Facial Recognition Locker

Autor

Apetroaie Razvan-Mihai

Introducere

- Metodele tradiționale de a obține accesul sunt metode de tipul “something you have” (descuierea ușii folosind o cheie) și “something you know” (parola propriului cont). Scopul proiectului este implementarea unei metode “something you are”, mai exact recunoașterea facială, oferind accesul fără necesitatea de a memora un cod sau a deține un obiect anume.
- Proiectul își propune controlarea unei încuietori electrice pe baza capturilor realizate de o camera atașată și a rezultatului dat de un server extern în urma procesării imaginilor.

Descriere generală

Ansamblul va avea atașată o cameră VGA ce va realiza capturi la apăsarea butonului. Imaginile vor fi trimise la un server extern folosind un modul Ethernet, unde acestea vor avea rol de input pentru modelul de recunoaștere facială implementat, urmând că răspunsul să fie transmis plăcuței Arduino. Microcontrolerul va putea atunci să acționeze asupra incuietoarei prin intermediul unui releu.

Hardware Design

Piese necesare

- Arduino UNO R3
- Modul extern Ethernet
- Cameră VGA
- Modul releu 12V
- Incuietoare electrică 12V
- Push button
- Cablu UTP
- Rezistențe 2×4,7k, 2x10k

- Conectori de pini
- Placă PCB de prototipare

Schema electrică

Inițial, proiectul își propusese să adauge un card SD pentru stocarea imaginilor și un numpad pentru o metodă alternativă de autentificare prin introducerea parolei, însă s-a renunțat la acestea din moment ce toți pinii pentru uz general au fost folosiți (camera necesită conectarea a 18 pini) și a fost necesară și remaparea pinilor.
Microcontroller-ul comunica cu shield-ul Ethernet prin pinii ICSP, fiind astfel necesară nefolosirea pinilor 10-13 (sunt conectați intern la pinii ICSP). Analog, pinii SCL și SDA corespund cu pinii A5, respectiv A4.
Pinii RX și TX sunt folosiți pentru buton și releu, la releu fiind de asemenea atașat un alimentator de 12V pentru încuietoare.

Software Design

Atât codul scris pentru plăcuța Arduino, cât și cel folosit pe server au fost urcate folosind git. Pentru programarea pe microcontroller s-a folosit Arduino IDE, iar pe server se găsește un script bash ce lansează 3 terminale care rulează în paralel script-uri Python.

Arduino IDE

Codul sursă se află în folder-ul “Ethernet_Camera”. Implementarea s-a început de la codul oferit de următorul ghid, care la rândul lui a folosit următoarea sursă. La apăsarea butonului, se va trimite un batch de 5 imagini, din moment ce unele imagini pot avea probleme de luminozitate sau poziționare.
La inițializare, se configurează pinul 6 ca și PWM clock pentru camera VGA și se configurează adresele MAC și IP, urmate de port-ul pentru transmiterea UDP. În urma testelor nu s-a observat o pierdere semnificativă a datelor în comparație cu protocolul TCP, UDP oferind totodată o viteză mai mare de transmitere a pachetelor.
Pe măsură ce se citesc pixelii, aceștia sunt salvați într-un buffer de dimensiune 1280 bytes (4 linii din imagine, a fost necesară optimizarea memoriei din moment ce SRAM-ul plăcuței Arduino are capacitate maximă de 2048 bytes), urmând a se transmite pachetul la umplerea acestuia.
Microcontroller-ul va primi un pachet UDP în caz afirmativ, urmând să deschidă încuietoarea pentru 2 secunde.

Python

Prin rularea script-ului run.sh se lansează în execuție script-urile python3:

- udp_server - șterge imaginile salvate anterior și preia datele de la microcontroller, salvându-le în format .txt în urma primirii unui fișier întreg
- convert - preia datele din fișierul .txt, după care le scrie sub formă RGB și adaugă header-ul BMP, urmând să salveze rezultatul într-un fișier bmp
- check_identity - compară encodarea imaginii bmp obținute cu encodarea imaginii de referință folosind librăria “face_recognition”. Dacă funcția returnează True, script-ul va trimite pachete UDP pentru batch-ul curent de imagini. Script-ul poate fi dezvoltat să compare captura cu mai multe imagini de referință pentru a oferi acces mai multor persoane.

Rezultate Obținute

În prima jumătate a execuției din demo, s-au obținut imaginile (primele 5):

Având ca imagine de referință ultima captură din set, programul a returnat valorile False, True, True, True, respectiv True, după care încuietoarea se deschide. Se observă o “shiftare” sau decupare a pozelor (posibil un bug la sincronizare), însă din moment ce se realizează 5 capturi iar “shiftarea” pare uniformă, nu este afectat rezultatul final (cel puțin într-o imagine se găsește fața întreagă). După boot-are, primele 2-3 poze ies întotdeauna cu luminozitate crescută, după care se stabilizează și restul pozelor au un rezultat normal, obținându-se o corectitudine consistentă.
În a doua jumătate a demo-ului, se testează cazul negativ prin capturarea unor imagini ce nu conțin nicio față, și prin urmare încuietoarea nu se deschide.

Concluzii

Utilizarea unei camere pe un Arduino Uno R3 este un bun exemplu de testare a limitelor acestui microcontroller. Atât numărul de pini cât și memoria SRAM au fost utilizate aproape la maxim, aducând dificultăți în implementare pe partea de optimizare și remapare a pinilor. Au existat conflicte în utilizarea pinilor din moment ce shield-ul Ethernet necesită ca pinii 10-13 să fie neutilizați, iar schema originală a camerei se folosea parțial de acești pini. Proiectul și-a atins scopul propus, însă ar putea fi îmbunătățit în câteva puncte în condiția în care se utilizează un microcontroller cu mai multe resurse:

- sistemul de lock are câteva vulnerabilități. Pentru a nu permite accesul unei persoane care deține o poză cu proprietarul, serverul ar putea face autentificarea folosindu-se de mai multe poze în diverse poziții sau de o filmare. De asemenea, semnalul trimis pentru deschiderea incuietorii ar trebui să conțină o măsură de securitate. În momentul de față, dacă un atacator se poate conecta la rețeaua locală și află adresa IP și portul microcontroller-ului, acesta poate trimite orice pachet UDP pentru descuiere
- se pot adauga mai multe funcționalități, precum autentificarea printr-o parolă introdusă la numpad și configurarea dispozitivului folosind tastatura și un display LCD.

Download

Github - FacialRecognitionLocker

Jurnal

- 25 Aprilie - Alegerea temei proiectului și crearea paginii de documentație
- 2-3 Mai - Testarea separată a componentelor exceptând camera
- 5-6 Mai - Acomodarea cu librăriile de lucru și realizarea unui ansamblu de autentificare printr-o parolă introdusă prin numpad
- 10-14 Mai - Procurarea unor materiale adiționale și documentarea pentru folosirea camerei pe Arduino
- 17 Mai - Testarea camerei
- 20-22 Mai - Realizarea lipiturilor pentru conexiuni
- 23-25 Mai - Implementarea programului pentru plăcuța Arduino și a script-urilor de pe server
- 26-28 Mai - Debugging și testare
- 30 Mai - Realizarea documentației

Bibliografie/Resurse

Documentație cameră:
https://www.instructables.com/OV7670-Arduino-Camera-Sensor-Module-Framecapture-T/
https://circuitdigest.com/microcontroller-projects/how-to-use-ov7670-camera-module-with-arduino
https://github.com/ComputerNerd/ov7670-no-ram-arduino-uno
ATmega328P Datasheet
PROGMEM
Arduino Ethernet
Librăria Python face_recognition
Model client-server în Python
Formatul BMP
PDF