

Masina de cocktail-uri

Nume: Popa George-Matei

Grupa: 331CB

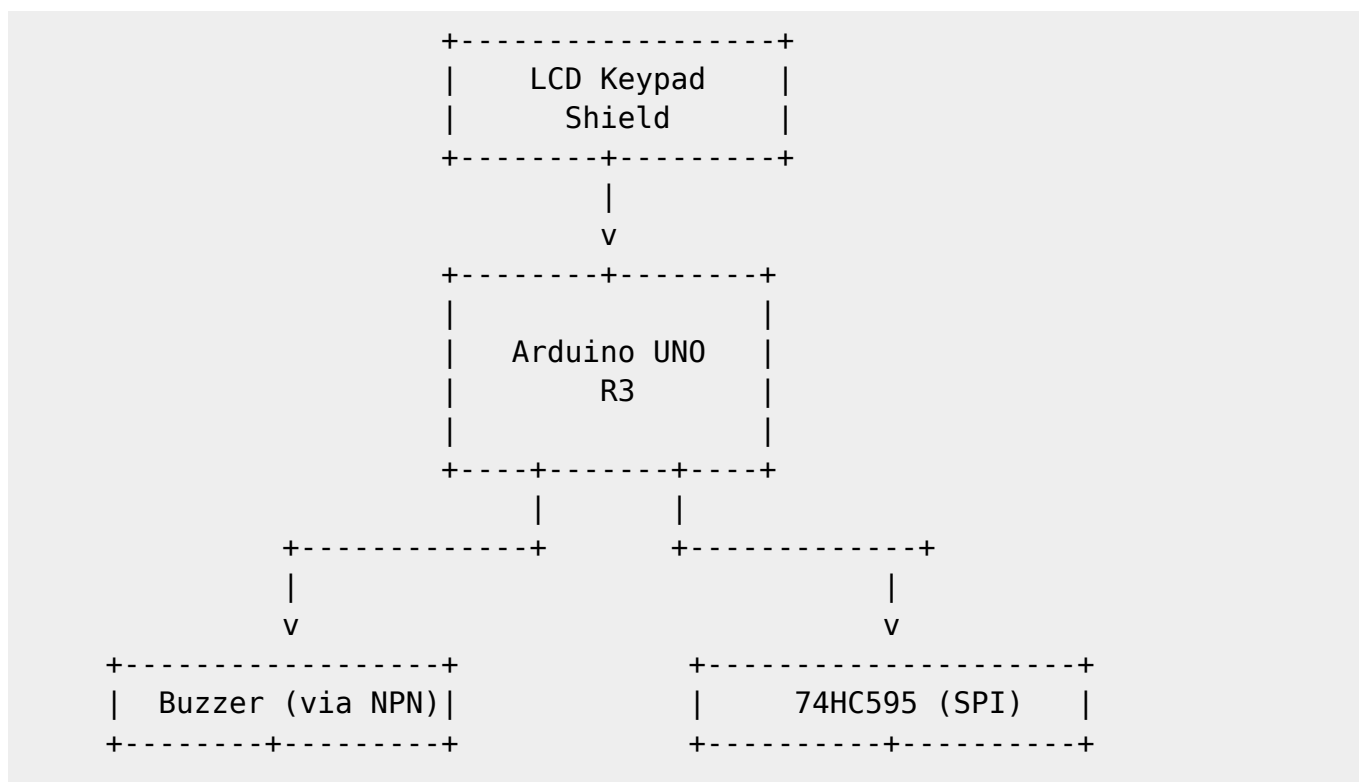
Asistent: Alex Văduva

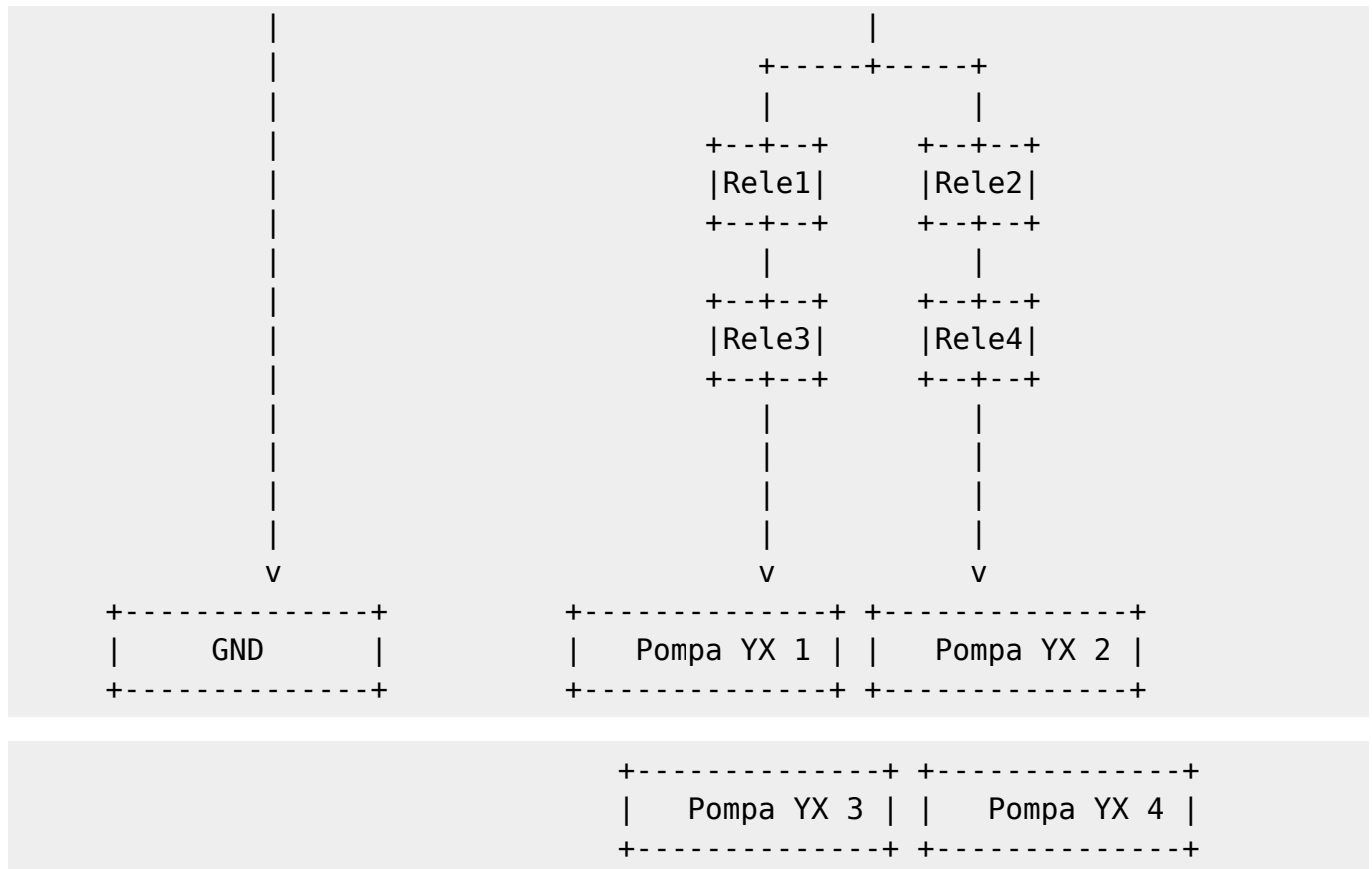
Introducere

My project consists of an automatic cocktail-making machine, based on the Arduino Uno R3 board and equipped with four pumps controlled via relays, an LCD keypad shield for selecting recipes, and a buzzer that signals and enhances the atmosphere with music. My goal was to automate the precise dosing of ingredients to ensure drink consistency and to create an interactive experience. The idea came to me at a party, when I was trying to make myself a drink and couldn't get the proportions right — that's when I thought it would be much easier if there were a machine to do it for me. I believe the project is useful both for parties and for home bars, offering an interactive and precise way of serving drinks.

Descriere generală

Schema bloc





Hardware Design

Hardware component list:

- Pump YX DC12V 385 (x4)
- Relay Module (x4)
- Arduino UNO R3 (x1)
- LCD keypad shield (x1)
- Buzzer (x1)
- NPN transistor
- 1k ohm resistor
- Switch
- Type C Plug
- Perfboard

YX DC12V 385 Pumps (x4)

- These are connected to the outputs of the relays and are responsible for distributing the liquids.
- Power is shared among all four pumps (red wire for VCC and black wire for GND).
- They are individually activated through the 4 relay modules.

Relay Modules (x4)

- These are isolated modules used to switch high-power loads using 5V signals from the Arduino.

- The relays' GND is common with the Arduino GND.
- Signal pins (IN1-IN4) are connected to digital pins on the Arduino.

Arduino UNO R3 + LCD Keypad Shield

- The main controller.
- The LCD with integrated buttons allows user interaction (selecting recipes, starting distribution, etc.).
- The LCD uses pins 4-10, while the buttons use an analog pin to detect presses through an internal voltage divider.

Buzzer

- Used to produce a sound at the beginning/end of distribution, for error alerts and playing ambient music.
- Connected to a digital pin via an npn transistor.

Switch

- Connected between GND and a digital pin with the internal pull-up resistor enabled.
- Used to start/stop the machine.

Type-C Plug

- Used to power the Arduino from a socket.

Perfboard

- Used for more secure and distributed connections (GND, VCC, etc.).
- Helps organize the circuit and provides more stable soldered connections.

Software Design

1. Cocktail Machine Controller setup()

- Initializes the LCD (16x2) and SPI interface for shift register control.

Configures the output pins:

- latchPin (for 74HC595 control),
- buzzerPin (for sound output, via PWM),
- Sets initial output state of pumps to OFF (0xFF).
- Begins serial communication for debugging (Serial.begin(9600)).
- Initializes the melody loop timer with noteStartTime = millis().

loop() Continuously plays a looped melody in the background (playLoopedMelody()).

Displays and navigates the Main Menu via menu():

- Option 1 - Have a drink: Select a predefined cocktail and desired strength.
- Option 2 - Make your own: Manually select pump volumes.
- Option 3 - Toggle Music: Enable or disable background melody.

Executes drink preparation logic depending on the selected menu page.

Main Menu Navigation: menu() Displays a list of options using the LCD.

Uses analog button input (button_read()) to navigate LEFT/RIGHT and SELECT.

Plays melody during menu interaction.

Sets the menu_page based on user selection:

- Page 1 = predefined drinks.
- Page 2 = custom drink creation.

Predefined Drinks Mode (Page 1) Drink Selection: Scroll through predefined cocktail options:

- Rum & Coke
- Long Island
- Raspberry Vodka

Uses button input to change selection and confirm with SELECT.

Strength Selection:

- Adjust strength (1 to maxstrength) using LEFT/RIGHT.
- Displays strength visually using repeated ++ symbols.

Execution: Calls DrinkSelect(choice, strength):

Prompts user to assign pumps for Rum and Coke.

Computes ingredient volumes:

- Rum = 40 + strength * 10 ml
- Coke = 40 ml

Calls Pump_drive(...) to activate pumps accordingly.

Custom Drink Mode (Page 2) For each pump (1 to 4), user can manually assign volume in ml.

Increments/decrements volume in 20ml steps via button input.

After configuration, calls Pump_drive(...) with custom values.

Melody Control: playLoopedMelody() Loops through an array of notes (melody) and plays them on the buzzer using tone().

Uses millis() for non-blocking delays.

Stops playback if musicOn == false.

Pump Control & Shift Register shiftOut(byte data): Sends a byte to 74HC595 via SPI to update pump states.

activatePump(pump_index): Activates a single pump by setting corresponding bit LOW in current_output_state and updating via shiftOut().

deactivatePumps(): Turns off all pumps by sending 0xFF.

Pump_drive(p1, p2, p3, p4):

For each non-zero amount:

Displays which pump is active.

Calculates duration based on flow rate (20 ml/s).

Activates the pump for the calculated time, playing the melody during operation.

Turns off all pumps after dispensing.

Analog Button Input: button_read()

Reads analog value from A0.

Maps the value to:

- RIGHT, UP, DOWN, LEFT, SELECT, or NONE (debounced).

Adds a delay to avoid bounce/misreads.

Ingredient Assignment: DrinkSelect(choice, strength) Prompts the user to select which pumps to use for Rum and Coke.

Adjusts amount[] array based on strength and recipe.

Executes pump drive for the selected ingredients.

Rezultate Obținute

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/prj2025/avaduva/george_matei.popa

Last update: **2025/06/01 17:09**

