استفاده از شماره شناسایی منحصر به فرد میکروکنترلر برای جلوگیری از کپی کردن غیر مجاز برنامه (ویرایش دوم)

نویسنده: اوژن کی نژاد

یکی از دغدغه های تولید کنندگان پروژه ها و محصولات مبتنی بر میکروکنترلرها، عدم دسترسی افراد غیرمجاز به محتوای حافظه برنامه و کپی کردن آن روی میکروکنترلرهای دیگر است و استفاده از روش قفل (Lock) کردن میکروکنترلر به همین منظور انجام می شود. اما با توجه به اینکه اخیرا با صرف هزینه کافی و استفاده از برخی روش ها می توان به چیپ میکروکنترلر دسترسی پیدا کرد و حتی محتوای حافظه برنامه میکروکنترلرهای قفل شده را تحت مراحل مشخصی کپی کرد، در این مقاله روشی برای میکروکنترلرهایی که دارای شماره شناسایی یکتا و منحصر به فرد هستند معرفی می شود که حتی در صورت دسترسی به محتوای حافظه برنامه توسط افراد غیر مجاز و سارقان اطلاعات، این محتوا قابلیت اجرا روی میکروکنترلرهای مشابه را نداشته باشد. در این روش ابتدا یک برنامه مشترک بوسیله پروگرامر روی هر تعداد میکروکنترلر که مورد نظر باشد، برنامه ریزی و اجرا می شود و اینگونه نیست که آن برنامه مشترک فقط روی یک میکروکنترلر اجرا شود. بنابراین در کاربردهای تولیدی می توان همان برنامه را به تعداد لازم و بصورت مشترک برای محصولات مورد استفاده قرار داد. اما اگر بعد از برنامه ریزی و یک بار اجرا به هر طریقی برنامه از حافظه هر کدام از آن میکروکنترلرها خوانده شود و روی چیپ مشابهی ریخته شود، در اینصورت خروجی آن به شکل مورد انتظار عمل نخواهد کرد و به این ترتیب سارقان اطلاعات نمی توانند با دستیابی به محتوای حافظه برنامه به مقصود خود برای کپی کردن غیر مجاز آن برسند.

این روش مبتنی بر چند پیش فرض است:

۱- از خانواده ای از میکروکنترلرها استفاده می کنیم که هر یک از آنها دارای یک شماره شناسایی یکتا و منحصر به فرد هستند.

۲- این شماره شناسایی در هنگام اجرای برنامه توسط CPU می تواند مورد دسترسی قرار بگیرد.

۳- حافظه غیر فراری مانند EEPROM یا Flash برای نوشتن مقادیری در اختیار CPU است. در توضیحات بعدی، فرض بر استفاده از EEPROM داخلی برای این منظور است. اما بر حسب ضرورت و نوع میکروکنترلر مورد استفاده ممکن است از حافظه Flash داخلی هم برای این منظور استفاده شود.

۴- بعد از برنامه ریزی میکروکنترلر و قبل از دسترسی افراد غیر مجاز به آن، حداقل یک بار برنامه روی آن اجرا شده است.

با این پیش فرض ها و برای اینکه محتوای Flash و EEPROM حتی در صورت کپی شدن، برای برنامه ریزی روی میکروکنترلر دیگر قابل استفاده نباشد، مراحل زیر قابل انجام است:

۱- در هنگام برنامه نویسی، مقادیر اولیه مشخصی را برای یک یا چند آدرس از حافظه EEPROM داخلی تعیین می کنیم. برای سادگی فعلا فرض می کنیم این کار برای یک آدرس انجام شود. اما در عمل برای بالا رفتن امنیت این روش می توان از چند آدرس استفاده کرد. مثلا فرض می کنیم که مقدار اولیه 0x55 یا هر مقدار مورد نظر دیگر توسط پروگرامر در آدرس مشخصی از EEPROM برنامه ریزی شود.

۲- نحوه کدنویسی به این صورت است که در ابتدای اجرای برنامه، مقدار موجود در آن آدرس (یا آدرس ها) از EEPROM توسط CPU بررسی می شود و اگر با مقدار اولیه (مثلا 0x55 برای یک آدرس) مطابقت داشت، شماره شناسایی میکروکنترلر توسط CPU خوانده می شود و در محدوده مشخصی از EEPROM ذخیره می شود. این ذخیره سازی می تواند توام با روش های رمزنگاری باشد تا از طریق تطبیق محتوای EEPROM با شماره شناسایی میکروکنترلری که قرار است اطلاعات از روی آن کپی شود، این ناحیه قابل شناسایی نباشد.

۳- اگر مقدار موجود در آدرس مذکور با مقدار اولیه آن تطبیق نداشت، CPU از این مرحله عبور می کند.

۴- بعد از ذخیره سازی شماره شناسایی در بار اولی که برنامه اجرا می شود، CPU مقدار اولیه نوشته شده در آن آدرس (یا آدرس های) مشخص در EEPROM را که در بند ۱ ذکر شد، تغییر می دهد. مثلا 0x55 به 0xff تغییر داده می شود.

۵- در ادامه CPU مجددا شماره شناسایی میکروکنترلر را می خواند و با ناحیه ای از EEPROM که باید این شماره شناسایی (یا تغییر یافته آن) وجود داشته باشد، مطابقت می دهد. حال چنانچه این تطابق برقرار بود که روند عادی اجرای برنامه به انجام می رسد. اما در صورت عدم تطابق، از طریق انجام عملیاتی مانند اجرای حلقه های بی نهایت و مانند آن روند اجرای برنامه مختل می شود تا عملکرد آن قابل استفاده نباشد.

با روشی که توضیح داده شد اگر هر تعداد میکروکنترلر با این برنامه و بوسیله پروگرامر برنامه ریزی و محتوای حافظه برنامه آن توسط CPU اجرا شود، به دلیل اینکه در بار اول اجرای برنامه، آدرسی در EEPROM که در حکم مجوز کپی کردن شماره شناسایی به ناحیه EEPROM است در مقدار اولیه خود قرار دارد، این عملیات کپی کردن شماره شناسایی به EEPROM یک بار انجام می شود و در ادامه هم به دلیل وجود مطابقت در محتوای EEPROM با شماره شناسایی، روند اجرای برنامه به صورت عادی طی خواهد شد. در بارهای بعدی هم به دلیل تغییر محتوای موجود در این آدرس، عملیات کپی کردن شماره شناسایی به EEPROM انجام نمی شود. اما به دلیل همان بار اولی که این کپی کردن انجام شده، مطابقت برای بارهای بعدی هم وجود خواهد داشت و اجرای برنامه روند عادی خود را در دفعات بعدی هم طی خواهد کرد.

حال فرض کنیم با هر روشی محتوای Flash و EEPROM میکروکنترلر روی آی سی مشابهی کپی شود. در این شرایط به دلیل اینکه بخشی از EEPROM که به عنوان مجوز کپی کردن شماره شناسایی توسط CPU به EEPROM عمل می کند در وضعیت اولیه خود قرار ندارد، بنابراین عملیات نوشتن شماره شناسایی یا تغییر یافته آن به

EEPROM حتی در بار اول اجرای برنامه روی میکروکنترلر جدید توسط CPU انجام نمی شود. در ادامه هم به دلیل اینکه حافظه EEPROM قبلا برای میکروکنترلر دیگری با شماره شناسایی متفاوتی مقداردهی شده و وجود مغایرت بین شماره شناسایی میکروکنترلر جدید و محتوای ناحیه متناظر در EEPROM، برنامه قابلیت اجرای روند عادی خود را روی میکروکنترلر جدید نخواهد داشت و عملا قابل استفاده نخواهد بود. نکته اصلی در این روش این است که فقط وقتی شماره شناسایی توسط CPU در EEPROM کپی می شود که آدرس یا آدرس هایی دارای مقدار اولیه معینی باشند که توسط پروگرامر روی آنها نوشته شده است. اما از آنجایی که این مقادیر اولیه بعدا تغییر می کنند و کپی کنندگان غیرمجاز هم به این مقادیر اولیه EEPROM دسترسی ندارند، بنابراین دسترسی به محتوای حافظه های Flash و EEPROM برای آنها سود چندانی نخواهد داشت و تنها با تجزیه و تحلیل کدهای اسمبلی برنامه و تغییر کدها ممکن است بتوانند از آن استفاده کنند.

۱۳۹۵/۰۶/۱۳