



**POLITECHNIKA  
GDAŃSKA**

WYDZIAŁ ELEKTRONIKI,  
TELEKOMUNIKACJI I INFORMATYKI



Imię i nazwisko studenta: Damian Fryza

Nr albumu: 165578

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: Zestaw laboratoryjny do pomiarów elementów L i C oparty na bezpośrednim interfejsie czujnik-mikrokontroler i mikrokontrolerze PIC16F1789

Tytuł projektu w języku angielskim: A laboratory circuit for L and C measurements based on a direct sensor-to-microcontroller interface circuit and a PIC16F1789 microcontroller

Opiekun pracy: dr hab. inż. Zbigniew Czaja

Data ostatecznego zatwierdzenia raportu podobieństw w JSA: 21.12.2020



## **STRESZCZENIE**

W ramach pracy wykonano układ pomiarowy do pomiarów wartości elementów L i C. Omówiono sposoby pomiaru indukcyjności i pojemności za pomocą interfejsu czujnik-mikrokontroler. Na podstawie wybranej metody zaprojektowano i wykonano prototyp układu pomiarowego opartego na mikrokontrolerze PIC16F1789. Przedstawiono konstrukcję i rozwiązania użyte przy projekcie, a także strukturę i działanie oprogramowania. Następnie wykonano badania eksperymentalne mające na celu zweryfikowanie poprawności zaimplementowanej procedury pomiarowej.

### **Słowa kluczowe:**

czujnik, mikrokontroler, PIC16F1789, interfejs, pomiar, indukcyjność, pojemność

### **Dziedzina nauki i techniki, zgodnie z wymaganiami OECD:**

Nauki inżynierskie i techniczne, elektrotechnika i elektronika.

## **ABSTRACT**

In this paper, a measuring system for measuring the values of L and C elements was made. Methods of measuring inductance and capacitance using the sensor-microcontroller interface were discussed. Based on the evaluation of the design method, a prototype of a measuring system based on the PIC16F1789 microcontroller was made. The structures and solutions used in the projects as well as the problem and software are presented. Then, experimental tests were performed to verify the correctness of the implemented measurement procedure.

### **Keywords:**

sensor, microcontroller, PIC16F1789, interface, measurement, inductance, capacitance

## Spis treści

WYKAZ SKRÓTÓW.....	5
1. Wstęp.....	6
2. Przegląd metod pomiaru indukcyjności i pojemności opartych o interfejs czujnik- mikrokontroler.....	7
2.1. Metoda oparta na ADC i AC.....	7
2.2. Metoda oparta na liczniku – pomiar czasu ładowania czujnika.....	9
2.3. Metoda oparta na liczniku – pomiar czasu rozładowywania czujnika.....	13
2.4. Podsumowanie.....	15
3. Opis mikrokontrolera PIC16F1789 i jego najważniejszych peryferiów.....	16
3.1. Licznik.....	17
3.2. Oscylator.....	20
3.3. Blok EUSART.....	21
3.4. Komparator analogowy.....	23
3.5. DAC.....	25
3.6. System przerwań.....	26
4. Stanowisko laboratoryjne.....	27
4.1. Schemat blokowy.....	27
4.2. Schemat układu pomiarowego.....	28
4.2.1. Jednostka sterująca.....	29
4.2.2. Układ zasilania.....	29
4.2.3. Gniazdo RJ-12 i ICSP.....	30
4.2.4. Układ MMUSB232.....	31
4.2.5. Przyciski sterujące oraz przycisk RESETU.....	32
4.2.6. Gniazda kołkowe żeńskie i męskie.....	33
5. Oprogramowanie mikrokontrolera.....	34
5.1. Struktura oprogramowania.....	34
5.2. Inicjalizacja.....	35
5.2.1. Oscylator.....	35
5.2.2. Porty.....	35
5.2.3. Obsługa przerwań.....	36
5.2.4. USART.....	37
5.2.5. Komparatory.....	37
5.2.6. Przetwornik cyfrowo-analogowy.....	38
5.2.7. Licznik.....	38
5.3. Funkcja pomiarowa.....	39
5.4. Obliczenia.....	41
5.5. Komunikacja po stronie mikrokontrolera.....	43

5.6. Komunikacja po stronie PC.....	44
5.6.1. Pojedynczy pomiar.....	45
5.6.2. Pomiar wielokrotny pojemności.....	45
5.6.3. Pomiar napięcia na wyjściu DAC.....	45
6. Badanie eksperymentalne.....	46
6.1. Test wewnętrznego oscylatora kwarcowego.....	46
6.2. Test DAC.....	48
6.3. Test poprawności pomiaru.....	50
7. Podsumowanie.....	54
WYKAZ LITERATURY .....	55
WYKAZ RYSUNKÓW.....	56
WYKAZ TABEL.....	58
DODATEK A: SCHEMAT IDEOWY UKŁADU POMIAROWEGO.....	59
DODATEK B: SCHEMAT OBWODU DRUKOWANEGO.....	60

## WYKAZ SKRÓTÓW

- AC – analog comparator (komparator analogowy),
- ICSP – in circuit serial programming (szeregowe programowanie wewnątrz obwodu),
- DAC – digital-analog converter (przetwornik cyfrowo analogowy),
- elementy L i C – elementy indukcyjne i pojemnościowe,
- PC – personal computer (komputer osobisty),
- mcu – microcontroller unit (mikrokontroler),
- PLL – phase locked loop (pętla synchronizacji fazy),
- UART – universal asynchronous receiver-transmitter (uniwersalny asynchroniczny nadajnik-odbiornik),
- USART – universal synchronous/asynchronous receiver-transmitter (uniwersalny synchroniczny/asynchroniczny nadajnik-odbiornik),
- FFT – fast fourier transform (szybka transformata fouriera),
- FVR – fixed voltage reference (stałe napięcie odniesienia).

## 1. Wstęp

Czujniki pojemnościowe i indukcyjne mają zastosowanie w wielu dziedzinach, takich jak automatyka, przemysł motoryzacyjny, medycyna, militaria, a także w zastosowaniach konsumenckich. Zwykle są wbudowane w urządzenia i stanowią elementy pomiarowe systemu do sterowania jego pracą. Czujniki mierzą fizyczne zmienne monitorowanego środowiska, obiektów oraz procesów. Mogą to być temperatura, wilgotność otoczenia, czy napięcie. Zmianie wielkości nieelektrycznej odpowiada zmiana wartości pojemności lub indukcyjności. Obecnie czujniki są często projektowane jako inteligentne czujniki, które składają się poza czujnikiem z bloków: kondycjonowania, jednostki przetwarzającej i sterującej oraz interfejsów służących do komunikacji. Daje to możliwość przetwarzania takiej wielkości do postaci cyfrowej, przechowywania jej, a także przesyłania przez dowolny bezprzewodowy interfejs do komputera, na serwer lub aplikacji w smartfonie. Ze względu na to, iż często takie systemy są zasilane bateryjnie lub energią pozyskiwaną ze środowiska, to powinny być energooszczędne i mieć niewielkie rozmiary. Takie wymagania spełniają mikrokontrolery, które wyposażone w przetwornik DAC, licznik i AC mogą przy niewielkim poborze energii przetwarzać wielkość elektryczną do postaci cyfrowej i obsługiwać wymianę informacji poprzez interfejsy przewodowe lub bezprzewodowe [1].

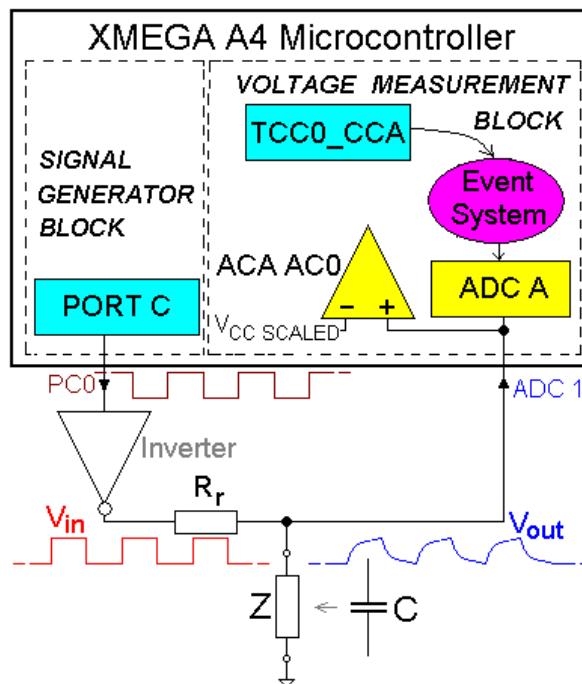
Celem niniejszej pracy jest zaprojektowanie i realizacja mikrosystemu pomiarowego do pomiarów wartości elementów L i C, który będzie komunikował się przy pomocy interfejsu UART z komputerem PC. Zadania, które należy wykonać w ramach pracy dyplomowej to zapoznanie się z metodami pomiaru opartymi na interfejsie czujnik – mikrokontroler, a także budowę i działaniem mikrokontrolera PIC16F1789, opracowanie koncepcji miernika, zaprojektowanie obwodu PCB, montaż układu, napisanie kodu programu na mikrokontroler realizującego pomiar i obsługę interfejsu UART oraz przeprowadzenie badań eksperymentalnych sprawdzających poprawność pracy układu.



## 2. Przegląd metod pomiaru indukcyjności i pojemności opartych o interfejs czujnik-mikrokontroler.

Rozdział dotyczy omówienia metod pomiaru pojemności i indukcyjności z wykorzystaniem bezpośredniego interfejsu czujnik-mikrokontroler. Przegląd ma na celu wybraniu najlepszej metody do zastosowania w projekcie dyplomowym.

### 2.1. Metoda oparta na ADC i AC.



Rys 2.1: Schemat podłączenia obwodu pomiarowego do peryferiów mikrokontrolera - metoda z ADC i AC [2].

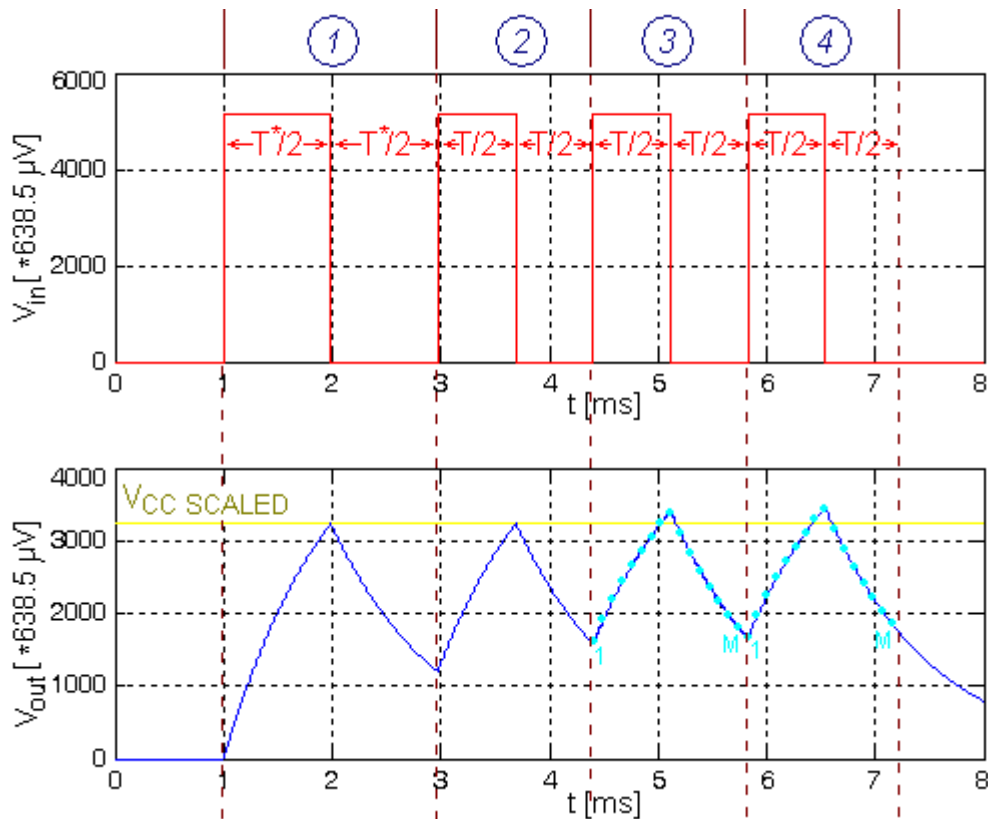
Na wyprowadzeniu mikrokontrolera dołączonego do rezystora  $R_r$  generowany jest przebieg prostokątny  $V_{in}$ . Rezystor  $R_r$  i czujnik pojemnościowy lub indukcyjny oznaczony jako  $Z$ , stanowią dzielnik napięcia (Rys 2.1). Napięcie  $V_{out}$  podane jest na drugie wyprowadzenie mikrokontrolera, które podłączone jest do AC oraz ADC. Układ AC zatrzymuje pomiar czasu po zrównaniu się przebiegu  $V_{out}$  z napięciem  $V_{ccscaled}$  oraz wyzwala przetwornik analogowo-cyfrowy. ADC próbuje napięcie  $V_{out}$ , a licznik określa chwilę  $t_m$  pobrania próbki  $V_m$ . Na podstawie algorytmu FFT wyznaczane są części rzeczywiste i urojone pierwszej harmonicznej sygnału wejściowego i wyjściowego. Na podstawie tych danych można obliczyć pojemność, stosując następujące wzory:

$$C = \sqrt{\frac{1 - K_1}{K_1 \cdot \omega^2 \cdot R_r^2}} \quad (2.1)$$

$$\omega = 4 \cdot \pi \cdot t_2 \cdot DIVx \cdot t_{CLK} \quad (2.2)$$

$$K_1 = \frac{V_{reout}^2 + v_{imout}^2}{V_{rein}^2 + v_{imin}^2} \quad (2.3)$$

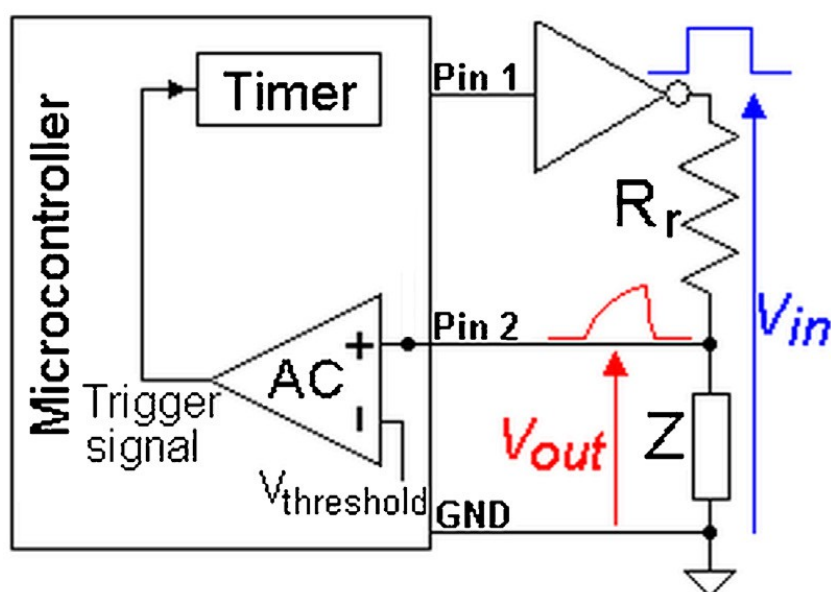
Gdzie  $V_{reout}$  i  $V_{imout}$  to część rzeczywista i urojona napięcia wyjściowego,  $V_{rein}$  i  $V_{imin}$  to część rzeczywista i urojona napięcia wejściowego,  $DIVx$  to wartość dzielnika częstotliwości licznika,  $t_2$  to czas pomiaru a  $t_{CLK}$  to okres zegara.



Rys 2.2: Przebieg napięcia wyjściowego i wejściowego na wyprowadzeniach mikrokontrolera - metoda z ADC i AC [2].

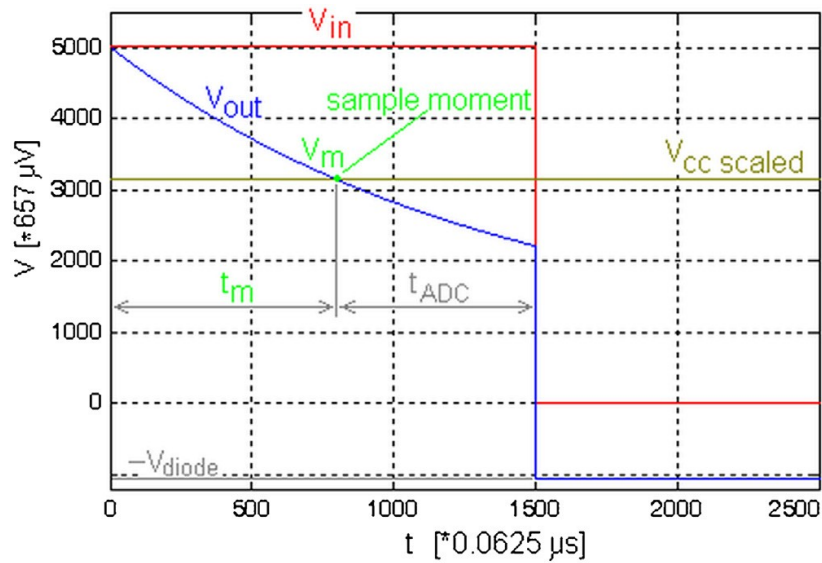
Na rys. 2.2 przedstawiono przebieg napięć pobudzających obwód pomiarowy  $V_{in}$  i odpowiedź czujnika  $V_{out}$  mierzoną przez ADC. Czujnik kilkakrotnie jest pobudzany i rozładowywany. Na przedstawionym wykresie cztery razy, przy czym próbki są pobierane z trzeciego i czwartego cyklu po ustabilizowaniu się odpowiedzi czujnika. Czasy pomiaru  $T$  powinny być, jak najkrótsze w celu oszczędzania energii, a pobieranie próbek w ciągu dwóch cykli poprawia dokładność pomiaru.

## 2.2. Metoda oparta na liczniku – pomiar czasu ładowania czujnika.



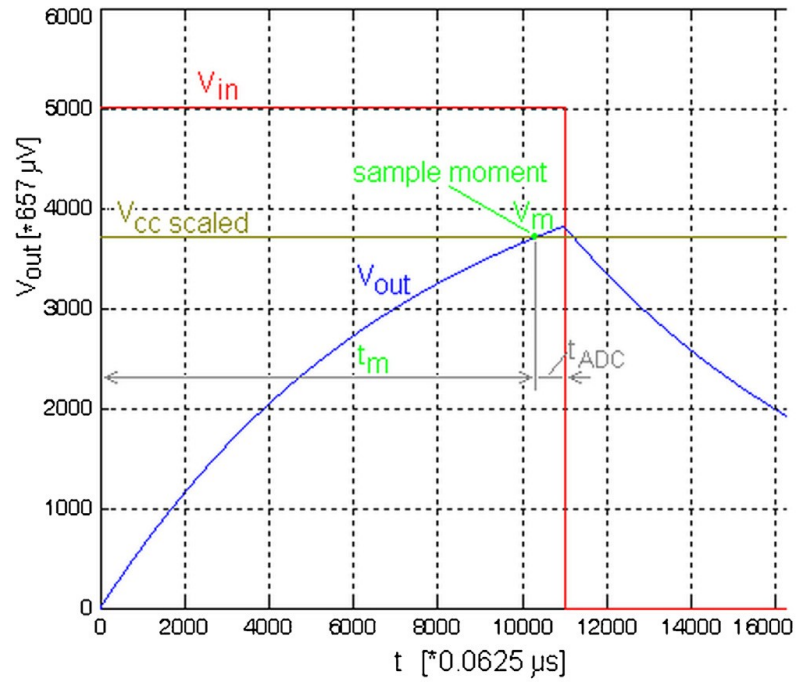
Rys 2.3: Schemat podłączenia obwodu pomiarowego do peryferiów mikrokontrolera - metoda z komparatorem i licznikiem [1].

Obwód pomiarowy jest taki sam, jak w przypadku wcześniejszej metody, jednak linia mikrokontrolera oznaczona jako Pin 2 podłączona jest wyłącznie do komparatora tak, jak na rys 2.3. W momencie wyzwolenia impulsu  $V_{in}$  uruchamiany jest licznik, który mierzy czas ładowania czujnika  $Z$ . Po osiągnięciu przez odpowiedź  $V_{out}$  poziomu  $V_{threshold}$  komparator analogowy wyzwała zatrzaśnięcie wartości licznika za pomocą sygnału zaznaczonego na schemacie, jako „trigger signal”. Na podstawie czasu ładowania obliczana jest pojemność lub indukcyjność przez mikrokontroler. Wykresy przedstawione na rys. 2.2 i rys. 2.3 są prawdziwe także dla tej metody. Jedyną różnicą jest fakt, iż nie mierzymy napięcia przetwornikiem ADC.



Rys 2.4: Przebieg napięcia obserwowany na czujniku indukcyjnym w trakcie pomiaru [1].

Na rys 2.2. widoczne są przebiegi napięcia zmieniającego się w czasie, na czerwono zaznaczono impuls napięcia generowanego przez mikrokontroler, zaś na niebiesko napięcie mierzone na czujniku indukcyjnym Z. W momencie uruchomienia procedury pomiarowej i wygenerowania impulsu pomiarowego dochodzi do ładowania czujnika. W przypadku pomiaru indukcyjności prowadzi to do wykładniczego spadku napięcia, wynikającego z ograniczenia przepływu prądu przez prąd wsteczny w rdzeniu cewki, który indukuje się w trakcie ładowania się pola magnetycznego aż do poziomu, na którym stabilizuje się prąd i napięcie. Na podstawie czasu pobrania próbki  $t_m$ , gdy napięcie na czujniku osiągnie wartość oznaczoną, jako  $V_m$  wyznaczona jest wartość indukcyjności czujnika.



Rys 2.5: Przebieg napięcia na czujniku pojemnościowym w trakcie pomiaru [1].

Dla pomiaru pojemności przebieg napięcia przedstawiony jest na rys. 2.3. W momencie wygenerowania przez mikrokontroler napięcia na wyprowadzeniu, do którego podłączony jest rezystor  $R_r$ , ładuje się pole elektryczne czujnika, poprzez gromadzenie się ładunków na okładkach kondensatora. Prowadzi to do wzrostu potencjału elektrycznego w węzle, do którego podłączony jest komparator analogowy. Na podstawie czasu ładowania  $t_m$ , dla którego napięcie  $V_{out}$  osiąga wartość  $V_m$  obliczana jest pojemność czujnika.

Obliczenia pojemności i indukcyjności dokonywane są na podstawie poniższych wzorów, pozwalają one dobrać również zakres dla rezystora  $R_r$ :

Dla czujnika indukcyjnego zakładając rezystancję wewnętrzną czujnika  $R_s$  i minimalny czas pomiaru  $t_{min}$ , który zależy od rozdzielczości licznika dostępnego w mikrokontrolerze, można dobrać wartość  $R_r$  przez wstawienie jej do wzoru [1]:

$$t_{min} = \frac{-\ln\left(\left(\frac{V_{ccscale}}{V_{in}} \cdot (R_s + R_r) - R_s\right) / R_r\right) \cdot L_{min}}{R_s + R_r} \quad (2.4)$$

Gdzie  $L_{min}$  to dolna wartość zakresu pomiarowego, który będzie mierzony. Jako, że do pomiarów używany jest licznik za  $t_{min}$  można podstawić [1]:

$$t_{min} = N_{min} \cdot T_{clk} \quad (2.5)$$

Gdzie  $T_{clk}$  to okres zegara taktującego a  $N_{min}$  to minimalna liczba impulsów zliczanych przez timer, przy dzielniku 1:1  $N_{min}=1$ .  $V_{ccscaled}$  należy wyznaczyć stosując formułę [1]:

$$V_{ccscaled} = V_{in} \cdot \frac{2R_s + R_r}{2 \cdot (R_s + R_r)} \quad (2.6)$$

Aby po pomiarze obliczyć wartość indukcyjności, należy skorzystać ze wzoru [1]:

$$L = \frac{-t_m \cdot (R_s + R_r)}{\left( \frac{\frac{V_m}{V_{in}} (R_s + R_r) - R_s}{R_r} \right)} \quad (2.7)$$

Dla czujnika pojemnościowego  $R_r$  wyznacza się, jako przedział [1]:

$$R_{rmin} < R_r < R_{rmax} \quad (2.8)$$

$$R_{rmin} = \frac{-N_{min} \cdot T_{clk}}{C_{min} \cdot \ln \left( 1 - \frac{V_{ccscaled}}{V_{in}} \right)} \quad (2.9)$$

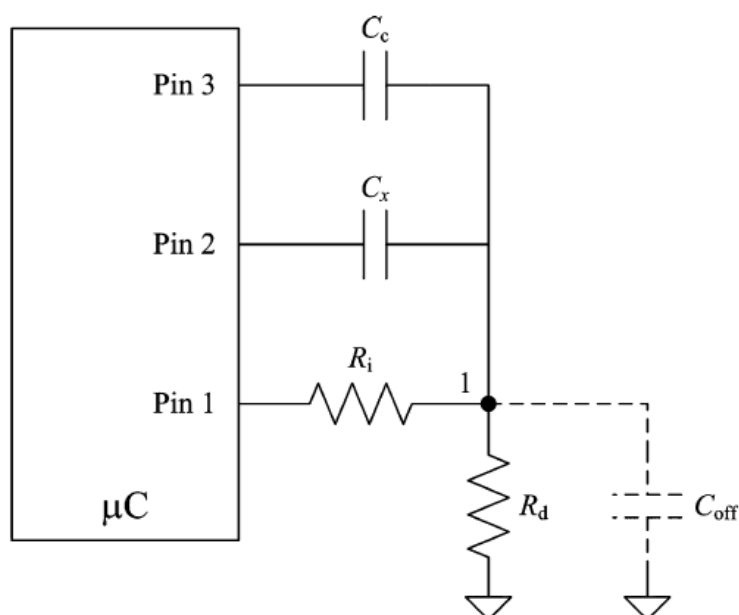
$$R_{rmax} = \frac{-N_{max} \cdot P_{max} \cdot T_{clk}}{C_{max} \cdot \ln \left( 1 - \frac{V_{ccscaled}}{V_{in}} \right)} \quad (2.10)$$

Gdzie  $N$  to  $2^n$ ,  $n$  to liczba bitów licznika.  $P_{max}$  to wartość dzielnika częstotliwości, maksymalna wartość to 1,  $T_{clk}$  to okres zegara a  $C_{min}$  oraz  $C_{max}$  to zakres mierzonych pojemności.  $V_{ccscaled}$  powinno mieć wartość około 0.66 z  $V_{in}$ .

Aby obliczyć wartość pojemności po pomiarze czasu  $t_m$ , należy użyć wzoru:

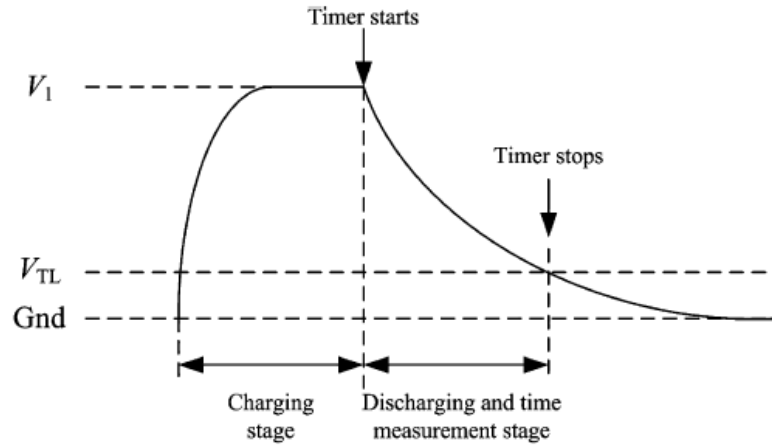
$$C = \frac{-t_m}{R_r \cdot \ln \left( 1 - \frac{V_{ccscaled}}{V_{in}} \right)} \quad (2.11)$$

### 2.3. Metoda oparta na liczniku – pomiar czasu rozładowywania czujnika.



Rys 2.6: Obwód pomiarowy - metoda z pomiarem czasu rozładowania [3].

$C_c$  to kondensator kalibracyjny,  $C_x$  to mierzona pojemność,  $C_{off}$  reprezentuje pojemność rozproszoną między węzłem 1 a masą (Rys. 2.6). Na wyprowadzenie mikrokontrolera podawane jest maksymalne napięcie odpowiadające napięciu zasilania  $V_1$ , które powoduje ładowanie kondensatorów. Gdy  $C_x$  zostanie naładowane wyprowadzenie oznaczone jako Pin 1, przestaje generować napięcie a Pin 2 i Pin 3 zostają ustawione w stan wysokiej impedancji. Uruchamiany jest również licznik, który mierzy czas rozładowania pojemności. Przy osiągnięciu określonego napięcia odpowiadającego stanowi niskiemu układów CMOS  $V_{TL}$  sprawdzanego na wyprowadzeniu Pin 1, licznik zostaje zatrzymany i obliczana jest pojemność na podstawie czasu rozładowywania, sumy pojemności  $C_x$  i  $C_{off}$ ,  $C_c$  i  $C_{off}$  oraz  $C_{off}$ , przez mikrokontroler.



Rys 2.7: Przebieg napięcia w węźle 1 w trakcie ładowania i rozładowywania pojemności [3].

Na rys. 2.7 przedstawione jest napięcie na rezystorze  $R_d$  w trakcie pomiaru. Licznik rozpoczyna pomiar w chwili, gdy czujnik w pełni się naładuje. Następnie rozpoczyna się rozładowywanie czujnika, a pomiar czasu zostaje zakończony po spadku napięcia do  $V_{TL}$ . Na podstawie poniższych wzorów dokonywane są obliczenia wartości pojemności [3]. Wszystkie pojemności ( $C_c$ ,  $C_x$ ,  $C_{off}$ ) razem stanowią pojemność  $C_{eq}$ , na podstawie której określony jest czas rozładowania:

$$T = R_d C_{eq} \ln \left( \frac{V_1}{V_{TL}} \right) \quad (2.12)$$

Na podstawie czasów rozładowania  $C_x + C_{off}$  oznaczonego jako  $T_x$ ,  $C_c + C_{off}$  oznaczonego jako  $T_c$  oraz czasu rozładowania  $T_{off}$ , które można przybliżyć za pomocą formuł:

$$T_x = k(C_x + C_{off}) \quad (2.13)$$

$$T_c = k(C_c + C_{off}) \quad (2.14)$$

$$T_{off} = k C_{off} \quad (2.15)$$

Można obliczyć współczynnik  $M$  i na jego podstawie oraz znanej pojemności  $C_c$  obliczyć szukaną wartość  $C_x$ :

$$M = \frac{T_x - T_{off}}{T_c - T_{off}} \quad (2.16)$$

$$C_x = M \cdot C_c \quad (2.17)$$

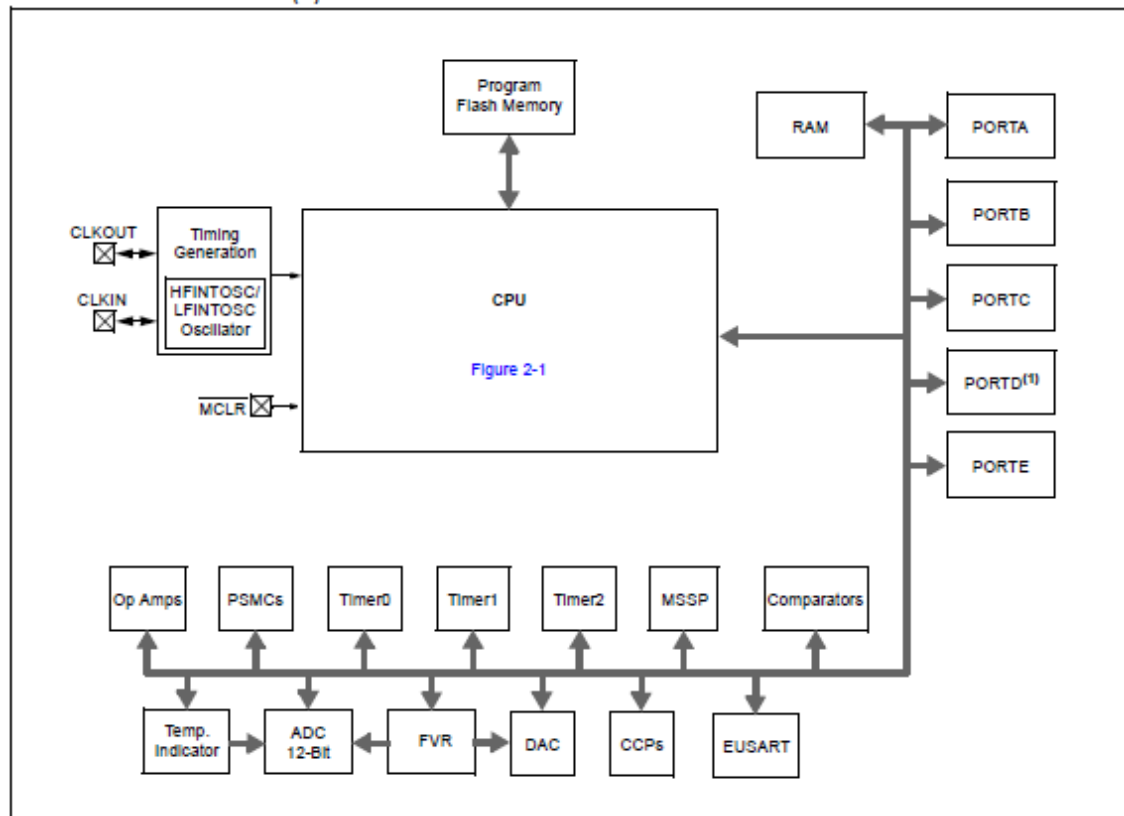


## **2.4. Podsumowanie**

Spośród metod omówionych w tej pracy wybrana została metoda oparta na pomiarze czasu ładowania obwodu czujnika z wykorzystaniem komparatora analogowego ze względu na prostą procedurę pomiarową i niewielką liczbę peryferiów użytych przez metodę pomiarową. Wybrana metoda zapewnia stosunkowo dużą dokładność pomiaru, krótki czas pomiaru oraz prosty algorytm pomiarowy przekładający się na prostotę oprogramowania mikrokontrolera.

### 3. Opis mikrokontrolera PIC16F1789 i jego najważniejszych peryferiów.

W tym rozdziale zostaną opisane podstawowe funkcje i budowa mikrokontrolera PIC16F1789, a także dokładnie zostaną omówione działanie i sposób konfiguracji układów peryferyjnych wykorzystywanych przez metodę pomiarową.



Rys 3.1: Schemat blokowy mikrokontrolera PIC16F1789 [5].

Użyty mikrokontroler to 8-bitowy mikrokontroler firmy Microchip z rodziny PIC16. MCU posiada 40 wyprowadzeń, czyli portów równoległych oznaczonych jako porty A, B, C, D, E. Ze schematu blokowego i dokumentacji mikrokontrolera można odczytać, że do dyspozycji użytkownika mamy:

- Procesor oparty na architekturze RISC (49 instrukcji), taktowany zegarem o maksymalnej częstotliwości 32 MHz.
- Pamięć programu Flash o pojemności 16 kB.
- Pamięć danych EEPROM o pojemności 256 bajtów.
- Pamięć RAM o pojemności 2048 bajtów.

- ### 3.1. Licznik.



17

na dużą rozdzielczość (16 bitów) oraz możliwość zatrzymania zliczania używając sygnału z komparatora analogowego przy narastającym zboczu na jego wyjściu.

Licznik konfiguruje się przy pomocy dwóch rejestrów: T1CON i T1GCON [5].

R/W-0/u	R/W-0/u	R/W-0/u	R/W-0/u	R/W-0/u	R/W-0/u	U-0	R/W-0/u
TMR1CS<1:0>		T1CKPS<1:0>		T1OSCEN	T1SYNC	—	TMR1ON
bit 7							bit 0

Rys 3.3: Bity konfiguracyjne w rejestrze T1CON [5].

Rejestr T1CON służy do ustawienia podstawowych funkcji licznika [5]:

- Bity 7 i 6 – odpowiadają za ustawienie źródła zegarowego, mogą to być:
  - 0b10 – zegar lub oscylator podłączony do odpowiednich wyprowadzeń mikrokontrolera,
  - 0b01 – Fosc, czyli wartość zegara systemowego,
  - 0b00 – Fosc/4, czyli wartość zegara instrukcji, którego częstotliwość to podzielona przez 4 częstotliwość zegara systemowego.
- Bity 5 i 4 – wartości tych bitów odpowiadają za wartość dzielnika częstotliwości:
  - 0b11 – dzielenie przez 8,
  - 0b10 – dzielenie przez 4,
  - 0b01 – dzielenie przez dwa,
  - 0b00 – brak dzielenia częstotliwości.

W tej pracy możliwość wyłączenia dzielenia częstotliwości licznika jest bardzo dużą zaletą. Brak podziału sygnału zegarowego zmniejsza błąd pomiaru ze względu na możliwość precyzyjniejszego określenia dokładnej chwili zatrzymania licznika. W przypadku, np. dzielenia częstotliwości przez 4, każde narastające zbocze liczy się jako 4 impulsy zegara. Jeśli zatrzymanie licznika nastąpi w dowolnym momencie po wystąpieniu narastającego zbocza zegara i tak liczy się on jako 4, co obniża rozdzielczość pomiaru.

- Bit 3 – uruchomienie dedykowanego obwodu oscylatora dla licznika timer1. W tej pracy jest on wyłączony, ponieważ zapewnia częstotliwość tylko do 32 kHz.
- Bit 2 – możliwość ustawienia synchronizowania licznika z zegarem systemowym.

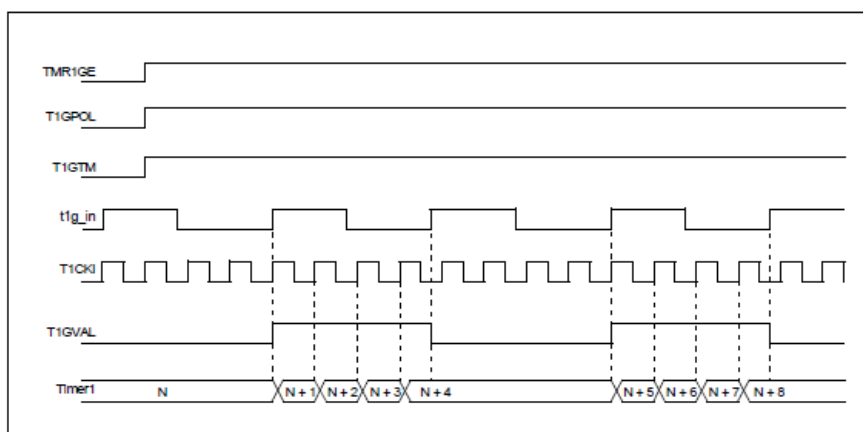
- Bit 0 – w momencie ustawienia tego bitu jako ‘1’ licznik rozpoczyna odliczanie, chyba, że jest kontrolowany przez bramkę, wówczas pozostaje aktywny i oczekuje na sygnał od bramki.

R/W-0/u	R/W-0/u	R/W-0/u	R/W-0/u	R/W/HC-0/u	R-x/x	R/W-0/u	R/W-0/u
TMR1GE	T1GPOL	T1GTM	T1GSPM	T1GGO/ DONE	T1GVAL	T1GSS<1:0>	
bit 7							bit 0

Rys 3.4: Bity konfiguracyjne w rejestrze T1GCON [5].

Rejestr T1GCON służy do ustawienia funkcji związanych z bramką licznika:

- Bit 7 – ustawienie tego bitu jako ‘1’ włącza kontrolę odliczania licznika przez bramkę.
- Bit 6 – ustawienie, czy licznik ma zliczać impulsy, gdy stan sygnału wejściowego bramki jest niski, czy wysoki.
- Bit 5 – uruchomienie „toggle mode”. Gdy tryb ten jest uruchomiony licznik rozpoczyna i zatrzymuje odliczanie na każde narastające zbocze pojawiające się na wejściu sygnałowym bramki. Jest to przedstawione na poniższym rysunku (Rys. 3.5):



Rys 3.5: Przebiegi czasowe licznika przy uruchomionym "toggle mode" [5].

- Bit 4 – tryb pojedynczego wyzwolenia. Gdy jest uruchomiony a bit 3 ustawiony jest, jako ‘1’ następne narastające zbocze na wejściu bramki spowoduje zwiększenie stanu licznika o 1.
- Bit 3 – musi być ustawiony, jako ‘1’ aby zmiana stanu na wejściu bramki zwiększyła stan licznika. Po kolejnej takiej samej zmianie stanu, na wejściu sygnałowym bramki, zostanie automatycznie wyzerowany.
- Bit 2 – wskazuje aktualny stan bramki.
- Bit 1 i 0 – wybór źródła dla wejścia bramki:

-0b11 – wyjście komparatora 2 (sync\_C2OUT),

-0b10 – wyjście komparatora 1 (sync\_C1OUT),

-0b01 – wyjście sygnalizujące, że licznik 0 (TIMER0) zappełnił się, na rys. 3.2 oznaczone, jako „From Timer0 overflow”. Umożliwia to zwiększenie rozdzielczości licznika. Pomiar uruchamiany jest dla licznika 0, a gdy dojdzie on do maksymalnej wartości uruchamiany jest licznik 1,

-0b00 – podłączenie do zewnętrznego wyprowadzenia mikrokontrolera (wyprowadzenie oznaczone jako RB5).

Wartość licznika przechowywana jest w rejestrach: TMR1H – 8 bardziej znaczących bitów, TMR1L – 8 mniej znaczących bitów.

## 3.2. Oscylator

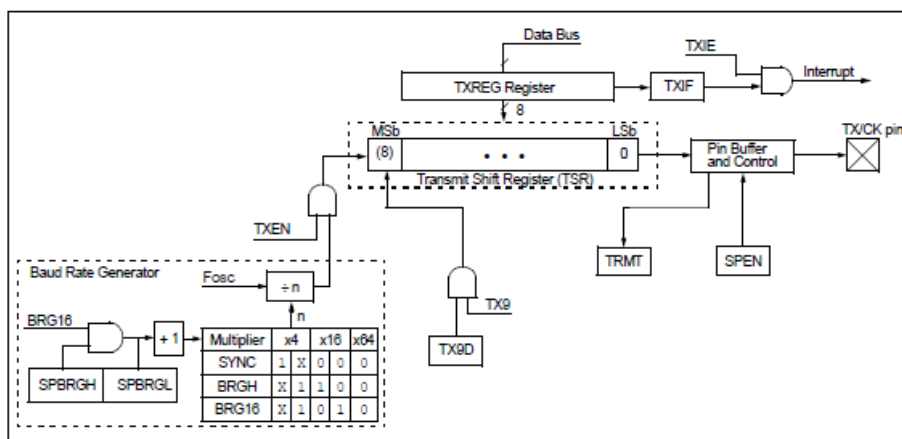
Wewnętrzny oscylator umożliwia programowe ustawienie częstotliwości od 31 kHz do 16 MHz. Przy użyciu układów PLL możliwe jest ustawienie częstotliwości 32 MHz. Przez wyprowadzenia OSC1 i OSC2 (RA6 i RA7) możliwe jest podłączenie zewnętrznego oscylatora kwarcowego lub RC działającego do częstotliwości 32 MHz. Aby skonfigurować oscylator, należy ustawić bity konfiguracyjne w rejestrze OSCCON.

R/W-0/0	R/W-0/0	R/W-1/1	R/W-1/1	R/W-1/1	U-0	R/W-0/0	R/W-0/0
SPLLEN	IRCF<3:0>				—	SCS<1:0>	
bit 7							bit 0

Rys 3.6: Bity konfiguracyjne w rejestrze OSCCON [5].

- Bit 7 – jeśli w słowach konfiguracyjnych układy PLL są cały czas włączone, to bit nie jest używany, w innym wypadku pozwala na włączenie układu PLL, powoduje to zwiększenie częstotliwości sygnału oscylatora.
- Bity 6-3 – ustawienie częstotliwości wewnętrznego oscylatora od 32 MHz (bez układów PLL od 16 MHz) do 31 kHz.
- Bity 1-0 – wybór zegara systemowego:
  - 0b1x – wewnętrzny oscylator,
  - 0b01 – oscylator licznika 1,
  - 0b00 – zegar wybrany w rejestrach konfiguracyjnych.

### 3.3. Blok EUSART



Rys 3.7: Blok EUSART [5].

Blok EUSART pozwala na wysyłanie i odbieranie danych i komunikację z zewnętrznymi urządzeniami np. przez interfejs RS-232. Konfiguruje się go za pomocą czterech rejestrów: TXSTA, RXSTA, BAUDCON, SPBRG.

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R-1/1	R/W-0
CSRC	TX9	TXEN <sup>(1)</sup>	SYNC	SENDB	BRGH	TRMT	TX9D
bit 7							bit 0

Rys 3.8: Bity konfiguracyjne w rejestrze TXSTA [5].

Rejestr TXSTA służy do ustawień odpowiedzialnych za przesył danych:

- Bit 7 – dla trybu synchronicznego pozwala ustawić, czy urządzenie ma być master, czy slave.
- Bit 6 – ustawienie dla ilu bitów ma przebiegać transmisja ('1'-9 bitów, '0' - 8 bitów).
- Bit 5 – włączenie układu do transmisji.
- Bit 4 – ustawienie trybu pracy: '1' - tryb synchroniczny, '0' - tryb asynchroniczny.
- Bit 3 – po ustawieniu '1' nadajnik wyśle tzw. „break charakter”, czyli znak przerywania składający się z bitu startu, 12 bitów '0' i bitu stopu, po przesłaniu znaku przerywania bit 3 zostanie wyzerowany.
- Bit 2 – ustawienie wysokiej prędkości transmisji.
- Bit 1 – status wyjściowego rejestru przesuwającego.
- Bit 0 – przechowuje 9 bit do transmisji, ponieważ rejestr wyjściowy ma tylko 8 bitów pojemności.

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R-0/0	R-0/0	R-0/0
SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D
bit 7							bit 0

Rys 3.9: Bity konfiguracyjne w rejestrze RCSTA [5].

Rejestr RCSTA dotyczy ustawień odpowiedzialnych za odbiór danych:

- Bit 7 – po wpisaniu '1' wyprowadzenia RB6 i RB7 są ustawione, jako linie interfejsu szeregowego.
- Bit 6 – uruchomienie odbioru 9 bitów.
- Bit 5 – dla trybu synchronicznego master ustawienie pojedynczego odbioru, po odbiorze bit jest zerowany.
- Bit 4 – dla trybu asynchronicznego uruchamia odbiornik, dla trybu synchronicznego ustawienie '1' uruchamia ciągły odbiór do czasu wyzerowania bitu.
- Bit 3 – ustawienie dla odbioru asynchronicznego przy 9 bitach, '1' - włącza detekcję adresu, gdy adres zostanie wykryty, uruchamiane jest przerwanie, '0' - wszystkie bity są odbierane, 9 bit może być użyty jako bit parzystości.
- Bit 2 i 1 – bity sygnalizacyjne błędów, bit 2 jest ustawiany jako '1' na przykład w momencie, kiedy nie zostanie ustawiony bit stopu (błąd ramki), bit 1 jest ustawiany jako '1' kiedy następuje przepełnienie rejestru.
- Bit 0 – przechowuje dziewiąty odebrany bit ponieważ rejestr wejściowy ma tylko 8 bitów pojemności.

Rejestr BAUDCON dotyczy bardziej zaawansowanych ustawień nieistotnych z punktu widzenia tej pracy, m.in. ustawienia dotyczące automatycznego wykrywania szybkości transmisji czy wybudzania układu ze stanu czuwania w przypadku odbioru.

Wartość rejestru SPBRG określa prędkość transmisji, wartość ta jest obliczana z poniższych wzorów:

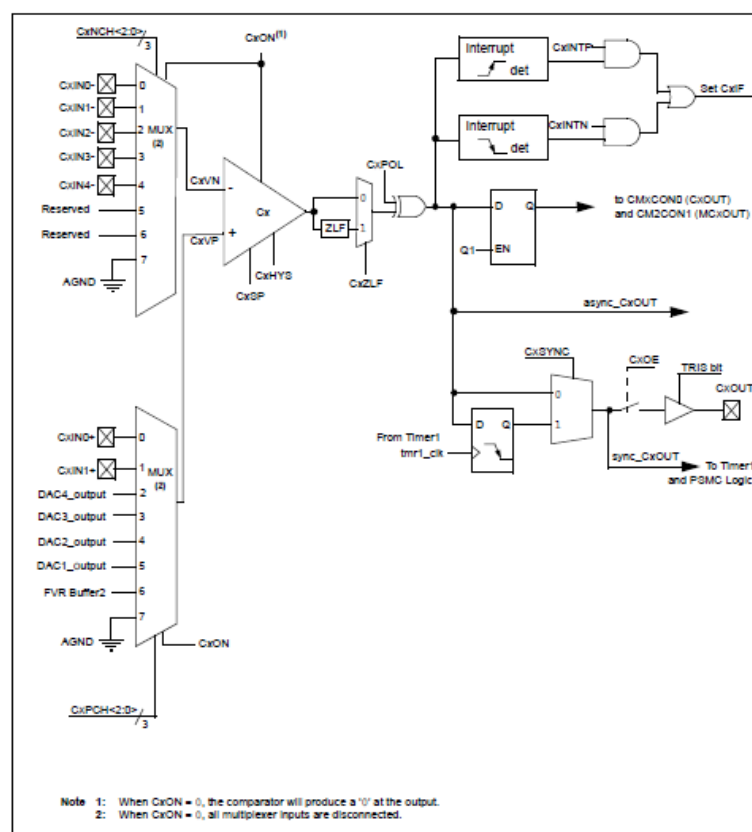
$$SPBRG = \left( \frac{(F/64)}{BAUDRATE} \right) - 1, \text{ dla } BRGH = '0' \quad (3.1)$$

$$SPBRG = \left( \frac{(F/16)}{BAUDRATE} \right) - 1, \text{ dla } BRGH = '1' \quad (3.2)$$

Gdzie F – częstotliwość zegara w Hz, BAUDRATE – szybkość transmisji w b/s.



### 3.4. Komparator analogowy



Rys 3.10: Schemat blokowy komparatora analogowego [5].

Komparator analogowy to układ elektroniczny pozwalający na porównanie dwóch sygnałów analogowych. Dostępne cztery komparatory są konfigurowane za pomocą dwóch rejestrów: CMxCON0 i CMxCON1.

R/W-0/0	R-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-1/1	R/W-0/0	R/W-0/0
CxON	CxOUT	CxOE	CxPOL	CxZLF	CxSP	CxHYS	CxSYNC
bit 7				bit 0			

Rys 3.11: Bity konfiguracyjne w rejestrze CMxCON0 [5].

- Bit 7 – gdy jest wpisana ‘1’, układ jest uruchomiony. Przy inicjalizacji i w momencie, kiedy nie jest to konieczne do wykonania programu przez MCU, komparatory powinny być wyłączone, by nie zużywały energii.
- Bit 6 – wartość na wyjściu komparatora.
- Bit 5 – gdy ustawiona jest ‘1’ wyjście komparatora podłączone jest do zewnętrznego wyprowadzenia mikrokontrolera.
- Bit 4 – ustawia polaryzację wyjścia, gdy bit jest ustawiony, jako ‘0’ to na wyjściu komparatora jest ‘1’, gdy sygnał podłączony do plusa jest większy od sygnału podłączonego do minusa.

- Bit 3 – filtr zerowej latencji. Przy szybko zmieniającym się stanie komparatora na wyjściu, mogą pojawiać się oscylacje, włączenie filtru umożliwia filtrowanie tych oscylacji.
- Bit 2 – wybór szybkości działania, można wybrać niższą szybkość, by zmniejszyć pobór energii przy rzadko zmieniającym się stanie wejść komparatora.
- Bit 1 – umożliwia włączenie funkcji histerezy.
- Bit 0 – synchronizacja z zegarem podłączonym do licznika 1.

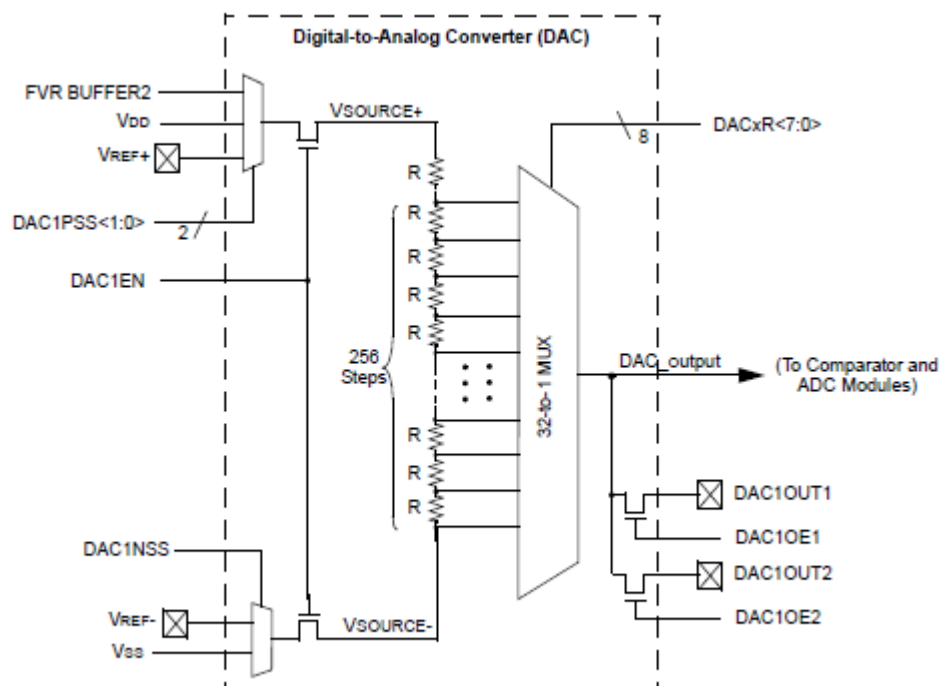
R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
CxINTP	CxINTN	CxPCH<2:0>			CxNCH<2:0>		
bit 7		bit 0					

Rys 3.12: Bity konfiguracyjne w rejestrze CxCON1 [5].

- Bit 7 – włączenie przerwania na narastające zbocze na wyjściu komparatora.
- Bit 6 – włączenie przerwania na opadające zbocze na wyjściu komparatora.
- Bity 5-3 – konfiguracja podłączenia dodatniego wejścia komparatora:
  - 0b111 – do masy,
  - 0b110 – do bufora wyjściowego FVR,
  - 0b101, 0b100, 0b011, 0b010 – do jednego z przetworników DAC,
  - 0b001, 0b000 – do jednego z zewnętrznych wyprowadzeń mikrokontrolera.
- Bity 2-0 – konfiguracja podłączenia ujemnego wejścia komparatora:
  - 0b111 – podłączenie do masy,
  - 0b110 – wyjście niepodłączone (input floating),
  - 0b101 – zarezerwowane,
  - 0b100, 0b011, 0b010, 0b001, 0b000 – podłączenie do jednego z zewnętrznych wyprowadzeń mikrokontrolera.

Dodatkowo w rejestrze CMOUT, bity 3-0 są kopiami wyjść komparatorów 4-1.

### 3.5. DAC



Rys 3.13: Schemat blokowy 8-bitowego przetwornika cyfrowo-analogowego [5].

Przetwornik cyfrowo-analogowy na podstawie wpisanej do przetwornika cyfrowej wartości wystawia na wyjściu analogowe napięcie proporcjonalne do tej wartości. 8-bitowy przetwornik DAC konfiguruje się za pomocą dwóch rejestrów: DAC1CON0 i DAC1CON1.

R/W-0/0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	U-0	R/W-0/0
DAC1EN	—	DAC1OE1	DAC1OE2	DAC1PSS<1:0>	—	—	DAC1NSS
bit 7							bit 0

Rys 3.14: Bity konfiguracyjne w rejestrze DAC1CON0 [5].

- Bit 7 – wpisanie '1' powoduje włączenie przetwornika i generowanie napięcia na jego wyjściu.
- Bit 5 i 4 – podłączenie wyjścia przetwornika do zewnętrznych wyprowadzeń mikrokontrolera DAC1OUT1 i DAC1OUT2.
- Bit 3 i 2 – wybór dodatniego potencjału źródła ( $V_{SOURCE+}$ ):
  - 0b10 – bufor wyjściowy FVR,
  - 0b01 – zewnętrzne wyjście mikrokontrolera  $V_{REF+}$ ,
  - 0b00 – dodatni potencjał napięcia zasilania.

- Bit 0 – wybór ujemnego potencjału źródła ( $V_{SOURCE-}$ ): '0' - masa, '1' - zewnętrzne wyjście mikrokontrolera  $V_{REF-}$ .

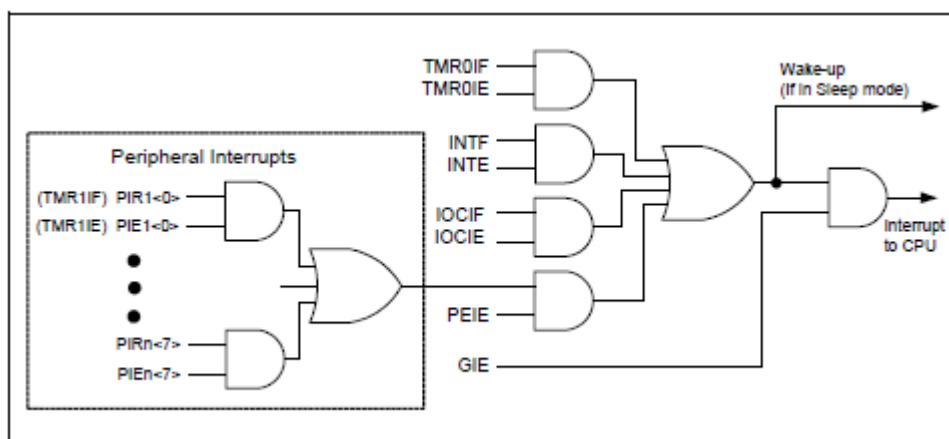
Do rejestru DAC1CON1 wpisywana jest wartość cyfrowa, która zostanie przetworzona na napięcie analogowe. Wzór pozwalający wyliczyć napięcie wyjściowe na podstawie wpisanej wartości to [5]:

$$V_{out} = \left( (V_{source+} - V_{source-}) \cdot \frac{DAC1CON1[7:0]}{2^8} \right) + V_{source-} \quad (3.3)$$

Wzór można przekształcić do postaci, w której można wyliczyć wartość rejestru na podstawie wartości napięcia, która ma zostać wygenerowana na wyjściu:

$$DAC1CON1[7:0] = \left( \frac{V_{out} - V_{source-}}{V_{source+} - V_{source-}} \right) \cdot 2^8 \quad (3.4)$$

### 3.6. System przerwań.



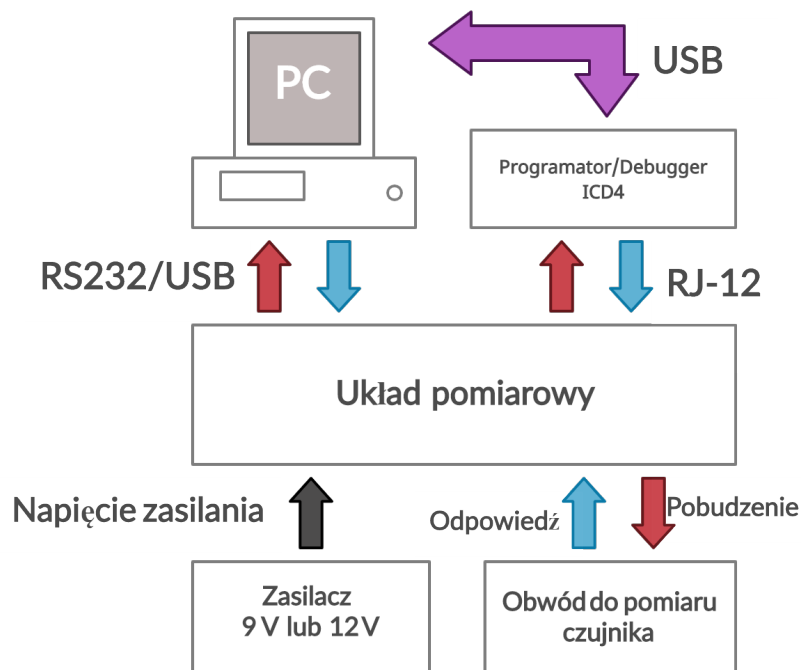
Rys 3.15: Schemat blokowy logiki systemu przerwań [5].

System przerwań pozwala na zatrzymanie normalnego przebiegu programu i wykonanie funkcji odpowiedzialnej za obsługę przerwania. Przerwania są generowane przez pobudzenie odpowiednio skonfigurowanego pinu nazwanego IOC lub INT albo od układów peryferyjnych. W momencie wystąpienia określonego zdarzenia, w rejestrze ustawiana jest flaga przerwania i uruchamiana jest napisana w języku C funkcja `_interrupt()`. W funkcji sprawdzane są warunki, która flaga została ustawiona. Po wykonaniu instrukcji należy wyzerować flagę przerwania, tak by układ nie zapętlił się w nim i wznowił wykonywanie programu od punktu, w którym został przerwany.

## 4. Stanowisko laboratoryjne

W tym rozdziale zostanie przedstawiony i omówiony schemat stanowiska laboratoryjnego i budowa układu pomiarowego.

### 4.1. Schemat blokowy



Rys 4.1: Schemat blokowy stanowiska laboratoryjnego.

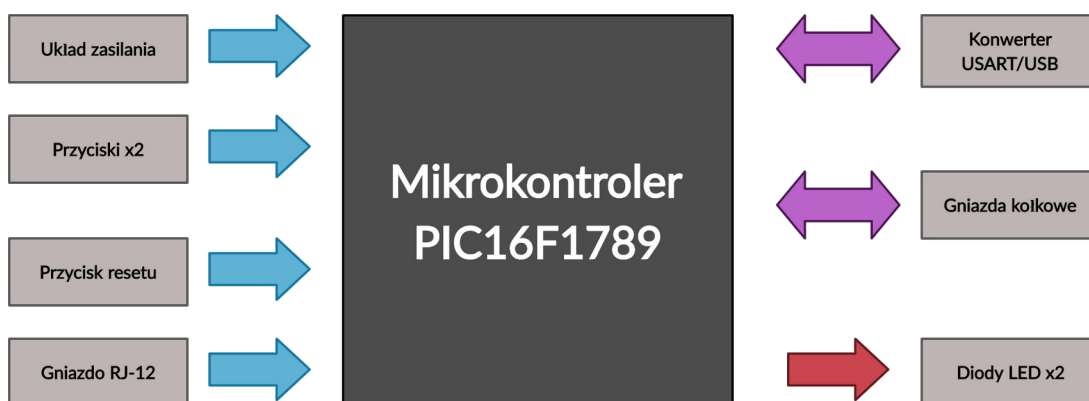
Stanowisko laboratoryjne składa się z czterech głównych bloków:

- Układu pomiarowego którego główną częścią jest mikrokontroler.
- Zasilacza 9 V lub 12 V.
- Komputera PC.
- Układu do pomiaru pojemności czujnika podłączonego do mikrokontrolera.

Dodatkowo w skład stanowiska może wchodzić programator/debbuger ICD4 podłączony do układu pomiarowego kablem telefonicznym sześciocyłowym przez gniazdo RJ - 12 i do komputera PC przez interfejs USB. Programator umożliwia programowanie mikrokontrolera, a także pracę w trybie debugowania, która pozwala śledzić stan pracy mikrokontrolera.

Zasilacz dostarcza do gniazda napięcie 9 V lub 12 V, które następnie za pomocą stabilizatora napięcia jest przetwarzane na 3,3 V, ponieważ takie jest potrzebne do zasilenia mikrokontrolera. Układ pomiarowy oparty jest na mikrokontrolerze PIC16F1789, który pełni rolę układu sterującego oraz dokonuje obliczeń. Do gniazd połączonych bezpośrednio z wyprowadzeniami mikrokontrolera wpinany jest rezystor wzorcowy oraz element indukcyjny lub pojemnościowy, które razem stanowią obwód do pomiaru czujnika. Komunikacja z komputerem odbywa się za pośrednictwem interfejsu UART, który za pośrednictwem układu MMUSB232 jest konwertowany na standard USB. Za pomocą komputera PC można rozpocząć pomiar i odczytać wyniki pomiaru za pomocą dowolnego terminala zainstalowanego w systemie komputera, który umożliwia obsługę transmisji dla standardu RS-232.

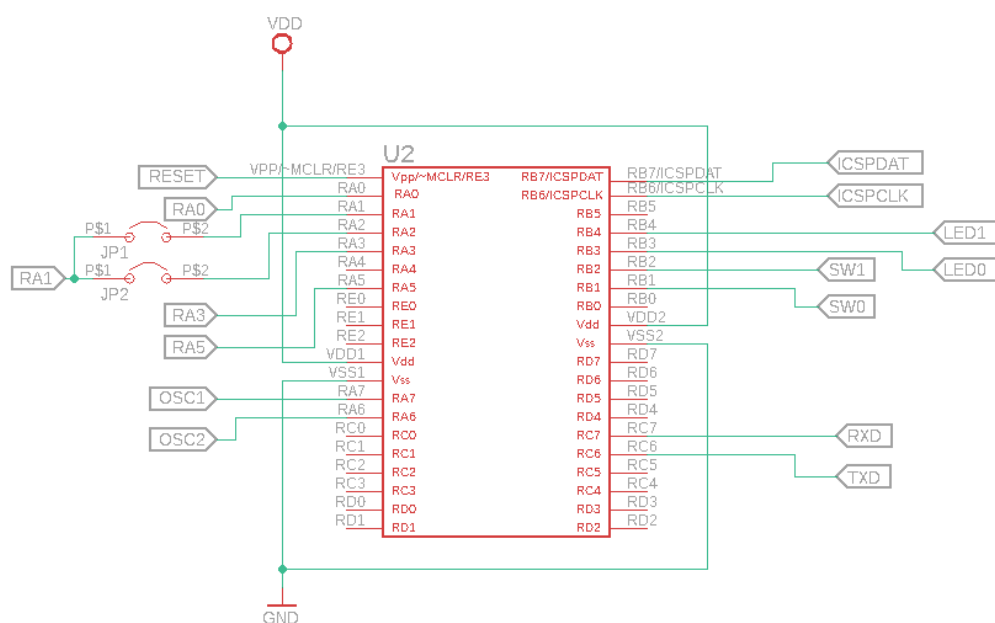
## 4.2. Schemat układu pomiarowego



Rys 4.2: Schemat blokowy układu pomiarowego

Głównym blokiem układu pomiarowego jest mikrokontroler PIC16F1789. Do niego zostały dołączone wszystkie elementy wymienione na schemacie blokowym (rys. 4.2). W tym rozdziale zostaną one opisane. Pełny schemat ideowy układu został umieszczony w dodatku A.

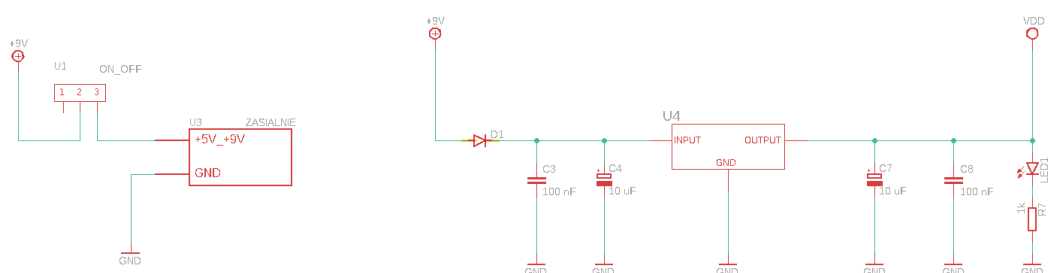
### 4.2.1. Jednostka sterująca



Rys 4.3: Schemat ideowy podłączenia mikrokontrolera w układzie pomiarowym.

Mikrokontroler pełni rolę jednostki sterującej i dokonującej obliczeń. Do jednostki podłączone są wszystkie elementy i układy wymienione w kolejnych podrozdziałach. Na schemacie ideowym do układu dla zwiększenia czytelności są dołączone etykiety oznaczające poszczególne węzły.

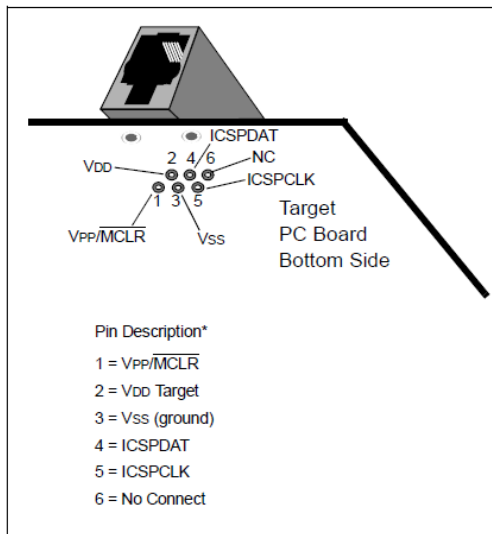
### 4.2.2. Układ zasilania



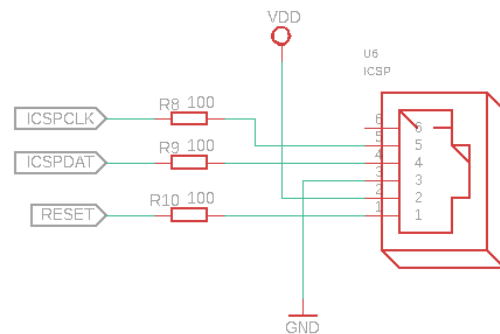
Rys 4.4: Schemat ideowy układu zasilania i włącznika.

Zasilacz podłączany jest do gniazda U3. Przez przełącznik dwupozycyjny U1 napięcie z zasilacza dołączone jest następnie do diody D1, która chroni układ przed niewłaściwą polaryzacją napięcia. Układ U4 to stabilizator napięcia LF33CV, który przetwarza napięcie z zasilacza na 3,3 V. Kondensatory ceramiczne C3 i C8 mają za zadanie filtrowanie zakłóceń o wyższych częstotliwościach zaś kondensatory elektrolityczne C4 i C7 zakłóceń o niższych częstotliwościach. LED1 to dioda elektroluminescencyjna sygnalizująca, że układ zasilania jest włączony.

### 4.2.3. Gniazdo RJ-12 i ICSP



Rys 4.5: Sposób podłączenia gniazda RJ-12 do mikrokontrolera [5].

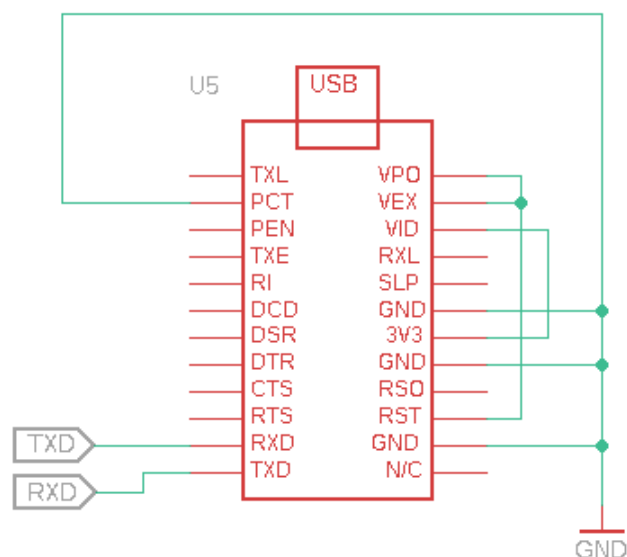


Rys 4.6: Schemat ideowy podłączenia gniazda RJ-12.

Gniazdo zostało połączone zgodnie z dokumentacją płytki rozwojowej EVBeasyPIC [7] oraz schematem z dokumentacji mikrokontrolera PIC16F1789 przedstawionym na rys. 4.5. Programowanie odbywa się za pomocą synchronicznego interfejsu szeregowego. Układ w trakcie programowania musi być w stanie resetu, stąd podłączona do niego linia RESET. ICSPDAT to linia danych, a ICSPCLK to sygnał zegarowy generowany przez programator taktujący transmisję między programatorem a MCU, generowanym przez programator.



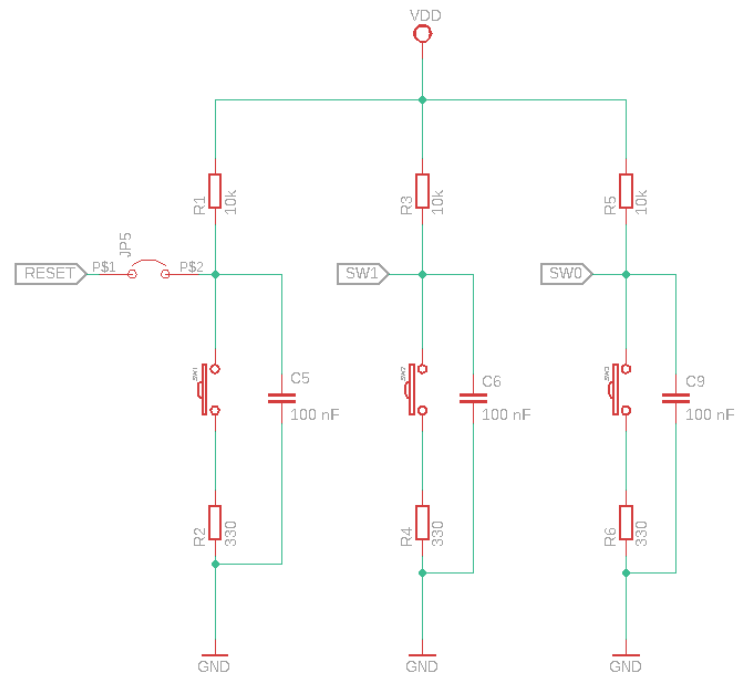
#### 4.2.4. Układ MMUSB232



Rys 4.7: Schemat ideowy podłączenia układu MMUSB232.

Układ został podłączony zgodnie z dokumentacją układu MMUSB232 firmy Propox [8]. Układ zasilany jest przez interfejs USB z komputerem PC. Wyprowadzenie PCT musi być podłączone do masy w przypadku zasilania przez interfejs USB, dla zewnętrznego zasilania zaś do tego zasilania. Wyprowadzenia GND muszą być podłączone do masy. VPO to zasilanie z interfejsu USB, czyli 5 V, VEX ustala główne źródło zasilania, przy zasilaniu z interfejsu USB musi być podłączone do VPO. Wyprowadzenie RST może być albo podłączone do przycisku resetu lub jest nie jest używane, to musi zostać podłączone do napięcia zasilania. VIO ustala napięcie na pinach komunikacyjnych, dlatego połączone jest z wyprowadzeniem 3V3, które jest wyjściem wewnętrznego regulatora LDO obniżającego napięcie zasilania z 5 V do 3.3 V. Po dołączeniu układu do komputera PC przez interfejs USB jest on automatycznie wykrywany przez system i instalowany jest sterownik do jego obsługi. Komunikacja po stronie komputera PC odbywa się za pomocą dowolnego terminala, który współpracuje z interfejsem RS-232.

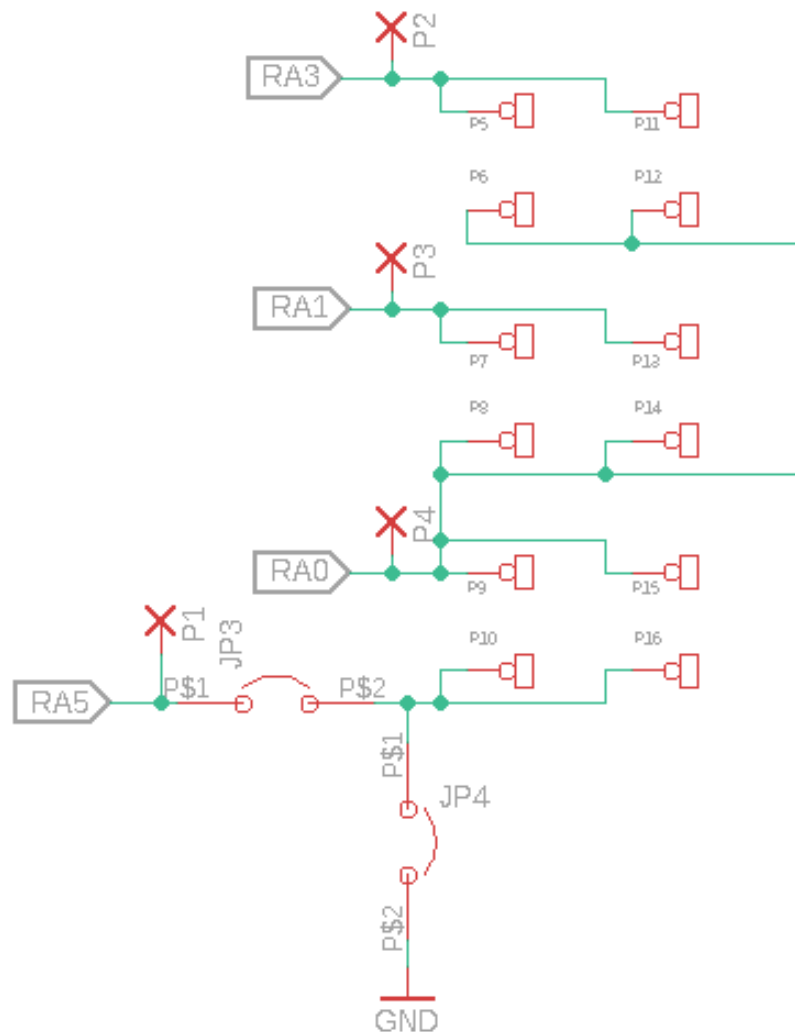
### 4.2.5. Przyciski sterujące oraz przycisk RESETU



Rys 4.8: Schemat ideowy podłączenia przycisków.

Przyciski zostały podłączone przez rezystory 10 k $\Omega$  (tzw. rezystory podciągające) do napięcia zasilania w celu ograniczenia prądu i uniknięcia nieustalonego stanu na wejściach cyfrowych w trakcie, gdy przyciski nie są przyciśnięte. Drgania zestyków są minimalizowane w sposób sprzętowy przez wykorzystanie filtrów RC. W momencie wciśnięcia przycisku odpowiednie wyprowadzenie mikrokontrolera zwierane jest do masy, co powoduje ustalenie się na nim stanu niskiego. Stan wejścia cyfrowego można odczytać z rejestrów PORTx mikrokontrolera.

#### 4.2.6. Gniazda kołkowe żeńskie i męskie



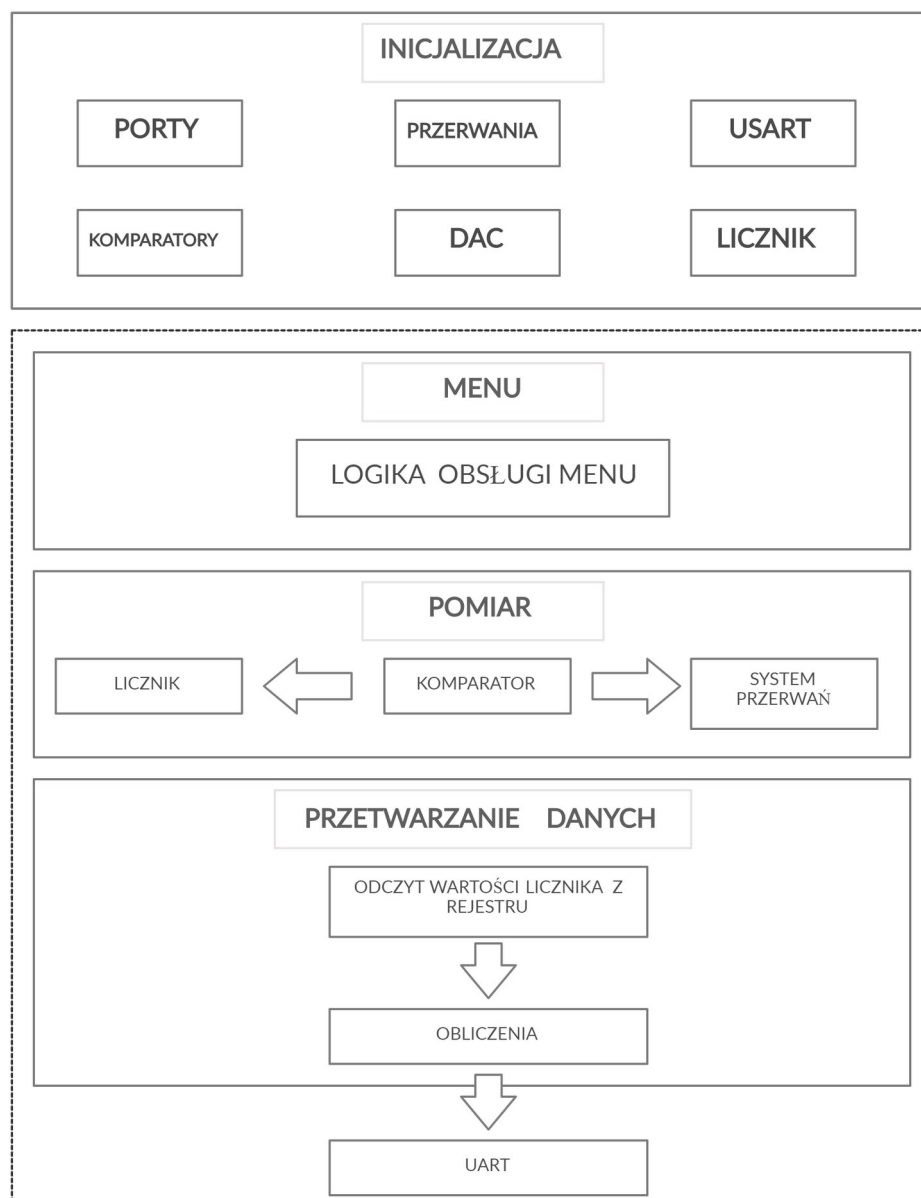
Rys 4.9: Schemat ideowy podłączenia gniazd kołkowych.

Aby układ był bardziej uniwersalny, w projekcie uwzględniono większą liczbę gniazd podłączonych do linii mikrokontrolera niż jest to wymagane przez metodę pomiarową. W niniejszej pracy wykorzystane zostały gniazda P11 i P13, aby wpiąć rezystor wzorcowy oraz gniazda P7 i P14 aby dołączyć mierzony kondensator. Wyprowadzenie RA3 służy do wystawienia impulsu pobudzającego, RA1 to ujemne wejście komparatora. RA0 służy do wystawienia stanu niskiego.

## 5. Oprogramowanie mikrokontrolera

W tym rozdziale zostanie zaprezentowana i omówiona struktura oprogramowania. Przedstawione zostaną również szczegółowo funkcje implementujące procedurę pomiarową i komunikację z komputerem PC. Oprogramowanie zostało napisane w środowisku MPLAB IDE dedykowanym dla mikrokontrolerów firmy Microchip, zaś układ programowany i testowany był z użyciem programatora/debuggera ICD4.

### 5.1. Struktura oprogramowania



Rys 5.1: Uproszczona struktura oprogramowania.

Na rys. 5.1. przedstawiona jest uproszczona struktura oprogramowania. Po uruchomieniu mikrokontrolera następuje inicjalizacja układu. Konfigurowane są porty, obsługa przerwań, komparatory analogowe, przetwornik cyfrowo-analogowy, licznik i interfejs USART. Po tym zostaje wysyłane przez interfejs USART do komputera PC menu, z którego użytkownik dokonuje wyboru odpowiedniego polecenia za pomocą znaków przesyłanych z klawiatury komputera PC przez terminal. Gdy zostaje uruchomiony pomiar, włączane są potrzebne do niego układy peryferyjne, generowany jest impuls prostokątny ładujący czujnik, a komparator porównuje jego odpowiedź z napięciem odniesienia. Po przekroczeniu wartości napięcia odniesienia przez sygnał odpowiedzi czujnika wysyłany jest sygnał do bramki, która zatrzymuje licznik. W obsłudze przerwania od komparatora analogowego ustawiana jest flaga, która sygnalizuje koniec procedury pomiarowej. Następnie zatrzymywane są układy peryferyjne i następuje odczyt wartości licznika, i wykonywane są obliczenia. Następnie wynik wysyłany jest w odpowiedniej postaci do komputera PC przez interfejs USART.

## **5.2. Inicjalizacja**

Inicjalizacja realizowana jest przez funkcję *void init\_pic(void)*. Po kolei ustawiane są rejestry odpowiedzialne za skonfigurowanie pracy oscylatora, portów wejścia/wyjścia i ich stanu początkowego, obsługi przerwań, interfejsu USART, komparatorów analogowych, przetwornika cyfrowo-analogowego oraz licznika.

### **5.2.1. Oscylator**

Oscylator ustawiony jest, jako wewnętrzny o częstotliwości 16 MHz przez wpisanie do rejestru OSCCON wartości 0x78.

## 5.2.2. Porty

```
//ports
TRISA=0xF6; //RA3, RA0 output
TRISE=0xE7; //RB4 and RB3 outputs - leds
TRISC=0xBF; //RC6 output - TxD
TRISD=0xFF; //port D as input
TRISE=0xFF; //port E as input

ANSELA=0x02; //RA1 analog
ANSELB=0x00;
ANSELC=0x00;
ANSELD=0x00;
ANSELE=0x00;

LATAbits.LATA3=0; //dont generate pulse at beginning
LATAbits.LATA0=0; //ground
LATBbits.LATB3=1; //switch off leds
LATBbits.LATB4=1; //switch off leds
```

Listing 5.1: Inicjalizacja portów.

Wyprowadzenie, pod które podłączone jest jedno z wejść komparatora analogowego, ustawione jest jako wejście analogowe, poza tym wszystkie porty są ustawione jako linie cyfrowe za pomocą rejestrów ANSELx. Jako wyjścia za pośrednictwem rejestrów TRISx zostały ustawione linie od wyjścia układu USART, a także linie dołączone do diod i piny RA3 i RA0, na których podawany jest impuls i poziom masy w trakcie pomiaru. W tym fragmencie kodu programu ustawiane są również wartości początkowe dla pinów przez rejestry LATx. Diody są wyłączone przez podanie stanu wysokiego na liniach RB3 i RB4, a na linii RA3 jest stan niski.

## 5.2.3. Obsługa przerwań

```
//interrupts
INTCON=0xD8; //GIE=1, PEIE=1, INTE=1, IOCIE=1, INTF=0, IOCIF=0

PIE1=0x20; //RCIE=1, received interrupt on
PIE2=0x20; //C1IE=1, comparator interrupt on
PIE3=0x00;
PIE4=0x00;

PIR1=0x00; //USART RECEIVE FLAG
PIR2=0x00; //COMPARATOR 1 FLAG

IOCBP=0x00;
IOCBN=0x06; // interrupt on change for RB1 and RB2
IOCBF=0x00; //interrupt on change flag
```

Listing 5.2: Inicjalizacja przerwań.

Aby odblokować przerwania, należy ustawić port INTCON. Bit GIE to globalne odblokowanie przerwań, PEIE to odblokowanie przerwań od urządzeń peryferyjnych. IOCIE to odblokowanie tzw. interrupt on change, które zostało zaimplementowane do testowania przycisków SW1 i SW2. W rejestrach PIRx ustawiane są przerwania dla konkretnych urządzeń peryferyjnych. W niniejszej pracy jest to przerwanie od odbioru danej przez USART oraz przerwanie od komparatora analogowego. Rejestry PIRx zawierają flagi przerwań, które w trakcie inicjalizacji powinny zostać wyzerowane. Rejestry IOCBP, IOCBN konfigurują interrupt on change, a rejestr IOCBF to rejestr z flagami dla IOC.

## 5.2.4. USART

```
//usart
TXSTA=0x20; //TX9=0, TXEN=1, SYNC=0, SENDB=0, BRGH=0,
RCSTA=0x90; //SPEN=1, RX9=0, CREN=1
BAUDCON=0x00;
SPBRG=((_XTAL_FREQ/64)/9600) - 1; // OSC/BAUD_RATE/64 - 1 (25 dec) (19 hex)
```

Listing 5.3: Inicjalizacja USART.

USART ustawiony jest do 8-bitowej asynchronicznej transmisji o prędkości 9600 bit/s. SPBRG określa prędkość transmisji na podstawie wartości wyliczonej ze wzoru (3.1). Takie same parametry transmisji muszą być ustawione w terminalu uruchomionym na komputerze PC.

## 5.2.5. Komparatory

```
//comparators
CM2CON0bits.C2ON=0;
CM3CON0bits.C3ON=0;
CM4CON0bits.C4ON=0; // CM2, CM3, CM4 disabled

CM1CON0=0x14; //CM1 disabled(enabled 80), polarity inverted,
               //internal output, latency filter disabled,
               //hysteresis disabled, normal power high speed, asynchronous
CM1CON1=0xA9; //rising edge interrupt, + DAC1 output, - C1IN1-(RA1)
```

Listing 5.4: Inicjalizacja komparatorów.

Aby uniknąć zbyt dużego poboru energii, komparatory muszą zostać wyłączone. Komparator konfigurowany jest przy pomocy rejestrów CM1CON0 i CM1CON1. Polaryzacja komparatora analogowego została odwrócona, ponieważ wartość referencyjna podana jest na dodatnie wyprowadzenie a odpowiedź czujnika na ujemne. Wówczas narastające zbocze na wyjściu komparatora pojawia się, gdy na wejściu ujemnym jest większe napięcie niż wejściu dodatnim. Komparator działa asynchronicznie względem licznika, wyłączony został filtr latencji oraz histereza. Prędkość działania została ustawiona jako szybka z normalnym poborem

energii. Przerwanie generowane jest tylko przy narastającym zboczu na wyjściu komparatora analogowego.

### 5.2.6. Przetwornik cyfrowo-analogowy

```
//dac
DAC1CON0=0xA0; //DAC1 enabled(DAC1EN=1), external output RA1,
               //Positive Vdd, Negative Vss
DAC1CON1=DAC; //DAC1 output voltage
```

Listing 5.5: Inicjalizacja DAC.

Do wyprowadzeń  $V_{source+}$  i  $V_{SOURCE-}$  dołączone jest napięcie zasilania i masa. Poprzez rejestr DAC1CON1 ustawiana jest wartość napięcia na wyjściu DAC1. Wyjście przetwornika C/A dołączone jest do zewnętrznego wyprowadzenia RA1 i zostało użyte podczas pomiarów charakterystyki przetwornika.

### 5.2.7. Licznik

```
//timer
T1CON=0x44; //system clock, prescaler 1:1, dedicated circuit disabled,
            //asynchronous, timer stop
T1GCON=0x82; // timer 1 gate disabled, counting when gate is low,
            //gate source is rising edge of comparator1
TMR1L=0x00;
TMR1H=0x00; //clear timer1
```

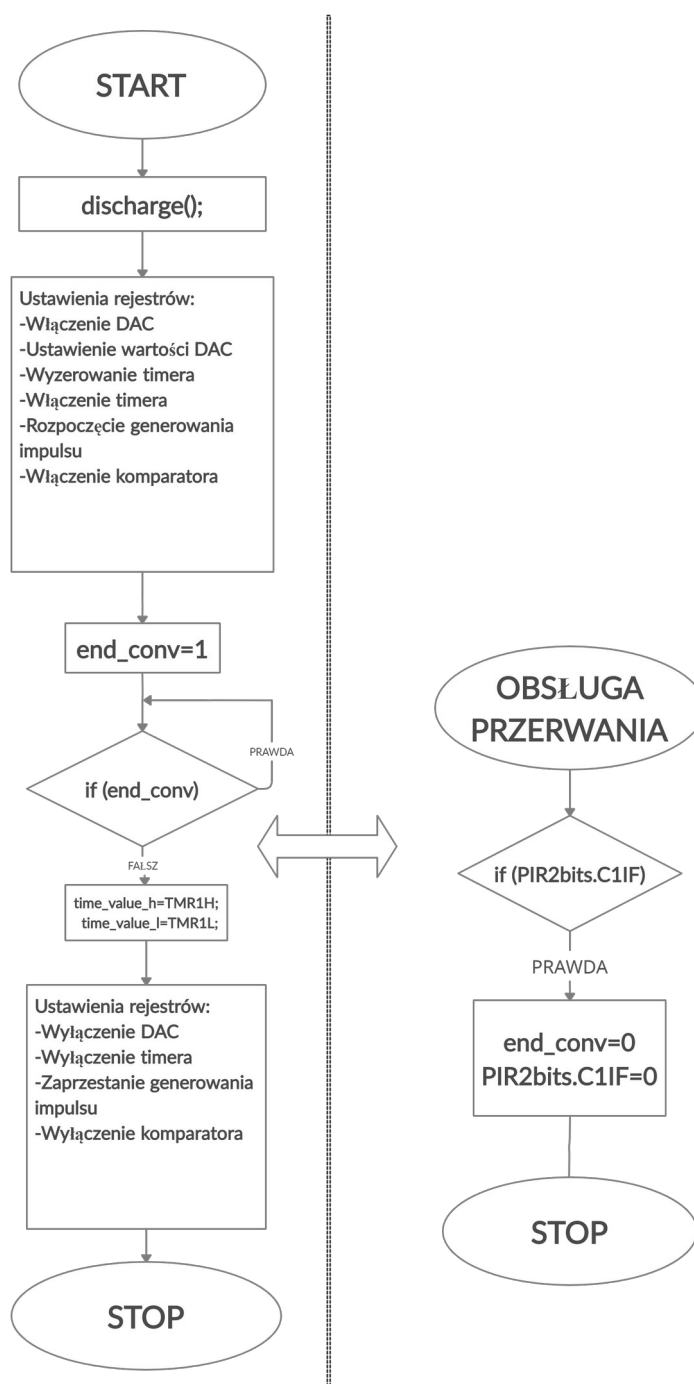
Listing 5.6: Inicjalizacja licznika.

Za pomocą rejestru T1CON ustawiane są parametry timera. Dla jak największej rozdzielczości pomiaru dzielnik częstotliwości jest wyłączony, tak samo, jak dedykowany obwód taktujący, który działa na częstotliwości 32 kHz. W rejestrze T1GCON ustawione są parametry bramki, która zatrzymuje pomiar czasu na narastające zbocze od komparatora analogowego przez wewnętrzny sygnał SYNC1\_out. Rejestry TMR1L i TMR1H przechowują odpowiednio 8 mniej znaczących bitów i 8 bardziej znaczących bitów licznika, więc w trakcie inicjalizacji są zerowane.



### 5.3. Funkcja pomiarowa

Funkcja pomiarowa wywoływana jest w głównej pętli programu. W funkcji wykonywany jest pomiar i zapisywany wynik, czyli zawartość licznika 1. Schemat algorytmu funkcji przedstawiony jest na rys. 5.8.



Rys 5.2: Diagram algorytmu funkcji pomiarowej.

```

//mesurment functions
void measurmentT(void)
{
    discharge();
    DAC1CON0bits.DAC1EN=1; //DAC1 enable
    DAC1CON1=DAC; //DAC1 output voltage

    TMR1L=0x00;
    TMR1H=0x00; //clear timer1

    T1CONbits.TMR1ON=1; //switch on timer1
    LATAbits.LATA3=1; //start generating pulse
    CM1CON0bits.C1ON=1; //comparator on

    end_conv=1;
    while(end_conv);

    time_value_h=TMR1H;
    time_value_l=TMR1L;
    T1CONbits.TMR1ON=0; //switch off timer1
    LATAbits.LATA3=0; //stop generating pulse
    CM1CON0bits.C1ON=0; //comparator off
    DAC1CON0bits.DAC1EN=0; //dac off

    return;
}

```

Listing 5.7: Funkcja pomiarowa.

```

//interrupt services
void __interrupt() ISR(void)
{
    //comparator
    if(PIR2bits.C1IF)
    {
        end_conv=0; //after rising edge on comparator
                    //gate stop timer, interrupt from
                    //comparator stops conversion
        PIR2bits.C1IF=0; //clear flag
    }
}

```

Listing 5.8: Obsługa przerwania od komparatora.

Na początku czujnik pojemnościowy jest rozładowywany przez czas większy niż czas najdłuższego pomiaru w zakładanym zakresie pomiarowym (500  $\mu$ s) za pomocą funkcji *void discharge(void)*. Następnie uruchamiany jest DAC i ustawiana jest wartość napięcia referencyjnego na jego wyjściu, zerowane są rejestry przechowujące zawartość licznika. Następuje uruchomienie licznika, rozpoczyna się generowanie impulsu pobudzającego i uruchomienie komparatora analogowego. W tym punkcie program zatrzymuje się w pętli while,

do czasu wyzerowania zmiennej `end_conv` w obsłudze przerwania od komparatora analogowego. Po zakończeniu pomiaru wartości rejestrów TMR1H i TMR1L zapisywane są w zmiennych, licznik jest wyłączany, impuls przestaje być generowany, wyłączany jest także komparator i DAC. Funkcja pomiarowa i obsługa przerwania przedstawione są na rys. 5.9 oraz rys. 5.10.

## 5.4. Obliczenia

```
//calculate functions
double calculateT(double t1, double Rr, double log)
{
    double Cx;

    Cx=(double) (t1*T_clk);
    Cx=(double) Cx/ (Rr*log);
    Cx=Cx*0.001;

    return Cx;
}

double calculateCcorr(double t1, double A, double B, double C, double D)
{
    double Ccorr;
    double time=t1*T_clk*0.001;

    Ccorr=D;
    Ccorr=Ccorr+C*time;
    Ccorr=Ccorr+B*time*time;
    Ccorr=Ccorr+A*time*time*time;

    return Ccorr;
}
```

Listing 5.9: Funkcje obliczeniowe.

Funkcja *calculateT* jest bezpośrednią implementacją wzoru 2.11. Do funkcji podawana jest zmierzona przez licznik wartość  $t1$ , następnie przemnożona przez okres  $T_{clk}$ , a także wartość użytego rezystora  $Rr$  i wartość logarytmu naturalnego  $log$ . Funkcja zwraca obliczoną pojemność w pF.

```

#define _XTAL_FREQ 16000000

//constant values
#define T_clk          62.5 // [ns]
#define V_pulse        3.3 // [V]
#define R_r            1.00212 // [MOM] 1MOM < R_r < 10 MOM
#define V_threshold    2.18 // 0.66*V_pulse
#define DAC            170 // DAC register value for mesurment, 2.18 V
#define LOG_VALUE      1.080593783165 // -ln(1 - V_threshold/V_pulse)
#define WSP_A          -6.5643818437e-7
#define WSP_B          0.0029126242209
#define WSP_C          -1.2504845490698
#define WSP_D          162.6680448334

```

Listing 5.10: Stałe użyte w obliczeniach.

Stałe użyte w pracy zostały wyliczone na podstawie wcześniej omówionych wzorów w podrozdziale 2.2, do obliczeń przyjęto następujące dane:

$$P_{max}=1:1=1 \quad (5.1)$$

$$N_{min}=2^8=256, N_{max}=2^{16}=65536 \quad (5.2)$$

$$V_{in}=3.3 V \quad (5.3)$$

$$V_{ccscaled}=0.66 \cdot 3.3 V = 2.178 V \approx 2.18 V \quad (5.4)$$

$$T_{clk} = \frac{1}{16 MHz} = 62.5 ns \quad (5.5)$$

$$C_{min}=100 pF, C_{max}=220 pF \quad (5.6)$$

Korzystając z wzorów (2.8), (2.9), (2.10) obliczono zakres dla  $R_r$ :

$$R_{rmin} = \frac{-256 \cdot 62.5 ns}{100 pF \cdot \ln\left(1 - \frac{2.18 V}{3.3 V}\right)} = 148070 \Omega$$

$$R_{rmax} = \frac{-65536 \cdot 62.5 ns \cdot 1}{220 pF \cdot \ln\left(1 - \frac{2.18 V}{3.3 V}\right)} = 17230000 \Omega$$

Na podstawie dokonanych obliczeń wybrano do pomiarów rezystor wzorcowy 1.00212 MΩ.

Do programu należy również wpisać wartość logarytmu naturalnego, wpisanie jej jako stałej przyspiesza pracę programu, ponieważ nie musi być wyliczona w trakcie jego pracy. Została obliczona w następujący sposób:

$$\log = -\ln\left(1 - \frac{2.18\text{ V}}{3.3\text{ V}}\right) = 1.080593783165$$

Funkcja `calculateCcorr` wyznacza współczynnik korekcji, który jest zwracany przez funkcję w pF. Obliczany jest on na podstawie podanych współczynników. Współczynniki zostały wyliczone dzięki wykreśleniu krzywej regresji na podstawie błędów pomiarowych w podrozdziale 6.3. Wykres przedstawiający błędy pomiarowe bez korekcji i krzywą to rys. 6.5.

## 5.5. Komunikacja po stronie mikrokontrolera

```
//usart functions
void WriteUSART(char data)
{
    while(!PIR1bits.TXIF); // Is the transmit shift register empty
    TXREG = data;           // Write the data byte to the USART

    return;
}

void WriteStringUSART(char* data)
{
    do
    { // Transmit a byte
        while(!TXSTAbits.TRMT); // Is the transmit shift register empty
        TXREG = *data;           // Write the data byte to the USART
    } while( *data++ );

    return;
}

void WriteStringCUSART(const char* data)
{
    do
    { // Transmit a byte
        while(!TXSTAbits.TRMT); // Is the transmit shift register empty
        TXREG = *data;           // Write the data byte to the USART
    } while( *data++ );

    return;
}
```

Listing 5.11: Funkcje do wysyłania danych przez USART.

Na rys. 5.13 są przedstawione trzy funkcje. Pierwsza służy do wysłania pojedynczego bajtu, czyli 8 bitów danych, ponieważ tyle mieści rejestr TXREG, który stanowi w tym wypadku bufor wyjściowy układu interfejsu USART. Druga funkcja odpowiada za wysłanie ciągu bajtów. Ostatnia to ta sama funkcja, ale służąca do wysłania ciągu stałych znaków.

```

char ReadUSART(void)
{
    while(!PIR1bits.RCIF); // Is the receive shift register empty
    return RCREG; // Return the received data
}

```

Listing 5.12: Funkcja do odbioru danych przez USART.

Funkcja na rys. 5.14 służy do odczytu danych. Program czeka aż rejestr przesuwany RCREG wypełni się, a następnie zwraca jego wartość. Odbiór danej uruchamia obsługę przerwania przedstawioną na rys. 5.15.

```

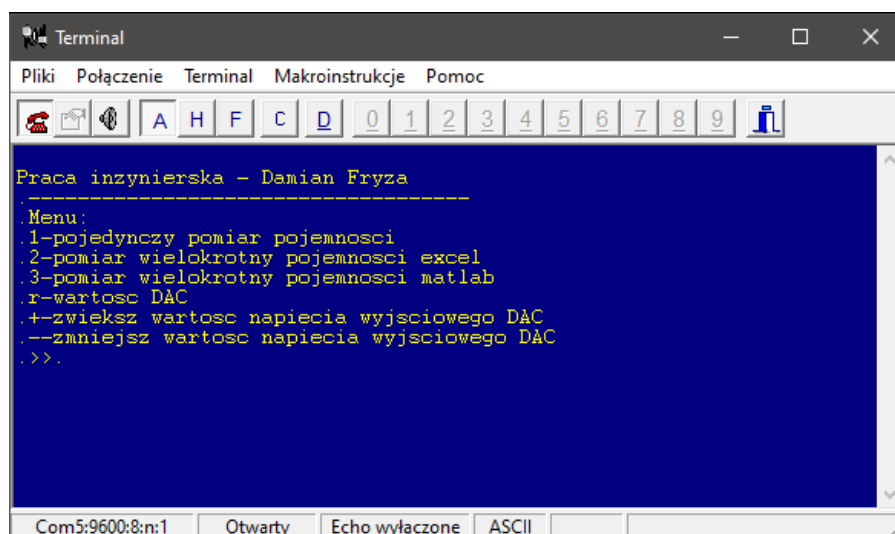
//usart
if(PIR1bits.RCIF)
{
    usart_data = ReadUSART(); //write received data to usart_data
    WriteUSART(usart_data);
    receive=1; //set receive for main function
    PIR1bits.RCIF=0; //clear flag
}

```

Listing 5.13: Obsługa przerwania od odbioru danej przez interfejs USART.

W obsłudze przerwania po odbiorze danej uruchamiana jest funkcja odbioru danej i odebrany bajt zapisywany jest w zmiennej *usart\_data*. Następnie odebrany bajt przesyłany jest z powrotem do komputera PC. Ustawiana jest zmienna *receive*, dzięki temu w głównej pętli programu sprawdzane są warunki i w przypadku odebrania znaku odpowiadającego za uruchomienie którejś z opcji w menu, uruchamiana jest odpowiednia funkcja programu (np. pomiar). Następnie zerowana jest flaga przerwania.

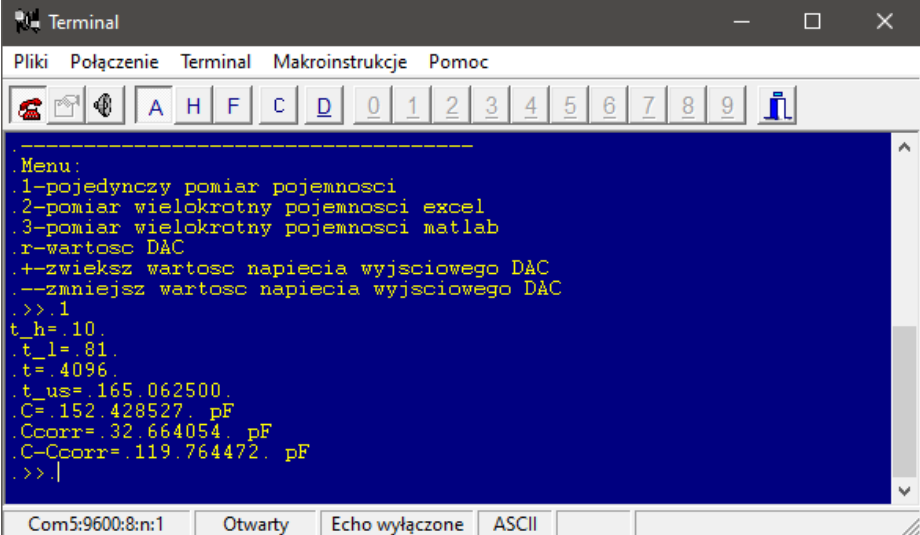
## 5.6. Komunikacja po stronie PC



Rys 5.3: Stan terminala po resecie układu.

Komunikacja z płytką pomiarową możliwa jest za pomocą dowolnego terminala instalowanego na komputerze PC, który umożliwia komunikację przez interfejs RS-232. Dostępne opcje zostają wypisane na ekranie.

### 5.6.1. Pojedynczy pomiar:



```
Terminal
Pliki Połączenie Terminal Makroinstrukcje Pomoc
[Icons] [A] [H] [F] [C] [D] [0] [1] [2] [3] [4] [5] [6] [7] [8] [9] [Enter]

-----
Menu:
.1-pojedynczy pomiar pojemnosci
.2-pomiar wielokrotny pojemnosci excel
.3-pomiar wielokrotny pojemnosci matlab
.r-wartosc DAC
.+-zwiększ wartosc napiecia wyjsciowego DAC
.--zmniejsz wartosc napiecia wyjsciowego DAC
.>>.1
t_h=.10.
t_l=.81.
t=.4096.
t_us=.165.062500.
C=.152.428527. pF
Ccorr=.32.664054. pF
C-Ccorr=.119.764472. pF
.>>.
```

Rys 5.4: Stan terminala po pojedynczym pomiarze.

Po wysłaniu „1” wykonuje się pojedynczy pomiar. Na ekranie zostają wypisane takie informacje jak stan obu rejestrów przechowujących wartość licznika, czas pomiaru w  $\mu$ s, zmierzona pojemność, współczynnik korekcji i zmierzona pojemność po zastosowaniu korekcji. Na rys. 32 przedstawiony został pojedynczy pomiar kondensatora o pojemności 119.82 pF.

### 5.6.2. Pomiar wielokrotny pojemności.

Po przesłaniu „2” lub „3” zostaje wykonany wielokrotny pomiar pojemności. Umożliwia to zapis danych bezpośrednio do pliku w trzech kolumnach (pojemność, pojemność po korekcji, czas) lub jako macierz X zgodnie ze składnią Matlaba o trzech kolumnach i dwustu czterdziestu wierszach.

### 5.6.3. Pomiar napięcia na wyjściu DAC

Przesłanie „r” powoduje wypisanie aktualnej wartości rejestru ustalającego wartość napięcia na wyjściu DAC. „+” i „-” służą do przeprowadzenia pomiarów charakterystyki przetwarzania przetwornika DAC.

## 6. Badanie eksperymentalne

W tym rozdziale opisane są testy wewnętrznego oscylatora kwarcowego, przetwornika cyfrowo-analogowego oraz metody pomiarowej. Wyniki pomiarów zostały ze sobą zestawione, opracowane i podsumowane.

### 6.1. Test wewnętrznego oscylatora kwarcowego

```
start:
banksel OSCCON
movlw 0x7B ;internal oscyllator 16 MHz
movwf OSCCON
banksel TRISB
clrf TRISB ; porty b jako wyjścia
banksel LATB
movlw 0xFF

loop:

clrf LATB
clrf LATB
clrf LATB
clrf LATB
clrf LATB

movwf LATB

clrf LATB

movwf LATB

clrf LATB

movwf LATB

clrf LATB

movwf LATB

clrf LATB

movwf LATB;5

clrf LATB
clrf LATB
clrf LATB
clrf LATB
clrf LATB

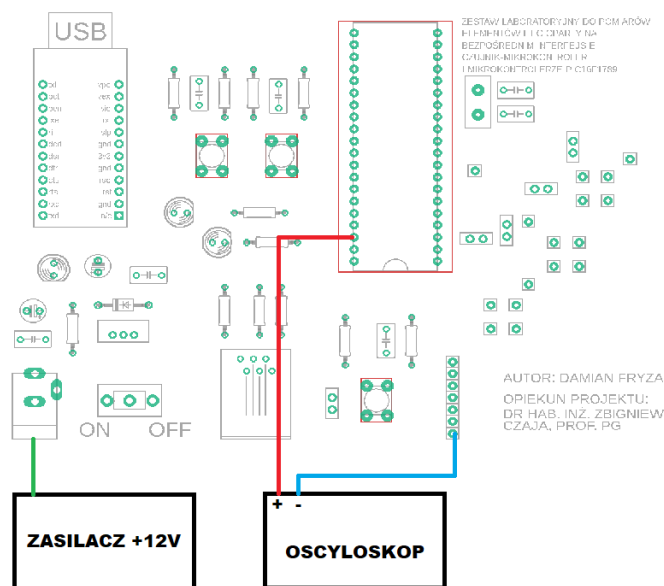
goto loop

end
```

Listing 6.1: Program do pomiaru częstotliwości oscylatora napisany w assemblerze.



Do pomiaru częstotliwości i obserwacji kształtu charakterystyki został wykorzystany oscyloskop. Podłączenie oscyloskopu do pomiaru zostało przedstawione na rys. 6.2.



Rys 6.1: Schemat podłączenia oscyloskopu do układu dla pomiaru częstotliwości zegara.

Zgodnie z dokumentacją mikrokontrolera PIC [5] wewnętrzny oscylator został fabrycznie skalibrowany, tak by odchyłka jego wartości częstotliwości była na poziomie  $\pm 1\%$ . Za pomocą programu włączającego i wyłączającego diodę napisanego w assemblerze, możliwe było zmierzenie przy pomocy oscyloskopu przebiegu napięcia generowanego na wyprowadzeniu RB3. Czas wykonywania użytych w pętli instrukcji, czyli *clrf* oraz *movwf*, to jeden okres zegara instrukcji [10], którego częstotliwość to jedna czwarta częstotliwości zegara systemowego ( $F_{osc}/4$ ). Na podstawie pomiarów długości oglądanego na oscyloskopie przebiegu została obliczona częstotliwość  $F_{osc}$ , następnie została porównana z wartością dokładną i obliczony został błąd względny.

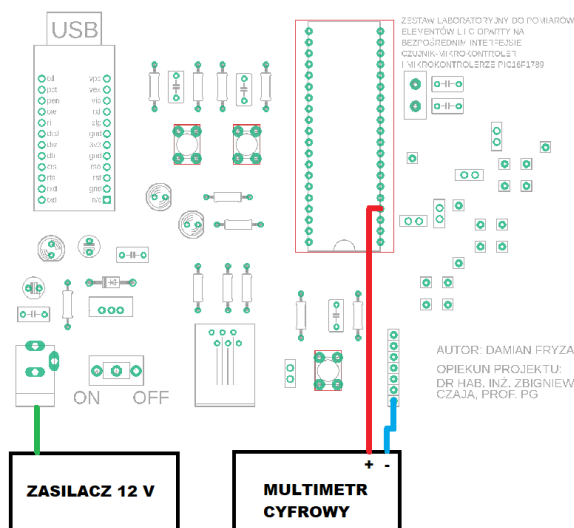
Tabela 6.1. Pomiar błędu wewnętrznego oscylatora kwarcowego

F_teoretyczna [MHz]	F_zmierzona [MHz]	Błąd [%]
16	15.9	0.7
32	31.8	0.7

Pomiary wykazały zgodność zmierzonej częstotliwości z informacją zamieszczoną w dokumentacji, odchyłka częstotliwości zmierzonej od teoretycznej wewnętrznego oscylatora jest mniejsza niż 1%.

## 6.2. Test DAC

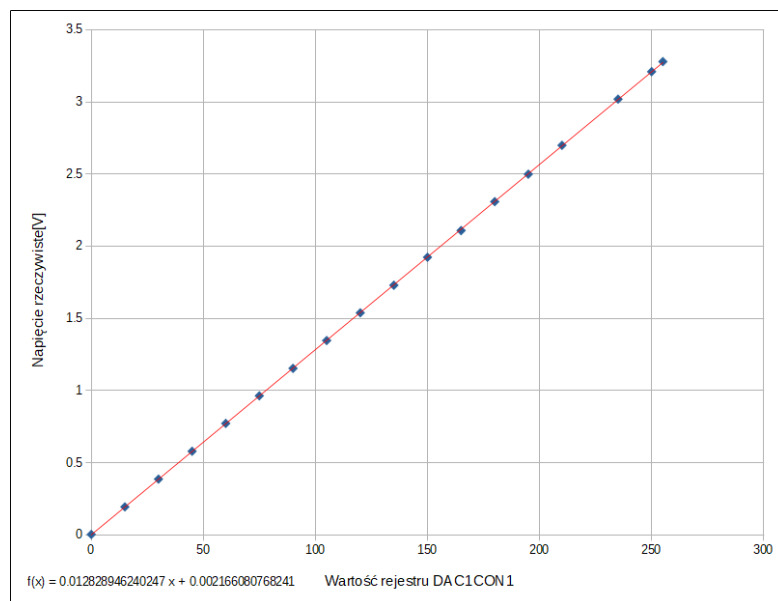
Przetestowano przetwornik cyfrowo-analogowy. Przy pomocy multimetru mierzono napięcie na wyprowadzeniu mikrokontrolera, do którego dołączone jest wyjście przetwornika cyfrowo-analogowego. Schemat podłączenia miernika został przedstawiony na rys. 6.4.



Rys 6.2: Podłączenie multimetru do testu DAC.

Tabela 6.2. Pomiar błędu napięcia  
wyjściowego przetwornika DAC.

DAC1CON1	Napięcie teoretyczne [V]	Napięcie rzeczywiste [V]	Błąd względny [%]
0	0.000	0.005	100.0
15	0.193	0.196	1.3
30	0.387	0.388	0.3
45	0.580	0.581	0.2
60	0.773	0.774	0.1
75	0.967	0.965	-0.2
90	1.160	1.155	-0.4
105	1.354	1.348	-0.4
120	1.547	1.54	-0.4
135	1.740	1.731	-0.5
150	1.934	1.925	-0.4
165	2.127	2.11	-0.8
180	2.320	2.31	-0.4
195	2.514	2.5	-0.5
210	2.707	2.7	-0.3
235	3.029	3.02	-0.3
250	3.223	3.21	-0.4
255	3.287	3.28	-0.2

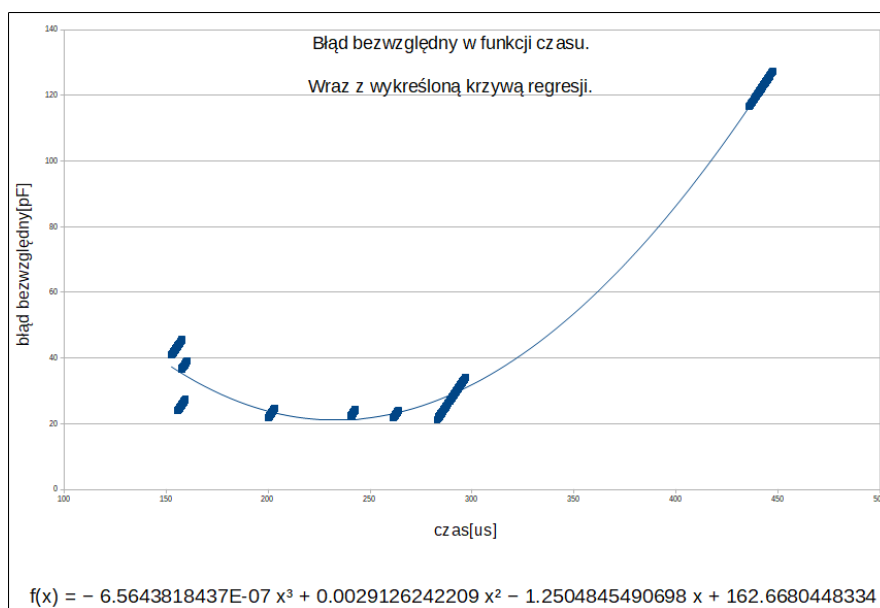


Rys 6.3. Wartości zmierzonego napięcia wyjściowego w funkcji wartości teoretycznych wyliczonych na podstawie wzoru.

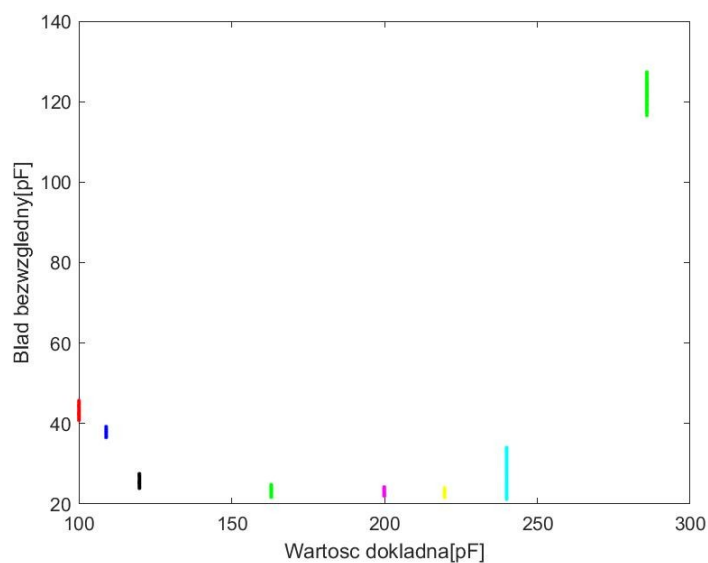
Ze względu na skończoną rozdzielczość przetwornika (8 bitów) wartości ustawione w rejestrze konfiguracyjnym nie odpowiadają całkowicie wartościom teoretycznym wynikających ze wzoru. Błąd względny między wartością teoretyczną a rzeczywistą, zmierzona na wyprowadzeniu mikrokontrolera multimetrem cyfrowym, dla większości pomiarów nie przekracza 1%. Z wykreślonej charakterystyki wynika, że wartości narastają w sposób prawie liniowy, co zostało przybliżone za pomocą krzywej regresji  $f(x)$ , której wzór zamieszczony jest pod wykresem (rys. 6.3).

Wartość napięcia wyjściowego w zależności od wartości rejestru konfiguracyjnego zostały obliczone ze wzoru 3.3.

### 6.3. Test poprawności pomiaru



Rys 6.4. Błąd bezwzględny w funkcji czasu.

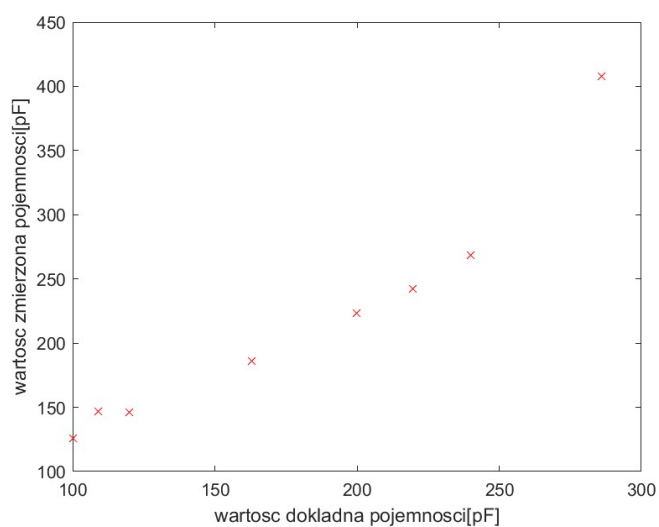


Rys 6.5. Przedstawienie rozstrzału błędów bezwzględnych dla pojemności z zakresu od 100 pF do 300 pF.

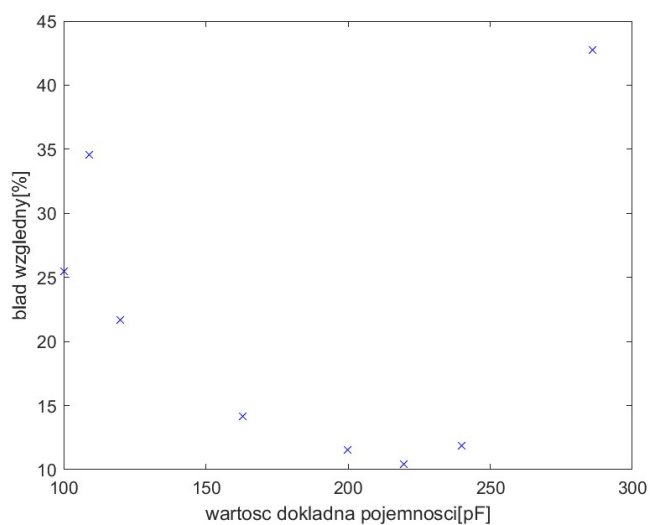
Wykonano 240 pomiarów dla każdego z 8 kondensatorów o wartościach: 100.06 pF, 108.94 pF, 119.94 pF, 162.97 pF, 199.94 pF, 219.65 pF, 239.97 pF oraz 285.85 pF. Na podstawie wykresu rys. 6.5 wyznaczono krzywą regresji o podanej pod wykresem funkcji  $f(x)$ . Dzięki krzywej możliwe jest wyliczenie współczynnika  $C_{corr}$  i programowa korekta wartości. W tabeli 6.3 zestawiono uśrednione pomiary i wpływ współczynnika korekcji na błąd pomiarowy.

Tabela 6.3. Uśrednione wyniki pomiarów

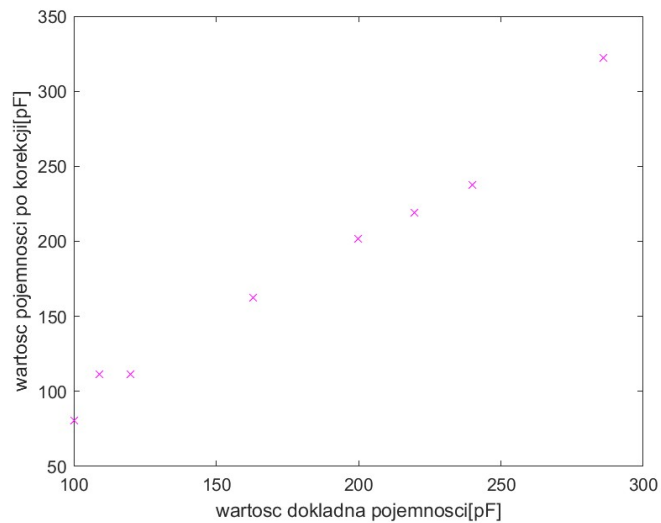
Wartość dokładna C [pF]	Wartość zmierzona C [pF]	Błąd względny C [%]	Współczynnik korekcji C <sub>corr</sub> [pF]	C po korekcji [pF]	Błąd względny C po korekcji [%]
100.06	125.59	25.5	44.82	80.77	19.3
108.94	145.56	33.6	35.36	110.20	-1.2
119.82	147.98	23.5	34.37	113.61	5.2
162.97	185.90	14.1	23.61	162.29	0.4
199.94	220.94	10.5	21.22	199.72	0.1
219.65	242.70	10.5	23.29	219.41	0.1
239.97	273.86	14.1	30.86	243.00	-1.3
285.85	398.75	39.5	85.53	282.70	1.1



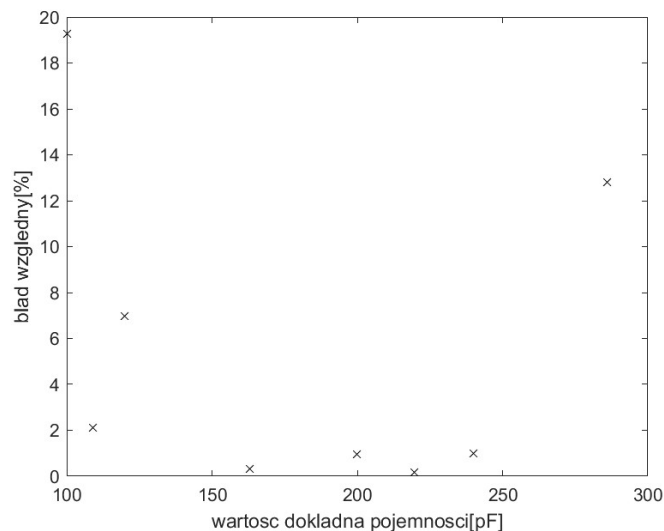
Rys 6.6: Wartość zmierzona C w funkcji wartości dokładnej pojemności.



Rys 6.7: Błąd względny dla wartości zmierzonej C w funkcji wartości dokładnej.



Rys 6.8: Wartość pojemności po korekcji w funkcji wartości dokładnej.



Rys 6.9: Błąd względny pojemności po korekcji w funkcji wartości dokładnej.

Na podstawie wykonanych pomiarów można wyciągnąć wniosek, że zrealizowany układ najlepiej działa w zakresie od 108.94 pF. Porównując wykresy błędów względnych rys. 6.8 i rys. 6.10. można wywnioskować, że korekcja znacznie obniżyła błąd pomiaru, wynikający z ograniczonej rozdzielczości licznika, błędów obciążenia i zaokrąglenia, i przede wszystkim z pojemności pasożytniczych wyprowadzeń mikrokontrolera, ścieżek obwodu drukowanego i gniazd, do których dołączony był kondensator. Błąd w tym zakresie nie przekracza po korekcji dla większości mierzonych wartości 2%. Największe błędy obserwowane są dla wartości na skraju zakresu pomiarowego. Żeby zminimalizować ten błąd, należałoby dobrać inny rezystor wzorcowy, którego wartość znajdowałaby się w środku wyznaczonego przedziału dla  $R_t$ .

wyznaczonego w podrozdziale 5.4. Niestety użyty mikrokontroler uniemożliwia rozszerzenie zakresu lub użycie innego rezystora. Dla precyzyjniejszych obliczeń a co za tym idzie precyzyjniejszego pomiaru, należałoby wybrać mikrokontroler 16-bitowy lub 32-bitowy, znacznie rozszerzyłoby to możliwości obliczeniowe układu. Zwiększenie rezystora powoduje wydłużenie czasu ładowania pojemności. Obliczenia na tak dużych wartościach czasu stają się niemożliwe i wyniki odbierane przez PC są błędne i za każdym razem inne, więc korekcja programowa również nie jest w tym zakresie możliwa.

## 7. Podsumowanie

W ramach pracy dyplomowej zrealizowano wszystkie założone zadania. Zaprojektowano układ do pomiaru czujników pojemnościowych i indukcyjnych. Następnie wykonano prototyp układu, napisano oprogramowanie na mikrokontroler PIC16F1789 implementujące metodę pomiarową, następnie zweryfikowano działanie metody pomiarowej dla kondensatorów z zakresu od 100 pF do 290 pF.

Układ działa poprawnie z założoną dokładnością, mierzy pojemności z przyjętego zakresu w krótkim czasie. Umożliwia on pomiar pojedynczy oraz wielokrotny pojemności czujników wraz z zapisem wyników w formacie programu Excel lub Matlab. Zestaw laboratoryjny jest obsługiwany za pośrednictwem interfejsu USART z poziomu komputera PC.

W ramach pracy dyplomowej zrealizowano następujące zadania:

- Przegląd literatury i dokumentacji mikrokontrolera.
- Zaprojektowanie i montaż układu.
- Dokonanie wstępnych testów działania układu i naniesienie poprawek.
- Napisanie programu realizującego pomiar, obliczenia i zarządzanie transmisją danych.
- Przeprowadzenie testów metody pomiarowej i opracowanie wyników, a także wyznaczenie funkcji korygującej pomiar i minimalizującej wpływ pojemności pasożytniczych na wynik pomiaru.

Podsumowując, praca została w pełni wykonana a efektem jest działający układ pomiarowy zgodny z przyjętymi założeniami.



## WYKAZ LITERATURY

- (1) Z. Czaja, Time-domain measurement methods for R, L and C sensors based on a versatile direct sensor-to-microcontroller interface circuit, *Sensors and Actuators A* 274 (2018) 199-210.
- (2) Z. Czaja, A method of measuring RLC components for microcontroller systems, *PRZEGLĄD ELEKTROTECHNICZNY*, ISSN 0033-2097, R. 93 NR 10/2017
- (3) F. Reverter, O. Casas, Direct interface circuit for capacitive humidity sensors, *Sensors and Actuators A* 143 (2008) 315-322.
- (4) S. Durawa. Praca dyplomowa: mikrokontrolerowy miernik do identyfikacji parametrycznej wieloelementowych modeli czujników, Politechnika Gdańska, 2014.
- (5) Dokumentacja mikrokontrolera PIC16F1789 (pobrana w dniu 24.04.2020 r.), <https://ww1.microchip.com/downloads/en/DeviceDoc/40001675C.pdf>.
- (6) Dokumentacja kompilatora MPLAB XC8 (pobrana w dniu 01.09.2020 r.), <https://ww1.microchip.com/downloads/en/DeviceDoc/52053B.pdf>
- (7) Dokumentacja płytki rozwojowej EVBeasyPIC\_pl (pobrana w dniu 24.04.2020 r.), [http://www.propox.com/download/docs/EVBeasyPIC\\_pl.pdf](http://www.propox.com/download/docs/EVBeasyPIC_pl.pdf)
- (8) Dokumentacja układu MMUSB232 (pobrana w dniu 01.06.2020 r.), [http://www.propox.com/download/docs/MMusb232\\_en.pdf](http://www.propox.com/download/docs/MMusb232_en.pdf)
- (9) Dokumentacja biblioteki stdio.h dla języka C. [https://www.tutorialspoint.com/c\\_standard\\_library/stdio\\_h.htm](https://www.tutorialspoint.com/c_standard_library/stdio_h.htm)
- (10) IC16Fxx instruction set. <https://microchipdeveloper.com/mcu1102:instruction-set>

# WYKAZ RYSUNKÓW

## Wykaz rysunków

Rys 2.1: Schemat podłączenia obwodu pomiarowego do peryferiów mikrokontrolera - metoda z ADC i AC [2].....	7
Rys 2.2: Przebieg napięcia wyjściowego i wejściowego na wyprowadzeniach mikrokontrolera - metoda z ADC i AC [2].....	8
Rys 2.3: Schemat podłączenia obwodu pomiarowego do peryferiów mikrokontrolera - metoda z komparatorem i licznikiem [1].....	9
Rys 2.4: Przebieg napięcia obserwowany na czujniku indukcyjnym w trakcie pomiaru [1].....	10
Rys 2.5: <i>Przebieg napięcia na czujniku pojemnościowym w trakcie pomiaru [1].....</i>	11
Rys 2.6: Obwód pomiarowy - metoda z pomiarem czasu rozładowania [3].....	13
Rys 2.7: Przebieg napięcia w węźle 1 w trakcie ładowania i rozładowywania pojemności [3]..	14
Rys 3.1: Schemat blokowy mikrokontrolera PIC16F1789 [5].....	16
Rys 3.2: Schemat blokowy licznika "timer1" [5].....	17
Rys 3.3: Bity konfiguracyjne w rejestrze T1CON [5].....	18
Rys 3.4: Bity konfiguracyjne w rejestrze T1GCON [5].....	19
Rys 3.5: Przebiegi czasowe licznika przy uruchomionym "toggle mode" [5].....	19
Rys 3.6: Bity konfiguracyjne w rejestrze OSCCON [5].....	20
Rys 3.7: Blok EUSART [5].....	21
Rys 3.8: Bity konfiguracyjne w rejestrze TXSTA [5].....	21
Rys 3.9: Bity konfiguracyjne w rejestrze RCSTA [5].....	22
Rys 3.10: Schemat blokowy komparatora analogowego [5].....	23
Rys 3.11: Bity konfiguracyjne w rejestrze CMxCON0 [5].....	23
Rys 3.12: Bity konfiguracyjne w rejestrze CxCON1 [5].....	24
Rys 3.13: Schemat blokowy 8-bitowego przetwornika cyfrowo-analogowego [5].....	25
Rys 3.14: Bity konfiguracyjne w rejestrze DAC1CON0 [5].....	25
Rys 3.15: Schemat blokowy logiki systemu przerwań [5].....	26
Rys 4.1: Schemat blokowy stanowiska laboratoryjnego.....	27
Rys 4.2: Schemat blokowy układu pomiarowego.....	28
Rys 4.3: Schemat ideowy podłączenia mikrokontrolera w układzie pomiarowym.....	29
Rys 4.4: Schemat ideowy układu zasilania i włącznika.....	29
Rys 4.5: Sposób podłączenia gniazda RJ-12 do mikrokontrolera [5].....	30
Rys 4.6: Schemat ideowy podłączenia gniazda RJ-12.....	30
Rys 4.7: Schemat ideowy podłączenia układu MMUSB232.....	31
Rys 4.8: Schemat ideowy podłączenia przycisków.....	32
Rys 4.9: Schemat ideowy podłączenia gniazd kołkowych.....	33
Rys 5.1: Uproszczona struktura oprogramowania.....	34

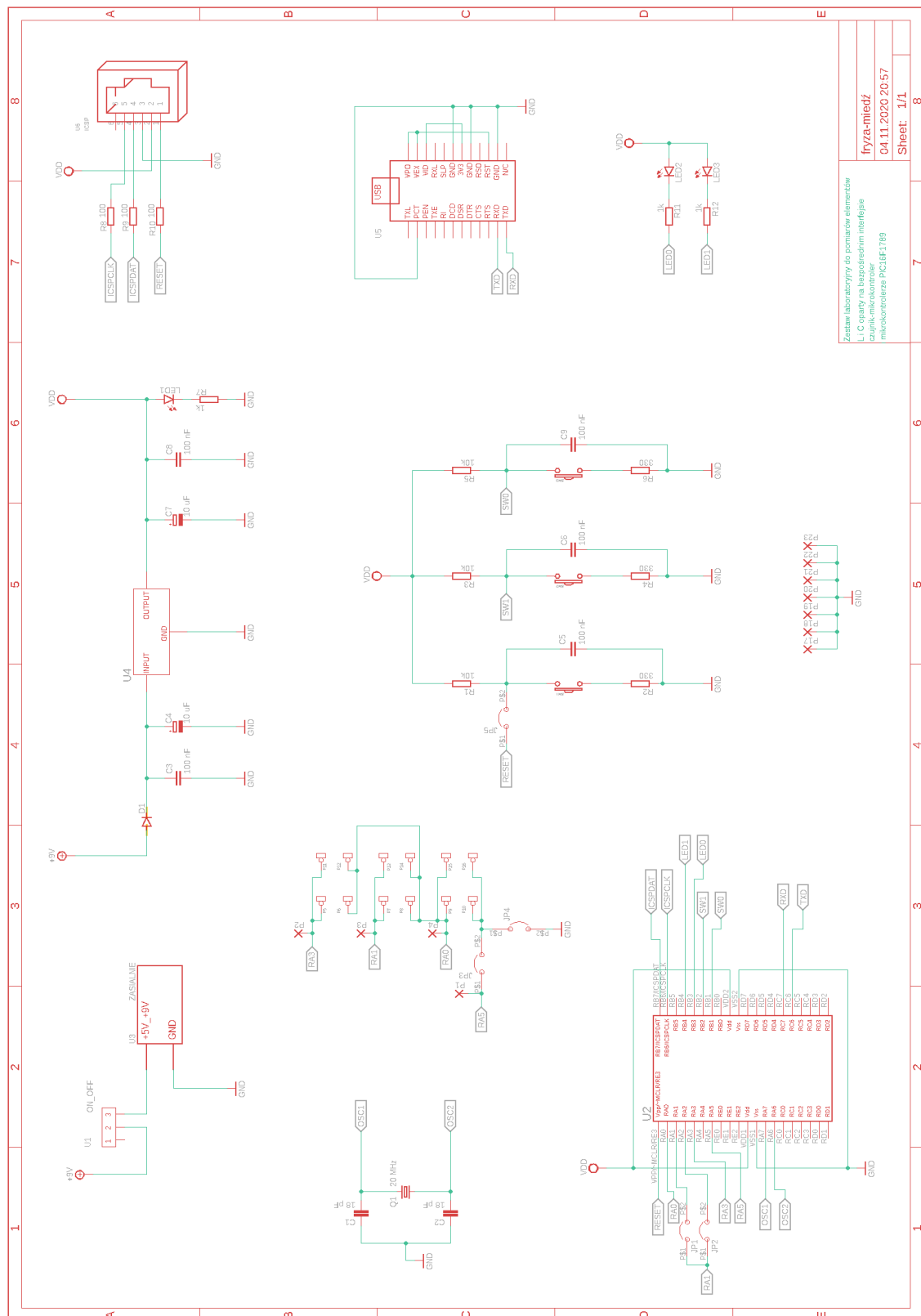
Rys 5.2: Diagram algorytmu funkcji pomiarowej.....	39
Rys 5.3: Stan terminala po resecie układu.....	44
Rys 5.4: Stan terminala po pojedynczym pomiarze.....	45
Rys 6.1: Schemat podłączenia oscyloskopu do układu dla pomiaru częstotliwości zegara.....	47
Rys 6.2: Podłączenie multimetru do testu DAC.....	48
Rys 6.3. Wartości zmierzonego napięcia wyjściowego w funkcji wartości teoretycznych wyliczonych na podstawie wzoru.....	49
Rys 6.4. Błąd bezwzględny w funkcji czasu.....	50
Rys 6.5. Przedstawienie rozstrzału błędu bezwzględnego dla pojemności z zakresu od 100 pF do 300 pF.....	50
Rys 6.6: Wartość zmierzona C w funkcji wartości dokładnej pojemności.....	51
Rys 6.7: Błąd względny dla wartości zmierzonej C w funkcji wartości dokładnej.....	51
Rys 6.8: Wartość pojemności po korekcji w funkcji wartości dokładnej.....	52
Rys 6.9: Błąd względny pojemności po korekcji w funkcji wartości dokładnej.....	52

## WYKAZ TABEL

### Indeks tabel

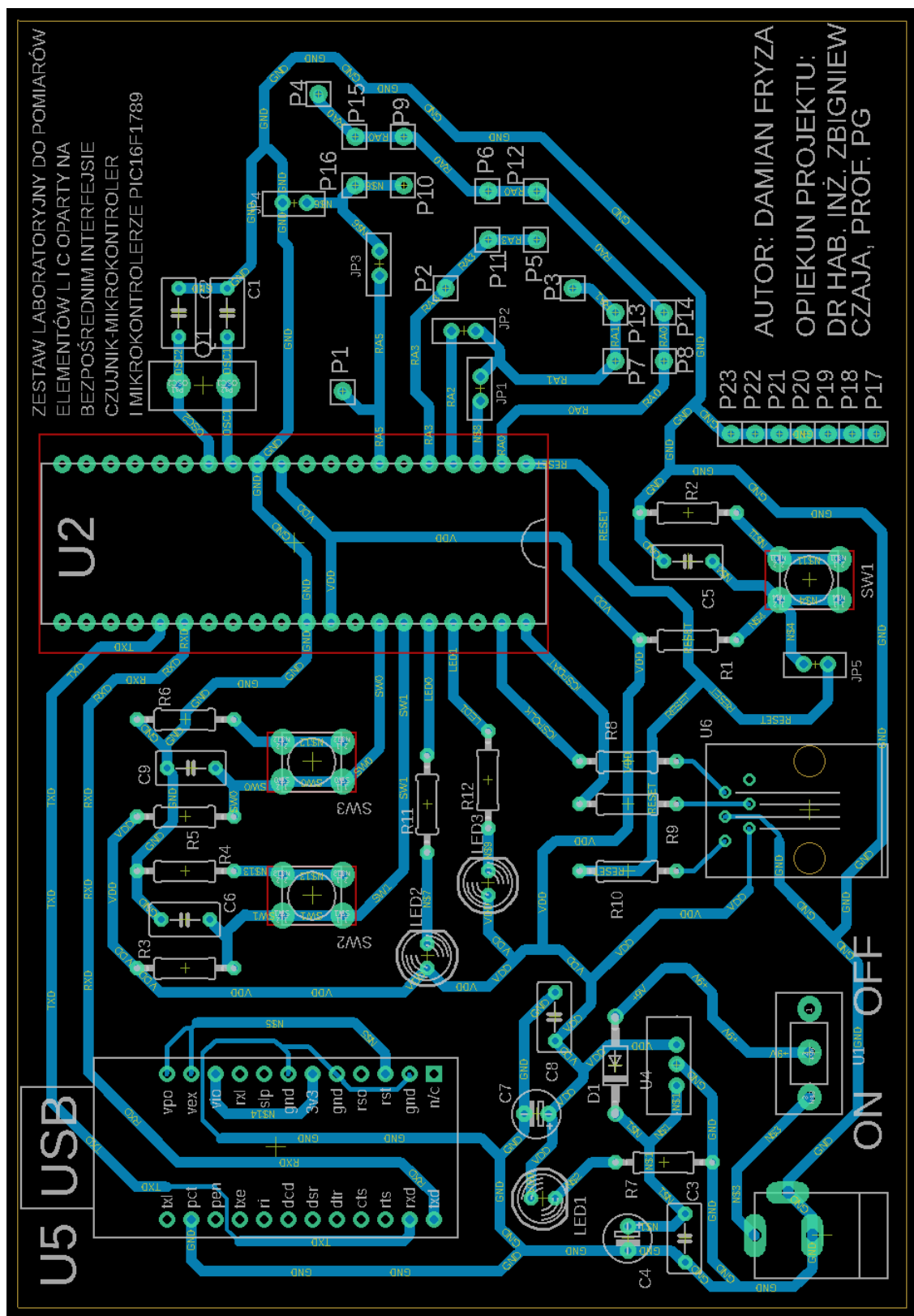
Tabela 6.1.Pomiar błędu wewnętrznego oscylatora kwarcowego.....	47
Tabela 6.2.Pomiar błędu napięcia wyjściowego przetwornika DAC.....	48
Tabela 6.3.Uśrednione wyniki pomiarów.....	51

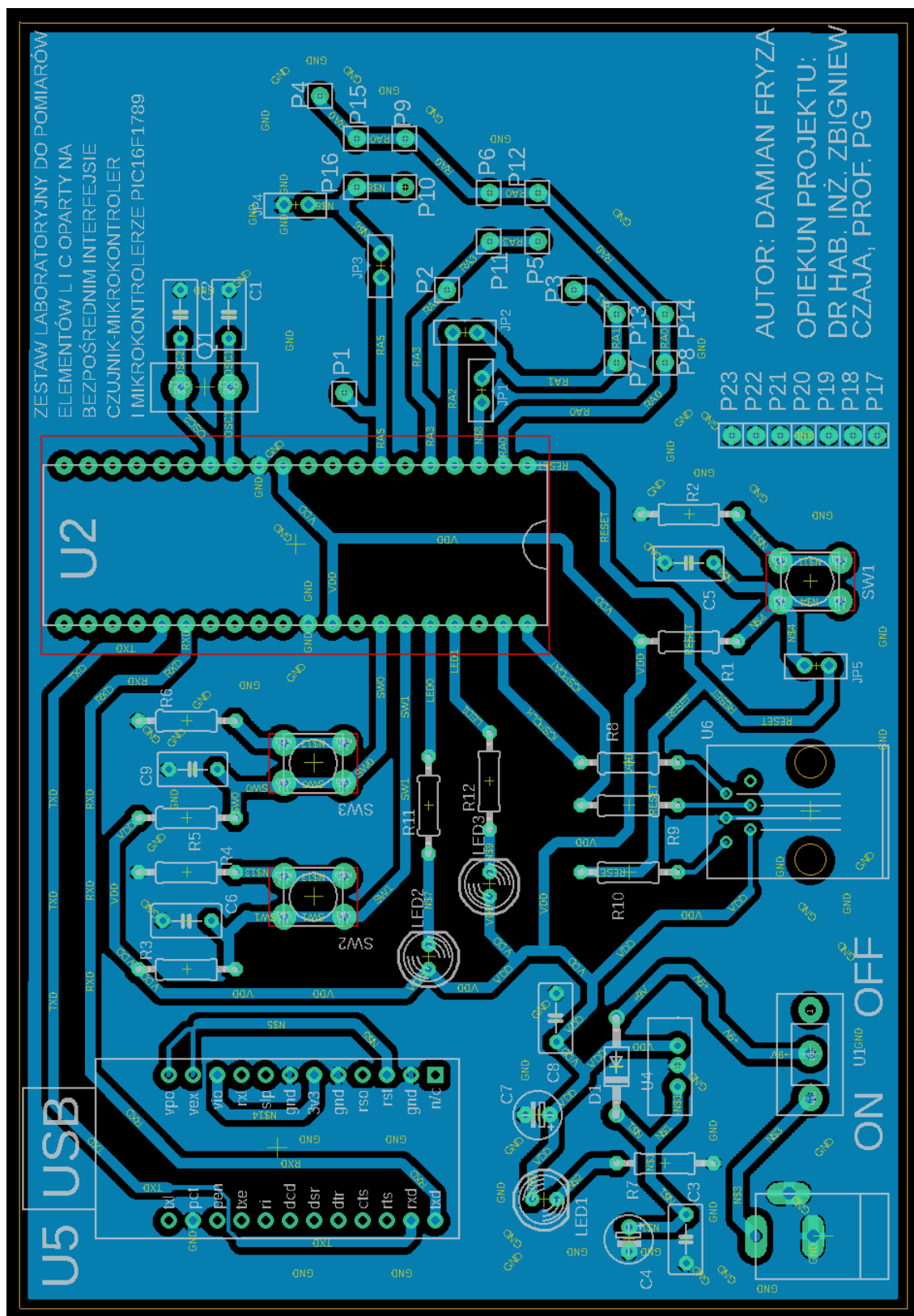
# DODATEK A: SCHEMAT IDEOWY UKŁADU POMIAROWEGO



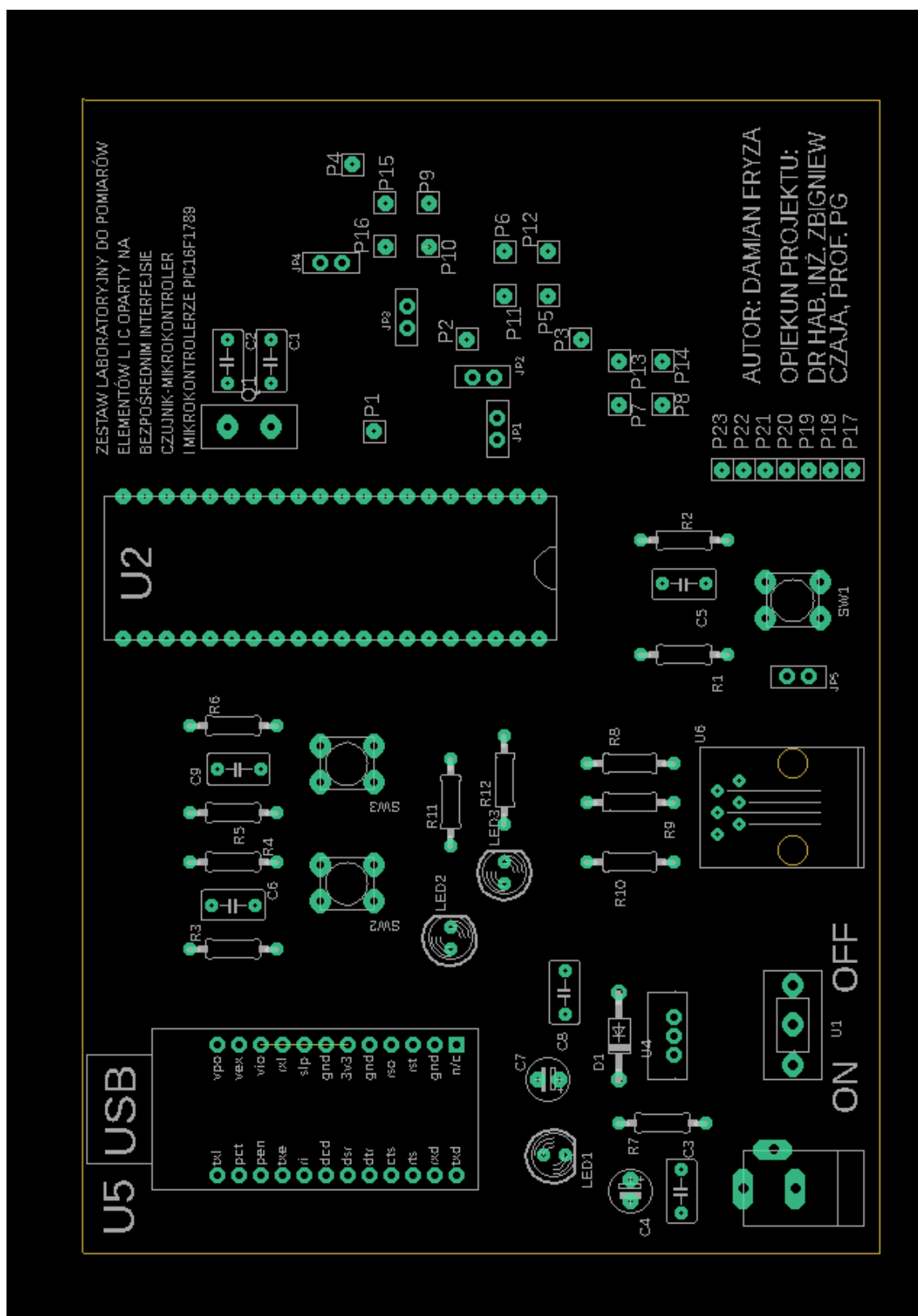
Rys A.1: Schemat ideowy

## DODATEK B: SCHEMAT OBWODU DRUKOWANEGO





Rys B.2: Schemat obwodu drukowanego z wypełnieniem miedzią.



Rys B.3: Schemat rozmieszczenia elementów na płytce PCB.