



WYDZIAŁ ELEKTRONIKI,
TELEKOMUNIKACJI
I INFORMATYKI

Imię i nazwisko studenta: Arkadiusz Borowicki

Nr albumu: 179991

Poziom kształcenia: Studia pierwszego stopnia

Forma studiów: stacjonarne

Kierunek studiów: Elektronika i telekomunikacja

Profil: Komputerowe systemy elektroniczne

PROJEKT DYPLOMOWY INŻYNIERSKI

Tytuł projektu w języku polskim: Inteligentny czujnik pojemnościowy oparty na układzie opóźniającym fazę i module Arduino Micro

Tytuł projektu w języku angielskim: Smart capacitance sensor based on a phase delay circuit and a Arduino Micro module

Opiekun pracy: dr hab. inż. Zbigniew Czaja

STRESZCZENIE PRACY

W niniejszym projekcie dyplomowym opracowano i zrealizowano układ laboratoryjny, którego zadaniem jest dokładne wyznaczenie wilgotności względnej powietrza za pomocą czujnika pojemnościowego. Pomiar pojemności opiera się na wykorzystaniu licznika mikrokontrolera pracującego w trybie przechwytywania zdarzeń. Ponadto układ pomiarowy zbudowany jest z komparatorów analogowych z dokładnymi napięciami referencyjnymi.

Opracowano i przetestowano prototyp kompaktowego, pojemnościowego układu laboratoryjnego opartego na 8-bitowym mikrokontrolerze ATmega32U4. Przeanalizowano niedokładność pomiarową pośrednio mierzonej pojemności, a także wykonano badania eksperymentalne. Wyniki potwierdziły, że maksymalny błąd względny wartości pojemności wyniósł mniej niż 0,1%, co odpowiada dokładności pomiaru 0,1 pF dla zakresu mierzonych wartości pomiędzy 160 pF a 200 pF, co z kolei odpowiada co najmniej 0,3% rozdzielczości wilgotności względnej dla komercyjnych czujników RH tj. HS1101 czy Philips H1.

Słowa kluczowe:

czujnik, układ laboratoryjny, mikrokontroler, układ przechwytywania, uniwersalny interfejs, dziedzina czasu, pojemność, wilgotność względna, ATmega32U4, HS1101

Dziedzina nauki i techniki, zgodnie z wymaganiami OECD:

Nauki inżynieryjne i techniczne: elektrotechnika, elektronika i inżynieria informatyczna

ABSTRACT

In this diploma project, a laboratory system was developed and implemented, the task of which is to accurately determine the relative humidity of the air using a capacitive sensor. Capacitance measurement is based on a universal interface for microcontrollers with an internal event capture circuit. The measurement system is based on precise analog comparators, ensuring accurate reference voltages.

A prototype of a compact, capacitive laboratory system based on the 8-bit ATmega32U4 microcontroller was developed and tested. The measurement inaccuracy of the indirectly measured capacitance was analyzed and experimental tests were performed. The results confirmed that the maximum relative error of the capacitance value was less than 0.1%, which corresponds to a measurement accuracy of 0.1 pF for the range of measured values between 160 pF and 200 pF, which in turn corresponds to at least 0.3% relative humidity resolution for commercial RH sensors i.e. HS1101 or Philips H1.

Keywords:

sensor, laboratory system, microcontroller, input capture, universal interface, time domain measurement, capacity, relative humidity, ATmega32U4, HS1101

The field of science and technology, in line with the OECD requirements:

Engineering and technical sciences: electrical engineering, electronics and information engineering

SPIS TREŚCI

1.	WSTĘP I CEL PRACY	8
2.	PRZEGŁĄD METOD POMIAROWYCH	9
2.1.	Metoda oparta na przesuwniku fazowym.....	9
2.2.	Metoda oparta na pomiarze przesunięcia fazowego.....	10
2.3.	Metoda pomiaru ładowania/rozładowania układu RC	11
2.4.	Wybór metody pomiarowej.....	12
3.	BADANIA SYMULACYJNE.....	14
3.1.	Analiza operatorowa obwodu	14
3.2.	Symulacja w programie LTSpice.....	16
3.2.1.	Podbloki układu.....	16
3.2.2.	Wyniki.....	19
4.	BUDOWA UKŁADU LABORATORYJNEGO	23
4.1.	Stanowisko laboratoryjne	23
4.2.	Inteligentny czujnik pojemnościowy	24
4.3.	Układ zasilania	25
4.4.	Moduł Arduino Micro	26
4.5.	Moduł SHTC3.....	27
4.6.	Moduł LCD.....	28
4.7.	Układ pomiarowy	29
4.8.	Złącza diagnostyczne.....	32
4.9.	Layout PCB	33
4.9.1.	Projekt	33
4.9.2.	Realizacja.....	35
5.	UŻYTE UKŁADY PERYFERYJNE MIKROKONTROLERA.....	36
5.1.	System przerwań.....	36
5.2.	USB – Universal Serial Bus.....	37
5.3.	ICP – Input Capture.....	38
5.4.	TWI – Two Wire Interface.....	39
6.	OPROGRAMOWANIE MIKROKONTROLERA	41
6.1.	Struktura programu	41
6.2.	Inicjalizacja urządzenia	41
6.3.	Pętla główna programu	42
6.4.	Funkcje urządzenia	44
6.4.1.	Kalibracja.....	44
6.4.2.	Pomiar temperatury i wilgotności względnej	45
6.4.3.	Pomiar pojemności.....	46

7.	INTERFEJS UŻYTKOWNIKA NA KOMPUTERZE PC	49
7.1.	Strona startowa.....	49
7.2.	Ustawienia ogólne	50
7.2.1.	Eksport wyników pomiarowych do programu Matlab	51
7.2.2.	Eksport dziennika zdarzeń	52
7.3.	Kalibracja	52
7.4.	Funkcje pomiarowe.....	53
7.5.	Charakterystyki pomiarów	56
7.6.	Dziennik zdarzeń	57
8.	WERYFIKACJA EKSPERYMENTALNA.....	59
8.1.	Stanowisko pomiarowe.....	59
8.2.	Pomiar pojemności wzorcowych	61
8.3.	Korekcja pomiarów	66
8.3.1.	Algorytm regresji liniowej.....	66
8.3.2.	Algorytm interpolacji Lagrange'a.....	68
9.	PODSUMOWANIE	73
	WYKAZ LITERATURY	74
	WYKAZ RYSUNKÓW.....	76
	WYKAZ TABEL	78
	WYKAZ LISTINGÓW.....	79
	DODATEK A – SYMULACJA LTSPICE	80
	DODATEK B – SCHEMAT IDEOWY	82
	DODATEK C – SCHEMAT MONTAŻOWY	84

WYKAZ WAŻNIEJSZYCH SKRÓTÓW I OZNACZEŃ

CMOS – (ang. *Complementary Metal-Oxide-Semiconductor*) technologia wykonywania tranzystorów polowych

ICES1 – (ang. *Input Capture Edge Select*) wybór zbocza wyzwalającego moduł przechwytywania zdarzeń zewnętrznych

ICIE1 – (ang. *Input Capture Interrupt Enable*) zezwolenie na obsługę przerwań od modułu przechwytywania zdarzeń zewnętrznych

ICNC1 – (ang. *Input Capture Noise Canceler*) filtr szumów modułu przechwytywania zdarzeń zewnętrznych

ICP – (ang. *Input Capture*) moduł przechwytywania zdarzeń zewnętrznych

ISP – (ang. *In System Programming*) szeregowe programowanie mikrokontrolera

LCD – (ang. *Liquid Crystal Display*) wyświetlacz ciekłokrystaliczny

MCU – (ang. *Microcontroller Unit*) mikrokontroler

PC – (ang. *Personal Computer*) komputer osobisty

PCB – (ang. *Printed Circuit Board*) płytka z połączeniami elektrycznymi i punktami lutowniczymi

PLL – (ang. *Phase-Locked Loop*) układ powielania częstotliwości taktowania mikrokontrolera

PWM – (ang. *Phase-Width Modulation*) modulacja szerokości impulsu

SMD – (ang. *Surface-Mount Device*) element elektroniczny, montowany powierzchniowo

TWEA – (ang. *TWI Enable Acknowledge*) potwierdzenie odebrania danych w interfejsie TWI

TWI – (ang. *Two-Wire Interface*) dwuprzewodowy interfejs, do wymiany informacji z urządzeniami

TWINT – (ang. *TWI Interrupt*) flaga przerwania interfejsu TWI

TWSR – (ang. *TWI Status Register*) rejestr statusowy dwu-przewodowego interfejsu szeregowego

TWSTA – (ang. *TWI Start*) bit generujący warunek startu na magistrali TWI

TWSTO – (ang. *TWI Stop*) bit generujący warunek stopu na magistrali TWI

USB – (ang. *Universal Serial Bus*) uniwersalna magistrala szeregową

XOR – (ang. *Exclusive Or*) bramka realizująca funkcję alternatywy rozłącznej

1. WSTĘP I CEL PRACY

Parametry środowiskowe takie jak temperatura i wilgotność są jednymi z głównych czynników wpływających na procesy technologiczne. Niejednokrotnie podczas procesów wytwórczych różnego rodzaju firm niezbędne jest spełnienie rygorystycznych wymagań jakościowych dla produkowanych wyrobów.

Przykładem może być produkcja scalonych układów elektronicznych, parametry środowiskowe wpływają na reakcje zachodzące na strukturach krzemu. Monitorowanie i korygowanie tych czynników jest kluczowe dla zachowania znamionowych parametrów tychże układów. Ponadto specjalistyczne urządzenia cechujące się niezawodną i długotrwałą pracą wymagają dokładnej stabilizacji wilgotności względnej, temperatury a czasami ciśnienia w środowisku wytwórczym. Podobnymi cechami charakteryzuje się sposób przechowania podzespołów, ponieważ zaniedbanie warunków środowiskowych może wpływać na korozję wyprowadzeń czy zwiększyć prawdopodobieństwo wyładowań ESD i uszkodzenia struktury krzemowej. Systemy monitorujące parametry środowiskowe znalazły także zastosowanie przy produkcji żywności. Zapewnienie optymalnej temperatury i wilgotności pozwala np. na wysuszenie składników do wartości nadającej się do ich dalszej obróbki. Natomiast w gospodarstwach rolnych szczególną uwagę zwraca się na optymalną wilgotność gleby wymaganą do poprawnego wzrostu roślin. Typowo systemy monitorujące parametry składają się z serii czujników dostarczających informacji o warunkach panujących w otoczeniu w postaci sygnałów elektrycznych.

Celem niniejszej pracy dyplomowej jest zaprojektowanie i realizacja układu laboratoryjnego umożliwiającego dokładną kalibrację czujnika HS1101, jak i monitorowanie wilgotności względnej w czasie rzeczywistym. Układ ma umożliwić także zmianę zakresu pomiarowego oraz konfigurację zmiennych parametrów. Sterowanie układem pomiarowym opiera się na module Arduino Micro bazującym na 8-bitowym mikrokontrolerze ATmega32U4. Uniwersalny interfejs czujnik-mikrokontroler bazuje na układzie przechwytywania zdarzeń wbudowanym w mikrokontroler. W ramach pracy powstała również aplikacja na komputer PC umożliwiająca sterowanie urządzeniem pomiarowym.

Na początku pracy zawarto porównanie metod pomiarowych dla małych pojemności opartych na układach z przesunięciem fazowym. Wyprowadzono model matematyczny idealnego obwodu pomiarowego, a model z uwzględnieniem pasożytniczych elementów poddano symulacjom. W kolejnych rozdziałach opisano także użyte układy peryferyjne mikrokontrolera oraz omówiono jego oprogramowanie skupiając się na funkcjach pomiarowych. Szczególną uwagę poświęcono wyjaśnieniu funkcji aplikacji na komputer. W ostatnim rozdziale przedstawiono weryfikację układu laboratoryjnego podając analizie dokładności pomiarowe z wykorzystaniem regresji liniowej oraz interpolacji Lagrange'a.

2. PRZEGŁĄD METOD POMIAROWYCH

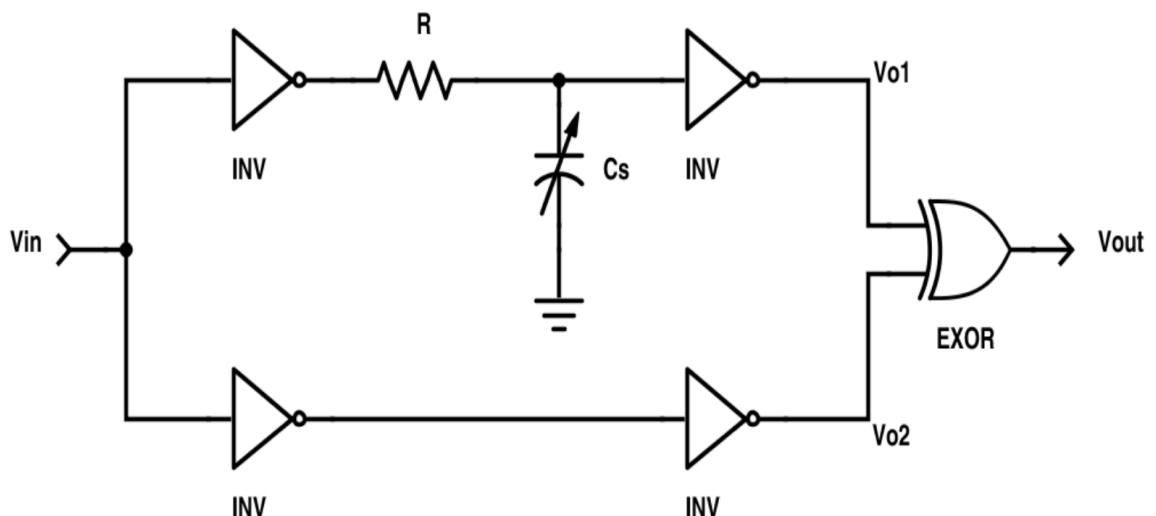
W niniejszym rozdziale zostaną zaprezentowane metody pomiarowe umożliwiające pomiar pojemności rzędu pF. Omawiane metody bazują na uniwersalnym interfejsie czujnik-mikrokontroler. Szczególną uwagę zwrócono na wykorzystaną w projekcie dyplomowym metodę pomiarową opartą na układzie opóźniającym RC.

2.1. Metoda oparta na przesuwniku fazowym

Pomiar wartości opóźnienia fazowego układu RC, ze znaną rezystancją umożliwia określenie wartości pojemności [1]. Przebiega on w następujący sposób: sygnał prostokątny V_{in} przechodzi przez dwa inwertery w obu kanałach. Ładowanie i rozładowanie pojemności badanej C_s (rys. 2.1.) wprowadza dodatkowe opóźnienie (górnego kanału) o wartości zależnej od wartości R i C .

Porównanie przebiegu wejściowego (dolny kanał) oraz opóźnionego o stałą RC odbywa się za pośrednictwem bramki XOR. Dzięki zastosowaniu w obu kanałach tej samej liczby bramek NOT wyeliminowano opóźnienia przez nie wprowadzane.

Częstotliwość sygnału prostokątnego V_{in} jest kluczowa z punktu widzenia dokładności pomiarowej, gdyż przekroczenie przez opóźnienie fazowe wartości 45° wprowadza znaczący błąd. Uzyskany na wyjściu bramki XOR sygnał PWM może być przekształcony do postaci analogowej, za pośrednictwem filtra dolnoprzepustowego. W takim przypadku wynikową wielkością fizyczną, niosącą informację o wartości pojemności jest napięcie, które można zmierzyć np. za pomocą przetwornika A/C mikrokontrolera.

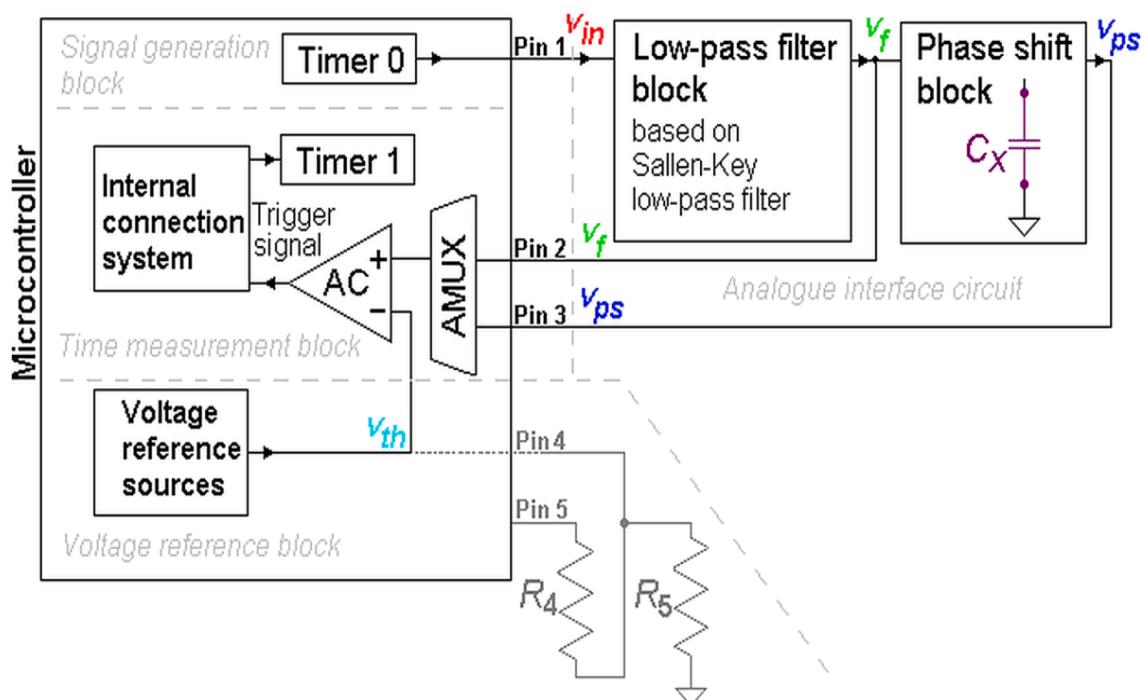


Rys. 2.1. Schemat ideowy układu pomiarowego przesuwnika fazowego [1].

2.2. Metoda oparta na pomiarze przesunięcia fazowego

W tej metodzie mikrokontroler sterujący układem pomiarowym realizuje dwa główne zadania. Generuje sygnał prostokątny o ustalonym okresie (rys. 2.2.), mierzy czas opóźnienia wprowadzanego poprzez filtr dolnoprzepustowy oraz układ opóźniający RC zawierający badaną pojemość. Głównym zadaniem filtra dolnoprzepustowego jest transformacja sygnału prostokątnego V_{in} pochodzącego bezpośrednio z mikrokontrolera na sygnał w przybliżeniu sinusoidalny V_f . Pobudzanie układu RC sygnałem V_f powoduje przesunięcie w fazie i zmniejszenie amplitudy tego sygnału na wyjściu wspomnianego bloku uzyskując sygnał V_{ps} .

Mikrokontroler zmierzywszy czasy po których następuje przekroczenie wartości wewnętrznego napięcia referencyjnego V_{th} , przez sygnały V_f i V_{ps} wyznacza wartości pojemości badanej C_x znajdującej się wewnątrz bloku układu opóźniającego [2]. Momenty przekroczenia napięcia V_{th} przez sygnały V_f i V_{ps} wykrywane są za pośrednictwem wbudowanego w mikrokontroler modułu komparatora analogowego. W przypadku, gdy mikrokontrolery nie posiadają wewnętrznych napięć odniesienia to używa się do ustalania wartości V_{th} dzielnika rezystancyjnego R_4 i R_5 . Zapewnienie stabilnego napięcia odniesienia w przypadku tej metody, jest kluczowe, dla uzyskania dużej dokładności pomiarowej. Również wydłużenie okresu przebiegu prostokątnego V_{in} pozwala na dokładniejsze wyznaczenie opóźnienia między sygnałami V_f i V_{ps} .



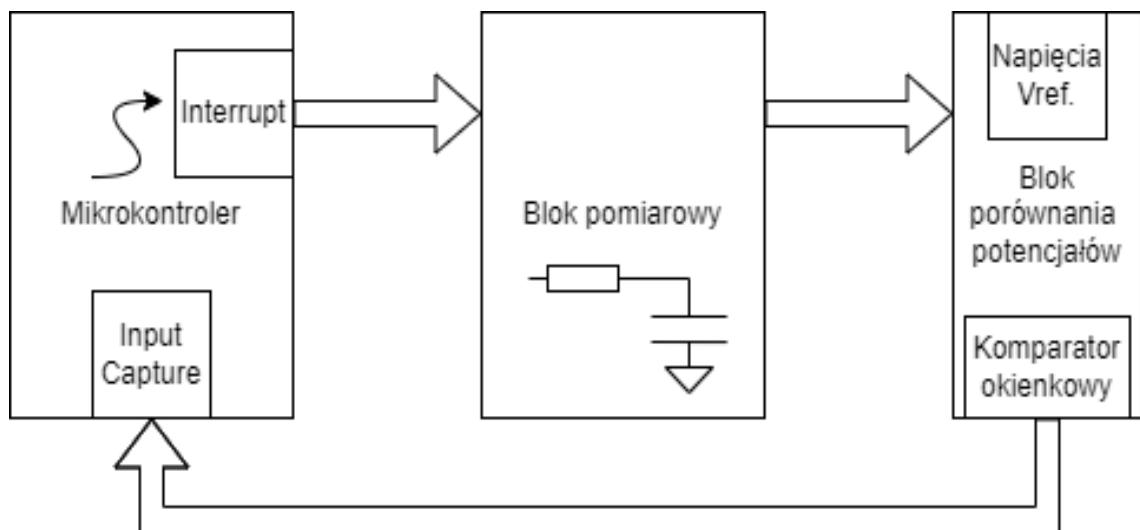
Rys. 2.2. Schemat blokowy układu dla metody pomiaru przesunięcia fazowego [2].

2.3. Metoda pomiaru ładowania/rozładowania układu RC

Zmierzony czas ładowania oraz rozładowania układu RC może posłużyć do wyznaczenia wartości badanej pojemności C, przy znanej rezystancji R [3]. Zadaniem mikrokontrolera jest pobudzenie układu RC oraz pomiar czasów odpowiedzi tego układu (rys. 2.3.). Wymagane jest, aby blok porównywania potencjałów generował sygnały informujące mikrokontroler o przekroczeniu ustalonego progu napięcia [4]. Wystąpienie tych zdarzeń służy do sterowania licznikiem mikrokontrolera pracującego w trybie przechwytywania.

Stąd każdorazowe przekroczenie potencjałów progowych przez napięcie na ładowanej/rozładowywanej pojemności uruchamia obsługę przerwania, wewnętrzną którego zostaje zapamiętany czas od poprzedniego wystąpienia zdarzenia. Wielokrotny pomiar tego czasu pozwala na uśrednianie wyników. Moduł mierzący czasy ładowania oraz rozładowania układu RC jest częścią mikrokontrolera stąd bezpośrednio wzywana jest systemem przerwań (rys. 2.3.).

Wspomniany moduł zawiera również filtr szumów, dlatego każde zakłócenie o czasie krótszym od 250 ns zostanie stłumione nie doprowadzając tym samym do niechcianego wywołania obsługi przerwania. Wadą filtra szumów jest opóźnienie sygnałów komparatora okienkowego, co skutkuje dłuższym czasem obsługi żądania.

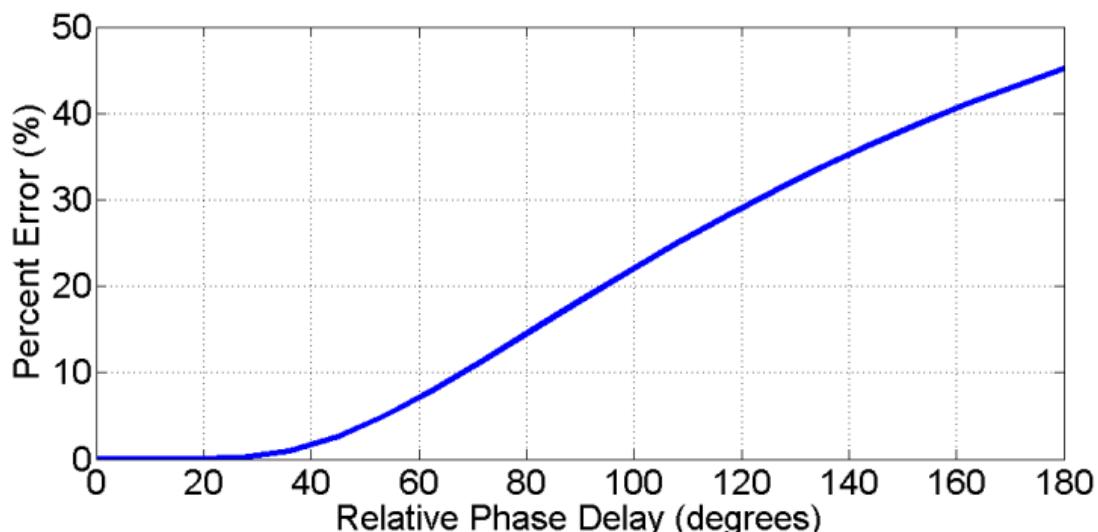


Rys. 2.3. Schemat blokowy układu dla metody pomiarowej bazującej na pomiarze czasu ładowania/rozładowania układu RC.

2.4. Wybór metody pomiarowej

Na potrzeby realizacji projektu rozważono zalety i wady każdej z wymienionych metod pomiarowych. Metoda oparta na przesuwniku fazowym została wykluczona ze względu na brak możliwości dokładnego określenia momentu przełączenia inwertera dołączonego do pojemności badanej. Noty katalogowe układów cyfrowych podają, że typową wartością jest połowa napięcia zasilania. Znajomość tej danej jest kluczowa dla uzyskania dużej dokładności pomiarowej. Tranzystory CMOS obwodów wejściowych układów cyfrowych wytworzone w strukturze wewnętrznej krzemu posiadają między innymi różne parametry ruchliwości nośników ładunku w zależności od rodzaju kanału. Rozbieżność tych wartości zależna jest od czystości krzemu produkowanych układów. Zatem parametry te mogą być odmienne w różnych partiach chipów. Czynnikiem dodatkowo wpływającym na punkt przełączenia stanu bramki jest temperatura otoczenia oraz samej struktury krzemowej, która pod wpływem wydzielanej energii dodatkowo się ogrzewa, zmieniając tym samym napięcie przełączające.

Wadą przesuwnika fazowego jest także duży błąd wzajemny, rosnący wraz ze wzrostem przesunięcia fazowego (rys. 2.4.). Chęć uzyskania wysokiej precyzji wymusza zastosowanie maksymalnych przesunięć fazowych sięgających granicy 20° . Pomiar opóźnienia mierzony za pomocą liczników nie zapewnia wystarczającej rozdzielczości pomiarowej, ze względu na niewielkie różnice czasowe pomiędzy sygnałami oraz ograniczoną rozdzielczość liczników mikrokontrolera. Czynnikiem zmniejszającym ten błąd może być zastosowanie tranzystorów MOS [5], których zadaniem jest całkowite rozładowanie lub naładowanie pojemności, w momencie przełączenia stanów logicznych na wyjściu inwertera. Niestety rozwiązanie to wnosi do układu znaczne pojemności pasektywiczne, które wymagają doświadczalnego wyznaczenia i korekcji.

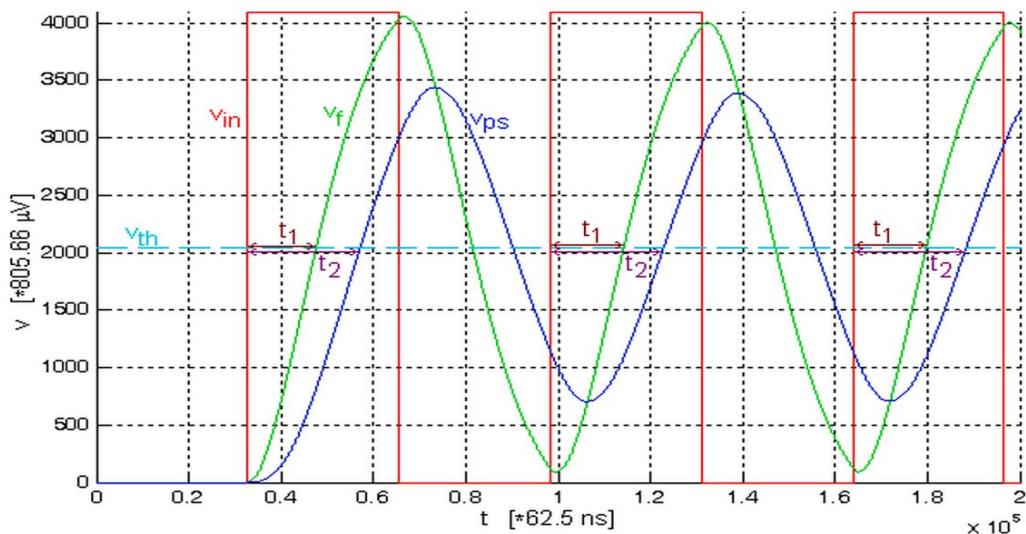


Rys. 2.4. Charakterystyka błędu wzajemnego w zależności przesunięcia fazowego [5].

Metoda oparta na pomiarze przesunięcia fazowego [2], również posiada wady. Wybrany zakres mierzonych pojemności nie jest łatwo ustawialny, zmiana zakresu wymaga przestrojenia filtra konwertującego sygnał prostokątny na sinusoidalny co wynika ze zmiany sygnału pobudzającego układ pomiarowy. Zbyt wysoka wartość częstotliwości sygnału pobudzającego znacząco ogranicza uzyskanie dużej dokładności pomiarowej, gdyż pomiary dokonywane są w domenie czasowej (rys. 2.5.). Istotnym czynnikiem staje się rozdzielczość licznika mikrokontrolera. Stąd dobór odpowiedniej częstotliwości sygnału pobudzającego wpływa na dokładność pomiarową.

Zakres mierzonych pojemności w projekcie określono od 100 do 300 pF. Tak szeroki zakres, przy zachowaniu akceptowalnej rozdzielczości mierzonej pojemności (0,1 pF), wymagałby zastosowania sygnału pobudzającego o częstotliwości na poziomie 180 Hz. Następne wykonanie serii 128 wyników, implikowałoby czas odpowiedzi układu pomiarowego na około 1 s. Dodatkowym czynnikiem wpływającym na dokładność jest tłumienie sygnału wprowadzane przez filtr dolnoprzepustowy. Każdorazowa konfiguracja filtru wymusza wyznaczenie współczynników tłumienia dla zadanej częstotliwości.

Metoda pomiarowa oparta na pomiarze czasu ładowania i rozładowania układu RC zapewnia największą dokładność pomiarową spośród omawianych [3]. Dodatkową jej zaletą jest prostota ustawiania napięć referencyjnych oraz rezystancji wzorcowych. Czasy ładowania i rozładowania zależą od rezystancji R, zatem zakres pomiarowy zostaje ustalony za pomocą jednego elementu. Straty energii wynikające z funkcjonowania układu pomiarowego zależą wyłącznie od prądów polaryzujących obwody wejściowe zastosowanych komparatorów analogowych. Zapewnienie stabilnego napięcia zasilającego oraz napięć referencyjnych gwarantuje wysoką dokładność pomiaru. Metoda oparta na ładowaniu/rozładowaniu układu RC spełnia wszystkie założone cele projektowe, co zaważyło na jej wyborze i realizacji w układzie rzeczywistym.



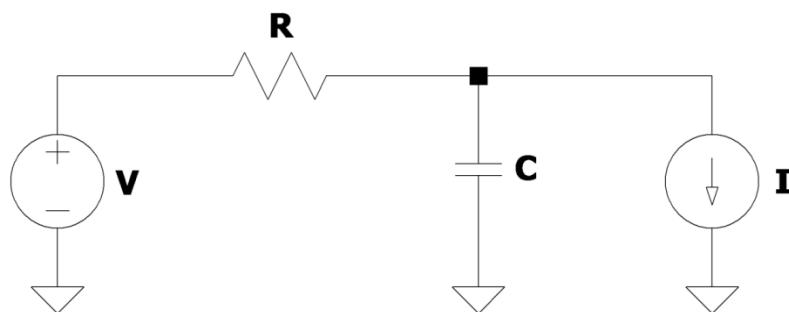
Rys. 2.5. Sygnały występujące w metodzie opartej na pomiarze przesunięcia fazowego [2].

3. BADANIA SYMULACYJNE

W rozdziale zostanie wyprowadzony z wykorzystaniem analizy operatorowej, model matematyczny obwodu, a także zostaną przedstawione teoretyczne charakterystyki czasów ładowania i rozładowania przy użyciu programu Matlab. Przeprowadzono również szczegółową symulację obwodu w programie LTSpice w celu oceny wpływu pojemności pasożytniczych na pracę układu.

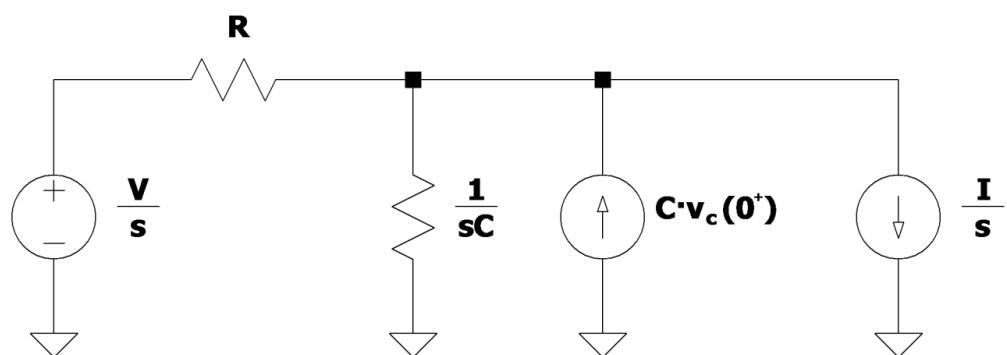
3.1. Analiza operatorowa obwodu

Obwód pomiarowy można przedstawić przy pomocy idealnych źródeł prądowych i napięciowych (rys. 3.1.). Źródło prądowe (I) symbolizuje prąd wejściowy komparatorów (prąd polaryzujący tranzystory wejściowe układu scalonego). Zgodnie z dokumentacją układu TLV3502AID prąd ten może osiągać maksymalne wartości $\pm 20 \text{ pA}$ [6]. Źródło napięciowe (V) odpowiada natomiast napięciu na wyjściu układu buforującego.



Rys. 3.1. Obwód pomiarowy.

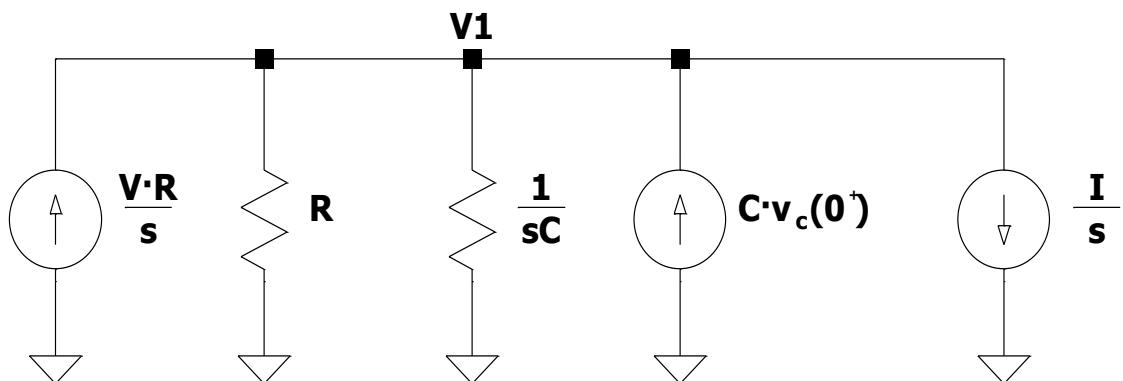
Przyjmując, że stan napięcia na wyjściu mikrokontrolera zmienił się w sposób skokowy w chwili $t = 0$, to w układzie pomiarowym będzie obowiązywać stan nieustalony. Do opisu matematycznego takich stanów służy analiza operatorowa [7] (rys. 3.2.).



Rys. 3.2. Analiza operatorowa obwodu pomiarowego.

Pojemność C zastąpiono równoległym połączeniem rezystancji o wartości $\frac{1}{s \cdot C}$ oraz źródła prądowego o wartości $C \cdot v_C(0^+)$, gdzie $v_C(0^+)$ oznacza wartość chwilową napięcia na pojmemności w chwili czasu $t \rightarrow 0^+$. Rezystancja R ma charakter liniowy, dlatego nie podlega przekształceniom operatorowym.

Do rozwiązania układu wykorzystano metodę potencjałów węzłowych. W tym celu należało dokonać kolejnego przekształcenia (rys. 3.3.), w którym zamieniono rzeczywiste źródło napięciowe utworzone z elementów rezystancji R oraz źródła $\frac{V}{s}$ na rzeczywiste źródło prądowe.



Rys. 3.3. Przekształcony obwód pomiarowy w dziedzinie operatorowej.

Dysponując modelem po przekształceniach możliwe jest wyprowadzenie wzoru na napięcie w węźle $V1$, czyli na napięcie na badanej pojmemności C (3.1.).

$$V1 = \frac{V \cdot \frac{R}{s} + C \cdot v_C(0^+) - \frac{I}{s}}{\frac{1}{R} + s \cdot C} \quad (3.1.)$$

Po przekształceniach wzoru (3.1.) oraz dokonaniu odwrotnej transformaty Laplace'a otrzymano wzór (3.2.) opisujący napięcie w dowolnej chwili czasu stanu nieustalonego.

$$V1 = V + \exp\left(-\frac{t}{R \cdot C}\right) \cdot \left(v_C(0^+) - V + \frac{I}{R}\right) - I \cdot R \quad (3.2.)$$

Podobnie, przekształcając wzór (3.2.) uzyskano zależność czasu ładowania oraz rozładowania pojmemności od wartości tej pojmemności (3.3.). Wybór danych do obliczeń czasów ładowania bądź rozładowania zależy od wartości parametrów użytych we wzorze (3.3.), ponieważ wyznaczanie czasu ładowania pojmemności będzie obejmowało inne warunki początkowe oraz inne wartości napięcia wyjściowego z bufora trójstanowego (V).

$$t = -C \cdot R \cdot \log \left(\frac{V_1 - V + I \cdot R}{v_c(0^+) - V + I \cdot R} \right) \quad (3.3.)$$

W kolejnym przekształceniu wzoru (3.3.) otrzymano wartość pojemności, w zależności od znanego czasu ładowania lub rozładowania (3.4.). Parametry V, I oraz R przyjmują stałą wartość na czas trwania stanu nieustalonego. Wyznaczanie pojemności może przebiegać zarówno z pomierzonego czasu ładowania jak i rozładowania dla obu przypadków parametry V_1, V oraz $v_c(0^+)$ będą posiadały różne wartości.

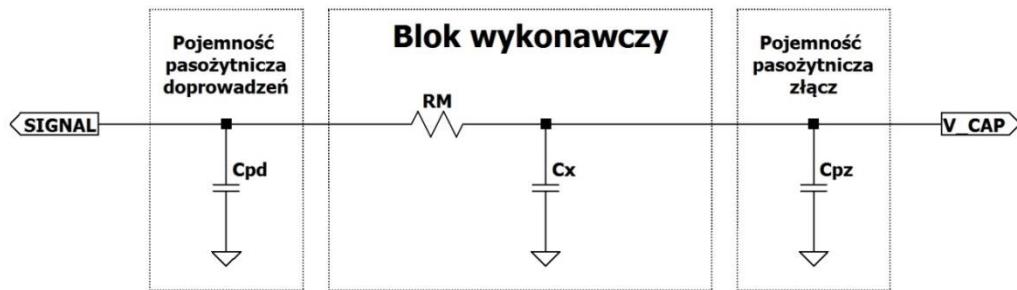
$$C = \frac{-t}{R \cdot \log \left(\frac{V_1 - V + I \cdot R}{v_c(0^+) - V + I \cdot R} \right)} \quad (3.4.)$$

3.2. Symulacja w programie LTSpice

Opis matematyczny udowodnił, że pomiar wartości pojemności, w idealnych warunkach jest możliwy poprzez zmierzenie czasu ładowania lub rozładowania pojemności. W podrozdziale poddano analizie obwód pomiarowy z uwzględnieniem pojemności pasożytniczych ścieżek PCB oraz tranzystorów obwodów wejściowych wykorzystanych układów scalonych.

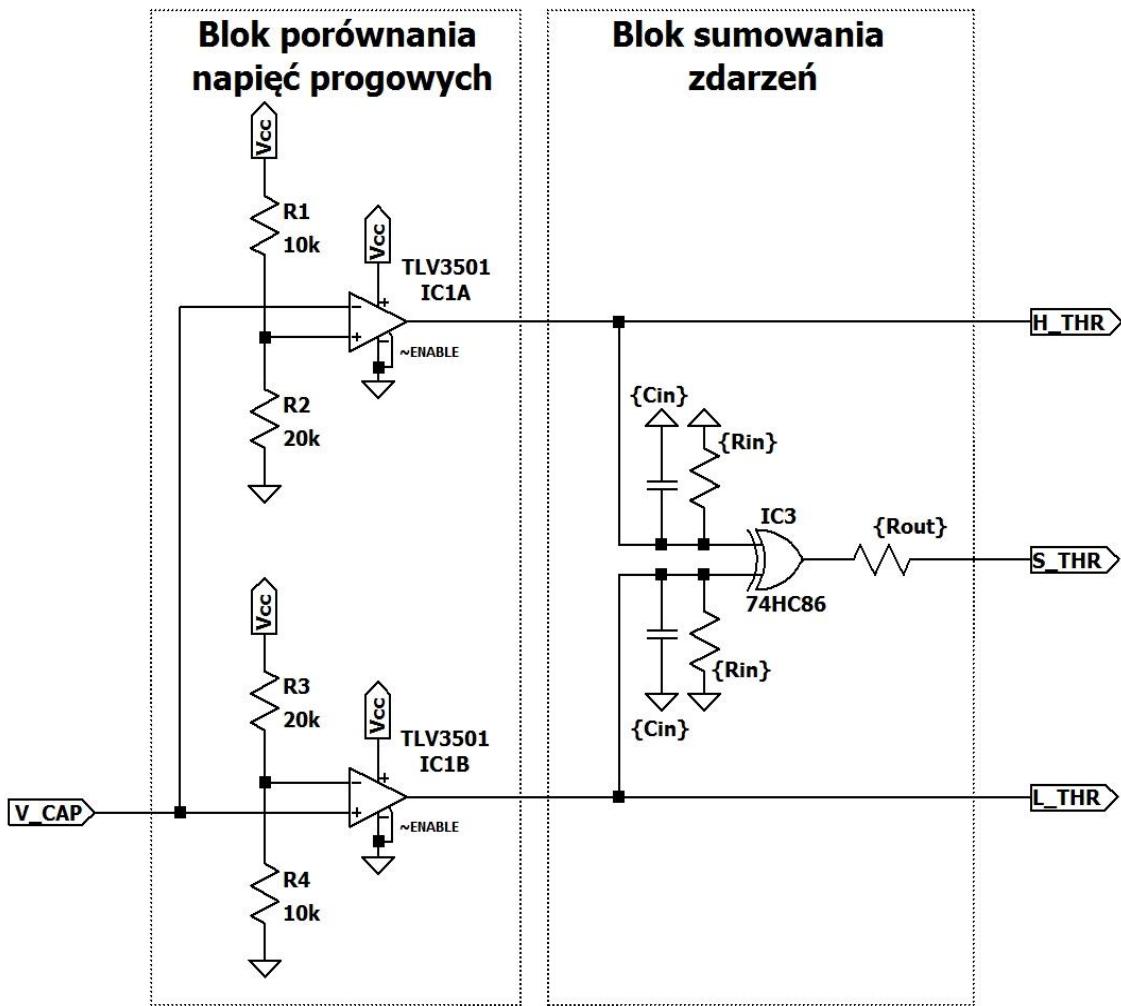
3.2.1. Podbloki układu

Blok pomiarowy (rys. 3.4.) symuluje pojemności pasożytnicze związane ze ścieżkami doprowadzającymi oraz pojemności złącz pomiarowych i diagnostycznych. Do wyprowadzenia SIGNAL dołączono jedno z wyjść mikrokontrolera poprzez trójstanowy bufor. Badana pojemność C_x oraz rezistor pomiarowy R_M tworzą układ RC. Rezystor ogranicza prąd ładowania i rozładowania pojemności przez sygnał pobudzający. Ważnym parametrem jest wartość rezystancji, ponieważ zbyt wysoka spowoduje, że prądy polaryzujące komparatorów nasycą układ pomiarowy. W takim wypadku ograniczony prąd popłynie do obwodów wejściowych nie ładując pojemności badanej.



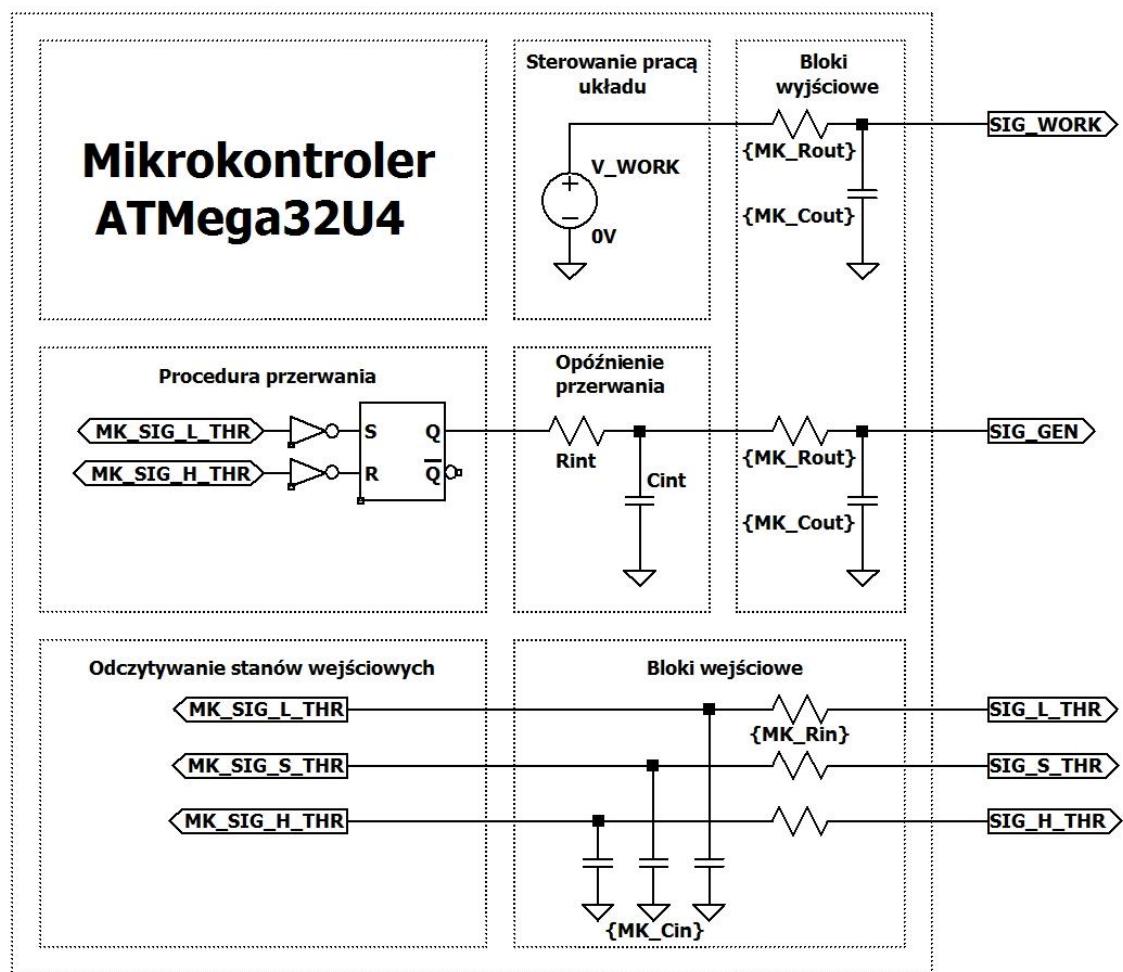
Rys. 3.4. Schemat bloku pomiarowego – symulacja.

Blok przetwarzający napięcie występujące na badanej pojemności (rys. 3.5.) składa się z dwóch komparatorów pracujących w trybie okna, które porównują poziom napięcia do zadanych progów napięć. Jeżeli napięcie uzyska poziom większy od górnego napięcia odniesienia, lub mniejszy od dolnego, będzie to skutkowało ustaleniem stanu wysokiego na wyjściu bramki XOR. Zbocze narastające tego sygnału spowoduje wyzwolenie przerwania i zmianę stanu sygnału generowanego przez mikrokontroler.



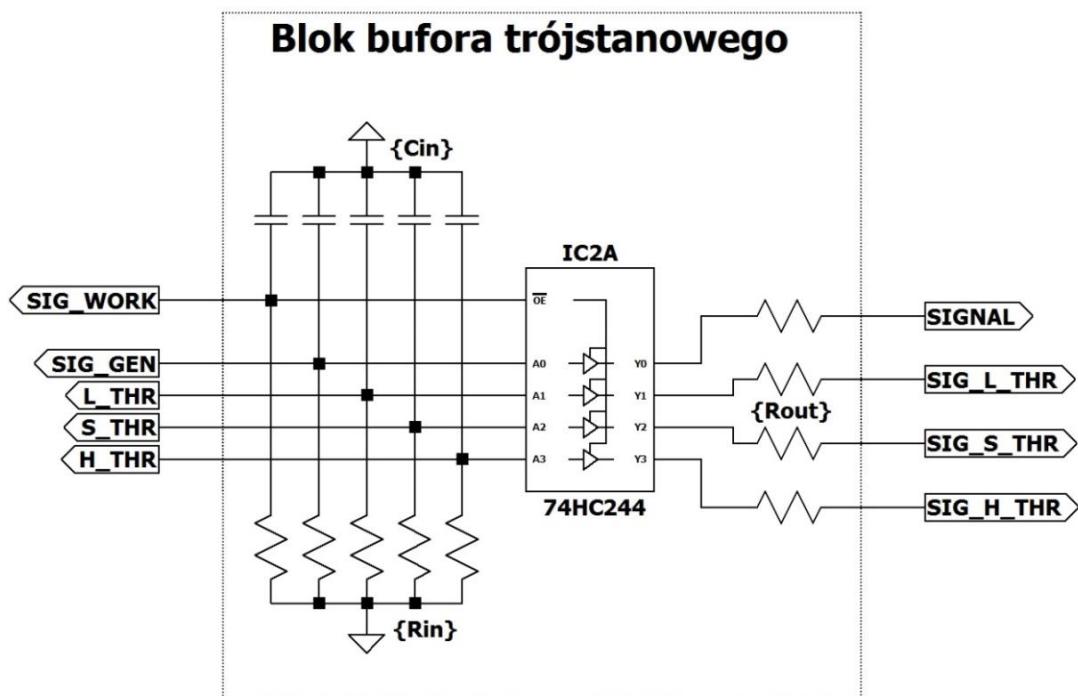
Rys. 3.5. Schemat bloku porównywania napięć progowych – symulacja.

Blok symulujący działanie mikrokontrolera (rys. 3.6.) przewiduje inne wartości pojemności i rezystancji pasożytniczej wejść oraz wyjść układu scalonego. Na potrzeby symulacji zaimplementowano uproszczoną obsługę przerwania składającą się z przerutnika RS. Z wywołaniem przerwania ściśle związane jest również opóźnienie jakie wnosi wykonanie kolejnych instrukcji wewnętrz jego obsługi. Z tego względu zastosowano układ opóźniający, symulujący czas wykonania przerwania.



Rys. 3.6. Schemat podłączenia mikrokontrolera – symulacja.

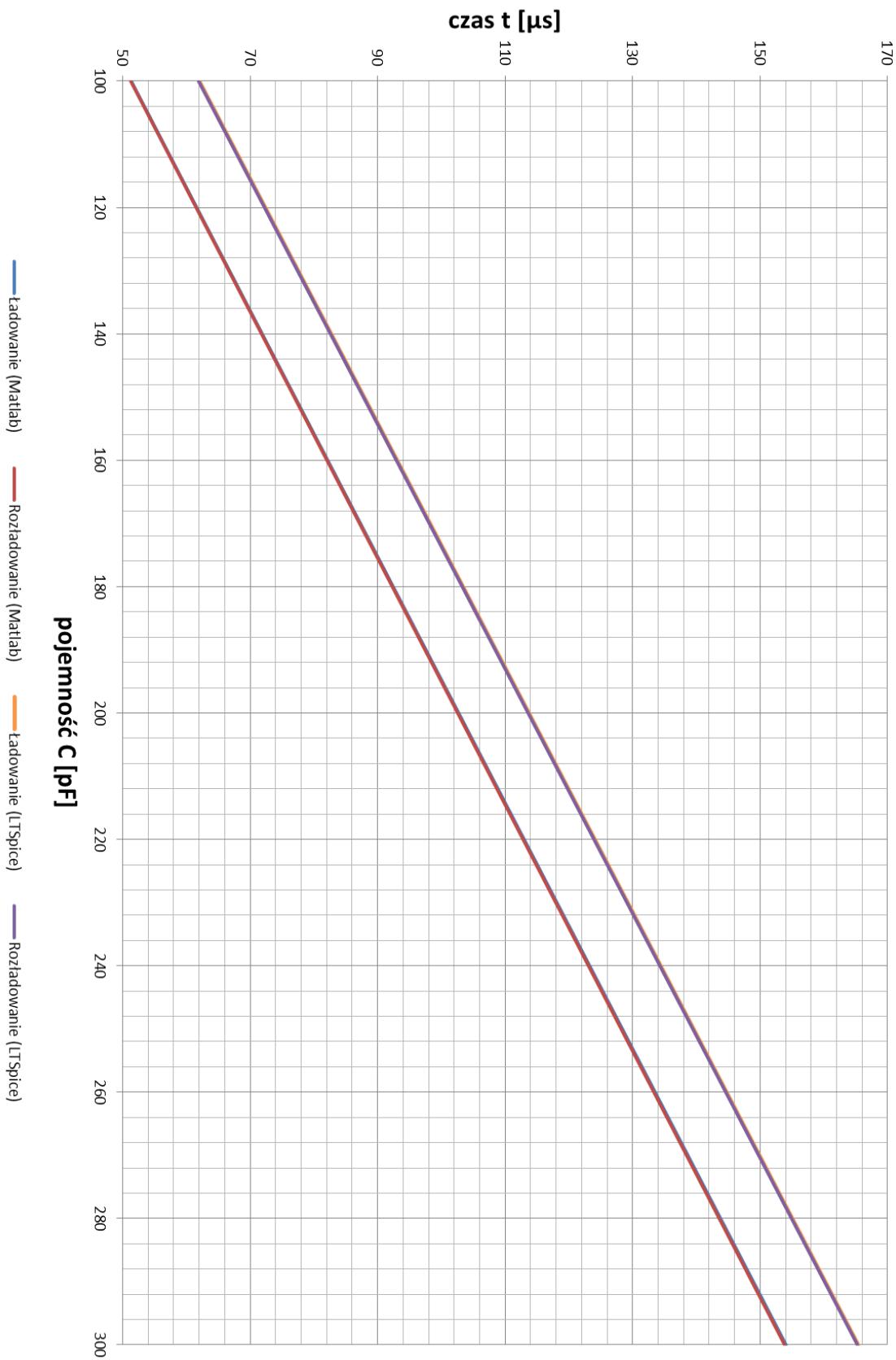
Blok bramki trójstanowej (rys. 3.7.) obejmuje pojemności i rezystancje pasożytnicze wejść układu. Wykorzystane do projektu bramki i bufory są wykonane w technologii CMOS, która cechuje się pojemnością pasożytniczą bramki tranzystora oraz niewielkim prądem upływu. Te parametry są reprezentowane przez pojemność Cin oraz rezystancję Rin . Podobna zasada występuje na wyjściach bramek, załączony jeden z tranzystorów *pull up* lub *pull down*, posiada niezerową rezystancję dren – źródło, co zostało również wzięte pod uwagę w postaci rezystancji $Rout$ na wyjściach. Do projektu wykorzystano bufor, ponieważ parametry jego wyjść są lepsze w porównaniu do wyjść mikrokontrolera (mniejsze pojemności i rezystancje pasożytnicze). Parametryzowanie obwodu pozwala na szybkie dostosowanie symulacji po realizacji fizycznego układu. Prawdą jest, że wejścia i wyjścia układów scalonych mogą posiadać różne wartości pojemnością pasożytniczą, nawet w obrębie pojedynczego układu. Wynika to ze struktury krzemowej oraz nieczystości krzemu wykorzystanego do produkcji. Dlatego też symulacja nie odzwierciedli dokładnego zachowania rzeczywistego układu, w którym występuje znacznie więcej zjawisk.



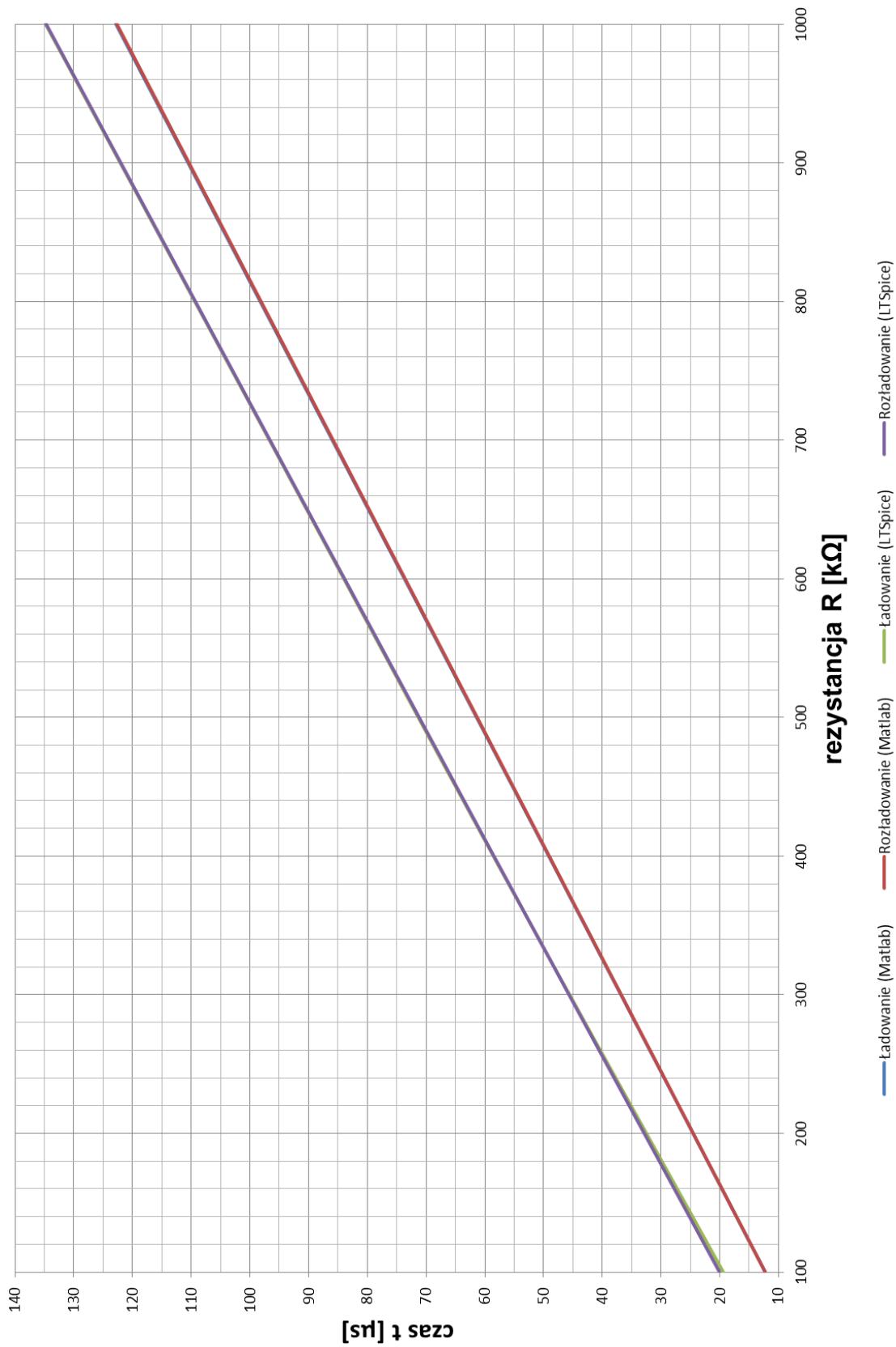
Rys. 3.7. Schemat bufora trójstanowego – symulacja.

3.2.2. Wyniki

Symulacje umożliwiają wyznaczenie wartości czasu ładowania i rozładowania układu badanego. Niewielki prąd polaryzujący komparatorów analogowych powoduje, że czasy te są do siebie zbliżone. Podczas realizacji fizycznej zachodzi konieczność doświadczalnego wyznaczenia prądu polaryzacji w celu poprawy dokładności pomiarowych. Charakterystyki czasu ładowania i rozładowania wyznaczone drogą obliczeń oraz symulacji mają charakter w przybliżeniu liniowy (rys. 3.8.). Charakterystyka pozwala na zobrazowanie wpływu pojemności i rezystancji pasożytniczych, które nie zostały wzięte pod uwagę w modelu matematycznym. Występowanie elementów pasożytniczych w układzie pomiarowym wydłuża czas ładowania i rozładowania pojemności. Wszystko to spowodowane jest poprzez zwiększoną pojemność elementu badanego o pojemność pasożytniczą występujące w układzie. Dodatkowym czynnikiem wpływającym na czas pomiaru jest czas potrzebny na obsługę przerwania. Opóźnienie jakie zostaje wprowadzone przez nie szacuje się na około 4 µs. Na ten czas składa się wywołanie przerwania, które wynosi 5 cykli zegara mikrokontrolera oraz wykonanie instrukcji wewnętrznej obsługi przerwania. Wszystko to powoduje przesunięcie charakterystyki w stronę dłuższych czasów ładowania i rozładowania o około 10 µs w całym mierzonym zakresie pojemności. W celu kompensacji wpływu elementów pasożytniczych, w obliczeniach użyto funkcji korygującej wartość zmierzonej pojemności. Współczynniki funkcji zostały wyznaczone doświadczalnie na podstawie pomiarów znanych, dokładnych wartości pojemności.



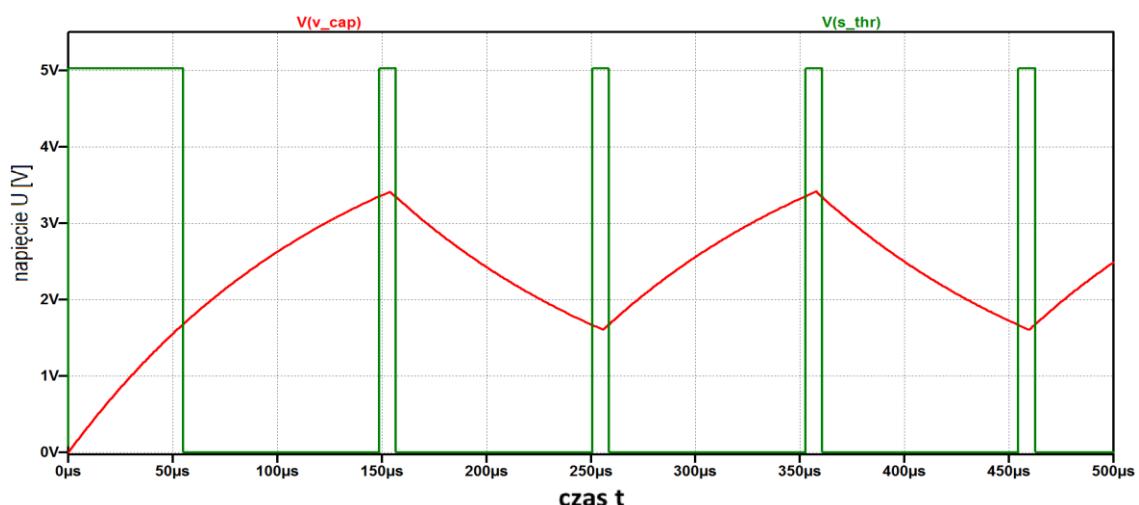
Rys. 3.8. Charakterystyka czasu ładowania i rozładowania dla rezystancji $R = 741,2 \text{ k}\Omega$.



Rys. 3.9. Charakterystyka czasu ładowania i rozładowania dla pojemności $C = 177,2 \text{ pF}$.

Wyznaczona charakterystyka czasu ładowania i rozładowania pojemności badanej $C = 177,2 \text{ pF}$ przy zmianie rezystancji rezystora pomiarowego w zakresie od $100 \text{ k}\Omega$ do $1 \text{ M}\Omega$ (rys. 3.9.) jest w przybliżeniu liniowa [4]. Przesunięcie charakterystyki wywołane przez uwzględnienie elementów pasożytniczych w układzie pomiarowym wprowadza nierównomierne przesunięcie charakterystyki układu idealnego. Oznacza to, że dla każdej wybranej rezystancji będą wymagane inne współczynniki korekcji tej charakterystyki. Dodatkową cechą jaką zaobserwowano podczas symulacji jest „ugięcie” charakterystyki ładowania w zakresie niskich wartości rezystancji tj. mniejszych od $200 \text{ k}\Omega$. Różnice te ściśle są powiązane z opóźnieniem wprowadzanym przez obsługę przerwania. Czym mniejsza rezystancja, tym w danym czasie badany element zgromadzi większy ładunek powodując, że po przełączeniu sygnału pobudzającego większy ładunek musi zostać oddany. Dlatego zaleca się wybór rezystancji większych bądź równych od $200 \text{ k}\Omega$. Czym rezystancja jest większa, tym wynik pomiarowy jest dokładniejszy. Jednak zbyt duża wartość rezystancji prowadzi do zjawiska, w którym ograniczony tą rezystancją prąd zdoła jedynie nasycić prądy polaryzujące obwodów wejściowych komparatorów analogowych. Wartość maksymalnej rezystancji, zapewniającej poprawność pomiaru została określona symulacyjnie na około $3 \text{ M}\Omega$. Należy jednak pamiętać, że w układzie rzeczywistym zachodzą również inne zjawiska, nie wzięte pod uwagę podczas symulacji.

Przykładowa charakterystyka obrazująca metodę pomiaru z uwzględnieniem elementów pasożytniczych pokazano na rys. 3.10. W układzie rzeczywistym zapewniono możliwość podejrzenia przebiegów przy pomocy oscyloskopu. Pomiar sondą oscyloskopową nie oddaje jednak rzeczywistych warunków panujących podczas pomiaru, gdyż sonda wnosi pewną dodatkową pojemność pasożytniczą w układ pomiarowy. Przekroczenie jednego z progów napięć referencyjnych powoduje wygenerowanie dodatniego impulsu sygnału S_THR wykorzystanego do wyzwalania pomiaru czasu przez licznik mikrokontrolera.



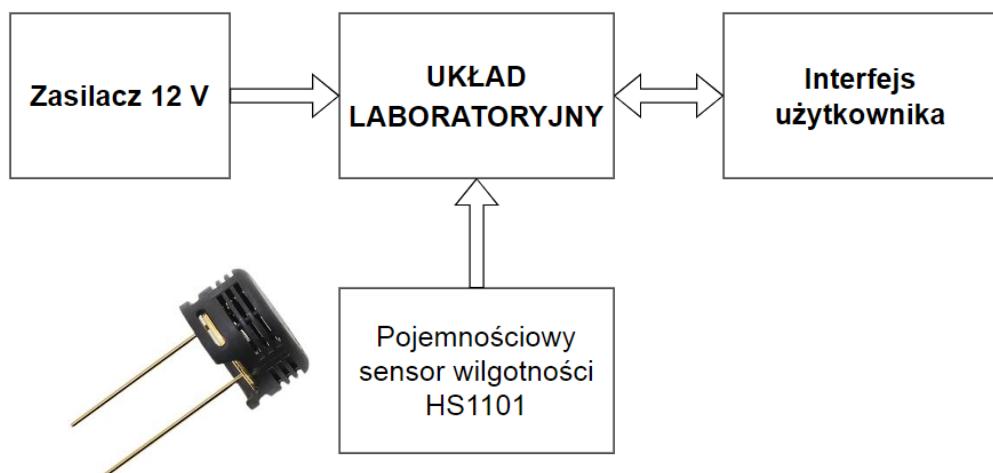
Rys. 3.10. Charakterystyka sygnałów w czasie pomiaru, $C = 177,2 \text{ pF}$, $R = 741,2 \text{ k}\Omega$.

4. BUDOWA UKŁADU LABORATORYJNEGO

W rozdziale zostanie zaprezentowana fizyczną realizacja układu inteligentnego czujnika pojemnościowego. Szczególną uwagę poświęcono omówieniu podukładu zasilania obwodu pomiarowego, a także modułów Arduino Micro, czujnika temperatury i wilgotności względnej SHTC3 oraz wyświetlacza alfanumerycznego LCD.

4.1. Stanowisko laboratoryjne

Dokonanie pomiaru pojemności układem laboratoryjnym jest możliwe tylko z wykorzystaniem interfejsu użytkownika (rys. 4.1.). Interfejs ten pozwala na wykonanie pomiaru jednokrotnego lub wielokrotnego z zaprogramowanym odstępem czasu, pojemności czujnika HS1101 [8]. Oprócz przeprowadzania pomiarów udostępnia on funkcję kalibracji układu pomiarowego. Podczas kalibracji możliwy jest pomiar poziomu napięcia na wyjściach bufora trójstanowego. Interfejs użytkownika realizuje także obliczenia oraz generuje skrypt dla programu Matlab, który można wykorzystać do dalszej analizy wyników pomiarowych. Interfejs użytkownika służy również do ustawiania parametrów układu laboratoryjnego takich jak: poziomy napięć progowych i napięć stanów na wyjściu bufora trójstanowego, czy współczynniki korekcji. Wartości tych nastaw są zapamiętywane w pamięci EEPROM mikrokontrolera, a ich aktualizacja jest możliwa za pomocą tego interfejsu. Po podłączeniu układu laboratoryjnego do komputera PC za pośrednictwem interfejsu USB dokonywana jest automatyczna detekcja układu i nawiązanie połączenia.



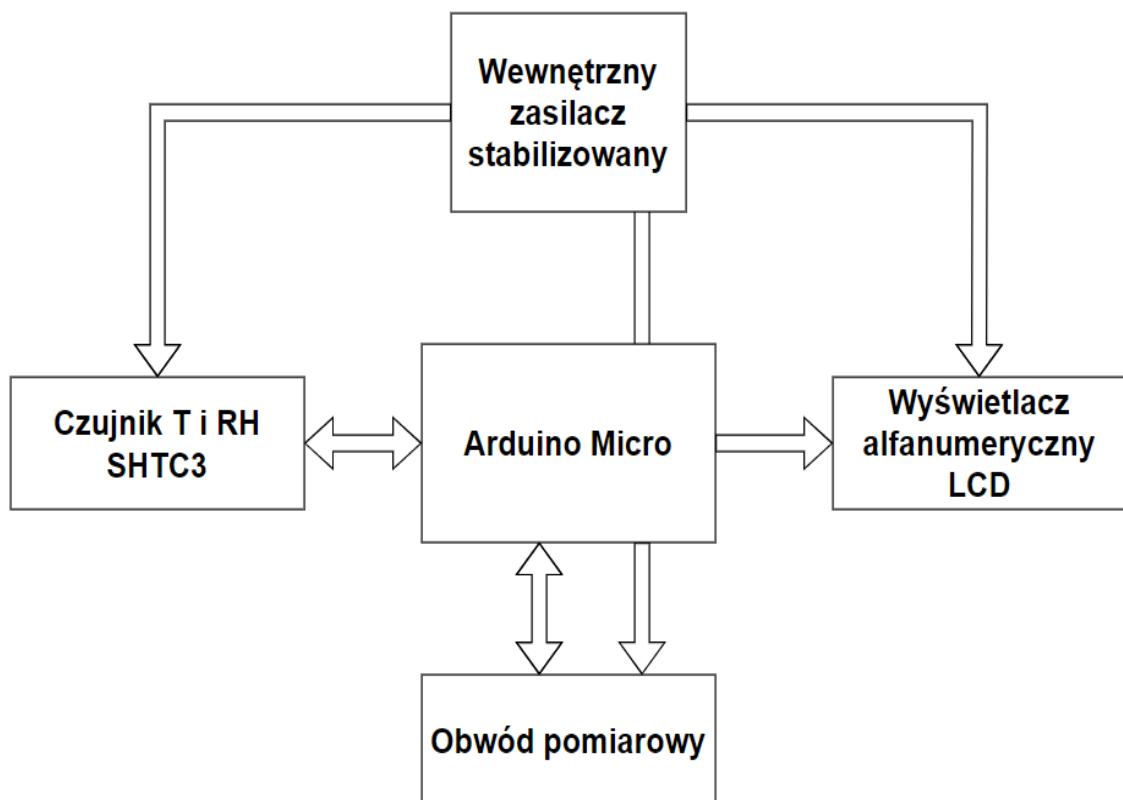
Rys. 4.1. Schemat blokowy stanowiska pomiarowego.

Do poprawnego działania układu laboratoryjnego wymagane jest zewnętrzne napięcie zasilające o wartości przynajmniej 8 V. Zaś obiektem badanym jest pojemościowy czujnik wilgotności względnej HS1101, którego pojemność w zależności od wartości wilgotności względnej, zmienia się w zakresie 161 – 193 pF. Układ laboratoryjny został przebadany i skalibrowany dla szerszego zakresu badanych pojemności.

4.2. Inteligentny czujnik pojemnościowy

Głównym blokiem sterującym układem laboratoryjnym jest moduł Arduino Micro bezpośrednio połączony z obwodem pomiarowym (rys. 4.2.). Aby zapewnić stabilny i dokładny pomiar zastosowano w układzie wewnętrzny zasilacz stabilizowany. W celu ograniczenia pobieranego prądu z wewnętrznego zasilacza zdecydowano, że moduł Arduino Micro zostanie zasilony z zewnętrznego źródła napięcia. Możliwe to jest, gdyż wspomniany moduł posiada wbudowany stabilizator monolityczny.

Kolejnym blokiem inteligentnego czujnika pojemnościowego jest czujnik temperatury i wilgotności względnej SHTC3. Układ ten zapewnia dodatkowe funkcje, tj. przybliżone sprawdzenie poprawności wyznaczonej wilgotności z czujnika pojemnościowego HS1101. Oba czujniki RH zostały umiejscowione możliwie blisko siebie, aby pracowały w tym samym środowisku (otoczeniu). Dodatkowe dane o temperaturze służą do uzyskania informacji o punkcie rosy, czyli o temperaturze, w której rozpocznie się proces skraplania pary wodnej, co znacząco może wpływać na wskazania czujnika pojemnościowego HS1101. Układ laboratoryjny dodatkowo został wyposażony w układ ekspozycji danych, czyli wyświetlacz alfanumeryczny LCD.



Rys. 4.2. Schemat blokowy układu laboratoryjnego.

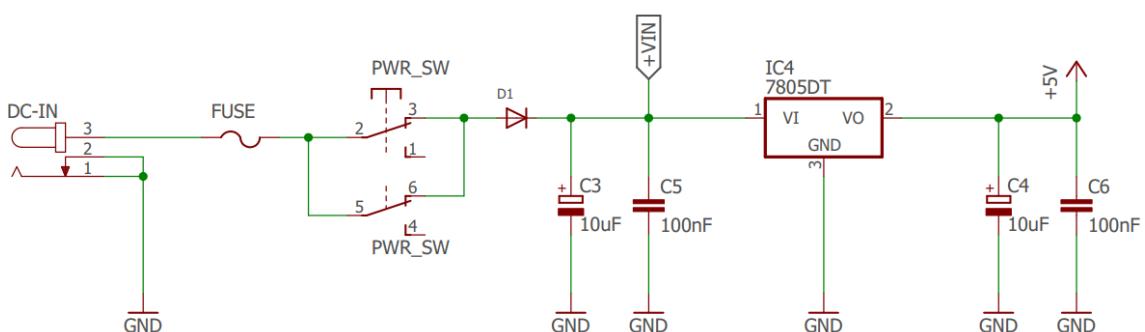
4.3. *Układ zasilania*

Stabilność napięcia zasilającego układ pomiarowy z punktu widzenia dokładności pomiarowej jest jednym z kluczowych czynników. Napięcie zasilania wpływa na poziom napięcia wyjściowego z bufora trójstanowego, które jest sygnałem wejściowym dla bloku pomiarowego. Stabilność napięcia w czasie pomiaru jest nieodzowna dla zapewnienia wysokiej dokładności pomiarowej, gdyż jednym z parametrów we wzorze (3.4.) jest napięcie sygnału pobudzającego blok pomiarowy. Wyklucza to użycie zasilania z portu USB ze względu na niestabilność napięcia, którego wartość może zawierać się od 4,5 do 5,5 V [9].

Zastosowany w projekcie stabilizator monolityczny 7805 (rys. 4.3.) zapewnia należytą stabilność napięcia w układzie pomiarowym. Współczynnik temperaturowy napięcia wyjściowego wynosi zaledwie $-1,1 \text{ mV}^{\circ}\text{C}$ [10]. Czas pomiaru pojemności jest wielokrotnie krótszy od bezwładności temperaturowej otoczenia, stąd zmiany temperatury nie wpłyną na stabilizowane napięcie w trakcie pomiaru.

Kolejnym czynnikiem wpływającym na stabilizację napięcia jest wartość pobieranego prądu (typowo 5 mV/A [10]). W trakcie pomiaru wartość prądu ulega zmianom w trakcie ładowania i rozładowania pojemności badanej. Pozostałe układy jak SHTC3 oraz LCD w tym czasie pozostają w spoczynku. Poziom wartości prądu jest zależny od rezystancji pomiarowej, która powinna zawierać się w zakresie od $100\text{ k}\Omega$ do $1\text{ M}\Omega$. Zatem w trakcie pełnego rozładowania pojemności prąd ten jest na tyle mały, że nie wpływa na stabilizowane napięcie.

Układ zasilania został wyposażony w diodę prostowniczą zabezpieczającą układ laboratoryjny przed podłączeniem zewnętrznego napięcia zasilającego o odwrotnej polaryzacji. Stabilizator napięcia do poprawnej pracy potrzebuje napięcie o 2 V większą z zewnętrznego napięcia zasilania w stosunku do napięcia stabilizowanego, wliczając w to dodatkowo potencjał odkładający się na diodzie prostowniczej. Układ laboratoryjny należy zasilać napięciem minimalnym 8 V.



Rys. 4.3. Schemat ideowy układu zasilającego.

4.4. Moduł Arduino Micro

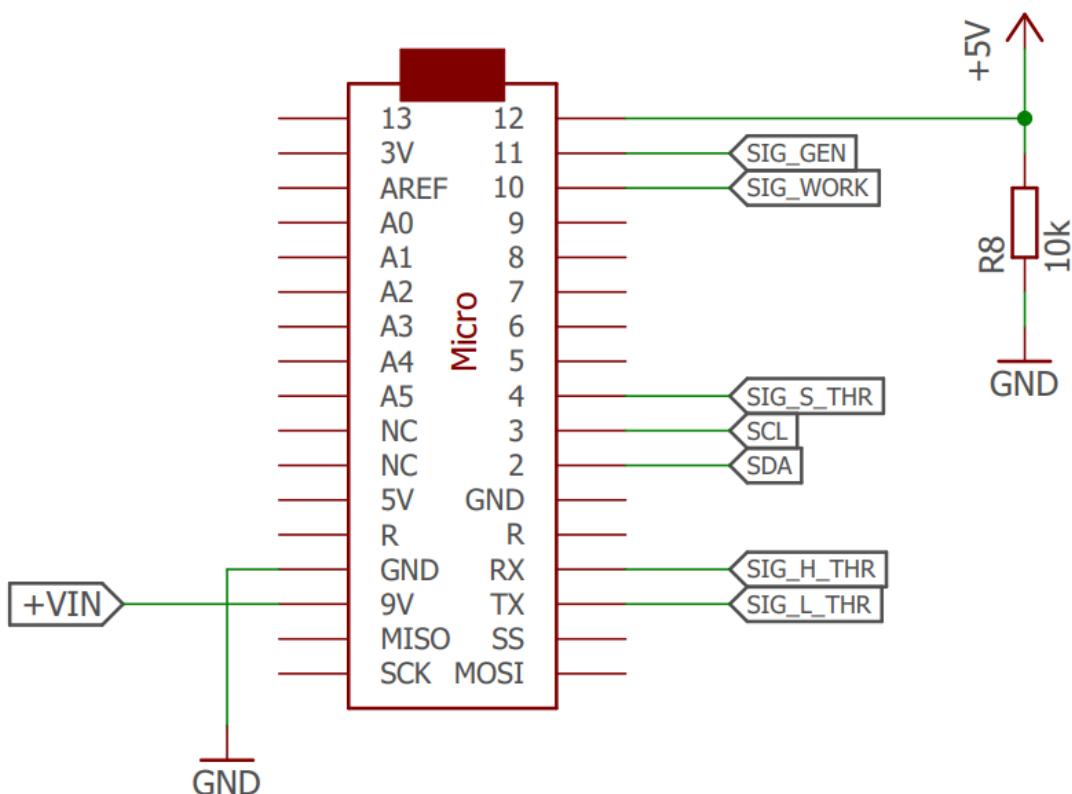
Zastosowany moduł Arduino Micro (rys. 4.4.) bazuje na 8-bitowym mikrokontrolerze ATmega32U4. Posiada wbudowane złącze Micro USB oraz rezonator kwarcowy o częstotliwości 16 MHz [11]. Na płytce PCB układu znajduje się także złącze do programowania szeregowego ISP oraz 3 diody LED, które w projekcie służą do sygnalizacji wykonania następujących czynności przez układ laboratoryjny: podłączenie do komputera nadzawanego, a także skomunikowanie z interfejsem użytkownika. Moduł posiada listwy kolkowe z wyprowadzeniami linii mikrokontrolera po obu stronach PCB. Moduł wyposażony został w stabilizator napięcia zasilania 5 V, o niewielkim poborze mocy. Mikrokontroler, przy zegarze 16 MHz, wymaga zasilania o wartości 5 V. Wewnątrz modułu znajduje się także drugi stabilizator monolityczny o napięciu wyjściowym 3,3 V.



Rys. 4.4. Moduł Arduino Micro [12].

Blok Arduino Micro (rys. 4.5.) podłączono z wyświetlaczem alfanumerycznym LCD oraz z czujnikiem temperatury i wilgotności względnej SHTC3 za pomocą interfejsu TWI [9]. Sygnały wyjściowe sterujące obwodem pomiarowym to m. in.: sygnał aktywacji buforów trójstanowych (SIG_WORK) oraz sygnał pobudzający układ pomiarowy RC (SIG_GEN). Pozostałe sygnały, tj. przekroczenie progów napięć przez napięcie na pojemności badanej (SIG_L_THR, SIG_H_THR) oraz suma tych zdarzeń (SIG_S_THR), są podłączone do wejść mikrokontrolera poprzez bufor trójstanowy. Tak jak wspomniano, wykorzystano wewnętrzny stabilizator modułu Arduino Micro do zasilania mikrokontrolera, w celu obniżenia prądu pobieranego ze stabilizatora 7805 zasilającego pozostałe bloki układu laboratoryjnego. W tym celu napięcie zewnętrzne (VIN) zostało podłączone do dedykowanego wejścia.

Dodatkowo, aby programowo sprawdzić obecność zewnętrznego napięcia zasilania np. pochodzącego z interfejsu USB, zastosowano rezystor R8. Podczas podłączenia Arduino Micro do portu USB, czerpie on z niego zasilanie. Brak napięcia zewnętrznego powoduje, że wejście nr 12 mikrokontrolera pozostaje niepodłączone (nie panuje na nim żadne napięcie), dlatego wejście to musi zostać podciągnięte do masy układu poprzez rezystor R8. Odczyt logicznego „0” z linii 12 mikrokontrolera świadczy o braku zewnętrznego napięcia zasilającego. Generowany jest wtedy stosowny komunikat do interfejsu użytkownika.

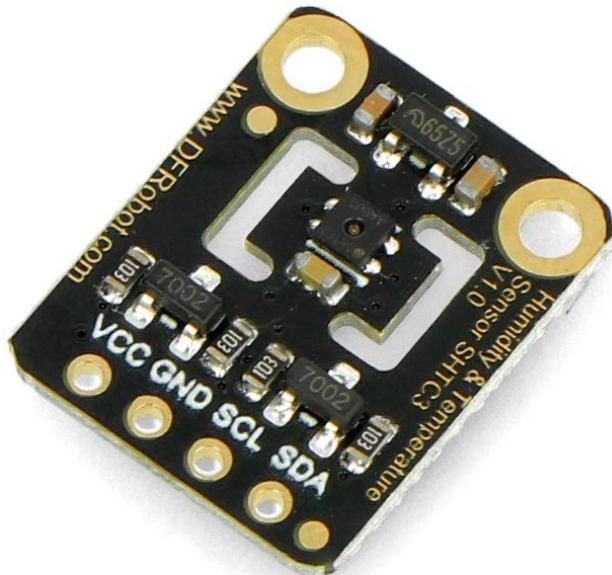


Rys. 4.5. Schemat ideowy podłączenia modułu Arduino Micro.

4.5. Moduł SHTC3

Moduł SHTC3 służący do pomiaru temperatury i wilgotności względnej użyto w celu potwierdzenia poprawności wyznaczonej wartości wilgotności względnej zmierzanej za pomocą czujnika pojemnościowego HS1101. Należy pamiętać, że jego typowa dokładność wyznaczania wilgotności względnej wynosi $\pm 2\%$ RH [13]. Dlatego nie należy go traktować jako elementu wzorcowego. Dodatkowo pomiar temperatury pozwala na uzyskanie informacji o temperaturze punktu rosy, który określa początek skraplania się pary wodnej, a tym samym przekroczenie zakresu pomiarowego czujnika wilgotności. Pomiar temperatury i wilgotności względnej trwa mniej niż 20 ms.

Moduł SHTC3 składa się z czujnika pracującego przy napięciu 3,3 V. Na płytce PCB (rys. 4.6.) zaimplementowano konwertery stanów logicznych z 5 V na poziom 3,3 V oraz monolityczny stabilizator napięcia zasilający czujnik. Umożliwia to bezpośrednie podłączenie modułu do układów pracujących przy napięciu 5 V [13].



Rys. 4.6. Moduł SHTC3 [12].

4.6. Moduł LCD

Zadaniem wyświetlacza alfanumerycznego LCD (rys. 4.7.) jest wyłącznie prezentacja danych pomiarowych. Stąd obecność wyświetlacza nie jest konieczna do prawidłowego działania urządzenia. Układ steruje się poprzez interfejs TWI, dzięki czemu mikrokontroler automatycznie wykrywa jego obecność podczas inicjalizacji systemu. Sterowanie wyświetlaczem jest realizowane w pętli głównej programu mikrokontrolera. Zastosowanie ekspandera wyjść PCF8574t znaczaco ogranicza liczbę potrzebnych do sterowania wyprowadzeń mikrokontrolera, co jest zaletą, ponieważ ze względu na wymiary płytki PCB zdecydowano żeby wyświetlacz był dołączany do układu laboratoryjnego za pomocą 4 przewodów, z czego 2 służą jako zasilanie modułu, a pozostałe 2 zapewniają komunikację z ekspanderem wyjść. Dedykowany ekspander do sterowania wyświetlaczami LCD posiada wbudowany potencjometr służący do regulacji kontrastu [12]. Linie interfejsu TWI wymagają podłączenia do nich rezystorów podciągających do zasilania, ponieważ wyjścia tego interfejsu są typu otwarty kolektor. W tym celu użyto w układzie laboratoryjnym rezystorów R6 oraz R7 o wartości rezystancji 10 kΩ.

Po każdym pomiarze interfejs użytkownika wysyła do układu laboratoryjnego obliczone wartości pojemności, wilgotności względnej i temperatury. Poza tym wysyłany jest średni czas ładowania i rozładowania badanej pojemności czujnika. Ten parametr jest wyświetlany również na wyświetlaczu LCD.

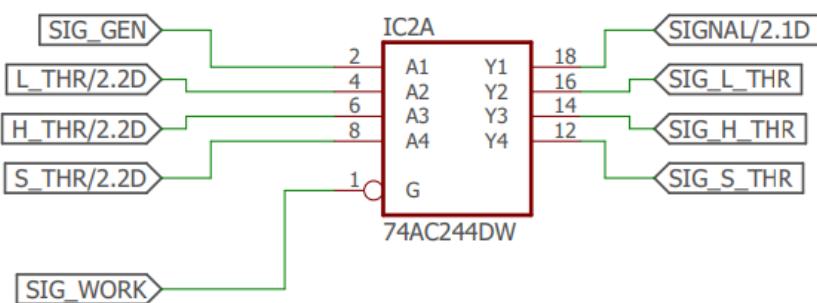


Rys. 4.7. Moduł wyświetlacza alfanumerycznego LCD [12].

4.7. Układ pomiarowy

Blok pomiarowy składa się z bufora trójstanowego, układu RC, toru przetwarzania sygnału pomiarowego oraz toru napięć referencyjnych. Jednym z ważniejszych parametrów tych układów są stabilne napięcia odniesienia, możliwe najwyższe napięcie niezrównoważenia komparatorów oraz niewielki prąd polaryzacji ich obwodów wejściowych.

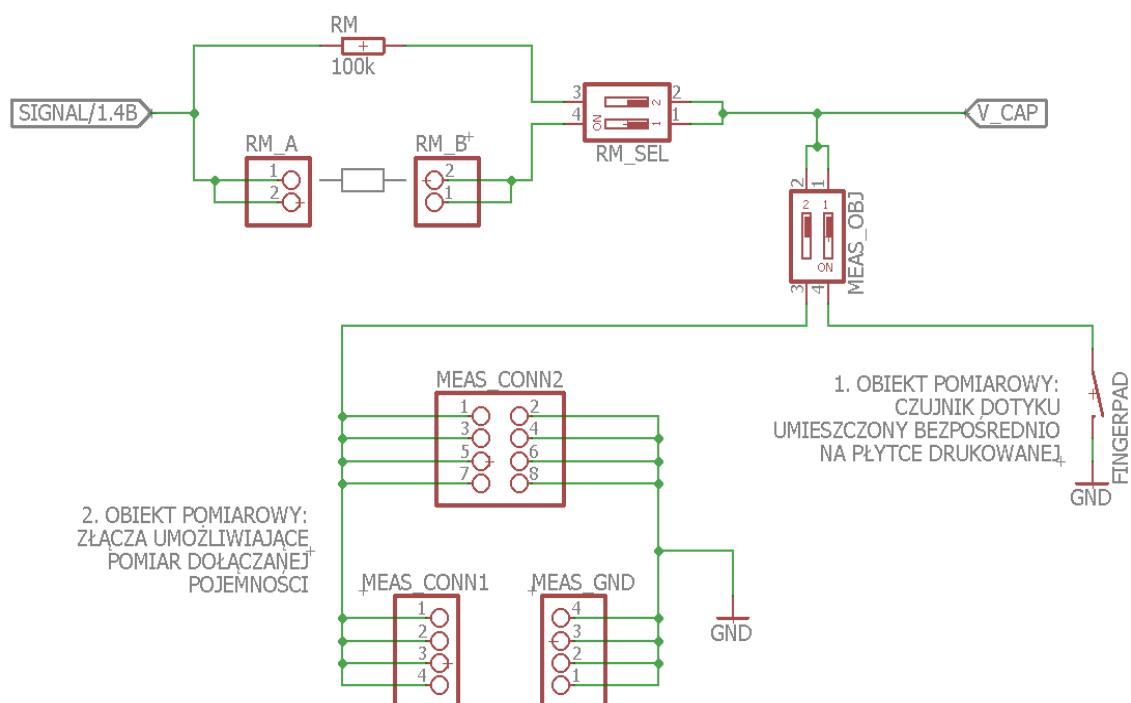
Bufor trójstanowy izoluje układ wykonujący pomiar od mikrokontrolera (rys. 4.8.), gdyż wyjście mikrokontrolera posiada większe pojemności pasożytnicze. Dodatkowo zapewnia on zwiększoną stabilność napięcia wyjściowego oraz większy prąd uzyskiwany z linii wyjściowych. Kolejną zaletą tych układów jest możliwość ustawienia linii w stan wysokiej impedancji, co znacząco zmniejsza pobór prądu w stanie uśpienia układu [14].



Rys. 4.8. Schemat ideowy bufora trójstanowego.

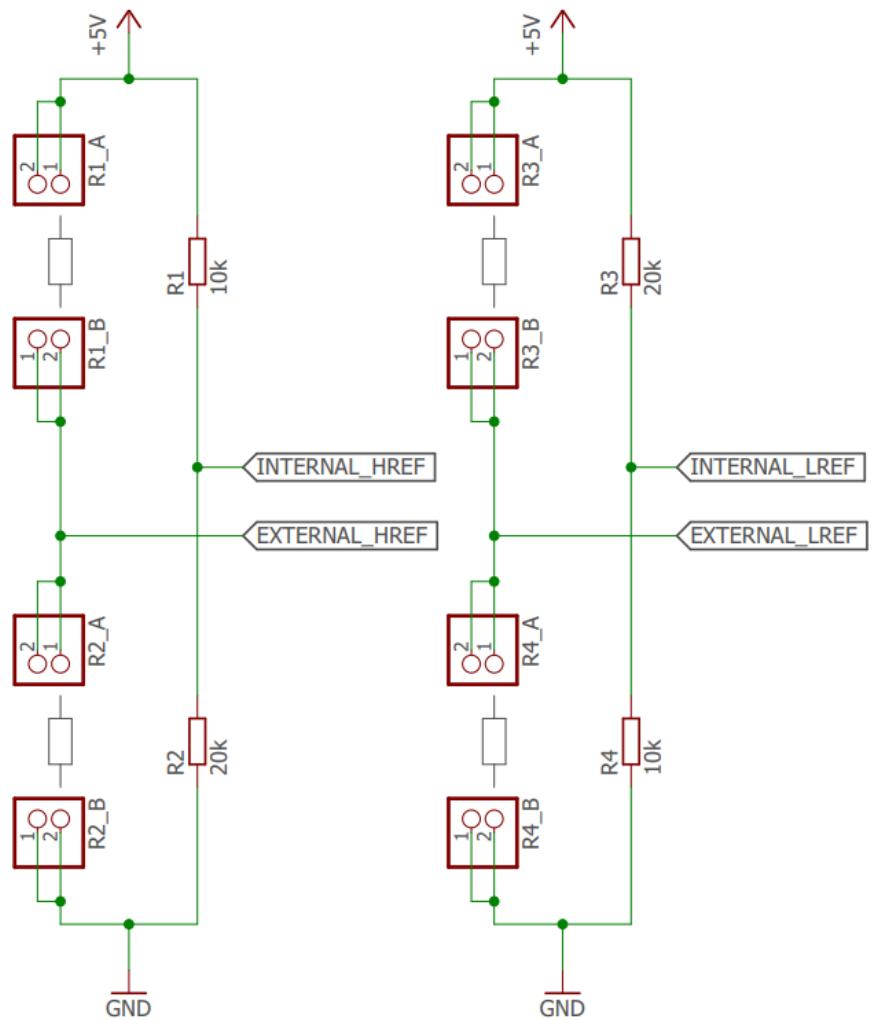
Układ pomiarowy RC (rys. 4.9.) składa się z rezystora pomiarowego RM oraz dołączanej pojemności badanej. Dodatkowo, na płytce PCB umieszczono czujnik dotyku, którego działanie opiera się na zmianie pojemności pod wpływem przyłożonego palca. Pojemność rośnie wraz ze wzrostem powierzchni styku czujnika z naskórkiem. Na pojemność mają również wpływ parametry skóry, np. jej wilgotność. Czujnik podłączono do listwy kołkowej. Do wyboru obiektu pomiarowego służą przełączniki DIPSWITCH.

Wartość rezystancji rezystora pomiarowego jest jednym z parametrów we wzorach obliczeniowych pojemności, dlatego należy zmierzyć tą wartość z możliwie największą dokładnością. Układ laboratoryjny umożliwia podłączenie wybranego przez użytkownika rezystora za pośrednictwem listwy kołkowej. Wartość rezystancji wbudowanej w układ pomiarowy wynosi $100\text{ k}\Omega$. Wybór wbudowanego lub zewnętrznego rezystora pomiarowego możliwy jest przez zmianę położenia przełączników DIPSWITCH.



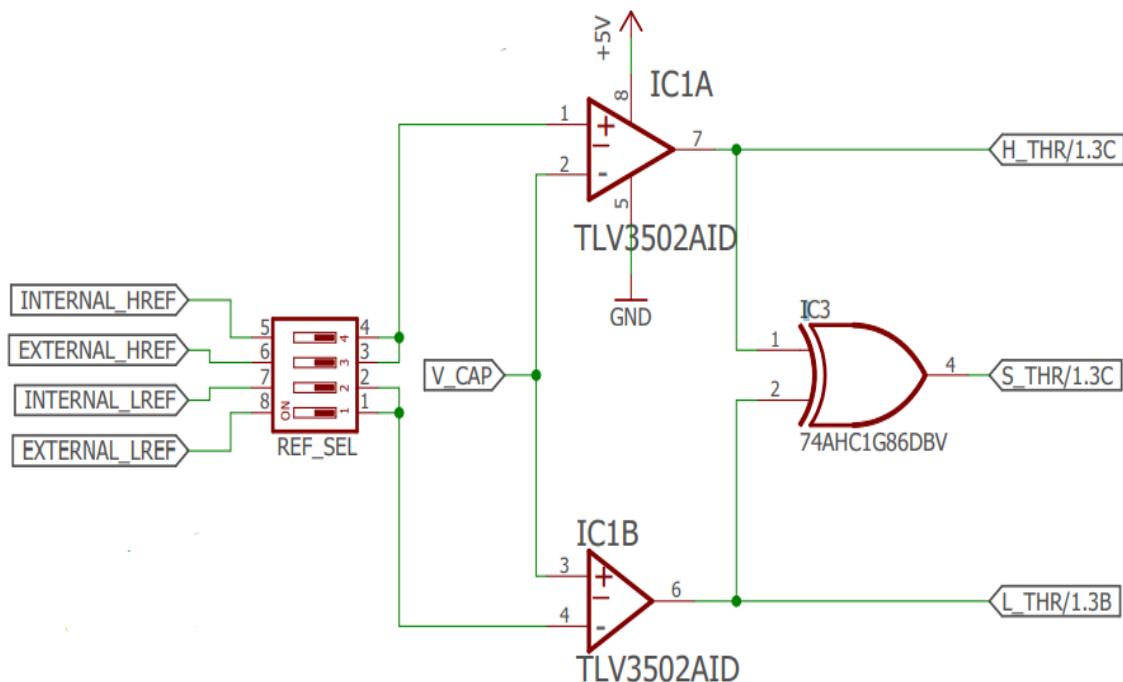
Rys. 4.9. Schemat ideowy układu pomiarowego RC.

Wartości napięć odniesienia ustawia się za pomocą precyzyjnych dzielników rezystancyjnych (rys. 4.10.). W układzie istnieje możliwość wyboru dzielników wbudowanych ($R_1, R_2; R_3, R_4$) lub zewnętrznych – konfigurowanych przez użytkownika. Do tego celu służą żelazkie listwy kołkowe. Wbudowane dzielniki zapewniają $2/3$ wartości napięcia zasilania (w przypadku górnego progu) oraz $1/3$ wartości napięcia zasilania (w przypadku dolnego progu). Wybór danego potencjału odniesienia jest dokonywany poprzez przełączniki DIPSWITCH.



Rys. 4.10. Schemat ideowy toru napięć referencyjnych.

W skład toru przetwarzania sygnału pomiarowego (rys. 4.11.) wchodzą dwa komparatory. Dla zapewnienia dużej dokładności pomiarowej, należy dobrać komparator analogowy o możliwie najmniejszym prądzie polaryzacji jego obwodów wejściowych. Wybrany został układ TLV3502AID, którego wyżej wymieniony parametr jest wystarczający do zapewnienia dużej dokładności pomiarowej. Prąd polaryzacji zgodnie z dokumentacją ma wartość ± 20 pA [6]. Ważnym parametrem jest także napięcie niezrównoważenia. Wartość ta wynosi 1 – 6,5 mV [6]. Natomiast czas propagacji sygnału wyjściowego tego układu wynosi 12 ns [6] i jest kilkukrotnie mniejsza od czasu trwania jednego cyku zegarowego mikrokontrolera, która dla zegara 16 MHz wynosi 62,5 ns. Zatem czas propagacji nie wpływa na opóźnienie odczytu sygnałów przez mikrokontroler. Wykorzystanie modułu przechwytywania zdarzeń zewnętrznych licznika mikrokontrolera wymaga, aby oba sygnały progowe były sprowadzone do jednej linii. Do tego służy bramka logiczna XOR, która dokonuje sumy modulo obu sygnałów.



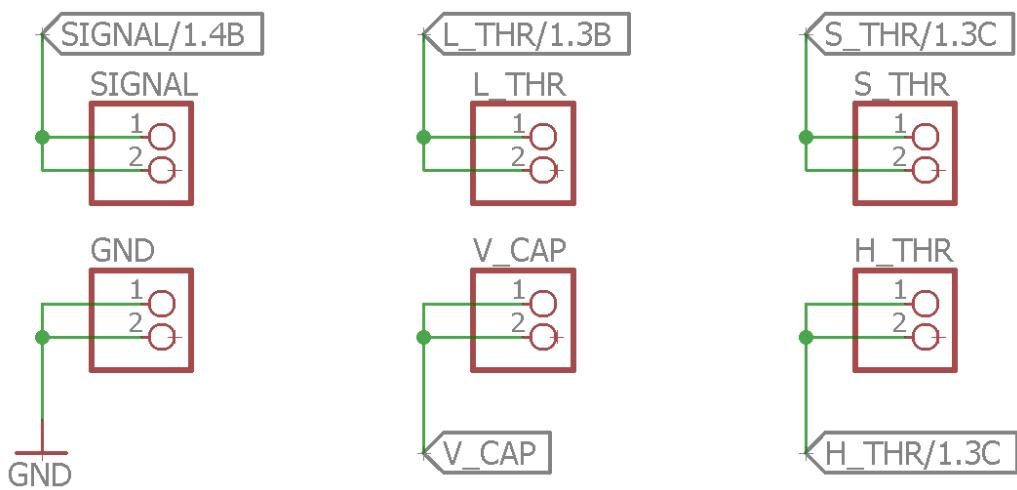
Rys. 4.11. Schemat ideowy toru przetwarzania sygnału pomiarowego.

4.8. Złącza diagnostyczne

Możliwość pomiaru sygnałów w układzie laboratoryjnym jest istotną cechą układów dydaktycznych, gdyż pozwala użytkownikowi na pełną analizę układu badanego. W tym celu zastosowano złącza diagnostyczne w postaci listew kołkowych, które umożliwiają podłączenie do układu oscyloskopu. Każda sonda oscyloskopowa wnosi pewną niewielką pojemność do układu, więc samo podłączenie sondy wywołuje zmianę badanej pojemności. Stąd, aby zminimalizować wpływ sondy należy przestawić ją w tryb pracy „x10”, ponieważ ma ona w tym trybie mniejszą pojemność.

Złącza diagnostyczne umożliwiają podłączenie oscyloskopu i obejrzenie na nim przebiegu sygnału ładowania i rozładowania pojemności. Wyprowadzony został również na złącze sygnał pobudzający blok pomiarowy. Podczas kalibracji urządzenia mikrokontroler na tym wyprowadzeniu wystawia sygnał prostokątny o częstotliwości 1 MHz. Umożliwia to doświadczalne zmierzenie opóźnień działania komparatora oraz poziomów napięć wyjściowych bufora trójstanowego.

Wyprowadzone zostały również sygnały osiągnięcia progu (L_THR – osiągnięty dolny próg, H_THR – osiągnięty górny próg), a także suma modulo powyższych zdarzeń (S_THR). Dzięki obserwacji tych sygnałów możliwe jest prześledzenie przebiegu czasowego procedury pomiarowej.



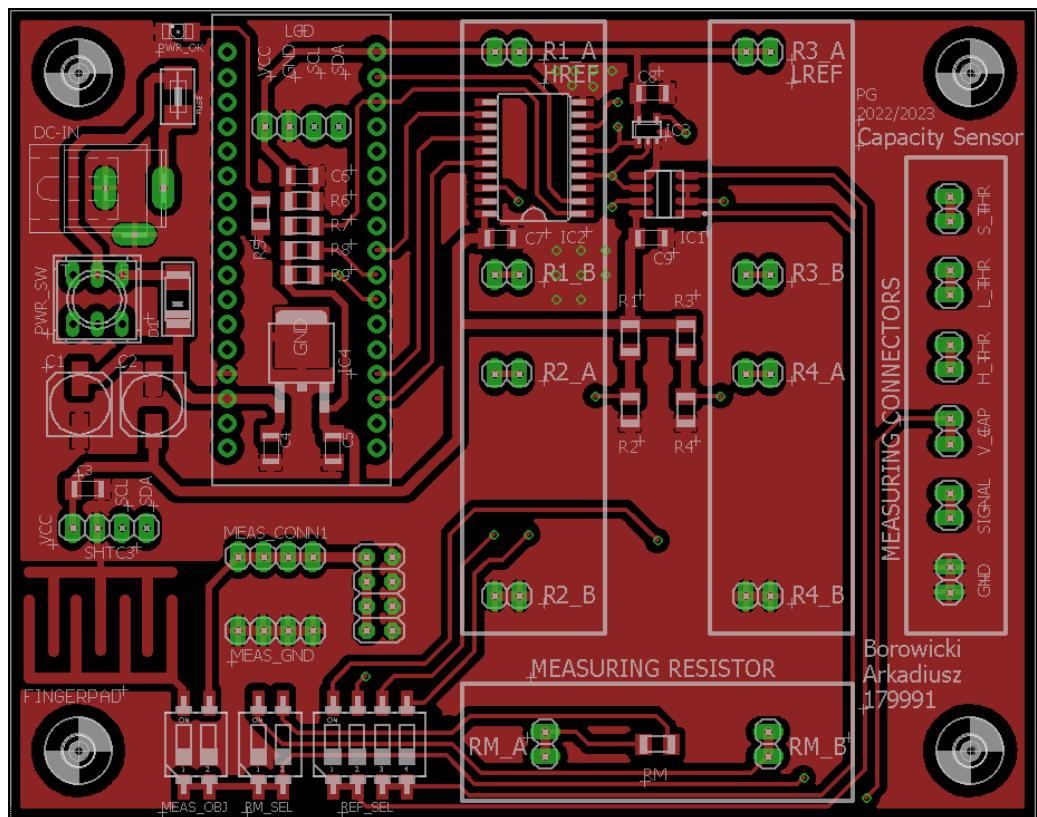
Rys. 4.12. Schemat ideowy złącz diagnostycznych.

4.9. Layout PCB

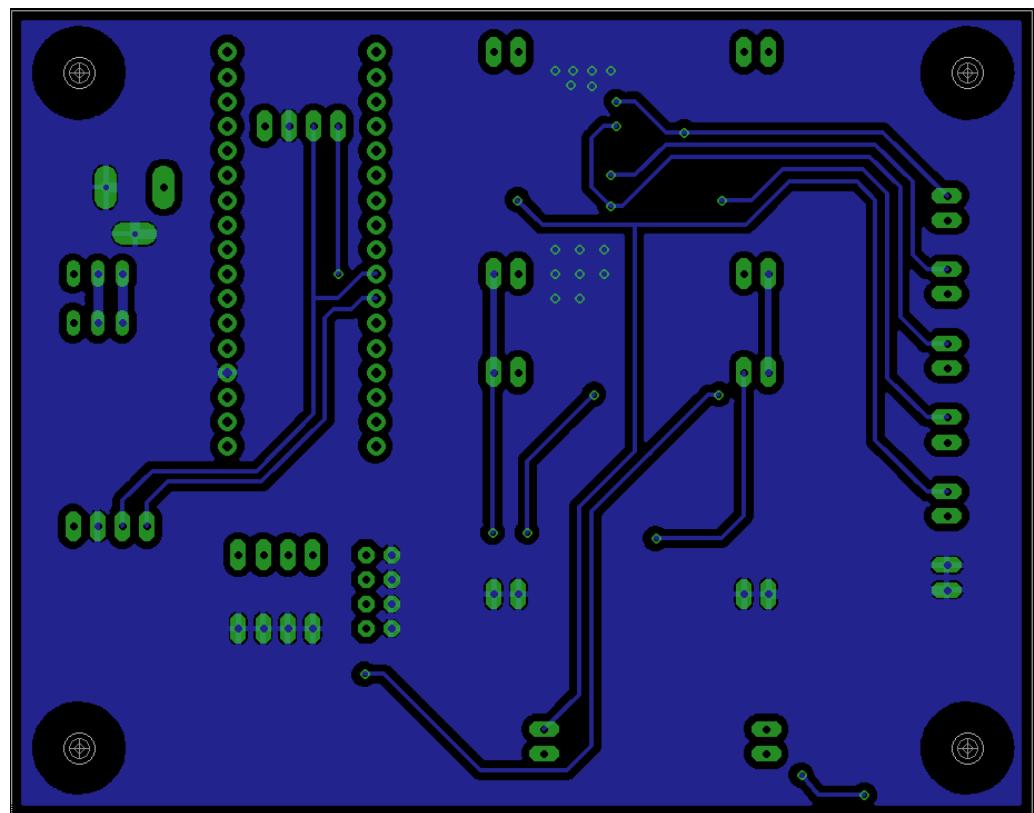
W podrozdziale zostanie zaprezentowany projekt płytki PCB oraz montaż rzeczywistego układu. Szczególną uwagę poświęcono sposobom prowadzenia ścieżek oraz umiejscowienia elementów elektronicznych.

4.9.1. Projekt

Projekt płytki PCB został opracowany przy użyciu programu Autodesk Eagle (rys. 4.13.; rys. 4.14.). Wynikowa płytka posiada wymiary 105 x 83 mm. Projekt wykonano w formie 2 stronnej płytki o szerokości ścieżek sygnałowych na poziomie 16 milsów oraz ścieżek zasilających 40 milsów. Szerokości ścieżek są wystarczające do zapewnienia minimalnej rezystancji oraz wystarczające do przenoszenia prądów przepływających przez układ laboratoryjny. Ponadto zastosowano rozlew masy po obu stronach płytki, w celu ograniczenia szumów mogących wpływać na dokładność pomiaru pojemności. Rozlew masy ma za zadanie ekranowanie ścieżek sygnałowych. Wadą tego rozwiązania jest zwiększoną pojemność ścieżek w stosunku do masy. Warstwa miedzi została pokryta soldemaską, której zadaniem jest zabezpieczenie pól kontaktowych przed zwarciami oraz zanieczyszczeniami. Aby ograniczyć liczbę otworów na płytce, zdecydowano się na montaż powierzchniowy SMD. Z punktu widzenia użytkownika, ważne jest także właściwe umiejscowienie złącz diagnostycznych, aby w łatwy sposób korzystać z urządzenia. Wejście zasilania zostało umiejscowione w lewym górnym rogu płytki, obok przełącznika zasilania. Złącza diagnostyczne zostały umiejscowione z prawej strony płytki PCB. Wszystkie elementy bloków pomiarowych starano się umieścić możliwie blisko siebie. Płytkę stoi na nóżkach, aby nie było styku płytki z podłożem, na którym stoi.



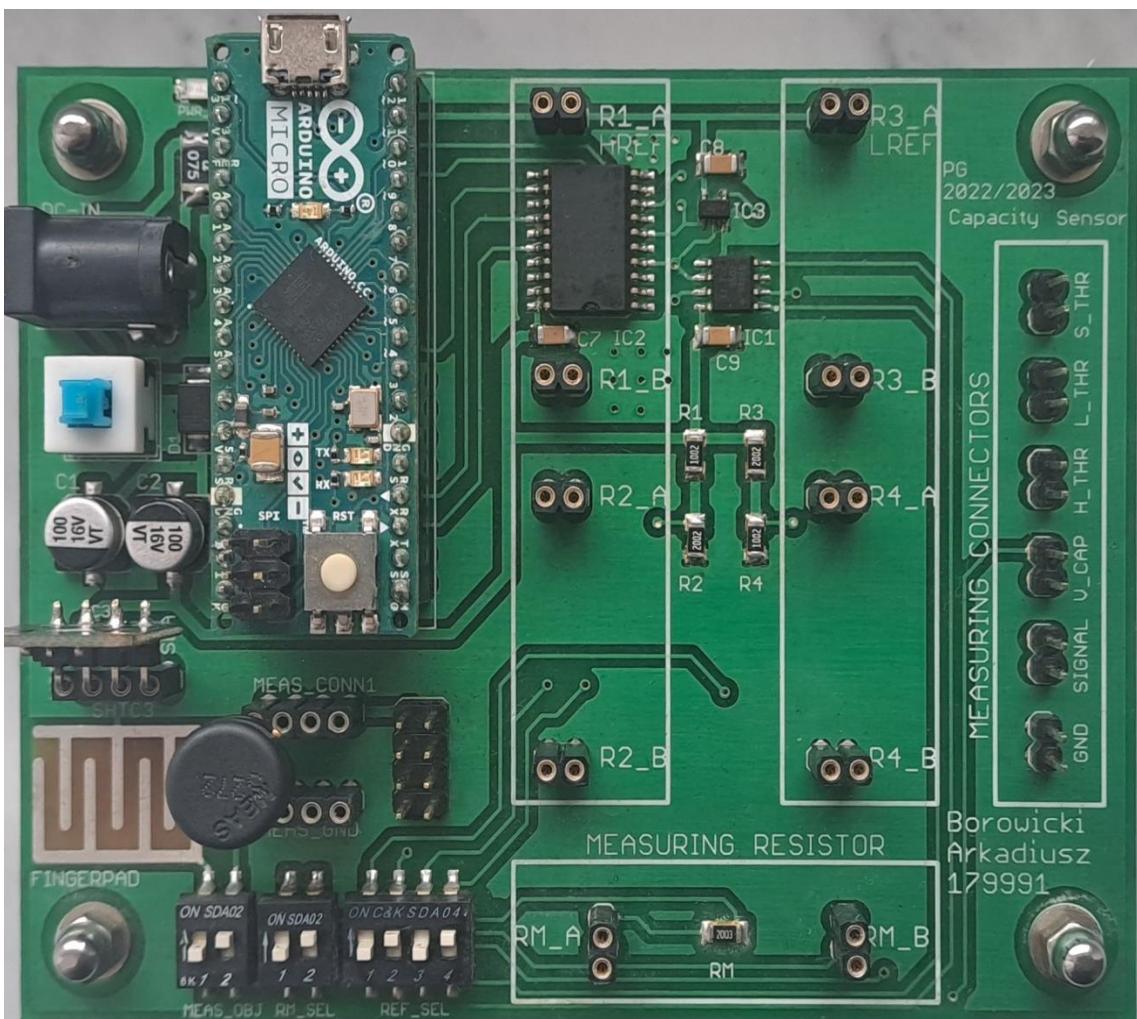
Rys. 4.13. Projekt mozaiki PCB – góra strona.



Rys. 4.14. Projekt mozaiki PCB – dolna strona.

4.9.2. Realizacja

Rzeczywisty układ laboratoryjny (rys. 4.15.) posiada złącza na moduł Arduino Micro, czujnik temperatury i wilgotności względnej SHTC3 oraz wyświetlacz LCD. Moduł wyświetlacza LCD został dołączony za pomocą przewodów do kątowego złącza kołkowego znajdującego się pod modułem Arduino Micro. Układ laboratoryjny posiada także złącze umożliwiające dołączenie czujnika pojemnościowego, co czyni ten układ uniwersalnym mogącym pracować z różnymi czujnikami. Na rynku są dostępne także inne moduły służące do pomiaru wilgotności, np. czujnik pary wodnej, którego działanie opiera się na badaniu pojemności, czy czujnik wilgotności gleby. Miejsce na dołączenie tych czujników zostało zrealizowane w postaci miskich listew kołkowych znajdujących się obok czujnika HS1101. Moduł SHTC3 umiejscowiono możliwie blisko czujnika pojemnościowego HS1101, tak aby pracowały w tych samych warunkach. Programowanie mikrokontrolera odbywa się za pośrednictwem złącza ISP. Moduł posiada także wbudowane złącze Micro USB służące do komunikacji z interfejsem użytkownika.



Rys. 4.15. Realizacja fizyczna układu laboratoryjnego.

5. UŻYTE UKŁADY PERYFERYJNE MIKROKONTROLERA

W rozdziale zostanie zaprezentowana struktura wewnętrzna mikrokontrolera ATmega32U4. Zostaną zaprezentowane rejesty konfiguracyjne oraz znaczenie poszczególnych bitów. Szczególną uwagę skupiono na omówieniu zastosowanych ustawień układów peryferyjnych użytych w projekcie.

5.1. System przerwań

System przerwań jest jednym z ważniejszych układów peryferyjnych mikrokontrolerów, gdyż pozwala na obsługę pozostałych układów peryferyjnych w trybie przerwań, dzięki czemu nie ma potrzeby odpytywania tych układów w głównej pętli programu.

Każde zdarzenie pochodzące od układu peryferyjnego ustawia bit żądania obsługi przerwania. W pierwszej kolejności zostają obsłużone przerwania o najwyższym priorytecie. Czas pomiędzy wystąpieniem zdarzenia, a jego obsługą wynosi minimum 5 cykli zegara taktującego mikrokontroler [9]. W tym czasie następuje zapisanie na stosie aktualnej wartości licznika programu, którego wartość w następnym kroku zostaje ustawiona na adres obsługi danego przerwania oraz zerowana jest flaga żądania obsługi przerwania. Po zakończeniu przerwania licznik programu zostaje załadowany wartością zapisaną na stosie. Cała procedura powrotu do wcześniejszej wykonywanego programu ponownie zajmuje minimum 5 cykli zegarowych.

Aby korzystać z systemu przerwań konieczne jest ustawienie bitu I (Global Interrupt Enable) w rejestrze statusowym SREG (rys. 5.1.) [9]. Przerwania zostają obsługiwane dopiero w momencie ustawienia tego bitu na wartość „1”. Dodatkowo każdy układ peryferyjny posiada maski przerwań, które uprzednio należy ustawić, aby możliwa była obsługa przerwań pochodzących od tych układów. Każde przerwanie ma określony priorytet. Oznacza to, że w pierwszej kolejności zostanie obsłużone przerwanie o najwyższym priorytecie. Pozostałe przerwania zostaną rozpatrzone bezpośrednio po jego zakończeniu.

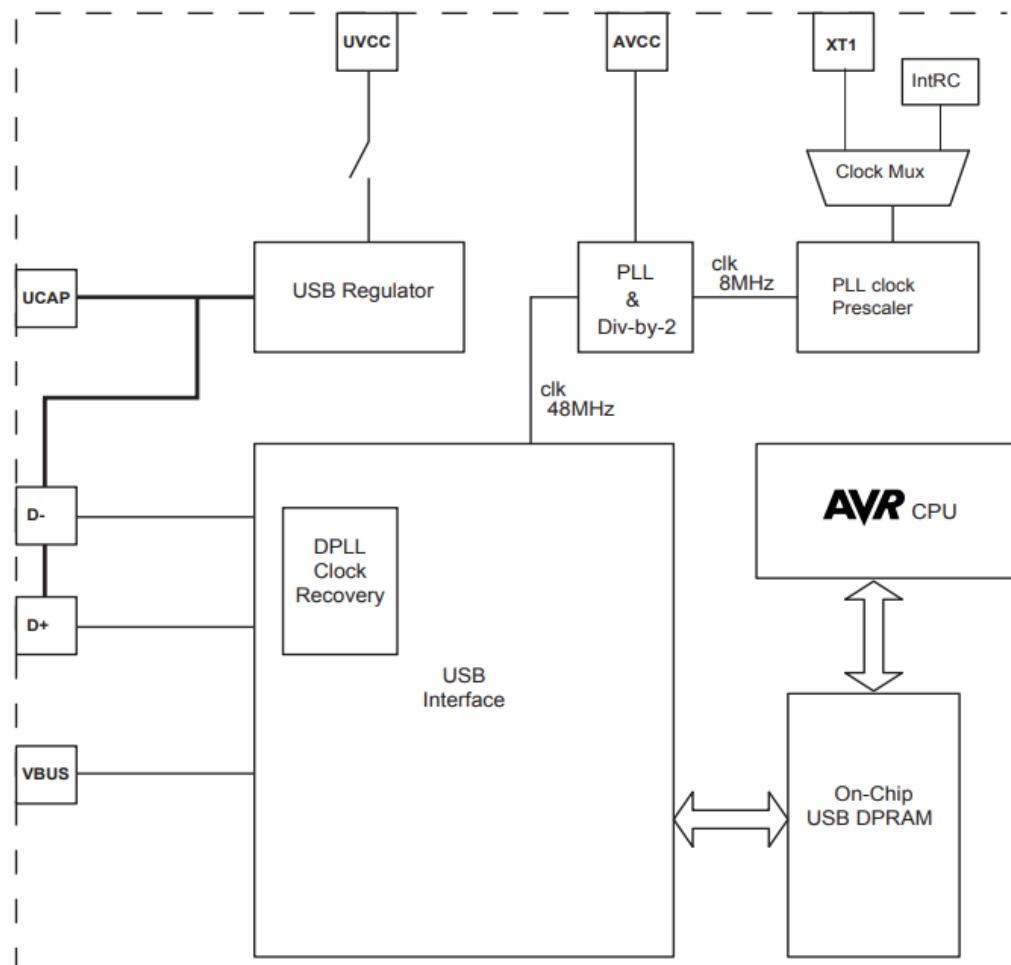
Bit	7	6	5	4	3	2	1	0	
	I	T	H	S	V	N	Z	C	SREG
Read/Write	R/W								
Initial Value	0	0	0	0	0	0	0	0	

Rys. 5.1. Rejestr statusowy mikrokontrolera ATmega32U4 [9].

5.2. USB – Universal Serial Bus

Do obsługi interfejsu USB wbudowanego w mikrokontroler podczas realizacji projektu dyplomowego posłużono się biblioteką LUFA USB [15] rekomendowaną przez firmę Microchip. Biblioteka pozwala na konfigurację deskryptorów oraz identyfikatora, co umożliwia automatyczną detekcję układu laboratoryjnego po podłączeniu do komputera PC.

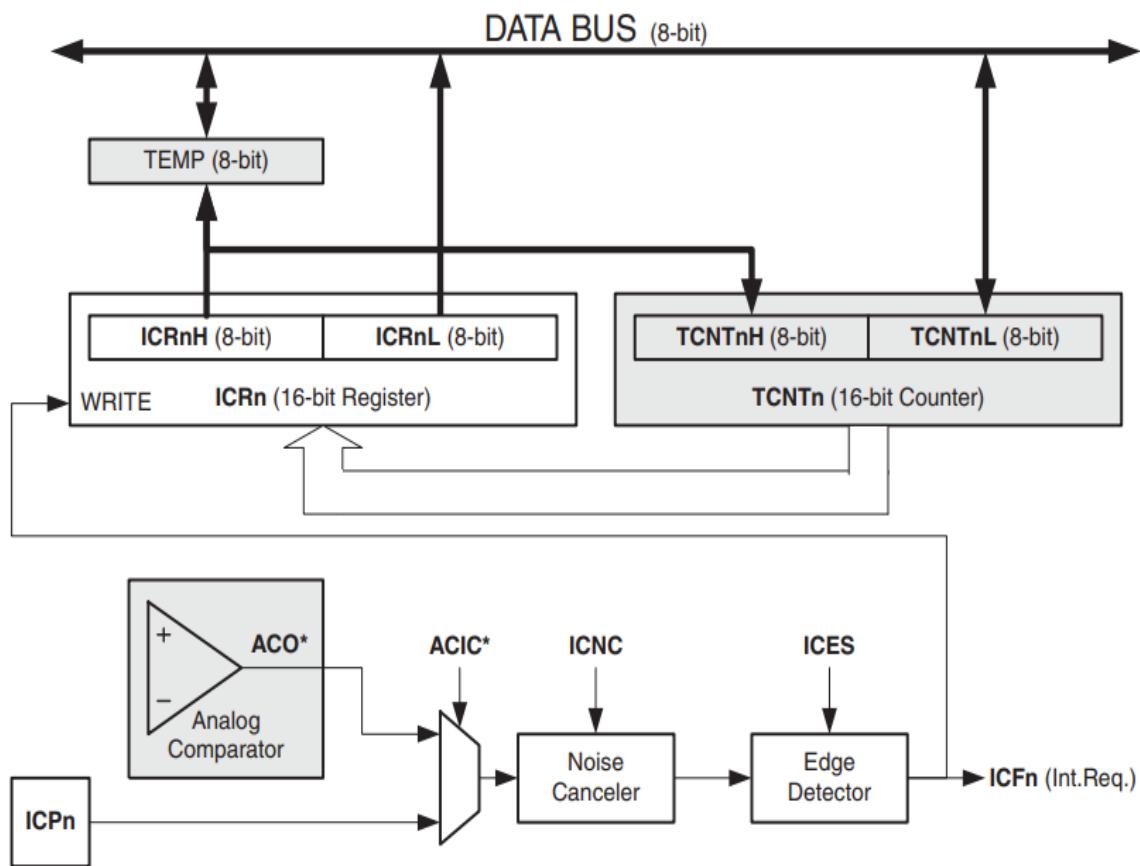
Interfejs USB umożliwia obsługę zdarzenia wynikającego z podłączenia go do komputera, a także nawiązanie połączenia z aplikacją. Jeśli mikrokontroler wykryje pojawienie się napięcia zasilania na wyprowadzeniu VBUS [9], co równoważne jest podłączeniu układu do portu USB komputera, to wywoływane jest przerwanie. Natomiast nawiązanie połączenia aplikacji z mikrokontrolerem odbywa się poprzez przesłanie pakietu, po którym również generowane jest przerwanie. Praca mikrokontrolera z maksymalną prędkością (przy częstotliwości zegara 16 MHz) możliwa jest tylko przy zasilaniu napięciem 5 V, dlatego w wartości napięć panujących na liniach danych stosuje się regulatorы napięcia (rys. 5.2.) dostosowujące do standardu USB (3 – 3,6 V).



Rys. 5.2. Schemat blokowy kontrolera USB [9].

5.3. ICP – Input Capture

Do realizacji projektu wykorzystano 16 bitowy licznik Timer/Counter1, taktowany bezpośrednio sygnałem zegarowym mikrokontrolera. Licznik może pracować w trybie przechwytywania zdarzeń zewnętrznych. W trybie tym wysyła sygnał przerwania po wykryciu zaprogramowanego zbocza na wejściu ICP1 mikrokontrolera zapamiętując aktualny stan licznika w rejestrze ICR1 (rys. 5.3.). Układ detektora zbocza został wyposażony w filtr szumów tłumiący impulsy o czasie mniejszym niż 4 cykle zegarowe. Filtr ten wprowadza opóźnienie równe 4 cyklom zegarowym [9]. Czas pomiędzy wystąpieniem kolejnych zdarzeń, jest mierzony z wykorzystaniem systemu przerwań.



Rys. 5.3. Schemat blokowy układu licznika w trybie przechwytywania zdarzeń [9].

Licznik pracuje w trybie normalnym. Zliczając impulsy zegarowe od wartości 0 do 65535, czas w którym licznik zliczy pełen zakres impulsów wynosi 4,096 ms dla $f_{osc} = 16$ MHz [9]. Natomiast maksymalny szacowany czas ładowania i rozładowania pojemności dla zadanego zakresu od 100 do 300 pF wynosi 200 μ s, jak przedstawiono w rozdziale 3. W celu zwiększenia dokładności pomiarowej zastosowano nadpróbkowanie i decymację.

Za włączenie filtra szumów w układzie, odpowiedzialny jest bit 7 w rejestrze TCCR1B (ICNC1 – Input Capture Noise Canceler) (rys. 5.4.) [9]. Tryb normalnego zliczania jest domyślnie ustawiony po sygnale zerowania. Uruchomienie licznika odbywa się poprzez wybór sygnału zegarowego za pomocą bitów CS12:10. Maksymalną prędkość zliczania uzyskuje się dla wyłączonego dzielnika częstotliwości (CS12 = „0”; CS11 = „0”; CS10 = „1”). Za wybór sposobu detekcji zbocza decyduje stan bitu 6 w rejestrze TCCR1B (ICES1 – Input Capture Edge Select), wartość logiczna „1” oznacza zbocze narastające, natomiast logiczne „0” – opadające [9].

Bit	7	6	5	4	3	2	1	0	
	ICNC1	ICES1	-	WGM13	WGM12	CS12	CS11	CS10	TCCR1B
Read/Write	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Rys. 5.4. Rejestr konfiguracyjny licznika 1 [9].

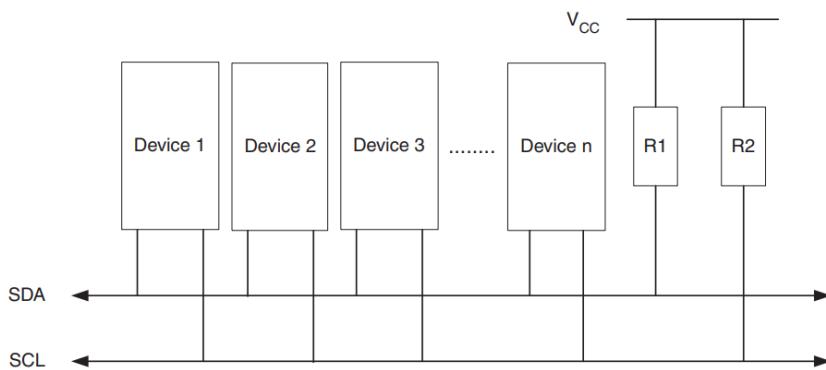
Za zezwolenie na przerwanie od układu przechwytywania zdarzeń dla licznika 1 odpowiedzialny jest bit 5 (ICIE1 – Input Capture Interrupt Enable) w rejestrze TIMSK1 (rys. 5.5.). Chcąc uaktywnić przerwanie, należy ustawić powyżej wspomniany bit rejestru na wartość logiczną „1”.

Bit	7	6	5	4	3	2	1	0	
	-	-	ICIE1	-	OCIE1C	OCIE1B	OCIE1A	TOIE1	TIMSK1
Read/Write	R	R	R/W	R	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Rys. 5.5. Rejestr masek przerwań licznika 1 [9].

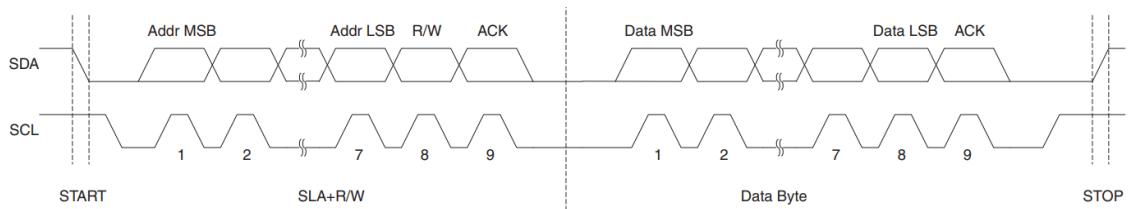
5.4. TWI – Two Wire Interface

Komunikacja z układami zewnętrznymi, tj. wyświetlaczem LCD z ekspanderem wyprowadzeń PCF8574t oraz czujnikiem temperatury i wilgotności względnej SHTC3, odbywa się poprzez szeregowy, synchroiczny interfejs TWI. Zaletą tego interfejsu jest potwierdzanie odbioru danych przez układ podrzędny. W momencie wystąpienia dziewiątego bitu transmisji danych z układu master do slave urządzenie podrzędne zwiera linię danych na czas jednego cyklu zegarowego sygnału taktującego transmisję. Mechanizm ten został wykorzystany do wykrycia obecności tych układów w układzie laboratoryjnym, dzięki czemu odłączenie jednego z układów nie zaburza transmisji TWI. Stan logiczny „1” na linii danych (SDA) i zegara (SCL) wymuszany jest poprzez rezystory podciągające R1 i R2 (rys. 5.6.) do napięcia zasilającego. Oznacza to, że w stanie bezczynności linie danych i zegara są w stanie logicznym „1”.



Rys. 5.6. Sposób podłączenia urządzeń w interfejsie TWI [9].

Transmisja danych zaczyna się od sekwencji start (rys. 5.7.). W kolejności wysyłany jest adres urządzenia podrzędnego. Układ nadzorzący odpowiedzialny jest za generację sygnału zegarowego oraz za wybór urządzenia podrzędnego do komunikacji.



Rys. 5.7. Typowa transmisja danych interfejsu TWI [9].

Po każdym bajcie danych występuje bit potwierdzenia odbioru wystawiany przez zaadresowanie odbierające dane urządzenie. Zakończenie transmisji sygnalizowane jest przez układ nadzorzący, poprzez generowanie sekwencji stop. Parametry transmisji ustawia się w rejestrze kontrolnym interfejsu TWI (rys. 5.8.). Sekwencja start generowana jest po ustawieniu bitu TWSTA a sekwencja stop za pomocą bitu TWSTO. Zakończenie kolejnych etapów transmisji sygnalizowane jest poprzez zerowanie flagi przerwania TWINT. Transmisja danych realizowana jest poprzez wpisanie danych do rejestrów danych TWDR. Natomiast odbiór danych może być sparametryzowany dzięki możliwości zaprogramowania znaku potwierdzenia w układzie nadzorującym. Potwierdzenia sygnalizowane jest przez bit TWEA (TWI Enable Acknowledge). Po każdym etapie transmisji aktualizowany jest status interfejsu w rejestrze TWSR [9], informujący o wystąpieniu błędu arbitrażu, czy błędu wynikającego z braku potwierdzenia.

Bit	7	6	5	4	3	2	1	0	TWCR
Read/Write	R/W	R/W	R/W	R/W	R	R/W	R	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Rys. 5.8. Rejestr kontrolny interfejsu TWI [9].

6. OPROGRAMOWANIE MIKROKONTROLERA

W rozdziale zostanie omówiony proces tworzenia oprogramowania na mikrokontroler ATmega32U4. Szczególną uwagę poświęcono na przedstawienie algorytmów programu, na podstawie których napisano kod. Zostaną także omówione najważniejsze funkcje takie jak: pomiarowe, kalibracyjne i komunikacyjne.

6.1. Struktura programu

Program został napisany w języku ANSI C przy użyciu środowiska EclipseIDE z pluginem AVR [16]. Składa się on z około 2500 linii kodu i zajmuje niemalże połowę pamięci programu mikrokontrolera.

Wykorzystanie interfejsu USB wymusiło wyłączenie układu WatchDog. Do obsługi interfejsu USB wykorzystano rekomendowaną przez firmę Microchip bibliotekę LUFA USB [15]. Biblioteka ta udostępnia szereg funkcjonalności, np. zaprogramowanie deskryptorów, dzięki czemu interfejs użytkownika jest w stanie rozpoznać podłączone urządzenie i nawiązać z nim komunikację. Do obsługi interfejsu USB wymagany jest włączony system przerwań. Status podłączenia modułu Arduino do komputera jest sygnalizowany za pomocą diod LED umieszczonych na tym module. Obsługa interfejsu USB polega na inicjalizacji interfejsu oraz wywołaniu w pętli głównej funkcji z biblioteki LUFA. Stąd nie ma możliwości wprowadzenia mikrokontrolera w stan uśpienia z mniejszym poborem energii.

6.2. Inicjalizacja urządzenia

Po resecie programu, wszystkie porty I/O mikrokontrolera pracują w trybie wejścia. Diody LED zostają wyłączone, a linie wejściowe ustawiane są w trybie pull-up. W ostatnim kroku linie sterujące diodami LED ustawiane są w tryb wyjściowy. Następnie zostaje sprawdzona obecność układów zewnętrznych, dołączonych do interfejsu TWI. Po wykryciu modułu wyświetlacza LCD wysyłany jest komunikat na wyświetlacz LCD, informujący o poprawnym działaniu programu.

Inicjalizacja USB (listing 6.1.) składa się z wyłączenia modułu WatchDog oraz wywołania funkcji inicjalizacyjnej z biblioteki LUFA [15]. Praca z interfejsem USB implikuje maksymalną częstotliwość sygnału zegarowego mikrokontrolera, dlatego dzielnik częstotliwości zostaje wyłączony. Na potrzeby przesyłania danych zostaje zadeklarowany strumień danych, który ułatwia formatowanie i parametryzowanie ciągów znaków wysyłanych do interfejsu użytkownika.

```

void Init_USB(void)
{
    /* USB initialization */
    cbi(MCUSR, WDRF);
    wdt_disable();
    clock_prescale_set(clock_div_1);
    USB_Init();
    CDC_Device_CreateStream(
        &VirtualSerial_CDC_Interface,
        &USB_Stream
    );
}

```

Listing 6.1. Funkcja inicjalizacji interfejsu USB.

Inicjalizacja modułu przechwytywania zdarzeń zewnętrznych polega na wyborze zbocza (listing 6.2.), które ma aktywować przerwanie podczas którego następuje zapamiętanie aktualnego stanu licznika 1. Działanie filtra szumów konfiguruje się za pomocą dyrektyw preprocesora [16]. Po inicjalizacji licznik jest wyłączony, uruchamiany jest on jedynie podczas pomiaru pojemności. W tym samym czasie aktywowane jest zezwolenie na obsługę przerwania.

```

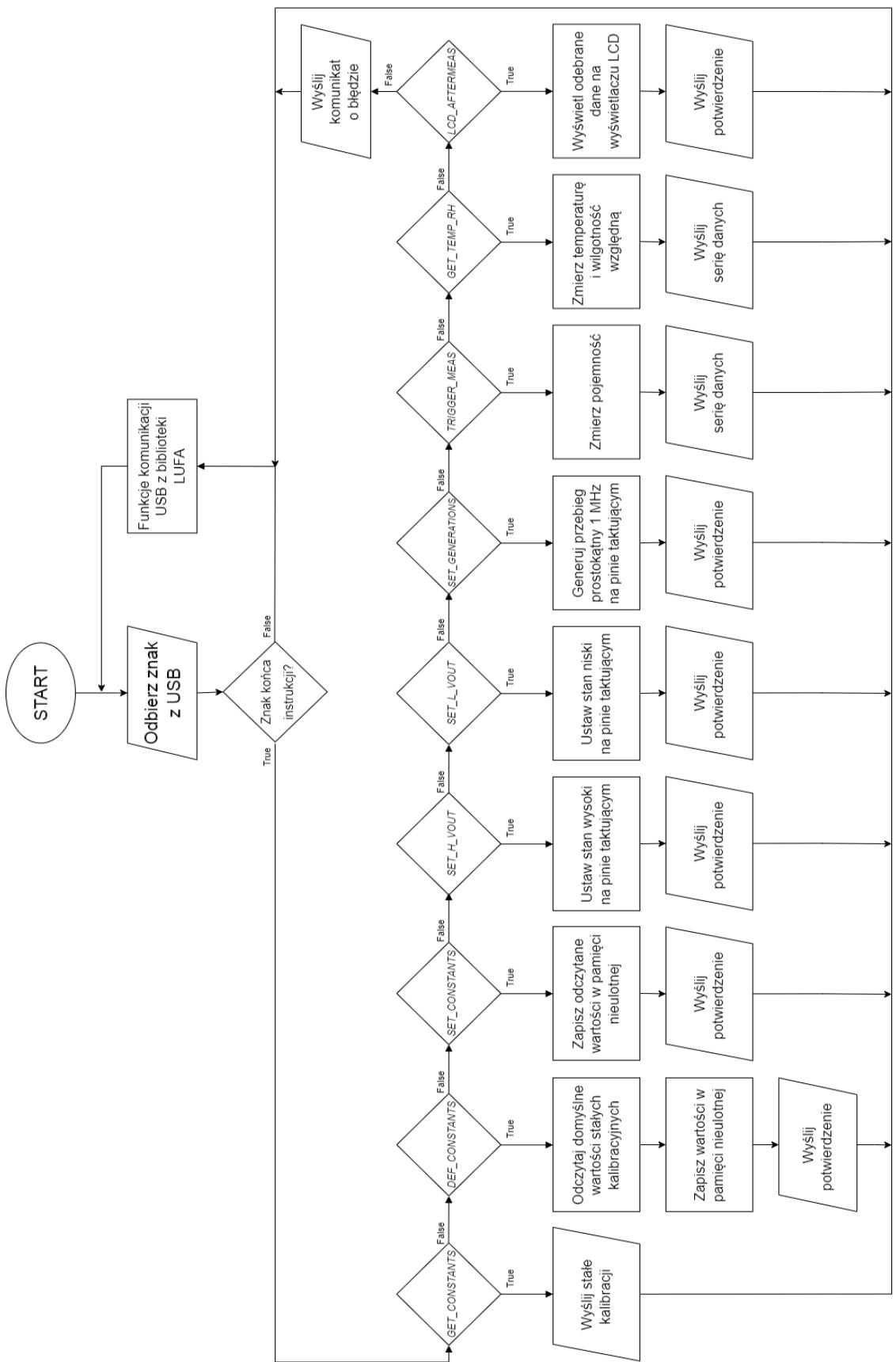
void Init_InputCapture(void)
{
#if ENABLE_NOISE_CANCELER
    /* Input Capture Noise Canceler For ICP1 */
    sbi(TCCR1B, ICNC1);
#endif
    /* Input Capture Edge Select - Rising Edge */
    sbi(TCCR1B, ICES1);
}

```

Listing 6.2. Funkcja inicjalizacji modułu przechwytywania .

6.3. Pętla główna programu

W pętli głównej program na mikrokontroler (rys. 6.1.) są realizowane: interpretacja instrukcji otrzymanych z interfejsu użytkownika, obróbka danych i odsyłanie komunikatu potwierdzenia, serii danych bądź informacji o błędzie. Wszelkie komunikaty o błędach są przesyłane do komputera PC. Układ laboratoryjny obsługuje 10 komend, np.: uruchamiających osobno pomiar pojemności oraz pomiar temperatury i wilgotności względnej otoczenia. Wykonanie serii pomiarów jest realizowane przez interfejs użytkownika, czyli program komputerowy, który co ustalony czas wywołuje pojedynczy pomiar w układzie laboratoryjnym. Zastosowanie funkcji obsługujących interfejs USB wywoływanych w pętli głównej programu funkcji uniemożliwia uśpienie mikrokontrolera na czas jego bezczynności oraz wymusza programowanie w trybie MultiTasking (wielozadaniowości) [16].



Rys. 6.1. Algorytm pętli głównej programu mikrokontrolera.

6.4. Funkcje urządzenia

Układ laboratoryjny obsługuje 10 komend, które można podzielić na trzy grupy:

- transfer danych kalibracyjnych,
- sterowanie liniami mikrokontrolera w celu kalibracji poziomów napięć oraz opóźnień,
- pomiar wielkości elektrycznych i fizycznych.

Po każdym wykonanym poleceniu układ laboratoryjny wysyła potwierdzenie jego wykonania, bądź przesyła serię danych pomiarowych. Urządzenie mierzy wartość temperatury i wilgotności względnej otoczenia z czujnikiem cyfrowego oraz wartość pojemności. Dane pomiarowe są zapamiętywane w pamięci operacyjnej mikrokontrolera. Po wykonaniu serii pomiarów następuje formatowanie danych oraz przesyłanie ich do interfejsu użytkownika. Maksymalny czas pomiaru wynosi 500 ms. Po przekroczeniu tej wartości generowany jest komunikat o błędzie.

6.4.1. Kalibracja

Kod pokazany na (listing 6.3.) jest w celu sterowania liniami mikrokontrolera oraz generację przebiegu prostokątnego o częstotliwości 1 MHz, na wyprowadzeniu pobudzającym układ pomiarowy. Przebieg jest generowany z wykorzystaniem wewnętrznego 8-bitowego licznika [17]. Ustawianie i zerowanie linii nr 11 modułu Arduino Micro pozwala na pomiar napięć na wyjściu bufora trójstanowego.

```
case SET_H_VOUT: /* Set Signal Pin as HIGH */
{
    DisableGenerations();
    SET_GEN_STATE;
    MEASURING_SYSTEM_ON;
    SEND_CONFIRMATION;
    break;
}
case SET_L_VOUT: /* Set Signal Pin as LOW */
{
    DisableGenerations();
    CLR_GEN_STATE;
    MEASURING_SYSTEM_ON;
    SEND_CONFIRMATION;
    break;
}
case SET_GENERATIONS: /* Set Generation on Signal Pin */
{
    EnableGenerations();
    MEASURING_SYSTEM_ON;
    SEND_CONFIRMATION;
    break;
}
```

Listing 6.3. Obsługa instrukcji kalibrujących.

Stałe kalibracji, wyznaczane na podstawie modelu matematycznego obwodu z czujnikiem pojemnościowym zostały zdefiniowane jako struktury danych. Są one przechowywane w nieulotnej pamięci EEPROM (listing 6.4.). Po każdorazowej inicjalizacji układu przesyłane są do pamięci danych [16]. Komenda żądania przesłania danych uruchamia ich formatowane i przesłanie do interfejsu użytkownika. Podczas wyznaczania wartości pojemności interfejs użytkownika korzysta bezpośrednio z danych otrzymanych z układu laboratoryjnego. Użytkownik ma możliwość zmiany wartości każdego z zapisanych elementów, poprzez wywołanie komendy żądania nadpisania stałych wartości kalibracji pamięci nieulotnej EEPROM.

```
const CONSTANTS_t EEMEM Constants_EEMEM =
{
    .H_THR    = 3.3500,
    .L_THR    = 1.6770,
    .H_VOUT   = 5.0230,
    .L_VOUT   = 0.0000,
    .R_MEAS   = 191100,
};

const CORRECTIONS_t EEMEM Corrections_EEMEM =
{
    .A0 = 3.2231,
    .A1 = -4.4837,
    .A2 = 2.7930,
    .A3 = -4.6443,
};
```

Listing 6.4. Definicja domyślnych wartości struktur danych kalibracyjnych.

6.4.2. Pomiar temperatury i wilgotności względnej

Temperatura i wilgotność względna jest mierzona za pomocą cyfrowego czujnika SHTC3. Komunikacja z czujnikiem odbywa się poprzez interfejs TWI. Zaletą jaka zdecydowała o wyborze tego podzespołu jest krótki czas pomiaru (15 ms) przy zapewnieniu wystarczającej dokładności pomiarowej $\pm 2\%$ RH oraz $\pm 0,2^\circ\text{C}$ [13]. Dodatkowo element posiada małe wymiary dzięki czemu można go umieścić blisko czujnika pojemnościowego HS1101, dla zachowania takich samych warunków panujących wokół badanego czujnika. SHTC3 charakteryzuje się niskim poborem prądu podczas pomiaru, który wynosi $900 \mu\text{A}$ [13]. Komunikacja może odbywać się na podwyższonej częstotliwości pracy interfejsu TWI (400 kHz). Pomiar rozpoczyna się po wysłaniu instrukcji przez interfejs TWI. Po czasie 15 ms mikrokontroler może odczytać dane z cyfrowego czujnika SHTC3. Wraz z wartościami temperatury i wilgotności przesyłana jest suma kontrolna, co znacząco ogranicza możliwości wystąpienia błędów transmisji.

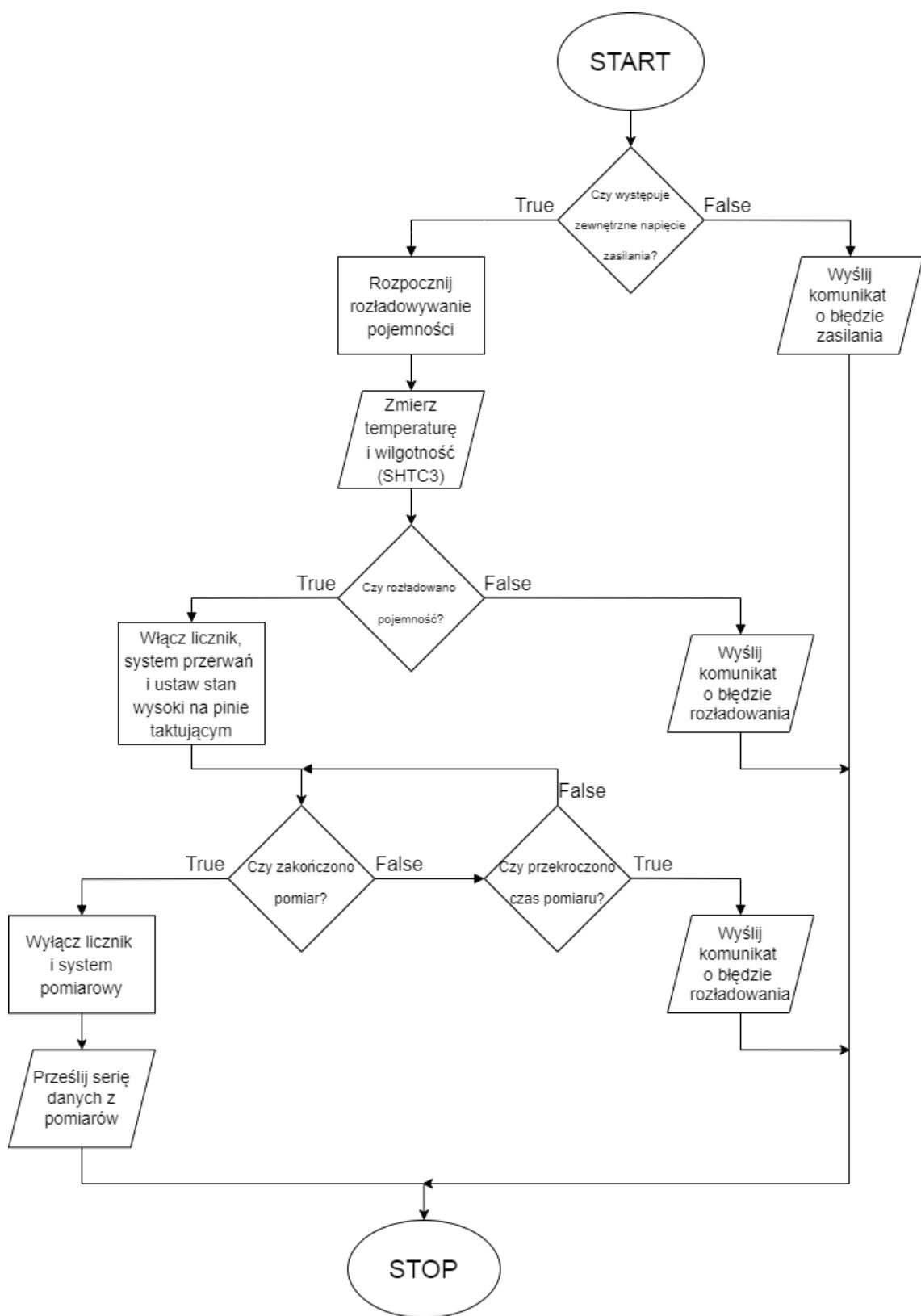
6.4.3. Pomiar pojemności

Główym zadaniem układu laboratoryjnego jest pomiar pojemności. Opiera się on na pomiarze czasów ładowania i rozładowania kondensatora znajdującego się w układzie. Po odebranie instrukcji aktywującej pomiar, w pierwszej kolejności sprawdzana jest obecność zewnętrznego napięcia zasilania (rys. 6.2.), ponieważ zestaw zasilany jest także z portu USB. W przypadku braku zewnętrznego napięcia zasilania, układ laboratoryjny informuje o tym interfejs użytkownika, nie wykonując pomiarów pojemności. W przeciwnym przypadku układ rozpoczyna rozładowywanie pojemności badanej poprzez aktywację obwodu pomiarowego i wystawienie stanu niskiego na linii pobudzającej układ pomiarowy. Rozpoczyna się również pomiar temperatury i wilgotności względnej czujnikiem cyfrowym SHTC3.

Gdy napięcie na kondensatorze osiągnie wartość napięcia poniżej dolnego progu referencyjnego generowany jest impuls wyzwalający układ przechwytywania zdarzeń zewnętrznych oraz generujący przerwanie. Program główny (listing 6.5.) oczekuje na ustawienie flagi zakończenia serii pomiarowej przez obsługę przerwania. W pętli sprawdzającej zakończenie serii pomiarowej następuje sprawdzenie jej czasu trwania. Przekroczenie tego czasu powoduje anulowanie dotychczasowych wyników oraz wysłanie komunikatu o błędzie do interfejsu użytkownika.

Obsługa przerwania (listing 6.6.) sprowadza się do zapamiętania w buforze danych wartości rejestru (ICR1) z przechwyconą wartością licznika przez moduł Input Capture [17] i wyzerowania licznika 1. W kolejnym kroku następuje sprawdzenie liczby wykonanych pomiarów. Przekroczenie granicznej wartości powoduje ustawienie flagi zakończenia pomiarów. Przed zakończeniem obsługi przerwania następuje także ustawienie przeciwnego stanu na linii pobudzającej układ pomiarowy, a tym samym zmiana fazy ładowania/rozładowania układu RC.

Po zakończeniu pomiarów w pierwszej kolejności deaktywowany jest system przerwań oraz moduł przechwytywania zdarzeń zewnętrznych. Obwód pomiarowy zostaje odłączony od zasilania, poprzez przejście bufora trójstanowego w stan wysokiej impedancji. Ostatnia wartość opóźnienia zostaje wyznaczona w stanie rozładowywania układu RC, stąd badana pojemność po jego zakończeniu będzie gotowa na szybkie wznowienie pomiarów. W ostatniej fazie następuje formatowanie danych i ich transfer do interfejsu użytkownika.



Rys. 6.2. Algorytm prezentujący pomiar pojemności.

```

/* Capacity Measuring Function */
STATUS_t CapacityMeasurement(void)
{
    /* Check Power Status */
    if (!PWR_STATUS)
    {
        sprintf_P(&USB_Stream, PSTR("Error: Check the power supply"));
        return Status_PowerError;
    }
    /* Discharge Capacity */
    CLR_GEN_STATE;
    MEASURING_SYSTEM_ON;
    /* Temperature and Humidity Measurement */
    STATUS_t Status = TemperatureMeasurement();
    /* Wait for LOW State on V_Cap Pin */
    while (!V_CAP_IS_UNDER_LTHR)
    {
        [...]
    }
    /* Measurement conditions ready */
    /* Set sample index to 0 */
    System.SampleIdx = 0;
    System.IsMeasurementEnd = false;
    EnableInputCapture();
    SET_GEN_STATE;
    /* Waiting for end measurement */
    cnt = 0;
    while (!System.IsMeasurementEnd)
    {
        [...]
    }
    /* Measurement End - Disable measuring system */
    DisableInputCapture();
    CLR_GEN_STATE;
    MEASURING_SYSTEM_OFF;
    /* Data presentation */
    [...]
}

```

Listing 6.5. Funkcja pomiaru pojemności.

Pomimo wysokiej optymalizacji kompilatora języka C, wygenerowany kod w assemblerze, składa się z szeregu zbędnych instrukcji obsługi przerwania. Obecne zajmowana pamięć przez kod obsługi przerwania to 100 bajtów, a czas potrzebny na jej wykonanie wynosi około 6 µs.

```

/* Input Capture Interrupt */
ISR(TIMER1_CAPT_vect, ISR_BLOCK)
{
    TCNT1 = 0;
    System.Samples[System.SampleIdx++] = ICR1;
    if (System.SampleIdx == MAX_SAMPLES_BUF_SIZE)
        System.IsMeasurementEnd = true;
    TOG_GEN_STATE;
}

```

Listing 6.6. Obsługa przerwania modułu przechwytywania zdarzeń zewnętrznych.

7. INTERFEJS UŻYTKOWNIKA NA KOMPUTERZE PC

Interfejs użytkownika na komputerze PC jest jednym z najważniejszych elementów projektu. Zapewnia on komunikację z układem laboratoryjnym, wykonuje obliczenia i prezentuje dane. W rozdziale zostanie zaprezentowana realizacja tego programu. Szczegółową uwagę poświęcono na omówienie funkcji matematycznych, korekcji charakterystyki oraz generacji skryptów dla programu Matlab.

Interfejs użytkownika został napisany w języku C# z wykorzystaniem bibliotek WindowsForms [18]. Kod programu składa się z około 3500 linii. Program zapewnia komunikację z urządzeniem laboratoryjnym, wykresła charakterystyki przejściowe z zarejestrowanych serii pomiarowych i generuje skrypt dla programu Matlab. Obsługiwany jest także szczegółowy dziennik zdarzeń umożliwiający monitorowanie działania systemu.

7.1. Strona startowa

Po uruchomieniu interfejsu użytkownika (rys. 7.1.) w pierwszym oknie ukazują się najważniejsze informacje dotyczące osób odpowiedzialnych za projekt. W dolnej części okna wyświetlana jest informacja o statusie połączenia z układem laboratoryjnym, pasek informacyjny jest widoczny z poziomu każdego okna. Zawiera on także informację o ostatnim zdarzeniu w systemie. Z poziomu okna istnieje możliwość przejścia do ustawień ogólnych, kalibracji układu czy obsługi pomiarów.



Rys. 7.1. Strona startowa interfejsu użytkownika.

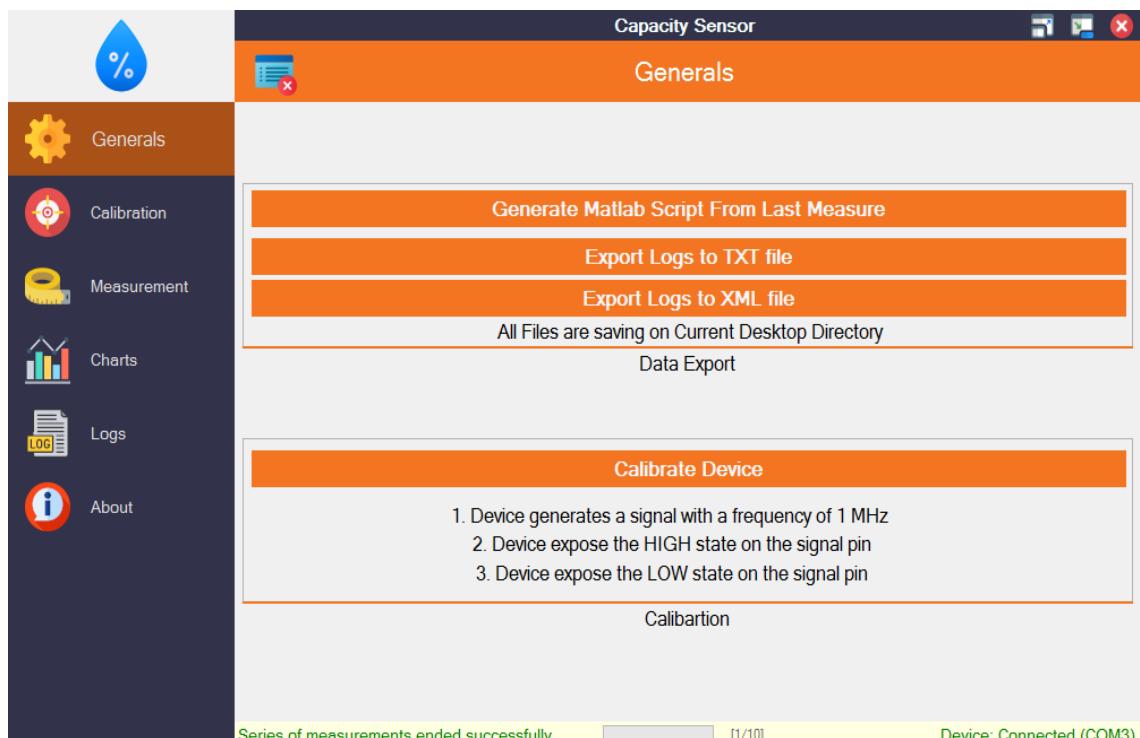
7.2. Ustawienia ogólne

Z poziomu okna ustawień ogólnych (rys. 7.2.) istnieje możliwość uruchomienia kalibracji urządzenia, która realizowana jest w trzech krokach:

- na wyprowadzeniu pobudzającym obwód RC, generowany jest przebieg prostokątny o częstotliwości 1 MHz,
- wyprowadzenie pobudzającym obwód pomiarowy przyjmuje stan wysoki,
- wyprowadzenie pobudzającym obwód pomiarowy przyjmuje stan niski.

Takie podejście pozwala na zmierzenie poziomów napięć wyjściowych z bufora trójstanowego, które mogą się zmieniać np. pod wpływem zmian temperatury otoczenia. Następnie zmierzone wartości napięć należy wprowadzić korzystając z okna kalibracji. Natomiast obserwacja sygnału 1 MHz za pomocą oscyloskopu pozwala określić opóźnienia wprowadzane przez układ, a tym samym na oszacowanie pojemności pasożytniczych obecnych w układzie.

Oprócz kalibracji, aplikacja umożliwia wyeksportowanie danych pomiarowych w postaci skryptu programu Matlab oraz dziennika zdarzeń w postaci pliku tekstowego lub XML [18]. Utworzone pliki zostają zapisane na pulpicie komputera, po czym są otwierane przez domyślne aplikacje.



Rys. 7.2. Ustawienia ogólne interfejsu użytkownika.

7.2.1. Eksport wyników pomiarowych do programu Matlab

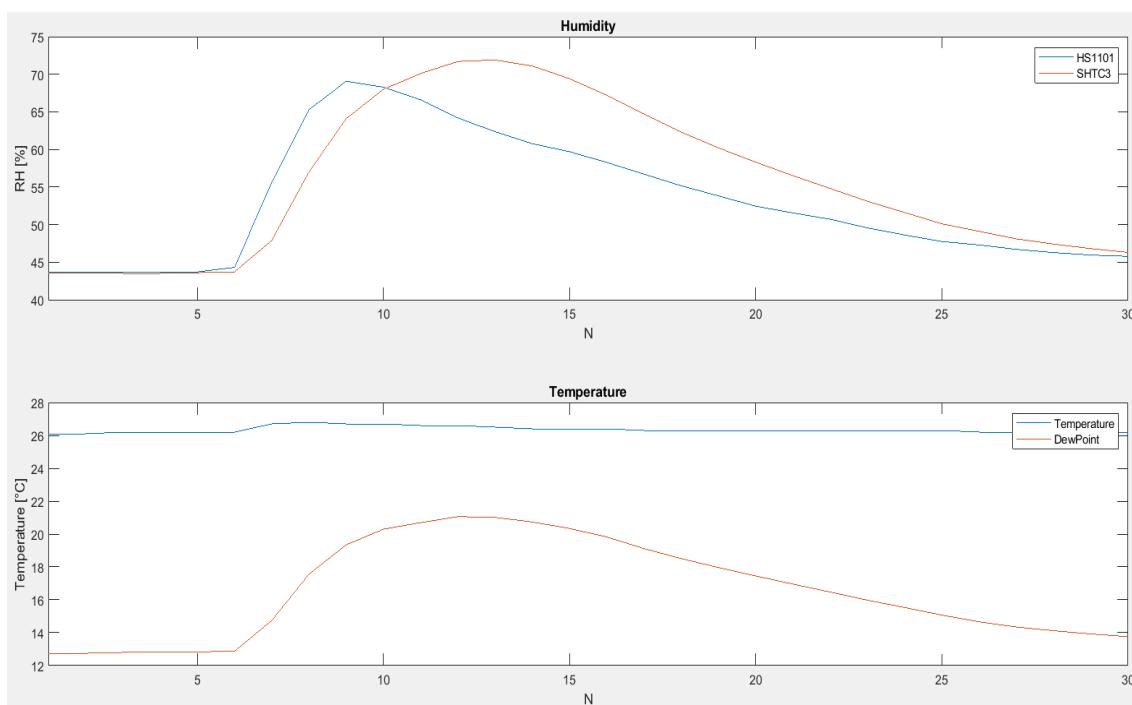
Skrypt Matlab służy do wykreszenia charakterystyk czasów ładowania oraz rozładowania w zależności od:

- pojemności z zakresu od 100 do 300 pF dla stałej wartości rezystancji RM,
- rezystancji pomiarowej RM z zakresu 100 kΩ do 1 MΩ dla stałej pojemności badanej.

Charakterystyki te pozwalają na ocenę wyników pomiarów dla podanych wartości rezystancji oraz mierzonej pojemności. Zbyt mała wartość rezystancji będzie skutkowała krótkimi czasami pomiarowymi, co za tym idzie zbyt szybkim ładowaniem pojemności, a tym samym niedokładnym pomiarem czasu.

Skrypt służy również do generacji histogramu z pomiarów wykonywanych w ramach serii pomiarowych. Odpowiednio duża liczba wyników powinna tworzyć rozkład Gaussa, co pokazano w dalszej części pracy. Histogram pozwala statystycznie ocenić czy pomiar został wykonany z należytą dokładnością. Przy pomocy skryptu wyznacza się także wykresy błędów losowych dla czasów ładowania i rozładowania badanej pojemności.

Wyznaczona zostaje także charakterystyka wilgotności względnej zmierzona za pomocą czujnika pojemnościowego HS1101 oraz cyfrowego SHTC3 (rys. 7.3.). Użytkownik ma możliwość ocenienia dokładności kalibracji układu laboratoryjnego oraz doświadczalnego sprawdzenia bezwadności i pojemności cieplnej obu podzespołów.



Rys. 7.3. Przykład charakterystyki wilgotności i temperatury w czasie.

7.2.2. Eksport dziennika zdarzeń

W przypadku dużej liczby przeprowadzonych pomiarów wygenerowany plik XML może posłużyć do prezentacji danych ze wszystkich wykonanych pomiarów w ciągu całej sesji. Na przykład, generując dokument XSLT przetwarzający plik XML (listing 7.1.) można ukazać sformatowane dane w przeglądarce internetowej. Funkcja ta będzie jednym z celów rozwoju tego projektu.

Dane zapisane w formacie tekstowym XML nie są przejrzyste dla użytkownika, dlatego zdecydowano również na eksport danych do formatu tekstowego, co pozwala na archiwizację danych pomiarowych pochodzących z całej serii.

```
<CapacitySensorLogs DateGenerated="2022-11-06 20:13:31">
    <Log Date="2022-11-06 19:13:09.401" From="User" Type="Info">The application was opened</Log>
    <Log Date="2022-11-06 19:13:10.525" From="Device" Type="Info">Device Connection Success - COM3</Log>
    <Log Date="2022-11-06 19:13:10.612" From="PC" Type="Info">Sent Command: GET_CONSTANTS (A)</Log>
    <Log Date="2022-11-06 19:13:10.615" From="Device" Type="Info">Data Received: 3.3500 1.6770 5.0230 0.0000 741200</Log>
    <Log Date="2022-11-06 19:13:10.703" From="PC" Type="Info">Sent Command: GET_CORRECTIONS (I)</Log>
    <Log Date="2022-11-06 19:13:10.706" From="Device" Type="Info">Data Received: 3.2231 -4.4837 2.7930 -4.6443</Log>
    <Log Date="2022-11-06 19:13:20.048" From="PC" Type="Info">Series start of full measurements</Log>
    <Log Date="2022-11-06 19:13:20.172" From="PC" Type="Info">Sent Command: TRIGGER_MEAS (M)</Log>
    <Log Date="2022-11-06 19:13:20.221" From="Device" Type="Info">Data Received: T 24.7 H 46.1 C 2605 1638 1644 1640...</Log>
    <Log Date="2022-11-06 19:13:21.164" From="PC" Type="Info">Sent Command: TRIGGER_MEAS (M)</Log>
    <Log Date="2022-11-06 19:13:21.213" From="Device" Type="Info">Data Received: T 24.7 H 46.0 C 2669 1637 1643 1638...</Log>
    <Log Date="2022-11-06 19:13:22.164" From="PC" Type="Info">Sent Command: TRIGGER_MEAS (M)</Log>
    <Log Date="2022-11-06 19:13:22.213" From="Device" Type="Info">Data Received: T 24.7 H 45.9 C 2695 1637 1643 1638...</Log>
    <Log Date="2022-11-06 19:13:23.176" From="PC" Type="Info">Sent Command: TRIGGER_MEAS (M)</Log>
    <Log Date="2022-11-06 19:13:23.224" From="Device" Type="Info">Data Received: T 24.7 H 45.8 C 2704 1638 1643 1638...</Log>
    <Log Date="2022-11-06 19:13:24.188" From="PC" Type="Info">Sent Command: TRIGGER_MEAS (M)</Log>
    <Log Date="2022-11-06 19:13:24.237" From="Device" Type="Info">Data Received: T 24.7 H 45.8 C 2758 1638 1644 1639...</Log>
    <Log Date="2022-11-06 19:13:25.069" From="PC" Type="Info">Series of measurements ended successfully</Log>
    <Log Date="2022-11-06 19:13:31.253" From="User" Type="Info">Log XML Document Generated - Desktop\Logs-201331.xml</Log>
</CapacitySensorLogs>
```

Listing 7.1. Przykład eksportu dziennika zdarzeń w formacie XML.

7.3. Kalibracja

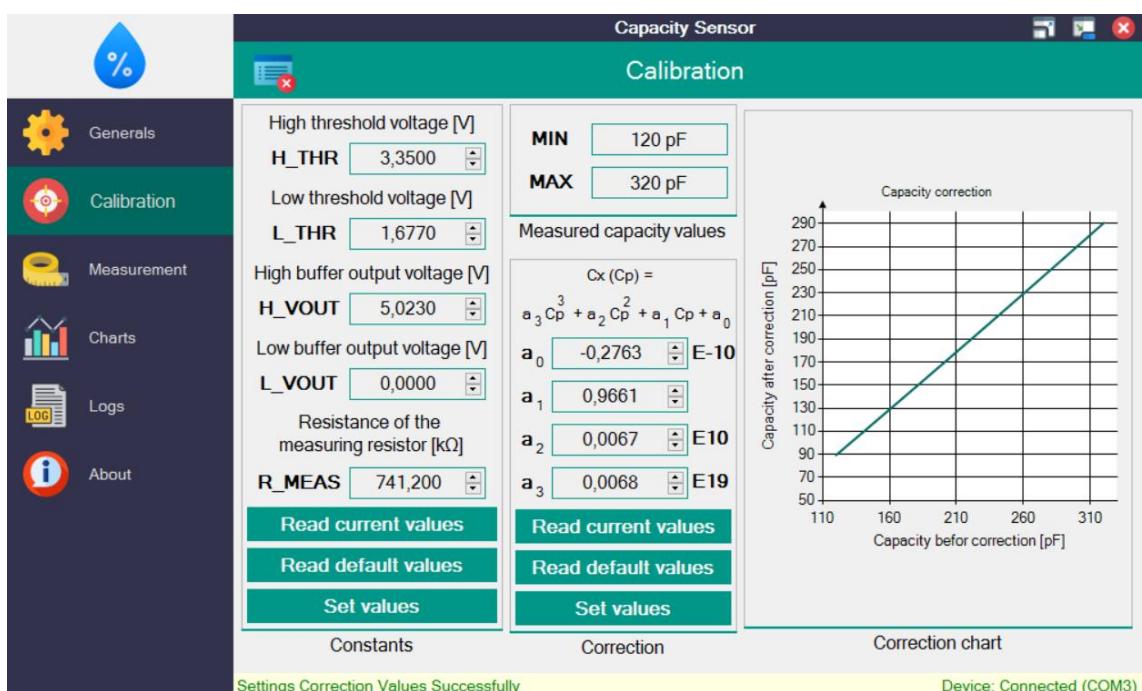
Kalibracja urządzenia w przypadku pomiaru wielkości pojemności rzędu pF jest jednym z kluczowych elementów wpływających na dokładność pomiarową. Zmiana temperatury bezpośrednio wpływa na stabilizowane napięcie przez układ 7805, czy na napięcie wyjściowe z bufora trójstanowego. Również występujące zakłócenia elektromagnetyczne mogą wpływać na dokładność pomiarów, np. zaindukowane napięcie na linii połączonej z badaną pojemnością może zostać zinterpretowane przez komparator jako wystąpienie zdarzenia, co w efekcie będzie powodowało błędne wywołanie przerwania. Wykonany układ laboratoryjny nie został odizolowany od takich wpływów.

Wartości współczynników kalibracji są zapisywane w pamięci nieulotnej EEPROM mikrokontrolera znajdującego się w układzie laboratoryjnym. Ostatnio zapisane wartości zostają odczytane przy każdej inicjalizacji układu laboratoryjnego. W pamięci FLASH są zapisane również domyślne wartości tych współczynników, które można wczytać z interfejsu użytkownika.

Wartości współczynników kalibracji są źródłem danych dla obliczeń matematycznych. Dane kalibracyjne, określone za pomocą wzoru (3.4.) to:

- napięcia referencyjne (H_THR, L_THR),
- napięcia wyjściowe bufora w stanie niskim i wysokim (H_VOUT, L_VOUT),
- rezystancja rezystora pomiarowego (R_MEAS),
- współczynniki korekcji charakterystyki pojemnościowej ($a_0 - a_3$).

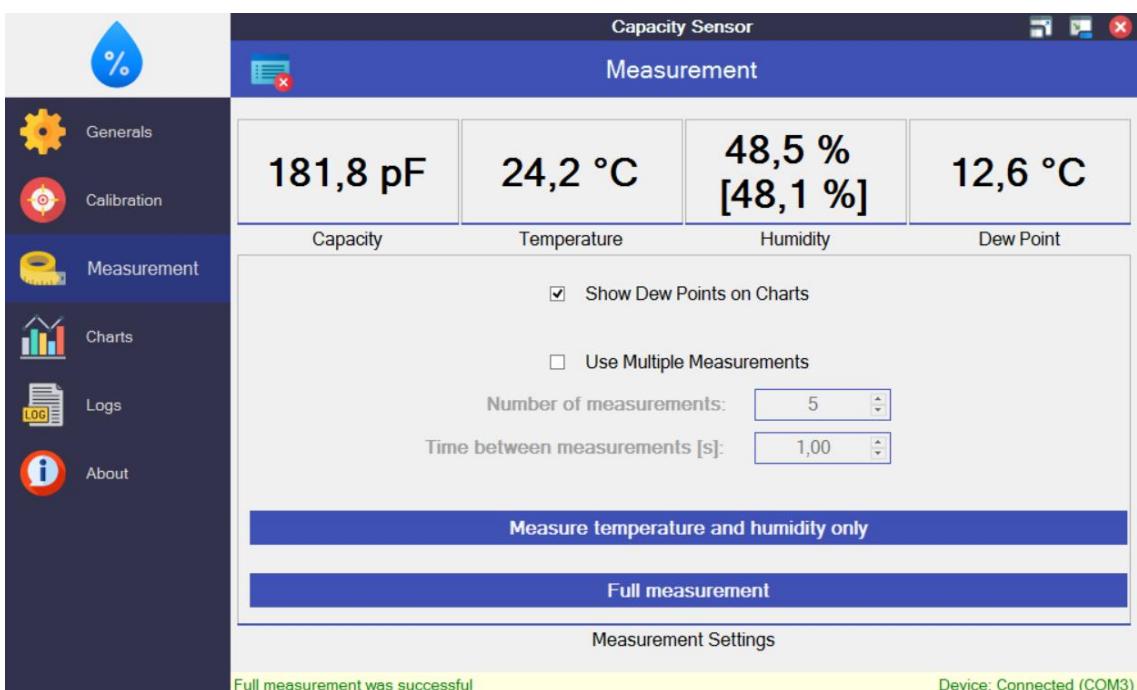
Dodatkowo na podstawie współczynników korekcji generowany jest wykres, którego celem jest zobrazowanie jakie wartości przyjmą zmierzone pojemności dla zakresu 120 – 320 pF.



Rys. 7.4. Kalibracja urządzenia z poziomu interfejsu użytkownika.

7.4. Funkcje pomiarowe

Pomiar pojemności może odbywać się w jednym z dwóch trybów (rys. 7.5.) – trybu pojedynczego pomiaru oraz trybu serii pomiarowej. W drugim przypadku generowany jest wykres. Wielkość wykresu zależy od liczby punktów pomiarowych ustalonej przez użytkownika. Można także ustawić czas pomiędzy pomiarami. Minimalna wartość tego czasu wynosi 500 ms. Dodatkowo w oknie przedstawione są wyniki ostatniego pomiaru. Wyświetlana jest wartość zmierzonej pojemności czujnika, temperatura otoczenia, temperatura punktu rosy oraz wskazania wilgotności względnej od czujnika SHTC3 i wyznaczona wilgotność względna otoczenia na podstawie zmierzonej pojemności czujnika. Druga wartość zapisana jest w nawiasie kwadratowym.



Rys. 7.5. Wywołanie pomiarów z poziomu interfejsu użytkownika.

Wyznaczenie wartości pojemności badanej z danych pomiarowych (listing 7.2.) sprowadza się do podzielenia kolejnych próbek czasu na te pochodzące od pomiaru czasu ładowania oraz pochodzące od pomiaru czasu rozładowania. W następnej kolejności na obu zbiorach wykonuje się nadpróbkowanie i decymacja. Wykonuje się to na danych pochodzących bezpośrednio z licznika mikrokontrolera. Stąd przejście na wartość czasu realizowane jest przez mnożenie przez czas trwania okresu cyklu jednego okresu zegarowego mikrokontrolera. Na podstawie tak otrzymanych danych oblicza się wartości pojemności. Pojemność wynikowa otrzymywana jest poprzez korekcję charakterystyki, na podstawie danych kalibracyjnych.

```

public static void ParseC(string Received)
{
    [...]
    var DP_Capacity = Tick * Oversampling(DischargingProbes);
    var CP_Capacity = Tick * Oversampling(ChargingProbes);

    DP_Capacity = Capacity(DP_Capacity, Calibration.R_MEAS, Calibration.J,
                           Calibration.L_THR, Calibration.H_THR, Calibration.L_VOUT) * 1E3;
    CP_Capacity = Capacity(CP_Capacity, Calibration.R_MEAS, Calibration.J,
                           Calibration.H_THR, Calibration.L_THR, Calibration.H_VOUT) * 1E3;

    var CapacityCorr = Correction((DP_Capacity + CP_Capacity) / 2.0);
    var CapacityRound = double.Parse(string.Format("{0:0.0}", CapacityCorr));

    MainForm.Instance.C = CapacityRound;
    MainForm.Instance.RH = CalcHumidity(CapacityRound);
    [...]
}

```

Listing 7.2. Funkcja wyznaczająca pojemność z danych pomiarowych.

W celu poprawy dokładności pomiarowej zwiększoną reprezentacją bitową wyników pomiarowych. Aby uzyskać o jeden bit większą reprezentację liczbową należy próbkować sygnał czterokrotnie. Zwiększenie rozdzielczości o N bitów wymaga wykonania 4^N próbek sygnału (listing 7.3.).

W programie mikrokontrolera zdefiniowano liczbę pomiarów na 128, tzn. 64 próbki ładowania oraz 64 rozładowania. Wykonanie takiej serii pozwala na zwiększenie reprezentacji bitowej o 3 bity [19]. Zatem zamiast reprezentacji próbki w zakresie 16 bitów (0 – 65535), uzyskano 19 bitów (0 – 524287). Kolejną z zalet operacji nadpróbkowania jest uśrednianie serii wyników pomiarowych. Zmniejsza to wartość błędów losowych o \sqrt{N} [19]. Algorytm nadpróbkowania i decymacji wyznacza sumę wartości próbek pomiarowych, a następnie przesuwa ten wynik o 3 bity w prawo.

```
public static double Oversampling(List<int> Probes)
{
    int bits = 3;
    int N = (int) Math.Pow(4, bits);
    long sum = 0;
    for (int i = Probes.Count - N; i < Probes.Count; i++)
        sum += Probes[i];
    double result = sum >> bits;
    return result * Math.Pow(2, -bits);
}
```

Listing 7.3. Funkcja realizująca nadpróbkowanie i decymację.

Jedną z głównych funkcji pomiarowych jest funkcja wyznaczająca wartość pojemności badanej, na podstawie zmierzonych czasów ładowania i rozładowania (listing 7.4.). Funkcja implementuje wzór (3.4.) wyznaczony dla idealnego obwodu pomiarowego, dlatego konieczna jest późniejsza korekcja charakterystyki wielomianem 3-go stopnia.

Metoda ta zawiera listę współczynników kalibracyjnych. Wartość pojemności zostaje obliczona na podstawie wyników serii pomiarów czasów ładowania i rozładowania badanego układu RC. Współczynniki wielomianu korygującego zostały wyznaczone doświadczalnie na podstawie pomiaru znanych, wzorcowych wartości pojemności.

```
public static double Capacity(double T, double RM, double JC,
    double VCapStop, double VCapStart, double VOut)
{
    return -T / RM / Math.Log((VCapStop - VOut + JC * RM) /
        (VCapStart - VOut + JC * RM));
}
```

Listing 7.4. Funkcja obliczająca pojemność.

Producent układu HS1101 w dokumentacji technicznej podaje, iż charakterystyka pojemności w zależności od wilgotności względnej nie przebiega w sposób liniowy [8]. Wzór konwersji został zasięgnięty z dokumentacji technicznej. Jest to odwrotny wielomian odpowiedzi czujnika pojemnościowego na zmiany wilgotności względnej otoczenia. Wzór (7.1) został zrealizowany za pomocą funkcji aplikacji interfejsu użytkownika (listing 7.5.).

$$RH = -3465,5 \cdot X^3 + 10732 \cdot X^2 - 10457 \cdot X + 3245,9 \quad (7.1)$$

$$X = \frac{C_{zmierzzone}}{C_{typowe}} \quad (7.2)$$

gdzie:

- RH – wilgotność względna [%],
- $C_{zmierzzone}$ – zmierzona wartość pojemności [pF],
- C_{typowe} – typowa wartość pojemności czujnika dla RH = 55% – 180 pF [8].

```
public static double CalcHumidity(double HS1101_Capacity)
{
    double X = HS1101_Capacity / 180.0;
    double RH = -3465.5 * Math.Pow(X, 3) +
                10732.0 * Math.Pow(X, 2) - 10457.0 * X + 3245.9;
    return RH;
}
```

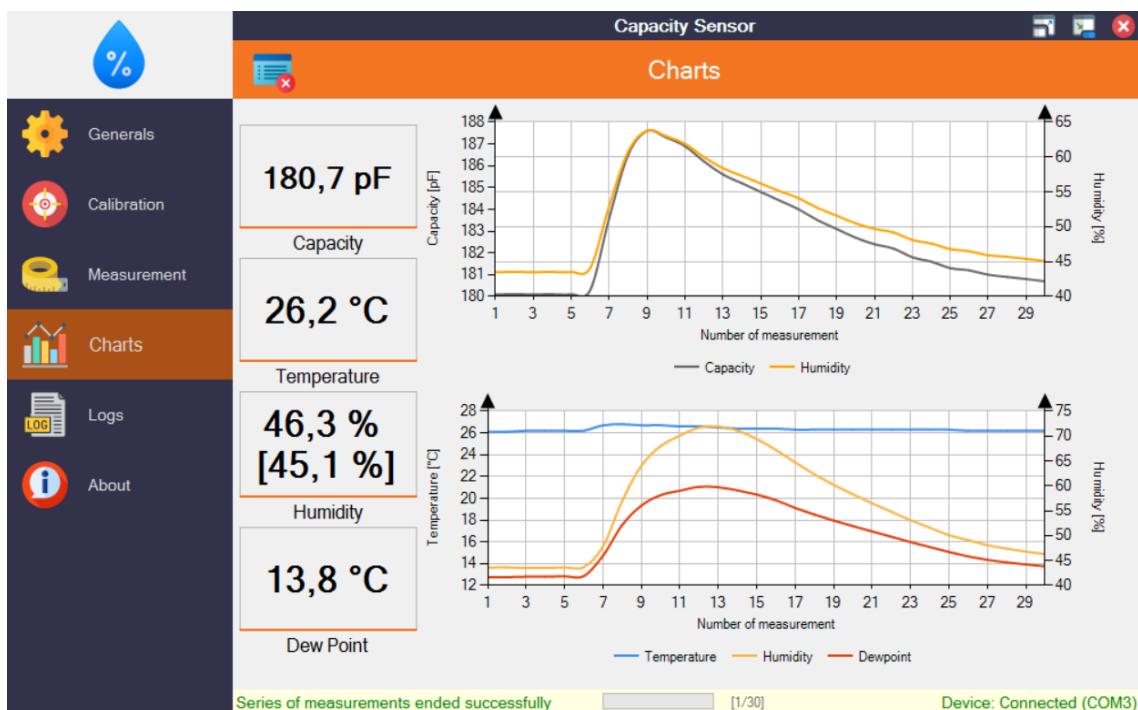
Listing 7.5. Funkcja obliczająca wartość wilgotności względnej.

7.5. Charakterystyki pomiarów

Zaletą systemu jest prezentacja danych w czasie rzeczywistym. Możliwość zaprogramowania serii pomiarów co zadany, ścisłe określony czas pozwala na rejestrację temperatury i wilgotności względnej otoczenia w długim okresie bez potrzeby ciągłego wyzwalania pomiaru przez użytkownika. Interfejs użytkownika rejestruje (rys. 7.6.):

- zmierzona pojemność,
- wilgotność względną na podstawie pojemności czujnika HS1101,
- temperaturę i wilgotność względną otoczenia zmierzona przez czujnik SHTC3,
- temperaturę punktu rosy wyznaczoną na podstawie temperatury i wilgotności.

Ponadto, wygenerowany skrypt dla programu Matlab umożliwia dodatkową prezentację wyników w postaci histogramu wskazań pojemności oraz wykresu błędów losowych. Użytkownik systemu posiada również możliwość ukrycia charakterystyki temperatury punktu rosy.



Rys. 7.6. Podgląd charakterystyk w interfejsie użytkownika.

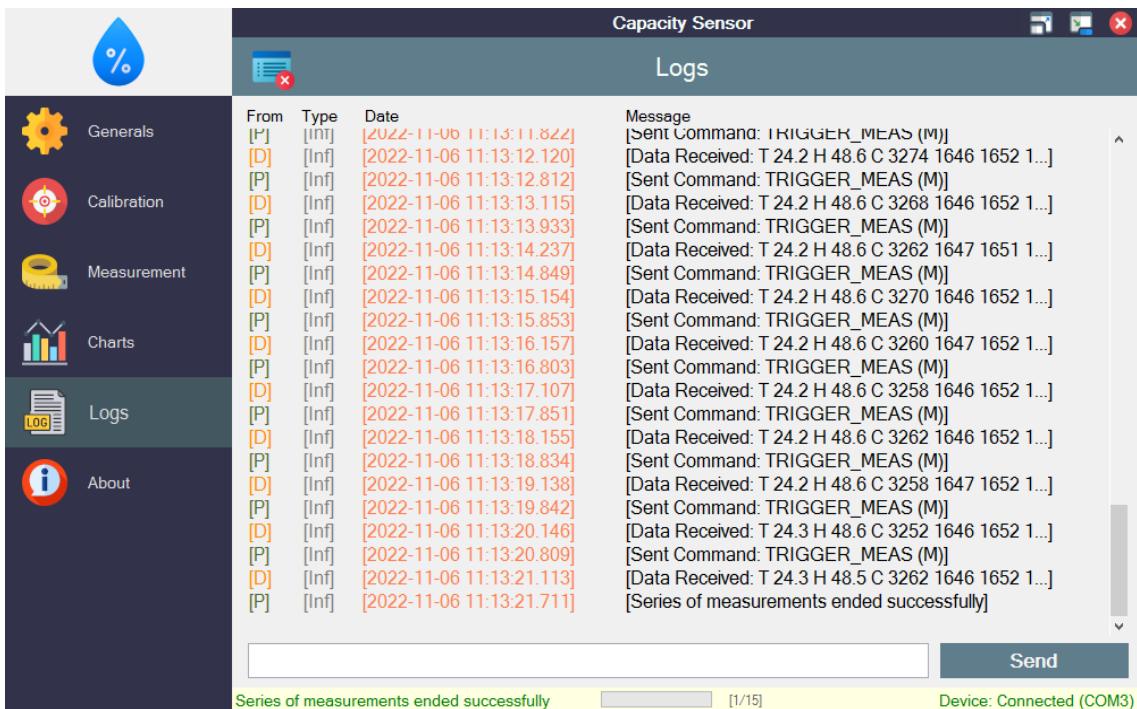
Charakterystyki przejściowe pozwalają na obserwację właściwości czasowych obu czujników wilgotności względnej. Zachowanie obu czujników przy wolnych zmianach wilgotności jest zbliżone do siebie. Natomiast dla nagłych zmian, spowodowanych, np. podmuchem wilgotnego powietrza widać już różnicę w przebiegach wynikającą z właściwości cieplnych, pojemności własnych, stałych czasowych odpowiedzi, czy wielkości obudowy.

7.6. Dziennik zdarzeń

Ważnym zadaniem systemu jest rejestracja zdarzeń. Należy wspomnieć, że aktualne informacje o błędach oraz wewnętrznych zdarzeniach systemu pojawiają się na pasku statusowym wyświetlonym w dolnej części okna.

Dziennik zdarzeń (rys. 7.7.) zawiera informacje dla użytkownika o sytuacjach, jakie zaszły w czasie sesji pomiarowej. Czas wystąpienia wydarzeń jest zapisywany z dokładnością do pojedynczych ms. Zapisywany jest również jego rodzaj i pochodzenie zdarzenia. Błędy wyróżniane są kolorem czerwonym.

Z poziomu ustawień ogólnych istnieje możliwość wyeksportowania danych w formacie tekstowym jak i XML. Z poziomu dziennika zdarzeń użytkownik może także zapoznać się z dostępnymi instrukcjami.



Rys. 7.7. Dziennik zdarzeń interfejsu użytkownika.

Dodatkową, zaimplementowaną funkcjonalnością jest terminal służący do bezpośredniej komunikacji z urządzeniem. Użytkownik z poziomu terminala ma możliwość wysłania dowolnej instrukcji (listing 7.6.). Terminal pozwala jedynie na przesyłanie komend pomiędzy układem laboratoryjnym a komputerem PC, zatem nie ma możliwości np. wywołania funkcji pomiaru pojemności, czy generacji skryptu dla programu Matlab.

```

typedef enum
{
    GET_CONSTANTS = 'A',      // Get Constants Values
    SET_CONSTANTS = 'S',      // Set Constants Values
    DEF_CONSTANTS = 'D',      // Set Default Constants Values
    GET_CORRECTIONS = 'I',    // Get Corrections Values
    SET_CORRECTIONS = 'O',    // Set Corrections Values
    DEF_CORRECTIONS = 'P',    // Set Default Corrections Values
    SET_L_VOUT = 'L',         // Set Signal Pin as LOW
    SET_H_VOUT = 'H',         // Set Signal Pin as HIGH
    SET_GENERATIONS = 'G',   // Set Generations on Signal Pin
    SET_NOMINAL = 'N',        // Set Signal Pin as NOMINAL (HIGH-Z)
    TRIGGER_MEAS = 'M',       // Trigger Measurement
    GET_TEMP_RH = 'R',        // Read Temperature and Humidity
    LCD_AFTERMEAS = 'W',      // Send Calculated Data To Display on LCD
} COMMAND_t;

```

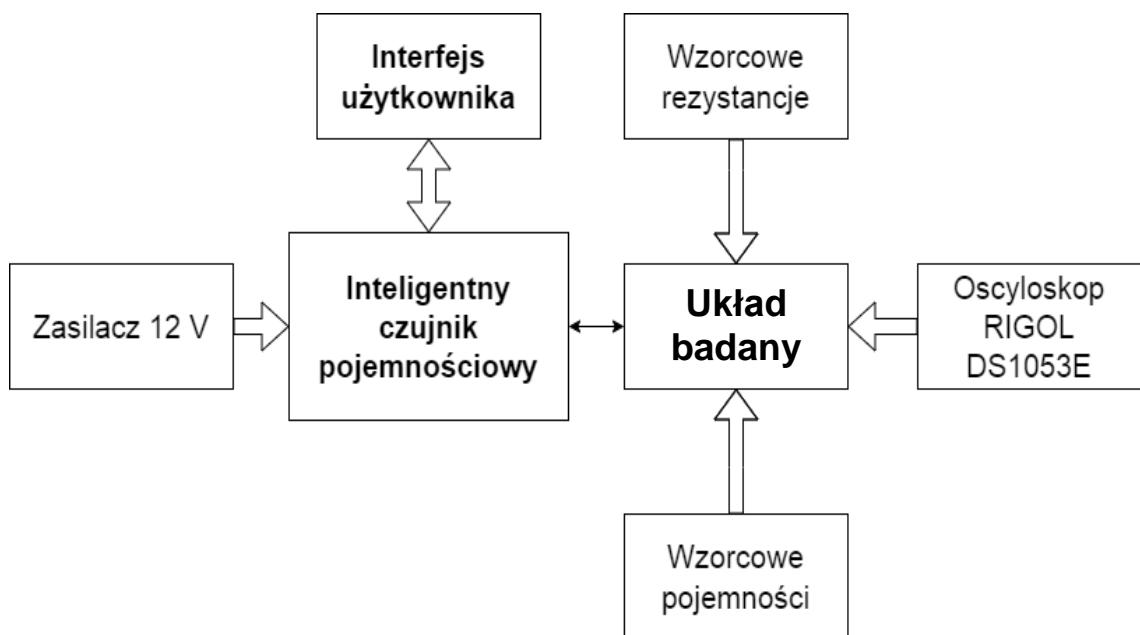
Listing 7.6. Zestaw instrukcji interpretowanych przez układ laboratoryjny.

8. WERYFIKACJA EKSPERYMENTALNA

W rozdziale zostanie przedstawiony proces kalibracji urządzenia pomiarowego. Zostaną wyznaczone eksperymentalne charakterystyki rzeczywiste układu pomiarowego. Szczególną uwagę poświęcono na omówienie błędów wynikających z nieidealności układu pomiarowego oraz na prezentację wyznaczania charakterystyki korygującej pomiar pojemności.

8.1. Stanowisko pomiarowe

Stanowisko pomiarowe pokazane na rys. 8.1. służy do pomiaru parametrów rzeczywistego układu laboratoryjnego. W celu wyznaczenia błędów wprowadzanych przez tor pomiarowy czujniki pojemnościowe zostały zastąpione zestawem pojemności wzorcowych. Uzyskane wyniki posłużyły do korekcji charakterystyki czujników. Ponadto porównano współczynniki korekcji otrzymane drogą regresji liniowej oraz interpolacji Lagrange'a 4 stopnia.



Rys. 8.1. Diagram stanowiska pomiarowego.

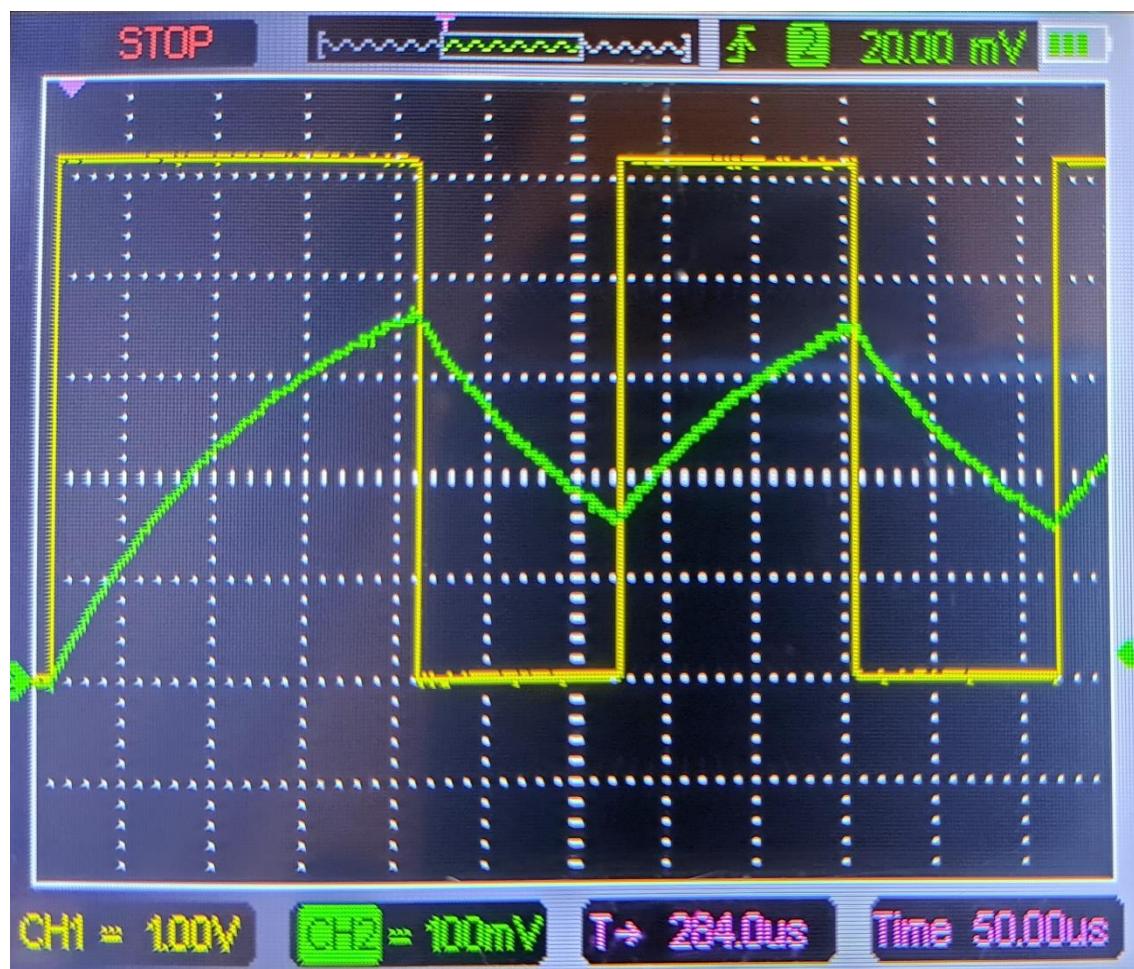
Za pomocą oscyloskopu RIGO DS1053E zaobserwowano charakterystykę przejściową podczas procedury pomiarowej. Dzięki temu doświadczalnie można określić opóźnienie jakie jest wprowadzane przez system przerwań mikrokontrolera. Sondę oscyloskopową podłączono do złącz diagnostycznych. Zastosowano tryb „x10” sondy oscyloskopowej, co znacząco zmniejsza pojemności wprowadzane przez nią do układu pomiarowego.

Całym systemem pomiarowym steruje aplikacja interfejsu użytkownika. System jest zasilany z zewnętrznego zasilacza 12 V, który zapewnia należytą stabilność wykonywanych pomiarów.

Do weryfikacji poprawności pracy układu pomiarowego (rys. 8.2.) użyto oscyloskopu dołączonego do układu badanego następująco: kanał pierwszy dołączono do sygnału pobudzającego układ pomiarowy (linia nr 11 modułu Arduino Micro), a kanał drugi do linii pobudzającej czujnik HS1101 (V_CAP) w celu obserwacji napięcia na układzie RC.

Mimo zastosowania trybu „x10” sondy oscyloskopowej wniosła ona pewną pojemność pasożytniczą, dlatego pomiar wymaga dodatkowej kalibracji. Obecne ustawienia progów referencyjnych podczas pomiaru wynosiły 3,33 V (H_THR) oraz 1,67 V (L_THR). Przekroczenie jednego z tych progów aktywuje przerwanie mikrokontrolera, który przełącza stan na wyjściu pobudzającym na przeciwny.

Pomiar czasu ładowania i rozładowania odbywa się pomiędzy przełączeniami stanów sygnału pobudzającego. W ramach jednego cyklu pomiarowego mierzonych jest 128 próbek czasu w trakcie ładowania i rozładowania układu RC. Oscylogramy umożliwiają także ocenę nabylenia zbocz sygnału wyjściowego z bufora trójstanowego.



Rys. 8.2. Przebieg ładowania i rozładowania rzeczywistego układu pomiarowego.

8.2. Pomiar pojemności wzorcowych

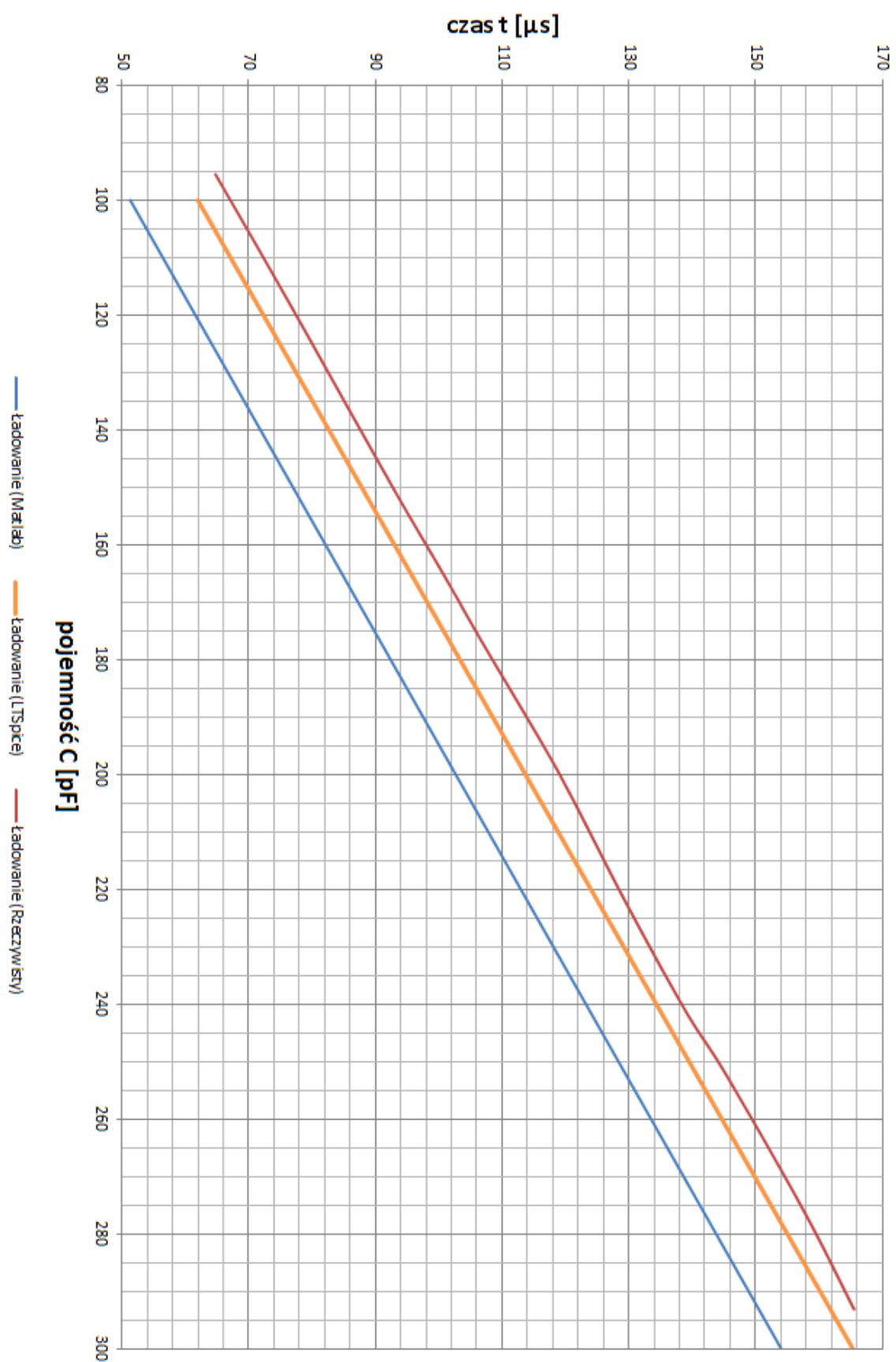
Wartości kondensatorów wzorcowych zmierzono za pomocą przyrządów: Agilent E4980A Precision LCR Meter oraz Agilent 16047A Test Fixture. Użyto kondensatorów posrebrzanych mikowych, które cechują takie parametry jak: dokładne wartości pojemności rzędu pojedynczych pF oraz stabilność pojemności w szerokim zakresie temperatury i napięcia zasilającego.

Zakres wartości pojemności zawiera się od 100 do 300 pF (tabela 8.1.). Dla każdego kondensatora wzorcowego umieszczonego w układzie badanym przeprowadzono 100 pomiarów, uzyskując tym samym dokładny wynik pojemności kondensatora. Każdy pomiar wielkości został poprzedzony po kilku minutach pracy kondensatora w obwodzie badanym, co pozwoliło na ustabilizowanie jego pracy.

Tabela 8.1. Wyniki pomiaru pojemności elementów wzorcowych [$R = 741,2 \text{ k}\Omega$].

Pojemność wzorcowa [pF]	Pojemność zmierzona [pF]	Błąd bezwzględny [pF]	Błąd względny [%]
95,487	126,183	30,696	32,147
116,656	147,601	30,945	26,526
149,561	180,222	30,662	20,501
165,738	196,822	31,084	18,755
177,278	208,429	31,149	17,570
199,371	231,355	31,984	16,042
220,511	250,606	30,096	13,648
240,784	270,195	29,411	12,215
252,610	283,432	30,822	12,202
276,201	307,149	30,948	11,205
293,120	322,389	29,269	9,985

Błędy bezwzględne pomiaru pojemności powstałe na skutek nieidealności układu laboratoryjnego wynoszą około 30 pF w całym zakresie pomiarowym. Wartość ta wynika z pojemności pasożytniczych występujących w rzeczywistym układzie laboratoryjnym. Stąd celem kalibracji jest wyeliminowanie wpływu elementów pasożytniczych. Czułość układu laboratoryjnego pozwala na pomiar nawet pojedynczych pF, dzięki stabilnej wartości błędów w szerokim zakresie badanych wartości pojemności.



Rys. 8.3. Porównanie charakterystyk czasów ładowania różnych modeli [$R = 741,2 \text{ k}\Omega$].

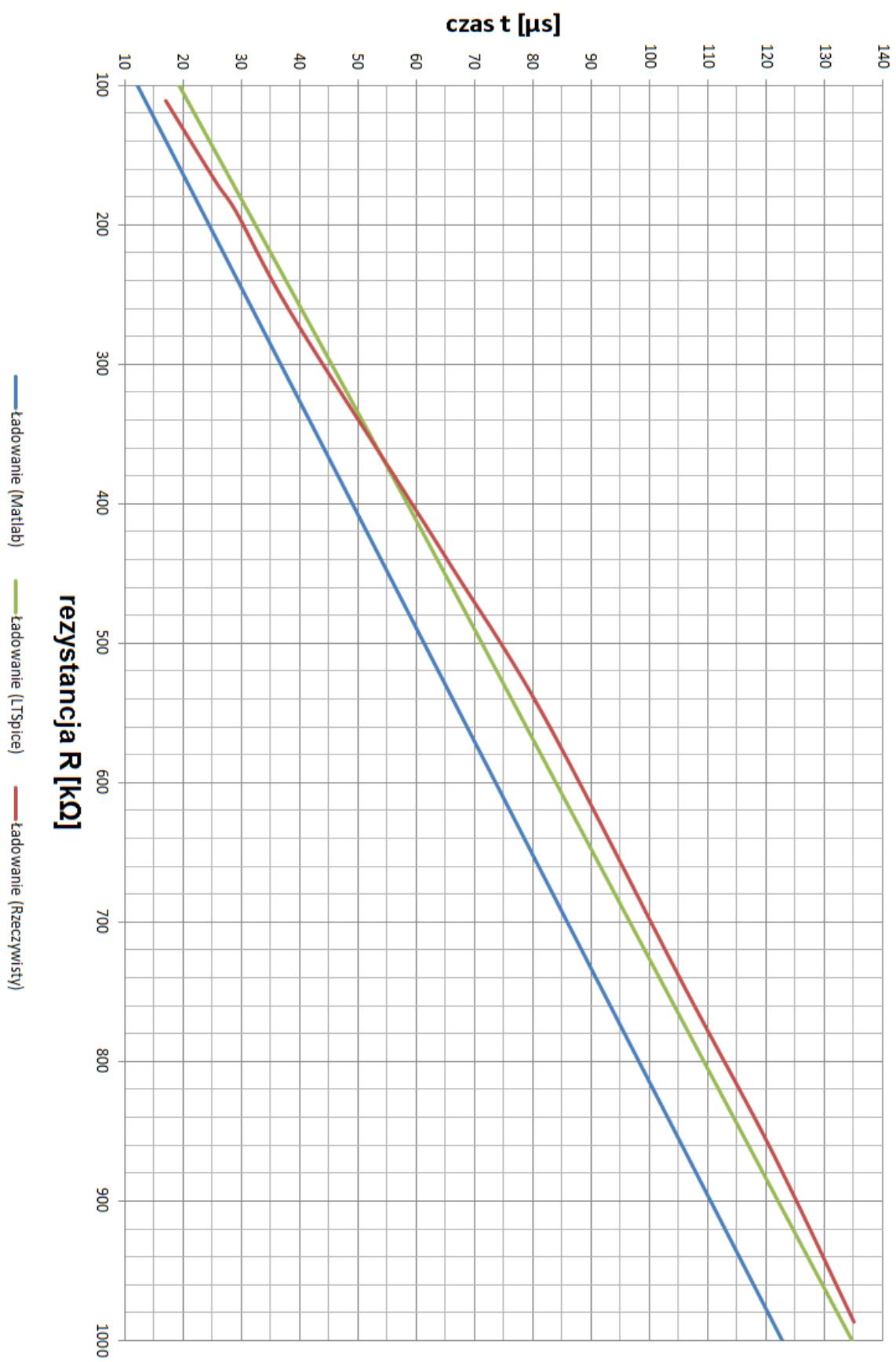
Jak pokazano na rys. 8.3. charakterystyki czasów ładowania układu RC w funkcji wartości pojemności mierzonej są w przybliżeniu liniowe. Ponadto, okazuje się, że oszacowane wartości pojemności pasożytniczych uwzględnionych w modelu symulacyjnym są zbyt optymistyczne. Układ rzeczywisty charakteryzuje się znacznie większymi wartościami pojemności pasożytniczych, gdyż nie wszystkie te pojemności da się uwzględnić w modelu symulacyjnym.

Z użyciem zestawu rezystancji wzorcowych (tabela 8.2.) została wyznaczona charakterystyka czasów ładowania pojemności wzorcowej o wartości 177,2 pF. Pokazano ją na rys. 8.4. Z rysunku widać, że zachodzi nieliniowa zależność pomiędzy zmianami rezystancji pomiarowej a czasami ładowania przy szacowanej wartości pojemności. Stąd można wywnioskować, że model układu badanego powinien być znacznie bardziej rozbudowany, niż przyjęto podczas symulacji.

Czyli należy pamiętać, że każdorazowa zmiana rezystora pomiarowego wymusza na użytkowniku ponowne wykonanie kalibracji urządzenia. W tym celu został opracowany skrypt dla programu Matlab wyznaczający współczynniki korekcji regresji liniowej oraz interpolacji Lagrange'a na podstawie danych pomiarowych. Dodatkowo wyznacza on charakterystykę służącą do korekcji punktów pomiarowych w zakresie od 100 do 300 pF, ponownie wykres błędów względnych, bezwzględnych oraz losowych na podstawie tych punktów pomiarowych.

Tabela 8.2. Wpływ rezystancji na czasy ładowania i rozładowania pojemności 177,2 pF.

Wartość rezystancji wzorcowej [kΩ]	Zmierzony czas ładowania [μs]	Zmierzony czas rozładowania [μs]
110,9	17,13	16,70
168,5	25,64	25,65
191,2	29,30	29,02
260,9	38,27	38,20
385,1	57,10	57,63
446	66,30	66,15
552,6	82,04	81,96
741,2	105,43	105,17
866,4	121,34	121,12
986,7	135,15	135,07



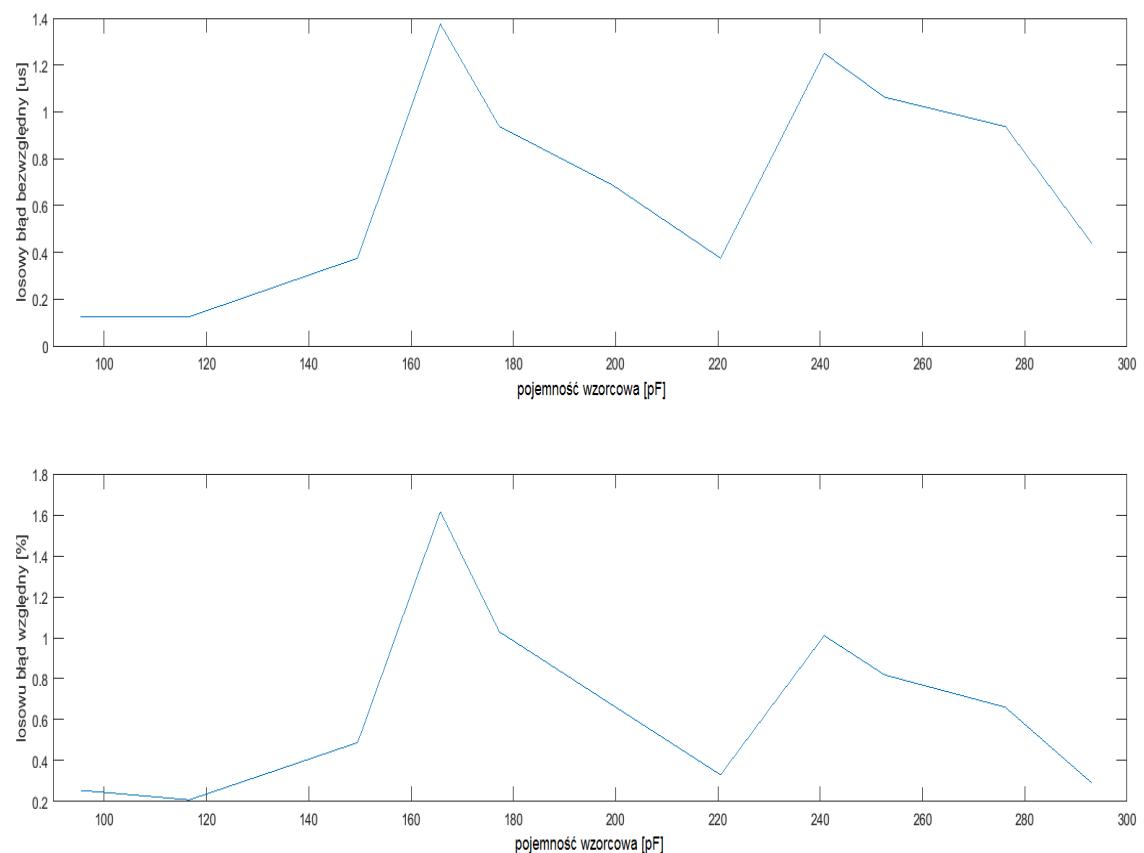
Rys. 8.4. Porównanie charakterystyk czasów ładowania pojemności 177,2 pF.

Błędy losowe powstałe na skutek między innymi oddziaływania czynników zewnętrznych pokazano na rys. 8.5. Dodatkowe ograniczenie pod wpływem zjawisk zewnętrznych mogłoby zapewnić ekranowana obudowa chroniąca np. doprowadzenia układu przed zewnętrznymi zakłóceniami elektromagnetycznymi.

Na błąd losowy może wpływać również niedokładność obsługi przerwania wykonywanego przez mikrokontroler ATmega32U4 oraz zbyt długi jej czas obsługi. Maksymalny błąd losowy sięga około $1,4 \mu\text{s}$, co równoważne jest około 22 cykli zegara taktującego mikrokontroler.

Błędy losowe zostały wyznaczone na podstawie serii składającej się ze 100 cykłów pomiarowych w krótkim odstępie czasowym dla każdej ze wzorcowych wartości pojemności. Nie zauważono znaczącego wpływu błędów losowych na wyniki pomiarowe.

Podczas każdego z cykłów pomiarowych 128 razy pomierzono wartości czasu ładowania oraz rozładowania. Błędy losowe wyznaczono korzystając z różnicy pomiędzy największą, a najmniejszą wartością w zbiorze. Następnie otrzymane tak 100 wartości z każdej serii zostały uśrednione. Analizę danych wykonano za pomocą programu Matlab.



Rys. 8.5. Wykresy błędów losowych.

8.3. Korekcja pomiarów

W podrozdziale zostaną zaprezentowane metody korekcji wyników pomiarowych. Szczególną uwagę poświęcono prezentacji skryptu dla programu Matlab służącego do wyznaczania współczynników korekcji. Omówiono również wykresy błędów pomiarowych.

8.3.1. Algorytm regresji liniowej

Regresja liniowa służy do aproksymacji zbioru danych pomiarowych za pomocą funkcji pierwszego stopnia $y = a \cdot x + b$ [20]. Współczynniki wzoru określone są wzorami (8.1.) oraz (8.2.). Funkcja ta służy do korekcji danych pomiarowych związanych z pomiarem pojemności. Algorytm realizujący regresję liniową (listing 8.1.) został zaimplementowany w skrypt dla programu Matlab.

$$a = \frac{\sum_{i=1}^n (y_i \cdot (x_i - \bar{x}))}{\sum_{i=1}^n (x_i - \bar{x})^2} \quad (8.1.)$$

$$b = \bar{y} - a \cdot \bar{x} \quad (8.2.)$$

gdzie:

- a, b – współczynniki regresji liniowej,
- \bar{x} – średnia wartość zbioru pojemności zmierzonych,
- \bar{y} – średnia wartość zbioru pojemności wzorcowych,
- x – zbiór pojemności zmierzonych,
- y – zbiór pojemności wzorcowych.

```
function [a, b, Y] = LinearRegression(x, y)
    x_avg = mean(x);
    y_avg = mean(y);
    a = sum(y .* (x - x_avg)) / sum((x - x_avg).^2);
    b = y_avg - x_avg * a;
    Y = a * x + b;
end
```

Listing 8.1. Funkcja realizująca algorytmu regresji liniowej.

Maksymalna uzyskana niepewność pomiaru pojemności po korekcji za pomocą funkcji liniowej wyniosła 0,678% w zakresie od 100 do 300 pF (tabela 8.3.). Oznacza to, że rozdzielcość pomiaru wilgotności względnej wynosi aż 3%. Aby zwiększyć jeszcze dokładność pomiarową użyto algorytmu interpolacji Lagrange'a.

Tabela 8.3. Korekcja z wykorzystaniem regresji liniowej.

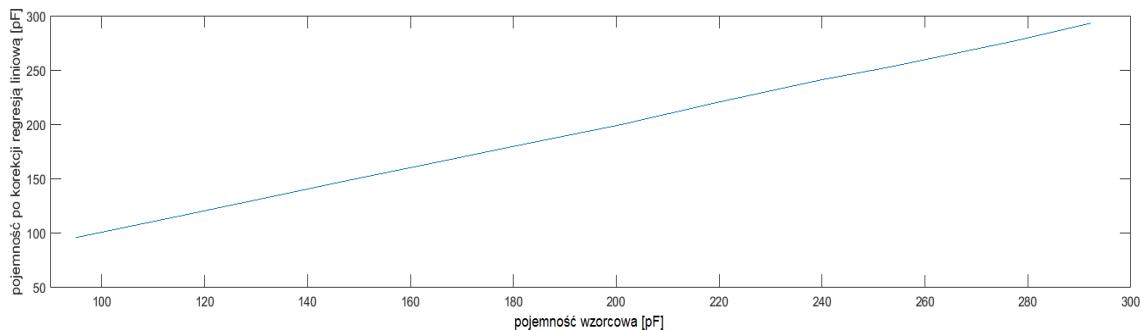
Pojemność wzorcowa [pF]	Pojemność po korekcji [pF]	Błąd bezwzględny [pF]	Błąd względny [%]
95,487	95,013	-0,474	-0,497
116,656	116,540	-0,117	-0,100
149,561	149,327	-0,233	-0,156
165,738	166,013	0,275	0,166
177,278	177,679	0,399	0,225
199,371	200,722	1,351	0,678
220,511	220,072	-0,438	-0,199
240,784	239,761	-1,024	-0,425
252,610	253,066	0,456	0,181
276,201	276,905	0,704	0,255
293,120	292,222	-0,899	-0,307

Wzór korygujący wartość pojemności zmierzonej przedstawia formuła (8.3.). Algorytm go realizujący zastosowano dla wszystkich zmierzonych pojemności.

$$C_{korekcja} = 1,0051 \text{ [pF/pF]} \cdot C_{zmierzzone} - 31,816 \text{ [pF]} \quad (8.3.)$$

gdzie:

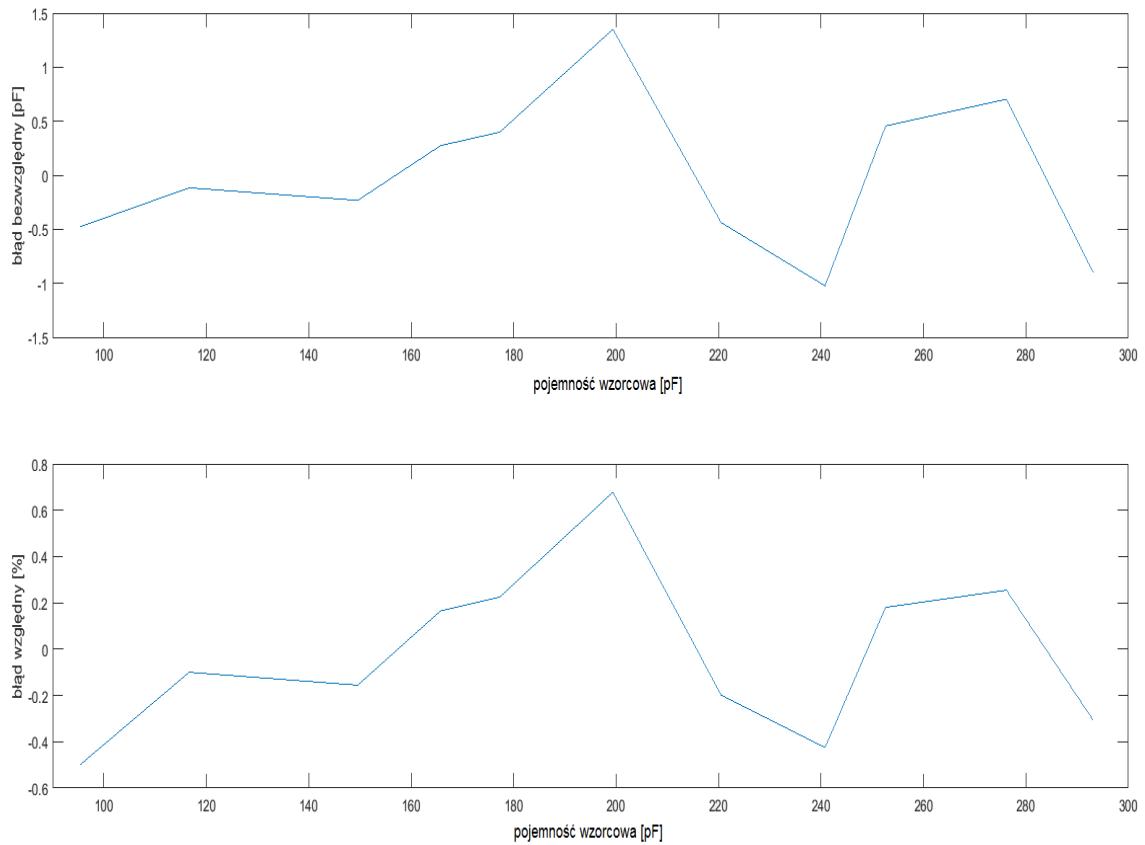
- $C_{korekcja}$ – wartość pojemności po korekcji [pF],
- $C_{zmierzzone}$ – wartość pojemności zmierzonej [pF].



Rys. 8.6. Wykres korekcji pojemności wielomianem pierwszego stopnia.

Aproksymacja wielomianem pierwszego stopnia (rys. 8.6.) nie jest wymagająca obliczeniowo, stąd może zostać zaimplementowana w programie mikrokontrolera. Niestety nie jest zbyt dokładna jak pokazano na wykresach na rys. 8.7.

Błędy wyznaczenia pojemności po korekcji wielomianem pierwszego stopnia (rys. 8.7.) w zakresie małych wartości pojemności, wynoszą około 0,2%. Największe błędy zaobserwowano dla wartości pojemności wzorcowej wynoszącej 200 pF.



Rys. 8.7. Wykresy błędów po korekcji pojemności wielomianem 1-go stopnia.

8.3.2. Algorytm interpolacji Lagrange'a

Interpolacja Lagrange'a jest numeryczną metodą interpolacji funkcji za pomocą wielomianu n -tego stopnia w wybranych $n + 1$ punktach [21]. Oznacza to, że otrzymany wykres wielomianu przechodzi dokładnie przez te punkty.

Na potrzeby projektu skorzystano z interpolacji wielomianem 3 stopnia. Stąd skorzysta się z 4 punktów o współrzędnych wartości pojemności wzorcowej mieszczących się w zakresie pojemności czujnika HS1101 (161 – 193 pF) [8].

$$W_n(x) = y_0 \frac{(x - x_1)(x - x_2) \dots (x - x_n)}{(x_0 - x_1)(x_0 - x_2) \dots (x_0 - x_n)} + y_1 \frac{(x - x_0)(x - x_2) \dots (x - x_n)}{(x_1 - x_0)(x_1 - x_2) \dots (x_1 - x_n)} + \dots + \\ y_n \frac{(x - x_0)(x - x_1) \dots (x - x_{n-1})}{(x_n - x_0)(x_n - x_1) \dots (x_n - x_{n-1})} \quad (8.4.)$$

gdzie:

- $y_0 - y_n$ – zbiór wartości pojemności wzorcowych,
- $x_0 - x_n$ – zbiór wartości pojemności zmierzonych.

Algorytm realizujący interpolację Lagrange'a został zaimplementowany w skrypcie dla programu Matlab (listing 8.2.). Kod implementuje wzór (8.4.) oparty na 4 punktach pomiarowych. Zastosowanie wielomianu 3-go stopnia pozwala na dokładną korekcję charakterystyki w wąskim przedziale wartości pojemności czujnika HS1101 (161 – 193 pF). Wyższe stopnie wielomianu spowodują komplikację obliczeń lecz nie poprawią znacząco dokładności pomiarowej układu laboratoryjnego.

```
function Poly = LagrangeInterpolation(IntX, IntY)
    syms x;
    Int(x) = ...
        IntY(1) * (x - IntX(2))*(x - IntX(3))*(x - IntX(4)) / ...
        ((IntX(1) - IntX(2))*(IntX(1) - IntX(3))*(IntX(1) - IntX(4))) +
    IntY(2) * (x - IntX(1))*(x - IntX(3))*(x - IntX(4)) / ...
        ((IntX(2) - IntX(1))*(IntX(2) - IntX(3))*(IntX(2) - IntX(4))) +
    IntY(3) * (x - IntX(1))*(x - IntX(2))*(x - IntX(4)) / ...
        ((IntX(3) - IntX(1))*(IntX(3) - IntX(2))*(IntX(3) - IntX(4))) +
    IntY(4) * (x - IntX(1))*(x - IntX(2))*(x - IntX(3)) / ...
        ((IntX(4) - IntX(1))*(IntX(4) - IntX(2))*(IntX(4) - IntX(3)));
    Poly = sym2poly(Int);
end
```

Listing 8.2. Implementacja algorytmu opartego na interpolacji Lagrange'a.

Dzięki skorzystaniu z interpolacji Lagrange'a osiągnięto błąd względny mniejszy od 0,1%. Takiej dokładności pomiaru odpowiada rozdzielcość odczytu wilgotności względnej na poziomie 0,3% dla zakresu pojemności od 161 do 193 pF (tabela 8.4.). Dla większego przedziału wartości pojemności uzyskano wartości błędu względnego na poziomie 0,5%. Przy zastosowaniu innych czujników pojemnościowych o szerszym zakresie pomiarowym jest możliwość rekonfiguracji wzoru interpolacji poprzez dobór punktów pomiarowych, tak aby w zadanym zakresie otrzymać najmniejsze błędy.

Tabela 8.4. Wyniki po korekcji z wykorzystaniem interpolacji Lagrange'a.

Pojemność wzorcowa [pF]	Pojemność po korekcji [pF]	Błąd bezwzględny [pF]	Błąd względny [%]
95,487	94,016	-1,471	-1,541
116,656	116,656	0	0
149,561	149,560	0	0
165,738	165,838	0,100	0,060
177,278	177,124	-0,156	-0,088
199,371	199,371	0	0
220,511	218,208	-2,302	-1,044
240,784	237,737	-3,047	-1,266
252,610	251,244	-1,366	-0,541
276,201	276,300	0,099	0,036
293,120	293,120	0	0

Wyznaczony wielomian korekcyjny 3-go stopnia został przedstawiony w formule (8.5.).

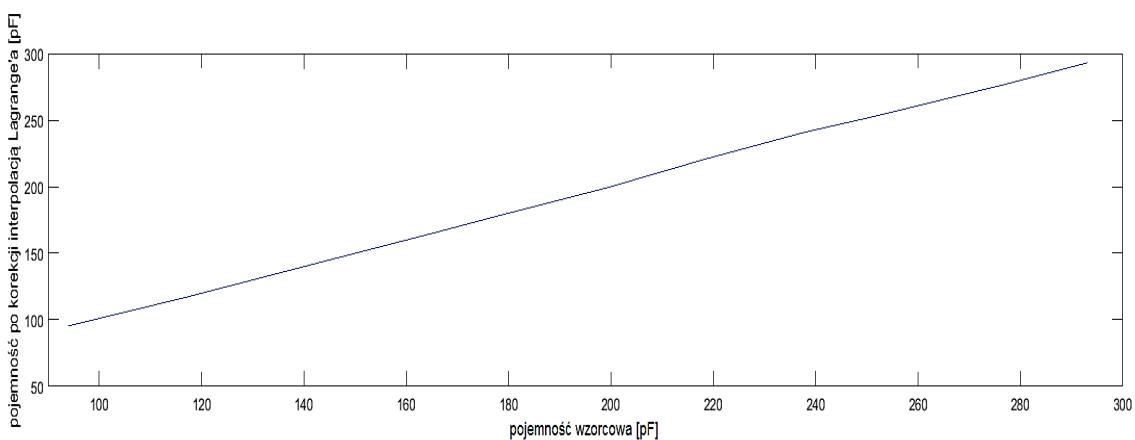
Do wyznaczenia współczynników wielomianu użyto cztery punkty pomiarowe (tabela 8.4.) o wartościach: 116,656 pF; 149,561 pF; 199,371 pF oraz 293,120 pF. Podane wartości wychodzą poza przedział czujnika HS1101, jednak aproksymacja charakterystyki pomiędzy punktami pomiarowymi w przypadku powyższych wartości powoduje najmniejsze błędy.

$$C_{korekcja} = 4,6 \cdot 10^{18} \cdot C_{zmierzone}^3 - 2,9 \cdot 10^9 \cdot C_{zmierzone}^2 + 1,6 \cdot C_{zmierzone} - 7,2 \cdot 10^{11} \quad (8.5.)$$

gdzie:

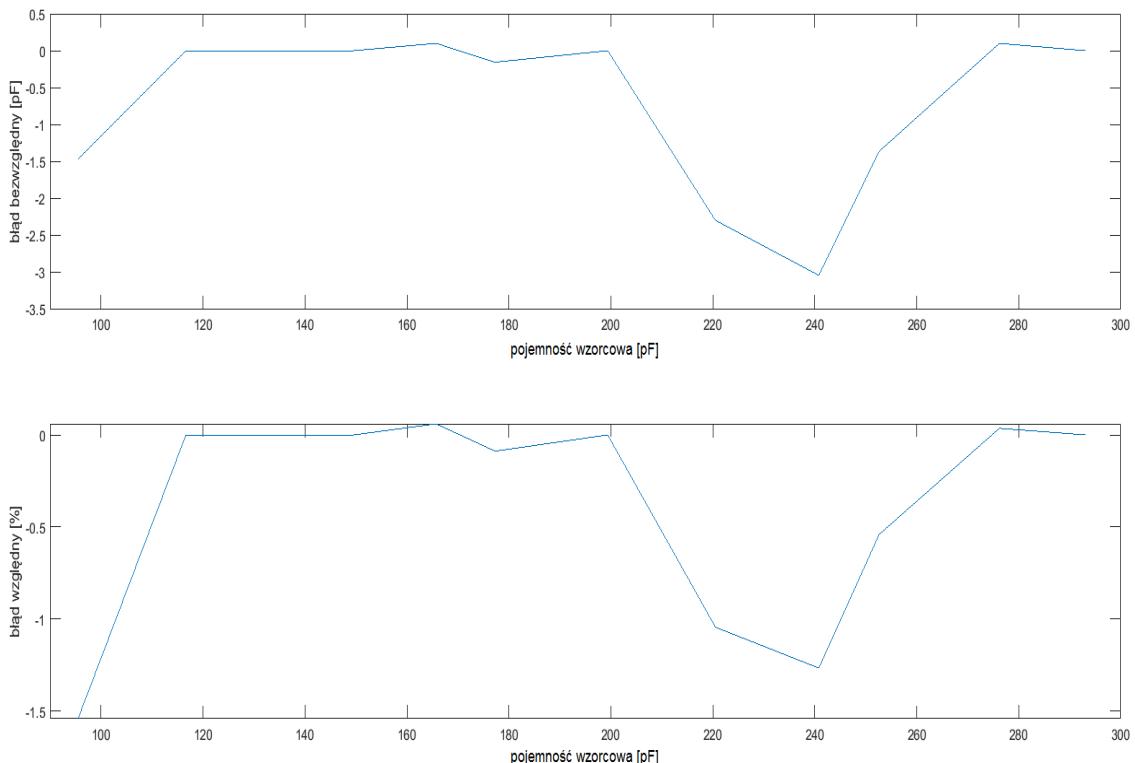
- $C_{korekcja}$ – wartość pojemności po korekcji [F],
- $C_{zmierzone}$ – wartość pojemności zmierzanej [F].

Korekcja oparta na interpolacji wielomianem 3-go stopnia (rys. 8.8.) jest wymagająca obliczeniowo, dlatego jej nie zaimplementowano w mikrokontrolerze. Aproksymacja ta zapewnia najlepszą korektę dla wartości pojemności dla czujnika HS1101. Należy pamiętać, że każdorazowa zmiana rezystora pomiarowego wymaga ponownej kalibracji układu pomiarowego.



Rys. 8.8. Wykres po korekcji pojemności wielomianem trzeciego stopnia.

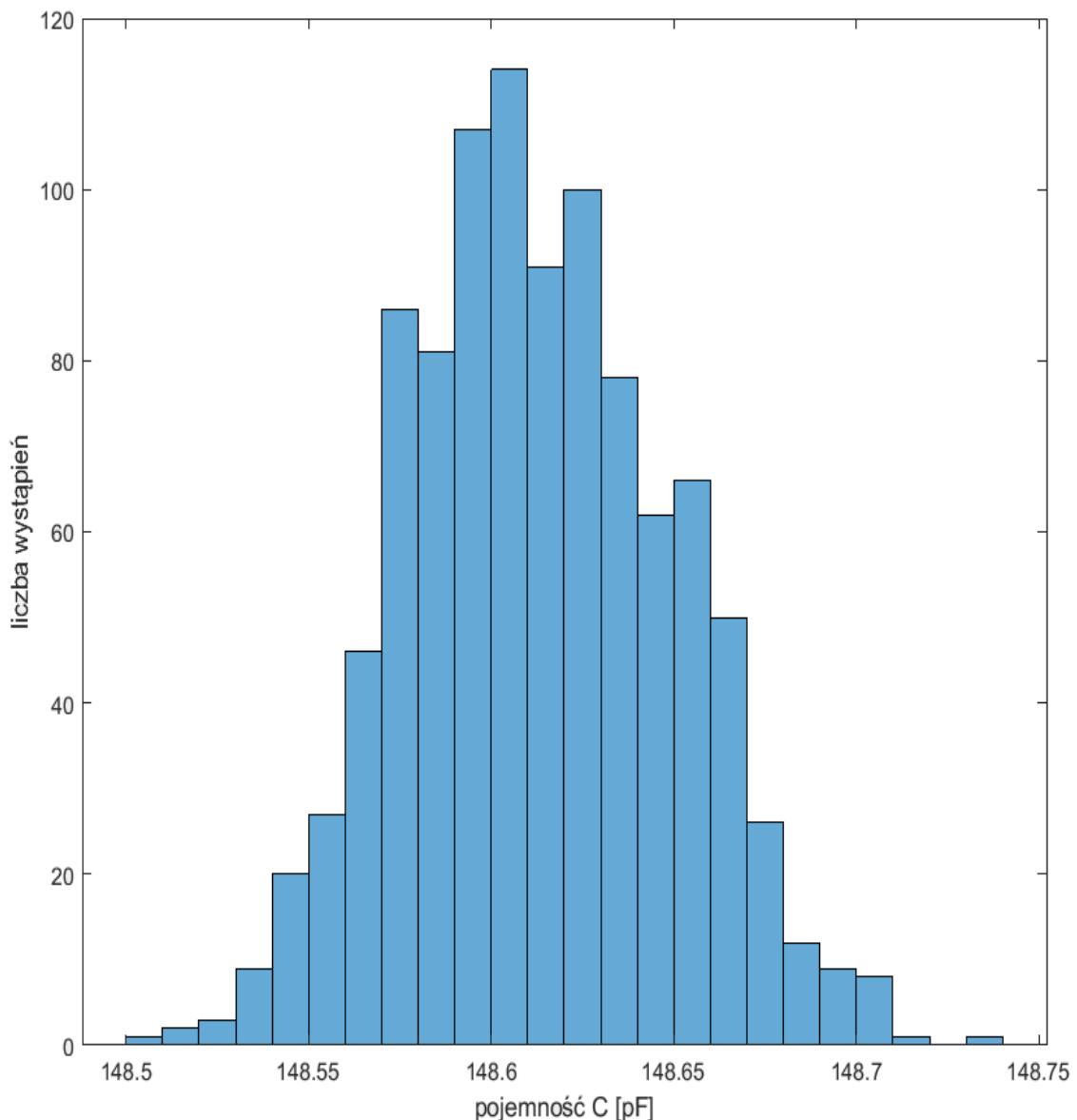
W przedziale od 115 do 200 pF skorygowane wartości pojemności zmierzonej są zbliżone do wartości wzorcowych (rys. 8.9.). Dla wartości większej niż 200 pF zaobserwowano wyraźny wzrost błędu do -1,2%. Podobne zjawisko zaobserwować można dla pojemności mniejszej od 100 pF. Pomiar tak niskich wartości pojemności obarczony jest błędem wynikającym z rozdzielczości licznika mikrokontrolera oraz zbyt długim czasem ładowania i rozładowania pojemności badanej podczas obsługi przerwania. Dlatego nie wzięto tego pomiaru pod uwagę podczas korekcji danych.



Rys. 8.9. wykresy błędów po korekcji wartości pojemności wielomianem 3-go stopnia.

Za pomocą histogramu możliwe jest graficzne przedstawienie rozkładu występowania poszczególnych wartości badanej pojemności. W celu jego wyznaczenia wykonano 1000 pomiarów dla wybranej w zbiorze wyników pomiarowych pojemności badanej 148,6 pF (rys. 8.10.). Pomiary były wykonane się co 0,5 s. Zgodnie z założeniami posiada on charakter zbliżony do rozkładu Gaussa.

Jednym z najważniejszych wniosków płynących z badań eksperymentalnych jest określenie dokładności pomiarowej, która wynosi 0,2 pF, co jest bardzo dobrym wynikiem. Odpowiada to rozdzielczości wilgotności względnej na poziomie 0,6% RH mierzonym w czasie ponad 10 minut. Natomiast znaczna większość wyników mieści się w granicach 148,58 – 148,64 pF.



Rys. 8.10. Histogram z serii 1000 pomiarów dla pojemności 148,6 pF.

9. PODSUMOWANIE

W niniejszej pracy dyplomowej zaprojektowano i zrealizowano układ inteligentnego sensora pojemnościowego bazującego na 8-bitowym mikrokontrolerze ATmega32U4. Wykorzystano wbudowany w mikrokontroler interfejs USB do komunikacji z interfejsem użytkownika. Napisano aplikację na komputer, która umożliwia kalibrację, analizę danych pomiarowych oraz monitorowanie wilgotności względnej w czasie rzeczywistym. Pomiar przeprowadzony jest czujnikiem pojemnościowym HS1101 jak i czujnikiem SHTC3. Dokonano także kalibracji wyników pomiarowych, dzięki której osiągnięto błąd względny na poziomie 0,1% dla zakresu mierzonych pojemności pokrywającego się ze wskazaniami czujnika HS1101, tym samym realizując wszystkie założone cele.

Część teoretyczna pracy zawiera przegląd metod pomiarowych przeznaczonych dla mikrokontrolerów. Omówiono wady i zalety każdej z metod oraz porównano wpływ parametrów na dokładność pomiarową. Szczególną uwagę poświęcono omówieniu modelu matematycznego idealnego układu pomiarowego, a także symulacji modelu z uwzględnieniem elementów pasożytniczych w programie LTSpice dla metody pomiaru ładowania/rozładowania układu RC. Okazało się, że model symulacyjny dokładnie odzwierciedla zjawiska występujące w rzeczywistym układzie pomiarowym.

Układ laboratoryjny wykonano tak, aby łatwo można było ustawać wartości napięć referencyjnych oraz wymieniać rezystory pomiarowe. Jednym z ograniczeń układu jest brak obudowy ekranującej. Stąd złącza diagnostyczne oraz układ pomiarowy są narażone na działanie zakłóceń zewnętrznych.

Pomimo to uzyskana dokładność pomiarowa jest zadowalająca, dokładność wyznaczania wilgotności względnej jest na poziomie 0,3% RH. Natomiast stabilność 10 minutowego pomiaru została doświadczalnie określona na poziomie 0,6% RH. Takie parametry charakteryzują dokładne urządzenia laboratoryjne. Dzięki wbudowanemu interfejsowi USB oraz dedykowanej aplikacji układ laboratoryjny może zostać wykorzystany do monitorowania wilgotności względnej w wielu środowiskach wymagających wysokiej stabilności RH. Natomiast stałe pojemności pasożytnicze wynoszące około 30 pF w całym zakresie badanych pojemności świadczą o poprawnie zaprojektowanym układzie pomiarowym. Dzięki łatwej zmianie zakresu pomiarowego możliwy jest również pomiar wartości pojemności rzędu pojedynczych pF. Świadczy to o dużej dokładności i stabilności układu laboratoryjnego.

WYKAZ LITERATURY

- [1] Meng Y., Dean R. N.: *A Technique for Improving the Linear Operating Range for a Relative Phase Delay Capacitive Sensor Interface Circuit.* IEEE Transactions on Instrumentation and Measurement Vol. 65 (3), 2016
- [2] Czaja Z.: *A measurement method for capacitive sensors based on a versatile direct sensor-to-microcontroller interface circuit.* Measurement Vol. 155, 107547, 2020
<https://doi.org/10.1016/j.measurement.2020.107547>
- [3] Czaja Z.: *A measurement method for lossy capacitive relative humidity sensors based on a direct sensor-to-microcontroller interface circuit.* Measurement Vol. 170, 108702, 2021
<https://doi.org/10.1016/j.measurement.2020.108702>
- [4] Czaja Z.: *Measurement method for capacitive sensors for microcontrollers based on a phase shifter.* Measurement Vol. 192, 110890, 2022
<https://doi.org/10.1016/j.measurement.2022.110890>
- [5] Meng Y., Dean R. N.: *Improving the phase delay capacitive interface circuit technique using MOSFET switches.* Measurement Science and Technology Vol. 31, 025107, 2019
<https://doi.org/10.1088/1361-6501/ab4a66>
- [6] Nota katalogowa układu komparatora TLV3502AID (data dostępu: 1.10.2022 r.)
<https://www.ti.com/lit/ds/sbos507a/sbos507a.pdf>
- [7] Skiba A., Tiliouine H.: *Stany nieustalone w obwodach elektrycznych. Przykłady i zadania.* Wyd. Politechniki Gdańskiej, Gdańsk 2022.
- [8] Nota katalogowa czujnika wilgotności HS1101 (data dostępu: 1.10.2022 r.)
<https://www.jameco.com/Jameco/Products/ProdDS/2082901.pdf>
- [9] Nota katalogowa mikrokontrolera ATmega32U4 (data dostępu: 1.10.2022 r.)
https://ww1.microchip.com/downloads/en/devicedoc/atmel-7766-8-bit-avr-atmega16u4-32u4_datasheet.pdf
- [10] Nota katalogowa stabilizatora monolitycznego 7805 (data dostępu: 1.10.2022 r.)
<https://www.ti.com/lit/ds/symlink/lm340.pdf>
- [11] Nota katalogowa modułu Arduino Micro (data dostępu: 1.10.2022 r.)
<https://docs.arduino.cc/hardware/micro>
- [12] Sklep internetowy z modułami elektronicznymi (data dostępu: 1.10.2022 r.)
<https://botland.com.pl/>
- [13] Nota katalogowa modułu SHTC3 (data dostępu: 1.10.2022 r.)
https://sensirion.com/media/documents/643F9C8E/6164081E/Sensirion_Humidity_Sensors_SHTC3_Datasheet.pdf
- [14] Nota katalogowa bufora trójstanowego, 74AHC244 (data dostępu: 1.10.2022 r.)
<https://www.ti.com/lit/ds/symlink/sn54ahc244-sp.pdf>
- [15] Nota katalogowa biblioteki LUFA USB (wersja oprogramowania: 210130)
<http://www.fourwalledcubicle.com/LUFA.php>

- [16] Kardaś M.: *Mikrokontrolery AVR. Język C. Podstawy programowania*. Wyd. ATNEL, Szczecin 2011.
- [17] Baranowski R.: *Mikrokontrolery AVR ATmega w praktyce*. Wyd. BTC, Warszawa 2005.
- [18] Sells C.: *Windows Forms Programming in C#*. Wyd. Addison-Wesley Professional, 2003.
- [19] Nota aplikacyjna algorytmu nadpróbkowania, *Microchip* (data dostępu: 1.10.2022 r.)
<https://ww1.microchip.com/downloads/en/appnotes/doc8003.pdf>
- [20] Analiza algorytmu regresji liniowej, *Matlab* (data dostępu: 1.10.2022r.)
https://www.mathworks.com/help/matlab/data_analysis/linear-regression.html
- [21] Analiza algorytmu interpolacji Lagrange'a, *Matlab* (data dostępu: 1.10.2022r.)
<https://www.mathworks.com/help/matlab/interpolation.html>

WYKAZ RYSUNKÓW

Rys. 2.1. Schemat ideowy układu pomiarowego przesuwnika fazowego [1].....	9
Rys. 2.2. Schemat blokowy układu dla metody pomiaru przesunięcia fazowego [2].....	10
Rys. 2.3. Schemat blokowy układu dla metody pomiarowej bazującej na pomiarze czasu ładowania/rozładowania układu RC	11
Rys. 2.4. Charakterystyka błędu względnego w zależności przesunięcia fazowego [5].....	12
Rys. 2.5. Sygnały występujące w metodzie opartej na pomiarze przesunięcia fazowego [2]. ...	13
Rys. 3.1. Obwód pomiarowy.....	14
Rys. 3.2. Analiza operatorowa obwodu pomiarowego.	14
Rys. 3.3. Przekształcony obwód pomiarowy w dziedzinie operatorowej.	15
Rys. 3.4. Schemat bloku pomiarowego – symulacja.....	16
Rys. 3.5. Schemat bloku porównywania napięć progowych – symulacja.	17
Rys. 3.6. Schemat podłączenia mikrokontrolera – symulacja.....	18
Rys. 3.7. Schemat bufora trójstanowego – symulacja.	19
Rys. 3.8. Charakterystyka czasu ładowania i rozładowania dla rezystancji $R = 741,2 \text{ k}\Omega$	20
Rys. 3.9. Charakterystyka czasu ładowania i rozładowania dla pojemności $C = 177,2 \text{ pF}$	21
Rys. 3.10. Charakterystyka sygnałów w czasie pomiaru, $C = 177,2 \text{ pF}$, $R = 741,2 \text{ k}\Omega$	22
Rys. 4.1. Schemat blokowy stanowiska pomiarowego.	23
Rys. 4.2. Schemat blokowy układu laboratoryjnego.	24
Rys. 4.3. Schemat ideowy układu zasilającego.	25
Rys. 4.4. Moduł Arduino Micro [12].	26
Rys. 4.5. Schemat ideowy podłączenia modułu Arduino Micro.	27
Rys. 4.6. Moduł SHTC3 [12].....	28
Rys. 4.7. Moduł wyświetlacza alfanumerycznego LCD [12].....	29
Rys. 4.8. Schemat ideowy bufora trójstanowego.	29
Rys. 4.9. Schemat ideowy układu pomiarowego RC.	30
Rys. 4.10. Schemat ideowy toru napięć referencyjnych.	31
Rys. 4.11. Schemat ideowy toru przetwarzania sygnału pomiarowego.....	32
Rys. 4.12. Schemat ideowy złącz diagnostycznych.....	33
Rys. 4.13. Projekt mozaiki PCB – górnna strona.	34
Rys. 4.14. Projekt mozaiki PCB – dolna strona.	34
Rys. 4.15. Realizacja fizyczna układu laboratoryjnego.	35
Rys. 5.1. Rejestr statusowy mikrokontrolera ATmega32U4 [9].....	36
Rys. 5.2. Schemat blokowy kontrolera USB [9].....	37
Rys. 5.3. Schemat blokowy układu licznika w trybie przechwytywania zdarzeń [9].....	38
Rys. 5.4. Rejestr konfiguracyjny licznika 1 [9].....	39
Rys. 5.5. Rejestr masek przerwań licznika 1 [9].....	39
Rys. 5.6. Sposób podłączenia urządzeń w interfejsie TWI [9].....	40

Rys. 5.7. Typowa transmisja danych interfejsu TWI [9].	40
Rys. 5.8. Rejestr kontrolny interfejsu TWI [9].	40
Rys. 6.1. Algorytm pętli głównej programu mikrokontrolera.	43
Rys. 6.2. Algorytm prezentujący pomiar pojemności.	47
Rys. 7.1. Strona startowa interfejsu użytkownika.	49
Rys. 7.2. Ustawienia ogólne interfejsu użytkownika.	50
Rys. 7.3. Przykład charakterystyki wilgotności i temperatury w czasie.	51
Rys. 7.4. Kalibracja urządzenia z poziomu interfejsu użytkownika.	53
Rys. 7.5. Wywołanie pomiarów z poziomu interfejsu użytkownika.	54
Rys. 7.6. Podgląd charakterystyk w interfejsie użytkownika.	57
Rys. 7.7. Dziennik zdarzeń interfejsu użytkownika.	58
Rys. 8.1. Diagram stanowiska pomiarowego.	59
Rys. 8.2. Przebieg ładowania i rozładowania rzeczywistego układu pomiarowego.	60
Rys. 8.3. Porównanie charakterystyk czasów ładowania różnych modeli [$R = 741,2 \text{ k}\Omega$].	62
Rys. 8.4. Porównanie charakterystyk czasów ładowania pojemności 177,2 pF.	64
Rys. 8.5. Wykresy błędów losowych.	65
Rys. 8.6. Wykres korekcji pojemności wielomianem pierwszego stopnia.	67
Rys. 8.7. Wykresy błędów po korekcji pojemności wielomianem 1-go stopnia.	68
Rys. 8.8. Wykres po korekcji pojemności wielomianem trzeciego stopnia.	71
Rys. 8.9. wykresy błędów po korekcji wartości pojemności wielomianem 3-go stopnia.	71
Rys. 8.10. Histogram z serii 1000 pomiarów dla pojemności 148,6 pF.	72

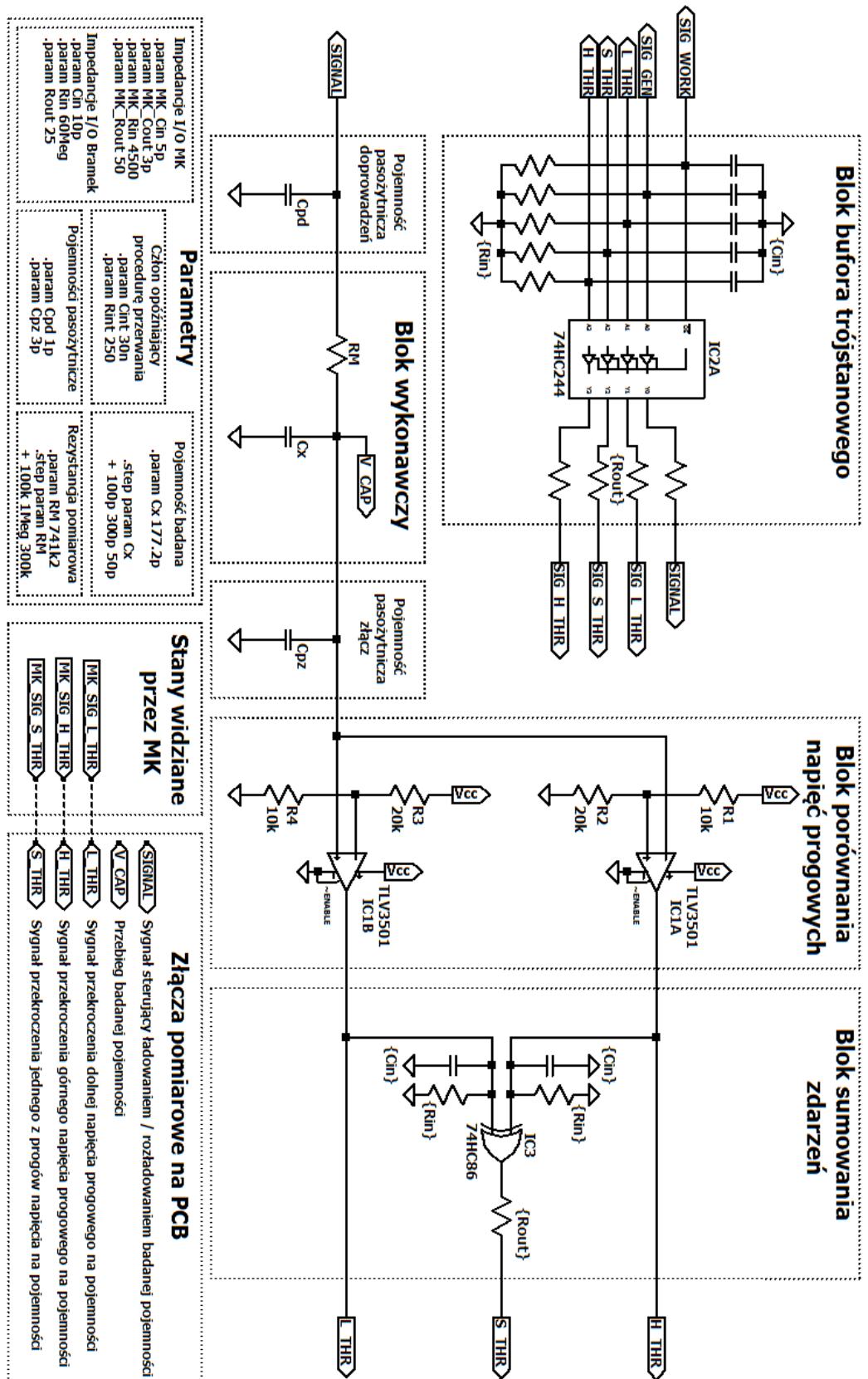
WYKAZ TABEL

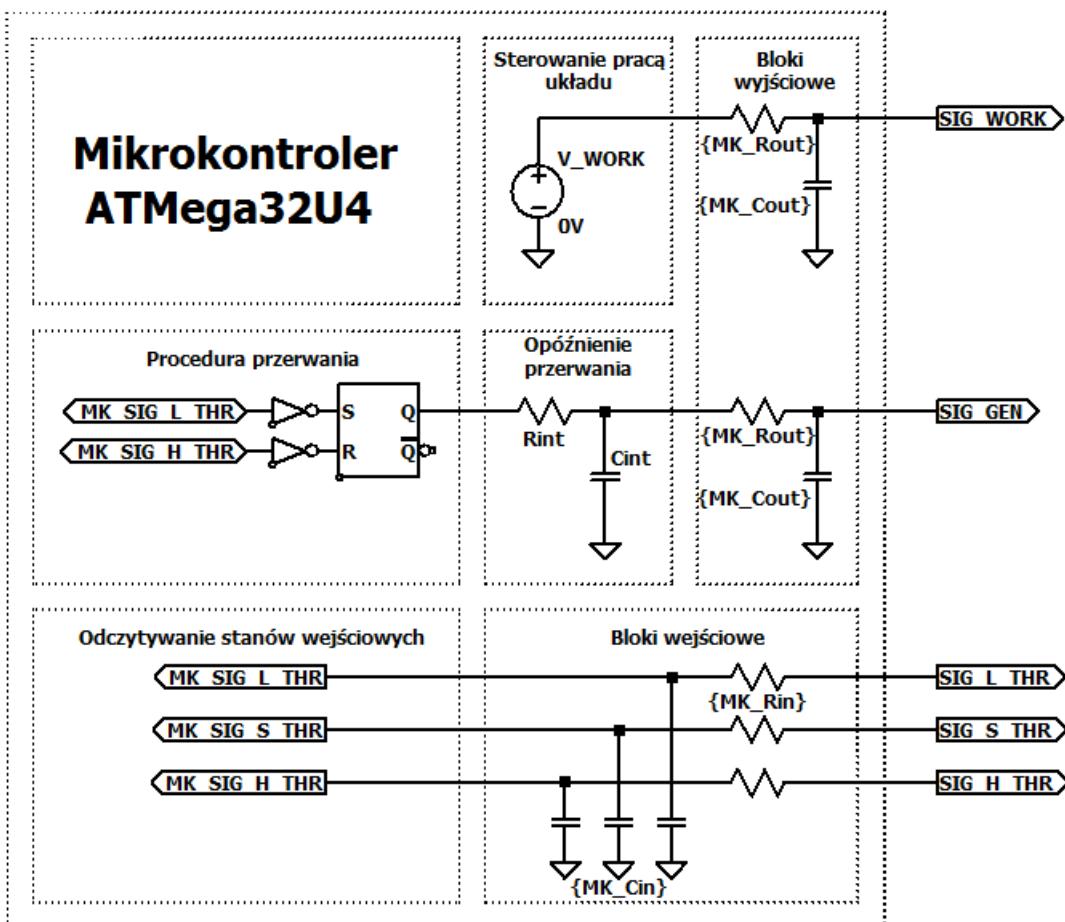
Tabela 8.1. Wyniki pomiaru pojemności elementów wzorcowych [$R = 741,2 \text{ k}\Omega$].	61
Tabela 8.2. Wpływ rezystancji na czasy ładowania i rozładowania pojemności 177,2 pF.	63
Tabela 8.3. Korekcja z wykorzystaniem regresji liniowej.	67
Tabela 8.4. Wyniki po korekcji z wykorzystaniem interpolacji Lagrange'a.....	70

WYKAZ LISTINGÓW

Listing 6.1. Funkcja inicjalizacji interfejsu USB	42
Listing 6.2. Funkcja inicjalizacji modułu przechwytywania	42
Listing 6.3. Obsługa instrukcji kalibrujących.	44
Listing 6.4. Definicja domyślnych wartości struktur danych kalibracyjnych.	45
Listing 6.5. Funkcja pomiaru pojemności.	48
Listing 6.6. Obsługa przerwania modułu przechwytywania zdarzeń zewnętrznych.	48
Listing 7.1. Przykład eksportu dziennika zdarzeń w formacie XML.	52
Listing 7.2. Funkcja wyznaczająca pojemność z danych pomiarowych.	54
Listing 7.3. Funkcja realizująca nadpróbkowanie i decymację.	55
Listing 7.4. Funkcja obliczająca pojemność.	55
Listing 7.5. Funkcja obliczająca wartość wilgotności względnej.	56
Listing 7.6. Zestaw instrukcji interpretowanych przez układ laboratoryjny.	58
Listing 8.1. Funkcja realizująca algorytmu regresji liniowej.	66
Listing 8.2. Implementacja algorytmu opartego na interpolacji Lagrange'a.	69

DODATEK A – SYMULACJA LTSPICE





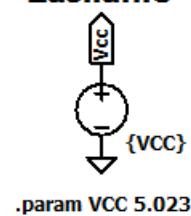
Biblioteki LTSpice

.inc 74HC.lib
.inc TLV3501.lib

Symulacje

```
.tran 0.5ms
.meas tran T1 find time when
+ V(MK_SIG_S_THR)=2.5 rise 1
.meas tran T2 find time when
+ V(MK_SIG_S_THR)=2.5 rise 2
.meas tran T3 find time when
+ V(MK_SIG_S_THR)=2.5 rise 3
.meas tran T4 find time when
+ V(MK_SIG_S_THR)=2.5 fall 2
```

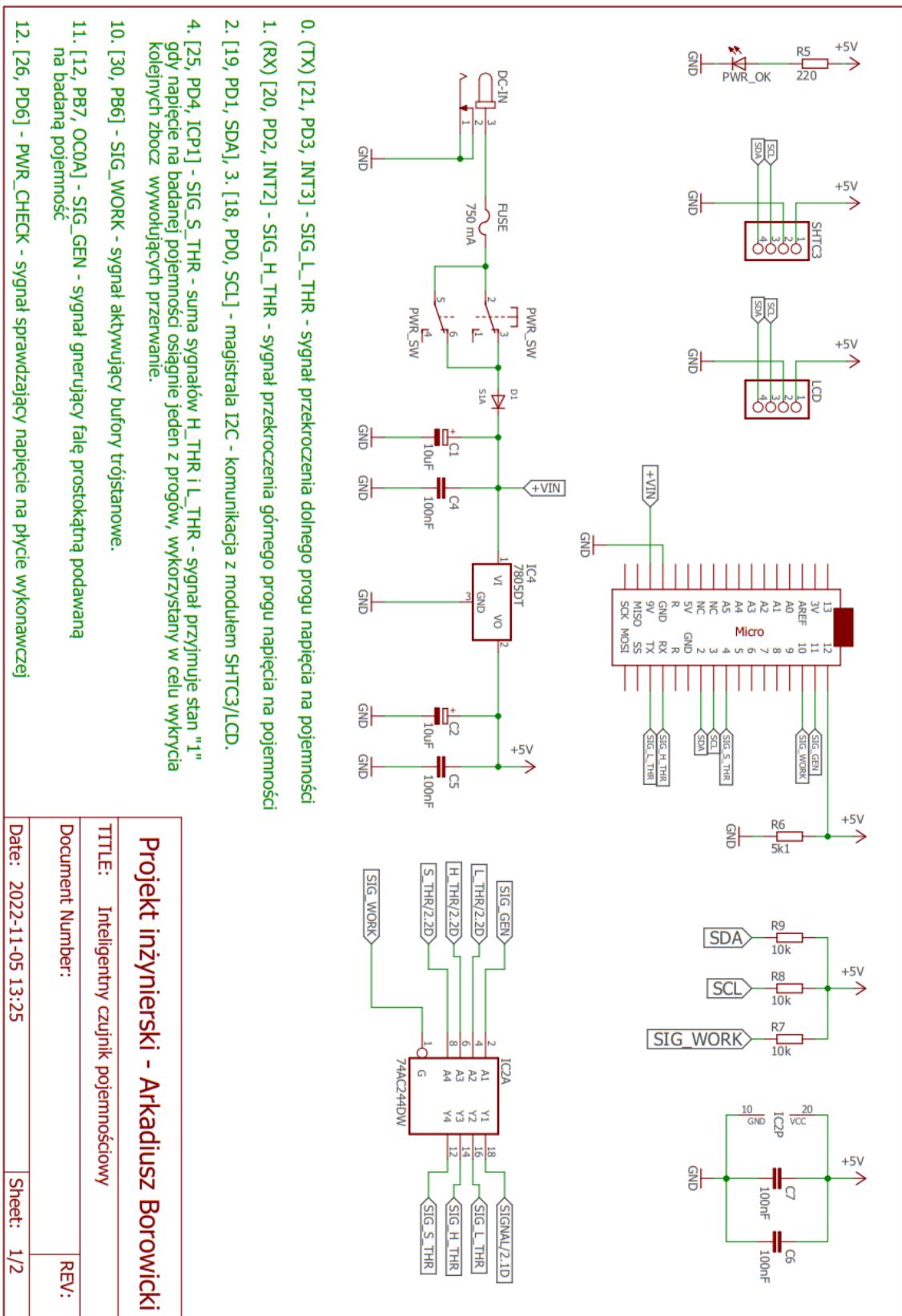
Zasilanie

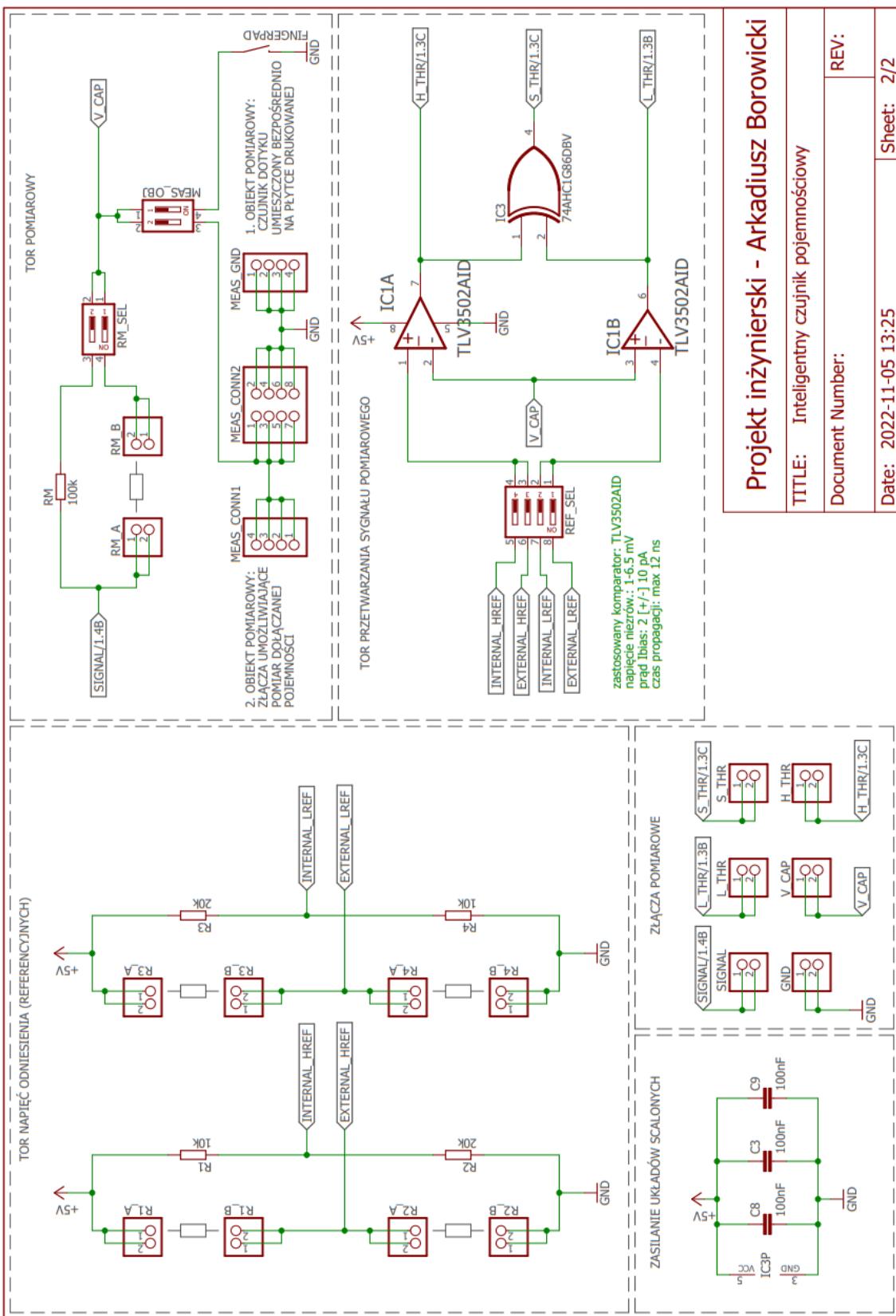


Pomiar zależności czasowych

.meas tran IntTime param T4-T1	Czas przerwania
.meas tran ChargingTime param (T3-T2)*1E6	Czas ładowania pojemności [us]
.meas tran ChargingTicks param (T3-T2)*16E6	Czas ładowania wyrażony w cyklach zegara procesora
.meas tran DischargingTime param (T2-T1)*1E6	Czas rozładowania pojemności [us]
.meas tran DischargingTicks param (T2-T1)*16E6	Czas rozładowania wyrażony w cyklach zegara procesora

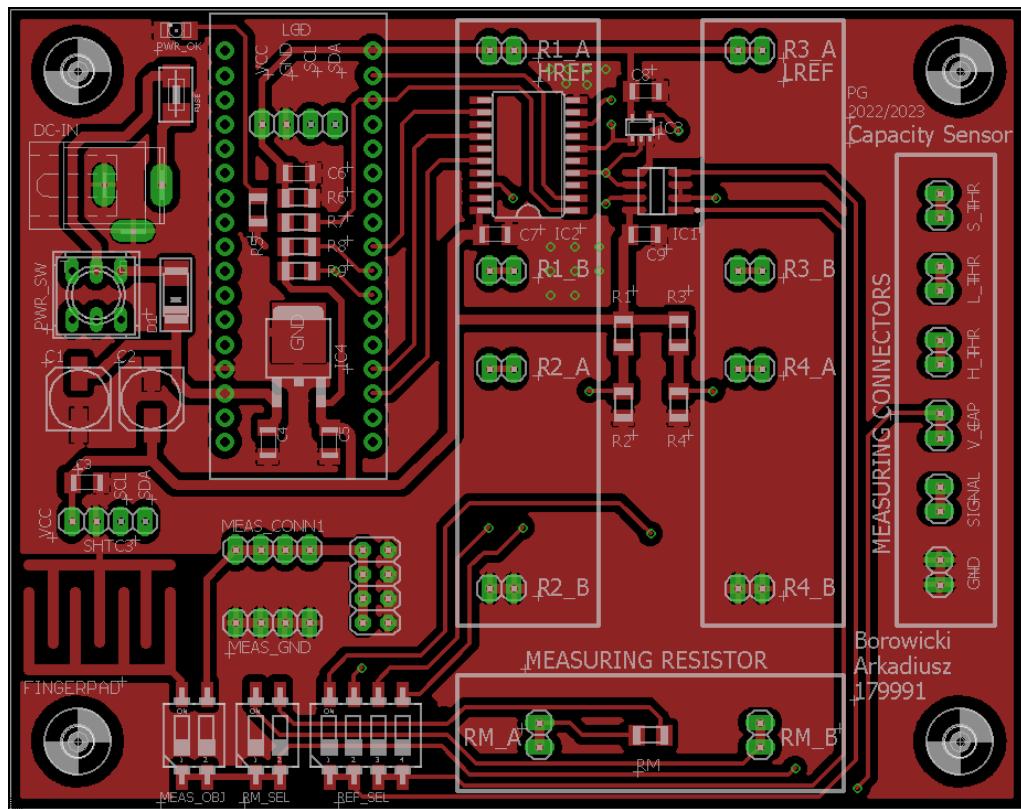
DODATEK B – SCHEMAT IDEOWY





DODATEK C – SCHEMAT MONTAŻOWY

Warstwa góra:



Warstwa dolna:

