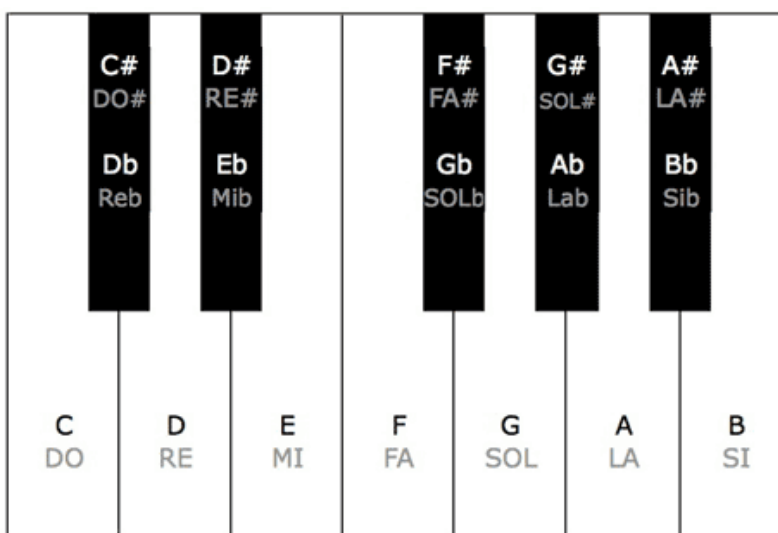




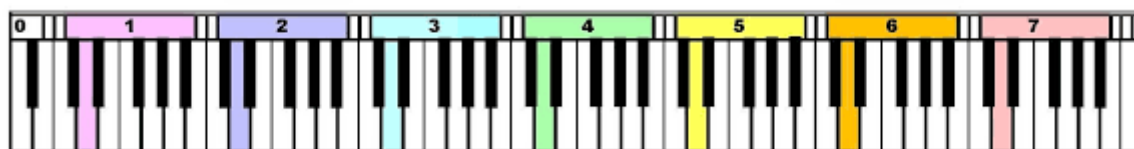
نواختن موسیقی با استفاده از نت های پیانو

هدف از انجام این آزمایش آشنایی با واحد های DAC و ADC میکروکنترلر است .

یک پیانو می تواند ۸۸ کلایه داشته باشد ، متشکل از ۸ اکتاو که هر اکتاو شامل ۷ عدد کلایه ی سفید و ۵ عدد کلایه ی مشکی است .



یک پیانوی ۸۸ کلایه ای شکلی مانند زیر دارد :



هر کدام از کلایه ها دارای فرکانس خاصی هستند که از طریق جدول و فرمول زیر قابل محاسبه است :

Octave #	MIDI Note Numbers											
	C	C#	D	D#	E	F	F#	G	G#	A	A#	B
-1	0	1	2	3	4	5	6	7	8	9	10	11
0	12	13	14	15	16	17	18	19	20	21	22	23
1	24	25	26	27	28	29	30	31	32	33	34	35
2	36	37	38	39	40	41	42	43	44	45	46	47
3	48	49	50	51	52	53	54	55	56	57	58	59
4	60	61	62	63	64	65	66	67	68	69	70	71
5	72	73	74	75	76	77	78	79	80	81	82	83
6	84	85	86	87	88	89	90	91	92	93	94	95
7	96	97	98	99	100	101	102	103	104	105	106	107
8	108	109	110	111	112	113	114	115	116	117	118	119
9	120	121	122	123	124	125	126	127				

می دانیم کلاویه ی A4 دارای فرکانس ۴۴۰ هرتز است و برای محاسبه ی هر کدام دیگر از کلاویه ها بایست از

$$f = 440 \times 2^{(n - 69) / 12}$$

استفاده نمود. علامت # نشانه کلاویه ی مشکی سمت راست یک کلاویه ی سفید است .

می دانید که بعد از آمدن trigger برای واحد DAC ، مقدار رجیستر DAC_DHR12R1 بر روی رجیستر DOR قرار می گیرد و از روی این رجیستر مقدار آنالوگ تولید می شود . ولتاژ ایجاد شده در خروجی آنالوگ برابر است با

$$DAC_{output} = V_{ref} * \frac{DOR}{4096}$$

برای ایجاد نت A4 در خروجی یک DAC بایست نقاطی از یک سینوسی را نمونه برداری شده به رجیستر DAC_DHR12R1 بدهید که به صورت دقت ۱۲ بیتی و راست چین است و مربوط به خروجی شماره ی ۱ از DAC می باشد . برای نمونه برداری از یک سینوسی بایست از فرمول زیر در پایتون یا matlab استفاده نمایید .

$$y_{SineDigital}(i) = (\sin(2\pi i/m) + 1) \times (2^n - 1) \quad i = 0, 1, \dots, m - 1$$

که m تعداد نقاط نمونه برداری است و n دقت DAC شماست که مقدار آن ۱۲ می باشد .

می خواهیم موسیقی Jesus loves me که یک آهنگ تک کلاویه ای است بر روی میکرو اجرا کنیم . sheet music آن به صورت زیر است :

Happy Birthday

Beginner Piano

Arrangement: Betacoustic

Words and Music by
Mildred J. Hill and Patty Hill



که برای اجرای آن در اکتاو میانی بایست بنا بر sheet بالا به ترتیب زیر اجرا شود :

C4 C4 D4 C4 F4 E4 ; happy birthday to you

C4 C4 D4 C4 G4 F4 ; happy birthday to you

C4 C4 A4 F4 E4 D4; happy birthday [your name]

C4 C4 A4 F4 G4 F4; happy birthday to you

برای پیاده سازی قطعه ای مانند بالا بایست سرعت اجرای هر نقطه ی سینوسی نمونه برداری شده را با توجه به فرکانس یک

کلاویه ، با استفاده از سیگنال trigger تغییر دهید . مثلا اگر $m=50$ آنگاه برای اجرای A4 بایست فرکانس تریگر $440*50$ هرتز باشد .

برای این آزمایش از تریگر update event از تایمر ۸ استفاده کنید .

مساله ی دیگر زمان اجرای هر نوت است . در sheet می توانید نوت های تو خالی را به مدت ۰.۵ ثانیه و مابقی را به مدت

۰.۲۵ ثانیه اجرا نمایید .(برای راهنمایی بیشتر نوت های تو خالی در انتهای هر خط در بالا قرار دارند)

قسمت اول پروژه :

برای انجام قسمت اول می بایست قطعه ی بالا را بر روی میکرو اجرا کنید . به این صورت که مقادیر سینوسی را بر روی یک آرایه uint16_t ذخیره نمایید و هر با استفاده از تایمر ۸ فاصله ی trigger بین آن ها را تغییر دهید تا فرکانس سینوسی ها تغییر کند سپس برای تعییت زمان اجرای هر نوت می توانید برای راحتی ایجاد زمان های در حد ثانیه از تایمر ۲ که دارای رجیستر ARR ۳۲ بیتی است استفاده کنید . برای تغییر زمان اجرای هر نوت هم می توانید مقدار رجیستر ARR را تغییر دهید. بعد از اتمام زمان اجرای هر نوت (یعنی بعد از اینکه update event در تایمر ۲ اتفاق افتاد و وقفه ی آن اجرا شد) ، بایست در سرویس دهنده ی وقفه تایمر ۲ ، مقدار ARR تایمر ۸ برای ایجاد فرکانس نوت بعدی را قرار دهید . همچنین مقدار ARR تایمر ۲ را هم برای تغییر زمان اجرای نوت بعدی تغییر دهید .

در نتیجه سه آرایه به سبب تعداد نوت های اجرایی در قطعه بالا خواهید داشت . یکی برای تعیین سینوسی نمونه برداری شده ، یکی مقدار رجیستر ARR تایمر ۲ برای زمان اجرای هر نوت و دیگری مقدار رجیستر ARR تایمر ۸ برای تعیین فرکانس تریگر برای تعیین فرکانس سینوسی که با هر بار اتفاق افتادن وقفه ی update در تایمر ۲ اندیس آرایه یکی به جلو می رود .

قسمت دوم پروژه :

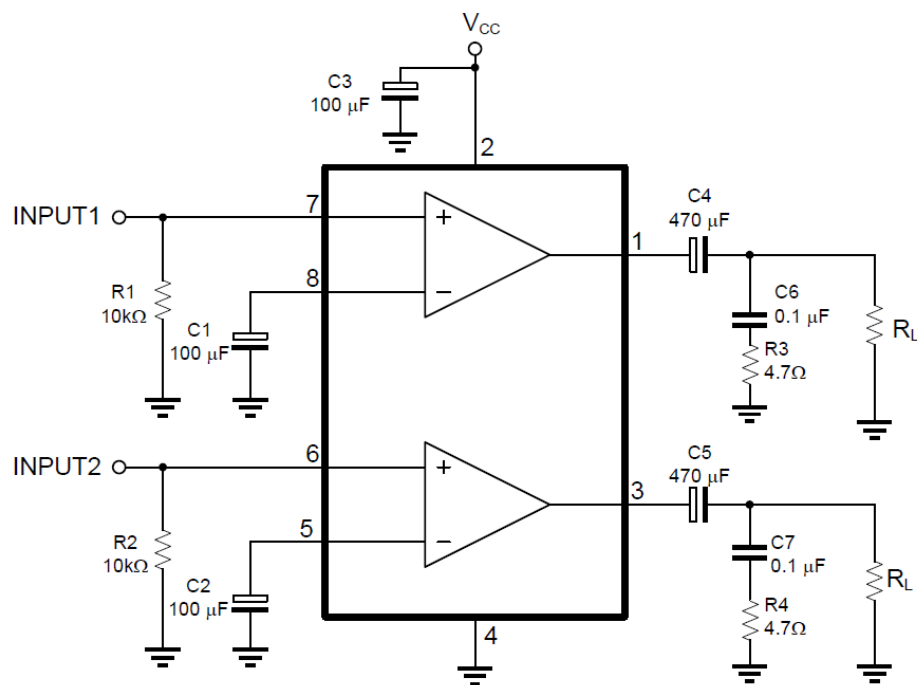
یک مقاومت متغیر متصل به کانال ۱ از ADC داریم. ADC را به صورت دقت ۱۲ بیتی و راست چین تنظیم کنید .وظیفه ی مقاومت، جابه جایی قطعه ی موسیقی بالا بین اکتاو ها است . یعنی ما مقادیر بین صفر و ۳.۳ ولت را به ۷ قسمت تقسیم می کنیم و با توجه به این که مقدار ولتاژ بر روی کانال ۱ در چه بازه ای قرار دارد یکی از اکتاو ها را اجرا می کنیم . به طور مثال اگر مقدار ولتاژ بین 0 و 3.3/7 قرار داشته باشد ، موسیقی بر روی اکتاو ۱ اجرا شود . در این صورت تنها تغییر بر روی آرایه مقادیر ARR خواهد بود که یک آرایه ی دو بعدی با سبب [7][notes] خواهیم داشت . که می بایست مقادیر فرکانس ها برای هر نوت در قطعه ی موسیقی در آن قرار بگیرد و توسط تابعی مقدار ARR تایمر ۸ تغییر کند .

مقاومت متغیر دیگری نیز داریم که صدای موسیقی را کم و زیاد می کند برای این کار بایست مقدار کانال دو از ADC خوانده شود و چون مقدار دقت آن بر روی ۱۲ بیت تنظیم شده است ، مقدار داده ای که بر روی DAC_DHR12R1 قرار میگیرد به صورت زیر تغییر می کند :

$$ySineDigital(i) /= ADC_DR/4096$$

یعنی مقدار آرایه ی سینوسی ها ی نمونه برداری شده در یک ضربی بین ۰ و ۱ ضرب می شوند . توضیح دیگر آن که قسمت اول و دوم پروژه را با هم انجام دهید ، علت جدا کردن بین قسمت اول و دوم فهم بهتر مساله است .

در صورت انجام عملی پروژه برای جلوگیری از خرابی میکرو از یک amplifier IC استفاده کنید مانند TDA2822 که در زیر می توانید به جای R_L یک بلندگو قرار دهید .



فایل‌های تحویلی:

۱. پروژه ی نرم افزار keil به همراه cubeMX
۲. گزارش کار در فرمت pdf
۳. در صورت کار عملی (فیلم کوتاهی از پروژه ی در حال اجرا) (دارای نمره تشویقی)

موفق باشید .