

Knock Lock

Team Pegasus:

Kasi Reddy Sreeman Reddy, Himanshu Rajpoot & Kaushik Singirikonda

Our main goals:

1. To implement a knock pattern as password using a piezo sensor. From the knock pattern we generate a string which contains ratios of time intervals between successive knocks. For example a string (100,50,25,100,...) means we should knock 5 times with 2nd time interval half the 1st and so on.
2. To implement a password changing mechanism using a push button. We should hold the button and give the new password as input and then leave the button.

Bonus goals achieved apart from the above:

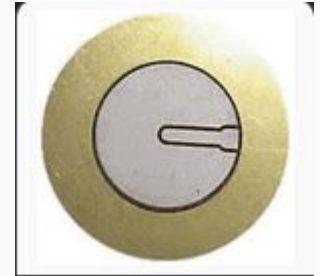
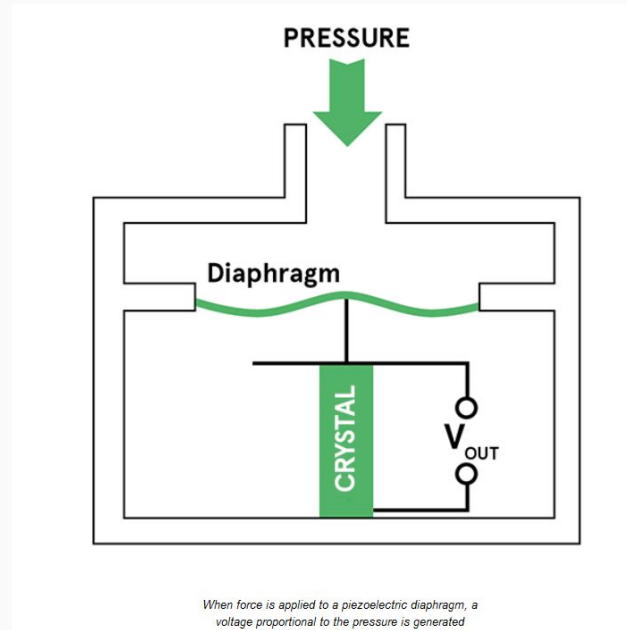
1. We implemented a standard password safety trick. If incorrect password is given 5 times, then it will stop for 30 seconds and after another 5 incorrect password it will wait for another 60 seconds and after another 5 incorrect passwords will wait for another 90 seconds and so on.
2. We implemented an LCD 16x2 display to show several things like “Voila, enter now!”, “Wrong password”, “Password changed” and a timer to wait after many incorrect attempts.

Sensor

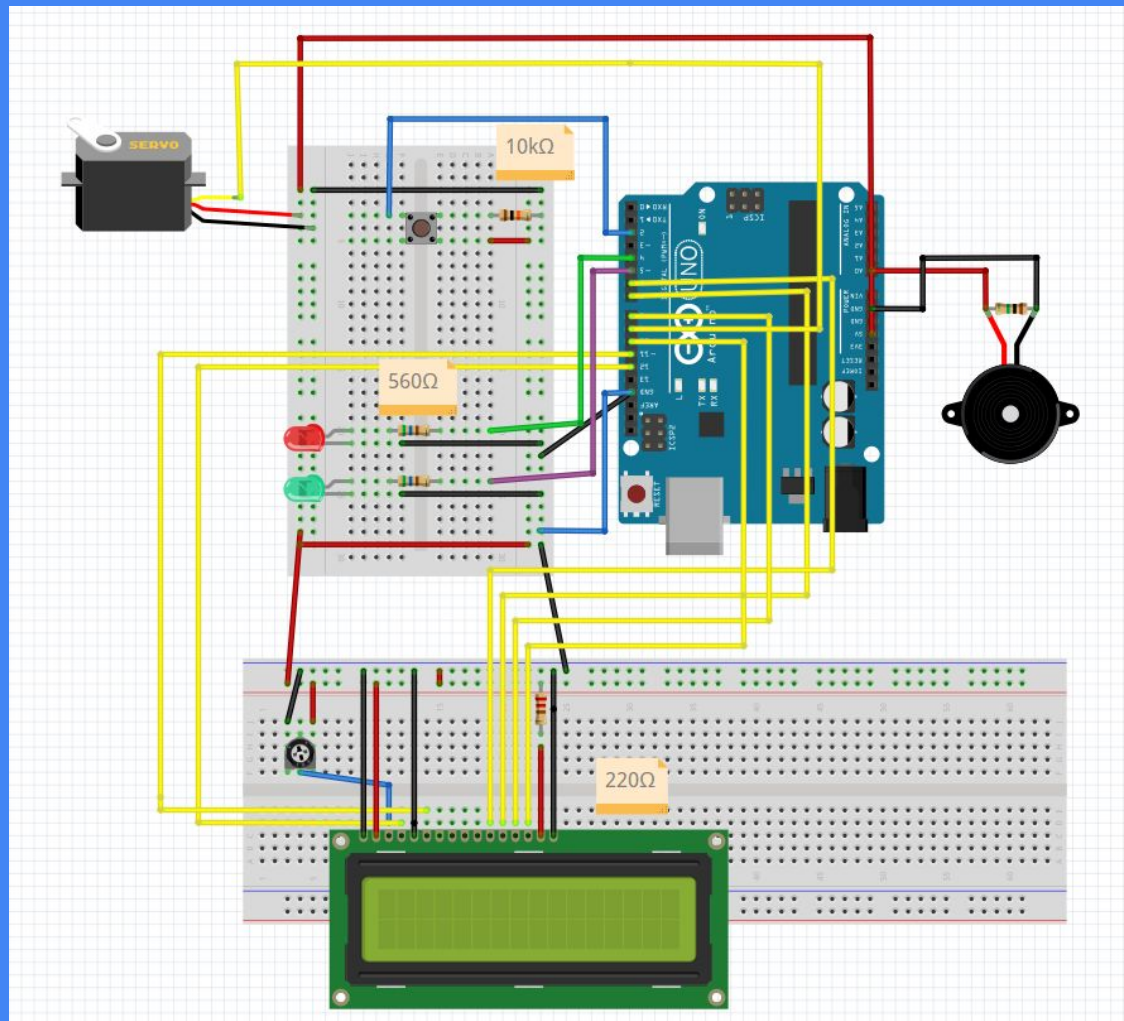
Piezo Sensor (knock detection)

A piezo sensor (piezoelectric sensor) is used to convert pressure into voltage, this can be utilized for knock detection.

This is highly sensitive to knock detection, and can be placed on the opposite side of a wooden door (which generally have good acoustic insulation).



Circuit Diagram



Implementation

Libraries Used:

1. Servo.h
2. LiquidCrystal.h

Functions:

1. Knock_listener
2. unlock_Door
3. Is_password_correct()

Limitations:

1. If a physical door is very thick then the knock will not be detected if it's on the other side, so we need to use a driller to make a hole and keep the piezo sensor on the opposite side.
2. Every time we tap the knock pattern anyone can hear our password and due to this our security will be less than something like a fingerprint password.

Lessons learnt:

The piezo sensors were more sensitive and precise than we expected them to be. It was harder for humans to implement exactly the same pattern of time intervals always. So we reduced the accuracy in our code such that it will accept reasonable inputs as correct.

Demo Link:

https://drive.google.com/file/d/1FF1W9Ac_JThznk2XX30bzoJaLrbdIZLo/view?usp=sharing

Warning

Please DO NOT try this at home.
Don't blame us if you get
robbed.



Thanks!

