

1 انواع Keypad ماتریسی و خطی کاربرد آن

سه Keypad به غیره ای از تاجی نویسی ها نوشتن به صورت خطی نوشتن به شکل شش یا هشت

ی دهند، در این هم قرار گرفته اند

در واقع همان طوری که در Proteus هم دیدیم Keypad ها بین 4 تا 2 و 3 و 4 ردیف

و ردیف و بین های A و B و C و D از طریق می کنیم و در نهایت اقرار در هم حال

میدار آن مقدار Low را بیان می دهد A و B و C و D شش ردیف بین می بینیم

الکترونیک نوشته شده و بعد از نوشتن این از ستون ها در نهایت نام قرار می دهیم و می بینیم که

شماره ستون را داشته باشیم می توان تشخیص دهیم که کدام صید نوشته شده است

انواع Keypad که در Proteus هم داریم Keypad - CALCULATOR

Keypad - SMALL CALC

Keypad - Phone

Keypad - TRELIS-C1

پیش از این نوشته شده 8 digispark.ir

2- بریده نوسان طیف چیست؟ چگونه می توان از بریده نوسان جلوگیری کرد؟

ضخامت هر طیف مناسب را فشرده یا به عبارتی تغییر وضعیت داده دوست صفت طیف به هم متصل شده و با فشرده شدن طیف بین دو قطب اتصال برقرار شده و پس از آن دو قطب از هم جدا شدند و جدا باز در آن لحظه تری نسبت به قبل به هم متصل شده و این حرکت ادامه می یابد یعنی تا زمانی که دو قطب روی هم قرار نگیرند در واقع به یک وضعیت پایدار برسند. در واقع تا زمانی که سیستم Stable شود می توان گفت هزاران بار دشارژ و شارژ انجام گرفته و طایر به صورتی است که این اتصال یک بار رخ داده در صحنه که وقتی دلفه و یا طایر افت مرده هم در زمان های کوتاه سطح شانهای مرتباً قطع و وصل شده تا پایدار شود. در مدارهای دیجیتال این پرتش ها و نویزها موجب اختلال می شود که در واقع نویزهای حاصل از تغییر حالت طیف موجب اتصال ضعیف و یونک به مدار شوند.

سه برای حل این مشکل می توان از لحاظ سخت افزاری از latch یا خازن استفاده کرد و دشارژ نباید شود و نویز کم می شود از لحاظ نرم افزاری می توان در برنامه keypad از تأخیر که بعد از فشار دادن کلید تأخیر می شود و بعد از دریافت وضعیت آن استفاده کرد. راه حل دیگر هم می توان گفت که یک مقصود همان در نظر گرفتن یک آلارم زمان بین نوسان از چند سطح شش به کمتر بود تعریف به اتصال را اطلاع دهد.

از robot → ضعیف

تعیین مختصات توانی در صفحه از صفحه کلید Keypad

→ Keypad (make key map (uses key map) row[] & col[] & rows & cols)
سه تابع پیش از Keypad می‌بازد به گونه‌ای که بتوان تعداد اوی مثل تعداد سطرها و ستون‌ها و مختصات مربوط به هر دکمه را بداند.

→ char getKey()
سه بار به نام‌های تعریف شده، داده‌ها می‌دهد که شماره‌ده Keypad را برمی‌گرداند.

→ char getKeyS()
سه بار به نام‌های که همان‌ها می‌دهد که شماره‌ده Keypad را برمی‌گرداند.

→ char waitForKey()
سه تابع به دو تابع می‌ماند که می‌دهد و در دو حالت می‌تواند به نام‌های getKey و getKeyS است. تفاوت به همراه با انتظار است.

→ KeyState getState()
سه حالت وضعیت می‌دهد که می‌تواند به نام‌های RELEASED و PRESSED باشد.

→ boolean KeyState changed()
سه تابع وضعیت می‌دهد که می‌تواند به نام‌های True و False را برمی‌گرداند. در غیر اینصورت False را برمی‌گرداند.

تابع برد ها آردوینو دارای یک پورت سریال هستند که به آن ها UART و USART نامیده می شود. در واقع این سریال ها سری های خاص و مشخص هستند. در آردوینو از سری های دیجیتال و آنالوگ برای راه اندازی پورت سریال استفاده می شود و به وسیله خط USB به پورت آردوینو به کامپیوتر متصل می شود.

سین ۱ یا TX فرستنده اطلاعات و سین ۰ یا RX برای گرفتن اطلاعات است که دارای قابلیت های مثل Interrupt و Buffer می باشد. خصوصیات آن ها از سین های ۰ و ۱ به عنوان پورت سریال استفاده می کنیم. دسترسی حق این را نداریم. از این سین ها به عنوان ورودی و خروجی دیجیتال استفاده می کنیم. کاربر های این کاربر می توان به تبادل اطلاعات و همین طور داده ها سین برد ها و دستگاه های خردی مثل اسکنر و ... استفاده می شود.

Serial Pin 0	Serial 1 Pin 8	Serial 2 Pin 16	Serial 3 Pin 14	در برد mega
0 (TX) و 1 (RX)	19 (RX) 18 (TX)	17 (RX) 16 (TX)	15 (RX) 14 (TX)	

digispark.ir
Arduino.cc

نسخه

تعریف مختصر نحوه کار توابع زیر

→ begin()

به ندرت داده را در واحد بیت بر ثانیه تنظیم می کند.

→ end()

به ارتباط برپایی را غیر فعال کرده و به RX و TX اجازه می دهد که برای درک وضعیت خود شروع به استفاده کنند.

→ ParseInt()

به دنبال یک مقیّر Integer معتبر شده و در سریال درج می گردد. (مقیّر Int از دنبال می گردد)

→ Find()

به از Serial buffer می خواند تا زمانی که هدف مورد نظر پیدا شود.

→ Println()

به چاپ کردن و به خطی می گردد. در واقع data را در Serial port به عنوان یک ASCII نوشته

→ read()

به داده ورودی را می گیرد. در واقع یک بیت می خواند و آن را برمی گرداند.

→ readStringUntil()

به کاراکترهای ورودی را از Serial buffer می خواند و به رشته تبدیل می کند.

→ write()

به داده باینری را در Serial port می نویسد. در واقع داده باینری را به یک یا چند بایت تبدیل می کند.

arduino.com

منبع 5