

نویز Noise

نشریه تخصصی برق و الکترونیک ECA / سال سوم / شماره ۳ / آذر ۱۳۹۱



● راه اندازی ماژول SIM900 توسط LabVIEW

● راه اندازی ماژول HC-05

● ساخت ماشین حساب لمسی توسط LCD 3.2"

● پروژه پخش فایل های صوتی

● آغاز کار با تراشه های XMEGA - کار با پورت ها

www.Noisemagazine.ECA.ir

منشور مجله نویز

الف - معرفی

ماهانمه نویز، نشریه ای است علمی، خبری، آموزشی که با تکیه بر خلاقیت و نوآوری کارشناسان ایرانی و انجام کار گروهی آنان به مخاطبان خود کمک می کند تا نیازهای خود را در شکل مکتوب در این نشریه محقق شده بیابند.

ب - اهداف

- ۱- پیشبرد فرهنگ علم گرایی در کشور
- ۲- ایجاد روحیه تلاش و امید در نسل جوان
- ۳- توسعه کمی و کیفی روزافزون به سوی نشریه ای وزین، فراگیر و کم اشکال
- ۴- اطلاع رسانی به روز علوم مرتبط

ج - اصول و ارزش ها

- ۱- احترام به مخاطب
- ۲- راستی در گفتار و رفتار میان دست اندر کاران نشریه
- ۳- تلاش مستمر برای بهبود
- ۴- استفاده از پتانسیل موجود در قشر تحصیل کرده کشور

د - حوزه های فعالیت (موضوعات کلیدی)

- ۱- آموزش مبانی علوم برق و الکترونیک
- ۲- آشنایی با تکنولوژی های نوین دنیای فناوری
- ۳- طراحی مدارات آنالوگ و دیجیتال
- ۴- آموزش و معرفی نکات کاربردی پردازنده ها
- ۵- مدارات مجتمع
- ۶- آموزش کاربردی نرم افزارهای تخصصی
- ۷- سیستم ها و مدارات مخابراتی
- ۸- سیستم های قدرت
- ۹- معرفی مشاهیر برق
- ۱۰- موارد کاربردی و عملی علم الکترونیک
- ۱۱- بخش های متنوع مرتبط با موضوعات برق و الکترونیک

* استفاده از مقاله های مجله، با ذکر مأخذ و رعایت حقوق نویسندگان بلامانع است.

* مجله نویز آماده دریافت آثار و مقالات ارسالی متخصصین و مهندسين است.

* لطفاً مقاله های خود را بصورت تایپ شده به همراه ضمیمه عکس های مورد استفاده ارسال نمایید.

* نشریه در ویرایش و اصلاح مطالب رسیده، آزاد است.

* چنانچه مطالب ارسالی ترجمه است، کپی اصل آن را ضمیمه نمایید.

صندوق پست الکترونیکی مجله: noisemagazine.eca@gmail.com



فهرست مطالب : صفحه

- شش روش بهینه طراحی، جهت افزایش طول عمر باتری گوشی های هوشمند..... ۲
- راه اندازی ماژول SIM900 توسط LabVIEW..... ۴
- کاربرد نیمه هادی ها در روش های مختلف استفاده از انرژی خورشیدی ۸
- صفحه نمایش های OLED..... ۱۴
- آموزش ارتباط با دات ماتریس درایور HD44780 و نوشتن کتابخانه برای LCD کاراکتری..... ۱۸
- راه اندازی ماژول HC-05..... ۲۶
- پروژه دستگاه تشخیص نشت گازهای خانگی..... ۳۴
- ساخت ماشین حساب لمسی بر روی LCD 3.2 اینچی..... ۴۲
- پروژه پخش فایل های صوتی..... ۵۰
- آموزش کار کردن با پورت های XMEGA..... ۶۰
- نگاهی گذرا به ساخت بردهای چند لایه..... ۷۰
- ربات شن نورد..... ۷۲
- جان لاجی برد..... ۷۸
- Silicon Nanowire Field Effect Transistor..... ۸۰



مترجم: فرزانه یوسفی پور

شش روش بهینه طراحی، جهت افزایش طول عمر باتری گوشی های هوشمند

توسعه روزافزون دستگاه های همراه و مراکز داده ها، رقابتی را بین طراحان مدار ایجاد کرده تا بتوانند با ارائه راه کارهای جدید مصرف را تا حد زیادی کاهش دهند. در کنفرانس بین المللی SSC امسال در سان فرانسیسکو، جایگاه مرکزی به شش تکنولوژی ذخیره توان اختصاص داشت. بعضی از این روش ها در تولیدات امسالی نمود پیدا خواهند کرد در حالیکه بقیه فقط در حد ایجاد علاقه برای سازندگان چیپ ها هستند.

شش تکنولوژی ذکر شده به شرح زیر می باشند:

۱. مصرف تا نزدیک آستانه ولتاژ

آکادمی ها مدت زیادی با ایده ی "راه اندازی چیپ ها با ولتاژی نزدیک به مقدار تعریف شده" (مقداری که برای روشن و خاموش شدن یک ترانزیستور نیاز است) بازی کرده اند. به نظر می رسد که این ایده امروزه در صنعت به کار گرفته شده است چرا که محققان اینتل در مورد یک چیپ از کلاس پنتیوم، ۳۲ نانومتر که طراحی خودشان می باشد ذکر کرده اند که این چیپ می تواند با ۱.۲ ولت (که برای پردازنده های امروزی اجباری است) کار کند و ولتاژ آن تا ۲۸۰ میلی ولت نیز می تواند کاهش پیدا کند. نقطه ی مطلوب برای کارآمد بودن انرژی در حد ۴۵۰ میلی وات است. در این سطح، چیپ های اینتل به آرامی در کمتر از ۱۰۰ مگاهرتز کار میکنند اما درعین حال انرژی ای حدود یک پنجم انرژی در ۱.۲ ولت را مصرف می کنند. برای جبران این نقص (کندی) از پردازنده های موازی استفاده می شود.

۲. به ی باریک تیغ

مهندسين معمولاً برای جلوگیری از خطاهایی که ممکن است در کلاک سیستم رخ دهد، چیپ ها را در ولتاژی بالاتر از آنچه نیاز دارند تغذیه می کنند. اگر چیپ ها دارای امکاناتی برای تشخیص خطا و تغییر ولتاژ کاری حین پردازش باشند، جهت ذخیره توان می توان چیپ ها را در کمترین ولتاژ کاری ممکن در حین پردازش راه اندازی کرد. اما محققان دانشگاه میشیگان در ann arbor و کالج hurveymudd در Claremont نشان دادند که فعالیت های حاصل شده در زمینه این تکنولوژی، بر روی پروسسور هایی با هسته آرم (cortex-m3) می تواند ذخیره انرژی را تا ۶۰ درصد افزایش دهد. این تیم، همچنین می گوید که این کار اولین اجرای طرح مدل تیغ روی یک پروسسور تجاری است.

۳. ترانزیستورهای کوچکتر

اینتل ۱.۴ بیلیون ترانزیستور ۳ بعدی را روی Ivy Bridge (نسل بعدی پروسسور ها) بسته بندی کرده است. ترانزیستور های سه بعدی برای کاهش نشستی در چیپ های جدید توان بالا مصرف شده اند. چیپ ۲۲-nm Ivy Bridge می تواند با همان سرعت چیپ های قبلی شرکت، اما با مصرف ولتاژی در حدود ۲۰۰ میلی ولت کمتر راه اندازی شود. اینتل همچنین طراحی مدارها و سطح هسته ها را برای رسیدن به مدیریت بهتر مصرف توان چیپ، یکی کرده است. یک سیستم روی چیپ که به صورت کد silvermont نام گذاری شده است و بر اساس فرآیند ساخت ترانزیستور می باشد، می تواند به عنوان تجهیز برای موبایل ها استفاده شود.

۴. قفل فاز تمام دیجیتال

حلقه های قفل فاز (که از داخل قفل می شوند و یک سیگنال ورودی را دنبال می کنند) اجزای حیاتی هستند که برای همزمان کردن کلاک پردازنده های مدرن و نیز برای افزایش سرعت و انتقال سیگنال های رادیویی استفاده می شود. در گذشته این مدارها با اجزای



آنالوگ ساخته می شدند اما امروزه در ساخت آن ها از اجزای تمام دیجیتال که یک دهم توان اجزای آنالوگ را مصرف می کنند و نیز آسانتر ساخته میشوند، استفاده می شود. مرکز توان موبایل سامسونگ پیشرفت جدیدی را روی حلقه قفل فاز ارائه کرده اند که با استفاده از آن یک مدار با ابعاد ۰.۱۲ میلی متر مربع فقط ۲.۵ میلی وات توان مصرف میکند. اینتل نیز ورژن جدیدی از مدار ها را به نمایش گذاشت که با استفاده از تکنولوژی ۲۲ نانو متری ساخته شده و توان کمی در حدود ۰.۷ میلی وات مصرف دارد.

۵. مبدل های هوشمند

روش دیگری که برای انتقال توان پایین کاربرد دارد، خازن سوئیچ شده هست که اغلب برای تبدیل سیگنال آنالوگ به دیجیتال استفاده میشود. تیمی در دانشگاه Oregon state در Corvallis شروع بهساخت اجزای توان پایینیکرده است که بوسیله ی نوسانساز حلقه ای شکل گرفته است که شامل یک مدار تست مشترک ساخته شده از مبدل های حلقه می باشد. تیم دیگری در National Chiao Tung University, in Hsinchu, Taiwan راهی را برای ذخیره کردن توان بوسیله کار روی سیگنال ها در دو مرحله جدا که یکی برای فرآیند های خام اولیه و دیگری برای تنظیم هر چه بهتر سیگنال می باشد ارائه کردند که هر کدام می توانند بهینه شوند.

۶. نسل بعدی RAM های دینامیک

مرکز ساخت حافظه سامسونگ و Hynix Semiconductor هر دو جزئیاتی از نسل بعدی Dram ها را هم زمان آشکار کردند، این حافظه ها توانایی راه اندازی پردازنده های امروزی را دارا هستند. نسل جدید آن ها که با نام DDR4 آمده اند، کاربرد های جالبی از مدار را نشان می دهد که اجازه میدهد سامسونگ ولتاژ تأمین برای مازول های حافظه را تا حد ۱.۵ یا ۱.۲ ولت کاهش دهد. این مازول ها همچنین شامل کلاک بهتر و الگوریتم سریع تر برای کدگشایی داده جهت ارسال و دریافت داده از حافظه می باشند. DDR4 ممکن است به زودی و تا سال ۲۰۱۳ تولید تجاری اش را شروع کند.

منبع:

<http://spectrum.ieee.org/semiconductors/processors/six-paths-to-longer-battery-life>

نویز Noise

مجله تخصصی برق و الکترونیک

از شماره ۱ تا شماره ۱۲

پروژه های عملی
تکنولوژی های نوین
آموزش نرم افزار
مقالات علمی

وب سایت تخصصی
برق و الکترونیک
www.ECA.ir

مجله تخصصی برق و الکترونیک نویز

لوح فشرده مجلات شامل:
- تمامی شماره های مجله تخصصی برق و الکترونیک نویز
- سورس تمامی پروژه های موجود در مجله

توضیحات تکمیلی : <http://eshop.eca.ir/link/NOISE.php>

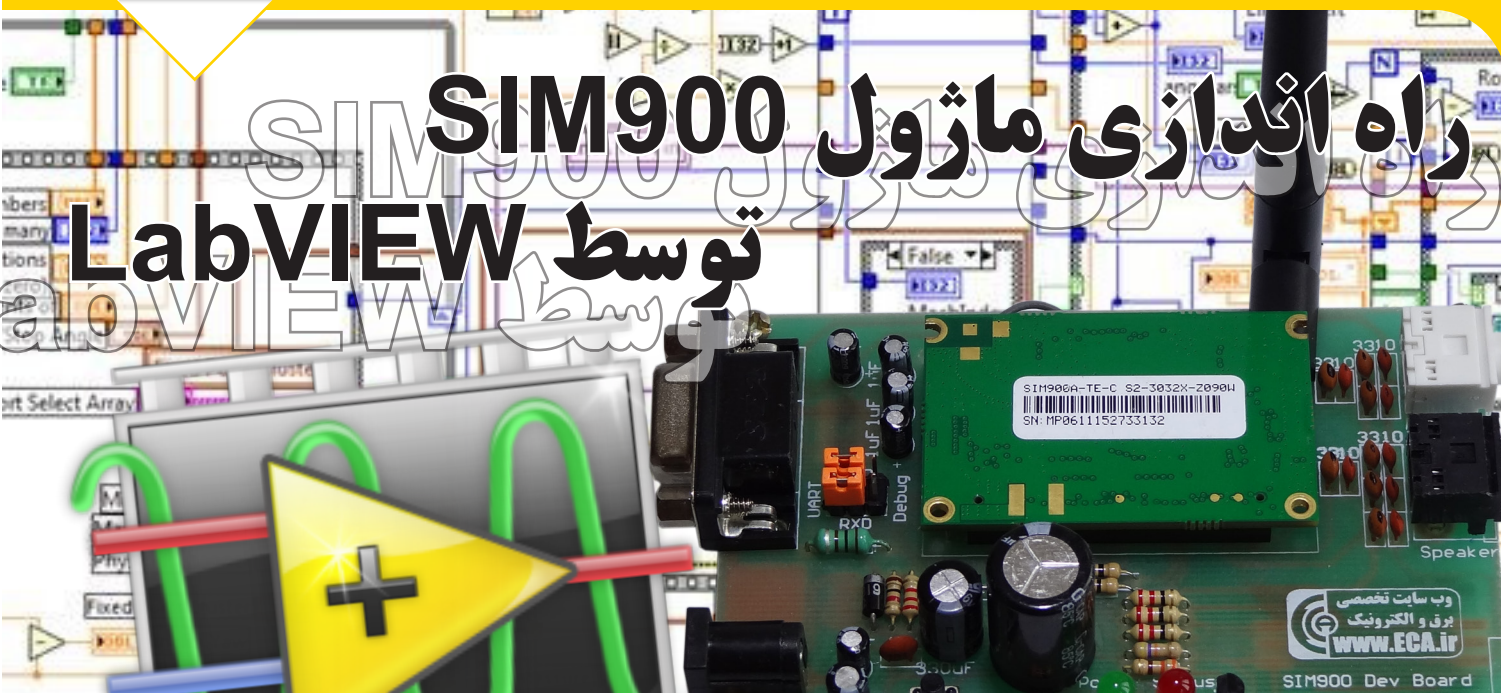


NI LabVIEW

DESIGN, ANALYSIS AND SIMULATION

لبویو

راه اندازی ماژول SIM900 توسط LabVIEW



نویسنده: فرشاد اکرمی

ماژول SIM900 یکی از پرکاربردترین ماژول های GSM موجود در بازار بوده که به جهت امکانات موجود بر روی این ماژول در بیشتر مصارفی که نیاز به ارتباط GSM و اتصال به شبکه های موبایل لازم باشد از این مورد استفاده قرار می گیرد.

نحوه برقراری ارتباط با این ماژول بسیار ساده می باشد. با ارتباط سریال دستورات به ماژول ارسال و ماژول طبق دستورات ذخیره شده در خود، عملیات متناظر با دستور مورد نظر را انجام می دهد.

علاوه بر ساده بودن ارتباط و طرز کارکرد ماژول، راه اندازی این ماژول تاحدی مشکل بوده و در صورتی که قطعات و طراحی PCB بر اساس دیتاشیت طراحی نگردد، این ماژول راه اندازی نشده و کار نخواهد کرد.

همین مشکل باعث گشته تا بسیاری از کسانی که برای اولین بار می خواهند از این ماژول استفاده نمایند، در مرحله راه اندازی دچار مشکل شده و دلسرد گردند. بهترین راه برای جلوگیری از اتلاف وقت، مطالعه کامل دیتاشیت و همچنین رعایت تمامی نکات لازم جهت جلوگیری از نویز و تأمین قطعات با کیفیت می باشد.

لازم به ذکر است که این پروژه توسط LABVIEW 2011 طراحی گشته و از ماژول کاربردی صنعتی SIM900 استفاده شده است. مدار این برد بر اساس دیتاشیت ماژول SIM900 طراحی گشته و جهت دریافت اطلاعات بیشتر از مدار دستگاه می توانید به دیتاشیت این ماژول مراجعه نمایید.

بیشتر بحث بر روی برنامه LABVIEW بوده و به علت استفاده از برد آماده، فقط اشاره ای به نحوه ارتباط و راه اندازی ماژول خواهیم نمود.

در این برنامه می خواهیم از طریق برنامه LABVIEW با ماژول SIM900 ارتباط برقرار کرده و پیام متنی را به شماره مورد نظر ارسال نماییم.

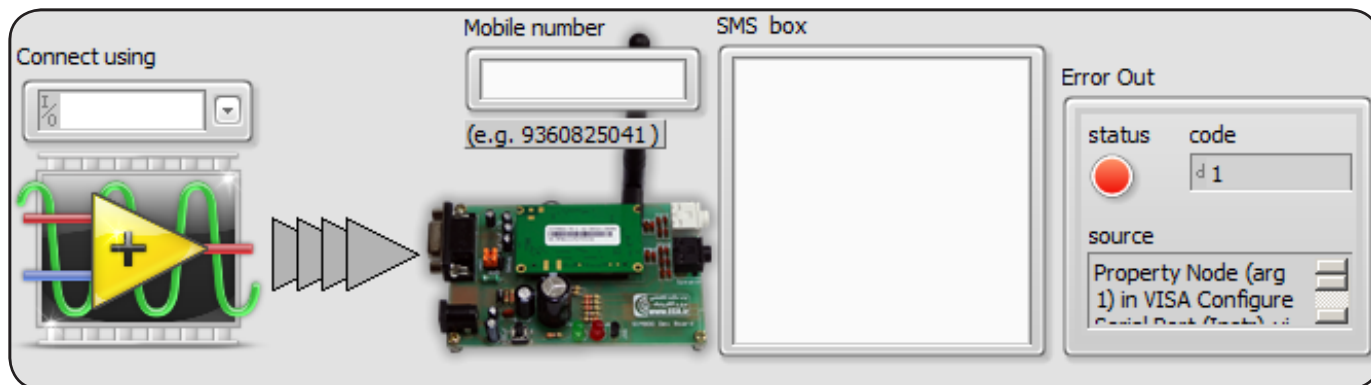
نحوه ارتباط با این ماژول از طریق سریال بوده و شما می بایست برای انجام هر کاری دستور مورد نظر را به ماژول ارسال نمایید. در صورتی که از یک برنامه ترمینال سریال (همانند هایپرترمینال) استفاده نمایید، دستورات ارسال SMS در چند مرحله می باشد که در زیر مراحل نوشته شده اند.

مرحله اول، ارسال دستور "AT+CMGS="+98000000000



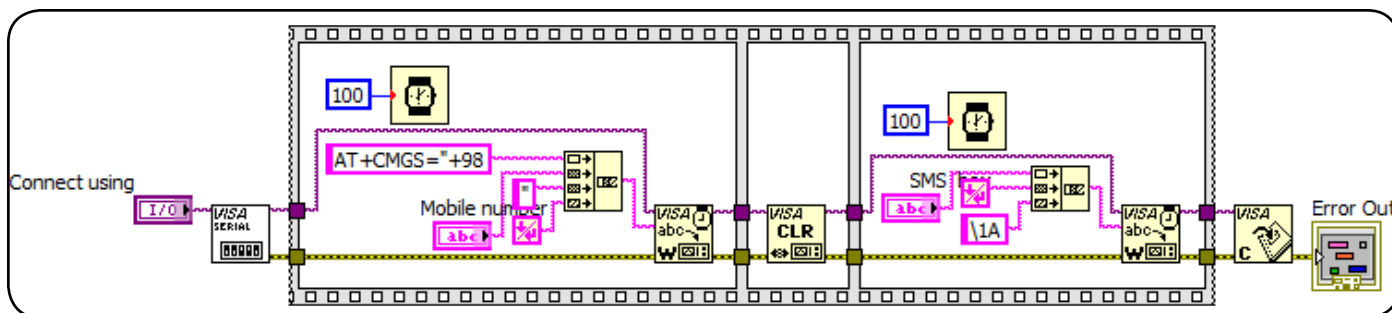
بوده که در جواب این دستور، ماژول کارکتر "<" را ارسال می کند. شما در این مرحله باید متن خود را ارسال و بعد از زدن ENTER برای ارسال SMS دکمه های Ctrl+Z را فشار دهید. اکنون همین مراحل را توسط برنامه LABVIEW پیاده سازی می کنیم.

برای برقراری سریال توسط نرم افزار Labview باید تولکیت Visa نصب شده باشد. بعد از نصب این تولکیت، توابع مورد نیاز برای کار کردن با پورت سریال در اختیار شما خواهند بود. شکل شماره ۱ برنامه نوشته شده برای برقراری ارتباط با ماژول SIM900 است.



شکل شماره ۱: برنامه نوشته شده front panel

همانطور که در تصویر front panel مشاهده می کنید، برنامه دارای چند قسمت ساده می باشد. شما در قسمت Connect using نام پورت COM که استفاده می کنید را مشخص می کنید. در قسمت Mobile number شماره موبایلی که قصد ارسال پیامک را دارید و در قسمت SMS box متن پیامک خود را وارد خواهید کرد. در صورتی که تمامی قسمت ها را به درستی پر کرده باشید با زدن دکمه play مراحل برنامه یک بار اجرا شده و SMS شما ارسال خواهد شد. قسمت اصلی برنامه یعنی بلوک دیاگرام آن در شکل شماره ۲ نشان داده شده است.



شکل شماره ۲: بلوک دیاگرام کلی برنامه

توابع اصلی برای برقراری ارتباط با نام VISA در تصویر مشخص بوده و بقیه توابع از توابع معمولی Labview هستند. برای اینکه مراحل به ترتیب اجرا شوند، تمام برنامه در یک Flat sequence Structure نوشته شده است. اولین قسمت برنامه، تابع VISA SERIAL است که در این تابع تمامی مشخصات پورت سریالی که می خواهید از طریق آن به دستگاه متصل شوید وجود دارد. در این برنامه ما فقط از قسمت VISA resource name خروجی گرفته ایم و مابقی قسمت ها بدون تغییر و در حالت پیش فرض قرار داده شده اند. در صورتی که احتیاج به تغییر baud rate و یا دیگر مقادیر داشتید می توانید با دابل کلیک بر روی این تابع به مشخصات آن دسترسی یابید.

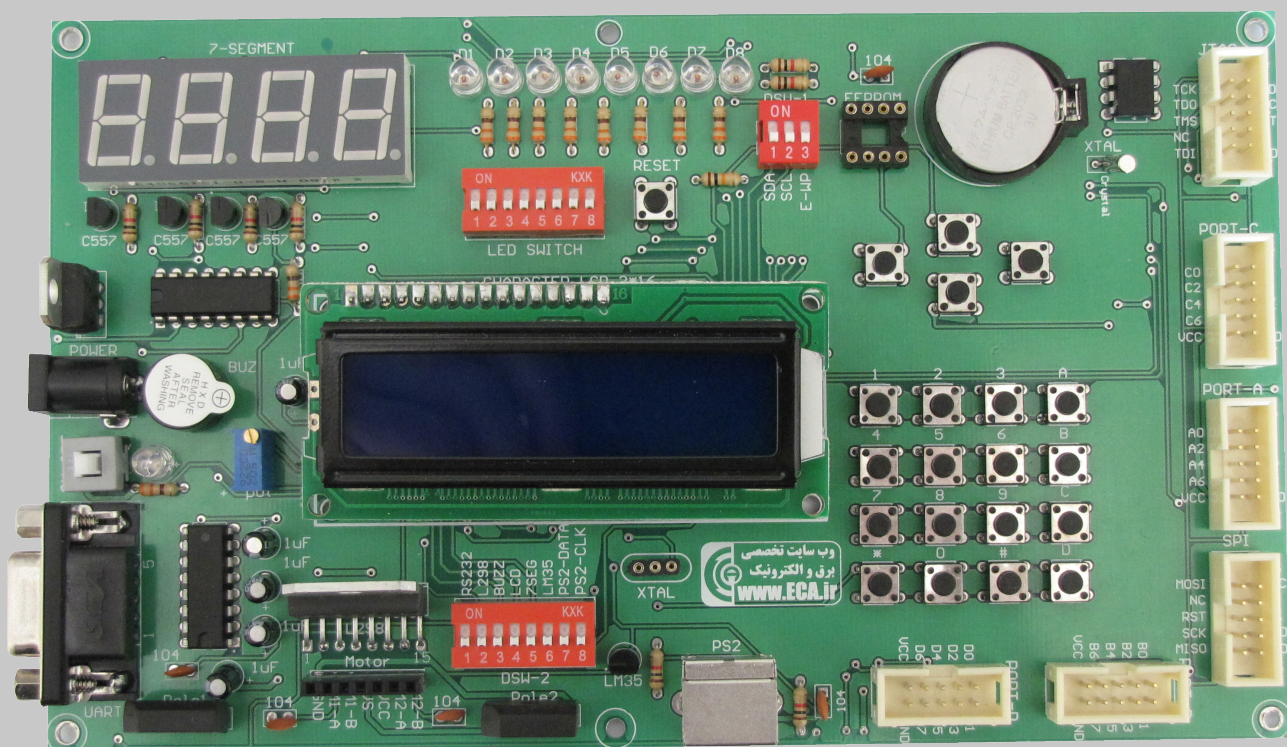
در این برنامه ۲ بار از تابع VISA abc W استفاده شده است. وظیفه این تابع ارسال اطلاعات شما از طریق پورت سریال است. این تابع از چند ورودی و چند خروجی تشکیل شده است. ورودی های که در این برنامه استفاده شده اند شامل VISA resource name و write buffer می باشند.

ورودی VISA resource name به VISA SERIAL وصل شده و تنظیمات سریال را دریافت کرده و ورودی write buffer به یک باکس string متصل می گردد. در مرحله اول ما می خواهیم شماره موبایل را از کاربر دریافت و به ماژول ارسال نماییم. تنها متغیر این قسمت شماره موبایل بوده و بقیه دستورات ثابت هستند، به همین منظور بقیه دستورات بصورت ثابت درون بلوک دیاگرام قرار داده شده است. برای اینکه چند متن string را به هم چسبانده و به تابع VISA Write ارسال کنیم، از تابع Concatenate Strings استفاده شده است. این تابع چند متن string را به یکدیگر چسبانده و در خروجی یک string کامل را ارائه می دهد. همانطور که در برنامه مشاهده می کنید، این تابع ۴ تکه، که تکه اول دستور AT+CMGS="+98 و سپس شماره موبایل که کاربر وارد می نماید و سپس علامت " و سپس Enter را به هم متصل و به تابع سریال ارسال می کند. دستور خروجی این تابع به صورت Enter AT+CMGS="+980000000000 ارسال می شود. عدد +98 پیش شماره شبکه مخابراتی ایران بوده و هر کشور عدد مخصوص به خود را دارد.

مرحله بعدی اجرای تابع VISA Clear است. وجود این تابع در برنامه الزامی نیست و فقط برای پاک نمودن

اطلاعات قبلی در بافر سریال می باشد. در صورتی که این تابع قرار نگیرد، به هنگام ارسال SMS دستور قبلی در قسمت بعدی متن SMS از کاربر دریافت و همانند قسمت اول از طریق تابع Concatenate به هم متصل و از طریق تابع VISA Write ارسال می شود. عبارت 1A مقدار متناظر Ctrl+Z برای ارسال دستور SMS می باشد. مرحله نهایی استفاده از تابع VISA Close است. توسط این تابع ارتباط سریال بسته شده و برنامه متوقف می شود. در ۲ مرحله از تابع تأخیر استفاده شده است. در صورتی که ارتباط سریال شما یکطرفه باشد، بهتر است از این توابع تأخیر جهت درست اجرا شدن برنامه استفاده نمایید. دیگر دستورات SIM900 را به همین نحو می توانید استفاده نمایید. برای دریافت تمامی دستورات SIM900 می توانید به فایل های ضمیمه مراجعه نمایید.

AVR Microcontrollers Training Board



امکانات دستگاه :

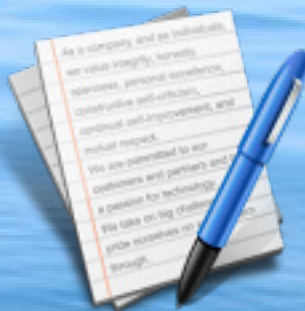
۱. میکرو کنترلر ATMEGA 32A
۲. LCD کاراکتری 2x16
۳. درایور موتور L298
۴. پورت سریال RS232
۵. پورت PS2
۶. ۸ عدد LED
۷. 7SEGMENT ۴ تایی
۸. IC ساعت DS1307 و باتری بک آپ
۹. EEPROM
۱۰. باز
۱۱. صفحه کلید 4x4
۱۲. شستی برای استفاده از اینترنت
۱۳. سنسور دما LM35
۱۴. پروگرامر STK200/300
۱۵. خروجی تمامی پورت ها به صورت BOX 2x5
۱۶. پورت ISP و JTAG جهت پروگرام کردن و دیباگ نمودن میکروکنترلر

توضیحات تکمیلی : <http://eshop.eca.ir/link/748.php>

آنچه شما در داندلود سنتر ECA خواهید یافت :

مقالات علمی، آموزشی

مقالات علمی و آموزشی در تمام زمینه های برق و الکترونیک و علوم مرتبط
آموزش زبان های برنامه نویسی، طراحی مدارات، تکنولوژی های نوین و ...
آموزش نرم افزارهای تخصصی برق و الکترونیک



پروژه های تکمیل شده

پروژه های کاملاً عملی و تست شده به همراه توضیحات کامل در زمینه های تخصص برق و الکترونیک
میکروکنترلرهای AVR، PIC و ARM
مدارهای مجتمع خطی، مدارات آنالوگ، کنترل، مخبرات و ...



سوالات کنکور

مجموعه سوالات کنکورهای کارشناسی ناپیوسته، کارشناسی ارشد و دکترا
نمونه سوالات آزمون های ورودی و استخدامی
دانشگاه های سراسری، آزاد و پیام نور



جزوات دانشگاهی

مجموعه جزوات درسی در مقاطع مختلف و دانشگاه های مختلف کشور
گزارش کارهای آزمایشگاهی
جزوات کنکور و درسی پرترین اساتید کشور



کنفرانس های داخلی

مقالات معتبرترین کنفرانس های داخلی کشور
کنفرانس های کشوری و بین المللی برگزار شده در دانشگاه ها و ارگان های دولتی



دانلود ECA
www.Download.ECA.ir



POWER

Power Systems , Generation , Transmission and Distribution

برق قدرت

کاربرد نیمه هادی ها در روش های مختلف استفاده از انرژی خورشیدی



Yadboroghi@gmail.com
نویسنده: علی یادبروچی

انرژی فتوولتائیک

انرژی فتوولتائیک (انرژی خورشیدی) تبدیل نور خورشید به الکتریسیته از طریق یک سلول فتوولتائیک (PVS) می باشد، که بطور معمول یک سلول خورشیدی نامیده می شود. سلول خورشیدی وسیله ای است برای تولید مستقیم الکتریسیته از نور مرئی و به عبارت دیگر این سلول ها الکتریسیته (برق)، با قدرت مفید ایجاد می کنند. این سلول ها یک ابزار غیر مکانیکی هستند که معمولاً از آلایژ سیلیکون ساخته می شود. نور خورشید از فوتون ها یا ذرات انرژی خورشیدی ساخته شده است. این فوتون ها که مقادیر متغیر انرژی را شامل می شوند، درست مشابه با طول موج های متفاوت طیف های نوری هستند. وقتی فوتون ها به یک سلول فتوولتائیک برخورد می کنند، ممکن است منعکس شوند، مستقیم از میان آن عبور کنند و یا جذب شوند. فقط فوتون های جذب شده انرژی را برای تولید الکتریسیته فراهم می کنند. وقتی که نور خورشید کافی یا انرژی توسط جسم نیمه رسانا جذب شود، الکترون ها از اتم های جسم جدا می شوند. (به دلیل اینکه آخرین الکترون یک اتم با گرفتن انرژی فوتون به لایه بالاتر رفته و می تواند از میدان پروتون خلاص شده و آزادانه در نیمه رسانا حرکت کند). رفتار خاص سطح جسم در طول ساختن باعث می شود سطح جلویی سلول که برای الکترون های آزاد بیشتر پذیرش یابد. بنابراین الکترون ها بطور طبیعی به سطح مهاجرت می کنند.

زمانی که الکترون ها موقعیت N را ترک می کنند، سوراخ هایی شکل می گیرد که تعداد الکترون ها زیاد بوده و هر کدام یک بار منفی را حمل می کنند و به طرف جلو سطح سلول پیش می روند، در نتیجه عدم توازن بار بین سلول های جلویی و سطوح عقبی یک پتانسیل ولتاژ شبیه قطب های مثبت و منفی یک باتری ایجاد می شود. وقتی که دو سطح از میان یک راه داخلی مرتبط می شود، الکتریسیته جریان می یابد. سلول فتوولتائیک قاعده بلوک ساختمان یک سیستم PV است. سلول های انفرادی می توانند در اندازه هایی از حدود ۱ cm تا ۱۰ cm از این سو به آن سو متغیر باشند. معمولاً ولتاژ خروجی اسمی یک صفحه خورشیدی ۱۲ ولت است.

کاربرد سلول های فتوولتائیک

سلول های فتوولتائیک کاربرد بسیار ساده ای دارند. سلول های تکی برای فراهم کردن انرژی مورد نیاز دستگاه های کوچکتر مانند ماشین حساب های الکترونیکی به کار می روند. آرایه های فتوولتائیک، الکتریسیته ای را تولید می کنند که قابل بازیافت است و عمدتاً در موارد نبود سیستم انتقال و توزیع الکترونیکی به کار می رود. برای مثال می توان به محل های دور از دسترس، ماهواره های مدارگرد، کاوشگرهای فضایی و ساختمان های مخابراتی دور از دسترس اشاره کرد. علاوه بر این استفاده از این نوع انرژی امروزه در محل هایی که شبکه توزیع وجود دارد نیز استفاده می شود.

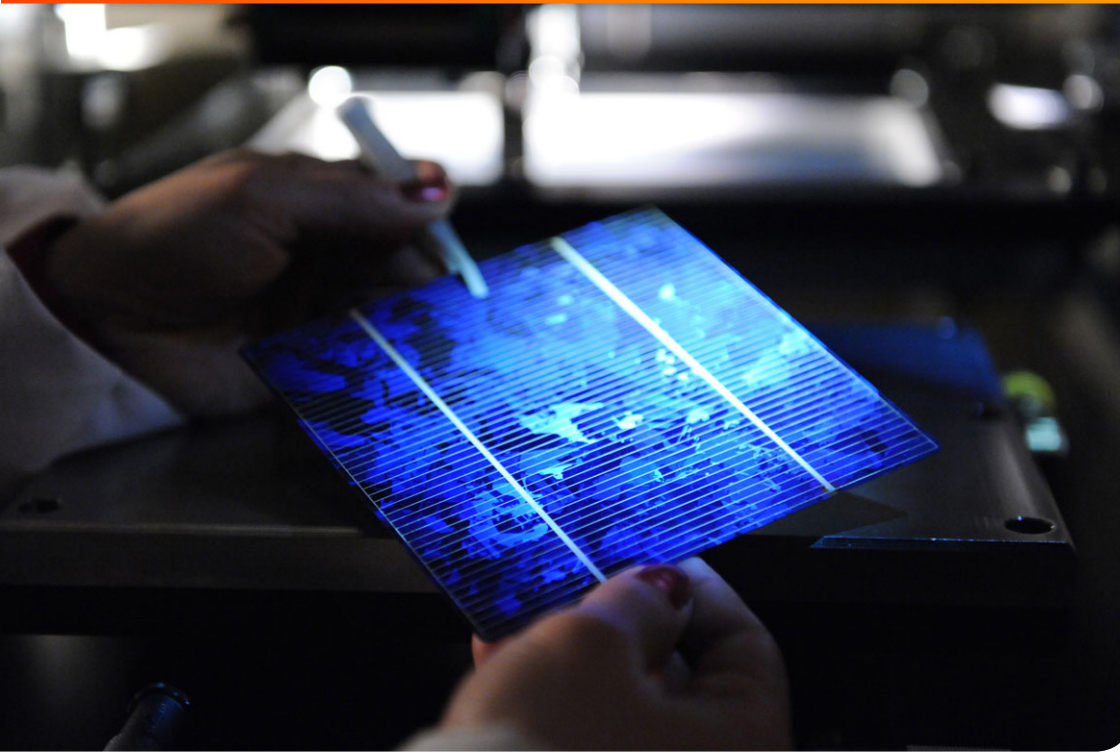
باتری خورشیدی

انرژی که از طریق خورشید به زمین می رسد ۱۰۰۰۰ بار بیشتر از انرژی مورد نیاز انسان است. در مرکز خورشید هر ثانیه ۷۰۰ تن هیدروژن به انرژی تبدیل می شود (به صورت فوتون یا نوترینو). دمای خورشید در مرکز آن ۱۵ میلیون و در سطح آن ۶ هزار درجه سانتیگراد است. انرژی تولید شده در سطح خورشید بعد از ۸ دقیقه به سطح زمین می رسد. نور خورشید که به زمین می رسد شامل طول موج های ۴۷ درصد فرو سرخ، ۴۶ درصد نور مرئی، ۷ درصد فرابنفش است. از این رو سلول های خورشیدی باید در ناحیه فرو سرخ و نور مرئی جذب بالایی داشته باشند. باتری های خورشیدی معمولاً از مواد نیمه رسانا و مخصوصاً سیلیسیم، تشکیل شده اند. هر اتم سیلیسیم با چهار اتم دیگر پیوند تشکیل می دهد و بدین صورت، شکل کریستالی آن پدید می آید. در باتری های خورشیدی به سیلیسیم مقداری جزئی ناخالصی اضافه می کنند. اگر اتم ناخالصی ۵ ظرفیتی باشد (اتم سیلیسیم ۴ ظرفیتی است)، آنگاه در ارتباط با چهار اتم سیلیسیم یک لایه ی آن بدون پیوند باقی می ماند (یک تک الکترون). به همین دلیل چون بار نسبی منفی پیدا می کند به آن سیلیسیم نوع N می گویند. در صورتی که اتم ناخالصی دارای ظرفیت ۳ باشد، آنگاه یک حفره اضافی ایجاد می شود. حفره را به گونه ای می توان گفت که جای خالی الکترون است، با بار مثبت (به اندازه الکترون) و جرمی برابر با جرم الکترون. که این امر هم باعث مثبت شدن نسبی ماده می شود و به آن سیلیسیم نوع P می گویند. هر باتری خورشیدی از ۶ لایه تشکیل شده که هر لایه را ماده ای خاص تشکیل می دهد.

عملکرد باتری خورشیدی

با اتصال یک نیمه هادی نوع P به یک نیمه هادی نوع N، الکترون ها از ناحیه N به ناحیه P و حفره ها از ناحیه P به ناحیه N منتقل می شوند. با انتقال هر الکترون به ناحیه P، یک یون مثبت در ناحیه N و با انتقال هر حفره به ناحیه N، یک یون منفی در ناحیه P باقی می ماند. یون های مثبت و منفی میدان الکتریکی داخلی ایجاد می کنند که جهت آن از ناحیه N به ناحیه P است. این میدان با انتقال بیشتر باربرها (الکترون ها و حفره ها)، قوی تر و قوی تر شده تا جایی که انتقال خالص باربرها به صفر می رسد. در این شرایط ترازهای فرمی دو ناحیه با یکدیگر هم سطح شده اند و یک میدان الکتریکی داخلی نیز شکل گرفته است. اگر در چنین شرایطی، نور خورشید به پیوند بتابد، فوتون هایی که انرژی آنها از انرژی شکاف نیمه هادی بیشتر است، زوج الکترون-حفره تولید کرده و زوج هایی که در ناحیه تهی یا حوالی آن تولید شده اند، شانس زیادی دارند که قبل از بازترکیب، توسط میدان داخلی پیوند از هم جدا شوند. میدان الکتریکی، الکترون ها را به ناحیه N و حفره ها را به ناحیه P سوق می دهد. به این ترتیب تراکم بار منفی در ناحیه N و تراکم بار مثبت در ناحیه P زیاد می شود. این تراکم بار، به شکل ولتاژی در دو سر پیوند قابل اندازه گیری است. اگر دو سر پیوند با یک سیم، به یکدیگر اتصال کوتاه شود، الکترون های اضافی ناحیه N، از طریق سیم به ناحیه P رفته و جریان اتصال کوتاهی را شکل می دهند. اگر به جای سیم از یک مصرف کننده استفاده شود، عبور جریان از مصرف کننده، به آن انرژی می دهد. به این ترتیب انرژی فوتون های نور خورشید به انرژی الکتریکی تبدیل می شود. هر چه میدان الکتریکی درون پیوند قوی تر باشد، ولتاژ مدار باز بزرگتری بدست می آید. برای دست یافتن به یک میدان الکتریکی بزرگ، باید اختلاف ترازهای فرمی دو ماده P و N از یکدیگر زیاد باشد. برای این منظور باید انرژی شکاف نیمه هادی بزرگ انتخاب شود. بنابراین ولتاژ مدار باز یک سلول خورشیدی با انرژی شکاف آن افزایش می یابد. اما افزایش انرژی شکاف سبب می شود، فوتون های کمتری توانایی تولید زوج الکترون-حفره داشته باشند و بنابراین جریان اتصال کوتاه کمتری نیز





تولید شود. بنابراین افزایش انرژی شکاف، روی ولتاژ مدار باز و جریان اتصال کوتاه سلول دو اثر متفاوت دارد.

فناوری های ساخت سلول های خورشیدی

در حال حاضر دو فناوری در ساخت سلول های خورشیدی غالب است: فناوری نسل اول و نسل دوم. فناوری نسل اول بر پایه ویفرهای سیلیکونی با ضخامت ۳۰۰-۴۰۰ میکرومتر است که ساختاری بلوری یا چند بلوری دارند که یا از بریدن شمش بدست می آیند یا از روش EFG و با کمک خاصیت موینگی رشد داده می شوند. فناوری نسل دوم یا

تکنولوژی لایه نازک، براساس لایه نشانی نیمه هادی روی بسترهای شیشه ای، فلزی یا پلیمری، در ضخامت های ۵-۳ است. هزینه مواد اولیه در تکنولوژی نسل دوم، پایین تر است و از آن گذشته، اندازه سلول تا ۱۰۰ برابر بزرگتر از اندازه سلول ساخته شده با تکنولوژی نسل اول است که مزیتی برای تولید انبوه آن محسوب می شود. در عوض بازدهی سلول های نسل اول، که اغلب سلول های بازار را تشکیل می دهند، به دلیل کیفیت بالاتر مواد، از بازدهی سلول های نسل دوم بیشتر است. انتظار می رود اختلاف بازدهی میان سلول های دو نسل با گذشت زمان کمتر شده و تکنولوژی نسل دوم جایگزین نسل اول شود.

در سال ۱۹۶۱، Shockley و Queisser با در نظر گرفتن یک سلول خورشیدی پیوندی به شکل یک جسم سیاه با دمای ۳۰۰ کلوین نشان دادند که بیشترین بازدهی یک سلول خورشیدی صرف نظر از نوع تکنولوژی بکار رفته در آن، ۳۰٪ است که در انرژی شکاف ۱.۱ eV یعنی انرژی شکاف گالیم آرسناید بدست می آید. بنابراین بازدهی سلول های خورشید نسل اول و دوم حتی در بهترین حالت نمی تواند از حوالی ۳۰٪ بیشتر شود. این در حالی است که حد کارنو برای تبدیل انرژی خورشیدی به انرژی الکتریکی ۹۵٪ است. و این مقدار تقریباً سه برابر بیشتر از بازدهی نهایی سلول های نسل اول و دوم است. بنابراین دستیابی به سلول هایی با بازدهی های دو تا سه برابر بازدهی های کنونی، امکان پذیر است. سلول های خورشیدی که دارای چنین بازدهی هایی باشند، نسل سوم سلول های خورشیدی نامیده می شوند. سلول های متوالی، سلول های خورشیدی چاه کوانتومی، سلول های خورشیدی نقطه کوانتومی، سلول های حامل داغ، نسل سوم سلول های خورشیدی را تشکیل می دهند.

کاربرد دیودها در سلول های خورشیدی با چیدمان های مختلف

همانطور که در قسمت های بالا ذکر شد، دیود یک قطعه الکتریکی است که به سبب یکسو سازی مسیر جریان الکتریکی در مدار استفاده می شود. دیود به عنوان یک قطعه الکتریکی داخل مدارها در دو نوع به کار گرفته می شود:

۱- سلول های تبدیل انرژی

۲- یکسو سازی جریان

در حقیقت صفحات خورشیدی شامل سلول های PV (فتوولتائیک) هستند که به صورت سری یا موازی بسته می شوند. هر سلول یک واحد تولید کننده الکتریسیته است. یک صفحه خورشیدی با ۳۶ سلول (هر سلول نزدیک به ۰.۵ ولت الکتریسیته تولید می کند) که به صورت سری به هم متصل اند، توانایی تولید ۱۸ ولت الکتریسیته را دارا هستند که برای شارژ کردن باتری های ۱۲ ولتی به کار می روند. اگر ظرفیت باتری ها بیشتر از ۱۲ ولت باشد می بایست چندین

صفحه خورشیدی به حالت سری با سیم به هم وصل کنیم تا ولتاژ مورد نیاز جهت پر کردن باتری ها افزایش پیدا کند. برای مثال دو صفحه متصل به هم که نزدیک به ۳۶ ولت برق تولید می کند برای شارژ کردن یک باتری ۲۴ ولتی استفاده می شود. اگر افزایش



شارژ را بخواهیم بر حسب آمپر بیان کنیم دو یا چند صفحه خورشیدی که به صورت موازی به همدیگر متصل می شوند می توانند مقدار آمپر خروجی را افزایش دهند.

هر سلول PV (فتوولتائیک) یکی از این دو عمل را انجام می دهد:

یا نیروی الکتریکی تولید می کند و یا نیروی الکتریکی تولید شده را مصرف می کند، که این امر بستگی دارد به این که سلول ها را در معرض نور خورشید قرار دهیم یا نه. همچنین دیودها در یکسو ساختن جریان های موجود در صفحات خورشیدی و یا باتری های خورشیدی که به صورت موازی بسته شده اند استفاده می شود. آنها (دیودها) از برگشت جریان الکتریکی موجود در سیم ها در حالت موازی ممانع می کنند؛ مانند دیودهای قدرت مصرف کننده.

تخلیه الکتریکی نیروی تولید شده نیز از طریق سری کردن باتری هایی که این نیرو در آن ها ذخیره شده، در شب و در زمانی که سلول ها به طور کامل در تاریکی هستند اتفاق می افتد.

همچنین در مواقعی که صفحات خورشیدی را به طور سری به مجموعه اضافه کنیم، از دیودها در یک محل انشعابی استفاده می کنیم. در حقیقت با وجود دیودها شاهد یک عایق بسیار قوی و مستحکم در مسیر اصلی جریان هستیم. در این جا دیود یک عایق داخلی است که زمانی که در معرض نور خورشید قرار می گیرد نقش خود را بسیار پر اهمیت تر از سلول های فتوولتائیک نشان می دهد. زمانی که سلول یا صفحه خورشیدی در حالت سری در مکانی باشد که نور خورشید به آن نرسد (و نیروی پتانسیل آن در حال مصرف باشد)، مقاومت و سختی سلول یا صفحه خورشیدی به طور شگرفی افزایش می یابد. در این حالت دیود می تواند با ایجاد یک مسیر باریک و انشعابی این مقاومت و سختی را به حداقل کاهش دهد. وقتی جریان الکتریکی بخواهد از مسیری باریک با مقاومت کم عبور کند، با اعمال نیرو مسیر خود را منحرف کرده و از دیود می گذرد و به صورت یک جریان متناوب پیرامون سلول فتوولتائیک را احاطه می کند تا به صفحه خورشیدی بعدی برسد و به همین ترتیب تا انتهای مدار پیش می رود. در این هنگام ولتاژ کاسته شده در حین مسیر دوباره از طریق رفع کردن سلول، فعال شده و افزایش می یابد.

PIC USB PROGRAMMER



جهت استفاده از میکرو کنترلر های PIC ساخت شرکت میکروچیپ شما به یک پروگرامر احتیاج خواهید داشت. روش های متعددی جهت پروگرام کردن میکروهای PIC از قبیل استفاده از پورت پرینتر و یا سریال وجود دارد، ولی در دنیای امروزی این پورت ها به علت کم بودن موارد مصرف دیگر بر روی رایانه های شخصی نصب نبوده و حتی بر روی رایانه های همراه نیز دیگر وجود نداشته و شما می بایست با خرید وسایل جانبی این پورت ها را نصب نمایید. ولی در عوض بر تعداد پورت های USB موجود بر روی رایانه ها افزایش یافته و امروزه رایانه ای بدون پورت USB موجود نمی باشد. این پروگرامر نیز بر اساس تکنولوژی USB طراحی گشته و به این معنی بوده که شما می توانید در هر رایانه از آن استفاده نمایید. در زیر خلاصه ای از مزایای این پروگرامر لیست شده است.

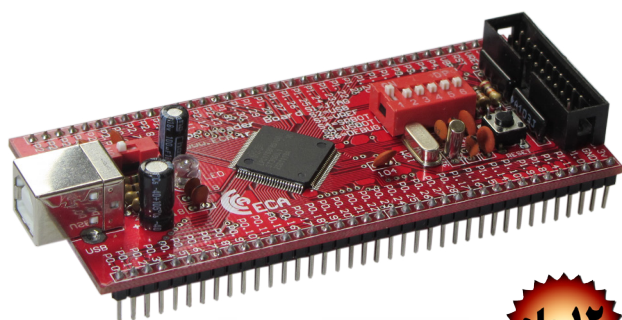
Device Programming Support

- PIC10
- PIC12
- PIC16
- PIC18
- PIC24
- dsPIC30
- dsPIC33
- PIC32
- KEELOQ® HCS
- 11 Series Serial EEPROM
- 24 Series Serial EEPROM
- 25 Series Serial EEPROM
- 93 Series Serial EEPROM
- MCP250xx CAN

- o هماهنگی با پورت های USB2.0
- o توانایی پروگرام نمودن تمامی میکرو کنترلر های PIC با بسته بندی DIP از طریق یک زیف سوکت
- o پشتیبانی از طریق نرم افزارهای مورد تأیید شرکت میکروچیپ
- o بدون نیاز به تغذیه خارجی (تغذیه از طریق پورت USB)
- o سوکت icsp به منظور پروگرام کردن انواع میکرو کنترلر های pic خارج از پروگرامر
- o قابلیت شناسایی و DETECT خودکار مدل میکرو کنترلر
- o قابلیت به روز رسانی Firmware به صورت خودکار
- o قابلیت پروگرام کردن میکرو های ۳.۳ ولتی
- o بدون نیاز به نصب درایور
- o سازگاری با تمامی سیستم عامل ها (Linux/Mac OS X /Windows)
- o دارای لایه محافظ زیر بورد
- o ۱۲ ماه گارانتی تعویض

توضیحات تکمیلی : <http://eshop.eca.ir/link/620.php>

LPC1768 Cortex-M3 Header Board



Cortex
Intelligent Processors by ARM®

۱۲ ماه
گارانتی تعویض

هدربرد (برد راه انداز) میکروکنترلرهای CORTEX M3 با امکانات اولیه جهت کار با میکروکنترلرهای LPC1768 شرکت فلیپس. با توجه به ۱۰۰ پایه بودن میکرو، دو ردیف پین هدر در اطراف برد قرار گرفته که بر روی ۲ بردبرد چسبیده به هم قابل قرارگیری است. از دیگر مزایای این برد راه انداز، تأمین ولتاژ مورد نیاز میکرو از طریق پورت USB به همراه کلید قطع وصل می باشد. یکی از امکانات ویژه این برد قابلیت پروگرام نمودن میکرو از طریق پورت USB می باشد. در این روش شما هیچ احتیاجی به سخت افزار خاصی نداشته و فقط از طریق وصل نمودن کابل USB به رایانه می توانید میکروکنترلر را پروگرام نمایید. جهت پروگرام کردن این میکرو می توانید از ۲ طریق ارتباط پورت USB و یا پورت JTAG اقدام نمایید.

❖ حداقل مدار جهت راه اندازی میکروکنترلر LPC1768

❖ بدون نیاز به پروگرامر، مجهز به بوت لودر USB

❖ دارای خروجی تمام پایه های ورودی خروجی به

ترتیب شمارشی

❖ دارای کانکتور full speed USB 2.0

❖ امکان نصب مستقیم برد بر روی بردبرد

❖ دارای خروجی ولتاژهای ۳.۳ و ۵ ولت

❖ دارای کلید قطع و وصل تغذیه

❖ امکان فعال و غیر فعال کردن پورت های تمامی امکانات جانبی میکرو، از قبیل

USB، JTAG، DEBUG، VREF و ...

❖ دارای خروجی کانکتور JTAG استاندارد

❖ دارای دکمه RESET

❖ کریستال 32.768KHZ برای راه اندازی RTC داخلی

❖ کریستال ۱۲ مگاهرتز برای راه اندازی میکرو کنترلر

توضیحات تکمیلی : <http://eshop.eca.ir/link/876.php>



❖ امکان فعال و غیر فعال کردن پورت های تمامی امکانات جانبی میکرو، از قبیل USB، JTAG، DEBUG، VREF و ...

❖ دارای خروجی کانکتور JTAG استاندارد

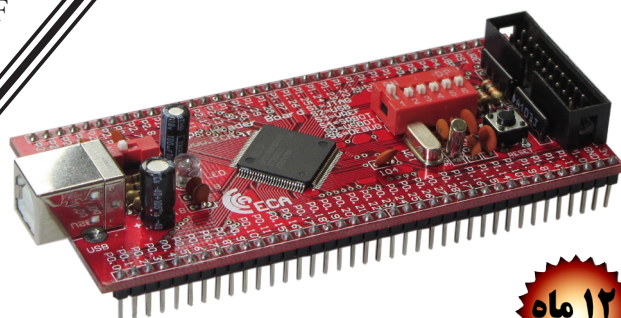
❖ دارای دکمه RESET

❖ کریستال 32.768KHZ برای راه اندازی RTC داخلی

❖ کریستال ۱۲ مگاهرتز برای راه اندازی میکرو کنترلر

توضیحات تکمیلی : <http://eshop.eca.ir/link/877.php>

LPC2368 Header Board



۱۲ ماه
گارانتی تعویض

❖ حداقل مدار جهت راه اندازی میکروکنترلر LPC2368

❖ بدون نیاز به پروگرامر، مجهز به بوت لودر USB

❖ دارای خروجی تمام پایه های ورودی خروجی به ترتیب شمارشی

❖ دارای کانکتور full speed USB 2.0

❖ امکان نصب مستقیم برد بر روی بردبرد

❖ دارای خروجی ولتاژهای ۳.۳ و ۵ ولت

❖ دارای کلید قطع و وصل تغذیه

❖ امکان فعال و غیر فعال کردن پورت های تمامی امکانات جانبی میکرو، از قبیل USB، JTAG، DEBUG، VREF و ...

❖ دارای خروجی کانکتور JTAG استاندارد

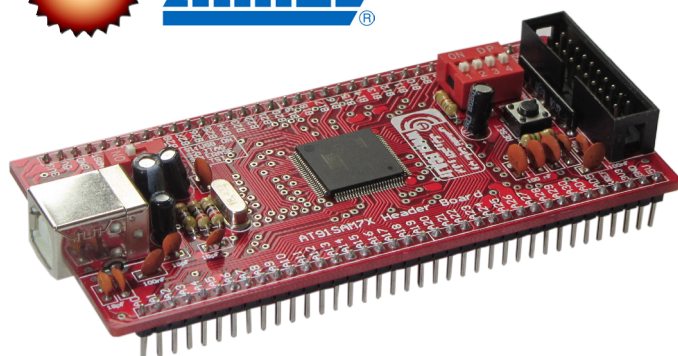
❖ دارای دکمه RESET

❖ کریستال 32.768KHZ برای راه اندازی RTC داخلی

❖ کریستال ۱۲ مگاهرتز برای راه اندازی میکرو کنترلر

توضیحات تکمیلی : <http://eshop.eca.ir/link/877.php>

AT91SAM7X256 Header Board



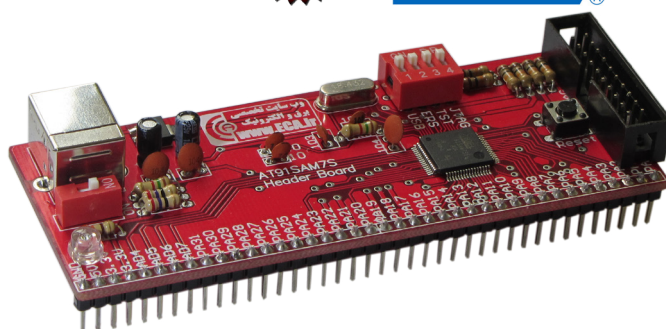
- ⊗ امکانات اولیه جهت راه اندازی میکروکنترلر AT91SAM7X256
- ⊗ بدون نیاز به پروگرامر
- ⊗ دارای خروجی تمام پایه های ورودی خروجی
- ⊗ امکان نصب مستقیم بر روی بردبرد
- ⊗ دارای خروجی ولتاژهای ۳.۳ و ۵ ولت
- ⊗ امکان استفاده از پروگرامر SAM-BA
- ⊗ دارای پورت JTAG
- ⊗ امکان کنترل پایه های TST, JTAGSEL و ERASE با استفاده از دیپ سوئیچ
- ⊗ دارای ۱۲ ماه گارانتی تعویض

توضیحات تکمیلی : <http://eshop.eca.ir/link/601.php>

AT91SAM7S256 Header Board

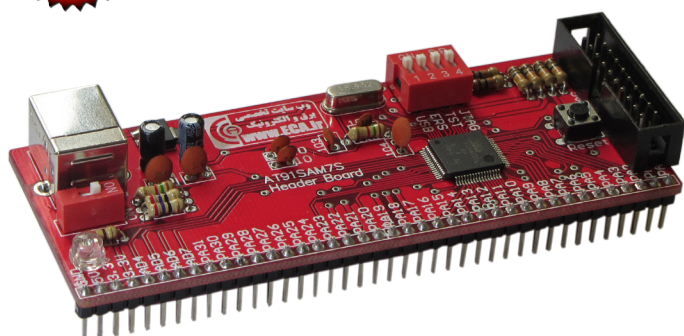


- ⊗ امکانات اولیه جهت راه اندازی میکروکنترلر AT91SAM7S256
- ⊗ بدون نیاز به پروگرامر
- ⊗ دارای خروجی تمام پایه های ورودی خروجی
- ⊗ امکان نصب مستقیم بر روی بردبرد
- ⊗ دارای خروجی ولتاژهای ۳.۳ و ۵ ولت
- ⊗ امکان استفاده از پروگرامر SAM-BA
- ⊗ دارای پورت JTAG
- ⊗ امکان کنترل پایه های TST, JTAGSEL و ERASE با استفاده از دیپ سوئیچ
- ⊗ دارای ۱۲ ماه گارانتی تعویض



توضیحات تکمیلی : <http://eshop.eca.ir/link/641.php>

AT91SAM7S64 Header Board



- ⊗ امکانات اولیه جهت راه اندازی میکروکنترلر AT91SAM7S64
- ⊗ بدون نیاز به پروگرامر
- ⊗ دارای خروجی تمام پایه های ورودی خروجی
- ⊗ امکان نصب مستقیم بر روی بردبرد
- ⊗ دارای خروجی ولتاژهای ۳.۳ و ۵ ولت
- ⊗ امکان استفاده از پروگرامر SAM-BA
- ⊗ دارای پورت JTAG
- ⊗ امکان کنترل پایه های TST, JTAGSEL و ERASE با استفاده از دیپ سوئیچ
- ⊗ دارای ۱۲ ماه گارانتی تعویض

توضیحات تکمیلی : <http://eshop.eca.ir/link/519.php>



قطعات الکترونیکی

Components

Active and Passive components

صفحه نمایش های

OLED



نویسنده: هوشیار مرادی
HOSHYARMORADI260@YAHOO.COM

OLED مخفف

Organic عبارت

Light Emitting Diode به

معنای دیود نوری از جنس مواد آلی

(ارگانیکی) می باشد. OLED نوعی صفحه

نمایش یا مانیتور است که از دو الکترود و یک لایه

نازک با پایه کربن تشکیل شده است. این لایه

نازک کربنی بین دو الکترود قرار می گیرد یکی از الکترودها که

کاتد است از جنس فلزبوده و الکترود دیگر یا آنده از جنس یک ماده شفاف مانند شیشه است. لایه نازک پایه کربن

از یک روزنه تزریق، یک لایه روزنه انتقال، یک لایه انتشار و یک لایه انتقال الکترود تشکیل شده است.

ساختار OLED

لایه های تشکیل دهنده OLED :

۱- لایه محافظ

۲- آنده: ترمینال منفی گسیل الکترود

۳- لایه ارگانیک: دو لایه رسانا و گسیل کننده نور

۴- کاتد: ترمینال مثبت گسیل الکترود

طرز کار OLED

OLED نیز مانند LED نور را تحت روندی به نام

الکتروفسفرواسنس گسیل می کنند.

این روند به صورت زیر تشریح می شود:

۱. ابتدا منبع برق اختلاف پتانسیلی در مدار OLED ایجاد

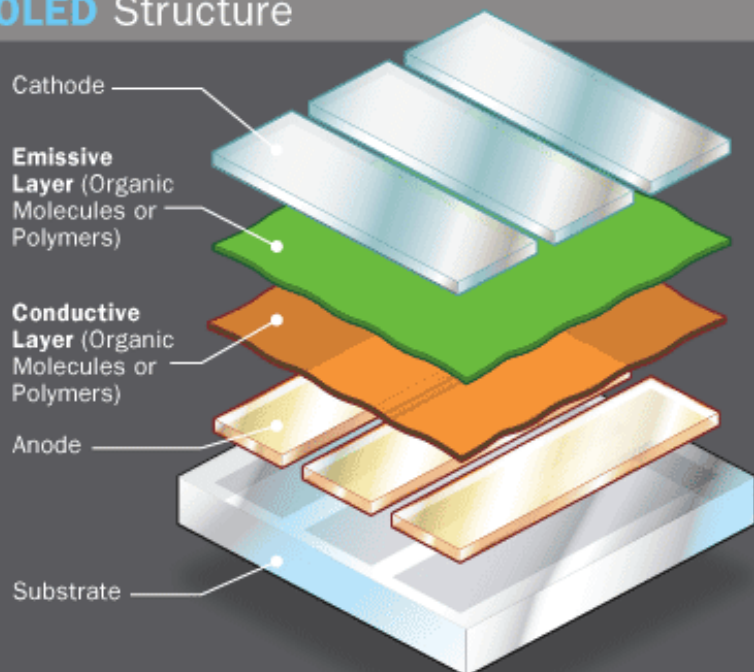
می کند.

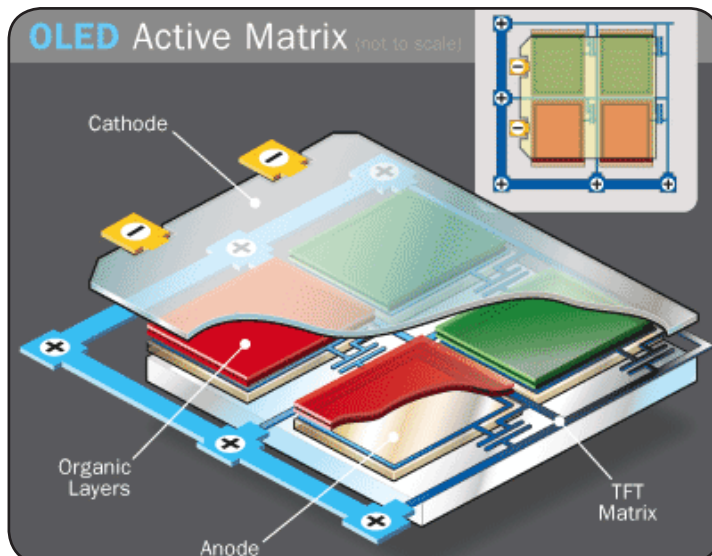
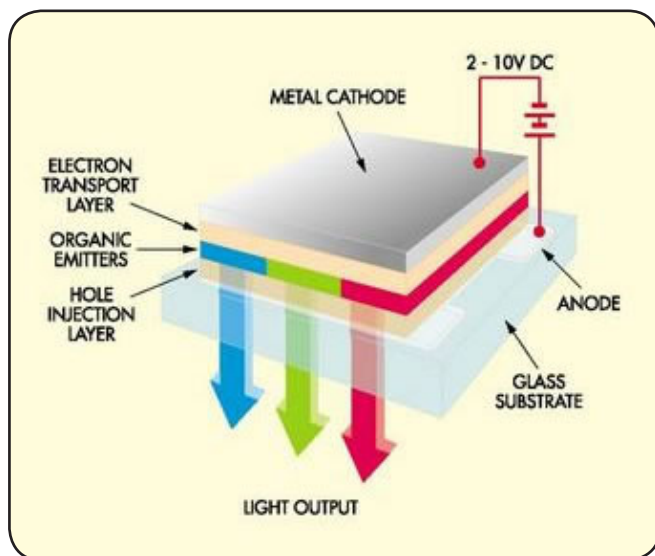
۲. جریان از کاتد به آنده در لایه های ارگانیک شارش

می کند.

۳. کاتد الکترودها را به لایه گسیل کننده می دهد.

OLED Structure





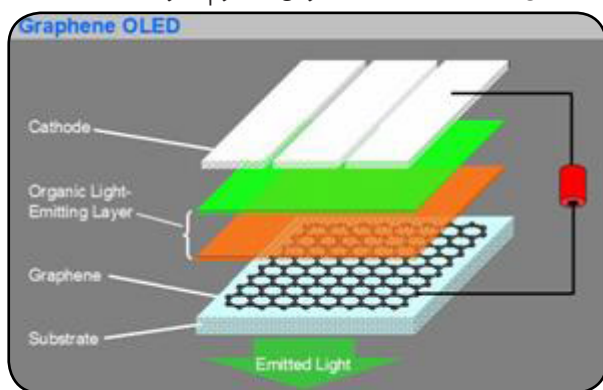
۴. آند الکترون‌هایی که به لایه رسانا می‌رسد را خارج می‌کند.

۵. در مرز بین لایه رسانا و گسیل کننده الکترون‌ها در فضاهای فاقد الکترون قرار می‌گیرد.

۶. رنگ نور نیز با توجه به جنس لایه‌های ارگانیک تعیین می‌شود.

فناوری صفحه نمایش OLED در اوایل سال ۱۹۸۰ به وسیله ایستمن کوداک اختراع شد. امروزه این فناوری در حال جایگزین شدن به جای فناوری LCD در تجهیزاتی مانند PDA ها و تلفن های موبایل است چون این نوع صفحه نمایش شفافتر، نازک تر و سریعتر است و نور بیشتری نسبت به مانیتورهای LCD تولید می کند و در عین حال مصرف برق آن هم پایین تر است به علاوه تولید انبوه آن ارزانتر تمام می شود.

کوچک بودن اجزای تشکیل دهنده سبب می شود که بتوان با استفاده از OLED کنتراست های بسیار بهتری ایجاد کرد. برای مثال قرار است که تلویزیون سونی با کنتراست 1000000:1 ارائه شود. اما مشکل بزرگ فعلی OLED گران بودن آن است. به طوری که فرایند تولید سخت و بسیار گران قیمت آن فعلا این فناوری را در دسترس عموم قرار نداده است. اما بعضی ها پیش بینی میکنند که ظرف ۵ سال آینده خیلی از ما در خانه خود تلویزیون و نمایشگر OLED خواهیم داشت.



صفحه نمایش های OLED جدید بر اساس گرافن

پژوهشگرانی از دانشگاه استنفورد با استفاده از گرافن به عنوان الکتروود شفاف توانستند به طور موفقیت آمیزی مدل کاملاً جدیدی از دیودهای نور-گسیل آلی (OLED) را توسعه دهند. این پیشرفت می‌تواند راه را برای تولید انبوه و ارزان OLED ها بر روی زیرلایه‌های پلاستیکی انعطاف پذیر با اندازه بزرگ و قیمت کم هموار کند به گونه‌ای که مانند کاغذ دیواری پیچیده شوند و در مواقع لزوم استفاده گردند.

محاسن و معایب نسبت به صفحه های نمایش LCD

با توجه به ساختار بسیار ساده پیکسل های OLED تولید اینگونه صفحه های نمایش بسیار ساده تر از صفحه های نمایش LCD یا Plasma است. OLED ها حتی می توانند بر روی سطوح بسیار ساده چاپ شوند و به همین دلیل هزینه تولید بسیار پایینتری نسبت به صفحه های نمایش LCD یا Plasma دارند.

بر همین اساس OLED ها را می توان بر روی سطوحی مانند مقوا یا لباس یا حتی کاغذهای معمولی چاپ کرد و صفحه های نمایشی بسیار نازک و زیبا را روانه بازار کرد.

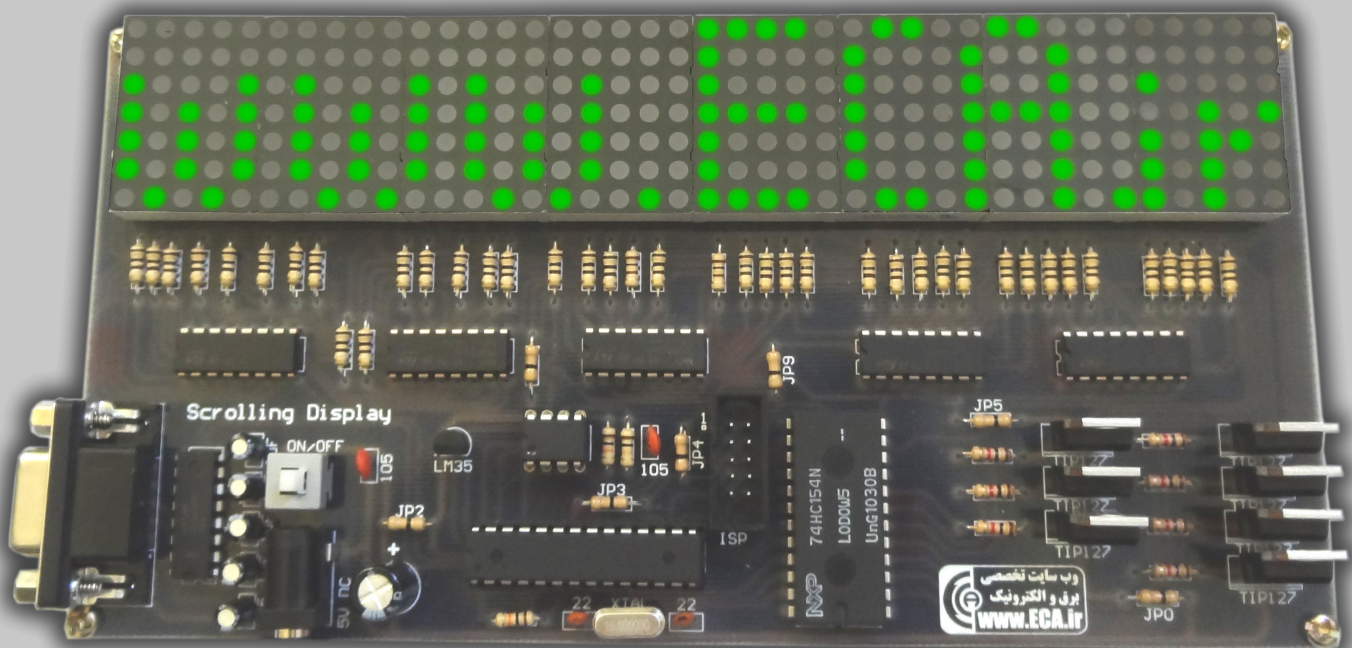
در OLED ها به دلیل اینکه هر پیکسل خود ساطع نور است، روشنایی و کنتراست بسیار بالاتری نسبت به صفحه های LCD را در اختیار کاربر قرار می دهند.

به دلیل سطوح بسیار زیاد انرژی و تفاوت میزان چگالی بارهای الکتریکی لایه Organic Emitters پیکسل های OLED قابلیت نمایش دامنه بسیار زیادی از رنگ ها را دارا می باشند.

به دلیل عدم نیاز به نور پیش زمینه پیکسل های OLED نسبت به LCD انرژی بسیار کمتری مصرف می کنند و برای دستگاه های قابل حمل بسیار مناسب هستند. OLED ها قابلیت مدیریت مصرف انرژی بسیار بهتری نسبت به LCD دارند، چون در رنگ های تیره به سادگی می توان میزان نور ساطع شده را کم کرد (یا خاموش کردن پیکسل هنگام استفاده از رنگ سیاه) اما در صفحات LCD چون همیشه نور پیش زمینه وجود دارد فقط با کم کردن میزان نور پیش زمینه می توان مصرف را پایین آورد.

چون نور ساطع شده توسط هر پیکسل مستقل از پیکسل های دیگر است زاویه دید این صفحات کاملاً ۱۸۰ درجه (۹۰ درجه از هر طرف) بوده و هیچ وقت رنگها از زاویه های مختلف اشتباه دیده نمی شوند. صفحات OLED زمان پاسخگویی بسیار پایینی دارند؛ در حال حاضر زمان پاسخگویی سریعترین صفحات LCD نزدیک به ۲ میلی ثانیه است در حالی که صفحات OLED معمولی زمان پاسخگویی حدود ۰,۰۱ میلی ثانیه دارند. اندازه پیکسل های OLED می تواند کوچکتر از پیکسل های LCD باشد به همین دلیل در یک صفحه می توان پیکسل های بیشتری را نسبت به LCD جا داد و صفحاتی با رزولوشن بالاتر تولید کرد. تنها عیب جدی صفحات OLED کم بودن طول عمر آنهاست. پیکسل های OLED فعلی حدود ۱۴۰۰۰ ساعت عمر می کنند. (۵۸۳ روز در صورت استفاده ۷/۲۴ یا ۳ سال و دو ماه در صورت استفاده ۱۲ ساعت در روز) در حال حاضر Panasonic و TOSHIBA توانسته اند طول عمر صفحات OLED را تا دو برابر زمان فعلی افزایش دهند و هنوز هم راه هایی برای بالا بردن طول عمر وجود دارد.

برد آموزشی تابلو روان

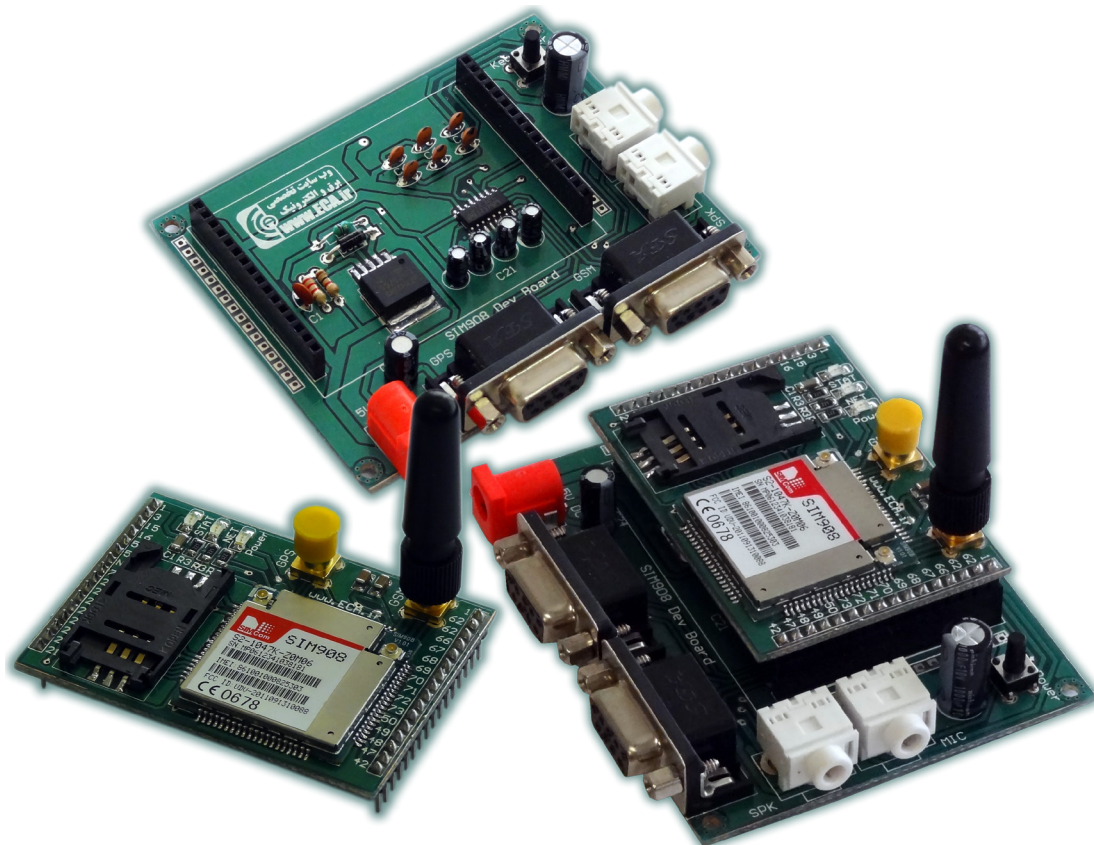


با توجه به زیاد شدن روز افزون استفاده از تابلوهای تبلیغاتی LED بردی جهت آموزش ساخت تابلوهای روان طراحی گردیده است تا ضمن آموزش صحیح ساخت انواع تابلو روان، بستری جهت ورود علاقمندان به بازار کار فوق ایجاد گردد. در این برد آموزشی سعی گردیده تا جدیدترین و بهینه ترین روش های مورد استفاده در بازار جهت نمایش متن و تصویر بر روی تابلو، آموزش و پیاده سازی شود.

- امکانات و محتویات برد آموزشی تابلو روان :
- ◀ صفحه نمایش ۴۰*۷ جهت نمایش نوشته
- ◀ سنسور دما جهت امکان نمایش دمای محیط بر روی تابلو
- ◀ امکان اتصال کریستال ساعت جهت نمایش ساعت و تاریخ دقیق
- ◀ درگاه سریال جهت اتصال به کامپیوتر
- ◀ EEPROM جهت ذخیره سازی اطلاعات
- ◀ امکان برنامه نویسی مستقیم توسط تمام کامپایلر های معتبر از قبیل Codevision و Bascom
- ◀ دارای پروگرامر STK300
- ◀ کابل سریال
- ◀ منبع تغذیه
- ◀ نمونه برنامه و فونت کامل فارسی و انگلیسی، برای راه اندازی تمامی بخش های برد

توضیحات تکمیلی : <http://eshop.eca.ir/link/1264.php>

برد کاربردی صنعتی SIM908



برد کاربردی صنعتی SIM908 در حقیقت یک تلفن همراه هوشمند است که فرمان های مختلف و متنوع دستگاه های طراحی شده را به درستی دریافت و ارسال می نماید و از سوی دیگر پاسخ های مربوطه را نیز از دستگاه یا کاربر انسانی دریافت کرده و به دستگاه های طراحی شده تحویل می دهد.

امکانی که این Gsm Modem را خاص نموده است دارا بودن GPS داخلی می باشد.

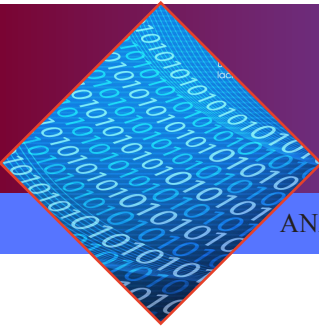
قابلیت های دستگاه:

محتویات محصول :

- ◀ برد کاربردی صنعتی SIM908
- ◀ ۱ عدد آنتن GSM
- ◀ ۱ عدد آنتن اکتیو GPS
- ◀ ۲ عدد کابل سریال
- ◀ DVD شامل راهنمای دستگاه

- ← ارسال اطلاعات هر دستگاه به سرعت و از راه دور
- ← ارسال و دریافت اطلاعات از طریق GPS,GPRS,GSM
- ← برخورداری از ماژول پیشرفته Sim908
- ← قابلیت کار کردن با تمامی اپراتورهای مخابراتی
- ← راه اندازی و استفاده فوق العاده آسان
- ← دارای کانکتورهای اسپیکر و میکروفن
- ← ارتباط ساده از طریق بسته RS232 با انواع سیستم ها و PLC ها
- ← تحقق رویای دسترسی سریع آسان و ارزان به اطلاعات دستگاه های مستقر در مکان های بسیار دور از محل کار مشتریان
- ← مجهز کردن دستگاه های الکترونیکی به یک موبایل هوشمند
- ← تهیه ردیاب دستی
- ← سرعت سنج های ماهواره ای
- ← مسافت و تنظیم ساعت های بسیار دقیق
- ← هماهنگ سازی زمان های عملکرد و دستگاه های توزیع شده در موقعیت های جغرافیایی مختلف
- ← ساخت انواع دستگاه های رهیب
- ← جاسازی دستگاه های صنعتی جهت کشف موقعیت جغرافیایی آنها
- ← ساخت ردیاب کودکان و سالمندان
- ← استفاده در پروژه های دانش بنیان

توضیحات تکمیلی : <http://eshop.eca.ir/link/1414.php>

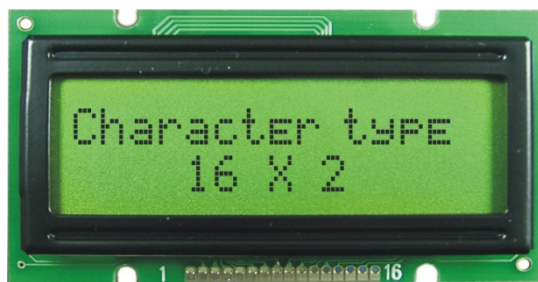


آموزش ارتباط با دات ماتریس درایور HD44780 و نوشتن کتابخانه برای LCD کاراکتری



نویسنده: علی پور ریکنده
alipoor90@gmail.com

شاید LCD کاراکتری یکی از پر کاربردترین user interface ها در مدارات میکروکنترلی باشد که بی شک مهمترین دلیل آن سادگی راه اندازی و برقراری ارتباط با این نوع نمایشگر می باشد. درایور موجود در این LCD ها غالباً کنترلر HD44780 ساخت هیتاچی بوده که وظیفه کنترلر مورد استفاده دریافت دستورات از میکروکنترلر و خاموش و روشن کردن دات ماتریس ها متناسب با دستور یا دیتای مورد نظر می باشد.



شکل شماره ۱

در این مقاله نحوه برقراری ارتباط با این LCD ها را فرا خواهید گرفت و می توانید کتابخانه دلخواه خود را برای کار با این نمایشگر طراحی کنید. نکته ۱: کدهایی که در این مقاله قرار داده شده در تمامی کمپایلرهای C استاندارد قابل استفاده است و مختص به کمپایلر خاصی نیست. نکته ۲: ما در این مقاله از ارتباط ۴ بیتی با LCD استفاده می کنیم چون ارتباط ۴ بیتی ۱۰ میلی ثانیه زمان بیشتر برای ارسال دستورات و دیتا نیاز دارد اما ۴ پایه کمتر از میکروکنترلر استفاده می شود.

6	E	فعال سازی LCD
7 - 14	D0 - D7	گذرگاه ۸ تایی اطلاعات و دستورالعمل
15	A	آند LED پس زمینه
16	K	کاتد LED پس زمینه

شماره پایه	سمبول	نحوه اتصال پایه
1	Vss	اتصال به زمین
2	Vdd	اتصال به ۵V+
3	VEE یا Vo	تنظیم کنتراست LCD
4	RS	کنترل رجیستر
5	RW	انتخاب مد خواندن یا نوشتن

جدول شماره ۱: شماره، اسم و وظیفه هر پین DCL

مروری بر پیکربندی سخت افزاری و pin out این نوع LCD ها

پین R/W وظیفه انتخاب مدکاری نمایشگر را دارد. یعنی می توان از طریق این پین و صفر و یک کردن آن اطلاعات را روی نمایشگر نوشت و یا از آن خواند. چون در این مقاله هدف ما فقط نوشتن روی نمایشگر می باشد، بنابراین پایه مورد نظر همواره به GND وصل می شود.

پین RS برای سوئیچ کردن بین رجیسترهای دستور و دیتا است که وقتی در حالت Low قرار بگیرد رجیستر دستور انتخاب می شود و برعکس.

پین EN برای لچ کردن و اجرای دستورات توسط درایور است.

از پین های DB4 تا DB7 نیز برای ارسال دیتا و دستورات استفاده می شود.

پین های DB0 تا DB3 را نیز برای جلوگیری از تاثیر نویز بر روی نمایشگر به زمین وصل می کنیم.

روند ارسال دیتا یا دستورات در مد ۴ بیتی به صورت زیر می باشد:

۱. پین EN را در وضعیت Low قرار می دهیم.

۲. با توجه به این که مورد ارسالی دستور است یا دیتا پین RS را در وضعیت مناسب قرار می دهیم (وضعیت High برای دیتا و وضعیت Low برای دستورات).

۳. ۴ بیت با ارزش دستور یا دیتای ارسالی را روی پایه های DB4 الی DB7 قرار می دهیم و سپس پین EN را در وضعیت High قرار می دهیم.

۴. بعد از ۱ میلی ثانیه پین EN را دوباره در وضعیت Low قرار می دهیم.

۵. ۱۰ میلی ثانیه صبر کرده و سپس ۴ بیت کم ارزش را روی پین های DB4 الی DB7 قرار می دهیم و سپس پین EN را به وضعیت High می بریم.

۶. بعد از ۱ میلی ثانیه پین EN را دوباره به وضعیت Low برمی گردانیم.

۷. ۱۰ میلی ثانیه برای اجرای دستورات یا نمایش دیتا صبر می کنیم.

نکته: منظور از علامت(*)، این است که وضعیت ست مه د نظر، تأثیر ندارد.

R/S	R/W	D7	D6	D5	D4	D3	D2	D1	D0	Instruction/Description
0	0	0	0	0	0	0	0	0	1	Clear Display and Home the Cursor
0	0	0	0	0	0	0	0	1	*	Return Cursor and LCD to Home Position
0	0	0	0	0	0	0	1	ID	S	Set Cursor Move Direction
0	0	0	0	0	0	1	D	C	B	Enable Display/Cursor
0	0	0	0	0	1	SC	RL	*	*	Move Cursor/Shift Display
0	0	0	0	1	DL	N	F	*	*	Set Interface Length
0	0	0	1	A	A	A	A	A	A	Move Cursor into CGRAM
0	0	1	A	A	A	A	A	A	A	Move Cursor to Display
0	1	BF	*	*	*	*	*	*	*	Poll the "Busy Flag"
1	0	D	D	D	D	D	D	D	D	Write a Character to the Display at the Current Cursor Position
1	1	D	D	D	D	D	D	D	D	Read the Character on the Display at the Current Cursor Position

جدول شماره ۲: دستورات و آدرس های درایور HD44780

نحوه عملکرد دستورات و چگونگی استفاده از آن ها

همانطور که در جدول شماره ۲ مشاهده می کنید در قسمت move directionset cursor اگر بیت S یک شود با ارسال هر کاراکتر محتویات قبلی روی LCD یک واحد به سمت راست منتقل می شوند.

همچنین اگر بیت ID یک شود با ارسال هر کاراکتر نشانگر یک خانه به سمت راست منتقل می شود. در دستور enable display/cursor اگر بیت D یک شود محتویات DDRAM روی صفحه، نمایش داده می شوند (یعنی LCD فعال می شود).

همچنین اگر بیت C یک شود نشانگر بر روی صفحه، نمایش داده می شود و در نهایت اگر بیت B یک شود نشانگر در حالت چشمک زن خواهد بود.

با یک کردن بیت SC در Move cursor/Shift Display شیفت دادن LCD در موقع نوشتن فعال می شود و بیت RL جهت آن را مشخص می کند که اگر یک شود جهت شیفت به سمت راست خواهد بود و اگر صفر شود جهت حرکت به سمت چپ خواهد بود.

بیت DL در رجیستر Set Interface Length مد ارتباط را مشخص میکند که برای ارتباط ۴ بیتی باید صفر شود. همچنین بیت N تعداد خطوط مورد استفاده را مشخص می کند که اگر صفر شود در LCD های دو خط از یک خط آن و در LCD های چهار خط از دو خط آن استفاده می شود و اگر یک شود از همه خطوط استفاده می شود. بیت F نیز سایز فونت مورد استفاده را مشخص می کند که برای استفاده از فونت ۵*۷ باید صفر شود. همچنین بیت های A برای ارسال آدرس و بیت های D برای ارسال دیتا می باشند. حال که با نحوه برقراری ارتباط و دستورات آشنا شدید وارد بخش نرم افزاری می شویم. نوشتن این کتابخانه بسیار ساده خواهد بود و فقط به یک تابع اصلی برای ارسال دستورات و دیتا نیاز داریم و بقیه توابع بر مبنای آن نوشته می شوند. البته اول به یک تابع برای ایجاد delay نیاز داریم.

```
#define cpuf 1 //define cpu frequency in mhz
void LCD_delay(unsigned char n)
{
    inti;
    for(i = 0; i < (n*1000*cpuf); i++);
}
```

حال تابع مربوط به ارسال دستور به صورت زیر خواهد بود که با استفاده از آن تابع ارسال دیتا را هم می نویسیم که در ابتدا نیاز است ماکروهایی برای تعیین پین های متصل به LCD نوشته شود. (xxpo مشخص کننده پورتی است که پین مورد نظر در آن وجود دارد مانند enpoPORTB که در آن پین EN نمایشگر، به پورت B نسبت داده شده و xxpi مشخص کننده پین مورد نظر از پورت انتسابی است مانند enpi0 که پین EN به پورت B و پین صفر آن متصل می شود. مزیت روش استفاده شده این است که هر پایه می تواند از یک پورت جداگانه باشد).

```
#define enpoPORTB //control lines port
#define rspo PORTB
#define db4po PORTA //data lines port
#define db5po PORTA
#define db6po PORTA
#define db7po PORTA
#define enpi0 //control lines pin number
#define rspi 1
#define db4pi 0 //data lines pin number
#define db5pi 1
#define db6pi 2
#define db7pi 3
////////////////////
void LCD_c(char command){
    enpo&= (~1 << enpi)^0xff; //clear en
    rspo&= (0 << rspi)^0xff; //clear rs
    //write high byte
    if((command&0x80) & db7po) db7po |= (1 << db7pi)^0x00; else db7po &= (1 << db7pi)^0xff;
    if((command&0x40) & db6po) db6po |= (1 << db6pi)^0x00; else db6po &= (1 << db6pi)^0xff;
    if((command&0x20) & db5po) db5po |= (1 << db5pi)^0x00; else db5po &= (1 << db5pi)^0xff;
    if((command&0x10) & db4po) db4po |= (1 << db4pi)^0x00; else db4po &= (1 << db4pi)^0xff;
    //execute
    enpo |= (1 << enpi)^0x00; //set en
    LCD_delay(1);
    enpo&= (~1 << enpi)^0xff; //clear en
    LCD_delay(10);
    //write low byte
    if((command&0x08) & db7po) db7po |= (1 << db7pi)^0x00; else db7po &= (1 << db7pi)^0xff;
    if((command&0x04) & db6po) db6po |= (1 << db6pi)^0x00; else db6po &= (1 << db6pi)^0xff;
    if((command&0x02) & db5po) db5po |= (1 << db5pi)^0x00; else db5po &= (1 << db5pi)^0xff;
    if((command&0x01) & db4po) db4po |= (1 << db4pi)^0x00; else db4po &= (1 << db4pi)^0xff;
    //execute
    enpo |= (1 << enpi)^0x00; //set en
    LCD_delay(1);
    enpo&= (~1 << enpi)^0xff; //clear en
    LCD_delay(10);
    db7po &= (1 << db7pi)^0xff;
    db6po &= (1 << db6pi)^0xff;
    db5po &= (1 << db5pi)^0xff;
    db4po &= (1 << db4pi)^0xff;
}
```

همچنین تابع ارسال دیتا را به صورت زیر تعریف می کنیم (منظور از دیتا کد کاراکتر مورد نظر می باشد).

```
void LCD_d(char data)
{
    rspo |= (1 << rs_pi) | 0x00; //set rs
    LCD_c(data);
    rspo &= (1 << rs_pi) ^ 0xff; //clear rs
}
```

قبل از ادامه کار تعدادی ماکرو و متغیر تعریف می کنیم که به آن نیاز خواهیم داشت:

```
#define lines 4 //1 to 4
#define columns 20 //8 to 20
#define auto_lower_line 1 //1 or 0
unsigned char LCD_x;
unsigned char LCD_y;
```

حال می توانیم از طریق این توابع دستورات و کاراکترها را به درایور ارسال کنیم.

اولین تابعی که نیاز است تابعی برای راه اندازی اولیه LCD است.

برای راه اندازی اولیه LCD باید اول ۳ بار دستور 0x03 را به نمایشگر ارسال کرد. سپس دستور 0x02 را برای پاک کردن DDRAM ارسال کرد و سپس برای انتخاب حالت ۴ بیتی و ۲ یا ۴ خط دستور 0x28 و برای حالت ۴ بیتی و ۱ خط دستور 0x20 را ارسال کرد و سپس دستورات لازم برای انتخاب حالت نشانگر و ... را ارسال کرد که در نهایت به صورت زیر خواهد شد:

```
void LCD_init(){
    LCD_c(0x03);
    LCD_c(0x03);
    LCD_c(0x03);
    LCD_c(0x02);
    if (lines == 1) LCD_c(0x20); else LCD_c(0x28);
    LCD_c(0x08);
    LCD_c(0x01);
    LCD_c(0x06);
    LCD_c(0x0f);
    LCD_x = 0;
    LCD_y = 0;
}
```

همچنین تابعی هم برای پاک کردن صفحه نمایش و بازگرداندن نشانگر به مکان اولیه نیاز است.

برای اینکار باید دستور 0x01 به درایور ارسال شود.

```
void LCD_clear()
{
    LCD_x = 0;
    LCD_y = 0;
    LCD_c(0x01);
}
```

یکی از دیگر توابع مورد نیاز یک تابع برای انتخاب حالت نشانگر است که دستورات مربوط به آن در صفحات قبل توضیح داده شد و این تابع به صورت زیر خواهد بود:

```
void LCD_cursor(char cursor)
{
    if (cursor == 'b')
        LCD_c(0x0f);
    if (cursor == 'f')
        LCD_c(0x0c);
    if (cursor == 'n')
        LCD_c(0x0e);
}
```


یک تابع نیز برای انتقال نشانگر نیاز داریم که بهتر است به جای انتقال نشانگر متغیرهای نگه دارنده مکان نشانگر را تغییر دهیم و در صورت نیاز به تغییر مکان نشانگر روی LCD میتوان از طریق توابع ارسال کاراکتر (که در ادامه بررسی میشوند) و ارسال کاراکتر (\0) NULL نشانگر را به مکان مورد نظر انتقال داد.

```
void LCD_gotoxy(char x, char y)
{
    LCD_x = x;
    LCD_y = y;
}
```

خوب حال تمام توابع کنترلی را تعریف کردیم و نوبت می رسد به توابع نمایشی. اول از همه باید یک تابع برای ارسال کاراکتر به LCD تعریف کنیم که مبنای دیگر توابع نمایشی نیز همین تابع خواهد بود.

برای ارسال یه کاراکتر به LCD باید اول یک دستور حاوی یک آدرس در حافظه DDRAM به درایور ارسال شود و بعد کد کاراکتر مورد نظر به صورت دیتا به درایور ارسال شود.

00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	10	11	12	13	14	15	16	17	18	19	← Character position (dec.)	
00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	10	11	12	13	14	15	16	17	18	19	← Row0 DDRAM address (hex)	
40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F	50	51	52	53	54	55	56	57	58	59	← Row1 DDRAM address (hex)	
14	15	16	17	18	19	1A	1B	1C	1D	1E	1F	20	21	22	23	24	25	26	27	28	29	2A	2B	2C	2D	← Row2 DDRAM address (hex)	
54	55	56	57	58	59	5A	5B	5C	5D	5E	5F	60	61	62	63	64	65	66	67	68	69	6A	6B	6C	6D	← Row3 DDRAM address (hex)	

جدول شماره ۳: آدرس خانه های MARDD

کدهای جدول شماره ۴ باید اول با 0x80، OR شوند و بعد به صورت دستور به درایور ارسال شوند و بعد از آن کد کاراکتر مورد نظر (که در واقع آدرس خانه های حافظه رام درایور هستند و دیتای آن کاراکتر در آنها ذخیره شده) که مطابق جدول اسکی نیز هستند به صورت دیتا ارسال شود.

Char. code	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
xxxx0000																
xxxx0001	!	1	A	Q	a	q	.	7	ç	4	ä	q				
xxxx0010	"	2	B	R	b	r	'	ı	ı	ı	ı	ı	ı	ı	ı	ı
xxxx0011	#	3	C	S	c	s	,	ı	ı	ı	ı	ı	ı	ı	ı	ı
xxxx0100	\$	4	D	T	d	t	,	ı	ı	ı	ı	ı	ı	ı	ı	ı
xxxx0101	%	5	E	U	e	u	.	ı	ı	ı	ı	ı	ı	ı	ı	ı
xxxx0110	&	6	F	V	f	v	.	ı	ı	ı	ı	ı	ı	ı	ı	ı
xxxx0111	'	7	G	W	g	w	.	ı	ı	ı	ı	ı	ı	ı	ı	ı
xxxx1000	(8	H	X	h	x	.	ı	ı	ı	ı	ı	ı	ı	ı	ı
xxxx1001)	9	I	Y	i	y	.	ı	ı	ı	ı	ı	ı	ı	ı	ı
xxxx1010	*	:	J	Z	j	z	.	ı	ı	ı	ı	ı	ı	ı	ı	ı
xxxx1011	+	;	K	[k	[.	ı	ı	ı	ı	ı	ı	ı	ı	ı
xxxx1100	,	<	L	¥	ı	ı	.	ı	ı	ı	ı	ı	ı	ı	ı	ı
xxxx1101	-	=	M]	m]	.	ı	ı	ı	ı	ı	ı	ı	ı	ı
xxxx1110	.	>	N	^	n	^	.	ı	ı	ı	ı	ı	ı	ı	ı	ı
xxxx1111	/	?	O	_	o	_	.	ı	ı	ı	ı	ı	ı	ı	ı	ı

جدول شماره ۴

می بینید که بعضی آدرس ها خالی هستند که مربوط به خانه های رزرو شده است و بعضی هم مربوط به حافظه CGRAM می باشند که در ادامه در مورد آن توضیح داده خواهد شد. تابع ارسال کاراکتر به صورت کد دست راست خواهد بود:

```
void LCD_putchar(char ch)
{
    if((LCD_x==columns)&&(LCD_y==0)&&(auto_lower_line))
    {LCD_y=1; LCD_x=0;}
    if((LCD_x==columns)&&(LCD_y==1)&&(auto_lower_line))
    {LCD_y=2; LCD_x=0;}
    if((LCD_x==columns)&&(LCD_y==2)&&(auto_lower_line))
    {LCD_y=3; LCD_x=0;}
    if(LCD_y==0){
        if(LCD_x==0) LCD_c(0x80);
        if(LCD_x==1) LCD_c(0x81);
        if(LCD_x==2) LCD_c(0x82);
        if(LCD_x==3) LCD_c(0x83);
        if(LCD_x==4) LCD_c(0x84);
        if(LCD_x==5) LCD_c(0x85);
        if(LCD_x==6) LCD_c(0x86);
        if(LCD_x==7) LCD_c(0x87);
        if(LCD_x==8) LCD_c(0x88);
        if(LCD_x==9) LCD_c(0x89);
        if(LCD_x==10) LCD_c(0x8a);
        if(LCD_x==11) LCD_c(0x8b);
        if(LCD_x==12) LCD_c(0x8c);
        if(LCD_x==13) LCD_c(0x8d);
        if(LCD_x==14) LCD_c(0x8e);
        if(LCD_x==15) LCD_c(0x8f);
        if(LCD_x==16) LCD_c(0x90);
        if(LCD_x==17) LCD_c(0x91);
        if(LCD_x==18) LCD_c(0x92);
        if(LCD_x==19) LCD_c(0x93);
    }
    if(LCD_y==1){
        if(LCD_x==0) LCD_c(0xc0);
        if(LCD_x==1) LCD_c(0xc1);
        if(LCD_x==2) LCD_c(0xc2);
        if(LCD_x==3) LCD_c(0xc3);
        if(LCD_x==4) LCD_c(0xc4);
    }
}
```

```

if(LCD_x==5) LCD_c(0xc5);
if(LCD_x==6) LCD_c(0xc6);
if(LCD_x==7) LCD_c(0xc7);
if(LCD_x==8) LCD_c(0xc8);
if(LCD_x==9) LCD_c(0xc9);
if(LCD_x==10) LCD_c(0xca);
if(LCD_x==11) LCD_c(0xcb);
if(LCD_x==12) LCD_c(0xcc);
if(LCD_x==13) LCD_c(0xcd);
if(LCD_x==14) LCD_c(0xce);
if(LCD_x==15) LCD_c(0xcf);
if(LCD_x==16) LCD_c(0xd0);
if(LCD_x==17) LCD_c(0xd1);
if(LCD_x==18) LCD_c(0xd2);
if(LCD_x==19) LCD_c(0xd3);}
if(LCD_y==2){
if(LCD_x==0) LCD_c(0x94);
if(LCD_x==1) LCD_c(0x95);
if(LCD_x==2) LCD_c(0x96);
if(LCD_x==3) LCD_c(0x97);
if(LCD_x==4) LCD_c(0x98);
if(LCD_x==5) LCD_c(0x99);
if(LCD_x==6) LCD_c(0x9a);
if(LCD_x==7) LCD_c(0x9b);
if(LCD_x==8) LCD_c(0x9c);
if(LCD_x==9) LCD_c(0x9d);
if(LCD_x==10) LCD_c(0x9e);
if(LCD_x==11) LCD_c(0x9f);
if(LCD_x==12) LCD_c(0xa0);
if(LCD_x==13) LCD_c(0xa1);
if(LCD_x==14) LCD_c(0xa2);
if(LCD_x==15) LCD_c(0xa3);
if(LCD_x==16) LCD_c(0xa4);
if(LCD_x==17) LCD_c(0xa5);
if(LCD_x==18) LCD_c(0xa6);
if(LCD_x==19) LCD_c(0xa7);}
if(LCD_y==3){
if(LCD_x==0) LCD_c(0xd4);
if(LCD_x==1) LCD_c(0xd5);
if(LCD_x==2) LCD_c(0xd6);
if(LCD_x==3) LCD_c(0xd7);
if(LCD_x==4) LCD_c(0xd8);
if(LCD_x==5) LCD_c(0xd9);
if(LCD_x==6) LCD_c(0xda);
if(LCD_x==7) LCD_c(0xdb);
if(LCD_x==8) LCD_c(0xdc);
if(LCD_x==9) LCD_c(0xdd);
if(LCD_x==10) LCD_c(0xde);
if(LCD_x==11) LCD_c(0xdf);
if(LCD_x==12) LCD_c(0xe0);
if(LCD_x==13) LCD_c(0xe1);
if(LCD_x==14) LCD_c(0xe2);
if(LCD_x==15) LCD_c(0xe3);
if(LCD_x==16) LCD_c(0xe4);
if(LCD_x==17) LCD_c(0xe5);
if(LCD_x==18) LCD_c(0xe6);
if(LCD_x==19) LCD_c(0xe7);}
LCD_d(ch);
LCD_x++;
}

```

همانطور که مشاهده میکنید در ابتدای تابع نیز چند شرط برای انتقال خودکار نشانگر به خط بعدی قرار داده شده است. برای نمایش متن روی LCD باید یه تابع تعریف کنیم که متن را گرفته و کاراکترهای آن را جدا کرده و به تابع نمایش کاراکتر بفرستد که این تابع به صورت زیر خواهد بود:


```
void LCD_puts(char *str)
{
    unsigned char i;
    for(i=0;*(str+i)!='\0';i++)
        LCD_putchar(*(str+i));
}
```

حال به آخرین بحث یعنی تعریف و نمایش کاراکتر دلخواه می‌رسیم. کاراکترهای 0x00 تا 0x07 که در جدول صفحات قبل نیز در مورد خالی بودن آن توضیح داده شده برای این منظور قابل دسترسی هستند و می‌توان ۸ کاراکتر دلخواه را در آن‌ها تعریف کرد.

در فونت ۵*۷ هر کاراکتر شامل ۷ کد است که در حالت باینری نشانگر روشن یا خاموش بودن نقطه متناظر آن است که برای تولید این کاراکترها می‌توانید از ابزار موجود در لینک زیر کمک بگیرید:

<http://www.quinapalus.com/hd44780udg.html>

برای تعریف این کاراکترها باید آدرس مورد نظر در CGRAM حافظه (0x00 تا 0x07) را به صورت دستور ارسال کرده و سپس یست سرهم ۷ کد مربوط به کاراکتر را به صورت دیتا ارسال کنیم که تابع آن به صورت زیر خواهد بود:

```
void LCD_defownchar(char adress,chara,charb,charc,chard,chare,charf,char g)
{
    LCD_c(adress|0X40);
    LCD_d(a);
    LCD_d(b);
    LCD_d(c);
    LCD_d(d);
    LCD_d(e);
    LCD_d(f);
    LCD_d(g);
}
```

و برای نمایش آن کاراکتر کافی است آدرس آنرا (0x00 تا 0x07) ارسال کنیم:

```
void LCD_putownchar(unsigned char adress)
{
    LCD_putchar(adress);
}
```

فایل نهایی برنامه در ضمیمه مجله موجود می‌باشد.

EPLAN Electric P8 2.1 SP1 Professional

به همراه فیلم‌های آموزشی



نرم افزار ePLAN P8 برنامه ای جامع و حرفه ای جهت طراحی و ایجاد نقشه های برقی است در واقع ePLAN P8 یک نرم افزار قدرتمند و باهوش برای طراحی نقشه های شماتیک برقی و تهیه دیگر مستندات لازم جهت اجرای یک پروژه بوده و با توجه به امکانات و تواناییهای فراوانی که در اختیار کاربران قرار می دهد توانسته است که در طی چندین سال اخیر به عنوان یک نرم افزار قابل اطمینان و قابل قبول در سطح جهانی حدود 8000 مشتری در 30 کشور جهان جذب نماید.

توضیحات تکمیلی :

<http://eshop.eca.ir/link/1386.php>

برد کاربردی - صنعتی GSM SIM900

برد کاربردی-صنعتی GSM SIM900 دارای تمامی امکانات لازم جهت استفاده کامل از ماژول SIM900 بوده و شما می توانید به راحتی آن را به رایانه یا میکروکنترلر متصل نمایید. در این برد کاربردی-صنعتی تمامی موارد ذکر شده در دیتاشیت SIM900 جهت راه اندازی این ماژول رعایت گشته و شما می توانید این برد را در تمامی مکان های صنعتی و پرنویز مورد استفاده قرار دهید. به دلیل استفاده از رگولاتور مخصوص این ماژول و خازن با ظرفیت بالا در طراحی مدار، شما می توانید با منابع تغذیه معمولی نیز این ماژول را راه اندازی نموده و دیگر لازم نیست که حتماً از منابع تغذیه با توان بالا استفاده نمایید.

یکی از قسمت های مهم این دستگاه قرار گرفتن ماژول SIM900-TE-C به جای ماژول خالی SIM900 بوده که به همین خاطر شما قادر به تعویض ماژول بوده و همچنین به جهت نویزگیری بالا و تکنولوژی ساخت بالای این برد، شما می توانید در هر مکانی از این دستگاه استفاده نمایید.

همچنین جهت افزایش کارایی دستگاه، از قطعات نویزگیر بر روی برد اصلی و آنتن مخصوص ضد نویز استفاده شده است.

با توجه به طراحی کامل و حرفه ای برد، این برد قابلیت استفاده در امور آموزشی را نیز دارا می باشد. شما توسط این برد می توانید به راحتی با ماژول SIM900 کار کرده و تمامی پروژه های خود را اجرا نمایید. از امکانات این برد می توان به موارد زیر اشاره نمود:

- اتصال مستقیم به رایانه از طریق درگاه سریال RS232
- اتصال مستقیم به میکروکنترلر
- امکان تغییر وضعیت دیاگ یا سریال ورودی میکروکنترلر
- خروجی بلندگو



توضیحات تکمیلی : <http://eshop.eca.ir/link/1276.php>

برد راه انداز GSM SIM900

قابلیت ها:

- ارسال اطلاعات هر دستگاه به سرعت و از راه دور
- ارسال و دریافت اطلاعات از طریق SMS, GPRS, GSM
- برخورداری از ماژول پیشرفته Sim900 (آپدیت شده و هماهنگ با سیستم مخابراتی کشور)
- قابلیت کار کردن با تمامی اپراتورهای مخابراتی (همراه اول، ایرانسل، تالیا، اسپادان، رایتل)
- راه اندازی و استفاده فوق العاده آسان
- ارتباط ساده از طریق بسته RS232 با انواع سیستم ها و PLCها
- تحقق رویای دسترسی سریع آسان و ارزان به اطلاعات دستگاههای مستقر در مکان های بسیار دور از محل کار مشتریان
- مجهز کردن دستگاههای الکترونیکی به یک موبایل هوشمند

خصوصیات :

این ماژول با هدف استفاده از GSM MODEM در پروژه های کاربردی مهیا گردیده است.

این ماژول دارای ابعاد بسیار کوچک (63x42mm) و قیمت مناسب می باشد که میتواند برای طراحی های مهندسی به عنوان یک ماژول مستقل در نظر گرفته شود.

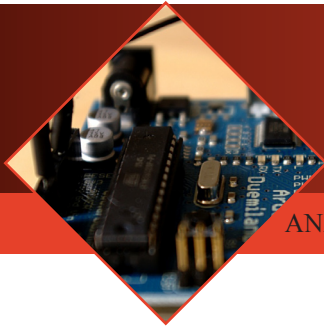
این ماژول دارای سوکت سیم کارت، کانکتور آنتن SMA و آنتن GSM مناسب می باشد.

دارای نمایشگر Power Net State که میتواند وضعیت ماژول را گزارش نماید.

خروجی N reset پایه (SIM900 16) نیز به عنوان ریست سخت افزار بیرون آمده است که در صورت نیاز میتوانید ماژول را ریست سخت افزاری نمایید.

توضیحات تکمیلی : <http://eshop.eca.ir/link/1413.php>

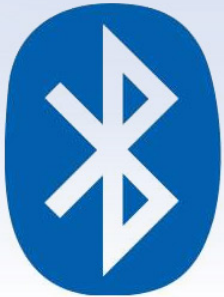




راه اندازی ماژول

HC-05

Serial – Bluetooth Module



Bluetooth

نویسنده : پرویز نظری

امروزه استفاده از بلوتوث برای ارسال و دریافت اطلاعات فراگیر شده و تقریباً همه ما حداقل با بلوتوث تلفن همراه خود یا بلوتوث کامپیوتر و ... کار کرده ایم. می توان گفت که اکثر تبادلات اطلاعاتی در یک فاصله معین، بین دو سخت افزار بدون استفاده از سیم ارتباطی، از طریق پروتکل بلوتوث قابل پیاده سازی می باشد.

فرض کنید می خواهیم یک ربات را از فاصله مثلاً ده متری کنترل کنیم، آیا میکروکنترلر مورد نظر شما دارای تکنولوژی بلوتوث می باشد؟ آیا برای تبادل اطلاعات با یک بلوتوث بر روی میکروکنترلر مورد نظر شما رجیستر و پین خاصی در نظر گرفته شده است؟ مثل پین های TXD و RXD در ارتباط UART یا MOSI، MISO و ... در ارتباط SPI. اگر جواب این سوال ها منفی است این مقاله می تواند برای برقراری ارتباط بی سیم از طریق بلوتوث مفید باشد.

بی شک برای برقراری چنین ارتباطی به سخت افزاری نیاز داریم تا بتوانیم از طریق آن دیتا را به بلوتوث داده و سپس از طریق پروتکل بلوتوث آن را به وسیله ی دیگر ارسال و یا دریافت کنیم.

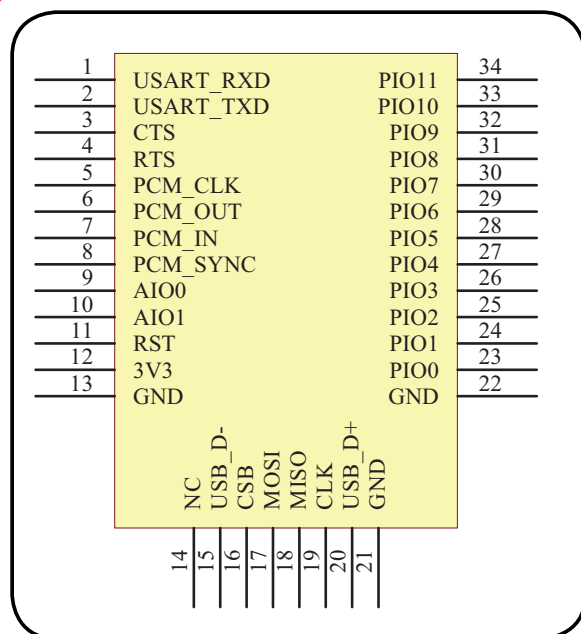
ماژول انتخابی که به معرفی آن خواهیم پرداخت ماژول HC-05 می باشد. توسط این ماژول می توان دیتا را در فواصل معین و با استفاده از پروتکل بلوتوث ارسال و دریافت کرد. مزیتی که ماژول مورد نظر دارا می باشد این است که می توان دیتا را به صورت سریال از طریق میکروکنترلر و یا کامپیوتر به ماژول انتقال داد و سپس خود ماژول دیتا را از طریق پروتکل بلوتوث به دستگاه و یا وسیله ی دارای بلوتوث ارسال خواهد کرد و همچنین می تواند داده های دریافتی از طریق پروتکل

بلوتوث را به صورت سریال تبدیل کرده و به کاربر تحویل دهد که این امر استفاده از ماژول را بی نهایت راحت کرده است. ارسال و دریافت اطلاعات به صورت سریال در ماژول از طریق پروتکل USART و با استفاده از پین های TX و RX صورت می گیرد. قبل از راه اندازی و شروع کار با ماژول بهتر است ابتدا با پایه های آن آشنا شویم.

حال با توجه به شکل شماره ۲ تعدادی از پایه های مهم ماژول را تشریح می کنیم؛ PIN1: پین مربوط به ارسال اطلاعات در



شکل شماره ۱



شکل شماره ۲: پایه های ماژول

ارتباط UART می باشد.
 PIN2: پین مربوط به دریافت اطلاعات در ارتباط UART می باشد.
 PIN11: اعمال ولتاژ صفر به این پین موجب رست شدن ماژول می شود.
 PIN12: تغذیه مثبت ماژول بوده و باید ولتاژ ۳٫۳ ولت به آن اعمال شود.
 PIN13: تغذیه منفی ماژول بوده و به زمین مدار وصل می شود.
 PIN31: فرکانس چشمک زدن LED متصل به این پایه بیانگر مد ماژول می باشد. (AT mode1 یا Atmode2)
 PIN32: خروجی LED متصل به این پایه قبل از برقراری ارتباط با دستگاه دیگر خاموش و پس از برقراری ارتباط روشن خواهد بود.
 اگر پین PIN34 در حالت high باشد تمامی دستورات AT com-mand قابل اجرا می باشد، در غیر این صورت تنها برخی از این دستورات اجرا خواهد شد.
 از جمله کاربرد این ماژول می توان به موارد زیر اشاره کرد:
 * ارتباط بین دو میکروکنترلر: در این حالت یکی از ماژول ها در حالت Slave و دیگری در حالت Master پیکربندی می شود.
 * ارتباط بلوتوث کامپیوتر یا تلفن همراه با میکروکنترلر: در این حالت ماژول به عنوان یک پورت مجازی بین بلوتوث کامپیوتر یا تلفن همراه و میکروکنترلر عمل می کند. در این کاربرد کامپیوتر یا تلفن همراه به عنوان Master و ماژول به عنوان Slave عمل می کنند.
 * می توان با پیکربندی این ماژول در حالت Master با دستگاه های دارای بلوتوث دیگر که در حالت Slave هستند (مانند پرینتر های دارای بلوتوث)، ارتباط برقرار کرد.
 این ماژول به صورت پیش فرض Slave تعریف شده است. اگر قصد ایجاد ارتباط با بلوتوث کامپیوتر یا تلفن همراه تان را دارید، این ماژول باید Slave تعریف شود (حالت پیش فرض)، اما اکثر دستگاه های دارای بلوتوث موجود در بازار، مثل پرینترها، به عنوان Slave عمل می کنند. برای برقراری ارتباط بین ماژول و چنین دستگاه هایی باید آن را به عنوان Master پیکربندی کرد. Salve یا Master بودن ماژول توسط دستورات AT commands مشخص می شود.
 منظور از AT commands چیست؟ AT commands یک سری دستورات هستند که از طریق ارتباط سریال به ماژول ارسال می کنیم و ماژول با دریافت آن ها، طبق نیاز کاربر تنظیم می شود.
 در جدول شماره ۱ یک سری از AT commands های مهم ذکر شده است.

Parameter	Response	Command	Application
None	OK	AT	Test
None	OK	AT+RESET	Reset
Get the soft version	AT+VERSION?	+VERSION: <Param> OK	Param: Version number
Restore default status	AT+ORGL	None	OK
Get module Bluetooth address	AT+ADDR?	+ADDR: <Param> OK	Param: Bluetooth address
Set/ inquire device's name	AT+NAME=<Param>	OK	Param: Bluetooth device name Default: "HC-05"
Set/ inquire device's name	AT+NAME?	1. +NAME:<Param> OK----success 2. FAIL----failure	Param: Bluetooth device name Default: "HC-05"
Get the remote Bluetooth device's name	AT+RNAME?<Param1>	1. +NAME:<Param2> OK----success 2. FAIL----failure	Param1: Remote Bluetooth device address Param2: Remote Bluetooth device address
Set/ inquire module role	AT+ROLE=<Param>	OK	Param: 0---- Slave role 1---- Master role 2---- Slave-Loop role Default: 0
Set/ inquire module role	AT+ROLE?	+ROLE:<Param> OK	

جدول شماره ۱

باید توجه داشت که هنگام ارسال دستور به ماژول در انتهای هر دستور باید کلید Enter (در صورت ارسال دستور از طریق کامپیوتر) و یا کد (0x0d) مربوط به این کلید (در صورت ارسال دستور از طریق میکروکنترلر) وارد شود.

برای مثال جهت Master بودن ماژول باید دستور "AT+ROLE=1\r\n" را دریافت کردیم یعنی ماژول در حالت Master قرار گرفته است.

توجه: \r\n همان فشردن کلید Enter یا نوشتن کد مربوط به این کلید در برنامه می باشد.

همچنین اگر در دستور فوق به جای عدد یک، صفر را ارسال کنیم این ماژول دوباره به عنوان Master عمل می کند.

این ماژول در دو مد AT mode1 و AT mode2 قابل استفاده است.

AT mode1 در این حالت Baud Rate = 38400 bps می باشد و LED متصل به پایه PIN31 با فرکانس ۲ هرتز چشمک می زند. برای وارد شدن به این حالت باید مراحل زیر را به ترتیب اجرا کرد:

۱. اتصال PIN34 به ولتاژ low

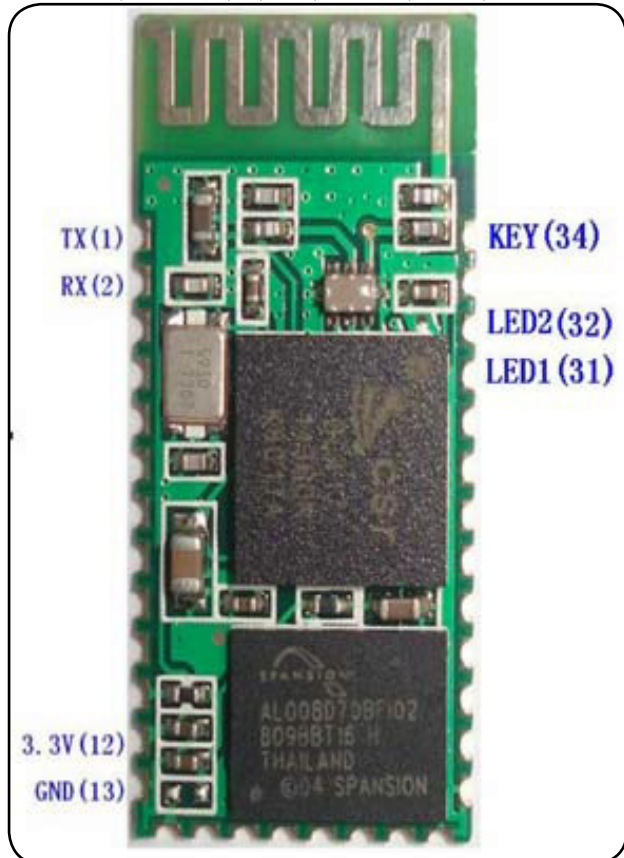
۲. اعمال تغذیه ماژول

۳. اتصال PIN34 به ولتاژ high

AT mode2 در این حالت Baud Rate = 9600 bps می باشد و LED متصل به پایه PIN31 با فرکانس ۱ هرتز چشمک می زند. برای وارد شدن به این حالت باید مراحل زیر را به ترتیب اجرا کرد:

۱. PIN34 را به PIN12 ماژول وصل می کنیم.

۲. با اعمال تغذیه ماژول این پین نیز به طور همزمان به سطح high می رود.

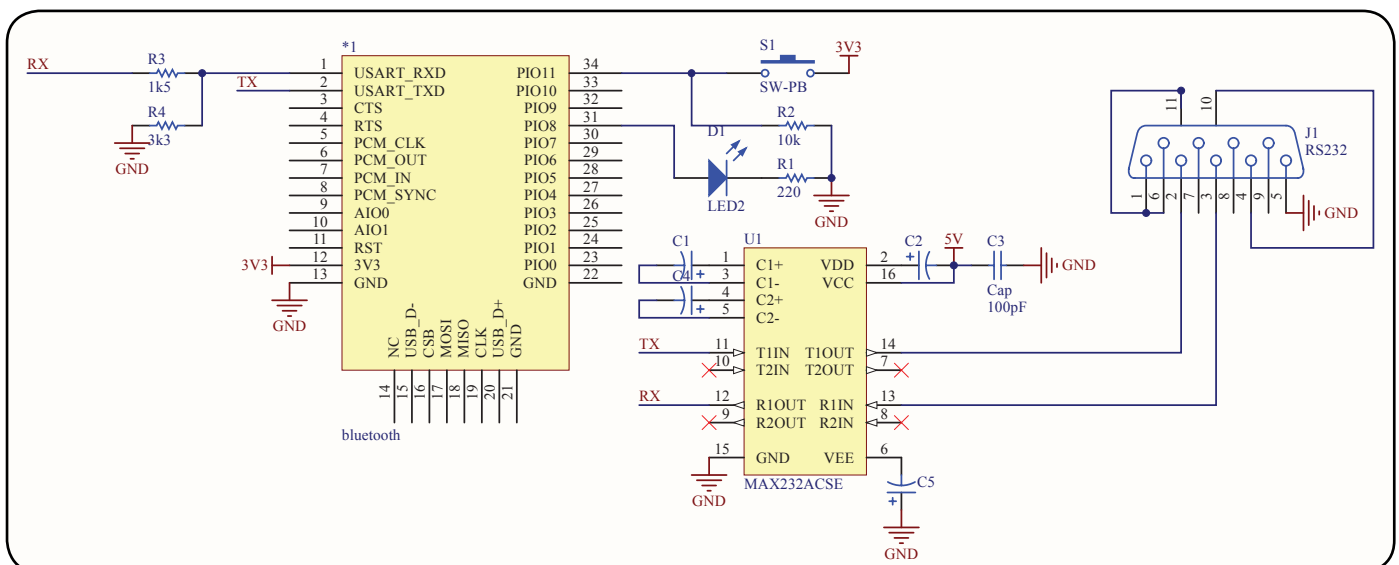


شکل شماره ۳

اتصال ماژول به کامپیوتر از طریق پورت com

پس از اطلاعاتی مختصری که در مورد ماژول و نحوه استفاده از آن ارائه شد، قصد داریم ماژول را از طریق پورت سریال به کامپیوتر وصل کنیم. در محیط نرم افزار Hyper Terminal دستورات AT commands را نوشته و با فشردن کلید Enter جواب ماژول به این دستورات را در محیط نرم افزار Hyper Terminal مشاهده کنیم.

ابتدا مدار را به صورت شکل شماره ۴ می بندیم:



شکل شماره ۴

توضیح شماتیک مدار

بخشی از مدار مربوط به مبدل RS232 به TTL می باشد. خروجی این قسمت از مدار TTL می باشد، درحالی که ماژول با ولتاژ ۳٫۳ ولت کار می کند. به همین دلیل برای وصل پایه TXD کامپیوتر به RXD ماژول از یک تقسیم مقاومتی استفاده کرده ایم تا ۵ ولت TTL را به ۳٫۳ ولت تبدیل کنیم. بین TXD ماژول هم مستقیماً به پین RXD مبدل وصل

می شود. چون ۳,۳ ولت برای TTL به عنوان سطح high قابل قبول است. کلید متصل به پین PIN34 برای انتخاب AT mode1 یا AT mode2 می باشد. در مورد این مدها قبلا بحث شد. فرکانس چشمک زدن LED نیز بیان گر کارکرد دستگاه در این مدها می باشد.

نحوه عملکرد برنامه

پس از وصل کردن ماژول به کامپیوتر، ماژول را به صورت زیر وارد مد AT mode1 می کنیم:

۱. اتصال PIN34 به ولتاژ Low

۲. اعمال تغذیه ماژول

۳. اتصال PIN34 به ولتاژ High

در این حالت LED با فرکانس ۲ هرتز چشمک می زند.

نرم افزار Hyper Terminal را باز کرده در قسمت Name نام دلخواهی را وارد می کنیم. سپس ok را می زنیم.

در پنجره باز شده، در قسمت connect using شماره پورتی را که بعد از وصل شدن ماژول اضافه شده است را انتخاب می کنیم. در قسمت configure تنظیمات زیر را اعمال می کنیم:

Baud Rate = 38400 bps , Data bits:8 , Parity:None , Stop bits: 1

و در نهایت Flow control: None سپس کلید ok را می زنیم و دوباره ok را کلیک می کنیم.

حال در محیط نرم افزار Hyper Terminal از منوی فایل گزینه Properties را می زنیم.

پنجره باز شده در تب settings روی ASCII Setup کلیک می کنیم و در پنجره باز شده

تیک گزینه های Echo typed characters locally و Append line feeds to incoming

line ends را می گذاریم. سپس دوباره بر روی ok کلیک می کنیم. حال با تایپ دستورات

AT command می توان جواب ماژول به این دستورات را مشاهده کرد.

توجه: برای تشخیص شماره پورت متصل به ماژول، روی computer کلیک راست کرده گزینه

Manage و سپس Device Manager را انتخاب کنید. در قسمت (Ports(COM & LPT) با

وصل کردن ماژول پورتی اضافه می شود که همان پورت مورد نظر می باشد.

توجه: در مورد نرم افزار Hyper Terminal در مجله نویز شماره ۱۱ بحث شده است. برای

اطلاعات بیشتر می توانید به این شماره مراجعه نمایید.

لیست کامل دستورات AT commands در دیتاشیت ماژول آورده شده است. در شکل شماره ۵

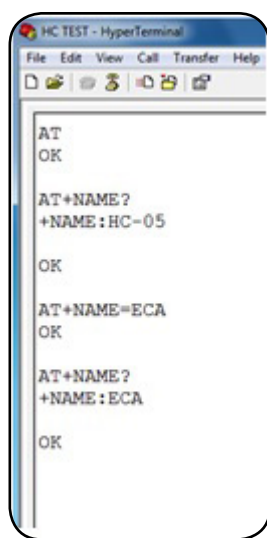
ارسال و دریافت چندین دستور نشان داده شده است.

با ارسال دستور AT، OK دریافت می شود. سپس نام بلوتوث را فراخوانی کرده ایم. و با دستور

بعدی نام بلوتوث را تغییر داده ایم.

برای آگاهی از سایر دستورات AT command و ERROR ها به دیتا شیت موجود در فایل

ضمیمه مجله مراجعه نمایید.



شکل شماره ۵

تبادل اطلاعات بین میکرو و کامپیوتر از طریق بلوتوث

هدف این پروژه ارسال یک رشته کاراکتر از میکرو به کامپیوتر و همچنین از کامپیوتر به میکرو، از طریق پروتکل بلوتوث می باشد.

ابتدا یک رشته کاراکتر از میکرو به صورت سریال به ماژول فرستاده می شود. سپس ماژول این دیتا را از طریق پروتکل

بلوتوث به بلوتوث لپ تاپ یا کامپیوتر شما می فرستد. این رشته دریافت شده در محیط نرم افزار Hyper Terminal

نمایش داده می شود. سپس یک رشته را در محیط این نرم افزار تایپ کرده و با فشردن کلید Enter این رشته از طریق

بلوتوث کامپیوتر به ماژول ارسال می شود. ماژول نیز رشته دریافتی را از طریق ارتباط سریال به میکرو می فرستد و در

نهایت این رشته در ال سی دی کاراکتری متصل به میکرو نمایش داده می شود.

میکروکنترلر مورد استفاده در این پروژه LPC1768، ساخت شرکت NXP می باشد.

شکل شماره ۴ شماتیک این پروژه را نشان می دهد. همان طور که مشاهده می شود چون میکروکنترلر و ماژول دارای ولتاژ

تغذیه ۳,۳ ولت می باشند، نیازی به تقسیم مقاومتی بر روی پین RXD ماژول نمی باشد.

پین های RXD و TXD از UART0 میکروکنترلر به ترتیب به پین های TXD و RXD از ماژول متصل شده اند. LCD به

پورت صفر میکروکنترلر متصل شده است. برای تنظیم کنتراست LCD نیز از یک پتانسیومتر استفاده کرده ایم.

پس ایجاد پروژه در نرم افزار Keil، فایل های lcd.c و system_LPC17xx.c پوشه پروژه کپی کنید و سپس فایل

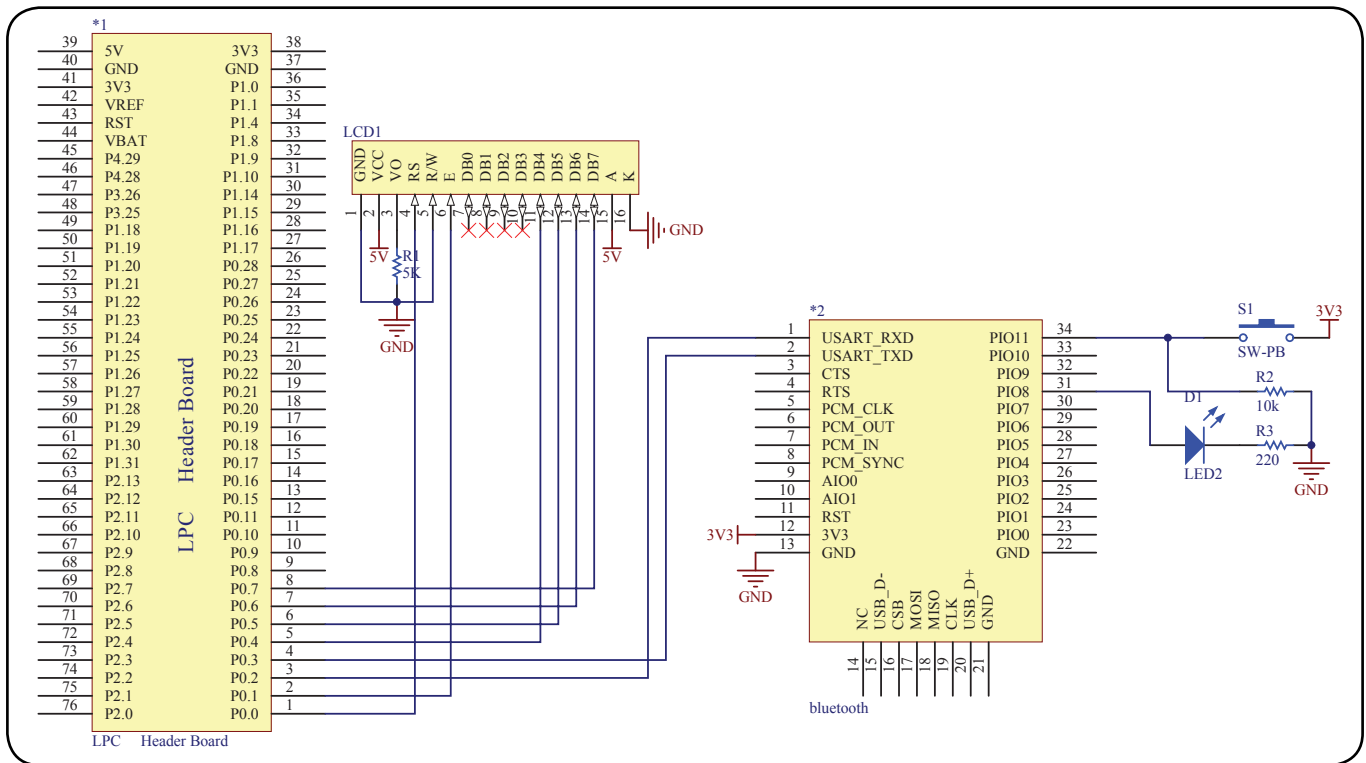
system_LPC17xx را Add کنید.

شرح کلی برنامه

ابتدا میکرو رشته ای به Hyper Terminal می فرستد و منتظر دستور کاربر می ماند. کاربر رشته را نوشته و با زدن

کلید Enter، رشته در ال سی دی نمایش داده می شود. سپس با فشردن دوباره کلید Enter توسط کاربر، برنامه به ابتدا

برگشته و این چرخه ادامه می یابد.



شکل شماره ۶

شرح قسمت های مختلف برنامه

```
#include <lpc17xx.h>
```

در دستور فوق فایل سرآیند میکروکنترلر را فراخوانی کرده ایم.

```
#define LCD_LPC1768
#define LCD_PORT_0
#define LCD_RS 0
#define LCD_E1
#define LCD_DB4 4
#define LCD_DB5 5
#define LCD_DB6 6
#define LCD_DB7 7
#include "lcd.h"
```

سپس پورتهای که ال سی دی به آن متصل است و پین های میکرو که به ال سی دی متصل هستند را به ترتیب در برنامه نوشته و کتابخانه مربوط به LCD را فراخوانی کرده ایم.

```
int j,m,i,n;
char code[]="Write The String: "
char str[20];
char get[16];
```

در این قسمت از برنامه چند متغیر از نوع int و سه رشته از نوع کاراکتر را تعریف کرده ایم که کاربردشان در ادامه خواهد آمد.

```
void putchar(char c)
{
while(!(LPC_UART0->LSR & 0x20));
LPC_UART0->THR=c;
}
```

تابع putchar

با فراخوانی این تابع، ابتدا میکروکنترلر منتظر می ماند تا رجیستر THR خالی شود، سپس کاراکتر C را در این رجیستر

قرار داده و ارسال می کند.

```
void puts(char *str)
{
    char k;
    while (k=*str++) putchar(k);
    putchar(10);
}
```

تابع puts

با فراخوانی این تابع، رشته ی کاراکتری که در آرگومان تابع نوشته می شود، از طریق میکرو ارسال می شود. در سطر آخر این تابع عدد ده را (که کد رفتن به سطر بعدی می باشد) ارسال می کنیم. یعنی ابتدا یک رشته، مثلا عبارت Write The String: ارسال می شود و این رشته در صفحه نرم افزار Hyper Terminal ظاهر می شود و مکان نما به سطر دوم می رود.

```
gets()
{
    i=0;
    j=1;
    while(i<16 && j==1){
        while(!(LPC_UART0->LSR & 0x1));
        get[i] = LPC_UART0->RBR;
        if (get[i] != 0x0d) i++;
        else j=0;
    }
}
```

تابع gets

آخرین تابع این برنامه نیز تابع gets() می باشد که با فراخوانی آن میکرو منتظر می ماند تا ثبات RBR برای دریافت کاراکتر جدید خالی شود. سپس کاراکتر را دریافت و در خانه i از رشته get[i] قرار می دهد. این عمل تا زمانی که کلید Enter، کد 0x0d، فشار داده نشده یا طول رشته وارد شده کمتر از ۱۷ کاراکتر باشد انجام می شود. در صورت فشردن کلید Enter یا زیاد شدن طول رشته از ۱۶ کاراکتر، رشته وارد شده را در get[i] ریخته و از تابع خارج می شود.

```
int main()
{
    SystemInit();
    lcd_init();
    lcd_clear();
    LPC_PINCON->PINSEL0=(1<<4)|(1<<6);
    {
```

تابع main

حال وارد تابع اصلی برنامه می شویم. ابتدا به کمک تابع SystemInit() تنظیمات مربوط به PLL انجام می شود. در این حالت $PCLK = 25\text{ M}$ و $CCLK = 100\text{ M}$ می باشد. ال سی دی را فعال و صفحه نمایش را پاک می کنیم و به دنبال آن رجیستر PINSEL0 را که مربوط به پین های TXD0 و RXD0 است را مقدار دهی می کنیم.

```
LPC_UART0->LCR=0x83;
LPC_UART0->DLM=0x00;
LPC_UART0->DLL=0x28;
LPC_UART0->LCR=0x03;
```

سپس رجیسترهای مربوط به UART0 را با توجه به مقادیر زیر مقدار دهی کرده ایم:

Baud Rate = 38400 bps , Data bits:8 , Parity:None , Stop bits: 1 , Flow control: None.


```
while (1)
{
    putchar(10);
    lcd_clear();
    puts(code);
    lcd_gotoxy(1,1);
    lcd_putsf("HC-05 Testing");
    gets();
    ++){for (m=1; m<=i; m
    lcd_gotoxy(2,m);
    lcd_putchar(get[m-1]);
}
```

در یک حلقه بی نهایت یک سطر به پایین رفته (ارسال عدد ۱۰)، سپس صفحه LCD را پاک می کنیم. در سطر بعدی رشته «Write The String:» را ارسال می کنیم تا در محیط نرم افزار Hyper Terminal نمایش داده شود. سپس در سطر اول و ستون اول از ال سی دی عبارت «HC-05 Testing» نمایش داده می شود. سپس تابع gets() فراخوانی می شود تا رشته تایپ شده در محیط نرم افزار به میکروکنترلر ارسال شود. میکرو این رشته از کاراکترها را گرفته و در سطر دوم و ستون اول نمایش می دهد.

```
while (1){
    while(!(LPC_UART0->LSR & 0x1));
    n = LPC_UART0->RBR;
    if (n == 0x0d) break;}
}
```

در یک حلقه دیگر منتظر می مانیم تا دوباره کلید Enter توسط کاربر فشار داده شود. پس از نوشتن برنامه، کامپایل کردن آن و پروگرام کردن میکروکنترلر، جهت استفاده از سخت افزار به صورت زیر عمل کنید:

تغذیه مدار را اعمال کنید. سپس پین PIN34 ماژول را به وسیله کلید، از زمین مدار قطع و به ولتاژ ۳٫۳ ولت وصل کنید تا ماژول وارد مد AT mode1 شود. در این حالت LED با فرکانس دو هرتز چشمک می زند. با بلوتوث کامپیوتر در قسمت Add Device جستجو کرده و پس از یافتن ماژول، آن را انتخاب کنید و Add را بزنید. در صورت درخواست رمز برای برقراری ارتباط، عدد 1234 را وارد کنید. این عدد رمز پیش فرض ماژول بوده و می توانید توسط دستورات AT cammands آن را تغییر دهید.

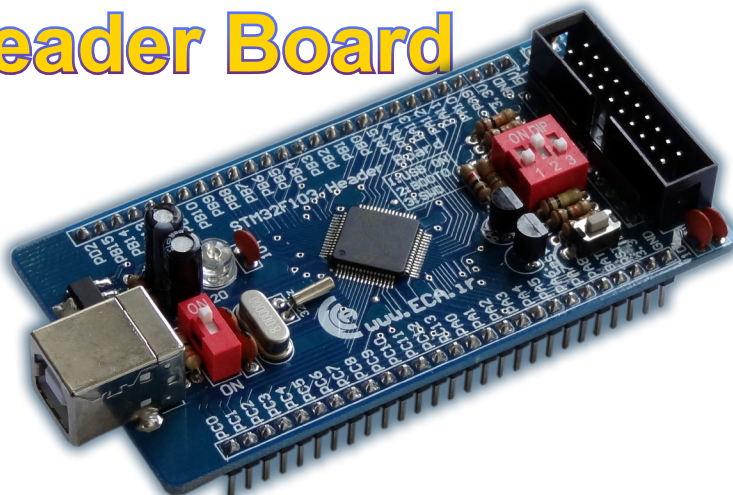
پس از Add کردن ماژول دو پورت com به سیستم شما اضافه خواهد شد. برای دیدن شماره پورت های اضافه شده، روی MyComputer کلیک راست کرده گزینه Manage و سپس Devis Manager را انتخاب کنید. در قسمت Ports(COM & LPT) پورت های اضافه شده مشاهده می شوند. یکی از پورت ها Outgoing و دیگری Incoming می باشد. شماره پورت مربوط به Outgoing را به خاطر بسپارید. حال نرم افزار Hyper Terminal را باز کرده تنظیمات مربوطه را انجام دهید. این تنظیمات و نحوه کار با نرم افزار در قسمت اتصال ماژول به کامپیوتر ذکر شد. فقط به هنگام انتخاب پورت، شماره پورت مربوط به Outgoing را وارد کنید و ادامه تنظیمات را طبق مراحل گفته شده انجام دهید.

هدف از این مقاله صرفاً آشنایی با نحوه برقراری ارتباط میکرو با دستگاه های دیگر از طریق بلوتوث و به کمک ماژول HC-05 Serial-Bluetooth بود. بی شک با کمی ابتکار می توان پروژه های مفید و کاربردی تری را در این زمینه تعریف، طراحی و اجرا نمود.

STM32F103RET6 Header Board

خانواده STM32 از سری میکروکنترلرهای ۳۲ بیتی مبتنی بر پردازنده ARM Cortex می باشد که با هدف ایجاد معنای جدیدی از آزادی عملکرد برای کاربران میکروکنترلرها ایجاد گشته است. در این خانواده از میکروکنترلرها تلفیقی از کارایی بالا، عملکرد همزمان، پردازش سیگنال های دیجیتال و کاربردهای کم توان در بالاترین سطح کیفی و با حفظ یکپارچگی کامل و سهولت توسعه، ارائه می گردد.

از جمله امکانات مهم میکروکنترلر STM32F103RET6، هسته Cortex M3 با سرعت 72MHz، اینترفیس موازی LCD، ۳ عدد ADC 12 بیتی، ۲ عدد DAC 12 بیتی و دارا بودن اینترفیس های متنوعی از جمله SDIO، USART، I2C، Can، USB و ... می باشد.



توضیحات تکمیلی : <http://eshop.eca.ir/link/1325.php>

ATXMEGA128A3U Header Board

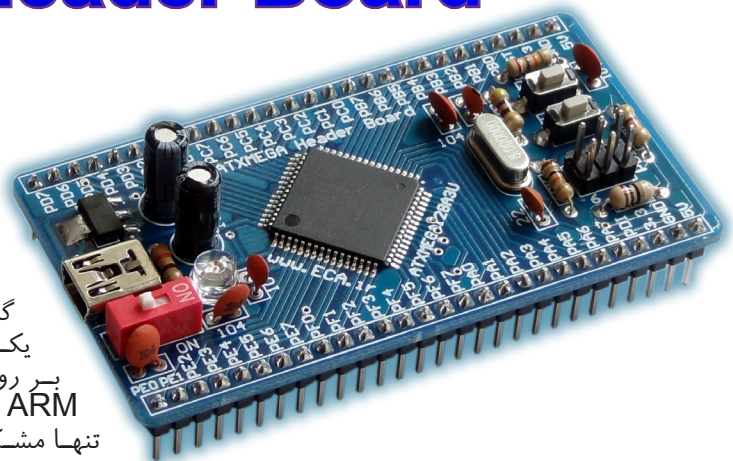
خانواده XMEGA نسل دوم از میکروکنترلرهای با مصرف توان پایین (Pico Power Technology) با قابلیت ها و کارایی بسیار بالاتر نسبت به سری قبل میکروهای AVR می باشند و قابلیت پردازش داده هم بصورت ۸ بیتی و هم به صورت ۱۶ بیتی را فراهم می نمایند همچنین از نظر معماری و ساختار تراشه این سری میکروکنترلرها مشابه خانواده قبل میکروهای AVR می باشند. در خانواده XMEGA تمامی مشکلات و کمبودهای نسل قبلی (MEGA) برطرف شده و قابلیت های جدید برای طراحان در نظر گرفته شده است.

یکی از نکات مهم در سری XMEGA امکانات و ماژول های موجود بر روی میکرو بوده که حتی این تعداد ماژول بر روی میکروهای نظیر ARM نیز موجود نیست.

تنها مشکلی که باعث شده تا این سری از میکروکنترلرها از استقبال کمتری نسبت به بقیه میکروکنترلرها روبرو شوند، هزینه بالای پروگرامر مخصوص این میکروکنترلرها بوده که با سری جدید این مشکل نیز مرتفع شده است.

در سری جدید، ماژول USB نیز به این خانواده اضافه شده است و شما می توانید بدون نیاز به هیچ پروگرامر خاصی و فقط از طریق پورت USB میکروکنترلر خود را پروگرام کنید. سرعت بالا و راحتی پروگرام نمودن از طریق این روش از مهمترین مزایای این سری از میکروکنترلرها می باشد.

توضیحات تکمیلی : <http://eshop.eca.ir/link/1324.php>



ماژول ضبط و پخش صدا VS1003

VS1003 یک تراشه همه کاره دکودر MP3 و WMA می باشد. این تراشه قابلیت ضبط صدا با کیفیت بالا و در باند وسیع را داراست، همچنین می تواند MIDI را هم از طریق فایل و هم با استفاده از MIDI keyboard سریال استاندارد پخش نماید.

امکانات ماژول :

قابلیت اتصال به تمام میکروکنترلرها از طریق رابط SPI
تامین ولتاژهای مورد نیاز تراشه و راه اندازی کل ماژول با تک

ولتاژ ۵ ولت

دارای خروجی هدفن به صورت استریو

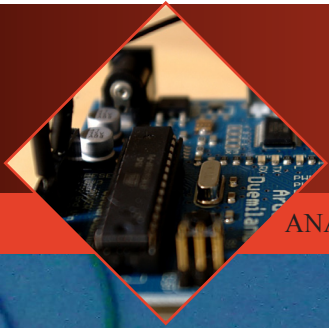
دارای ورودی میکروفن به صورت مونو

ایزوله نمودن مناسب در برابر نویز و تغییرات ولتاژ، به جهت بالا بودن

کیفیت ضبط و پخش



توضیحات تکمیلی : <http://eshop.eca.ir/link/1326.php>



پروژه دستگاه تشخیص نشت گازهای خانگی



**CARBON MONOXIDE
BE ALARMED!**

نویسنده: سوران آراسته soran.arasteh@gmail.com

در جامعه امروزی هر روز مشاهده می شود که بر اثر بی احتیاطی، افراد زیادی به دلیل نشت گازهای خانگی (LPG) جان خود را از دست می دهند.

با ورود سنسورهای تشخیص گاز و آگاهی از نحوه عملکرد آن ها، افراد و شرکت های زیادی دست به ساخت دستگاه هایی زدند که قادر به تشخیص گازهای پراکنده در محیط شود. تفاوت عمده این دستگاه ها بیشتر در سرعت عملکرد و یا امکانات آن می باشد. پروژه ای که در حال حاضر مشاهده می فرمایید دستگاهی است که قادر به تشخیص گاز بوده و با توجه به مقدار گاز پخش شده عملکردهای مختلفی را به صورت هوشمند انجام می دهد.

شرح عملکرد کلی دستگاه

این دستگاه شامل دو قسمت، که یکی فرستنده و دیگری گیرنده (برد اصلی) می باشد تشکیل شده است. قسمت فرستنده در بالای سقف منزل نصب شده و قسمت مادربرد در مکان مورد نظر کاربر و با توجه با اینکه در قسمت مذکور دسترسی لازم به برق و خط تلفن موجود باشد نصب می گردد.

دلیل نصب فرستنده در سقف این است که گاز نشتی به دلیل کمبود وزن ابتدا به بالای سقف می رود و در صورت وجود سنسور مربوطه در بالاترین قسمت منزل می توان سریع ترین نتیجه را حاصل کرد.

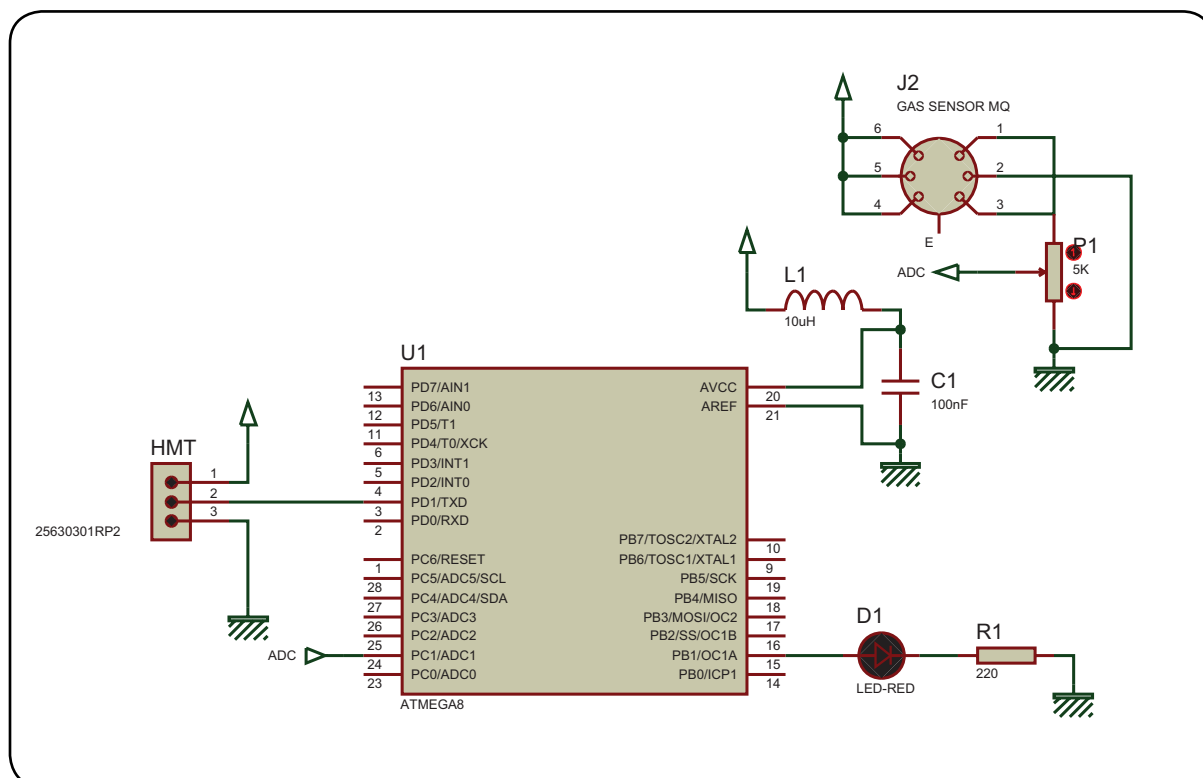
حال به تشریح عملکرد اصلی دستگاه می پردازیم، در این دستگاه با روشن کردن هر دو قسمت فرستنده و گیرنده بعد از چند لحظه با هم ارتباط برقرار کرده و فرستنده دیتای دریافتی از سنسور را با استفاده از ADC میکرو به اعداد دیجیتال تبدیل کرده و از طریق پروتکل UART به گیرنده ارسال می کند.

در قسمت گیرنده با دریافت دیتا، عملکرد لازم با توجه به مقدار گاز موجود در هوای منزل تشخیص داده شده و انجام

می گیرد.

در وضعیتی که هیچ گازی در محیط احساس نشود بر روی صفحه نمایش پیغام عادی بودن هوا نمایش داده می شود و همه ی خروجی ها غیر فعال می باشند. همچنین دو LED بر روی مادربرد وجود دارد که LED سبز رنگ وضعیت عادی و LED قرمز وضعیت غیر عادی را نمایش می دهد.

به محض تشخیص گاز توسط قسمت گیرنده و ارسال دیتای مبتنی بر وجود گاز در محیط، ابتدا آلارم مدار به صدا در خواهد آمد و پیغام وجود مقدار جزئی گاز بر روی نمایشگر نمایان می شود. اگر مقدار گاز کم شود آلارم قطع شده و مدار به وضعیت عادی برمی گردد اما اگر مقدار نشت گاز همچنان افزایش یابد بعد از به صدا در آمدن زنگ دستگاه، شیر گاز را قطع خواهد کرد (این عمل با استفاده از شیربرقی انجام می گیرد). حال بعد از قطع گاز اگر در منزل آتش سوزی پیش آمده باشد چون همچنان مقدار دود حاصل از آتش سوزی رو به افزایش خواهد بود، دستگاه به ۸ شماره دلخواه زنگ خواهد زد. یک خروجی دیگر نیز برای مدار در نظر گرفته شده که به نیاز کاربر مورد استفاده قرار می گیرد. در همه ی حالت های فوق کاربر می تواند با فشار کلیدی که بر روی برد تعبیه شده خروجی ها را مجدداً غیر فعال کند.



شکل شماره ۱: شماتیک بخش فرستنده

تشریح عملکرد بخش فرستنده

در برنامه فرستنده در ابتدا میکرو مورد استفاده معرفی شده سپس فرکانس کاری و نرخ انتقال دیتا یا باودریت میکروکنترلر معرفی شده است.

در خطوط بعد پورت b.0 به عنوان خروجی جهت راه اندازی LED نشانگر روشن بودن دستگاه پیکربندی شده و همچنین ADC تراشه جهت اسکن گاز موجود در محیط فعال سازی شده است. در این جا توضیحات مختصری راجع به ADC و نحوه راه اندازی آن داده خواهد شد. ADC چیست:

گاهی نیاز است که یک کمیت بیرونی مانند دما، شدت نور، .. اندازه گیری شود. جهت این کار از سنسورهای مربوطه استفاده می شود. سنسورها کمیت موردنظر را به ولتاژ، جریان و یا سایر کمیت های الکتریکی که اندازه گیریشان کار دشواری نیست تبدیل می کنند. ولتاژ تهیه شده توسط سنسور را می توان به یک کانال ADC اعمال کرده و مقدار دیجیتال آن را تولید کنیم و این مقدار دیجیتال را با محاسبات ریاضی به عددی دسیمال جهت نمایش تبدیل کنیم. ADC های موجود در میکروکنترلرهای AVR ۱۰ بیتی بوده و بنابراین می توانند اعداد بین ۰ تا ۱۰۲۳ را در خود ذخیره کنند. مثلاً برای صفر ولت عدد صفر، برای پنج ولت عدد ۱۰۲۳ و برای ۲.۵ ولت عدد ۵۱۱ را در خود ذخیره می کنند. برای استفاده از این قابلیت باید ابتدا مثل سایر امکانات میکروکنترلر، پیکربندی آن را انجام داد. نحوه پیکربندی ADC:

شکل کلی راه اندازی به صورت زیر می باشد:

CONFIG ADC = single, PRESCALER = AUTO, REFERENCE = opt

همانطور که در جدول شماره ۱ ملاحظه می شود دو مد کاری برای ADC وجود دارد:

ADC	Running mode. May be SINGLE or FREE.
PRESCALER	A numeric constant for the clock divider. Use AUTO to let the compiler generate the best value depending on the XTAL
REFERENCE	The options depend on the used micro. Some chips like the M163 have additional reference options. In the definition files you will find : ADC_REFMODEL = x This specifies which reference options are available. The possible values are listed in the table below.

جدول شماره ۱

۱) Single: در این حالت هر موقعی که در برنامه کانال ADC مورد نظر فراخوانی شود، مقدار آن توسط میکرو گزارش می شود.

۲) Free: در این حالت خود میکرو به صورت پیش فرض در زمانهای معین کانال را چک کرده و به برنامه گزارش می دهد.

قسمت PRESCALER مربوط به انتخاب فرکانس نمونه برداری کانال می باشد که در صورت انتخاب auto خود میکرو بهترین کلاک را انتخاب خواهد کرد.

در بخش REFERENCE که مربوط به انتخاب ولتاژ مرجع برای ADC می باشد چندین حالت به صورت زیر وجود دارد:

۱) Off: ولتاژ مرجع داخلی که برابر ۲,۵۶ ولت است خاموش شده و ولتاژ وصل شده به پایه Aref به عنوان ولتاژ مرجع در نظر گرفته می شود.

۲) AVcc: ولتاژ پایه AVcc به عنوان ولتاژ مرجع در نظر گرفته می شود.

۳) Internal: از ولتاژ مرجع داخلی ۲,۵۶ ولت استفاده می شود.

در برنامه دو متغیر به نام های A و AD معرفی شده اند که قبل از کاربرد هر کدام توضیحی در مورد متغیر داده می شود:

متغیر چیست؟ متغیر نامی برای کلمات حافظه است که داده ها در آن قرار می گیرند و ممکن است در طول اجرای برنامه تغییر کنند. برای دسترسی به متغیرها از نامشان استفاده می شود.

متغیر AD مقدار اسکن شده کانال ADC را در خود ذخیره می کند که این عدد می تواند با توجه با ۱۰ بیتی بودن ADC ۰ تا ۱۰۲۳ باشد.

متغیر A مقدار رشته ای متغیر AD را در خود جهت ارسال به گیرنده ذخیره می کند.

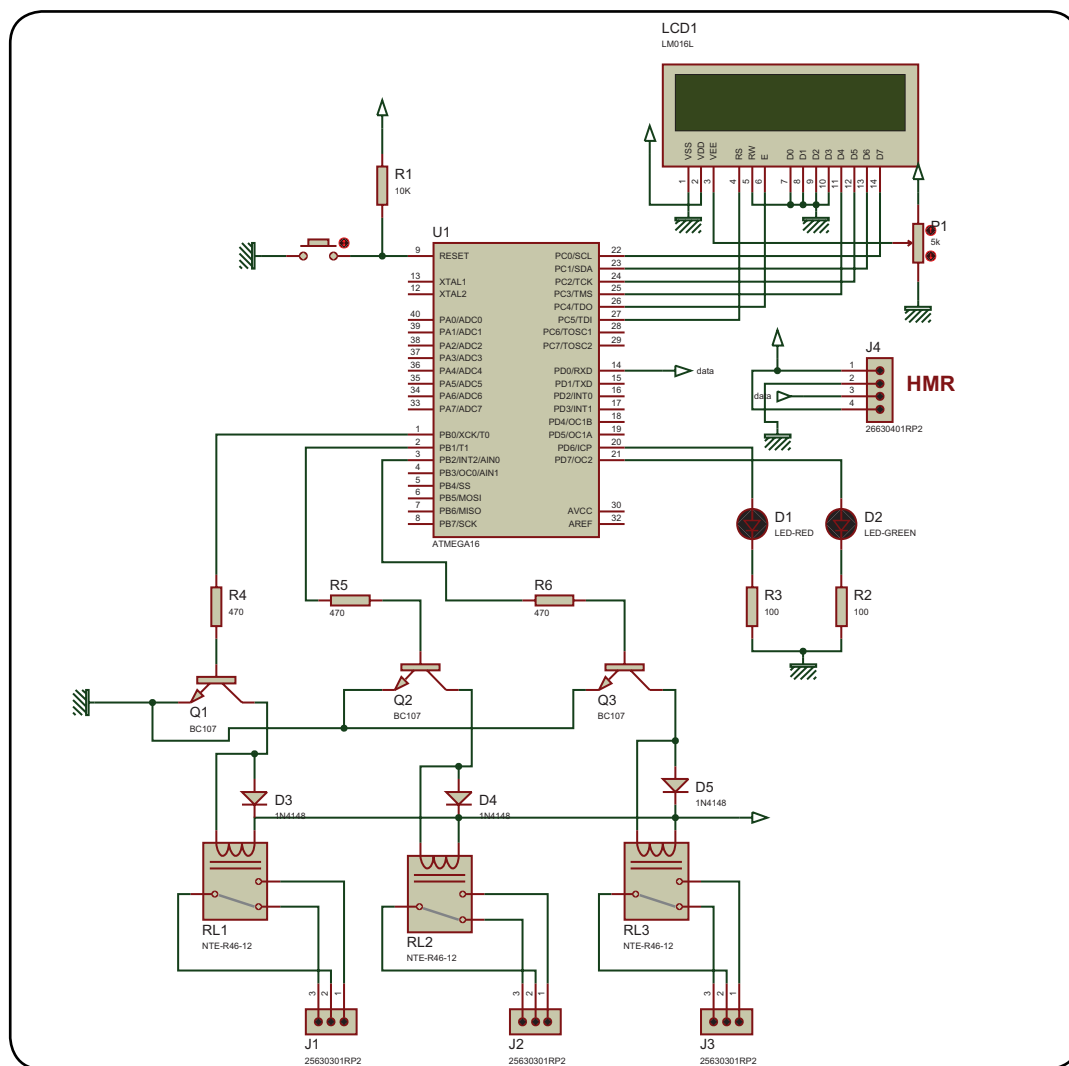
در مدار فرستنده در یک حلقه بی پایان مدام کانال مربوطه ADC چک شده و سپس مقدار عددی آن به رشته تبدیل می شود بعد از طی این مراحل با استفاده از دستور print مقدار رشته ای متغیر A از طریق کانال uart به گیرنده ارسال می شود.

برنامه نوشته شده برای فرستنده به زبان بیسیک:

```
$regfile = "m8def.dat"
$crystal = 1000000
$baud = 9600
'-----
Config Portb.1 = Output
ConfigAdc = Single , Prescaler = Auto , Reference = Avcc
Start Adc
Enable Adc
'-----
Dim A As String * 13
Dim Ad As Word
'-----
Set Portb.1
Wait 2
Main:
Do
Ad = Getadc(1)
A = Str(ad)
Waitms 75
Print A ; Chr(13);
Loop
End
```

تشریح عملکرد بخش گیرنده

دستگاه با دریافت مقدار گاز موجود، آن را طبق ضوابط از پیش تعیین شده چک کرده و به صورت خودکار عمل لازم را انجام می دهد.



شکل شماره ۲: شماتیک بخش گیرنده

برنامه نوشته شده برای گیرنده به زبان بیسیک:

```
$regfile = "m16def.dat"
$crystal = 1000000
$baud = 9600
'-----
ConfigLcd = 16 * 2
ConfigLcdpin = Pin ,Rs = Portc.5 , E = Portc.4 , Db4 = Portc.3 , Db5 = Portc.2 , Db6 = Portc.1 ,
Db7 = Portc.0
Cursor Off
'-----
Config Portb.0 = Output
Config Portb.1 = Output
Config Portb.2 = Output
Config Portb.3 = Output
Config Portd.6 = Output
Config Portd.7 = Output
Config Portb.4 = Input
'-----
```



```

Dim A As String * 13
Dim Ad As Word
Dim Tab As Byte
'-----
Designer:
Cls
Locate 1 , 15 : Lcd "www.ECA.ir"
Locate 2 , 15 : Lcd "www.ECA.ir"
For Tab = 0 To 13
ShiftLcd Left
Waitms 500
Next Tab
Wait 1
Main:
Do
Input A
Ad = Val(a)
If Ad <= 40 Then
Cls
LcdAd: Locate 1 , 4: Lcd "Normal Air"
Locate 2 , 3: Lcd "Disable Out"
Reset Portb.3: Reset Portb.2: Reset Portb.0: Reset Portb.1
Set Portd.7
Reset Portd.6
End If
If Ad > 40 Then
If Ad <= 120 Then
Cls
Locate 1 , 2: Lcd "low Gas sensing"
Locate 2 , 3: Lcd "Enable Alarm"
Set Portb.3: Reset Portb.2: Reset Portb.0: Reset Portb.1
Reset Portd.7
Set Portd.6
If Pinb.4 = 0 Then Reset Portb.3
End If
End If
If Ad > 120 Then
If Ad <= 200 Then
Cls
Locate 1 , 2: Lcd "Increased gas"
Locate 2 , 2: Lcd "Off gas valve"
Set Portb.2: Set Portb.3: Reset Portb.0: Reset Portb.1
Reset Portd.7
Set Portd.6
If Pinb.4 = 0 Then
Reset Portb.3
Reset Portb.2
End If
End If
End If
If Ad > 200 Then
If Ad <= 500 Then
Cls
Locate 1 , 2: Lcd "Increased gas"
Locate 2 , 1: Lcd "Contact Support"
Set Portb.2: Set Portb.3: Set Portb.0: Reset Portb.1

```

```

Reset Portd.7
Set Portd.6
If Pinb.4 = 0 Then
Reset Portb.3
Reset Portb.2
Reset Portb.0
End If
End If
End If
If Ad > 500 Then
If Ad <= 1023 Then
Cls
Locate 1 , 2: Lcd "Increased gas"
Locate 2 , 6: Lcd "On Air..."
Set Portb.2: Set Portb.3: Set Portb.0: Set Portb.1
Reset Portd.7
Set Portd.6
If Pinb.4 = 0 Then
Reset Portb.3
Reset Portb.2
Reset Portb.0
Reset Portb.1
End If
End If
End If
If Pinb.4 = 0 Then
Reset Portb.0
Reset Portb.1
Reset Portb.2
Reset Portb.3
End If
Loop
End
    
```

در برنامه گیرنده نیز چند خط اول مربوط به معرفی میکروکنترلر به کامپایلر، تعیین فرکانس کاری، باوودریت و ورودی/خروجی کردن پین ها می باشد.

در خط بعدی متغیرها معرفی شده اند که کاربرد آنها به ترتیب زیر است:

(۱) Tab: در تابع Designer جهت حرکت متن در حلقه for next استفاده شده است.

(۲) A: این متغیر همان متغیر در قسمت فرستنده می باشد که توسط گیرنده دریافت می شود و همچنان به صورت رشته است.

(۳) Ad: این متغیر مقدار عددی متغیر A را در خود ذخیره می کند تا بتوان از این مقدار در قسمت های مختلف برنامه استفاده کرد.

نحوه کارکرد توابع مورد استفاده در برنامه گیرنده

تابع Designer:

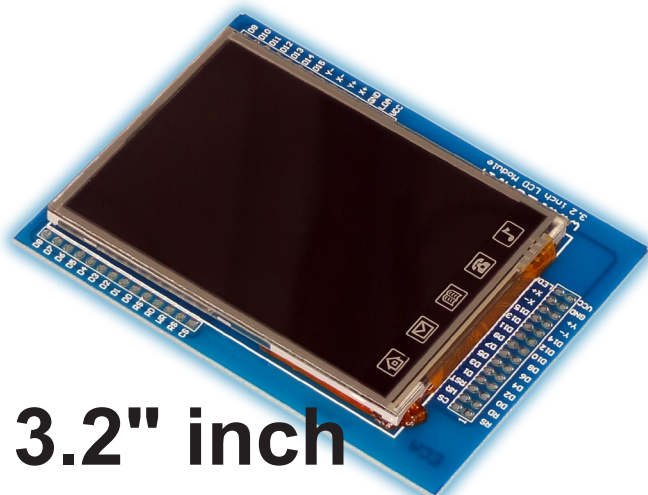
این تابع جهت نمایش یک تبلیغ ساده و نمایش یک متن با افکت، بر روی LCD می باشد که شیفت دادن متن توسط یک حلقه for next مهیا شده است.

در دستورات شیفت فقط امکان شیفت دادن یک واحدی وجود دارد که در صورت نیاز به شیفت دادن با تعداد زیاد باید از حلقه ها استفاده کرد.

تابع Main:

در این تابع که قسمت اصلی برنامه نیز می باشد ابتدا مقدار A از پروتکل UART دریافت شده و سپس مقدار رشته ای متغیر A به صورت عددی در متغیر AD قرار می گیرد. مقدار این متغیر بیان کننده مقدار گاز موجود در محیط می باشد که با استفاده از شرط های if موجود در برنامه این مقدار تست شده و با توجه به گاز موجود در محیط در مورد آن تصمیم گیری می شود.

ماژول های TFTLCD به همراه تاج اسکرین



3.2" inch

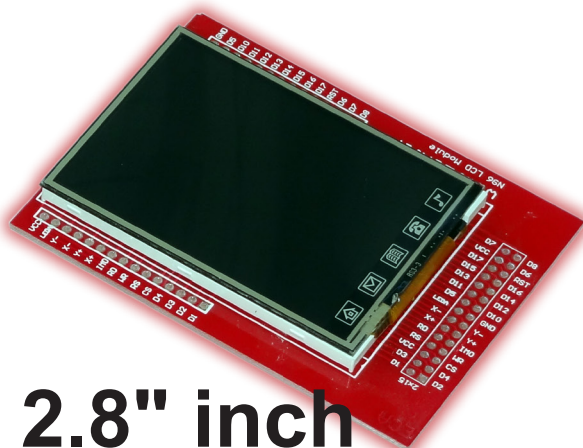
این ماژول جهت کارکرد راحتتر با LCD های رنگی ۳،۲ اینچی ساخته شده و تمام پایه های مورد استفاده این LCD و همچنین پایه های تاج اسکرین بیرون کشیده شده است. ماژول فوق قابلیت اتصال بر روی برد برد و تمام بردهای کاربردی و صنعتی را دارا می باشد. در ضمن ماژول دارای بسته بندی مناسبی جهت حفاظت از شکستگی در مرسوله های پستی می باشد.

اطلاعات مربوط به LCD :

LCD رنگی فوق دارای چیپست به شماره ی SSD1289 بوده که می تواند از طریق ۱۶ خط داده با انواع میکروکنترلر های ۸ و ۳۲ بیتی از قبیل AVR-XMEGA-PIC-ARM ارتباط برقرار کرده و اطلاعات مربوط به تصاویر را از آنها دریافت کند. برای LCD مورد بحث امکان راه اندازی در هر ۲ مود ۸ و ۱۶ بیتی وجود دارد این ماژول برای عملکرد ۱۶ بیتی طراحی شده و تغییر مود عملکرد بر عهده کاربر خواهد بود.

رزولیشن تصویر : ۳۲۰*۲۴۰ پیکسل
ابعاد : ۵،۵*۷،۵ سانتی متری (۳،۲ اینچ)
چیپست کنترلی - درایور: SSD1289

توضیحات تکمیلی : <http://eshop.eca.ir/link/897.php>



2.8" inch

این ماژول جهت کارکرد راحتتر با LCD های رنگی ۲،۸ اینچی ساخته شده و تمام پایه های مورد استفاده این LCD و همچنین پایه های تاج اسکرین بیرون کشیده شده است. ماژول فوق قابلیت اتصال بر روی برد برد و تمام بردهای کاربردی و صنعتی را دارا می باشد. در ضمن ماژول دارای بسته بندی مناسبی جهت حفاظت از شکستگی در مرسوله های پستی می باشد.

اطلاعات مربوط به LCD :

LCD رنگی فوق دارای چیپست به شماره ی ili9325-ili9320 بوده که می تواند از طریق ۱۶ خط داده با انواع میکروکنترلر های ۸ و ۳۲ بیتی از قبیل AVR-XMEGA-PIC-ARM ارتباط برقرار کرده و اطلاعات مربوط به تصاویر را از آنها دریافت کند. برای LCD مورد بحث امکان راه اندازی در هر ۲ مود ۸ و ۱۶ بیتی وجود دارد که برای اینکار می بایستی مقاومت های روی LCD را تغییر دهید و یا از پایه IM۰ استفاده نمایید که برای این کار نیاز به لحیم کاری LCD می باشد. این ماژول برای عملکرد ۱۶ بیتی طراحی شده و تغییر مود عملکرد بر عهده کاربر خواهد بود. این LCD در بازار موبایل، به نام LCD Nokia N96 چینی شناخته می شود که دارای چیپست های بسیار متنوعی هستند که تنها مدل های خاصی از آنها قابل راه اندازی هستند.

رزولیشن تصویر : ۳۲۰*۲۴۰ پیکسل

ابعاد : ۶،۵*۵ سانتی متری (۲،۸ اینچ)

چیپست کنترلی - درایور : ili9325- ili9320

توضیحات تکمیلی : <http://eshop.eca.ir/link/619.php>

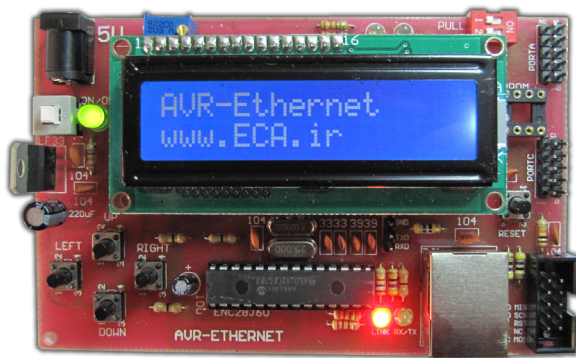
نرم افزار Jimbo LCD :

جهت راه اندازی هرچه ساده تر LCD های رنگی توسط میکروکنترلرهای مختلف (ARM-AVR) نرم افزار Jimbo LCD از طرف وب سایت ECA طراحی گشته است تا کاربران به راحتی بتوانند عکس، افکت و سایر موارد مرتبط را بر روی LCD های رنگی پیاده سازند.



توضیحات تکمیلی : http://eshop.eca.ir/link/jimbo_lcd.php

AVR Ethernet Board

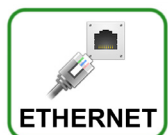


در این دستگاه از تراشه ENC28J60 ساخت شرکت microchip استفاده شده است. این تراشه تمامی شرایط مورد نیاز برای اتصال به شبکه TCP/IP (لایه فیزیکی، mac و...) را یکجا فراهم می کند. نحوه برقراری ارتباط این تراشه با میکرو کنترلر به وسیله پروتکل پر سرعت SPI می باشد. در این دستگاه علاوه بر خود تراشه ENC28J60 یک میکرو کنترلر ATMEGA32A به همراه یک LCD شستی و... قرار داده شده است. همچنین جهت سهولت در امر برنامه نویسی و یادگیری نحوه استفاده از این برد، فایل آموزشی به همراه کتابخانه های لازم جهت برنامه نویسی و پروژه نمونه در CD محصول قرار گرفته است.



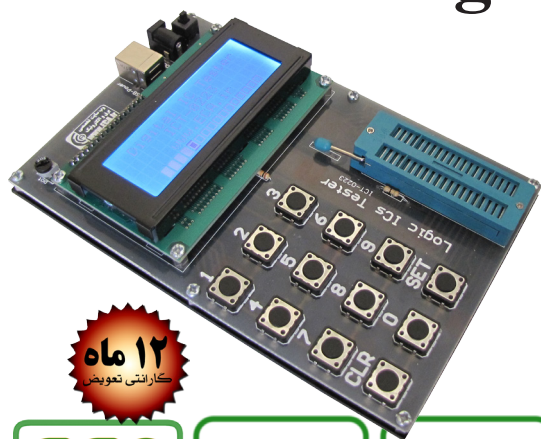
امکانات و مشخصات برد:

- * میکرو کنترلر ATMEGA32A
- * کنترلر شبکه ENC28J60 10Mbit
- * 4 عدد شستی مکان نما
- * دارای LCD کاراکتری 16*2
- * امکان اتصال حافظه EEprom
- * کانکتور ISP برای پروگرام کردن برد
- * کانکتور خروجی برای دو پورت A و C میکرو کنترلر
- * نمایش ارسال و دریافت اطلاعات شبکه از طریق LED



توضیحات تکمیلی : <http://eshop.eca.ir/link/829.php>

Digital ICs Tester



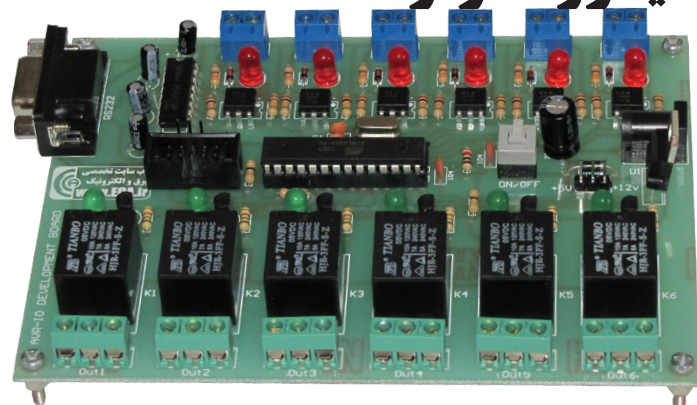
یکی از مشکلات موجود بر سر راه علاقمندان و طراحان الکترونیک که همیشه باعث اتلاف وقت بسیار زیادی می شود، خرابی و معیوب شدن قطعات الکترونیکی بوده که هیچ گاه از ظاهر قابل تشخیص نبوده و فقط با تست قطعه معیوب می توان از خرابی آن آگاه شد. این دستگاه یک تست کننده انواع IC مخصوص آزمایشگاه ها و جزئی لاینفک از تجهیزات طراحان حرفه ای می باشد که دارای امکاناتی جهت تست انواع IC های پر کاربرد خانواده های TTL و CMOS و ULN و... می باشد و می تواند در کمترین زمان ممکن قطعات را تست و سالم بودن یا خراب بودن تک تک گیت ها و یا کلیت قطعه را مشخص نماید.

قطعات قابل تست دستگاه :

- « IC های پر کاربرد سری 74XX
- « IC های پر کاربرد سری 40XX
- « IC های پر کاربرد سری 45XX
- « IC های پر کاربرد سری ULN 20XX
- « IC های پر کاربرد سری ULN 28XX

توضیحات تکمیلی : <http://eshop.eca.ir/link/633.php>

برد توسعه ورودی خروجی میکرو کنترلر AVR



برد توسعه AVR جهت استفاده از پورت های ورودی و خروجی میکرو کنترلرهای AVR طراحی شده است.

کاربردها :

مصارف آموزشی، جهت آشنایی و تسلط بر امکانات میکرو کنترلرهای AVR
استفاده در پروژه های هوشمندسازی خانه و سایر مصارف کنترلی در محیط های صنعتی به عنوان دستگاه های کنترلی مانند PLC
استفاده در پروژه های اینترفیس آموزشی و کاربردی توسط پورت سریال
همانک با نرم افزارهای برق و الکترونیک مانند MATLAB و LABVIEW و...



توضیحات تکمیلی : <http://eshop.eca.ir/link/621.php>

ساخت ماشین حساب لمسی بر روی 3.2 LCD اینچی



نویسنده: یوسف خداپرست
bemu14@yahoo.com

می خواهیم با استفاده از
برد آموزشی NXP LPC1768

ورژن ۳ (موجود در فروشگاه ECA) و با

استفاده از نمایشگر گرافیکی ۳،۲ اینچی و صفحه

لمسی آن به عنوان کلید یک ماشین حساب لمسی
طراحی کنیم.

هدف از این پروژه آشنایی با نحوه استفاده از LCD گرافیکی موجود، و
همچنین آشنایی با نحوه عملکرد صفحات لمسی می باشد.

LCD گرافیکی استفاده شده در این پروژه شامل یک جهت راه اندازی می باشد. برای

استفاده از این کتابخانه باید فایل های Font_24x16.h، lcd.c، lcd.h و sys.h در مسیر پروژه و یا در مسیر کامپایلر keil
کپی شود.

همچنین در این پروژه تاج اسکرین LCD به عنوان keypad برای ماشین حساب به کار گرفته شده است. LCD استفاده
شده دارای یک کتابخانه با نام touch برای کار با صفحه لمسی می باشد که برای استفاده از این کتابخانه باید فایل های
touch.h، touch.c، i2c.h، i2c.c، sys.h، exti.h در مسیر پروژه موجود باشد.

این فایل ها اگر در مسیر پروژه کپی شود کتابخانه به طور مثال به صورت #include "lcd.h" معرفی می شود و اگر این

فایل ها در مسیر کامپایلر keil یعنی مسیر Keil\ARM\INC\NXP\LPC17xx کپی شود کتابخانه به صورت #include <lcd.h> معرفی می شود.

فایل های مور نیاز در ضمیمه مجله ارائه شده است.

به توضیح توابعی از کتابخانه ها می پردازیم که در این پروژه مورد استفاده قرار گرفته اند:

Read_ADS(u16 *x,u16 *y):

این تابع برای خواندن مقدار آنالوگ قسمت های مختلف صفحه لمسی به کارگرفته می شود. هر قسمت از LCD دارای دو مولفه x , y می باشد که این تابع این مولفه ها را در اشاره گرهای x , y ذخیره می کند. البته برای تبدیل این مقادیر به موقعیت از تناسب استفاده می شود که در ادامه به آن اشاره می شود.

LCD_ShowString (u8 x,u16 y,char *p)

رشته p را در مختصات x , y به نمایش در می آورد.

LCD_ClearLn (unsigned int ln,unsigned int col)

به اندازه ۲۰ کاراکتر را از مختصات ln و col به بعد را پاک می کند.

LCD_ShowChar(u8 x,u16 y,u8 num,u8 size,u8 mode)

یک کاراکتر را در مختصات x , y به نمایش در می آورد

LCD_Bitmap (unsigned int x, unsigned int y, unsigned int w, unsigned int h, unsigned char *bitmap)

با استفاده از این تابع تصویر ذخیره شده در حافظه میکروکنترلر در LCD به نمایش در آورده می شود.

شرح کلی برنامه

عملکرد برنامه به این صورت است که تصویر ماشین حساب در LCD نمایش داده می شود و کاربر می تواند از طریق لمس کلیدهای ماشین حساب از طریق LCD محاسبات موردنظر خود را انجام دهد و نکته قابل توجه این است که لمس قسمت صفحه نمایش ماشین حساب به عنوان کلید ON/C عمل می کند، یعنی با لمس قسمت مورد نظر صفحه نمایش پاک می شود.

شرح قسمت های مختلف برنامه

```
#include <LPC17xx.h>
#include <string.h>
#include <stdlib.h>
#include <stdio.h>
#include "sys.h"
#include "i2c.h"
#include "exti.h"
#include "lcd.h"
#include "touch.h"
```

فایل های سرآیند و الحاقی برنامه در این قسمت از برنامه معرفی شده است و در قسمت های مختلف برنامه، توابع این کتابخانه ها به کار برده شده است.

```
extern unsigned char glImage_calcu[] ;
unsigned char matrix[4][4]={{'7','8','9','+'}
,{'4','5','6','-' }
,{'1','2','3','*'}
,{'0','.', '=', '/'}}
};
float number1 =0;
float number2 =0;
float result=0.0;
static char last_key ;
unsigned char mmz=0,count=0, data_in=0;
char save_str[32] ;
char lcd_buf[32] ;
char helper[32] ;
char str[32] ;
unsigned char i, j, Code, key ;
unsigned int x,y ;
```

در اینجا متغیرهای سراسری و آرایه های مورد استفاده در چند تابع از برنامه معرفی شده اند.

آرایه `gImage_calcu[]`

این آرایه شامل کدهای تصویر ماشین حساب که در LCD به نمایش در می آید می باشد و چون این کدها در یک فایل دیگر (`calcu.c`) معرفی شده اند برای استفاده از این کدها در فایل جاری برنامه این آرایه به صورت `extern` معرفی شده است.

آرایه `matrix [4][4]`

این آرایه یک آرایه دوبعدی ۴ در ۴ می باشد که در آن کدهای کلیدهای مختلف ماشین حساب معرفی شده است و این آرایه برای تشخیص کد کلید لمس شده به کار برده می شود.

```
void delay(){
int z ;
for(z=0 ; z<2500000 ; z++) ;
}
```

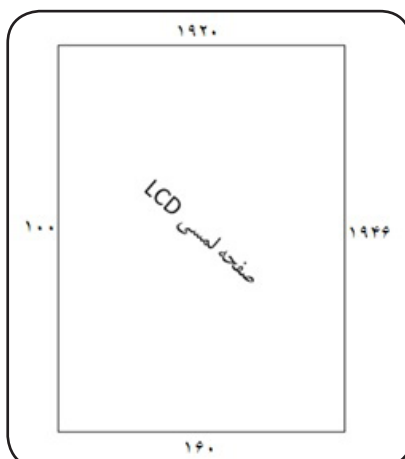
تابع `delay`

تابع تأخیر مورد استفاده در خواندن صفحه لمسی می باشد و برای جلوگیری از خطای لرزش دست در حین لمس صفحه کلید ماشین حساب می باشد.

```
char Key_Code(void)
{
while(1)
{
if(Pen_Point.Key_Sta==Key_Down)
{
Read_ADS(&Pen_Point.X,&Pen_Point.Y);
x=(240*(Pen_Point.X-100))/1846 ;
y=(320*(1920-Pen_Point.Y))/1760 ;
delay();
Pen_Point.Key_Sta=Key_Up ;
if(x<225&&x>5&&y<112&&y>33) {Code='c';
return Code;
}
for(i=0;i<4;i++)
{
for(j=0;j<4;j++)
{
if(x<52+j*59&&x>1+j*59&&y<163+i*50&&y>121+i*50)
{
Code = matrix [i] [j] ;
return Code ;
}}}}}}
}
```

تابع `Key_Code`

این تابع صفحه لمسی LCD را اسکن کرده و کد متناسب را برمی گرداند. برای خواندن صفحه لمسی به طور غیر مستقیم از تکنیک وقفه استفاده شده است به این صورت که پایه P2.13 در کتابخانه `touch` به صورت ورودی وقفه تعریف شده است و با لمس صفحه لمسی ولتاژ پایه P2.13 از مقدار ۳٫۳۷ به مقدار ۰ نزول پیدا می کند و چون وقفه به صورت لبه پایین رونده تنظیم شده است بنابراین وقفه رخ داده و در تابع `Key_Down` به این وقفه وضعیت متغیر `Pen_Point.Key_Sta` برابر با `Key_Down` می شود و با توجه به این مطلب در تابع `Key_Code` توسط دستور `if(Pen_Point.Key_Sta==Key_Down)` تست می شود که آیا صفحه لمسی، لمس شده است یا نه. سپس توسط تابع `Read_ADS` مقدار آنالوک تحویلی نقطه لمس شده صفحه لمسی خوانده شده و در متغیرهای `Pen_Point.X` و `Pen_Point.Y` ذخیره شده و سپس به موقعیت تبدیل می شود.



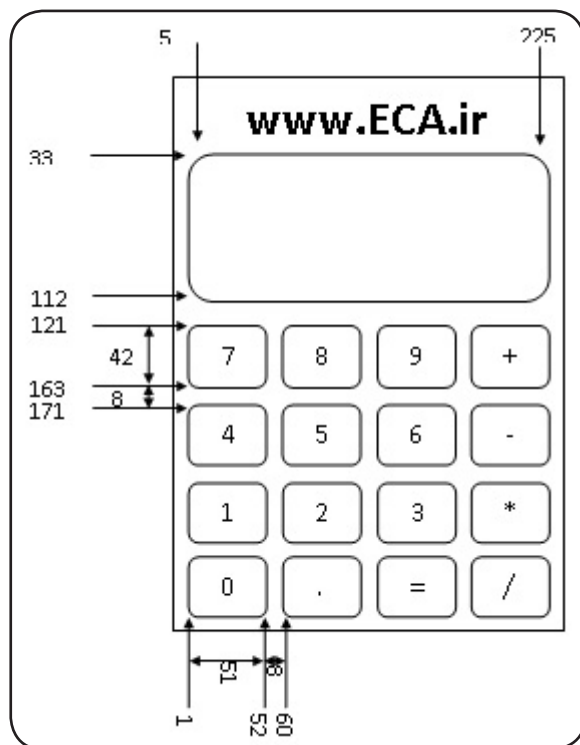
شکل شماره ۱

در شکل شماره ۱ اعداد ثبت شده، مقدار آنالوگ تحویلی صفحه لمسی در چهار ضلع آن می باشد که توسط تابع Read_ADS در یک فایل دیگر خوانده شده اند. با برقراری دو تناسب زیر موقعیت نقطه لمس شده LCD بدست می آید. در

$$\begin{array}{|c|c|} \hline 1920-160 & 320 \\ \hline 1920- & \\ \text{Pen_Point.X} & y \\ \hline \end{array} \Rightarrow x = \frac{240 \times (\text{Pen_Point.X} - 100)}{1846}$$

$$\Rightarrow y = \frac{320 \times (1920 - \text{Pen_Point.Y})}{1760}$$

نهایت با مقایسه موقعیت نقطه لمس شده و موقعیت کلید های ماشین حساب کد متناسب با آن برگردانده می شود. شده لمس موقعیت: ۷ کلید >> x<52 && x>1 && y<163 && y>121



شکل شماره ۲

اعداد ثبت شده در شکل موقعیت یک کلید از ماشین حساب را نشان می دهد و چون شکل دارای نظم هندسی است و کلیدها و فاصله بین آنها دقیقاً مثل هم است، موقعیت سایر کلیدها نیز مشخص می شود که از این قاعده در نوشتن تابع برای تشخیص موقعیت لمس شده و برگرداندن کد متناسب با آن استفاده شده است.

```
void display(){
    sprintf(lcd_buf,"%f",number1);
    LCD_ShowString(10,90,"");
    LCD_ShowString(15,90,lcd_buf);
    LCD_ClearLn(10,75);
    LCD_ShowChar(15,75,'=',16,0);
}
```

تابع display

این تابع برای نمایش نتیجه نهایی محاسبات در LCD ایجاد شده است.

```
void Clear(){
    *save_str=NULL;
    *str=NULL;
    *lcd_buf=NULL;
    number1 = 0;
    number2 = 0;
    last_key = 0;
    LCD_ShowString(10,90,"");
    LCD_ShowString(15,90,"0");
    data_in = 0;
    LCD_ClearLn(10,45);
    LCD_ClearLn(10,60);
    LCD_ClearLn(10,75);
    LCD_ShowChar(200,60,' ',16,0);
}
```

تابع Clear

این تابع مقادیر تمام متغیر ها را به مقدار صفر برگردانده و همچنین صفحه نمایش را پاک می کند. در واقع این تابع همان کاری را انجام می دهد که کلید ON/C در تمام ماشین حساب ها انجام می دهد.

```
void config_number(float f_number){
    static float division[]={1,10,100,1000,10000,100000,1000000};
    f_number=(f_number/division[mmz]);
    result=f_number;
}
```

تابع config_number

تابعی که از آن برای کار با اعداد اعشاری استفاده می شود. به طور مثال در این برنامه، دریافت عدد ۱۲٫۳ به این صورت است که تابع calculator این عدد را به صورت ۱۲۳ دریافت می کند و تعداد اعداد بعد ممیز (mmz) را نیز تعیین می کند که در این مثال mmz=1 می شود سپس با فراخوانی تابع config_number عدد ذخیره شده ۱۲۳ به مقدار واقعی ۱۲٫۳ تبدیل می شود.


```
signed char test_key (char key){
if ((key == '.')|| ((key >= '0') && (key <= '9')) return 1;
else return 0;
}
```

تابع test_key

این تابع برای تست دریافت کد اعداد ۰ تا ۹ یا ممیز می باشد و در صورتی کد دریافتی جزو این کدها باشد عدد ۱ و در غیر اینصورت عدد ۰ برگردانده می شود.

```
void ack_opretion(char opretion){
signed char res = 0;
switch (opretion)
{
case '+': number1 += number2; break;
case '-': number1 -= number2; break;
case '*': number1 *= number2; break;
case '/':
if (number2 != 0)
{number1 /= number2; }
else
res = 1;
break;
}
if (res == 0){
if ((number1 >= -999999999) && (number1 <=999999999999999999.999999))
display();
else{
LCD_ClearLn(10,60);
LCD_ClearLn(10,75);
LCD_ShowString(10,90," ");
LCD_ShowString(15,90,"error number");
}
}
if (res == 1){
LCD_ShowString(10,90," ");
LCD_ShowString(15,90,"*ERROR*");
}
}
```

تابع ack_opretion

این تابع برای تشخیص نوع عملیات محاسبه و انجام آن و همچنین نمایش نتیجه محاسبات می باشد.

```
void calculator(){
LCD_ShowString(15,90,"0");
while(1)
{
start:
key = Key_Code() ;
if(key=='c')
{
Clear();
goto start;
}
if (test_key(key)) {
if (data_in != 9)
{
data_in++;
if((key=='.')&&(count!=1))
```

```

{
if(data_in==1) strcat(str,"0. ");
else strcat(str,".");
count=1;
data_in--;
LCD_ClearLn(10,75);
LCD_ShowString(15,75,str);
goto start;
}
if(key=='.')
{
data_in--;
goto start;
}

if(count==1) mmz++;
key=key-'0';
sprintf(helper,"%d",key);
strcat(save_str,helper);
strcat(str,helper);
LCD_ClearLn(10,75);
LCD_ShowString(15,75,str);
}
else{
if(key!='') LCD_ShowChar(200,60,key,16,0);
if (data_in != 0){
if (last_key == 0){
LCD_ClearLn(10,45);
LCD_ClearLn(10,60);
LCD_ShowString(15,45,str);
number1 =atof(save_str);
*save_str=NULL;
*str=NULL;
config_number(number1);
number1=result;
count=0;
mmz=0;
}
else{
LCD_ClearLn(10,60);
LCD_ShowString(15,60,str);
number2 = atof(save_str);
*save_str=NULL;
*str=NULL;
config_number(number2);
number2=result;
count=0;
mmz=0;
}
}
}

data_in = 0;
if (last_key != 0) ack_opretion(last_key);
if (key != '=')
last_key = key;
else
last_key = 0;
if((data_in==0)&&(last_key!=0))
LCD_ShowString(15,45,lcd_buf);
}
}
}
}

```

تابع calculator

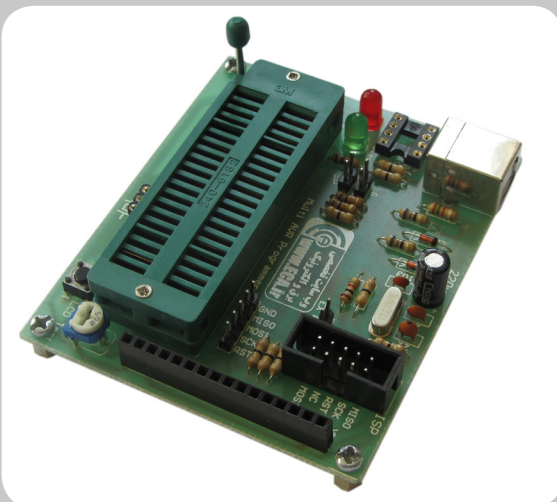
این تابع، تابع محاسبه گر برنامه می باشد و تابع ارتباط دهنده تمام توابع به یکدیگر می باشد. اعداد و عملگرهایی که از سوی کاربر وارد می شود، توسط این تابع تحویل گرفته شده و پردازش های لازم از جمله ذخیره، نمایش و... بر روی آنها انجام می شود تا به نتیجه موردنظر بیانجامد.

```
int main(){
SystemInit();
LCD_Init();
Touch_Init();
LCD_Bitmap(0,0,240,320,gImage_calcu);
calculator();
}
```

تابع main

تابع اصلی برنامه که تنظیمات اولیه میکروکنترلر، LCD و تاج در این تابع انجام می پذیرد و همچنین نمایش تصویر ماشین حساب در LCD در این تابع انجام می پذیرد.

Multi AVR Programmer

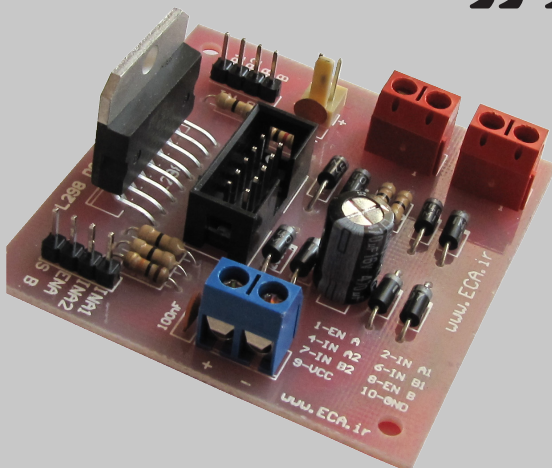


این پروگرامر علاوه بر استفاده از تکنولوژی USB جهت اتصال به رایانه دارای امکانات و ویژگی های دیگری بوده که می تواند محدوده وسیعی از احتیاجات شما را برآورده سازد. امکان تست LCD، پروگرام کردن میکروهای معیوب، رنج وسیع از میکروهای قابل برنامه ریزی، هماهنگی با انواع سیستم عامل ها و... از امکانات ویژه این پروگرامر است که در ادامه تمامی آنها معرفی می شوند.

- ▶ هماهنگی با پورت های USB ۱.۱ و USB ۲.۰
 - ▶ بدون نیاز به تغذیه خارجی
 - ▶ مجهز به کانکتور ISP خروجی
 - ▶ پروگرام کردن تمام میکروها فقط از طریق یک سوکت ZIF
 - ▶ سازگاری با سیستم عامل های (Linux / Mac OS X / Windows)
 - ▶ پشتیبانی از تمامی میکروکنترلرهای خانواده AVR
 - ▶ پشتیبانی از حافظه های سریال سری ۲۴Cxx
 - ▶ توانایی تست LCD های کاراکتری
 - ▶ استفاده از سوکت ZIF برای پروگرام کردن، جهت جلوگیری از صدمه رسیدن به تراشه
 - ▶ DVD شامل نرم افزارهای مورد نیاز به همراه تمامی دیتاشیت تمامی میکروها
 - ▶ دارای ۱۲ ماه گارانتی
- توضیحات تکمیلی :

<http://eshop.eca.ir/link/489.php>

درایور راه انداز موتور L298



یکی از کارهای که می توان با میکروکنترلر انجام داد کنترلر سرعت و یا تغییر جهت انواع موتورهای الکتریکی می باشد. میکروکنترلرها به علت کم بودن جریان خروجی به تنهایی قادر به راه اندازی موتورهای الکتریکی نبوده و از درایورها جهت کنترل موتورهای الکتریکی استفاده می نمایند. یکی از درایورهای پر کاربرد جهت استفاده برای موتورهای الکتریکی درایور L298 است که قادر به کنترل ۲ موتور با ولتاژ ۵۰ ولت و حداکثر جریان ۲ می باشد. ماژول فوق با استفاده از درایور موتو L298 طراحی گردیده و توانایی کنترل همزمان ۲ موتور DC را دارد.

حداکثر جریان قابل تامین برای هر موتور : ۲A
حداکثر ولتاژ قابل تامین هر موتور : ۵۰V
قابل راه اندازی با تمام میکروکنترلرها

توضیحات تکمیلی :

<http://eshop.eca.ir/link/87106.php>

پروگرامر و دیباگر J-LINK V8.0

برخی از ویژگی های مهم این دستگاه :

قابلیت آپدیت



۱۲ ماه
گارانتی تعویض



- ارتباط با کامپیوتر از طریق پورت USB 2.0
- توانایی پروگرام و دیباگ در کامپایلرهای مختلف (IAR, Keil, ...)
- پشتیبانی کامل از تمام میکروکنترلرهای

ARM7/ARM9/ARM11 (Atmel, Philips, Cortex-M0/M1/M3,...)

- پشتیبانی از Serial Wire Debug (SWD)
- پشتیبانی از Serial Wire Viewer (SWV)
- قابلیت شناسایی اتوماتیک قطعات
- سرعت JTAG تا 12MHZ
- سرعت دانلود تا 720 Kbytes/second
- سرعت DCC تا 800 Kbytes/second
- سازگاری کامل با IAR Embedded Workbench IDE
- بدون نیاز به تغذیه (تامین تغذیه از طریق پورت USB)
- قابلیت نمایش JTAG signal و قابلیت اندازه گیری target voltage
- پشتیبانی از دستگاههای متعدد
- دارای سیستم plug and play
- دارای کانکتور JTAG ۲۰ پین استاندارد
- رنج ولتاژ کاری بالا از 1.2V تا 3.3V (قابلیت کار تا 5V)
- دارای کابل USB و کابل ریبون ۲۰ پین
- دارای سیستم تغذیه هوشمند (محافظت در برابر اضافه بار)
- قابلیت استفاده از JTAG به صورت ۱۴ پین
- پشتیبانی از Memory viewer
- پشتیبانی از تمامی نرم افزارهای Flash programming
- پشتیبانی از سیستم Flash DLL
- پشتیبانی از Software Developer Kit (SDK)
- پشتیبانی از Embedded Trace Buffer (ETB)

**سازگاری کامل با
ویندوز ۷**

توضیحات تکمیلی : <http://eshop.eca.ir/link/506.php>

پروژه پخش فایل های صوتی Audio Player



نویسنده: یوسف خداپرست
bemu14@yahoo.com

هدف از این پروژه کار با ماژول VS1003B جهت پخش فایل های MP3 ذخیره شده در حافظه MMC می باشد. کاربرد ماژول VS1003B ضبط و پخش صدا می باشد که در این پروژه، قابلیت پخش صدای این ماژول به کار گرفته شده است.

در این پروژه پروتکل SPI، برای ارتباط میکروکنترلر با حافظه MMC به کار گرفته شده است و برای ارتباط میکروکنترلر با ماژول VS1003B از پایه های GPIO میکروکنترلر بهره گرفته شده است. البته ماژول VS1003B قابلیت برقراری ارتباط از طریق پروتکل SPI را نیز دارد. همچنین در این پروژه از LCD کاراکتری، به عنوان واسط گرافیکی میان کاربر و میکروکنترلر استفاده شده است.

فایل هایی که برای راه اندازی قسمت های مختلف به کار برده شده است :

VS1003B.h و VS1003B.c : هدر و سورس مربوط به راه اندازی ماژول VS1003B

Lcd.h : کتابخانه ای برای راه اندازی LCD کاراکتری با میکروکنترلر های LPC17xx و LPC23xx

FS_CM3.lib : کتابخانه ای استاندارد برای کار با حافظه های فلش NAND و NOR و کارت های حافظه از جمله MMC

می باشد. برای استفاده از این کتابخانه باید فایل File_Config.c ، در مسیر پروژه موجود باشد و البته برای دسترسی به کتابخانه FS_CM3.lib در مسیر نرم افزار keil باید لایسنس مربوط به آن نصب شده باشد. در اینصورت این کتابخانه

باید از مسیر زیر به پروژه اضافه گردد :

C:\Keil\ARM\RV31\LIB\FS_CM3.lib

File_Config : برای پیکربندی کتابخانه FlashFS استفاده می شود که با استفاده از Wizard نرم افزار Keil می توان آنرا به صورت گرافیکی انجام داد. در این فایل حافظه های مورد استفاده و درایو پیش فرض و چند گزینه دیگر تنظیم می شوند.

SPI_LPC17xx.c : این کتابخانه برای راه اندازی بخش SPI در میکروکنترلر lpc17xx می باشد. که در این پروژه از بخش SPI برای ارتباط میکروکنترلر و حافظه MMC بهره گرفته شده است.

system_LPC17xx.h : این کتابخانه برای تنظیمات اولیه سیستم میکروکنترلر می باشد.

همچنین فایل Retarget.c نیز در مسیر پروژه موجود باشد.

شرح کلی پروژه

قسمت های مختلف پروژه :

هدربرد میکروکنترلر LPC17xx : به عنوان پردازنده مرکزی پروژه

ماژول VS1003B : دیکودر فایل های صوتی

LCD کاراکتری : نمایش عملیاتی که در حال انجام است

حافظه MMC : حافظه نگهدارنده فایل هایی که کاربر قصد پخش آنها را دارد

چهار کلید فشاری : برای انجام عملیات Forward، Back، Pause/Play و Stop توسط کاربر

عملکرد کلی برنامه به اینصورت است که در ابتدای راه اندازی در سطر اول LCD عبارت Audio Player نمایش داده می

شود و وضعیت MMC بررسی می شود اگر MMC داخل سوکت قرار گرفته باشد تعداد فایل های صوتی در سطر دوم

LCD نمایش داده می شود و اگر MMC داخل سوکت قرار نگرفته باشد عبارت insert memory در سطر دوم LCD

نمایش داده می شود و در اینصورت باید MMC در داخل سوکت قرار داده شود سپس با فشار کلید Play/Pause پخش

فایل صوتی اول شروع و اسم آن نیز در LCD نشان داده می شود و دوباره با فشار دادن هریک از کلید ها عملیات متناظر

یعنی Forward، Back، Pause و Stop انجام شده و در LCD نمایش داده می شود.

اگر کارت حافظه MMC در حین اجرای فایل دوباره از سوکت خارج شود عبارات Audio Player و Insert Memory

برای بیان این وضعیت در LCD نمایش داده می شود.

همچنین قابلیت جستجو نیز به این پروژه اضافه شده است به اینصورت که با فشار دادن و نگهداشتن هریک از کلیدهای

Forward یا Back اسم فایل های صوتی یک به یک در LCD نمایش داده می شود و با رها کردن کلید موقع نمایش

اسم فایل مورد نظر، همان فایل به اجرا گذاشته می شود.

شرح جزئی قسمت های مختلف برنامه :

```
#include <stdio.h> /* standard I/O .h-file */
#include <string.h> /* string and memory functions */
#include <LPC17xx.H> /* LPC17xx definitions */
#include "File_Config.h"
#include "VS1003B.c"
#define LCD_LPC1768 /*lcd port and pin definitions */
#define LCD_PORT_0
#define LCD_RS 0
#define LCD_E 1
#define LCD_DB4 2
#define LCD_DB5 3
#define LCD_DB6 4
#define LCD_DB7 5
#include "lcd.h"

static char NAME[64];
static int Pause=0,File_Num=0,F=0,B=0,P=0,S=0,files,CH;
```


در قسمت فوق فایل های سرآیند و الحاقی و همچنین متغیرهای کلی مورد استفاده در توابع مختلف برنامه معرفی شده و همچنین پایه های موردنظر میکروکنترلر برای ارتباط با LCD در این قسمت تعریف شده اند.

```
/*-----*
 *          Flash Memory Card Directory          *
 *-----*/
static void directory (int num)
{
    U32 i,j=0;
    char ch;
    FINFO info;
    files = 0;
    info.fileID = 0;
    while (ffind ("*.mp3",&info) == 0)    /* mp3 file directory */
    {
        j++;
        i = 0;
        while (strlen((const char *)info.name+i) > 64)
        {
            ch = info.name[i+64];
            info.name[i+64] = ch;
            i += 64;
        }
        if(j==num)
        {
            *NAME=NULL;
            strcpy(NAME,&info.name[i]);
        }
        files++;
    }
    info.fileID = 0;
    while (ffind ("*.wav",&info) == 0)    /* wav file directory */
    {
        j++;
        i = 0;
        while (strlen((const char *)info.name+i) > 64)
        {
            ch = info.name[i+64];
            info.name[i+64] = ch;
            i += 64;
        }
        if(j==num)
        {
            *NAME=NULL;
            strcpy(NAME,&info.name[i]);
        }
        files++;
    }
    if (info.fileID == 0)
    {
        lcd_gotoxy(2,1);
        lcd_putsf("No File for play");
    }
}
```

این قسمت از برنامه مربوط به دایرکتوری کارت حافظه MMC می باشد و از فایل های MP3 و WAV موجود در کارت MMC دایرکتوری می گیرد. تابع **directory** یک آرگومان با نام **num** دارد که مربوط به شماره فایل موردنظر در دایرکتوری کارت حافظه می باشد. تابع اصلی برنامه با فراخوانی این تابع به تعداد فایل های صوتی داخل کارت حافظه دسترسی پیدا می کند و توابع عملیاتی برنامه با فراخوانی این تابع به نام فایل های صوتی مختلف موجود در کارت حافظه دسترسی پیدا می کنند.

```
/*-----*
*      Pause the playing file & play the paused file function      *
*-----*/

void Pause_Op(void)
{
    switch(Pause)
    {
        case 0: Pause=1; break;
        case 1: Pause=0; break;
    }
    delay(500);
    lcd_gotoxy(1,1);
    lcd_putsf("Pause ");
    while(Pause==1)
    {
        if((LPC_GPIO0->FIOPIN & 0x10000)==0)
        {
            Pause=0;
            lcd_gotoxy(1,1);
            lcd_putsf("Play ");
            delay(500);
        }
        if((LPC_GPIO0->FIOPIN&0x80000)==0)
        {
            delay(500);
            Pause=0;
            S=1;
            lcd_gotoxy(1,1);
            lcd_putsf("Stop ");
            goto end;
        }
        if((LPC_GPIO0->FIOPIN & 0x20000)==0) {delay(200); F=1; goto end;}
        if((LPC_GPIO0->FIOPIN & 0x40000)==0) {delay(200); B=1; goto end;}
        if((LPC_GPIO0->FIOPIN&0x80000)==0) {delay(200); P=1; goto end;}
    }
    end:
}
```

قسمت فوق از برنامه مربوط به تابع عملیات **Pause** برنامه است که توسط تابع **play** فراخوانی می شود. چون برنامه به صورت **polling** نوشته شده است به همین دلیل در این تابع وضعیت سایر کلید ها نیز بررسی می شود.

```
/*-----*
*      play audio file function      *
*-----*/

static void play (char *name)
{
    FILE *f;
    VS1003B_SoftReset();
```

```

VS1003B_Init();
f = fopen (name,"r");      // open the file for reading
VS1003B_XDCS_L();

while ((CH = fgetc (f)) != EOF) // read the characters from the file
{
    BITWAIT_DREQ_HIGH();
    VS1003B_WRITE_DATA(CH,8);

    if((LPC_GPIO0->FIOPIN & 0x10000)==0) Pause_Op();
    if((LPC_GPIO0->FIOPIN & 0x20000)==0) {delay(200); F=1; goto end;}
    if((LPC_GPIO0->FIOPIN & 0x40000)==0) {delay(200); B=1; goto end;}
    if((LPC_GPIO0->FIOPIN&0x80000)==0)
    {
        delay(100);
        lcd_gotoxy(1,1);
        lcd_putsf("Stop ");
        goto end;
    }
    if(S==1) goto end;
    if(F==1 || B==1 || P==1) goto end;
}
F=1;
end:
VS1003B_XDCS_H();
fclose (f);      // close the input file when done
S=0;
}

```

این تابع عملیات پخش فایل های صوتی را برعهده دارد. بدین صورت که این تابع نام فایل را به عنوان آرگومان ورودی دریافت کرده و عملیات باز کردن فایل، خواندن بایت به بایت فایل، ارسال بایت ها به ماژول VS1003B برای دیکد شدن و اجرا و بستن فایل بعد از بایت انتهایی فایل را انجام می دهد. همچنین اگر هریک از کلیدهای عملیاتی فشار داده شود عملیات خواندن و پخش فایل متوقف شده و برنامه از این تابع خارج می شود. این تابع برای راه اندازی اولیه حافظه

```

/*-----*
*           Initialize a Flash Memory Card           *
*-----*/
static void init_card (void)
{
    U32 retv;
    while ((retv = finit ()) != 0) /* Wait until the Card is ready*/
    {
        if (retv == 1)
        {
            lcd_gotoxy(1,1);
            lcd_putsf("Audio Player ");
            lcd_gotoxy(2,1);
            lcd_putsf("insert Memory");
            delay(100);
        }
    }
    lcd_gotoxy(2,1);
    if (File_Num != 0) lcd_putsf("      ");
}

```


MMC می باشد و همچنین تشخیص قرار گرفتن یا قرار نگرگفتن کارت حافظه در داخل سوکت توسط این تابع انجام می پذیرد.

```
/*-----*
 *      display the file number in directory and file name      *
 *-----*/
void display_Op(void)
{
    lcd_gotoxy(2,1);
    lcd_putsf(" ");
    lcd_gotoxy(2,1);
    lcd_putsf(NAME);
    lcd_gotoxy(1,1);
    lcd_putsf(" ");
    lcd_gotoxy(1,13);
    lcd_puts(File_Num);
    lcd_gotoxy(1,1);
    lcd_putsf("Play");
}
```

نمایش عملیات play و نام فایل در حال پخش توسط این قسمت از برنامه انجام می شود.

```
/*-----*
 *      goto play the Forward file function      *
 *-----*/
void Forward_Op(void)
{
    File_Num++;
    if(File_Num>files) File_Num=1;
    directory(File_Num);
    delay(200);
    display_Op();
    F=0;
    play(NAME);
}
```

عملیات Forward (پرش به فایل بعدی) توسط این تابع انجام می شود.

```
/*-----*
 *      goto play the Back file function      *
 *-----*/
void Back_Op(void)
{
    File_Num--;
    if(File_Num<1) File_Num=files;
    directory(File_Num);
    delay(200);
    display_Op();
    B=0;
    play(NAME);
}
```

عملیات Back (پرش به فایل قبلی) توسط این تابع انجام می شود.

```
/*-----*
 *          start to play function          *
 *-----*/

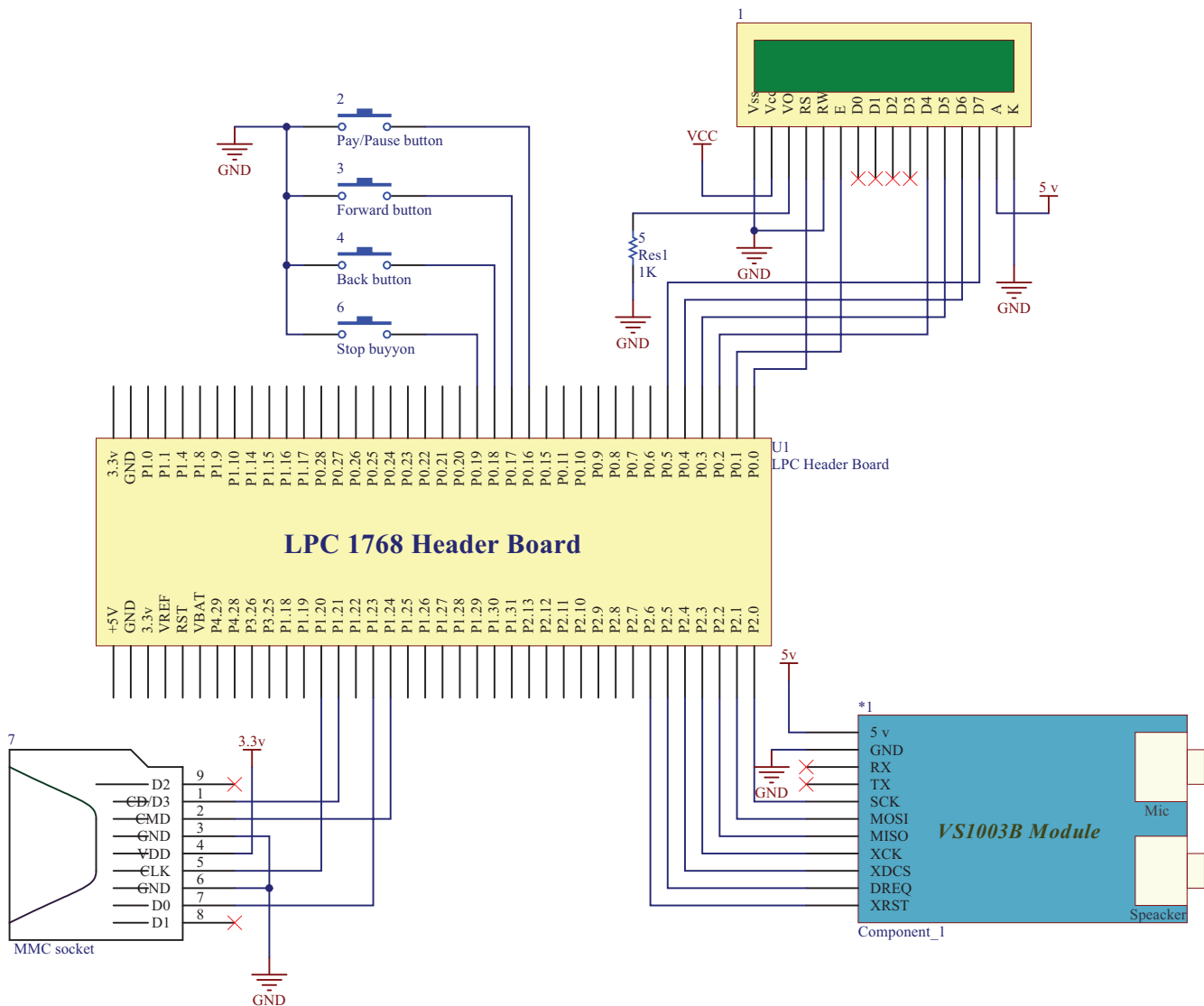
void Play_Op(void)
{
    if (File_Num==0) File_Num=1;
    directory(File_Num);
    delay(200);
    display_Op();
    P=0;
    play(NAME);
}
```

این تابع از برنامه برای شروع پخش فایل های صوتی می باشد.

```
/*-----*
 *          Main function          *
 *-----*/

int main (void)
{
    char buf[10];
    SystemInit();
    lcd_init();
    delay_init();
    VS1003B_SoftReset();
    VS1003B_Init();
    lcd_clear();
    lcd_gotoxy(1,1);
    lcd_putsf("Audio Player");
    init_card ();
    directory(1);
    lcd_gotoxy(2,1);
    sprintf(buf,"%d Audio file",files);
    lcd_putsf(buf);
    while(1)
    {
        init_card ();
        if((LPC_GPIO0->FIOPIN&0x20000)==0 || F==1) Forward_Op();
        if((LPC_GPIO0->FIOPIN&0x40000)==0 || B==1) Back_Op();
        if((LPC_GPIO0->FIOPIN&0x10000)==0 || P==1) Play_Op();
    }
}
```

در این قسمت از برنامه تابع اصلی برنامه آورده شده است. در این تابع تنظیمات اولیه مربوط به میکروکنترلر، ماژول VS1003B و LCD انجام می گیرد و همچنین این تابع توسط حلقه بینهایت وضعیت کلیدهای فشاری را بررسی می کند و توابع متناظر با هر کدام را برای اجرا فراخوانی می کند.



شماتیک مدار

LabVIEW 2012 Professional Full AddOns



نرم افزار Lab View که مخفف عبارت Laboratory Virtual Instrument Engineering Workbench می باشد، یک زبان برنامه نویسی گرافیکی می باشد که به صورت گسترده ای برای کاربرد های مختلفی در صنایع، تحصیلات، آموزش و تحقیقات آزمایشگاهی به عنوان یک مدل استاندارد برای جمع آوری و پردازش داده ها و همچنین وسیله ای جهت کنترل و شبیه سازی ابزارهای مجازی درآمده است. این برنامه یک نرم افزار قدرتمند و قابل انعطاف جهت تجزیه و تحلیل سیستم های اندازه گیری است. در این مجموعه می توانید مجموعه کاملی از Add-On و Modules و Toolkits های موجود برای این نرم افزار را دریافت و استفاده کنید.

این محصول در قالب ده عدد DVD با لایسنس معتبر و بسته بندی شکیل ارائه گشته است

توضیحات تکمیلی :

<http://eshop.eca.ir/link/labview.php>

برد Mini2440 + 3.5" LCD

این برد بر اساس توانایی ها و قابلیت های میکروپروسسور S3C2440 طراحی شده است. این برد در اصل Embedded MotherBoard می باشد، از این رو که با فرکانس کاری ۵۳۳ مگاهرتز و داشتن پورت های همچون سریال، USB، صدا، شبکه، حافظه SD/MMC، VGA و... توانایی های یک مادر برد را دارا می باشد.

همچنین امکان نصب سیستم عامل های Windows CE، NET و Linux بر روی مادر برد به راحتی امکان پذیر می باشد. داشتن نمایشگر LCD رنگی ۳.۵" با دقت ۳۲۰*۲۴۰ پیکسل به همراه صفحه حسگر لمسی و امکان اتصال برد مبدل VGA برای نمایشگر های CRT و LCD معمولی از دیگر قابلیت های جذاب این برد می باشد.

ابعاد این برد ۱۰*۱۰ سانتی متر می باشد. ابعاد بسیار کوچک این برد جهت کاربردهای جیبی (Portable) و

فضاهایی که نیاز به یک مادر برد با سایز بسیار کوچک است، بسیار مناسب می باشد.

این برد نیاز به یک تغذیه ۵ ولت تک داشته و با توجه به عدم تلفات حرارتی فاقد هرگونه فن و سیستم خنک کننده می باشد. پورت ها و خروجی ها :

:: یک عدد پورت شبکه (RJ45) با سرعت 10/5 Mbps

:: سه عدد پورت سریال

:: یک عدد پورت HOST USB

:: یک عدد پورت USB DEVICE

:: یک عدد سوکت کارت های حافظه SD/MMC

:: یک عدد خروجی استریو صوت

:: یک عدد میکروفون

:: یک عدد کانکتور ۱۰ پین JTAG

:: چهار نمایشگر LED

:: شش عدد کلید فشاری همه منظوره

:: یک عدد باز روی خروجی PWM

:: یک عدد پتانسیومتر روی ورودی A/D

:: یک عدد کانکتور ۲۰ پین به منظور اتصال دوربین

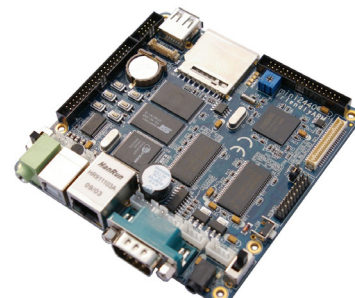
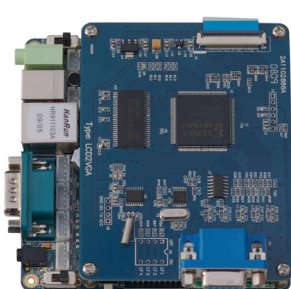
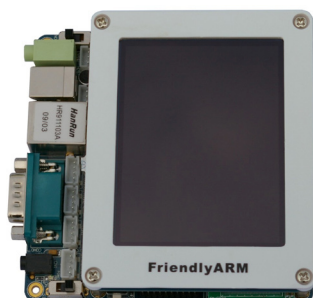
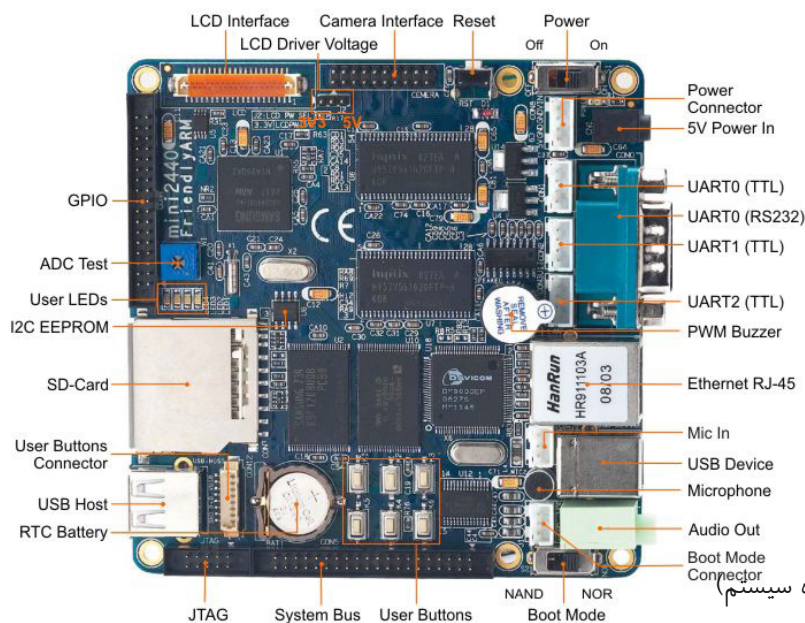
:: یک عدد باتری پشتیبانی حافظه RTC روی برد

:: تغذیه ورودی ۵ ولت به همراه کلید و نمایشگر تغذیه

:: یک عدد کانکتور ۴۰ پین به منظور اتصال LCD

:: دو عدد کانکتور توسعه (۳۴ پایه I/O و ۴۰ پایه گذرگاه سیستم)

:: یک عدد کلید ریست میکرو پروسور

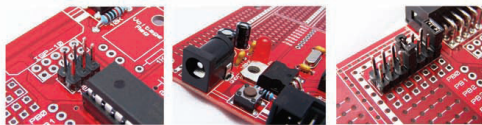


توضیحات تکمیلی : <http://eshop.ea.ir/link/851.php>

معرفی کتاب

بر اساس آخرین استاندارد
وزارت کار و امور اجتماعی
سازمان آموزش فنی و حرفه‌ای
کد استاندارد ۰-۲۳/۹۳/۱/۱

مرجع جامع میکروکنترلر AVR



به همراه CD

رضا سپاس‌یار
یداله مهریزی



مجموعه پروژه‌های پیشرفته اتوماسیون

Collection of Advanced Automation Projects

SIEMENS

تألیف: مهندس مبین محسن زاده
(به اهتمام شرکت صنعتی مبین مامطیر)



- پروژه‌های کاربردی با نرم افزارهای مانیتورینگ WinCC Flexible 2008 , ProTool v6.0
- پروژه‌های ترکیبی از سیستم‌های PLC و SCADA
- به کارگیری منطق فازی در PLC به کمک نرم افزار Fuzzycontrol++
- پروژه‌های کاربردی استفاده از شبکه‌های صنعتی Ethernet و MPI-Profinet-DP, FDL, FMS
- پروژه‌های کاربردی به کمک نرم افزار جامع DCS زیمنس PCS7 v7.1
- پروژه‌ای جامع از ارتباط بین PLC و Inverter-Micro Master با قابلیت مانیتورینگ روی HMI
- پروژه‌ای کاربردی از ارتباط Remote I/O های ET200M با PLC
- بررسی کاربردی تجهیزات هیدرولیک و ...



محتویات DVD شامل:

- فایل اصلی کلیه پروژه‌های کتاب به همراه راه حل آن‌ها به صورت Archive (zip)
- Manual Collections تجهیزات ایزاردقیق و اتوماسیون صنعتی SIEMENS
- نرم افزار قدرتمند شبیه‌ساز پروژه‌های صنعتی Automation Studio v5.0
- نرم افزارهای کاربردی و مقالات تخصصی کاربردی اتوماسیون صنعتی

BIG BANG
Scientific Series

ویرایش دوم

مرجع کامل

اینورترهای

سیستم کنترل موتور



تألیف

مهندس مبین محسن زاده

مهندس امین محسن زاده

به اهتمام:

شرکت مهندسی ایمن تابلو



همراه با DVD

پروژه‌های عملی با میکروکنترلرهای

AVR

مهندس علیرضا ملکی

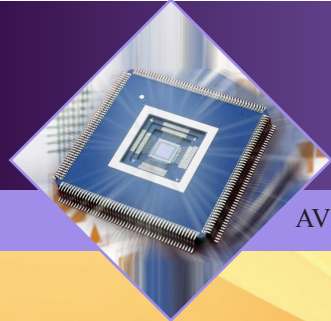
مهندس حمزه نساج پور



و دهها پروژه دیگر

ارسال و دریافت صدا با RFM12b

فیلترهای دیجیتال



AVR , PIC , ARM

Proseccors

پردازنده ها

آموزش کار کردن با پورت های

Xmega

نویسنده: فرامرز



پورت های I/O (درگاه های ورودی - خروجی) برای تبادل دیتا بین میکروکنترلر و device های خارجی که از خارج به میکرو متصل می شوند (اجزای جانبی) مورد استفاده قرار می گیرند. در این قسمت به بررسی پورت های میکروکنترلر Xmega (سری جدید میکروکنترلرهای Atmel) خواهیم پرداخت. قبل از بررسی پورت های این خانواده از میکروکنترلرهای AVR، ویژگی ها و امکانات قدرتمند و منحصر به فردی که پورت های این میکروها دارند آورده شده و سپس به بررسی هر قسمت می پردازیم و در نهایت بخش اول را با انجام دو پروژه جامع در محیط AVR Studio به اتمام می رسانیم.

پورت های Xmega

خانواده Xmega از سری جدید میکروکنترلرهای AVR دارای پورت های انعطاف پذیر و قدرتمندی بوده که هر پورت آن دارای ۸ پین می باشد. لازم به ذکر است که تک تک پایه های هر پورت به صورت بیتی قابل دسترسی می باشد یعنی می توان یک پین از پورت را به صورت مستقل از پایه های دیگر ورود و یا خروجی تعریف کرد. از دیگر ویژگی های پورت های Xmega می توان به وقفه ها، حسگرهای ورودی سنکرون و آسنکرون اشاره کرد. همچنین هر پایه ای از پورت را می توان به صورت Wired-or, Wired-And, Bus-keeper, Totem-pole, invertedio تنظیم کرد. همچنین هر پایه به صورت مجزا از پایه های دیگر دارای مقاومت Pull-Up و Pull-Down داخلی می باشد.

پورت های مجازی

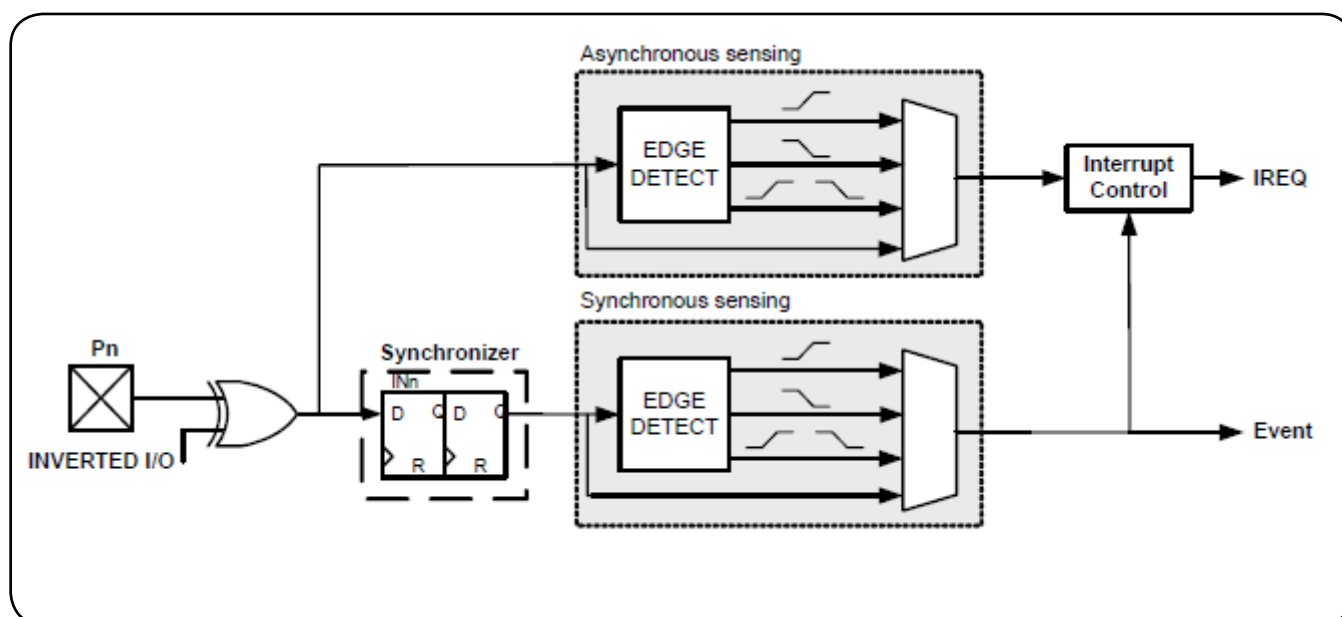
با توجه به محل قرار گیری پورت های مجازی در حافظه ی DATA MEMORY میکروکنترلرهای XMEGA، سرعت دسترسی CPU به این رجیسترها از رجیسترهای پورت های معمولی بیشتر است. بنابراین یکی از تفاوت های پورت های معمولی با پورت های مجازی در سرعت دسترسی CPU به آن ها می باشد. چهار عدد پورت مجازی در میکروهای XMEGA وجود دارد. لازم به ذکر است که پورت های مجازی باید به یکی از پورت های واقعی map شود (متصل گردد) در این صورت نوشتن در رجیستر پورت های مجازی همانند نوشتن در رجیستر پورت های معمولی می باشد.

حسگرهای ورودی

حسگرهای ورودی برای تشخیص لبه و سطح پالس روی پایه ی مورد نظر مورد استفاده قرار می گیرند. حسگرهای ورودی می توانند لبه ی بالا رونده، لبه ی پایین رونده و یا هر دو لبه و یا سطح پایین پالس را تشخیص دهند. بنابراین پایه ای که به عنوان حسگر ورودی مورد استفاده قرار می گیرد، باید حساس به یکی از حالت های قبل پیکربندی شود. لازم به ذکر است که سطح بالای پالس (high level) توسط حسگرهای ورودی تشخیص داده نمی شود بنابراین برای تشخیص سطح بالای پالس باید پایه ی ورودی را حساس به سطح پایین پالس قرار دهیم و سپس معکوسگر پایه ی مورد نظر را فعال کنیم، در این صورت سطح بالای پالس نیز قابل تشخیص خواهد بود.

تمام پایه های I/O دو نوع حسگر ورودی سنکرون و آسنکرون را پشتیبانی می کنند. که حسگرهای ورودی سنکرون برای عمل آشکارسازی نیازمند حضور کلاک (peripheral clock) می باشد در حالی که حسگرهای آسنکرون نیازمند به حضور

کلاک نمی باشند.



شکل شماره ۱

وقفه ی پورت ها

یکی دیگر از قابلیت های پورت های XMEGA نسبت به AVR های معمولی (خانواده MEGA) این است که تمام پایه های I/O میکرو های XMEGA قادر به تولید وقفه می باشند. هر یک از تنظیمات حسگرهای ورودی می تواند برای تولید وقفه مورد استفاده قرار گیرد. استفاده از وقفه بستگی به این دارد که کدام نوع حسگر مورد استفاده قرار می گیرد (حسگر سنکرون یا حسگر آسنکرون) لازم به توضیح است که حسگرهای سنکرون قادر به ایجاد وقفه در تمام تنظیمات حسگرهای ورودی می باشند در حالی که در حسگرهای آسنکرون فقط پایه ی ۲ هر پورت تمام تنظیمات را پشتیبانی می کند. در جداول شماره ۱ و ۲ حالت هایی که حسگرهای سنکرون و آسنکرون پشتیبانی می کنند آورده شده است.

Sense Settings	Supported	Interrupt Description
Rising edge	Yes	Always triggered
Falling edge	Yes	Always triggered
Both edges	Yes	Always triggered
Low level	Yes	Pin level must be kept unchanged during wake up

جدول شماره ۱

Sense Settings	Supported	Interrupt Description
Rising edge	No	-
Falling edge	No	-
Both edges	Yes	Pin value must be kept unchanged during wake up
Low level	Yes	Pin level must be kept unchanged during wake up

جدول شماره ۲

معرفی رجیسترها

رجیستر DIR - (Data Direction Register):

Bit	7	6	5	4	3	2	1	0	
+0x00	DIR[7:0]								DIR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

این رجیستر مانند رجیستر DDR در سری Mega از خانواده AVR عمل می کند. با نوشتن یک منطقی در این رجیستر پین مورد نظر به صورت خروجی و با نوشتن صفر منطقی پین مورد نظر به صورت ورودی پیکر بندی می شود و نحوه مقدار دهی به این رجیستر در AVR Studio به صورت زیر می باشد.

مثال `PORTA_DIR = 0xFF;` `PORTX_DIR = value;`

که X نام پورت می باشد. به عنوان مثال تمام پایه های پورت A در مثال بالا به صورت خروجی تعریف شده است.

رجیسترهای DIRSET, DIRCLR و DIRTGL:

Bit	7	6	5	4	3	2	1	0
+0x01	DIRSET[7:0]							DIRSET
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
+0x02	DIRCLR[7:0]							DIRCLR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
+0x03	DIRTGL[7:0]							DIRTGL
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial Value	0	0	0	0	0	0	0	0

با نوشتن یک منطقی در رجیستر DIRSET بیت متناظر با آن پین در رجیستر DIR یک می شود. با نوشتن یک منطقی در رجیستر DIRCLR بیت متناظر با آن پین در رجیستر DIR صفر می شود. و با نوشتن یک منطقی در رجیستر DIRTGL بیت متناظر با آن پین در رجیستر DIR Toggle (برعکس NOT) می شود.

مثال :

`PORTA_DIR = 0x0F;`

`PORTA_DIRSET = 0xFF0;`

در AVR های سری Mega برای تغییر وضعیت چند پین از یک پورت ابتدا باید مقدار رجیستر مورد نظر را بخوانیم سپس تغییرات مورد نظر را روی مقدار خوانده شده اعمال کنیم و دوباره باید مقدار جدید (تغییر داده شده) را در رجیستر مورد نظر بنویسیم. همانطور که می بینیم این کار زمان بر است ولی در خانواده Xmega این مشکل نیز رفع شده. به عنوان مثال در نمونه بالا پایه های ۰ تا ۳ پورت A خروجی تعریف شده اند. حال اگر در یک جایی از برنامه نیاز باشد که چهار پایه ی دیگر پورت A را نیز خروجی قرار دهیم فقط کافی است مانند مثال بالا بیت های آن پورت را در رجیستر DIRSET یک کنیم با این کار بدون نیاز به خواندن مقدار رجیستر، DIR و سپس تغییر آن، می توانیم به صورت مستقیم وضعیت این رجیستر را تغییر دهیم. پس در واقع این سه رجیستر برای تغییر وضعیت رجیستر DIR مورد استفاده قرار می گیرند که این کار باعث بالا رفتن سرعت پورت ها در خانواده Xmega می شود.

رجیستر OUT :

Bit	7	6	5	4	3	2	1	0
+0x04	OUT[7:0]							OUT
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial Value	0	0	0	0	0	0	0	0

این رجیستر مانند رجیستر PORT در سری mega می باشد. یعنی این رجیستر وضعیت خروجی پایه را تعیین می کند. با نوشتن یک منطقی در این رجیستر پایه ی مورد نظر به سطح منطقی High می رود و با نوشتن صفر منطقی در این رجیستر پایه ی مورد نظر به سطح منطقی Low می رود.

`PORTX_OUT = value` >> `PORTK_OUT = 0xAA;`

که X نام پورت مورد نظر می باشد.

رجیسترهای OUTSET, OUTCLR, OUTTGL :

Bit	7	6	5	4	3	2	1	0
+0x05	OUTSET[7:0]							
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
+0x06	OUTCLR[7:0]							
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
+0x07	OUTTGL[7:0]							
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial Value	0	0	0	0	0	0	0	0

با نوشتن یک منطقی در رجیستر OUTSET بیت پایه ی مورد نظر در رجیستر OUT یک می شود و پایه به سطح High می رود. با نوشتن یک منطقی در رجیستر OUTCLR بیت متناظر با آن پین در رجیستر OUT صفر می شود و با نوشتن یک منطقی در رجیستر OUTTGL بیت متناظر با آن پین در رجیستر OUT، Not (برعکس) می شود. کاربرد این رجیسترها همانطور که در توضیح رجیسترهای DIR بیان شد برای بالا بردن سرعت پورت های Xmega در هنگام تغییر وضعیت رجیسترها می باشد.

رجیستر IN :

Bit	7	6	5	4	3	2	1	0
+0x08	IN[7:0]							
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial Value	0	0	0	0	0	0	0	0

این رجیستر برای خواندن وضعیت پورت ها مورد استفاده قرار می گیرد.
(مثال)

a=PORTB_IN;

در مثال بالا وضعیت پورت B خوانده شده و در متغیر a ریخته می شود.

رجیستر INTCTRL :

Bit	7	6	5	4	3	2	1	0
+0x09	-	-	-	-	INT1LVL[1:0]		INT0LVL[1:0]	
Read/Write	R	R	R	R	R/W	R/W	R/W	R/W
Initial Value	0	0	0	0	0	0	0	0

با مقدار دهی به این رجیستر می توان وقفه های موجود در پورت را فعال، وسطح اولویت وقفه ها را تعیین کرد.
بیت های INT0LVL[1: 0]: مقدار این بیت ها طبق جدول شماره ۴ وظیفه ی فعال سازی و تعیین اولویت وقفه صفر پورت را بر عهده دارند.
بیت های INT1LVL[1: 0]: مقدار این بیت ها طبق جدول شماره ۴ وظیفه ی فعال سازی و تعیین اولویت وقفه یک پورت

را بر عهده دارند.

Interrupt Level Configuration	Group Configuration	Description
00	OFF	Interrupt disabled.
01	LO	Low-level interrupt
10	MED	Medium-level interrupt
11	HI	High-level interrupt

جدول شماره ۴

اولویت وقفه ها در بخش وقفه ها به صورت کامل مورد بحث قرار خواهد گرفت.

رجیستر INT0MASK:

Bit	7	6	5	4	3	2	1	0
+0x0A	INT0MSK[7:0]							INT0MASK
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial Value	0	0	0	0	0	0	0	0

با نوشتن یک منطقی در این رجیستر پایه های متناظر با آن بیت ها به عنوان ورودی وقفه صفر پورت انتخاب می شود.

رجیستر INT1MASK:

Bit	7	6	5	4	3	2	1	0
+0x0B	INT1MSK[7:0]							INT1MASK
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial Value	0	0	0	0	0	0	0	0

با نوشتن یک منطقی در این رجیستر پایه های متناظر با آن بیت ها به عنوان ورودی وقفه یک پورت انتخاب می شود.

رجیستر INTFLAGS:

Bit	7	6	5	4	3	2	1	0
+0x0C	-	-	-	-	-	-	INT1IF	INT0IF
Read/Write	R	R	R	R	R	R	R/W	R/W
Initial Value	0	0	0	0	0	0	0	0

بیت پرچم ورودی های وقفه صفر (INT0IF): این بیت در صورتی یک می شود که یک تغییر در روی ورودی های وقفه صفر پورت با توجه به تنظیمات حسگر ها اتفاق می افتد.

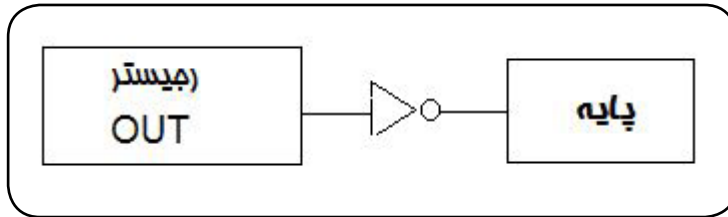
بیت پرچم ورودی های وقفه صفر (INT1IF): این بیت در صورتی یک می شود که یک تغییر در روی ورودی های وقفه یک پورت با توجه به تنظیمات حسگر ها اتفاق می افتد.

رجیستر PINnCTRL:

Bit	7	6	5	4	3	2	1	0
	SRLEN	INVEN	OPC[2:0]			ISC[2:0]		
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial Value	0	0	0	0	0	0	0	0

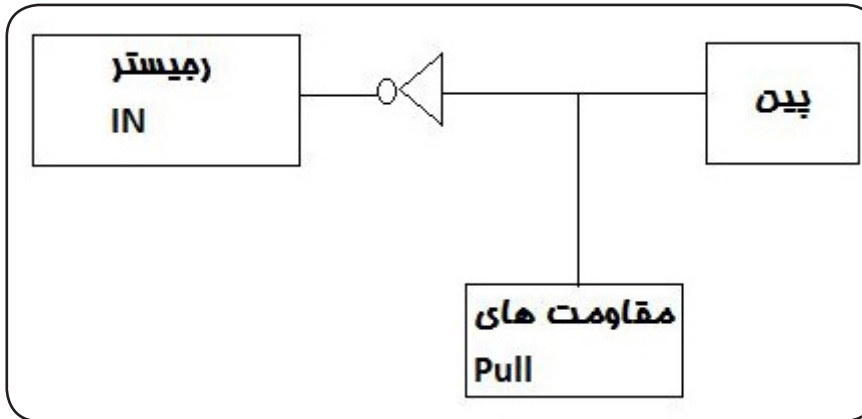
این رجیستر برخی تنظیمات ویژه را برای پایه مورد نظر انجام می دهد. این رجیستر به صورت جداگانه برای هر پین از پورت ها وجود دارد.

SRLEN: با یک شدن این بیت کنترل **Slew Rate** فعال می شود. با فعال شدن این ویژگی، **Rise Time** (زمان صعود) و **Falling Time** (زمان نزول) ولتاژ خروجی (پالس) افزایش یافته و از تغییرات سریع ولتاژ جلوگیری می کند. که مقدار این افزایش بین 50% تا 150% بوده و به ولتاژ، دما و جریان بار بستگی دارد.



شکل شماره ۲

INVEN: با فعال شدن این بیت (نوشتن یک منطقی در این بیت) اگر پایه به عنوان خروجی تعریف شده باشد. وضعیت خروجی پایه برعکس (Not) می شود. یعنی اگر مقدار رجیستر **Out** برای پایه ی مورد نظر یک باشد خروجی در سطح پایین (Low) و اگر رجیستر



شکل شماره ۳

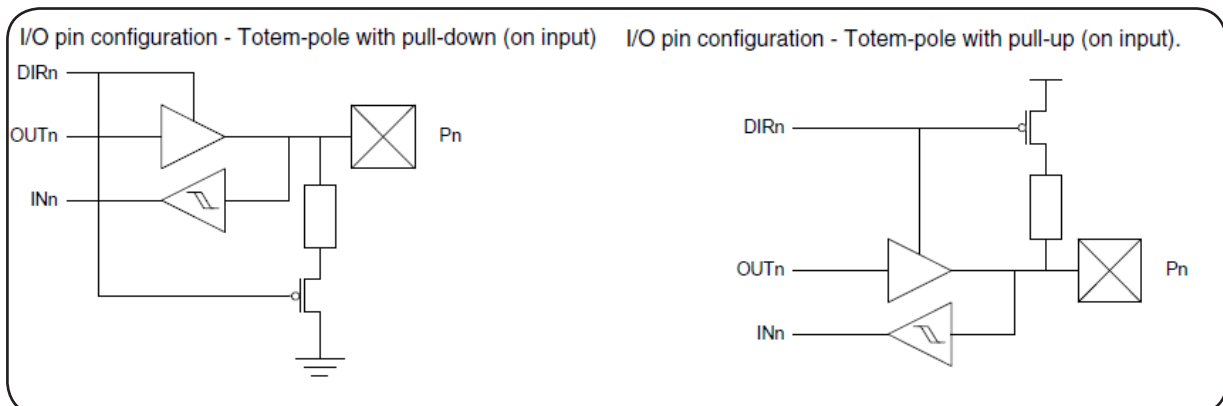
Out برای پایه صفر باشد وضعیت خروجی پایه یک (High) می شود. در واقع در حالت خروجی این ویژگی را می توان به صورت شکل شماره ۲ مدل کرد.

ولی اگر پایه به صورت ورودی پیکر بندی شده باشد. وضعیت پایه پس از تأثیر گذاری مقاومت های بالا کش و پایین کش NOT شده و در رجیستر **IN** قرار می گیرد. بیت های **OpC [2: 0]** با توجه به مقدار این بیت ها می توان یکی از قابلیت های جدول شماره ۵ را برای پایه مورد نظر تعریف کرد.

OPC[2:0]	Group Configuration	Description	
		Output Configuration	Pull Configuration
000	TOTEM	Totem-pole	(N/A)
001	BUSKEEPER	Totem-pole	Bus-keeper
010	PULLDOWN	Totem-pole	Pull-down (on input)
011	PULLUP	Totem-pole	Pull-up (on input)
100	WIREDOR	Wired-OR	(N/A)
101	WIREDAND	Wired-AND	(N/A)
110	WIREDORPULL	Wired-OR	Pull-down
111	WIREDANDPULL	Wired-AND	Pull-up

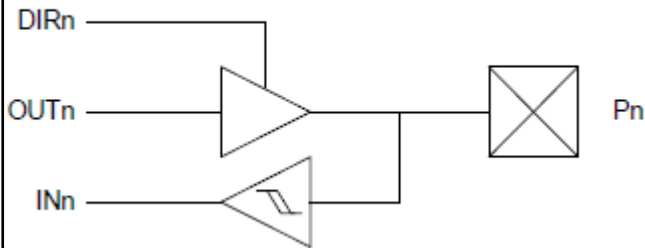
جدول شماره ۵

TOTEM: این ویژگی مانند پورت های سری **Mega** از خانواده **AVR** می باشد. یعنی با فعال شدن این ویژگی جریان



شکل شماره ۵

I/O pin configuration - Totem-pole (push-pull)

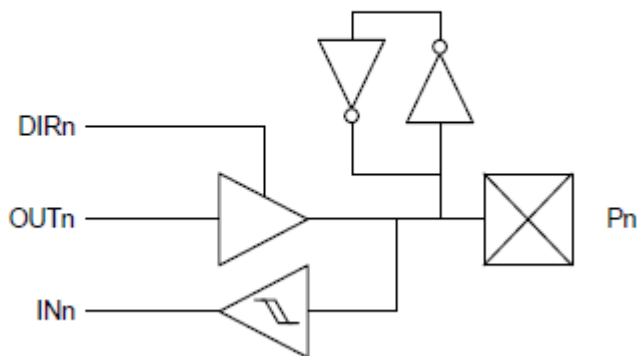


شکل شماره ۴

خروجی پایه از یک درایور (تقویت کننده جریان) تامین می شود (اگر پایه خروجی تعریف شود).

Buskeeper: با فعال شدن این ویژگی برای پایه ی مورد نظر، اگر پایه به صورت خروجی پیکر بندی شده باشد. با تغییر وضعیت از حالت خروجی به ورودی آخرین وضعیت پایه نگهداری می شود. به عنوان مثال اگر وضعیت پایه در حالت خروجی High باشد و وضعیت پایه از حالت خروجی به ورودی تغییر کند مقاومت Pull Up داخلی فعال شده و سطح پایه را در حالت ورودی همچنان در یک منطقی (High) نگه می دارد.

I/O pin configuration - Totem-pole with bus-keeper



شکل شماره ۶

Pull Down: مقاومت پایین کش داخلی را فعال می کند.

Pull Up: مقاومت بالا کش داخلی را فعال می کند.

WIREDOR: اگر چندین پایه با فعال بودن این ویژگی به هم وصل شوند (این ویژگی این امکان را می دهد که چندین پایه می توانند در حالت خروجی به هم متصل شوند). وضعیت این پایه ها با هم OR می شوند یعنی اگر وضعیت یکی از پایه ها یک باشد خروجی کل یک می شود. و اگر وضعیت همه ی پایه ها صفر باشد در صورت نبود مقاومت پایین کش خارجی وضعیت کل به حالت tri-state

(امپدانس بالا) می رود. و اگر مقاومت پایین کش وصل شود وضعیت کل صفر می شود.

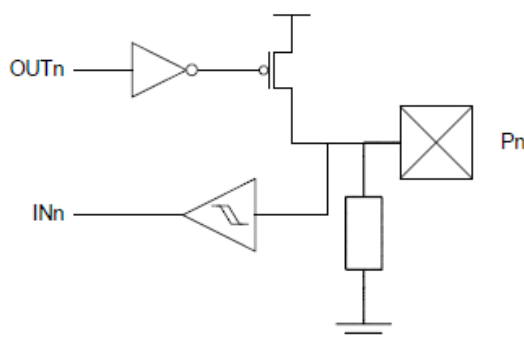
WIREDAND: این ویژگی هم مانند ویژگی WIREDOR می باشد فقط با این تفاوت که وضعیت خروجی پایه ها در این حالت AND منطقی می شوند و خروجی کل در صورتی یک است که وضعیت همه ی پایه های متصل شده به هم یک باشد.

WIREDORPULL: این ویژگی همانند ویژگی WIREDOR بوده فقط با این تفاوت که یک مقاومت پایین کش داخلی هم در این حالت فعال می شود. یعنی اگر وضعیت همه ی پایه های متصل شده به هم صفر باشد وضعیت کل صفر می شود. (tri-state نمی شود).

WIREDAND PULL: این ویژگی همانند ویژگی WIREDAND بوده فقط با این تفاوت که یک مقاومت بالا کش داخلی هم در این حالت فعال می شود.

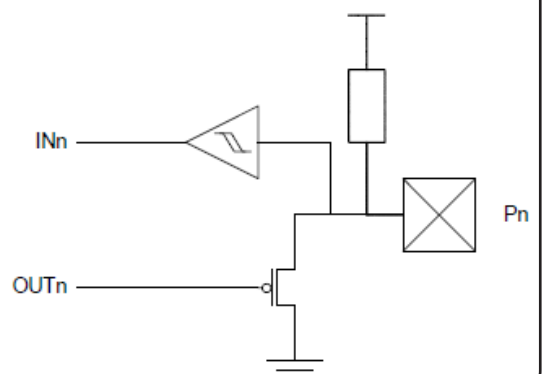
بیت های [2: 0] ISC: مقدار این بیت ها طبق جدول شماره ۶ یکی از تنظیمات حسگرها را برای پایه ی مورد نظر انتخاب می کنند.

Output configuration - Wired-OR with optional pull-down.



شکل شماره ۷

Output configuration - Wired-AND with optional pull-up.



شکل شماره ۸

ISC[2:0]	Group Configuration	Description
000	BOTHEDGES	Sense both edges
001	RISING	Sense rising edge
010	FALLING	Sense falling edge
011	LEVEL	Sense low level ⁽¹⁾
100		Reserved
101		Reserved
110		Reserved
111	INPUT_DISABLE	Digital input buffer disabled ⁽²⁾

جدول شماره ۶

به عنوان مثال می خواهیم ویژگی INVERT (معکوس گر) و Buskeeper را برای پین صفر پورت A فعال کنیم. برای این منظور در محیط AVR – Studio این گونه عمل کنیم.

>>PORTA_PIN0CTRL = 0X48;

INVERT Buskeeper

رجیستر MPCMASK (Multi_pin configuration mask register) :

Bit	7	6	5	4	3	2	1	0
+0x00	MPCMASK[7:0]							MPCMASK
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial Value	0	0	0	0	0	0	0	0

با مقدار دهی به این رجیستر تنظیمات مشترک چندین پایه ی یک پورت در یک زمان صورت می گیرد. عملکرد این رجیستر را با یک مثال توضیح می دهیم.

فرض کنید می خواهیم مقاومت بالا کش تمام پایه های پورت A را فعال کنیم. برای این کار با مقدار دهی به رجیستر PINnCTRL که برای هر پایه به صورت مجزا وجود دارد، این عمل را به صورت زیر انجام می دهیم.

PORTA_PIN0CTRL=0X18;

PORTA_PIN1CTRL=0X18;

تا

PORTA_PIN7CTRL=0X18;

که همانطور که دیده می شود برای این کار باید ۸ رجیستر را مقدار دهی کنیم که این کار زمان CPU را تلف می کند. حال این عمل را توسط رجیستر MPCMASK انجام می دهیم.

برای این کار بیت های پایه هایی را که می خواهیم مقاومت بالا کش آن ها فعال شود در این رجیستر یک می کنیم و سپس در خط بعدی مقاومت بالا کش یکی از پایه های همان پورت را فعال می کنیم در نتیجه این تنظیمات به کل پایه هایی که در رجیستر MPCMASK یک شده اند اعمال می شود.

PORTCFG_MPCMASK=0XFF;

PORTA_PIN0CTRL=0X18;

بنابراین این رجیستر تنظیمات رجیستر PINnCTRL خط پایین خودش را به تمام پایه هایی که بیت متناظر آن پایه ها در این رجیستر یک شده است، اعمال می کند.

رجیستر VPCTRLA:

Bit	7	6	5	4	3	2	1	0
+0x02	VP1MAP[3:0]				VP0MAP[3:0]			
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial Value	0	0	0	0	0	0	0	0

بیت های [0: 3] VP1MAP: مقدار این بیت ها طبق جدول ۶-۱۳ یکی از پورت ها را به پورت مجازی یک map می کند(متصل می کند).

بیت های [0: 3] VP0MAP: مقدار این بیت ها طبق جدول ۶-۱۳ یکی از پورت ها را به پورت مجازی صفر map می کند(متصل می کند).

رجیستر VPCTRLB:

Bit	7	6	5	4	3	2	1	0
+0x03	VP3MAP[3:0]				VP2MAP[3:0]			
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial Value	0	0	0	0	0	0	0	0

بیت های [0: 3] VP3MAP: مقدار این بیت ها طبق جدول ۶-۱۳ یکی از پورت ها را به پورت مجازی سه map می کند(متصل می کند).

بیت های [0: 3] VP2MAP: مقدار این بیت ها طبق جدول ۶-۱۳ یکی از پورت ها را به پورت مجازی دو map می کند(متصل می کند).

Table 13-6. Virtual Port mapping.

VPnMAP[3:0]	Group Configuration	Description
0000	PORTA	PORTA mapped to virtual Port n
0001	PORTB	PORTB mapped to virtual Port n
0010	PORTC	PORTC mapped to virtual Port n
0011	PORTD	PORTD mapped to virtual Port n
0100	PORTE	PORTE mapped to virtual Port n
0101	PORTF	PORTF mapped to virtual Port n
0110	PORTG	PORTG mapped to virtual Port n
0111	PORTH	PORTH mapped to virtual Port n
1000	PORTJ	PORTJ mapped to virtual Port n
1001	PORTK	PORTK mapped to virtual Port n
1010	PORTL	PORTL mapped to virtual Port n
1011	PORTM	PORTM mapped to virtual Port n
1100	PORTN	PORTN mapped to virtual Port n

جدول شماره ۷

Table 13-6. Virtual Port mapping. (Continued)

VPnMAP[3:0]	Group Configuration	Description
1101	PORTP	PORTP mapped to virtual Port n
1110	PORTQ	PORTQ mapped to virtual Port n
1111	PORTR	PORTR mapped to virtual Port n

جدول شماره ۸

رجیستر CLKEVOUT:

این رجیستر کلاک CPU و خروجی کانال 0 Event system را به پایه ی ۷ یکی از پورت ها طبق جدول زیر متصل می کند.

Bit	7	6	5	4	3	2	1	0
+0x04	-	-	EVOUT[1:0]		-	-	CLKOUT[1:0]	
Read/Write	R	R	R/W	R/W	R	R	R/W	R/W
Initial Value	0	0	0	0	0	0	0	0

بیت های Clk Out [1: 0]: این بیت ها طبق جدول زیر کلاک CPU را به پایه ی ۷ یکی از پورت ها متصل می کند.

Clock output configurations

CLKOUT[1:0]	Group Configuration	Description
00	OFF	Clock out disabled
01	PC7	Clock output on Port C pin 7
10	PD7	Clock output on Port D pin 7
11	PE7	Clock output on Port E pin 7

جدول شماره ۹

بیت های EV Out [1: 0]: با مقدار دهی به این بیت ها هم، خروجی کانال صفر Even system به پایه ۷ یکی از پورت ها طبق جدول زیر متصل می شود.

Event Channel 0 output configurations

EVOUT[1:0]	Group Configuration	Description
00	OFF	Event out disabled
01	PC7	Event Channel 0 output on Port C pin 7
10	PD7	Event Channel 0 output on Port D pin 7
11	PE7	Event Channel 0 output on Port E pin 7

جدول شماره ۱۰

به عنوان مثال می خواهیم کلاک CPU را به پایه ی ۷ پورت C وصل کنیم. برای این کار در محیط AVR – studio این گونه عمل می کنیم.

PORTCFG_CLKEVOUT = 0x01;

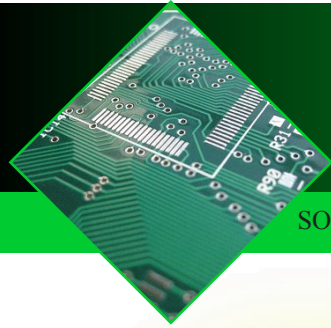
SIMATIC Step 7 (5.5) Pro Edition 2010



شرکت زیمنس برای سیستمهای اتوماسیون، سری PLC های S400-S300-S200 (معروفترین و پر کاربردی ترین سری در ایران) را تولید کرده که برنامه ریزی و پیکربندی سخت افزاری این سری از PLC ها را میتوان با استفاده از نرم افزار SIMATIC MANAGER انجام داد. زبان برنامه نویسی مورد استفاده در این نرم افزار STEP ۷ است که در حالت گرافیکی (استفاده از بلوکهای آماده موجود در برنامه) بسیار ساده نیز هست. بیش از 90% از نقشه های موجود در صنعت (برنامه نوشته شده در PLC) با دیاگرامهای LADER است که همانند کنتاکتها و رله ها در مدارهای کنترل رسم میشوند. پس از برنامه نویسی در این نرم افزار با استفاده از قسمت SIMULATION نرم افزار میتوان برنامه نوشته شده را شبیه سازی کرده و در صورت وجود خطا آن را برطرف کرد.

توضیحات تکمیلی :

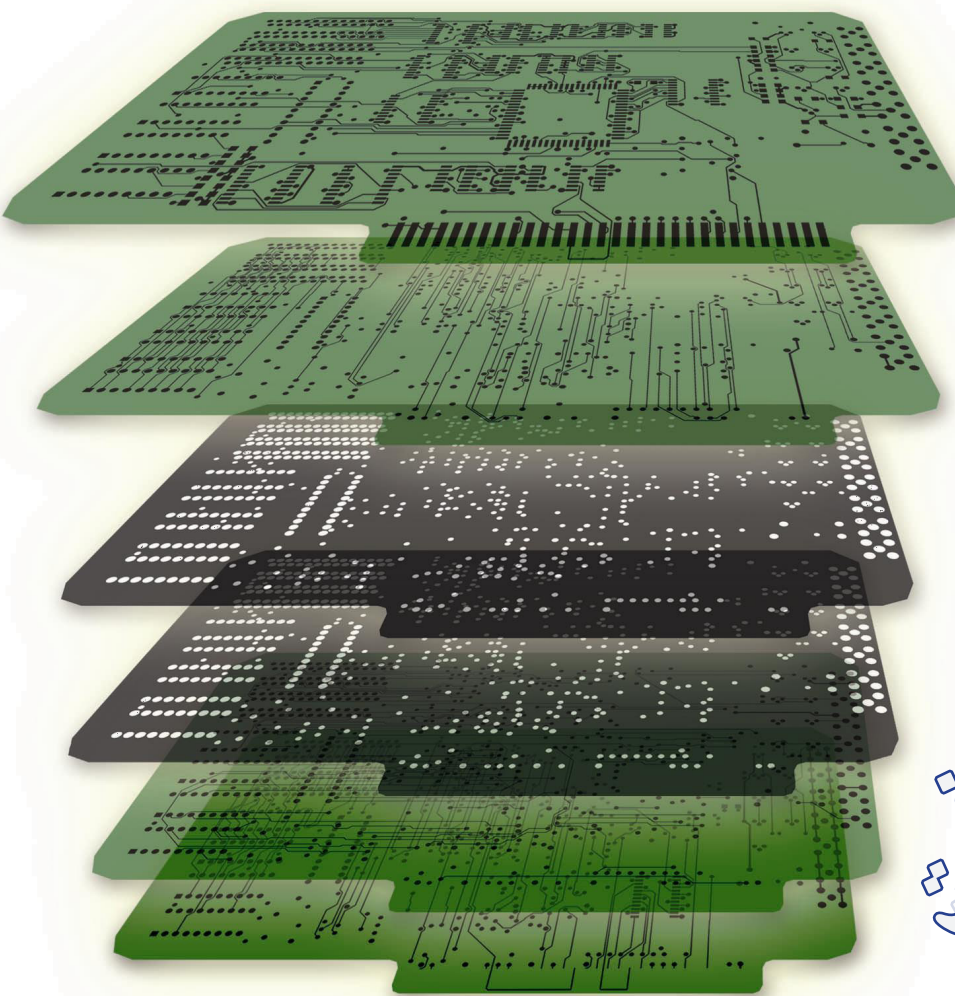
<http://eshop.eca.ir/link/simatic.php>



PCB

SOLDERING AND MANUFACTURING

مدار چاپی



بردهای چند لایه

نویسنده: گروه صنعتی کارالکترونیک

www.karapcb.com

استفاده از بردهای چند لایه به طور قابل ملاحظه ای در طول سال های گذشته افزایش یافته است. عمده ترین مزیت بردهای چند لایه، ایجاد ارتباطات داخلی در میان لایه ها و کاستن از حجم و وزن برد مدار چاپی است. در بردهای چند لایه می توان ارتباطات داخلی را در لایه های درونی جای داده و لذا در لایه های بیرونی فضای کافی برای Heat Sink ایجاد نمود و نیز امکان ایجاد لایه هایی به عنوان سطوح جاذب نویز Shield و Ground و شاسی مثبت و منفی Power وجود دارد.

فرآیند تولید بردهای چند لایه اساسا توسعه همان فرآیندهایی است که در ساخت بردهای دورو با سوراخ متالیزه مورد استفاده می باشند. مدارات چند لایه شامل تعدادی لایه های جداگانه با الگوی مداری مرتبط اما متفاوتند که بر روی یکدیگر صفحه بندی شده و تحت فشار و حرارت با استفاده پیوند دهنده ای که با نام Pre - Preg یا رزین B - Stage خوانده می شود به یکدیگر متصل می شوند. ورق های لایه Pre - Preg در عین حال نقش عایق (دی الکتریک) را به منظور جداسازی الکتریکی لایه ها از یکدیگر به عهده داشته و ارتباط الکتریکی میان لایه های مختلف در نقاط معین از طریق ایجاد سوراخ های متالیزه برقرار می گردد. این ورق ها شامل رزینی هستند که پلیمریزاسیون آن ها کامل نشده و در مرحله B - Stage نگه داشته شده اند. رزین در این مرحله هنوز قابلیت ذوب شدن دارد و لذا با افزایش درجه حرارت مجددا نرم شده و تحت فشار و حرارت پلیمریزاسیون آن تکمیل می گردد. از مشخصه هایی که در انتخاب Pre - preg مناسب در نظر گرفته می شود، علاوه بر ضخامت آن که در تعیین ضخامت نهایی برد چند لایه موثر است، زمان ژل شدگی، روان شدگی رزین و مقدار رزین نیز مهم می باشد.

در استاندارد های نظامی طول دوره مجاز نگه داری Pre - Preg در دمای پنج درجه سانی گراد را حداکثر شش ماه اعلام می دارند. در حالی که اگر دمای انبار به 20°C افزایش یابد، طول این دوره تقریبا به سه ماه کاهش خواهد یافت. بدیهی

است که استفاده از Pre-Preg پس از طی دوره زمانی منوط به انجام آزمایش ها و بررسی مشخصه های آن نظیر زمان ژل شدگی، روان شدگی رزین، میزان رطوبت و... می باشد.

جهت تولید بردهای چند لایه ابتدا باید لایه های داخلی آماده گردند. به این منظور ابتدا صفحه مدار چاپی اولیه به ضخامت مورد نظر را انتخاب کرده و پس از برش به ابعاد مورد نیاز و انجام فرآیندهای ضروری از قبیل اکسیدزدایی و چربی گیری به طریق شیمیایی یا مکانیکی و گاهی هم توافمان، با بکار گیری فتورزیست و نوردهی و متعاقب آن مس بری، طرح مداری مربوط را بر روی آن ایجاد می نمایند. در مراحل بعدی به منظور چسبندگی بیشتر سطح برد (لایه های داخلی) به رزین Pre-Preg، سطح مسی مدار ها بر روی لایه های داخلی را به طریق شیمیایی و یکنواخت اکسید می نمایند. در این مرحله برای اجرای عملیات چینی و پرس، تعداد معینی از انواع Pre-Preg به ابعاد و ضخامت مناسب را اختیار کرده و مابین لایه ها قرار می دهند. برای ایجاد خارجی ترین لایه ها نیز از ورقه نازک مسی (فویل مسی) استفاده می شود. یعنی چنین ساختاری اجرا می گردد :

(لایه اول) فویل مسی / Pre-Preg / لایه داخلی / Pre-Preg / لایه داخلی / Pre-Preg /.../ فویل مسی (لایه آخر)

برای حفظ یکنواختی و عایقی در تمام نقاط لازم است حداقل دو ورق Pre-Preg میان هر دو لایه متوالی استفاده شود. به عنوان یک قاعده سرانگشتی می توان گفت که ضخامت Pre-Preg مورد استفاده باید حداقل دو برابر ضخامت لایه های مسی باشد که قرار است میان آن ها عایق شود. برای حفظ انطباق کامل لایه ها از یک سیستم و روش پین زنی دقیق و مناسب استفاده می شود. محل هر یک از این پین ها بر روی طرح مداری هر یک از لایه های داخلی با مختصات کاملاً یکسان تعیین گردیده است.

پس از لایه چینی برای بکارگیری حرارت و فشار جهت به هم چسبیدن لایه ها از یک دستگاه پرس مکانیکی مجهز به حرارت (و در برخی دستگاه ها مجهز به تخلیه هوا) استفاده می شود. در جریان افزایش دما با شیب معین و کنترل شده، درجه حرارت از دمای گذار شیشه ای T_g رزین بالاتر رفته و رزین شروع به نرم شدن کرده و تمامی فضای خالی میان مدار های مس بری شده را پر میکند و به عنوان یک چسباننده، لایه ها را به هم متصل می نماید. جهت توزیع یکسان حرارت از صفحات فولاد زنگ نزن به ضخامت تقریبی ۶-۸ mm استفاده می شود. طول و عرض این صفحات برای یکنواختی توزیع دما لازم است حدود ۳۰-۴۰ mm بزرگ تر از ابعاد برد و نیز حدود ۵۰-۱۰۰ mm کوچکتر از طبقات دستگاه پرس باشد. برای جداسازی هر دسته از لایه های سازنده یک برد چند لایه از دسته های دیگر نیز از ورق های فولادی مشابه با ضخامت کمتر استفاده می نمایند.

هر یک از بردهای چند لایه پرس شده به منظور امکان پذیری اجرای عملیات سوراخ کاری در نقاط معین و مشخص شده در طرح مداری (خارج از کادر مدار) به عنوان مرجع سوراخ کاری، ابتدا سوراخ شده و سپس پین گذاری می شوند. سوراخ کردن این نقاط مرجع باید با دقت بسیار بالا و به وسیله دستگاه مخصوص انجام پذیرد زیرا هرگونه خطا خارج از حد مجاز منجر به اتصال نابجای لایه ها از طریق سوراخ های متالیزه شده و در نتیجه ضایعات به بار خواهد آمد.

پس از انجام عملیات سوراخ کاری، لازم است به منظور تسهیل متالیزاسیون و نیز افزایش میزان چسبندگی لایه فلزی به جداره سوراخ ها و هم چنین بهبود اتصال جداره فلزین سوراخ های متالیزه با لایه های مسی پد های تعبیه شده در لایه های داخلی داخل هر یک از سوراخ ها با اجرای فرآیندی شیمیایی از هرگونه زواید رزینی و شیشه ای پاک گردد. به این فرآیند Desmearing می گویند.

با اجرای آبکاری و متالیزه کردن جدار سوراخ ها ارتباط الکتریکی میان لایه ها از طریق سوراخ های هادی شده برقرار می گردد و از این پس با برد چند لایه نظیر یک برد دو رو با سوراخ های متالیزه رفتار خواهد شد.

مجموعه کامل مجلات تخصصی برق و الکترونیک

مجموعه کامل مجلات تخصصی برق و الکترونیک ، شامل



مجموعه ۱۳ ساله مجلات EPE

مجموعه ۱۷ ساله مجلات elektor

مجموعه ۲۴ ساله مجلات Circuit Cellar

مجموعه ۷ ساله مجلات Servo Magazine

مجموعه ۸ ساله مجلات Nuts and Volts

توضیحات تکمیلی :

<http://eshop.eca.ir/link/paper.php>



ربات شن نورد

نسل جدیدی از ربات های درازپا، جهان صنعتی را متحول خواهند کرد

مترجم: سارا رشیدی s.rashidi.82@gmail.com

مارمولک دم راه راه بر روی ذرات و دانه های نازک شیشه می ایستد و وزنش را جابجا می کند. دانه ها، در زیر پایش می لغزند و موجود لکه دار، برای نگه داشتن بهتر خود، انگشتان دوکی شکل خود را به حالت کشیده روی شیشه قرار میدهد. ناگهان شروع به دویدن کرده، تمام سطوح ذرات را با چابکی شگفت-انگیزی طی می کند و با ضربه زدن به دانه ها، انگشتان خود را می کشد و پاهای خود را به صورت شلاقی به جلو و عقب می اندازد. هر طرف از بدن مارمولک کش می آید و سپس به دور خود حلقه می زند شبیه خرنده ای که با حرکت سریعش، چندین متر در هر ثانیه به پیش می رود.

در گودبرداری یک سال گذشته در صحرای موهوای کالیفرنیا مارمولکی توجه ما را به خاطر آن پاهای واقعا عجیب و غریبش جلب کرد و به یک آزمایشگاه در جورجیا تک منتقل شد، دست و پای بلند و استخوانی به خرنده اجازه می دهد تا بر روی شن و ماسه، سنگ و انواع دیگری از زمین که در صحرا وجود دارد، حرکت کند. در آزمایشگاه مقداری از خاک صحرا را که برای مطالعه آورده شده بر روی صفحه ای ریخته و با دمیدن باد در آن خاکی صاف و سفت به دست می آوریم، سپس چگونگی مقابله مارمولک با این شرایط را بررسی می کنیم.

علاقه مندی ما صرفا از نظر بیولوژیکی نیست. کاوشگران ما در مرکز فناوری جورجیا، Koditschek و Komsuoglu در دانشگاه پنسیلوانیا در فیلادلفیا و دیگر همکارانمان امیدوار هستیم که با مطالعه مارمولک دم راه راه و مجموعه ای از حیوانات وحشی و موجودات دیگر بیابان، بتوانیم نسخه سریع تری از ربات شش پا، با نام SandBot بسازیم. در هنگام عبور از اراضی خشک، ربات مسیر ثابت، به اندازه دو برابر طول بدنش را در هر ثانیه می پیماید. (برای مقایسه، یک سگ آموزش دیده به اندازه چهار برابر طول بدنش را در هر ثانیه می پیماید.) اما ربات شن نورد (SandBot) در اولین سفر کوتاه خود، با پاهای هلالی شکل، در سراسر دانه های شیشه ای، با موفقیت کمتری چاله حفر می کند و تنها پس از چند مرحله در آن گیر می کند. مشخص است که خاک از نوع شن و ماسه برای یک ربات یکی از سخت ترین مواردی است که جهت آزمودن بکار می رود.

از جهتی خاک از نوع شن و ماسه ذاتا لغزنده است، به این دلیل پیوسته بین دو حالت جامد بودن و سیال بودن در حال تغییر می باشد. فیزیکدانان هنوز تصویر کاملی از مکانیسم ذرات شن و ماسه در دست ندارند، به همین دلیل توجه ما به مارمولک و استراتژی های هوشمندانه تکامل یافته این نوع خاک، در مقابله با سطوح شنی جلب شده است. برای مثال، ما متوجه شده ایم که انگشتان بلند پای مارمولک در هر مرحله به عمق شن و ماسه فرو می روند. به نظر می رسد که این مورد اجازه می دهد که مارمولک پاهای خود را با فشار آوردن از سطح پایینی خاک بیرون آورد و نیز قسمت پوسته ای پایداری کمتری نسبت به سطوح پایینتر دارد. آشنایی اولیه نشان می دهد که توانایی فرورفتن انگشتان پای مارمولک در داخل زمین، او را قادر می سازد تا زمین سخت را بپیماید و تا ۷۵ درصد از سرعت آن بر روی زمین سخت را حفظ نماید. حیوانات صحرایی با سطوح مختلفی از شن و ماسه روبرو هستند و هر کدام به نحوی با سطح زمین برخورد می کنند و به عنوان مدرکی ارزشمند برای تصحیح ربات شن نورد بکار می روند.

نهایتا، ما می خواهیم رباتی بسازیم که بتواند از روی هر نوع زمین سختی مانند غزال عبور کند و از درختان پوسته پوسته شده بلند و ساختمان ها مانند ستیاب بالا رفته و یا در توده ای لغزنده از برگ و یا گل و لای همانند یک مار مانور دهد. حداقل اکنون تعداد محدودی از ربات ها توانسته اند برای مدت زمان کوتاهی دیوارهای عمودی را بپیمایند و از سطوح شیبدار پوشیده از برگ عبور کنند یا حتی از روی سطوح یخ زده بگذرند. نهایتا، ربات ها با قابلیت جابجایی سریع، توانستند تحولی در یافتن و فرار از مناطق خطرناک در تمامی انواع سطح زمین نه تنها بر روی زمین، که در ماه، مریخ و فراتر از آن را نیز ایجاد کنند.

اول از همه، ماشین آلات ما نیاز دارند که بر سطح زمین از جنس شن و ماسه غلبه کرده و بر آن فائق آیند. اگر نیاز بود بالی برای پرواز یا باله ای برای شنا بسازیم نیازمند بهره گیری از معادلات Navier-Stokes که قوانین تعیین شده ای برای جریان سیال هستند، بودیم. اما برای مواد پیچیده ای مانند شن و ماسه، هنوز مدل معادلی وجود ندارد. بنابراین ما مجبور هستیم از همان ابتدا، با بررسی خواص فیزیکی ذرات تشکیل دهنده زمین شروع کنیم. پس از حدود دو سال مطالعه و آزمایش، در کنفرانس فیزیکدانان، ربات شناسان، و زیست شناسان فکر میکنم ما برخی از قوانین اساسی در توصیف حرکت ربات ها در سراسر سطوح دانه ای زمین را شناسایی کرده ایم. هرچند استفاده از آن دانش جهت طراحی ربات های مخصوص شنزار، هرگز آسان نخواهد بود.

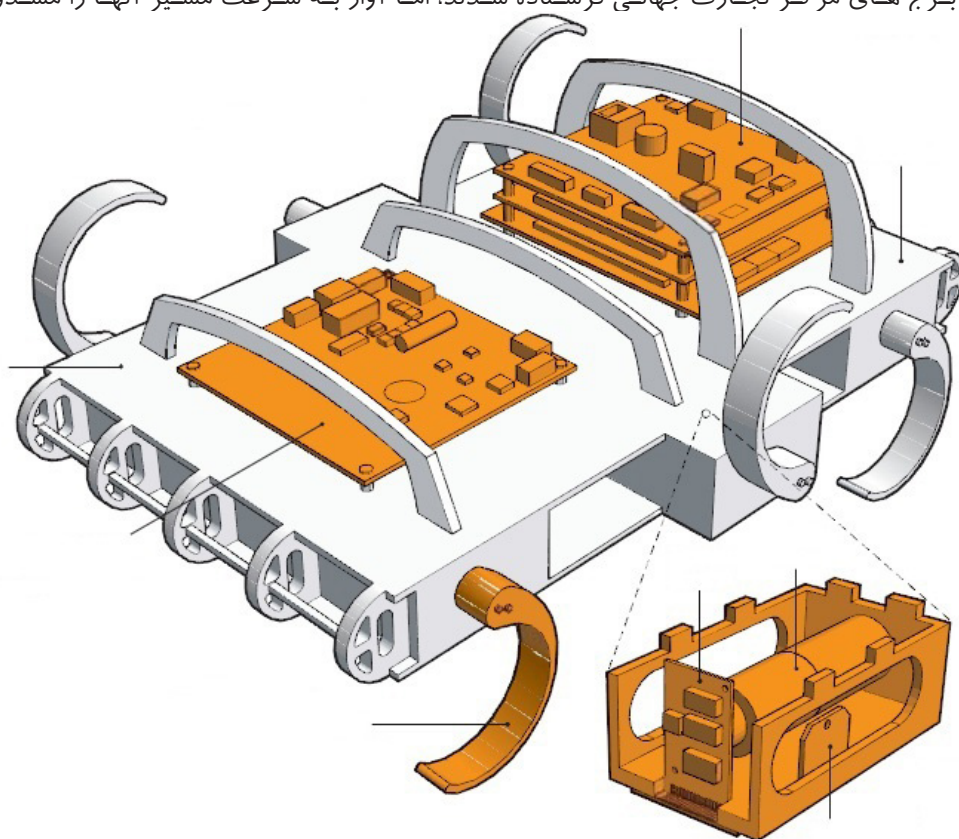
در نظر بگیرید که انسان چگونه خود را بر روی زمین حرکت می دهد. مکان هایی مانند جاده ها و بزرگراه ها که سرمایه گذاری های عظیمی شده، جابجایی با ماشین یا قطار را نسبتا ساده شده است. در واقع، وسایل نقلیه ما به تمامی امکانات هموارکننده مهندسی نیازمند بوده و بدون اینها، آدمی نمی تواند به دوردست ها برود. اما برای وسایل نقلیه و همچنین برای ربات ها، مساحت بزرگی از زمین تا حد زیادی غیر قابل دسترس می باشد. حدود ۳۰ درصد از مساحت بیابان و یک پنجم آن با نوعی از شن و ماسه پوشیده شده است.

تنها مسئله، شن و ماسه نیست، انتظار می رود ربات های متحرک در مناطقی که بلایای طبیعی رخ داده، جبهه های جنگی و مکان هایی که پر از قلوه سنگ و غیرقابل پیش بینی و غیرقابل عبور هستند بیشترین فایده را داشته باشند. به عنوان مثال، در سال ۲۰۰۱، ربات ها پس از سقوط برج های مرکز تجارت جهانی فرستاده شدند، اما آوار به سرعت مسیر آنها را مسدود

کرد و یا باعث درهم شکستن ربات ها شد. به همین شکل، هنگامی که معدن زغال سنگ در ساگو (Sago, W.Va)، در سال ۲۰۰۶ فرو ریخت، یک ربات امداد و نجات، پیش از آنکه در گل و لای گیر کند، در حدود ۷۰۰ متر از ورودی را طی کرد که حتی مسائل بی خطر مانند شن و برگ های افتاده نیز می تواند ربات را متوقف کند.

مارمولک دم راه راهی و خرچنگ دارای پشت، اسرار بسیاری برای حرکت سریع و نحوه دویدن دارند مانند اینکه اندام چه میزان انعطاف بر روی شن و ماسه دارند. پشت خرچنگ فشار اعمال شده توسط پاهای خود را بصورت بی سیم اندازه گیری کرده و ارسال می کند.

به طور خلاصه، ربات هایی که بر روی چرخ ها حرکت می کنند



مسیرهایی را که به محدوده عملکردشان نزدیک است را طی می کنند. ربات های دارای پا، که از حرکات حشرات و یا حیوانات و خزندگان تقلید می نمایند پیشنهاد مناسبی هستند، اما مکانیسم حرکتی آنها ساده نیست. چون شناخت زیادی از نیرویی که بین پا و سطح زمین ایجاد می گردد تا حرکت صورت گیرد، وجود ندارد. راه حل متداول روش آزمون و خطاست. اما ما راه درازی را سپری کرده ایم. قدمت اولین ربات کامپیوتری کنترلی دارای پا به صده ۱۹۶۰ برمی گردد، زمانی که اولین بار رابرت مک گی اسب ساختگی خود را در دانشگاه کالیفرنیا جنوبی، واقع در لس-آنجلس خلق کرد. پس از آن، مک گی این پروژه را در دانشگاه ایالتی اوهایو در کلمبوس، دنبال کرد که اولین ربات مستقل دارای پا را در سال ۱۹۷۶ ساخت. این دستگاه، که به عنوان جانور شش پا شناخته شد، می توانست مسیر خود را در سراسر بلوک های چوبی خانگی را به آرامی طی می کرد.

بعد از یک دهه، مک گی و همکارانش سیستم تعلیق تطبیقی خودرو با طول ۵ متر که اولین دستگاه مستقل دارای پا بود را برای مقابله با محیط های خارجی ایجاد کردند. ربات هر پا را به دقت بر زمین قرارداد، در هر ثانیه کسری از طول بدن را به آرامی طی کرده و سپس با وجود گشتاور مفاصل، نیروهای واکنشی لازم بین زمین و بدن خود را ایجاد کرده، به جلو هل می داد.

مرحله بعدی ربات های پادار، که توسط مارک رابرت در دانشگاه کارنگ ملون در پیتسبورگ و بعدها در دانشگاه MIT بود راهنمایی شد، ماشین هایی با مهارت های پویایی ساخته شدند. مهارت پویا توانایی تبادل انرژی پتانسیل و انرژی جنبشی به شیوه ای کنترل شده و تفاوت بین یک کانگوروی جهنده و ماشین است. خم پاهای کانگورو انرژی بالقوه را ذخیره می کند، که اجازه می دهد به آسانی از موانع عبور کند. توانایی هدایت انرژی مکانیکی بدن جهت حرکت بر روی سطوح غیرقابل پیش بینی برای ربات ها بسیار مهم است. تولیدات رابرت، اساسا متحرک هستند و برای حفظ تعادل در جهش و زمان جفت شدن با یکدیگر استفاده می شوند. این ربات ها هنوز هم رکورد سرعت حرکت روی زمین با ۲۱ کیلومتر در ساعت حفظ کرده اند، با این حال آنها سرسختانه برای محیط آزمایشگاهی کنترل شده طراحی شده اند.

ربات RHex، طراحی شده توسط مارتین بهلر، ربات شناس و عضو گروه کودیتچیک Koditschek در سال ۱۹۹۹، ربات ها را تا مرحله بعد ارتقا دادند. این دستگاه خودمختار، در بخش زیست شناسی رابرت کامل، از دانشگاه کالیفرنیا، برکلی، دارای شش پا هستند که از قسمت خارجی بدن به مرکز متصل شده است. این پیکربندی ربات به ثبات و پایداری زیادی نیاز دارد، موارد سریعتری نیز به وجود آمده اند که RHex تنها ماشین دارای پا است که می تواند در مسیرهای ناهموار و زمین های شکسته با سرعتی بالا و به اندازه طی کردن طول بدن در هر ثانیه به حرکت خود ادامه دهد.

ربات RHex به نوبه خود، مدلی برای خانواده ی ربات ها است که هر کدام توسط یک موتور واقع در مفصل میانی، به حرکت درمی آیند. ربات آبی که فرزندان خود را در میان دیگران قرار می دهد و اساس ربات RHex شناگر با باله است که به دو ربات دیواره نورد با نام دیناکلیمر Dynoclimber و ربات شنزار مجهز شده است.

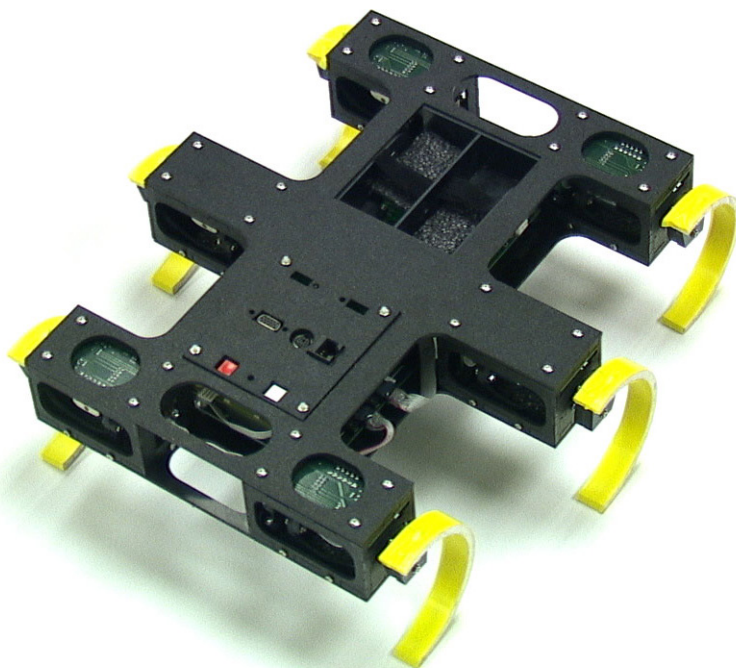
دونده سنگپوش

SandBot یک مسیر پر از دانه های خاک رس را طی می کند که جهت آماده سازی ربات برای انواع بسیاری از گرد و غبار، شن و ماسه و خاک سست است که در خارج از آزمایشگاه با آنها روبرو می شود.

در اوایل سال ۲۰۰۷، Komsuoglu ربات شن نورد را با استفاده از مدل RHex و زیرساخت های مدولار براساس خلاقیت خود طراحی و در کمتر از یک ماه ساخت.

با ۲ کیلوگرم، وزن آن کمتر از یک چهارم وزن RHex می باشد. به مانند SandBot، RHex دارای شش پای سازگار و به طور مجزا کنترل شده که هر یک از آنها یک نوار نیم دایره از جنس پلاستیک دارند. همچنین مانند RHex، البته با الهام از حشرات ها با "سه پایه متناوب" راه می رود. این ربات با ۳ پا که دارد حرکت خود را انجام می دهد و با پای جلو و عقب حرکت رو به یک به سمت و با پای میانی حرکت مخالف جهت بقیه را انجام می دهد. پس از هر مرحله، پایه های خود را جایگزین کرده و بدن را به جلو رانده، پس از آن به اطراف می چرخد.

داخل ربات شن نورد از نقاط مجزائی تشکیل شده است و از طریق یک شبکه همزمان که RiSEBus نامیده می شود، ارتباط دارند که مشابه نسخه اولیه ربات کوهنورد است. در مفصل ران هر یک از پاها ۱۱ وات از طریق موتور DC ارسال می شود. شش موتور کنترلی

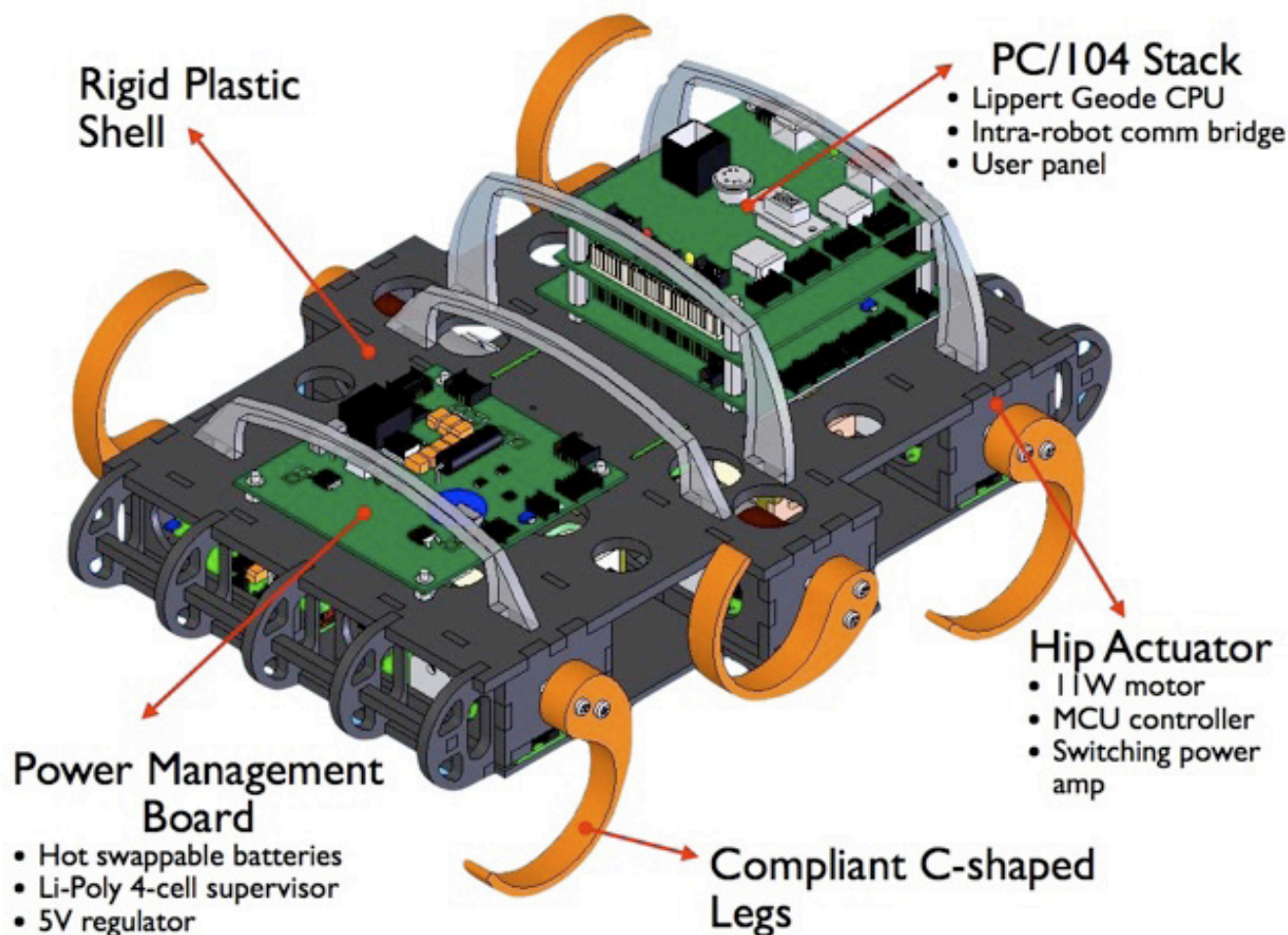


به یک کامپیوتر مرکزی مرتبط شده اند که توانایی از مغز ربات شن نورد هستند و در تصمیم گیری های رفتاری سطح بالا تمرکز دارند. دستورات از پردازنده مرکزی به کنترل کننده های موتور آموزش داده می شوند تا پاها را با سرعت و در جهت مورد نظر حرکت دهند. کنترل ردیابی موقعیت اختلاف میان حالت واقعی پا و حالت مورد نظر را تعیین می کند. سپس کنترل کننده ولتاژ مورد نیاز برای اصلاح خطا را محاسبه و صدق آن را به موتور با استفاده از تقویت کننده قدرت است. این عمل باعث می گردد پاها با سرعت مناسب به حرکت در مسیر خود بپردازند.

برای صرفه جویی در قدرت محاسباتی ربات، کامپیوتر دستورات را با سرعت نسبتاً کندی، حدود ۱۰۰ بار در ثانیه ارسال می کند، که زمان خود را برای انجام دادن کارهای دیگر آزاد می کند. به عنوان مثال، ممکن است پردازنده ی مرکزی اعمال گردد. به یکی از میکروکنترلرها فرمان می دهد که یکی از پاهای خود را باید با سرعت خاصی و با شروع از یک موقعیت خاص حرکت بدهد. از آن پس، همه ردیابی های مربوط به موقعیت آن پا توسط میکروکنترلر صورت می گیرد، که می تواند سنسورهای خود را در فرکانس بسیار بالاتری از ۱ کیلوهرتز بازیابی و بدست آورد.

این طرح اجازه می دهد تا پردازنده مرکزی، با استفاده از بسته های اطلاعاتی فشرده شده که نیاز به حداقل قدرت محاسباتی برای رمزگشایی دارند، با ساق پا ارتباط برقرار کنند.

تفکیک وظایف کنترلی، پردازنده مرکزی را از اجرای برنامه های طولانی مدت رها می کند. پردازنده مرکزی ممکن است از یک دوربین برای ارزیابی تفاوت بین ارتباط خود با وقایع بصیری و آنچه باید باشد استفاده کند، و یا چگونه باید باشد و نیز فهمیدن اینکه چگونه سطحی که بر مبنای بازدهی وابسته است، خطرناک بوده که از طریق سنسورهای در پایه اصلاح می شود.



طراحی ربات شن نورد، در آزمایش هایی با وجود موجودات طبیعی شنزار در آزمایشگاه Goldman انجام شده است. برای مثال، مارمولک دم راه بیش از هر نوع موجود دیگری می تواند با سرعت بالایی از زمین شن و ماسه عبور کند. در مقابل خرچنگ تطبیق پذیری کمتری دارد. در توده ای از زمین، اندام ها و پاهای خود را ۴ سانتی متر از پوسته ی بدنش دراز می کند و با سرعت ۱ متر در ثانیه سریع حرکت می کند اما در برخی خاک های ضعیف خرچنگ در خاک گرفتار می شود. عقرب می تواند در هر ثانیه چند برابر طول بدنش را در روی زمین طی کند و در مسیرهای شیب دار به راحتی به طرف پایین فرود می آید.

مشاهدات ما از مارمولک، خرچنگ و عقرب در شرایط مختلف، به شکل دهی نظریه ما در مورد حرکت در شن و ماسه کمک کرده اند. ما معتقدیم که این پروژه اولین تلاش ها برای ترکیب اندازه گیری مستقیم جریان فیزیکی زیرلایه ها با مشاهده تأییراتی که دوندۀ بر روی زمین وارد کرده و حرکات بدن آنها دارد را نشان می دهد. به طور کلی، وزن حیوان، شکل پا، و

نحوه راه رفتن، همه و همه مقدار فشار خاصی که به شن و ماسه وارد می شود را تعیین می کنند. براساس این مدل، مارمولک به مشخصات قدرت مواد موجود در شن پیدا می کند. تا از بین مواد لغزیده و دست و پا بزنند، کاریکه خرچنگ شیخ بر روی زمین نرم به پایان رسانده است.

هر گام حرکتی خزننده، می تواند بسیار آموزنده باشد. با هرگام بلند، زمانیکه یک پا در بین شن و ماسه حرکت می کند نیروهای کششی تولید شده می توانند خواص مواد جامد و مایع را نشان دهند. اگر تنش-های ایجاد شده توسط پا بیش از یک حد آستانه ای باشد، مواد جریان می یابند و می لغزند. اما در صورتی که این تنشها کاهش یابند می تواند به اندازه کافی سفت شود. برای مثال، اگر نیروهای رو به پایین تولید شده توسط اندام و وزن ربات با فشار داخلی شن و ماسه که تابعی از عمق آن است متعادل شود، این مورد رخ می دهد.

حقیقت دیگر این است که رفتار حرکتی ذرات شن و ماسه، بستگی به آن دارد که در گذشته چه اتفاقاتی بر آن افتاده است. برای مثال بخشی از شن و ماسه سخت توسط یک گام دچار آشفته‌گی شده و ممکن است پراکنده شود و زمانیکه گام بعدی به مواد ضربه می زند، نیروهای تولید شده توسط یک گام در شرایط مختلف می تواند به طور چشمگیری متفاوت باشد، بطور مثال مقاومت نفوذ با ضریب ۱,۶ بین مواد محکم متفاوت است. پیش بینی اینکه تا چه حد، در حالت های مختلف قسمت های بدن باید در خاک نفوذ کنند، کار پیچیده ای است.

برای یافتن اینکه چگونه ربات شن نورد بهترین مانور را در شن و ماسه می تواند اجرا کند، ما ربات را در معرض انواع مختلفی از محیط ذرات و دانه ها که کاملاً کنترل شده اند قرار دادیم. محیط را با استفاده از یک مسیر ۲,۵ مترساخته شده توسط Chen Li، دانشجوی فارغ التحصیل از آزمایشگاه Goldman آتلانتا است، کنترل کردیم. به نظرمی رسید مسیرشبه وان حمام طولانی مرتب شده که آن را با ۹۰ کیلوگرم دانه خاک رس پر کرده اند. حفره های کوچکی در پایین سطح مسیر وجود داشت که از طریق آن ما می توانستیم هوا را به داخل وارد کنیم که باعث بلند کردن دانه های خاک رس و تحریک آن قبل از استقرار درحالت بسته پراکنده شده وجود دارد. چرا دانه های خاک رس و شن و ماسه واقعی نه؟ ما دریافتیم که هر ذره به اندازه کافی بزرگ است تا ما از ورود آنها به موتور نگران نباشیم و در عین حال نور به اندازه کافی با پمپ کردن هوا به محفظه وجود داشته باشد. از آزمایش های جداگانه، دریافتیم که تا زمانی که از تکه ها و دانه های سخت ساخته شده اند، شبیه بودن مواد مهم نیست.

توسط شن و ماسه و سایر ذرات ما می توانیم قدرت زمین را در تحمل حجمی از مواد جامد محاسبه کنیم که کسری از نیرویی است که توسط ذرات ریز بر روی زمین اعمال می گردد.

به طور معمول، کسر حجمی جامد بین ۵۸ تا ۶۴ درصد برای موادی مانند شن و ماسه و یا انبوهی از دانه ها می افتد. کسری پایین تر به این معنی است که به طور متوسط، امتیاز کمتری از تماس بین دانه ها و مواد جامد کمتر است وجود دارد. از آنجائی که ربات RHex در راه رفتن بر روی سطوح مختلف موفق بوده است، به این نتیجه رسیدیم که ربات هر چه کوچکتر ولی پر قدرت تر باشد حرکت بهتر و آسانتری روی شن و ماسه خواهد داشت. ما اشتباه کردیم. در آزمایش های اولیه، مواد بسته بندی شده که ۶۳ درصد از کسر حجمی جامد بود را بر روی سطح قرار داد، و فرکانس از راه رفتن سه پایه متناوب تا ۵ دور در هر ثانیه مشخص کردیم. پیش از این، این ربات با استفاده از این پارامترها در سراسر زمین سفت و سخت بی وقفه پیش رفته بود.



هر چند، در این زمان، تنها پس از طی چند مرحله در خاک گیر کرد. مانند لاستیک های ماشینی که در گل و لای گیر افتاده، ربات ساخته شده با پاهای چرخشی سریع، مطلقاً هیچ حرکت رو به جلو در مسیر پر از خاک رس نداشت. اولین فرض ما به شکل ناامیدکننده ای بود چرا که ربات شن نورد، برای راه رفتن بر روی شن و ماسه خیلی سنگین بود و ما مجبور به طراحی دوباره ربات شدیم.

اما ما تصمیم گرفتیم کمی بیشتر روی آن کار کنیم. Komuosglu مشاور تلفنی، از دفتر خود واقع در Penn، پیشنهاد کرد که ما جهت حرکت اندام ها و راه رفتن ربات را، اندکی تغییر دهیم تا در بخش هایی از چرخه، پاها سریعتر بچرخند و در بخش های دیگر کندتر باشند. او از مطالعات قبلی می دانست که برخی از ربات ها با چنین راه رفتن متفاوتی، حداقل در سطوح سخت و محکم اجرای بهتری دارند. به نظر می-رسید که ارزش یک حرکت را دارد. Komuosglu از طریق تلفنی میزان تغییرات را به ما اعلام کرد، ما آنها را وارد برنامه کنترلی کردیم، مثل سحر و جادو بود، ربات شروع به حرکت کرد! ربات پاهای خود را پنج بار در ثانیه به شکل دورانی می چرخاند، اما هنوز حرکت سریع به طرف پائین مسیر، تنها به میزان درازای یک طول بدن در هر ثانیه بود. مطالعات بیشتر نشان داد که هر یک از اندام ها در ذرات ریز نفوذ می کنند و تا وقتی که وزن ربات را برای ثبات کافی در فشار بالا و رو به جلوی ماشین تامین کند، به حرکت خود ادامه میدهد. همراه با Paul Umbanhowar، مهندس مکانیک در دانشگاه Northwestern: بعد از آنکه ما در توضیح رابطه بین حجم و جسم، فرکانس چرخش اندام و عمق نفوذ اندام در هر مرحله، مدل حرکتی را بسط دادیم. در هر دو مدل و شواهد تجربی نشان می داد اگر فرکانس چرخش اندام ربات را افزایش دهیم، ربات بیشتر در مواد غرق می شوند و اندازه هر گام کاهش می یابد، باعث از دست دادن فاجعه بار سرعت می شود که کاملاً مخالف با آنچه که بر روی زمین سخت رخ می دهد. پیشرفتی دیگر: بهبود دیگری که ما برای ساخت پای بهتر برای ربات شن نورد در حال کار بر روی آن هستیم توان بخشیدن به آن برای گرفتن شن و ماسه است تا فقط مثل مارمولک دم راه راه عمل کند. به این منظور، نیروهایی که بر پا تاثیر می گذارند تا در مواد با حجم های مختلف نفوذ کنند را اندازه گیری می-کنیم. آزمایش ها بصورت فریبده ای ساده به نظر می رسند: ما شتاب سنجی را داخل اشیای دیسک مانند ساده ای جاسازی می کنیم و بعد آنها را در انبوهی از شن و ماسه قرار می دهیم.

نتایج نشان می دهند زمانیکه پا به زمین برخورد می کند نیروهای تولید شده در حجم زیاد و یا در قسمت های کوچکی از مواد، کیفیت های مختلفی دارند. وقتی پای طراحی شده در یک ماده با حجم کم می افتد، نیروی وارده تا زمانی که جسم به ثابت بایستد، افزایش می یابد و هنگامی که شی بر روی یک ماده با حجم بالا بسته بندی شده می افتد، در زمان نفوذ نیرو کاهش می یابد. برای بررسی نیروهای کششی و بلند کردن که در بخش های دیگر از هر مرحله بوجود می آیند، از حرکت پا و انگشتان پا در امتداد مسیرهای دانه، از یک بازوی رباتیک استفاده می کنیم.

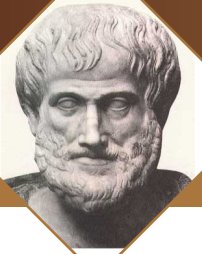
با وجود مدلی کامل از رفتار منحصربه فرد حرکتی دانه ها، ما باید شبیه سازی را دوباره دسته بندی کنیم. Yang Ding دانشجوی فارغ التحصیل از مرکز Goldman، شبیه سازی های کامپیوتری را که مدلی از برخورد اشیاء با شن و ماسه، دانه ها و سایر ذرات پوششی می باشد را توسعه داده است. امیدواریم که این آزمایش ها و شبیه سازی ها در مورد پا، در نهایت به توسعه سنجش های جدید و سیستم کنترلی منجر گردد تا برای ربات شن نورد جستجوگر غذا را قادر سازد و حرکات پا را با تغییرات زمین انطباق دهد. شن و ماسه تنها نوع محیط زیست و سطوح زمینی نیست بلکه گل و لای و بستر سست برگها نیز نمایش جامد و مایعی از ویژگی های سطحی دانه های هستند که این ربات ها می توانند با آن مقابله کنند. مشاهدات ما از مارمولک، خرچنگ و عقرب در حقیقت به شکل دهی نظریه ما در مورد حرکت بر روی شن و ماسه با مدل فیزیکی که از روی پاها و مغز آنها ساخته شده است، کمک کرده اند، روزی ربات ها باید قادر باشند در تمام محیط سنگی یا شنی پیش بروند و یاد بگیرند چگونه قدم به قدم با تغییرات روی زمین، رفتار کنند.

جهت جستجو بر روی سطح سیارات دیگر می توانیم هزاران ربات پادار را تصور کنیم که به سادگی از توده های انبوهی از سنگ و زمین های شنی بالابروند که هنوز چالشی بزرگ، برای ماشین آلات امروزی است. اما این مسئله، چیزی است که لاک پشت دریایی نیز میتواند انجام دهد، این حیوانات شناخته شده، باید از یک سوراخ در عمق زمین، بالا بیایند و از بین علفزارها و باقی مانده های زباله های اطراف با دست و پا زدن به سختی عبور کرده و از روی شنزارها حرکت کرده و ماسه ها را برای رسیدن به آب، کنار زده، به مکانی می رسند که در آن، بقیه عمر خود را می گذرانند.

ما همچنین برای الهام گرفتن، زیر زمین را نیز جستجو می کنیم، اکنون در حال مطالعه مارمولکی به نام sandfish (ماهی شنزار) با استفاده از سرعت بالای اشعه X هستیم که می تواند شن و ماسه را در یک چشم بصری به هم زدن به عقب بزند و سپس از بین مواد زیرزمینی به جلو حرکت کند. البته ما امیدواریم این موجودات به عنوان سرنخی جهت اینکه یک ربات چگونه با دست و پا زدن می تواند از بین یک منطقه فاجعه زده غیر قابل پیش بینی بعد از یک زلزله یا سیل و یا حفاری رو به پایین برای شناسایی معادن زیرزمینی، فراهم کند. با وجود طبیعت به عنوان راهنمایمان، ما انتظار داریم که این ربات ها به زودی در برخی از توانایی های باورنکردنی جدید مسلط شوند.

<http://spectrum.ieee.org/robotics/robotics-software/march-of-the-sandbots>

منبع :



جان لاجی برد



اختراع تلویزیون به وسیله ی تنها یک نفر انجام نگرفت ولی آن کس که این اختراع را کامل کردو قابل استفاده کرد شخصی به نام جان لاجی برد بود.

جان لاجی برد روز سیزدهم اوت ۱۸۸۸ در هلنسبرگ واقع در نزدیکی کلاسگو به دنیا آمد. پدر او یک کشیش تحصیل کرده بود ولی در فقر به سر می برد. لاجی دی هنگام تحصیل در مدرسه به حدی از خود هوش نشان داد که حتی در ۱۲ سالگی به کمک همکلاسانش یک خط تلفن به وجود آورد.

او پس از پایان تحصیلات متوسطه در دانشگاه گلاسکو مشغول تحصیل گردید و علی رغم آنکه از سلامتی برخوردار نبود کمک مهندس شد و در ۱۶ سالگی با حقوق ۳۰ شیلینگ در هفته در یک شرکت الکترونیک مشغول کار گردید.

پس پایابن جنگ جهانی اول او شروع به تولید جوراب نمود و از این راه حدود ۱۶۰۰۰ پوند تحصیل می کرد. بعد از آن او کارگاه کوچکی برای تولید مربا و سس افتتاح کرد ولی به علت آنکه این اقلام زیاد مصرف نداشت کارگاه خود را تعطیل کرد.

در سال ۱۹۲۳ در موقعی که از عمرش ۳۴ سال می گذشت او به لندن مراجعت کرد و تصمیم گرفت به هر قیمتی که شده اختراع تلویزیون خود را تکمیل کند. برای این منظور او یک قفسه

قدیمی چای را که یک جعبه ی مقوایی مخصوص بسته بندی کلاه در آن بود تهیه کرد و یک دیسک مدور در آن ایجاد نمود. سپس از یک سمساری یک موتور الکتریکی خرید و دیسک مدور را که جندین سوراخ در آن ایجاد کرده بود در آن نصب کرد. وقتی که دیسک را به گردش در آورد اشیایی که به آن نور می تابیدند به صورت نقطه های نورانی ظاهر گشت و انواع و اقسام وسایل دیگر از قبیل لامپ پرتو افکنی و سلول سلینیوم و لامپ نئون و سوپاپ های رادیو و.... را به کار گرفت و برای دریافت انتقال علائم به تعدیلات زیادی دست زد. سرانجام تلاش های مستمر او در بهار ۱۹۲۴ ثمر داد و او توانست سایه صلیب مالتیس را تا فاصله سه یارد منعکس کند.

متأسفانه در این زمان برد، پول کافی برای ادامه دادن به آزمایش های خود را در اختیار نداشت. برای دریافت کمک در روزنامه اعلانی چاپ کرد. در نتیجه ی این اعلانات او تانست مقداری پول فراهم آورد. در سال ۱۹۲۵ پسر یک مغازه دار به نام گوردون سلفریتس به ملاقاتش آمد و مقداری پول به او داد. برد تلویزیون خود را برای او به کار انداخت.

تصاویر درهم و برهم به روی پرده ظاهر گشت.

در دوم اکتبر ۱۹۲۵ برد برای انتقال تصاویر روی پرده دستگاه دیگری ساخت. وقتی که آپارات را به کار انداخت از نتیجه ای که حاصل شد تعجب کرد زیرا حالا دیگر تمامی تصاویر به طور کامل و بدون نقص به روی پرده ظاهر گشتند. در ژانویه ۱۹۲۵ برد با موفقیت نخستین نمایش تلویزیونی خود را نشان داد.

برد تحقیقات مربوط به تلویزیون را در سراسر زندگی خود ادامه داد و علی رغم آنکه بیمار بود و از بی پولی رنج می برد او تلاش خود را روی تلویزیون رنگی خود حتی تا سال ۱۹۴۵ ادامه داد.

مخترع فلک زده سر انجام بی این که از نظر مادی از تلاش های مستمر خود سودی ببرد در سال ۱۹۴۶ اثر آنفولانزای شدید در گذشت. اگر چه بعد از مرگ او تجلیل های زیادی به عمل آمد ولی چه فایده که او در طول عمر خود در فقر زیست و تا زنده بود کسی به کار مهمی که انجام داده بود اعتنایی نکرد.



**فروشگاه تخصصی
برق و الکترونیک**









www.eShop.ECA.ir

پذیرش نمایندگی فعال
از تمامی فروشگاه های برق و الکترونیک سراسر کشور

**فروشگاه تخصصی
برق و الکترونیک ECA**

www.eShop.ECA.ir

مرکز دانلود ECA

« مقالات علمی »
« کنفرانس های داخلی »
« سوالات کنکور »
« پروژه های تکمیل شده »
« و... »

www.Download.ECA.ir

**طراحی، تولید و واردات کلیه بردها و قطعات
مربوط به سیستم های Embedded**

**ARM7/ARM9/ARM11
(Atmel, Philips, Cortex-M3/
A9/A15/...)**

www.eShop.ECA.ir



وب سایت تخصصی برق و الکترونیک (ECA) افتخار دارد با بیش از ۱۳۰۰۰۰ عضو، یکی از بزرگترین وب سایت های تخصصی ایران را تشکیل داده و پاسخگوی نیاز دانشجویان، متخصصان، محققان و صنعتگران عرصه برق و الکترونیک کشور باشد. بی شک یکی از اهداف این انجمن، برقراری ارتباط بین صنعت و جامعه می باشد. لذا از تمام شرکت ها، کارخانجات و موسسات صنعتی علمی آموزشی دولتی و خصوصی، تقاضا مندیم در صورت تمایل به عقد قرارداد تبلیغاتی و یا قبول اسپانسر برای مجله تخصصی نوپز از طرق زیر با ما در ارتباط باشند.

تلفن : ۰۴۱۱-۵۵۳۹۷۸۵
فکس : ۰۴۱۱-۵۵۳۹۷۶۹
ایمیل : adver.eca@gmail.com

Silicon Nanowire Field Effect Transistor

Kianoosh Salek Soltani ^a, Siavash Salek Soltani ^b

^a Department of Electrical Engineering, Islamic Azad University South Tehran Branch, Tehran, Iran

^b Department of Chemistry, Malek-Ashtar University of Technology, Isfahan, Iran

Abstract

Silicon nanowire field-effect transistors (SiNW-FETs) have recently drawn tremendous attention as a promising tool in biosensor design because of their ultrasensitivity, selectivity, and label-free and real-time detection capabilities. For practical uses, SiNW-FETs can be delicately designed to be a reusable device via a reversible surface functionalization method. In the fields of biological research, SiNW-FETs are employed in the detections of proteins, DNA sequences, small molecules, cancer biomarkers, and viruses. The methods by which the SiNW-FET devices were integrated with these representative examples and advanced to virus infection diagnosis or early cancer detection will be discussed. In addition, the utilization of SiNW-FETs in recording the physiological responses from cells or tissues will also be reviewed.

Keywords: Field effect transistor; Silicon nanowire; Biosensor

1. Introduction

Sensing using a planar field effect transistor (FET) has been demonstrated for decades, but lower sensitivity has limited its applications. Compared to the surface region of a planar device, due to the ultrahigh surface-to-volume ratio, nanowires and carbon nanotubes are ideal choices for sensors due to the introduced depletion/accumulation of charges near the surface as a result of surface binding/adsorption of foreign molecules and species. Biological specificity, combined with the semiconducting properties, enables nanocable-based biosensors to selectively detect proteins in solution by using direct electronic readout without the need for labeling [1].

Quantification and analysis of biological processes are of utmost importance for biomedical applications and cellular programming investigation. However, it is challenging to convert the biological

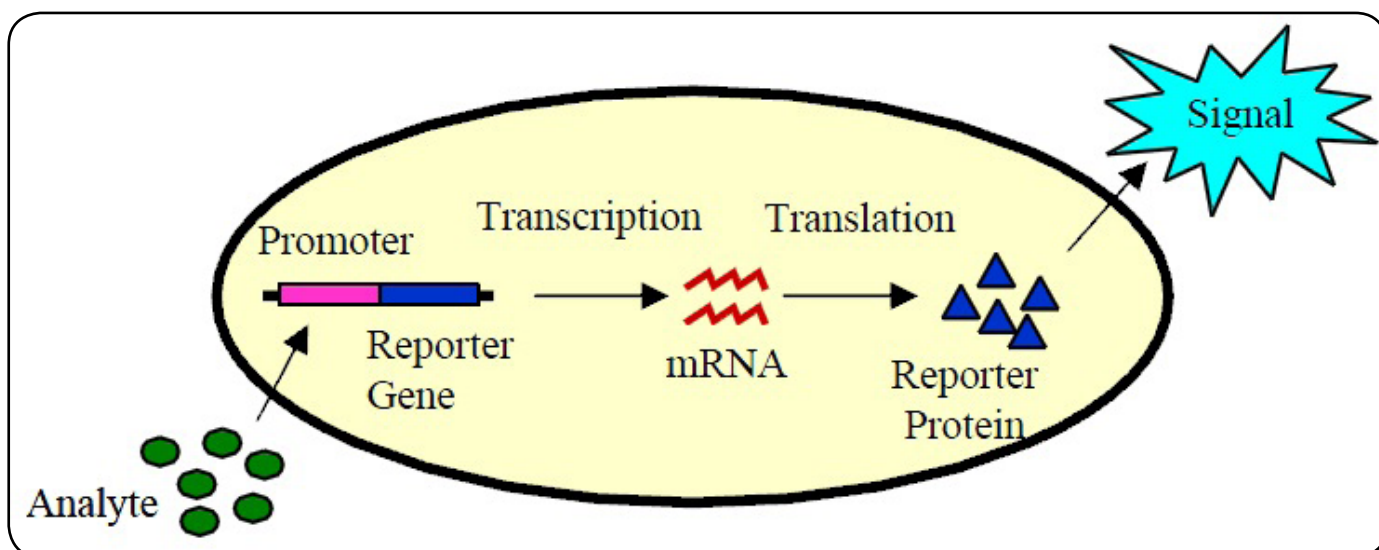


Figure 1. Anatomy of a bioreporter organism. Upon exposure to a specific analyte, the promoter/reporter gene complex is transcribed into messenger RNA (mRNA) and then translated into a reporter protein that is ultimately responsible for generating a signal [4].

information into an electronic signal due to the difficulties of connecting an electronic device into a biological environment. In recent years, there has been dramatic development of electrochemical biosensors because of their applications in toxicity testing, chemical analysis, medical diagnosis, food industry, environmental monitoring, and many other areas. An electrochemical biosensor, as defined by IUPAC, is a self-contained integrated device that allows for specific analytical detection by using a biological recognition element (a biochemical receptor) in direct spatial contact with a transduction element [2]. A biosensor is composed of two elements: a biological recognition unit able to interact specifically with a target, and a transducer able to convert a change in property of the solution or surface, due to complex formation, into a recordable signal. In contrast with conventional bioassays, biosensors allow the detection of molecular interactions as they take place, without requiring auxiliary procedures, making them highly attractive for biotechnological applications [3]. Therefore, the presence of a signal indicates that the bioreporter has sensed a particular target agent in its environment.

Molecular recognition is central to biosensing. Since the first biosensor was developed by Updike and Hicks (1967) many biosensors have been studied and developed. As shown in Figure 2, a biosensor can be defined as a compact analytical device or unit incorporating a biological or biologically derived sensitive recognition element integrated or associated with a physio-chemical transducer. Initially, biosensor recognition elements were isolated from living systems. However, many biosensor recognition elements now available are not naturally occurring but have been synthesized in the laboratory [5].

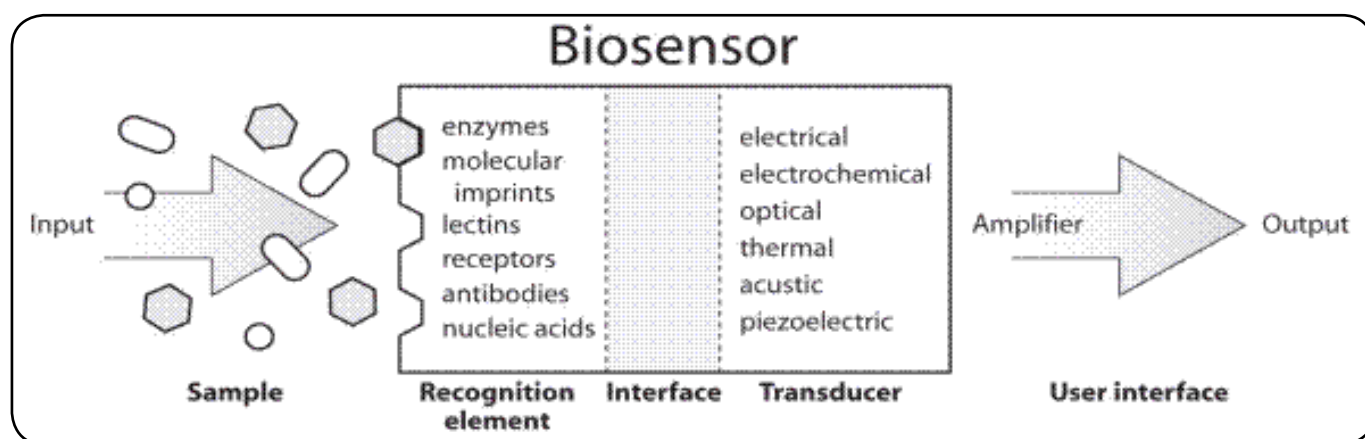


Figure 2. Configuration of a biosensor showing biorecognition, interface, and transduction elements.

Figure 3 shows the illustration of a nanoscale FET biosensor with a cross-sectional view. The semiconductor channel is placed between the source and drain electrodes with a gate electrode on the bottom to modulate the conductivity of the semiconductor channel. Target molecules can be recognized by the receptor modified on the channel surface through strong binding affinity. In recent years, one-dimensional semiconducting nanomaterials, such as silicon nanowires and carbon nanotubes, configured with FETs have attracted great attention because they are an ideal biosensor with high selectivity and sensitivity, real-time response, and label-free detection capabilities [6].

2. FET based biosensors

From the electrochemical point of view, SiNW-FET-based biosensors are a three-electrode system, including source, drain, and gate electrodes. The function of the source and drain electrodes is to bridge the semiconductor channel made of SiNWs and the gate electrode is responsible for modulating the channel conductance.

A diversity of FET-based biosensors has been employed for biological applications. Enzyme-modified FETs comprise a redox active enzyme integrated with an electronic circuitry to give a real-time quantitative analysis of the enzyme substrate, e.g., sensing glucose from a catalytic reaction in the presence of glucose oxidase. Cell-based FETs were exploited to detect the released biochemical agents or real-time cellular responses from living cells, such as action potentials from neuron cells. In general, immunologically functionalized FETs are the most frequently used biosensors. For ex-

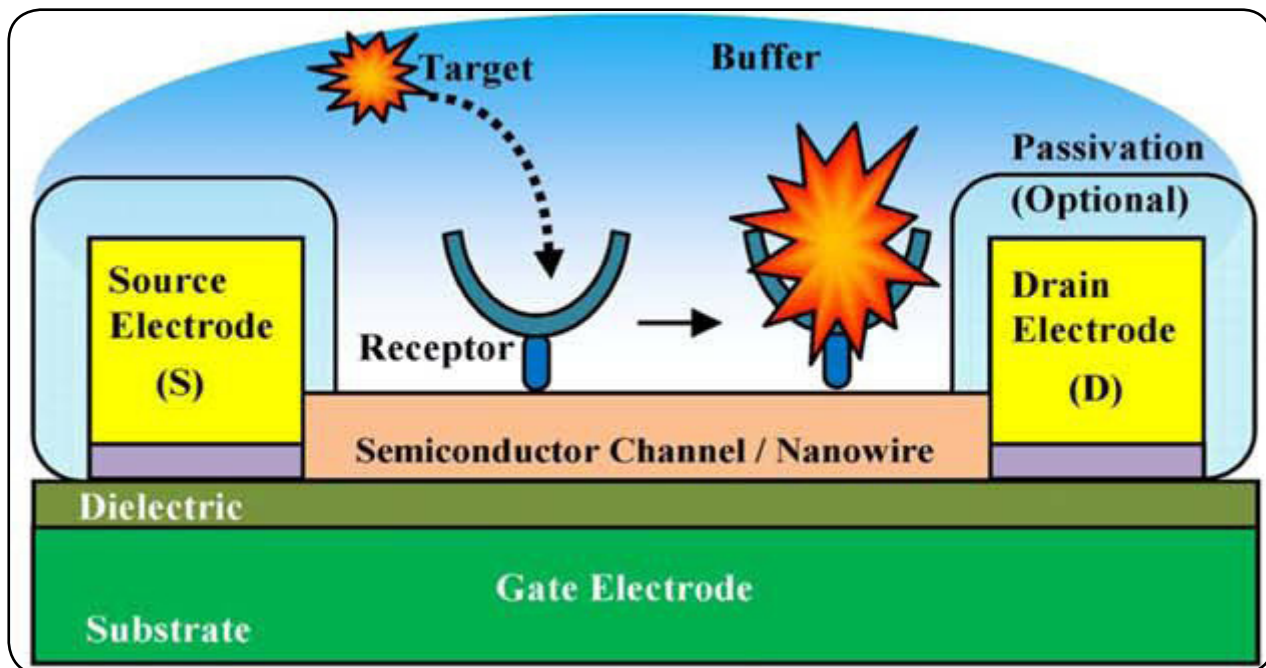


Figure 3. The illustration of a nanoscale FET biosensor.

ample, an antibodymodified FET sensor can be used to detect the corresponding antigen [7].

3. Overview of SiNW-FETs

Taking advantage of the well-developed silicon industry, SiNW-FETs can benefit from existing and mature silicon industry processing techniques and fabrications. In the synthetic reactions that prepare SiNWs, different sizes, shapes, and dopants of SiNWs could be precisely tailored [8]. Because SiNWs could be well-controlled during the wire growth, the performance exhibits high reproducibility. Therefore, the n/p type semiconducting property, doping density, and charge mobility in a SiNW-FET can be designed in advance.

There are two major fabrication techniques in preparing SiNW-FETs: “top-down” and “bottom-up”. The top-down method is carried out through lithographic processes combined with an electron-beam technique that defines SiNW-FETs by physically etching a single-crystalline silicon wafer. On the other hand, the “bottom-up” processes start with the growth of SiNWs, normally in a chemical vapor deposition (CVD) reaction, followed by SiNW assembly and electrode fabrication via the photolithographic or electronbeam lithographic procedures [9].

The top-down method for the SiNW-FET fabrication is based on lithographic processing on a silicon-on-insulator (SOI) wafer. As illustrated in Figure 4(a, i), the structure of an SOI wafer contains three layers: substrate Si wafer, buried silicon dioxide (about 200-400 nm thick), and top Si layer (about 50-100 nm thick). Through the standard procedures of photolithography, reactive ion etching (RIE), ion implantation, electron-beam lithography, and thermal evaporation, SiNWs and the connecting electrodes can be defined to form SiNW-FET devices, in which the width of SiNWs could reach the scale of 100 nm. A typical top-down process to fabricate SiNW-FETs is illustrated schematically in Figure 4(a) [10-11].

In Step 1, the Si layer is doped with low-density boron or phosphorous of $\sim(10)^{15}/(\text{cm})^3$ (about 10-20 $\Omega \text{ cm}$). Therefore, the n/p type semiconducting property and doping ratio of SiNWs are determined (Figure 4(a, i)). Step 2 is to define the source and drain leads with heavy doping $10^{19}/(\text{cm})^3$, of which the patterns are drawn with a photomask design (Figure 4(a, ii)). In Step 3, the micrometer-sized source and drain electrodes are finished by RIE etching (Figure 4(a, iii)). Step 4 is to fabricate the nanometer-sized SiNWs with an electric-resist pattern and RIE etching (or tetramethylammonium hydroxide etching) (Figure 4(a, iv)). Subsequently, a thermal evaporation is used to make the contact leads and back-gate, and finally an insulator layer (e.g., $(\text{Al})_2\text{O}_3$, $(\text{SiO})_2$, or $(\text{Si})_3\text{N}_4$) is coated on the SiNW-FET devices. Compared with the bottom-up method, the top-down approach is more complex because the process relies on high-resolution lithography [12-13]. Figure 5 presents a schematic view of a junctionless nanowire gated resistor. Having no junctions

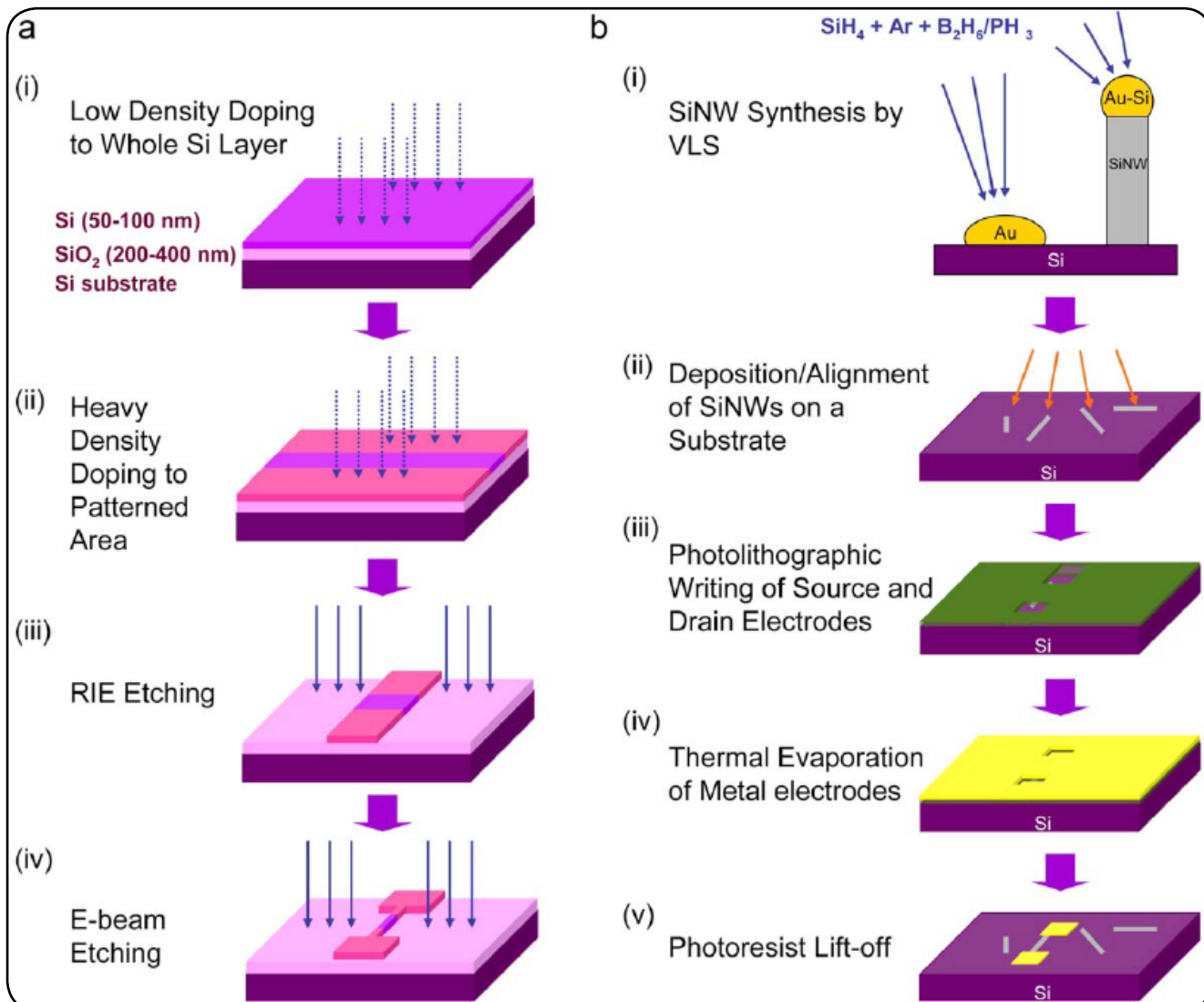


Figure 4. (a) Schematic illustration of a typical "top-down" process to fabricate SiNW FETs. (i) In Step 1, the silicon layer is doped with low-density boron or phosphorous. (ii) In Step 2, specific regions defined with a photomask pattern receive heavy doping. (iii) In Step 3, the micrometer-sized source and drain electrodes are finished by RIE etching. (iii) The following Step 4 is to fabricate the nanometer-sized SiNWs with an electric-resist pattern and RIE etching. (b) An illustration of a "bottomup" method to fabricate SiNW-FETs. (i) The growth of SiNWs in CVD reaction via the VLS mechanism. (ii) Deposition/alignment of SiNWs on a silicon substrate. (iii) A photomask pattern to define source/drain electrodes. (iv) Thermal evaporation to deposit the source/drain contacts. (v) Lift-off the remaining photoresist with Remover PG.

presents a great advantage. Modern transistors have reached such small dimensions that ultra-sharp doping concentration gradients are required in junctions: typically the doping must switch from n-type with a concentration of $1 \times 10^{19} \text{ cm}^{-3}$ to p-type with a concentration of $1 \times 10^{18} \text{ cm}^{-3}$ within a couple of nanometres. This imposes severe limitations on the processing thermal budget and necessitates the development of costly millisecond annealing techniques. In a junctionless gated resistor, on the other hand, the doping concentration in the channel is identical to that in the source and drain. The underlying insulator layer (buried oxide) is not shown. In a classical trigate device, the source and drain are heavily doped n-type and the channel region under the gate is lightly doped p-type. In the junctionless gated resistor, the silicon nanowire is uniformly doped n-type and the gate material is p-type polysilicon. Opposite dopant polarities are used for p-channel devices [14].

4. Array design of SiNW-FET

The SiNW-FET devices could be fabricated following a standard photolithographic procedure with a mask design depicted in Figure 6(a). The synthesized SiNWs were dispersed on a SiO₂/Si substrate (typically 400 nm-thick SiO₂). The as-dispersed SiNWs in the central area (the reddish

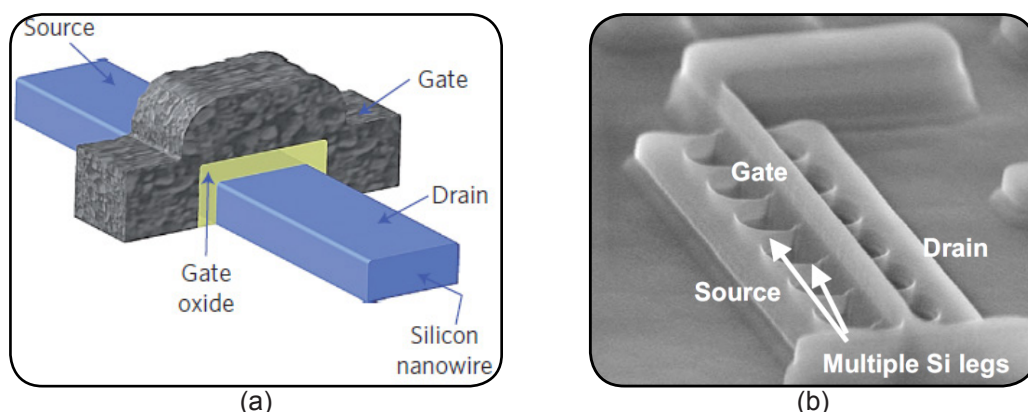


Figure 5. (a) Schematic of an n-channel nanowire transistor. (b) The scanning electron microscope of the Tri-gate transistor showing multiple Si legs [15].

rectangles in Figure 6(a) and (b)) were electrically connected by metal leads (represented in yellow in Figure 6(a)). The surfaces of the metal electrodes were further coated with an insulating layer to prevent electric leakage during sensing experiments. The bottom inset graph enlarged from the red region in Figure 6(b) displays the array design, in which the individual nanowire device was connected by metal electrodes with a separation of several micrometers. The individual SiNW situation can be seen with the image scanned by atomic force microscope (AFM), as shown in Figure 6(c) [16].

The experimental setup involved in electrical measurements includes a silicon chip ($1.5\text{mm} \times 1.5\text{mm}$) containing SiNW-FET device arrays, a PDMS microfluidic channel ($6.25\text{mm} \times 0.5\text{mm} \times 0.55\text{mm}$), and a detection system. First, the silicon chip containing SiNW-FET device arrays was mounted on a plastic circuit board and electrically connected with $30\mu\text{m}$ -diameter aluminum wires (Figure 6(d)) before electrical measurement. The PDMS microfluidic channel was then placed in the middle of the chip to allow sample solution delivery onto the SiNW-FET arrays (Figure 6(e)). The detection system including a lock-in amplifier and a current pre-amplifier was to record the electrical signals resulting from the binding events occurring on the SiNW-FET surface during sensing experiments (Figure 6(f)).

It is noted that the laminar flow in an ordinary microfluidic channel used in FET-based measurements may restrict the detection sensitivity due to the diffusion-limited sample delivery [17]. Comparatively, a specially designed microscale solution chamber with efficient sample mixing during the fluid exchange has been demonstrated to improve the detection sensitivity [18].

5. Applications of SiNW-FET sensors

5.1. Recording electrical and transmitter signals from cells

Using nano- and neuro-technologies to couple electrical interfacing with neural systems has great potential to unveil many details of neuron studies. In the past few years, SiNW-FETs and CNT-FETs have been applied for electrophysiological measurements by recording signals from neuron cells and tissues, e.g., recording the electrical signal from a single neuron and cardiomyocyte cells and detecting the released neurotransmitter of CgA from living neurons. In this section, we will introduce some current studies that demonstrate how SiNW-FET has been used to record these cell signals. Patolsky et al. have reported that hybrid SiNW-FET arrays integrated with individual axons and/or dendrites are capable of recording electrical signals from a single neuron cell. On the designed SiNW-FET array structures, they defined the adhesive zones where poly-lysine was patterned for neuron cell growth [20].

5.2. Protein-protein interaction

A huge number of approaches have been developed to understand molecular complex interactions, such as protein-protein or protein-small molecule interactions. For example, a fluorescence detection method combined with a fiber-optic biosensor has been established to study the binding kinetics of immunoglobulin G (IgG)/ anti-mouse IgG and human heart-type fatty acid-binding protein (its antibody). However, this labeling detection method was limited by some drawbacks.

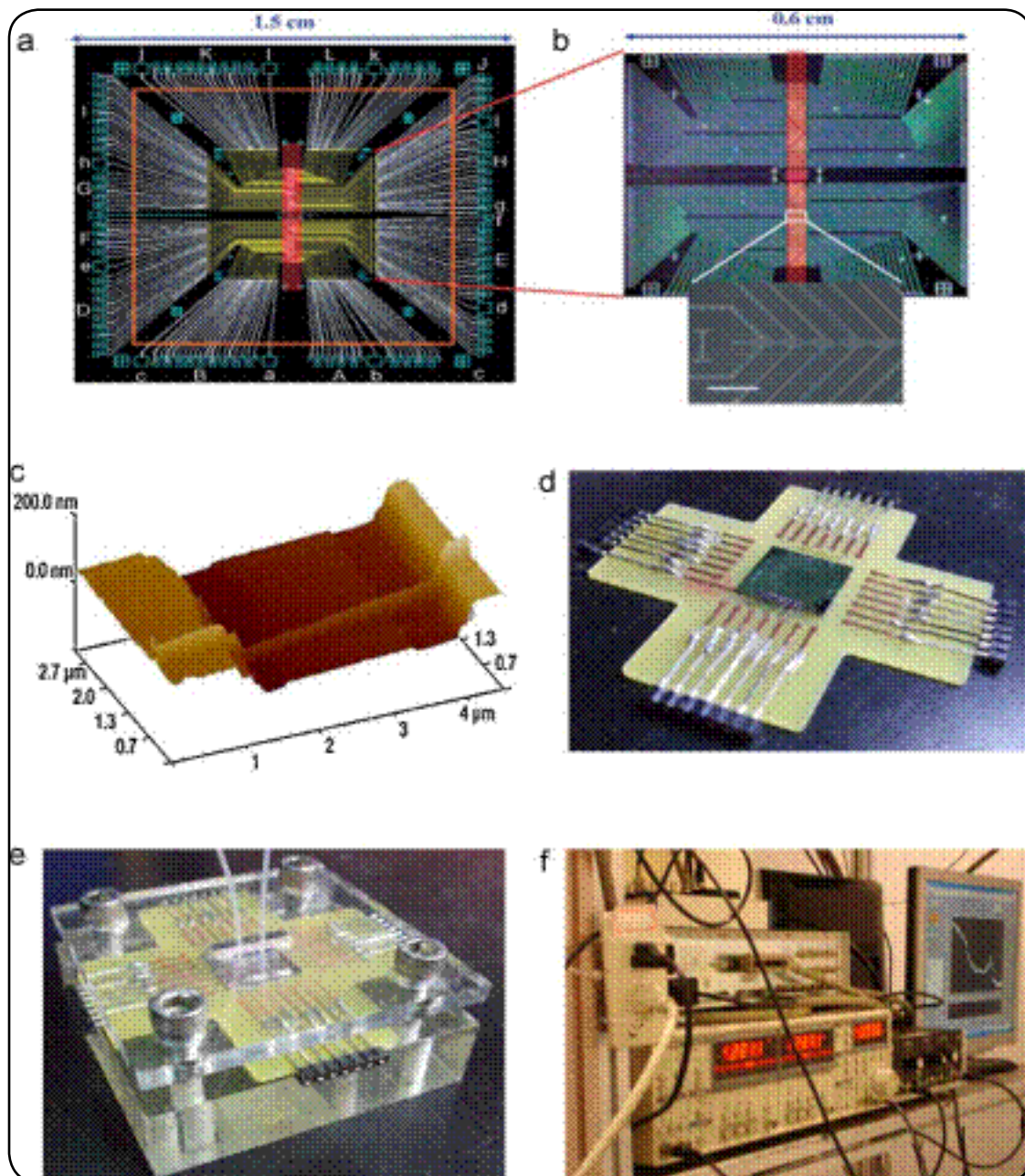


Figure 6. (a) Mask design for the photolithographic fabrication of SiNW-FET device arrays. (b) Device arrays on a magnified scale. Top: Optical image of the circuits in the area of the yellow square in (a); bottom: SEM image of a SiNW-FET array with a source-drain separation of $2\ \mu\text{m}$. The scale bar is $50\ \mu\text{m}$. (c) The topograph of a SiNW-FET scanned by AFM. A SiNW of $50\ \text{nm}$ in diameter is connected by two Ni/Al ($70\ \text{nm}/100\ \text{nm}$ in thickness) electrodes of $2\ \mu\text{m}$ in separation. (d) The SiNW-FET device arrays on a silicon chip ($1.5\text{mm} \times 1.5\text{mm}$) are connected to a plastic circuit board with aluminum wires ($3\ \mu\text{m}$ in diameter). (e) A sample solution was delivered onto the SiNW-FET arrays through a PDMS microfluidic channel ($6.25\text{mm} \times 0.5\text{mm} \times 0.05\text{mm}$), which was designed to couple with the SiNW-FET device arrays. (f) The variation of electrical signals was monitored by a detection system that combined a lock-in amplifier and a current preamplifier [19].

For instance, the surface characteristics of small proteins might be changed after chemical labeling, thus varying the labeling efficiency for different proteins, which consequently makes accurate quantification detection difficult. Also this labeling technique usually requires a huge amount of time for the labeling procedures. In recent years, some label-free detection techniques, such as surface plasmon resonance imaging (SPRI), AFM, and SiNW-FET and CNT-FET have been invented for sensing protein-protein interactions [21].

5.3. Biomarker detection

A biomarker is generally defined as something that can be used as an indicator for a particular disease state or some other biological state of an organism. For that reason, the detection of specific biomarkers can be applied to disease screening. For example, prostate-specific antigen (PSA) has already been widely applied to prostate cancer diagnosis. However, these biomarkers usually exist

in the blood in extremely low concentrations. Therefore, finding a method to rapidly and precisely detect these biomarkers is an important issue in clinical diagnoses. Recently, Zheng et al. utilized a SiNW-FET array for the detection of multiple cancer markers [22].

5.4. Three-dimensional localized bioprobes

Although a multitude of SiNW-FETs and CNT-FETs have been exploited to record extracellular electrical signals, these devices are normally created on planar substrates, making it difficult for the devices to detect the signals from arbitrary localization in three dimensions (3D). Therefore, it is highly desirable to develop a movable 3D nano-FET, containing the necessary source and drain electrical connections, that can be moved to contact a cell and even into the cell. Recently, Tian et al. made a movable 3D nano-FET through the synthetic integration of kinked SiNWs [23].

5.5. DNA hybridization

SiNW-FETs were adapted for the detection of DNA or RNA. Due to the large amount of negative charges in the phosphate backbones of DNA or RNA, SiNW-FETs offer a good candidate for monitoring DNA or RNA hybridizations, because the hybridizations cause the accumulation or depletion of charge carriers in the SiNW-FET, leading to a conductance change. Peptide nucleic acid (PNA), an artificially synthesized polymer similar to DNA, is commonly used in biological research, especially in DNA or RNA hybridizations. Hahm et al. have reported the real-time and label-free detection of DNA with a PNA-modified SiNW-FET [24].

6. Conclusion

Over the past years, the invention of biosensors has become increasingly important for many biological purposes. Biosensors constructed based on traditional methods usually require massive sample preparations or have low detection sensitivity. In contrast, nanoscale techniques have attracted more attention because of their abilities to overcome these difficulties. In this review, we described the fabrications and properties of SiNW-FET devices. In addition, we discussed the potential applications of SiNW-FETs in the fields of biomedical sciences.

References

- [1] J.H. He, Y.Y. Zhang, J. Liu, D. Moore, G. Bao, Z.L. Wang, "ZnS/Silica Nanocable Field Effect Transistors as Biological and Chemical Nanosensors", *Physical Chemistry Letters*, American Chemical Society, (2007).
- [2] D.R. Thevenot, K. Toth, R.A. Durst, G.S. Wilson, "Electrochemical biosensors: recommended definitions and classification", *Biosensors & Bioelectronics*, Elsevier, (2001).
- [3] D. Altschuh, "Immunochemistry and Biosensors", Parc d'Innovation, (2004).
- [4] S. Ripp, G.S. Saylor, "Advanced Bioreporter Technologies For Targeted Sensing of Chemical and Biological Agents", *Biosensors*, Center for Environmental Biotechnology, (2002).
- [5] J.P. Chambers, B.P. Arulanandam, L.L. Matta, A. Weis, J.J. Valdes, "Biosensor Recognition Elements", *Current Issues in Molecular Biology*, Horizon Scientific Press, (2008).
- [6] K.I. Chen, B.R. Li, Y.T. Chen, "Silicon nanowire field-effect transistor-based biosensors for biomedical diagnosis and cellular recording investigation", *Nanotoday*, Elsevier, (2011).
- [7] Q. Qing, S.K. Pal, B. Tian, Xiaojie Duan, B.P. Timko, T. Cohen-Karni, V.N. Murthy, C.M. Lieber, "Nanowire transistor arrays for mapping neural circuits in acute brain slices", *Applied Physical Sciences*, Proceedings of the National Academy of Sciences, (2010).
- [8] D.D.D. Ma, C.S. Lee, F.C.K. Au, S.Y. Tong, S.T. Lee, "Small-Diameter Silicon Nanowire Surfaces", *Science*, (2003).
- [9] F. Patolsky, G. Zheng, C.M. Lieber, "Fabrication of silicon nanowire devices for ultrasensitive, label-free, real-time detection of biological and chemical species", *Nature Protocols*, Nature Publishing Group, (2006).
- [10] O. Knopfmacher, A. Tarasov, W. Fu, M. Wipf, B. Niesen, M. Calame, C. Schonenberger, "Nernst Limit in Dual-Gated Si-Nanowire FET Sensors", *Nano Letters*, American Chemical Society, (2010).
- [11] A.L. Briseno, S.C.B. Mannsfeld, S.A. Jenekhe, Z. Bao, Y. Xia, "Introducing organic nanowire transistors", *Materials Today*, Elsevier, (2008).
- [12] B. Reddy Jr, B.R. Dorvel, J. Go, P.R. Nair, O.H. Elibol, G.M. Credo, J.S. Daniels, E.K.C. Chow,

- X. Su, M. Varma, M.A. Alam, R. Bashir, "High-k dielectric Al₂O₃ nanowire and nanoplate field effect sensors for improved pH sensing", Biomed Microdevices, Springer, (2010).
- [13] F. Larki, S.D. Hutagalung, A. Dehzangi, E.B. Saion, A. Abedini, A.M Abdullah, M.N. Hamidon, J. Hassan, "Electronic Transport Properties of Junctionless Lateral Gate Silicon Nanowire Transistor Fabricated by Atomic Force Microscope Nanolithography", Microelectronics and Solid State Electronics, Scientific & Academic, (2012).
- [14] J.Pierre Colinge, C.W Lee, A. Afzalian, N.D. Akhavan, R. Yan, I. Ferain, P. Razavi, B. O'Neill, A. Blake, M. White, A.M. Kelleher, B. McCarthy, R. Murphy, "Nanowire transistors without junctions", Nature Nanotechnology, Macmillan Publishers Ltd, (2010).
- [15] R. Chau, B. Boyanov, B. Doyle, M. Doczy, S. Datta, S. Harelend, B. Jin, J. Kavalieros, M. Metz, "Silicon nano-transistors for logic applications", Physica E, Elsevier, (2003).
- [16] C.C. Tsai, P.L. Chiang, C.J. Sun, T.W. Lin, M.H. Tsai, Y.C. Chang, Y.T. Chen, "Surface potential variations on a silicon nanowire transistor in biomolecular modification and detection", Nanotechnology, Institute Of Physics, (2011).
- [17] P.E. Sheehan, L.J. Whitman, "Detection Limits for Nanoscale Biosensors", Nano Letters, American Chemical Society, (2005).
- [18] J.H. Chua, R.E. Chee, A. Agarwal, S.M. Wong, G.J. Zhang, "Label-Free Electrical Detection of Cardiac Biomarker with Complementary Metal-Oxide Semiconductor-Compatible Silicon Nanowire Sensor Arrays", Analytical Chemistry, American Chemical Society, (2009).
- [19] P.L. Chiang, T.C. Chou, T.H. Wu, C.C. Li, C.D. Liao, J.Y. Lin, M.H. Tsai, C.C. Tsai, C.J. Sun, C.H. Wang, J.M. Fang, Y.T. Chen, "Nanowire transistor-based ultrasensitive virus detection with reversible surface functionalization", Chemistry – An Asian journal, (2010).
- [20] F. Patolsky, B.P. Timko, G. Yu, Y. Fang, A.B. Greytak, G. Zheng, C.M. Lieber, "Detection, Stimulation, and Inhibition of Neuronal Signals with High-Density Nanowire Transistor Arrays", Science, (2006).
- [21] G. Pampalakis, S.O. Kelley, "An Electrochemical Immunosensor Based on Antibody-Nanowire Conjugates", Analyst, (2009).
- [22] G. Zheng, F. Patolsky, Y. Cui, W.U. Wang, C.M. Lieber, "Multiplexed electrical detection of cancer markers with nanowire sensor arrays", Nature biotechnology, Nature Publishing Group, (2005).
- [23] B. Tian, T. Cohen-Karni, Q. Qing, X. Duan, P. Xie, C.M. Lieber, "Three-dimensional, flexible nanoscale field effect transistors as localized bioprobes", Science, National Institutes of Health, (2010).
- [24] J. Hahm, C.M. Lieber, "Direct ultrasensitive electrical detection of DNA and DNA sequence variations using nanowire nanosensors", Nano letters, American Chemical Society, (2004).

Altium Designer 10.1271.26245



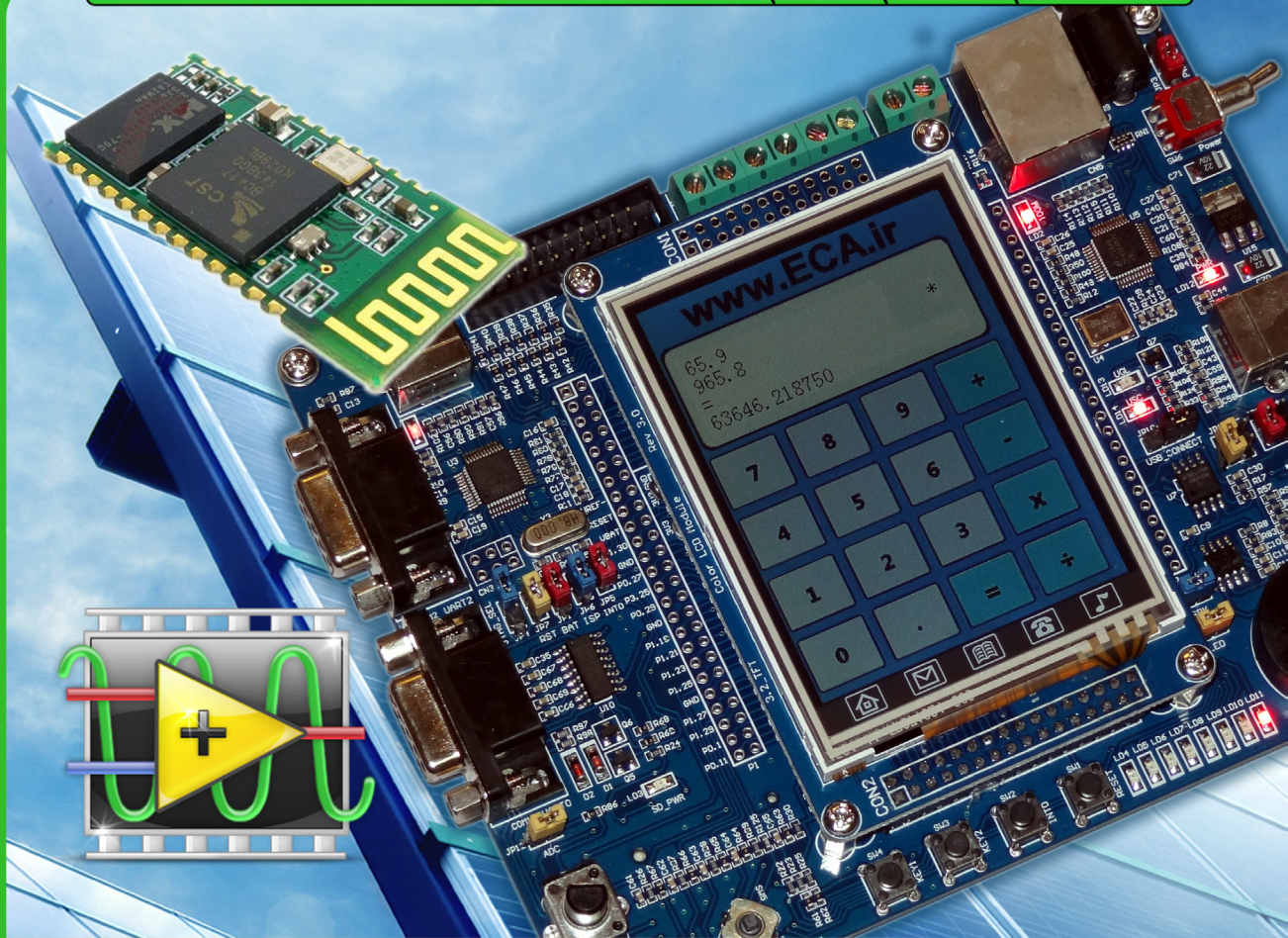
Altium Designer یا همان PROTEL DXP نرم افزار قدرتمندی است که برای پیاده سازی شماتیک، طراحی PCB و آنالیز مدارهای آنالوگ و برخی مدارهای دیجیتال طراحی شده است. یکی از مزایای این نرم افزار دسته بندی مناسب کتابخانه ها بنحوی است که با صرف زمان کوتاهی قطعه مورد نظر را خواهید یافت. آنالیز مدارهای آنالوگ در پروتل، توسط تحلیل گر پی اسپایس انجام می شود. محیط طراحی PCB در پروتل، بدلیل داشتن کتابخانه هایی کامل و بدون نقص معروف است و این امر سبب رفاه بیشتر کاربر در حین طراحی انواع PCB با این نرم افزار، خواهد شد. در این نسخه امکان شبیه سازی و کد نویسی برخی از FPGA ها نیز فراهم شده است که زمان طراحی و پیاده سازی را حداقل می کند. ورژن جدید این نرم افزار حرفه ای دارای تحولات و پیشرفت های زیادی بوده که کار شما را برای طراحی نسل های بعدی مدارات الکترونیک بسیار ساده کرده است.

توضیحات تکمیلی :

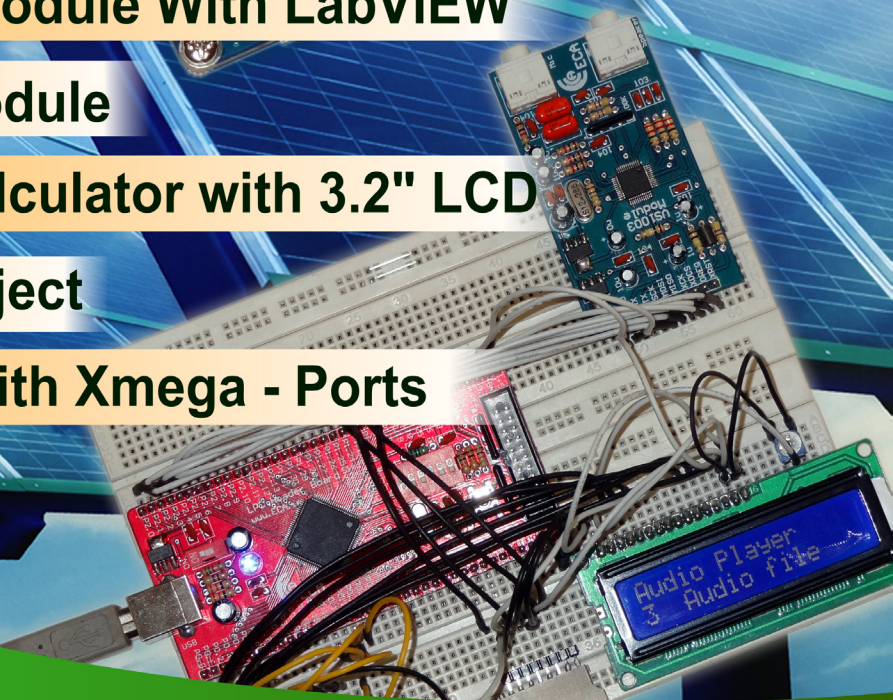
<http://eshop.eca.ir/link/1404.php>

Noise

ECA Electrical & Electronics Magazine / Vol.3 / No.13 / Dec 2012



- Driving SIM900 Module With LabVIEW
- Driving HC-05 Module
- Touch Screen Calculator with 3.2" LCD
- Audio Player Project
- Getting started with Xmega - Ports



www.Noisemagazine.ECA.ir