آزمایش ششم

# مقدمه

در این آزمایش به کار با موتور DCدارای رمزگذار چرخشی (Rotary Encoder) و نوسان‌ساز (Oscillator) خواهید پرداخت و در این حین به یادگیری مدولاسیون عرض پالس (PWM)، وقفه خارجی (External Interrupt) و همچنین روش دیگری برای هم‌زمان‌سازی خواهید پرداخت. همچنین از GPIO، LCD و Timer که در آزمایش‌های قبل یاد گرفتید نیز استفاده خواهید کرد. کتابخانه‌های مورد نیاز این آزمایش در یک فایل Zip در صفحه درس قرار داده شده است.

* برای راه‌اندازی مدولاسیون عرض پالس (PWM) که با آن در درس ریزپردازنده آشنا شده‌اید، می‌بایست از دستورات زیر استفاده کنید:

#include "pwm.h"

...

void PWM\_Init(cycleTime); //cycleTime is PWM period

void PWM\_SetDutyCycle(pin, dutyCycle); //dutyCycle is proportion of ON time

void PWM\_Start(pin); //pin is PWM\_1 ... PWM\_6.

void PWM\_Stop(pin);

* برای راه‌اندازی قسمت Encoder موتور، به Timer و External Interrupt نیاز دارید. Encoder یک صفحه شطرنجی است که بازخورد حرکت موتور را به شکل پالس مربعی به ما می‌دهد. از Encoder برای تنظیم دقیق سرعت موتور استفاده می‌شود. به کمک وقفه خارجی فاصله هر پله Encoder را متوجه می‌شوید و به کمک Timer فاصله را اندازه‌گیری می‌کنید. یک Oscillator نیز به میکرو متصل شده است که به صورت منظم با فرکانس ثابت پالس مربعی تولید می‌کند. روش اندازه‌گیری زمان پالس Oscillator مشابه Encoder می‌باشد.
* برای استفاده از وقفه خارجی (External Interrupt) که با آن در درس ریزپردازنده آشنا شده‌اید، به تابع زیر نیاز خواهید داشت:

#include "extintr.h"

...

void EINT\_AttachInterrupt(intNumber, funPtr, intMode);

//intNumber is EINT\_0 ... EINT\_3.

//funPtr is “void->void” function pointer for CallBack user custom function.

//intMode is LOW, HIGH, FALLING, RISING.

* دومین روش برای هم‌زمان سازی کارها در میکروکنترلرها استفاده از وقفه‌هایی مانند وقفه Timer و وقفه خارجی است. کارهایی که قابلیت انتقال به تابع روتین وقفه را دارند، می‌بایست به روتین منتقل شوند و سایر کارها مانند گذشته در حلقه بی‌نهایت قرار می‌گیرند. گاهی تمام کارها را می‌شود به روتین انتقال داد و می‌بایست در تابع اصلی فقط یک حلقه خالی قرار داد. توجه کنید که کارهای زمان‌بر بهتر است در روتین‌ها قرار نگیرند و با متغیرهایی مشابه پرچم (flag) به حلقه اصلی دستور اجرا را داد. به مثال زیر توجه کنید.

void isr1(){

Task1();

}

void isr2(){

Task2();

}

void isr4(){

flag = 1; // for Task4();

}

void main(){

attach\_interrupt(isr1());

attach\_interrupt(isr2());

attach\_interrupt(isr4());

while(1){

Task3();

if (flag){

Task4();

flag = 0;

}

}

}

* برای استفاده از کتابخانه‌ها، هنگام انتخاب Run-Time Env. فقط CMSIS-CORE را انتخاب کنید. تمام فایل‌ها را در محیط ویندوز در کنار فایل main.c خود قرار دهید. فایل‌های .c و .s را در محیط برنامه keil به کنار فایل main.c واقع در Source Group 1 اضافه کنید. (کلیک راست روی فولدر، Add existing Files…)
* فراموش نکنید که در ابتدای برنامه خود تابع SystemInit() را فراخوانی کنید.
* حتما فایل‌های کتابخانه‌ها را مطالعه کنید و با نحوه کار آن‌ها آشنا شوید.
* **دقت کنید که در این آزمایش استفاده از توابع Delay و توابع مشابه آن ممنوع بوده و برای زمان‌بندی می‌بایست از Timer استفاده کنید. در صورت استفاده، نمره قسمت مربوطه صفر در نظر گرفته می‌شود!**

# پیش‌آزمایش

* نحوه کار مدولاسیون عرض پالس در میکروکنترلرها به چه صورت است؟
* نحوه کار وقفه‌ها در میکروکنترلرها به چه صورت است؟
* مزیت‌های استفاده از وقفه‌های میکروکنترلر چیست؟
* تفاوت چهار حالت وقفه خارجی (Rising، Falling، High، Low) چیست؟
* در مورد مشخصه‌های یک پالس (Period، Frequency، DutyCycle، Offset) تحقیق کنید.
* چرا موتور DC را مستقیم به میکروکنترلر متصل نمی‌کنند و از یک مدار واسط مانند Driver استفاده می‌شود؟

# صورت آزمایش

## اتصالات

اتصالات روی بورد در این آزمایش به شرح زیر است:

* اتصال پایه‌های LCD‌ (RS,RW,E) به P0.0..P0.2
* اتصال پایه‌های LCD‌ DB4..DB7 به P0.4..P0.7
* اتصال پایه‌های موتور (A,B,EN) به P2.0..P2.2 (PWM1.. PWM3)
* اتصال پایه‌ی خروجی Encoder موتور به P2.10 (EINT0)
* اتصال پایه‌ی خروجی نوسان‌ساز به P2.11 (EINT1)

## قسمت اول

تابعی بنویسید که بتواند سرعت موتور را با دقت ۸ بیت (255) به حرکت در بیاورد. جهت حرکت اهمیت ندارد و همچنین تابع نیازی به حرکت در دو جهت ندارد. این تابع را با ۵۰ درصد سرعت فراخوانی کنید و برنامه خود را تکمیل کنید. دقت کنید که مدار Driver این موتور دارای پایه Enable (Active High) می‌باشد و برای راه‌اندازی موتور می‌بایست آن را مقداردهی کنید.

برنامه نوشته شده و تصویر موتور در حال چرخش را در گزارش خود قرار دهید.

## قسمت دوم

برنامه‌ای بنویسید که بتواند فرکانس دقیق نوسان‌ساز متصل به میکرو را در هر لحظه اندازه‌گیری کند. می‌دانیم که فرکانس برابر با معکوس زمان می‌باشد. فرکانس را هر لحظه در LCD نمایش دهید. در خط دوم نیز شماره گروه خود را بنویسید.

برنامه نوشته شده و تصویر خروجی LCD را در گزارش خود قرار دهید.

## قسمت چهارم (۵۰ درصد امتیازی)

برنامه‌ای بنویسید که با الگوریتمی به صورت خودکار با کاهش و افزایش سرعت موتور و به کمک خواندن فرکانس خروجی از Encoder موتور، آن را به سرعت هدف که برابر با **۳۰۰۰ دور بر دقیقه (RPM)** می‌باشد، برساند. در نظر داشته باشید که صفحه Encoder این موتور، دارای ۴۰ پله می‌باشد. سرعت هدف و سرعت لحظه‌ای موتور را در دو خط مجزا در LCD نشان دهید. (موتور باید از حداکثر سرعت شروع کند.)

برنامه نوشته شده، محاسبات انجام شده، توضیح الگوریتم استفاده شده و یک فیلم کوتاه از لحظه شروع تا رسیدن به سرعت هدف را در گزارش خود قرار دهید.

موفق باشید