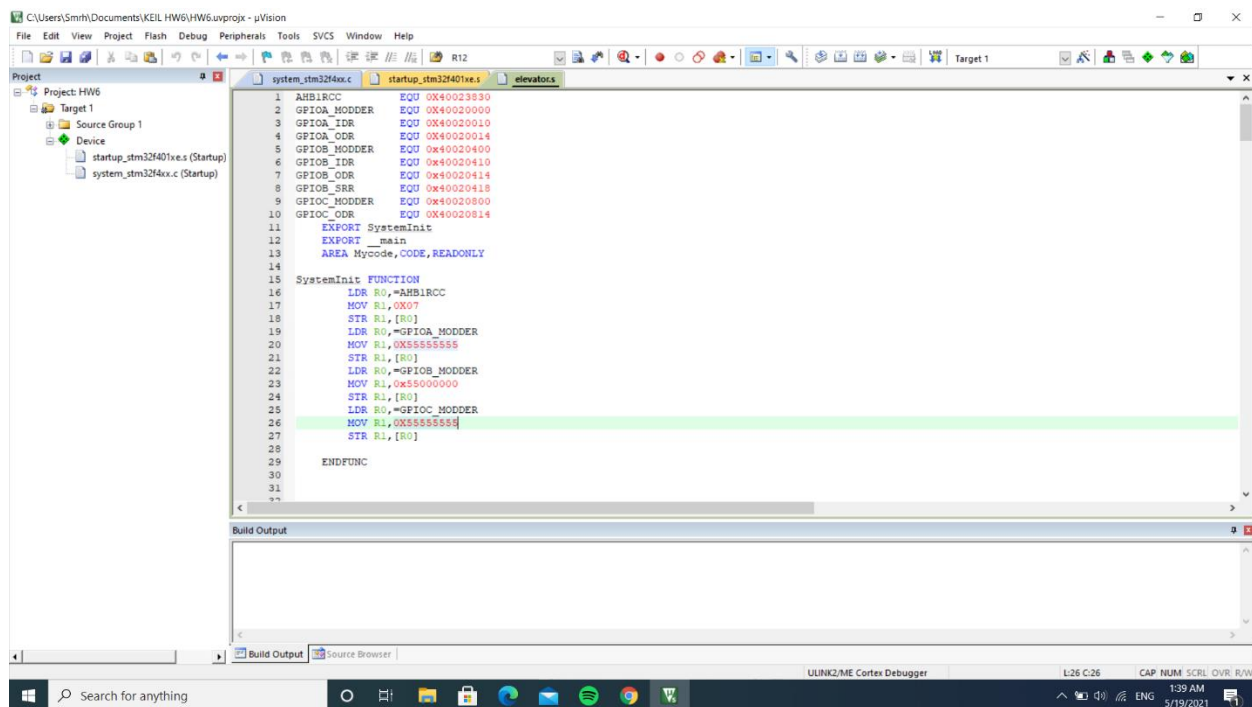


بسمه تعالی

گزارش تکلیف 6 ریزپردازنده

سید محمدرضا حسینی

97243129



در ابتدا با توجه به جدول مموری مپ و افست های رجیستر آدرس های مورد نیاز را پیدا کرده و به عنوان چند متغیر تعریف میکنیم .

با توجه به اینکه فقط پورت های رجیستر A , B را نیاز داریم پس آدرس های این دو رجیستر را ذخیره میکنیم.

برای استفاده از GPIOA , GPIOB, GPIOC باید کلاک این دو رجیستر در باس AHB1RCC را فعال کنیم که آدرس آن به صورت AHB1RCC ذخیره کرده ایم . برای اینکه کلاک سه رجیستر A,B,C را فعال کنیم این سه مقدار را یک گذاشته و بقیه را صفر میگذاریم و مقدار 7H را در خانه AHB1RCC از حافظه ذخیره میکنیم.

با توجه به اینکه فقط از رجیستر GPIOC, GPIOA, GPIOB استفاده کنیم در ابتدا PORT MODE REGISTER که نشان دهنده عملکرد هر پین خروجی است را مقدار دهی میکنیم . چون میخواهیم از GPIOA برای نمایش بر روی 7 SEGMENT استفاده کنیم و کلا 15 MODDER وجود دارد برای خروجی دادن هر مودر باید مقدار 01 را به آن وارد کنیم . پس مقدار 55555555H را در قسمت GPIOA MODDER ذخیره میکنیم . برای اینکه بتوانیم از GPIOC هم به عنوان خروجی برای روشن کردن LED ها استفاده کنیم مشابه GPIOA ، مقدار 55555555H را در خانه GPIOC_MODDER حافظه ذخیره میکنیم .

برای حالت دهی GPIOB MODDER با توجه به اینکه 4 بیت اول به عنوان خروجی و بیت های بعدی به عنوان ورودی استفاده میشود در خانه GPIOB_MODDER از حافظه مقدار 55000000H را ذخیره میکنیم

در حالت ابتدایی که شروع به کار نکرده مقدار FFFFH را در خانه GPIOA_ODR از حافظه که برای نمایش خروجی استفاده میشود ذخیره میکنیم تا این مقدار در 7 SEGMENT ها به نمایش درآید. و مقدار FFFFH را نیز به GPIOC_ODR میدهیم تا تمام LED ها روشن شوند.

سپس منتظر میمانیم تا به فشرده شدن BUTTON از که از طریق ورودی GPIOB متوجه آن میشویم مقدار 0000H در 7 SEGMENT ها به نمایش درآید و منتظر وارد کردن مبدا و مقصد از کاربر میمانیم.

با فشرده شدن BUTTON از حلقه انتظار خارج شده و ردیف های KEYPAD به ترتیب فعال میشوند و تمامی 7SEGMENT ها و LED ریست شده و با فعال شدن هر ردیف منتظر وارد شدن مقادیر از طرف کاربر میباشد. ابتدا ردیف 1 از طریق 1 کردن PB12 فعال شده و کاربر میتواند هر کدام از عدد های بین 1 تا 3 را انتخاب کند و در صورتی که پس از فعال شدن مقداری وارد نشود ردیف های بعدی به ترتیب فعال شده و مطابق توضیح بالا منتظر وارد شدن مقدار میمانند و به این روش میتوان گفت که کاربر هر کدام از رقم های بین 1 تا 9 را میتواند وارد کند. پس از وارد شدن مقدار تابع نوشته شده به که برای ذخیره و نمایش هر رقم نوشته شده فعال میشود.

پس از اینکه کاربر مقدار ابتدایی یا مبدا را انتخاب کرد و وارد تابع نوشته شده به ازای اون رقم شد مقدار رجیستر شماره 6 پردازنده بررسی میشود که آیا این رقم مقصد است یا مبدا در صورت صفر بودن مقدار موجود در رجیستر، عددی که کاربر از طریق KEYPAD وارد کرده بر روی 7SEGMENT به نمایش در می آید و مقدار آن به صورت X000H در رجیستر شماره 4 پردازنده ذخیره شده و رجیستر 6 یک میشود و LED ای که بالای 7 SEGMENT نخست قرار دارد روشن میشود. این روشن شده LED از طریق قرار دادن 8000H در خانه GPIOC_ODR حافظه اتفاق افتاده و این مقدار را در R5 نگه میداریم و پس از آن تابع DELAY صدا زده میشود تا عدد وارد شده کاربر به عنوان چندین عدد ذخیره نگردد و در این تابع مقدار R4 را در R8 ذخیره میکنیم. و پس از ایجاد تاخیر به تابع اولیه برگشته و منتظر وارد کردن رقم دوم از طرف کاربر میمانیم.

پس از اینکه کاربر عدد دوم را وارد کرد به تابع نوشته شده به ازای اون رقم رفته مقدار آن رقم را در R4 ذخیره کرده و مقدار R6 را بررسی میکنیم که این بار مقدارش 1 است و به همین دلیل به قسمت SHIFTER میرویم

در تابع شیفت مقدار عدد دوم که در R4 است را در R9 ریخته و R4 را چهار بار شیفت به راست می‌دهیم تا مقادیر که به صورت X000 وجود دارد به 0X00 تبدیل شود. مقدار اولیه که در R8 است را با مقدار به دست آمده در R4 ، OR می‌کنیم و در این رجیستر نگه داری می‌کنیم.

سپس مقدار مبدا و مقصد که به ترتیب در R8 و R9 ذخیره شده مقایسه می‌کنیم اگر $R9 > R8$ به سراغ قسمت UP می‌رویم ، اگر $R8 > R9$ به سراغ قسمت DOWN می‌رویم و در صورتی که دو رقم برابر باشد آسانسور به حالت شروع می‌رود و منتظر شروع از ابتدا می‌شود.

در قسمت UP میزان تفاوت R8,R9 را در R3 نگه می‌داریم تا بتوانیم با آن شمارش را داشته باشیم.

سپس مقدار اولیه بر روی 7SEGMENT ها به نمایش در می آید.

مقدار اولیه که در R8 است را به اندازه 1000H افزایش می‌دهیم و به اندازه R10 مرتبه آن را به سمت راست شیفت می‌دهیم و در R12 ذخیره می‌کنیم. به عنوان مثال اگر مقدار R8 پس از افزایش 2000H باشد پس از شیفت به 0200H تبدیل می‌شود.

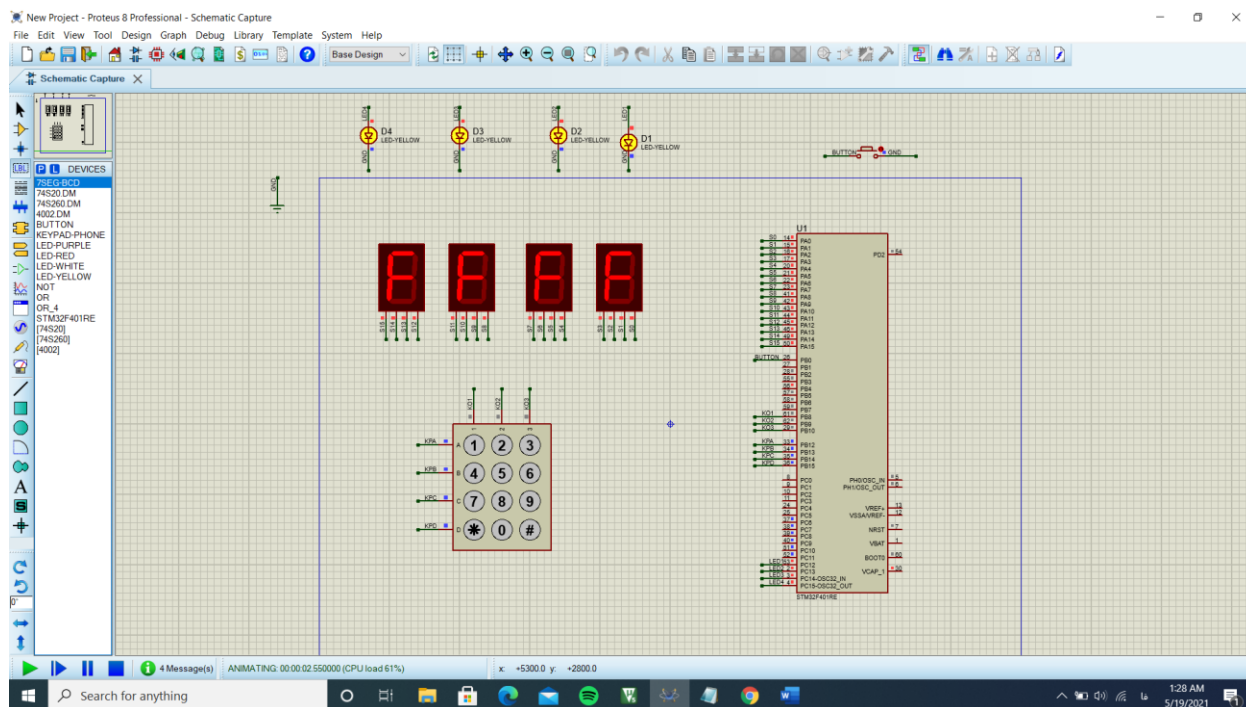
در ادامه R4 و R12 را OR کرده در R4 ذخیره می‌کنیم و در 7SEGMENT به نمایش در می آوریم.

پس از آن مقدار موجود در R5 یک واحد به راست شیفت پیدا میکند تا LED بعدی روشن شود.

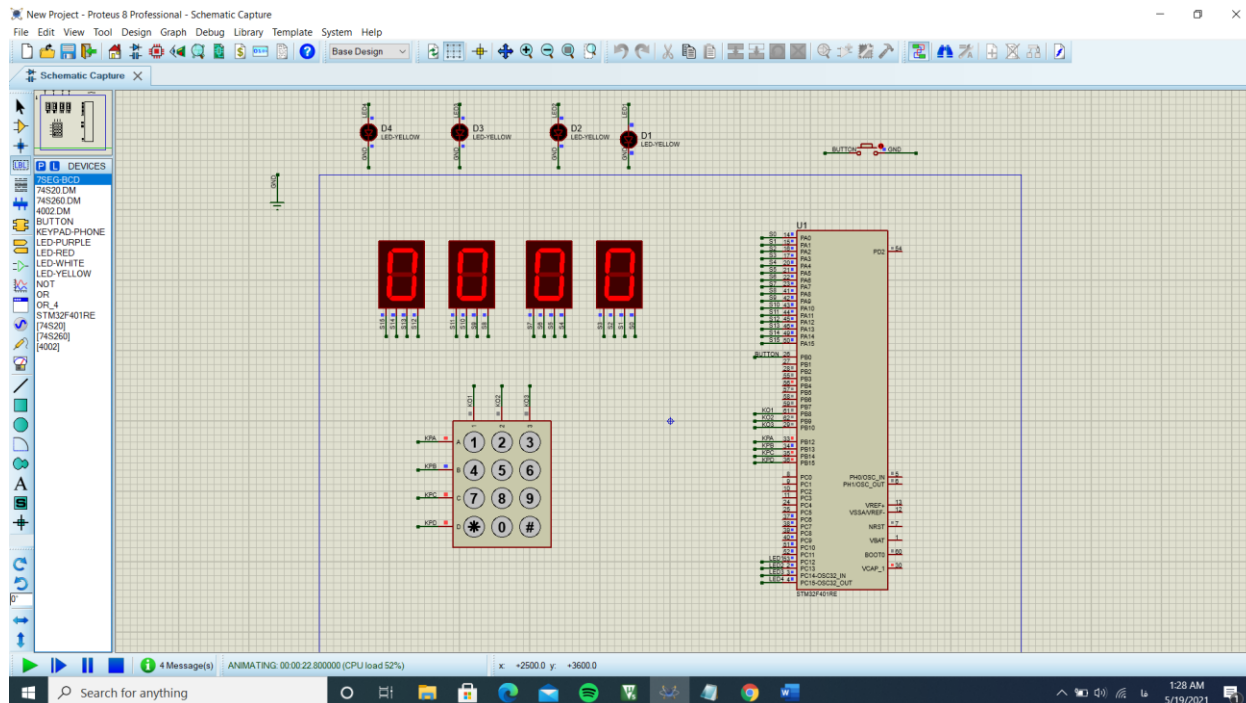
سپس تاخیری به اندازه 3FFFFFFH ایجاد می‌کنیم تا مقادیر به صورت طبیعی در 7 SEGMENT ها به نمایش درآیند.

پس از هر اجرا مقدار R10 به اندازه 4 واحد افزایش میابد تا شیفت ها به صورت درست انجام شود و در صورتی مقدار آن 16 شود ، مقدارش به 0 تغییر میابد و مقدار R4 نیز صفر می‌شود و مقدار 10000H در R5 قرار می‌گیرد تا بدلیل ریست شدن دوباره اولین LED روشن شود. پس از هر اجرای یک واحد از مقدار R3 کم می‌شود تا شمارش مقادیر به درستی انجام شود. در صورتی که مقدار R3 صفر شود یعنی به مقصد و عدد نهایی رسیده ایم و به حلقه پایانی می رویم.

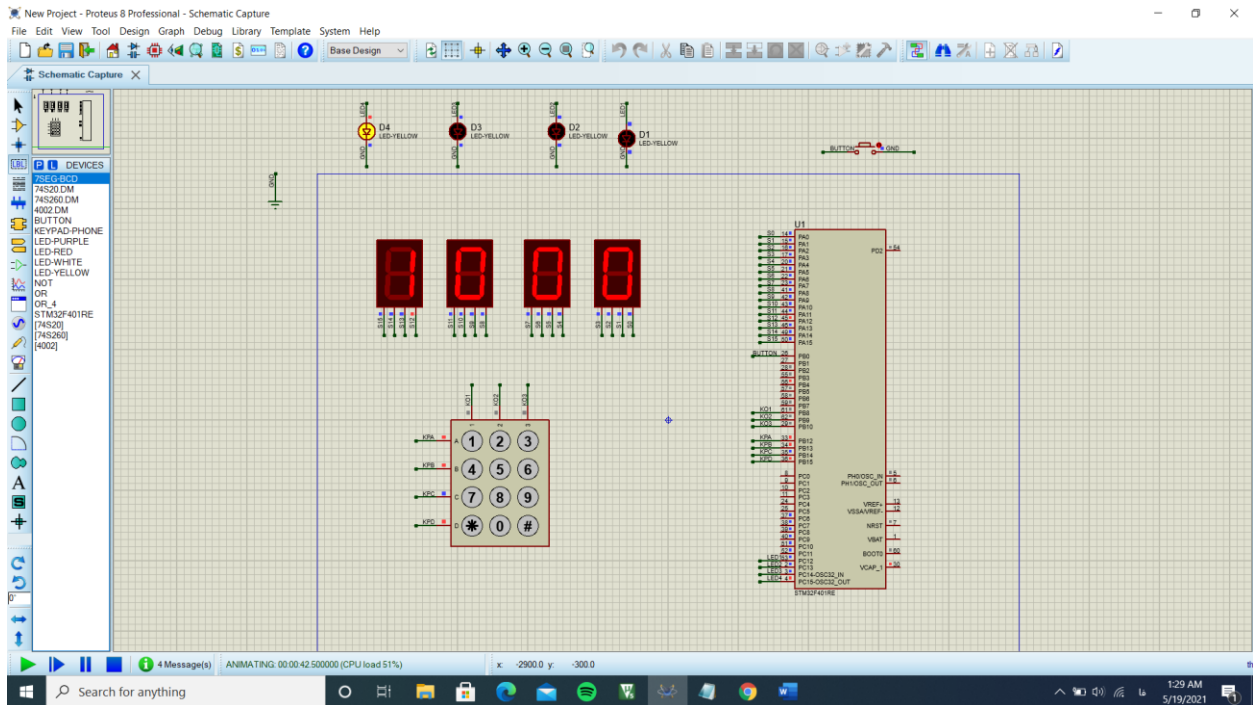
در صورتی که به قسمت DOWN برویم مشابه بالا عمل می‌کنیم فقط تفاوت آن این است که از مقدار اولیه 1000H کم می‌شود و در R3 نیز به جای R9-R8 مقدار R8-R9 قرار می‌گیرد.



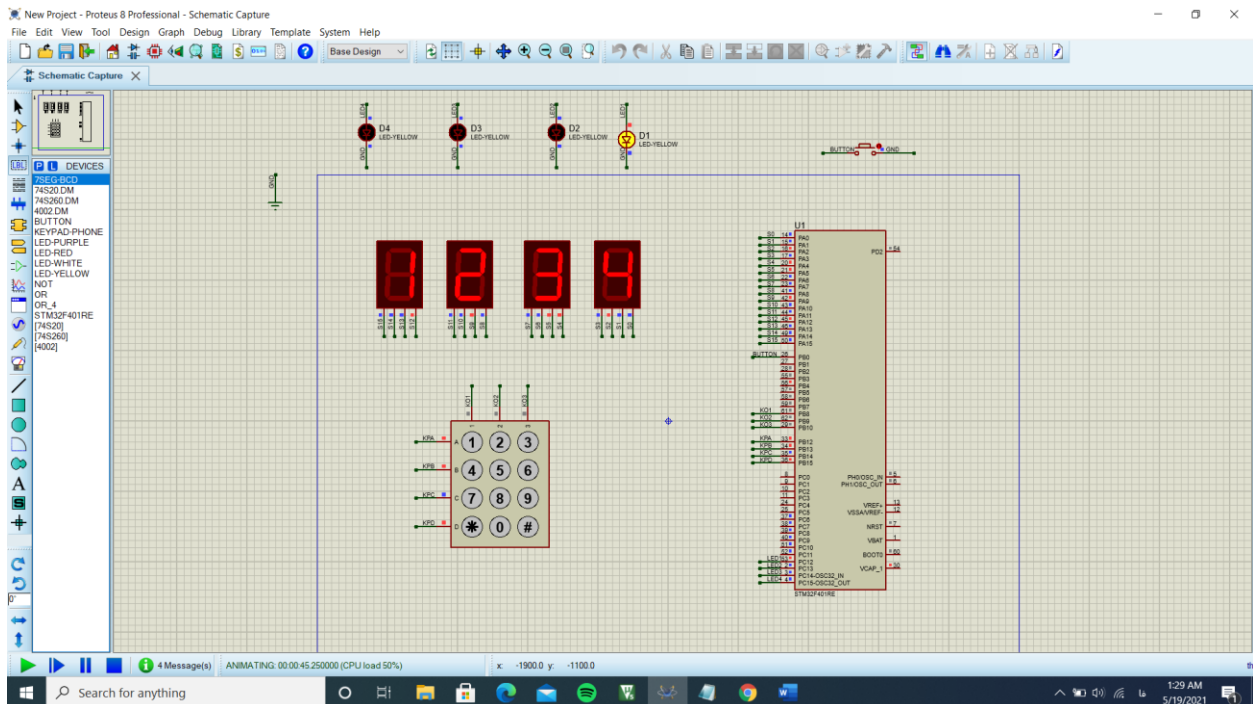
در حالت ابتدایی در نمایشگر مقدار FFFFH نشان داده میشود . تا هنگامی که BUTTON فشرده نشود در این حالت باقی میمانیم .



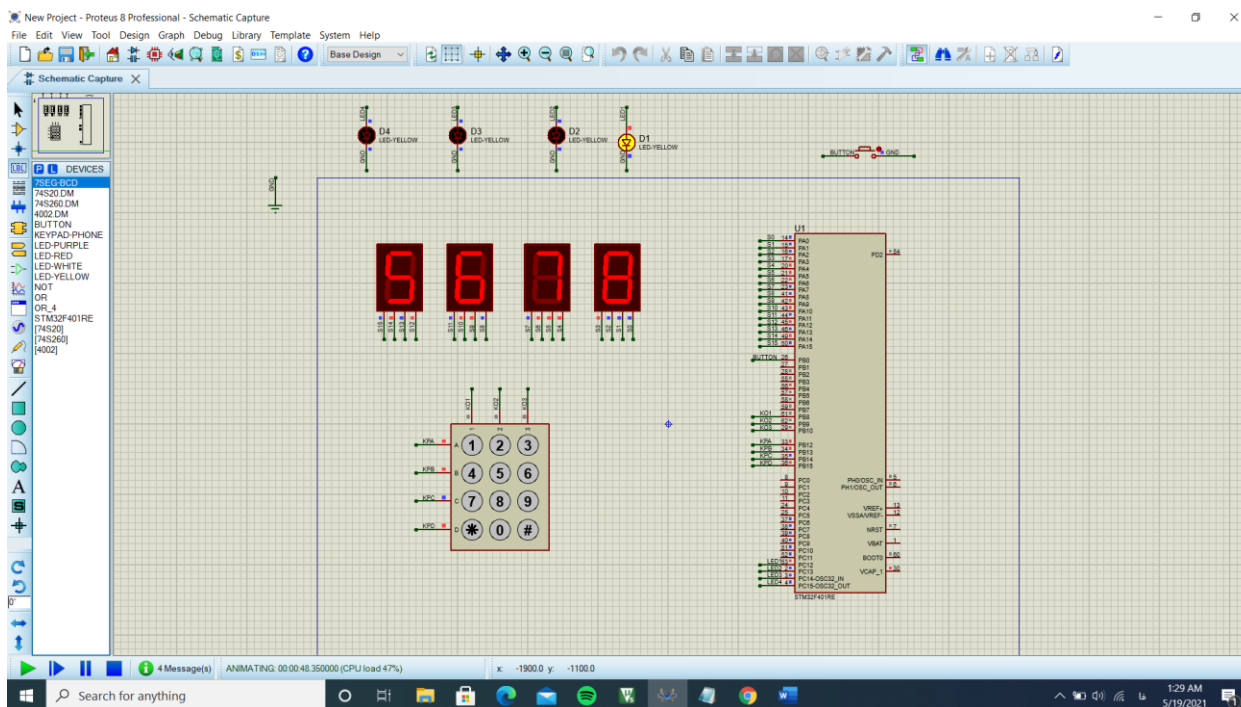
پس از فشردن BUTTON به حالت ریست رفته و مقدار 0000 بر روی 7SEGMENT نشان داده میشود ومنتظر وارد کردن مقادیر از طریق KEYPAD باقی میماند.



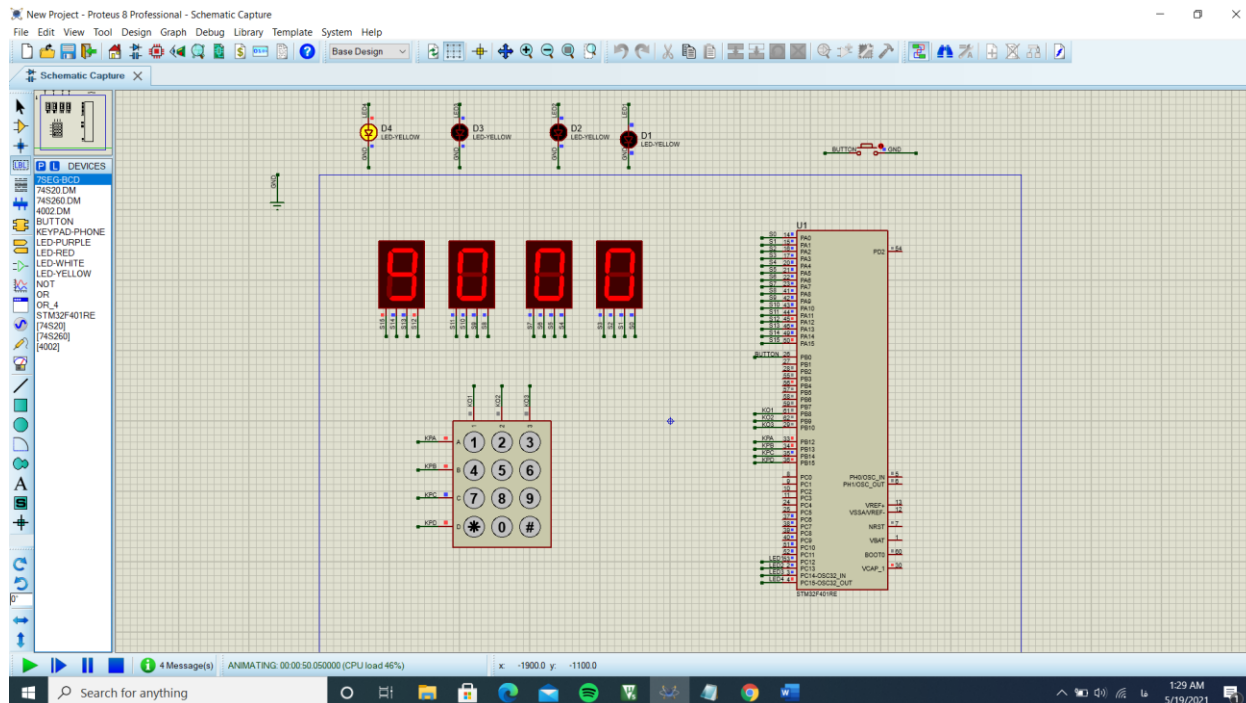
مقدار 1 و 9 از طریق KEYPAD به ترتیب وارد شده است



مقادیر 1234 به ترتیب نشان داده شده



پس از اینکه 1234 نمایش داده شد ، 7SEGMENT ها ریست شده و ادامه مقادیر به ترتیب به نمایش در می آید .



در نهایت با رسیدن به مقدار نهایی نمایشگر ها در همان حالت باقی میمانند

اجرای مرحله به مرحله مثال موجود در فایل سوال که ورودی های 3 و 7 وارد شده اند.

