

Laboratorium nr 5-6

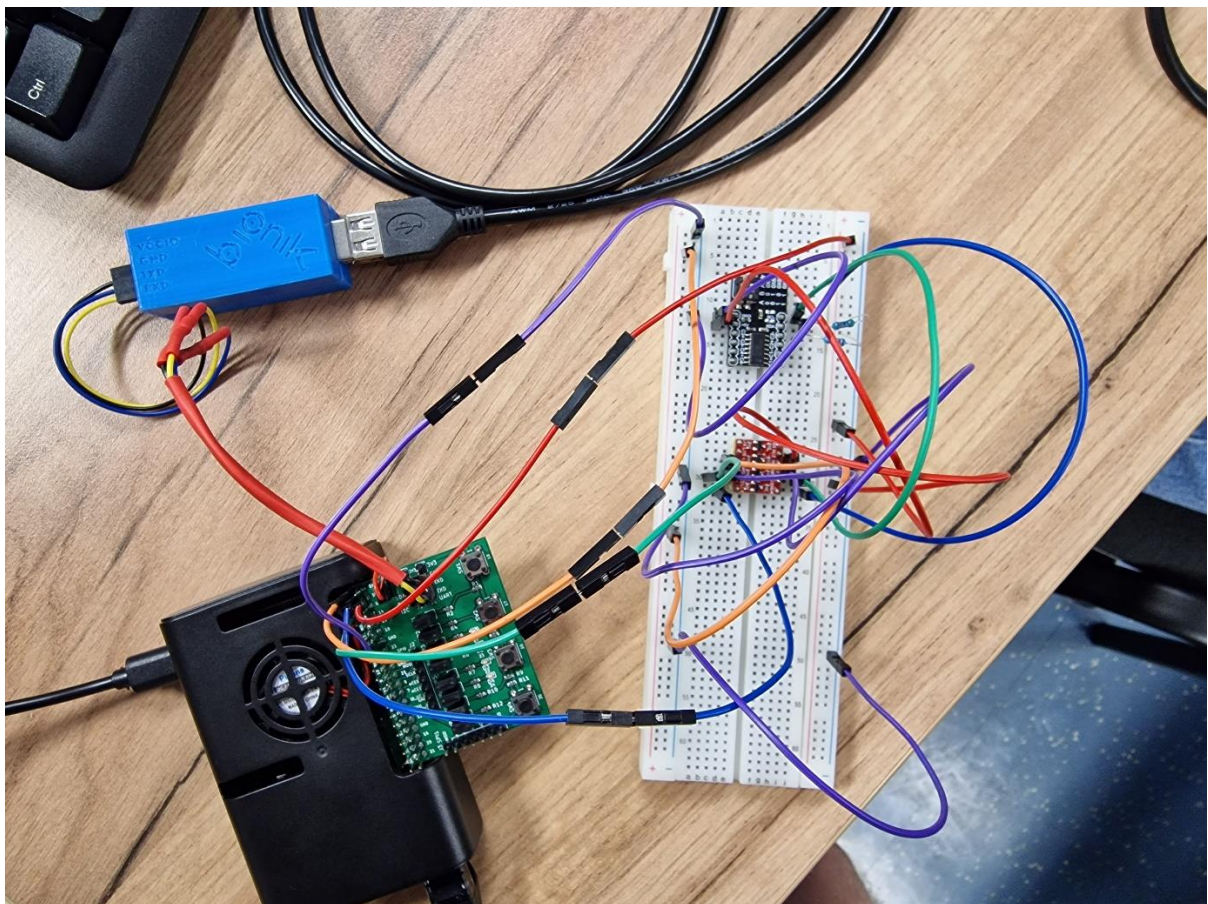
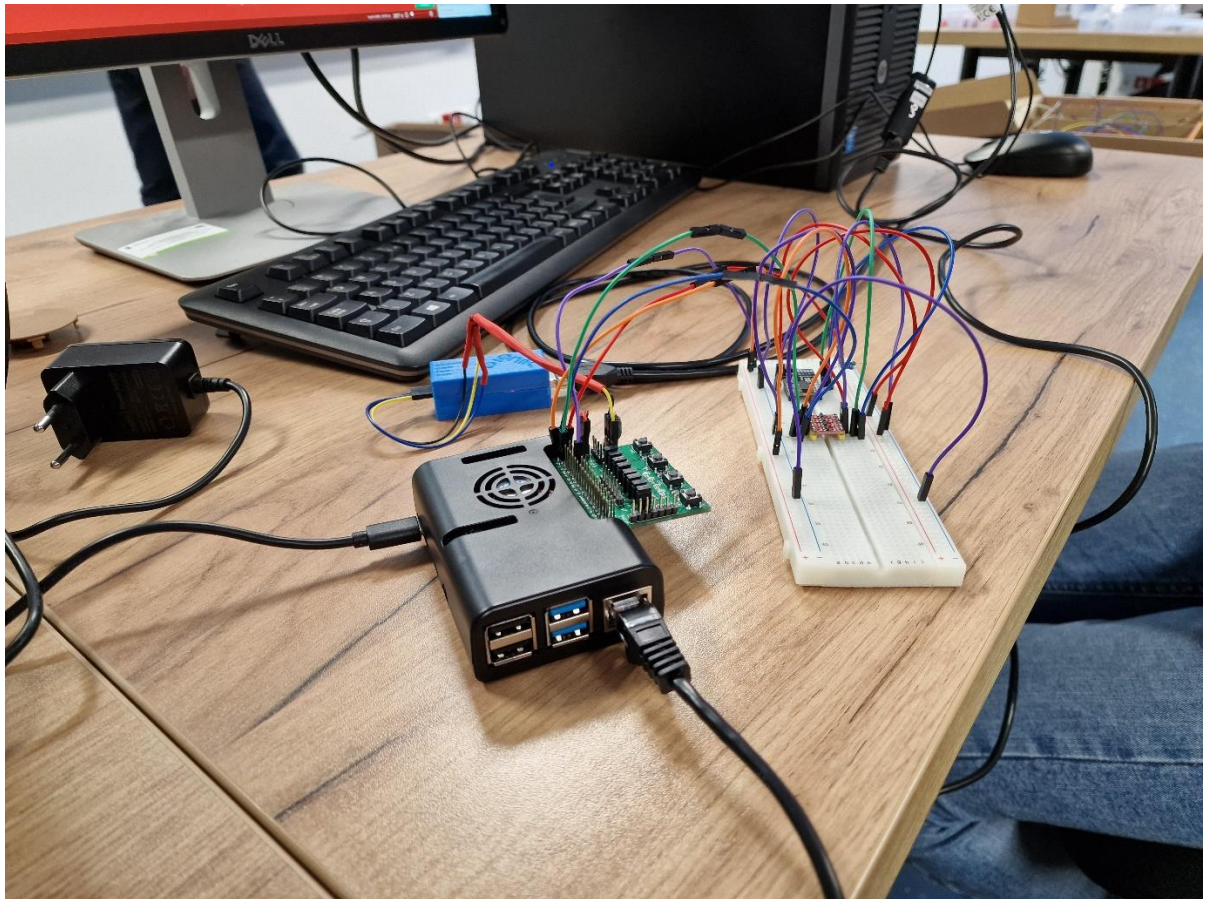
SKPS – mini projekt

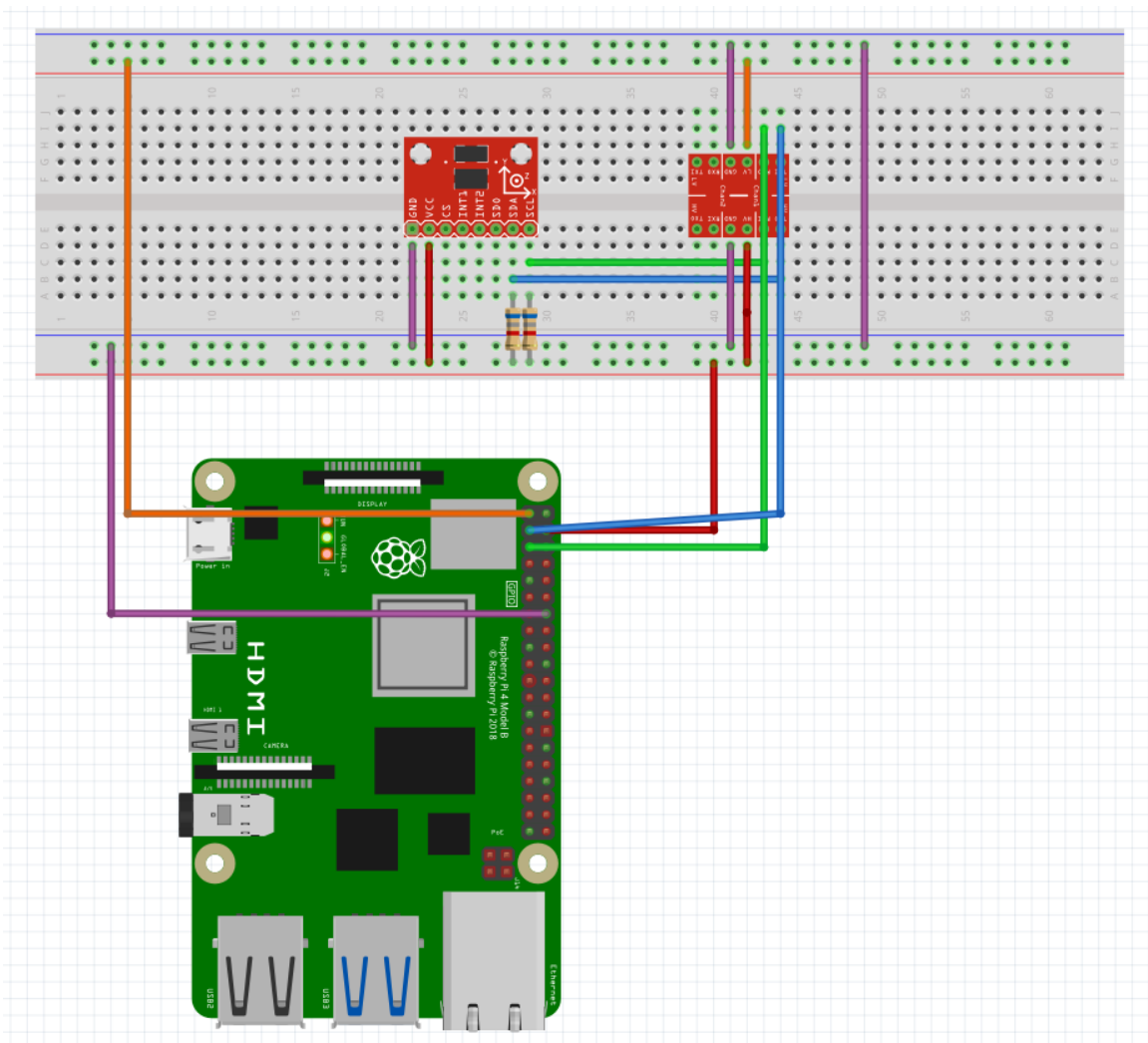
1. Temat projektu

Na podstawie podanych w skrypcie laboratoryjnych projektów wybraliśmy projekt nr 1, czyli zdalny oscyloskop 4-kanałowy. Naszym zadaniem jest przygotowanie schematu działania urządzenia, podłączenia go w prawidłowy sposób do naszego urządzenia oraz przygotowania oprogramowania obsługującego urządzenie i prezentującego wyniki. Efektem końcowym naszej pracy powinien wyświetlany na komputerze PC wykres zmiany napięcia mierzonego przez oscyloskop 4-kanałowy. RPi-4 będzie zbierać informację z wyjścia, a następnie przekazywać na hosta, na którym będzie odbywać się obróbka danych. Kończącym efektem będzie prezentowanie wykresu zmiany wysokości napięcia.

2. Przygotowanie do projektu

Pracę rozpoczęliśmy od przeczytania dokumentacji dla obsługi oscyloskopu, a dokładniej modelu MCP3424. Poznaliśmy sposób jego działania oraz przykładowe użycie. Następnie przystąpiliśmy do stworzenia schematu podłączenia MCP3424 do płytki RPi-4. Znajduje się on na naszym repozytorium w folderze cw5/schemat_podlaczenie/schemat.fzz. Po zaakceptowaniu naszej pracy przez prowadzącego przystąpiliśmy do pierwszego podłączenia fizycznego. Podłączenie wyglądało w taki sposób:





Oryginalne zdjęcia można znaleźć również na naszym repozytorium: https://gitlab-stud.elka.pw.edu.pl/ksulkows/skps21z_kamilsulkowski_karolkasperek/-/tree/main/cw5/schemat_podlaczenie .

Prowadzący zaakceptował fizyczną realizację i mogliśmy sprawdzić, czy rzeczywiście urządzenie jest podłączone poprawnie.

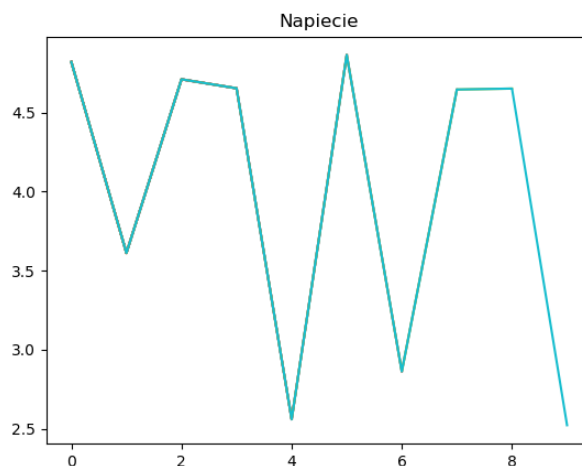
3. Obsługa urządzeń wejścia-wyjścia

Przystąpiliśmy do uruchomienia RPi4 z systemem OpenWRT. Po poprawnym uruchomieniu przystąpiliśmy do sprawdzenia, czy nasze podłączone urządzenie jest wykrywane. Na początku nie było ono aktywne, jednak po doinstalowaniu odpowiednich pakietów i rozszerzeń: `i2c-tools`, `kmod-i2c-bcm2835`, `kmod-i2c-core`, `libi2c` oraz dodaniu linijek do pliku `/boot/config.txt`: `dtoverlay=i2c_arm=0n` oraz `dtoverlay=i2c1=on` mogliśmy zobaczyć, że nasze urządzenie było widoczne.


```
root@OpenWrt:/# i2cdetect -y 1
    0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
10:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
20:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
30:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
40:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
50:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
60:  --  --  --  --  --  --  --  68  --  --  --  --  --  --  --
70:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
```

4. Przygotowanie prezentacji wyników na komputerze PC

Na początku stworzyliśmy przykładowy generator wartości napięcia, który miał symulować działanie naszego urządzenia. Wyniki były zapisywane do pliku. Następnie stworzyliśmy kolejny program, mający na celu odczytywanie z pliku generowanych wartości napięcia i tworzenie wykresu prezentującego dane. Wszystkie programy tworzone do generowania i obróbki danych są pisane w języku Python. Następnym krokiem będzie już odczytywanie danych z naszego urządzenia.



5. Tworzenie pakietu na OpenWRT.

Przystąpiliśmy do tworzenia oprogramowania mającego na celu odczytywanie informacji z wejścia i przesyłanie ich na komputer PC. Jednak po długich i usilnych próbach nie udało się nam dojść do satysfakcjonującej wersji oprogramowania. Uznaliśmy, że nie zdążymy wykonać zadania w wyznaczonym czasie, dlatego zakończyliśmy naszą pracę na wyżej przedstawionych efektach.