Tick-food-tock

Nume: Butacu Laura-Diana

Grupa: 331CB

Introducere

Tick-food-tock este un sistem interactiv de gestionare a timpului necesar pentru prepararea rețetelor cu precizie și siguranță. Inspirat de nevoia de a evita accidentele provocate de erori de temporizare sau distragera atenției atunci când gătim, acest dispozitiv oferă o modalitate simplă de a seta și urmări timpul de preparare. Cu afișare digitală, avertizări vizuale și audio, precum și un sistem de detectare a fumului și dispersare a acestuia, proiectul își propune să asigure un mediu de gătit mai sigur și să ofere utilizatorilor o experiență mai plăcută în bucătărie, pentru a se putea bucura în final de preparatele lor.

Descriere generală

Laptop-ul personal va asigura prin cablu USB tensiunea necesară de 5V plăcuței Arduino. Sistemul are în componența sa un buton prin care se accesează modurile de setup: cu valori prestabilite pentru anumite preparate sau custom. În stadiul custom, utilizatorul are posibilitatea de a alege, prin intermediul altui buton, unitatea de măsură a timpului pentru care setează valori cu butoanele de ”+” și ”-”: ore, minute sau secunde. Countdown-ul este afișat pe ecranul LCD și începe după ieșirea din modul de setup și apăsarea butonului de “start/stop”. Trecerea timpului este semnalată și vizual prin intermediul unui LED care își ajustează culoarea în funcție de timpul rămas, dar și de sunetul “tick-tock” al buzzer-ului. De asemenea, în cazul în care utilizatorul se răzgândește, este posibilă setarea timpului cu o nouă valoare și reînceperea countdown-ului prin apăsarea butonului “start/stop”.

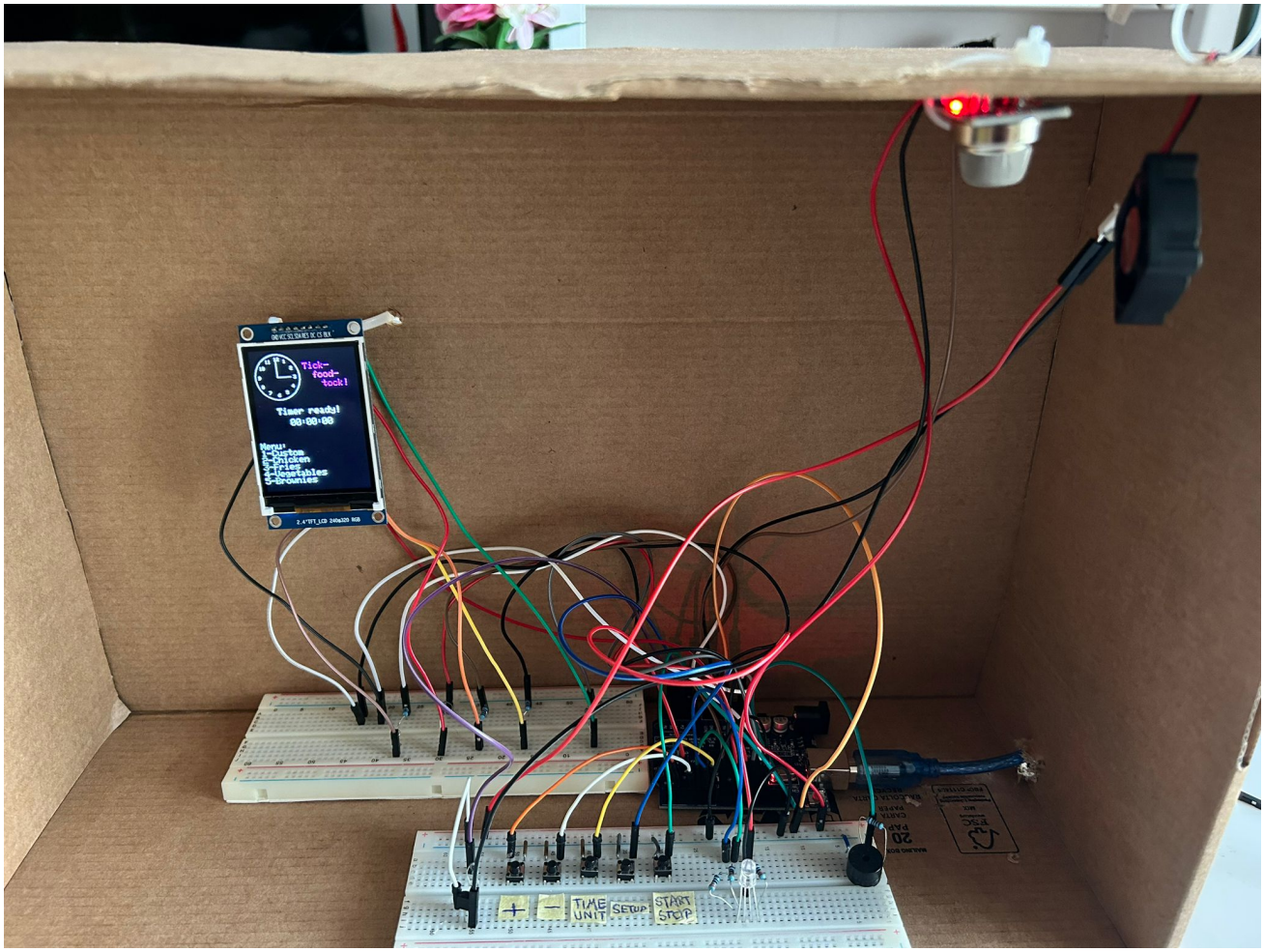
Când timpul s-a scurs, buzzer-ul va suna iar LED-ul va lumina verde intermitent pentru a atenționa utilizatorul că mâncarea este gata. Poftă bună!

Ca măsură de siguranță în cazul în care a fost setată o durată greșită și preparatele s-ar arde, senzorul disponibil va detecta fumul și ventilatorul va începe să-l împrăștie, timpul este automat oprit, LED-ul va lumina roșu intermitent și buzzer-ul va suna pentru a alerta în acest sens. Pentru a reveni în modul inițial al timer-ului, se apasă butonul “start/stop”.

Hardware Design

- Arduino UNO (ATmega328p)
- Modul LCD de 2.4'' cu SPI și Controller ILI9341 (240×320 px)
- Senzor fum MQ-2
- Ventilator 5V
- Breadboard
- Led RGB 5 mm cu catod comun
- 5 butoane - Push Buttons
- Buzzer pasiv
- Fire de legătură
- Rezistoare
- Tranzistor NPN 2N2222

Implementarea hardware după schema electrică:



Se poate observa funcționalitatea ecranului (**SPI**), care a necesitat la conectare rezistențe de 10k la pinii digitali **8, 9, 10, 11, 12, 13** și alimentare de 5V. Led-ul RGB cu catod comun l-am conectat la pinii Arduino pentru PWM: **3, 5, 6** cu rezistențe de 220 pentru a-i putea controla ulterior culoarea în funcție de trecerea timpului. Cele cinci butoane le-am conectat atât la pini digitali (**2, 4, 7**) cât și la pini analogici (**A2, A3**) pentru a utiliza întreruperi. Buzzer-ul l-am înseriat cu o rezistență de 100 pentru a obține un sunet mai fin și l-am conectat la **A1**. Pentru a putea controla ventilatorul, l-am conectat cu un tranzistor NPN: GND la colector, alimentarea la 5V, emițătorul tranzistorului la GND și

baza înseriată cu o rezistență de 220 și conectată la pinul **A4**. În final, senzorul MQ-2 funcționează cu alimentare de 5V, iar pentru a detecta fumul citesc valoarea de pe **A0**.

Software Design

Am dezvoltat codul în IDE-ul Arduino și am folosit următoarele biblioteci:

- SPI.h: pentru comunicarea prin protocolul SPI cu ecranul LCD
- Adafruit_GFX.h și Adafruit_ILI9341.h: pentru scrierea textului și desenare pe ecran
- MQ2.h: pentru inițializarea senzorului de fum și citirea valorilor detectate
- TimeLib.h: pentru a contoriza trecerea timpului în cadrul countdown-ului
- PinChangeInterrupt.h: pentru întreruperi

Pe parcursul programului, am considerat că timer-ul se poate afla într-unul din cele 9 moduri prezentate mai jos și, în funcție de acest mod, în loop() gestionez apăsarea butoanelor, trecerea timpului și afișarea mesajelor specifice pe ecran:

- MODE_IDLE: modul principal al timer-ului, în care se află la începutul programului, înainte de începerea timpului sau în cazul în care se dorește resetarea acestuia cu o nouă valoare
- MODE_SETUP_CUSTOM: varianta de setare a timpului cu valori customizabile, unde utilizatorul poate alege numărul de ore, minute și secunde dorite
- MODE_CHICKEN_TENDERS, MODE_FRIES, MODE_VEGETABLES, MODE_BROWNIES: moduri în care timpul pentru preparatele respective este prestabilit
- MODE_RUNNING: timer-ul a început și se contorizează trecerea timpului
- MODE_RINGING: timer-ul s-a terminat și utilizatorul este atenționat
- MODE_SMOKE: în cazul în care se detectează fum din cauza arderii preparatelor, utilizatorul este atenționat

În **setup()** pornesc mai întâi interfața serială pentru comunicarea cu ecranul LCD și aștept calibrarea senzorului. Se afișează un mesaj când este gata și apoi, cu ajutorul funcțiilor din biblioteca LCD-ului, desenez ceasul “Tick-food-tock”. Stabilesc modul de utilizare pentru pinii LED-ului, buzzer-ului, ventilatorului ca OUTPUT și pe cel din urmă îl setez și pe LOW ca să fie oprit, iar pentru butoane activez întreruperile și le configurez cu INPUT_PULLUP. În handler-ul întreruperilor, marchez butonul respectiv ca fiind apăsat prin setarea cu “true” a variabilei bool corespunzătoare, care va deveni ulterior “false” la eliberarea acestuia și fac debouncing pentru a evita citirile multiple.

Funcția de **loop()** este compusă din trei părți principale unde configurez fiecare mod în parte:

1. Mode management
2. Time management
3. LCD print

Mode management: verific starea butoanelor.

- MODE_IDLE:

Dacă este apăsat butonul **setup**, se trece în MODE_SETUP_CUSTOM.

Dacă este apăsat butonul **start/stop**, se trece în MODE_RUNNING, începe timer-ul, se aprinde LED-ul roșu și se setează parametrii pentru proporția culorii.

- MODE_SETUP_CUSTOM:

Dacă este apăsat butonul **start/stop**, se revine în MODE_IDLE și se setează timpul cu valorile configurate.

Dacă este apăsat butonul **setup**, se trece la următorul mod: MODE_CHICKEN_TENDERS.

Când este apăsat butonul **timeUnit**, se trece prin unitățile de măsură disponibile: ore, minute și secunde.

Când sunt apăsate butoanele **+** și **-** se actualizează orele, minutele și secundele respectiv. Ora poate fi setată în range-ul 0-9 (atunci când se ajunge la 9 și se apasă ”+”, devine 0 și când este 0 și se apasă ”-” devine 9), iar minutele și secundele în 0-59 (atunci când se ajunge la 59 și se apasă ”+” devine 0 și când este 0 și se apasă ”-” devine 59).

- MODE_CHICKEN_TENDERS, MODE_FRIES, MODE_VEGETABLES, MODE_BROWNIES:

Când este apăsat butonul **start/stop**, se revine în MODE_IDLE și se setează timpul cu valorile prestabilite.

Când este apăsat butonul **setup**, se trece la următorul mod de configurare. Ordinea este: custom → chicken_tenders → fries → vegetables → brownies → custom.

- MODE_RUNNING:

Când este apăsat butonul **start/stop**, utilizatorul dorește oprirea timer-ului, așa că se revine în MODE_IDLE cu timpul configurat anterior și se sting LED-ul și buzzer-ul.

- MODE_RINGING:

Când este apăsat butonul **start/stop**, se revine în MODE_IDLE, se resetează timer-ul și se opresc LED-ul și buzzer-ul.

- MODE_SMOKE:

Când este apăsat butonul **start/stop**, se revine în MODE_IDLE, se resetează timer-ul și se opresc LED-ul, buzzer-ul și ventilatorul.

Time management: operații ce se întâmplă în timpul countdown-ului.

- MODE_RUNNING: se actualizează timpul curent rămas. Când s-a terminat, se trece în MODE_RINGING. Se actualizează iluminarea LED-ului în funcție de raportul dintre timpul curent și timpul de setup pentru a se face trecerea de la roșu la verde cu valori între 0 și 255. Pe măsură ce trec secundele, buzzer-ul scoate și un sunet “tick-tock”. Senzorul citește valori și în momentul în care găsește o valoare semnificativ mai mare decât cea de calibrare, înseamnă că a detectat fum și se trece în MODE_SMOKE.
- MODE_RINGING: LED-ul luminează verde intermitent și buzzer-ul scoate un sunet pentru a atenționa utilizatorul că preparatele sunt gata.
- MODE_SMOKE: LED-ul luminează roșu intermitent, se aprinde ventilatorul pentru a împrăștia fumul și buzzer-ul scoate un sunet țipător.

LCD print: afișez mesaje sugestive pentru fiecare mod în parte.

- MODE_IDLE: “Timer ready!” cu timpul setat și meniul.

- MODE_SETUP_CUSTOM: “Setup mode custom”, unitatea de măsură a timpului la setarea curentă și modificările timpului.
- MODE_CHICKEN_TENDERS, MODE_FRIES, MODE_VEGETABLES, MODE_BROWNIES: “Setup mode -” și timpul prestabilit.
- MODE_RUNNING: “Counting down..” și timpul rămas.
- MODE_RINGING si MODE_SMOKE: mesaje de informare și atenționare.

Rezultate Obţinute

În final, am reușit să implementez ce mi-am propus. Mai jos adaug un demo în care se poate observa modul de funcționare al timer-ului, precum și situația în care s-ar detecta fum:

https://www.youtube.com/watch?v=MpRps1iIb4I

Concluzii

Pe parcursul dezvoltării, am întâmpinat probleme clasice (cum ar fi debounce pentru butoane, fire nefuncționale, breadboard nu prea calitativ), dar cu siguranță dificultatea cea mai mare a fost în utilizarea ecranului LCD, atât din punct de vedere hardware, pentru că am avut niște probleme recurente la realizarea conexiunilor (pinii piesei au fost denumiți diferit), cât și din punct de vedere software, întrucât biblioteca nu dispune de o funcție de “clear text” și a necesitat ca în fiecare frame să fac fill peste textul pe care îl voiam șters, ceea ce a durat mult mai mult decât m-aș fi așteptat.

Cu toate acestea, experiența proiectului a fost una interesantă și diferită (într-un sens bun) față de ce am implementat până acum, iar după ce am înțeles cum funcționează lucrurile, pot spune că am lucrat cu plăcere la el și că satisfacția pe care o simți atunci când vezi rezultatul final și știi că ai realizat ceva de care poți să te folosești în viața de zi cu zi nu se compară cu nimic altceva.

Download

O arhivă ce conține codul sursă:

tick-food-tock.zip

Jurnal

- 04.05.2024: Realizarea paginii proiectului: introducerea, descrierea generală, schema bloc, hardware design
- 11.05.2024: Verificarea funcționalității pieselor comandate
- 12.05.2024: Probleme cu conectarea LCD-ului

- 15.05.2024: Fixed ^ 😀
- 16.05.2024: Adăugare poză cu implementarea hardware și explicații + scrierea unui cod minimal pentru evidențierea funcționalității unor componente
- 20.05.2024: Scrierea codului pentru structura timer-ului și butoane, probleme la apăsarea butoanelor și afișarea pe LCD
- 21.05.2024: Adăugare întreruperi + debouncing, cod pentru ventilator, LED. Realizarea meniului complet, al sunetului de tick-tock la scurgerea timpului și introducerea modulului pentru detecția fumului
- 22.05.2024: Realizarea afișării pe ecranul LCD în loc de interfața serială
- 23.05.2024: Desenarea ceasului pe ecran și ultimele retușuri legate de cod și afișare
- 24.05.2024: Realizarea cutiei si design-ului exterior final al proiectului 😀

Bibliografie/Resurse

Resurse software:

- https://cdn-learn.adafruit.com/downloads/pdf/adafruit-gfx-graphics-library.pdf
- https://github.com/harshkzz/Arduino-with-MQ2-Sensor-and-Serial-Monitor
- https://github.com/adafruit/Adafruit_ILI9341
- https://www.circuitbasics.com/how-to-use-active-and-passive-buzzers-on-the-arduino/
- https://www.hackster.io/doug-domke/pulse-width-modulation-and-leds-7d7a21
- https://docs.arduino.cc/built-in-examples/digital/Debounce/
- https://dronebotworkshop.com/interrupts/

Resurse hardware:

- https://www.hackster.io/techmirtz/using-common-cathode-and-common-anode-rgb-led-with-arduino-7f3aa9
- https://robu.in/mq2-sensor-interfacing-with-arduino-gas-and-smoke-detection/
- https://www.instructables.com/Interfacing-Buzzer-to-Arduino/
- https://www.instructables.com/Arduino-Button-with-no-resistor/

Export to PDF