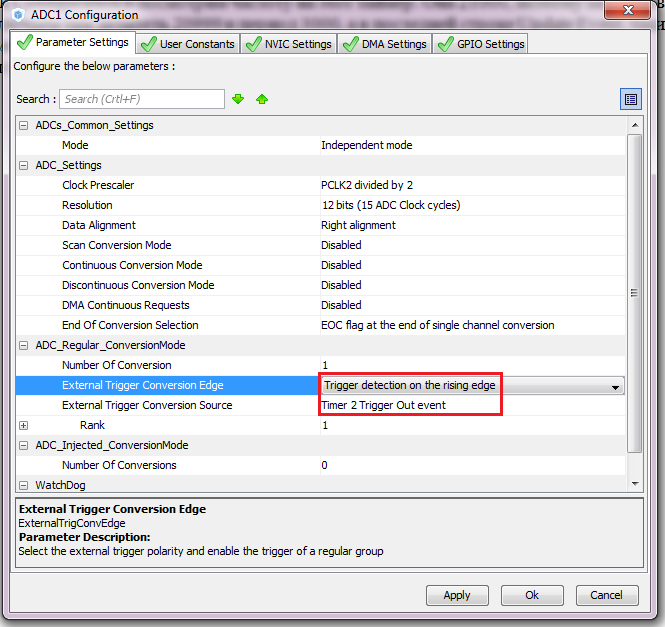
**HAL. ADC. Regular Channel. Trigger**

We create the project from **ADC\_REGULAR**, we call it **ADC\_REGULAR\_TRIGGER**.

Start the Cube. We add 2 timer, including in it the internal clock. We also **include PD12**on the output.Then go to **Clock Configuration** and see the frequency for this timer (page 67 Ref Manual). It is **21000**, so go to the **Configuration**tab  and include the divisor **20999** and the period **3000**, and in the last line of the Update Event, so we set the timer for a period of **3** seconds, and we also need to turn on the interrupt timer.We go into the ADC, turn off the interrupts. And in 1 tab, turn on the trigger



Generate the project, open it, configure the programmer, connect the file **lcd.c** and **compile the**code.

Reduce the delay, clear the display, start the timer and ADC in the main function.

        LCD\_String (str);

**HAL\_Delay (500);**

        LCD\_Clear ();

        LCD\_SetPos (4, 0);

        LCD\_SendChar ('s');

        LCD\_SetPos (8, 1);

        LCD\_SendChar ('t');

        LCD\_SetPos (12, 2);

        LCD\_SendChar ('m');

        LCD\_SetPos (16, 3);

        LCD\_SendChar ('3');

        LCD\_SendChar ('2');

        HAL\_Delay (500);

**LCD\_Clear ();**

**HAL\_TIM\_Base\_Start\_IT (& htim2);**

**HAL\_ADC\_Start (& hadc1);**

Let's add a function for processing the timer interrupt by the match flag (page 842 of the HAL library manual) and we will change the state of the LED

/ \* USER CODE BEGIN 4 \* /

**void HAL\_TIM\_PeriodElapsedCallback (TIM\_HandleTypeDef \* htim2)**

**{**

**HAL\_GPIO\_TogglePin (GPIOD, GPIO\_PIN\_12);**

**}**

/ \* USER CODE END 4 \* /

Now take the data from the ADC and show them on the display. Nothing special is needed to do this. You just need to comment out the unnecessary code and take away some delay.

  while (1)

  {

**//**                HAL\_ADC\_Start (& hadc1); // run the analog-to-digital conversion

// HAL\_ADC\_PollForConversion (& hadc1,100); // wait for the end of transformations

                        u = ((float) HAL\_ADC\_GetValue (& hadc1)) \* 3/4096; // enter the result of the transformations into a variable

// HAL\_ADC\_Stop (& hadc1); // stop the conversions

                sprintf (str, "%. 2fv", u); // convert the result to a string

                LCD\_SetPos (0,3); // show the result on the LCD display

                LCD\_String (str);

                HAL\_Delay (100); // delay before next cycle

  / \* USER CODE END WHILE \* /

We compile. Stitching. We are looking at the results.