

```

1  ;***** AVR I2C LIBRARY *****
2  ;*
3  ;* Title       : I2C Single Master Implementatie
4  ;* Version     : 1.0
5  ;* Last updated  : 2006.11.02
6  ;* Target      : alle AVR
7  ;*
8  ;* Support email   : vdhoydonksd@skynet.be
9  ;*
10 ;* DESCRIPTION
11 ;* Basis routines om te communiceren met een I2C slave. Deze
12 ;* implementatie is voor één master en meerdere slaves op de
13 ;* I2C bus.
14 ;*
15 ;* Eigenschappen :
16 ;* * Er zijn geen interrupts gebruikt en zijn dus vrij.
17 ;* * Door de delay aan te passen kan normal of fast mode gekozen worden.
18 ;*   (standaard -> normal mode)
19 ;* * Alle AVR controllers kunnen gebruikt worden.
20 ;*
21 ;* I2C functies :
22 ;* 'i2c_start' -      Dwingt een start conditie af en zend
23 ;*                  adres en transfer richting.
24 ;* 'i2c_rep_start' -  Dwingt een repaeted start conditie af en
25 ;*                  zend address en transfer richting.
26 ;* 'i2c_do_transfer' - Zend of ontvangt data afhankelijk
27 ;*                  van de richting in address/dir bit.
28 ;* 'i2c_stop' -      Beëindigd de communicatie door een
29 ;*                  stop conditie af te dwingen.
30 ;*
31 ;* STATISTICS
32 ;* Lengte code : 79 words
33 ;* Registers   : r22,r23,r24,r25
34 ;* Interrupts  : geen
35 ;* Andere      : Twee I/O pins van portD
36 ;* XTAL waarde : Delay berekend voor 8MHz
37 ;*             bij herberekening delay alle waarden mogelijk
38 ;*
39 ;*****
40
41 ;.include "2313def.inc"
42
43 ;constanten
44 .equ    sclp    = 1
45 .equ    sdap    = 0
46 .equ    i2crd   = 1
47 .equ    i2cwr   = 0
48 .equ    b_dir   = 0
49
50 ;variabelen
51 .def    i2cdelay = r26
52 .def    i2cdata  = r27
53 .def    i2cadr   = r28
54 .def    i2cstat  = r29
55
56
57
58

```

```

59
60 ;*****
61 ;*           I2C communicatie           *
62 ;*****
63
64
65 ;***** FUNCTIES *****
66
67
68 ;*****
69 ;*
70 ;* FUNCTION
71 ;* i2c_hp_delay
72 ;* i2c_qp_delay
73 ;*
74 ;* DESCRIPTION
75 ;* hp - halve i2c clock periode delay (normaal: 5.0us / fast: 1.3us)
76 ;* qp - kwart i2c clock periode delay (normaal: 2.5us / fast: 0.6us)
77 ;*
78 ;* USAGE
79 ;* geen parameters
80 ;*
81 ;* RETURN
82 ;* geen parameters
83 ;*
84 ;*****
85
86 i2c_hp_delay:
87     ldi i2cdelay, 0x0D
88 delay_lp1:
89     dec i2cdelay
90     brne delay_lp1
91     nop
92     ret
93
94
95 i2c_qp_delay:
96     ldi i2cdelay, 0x06
97 delay_lp2:
98     dec i2cdelay
99     brne delay_lp2
100     nop
101     nop
102     ret
103
104
105
106 ;*****
107 ;*
108 ;* FUNCTION
109 ;* i2c_rep_start
110 ;*
111 ;* DESCRIPTION
112 ;* repaeted start afdwingen en slave address verzenden
113 ;*
114 ;* USAGE
115 ;* i2cadr - bevat het slave address en de transfer richting
116 ;* bit0 = 0 -> schrijven

```

```

117  ;* bit0 = 1 -> lezen
118  ;*
119  ;* RETURN
120  ;* carry flag - gecleared als de slave reageert
121  ;*
122  ;* NOTE
123  ;* routine wordt opgevolgd door i2c_start
124  ;*
125  ;*****
126
127  i2c_rep_start:
128      sbi DDRD, sclp
129      cbi DDRD, sdap
130      rcall i2c_hp_delay
131      cbi DDRD, sclp
132      rcall i2c_qp_delay
133
134
135
136  ;*****
137  ;*
138  ;* FUNCTION
139  ;* i2c_start
140  ;*
141  ;* DESCRIPTION
142  ;* start conditie afdwingen en slave address verzenden
143  ;*
144  ;* USAGE
145  ;* i2cadr - bevat het slave address en de transfer richting
146  ;* bit0 = 0 -> schrijven
147  ;* bit0 = 1 -> lezen
148  ;*
149  ;* RETURN
150  ;* carry flag - gecleared als de slave reageert
151  ;*
152  ;* NOTE
153  ;* routine wordt opgevolgd door i2c_write
154  ;*
155  ;*****
156
157  i2c_start:
158      mov i2cdata, i2cadr
159      sbi DDRD, sdap
160      rcall i2c_qp_delay
161
162
163
164  ;*****
165  ;*
166  ;* FUNCTION
167  ;* i2c_write
168  ;*
169  ;* DESCRIPTION
170  ;* Schrijft één byte data over de i2c bus. Kan zowel data, commando
171  ;* of address zijn.
172  ;*
173  ;* USAGE
174  ;* i2cdata - bevat de data die verstuurd moet worden

```

```

175  ;*
176  ;* RETURN
177  ;* carry flag - geset als de slave reageert
178  ;*
179  ;* NOTE
180  ;* routine wordt opgevolgd door i2c_get_ack
181  ;*
182  ;*****
183
184  i2c_write:
185      sec
186      rol i2cdata
187      rjmp  i2c_write_first
188
189  i2c_write_bit:
190      lsl i2cdata
191
192  i2c_write_first:
193      breq  i2c_get_ack
194      sbi DDRD, sclp
195
196      brcc  i2c_write_low
197      nop
198      cbi DDRD, sdap
199      rjmp  i2c_write_high
200
201  i2c_write_low:
202      sbi DDRD, sdap
203      rjmp  i2c_write_high
204
205  i2c_write_high:
206      rcall i2c_hp_delay
207      cbi DDRD, sclp
208      rcall i2c_hp_delay
209
210      rjmp  i2c_write_bit
211
212
213
214  ;*****
215  ;*
216  ;* FUNCTION
217  ;* i2c_get_ack
218  ;*
219  ;* DESCRIPTION
220  ;* Controleer of de slave met een ack antwoord.
221  ;*
222  ;* USAGE
223  ;* wordt door i2c_write gebruikt
224  ;*
225  ;* RETURN
226  ;* carry flag - gecleared als de slave reageert
227  ;*
228  ;*****
229
230  i2c_get_ack:
231      sbi DDRD, sclp
232      cbi DDRD, sdap

```

```

233     rcall    i2c_hp_delay
234     cbi DDRD, sclp
235
236 i2c_get_ack_wait:
237     sbis    pind, sclp
238
239     rjmp    i2c_get_ack_wait
240
241     clc
242     sbis    pind, sdap
243     sec
244     rcall    i2c_hp_delay
245     ret
246
247
248
249 ;*****
250 ;*
251 ;* FUNCTION
252 ;* i2c_do_transfer
253 ;*
254 ;* DESCRIPTION
255 ;* Voert een transfer op de i2c bus uit. Vanuit deze routine wordt
256 ;* ofwel i2c_write ofwel i2c_read aangeroepen.
257 ;*
258 ;* USAGE
259 ;* i2cadr - bevat dezelfde richting als wanneer i2c_start wordt
260 ;* aangeroepen.
261 ;*
262 ;* RETURN
263 ;* Hangt ervan af er een read of een write plaats vindt.
264 ;*
265 ;* NOTE
266 ;* routine wordt opgevolgd door i2c_read
267 ;*
268 ;*****
269
270 i2c_do_transfer:
271     sbrs    i2cadr, b_dir    ;als het address op een 0 eindigd
272     rjmp    i2c_write    ;wordt er naar write gesprongen, als
273                     ;het op een 1 eindigd wordt er
274                     ;doegegaan naar read.
275
276
277
278 ;*****
279 ;*
280 ;* FUNCTION
281 ;* i2c_read
282 ;*
283 ;* DESCRIPTION
284 ;* Leest één byte data over de i2c bus.
285 ;*
286 ;* USAGE
287 ;* carry flag - Als de carry geset is wordt er een no acknowledge aan
288 ;* de slave gegeven, dit wijst op een laatste read
289 ;* operatie. Als de carry gecleared is wordt er een
290 ;* acknowledge aan de slave gegeven, wat aangeeft dat

```

```

291  ;*          er nog data moet volgen.
292  ;*
293  ;* RETURN
294  ;* i2cdata - bevat de ontvangen data.
295  ;*
296  ;* NOTE
297  ;* routine wordt opgevolgd door i2c_put_ack.
298  ;*
299  ;*****
300
301  i2c_read:
302      rol i2cstat
303
304      ldi i2cdata, 0x01
305  i2c_read_bit:
306      sbi DDRD, sclp
307      rcall i2c_hp_delay
308
309      cbi DDRD,sclp
310      rcall i2c_hp_delay
311
312      clc
313      sbic  pind, sdap ;als sda hoog is wordt de carry geset
314      sec
315
316      rol i2cdata
317      brcc i2c_read_bit
318
319
320
321  ;*****
322  ;*
323  ;* FUNCTION
324  ;* i2c_put_ack
325  ;*
326  ;* DESCRIPTION
327  ;* Stuur een acknowledge naar de slave.
328  ;*
329  ;* USAGE
330  ;* Door i2c_read gebruikt
331  ;*
332  ;* RETURN
333  ;* geen parameters
334  ;*
335  ;*****
336
337  i2c_put_ack:
338      sbi DDRD, sclp ; force scl laag
339
340      ror i2cstat
341      brcc i2c_put_ack_low
342      cbi DDRD, sdap
343      rjmp i2c_put_ack_high
344  i2c_put_ack_low:
345      sbi DDRD,sclp
346
347  i2c_put_ack_high:
348      rcall i2c_hp_delay

```

```

349     cbi DDRD, sclp
350
351 i2c_put_ack_wait:
352     sbis     pind, sclp
353     rjmp    i2c_put_ack_wait
354     rcall   i2c_hp_delay
355     ret
356
357
358
359 ;*****
360 ;*
361 ;* FUNCTION
362 ;* i2c_stop
363 ;*
364 ;* DESCRIPTION
365 ;* Stop conditie afdwingen.
366 ;*
367 ;* USAGE
368 ;* no parameters
369 ;*
370 ;* RETURN
371 ;* no parameters
372 ;*
373 ;*****
374
375 i2c_stop:
376     sbi DDRD, sclp ;force scl laag
377     sbi DDRD, sdap ;force sda laag
378     rcall   i2c_hp_delay ;halve periode delay
379     cbi DDRD, sclp ;geef scl terug vrij
380     rcall   i2c_qp_delay ;quart periode delay
381     cbi DDRD, sdap ;geef sda terug vrij
382     rcall   i2c_hp_delay ;halve periode delay
383     ret
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406

```

407
408
409
410
411
412
413
414
415
416
417
418