Flipper One

Introducere

Proiectul **Flipper One** reprezinta o mini-replica educationala, realizata pe Arduino UNO, inspirata conceptual de dispozitivul de tip multi-tool folosit pentru pen-testing-ul comunicatiilor simple RF, RFID/NFC si WiFi.

Proiectul este construit in jurul unei placi **Arduino UNO R3 / ATmega328P** si integreaza un ecran OLED, butoane prin expander I2C, LED RGB, module RF 433 MHz, modul PN532 pentru RFID/NFC, modul ESP-01 pentru scanare WiFi si modul MicroSD pentru stocare.

Produsul final a fost impartit in doua firmware-uri separate:

- **Flipper One Main** - firmware-ul principal, care include RFID, WiFi, MicroSD si captura/replay RF intr-un mod raw;
- **Flipper One RF** - firmware pentru RF, bazat pe decodare cu RCSwitch, retransmitere si salvare pe card MicroSD.

Aceasta impartire a fost necesara din cauza limitarilor severe de memorie ale microcontrollerului ATmega328P. Incercarea de a include simultan RFID, WiFi, SD, OLED, RF cu RCSwitch intr-un singur firmware depasea limita de Flash. Prin urmare, am modularizat proiectul in doua coduri care folosesc aceeasi platforma hardware, dar sunt incarcate separat in functie de functia dorita.

Functionalitati principale implementate:

- citirea UID-urilor RFID folosind modulul PN532;
- salvarea UID-urilor RFID pe card MicroSD;
- scanarea retelelor WiFi din apropiere folosind ESP-01 / ESP8266;
- salvarea scanarilor WiFi pe card MicroSD;
- analiza retelelor WiFi detectate: RSSI, securitate, canal, WPS, MAC/BSSID si observatii simple;
- capturarea raw a semnalelor RF 433 MHz si retransmiterea lor in firmware-ul principal;
- capturarea/decodarea RF cu RCSwitch in firmware-ul RF dedicat;
- salvarea codurilor RF decodate in fisiere separate pe cardul MicroSD;
- interfata utilizator pe OLED 128×64;
- control prin 4 butoane conectate prin PCF8574;
- feedback vizual prin LED RGB.

Din punct de vedere al rezultatului final, modulul RFID detecteaza cardurile si salveaza UID-urile pe SD, modulul WiFi realizeaza scanari, salveaza rezultatele si afiseaza o mini-analiza a retelelor. Partea RF are cateva imperfectiuni: captura raw este foarte sensibila la zgomotul receptorului, iar varianta cu RCSwitch este mai curata din punct de vedere software, dar depinde puternic de protocolul telecomenzii si de calitatea semnalului primit.

Descriere generala

Sistemul este organizat in jurul placii **Arduino UNO R3**, bazata pe microcontrollerul **ATmega328P**. Arduino coordoneaza toate modulele externe, citeste butoanele, actualizeaza afisajul OLED si executa functiile alese din meniu.

Interactiunea cu utilizatorul se face prin:

- un ecran **OLED SH1106 128×64 SPI**, folosit in mod text prin biblioteca U8x8;
- 4 butoane conectate la un **PCF8574** pe I2C;
- un **LED RGB** conectat la pini PWM pentru feedback vizual.

Din cauza numarului de pini necesarui pentru conectarea tuturor modulelor am optat spre a fololsi un expander I2C PCF8574 la care am legat cele 4 butoane. Astfel, cele 4 butoane folosesc aceeasi magistrala I2C ca modulul PN532.

Partea de RF este realizata cu doua module de 433 MHz:

- **SRX887** - receptor, conectat pe D2, pin care suporta intrerupere externa INT0;
- **STX882** - emitator, conectat pe D8.

In firmware-ul principal, semnalul RF este tratat raw: programul memoreaza duratele dintre tranzitiile de nivel logic detectate pe D2 si incearca sa le retransmita pe D8. Aceasta abordare, pe Arduino UNO, este limitata de memorie si este foarte sensibila la zgomot, dar apropiata conceptual de un sistem de captura raw,. In firmware-ul RF separat, semnalul este tratat cu biblioteca **RCSwitch**, care incearca sa decodeze protocoale RF simple de tip fixed-code. In loc sa salveze sute de tranzitii, firmware-ul salveaza doar valoarea decodata, numarul de biti, protocolul si lungimea pulsului. Aceasta metoda este mult mai eficienta ca memorie, dar functioneaza doar pentru protocoalele suportate de RCSwitch si pentru telecomenzi compatibile.

Modulul **PN532 RFID/NFC** este conectat pe I2C si este folosit pentru citirea UID-urilor cardurilor. Pentru a reduce consumul de memorie, in firmware-ul principal am folosit o implementare minimala pentru comenzile PN532 necesare: initializare, configurare SAM si citire UID.

Modulul **ESP-01 / ESP8266** este conectat prin SoftwareSerial. Arduino ii trimite comenzi AT, iar ESP-ul raspunde cu lista retelelor WiFi detectate. Datele sunt salvate pe cardul MicroSD in fisierul `WIFIDB.TXT`, apoi sunt citite si analizate din fisier, nu pastrate integral in RAM din cauza constraint-urilor de memorie.

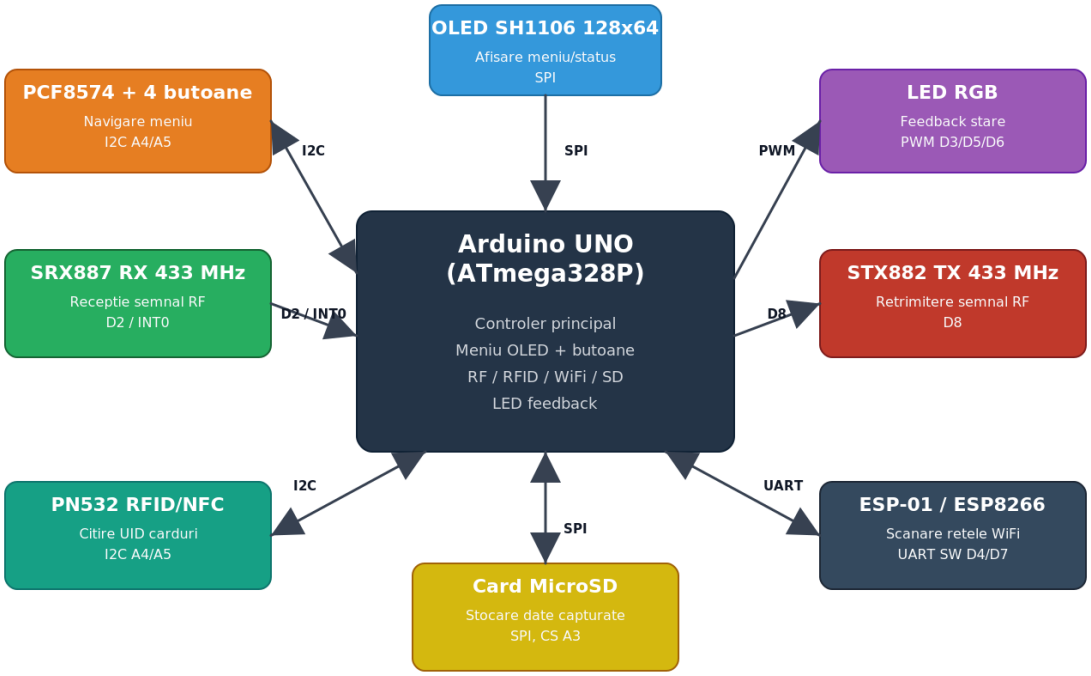
Modulul **MicroSD** este conectat pe SPI si este folosit pentru stocarea persistenta a datelor:

- UID-uri RFID in fisiere `.UID`, indexate in `UIDIDX.TXT`;
- scanari WiFi in `WIFIDB.TXT`;
- coduri RF decodate in firmware-ul RF, in fisiere `.RFC`, indexate in `RFIDX.TXT`.

Logica generala a dispozitivului este bazata pe o masina de stari:

- **Main Menu** - utilizatorul alege modul de lucru;
- **RF Capture / RF Emit** - captura si retransmitere RF;
- **RFID Scan / Save UID** - citire si salvare UID;
- **WiFi Scan / View Networks** - scanare si afisare retele WiFi;
- **SD Browser** - vizualizarea fisierelor salvate;

- **Status** - afisarea starii sistemului.



Hardware Design

Ambele variante de firmware folosesc aceeasi placa Arduino UNO si aceleasi conexiuni fizice. Diferenta este la nivel software: firmware-ul incarcat pe Arduino decide ce module sunt folosite efectiv.

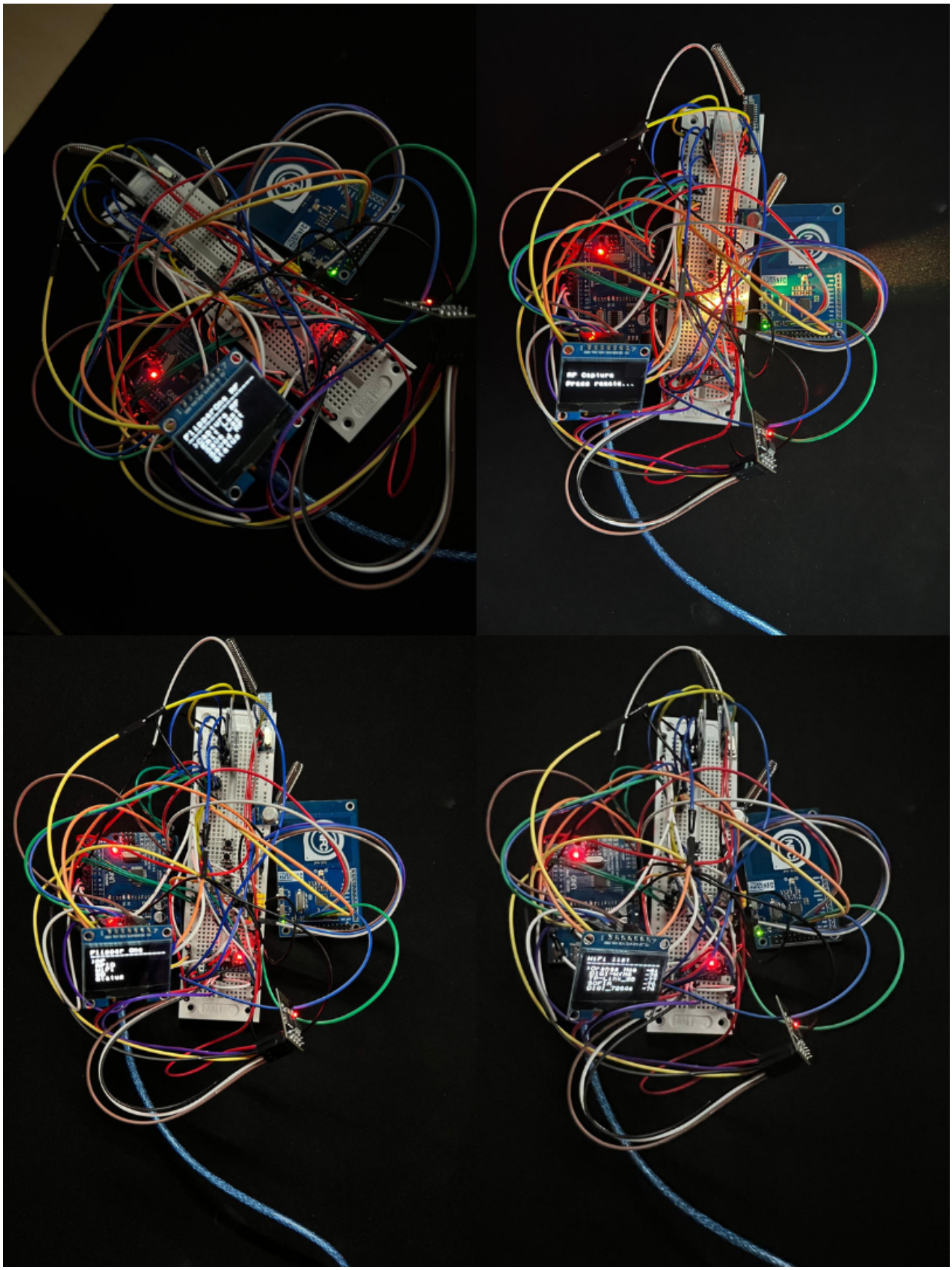
Montajul este realizat pe breadboard si include urmatoarele module:

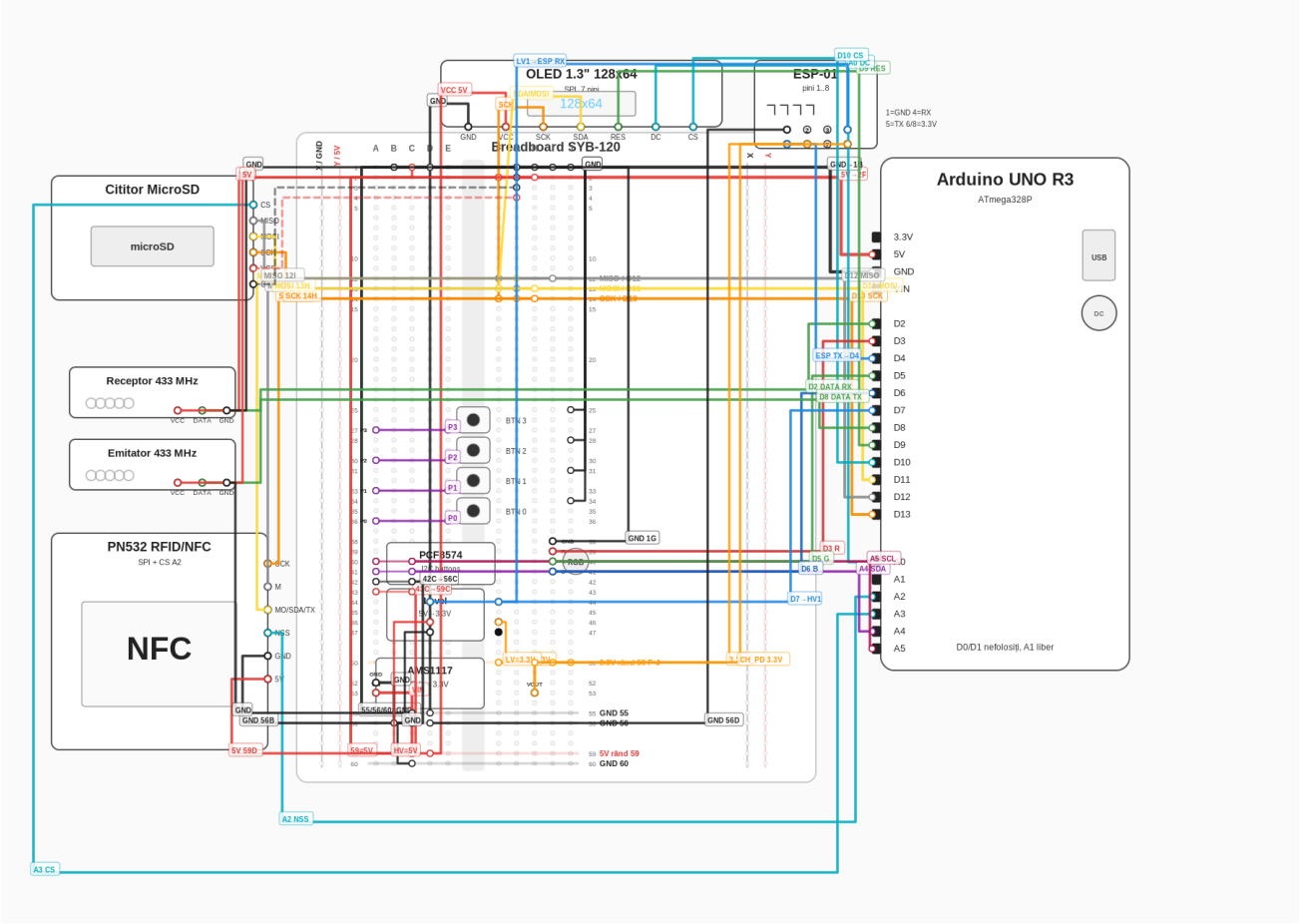
- Arduino UNO R3 / ATmega328P;
- OLED SH1106 128×64 pe SPI;
- receptor RF SRX887 433 MHz;
- emitator RF STX882 433 MHz;
- modul PN532 RFID/NFC pe I2C;
- expander PCF8574 pentru butoane pe I2C;
- 4 butoane tactile;
- LED RGB;
- ESP-01 / ESP8266 pentru scanare WiFi;
- regulator AMS1117 3.3V pentru ESP-01;
- modul MicroSD pe SPI.

Alimentarea este impartita astfel:

- **5V** pentru Arduino, OLED, PN532, PCF8574, module RF si MicroSD;
- **3.3V** pentru ESP-01, generat prin AMS1117;
- **GND comun** pentru toate modulele.

PN532 si PCF8574 sunt pe aceeasi magistrala I2C, iar OLED-ul si MicroSD-ul impart magistrala SPI. Pentru a evita conflictele pe SPI, fiecare dispozitiv are pinul sau de chip select: OLED pe D10 si SD pe A3.





Componente folosite

Componenta	Rol in proiect	Interfata / observatii
Arduino UNO R3 / ATmega328P	Unitatea centrala a sistemului	GPIO, PWM, SPI, I2C, SoftwareSerial
Breadboard SYB-120	Suport pentru prototipare si distributie alimentare	5V, 3.3V, GND
OLED 1.3'' SH1106 128×64	Afisarea meniurilor si a rezultatelor	SPI hardware, U8x8 fara framebuffer
SRX887 433 MHz	Receptie RF	DATA pe D2 / INT0, CS la GND in modul activ
STX882 433 MHz	Transmisie RF	DATA pe D8
LED RGB KY-016	Feedback vizual	PWM pe D3, D5, D6
PCF8574	Extinderea pinilor pentru butoane	I2C pe A4/A5
4 butoane tactile	Navigare in meniu	P0-P3 pe PCF8574
PN532 RFID/NFC	Citire UID carduri/tag-uri	I2C, reset pe A1
ESP-01 / ESP8266	Scanare retele WiFi	SoftwareSerial D4/D7, alimentare 3.3V
AMS1117 3.3V	Regulator pentru ESP-01	VIN 5V, VOUT 3.3V
Modul MicroSD	Stocare persistenta	SPI, CS pe A3

Maparea pinilor Arduino

Pin Arduino	Conectat la	Justificare
5V	OLED, PN532, PCF8574, RF, MicroSD, AMS1117 VIN	Alimentare module compatibile 5V
GND	Toate modulele	Referinta comuna
D2	DATA receptor SRX887	Pin cu intrerupere externa INT0
D3	LED RGB - R	PWM
D4	ESP-01 TX catre Arduino	SoftwareSerial RX
D5	LED RGB - G	PWM
D6	LED RGB - B	PWM
D7	ESP-01 RX de la Arduino	SoftwareSerial TX
D8	DATA emitator STX882	Iesire digitala pentru transmisie RF
D9	RST OLED	Reset display
D10	CS OLED	Chip select SPI OLED
D11	MOSI SPI	Date catre OLED si MicroSD
D12	MISO SPI	Date de la MicroSD
D13	SCK SPI	Clock SPI
A0	DC OLED	Selectare comanda/date OLED
A1	RESET PN532	Reset hardware PN532
A2	IRQ PN532	Pin rezervat pentru IRQ PN532
A3	CS MicroSD	Chip select SD
A4	SDA PCF8574 + PN532	Magistrala I2C
A5	SCL PCF8574 + PN532	Magistrala I2C

Conexiuni

OLED SPI

Pin OLED	Arduino
GND	GND
VCC	5V
SCK	D13
SDA / MOSI	D11
RES	D9
DC	A0
CS	D10

MicroSD

Pin MicroSD	Arduino
VCC	5V
GND	GND
SCK	D13
MOSI	D11
MISO	D12
CS	A3

PN532 RFID/NFC

Pin PN532	Arduino
VCC / 5V	5V
GND	GND
SDA	A4
SCL	A5
RST	A1
IRQ	A2

PCF8574 si butoane

Pin PCF8574	Conexiune
VCC	5V
GND	GND
SDA	A4
SCL	A5
P0	Buton UP
P1	Buton DOWN
P2	Buton SELECT
P3	Buton BACK

RF 433 MHz

Modul	Pin	Arduino / conexiune
SRX887	VCC	5V
SRX887	GND	GND
SRX887	CS	GND pentru activare
SRX887	DATA	D2 / INT0
SRX887	ANT	Antena 433 MHz
STX882	VCC	5V
STX882	GND	GND
STX882	DATA	D8
STX882	ANT	Antena 433 MHz

ESP-01

Pin ESP-01	Conectat la	Observatii
VCC	3.3V AMS1117	ESP-ul nu se alimenteaza direct din 5V
GND	GND comun	Masa comuna
TX	D4 Arduino	Arduino primeste prin SoftwareSerial
RX	D7 Arduino	Arduino trimite comenzi AT
CH_PD / EN	3.3V	Activare modul

Software Design

Mediu de dezvoltare

Proiectul a fost dezvoltat in **Visual Studio Code** folosind **PlatformIO** si toolchain-ul AVR-GCC pentru Arduino UNO. Framework-ul folosit este Arduino, deoarece proiectul integreaza mai multe biblioteci si periferice externe.

Biblioteci folosite:

- `U8g2 / U8x8` - afisare text pe OLED SH1106 fara framebuffer, pentru economie de RAM;
- `SD` - acces la cardul MicroSD;
- `Wire` - comunicatie I2C cu PCF8574 si PN532;
- `SPI` - comunicatie cu OLED si MicroSD;
- `SoftwareSerial` - comunicatie seriala cu ESP-01;
- `RCSwitch` - folosit doar in firmware-ul RF dedicat.

Pentru a evita depasirea memoriei Flash, proiectul este impartit in doua environment-uri PlatformIO:

```
[env:main_demo]
; Flipper One Main: RFID + WiFi + SD + raw RF

[env:rf_demo]
; Flipper One RF: RF RCSwitch + SD
```

Motivul modularizarii in doua firmware-uri

Microcontrollerul ATmega328P are doar 32 KB Flash si 2 KB SRAM. In timpul dezvoltarii, incercarea de a include toate functiile intr-un singur firmware a dus la depasirea limitei de Flash sau la instabilitate runtime, mai ales cand erau folosite simultan:

- biblioteca SD;
- biblioteca RCSwitch;
- SoftwareSerial pentru ESP;
- driverul PN532;
- meniurile OLED;
- analiza WiFi;

Solutia finala a fost separarea proiectului in doua coduri:

- firmware-ul **main_demo**, pentru functionalitate RFID/WiFi/SD si raw RF;
- firmware-ul **rf_demo**, pentru functionalitate specializata RF cu RCSwitch si salvare pe SD.

Firmware 1: Flipper One Main

Firmware-ul principal si cel mai “product ready” din cele 2:

- captura RF raw;
- retransmitere RF raw;
- scanare RFID;
- salvare UID RFID pe card MicroSD;
- scanare WiFi cu ESP-01 si salvare autoamta in `WIFIDB.TXT` a retelelor;
- analiza WiFi pe OLED;
- browser SD pentru fisierele salvate;
- status general.

Captura RF raw

In modul raw, receptorul RF este conectat la D2, iar codul foloseste intreruperea externa INT0 pentru a masura timpul dintre tranzitiile semnalului. Fiecare durata este salvata intr-un buffer de tip `uint16_t`.

Avantajul acestei abordari este ca nu presupune cunoasterea protocolului. In teorie, poate captura orice secventa de impulsuri OOK/ASK suficient de simpla.

Dezavantajul major este sensibilitatea la zgomot. Receptorul SRX887 poate genera tranzitii chiar si in lipsa unui semnal real, din cauza AGC-ului si a zgomotului RF din mediu. Astfel, bufferul se poate umple cu zgomot inainte ca utilizatorul sa apese telecomanda. Din acest motiv, captura RF este cea mai instabila parte a proiectului.

RFID

Modulul PN532 este controlat prin I2C. Pentru economie de memorie, nu este folosita biblioteca completa Adafruit PN532, ci un driver minimal care implementeaza doar comenzile necesare:

- `GetFirmwareVersion` pentru verificarea prezentei modulului;
- `SAMConfiguration` pentru configurarea modului normal;
- `InListPassiveTarget` pentru citirea cardurilor ISO14443A.

Dupa detectarea unui card, UID-ul este afisat pe OLED si poate fi salvat pe SD intr-un fisier `.UID`. Numele fisierului este generat automat, iar indexul fisierelor salvate este pastrat in `UIDIDX.TXT`.

Exemplu format fisier RFID:

```
LEN=4
UID=04A1B2C3
```

WiFi

Pentru scanarea WiFi este folosit un modul ESP-01 / ESP8266 cu firmware AT. Arduino comunica prin

SoftwareSerial si trimite comanda:

```
AT+CWLAP
```

Raspunsurile de forma `+CWLAP:(...)` sunt salvate automat pe cardul MicroSD in fisierul `WIFIDB.TXT`. Am luat aceasta decizie de implementare pentru a economisi RAM, retelele nefiind pastrate intr-un vector mare, ci scrise direct pe SD si apoi citite la nevoie.

Date analizate pentru fiecare retea:

- SSID;
- RSSI;
- BSSID / MAC;
- canal;
- tip securitate;
- WPS;
- cipher pairwise/group;
- protocol b/g/n;
- observatii simple despre calitatea semnalului si congestie.

SD Browser

Firmware-ul principal include un browser simplu de fisiere salvate. Acesta citeste indexul linie cu linie si afiseaza fisierele pe OLED. Pentru a evita directoarele si operatii costisitoare pe SD, sunt folosite fisiere index:

- `UIDIDX.TXT` pentru UID-uri RFID;
- `RFIDX.TXT` pentru capturi RF raw, in varianta initiala;
- `WIFIDB.TXT` pentru scanarea WiFi curenta.

Firmware 2: Flipper One RF

Firmware-ul RF este o varianta specializata pentru testarea RF 433 MHz. Acesta include:

- OLED;
- PCF8574 + butoane;
- LED RGB;
- STX882 / SRX887;
- RCSwitch;
- MicroSD pentru salvarea codurilor RF decodate.

Nu include PN532, ESP-01 sau analiza WiFi, tocmai pentru a pastra suficient spatiu pentru RCSwitch, pentru logica de salvare RF si ca devena redundant.

Meniul firmware-ului RF:

- **Capture RF** - asteapta un cod RF compatibil RCSwitch;
- **Emit Last** - retransmite ultimul cod capturat sau incarcat;

- **Save Last** - salveaza codul RF curent pe SD;
- **Saved RF** - listeaza fisierele RF salvate si permite incarcarea lor;
- **Status** - afiseaza valoarea, bit length, protocolul, delay-ul si starea SD.

Captura RF cu RCSwitch

In acest firmware, in loc sa se salveze tranzitiile brute, biblioteca RCSwitch incearca sa decodeze semnalul. Daca reuseste, firmware-ul retine:

- `value` - valoarea numerica decodata;
- `bits` - numarul de biti;
- `protocol` - protocolul detectat de RCSwitch;
- `delayUs` - lungimea pulsului.

Aceasta reprezentare este mult mai compacta decat raw pulse capture. Un cod RF poate fi salvat in cativa bytes, in timp ce varianta raw are nevoie de un buffer de durate.

Salvarea RF pe SD

In firmware-ul RF, codurile decodate pot fi salvate pe card MicroSD in fisiere separate, similar cu salvarea UID-urilor RFID. Exista un fisier index `RFIDX.TXT`, iar fiecare cod RF este salvat intr-un fisier `.RFC`.

La selectarea unui fisier din meniul **Saved RF**, codul este incarcat in RAM si poate fi retransmis prin **Emit Last**.

Limitari RF

Desi firmware-ul RF este mai eficient si mai bine structurat decat captura raw, rezultatele practice nu au fost perfecte. Uneori receptorul produce zgomot si RCSwitch nu decodeaza nimic valid. Alteori semnalul telecomenzii nu este compatibil cu protocoalele suportate. Astfel, partea RF ramane cea mai instabila parte a proiectului.

Folosirea conceptelor din laboratoare

Proiectul foloseste mai multe concepte din laboratoarele de Proiectarea cu Microprocesoare. Am incercat sa le aplic practic, nu doar separat, ci intr-un sistem in care toate modulele trebuie sa functioneze impreuna:

- Laboratorul 0 - GPIO - configurarea pinilor digitali pentru LED RGB, RF TX, chip select-uri SPI si semnale de control.
- Laboratorul 1 - UART - comunicatia cu ESP-01 prin SoftwareSerial si comenzi AT.

- Laboratorul 2 - Intreruperi - folosirea pinului D2 / INT0 pentru receptia RF raw si pentru receptia RCSwitch.
- Laboratorul 3 - Timere si PWM - controlul LED-ului RGB cu `analogWrite()` pe pinii D3, D5 si D6.
- Laboratorul 5 - SPI - comunicatia cu OLED-ul si cardul MicroSD pe aceeasi magistrala SPI, cu CS-uri separate.
- Laboratorul 6 - I2C - comunicatia cu PCF8574 si PN532 pe A4/A5.

Pe langa laboratoarele propriu-zise, am folosit si concepte de organizare software: masina de stari pentru meniuri, fisiere index pe SD, separarea proiectului in doua firmware-uri si optimizarea memoriei pentru ATmega328P.

Optimizari importante

Pentru a rula pe Arduino UNO, au fost necesare mai multe optimizari:

- folosirea U8x8 in loc de un framebuffer complet pentru OLED;
- salvarea scanarilor WiFi direct pe SD, nu in RAM;
- folosirea de stringuri in PROGMEM;
- implementare minimala PN532;
- limitarea bufferelor locale;
- dezactivarea Serial debug in build-ul final;
- modularizarea in doua firmware-uri separate.

Fara aceste modularizari proiectul ar fi fost prea mare pentru ATmega328P. In momentul de fata, firmware-ul main atinge 99.5% din memoria flash.

Rezultate obtinute

In urma implementarii, asamblarii si testarii FlipperOne, s-au obtinut urmatoarele rezultate ce confirma atingerea si depasirea obiectivelor initial propuse:

1. RFID:
   - PN532 este initializat corect pe I2C;
   - cardurile/tag-urile ISO14443A sunt detectate;
   - UID-ul este afisat corect pe OLED;
   - UID-ul poate fi salvat pe MicroSD;
   - fisierele salvate pot fi regasite prin index.
2. Wi-Fi, a depasit asteptarile:
   - ESP-01 raspunde la comenzile AT, scaneaza retelele din jur si le salveaza automat in `WIFIDB.TXT`;
   - lista retelelor poate fi afisata pe OLED;
   - detaliile fiecarei retele pot fi analizate;
   - analiza include RSSI, securitate, canal, MAC/BSSID, WPS si observatii simple.
3. RF raw, foarte sensibil la zgomot:
   - Captura RF raw este instabila in practica. Uneori receptorul capteaza foarte mult zgomot, alteori semnalul util nu poate fi separat clar de zgomot. Aceasta problema vine atat din natura

receptorului RF, cat si din limitarile Arduino UNO.
    - Raw RF ramane insa interesant ca experiment, deoarece arata cum pot fi masurate tranzitiile folosind intreruperi, dar nu este suficient de stabil pentru o demonstratie sigura de tip “captureaza telecomanda si retransmite”.
4. RF cu RCSwitch:
    - Firmware-ul RF cu RCSwitch este mai curat si mai eficient, dar nu rezolva complet problema. Daca telecomanda foloseste un protocol suportat si semnalul este curat, codul poate fi decodat si salvat. Daca semnalul este zgomotos sau protocolul nu este compatibil, nu apare nicio captura valida.
    - Totusi, modulul RF dedicat ar trebui sa aiba urmatoarele functionalitati functionale: captura RCSwitch atunci cand semnalul este compatibil, afisarea codului decodat, retransmiterea codului decodat, salvarea codului pe SD in fisiere `.RFC`, listarea si incarcarea codurilor RF salvate.

Probleme intalnite

Cele mai mari probleme au fost:

- zgomotul RF produs de receptor in lipsa unui semnal util;
- memoria Flash foarte limitata;
- memoria RAM limitata, mai ales cand SD, SoftwareSerial si OLED sunt folosite simultan;
- conflictele intalnite pe parcursul motanrii pe SPI intre OLED si MicroSD;

Concluzii

Proiectul a iesit mai bine decat ma asteptam, mai ales avand in vedere numarul mare de module conectate la un Arduino UNO. Cele mai stabile si reusite parti sunt RFID-ul si WiFi-ul: UID-urile sunt citite si salvate corect, iar scanarea WiFi impreuna cu analiza retelelor functioneaza foarte bine.

Partea RF a fost cea mai dificila. Captura raw este foarte expusa la noise, iar varianta cu RCSwitch, desi mult mai eficienta, depinde de protocoalele suportate si de calitatea semnalului. Din acest motiv, RF-ul ramane o functionalitate partiala si experimentala in proiect. Pe langa RF, resursele foarte limitate ale lui ATmega328P au adus mai multe **constrangerile de memorie** greu de rezolvat, combinatia dintre OLED, SD, ESP, PN532, RF Raw/RCSwitch a fortat optimizari serioase. In final, modularizarea in doua firmware-uri separate a fost solutia corecta: firmware-ul principal pentru RFID/WiFi/SD/RFraw si firmware-ul RF pentru captura/decodare/salvare RF.

Link-uri si Resurse

Pentru documentarea si realizarea acestui proiect, am consultat urmatoarele resurse tehnice si bibliografii:

Cod

- **GitHub:** FlipperOne

Laboratoare PM

Resurse utilizate pentru implementarea interfetelor, perifericelor si comunicatiilor cu modulele externe:

- Lab 0 - GPIO
- Lab 1 - UART
- Lab 2 - Intreruperi
- Lab 3 - Timere / PWM
- Lab 5 - SPI
- Lab 6 - I2C / TWI

Datasheet-uri si Documentatii Componente

Documentatia tehnica oficiala a componentelor si bibliotecilor utilizate:

- **Microcontroler:** ATmega328P Datasheet
- **Arduino UNO:** Arduino UNO Rev3 Documentation
- **PlatformIO:** PlatformIO Documentation
- **OLED SH1106 / U8g2-U8x8:** U8g2 / U8x8 Library
- **RCSwitch:** RCSwitch Library
- **ESP8266 / ESP-01 AT Commands:** ESP8266 AT Instruction Set
- **PN532 RFID/NFC:** PN532 NXP User Manual
- **Arduino SD Library:** SD Library Documentation

Export to PDF