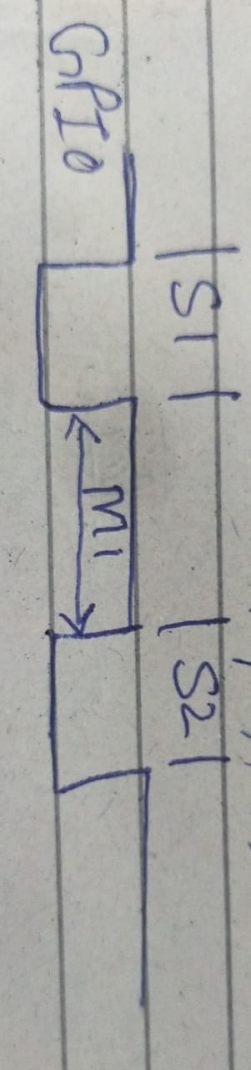
### Button Press Working:

#### Scenario:

Using the following logic to increase the volume by 10 units in my Audio Application, when I press the button for like in an instant way, it does nothing but when I pressed it(kind of hold like, for little longer than instant press) then it increment the volume.

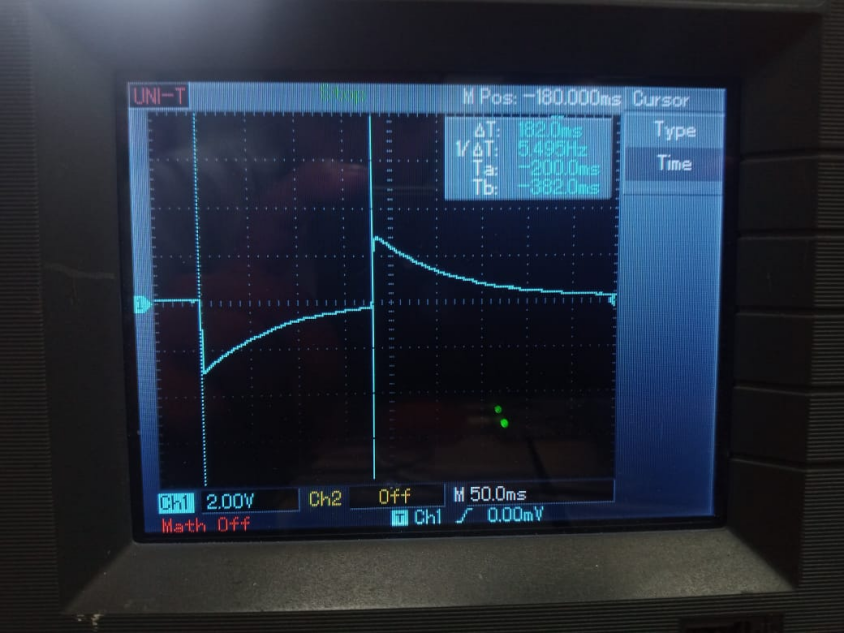
**Assumption:**

Let assume S1=S2= 100ms, and M1 = 500ms kind of seem like humanly possible.



**Test:**

**Normal Button press(like instant):**



so duration is 180ms(approx.), here no volume changes.

**Work Button Press:**



Duration is like 280ms, and upon here Volume changes.

#### My Solutions:

* First I think I should the loop Iteration time, which should be like 5 to 10 times faster than the button press duration for properly working,
  + Solution:  
     Printing Some text like after every 100 or like 1000 iteration and noting the time, then dividing the time by that amount of iteration value, which will give the time of each iteration.

if(count == 1000)

{

printf("hello, 1000 cycles completed...\n\r");

count = 0;

}

count++;

* + Results:  
     iteration of our loop is approximately like 100ms(11.3s/100).

Using the below statement:  
if(BoardController::VolumeUpKeyTicks > 1 && BoardController::VolumeUpKeyTicks < MEDIUMPRESSDURATION)

The main loop has to be executed 2 time with in the press button(holding it for at least two iteration to change the volume), so by change the number **1 to 0**, will make the button to work(mean give us better result from the previous one) upon the single iteration. Technically this will mean that first we have to press the button for around 300ms for loop iteration of 100ms, now we reduced it by half like 150ms button press.

**Main Solution:**

We have delay of actually like: vTaskDelay(10); -> 100ms

So I reduce it like: vTaskDelay(1 / portTICK\_PERIOD\_MS); -> 10ms

Now we making again the value from 0 to 1, because as the loop iteration time is much reduced, and the reason of doing so is used for the avoiding he debouncing of button, so we can have like to iterate the loop while doing nothing and the just change the volume.

#### Code Structure:(Given)

While(1)

{

if(gpio\_get\_level(BUTTON\_VOLUP\_ID) == 0)

    {

        BoardController::inactivity\_counter = xTaskGetTickCount();

        if (BoardController::VolumeUpKeyTicks < MEDIUMPRESSDURATION)

        {

            BoardController::VolumeUpKeyTicks++;

        }

    }

    else

    {

        BoardController::VolumeUpKeyTicks = 0;

    }

if(BoardController::VolumeUpKeyTicks > 1 && BoardController::VolumeUpKeyTicks < MEDIUMPRESSDURATION){

            BoardController::VolumeButtonEventCallBack(10);

            BoardController::VolumeUpKeyTicks = MEDIUMPRESSDURATION\*2;      //Give it a bigger number to chew

            UpKeyPress = true;

        }

vTaskDelay(10);

}

#### Code Structure:(Updated)

While(1)

{

if(gpio\_get\_level(BUTTON\_VOLUP\_ID) == 0)

    {

        BoardController::inactivity\_counter = xTaskGetTickCount();

        if (BoardController::VolumeUpKeyTicks < MEDIUMPRESSDURATION)

        {

            BoardController::VolumeUpKeyTicks++;

        }

    }

    else

    {

        BoardController::VolumeUpKeyTicks = 0;

    }

if(BoardController::VolumeUpKeyTicks > 1 && BoardController::VolumeUpKeyTicks < MEDIUMPRESSDURATION){

            BoardController::VolumeButtonEventCallBack(10);

            BoardController::VolumeUpKeyTicks = MEDIUMPRESSDURATION\*2;      //Give it a bigger number to chew

            UpKeyPress = true;

        }

vTaskDelay(1/ portTICK\_PERIOD\_MS); }