

# Casă de marcat self pay

## Introducere

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



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

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 16×2	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 100Ω	4	Limitare curent pentru LED și pentru pinul RX al HC-06
10	Fire jumper	1 set	Conexiuni
11	Cablu USB micro	1	Alimentare și programare Xplained Mini
12	Suport pentru cântar	1	Structură mecanică celula 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 mesajele 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.

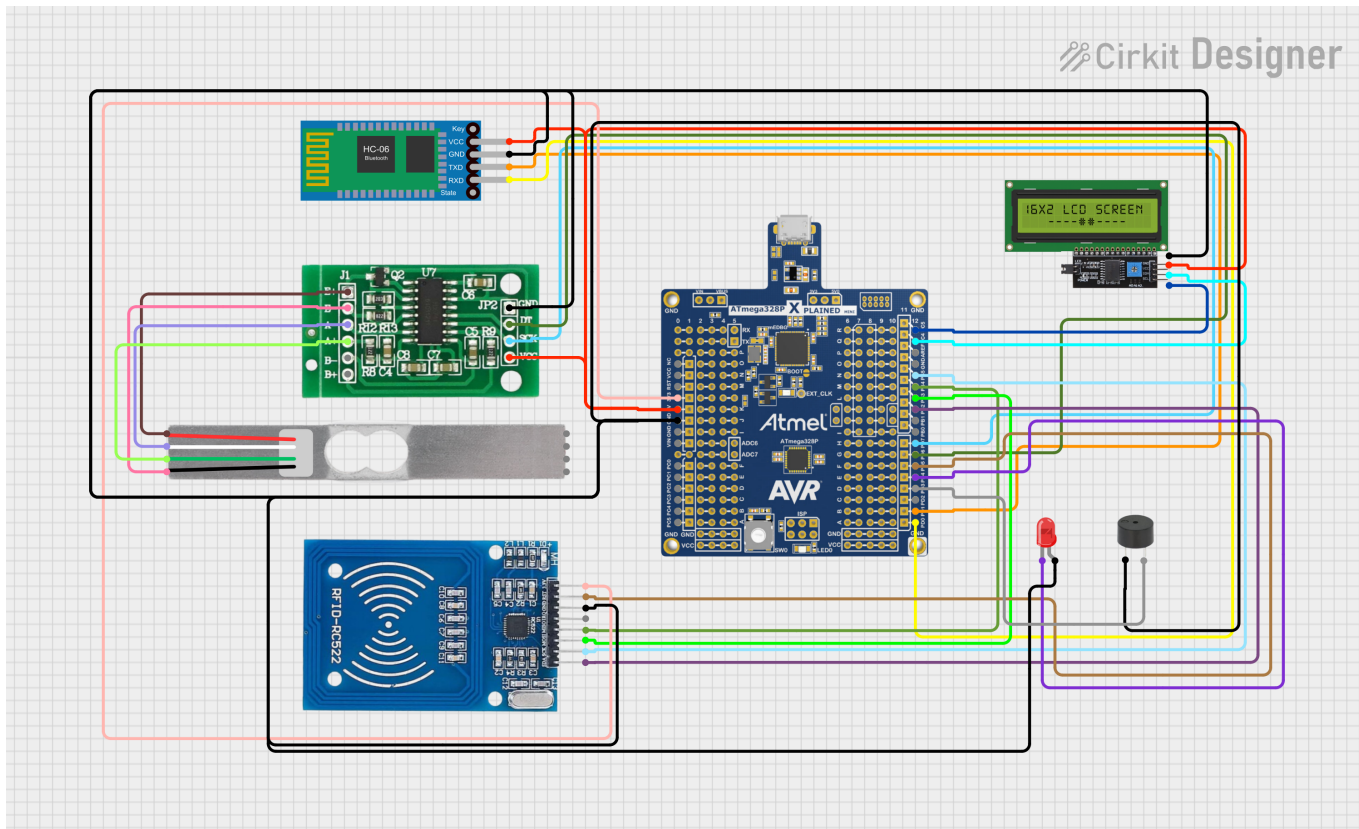
## Pini utilizați și justificarea alegerii

Componentă	Pin ATmega328P	Protocol	Justificare
------------	-------------------	----------	-------------

HC-06 TX	PD0 (TX 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	PD1 (RX 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 alăturat lui 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 modului RC522; ales din pinii 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: leșire digitală pentru buzzerul pasiv. Frecvența este generată asincron folosind Timer/Counter1 și o rutină de tratare a întreruperilor (ISR) pentru a asigura acuratețea tonurilor.
LED roșu	PD4	GPIO	Pin GPIO simplu pentru controlul LED-ului de eroare; rezistență de 100Ω în serie

Pinul **IRQ** al modului RC522 este lăsat **neconectat** în mod intenționat. Bibliotecia folosită pentru comunicația cu RC522 utilizează metoda polling, nu întreruperi hardware, deci pinul IRQ nu este necesar pentru funcționarea corectă a proiectului.

## Schema electrică



Schema electrică a proiectului grupează componentele în două zone distincte:

- **Stânga** — module de intrare: HC-06 (Bluetooth), HX711 cu celula de sarcină și RC522 (RFID)
- **Dreapta** — module de ieșire: LCD 16x2, LED roșu și buzzer pasiv

Pentru claritatea schemei, firele sunt colorate după funcție și protocol:

Culoare	Semnificație
Roșu	VCC 5V
Negru	GND
Roz	VCC 3.3V (exclusiv RC522)
Portocaliu	UART TX
Galben	UART RX
Albastru deschis	SPI SCK
Verde	SPI MOSI
Verde deschis	SPI MISO
Violet	SPI SS/NSS și GPIO LED
Maro	SPI RST
Albastru închis	I2C SCL
Cyan	I2C SDA
Gri	GPIO Buzzer

Proiectul folosește două tensiuni de alimentare, ambele furnizate de placa ATmega328P Xplained Mini prin intermediul conectorului USB:

- **5V** — folosit pentru HC-06, HX711, LCD, LED și buzzer.
- **3.3V** — folosit exclusiv pentru modulul RC522. Acesta nu este compatibil cu 5V; alimentarea incorectă poate deteriora permanent modulul.

# Software Design

## Mediu de dezvoltare și bibliotecile folosite

Firmware-ul a fost programat în Visual Studio Code, folosind extensia PlatformIO, care oferă un mediu integrat pentru compilarea codului, gestionarea dependențelor și încărcarea binarului rezultat direct pe microcontroler.

Proiectul nu utilizează biblioteci externe de tip third-party. Toate driverele pentru periferice (LCD, RC522, HX711) și protocoalele de comunicație (UART, SPI, I2C) au fost implementate de la zero peste registrele hardware ale ATmega328P.

Sunt utilizate exclusiv biblioteci standard din setul AVR-GCC:

- **avr/io.h** — definițiile registrelor hardware ale ATmega328P (PORTD, DDRB, SPCR, TWCR etc.); esențială pentru orice program AVR
- **avr/pgmspace.h** — macrourele PROGMEM și memcpy\_P pentru stocarea și citirea datelor din memoria Flash; folosită pentru baza de date de produse
- **util/delay.h** — funcțiile \_delay\_ms() și \_delay\_us() pentru temporizări precise bazate pe F\_CPU; folosite în inițializarea LCD, protocoalele SPI/I2C și logica de feedback
- **stdint.h** — tipuri de date cu dimensiune fixă (uint8\_t, uint16\_t, int8\_t); critice pe un MCU de 8 biți unde dimensiunea tipurilor contează
- **string.h** — funcții de manipulare șiruri (strcmp, strcpy, strcat, memset); folosite pentru parsarea comenzilor UART și formatarea textului pentru LCD
- **stdlib.h** — funcțiile itoa() și ltoa() pentru conversia numerelor în șiruri de caractere afișabile pe LCD sau trimise prin UART

## Elementul de noutate

Elementul de noutate al proiectului constă în integrarea unui mecanism de validare a greutateii în fluxul de scanare, inspirat din sistemele self-checkout din supermarketurile moderne.

Spre deosebire de un simplu sistem de scanare și afișare, SmartScale Checkout blochează progresul sesiunii dacă greutatea produsului plasat pe cântar nu corespunde greutății declarate în baza de date (toleranță +-10%). Această validare elimină posibilitatea scanării unui produs și plasării altuia pe cântar. Un alt element distinctiv este mecanismul de supervisor override. Scanarea unui cod special deblochează sesiunea fără validarea greutății, simulând intervenția unui angajat în caz de eroare, funcționalitate prezentă în terminalele self-checkout reale.

Interfața mobilă dezvoltată în MIT App Inventor completează sistemul cu un flux complet: scanare cod de bare prin camera telefonului, transmisie Bluetooth către microcontroler, afișare bon fiscal digital la finalul sesiunii.

## Scheletul proiectului

Firmware-ul este organizat în module separate, fiecare responsabil de un periferic:

- **uart.cpp** — inițializare și comunicație serială cu HC-06
- **i2c.cpp** — protocolul pentru comunicația cu LCD
- **lcd.cpp** — driver pentru afișajul LCD 16×2
- **spi.cpp** — protocol SPI pentru comunicația cu RC522
- **rc522.cpp** — driver pentru cititorul RFID
- **load\_cell.cpp** — citire și calibrare HX711
- **buzzer.cpp** — generare semnale sonore prin PWM
- **main.cpp** — logica principală: baza de date, sesiunea curentă și mașina de stări

Interacțiunea dintre module urmează un flux unidirecțional. Toate modulele perifericelor sunt apelate în main, cel care coordonează activitatea proiectului, fiind independente unele de celelalte.

**Mașina de stări** are 5 stări:

- **STATE\_SCAN** — starea principală, așteaptă cod de bare
- **STATE\_DELETE** — așteaptă codul produsului de șters
- **STATE\_WEIGHT\_ERROR** — blocat până la validarea greutății sau cod supervisor
- **STATE\_PAYMENT** — așteaptă cardul RFID și confirmă plata

**Validarea funcționării** a fost realizată în două etape:

- **Testare individuală** — fiecare modul a fost testat separat printr-un program dedicat de test, controlat prin Bluetooth de pe smartphone
- **Testare integrată** — după integrarea tuturor modulelor, s-a parcurs fluxul complet: scanare produs, validare greutate, finalizare sesiune cu card RFID și afișare bon fiscal

## Utilizarea funcționalităților din laborator

Proiectul SmartScale Checkout utilizează direct cunoștințele acumulate în cadrul laboratoarelor, atât la nivel teoretic cât și practic.

### GPIO

Configurarea pinilor folosește exact mecanismul prezentat în laboratorul 0, prin registrele DDRn și PORTn:

- LED-ul roșu pe PD4 este configurat ca ieșire prin DDRD și controlat prin PORTD, identic cu exemplul de aprindere a unui LED
- Pinii HX711 (PD6, PD7) sunt configurați explicit ca intrare, respectiv ieșire, prin DDR și PIN, pentru implementarea protocolului proprietar al amplificatorului prin bit-banging

## USART

Comunicația serială cu modulul Bluetooth HC-06 utilizează perifericul USART0 al ATmega328P, configurând registrele prezentate în laboratorul 1:

- UBRR0H și UBRR0L pentru setarea baud rate-ului la 9600, calculat cu formula  $F\_CPU/(UART\_BAUD\_RATE*16)-1$
- UCSR0B pentru activarea transmisiei și recepției prin biții TXEN0 și RXEN0
- UCSR0C pentru formatul cadrului: 8 biți de date, fără paritate

## PWM

Buzzerul pasiv necesită un semnal cu frecvență variabilă pentru a produce sunet. Teoria PWM din laboratorul 3 a stat la baza implementării funcției beep(), care generează semnalul pe PD3, alternând starea pinului la un interval calculat din frecvența dorită.

## SPI

Comunicația cu modulul RFID RC522 utilizează interfața SPI hardware a ATmega328P, configurată prin registrele studiate în laboratorul 5:

- SPCR pentru activarea SPI în modul master (bitul MSTR), activarea interfeței (SPE) și setarea prescaler-ului
- SPSR cu flag-ul SPIF pentru așteptarea finalizării transferului
- SPDR pentru scrierea și citirea datelor
- Pinul SS (PB2) este controlat manual prin PORTB, identic cu mecanismul de Chip Select descris în laborator

RC522 folosește arhitectura Master-Slave descrisă în laborator: ATmega este master, RC522 este slave, selectat prin SS activ LOW.

## I2C

Display-ul LCD 16x2 cu expander PCF8574 comunică prin protocolul I2C (TWI), implementat folosind registrele prezentate în laboratorul 6:

- TWBR pentru setarea frecvenței SCL la 100kHz
- TWCR pentru generarea condițiilor START și STOP, prin biții TWSTA, TWSTO și TWINT
- TWDR pentru scrierea datelor către expander
- Secvența completă de comunicație urmează pașii din laborator: START → adresă slave → date →

STOP

Adresa expanderului PCF8574 (0x27) este transmisă conform protocolului descris în laborator, cu bitul de scriere (0) pe poziția LSB.

## Calibrarea celulei de sarcină

Calibrarea HX711 a presupus determinarea a două valori:

**Tara (offset)** — valoarea raw returnată de HX711 cu cântarul gol, determinată prin citirea mediei a 5 valori consecutive după stabilizarea mecanică a sistemului:

```
Tara = 232121 (unități raw)
```

**Factorul de calibrare** — numărul de unități raw per gram, determinat prin plasarea unui obiect cu greutate cunoscută pe cântar. S-a folosit un smartphone cu greutatea verificată pe un cântar de bucătărie (266g):

```
Raw cu greutate = 259448  
Diferență = 259448 - 232121 = 27327 unități  
Factor = 27327 / 266g = 102.7 unități/gram
```

Formula de conversie implementată în firmware:

```
greutate(g) = (raw - 232121) / 102.7
```

Validarea calibrării s-a făcut prin măsurarea aceluiași obiect. Rezultatul returnat de firmware a fost 266g, identic cu valoarea de referință.

Toleranța de validare a fost setată la  $\pm 10\%$  față de greutatea declarată în baza de date, justificată de precizia celulei de sarcină de 20kg pe produse mici și de variațiile normale ale ambalajelor.

## Optimizări realizate

**Memorie Flash — PROGMEM:** Baza de date cu cele 7 produse (nume, cod, preț, greutate) este stocată în memoria Flash folosind directiva PROGMEM, nu în SRAM. Pe ATmega328P cu doar 2KB SRAM, stocarea unui array de structuri în RAM ar consuma inutil memorie necesară stack-ului și variabilelor dinamice. Citirea se face prin memcopy\_P la momentul accesării. În total, implementarea curentă permite stocarea a aproximativ 500 de produse, luând în calcul că un produs ocupă 33 de bytes de memorie în structura curentă.

**Medie a 10 citiri HX711:** HX711 produce valori cu zgomot. Citirea unei singure valori raw poate genera erori de validare false. Funcția hx711\_read\_avg() calculează media aritmetică a 10 citiri consecutive, reducând zgomotul și îmbunătățind stabilitatea măsurătorilor.

**Modularizare pe fișiere separate:** Împărțirea codului pe module (uart, i2c, lcd, spi, rc522, load\_cell, buzzer) permite compilarea separată și recompilarea doar a modulelor modificate, reducând timpul


de build. De asemenea, ușurează urmărirea codului și îmbunătățirea lui pe termen lung.

## Demo

În videoclipul de mai sus se prezintă proiectul în stadiul final, după milestone-ul pentru software. De asemenea, se trece și printr-o rutină de scanare, primele produse fiind validate prin greutate, cel din urmă fiind validat cu ajutorul codului de supervisor.

## Concluzii

## Download

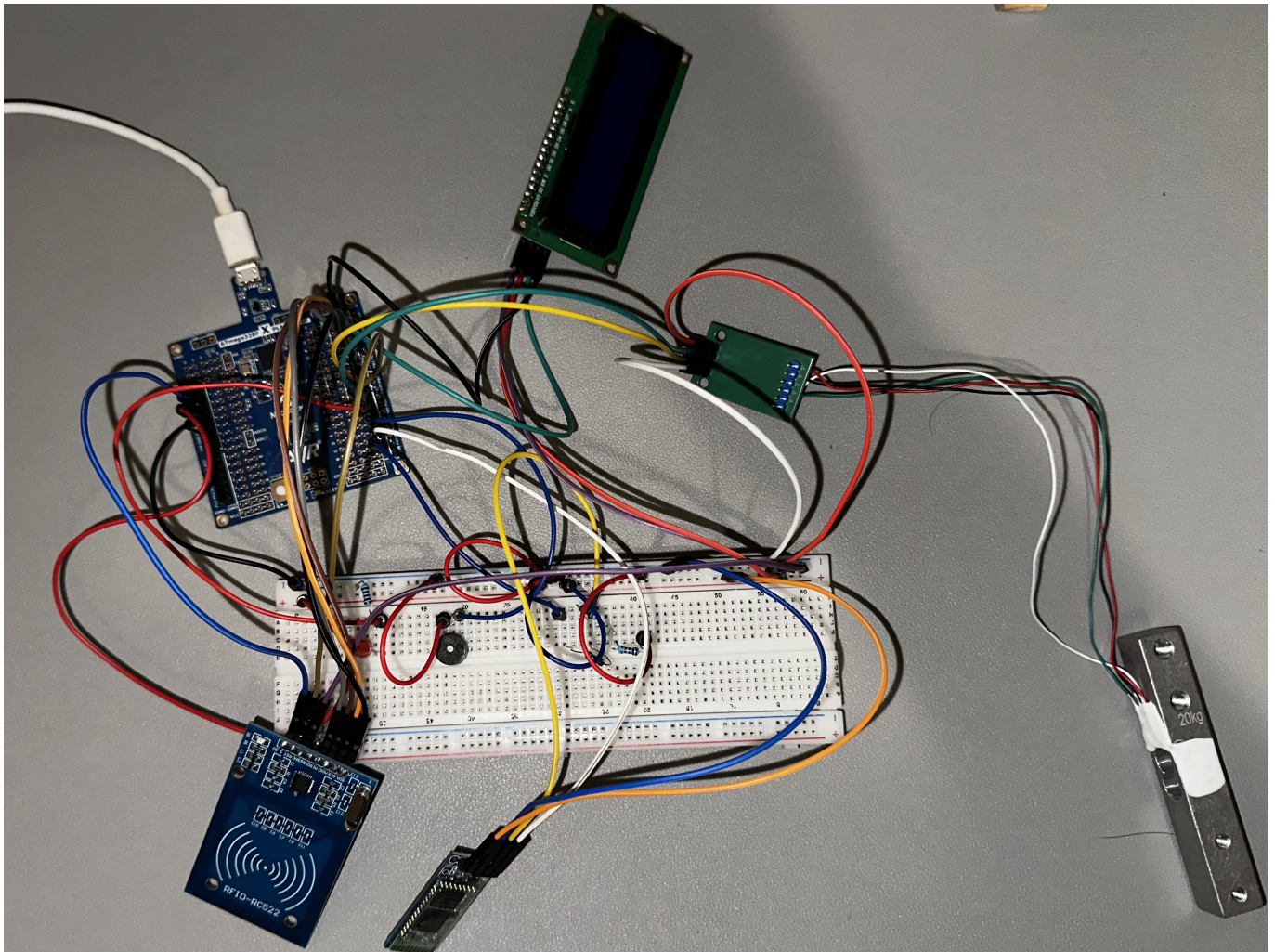
O arhivă (sau mai multe dacă este cazul) cu fișierele obținute în urma realizării proiectului: surse, scheme, etc. Un fișier README, un ChangeLog, un script de compilare și copiere automată pe uC crează întotdeauna o impresie bună .

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**.

[Export to PDF](#)

## Jurnal

### Stadiul proiectul la milestone-ul de hardware



Imaginea prezintă montajul complet al proiectului pe breadboard, în stadiul de prototipare. Se pot identifica toate componentele care alcătuiesc sistemul dezvoltat. În filmulețul de prezentare, se demonstrează funcționarea individuală a tuturor componentelor și, mai mult, că este posibilă și comunicarea între smartphone și placa de dezvoltare. Testarea s-a făcut individual, în primă instanță, pentru a mă asigura că pot fi folosite ulterior în produsul final.

## Bibliografie/Resurse

### Resurse hardware

Nr. Componentă	Link
1 ATmega328P Xplained Mini	<a href="https://ww1.microchip.com/downloads/en/DeviceDoc/50002659A.pdf">https://ww1.microchip.com/downloads/en/DeviceDoc/50002659A.pdf</a>
2 Modul Bluetooth HC-06	<a href="https://components101.com/sites/default/files/component_datasheet/HC06%20Bluetooth%20Module%20Datasheet.pdf">https://components101.com/sites/default/files/component_datasheet/HC06%20Bluetooth%20Module%20Datasheet.pdf</a>
3 Călușă de sarcină 20 kg	<a href="https://www.gotronic.fr/pj-460.pdf?srsltid=AfmBOoqofpsrpgauUtd77Wck0RIZxTGMKEYsWmxeX4NpY7LXV8xWz">https://www.gotronic.fr/pj-460.pdf?srsltid=AfmBOoqofpsrpgauUtd77Wck0RIZxTGMKEYsWmxeX4NpY7LXV8xWz</a>
4 Modul amplificator HX711	<a href="https://www.google.com/url?sa=t&amp;source=web&amp;rct=j&amp;opi=899784496&amp;url=https://community.infineon.com/gfawx74859/attachments/gfawx74859/CodeExamples/546/7/HX711_v0_0_B.pdf&amp;ved=2ahUKEwjwJXtc-UAxXBS_EDHTT1MnUQFnoECB0AQ&amp;usq=AOvWaw1pjHusoY7T_H1IsxugKvkg">https://www.google.com/url?sa=t&amp;source=web&amp;rct=j&amp;opi=899784496&amp;url=https://community.infineon.com/gfawx74859/attachments/gfawx74859/CodeExamples/546/7/HX711_v0_0_B.pdf&amp;ved=2ahUKEwjwJXtc-UAxXBS_EDHTT1MnUQFnoECB0AQ&amp;usq=AOvWaw1pjHusoY7T_H1IsxugKvkg</a>
5 Display LCD 16x2	<a href="https://www.vishay.com/docs/37484/lcd016n002bcthet.pdf">https://www.vishay.com/docs/37484/lcd016n002bcthet.pdf</a>
6 PCF8574 Remote 8-Bit I/O Expander for I2C Bus	<a href="https://www.ti.com/lit/ds/symlink/pcf8574.pdf">https://www.ti.com/lit/ds/symlink/pcf8574.pdf</a>
7 Modul RFID RC522	<a href="https://www.nxp.com/docs/en/data-sheet/MFRC522.pdf">https://www.nxp.com/docs/en/data-sheet/MFRC522.pdf</a>

Last update: 2026/05/25 10:37 pm:prj2026:vlad.radulescu2901:alexandru.istrate04 <http://ocw.cs.pub.ro/courses/pm/prj2026/vlad.radulescu2901/alexandru.istrate04>

---

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: **2026/05/25 10:37**