

فربد فولادی-98243045

عرفان رفیعی اسکوئی-98243045

در ابتدا پورت های مورد نیاز برای ابزار مورد استفاده در پروژه را `define` کرده ایم برای ساده شدن کار در ادامه. همچنین امضای توابعی که به آن ها نیاز داریم را تعریف می کنیم.

در ابتدا یک `enum` تعریف می کنیم که حالات وضعیت های `idle` و `ok` و `checking` و `warning` و `danger` و `cooling` و `off` را مشخص می کند.

در ادامه به `initialize` کردن `timer` ها و `GPIO` ها و `ADC` می پردازیم.

در حلقه بی نهایت داخل کد چک می کنیم که اگر `state` برابر با `off` نبود مقدار دماسنج را در سون سگمنت نمایش دهد. در غیر این صورت اگر `state` برابر با `ok` یا `warning` باشد `ADC` را فعال می کند.

در تمام توابعی که برای `initialize` کردن هستند متناسب با دستورات از قبل تعریف شده در `HAL`، `initialize` شده اند.

در تابع `Seven_Segment_Decoder` با یک سویچ کیس مقدار مد نظر را دریافت می کنیم و با استفاده از تابع `Seven_Segment_Show` پین های مورد نیاز را `wright` می کنیم و مقدار `temp` را که محاسبه می کنیم به نمایش می گذاریم.

در تابع `Get_Temperature` متناسب با خروجی (`HAL_ADC_GetValue(&hadc1)`) که در `AD_RES` ذخیره می شود مقدار دما را خروجی می دهیم.

در تابع (`HAL_GPIO_EXTI_Callback(uint16_t GPIO_Pin)`) بررسی می کنیم که با فشردن هر یک از دکمه ها چه اتفاقی بیفتد و متناسب با هر یک `state` را نیز تغییر می دهد تا عملکرد مناسب با آن `state` را دریافت کنیم.

حال در تابع HAL_TIM_PeriodElapsedCallback(TIM_HandleTypeDef *htim)

که می توان گفت اصل کد در این قسمت است به این صورت است که متناسب با هر state که در آن قرار دارد عملیات متناسب با آن را انجام می دهد. اول چک می کند که htim2 فعال شده است یا نه. اگر فعال شده باشد به بررسی state ها می پردازد. حال اگر در حالت checking باشیم LED نارنجی به صورت چشمک زن در می آید و دما نیز محاسبه می شود.

اگر در state, danger باشیم LED قرمز به شکل چشمک زن در می آید و چک می کند که باید سیستم را خاموش کند را نه! (با قرار دادن state در حالت off و صفر کردن مقدار danger_counter)

اگر state, idle باشد اول چک می کند که باید به حالت idle برود یا خیر و اگر این شرط برقرار بود idle_counter را صفر می کند و state را برابر با off قرار می دهد.

اگر state, برابر با cooling بود سه حالت را بررسی می کند:

1. انتظار را بررسی می کند.
2. دما را نشان می دهد.
3. سیستم را در وضعیت idle قرار می دهد. به شرطی که در وضعیت danger باشد.

اگر هم state, off باشد سیستم را ریست می کند و اگر هیچ یک از این وضعیت ها نبود دما را حساب مبد کند و نمایش می دهد.

تابع Calculate_Temperature هم با بررسی سه حالت میانگین دما را حساب می کند. و با تابع Set_State وضعیت را ست می کند. به این شکل که:

1. کمتر مساوی 35 -> state = ok
2. بین 35 تا 45 -> state = warning
3. هیچ کدام از حالات بالا -> state = danger

متناسب با هر یک از این 3 حالت LED ها نیز ست می شوند.



