۲.

- 1x4
- 3x4
- 4x4

برای مثال در نمونه آخر ۴ ستون و ۴ ردیف وجود دارد که برای هر ردیف و هر ستون یک سیم قرار داده شده است. ستون‌ها به پین‌های ورودی و سطرها به پین‌های خروجی متصل می‌شوند. در حالت عادی ورودی و خروجی‌ها روی حالت high قرار داده می‌شوند و برای مثال درصورت فشردن کلید ۱ ۱ ابتدا سطر خروجی ۰ شده و در پی آن ستون خروجی نیز ۰ می‌شود و به این صورت مشخص می‌شود که کدام کلید فشرده شده است.

۳. پدیده نوسان هنگام فشرده شدن کلید رخ می‌دهد و در پی آن سیگنال ارسالی توسط کلید چند بار تغییر کرده که این تغییر می‌تواند به منزله چند بار فشرده شدن آن در نظر گرفته شود.

این اشکالات را به ۲ روش می‌توان برطرف نمود:

- قرار دادن یک خازن در دو طرف کلید برای جلوگیری از تغییرات
- روش بهتر در نظر نگرفتن تغییرات پیاپی و کوتاه مدت می‌باشد (انتظار تا زمان ثابت شدن سیگنال) این روش به صورت نرم‌افزاری انجام می‌شود.

۴.

- Keypad(makeKeymap(userKeymap), row[], col[], rows, cols): یک سازنده (constructor) برای کلاس keypad می‌باشد که ورودی آن کلیدهای مورد نظر، پین ردیف‌ها، پین ستون‌ها، تعداد ردیف‌ها و تعداد ستون‌ها می‌باشد.

- Char getKey(): کاراکتر مربوط به کلید فشرده شده را برمی‌گرداند (در صورتی که کلیدی فشرده نشده باشد کاراکتر خالی null برمی‌گرداند.).

- Char getKeys(): در صورتی که بیش از یک کلید همزمان فشرده شده باشند به ما خروجی true می‌دهد و در غیر این صورت false

- char waitForKey(): تا زمانی که یک کلید فشرده شود صبر کرده و کاراکتر آن را خروجی می‌دهد.

- KeyState getState(): وضعیت تمام کلیدها را به ما خروجی می‌دهد (آزاد، فشرده شد، رها شد، نگهداشته شده)

- boolean keyStateChanged (): توسط یک بولین به ما اطلاع می‌دهد که آیا وضعیت کلید تغییر کرده است یا نه

۵. ارتباطات سریال در آردواینو کاربردهای متفاوتی از جمله اتصال دو برد به یکدیگر، اتصال به کامپیوتر، اتصال به usb و ... دارد. این ارتباط از طریق پورت‌های TX, RX صورت می‌گیرد که به صورت ضربدری متصل می‌شوند (دریافت درRX و ارسال از طریق TX). همچنین پایه گراند دو دستگاه نیز باید دارای ولتاژ ۰ برابر باشند.

۶.

- Begin():
پورت سریال را باز کرده و اتصال آن را برقرار می‌کند و قابلیت ارسال و دریافت را در این پین فعال می‌کند. (آرگومان سرعت ارتباط را مشخص می‌کند که استاندارد آن ۹۶۰۰ می‌باشد.)

- End():
پین‌های فعال شده را آزاد کرده و به ارتباطات پایان می‌دهد.

- Find():
به عنوان آرگومان یک متغیر هدف را ورودی گرفته و داده‌ها را از بافر می‌خواند و درصورت مشاهده شدن هدف ورودی داده شده true و در غیر اینصورت false خروجی می‌دهد.

- parseInt():
از ورودی پین یک عدد صحیح خوانده و به ما خروجی می‌دهد. در آرگومان‌های آن قابلیت هندل کردن برخی خطاها مانند ورودی های غیر عددی و یا فضای خالی فراهم شده است.

- println():
در خروجی پورت یک رشته را که از طریق آرگومان دریافت کرده است به همراه کاراکتر خط جدید نمایش می‌دهد. همچنین می‌توان در آرگومان نحوه دیکد کردن ویا همان فرمت رشته ورودی را مشخص نمود.

- read():
اولین بایتی که به صورت ورودی وارد پین شده است را به ما برمی‌گرداند (در صورت موجود نبودن منفی ۱ خروجی می‌دهد)

- readStringUntil():
به عنوان آرگومان یک کاراکتر می‌گیرد و تا زمانی که کاراکتر مورد نظر دریافت شود ورودی پین را ذخیره کرده و به صورت یک رشته به ما خروجی می‌دهد.

- write():

یک بایت داده را از طریق خروجی ارسال می‌کند.