

به نام خدا



پروژه مدار منطقی

دکتر سمیه کاشی

اعضای تیم

محمد صادق همتی

مهداد چراغی

تانیای رباطی

کلیت موضوع پروژه ما برنامه ریزی یک دستگاه رمزنگاری است که کاربر با ورود رمز خود وارد سیستم شده و میتواند رمز خود را عوض کند و در صورت نادرست بودن رمز وارد شده ، خطای مربوطه را نشان میدهد .

پروژه سه بخش اصلی دارد : صفحه کلید ، قطعه آردوینو ، صفحه نمایش یا ال سی دی

صفحه کلید قطعه ورودی است که کاربر با استفاده از دکمه های آن رمز خود را وارد میکند . به عبارتی این قطعه ارتباط بین کاربر و آردوینو است . ما از keypad_phone برای اینکار استفاده میکنیم .

آردوینو قطعه پردازش ما است که ما با استفاده از کد سی پی پی آن را برنامه نویسی میکنیم . اطلاعات وارد شده توسط کاربر از طریق صفحه کلید به این قطعه منتقل و پردازش روی اطلاعات انجام میشود و خروجی در ال سی دی نمایش داده میشود .
LCD همان دستگاه خروجی است .

ابتدا کتابخانه و IDE قطعه Arduino را نصب کردیم .

مهم ترین قطعه پروژه آردوینو است . این قطعه ارتباط میانی صفحه کلید و LCD است . نحوه ارتباط آردوینو با صفحه کلید به این صورت است که همه ی سیم های خروجی صفحه کلید به عنوان ورودی به آردوینو داده میشوند . در آردوینو هر کدام از این سیم ها با استفاده از کتابخانه keypad.h به یکی از درایه های آرایه دو بعدی مپ میشوند . به طور مثال وقتی کاربر کلید 5 را فشار میدهد ، سیگنالی از سطر 2 و ستون 2 به آردوینو میدهد که این سیگنال همان کلید 5 در آرایه دو بعدی متناظر است .

سیم های خروجی آردوینو به عنوان ورودی به صفحه نمایش داده میشوند که با استفاده از کتابخانه LiquidCrystal.h اعداد و حروف خروجی در نمایشگر نشان داده میشوند .

یک آرایه به عنوان رمز پیش فرض که در اینجا 0000 است میسازیم .

یک استک برای ورودی هایی که کاربر از طریق صفحه کلید میدهد ، میسازیم .

در ابتدا با استفاده از تابع display screen استرینگ enter pass را به کاربر نشان میدهد تا کاربر رمز خود را وارد کند . سپس برنامه وارد لوپ بینهایت میشود که هر لحظه ورودی صفحه کلید را چک میکند . در این لوپ اگر کاربر یکی از اعداد را وارد کند این عدد به استک push میشود . برای امنیت بیشتر اعداد وارد شده به صورت نامعلوم نمایش داده میشوند . اگر * وارد شود ، top را از استک pop میکند . اگر # وارد شود ، استرینگ وارد شده را از ورودی میخواند و با رمز اصلی چک میکند . اگر برابر بودند تابع unlock صدا زده میشود . اگر برابر نبودند تابع incorrect ، که این دو هرکدام به ترتیب کار های زیر را انجام میدهند :

تابع unlock نمایشگر را پاک میکند و استرینگ Access Granted را نمایش میدهد .

بعد از 1 ثانیه دو آپشن changepass و exit روی نمایشگر نشان داده میشود که کاربر با ورود عدد 1 یا 2 یکی از این ها را انتخاب میکند . اگر عددی به جز این دو وارد شود خطای Wrong Input را میدهد و دوباره دو آپشن فوق را به کاربر نشان میدهد .

تابع incorrect صفحه نمایش را پاک میکند ، به کاربر code incorrect را نشان میدهد و دوباره صفحه نمایش را برای ورود رمز آماده میکند .

اگر کاربر exit را انتخاب کند صفحه نمایش دوباره برای وارد کردن رمز آماده میشود .

اگر change pass را انتخاب کند ، هر کلیدی که کاربر وارد میکند اگر # نباشد و سائز استک کمتر از 4 باشد ، به استک push میشود . اگر کاربر * را وارد کند ، عددی که قبلش وارد کرده یا همان top از استک pop میشود . اگر سائز استک برابر 4 شد ، حلقه break میشود و اعداد وارد شده در آرایه رمز فرعی ذخیره میشوند . سپس ال سی دی استرینگ confirm pass را نمایش میدهد . کاربر دوباره رمز جدید را وارد میکند و آردوینو مطابق بالا عمل میکند . سپس بررسی میکند اگر رمزی که در آرایه رمز فرعی ذخیره شده با رمزی که کاربر دوباره وارد کرده برابر باشد ، آن رمز را به عنوان رمز اصلی در آرایه رمز اصلی میریزد و password changed را به کاربر نشان میدهد . در غیر این صورت mismatch روی صفحه نمایشگر نشان داده میشود . در نهایت هم دوباره صفحه نمایشگر را برای ورود رمز آماده میکند .

یکی از مشکلات اصلی که در هنگام انجام پروژه داشتیم ، پیاده سازی * برای پاک کردن یک رقم از رمز وارد شده بود که نتیجه آن استفاده از استک برای نگه داری مقادیر ورودی توسط کاربر بود .

دیگری این بود که ابتدا کاربر را به وارد کردن چهار رقم محدود کرده بودیم و هر رقمی که وارد میشد اگر اشتباه میبود همانجا به کاربر گفته میشد رمز شما اشتباه است . که این باعث به خطر انداختن امنیت میشد چون با این روش ما $4 * 10$ یعنی 40 جایگشت برای پیدا کردن رمز آن فرد داشتیم . اما ما این مشکل با استفاده از # رفع کردیم . به طوری که به کاربر اجازه دادیم هر چند رقم میخواهد رمز وارد کند و سپس وقتی # را فشرد ، بررسی شود که آیا رمز درست است یا نه . این باعث شد از 40 جایگشت حداقل به 10000 جایگشت برای پیدا کردن رمز شخص نیاز باشد .