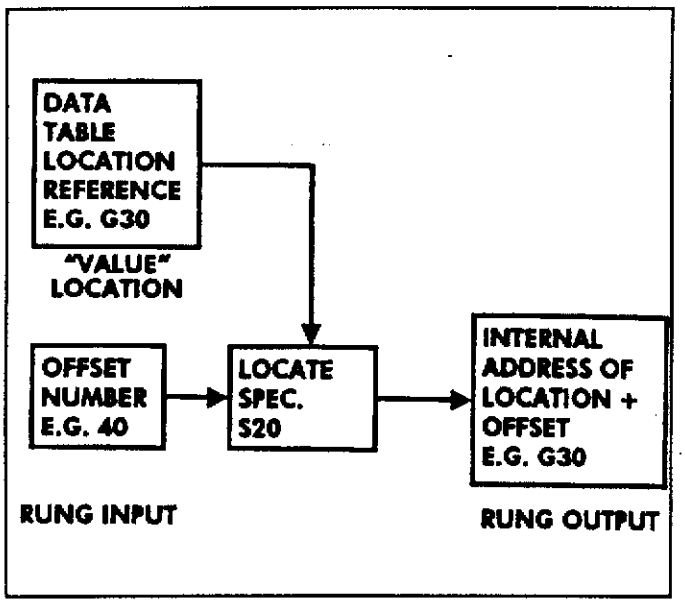
CEGELEC
INDUSTRIAL CONTROLS

Locate Data Table Address

LOCATE S20

DESCRIPTION



The special functions LOCATE finds the internal memory address of any specified table location.

Your program needs to know internal memory addresses when it uses some other types of special function such as MOVE (T20) and ADDARAY (T12). The software data for any special function explains whether it makes use of internal addresses and whether you need to use LOCATE in conjunction with it.

The reason for needing to use LOCATE is that in a GEM80 Controller, the internal address of any given table location is not fixed. It varies depending on what other table locations are used in your program. When you recompile a program after editing it, the internal address of a given table location could finish up different to what it was before. You therefore need to find it again with another LOCATE.

FEATURES

- Finds address of defined table location.
- Adds variable offset.
- Checks validity of resultant address.

SPECIFICATION

Control language:-

```
        LOCATE
X-------SPEC-----VALUE---------Q
          S20       Zm
```

Table 1 **Program rung data**

Sym	Location	Use	Range
X	Rung Input	Offset	±9999
Zm	Value location	Table location to be found	Any table usable by special functions
Q	Rung output	GEM80 address	Hex code not meaningful to user. Under fault conditions, value is -1 (@FFFF)

Table 2 **Fault Conditions**

Condition	Rung Output	
	Numeric Value	States ON
PROGRAMMING ERRORS no VALUE after SPEC VALUE is a number instead of an address	 −1 −1	 All All
STATUS Offset X <-9999 or >9999 Calculated address (VALUE Zm) in table not usable by special functions Calculated address (VALUE Zm plus offset X) beyond end of data table's memory area.	 −1 −1 −1	 All All All

Notes: Where special function make use of internal addresses (provided by LOCATE) as input, they first check that these addresses fall inside the table's area of memory.

When LOCATE gives an output of −1 under any of the fault conditions listed in Table 2, then, since −1 is not an address inside the tables area, the particular special function will treat this as a fault condition and produce its own fault code. Refer to the software data for individual special functions for details of their fault codes.

APPLICATION NOTES

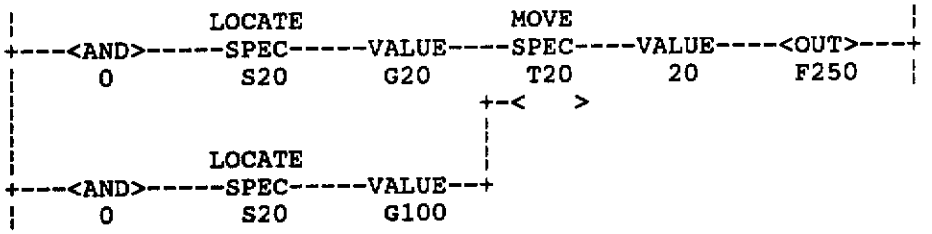
1. POTENTIAL MISUSE

The LOCATE function is intended primarily for use with other special functions that require internal addresses as rung or value table inputs. As noted in the description, internal addresses can change when a program is edited. After any program recompilation therefore, you need to use LOCATE at least once to again find each internal address used in the program.

On older models of GEM80 Controller that do not have the recompilation facility, we still recommend that you use LOCATEs in your program. We do not guarantee that internal address values will be the same for different implementations of a particular model of Controller. If you don't use LOCATE, you could run into unforseen problems.

2. USE WITH MOVE (T20) WITH A FIXED OFFSET

When you use the MOVE special function T20 in your program, you should take particular care. If wrongly programmed, MOVE can transfer data to the wrong place, and overwrite good data. Use a fixed offset (zero) whenever practical. This reduces the chances of a bug elsewhere in your program causing MOVE to transfer data to the wrong place.

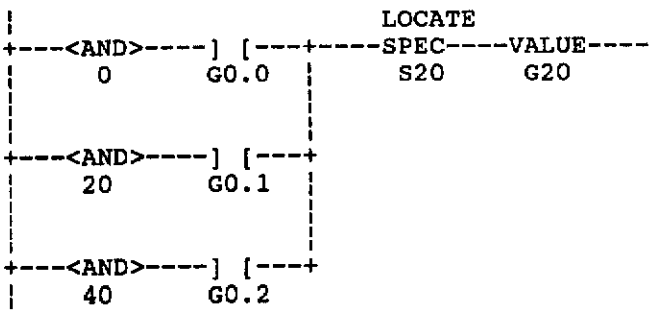


This rung copies the data from G20 to G39 into G100 to G119 respectively.

3. USE WITH MOVE (T20) WITH A VARIABLE OFFSET

If you substitute the following program for the upper branch in the previous example, it makes the data to be copied dependant on which state in G0 is ON.

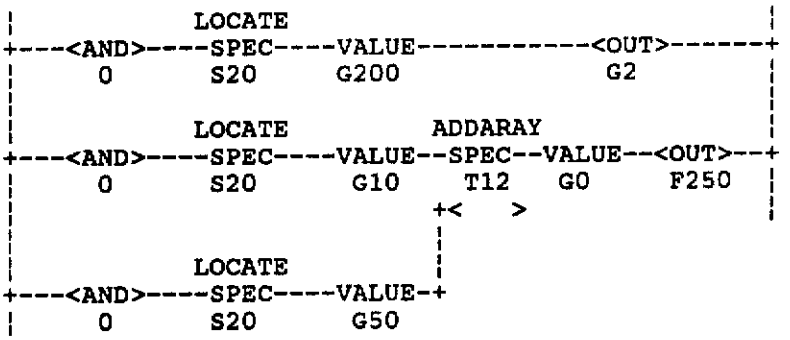
i.e. G20 to G39 if G0.0 is ON,
G40 to G59 if G0.1 is ON,
etc.



The rung assumes only one bit in G0 is ON at any one time, e.g. G0 is a sequencer function. Otherwise you would need to add the branches together instead of wired-O Ring them as shown.

4. USE WITH ADDARAY (T12)

The couple of rungs below add together arrays of G-table locations, where the first array begins at G10, the second array begins at G50, and the result of the addition goes into an array starting at G200. The internal address of each of these starting points has to be found using LOCATE.



The quantity of table locations in the arrays (which obviously have to be all the same size) is held in G1. For further details, refer to software data for array special functions T12-T15.

5. FORMAT OF INTERNAL ADDRESSES

On most GEM80 Controllers, if you use LOCATEs to find the internal addresses of two adjacent table locations such as G100 and G101, and if you look at the resultant numbers, you will find that the second one is 2 more than the first. However, on newer models of GEM80 Controller, you may find that the numbers differ by only 1.

This is because the internal addresses are byte addresses on the first model of Controller, but word addresses on the second. The set of special functions within any model of Controller will, however, be consistent; they will either all use byte addresses or else all word addresses.

CEGELEC
INDUSTRIAL CONTROLS

Kidsgrove, Stoke-on-Trent, ST7 1TW, England
Tel: (0782) 783511 Fax: (0782) 776329

Publication No. T322 Issue 5