



پروژه مبانی سیستم های نهفته و بی درنگ



محمد مهدی محقولى ۹۶۲۴۳۰۶۱

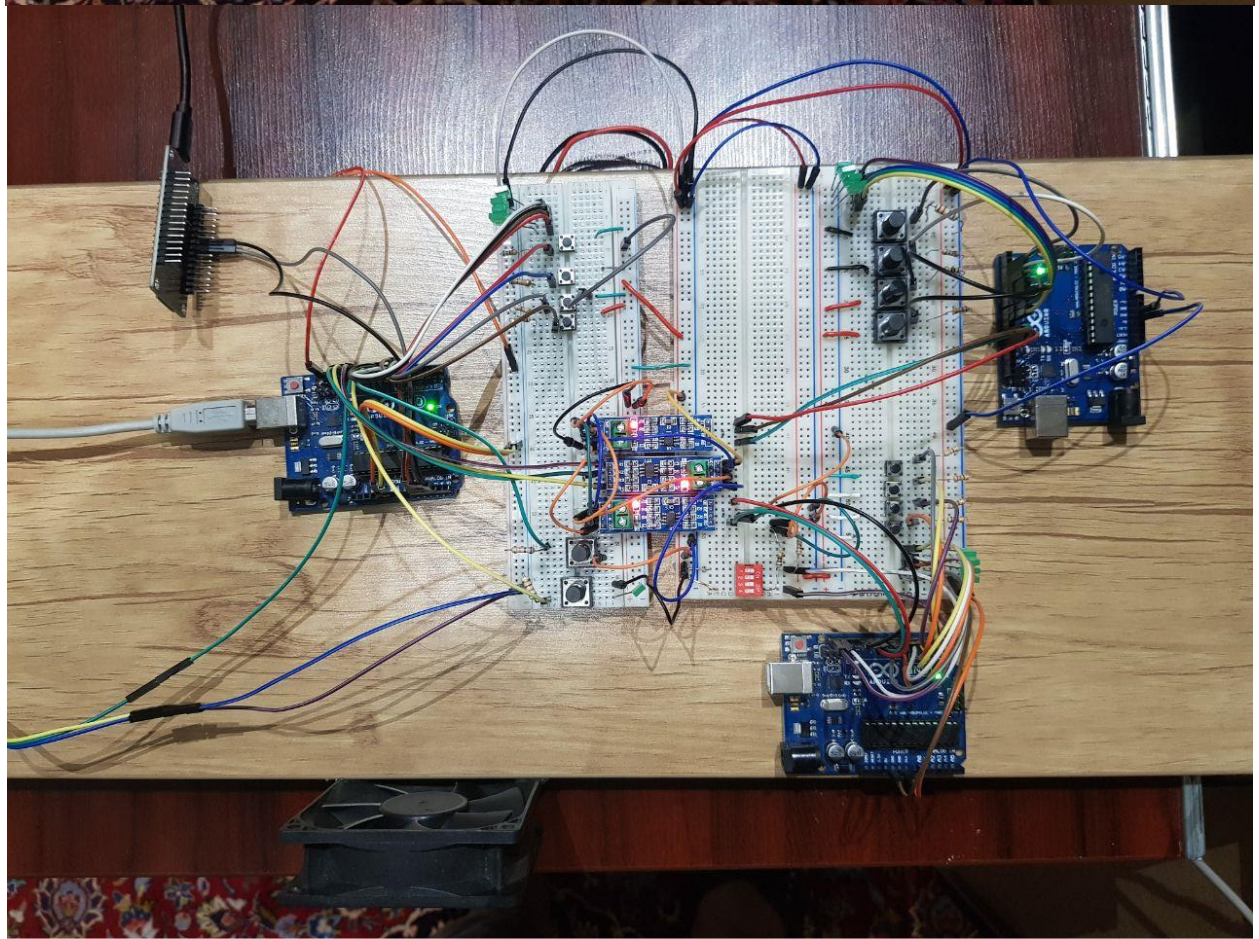
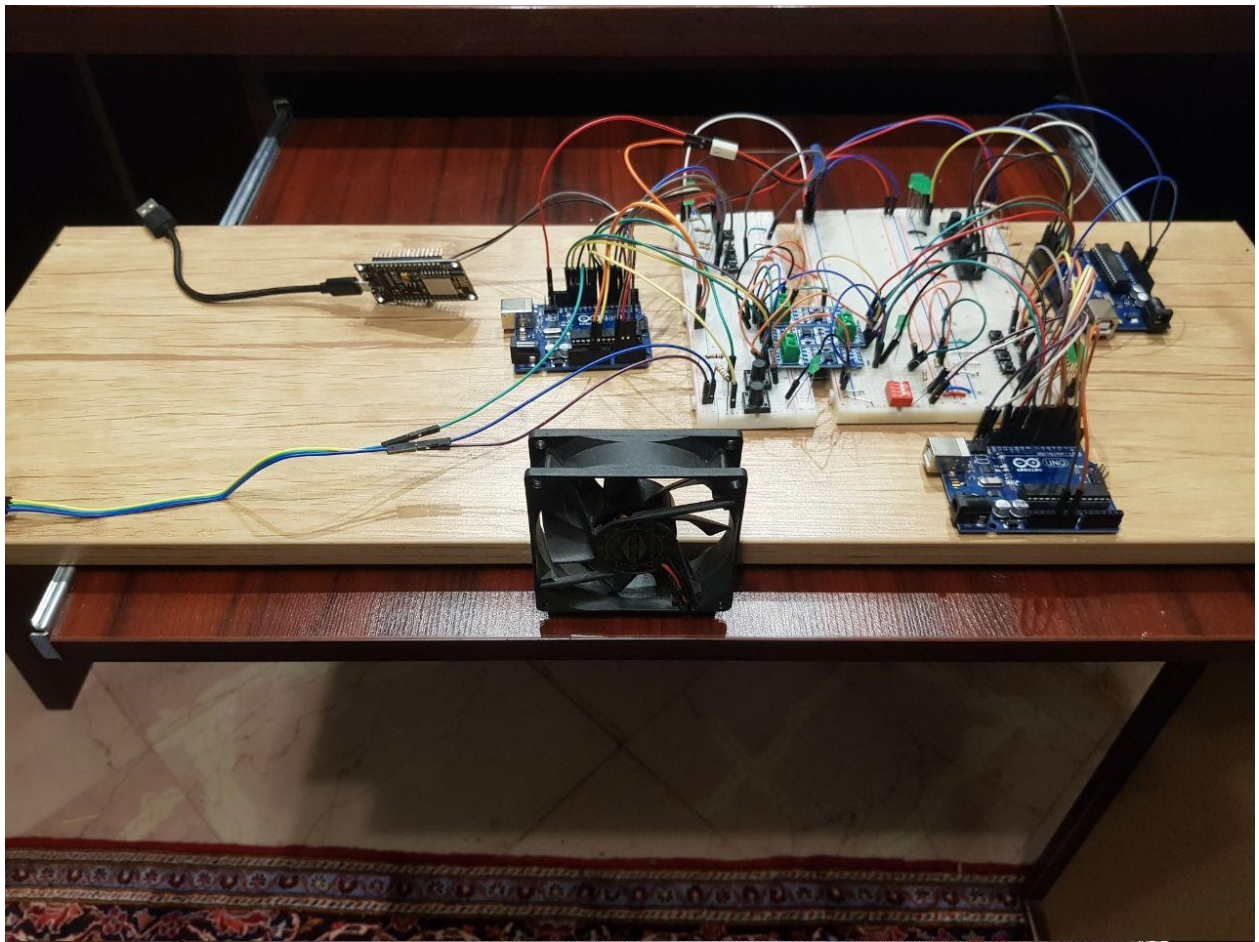
علیرضا علیدوستى ۹۶۲۴۳۰۴۷

مقدمه: پروژه پیاده سازی شدی، مطابق صورت پروژه یک سیستم مدیریت روشنایی و دمای محیط است که از چند سویچ محلی و یک

سویچ مرکزی تشکیل شده. هر کدام از سویچ های محلی می توانند یک سویچ ۴ پل به همراه ۴ led باشند و یا یک گره ترموستات که به کمک سنسور دما، دما را اندازه گیری می کند و یک دمای مطلوب تعریف شده دارد. در صورت سوال ذکر شده که گره ترموستات به یک گرم کنند متصل باشد و اگر دما از دمای مطلوب کم تر شد، گرم کننده فعال شود. ولی ما به دلیل این که مدار را روی برد پیاده کردیم و امکان اتصال گرم کننده نداشتیم تغییری در صورت سوال ایجاد کردیم و فرض کردیم که اگر دما از مقدار مطلوب بالا تر رفت، خنک کننده (پنکه) فعال شود. گره master می تواند با انتخاب هر کدام از slave ها وضعیت آنها را به صورت local مشاهده کند و در صورت لزوم در آن تغییر ایجاد کند. اگر گره slave یک سویچ ۴ پل باشد وضعیت آن شامل وضعیت روشن یا خاموش بودن led ها می شود که به طور متناظر روی led های گره master نیز وضعیت نمایش داده می شود و می توان هم به کمک کلید های محلی هم به کمک کلید های master آنها را روشن یا خاموش کرد. اگر گره slave، گره ترموستات باشد وضعیت آن شامل دمای اندازه گیری شده و دمای مطلوب می باشد که روی یک نمایشگر در slave نمایش داده می شود و master نیز با انتخاب گره ترموستات وضعیت دما و دمای مطلوب slave را روی نمایشگر خود نمایش می دهد. slave ترموستات می تواند با کمک ۲ دکمه ای که دارد با هر با فشار دادن آنها دمای مطلوب را یک درجه کاهش یا افزایش دهد. master نیز در هنگام اتصال با دو دکمه می تواند این کار را روی slave انجام دهد. (نکته: در اینجا ما به دلیل محدودیت تعداد پایه روی arduino uno به جای استفاده از نمایشگر در master و گره ترموستات، هر کدام از آنها را به یک ماژول wifi متصل کردیم که به کمک آن وضعیتی که باید رو نمایشگر ها نمایش داده شود را در یک وب سرور ایجاد شده توسط ماژول نمایش می دهیم. اتصال باس بین نود ها به کمک ماژول های rs485 ایجاد شده.

همچنین به دلیل محدودیت تعداد برد های آردوینو، برای پیاده سازی یک master و سه slave (دو سویچ روشنایی و یک گره ترموستات) دو شماره slave به یک برد آردوینو تخصیص داده شده که یکی مربوط به سویچ روشنایی و دیگری مربوط به ترموستات است.

ما از دو ورودی برای انتخاب سویچ slave استفاده کردیم در نتیجه در نهایت می توانیم ۳ slave را متصل کنیم (شماره ۰ رزرو است) اما اگر پین های ورودی را بیشتر کنیم و با اضافه کردن ماژول های بیشتر max485 می توان تعداد slave ها را به تعداد دلخواه اضافه کرد



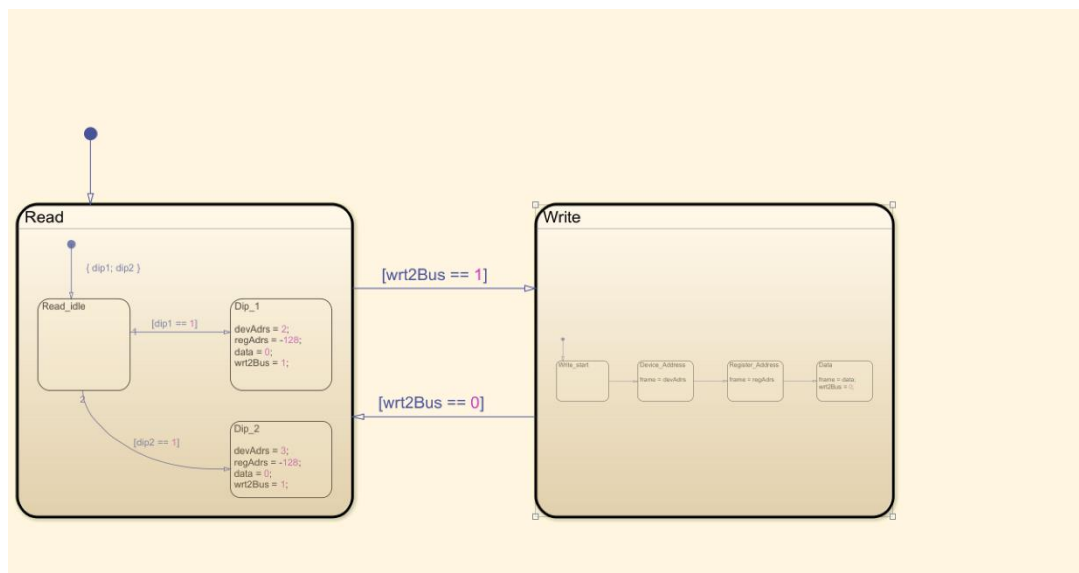
نیازمندی ها

priority	Requirement	ID	parent
۱	یک گره master داریم و تعداد slave ها باید تا ۳ تا قابل توسعه باشد یعنی تعداد slave ها میتواند از ۱ تا ۴ تا متغیر باشد	۱	null
۱	Master در هر لحظه فقط با یک یا هیچکدام از slave ها ارتباط دارد	۲	۱
۲	با انتخاب شماره slave از طریق dip switch ، گره master باید با slave مورد نظر sync شود	۳	۱
۱	Slave میتواند یک سویچ روشنایی چهار پل یا یک گره ترموستات باشد	۴	۱
۱	هر گره سویچ روشنایی دارای چهار پل و چهار led متناظر است	۵	۴
۱	در هر سویچ روشنایی هر پل با یک led متناظر است و led با تغییر وضعیت پل روشن یا خاموش میشود	۶	۵
۱	گره master دارای چهار پل و چهار led متناظر است	۷	۱
۱	در master هر پل با یک led متناظر است و led با تغییر وضعیت پل روشن یا خاموش میشود	۸	۷
۱	Sync شدن master و slave سویچ روشنایی به این معنی است که وضعیت led ها در slave به master اطلاع داده شود و master آن را همانگونه روی led های خود نمایش دهد	۹	۳
۳	Sync شدن master با slave گره ترموستات به این معنی است که درجه دما و دمای مطلوب روی slave انتخاب شده به master اطلاع داده شود.	۱۰	۳
۱	ارتباط بین master با slave ها مطابق پروتکل اعلام شده میباشد	۱۱	۳
۱	ارتباط master با slave ها بر بستر bus عمومی به صورت halfduplex ایجاد شود یعنی در هر لحظه فقط یک نفر بتواند روی bus بنویسد و همه میتوانند از آن بخوانند	۱۲	۱۱
۲	با تغییر وضعیت led ها بوسیله push button مربوط به هر کدام به صورت local در slave سویچ روشنایی انتخاب شده، منجر به تغییر وضعیت led ها به طور متناظر در led های master شود	۱۳	۹
۲	با تغییر وضعیت led ها بوسیله push button مربوط به هر کدام در master، منجر به تغییر وضعیت led ها به طور متناظر در led های slave سویچ روشنایی انتخاب شده شود	۱۴	۹
۳	هر گره ترموستات یک سنسور دماسنج دارد که دمای محیط را اندازه گیری میکند و یک مقدار دمای مطلوب برای آن تعریف میشود	۱۵	۴
۳	به هر گره ترموستات دو push button متصل است که با هر بار کلیک کردن روی آن ها میتوان دمای مطلوب را یک درجه کاهش یا یک درجه افزایش داد	۱۶	۱۵
۳	به هر گره ترموستات یک خنک کننده متصل است	۱۷	۴
۴	گره master و گره ترموستات به module .wifi متصل هستند	۱۸	۴
۴	اطلاعات دمای اندازه گیری شده و دمای مطلوب روی slave بر روی webserver ایجاد شده توسط wifi module در هر ۱۰،۳±۱ میلی ثانیه (با توجه به baud rate و میزان تاخیر ۱۰ ms لحاظ شده در هر بار ارسال سه فریم در هر بار ارسال با توجه به پروتکل ارتباطی) قابل مشاهده است	۱۹	۱۰
۳	اگر در گره ترموستات دمای اندازه گیری شده بیشتر از دمای مطلوب بود fan روشن شود	۲۰	۱۷
۲	در این پروژه کلید ها با مقاومت pull down پیاده سازی میشود	۲۱	۱
۱	با انتخاب یک slave توسط master وضعیت slave، با حداکثر ۲۰ میلی ثانیه تاخیر به master منتقل شود.	۲۲	۳
۱	با تغییر وضعیت روی master، تغییرات با حد اکثر ۲۰ میلی ثانیه تاخیر روی slave اعمال شود	۲۳	۳

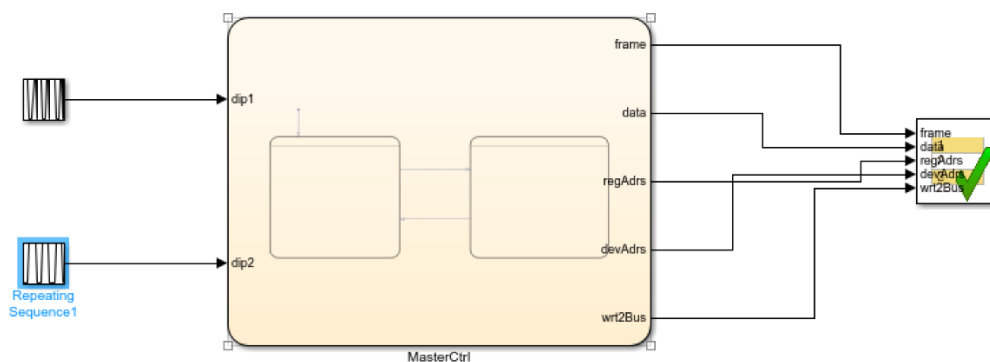
تست مدل:

برای انجام تست های لازم از ماژول test sequence استفاده شد که با توجه به ورودی هایی که بوسیله ماژول repeating sequence به فراهم میشود نتیجه نهایی و مورد انتظار بوسیله test case ها مورد ارزیابی قرار میگیرد

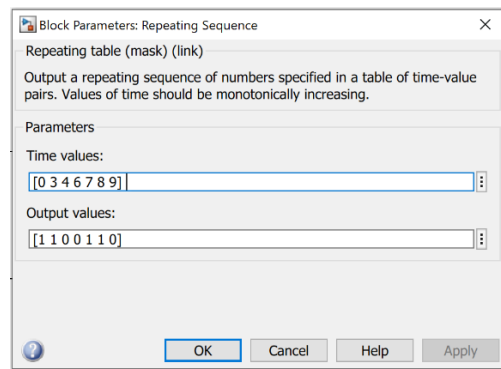
Master:



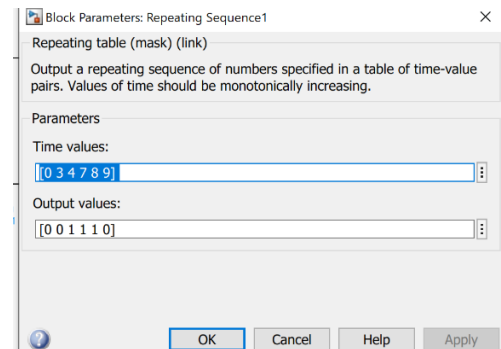
در کنترلر مربوط به master و روی های ما شامل سویچ های انتخاب بین دو slave است یعنی Dip1 و Dip2 و خروجی های ما شامل سیگنال های مورد نیاز برای ارتباط بوسیله bus ، Frame ، Data ، Regard ، Devadr ، Wrt2bus



ورودی ها و تست کیس های ما شامل انتخاب slave یک یا انتخاب slave دو یا هر دو به صورت همزمان است (که در این حالت به دلیل اولویت در action ها slave با اولویت بالاتر انتخاب میشود) یا هیچ کدام میشود

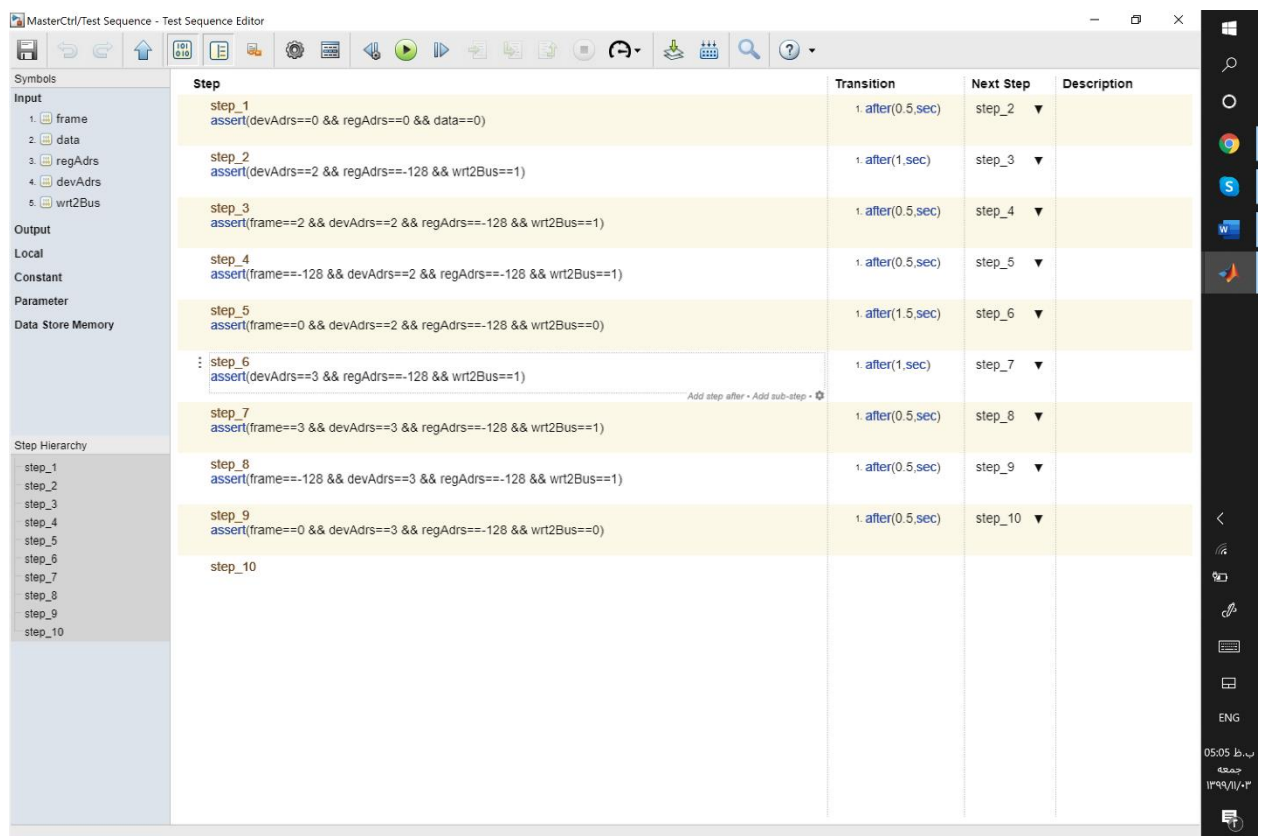


:Dip1



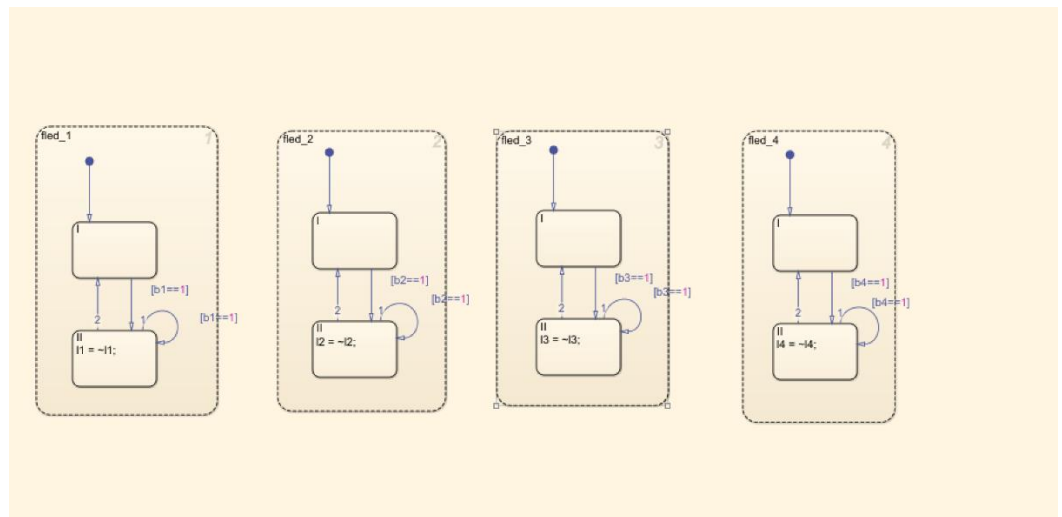
:Dip2

خروجی ها نیز در زمان های مشخص مورد بررسی قرار گرفتند و چون 5 cycle طول میکشد تا تمامی خروجی های مورد انتظار slave انتخاب شده تولید شود بنابر این برای هر کدام پنج assertion نوشته شده



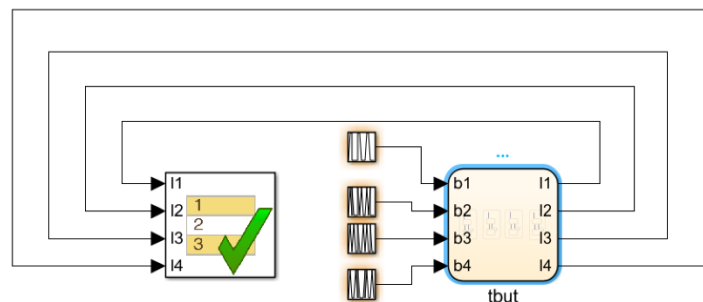
کنترل کننده led ها:

در این کنترل کننده با توجه به event که باعث تغییر خروجی و روشنایی led ها میشود در واقع این event همانطور که در صورت پروژه نیز گفته شده ممکن است از منابع متفاوتی ایجاد شود ولی در نهایت نتیجه و نهایی آن خاموش شدن led ها است

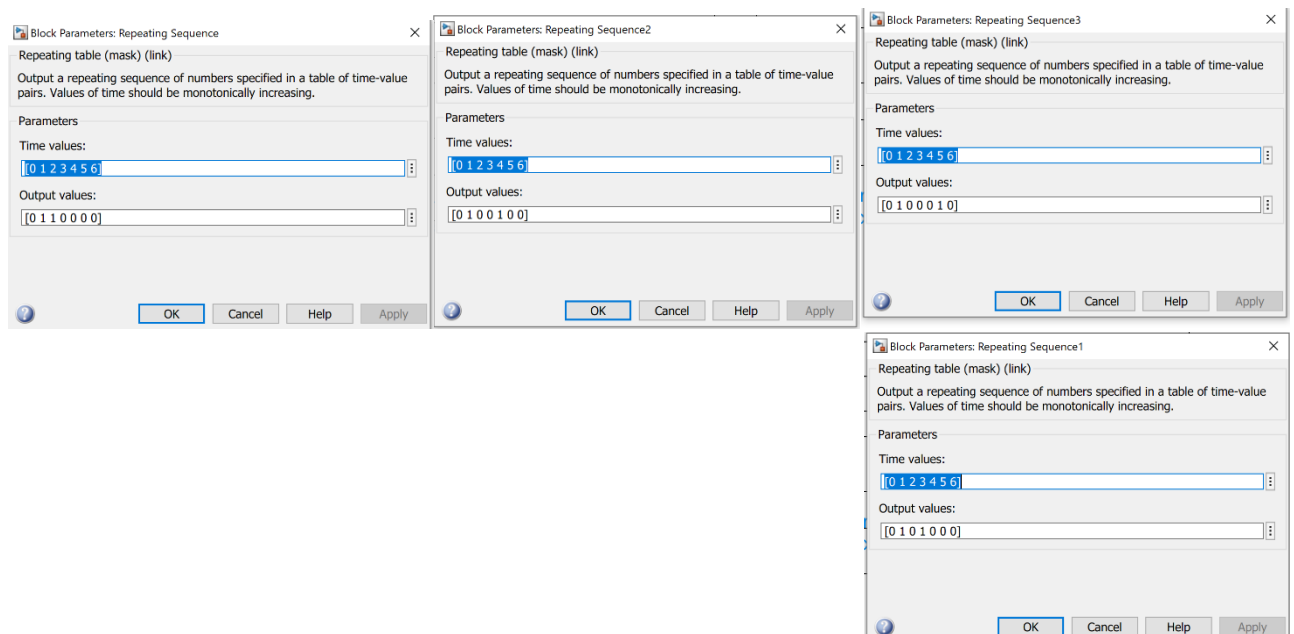


ورودی ها: B1, B2, B3, B4

خروجی ها: L1, L2, L3, L4



که برای نوشتن تست کیس مواردی مانند ایجاد event برای همه آن ها یا فقط برای یک مورد یا نبود event برای هیچ کدام استفاده شد



و در نهایت خروجی های مورد انتظار برای همه ورودی ها در زمان مشخص مورد ارزیابی قرار گرفت

Unit test : تست واحد در فولدر تست در پوشه master و slave پیوست شده.

تست عملکرد blackbox: برای انجام این تست برای هر کدام از نیازمندی ها که امکان داشت یک تست black box طراحی کردیم و آن را روی بورد پیاده کرده و صحت عملکرد آن را بررسی کردیم. تست ها و نتایج به شرح زیر هستند:

id	Req id	test	P/F	توضیحات
1	۱	یک گره master ایجاد شود و دو آردوئینو هر کدام با دو پیاده سازی برای slave با شماره id های ۱ و ۲ و ۳ ایجاد شود و صحت عملکرد و ارتباطات بررسی شود	pass	پیاده سازی انجام شد و slave ها به ترتیب اضافه شد و صحت عملکرد master در اتصال به هر ۳ slave و مشاهده وضعیت آنها ممکن بود
2	۲ و ۱۴	هر بار با deep switch یکی از slave ها را انتخاب کنیم و هر بار تنها وضعیت یکی را عیناً ببینیم و با تغییر وضعیت در master تنها وضعیت slave انتخاب شده تغییر کند	pass	
3	۳	از Master یک slave را انتخاب کنیم و وضعیت slave را عیناً در master مشاهده کنیم	Pass	
4	۴	همزمان هم گره ترموستات در مدار داشته باشیم هم switch روشنایی	pass	
5	۶ و ۸	در هر سویچ با کلیک بر روی هر push button تنها یک چراغ تغییر وضعیت دهد	Pass	
6	۱۳	هر بار با deep switch یکی از slave ها را انتخاب کنیم و هر بار تنها وضعیت یکی را عیناً ببینیم و با تغییر وضعیت در slave، وضعیت در master نیز تغییر کند	Pass	
7	۱۰	در master یک گره ترموستات را انتخاب کنیم و دما و دمای مطلوب را روی نمایشگر هر دو گره مشاهده کنیم	Pass	
8	۱۶	دکمه ها را فشار دهیم و مشاهده کنیم که در نمایشگر تغییر اعمال می شود	Pass	
9	۱۸ و ۱۹	مشاهده کنیم که وضعیت گره ترموستات در وب سرویس قابل مشاهده است و تغییرات اعمال می شوند	Pass	
10	۲۰	دمای ترشولد را به کمک دکمه ها پایین بیاوریم تا دمای اندازه گیری شده بیشتر شود و پنکه وصل شود	Pass	
11	۲۱	دمای ترشولد را به کمک دکمه ها بالا ببریم تا دمای اندازه گیری شده کم تر شود و پنکه قطع شود	pass	

تست های whitebox:

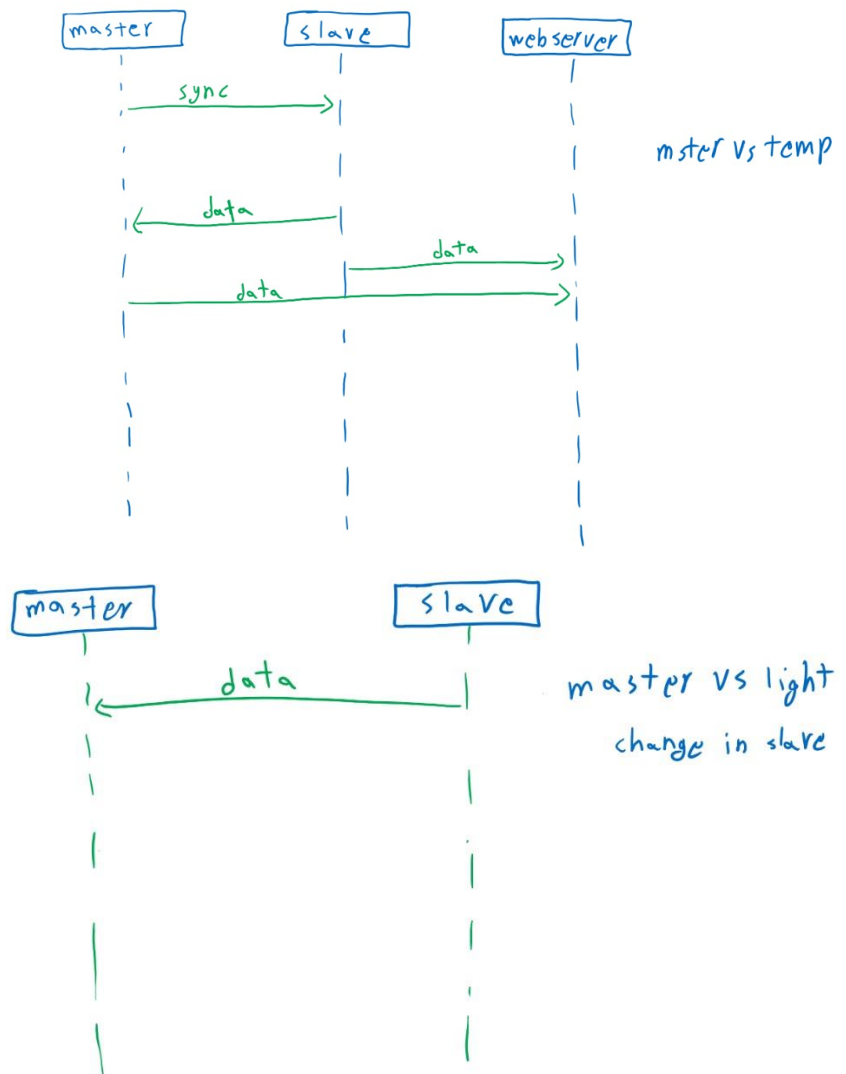
id	Req id	test	P/F	توضیحات
1	۲۱	با فشار دادن دکمه ها، پین متصل به آن مقدار ۱ منطقی بگیرد (به ازای هر دکمه)	pass	
۲	۱۱ و ۱۲	در هر جا که می خواهیم مقادیر را روی سریال بنویسیم، با هر بار نوشتن مقدار را روی پورت سریال چاپ کنیم، و هر جا که می خوانیم نیز مقدار خوانده شده را چاپ کنیم. هردو را در پنجره serial	pass	

		monitor زیر نظر بگیریم که زمانی که باید مقادیری نوشته شود سه فریم مطابق پروتکل نوشته شود و سپس همان ۳ فریم مطابق پروتکل خوانده شود (هر جا که خواندن و نوشتن داریم)		
۳	۱۱	همزمان دو دکمه را یکی در slave انتخاب شده و دیگری را در master فشار می دهیم. نباید بتوانند همزمان باس را در اختیار بگیرند (کنترل را ۱ کنند)	نتوانستیم بررسی دقیق کنیم ولی با فشار دادن دکمه ها به صورت دستی این اتفاق نمی افتاد یا سیستم fail نمی شد.	

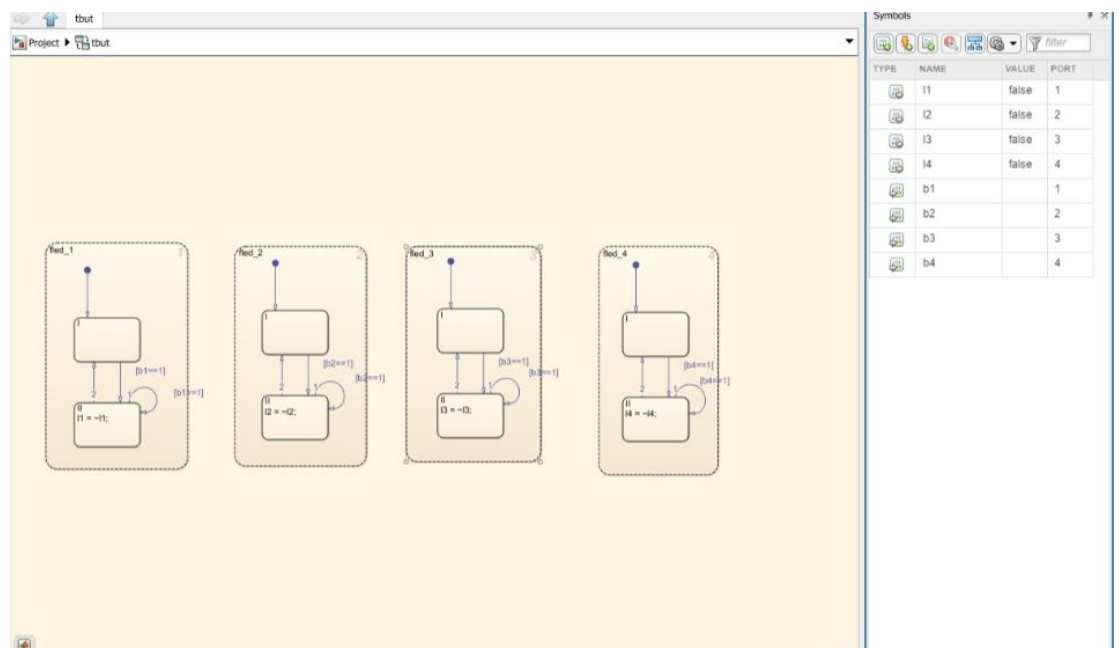
پیاده سازی و چالش های پروژه:

برای پیاده سازی پروژه از Arduino uno برای هر کدام از سوییچ ها استفاده کردیم ، برای ارتباط بین ۳ آردوئینو از ماژول max485 استفاده شد. برای هر کدام از سوییچ ها ۴ led به همراه ۴ push button در نظر گرفتیم که pushbutton ها به صورت pull down بسته شدند. برای انتخاب سوییچ در مستر از دیپ سوییچ استفاده شد. از DHT11 به عنوان سنسور دما استفاده کردیم و برای نمایش دیتای دما از ماژول وایفای NodeMCU v3 استفاده شد. به دلیل محدودیت در تعداد آردوئینو ها (۳ آردوئینو در اختیار داشتیم) برای پیاده سازی ۲ سوییچ ۴ پل و یک گره ترموستات و یک master ، یکی از آردوئینو ها را هم به یک گره ترموستات و هم به یک سوییچ ۴ پل اختصاص دادیم. نحوه کار این ماژول به این صورت است که اگر master، slave شماره ۲ را صدا زد کد مربوط به سوییچ اجرا می شود و اگر شماره ۳ را صدا زد، کد گره ترموستات اجرا می شود. یک آردوئینو هم به طور اختصاصی switch slave است و روی شماره ۱ صدا زده می شود. و یک آردوئینو هم نود master را میزبانی می کند.

ابتدا نیازمندی ها را مشخص کردیم و برای فرآیند ها sequence diagram رسم کردیم.



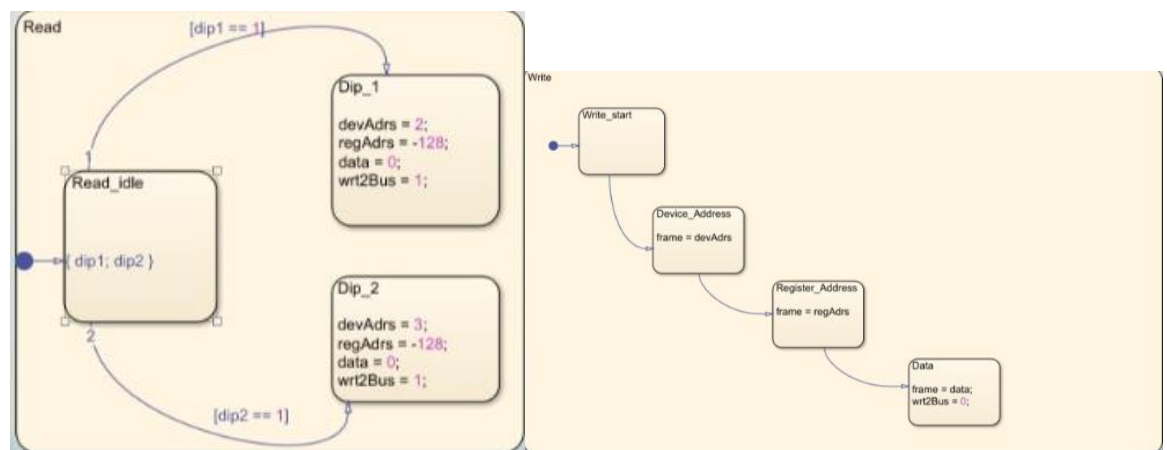
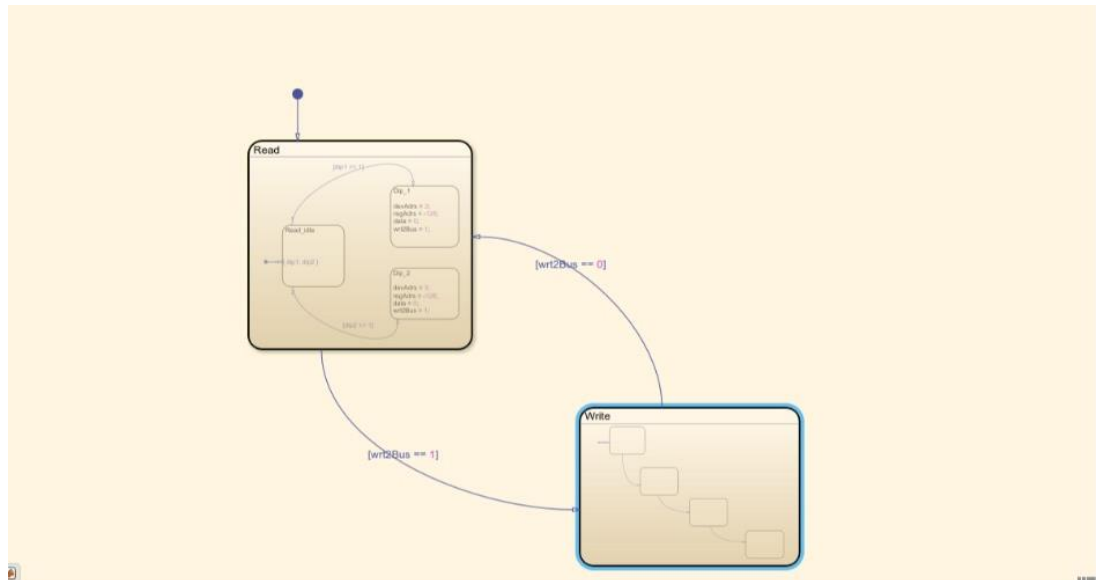
در پیاده سازی ابتدا فرض کردیم که هر کدام از نود ها یک سویچ ۴ پل هستند که ۴ led را کنترل می کنند. با toggle کردن هر کدام از پل ها مقدار led هم toggle می کند. برای پیاده سازی این بخش ابتدا مدل را در simolync ایجاد کردیم:



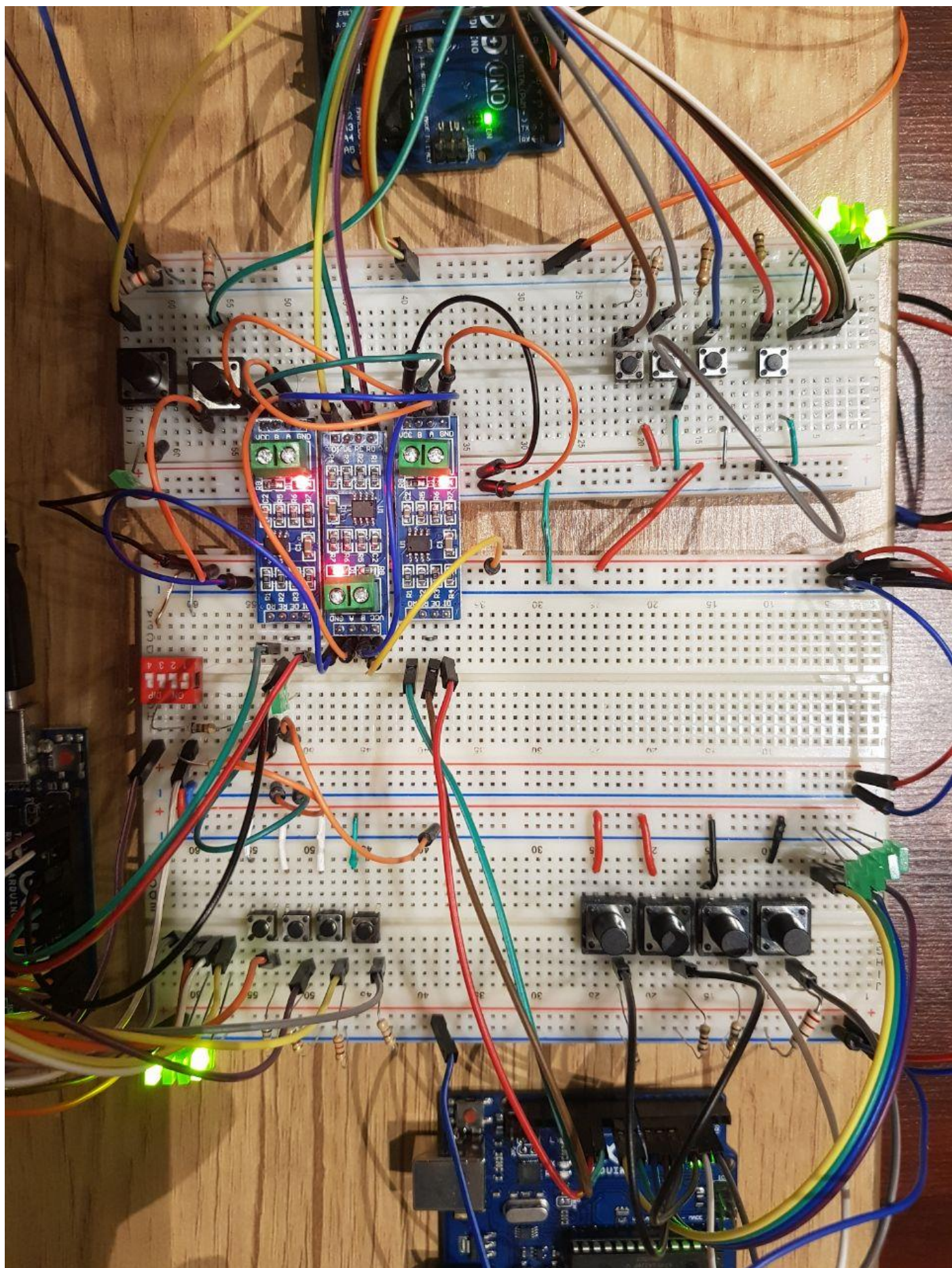
در این مدل به صورت موازی برای هر کدام از پایه ها که سیگنال bx (button) فعال شد، وارد استیت toggle می شویم و لامپ را روشن یا خاموش می کند و دوباره به استیت idle بر می گردد.

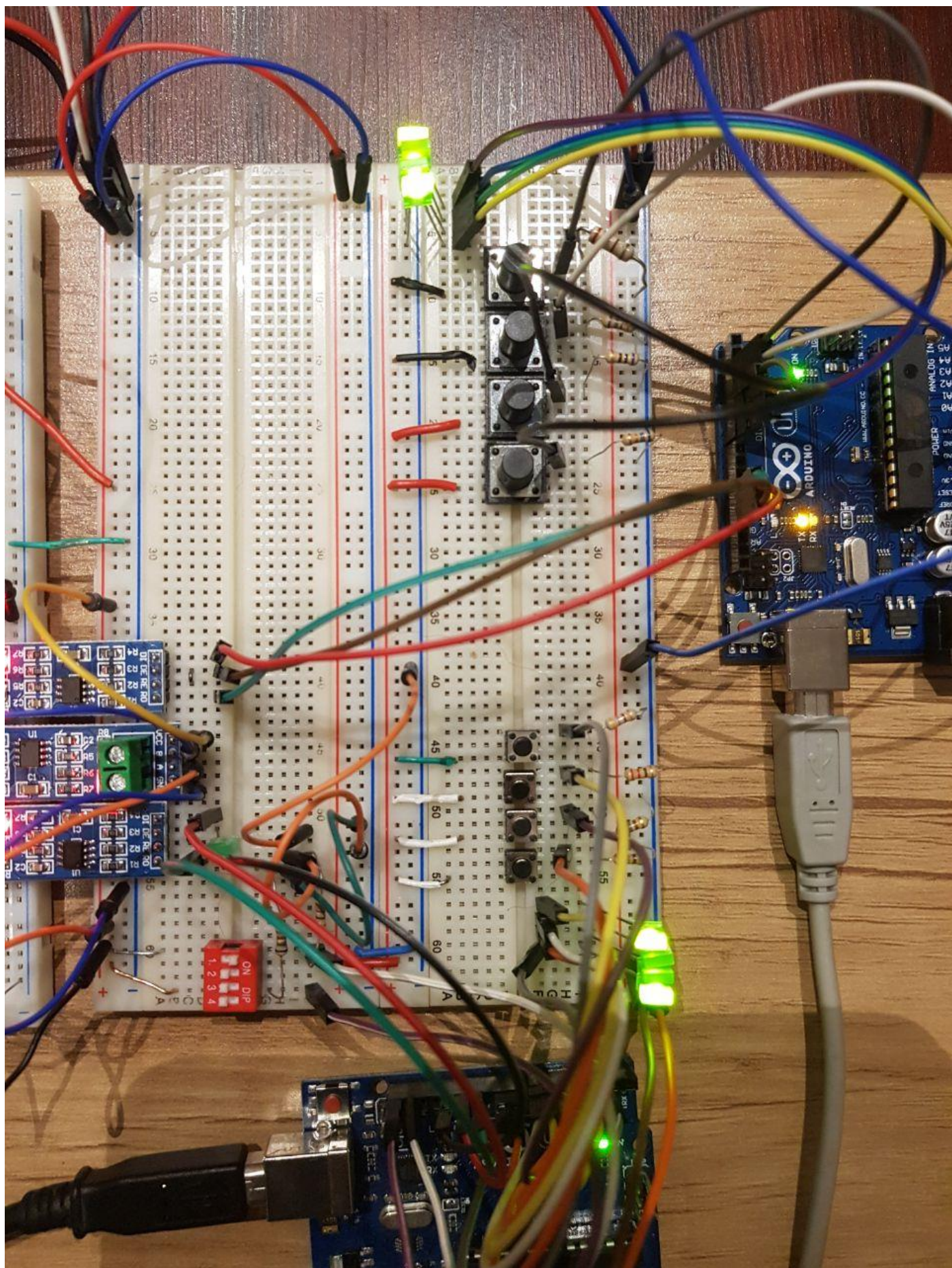
سپس خروجی کد گرفتیم و بر روی خروجی پیاده سازی مربوط به آردوینو را انجام دادیم و نتیجه به دست آمد .

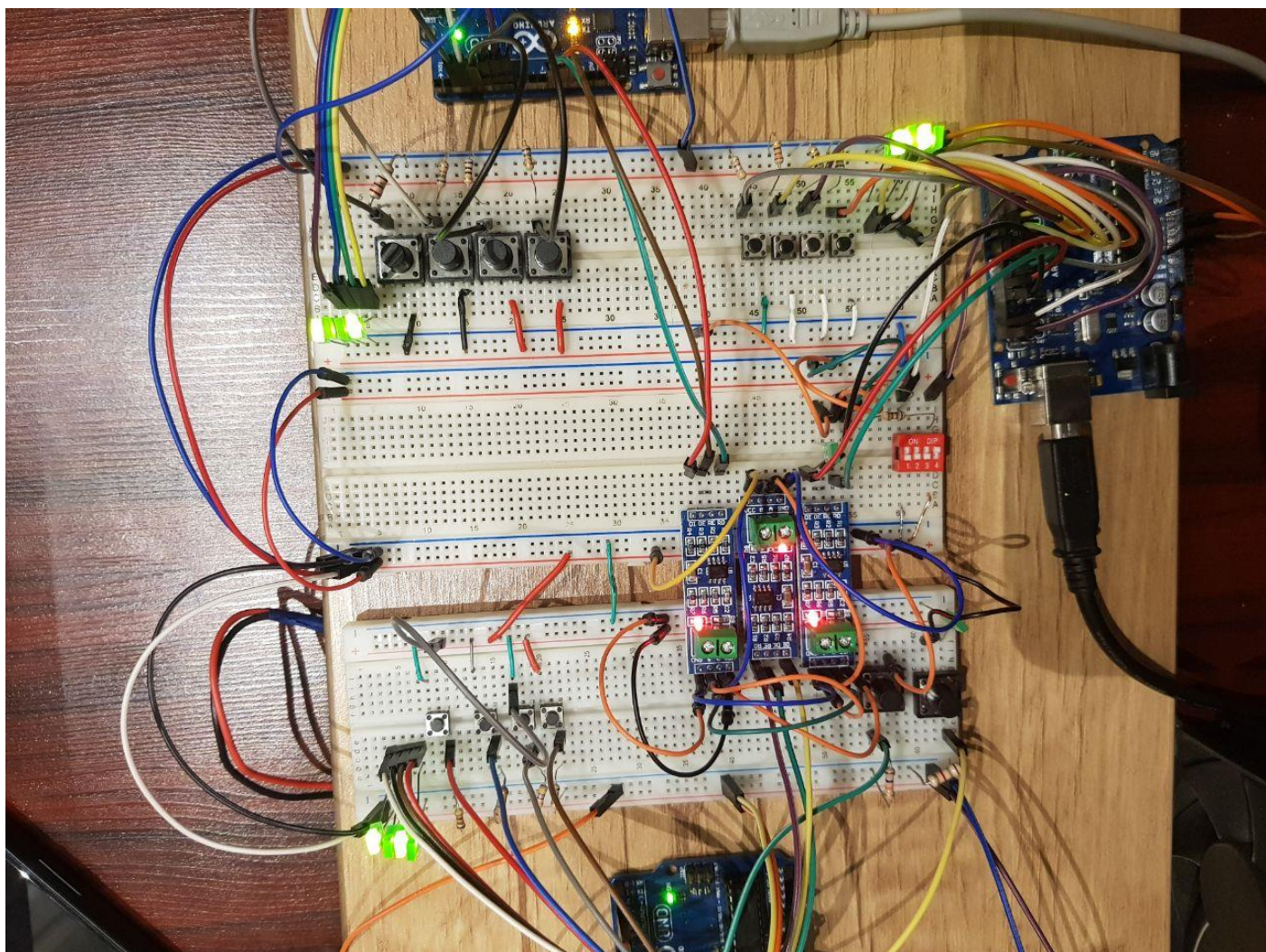
در مرحله بعد باید ارتباط بین نود ها را برقرار می کردیم. برای این کار ابتدا مدل را پیاده سازی کردیم:



سپس کد را استخراج کردیم و سعی کردیم پیاده سازی را انجام دهیم. برای پیاده سازی این بخش به کمک مدل ایجاد شده و مطابق پروتکل تعیین شده با ایجاد تغییراتی در کد پیاده سازی را انجام دادیم. این مدل را ابتدا برای اتصال دو نود ایجاد کردیم. سپس کد را به گونه ای تغییر دادیم که با انتخاب شماره دیوایس به کمک دیپ سویچ، مقدار فریم اول تغییر کند و بتوان روی هر slave با چک کردن فریم اول متوجه شد که دیتا برای کدام فریم است. در نتیجه مدل توسعه پذیر شد و حال می توانیم تعداد slave ها را زیاد کنیم. و در نهایت کار کرد ها را ایجاد کردیم.

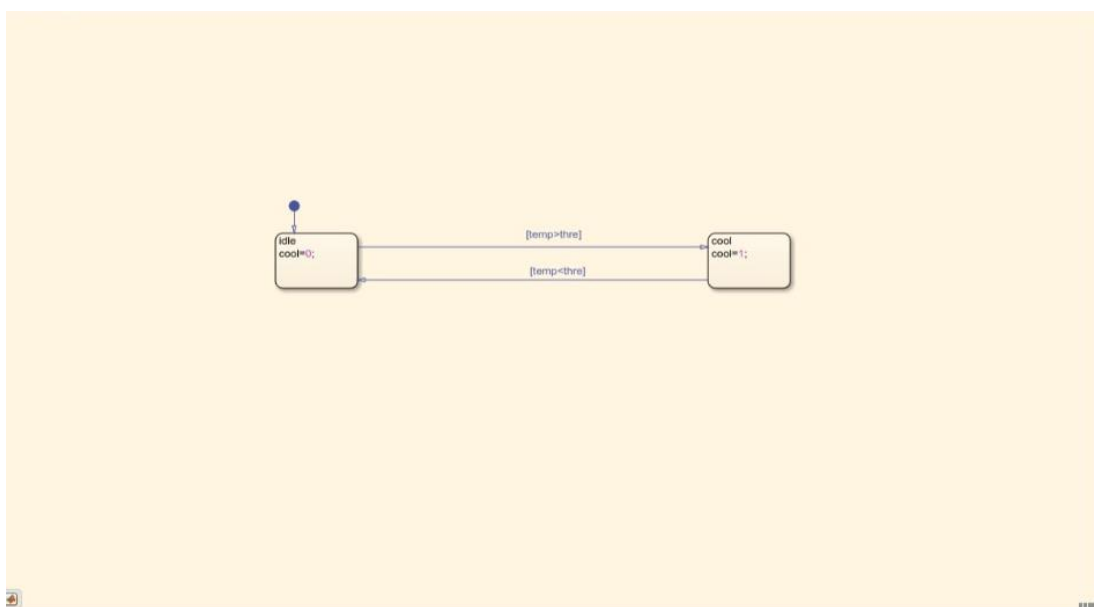






امتیازی:

برای پیاده سازی این بخش نیز ابتدا مدل را برای نود ترموستات ایجاد کرده و کد را استخراج کردیم. (به این صورت که اگر دما از دمای مطلوب بالا تر رود به mode خنک کردن می رویم و اگر کم تر یا مساوی باشد به مود idle می رویم. مقدار دمای مطلوب به عنوان ورودی داده می شود. در صورتی که هر کدام از سیگنال های بالا بردن یا پایین بردن دمای مطلوب فعال شوند، دما یک درجه زیاد یا کم می شود و در همان استیت می ماند



بعد از این که از عملکرد دماسنج، کلید ها ، پتکه و در نهایت تمام نود اطمینان حاصل شد، ارتباط با master را شکل دادیم به این صورت که با انتخاب slave توسط مستر در دو مرتبه هر بار در ۳ فریم طبق پروتکل داده دمای اندازه گیری شده و دمای مطلوب برای master ارسال می شود. برای نمایش این داده ها در هر دو طرف به جای نمایش روی lcd از ماژول wifi و استفاده از وبسرویس آن استفاده شده.

برای اتصال ماژول وای فای ابتدا تنظیمات کانفیگ آن را انجام دادیم و از طریق دو پورت سریال داده ها از نود ها برای آن ارسال می شود و ماژول آن را روی وب سرور نمایش می دهد. همچنین می توان عملکرد دکمه ها را نیز به این ماژول منتقل کرد.

چالش ها:

یکی از چالش ها اجرای همزمان دیباگ کردن به وسیله serial monitor و استفاده از پورت های سریال برای ارتباط با دیگر نود ها بود (به دلیل کم بودن پورت های سریال روی Arduino uno). برای حل این چالش ، از کتابخانه softwareserial استفاده کردیم که دو پین دیجیتال را به پین سریال تبدیل می کند (شبیه سازی می کند) و از آن برای ارتباط بین نود ها استفاده کردیم.

چالش بعدی مشکل کم بودن تعداد پین ها روی Arduino uno بود. مشکل مخصوصا زمانی نمود پیدا می کرد که می خواستیم روی یک آردوئینو هم ۴ led هم 4 دکمه و هم پایه های lcd را متصل کنیم . برای حل مشکل کمبود پین دیجیتال از پین های آنالوگ هم استفاده کردیم. این امکان روی آردوئینو وجود داشت که از پین های آنالوگ هم مقدار دیجیتال بخوانیم یا بنویسیم. از ان پین ها برای روشن کردن چراغ ها استفاده شد. همچنین برای حل مشکل تعداد بالای پایه های lcd، تصمیم گرفتیم که یک ماژول وای فای را به master و گره ترموستات متصل کنیم که بتوانیم داده روی هر دو گره را روی وب سرور ایجاد شده توسط این ماژول مشاهده کنیم.

چالش دیگر پروژه ایجاد دو پورت سریال بر روی یک آردوئینو بود (یکی برای ارتباط بین نود ها و دیگری برای ارتباط با ماژول وایفای). زیرا به طور همزمان نمی شود دو نمونه از software serial داشت. برای حل این مشکل پورت سریال آردوئینو را به ارتباط با وایفای اختصاص دادیم و از تابع listen(). برای تعیین پورت هایی که باید listen کنند استفاده کردیم.

برای ایجاد ارتباط بین ماژول ها به کمک باس سریال مطابق پروتکل در هر بار نوشتن برای هر کنترلری که می خواهد بنویسد پین کنترل را ۱ می کنیم و ۳ مقدار را روی سریال می نویسیم. سپس پین کنترل را ۰ می کنیم (کنترل ۱ برای نوشتن و ۰ برای خواندن). و برای خواندن هر زمان که روی باس مقدار وجود داشت، ۳ فریم می خوانیم، اگر ۳ فریم خوانده شده بود فریم ها را بررسی می کنیم و بر حسب مورد عملکرد متناظر را انجام می دهیم.