



دانشگاه آزاد اسلامی واحد رودان

جزوء درس :

کاربرد رایانه در برق (کاردانی)

نام استاد: عبدالرضا صفایی

(www.rzasafaei.ir)

رشته: برق

دی ماه ۱۳۹۵

معرفی میکروکنترلرهای AVR

۱-۱ میکروکنترلر چیست؟

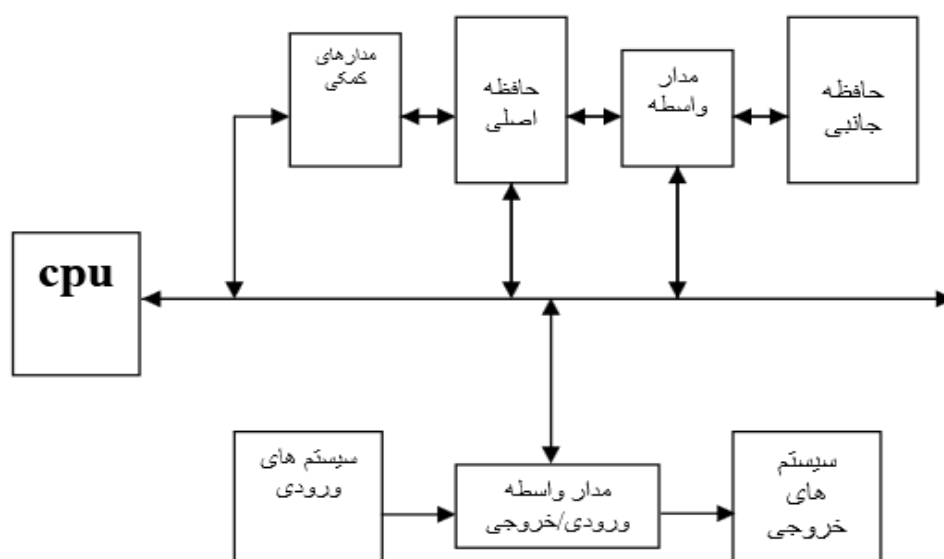
برای درک بهتر مفهوم میکروکنترلر بهتر است ابتدا روندی که منجر به تولید این قطعات گردید بررسی گردد. کامپیوترها امروزه نقشی غیر قابل انکار در زندگی بشر دارند. گسترش کاربرد کامپیوتر در علوم مختلف در چند دهه اخیر به حدی زیاد بوده که امروزه تصور زندگی بدون آن بسیار سخت است. کامپیوترها با تمام پیچیدگیهایشان، همگی بلوک دیاگرام ساده ای مانند شکل ۱-۱ دارند.



شکل ۱-۱ بلوک دیاگرام بسیار ساده از یک کامپیوتر

سه بلوک اصلی نشان داده شده در این شکل، واحد مرکزی پردازش (CPU)، حافظه و ادوات ورودی-خروجی (I/O) می باشند. واحد مرکزی پردازش در واقع واحد اصلی پردازشگر در هر سیستم کامپیوتری می باشد که از واحدهای متعددی مانند ALU، واحد کنترل، رجیستر و دکودر دستورالعمل و مدارات تولید پالس داخلی تشکیل شده است. در آغاز روند تولید کامپیوترها واحدهای مختلف CPU به صورت ماژولهای جداگانه طراحی می شد و از کنار هم قرار گرفتن آنها و ارتباط واحدها با یکدیگر CPU تشکیل می گردید. در اوایل دهه ۱۹۷۰ این فکر در طراحان سیستمهای کامپیوتری تقویت گردید که آیا می توان تمامی واحدهای مورد نیاز برای تشکیل یک CPU را در داخل یک تراشه (IC) قرارداد؟ تلاشهای محققان در سال ۱۹۷۱ نتیجه داد و شرکت اینتل موفق شد یک CPU با نام intel4004 تولید کند. این IC شامل یک CPU کامل بود که تماماً بر روی یک تراشه قرار داده شده بود. این واحد مرکزی پردازش جدید را میکروپروسسور یا ریزپردازنده نامگذاری کردند.

تحولی که معرفی میکروپروسورها بوجود آوردند کمتر از تحولی که ساختن کامپیوترها ایجاد نمود، نبود. استقبال فراوان به عمل آمده از میکروپروسورها، کارخانه های بسیاری را نیز درگیر طراحی و معرفی انواع آنها نمود که هر یک در نحوه ساخت، ساختمان داخلی و نحوه ارتباط با حافظه و ورودی-خروجی با یکدیگر تفاوتی داشتند. از ریزپردازنده هاییکه در دهه ۱۹۷۰ تولید گردید می توان به ریزپردازنده قدرتمند Z80 ساخت شرکت زایلوگ اشاره کرد. حال با تولید میکروپروسورها، سیستمهای کامپیوتری کاملاً مبتنی بر میکروپروسورها شده و پیشرفت سریع علوم مرتبط با آنها آغازگردید. در یک سیستم کامپیوتری مبتنی بر میکروپروسورها بخش های مختلفی وجود دارند که هر یک وظیفه خاصی را در داخل سیستم بر عهده دارند. بلوک دیاگرام شکل ۲-۱ چنین سیستمی را نشان می دهد. اجزا نشان داده شده در این شکل، اجزا معمول تشکیل دهنده یک سیستم میکروپروسوری می باشند.



توضیح مختصری درباره واحدهای مختلف یک سیستم میکروپروسوری می تواند اهمیت این واحدها را نمایان سازد. حافظه اصلی از دو مجموعه تشکیل می گردد. حافظه های ROM و حافظه های RAM که هر یک در سیستم میکروپروسوری وظیفه خاصی برعهدهدارند. حافظه ROM همانطور که از نامش پیداست

حافظه ای تنها خواندنی است. علت استفاده از این نوع حافظه در سیستم ها، غیر فرار بودن اطلاعات آن است. به این ترتیب داده ای که در یک چنین حافظه ای قرار گرفته باشد، حتی پس از قطع تغذیه سیستم نیز دست نخورده باقی می ماند. بنابراین اینگونه حافظه ها کاندیدای مناسبی برای ذخیره سازی اطلاعات دائمی در سیستم هستند. در یک سیستم میکروپروسسوری عملکرد سیستم توسط برنامه ای که در حافظه ذخیره شده و به ترتیب از خانه های متوالی حافظه توسط CPU خوانده و اجرا می شود، تعیین می گردد. تمامی یا بخشی از این برنامه در هر سیستم میکروپروسسوری در داخل ROM قرار می گیرد. در هر سیستم میکروپروسسوری در ضمن اجرای برنامه بنا به دلایل مختلف نیاز به ذخیره سازی اطلاعات بصورت داده ها و یا حتی برنامه های موقت می باشد، لذا استفاده از حافظه قابل نوشتن در سیستم میکروپروسسوری ضروری است. برای این منظور از حافظه های RAM استفاده می گردد. برای ساخت حافظه های RAM و ROM از تکنولوژی های مختلف با مزایا و معایب خاص خود استفاده می گردد که از جمله آنها می توان به EPROM، EEPROM، PROM برای ساخت ROM و تکنولوژی استاتیکی و دینامیکی برای ساخت RAM اشاره کرد.

هر سیستم میکروپروسسوری محلی برای ذخیره دائمی اطلاعات نرم افزار سیستم احتیاج دارد. این واحد حافظه علاوه بر اینکه باید قابلیت ذخیره اطلاعات را داشته باشد، باید قابلیت بازنویسی و تغییر اطلاعات و نرم افزار سیستم را نیز داشته باشد. برای این منظور از حافظه جانبی استفاده می گردد. در سیستمهای میکروپروسسوری قدیمی تر برای ساخت حافظه های جانبی از ادوات نیمه هادی استفاده نمی شد و به جای آنها از وسایل دارای قسمتهای مکانیکی استفاده می گردید که سبب بروز مشکلاتی مانند سرعت پایین، حجم ذخیره سازی کم و می گردید. امروزه با پیشرفت تکنولوژی ساخت ادوات نیمه هادی از تراشه های نیمه هادی برای حافظه جانبی استفاده می گردد. در اینجا اشاره به این نکته نیز لازم است که به علت سرعت بالای میکروپروسسور و سرعت پایین حافظه ها، ارتباط بین حافظه جانبی و حافظه اصلی با سایر قسمت ها در سیستم میکروپروسسوری از طریق یک مدار واسط انجام می گیرد.

دستگاههای ورودی-خروجی (I/O) قسمت دیگری از یک سیستم میکروپروسسوری هستند. هر سیستم میکروپروسسوری زمانی می تواند کامل باشد که امکان ارتباط با جهان خارج برایش فراهم شده باشد. میکروپروسسورها اطلاعات را توسط دستگاههای ورودی-خروجی دریافت کرده و پس از پردازش اطلاعات و حصول نتیجه مورد نظر، نتیجه پردازشها را بر روی دستگاههای خروجی ارسال می کنند. تجهیزات بسیاری به عنوان دستگاه های ورودی می توان به کارگرفت مانند صفحه کلید، انواع سنسورها، مبدل آنالوگ به دیجیتال. از مهمترین دستگاههای خروجی نیز می توان به نمایشگر، انواع موتورها و چاپگرها اشاره کرد. در مورد دستگاههای ورودی-خروجی دو نکته قابل توجه است. نکته اول آن است که اغلب دستگاههای ورودی داده هایی به صورت دیجیتال ندارند و قبل از اینکه داده های یک دستگاه ورودی وارد میکروپروسسور شود باید ابتدا به فرمت دیجیتال تبدیل گردد. اغلب سنسورهایی که در سیستمهای میکروپروسسوری به کار برده می شود از این دسته دستگاههای ورودی هستند. این مورد در رابطه با دستگاههای خروجی نیز صدق می کند. نکته دوم هم آن است که دستگاههای ورودی-خروجی نرخ انتقال داده نسبتاً پائینی دارند و ارتباط مستقیم این دستگاهها با میکروپروسسور تقریباً غیر ممکن است. به همین دلیل برای اتصال دستگاههای ورودی-خروجی به میکروپروسسور نیاز به مدارهای واسطه می باشد که در بلوک دیاگرام ۱-۲ آورده شده است.

یک سیستم میکروپروسسوری علاوه بر قسمتهای مختلفی که در بلوک دیاگرام شکل ۱-۲ آورده شده است، نیاز به قسمتهای دیگری نیز دارد. گذرگاه آدرس و گذرگاه داده جز قسمتهای مهم هر سیستم میکروپروسسوری هستند و هنگام طراحی سیستم باید برای آنها مدارات جداگانه ای توسط تراشه ها ایجاد کرد. از آنجا که میکروپروسسور دارای مدارات ترتیبی هستند، برای عملکردشان احتیاج به مدارات تولید پالس ساعت دارند که این مدارات هم به صورت جداگانه توسط قطعات آنالوگی مانند خازن و کریستال باید ایجاد شود. باید به این نکته توجه کرد که قسمتهای نام برده شده در حقیقت کمترین تجهیزاتی است که هر سیستم میکروپروسسوری

باید داشته باشد و در صورت نیاز وابسته به طراحی سیستم مورد نظر، تجهیزات دیگری مانند شمارنده ها، مبدل های آنالوگ به دیجیتال و نیز می توان برای سیستم میکروپروسسوری در نظر گرفت.

با پیشرفت روزافزون علوم مرتبط با میکروپروسورها و سیستمهای مبتنی بر آنها، این سیستمها وارد عرصه های مختلف زندگی بشر گردیدند و کنترل بسیاری از امور بر عهده سیستمهای میکروپروسسوری قرار گرفت. وجود بخشهای مختلف در یک سیستم میکروپروسسوری و طراحی جداگانه هر یک از آنها و اتصال این بخشها به یکدیگر سبب گردید تا حجم سیستمهای میکروپروسسوری زیاد شده و بعضاً برای کاربردهای خاصی که نیاز به یک سیستم کنترلی کوچک می بود، استفاده از این سیستمهای میکروپروسسوری مشکل ساز می گردید. وجود این مشکلات سبب شد تا طراحان به فشرده سازی سیستمهای میکروپروسسوری بیاورند و این نکته که آیا می توان یک سیستم کامل میکروپروسسوری را داخل یک تراشه طراحی کرد؟

در سال ۱۹۷۶ اولین تراشه که حاوی یک سیستم کامل میکروپروسسوری در داخل خود بود تولید شد. این تراشه با نام intel8748 شامل ۱۷۰۰۰ ترانزیستور در یک مدار مجتمع بود و دارای یک CPU، ۱ کیلوبایت EPROM، ۶۴ بیت RAM، ۲۷ پایه I/O و یک تایمر ۸ بیتی بود. چنین تراشه ای که یک سیستم کامل میکروپروسسوری را به همراه کلیه اجزای آن در داخل خود داشته باشد، میکروکنترلر نامیده می شود. در شکل ۳-۱ بلوک دیاگرام داخلی یک میکروکنترلر ساده آورده شده است.

به کارگیری گسترده میکروکنترلرها در کنترل فرایندهای ساده و صنعتی سبب شد تا شرکت های بسیاری در دنیا به تولید این تراشه ها اقدام کنند که می توان به میکروکنترلرهای ۸۰۵۱ ساخت اینتل، EZ8 ساخت زایلوگ و PIC ساخت میکروچیپ اشاره کرد. یکی از جدیدترین و قویترین میکروکنترلرهای که به بازار الکترونیک عرضه شده و امروزه در بسیاری از کاربردهای صنعتی، رباتیک، کنترلی کاربرد دارند، میکروکنترلرهای AVR ساخت شرکت معتبر Atmel می باشد. این میکروکنترلرها برای اولین بار در سال ۱۹۹۶ عرضه شدند.

تنوع بسیار زیاد این میکروکنترلرهای AVR و قابلیت‌های متفاوت هر یک از آنها سبب علاقه مندی طراحان به این میکروکنترلرها گردیده است. سادگی، امکانات گسترده، قیمت ارزان، مصرف توان کم، زبانهای برنامه نویسی متعدد و سطح بالا، فناوری حافظه پیشرفته و تواناییهای دیگر میکروکنترلرهای خانواده AVR، آنها را به یکی از پرکاربردترین میکروکنترلرها تبدیل کرده است. از جمله کاربردهای AVR ها می توان به کاربردهای خودروهای موتوری، کنترل صنعتی، سروورهای شبکه، حسگر، تلفنهای RF، روباتیک، اسباب بازیها و اشاره کرد.

میکروکنترلر

میکرو در حقیقت یک کامپیوتر تک تراشه ارزانیست است. تک تراشه به این معناست که تمامی اجزا کامپیوتر از جمله حافظه، درگاههای ورودی / خروجی، واحد پردازشگر مرکزی و سایر ملزومات آن درون یک تراشه کوچک جای گرفته اند.

مهمترین ویژگیهای میکروکنترلرهای مذکور به صورت زیر است:

۱- کارایی بالا و توان مصرفی کم

۲- ۳۲ رجیستر (ثبات) ۸ بیتی

۳- ۳۲ کیلو بایت حافظه FLASH داخلی قابل برنامه ریزی با قابلیت ده هزار بار نوشتن و پاک کردن

۴- ۲ کیلو بایت حافظه داخلی SRAM

۵- دو تایمر/شمارنده هشت بیتی

۶- چهار کانال PWM

۷- هشت کانال مبدل A/D ده بیتی

۸- یک مقایسه کننده آنالوگ داخلی

۹- ارتباط سریال برای برنامه ریزی: ISP

۱۰- USART سریال قابل برنامه ریزی

۱۱- دارای شش حالت SLEEP

۱۲- منابع وقفه داخلی و خارجی

۱۳- اسیلاتور داخلی RC

۱۴- کار با ولتاژ ۴,۵ تا ۵,۵

۱۵- فرکانس کاری ۰ تا ۱۶ مگاهرتز

اجزای داخلی میکروکنترلر

واحد پردازشگر مرکزی:

این واحد قلب میکرو است و نقش پردازش دستورالعملها را بر عهده دارد.

حافظه

انواع حافظه های موجود در میکرو شامل FLASH- RAM- EEPROM و رجیسترها میباشد.

حافظه FLASH

این نوع حافظه برای ذخیره برنامه اجرایی میکرو مورد استفاده دارد. این حافظه قابلیت هزاران بار پاک شدن و مجددا برنامه ریزی شدن را دارند. این بدان معناست که شما بارها و بارها میتوانید برنامه میکرو را تغییر دهید.

حافظه RAM

این حافظه برای ذخیره داده هاست. متغیر هایی که در طول برنامه تعریف میشوند در آن جای دارند. این نوع حافظه فرار بوده و با قطع برق تمامی اطلاعات آن پاک میشوند.

حافظه EEPROM

این حافظه برای ذخیره برخی مقادیر برنامه که پس از قطع برق نیز باقی بمانند استفاده میشود.

رجیسترها

رجیسترها حافظه های کوچکی هستند که درون خود پردازنده پیاده سازی شده اند. از آنها برای ذخیره سازی موقت داده ها و همچنین ذخیره برخی داده های کنترلی استفاده میشود.

واحد ورودی - خروجی Input-Output

این واحد واسط ارتباط میکرو با محیط اطراف است. تعداد پایه های ورودی - خروجی می تواند از ۳ تا ۹۰ پایه باشد. با برنامه ریزی های خاص میتوان از هر یک از پایه ها بصورت ورودی یا خروجی استفاده کرد. علاوه براین برخی میکرو ها قابلیت ورودی آنالوگ را دارا میباشند که برای اندازه گیری مقادیر آنالوگ حاصل از حسگر ها مورد استفاده دارد.

اسیلاتور

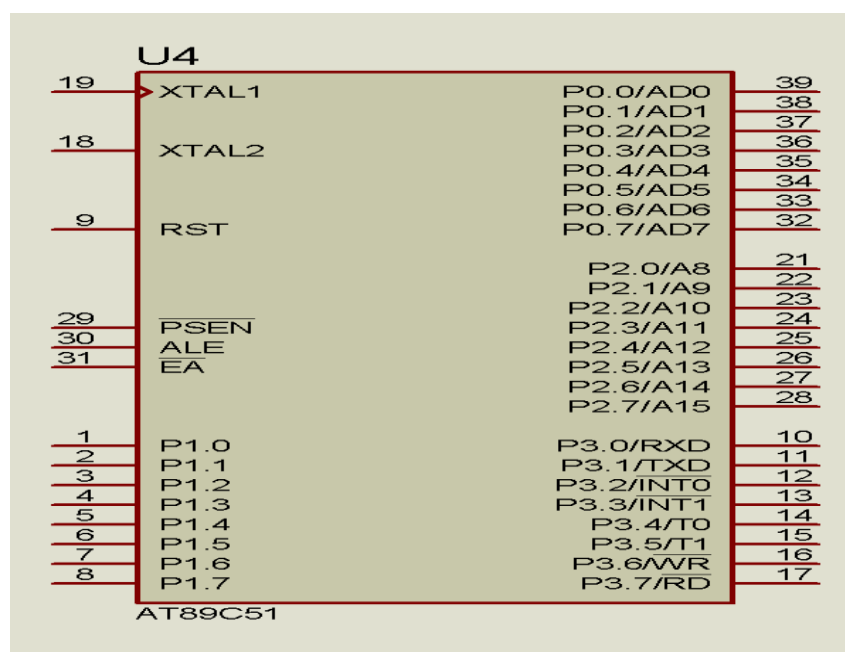
اصولا هر پردازنده ای برای فعالیت و همزمانی میان اجزا خود به یک عامل هماهنگ کننده نیاز دارد. در میکرو ها این عامل یک موج مربعی است که پالس ساعت نامیده میشود که نقش هماهنگی میان تمام اجزا و تعیین سرعت پردازش را دارد. اغلب میکرو ها امروزه یک اسیلاتور داخلی برای تولید پالس دارند و همچنین قابلیت اتصال به یک اسیلاتور خارجی را دارا میباشند.

تایمر - شمارنده

میکروها دارای تایمر های ۸ و ۱۶ بیتی هستند که به ترتیب میتوانند تا ۲۵۵ و ۶۵۵۳۵ بشمارند. علاوه بر این تایمرها میتوانند به عنوان شمارنده پیکر بندی شوند و برای شمارش دفعات وقوع یک پدیده خاص بکار روند.

پایه ها و مدار داخلی ۸۰۵۱

پایه های میکروکنترلر ۸۰۵۱ در شکل (۲-۲) نشان داده شده اند



شکل ۲-۲: پایه های ۸۰۵۱

همانگونه که در شکل بالا مشاهده میکنید یک میکروی ۴۰ پایه است که از نظر ساختار میکرو با اکثر میکروهای سری ۸۰۵۱ که ۴۰ پایه دارند ساختمان مشابهی دارد.

پورت صفرویک و دو و سه ۳۲ پایه از میکروی ۴۰ پایه را تشکیل میدهند. پورت صفر که اصطلاحاً درین باز گفته میشود اگر بخواهیم از آن به عنوان ورودی-خروجی استفاده کنیم باید از مقاومت های بالاکش استفاده کنیم. پورت های یک و دو و سه در حالت عادی میتوانند به عنوان ورودی و خروجی قرار گیرند.

تغذیه آی سی که باید به ولتاژ ۵+ وصل شود به دو صورت انجام میشود:

۱- اگر تغذیه ۸۰۵۱ را بخواهید با منبع DC انجام دهیم از یک آی سی رگولاتور استفاده میکنیم. به عنوان مثال رگولاتور ۷۸۰۵ ولتاژ ۹ ولت DC را به ۵ ولت تبدیل میکند.

۲- اگر تغذیه با منبع AC انجام شود در ابتدا با استفاده از یک ترانسفورماتور ولتاژ برق شهر را به یک سطح پایین تر میرسانیم و سپس با استفاده از مبدل یکسوساز دیودی پل AC به DC تبدیل میشود و سپس با استفاده از یک آی سی رگولاتور سطح ولتاژ ۵ ولت تولید میشود.

GND: باید به صفر ولت یا زمین مدار وصل شود.

RST: این پایه که برای ریست کردن و راه اندازی مجدد میکرو است که باید به صفر متصل باشد مگر زمانی که میخواهیم میکرو را ریست کنیم. این پایه باعث میشود میکرو برنامه خود را از خط اول اجرا کند.

۱، ۲ XTAL که باید به کریستال متصل شوند. کریستال قطعه ای است که برای تولید کلاک مورد نیاز برای میکرو به کار میرود. باید دو سر آنرا با دو خازن ۳۳ پیکوفاراد به زمین متصل کرد. کریستال میتواند فرکانسهای متفاوتی ایجاد کند که معمولاً برای میکرو کنترلر ۸۰۵۱ از کریستال با فرکانس ۱۱ مگاهرتز استفاده میشود توجه شود که میکرو برای انجام هر دستور یک زمانی نیاز دارد که این زمان برای هر دستور ممکن است یک یا دو یاحتی سه سیکل باشد.

منطق جاروب و جدول گلایف:

در ساخت نمایشگرهای تبلیغاتی با توجه به موارد استفاده و مکانی که باید نصب شوند نوع نمایشگر متفاوت است. یعنی تعداد led های استفاده شده، رنگ و نوع آنها و حتی فاصله بین آنها در نمایشگر، فرق میکند به عنوان مثال، برای نمایش ساعت در اتوبوس ها از یک نمایشگر ۸×۲۴ با led های مات استفاده میشود؛ در حالی

که برای تبلیغات یک مغازه، یک نمایشگر 32×20 با led های پور نور (high light/bright) مورد استفاده قرار میگیرد. مهمترین در ساخت نمایشگر ها یا تابلو های تبلیغاتی مذکور این است که هر led در بدنه نمایشگر باید قابل کنترل باشد. به عبارت دیگر بتوان هر led را در هر لحظه خاموش یا روشن کرد. شاید اولین راه حلی که به ذهن میرسد، این باشد که برای هر led یک پایه از میکرو را در نظر بگیریم تا بتوانیم در هر لحظه آن را خاموش یا روشن کنیم، ولی بلافاصله متوجه می شویم که این کار عملی نیست، زیرا به عنوان مثال برای ساخت یک نمایشگر 8×32 به ۲۵۶ عدد led و به دنبال آن به همین پایانه قابل کنترل توسط میکرو کنترلر نیاز داریم در مورد نمایشگر های بزرگتر، ممکن است تعداد led ها به چندین هزار عدد هم برسد.

تکنیک جاروب (scan)

برای حل این مشکل، راه حل بسیار کار آمد تر و کم هزینه تری نیز هم وجود دارد. این روش از ضعف چشم انسان در دنبال کردن پدیده های سریع استفاده میکند. ساختار چشم انسان به گونه ای است که اگر پدیده ای در حدود ۴۰ بار (یا بیشتر) در ثانیه تکرار شود، دیگر قادر به تشخیص نوسان آن نیست؛ بنابراین از دید انسان، نور چراغی که با فرکانس بیش از ۴۰ HZ روشن و خاموش میشود، بدون لرزش و ثابت به نظر میرسد.

به این ترتیب اگر ما یک LED را که در بدنه نمایشگر به کار رفته است، در هر ثانیه ۵۰ بار روشن و خاموش کنیم، از دید ناظر، روشن به نظر میرسد. بنا به خاصیت گفته شده، ساختار ماتریسی شکل روبرو برای این نمایشگر پیشنهاد می شود.

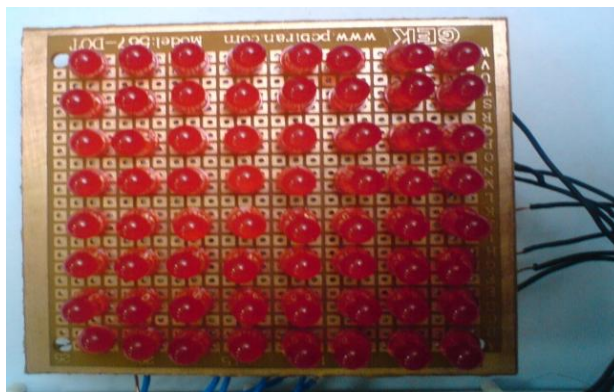


شکل ۲-۲: بلوک دیاگرام تابلو روان برای پردازنده

همانطور که مشاهده میکنید در تصویر فوق من نمونه ای از جدول گلایف را برای حرف (a) و صورتک خندان ترسیم کردم و در ستون سمت چپ هر تصویر کد هگزمربوط به هر سطر را درج کردم. که با فرض این بوده که پهنای هر کاراکتر هشت پیکسل بوده و به ازای هر پیکسل فعال بیت مرتبط با آن یک در نظر گرفته شده. در نتیجه در دو مثال فوق که ارتفاع هر کاراکتر ۱۲ سطر است، برای ذخیره اطلاعات هر کاراکتر به ۱۲ بایت نیاز داریم. جهت نمایش مناسب تصاویر متحرک باید حداقل ۲۴ تصویر در ثانیه نمایش داده شود. حال فرض کنید شما یک تابلو با ۳۲ سطر می خواهید طراحی کنید و از جاروب سطری هم استفاده می کنید در این حالت زمان نمایش هر فریم تصویر برابر با ۴۱,۶ میلی ثانیه خواهد بود و در هر فریم ۳۲ سطر جهت جاروب داریم پس زمان روشن بودن هر سطر برابر با ۱,۳ میلی ثانیه خواهد بود.

خوب شما مدار را طراحی و میسازید اما در پایان متوجه میشوید که نور ال ای دی ها بسیار کم تر از حالت معمولی است و حسایی متعجب خواهید شد که چرا با وجود استفاده ازال ای دی های مرغوب نور تابلو روان تا این حد کم است؟!

نکته اینجاست که شما هر ال ای دی را فقط به مدت ۱,۳ میلی ثانیه روشن نگاه میدارید و سپس به مدت ۳۱ برابر این مدت خاموش نگاه میدارید (به خاطر جاروب ۳۱ سطر بعدی) یعنی ۱,۳ میلی ثانیه روشن و ۴۰,۳ میلی ثانیه خاموش است. و در واقع اثر نور ال ای دی در چشم به میزان قابل توجه ای کاهش می یابد.



آزمایش شماره ۳، $330\text{ k}\Omega$

آزمایش

اتصال LED به میکرو و کنترل تاخیر چشمک زدن آن

وسایل مورد نیاز:

۱- مقاومت: $10\text{ k}\Omega$ ، 330Ω ، 100Ω

۲- برد مورد (bread board)

۳- منبع تغذیه

۴- سیم رابط

۵- کریستال 8MHz

۶- ۱ عدد خازن الکترولیتی $10\mu\text{F}$ و ۲ عدد خازن عدسی 22PF

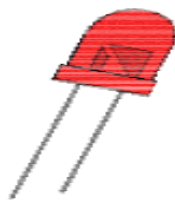
۷- میکروکنترلر (ATMEGA32 AVR)

۸- ۱ عدد LED

۹- رگلاتور 7805

هدف آزمایش:

در این آزمایش روشهای ایجاد تاخیر زمانی و استفاده از آن در کنترل یک LED را بررسی خواهیم کرد.

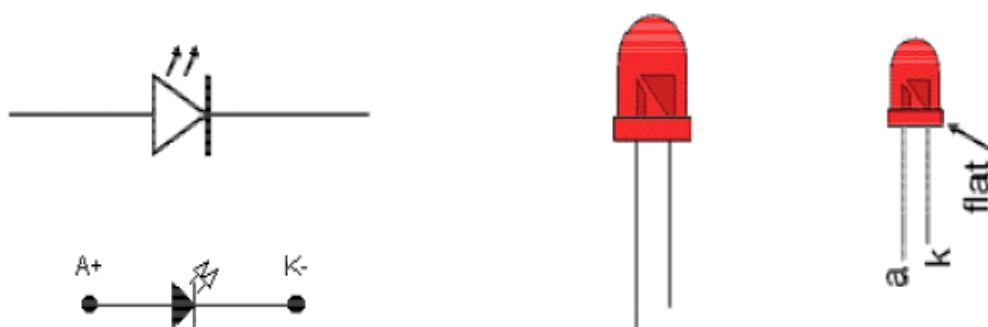


ابتدا عملکرد اجزاء مورد استفاده در این آزمایش را مورد بررسی قرار می دهیم.

LED:

دیود های نورانی و نحوه عملکرد آنها یک LED در واقع دیودی است که با روشن شدن از خود نور ساطع میکند. در این حالت ، اختلاف پتانسیلی در حدود 0.7 ولت بین دو سر آند و کاتد آن مشاهده می گردد .

در زیر شکل یک LED ساده را می بینید:



چطور یک دیود نور تولید می کند؟

نور شکلی از انرژی است، نور از اجزا بسیار ریزی به نام فوتون تشکیل می شود، فوتون ها انرژی و لختی دارند اما جرم ندارند در واقع فوتون ها در نتیجه حرکت الکترون ها آزاد می شوند در یک اتم الکترون ها در اربیتال های دور هسته می چرخند الکترون های اربیتال های مختلف مقدار انرژی متفاوتی دارند، کلاً الکترون ها با انرژی بیشتر در اربیتال های دور تر از هسته حرکت می کنند . برای یک الکترون برای پرش از یک اربیتال پایین به بالا چیزی که باید بگیرد انرژی است برعکس الکترون وقتی از اربیتال بالا به پایین می افتد انرژی آزاد می کند این انرژی به شکل فوتون آزاد می شود، یک فوتون بیشتر فوتون بیشتری آزاد می کند که با فرکانس بیشتر مشخص می شود. همان طور که در بخش قبل دیدیم الکترون هایی که از دیود عبور می کنند می توانند در حفره های لایه P بیفتند. این یک فوتون از باند رسانایی به اربیتال پایین تر است بنابر این الکترون ها انرژی به شکل فوتون آزاد می کنند این در هر دیود رخ می دهد اما فقط وقتی

شما فتون ها را می بینید که دیود از ماده خاصی ساخته شده باشد برای مثال اتم ها در یک دیود سیلیکون استاندارد به نحوی چیده شده اند که افت الکترون فاصله کمی دارد بنابر این فرکانس فتون ها به قدری کم است که با چشم انسان دیده نمی شود این در بخش مادون قرمز طیف نور است که لزوماً چیز بدی نیست البته LED های مادون قرمز برای کنترل های از راه دور مناسب اند.

دیود های منتشر کننده نور مری (VLEDs) مانند آنهایی که شماره های یک ساعت دیجیتال را روشن می کنند از ماده هایی با این خصوصیت ساخته شده اند که فاصله بین اربیتال رسانایی آنها و اربیتال پایین تر بیشتر است. اندازه این فاصله فرکانس فتون را نشان می دهد به عبارت دیگر این فاصله رنگ نور را مشخص می کند. در حالی که همه دیود ها نور آزاد می کنند اغلب آنها این کار را به طور موثری انجام نمی دهند یک دیود معمولی ماده نیمه رسانا خودش مقدار زیادی انرژی نوری جذب می کند. LED ها به نحوی ساخته می شوند که نور را در جهت خاصی متمرکز می کند. همانطور که در تصویر می بینید اغلب نور تولید شده در کناره دیود از طریق قسمت گرد بیرون می آید.

LED ها مزایای زیادی بر لامپ های معمولی دارند. یکی از آنها این است که آنها فیلامان ندارند که بسوزد بنابر این عمر طولانی دارند. به علاوه حباب پلاستیکی شان دوامشان را بیشتر می کند. و همچنین خیلی راحت در مدارهای الکترونیکی قرار می گیرند. اما مزیت اصلی آنها کیفیت آنها است در مقایسه با لامپ های معمولی فرآیند تولید نور باعث تولید مقدار زیادی گرما نمی شود.

فیلامانی برای گرم شدن ندارد) برای تولید نور باید فیلامان گرم شود.

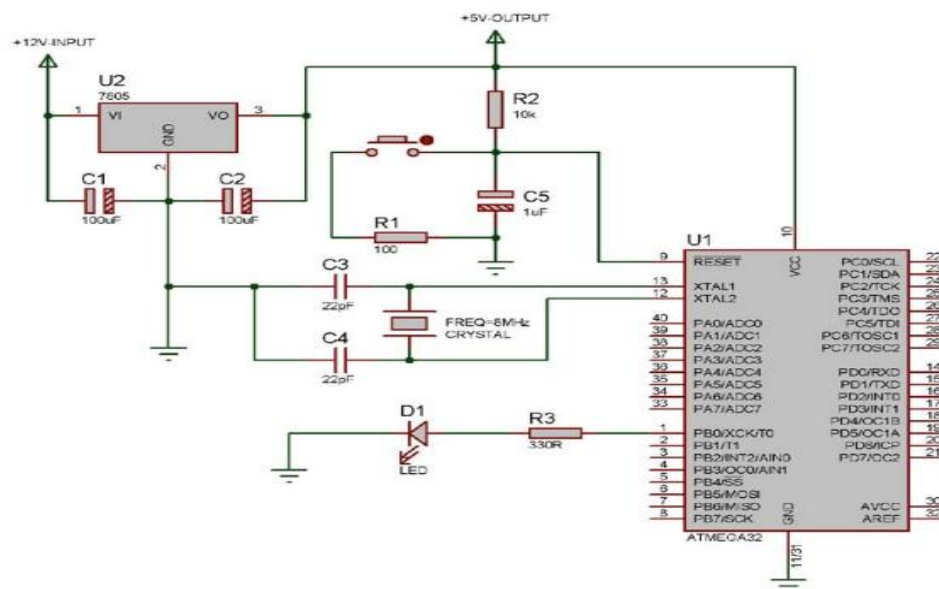
شرح آزمایش

در مرحله اول پیاده سازی هر آزمایش می بایست مدار راه انداز میکرو که همان اتصال کریستال و خازن ها و همچنین بستن رگلاتور و مدار RESET میکرو می باشد که در آزمایش قبل مورد بررسی قرار گرفت.

قسمت اول: در این مدار با توجه به نوع عملکرد آن که می بایست یک دیود نورانی (LED) را در یک زمان مشخص خاموش و روشن کند. که این جا زمان مشخص شده برابر با ۱ ثانیه می باشد و در یک ثانیه این دیود می بایست خاموش و در ثانیه بعد روشن شود و این سیکل به همین صورت باقی بماند.

برای شروع نیازمند بستن یک LED به این مدار هستیم که LED را همانند مدار زیر توسط یک مقاومت ۳۳۰ اهم به پایه شماره ۱ که پایه شماره صفر پورت B می باشد وصل می کنیم سر دوم LED را با توجه به یکی از روشهای شناخت پایه های LED پایه ای را که باید به منفی یا زمین وصل شود را پیدا کرده و به زمین وصل می کنیم. مدار آماده است و زمان اجرای برنامه و پروگرام برنامه در این میکرو کنترلر می باشد.

✍️ برای جلوگیری از آسیب رسیدن به پین میکرو، بین آن و LED باید مقاومتی (حدود ۳۰۰ الی ۴۰۰ اهم) به عنوان محدود کننده جریان قرار داد. بطور کلی نباید جریان عبوری از LED از ۲۵mA تجاوز کند.



قسمت دوم: در این بخش با در نظر گرفتن پورت های که ورودی یا خروجی شده اند برنامه را می نویسیم . همانطور که در بخش های قبل در مورد برنامه نویسی توضیح داده شده است به توضیح بعضی از قسمت ها می پردازیم .

```
01- $regfile = "m32def.dat"
02- $crystal = 8000000
03- Config PortB.0 = Output
04- PortB.0 = 0
05-   Do
06-     PortB.0 = Not PortB.0
07-     Waitms 256
08-   Loop
09- End
```

در خط ۱ – معرفی میکرو مورد استفاده در این آزمایش.

در خط ۲ – کریستال استفاده شده که اینجا ۸ مگا هرتز می باشد

در خط ۳ – پیاده سازی پورت B.0 به عنوان خروجی

در خط ۴ – پورت B.0 را برابر 0 قرار می دهیم (ولتاژ خروجی برابر صفر ولت خواهد شد).

در خط ۵ – حلقه برای اجرای برنامه

در خط ۶ – پورت PB.0 را در هر مرحله از اجرا حلقه معکوس می کند به این معنی که اگر صفر باشد به یک و اگر یک بود به صفر تبدیل میکند.

در خط ۷ – زمان توقف ، به این معنی که از اجرای دستور بعدی تا مدت زمان خاصی جلوگیری کرده و در همین خط باقی می ماند هنگامی که از کریستال ۴ مگا هرتز استفاده شود زمان تاخیر دستور Waitms1 تقریباً برابر 1ms می باشد . البته برای بدست آوردن زمان دقیق می بایست از تایمر های میکرو استفاده کرد

در خط ۸ – ادامه حلقه تا بی نهایت

در خط ۹ – پایان

و بعد از نوشتن برنامه آن را کامپایل نموده و در میکرو برنامه ریزی می کنیم و منبع تغذیه را وصل کرده و نتیجه را مشاهده می کنیم.

سوالات و پروژه های پیشنهادی

۱. هشت عدد LED را به پورت B متصل نموده و می خواهیم یک رقص نور درست نماییم به گونه ای که ، فاصله زمانی روشن بین آنها ۱ ثانیه و از سمت چپ به راست باشد (کم ترین تا پر ارزش ترین بیت پورت B)؟
۲. برنامه تمرین قبل را بگونه ای تغییر دهید که حرکت رقص نور از سمت راست به چپ باشد ؟
۳. پروژه های بالا را یک بار با اسیلاتور داخلی و یک بار با اسیلاتور خارجی (کریستال) در فرکانس های ۸۰۴۰۱ پیکر بندی کنید و نتیجه را با هم مقایسه کنید ؟
۴. ولتاژی که بین دو سر آند و کاتد دیود می افتد چقدر است ؟ با توجه به این ولتاژ محدوده مقاومت های مجاز بین LED و میکرو را برای داشتن روشنایی مناسب و عدم آسیب رسیدن به پین میکرو را تعیین کنید؟

آزمایش شماره ۴

هدف آزمایش طراحی و پیاده سازی چراغ چشمک زن به همراه یک کلید

وسایل مورد نیاز:

۱- مقاومت: $10k\Omega$ ، 100Ω ، $330k\Omega$

۲- برد مورد (bread board)

۳- منبع تغذیه

۴- سیم

۵- کریستال 8MHZ

۶- ۱ عدد خازن الکترولیتی 10 μ F و ۲ عدد خازن عدسی 22PF

۷- ۲ عدد کلید فشاری

۸- میکرو کنترلر (AVR) ATMEGA32

۹ - ۸ عدد LED

۱۰- رگلاتور 7805

هدف آزمایش:

آشنایی با دستور کلید و نحوه خواندن ورودی دیجیتال از میکرو را مورد بررسی قرار می دهیم.

ابتدا عملکرد اجزاء مورد استفاده در این آزمایش را مورد بررسی قرار می دهیم.

صفحه کلید ماتریسی و کلید تکی

برای ارتباط و ارسال دستورات و اطلاعات به میکرو به دستگاه های ورودی نیاز است . کلید های مکانیکی یکی از ساده ترین وسایل ورودی برای میکروکنترلر می باشند. کلید ها در دو نوع ساده (تکی) و ماتریسی (صفحه کلید) وجود دارند .

قبل از اینکه به پیکر بندی کلید پردازیم لازم به ذکر است پورت یا پین هایی که به عنوان ورودی قرار می گیرند را می توانیم به صورت خارجی با یک مقاومت ۱۰ کیلو اهم Pull-Up یا Pull-Down کرد تا در هنگام فشردن کلید تغییر وضعیت ولتاژ در آن پین بدون نویز و با اطمینان از فشرده شدن کلید عملیات مربوط به آن را انجام بدهد و هنگامی که یک پین را به عنوان ورودی انتخاب می کنیم می توانیم مقاومت های Pull-up داخلی را فعال یا غیر فعال کنیم. مثلاً

Config portb.0 = input

PortB. 0 =1

در خط اول پورت B پین 0 را به عنوان ورودی تعریف می کنیم و در خط دوم با یک کردن این پین مقاومت های PULL-UP را فعال می کنیم . و همچنین با قرار دادن ۰ به جای ۱ در این خط مقاومت های PULL-UP را می توان غیر فعال کرد.

کلید تکی که در مدار ریست از آن استفاده شد به این صورت است که اتصال بین دو پین ایجاد می کند . که اگر بخواهیم تعداد کلید ها را افزایش دهیم مثلاً به ۱۵ کلید می بایست ۱۵ پین از میکرو را اشغال کنیم . که این یکی از محدودیت های موجود آن می باشد . البته برای کار هایی که به تعداد کلید های محدود نیازمند است از این نوع کلید ها استفاده و بهره کافی برده می شود . در این مرحله با دستورات کلید تکی آشنا می شویم.

Config Debounce = X

Debounce مدت زمان تاخیر برای فشردن کلید است که از لغزش های موجود بروی کلید های مکانیکی جلوگیری و از فشردن شدن کلید نوعی اطمینان را ایجاد می کند. در صورتی که این دستور را استفاده نکنیم مقدار پیش فرض آن ۲۵ms است.

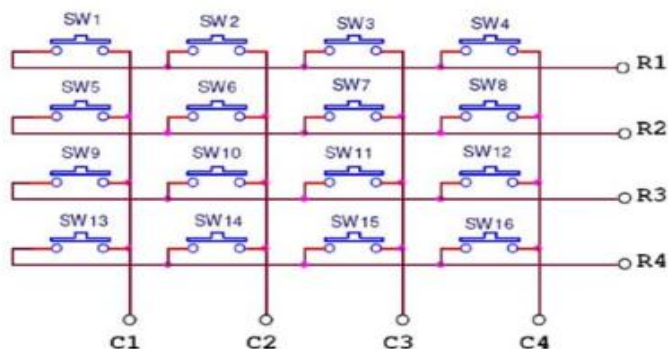
و با دستور زیر می توان وضعیت پین مورد نظر از یک پورت را خواند و با توجه به شرط برقراری state به برچسب یا تابع تعریف و مشخص شده پرش کرد . که عبارت نوشته شده در براکت به معنی آن است که می توان در برنامه عبارت SUB قید شود یا نشود.

DEBOUNCE Px.y , state , label [, SUB]

Example : Debounce PinB. 1, 0, Decrement , Sub

در مثال بالا در هنگامی که پین ۱ پورت B صفر شود به تابع Decrement می رود.

صفحه کلید از چند سطر و ستون تشکیل می شود. این سطر و ستون ها در حالت عادی هیچ اتصالی با هم نداشته و در صورت فشردن هر کلید سطر و ستون مربوط به آن کلید به هم متصل می گردد.



روشهای مختلفی برای پویش و شناسایی کلید های یک صفحه کلید وجود دارد.

برای این منظور یک صفحه کلید 4x4 در نظر می گیریم.

۱- پویش صفحه کلید با دستور (Getkbd):

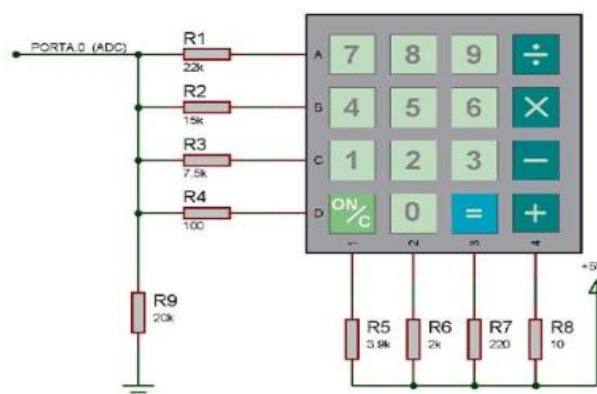
در این روش تمام سطر ها را با یک مقاومت (Pull-up) به VCC وصل کرده و ستون های صفحه کلید را به پورتی که به عنوان پورت مورد استفاده صفحه کلید است متصل می کنیم (پایه های از پورت که خروجی هستند). در این روش میکرو در هر نوبت یک ستون را صفر می کند و با فشردن کلید، سطر مربوط به آن کلید نیز صفر می شود که میکرو با در نظر گرفتن سطر و ستون صفر شده کلید فشرده شده را تشخیص می دهد.

می توان فاصله زمانی مناسب دفعات پویش صفحه کلید را با توجه به سرعت کلید زنی و متوسط زمان نگه داری کلید فشرده شده محاسبه نمود.

۲- پویش صفحه کلید با استفاده از یک پایه میکرو:

روش قبل هم از دقت بالایی برخوردار است ولی برای پویش صفحه کلید تعداد زیادی از پایه های میکرو درگیر می شدند به عبارتی برای اتصال یک صفحه کلید 4x4 تعداد ۸ پایه از میکرو درگیر می شوند و البته زمانی که به پایه های I/O نیاز داشته باشیم سودمند نخواهد بود و روشهای دیگری نیز برای رفع این مشکل وجود دارد که در این قسمت به ساده ترین آن می پردازیم.

با استفاده از پایه ADC داخلی میکرو و استفاده از تقسیم ولتاژ مقاومتی می توان روشی برای پویش صفحه کلید طراحی نمود. و.



با این روش با استفاده از یک پایه میکرو می توان یک صفحه کلید 4×4 را به راحتی و بدون اشغال پایه های دیگر مورد استفاده قرار داد در این مدار مقادیر R1 تا R2 به گونه ای انتخاب شده اند تا با فشردن کلید های مختلف ولتاژ مختلفی را در ADC ایجاد کند تا ولتاژ خروجی هر کلید منحصر به فرد باشد. برنامه مربوط به پویش صفحه کلید را می توان یک بار با خواندن مقادیر از ADC به صورت دقیق پیاده سازی کرد.

در این قسمت به توضیح در مورد اتصال صفحه کلید به روش اول می پردازیم:

اتصال صفحه کلید به AVR در BASCOM بسیار ساده می باشد. تنها کافی است تا صفحه کلید را به یکی از پورت های آن متصل کنیم و سپس با استفاده از دستور زیر صفحه کلید را پیکربندی می کنیم:

Config Kbd = PortX , Debounce = Y , Delay = Z

در این دستور به جای X پورتی را که صفحه کلید به آن وصل است را مشخص می کنیم. و با مقدار Y مدت زمان فشردن کلید و Z هم مدت زمانی را که منتظر فشردن کلید است را تعیین می کند و در صورتی که این عبارات (Y,Z) را وارد نکنیم مقدار پیش فرض خود BASCOM مورد استفاده قرار می گیرد.

و دستوری که مقدار را از صفحه کلید می خواند دستور زیر می باشد:

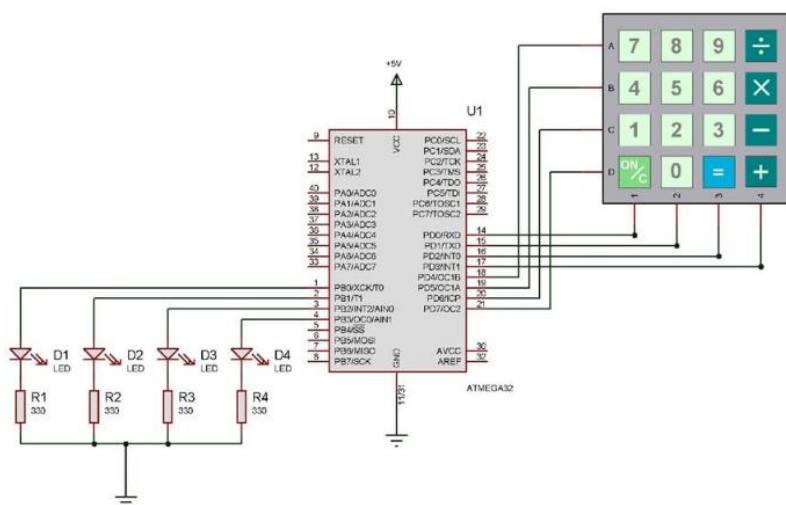
X = Getkbd ()

X مقدار خروجی یا برگرداندن مقدار عددی کلید فشرده شده است. در اینجا می بایست اشاره کرد که پس از فشردن صفحه کلید توسط این تابع یک مقدار عددی باز گردانده می شود و برای اینکه نتیجه حاصل با کلید فشرده شده یکی باشد از جدول Data که در انتهای برنامه قرار میگیرد استفاده می شود.

برای مثال:

```
$regfile = "m32def.dat"
$crystal = 8000000
Config kbd = portd, Debounce = 50, Delay = 255
Config Portb = Output
Dim A as byte
Do
  A = Getkbd ()
  A = Lookup (a, Data_key)
  If A < 16 Then Portb = A
Loop
End
Data_key:
Data 7 , 8 , 9 , 15 , 4 , 5 , 6 , 14 , 1 , 2 , 3 , 13 , 12 , 0 , 11 , 10
```

شماتیک مدار میکرو به همراه صفحه کلید که مقدار ورودی صفحه کلید را توسط LED هایی که به پورت B متصل است نمایش می دهد. که مقدار نمایشی آن به صورت دیجیتال از صفر تا ۱۵ را نمایش می دهد.

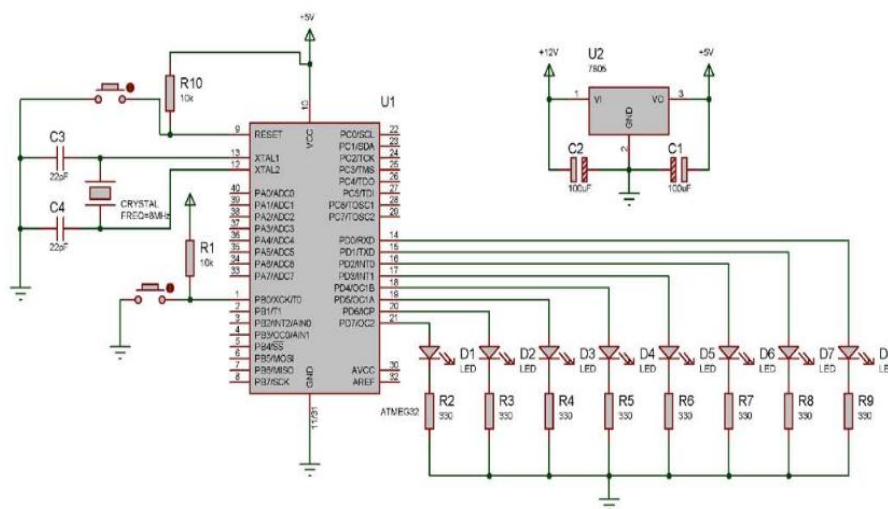


شرح آزمایش:

در مرحله اول پیاده سازی هر آزمایش می بایست مدار راه انداز میکرو که همان اتصال کریستال و خازن ها و همچنین بستن رگلاتور و مدار RESET میکرو می باشد که در آزمایش دوم مورد بررسی قرار گرفت.

در این مدار با توجه به نوع عملکرد آن که می بایست با زدن کلید فشاری مدار شروع به کار کرده و هشت دیود نورانی (LED) را در یک زمان مشخص به ترتیب خاموش و روشن کند. که این جا زمان مشخص شده برابر با ۱ ثانیه می باشد و در یک ثانیه هر LED به ترتیب روشن و خاموش می شود و این سیکل به همین صورت باقی می ماند.

برای شروع با توجه به توضیح بالا نیازمند بستن هشت LED و یک میکرو سوئیچ (کلید تکی) به این مدار هستیم که LED ها را همانند مدار زیر به پایه های شماره ۱۴ تا ۲۱ که پایه شماره صفر پورت D تا پایه شماره ۷ همین پورت می باشد وصل می کنیم سر دوم LED ها را با توجه به یکی از روشهای شناخت پایه های LED پایه ای را که باید به منفی یا زمین وصل شود را پیدا کرده و توسط هشت مقاومت ۳۳۰ اهم به زمین وصل می کنیم. پایه شماره ۱ که پایه شماره صفر پورت B می باشد را به یک میکرو سوئیچ و پایه دوم سوئیچ را به زمین (GND) مدار و همان پایه میکرو را با یک مقاومت ۱۰ کیلو اهمی به $VCC (+V5)$ وصل می کنیم. مدار آماده است و زمان ، زمان اجرای برنامه و نصب برنامه در این میکرو کنترلر می باشد.



نرم افزار میکرو:

```
$regfile = "m32def.dat"
```

```
$crystal = 8000000
```

```
Config Portd = Output
```

```
Config Portb.0 = Input
```

```
Portb.0 = 1
```

'Pull up resistor is Activated

```
Do
```

```
Debounce Pinb.0, 0, Start_lbl
```

```
Loop
```

```
Start_lbl:
```

```
Portd = 128
```

```
Do
```

```
Rotate Portd , Right
```

```
Waitms 500
```

```
Loop
```

```
End
```

'Flashing From PB7 to PB0

در خط ۱ - معرفی میکرو مورد استفاده در این آزمایش.

در خط ۲ - کریستال استفاده شده که اینجا ۸ مگا هرتز می باشد

در خط ۳ - پیاده سازی پورت D به عنوان خروجی

در خط ۴ - پیاده سازی پورت B.0 به عنوان ورودی

در خط ۵ - پین شماره 0 پورت B را برابر ۱ قرار می دهیم . و این کار برای قرار دادن مقاومت Pull up داخلی می باشد و برای این مدار می توان این دستور را قرار نداد چرا که این پین میکرو به صورت خارجی نیز pull up شده است.

در خط ۶ - یک حلقه ایجاد می کنیم تا دستورات اجرا گردند. که این حلقه تا زمانی ادامه پیدا می کند که کلیدی فشرده نشود و پس از برقرار یا صفر شدن این پین B.0 به برجسی که به آن اشاره می کند پرش می کند.

در خط ۷ - نام برجسب این خط را مشخص میکند.

در خط ۸ - پورت D را برابر ۱۲۸ قرار می دهیم.

در خط ۹ - حلقه بی پایان برای اجرای دستورات داخل آن

در خط ۱۰ - با هر بار اجرای این دستور بیت های این پورت یک بار به راست شیفت پیدا می کند . یعنی از بیت پر ارزش به بیت کم ارزش حرکت می کند.

در خط ۱۱ - زمان توقف ، به این معنی که از اجرای دستور بعدی تا مدت زمان خاصی جلوگیری کرده و در همین خط باقی می ماند.

در خط ۱۲ - ادامه حلقه تا بی نهایت

در خط ۱۳ - پایان



سوالات و پروژه های پیشنهادی

۱- برنامه بالا را چگونه ای تغییر دهید که با هر بار فشار کلید مدت زمان تاخیر آن افزایش یابد؟

۲- برنامه بالا را با اسلاتور داخلی ۱ مگا هرتز و ۴ مگا هرتز نیز آزمایش کنید؟

۳- سعی کنید کلید دیگری به مدار اضافه کنید . چگونه ای که با زدن آن حرکت خاموش روشن شدن LED ها از چپ به راست تغییر کند . (بیت کم ارزش به سمت بیت پر ارزش)؟

۴- صفحه کلیدی به مدار ببندید و برنامه ای برای آن بنویسید که با زدن کلید ۱ رقص نور از بیت پر ارزش به سمت کم ارزش صورت گیرد و با زدن کلید ۲ جهت حرکت خاموش و روشن شدن LED ها تغییر کند و با زدن کلید ۳ توقف کند به گونه ای که اگر بار بعدی کلید ۳ را فشار دهد به حرکت ادامه دهد؟

آزمایش شماره ۵

آزمایش

طراحی و پیاده سازی چراغ راهنمایی و رانندگی

وسایل مورد نیاز:

۱- مقاومت: (6) , 330Ω , $10k\Omega$

۲- برد مورد (bread board)

۳- منبع تغذیه

۴- سیم

۵- کریستال 8MHZ

۶- ۱ عدد خازن الکترولیتی $10\mu F$ و ۲ خازن عدسی 10PF

۸- ۶ عدد LED و از هر رنگ (سبز ، زرد ، قرمز) ۲ عدد

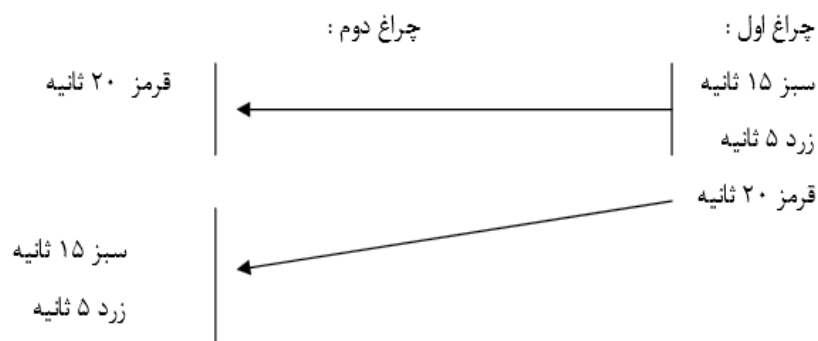
۹- رگلاتور 7805



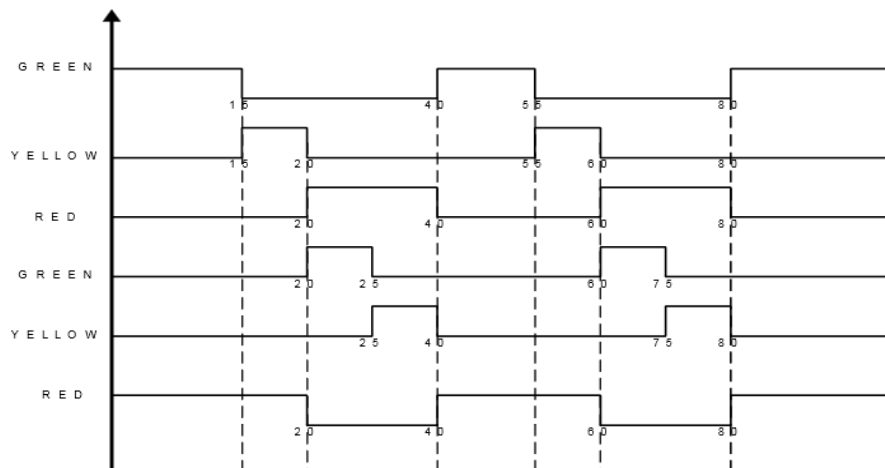
هدف آزمایش : آشنایی با یکی از کاربردهای میکرو

شرح آزمایش:

چراغ کنترل ترافیکی بسازید که بتواند بطور هم زمان رفت آمد دو خیابان را کنترل کند . دو چراغ را طوری طراحی کنید که وقتی لامپ سبز چراغ اول روشن است لامپ قرمز چراغ دوم روشن باشد سپس پس از طی زمان ۱۵ ثانیه لامپ زرد چراغ اول روشن شود ولی همچنان لامپ قرمز چراغ دوم روشن باشد و پس از طی زمان ۵ ثانیه لامپ قرمز چراغ اول روشن شود و لامپ قرمز چراغ دوم خاموش و لامپ سبز آن روشن شود و پس از طی ۱۵ ثانیه لامپ زرد چراغ دوم روشن شود ولی همچنان لامپ قرمز چراغ اول روشن باشد و پس از طی زمان ۵ ثانیه لامپ قرمز چراغ دوم روشن شود و لامپ قرمز چراغ اول خاموش و این روند تکرار شود . به نمودار زیر توجه کنید.

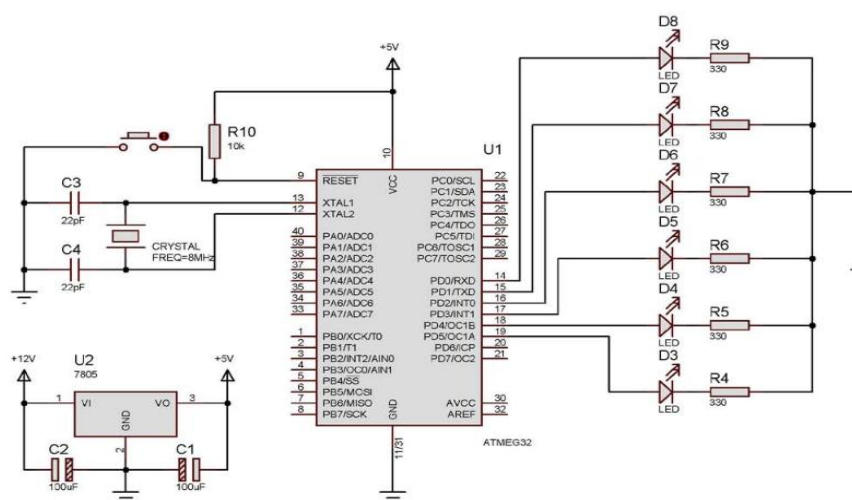


نمودار زمانی و پالس چراغ اول و دوم به صورت زیر می باشد:



طراحی مدار کنترل چراغ راهنمایی و رانندگی:

همانند آزمایش قبل مدار راه انداز را به میکرو وصل کرده و مابقی کار های که در این آزمایش می بایست انجام داد را شروع می کنیم . وصل کردن ۶ LED با رنگهای سبز و زرد و قرمز که به ترتیب از پایه 0 پورت D شروع به بستن LED ها به ترتیب از سبز به پایه 0 این پورت و زرد به پایه 1 این پورت و قرمز به پایه 2 و دوباره برای چراغ دوم نیز سبز به 3 و زرد به 4 و قرمز به 5 توسط ۶ مقاومت 330 اهمی وصل کرده و پایه دوم هر کدام از این دیود ها را به زمین مدار متصل می نمایم. در این قسمت قابل ذکر است که برای یک چهار راه به دو چراغ راهنمایی دیگر به غیر از این دو چراغ نیاز است . که این دو چراغ را در این مدار می توانید با انشعاب گرفتن از زوج مقابل آن، درست کنید.



برنامه نوشته شده برای این آزمایش:

```
$regfile = "m32def.dat"
$crystal = 8000000
Config Portd = Output
Do
    Portd = 0
    Portd.0 = 1
    Portd.5 = 1
    Wait 15
    Portd.0 = 0
    Portd.1 = 1
    Wait 5
    Portd.1 = 0
    Portd.2 = 1
    Portd.3 = 1
    Portd.5 = 0
    Wait 15
    Portd.3 = 0
    Portd.4 = 1
    Wait 5
Loop
End
```

۱- معرفی نام میکروکنترلر و در خط

۲- تعیین میزان فرکانس کریستال به واحد هرتز

۳- تعیین پورت خروجی که در این برنامه پورت D به عنوان خروجی تعریف شده است.

۴- سرآغاز حلقه

۵- پورت D0 و پورت D5 را در هر وضعیتی که باشند یک و مابقی را صفر می کند.

۸- انتظار یا توقف برنامه به مدتی که در مقابل آن قرار دارد البته با توجه به فیوز بیت های کریستال داخلی یا خارجی که تعیین شده است زمان توقف متفاوت است

۹- پورت D0 صفر می کند و در خط ۱۰- پورت D1 را یک می کند.

۱۱- انتظار یا توقف برنامه به مدتی که در مقابل آن قرار دارد

- ۱۲-) پورت D1 را صفر می کند و در خط ۱۳ پورت D2 و در خط ۱۴ پورت D3 را یک می کند و در خط -
- ۱۵-) پورت D5 را صفر می کند.
- ۱۶-) انتظار یا توقف برنامه به مدتی که در مقابل آن قرار دارد
- ۱۷-) پورت D3 را صفر و در خط ۱۸ پورت D4 را یک می کند.
- ۱۹-) انتظار یا توقف برنامه به مدتی که در مقابل آن قرار دارد
- ۲۰-) انتهای حلقه و بازگشت به ابتدای حلقه و تکرار برنامه
- ۲۱-) پایان برنامه



سوالات و پروژه های پیشنهادی

- ۱- برنامه بالا را بگونه ای تغییر دهید که با هر بار فشار کلید مدت زمان تاخیر افزایش یابد؟
- ۲- با قرار دادن ۴ کلید به طوری که هر دو کلید برای یک سمت چهار راه قرار گیرد و با فشردن آنها (به معنی وجود ترافیک پشت چراغ) زمان روشن بودن چراغ سبز برای سمتی که تعداد کلید بیشتری فعال است بیشتر گردد؟

آزمایش شماره ۶

آزمایش

نحوه عملکرد LCD و اتصال آن به میکرو

وسایل مورد نیاز:

۱- LCD ۱۶*۲

۲- مقاومت: $10k\ \Omega$ و $100\ \Omega$

۳- برد مورد (bread board)

۴- منبع تغذیه

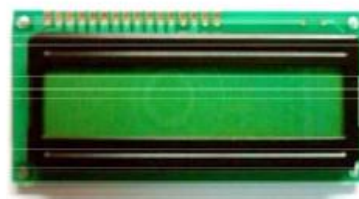
۵- سیم رابط

۶- کریستال 8MHZ

۷- ۲ عدد خازن الکترولیتی 100 μ F و ۱ عدد خازن 1 μ F و ۲ عدد خازن عدسی 22PF

۸- میکروکنترلر ATMEGA 32

۹- رگلاتور 7805



هدف آزمایش

در بیشتر مدارهای میکروکنترلی نیاز به نمایش اطلاعات به کاربر را داریم و اغلب این کار را با روشهای مختلف مانند LED ها یا 7Segment یا LCD انجام می دهیم . اما LCD ها به خاطر داشتن قابلیت های بیشتر نمایش و ابعاد و برنامه ریزی آسان مورد توجه بیشتری قرار می گیرند . در این آزمایش با نحوه عملکرد LCD های کارکتری و انواع LCD و همچنین نحوه اتصال آن به میکرو و دستورات پر کاربرد مربوط به صفحه نمایش کارکتری آشنا می شوید.

ابتدا عملکرد اجزاء مورد استفاده در این آزمایش را مورد بررسی قرار می دهیم.

صفحه نمایش LCD :

امروزه از LCD ها در کار برد های مختلفی از جمله مولتی متر های دیجیتال ، وسایل اندازه گیری دیجیتالی ، کرونومترهای اتومبیل ،... به عنوان نمایشگر کم مصرف ، ساده و با وضوح مناسب استفاده می شود.

کریستال مایع در سال ۱۸۸۸ توسط گیاه شناس استرالیایی به نام فردریش رینیزر (Fredrich Reinitzer) کشف شد و لامپ اشعه کاتدی نیز در سال ۱۸۹۷ توسط کارل براون (Karl Braun) دانشمند آلمانی اختراع گردید. اگر چه کریستال مایع ۹ سال پیش از اختراع لامپ اشعه کاتدی کشف شده بودند، ایده استفاده از آنها تا سال ۱۹۵۸ مسکوت باقی ماند. در این زمان دکتر گلن براون (Gelen Brown) از ایلات متحده، اولین مقاله را درباره LCD منتشر کرد. سرانجام در سال ۱۹۶۷ جیمز فرگاسون (Fergason James) با کشف فناوری TNLCD اولین نمایشگر کریستال مایع واقعی را تولید کرد.

گروه RCA (Radio Corporation of America)، در حدود یک سال بعد، نمایشگر LCD را با فناوری پخش پویا DSM (Dynamic Scattering Mode) ارائه نمود و در سال ۱۹۷۲ اولین ساعت LCD بر پایه نظریه فرگاسون ساخته شد. پس از آن در سال ۱۹۷۳ شرکت (Sharp) اولین ماشین حساب جیبی را با استفاده از فناوری DSMLCD عرضه کرد.

در سال ۱۹۷۹ والتر اسپیر (Walter Spear) و پتر لکومبر (Peter Lecomber) اولین LCD رنگی را با استفاده از ترانزیستور فیلم نازک TFT (Thin Film Transistor) طراحی نمودند.

LCD ها شامل انواع کارکتری و گرافیکی هستند که در اینجا نوع کارکتری آن را معرفی و نحوه استفاده آن را شرح می دهیم.

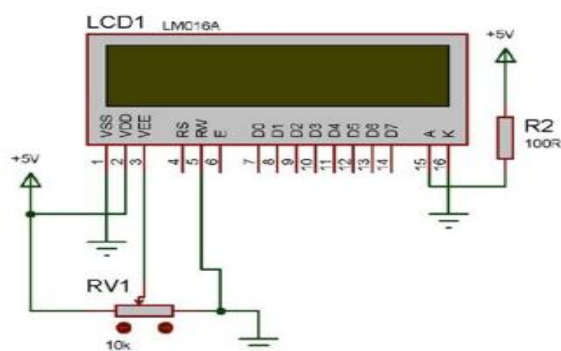
ویژگی اصلی LCD های کنونی عبارتند از:

- قابلیت نمایش اعداد، انواع کارکترها و حتی گرافیک.
- ترکیب یک کنترلر در داخل LCD موجب ساده سازی مدار و نحوه ارتباط آن با دنیای خارج می گردد.
- کارکترهایی مانند حروف فارسی و... را می توان با ارسال فرمان های خاصی به LCD تعریف کرد و نمایش داد.

مدار راه انداز LCD :

- ۱- پایه ۱ به زمین (GND) وصل می گردد.
- ۲- پایه ۲ به $VCC 5V$ وصل می شود.
- ۳- پایه ۳ به یک مقاومت متغیر که یکی از دو سر مقاومت به VCC و سر دیگر آن به GND وصل شده باشد.

۴- LCD ها اغلب دارای ۱۶ پایه می باشند که همانطور که در شکل می بینید پایه های ۱ تا ۱۴ برای تغذیه و ارسال اطلاعات می باشد و دو پایه ۱۵ و ۱۶ برای روشن کردن دیود نوری (LED) پس زمینه بکار می رود که در این مدار همواره روشن است با استفاده از ترانزیستور BJT می توانید نور آن را توسط میکرو کنترلر یا با یک مقاومت وابسته به نور (LDR) آن را به طور خود کار و با توجه به نور محیط کنترل کنید .



شرح پایه های LCD : در شکل بالا یک ۱۶*۲ LCD را مشاهده می کنید . که دارای ۱۶ پایه بوده که شرح مختصری در باره این پایه در این قسمت بیان می شود.

۱- پایه منفی یا GND مدار LCD

۲- پایه مثبت یا VCC مدار LCD که ولتاژ تغذیه ۵ ولت می باشد

۳- پایه تنظیم نور نمایش کارکترها

۴- RS یا انتخاب کننده ثبات LCD که شامل دو فرمان زیر می شود.

- ارسال فرمان به LCD (کد ۰ یا برای اجرای این دستور این بیت را صفر می کنیم.
- ارسال داده به LCD (کد ۱ یا برای اجرای این دستور این بیت را یک می کنیم.

۵- RW از این پایه در تعیین نوع عمل در خواستی از LCD استفاده می شود . در عمل خواندن اطلاعات از LCD این پایه برابر یک و در عمل نوشتن به LCD (ارسال فرمان یا داده) این پایه صفر می گردد.(و معمولاً چون عمل نوشتن صورت می گیرد این پایه را به GND وصل می کنیم).

۶- E با ایجاد پالس مثبت یا منفی بر روی این پایه ، LCD عمل درخواستی را با توجه به وضعیت RS و R/W انجام می دهد.

۷- پایه های داده D0~D7: از این پایه ها برای ارتباط و تبادل داده یا فرمان با LCD استفاده می شود. در عملیات نوشتن بایت مورد نظر در محل این پایه ها قرار داده می شود و در عملیات خواندن، بایت های مختلفی از جمله بایت وضعیت LCD در این پایه ها ظاهر می شود.

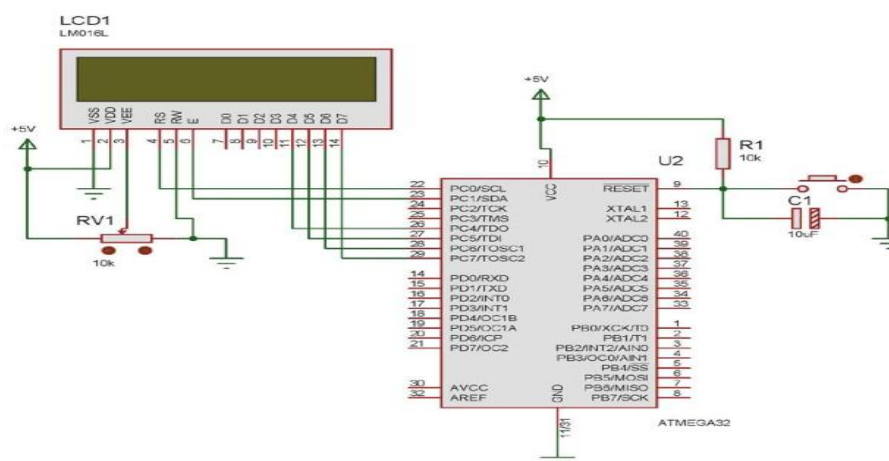
نحوه اتصال به میکرو: با اینکه این نوع LCD های کارکتری دارای ۸ پایه برای تبادل ارتباط هستند ولی قابلیت ارتباط با ۴ پین را نیز دارند که این امر باعث صرفه جویی در استفاده از پایه های میکرو می گردد. با توجه به شکل (۱)

شرح آزمائش

در شکل () حداقل مدار راه اندازی این LCD 16*2 توسط میکرو کنترلر ATmega32 نشان داده شده است . همانطور که مشاهده می کنید باس داده D4~D7 به پین های ۴ تا ۷ و پایه های کنترلی E,RS به پین های 0 و 1 پورت C متصل شده اند.

✍ با توجه به اینکه باس داده و پایه های کنترل LCD به پورت C متصل شده است باید ارتباط JTAG را در جدول فیوزیت های مربوط به میکرو کنترلر غیر فعال نکته :

کنید تا پورت C به صورت کامل در دسترس باشد.



نحوه پیکر بندی LCD در BASCOM

در این بخش دستورات مربوط به پیکر بندی LCD متنی را معرفی می کنیم:

۱- پیکر بندی باس LCD : اولین موردی که برای استفاده از یک LCD متنی مورد توجه قرار می گیرد نوع اتصال آن است . اگر در پروژه ای تعداد پایه های استفاده شده از AVR اهمیت نداشته بلکه نمایش سریع داده ها مورد نظر باشد از هر ۸ پایه ~ DB0 DB7 استفاده می گردد که به همراه پایه های دیگر حداقل ۱۰ پایه از میکرو را اشغال می کند و برعکس اگر هدف کمتر کردن تعداد پایه های میکرو برای ارتباط با LCD باشد و نمایش سریع داده ها اهمیت نداشته باشد از مد ۴ بیتی آن استفاده می نماییم . در این حالت فقط از پایه های DB4 تا DB7 استفاده می شود و پایه های DB0 تا DB3 به جایی متصل نمی گردد. در این مد حداقل ۶ پایه مورد استفاده قرار می گیرد. برای تعیین باس استفاده شده در سخت افزار به نرم افزار BASCOM از دستور زیر استفاده می شود که n می تواند مقدار ۴ یا ۸ را به خود بگیرد . ضمناً پیش فرض کامپایلر حالت $n = 4$ می باشد.

Config LCDbus = n

۲- پیکر بندی پایه های متصل به LCD:

Config LCDPIN = pin , DB 4= PN , DB 5= PN , DB 6= PN , DB 7= PN , E=PN , RS =PN

در این دستور PN پایه ای دلخواه از میکرو کنترلر بوده و هر پایه از LCD را که بخواهیم می توان به یکی از پایه های میکرو که در اینجا معرفی می گردد متصل نمود. به عنوان مثال اگر بخواهیم از پورت C برای اتصال به LCD استفاده نماییم دستور فوق به صورت زیر نوشته می شود.

Config LCDPIN = PIN , DB 4= portc. 4, DB 5= portc. 5, DB 6= portc. 6, DB 7= portc. 7, E = portc. 1, RS = portc.0

۳- تعیین نوع LCD :

LCD ها در انواع مختلف در بازار موجود می باشند اما متداولترین آنها با اندازه 16×2 می باشد . بدین معنا که دارا ۲ سطر و ۱۶ ستون یا در هر سطر ۱۶ کارکتر قرار می گیرد . پیش فرض کامپایلر نیز همین است . در دستور زیر Lcdtype مشخص کننده نوع LCD بوده که می تواند

حالت‌های ۱۶×۱، ۱۶×۲، ۲۰×۲، ۲۰×۴، ۴۰×۴ و ... باشد .

CONFIG LCD = Lcdtype

Example: Config LCD = ۱۶*۲

دستورات پر کاربرد مربوط به LCD:

- **دستور نمایش اطلاعات :** توسط این دستور می توانید مقادیر دلخواه خود را نمایش دهید. در اینجا به جای VAR می توان یک متغیر یا یک رشته ثابت را قرار داد مانند مثال:

LCD VAR

Example 1: LCD "Hello !"

Example 2: LCD "X = " ; Y

- **دستور CLS :** این دستور باعث پاک شدن کل صفحه نمایش می شود.

CLS

- **دستور Display :** توسط این دستور می توان صفحه نمایش LCD را خاموش یا روشن نمود.

Display On (or Off)

- **دستور Cursor :** توسط این دستور می توان مکان نما LCD را تنظیم نمود.

Cursor On (or Off or Blink or Noblink)

حالت روشن ON یا خاموش OFF و یا چشمک زدن Blink و یا چشمک نزدن NoBlink را برای مکان نما تنظیم نمود.

- **دستور Home :** نوشتن این دستور مکان نما را به سطر اول و ستون اول انتقال می دهد.

Home

اما با نوشتن این دستور به صورت (Home Upper (or Lower or Third or Fourth به ترتیب مکان نما را به اولین ستون سطر اول یا اولین ستون سطر دوم یا به اولین ستون سطر سوم یا اولین سطر ستون چهارم قرار دهد . در ضمن حالت خلاصه شده دستور نیز به صورت Home u (or l or t or f وجود دارد.

Example: Home Lower

- **دستور Locate**: این دستور مکان نما را به محل دلخواه انتقال می دهد . که x تعیین کننده سطر و y مشخص کننده شماره ستون است.

Locate x, y

Ex: Locate1,5

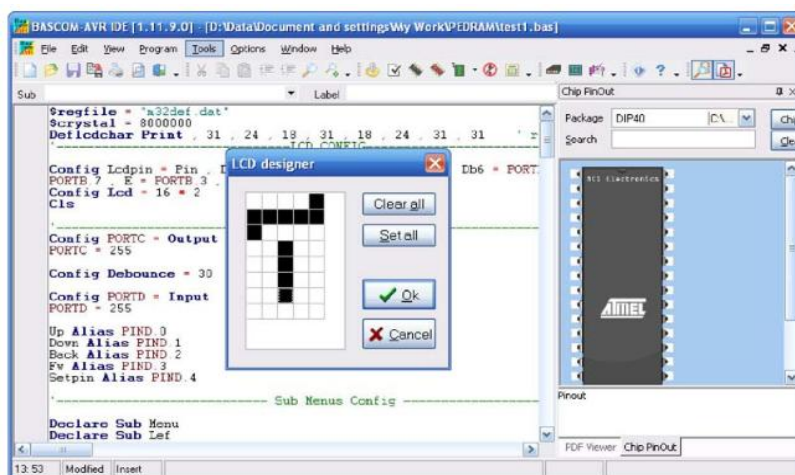
با دستور بالا مکان نما به سطر اول ستون پنجم می رود.

- **دستور Shiftlcd**: این دستور کل صفحه نمایش را یک واحد به سمت چپ یا راست منتقل می کند

Shiftlcd Left (or Right)

Ex : Shiftlcd Right

- **دستور Deflcdchar**: برای تعریف کارکتر جدید در LCD از این دستور استفاده می شود و برای تعریف متغیر جدید می توان به منوی Tools و به قسمت LCD Designer ابتدا حرف یا علامت دلخواه را طراحی نموده سپس به محض کلیک کردن بر روی دکمه OK این دستور به صورت اتومات به برنامه اضافه می شود.



برنامه نوشته شده برای این آزمایش: این برنامه پیغام Hello را در وسط سطر اول نمایش می دهد.

```
$regfile = "m32def.dat"
$crystal = 8000000
'-----LCD CONFIG-----
Config Lcdpin = Pin, Db4 = PortC.4, Db5 = PortC.5, Db6 = PortC.6, Db7 = _
PortC.7, E = PortC.1, Rs = PortC.0
Config LCD = 16 * 2
Cls
Dim A as byte
Do
    Home
    For A = 0 to 4
        ShiftCursor Right
    Next A
    LCD "Hello!"
    Waitms 500
Loop
End
```

در دو خط اول و دوم به معرفی میکرو و فرکانس کاری میکرو تعیین شده و در خط چهارم به تنظیم پایه های LCD و پایه هایی که به میکرو متصل شده می پردازیم و در این قسمت همانطور که در شکل () می بینید پایه های دیتا LCD به پورت C پایه های 4 تا 7 و پایه های کنترل LCD به پایه های 0 و 1 این پورت متصل شده است.

در خط ۶ به نوع LCD را تعیین می کنیم که در این جا از LCD ۱۶*۲ استفاده شده است.

در خط ۷ کل صفحه نمایش پاک شده و در خط بعد یک متغیر به نام A و از نوع بایت تعریف می کنیم.

در خط ۹ و ۱۶ به تعریف حلقه بی پایان می پردازیم.

در خط ۱۰ مکان نما را به سطر اول و ستون اول انتقال می دهد و در خط بعدی به تعریف یک حلقه FOR که ۵ با مکان نما را به جلومی برد . البته می توان از دستور زیر به جای حلقه FOR و دستور Shiftlcd از این دستور استفاده کرد.

Locate 1,5

در خط ۱۳ پیغام Hello! در مرکز صفحه ، نمایش داده می شود

در خط ۱۴ یک تاخیر ایجاد شده . و ادامه حلقه.

در خط ۱۶ اعلان پایان برنامه می باشد.



سوالات و پروژه های پیشنهادی

- ۱- برنامه بالا را به گونه ای تغییر دهید که اعداد زوج دو رقمی را با ۱ ثانیه تاخیر در وسط سطر اول نمایش دهد؟
- ۲- برنامه ای برای میکرو بنویسید که پیغام Hello در وسط سطر اول و نام شما را در وسط سطر دوم نمایش دهد و بعد از ۵ ثانیه به سمت چپ حرکت کند به گونه ای که اثری از پیام Hello و نام شما در صفحه نمایش باقی نماند ؟
- ۳- یک شمارنده سه رقمی طراحی کنید به گونه ای که با هر بار زدن کلید یک بار به عدد قبلی اضافه کند ؟
- ۴- سعی کنید آزمایش بالا را با تایمر / کانتر میکرو نیز آزمایش کنید ؟
- ۵- یک صفحه کلید به میکرو ببندید و مقادیر خروجی صفحه کلید را در میکرو مشاهده کنید؟

آزمایش شماره ۷

آزمایش

آشنایی با ۳۵LM و ADC میکرو و نمایش دمای محیط بروی LCD

وسایل مورد نیاز:

۱- LCD 16*2

۲- سنسور LM 35

۳- مقاومت: $10k \Omega$ و 100Ω

۴- برد مورد (bread board)

۵- منبع تغذیه

۶- سیم رابط

۷- کریستال ۸MHz

۸- ۲ عدد خازن الکترولیتی $100\mu F$ و ۱ عدد خازن $1\mu F$ و ۲ عدد خازن عدسی 22PF

۹- میکروکنترلر ATMEGA 32

۱۰- رگلاتور 7805