

# Appendix C

## Using Counter Variables in BASIC Programs

You can use counter elements in BASIC programs. To do this, you must declare the counter as a global, double integer array. For example, COMMON COUNTER!(4). The elements of the counter data structure can be accessed as follows:

Element:	Description:
0	Current
1	CPreset
2	Reserved for system
3	bit 7        LD bit 15       CR bit 23       QD bit 31       QU
4	Reserved for system

### WARNING

**DO NOT MODIFY TIMER ELEMENTS 0, 1, 3, AND 4 FROM A BASIC PROGRAM. THIS WILL CAUSE THE TIMER TO OPERATE INCORRECTLY, RESULTING IN UNPREDICTABLE MACHINE OPERATION.**

Within a BASIC program you may want to:

- change a counter's preset
- use the Current element of a counter for data display, logic sequencing, or other purposes
- monitor the status of a counter

**To change the preset value by using a BASIC statement**

- Enter a statement in this format:

`name!(1)=new_preset_value`

where NAME! is the name of the counter variable

The value of 1 is the location of the element CPreset within a Counter data type.

Example:

`COUNTER1!(1)=50`

This statement is setting the preset value for the global variable COUNTER1 to 50.

**To specify the Current element of a counter in a BASIC statement**

- Enter the variable in this format:

`NAME!(0)`

where NAME is the name of the counter variable

The value of 0 is the location of the element Current within a Counter data type.

**To determine if a counter has reached its preset by monitoring element QU in a BASIC program**

- Use the BASIC function BIT\_SET@ to test the value of QU.

Example:

```
IF BIT_SET@(COUNTER1!(3),31) THEN 250
```

This statement examined bit 31 of element 3 in the global variable COUNTER1.