Mark	

Team name:	*A1*		
Homework number:	*HOMEWORK 2*		
Due date:	08/10/2023		
Contribution	NO	Partial	Full
Monti Pietro			*x*
Moretto Alessia			*x*
Pallotto Francesco			*x*
Perna Alessandro			*x*
Ventura Ludovico			*x*
Notes:			

Project name	"Play a song"		
Not done	Partially done (major problems)	Partially done (minor problems)	Completed
			x

Explanation:
We successfully completed the homework.

Part 1:

The request is to play the song with the speaker when the microphone detects a loud sound. To do this we exploited the HAL_GPIO_EXTI_Callback function: in case of an interrupt sent by pin 8, linked to the microphone, we set a variable called "flag" to 1.

Then, inside while(1), we check, as condition to play the song, the value of the flag, then resetting it to 0 only once the sequence of notes has been entirely played, in order to be ready to sense new interrupts, but also to avoid the song to restart without the loud sound.

Part 2:

The way we chose not to use HAL_DELAY functions is the exploitation of timer 2 peripheral. Our general idea was to measure the time passing using timer 2, channel 1, and to exploit the interrupt sent by the peripheral, every time the counter reaches the value set in the autoreload register, as a trigger to pass from a note to the following one in the score declared.

Starting from the last code version seen at lecture, we modified the structure of "playnote" function configuring timer 2 autoreload register with the function __HAL_TIM_SET_AUTORELOAD(&htim2, note_playing.duration*10*TEMPO). The argument is expressed in milliseconds (to be coherent with the previous HAL_DELAY functions), and the factor 10 multiplication is necessary to "slow down" to 1 kHz the effective timer frequency, whose theoretical minimum is 10 kHz.

We decided no more to use the "playsong" function in order to be able to exploit the interrupt sent by the timer note by note. Inside the function HAL_TIM_PeriodElapsedCallback we call "playnote" to

change note every time timer 2 sends an interrupt. To select the term of the score, we use a variable called “note_counter”.

By means of the variable “song_playing”, we are able to distinguish between the cases in which notes are played or not: in the former case, in the HAL_GPIO_EXTI_Callback, when receiving the interrupt signal from the microphone pin, we start timer 2 using HAL_TIM_Base_Start_IT(&htim2); in the latter, at the end of the song, we block timer 2 operation with HAL_TIM_Base_Stop_IT(&htim2), after setting its autoreload value to 10 to let it react fast when a new snap of fingers is done (__HAL_TIM_SET_AUTORELOAD(&htim2, 10)).

Professor comments: