

ساخت ماشین حساب با RPi Pico

GitHub: [SAMBSIREX/Rpi-pico-Calculator](https://github.com/SAMBSIREX/Rpi-pico-Calculator): DLY Calculator which Rpi pico

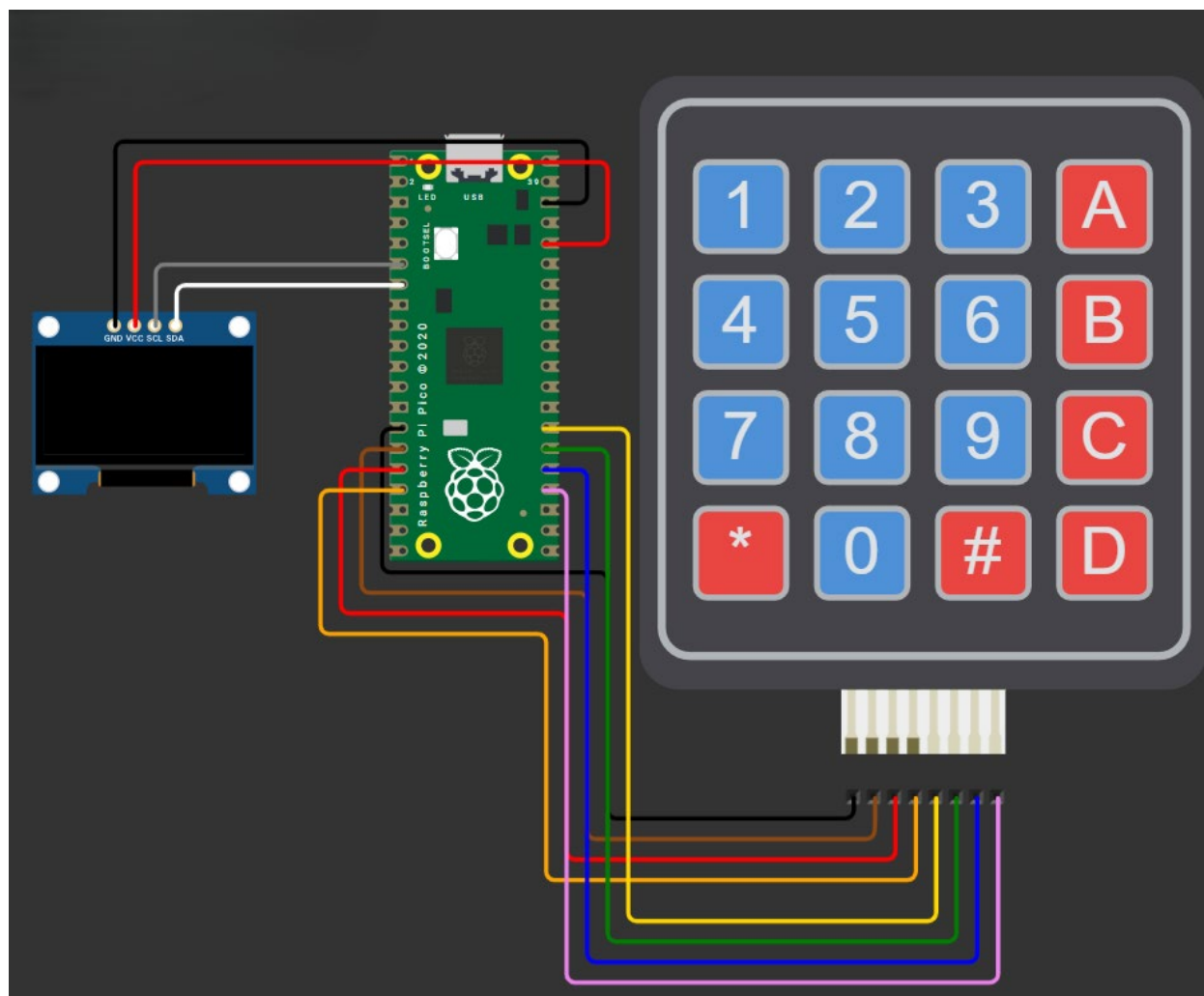
قطعات:

کافه ربات | Pico قیمت و خرید برد رسیبری یای (رزبری یای) پیکو

I2C تک رنگ 0.91 اینچ ارتباط OLED قیمت و خرید ماژول نمایشگر

کیبورد | کافه ربات matrix قیمت و خرید کی پد فلت 4 در 4 ماتریسی

شماتیک اتصالات:



کد پایتون:

```
#https://github.com/SAMSBSIREX/Rpi-pico-Calculator  
#Rpi-pico-Calculato
```

```
#Monitor >>> Rpi Pico  
#GND >>> 38 GND  
#VCC >>> 36 3V3(OUT)  
#SCK >>> 7 GP4  
#SDA >>> 6 GP5
```

```
#Keypad 4x4 Matrix >>> Rpi Pico  
#R1 >>> 14 GP10  
#R2 >>> 15 GP11  
#R3 >>> 16 GP12  
#R4 >>> 17 GP13  
#C1 >>> 27 GP21  
#C2 >>> 26 GP20  
#C3 >>> 25 GP19  
#C4 >>> 24 GP18
```

```
from machine import Pin, I2C  
import OLED  
import utime
```

```
i2c = I2C(0, scl=Pin(5), sda=Pin(4))  
oled = OLED.SSD1306_I2C(128, 32, i2c)
```

```
matrix_keys = [['1','2','3','A'],  
                ['4','5','6','B'],  
                ['7','8','9','C'],  
                ['*','0','#','D']]
```

```
rows = [10,11,12,13]  
cols = [21,20,19,18]
```

```
row_pins = [Pin(r, Pin.OUT) for r in rows]  
col_pins = [Pin(c, Pin.IN, Pin.PULL_DOWN) for c in cols]
```

```
oled.fill(0)  
oled.text("@SAMSBSIREX",1,1)  
oled.text("Press any Key",1,16)  
oled.show()  
utime.sleep(0.5)
```

```

def scankeys():
    for i, row in enumerate(row_pins):
        row.high()
        for j, col in enumerate(col_pins):
            if col.value():
                key = matrix_keys[i][j]
                utime.sleep(0.3)
                row.low()
                return key
        row.low()
    return None

```

```

expr = ""

```

```

while True:
    key = scankeys()
    if key:
        if key == "*":
            expr = ""
        elif key == "#":
            try:
                result = str(eval(expr))
                expr = result
            except:
                expr = "Error"
        elif key in "ABCD":
            if key == "A": expr += "+"
            if key == "B": expr += "-"
            if key == "C": expr += "*"
            if key == "D": expr += "/"
        else:
            expr += key

    oled.fill(0)
    oled.text(expr, 1, 16)
    oled.show()

    utime.sleep(0.1)

```

ویدیوی پروتوتایپ:

<https://drive.google.com/file/d/14QvRLHgqntjoH8li0oQVaMROSLNQgFqC/view?usp=sharing>

راهنما نحوه لهیم کاری پرتوتایپ:

چون قطعات ماژولار هستند، نکات کوتاه و عملی:

ابتدا فقط قطعات اصلی و کوچک را لحیم کن (مثل برد پیکو)

کانکتورها، سوکت‌ها و هدرهای مادگی را آخر لحیم کن تا ماژول‌ها قابل جداسازی و تعویض بمانند

برای و ماژول‌های حساس از سوکت استفاده کن تا نیاز به لحیم‌زدن مستقیم نباشد

قبل از لحیم‌کردن هدرها، مکان و جهت ماژول‌ها را تست کن و مطمئن شو پین‌ها هم‌ترازند

هنگام تست نهایی، ابتدا فقط برد را بدون ماژول‌ها روشن کن و ولتاژهای کلیدی را اندازه بگیر، سپس ماژول‌ها را یکی‌یکی متصل کن و تست کن

نحوه کار:

این برنامه یک حلقه اصلی دارد که به‌صورت پیوسته منتظر فشردن کلید از کلید ماتریسی است. برای خواندن کلید، هر سطر به‌طور متناوب فعال می‌شود و ستون‌ها خوانده می‌شوند؛ اگر در تقاطع سطر فعال و یک ستون، اتصال برقرار باشد یعنی کلیدی فشرده شده است و برنامه نماد آن کلید را از ماتریس نگاشت کلیدها می‌گیرد

پس از تشخیص کلید، برنامه آن را به رشته‌ای که عبارت فعلی را نگه می‌دارد اضافه می‌کند یا بر اساس نوع کلید رفتار مناسب را انجام می‌دهد: کلید پاک کردن رشته را خالی می‌کند، کلید محاسبه تلاش می‌کند عبارت را محاسبه کند و نتیجه یا پیام خطا را جایگزین عبارت می‌کند، و کلیدهای ویژه به نمادهای عملیاتی تبدیل می‌شوند (جمع، تفریق، ضرب، تقسیم، توان، باقی مانده)

برای جلوگیری از خواندن‌های ناخواسته هنگام فشردن، پس از شناسایی کلید یک مکث کوتاه انجام می‌شود تا از ثبت چندباره یک فشردن جلوگیری شود. پس از هر تغییر در عبارت، نمایشگر پاک می‌شود و متن عبارت یا نتیجه در موقعیت تعیین‌شده نمایش داده می‌شود

محاسبه عبارت با استفاده از یک تابع ارزیابی رشته‌ای انجام می‌شود؛ اگر عبارت نادرست یا موجب خطا (مثل تقسیم بر صفر) شود، برنامه آن را گرفته و به‌جای نتیجه عبارت "خطا" را نمایش می‌دهد. به همین دلیل نگارش و نحوه وارد کردن عملگرها و اعداد باید صحیح باشد

سیم‌بندی سخت‌افزاری به‌صورتی است که سطرها به پایه‌های خروجی و ستون‌ها به پایه‌های ورودی با حالت کشش-پایین متصل شده‌اند؛ وقتی سطری بالا می‌رود و کلیدی فشار داده شده، پایه ستون متناظر مقدار منطقی یک می‌گیرد و این حالت توسط برنامه تشخیص داده می‌شود

در مجموع، جریان کار به این ترتیب است: اسکن کلید <<< شناسایی کلید <<< به‌روزرسانی عبارت یا انجام عملیات مخصوص <<< نمایش عبارت یا نتیجه <<< تکرار