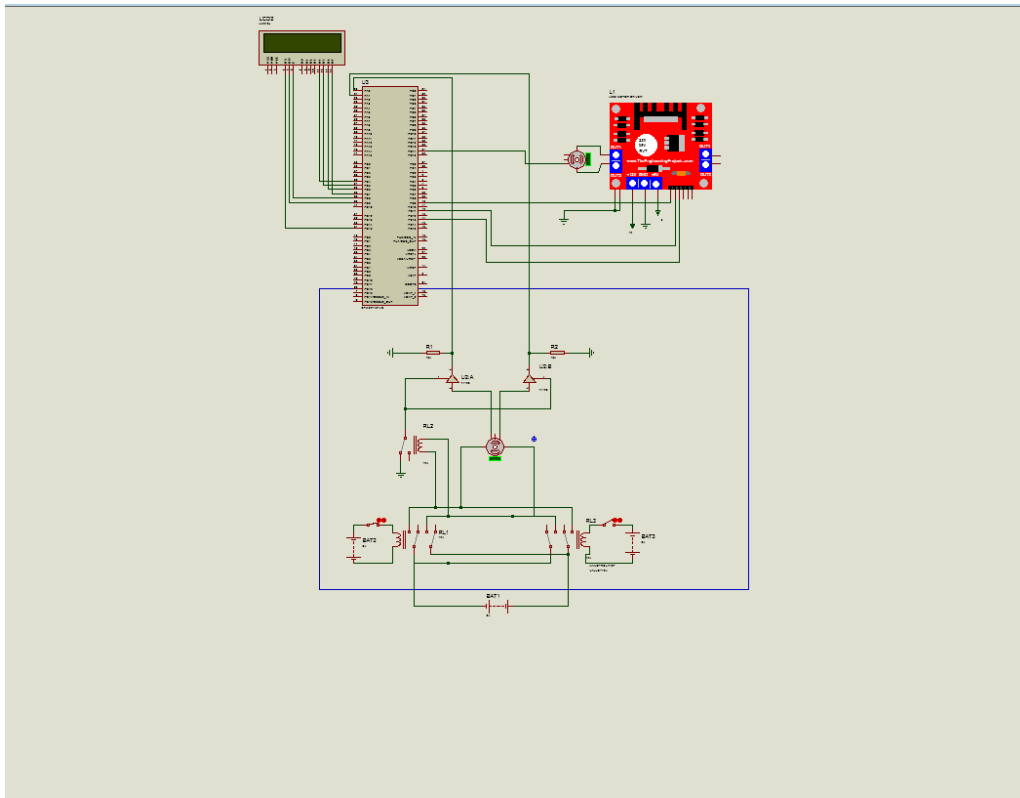


" به نام خدا "

مروارید لعل نور 9623097

تمرین سری دوم درس سیستم های ریزپردازنده

در انجام این تمرین سعی شده از تایمر های متفاوت 1 و 2 و 3 استفاده کردیم . بین ها همانطور که در پروژه پروتئوس هم قابل مشاهده است به صورت زیر به یکدیگر وصل شده اند و فایل .hex مربوط به پروژه keil هم برای شبیه سازی در پروتئوس استفاده شده است .



منطق کلی ای که مورد استفاده قرار دادم این بود که بازه تغییرات کانتر تایمر 2 را از 30 تا 60 در نظر گرفتم که برای تغییر سرعت نیاز به چرخاندن زیاد روتاری انکودر نباشد و سپس با استفاده از قطعه کد پایین آنرا بین 200 تا 920 map کردم :

```
if (is_cw == 1)
{
    new_cnt = 200 + (TIM2->CNT - 30)*24;
    TIM1->CCR1 = new_cnt ;
}
else
{
    new_cnt = 200 - (TIM2->CNT - 30)*24;
    TIM1->CCR1 = new_cnt ;
}
```

و برای کنترل جهت چرخش با استفاده از اینترپیت تایمر 2 در هنگام تمام شدن پالس تعیین کردم که اگر مقدار کانتر از 30 بیشتر بود ساعت گردد و در غیر این صورت پاد ساعتگرد حرکت کند .

همچنین برای آنکه سرریز و زیر ریز رخ ندهد از قطعه کد زیر استفاده کردم که اجازه نمیدهد در بازه ی 0 تا 60 مقدار کانتر تایمر از 8 کمتر و یا از 52 بیشتر شود :

```
* USER CODE BEGIN 3 */
if(TIM2->CNT > 52)
    TIM2->CNT = 52;
if(TIM2->CNT < 8)
    TIM2->CNT = 8;
```

برای نمایش مقدار دور موتور روی ال سی دی نیز از یک تایمر با فرکانس 2 هرتز (0.5 ثانیه) استفاده شده که هر 0.5 ثانیه ال سی را آپدیت میکند و مقدار محاسبه شده برای rpm را نشان میدهد . (ساعتگرد یا پاد ساعتگرد بودن چرخش به کمک ضرب یک عدد منفی در مقدار دور موتور روی ال سی دی نمایش داده میشود)

برای محاسبه دور موتور با توجه به آنکه با یک دور موتور کانتر مربوطه 96 تا تغییر میکرد (با پیاده سازی بدست آمده است) از تایمر 4 برای کپچر و انجام این محاسبات استفاده شده :

```
void HAL_TIM_IC_CaptureCallback(TIM_HandleTypeDef *htim)
{
    if(htim->Instance == TIM4)
    {
        captured = __HAL_TIM_GetCompare(&htim4, TIM_CHANNEL_3);
        TIM4->CNT = 0;
        rpm = 100000.0 / (captured * 96.0);
    }
}
```

برای تعویض جهت موتور در وسط بازه (وقتی به 30 برسد و سرعتش کم شود) از تایمر 2 به این صورت استفاده کردیم که با رسیدن کانتر به 30 با تبدیل ورودی موتور (که به پین e11 و e13 میکرو متصل اند) از 0 - 1 به 1 - 0 جهت آنرا تغییر دهد و فلگ مربوط به ساعتگرد بودن را 1 کند :

```
void HAL_TIM_PWM_PulseFinishedCallback(TIM_HandleTypeDef *htim)
{
    if(htim->Instance == TIM2)
    {
        if(TIM2->CNT > 30)
        {
            is_cw = 1;
            HAL_GPIO_WritePin(GPIOE, GPIO_PIN_11, GPIO_PIN_SET);
            HAL_GPIO_WritePin(GPIOE, GPIO_PIN_13, GPIO_PIN_RESET);
        }
        else
        {
            is_cw = 0;
            HAL_GPIO_WritePin(GPIOE, GPIO_PIN_13, GPIO_PIN_SET);
            HAL_GPIO_WritePin(GPIOE, GPIO_PIN_11, GPIO_PIN_RESET);
        }
    }
}
```

پیاده سازی عملی این برنامه انجام شده و ویدئو مربوطه ضمیمه شده است .

با تشکر