

۱. در این پروژه باید الگوریتم فوق را به صورت هم‌طراحی سخت‌افزار و نرم‌افزار، در محیط شبیه‌سازی GEZEL با استفاده از میکروکنترلر 8051، پیاده‌سازی نمایید. شبیه‌ساز GEZEL امکان استفاده از یک میکروکنترلر 8051 به صورت یک ipcore و تعریف ارتباطات مورد نیاز بین سخت‌افزار تولید شده و کد نرم‌افزاری را فراهم می‌نماید. مثال ارائه شده در بخش 13.2 از کتاب مرجع، سیستم پیاده‌سازی شده با استفاده از زبان GEZEL به همراه یک میکروکنترلر 8051 را نشان می‌دهد. در این مثال کد نرم‌افزار یا Software Driver به زبان C، با استفاده از روش Memory-Mapped و براساس محدودیت‌های میکروکنترلر 8051 نوشته شده است. کد قسمت نرم‌افزاری در این مثال با استفاده از کامپایلر sdcc کامپایل شده و خروجی آن در کد gezel مورد استفاده قرار گرفته است. سرانجام پس از آماده شدن سیستم، عملیات شبیه‌سازی با استفاده از دستور gplatform انجام شده است.

- کامپایلر sdcc را می‌توانید با استفاده از دستور زیر بر روی سیستم خود نصب نمایید.

> sudo apt-get install sdcc

۱. در این پروژه قسمت نرم‌افزاری وظیفه انتقال داده‌های مربوط به تصاویر به قسمت سخت‌افزاری را دارد. به این منظور باید مشابه مثال، یک محدوده از حافظه به اندازه ذخیره یک تصویر را به صورت مشترک بین سخت‌افزار و نرم‌افزار تعریف نمایید. سپس قسمت نرم‌افزاری باید به ترتیب مقادیر مربوط به تصاویر که در آرایه‌های متناظر ذخیره شده‌اند را درون این حافظه مشترک نوشته و پس از انتقال هر تصویر منتظر بماند تا عملیات مربوط به آن تصویر به پایان برسد و سپس تصویر بعدی را انتقال دهد.

- هر تصویر را به صورت یک ماتریس 10×10 شامل ۱۰۰ عدد تصادفی ۸بیتی (۰ تا ۲۵۵) در نظر بگیرید.

- به منظور نمایش نتایج، مقادیر فیلتر شده پیکسل‌ها را در قسمت سخت‌افزاری با استفاده از دستور display نمایش دهید.

- براساس نیاز می‌توانید از رجیسترهای اضافی به منظور کنترل فرآیند اجرای الگوریتم استفاده نمایید.

۲. گزارش مختصری از فرآیند انجام پروژه به همراه تصاویر مناسب، در قالب یک گزارش با فرمت خواسته شده به انضمام کدهای نوشته شده و فایل‌های خروجی ابزار را به صورت فشرده، همراه با نام و شماره دانشجویی در سامانه درس‌افزار بارگذاری نمایید (قالب گزارش از قسمت فایل‌ها قابل دسترسی است).

- فایل ارسالی حتماً باید شامل نام و شماره دانشجویی باشد.