حسن کاظمی طهرانی

9629041

آزمایشگاه ریزپردازنده

آزمایش 2

پیش گزارش:

کد ها و شبیه ساز مدار درپیوست موجود است.

انواع keypad ها و نحوه کار آن ها:

Keypad ها قطعاتی هستند که با طراحی ای که دارند به ما کمک میکنند که با تعداد کمی پین تعداد زیادی ورودی ( به صورت دکمه) را بتوانیم از محیط بیرون دریافت کنیم.کیپد ها انواع مختلفی دارد که برخی از آنها با میکروسوییچ کار میکنند ( کیپد های ماتریسی معمولی) و دارای سایز های متفاوت هستند(4\*4 یا 3\*4) که به این صورت کار میکنند که کلید ها به صورت ماتریسی قرار دارند و با فشار دادن آن کلید سیم های سطر و ستون مربوطه به هم متصل می شوند و در میکرو با اسکن می توانیم متوجه شویم که کدام دکمه فشرده شده است.همچنین نوع دیگر آنها نوع خازنی است که به جای میکروسوییچ از خازن استفاده شده و با نزدیک کردن انگشت ها به آن ها خازن ها تخلیه الکتریکی می شوند و با مدار تعبیه شده متوجه می شویم که چه ورودی به ما داده شده است.

پدیده نوسان:

پدیده نوسان اغلب در کیپد های ماترسی اتفاق می افتد که به علت نویز ممکن است به اشتباه تصور شود که یک دکمه چند بار زده شده است و برای جلوگیری از آن یکی از راه ها گذاشتن خازن در دوسر میکروسوییچ است ولی چون در کیپد ها به دوسر سوییچ دسترسی نداریم از روش های نرم افزاری برای جلوگیری از این پدیده استفاده می شود به این صورت که یک تایمری ست می شود و هرگاه تغییر ولتاژ به یک مدت خاصی باقی ماند ( به این مدت debouncing delay گفته می شود ) آن را به عنوان ورودی صحیح بپذیرد زیرا نویز ها به مدت های بسیار کوتاهی بر روی سیم میمانند در نتیجه با این روش نرم افزاری می توان از این پدیده جلوگیری کرد.

یکی دیگه از روش ها استفاده از latch ها در مسیر کلید می باشد که موجب می شود تنها در لحظه های خاصی ورودی بگیریم.

تعریف مختصر از توابع مورد نیاز:

* Keypad() : برای شروع استفاده از کیپد در آردویینو باید object آن تعریف و مقدار دهی شود که این تابع در واقع constructor این شی می باشد که مشخص میکند سطر ها و ستون ها به کدام پین متصل شده اند و هر کلید به کدام کاراکتر مپ شود و همچنین تعداد سطر ها و ستون ها را به عنوان ورودی میگیرد.
* getKey() : با این متد می توانیم همواره مشاهده کنیم که ورودی کیپد ما چیست.اگرهیچ کلیدی زده نشود false بر میگرداند اما اگر کلیدی زده شود کاراکتر مربوط به آن برگردانده می شود. نکته این متد این است که کد را متوقف نمی کند و blocking نیست و صبر نمی کند تا کلیدی زده شود سپس خروجی برگرداند.اگر خروجی نباشد false برمیگرداند.
* getKeys() :این متد وقتی مقدار درست برمیگرداند که چندین کلید همزمان فشرده شده باشد.
* waitforKey() : ورژنblocking تابع getkey میباشد که درصورتی که کلیدی وارد نشود کد در آن خط متوقف می شود و با فشردن کلید کاراکتر مربوط به آن بازگردانده می شود.
* getState() : وضعیت کلید های کیپد مارا برمیگرداند که میتواند یکی از موارد زیر باشد-

IDLE یعنی بیکار بوده است.

,PRESSED فشرده شده است.

,RELEASED تازه رها شده.

,HOLD مدتی نگه داشته شده است.

* keyStateChanged() :نشان دهنده این است که آیا کلیدی وضعیتش تغییر کرده است یا خیر.

نحوه و کاربردهای ارتباط سریال در آردوینو:

برای استفاده از ارتباط سریال در آردوینو می بایست از پین های قسمت communication استفاده کرد که پروتکل های ارتباط سریال را فراهم میکند.این پین ها علاوه بر اینکه می توان از آنها به عنوان پین دیجیتال استفاده کرد همچنین تنها پین هایی هستند که از ارتباط سریال پشتیبانی می کنند و یکی از مهم ترین کاربرد های ارتباط سریال اتصال میکرو های مختلف به یکدیگر برای ارتباط و یا اتصال usb و.... به میکرو است. قسمت RX برای شنود و قسمت TX برای ارسال پیام هست و با یکسان کردن ولتاژ مرجع صفر بین دو دستگاه می توانند این دو دستگاه با هم ارتباط برقرار کنند.

تعریف مختصر توابع ارتباطات سریال:

Begin() : یک عدد به عنوان ورودی می گیرد و با اون مقدار بر ثانیه ارتباط سریال را از پورت مربوطه آغاز میکند وپورت مربوطه را باز میکند و آماده خواندن و ارسال از آن پورت مخصوص می شود.

End() : برعکس تابع قبلی ارتباط سریال را قطع می کند و پورت مربوطه از ارتباط آزاد می شود.

Find() : بر روی پورت ورودی سریال منتظر دریافت ورودی تابع می شود و هنگامی که آن داده خاص بر روی پورت ظاهر شد trueو در صورت timeout مقدار false برمیگردد.

parseInt() : در انتظار اولین عدد صحیح روی پورت ورودی سریال می ماند و در صورت تمام شدن وقت (timeout)متوقف می شود.

println() : ورودی خود را به همراه \n به صورت رشته روی پورت خروجی به صورت ASCII مینویسد.

read() : اولین بایت ورودی را درصورت وجود برمیگرداند در غیر اینصورت -1 باز میگردد.

readStringUntil() : کاراکتر ها را تا زمانی که timeout نشده یا تمام نشده و یا به کاراکتر موردنظر نرسیده(که به عنوان ورودی تابع داده می شود) از روی بافر پورت ورودی سریال می خواند و برروی یک رشته می نویسد.

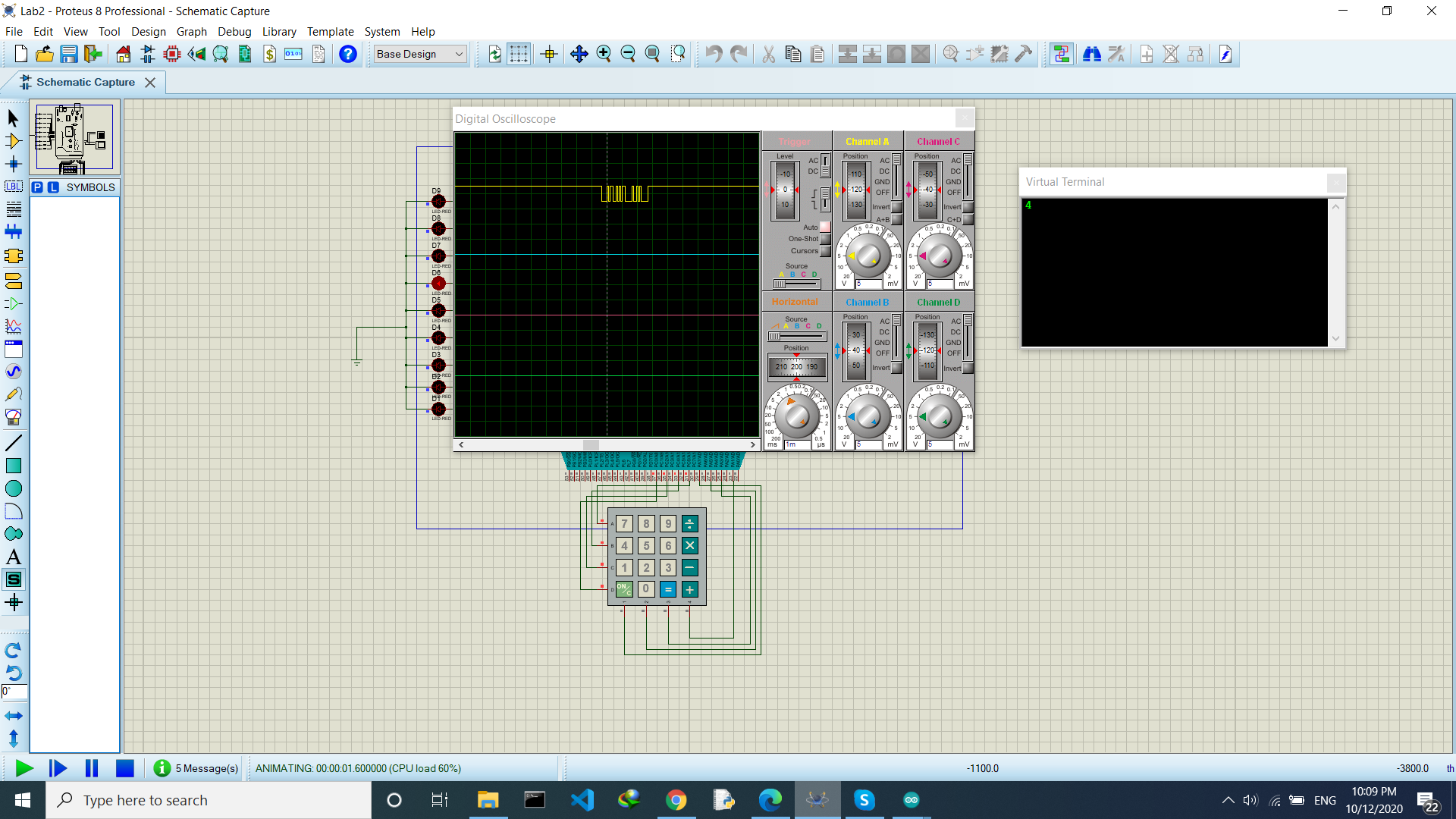
write() : ورودی خود را همانطور که هست بدون تغییر روی پورت خروجی سریال مینویسد.

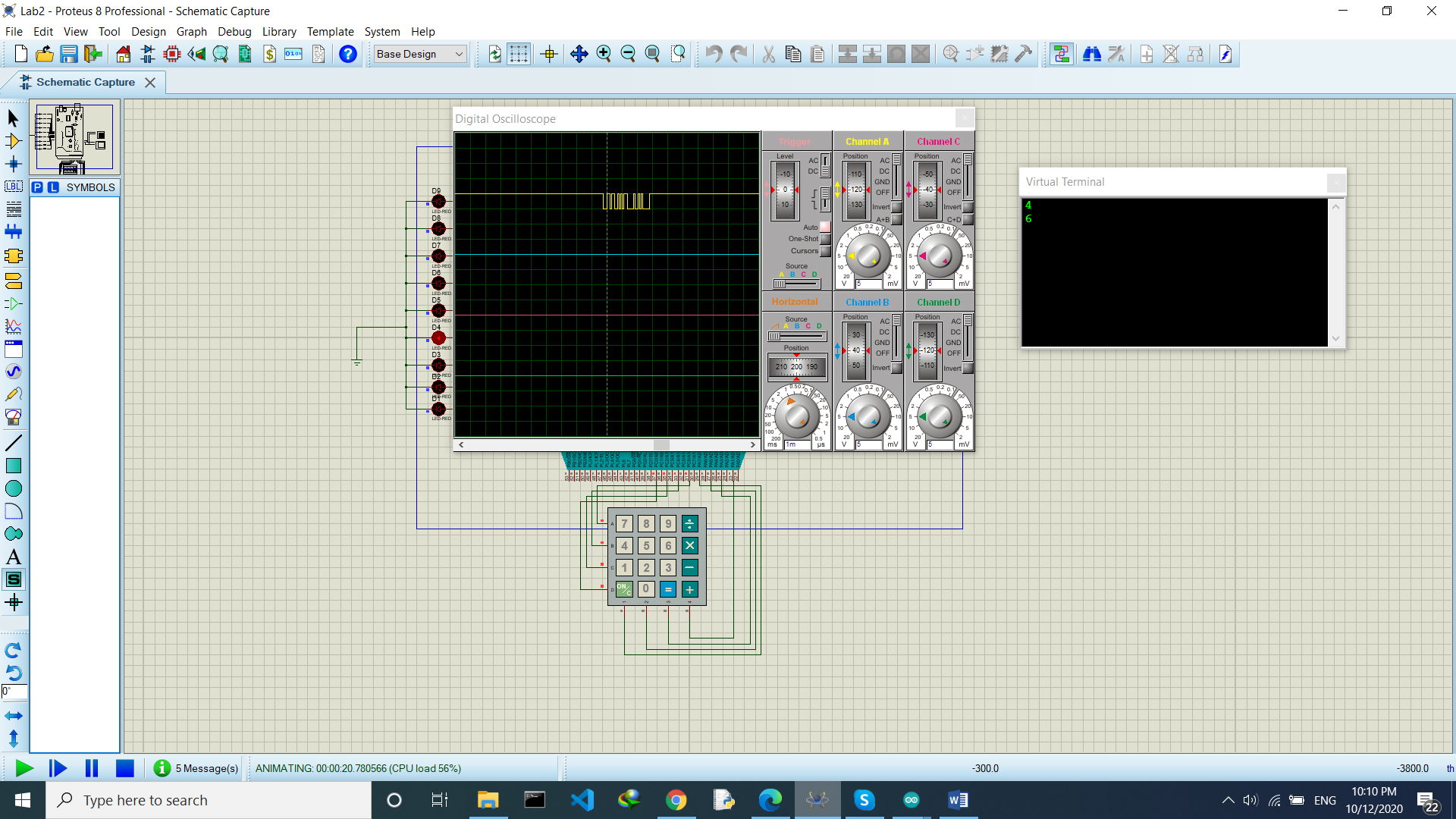
گزارش: برای هر 3 قسمت از یک مدار استفاده شده است.برای قسمت های 1 و 2 در قسمت code فولدر LAB2 و برای قسمت 3 در قسمت codes فولدر LAB2\_3 کد های مربوطه قرار داده شده است.

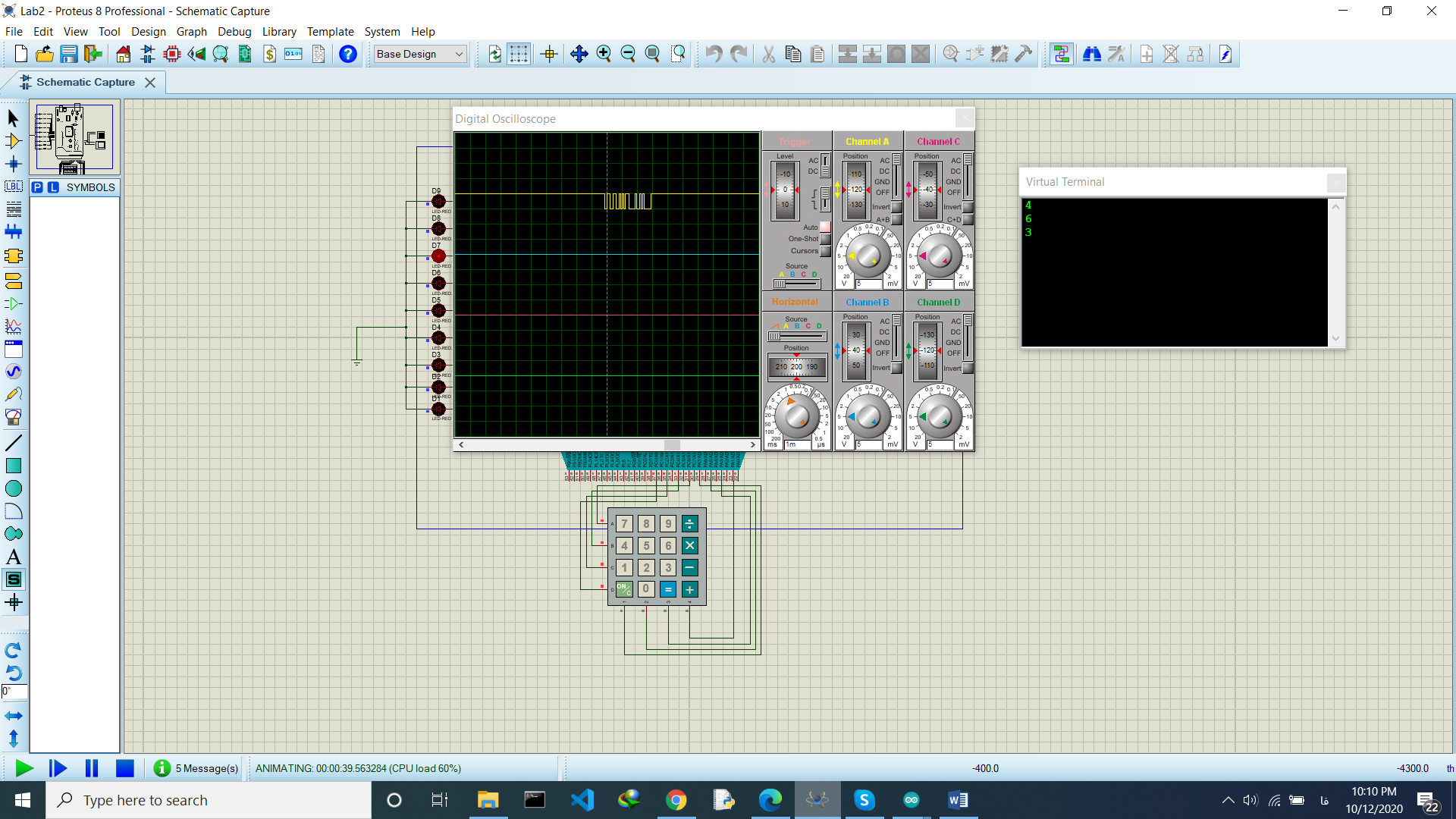
برای قسمت های 1 و 2 پس از ایجاد تنظیمات اولیه با هر بار دریافت از کلید ابتدا led هارا خاموش می کنیم سپس ورودی را روی ترمینال چاپ می کنیم و سپس led مربوط به آن عدد را روشن میکنیم.

برای قسمت 3 روند مشابه است و صرفا پس از دریافت از ترمینال led مربوطه را روشن میکنیم.

بخش هایی از اجرای قسمت های 1 و2 :







بخش هایی از اجرای قسمت 3 :

