

```

1  ;***** main.s *****
2  ; Program written by: put your names here: AYUSH SHRIVASTAVA
3  ; Date Created: 8/25/2013
4  ; Last Modified: 2/24/2015
5  ; Section 1-2pm      TA: Youngchun Kim
6  ; Lab number: 5
7  ; Brief description of the program
8  ;   A traffic light controller with 3 inputs and 8 output
9  ; Hardware connections
10 ;The "don't walk" and "walk" lights must be PF1 and PF3 respectively, but where to attach the others
    have some flexibility.
11 ;Obviously, you will not connect both inputs and outputs to the same pin.
12
13 ;Red south      PA7      PB5      PE5
14 ;Yellow south   PA6      PB4      PE4
15 ;Green south    PA5      PB3      PE3
16 ;Red west       PA4      PB2      PE2
17 ;Yellow west    PA3      PB1      PE1
18 ;Green west     PA2      PB0      PE0
19 ;Table 5.1. Possible ports to interface the traffic lights (PF1=red don't walk, PF3=green walk).
20
21 ;Walk sensor     PA4      PD2      PE2
22 ;South sensor    PA3      PD1      PE1
23 ;West sensor     PA2      PD0      PE0
24 ;Table 5.2. Possible ports to interface the sensors.
25 SYSCTL_RCGC2_R      EQU 0x400FE108
26 SYSCTL_RCGCGPIO_R   EQU 0x400FE608
27 GPIO_PORTA_DATA_R    EQU 0x400043FC
28 GPIO_PORTA_DIR_R     EQU 0x40004400
29 GPIO_PORTA_AFSEL_R   EQU 0x40004420
30 GPIO_PORTA_DEN_R     EQU 0x4000451C
31 GPIO_PORTB_DATA_R    EQU 0x400053FC
32 GPIO_PORTB_DIR_R     EQU 0x40005400
33 GPIO_PORTB_AFSEL_R   EQU 0x40005420
34 GPIO_PORTB_DEN_R     EQU 0x4000551C
35 GPIO_PORTD_DATA_R    EQU 0x400073FC
36 GPIO_PORTD_DIR_R     EQU 0x40007400
37 GPIO_PORTD_AFSEL_R   EQU 0x40007420
38 GPIO_PORTD_DEN_R     EQU 0x4000751C
39 GPIO_PORTE_DATA_R    EQU 0x400243FC
40 GPIO_PORTE_DIR_R     EQU 0x40024400
41 GPIO_PORTE_AFSEL_R   EQU 0x40024420
42 GPIO_PORTE_DEN_R     EQU 0x4002451C
43 GPIO_PORTF_DATA_R    EQU 0x400253FC
44 GPIO_PORTF_DIR_R     EQU 0x40025400
45 GPIO_PORTF_AFSEL_R   EQU 0x40025420
46 GPIO_PORTF_DEN_R     EQU 0x4002551C
47 NVIC_ST_CURRENT_R    EQU 0xE000E018
48 NVIC_ST_CTRL_R       EQU 0xE000E010
49 NVIC_ST_RELOAD_R     EQU 0xE000E014
50 GPIO_PORTE_AMSEL_R   EQU 0x40024528
51 GPIO_PORTE_PCTL_R    EQU 0x4002452C
52 GPIO_PORTB_AMSEL_R   EQU 0x40005528
53 GPIO_PORTB_PCTL_R    EQU 0x4000552C
54 GPIO_PORTF_AMSEL_R   EQU 0x40025528
55 GPIO_PORTF_PCTL_R    EQU 0x4002552C
56
57 NUMBER               EQU 800000
58
59 goN1      EQU 0
60 goN2      EQU 1
61 goN3      EQU 2
62 waitN     EQU 3
63 stopN     EQU 4
64 goE1      EQU 5
65 goE2      EQU 6
66 goE3      EQU 7
67 waitE     EQU 8
68 stopE     EQU 9
69 pOn       EQU 10
70 pBlink1On EQU 11
71 pBlink1Off EQU 12

```

```
72  pBlink2On    EQU    13
73  pBlink2Off   EQU    14
74  pBlink3On    EQU    15
75  pBlink3Off   EQU    16
76  pBlink4On    EQU    17
77  pBlink4Off   EQU    18
78  pStop        EQU    19
79
80
81
82
83      AREA     DATA, ALIGN=2
84
85  STRUCT
86      FILL 1,0; 0
87      FILL 1,0x21;1
88      FILL 1,0x2;2
89      FILL 1,200;3
90      FILL 1,goN2;4
91      FILL 1,goN1;5
92      FILL 1,goN2;6
93      FILL 1,goN2;7
94      FILL 1,goN2;8
95      FILL 1,goN2;9
96      FILL 1,goN2;10
97      FILL 1,goN2 ; 11
98      FILL 1,0x21;12
99      FILL 1,0x2;13
100     FILL 1,200;14
101     FILL 1,goN3;15
102     FILL 1,goN1;16
103     FILL 1,goN3;17
104     FILL 1,goN3;18
105     FILL 1,goN3;19
106     FILL 1,goN3;20
107     FILL 1,goN3;21
108     FILL 1,goN3 ; 22//goN2
109     FILL 1,0x21;23
110     FILL 1,0x2;24
111     FILL 1,100;25
112     FILL 1,waitN;26
113     FILL 1,goN1;27
114     FILL 1,waitN;28
115     FILL 1,waitN;29
116     FILL 1,waitN;30
117     FILL 1,waitN;31
118     FILL 1,waitN;32
119     FILL 1,waitN ; 33//goN3
120     FILL 1,0x22;34
121     FILL 1,0x2;35
122     FILL 1,200;36
123     FILL 1,stopN;37
124     FILL 1,goN1;38
125     FILL 1,stopN;39
126     FILL 1,stopN;40
127     FILL 1,stopN;41
128     FILL 1,stopN;42
129     FILL 1,stopN;43
130     FILL 1,stopN;44
131     FILL 1,0x24;45
132     FILL 1,0x2;46
133     FILL 1,200;47
134     FILL 1,goE1;48
135     FILL 1,goN1;49
136     FILL 1,goE1;50
137     FILL 1,goE1;51
138     FILL 1,pOn;52
139     FILL 1,pOn;53
140     FILL 1,goE1;54
141     FILL 1,goE1 ;55
142     FILL 1,0x0C;56
143     FILL 1,0x2;57
```

//stopN

```
144 FILL 1,200;58
145 FILL 1,goE2;59
146 FILL 1,goE2;60
147 FILL 1,goE1;61
148 FILL 1,goE2;62
149 FILL 1,goE2;63
150 FILL 1,goE2;64
151 FILL 1,goE2;65
152 FILL 1,goE2 ; 66//goE1
153 FILL 1,0x0C;67
154 FILL 1,0x2;68
155 FILL 1,200;69
156 FILL 1,goE3;70
157 FILL 1,goE3;71
158 FILL 1,goE1;72
159 FILL 1,goE3;73
160 FILL 1,goE3;74
161 FILL 1,goE3;75
162 FILL 1,goE3;76
163 FILL 1,goE3 ; 77//goE2
164 FILL 1,0x0C;78
165 FILL 1,0x2;79
166 FILL 1,100;80
167 FILL 1,waitE;81
168 FILL 1,waitE;82
169 FILL 1,goE1;83
170 FILL 1,waitE;84
171 FILL 1,waitE;85
172 FILL 1,waitE;86
173 FILL 1,waitE;87
174 FILL 1,waitE ;88//goE3
175 FILL 1,0x14;89
176 FILL 1,0x2;90
177 FILL 1,200;91
178 FILL 1,stopE;92
179 FILL 1,stopE;93
180 FILL 1,goE1;94
181 FILL 1,stopE;95
182 FILL 1,stopE;96
183 FILL 1,stopE;97
184 FILL 1,stopE;98
185 FILL 1,stopE ; 99//waitE
186 FILL 1,0x24;100
187 FILL 1,0x2 ;101
188 FILL 1,200;102
189 FILL 1,pOn;103
190 FILL 1,goN1;104
191 FILL 1,goE1;105
192 FILL 1,goN1;106
193 FILL 1,pOn;107
194 FILL 1,pOn;108
195 FILL 1,pOn;109
196 FILL 1,pOn ;110 //stopE
197 FILL 1,0x24;111
198 FILL 1,0x8 ;112
199 FILL 1,200;113
200 FILL 1,pBlink1On;114
201 FILL 1,pBlink1On;115
202 FILL 1,pBlink1On;116
203 FILL 1,pBlink1On;117
204 FILL 1,pOn;118
205 FILL 1,pBlink1On;119
206 FILL 1,pBlink1On;120
207 FILL 1,pBlink1On;121 //pOn
208 FILL 1,0x24;122
209 FILL 1,0x2;123
210 FILL 1,30;124
211 FILL 1,pBlink1Off;125
212 FILL 1,pBlink1Off;126
213 FILL 1,pBlink1Off;127
214 FILL 1,pBlink1Off;128
215 FILL 1,pOn; 129
```

```
216 FILL 1,pBlink1Off;130
217 FILL 1,pBlink1Off;131
218 FILL 1,pBlink1Off ;132 //pBlink1On
219 FILL 1,0x24;133
220 FILL 1,0x0;134
221 FILL 1,30;135
222 FILL 1,pBlink2On;136
223 FILL 1,pBlink2On;137
224 FILL 1,pBlink2On;138
225 FILL 1,pBlink2On;139
226 FILL 1,pOn;140
227 FILL 1,pBlink2On;141
228 FILL 1,pBlink2On;142
229 FILL 1,pBlink2On;143 //pBlink1Off
230 FILL 1,0x24;144
231 FILL 1,0x2;145
232 FILL 1,30;146
233 FILL 1,pBlink2Off;147
234 FILL 1,pBlink2Off;148
235 FILL 1,pBlink2Off;149
236 FILL 1,pBlink2Off;150
237 FILL 1,pOn;151
238 FILL 1,pBlink2Off;152
239 FILL 1,pBlink2Off;153
240 FILL 1,pBlink2Off;154 //pBlink2On
241 FILL 1,0x24;155
242 FILL 1,0x0;156
243 FILL 1,30;157
244 FILL 1,pBlink3On;158
245 FILL 1,pBlink3On;159
246 FILL 1,pBlink3On;160
247 FILL 1,pBlink3On;161
248 FILL 1,pOn;162
249 FILL 1,pBlink3On;163
250 FILL 1,pBlink3On;164
251 FILL 1,pBlink3On;165 //pBlink2Off
252 FILL 1,0x24;166
253 FILL 1,0x2;167
254 FILL 1,30;168
255 FILL 1,pBlink3Off;169
256 FILL 1,pBlink3Off;170
257 FILL 1,pBlink3Off;171
258 FILL 1,pBlink3Off;172
259 FILL 1,pOn;173
260 FILL 1,pBlink3Off;174
261 FILL 1,pBlink3Off;175
262 FILL 1,pBlink3Off;176 //pBlink3On
263 FILL 1,0x24;177
264 FILL 1,0x0;178
265 FILL 1,30;179
266 FILL 1,pBlink4On;180
267 FILL 1,pBlink4On;181
268 FILL 1,pBlink4On;182
269 FILL 1,pBlink4On;183
270 FILL 1,pOn;184
271 FILL 1,pBlink4On;185
272 FILL 1,pBlink4On;186
273 FILL 1,pBlink4On;187 //pBlink3Off
274 FILL 1,0x24;188
275 FILL 1,0x2;189
276 FILL 1,30;190
277 FILL 1,pBlink4Off;191
278 FILL 1,pBlink4Off;192
279 FILL 1,pBlink4Off;193
280 FILL 1,pBlink4Off;194
281 FILL 1,pOn;195
282 FILL 1,pBlink4Off;196
283 FILL 1,pBlink4Off;197
284 FILL 1,pBlink4Off ;198//pBlink4On
285 FILL 1,0x24;199
286 FILL 1,0x0;200
287 FILL 1,30;201
```

```

288     FILL 1,pStop;202
289     FILL 1,pStop;203
290     FILL 1,pStop;204
291     FILL 1,pStop;205
292     FILL 1,pOn;206
293     FILL 1,pStop;207
294     FILL 1,pStop;208
295     FILL 1,pStop ;209 //pBlink4Off
296     FILL 1,0x24;210
297     FILL 1,0x2;211
298     FILL 1,200;212
299     FILL 1,goN1;213
300     FILL 1,goN1;214
301     FILL 1,goE1;215
302     FILL 1,goN1;216
303     FILL 1,pOn;217
304     FILL 1,goN1;218
305     FILL 1,goE1;219
306     FILL 1,goN1;220                                     //pStop
307
308
309
310
311
312     ALIGN
313     AREA    |.text|, CODE, READONLY, ALIGN=2
314     THUMB
315     EXPORT  Start
316     IMPORT  PLL_Init
317
318 Start
319 ; the DCM.DLL -pCM4 -dLaunchPadDLL debugger setting does NOT grade or simulate traffic
320 ; the DCM.DLL -pCM4 -dEE319KLab5 debugger setting DOES not test all specifications for EE319K
321 ; INITIALIZATION
322
323
324     LDR R1, =SYSCTL_RCGC2_R; SYSTEM CLOCK FOR PORTS BEF
325     LDR R2, [R1];
326     ;//AND R2,R2,#0;
327     ORR R2,R2,#0X32;
328     STR R2,[R1];
329     NOP
330     NOP
331
332     LDR R1, =GPIO_PORTE_AMSEL_R; PORT E AMSEL
333     LDR R2, [R1];
334     AND R2,R2,#0X8;
335     ;//ORR R2,R2,#0X5;
336     STR R2,[R1];
337
338     LDR R1, =GPIO_PORTE_PCTL_R; PORT E PCTL
339     LDR R2, [R1];
340     AND R2,R2,#0X00;
341     ;//ORR R2,R2,#0X5;
342     STR R2,[R1];
343
344     LDR R1, =GPIO_PORTE_DIR_R; PORT E DIR
345     LDR R2, [R1];
346     AND R2,R2,#0X8;
347     ;//ORR R2,R2,#0X5;
348     STR R2,[R1];
349
350     LDR R1, =GPIO_PORTE_AFSEL_R; PORT E AFSEL
351     LDR R2, [R1];
352     AND R2,R2,#0X8;
353     ;//ORR R2,R2,#0X5;
354     STR R2,[R1];
355
356     LDR R1, =GPIO_PORTE_DEN_R; PORT E DEN
357     LDR R2, [R1];
358     ;//AND R2,R2,#0X8;
359     ORR R2,R2,#0X7;

```

```
360     STR R2, [R1];
361
362
363
364
365
366
367
368     LDR R1, =GPIO_PORTB_AMSEL_R; PORT B AMSEL
369     LDR R2, [R1];
370     AND R2,R2,#0X40;
371     ;//ORR R2,R2,#0X5;
372     STR R2,[R1];
373
374     LDR R1, =GPIO_PORTB_PCTL_R; PORT B PCTL
375     LDR R2, [R1];
376     MOV R2,#0X00;
377     ;//ORR R2,R2,#0X5;
378     STR R2,[R1];
379
380     LDR R1, =GPIO_PORTB_DIR_R; PORT B DIR
381     LDR R2, [R1];
382     ;//AND R2,R2,#0X8;
383     ORR R2,R2,#0X3F;
384     STR R2,[R1];
385
386     LDR R1, =GPIO_PORTB_AFSEL_R; PORT B AFSEL
387     LDR R2, [R1];
388     AND R2,R2,#0X40;
389     ;//ORR R2,R2,#0X5;
390     STR R2,[R1];
391
392     LDR R1, =GPIO_PORTB_DEN_R; PORT B DEN
393     LDR R2, [R1];
394     ;//AND R2,R2,#0X8;
395     ORR R2,R2,#0X3F;
396     STR R2,[R1];
397
398     LDR R1, =GPIO_PORTF_AMSEL_R; PORT F AMSEL
399     LDR R2, [R1];
400     AND R2,R2,#0X5;
401     ;//ORR R2,R2,#0X5;
402     STR R2,[R1];
403
404     LDR R1, =GPIO_PORTF_PCTL_R; PORT F PCTL
405     LDR R2, [R1];
406     MOV R2,#0X00;
407     ;//ORR R2,R2,#0X5;
408     STR R2,[R1];
409
410     LDR R1, =GPIO_PORTF_DIR_R; PORT F DIR
411     LDR R2, [R1];
412     ;//AND R2,R2,#0X8;
413     ORR R2,R2,#0XA;
414     STR R2,[R1];
415
416     LDR R1, =GPIO_PORTF_AFSEL_R; PORT F AFSEL
417     LDR R2, [R1];
418     AND R2,R2,#0X5;
419     ;//ORR R2,R2,#0X5;
420     STR R2,[R1];
421
422     LDR R1, =GPIO_PORTF_DEN_R; PORT F DEN
423     LDR R2, [R1];
424     ;//AND R2,R2,#0X8;
425     ORR R2,R2,#0XA;
426     STR R2,[R1];
427
428     BL    PLL_Init      ; running at 80 MHz
429
430
431     LDR R1, =NVIC_ST_CTRL_R; SYSTICK TIMER INITIALIZATION
```

```

432     LDR R2, [R1];
433     AND R2,R2,#0;
434     ORR R2,R2,#0x5;
435     STR R2,[R1];
436     ; R5 IS THE POINTER OF WHERE THE STRUCT START IS
437     LDR R5, =STRUCT;
438     MOV R6,#1; R6 IS RELATIVE VALUE
439     ADD R5,R5,R6;
440
441     MOV R7,R5;
442     ;ADD R7,R5,R6; R7 IS THE ACTUAL ADDRESS
443     ;MOV R8,R8,#0; R8 WILL CARRY THE DATA
444
445
446
447
448
449 loop
450
451     LDR R1, =GPIO_PORTB_DATA_R; PORT B DATA
452     LDR R2, [R7]; GETTING DATA FROM R7 POINTER
453     STR R2,[R1]; STORING IT BACK
454
455     ADD R7,R7,#1
456
457     LDR R1, =GPIO_PORTF_DATA_R; PORT F DATA
458     LDR R2, [R7]; GETTING DATA FROM R7 POINTER
459     STR R2,[R1]; STORING IT BACK
460
461     ADD R7,R7,#1
462
463     LDR R0,[R7]; R0 GETS THE MULTIPLE OF 10MS
464     BL SysTick_WAIT10MS;
465
466     ADD R7,R7,#1;
467     ; NOW GOING INTO STAGES TO GET THE INPUT AND CALCULATING THE NEXT STATE
468     LDR R1, =GPIO_PORTE_DATA_R; PORT E DATA
469     LDR R2, [R1]; R2 HAS THE DATA OF INPUTS
470     ADD R7,R7,R2; DONT CHANGE R2 ANYMORE
471     ;R7 HAS THE POINTER OF THE ADDRESS OF WHERE THE NEXT STAGES ARE
472
473     LDR R6,[R7]; R2 HAS THE MULTIPLE OF WHERE TO GO NEXT
474     MOV R2,#11;
475     MUL R6,R6,R2;
476     ADD R7,R5,R6;
477
478
479
480     B     loop
481
482 SysTick_Wait
483     ;R0 IS THE NUMBER TO PASS FOR THE NUMBER OF DELAY
484     LDR R1,=NVIC_ST_RELOAD_R;
485     LDR R2, [R1];
486     MOV R2,R0; MAKE RELOAD VALUE THE NUMBER PASSES
487     STR R2,[R1];
488     LDR R1,=NVIC_ST_CURRENT_R;
489     LDR R2, [R1];
490     MOV R2,#0; MAKE CURRENT 0
491     STR R2,[R1];
492     LDR R1,=NVIC_ST_CTRL_R; GET THE CONTROLLER VALUE
493 WAIT_LOOP
494     LDR R2, [R1];
495     AND R2,R2,#0x10000;
496     CMP R2,#0;
497     BNE WAIT_LOOP;
498     BX LR;
499
500 SysTick_WAIT10MS
501     ;R0 IS THE NUMBER OF 10MS NEEDED TO WAIT
502     ;R1-3 ARE SCRATCH
503     MOV R3,R0;

```

```
504     PUSH {LR};
505     CMP R3,#0;
506     BEQ DONE_WAIT10MS
507 WAIT10MS_LOOP
508     SUB R3,#1;
509     LDR R0,=NUMBER;
510     BL SysTick_Wait;
511     CMP R3,#0;
512     BNE WAIT10MS_LOOP
513 DONE_WAIT10MS
514     POP {PC};
515
516
517     ALIGN          ; make sure the end of this section is aligned
518     END            ; end of file
519
```