

# آسانسور دوبل هوشمند

سعید خسروآبادی

[s\\_khosro2003@yahoo.com](mailto:s_khosro2003@yahoo.com)

موسسه آموزش عالی سجاد

احسان ریحانی

[ereihani@yahoo.com](mailto:ereihani@yahoo.com)

موسسه آموزش عالی سجاد

## چکیده :

در این پروژه چگونگی انتخاب یک کابین و کنترل آن بدین صورت است که زمان رسیدن شخص به مقصد توسط هر دو کابین محاسبه شده و با یکدیگر مقایسه می شود. هر کدام که زمان کمتری را در محاسبه به دست آورد جهت سرویس دهی انتخاب می شود. محاسبه زمان برای هر کابین به این صورت می باشد که فرض می شود فاصله هر دو طبقه متوالی پس از بسته شدن درب و حرکت آسانسور به بالا یا پایین در مدت ۳ ثانیه طی شده و در هر طبقه ای که شخص یا اشخاص قصد سوار یا پیاده شدن از آسانسور را داشته باشند متوسط ۵ ثانیه توقف دارد و درب باز می ماند. باز و بسته شدن درب و حرکت در آسانسورهای واقعی توسط سنسور کنترل می شود. بدین صورت که اگر طی زمان خاصی که حدودا ۱ تا ۲ ثانیه می باشد کسی پیاده یا سوار نشود، درب بسته و آسانسور شروع به حرکت می کند. سیستم مذکور زمان توقف ۵ ثانیه را در طبقات مورد نظریه صورت متوسط در نظر خواهد گرفت.

## واژگان کلیدی: آسانسور، میکروکنترلر

## ۱-مقدمه

این آسانسور دارای دو کابین مجزا از یکدیگر می باشد که توسط یک سیستم مرکزی کنترل و فرماندهی می شوند. در این آسانسور بر خلاف آسانسورهای معمولی کلید انتخاب طبقات بیرون از کابین قرار دارند و شخص درخواست کننده سرویس، قبل از سوار شدن به آسانسور به جای فشردن کلید درخواست، کلید طبقه مورد نظرش (مقصد) را فشار می دهد. سیستم مرکزی با توجه به تعداد مسیر اشخاصی که قبلا در هر یک از دو کابین سوار بوده اند کابینی را برای سرویس دهی انتخاب می کند که بتواند شخص را زودتر به مقصد برساند. از مزایای این سیستم صرفه جویی قابل ملاحظه در وقت و انرژی می باشد.

## ۲-شمای کلی پروژه :

آسانسور مورد بحث برای یک ساختمان ۱۲ طبقه با دو کابین طراحی و برنامه ریزی شده است. اما قابلیت افزایش طبقات و کابینها را نیز به سادگی داراست. اکنون برای ملموس تر شدن توضیحات به ارائه یک مثال ساده می پردازیم. برای سادگی کار فرض می کنیم در ابتدا هیچ شخصی سوار آسانسور نیست و هر دو کابین خالی و ساکن

هستند. همچنین کابین یک در طبقه سوم و کابین دو در طبقه دهم می باشد. شخص قصد دارد از طبقه ششم به طبقه چهارم برود. در نتیجه کابین اول که در طبقه سوم قرار دارد بایستی سه طبقه به سمت بالا و پس از ۵ ثانیه توقف و سوار کردن شخص مورد نظر دو طبقه به سمت پایین حرکت کند یعنی در مجموع

$$t_1 = (3 + 2) \times 3 + 5 = 20 \text{sec}$$

طول خواهد کشید تا شخص به طبقه ششم برسد. محاسبه زمان به صورت مشابه برای کابین دوم نیز نتیجه

$$t_2 = (4 + 2) \times 3 + 5 = 23 \text{sec}$$

را خواهد داشت که پس از مقایسه  $t_1$  و  $t_2$  بدیهی است که کابین اول برای سرویس دهی انتخاب می شود. حال همین طریق محاسبه زمان برای هر چند نفر که در کابین ها حضور دارند و در طبقات خاصی قرار است پیاده شوند محاسبه خواهد شد و بر اساس آن سیستم به اشخاص جدید سرویس دهی می کند.

### ۳- شبیه سازی سیستم:

#### ۳-۱- صفحه کلید:

در سیستم مورد بحث چون ساختمان ۱۲ طبقه می باشد. یک صفحه کلید ۱۲ تایی برای انتخاب طبقات در هر طبقه مورد نیاز است. از این رو که امکان نمایش ۱۲ صفحه کلید به عبارتی ۱۴۴ کلید در قطع شماتیک وجود ندارد سیستم طوری طراحی شده است که فقط دارای یک صفحه کلید است و شخص درخواست کننده سرویس در هنگام انتخاب از روی صفحه کلید بایستی دو کلید را بفشاریم. کلید اول عنوان طبقه ای است که شخص در آنجا حضور دارد ( مبدا ) و کلید دوم عنوان طبقه ای است که شخص قصد رفتن به آنجا را دارد تلقی می شود ( مقصد ).

#### ۳-۲ نمایشگرها (7-segment)

دو جفت نمایشگر ۷ قسمتی در سیستم وجود دارد که هر جفت مخصوص یک کابین هستند به علت اینکه طبقات از ۰ تا ۱۱ شماره گذاری شده اند و اعداد ۱۰ و ۱۱ دو رقمی هستند از دو نمایشگر برای هر کابین استفاده شده است. نمایشگرها را در هر لحظه محل حضور هر کابین را نشان می دهند. مثلاً اگر یک کابین در طبقه سوم

باشد و رو به بالا حرکت کند نمایگر عدد 03 نشان می دهد و تا زمانی که کابین به طبقه چهارم نرسیده است عدد روی نمایشگر حفظ خواهد شد. اما به محض رسیدن به طبقه چهارم عدد روی نمایشگر به 04 تغییر خواهد یافت.

**۳-۳- LED ها:** برای نشان دادن عملکرد درب ها و موتورهای آسانسور به صورت شماتیک از LED های سبز و قرمز رنگ استفاده شده است. موتورها به صورت بالا رونده و پایین رونده می باشند و درب ها هم در حالت باز یا بسته قرار قرار دارند برای موتور بالا رونده LED قرمز و برای موتور پایین رونده LED سبز در نظر گرفته شده است. اگر آسانسور ثابت باشد و حرکتی نکند این دو LED خاموش خواهند بود. برای درب آسانسور نیز دو LED در نظر گرفته شده که LED سبز مبین باز بودن و LED قرمز مبین بسته بودن درب خواهد بود. واضح است که این دو LED هیچ گاه با هم خاموش یا روشن نمی شوند.

تمام موارد این قسمت عینا برای کابین ۲ نیز تکرار شده اند.

#### ۴- سخت افزار:

کلا در سخت افزار این سیستم از تراشه میکروکنترلر، تراشه های جانبی جهت بافر کردن پورت های خروجی تراشه AND، صفحه کلید، نمایشگر و ترانزیستور، LED و مقاومت استفاده شده است.

#### ۴-۱- میکروکنترلر: تراشه (8051)

این تراشه دارای چهار پورت ۸ بیتی ( $P_1$  تا  $P_3$ ) می شود که از پورت صفر ( $P_0$ ) آن جهت روشن کردن نمایشگرهای a تا l استفاده شده است. پورت ۱ ( $P_1$ ) جهت اتصال صفحه کلید به میکروکنترلر به کار رفته است.

از پورت ۳ ( $P_3$ ) بجز پایه های ۱۲ و ۱۳ که برای استفاده از وقفه های خارجی استفاده شده اند و همچنین دو پایه از پورت ۲ ( $P_2$ ) برای نشان دادن وضعیت موتورها و درب ها در هر لحظه توسط LED های سبز و قرمز استفاده شده است و در نهایت پورت ۲ ( $P_2$ ) که از چهار پایه آن برا راه اندازی (Switching) نمایشگرهای موجود در سیستم استفاده شده است. پایه ۱۲ (int o) نیز در هنگام فعال شدن مشخص کننده فشرده شدن کلید می باشد که در ادامه در مورد آن توضیح داده خواهد شد. پایه ۳۱ (EA) نیز به علت عدم وجود حافظه خارجی همیشه به ولتاژ مثبت (high) متصل می باشد.

#### ۴-۲- صفحه کلید:

در این سیستم از یک صفحه کلید ماتریسی ( $3 \times 4$ ) استفاده شده است که برای درخواست سرویس بایستی دو کلید فشرده شود. (کلید اول به عنوان مبدا و کلید دوم به عنوان مقصد می باشد).

برای اتصال صفحه کلید به تراشه میکروکنترلر از ۷ پایه پورت ۱ ( $P_1$ ) این تراشه استفاده شده است. همچنین سطرهای این ماتریس علاوه بر اتصال به تراشه میکروکنترلر به ورودی های یک تراشه AND نیز متصل شده اند و خروجی گیت AND به پایه ( $P_{3,2}$ ) متصل شده است و با فشار کلید، این پایه فعال شده و در حین اجرای برنامه اصلی این پایه بررسی می شود تا در دریافت کلید انجام گیرد.

#### ۴-۳- نمایشگرها (7-segment):

در این سیستم از دو جفت نمایشگر ۷ قسمتی برای نشان دادن محل کابین ها استفاده شده است نمایشگرها از نوع کاتد مشترک می باشند یعنی برای روشن شدن آن ها پایه مشترک (کاتد) بایستی به صفر و تک تک آندهای ۷ قسمتی به ولتاژ مثبت وصلند. به این لحاظ این که LED ها برای روش شدن کامل به جریانی حدود  $10\text{ mA}$  نیاز دارند و با توجه به این که جریان خروجی پایه های میکروکنترلر  $4\text{ mA}$  می باشند در نتیجه اتصال مستقیم میکروکنترلر به نمایشگرها موجب کشیدن جریان زیاد و آسیب دیدن تراشه می شود. برای رفع این مشکل از یک تراشه 74573 که در حالت خاص مانند یک بافر معمولی عمل می کند استفاده شده است. این تراشه توانایی ایجاد جریان لازم برای روشن شدن LED ها را داراست. ورودی های این تراشه به میکروکنترلر و خروجی های آن با یک مقاومت کم اهم به سرهای آند نمایشگر متصل می شود. پایه کاتد نمایشگرها نیز به یک ترانزیستور منفی NPN که از آن به عنوان یک سوئیچ استفاده شده است متصل می شود. طریقه اتصال ترانزیستورها به این صورت است که کلکتور به کاتد نمایشگر و امیتر به زمین متصل شده است. بیس این ترانزیستورها نیز به پایه های میکروکنترلر متصل است. این پایه توسط نرم افزار موجود کنترل و فرمان دهی می شوند به این صورت که هرگاه بیس یک ترانزیستور صفر شود در کلکتور یک خواهیم داشت و ترانزیستور خاموش بوده و در نتیجه نمایشگر مربوط به آن ترانزیستور نیز خاموش خواهد بود. اما وقتی در پایه بیس یک داشته باشیم ترانزیستور به حالت اشباع رفته و کلکتور به امیتر متصل و در نتیجه در کلکتور صفر خواهیم داشت و در نتیجه این عمل روشن شدن نمایشگر مربوط به آن ترانزیستور می باشد.

در مورد نمایشگر توضیح این نکته ضروری است که نمایشگرها به صورت مالتی پلکس به یکدیگر متصل شده اند یعنی تمام حروف مشابه چهار نمایشگر (a تا g) به ترتیب به یکدیگر متصل شده اند. اما در هر لحظه تنها یکی از

نمایشگرها روشن و بقیه نمایشگر خاموش می باشند اما چون برنامه اصلی مدام در حال جاروب کردن (Refresh نمایشگرها می باشد و به علت فرکانس خاص کار (حدود 60 Hz) هر چهار نمایشگر همیشه روشن دیده می شوند.

## ۵- نرم افزار سیستم :

در ابتدای برنامه یا پس از Reset شدن هر دو کابین در طبقه همکف که شماره 00 برای آن در نظر گرفته شده است قرار می گیرند. همچنین در حافظه هایی که مخصوص نگهداری شماره طبقات است هیچ عددی وجود ندارد. در این حالت کار سیستم فقط جاروب (Refresh) نمایشگرها جهت نمایش عدد 00 و جاروب صفحه کلید برای دریافت کلید می باشد. به محض فشردن کلید، زیر برنامه SCAN اجرا شده و کد کلید فشرده شده شناسایی و در محلی از حافظه ذخیره سازی می شود. اما همان طور که قبلا توضیح داده شد این سیستم کلید اول را به عنوان مبدا مشخص در نظر می گیرد. به همین لحاظ تا فشردن کلید دوم به برنامه اصلی برگشته و در ادامه آن را اجرا می کند. هنگامی که کلید دوم فشرده شد به عنوان مقصد مشخص تلقی شده و در حافظه قرار می گیرد. در این حالت که هر دو کابین زمان یکسانی را برای رساندن شخص به مقصد دارا هستند توسط نرم افزار کابین یک برای سرویس دهی انتخاب می شود. در حالت کلی هر جفت کلید که وارد شد در مکانی از حافظه قرار می گیرد. سپس عملیات محاسبه زمان برای به مقصد رساندن شخص که تقاضای سرویس کرده است توسط هر دو کابین محاسبه می شود. زمان کمتر، انتخاب بهتر را مشخص می کند. سپس دو کلید فشرده شده در میان عددهای قبل که در حافظه کابین مورد نظر وجود داشته جاسازی شده تا کابین پس از رسیدن به این طبقات توقف کرده درب را باز و مسافر را سوار یا پیاده کند. تقاضای بعد نیز به همین روش در حافظه یکی از کابین ها قرار گرفته و در کل سرویس دهی بهینه انجام می شود. توضیح این نکته ضروری است که صفحه کلید حدودا هر 20 ms جاروب می شود که بایستی علاوه بر دریافت کلید عمل لرزش گیری نیز روی آن انجام شود. همچنین اگر کلیدی به مدت زیاد فشرده شود و یا در اثر خرابی همیشه در حالت وصل باشد برای جلوگیری از خواندن این کلید یک نسبت تعریف شده است که هنگامی که کلید فشرده شده است این بیت مقدار صفر را خواهد داشت و به محض باز شدن کلید، در جاروب بعدی صفحه کلید مقدار این بیت یک خواهد شد. این بیت نیز هر 20 ms چک می شود. در صورت یک بودن آن برنامه اجرا می شود و اگر صفر بود به منزله فشردن کلید می باشد و در چک کردن بعدی در صورت صفر بودن این بیت، این مفهوم استنباط می شود که کلید هنوز رها نشده است و نرم افزار کلید را در حافظه قرار می دهد. هنگامی که نرم افزار از زیر برنامه SCAN برمی گردد پایه P2.0 که به عنوان یک قفل تعیین شده است بررسی می شود. این پایه تمام شدن وقت 3 ثانیه جابجایی یا 5 ثانیه توقف کابین یک را مشخص می کند. این یک

بودن پایه به معنی تمام شدن این وقت است. پس هنگامی که این پایه چک شد و صفر بود مشخص می شود کابین یک برای جابجایی و یا توقف مشغول است و نرم افزار به بررسی کردن کابین دوم می پردازد. برای این کابین نیز پایه مشابهی که P2.1 می باشد تعریف شده است. هنگامی که این پایه ها یک باشند شماره طبقات موجود در هر حافظه که مقصدهای مورد نظر هستند می توانند با محل حضور کابین ها مقایسه شوند. اما هنگامی که در حافظه کابین یک عددی نباشد برنامه به بررسی کابین دو که بررسی پایه P2.1 و تعداد اعداد موجود در حافظه دو می باشد، پرس می کند. ولی اگر عددی یا عددی در در حافظه کابین یک موجود باشد طبقه ای که کابین یک در آنجاست با اولین عدد موجود در حافظه کابین یک مقایسه شده در صورت مساوی نودن درب بسته و موتور مربوطه روشن می شود و کابین یک به سمت طبقه مورد نظر حرکت کرده و پس از 3 ثانیه که زمان یک طبقه جابجایی است عدد نمایش داده شده کابین یک، یک واحد اضافه یا کم می شود. به همین ترتیب کابین دو نیز بررسی می شود. برای ایجاد زمان حرکت کابین به یک طبقه بالا یا پایین تر که 3 ثانیه می باشد از تایمر 0 استفاده شده است. این تایمر یک زمان 20 ms را محاسبه کرده، سپس از رجیسترهایی که برای هر کابین عدد 150 در آن ذخیره شده یک واحد کم کرده و مجدداً به بررسی کردن کابین یک می پردازد. پس از آن دوباره یک تایمر 20 ms گرفته شده و یک داده از رجیستر مربوط که زمان 3 ثانیه را محاسبه می کند، کم می کند. این مرحله تا پایان صفر شدن رجیسترها ادامه می یابد. صفر شدن هر یک از این رجیسترها به معنای پایان وقت 3 ثانیه مربوط به کابین مورد نظر می باشد و کابینی که رجیستر مربوط به آن صفر شده می تواند پس از بررسی کابین و عدد جدید حافظه وضعیت مناسب برای موتور و درب آن کابین ایجاد کند. نمایشگر خاموش می باشند اما چون برنامه اصلی مدام در حال جاروب کردن (Refresh) نمایشگرها می باشد و به علت فرکانس خاص کار (حدود 60 Hz) هر چهار نمایشگر همیشه روشن دیده می شوند.

## ۵-۱-نمای کلی برنامه:

در ابتدای برنامه یا پس از Reset شدن هر دو کابین در طبقه همکف که شماره 00 برای آن در نظر گرفته شده است قرار می گیرند. همچنین در حافظه هایی که مخصوص نگهداری شماره طبقات است هیچ عددی وجود ندارد. در این حالت کار سیستم فقط جاروب (Refresh) نمایشگرها جهت نمایش عدد 00 و جاروب صفحه کلید برای دریافت کلید می باشد. به محض فشردن کلید، زیر برنامه SCAN اجرا شده و کد کلید فشرده شده شناسایی و در محلی از حافظه ذخیره سازی می شود. اما همان طور که قبلاً توضیح داده شد این سیستم کلید اول را به عنوان مبدا مشخص در نظر می گیرد. به همین لحاظ تا فشردن کلید دوم به برنامه اصلی برگشته و در ادامه

آن را اجرا می کند. هنگامی که کلید دوم فشرده شد به عنوان مقصد مشخص تلقی شده و در حافظه قرار می گیرد. در این حالت که هر دو کابین زمان یکسانی را برای رساندن شخص به مقصد دارا هستند توسط نرم افزار کابین یک برای سرویس دهی انتخاب می شود. در حالت کلی هر جفت کلید که وارد شد در مکانی از حافظه قرار می گیرد. سپس عملیات محاسبه زمان برای به مقصد رساندن شخص که تقاضای سرویس کرده است توسط هر دو کابین محاسبه می شود. زمان کمتر، انتخاب بهتر را مشخص می کند. سپس دو کلید فشرده شده در میان عدهای قبل که در حافظه کابین مورد نظر وجود داشته جاسازی شده تا کابین پس از رسیدن به این طبقات توقف کرده درب را باز و مسافر را سوار یا پیاده کند. تقاضای بعد نیز به همین روش در حافظه یکی از کابین ها قرار گرفته و در کل سرویس دهی بهینه انجام می شود.

توضیح این نکته ضروری است که صفحه کلید حدوداً هر 20 ms جاروب می شود که بایستی علاوه بر دریافت کلید عمل لرزش گیری نیز روی آن انجام شود. همچنین اگر کلیدی به مدت زیاد فشرده شود و یا در اثر خرابی همیشه در حالت وصل باشد برای جلوگیری از خواندن این کلید یک نسبت تعریف شده است که هنگامی که کلید فشرده شده است این بیت مقدار صفر را خواهد داشت و به محض باز شدن کلید، در جاروب بعدی صفحه کلید مقدار این بیت یک خواهد شد. این بیت نیز هر 20 ms چک می شود. در صورت یک بودن آن برنامه اجرا می شود و اگر صفر بود به منزله فشرده شدن کلید می باشد و در چک کردن بعدی در صورت صفر بودن این بیت، این مفهوم استنباط می شود که کلید هنوز رها نشده است و نرم افزار کلید را در حافظه قرار می دهد. هنگامی که نرم افزار از زیر برنامه SCAN برمی گردد پایه P2.0 که به عنوان یک قفل تعیین شده است بررسی می شود. این پایه تمام شدن وقت 3 ثانیه جابجایی یا 5 ثانیه توقف کابین یک را مشخص می کند. این یک بودن پایه به معنی تمام شدن این وقت است. پس هنگامی که این پایه چک شد و صفر بود مشخص می شود کابین یک برای جابجایی و یا توقف مشغول است و نرم افزار به بررسی کردن کابین دوم می پردازد. برای این کابین نیز پایه مشابهی که P2.1 می باشد تعریف شده است. هنگامی که این پایه ها یک باشند شماره طبقات موجود در هر حافظه که مقصدهای مورد نظر هستند می توانند با محل حضور کابین ها مقایسه شوند. اما هنگامی که در حافظه کابین یک عددی نباشد برنامه به بررسی کابین دو که بررسی پایه P2.1 و تعداد اعداد موجود در حافظه دو می باشد، پرش می کند. ولی اگر عددی یا عدهایی در در حافظه کابین یک موجود باشد طبقه ای که کابین یک در آنجاست با اولین عدد موجود در حافظه کابین یک مقایسه شده در صورت مساوی نودن درب بسته و موتور مربوطه روشن می شود و کابین یک به سمت طبقه مورد نظر حرکت کرده و پس از 3 ثانیه که زمان یک طبقه جابجایی است عدد نمایش داده شده کابین یک، یک واحد اضافه یا کم می شود. به همین ترتیب کابین دو نیز بررسی می شود. برای ایجاد زمان حرکت کابین به یک

طبقه بالا یا پایین تر که 3 ثانیه می باشد از تایمر 0 استفاده شده است. این تایمر یک زمان 20 ms را محاسبه کرده، سپس از رجیسترهایی که برای هر کابین عدد 150 در آن ذخیره شده یک واحد کم کرده و مجدداً به بررسی کردن کابین یک می پردازد. پس از آن دوباره یک تایمر 20 ms گرفته شده و یک داده از رجیستر مربوط که زمان 3 ثانیه را محاسبه می کند، کم می کند. این مرحله تا پایان صفر شدن رجیسترها ادامه می یابد. صفر شدن هر یک از این رجیسترها به معنای پایان وقت 3 ثانیه مربوط به کابین مورد نظر می باشد و کابینی که رجیستر مربوط به آن صفر شده می تواند پس از بررسی کابین و عدد جدید حافظه وضعیت مناسب برای موتور و درب آن کابین ایجاد کند.

## ۶- نتیجه گیری

با پیشرفت نرم افزاری در زمینه میکرو کنترلر ها می توان عملکرد آسانسور ها را بهبود بخشید به طوریکه می توان قابلیت های یک آسانسور را تا حد زیادی افزایش داد. به عنوان مثال با الگوریتم های پیچیده تر زمان دسترسی فرد به طبقات مختلف به صورت جداگانه نمایش داده شده و فرد می تواند طبقات مورد نظر را با در دست داشتن زمان دسترسی به هر طبقه انتخاب کند.

## ۷- مراجع

[1]:[www.elevatorlift.com](http://www.elevatorlift.com)

[2]:[www.cedes.com](http://www.cedes.com)

[3]:[www.fmv.un.ige.ch](http://www.fmv.un.ige.ch)

[4]:[www.archenergy.com](http://www.archenergy.com)

[5]:[www.cs.cna/mers.se](http://www.cs.cna/mers.se)

[6]:[www.deere.com](http://www.deere.com)