








MAIN

```
1  /*-----*\
2  Use :
3  \*-----*/
4  void FCM_Main()
5  {
6
7      // Name: Interrupt, Type: Interrupt: Enable INTO
8      EICRA |= (1 << ISC01) | (1 << ISC00);
9      sei();
10     EIMSK |= (1 << INT0);
11
12     // Name: Interrupt, Type: Interrupt: Enable INT1
13     EICRA |= (1 << ISC11) | (1 << ISC10);
14     sei();
15     EIMSK |= (1 << INT1);
16
17     // Name: Connection Point, Type: Connection Point: [A]: A
18     FCC_Main_A:
19     ;
20
21     // Name: Calculation, Type: Calculation:
22     // clock = 0
23     // luci[0] = 0b00000000
24     // luci[1] = 0b00000001
25     // luci[2] = 0b00000010
26     // luci[3] = 0b00000100
27     // luci[4] = 0b00001000
28     // luci[5] = 0b00010000
29     // luci[6] = 0b00100000
30     // luci[7] = 0b01000000
31     // luci[8] = 0b10000000
32     // luci[9] = 0b00000000
33     FCV_CLOCK = 0;
34     FCV_LUCI[0] = 0;
35     FCV_LUCI[1] = 1;
36     FCV_LUCI[2] = 2;
37     FCV_LUCI[3] = 4;
38     FCV_LUCI[4] = 8;
39     FCV_LUCI[5] = 16;
40     FCV_LUCI[6] = 32;
41     FCV_LUCI[7] = 64;
42     FCV_LUCI[8] = 128;
43     FCV_LUCI[9] = 0;
44
45     // Name: Loop, Type: Loop: While 1
46     while (1)
47     {
48
49         // Name: Input, Type: Input: D5 -> dip0
50         FCV_DIP0 = GET_PORT_PIN(D,5);
```

```
51
52 // Name: Input, Type: Input: D6 -> dip1
53 FCV_DIP1 = GET_PORT_PIN(D,6);
54
55 // Name: Input, Type: Input: D7 -> dip3
56 FCV_DIP3 = GET_PORT_PIN(D,7);
57
58 // Name: Switch, Type: Switch: dip0 + 2 * dip1 + 4 * dip3?
59 switch (FCV_DIP0 + 2 * FCV_DIP1 + 4 * FCV_DIP3)
60 {
61     case 1:
62     {
63         // Name: Calculation, Type: Calculation:
64         // reset = 2
65         FCV_RESET = 2;
66
67         break;
68     }
69     case 2:
70     {
71         // Name: Calculation, Type: Calculation:
72         // reset = 3
73         FCV_RESET = 3;
74
75         break;
76     }
77     case 3:
78     {
79         // Name: Calculation, Type: Calculation:
80         // reset = 4
81         FCV_RESET = 4;
82
83         break;
84     }
85     case 4:
86     {
87         // Name: Calculation, Type: Calculation:
88         // reset = 5
89         FCV_RESET = 5;
90
91         break;
92     }
93     case 5:
94     {
95         // Name: Calculation, Type: Calculation:
96         // reset = 6
97         FCV_RESET = 6;
98
99         break;
100    }
```

	101	case 6:
	102	{
	103	// Name: Calculation, Type: Calculation:
	104	// reset = 7
	105	FCV_RESET = 7;
	106	
	107	break;
	108	}
	109	case 7:
	110	{
	111	// Name: Calculation, Type: Calculation:
	112	// reset = 8
	113	FCV_RESET = 8;
	114	
	115	break;
	116	}
	117	default:
	118	{
	119	// Name: Calculation, Type: Calculation:
	120	// reset = 9
	121	FCV_RESET = 9;
	122	
	123	}
	124	}
	125	
	126	// Name: Output, Type: Output: luci[clock] -> PORTB
	127	SET_PORT(B, (FCV_LUCI[FCV_CLOCK]));
	128	
	129	// Name: Decision, Type: Decision: clock = reset?
	130	if (FCV_CLOCK == FCV_RESET)
	131	{
	132	
	133	// Name: Goto Connection Point, Type: Goto Connection Point: [A]: A
	134	goto FCC_Main_A;
	135	
	136	// } else {
	137	
	138	}
	139	
	140	
	141	}
	142	
	143	}
	144	

Macro "clock"

```
1 /*-----*\
2   Use :genera l'impulso per l'avanzamento del contatore
3 /*-----*/
4 void FCM_clock()
5 {
6
7   // Name: Calculation, Type: Calculation:
8   // clock = clock + 1
9   FCV_CLOCK = FCV_CLOCK + 1;
10
11 }
12
```

Macro "reset"

```
1 /*-----*\
2   Use :
3 /*-----*/
4 void FCM_RESET()
5 {
6
7   // Name: Calculation, Type: Calculation:
8   // clock = 0
9   FCV_CLOCK = 0;
10
11 }
12
```