

Jak zaprogramować procesor Waleta z pomocą przeglądarki Chrome

Do wgrywania plików binarnych do procesorów ESP firma Expressive przygotowała dla systemu Windows narzędzia dostępne pod adresami:

https://docs.espressif.com/projects/esp-test-tools/en/latest/esp32/production_stage/tools/flash_download_tool.html lub

https://docs.espressif.com/projects/esp-at/en/latest/esp32/Get_Started/

Stamtąd też można ściągnąć pakiet oprogramowania jako plik .zip

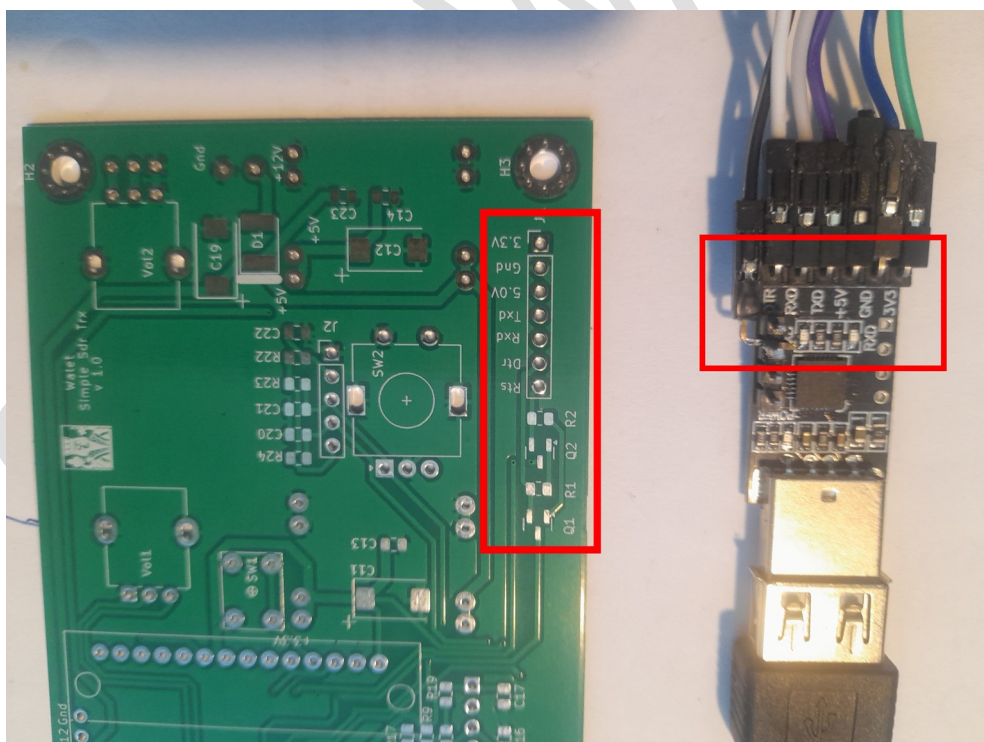
https://dl.espressif.com/public/flash_download_tool.zip, w którym znajduje się plik wykonywalny .exe działający jako okno IDE pozwalający na wgranie plików binarnych do procesora.

Niezależnie od tego, niemal identyczną rolę pełni dostępna w sieci aplikacja zbliżona w wyglądzie do narzędzia instalowanego na platformie Windows z tym, że działa ona **wyłącznie w przeglądarce Chrome**.

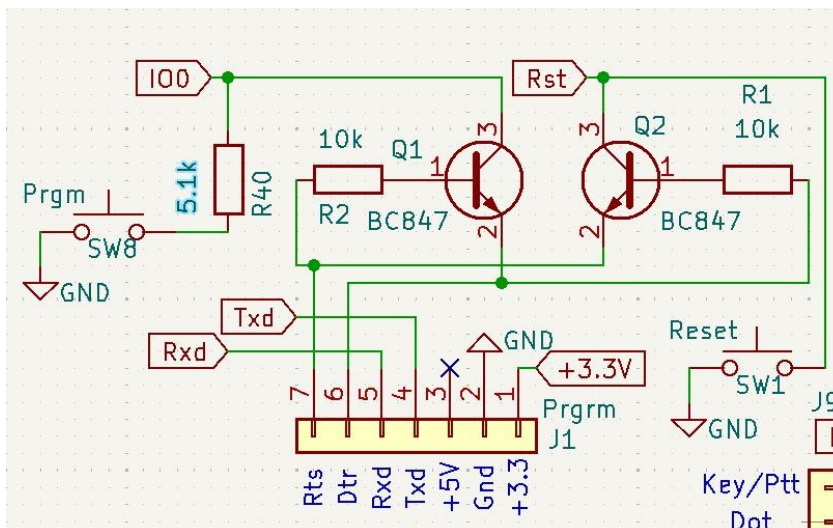
1. Przeglądarka Chrome ma komponenty pozwalające na bezpośrednie programowanie układu ESP32 A1S bez konieczności instalacji Arduino, bibliotek i innych zależności.

Odpowiednie narzędzie znajduje się na stronie : <https://espressif.github.io/esptool-js/>

2. Aby wykorzystać to narzędzie należy przede wszystkim zapewnić kontakt elektryczny procesora na płytce z komputerem PC za pomocą opisanego wcześniej (np. w wykazie elementów płytki h0) konwertera USB/RS232 z pełną sygnalizacją (RTS oraz DTR).



Fotografia pokazuje gdzie na płytce znajdują się obwody odpowiedzialne za komunikację oraz konwerter z pełną sygnalizacją RS. Do programowania procesora **nie jest potrzebne** zewnętrzne zasilanie bo wystarcza to z portu USB PC.

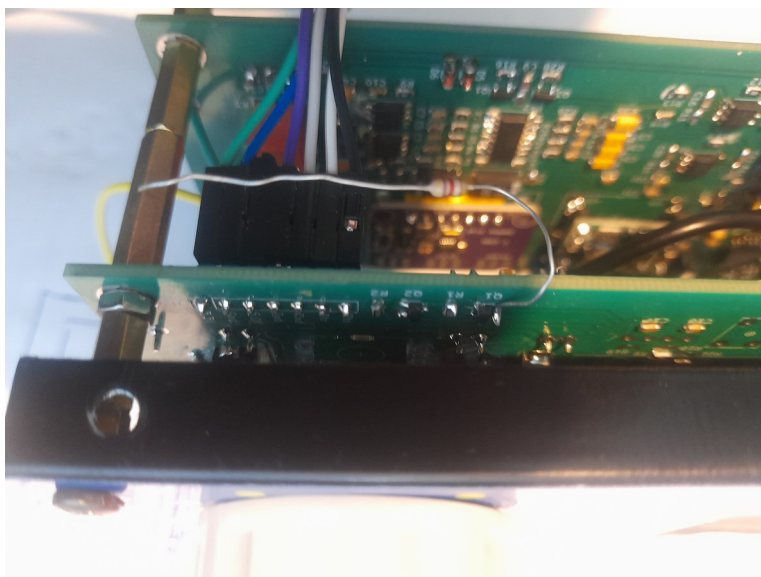


Niestety, okazało się, że obecne rozwiązanie elektryczne nie uwzględniało tej formy ładowania programu i musiałem zmodyfikować tą część układu przez dołożenie przycisku wprowadzania procesora w tryb programowania (SW8 oraz R40). Te elementy pojawią się w następnej wersji płytek o ile będzie taka potrzeba.

Platforma Arduino radzi sobie lepiej z ładowaniem programu i nie wymaga ręcznego przełączenia procesora w tryb programowania

dlatego to rozwiązanie pojawia się tak późno i obecne płytki nie zawierają dodatkowych elementów.

Ale na razie, dla istniejących płytek, można zastosować rozwiązanie tymczasowe polegające na



tymczasowym dolutowaniu do kolektora Q1 opornika 5.1k oraz zwarcie go do masy w momencie startu programowania o czym będzie dalej. Zdjęcie pokazuje jak to zrobiłem na swojej płytce. Wartość opornika R40 nie jest krytyczna (1-10k) a tu widać opornik 4.7k dolutowany do kolektora Q1, który zwieram do metalowej tulejki połączonej z masą w momencie startu programowania.

3. Kolejne wersje binarne oprogramowania, obok kodu źródłowego Arduino, będą się pojawiać na GitHub w odpowiednim katalogu z numerem wersji a do zaprogramowania układu potrzebne są 4 pliki binarne i na przykład obecnie mają one nazwę:

boot_app0.bin

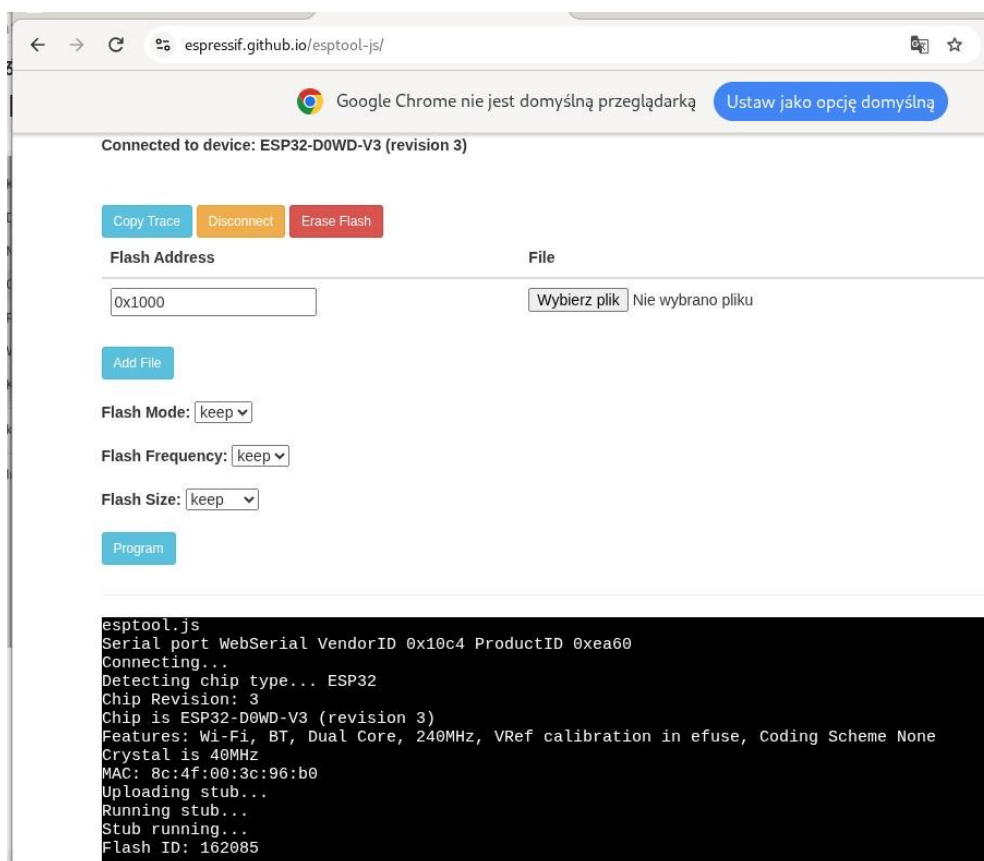
ssd_trx01.ino.bin

ssd_trx01.ini.bootloader.bin

ssd_trx01.ino.partitions.bin

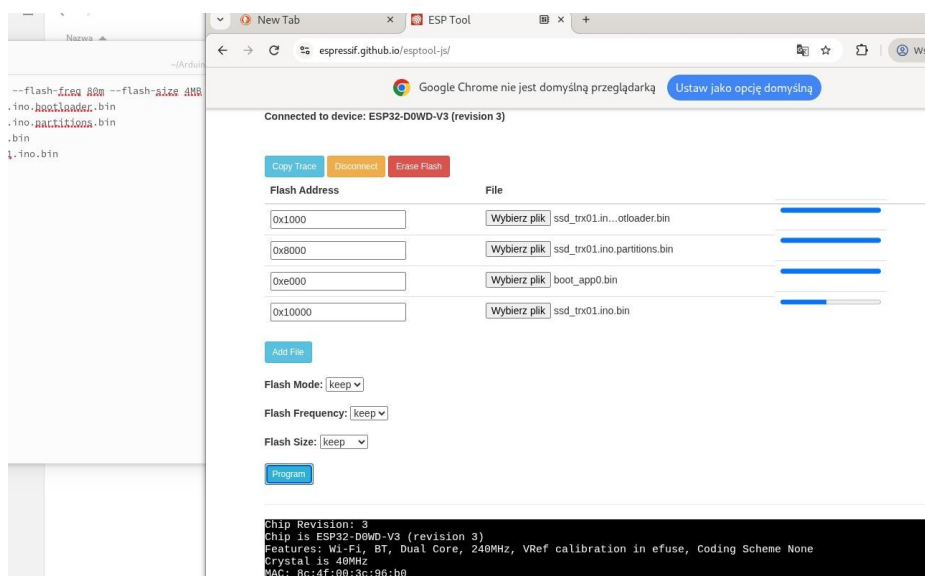
Dodatkowo, podświetliłem plik **flash_args**, który jest bardzo pomocny podczas definiowania treści wgranej do procesora. Kolejne wersje oprogramowania zawierać będą zestaw pokazanych pięciu plików ale być może będą się różnić rdzeniem nazwy, który tu ma postać **ssd_trx01**

Pojawi się okno z możliwością wyboru portu, w moim przypadku to ostatni na dole port **ttyUSB1** identyfikowany dodatkowo przez symbol chip-u **CP2102**. Po wyborze (kliknięciu) na właściwy port aktywny stanie się klawisz **Połącz (Connect)** i powinien pojawić się komunikat w oknie konsoli na dole aplikacji o skutecznym połączeniu.



Od tego momentu można zwolnić przycisk programowania i przejść do wyboru plików oraz uruchomienia procedury download.

5. Właściwe programowanie wymaga podania dla każdego z czterech plików wymienionych w p.3 zarówno nazwy jak i adresu gdzie powinien ten plik się zaczynać. Robi się to zamieszczając w polu adresowym (z lewej) odpowiednią wartość skojarzoną ze ścieżką do pliku i tu jest pomocny widoczny w tle plik `flash_args` zawierając właśnie potrzebne dane. Kolejne pliki dodaje się klawiszem **Add File**.



Zdjęcie pokazuje stan aplikacji po wybraniu czterech niezbędnych plików z odpowiednimi adresami po uruchomieniu programowania za pomocą klawisza **Program**. Przesłanie plików zajmuje nie więcej niż kilkanaście sekund a po zakończeniu należy układ zresetować klawiszem RST lub poprzez zasilanie i procesor powinien już działać pod kontrolą nowego oprogramowania.

Przy sprawnym połączeniu płytki procesora z komputerem PC wgranie oprogramowania zajmuje kilka minut.