

Projekt – Zapisanie danych na karcie SIM			
Imię i nazwisko	Piotr Walczak	Przedmiot	Systemy automatycznej identyfikacji
Nr albumu	151556	Data wykonania	06.06.2026 r.
Kierunek	Informatyka - Internet Przedmiotów - Sem. 3 - Stopień 2 - Studia stacjonarne - magisterskie		

Spis treści

Temat projektu i zadania szczegółowe	2
Realizacja	2
Zadanie 1 – Wybranie pliku zawierającego książkę telefoniczną.....	3
Model Gemini Pro.....	3
Model ChatGpt 5.5	3
Zadanie 2 – Odczytanie zawartości jednego wybranego rekordu pliku zawierającego książkę telefoniczną	4
Model Gemini Pro.....	4
Model ChatGpt 5.5	4
Zadanie 3 – Prezentacja i interpretacja zawartości rekordu	6
Model Gemini Pro.....	6
Model ChatGpt 5.5	7
Zadanie 4 – Utworzenie komendy APDU w odpowiednim formacie w celu zapisania danych na karcie SIM.	8
Model Gemini Pro.....	8
Model ChatGpt 5.5	9
Zadanie 5 – Sprawdzenie poprawności formatu zmodyfikowanego pliku	10
Model Gemini Pro.....	10
Model ChatGpt 5.5	10
Podsumowanie	10
Załączniki	10

Temat projektu i zadania szczegółowe

Tematem projektu jest zapisanie danych na karcie SIM, a konkretnie w pliku zawierającym książkę telefoniczną. Należy to zrobić wykorzystując dwa duże modele językowe. Do realizacji zadań skorzystano z Google Gemini 3.1 Pro z rozszerzonym poziomem myślenia oraz ChatGpt 5.5 w wersji darmowej. Pełne rozmowy z obydwojema modelami znajdują się w załącznikach umieszczonych na końcu raportu.

W ramach projektu określono następujące cele szczegółowe:

1. Wybranie pliku zawierającego książkę telefoniczną.
2. Odczytanie zawartości jednego wybranego rekordu pliku zawierającego książkę telefoniczną.
3. Prezentacja i interpretacja zawartości pliku.
4. Utworzenie komendy APDU w odpowiednim formacie w celu zapisania danych na karcie SIM.
5. Sprawdzenie poprawności formatu zmodyfikowanego pliku.

Realizacja

By zrealizować zadanie napisano następującego prompta do obu modeli językowych:

Muszę zrealizować projekt na systemy automatycznej identyfikacji, który polega na czymś takim:
AI - wykonanie zadania (Zapisanie danych na karcie SIM) (pełne)

Punkty do realizacji: 1. Wybranie pliku zawierającego książkę telefoniczną. Należy tutaj podać informacje z jakiego narzędzia skorzystać, np. EasyReader, podać potrzebne komendy APDU oraz zrozumieć odpowiedź jaką otrzymano od karty.

2. Odczytanie zawartości jednego wybranego rekordu pliku zawierającego książkę telefoniczną. Należy podać wszystkie komendy APDU jakich użyto, żeby wykonać to polecenie. Również należy poznać długość tego rekordu, żeby go odczytać, więc podaj skąd wzięłeś informacje na ten temat.

3. Prezentacja i interpretacja zawartości rekordu. Należy tutaj rozebrać otrzymaną odpowiedź na fragmenty i powiedzieć co symbolizuje każdy z nich.

4. Utworzenie komendy APDU w odpowiednim formacie w celu zapisania danych na karcie SIM. Należy zmodyfikować odczytany rekord tak, by zgadzał się z poprzednim formatem, ale jednocześnie zawierał nowe, wprowadzone przez użytkownika informacje. Oczywiście należy podać wszystkie komendy APDU.

5. Sprawdzenie poprawności formatu zmodyfikowanego pliku. Znowu należy podać komendę APDU potrzebną do sprawdzenia rekordu. Jak to zrobić? Dodatkowo załączam dokument ze standardem dotyczącym kart SIM, z którego możesz korzystać podczas realizacji zadania.

Dodatkowo każdy z modeli otrzymał plik ts_100977v081400p, który zawiera normę ETSI TS 100 977 dotyczącą kart SIM.

Zadanie 1 – Wybranie pliku zawierającego książkę telefoniczną

Model Gemini Pro

Odpowiedź modelu:

1. Wybranie pliku zawierającego książkę telefoniczną

Do przesyłania poleceń do karty poprzez czytnik kart (np. zgodny ze standardem PC/SC) możesz skorzystać z programów edukacyjnych takich jak wspomniany EasyReader, darmowego APDU Scanner lub napisać krótki skrypt w języku Python z użyciem dedykowanej biblioteki `pyscard`.

Plik z zapisaną książką telefoniczną nosi nazwę EF_{ADN} (Abbreviated Dialling Numbers) i znajduje się w katalogu telekomunikacyjnym $DF_{TELECOM}$.

Najpierw musimy użyć polecenia `SELECT`, aby "wejść" do odpowiednich katalogów i na końcu wybrać poszukiwany plik: PDF

- Wybór Katalogu Głównego (MF): `A0 A4 00 00 02 3F 00` PDF + 2
- Wybór Katalogu $DF_{TELECOM}$: `A0 A4 00 00 02 7F 10` PDF + 2
- Wybór Pliku z Książką Telefoniczną (EF_{ADN}): `A0 A4 00 00 02 6F 3A` PDF + 2

Karta powinna w odpowiedzi zwrócić status (SW1 SW2), np. `9F XX`, co oznacza powodzenie operacji oraz że jest dostępne `XX` bajtów dodatkowych informacji do pobrania. Należy wtedy wykorzystać polecenie `GET RESPONSE` (czyli komendę w postaci `A0 C0 00 00 XX`), aby zobaczyć, co nam karta odpowiedziała na temat tego pliku. PDF + 2

Model ChatGpt 5.5

Odpowiedź modelu:

1. Wybór pliku książki telefonicznej

Ścieżka:

```
MF (3F00)
├── DFTELECOM (7F10)
│   └── EFADN (6F3A)
```

APDU:

```
A0 A4 00 00 02 3F 00 ; SELECT MF
A0 A4 00 00 02 7F 10 ; SELECT DFTELECOM
A0 A4 00 00 02 6F 3A ; SELECT EFADN
```

Po `SELECT` karta zwraca:

```
9F XX
```

co oznacza konieczność wykonania:

```
A0 C0 00 00 XX ; GET RESPONSE
```

Oba modele poprawnie wybrały plik, w którym znajdowała się książka telefoniczna. Dodatkowo Gemini Pro podał dużo dodatkowych informacji pomagających rozszerzyć wiedzę o strukturze plików.

Zadanie 2 – Odczytanie zawartości jednego wybranego rekordu pliku zawierającego książkę telefoniczną

Model Gemini Pro

Odpowiedź modelu:

2. Odczytanie zawartości wybranego rekordu książki telefonicznej

Odpowiedź z komendy `GET RESPONSE` na pliku `EFADN` mówi nam, jaką strukturę ma plik. Książka telefoniczna na karcie to plik typu "linear fixed", czyli składa się on z konkretnych "rekordów" (kontaktów) o stałej i niezmienniej długości danych. Skąd wziąć jego długość? Znajduje się ona zawsze w 15. bajcie danych, jakie pobraliśmy wspomnianą komendą `GET RESPONSE`. PDF + 2

Założmy, że odpowiedź karty w 15. bajcie zawierała liczbę `1C` (czyli 28 w systemie dziesiętnym). Nasz rekord zajmuje więc dokładnie 28 bajtów pamięci. PDF + 1

Do odczytu pierwszego z brzegu rekordu w trybie "absolutnym" użyjemy komendy `READ RECORD`: PDF + 2

- o Polecenie APDU: `A0 B2 01 04 1C`
 - o `B2` – Instrukcja `READ RECORD`. PDF + 2
 - o `01` – Numer wskazanego rekordu. PDF + 1
 - o `04` – Tryb odczytu absolutnego wskazanego powyżej rekordu. PDF + 1
 - o `1C` – Spodziewana, wyliczona wcześniej długość rekordu z danymi. PDF + 1

Model ChatGpt 5.5

Odpowiedź modelu:

2. Odczyt rekordu

Długość rekordu odczytujemy z odpowiedzi `GET RESPONSE` dla `EFADN`.

Odczyt pierwszego rekordu:

```
A0 B2 01 04 XX ; READ RECORD
```

gdzie:

- `01` – numer rekordu,
- `04` – tryb absolutny,
- `XX` – długość rekordu.

Dodatkowo poproszono model o podanie informacji skąd wziąć długość rekordu:

Dodaj informację skąd wzięłeś długość rekordu w 2 zadaniu i gdzie dokładnie znajduje się ta długość rekordu

Odpowiedź modelu:

Do punktu 2 warto dodać dokładnie taki fragment:

Długość rekordu została odczytana z odpowiedzi na komendę **GET RESPONSE** wykonaną po wybraniu pliku EFADN (6F3A). Zgodnie ze specyfikacją ETSI TS 11.11, dla plików typu **Linear Fixed** odpowiedź GET RESPONSE zawiera informację o długości pojedynczego rekordu (*Record Length*). Pole to znajduje się w **14. bajcie odpowiedzi FCP (File Control Parameters)** zwracanej przez kartę po komendzie GET RESPONSE. Wartość tego bajtu jest następnie wykorzystywana jako parametr **Le** w komendzie READ RECORD.

Przykład:

```
A0 A4 00 00 02 6F 3A ; SELECT EFADN
→ 9F 0F

A0 C0 00 00 0F ; GET RESPONSE
→ 00 00 00 20 00 00 00 00 00 00 00 00 20 02
```

Interpretacja:

```
Bajt 4 = 20h → rozmiar pliku = 32 bajty
Bajt 14 = 20h → długość rekordu = 32 bajty
Bajt 15 = 02h → liczba rekordów = 2
```

Oba modele poprawnie odczytały zawartość rekordu. Gemini Pro znowu okazał się lepszy, ponieważ nie potrzebował żadnych dodatkowych instrukcji.

Zadanie 3 – Prezentacja i interpretacja zawartości rekordu

Model Gemini Pro

Odpowiedź modelu:

3. Prezentacja i interpretacja zawartości rekordu

Po wydaniu komendy karta odsyła 28 bajtów danych kontaktu oraz kod sukcesu 90 00 (normal ending of the command). Spójrzmy na rozbiecie na elementy zgodnie ze specyfikacją standardu z sekcji opisującej strukturę EF_{ADN} :

Przykładowa odpowiedź: 4B 6F 77 61 6C 73 6B 69 FF FF FF FF FF FF 06 81 21 43 65 87 F9 FF FF FF FF FF FF FF FF 90 00 PDF + 2

- Bajty 1 do 14 (Alpha Identifier):** Wartość 4B 6F 77 61 6C 73 6B 69 FF FF FF FF FF FF . To imię lub nazwa kontaktu zakodowana w kodzie ASCII lub SMS default 7-bit z wyzerowanym ósmym bitem. Szesnastkowe 4B 6F 77 61 6C 73 6B 69 to słowo "Kowalski". Wolne miejsce aż do długości zadeklarowanego formatu musi zostać dopełnione bajtami zapisanymi jako FF . PDF + 4
- Bajt 15 (Length of BCD number):** Wartość 06 . Opisuje długość zajmowaną przez faktyczny numer telefonu z parametrami. Tutaj jeden bajt zostanie zużyty na typ, a pozostałe pięć na numer. PDF + 1
- Bajt 16 (TON and NPI):** Wartość 81 . Definiuje on format numeracji i jej typ (z ang. Type Of Number and Numbering Plan Identifier). Wartość ta identyfikuje w sieci format numeru, a "81" jest domyślną wartością kodującą dany numer m.in jako nieznany (najbardziej uniwersalny w przypadku numerów z prefiksem lub bez prefiksu międzynarodowego). PDF + 2
- Bajty 17 do 26 (Dialling Number):** Wartość 21 43 65 87 F9 FF FF FF FF FF FF . To nasz główny numer telefonu, zakodowany algorytmem odwróconego BCD (Binary Coded Decimal), gdzie nieużywane wartości wypełnia się znakiem F . Jeśli go odwrócimy: 21->12, 43->34, 65->56 itd., uzyskamy numer telefonu "123456789". PDF + 3
- Bajt 27 oraz Bajt 28 (Capability / Extension Record Identifier):** Obie wartości to FF FF . Używa się ich rzadko. Mogłyby posłużyć jako odniesienia do innego pliku (rozszerzenia), na wypadek, gdyby na przykład numer telefonu był tak długi, że nie zmieściłby się w klasycznym buforze przewidzianym w pamięci dla rekordu (max 20 cyfr). Wartość "FF" oznacza nieużyte właściwości. PDF + 4

Model ChatGpt 5.5

Odpowiedź modelu:

3. Interpretacja zawartości rekordu

4A 41 4E FF...FF | 06 | 91 | 84 21 43 65 F7 | FF | FF

- **Alpha Identifier:** 4A 41 4E → ASCII → JAN, pozostałe FF oznaczają niewykorzystane bajty.
- **Length of Number:** 06 → długość numeru telefonu.
- **TON/NPI:** 91 → numer międzynarodowy (ISDN/E.164).
- **Number:** 84 21 43 65 F7 → zapis BCD (cyfry odczytywane są od młodsze półbajtu):
 - 84 → 48
 - 21 → 12
 - 43 → 34
 - 65 → 56
 - F7 → 7 (F = dopełnienie)

Po połączeniu: 481234567.

Dzięki TON/NPI = 91 numer interpretowany jest jako:

+481234567

- **Capability Identifier:** FF → brak dodatkowych parametrów.
- **Extension Record Identifier:** FF → brak rekordu rozszerzającego.

Wynik:

Nazwa: JAN
Numer: +481234567
Typ: międzynarodowy

Model miał duże problemy ze zrozumieniem polecenia i wielokrotnie pomijał kluczowe dla interpretacji informacje. Również bardzo często podawał niepotrzebnie długie i zawiłe informacje, których nie dało się przekleić do raportu.

Nie licząc drobnych zawirowań przy użytkowaniu modelu ChatGpt 5.5, obu modelom udało się poprawnie zinterpretować zawartość odczytanego rekordu.

Zadanie 4 – Utworzenie komendy APDU w odpowiednim formacie w celu zapisania danych na karcie SIM.

Model Gemini Pro

Odpowiedź modelu:

4. Utworzenie komendy APDU do zapisania danych (modyfikacja rekordu)

Wykorzystamy instrukcję `UPDATE RECORD`, aby zmodyfikować odczytany rekord. Chcemy nadać nową tożsamość "Jan" z numerem "987654321". PDF

Musimy sprokurować pakiet zgodny z poprzednim zestawieniem: PDF + 1

- o Imię (Alpha Identifier - 14 bajtów): "Jan" (J: `4A`, a: `61`, n: `6E`). Pozostałe 11 bajtów jest puste, więc uzupełniamy jako: `4A 61 6E FF FF FF FF FF FF FF FF FF FF`. PDF + 2
- o Długość (Length - 1 bajt): `06`. PDF + 1
- o Typ numeru (TON/NPI - 1 bajt): `81`. PDF + 1
- o Numer telefonu (Dialling Number - 10 bajtów): Algorytm odwróconego BCD. "987654321" pod dopełnieniu niewykorzystanego miejsca literką F do uzyskania pełnego bajtu, a następnie obróceniu połówkami to ciąg: `89 67 45 23 F1`. Dalej 5 bajtów dopełnienia `FF`: `89 67 45 23 F1 FF FF FF FF FF`. PDF + 2
- o Opcje i Przedłużenie parametrów (Capability/Extension - 2 bajty): Nie ruszamy, więc wysyłamy: `FF FF`. PDF + 1

Dostajemy nowy sformatowany payload do wrzucenia na kartę o idealnej długości 28 bajtów (1C):

`4A 61 6E FF FF FF FF FF FF FF FF FF FF 06 81 89 67 45 23 F1 FF FF FF FF FF`
`FF`

Komenda zapisująca wpis do pierwszego rekordu przyjmie formę:

`A0 DC 01 04 1C 4A 61 6E FF FF FF FF FF FF FF FF FF FF 06 81 89 67 45 23 F1 FF`
`FF FF FF FF FF FF` PDF + 1

Model ChatGpt 5.5

Odpowiedź modelu:

4. Utworzenie komendy APDU do zapisu danych

Zmiana:

```
JAN → ADAM  
+481234567 → +48500111222
```

Kodowanie:

```
ADAM → 41 44 41 4D
```

```
48500111222  
→ 48 50 01 11 22 2  
→ 84 05 10 11 22 F2 (BCD)
```

Nowy rekord:

```
41 44 41 4D FF FF FF FF FF FF FF FF FF FF FF FF  
06 91  
84 05 10 11 22 F2 FF FF FF FF  
FF FF
```

Komenda zapisu:

```
A0 DC 01 04 20  
41 44 41 4D FF FF FF FF FF FF FF FF FF FF FF FF  
06 91  
84 05 10 11 22 F2 FF FF FF FF  
FF FF
```

Model znowu potrzebował dodatkowych instrukcji, żeby podać wszystkie niezbędne elementy do wykonania tego zadania:

Pokaż w jakiś sposób przekształcasz imię ADAM na system hex i nowy numer telefonu na system hex. Zawrzyj to w odpowiedzi

Ostatecznie oba modele poprawnie zmodyfikowały rekord w książce telefonicznej.

Zadanie 5 – Sprawdzenie poprawności formatu zmodyfikowanego pliku

Model Gemini Pro

Odpowiedź modelu:

5. Sprawdzenie poprawności formatu modyfikowanego pliku

Jeśli po wykonaniu komendy zapisu karta odeśle flagę statusu `90 00`, oznacza to "normal ending of the command", czyli karta przyjęła dane operację i zatwierdziła ją w pamięci EEPROM. Oprócz śledzenia statusu odpowiedzi z poprzedniej komendy, najbezpieczniej sprawdzić pożądany rezultat nakazując programowi wysłanie ponownie zredagowanej na samym początku wiadomości `READ RECORD : PDF + 4`

APDU do odczytu wpisu: `A0 B2 01 04 1C PDF +1`

Poprawne, wyzerowane zachowanie karty powinno zaowocować zwrotem wcześniej zdefiniowanego i wysłanego payloadu (28 bajtów hex), na którego końcu pojawi się informacja z warunkiem statusowym "SUCCES":

Odpowiedź: `4A 61 6E FF FF FF FF FF FF FF FF FF FF FF FF 06 81 89 67 45 23 F1 FF FF FF FF FF FF FF FF 90 00 PDF +1`

Model ChatGpt 5.5

Odpowiedź modelu:

5. Sprawdzenie poprawności zapisu

Odczyt zmodyfikowanego rekordu:

```
A0 B2 01 04 20
```

Oczekiwana odpowiedź:

```
41 44 41 4D ... 06 91 84 05 10 11 22 F2 ... FF FF
90 00
```

Weryfikacja:

- `41 44 41 4D` → ADAM
- `84 05 10 11 22 F2` → +48500111222
- `90 00` → zapis wykonany poprawnie

Wynik: rekord zawiera dane **ADAM**, **+48500111222**, a jego format jest zgodny ze strukturą EFADN.

Oba modele poprawnie odczytały i zinterpretowały wynik.

Podsumowanie

Oba modele poradziły sobie ze wszystkimi powierzonymi im zadaniami, jednak model Gemini Pro poradził sobie znacznie lepiej niż Chat Gpt 5.5. Nie potrzebował on dodatkowych poleceń, czy wyjaśniania szczegółów. Za każdym razem poprawnie zinterpretował zadane mu pytania.

Załączniki

1. Rozmowa z Google Gemini Pro: <https://gemini.google.com/share/3a43ffa13732>
2. Rozmowa z ChatGpt 5.5: <https://chatgpt.com/share/6a23fe6a-2214-83ed-8916-5cff8a27aab6>