

	<pre> /*-----*/ Use :Example program showing how to use a quadrature encoder component. /*-----*/ void FCM_Main() { </pre>
	<pre> // Name: Call Component Macro, Type: Call Component Macro: gLCD_ILI9341_SPI1::Initialise() FCD_0a2b1_gLCD_ILI9341_SPI1_Initialise(); </pre>
	<pre> // Name: Call Component Macro, Type: Call Component Macro: QuadEncoder2::Enable() FCD_017b1_QuadEncoder2_Enable(); </pre>
	<pre> // Name: Call Component Macro, Type: Call Component Macro: QuadEncoder2::ResetCounter() FCD_017b1_QuadEncoder2_ResetCounter(); </pre>
	<pre> // Name: Calculation, Type: Calculation: // \$PORTD.7 = 0 // COD1 = 0 // COD2 = 0 // Count = 0 SET_PORT_PIN(D, 7, 0); FCV_COD1 = 0; FCV_COD2 = 0; FCV_COUNT = 0; </pre>
	<pre> // Name: Calculation, Type: Calculation: // Codice1 = 15 // Codice2 = -11 FCV_CODICE1 = 15; FCV_CODICE2 = -11; </pre>
	<pre> // Name: Loop, Type: Loop: While 1 while (1) { </pre>
	<pre> // Name: Call Component Macro, Type: Call Component Macro: changed=QuadEncoder2::CheckForChanges() FCV_CHANGED = FCD_017b1_QuadEncoder2_CheckForChanges(); </pre>
	<pre> // Name: pulsante push : acquisisce, Type: Input: D6 -> SwA FCV_SWA = GET_PORT_PIN(D,6); </pre>
	<pre> // Name: pulsante premuto, Type: Decision: SwA = 1? if (FCV_SWA == 1) { </pre>
	<pre> // Name: Delay, Type: Delay: 100 ms FCI_DELAYBYTE_MS(100); </pre>
	<pre> // Name: accende i led in sequenza ogni volta che si preme SwA, Type: Switch: CountSwA? switch (FCV_COUNTSWA) { </pre>
	<pre> case 1: { </pre>
	<pre> // Name: Call Component Macro, Type: Call Component Macro: Count=QuadEncoder2::ReadCounter() FCV_COUNT = FCD_017b1_QuadEncoder2_ReadCounter(); </pre>
	<pre> // Name: BLUE, Type: Call Component Macro: gLCD_ILI9341_SPI1::SetForegroundColour(0, 0, 255) FCD_0a2b1_gLCD_ILI9341_SPI1_SetForegroundColour(0, 0, 255); </pre>
	<pre> // Name: NUMEROP INSERITO, Type: Call Component Macro: gLCD_ILI9341_SPI1::PrintNumber(Count, 100, 150, 12, 0) FCD_0a2b1_gLCD_ILI9341_SPI1_PrintNumber(FCV_COUNT, 100, 150, 12, 0); </pre>
	<pre> // Name: Decision, Type: Decision: Count = Codice1? if (FCV_COUNT == FCV_CODICE1) </pre>

```
// Name: Decision, Type: Deci
if (FCV COUNT == FCV CODICE2)
```

	<pre>// Name: VERDE, Type: Call Component Macro: gLCD_ILI9341_SPI1::SetForegroundColour(0, 255, 0) FCD_0a2b1_gLCD_ILI9341_SPI1_SetForegroundColour(0, 255, 0);</pre>
	<pre>// Name: Call Component Macro, Type: Call Component Macro: gLCD_ILI9341_SPI1::DrawCircle(80, 80, 10, 1, 1) FCD_0a2b1_gLCD_ILI9341_SPI1_DrawCircle(80, 80, 10, 1, 1);</pre>
	<pre>// Name: VERDE, Type: Call Component Macro: gLCD_ILI9341_SPI1::SetForegroundColour(0, 255, 0) FCD_0a2b1_gLCD_ILI9341_SPI1_SetForegroundColour(0, 255, 0);</pre>
	<pre>// Name: Call Component Macro, Type: Call Component Macro: gLCD_ILI9341_SPI1::DrawCircle(160, 80, 10, 1, 1) FCD_0a2b1_gLCD_ILI9341_SPI1_DrawCircle(160, 80, 10, 1, 1);</pre>
	<pre>// Name: Calculation, Type: Calculation: // COD2 = Count FCV_COD2 = FCV_COUNT;</pre>
	<pre>// Name: Delay, Type: Delay: 500 ms FCI_DELAYINT_MS(500);</pre>
	<pre>} else {</pre>
	<pre>// Name: ROSSO, Type: Call Component Macro: gLCD_ILI9341_SPI1::SetForegroundColour(255, 0, 0) FCD_0a2b1_gLCD_ILI9341_SPI1_SetForegroundColour(255, 0, 0);</pre>
	<pre>// Name: Call Component Macro, Type: Call Component Macro: gLCD_ILI9341_SPI1::DrawCircle(160, 80, 10, 1, 1) FCD_0a2b1_gLCD_ILI9341_SPI1_DrawCircle(160, 80, 10, 1, 1);</pre>
	<pre>// Name: Calculation, Type: Calculation: // COD2 = 0 FCV_COD2 = 0;</pre>
	<pre>// Name: Delay, Type: Delay: 500 ms FCI_DELAYINT_MS(500);</pre>
	<pre>}</pre>
	<pre>} else {</pre>
★	<pre>// Name: Decision, Type: Decision: Count = Codice1? if (FCV_COUNT == FCV_CODICE1) {</pre>
	<pre>// Name: VERDE, Type: Call Component Macro: gLCD_ILI9341_SPI1::SetForegroundColour(0, 255, 0) FCD_0a2b1_gLCD_ILI9341_SPI1_SetForegroundColour(0, 255, 0);</pre>
	<pre>// Name: Call Component Macro, Type: Call Component Macro: gLCD_ILI9341_SPI1::DrawCircle(160, 80, 10, 1, 1) FCD_0a2b1_gLCD_ILI9341_SPI1_DrawCircle(160, 80, 10, 1, 1);</pre>
	<pre>// Name: Calculation, Type: Calculation: // COD1 = Count FCV_COD1 = FCV_COUNT;</pre>
	<pre>// Name: Delay, Type: Delay: 500 ms FCI_DELAYINT_MS(500);</pre>
★	<pre>// Name: Decision, Type: Decision: Count = Codice2? if (FCV_COUNT == FCV_CODICE2) {</pre>

	<pre>// Name: VERDE, Type: Call Component Macro: gLCD_ILI9341_SPI1::SetForegroundColour(0, 255, 0) FCD_0a2b1_gLCD_ILI9341_SPI1_SetForegroundColour(0, 255, 0);</pre>
	<pre>// Name: Call Component Macro, Type: Call Component Macro: gLCD_ILI9341_SPI1::DrawCircle(80, 80, 10, 1, 1) FCD_0a2b1_gLCD_ILI9341_SPI1_DrawCircle(80, 80, 10, 1, 1);</pre>
	<pre>// Name: VERDE, Type: Call Component Macro: gLCD_ILI9341_SPI1::SetForegroundColour(0, 255, 0) FCD_0a2b1_gLCD_ILI9341_SPI1_SetForegroundColour(0, 255, 0);</pre>
	<pre>// Name: Call Component Macro, Type: Call Component Macro: gLCD_ILI9341_SPI1::DrawCircle(160, 80, 10, 1, 1) FCD_0a2b1_gLCD_ILI9341_SPI1_DrawCircle(160, 80, 10, 1, 1);</pre>
	<pre>// Name: Calculation, Type: Calculation: // COD2 = Count FCV_COD2 = FCV_COUNT;</pre>
	<pre>// Name: Delay, Type: Delay: 500 ms FCI_DELAYINT_MS(500);</pre>
	<pre>// } else { } // } else { } } break; } case 3: {</pre>
	<pre>// Name: Call Component Macro, Type: Call Component Macro: Count=QuadEncoder2::ReadCounter() FCV_COUNT = FCD_017b1_QuadEncoder2_ReadCounter();</pre>
	<pre>// Name: BLUE, Type: Call Component Macro: gLCD_ILI9341_SPI1::SetForegroundColour(0, 0, 255) FCD_0a2b1_gLCD_ILI9341_SPI1_SetForegroundColour(0, 0, 255);</pre>
	<pre>// Name: NUMEROP INSERITO, Type: Call Component Macro: gLCD_ILI9341_SPI1::PrintNumber(Count, 100, 150, 12, 0) FCD_0a2b1_gLCD_ILI9341_SPI1_PrintNumber(FCV_COUNT, 100, 150, 12, 0);</pre>
◆	<pre>// Name: Decision, Type: Decision: COD1? if (FCV_COD1) {</pre>
★	<pre>// Name: Decision, Type: Decision: Count = Codice2? if (FCV_COUNT == FCV_CODICE2) {</pre>
	<pre>// Name: VERDE, Type: Call Component Macro: gLCD_ILI9341_SPI1::SetForegroundColour(0, 255, 0) FCD_0a2b1_gLCD_ILI9341_SPI1_SetForegroundColour(0, 255, 0);</pre>

■	// Name: Call Component Macro, Type: Call Component Macro: gLCD_ILI9341_SPI1::DrawCircle(80, 80, 10, 1, 1) FCD_Oa2b1_gLCD_ILI9341_SPI1_DrawCircle(80, 80, 10, 1, 1);
■	// Name: VERDE, Type: Call Component Macro: gLCD_ILI9341_SPI1::SetForegroundColour(0, 255, 0) FCD_Oa2b1_gLCD_ILI9341_SPI1_SetForegroundColour(0, 255, 0);
■	// Name: Call Component Macro, Type: Call Component Macro: gLCD_ILI9341_SPI1::DrawCircle(160, 80, 10, 1, 1) FCD_Oa2b1_gLCD_ILI9341_SPI1_DrawCircle(160, 80, 10, 1, 1);
■	// Name: Calculation, Type: Calculation: // COD2 = Count FCV_COD2 = FCV_COUNT;
⌚	// Name: Delay, Type: Delay: 500 ms FCI_DELAYINT_MS(500);
	} else {
■	// Name: Call Macro, Type: Call Macro: ALLARME() FCM_ALLARME();
	}
	} else {
■	// Name: Call Macro, Type: Call Macro: ALLARME() FCM_ALLARME();
	}
	break;
	} // default:
	}
◆	// Name: codice corretto, Type: Decision: COD1 = Codice1 AND COD2 = Codice2? if (FCV_COD1 == FCV_CODICE1 & FCV_COD2 == FCV_CODICE2) {
■	// Name: Call Macro, Type: Call Macro: APERTO() FCM_APERTO();
	} else {
■	// Name: Call Component Macro, Type: Call Component Macro: gLCD_ILI9341_SPI1::ClearDisplay() FCD_Oa2b1_gLCD_ILI9341_SPI1_ClearDisplay();
■	// Name: Call Component Macro, Type: Call Component Macro: Count=QuadEncoder2::ReadCounter() FCV_COUNT = FCD_017b1_QuadEncoder2_ReadCounter();
■	// Name: Call Component Macro, Type: Call Component Macro: gLCD_ILI9341_SPI1::PrintNumber(Count, 100, 10, 12, 0) FCD_Oa2b1_gLCD_ILI9341_SPI1_PrintNumber(FCV_COUNT, 100, 10, 12, 0);
⌚	// Name: elimina flickering, Type: Delay: 100 ms FCI_DELAYBYTE_MS(100);
	}
◆	// Name: Ritardo, Type: Decision: Delay > 2000? if (FCV_DELAY > 2000) {
■	// Name: Calculation, Type: Calculation: // Delay = 0 FCV_DELAY = 0;
	} else {
■	// Name: , Type: Calculation: // Delay = Delay + 1 FCV_DELAY = FCV_DELAY + 1;
	}
■	}
	}


```

/*=====*\
Use :
\*=====*/
void FCM_ALLARME()
{

```

```

// Name: terzo tentativo errato : ALLARME, Type: Loop: While 1
while (1)
{

```

```

// Name: Call Component Macro, Type: Call Component Macro: gLCD_ILI9341_SPI1::SetForegroundColour(255, 0, 0)
FCD_0a2b1_gLCD_ILI9341_SPI1_SetForegroundColour(255, 0, 0);

```

```

// Name: Call Component Macro, Type: Call Component Macro: gLCD_ILI9341_SPI1::Print("ALLARME", 100, 150, 12, 0)
FCD_0a2b1_gLCD_ILI9341_SPI1_Print("ALLARME", 8, 100, 150, 12, 0);

```

```

// Name: Call Component Macro, Type: Call Component Macro: gLCD_ILI9341_SPI1::DrawCircle(80, 80, 10, 0, 1)
FCD_0a2b1_gLCD_ILI9341_SPI1_DrawCircle(80, 80, 10, 0, 1);

```

```

// Name: Call Component Macro, Type: Call Component Macro: gLCD_ILI9341_SPI1::DrawCircle(160, 80, 10, 0, 1)
FCD_0a2b1_gLCD_ILI9341_SPI1_DrawCircle(160, 80, 10, 0, 1);

```

```

// Name: Call Component Macro, Type: Call Component Macro: gLCD_ILI9341_SPI1::DrawCircle(240, 80, 10, 0, 1)
FCD_0a2b1_gLCD_ILI9341_SPI1_DrawCircle(240, 80, 10, 0, 1);

```

```

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

```

```

// Name: Call Component Macro, Type: Call Component Macro: gLCD_ILI9341_SPI1::ClearDisplay()
FCD_0a2b1_gLCD_ILI9341_SPI1_ClearDisplay();

```

```

}

```

```

}

```

```

/*=====*\
Use :
\*=====*/
void FCM_APERTO()
{

```

```

// Name: Call Component Macro, Type: Call Component Macro: gLCD_ILI9341_SPI1::ClearDisplay()
FCD_0a2b1_gLCD_ILI9341_SPI1_ClearDisplay();

```

```

// Name: Call Component Macro, Type: Call Component Macro: gLCD_ILI9341_SPI1::SetForegroundColour(0, 255, 0)
FCD_0a2b1_gLCD_ILI9341_SPI1_SetForegroundColour(0, 255, 0);

```

```

// Name: Call Component Macro, Type: Call Component Macro: gLCD_ILI9341_SPI1::Print("APERTO", 100, 150, 12, 0)
FCD_0a2b1_gLCD_ILI9341_SPI1_Print("APERTO", 7, 100, 150, 12, 0);

```

```

// Name: Output, Type: Output: 1 -> D7
SET_PORT_PIN(D,7,(1));

```

```

// Name: Delay, Type: Delay: 3 s
FCI_DELAYBYTE_S(3);

```

```

// Name: Output, Type: Output: 0 -> D7
SET_PORT_PIN(D,7,(0));

```

```

// Name: , Type: Calculation:
// Count = 0
// COD1 = 0
// COD2 = 0
// CountSwA = 0
FCV_COUNT = 0;
FCV_COD1 = 0;
FCV_COD2 = 0;
FCV_COUNTSWA = 0;

```

```

// Name: Call Component Macro - Pulsante premuto, Type: Call Component Macro: QuadEncoder2::ResetCounter()
FCD_017b1_QuadEncoder2_ResetCounter();

```

```

// Name: Call Component Macro, Type: Call Component Macro: gLCD_ILI9341_SPI1::SetForegroundColour(0, 0, 255)
FCD_0a2b1_gLCD_ILI9341_SPI1_SetForegroundColour(0, 0, 255);

```

```

}

```