

Smart Helmet Lock

- Autor: **Huțu Matei-Alexandru**
- Grupa: 331CA
- GitHub: [Matei2807/Proiect-PM](https://github.com/Matei2807/Proiect-PM)

Introducere

Smart Helmet Lock este un sistem de securitate bazat pe microcontroler-ul ATmega328P, conceput pentru a bloca fizic și a securiza breteaua (D ring) a unei căști moto într-o cutie sau pe sașiu motocicletei.

Scopul proiectului este de a oferi o soluție smart și rapidă motocicliștilor care doresc să își lase casca în siguranță lângă motocicletă, fără a fi nevoiți să o transporte manual în magazine, la facultate sau la locul de muncă.

Ideea de pornire a venit din inconvenientul real al echipamentului de protecție voluminos. Căștile moto sunt scumpe și incomode de transportat ca bagaj de mână. Majoritatea soluțiilor de pe piață implică lacăte mecanice cu chei sau cifruri care sunt greu de operat cu mânuși și de calitate proastă.

De ce este util? Sistemul rezolvă această problemă folosind autentificare **RFID** (prin apropierea unui breloc/card) pentru descuierea unui mecanism acționat de un servomotor. Sistemul este dotat cu feedback vizual (semafor LED), feedback acustic (buzzer) și, cel mai important, un sistem de alarmă activat de un senzor de vibrații în cazul tentativelor de furt.

Ipoteză: Consider că utilizarea unei astfel de încuietori va îmbunătăți semnificativ siguranța, deoarece previne furtul prin emiterea unei alarme sonore, descurajând acțiunile de efracție. Mai mult folosirea modulului RFID îl va face mult mai ușor de folosit cu mânuși (sau fără) față de încuietoarele de pe piață.

Descriere generală

Arhitectura proiectului se face în jurul microcontrolerului **ATmega328P** (pe placa Xplained Mini). Sistemul este gândit sub forma unui automat de stări (Unlocked, Locked, Tamper Warning, Alarm).

Interacțiunea modulelor:

- **Modulul RFID (RC522)** comunică cu microcontrolerul prin protocolul **SPI**. Este folosit pentru citirea UID-ului de pe card. Dacă UID-ul corespunde cu master key-ul salvat în memorie, controller-ul schimbă starea sistemului. Deoarece modulul funcționează la 3.3V, liniile de date dinspre microcontroller (5V) trec printr-un divizor de tensiune pentru a proteja modulul.
- **Senzorul de Vibrații (SW-420)** este conectat la un pin setat să genereze o **întrerupere externă**. La detecția unei mișcări, codul iese instant din rutina curentă și semnalează efracția.

- **Servomotorul (SG90)** reprezintă actuatorul mecanic. Primește un semnal **PWM de 50Hz** generat hardware de **Timer 1** (16-bit), modificând ciclul de lucru pentru a roti tija la 0° (Încuiat) sau 90° (Descuiat).
- **Buzzerul Pasiv** este modulat cu un semnal dreptunghiular de 1-2kHz prin **Timer 2** (Fast PWM) pentru emiterea tonurilor de confirmare sau a sirenei de alarmă.
- **Semaforul LED** folosește pini GPIO pentru a oferi utilizatorului feedback vizual imediat (Verde = Deschis, Roșu = Încuiat, Galben = Avertisment).

Schema Bloc:



Hardware Design

Proiectul cuprinde componente care operează atât la 5V, cât și la 3.3V, împărțind GND-ul.

Listă de componente:

| Nume componenta | Model | Protocol |
|-----------------|-------------------------------|----------------------|
| Microcontroller | ATmega328P Xplained Mini | USB, UART |
| Modul RFID | RC522 | SPI |
| Servomotor | SG90 | PWM |
| Senzor Vibrații | SW-420 | Digital Input (INT0) |
| Buzzer pasiv | - | Fast PWM |
| LED-uri | 3 x 5mm (roșu, galben, verde) | GPIO |
| Alte elemente | Rezistențe (330Ω, 1kΩ, 2kΩ) | - |

Descriere detaliată a componentelor hardware:

Modul RFID RC522:

Modulul este folosit pentru citirea cardurilor și funcționează la 3.3V. Comunică cu ATmega328P prin interfața SPI. Deoarece pinii microcontrolerului trimit semnale logice de 5V, pe liniile MOSI, SCK și SDA am implementat un divizor de tensiune (cu rezistențe de 1kΩ și 2kΩ) pentru a proteja modulul.

Conectarea pinilor:

| Pin RC522 | Pin ATmega328P |
|-----------|--------------------------------|
| VCC | 3.3V |
| GND | GND |
| RST | 3.3V |
| MISO | PB4 |
| MOSI | PB3 (prin divizor de tensiune) |
| SCK | PB5 (prin divizor de tensiune) |
| SS | PB2 (prin divizor de tensiune) |

Servomotor SG90:

Acționează mecanismul fizic (zăvorul) pentru a bloca inelele căștii. Este controlat printr-un semnal PWM de 50Hz, generat hardware de Timerul 1 pe 16-biți.

Conectarea pinilor:

| Pin SG90 | Pin ATmega328P |
|-------------------------|----------------|
| VCC (Fir Roșu) | 5V |
| GND (Fir Maro) | GND |
| Semnal (Fir Portocaliu) | PB1 - OC1A |

Senzor de Vibrații SW420:

Folosit ca declanșator pentru alarma. Este conectat la un pin configurat să genereze o întrerupere hardware la orice tranziție a semnalului declanșat de o vibrație.

Conectarea pinilor:

| Pin SW-420 | Pin ATmega328P |
|------------------|----------------|
| VCC | 5V |
| GND | GND |
| DO (Digital Out) | PD2 - INT0 |

Buzzer Pasiv:

Furnizează feedback acustic (beep-uri scurte la scanarea cardului și sirena de alarmă la tampering). Semnalul este o undă generată prin Fast PWM (~1-3kHz) folosind Timerul 2.

Conectarea pinilor:

| Pin Buzzer | Pin ATmega328P |
|------------|----------------|
| I/O | PD3 - OC2B |
| VCC | 5V |
| GND | GND |

LED-uri Stare (Semafor):

Sistemul folosește 3 LED-uri indicatoare. Pentru a proteja LED-urile împotriva arderii, fiecare are în serie câte o rezistență de limitare a curentului de 330Ω între anod și microcontroller.

Conectarea pinilor:

| Componentă | Pin ATmega328P |
|---------------------|----------------|
| Anod LED Roșu (+) | PD5 |
| Anod LED Galben (+) | PD6 |
| Anod LED Verde (+) | PD7 |
| Catozi LED-uri (-) | GND |

Schema Electrica:



TODO rezultatele asamblare hardware

Software Design

1. Mediul de dezvoltare și librării

Proiectul a fost dezvoltat în **C**, folosind **PlatformIO**. O caracteristică importantă a implementării este lipsa bibliotecilor Arduino. Tot codul este scris încât să interacționăm direct cu regiștrii. Acest lucru asigură o folosire mică de memorie și un control strict al temporizărilor (timere, întreruperi).

2. Algoritmul principal: Automatul de Stări

Logica centrală a sistemului este implementată în fișierul `main.c` printr-un automat de stări (FSM). Funcționarea decurge în următorii pași:

- Inițializarea** - Se configurează pinii GPIO, interfața UART, protocolul SPI, și se pornesc timerele și întreruperile (`sei()`). Sistemul pornește implicit în starea `STATE_UNLOCKED`.
- Așteptare (Unlocked)** - LED-ul verde este aprins. Sensorul de vibrații este ignorat intenționat. Sistemul așteaptă scanarea prin SPI a cardului RFID cu UID-ul master.
- Securizarea (Locked)** - La validarea cardului, se emite un beep, servo-ul coboară tija (0°), LED-ul roșu se aprinde, iar flag-ul de întreruperi pentru tamper este activat.
- Avertisment (Tamper Warning)** - Dacă o vibrație declanșează întreruperea `INT0`, se trece într-o stare de avertisment (LED Galben) de 1 secundă. Aceasta permite deblocarea sistemului dacă mișcarea a fost făcută chiar de proprietar înainte de a pune cardul.
- Alarma (Alarm)** - Dacă avertismentul expiră fără prezentarea cardului Master, începe bucla de alarmă vizuală și sonoră de 5 secunde.
- leşirea non-blocantă** - În timpul alarmei sau avertismentului, se folosește o funcție `wait_and_check_master_card()`. Aceasta sparge funcțiile de `delay` în pași de 50ms, verificând constant antena RFID, permițând oprirea instantanee a alarmei.

3. Implementarea modulelor

Pentru a asigura modularitatea codului, funcționalitățile sunt împărțite în fișiere specifice, asemănător cu laboratoarele:

1. `timer.c` (Generare semnale PWM - Lab 3)

- Timer 1 (16-bit):** Setează pinul PB1 (OC1A) în modul Fast PWM. Registrul `ICR1` este folosit ca TOP pentru a seta o frecvență fixă de 50Hz (perioadă de 20ms). Modificarea registrului `OCR1A` controlează ciclul de lucru (1000 tick-uri pentru 0° , 4800 tick-uri pentru 180°), mișcând brațul servomotorului.
- Timer 2 (8-bit):** Setează pinul PD3 (OC2B) în Fast PWM. Este pornit și oprit cu un prescaler de 64 pentru a genera un semnal de $\sim 3\text{kHz}$, acționând buzzerul pentru beep-uri.

2. `tamper.c` (Întreruperi externe - Lab 2)

- Configurează pinul PD2 să declanșeze întreruperea `INT0`. Registrul `EICRA` pentru a declanșa întreruperea la orice modificare logică a pinului (mișcare a sensorului). Rutina (ISR) modifică o variabilă globală volatilă `uint8_t vibration_detected`.

3. `spi.c` și `rfid.c` (Comunicație SPI - Lab 5)

- Microcontrolerul este configurat ca master.
- Codul de comunicare cu modulul RC522 folosește funcții low-level manipulând pinul SS și trimițând date de control pentru a porni antena și a prelua răspunsul de 4 octeți (UID-ul cardului).

4. uart.c (Comunicație Serială - Lab 1)

- Inițializează regiștrii UBRR0H și UBRR0L pentru a seta rata de transfer la 9600 baud. Este folosit pentru afișarea stării curente a sistemului pe laptop prin funcția `UART_print()`.

Rezultate Obținute

Care au fost rezultatele obținute în urma realizării proiectului vostru.

Concluzii

Jurnal

- **10.04.2026 - 01.05.2026:** Stabilirea temei, achiziționarea componentelor, testarea individuală a modulelor. Realizarea milestone 1 (Introducere, Arhitectură Bloc, Lista de componente).
- **02.05.2026:** Realizare schema electrică a proiectului în Cirkuit Designer, adăugare tabele la Hardware Design și completarea Software Design.

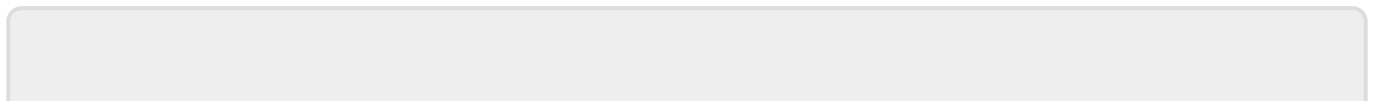
Bibliografie/Resurse

Resurse Hardware:

- [Datasheet Microcontroler ATmega328P](#)
- [User Guide Xplained Mini ATmega328P](#)
- [Datasheet RC522 \(Modul RFID\)](#)
- [Datasheet SG90 Micro Servo](#)

Resurse Software și Tool-uri:

- Google Drawings (Pentru realizarea schemei bloc)
- Cirkuit Designer (Pentru realizarea schemei electrice)
- GitHub (Pentru versionarea codului sursă)



From:

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

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2026/bianca.popa1106/matei.hutu>



Last update: **2026/05/02 20:49**