**Module Requirements of check\_dips function:**

There are four units --- unit0, unit1, unit2, unit3.

Unit0 has 128 channels -- Represented by number 0

Unit1 has 96 channels -- Represented by number 1

Unit2 has 96 channels-- Represented by number 2

Unit3 has 96 channels-- Represented by number 3

Status of these channels is acquired by System1 and System2 and is available in survstatbuf[], survstatbuf2[] respectively. Byte allocation to various units is as follows:

survstatbuf[0] to survsattbuf[15] contains the status of 128 channels of Unit0 ( 8 \*16 BYTES = 128, each byte represents the status of 8 channels)

survstatbuf[16] to survstatbuf[27] contains the status of 96 channels of Unit1 ( 8 \* 12 bytes = 96)

survstatbuf[28] to survstatbuf[39] contains the status of 96 channels of Unit2 ( 8 \* 12 bytes = 96)

survstatbuf[40] to survstatbuf[51] contains the status of 96 channels of Unit3 ( 8 \* 12 bytes = 96)

The purpose of program is to verify the status of the intended channels of any unit w.r.t the required ON/OFF state.

**Input parameters:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| S.No. | Parameter name | Data Type | Global/Local | Usage |
|  | dips[100] | unsigned char | Local | * This is an array which contains the number of channels to be checked and the information about the unit number and its corresponding channel number. * 0th index corresponds to number of channels to be checked, 1st index corresponds to unit number, 2nd index corresponds to channel number. Same is followed till the number of channels. |
|  | onoff | bool | Local | 0 - to be checked for OFF;  1 - to be checked for ON |
|  | survstatbuf[52] | unsigned char | Global | * contains the status of channels as received from System1. * Description of the byte allocation is detailed in the subsequent flow. |
|  | survstatbuf2[52] | unsigned char | Global | * contains the status of channels as received from System2 |
|  | lookup\_status[4][128] | typedef struct STATUSPARMS | Global | * First dimension of the array represents unit number, 2nd dimension of the array represents the channel number of the unit. * Used to identify whether common channel exists. * For the present logic, this structure is to be considered for first dimension array element 0 ie. Only lookup\_status[0][channel] is to be taken into account. |

Struct STATUSPARMS is defined as:

typedef struct status

{

char name[30];

unsigned short unit;

unsigned short channel;

unsigned short common;

unsigned short type;

}STATUSPARMS;

Description of the fields of the above structure for lookup\_status[0][channel]:

* name –not required for this logic
* unit- represents unit number.

i.e lookup\_status[0][channel].unit can take values 0, 1, 2, 3 and 300

0- represents unit 0

1 – represents unit 1

2 – represents unit 2

3- represents unit 3

300 – represents that the channel referred in the 2nd dimension of the array is a common channel

* channel –represents channel number . Not required for this logic
* common –represents whether the channel referred in the 2nd dimension array element has a common channel number or not.

- lookup\_status[0][channel].common can take values from 0 to 127 and 300

- if lookup\_status[0].[channel].common != 300 == > there is common channel number for the channel number referred in the 2nd dimension of this array.

- if lookup\_status[0].[channel].common == 300 == > there is no common channel number for the channel number referred in the 2nd dimension of this array.

* type – not required for this logic

**Output:** Function return value (either 0 or 1).

**Logical requirements:**

**If it is intended to verify the status of the unit0 channel, then following are the requirements:**

if lookup\_status[unit0][chno].unit != 300 && lookup\_status[unit1][chno].common !=300 then the return value of get\_tmr\_status\* is to be caught. If the return value of this function is not equal to “onoff” then the result of status check for the intended channel number should be 0, otherwise 1.

if lookup\_status[unit1][chno].unit != 300 && lookup\_status[unit1][chno].common ==300 then the return value of get\_dmr\_status\*\* is to be caught. This function returns the status check of the intended channel.

**If the status of the channels of the units, other than the unit0 is to be checked, following are the requirements:**

If intention is to be checked of ON, and the status of the channel in the survstatbuf[] is 1, result of status check for this channel is 1, otherwise 0.

If intention is to be checked of OFF, and the status of the channel in the survstatbuf[] is 0, result of status check for this channel is 1, otherwise 0.

**Function output:**

If the status of all the channels listed in the buffer dips[], is as expected then the return value of the function should be 1, otherwise 0.

\*\*\*\*\*/

\*get\_tmr\_status is problem2 which we have addressed. It returns either 0 or 1 or 2. 0 implies OFF state, 1 implies ON state and 2 implies undefined state.

\*\*get\_dmr\_status is problem1, which we have addressed. It returns 0 or 1. Here, 0 implies system is in not in expected state. 1 implies systems is in expected state.