Calculator Portabil

Nume: Stan Andrei Razvan

Grupa: 333CA

Prezentare

Pentru acest proiect am decis sa merg pe o **directie bazata mai mult pe software** si sa realizez un **calculator portabil**.

Acesta va putea efectua urmatoarele operatii aritmetice:

- 1. adunare
- 2. scadere
- 3. inmultire
- 4. impartire

Acesta va mai dispune si de un **istoric** al ultimelor rezultate obtinute (folosind **EEPROM-ul** pe care il are placuta Arduino) si de posibilitatea de a afisa ora si data datorita modulului RTC.

Acesta va respecta si ordinea efectuarii operatiilor.

Schema Bloc

×

Hardware Design

Piese utilizate

Componenta in jurul careia se invarte proiectul este o placuta de dezvoltare **Arduino UNO cu procesor ATmega328P**.

Restul componentelor sunt urmatoarele:

- 1x Modul ceas DS1302 RTC;
- 1x Ecran LCD 1602 I2C;

- 1x Matrice de butoane 4×4;
- 3x Buton simplu:
 - Buton CLEAR (sterge expresia)
 - Buton TIME (afisează ora curenta)
 - Buton HISTORY (afisează istoricul calculelor)
- 3x Rezistenta de 10kΩ;
- 1x Switch On/Off;
- 1x Soclu baterie;
- 1x Baterie 9V.

Bill Of Materials

Componenta	Link	Datasheet
DS1302 RTC	Optimus Digital	Data Sheet
Ecran LCD 1602 I2C	Optimus Digital	Data Sheet
Matrice de butoane	Optimus Digital	N∖A
Arduino UNO cu ATmega328P	ArduShop	Data Sheet
LED, Rezistente, baterie, etc.	Kit Plusivo	N/A

Detalierea componentelor principale

Modului de ceas DS1302 RTC este folosit pentru functionalitatea calculatorului de a arata ora. Folosind unul dintre cele doua butoane se va putea schimba intre modul de calcul si cel de ceas, care afiseaza informatiile de la RTC. Acest modul este conectat la Arduino folosind I2C.

Conectarea se realizeaza in felul urmator: SDA \rightarrow SDA de pe Arduino SCL \rightarrow SCL de pe Arduino.

Ecranul LCD este utilizat in proiect pentru a afisa datele. Este conectat in acelasi mod ca si RTC-ul deoarece si acesta foloseste I2C.

Din moment ce **I2C** este un bus pentru transmisie de date serială master-slave am putut sa conectez **ambele** componente la **aceeasi pini** de pe Arduino.

Adresele pe care le folosesc componentele acestea sunt: $0x27 \rightarrow LCD$ $0x68 \rightarrow RTC$

Matricea de butoane este folosita pentru input.

Conectarea se realizeaza in felul urmator:

Pinii de pe matrice	Pinii de pe arduino
C4	7
C3	6
C2	5
C1	4
R1	11

R2	10
R3	9
R4	8

Schematic pe breadboard

Initital am facut proiectul pe **breadboard** iar acesta este modul in care am legat componentele pe breadboard:

×

Schematic final

Pentru proiectul final am lipit componentele pe un PCB universal. Conexiunile s-au schimbat din moment ce am mai adaugat componente:

- O baterie;
- Un switch pentru a porni device-ul;
- Un buton pentru functionalitatea de history pe care am uitat sa il adaug initial.

O alta schimbare este modul in care am conectat matricea de butoane. Am actualizat ulterior tabelul de mai sus cu conexiunile finale.

×

Video hardware

Dovada ca cele trei componente principale functioneaza

In clipul de mai jos se pot observa toate functinalitatile proiectului si cum opereaza acestea.

Clip cu proiectul montat pe PCB universal

Software

Codul pentru acest proiect a fost scris folosind VsCode si Platformio.

Am folosit limbajul de programare specific Arduino unde am inclus diferite librarii externe pentru comunicarea cu perifericele.

Scurta descriere a functionalitatilor

Calculator

- Introdu expresii (ex: 3+5*2)
- Apasa # pentru a calcula (buton colt dreapta-jos de pe matrice)
- Rezultatul este afisat si salvat automat în EEPROM.

Afisare ora

- Apasa butonul corespunzator (de pe pinul **2**)
- Se afisează ora, minutul si secunda curente folosind modulul RTC.

Istoric calcule

- Apasa butonul corespunzator (de pe pinul **3**) pentru a intra în modul istoric
- Se pot naviga ultimele 10 rezultate salvate
 - Tasta A de pe matrice urmatorul rezultat
 - Tasta B de pe matrice rezultatul precedent.

Ștergere expresie

• Apasa butonul corespunzator (de pe pinul **12**) pentru a reseta expresia sau a reveni din alta stare in starea de calculator.

Librarii incluse

Librarie	Motivul utilizarii	Link descarcare
Wire	Comunicare I2C, folosita pentru RTC si LCD	
LiquidCrystal_I2C	Control LCD 16×2 cu interfata I2C	Link
Adafruit_Keypad	Gestionare tastatura matriciala 4×4	Link
uRTCLib	Comunicare cu modulul RTC DS3231	Link
EEPROM (builtin)	Citire si scriere date in EEPROM-ul de pe Arduino pentru functia de istoriei	N/A
Arduino (builtin)	Functii de baza precum `pinMode`, `digitalRead` etc, specifice Arduino	N/A

De asemenea am inclus si **myStack** care este o implementare simpla de stiva pe care am scris-o deoarece limbajul Arduino nu dispune de o implementare de stiva precum C++.

Detalierea codului

Variabile constante si globale

- Keypad: configurat cu pini specifici pentru coloane si randuri
- LCD și RTC: initializati cu adrese I2C
- EEPROM:
 - Adresa 0 indexul actual
 - De la adresa 4 rezultate salvate (max. 10, cate 8 bytes/float)
- Moduri: enumeratie cu 3 stari calculator, ceas si istoric
- Butoane: pinii pentru butoanele de Clear, History si Time
- Flag-uri volatile pentru intreruperi:
 - timeButtonPressed
 - historyButtonPressed

Cum se realizeaza evaluarea unei expresii

Calculul expresiei introduse de utilizator prin tastatura se face manual, fara a folosi o biblioteca de evaluare matematica. In schimb, am ales sa folosesc un **algoritm clasic bazat pe doua stive** – una pentru numere si una pentru operatori – pentru a respecta **ordinea efectuarii operatiilor**.

Practic, functia **evaluateExpression** primeste un sir de caractere pe care le parcurge rand pe rand, in urmatorul mod:

- 1. Se definesc doua stive:
 - values stiva pentru numere (float)
 - ops stiva pentru operatori (char)
- 2. Cat timp intalnim un numar sau un '.' acesta este adaugat in buffer-ul num
- 3. Cand se intalneste un operator, buffer-ul num este transformat in float si este adaugat pe stiva de valori
- 4. Se compara valoarea de "precedenta" a operatului curent cu cea a operatorului din varful stivei de operatori
 - daca operatorul din stiva are o valoare mai mare sau egala se aplica ultimelor doua valori din stiva de valori
 - se repeta acest proces pana cand stiva de operatori este goala sau operatorul curent are o valoare mai mare
- 5. Rezultatul final este ultimul element din stiva de valori.

Intreruperi

- ISR-urile sunt folosite pentru:
 - Butonul Time (pin 2)
 - Butonul History (pin 3)

• Seteaza flag-uri volatile care sunt procesate in loop()

Alte functii importante

- saveToHistory(result):
 - Salveaza rezultatul in EEPROM
 - Actualizeaza indexul circular în EEPROM
- loadFromHistory(index):
 - Recupereaza un rezultat salvat din EEPROM

• printTime():

- Afisează ora curenta citită din RTC
- showHistory():
 - Afiseaza rezultatul curent din istoric, indexat de currentHistoryView

• clearCalculator():

• Goleste expresia, curata ecranul si revine în modul calculator

Observatii

- Pentru setarea orei initiale, se poate decomenta linia cu rtc.set(...) din setup(), se incarca codul pe placuta, se comenteaza inapoi linia si se incarca din nou codul (pentru ca RTC-ul are o baterie si va 'incrementa' singur timpul). Timpul este calculat folosind macro-ul **TIME**, care ia ora de pe PC.
- Sistemul gestionează doar expresii fără paranteze deoarece nu mai aveam destui pini digitali pe Arduino sa mai adaug si butoane pentru paranteze.
- Istoricul este ciclic la mai mult de 10 calcule, cele vechi sunt suprascrise

Download cod

proiect_pm_stan_andrei_razvan.zip

Rezultate Obținute

Care au fost rezultatele obținute în urma realizării proiectului vostru.

Export to PDF

http://ocw.cs.pub.ro/courses/

From: http://ocw.cs.pub.ro/courses/ - CS Open CourseWare	
Permanent link: http://ocw.cs.pub.ro/courses/pm/prj2025/iivasciuc/andrei_razvan.stan	×
Last update: 2025/05/21 18:58	