

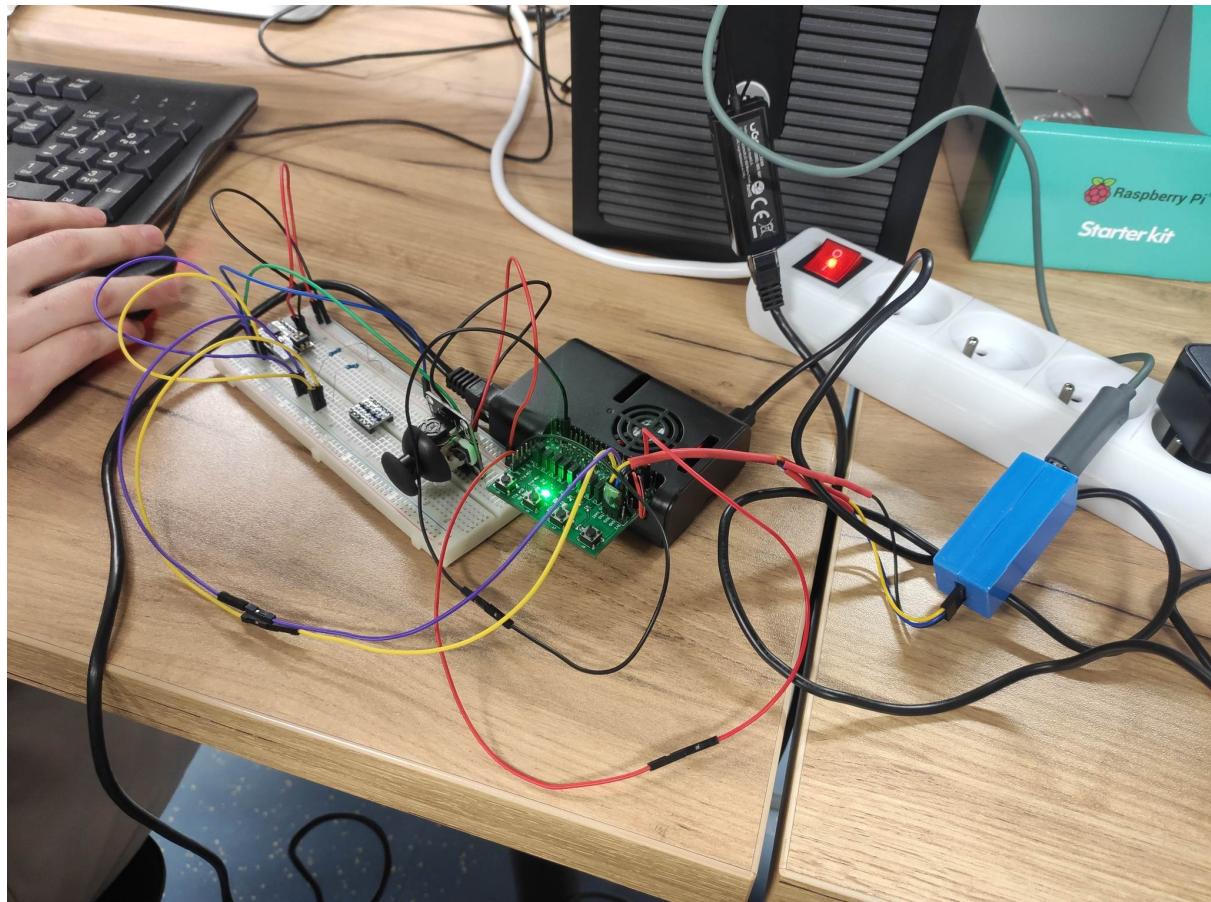
Hubert Gołębiowski, 313804
Jakub Rozkosz, 313588

SKPS - lab 5 oraz 6

Temat projektu:

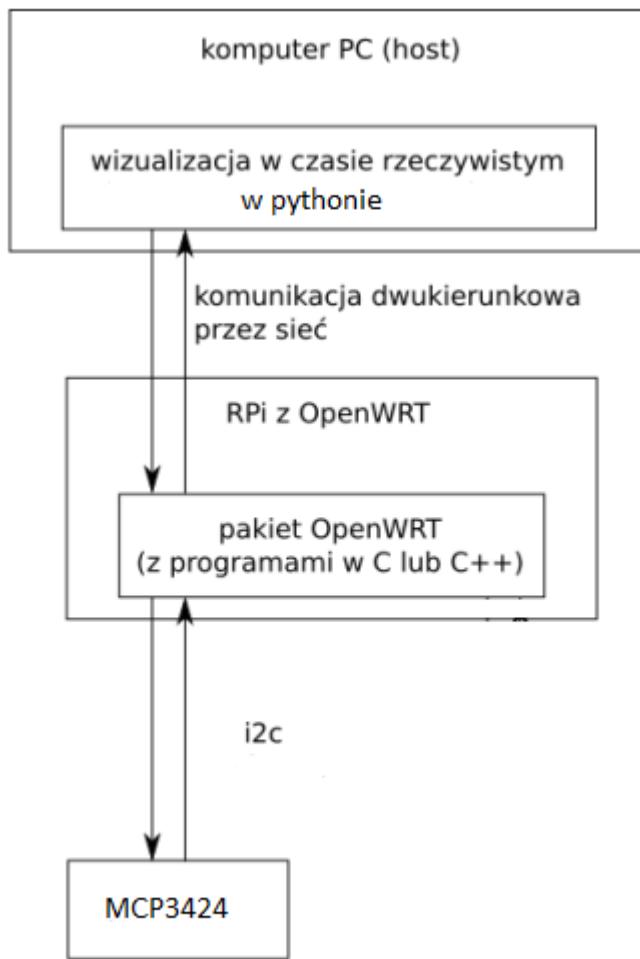
Zdalny oscyloskop 4-kanałowy.

Zbudowany układ:



Układ zawiera: płytę RPi, oscyloskop, oraz joystick.

Schemat architektury:



Działanie programu na podstawie plików:

Plik i2c.cpp na RPi:

Zawiera klasę I2CConnector, która obsługuje połączenie z przetwornikiem za pomocą magistrali I2C. Metoda init() inicjuje połączenie, a metody readData() i setConfig() służą odpowiednio do odczytania wartości napięcia oraz do wpisania własnych ustawień do rejestru konfiguracyjnego przetwornika. Ostatecznie nie skorzystaliśmy z opcji modyfikacji ustawień, gdyż domyślne opcje były zadowalające. W klasie jest jeszcze jedna metoda convertData(), która jest metodą pomocniczą wykorzystywaną przez readData(). Służy ona do konwersji czytanych bajtów na typ signed integer.

Plik tcp_connector.cpp na RPi:

Program, który odpowiada za nawiązanie komunikacji TCP. Metoda set_tcp_connection() inicjalizuje socket'a, łączy go z adresem IP localhost oraz portem ustalonym przy inicjalizacji obiektu (tutaj 8080), następnie nasłuchiwa czy nie przyjdzie próba połączenia się z hostem. Jeżeli takie przyjdzie to zostaje nawiązane połączenie. Korzystając z metody send_data(float data) wysyłane są dane na ip, z którego przyszła próba połączenia. Metoda close_connection() zamyka połączenie.

Program adc.cpp na RPi:

Główny program aplikacji. Co 0.3 sekundy program odczytuje wartość napięcia z przetwornika i przesyła rezultat gniazdem TCP na komputer hosta. Aby uzyskać faktyczne napięcie, liczba odczytana z przetwornika musi zostać pomnożona przez wartość najmniej znaczącego bitu LSB oraz podzielona przez wzmacnienie PGA, które dla naszych ustawień wynosiły odpowiednio: $LSB = 0.001V$ $PGA = 1$, dlatego ostatecznie mnożymy odczytaną wartość przez 0.001.

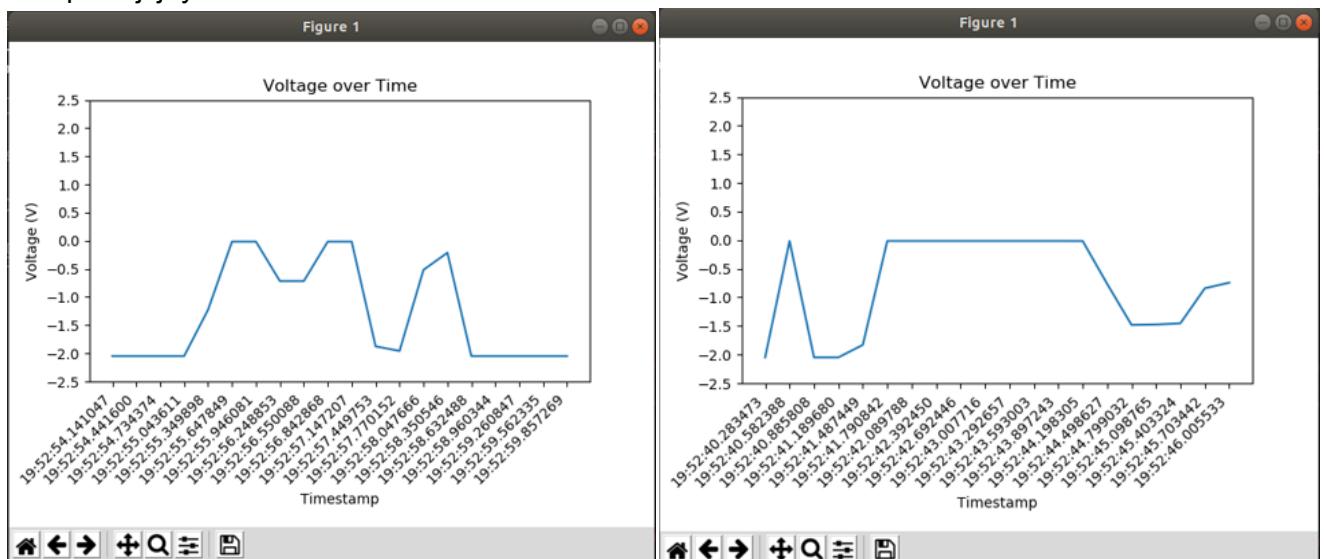
Program klient.py na hoście:

Program poprzez socket nawiązuje połączenie z RPi. Aktywnie nasłuchuje na porcie 8080. Kiedy odbierze wartość przeslaną przez RPi tj. zmierzona przez przetwornik wartość napięcia, aktualizuje dane wizualizowane na wykresie napięcia od czasu.

Rezultaty:

Podczas pracy programu, odchylając joystick w lewo zostaje zmienione napięcie (im bardziej odchylone tym większa wartość napięcia), co jest na bieżąco prezentowane na wykresie.

Na poniższych dwóch zdjęciach znajdują się przykłady prezentowanych wykresów podczas manipulacji joystickiem.



W specyfikacji oscyloskopu posiadanego na laboratorium można wyczytać, iż odczytuje on wartości od -2.048V do 2.048V i pod takie wartości pierwotnie została wyskalowana os y na wykresie. Ostatecznie udało nam się jedynie uzyskać wartości od -2.048V do 0V (dla oryginalnej pozycji joysticka było to -2.048V, a przy wychylaniu w lewo wartości rosły aż do 0V). Prawdopodobnie przyczyną tego było złe podłączenie joysticka do przetwornika (Ch1- był podłączony do osi x joysticka, a Ch1+ do uziemienia).

Wynik działania programu w terminalu: