

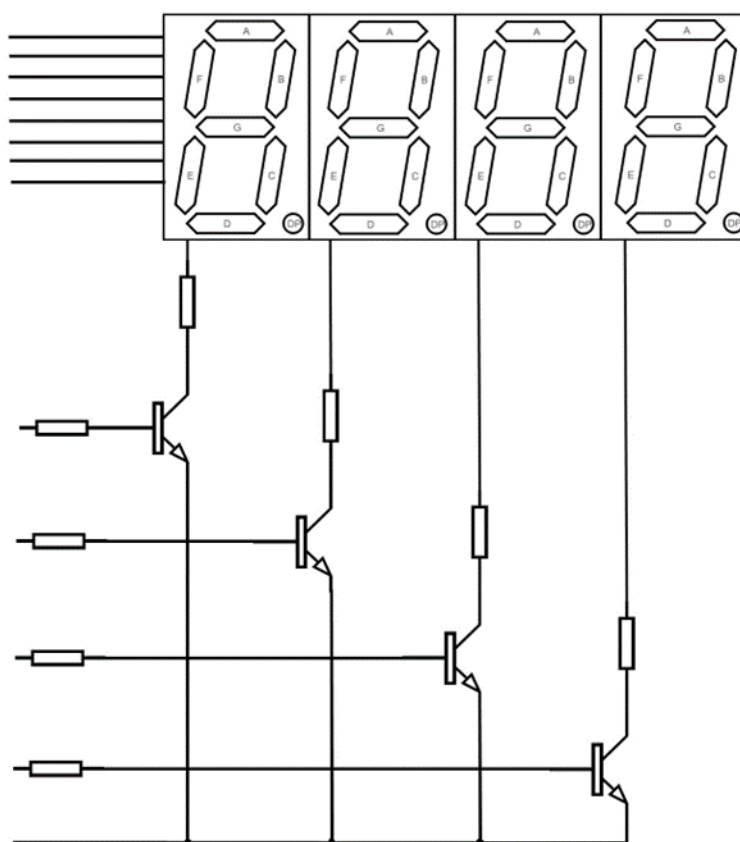
بسمه تعالی

گزارش آزمایش ۵ آزمایشگاه ریزپردازنده

سید محمدرضا حسینی

حسین حاجی رومنان

۷ سگمنت ها بدلیل پایه های زیاد همیشه چالشی برای طراحان مدار بوده است، برای راه اندازی ۴ عدد ۷ سگمنت بطور همزمان در صورتی که بخواهیم بصورت مستقل آنها را راه اندازی کنیم حداقل به ۳۲ پایه از میکروکنترلر نیاز داریم که این روش مناسبی نیست، بهترین روش برای راه اندازی چندین ۷ سگمنت بطور همزمان راه اندازی آنها بصورت موازی است، در این روش ۷ پایه دیتا بصورت مشترک به ۷ پایه ی دیتای ۷ سگمنت ها متصل میشود، سپس پایه مشترک هر ۷ سگمنت بصورت جدا به میکروکنترلر متصل میشود. مانند شکل زیر:



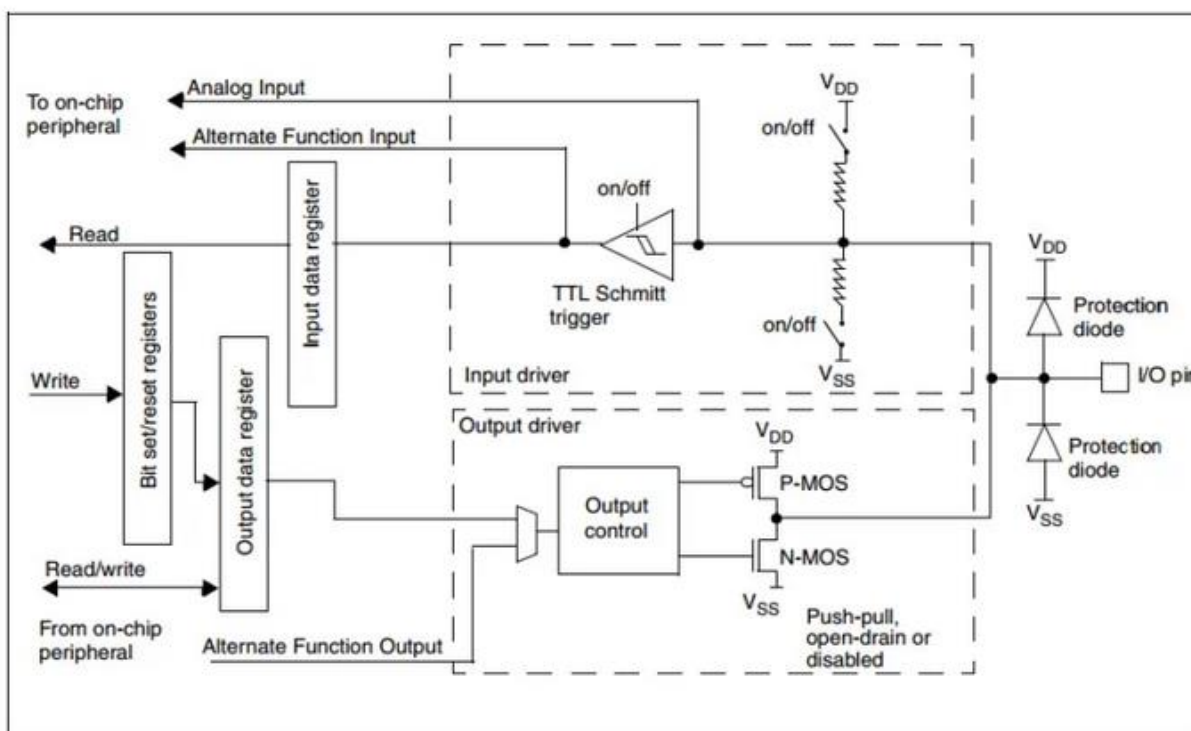
برای راه اندازی ابتدای دیتای رقم اول به آن داده میشود، سپس پایه مشترک رقم اول فعال و مابقی رقم ها غیر فعال میگردد

در گام بعدی دیتای رقم دوم داده میشود، سپس پایه مشترک رقم دوم فعال و مابقی رقم ها غیر فعال میگردد و همین کار ادامه داده میشود تا رقم آخر، این مراحل باید بسیار سریع انجام شود تا از خطای چشم انسان سریعتر باشد و همه رقم ها را بصورت همزمان روشن ببیند.

۲

در میان ۷ سگمنت های کاتد مشترک و آند مشترک، نوع آند مشترک آن از محبوبیت بیشتری برخوردار است، علت این امر آنست که بیشتر مدارهای منطقی که طراحی و ساخته میشوند مصرف کننده انرژی هستند و نه تولید کننده آن، عبارتی در آند مشترک برای فعال کردن پایه های دیتا ۷ سگمنت باید زمین مدار را تامین کنیم و این امری آسان تر است.

۳



با توجه به شکل بالا بخش gpio از بخش های مختلفی تشکیل شده است، بخش های اصلی آن شامل بخش کنترل کننده خروجی برای پوش پول و open drain خروجی است، بخش ورودی آنالوگ جهت راه اندازی بخش ADC، دو عدد دیود جهت حفاظت از ولتاژ ورودی به پایه، بخش پول آپ و پول دان ورودی، بخش رجیستر ها که تنظیمات مربوط به کارکرد پایه را میتوان مشخص کرد.

۴.

**Table 20. Port bit configuration table**

| Configuration mode        |                 | CNF1 | CNF0 | MODE1                                 | MODE0 | PxODR register |            |
|---------------------------|-----------------|------|------|---------------------------------------|-------|----------------|------------|
| General purpose output    | Push-pull       | 0    | 0    | 01<br>10<br>11<br>see <i>Table 21</i> |       | 0 or 1         |            |
|                           | Open-drain      |      | 1    |                                       |       | 0 or 1         |            |
| Alternate Function output | Push-pull       | 1    | 0    |                                       |       |                | Don't care |
|                           | Open-drain      |      | 1    |                                       |       |                | Don't care |
| Input                     | Analog          | 0    | 0    | 00                                    |       | Don't care     |            |
|                           | Input floating  |      | 1    |                                       |       | Don't care     |            |
|                           | Input pull-down | 1    | 0    |                                       |       | 0              |            |
|                           | Input pull-up   |      |      |                                       |       | 1              |            |

**Table 21. Output MODE bits**

| MODE[1:0] | Meaning                     |
|-----------|-----------------------------|
| 00        | Reserved                    |
| 01        | Maximum output speed 10 MHz |
| 10        | Maximum output speed 2 MHz  |
| 11        | Maximum output speed 50 MHz |

با توجه به دو جدول فوق حالات مختلف پایه های stm32 را میتوان دید. جدول بالا به ما این را می‌فهماند که اگر بخواهیم پین مورد نظر را خروجی از نوع پوش-پول کنیم باید در چهار بیتی که در اختیار داریم، در دو بیت CNF عدد “00” را نوشته و در دو بیت mode بر اساس سرعتی که می‌خواهیم، عدد موردنظر را قرار دهیم، که من می‌خواهم از سرعت 50Mhz استفاده کنم، که میشود “11” و اگر بخواهیم به صورت هگز بنویسیم می‌شود 0x3.

۵.

از این رجیستر برای قفل کردن تنظیمات پیکر بندی یک پورت یا پایه استفاده میشود، این قفل بودن تا ریست مجدد میکرو باقی خواهد ماند.

۶.

برای پروگرم کردن این میکرو میتوان از مد سریال استفاده کرد، برای اینکار پایه های PA13 و PA14 مورد استفاده قرار میگیرند، این پایه ها بصورت swclk و swdio خواهند بود و عملیات پروگرم کردن از طریق آنها صورت میگیرد.

۷.

در این میکروکنترلر ۴ سرعت مختلف برای حالت خروجی قابل فعالسازی است:

Low speed :00

Medium speed :01

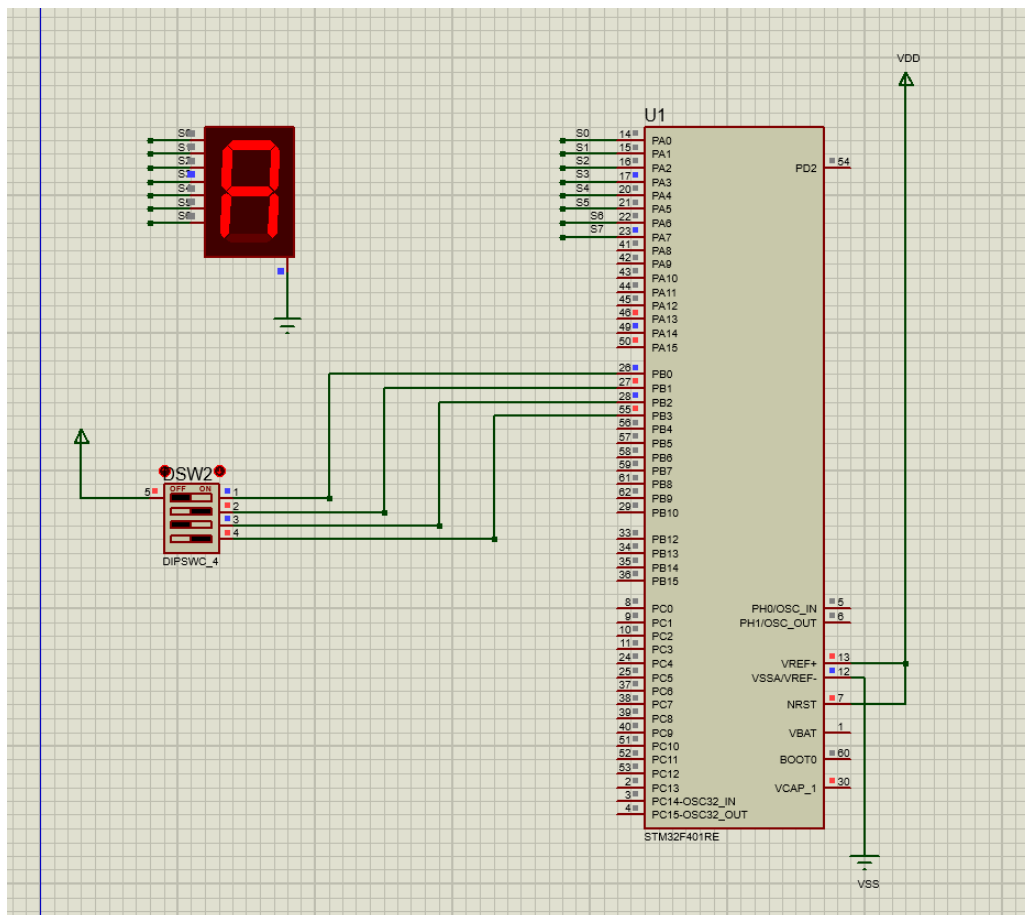
High speed :10

Very high speed :11

این سرعت بیانگر سرعت تغییر حالت بین حالت 0 و 1 در خروجی پایه خواهد بود

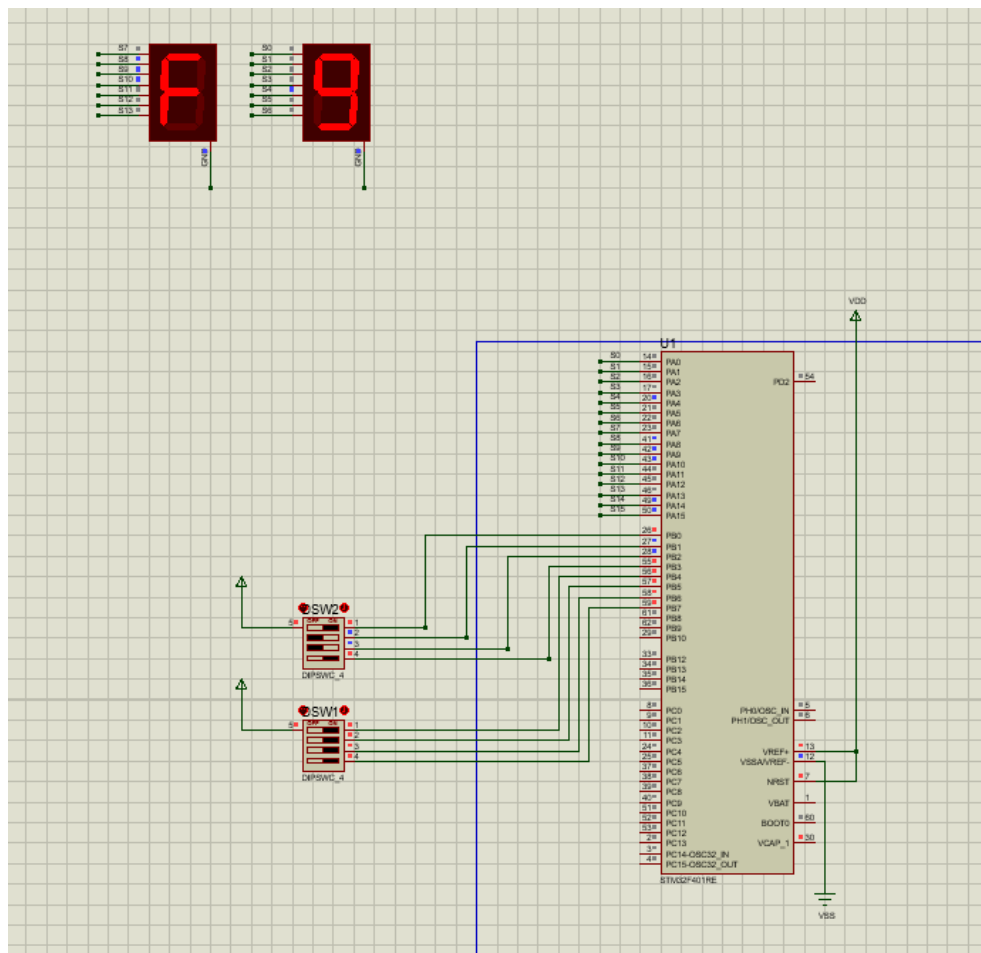
## سوالات پیاده سازی:

(۱)



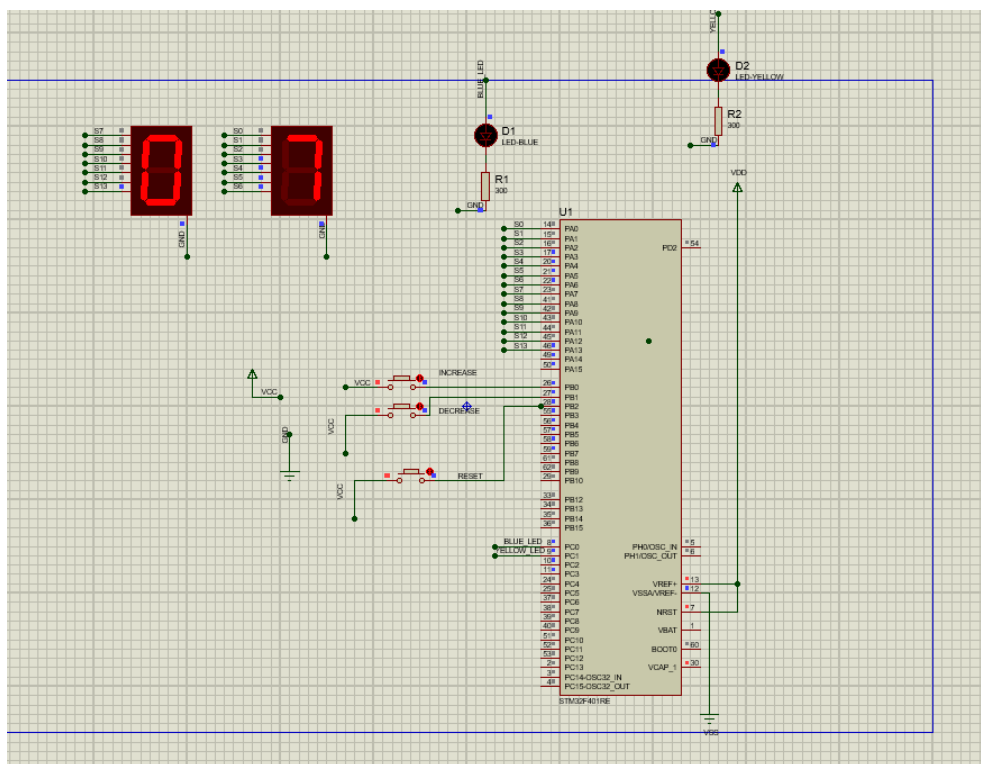
ابتدا کلاک GPIOA و GPIOB را فعال کرده و سپس پین ها را با استفاده از moder تنظیم میکنیم. چون از dip switch استفاده میکنیم ، پین های ۳ - ۰ را به صورت ورودی تنظیم کرده و pull down میگذاریم. با استفاده از متغیر ا ورودی های GPIOB را نگه میداریم و در هنگامی که ا با ورودی GPIOB برابر نباشد ، مقدار خروجی GPIOA را تغییر میدهیم.

(۲)



مطابق سوال قبل پین های ۴-۷ را تنظیم میکنیم و با تغییر مقدار ورودی در GPIOB مقادیر GPIOA را تغییر میدهیم . و بر روی ۷ SEG نمایش میدهیم.

(۳)



در این سوال نیز مطابق سوال قبل کلاک های GPIOA و GPIOB و GPIOC را فعال میکنیم و پین های را با استفاده از MODER تنظیم میکنیم. با استفاده از IF متوجه می شویم که کدام پین فعال شده و با استفاده از آن میتوانیم مقدار اولیه ا را اضافه کرده یا از آن بکاهیم. با استفاده از BTN سوم نیز میتوان مقدار ا را ریست کرد.