

Smart Easy-Box

Autori: Cărăuleanu Valentin Gabriel

Grupa: 331CA

Introducere

Proiectul presupune realizarea unui sistem integrat de securitate, bazat pe două microcontrolere Arduino care comunică între ele prin protocolul I2C. Acesta utilizează diverși senzori și module pentru a oferi acces securizat și monitorizarea mediului, asigurând astfel integritatea conținutului din interior.

Descriere generală

1. Arduino Master

Rol: Controlul central al sistemului, interfața cu utilizatorul.

Componente conectate:

- **LCD 16x2 I2C** – afișează statusul, ora, temperatura, umiditatea și nivelul de sunet.
- **Keypad 4x4** – permite introducerea codului de acces.
- **Servo motor** – controlează mecanismul de blocare/deblocare al ușii.
- **LED** – se aprinde când ușa este deblocată.
- **Buzzer** – avertizează la acces neautorizat sau mișcări puternice.
- **Ultrasonic Sensor** – detectează prezența unui utilizator în apropiere.
- **Conexiune I2C cu Slave** – solicită date de la unitatea slave.

2. Arduino Slave

Rol: Colectarea datelor de mediu și detectarea mișcărilor neautorizate.

Componente conectate:

- **DHT11** – măsoară temperatura și umiditatea.
- **KY-037** – microfon analogic pentru detectarea nivelului de zgomot.
- **SW-530** – senzor de înclinare (pentru detectarea mișcărilor).
- **DS1302 RTC** – modul de ceas în timp real pentru data și ora curente.

Masterul interoghează Slave-ul prin I2C pentru a primi un `SensorPacket` care conține:

- Ora și data
- Temperatura și umiditatea
- Nivelul de sunet
- Starea de înclinare (tilt)

Dacă **tilt** este "true", sistemul presupune o tentativă de furt și activează alarma (buzzer).

Accesul se face prin cod PIN introdus pe keypad. Dacă codul este corect, se deblochează ușa și se aprinde LED-ul.

În lipsa utilizatorului, ecranul LCD se stinge automat.

Use-case-uri și fluxuri de lucru

1. Accesul utilizatorului la seif

Scop: Permite deblocarea dulapului doar de către utilizatori autorizați.

Flux:

- Senzorul ultrasonic detectează prezența unei persoane.
- LCD-ul afișează promptul pentru introducerea codului.
- Utilizatorul introduce codul PIN pe tastatură.
- Codul este verificat: dacă este corect: servomotorul deschide dulapul, LED-ul se aprinde, apare mesajul "Access Granted"; dacă este greșit: apare un mesaj sugestiv; după 3 încercări greșite, buzzer-ul rămâne pornit.
- Se apasă `B` pentru a rebloca seiful, `A` pentru a schimba cifrul.

2. Setarea sau schimbarea codului de securitate

Scop: Permite configurarea sau schimbarea codului de securitate de către utilizator.

Flux:

- Seiful este deblocat.
- Utilizatorul apasă `A`.
- Se cere introducerea unui nou cod (4 cifre).
- Codul este confirmat prin reintroducere.
- Dacă cele două coduri se potrivesc, noul cod este salvat.
- Dacă nu se potrivesc, apare mesajul "Code mismatch".

3. Monitorizarea mediului și alertare

Scop: Se monitorizează parametrii de mediu și se generează alerte în caz de manipulare.

Arduino Slave măsoară periodic:

- Temperatura și umiditatea (DHT11)
- Nivelul de sunet (KY-037)
- Timpul curent (DS1302)
- Înclinarea (SW-520D)

Dacă este detectată înclinare:

- La următoarea cerere I2C, flagul `tilt` este transmis

Arduino Master cere periodic datele de la slave, chiar și fără prezență umană. Dacă `tilt = true`:

- Sistemul activează alarma (buzzer-ul pornește permanent)
- Se consideră tentativă de furt

4. Afișarea informațiilor pe ecran

Scop: Afișează informațiile esențiale într-un format concis.

Format LCD:

- Rând 1: Icon lacăt, temperatură, umiditate, sunet
- Rând 2: Ora curentă și zona pentru cod: `H:hh [] M:mm`

Schema Bloc



Gantt Chart



Hardware Design

Sistemul este alcătuit din două plăci de dezvoltare Arduino Uno, care comunică între ele prin magistrala I2C în configurație Master-Slave. Partea de Master controlează afișajul LCD 1602 cu interfață I2C, un servo-motor SG90 pentru închiderea fizică a seifului, o tastatură matricială 4x4 pentru introducerea codului PIN, un buzzer pentru semnalizare acustică și un LED de stare. Placa Slave este echipată cu senzori de mediu: un DHT11 pentru temperatură și umiditate, un senzor de sunet KY-037 pentru detecția zgomotului ambiental, un modul de timp real RTC DS1302 pentru afișarea orei curente și un senzor de înclinare SW-520D, conectat la o întrerupere externă pentru semnalizarea mișcărilor neautorizate. Întregul ansamblu este alimentat prin USB, fiind gândit să funcționeze într-un sistem închis și compact, ideal pentru securizarea unui obiect fizic.

Bill of Materials:

Nr.	Componentă	Cantitate	Pret (RON)
1	Arduino Uno	2	35
3	LCD 16x2 cu I2C	1	16
4	Tastatură Keypad 4x4	1	7
5	Servomotor SG90	1	12
6	Modul DHT11 (temp+umiditate)	1	12
7	Modul KY-037 (microfon)	1	5
8	Modul SW-520 (tilt sensor)	1	5
9	Modul RTC DS1302	1	15
10	Senzor ultrasonic HC-SR04	1	20
11	LED roșu	1	~0
12	Rezistor 220Ω	1	~0
13	Buzzer activ	1	2
14	Fire jumper M-M și F-F	~20	30
15	Breadboard	1	14

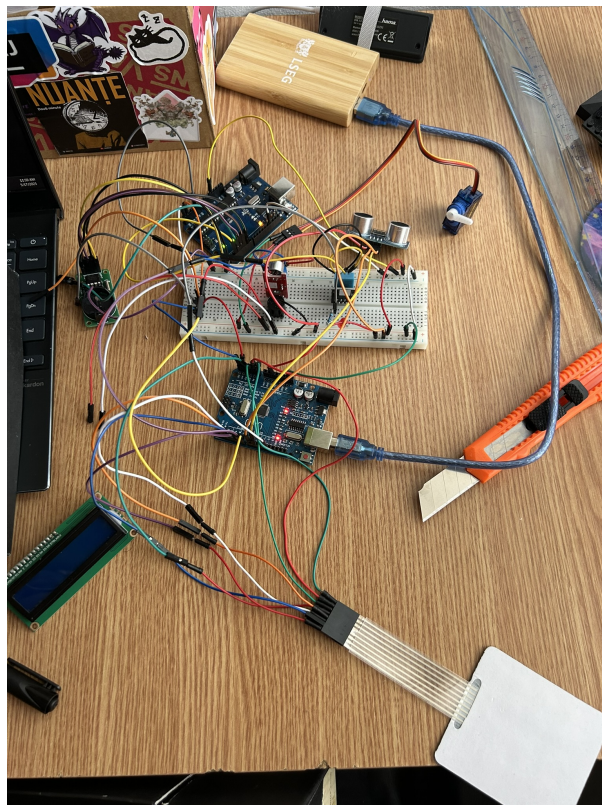
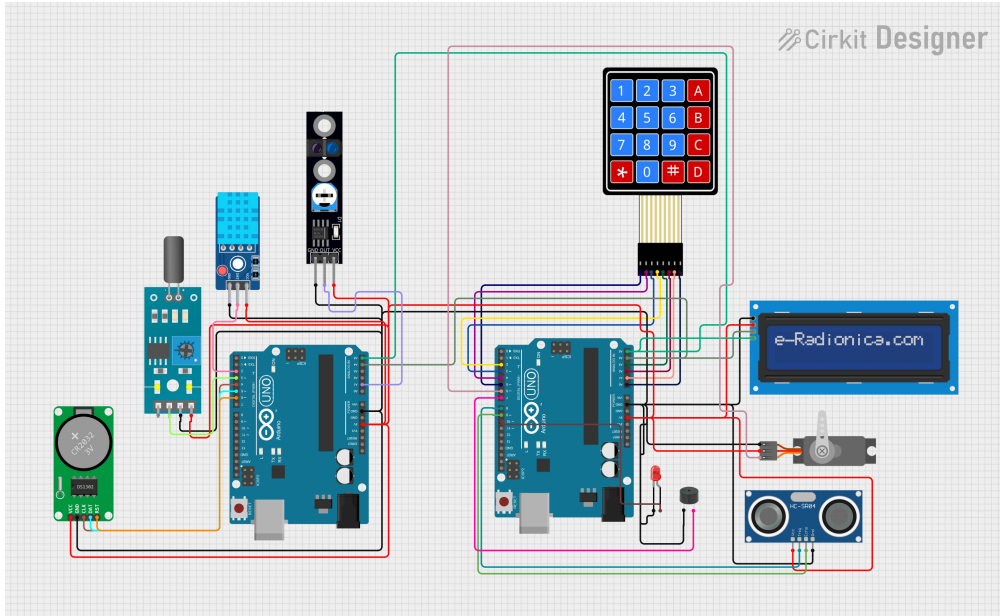
TOTAL: $2 \times 35 + 16 + 7 + 12 + 12 + 5 + 5 + 15 + 20 + 0 + 0 + 2 + 30 + 14 = \sim 210$

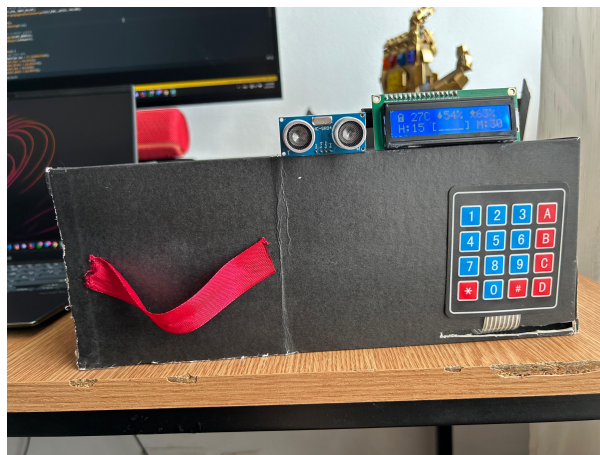
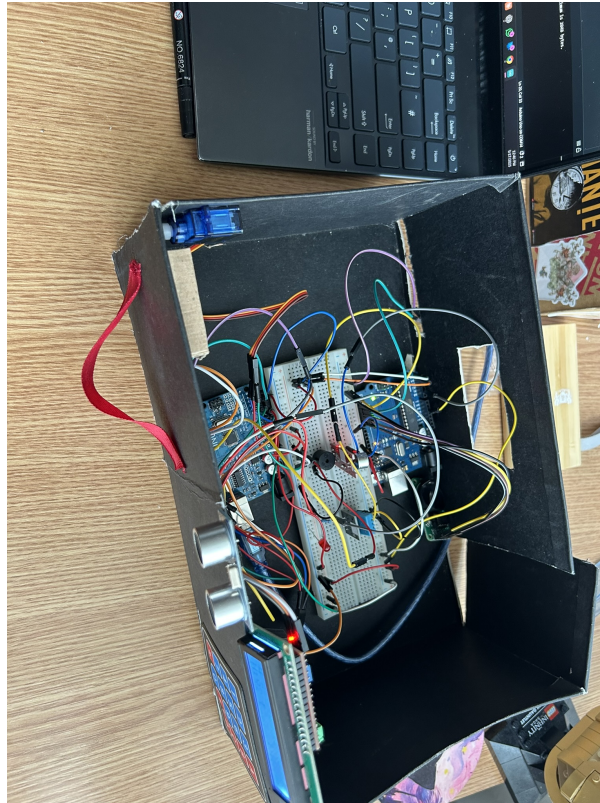
Conexiuni Master

Componentă	Pin Arduino	Descriere
LCD (I2C)	A4 (SDA), A5 (SCL)	Comunicație I2C pentru afișaj 16x2 LCD
Tastatură (Keypad)	2, 3, 4, 5, A0-A3	Tastatură 4x4 - pini digitali + analogici
Servomotor	6	Pin PWM pentru mecanismul de blocare/deblocare
Senzor ultrasonic	8 (Trig), 9 (Echo)	Detectare proximitate cu HC-SR04
Buzzer	7	Alarma pentru acces greșit
LED (stare deschis)	10	Se aprinde când se deblochează ușa
Arduino Slave	A4 (SDA), A5 (SCL)	Comunicație I2C master către slave

Conexiuni Slave

Componentă	Pin Arduino	Descriere
Senzor DHT11	2	Măsoară temperatura și umiditatea
Senzor Sunet KY-037	A1	Citire analogică a nivelului de sunet
Senzor Înclinare SW-530	3 (INT)	Detectează mișcări prin întrerupere
Modul Ceas DS1302	4 (CLK), 5 (DAT), 6 (RST)	Ceas în timp real prin protocol 3-fire
Arduino Master	A4 (SDA), A5 (SCL)	Comunicație I2C în modul slave





Software Design

Software-ul utilizat:

GITHUB: <https://github.com/gabriel-2802/Arduino-Smart-Locker>

Listă Librării Arduino Utilizate

Library Name	Descriere
-----	-----
`Wire.h`	Comunicare I2C între Arduino master și slave
`DHT.h`	Citirea senzorului de temperatură și umiditate DHT11
`RtcDS1302.h`	Controlul modulului RTC (ceas în timp real) DS1302
`LiquidCrystal_I2C.h`	Controlul ecranului LCD 1602 prin magistrala I2C

`Keypad.h`	Gestionarea tastaturii matriciale 4x4
`Servo.h`	Controlul servomotorului SG90
`Arduino.h`	Bibliotecă implicită Arduino, necesară pentru funcționare generală

- Mediu de dezvoltare: ARDUINO IDE
- Acest proiect este inspirat și bazat parțial pe structura și conceptele din proiectul Arduino Electronic Safe realizat de Uri Shaked.

Clase/structuri utilizate:

```
/**
 * Arduino Electronic Safe
 *
 * Copyright (C) 2020, Uri Shaked.
 * Released under the MIT License.
 */
class SafeState {
public:
    SafeState();
    void lock();
    bool unlock(String code);
    bool locked();
    bool hasCode();
    void setCode(String newCode);

private:
    void setLock(bool locked);
    bool _locked;
};

// custom structs, enums
enum LockedPhase { PHASE_IDLE, PHASE_WAIT_INPUT, PHASE_VERIFY,
PHASE_NOTIFICATION };
enum UnlockedPhase { PHASE_UNLOCK_IDLE, PHASE_UNLOCK_OPTIONS,
PHASE_NEW_CODE_SETUP };
enum CodeSubPhase { CODE_ENTER, CODE_CONFIRM };

struct SensorPacket {
    uint8_t hour;
    uint8_t minute;
    uint8_t day;
    uint8_t month;
    uint16_t year;

    float temperature;
    float humidity;
};
```

```
uint8_t soundPercent;  
bool tilt;  
} __attribute__((packed));
```

Funcții Master Arduino

Inițializare și configurare

```
void setup()  
// Inițializează LCD-ul, servo-ul, senzorii, buzzerul, LED-ul, I2C, etc.  
// Afișează mesajul „Welcome” și setează sistemul ca fiind blocat sau  
deblocat în funcție de starea salvată.
```

```
void resetScreenState()  
// Golește ecranul și resetează indicatorii interni și codurile introduse.  
// Este apelată după o perioadă lungă de inactivitate.
```

Comunicare cu Arduino Slave

```
void requestSensorData()  
// Trimite o solicitare I2C către slave (adresă definită) pentru a primi  
structura SensorPacket.  
// Conține ora, temperatura, umiditatea, nivelul de sunet și detectarea de  
mișcare (tilt).
```

Control fizic: Blocare și Deblocare

```
void lock()  
// Mută servomotorul în poziția de blocare și oprește LED-ul.
```

```
void unlock()  
// Mută servomotorul în poziția de deblocare și aprinde LED-ul.
```

```
bool isUserClose()  
// Măsoară distanța cu senzorul ultrasonic (TRIG/ECHO).  
// Dacă este sub un prag predefinit, consideră că utilizatorul este în  
apropiere.
```

Afișare și notificări

```
void showNotification(const String& msg, bool isLockedPhase)
// Afișează un mesaj centrat pe LCD și setează automat faza următoare în
funcție de context (locked/unlocked).
```

```
void displaySensorSummary(bool locked)
// Afișează temperatura, umiditatea, sunetul, ora și simbolul de lacăt
(închis/deschis) pe ecran.
```

Faze când sistemul este blocat (Locked)

```
void safeLockedLogicStep()
// Selectează și apelează funcția corespunzătoare fazei curente de blocare.
```

```
void handleLockedIdle()
// Solicită date de la slave, verifică tilt, afișează senzorii și pregătește
ecranul pentru introducerea codului.
```

```
void handleLockedWaitInput()
// Citește codul PIN de la tastatură și afișează * pentru fiecare cifră.
```

```
void handleLockedNotification()
// După succes sau eșec, așteaptă 1.5 secunde și revine în faza inițială.
```

Faze când sistemul este deblocat (Unlocked)

```
void safeUnlockedLogicStep()
// Selectează și apelează funcția corespunzătoare fazei curente de deblocare.
```

```
void handleUnlockIdle()
// Afișează opțiunile: „A” pentru schimbare cod, „B” pentru blocare.
```

```
void handleUnlockOptions()
// Procesează apăsarea tastelor A sau B pentru a intra în faza de
configurare sau de blocare.
```

```
void handleNewCodeSetup()
// Permite introducerea unui cod nou și confirmarea acestuia într-o singură
funcție.
// Dacă cele două coduri coincid, noul cod este salvat.
```

Buclo principală

```
void loop()  
// Verifică la intervale regulate proximitatea utilizatorului.  
// Solicită date de la slave și verifică senzorul de înclinare (tilt).  
// Apelează logica de blocare sau deblocare doar dacă utilizatorul este  
prezent.
```

Funcții Arduino Slave

Inițializare și configurare

```
void setup() // Inițializează senzorul DHT11 (temperatură/umiditate) //  
Setează ora și data pe modulul RTC // Configurează senzorul de înclinare  
(tilt) și întreruperea externă // Porneste RTC-ul și inițializează  
comunicarea I2C ca slave
```

Funcții de întrerupere și comunicare

```
void onRequest() // Este apelată automat de Arduino Master prin I2C. //  
Trimite structura SensorPacket către Master. // Dacă în acest moment  
tiltDetected este true, se trimite ca atare și apoi se resetează.
```

```
void onTilt() // Funcție apelată la declanșarea întreruperii externe de la  
senzorul SW-520. // Setează flag-ul tiltDetected pe true.
```

Buclo principală (loop)

```
void loop() // Citește data și ora curente de la modulul RTC și  
actualizează packet // Citește temperatura și umiditatea de la senzorul  
DHT11 // Citește semnalul analogic de la microfonul KY-037 și îl transformă  
într-un procent // Copiază flag-ul tilt în packet și apoi îl resetează //  
Așteaptă 250ms înainte de următoarea lectură (pentru eficiență)
```

Rezultate Obținute

- Sistemul de acces a fost implementat cu succes, permițând blocarea/deblocarea fizică prin introducerea unui cod PIN pe tastatura matricială.
- Informațiile despre temperatură, umiditate, sunet și înclinare sunt colectate în timp real de Arduino Slave și transmise cu succes către Arduino Master prin I²C.
- Ecranul LCD afișează datele ambientale și oferă interfață intuitivă pentru utilizator.
- În cazul unei tentative de manipulare a sistemului (detectare de tilt), sistemul reacționează prin activarea buzzerului de alarmă și blochează accesul.
- LED-ul montat indică vizual starea de blocare a seifului (ON = deblocat).

Concluzii

- Comunicarea Master-Slave prin I²C s-a dovedit stabilă și eficientă.
- Utilizarea întreruperilor pentru senzorul de înclinare permite reacții imediate la tentative de manipulare.
- Arhitectura modulară (separare Master-Slave) permite extinderea sistemului cu ușurință
- Proiectul oferă un exemplu funcțional de integrare între senzori digitali/analogici, module RTC și controlere multiple într-un sistem embedded.

Bibliografie/Resurse

Resurse Software <https://wokwi.com/>

Resurse Hardware

Component	Datasheet Link
Arduino Uno	Farnell PDF
LCD 1602 I2C Display	HandsOnTec PDF
SG90 Servo Motor	Imperial College PDF
DHT11 Temperature & Humidity Sensor	AllDataSheet DHT11
KY-037 Sound Sensor	AllDataSheet KY-037
SW-520D Tilt Sensor	Funduino PDF
Rtc DS1302 RTC Module	RS Components PDF

[Export to PDF](#)

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

Permanent link:
<http://ocw.cs.pub.ro/courses/pm/prj2025/ccristi/valentin.carauleanu>



Last update: **2025/05/28 07:45**

