

## Wstępny opis wybranego rozwiązania

Elementem pomiarowym w naszym układzie będą potencjometry liniowe LCF 100 5k. Do przetwarzania danych pomiarowych oraz do mierzenia zmiany napięcia użyjemy raspberry pi pico. Odległość między walcami będzie wyświetlana na dwóch wyświetlaczach TM1637 6-cyfrowych, odpowiednio dla prawej i lewej strony. Dodatkowym wyświetlaczem będzie LCD 2.4" ILI9341. Wszystko zostanie umieszczone w obudowie wykonanej w technologii druku 3D.



1 LCF 100 5k



2 Raspberry pi pico



3 Wyświetlacz TM1637



4 LCD 2.4" ILI9341

## Zasilenie części logicznej i potencjometrów

Stanęliśmy przed problemem jak zasilić nasz układ, który składa się z Raspberry Pi Pico wymagającym zasilania 5 lub 3,3V i potencjometru, który możemy zasilić napięciem do 42V. Postanowiliśmy pójść na kompromis i kupić zasilacz 12V, który będzie bezpośrednio zasiliał potencjometry co pozwoli nam zmniejszyć zakłócenia i dokładność pomiaru. Za pomocą przetwornika „step-down” będziemy zasilать Raspberry i urządzenia dodatkowe takie jak wyświetlacze i przyciski.

Kolejnym problemem do rozwiązania był zakres pomiarowy Raspberry, który mieści się w przedziale 0-3V, a jak wcześniej wspomnieliśmy potencjometr będziemy zasilать napięciem 12V. Rozwiązaniem tego problemu jest zastosowanie dzielnika napięcia zbudowanego z rezystorów precyzyjnych 300k i 1M co pozwoli nam na zmniejszenie napięcia z 12V do 2,77V czyli do zakresu bezpiecznego dla Pico.

$$I = U/R$$

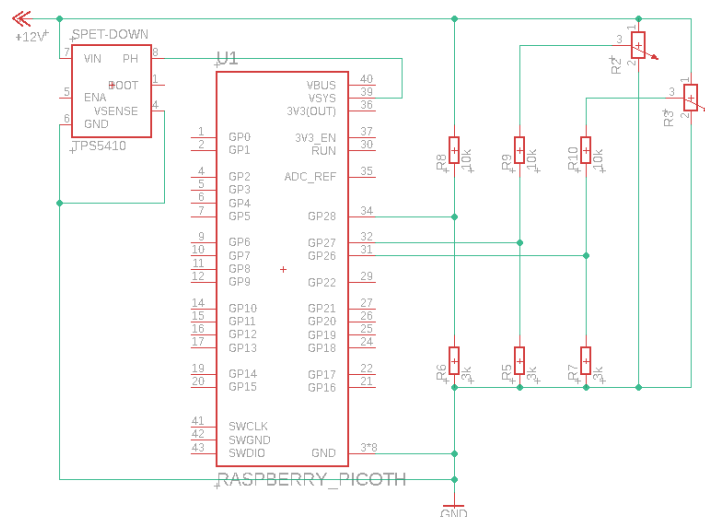
$$I = 12 / (300000 + 1000000) = 9,23 \cdot 10^{-6} \text{ A}$$

$$U = 9,23 \cdot 10^{-6} \cdot 300000 = 2,77 \text{ V}$$

Zastosujemy 3 takie dzielniki, dwa z nich będą odpowiedzialne za obniżenie sygnału z potencjometrów, a jeden będzie wpięty między zasilanie a masę i jego wyprowadzenie będzie podłączone do 3 wyprowadzenia przetwornika ADC celem kompensacji zmian napięcia zasilającego potencjometr.



5 Przetwornica step-down



## Wnioski i podsumowanie

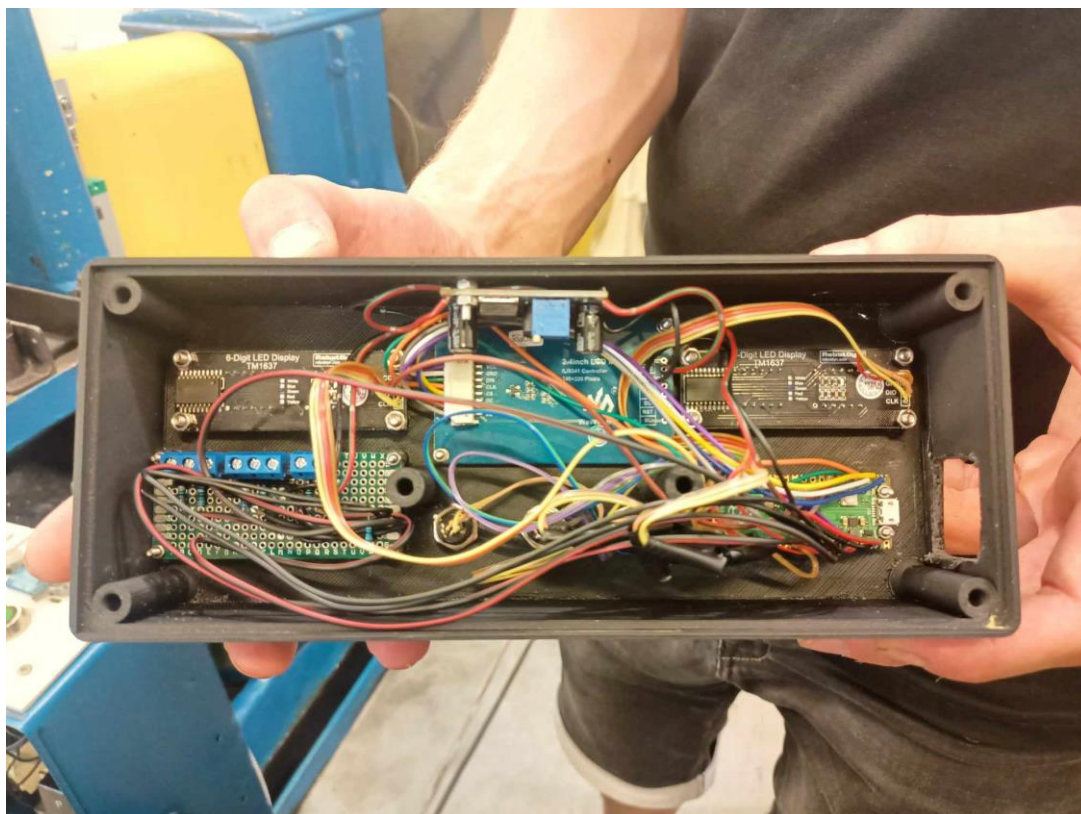
Ostatecznie udało się uruchomić poprawnie działający układ, ale przysporzył on kilka problemów. Niemniej jednak przyda się podczas zajęć laboratoryjnych, gdzie pomoże studentom ustawić dokładne wartości i równoległe ustawienie walców podczas walcowania.

Pierwszym problemem była konfiguracja środowiska umożliwiającego programowanie Pico, ostatecznie problem został rozwiązany przez dodanie rozszerzenia do Arduino IDE pozwalającego na programowanie raspberry pi pico. Oceniając z perspektywy czasu wybór pico był złym pomysłem, lepszym rozwiązaniem byłoby zastosowanie ESP32, które również posiada 12 bitowy przetwornik ADC, ale przysparza mniej problemów związanych z programowaniem oraz wykorzystaniem bibliotek.

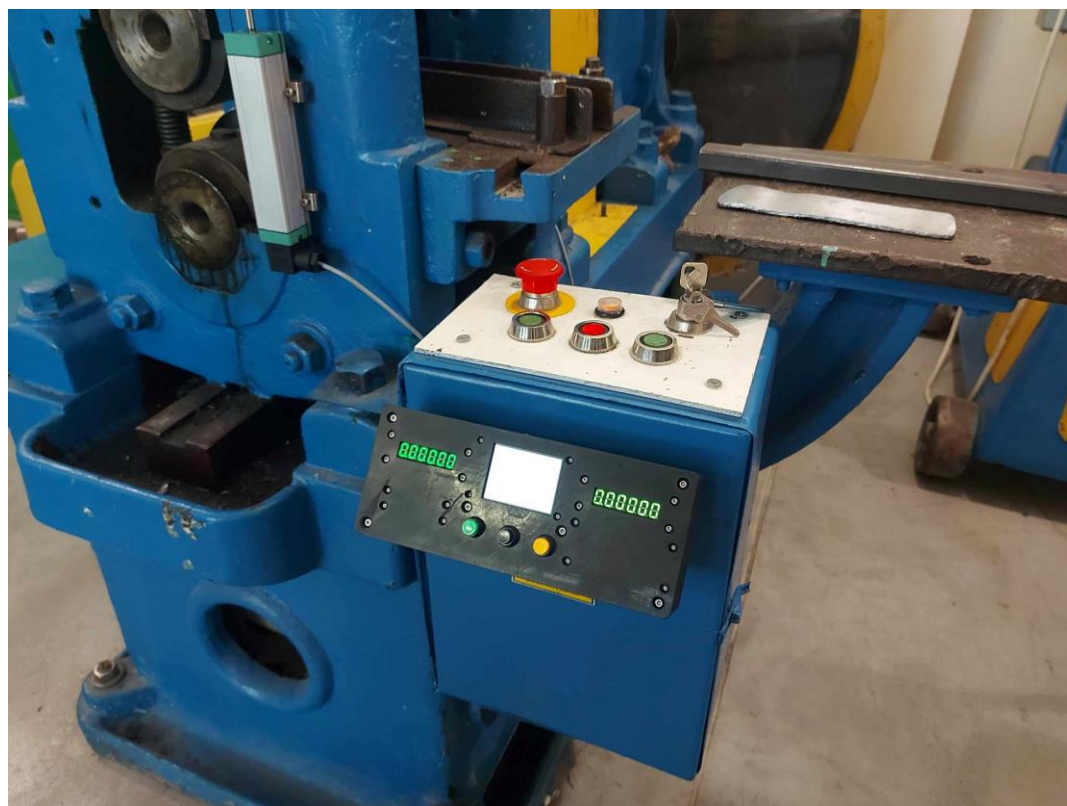
Drugim napotkanym problemem były bardzo duże wahania podczas próbkowania. Jest to prawdopodobnie spowodowane zasilaczem niskiej klasy. Poprawnym rozwiązaniem tego problemu powinien być odpowiedni stabilizator napięcia i dodatkowe filtry żeby zniwelować tętnienia napięcia. Problem został rozwiązany przez pobranie odpowiedniej ilości próbek i wyciągnięciu z nich średniej. To rozwiązanie ma jednak wadę w postaci zwłoki pomiędzy zmianą odległości a wyświetleniem tej wartości.

Trzecim problemem było uruchomienie wyświetlacza LCD. Problem polegał na braku kompilacji przykładu dołączonego do wyświetlacza. Problem nie został rozwiązany. Mogło być to spowodowane wykorzystaniem bibliotek oraz przykładu przystosowanego do pracy z arduino, a nie pico.

Ostatnim problemem są miękkie sprężyny które unoszą jeden z walców do góry. Powodują one słabe dociśnięcie prowadnic do śruby ustalającej grubość walcowania. Powoduje to przekoszenie walca od pewnej wysokości i na wyświetlaczach pojawiają się nieprawdziwe wartości.



7 Zmontowany układ



8 Działający układ