

به نام خدا

تلفن گویای نمرات دانشجویان

نویسندگان

محمد نحوی، سمیه طبسی، فریبا محمد زاده

کلمات کلیدی

خط تلفن، ویژوال بیسیک، پورت موازی، دیتابیس اکسس.

چکیده

در این مقاله از طریق پورت موازی و سخت افزار طراحی شده بوسیله نرم افزار نوشته شده به زبان VB، شماره دانشجویی و رمز از کاربر دریافت شده و نمرات مربوط به دروس مختلف خوانده می شوند.



به نام خدا

مقدمه :

عصری که ما در آن زندگی می کنیم عصر فن آوری اطلاعات (IT) است . با توجه به این نکته اطلاع رسانی در سطح جامعه در تمامی زمینه ها اهمیت فوق العاده ای دارد . نکته مائز اهمیت تسریع در امر اطلاع رسانی می باشد ، برای رسیدن به این مهم باید مصار زمان و مکان از میان برداشته شود تا بتوان در هر مکان و در هر زمان به اطلاعات مورد نیاز دسترسی پیدا کرد . اینترنت یکی از ابزار هایست که تا مدودی مصار زمان و مکان را از میان برداشته است . شما در هر مکان یا در هر زمان می توانید اطلاعات مورد نیاز خود را از سایت مورد نظر خود دریافت کنید . ولی مسئله اصلی این است که آیا اینترنت در همه جا و در هر زمان قابل دسترسی است یا خیر ؟ اگر به اطراف خود نگاهی بیندازید متوجه می شوید تلفن به عنوان یکی از لوازم زندگی در همه جا یافت می شود و متی برای استفاده از اینترنت نیز با تلفن سروکار دارید . با توجه به محدودیت های فردی (تمامی افراد جامعه روش استفاده از اینترنت را نمی دانند) و همچنین محدودیت های محیطی (همواره و در همه جا نمی توان به اینترنت دسترسی پیدا کرد) نیاز به ابزاری که قابل استفاده برای تمامی اقشار جامعه باشد و همچنین برای تمامی افراد و در همه جا قابل دسترسی باشد احساس می شود . تلفن گویا با کاربری بسیار ساده و قابل استفاده برای تمامی افراد جامعه و همچنین دسترسی در تمامی نقاط کشور به نظر راه حل مناسبی برای این مشکل می باشد . ما نیز با سافت تلفن گویا سعی کردیم قدمی هر چند کوچک در راه همگانی کردن و تسریع اطلاع رسانی در دانشگاه ، نهیم . امیدواریم این مجموعه گامی موثر برای دانشجویان در پروژه های تلفن و برنامه نویسی تمت ویندوز باشد .

در پایان از زحمات استاد گرامی جناب آقای مهندس بقایی نژاد و جناب آقای دکتر مدادینیا کمال تشکر را داریم .

به امید سرافزازی و پیشتازی کشورمان ایران در تمامی عرصه های علمی و عملی در دنیا .

عملکرد پروژه :

این پروژه به منظور اطلاع از نمرات دروس مختلف دانشجویان از راه دور از طریق پخش سیگنال صوتی بر روی خط تلفن می باشد . برای این منظور یک کامپیوتر از طریق پورت موازی به یک مدار الکترونیکی که متصل به خط تلفن می باشد متصل شده است . این مدار الکترونیکی متشکل از مدار تشخیص زنگ ، مدار اشغال خط و ارسال صوت و مدار دیکر تن می باشد .

برنامه در حالت عادی در حال یک کردن بیت زنگ می باشد ، اگر این بیت از منطق صفر به یک برود به معنی یک بار زنگ خوردن می باشد . پس از ۴ بار زنگ خوردن کامپیوتر خط را اشغال کرده و پس از پخش چند پیام و دریافت شماره دانشجویی و پسورد ، در صورتی که شماره دانشجویی و پسورد را درست وارد کرده باشد ، وارد سیستم شده و می تواند نمرات خود را دریافت کند .

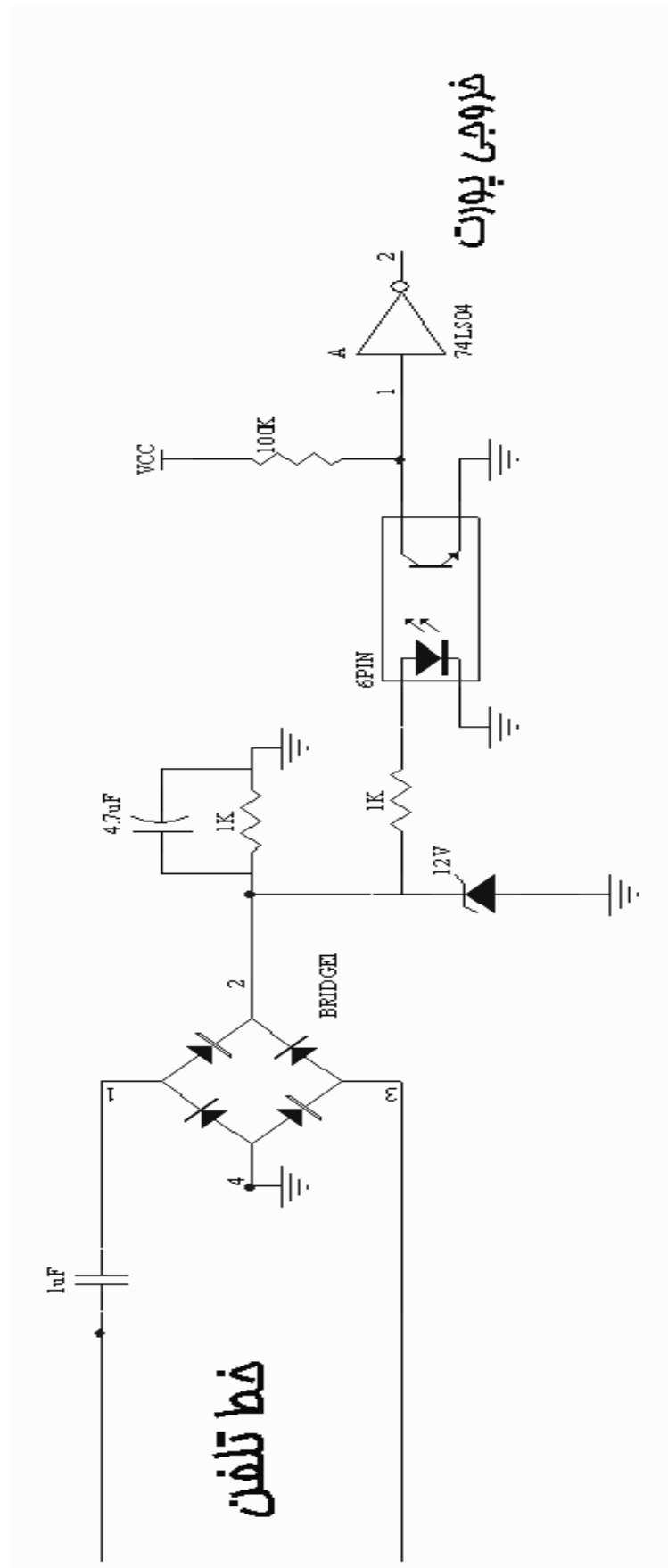
سخت افزار

سخت افزار :

مدار الکترونیکی استفاده شده برای این پروژه شامل سه قسمت اصلی می باشد :

- ۱- مدار تشخیص زنگ .
 - ۲- مدار اشغال خط و پخش صدا .
 - ۳- مدار دیکدر تن .
- در زیر به بررسی هر یک از این مدارات می پردازیم .

تاریخ ۱۳۹۷



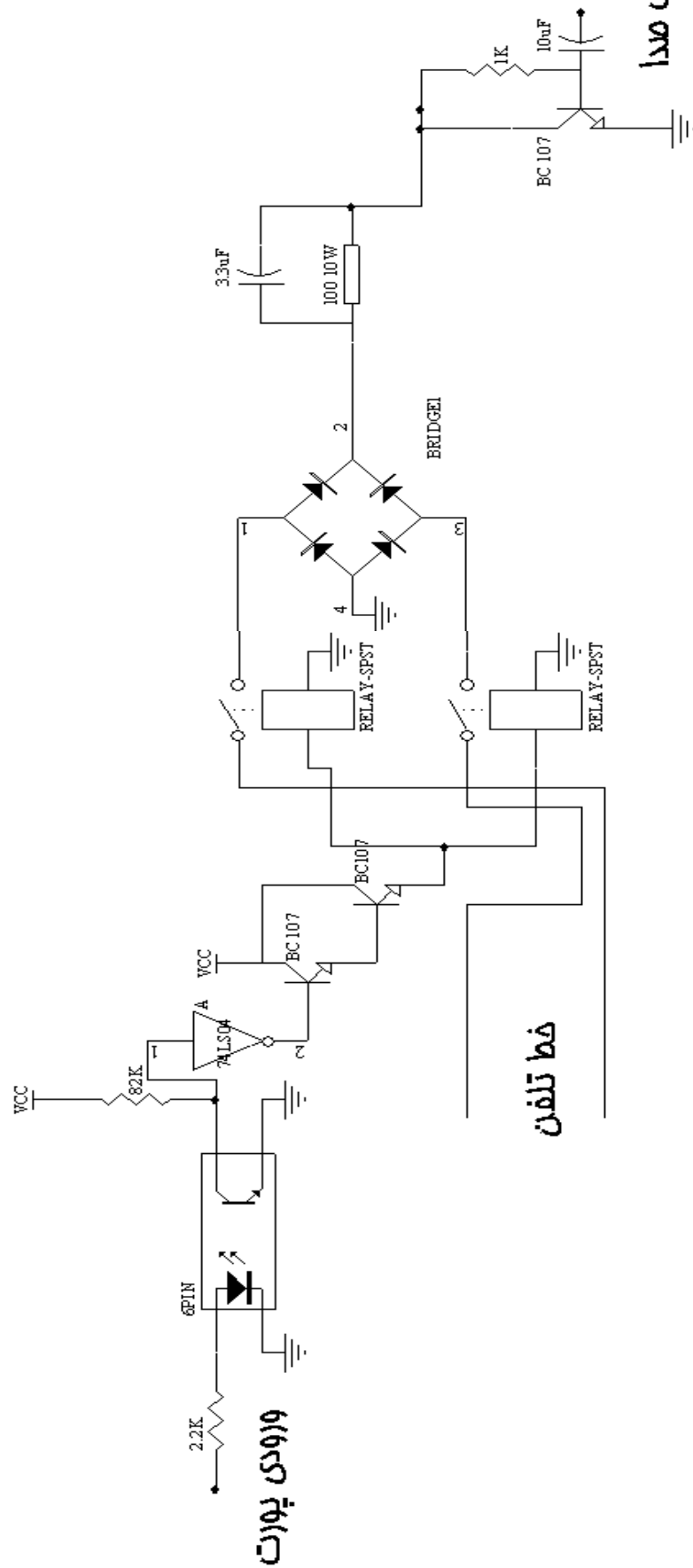
روشهای مختلفی برای تشخیص سیگنال زنگ وجود دارند ، در روش استفاده شده در این مدار در ابتدا توسط یک خازن ولتاژ DC یا همان ولتاژ ثابت ۴۰ ولت حذف می شود سپس با توجه به این نکته که سیگنال AC از خازن عبور می کند با قرار دادن یک پل دیودی بعد از خازن و همچنین یک خازن و مقاومت می توان سیگنال AC زنگ را تبدیل به یک ولتاژ DC کرد . این ولتاژ DC متصل به یک OPTO-CO شده و خروجی اپتوکوپلر NOT شده و به پورت موازی متصل شده است . در حالت عادی مدار به دلیل وجود خازن هیچ ولتاژی را عبور نمی دهد . و خروجی اپتوکوپلر در منطق یک ولت می باشد و پورت از خروجی NOT صفر را می خواند . اگر سیگنال زنگ وارد مدار شود از خازن عبور کرده و پس از عبور از پل دیودی یکسو شده و بوسیله خازن و مقاومت تبدیل به یک ولتاژ DC با ریبیل کم می شود . زمانی که این ولتاژ به ورودی اپتوکوپلر می رسد ترانزیستور درونی آن روشن شده و خروجی آن به منطق صفر می رود ، در نتیجه پورت از خروجی NOT منطق یک را می خواند که معنی سیگنال زنگ می باشد .

چند نکته :

علت استفاده از IC ۷۴۰۴ برای سهولت کار است . می توان با انجام تغییرات کوچکی در نرم افزار این IC را از مدار حذف کرد .

نکته دیگر اینکه چون ولتاژ و جریان قابل تحمل کلکتور - امیتر ترانزیستور درونی OPTO ۴ پایه مقدار کمتری از مقدار مورد نیاز است ، ناچار به استفاده از اپتوکوپلر ۶ پایه هستیم . اپتوکوپلر ۶ پایه با توجه به ولتاژ قابل توجهی که می تواند بر روی کلکتور - امیتر خود تحمل کند ، نیاز مدار مرتفع می کند .

۲- مدار اشغال خط و پخش صدا :



قبل از بحث در مورد مدار اشغال خط لازم است توضیحات مختصری درباره روش تشخیص اشغال شدن خط در مرکز تلفن داده شود .

زمانی که شما گوشی تلفن را برمی دارید (خط را اشغال می کنید) در مرکز تلفن به دلیل مقاومتی که با خط سری شده و جریانی که از خط کشیده می شود اختلاف پتانسیلی در دو سر مقاومت ایجاد شده و مرکز تلفن این ولتاژ را به معنی برداشته شدن گوشی یا اشغال خط می داند و سیگنال بوق ممتد را ارسال می کند . با توجه به این نکته برای اشغال کردن خط تلفن باید ولتاژ دو سر خط از ۴۰ ولت به حدود ۱۲ ولت برسد و جریان حداقل در حدود 40mA از خط تلفن کشیده شود تا خط تلفن اشغال شود .

برای این منظور مدار اشغال خط تلفن متشکل است از یک پل دیودی ، مقاومت خازن سری با خط و یک ترانزیستور که کلکتور آمیتر آن سری با خط تلفن می باشد .

در ابتدای ورود خط تلفن به مدار یک پل دیودی قرار دارد . برای توضیح علت استفاده از پل دیودی باید توجه داشت که همواره باید ولتاژ دو سر کلکتور - آمیتر ترانزیستور استفاده شده برای اشغال خط و انتقال صدا مثبت باشد تا مدار درست کار کند . توجه داشته باشید که اگر جهت دوشاخه ای که وارد پریز تلفن می شود تغییر کند نباید بر عملکرد مدار تاثیر بگذارد . اگر پل دیودی در این مدار وجود نداشته باشد در یک جهت دوشاخه ای که وارد پریز شده ولتاژ کلکتور - آمیتر مثبت شود و در حالت دیگر منفی ، پس در یک حالت مدار خط را اشغال می کند و در حالت دیگر مدار را اشغال نمی کند .

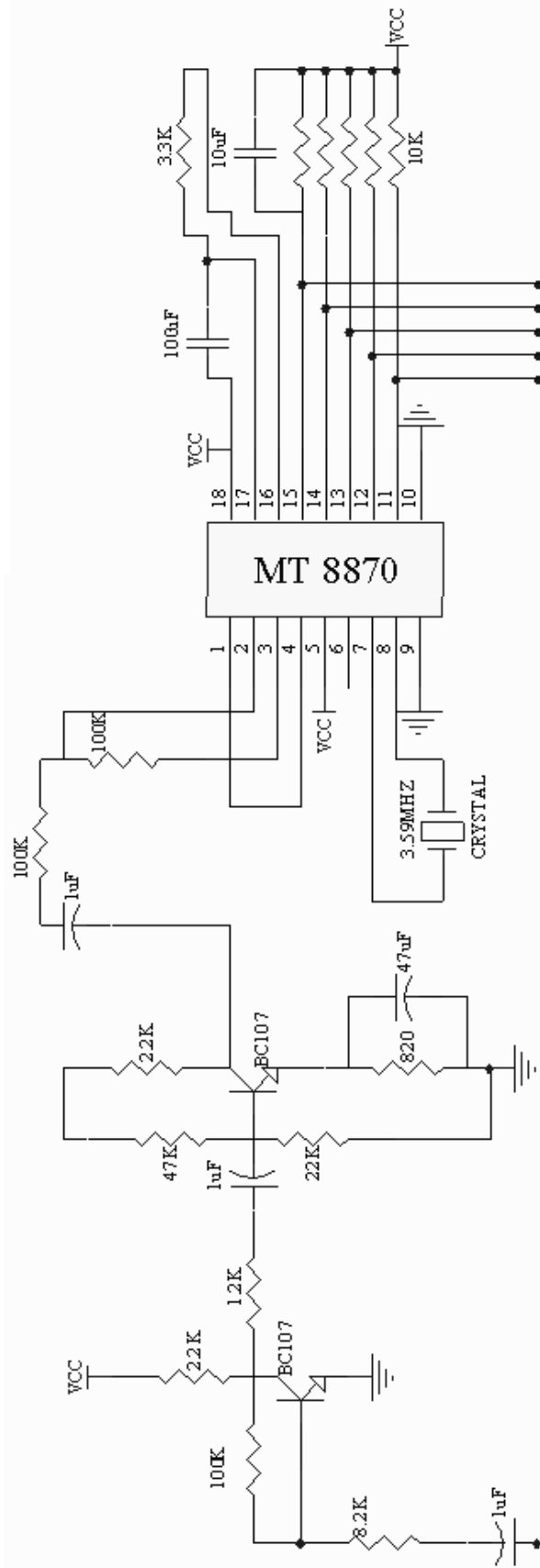
پس از پل دیودی یک مقاومت ۱۰۰ اهم با توان ۱۰ وات و یک خازن ۳،۳ میکرو فاراد قرار دارد . علت وجود مقاومت افت ولتاژ و برای محدود کردن جریان است و خازن نیز برای عبور سیگنال صوتی بدون عبور از مقاومت ۱۰۰ اهم می باشد .

سیگنال صوتی از کارت صدا متصل به بیس ترانزیستور می شود . و از طریق آن سوار بر ولتاژ DC خط می شود .

باید توجه داشت که جریان خروجی کارت صدا در حدود چند میلی آمپر است و نباید از آن بیش از این مقدار جریان کشیده شود ، در غیر این صورت ممکن است کارت صدا خراب شود . نکته دیگری که باید به آن توجه کرد زمین مدار و زمین کامپیوتر است . دقت داشته باشید که خروجی کارت صدا STEREO است به این معنی که دو خروجی دارد و یک زمین ، با توجه به نکته بالا باید سیگنالی را به ورودی مدار متصل کرد که زمین آن با زمین پورت یا همان بدنه کیس یکی باشد ، در غیر این صورت سیگنال صوتی خروجی بسیار ضعیف خواهد بود .

۳- مدار دیکدرتن :

תהיה נחלתך



و خروجی از مدار اشغال خط
و ورودی مدار دیگر تن -

شماره گیری تن به معنای فرستادن چند فرکانس خواص به جای شماره فشار داده شده می باشد .
با توجه به جدول بالا برای هر شماره دو فرکانس با دامنه پیک تا پیک سه ولت فرستاده می شود . آی
سی 8870 MT آی سی دیکدر تن می باشد که ورودی آن به خط تلفن وصل می شود و در خروجی
پس از تشخیص عدد وارد شده ابتدا یک بیت را از منطق صفر به یک می برد و سپس عدد به صورت
BCD در خروجی ظاهر می شود . این عدد توسط پورت خوانده می شود .

نکته قابل توجه این است که برای تشخیص شماره کلید های فشرده شده باید دامنه فرکانس دریافتی در
حدود ۳ ولت باشد ، و با توجه به نکته بالا و اینکه هر چه مسافت طی شده سیگنال طولانی تر باشد
این سیگنال بیشتر تضعیف می شود ، حتما باید قبل از اتصال آن به دیکدر تن سیگنال را تقویت نمود .

توجه :

برای اتصال مدار اشغال خط به خط تلفن از دو رله که ورودی آن ها متصل به دو سر خط تلفن و
خروجی آنها متصل به خط تلفن وارد شده به مدار اشغال خط است ، استفاده شده است . یک بیت برای
اشغال خط از پورت کامپیوتر به ورودی یک ایتوکوپلر متصل شده است . خروجی ایتوکوپلر NOT شده
و به بیس ترانزیستور می رود . امیتر این ترانزیستور نیز به بیس ترانزیستور دیگر رفته است . و امیتر
ترانزیستور دوم به سر مثبت رله متصل شده است .

علت استفاده از رله برای جلوگیری از وارد شدن نویز به کارت صدا و مدار دیکدر تن به هنگام وارد
شدن سیگنال زنگ به مدار می باشد . دامنه سیگنال زنگ در حدود ۱۰۰ ولت پیک تا پیک می باشد . و
اگر حتی یک لحظه نویزی در زمین مدار یا در جای دیگر مدار ایجاد کند ، ممکن است که مدار های
دیگر را دچار مشکل کند . به همین دلیل هم زمین خط تلفن و هم سر دیگر آن توسط رله به مدار
اشغال خط متصل می شود .

مشکلات سخت افزاری :

مدار اشغال خط بالا مداری کاربردی می باشد . ولی این مدار بر روی یکی از خطوط دانشگاه درست کار نمی کرد که علت آن نیز مشخص نشد . همچنین باید توجه داشت آی سی دیکدر تن باید حتما در جهت مثبت خط تلفن قرار گرفته باشد به این معنی که اگر جهت دوشاخه تلفن عوض شود این آی سی کار نخواهد کرد که البته دور از انتظار نیست . زیرا مراکز تلفن نیز همواره در جهت مثبت سیگنال تن را دریافت می کنند .

اگر قصد استفاده از بیت های پورت موازی را دارید ، باید به این نکته توجه داشته باشید که هر بیت پورت موازی حداکثر تا 20MA جریان می دهد و برای جریان کشیدن از پورت با توجه به نیاز باید از مدار دارلینگتون استفاده شود .

نرم افزار

نرم افزار :

ظاهر نرم افزار به صورت زیر می باشد :

The screenshot shows a software window titled "data" with a grey background. It contains several input fields and labels in Persian. The labels are in bold black text, and the input fields are white rectangles with black borders. The labels and their corresponding values are as follows:

پسورد وارد شده	شماره دانشجویی وارد شده
101	8030114001

پسورد	شماره دانشجویی
101	8030114001

میکره	الکترونیک	مدار
10	11	12

با توجه به توضیحاتی که در مورد نرم افزار VB داده شد و به علت استفاده از فایل های صوتی و پایگاه داده در این پروژه و با توجه به اینکه که برنامه نویسی تحت ویندوز به راحتی قابلیت دسترسی و پخش فایل های صوتی و همچنین دسترسی ساده آن به DATA BASE را فراهم می کند ، تصمیم بر آن شد که از زبان برنامه نویسی ویژال بیسیک در این پروژه استفاده شود .

قالب نرم افزار :

نرم افزار متشکل از دو TIMER ، یک DB ، یک میله ابزار مالتی مدیا ، چندین LABEL و چندین TEXT می باشد .

طراحی کلی نرم افزار به این صورت است که یکی از دو تایمر هر ۴ میلی ثانیه فراخوانی شده و بیت مربوط به زنگ را چک می کند پس از ۴ بار دریافت سیگنال زنگ تایمر غیر فعال شده و بیت مربوط به اشغال خط به منطق یک می رود و رویداد MAIN که مهم ترین رویداد است فراخوانی می شود . در این رویداد پس از پخش چند فایل صوتی برنامه شروع به دریافت شماره دانشجویی می کند ، سپس با پخش پیام وارد کردن پسورد ، پسورد را دریافت می کند . توجه داشته باشید که اعداد وارد شده به صورت کارکتر به کارکتر دریافت می شود برای تبدیل این کارکتر ها به عدد مورد نظر وارد شده اعداد در ۱۰ به توان ارزش دهدهی خود ضرب می شود . این عدد به عنوان شماره دانشجویی و پسورد ذخیره می شود . در این مرحله رویداد FIND با جستجو در دیتا بیس شماره دانشجویی مورد نظر را یافته و پسورد آن را با پسورد وارد شده مطابقت می دهد در صورت صحت این دو مقدار متغیر CHECKPASS به منطق TRUE نشانده می شود ، در غیر این صورت مقدار آن FALSE خواهد بود . پس از رویداد FIND اگر بیت مورد نظر TRUE باشد ، نرم افزار پس از پخش پیام های مربوط به کد دروس کد وارد شده را دریافت می کند و نمره مربوط به آن را پخش می کند . این کار تا سه بار تکرار می شود و در پایان پیغام خدا حافظی پخش شده و خط آزاد می شود و دوباره TIMER فعال شده و برنامه به روال عادی خود برمی گردد .

در زیر به توضیح هر یک از رویداد های استفاده شده در نرم افزار می پردازیم :

رویداد TIMER :

این رویداد برای تشخیص زنگ می باشد . به این صورت که هر 4MS فراخوانی می شود و بیت مربوط به زنگ را چک می کند . در صورتی که چهار بار این بیت فعال شود ، TIMER غیر فعال می شود و رویداد MAIN فراخوانی می شود .

رویداد MAIN :

این رویداد اصلی ترین رویداد نرم افزار می باشد . در این رویداد به وسیله میله ابزار مالتی مدیا فایل SALAM.WAV که در شاخه پروژه قرار دارد پخش می شود . در حین پخش شدن پیام رویداد WAIT فراخوانی می شود . پس از پایان یافت پخش پیغام ، پیغام دریافت شماره دانشجویی پخش می شود . توجه داشته باشید پس هر پیغامی رویداد WAIT فراخوانی می شود زیرا برای پخش کامل پیغام ها باید منتظر بود تا تمام پیغام پخش شود ، در غیر اینصورت پیغام ناقص پخش خواهد شد .

پس از پخش پیغام نرم افزار منتظر دریافت شماره دانشجویی می باشد . قبل شروع به دریافت پیغام ساعت سیستم در متغیر Time ذخیره می شود ، اگر تا ۱۵ ثانیه شماره دانشجویی به طور کامل وارد نشود نرم افزار از حالت دریافت شماره خارج شده و خط را آزاد خواهد کرد .

از حلقه WHILE برای تکرار دریافت اعداد وارد شده توسط کاربر استفاده شده است ، به این صورت که متغیر شرط حلقه به ازای هر عدد دریافتی یک واحد اضافه می شود و تا زمانی که این متغیر به عدد ۱۰ نرسیده باشد این حلقه تکرار می شود . در هنگام دریافت شماره دانشجویی مدام پایه پانزدهم آی سی دیکدر تن که مربوط به دریافت عدد می باشد چک می شود ، اگر این بیت یک شود عدد از پورت خوانده می شود ، البته این کار دو بار تکرار می شود تا اثر نویز بر روی مدار کم شود . پس از دریافت عدد تاخیری در حدود 2MS بوسیله تایمر دیگر برای دفع نویز و جلوگیری از دریافت دوباره عدد قبلی قرار داده شده است . برای اینکه این عدد چندین بار خوانده نشود پس از خواند عدد از پورت عدد ۵ در متغیر D1 قرار داده می شود . توجه داشته باشید که در هنگام چک کردن بیت ۱۵ آی سی دیکدر متغیر D1 نیز چک می شود ، اگر این متغیر عددی غیر از صفر باشد شرط برقرار نبوده و عدد از پورت خوانده نمی شود . زمانی که کلید فشرده می شود مدتی شاید در حدود چندین میلی ثانیه بیت ۱۵ آی سی دیکدر تن در منطق یک قرار دارد که ممکن است یک عدد چندین بار خوانده شود برای جلوگیری

از این اتفاق از زمانی که بیت ۱۵ سی دیکدر در منطق یک قرار گرفت تا زمانی که این بیت دوباره به منطق صفر برود پورت فقط یک بار خوانده خواهد شد .

پس از اینکه ۱۰ رقم شماره دانشجویی به طور کامل دریافت شد ، پیغام مربوط به وارد کرد پسورد پخش می شود . و پس از آن دوباره یک حلقه WHILE با مشخصات حلقه بالا وجود دارد که سه رقم برای پسورد را دریافت و ذخیره می کند .

پس از دریافت و ذخیره سازی شماره دانشجویی و پسورد در آرایه های NUMBER و CODE نوبت به تبدیل این ارقام به عدد مورد نظر است .

برای تبدیل این ارقام به عدد هر رقم در ارزش دهی خود ضرب می شود مثلاً عدد ۸۰۳۰۱۱۴۰۰۰ ، رقم ۸ باید در 10^9 ضرب شود . این کار برای آرایه CODE نیز انجام می شود .

پس از انجام عمل تبدیل تابع FIND فراخوانی می شود و اگر شماره دانشجویی و پسورد درست وارد شده باشد متغیر CHECK PASS مقدار TRUE خواهد داشت و نرم افزار پیغام مربوط به کد درسها و همچنین پیغام وارد کردن کد درس مورد نظر را پخش می کند و به مدت سه ثانیه منتظر وارد کردن کد درس می شود . اگر تا سه ثانیه عددی وارد نشد ، نرم افزار دوباره پیغام وارد کردن کد درس مورد نظر را پخش می کند این کار تا ۱۵ ثانیه ادامه خواهد یافت . اگر ظرف ۱۵ ثانیه هیچ عددی دریافت نشد برنامه از حالت دریافت خارج شده و خط را آزاد می کند . اگر عدد وارد شده بین ۱ تا ۳ نباشد نرم افزار به آن رسیدگی نخواهد کرد ، ولی اگر یکی از اعداد ۱ ، ۲ و یا ۳ باشد ، نمره مربوط به آن کد خوانده خواهد شد .

برای پخش کردن نمره دروس باید توجه داشت که نمره ممکن است عدد صحیح و یا اعشاری باشد . برای خواند قسمت صحیح نمره پس از انجام محاسبات کوتاه قسمت اعشاری از قسمت صحیح جدا شده و به صورت جداگانه به ترتیب در gradplay و gradchar ذخیره می شوند . عدد درون متغیر gradplay تبدیل به یک رشته شده و در متغیر Z ذخیره می شود . به همین ترتیب مقدار اعشاری نیز تبدیل به رشته می شود . برای ارسال نمرات به صورت صوت فایل های صوتی با نام اعداد ذخیره می شود . و سپس رشته عدد فایل مربوط به خود را باز می کند و پخش می شود و این کار تا سه بار تکرار می شود .

توجه داشته باشد که اگر نمره مقدار اعشاری نداشته باشد ، پس خوانده شدن مقدار صحیح آن پیغام (" تمام ") پخش می شود .

در پایان رویداد END فراخوانی می شود ، خط آزاد شده و رویداد MAIN به پایان می رسد . پس از آن برنامه به رویداد TIMER باز می گردد ، TIMER فعال شده و نرم افزار به حالت عادی باز می گردد . در زیر به توضیح نحوه عملکرد رویداد های استفاده شده در رویداد MAIN می پردازیم .

رویداد : WAIT

این رویداد به منظور از اطمینان از پخش کامل فایل صوتی می باشد . اگر شما دو فایل صوتی را پشت سر هم و به صورت زیر پخش کنید ، هیچ وقت فایل اول پخش نخواهد شد .

```
MMC.Command = "close"
```

```
MMC.FileName = App.Path & "\salam.wav"
```

```
MMC.Command = "open"
```

```
MMC.Command = "play"
```

```
'-----'
```

```
MMC.Command = "close"
```

```
MMC.FileName = App.Path & "\numberd.wav"
```

```
MMC.Command = "open"
```

```
MMC.Command = "play"
```

برای پخش کامل هر دو فایل باید پس از شروع به پخش فایل منتظر بود تا فایل به پایان آن برسد و سپس فایل بعدی پخش شود .

برای این منظور از خاصیت MODE که جز خواص میله ابزار مالتی مدیا می باشد ، استفاده شده است . تا زمانی که MODE در حالت PLAY باشد نرم افزار در این رویداد می ماند . و پس از آنکه از حالت PLAY خارج شد نرم افزار از رویداد WAIT خارج می شود .

رویداد : DELAY

این رویداد برای ایجاد تاخیر در زمان های مورد نیاز می باشد . به علت سرعت بالای کامپیوتر در اجرای دستورات ممکن است عددی را از پورت می خواند ، عدد درستی نباشد ، برای حل این مشکل این کار دوبار با تاخیر انجام می شود و مقدار اولیه با ثانویه مقایسه می شود ، اگر این دو مقدار برابر باشند عدد خوانده شده از پورت درست است ، در غیر این صورت این کار باید دوباره تکرار شود .

رویداد : FIND

قبل از توضیح رویداد FIND لازم است که توضیحات مختصری درباره نحوه کار با دیتا بیس در این پروژه داده شود. هر دیتا بیس شماره چندین رکورد و چندین فیلد کاری می باشد. به عنوان مثال در پروژه نمره دانشجوی، پسورد و نمرات دروس جز فیلدها (ستون ها) و به هر یک از سطرها که شامل یک شماره دانشجوی، یک پسورد، و نمرات می باشد، رکورد گفته می شود. برای کار با دیتا بیس باید فیلد های آن را به TEXT ها یا LABEL ها نسبت داد تا بتوان از آن استفاده کرد. در این پروژه ۵ TEXT که هر یک از فیلد ها به یکی از آنها نسبت داده شده قرار دارد. برای دریافت داده فیلد ها و رکوردهای مختلف می توان اشاره گر رکورد دیتا بیس را به رکورد مورد نظر برده و سپس دادهای لازم را از TEXT ها خواند.

توضیحات بالا به صورت زیر در این رویداد استفاده شده است.

این رویداد تنها رویدادی است که به طور مستقیم با دیتا بیس سروکار دارد. این رویداد پس از دریافت شماره دانشجویی و پسورد، شروع به جستجو در دیتا بیس می کند به این تربیت که دیتا بیس را به رکورد اول می برد و سپس شماره دانشجویی را از یکی از TEXT ها خوانده و اگر برابر با شماره دانشجویی دریافتی بود پسورد آن را چک می کند، در غیر این صورت اشاره گر رکورد دیتا بیس را یک واحد افزایش داده و این روند تا پایان دیتا بیس ادامه می یابد. برای تشخیص پایان دیتا بیس از خاصیت EOF دیتا بیس استفاده شده است. به این معنی که اگر دیتابیسی به آخرین رکورد خود برسد Data.Recordset.EOF مقدار TRUE خواهد داشت. پس چک کردن شماره دانشجویی و پسورد وارد شده توسط کاربر اگر کد های وارد شده درست باشد مقدار TRUE به متغیر CHECK PASS داده خواهد شد، در غیر این صورت مقدار آن FALSE خواهد بود.

رویداد ENDS :

این رویداد زمانی فراخوانی می شود که نرم افزار به هر دلیلی بخواهد کار خود را به اتمام برساند. در این رویداد ابتدا پیغام خداحافظی پخش شده و سپس خط آزاد می شود.

تابع Lpt_convert :

این تابع پورت موازی با آدرس 0x378 را تبدیل به پورت ورودی می کند، اگر این کار به درستی انجام شود، مقدار True و اگر این تبدیل به درستی انجام نشود مقدار False برگردانده می شود.

نکته مهم :

باید توجه داشت که برای استفاده از متغیرهای در چندین رویداد و تابع باید این متغیرها به صورت عمومی تعریف شود. و همچنین برای استفاده از دستورات ورودی و خروجی پورت های در VB باید DLL آن را به پروژه اضافه نمود. تمامی کارهای بالا در modules به صورت زیر نوشته شود:

```
Declare Sub vbOut Lib "win95io.dll" (ByVal nPort As Integer, ByVal nData As Integer)
Declare Function vbInp Lib "win95io.dll" (ByVal nPort As Integer) As Integer
'-----'
Public Const lpt = &H378
Public Const pi = 3.14159265358979
Public s1(1000) As Double
Public uf As Double
Public numspec As Integer
Public hurmony(1000) As Double
Public rhurmony(500) As Double
Public t As Boolean
Public temp As Integer
Public Function lpt_convert() As Boolean
Dim i As Integer
i = vbInp(lpt + 2 + &H400)
vbOut (lpt + 2 + &H400), i Or 32
i = vbInp(lpt + 2)
vbOut (lpt + 2), i Or 33
End Function
'-----'
```

نام DLL استفاده شده در این نرم افزار WIN95IO می باشد که با همین نام قابل دریافت از اینترنت می باشد. VBINP نام تابعی است که برای خواندن از پورت از آن استفاده می شود و VBOUT نام تابعی است که برای نوشتن در پورت از آن استفاده می شود.

تابع LPT_CONVERT برای تبدیل پورت 0x378 به یک پورت ورودی می باشد. برای انجام این کار باید بیت پنجم بایت با آدرس 0x37A + 0x400 و همچنین بیت اول و پنجم بایت با آدرس 0x37A را در منطق یک نشانند.

توجه:

اگر در هنگام تبدیل پورت موازی به پورت ورودی پورت به جایی متصل باشد، این تبدیل به درستی انجام نخواهد شد و تابع LPT_CONVERT مقدار False را بر می گرداند. برای حل این مشکل باید یک

بار هنگامی که پورت متصل به مدار است بیت های آن به دور دستی خواند و در هنگام تست پورت در تابع LPT_CONVERT برای صحت ورودی شدن پورت به جای عدد ۲۵۵ عدد بدست آمده از مدار را ذخیره کرد (البته می توان در زمان اجرای برنامه که این تابع تبدیل اجرا می شود پورت را از مدار جدا کرد).

نرم افزار نوشته شده به زبان VB :

```
Public Sub main()  
Static i As Integer  
Static j As Integer  
Static d1 As Byte  
'----- enter  
MMC.Command = "close"  
  
MMC.FileName = App.Path & "\salam.wav"  
  
MMC.Command = "open"  
  
MMC.Command = "play"  
  
wait  
'----- read masage  
MMC.Command = "close"  
  
MMC.FileName = App.Path & "\numberd.wav"  
  
MMC.Command = "open"  
  
MMC.Command = "play"  
  
wait  
'----- for recieve number  
j = 1  
d1 = 0  
time = Timer  
Text1.Text = time  
While j <= 10      '---If j <= 10 Then  
i = vbInp(lpt) And 32  
If i = 32 And (d1 = 0) Then  
    delay  
    If dlynumber Then  
        dlynumber = False  
        Timer2.Enabled = True  
        i = vbInp(lpt) And 32  
        If i = 32 And (d1 = 0) Then  
            number(j) = vbInp(lpt) And 15  
            Text7.Text = number(j)  
            If (number(j) = 10) Then  
                number(j) = 0  
            End If  
            j = j + 1  
            d1 = 5  
        End If  
    End If  
End While
```

```

End If
Else:
    If Timer - time >= 15 Then
        GoTo l3
    End If
End If

i = vbInp(lpt) And 32
If i = 0 Then
    delay
    i = vbInp(lpt) And 32
    If i = 0 Then
        d1 = 0
    End If
End If
Wend
'-----for read message
MMC.Command = "close"

MMC.FileName = App.Path & "\pass.wav"

MMC.Command = "open"

MMC.Command = "play"

wait
'-----for recieve code
time = Timer
j = 1
d1 = 0
While j <= 3 '--If j <= 3 Then
    i = vbInp(lpt) And 32
    If i = 32 And (d1 = 0) Then
        delay
        If dlynumber Then
            dlynumber = False
            Timer2.Enabled = True
            i = vbInp(lpt) And 32
            If i = 32 And (d1 = 0) Then
                code(j) = vbInp(lpt) And 15
                Text8.Text = code(j)
                If (code(j) = 10) Then
                    code(j) = 0
                End If
                d1 = 5
                j = j + 1
            End If
        End If
    End If
End If

```

Else:

 If Timer - time >= 15 Then

 GoTo l3

 End If

End If

i = vbInp(lpt) And 32

If i = 0 Then

 delay

i = vbInp(lpt) And 32

 If i = 0 Then

 d1 = 0

 End If

End If

Wend

‘-----end of read

‘-----for convert num

Dim x As Double

Dim y As Integer

x = 1

ncode = 0

ccode = 0

y = 9

For i = 1 To 10

For j = y To 1 Step -1

x = x * 10

Next j

ncode = ncode + number(i) * x

x = 1

y = y - 1

Next i

Text7.Text = ncode

‘-----for convert code

x = 1

y = 2

For i = 1 To 3

For j = y To 1 Step -1

x = x * 10

Next j

ccode = ccode + code(i) * x

x = 1

y = y - 1

Next i

Text8.Text = ccode

‘-----open data base


```

find
If checkpass Then
checkpass = False
'-----

MMC.Command = "close"

MMC.FileName = App.Path & "\number1.wav"

MMC.Command = "open"

MMC.Command = "play"

wait
'-----

MMC.Command = "close"

MMC.FileName = App.Path & "\code1.wav"

MMC.Command = "open"

MMC.Command = "play"

wait
'-----get num of grade
Dim check As Byte
Dim d2 As Byte
Dim time1 As Double
check = 1
l2:

d2 = 0
j = 1
time = Timer
time1 = Timer
l1:

i = vbInp(lpt) And 32
If (i = 32) And (d2 = 0) Then
    delay
    i = vbInp(lpt) And 32
    If (i = 32) And (d2 = 0) Then
        d2 = 5
        grade = vbInp(lpt) And &HF
        GoTo l5
    Else:

    If Timer - time >= 3 Then
        MMC.Command = "close"

        MMC.FileName = App.Path & "\number1.wav"

        MMC.Command = "open"

```

```

MMC.Command = "play"
wait
time = Timer
End If
If Timer - time1 >= 15 Then
GoTo l3
End If
End If
End If

i = vbInp(lpt) And 32
If (i = 0) Then
delay
i = vbInp(lpt) And 32
If (i = 0) Then
d2 = 0
End If
End If
GoTo l1
'-----read from file grade

'-----after resive numreh
l5:
Dim gradplay As Double
Dim gradchar As Byte
Dim gradfloat As Double
Dim z As String
'-----
If (grade = 1) Then
gradplay = Text4.Text
End If
If (grade = 2) Then
gradplay = Text5.Text
End If
If (grade = 3) Then
gradplay = Text6.Text
End If
If (grade > 4) Then
GoTo l3
End If
'-----
gradchar = gradplay
If gradchar > gradplay Then
gredchar = gradchar - 1
End If
gradplay = gradplay - gradchar
z = gradchar

```

```

-----'
MMC.Command = "close"
MMC.FileName = App.Path & "\" + z + ".wav"
MMC.Command = "open"
MMC.Command = "play"
wait
'----- va
If (gradplay <> 0) Then
MMC.Command = "close"
MMC.FileName = App.Path & "\" + va + ".wav"
MMC.Command = "open"
MMC.Command = "play"
wait
End If
'----- mantissa
z = gradplay
If z = 0 Then
z = "tamam"
MMC.Command = "close"
MMC.FileName = App.Path & "\" + z + ".wav"
MMC.Command = "open"
MMC.Command = "play"
wait
Else:
MMC.Command = "close"
MMC.FileName = App.Path & "\" + z + ".wav"
MMC.Command = "open"
MMC.Command = "play"
wait
End If
'-----
If check >= 3 Then
GoTo 13
Else:

```

```

check = check + 1
'-----
MMC.Command = "close"

MMC.FileName = App.Path & "\number1.wav"

MMC.Command = "open"

MMC.Command = "play"

wait
'-----
GoTo l2
End If
l3:

ring = 0
ends
Exit Sub
End If
ends
End Sub

Private Sub form_load()
Dim i As Byte
'-----
checkpass = False
checkring = False
dlynumber = True
ring = 0
d = 0
If Not lpt_convert Then
MsgBox "can not convert lpt port in address(0x378)to vbinput mode." & vbCrLf & vbCrLf &
"convert bios configuration to epp mode.", 16, "error"
End If
i = vbInp(lpt + 2(
vbOut lpt + 2, i Or 2
End Sub
Public Sub ends()

MMC.Command = "close"

MMC.FileName = App.Path & "\bay.wav"

MMC.Command = "open"

MMC.Command = "play"

Wait

```

```

'-----
i = vbInp(lpt + 2)
vbOut lpt + 2, i Or 1
ring = 0
End Sub

Private Sub Timer1_Timer()
If (vbInp(lpt) >= 128) And (d = 0) Then
d = 5
End If
i = vbInp(lpt)
If ((i And 128) = 0) Then
Text1.Text = "noring"

d = 0
End If
'-----

If (ring <= 4) = True Then
i = vbInp(lpt + 2)
vbOut lpt + 2, i Or 1
Else:

d = 0
i = vbInp(lpt + 2)
vbOut lpt + 2, i And 254
ring = 0
Timer1.Enabled = False
main
Timer1.Enabled = True
'-----

End If
End Sub
Public Sub wait()
l6:

If MMC.Mode = mciModePlay Then
GoTo l6
End If
End Sub
Public Sub find()
Dim g As String
Dim gsh As String
Dim gc As String
Data1.Recordset.MoveFirst
gsh = ncode
gc = ccode
l1:

g = Text2.Text
If g = gsh Then
g = Text3.Text

```

```
If g = gc Then  
    checkpass = True  
    Text1.Text = "pass ok"
```

```
Else:
```

```
    Text1.Text = "error pass"
```

```
    Exit Sub
```

```
End If
```

```
Else:
```

```
If Data1.Recordset.EOF = True Then
```

```
    -----'message not number student
```

```
Exit Sub
```

```
End If
```

```
Data1.Recordset.MoveNext
```

```
GoTo 11
```

```
End If
```

```
End Sub
```

```
Private Sub delay()
```

```
    Dim g As Integer
```

```
    Dim g1 As Integer
```

```
    For g = 0 To 100
```

```
        For g1 = 0 To 10
```

```
            Next g1
```

```
        Next g
```

```
    End Sub
```

```
Private Sub Timer2_Timer()
```

```
    Timer2.Enabled = False
```

```
    dlynumber = True
```

```
End Sub
```