

Main

```
1  /*-----*\
2  Use :
3  \*-----*/
4  void FCM_Main()
5  {
6      //Local variable definitions
7      MX_BOOL FCL_X = (0); // numero di situazioni start-stop-reset
8
9
10     // Name: Interrupt, Type: Interrupt: Enable INTO
11     EICRA |= (1 << ISC01) | (1 << ISC00);
12     sei();
13     EIMSK |= (1 << INTO);
14
15     // Name: Interrupt, Type: Interrupt: Enable INT1
16     EICRA |= (1 << ISC11) | (1 << ISC10);
17     sei();
18     EIMSK |= (1 << INT1);
19
20     // Name: Loop, Type: Loop: While 1
21     while (1)
22     {
23
24         // Name: Interrupt, Type: Interrupt: Enable IOC2
25         sei();
26         PCMSK2=0x10;
27         PCICR |= (1 << PCIE2);
28
29         // Name: Call Component Macro, Type: Call Component Macro: led_7seg_quad1::ShowDigit(3, unita, 0)
30         FCD_ofcal_led_7seg_quad1_ShowDigit(3, FCV_UNITA, 0);
31
32         // Name: Call Component Macro, Type: Call Component Macro: led_7seg_quad1::ShowDigit(2, decine, 0)
33         FCD_ofcal_led_7seg_quad1_ShowDigit(2, FCV_DECINE, 0);
34
35         // Name: Call Component Macro, Type: Call Component Macro: led_7seg_quad1::ShowDigit(1, centinaia, 1)
36         FCD_ofcal_led_7seg_quad1_ShowDigit(1, FCV_CENTINAIA, 1);
37
38         // Name: Call Component Macro, Type: Call Component Macro: led_7seg_quad1::ShowDigit(0, migliaia, 0)
39         FCD_ofcal_led_7seg_quad1_ShowDigit(0, FCV_MIGLIAIA, 0);
40
41         // Name: Delay, Type: Delay: 10 ms
42         FCI_DELAYBYTE_MS(10);
43
44     }
45 }
46
47 }
48
```

ingresso

```
1  /*-----*\
2  Use :ad ogni pressione del pulsante unita iva a 1
3  \*-----*/
4  void FCM_ingresso()
5  {
6
7      // Name: Interrupt, Type: Interrupt: Enable TMR1
8      TCCR1B &= 0xf8;
9      TCCR1B |= 0x03;
10     sei();
11     TIMSK1 |= (1 << TOIE1);
12
13 }
14
```

start

```
1 /*-----*\
2   Use :attiva int0
3  \*-----*/
4 void FCM_start()
5 {
6
7   // Name: Interrupt, Type: Interrupt: Enable TMR1
8   TCCR1B &= 0xf8;
9   TCCR1B |= 0x04;
10  sei();
11  TIMSK1 |= (1 << TOIE1);
12
13 }
14
```













stop




```
1 /*-----*\
2   Use :ferma il conteggio
3  \*-----*/
4 void FCM_stop()
5 {
6
7   // Name: Interrupt, Type: Interrupt: Disable TMR1
8   TIMSK1 &= ~(1 << TOIE1);
9
10 }
11
```

reset

```
1 /*-----*\
2   Use :azzera i contatori
3  \*-----*/
4 void FCM_reset()
5 {
6
7   // Name: Calculation, Type: Calculation:
8   // unita = 0
9   // decine = 0
10  // centinaia = 0
11  // migliaia = 0
12  FCV_UNITA = 0;
13  FCV_DECINE = 0;
14  FCV_CENTINAIA = 0;
15  FCV_MIGLIAIA = 0;
16
17 }
18
```

conteggio decimale

	1	/*-----*/
	2	Use :conteggio decimale
	3	/*-----*/
	4	void FCM_conteggio()
	5	{
	6	
	7	// Name: Decision, Type: Decision: unita > 9?
	8	if (FCV_UNITA > 9)
	9	{
	10	
	11	// Name: Calculation, Type: Calculation:
	12	// unita = 0
	13	FCV_UNITA = 0;
	14	
	15	// Name: Decision, Type: Decision: unita = 0?
	16	if (FCV_UNITA == 0)
	17	{
	18	
	19	// Name: Calculation, Type: Calculation:
	20	// decine = decine + 1
	21	FCV_DECINE = FCV_DECINE + 1;
	22	
	23	// Name: Decision, Type: Decision: decine > 9?
	24	if (FCV_DECINE > 9)
	25	{
	26	
	27	// Name: Calculation, Type: Calculation:
	28	// decine = 0
	29	FCV_DECINE = 0;
	30	
	31	// Name: Decision, Type: Decision: decine = 0?
	32	if (FCV_DECINE == 0)
	33	{
	34	
	35	// Name: Calculation, Type: Calculation:
	36	// centinaia = centinaia + 1
	37	FCV_CENTINAIA = FCV_CENTINAIA + 1;
	38	
	39	// Name: Decision, Type: Decision: centinaia > 9?
	40	if (FCV_CENTINAIA > 9)
	41	{
	42	
	43	// Name: Calculation, Type: Calculation:
	44	// centinaia = 0
	45	FCV_CENTINAIA = 0;
	46	
	47	// Name: Decision, Type: Decision: centinaia = 0?
	48	if (FCV_CENTINAIA == 0)
	49	{
	50	

	51 // Name: Calculation, Type: Calculation: 52 // migliaia = migliaia + 1 53 FCV_MIGLIAIA = FCV_MIGLIAIA + 1; 54
	55 // Name: Decision, Type: Decision: migliaia > 9? 56 if (FCV_MIGLIAIA > 9) 57 { 58
	59 // Name: Calculation, Type: Calculation: 60 // migliaia = 0 61 FCV_MIGLIAIA = 0; 62
	63 // } else { 64 } 65 } 66 // } else { 67 } 68 } 69 // } else { 70 } 71 } 72 } 73 } 74 // } else { 75 } 76 } 77 } 78 } 79 // } else { 80 } 81 } 82 } 83 // } else { 84 } 85 } 86 } 87 // } else { 88 } 89 } 90 } 91 } 92 }

conteggio modulo

```
1  /*-----*\
2      Use :
3  \*-----*\
4  void FCM_conteggio_modulo()
5  {
6
7      // Name: Calculation, Type: Calculation:
8      // unita = unita + 1
9      // unita = unita % 10
10     FCV_UNITA = FCV_UNITA + 1;
11     FCV_UNITA = FCV_UNITA % 10;
12
13     // Name: Decision, Type: Decision: unita = 0?
14     if (FCV_UNITA == 0)
15     {
16
17         // Name: Calculation, Type: Calculation:
18         // decine = decine + 1
19         // decine = decine % 6
20         FCV_DECINE = FCV_DECINE + 1;
21         FCV_DECINE = FCV_DECINE % 6;
22
23         // Name: Decision, Type: Decision: decine = 0?
24         if (FCV_DECINE == 0)
25         {
26
27             // Name: Calculation, Type: Calculation:
28             // centinaia = centinaia + 1
29             // centinaia = centinaia % 10
30             FCV_CENTINAIA = FCV_CENTINAIA + 1;
31             FCV_CENTINAIA = FCV_CENTINAIA % 10;
32
33             // Name: Decision, Type: Decision: centinaia = 0?
34             if (FCV_CENTINAIA == 0)
35             {
36
37                 // Name: Calculation, Type: Calculation:
38                 // migliaia = migliaia + 1
39                 // migliaia = migliaia % 10
40                 FCV_MIGLIAIA = FCV_MIGLIAIA + 1;
41                 FCV_MIGLIAIA = FCV_MIGLIAIA % 10;
42
43                 // } else {
44                 }
45
46                 // } else {
47                 }
48
49                 }
50
51         // } else {
52         }
53     }
54 }
55 }
56 }
```

ingresso

```
1  /*-----*\
2      Use :ad ogni pressione del pulsante unita iva a 1
3  \*-----*/
4  void FCM_ingresso()
5  {
6
7      // Name: Interrupt, Type: Interrupt: Enable TMR1
8      TCCR1B &= 0xf8;
9      TCCR1B |= 0x03;
10     sei();
11     TIMSK1 |= (1 << TOIE1);
12
13 }
14
```