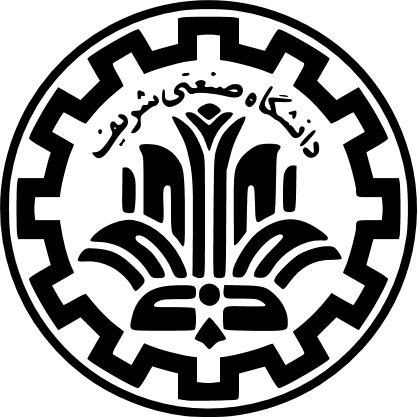
****

**آزمایشگاه سخت‌افزار**

**فاز چهارم پروژه لاجیک آنالیزر**

**نرم‌افزاری و تستر دیجیتال**

استاد: دکتر اجلالی

تیم شماره ۷ - اعضای تیم:

مهرداد صابری - ۹۷۱۱۰۱۳۳

محمد مهدوی - ۹۷۱۱۰۲۲۸

مسیح اسکندر - ۹۷۱۰۵۷۳۶

# **۱ هدف فاز و کارهای انجام شده**

در فاز چهارم یا نهایی از پروژه می‌خواهیم که بخش تستر دیجیتال برنامه‌ را پیاده‌سازی کنیم. در این بخش کاربر باید بتواند دو سیگنال را به عنوان خروجی‌های آردوینو در محیط نرم‌افزاری انتخاب کند. برای مشاهده‌ی کارایی این بخش، این دو خروجی را به یک مدارمجتمع می‌دهیم و خروجی حاصل از انجام عملیات را به ورودی آردوینو می‌دهیم تا به کمک لاجیک‌آنالایزر در نرم‌افزارمان نشان داده‌شود.

علاوه بر این، دو تغییر دیگر نیز در این فاز از پروژه داده‌شده است که اولی اضافه کردن گزینه‌ی رنگ‌بندی یکی در میان به نمودارها است به صورتی که هر دو نمودار مجاور رنگ متفاوتی داشته باشند تا تمایزدادنشان ساده‌تر گردد. همچنین حداکثر مقدار زوم کردن روی سیگنال‌ها تا مشاهده‌ی یک کلاک افزایش یافته‌است.

# **۲** **کدها**

به عنوان روشی که کاربر بتواند سیگنال‌های خروجی دلخواه آردوینواش را مشخص کند، یک جعبه متنی در نظر گرفته‌ایم که کاربر با نوشتن یک رشته از 0 و 1 در آن و زدن یکی از دکمه‌های Set Out1 یا Set Out2 می‌تواند این رشته‌ی سیگنال را به یکی از خروجی‌های اول یا دوم اختصاص دهد. شبه‌کد مربوط به این بخش در زیر قابل مشاهده است.

def set\_output\_textbox():

    def submit\_output(index):

        global output\_signals, output\_inds

        out\_text = text\_var.get()

        text\_var.set("")

        if not out\_text or not re.match("[01]\*", out\_text):

            return

        output\_signals[index] = [int(c) for c in out\_text]

        output\_inds[index] = 0

        init\_plot()

    output\_label = Label(window, text="Enter Output Signal Sequence", font=font.Font(size=20))

    text\_var = StringVar()

    output\_textbox = Entry(window, textvariable=text\_var, font=font.Font(size=20))

    sub\_btn1 = Button(window, text = "Set Out1", command = lambda : submit\_output(0), font=font.Font(size=20), bg='#0052cc', fg="#ffffff")

    sub\_btn2 = Button(window, text = "Set Out2", command = lambda : submit\_output(1), font=font.Font(size=20), bg='#0052cc', fg="#ffffff")

    output\_label.place(x=180, y= height - 300, height=50, width=500)

    output\_textbox.place(x=700, y= height - 300, height=50, width=1600)

    sub\_btn1.place(x=2320, y=height - 300, height= 50, width=150)

    sub\_btn2.place(x=2500, y=height - 300, height= 50, width=150)

حال باید نمودار این دو سیگنال مشخص شده توسط کاربر را در کنار بقیه‌ی سیگنال‌ها نمایش دهیم. برای اینکار در تابع init\_plot() که برای هربار آپدیت شدن نمودار‌ها صدا زده می‌شود، دو سیگنال جدیدمان را نیز به شکلی که در کد زیر است اضافه می‌کنیم.

out\_colors = ["fuchsia", "indigo"]

for i in range(len(output\_signals)):

    x\_out = [j for j in range(len(output\_signals[i]) + 1)]

    ydata = output\_signals[i][output\_inds[i]:] + output\_signals[i][:output\_inds[i]]

    axs\_step += [axs[ind].step(x\_out, [ydata[0]] + ydata)[0]]

    axs[ind].set\_yticks([0,1])

    axs\_step[ind].set\_color(out\_colors[i % 2])

    axs[ind].set\_ylabel("Output signal{}".format(

        i + 1), rotation=0, fontsize=20, labelpad=50, horizontalalignment="center")

    ind += 1

در نهایت نیز باید پس از مقدار دهی شدن سیگنال‌های خروجی، در هر کلاک مقدار این دو سیگنال در زمان کنونی را برای آردوینو ارسال کنیم. برای اینکار یک آرایه‌ای دو عضوی output\_inds تعریف می‌کنیم که نشان می‌دهد در کلاک کنونی در حال پردازش کدام عضو از آرایه‌ای هرکدام از سیگنال‌های خروجی هستیم. این اندیس‌ها از مقدار ۰ در ابتدای برنامه شروع می‌شوند و به صورت دوری در هر کلاک یک واحد به جلو می‌روند.

بنابراین کد زیر را به تابع update\_signals() که در ریسه‌ی جدایی از برنامه اجرا می‌شود و وظیفه‌ی ارتباط برنامه با آردوینو را دارد اضافه می‌کنیم.

num\_to\_write = str(sum((2\*\*i)\*output\_signals[i][output\_inds[i]]

for i in range(len(output\_signals))))

for i in range(len(output\_inds)):

    output\_inds[i] = (output\_inds[i] + 1) % len(output\_signals[i])

ser.write(num\_to\_write.encode())

در نهایت نیز برای اضافه کردن ترکیب رنگی‌های متفاوت به نمودار‌های برنامه، یک دکمه با نام Color قرار می‌دهیم که با کلیک روی آن ترکیب رنگی تغییر می‌کند. در کل سه ترکیب رنگی (آبی، قرمز)، (سبز، بنفش)، و (مشکی، نارنجی) برای نمودارها به صورت یکی در میان داریم.

colors = [['red', 'blue'], ['green', 'purple'], ['black', 'orange']]

current\_color = 0

def set\_color\_button():

    def switch\_color():

        global current\_color, colors

        current\_color = (current\_color + 1) % len(colors)

        init\_plot()

    color\_button = Button(window, text="Color", command=switch\_color,

                         bg='#0052cc', fg='#ffffff', font=font.Font(size=20))

    color\_button.place(x=50, y=height // 2 + 300, height=100, width=100)

برای نمایش رنگی نمودارها نیز خط زیر برای هر نمودار در تابع init\_plot() اجرا می‌شود.

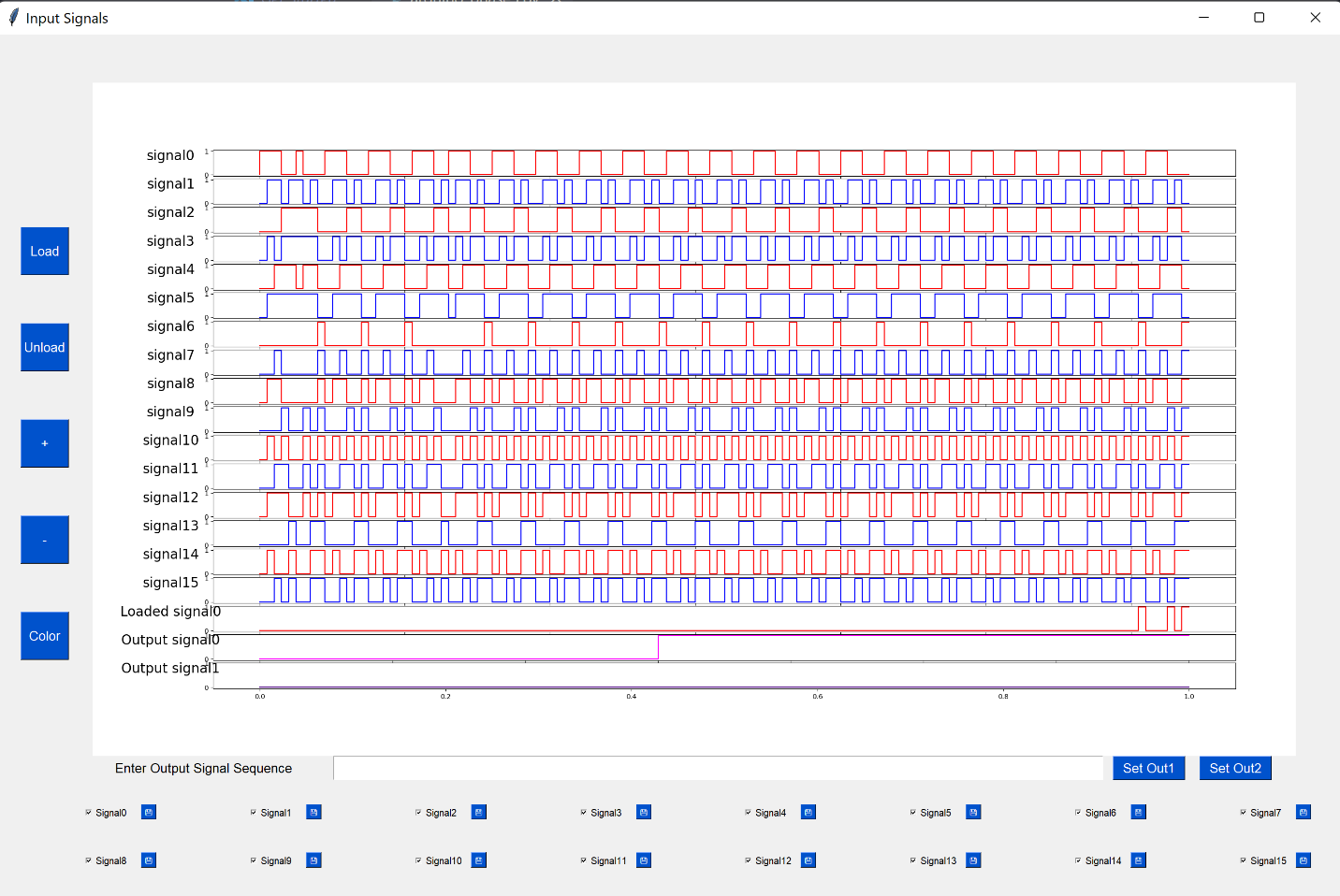
axs\_step[ind].set\_color(colors[current\_color][ind % 2])

# **۳ محیط کاربری جدید نرم‌افزار**



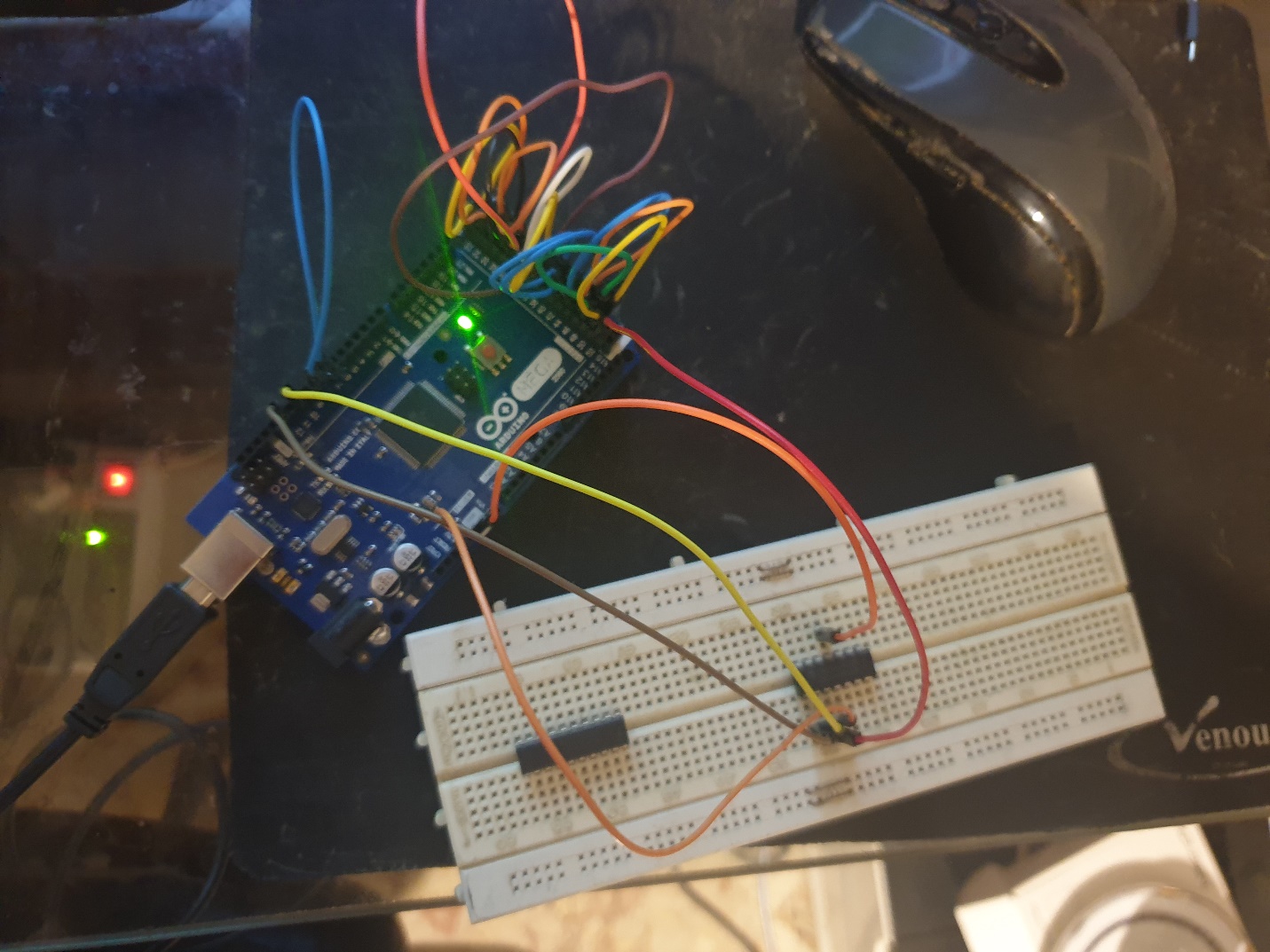
شکل 1 – جعبه متنی و دکمه‌های مقداردهی سیگنال‌های خروجی آردوینو

همانطور که در بخش قبل توضیح داده‌شد، با وارد کردن رشته‌ای از 0 و 1 در جعبه‌ی متنی شکل 1 و زدن یکی از دکمه‌های Set Out1 و Set Out2 می‌توانیم خروجی‌های آردوینو را مقداردهی کنیم.



شکل 2 – نمای کلی نرم‌افزار

همانطور که در شکل 2 قابل مشاهده‌است، دو سیگنال Output signal0 و Output signal1 به نمودار‌ها اضافه گشته‌اند. همچنین دکمه‌ی Color برای تغییر ترکیب رنگی نمودار‌ها نیز به برنامه افزوده‌شده.



شکل 3 – تست برنامه با استفاده از مدار مجتمع

در شکل 3 تصویر فیزیکی آردوینو، بردبورد و مدار مجتمع استفاده‌شده آمده است. در این تصویر دو سیگنال خروجی به مدارمجتمع 7400 که عملیات NAND انجام می‌دهد داده‌شده‌اند و خروجی قطعه به عنوان ورودی به آردوینو برگردانده شده‌است.