به نام خدا

در این جلسه نیز مبحث PWM را دنبال مي‌کنیم . براي درك مطالب این جلسه حتماً باید جلسه قبل را مطالعه کرده باشید . در این جلسه به آي‌سي L۲۹۸ نیز اشاراتي شده است که دوستان مي‌توانند جهت یاداوري ، بخش مربوط به L۲۹۸ در جلسه‌ي ۱۷ را نیز مرور کنند .

ابتدا با توابعي که براي ایجاد وقفه در اجراي دستورات برنامه توسط CodeVision براي کاربران در نظر گرفته شده آشنا مي‌شویم.

همان‌طور که در جلسه پیش دیدیم ، در قسمت‌هایي از برنامه ممکن است نیاز پیدا کنیم تا براي لحظاتي روند اجراي دستورات را متوقف کنیم . CodeVision براي این کار توابعي را از پیش تنظیم کرده است . (در مورد مبحث توابع در زبان C در آینده مفصل توضیح خواهیم داد)

Delay دستوری برای توقف یا تاخیر

براي ایجاد تاخیر در روند اجراي دستورات ، CodeVision دو تابع زیر را در اختیار ما قرار داده است.

```
delay_ms( );
 delay_us( );
```

تابع delay_ms() براي ایجاد تاخیرهایي در حد میلي ثانیه به کار مي‌رود. در داخل پرانتز، یك عدد صحیح مثبت مي‌نویسیم که نشان دهنده اندازه تاخیر مورد نیز ما بر حسب میلي ثانیه است. به بیان ساده‌تر، مثلاً اگر داخل پرانتز عدد ۱۰۰را بنویسیم، روند اجراي برنامه به اندازه ۱۰۰میلي ثانیه در همان خط متوقف خواهد شد.
تابع delay_us() براي ایجاد تاخیرهایي در حد میکروثانیه به کار مي‌رود. نحوه‌ي استفاده از آن دقیقاً مانند delay_ms() است.
به عنوان یك مثال عملي ، همان برنامه ایجاد PWM ۲.۵ ولت را با استفاده از توابع delay بازنویسي مي‌کنیم.

```
while(۱)
{
PORTB.٤=۱;
delay_ms(٥);   // ٥ milliseconds delay
PORTB.٤=۰;
delay_ms(٥);   // ٥ milliseconds delay
}
```

تنها نکته بسیار مهم در استفاده از توابع delay اضافه کردن #include<delay.h> دقیقا بعد از #include<mega16.h> در ابتدای برنامه ی شماست بدون اضافه کردن #include<delay.h> شما نمی توانید از دستورات delay استفاده کنید . دستور #include به معنی اضافه کردن و delay.h

به معنی دستورات و امکانات فایل delay.h است که در این خصوص در آینده بیشتر صحبت می کنیم ولی بیایید در حال حاضر از بحث اصلی منحرف نشویم.

دقت کنید که این دستور نیازي به « ; » ندارد !

با آموختن تابع delay ، مي‌توانيد هر نسبت ولتاژ يا PWM كه مي‌خواهيد بر روي پايه‌هاي خروجي ايجاد كنيد . به عبارت دیگر اگر نیاز به ۲.۵ ولت برای کمرنگ روشن کردن یک چراغ داشتید کافی است یک pwm ۵۰ درصد روی پایه ی متصل به چراغ بدهید در آن صورت طبق مطالب درس قبل رفتار چراغ شبیه به این است که ۲.۵ ولت به آن وصل شده.

همانطور كه مي‌دانيد موتورهاي متعرفي كه براي ساخت ربات‌ها استفاده مي‌شود ، ممكن است ولتاژهاي كاري مختلفي داشته باشند (مثلاً ۱۲ولت، ۲٤ ولت، ٦ ولت و (... و براي راه‌اندازي آن‌ها بايد از درايورهاي موتور مثل L۲۹۸ استفاده كنيم. سوالي كه ممكن است پيش آيد اين است كه وقتي ما ميكروكنترلر را به درايورهاي موتور(مثل L۲۹۸) وصل مي‌كنيم و از تكنيك PWM براي كنترل سرعت موتور استفاده مي‌كنيم، چه وضعيتي پيش مي‌آيد؟ مثلاً وقتي ما PWMمربوط به ولتاژ ۲.۵ولت را توليد مي‌كنيم، درايور ما چه عكس العملي نشان مي‌دهد؟ آيا ولتاژ ۲.۵ ولت بر روي پايه‌هاي موتور قرار مي‌گيرد؟

براي پاسخ به اين سوال بايد به ساختار PWM دقت كنيم، ما وقتي PWM مربوط به ۲.۵ ولت را توليد مي‌كنيم، در حقيقت سطح ولتاژ خروجي را با فواصل زماني برابر ۰ و ۱ مي‌كنيم، پس اگر اين خروجي را، به ورودي L۲۹۸ وصل كنيم(مثلاً پايه‌ي ۷)، L۲۹۸نيز موتور را با همين الگو كنترل مي‌كند و ولتاژي كه به موتور مي‌دهد را ۰ و ۱ مي‌كند. و همانطور كه مي‌دانيد، L۲۹۸هر ولتاژي كه بر روي پايه‌ي شماره‌ي ٤ آن قرار گرفته باشد را بر روي موتور قرار مي‌دهد (اگر ولتاژ كاري موتور ۱۲ ولت باشد، بايد اين پايه به ۱۲ ولت متصل شود) . پس جواب سوال بالا منفي است ! وقتي ما PWMمربوط به ۲.۵ ولت را توليد مي‌كنيم، در حقيقت سطح ولتاژ خروجي در ۵۰ درصد زمان ۱ و بقيه‌ي زمان ۰ است. پس اگر همان طور كه در بالا اشاره شد، اين PWMبه درايوري مثل L۲۹۸داده شود، و ولتاژ پايه‌ي ٤ آن ۱۲ولت باشد، درايور، ولتاژ ٦ ولت را به موتور مي‌دهد. در نتيجه اهميتي ندارد چه ولتاژي بر روي پايه‌ي ٤ L۲۹۸ قرار گرفته باشد، وقتي كه ما PWM مربوط به ۲.۵ولت را توليد مي‌كنيم، درايور ولتاژي كه به موتور مي‌دهد را ۵۰ درصد مي‌كند. در نتيجه بهتر است از اين به بعد به جاي آن كه بگوييم PWM مربوط به ۲.۵ ولت، بگوييم PWM پنجاه درصد .يا به جاي PWM مربوط به ۱ ولت، بگوييم PWM بيست درصد .

PWM در ميكروكنترلرهاي AVR

انجام تنظيمات اوليه براي استفاده از PWM براي راه اندازي موتور در ميكروكنترلرهاي AVR كمي پيچيده است، اما در اينجا هم CodeWizard به كمك ما آمده است و كار را كمي ساده‌تر كرده است. ما در جلسه آينده بخشي از تنظيمات CodeWizard را بدون توضيح مطرح مي‌نماييم، زيرا توضيح هر بخش از آن نيازمند مقدمات مفصلي است و تاثير چنداني هم در روند كار ما ندارد، اما به دوستاني كه مي‌خواهند ميكروكنترلر را كاملاً حرفه‌اي دنبال كنند، پيشنهاد مي‌كنم از منابعي كه قبلاً معرفي شده است، مطالب را تكميل كنند .

به هر حال دوستان عزيز با انجام اين تنظيمات اوليه مختصر در CodeWizard، مي‌توانند از الگويي به مراتب ساده‌تر از آنچه تا به حال آموخته‌ايم، براي ايجاد PWM براي هدايت موتورهاي ربات استفاده نمايند.

در ميكروكنترلرهاي خانواده‌ي AVR ، نيازي نيست در هربار استفاده از PWM، چندين خط برنامه بنويسيم. در ATmega١٦ چهار پايه مشخص از آي سي به اين موضوع اختصاص داده شده

است. يعني اين چهارپايه علاوه بر كاربردهاي معمولي خود، اين قابليت را دارند كه در مواقع لزوم براي توليد PWM استفاده شوند.

حال سوال اينجاست كه اين چهارپايه چه تفاوتي با بقيه پايه‌هاي خروجي آي‌سي دارند كه آن‌ها را از ساير پايه‌هاي خروجي ميكروكنترلر متمايز مي‌سازد ؟

براي اين چهارپايه نيازي به اجراي الگويي كه تا به حال براي ايجاد PWM فراگرفته‌ايد نيست. در اين روش، فقط شما بايد يك عدد صحيح بين ۰ تا ۲۵۵ انتخاب كنيد، و طبق الگوي زير آن را در برنامه‌ي خود بنويسيد .

يك عدد صحيح بين ۰ تا ۲۵۵ = نام رجيستر مربوطه;

اين عدد، بيانگر توان PWMشماست ، و شما توان PWM مورد نياز خود را با اين عدد مشخص مي‌كنيد. كه ۲۵۵ بالاترين توان و مربوط بهPWM ۱۰۰درصداست، و ۰ پايين‌ترين توان و مربوط به PWM ۰درصد است . به عنوان مثال اگر اين عدد را ۱۲۸ قرار دهيد، همان PWM ۵۰ درصد را ايجاد كرده‌ايد. يا مثلا اگر اين عدد ۵۱ باشد، PWM ۲۰درصد بر روي پايه قرار داده‌ايد.

رجيسترهاي مربوط به اين ٤ پايه

همانطور كه مي‌دانيد، براي پايه‌هايي كه در CodeWizard به صورت خروجي تعريف شده‌اند، رجيستري به نام «PORTx» وجود دارد كه هر مقداري در اين رجيستر قرار داده شود، مقدار پايه‌هاي خروجي متناظر با آن رجيستر را مشخص مي‌كند. (رجوع به جلسه‌ي ٢٤،تعريف رجيستر PORTx)

در اين جلسه با ٤ رجيستر ديگر آشنا مي‌شويم، كه وقتي تنظيمات مربوط به PWM موتور در CodeWizard را انجام دهيم، هر مقداري كه در آن‌ها ريخته شود، توان PWM پايه‌ي متناظر را مشخص مي‌كنند .

اين رجيسترها OCR۰، OCR۱AL ، OCR۱BL و OCR۲ نام دارند كه به ترتيب، متناظر پايه‌هاي PB.۳، PD.۵، PD.٤و PD.۷ هستند.

پس مثلاً اگر در بخشي از برنامه‌ي خود بنويسيم :

```
OCR۰=۱۲۷;
```

در حقيقت بر روي پايه PB.۳ ميكروكنترلر ، PWM ۵۰ درصد به وجود آورده‌ايم.

به مثال‌هاي ديگري توجه كنيد : (توضيح هر دستور در جلوي دستور و بعد از // آورده شده است)

```
OCR١AL=٥١; // ٢٠% Duty Cycle on PD.٥
OCR١BL=٢٥٥; //١٠٠% Duty Cycle on PD.٤

OCR٢=٠;  //٠% Duty Cycle on PD.٧
```

در جلسه‌ي آينده، در مورد نحوه انجام تنظيمات اوليه جهت توليد PWM در CodeWizard توضيح خواهيم داد.