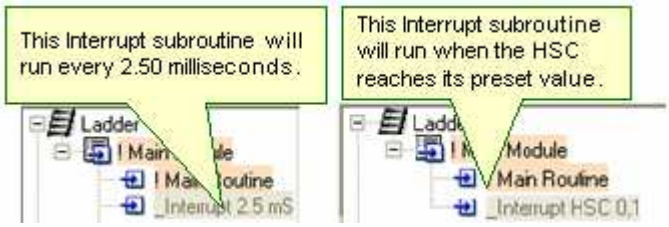
Interrupt Routines

Interrupt routines cause:

- A program to stop immediately, whenever the interrupt is activated, even if the program is in the middle of scanning a net in another subroutine.
- A jump to the Interrupt subroutine. An Interrupt subroutine must have the **exact name shown in the examples below**.
- When the interrupt routine is finished, the program returns to where it was interrupted, and continues from that point until the next Interrupt arrives.

Interrupt routines are generally used with Immediate elements, for example to turn an output ON in case of an alarm or emergency. To call an interrupt routine:

1. Include an Interrupt subroutine of the **correct name** in your program; the subroutine is executed **automatically** when the condition for calling it is filled.



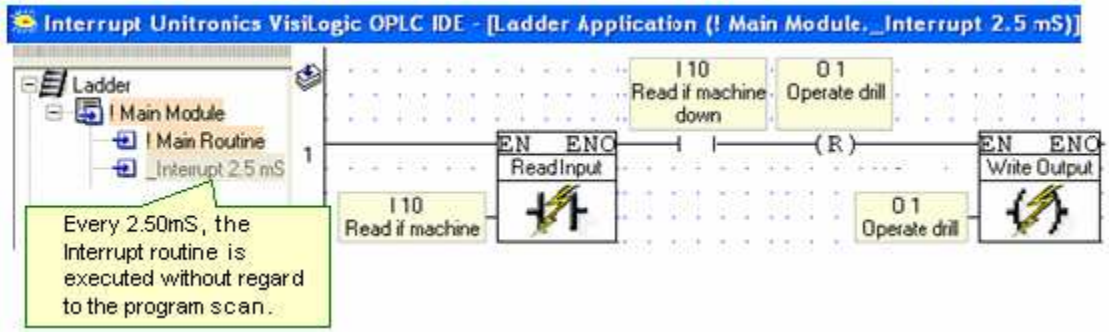
Note ♦ If the **name** of the subroutine is **incorrect,** the subroutine **will not function as an Interrupt** routine.

♦ Interrupt features are not supported by the V120-12 series.

Sample applications showing how to use Interrupt routines in conjunction with Immediate elements may be located in :::\ ProgramFiles\Unitronics\VisiLogic\Examples.

2.5 mS Interrupt Routine

This function is timed-based. Call it by naming a subroutine **_Interrupt 2.5 mS**



Including an **_Interrupt 2.5 mS** subroutine in the Ladder application causes:

- The program scan to pause every 2.50 mSec.
- A jump to the subroutine named **_Interrupt 2.5 mS**
  Note that the interrupt routine should be as short as possible, and must not exceed approximately 0.5 mSec.

When the interrupt routine is finished, the program returns to where it was interrupted, and continues from that point until the next Interrupt arrives.

Note ♦ The Subroutine _Interrupt 2.5 mS will run for the first time **after** the first Ladder scan is run.

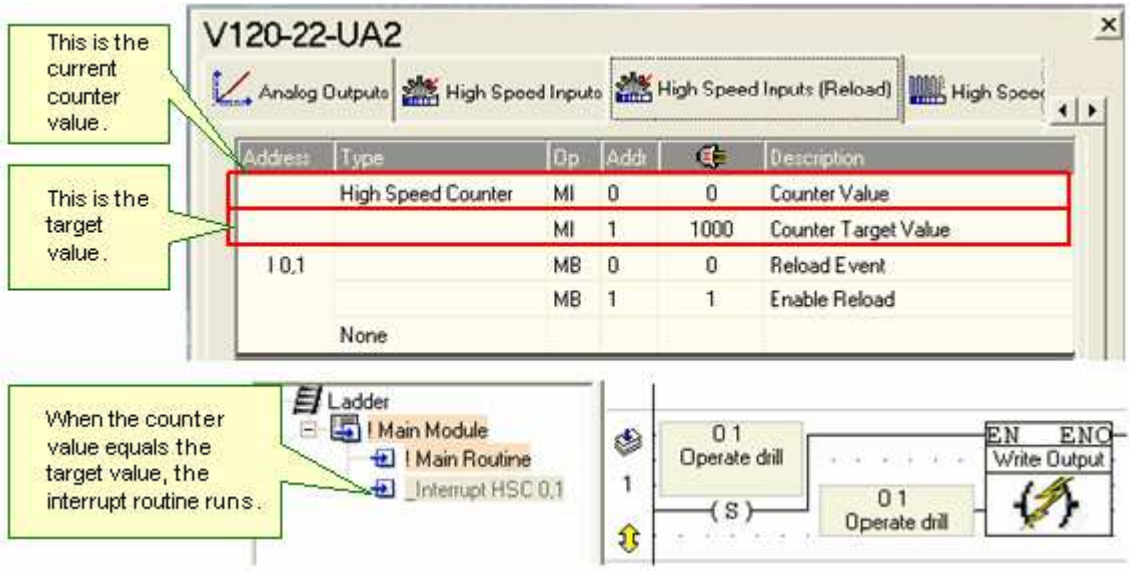
1.25 mS Interrupt Routine

This function is supported by Enhanced Vision models **only**. Call it by naming a subroutine **_Interrupt 1.25 mS**

It functions exactly like the 2.5mS Interrupt routine described above.

Interrupt HSC

This function is called according to the current value of a high-speed counter. The program stops immediately and executes the subroutine when the Counter Value reaches the Counter Target Value.



The interrupt function is included in the program by naming a subroutine **_Interrupt x,x** where the first x is the high-speed counter, and the second x is the reload. These subroutines must be named in accordance with your Hardware Configuration as:

- _Interrupt HSC 0,1
- _Interrupt HSC 2,3
- _Interrupt HSC 4,5

When the interrupt routine is finished, the program returns to where it was interrupted, and continues from that point until the next Interrupt arrives.

Related topics

Immediate Elements

Program Sequencing: Modules, Subroutines, Labels & Jumps