

" به نام خدا "

مروارید لعل نور 9623097

گزارش تمرین سری سوم درس میکرو

در ابتدا برای بدست آوردن مقادیر سینوسی از تابع \sin که در کتابخانه `math` قرار دارد استفاده کرده و مقادیر بدست آمده را در آرایه ای به همین نام (`sine[NUMBER_OF_SAMPLES]` ، تعداد نمونه ها در ابتدای کد برابر 50 تعریف شده است) ذخیره کرده و آنها را به کانال 1 DAC می‌دهیم و سپس تایمر ها را فعال می‌کنیم . تایمر 2 دارای `update event` برابر با 25000 است زیرا :

$$\text{Update event} = 72 * 10^9 / (72 * 10000 * 4)$$

و تایمر 6 دارای `PSC = 7200` و `period = 10000` است .

آرایه ی `note_length` میزان کشیدگی (زمان اجرا) هر نوت را در خود دارد و طول آن 31 است (26 نوت + سکوت های مابین) که در این آرایه `duration` مقدار زمانی است که برای یک میزان کامل در نظر گرفتیم و برای نت های مختلف با توجه به زمان آن نوت به 2 یا 4 یا 8 تقسیم شده است .

یک آرایه دو بعدی `31*7` از نت ها در اکتاو های مختلف داریم که `converted_notes` نام دارد و حاوی مقدار تبدیل شده عدد هر نوت در جدول به فرکانس است . (مقادیر 1 تعیین کننده سکوت میان نوت ها هستند)

آرایه ی `converted_note` هم شامل نوت هایی است که قرار است اجرا شوند و از آنجایی که اکتاو ابتدایی به صورت پیش فرض 4 است ، مقادیر نوت های این اکتاو به عنوان مقدار اولیه در آن قرار داده شده است .

متغیر `adc_reset` برای ای تعریف شده است که پایان اجرای 31 نوت پس سر هم را به ما نشان دهد و با توجه به مقداری که پتانسیومتر برای شماره اکتاو بعدی تعیین کرده است ، نوت های اکتاو بعدی را اجرا کند .

در حلقه `while` همواره چک می‌کنیم که اگر خواندن نوت ها تمام شده بود و فلگ `adc_reset` فعال شده بود شماره اکتاو جدید چه خواهد بود . برای این کار مقدار ADC که با تغییر پتانسیومتر تعیین شده را می‌خوانیم و از آنجا که بیشترین مقدار آن میتواند 4095 باشد به این شکل آنرا به بازه ی 0 تا 7 `map` می‌کنیم :

```
ADC_VALUE = HAL_ADC_GetValue(&hadc1);
```

```
octave_number = (uint8_t) (ADC_VALUE*7/4094);
```

و سپس با توجه به این مقدار نوت های مربوطه را درون آرایه ی `converted_note` قرار می‌دهیم تا برای اجرا استفاده شود.

برای اجرای نوت ها تایمر 6 همواره تابع `HAL_TIM_PeriodElapsedCallback` را اجرا میکند و به ترتیب و یکی یکی نوت ها را اجرا میکند به این صورت که مقدار `ARR` تایمر 6 زمان اجرای نوت و `ARR` تایمر 2 مقدار فرکانس آن نوت را در خود ذخیره کرده و در انتها نیز چک می‌کند که آیا به نوت 31 ام رسیده یا خیر ، اگر رسیده بود اکتاو جدید را از نوت 0 ام شروع به اجرا میکند.

توضیحاتی در رابطه با فیلم پیاده سازی :

در قسمت اول ویدیو مشاهده میشود که با تغییر پتانسیو متر مربوط به تعیین اکتاو ها اهنگ مورد نظر در شکل های متفاوتی (بم تر یا زیر تر) اجرا میشود ، در قسمت بعد با تغییر پتانسیومتر دیگری میزای بلندی صدا تغییر میکند .

همچنین خروجی میکرو بر روی اسیلوسکوپ نمایش داده شده است که تغییر فرکانس ها نیز مشهود باشد .

با تشکر