

# Plug & Play ChatGPT

**Student:** Nicolae Andrei

**Grupa:** 334CC

## Introducere

Obiectivul principal al proiectului consta in integrarea unui API al unui AI precum ChatGPT in workflow-ul unui developer, intr-o maniera plug & play.

Concret, proiectul consta intr-un dispozitiv pe care il poti conecta la PC, care poate asculta si decoda pachetele USB trimise de o tastatura, se conecteaza la internet si poate face un API request pentru ChatGPT, iar raspunsul primit il tasteaza la PC, ca si cum ar fi tastat de o tastatura normala.

De asemenea, el poate functiona si in mod standalone, conectand doar tastatura, fiind alimentat de un acumulator, iar procesul este acelasi, insa raspunsul primit il afiseaza pe un display TFT.

Ideea mi s-a parut utila tocmai din perspectiva plug & play, in sensul ca nu ai avea nevoie de un program specializat pe computer pentru a interactiona cu API-ul, ci ai un mic black box fizic care pur si simplu functioneaza.

## Descriere generala

### Schema bloc - mod middleman



### Schema bloc - mod standalone



Functionalitatea de baza a proiectului a fost de a avea un dispozitiv plug & play, care poate fi folosit pentru a accesa functionalitatile unui API precum ChatGPT, pe orice PC si in orice aplicatie de pe acesta - de exemplu, pentru a face refactoring/documentare pe o bucata de cod, pentru a genera cod boilerplate, pentru a corecta/reformula o sectiune dintr-un text, pentru a scrie un template de e-mail, traduceri de text, sau pur si simplu pentru a afla o informatie noua, intr-un mod mai rapid decat daca am deschide browser-ul. Acest lucru se face prin apasarea unui shortcut predefinit, tastarea interogarii care sa fie trimisa la API si apoi tastarea raspunsului de catre microcontroler. Atunci cand nu se asculta interogarea, controlerul pur si simplu redirectioneaza pachetele USB catre PC. In modul simplu, microcontrolerul poate primi doar date de la tastatura, insa putem extinde functionalitatile daca rulam un script batch pe PC, care ne lasa sa trimitem un text selectat pe calculator catre microcontroler, prin monitorul serial. Folosind display-ului TFT, pentru a selecta diferite prompt-uri,

care sa fie inserate inaintea textului scris de la tastatura, pentru a comenzi folosite des, spre exemplu: "Translate this text to english:", "Refactor this code, don't write any other comments: ", "Correct any mistakes you find in this text:".

Adaugarea display-ului a dus la crearea modului standalone: dispozitivul sa poata fi folosit si in cazul in care nu exista acces la un calculator, iar din cateva teste, am observat ca acesta are o autonomie de ~3 ore.

## Hardware Design



In cadrul proiectului am folosit urmatoarele componente:

- Raspberry pi pico: care ruleaza un program care foloseste biblioteca TinyUSB pentru a receptiona pachetele USB de la tastatura
- Raspberry pi pico w: care ruleaza un program scris in CircuitPython, decodeaza pachetele primite de la primul Pico prin UART, face API requests, afiseaza raspunsul pe display sau il tasteaza la PC
- Display TFT 2.0": utilizat pentru afisarea raspunsurilor sau a diferitelor comenzi dintr-un meniu.
- Encoder rotativ cu buton OKY3431-4: folosit pentru selectarea optiunilor din meniu.
- Modul incarcare acumulator OKY3404: incarca acumulatorul atunci cand dispozitivul este conectat la PC, sau alimenteaza circuitul, folosind acumulatorul, in mod standalone.
- Acumulator 18650 Rakieta: cu capacitatea de [12000mAh](#) (hehe, cum sa nu), 3.7V
- Modul port USB: prin care se poate conecta o tastatura la dispozitiv
- Dioda Schottky: plasata dupa modulul de incarcare, pentru a nu lasa tensiunea de 5V de pe VSYS sa ajunga la pinii bateriei si sa distruga portul USB de pe PC. - inlocuita cu un P-MOSFET (asa cum este recomandat in sectiunea 3.6 din [datasheet-ul de la pico](#), pentru a avea un voltage drop mai mic de la VBAT la VSYS)

## Software Design

Prima problema care trebuie rezolvata este citirea pachetelor USB care sunt generate atunci cand se apasa pe taste. Acest lucru nu este exact trivial, intrucat exista diferite conventii folosite de diferite tastaturi pentru comunicarea prin USB. Exista insa biblioteca TinyUSB [10], si implementarea sa pentru microcontrolerul RP2040 [11], care abstractizeaza din complexitatea protocolului. Astfel, se poate folosi capabilitatea de PIO (programmable I/O) a microcontrolerului pentru a adauga si citi interfete USB suplimentare. Implementarea aceasta este scrisa in C, insa majoritatea bibliotecilor destinate microcontrolerului sunt scrise folosind limbajul CircuitPython, asa ca am modificat exemplul `capture_hid_report.c`, pentru a trimite rapoartele USB printr-o conexiune UART catre un Raspberry pi pico w, programat folosind CircuitPython.

Urmatorul pas este parsarea acestor pachete. Formatul pachetelor depinde de asemenea de tipul tastaturii si a functionalitatilor sale (5-key rollover vs n-key rollover, diferite taste cu roluri macro/mouse etc.). Implementarea din proiect ia in calcul cazul cel mai simplu: 1 byte care reprezinta tastele modifier (Shift, Alt, Win, Ctrl), si un range de bytes care reprezinta tastele apasate (care sunt decodati folosind [acest tabel](#)). Acesti parametrii sunt specificati in cadrul functiei care parseaza

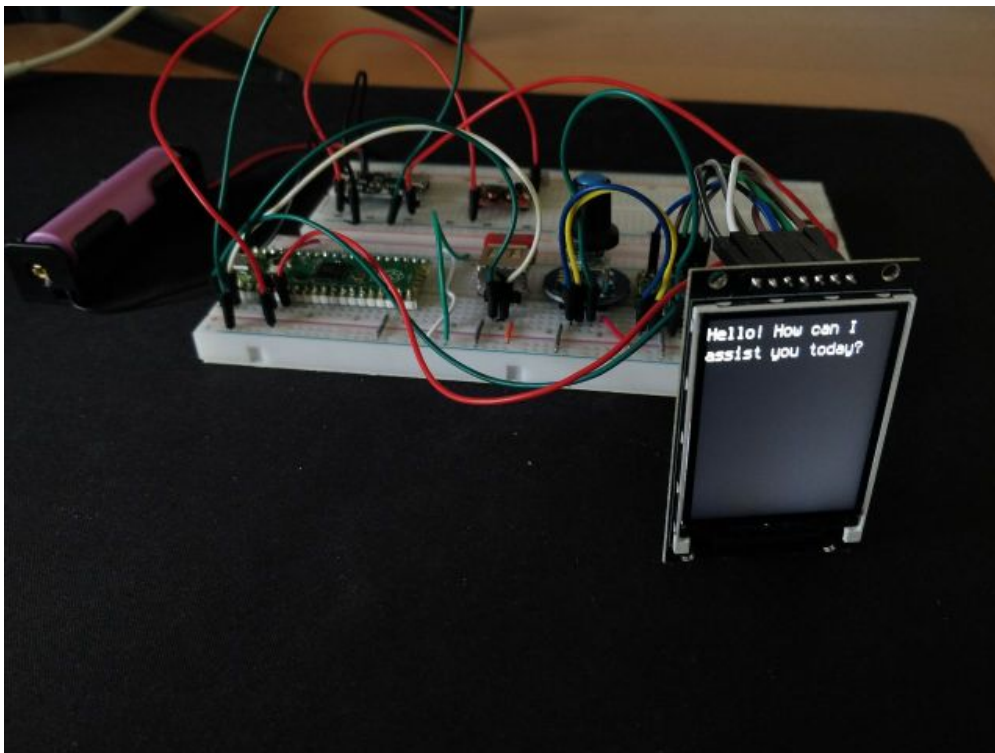
pachetele.

Acum ca putem citi un text de la tastatura, putem face un API request si receptiona raspunsul. Pentru a putea tasta raspunsul la un PC, exista biblioteca `adafruit_hid`, prin care putem face microcontrolerul sa se comporte ca o tastatura. Raspunsul este tastat pe masura ce este primit de catre microcontroler.

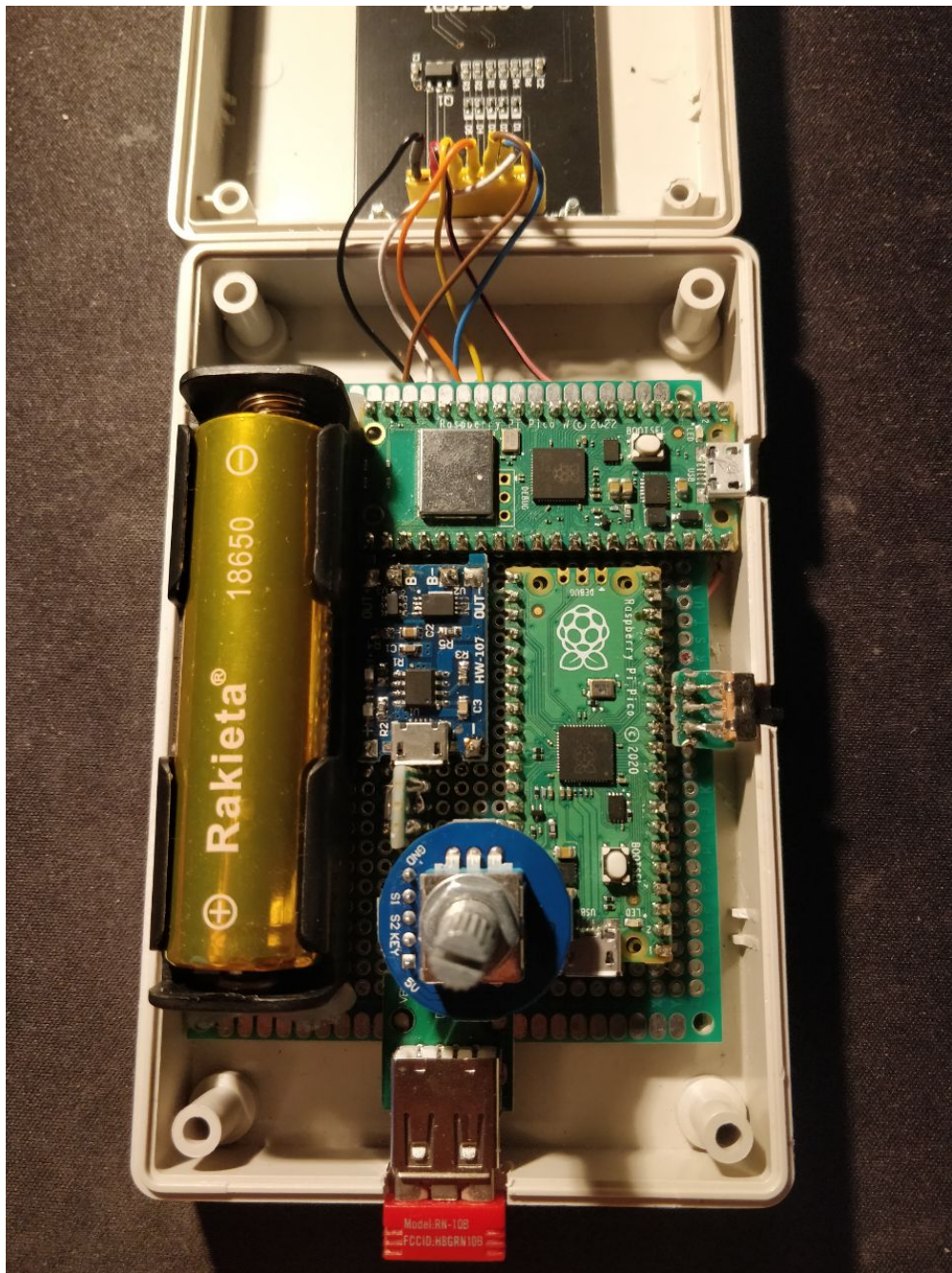
Dorim de asemenea sa afisam textul receptionat pe un display TFT, pentru acesta existand bibliotecile `displayio` (pentru interfatarea prin SPI), `adafruit_ili9341` (pentru cateva setari specifice display-ului ales), `terminalio` (pentru a afisa text).

In final, pentru a interfata cu display-ul, pentru a selecta diferite template-uri (despre care voi vorbi in sectiunea functionalitati), vom folosi un encoder rotativ cu buton, care este controlat prin biblioteca `rotaryio`.

## Rezultate Obtinute







DEMO: <https://youtu.be/ocueh1Meev0>

## Concluzii

Initial, am plecat cu motivatia de a vedea daca un astfel de proiect este posibil, intrucat, desi partea hardware pare destul de simpla, partea software este relativ complexa si sunt multe puncte in care m-am blocat pe parcursul dezvoltarii, iar in final a fost foarte satisfactor sa vad ca proiectul chiar functioneaza.

## Download

<https://github.com/nan-dre/plugin-play-chatgpt>

<https://github.com/nan-dre/plugin-play-chatGPT-docs>

## Jurnal

5.05.2023 – pagina ocw, documentatie initiala

10.05.2023 - prototip initial pe breadboard + functionalitati de baza

18.05.2023 - finalizare implementare hardware

26.05.2023 - finalizare implementare software

## Bibliografie/Resurse

<https://datasheets.raspberrypi.com/picow/pico-w-datasheet.pdf>

<https://github.com/hathach/tinyusb>

<https://github.com/sekigon-gonnoc/Pico-PIO-USB>

<https://github.com/hathach/tinyusb/blob/5e023fa2ca4c20f06b6e0dc12f6f044a7d4e14bd/src/class/hid/hid.h#LL1016C19-L1016C19>

<https://www.youtube.com/watch?v=wdgULBpRoXk>

<https://stackoverflow.com/questions/74711107/openai-api-continuing-conversation>

<https://blog.smittytone.net/2022/02/16/pico-usb-serial-communications-with-circuitpython/>

Export to PDF

From:

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

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2023/ncaroi/plug>



Last update: **2023/05/28 20:48**