



آزمایشگاه سخت افزار

فانکشن ژنراتور و اسیلوسکوپ نرم افزاری
گزارش اول

دانشکده مهندسی کامپیوتر
دانشگاه صنعتی شریف
نیم سال دوم ۰۱ - ۰۰

استاد:
جناب آقای دکتر اجلالی

اعضای تیم:
محمد مهدی جراحی - ۹۷۱۰۵۸۴۴
نگین جعفری - ۹۷۱۰۵۸۵۵
مهسا امانی - ۹۷۱۰۵۷۶۹



فهرست مطالب

۱	مراحل انجام بخش اول پروژه	۲
۱.۱	سخت افزار	۲
۱.۱.۱	مدار سخت افزار	۲
۲.۱.۱	نرم افزار ریخته شده روی سخت افزار	۳
۲.۱	نرم افزار روی PC	۵

فهرست تصاویر

۱	مدار با مقاومت های ثابت	۲
۲	مدار با مقاومت متغیر یا ولوم	۳
۳	مقادیر ارسال شده به کمک سریال (مشاهده در نرم افزار آردوینو)	۴
۴	رسم نمودار اعداد ارسال شده به کمک سریال	۷

فهرست برنامه ها

۳	Hardware/sketch ۰/sketch ۰.ino	
۵	Software/requirements.txt	
۵	Software/arduino_serial.py	
۵	Software/main.py	
۶	Software/plotter.py	

۱ مراحل انجام بخش اول پروژه

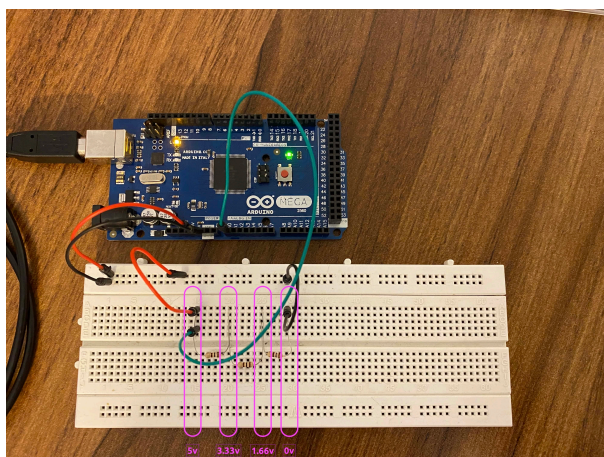
پروژه ما از دو بخش تشکیل می‌شود: بخش نمونه‌برداری (سخت‌افزاری) و بخش رسم نمودار (نرم‌افزاری). در زیر به توضیح کارهای انجام‌شده‌ی سخت‌افزار و نرم‌افزار به صورت جداگانه می‌پردازیم.

۱.۱ سخت‌افزار

سخت‌افزار ما از یک برد آردوئینو به عنوان میکروکنترلر، مداری در کنار آردوئینو، و کد ریخته‌شده روی آردوئینو تشکیل شده‌است.

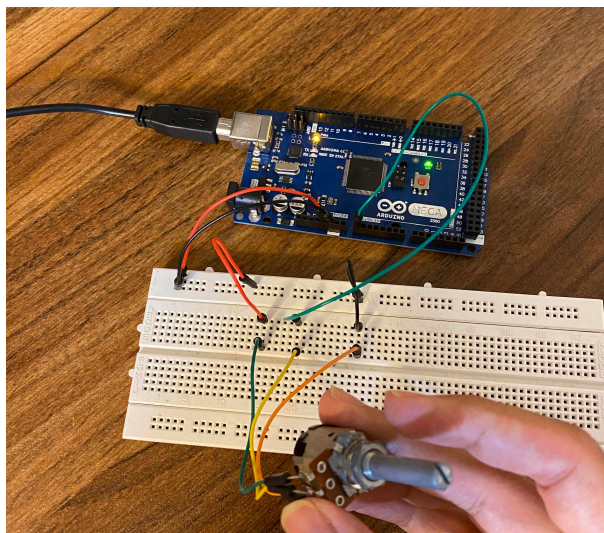
۱.۱.۱ مدار سخت‌افزار

در ابتدا ما مدار را به کمک مقاومت‌های ثابت ساختیم.



شکل ۱: مدار با مقاومت‌های ثابت

سپس ما مدار را به کمک مقاومت متغیر یا همان ولوم ساختیم.



شکل ۲: مدار با مقاومت متغیر یا ولوم

در این مدارها، ما صرفاً پایه‌ی میانی مقاومت متغیر را به ورودی ADC آردوینو وصل کرده‌ایم.

۲.۱.۱ نرم‌افزار ریخته‌شده روی سخت‌افزار

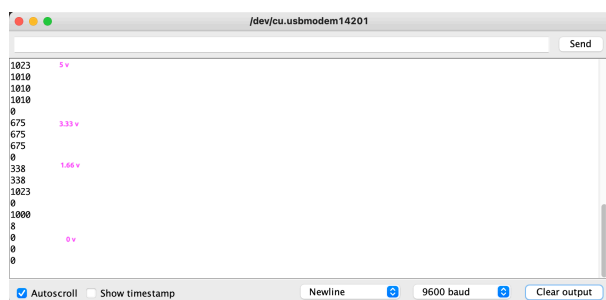
بر روی آردوینو کد ساده‌ای ریخته‌ایم که هر ثانیه ورودی ADC را بخواند و از طریق Serial ارسال کند. کد به شرح زیر است.

```
1 void setup() {  
2   Serial.begin(9600);           // setup serial  
3 }  
4  
5 void loop() {  
6   // put your main code here, to run repeatedly:  
7   delay(1000);  
8  
9   int val = analogRead(0);      // read the input pin  
10  Serial.println(val);          // debug value  
11 }
```

در تابع setup پورت سریال را باز کرده و نرخ داده را روی ۹۶۰۰ bps تنظیم می‌کنیم.

در تابع loop نیز هر یک ثانیه یک بار مقدار آنالوگ را از روی ورودی ۰ می‌خوانیم و سپس آن را در پورت سریال می‌نویسیم.

خروجی زیر مقادیر چاپ شده در پورت سریال پس از اجرای برنامه می‌باشد.



شکل ۳: مقادیر ارسال شده به کمک سریال (مشاهده در نرم افزار آردوینو)



۲.۱ نرم افزار روی PC

برای نوشتن برنامه‌ای که بتواند داده‌ها را از پورت سریال خوانده و نمودار زنده از روی آن رسم کند، نیاز به دو کتابخانه زیر داریم.

```
1 pyserial
2 matplotlib
```

کتابخانه‌ی pyserial برای دسترسی به پورت سریال استفاده می‌شود.

کتابخانه‌ی matplotlib نیز به طور کلی برای رسم نمودار در زبان برنامه‌نویسی پایتون مورد استفاده قرار می‌گیرد. به کمک این کتابخانه می‌توان نمودارهای ثابت، پویا و یا حتی تعاملی ایجاد کرد. در این پروژه از این کتابخانه برای رسم نمودار زنده از داده‌ها استفاده خواهیم کرد.

با استفاده از قطعه کد زیر داده‌های چاپ شده روی پورت سریال را استخراج می‌کنیم.

```
1 import serial
2
3 ser = serial.Serial(port='/dev/cu.usbmodem14201', baudrate
4     =9600, timeout=5)
5
6 def read_adc():
7     return int(ser.readline().strip())
```

برای اجرای برنامه کافی است که فایل main را اجرا کنید. در فایل main نیز ابتدا تابع live-plotter صدا زده می‌شود.

```
1 from arduino_serial import *
2 from plotter import live_plotter
3
4 # plot live data
5 live_plotter()
6 # close serial port
7 ser.close()
```

در تابع live-plotter یک instance از ماژول FuncAnimation از کتابخانه‌ی matplotlib گرفته می‌شود. این ماژول با فراخوانی مکرر تابعی که در ورودی می‌گیرد، نمودار پویا رسم می‌کند. خروجی تابع plt.gcf() نمودار فعلی

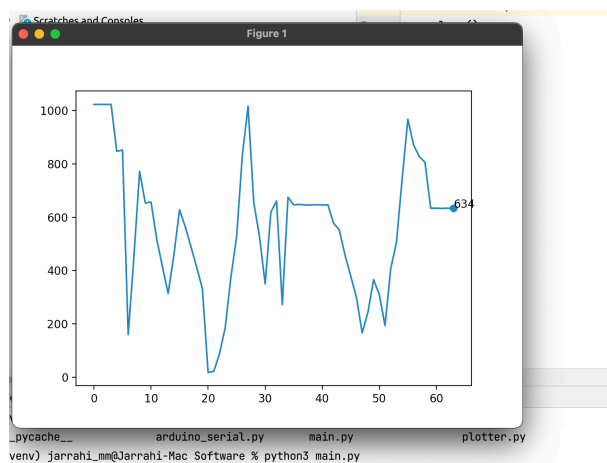


است که به عنوان ورودی ماژول گفته شده ارسال می شود.

تابع `animate` نیز همان تابعی است که ماژول گفته شده مکرراً فراخوانی می کند. در این تابع داده‌ی خوانده شده از روی پورت سریال به لیستی اضافه می شود. این لیست کمک می کند بتوانیم در فازهای بعد از داده‌های ذخیره شده برای نمایش در خروجی استفاده کنیم. سپس آن داده‌ی خوانده شده به نمودار اضافه شده و نمودار آپدیت می شود. مقدار آخرین داده نیز بر روی نمودار نوشته می شود.

```
1 import matplotlib as mpl
2 import matplotlib.pyplot as plt
3 from matplotlib.animation import FuncAnimation
4 from arduino_serial import read_adc
5
6 # for hiding buttons
7 mpl.rcParams["toolbar"] = "None"
8
9 inputs = []
10
11 # plot live data
12 def live_plotter():
13     ani = FuncAnimation(plt.gcf(), animate)
14     plt.tight_layout()
15     plt.show()
16
17
18 # animating each input data
19 def animate(i):
20     # get data
21     inputs.append(read_adc())
22     # print(inputs[-1])
23     # clear axis
24     plt.cla()
25     # plot data
26     plt.plot(inputs)
27     plt.scatter(len(inputs) - 1, inputs[-1])
28     # show the data on the plot
29     plt.text(len(inputs) - 1, inputs[-1] + 2, "{}".format(
        inputs[-1]))
```

حاصل اجرای برنامه‌ی بالا به همراه سخت افزار آردوینو را در شکل زیر مشاهده می کنید.



شکل ۴: رسم نمودار اعداد ارسال شده به کمک سریال