

# Wielokanałowy woltomierz – rejestrator

Prezentowany woltomierz wykonuje pomiary napięć występujących na 14 wejściach. Wyniki pomiarów są zapamiętywane w pliku tekstowym na karcie SD. Układ jest podzielony na dwie części: cyfrową i analogową. Część cyfrowa jest zbudowana w oparciu o 32-bitowy mikrokontroler Microchipa z rodziny PIC32MX i oprócz mikrokontrolera zawiera: interfejs karty SD, pamięć EEPROM i interfejs użytkownika składający się z wyświetlacza graficznego, impulsatora i dwóch przycisków. Część analogowa to głównie dzielniki napięcia, wtórniki napięciowe i układy zabezpieczające przez przepięciami na wejściu przetwornika.

**Rekomendacje:** przyrząd doskonale nadaje się do roli centrum rozbudowanego systemu pomiarowego z możliwością dokumentowania wyników pomiarów.

Pomiar napięcia ma kluczowe znaczenie w elektronice i elektrotechnice. Pomiary innych wielkości często są wykonywane metodą pośrednią poprzez pomiar napięcia, na przykład w cęgowych miernikach prądu stałego lub przemiennego z wyjściem napięciowym wyskalowanym jako 1 mV/A, 10 mV/A itp. Również pomiar natężenia prądu metodą tech-

niczną jest pomiarem spadku napięcia na rezystorze pomiarowym. W wielu czujnikach wielkości nieelektrycznych sygnał napięciowy jest wyjściowym sygnałem proporcjonalnym do mierzonej wielkości. Przykładem może być czujnik ciśnienia atmosferycznego.

Szczególna rola pomiaru napięcia została dostrzeżona wiele lat temu przez produ-

## W ofercie AVT\*

AVT-5458 A

Podstawowe informacje:

Dodatkowe materiały na FTP:

<ftp://ep.com.pl>, user: 76305, pass: 8741rnfv

• wzory płytek PCB

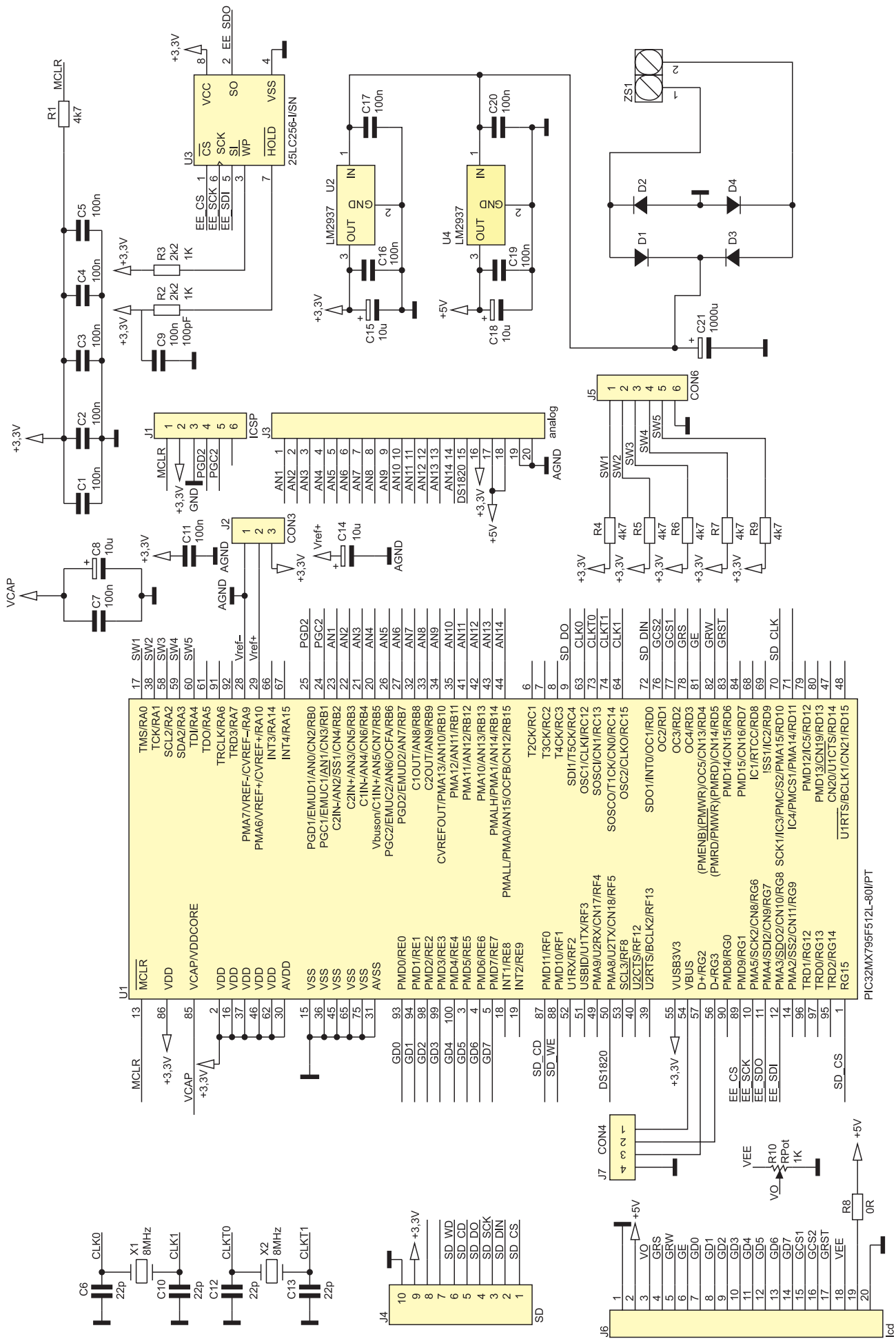
Projekty pokrewne na FTP:

(wymienione artykuły są w całości dostępne na FTP)

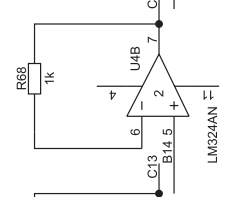
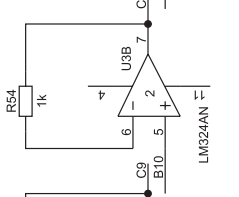
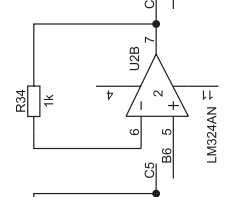
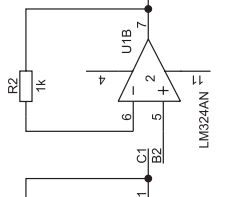
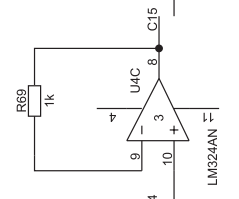
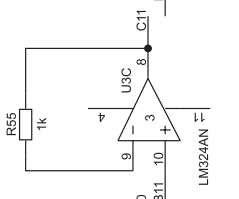
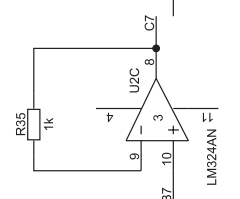
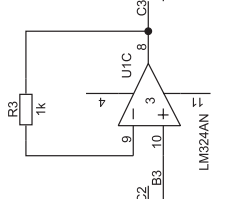
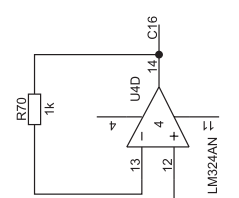
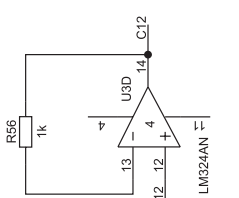
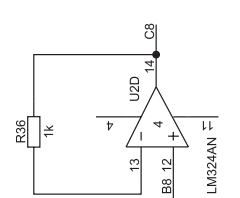
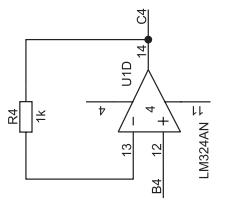
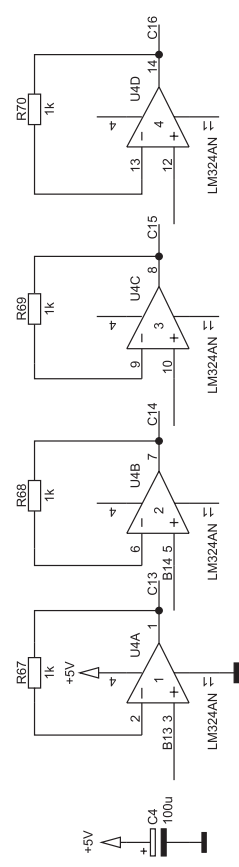
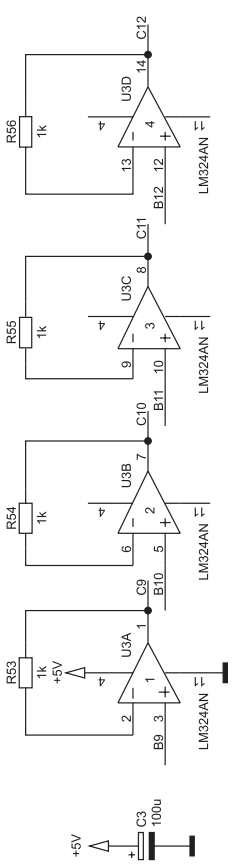
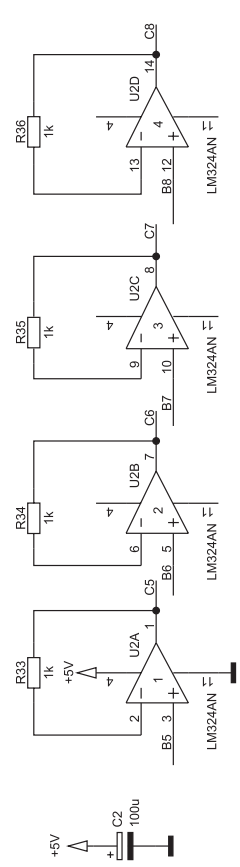
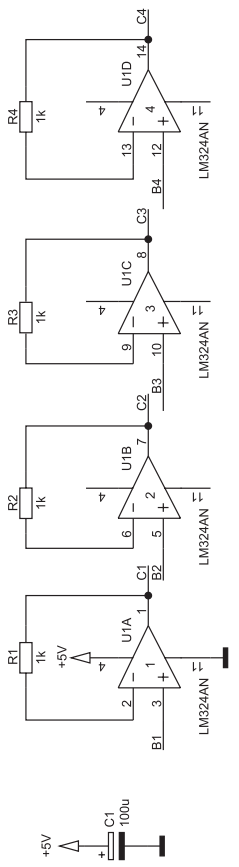
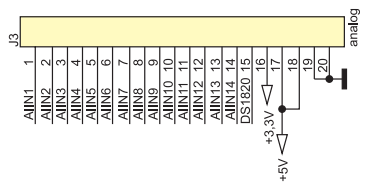
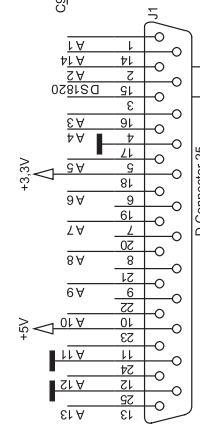
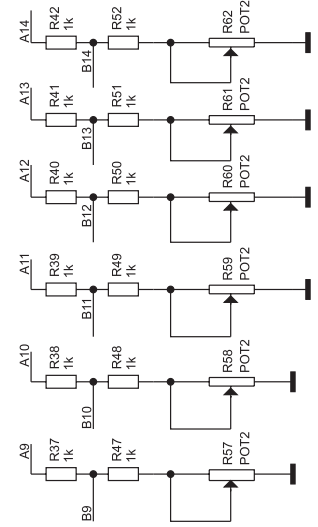
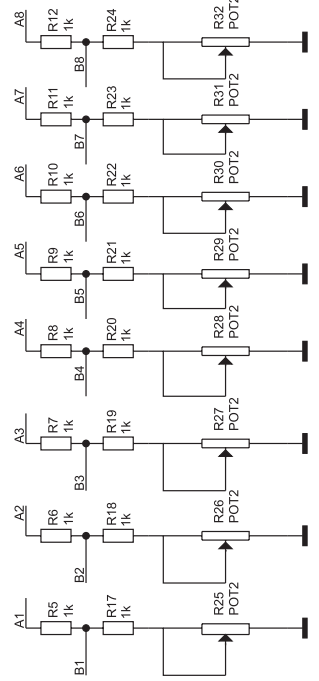
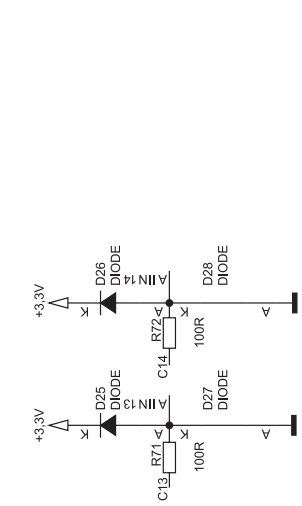
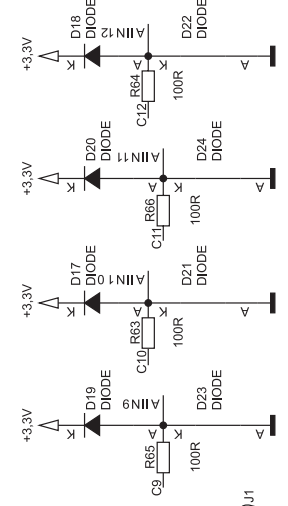
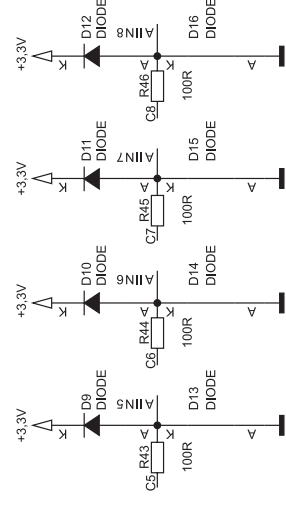
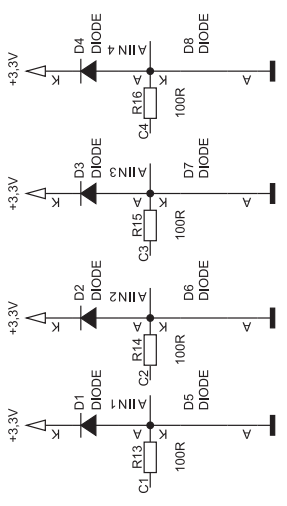
AVT-xxxx

\* Uwaga:  
Zestawy AVT mogą występować w następujących wersjach:  
AVT xxxx UK to zaprogramowany układ. Tylko i wyłącznie. Bez elementów dodatkowych.  
AVT xxxx A płytka drukowana PCB (lub płytki drukowane, jeśli w opisie wyraźnie zaznaczono), bez elementów dodatkowych.  
AVT xxxx A+ płytka drukowana i zaprogramowany układ (czyli połączenie wersji A i wersji UK) bez elementów dodatkowych.  
AVT xxxx B płytka drukowana (lub płytki) oraz komplet elementów wymienionych w załączniku pdf  
AVT xxxx C to nic innego jak zmontowany zestaw B, czyli elementy wlutowane w PCB. Należy mieć na uwadze, że o ile nie zaznaczono wyraźnie w opisie, zestaw ten nie ma obudowy ani elementów dodatkowych, które nie zostały wymienione w załączniku pdf  
AVT xxxx CD oprogramowanie (nieczęsto spotykana wersja, lecz jeśli występuje, to niezbędne oprogramowanie można ściągnąć, klikając w link umieszczony w opisie kitu)

Nie każdy zestaw AVT występuje we wszystkich wersjach! Każda wersja ma załączony ten sam plik pdf! Podczas składania zamówienia upewnij się, którą wersję zamawiasz! (UK, A, A+, B lub C). <http://sklep.avt.pl>



Rysunek 1. Schemat woltomierza: a) części cyfrowej – sterownika woltomierza, b) części analogowej – obwodów wejściowych



centów mikrokontrolerów. Dzisiaj trudno znaleźć mikrokontrolery (szczególnie te trochę bardziej rozbudowane) bez wbudowanego przetwornika A/C z wejściem napięciowym. Te przetworniki mają coraz lepsze parametry. Nie myślę tu tylko o zwiększaniu rozdzielczości, ale również o poprawianiu innych, równie ważnych parametrów, takich jak na przykład liniowość przetwarzania czy szybkość działania. Wbudowanie możliwości pomiaru napięcia w mikrokontrolery daje

spore możliwości. Za pomocą jednego przetwornika można mierzyć napięcie w wielu kanałach pomiarowych. Wtedy pomiar odbywa się sekwencyjnie, poprzez przełączanie wejścia przetwornika za pomocą wbudowanego multiplexera analogowego. Wyniki otrzymane na wyjściu przetwornika A/C prawie zawsze wymagają przeliczeń i w takiej sytuacji przetwornik zintegrowany w mikrokontrolerem jest bardzo wygodnym rozwiązaniem. Jeżeli potrzebujemy wykonać szybko wiele pomiarów, to w mikrokontrolerach 32-bitowych można znaleźć połączenie przetwornika z kanałem DMA. Program główny nie musi zajmować się wyzwaniem i sprawdzaniem warunku zakończenia konwersji oraz zapisywaniem wyników do bufora w pamięci RAM – automatycznie wykona to mechanizm DMA.

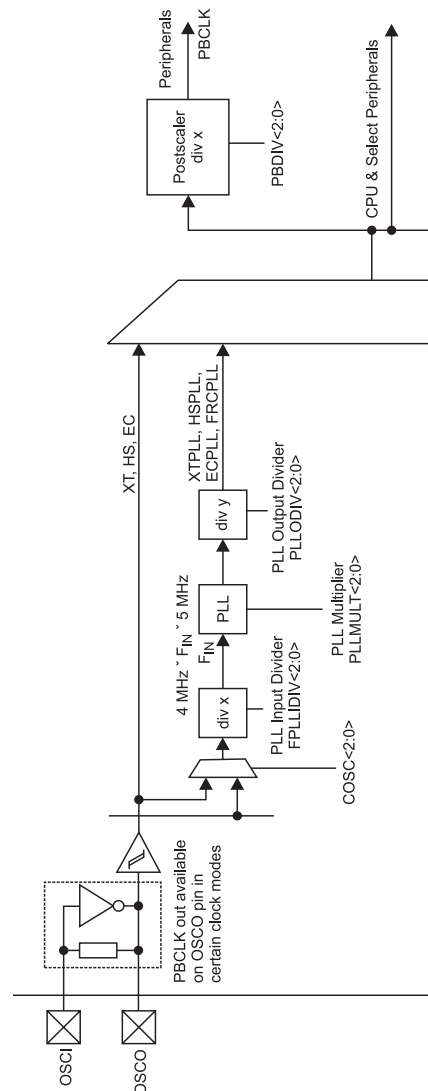
**Założenia**

Prezentowany w artykule woltomierz ma za zadanie wykonać pomiary napięcia w 14 kanałach pomiarowych (ma 14 wejść napięciowych). Wyniki pomiarów są zapisywane na karcie SD w pliku tekstowym. Układ został podzielony na dwie części: cyfrową i analogową. Część cyfrowa jest zbudowana w oparciu o 32-bitowy mikrokontroler firmy Microchip z rodziny PIC32MX i oprócz mikrokontrolera zawiera: interfejs karty SD, pamięć EEPROM i interfejs użytkownika składający się z wyświetlacza graficznego, impulsatora i dwóch przycisków. Część analogowa to głównie dzielniki napięcia, wtórniki napięciowe i układy zabezpieczające przez przepięciami na wejściu przetwornika.

**Część cyfrowa – sterownik woltomierza**

Na **rysunku 1** pokazano schemat części cyfrowej – sterownika woltomierza. Głównym elementem jest mikrokontroler PIC32MX795F512L. Jest to szybki, bardzo dobrze wyposażony mikrokontroler, z dużą pamięcią programu i pamięcią RAM. Ma wbudowany szybki, 10-bitowy przetwornik z 16 wejściami analogowymi. Oprócz przetwornika do komunikacji z kartą SD w trybie SPI będzie wykorzystany sprzętowy interfejs SPI, wbudowany moduł sprzętowego zegara RTC i linie portów do komunikacji z wyświetlaczem LCD, do obsługi przycisków i impulsatora zastosowanych w interfejsie użytkownika.

Źródłem zegara taktującego mikrokontroler jest oscylator kwarcowy o częstotliwości 8 MHz stabilizowany kwarem X1. Sygnał z oscylatora jest podawany na wejście układu PLL, w którym najpierw jego częstotliwość jest dzielona przez 2 w module divx, a potem powielana przez 20 w module PLL (**rysunek 2**). Układ taktowania jest programowany za pomocą bitów konfiguracyjnych mikrokontrolera. Na **listingu 1** pokazano de-



**Rysunek 2. Schemat blokowy układu taktowania mikrokontrolera**

finicje wszystkich bitów konfiguracyjnych mikrokontrolera.

Drugi oscylator (*secondary oscillator*) jest używany do taktowania sprzętowego układu zegara czasu rzeczywistego RTTC. Oscylator „zegarkowy” o częstotliwości 32,768 kHz jest dołączony do wyprowadzeń CLK0 i CLK1. Zegar czasu rzeczywistego jest niezbędny do pracy systemu plików FAT16/32 używanego przy zapisie pliku tekstowego z wynikami pomiarów.

Moduł RTTC (**rysunek 3**) jest dość rozbudowanym modulem zawierającym liczniki czasu (godziny, minuty, sekundy) liczące w formacie 24 godzinnym, liczniki kalendarza (dzień tygodnia, dzień miesiąca, miesiąc i rok), konfigurowane alarmy, oraz programowana kalibrację dokładności odmierzenia czasu.

Zasadniczym blokiem funkcjonalnym używanym w woltomierzu jest wbudowany w strukturę przetwornik A/C o rozdzielczości 10 bitów, wykonujący konwersje SAR (*Successive Approximation Register*). Na wyjściu przetwornika umieszczono 16-wejściowy multiplexer analogowy. Napięciem

**Wykaz elementów Płytki sterownika**

- Rezystory:** (SMD 1206)  
R1, R4...R7, R9: 4,7 kΩ  
R2, R3: 2,2 kΩ  
R8: 0 Ω (zwora)  
R10: 22 kΩ (potencjometr)
- Kondensatory:** (SMD 0805)  
C1...C5, C7: 100 nF  
C6, C10, C12, C13: 22 pF  
C8, C14, C15, C18: 10 μF/16V (elektrolit.)  
C9, C11, C16, C17, C19, C20: 100 nF (SMD 1206)  
C21: 1000 μF/25 V (elektrolit.)
- Półprzewodniki:**  
U1: PIC32MX795F512L-80I/PT (zaprogramowany)  
U2: LM2937-3,3  
U3: 25LC256-I/SN  
U4: LM2937-5
- Inne:**  
X1: 8 MHz (rezonator kwarcowy)  
X2: 32,768 kHz (rezonator kwarcowy)  
Listwy podwójnych goldpinów, raster 2,54 mm

**Płytką wejść analogowych**

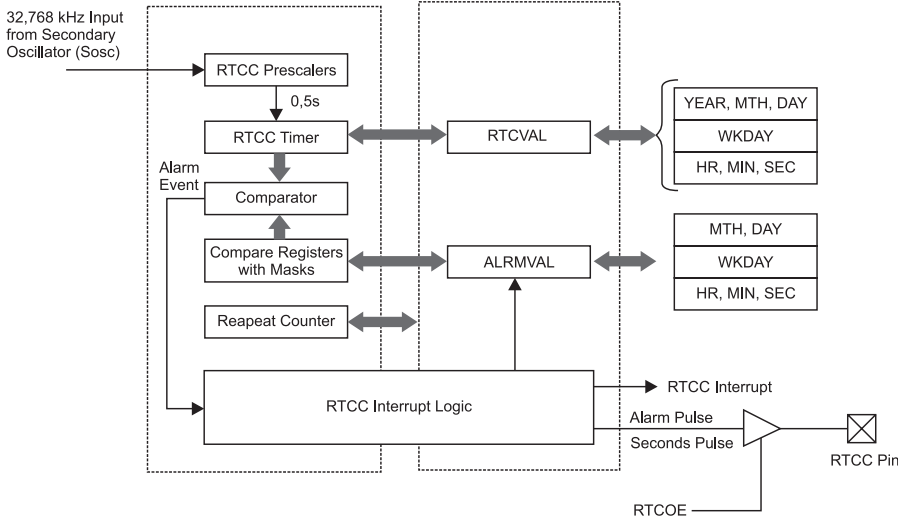
- Rezystory:** (metalizowane, 0,6 W)  
R5...R12, R17...R24, R37...R42, R47...R52: 1 kΩ  
Rezystory w dzielnikach dobierane zależnie od wymaganego stopnia podziału  
R3, R4, R33...R36, R53...R56, R67...R71: 10 kΩ  
R13...R16, R43...R46, R63...R66, R71, R72: 1 kΩ  
R25...R32, R57...R62: precyzyjne potencjometry wieloobrotowe
- Kondensatory:**  
C1...C4: 100 μF/16 V (elektrolit.)
- Półprzewodniki:**  
U1...U4: LM324AN (DIP14)
- Inne:**  
Listwy podwójnych goldpinów raster 2,54 mm  
Płytką drukowaną.  
J1: Złącze DSUB25 do montażu w płytce drukowanej (przewlekane)

**Płytką karty SD**

- Rezystory:** (SMD 1206)  
R1, R3...R8: 10 kΩ  
R2: 180 Ω  
Złącze karty SD do montażu powierzchniowego

**Pozostałe elementy**

- Wyświetlacz graficzny monochromatyczny 128×64 piksele z podwójnym sterownikiem KS0108 HY12864K  
Impulsator ze stykiem zwiernym 2 przyciski zwiernie (START, STOP)

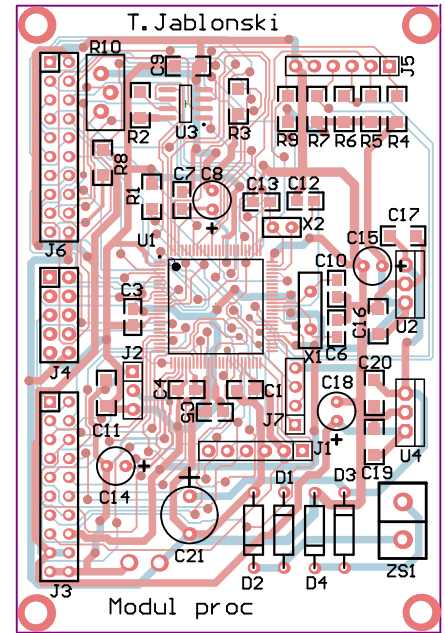


Rysunek 3 Schemat blokowy modułu RTCC

referencyjne może być napięcie zasilające mikrokontroler lub napięcie zewnętrzne, podawane na odpowiednie wejścia analogowe. Kanały pomiarowe są skanowane w trybie automatycznym lub przez programowe wskazanie wejścia pomiarowego. Wyzwalanie pomiaru może odbywać się z kilku wybieranych źródeł. Schemat blokowy przetwornika pokazano na **rysunku 4**. Sygnał analogowy jest podawany poprzez multiplexer na wejście układu próbkująco – pamiętającego SHA

(Sample and Hold Amplifier), a następnie konwertowany. Wynik konwersji można zapisać w rejestrze wyjściowym w jednym z ośmiu formatów. Moduł przetwornika jest wyposażony bufor mogący pomieścić 16 wyników pomiarów. Do detekcji zapełnienia bufora do połowy jest wykorzystywany bit statusowy BUFS.

W nowych typach mikrokontrolerów Microchip zrezygnował z wbudowywania w strukturę układu obszaru pamięci



Rysunek 5. Schemat montażowy płytki sterownika

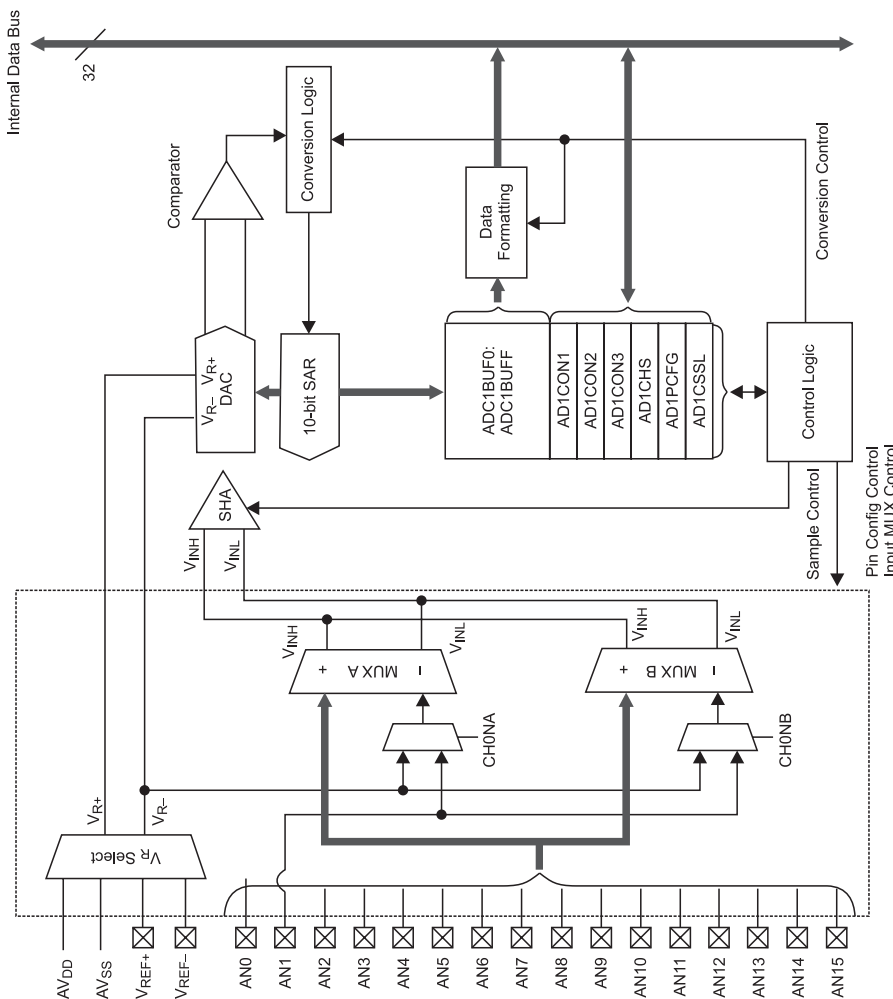
EEPROM. Nasz woltomierz potrzebuje pamięci nieulotnej do przechowywania nastaw niezbędnych w czasie pracy. Dlatego zastosowałem zewnętrzną pamięć EEPROM typu 25LC256 (U3) z interfejsem SPI.

Cały układ jest zasilany dwoma napięciami: +3,3 V i +5 V. Napięcie +5 V jest używane do zasilania wyświetlacza LCD i wzmacniaczy operacyjnych w module analogowym. Napięcie +3,3 V zasilają wszystkie pozostałe układy sterownika cyfrowego: mikrokontroler, kartę SD i pamięć EEPROM.

### Część analogowa

Schemat modułu analogowego pokazano na **rysunku 5**. Ma on do wykonania trzy zasadnicze zadania:

- Dopasowanie zakresu napięcia mierzzonego do zakresu dopuszczalnego dla przetwornika A/C; standardowo jest to realizowane za pomocą rezystancyjnych dzielników wejściowych.
- Zapewnienie niskiej impedancji źródła napięcia podawanego na wejście przetwornika.



Rysunek 4. Schemat blokowy przetwornika A/C

REKLAMA

Projekty na...Texas

**STM32**

Przebieg na mapie: Franklin St, Congress St, Prairie St

Arrival 1:32

[www.stm32.eu](http://www.stm32.eu)

ST logo and KAMAMI logo with 'life.augmented' text.

- Zabezpieczenie wejścia przetwornika przed zbyt wysokim napięciem lub napięciem o polaryzacji odwrotnej.

Dzielniki napięciowe są zbudowane z rezystorów metalizowanych o tolerancji 1% i wieloobrotowych potencjometrów. Pozwala to na uniknięcie stosowania trudnodostępnych i drogich rezystorów precyzyjnych, ale za to jest konieczne kalibrowanie dzielnika według wskazań wzorcowych. Oczywiście, nic nie stoi na przeszkodzie by dobrać i zastosować dzielnik tylko z rezystorów (bez potencjometrów). Mierzone napięcie wejściowe podawane na wejścia AN1x po podzieleniu trafia na wejście nieodwracające wzmacniacza operacyjnego pracującego w konfiguracji wtórnika napięciowego. Wtórnik ma wzmocnienie 1, bardzo dużą impedancję wejściową i małą impedancję wyjściową i jest zbudowany ze wzmacniaczy operacyjnych popularnego poczwórnego układu LM324 zasilanego napięciem niesymetrycznym +5 V. Zastosowanie wtórników dodatkowo chroni wejścia przetwornika wbudowanego w mikrokontroler.

Mała impedancja wyjściowa tego układu powoduje, że jest spełniony wymóg minimalnej rezystancji źródła napięcia podawanego na wejście przetwornika. Jeżeli ta rezystancja jest zbyt duża, to nie zdąży naładować się kondensator próbkujący na wejściu przetwornika, a co za tym idzie, pomiar będzie błędny. Z kolei, duża impedancja wejściowa nie powoduje znaczącego obciążania dzielnika wejściowego. Dlatego można stosować dzielniki o dużych rezystancjach nieobciążających mierzonych obwodów.

Układ zabezpieczenia przed przepięciami w zasadzie powinien być umieszczony na wejściu bufora napięciowego, aby chronił również jego układy. Ja zdecydowałem się umieścić taki układ na wejściu przetwornika (wyjściu bufora). Jest to pewien kompromis podyktowany po pierwsze tym, że LM324 jest zasilany napięciem +5 V, a mikrokontroler napięciem +3,3 V. Uszkodzenie wzmacniacza operacyjnego może spowodować podanie na wejście przetwornika zbyt wysokiego napięcia i w konsekwencji uszkodzenie mikrokontrolera. Po drugie, przepięcia pojawiające się na wejściu wzmacniacza operacyjnego mogą oczywiście go uszkodzić, ale LM324 jest układem bardzo tanim a po umieszczeniu go w podstawce – łatwym do wymiany. Optymalnym wyjściem byłoby zastosowanie układów zabezpieczających na wejściu wtórnika i na wejściu przetwornika, ale z uwagi na dużą liczbę wejść pomiarowych takie rozwiązanie spowodowałoby jeszcze większą rozbudowę układu.

Zgodnie notą katalogową mikrokontrolera napięcie na dowolnej linii portu musi mieścić się w zakresie  $VDD+0,3 V \dots VSS-0,3 V$ .

**Listing 1. Definicje bitów konfiguracyjnych mikrokontrolera**

```
#if defined (__PIC32MX__)
#pragma config FPLLMUL = MUL_20 // PLL Multiplier
#pragma config FPLLIDIV = DIV_2 // PLL Input Divider
#pragma config FPLLODIV = DIV_1 // PLL Output Divider
#pragma config FPBDIV = DIV_2 // Peripheral Clock divisor
#pragma config FWDTEN = OFF // Watchdog Timer
#pragma config WDTPS = PS1 // Watchdog Timer Postscale
#pragma config FCKSM = CSDCMD // Clock Switching & Fail Safe Clock Monitor
#pragma config OSCIOFNC = OFF // CLKO Enable
#pragma config POSCMOD = HS // Primary Oscillator
#pragma config IESO = OFF // Internal/External Switch-over
#pragma config FSOSCEN = OFF // Secondary Oscillator Enable (KLO was off)
#pragma config FNOSC = PRIPLL // Oscillator Selection
#pragma config CP = OFF // Code Protect
#pragma config BWP = OFF // Boot Flash Write Protect
#pragma config PWP = OFF // Program Flash Write Protect
#pragma config ICESEL = ICS_PGx1 // ICE/ICD Comm Channel Select
#pragma config DEBUG = ON // Background Debugger Enable
#endif
```

**Listing 2. Inicjalizacja przetwornika A/C**

```
//inicjalizacja przetwornika
void ADCInit(void) {
    CloseADC10(); //zatrzymaj ADC
    AD1PCFG=3; //ANO,AN1 digital-reszta analog
    AD1CON1 = 0x00E0; //automatyczna konwersja po zakończeniu próbkowania
    AD1CSSL = 0; //bez skanowania wejść
    AD1CON2 = 0; //MUXA, AVss/AVdd used as Vref+/-
    AD1CON3 = 0x1F3F; //max sample time = 31Tad
    AD1CON1SET = 0x8000; //włącz ADC
}
```

**Listing 3. Odczytanie kanału pomiarowego**

```
int ReadADC( int channel)
{
    int adcRead;
    AD1CHSbits.CH0SA = channel+2; // 1. Wybór kanału pomiarowego
    AD1CON1bits.SAMP = 1; // 2. start próbkowania
    while (!AD1CON1bits.DONE); // 3. Czekaj na zakończenie konwersji
    adcRead = ADC1BUF0; // odczytaj przekierwertowana wartość
    IFS1CLR = 0x0002; // 5. Zeruj flagę przerwania
    AD1CON1bits.DONE = FALSE;
    return adcRead; // 6. Zwracanie odczytanej wartości
}
```

Zabezpieczenie przed przekroczeniem tej wartości polega na dołączeniu do każdego wejścia analogowego miernika dwóch diod Shottky'ego. Dla toru pierwszego będą to diody D1 i D5. Kiedy napięcie na w punkcie AIN1 połączonym z wejściem analogowym przetwornika jest niższe niż VDD, które wynosi +3,3 V (z powodu podania zbyt dużego napięcia na wejście pomiarowe ANO1), to dioda D1 jest spolaryzowana zaporowo. Po przekroczeniu napięcia +3,3 V w punkcie AIN1 dioda D1 zostaje spolaryzowana w kierunku przewodzenia i nie pozwala na wymuszenie napięcia wyższego niż  $VDD+0,3 V$ . Prąd płynący przez D1 spolaryzowaną w kierunku przewodzenia powoduje spadek napięcia na rezystorze R13. Podobnie wygląda sytuacja w wypadku podania napięcia ujemnego na wejście ANO1. Dioda D5 zostaje spolaryzowana w kierunku przewodzenia nie dopuszczając do spadku napięcia w punkcie AIN1 poniżej potencjału  $VSS-0,3 V$ . Taki układ zabezpieczenia przed przepięciami jest skutecznym i szeroko stosowany.

### Pomiar napięcia

Pomiar napięcia polega na:

- Podzieleniu napięcia wejściowego za pomocą dzielnika wejściowego.
- Wyborze wejścia pomiarowego.
- Wykonanie konwersji napięcia z wejścia pomiarowego przez przetwornik.
- Przeliczeniu wartości z przetwornika na mierzone napięcie.

Wybór dzielnika wejściowego zależy od kilku czynników. Po pierwsze, musimy znać zakres pomiarowy napięcia wejściowego i zakres napięć wejściowych przetwornika. Tę drugą wartość określa napięcie referencyjne. Napięciem referencyjnym w przypadku naszego mikrokontrolera może być napięcie VDD lub napięcie zewnętrzne podawane na specjalnie wydzielone wyprowadzenia VRef- i Vref+. Użycie napięcia VDD upraszcza układ, bo nie trzeba dodatkowych źródeł napięcia referencyjnego. Można go stosować w wypadku, gdy nie jest wymagana duża dokładność pomiarów i można się pogodzić z szumami na zasilaniu VDD. Kiedy jest oczekiwana większa dokładność, to można dołączyć napięcie referencyjne ze specjalizowanego układu scalonego. Do tego celu zostało przygotowane złącze CON3.

Po wybraniu napięcia referencyjnego i zakresu pomiarowego można określić jaki ma być stopień podziału dzielnika wejściowego. Na przykład, jeżeli zakres pomiarowy to 12 V, a napięcie referencyjne 3,3 V, to napięcie wejściowe musi być podzielone przez  $12/3,3=3,63634$ . W handlu można znaleźć precyzyjne dzielniki rezystancyjne dzielące przez potęgę liczby dziesiętej: 1/10, 1/100, 1/1000, ale trudno znaleźć tak nietypowy podział jak 3,63634. Rozwiązaniem mogłoby być użycie napięcia referencyjnego 2,00 V i zakresu pomiarowego 20 V. Jednak, gdy potrzebujemy zakresu 12 V, to należy dążyć by większość mierzonych napięć była bliska

**Listing 4. Pomiar napięcia z konwersją dla zakresu 12 V**

```
float GetAdcV(char channel){
    unsigned int adc,i;
    float result;
    adc=0;
    adc=ReadADC(channel);
    result=(float)adc/85.33;
    return(result);
}
```

zakresowi pomiarowemu. Zakres 20 V wprowadzałby tu dodatkową niepewność pomiaru. Dodatkowo, ze względu na to, że wartość wyjściowa z przetwornika musi być przeliczona stosując się napięcia referencyjne o wartościach będących potęgami liczby 2, na przykład 2,048 V ( $2^{11}$ ). W takim przypadku zmiana wartości na najmłodszym bicie wyniku dla przetwornika 10-bitowego odpowiada 2,048 V/1024=2 mV. Dla zakresu 2 V wartość odczytaną mnożymy przez 0,002, dla zakresu 20 V przez 0,02, dla zakresu 200 V przez 0,2. Kiedy przyjmiemy napięcie referencyjne równe 3,3 V, to dla przetwornika 10-bitowego zmiana na najmłodszym bicie odpowiada wartości 3,3 V/1024=3,223 mV.

W pierwszym wypadku do obliczeń możemy zastosować proste obliczenia stałoprzecinkowe (ze wstępnym przesuwaniem przecinka), a w drugim stosować arytmetykę zmiennoprzecinkową ze konsekwencjami zaokrągleń.

Z powyższych rozważań można wywnioskować, z jakiego powodu zdecydowałem się na dzielnik z precyzyjnymi potencjometrami wielobrotowymi. Pozwala on na elastyczne dobranie wartości podziału i napięcia referencyjnego. Wady to możliwość zmiany podziału pod wpływem temperatury, wilgotności, lub starzenia się potencjometru. Żeby je zminimalizować należy dążyć, by rezystancja potencjometru miała jak najmniejszy udział w dzielniku. Aby było to możliwe, w szereg z każdym z potencjometrów włączono dodatkowy rezystor.

Wybór wejścia analogowego odbywa się za pomocą dwu multiplexerów MUXA i MUXB programowanych przez bity CHOSA[4:0] i CHOSB[4:0] rejestru A1CHS. Takie rozwiązanie pozwala na doprowadzenie do wejść odwracającego i nieodwracającego wzmacniacza SHA sygnałów napięciowych z dwóch różnych wejść analogowych i pomiar sygnału różnicowego. Po połączeniu przez multiplexer wejścia odwracającego z masą można mierzyć sygnały asymetryczne (SE). Ponieważ nasz miernik ma mierzyć napięcia względem masy, to trzeba wybrać tę ostatnią konfigurację.

Tak rozbudowany moduł peryferyjny, którym jest przetwornik A/C wymaga wstępnej inicjalizacji. Ja założyłem, że wejścia analogowe nie będą automatycznie skanowane, a o tym, na którym wejściu aktualnie będzie wykonywany pomiar będzie decydował program użytkownika. Po zakończeniu próbkowania moduł ma wykonać automatyczna

**Listing 5. Funkcja sprawdzająca warunek wyzwolenia pomiaru**

```
//sprawdzenie warunku czasu (momentu) wyzwolenia pomiaru
char MeasureTrig(unsigned char interwal)
{
    int period;
    int sec,min;
    GetTime();
    if((TimeInt.sec==Time.sec)&&(TimeInt.min==Time.min))
    {
        period=interwal*30;
        sec=period%60;
        min=period/60;
        TimeInt.sec=TimeInt.sec+sec;
        if(TimeInt.sec>59)
        {
            TimeInt.sec=TimeInt.sec-60;
            ++TimeInt.min;
        }
        TimeInt.min=TimeInt.min+min;
        return(1);//czas jest równy
    }
    Else return(0); //czas nie jest równy
}
```

konwersję na postać cyfrową. W czasie testowania miernika używałem jako napięcia referencyjnego napięcia zasilania mikrokontrolera o wartości +3,3 V. Na końcu inicjalizacji moduł przetwornika jest włączany. Procedurę inicjalizacji pokazano na **listingu 2**.

Kiedy inicjalizacja jest wykonana, można wykonać pomiary. Pomiar wygląda standardowo: najpierw jest wybierane wejście pomiarowe przez zapisanie bitów CHOSA[4:0]. W mierniku są używane wejścia od AN2 do AN16, więc do numeru kanału (numerowanego od zera) jest dodawana liczba 2. Potem jest inicjowane próbkowanie przez ustawienie bitu SAMP w rejestrze AD1CON1. Po zakończeniu konwersji sygnałizowanym przez ustawienie bitu DONE w rejestrze AD1CON1 można odczytać wartość po konwersji z rejestru ADC1BUF0. Procedurę odczytania wybranego kanału pomiarowego pokazano na **listingu 3**.

Ostatnim krokiem przy pomiarze napięcia jest przeliczenie wartości, aby reprezentowała napięcie w woltach. Załóżmy, że zakres pomiarowy wynosi +12 V, a napięcie referencyjne +3,3 V. Jeżeli ustawimy dzielnik w taki sposób, aby przy napięciu +12 V na wejściu pomiarowym na wejściu przetwornika było +3,3 V, to możemy przyjąć, że wartość  $2^{10}=1024$  odpowiada napięciu +12 V. Zatem zmiana wartości na najmniej znaczącym bicie odpowiada  $12\text{ V}/1024=0,01172\text{ V}$ . Żeby przeliczyć wartość odczytaną z przetwornika na napięcie skalowane w woltach trzeba ją pomnożyć przez 0,01172 lub podzielić przez 85,3333. Na przykład, dla połowy zakresu otrzymujemy na wyjściu przetwornika 512 odpowiadającą napięciu  $512 \times 0,01172=6,00064\text{ V}$  lub  $512/85,33=6,00023\text{ V}$ . Dokładność takich obliczeń zależy od biblioteki działań zmiennoprzecinkowych zastosowanego kompilatora. Przeprowadziłem symulację obliczeń w docelowym układzie z zastosowaniem emulatora/programatora PICKit3 i środowiska MPLABX IDE. Okazało się, że dla kompilatora MPLAB XC32 V1.30 i stosowania zmiennych typu *float* dokładność zaokrągleń obliczeń dla interesującego mnie drugiego miejsca po przecinku

jest bardzo dobra. Procedurę pomiaru napięcia i konwersji na wartość w woltach pokazano na **listingu 4**.

W przyrządach rejestrujących mierzone wielkości jednym z głównych parametrów jest interwał wykonywania pomiarów. Duża częstotliwość samych pomiarów i ich rejestracji będzie powodowała szybkie zwiększanie objętości pliku z zarejestrowanymi wartościami. Może to też utrudniać analizę wykonanych pomiarów. Dlatego należy ustawić interwał pomiarów na takim poziomie, by ilość rejestrowanych pomiarów była wystarczająca, ale nie wyższa niż to konieczne. W mierniku jest dostępna funkcja wykonywania pomiarów z interwałem od 30 sekund do 10 minut, ustawianym z rozdzielczością 30 sekund. To oczywiście jest tylko jedna z możliwości. Dla innych zastosowań może to być zakres nieodpowiedni. Do odliczania czasu pomiędzy pomiarami jest używany moduł zegara czasu rzeczywistego RTCC. Konstrukcja urządzenia nie przewiduje stosowania podtrzymania baterijnego. W momencie wyzwolenia pomiarów zegara jest ustawiany na czas 00:00:00 i ze stałe ustawioną datą. Czas i data zegara są używane również w trakcie tworzenia pliku tekstowego z wynikami pomiarów. Moduł RTCC inicjalizuje się i ustawia funkcją biblioteczną *RtccOpen* z argumentami zawierającymi czas, datę i kalibrację dryftu częstotliwości taktującej – *RtccOpen(0, 0x14012702, 0);*

REKLAMA



Projekty na...  
**STM32**

[www.stm32.eu](http://www.stm32.eu)

life.augmented

**KAMAMI**

Odmierzany czas z modułu RTCC jest odczytywany przez funkcje biblioteczną *GetTime*. Odczytane wartości są zapisywane do struktury *Time*. Moment wyzwolenia pomiaru jest określany przez funkcję *MeasureTrig* (interwał) pokazaną na **listingu 5**. Po odczytaniu czasu z RTCC jest on porównywany z czasem zapisanym w strukturze *TimeInt*. Jeżeli czas odczytany jest równy czasowi zapisanemu w *TimeInt*, to funkcja zwraca „1” sygnalizując, że należy wykonać pomiar i zapisanie danych do pliku. Ponadto, do czasu w strukturze *TimeInt* jest zapisywany nowy czas równy czasowi bieżącemu plus interwał, tak by po upływie czasu równemu interwałowi wyzwolić kolejny pomiar. Na początku pomiarów do *TimeInt* jest wpisywana wartość interwału (**listing 6**). Można tak zrobić, bo zegar odlicza od wartości 00:00:00.

### Karta SD – zapisywanie pomiarów w pliku tekstowym

Karta SD wykorzystywana do zapisywania pomiarów ma wiele zalet. Jest powszechnie dostępna, tania i łatwo napisać procedurę jej obsługi. Ponieważ w wielu zastosowaniach (na przykład w aparatach fotograficznych) potrzeba coraz większych pojemności, to możemy zastosować posiadane, stare, sprawne karty o mniejszych pojemnościach, np. poniżej 1 Gb. Żeby móc odczytywać zapisane pliki w czytnikach podłączonych do komputera PC, to karta musi być sformatowana zgodnie z systemem plików FAT16 lub FAT32.

W mierniku zostały wykorzystane procedury biblioteki obsługi systemu plików FAT16 i FAT32 dostarczane bezpłatnie przez Microchipa. Jest to znana i ciągle rozwijana biblioteka pozbawiona znaczących błędów. Mikrokontroler komunikuje się z kartą za pomocą interfejsu szeregowego SPI. W trakcie inicjalizacji karty interfejs SPI jest emulowany programowo. Po zainicjowaniu karty i odczytaniu rejestru CSD jest ustalana pojemność karty i dopuszczalna prędkość transmisji. Wtedy do transmisji SPI jest wykorzystywany interfejs SPI mikrokontrolera.

Obsługa karty SD sprowadza się do:

- Detekcji włożenia karty w złącze.
- Zainicjowania systemu plików.
- Otwarcia pliku do zapisu.
- Zapisywania danych tekstowych w otwartym pliku.
- Zamknięcia pliku.

Detekcja włożenia karty w gniazdo jest wykonywana poprzez sprawdzanie poziomu logicznego na wejściu SD\_CD (*Card Detect*). Jeżeli SD\_CD ma poziom niski, to oznacza, że karta jest włożona w złącze. Takie sprawdzanie jest wykonywane przez każdym zainicjowaniem systemu plików i jeżeli program wykryje brak karty, to jest generowany odpowiedni komunikat o błędzie i pomiar nie będzie mógł być wykonany. Inicjowanie systemu plików jest wykonywane przez

funkcję biblioteczną *FSInit*. Program próbuje to robić 100 razy w pętli. Jeżeli w trakcie tych prób *FSInit* zawsze zwróci zero, to zostanie zgłoszony błąd systemu plików i pomiar nie zostanie wykonany (**listing 7**).

Po zainicjowaniu systemu plików można próbować otworzyć plik tekstowy za pomocą funkcji *FSfopen(filename, w)*. Pierwszy argument to wskaźnik na bufor z nazwą pliku, a drugi to atrybut pliku. Plik w którym mamy coś zapisywać musi mieć atrybut „w” (*write*). Funkcja *FSfopen* sprawdza czy w głównym katalogu na karcie jest plik o tworzonej nazwie. Jeżeli tak, to poprzedni plik jest niszczone i w jego miej-

scie jest tworzony nowy plik. W obecnej wersji program nie sprawdza czy w katalogu głównym jest plik o takiej samej nazwie i nie wysyła ostrzeżenia o zniszczeniu poprzedniego pliku. Jeżeli poprzedni plik jest potrzebny, to trzeba go przekopiować na inny nośnik lub do jakiegoś podkatalogu. Prawidłowo otwarty plik zwraca wskaźnik na strukturę pliku *FSFILE*. Jeżeli zostanie zwrócona wartość zera, to oznacza to niepowodzenie otwarcia pliku i w tym przypadku pomiar i jego rejestracja nie będą mogły zostać wykonane (**listing 8**).

Do prawidłowo otwartego pliku można zapisać dane używając funkcji *WriteFileData*

```

Listing 6. Inicjalizacja zegara
//zerowanie zegara i wpisanie interwału
void StartTime(void){
    int period;
    period =ReadEE(APERIOD); //ustawiony wcześniej interwał pomiaru
    period=period*30;
    TimeInt.min=period/60;//zapisanie do TimeInt
    TimeInt.sec=period%60;
    RtccOpen(0, 0x14012702, 0);//ustawieni czasu 00:00:00 I daty
    GetTime();
}
    
```

```

Listing 7. Inicjalizacja systemu plików
for(i=0;i<100;i++){
    if(FSInit()) break;
}
if(i>998) error|=4; //bład systemu plików
else error&=~4; //kasowanie błędu systemu plików
    
```

```

Listing 8. Otwarcie pliku i sygnalizacja błędu karty SD
pointer =FSfopen (filename, "w");
if (pointer == NULL){
    error|=4;//blad systemu plików
}
else
error&=~4);

if((error&4)==4)
{
    atr=0;
    LcdTxtPgm(" Bład karty SD ",0 ,1);
    LcdTxtPgm(" Pomiar nie moze byc",0 ,2);
    LcdTxtPgm(" wykonany",0 ,3);

    LcdTxtPgm(" Nacisnij STOP",0 ,5);
    while(1){
        key=GetKey();
        if(key==KSTOP)//nacisniecie STOP - koniec pomiarow
        {
            ClrLcd();//czyszc ekran
            DelayMs(100);
            return;
        }
    }
}
    
```

```

Listing 9. Funkcja ustawiania interwału pomiarów
//ustawianie interwału pomiaru
void FsSetPeriod(void){

    int period,period1;
    unsigned char sec,min,key;
    period=ReadEE(APERIOD);
    ClrLcd();
    LcdTxtPgm(" ",2, 1);
    LcdTxtPgm(" Funkcja ",4, 0);
    LcdTxtPgm(" Interwał Pomiarow ",1, 1);
    atr=1;
    LcdTxtPgm("obrot zmiana nac. ACC",0, 7);
    atr=0;
    LcdTxtPgm("pomiar co : ",0, 3);
    while(1){
        period1=period*30;
        min=period1/60;
        sec=period1%60;
        Disp2Dec(min,9,3);
        Disp2Dec(sec,12,3);
        key=GetKey();
        if(key==IUP) ++period;
        if(period>20) period=20;
        if(key==IDOWN) {
            --period;
            if(period==0) period=1;
        }
        if(key==IACC)
            break;
    }
    WriteEE(APERIOD, period);
}
    
```



z argumentami: długość bufora z danymi, wskaźnik na bufor z danymi i wskaźnik na strukturę otwartego pliku. W naszym wypadku w buforze z danymi będą informacje tekstowe o zmierzonych napięciach, rozdzielone znakami tabulacji, zakończone znakiem nowej linii i powrotu karetki. Każde wywołanie `WriteFileData` spowoduje zapisanie nowego wiersza z pomiarami. Plik nie wymaga żadnych dodatkowych wpisów (typu nagłówka), ponieważ jest plikiem tekstowym.

Po zakończeniu pomiarów plik musi zostać zamknięty, bo w przeciwnym razie nie będzie go można odczytać. Do tego celu jest przeznaczona funkcja `FSfclose(pointer)`. Jeżeli funkcja zwróci zero, to sygnalizuje, że zamknięcie pliku przebiegło prawidłowo.

### Funkcje miernika – menu funkcyjne

Miernik został wyposażony w funkcje ustawiania interwału pomiarów, ustawiania na-

pięcia alarmu i testu wejść. Funkcje są wybierane z menu funkcyjnego, do którego wchodzi się z programu głównego po naciśnięciu ośki impulsatora.

Funkcja ustawiania interwałów pomiarów pozwala na ustawienie i zapamiętanie interwału od 30 sekund do 10 minut z rozdzielczością 30 sekund (**listing 9**).

Funkcja napięcia alarmu pozwala na zaprogramowanie wartości progowej. Jeżeli napięcie spadnie poniżej tej wartości miernik może wykonać jedną z akcji:

- Zatrzymać pomiar.
- Zaznaczyć w wiersz z pomiarami poprzez dopisanie na końcu `U<Ud`.
- Nie wykonywać żadnej akcji.

Warunek sprawdzania wartości progowej jest sprawdzany przy każdym pomiarze. Jeżeli napięcie spadnie poniżej pomiędzy pomiarami i potem powróci do wartości wyższej w trakcie pomiaru, to nie zostanie to zarejestrowane i miernik nie wykona żadnej akcji. Funkcję ustawiania wartości progowej pokazano na **listingu 10**. Najpierw jest ustawiana wartość napięcia progowego, a potem rodzaj wykonywanej akcji. Funkcja testująca wejścia ciągle mierzy napięcia na wszystkich wejściach miernika i wyświetla je na wyświetlaczu miernika. Ponieważ wyświetlenie wszystkich 14 napięć na jednym ekranie nie jest możliwe, to zostało ono podzielone na 2 ekrany po 7 pomiarów. Przełączanie pomiarów odbywa się przez kręcenie ośką impulsatora.

### Montaż, uruchamianie i obsługa miernika

Woltomierz zmontowano na 3 płytkach: płytce sterownika, analogowej i karty SD. Połączenia pomiędzy płytkami są wykonywane za pomocą złącz IDC10 i IDC20 i odpowiednich taśm przewodów.

Schemat montażowy płytki sterownika pokazano na **rysunku 5**. Jej poprawne zmontowanie wymaga przylutowania mikrokontrolera, ale jest on umieszczony w stosunkowo dużej obudowie (jak na dzisiejsze standardy) i ta czynność nie powinna sprawić problemów, podobnie jak montaż pozostałych elementów. Mikrokontroler programuje

**Listing 10. Funkcja ustawienia napięcia alarmu**

```
//funkcja ustawiania napięcia alarmu
//zmienna failvolt zawiera ustawiony próg napięcia
void FsSetUd(void) {
    int ud;
    char key, action;
    char buf[10], buf1[10];
    buf[0]=ReadEE(AULIMIT);
    buf[1]=ReadEE(AULIMIT+1);
    buf[2]=ReadEE(AULIMIT+2);
    buf[3]=ReadEE(AULIMIT+3);
    buf[4]=0;
    ud=(buf[0]-0x30)*100;
    ud=ud+(buf[1]-0x30)*10;
    ud=ud+(buf[3]-0x30);
    ClrLcd();

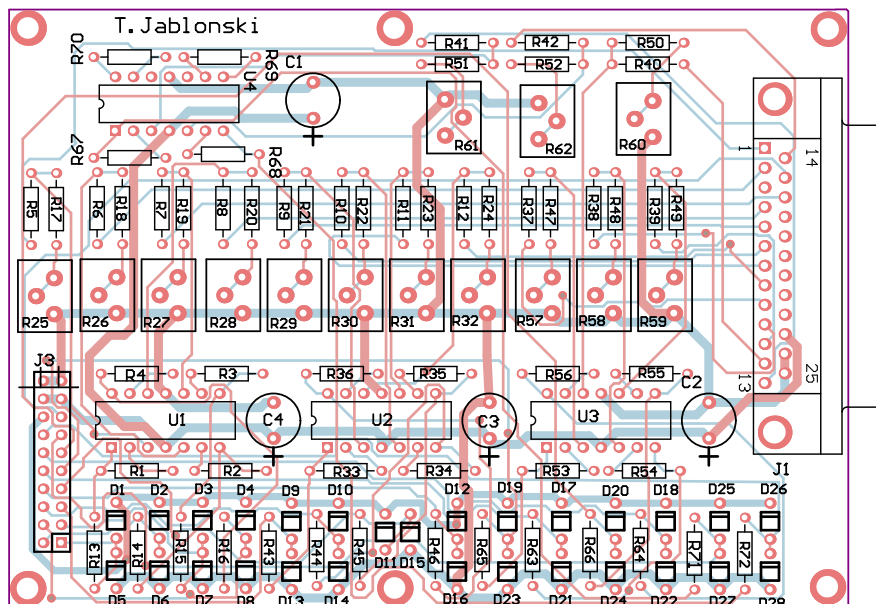
    LcdTxtPgm(„ Funkcja „, 4, 0);
    LcdTxtPgm(„ Ustaw Alarm Napięcia „, 0, 1);
    atr=1;
    LcdTxtPgm(„obrot zmiana nac. ACC“, 0, 7);
    atr=0;
    LcdTxtPgm(„USTAW NAPIECIE“, 2, 3);
    LcdTxtPgm(„ Ud=      V“, 2, 4);
    while(1) {
        LcdTxt(buf, 6, 4);
        key=GetKey();
        if(key==IUP)
        {
            ud=ud+1;
            if(ud>110) ud=120; //maksymalnie 11V
        }
        if(key==IDOWN)
        {
            ud=ud-1;
            if(ud<10) ud=10; //minimalnie 1V
        }
        if(key==IACC) break;
        sprintf(buf1, \"%d\", ud);
        if(ud>=100) {
            buf[0]=buf1[0];
            buf[1]=buf1[1];
            buf[2]='.';
            buf[3]=buf1[2];
            buf[4]=0;
        }
        else
        {
            buf[0]='0';
            buf[1]=buf1[0];
            buf[2]='.';
            buf[3]=buf1[1];
            buf[4]=0;
        }
    } //end while(1)
    //zapisanie ustawionego napięcia alarmu
    failvolt=(float)ud/10.0;
    WriteEE(AULIMIT, buf[0]);
    WriteEE(AULIMIT+1, buf[1]);
    WriteEE(AULIMIT+2, '.');
    WriteEE(AULIMIT+3, buf[3]);
    action=ReadEE(AUF);
    //ClrLcd();
    LcdTxtPgm(„Po wykryciu U<Ud“, 1, 5);
    //ustawienie akcji po wykryciu obniżenia napięcia
    while(1) {
        if(action==0) LcdTxtPgm(„ Nic nie rob „, 2, 6);
        if(action==1) LcdTxtPgm(„Koniec pomiarow“, 2, 6);
        if(action==2) LcdTxtPgm(„Zaznacz w pliku“, 2, 6);
        key=GetKey();
        if(key==IUP)
        {
            action=action+1;
            if(action==3) action=0;
        }
        if(key==IDOWN)
        {
            action=action-1;
            if(action<0) action=2;
        }
        if(key==IACC) break;
    }
    WriteEE(AUF, action);
}
```

REKLAMA

Projekty na STM32

www.stm32.eu

ST KAMAMI life.augmented

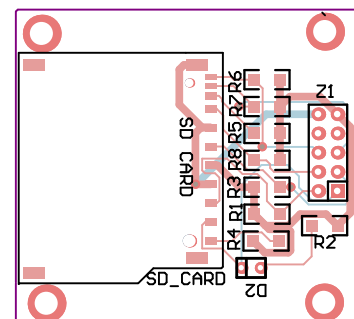


Rysunek 6 Schemat montażowy płytki analogowej

się po zasileniu poprzez złącze J1 ICSP z wyprowadzeniami zgodnymi z programatorem PICKit3.

Schemat montażowy modułu analogowego pokazano na **rysunku 6**. Jest ona zbudowana z elementów do montażu przewlekane. Warto zastosować podstawki precyzyjne pod układy LM324. Można je będzie łatwo wymienić w przypadku uszkodzenia.

Złącze karty SD oraz kilka dodatkowych elementów (rezystory podciągające i dioda LED) zostały umieszczone na osobnej płytce (**rysunek 7**). Płytkę jest połączona z płytka sterownika za pomocą złącz IDC10 zaciskanych na taśmie przewodów. Pozwala to na większą elastyczność w umieszczaniu złącza w obudowie.



Rysunek 7. Schemat montażowy płytki karty SD

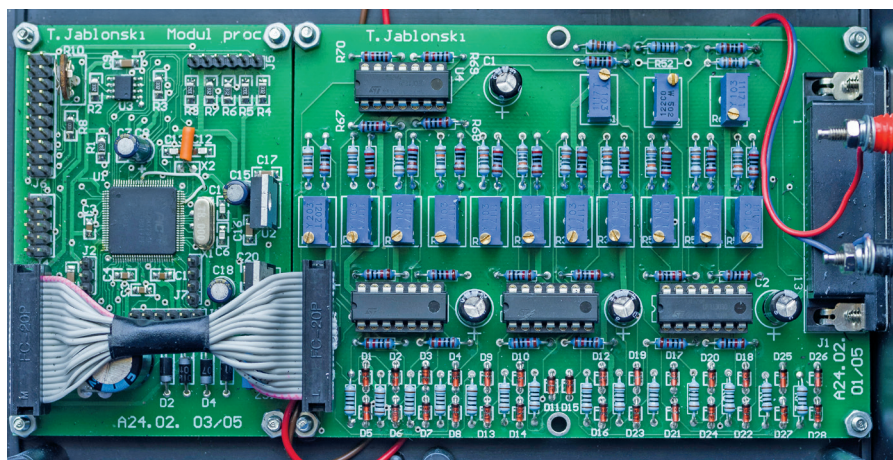
nika. Będzie to dodatkowa sygnalizacja prawidłowości pracy tej części miernika, która odpowiada za zapis wyniku na karcie SD. Schemat ideowy płytki pokazano na **rysunku 10**, natomiast jej wygląd na **fotografii 11**.

Elementy manipulacyjne: impulsator ze zwiernym stykiem oraz styki START i STOP są dołączane do złącza J5. Styk impulsatora A łączymy z sygnałem SW1, styk B z SW2. Styk zwierany naciśnięciem osi impulsatora z SW3, styk START z SW4, a styk STOP z SW5. Wszystkie styki łączymy tak by zwieraly sygnały SWx z masą. Do wprowadzenia mierzonych sygnałów analogowych jest przeznaczony złącze DSUB25 na płytce analogowej. W modelowym rozwiązaniu wykonałem kabel z wtykiem DSUB25. Przewody z drugiej strony zostały zakończone wtykami bananowymi połączonymi z krokodylkami.

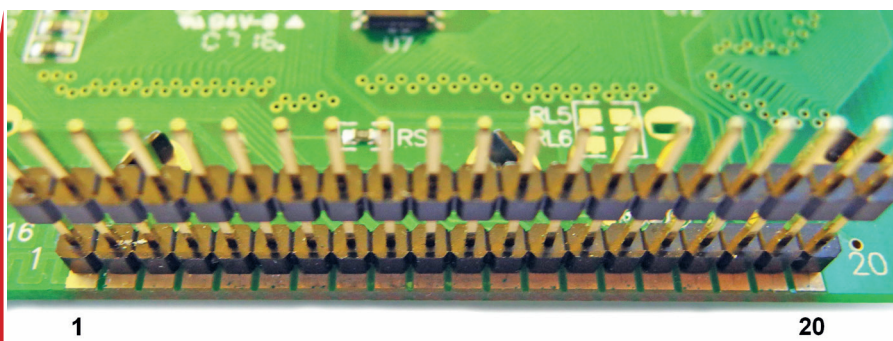
Uruchamianie miernika rozpoczynamy od zasilania modułu sterownika napięciem

Po zmontowaniu wszystkich płytek trzeba je połączyć złączami IDC. Na **fotografii 8** pokazano połączenie pomiędzy płytkami sterownika i analogową. Wyświetlacz LCD podłącza się do złącza J6. Numery pinów tego złącza są zgodne z numerami pinów wyświetlacza – pokazano je na **rysunku 9**. Płytkę złącza karty SD jest łączona z płytka sterownika (złącze J4) za pomocą 10-żyłowego przewodu zakończonego dwoma złączami IDC10. Dioda D2 sygnalizuje sygnał wybrania karty SD. Można ją podłączyć za pomocą kabli i umieścić w obudowie mier-

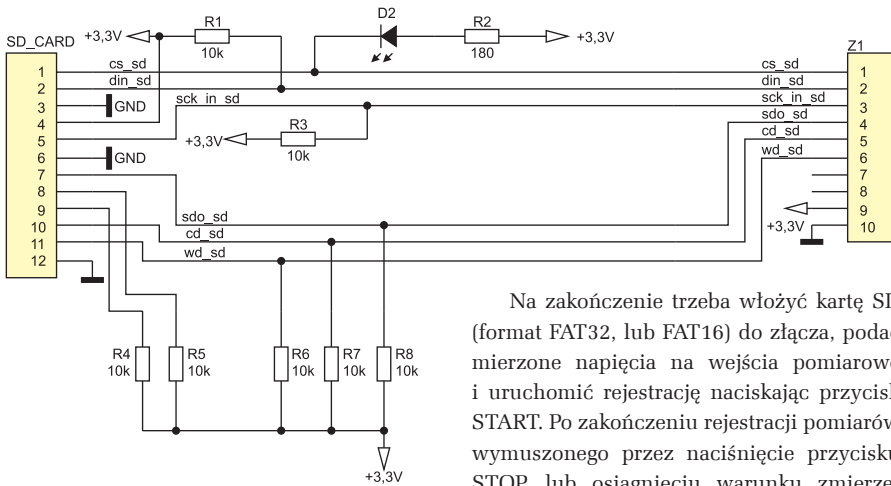
Pin wyświetlacza/złącza J6	Funkcja
1 – GND	Masa
2 – Vcc (+5 V)	Plus zasilania
3 – Vo	Zasilanie driverów (kontrast wyświetlacza)
4 – GRS	Wybór dane/instrukcje
5 – GRW	Sygnal zapis/odczyt
6 – GE	Sygnal aktywacji magistrali (enable)
7 – GD0	Linia danych
8 – GD1	Linia danych
9 – GD2	Linia danych
10 – GD3	Linia danych
11 – GD4	Linia danych
12 – GD5	Linia danych
13 – GD6	Linia danych
14 – GD7	Linia danych
15 – CS1	Wybór sterownika KS108 #1
16 – CS2	Wybór sterownika KS108 #2
17 – RST	Zerowanie sterownika
18 – VEE	Wyjście ujemnego napięcia z wewn. przetwornicy
19 – LEDA	+ zasilania podświetlenia
20 – LEDK	- zasilania podświetlenie



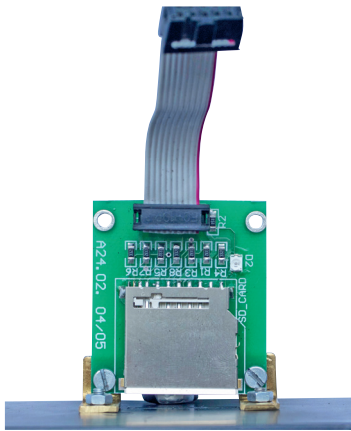
Fotografia 8. Połączenie pomiędzy płytkami sterownika i analogową



Rysunek 9. Wyprowadzenia wyświetlacza



Rysunek 10. Schemat płytki karty SD



Fotografia 11. Płytki karty SD

ok. 7...8 V AC. Po sprawdzeniu poprawności napięć zasilających +3,3 V DC i +5 V DC można dołączyć programator i zaprogramować pamięć Flash mikrokontrolera. Użyłem do tego celu programatora PICkit3 i programu MPLAB X. W kolejnym kroku dołączamy wyświetlacz LCD. Jeżeli po włączeniu zasilania na wyświetlaczu pojawi się ekran z ustawieniami początkowymi, to oznacza, że mikrokontroler jest poprawnie zaprogramowany, jest taktowany i prawidłowo wykonano połączenie: płytki sterownika – wyświetlacz. Jeżeli na wyświetlaczu nic się nie wyświetla, to w pierwszej kolejności trzeba spróbować wyregulować kontrast potencjometrem R10, potem sprawdzić poprawność połączenia i w końcu poprawność montażu płytki sterownika.

Po połączeniu wszystkich elementów urządzenia między sobą: płytek sterownika, analogowej, karty SD, wyświetlacza i elementów manipulacyjnych można przystąpić do sprawdzenia działania kompletnego miernika. Po uruchomieniu na ekranie głównym wyświetlane są: interwał pomiaru, napięcie alarmu i ewentualny błąd karty SD (jeżeli nie jest włożona do złącza). Po przyciśnięciu ośki impulsatora wchodzimy do menu funkcyjnego i sprawdzamy działanie wyżej opisanych funkcji. Przy okazji sprawdzamy poprawność działania impulsatora.

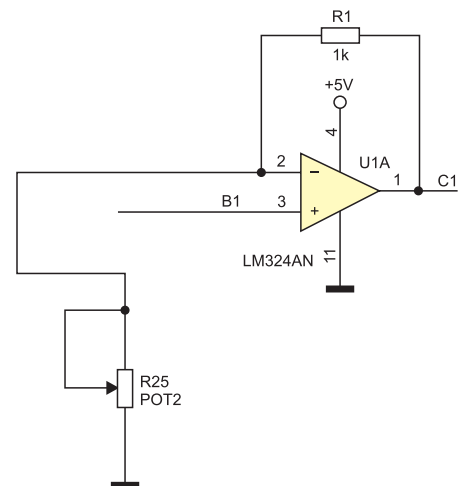
Na zakończenie trzeba włożyć kartę SD (format FAT32, lub FAT16) do złącza, podać mierzone napięcia na wejścia pomiarowe i uruchomić rejestrację naciskając przycisk START. Po zakończeniu rejestracji pomiarów wymuszonego przez naciśnięcie przycisku STOP, lub osiągnięciu warunku zmierzenia napięcia niższego niż napięcie alarmu można wyjąć kartę SD ze złącza i odczytać plik *pomiar.txt*. Jeżeli wszystkie czynności sprawdzające przebiegną pomyślnie to można zakończyć proces uruchamiania i testowania miernika.

### Uwagi końcowe

Przedstawiony tutaj miernik umożliwi pomiar napięcia w 14 kanałach pomiarowych. Zastosowane rozwiązania sprzętowe i użycie mikrokontrolera o bardzo dużych zasobach pozwalają na szereg modyfikacji pracy urządzenia. Bez problemu można zmodyfikować w prosty sposób program i dzielniki wejściowe, aby na różnych wejściach pomiarowych mierzyć napięcia w różnych zakresach pomiarowych. Ponieważ użyłem buforów wejściowych z rezystorem w obwodzie sprzężenia zwrotnego, to w łatwy sposób można bufor przekształcić we wzmacniacz nieodwracający o regulowanym wzmacnieniu. Na przykład, dla pierwszego wejścia pomiarowego trzeba zewrzeć rezystor R5, wylutować rezystor R17 i połączyć rezystor R25 z wejściem odwracającym wzmacniacza operacyjnego. Powstaje w ten sposób klasyczny wzmacniacz nieodwracający o wzmacnieniu równym  $G=1+R1/R25$  (rysunek 12). Ponieważ w takim przypadku napięcie wejściowe jest podawane bezpośrednio na wejście wzmacniacza operacyjnego trzeba zadbać o dodatkowe zabezpieczenia przez przepięciami, lub mieć pewność, że napięcie mierzone nie będzie wyższe niż napięcie zasilania.

Układ pomiarowy z rys. 12 był wykorzystywany do pomiaru prądu za pomocą miernika cęgowego z wyjściem napięciowym o współczynniku 10 mV/A i zakresie pomiarowym 50 A. Dla 50 A na wyjściu miernika będzie 500 mV. Aby na wyjściu wzmacniacza uzyskać napięcie 3,3 V, wzmacnienie  $G=3,3\text{ V}/0,5\text{ V}=6,6$ . W praktyce można podać na wejście 0,5 V i tak ustawić R25, aby na wyjściu wzmacniacza było 3,3 V.

Czternaście analogowych kanałów pomiarowych umożliwia jednoczesny pomiar wielu różnych wielkości analogowych, któ-



Rysunek 12. Zmodyfikowane wejście pomiarowe

re mogą być konwertowane na napięcie. Jednocześnie po modyfikacji układu wejściowego można sobie wyobrazić pomiar sygnałów analogowych i wielkości cyfrowych. W czasie testów wykonywałem i rejestrowałem pomiar temperatury za pomocą czujnika DS18B20 z magistralą 1-wire.

Pewnym ograniczeniem jest tworzenie w momencie wyzwolenia pomiaru jednego pliku o takiej samej nazwie. Wynika to z tego, że zegar RTC nie jest ustawiany i nie pracuje z czasem rzeczywistym (brak jest baterijnego podtrzymania). Plik *pomiar.txt* jest za każdym razem tworzony na nowo i z czasem 00:00:00. Można ten problem obejść przez wykrywanie czy plik o tworzonej nazwie już istnieje i jeżeli tak, to zostanie wykonana zmiana jego nazwy na przykład na *pomiar1.txt*.

Projekt może być bazą do budowy bardziej lub mniej zaawansowanego rejestratora mierzonych wielkości. Pierwotnie do celu rejestracji chciałem wykorzystać połączenie USB i aplikację uruchamianą na komputerze PC, ale w końcu rozwiązaniem z kartą SD wydało mi się bardziej uniwersalne. Rejestracja na karcie wymaga tylko zasilania urządzenia, a zapisane dane można potem bez problemu zapisać w pamięci komputera i dowolnie potem analizować lub archiwizować.

Tomasz Jabłoński, EP

REKLAMA

Projekty na...  
**STM32**  
  
[www.stm32.eu](http://www.stm32.eu)