

Smart Tennis Racket

Introducere

Ce face

Smart Tennis Racket este un dispozitiv electronic montabil pe racheta de tenis care detectează automat loviturile și oferă feedback obiectiv jucătorului în timp real. Concret, sistemul:

- detectează startul și sfârșitul unei lovituri din semnătura inerțială a swing-ului;
- estimează **tipul loviturii** — *forehand vs backhand* — pe baza sensului de rotație al brațului;
- estimează **intensitatea loviturii** — *slab / mediu / puternic* — din vârful de accelerație;
- estimează **viteza unghiulară maximă** a swing-ului prin integrarea giroscopului;
- afișează un **contor de lovituri** și statistici live pe un display TFT 1.8”;
- oferă **feedback vizual instant** printr-un LED RGB (verde = lovitură bună, roșu = slabă);
- permite reset printr-un buton fizic.

Scop

Antrenament obiectiv pentru jucătorii amatori. Autoevaluarea unui jucător amator se bazează pe simțuri (“a sunat bine”, “a fost puternic”), nu pe date măsurate. Dispozitivul transformă o ședință de antrenament într-un set de metrici concrete: număr lovituri, raport forehand/backhand, distribuția intensităților, viteză medie de swing — fără cost ridicat și fără dependență de un antrenor sau de o aplicație telefonică.

Ideea de pornire

Soluțiile comerciale (Babolat Pop, Sony Smart Tennis Sensor, Zepp Tennis) sunt scumpe, închise ca platformă și au fost discontinue de mai multe ori, lăsând utilizatorii fără suport. Plecând de la observația că un IMU 6-DOF de tipul MPU6050 acoperă cea mai mare parte a senzoricii necesare, ne-am propus să construim un echivalent open, ieftin și deschis pentru extensii.

Utilitate

- feedback obiectiv în timp real, fără telefon și fără cont online;
- statistici per ședință, vizibile direct pe ecranul atașat de rachetă;
- alimentare autonomă pe baterie 9V → uzabil pe teren.

Descriere generală

Sistemul este construit în jurul plăcii de dezvoltare **ATmega328P Xplained Mini**, care joacă rolul de coordonator central. Microcontrollerul citește continuu date inerțiale de la senzorul MPU6050 prin magistrala **I2C**, le procesează printr-o pipeline simplă de detecție și clasificare, după care actualizează simultan două canale de output: ecranul TFT (text și statistici) prin **SPI** și LED-ul RGB (feedback luminos) prin trei canale **PWM**. Un buton conectat ca GPIO permite reset-ul contorului.

Schemă bloc



- **MPU6050** ⇔ **ATmega328P** — *I2C, 100 kHz*: la fiecare ~10 ms, MCU-ul (master) citește 14 octeți (3x accel + temp + 3x gyro) de la slave-ul MPU6050 (adresa 0x68). Sunt folosite primitivile `twi_start / twi_write / twi_read_ack / twi_read_nack / twi_stop` din driverul de la Lab 6.
- **ATmega328P** → **modul detecție swing** (*software*): un filtru pe magnitudinea accelerației ($||a|| - g$) detectează startul și sfârșitul unei lovituri.
- **ATmega328P** → **modul clasificare** (*software*): pe baza semnului ω_z (giroscop, axa verticală a încheieturii) și a vârfului de accelerație, decide *forehand/backhand* și *slab/mediu/puternic*.
- **ATmega328P** → **ST7735 TFT 1.8"** — *SPI, master → slave*: după fiecare lovitură detectată, MCU-ul redesează zona de text și afișează:
 - HITS: 24
 - SHOT: FOREHAND
 - POWER: HIGH
 - SWING: 320 deg/s
- **ATmega328P** → **LED RGB (R+G)** — *PWM*: 2 canale PWM independente, generate de Timer 0 (OC0B = verde) și Timer 2 (OC2B = roșu), modulează intensitatea celor două culori. LED-ul are anod comun → driver-ul software inversează duty-cycle-ul (0 = stins, 255 = maxim).
- **Buton** → **ATmega328P** — *GPIO + INT0*: tratat în întrerupere externă pe PD2, cu debouncing software. La apăsare = reset contor lovituri.

Module software

| Modul | Sursă | Descriere |

<code>twi.c/.h</code>	Lab 6 (preluat, F_CPU adaptat la 16 MHz)	driver I2C low-level
<code>mpu6050.c/.h</code>	scris (după pattern-ul <code>mp_l3115a2</code> din Lab 6)	init, citire accel/gyro burst, calibrare offset
<code>spi.c/.h</code>	Lab 5 (preluat)	driver SPI low-level
<code>st7735.c/.h</code>	Lab 5 (preluat, adaptat pentru pini noi)	driver TFT 1.8"

timers.c/.h	Lab 3 (preluat, extins)	configurare PWM 2 canale pe Timer 0 + Timer 2
rgb.c/.h	scris	wrapper PWM pentru LED RGB (R+G) cu efecte (solid / blink / fade)
swing_detect.c/.h	scris	detecrie swing și clasificare
main.c	scris	bucla principală, integrare module, FSM

Hardware Design

Listă de piese

^ # ^ Componentă ^ Cantitate ^ Rol în proiect ^

1	ATmega328P Xplained Mini (placă dezvoltare)	1	coordonator central, programator on-board
2	MPU6050 (modul GY-521) — accelerometru + giroscop 3 axe	1	senzor inerțial pentru dectecția mișcării (I2C, adresa 0x68)
3	LED RGB 5 mm, 4 pini, anod comun, opac	5	feedback vizual (PWM, doar canalele R+G); restul ca rezervă
4	Display TFT color 1.8", controller ST7735, interfață SPI	1	afișare statistici și tip lovitură
5	Kit rezistențe 1/4 W, 30 valori, 600 buc, 10 Ω - 1 MΩ	1	limitare curent LED (2× 220 Ω pe R și G); pull-up I2C dacă e necesar
6	Breadboard MB-102, 830 puncte	1	suport prototipare
7	Fire dupont 20 cm, mamă-mamă, set 40	1	conexiuni breadboard ↔ Xplained Mini ↔ module
8	Fire dupont 20 cm, tată-mamă, set 40	1	conexiuni mixte (pini Xplained ↔ module)
9	Suport baterie 9V cu jack 5.5×2.1 mm	1	alimentare portabilă
10	Buton tactil PCB mini 6×6×5, 2 pini	1	reset contor lovituri

Software Design

Rezultate Obținute

Concluzii

Download

Bibliografie/Resurse

Resurse hardware

- Datasheet ATmega328P — [Microchip](#)
- MPU-6000/MPU-6050 Register Map and Descriptions — [InvenSense](#)
- MPU-6000/MPU-6050 Product Specification — [InvenSense](#)
- ST7735 datasheet — [Sitronix](#)
- ATmega328P Xplained Mini User Guide — [Microchip](#)

Resurse software

- Laborator 1 (USART) — <https://ocw.cs.pub.ro/courses/pm/lab/lab1-2023-2024>
- Laborator 2 (Înteruperi & Timere) — <https://ocw.cs.pub.ro/courses/pm/lab/lab2-2023-2024>
- Laborator 3 (Timere & PWM) — <https://ocw.cs.pub.ro/courses/pm/lab/lab3-2023-2024>
- Laborator 5 (SPI & ST7735) — <https://ocw.cs.pub.ro/courses/pm/lab/lab5-2023-2024>
- Laborator 6 (I2C) — <https://ocw.cs.pub.ro/courses/pm/lab/lab6-2023-2024>

Lucrări conexe

- jrowberg / i2cdevlib — bibliotecă C++ MPU6050 (referință pentru registre): <https://github.com/jrowberg/i2cdevlib/tree/master/Arduino/MPU6050>
- “Tennis Stroke Recognition Using Inertial Data” — articole academice pe IEEE Xplore (de citat ulterior, în etapa de etapa 3 documentație).

[Export to PDF](#)

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

Permanent link:
<http://ocw.cs.pub.ro/courses/pm/prj2026/atoader/bianca.gorgovan>



Last update: **2026/05/09 20:22**