بسم الله الرحمن الرحیم

پرهام کریمی ریکنده 9423090

1. در ورودي تمام توابع یک اشاره گر به تایپ lcd\_tورودي داده شده است.علت چیست؟
   1. برای استفاده از GPIO نیاز داریم شماره ی پین های متصل به LCD و پورت های آنها را بدانیم:
      1. برای ارسال داده و دستور به ال سی دی نیاز داریم که از GPIO استفاده کنیم؛ چراکه درواقع نوع ارسال داده ها موازی است.
      2. پایه های کنترلی که در واقع پروتکل ارسال و دریافت را تشکیل می دهند، باید توسط GPIO کنترل شوند.
      3. در برخی موارد نیاز به خواندن از رجیستر های ال سی دی داریم که برای تنظیم پایه های دیتا به صورت ورودی باز هم به اطلاعات پین های متصل نیاز داریم
   2. برای تنظیم مد کاری ال سی دی متغیری در این تایپ گنجانده شده
   3. با این کار (تعریف این تایپ) تمامی اطلاعات مذکور را در یک ساختار یکپارچه کرده ایم
   4. چرا اشاره گر؟ 🡨 میتوانیم یک instance از این ساختار در بیرون main بسازیم تا global تعریف شود و سپس تنظیم مقادیر آن را در main انجام دهیم. هر بار که در هر تابعی نیاز به استفاده از این ساختار داشتیم، آدرس این instance را ارسال میکنیم.
      1. فایده: با این کار instance ساخته شده در ناحیه ی data (BSS) حافظه قرار خواهد گرفت اما اگر در main تعریف شود، از /انجایی که "محلی" است، در stack قرار می گیرد. خب که چی؟! 🡨 خب ما ترجیحمون به خالی موندن stack هست چون در فراخوانی توابع، متغیر های توابع [که محلی هستند]، در stack قرار می گیرند و متغیرهایی مانند این اگر در main تعریف شوند، در تمام طول اجرای تابع main (در واقع تمام مدت کارکرد میکرو!!) در stack باقی میمانند و جای ما رو تنگ میکنن!! پس میایم global تعریفش میکنیم.

یه چیز دیگه هم اینه که وقتی آدرس رو به تابع میفرستیم، فقط داریم یه متغیر نهایتا 32 بیتی رو جابجا میکنیم اما وقتی خودش رو بفرستیم به تابع، داریم به اندازه ی تمام متغیر های داخل این ساختار، جا اشغال میکنیم!

* 1. مطالعه ی این سوال و جواب هاش مفیده:

https://stackoverflow.com/questions/29708677/difference-between-global-variables-and-variables-declared-in-main-in-arm-c