بررسی و راه اندازی روتاری انکودر با میکروکنترلر STM32

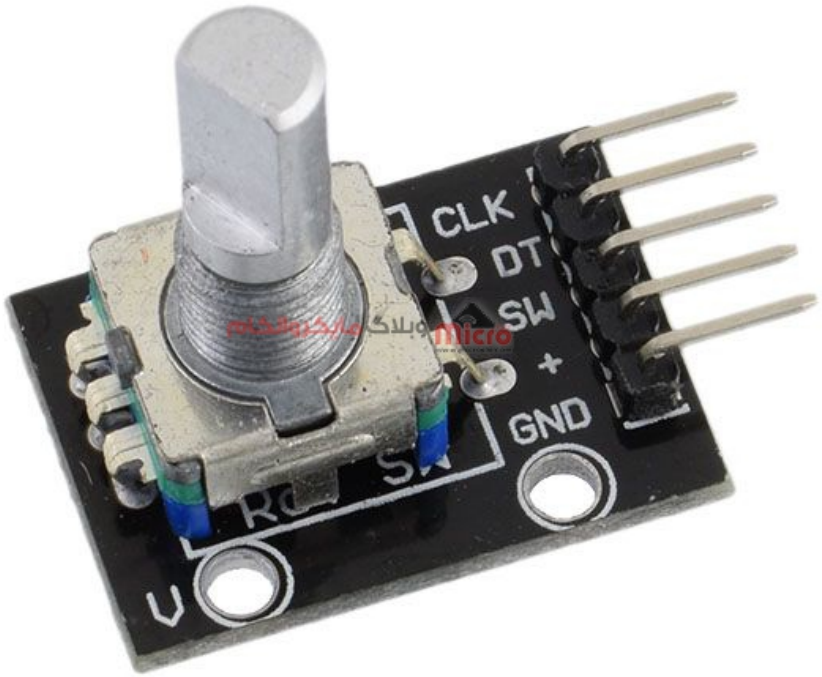
تاریخ انتشار۱۴ مرداد, ۱۴۰۰ توسط آرش فتاحی

با عرض سلام خدمت همراهان سایت مایکروالکام. روتاری انکودر یک قطعه الکترومکانیکی است که با چرخاندن آن، سیگنال‌ها و پالس‌هایی توسط آن تولید می‌شود که می‌توان از طریق آن‌ها، جهت و تعداد چرخش روتاری انکودر را تعیین و مشخص نمود.به دلیل استحکام و کنترل دیجیتالیِ خوبِ روتاری انکودرها، از آن‌ها در بسیار از تجهیزات الکترونیکی همچون رباتیک، ماشین‌های CNC، پرینترها و پروژه‌های الکترونیکی استفاده می‌گردد. در این مطلب نحوه راه اندازی روتاری انکودر با میکروکنترلر STM32 مورد بررسی قرار خواهد گرفت. پس با من تا انتهای مطلب همراه

باشید. همچنین میتونید سایر مطالب من رو از این قسمت مطالعه کنید.

معرفی انواع روتاری انکودرها

یکی از خوبی‌های این قطعه، چرخاندن آزادانه و بدون محدودیت آن‌هاست که امکان کنترل دقیقِ پارامترهای گوناگون بر روی مدارهای طراحی شده را به راحتی امکان پذیر می‌کند. دو مدل از روتاری انکودرهای پرکاربرد وجود دارد که به یکی از آن‌ها مطلق (Absolute) و به دیگری افزایشی (Incremental) می‌گویند. انکودرِ مطلق می‌تواند موقعیت دقیقِ دستگیره روتاری انکودر را به صورت درجه، به کاربر نشان دهد. در حالی که انکودرِ افزایشی اطلاعات مربوط به تعداد دفعاتیِ که شفتِ آن حرکت کرده است را در دسترس قرار می‌دهد. روتاری انکودری که در این مطلب بررسی و کدنویسی شده است، از نوع افزایشی می‌باشد.



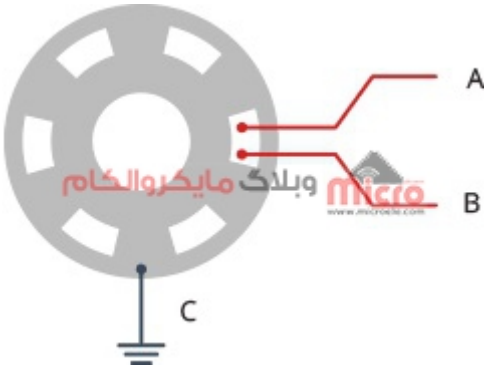
روتاری انکودر

مقایسه روتاری انکودر با پتانسومتر

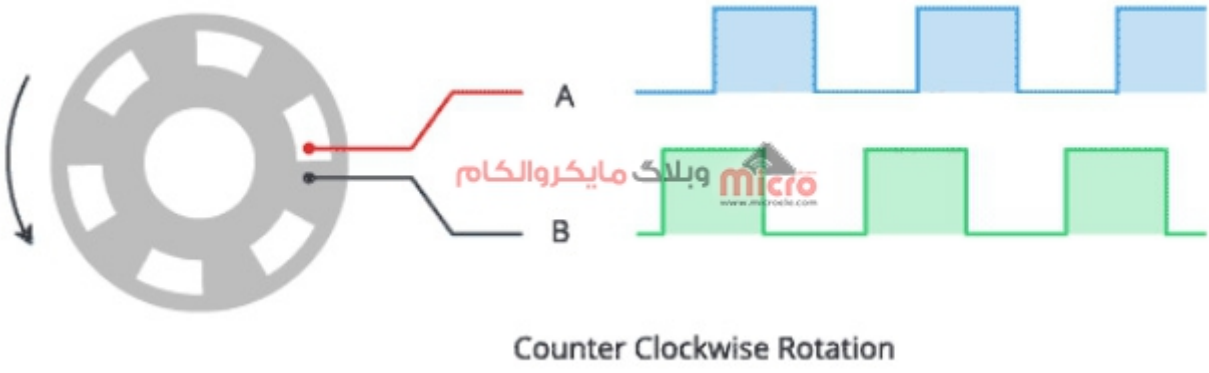
روتاری انکودرها را می‌توان به صورت کامل و بدون توقف به هر دو جهت چرخاند. درحالی که پتانسیومتر به اندازه سه چهارم محیط دایره می‌چرخد. پتانسومترها برای زمانی که می‌خواهیم از موقعیت دقیق شفتِ آن مطلع شویم کاربرد بیشتری دارند. اما زمانی که تغییر موقعیت شفت و جهت چرخش آن، اهمیت بیشتری داشتهَ باشد، روتاری انکودر بسیار کاربردی‌تر خواهد بود.

تشریح نحوه عملکرد روتاری انکودرها

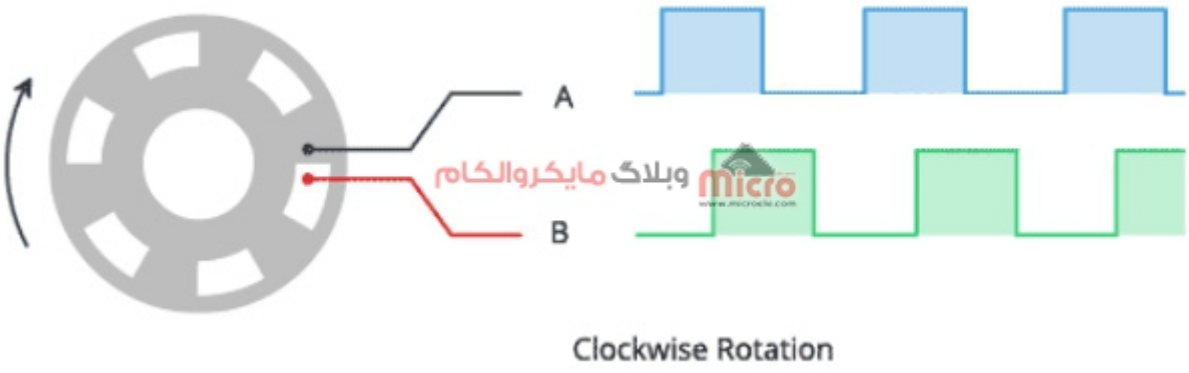
در داخل روتاری انکودر، یک دیسک شکاف دار متصل به پایه مشترک C و دو پایه تماس A و B وجود دارد که در شکل زیر نشان داده شده است. هنگامی که روتاری انکودر را می چرخانیم، A و B با پایه زمینِ مشترکِ C، به ترتیب خاصی که در جهت چرخاندن شفت قرار دارد، تماس برقرار می‌کنند. به این دلیل که یک پین، قبل از پین دیگر در تماس با زمینِ مشترک قرار می‌گیرد، سیگنال‌هایی که با هم 90 درجه اختلاف فاز دارند تولید می‌گردند. به این روش، رمزگذاری quadrature می‌گویند.



دیسک شکاف دار در داخل روتاری انکودر



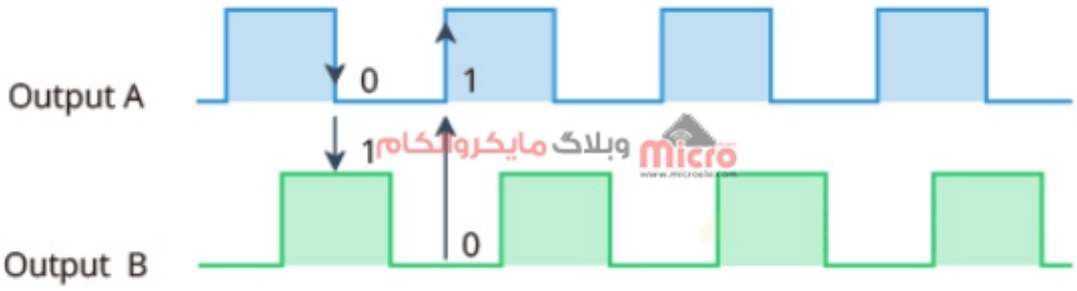
چرخش روتاری انکودر در جهت خلاف ساعت



چرخش روتاری در جهت عقربه‌های ساعت

وقتی روتاری را در جهت عقربه‌های ساعت می‌چرخانید، ابتدا پین A و سپس پین B به زمین متصل می‌شوند. زمانی که روتاری را در جهت خلاف جهت عقربه‌های ساعت بچرخانیم، ابتدا پین B و بعد از آن پین A به زمین متصل می شود. با ردیابی زمانی که هر پین به زمین متصل می‌شود و از زمین جدا می‌شود، می‌توانیم از این تغییرات سیگنال برای تعیین جهت چرخش روتاری استفاده کنیم. شما می‌توانید این کار را با مشاهده وضعیت B در هنگام تغییر حالت A نیز انجام دهید.

اگر A = !B، پس روتاری در جهت عقربه‌های ساعت چرخانده شده است.

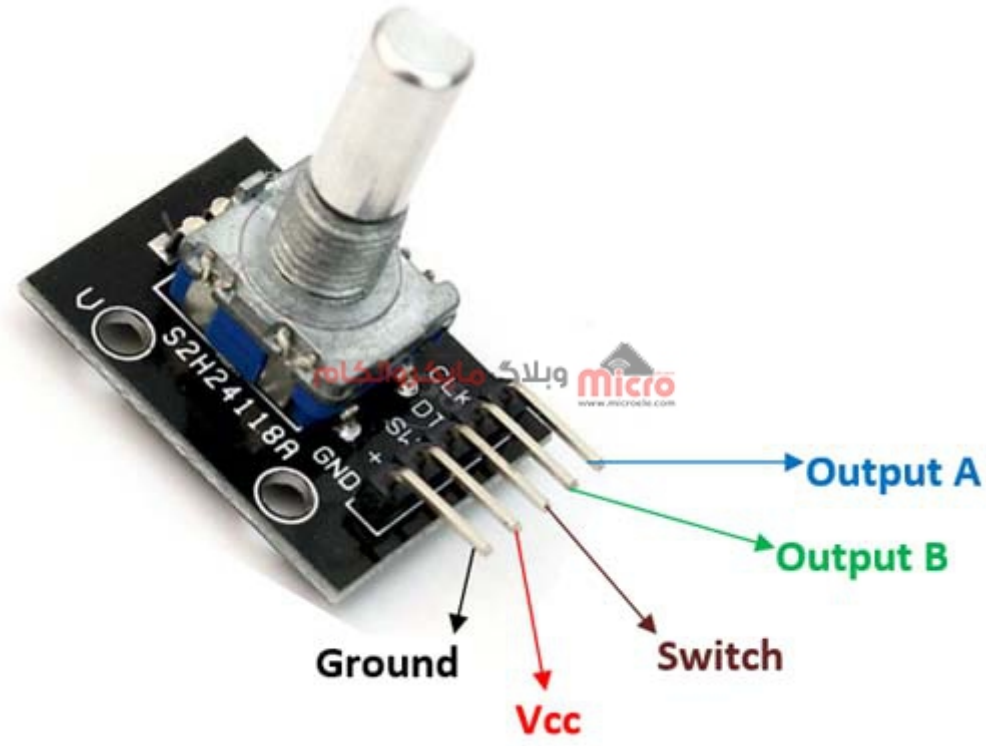


تولید پالس توسط روتاری انکودر

اگر A = B باشد، روتاری در خلاف جهت عقربه‌های ساعت چرخانده شده است.

تولید پالس با چرخاندن روتاری انکودر

معرفی پایه‌های روتاری انکودر



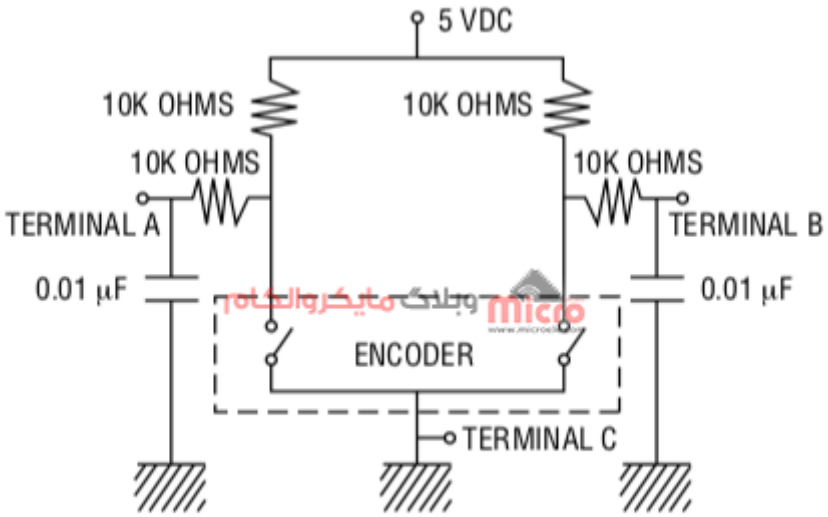
پایه‌های ماژول روتاری انکودر

نام پایه	توضیحات

GND	اتصال به زمین
VCC	اتصال به تغذیه 5+ یا 3.3+ ولت
Switch	کلید روتاری که با فشردن آن به سمت پایین با توجه به نوع ماژول High یا Low می‌شود.
Output B (DT)	مشابه خروجی CLK است، اما 90 درجه با CLK اختلاف فاز دارد. از این خروجی، جهت تعیین جهت چرخش استفاده می‌گردد.
Output A (CLK)	خروجی اصلیِ پالس برای تعیین تعداد چرخش است. هر بار که روتاری به اندازه یک واحد در هر جهت می چرخد، خروجی CLK در یک سیکل، HIGH و سپس LOW می‌شود.

نحوه دیبانس کردن روتاری انکودر

در صورت استفاده مستقیم از روتاری انکودر و بدون استفاده از ماژول‌های آماده، دقت شود که حتما مانند تصویر زیر از **دو عدد خازن 10nF** جهت دیبانس کردنِ روتاری، در مدار استفاده شود. در غیر این صورت در هنگام چرخش روتاری با سرعت بالا، جهت چرخش به درستی توسط میکروکنترلر شناسایی نمی‌شود. همچنین دقت شود که پایه‌های CLK، DT و SW با **یک مقاومت 10k بصورت Pull-Up** وصل شوند.



مدار دیبانس کردن روتاری انکودر

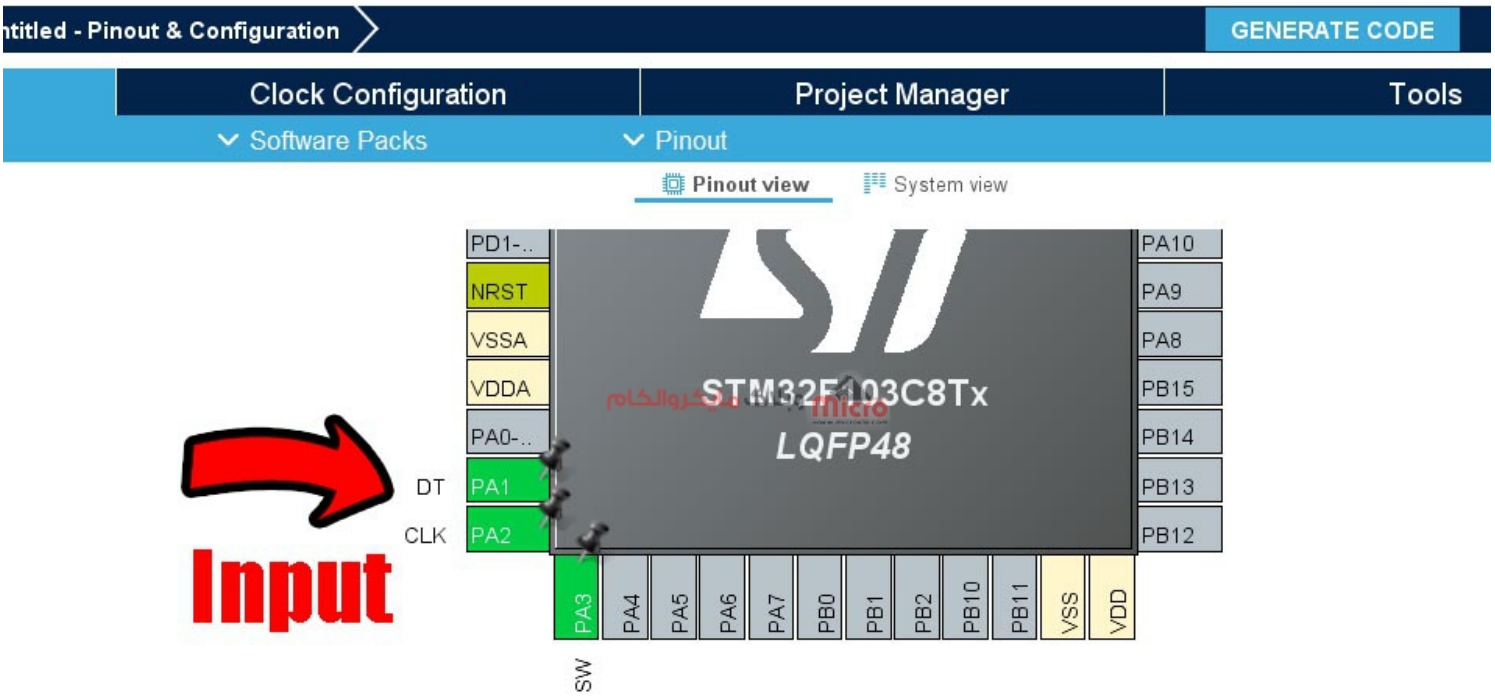
هدف از ارائه این مطلب

در این‌جا قصد داریم با استفاده از یک روتاری انکودر و استفاده از میکروکنترلر STM32، مقدار یک متغیر به نام counter را افزایش و کاهش دهیم. با چرخاندن روتاری در جهت عقربه‌های ساعت مقدار متغیر counter یک واحد افزایش و با هر بار چرخاندن آن در جهت خلاف ساعت، مقدار این متغیر یک واحد کاهش خواهد یافت. همچنین با فشردن کلید روتاری به سمت پایین، مقدار متغیر بر روی 0 باز خواهد گشت. در ادامه به پیکربندی و ساخت پروژه در CubeMX برای برنامه نویسی میکرو خواهیم پرداخت.

پیکربندی و تولید پروژه در نرم افزار STM32 CUBEMX

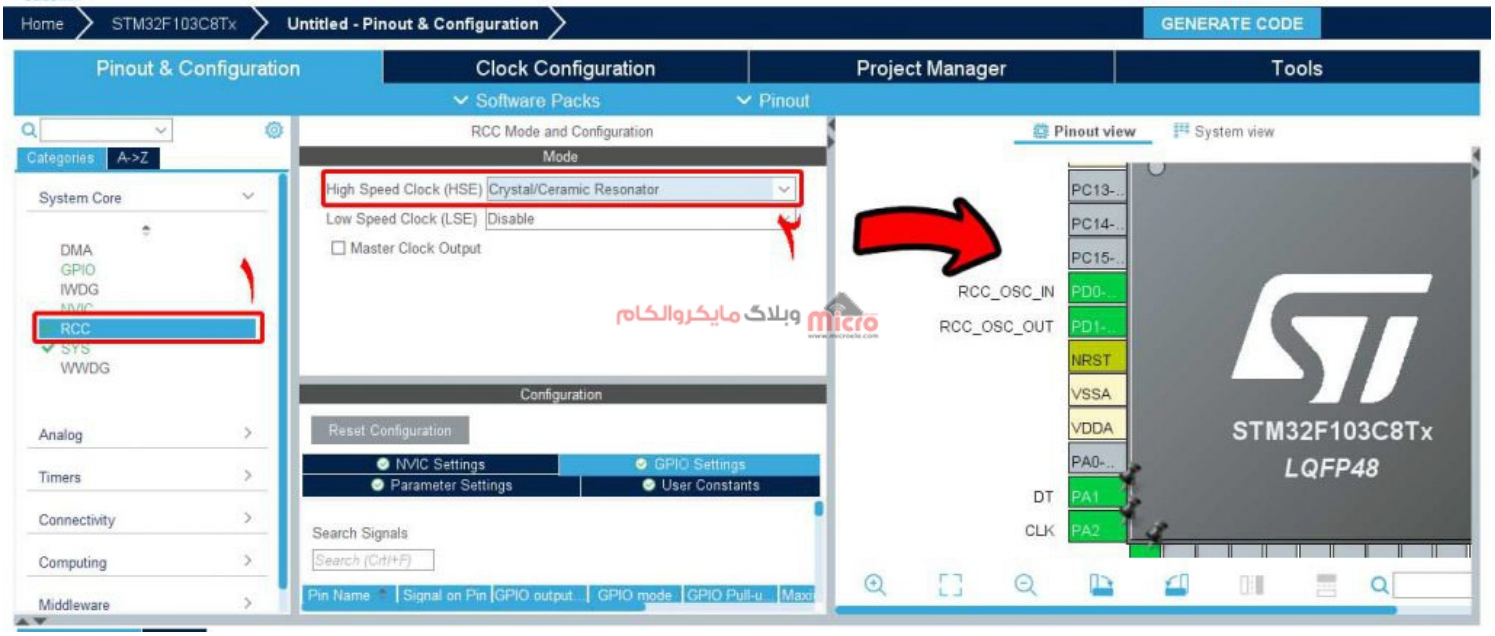
میکروکنترلر مورد نظر خود، که در این‌جا برای ما STM32 F103C8 می‌باشد را در نرم افزار CubeMX انتخاب کنید.

سه پایه دلخواه را در محیط CubeMX به صورت ورودی برای پایه‌های روتاری (CLK، DT و SW) انتخاب و تعریف کنید.

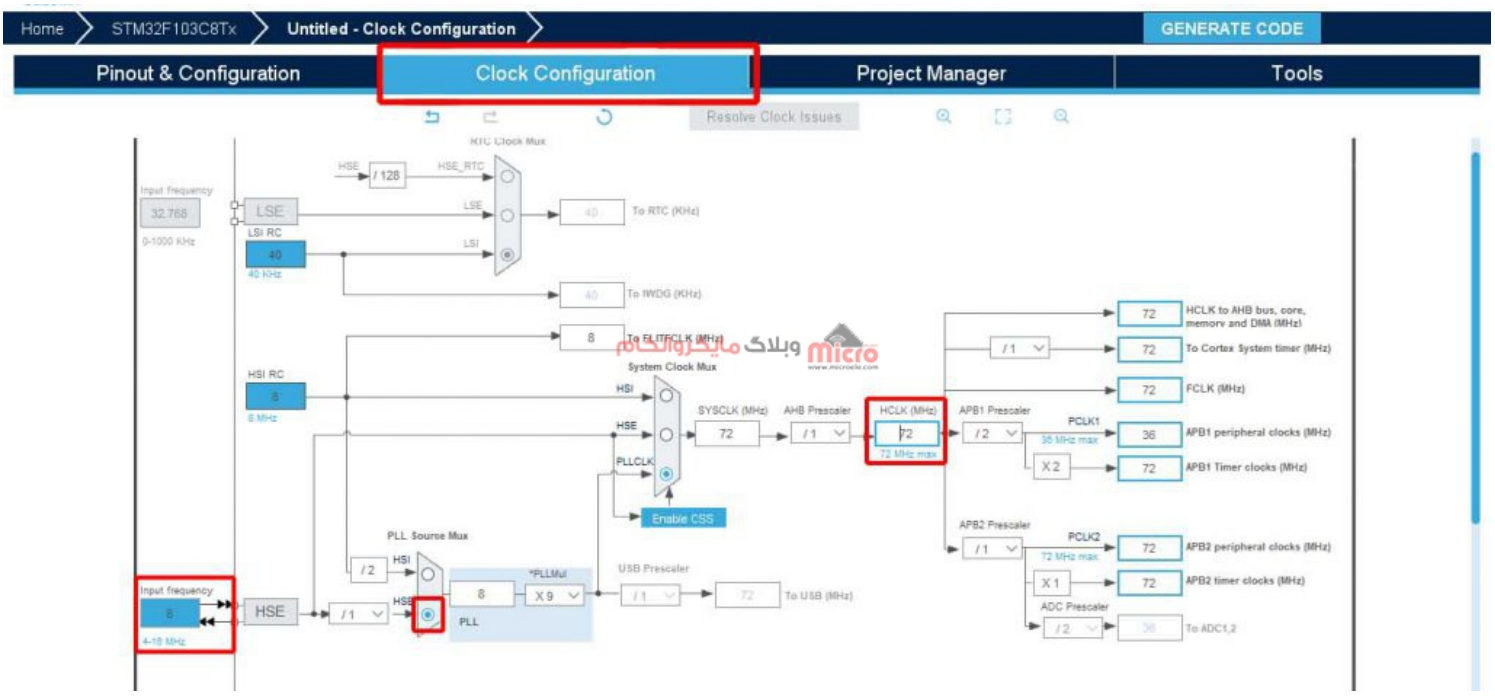


اعمال تنظیمات اولیه در *STM32CUBEMX*

از آن‌جایی که در این حالت از وقفه‌های خارجی استفاده نمی‌کنیم، برای عملکرد روان‌ترِ روتاری انکودر، میکرو را بر روی کریستال خارجی و فرکانس کاری بالا قرار دهید.



اعمال تنظیمات مربوط به فرکانس کاریِ میکرو در محیط *STM32 CUBEMX*

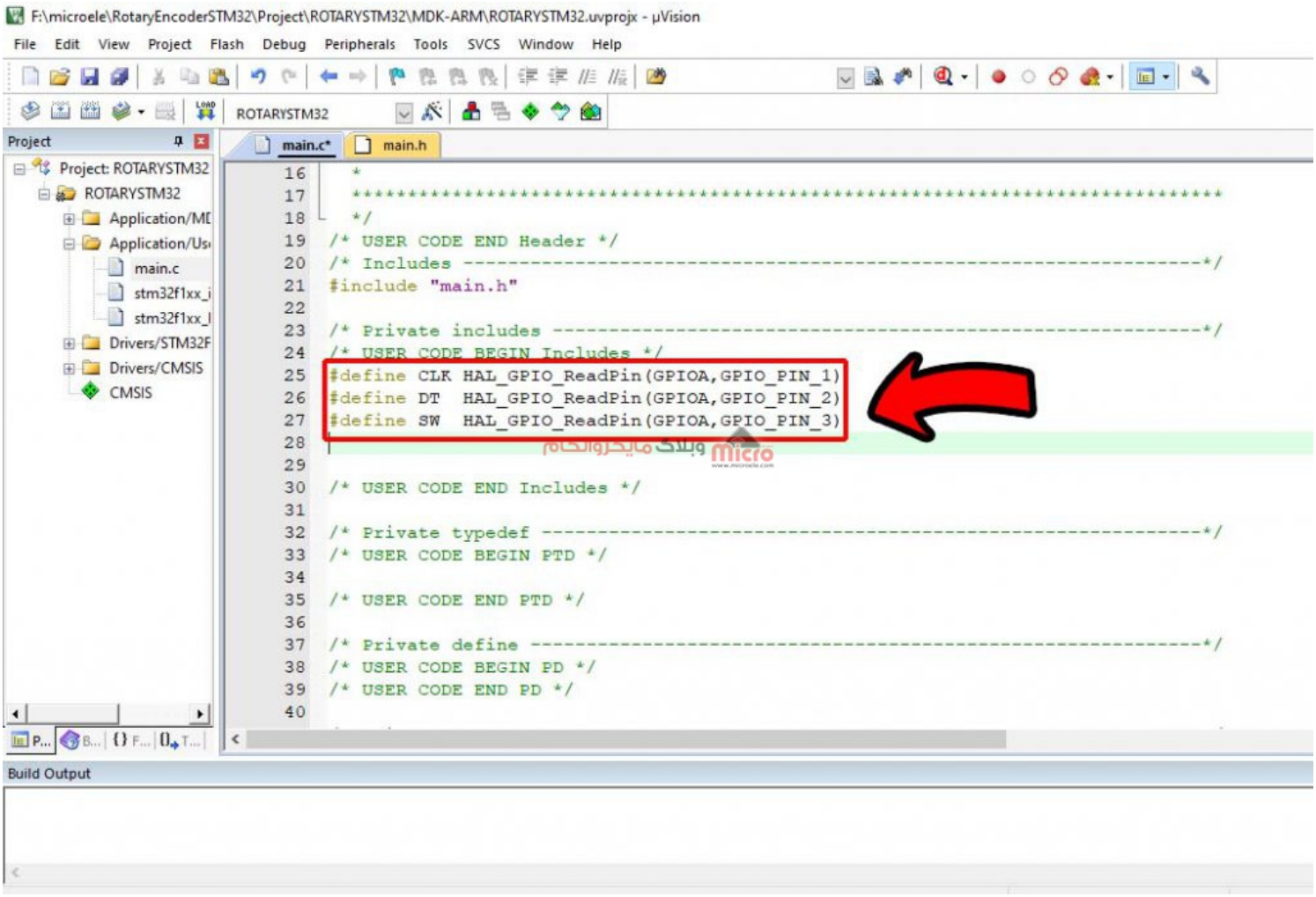


اعمال تنظیمات مربوط به فرکانس کاری میکرو در محیط *STM32 CUBEMX*

به سربرگ Project Manager رفته و IDE مورد نظر را جهت توسعه برنامه انتخاب و پروژه را ذخیره و Generate می‌کنیم.

برنامه نویسی برای راه اندازی روتاری انکودر در محیط Keil

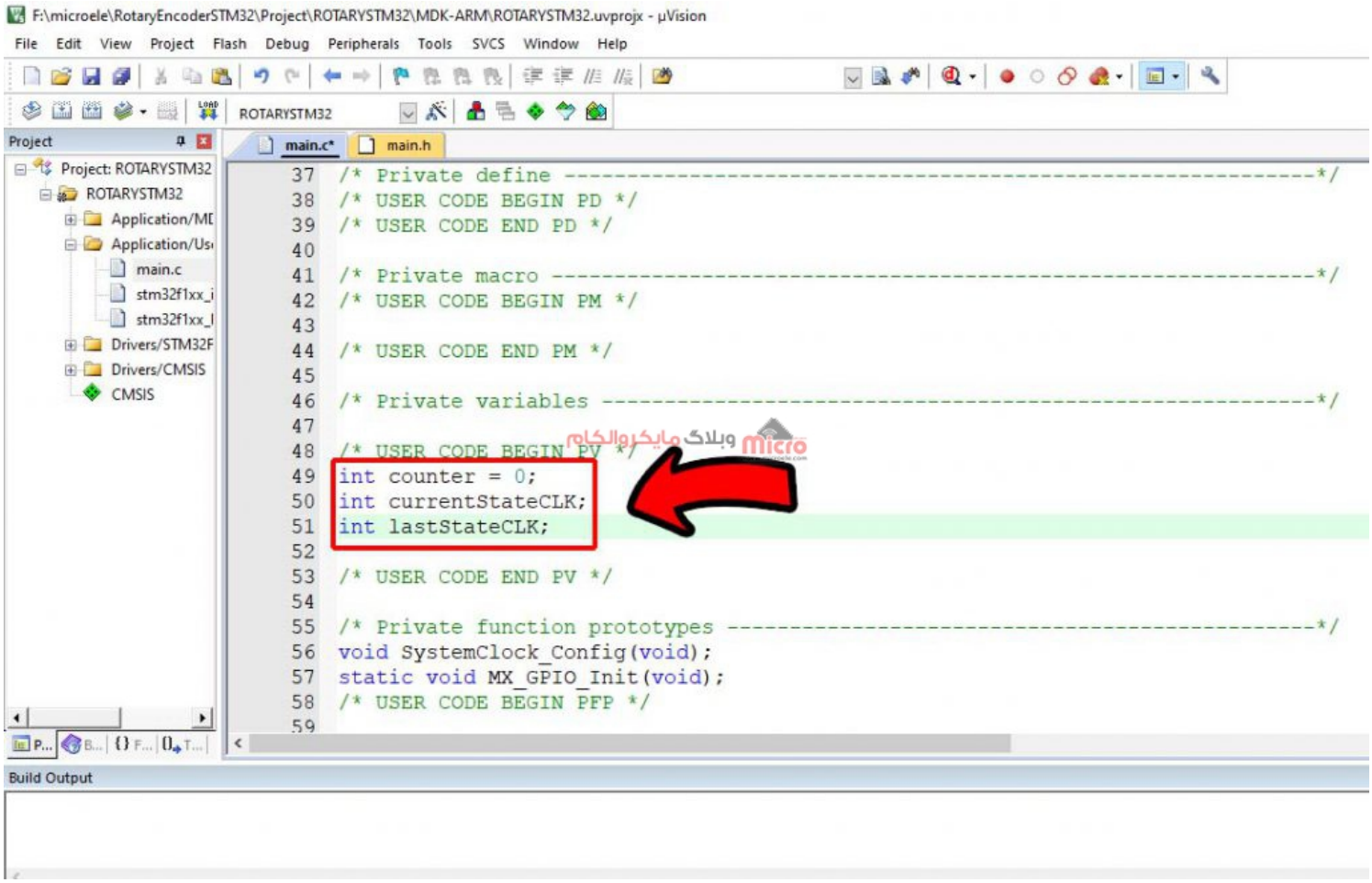
ابتدا پورت و شماره پایه‌های روتاری انکودر متصل شده به میکروکنترلر STM32 را از طریق define، در برنامه معرفی می‌کنیم.



تعریف پایه‌های روتاری انکودر در محیط *Keil*

کد های مربوط:

```
#define CLK HAL_GPIO_ReadPin(GPIOA,GPIO_PIN_1)
#define DT  HAL_GPIO_ReadPin(GPIOA,GPIO_PIN_2)
#define SW  HAL_GPIO_ReadPin(GPIOA,GPIO_PIN_3)
```

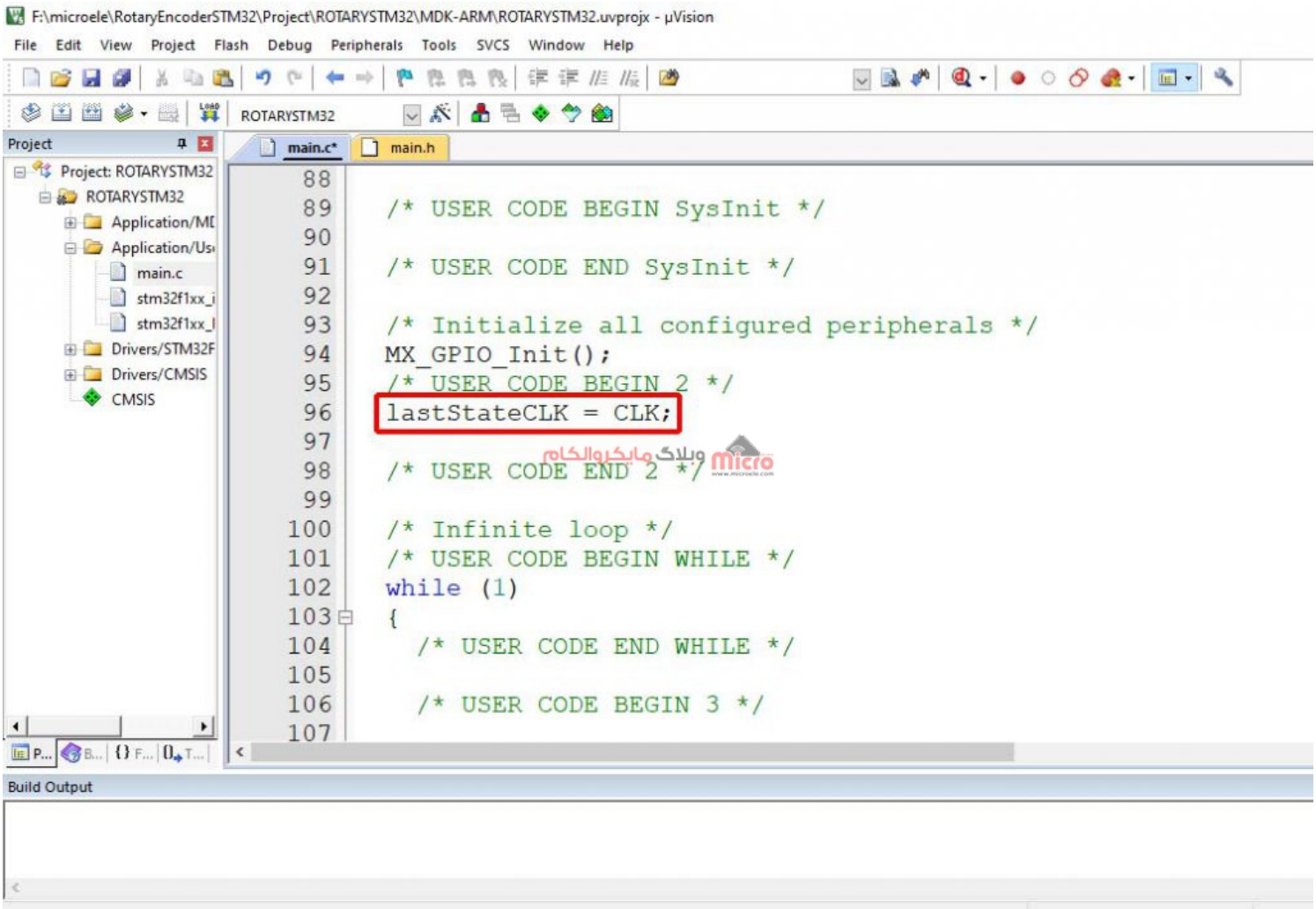


تعریف متغیرها در محیط *Keil*

کد های مربوط:

```
int counter = 0;
int currentStateCLK;
int lastStateCLK;
```

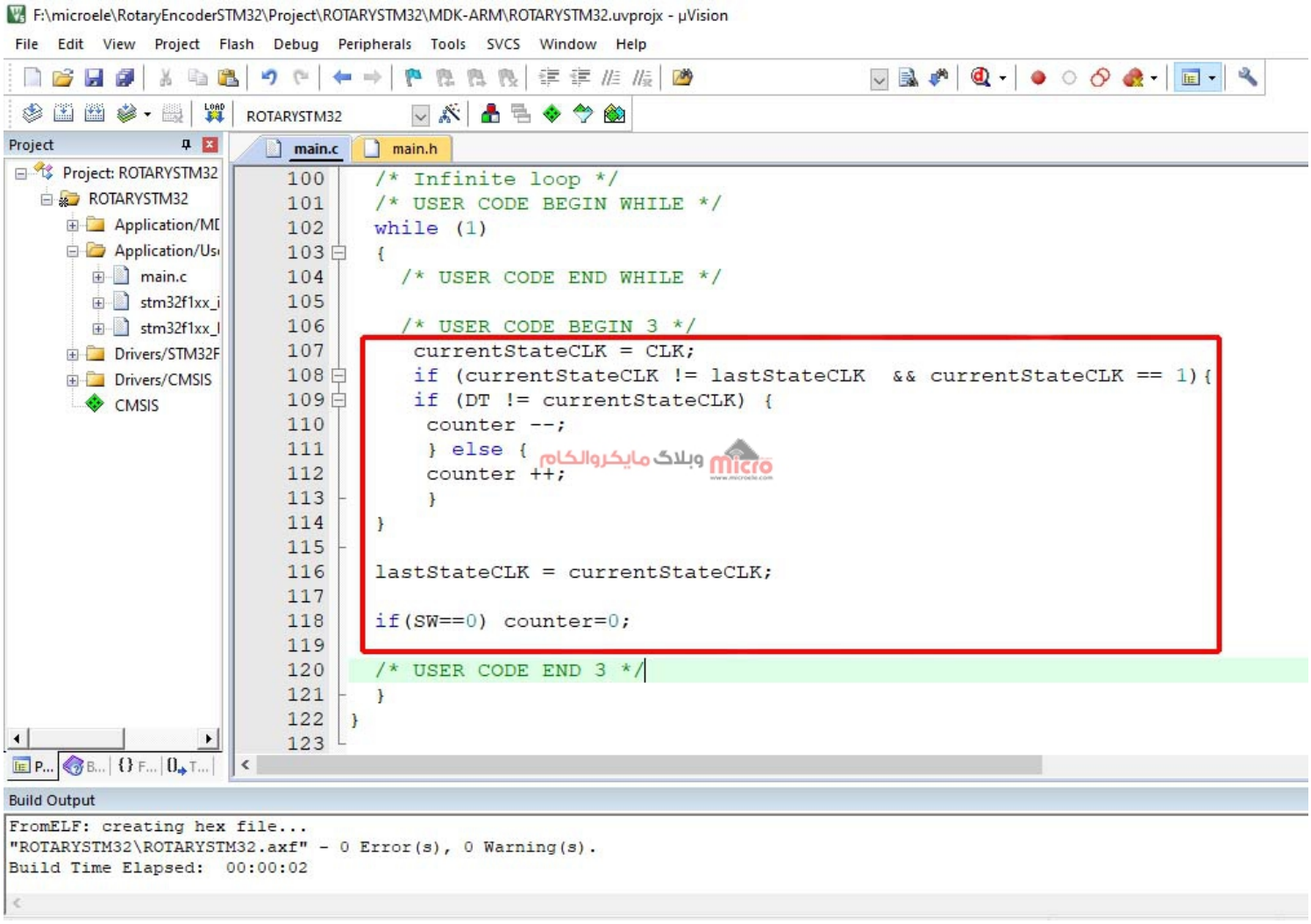
قبل از حلقه while، آخرین وضعیتِ پین CLK روتاری را خوانده و در متغیر lastStateCLK ذخیره می‌کنیم.



خواندن وضعیت پایه *CLK* در روتاری انکودر

```
lastStateCLK = CLK;
```

در این مرحله حلقه while را مانند شکل زیر تکمیل کنید.



تکمیل حلقه *while*

کد های مربوط:

```
currentStateCLK = CLK;
if (currentStateCLK != lastStateCLK  && currentStateCLK == 1){
         if (DT != currentStateCLK) {
            counter --;
          } else {
            counter ++;
          }
  }
    lastStateCLK = currentStateCLK;
    if(SW==0) counter=0;
```

بررسی کد های راه اندازی روتاری انکودر با STM32

1- متغیرهای currentStateCLK و lastStateCLK، وضعیت خروجی CLK را در خود نگه می‌دارند که برای تعیین تعداد چرخش‌های روتاری کاربرد دارند.

2- قبل از حلقه اصلی، مقدار حالِ حاضر CLK را خوانده و آن را در متغیر lastStateCLK ذخیره می‌کنیم.

```
lastStateCLK = CLK;
```

3- در حلقه اصلی، مجدد وضعیت CLK را بررسی کرده و با مقدار ذخیره شده در lastStateCLK مقایسه می‌کنیم. اگر با یکدیگر تفاوت داشته باشند به این معنا خواهد بود که روتاری انکودر چرخانده شده است و پالس تولید شده است. همچنین بررسی می‌کنیم که مقدار currentStateCLK برابر 1 باشد تا بدین صورت تغییر وضعیت روتاری را به عنوان یک واحد در نظر بگیریم، تا از دو دفعه شمارش شدن آن جلوگیری شود.

```
currentStateCLK = CLK;
if (currentStateCLK != lastStateCLK  && currentStateCLK == 1);
```

4- در صورتی که مقادیر CLK و lastStateCLK با یکدیگر متفاوت و نابرابر باشند، به این معنا خواهد بود که روتاری انکودر در خلاف جهت ساعت چرخانده شده است. بنابراین مقدار متغیر counter را یک واحد کاهش می‌دهیم. اگر هر دو مقدار یکی باشند، به این معنا خواهد بود که روتاری انکودر در جهت ساعتگرد چرخیده است. پس مقدار متغیر counter را افزایش می‌دهیم.

```
if (currentStateCLK != lastStateCLK  && currentStateCLK == 1){
        if (DT != currentStateCLK) {
           counter --;
         } else {
           counter ++;
         }
 }
```

5- مجدد آخرین وضعیت CLK را خوانده و lastStateCLK را برابر با currentStateCLK قرار می‌دهیم.

```
lastStateCLK = currentStateCLK;
```

6- در بخش آخر نیز، پین مربوط به کلید روتاری را خوانده و متغیر counter را در صورت فشرده شدن آن، برابر با 0 قرار می‌دهیم.

```
if(SW==0) counter=0;
```

البته می‌توانید از دستور hal_delay نیز برای دیبانس کردنِ کلیدِ روتاری در هنگام فشردن آن استفاده کنید.

در نهایت برنامه را کامپایل و بر روی میکرو، پروگرام می‌کنیم. می‌توانید از بخش Debug متغیر counter را در قسمت watch اضافه کرده و با چرخاندن روتاری، تغییر مقادیر را مشاهده کنید. همچنین برای عملکرد بهتر می‌توان با راه‌اندازی واحد وقفه خارجی (EXTI) برای پین‌های روتاری، کدهای نوشته شده را به جای حلقه while، در فایل it.c نیز جایگزین کنید.

کد های کامل

برای دانلود سورس کد کامل روی این لینک کلیک کنید.

نتیجه گیری

در این مطلب نحوه راه اندازی روتاری انکودر و نحوه عملکرد آن توضیح داده شد، همچنین نحوه کدنویسی و راه‌اندازی آن با میکروکنترلر STM32 در محیط CubeMX و نرم افزار keil تشریح گردید. به دلیل استحکام و کنترل دیجیتالیِ خوبِ روتاری انکودرها، از آن‌ها درتجهیزاتِ رباتیک، ماشین‌های CNC پرینترها و بسیاری از پروژه‌های الکترونیکی استفاده می شود. از روش استفاده شده جهت کدنویسی روتاری انکودر در این مطلب، می‌توان برای سایر میکروکنترلرها و بردها مبتنی بر AVR،PIC، آردوینو و دیگر میکروکنترلرهای خانواده ARM نیز استفاده نمود.

امیدوارم از این آموزش کمال بهره را برده باشید. در صورتی که هرگونه نظر یا سوال داشتید درباره این آموزش لطفا اون رو در انتهای همین صفحه در قسمت دیدگاه ها قرار بدید. در کوتاه ترین زمان ممکن به اون ها پاسخ خواهم داد. اگر این مطلب براتون مفید بود، اون رو حتما به اشتراک بگذارید. همینطور میتونید این آموزش را پس از اجرای عملی توی اینستاگرام با هشتگ microelecom# به اشتراک بگذارید و پیج مایکروالکام (@microelecom) رو هم منشن کنید.