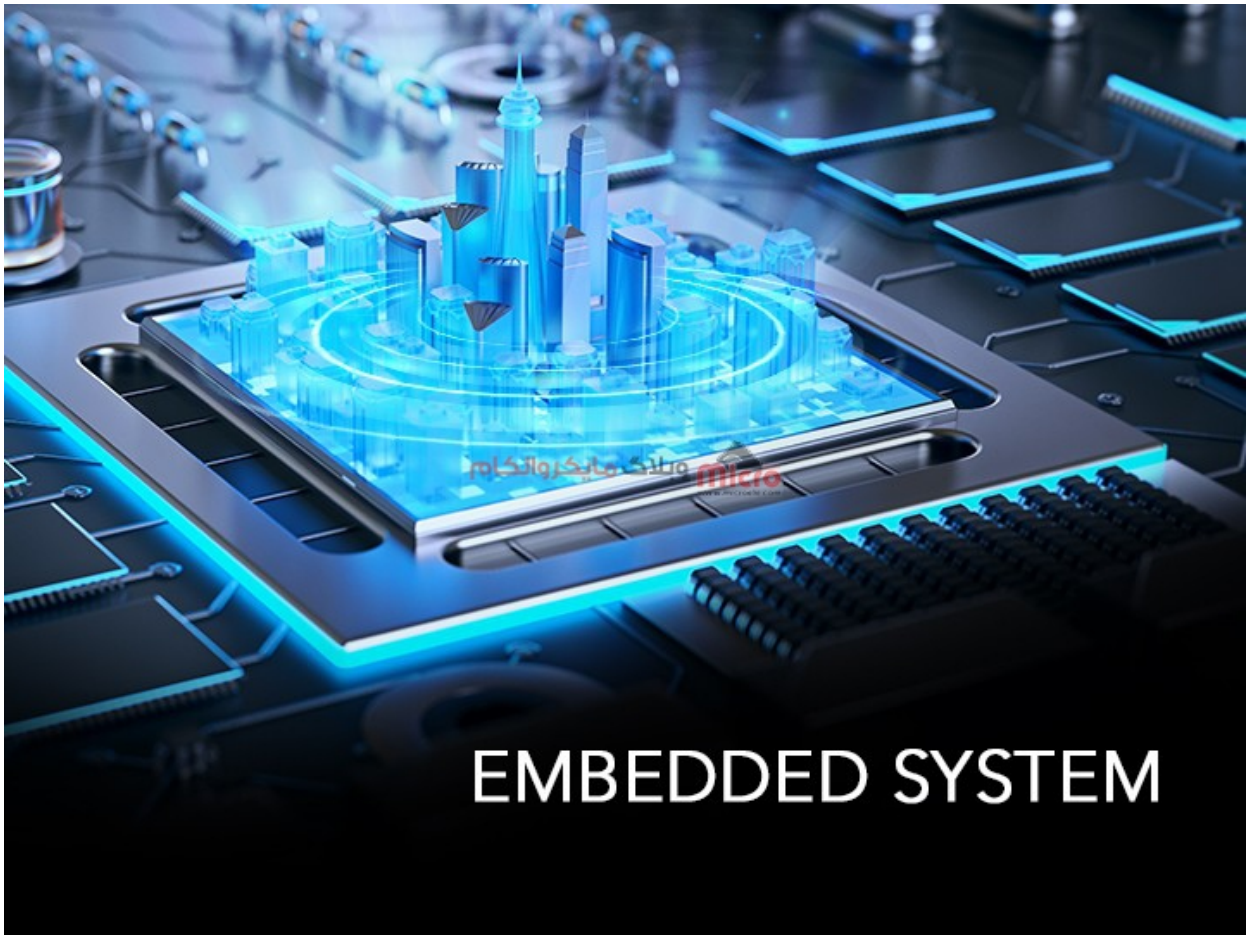
12 مهارت مهم و اساسی برای توسعه دهندگان سیستم های EMBEDDED

تاریخ انتشار۱۱ شهریور, ۱۴۰۱ توسط سید حسین سلطانی

سلام و درود خدمت همراهان همیشگی مایکروالکام. یکی از پوزیشن های شغلی اکثر شرکت ها تحت عنوان Embedded System Designer می‌باشد. برای تبدیل شدن به یک طراح سیستم های امبدد نیاز به برخی مهارت ها است. در این مطلب به موارد اصلی آن پرداخته شده است. پس با من تا انتهای مطلب همراه باشید. همچنین شما میتونید سایر مطالب من رو از این قسمت دنبال کنید.

توسعه سیستم های نهفته (Embedded System)

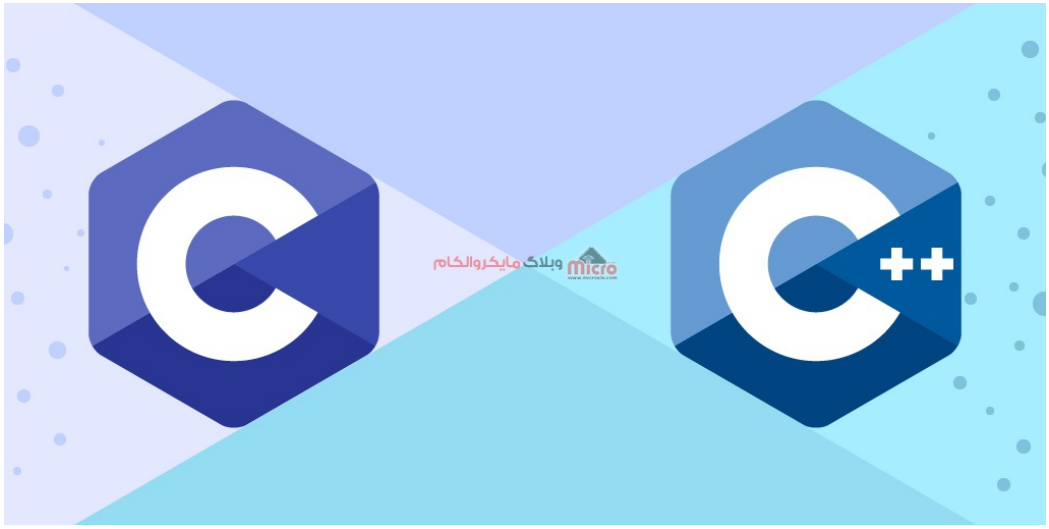
صنعت الکترونیک بزودی نیاز مبرمی به توسعه دهندگان جدید شامل طراحان PCB و توسعه دهندگان امبدد (Embedded System) خواهد داشت. هر چند شاهد این نیاز تا به امروز بوده ایم. برای یک مهندس جدید سیستم امبدد یک پل بین ساخت سخت افزار و نرم افزار است. یک توسعه دهنده سیسم های امبدد (Embedded System) باید تا حدودی در باره سخت افزاری که با آن کار می‌کند اطلاعاتی داشته باشد. اما همین کافی نبوده و مهارت های دیگری نیز مورد نیاز خواهد بود که در ادامه به آن پرداخته شده است.



سیستم های امبدد (Embedded Sysytem)

1- برنامه نویسی C/C++

یکی از پرچمدارترین و پر طرف دار ترین زبان ها برای برنامه نویسی سیستم های Embedded زبان C/C++ می‌باشد. اضافه بر این IDE های متنوعی نیز با قابلیت پشتیبانی از این زبان برنامه نیسی نیز وجود دارد. بسیاری از پروژه های اوپن سورس یا کتابخانه های موجود نیز با زبان C/C++ توسعه داده شده است. لذا همین دلیلی بر محبوبیت و همه گیری این زبان خواهد بود و فراگیری آن امری بدیهی و مهم می‌باشد.



برنامه نویسی C - C++

2- برنامه نویسی پایتون (Python)

اخیرا زبان برنامه نویسی پایتون بعنوان یکی از اصلی ترین زبان ها برای استفاده توسعه دهندگان نرم افزار برای ساخت نرم افزار های PC، برنامه های تحت وب و گاها اپلیکیشن های موبایل نیز ظاهر شده است. در الکترونیک نیز برخی از IDE های میکروکنترلر ها از زبان MicroPython پشتیبانی کرده و می‌توان برای آن به زبان پایتون کد نوشت.

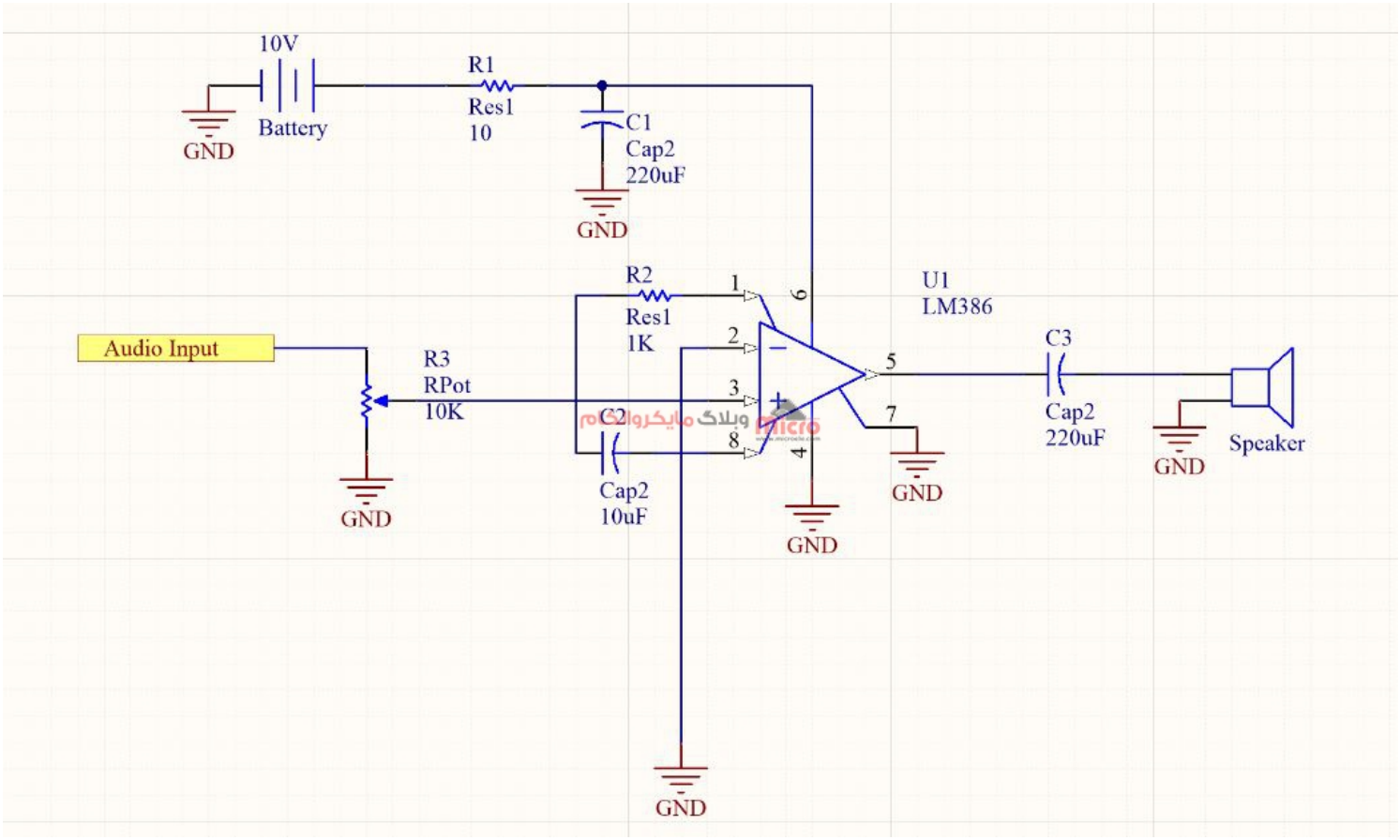
زبان برنامه نویسی پایتون

3- مهارت بهینه سازی الگوریتم

زمانی که یک الگوریتم اصلی توسعه و تایید شد، باید بدانیم آن را چگونه بهینه تر نماییم. **بهینه کردن الگوریتم همانا و کاهش حجم کاری سیستم و صرفه جویی در منابع نیز همانا**. پس بهینه کردن الگوریتم به کاهش مراحل محاسباتی اضافه یا غیر ضروری اشاره دارد. لذا باید این حقیقت را پذیرفت که باید در پایان مشخص کردن الگوریتم خود را چندین بار بررسی کرده و اضافات آن را حذف و آن را بهینه نمود.

4- خواندن شماتیک و نقشه های فنی

این مهارت یکی از مهارت های اصلی و مهم برای هر مهندس الکترونیک می‌باشد. یک طراح سیستم های نهفته (Embedded System) ممکن است در طراحی سخت افزار و مدار ورود پیدا کند. لذا بایستی از طراحی و پیش برد این هدف نیز اطلاع داشته باشد.



شماتیک یک مدار الکترونیکی

5- سیگنال دیجیتال و زمان بندی

در نگاه اول ممکن این مهارت گسترده باشد. اما فهمیدن و درک اینکه سطوح سیگنال دیجیتال چیست و چگونه نمودار زمانی آنها را بررسی کرد یکی از مهم ترین کار ها در طراحی مدار می‌باشد. اگر از رفتار سیگنال ها اطلاع داشته باشیم، می‌توان یک PCB درست و اصولی با رعایت الزمات کنترل امپدانس، PCB های فرکانس بالا و... طراحی کرد.

6- ارتباط و پروتکل های رایج دیجیتال

سیستم های امبدد مبتنی بر میکروکنترلر یا CPU ها عموما از رابط و پروتکل های رایج برای ارسال و دریافت دیتا و سیگنال استفاده می‌کنند. یکی دیگر از موارد مهم و مورد نیاز درک رفتار این پروتکل ها و نحوه عملکرد آنها خواهد بود. مثلا هنگامی که قالب بندی ارتباط سریال را ندانیم با بررسی سیگنال های تایمینگ آن، از صحت عملکرد آن مطلع نخواهیم شد مگر آنکه بدانیم قالب بندی چیست.

امروزه صنعت الکترونیک تا حد زیادی با مجموعه ای از رابط های سریال که بیشتر در پردازنده های امبدد و ASIC ها استفاده می‌شود همگرا شده است. چهار رابط سریال رایج عبارت اند از UART، I2C، SPI و GPIO های قابل تنظیم می‌باشد.

متناسب با نوع قطعات مصرفی این رابط ها ممکن بصورت سیگنال های پر سرعت بوده مانند FPGA ها و در برخی از میکروکنترلر ها رفتار عادی یا کم سرعتی داشته باشند. لذا دانستن این امر مهم بوده تا بتوان یک PCB و برد اصولی متناسب با قطعات مورد استفاده خود در سیستم تعبیه و طراحی کرد.

در سیستم های پیشرفته تر متناسب با نوع قطعات مصرفی برای ارتباط با چیپ اصلی از رابط های سرعت بالا استفاده می‌شود. این رابط ها شامل انواع زیر می‌باشد.

- انواع اترنت (SGMII، RMII، MII و ... عموما تا 1Gbit)
- ارتباط USB (عموما از USB 2.0 اما در برخی از سیستم های جدید USB 3.0)
- DDR2 و بیشتر
- PCIe ها عموما Gen2 یا بروز تر
- SerDes (ممکن است استاندارد نباشد)

7- نصب و دستورات پایه لینوکس

برخی از سیستم های امبدد مبتنی بر توزیع لینوکس می‌باشند. فریمور بیشتر تجهیزات شبکه و تک بردی اکثرا مبتنی برروی یک توزیع لینوکس ساخته شده اند. از همین رو دانستن دستورات پایه لینوکس برای کار با این سیستم ها مهم

خواهد بود. در سطح ابتدایی به برخی از دستورات لینوکس جهت استفاده در ترمینال نیاز خواهیم داشت. در سطح پیشرفته تر، ممکن است نیاز به نصب و اجرای توزیع لینوکس برروی یک سیستم جدید داشته باشیم.

8- ترمینال پایه و برنامه های کاربردی SSH

بدلیل اینکه ممکن است بیشتر سیستم ها برروی لینوکس کار کنند یا در سیستم عامل های بلادرنگ (RTOS)، ممکن است برای برقرای ارتباط با دستگاه توسط سیستم خود نیاز به استفاده از یک ترمینال با تونل SSH داشته باشیم. برخی از ترمینال های کاربردی در ادامه معرفی شده است.

- Tera-Term: برقرای ارتباط TCP/IP یا ارتباط سریال با پورت COM
- PuTTY: عملکردی مشابه TeraTerm اما با قابلیت های بیشتر نسبت به آن
- Git Bash: استفاده به عنوان خط فرمان لینوکسی در دستگاه محلی
- WinSCP: از آن می‌توان برای ارسال فایل به یک سیستم راه دور با آپشن های مشابه TeraTerm و Putty استفاده کرد اما دارای رابط کاربری گرافیکی می‌باشد.

9- ابزار های پروگرم کردن و دیباگینگ

پردازنده و حافظه های مختلف مستلزم استفاده از ابزار توسعه مختلف و متناسب خود می‌باشند. این ابزار برای پروگرم کردن توسط USB به سیستم متصل شده و نرم افزار مخصوص خود را دارند. برای دیباگینگ باید سیستم از طریق ارتباط سریال از طریق پورت COM در حالت debuging قرار گیرد.

هنگامی که دستگاه در این حالت تنظیم شد، می‌تواند ارور ها و نحوه عملکرد خود را از طریق ارتباط سریال ارسال کرده و در سیستم برروی یک محیط مخصوص آنها را مشاهده نمود.

10- سیستم کنترل ورژن

هنگامی که برروی یک پروژه کار می‌کنیم، هر نسخه توسعه یافته از پروژه باید در جایی ذحیره شده و تغییرات آن مشخص باشد. به بیانی ساده به این فضا سیستم کنترل ورژن گویند. با این کار اگر نیاز باشد به ورژن قبلی دسترسی داشته باشیم و اصلاحی انجام دهیم یا از بک آپ استفاده کنیم براحتی امکان پذیر خواهد بود. رایج ترین این سیستم ها Git بوده که اولین بار توسط خالق لینوکس یعنی لیوس تورو والدز توسعه داده شد.

سیستم مدیریت ورژن گیت

11- مقدمات پایه توسعه وب

این مهارت شاید در نگاه اول غیر منتظره و بی ربط به نظر آید. اما باید گفت یک توسعه دهنده سیستم های امبدد ممکن است گاهی در پروژه خود نیاز به تعامل و برقرای ارتباط با یک فضای ابری یا وب سرور داشته باشد. پس باید با یکسری از مقدمات اولیه وب آشنایی داشته باشد. بعنوان مثال برای یک طراح Embedded System نحوه ارسال و دریافت و کار با API ها، فرمت های رایج دیتا مانند JSON، کار با دیتابیس مانند SQL و... بسیار کارآمد خواهد بود.

12- مهارت های نرم

شاید این عنوان کمی سوال بر انگیز باشد. آخرین اما نه کم اهمیت ترین از بین این موارد، مهارت های نرم است. برخی از مجموعه مهارت ها وجود دارد که در بین توسعه دهندگان دارای محبوبیت بالایی است اما چیزی نیست که براحتی در کلاس های آموزش یاد داده شود. به این دست مهارت ها، مهارت های نرم گفته می‌شود.

وظیفه یک توسعه دهنده Embedded System فراتر از نوشتن کد و برنامه نویسی است. بلکه باید با همکاران بخش طراح مدار و PCB همکاری داشته تا یک سیستم جدید بدرستی خلق گردد.

نتیجه گیری

آنچه که در این مطلب به آن پرداخته شد همه مهارت های مورد نیاز یک طراحی امبدد کار نیست. چرا که یک طرح و پروژه پیشرفته تر مستلزم بکار گیری و استفاده از مهارت های دیگر و بیشتری می‌باشد. لذا این موارد بخشی از مهارت های لازم برای تبدیل شدن به یک طراح سیستم های نهفته (Embedded System) در کنار تجارب عملی دیگر است.

امیدوارم از این آموزش کمال بهره را برده باشید. در صورتی که هرگونه نظر یا سوال داشتید درباره این آموزش لطفا

اون رو در انتهای همین صفحه در قسمت دیدگاه ها قرار بدید. در کوتاه ترین زمان ممکن به اون ها پاسخ خواهم داد. اگر این مطلب براتون مفید بود، اون رو حتما به اشتراک بگذارید. همینطور میتونید این آموزش را پس از اجرای عملی توی اینستاگرام با هشتگ microelecom# به اشتراک بگذارید و پیج مایکروالکام (microelecom@) رو هم منشن کنید.