Vintage Phone

Introducere

Acest proiect constă în readucerea la viață a unui telefon fix cu rotiță (EM-72 fabricat în 1977) și îmbunătățirea acestuia prin adăugarea unor funcționalități noi. Principalele funcții pe care le va avea:

- inițierea și primirea apelurilor (GSM)
- memorarea numerelor de telefon pentru apelare rapidă
- reținerea unui istoric cu ultimele apeluri
- comenzi rapide

Descriere generală

Planul meu este să păstrez carcasa/design-ul telefonului cât mai intactă pentru look-ul "vintage", prin urmare noile componente se vor regăsi, cel mai probabil, într-o carcasă noua, separată. Va trebui să verific starea telefonului (de ex. dacă microfonul și difuzorul mai funcționează) și în funcție de surprizele găsite, să achiziționez noile componente.Pentru comunicarea in rețeaua 2G voi folosi modulul GSM-SIM800L iar pentru microcontroller voi folosi un Arduino Pro Mini (Atmega328p) împreună cu un display OLED pentru vizualizarea datelor. Stocarea datelor precum numere de telefon, istoricul apelurilor se va face printr-o memorie EEPROM (24FC64).



Hardware Design

Listă componente:

- GSM SIM800L
- Arduino Pro Mini
- Ecran OLED 0.96" I2C
- Modul DC-DC Step-Down MP1584EN
- Microfon de la nişte căşti
- Difuzorul original
- Buzzer
- Rezistente 10KOhm



Schemă electrică:



Testarea parțială:



Modulul SIM800L comunică cu Arduino prin interfața serială (USART) (am folosit un Arduino UNO pt testare, voi schimba cu Pro Mini), iar interacționarea cu acesta se face prin comenzi AT. Sincronizarea baud-rate-ului se face automat la trimiterea primului mesaj catre chip. I-am adăugat un condensator între VCC si GND (1000 μ F), pentru stabilitate la burst-uri de 2A (vezi jurnal pentru detalii). Pentru conexiunea dintre pinul RX de la chip si Arduino am folosit un divizor de tensiune pentru a coborî tensiunea la 3.3V

Ecranul OLED și memoria EEPROM comunică prin I2C cu Arduino, iar pentru coborârea tensiunii de la 9V la 4V (tensiunea recomandată pentru SIM800L) am folosit modulul step-down MP1584EN.

Astfel, am reușit să inițiez și să primesc apeluri telefonice/SMS-uri. Am testat de mai multe ori, totul pare să funcționeze cum trebuie, singura problemă este că persoana cu care vorbesc își aude propria voce, însă nu am gasit o rezolvare.

Mecanismul de formare a numarului l-am descifrat cu ajutorul unui multimetru si functioneaza in felul urmator:

- Avem 3 fire care ies de la rotita cu urmatoarele culori : **R**osu, **G**alben, **V**erde. Firul **R** va fi conectat la **GND**, celelalte 2 vor fi conectate la INPUT cu rezistente de PULL-UP. Pentru a semnala formarea numarului (momentul in care rotita incepe sa se invarta), firul **R** si firul **G** vor avea continuitate (la input vom avea LOW, discontinuitate), iar cand rotita ajunge la final, continuitatea nu mai este. Pentru a identifica cifra formata, firul **R** si **V** sunt in general in continuitate dar formeaza pauze de discontinuitate de cca 50ms, la fiecare cifra parcursa, exemplu: daca formam cifra 7, vom avea 7 pauze de discontinuitate de cca 50ms. Acest lucru este simplu de interpretat in cod deoarece Arduino va vedea 7 schimbari de semnal LOW-HIGH.

Pentru a vedea daca receptorul este ridicat sau nu, am lipit 2 fire de mecanismul care tine receptorul (IN si GND, cand receptorul este pus la IN vom avea HIGH iar cand este ridicat, LOW.



Software Design

Link GitHub : https://github.com/Alexe1289/proiect_pm

Mediu de dezvoltare și limbaj de programare:

Arduino IDE — mediu principal de programare Limbaj: C++ (Arduino Sketch)

Librării folosite:

- SoftwareSerial.h - folosită pentru a avea 2 conexiuni seriale (una la SIM800L, cealalta la Serial Monitor pentru debugging)

- Adafruit_GFX.h folosită pentru a afișa pe ecranul OLED imagini (bitmaps)
- Adafruit_SH110X.h folosită pentru a interacționa cu display-ul OLED
- Wire.h folosită pentru comunicarea prin I2C cu display-ul

Descrierea codului:

- Pentru inceput, in functia de setup(), setez rezistentele de pull-up, conexiunea USART intre modul SIM800L si Arduino si comunicarea I2C cu display-ul. Dupa care pentru a stabili baudrate-ul cu SIM800L (lucru care se face automat la primul mesaj transmis), trimit un mesaj "AT". - In functia de loop, putem fi in urmatoarele stari:

- 1. a fost ridicat receptorul ceea ce inseamna o schimbare a tensiunii pin-ului (LOW-HIGH, HIGH-LOW), deci:
 - 1. raspundem unui apel primit prin trimiterea comenzii "ATA"
 - 2. urmeaza sa formam un numar
- 2. receptorul a fost pus jos ceea ce inseamna exact opusul al celor de mai sus, mai exact:
 - 1. inchidem apelul curent prin trimiterea comenzii "ATH"
 - 2. se reseteaza vectorul cu numere formate

- Functioneaza exact cum am descris in partea hardware. Daca rotita se invarte, atunci pe pin-ul 5 vom primi LOW, deci putem incepe sa adaugam intr-un vector de caractere numerele formate. Numarul de schimbari HIGH-LOW pe pin-ul 6, semnifica ce numar a fost format. Dupa ce rotita a ajuns la final (pin 5 este din nou HIGH), afisam pe display numerele din vector. Cand se ajunge la 10 numere se initiaza apelul prin comanda "**ATD+(NUMAR_TELEFON);**"

Inainte de formarea numarului se verifica semnalul si accesul in retea in felul urmator:

Se trimite o comanda "AT+CSQ" pentru a vedea calitatea semnalului. Raspunsul va fi de forma "
 +CSQ: <rssi>,<ber>"

```
<rssi> - Received Signal Strength Indication (nivelul semnalului):
0 → -113 dBm (semnal foarte slab)
31 → -51 dBm (semnal excelent)
99 → necunoscut / nu se poate determina
<ber> - Bit Error Rate (rata de erori):
```

- Pentru vizualizarea semnalului, am luat niste iconite de pe acest site si apoi le-am convertit in bytearrays(date constante care vor fi retinute in memoria **flash**) cu ajutorul image2cpp, apoi am apelat functia "**.drawBitmap**"

- Se trimite o comanda "**AT+CREG?**" pentru a vedea daca avem acces in retea (in cazul meu, Orange). Daca primim: +CREG: 0,1, e totul ok.

- La primirea unui apel, generez prin buzzer 3 bip-uri scurte de 1000HZ folosind functia **tone**.

Rezultate Obținute



Videoclip de prezentare

Jurnal

#1 - am avut ceva probleme cu modulul deoarece nu reușea să se conecteze în rețea. Își lua "RESET" la fiecare 8s, iar în urma documentării, am aflat că placa are o problema din fabrică. Condensatorul între VCC si GND este subdimensionat (are 100 μ F, iar producătorul chip-ului recomandă 1000 μ F) și nu făcea față la burst-uri de 2A. Problema mai persistă uneori, așa că probabil voi pune un condensator mai mare.

#2 - Am schimbat microfonul cu unul de la căști deoarece cel original nu părea să meargă. Difuzorul, însă, a funcționat. O poza cu improvizația:



#3 - Mecanismul soneriei lipsește așa că voi pune un buzzer, provizoriu.

#4 - Am adaugat un buton de "panica", da reset la arduino si SIM800L, deoarece existau situatii cand modulul GSM nu reusea sa se conecteze in retea.

#5 - Am renuntat la memoria EEPROM pentru a retine apeluri, din cauza problemelor aparute. Mi-am dat seama ca Arduino-ului nu ii place comunicarea SoftwareSerial + I2C (se blocheaza). Asta inseamna ca nu puteam sa fac debug la ce afisam

#6 - Am lipit firele si componentele pe perfboard pentru mai multa stabilitate.



Bibliografie/Resurse

https://www.flaticon.com/

×

https://javl.github.io/image2cpp/

https://lastminuteengineers.com/sim800l-gsm-module-arduino-tutorial/

https://www.instructables.com/How-to-Display-Images-on-OLED-Using-Arduino/

GPT + Google

Export to PDF

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

Permanent link: http://ocw.cs.pub.ro/courses/pm/prj2025/ccristi/alexandru.beiu

Last update: 2025/05/25 23:56