


How fast can you chug?

Autor: Nedelcu Stefan Marian

Grupa: 331CB

Introducere

Proiectul constă într-un dispozitiv care măsoară cât de repede poate o persoană să golească un pahar, făcând petrecerile mai amuzante 

Descriere generală

Dispozitivul este compus dintr-un senzor de greutate pe care va fi pus un pahar. În momentul în care paharul este ridicat, pornește un cronometru (afișat pe 7 segmente). Când paharul este așezat la loc, cronometrul se oprește indicând astfel cât de repede a fost golit acesta. Pentru a fi resetat cronometrul se apasă un buton.



Hardware Design

Lista piese:

Nume piesa	Numar
Arduino uno	1
Butoane	1
Afișaj 4 digits - 7 segmente	1
Senzor greutate	1
Breadboard mini	1
Baterie 9V	1
Rezistente	
Fire	



Device-ul se folosește de un afișaj led de 4 caractere pe 7 segmente și anod comun (nu l-am găsit în Tinkercad pt. a fi reprezentat pe schemă).

Acesta folosește 13 pini: 7 pentru cele 7 segmente, unul pentru punctul de după cifră, 4 pentru a selecta pe care cifră se efectuează scrierea și unul pentru alimentare.

Proiectul este alimentat folosind o baterie de 9V prin pinul Arduino Vin (pentru o mobilitate mai mare-nu trebuie conectat în permanentă la un PC).

Software Design

Pe parcursul dezvoltării proiectului am folosit Arduino IDE. Am folosit două biblioteci auxiliare:

1. "TimerOne.h"-folosit pentru a crea o întrerupere ce ajută la implementarea timerului
2. "SevenSegmentDisplay.h"-folosit pentru facilitarea afișării caracterelor pe afișajul cu 7 segmente

Implementarea se folosește de 2 variabile cheie:

- timer: acesta este incrementat la fiecare apel al întreruperii
- isTimerRunning: reține dacă timerul rulează sau este oprit

Timerul este implementat cu ajutorul bibliotecii "TimerOne.h", alegând să creez o întrerupere care se va apela la fiecare 0.01s-în care, dacă timerul ar trebui să ruleze, se incrementează.

În funcția *setup()* se inițializează toți pinii (cei 12 ai displayului ca output iar cel corespunzător senzorului de greutate și a butonului de reset ca INPUT_PULLUP).

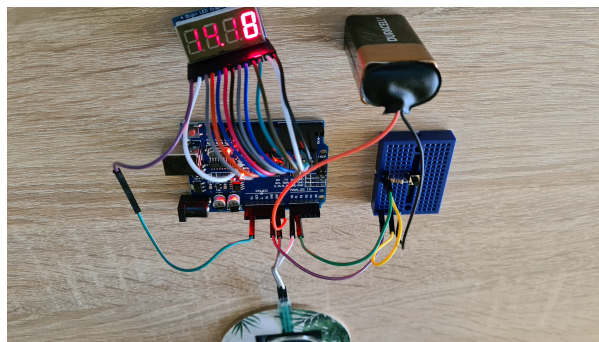
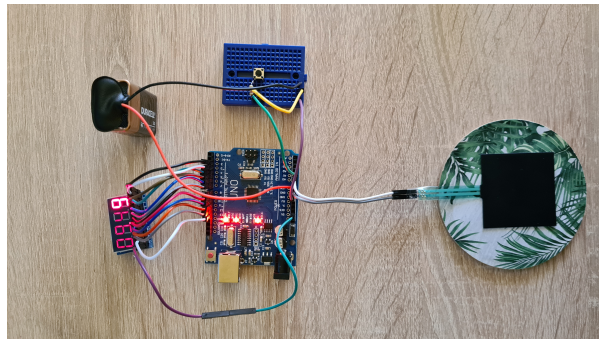
În funcția *loop()* se determină valoarea care trebuie afișată pe timer (fiecare cifră a lui time drept char) și se afișează folosind biblioteca "SevenSegmentDisplay.h". Aceasta se ocupă doar de afișajul unui singur digit (folosind funcția `timerDisplay.displayCharacter©`-din acest motiv fiind nevoie de următoarele funcții auxiliare:

- `pickDigit(x)` → selectează cifra care trebuie afișată la un moment dat (x putând lua valori între 1 și 4 reprezentând cifra aleasă), setând pinul corespunzător ei pe LOW iar pe ceilalți pe HIGH
- `clearLEDs()` → stinge toate Ledurile cifrei selectate la momentul respectiv, pentru a mă asigura că nu există conflicte în afișaj.

De asemenea, în *loop*, în funcție de valoarea citită de pe senzor se setează variabila `isTimerRunning` (dacă senzorul este apăsat timerul stă pe loc, iar dacă nu este apăsat atunci timerul curge). Când este apăsat butonul reset, variabilei timer îi este atribuită valoarea 0.

Rezultate obtinute

[Video demonstrativ](#)



Concluzii

Proiectul a fost o experienta hands-on foarte buna, din care am invatat mai multe despre placuta Arduino si modul de lucru cu ea. Am aflat cum functioneaza un afisaj cu 7 segmente, mai ales cu mai multe caractere, experimentand in acelasi timp cu diverse biblioteci care faciliteaza lucrul cu acesta.

De asemenea, am exersat mai mult lucrul cu intreruperi si timere.

Download

[Cod sursa](#)

Jurnal

25.04.2021 : Alegere tema, completare introducere si descriere generala.

15.06.2021 : Cumpararea tuturor pieselor necesare.

22.06.2021 : Montarea afisajului pe 7 segmente si a senzorului

27.06.2021 : Implementarea algoritmului pentru timer

30.06.2021 : Atasarea butonului de reset si a bateriei de 9V

Bibliografie/Resurse

- [Laboratorul de timere/intreruperi](#)
- [SevenSegmentDisplay.h](#)
- [Chug.pdf](#)

From:

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

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2021/alazar/chug>



Last update: **2021/05/31 12:59**