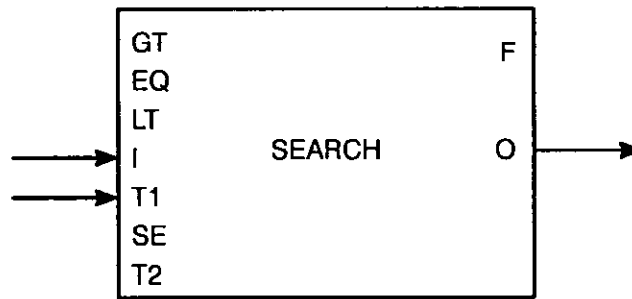


# 29.0 SEARCH

This function can be used in AutoMax Control Block tasks and UDC Control Block tasks.



## Function

Compare INPUT against selected TABLE elements.

Search the selected table for a match according to the comparison options selected by the three BOOLEAN inputs COMPARE\_GTR, COMPARE\_EQU and COMPARE\_LES. The search starts at the top of the table (array element 0) and tests INPUT against each element in the table until either a match is found or the end of the table is reached (last element in the array). If a match occurs, the search function is terminated and the index into the table where the match occurred is written to OUTPUT. FOUND is then set true. If no match occurred, then OUTPUT is set to a value of -1 and FOUND is set FALSE.

## Program Statement

```
CALL SEARCH (COMPARE_GTR=<boolean literal>,           &
             COMPARE_EQU=<boolean literal>,           &
             COMPARE_LES=<boolean literal>,           &
             INPUT=input%,                             &
             SELECT=select@,                           &
             TABLE1=table1%, TABLE2=table2%,       &
             FOUND=found@, OUTPUT=output%)
```

## Inputs

I (INPUT) =

Data input, type INTEGER. This parameter must be specified as a numeric symbol only (literal value not accepted).

GT (COMPARE\_GTR) =

Compare INPUT > TABLE(n), type BOOLEAN. This is an optional parameter. The default for this parameter is FALSE. It must be entered explicitly as a boolean literal.

**EQ (COMPARE\_EQU) =**

Compare INPUT = TABLE(n), type BOOLEAN. This is an optional parameter. The default for this input is FALSE. It must be entered explicitly as a boolean literal.

**LT (COMPARE\_LES) =**

Compare INPUT < TABLE(n), type BOOLEAN. This is an optional parameter. The default for this parameter is FALSE. It must be entered explicitly as a boolean literal.

**T1 (TABLE1) =**

Search table #1, type INTEGER. This parameter must be specified as a variable name only (literal value not accepted). TABLE1 must be defined as a single dimension INTEGER array of length N. This array definition is used when you define the variable using a LOCAL or COMMON statement in the task.

**SE (SELECT) =**

This input selects the TABLE that is to be searched, type BOOLEAN. The default is FALSE. When FALSE, TABLE1 will be searched, else TABLE2 will be searched. If this parameter is specified, then parameter TABLE2 must also be specified.

**T2 (TABLE2) =**

Search table #2, type INTEGER. This parameter is optional. If specified it must be a variable name only (literal value is not accepted). If TABLE2 is specified then the parameter SELECT must also be specified. TABLE2 must be defined as a single dimension INTEGER array. The length of TABLE2 must be equal to TABLE1. This array definition is used when you define the variable using a LOCAL or COMMON statement in the task.

## Outputs

**O (OUTPUT) =**

Data output, type INTEGER. This parameter must be specified. If a match occurs, OUTPUT is set equal to the array element (0 to N) in the table that met the search requirement. If no match occurs, then OUTPUT is set equal to -1.

**F (FOUND) =**

Found match output, type BOOLEAN. This is an optional parameter. This output is TRUE if an element in the selected table matched the INPUT according to the selected comparison parameters (COMPARE\_GTR, COMPARE\_EQU or COMPARE\_LES). If no match occurs, this output is set FALSE.

## Notes

1. The order in which the comparison options are programmed is unimportant. However, the number of comparison options

selected (programmed TRUE) must be a minimum of one and a maximum of two. If this requirement is not met, a compilation error will occur.

2. TABLE1 (and TABLE2 if specified) must be defined as a single dimension INTEGER array whose size (number of elements) is defined in a variable definition statement (LOCAL or COMMON) (e.g. LOCAL TABLE1%(16)). The size of TABLE2 if specified must be the same as TABLE1. The SEARCH block will begin searching through the selected table at the first element in the array, which is element 0, not element 1.
3. When loading data into the table(s) it may be necessary to manually sort the data in ascending or descending order.

Example 1: If COMPARE\_GTR mode has been specified, then sort the data in descending order.

```
INPUT = 15          TABLE 1%(0) = 40
                   TABLE 1%(1) = 30
                   TABLE 1%(2) = 20
then               TABLE 1%(3) = 10

OUTPUT = 3 and FOUND = TRUE
```

Example 2: If COMPARE\_LES mode has been specified, then sort the data in ascending order.

```
INPUT = 1501       TABLE 1%(0) = 100
                   TABLE 1%(1) = 200
                   TABLE 1%(2) = 300
then               TABLE 1%(3) = 400

OUTPUT = 1 and FOUND = TRUE
```