

GPI Input Functions

Created to simplify programming of RCP panels and other GPI input-intensive TSI1000 configurations:

IGM function

IGM(channel, length, offset, transition, cascade): ("input group mixer") multi-selects GPI inputs, returning a list of relative numbers representing the selected GPI inputs in a group of consecutive GPI inputs. In a typical application, pressing a button on an RCP panel allows more than one button in a group to be selected at the same time. IGM returns a list of comma-separated numbers starting from "offset".

Channel: starting address.

Length: number of consecutive addresses monitored, including the starting address. Default: 1

Offset: The value returned when the first GPI input of the group is selected. Other returned values are consecutive. This allows control of the starting value of the returned list. Default: 0

Transition: A value of "1" enables detection of a off-to-on transition, a value of "2" enables detection of an on-to-off transition, and a value of "3" enables detection of both transition edges. A value of "0" causes the function to return an empty string. Default: "1" (off-to-on transition).

Cascade: This allows another input list to be appended to the list returned by IGM. This is useful for cascading IGM functions for the handling of disjointed ranges of inputs. Default: no list.

Example IGM(4,3,1) multi-selects GPIs 4,5,6 and returns some combination of the numbers "1","2","3".

IGS function

IGS(channel, length, offset, transition, cascade, user bit, variable)): ("input group select") single-selects GPI inputs, returning a single value, the relative number of the most-recently-changed GPI input. In a typical application, pressing a button on an RCP panel selects the last-pressed button in a group of buttons, allowing one button to be selected at a time, while automatically canceling previous selections.

See IGM function for use of channel, length, offset, transition, cascade parameters.

User bit: A value of 0 to 7 allows more than one IGS function to monitor the same GPI input. A unique user bit value would be used in each instance of IGS functions that monitor overlapping ranges of GPI inputs in a given TSI1000 configuration. For example if IGS(0,2,1) and IGS(0,3,1) (overlapping GPI inputs 0 and 1), then IGS(0,2,1,,,0,*S) and IGS(0,3,1,,,1,*S) would be programmed, where the user bit numbers are "0" and "1" respectively. Default: 0

Variable: This static variable is required for working memory for detecting and tracking GPI input changes. IGS functions used in the same display, GPI output or output control must have different static variable names. Default: none; a static variable name, starting with the asterisk character, must be supplied.

Example: IGS(0,3,1,,, *S) would single-select one of GPI inputs 0, 1 and 2, returning one of numbers "1", "2" or "3", depending on which GPI input was last toggled from the off to the on state.

IVT function

IVT(channel, transition, modulus): returns a modulus of the count of the number of transitions detected on a given GPI input. In a typical application this can be used to toggle through a set of numbers by repeated presses of a key on an RCP panel.

See IGM function for use of channel and transition.

The modulus value causes the returned transition count to be limited to the range zero to one less than the “modulus” parameter value. Default value: 2 (this causes the IVT function to return “0” or “1” on alternating transitions, making the returned value behave as an on-off toggle).

Example: IVT(0,1,4) returns one of values “0”, “1”, “2”, “3” on each transition of GPI input “0” to from the off to the on state.

IVC function

IVC(channel, length, transition, user bit): Returns a value of 1 when a change is detected on a given set of GPI inputs. The 1 is returned for only one evaluation of the expression; subsequent evaluations without another transition on the set of GPI inputs will return “0”. Therefore this function must usually be used in conjunction with a variable in order to trap GPI input transitions.

See the IGS function for use of the channel, length, transition, and user bit parameters.

Example: sv(COUNT,if(ivc(0,4,1,0),add(v(COUNT)1,),v(COUNT))) will increment variable COUNT each time any of GPI inputs 0 to 3 toggle from the off to the on state.

N2B Function

N2B(List, Offset): converts list of numbers to a 64-bit bitmap. This function is useful for converting returned lists from the IGS and IGM functions to a single number, which can be used to drive multiple-bit GPI outputs.

List: List of numbers separated by commas. Default: none.

Offset: A value subtracted from each number in the list, for the purpose of normalizing the values before converting the list to a number. Default: 0

Example: N2B(“0,1,2”,0) returns the number “7”. N2B(“10,11,12”,10) also returns the number “7”.

N2B(IGM(0,4,0)) applied to a set of 4 RCP buttons starting at GPI input address “0” would return “0” when the first button is selected, “1” when the second button is selected and “3” when the first two buttons are selected simultaneously. This expression could be directly inserted into an output control of size 4 to control the RCP panel LEDs in accordance with the button selections.

B2N function

B2N(Number,Offset) converts a 64-bit decimal number to a comma-separated list of numbers representing the bits set in the binary representation of the number. The lowest possible bit number that can be returned is “0”, the second lowest bit number that can be returned is “1”, and so on.

Number: a 64-bit number represented as a decimal value, between zero and 18446744073709551615.

Offset: A value which is added to each bit number before the list is returned.

Example: B2N(15,0) returns “0,1,2,3”, a list of the bits which are set in the binary representation of the number “15”. B2N(15,5) returns “5,6,7,8”.