```
//*****************************************************************************
//**
//**  Source name:   E:\NOF arduino\flowcode\arduinocounter Quad.fcfx
//**  Title:
//**  Description:
//**  Device:        ARD.ATMEGA.ATMEGA328P
//**
//**  Generated by:  Flowcode v8.1.0.8
//**  Date:          Thursday, January 03, 2019 11:03:24
//**  Users:         1
//**  Registered to: Riccardo Monti
//**  License key: XXXXX
//**
//**
//**  https://www.matrixtsl.com
//**
//*****************************************************************************


#define MX_ARD

#define MX_CAL_ARD

#define MX_CLK_SPEED 16000000

#define FCP_NULL Unconnected_Port

#define MX_UART_ID
#define MX_UART_UCSRC

#include <stdlib.h>
#include <stdio.h>
#include <math.h>
#include <avr\io.h>
#include <avr\interrupt.h>
#include <avr\eeprom.h>
#include <avr\wdt.h>


/*=========================================================================*\
  Use :Include the type definitions
\*=========================================================================*/
#include "C:\ProgramData\MatrixTSL\FlowcodeV8\CAL\internals.c"




/*=========================================================================*\
  Use :panel
     :Variable declarations
     :Macro function declarations
\*=========================================================================*/
```

```
#define FCV_FALSE (0)
#define FCV_TRUE (1)
MX_GLOBAL MX_UINT8 FCV_UNITA = (0x0); // unità
MX_GLOBAL MX_UINT8 FCV_DECINE = (0x0); // decine
MX_GLOBAL MX_UINT8 FCV_CENTINAIA = (0x0); // centinaia
MX_GLOBAL MX_UINT8 FCV_MIGLIAIA = (0x0); // migliaia


/*=========================================================================*\
  Use :lut
     :Variable declarations
     :Macro function declarations
\*=========================================================================*/
#define FCVsz_00fb1_lut__FLOATFIXEDLIST 1
#define FCVsz_00fb1_lut__INTLIST 16
#define FCVsz_00fb1_lut__FLOATLIST 1
#define FCVsz_00fb1_lut__INTFIXEDLIST 1
#define FCD_00fb1_lut__INTLIST(ix) pgm_read_byte(&(FCD_00fb1_lut__INTLIST_LUT[ix]))
ROMARRAY_(MX_UINT8) FCD_00fb1_lut__INTLIST_LUT ROMARRAY_E =
{
// Property added elements
  63, 6, 91, 79, 102, 109, 125, 7, 127, 111, 119, 124, 57, 94, 121, 113
// Dynamically added elements

};


/*=========================================================================*\
  Use :led_7seg_quad1
     :Variable declarations
     :Macro function declarations
\*=========================================================================*/
void FCD_0fca1_led_7seg_quad1__ShowDigit(MX_UINT8 FCL_DIGIT, MX_UINT8 FCL_VALUE,
MX_UINT8 FCL_DECIMALPOINT);

/*=========================================================================*\
  Use :Include the chip adaption layer
\*=========================================================================*/
#include "C:\ProgramData\MatrixTSL\FlowcodeV8\CAL\includes.c"


/*=========================================================================*\
  Use :lut
     :Macro implementations
\*=========================================================================*/

/*=========================================================================*\
  Use :led_7seg_quad1
     :Macro implementations
\*=========================================================================*/
/*=-----------------------------------------------------------------------=*\
  Use :Set the number and decimal point to be displayed in the given digit of the display.
     :
     :Parameters for macro ShowDigit:
```

```
      :  Digit : Which of the four digits to change. (0...3)
      :  Value : The number value to set the digit to (0...15), 16=Clear
      :  DecimalPoint : Whether to show the decimal point for the chosen digit.
\*=----------------------------------------------------------------------------=*/
void FCD_0fca1_led_7seg_quad1__ShowDigit(MX_UINT8 FCL_DIGIT, MX_UINT8 FCL_VALUE,
MX_UINT8 FCL_DECIMALPOINT)
{

   // Common_Pin0 = Display_Type
   // Common_Pin1 = Display_Type
   // Common_Pin2 = Display_Type
   // Common_Pin3 = Display_Type
   SET_PORT_PIN(B, 0, 0);
   SET_PORT_PIN(B, 1, 0);
   SET_PORT_PIN(B, 2, 0);
   SET_PORT_PIN(B, 3, 0);

   //Comment:
   //Configure the segments

   if (FCL_VALUE < 16)
   {

      FCL_VALUE = FCD_00fb1_lut__INTLIST(FCL_VALUE);

      // pin0 = Display_Type == (.Value & 1)
      // .Value = .Value >> 1
      // pin1 = Display_Type == (.Value & 1)
      // .Value = .Value >> 1
      // pin2 = Display_Type == (.Value & 1)
      // .Value = .Value >> 1
      // pin3 = Display_Type == (.Value & 1)
      // .Value = .Value >> 1
      // pin4 = Display_Type == (.Value & 1)
      // .Value = .Value >> 1
      // pin5 = Display_Type == (.Value & 1)
      // .Value = .Value >> 1
      // pin6 = Display_Type == (.Value & 1)
      SET_PORT_PIN(D, 0, 0 == (FCL_VALUE & 1));
      FCL_VALUE = FCL_VALUE >> 1;
      SET_PORT_PIN(D, 1, 0 == (FCL_VALUE & 1));
      FCL_VALUE = FCL_VALUE >> 1;
      SET_PORT_PIN(D, 2, 0 == (FCL_VALUE & 1));
      FCL_VALUE = FCL_VALUE >> 1;
      SET_PORT_PIN(D, 3, 0 == (FCL_VALUE & 1));
      FCL_VALUE = FCL_VALUE >> 1;
      SET_PORT_PIN(D, 4, 0 == (FCL_VALUE & 1));
      FCL_VALUE = FCL_VALUE >> 1;
      SET_PORT_PIN(D, 5, 0 == (FCL_VALUE & 1));
      FCL_VALUE = FCL_VALUE >> 1;
      SET_PORT_PIN(D, 6, 0 == (FCL_VALUE & 1));

      // Pin7 = Display_Type == (.DecimalPoint & 1)
      SET_PORT_PIN(D, 7, 0 == (FCL_DECIMALPOINT & 1));
```

```
} else {

    if (FCL_VALUE == 16)
    {

        // pin0 = Display_Type == 0
        // pin1 = Display_Type == 0
        // pin2 = Display_Type == 0
        // pin3 = Display_Type == 0
        // pin4 = Display_Type == 0
        // pin5 = Display_Type == 0
        // pin6 = Display_Type == 0
        SET_PORT_PIN(D, 0, 0 == 0);
        SET_PORT_PIN(D, 1, 0 == 0);
        SET_PORT_PIN(D, 2, 0 == 0);
        SET_PORT_PIN(D, 3, 0 == 0);
        SET_PORT_PIN(D, 4, 0 == 0);
        SET_PORT_PIN(D, 5, 0 == 0);
        SET_PORT_PIN(D, 6, 0 == 0);

    // } else {

    }

}

switch (FCL_DIGIT)
{
    case 1:
    {
        // Common_Pin1 = 1 - Display_Type
        SET_PORT_PIN(B, 1, 1 - 0);

        break;
    }
    case 2:
    {
        // Common_Pin2 = 1 - Display_Type
        SET_PORT_PIN(B, 2, 1 - 0);

        break;
    }
    case 3:
    {
        // Common_Pin3 = 1 - Display_Type
        SET_PORT_PIN(B, 3, 1 - 0);

        break;
    }
    default:
    {
        // Common_Pin0 = 1 - Display_Type
        SET_PORT_PIN(B, 0, 1 - 0);
```

```
      }
   }

}


/*=====================================================================*\
  Use :panel
     :Macro implementations
\*=====================================================================*/


/*=====================================================================*\
  Use :Main
\*=====================================================================*/
int main()
{
   MCUSR=0x00;


   // Name: Loop, Type: Loop: While 1
   while (1)
   {

      // Name: Call Component Macro, Type: Call Component Macro: led_7seg_quad1::ShowDigit(3, unita, 0)
      FCD_0fca1_led_7seg_quad1__ShowDigit(3, FCV_UNITA, 0);

      // Name: Call Component Macro, Type: Call Component Macro: led_7seg_quad1::ShowDigit(2, decine,
0)
      FCD_0fca1_led_7seg_quad1__ShowDigit(2, FCV_DECINE, 0);

      // Name: Call Component Macro, Type: Call Component Macro: led_7seg_quad1::ShowDigit(1,
centinaia, 0)
      FCD_0fca1_led_7seg_quad1__ShowDigit(1, FCV_CENTINAIA, 0);

      // Name: Call Component Macro, Type: Call Component Macro: led_7seg_quad1::ShowDigit(0, migliaia,
0)
      FCD_0fca1_led_7seg_quad1__ShowDigit(0, FCV_MIGLIAIA, 0);

      // Name: Calculation, Type: Calculation:
      // unita = unita + 1
      FCV_UNITA = FCV_UNITA + 1;

      // Name: Decision, Type: Decision: unita > 9?
      if (FCV_UNITA > 9)
      {

         // Name: Calculation, Type: Calculation:
         // unita = 0
         FCV_UNITA = 0;

         // Name: Decision, Type: Decision: unita = 0?
         if (FCV_UNITA == 0)
```

```
{

    // Name: Calculation, Type: Calculation:
    // decine = decine + 1
    FCV_DECINE = FCV_DECINE + 1;

    // Name: Decision, Type: Decision: decine > 9?
    if (FCV_DECINE > 9)
    {

        // Name: Calculation, Type: Calculation:
        // decine = 0
        FCV_DECINE = 0;

        // Name: Decision, Type: Decision: decine = 0?
        if (FCV_DECINE == 0)
        {

            // Name: Calculation, Type: Calculation:
            // centinaia = centinaia + 1
            FCV_CENTINAIA = FCV_CENTINAIA + 1;

            // Name: Decision, Type: Decision: centinaia > 9?
            if (FCV_CENTINAIA > 9)
            {

                // Name: Calculation, Type: Calculation:
                // centinaia = 0
                FCV_CENTINAIA = 0;

                // Name: Decision, Type: Decision: centinaia = 0?
                if (FCV_CENTINAIA == 0)
                {

                    // Name: Calculation, Type: Calculation:
                    // migliaia = migliaia + 1
                    FCV_MIGLIAIA = FCV_MIGLIAIA + 1;

                    // Name: Decision, Type: Decision: migliaia > 9?
                    if (FCV_MIGLIAIA > 9)
                    {

                        // Name: Calculation, Type: Calculation:
                        // migliaia = 0
                        FCV_MIGLIAIA = 0;

                    // } else {

                    }

                // } else {

                }
```

```
                    // } else {

                    }

                // } else {

                }

            // } else {

            }

        // } else {

        }

    // } else {

    }


  }

  mainendloop: goto mainendloop;
}



/*=====================================================================*\
  Use :Interrupt
\*=====================================================================*/
```