



آزمایشگاه سخت افزار

فاز چهارم پروژه لاجیک آنالیزر
نرم افزاری و تستر دیجیتال

استاد: دکتر اجلاّلی

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 مدار مجتمع می‌دهیم و خروجی حاصل از انجام عملیات را به ورودی آردوینو می‌دهیم تا به کمک لاجیک آنالایزر 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 تمایز دادنشان 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 آردوینویش 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 ریشه‌ی 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.

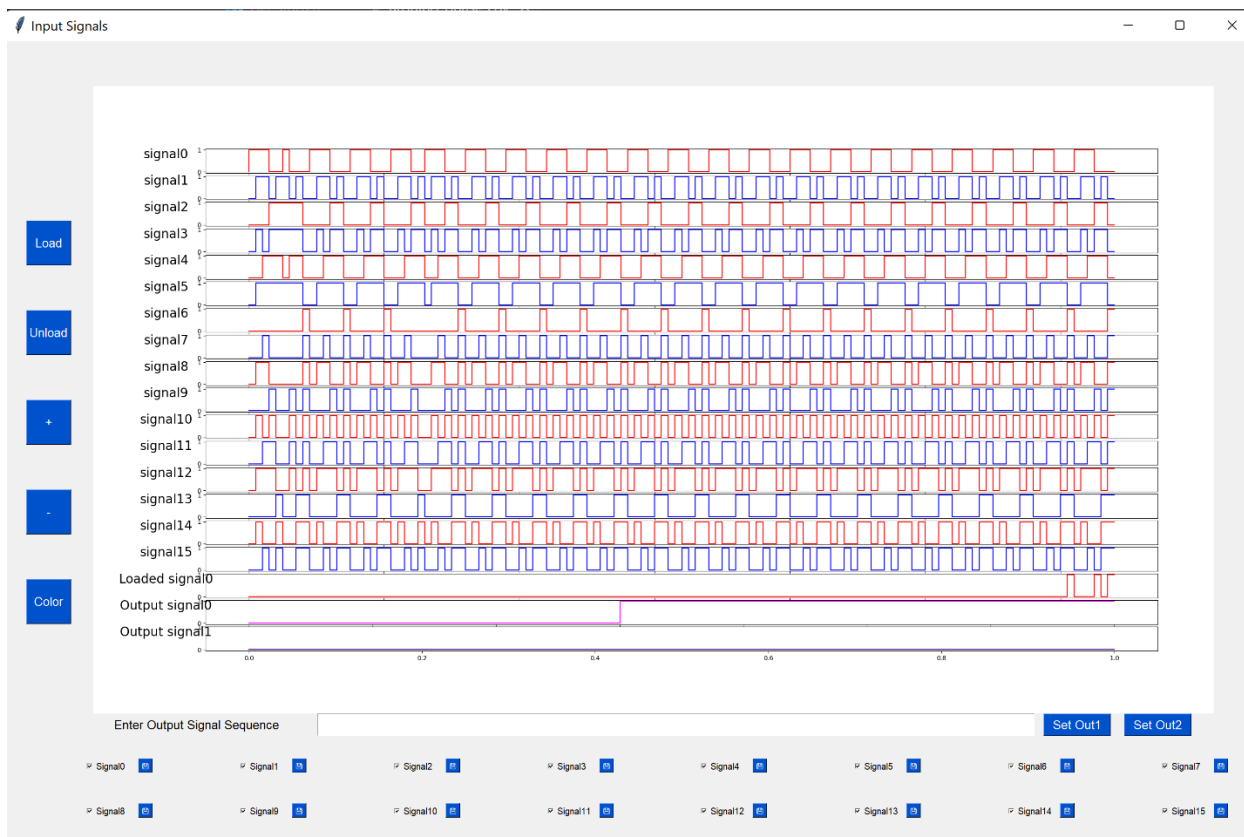
```
axs_step[ind].set_color(colors[current_color][ind % 2])
```

۳ محیط کاربری جدید نرم‌افزار

Enter Output Signal Sequence
1100111|
Set Out1
Set Out2

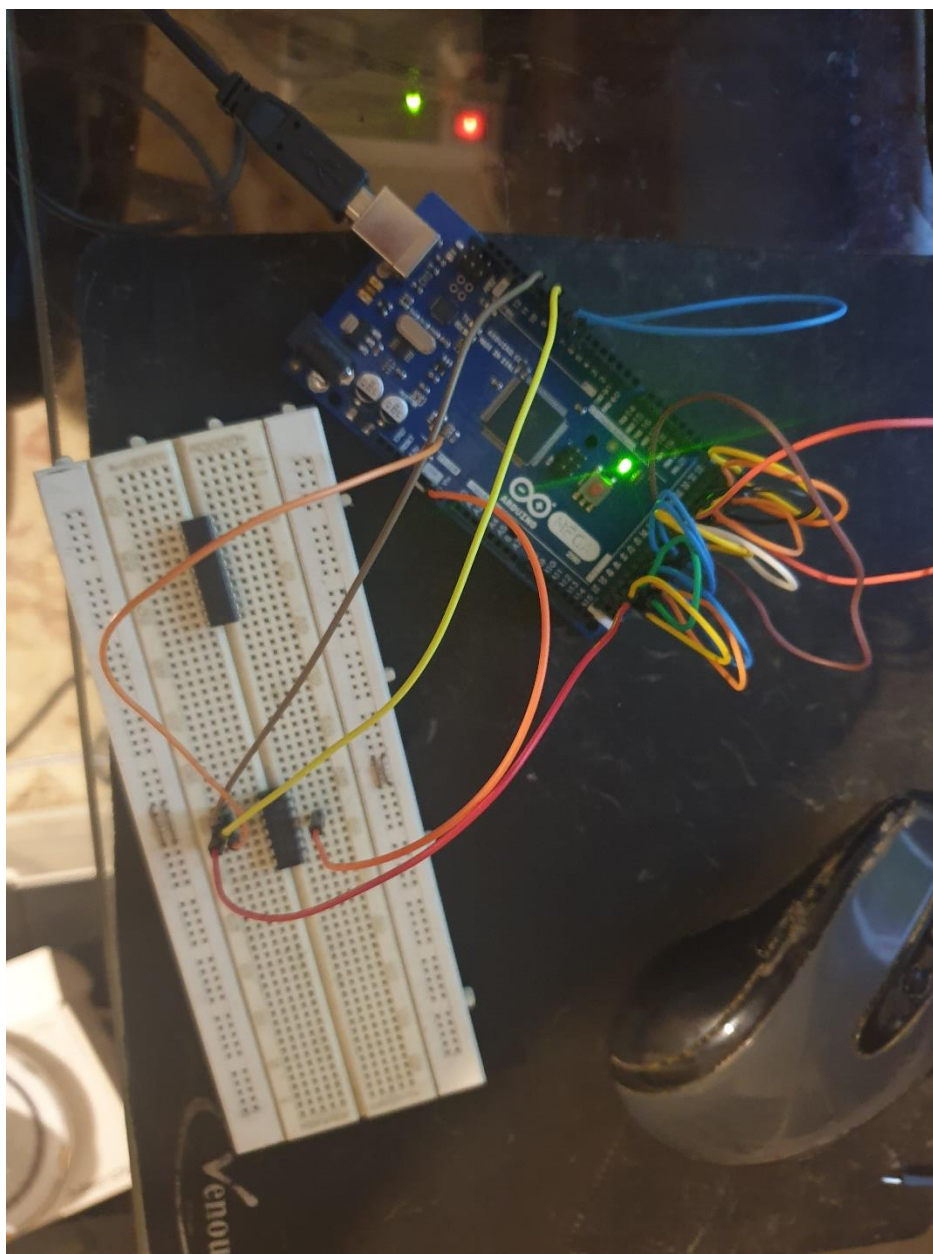
شکل 1 - جعبه متنی و دکمه‌های مقداردهی سیگنال‌های خروجی آردوینو

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 نرم افزار

همانطور که در شکل 2 قابل مشاهده است، دو سیگنال Output signal1 و Output signal0 به نمودارها اضافه گشته اند. همچنین دکمه ی Color برای تغییر ترکیب رنگی نمودارها نیز به برنامه افزوده شده.



شکل 3 - تست برنامه با استفاده از مدار مجتمع

در شکل 3 تصویر فیزیکی آردوینو، بردبورد و مدار مجتمع استفاده شده آمده است. در این تصویر دو سیگنال خروجی به مدار مجتمع 7400 که عملیات NAND انجام می دهد داده شده اند و خروجی قطعه به عنوان ورودی به آردوینو برگردانده شده است.