

```

1  /+-----+
2  Use :
3  \+-----+
4  void FCM_Main()
5  {
6
7  // Name: Call Macro, Type: Call Component Macro: LCDI2C1::Start()
8  FCD_0be11_LCDI2C1_Start();
9
10 // Name: Loop, Type: Loop: While 1
11 while (1)
12 {
13
14 // Name: Group, Type: Group
15 {
16
17 // Name: dip0, Type: Input: B0 -> dip0
18 FCV_DIP0 = GET_PORT_PIN(B,0);
19
20 // Name: Input, Type: Input: B1 -> dip1
21 FCV_DIP1 = GET_PORT_PIN(B,1);
22
23 // Name: Input, Type: Input: B2 -> dip2
24 FCV_DIP2 = GET_PORT_PIN(B,2);
25
26 // Name: Input, Type: Input: B3 -> dip3
27 FCV_DIP3 = GET_PORT_PIN(B,3);
28
29 // Name: Calculation, Type: Calculation:
30 // BCD_1[0] = 0
31 // BCD_1[1] = 1
32 // BCD_1[2] = 2
33 // BCD_1[3] = 3
34 // BCD_1[4] = 4
35 // BCD_1[5] = 5
36 // BCD_1[6] = 6
37 // BCD_1[7] = 7
38 // BCD_1[8] = 8
39 // BCD_1[9] = 9
40 FCV_BCD_1[0] = 0;
41 FCV_BCD_1[1] = 1;
42 FCV_BCD_1[2] = 2;
43 FCV_BCD_1[3] = 3;
44 FCV_BCD_1[4] = 4;
45 FCV_BCD_1[5] = 5;
46 FCV_BCD_1[6] = 6;
47 FCV_BCD_1[7] = 7;
48 FCV_BCD_1[8] = 8;
49 FCV_BCD_1[9] = 9;

```

```

50
51 // Name: Switch, Type: Switch: dip0 + 2 * dip1 + 4 * dip2 + 8 * dip3?
52 switch (FCV_DIP0 + 2 * FCV_DIP1 + 4 * FCV_DIP2 + 8 * FCV_DIP3)
53 {
54 case 1:
55 {
56 // Name: Calculation, Type: Calculation:
57 // setA = 1
58 FCV_SETA = 1;
59
60 break;
61 }
62 case 2:
63 {
64 // Name: Calculation, Type: Calculation:
65 // setA = 2
66 FCV_SETA = 2;
67
68 break;
69 }
70 case 3:
71 {
72 // Name: Calculation, Type: Calculation:
73 // setA = 3
74 FCV_SETA = 3;
75
76 break;
77 }
78 case 4:
79 {
80 // Name: Calculation, Type: Calculation:
81 // setA = 4
82 FCV_SETA = 4;
83
84 break;
85 }
86 case 5:
87 {
88 // Name: Calculation, Type: Calculation:
89 // setA = 5
90 FCV_SETA = 5;
91
92 break;
93 }
94 case 6:
95 {
96 // Name: Calculation, Type: Calculation:
97 // setA = 6
98 FCV_SETA = 6;

```

```

99
100         break;
101     }
102     case 7:
103     {
104         // Name: Calculation, Type: Calculation:
105         // setA = 7
106         FCV_SETA = 7;
107
108         break;
109     }
110     case 8:
111     {
112         // Name: Calculation, Type: Calculation:
113         // setA = 8
114         FCV_SETA = 8;
115
116         break;
117     }
118     case 9:
119     {
120         // Name: Calculation, Type: Calculation:
121         // setA = 9
122         FCV_SETA = 9;
123
124         break;
125     }
126     default:
127     {
128         // Name: Calculation, Type: Calculation:
129         // setA = 0
130         FCV_SETA = 0;
131
132     }
133 }
134
135 }
136
137 // Name: Call Macro, Type: Call Component Macro: LCDI2C1::Cursor(10, 0)
138 FCD_Obe11_LCDI2C1_Cursor(10, 0);
139
140 // Name: Call Macro, Type: Call Component Macro: LCDI2C1::PrintNumber(setA)
141 FCD_Obe11_LCDI2C1_PrintNumber(FCV_SETA);
142
143 // Name: Group, Type: Group
144 {
145
146     // Name: dip0, Type: Input: B4 -> dip0
147     FCV_DIP0 = GET_PORT_PIN(B,4);
148

```

```

149     // Name: Input, Type: Input: B5 -> dip1
150     FCV_DIP1 = GET_PORT_PIN(B,5);
151
152     // Name: Input, Type: Input: B6 -> dip2
153     FCV_DIP2 = GET_PORT_PIN(B,6);
154
155     // Name: Input, Type: Input: B7 -> dip3
156     FCV_DIP3 = GET_PORT_PIN(B,7);
157
158     // Name: Calculation, Type: Calculation:
159     // BCD_1[0] = 0
160     // BCD_1[1] = 1
161     // BCD_1[2] = 2
162     // BCD_1[3] = 3
163     // BCD_1[4] = 4
164     // BCD_1[5] = 5
165     // BCD_1[6] = 6
166     // BCD_1[7] = 7
167     // BCD_1[8] = 8
168     // BCD_1[9] = 9
169     FCV_BCD_1[0] = 0;
170     FCV_BCD_1[1] = 1;
171     FCV_BCD_1[2] = 2;
172     FCV_BCD_1[3] = 3;
173     FCV_BCD_1[4] = 4;
174     FCV_BCD_1[5] = 5;
175     FCV_BCD_1[6] = 6;
176     FCV_BCD_1[7] = 7;
177     FCV_BCD_1[8] = 8;
178     FCV_BCD_1[9] = 9;
179
180     // Name: Switch, Type: Switch: dip0 + 2 * dip1 + 4 * dip2 + 8 * dip3?
181     switch (FCV_DIP0 + 2 * FCV_DIP1 + 4 * FCV_DIP2 + 8 * FCV_DIP3)
182     {
183     case 1:
184     {
185         // Name: Calculation, Type: Calculation:
186         // setB = 1
187         FCV_SETB = 1;
188
189         break;
190     }
191     case 2:
192     {
193         // Name: Calculation, Type: Calculation:
194         // setB = 2
195         FCV_SETB = 2;
196
197         break;

```

```

198     }
199     case 3:
200     {
201         // Name: Calculation, Type: Calculation:
202         // setB = 3
203         FCV_SETB = 3;
204
205         break;
206     }
207     case 4:
208     {
209         // Name: Calculation, Type: Calculation:
210         // setB = 4
211         FCV_SETB = 4;
212
213         break;
214     }
215     case 5:
216     {
217         // Name: Calculation, Type: Calculation:
218         // setB = 5
219         FCV_SETB = 5;
220
221         break;
222     }
223     case 6:
224     {
225         // Name: Calculation, Type: Calculation:
226         // setB = 6
227         FCV_SETB = 6;
228
229         break;
230     }
231     case 7:

```

```

231     case 7:
232     {
233         // Name: Calculation, Type: Calculation:
234         // setB = 7
235         FCV_SETB = 7;
236
237         break;
238     }
239     case 8:
240     {
241         // Name: Calculation, Type: Calculation:
242         // setB = 8
243         FCV_SETB = 8;
244
245         break;
246     }
247     case 9:
248     {
249         // Name: Calculation, Type: Calculation:
250         // setB = 9
251         FCV_SETB = 9;
252
253         break;
254     }
255     default:
256     {
257         // Name: Calculation, Type: Calculation:
258         // setB = 0
259         FCV_SETB = 0;
260
261     }
262 }
263
264 }

```

```

266 // Name: Call Macro, Type: Call Component Macro: LCDI2C1::Cursor(11, 0)
267 FCD_Obel1_LCDI2C1_Cursor(11, 0);
268
269 // Name: Call Macro, Type: Call Component Macro: LCDI2C1::PrintNumber(setB)
270 FCD_Obel1_LCDI2C1_PrintNumber(FCV_SETB);
271
272 // Name: Group, Type: Group
273 {
274
275     // Name: dip0, Type: Input: D0 -> dip0
276     FCV_DIP0 = GET_PORT_PIN(D,0);
277
278     // Name: Input, Type: Input: D1 -> dip1
279     FCV_DIP1 = GET_PORT_PIN(D,1);
280

```

```

280
281 // Name: Input, Type: Input: D4 -> dip2
282 FCV_DIP2 = GET_PORT_PIN(D,4);
283
284 // Name: Input, Type: Input: D5 -> dip3
285 FCV_DIP3 = GET_PORT_PIN(D,5);
286
287 // Name: Calculation, Type: Calculation:
288 // BCD_1[0] = 0
289 // BCD_1[1] = 1
290 // BCD_1[2] = 2
291 // BCD_1[3] = 3
292 // BCD_1[4] = 4
293 // BCD_1[5] = 5
294 // BCD_1[6] = 6
295 // BCD_1[7] = 7
296 // BCD_1[8] = 8
297 // BCD_1[9] = 9
298 FCV_BCD_1[0] = 0;
299 FCV_BCD_1[1] = 1;
300 FCV_BCD_1[2] = 2;
301 FCV_BCD_1[3] = 3;
302 FCV_BCD_1[4] = 4;
303 FCV_BCD_1[5] = 5;
304 FCV_BCD_1[6] = 6;
305 FCV_BCD_1[7] = 7;
306 FCV_BCD_1[8] = 8;
307 FCV_BCD_1[9] = 9;
308
309 // Name: Switch, Type: Switch: dip0 + 2 * dip1 + 4 * dip2 + 8 * dip3?
310 switch (FCV_DIP0 + 2 * FCV_DIP1 + 4 * FCV_DIP2 + 8 * FCV_DIP3)
311 {
312     case 1:
313     {
314         // Name: Calculation, Type: Calculation:
315         // setC = 1
316         FCV_SETC = 1;
317
318         break;
319     }
320     case 2:
321     {
322         // Name: Calculation, Type: Calculation:
323         // setC = 2
324         FCV_SETC = 2;
325
326         break;
327     }
328     case 3:
329     {

```

```

330         // Name: Calculation, Type: Calculation:
331         // setC = 3
332         FCV_SETC = 3;
333
334         break;
335     }
336     case 4:
337     {
338         // Name: Calculation, Type: Calculation:
339         // setC = 4
340         FCV_SETC = 4;
341
342         break;
343     }
344     case 5:
345     {
346         // Name: Calculation, Type: Calculation:
347         // setC = 5
348         FCV_SETC = 5;
349
350         break;
351     }
352     case 6:
353     {
354         // Name: Calculation, Type: Calculation:
355         // setC = 6
356         FCV_SETC = 6;
357
358         break;
359     }
360     case 7:
361     {
362         // Name: Calculation, Type: Calculation:
363         // setC = 7
364         FCV_SETC = 7;
365
366         break;
367     }
368     case 8:
369     {
370         // Name: Calculation, Type: Calculation:
371         // setC = 8
372         FCV_SETC = 8;
373
374         break;
375     }
376     case 9:
377     {
378         // Name: Calculation, Type: Calculation:
379         // setC = 9

```

```

380         FCV_SETC = 9;
381
382         break;
383     }
384     default:
385     {
386         // Name: Calculation, Type: Calculation:
387         // setC = 0
388         FCV_SETC = 0;
389
390     }
391 }
392
393 }
394
395 // Name: Call Macro, Type: Call Component Macro: LCDI2C1::Cursor(12, 0)
396 FCD_0be11_LCDI2C1_Cursor(12, 0);
397
398 // Name: Call Macro, Type: Call Component Macro: LCDI2C1::PrintNumber(setC)
399 FCD_0be11_LCDI2C1_PrintNumber(FCV_SETC);
400
401 // Name: Group, Type: Group
402 {
403
404     // Name: dip0, Type: Input: D6 -> dip0
405     FCV_DIP0 = GET_PORT_PIN(D,6);
406
407     // Name: Input, Type: Input: D7 -> dip1
408     FCV_DIP1 = GET_PORT_PIN(D,7);
409
410     // Name: Input, Type: Input: C0 -> dip2
411     FCV_DIP2 = GET_PORT_PIN(C,0);
412
413     // Name: Input, Type: Input: C1 -> dip3
414     FCV_DIP3 = GET_PORT_PIN(C,1);
415
416     // Name: Calculation, Type: Calculation:
417     // BCD_1[0] = 0
418     // BCD_1[1] = 1
419     // BCD_1[2] = 2
420     // BCD_1[3] = 3
421     // BCD_1[4] = 4
422     // BCD_1[5] = 5
423     // BCD_1[6] = 6
424     // BCD_1[7] = 7
425     // BCD_1[8] = 8
426     // BCD_1[9] = 9
427     FCV_BCD_1[0] = 0;
428     FCV_BCD_1[1] = 1;
429     FCV_BCD_1[2] = 2;

```

```

430     FCV_BCD_1[3] = 3;
431     FCV_BCD_1[4] = 4;
432     FCV_BCD_1[5] = 5;
433     FCV_BCD_1[6] = 6;
434     FCV_BCD_1[7] = 7;
435     FCV_BCD_1[8] = 8;
436     FCV_BCD_1[9] = 9;
437
438     // Name: Switch, Type: Switch: dip0 + 2 * dip1 + 4 * dip2 + 8 * dip3?
439     switch (FCV_DIP0 + 2 * FCV_DIP1 + 4 * FCV_DIP2 + 8 * FCV_DIP3)
440     {
441     case 1:
442     {
443         // Name: Calculation, Type: Calculation:
444         // setD = 1
445         FCV_SETD = 1;
446
447         break;
448     }
449     case 2:
450     {
451         // Name: Calculation, Type: Calculation:
452         // setD = 2
453         FCV_SETD = 2;
454
455         break;
456     }
457     case 3:
458     {
459         // Name: Calculation, Type: Calculation:
460         // setD = 3
461         FCV_SETD = 3;
462
463         break;
464     }
465     case 4:
466     {
467         // Name: Calculation, Type: Calculation:
468         // setD = 4
469         FCV_SETD = 4;
470
471         break;
472     }
473     case 5:
474     {
475         // Name: Calculation, Type: Calculation:
476         // setD = 5
477         FCV_SETD = 5;
478
479         break;

```

```

480     }
481     case 6:
482     {
483         // Name: Calculation, Type: Calculation:
484         // setD = 6
485         FCV_SETD = 6;
486
487         break;
488     }
489     case 7:
490     {
491         // Name: Calculation, Type: Calculation:
492         // setD = 7
493         FCV_SETD = 7;
494
495         break;
496     }
497     case 8:
498     {
499         // Name: Calculation, Type: Calculation:
500         // setD = 8
501         FCV_SETD = 8;
502
503         break;
504     }
505     case 9:
506     {
507         // Name: Calculation, Type: Calculation:
508         // setD = 9
509         FCV_SETD = 9;
510
511         break;
512     }
513     default:
514     {
515         // Name: Calculation, Type: Calculation:
516         // setD = 0
517         FCV_SETD = 0;
518
519     }
520 }
521
522 }

```

```

523
524 // Name: Call Macro, Type: Call Component Macro: LCDI2C1::Cursor(13, 0)
525 FCD_0be11_LCDI2C1_Cursor(13, 0);
526

```

```

527 // Name: Call Macro, Type: Call Component Macro: LCDI2C1::PrintNumber(setD)
528 FCD_0be11_LCDI2C1_PrintNumber(FCV_SETD);

```

```

529
530 // Name: Calculation, Type: Calculation:
531 // totale = setA * 1000 + setB * 100 + setC * 10 + setD
532 FCV_TOTALE = FCV_SETA * 1000 + FCV_SETB * 100 + FCV_SETC * 10 + FCV_SETD;
533

```

```

534
535 }
536

```

```

537 }
538

```