

• پین نزارش از ۲۰

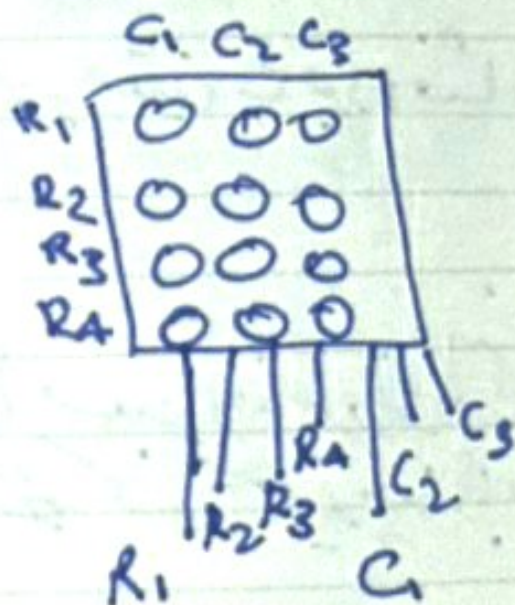
• محاسبه برای ایدیت زارده

• کد های مودرینز ← لایبرری KeyPad، استفاده از

Serial، ...

• انواع KeyPad از ماتریس، کارکردشان :

KeyPad 4x3, 4x4



• KeyPad که خارجی با

تزیین شدن انگشت خفیه می شوند
مخفی می شود کدام کلید فشار
دارد شده.

• کلید که ماتریس اگر فشار داده شوند، آن سطر بتوان

به هم وصل می شوند می توان کلید را تنخفیه دارد.

به صورت دیفالت Row را به OUTPUT میکرو وصل می کنند و HIGH را
می کنند. Column را به INPUT وصل می کنند (INPUT-PULLUP)

• برای Scanning و پیدا کردن Key فست رده شده ،
مدرس طرا را یکبار LOW می کنند ، اگر column ای low
شده بود یعنی کلیدی که در سطر ۲ و ستون ۷ برده فشار
داده شده .

• Bounce : فست را این کلید سبب نویز و نویزهای
که ممکن است این برداشت شود که کلید چند بار
فشار داده شده است (خطا)

راه حل ۱ : استفاده از خازن در سمت کلید (چون کلید
زیاد است منطقی نیست)

② خوراکهای جانبری Keypad یک de bouncing
de la
که باید Key حداقل مدت زمان مشخص فشرده شده باشد تا


14.

③ استفاده از catch که پس از throw key را می‌تواند

① KeyPad (makeKey...)
map to ~ KeyPad as constructor

ناراکه دکمه، { pin } سطح و ستون و تعداد
سطح و ستون را می گیرد و keypad object به ما می دهد.

① `char getKey()` → `non-blocking` مانند `get`
در صورت فش رادر شدن `key` ، `char`
آن را برمی گرداند.

حَدَّث می کند عذرمان \rightarrow `getKeys()` | `bool`  چند کلید مفاد داده شده اند یا خیر.

فانتسی block
ing که صبر می کند → char waitforkey() کلید منت را دارد

④ `KeyState getState()` → استیت key را برمی گرداند

• IDLE, PRESSED, RELEASED, HOLD

⑤ `bool keyStateChanged` → در صورت تغییر state

• `no` & `true` & key

`parseInt()` → `validInt` ورودی به دنبال

ورودی می‌گیرد. می‌توان به آن `lookahead` دار که ۳ بایت

`SKIP_WHITESPACE`, `SKIP_NONE`, `SKIP_ALL`

دارد. `ignore` نیز می‌توان دارد. مثلاً اگر که ۱,۰۰۰

عدد را بخواند

`println()` → `print` می‌کند (بایت)، `ASCII` به فرم

`read()` → (Byte) دیتای ورودی از `serial` را می‌خواند

`readStringsUntil()` → `terminator` که می‌دهیم
می‌خواند، `return string` می‌کند

`write()` → دیتا را به فرم `byte` ارسال می‌کند