

Smart Access Control System

Introduction

The project builds an electronic lock system. It manages access to a secure container using a digital interface. The project uses an Arduino Nano to process user inputs and control a DC motor mechanism.

Function: It reads button presses and compares the sequence to a stored code. Goal: Create a compact, electronic alternative to physical padlocks. Idea: The concept originated from observing analog security boxes. Digital systems offer more flexibility. Utility: This system secures personal items in lockers or small safes. It teaches users about voltage dividers and motor control.

General Description

The system consists of four primary modules. The user interacts with the input module. The processing module evaluates data. The feedback and actuation modules execute the final response.

Input Module: Two buttons connected to one analog pin. They use a voltage divider to send different signals.

Processing Module: An Arduino Nano. It monitors voltage levels and runs the logic.

Feedback Module: An I2C LCD screen and a buzzer. These provide visual and audio status updates.

Actuation Module: A DRV8833 driver and an N20 motor. This hardware moves the locking bolt.

Hardware Design

The design focuses on efficiency. It uses a single analog pin for multiple buttons to save resources.

Parts List: Arduino Nano (ATmega328P) I2C LCD 1602 Display DRV8833 Dual Motor Driver N20 DC Gear Motor Passive Buzzer 10k Ω Resistor 2.2k Ω Resistor Breadboard and jumper wires 9V Battery or 6V battery pack

Electrical Schematic Logic: The buttons use a voltage divider. The output voltage V_{out} depends on which button connects to the circuit.

$$V_{out} = V_{in} \cdot \frac{R_{pulldown}}{R_{series} + R_{pulldown}}$$

Software Design

The firmware handles real-time input monitoring. It uses the PlatformIO development environment.

Libraries: LiquidCrystal_I2C.h for display and Wire.h for communication.

Algorithm: The code performs

a loop to check the analog value of pin A0.

```
Value ~1023: No button pressed.  
Value ~0: Button 1 pressed.  
Value ~184: Button 2 pressed.
```


Logic: Successful code entry triggers a high signal to the DRV8833. The motor runs for 2000 milliseconds.

Rezultate Obținute TODO

Care au fost rezultatele obținute în urma realizării proiectului vostru.

Concluzii

Download

O arhivă (sau mai multe dacă este cazul) cu fișierele obținute în urma realizării proiectului: surse, scheme, etc. Un fișier README, un ChangeLog, un script de compilare și copiere automată pe uC crează întotdeauna o impresie bună .

Fișierele se încarcă pe wiki folosind facilitatea **Add Images or other files**. Namespace-ul în care se încarcă fișierele este de tipul **:pm:prj20??:c?** sau **:pm:prj20??:c?:nume_student** (dacă este cazul).
Exemplu: Dumitru Alin, 331CC → **:pm:prj2009:cc:dumitru_alin**.

Jurnal

Puteți avea și o secțiune de jurnal în care să poată urmări asistentul de proiect progresul proiectului.

Bibliografie/Resurse

Listă cu documente, datasheet-uri, resurse Internet folosite, eventual grupate pe **Resurse Software**

și Resurse Hardware.[Export to PDF](#)

From:

<http://ocw.cs.pub.ro/courses/> - **CS Open CourseWare**

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2026/bianca.popa1106/robert.springer>Last update: **2026/05/04 12:49**