پروژه باز کردن قفل درب

محمدحسین عشوری – ۹۷۱۴۹۰۶۸

درس آزمایشگاه ریزپردازنده

نحوه عملکرد وقفه :

به صورت کلی وقفه به استفاده بهینه از منابع کمک میکند ، فرض کنید برنامه ای نوشتیم که میکرو قرار است از صفحه کلید یک عددی را بخواند ، حال اگر بخواهیم خواندن به روش سرکشی صورت بگیرد ، میکرو باید دائما برود و ورودی ها را چک کند و ببیند که آیا دکمه ای فشرده شده است یا خیر ، که در این صورت میکرو اکثر زمان خود را صرف چک کردن ورودی ها می کند که اکثر این اوقات هم اصلا کلیدی فشرده نشده است اما با این حال میکرو باید سرکشی انجام دهد چون ممکن است کلیدی فشرده شود و تنها راه مطلع شدن سرکشی است.

راه حل حذف روش سرکشی استفاده از وقفه است ، ما با وقفه می توانیم روال اصلی برنامه را متوقف کنیم و به تابع سابروتین وقفه مورد نظر پرش کرده و دستورات آن را انجام داده و دوباره به روال اصلی برگردیم.

وقفه ها را میتوان حساس به پالس به چهار حالت تقسیم کرد:

·  وقفه در لبه بالا رونده پالس رخ دهد

·  وقفه در لبه پایین رونده پالس رخ دهد

·  وقفه در سطح  ۰ منطقی  رخ دهد

·  وقفه به ازای هر تغییر پالس یعنی از ۰ به ۱ و از ۱ به ۰ رخ دهد

ایجاد وقفه در تنظیمات اولیه خود کدویژن موجود است فقط باید وقفه مربوطه را فعال کرده و کد هایی را که میخواهیم در زمان وقفه ایجاد شوند را در درون تابع وقفه مورد نظر قرار دهیم تا در زمانی که میکرو وقفه می خورد آن کد ها اجرا شوند.

نکته دیگری که در استفاده از وقفه ها باید به آن اشاره کرد غیر فعال کردن پرچم ها در ابتدای تابع سابروتین است ، در واقع به این دلیل که ما نمی خواهیم  زمانی که در حال رسیدگی به یک وقفه هستیم ، یک وقفه دیگری نیز رخ دهد ، به همین دلیل پرچم وقفه را غیر فعال میکنیم تا در زمانی که در حال رسیدگی به وقفه حال حاضر هستیم وقفه دیگری صورت نگیرد .هر وقت وقفه حال حاضر به اتمام رسید ، دوباره پرچم وقفه را فعلا می کنیم.

این کار با استفاده از این دو دستور انجام می شوند :

#asm(“sei”); // برای فعال کردن پرچم وقفه

#asm(“cli”);// برای غیر فعال کردن پرچم وقفه

توضیحات پروژه :

در این پروژه قرار است رمز کاربر با رمز وارد شده از صفحه کلید مقایسه شود و در صورت یکسان بودن یک چراغ سبز روشن شود( چراغ سبز نماینده همان درب الکتریکی است که میتوان به جای آن موتور های DC یا ... نیز گذاشت ولی به این دلیل که پروتئوس بنده مشکل داشت از LED استفاده کردم).

چالشی که در این باره وجود دارد مقایسه رمز کاربر است ، اگر ما رمز کاربر را در یک متغیر عمومی ذخیره کنیم، برنامه همواره به درستی کار می کند ، مشکل آن زمانی به وجود می آید که میکرو به هر دلیلی از اول راه اندازی شود در این صورت رمز کاربر نیز از بین می رود .

برای مرتفع کردن این مشکل ما نیاز داریم که رمز کاربر را در حافظه EEPROM خود میکرو ذخیره کرد به همین دلیل یک متغیر در حافظه به نام PASS ایجاد می کنیم و از این به بعد رمز کاربر را در آنجا نگهداری میکنیم ، با این کار اگر میکرو از اول هم راه اندازی رمز کاربر به این دلیل که از حافظه میکرو خوانده می شود از بین نمی رود.

نحوه کار پروژه به این صورت است که در تابع MAIN  بعد از مقدار دهی به ثبات های میکرو ما پرچم وقفه را فعال میکنیم و برای اینکه تفاوت را بتوان در اجرا دید در حلقه اصلی کاری که انجام می دهیم نشان دادن یک متن متحرک است. در این بین حال اگر وقفه ای صورت بگیرد تمام وضعیت های حال حاضر میکرو که شامل جایگاه متن در صفحه Lcd  و ... ذخیره شده و سیستم به وقفه رسیدگی می کند.

در تابع سابروتین وقفه ما کد هایی را که قبلا به صورت سرکشی بررسی می کردیم را قرار میدهیم . در واقع همان عمل سرکشی از صفحه کلید را اینبار وقتی که وقفه صورت گرفته انجام می دهیم.

رمز را از صفحه کلید می خوانیم و اگر رمز وارد شده صحیح بود چراغ سبز میشود ولی اگر رمز کاربر با رمزی که در حافظه قرار دارد همخوانی نداشت ، از تعداد دفعاتی (متغیر j ) که کاربر مجاز به زدن رمز است کم می شود و بعد از سه بار اشتباه وارد کردن رمز ، سیستم قفل می شود و دیگر به کاربر اجازه زدن رمز را نمی دهیم.

اینجا نیز برای اینکه امنیت بالاتر رود بهتر است تعداد دفعاتی که کاربر می تواند رمز را وارد کند به جای اینکه متغیر j  را عمومی تعریف کنیم آن را در حافظه ذخیره کنیم، به این دلیل که اگر تعداد دفعات مجاز (متغیر j ) یک متغیر عمومی باشد همان مشکل پسورد پیش آمده و با هر با راه اندازی مجدد میکرو ، کاربر میتواند دوباره رمز های مختلف را امتحان کند.