



Write a self-contained function block in 61131-3 structured text that implements a counter.



In IEC 61131-3 Structured Text, a counter can be implemented using a function block. Here's a simple example of a function block for an up-counter:

iebst

Copy code

```
FUNCTION_BLOCK UpCounter
    VAR_INPUT
        Reset: BOOL := FALSE;
        Enable: BOOL := FALSE;
    END_VAR

    VAR_OUTPUT
        Count: UDINT := 0;
    END_VAR

    VAR
        PrevEnable: BOOL := FALSE;
    END_VAR

    METHOD Run: BOOL
        IF Reset THEN
            Count := 0;
        ELSIF Enable AND NOT PrevEnable THEN
            Count := Count + 1;
        END_IF;

        PrevEnable := Enable;
        RETURN TRUE;
    END_METHOD
END_FUNCTION_BLOCK
```

To use this function block, you would need to create an instance of it in your main program and call the `Run` method periodically, providing the appropriate input values for the `Reset` and `Enable` signals.

iebst

Copy code

```
PROGRAM Main
    VAR
        Counter: UpCounter;
        ResetSignal: BOOL := FALSE;
        EnableSignal: BOOL := FALSE;
    END_VAR
END_PROGRAM

(* Inside your main control loop, call the Run method with the Reset and Enable signals *)
Counter.Reset := ResetSignal;
Counter.Enable := EnableSignal;
Counter.Run();
```