

Casă de marcat self pay

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

SmartScale Checkout este un sistem de casă de marcat self-service controlat de un microcontroler ATmega328P. Utilizatorul scanează produsele cu telefonul (folosind o aplicație de scanare coduri de bare), trimite produsul prin Bluetooth la placuță, îl plasează pe cântar pentru validarea greutății, și la final efectuează o plată prin apropierea unui card RFID. Pe tot parcursul, un display LCD afișează produsele adăugate și totalul, iar un buzzer confirmă fiecare acțiune cu un beep.

Proiectul demonstrează integrarea mai multor module hardware într-un sistem coerent, simulând funcționalitatea unui terminal self-pay întâlnit în supermarketuri moderne.

Punctul de plecare a fost observarea că terminalele self-checkout din magazine sunt, în esență, sisteme embedded cu câteva periferice bine definite. Proiectul și-a propus să reproducă aceeași logică la scară redusă, pe o platformă de dezvoltare accesibilă.

Pentru utilizatori, demonstrează că un sistem complet de casă de marcat poate fi construit cu componente ieftine și un microcontroler de 8 biți. Pentru student, reprezintă o oportunitate de a lucra simultan cu UART, SPI, I2C, PWM și GPIO pe același microcontroler, rezolvând probleme reale de sincronizare și validare a datelor.

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>



Descrierea schemei bloc

Intrări:

- HC-06 (UART) — primește de la smartphone, prin Bluetooth, șirul de caractere reprezentând produsul scanat
- HX711 + celulă de sarcină — amplifică și digitalizează forța exercitată pe cântar; comunică cu MCU prin protocol serial sincron pe 2 fire
- Modul RFID RC522 (SPI) — detectează cardul RFID prezentat pentru finalizarea plății

MCU + software:

- ATmega328P Xplained Mini — orchestrează toate perifericele, rulează logica de validare și menține o bază de date minimală de produse (nume, preț, greutate așteptată)
- Firmware — gestionează mașina de stări (așteptare produs → scanare → validare greutate → adăugare în coș → plată), calculează totalul și tratează erorile

Ieșiri:

- LCD (I2C) — afișează numele produsului curent, greutatea măsurată, totalul acumulat și mesajele de eroare
- Buzzer pasiv (PWM) — emite beep la scanare reușită și ton diferit la eroare
- LED roșu (GPIO) — se aprinde când greutatea măsurată nu corespunde produsului scanat

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

Nr.	Componentă	Cantitate	Detalii
1	ATmega328P Xplained Mini	1	Placă de dezvoltare principală
2	Modul Bluetooth HC-06	1	Comunicație UART, 9600 baud
3	Celulă de sarcină 20 kg	1	Tip bare, punte Wheatstone
4	Modul amplificator HX711	1	ADC 24-bit pentru celula de sarcină
5	Display LCD 16x2	1	Cu modul I2C (PCF8574) inclus
6	Modul RFID RC522	1	Include card și tag de test
7	Buzzer pasiv 5V	1	Control prin semnal PWM
8	LED roșu 5mm	1	Indicator eroare validare greutate
9	Rezistență 220Ω	1	Limitare curent LED

10	Breadboard + fire jumper	1 set	Prototipare conexiuni
11	Cablu USB micro	1	Alimentare și programare Xplained Mini
12	Suport/platformă pentru cântar	1	Structură mecanică celulă de sarcină

Componente utilizate și rolul lor

ATmega328P Xplained Mini

Microcontroler principal al proiectului. Rulează firmware-ul care coordonează toate modulele, procesează datele primite prin Bluetooth, validează greutatea măsurată de cântar, controlează afișajul LCD, generează semnalele de feedback (buzzer, LED) și gestionează plata prin RFID. Operează la 16 MHz și dispune de periferice hardware UART, SPI și I2C utilizate simultan în acest proiect.

Modul Bluetooth HC-06

Modul de comunicație wireless care permite trimiterea datelor de la smartphone către microcontroler. Utilizatorul scanează codul de bare al unui produs cu telefonul și trimite datele prin Bluetooth. HC-06 operează ca dispozitiv slave și comunică cu ATmega prin protocolul UART la 9600 baud. Alimentat la 5V, cu protecție pe linia RX prin divizor de tensiune (pinul RX al HC-06 este compatibil cu 3.3V, nu cu 5V).

Celulă de sarcină 20 kg + Modul amplificator HX711

Celula de sarcină este un traductor mecano-electric care convertește forța aplicată (greutatea unui obiect) într-un semnal electric de tensiune foarte mic, prin intermediul unei punți Wheatstone. Semnalul analogic este amplificat și digitizat de modulul HX711, un convertor analog-digital de 24 biți dedicat aplicațiilor de cântărire. ATmega citește datele prin protocol serial sincron pe 2 fire (SCK și DT). Rolul în proiect: validarea greutății produsului scanat față de greutatea așteptată stocată în firmware.

Modul RFID RC522

Modul de citire RFID care operează la frecvența de 13.56 MHz (standard ISO 14443A). Detectează cardul sau tag-ul RFID prezentat de utilizator și transmite un identificator unic (UID) către microcontroler prin protocolul SPI. Rolul în proiect: simularea plății — prezentarea cardului finalizează sesiunea de cumpărături. Alimentat obligatoriu la 3.3V (nu 5V).

Display LCD 16x2 cu modul I2C PCF8574

Afișaj cu cristale lichide cu 2 rânduri a câte 16 caractere fiecare. Modulul I2C PCF8574 reduce numărul de pini necesari de la 8 la 2 (SCL și SDA), comunicând cu ATmega prin protocolul I2C. Afișează: numele produsului scanat, greutatea măsurată, totalul acumulat și mesaje de eroare.

Buzzer pasiv

Traductor electroacustic care necesită un semnal de frecvență variabilă pentru a produce sunet (spre deosebire de buzzerul activ care are oscilator intern). Este controlat prin semnal PWM generat pe pinul OC2B (PD3) al ATmega. Produce un beep scurt la fiecare scanare reușită și un ton diferit la eroare de greutate.

LED roșu

Indicator vizual de eroare. Se aprinde când greutatea măsurată nu corespunde cu greutatea așteptată a produsului scanat. Conectat pe pinul PD4 prin rezistență de limitare a curentului de 100Ω.

Pini utilizați și justificarea alegerii

Componentă	Pin ATmega328P	Protocol	Justificare
HC-06 TX	PD1 (RX UART)	UART	Pinii PD0/PD1 sunt dedicați hardware pentru UART (USART0), asigurând comunicație serială fiabilă la 9600 baud fără consum de resurse software
HC-06 RX	PD0 (TX UART)	UART	Idem, pinul de transmisie hardware UART
HX711 SCK	PD7	GPIO (bit-bang)	HX711 folosește un protocol serial proprietar simplu, implementat prin bit-banging pe GPIO standard; PD7 ales pentru a evita conflicte cu SPI și UART
HX711 DT	PD6	GPIO (bit-bang)	Pinul de date al HX711; PD6 ales alături de PD7 pentru organizare logică
RC522 SCK	PB5	SPI	PB5 este pinul hardware dedicat SCK al interfeței SPI (MSTR) a ATmega328P
RC522 MISO	PB4	SPI	PB4 este pinul hardware dedicat MISO al interfeței SPI
RC522 MOSI	PB3	SPI	PB3 este pinul hardware dedicat MOSI al interfeței SPI
RC522 SS (SDA)	PB2	SPI	PB2 este pinul hardware dedicat SS al interfeței SPI, folosit pentru selectarea RC522
RC522 RST	PD5	GPIO	Pin GPIO simplu pentru resetarea modulului RC522; ales din pini disponibili
LCD SCL	PC5	I2C	PC5 este pinul hardware dedicat SCL al interfeței I2C (TWI) a ATmega328P

LCD SDA	PC4	I2C	PC4 este pinul hardware dedicat SDA al interfeței I2C (TWI) a ATmega328P
Buzzer	PD3 (OC2B)	PWM	PD3 este ieșirea comparatorului B al Timer/Counter2, permițând generarea semnalului PWM hardware pentru controlul frecvenței buzzerului pasiv
LED roșu	PD4	GPIO	Pin GPIO simplu pentru controlul LED-ului de eroare; rezistență de 100Ω în serie

Schema electrică



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 plănuiți să le implementați
- (etapa 3) surse și funcții implementate

Rezultate Obținute

Care au fost rezultatele obținute în urma realizării proiectului vostru.

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/prj2026/vlad.radulescu2901/alexandru.istrate04>



Last update: **2026/05/13 09:06**