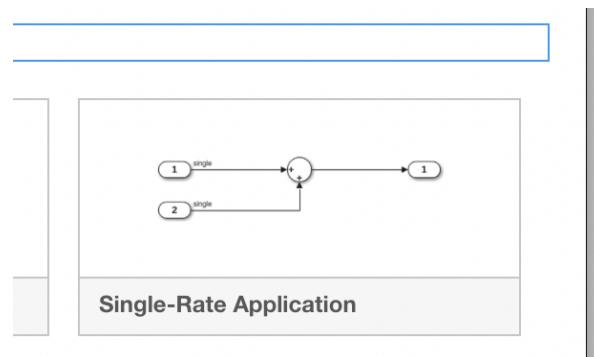


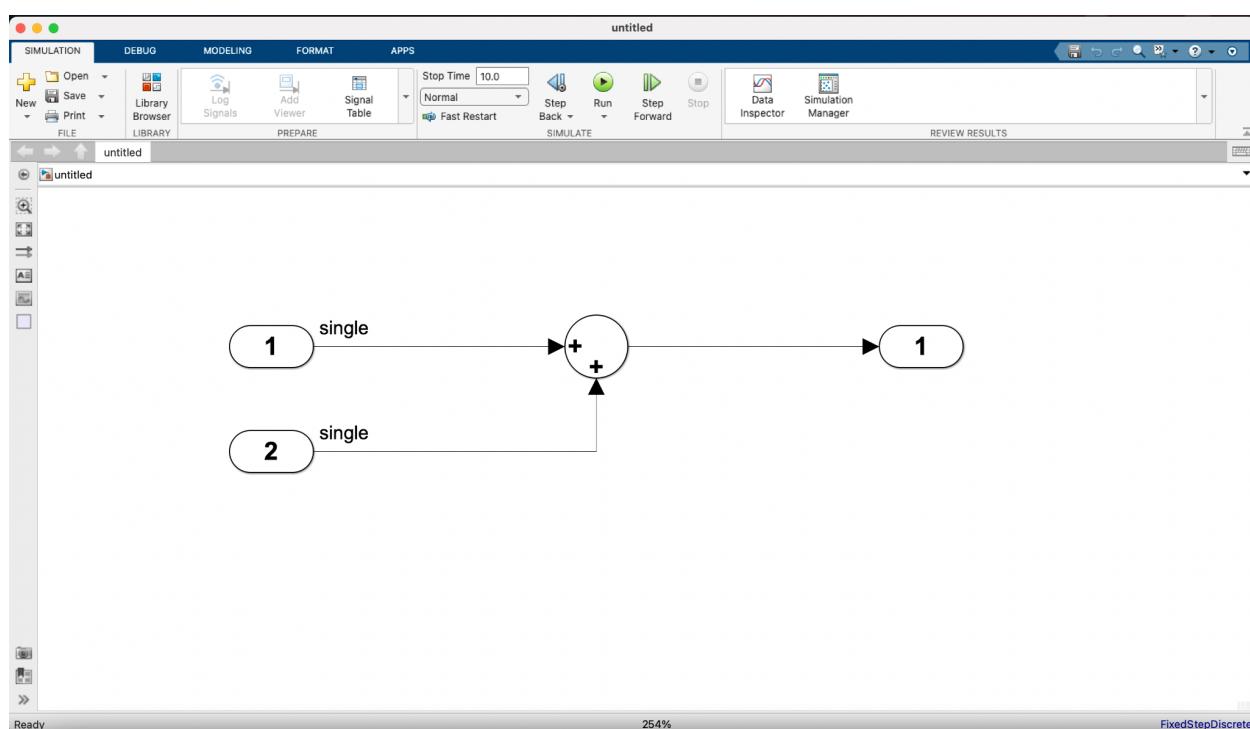
سوال چهارم:

(الف)

بعد از باز کردن برنامه متلب و ورود به سیمولینک کافی است تا از بخش Embedded Coder قالب مورد نظر صورت سوال رو ایجاد کنیم.



همانطور که در صورت سوال نیز ذکر شده است ما در این قالب یک جمع کننده ساده داریم که به صورت زیر میباشد:



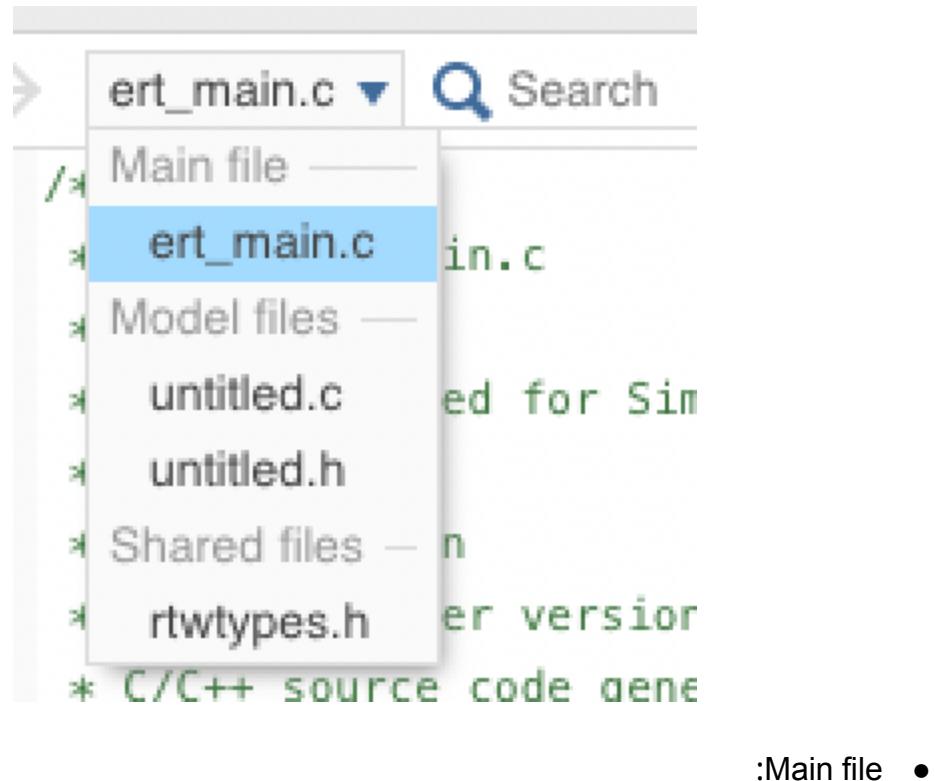
در ادامه برای کار راحت شدن کار با مدل و ... مقداری تنظیمات آن را تغییر میدهیم.

در ادامه قصد داریم تا بیلد بگیریم. که برای این کار در قسمت apps Embedded Coder را انتخاب کرده و بر روی Build کلیک میکنیم. سپس مدل به زبان مورد نظر (س) تولید خواهد شد و همچنین گزارش هم با توجه به تنظیمات آن به صورت دستی به صورت خودکار ایجاد و باز خواهد شد.

گزارش ایجاد شده به شکل زیر است:

The screenshot shows the 'Code Generation Report' dialog box for an unnamed model. The left sidebar lists report sections: Content (Summary, Subsystem Report, Code Interface Report, Traceability Report, Static Code Metrics Report, Code Replacements Report, Coder Assumptions), and Code (Main file: ert_main.c, Model files: untitled.c, untitled.h, Shared files: rtwtypes.h). The main area displays the 'Code Generation Report for 'untitled'' with three tabs: Model Information, Configuration settings at time of code generation, and Code Information. The Model Information tab shows details like Author: apple, Last Modified By: The MathWorks, Inc., Model Version: 1.0, and Tasking Mode: SingleTasking. The Configuration settings at time of code generation tab shows System Target File: ert.tlc, Hardware Device Type: Intel->x86-64 (Windows64), Simulink Coder Version: 9.8 (R2022b) 13-May-2022, Timestamp of Generated Source Code: Fri Apr 21 21:33:21 2023, Location of Generated Source Code: /Users/apple/MATLAB/Projects/hw2_q1/untitled_ert_rtw, Type of Build: Model, and Objectives Specified: Execution efficiency, Traceability. The Additional Information tab is empty. At the bottom right are OK and Help buttons.

بخش های اصلی کد ایجاد شده:



این بخش شامل کد ert main.c باشد که همان ابع اصلی و main مدل ما باشد که در آن ابتدا مدل init شود و در ادامه حلقه بینهایت برنامه برای اجرای تസکهای مدل اجرا می شود. همچنین ابع rt_oneStep نز در ان فال قرار دارد که در صورتی که همگام با زمان کلاک مدل فراخوانی شود، مدل یک بار اجرا شود و خروجی مدل تولید میشود. پس برای اجرای بن درنگ مدل در حلقه بن نهایت میتوان تابع rt_oneStep را فراخوانی کرد ا به طور نامحدود مدل اجرا شود.

```

Code
ert_main.c ▾ Q Search
27 * By the end of this function the generated code must have been
28 * interrupts and floating point context switches are target specific. This
29 * example code indicates where these should take place relative to executing
30 * the generated code step function. Overrun behavior should be tailored to
31 * your application needs. This example simply sets an error status in the
32 * real-time model and returns from rt_OneStep.
33 */
34 void rt_OneStep(void);
35 void rt_OneStep(void)
36 {
37     static boolean_T OverrunFlag = false;
38
39     /* Disable interrupts here */
40
41     /* Check for overrun */
42     if (OverrunFlag) {
43         return;
44     }
45
46     OverrunFlag = true;
47
48     /* Save FPU context here (if necessary) */
49     /* Re-enable timer or interrupt here */
50     /* Set model inputs here */
51
52     /* Step the model */
53     untitled_step();
54
55     /* Get model outputs here */
56
57     /* Indicate task complete */
58     OverrunFlag = false;
59

```

/Users/apple/MATLAB/Projects/hw2_q1/untilted_ert_rtw/ert_main.c

Ln 8 Col 54

:Model files •

شامل کد و هدر فال ورودی، خروجی تعریف شده مدل ما و نحوه محاسبه خروجی از ورودیها بیاشد.
منطق و عملیات مدل با گرفتن ورودی ها برای تولید خروجی پیاده سازی شده است.

:Shared files •

متغیرهای به صورت دیفالت مقداردهی شده اند که بتوان از آنها به صورت آماده در کدهای دیگر استفاده کرد.

```

Code
rtwtypes.h ▾ Q Search
29 * long: 32    long long: 64
30 * native word size: 64
31 * Byte ordering: LittleEndian
32 * Signed integer division rounds to: Zero
33 * Shift right on a signed integer as arithmetic shift: on
34 ****
35
36 **** Fixed width word size data types:
37 * int8_T, int16_T, int32_T      - signed 8, 16, or 32 bit integers   *
38 * uint8_T, uint16_T, uint32_T   - unsigned 8, 16, or 32 bit integers *
39 * real32_T, real64_T          - 32 and 64 bit floating point numbers*
40 ****
41 *
42 typedef signed char int8_T;
43 typedef unsigned char uint8_T;
44 typedef short int16_T;
45 typedef unsigned short uint16_T;
46 typedef int int32_T;
47 typedef unsigned int uint32_T;
48 typedef long long int64_T;
49 typedef unsigned long long uint64_T;
50 typedef float real32_T;
51 typedef double real64_T;
52
53 **** Generic type definitions: boolean_T, char_T, byte_T, int_T, uint_T,      *
54 * real_T, time_T, ulong_T, ulonglong_T.                                *
55 * ****
56 *
57 typedef double real_T;
58 typedef double time_T;
59 typedef unsigned char boolean_T;
60 typedef int int_T;

```

/Users/apple/MATLAB/Projects/hw2_q1/slprj/ert/_sharedutils/rtwtypes.h

Ln 38 Col 46

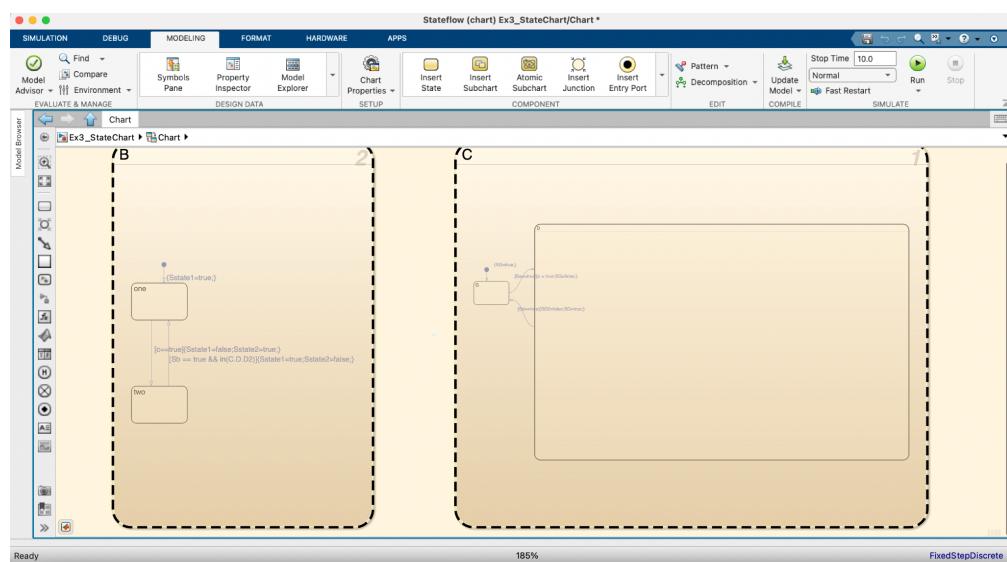
:Other files •

بخش Other files در صورت استفاده از مازول های آردوینو شامل کتابخانه های آن خواهد بود.

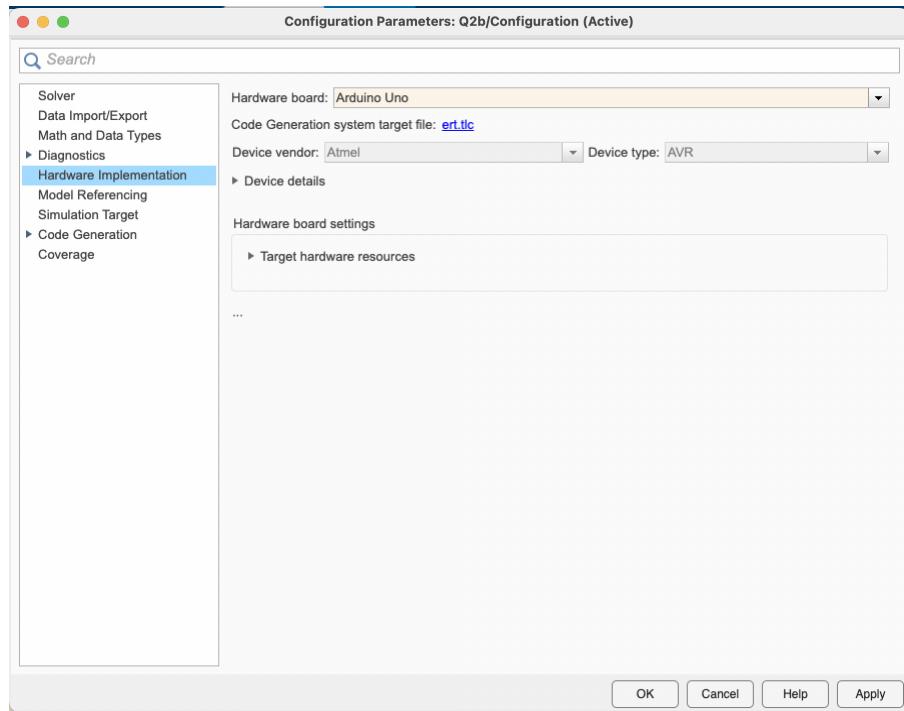
(ب)

طبق صورت سوال لازم است تا ابتدا مدل اولیه را مقداری تغییر دهیم.

برای این منظور باید برای هر وضعیت در واقع خروجی های جدیدی ایجاد کنیم. (ا بودن خروجی به معنای حضور در آن استیت است). از کلمه کلیدی Entry استفاده میکنیم و در هنگام ورود باید با استفاده از این کی ورد مقدار خروجی را برابر با ۱ و در هنگام خروج با کلمه کلیدی دیگری مانند Exit آن را صفر میکنیم. این کار را لازم است تا برای تمام استیت هایمان تکرار کنیم. حال همه چیز آماده است تا فایل های مورد نیاز برای آردوینو را ایجاد کنیم.



در ادامه نیاز است تا کد های مربوطه برای اجرا برای روی برد Arduino UNO تولید کنیم که در مرحله اول لازم است تا تنظیمات رو عوض کنیم مانند شکل زیر:



در ادامه بر روی دکمه بیلد کلیک میکنیم تا بر اساس نوع سخت افزار انتخاب شده برای ما کد مورد نظر را تولید کند که از این کد قرار است در قسمت (د) این تمرین استفاده کنیم. فایل های تولید شده همراه همین فایل تمرین ارسال شده است. به توضیح دقیق فایل های ایجاد شده در این قسمت پرداخته نمی شود به دلیل اینکه در قسمت الف همین سوال با جزئیات زیاد توضیح داده شده اند.

(ج)

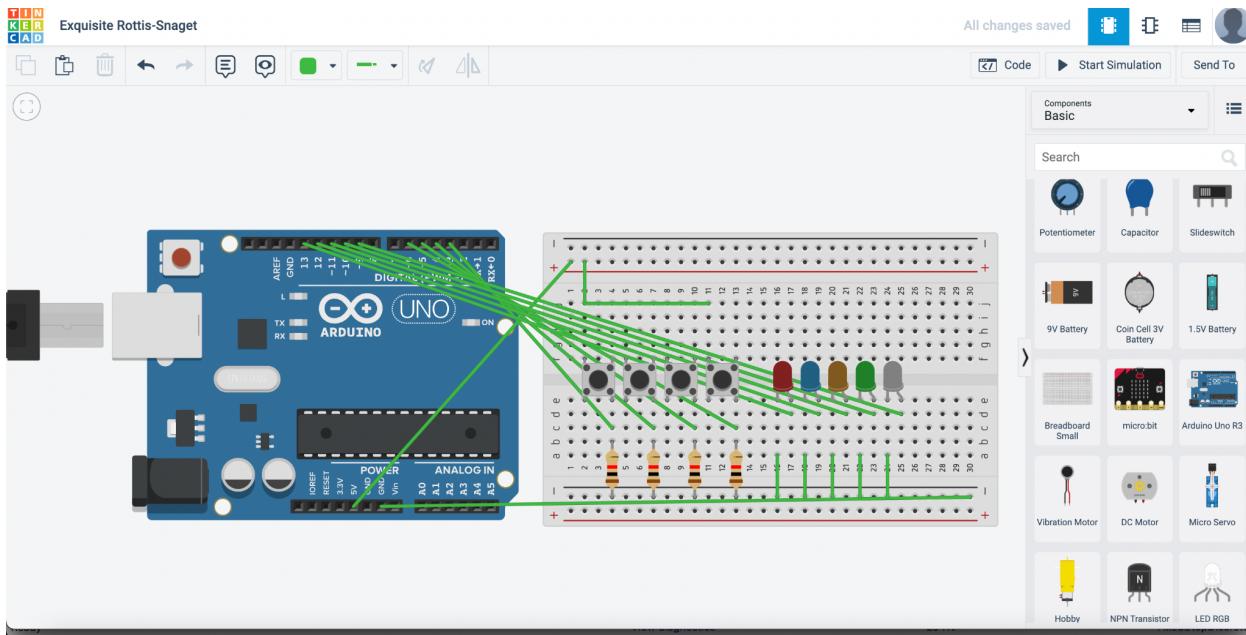
این قسمت از سوال را به طور کامل متوجه نشدم و بر اساس دانسته های قبلی و درک از سوال جوابی تهیه شده است.

برای نحوه دسترسی به ورودی ها و خروجی هایمان تعدادی Struct وجود دارد که این کار از طریق آن ها قابل انجام است. مثل داریم که: U_EX3_StateChart input آن فراخوانی میشود که به این صورت ما به یک ورودی دسترسی پیدا کرده ایم. از طرف دیگر هم مثل داریم:

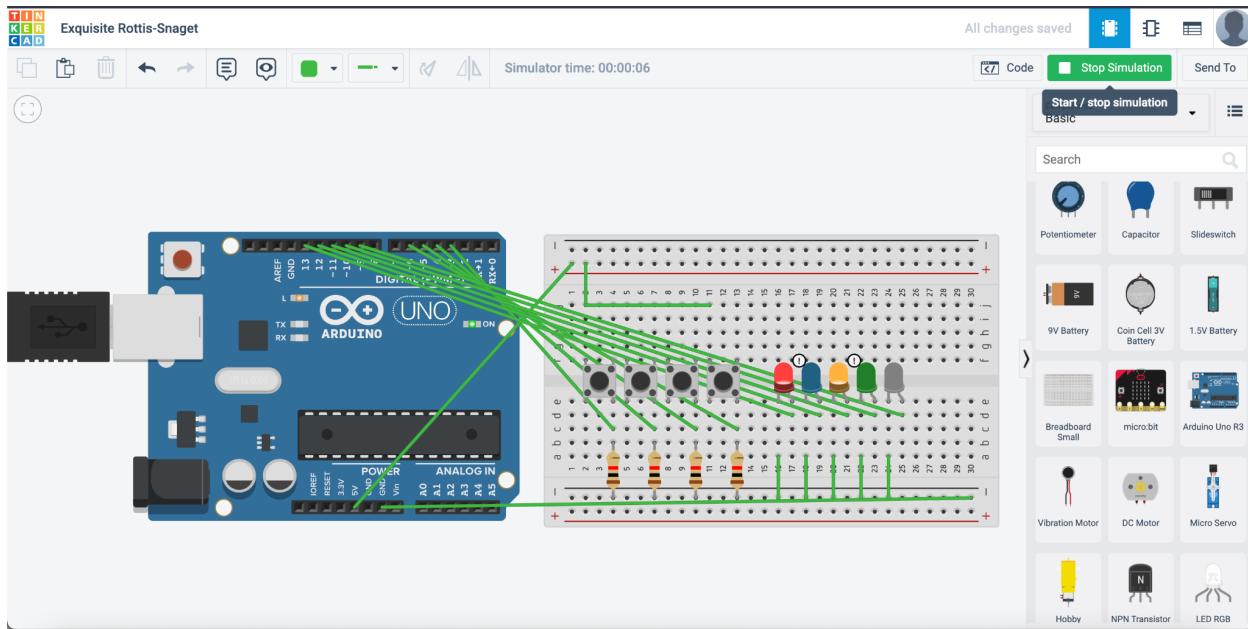
EX3_StateChart_Y که میتوانیم بر روی آن output را فراخوانی کنیم و به این صورت به خروجی ها نیز دسترسی خواهیم داشت.

(د)

شکل زیر مدار را در حالت خاموش و اولیه نشان میدهد.



اما شکل زیر بعد از زدن دکمه شبیه سازی می باشد:



توجه شود که فایل BRD خروجی نیز همراه پروژه ارسال شده است.