

# Punching Machine

## Introducere

Se dorește implementarea unui aparat de măsurat forța loviturii cu pumnul la o scară mai mică, pentru a ușura portabilitatea. Aparatul va indica un anumit scor obținut în urma loviturii unei mingi/sac de box.

## Descriere generală

### Schema bloc

## Hardware Design

## Piese

### ESP32:

- Placă programabilă este un ESP32. Am ales această placă pentru a putea beneficia atât de memoria crescută cât și de capacitățile WiFi de care dispune;

### Ecran LCD 1602:

- Tensiune de alimentare: 3.3V/5V;
- Tensiune alimentare backlight: 4.2V;
- Curent: 2mA;
- Modulul comunică prin intermediul unui convertor I2C;

### Receptor laser OKY3301-1:

- Tensiune de alimentare: 2.7V - 5.5V (recomandat 5V);
- Curent: 3mA;
- Output normal pe LOW, când primește laser trece pe HIGH;

### Dioda laser KY-008:

- Tensiune de alimentare: 5V;
- Lungime undă laser: 650nm;

## Electromagnet 34/18:

- Tensiune de alimentare: 12V;
- Curent: 0.5A;
- Forța de atracție: 18kg/180N;

## Modul cu Tranzistor de Putere IRF520:

- Tensiune maximă: 24V;
- Curent maxim: 5A
- Tensiune poartă: 3.3V, 5V;
- Timpi de comutație mici;
- RDS:0.27 ohmi;
- Sarcină poartă: 16nC

## Schema hardware



## Software Design

Voi utiliza ESP-IDF ca development framework. Pentru măsurarea puterii cu care se lovește, se va măsura timpul în care un circuit format dintr-un laser și un receptor este întrerupt de trecerea bratului aparatului prin fața laserului. Aparatul creează un acces point pentru a putea fi controlat într-un browser. Pentru o măsurare cât mai bună, se va calcula diferența în microsecunde dintre începerea întreruperii laserului, până la terminarea acesteia. De asemenea detectarea bratului se realizează prin întreruperi în cod, generându-se un eveniment și procesarea evenimentului într-un task separat, posibil chiar și pe alt procesor, ESP32-WROOM având 2 core-uri.

- **/connect** - aparatul așteaptă ca un controller să se conecteze. Conectarea se realizează prin intermediul unei aplicații pe telefon sau dintr-un browser.
- **/try** - după ce un controller a fost conectat la aparat, acesta va aștepta o cerere POST ce va conține un nume ("name=...") pentru a putea stabili numele jucătorului.
- **/scores** - returnează un răspuns la o cerere GET ce conține un text în format CSV cu primele N cele mai mari scoruri. Numărul de scoruri reținute poate fi configurat din sursa aparatului.
- **/reset** - rețetează aparatul în cazul unei probleme.
- **/player** - întoarce o pagină pentru a putea introduce numele jucătorului. Folosit în special pentru controlare din browser.

Aplicația pentru telefon a fost realizată folosind MIT App Inventor și este disponibilă pentru descărcare în secțiunea Download. Aceasta are nevoie de adresa IP afișată pe ecranul LCD al aparatului pentru a-l putea controla.

## Rezultate Obținute

Am reușit să implementez un aparat destul de complex, mai ales din punct de vedere software, care poate fi montat ca un fidget puțin mai mare pentru birou, ignorând zgomotul creat 😊.

## 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/prj2024/vstoica/mihai.tecliceanu>



Last update: **2024/05/26 22:30**

