

# دانشگاه فنی و حرفه‌ای

## دانشکده شهید شمسی پور

پایان نامه کارشناسی

رشته فناوری ارتباطات و اطلاعات (ICT)

گرایش بهره برداری از سیستم‌های مخابراتی

## نمایش و کنترل دمای محیط در کامپیوتر با LabVIEW

نگارنده

میلاد جهاننیده (۹۲۳۰۱۳۵۴۴)

استاد راهنما

مهندس وحید بهلوری

بهمن ۱۳۹۴



## چکیده

در این پروژه هدف طراحی یک سیستم مانیتورینگ، کنترل و ثبت اطلاعات دمایی در یک بازه مشخص بوده است. یک حسگر دمای LM35 به میکروکنترلر ATMEGA8 و مدار مجتمع FT232R که یک مبدل سریال به USB و بر عکس است، متصل شده است. علت استفاده از این مدار مجتمع حذف پورت سریال در رایانه های امروزی است. کنترل کننده دما یک کنترل روشن-خاموش با باند مرده انتخاب شده است. در برنامه نوشته شده در نرم افزار LabVIEW قابلیت اجرا در دو حالت اتومات و دستی پیش بینی شده و قابلیت نمایش و ذخیره اطلاعات در یک فایل متنی بصورت پیوسته و در بازه های زمانی مشخص دیده شده است. این پروژه شامل طراحی مدارات الکترونیکی، برد مدار چاپی، طراحی برنامه نرم افزاری LabVIEW و برنامه میکروکنترلر شده و همچنین قسمت های سخت افزاری ساخته شده است. نتایج نشان دهنده عملکرد مناسب کنترل کننده و نرم افزار مانیتورینگ می باشد.

## کلمات کلیدی

کنترل دما، مانیتورینگ دما، پروتکل سریال، LabVIEW

**نکته مهم :** برنامه میکروکنترلر و LabVIEW در پیوست ها بررسی شده اند.

این فایل برای دانلود و مطالعه عموم در وبسایت **Melec.ir** منتشر شده است.

## فهرست مطالب

۱	نمایش و کنترل دمای محیط در کامپیوتر با LabVIEW
۵	فصل ۱ مقدمه
۵	۱-۱- پیشگفتار
۵	۲-۱- بیان مساله
۵	۳-۱- فرضیات مساله
۶	۴-۱- اهداف تحقیق
۷	فصل ۲ طراحی سیستم کنترل دما
۷	۱-۲- مقدمه طراحی سیستم کنترل دما
۸	۲-۲- مفهوم باند مرده
۸	۳-۲- انتخاب سنسور دما
۱۰	۴-۲- انتخاب میکروکنترلر
۱۱	۵-۲- نصب سنسور دمای LM35 به میکروکنترلر
۱۲	۶-۲- رله های سیستم کنترل
۱۲	۷-۲- اتصال نمایشگر روی پروژه
۱۳	۸-۲- پایه های LCD کارکتری
۱۴	۹-۲- اتصالات نهایی روی میکروکنترلر
۱۵	فصل ۳ طراحی سیستم مانیتورینگ دما
۱۵	۱-۳- شروع طراحی
۱۶	۲-۳- آرسی مبدل USB به سریال
۱۶	۳-۳- بسته نرم افزاری VISA
۱۷	۴-۳- شماتیک آرسی FT232R
۱۸	فصل ۴ بررسی نتایج و پیشنهادات
۱۸	۱-۴- بررسی نتایج
۱۸	۲-۴- پیشنهادات
۱۹	فهرست مراجع
۲۰	پیوست أ
۴۱	The ambient temperature monitoring and control in LabVIEW software

## فهرست اشکال

۷.....	شکل (۱-۲) بلوک دیاگرام سخت افزار پروژه
۸.....	شکل (۲-۲) عملکرد باند مرده
۹.....	شکل (۳-۲) پایه های سنسور LM35
۱۰.....	شکل (۴-۲) میکروکنترلر AVR
۱۱.....	شکل (۵-۲) اتصال سنسور دما به میکروکنترلر
۱۲.....	شکل (۶-۲) دو عدد رله برای فن و هیتر برقی
۱۲.....	شکل (۷-۲) پایه های LCD
۱۳.....	شکل (۸-۲) پتانسیومتر برای تغییر نور نمایشگر
۱۴.....	شکل (۹-۲) اتصالات نهایی میکروکنترلر
۱۵.....	شکل (۱-۳) بلوک کلی پروژه
۱۶.....	شکل (۲-۳) آرسی FT232RL
۱۶.....	شکل (۳-۳) حالت کلی یک برنامه اینترفیس با لب ویو
۱۷.....	شکل (۴-۳) شماتیک اتصال مبدل به میکروکنترلر
۱۷.....	شکل (۵-۳) شبیه سازی در پروتیوس
۲۴.....	شکل (۱-۴) محیط کدویژن
۲۴.....	شکل (۲-۴) ایجاد پروژه
۲۴.....	شکل (۳-۴) استفاده از کدویزارد
۲۵.....	شکل (۴-۴) نوع میکروکنترلر
۲۵.....	شکل (۵-۴) محیط کدویزارد
۲۶.....	شکل (۶-۴) انتخاب چیپ
۲۶.....	شکل (۷-۴) تعیین پایه ها به عنوان ورودی و خروجی
۲۷.....	شکل (۸-۴) تولید کدها
۲۷.....	شکل (۹-۴) ذخیره برنامه
۳۳.....	شکل (۱۰-۴) برنامه پروگرامر
۳۴.....	شکل (۱۱-۴) تنظیم فیوز بیت ها
۳۴.....	شکل (۱۲-۴) برنامه نهایی لب ویو
۳۵.....	شکل (۱۳-۴) برنامه نویسی لب ویو
۳۵.....	شکل (۱۴-۴) فرمت بندی اطلاعات
۳۶.....	شکل (۱۵-۴) بلوک های پورت سریال
۳۶.....	شکل (۱۶-۴) قسمت کنترل کننده

شکل (۴-۱۷) نوشته برای نمایش روی سخت افزار ..... ۳۷

شکل (۴-۱۸) ذخیره دما و تاریخ در یک فایل متنی ..... ۳۷

شکل (۴-۱۹) سخت افزار نهایی پروژه ..... ۳۹

## فهرست جداول

جدول (۱-۲) نام پایه ها و عملکرد هر پایه LCD.....۱۳

## فصل ۱ مقدمه

### ۱-۱- پیشگفتار

هدف در این پروژه طراحی سیستم نمایش دمای محیط در کامپیوتر و همچنین کنترل دمای محیط است. سیستمی که طراحی شد، دارای یک سخت افزار است که روی آن سنسور برای اندازه گیری دما و دو عدد رله برای روشن و خاموش کردن هیتر و تهویه تعبیه شده است. همچنین برای ارتباط با کامپیوتر باید از یک عدد آیسی مبدل سریال به USB استفاده شده است. برای نمایش وضعیت رله ها و همچنین دما، روی خود سخت افزار نیز نمایشگری طراحی شده است و همه این قسمت ها توسط میکروکنترلر کنترل می شود. قسمت دوم نرم افزار کامپیوتری می باشد که با زبان برنامه نویسی گرافیکی و با برنامه LabVIEW نوشته شده است. در قسمت نرم افزار کامپیوتری دما از پورت سریال دریافت و پردازش می شود. و در نهایت در یک فایل هر چند ثانیه یکبار ذخیره می شود. برای کنترل دما از کنترلر باند مرده استفاده خواهیم کرد.

### ۱-۲- بیان مساله

در بسیاری از محیطها بدلیل عدم وجود کنترلر دمای محیط، مقدار زیادی انرژی از دست می رود و همچنین در بسیاری از محیطها مانند اتاق های کودک، گلخانه ها و دامداری ها و... نیاز است که دما در یک بازه مشخصی باشد و تغییرات دما خسارات مالی و حتی جانی می تواند داشته باشد. پس کنترل و نگهداری دما در یک بازه مشخص می تواند در صرفه جویی انرژی کمک و همچنین از به بار آمدن خسارات جلوگیری کند.

### ۱-۳- فرضیات مساله

در طراحی مانیتورینگ و کنترل دمای محیط چندتا فرض را برای طراحی در نظر می گیریم که سخت افزار و نرم افزار پروژه بنا به این فرضیات پیاده سازی شده است. فرض اول این است که تغییرات دمای محیطی که می خواهیم این کنترل کننده نصب شود در حد تغییرات محیط زندگی انسان می باشد. پس در انتخاب قطعات و دمای کاری آنها با این فرض انتخاب خواهند شد. فرض دوم این است که یک تهویه هوا یا فن در محیط وجود دارد که اگر دما بیشتر از دمای محیط شد با این تهویه دما را پایین آورده شود. فرض سوم وجود یک هیتر یا بخاری برقی است. در صورت سرد شدن محیط با این هیتر دما را گرم کند.



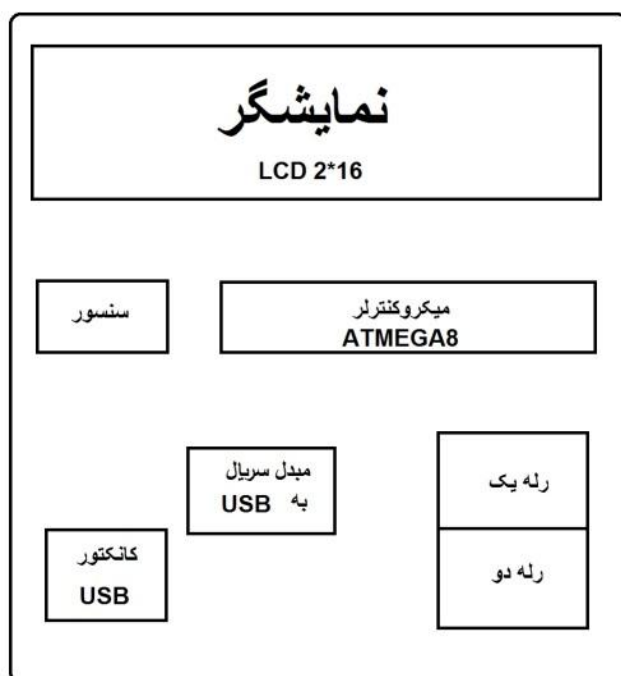
#### ۴-۱- اهداف تحقیق

طراحی و ساخت یک سیستم سخت افزاری و نرم افزاری مانیتورینگ و کنترل دمای محیط با کنترلر باند مرده و نرم افزار کامپیوتری می باشد. پروژه نهایی یک سخت افزار که سنسور و رله های هیتر و فن روی آن نصب هستند. نرم افزار نهایی یک نرم افزار با LabVIEW که کنترل کننده باند مرده روی آن پیاده شده و همچنین دما بصورت پیوسته در داخل یک فایل ذخیره می شود است.

## فصل ۲ طراحی سیستم کنترل دما

### ۱-۲- مقدمه طراحی سیستم کنترل دما

این پروژه در حقیقت از دو قسمت کلی برد میکروکنترلری و برنامه کامپیوتری تشکیل شده است. در قسمت برد میکروکنترلری یک عدد سنسور دما، رله، نمایشگر، میکروکنترلر و مبدل پورت سریال به USB است. در قسمت کامپیوتر یک برنامه توسط نرم افزار نوشته شده است. در هر دو طرف پروتکل ها و استانداردها پیاده سازی می شود.

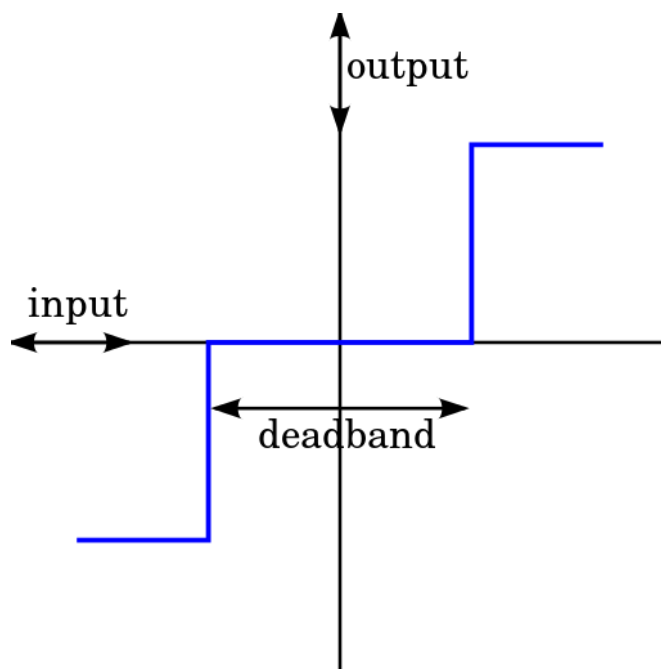


شکل (۱-۲) بلوک دیاگرام سخت افزار پروژه

## ۲-۲- مفهوم باند مرده

در فرآیندهای ابزار دقیق، گستره (بازه) ای که در آن سیگنال ورودی با تغییر جهت می‌تواند تغییر کند، بدون آن که منجر به تغییر محسوس یا قابل مشاهده‌ای در سیگنال خروجی شود را باند سکوت می‌گویند.

برای درک بهتر مفهوم باند سکوت می‌توان از مثال ساده کنترل دمای اتاق با استفاده از ترموستات استفاده کرد. وقتی اتاق سردتر از دمای تنظیم شده باشد، ترموستات عمل کرده و سیستم گرمایش روشن و اتاق شروع به گرم شدن می‌کند. این گرم شدن تا جایی ادامه پیدا می‌کند که دما از میزان باند سکوت ترموستات بیشتر شده و ترموستات قطع کرده و سیستم گرمایش خاموش شود. سپس اتاق شروع به سرد شدن می‌کند تا دوباره ترموستات عمل کند. این چرخه مدام ادامه دارد و دمای اتاق با دامنه تغییراتی برابر با باند سکوت ترموستات تثبیت می‌شود.



شکل (۲-۲) عملکرد باند مرده

باند سکوت باعث می‌شود اختلاف فاز تاخیری بین سیگنال ورودی و سیگنال خروجی به وجود بیاید به طوری که با وجود تغییر سیگنال ورودی در جهت افزایشی یا کاهشی، سیگنال خروجی ثابت می‌ماند و پس از افزایش سیگنال ورودی به اندازه بیش از مقدار باند سکوت، سیگنال خروجی شروع به تغییر می‌کند.

## ۲-۳- انتخاب سنسور دما

برای اندازه‌گیری کمیت دما ما از سنسور LM35 استفاده می‌کنیم و همچنین محدوده اندازه‌گیری را مثبت و از صفر تا ۹۹ درجه سانتی‌گراد انتخاب می‌کنیم.



## ۲-۳-۳- مشخصات سنسور دما LM35

خروجی: ۱۰ میلی ولت به ازای هر درجه سانتی گراد

رنج اندازه گیری دما: ۵۵- تا ۱۵۰ درجه

اندازه گیری دما بر حسب: درجه سانتی گراد (سلسیوس)

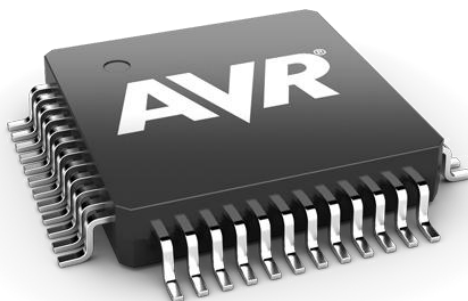
دقت اندازه گیری: ۰.۰۵ درجه در دمای ۲۵

رنج ولتاژ کاری: ۴ تا ۳۰ ولت

جریان مصرفی: ۶۰ میکرو آمپر

## ۲-۴- انتخاب میکروکنترلر

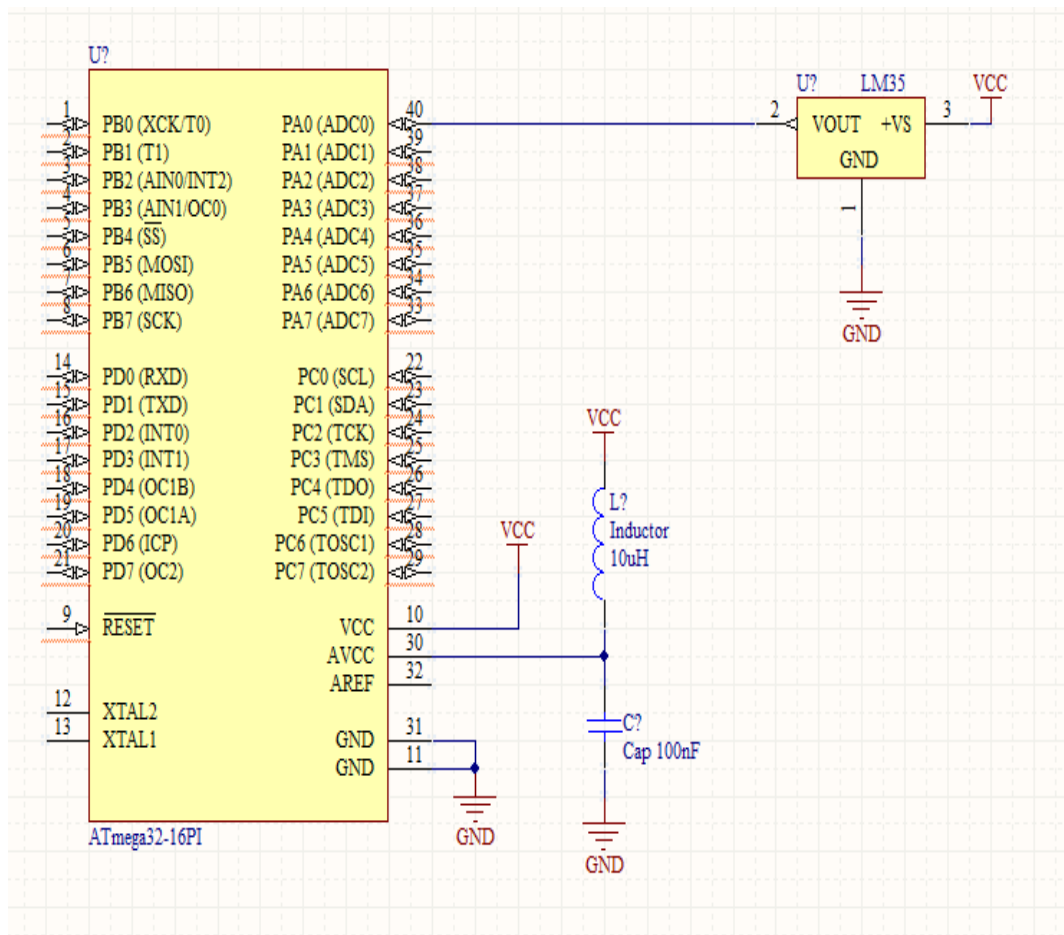
از یک میکروکنترلر AVR به عنوان پردازنده استفاده شده است. چون اکثر میکروکنترلرهای AVR دارای مبدل آنالوگ به دیجیتال داخلی می‌باشند (میکروکنترلر انتخابی پروژه ۸ مگا می‌باشد) از مبدل خود میکروکنترلر برای تبدیل خروجی ولتاژ سنسور به مقدار عددی استفاده می‌کنیم. جزئیات قسمت تبدیل آنالوگ به دیجیتال و پردازش داده‌ها مانند تابع تبدیل سنسور، رزولیشن مبدل و قالب بندی برای ارسال در قسمت برنامه نویسی میکروکنترلر بررسی شده است.



شکل (۲-۴) میکروکنترلر AVR

## ۲-۵- نصب سنسور دمای LM35 به میکروکنترلر

برای شروع سنسور LM35 را به میکروکنترلر وصل شده است برای اینکار یکی از هشت کانال ADC موجود بر روی میکروکنترلر یکی را انتخاب و خروجی سنسور مستقیم وصل می‌شود.

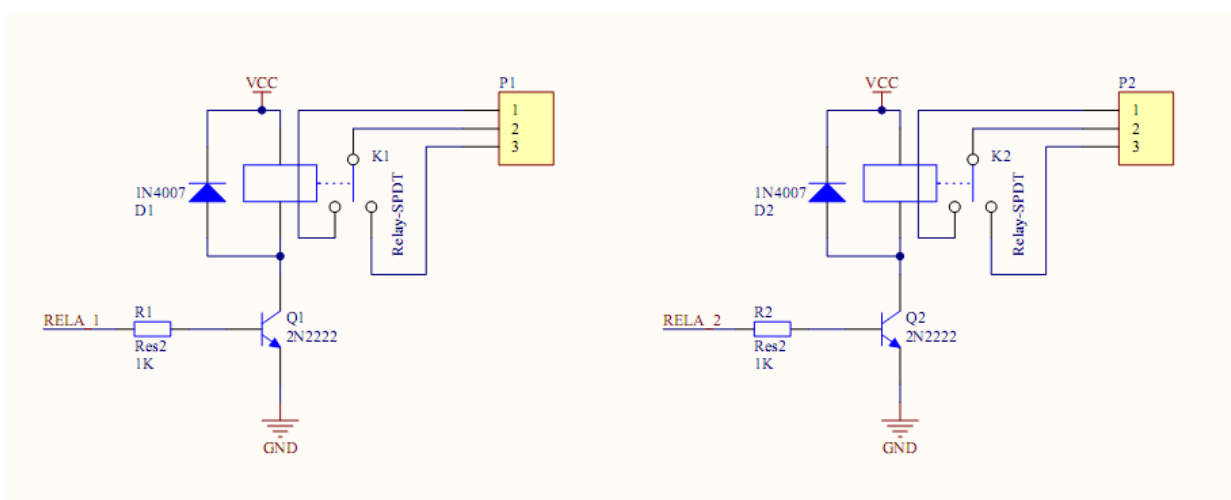


شکل (۲-۵) اتصال سنسور دما به میکروکنترلر

تغذیه مبدل آنالوگ به دیجیتال میکروکنترلر و همچنین تغذیه خود میکروکنترلر باید وصل شوند برای جلوگیری از نویز گرفتن ADC میکروکنترلر از یک سلف ۱۰ میکروهنری و یک خازن ۱۰۰ نانوفاراد استفاده شده است. حال مبدل USB به پورت سریال به میکروکنترلر وصل می‌شود.

## ۶-۲- رله های سیستم کنترل

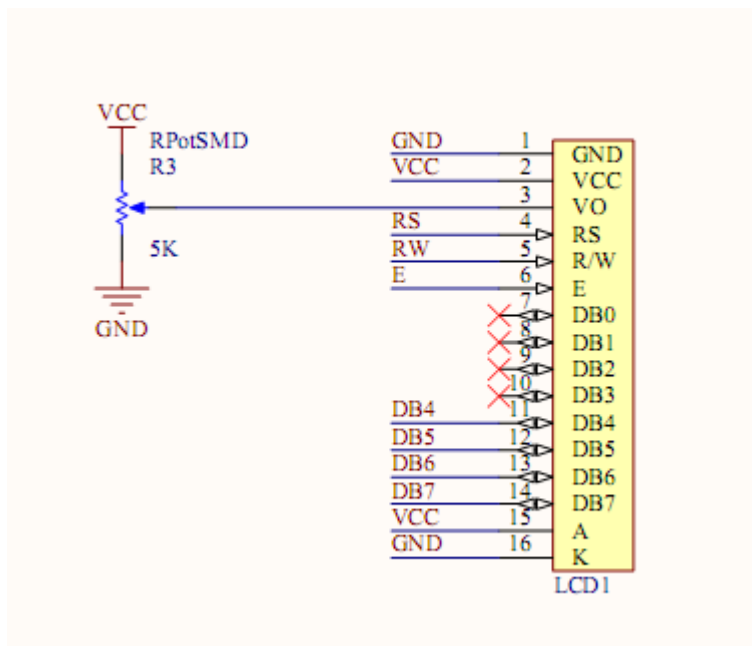
دو عدد رله یکی برای هیتر برقی و دیگری برای فن یا تهویه برقی می باشد. در این سیستم از رله های میلون استفاده شده است. بوبین این رله ها با ۵ ولت تحریک می شود. دیود به عنوان هرزگرد انتخاب شده است. ترانزیستورها هم برای تأمین جریان بوبین رله بصورت سوئیچ استفاده شده اند.



شکل (۶-۲) دو عدد رله برای فن و هیتر برقی

## ۷-۲- اتصال نمایشگر روی پروژه

برای اتصال LCD به میکروکنترلر ابتدا پایه های LCD 2\*16 را بررسی می شوند و همچنین یک عدد پتانسیومتر برای تنظیم نور نمایشگر LCD استفاده شده است.



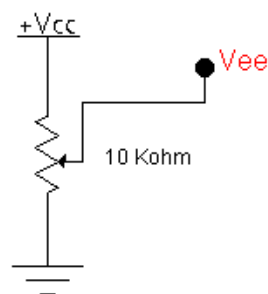
شکل (۷-۲) پایه های LCD

## ۸-۲- پایه‌های LCD کارکتری

جدول (۱-۲) نام پایه‌ها و عملکرد هر پایه LCD

شماره پایه	نام پایه	عملکرد
1	Vss	زمین، GND
2	Vcc	تغذیه مثبت، 5v
3	Vee	تنظیم نور کاراکترها (کنتراست)
4	RS	اگر RS=0 باشد مقدار ورودی به عنوان یک دستور می‌باشد اما اگر RS=1 باشد مقدار ورودی یک داده برای چاپ شدن است.
5	R/W	اگر چیزی در LCD نوشته شود این پایه باید صفر باشد و اگر نوشته‌ای از LCD مقدارش خوانده شود باید آن باید یک شود.
6	E	پس از انجام هر عملیات ارسال یا دریافت باید پایه‌ی E را یکبار صفر و یکبار یک کرد تا اطلاعات ثبت شوند.
7 - 14	DB <sub>0</sub> - DB <sub>7</sub>	مسیر ورود و خروج اطلاعات LCD
15	Anod	تغذیه‌ی مثبت چراغ LCD
16	Katod	تغذیه‌ی منفی چراغ LCD

پایه‌ی Vee می‌تواند به صورت زیر برای کم و زیاد کردن نور نوشته‌ها به کار رود. اگر سر فلش به طرف منفی باشد، بیشترین نور، و اگر به طرف مثبت باشد، کمترین نور را خواهد داشت.

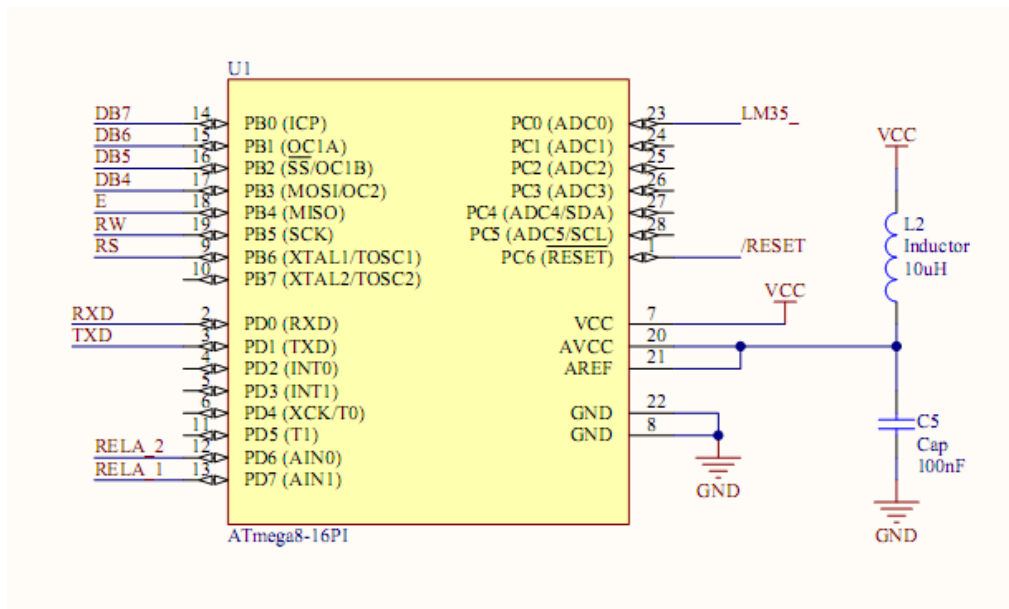


شکل (۸-۲) پتانسیومتر برای تغییر نور نمایشگر



## ۹-۲- اتصالات نهایی روی میکروکنترلر

بعد از اتصال LCD و سنسور دما و پایه‌هایی تغذیه و تغذیه قسمت مبدل آنالوگ به دیجیتال و همچنین رله نوبت به اتصال مبدل سریال به USB می‌رسد که در فصل بعدی بررسی می‌شود.



شکل (۹-۲) اتصالات نهایی میکروکنترلر

## فصل ۳ طراحی سیستم مانیتورینگ دما

### ۳-۱- شروع طراحی

برای نمایش دما روی کامپیوتر به سه قسمت نیاز داریم قسمت اول سخت افزار هست که داری حسگر و مبدل هست و بعد از حس دما روی برد میکروکنترلی تعبیه شده است که وظیفه آن خواندن مقدار سنسور و ارسال آن به کامپیوتر می باشد برای ارسال داده ها به کامپیوتر به مدار اینترفیس نیاز هست که قسمت دوم پروژه را تشکیل می دهد و در آخر داده های دریافتی توسط پورت کامپیوتر باید نمایش داده شوند که برای اینکار از نرم افزار لب ویو که یک محیط برنامه نویسی گرافیکی و پیشرفته است استفاده می شود.



شکل (۳-۱) بلوک کلی پروژه

عملکرد کلی این برد بدین صورت است که دما را با یک سنسور دما اندازه گیری کند و آن را بعد از پردازش به کامپیوتر ارسال کند. نکته مهم برای اینکه نرم افزار نوشته شده برای کامپیوتر یونیورسال یا به عبارتی انعطاف پذیر باشد و باز طراحی سخت افزار باعث بازطراحی نرم افزار نشود تابع تبدیل دما و پردازش های دیگر را در میکروکنترلر انجام می شود و فقط کار دریافت داده از پورت سریال و نمایش آن به کامپیوتر سپرده می شود.

### ۲-۳- آیسوی مبدل USB به سریال

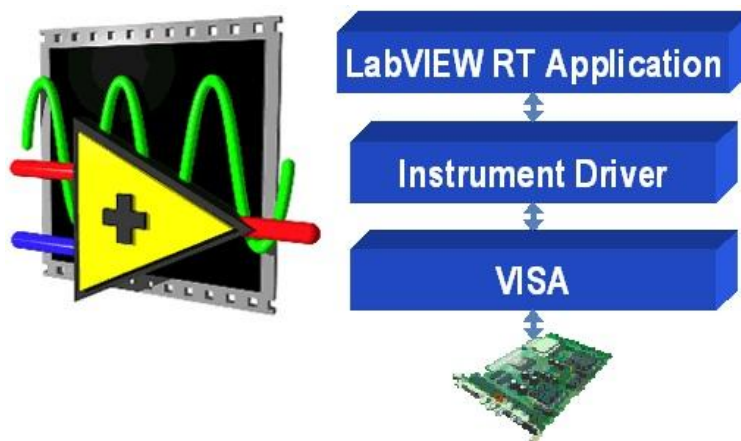
برای قسمت اینترفیس از پورت سخت افزاری USB کامپیوتر استفاده شده است. برای برنامه نویسی بصورت مجازی در هر دو طرف از پورت کام استفاده شده است. بطور ساده از یک مبدل USB به COM استفاده می شود. برای اینکار از آی سی مبدل های موجود در بازار مانند FT232RL استفاده می شود.



شکل (۲-۳) آیسوی FT232RL

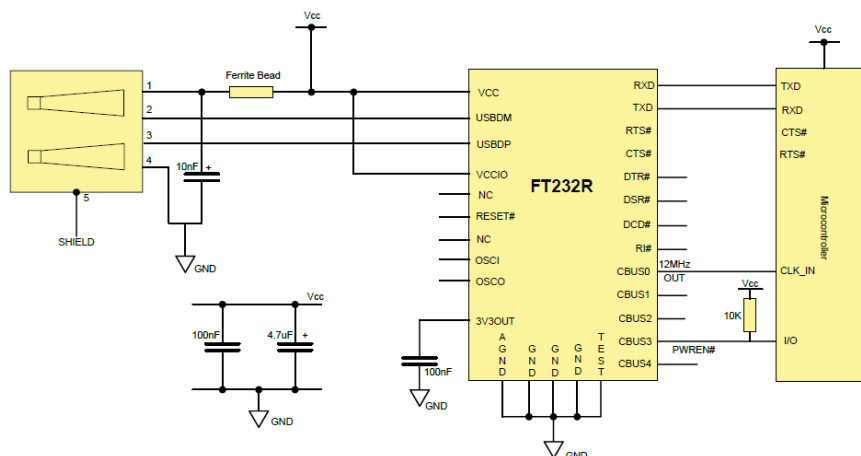
### ۳-۳- بسته نرم افزاری VISA

حال نوبت برنامه نویسی قسمت کامپیوتر می باشد که برای اینکار از نرم افزار لب ویو استفاده می شود. البته باید یک بسته نرم افزاری دیگر که حاوی کتابخانه های ارتباط سخت افزاری لب ویو می باشد نیز روی سیستم نصب باشد. در این مرحله اول نرم افزار لب ویو نصب و بعد از آن VISA نصب می شود.



شکل (۳-۳) حالت کلی یک برنامه اینترفیس با لب ویو

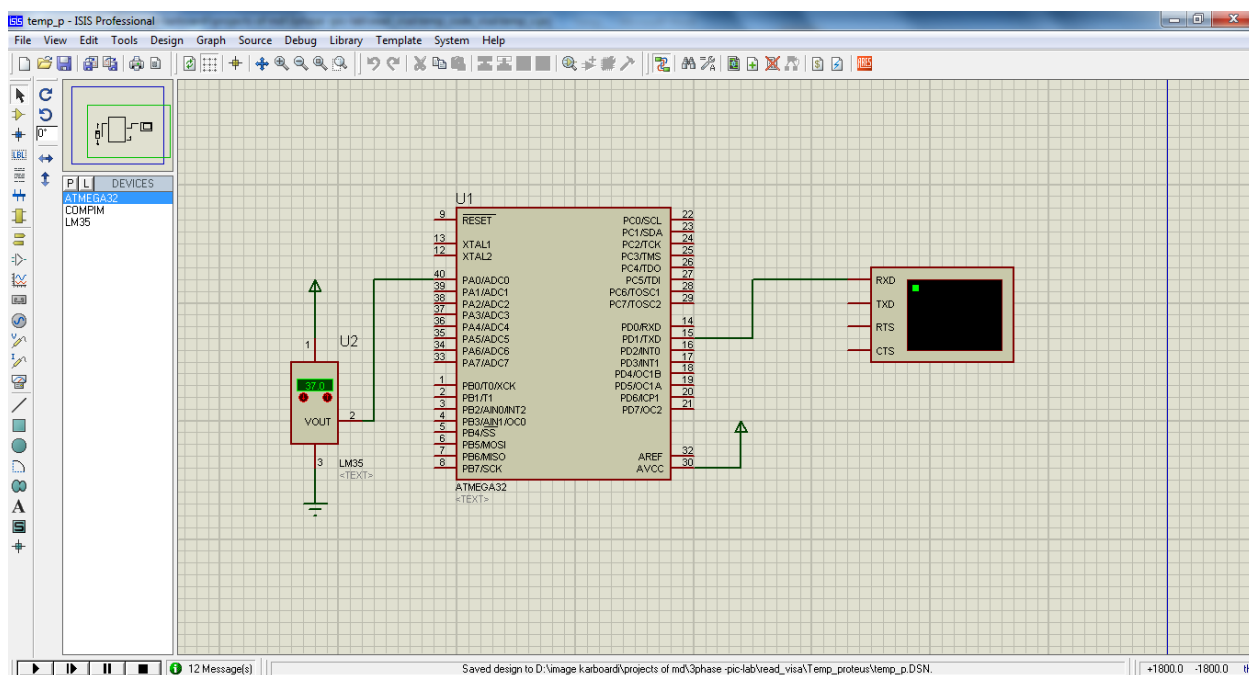
### ۴-۳- شماتیک آیسی FT232R



شکل (۴-۳) شماتیک اتصال مبدل به میکروکنترلر

دراپور این آی سی را می توان از سایت سازنده آن برای سیستم عامل مورد نیاز خود دانلود کرد.

<http://www.ftdichip.com>



شکل (۵-۳) شبیه سازی در پروتیوس

برای شبیه سازی از نرم افزار پروتیوس استفاده می شود و قطعات میکروکنترلر AVR و LM35 را که قطعات اصلی پروژه هستند به محیط کار آورده می شوند و برای شبیه سازی ارسال اطلاعات از ترمینال مجازی استفاده می شود. در طراحی مدار نهایی از میکروکنترلر ATMEGA8A استفاده شده است.

## فصل ۴ بررسی نتایج و پیشنهادات

### ۴-۱- بررسی نتایج

نتیجه نهایی پروژه یک کنترل کننده دمای محیط با کامپیوتر می باشد. در این پروژه، دما توسط سنسور دما اندازه گیری و بعد با مبدل آنالوگ به دیجیتال میکروکنترلر دما بصورت ولتاژ دریافت و با تابع تبدیل به مقدار درجه سانتی گراد تبدیل می شود و بعد در نمایشگر روی پروژه نمایش و همچنین به کامپیوتر ارسال می شود. در کامپیوتر دما دریافت و اگر حالت اتوماتیک فعال باشد با بازه های حداقل و حداکثر مقایسه می شود. از روی مقایسه انجام شده سیگنال کنترلی رله ها تولید و به سمت میکروکنترلر ارسال می شود. همچنین در LabVIEW عمل نمایش دما و ذخیره سازی در یک فایل متنی نیز انجام می شود.

### ۴-۲- پیشنهادات

برای نسل های آینده پروژه پیشنهاد می شود دما تحت پروتکل HTTP و TCP/IP به کامپیوتر منتقل شود و همچنین از میکروکنترلر های دیگری مانند ARM یا PIC استفاده شود و همچنین ارتباط با موبایل از طریق WIFI و یا Bluetooth می تواند پروژه را جالبتر کند. همچنین پروژه برای محیط های صنعتی هم می تواند بهینه شود. مثلاً از سنسوری با بازه دمایی و دقت بالا استفاده شود. و المان های برد نیز باید بازه دمایی گسترده ای را داشته باشند.

## فهرست مراجع

- [1] Mazidi, Muhammad Ali, Sarmad Naimi, and Sepehr Naimi. AVR microcontroller and embedded systems: using assembly and C. Prentice Hall Press, 2010.
- [2] Kernighan, Brian W., and Dennis M. Ritchie. The C programming language. Vol. 2. Englewood Cliffs: prentice-Hall, 1988.
- [3] Beyon, Jeffery Y. LabVIEW programming, data acquisition and analysis. Prentice Hall PTR, 2000.
- [4] Chip, F. T. D. I. "FT232R-USB UART IC." Future Technology Devices International Limited. DataSheets. Web 15 (2012).
- [5] Sheet, LM35 Data. "National Semiconductor Corporation." (2000).
- [6] Atmel, AVR. "ATmega8 Datasheet."

برای دانلود پروژه های الکترونیکی بیشتر به وبسایت ما به آدرس [Melec.ir](http://Melec.ir) مراجعه فرمائید.

## پیوست أ

### أ-۱- میکروکنترلر پروژه

کلمه میکروکنترلر از دو کلمه میکرو و کنترلر تشکیل شده است.

\*میکرو یک واحد یونانی است و برابر با ۱۰ به توان منفی ۶ متر است.

\*کنترلر یعنی کنترل کننده به تعبیری یعنی "مغز" البته بدون تفکر فقط دستوراتی که به اون داده می شود را به نحو احسن انجام می دهد.

### أ-۲- تفاوت میکروپروسسور و میکروکنترلر

میکروپروسسور یک پردازنده است و برای کار باید به آن چیپ های حافظه و چیزهای دیگری را به اون اضافه کرد این امکان به دلیل است که حافظه مناسب و دیگر قطعات مانند تایمر و... بر حسب کار طراحی شوند ولی مدار خیلی پیچیده می شود و از لحاظ هزینه هم هزینه بیشتر می شود به همین دلیل امروزه از میکروپروسسورها در مدارهای کنترلی کمتر استفاده می شود.

### أ-۳- بخش های مختلف میکروکنترلر

CPU واحد پردازش

ALU واحد محاسبات

I/O ورودی ها و خروجی ها

RAM حافظه موقت میکروکنترلر

ROM حافظه نگهداری برنامه

Timer برای ایجاد زمان های مختلف

برای دانلود پروژه های الکترونیکی بیشتر به وبسایت ما به آدرس [Melec.ir](http://Melec.ir) مراجعه فرمائید.

#### أ-۴- برنامه ریزی میکروکنترلر

میکروکنترلرها دارای کامپایلرهای خاصی می باشند که با زبان های basic, Assembly و C می توان برای آنها برنامه نوشت سپس برنامه نوشته شد توسط دستگاهی به نام Programmer برنامه ریزی میشود. در این دستگاه آرسی قرار می گیرد و توسط یک کابل به یکی از درگاه های کامپیوتر وصل می شود و برنامه نوشته شده روی آرسی منتقل و ذخیره می شود.

#### أ-۵- کاربردهای میکروکنترلر

این آی سی ها حکم یک کامپیوتر در ابعاد کوچک و قدرت کمتر را دارند بیشتر این آی سی ها برای کنترل و تصمیم گیری استفاده می شوند چون طبق الگوریتم برنامه ی آن عمل می کند این آی سی ها از کنترل ربات ها تا استفاده در کارخانه صنعتی کار برد دارند.

#### أ-۶- مقایسه خانواده های مختلف میکروکنترلرها

#### أ-۷- خانواده AVR

میکروهای AVR دارای انعطاف پذیری غیر قابل مقایسه و بی همتایی هستند. آنها قادر به ترکیب هر نوع کدی با یک معماری کارآمد از طریق زبانهای C و Assembly هستند و قادرند از طریق این برنامه ها تمام پارامترهای ممکن در یک سیکل یا چرخه ماشین را با دقت بسیار بالا اجرا کنند.

#### أ-۸- ویژگی های کلیدی AVR

- ❖ دارای سیستمی با بهترین هماهنگی
- ❖ دارای بالاترین کارایی و اجرا در CPU (یک دستورالعمل در هر سیکل کلاک)
- ❖ دارای کدهایی با کوچکترین سایز
- ❖ دارای حافظه خود برنامه ریز
- ❖ دارای واسطه JTAG که با IEEE 1149.1 سازگار است.
- ❖ دارای سخت افزار ضرب کننده روی خود پردازنده
- ❖ دارای بهترین ابزارها برای پیشرفت و ترقی
- ❖ دارای حالات زیادی برای ترفیع دادن
- ❖ میکرو کنترلر AVR به منظور اجرای دستورالعمل های قدرتمند در یک سیکل کلاک ساعت به اندازه کافی سریع است و می تواند آزادی عملی احتیاج به منظور بهینه سازی توان مصرفی را فراهم کند.
- ❖ میکروکنترلر AVR بر مبنای معماری RISC (کاهش مجموعه ی دستورالعمل های کامپیوتر) پایه گذاری شده و مجموعه ای از دستورالعمل ها را که با ۳۲ ثبات کار می کنند ترکیب می کند.
- ❖ به کارگرفتن حافظه از نوع Flash که AVR ها به طور یکسان از آن بهره می برند از جمله مزایای آنها است.
- ❖ یک میکرو AVR می تواند با استفاده از یک منبع تغذیه 2.7 تا 5.5 ولتی و از طریق شش پین ساده در عرض چند ثانیه برنامه ریزی شود یا Program شود.

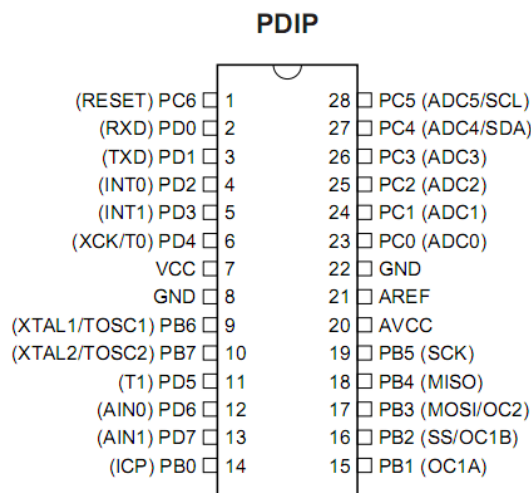


برای دانلود پروژه های الکترونیکی بیشتر به وبسایت ما به آدرس [Melec.ir](http://Melec.ir) مراجعه فرمائید.

❖ میکروهای AVR در هر جا که باشند با 1.8 ولت تا 5.5 ولت تغذیه می‌شوند.

#### آ-۹- مشخصات سخت افزاری میکروکنترلر ATMEGA8A

شکل ظاهری و پایه‌ها : ATMEGA8A در سه نوع بسته بندی PDIP با ۲۸ پایه و TQFP با ۳۲ پایه و MLF با ۳۲ پایه ساخته شده است.



شکل(آ-۱) پایه‌های میکروکنترلر ATMEGA8

#### آ-۱۰- ویژگی‌های اصلی میکروکنترلر ATMEGA8A

##### Features

- High-performance, Low-power AVR® 8-bit Microcontroller
- Advanced RISC Architecture
  - 130 Powerful Instructions - Most Single-clock Cycle Execution
  - 32 x 8 General Purpose Working Registers
  - Fully Static Operation
  - Up to 16 MIPS Throughput at 16 MHz
  - On-chip 2-cycle Multiplier
- Nonvolatile Program and Data Memories
  - 8K Bytes of In-System Self-Programmable Flash
  - Endurance: 10,000 Write/Erase Cycles
  - Optional Boot Code Section with Independent Lock Bits
- In-System Programming by On-chip Boot Program
- True Read-While-Write Operation
  - 512 Bytes EEPROM

برای دانلود پروژه های الکترونیکی بیشتر به وبسایت ما به آدرس [Melec.ir](http://Melec.ir) مراجعه فرمائید.

Endurance: 100,000 Write/Erase Cycles

- 1K Byte Internal SRAM
- Programming Lock for Software Security
- Peripheral Features
- Two 8-bit Timer/Counters with Separate Prescaler, one Compare Mode
- One 16-bit Timer/Counter with Separate Prescaler, Compare Mode, and Capture

Mode

- Real Time Counter with Separate Oscillator
- Three PWM Channels
- 8-channel ADC in TQFP and MLF package

Six Channels 10-bit Accuracy

Two Channels 8-bit Accuracy

- 6-channel ADC in PDIP package

Four Channels 10-bit Accuracy

Two Channels 8-bit Accuracy

- Byte-oriented Two-wire Serial Interface
- Programmable Serial USART
- Master/Slave SPI Serial Interface
- Programmable Watchdog Timer with Separate On-chip Oscillator
- On-chip Analog Comparator
- Special Microcontroller Features
- Power-on Reset and Programmable Brown-out Detection
- Internal Calibrated RC Oscillator
- External and Internal Interrupt Sources
- Five Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power-down, and Standby

• I/O and Packages

- 23 Programmable I/O Lines
- 28-lead PDIP, 32-lead TQFP, and 32-pad MLF

• Operating Voltages

- 2.7 - 5.5V (ATmega8L)
- 4.5 - 5.5V (ATmega8)

• Speed Grades

- 0 - 8 MHz (ATmega8L)
- 0 - 16 MHz (ATmega8)

• Power Consumption at 4 Mhz, 3V, 25°C

- Active: 3.6 mA

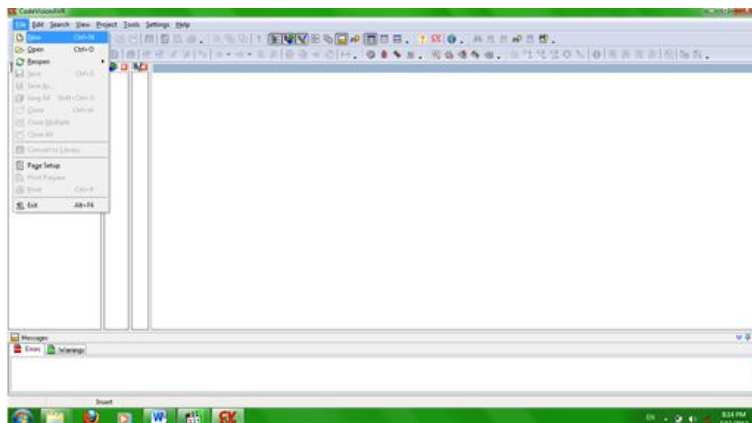
برای دانلود پروژه های الکترونیکی بیشتر به وبسایت ما به آدرس [Melec.ir](http://Melec.ir) مراجعه فرمائید.

- Idle Mode: 1.0 mA

- Power-down Mode: 0.5  $\mu$ A

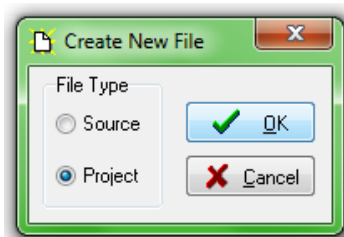
## أ-۱۱- کار با کامپایلر کدویژن

برای برنامه نویسی میکروکنترلرهای AVR کامپایلرهای مختلفی وجود دارد. در این پروژه از کدویژن استفاده شده است. نمایی از محیط نرم افزار را در شکل زیر مشاهده می شود.



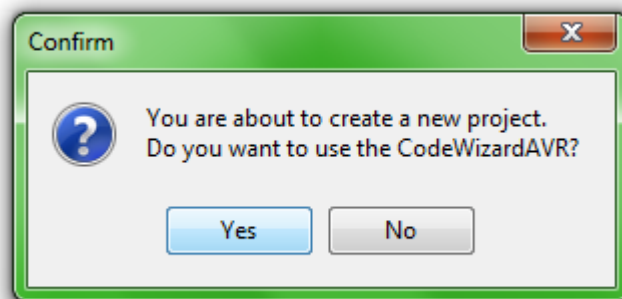
شکل (۱-۴) محیط کدویژن

برای شروع کار با کامپایلر، ابتدا باید پروژه تعریف شود برای اینکار از منوی فایل NEW را انتخاب می شود.



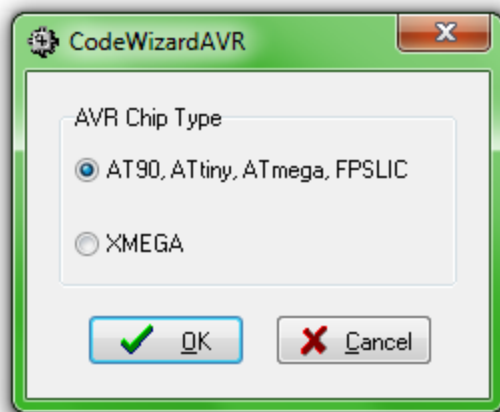
شکل (۲-۴) ایجاد پروژه

سپس در پنجره باز شده project انتخاب می شود.



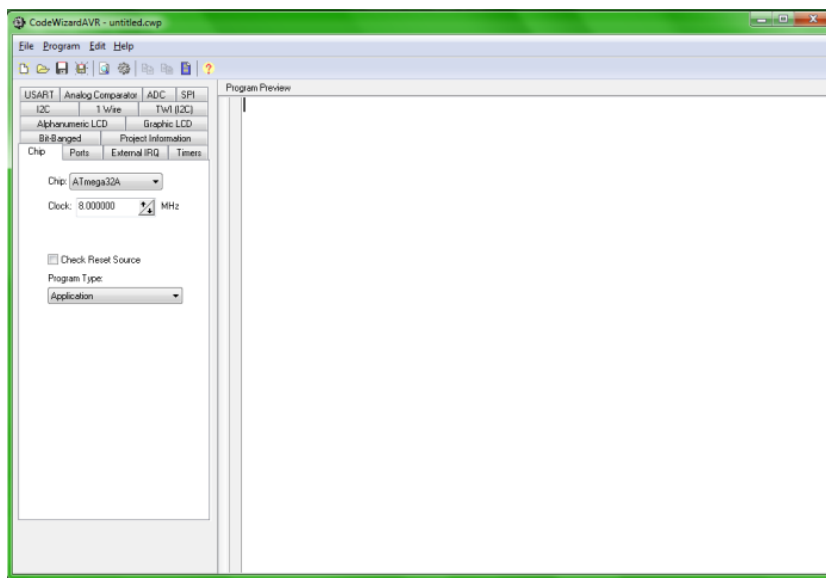
شکل (۳-۴) استفاده از کدویزارد

برای دانلود پروژه های الکترونیکی بیشتر به وبسایت ما به آدرس [Melec.ir](http://Melec.ir) مراجعه فرمائید.



شکل (۴-۴) نوع میکروکنترلر

پنجره‌ای باز می‌شود که در آن از کاربر سؤال می‌کند که آیا می‌خواهید از کدویزارد برنامه استفاده یا نه؟ که در این پروژه از کدویزارد استفاده می‌شود. چون ورژن جدید کامپایلر از سری جدید AVR یعنی XMEGA هم پشتیبانی می‌کند در مرحله بعد باید نوع خانواده میکروکنترلر انتخاب شود.



شکل (۴-۵) محیط کدویزارد

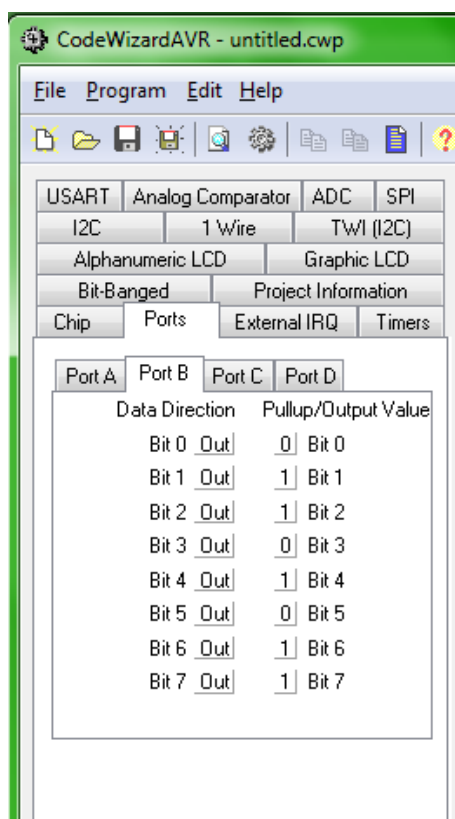
بعد اینکه OK زده شد برنامه کد ویزارد باز می‌شود. در این پنجره نوع میکروکنترلر، پالس ساعت، رجیسترها و قسمت‌های مختلف میکروکنترلر تنظیم می‌شوند.

برای دانلود پروژه های الکترونیکی بیشتر به وبسایت ما به آدرس [Melec.ir](http://Melec.ir) مراجعه فرمائید.



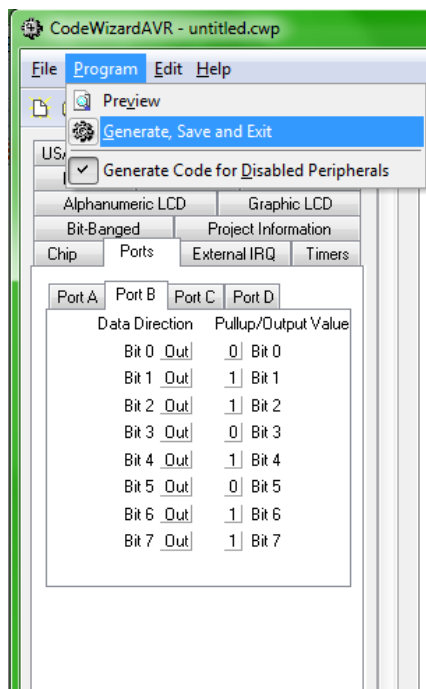
شکل (۴-۶) انتخاب چیپ

از زبانه chip نوع میکروکنترلر و فرکانس کاری آن را انتخاب می شود. طبق شکل زیر پایه ها را بصورت ورودی و خروجی تنظیم می شوند.



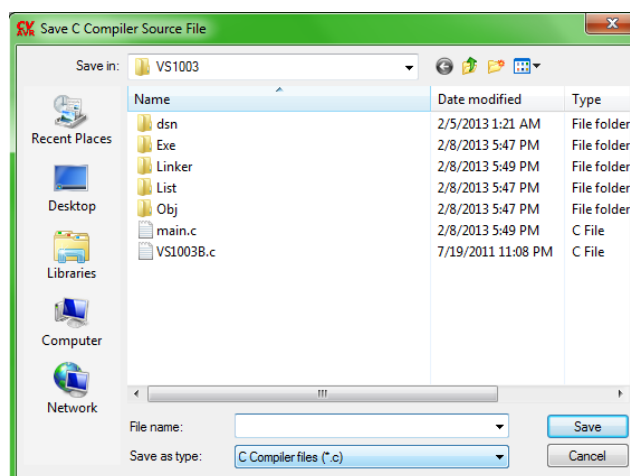
شکل (۴-۷) تعیین پایه ها به عنوان ورودی و خروجی

برای دانلود پروژه های الکترونیکی بیشتر به وبسایت ما به آدرس [Melec.ir](http://Melec.ir) مراجعه فرمائید.



شکل (۴-۸) تولید کدها

بعد از تنظیم کلیه رجیسترهایی که در پروژه نیاز است طبق شکل زیر از منوی PROGRAM زیرمنوی Generate، Save and Exit را انتخاب می‌شود.



شکل (۴-۹) ذخیره برنامه

برای دانلود پروژه های الکترونیکی بیشتر به وبسایت ما به آدرس [Melec.ir](http://Melec.ir) مراجعه فرمائید.

## أ-۱۲- برنامه نویسی میکروکنترلر پروژه

```
#include <mega8.h>
#include <delay.h>
#include <alcd.h>
#include <stdio.h>
```

فراخوانی هدر فایل های مورد نیاز برای پروژه

```
#define RELAY_1 PORTD.6
#define RELAY_2 PORTD.7
```

تعیین پایه های رله ها

```
#define ADC_VREF_TYPE 0x60
```

تعیین رفرنس مبدل آنالوگ به دیجیتال

```
interrupt [USART_RXC] void usart_rx_isr(void)
{
char status.data;

status=UCSRA;
data=UDR;

if ((status & (FRAMING_ERROR | PARITY_ERROR | DATA_OVERRUN))==0)
{
rx_buffer[rx_wr_index++]=data;
#if RX_BUFFER_SIZE == 256
// special case for receiver buffer size=256
if (++rx_counter == 0) rx_buffer_overflow=1;
#else
if (rx_wr_index == RX_BUFFER_SIZE) rx_wr_index=0;
if (++rx_counter == RX_BUFFER_SIZE)
{
rx_counter=0;
rx_buffer_overflow=1;
}
#endif
}
}
}
```

برای دانلود پروژه های الکترونیکی بیشتر به وبسایت ما به آدرس [Melec.ir](http://Melec.ir) مراجعه فرمائید.

```
unsigned char read_adc(unsigned char adc_input)
{
ADMUX=adc_input | (ADC_VREF_TYPE & 0xff);
delay_us(10);
ADCSRA|=0x40;
while ((ADCSRA & 0x10)==0);
ADCSRA|=0x10;
return ADCH;
}
```

تابع بالا برای خواندن مقدار مبدل آنالوگ به دیجیتال است و این تابع توسط کد ویزارد خود کدویژن تولید شده است. عملکرد کلی آن هم بدین صورت است که هر بار این تابع در برنامه فرخوانی می شود. این تابع مقدار سنسور را می خواند.

```
void main(void){
```

تابع اصلی برنامه

```
unsigned char _temp;
```

متغیری که مقدار دما را در خود نگه می دارد. یعنی مقدار خوانده شده از ADC را نگه می دارد.

```
unsigned char _buf_1[17],_r_buf[21];
```

دوتا متغیر آرایه ای که از آنها برای ذخیره اطلاعات استفاده می شود.

```
unsigned char i;
```

متغیر I برای شمارش تعداد گام های حلقه for بکار می رود.

```
DDRD=(1<<6)|(1<<7);
```

تعیین پایه های میکرو به عنوان خروجی برای رله ها

```
ADMUX=ADC_VREF_TYPE & 0xff;
```

```
ADCSRA=0x86;
```

### أ-۱۳- رجیستر آنالوگ به دیجیتال

این رجیستر ها باز توسط کدویزارد تنظیم می شوند. در کل ADMUX کانال پیش فرض را برای ADC تعیین می کند و رجیستر ADCSRA برای تنظیم مد کاری مبدل و فرکانس نمونه برداری است.

```
UCSRA=0x00;
```

```
UCSRB=0x98;
```

```
UCSRC=0x86;
```

```
UBRRH=0x00;
```

```
UBRRL=0x33;
```



برای دانلود پروژه های الکترونیکی بیشتر به وبسایت ما به آدرس [Melec.ir](http://Melec.ir) مراجعه فرمائید.

#### أ-۱۴- تنظیم رجیستر های پورت سریال

این رجیستر ها نیز توسط کدویزارد تنظیم شده اند. بصورت پیش فرض و با بادریت ۹۶۰۰ تنظیم شده است.

```
LCD_init(16);
```

تنظیم LCD در حالت 16ستون و راه اندازی اولیه LCD

```
LCD_clear();
```

پاک کردن LCD

```
LCD_putsf(" Milad Jahandideh ");
```

نوشتن اسم بصورت لاتین روی LCD

```
delay_ms(3000);
```

تأخیر ۳ ثانیه ای برای مشاهده کامل اسم

```
LCD_clear();
```

پاک کردن LCD

```
LCD_putsf("Connect USB and ");
```

```
delay_ms(1);
```

```
LCD_gotoxy(0,1);
```

```
LCD_putsf(" Run PC Program ");
```

```
RELAY_1=1;
```

```
RELAY_2=1;
```

```
delay_ms(100);
```

```
RELAY_1=0;
```

```
RELAY_2=0;
```

تست رله ها (خاموش و روشن کردن)

```
#asm("sei")
```

دستور بالا وقفه سراسری را فعال می کند. برای اینکه بتوان اطلاعات ارسال شده توسط کامپیوتر را دریافت کرد باید وقفه ای در سیستم رخ دهد که این دستور باعث مجوز وقفه می شود یعنی اگر این دستور نباشد وقفه رخ نمی دهد و اطلاعات از بین می روند.

برای دانلود پروژه های الکترونیکی بیشتر به وبسایت ما به آدرس [Melec.ir](http://Melec.ir) مراجعه فرمائید.

```
while(1){
```

حلقه بینهایت تکرار این حلقه تا وقتی دستگاه روشن هست تکرار می شود.

```
_temp=read_adc(0);
```

تابع بالا مقدار سنسور دما را از ADC می خواند و داخل متغیر `_temp` می ریزد.

```
_temp*=2;
```

مقدار خوانده شده اطلاعات از سنسور ولتاژ هست و این ولتاژ باید به دما تبدیل شود. با ضرب کردن مقدار خوانده شده به ۲ این کار انجام می شود یعنی در این حالت مقدار خوانده شده بین ۰ تا ۲۵۵ هست با ضرب در ۲ که عبارتی تابع تبدیل سنسور است. با این کار مقدار خوانده شده از ADC به دما تبدیل می شود. البته کمی خطا پیش می آید ولی چون سیستم حساس نیست قابل چشم پوشی هست.

مثلاً مقدار دما ۲۰ درجه سانتی گراد هست و این سنسور به ازای هر درجه سانتی گراد ۱۰ میلی ولت ولتاژ تولید میکند پس ولتاژ خروجی سنسور ۲۰۰ میلی ولت هست و مقدار خوانده شده از adc عدد ۱۰ خواهد شد و با ضرب ۱۰ به ۲ درست مقدار دما بدست می آید.

$$(10mV*255)/5 = x$$

۱۰ میلی ولت مقدار هر درجه سنسور هست که بالا توضیح داده شد.

۲۵۵ حداکثر مقدار خروجی در ۵ ولت است یعنی اگر ۵ ولت به پایه ADC داده شود حداکثر عدد ۲۵۵ است.

```
printf(_buf_1,"T=%d°C R1=%d R2=%d",_temp,223,RELAY_1,RELAY_2);
```

قالب بندی اطلاعات برای نمایش در LCD

بطور ساده این تابع یکسری اطلاعات عدد، علامت، حرف و... را دریافت و تبدیل به یک رشته یا کاراکتر می کند که برای نمایش روی یک LCD کاراکتری نیاز است.

```
sprint(buff,"T=%d",temp)
```

متغیر `buff` یک سری رشته را می تواند نگه دارد و اگر `temp` را ۳۵ درجه نظر گرفته شود پس چیزی که روی نمایشگر از خروجی این تابع دیده می شود بصورت `T=35` است.

أ-۱۵- ارسال دما به کامپیوتر

```
printf("%d",_temp);//send Temp for PC
```

تابع بالا هم مثل تابع `sprintf` است با این تفاوت که مقدار خروجی را به پورت سریال ارسال می کند. یعنی این دفعه دما را به کامپیوتر می فرستد و عملکردش مثل تابع قبلی هست و فقط `buff` را ندارد.

برای دانلود پروژه های الکترونیکی بیشتر به وبسایت ما به آدرس [Melec.ir](http://Melec.ir) مراجعه فرمائید.

## أ-۱۶- دریافت اطلاعات از کامپیوتر

بطور خیلی ساده HelloE2ND 1 RL1 قالب دریافت اطلاعات است یعنی اطلاعات همیشه اینطوری دریافت می شوند البته چند قسمت این جمله متغیر هست. علت این کار تعریف یک استاندارد هست تا اگر یک بایت گم شد سیستم بهم نریزد. RL همیشه ثابت است 11 ها همان مقدار صفر یا یک بودن رله را مشخص می کنند مثلاً اگر HelloE2ND 0 RL0 باشد یعنی رله ها توسط لب ویو خاموش شده اند. قسمت بعدی این متغیر HELLO هست که می تواند تا ۱۶ کاراکتر هر کلمه ای باشد.

```
if(getchar()=='R' & getchar()=='L') // RL1 1 HelloE2ND
```

```
{
```

خط بالا اول چک می کند تا ۲ بایت دریافت شده به ترتیب RL باشد.

```
if(getchar()=='1')
```

اگر بایت بعدی ۱ دریافت شد یعنی رله اول روشن

```
RELAY_1=1;
```

```
else
```

```
RELAY_1=0;
```

اگر بایت بعدی 0 دریافت شد یعنی رله اول خاموش

```
if(getchar()=='1')
```

```
RELAY_2=1;
```

اگر بایت بعدی ۱ دریافت شد یعنی رله دوم روشن

```
else
```

```
RELAY_2=0;
```

اگر بایت بعدی ۰ دریافت شد یعنی رله دوم خاموش

```
for(i=0;i<21;i++){
```

```
_r_buf[i]=getchar();
```

```
if(_r_buf[i-3]=='E' & _r_buf[i-2]=='2' & _r_buf[i-1]=='N' & _r_buf[i]=='D'){
```

```
_r_buf[i-3]=0;
```

```
goto _down;
```

```
}
```

```
}
```

```
_down:
```

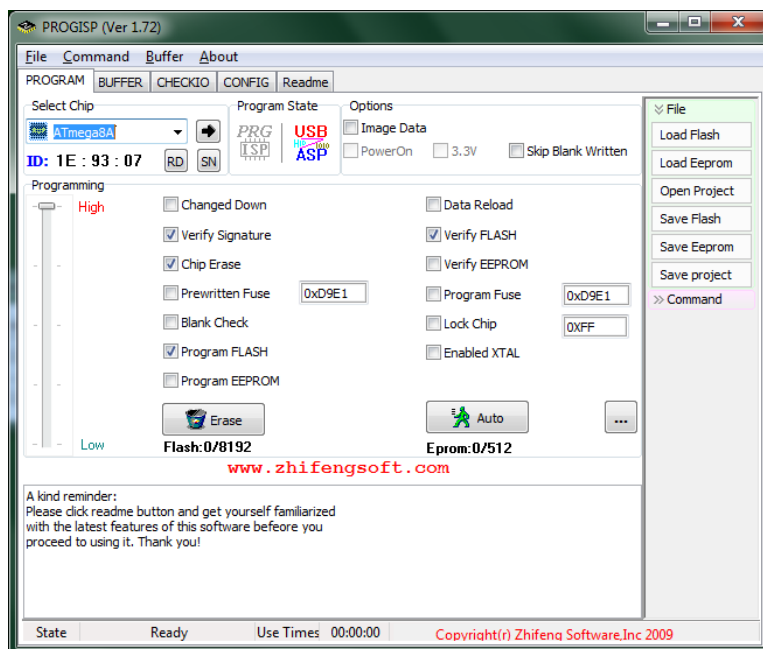
برای دانلود پروژه های الکترونیکی بیشتر به وبسایت ما به آدرس [Melec.ir](http://Melec.ir) مراجعه فرمائید.

چند خط بالا جمله‌ای که به جای HELLO فرستاده شده است را از دل پیام استخراج می‌کند. اول انتهای پیام را که 2ND است پیدا و از آن به قبل تا وضعیت رله‌ها را به عنوان جمله تلقی می‌کند.

```
}  
LCD_clear();  
LCD_puts(_buf_1);  
delay_us(100);  
LCD_gotoxy(0,1);  
LCD_puts(_r_buf);  
delay_ms(150);  
}  
}
```

### ا-۱۷- نحوه پروگرام کردن میکروکنترلر

برای پروگرام کردن میکروکنترلر از نرم افزار PROGISP استفاده می‌شود و سخت افزار پروگرامر یک پروگرامر USB AVR است. در شکل زیر نمای از این نرم افزار آورده شده است. از قسمت Load Flash کد خروجی یا هگز برنامه باز می‌شود و بعد از قسمت Select Chip نوع میکروکنترلر انتخاب و بعد Erase زده می‌شود تا میکروکنترلر را پاک کند و بعد Auto را زده می‌شود تا میکروکنترلر پروگرام شود.

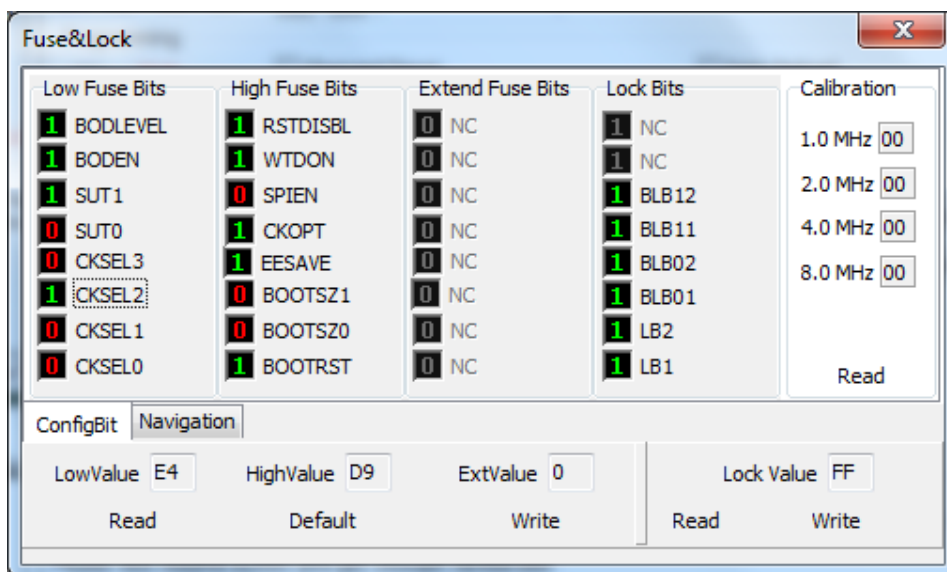


شکل (۴-۱۰) برنامه پروگرامر

برای دانلود پروژه های الکترونیکی بیشتر به وبسایت ما به آدرس [Melec.ir](http://Melec.ir) مراجعه فرمائید.

### ۱۸- تنظیم فیوزبیت های میکروکنترلر

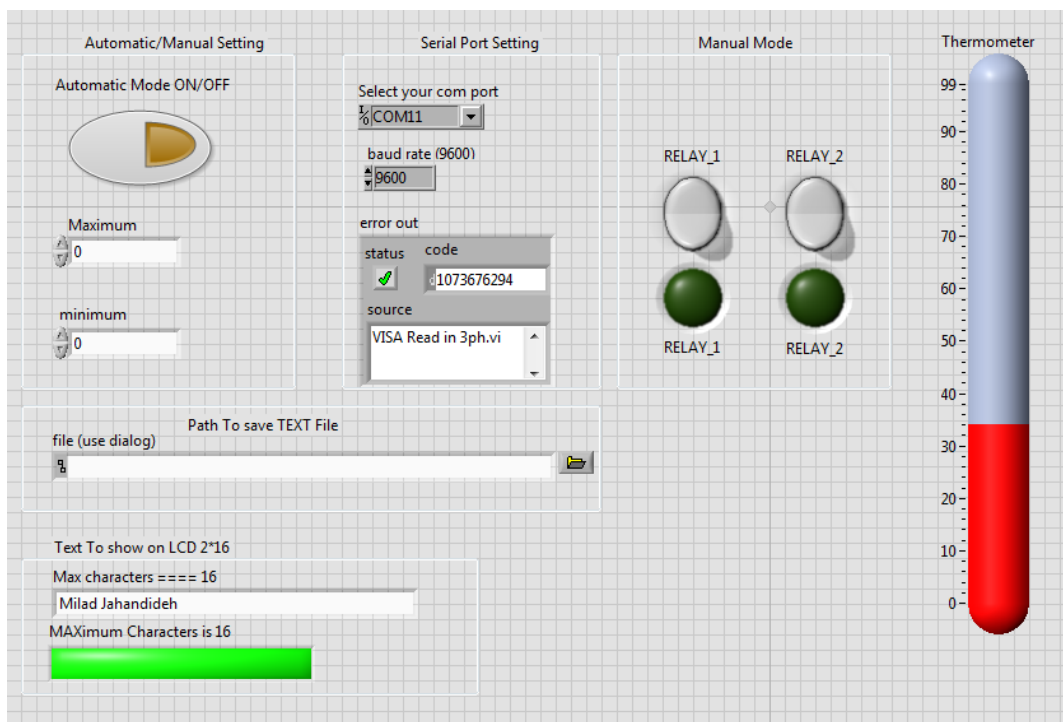
برای تنظیم فیوز بیتها از منوی سه تا نقطه استفاده می شود که باید بصورت شکل زیر تنظیم شده باشند.



شکل (۴-۱۱) تنظیم فیوز بیتها

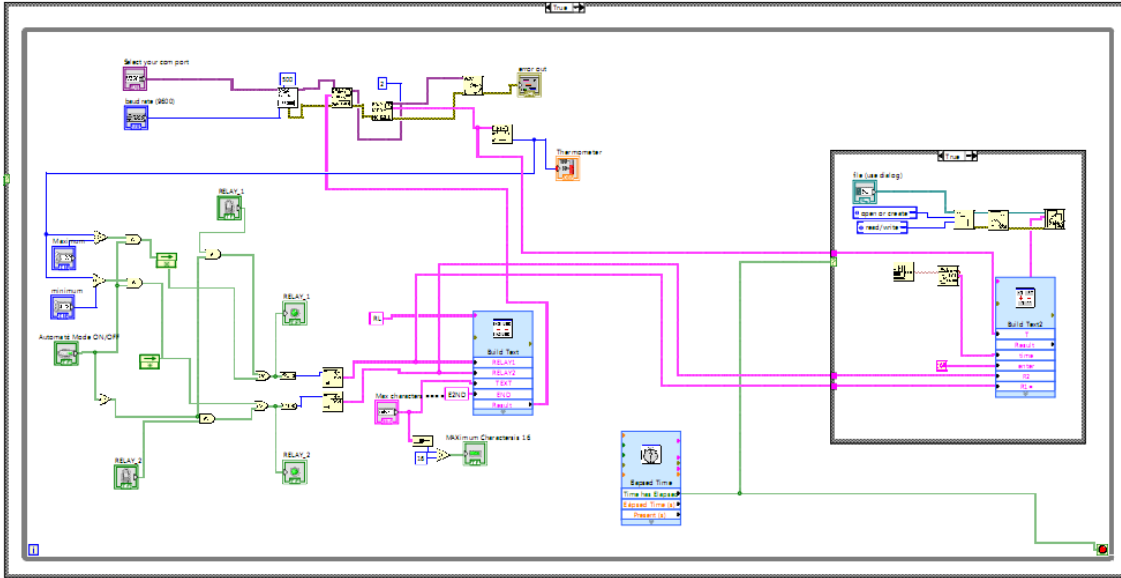
### ۱۹- برنامه LabVIEW

برای طراحی نرم افزار کامپیوتر از لب ویو استفاده شده است. در شکل زیر عکسی از خروجی نهایی نرم افزار کامپیوتر مشاهده می شود و همچنین تشریح عملکرد بلوک دیگرام های برنامه نویسی در ادامه بررسی می شوند.



شکل (۴-۱۲) برنامه نهایی لب ویو

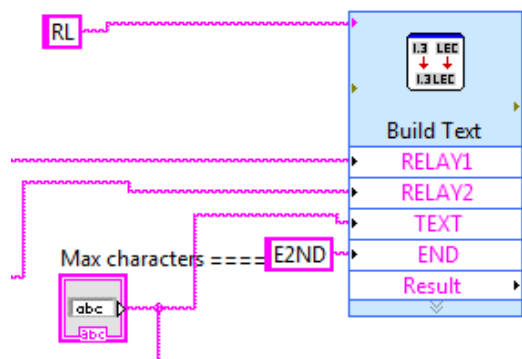
برای دانلود پروژه های الکترونیکی بیشتر به وبسایت ما به آدرس [Melec.ir](http://Melec.ir) مراجعه فرمائید.



شکل (۴-۱۳) برنامه نویسی لب ویو

۲۰- فرمت بندی اطلاعات پورت سریال

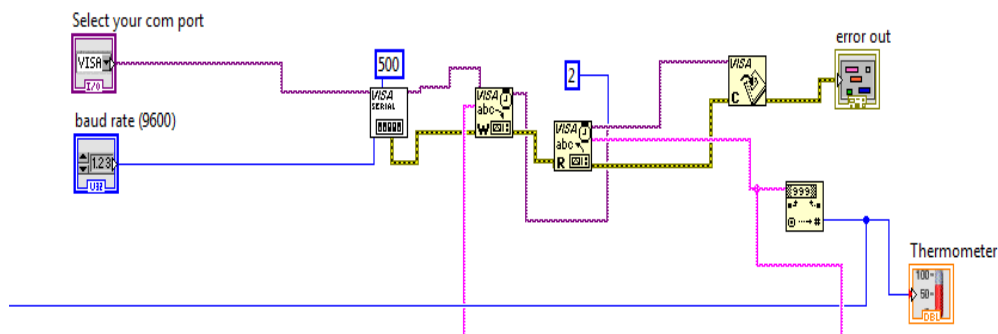
فرمت بندی پورت سریال بصورت شکل زیر می باشد. وضعیت کلیدها، نوشته را با فرمتی که اول با RL شروع می شود و با E2ND به پایان می رسد. و این قالب هر ۵۰۰ میلی ثانیه یکبار به میکروکنترلر ارسال می شود.



شکل (۴-۱۴) فرمت بندی اطلاعات

برای دانلود پروژه های الکترونیکی بیشتر به وبسایت ما به آدرس [Melec.ir](http://Melec.ir) مراجعه فرمائید.

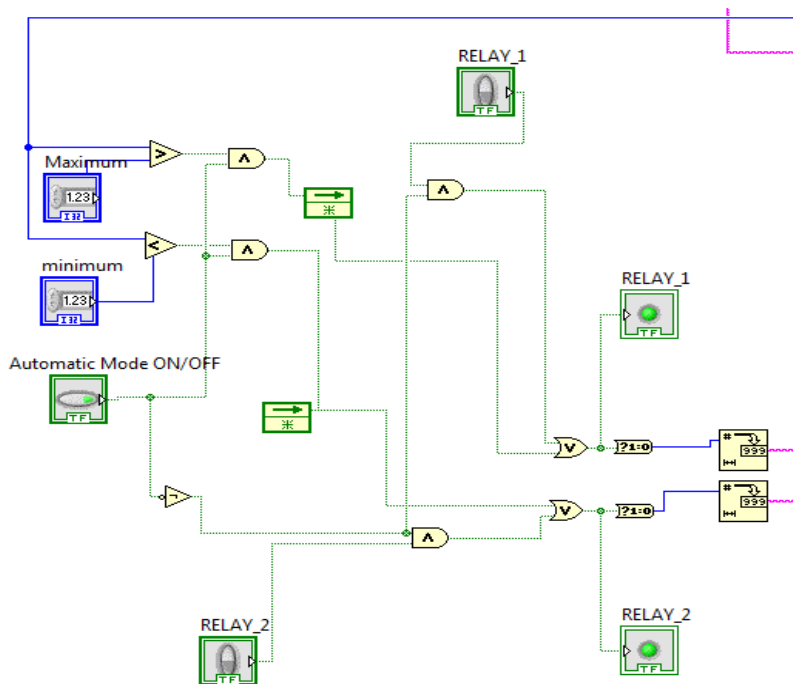
أ-۲۱- قسمت ارتباط با سریال و نمایش دمای دریافت شده



شکل(۴-۱۵) بلوک های پورت سریال

أ-۲۲- برنامه حالت دستی و اتوماتیک

در این قسمت با گیت های منطقی حالت اتوماتیک و دستی انتخاب می شود و در صورت روشن بودن یکی از حالت ها حالت دیگر غیر فعال می شود. همچنین مقادیر حداکثر و حداقل از کاربر دریافت و با مقدار سنسور مقایسه می شود.

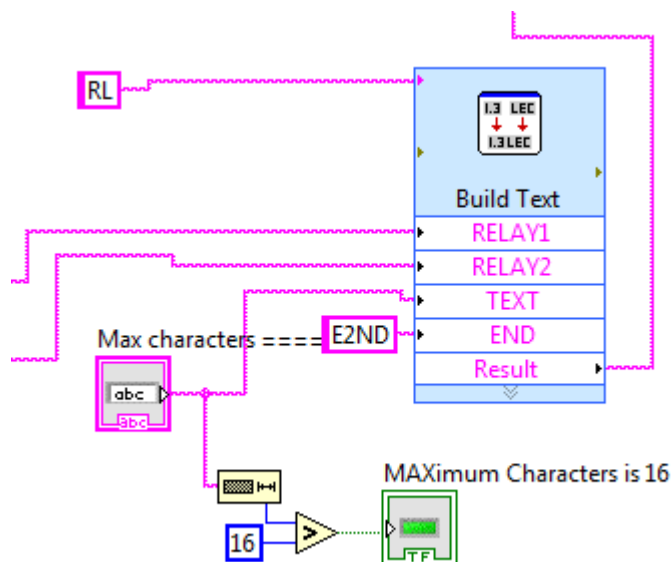


شکل(۴-۱۶) قسمت کنترل کننده

أ-۲۳- قسمت ورودی تکست برای نمایش روی LCD

قسمت ورودی تکست و مقایسه رشته برای اینکه بزرگتر از ۱۶ کاراکتر نشود اگر بزرگتر از ۱۶ کاراکتر شود LED متناظر روی LabVIEW قرمز می شود.

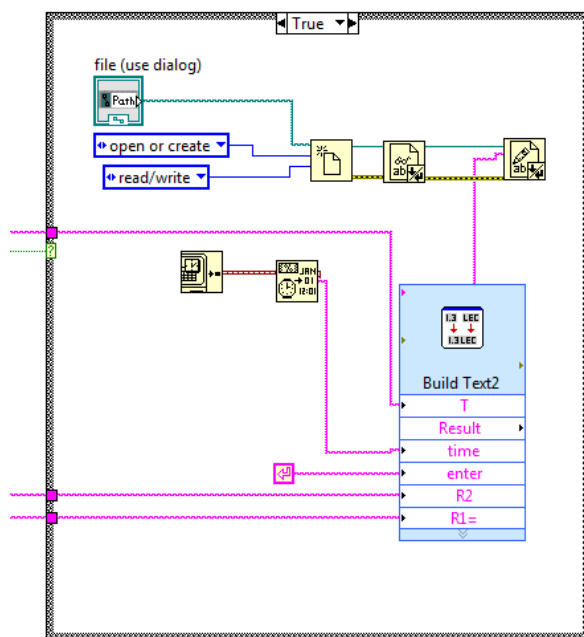
برای دانلود پروژه های الکترونیکی بیشتر به وبسایت ما به آدرس [Melec.ir](http://Melec.ir) مراجعه فرمائید.



شکل (۴-۱۷) نوشته برای نمایش روی سخت افزار

۲۴-ا- ذخیره سازی اطلاعات در یک فایل متنی

قسمت ذخیره سازی اطلاعات در یک فایل متنی بصورت یک تایمر که هر ۵ ثانیه یکبار اجازه اجرای بلوک زیر را می دهد و برنامه ، اطلاعات را در یک فایل متنی ذخیره می کند.



شکل (۴-۱۸) ذخیره دما و تاریخ در یک فایل متنی

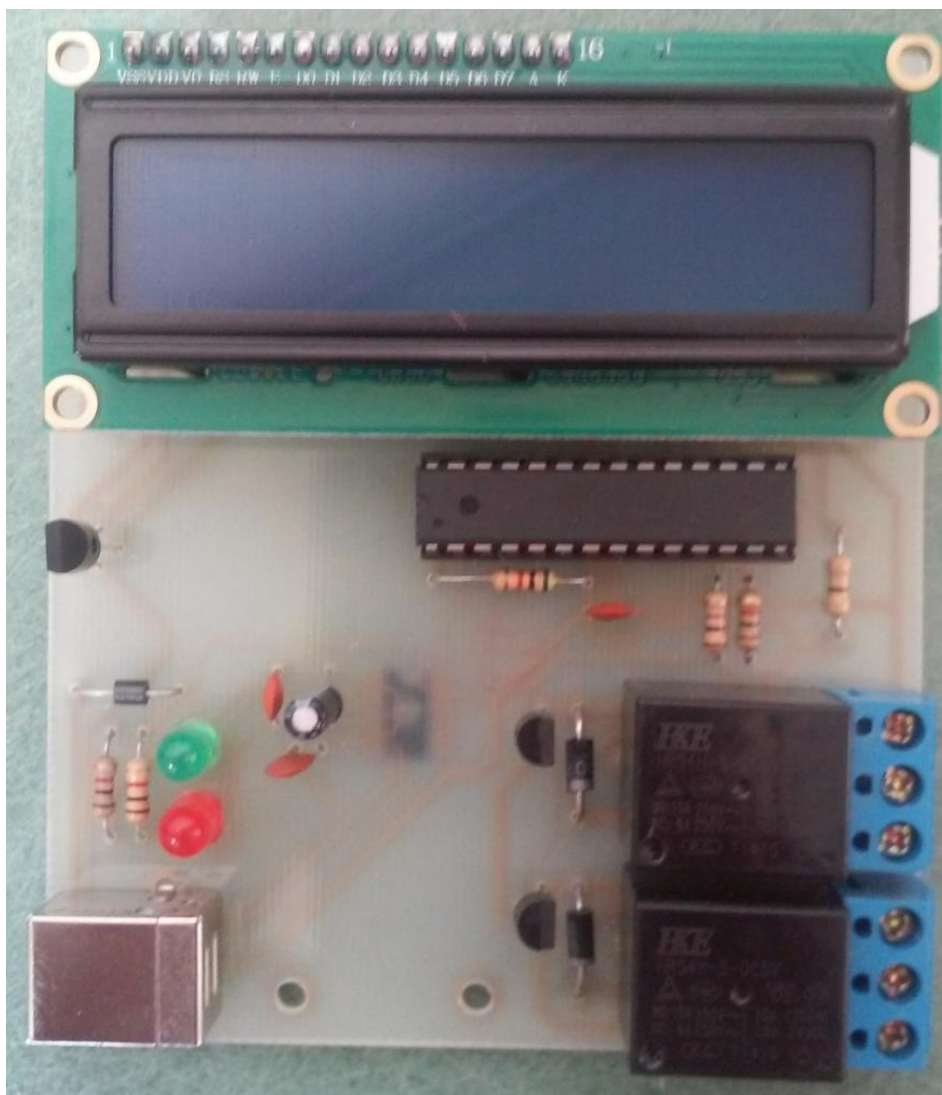


برای دانلود پروژه های الکترونیکی بیشتر به وبسایت ما به آدرس [Melec.ir](http://Melec.ir) مراجعه فرمائید.

أ-۲۵- لیست قطعات سخت افزار پروژه

- میکروکنترلر ATMEGA8A-DIP
- آی سی مبدل FT232R
- نمایشگر LCD ۱۶\*۲
- سنسور دمای LM35
- سوکت ۲۸ پین
- پتانسیومتر پیچی ۵ کیلو اهم
- رله ۵ ولت کنتاکت ۱۰ آمپر ۲ عدد
- ترمینال آبی ۳ پین ۲ عدد
- دیود ۴۰۰۷ یا ۴۱۴۸ ۲ عدد
- ترانزیستور N2222۱ دو عدد
- کانکتور USB-B مادگی
- کابل USB-B
- آل ای دی سبز و قرمز
- فریت بیت
- خازن ۱۰۰ نانو عدسی ۴ عدد
- مقاومت ۱ کیلو اهم ۲ عدد
- مقاومت صفر اهم
- مقاومت ۴.۷ کیلواهم
- سلف ۱۰ میکروهانری مقاومتی
- پین هدر نری و مادگی ۴۰\*۱ یک شاخه
- خازن ۴.۷ میکروفاراد
- مقاومت ۲۲۰ اهم دو عدد

برای دانلود پروژه های الکترونیکی بیشتر به وبسایت ما به آدرس [Melec.ir](http://Melec.ir) مراجعه فرمائید.



شکل (۴-۱۹) سخت افزار نهایی پروژه

برای دانلود پروژه های الکترونیکی بیشتر به وبسایت ما به آدرس [Melec.ir](http://Melec.ir) مراجعه فرمائید.

## **Abstract**

**The aim of this project is designed to control and monitoring ambient temperature. The monitoring tool is LabVIEW software. The structure of controller is based on ON-OFF with dead band. LM35 sensor has been chosen for sense the ambient temprature. The ATMEGA8 microcontroller is used for data transfer to LabVIEW by RS232 protocol. This project is designed and fabricated for two modes. One mode is manual and other mode is automatic.**

**Keywords: Tempreature Control, Monitoring, LabVIEW, RS232 Interface,**

برای دانلود پروژه های الکترونیکی بیشتر به وبسایت ما به آدرس [Melec.ir](http://Melec.ir) مراجعه فرمائید.



**Shamsipour  
Technical and Vocational  
College**

Shamsipour Technical and Vocational University

# **The ambient temperature monitoring and control in LabVIEW software**

By:

Milad Jahandideh

Supervisor:

Instructor.Vahid Bohlouri

January 2016