

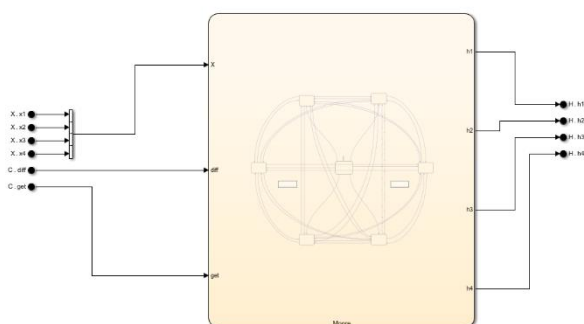
شماتیک سیستم کنترلگر مورد نظر در قسمت زیر آورده شده است :

### توصیف ورودی‌های سیستم کنترلی :

سیستم به عنوان ورودی، بردارهای  $u, C, X=[x_i]$  را دریافت می‌کند که به ترتیب دمای اتاق‌ها، ثابت‌های تعیین شده برای مساله، و دمای محیط خارج در لحظه‌ی فعلی می‌باشد. در توصیف سیستم فرض بر این گذاشته شده که دمای محیط خارج نیز بتواند تغییر کند. ( هر چند در سوال صورت پروژه این مقدار ثابت و برابر ۶ بود )

### توصیف خروجی‌های سیستم کنترلی :

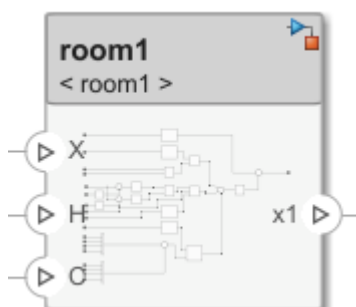
سیستم بردار  $H=[h_i]$  را به عنوان خروجی به اتاق‌ها برمی‌گرداند تا بتوانند حالت خود را ( که با دمای آن‌ها مشخص می‌گردد ) به کمک آن به روز رسانی نمایند. طبق خواسته مساله، اتاق‌ها باید دمایی بین ۱۵ تا ۲۰ درجه داشته باشند و همچنین مقدارهای  $H$  عدد صحیح ۰ یا ۱ است.



شماتیک سیستم‌های تعیین‌کننده‌ی دمای خانه‌ها در قسمت زیر آورده شده است :

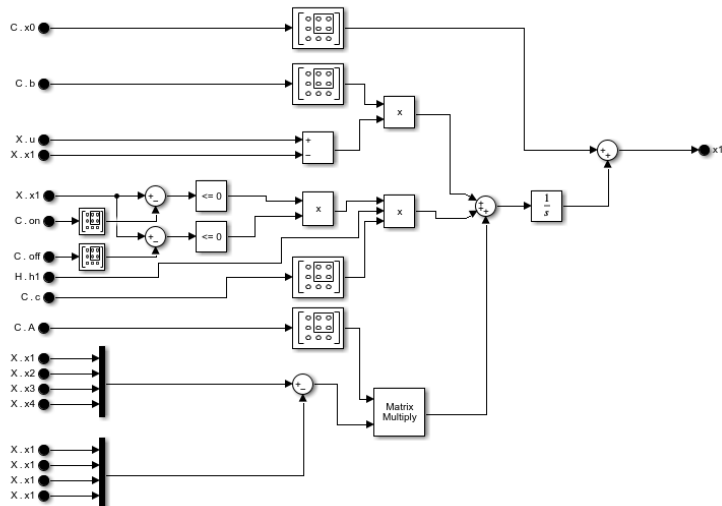
### توصیف ورودی‌های سیستم اتاق :

سیستم شماره‌ی  $i$ ، به عنوان ورودی، بردارهای  $X, H, C$  را دریافت می‌کند که به ترتیب بیانگر ثابت‌های مساله، وضعیت روشن و یا خاموش بودن بخاری‌ها و همچنین دمای اتاق‌ها در لحظه‌ی فعلی می‌باشد. این مقدارها برای به‌روزرسانی حالت سیستم ( که همان اتاق است ) در زمان‌ها پیوسته، طبق معادله‌ی داده شده به کار می‌روند، همچنین شماتیک زیر، معادله‌ی گفته شده را پیاده سازی کرده است.



### توصیف خروجی‌های سیستم اتاق :

این سیستم تنها پورت  $x_i$  که در هر لحظه دمای سیستم را نشان میدهد را به عنوان خروجی باز می‌گرداند. این چهار پورت خروجی در یک ماژول join به یکدیگر پیوسته و باس حالت اتاق‌ها را که شامل دمای خارج و دمای ۴ اتاق است را خواهند ساخت. شماتیک ماژول join در قسمت زیر آمده است.



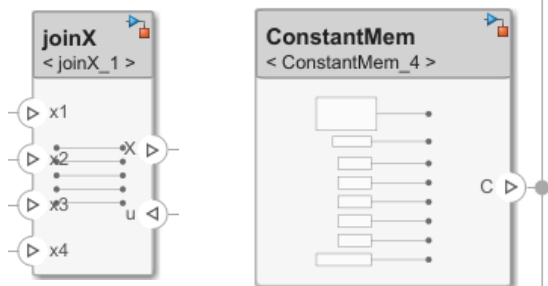
شماتیک سیستم‌های کمکی افزوده شده به دنیا

### ماژول join :

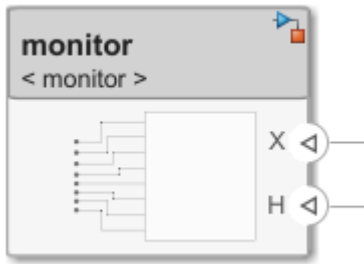
همانطور که در شکل مقابل دیده می‌شود، این ماژول خروجی تک‌تک سیستم‌های اتاق را به همراه دمای خارج گرفته و بردار وضعیت کلی سیستم را ( $X$ ) برمی‌گرداند.

### ماژول ConstantMem :

مقادیر ثابت‌های مساله، به صورت ماتریس‌های طبق گفته‌ی صورت پروژه، در این ماژول قرار می‌گیرد و به کمک interface تعیین شده برای آن، ثابت‌ها را به سیستم‌های نیازمند آن‌ها می‌رساند.



## ماژول Monitor :



شامل scope برای رسم نمودار حالت سیستم می‌باشد. در صورت نیاز می‌توان قطعات جدید را به آن اضافه نمود. به عنوان ورودی، پورت‌های X, H را دریافت می‌کند که بیانگر خروجی کنترلر و حالت کلی سیستم در هر لحظه می‌باشد.

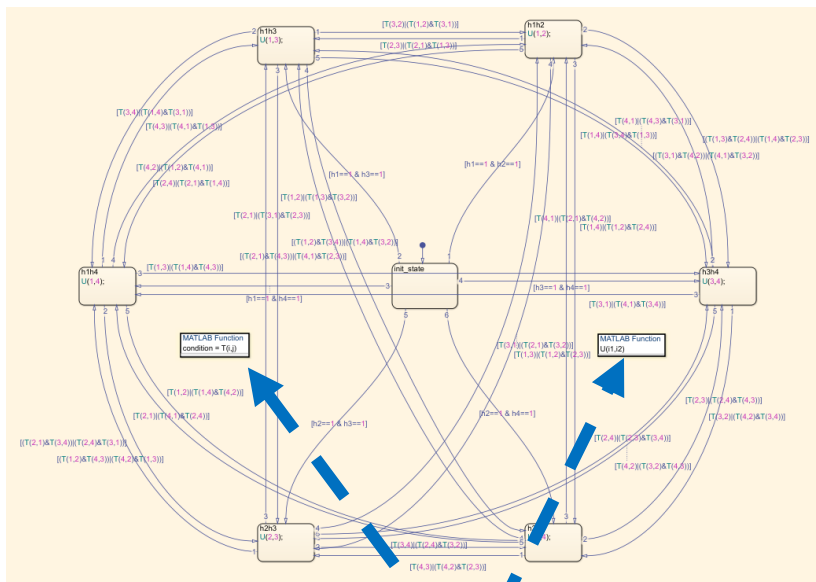
## Interface های تعیین شده برای مدار مساله :

C		
A	double	[4 4]
b	double	[4 1]
c	double	[4 1]
x0	double	[4 1]
off	double	[4 1]
on	double	[4 1]
get	double	[4 1]
diff	double	[4 1]
controller_in		
u	double	1
x1	double	1
x2	double	1
x3	double	1
x4	double	1
controller_out		
h1	int8	1
h2	int8	1
h3	int8	1
h4	int8	1

جدول مقابل، interface مربوط به باس های به کاررفته در مدار مساله را نشان می‌دهد. باس C برای انتقال ثابت‌های مساله به‌کار رفته است و شامل بردار های ثابت مساله که در صورت پروژه گفته شده می‌باشد. باس controller\_in در واقع همان باس مربوط به H یا حالت کلی سیستم می‌باشد که شامل دمای اتاق ها و محیط خارج می‌باشد. باس controller\_out شامل بردار H می‌باشد که در هر لحظه تعیین می‌کند که بخاری های مربوط به هر اتاق روشن باشند یا نه.

## جزئیات پیاده‌سازی کنترل‌کننده و بررسی ماشین حالت آن :

همان‌طور که در شکل مقابل دیده می‌شود، ماشین حالت مربوط به کنترلر، دارای ۷ حالت می‌باشد که متشکل از یک حالت برای initiate کردن وضعیت اولیه سیستم با توجه به محتوای حافظه و فرضیات در نظر گرفته شده برای مساله است ( که در این جا اتاق هایی که در ابتدا در آن ها بخاری وجود دارند می‌باشد) و شش حالت برای وجود بخاری در اتاق‌های مختلف می‌باشد  $\binom{4}{2} = 6$  که در نام گذاری آن ها مشاهده می‌گردد. ( نام‌گذاری h1h2 به معنی وجود بخاری در اتاق‌های ۱, ۲ می‌باشد). همان‌طور که دیده می‌شود، اگر شرایط ذکر شده مساله برقرار باشد، کنترلر از هر حالت می‌تواند به حالت های دیگر ( به جز حالت اولیه) برود. بنابراین ماشین حالت دارای یک گرافت کامل مرتبه ۶ خواهد بود. همچنین برای خلاصه کردن هرچه بهتر و افزایش خوانایی ماشین حالت، از دوتابع متلب جهت بیان کردن شرایط و همچنین مقداردهی خروجی استفاده نموده ایم.



```
function U(i1,i2)
[h1,h2,h3,h4]=deal(int8(0),int8(0),int8(0),int8(0));
if((i1==1)||(i2==1))
h1=int8(1);
end
if((i1==2)||(i2==2))
h2=int8(1);
end
if((i1==3)||(i2==3))
h3=int8(1);
end
if((i1==4)||(i2==4))
h4=int8(1);
end
end
```

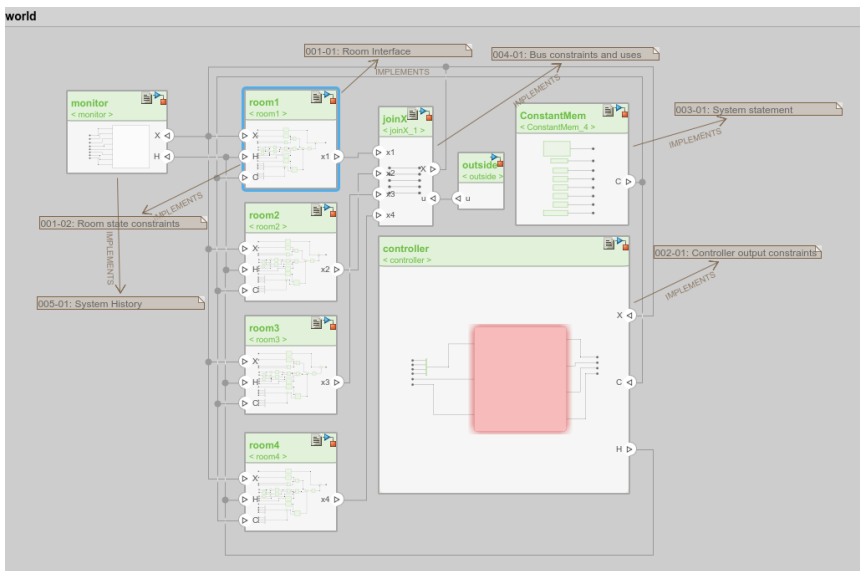
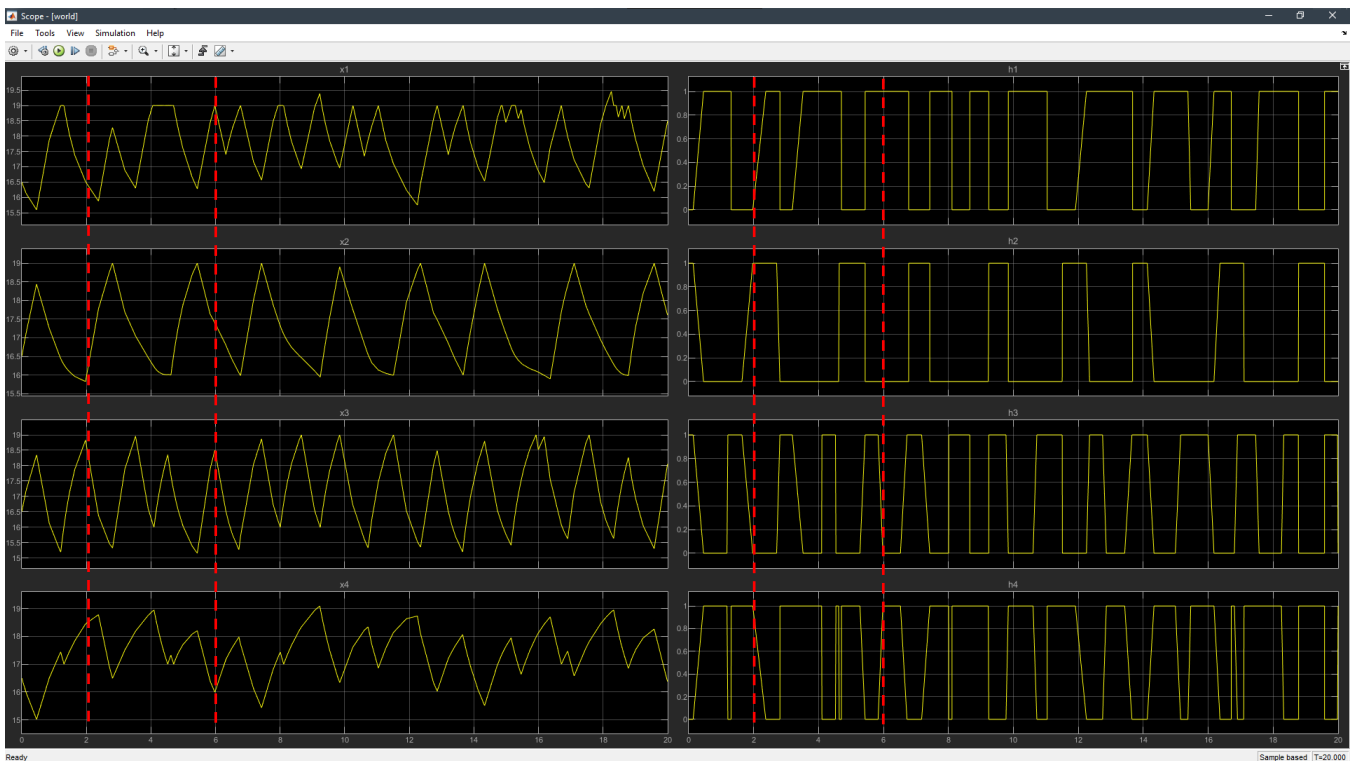
```
function condition = T(i,j)
condition = (X(j)<=get(j))&((X(i)-X(j))>=diff(j));
end
```

## توصیف نیازمندی‌های مساله :

جدول زیر، وضعیت نیازمندی‌های پیشنهادشده و پیاده‌سازی آن‌ها را نشان می‌دهد. نیازمندی‌ها به شرح زیر کدگذاری شده‌اند : ۱- نیازمندی‌های مربوط به اتاق‌ها با کد ۰۰۱ شروع می‌شوند. ۲- نیازمندی‌های مربوط به سیستم کنترل‌کننده با کد ۰۰۲ شروع می‌شوند. ۳- نیازمندی‌های مربوط به قطعات جانبی با کد ۰۰۳ شروع می‌شوند. ۴- نیازمندی‌های مربوط به ارتباط اجزا و باس‌ها با کد ۰۰۴ شروع می‌شوند. ۵- نیازمندی‌های مربوط به تجربه‌ی کاربری و ارتباط با کاربر با کد ۰۰۵ شروع می‌شوند.

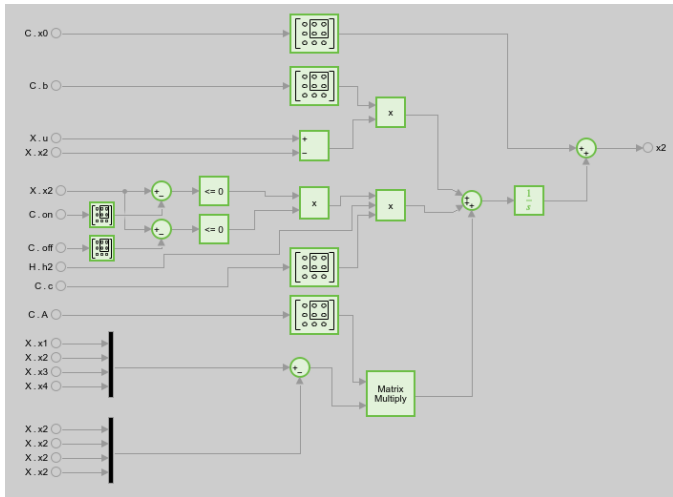
Index	ID	Summary	Implemented
initialS...			
1	001-01	Room Interface	
2	001-02	Room state constraints	
3	002-01	Controller output constraints	
4	002-02	Controller initialization	
5	003-01	System statement	
6	004-01	Bus constraints and uses	
7	005-01	System History	

نتیجه‌ی شبیه‌سازی پس از اجرای مدل برای مدت ۲۰ ثانیه با فرضیات موجود در صورت پروژه به صورت زیر می‌باشد :



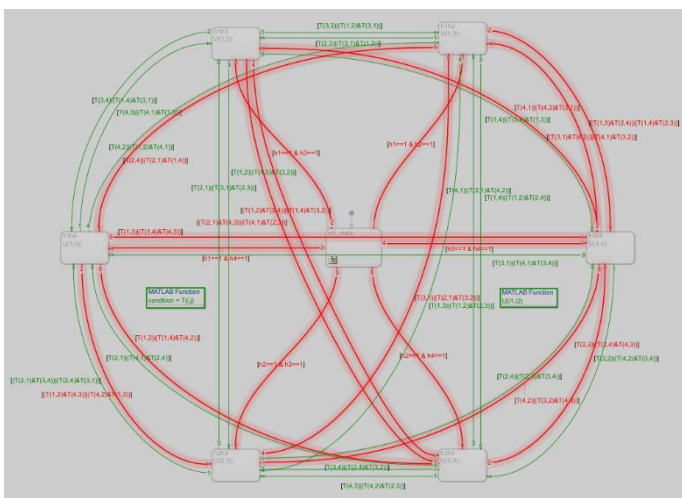
در رابطه با تصویر بالا نکات زیر مهم می‌باشد :

- در نمودار های سمت راست که مربوط به بردار  $H$  که همان وضعیت بخاری‌ها در هر لحظه می‌باشد، در هر زمان، تنها دو نمودار ۱ و بقیه ۰ هستند.
- در نمودار های سمت چپ که دمای اتاق‌ها بر حسب زمان را نشان می‌دهند، دما در هر لحظه عددی بین ۱۵ تا ۲۰ است که نشان دهنده‌ی



صحت کارکرد کنترل‌کننده‌ی توصیف شده برای سیستم می‌باشد. ( پاسخ سوال اول صورت پروژه )

۳- معیار جعبه‌ی سیاه بر روی نیازمندی‌های سیستم تست شد و نتیجه در بالا گزارش شد. برای معیار جعبه‌ی سفید، بعد از بررسی سوال داده شده در صورت پروژه، گزارش *coverage* توسط متلب تولید گردید، همانطور که در شکل زیر دیده میشود، شماری از انتقال ها در سوال داده شده پوشش داده شده اند. گزارش پوشش در پروژه‌ی ارسالی قرار دارد و فایل مربوط به پوشش کنترل‌کننده در فایل مربوطه قرار داده شده در پروژه قرار دارد.



۴- تولید کد تنها برای قسمت کنترلر گفته شده انجام گرفته است، چرا که بقیه‌ی قسمت‌ها به صورت زمان پیوسته مدل شده اند و امکان پیاده‌سازی آن ها و شبیه‌سازی‌شان با حل‌کننده‌ی زمان گسسته وجود نداشت. بعد از استخراج کد بخش کنترل‌کننده و تجمیع آن در یک فایل C، نوبت به بخش پایانی که همان نوشتن یونیت تست برای اطمینان حاصل نمودن از صحت کارکرد کنترلر می‌رسد. کد ساخته شده در پوشه‌ی کد مربوط به کنترلر قرار دارد.

۵- دو نمونه از یونیت تست‌های نوشته شده برای کنترل کننده در شکل زیر قرار داده شده است. مقادیر ورودی و خروجی با توجه به قید های مساله تعیین شده اند و تست ها با موفقیت پاس شده اند.

```
E:\EmbedProject\controller_unit_test\ControllerTest> pio test
Verbose mode can be enabled via '-v, --verbose' option
Collected 1 items

Processing test_desktop in native environment
Building...
test\test_desktop\test_controller.c:31:test_step_1 [PASSED]
test\test_desktop\test_controller.c:36:test_step_2 [PASSED]

-----
2 Tests 0 Failures 0 Ignored
Ok
===== [PASSED] Took 1.31 seconds =====

Test      Environment  Status  Duration
-----
test_desktop native    PASSED  00:00:01.308
===== 1 succeeded in 00:00:01.308 =====
```

۶- همانطور که در قسمت بالا نیز ذکر شد، امکان تولید کد برای کل ماژول word به دلیل استفاده از ماژول های زمان پیوسته و زمان گسسته وجود نداشت، لذا برای تست کردن کنترلر که ماژول زمان گسسته است و قابل پیاده‌سازی بر روی برد آردوینو می‌باشد، از امکان تولید کد متلب

استفاده نمودیم و کد را تولید کرده در افزونه‌ی platformIO تست نمودیم.