

کدهای **arduino** همگی در فایل آپلود شده قرار داده شده اند. همانطور که در ویدئوی توضیحی گفته شده است (برای کیپد های ماتریسی یک کلید از دو صفحه ی رسانا تشکیل شده است و در زمانی که این دو صفحه به هم متصل باشند، جریان برقرار خواهد شد و در غیر این صورت جریان قطع خواهد شد. اما به جز این دو حالت، یک حالت دیگر هم ممکن است. که همان **bouncing** است. زمانی که دو صفحه ی جدا از هم به مقدار کافی به یک دیگر نزدیک شوند، برای هوای بین آنها، پدیده ی شکست الکتریکی اتفاق می افتد که این به معنی است که مولکول های هوا میتوانند در زمان های بسیار کوتاهی در نقش یک رسانا عمل کنند که به این پدیده **Bouncing** گفته میشود. این جریان نوسانی ممکن است به قطعات مدار آسیب وارد کند. حال برای جلوگیری از این پدیده، یک خازن را به صورت موازی با کلید می بندیم. در این صورت اگر این پدیده اتفاق بیوفتد جریان نوسانی صرف شارژ شدن خازن می شود و آسیبی به دیگر قطعات نمی رساند.

**Constructor keypad:** با گرفتن نام کلیدها، شماره ی پین های سطرها و ستونها، و تعداد سطرها و ستونها، یک شی از کلاس **keypad** درست می کند

**getKey()**: کاراکتر کلید فشرده شده را برمیگرداند

**getKey()**: یک آرایه از کاراکتر کلیدهای فشرده شده را برمیگرداند

**waitForKey()**: این تابع منتظر فشرده شدن کلید می ماند. به این معنی که تا زمانی که کلیدی فشرده نشود، برنامه به خط بعدی نمی رود. همچنین در صورت فشرده شدن کلید، کاراکتر آن کلید برگردانده می شود.

**getState()**: این تابع وضعیت هر کدام از دکمه هارا بر میگرداند که می توانند 4 حالت Idle, Pressed, Released, Hold داشته باشند.

`keyStateChanged()`: این تابع اگر وضعیت یک کلید تغییر کند **true** و در غیر این صورت **false** بر میگرداند.

نحوه و کاربرد ارتباطات سریال در اردینو: ارتباطات سریال در اردینو با کمک پین های TX, RX و تحت منتطق TTL انجام می شود. از سریال برای ارتباط برد با برد های دیگر یا دستگاه های دیگر استفاده می شود.

تعریف و نحوه کار با تابع های ارتباطات سریال:

`begin()`: سرعت انتقال داده را بر حسب بیت بر ثانیه مشخص میکند.

`end()`: ارتباط سریال را میبندد .

`find()`: از serial buffer داده را تا زمانی میخواند که داده ی آرگومان را پیدا کند

اگر داده پیدا شد مقدار **true** و در غیر این صورت مقدار **false** بر می گرداند.

`parseInt()`: داده را تا هنگامی که **integer** بعدی را تشخیص دهد می خواند و در

صورت تمام شدن زمان تایمر **terminate** می شود.

`println()`: داده را به صورت متن روی سریال خروجی می نویسد.

`read()`: داده ی ورودی را می خواند.

`readStringUntil()`: کاراکتر های ورودی را از سریال بافر دریافت می کند و سپس به

**string** تبدیل می کند و اگر زمان تایمر تمام شود **terminate** می شود.

`write()`: دیتای باینری را می فرستد.