

## دانشگاه صنعتی امیر کبیر ( پلی تکنیک تهران )

گزارش کار تمرین پروژه درس سیستمهای ریزپردازندهای و مدارهای واسطه

محمد جواد رنجبر 9523048

امیرحسین شیخ السلامی 9523070

محمد حسن بیگلو 9523412

تابستان 99

## صورت مساله پروژه :

هدف این پروژه ، طراحی یک سیستم برای کنترل هوشمند دمای اتاق است .

طرح کلی سیستم به این شکل است که سیستم گرامایشی که شوفاژ یا یک رادیاتور است توسط یک شیراین شیر رادیاتور کنترل میشود ، و آلتراسونیک باز بودن در یا بسته بودن در را بررسی میکند و همینطور با سنسور دما به صورت دایم دما را اندازه میگیریم .

## بخش اول - اندازه گیری دما :

تایمر ۲ جوری تنظیم شده که هر ۱ ثانیه به بار تریگ میشه بعد اپدیت ایونتش هم فعاله بعد adc هم میذاریم که با اپدیت ایونت تایمر ۲ شروع کنه به تبدیل بعد تو اینترپتیش میایم مقدارش را به صورت فلوت بدست میارم تقسیم بر ۱۰ میکنیم که دما به دست بیاید

## بخش دوم - تعیین وضعیت درب اتاق :

تایمر ۳ رو تنظیم کردیم که اینترآپ ۱۰ یا ۱۱ میکرو ثانیه بده وقتی تابع read\_ultra صدا زده میشه، پایه تریگ های میشه تایمر ۳ هم شروع میشه که ۱۰ میکرو ثانیه رو بسازه...وقتی اینترپت ایت ۱۰ میکرو ثانیه اومد، میایم پایه تریگ رو لو میکنیم و تایمر ۵ که اینپوت کپچره رو فعال میکنیم تایمر ۵ هم جوری تنظیم شده که هر استپش ۱ میکرو ثانیه است. بار اولی که لبه بالا اکو میاد، مقدار کانتز ذخیره میشه و تنظیم میکنیم که حساس بشه به لبه پایین رونده بار دوم که اینترپت اومد باز هم مقدار کانتز رو ذخیره میکنیم بعد اینا از هم کم میکنیم بعد ضرب در سرعت صوت و تقسیم بر ۲ میکنیم که مسافت بدست بیاد بعد دوباره حساسش میکنیم به لبه بالا رونده.

## بخش سوم - کنترل شیر رادیاتور :

برای این کار از stepper motor استفاده کرده ایم . برای یک موتور n دندانه ای در هر پله موتور به اندازه  $360/n$  میچرخد روش یک پله چرخیدن موتور به شکل زیر است .

step \ wire	Blue	Pink	Yellow	Orange
1	on	off	off	off
2	Off	on	off	off
3	off	off	on	off
4	off	off	off	on

جدول ۱: مراحل مربوط به طی یک پله به روش full-stepping

با استفاده از جدول بالا در برای قدم یک pin را set میکنیم و بقیه را reset میکنیم سپس برای اینکه مقدار زاویه مناسب به دست بیاید باید عدد n را بدانیم در این موتور برای ما 2000 بود و باید برای 360 باید این تعداد قدم طی میکردیم و بین هر قدم باید یک میلی ثانیه دیلی باشد .

از آنجا که موتور جریان زیادی میکشد برای درایوکردنش از uln2003 کمک گرفته ایم

### بخش چهارم – ارتباط با میکروکنترلر از طریق SMS

برای این بخش به خاطر اینکه ماژول SIM800C در دسترس نبود با استفاده از Arduino due اطلاعات را ارسال میکردیم .

و به این صورت ابتدا uart4 و DMA را فعال میکنیم سپس برای ارسال داده ها از uart4\_TX استفاده میکنیم برای اینکار یک بافر تعریف کرده ایم و سپس اطلاعات مورد نیاز را در این بافر میریزیم و با دستور HAL\_UART\_Transmit ارسال میکنیم

برای دریافت با استفاده از HAL\_UART\_Receive\_DMA به صورت دائم منتظر دریافت داده است در بافر خود میریزد ما در فانکشن HAL\_UART\_RxCpltCallback این اطلاعات را به بافر خودمون منتقل میکنیم و به صورت استرینگ در میاوریم سپس در فانکشن Compare\_Strigns اگر جز دستورات مورد نظر بود فلگ مخصوص آن دستور را فعال میکنیم تا آن دستور اجرا شود .

### صورت مساله کلی :

حال در تابع main ابتدا به صورت دائم فاصله و دما اندازه گیری میشود و در ابتدا auto mode on پس بر اساس دمای مطلوب ( ما 27 درجه در نظر گرفته ایم ) موتور شیر آب را باز میکند ( به اندازه اختلاف زاویه مناسب میچرخد ) و در صورت بسته شدن در به صورت خودکار به انتهای زاویه (270) میرود و در صورت باز بودن ، باز برمیگردد به حالت پیدا کردن دمای مناسب حال اگر پیام turn off auto mode را دریافت کنیم فلگ آن تغییر میکند و موتور از این به بعد با دمای مطلوب دیفالتی که تعیین کرده ایم کار میکند همچنان مقدار 27 است تا اینکه کاربر و Set Temp to 32 deg را ارسال کنید میتواند دمای مطلوب را تغییر دهد و موتور از آن به بعد با آن دما کار میکند کاربر با ارسال هر پیام ، پیام متناسب با آن را دریافت میکند

فانکشن ها :

### **void read\_ultra(void)**

همان طور که در بالا توضیح داده شد برای خواندن فاصله است به این صورت که تایمر را استارت میکند و در تابع آن متغیر دیستنس تغییر میکند این متغیر چون در اینتراپت تغییر میکند برای اینکه به مقدارش دسترسی داشته باشیم باید به صورت volatile تعریف میشد

### **void stepper(int direction)**

این تابع برای این است که موتور به اندازه یک استپ حرکت کند

### **void stepper\_degree(int teta,int direction)**

در این تابع مقدار زاویه و جهتی که موتور میخواهیم حرکت کند را میدهیم و در یک لوپ به تعداد متناسب استپ در جهت مناسب حرکت میکند

### **void Send\_Stats()**

با استفاده از فلگ های مربوط به دستور ها ( خاموش یا روشن بودن حالت خودکار ) و همینطور باز یا بسته بودن در پیام مناسب را برای کاربر ارسال میکند

### **void HAL\_UART\_RxCpltCallback(UART\_HandleTypeDef \*huart)**

در این تابع پیام دریافتی از بافر را میگیریم و در بافر مورد نظر خودمون با انتهای 0 ذخیره میکنیم تا بتوانیم از توابع استرینگ استفاده کنیم سپس تابع Compare\_Strigns را صدا میزنیم تا فلگ های مربوط به دستور ها را تنظیم کند

### **void Compare\_Strigns()**

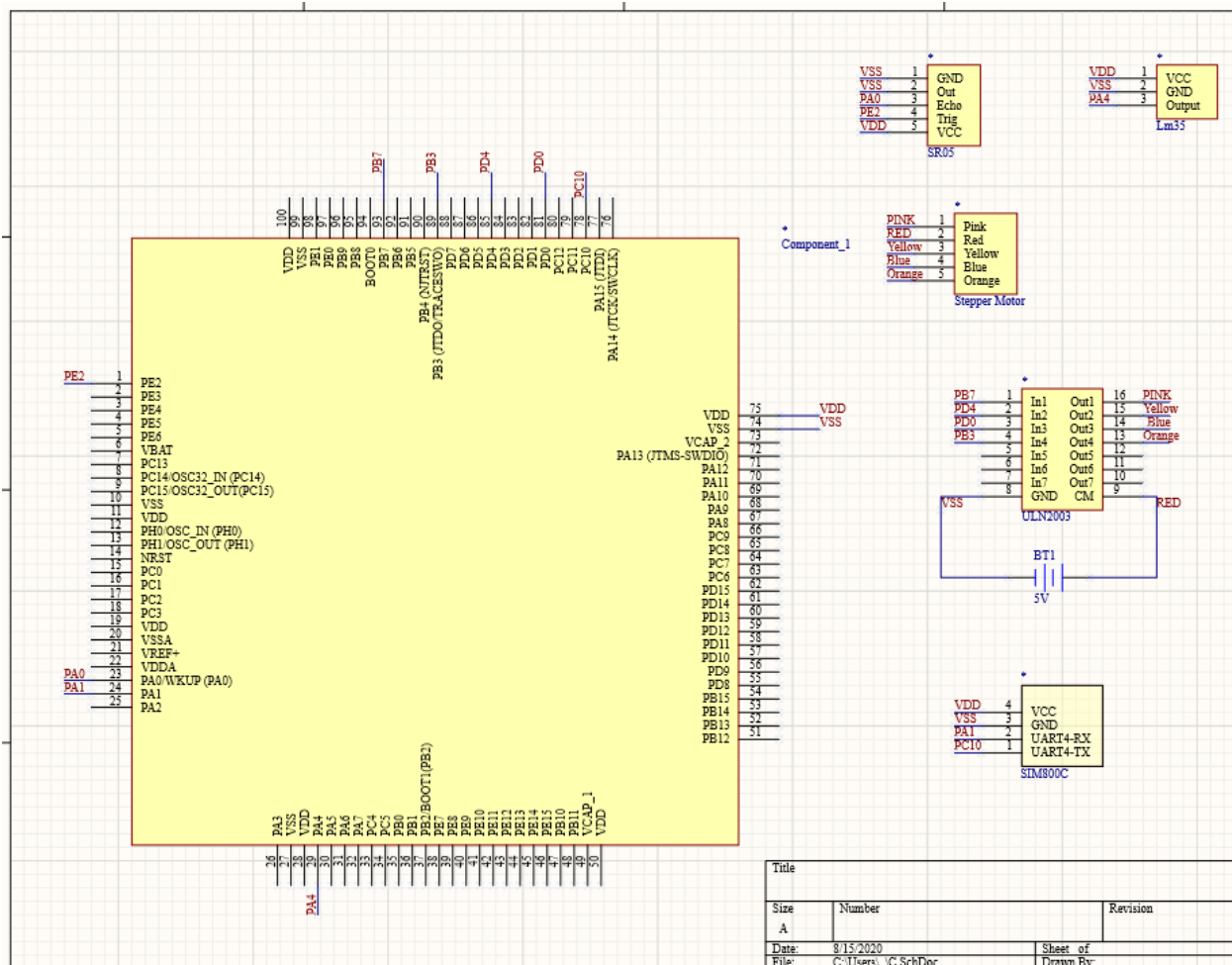
در این تابع پیام ارسالی کاربر را با پیام هایی که میتواند باشد مقایسه میکنیم و در صورت دریافت هر پیام مطابق با آن عملیات انجام میدهیم

برای تنظیم دما که از دستور Set Temp to 25 deg استفاده میشود باید عدد دما حتما به صورت دو رقمی نوشته شود .

**void HAL\_ADC\_ConvCpltCallback(ADC\_HandleTypeDef\* hadc)**

مقدار را میخواند و همانطور که گفته شد به دما تبدیل میکند

### نقشه مدار :



پینوشت : در مدار خودمون از آردوینو به جای ماژول sim500 استفاده کرده ایم ولی همینکار را انجام میدهد و همین پین ها فقط مورد استفاده قرار میگیرند

