
253:            switch(pta)
254:            {
255:              case 'A':
256:                 CLRBIT(PORTA.OUT,n);
257:                 break;
258:              case 'B':
259:                 CLRBIT(PORTB.OUT,n);
260:                 break;
261:              case 'C':
262:                 CLRBIT(PORTC.OUT,n);
263:                 break;
264:              case 'D':
265:                 CLRBIT(PORTD.OUT,n);
266:                 break;
267:              case 'E':
268:                 CLRBIT(PORTE.OUT,n);
269:                 break;
270:              case 'F':
271:                 CLRBIT(PORTF.OUT,n);
272:                 break;
273:              case 'H':
274:                 CLRBIT(PORTH.OUT,n);
275:                 break;
276:              case 'J':
277:                 CLRBIT(PORTJ.OUT,n);
278:                 break;
279:              case 'K':
280:                 CLRBIT(PORTK.OUT,n);
281:                 break;
282:              case 'Q':
283:                 CLRBIT(PORTQ.OUT,n);
284:                 break;
285:              case 'R':
286:                 CLRBIT(PORTR.OUT,n);
287:                 break;
288:            }
289:     }
290:     else
291:     {
292:         switch(pta)
293:         {
294:             case 'A':
```

```
295:                 SETBIT(PORTA.OUT,n);
296:                 break;
297:             case 'B':
298:                 SETBIT(PORTB.OUT,n);
299:                 break;
300:             case 'C':
301:                 SETBIT(PORTC.OUT,n);
302:                 break;
303:             case 'D':
304:                 SETBIT(PORTD.OUT,n);
305:                 break;
306:             case 'E':
307:                 SETBIT(PORTE.OUT,n);
308:                 break;
309:             case 'F':
310:                 SETBIT(PORTF.OUT,n);
311:                 break;
312:             case 'H':
313:                 SETBIT(PORTH.OUT,n);
314:                 break;
315:             case 'J':
316:                 SETBIT(PORTJ.OUT,n);
317:                 break;
318:             case 'K':
319:                 SETBIT(PORTK.OUT,n);
320:                 break;
321:             case 'Q':
322:                 SETBIT(PORTQ.OUT,n);
323:                 break;
324:             case 'R':
325:                 SETBIT(PORTR.OUT,n);
326:                 break;
327:         }
328:     }
329: }
330:
331: //_____________________________________________________________________________________________________________________________
332:
333: unsigned char getpin(char *door)
334: {
335:     //Reads the port pin (high = 1, low = 0)
336:
```

```
337:     unsigned char x,n,pta;
338:
339:     pta = door[0];      //Port
340:     n = door[1] - 48;   //Pin
341:     switch(pta)
342:     {
343:         case 'A':
344:             x = TSTBIT(PORTA.IN,n);
345:             break;
346:         case 'B':
347:             x = TSTBIT(PORTB.IN,n);
348:             break;
349:         case 'C':
350:             x = TSTBIT(PORTC.IN,n);
351:             break;
352:         case 'D':
353:             x = TSTBIT(PORTD.IN,n);
354:             break;
355:         case 'E':
356:             x = TSTBIT(PORTE.IN,n);
357:             break;
358:         case 'F':
359:             x = TSTBIT(PORTF.IN,n);
360:             break;
361:         case 'H':
362:             x = TSTBIT(PORTH.IN,n);
363:             break;
364:         case 'J':
365:             x = TSTBIT(PORTJ.IN,n);
366:             break;
367:         case 'K':
368:             x = TSTBIT(PORTK.IN,n);
369:             break;
370:         case 'Q':
371:             x = TSTBIT(PORTQ.IN,n);
372:             break;
373:         case 'R':
374:             x = TSTBIT(PORTR.IN,n);
375:             break;
376:     }
377:     return(x);
378: }
```

```
379:
380: //_____________________________________________________________________________________________________________________________
381:
382: void togglepin(char *door)
383: {
384:     //Toggles the port pin
385:
386:     unsigned char n,pta;
387:
388:     pta = door[0];      //Port
389:     n = door[1] - 48;   //Pin
390:     switch(pta)
391:     {
392:         case 'A':
393:             TGLBIT(PORTA.OUT,n);
394:             break;
395:         case 'B':
396:             TGLBIT(PORTB.OUT,n);
397:             break;
398:         case 'C':
399:             TGLBIT(PORTC.OUT,n);
400:             break;
401:         case 'D':
402:             TGLBIT(PORTD.OUT,n);
403:             break;
404:         case 'E':
405:             TGLBIT(PORTE.OUT,n);
406:             break;
407:         case 'F':
408:             TGLBIT(PORTF.OUT,n);
409:             break;
410:         case 'H':
411:             TGLBIT(PORTH.OUT,n);
412:             break;
413:         case 'J':
414:             TGLBIT(PORTJ.OUT,n);
415:             break;
416:         case 'K':
417:             TGLBIT(PORTK.OUT,n);
418:             break;
419:         case 'Q':
420:             TGLBIT(PORTQ.OUT,n);
```

```
421:             break;
422:         case 'R':
423:             TGLBIT(PORTR.OUT,n);
424:             break;
425:     }
426: }
427:
428: //*****************************************************************************************************************************
```