```
//*****************************************************************************
//**
//**  Source name:   C:\Users\io\Desktop\flow test\semaforo a chiamata.fcfx
//**  Title:
//**  Description:
//**  Device:        ARD.ATMEGA.ATMEGA328P
//**
//**  Generated by:  Flowcode v8.1.0.8
//**  Date:          Thursday, January 03, 2019 11:32:37
//**  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"



MX_UINT8 FCLV_LOOP1;
MX_UINT8 FCLV_LOOP2;
MX_UINT8 FCLV_LOOP3;
MX_UINT8 FCLV_LOOP4;
MX_UINT8 FCLV_LOOP5;
MX_UINT8 FCLV_LOOP6;
MX_UINT8 FCLV_LOOP7;
MX_UINT8 FCLV_LOOP8;
```

```
MX_UINT8 FCLV_LOOP9;


/*=====================================================================*\
  Use :panel
     :Variable declarations
     :Macro function declarations
\*=====================================================================*/
#define FCV_FALSE (0)
#define FCV_TRUE (1)
MX_GLOBAL MX_BOOL FCV_CHIAMATA = (0);


/*=====================================================================*\
  Use :ButtonCtrl
     :Variable declarations
     :Macro function declarations
\*=====================================================================*/

/*=====================================================================*\
  Use :switch_base1
     :Variable declarations
     :Macro function declarations
\*=====================================================================*/
MX_UINT8 FCD_05262_switch_base1__ReadState();
void FCD_05262_switch_base1__WaitUntilHigh();
void FCD_05262_switch_base1__WaitUntilLow();

/*=====================================================================*\
  Use :ButtonCtrl
     :Variable declarations
     :Macro function declarations
\*=====================================================================*/

/*=====================================================================*\
  Use :switch_base
     :Variable declarations
     :Macro function declarations
\*=====================================================================*/
MX_UINT8 FCD_05261_switch_base__ReadState();
void FCD_05261_switch_base__WaitUntilHigh();
void FCD_05261_switch_base__WaitUntilLow();

/*=====================================================================*\
  Use :component_label1
     :Variable declarations
     :Macro function declarations
\*=====================================================================*/

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

```
void FCD_03d96_led_base__TurnOn();
void FCD_03d96_led_base__TurnOff();

/*============================================================================*\
  Use :component_label1
      :Variable declarations
      :Macro function declarations
\*============================================================================*/

/*============================================================================*\
  Use :led_base
      :Variable declarations
      :Macro function declarations
\*============================================================================*/
void FCD_03d95_led_base__TurnOn();
void FCD_03d95_led_base__TurnOff();

/*============================================================================*\
  Use :component_label1
      :Variable declarations
      :Macro function declarations
\*============================================================================*/

/*============================================================================*\
  Use :led_base
      :Variable declarations
      :Macro function declarations
\*============================================================================*/
void FCD_03d94_led_base__TurnOn();
void FCD_03d94_led_base__TurnOff();

/*============================================================================*\
  Use :component_label1
      :Variable declarations
      :Macro function declarations
\*============================================================================*/

/*============================================================================*\
  Use :led_base
      :Variable declarations
      :Macro function declarations
\*============================================================================*/
void FCD_03d93_led_base__TurnOn();
void FCD_03d93_led_base__TurnOff();

/*============================================================================*\
  Use :component_label1
      :Variable declarations
      :Macro function declarations
\*============================================================================*/

/*============================================================================*\
  Use :led_base
      :Variable declarations
```

```
      :Macro function declarations
\*=========================================================================*/
void FCD_03d92_led_base__TurnOn();
void FCD_03d92_led_base__TurnOff();

/*=========================================================================*\
  Use :component_label1
      :Variable declarations
      :Macro function declarations
\*=========================================================================*/

/*=========================================================================*\
  Use :led_base
      :Variable declarations
      :Macro function declarations
\*=========================================================================*/
void FCD_03d91_led_base__TurnOn();
void FCD_03d91_led_base__TurnOff();

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


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

/*=========================================================================*\
  Use :switch_base1
      :Macro implementations
\*=========================================================================*/
/*=-----------------------------------------------------------------------=*\
  Use :Reads the button state as 0 for released or 1 for pressed
      :Performs debounce if required
      :
      :Returns : MX_UINT8
\*=-----------------------------------------------------------------------=*/
MX_UINT8 FCD_05262_switch_base1__ReadState()
{
   //Local variable definitions
   MX_UINT16 FCL_DEL_COUNT;
   MX_UINT8 FCL_OLD_SWITCHVAL;
   MX_UINT8 FCR_RETVAL;


   #if (1)

      if (0 != GET_PORT_PIN(D, 2))
      {

         // .Return = 1
```

```
            FCR_RETVAL = 1;

        } else {

            // .Return = 0
            FCR_RETVAL = 0;

        }

        // .del_count = 0
        // .old_switchval = .Return
        FCL_DEL_COUNT = 0;
        FCL_OLD_SWITCHVAL = FCR_RETVAL;

        #if (0) // 0 > 0

        //Code has been optimised out by the pre-processor
        // #else

        #endif

    #else

    //Code has been optimised out by the pre-processor
    #endif

    return (FCR_RETVAL);

}

/*=-----------------------------------------------------------------------=*\
   Use :Waits until the switch is in state 'high'
       :The interpretation of 'high' depends on the polarity
\*=-----------------------------------------------------------------------=*/
void FCD_05262_switch_base1__WaitUntilHigh()
{
    //Local variable definitions
    MX_UINT8 FCL_SWITCHVAL = (0xff); // The state of the pin
    MX_UINT16 FCL_DEL_COUNT;
    MX_UINT8 FCL_OLD_SWITCHVAL;


    #if (1)

        while (1)
        {

            // .switchval = pin
            FCL_SWITCHVAL = GET_PORT_PIN(D, 2);

            #if (0) // 0 == 1

            //Code has been optimised out by the pre-processor
            // #else
```

```
            #endif

            // .del_count = 0
            // .old_switchval = .switchval
            FCL_DEL_COUNT = 0;
            FCL_OLD_SWITCHVAL = FCL_SWITCHVAL;

            #if (0) // 0 > 0

            //Code has been optimised out by the pre-processor
            // #else

            #endif


            if ((FCL_SWITCHVAL == 0) == 0) break;
        }

    // #else

    //Code has been optimised out by the pre-processor
    #endif

}

/*=-------------------------------------------------------------------------=*\
  Use :Waits until the switch is in state 'low'
      :The interpretation of 'high' depends on the polarity
\*=-------------------------------------------------------------------------=*/
void FCD_05262_switch_base1__WaitUntilLow()
{
    //Local variable definitions
    MX_UINT8 FCL_SWITCHVAL; // The state of the pin
    MX_UINT16 FCL_DEL_COUNT;
    MX_UINT8 FCL_OLD_SWITCHVAL;


    #if (1)

        while (1)
        {

            // .switchval = pin
            FCL_SWITCHVAL = GET_PORT_PIN(D, 2);

            #if (0) // 0 == 1

            //Code has been optimised out by the pre-processor
            // #else

            #endif

            // .del_count = 0
```

```
            // .old_switchval = .switchval
            FCL_DEL_COUNT = 0;
            FCL_OLD_SWITCHVAL = FCL_SWITCHVAL;

            #if (0) // 0 > 0

            //Code has been optimised out by the pre-processor
            // #else

            #endif


            if ((FCL_SWITCHVAL != 0) == 0) break;
        }

    // #else

    //Code has been optimised out by the pre-processor
    #endif

}


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

/*============================================================================*\
   Use :switch_base
       :Macro implementations
\*============================================================================*/
/*=---------------------------------------------------------------------------=*\
   Use :Reads the button state as 0 for released or 1 for pressed
       :Performs debounce if required
       :
       :Returns : MX_UINT8
\*=---------------------------------------------------------------------------=*/
MX_UINT8 FCD_05261_switch_base__ReadState()
{
    //Local variable definitions
    MX_UINT16 FCL_DEL_COUNT;
    MX_UINT8 FCL_OLD_SWITCHVAL;
    MX_UINT8 FCR_RETVAL;


    #if (1)

        if (0 != GET_PORT_PIN(D, 2))
        {

            // .Return = 1
            FCR_RETVAL = 1;
```

```
} else {

    // .Return = 0
    FCR_RETVAL = 0;

}

// .del_count = 0
// .old_switchval = .Return
FCL_DEL_COUNT = 0;
FCL_OLD_SWITCHVAL = FCR_RETVAL;

#if (1) // 10 > 0

    while (FCL_DEL_COUNT < 10)
    {

        FCI_DELAYBYTE_MS(1);

        if (0 != GET_PORT_PIN(D, 2))
        {

            // .Return = 1
            FCR_RETVAL = 1;

        } else {

            // .Return = 0
            FCR_RETVAL = 0;

        }

        if (FCR_RETVAL == FCL_OLD_SWITCHVAL)
        {

            // .del_count = .del_count + 1
            FCL_DEL_COUNT = FCL_DEL_COUNT + 1;

        } else {

            // .del_count = 0
            FCL_DEL_COUNT = 0;

        }

        // .old_switchval = .Return
        FCL_OLD_SWITCHVAL = FCR_RETVAL;


    }

// #else

//Code has been optimised out by the pre-processor
```

```
        #endif

    #else

    //Code has been optimised out by the pre-processor
    #endif

    return (FCR_RETVAL);

}

/*=-----------------------------------------------------------------------=*\
   Use :Waits until the switch is in state 'high'
       :The interpretation of 'high' depends on the polarity
\*=-----------------------------------------------------------------------=*/
void FCD_05261_switch_base__WaitUntilHigh()
{
    //Local variable definitions
    MX_UINT8 FCL_SWITCHVAL = (0xff); // The state of the pin
    MX_UINT16 FCL_DEL_COUNT;
    MX_UINT8 FCL_OLD_SWITCHVAL;


    #if (1)

        while (1)
        {

            // .switchval = pin
            FCL_SWITCHVAL = GET_PORT_PIN(D, 2);

            #if (0) // 0 == 1

            //Code has been optimised out by the pre-processor
            // #else

            #endif

            // .del_count = 0
            // .old_switchval = .switchval
            FCL_DEL_COUNT = 0;
            FCL_OLD_SWITCHVAL = FCL_SWITCHVAL;

            #if (1) // 10 > 0

                while (FCL_DEL_COUNT < 10)
                {

                    FCI_DELAYBYTE_MS(1);

                    // .switchval = pin
                    FCL_SWITCHVAL = GET_PORT_PIN(D, 2);

                    #if (0) // 0 == 1
```

```
            //Code has been optimised out by the pre-processor
            // #else

            #endif

            if (FCL_SWITCHVAL == FCL_OLD_SWITCHVAL)
            {

               // .del_count = .del_count + 1
               FCL_DEL_COUNT = FCL_DEL_COUNT + 1;

            } else {

               // .del_count = 0
               FCL_DEL_COUNT = 0;

            }

            // .old_switchval = .switchval
            FCL_OLD_SWITCHVAL = FCL_SWITCHVAL;


         }

      // #else

      //Code has been optimised out by the pre-processor
      #endif


      if ((FCL_SWITCHVAL == 0) == 0) break;
   }

   // #else

   //Code has been optimised out by the pre-processor
   #endif

}

/*=-----------------------------------------------------------------------=*\
  Use :Waits until the switch is in state 'low'
      :The interpretation of 'high' depends on the polarity
\*=-----------------------------------------------------------------------=*/
void FCD_05261_switch_base__WaitUntilLow()
{
   //Local variable definitions
   MX_UINT8 FCL_SWITCHVAL; // The state of the pin
   MX_UINT16 FCL_DEL_COUNT;
   MX_UINT8 FCL_OLD_SWITCHVAL;


   #if (1)
```

```
while (1)
{

    // .switchval = pin
    FCL_SWITCHVAL = GET_PORT_PIN(D, 2);

    #if (0) // 0 == 1

    //Code has been optimised out by the pre-processor
    // #else

    #endif

    // .del_count = 0
    // .old_switchval = .switchval
    FCL_DEL_COUNT = 0;
    FCL_OLD_SWITCHVAL = FCL_SWITCHVAL;

    #if (1) // 10 > 0

        while (FCL_DEL_COUNT < 10)
        {

            FCI_DELAYBYTE_MS(1);

            // .switchval = pin
            FCL_SWITCHVAL = GET_PORT_PIN(D, 2);

            #if (0) // 0 == 1

            //Code has been optimised out by the pre-processor
            // #else

            #endif

            if (FCL_SWITCHVAL == FCL_OLD_SWITCHVAL)
            {

                // .del_count = .del_count + 1
                FCL_DEL_COUNT = FCL_DEL_COUNT + 1;

            } else {

                // .del_count = 0
                FCL_DEL_COUNT = 0;

            }

            // .old_switchval = .switchval
            FCL_OLD_SWITCHVAL = FCL_SWITCHVAL;


        }
```

```
        // #else

        //Code has been optimised out by the pre-processor
        #endif


        if ((FCL_SWITCHVAL != 0) == 0) break;
      }

   // #else

   //Code has been optimised out by the pre-processor
   #endif

}


/*=========================================================================*\
  Use :component_label1
     :Macro implementations
\*=========================================================================*/

/*=========================================================================*\
  Use :led_base
     :Macro implementations
\*=========================================================================*/
/*=-----------------------------------------------------------------------=*\
  Use :Turn the LED off.
\*=-----------------------------------------------------------------------=*/
void FCD_03d96_led_base__TurnOn()
{

   #if (1)

      // pin = polarity
      SET_PORT_PIN(B, 4, 1);

   // #else

   //Code has been optimised out by the pre-processor
   #endif

}

/*=-----------------------------------------------------------------------=*\
  Use :Turn the LED on.
\*=-----------------------------------------------------------------------=*/
void FCD_03d96_led_base__TurnOff()
{

   #if (1)

      // pin = 1 - polarity
```

```
        SET_PORT_PIN(B, 4, 1 - 1);

    // #else

    //Code has been optimised out by the pre-processor
    #endif

}


/*=====================================================================*\
  Use :component_label1
     :Macro implementations
\*=====================================================================*/

/*=====================================================================*\
  Use :led_base
     :Macro implementations
\*=====================================================================*/
/*=---------------------------------------------------------------=*\
  Use :Turn the LED off.
\*=---------------------------------------------------------------=*/
void FCD_03d95_led_base__TurnOn()
{

    #if (1)

        // pin = polarity
        SET_PORT_PIN(B, 3, 1);

    // #else

    //Code has been optimised out by the pre-processor
    #endif

}

/*=---------------------------------------------------------------=*\
  Use :Turn the LED on.
\*=---------------------------------------------------------------=*/
void FCD_03d95_led_base__TurnOff()
{

    #if (1)

        // pin = 1 - polarity
        SET_PORT_PIN(B, 3, 1 - 1);

    // #else

    //Code has been optimised out by the pre-processor
    #endif

}
```

```
/*=========================================================================*\
  Use :component_label1
     :Macro implementations
\*=========================================================================*/

/*=========================================================================*\
  Use :led_base
     :Macro implementations
\*=========================================================================*/
/*=-----------------------------------------------------------------=*\
  Use :Turn the LED off.
\*=-----------------------------------------------------------------=*/
void FCD_03d94_led_base__TurnOn()
{

   #if (1)

      // pin = polarity
      SET_PORT_PIN(B, 5, 1);

   // #else

   //Code has been optimised out by the pre-processor
   #endif

}

/*=-----------------------------------------------------------------=*\
  Use :Turn the LED on.
\*=-----------------------------------------------------------------=*/
void FCD_03d94_led_base__TurnOff()
{

   #if (1)

      // pin = 1 - polarity
      SET_PORT_PIN(B, 5, 1 - 1);

   // #else

   //Code has been optimised out by the pre-processor
   #endif

}


/*=========================================================================*\
  Use :component_label1
     :Macro implementations
\*=========================================================================*/

/*=========================================================================*\
```

```
  Use :led_base
     :Macro implementations
\*=========================================================================*/
/*=-----------------------------------------------------------------=*\
  Use :Turn the LED off.
\*=-----------------------------------------------------------------=*/
void FCD_03d93_led_base__TurnOn()
{

   #if (1)

      // pin = polarity
      SET_PORT_PIN(B, 0, 1);

   // #else

   //Code has been optimised out by the pre-processor
   #endif

}

/*=-----------------------------------------------------------------=*\
  Use :Turn the LED on.
\*=-----------------------------------------------------------------=*/
void FCD_03d93_led_base__TurnOff()
{

   #if (1)

      // pin = 1 - polarity
      SET_PORT_PIN(B, 0, 1 - 1);

   // #else

   //Code has been optimised out by the pre-processor
   #endif

}


/*=========================================================================*\
  Use :component_label1
     :Macro implementations
\*=========================================================================*/

/*=========================================================================*\
  Use :led_base
     :Macro implementations
\*=========================================================================*/
/*=-----------------------------------------------------------------=*\
  Use :Turn the LED off.
\*=-----------------------------------------------------------------=*/
void FCD_03d92_led_base__TurnOn()
{
```

```
    #if (1)

        // pin = polarity
        SET_PORT_PIN(B, 2, 1);

    // #else

    //Code has been optimised out by the pre-processor
    #endif

}

/*=---------------------------------------------------------------------=*\
  Use :Turn the LED on.
\*=---------------------------------------------------------------------=*/
void FCD_03d92_led_base__TurnOff()
{

    #if (1)

        // pin = 1 - polarity
        SET_PORT_PIN(B, 2, 1 - 1);

    // #else

    //Code has been optimised out by the pre-processor
    #endif

}


/*=====================================================================*\
  Use :component_label1
      :Macro implementations
\*=====================================================================*/

/*=====================================================================*\
  Use :led_base
      :Macro implementations
\*=====================================================================*/
/*=---------------------------------------------------------------------=*\
  Use :Turn the LED off.
\*=---------------------------------------------------------------------=*/
void FCD_03d91_led_base__TurnOn()
{

    #if (1)

        // pin = polarity
        SET_PORT_PIN(B, 1, 1);

    // #else
```

```
    //Code has been optimised out by the pre-processor
    #endif

}

/*=-----------------------------------------------------------------------=*\
  Use :Turn the LED on.
\*=-----------------------------------------------------------------------=*/
void FCD_03d91_led_base__TurnOff()
{

    #if (1)

        // pin = 1 - polarity
        SET_PORT_PIN(B, 1, 1 - 1);

    // #else

    //Code has been optimised out by the pre-processor
    #endif

}


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


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


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

        // Name: Input, Type: Input: D2 -> CHIAMATA
        FCV_CHIAMATA = GET_PORT_PIN(D,2);

        // Name: Delay, Type: Delay: 200 ms
        FCI_DELAYBYTE_MS(200);

        // Name: Decision, Type: Decision: CHIAMATA?
        if (FCV_CHIAMATA)
        {

            // Name: Loop, Type: Loop: Loop  5 times
            for (FCLV_LOOP1=0; (FCLV_LOOP1)<(5); (FCLV_LOOP1)++)
```

```
{

   // Name: ciclo ROSSO, Type: Loop: Loop  2 times
   for (FCLV_LOOP2=0; (FCLV_LOOP2)<(2); (FCLV_LOOP2)++)
   {

      // Name: rosso 1, Type: Output: 1 -> B2
      SET_PORT_PIN(B,2,(1));

      // Name: rosso 2, Type: Output: 1 -> B5
      SET_PORT_PIN(B,5,(1));

      // Name: verde 2, Type: Output: 0 -> B3
      SET_PORT_PIN(B,3,(0));

      // Name: verde 1, Type: Output: 0 -> B0
      SET_PORT_PIN(B,0,(0));

      // Name: Delay, Type: Delay: 1 s
      FCI_DELAYBYTE_S(1);


   }

   // Name: Loop, Type: Loop: Loop  5 times
   for (FCLV_LOOP3=0; (FCLV_LOOP3)<(5); (FCLV_LOOP3)++)
   {

      //Comment:
      //CICLO SEMAFORO verde 1

      // Name: verde 1, Type: Output: 1 -> B0
      SET_PORT_PIN(B,0,(1));

      // Name: arancio 1, Type: Output: 0 -> B1
      SET_PORT_PIN(B,1,(0));

      // Name: rosso 1, Type: Output: 0 -> B2
      SET_PORT_PIN(B,2,(0));

      // Name: verde 2, Type: Output: 0 -> B3
      SET_PORT_PIN(B,3,(0));

      // Name: arancio 2, Type: Output: 0 -> B4
      SET_PORT_PIN(B,4,(0));

      // Name: rosso 2, Type: Output: 1 -> B5
      SET_PORT_PIN(B,5,(1));

      // Name: Delay, Type: Delay: 4 s
      FCI_DELAYBYTE_S(4);

      // Name: VERDE - GIALLO 1, Type: Loop: Loop  3 times
      for (FCLV_LOOP4=0; (FCLV_LOOP4)<(3); (FCLV_LOOP4)++)
```

```
{

   // Name: arancio 1, Type: Output: 1 -> B1
   SET_PORT_PIN(B,1,(1));

   // Name: Delay, Type: Delay: 1 s
   FCI_DELAYBYTE_S(1);


}

// Name: VERDE - GIALLO 1 -  LAMPEGGIANTE, Type: Loop: Loop  3 times
for (FCLV_LOOP5=0; (FCLV_LOOP5)<(3); (FCLV_LOOP5)++)
{

   //Comment:
   //LAMPEGGIO ARANCIO 1

   // Name: arancio 1, Type: Output: 1 -> B1
   SET_PORT_PIN(B,1,(1));

   // Name: Delay, Type: Delay: 500 ms
   FCI_DELAYINT_MS(500);

   // Name: arancio 1, Type: Output: 0 -> B1
   SET_PORT_PIN(B,1,(0));

   // Name: Delay, Type: Delay: 500 ms
   FCI_DELAYINT_MS(500);


}

// Name: LoopCiclo ROSSO, Type: Loop: Loop  2 times
for (FCLV_LOOP6=0; (FCLV_LOOP6)<(2); (FCLV_LOOP6)++)
{

   // Name: rosso 1, Type: Output: 1 -> B2
   SET_PORT_PIN(B,2,(1));

   // Name: rosso 2, Type: Output: 1 -> B5
   SET_PORT_PIN(B,5,(1));

   // Name: verde 1, Type: Output: 0 -> B0
   SET_PORT_PIN(B,0,(0));

   // Name: verde 2, Type: Output: 0 -> B3
   SET_PORT_PIN(B,3,(0));

   // Name: Delay, Type: Delay: 1 s
   FCI_DELAYBYTE_S(1);


}
```

```
// Name: Loop, Type: Loop: Loop  5 times
for (FCLV_LOOP7=0; (FCLV_LOOP7)<(5); (FCLV_LOOP7)++)
{

   //Comment:
   //CICLO SEMAFORO verde 2

   // Name: verde 1, Type: Output: 0 -> B0
   SET_PORT_PIN(B,0,(0));

   // Name: arancio 1, Type: Output: 0 -> B1
   SET_PORT_PIN(B,1,(0));

   // Name: rosso 1, Type: Output: 1 -> B2
   SET_PORT_PIN(B,2,(1));

   // Name: verde 2, Type: Output: 1 -> B3
   SET_PORT_PIN(B,3,(1));

   // Name: arancio 2, Type: Output: 0 -> B4
   SET_PORT_PIN(B,4,(0));

   // Name: rosso 2, Type: Output: 0 -> B5
   SET_PORT_PIN(B,5,(0));

   // Name: Delay, Type: Delay: 4 s
   FCI_DELAYBYTE_S(4);

   // Name: VERDE - GIALLO  2, Type: Loop: Loop  3 times
   for (FCLV_LOOP8=0; (FCLV_LOOP8)<(3); (FCLV_LOOP8)++)
   {

      // Name: arancio 1, Type: Output: 1 -> B4
      SET_PORT_PIN(B,4,(1));

      // Name: Delay, Type: Delay: 1 s
      FCI_DELAYBYTE_S(1);


   }

   // Name: VERDE - GIALLO  2 - LAMPEGGIANTE, Type: Loop: Loop  3 times
   for (FCLV_LOOP9=0; (FCLV_LOOP9)<(3); (FCLV_LOOP9)++)
   {

      //Comment:
      //LAMPEGGIO ARANCIO 2

      // Name: arancio 2, Type: Output: 1 -> B4
      SET_PORT_PIN(B,4,(1));

      // Name: Delay, Type: Delay: 500 ms
      FCI_DELAYINT_MS(500);
```

```
            // Name: arancio 2, Type: Output: 0 -> B4
            SET_PORT_PIN(B,4,(0));

            // Name: Delay, Type: Delay: 500 ms
            FCI_DELAYINT_MS(500);


          }

          // Name: Goto CICLO SEMAFORO 1, Type: Goto Connection Point: [A]: A
          goto FCC_Main_A;


        }


      }


   }

   // Name: CICLO SEMAFORO 1, Type: Connection Point: [A]: A
FCC_Main_A:
   ;

} else {

   //Comment:
   //LAMPEGGIO ARANCIO

   // Name: verde 1, Type: Output: 0 -> B0
   SET_PORT_PIN(B,0,(0));

   // Name: arancio 1, Type: Output: 1 -> B1
   SET_PORT_PIN(B,1,(1));

   // Name: rosso 1, Type: Output: 0 -> B2
   SET_PORT_PIN(B,2,(0));

   // Name: verde 2, Type: Output: 0 -> B3
   SET_PORT_PIN(B,3,(0));

   // Name: arancio 2, Type: Output: 1 -> B4
   SET_PORT_PIN(B,4,(1));

   // Name: rosso 2, Type: Output: 0 -> B5
   SET_PORT_PIN(B,5,(0));

   // Name: Delay, Type: Delay: 1 s
   FCI_DELAYBYTE_S(1);

   // Name: arancio 1, Type: Output: 0 -> B1
   SET_PORT_PIN(B,1,(0));
```

```
            // Name: arancio 2, Type: Output: 0 -> B4
            SET_PORT_PIN(B,4,(0));

            // Name: Delay, Type: Delay: 1 s
            FCI_DELAYBYTE_S(1);

        }


    }

    mainendloop: goto mainendloop;
}



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