

## تمرین سری 5 امبدد

### سوال (1)

ADC TYPE	TIME COMPLEXITY	SPACE COMPLEXITY
FLASH-BASED ADC(N-BIT)	O(1) FOR EACH CONVERSION	O(2^N) FOR N-BIT ADC
SUCCESSIVE APPROXIMATION-BASED(N-BIT)	O(log(n)) FOR EACH CONVERSION	O(log(n)) FOR N-BIT ADC

### سوال (2)

اگر به جدول نگاه بندازیم میبینیم که min و max بازه ی مورد تایید برای تشخیص گاز توسط سنسور ما بین 300ppm تا 10 000ppm (part per million) است. در بخش دیگر datasheet بازه ی مقاومت را داریم برای سنسور است و برای یک گاز اندازه گیری میشود. این بازه بین 2kΩ و 20kΩ میباشد. شکل ها در ذیل آورده شده اند :

Concentration		300-10000ppm ( Combustible gas)
Sensing Resistance	R <sub>s</sub>	2KΩ-20KΩ(in 2000ppm C <sub>3</sub> H <sub>8</sub> )

حال برای بدست آوردن یک تابع همگر معادله زیر را حل میکنیم :

$$F(x(t)) = ax(t) + b$$

$$F(300) = 300a + b = 2k\Omega$$

$$F(10\ 000) = 10\ 000a + b = 20k\Omega$$

حال دستگاه را حل کرده و به مقدار  $b = 1445$  و  $a \approx 1.85$  میرسیم. پس طبق مدل داده شده تابع همگر ما میشود :

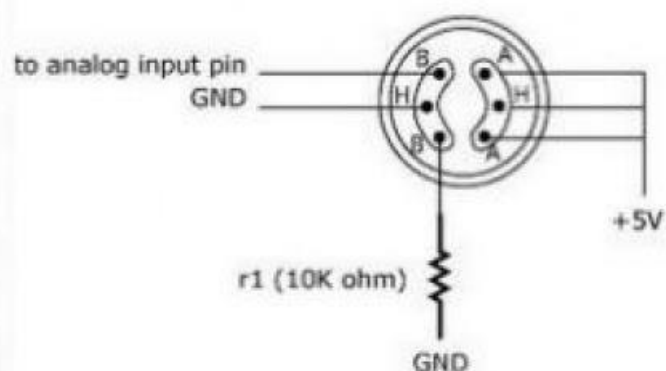
$$F(x(t)) = 1.85x(t) + 1445\Omega$$

البته توجه شود که این مدلی که ما در نظر گرفتیم برای گاز داده شده در datasheet بود که گاز پروپان ( $C_3H_8$ ) بود و بازه مقاومت ممکن است برای هر گاز متفاوت باشد. در حالت کلی بر حسب نوع گاز و environment اطراف فرق دارد میتواند ( $R_s/R_0$ ) های متفاوت را دربرگیرد.

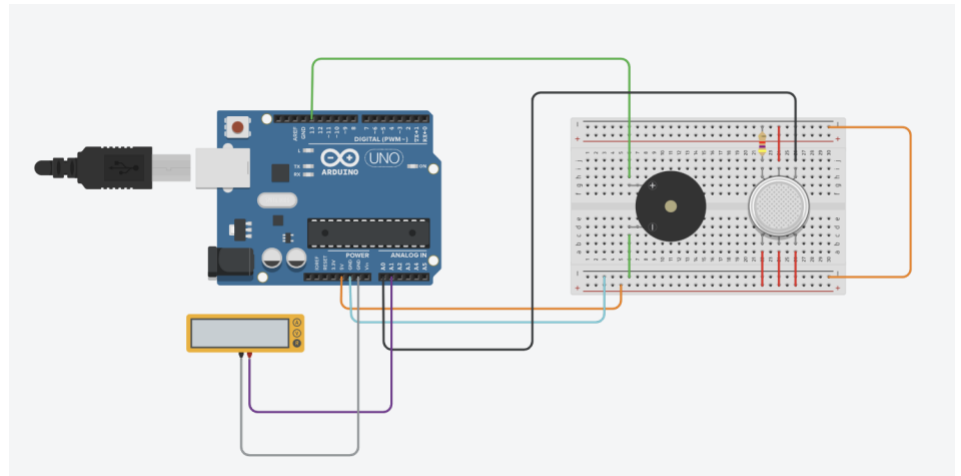
### سوال (3)

(الف)

طبق متن سوال یک سنسور داریم که باید در tinkercad شبیه سازی شود و نحوه اتصال پین های آن به صورت زیر هستند :



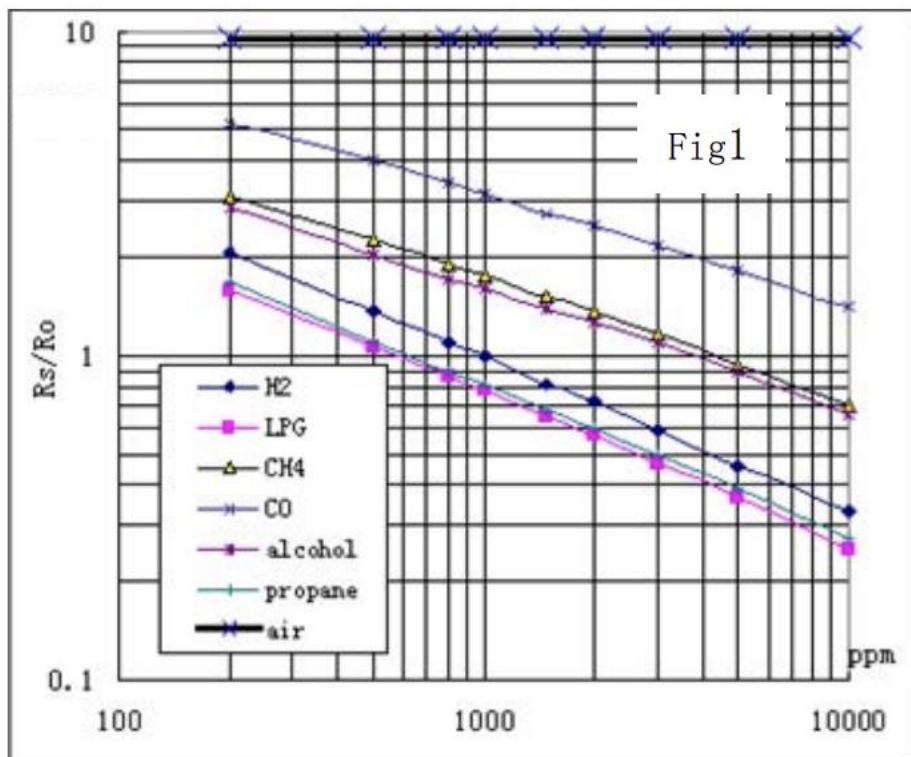
مدار خواسته شده را شبیه سازی میکنیم :



لینک :

[Q3.a](#)

حال برای تابع تبدیل همگر سنسور و کالیبره کردن آن طبق datasheet داریم :



طبق گراف و یک multimeter که به مدار وصل کردیم در شکل minimum مقدار خروجی گاز ما

با یک adc convertor مقدار

306 بوده در 300ppm و در maximum حالت مقدار 748 بوده در 10 000ppm پس طبق این دو مقدار محاسبه میکنیم و داریم :

Serial Monitor

748

Serial Monitor

306

$$F(x(t)) = ax(t) + b$$

$$F(300) = 300a + b = 306$$

$$F(10\ 000) = 10\ 000a + b = 748$$

معادله را حل میکنیم و جواب تقریبی زیر بدست میاید :

$$F(x(t)) = 0.039175x(t) + 478.247$$

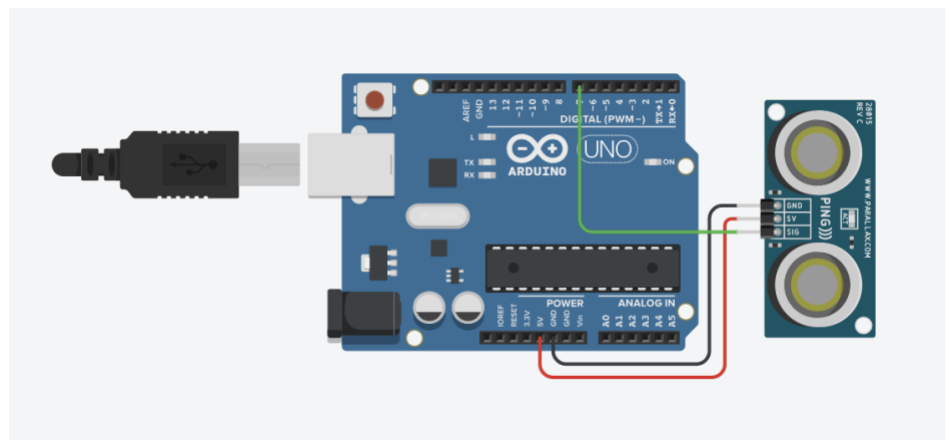
(ب)

سنسور SR04 یک نوع سنسور فاصله اولتراسونیک است که میتواند فاصله بین سنسور و یک شیء را اندازهگیری کند. این سنسور با استفاده از یک ترانسدیوسر امواج صوتی با فرکانس بالا (معمولاً حدود 40 کیلوهرتز) را از خود تابانده، که سپس از یک شیء برخورد می کند و به سنسور باز می گردد، کار می کند.

سنسور زمانی که امواج صوتی به شیء برخورد می کنند و به سنسور باز می گردند را اندازهگیری کرده و از این زمان برای محاسبه فاصله تا شیء استفاده می کند. این با ضرب زمان در سرعت صدا در هوا و تقسیم بر ۲ انجام می شود، زیرا امواج صوتی باید به شیء برخورد کرده و به سنسور باز گردند.

برای استفاده از سنسور SR04 با آردوینو، می توانید سنسور را به پین های ورودی/خروجی دیجیتال آردوینو متصل کرده و از یک برنامه ساده برای فعال سازی سنسور و خواندن اندازه گیری فاصله استفاده کنید.

پیاده سازی آن نیز به صورت زیر است :

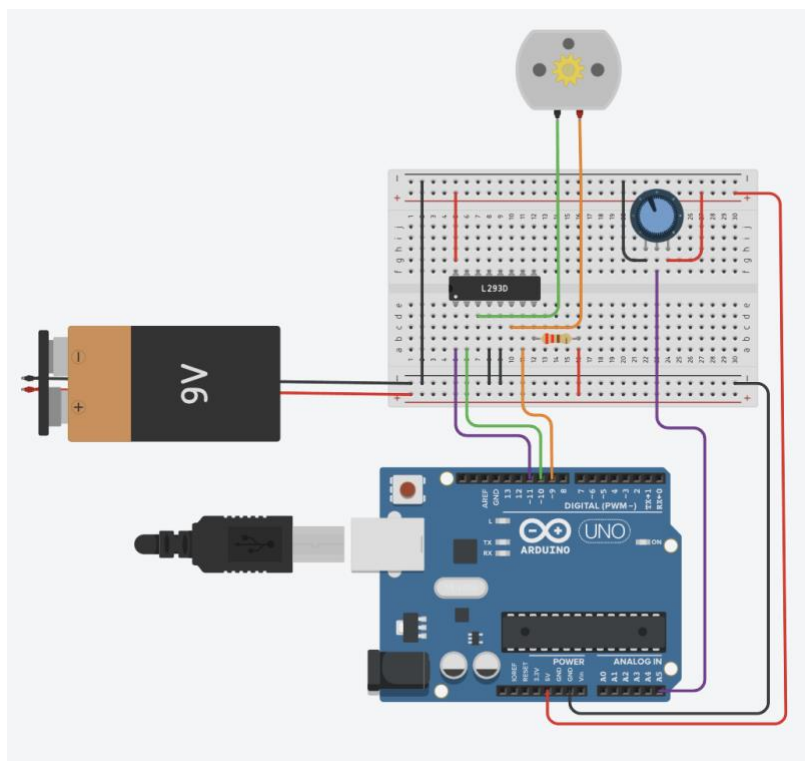


لینک :

[Q3.b](#)

(ج) مدار داده شده را با یک potentiometer و dc motor و h-bridge motor driver

پیاده سازی میکنیم. در اینجا کار پتانسیومتر فرمان عقب جلو کردن و همچنین ایستادن را انجام میدهد و 1293d در اصل با اختلاف پتانسیل ایجاد شده شدت را منتقل میکند. در شکل زیر داریم :

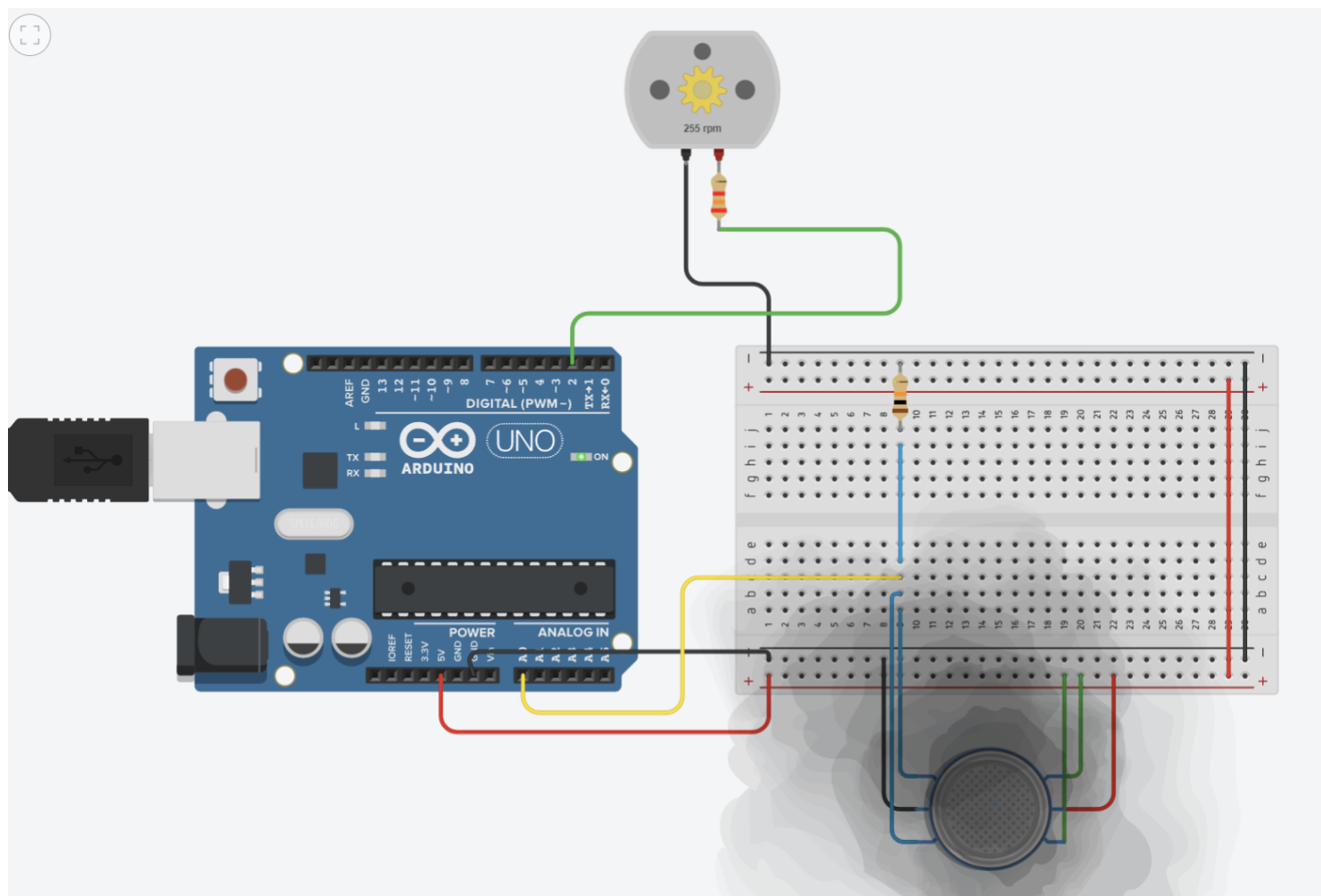


لینک :

[Q3.c](#)

(د)

همانطور که در شبیه سازی قبلی دیدیم ماکسیمم سرعت فن 255rpm و مینیمم آن 0rpm است و در شبیه سازی جدید نیز این موضوع را رعایت کرده ایم. طبق شکل زیر داریم :



اکنون با توجه به اینکه تابعی که در بخش الف محاسبه شد، اعدادی بین ۰ تا ۱۰۲۳ تولید میکند (در واقعیت با خواندن مقدار خروجی سنسور از طریق آردینو و استفاده از ADC داخلی آن که ۱۰ بیتی است، این بازه عدد تولید میشود) و با توجه به اینکه سرعت موتور اعدادی بین ۰ تا ۲۵۵ را ورودی میگیرد، یعنی ۲۵۵ بیشترین سرعت موتور را تولید میکند و کمترین سرعت نیز با مقدار ۰ حاصل میشود. بنابراین نیاز است تابع زیر استفاده شود تا مقادیر بین ۰ تا ۱۰۲۳ را به مقادیر بین ۰ تا ۲۵۵ نظیر کنیم:

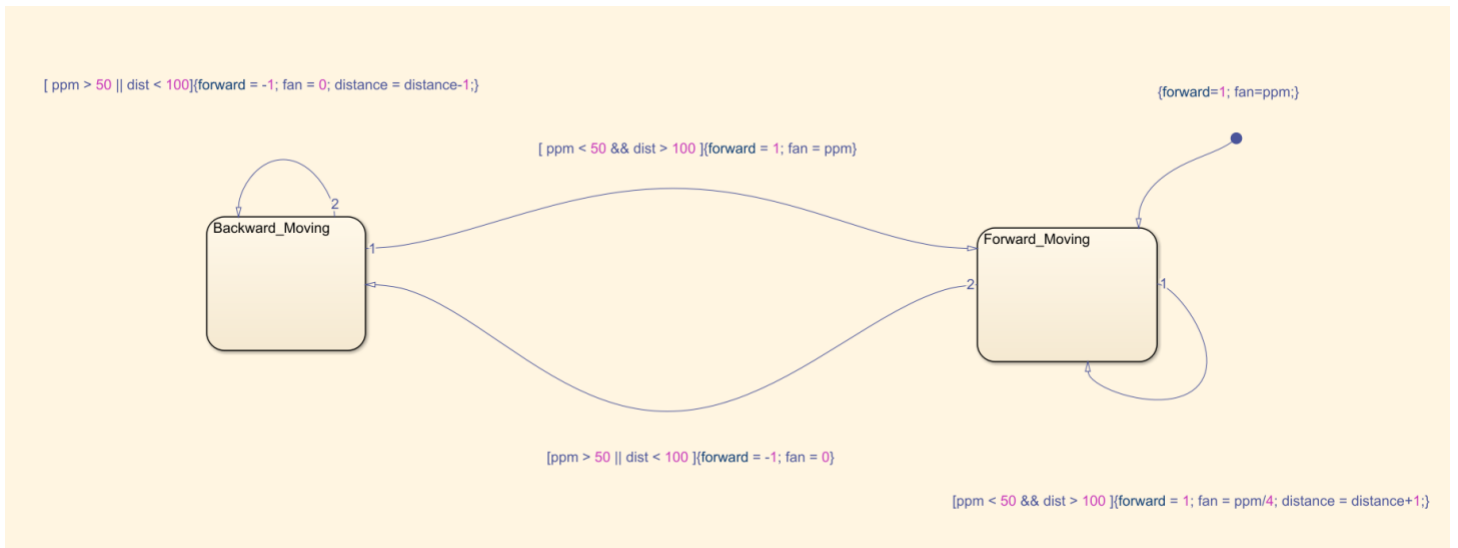
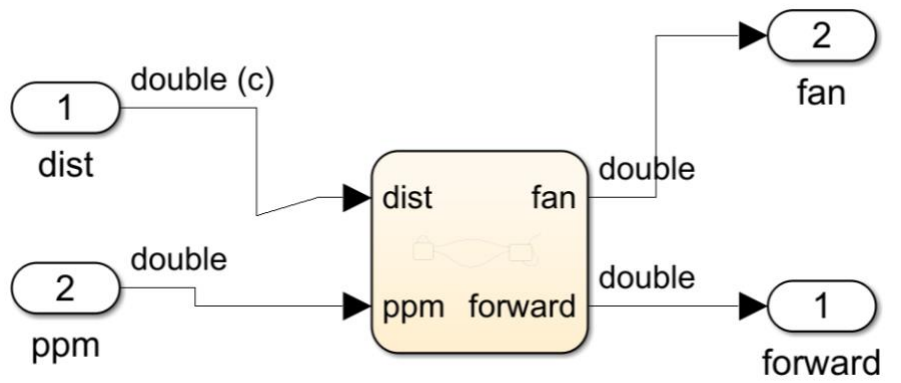
$$F(x) = \frac{x}{1024} * 256 = \frac{x}{4}$$

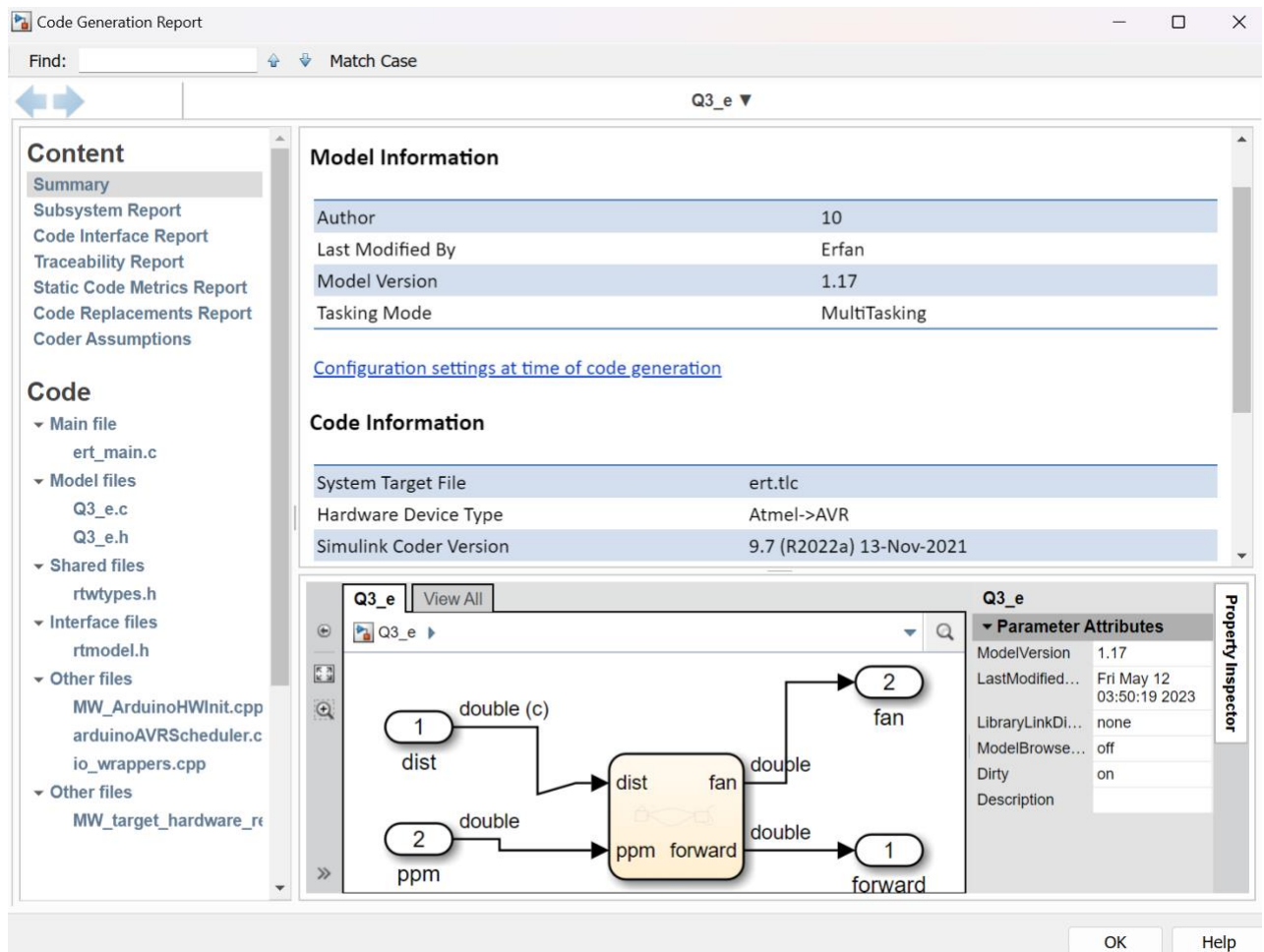
لینک :

[Q3.d](#)

(۵)

در Simulink مدار خواسته شده را با دو ورودی dist و ppm برای فاصله و غلظت داریم و خروجی برای فن و حرکت به جلو یا عقب را نیز داریم. شروط نیز آورده شده اند.

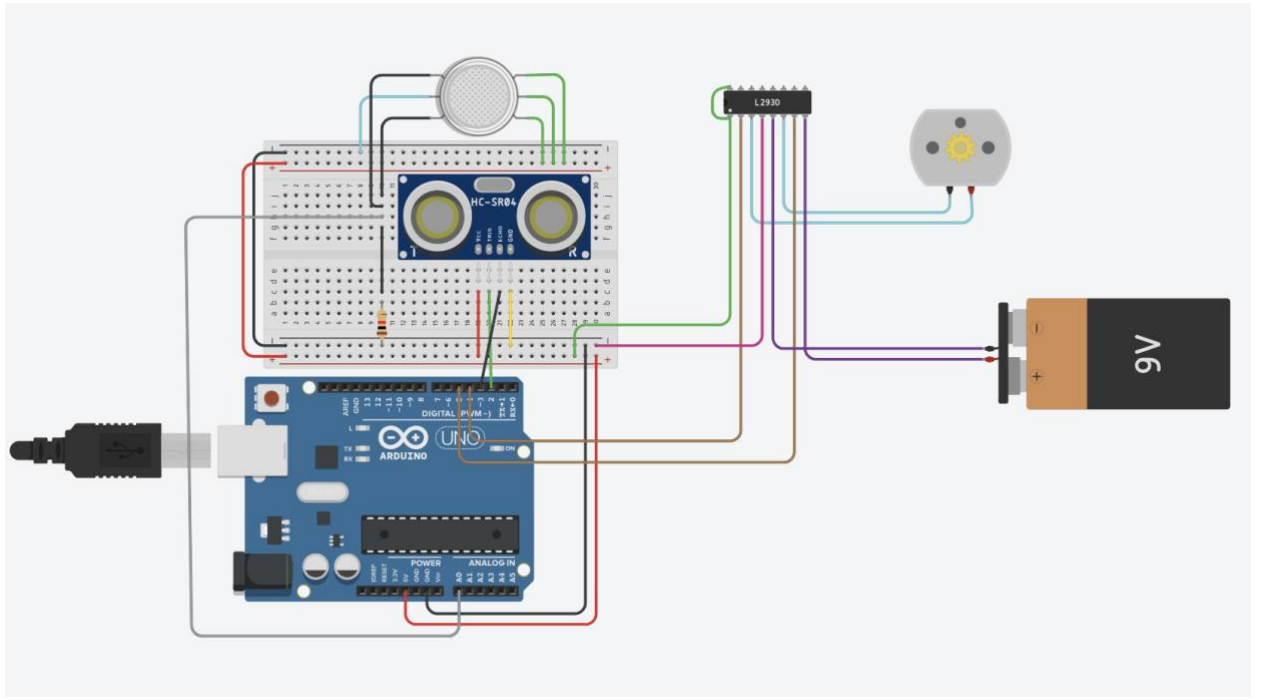




1. I/O Blocks: Use I/O blocks from the Simulink library to interface with sensors and actuators. These blocks can be found in the "Simulink" > "Sources" and "Simulink" > "Sinks" libraries. Examples include the "Analog Input" and "Analog Output" blocks for reading sensor data and controlling actuators, respectively.
2. Custom C/C++ Code: You can write custom C/C++ code to interface with sensors and actuators using the "S-Function" or "C Caller" blocks. These blocks allow you to integrate your own code into the Simulink model. You can find these blocks in the "Simulink" > "User-Defined Functions" library.
3. Device Driver Blocks: Some hardware support packages provide device driver blocks that are specifically designed to interface with certain sensors and actuators. These blocks can be found in the "Simulink Support Package for <Your Target Hardware>" library. For example, if you are using an Arduino board, you can find the corresponding blocks in the "Simulink Support Package for Arduino Hardware" library.
4. External Mode: Use the External mode feature to communicate with your target hardware during simulation. This allows you to send and receive data between the Simulink model and the target hardware in real-time. To enable External mode, go to "Simulation" > "Mode" > "External" in the Simulink toolbar.

ز) شکل نهایی پیاده سازی :





لینک :

[Q3](#)