

Smart Piggy Bank

Introducere

Prezentarea pe scurt a proiectului vostru:

- ce face
- care este scopul lui
- care a fost ideea de la care ați pornit
- de ce credeți că este util pentru alții și pentru voi

Proiectul constă în realizarea unei pușculițe inteligente, care primește monede, le stochează și poate elibera toate monedele de același fel.

Scopul acestei lucrări este de a ușura stocarea monedelor și a folosirii acestora. Ideea inițială a fost de a îmbunătăți un obiect deja existent, apoi îmi venise în minte cum o pușculiță normală poate doar să țină banii, dar nu știi cât ai stocat decât dacă ai scoate toți banii. Astfel, am încercat să vin cu o implementare practică și poate interesantă.

Consider că este un proiect util, promovând aspectul de a economisi monede, de a avea control asupra sumei introduse/ scoase, dar și de a nu mai avea monede prin toate buzunarele sau sub canapele.

Descriere generală

O schemă bloc cu toate modulele proiectului vostru, atât software cât și hardware însoțită de o descriere a acestora precum și a modului în care interacționează.

Exemplu de schemă bloc: <http://www.robs-projects.com/mp3proj/newplayer.html>

Proiectul are ca scop sa automatizeze procesele de depunere, contorizare, stocare și eliberare a monedelor.

Procesul începe prin inserarea monedei într-o fantă laterală. Sortarea este realizată pasiv, bazându-se pe gravitație și dimensiuni fizice. Monedele parcurg un traseu înclinat cu orificii calibrate; fiecare monedă ignoră orificiile mai mici până ajunge la fanta corespunzătoare diametrului său, căzând în tubul de stocare dedicat.

La fiecare tub se află un modul senzor fotoelectric cu infraroșu, care trimite un semnal. Placa primește datele, le procesează și incrementează suma curentă în funcție de moneda introdusă.

Folosind modulul HC-06 pentru a comunica prin Bluetooth cu placa, utilizatorul poate trimite prin telefon o comandă prin care să extragă un tip de monedă. Dacă moneda respectivă nu există sau nu a fost introdusă până în acel moment, se va afișa un mesaj.

Pentru extragerea unui tip de monedă, se va activa un servomotor SG90 pus la fiecare tub pentru a elibera baza tubului pentru câteva secunde, timp în care toate monedele din tub vor cădea pe o fantă. La finalul procesului, se va afișa un mesaj de confirmare atât pe ecranul OLED, cât și pe telefon prin transmiterea Bluetooth.

Schema Bloc



Intrări:

- Modul senzor fotoelectric cu infraroșu;
- Modul Bluetooth HC-06

Ieșiri:

- Servomotoare SG90
- Ecran OLED gme12864

Hardware Design

Aici puneți tot ce ține de hardware design:

- listă de piese
- scheme electrice (se pot lua și de pe Internet și din datasheet-uri, e.g. <http://www.captain.at/electronic-atmega16-mmc-schematic.png>)
- diagrame de semnal
- rezultatele simulării

Componentă	Cantitate	Detalii
ATmega328P Xplained Mini	1	Placă de dezvoltare principală
Modul Bluetooth HC-06	1	Comunicație UART
Display OLED gme12864	1	Cu modul I2C inclus
Modul Senzor Fotoelectric Infrarosu in Forma de U	3	Contorizare monedă
Servomotor SG90	3	Extragere monede
Cablu USB micro	1	Alimentare placă

Fire + conectori	x	Conectarea componentelor
------------------	---	--------------------------

Descrieri și pini

Senzori Fotoelectrici

Folosiți pentru contorizare și un control al fluxului monedelor. Dintre aceștia, 3 sunt folosiți pentru a contoriza monedele (5/10/50 bani), iar ultimul pentru a identifica când iese o monedă în etapa de extragere a banilor. Folosim pinii PC0-3 deoarece sunt pentru I/O analogic.

Componentă	Pin
Senzor 1	PC0
Senzor 2	PC1
Senzor 3	PC2
Senzor 4	PC3

Servomotoare

Folosiți pentru a extrage monedele (3 din ei) + 1 pentru a deschide ușa din spate. Pinii PD5-6 sunt asociați lui Timer0, iar PB1-2 pentru Timer1.

Componentă	Pin
Servo 1	PB1
Servo 2	PB2
Servo 3	PD5
Servo 4	PD6

Modul HC-06

Pentru a asigura o comunicare între placă și telefon, folosim un modul Bluetooth. Placa va comunica cu această componentă prin UART. Folosim pinii PD0 pentru RX și PD1 pentru Tx-ul plăcii.

Componentă	Pin
RX Modul	PD1
TX modul	PD0

Display

Display-ul este folosit pentru a afișa mesaje informative. Folosim pinii dedicați modulului I2C, anume PC4-5

Componentă	Pin
------------	-----

SCL	PC5
SDA	PC4

Schema Electrică



Pe lângă pinii folosiți și distribuirea acestora pentru toate componentele, ar mai fi de menționat și sursa de curent: o baterie externă, din care vor fi trase 2 cabluri: unul care va fi conectat cu servomotoarele, deoarece necesită mai mult curent, iar cel de-al doilea cablu va fi conectat la portul micro usb al plăcii.

Software Design

Descrierea codului aplicației (firmware):

- mediu de dezvoltare (if any) (e.g. AVR Studio, CodeVisionAVR)
- librării și surse 3rd-party (e.g. Procyon AVRlib)
- algoritmi și structuri pe care plănuți să le implementați
- (etapa 3) surse și funcții implementate

1. Mediul de dezvoltare

- Platformă: PlatformIO (ca extensie în Visual Studio Code)
- Framework: AVR-libc
- Target Hardware: ATmega328P (placa Xplained Mini)
- Compilator: AVR-GCC

2. Librării și surse 3rd-party

Proiectul a fost dezvoltat evitând librăriile standard „grele” specifice Arduino, bazându-se pe interacțiunea directă cu regiștrii pentru eficiență maximă.

- Librării standard AVR: `<avr/io.h>`, `<avr/interrupt.h>`, `<util/delay.h>`, `<avr/eeprom.h>`, `<avr/pgmspace.h>`.
- Module Custom (Dezvoltate intern):
 - `adc.h` / `adc.c`: Pentru configurarea și citirea convertoarelor analog-digitale.
 - `usart.h` / `usart.c`: Pentru comunicația serială (Bluetooth/PC).
 - `timers.h` / `test_servos.c`: Pentru generarea semnalelor PWM necesare servomotoarelor.

- `twi.h` / `twi.c`: Implementarea protocolului I2C/TWI hardware.
- `font.h`: Maparea caracterelor (font) stocate în memoria Flash (PROGMEM) pentru ecranul OLED.

Am folosit aceste biblioteci pentru a evidenția cum funcționează perifericele la nivel hardware și de regiștri.

3. Algoritmi și structuri implementate

- Mașină de Stări (Finite State Machine): Arhitectura principală a sistemului este un FSM bazat pe structura `system_state_t` cu stările: `STATE_IDLE`, `STATE_INSERT`, `STATE_AUTH`, `STATE_SELECT_AMOUNT`, `STATE_ERROR`. Această structură izolează logic evenimentele și previne blocajele codului
- Gestiunea Memoriei (EEPROM): Un algoritm care sincronizează vectorii din memoria RAM (`coin_counts`) cu cei din memoria nevolatilă (`eeeprom_coin_counts`) pentru a persista datele în caz de pană de curent, scriind doar când datele se modifică pentru a preveni uzura celulelor EEPROM
- Pentru citirea senzorilor IR s-a implementat o fereastră de prag: moneda este considerată validată doar dacă semnalul trece de pragul superior (500), iar senzorul se eliberează doar când scade sub pragul inferior (300)
- Algoritm Anti-Blocaj: Un timer software implementat în bucla de scanare a senzorilor, care contorizează iterațiile. Dacă o monedă obturează senzorul mai mult de 60 de iterații (aprox. 3 secunde), sistemul detectează și semnalează un blocaj mecanic.

Integrarea funcționalităților din laborator

Utilizarea resurselor hardware a fost justificată astfel:

- ADC (Laboratorul 4): Utilizat pentru a citi în timp real valorile de la modulele foto-interruptoare (senzori IR de monede) conectate la pinii PC0, PC1, PC2.
- Timere (Laboratorul 3): Utilizate (Timer0 / Timer1) pentru a genera semnalul de 50Hz necesar controlului celor 3 servomotoare SG90 care deschid trapele tuburilor.
- Întreruperi (Laboratorul 2): Funcția `ISR(USART_RX_vect)` este esențială pentru a prelua comenzile text transmise via Bluetooth asincron, caracter cu caracter, fără a întrerupe bucla principală de monitorizare a monedelor.
- Comunicare Serială (Laboratorul 1): Reprezintă interfața de comunicație cu modulul Bluetooth HC-06 pentru introducerea parolei și comenzilor de retragere (dispense).
- I2C (Laboratorul 6): Comunicarea cu display-ul OLED SSD1306 se face pe bus-ul I2C pentru a oferi feedback vizual utilizatorului privind soldul curent (fără a-l forța să folosească telefonul).

Calibrarea senzorică

Elementele de tip foto-interruptor (barieră IR) au fost calibrate în două etape:

- Hardware: Ajustarea potențioanelor aflate pe modulele IR astfel încât valoarea de „repaus” (tub gol) să citească o tensiune minimă stabilă (citire ADC cuprinsă între 30-50).
- Software: Setarea pragului de declanșare din software la o valoare suficient de ridicată (`val > 500`)

pentru a evita interferențele (zgomotul optic ambiental), dar și un prag de eliberare separat (val < 300) pentru a oferi imunitate la fluctuațiile de tensiune pe tranziție.

Optimizări realizate

- Eliminarea fenomenului “ADC Bleed/Ghosting”: Deoarece ATmega328P dispune de un singur ADC multiplexat intern, citirea rapidă în buclă a senzorilor cauza “scurgerea” valorii de pe un senzor pe următorul din cauza condensatorului intern S/H (Sample and Hold). S-a optimizat acest proces aplicând tehnica de Dummy Read (citire în gol de tranziție) urmată de un `_delay_ms(2)`, asigurând citiri precise pentru monedele aflate în mișcare rapidă.
- Recepția non-blocantă via Bluetooth: În loc ca placa să aștepte (`while(busy)`) primirea cuvântului complet pe serială, s-a implementat o preluare complet asincronă via ISR. Cuvântul este construit într-un buffer în fundal (`received_command`), iar procesorul este notificat cu un flag (`command_ready = 1`) doar când se detectează un delimitator valid (`\n` sau `\r`).

Elementul de noutate al proiectului

Comparativ cu proiectele clasice de pușculițe digitale care doar adună valori brute, acest proiect aduce o noutate majoră: gestiunea pe categorii (inventar hardware) și o arhitectură hibridă de interacțiune.

Download

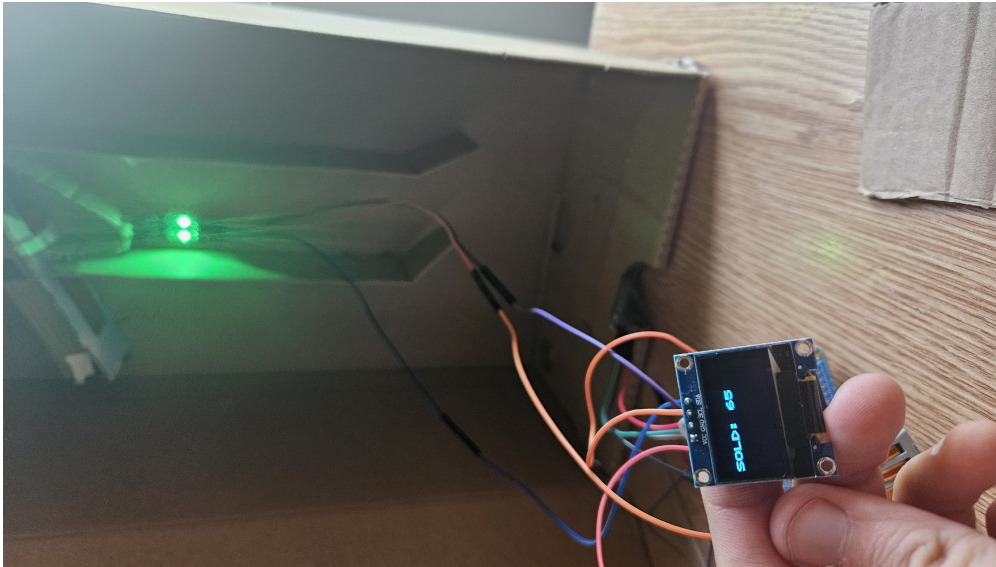
Link pe GitHub către [proiect](#).

Milestone Hardware

Demo

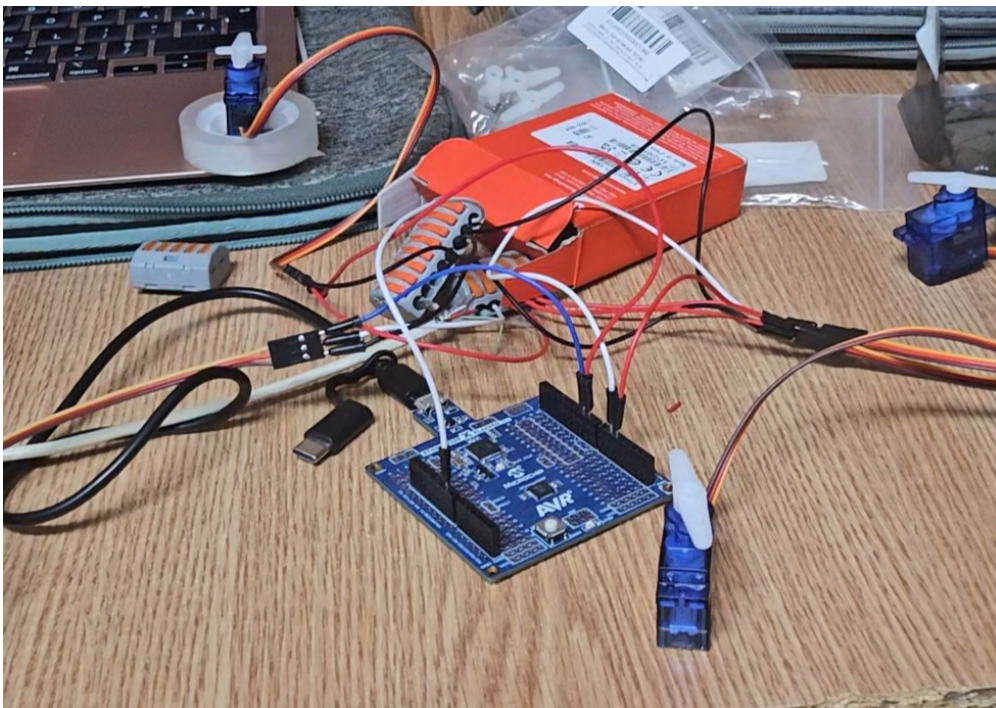
Link-ul către demo poate fi accesat [aici](#).

Testare modul IR



Aici am testat traseul monedei, dacă este citită și afișarea sumei pe ecran.

Testare Servomotoare



Scurt test prin care am testat 4 servomotoare, alimentate la sursa externă. (alimentarea de la placa nu este suficientă pentru atâtea servomotoare)

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/prj2026/vlad.radulescu2901/rares.anton> 

Last update: **2026/05/24 11:58**