



دانشگاه صنعتی شریف
دانشکده مهندسی کامپیوتر

گزارش کار پروژه ساختار و زبان کامپیوتر

نگارش

حنانه رفیعی
ریحانه سید علوی
فاطمه غزاله فتحی
فاطمه اسماعیلی

استاد راهنمای پروژه

استاد حسین اسدی

شهریور ۱۴۰۳



عنوان پروژه:

پروژه اول : شبیه ساز مدارات منطقی با قابلیت طراحی، انتقال و اجرای زنده روی برد آردوینو

فهرست مطالب

۱	مقدمه	۱
۱	اهداف پروژه	۱-۱
۲	تجهیزات و نرم افزارها	۲-۱
۲	روش کار	۳-۱
۴	چالش ها	۴-۱
۵	نتیجه	۵-۱

فصل ۱

مقدمه

منطق قابل پیکربندی مجدد Reconfigurable Logic یکی از مفاهیم نوین در مهندسی برق و کامپیوتر است که امکان تغییر ساختار سخت افزار دیجیتال را به صورت انعطاف پذیر و پویا فراهم می کند. این فناوری به ما این امکان را می دهد که مدارهای منطقی را متناسب با نیازهای خاص هر کاربرد تنظیم کنیم و در نتیجه، عملکرد سیستم را بهبود ببخشیم.

یکی از ویژگی های مهم این فناوری، قابلیت برنامه ریزی مجدد مدارهای دیجیتال از طریق ابزارهای نرم افزاری است، بدون اینکه نیازی به تغییرات فیزیکی در سخت افزار باشد. این ویژگی باعث می شود که سیستم ها بتوانند به طور خودکار با تغییر شرایط و نیازهای مختلف سازگار شوند و بهره وری بالاتری داشته باشند.

در این پروژه، هدف ما آشنایی بیشتر با این مفهوم و استفاده از بردهای آماده آردوینو برای پیاده سازی عملی آن است.

۱-۱ اهداف پروژه

۱. پیاده سازی ارتباط مؤثر بین برد و کامپیوتر.
۲. آشنایی مقدماتی با منطق قابل پیکربندی.
۳. ایجاد فرصت برای تجربه عملی و کار با برد های آردوینو و اجزای الکترونیکی مانند LED ها و دیپ سوئیچ ها.
۴. توسعه یک نرم افزار شبیه ساز با رابط کاربری گرافیکی برای طراحی و تست مدارهای منطقی.

۲-۱ تجهیزات و نرم افزارها

۱. برد آردوینو
۲. ۸ عدد سوییچ و ۴ عدد LED
۳. BreadBord و کابل USB
۴. نرم افزار IDE ARDUINO
۵. نرم افزار intelige برای زبان جاوا
۶. نرم افزار putty

۳-۱ روش کار

در ابتدا کار را به دو بخش سخت افزار و نرم افزار تقسیم بندی کردیم؛ دو نفر کار نرم افزار و منطق شبیه ساز را پیاده کردند و دو نفر بخش سخت افزار و ارتباط سریالی را پیاده کردند. همچنان در ابتدا با برد کار کرده و با عملکرد آن آشنا شدیم.

در بخش نرم افزار منطق کد را به گونه ای پیاده سازی کردیم که خروجی LED ها برای حالت های متفاوت از جداول استخراج و در برنامه ذخیره شود. همچنین با سریال این منطق را به برد منتقل می کنیم. دستورهای تعریف شده در این بخش:

new circuit:

با این دستور برنامه آماده دریافت یه مدار و منطق جدید می شود.

create table with "number of inputs" inputs and "number of outputs" outputs:

با این دستور یک جدول درستی با تعداد ورودی و خروجی مشخص شده جدید ساخته شده و به مدار اضافه می شود.

در خط بعدی این دستور باید المنت های جدول وارد شود که البته یکسری محدودیت دارد مثلا ایندکس in ها باید بین ۰ تا ۷ باشد و ایندکس c ها بین ۱ تا ۸ و ایندکس led ها ۰ تا ۳ باشد
یک مثال از فرمت مورد قبول :

in° in\c\led°

finish:

این دستور نشان دهنده این است که جداول درستی مدار به پایان رسیده است. همچنین پس از وارد کردن این دستور مدار به برد فرستاده شده و نرم افزار و برد آماده دریافت ورودی می شوند. همچنین بعد از این دستور فقط میتوان مدار جدید ساخت یا input وارد کرد (تا وقتی که دوباره مدار جدید ساخته شود)

input "binary string"

این دستور یک نمونه ورودی را به برنامه داده و خروجی متناسب با آن پرینت می شود. (ورودی حتما باید یک رشته باینری با طول بین ۱ تا ۸ باشد)
بعد از وارد کردن input ها خروجی led ها به فرمت زیر نمایش داده میشود :

LED° = on

LED\ = off

LED۲ = dont care

LED۳ = dont care

که روشن و خاموش بودن led ها را در بخش سخت افزار میتوانید مشاهده کنید.

exit:

با این دستور برنامه به پایان می رسد.

در بخش سخت افزار ابتدا برد را به سیستم وصل کرده و ورودی و خروجی دادن به برد را استفاده از نرم افزار IDE Arduino امتحان کردیم. سپس در IDE برد را برنامه نویسی کردیم و از طریق کابل USB آن را روی برد آپلود کردیم تا با توجه به منطق دریافت شده، متناسب با هر ورودی خروجی دهد.

در مرحله آخر دو بخش را با استفاده از سریال آردوینو بهم متصل کرده و منطق را از نرم افزار شبیه ساز به برد آردوینو منتقل کردیم.

۴-۱ چالش‌ها

- ۱- در بخش کدهای سطح بالا، یکی از چالش‌های پیش‌روی ما، ترتیب‌دهی و منسجم‌سازی جدول درستی‌ها به صورت یک جدول واحد بود.
- ۲- مدیریت ورودی‌ها و خروجی‌های غیرمستقیم (یعنی زمانی که خروجی یک جدول به عنوان ورودی جدولی دیگر استفاده شود) یکی از عوامل اصلی این پیچیدگی محسوب می‌شد.
- ۳- حفظ منطق مدار و توجه به حالت‌های `don't care` و کلیدهای `don't care` نیز از اهمیت بسزایی برخوردار بود.
- ۴- یک چالش دیگر مربوط به مدیریت شرایط متناقض با هم بود.
- ۵- ایجاد ارتباط سریالی بین نرم افزار جاوا و برد آردوینو و برقراری ارتباط صحیح بین پورت مورد نظر و برنامه‌ها از مسائلی بود که به دقت بالایی نیاز داشت.
- ۶- هندل کردن تایپ کستینگ مناسب برای داده مورد نظر پیش از سریال کردن داده نیز از بخش‌های حائز اهمیت بود.
- ۷- در بخش آپلود کد شبه C در برد، باید به طور بهینه از حافظه استفاده می‌کردیم چون مقدار حافظه‌ی آردوینو کم است و همچنین در صورت استفاده‌ی متعدد از تایپ `int` حافظه به مقدار زیادی میشد و گاهی استفاده از این نوع داده، باعث درگیری نامشخص و زیادی از حافظه می‌شد.
- ۸- به دلیل تفاوت روند دریافت ورودی توسط دیپ سوئیچ و یا ورودی دادن به طور مستقیم، هم در بخش جاوا و هم در بخش کد نویسی آردوینو، دارای چالش‌هایی بودیم که با تنظیم نقاطی برای شروع دریافت خواندن ورودی‌ها، این مشکل را حل کردیم.
- ۹- در فرستادن یک رشته از طریق سریال در سمت جاوا، چالش ما این بود که برای سریال کردن این دیتا، باید حتما در همان مرحله، ارتباطی با ترمینال جاوا با استفاده از کلاس `scanner` گرفته میشد تا بعد از آن، رشته نمایش دهنده‌ی مدار که در حافظه ذخیره شده بود ارسال شود.
- ۱۰- در دریافت رشته از طریق سریال در سمت آردوینو، باید این موضوع رعایت می‌شد که فقط یک بار تابع `Serial.readString()` فراخوانی بشود و رشته را ذخیره کند تا دفعات بعدی نیاز به خواندن دوباره‌ی آن رشته از روی سریال نباشد به این دلیل که این کار باعث می‌شود که برنامه فقط بتواند یک بار ورودی را دریافت کند و فقط همان خروجی را نشان دهد.
- ۱۱- در بخش آردوینو باگ‌هایی به وجود می‌آمد که به این دلیل بود که باید برای اطمینان از عدم بروز باگ‌های غیرقابل پیش‌بینی، عملیات‌های مربوط به دریافت دیتا از روی سریال، در بخش جدایی از کد‌هایی که صرفاً مربوط به برد هستند قرار می‌گرفتند.

۵-۱ نتیجه

در این پروژه، هدف اصلی پیاده‌سازی یک شبیه‌ساز مدارات منطقی با قابلیت طراحی، انتقال و اجرای زنده روی برد آردوینو بود. این پروژه به ما این امکان را داد تا با مفهوم منطق قابل پیکربندی مجدد آشنا شویم و عملکرد آن را در عمل مشاهده کنیم. از آنجا که این نوع منطق امکان بازآرایی سخت‌افزار را به صورت پویا فراهم می‌کند، تجربه‌ی کار با آن بر روی برد آردوینو به وضوح نشان داد که چگونه می‌توان از این تکنولوژی در پروژه‌های مختلف مهندسی استفاده کرد.

با طراحی و پیاده‌سازی یک نرم‌افزار شبیه‌ساز با توانستیم جداول صحت مختلف را به راحتی تعریف و تست کنیم. این نرم‌افزار به عنوان پل ارتباطی میان کامپیوتر و برد آردوینو عمل کرده و انتقال طراحی مدارات منطق به سخت‌افزار و بررسی عملکرد آن را تسهیل می‌کند. همچنین، کار با اجزای الکترونیکی مانند LED ها و دیپ سوئیچ‌ها باعث بهبود توانمندی‌های عملی در زمینه طراحی مدارات دیجیتال شد. در طول پروژه، با چالش‌هایی مواجه شدیم که شامل مشکلات نرم‌افزاری در ارتباط میان کامپیوتر و برد آردوینو و همچنین تطابق صحیح ورودی‌ها و خروجی‌ها بر روی سخت‌افزار بود. اما با استفاده از تست‌های مختلف و بهینه‌سازی کدها، توانستیم این مشکلات را برطرف کرده و عملکرد سیستم را به حالت مطلوب رساندیم.

این پروژه نه تنها به ما کمک کرد تا مهارت‌های طراحی و شبیه‌سازی مدارات منطقی را بهبود بخشیم، بلکه فرصتی برای تجربه عملی با سخت‌افزارهای آردوینو و مدارات دیجیتال فراهم ساخت. در نهایت، این پروژه قابلیت‌های بالقوه سیستم‌های مبتنی بر منطق قابل پیکربندی مجدد را در حوزه‌های مختلف مهندسی، از جمله سیستم‌های دیجیتال و پردازش سیگنال، به نمایش گذاشت.