

# Włącznik urządzeń zasilanych z sieci 230 V AC

Sterownik urządzeń zasilanych z sieci 230 V AC jest niewątpliwym udogodnieniem życia. Zapewne każdy zetknął się z sytuacją, kiedy chciał włączyć lub wyłączyć dane urządzenie o konkretnej godzinie bądź po upływie określonego czasu. Dzięki połączeniu zegara czasu rzeczywistego oraz mikrokontrolera, możemy zostać wyręczeni w tej kwestii przez prezentowany poniżej sterownik.

**Rekomendacje:** urządzenie przyda się w automatyce domowej, do sterowania zraszaniem ogrodu itp.

Schemat ideowy sterownika pokazano na **rysunku 1**. Urządzenie kontrolowane jest przez mikrokontroler ATmega16. Taktowanie mikroprocesora jest ustawione na wewnętrzny oscylator RC o częstotliwości 8 MHz. Cały układ zasilany jest napięciem sieciowym 230 VAC. Poprzez transformator, mostek prostowniczy oraz kondensatory otrzymujemy napięcie rzędu kilkunastu woltów, które stabilizujemy poprzez układ L7805 do 5 V DC.

Urządzenie jest wyposażone w wyświetlacz LCD 2x16 ze sterownikiem HD44780. Do sterowania użytkownik ma do dyspozycji 5 przycisków, które są ułożone w taki sposób, aby intuicyjnie domyśleć się ich przeznaczenia (**rysunek 2**). Do dyspozycji użytkownika jest także zworka 2 pozycyjna, za pomocą której możemy blokować lub odblokować urządzenie. Oczywiście, zamiast zworki w prosty sposób można podpiąć dowolny przełącznik, wedle własnych potrzeb. Elementami odpowiadającymi za włączanie lub wyłączanie urządzeń końcowych są dwustykowe przekaźniki z cewką na 12 V. Do styków przekaźników doprowadzone są napięcia sprzed transformatora – napięcie sieciowe 230 V AC. Styki są połączone ze złączami ARK. Po załączeniu przekaźnika na złączu ARK występuje napięcie 230 V AC.

Odmierzaniem czasu i obsługa kalendarza zajmuje się układ scalony PCF8583F. Jest on taktowany rezonatorem kwarcowym o częstotliwości 32,768 kHz. Układ ma wyjście generujące przerwania, które wykorzystano w tym projekcie. Przerwanie jest gene-



rowane dokładnie co 1 sekundę. Warto także dodać, iż po wyłączeniu zasilania sieciowego nie tracimy zapisanej daty oraz godziny z układu PCF, ponieważ zasilanie tego układu jest podtrzymywane za pomocą baterii CR2032. W urządzeniu zamontowano dodatkowe złącze służące do programowania mikrokontrolera wlutowanego na płytce drukowanej (ISP). Ułatwia to uruchomienie układu oraz aktualizację oprogramowania. Zarówno na schemacie jak i na płytce możemy zauważyć wyprowadzenie 3-pinowe o nazwie RC5. Jest to złącze, które w tej wersji urządzenia jest nieużywane, ale w przyszłości posłuży do implementacji obsługi zdalnego sterowania za pomocą podczerwieni.

## Zasada działania

Zadaniem sterownika jest włączanie i wyłączanie urządzeń o zadanej godzinie. Zastosowanie transformatora zwiększa wymiary gotowego urządzenia, ale za to jesteśmy w stanie zapewnić zasilanie zarówno dla sterownika, jak i dla dołączonych urządzeń wykorzystując do tego celu pojedyncze gniazdko sieciowe. Do dyspozycji użytkownika są 2 wyjścia, które są sterowane **niezależnie**. W efekcie końcowym, za pomocą jednego gniazdko zasilamy nasz sterownik oraz 2 urządzenia końcowe.

Użytkownik ma możliwość zaprogramowania czasu włączenia lub wyłączenia podłączonych urządzeń za pomocą menu na wyświetlaczu LCD. Mamy do dyspozycji dwa wyjścia, czyli dwa urządzenia, którymi możemy sterować zupełnie niezależnie. Przy pracy programowej jesteśmy w stanie ustawić dwa tryby: tryb cykliczny

## W ofercie AVT\*

AVT-5408 A

### Podstawowe informacje:

- Zasilanie 230 V AC.
- Sterowanie dwoma odbiornikami 230 V AC/7 A.
- Wyjścia przekaźnikowe.
- Dwa niezależne programy: cykliczny (załączanie/wyłączanie cyklicznie, o ustalonych porach) i chwilowy (załączanie na pewien czas).
- Podtrzymanie baterijne zegara RTC.
- Menu z wyświetlaczem i przyciskami.
- Złącze do programowania ISP

### Dodatkowe materiały na CD lub FTP:

<http://ep.com.pl>, user: 62828, pass: 18ofqn10

- wzory płytek PCB
- karty katalogowe i noty aplikacyjne elementów oznaczonych w Wykazie elementów kolorem czerwonym

### Projekty pokrewne na CD/FTP:

(wymienione artykuły są w całości dostępne na CD)

AVT-1714 Automatyczny włącznik kina

domowego EP 12/2012

AVT-1689 Przełącznikowy wyłącznik czasowy

EP 8/2012

\* 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 wymieniony w załączniku pdf 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 C 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>

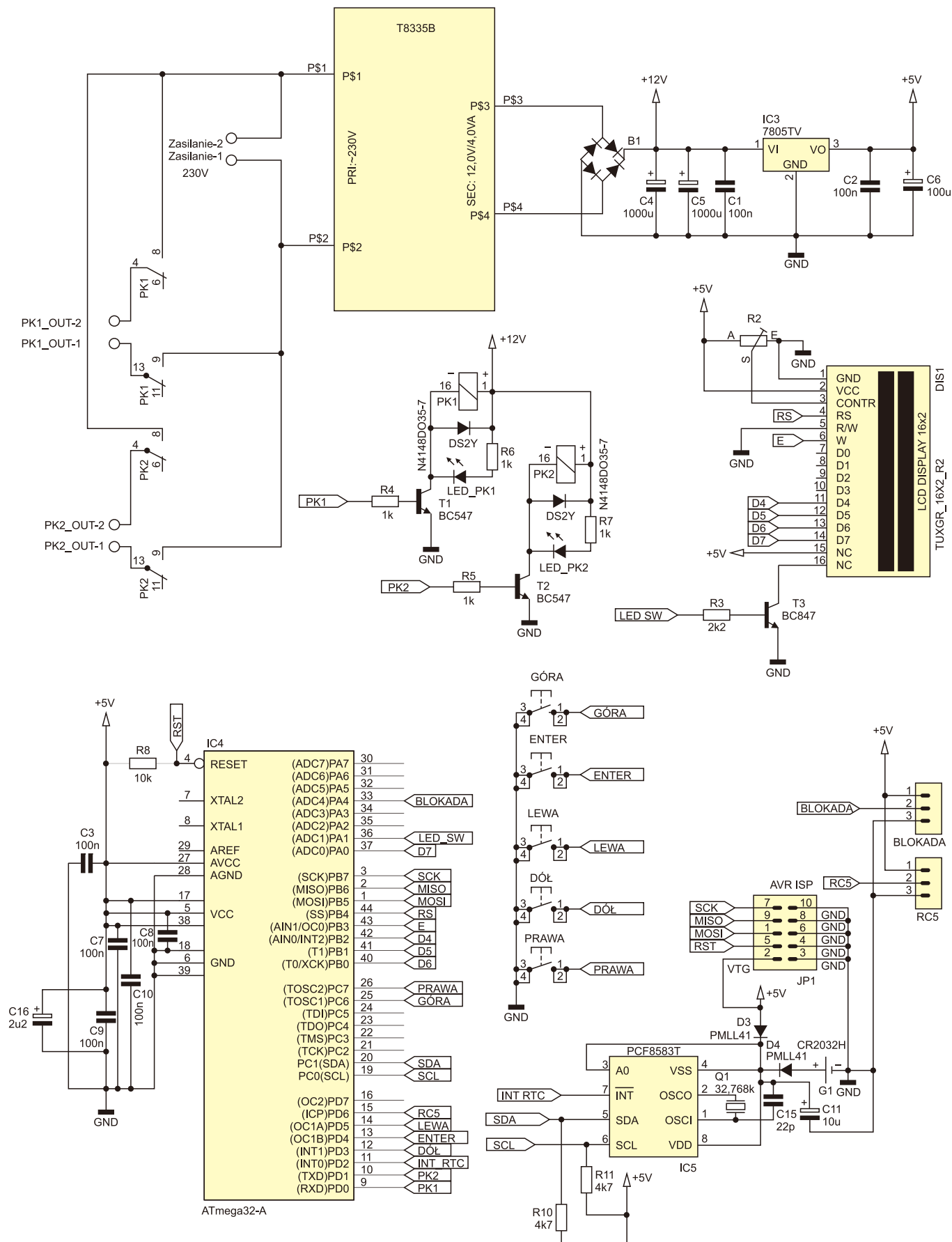
oraz tryb chwilowy. W trybie cyklicznym ustawiamy godzinę, o której ma się włączyć oraz wyłączyć urządzenie. Drugi to tryb chwilowy, w którym ustawiamy po jakim czasie ma się włączyć urządzenie oraz po jakim wyłączyć. Poza pracą programową możemy także na stałe włączyć lub wyłączyć zasilanie w dołączonych urządzeniach.

Użytkownik ma możliwość ustawienia daty oraz czasu, które po wprowadzeniu i zatwierdzeniu zmian są zapamiętywane przez układ PCF8583. Istnieje także

możliwość przywrócenia nastaw domyślnych. Domyślne ustawienia dla zegara i kalendarza, to 21 grudnia 1992, godzina 20:00.

### Obsługa sterownika z poziomu użytkownika

Do obsługi menu służą 4 klawisze, których opis funkcjonalny pokazano na rysun-



Rysunek 1. Schemat ideowy włącznika programowanego

**Wykaz elementów**

**Rezystory:** (SMD 0805)

- R3: 2,2 kΩ
- R4...R7: 1 kΩ
- R8: 10 kΩ
- R10, R11: 4,7 kΩ
- R2: 10 kΩ (potencjometr)

**Kondensatory:**

- C1...C3, C7...C10: 100 nF (SMD 0805)
- C11: 10 μF (SMD 3528)
- C15: 22 pF (SMD 0805)
- C16: 2,2 μF (elektrolityczny)

**Półprzewodniki:**

- B1: mostek prostowniczy 1,5 A/50 V
- D1, D2: 1N4148
- D3, D4: LL4148 (SOD-80)
- IC3: L7805 – IC3
- IC4: ATmega16AU
- IC5: PCF8583F
- LED\_PK1, LED\_PK2: dioda LED, zielona, 3 mm
- T1, T2: BC547
- T3: BC847

**Inne:**

- Transformator T08335B
- PK1, PK2: przekaźnik Zettler AZ822-2C-12DSE
- JP1: gniazdo IDC-10 – JP1
- GORA, DOL, PRAWA, LEWA, ENTER: przyciski 6×3 mm
- DIS1: wyświetlacz LCD 2×16
- Q1: kwarc 32,768kHz
- G1: bateria CR2032
- Złącza ARK2 – 3 sztuki

**ku 2.** Po włączeniu zasilania na wyświetlaczu LCD pojawi nam się jeden z dwóch ekranów w zależności od ustawienia zworki BLOKADA (rysunek 3). W sytuacji gdy urządzenie jest zablokowane, nie reaguje ono na przyciski. Na wyświetlaczu w górnej linii widnieje aktualna data (domyślnie jest 21 grudnia 1992), natomiast w dolnej linii możemy zaobserwować aktualną godzinę (domyślnie godzina 20:00). Kiedy przełączymy zworkę BLOKADA w stan odblokowany, mamy do naszej dyspozycji menu, którego schemat możemy podejrzeć na rysunku 4. Po menu poruszamy się za pomocą przycisków zgodnie ze znakami na wyświetlaczu oraz ułożeniem przycisków (rys. 2). Przyciski PRAWA oraz LEWA to przyciski nawigacyjne. Przyciski GORA i DOL to przyciski do zmiany wartości aktualnie wybranego parametru (dzień, rok, godzina, itp.). Wybór zatwierdzamy środkowym przyciskiem ENTER. Ponadto po odblokowaniu naszego urządzenia zostanie włączone podświetlenie wyświetlacza LCD. Natomiast po jego zablokowaniu, wyświetlacz będzie podświetlony jeszcze przez 10 sekund a następnie podświetlenie zostanie wyłączone.

**Ustawienia przekaźników**

Pierwsze dwie pozycje w menu to ustawienia przekaźnika pierwszego oraz ustawienia przekaźnika drugiego. Zarówno dla pierwszego jak i dla drugiego przekaźnika ustawienia są jednakowe, wobec czego opiszę je tylko raz. Należy jednak pamiętać

o tym, iż oba przekaźniki działają niezależnie od siebie.

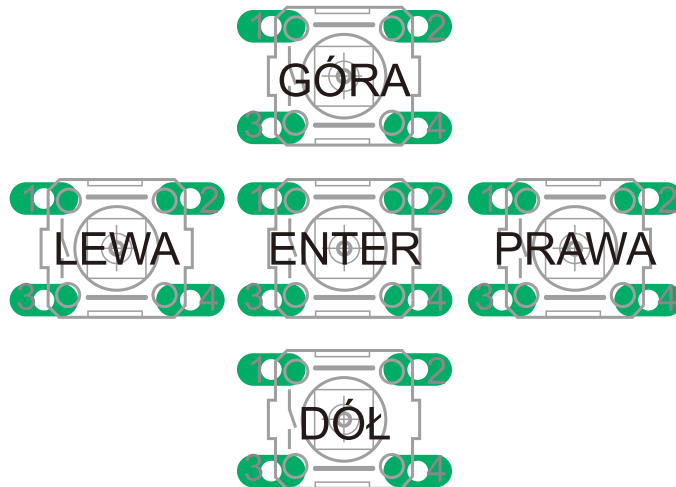
Po wejściu do menu przekaźnika mamy do wyboru kilka opcji. Pierwsze dwie opcje to włączenie lub wyłączenie przekaźnika. Jego stan jest sygnalizowany także poprzez zaświecenie lub zgaśnienie diody LED przy odpowiednim przekaźniku.

Istnieją jeszcze dwie pozycje menu ustawień przekaźników: *Program* oraz *Tryb pracy*. Są one ze sobą bezpośrednio powiązane. W zależności od wyboru trybu, w którym ma pracować przekaźnik, zmieniają się opcje do ustawienia w menu *Program*. W menu *Tryb pracy* mamy do wyboru dwie opcje: *Tryb chwilowy* oraz *Tryb cykliczny*. Po wybraniu i zatwierdzeniu trybu pracy zostajemy automatycznie przeniesieni na pozycję *Program*. Aby ustawić konkretne czasy włączania i wyłączania naszych urządzeń wystarczy wcisnąć *ENTER*.

Tryb chwilowy jest trybem, który działa jednorazowo od momentu ustawienia, stąd nazwa *chwilowy*. Wybierając ten tryb, w menu *Program* będziemy proszeni o ustawienie czasu, po którym ma się włączyć nasze urządzenie. Początkowa wartość to 0 minut, co oznacza natychmiastowe włączenie urządzenia zaraz po zatwierdzeniu wprowadzonych ustawień. Ustawienia czasu regulujemy za pomocą przycisków *GÓRA* i *DÓŁ*. Krok regulacji to 15 minut, a maksymalny czas, jaki możemy ustawić, to 240 minut.

Kiedy wybierzemy czas, po którym urządzenie ma się włączyć, jesteśmy proszeni o podanie czasu, po którym urządzenie ma się wyłączyć od momentu jego włączenia. W tym ustawieniu nie ma już opcji 0 minut, ponieważ w takim przypadku urządzenie by się po prostu nie włączyło. Po wybraniu czasu wyłączenia urządzenia wymagane jest potwierdzenie ustawionych czasów. Jeśli potwierdzimy, wybierając opcję *Tak*, to od tego momentu przekaźnik rozpoczyna programową pracę, zgodnie z ustawieniami, które wybraliśmy. W przypadku braku potwierdzenia (wybór *Nie*) jesteśmy ustawieni w menu głównym a przekaźnik pozostaje w stanie pracy, w którym był przed wcześniejszymi ustawieniami.

Tryb cykliczny to tryb, w którym program jest powtarzany codziennie. Po wyborze tego trybu, w menu *Program* pojawia się drugi wariant ustawień. Jako pierwszy parametr ustawiamy godzinę, o której nasze

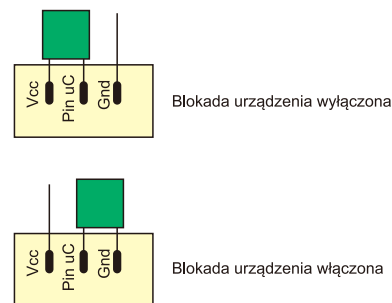


Rysunek 2. Funkcje klawiszy

urządzenie ma się włączać. Tym razem przyciski *PRAWA* oraz *LEWA* służą do wyboru zmienianej wartości. Aktualnie edytowaną wartość możemy poznać po tym, że mruga. Zwiększamy lub zmniejszamy wartości poprzez przyciski *GÓRA* oraz *DÓŁ*. Po ustawieniu czasów wciskamy *ENTER*, co powoduje przejście do menu, w którym ustawiamy dokładną godzinę wyłączenia urządzenia. Analogicznie jak w trybie chwilowym jesteśmy proszeni o potwierdzenie wprowadzonych zmian. Dla przykładu godzinę włączenia ustawiamy na 9:00, a wyłączenia na 10:00. Urządzenie włączy się dokładnie o godzinie 9:00 i przez cały czas, aż do 10:00, będzie włączone. Jeżeli wpiszemy odwrotnie godziny (celowo lub nie), czyli: godzina włączenia: 10:00, a godzina wyłączenia: 9:00, to w tym wypadku urządzenie włączy się o godzinie 10:00, a wyłączy dopiero o godzinie 9:00 kolejnego dnia.

**Ustawienia czasu i daty**

Po wejściu do menu ustawień czasu, możemy ustawić godziny i minuty. Mrugająca liczba to aktualnie zmieniana wartość. Możemy ją zwiększać (*GÓRA*) lub zmniejszać (*DÓŁ*). Po wciśnięciu *ENTER* musimy zatwierdzić zmianę czasu. Jeśli zatwierdziliśmy zmianę, to wartość godziny i minuty jest umieszczana w buforze, a następnie wysyłana do układu PCF. Sytuacja z datą wygląda analogicznie z tą różnicą, że zmienna przechowująca informację o aktualnym



Rysunek 3. Funkcjonowanie zworki „Blokada”

**Listing 1. Funkcja sprawdzania stanu przycisku**

```
uint8_t buttonDown(uint8_t BUTTON)
{
    if( BUTTON )
    {
        _delay_ms(150);
        if( BUTTON ) return 1;
    }
    return 0;
}
```

**Listing 2. Sterowanie podświetlaniem**

```
void lcdLight(uint8_t STATUS)
{
    if(STATUS == 1)
        PORT(LCD_LIGHT_PORT) |= (1 << LCD_LIGHT_PIN);
    else
        PORT(LCD_LIGHT_PORT) &= ~(1 << LCD_LIGHT_PIN);
}
```

**Listing 3. Funkcja obsługi Timera 1**

```
ISR(TIMER1_COMPA_vect)
{
    //Czas wylaczania podswietlenia LCD
    if(led_off_flaga == 1)
    {
        led_off_cnt++;
        if(led_off_cnt > 100)
        {
            led_off_cnt = 101;
            led_off_flaga = 0;
        }
    }
    //Czas mrugania aktywnych wartosci
    blink_cnt++;
    if(blink_cnt > 9)
        blink_cnt = 0;
}
```

roku jest zapisywana do pamięci EEPROM mikrokontrolera. Do układu PCF jest wysyłana jedynie informacja o tym czy dany rok jest przestępny, czy nie. Obliczenia związane z tą operacją wykonuje mikrokontroler. Ze względów bezpieczeństwa zmiana oraz zatwierdzenie zmiany czasu powoduje wyłączenie przekaźników z pracy programowej oraz ich wyłączenie.

**Programowe zerowanie ustawień**

Ostatnią pozycją w naszym menu jest przywrócenie nastaw domyślnych. Jest to nieskomplikowany element menu, którego zadaniem jest przywrócenie domyślnych wartości czasu oraz daty, czyli 21 grudnia 1992, godzina 20:00. Ponadto zerowanie powoduje ustawienie obu przekaźników w pozycji wyłączonej.

**Oprogramowanie**

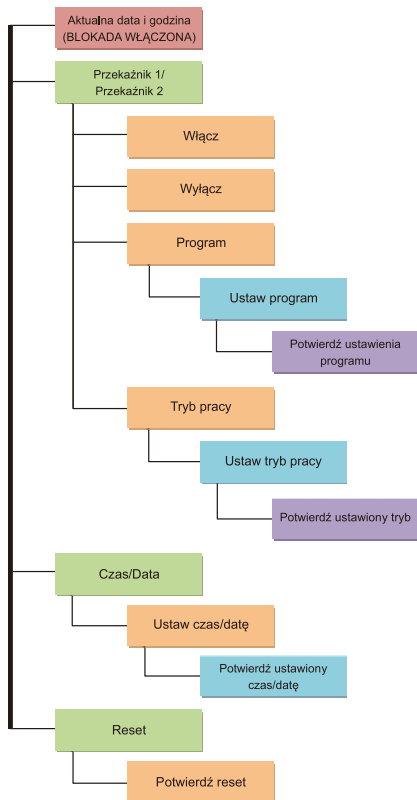
W programie wykorzystałem dwie biblioteki do obsługi peryferii. Pierwszą jest biblioteka do obsługi wyświetlacza LCD. W katalogu LCD znajduje się plik lcd44780.h, w którym możemy zmienić przypisanie poszczególnych pinów wyświetlacza do mikrokontrolera. Drugą biblioteką jest biblioteka komunikacji I<sup>2</sup>C między układem PCF8583 a mikroprocesorem. W katalogu I2C mamy plik o nazwie *i2c\_soft\_cfg.h*, w którym możemy zmienić konfigurację interfejsu I<sup>2</sup>C. W projekcie użyłem jednak takiego dołączenia, aby wykorzystać alternatywne funkcje odpowiednich pinów procesora.

Komunikacja między mikrokontrolerem i układem PCF8583F odbywa się poprzez magistralę I<sup>2</sup>C. Nasz mikroprocesor pełni w tym zestawieniu funkcję urządzenia w trybie *Master*, natomiast układ PCF jest urządzeniem pracującym w trybie *Slave*. Ze względu na to, że będziemy musieli odczytywać oraz zapisywać informacje do i z układu RTC musimy znać jego adres. Zgodnie z notą katalogową układu PCF8583F, kiedy podłączymy na stałe do zasilania pin A0, adres układu ustalany jest na 0xA2. Do wejścia INT0 mikrokontrolera jest dołączone wyprowadzenie INT układu PCF. Dzięki temu jest generowane przerwanie dokładnie co 1 sekundę, w takt „tykania” układu RTC. Za pomocą dwóch funkcji z biblioteki I<sup>2</sup>C, w prosty i przejrzysty sposób możemy odczytywać oraz zapisywać kolejne komórki pamięci układu PCF. Należy pamiętać, że dane przechowywane w pamięci układu PCF są w postaci kodu BCD. Aby poprawnie odczytać lub zapisać wartości,

**Listing 4. Obsługa włączenia/wyłączenia przekaźnika w trybie cyklicznym**

```
case cykliczny: //Tryb cykliczny
{
    if(godzinal_on < godzinal_off)
    {
        if(godzina == godzinal_on && minuta == minatural_on)
            RELAY1_ON;
        if(godzina == godzinal_off && minuta == minatural_off)
            RELAY1_OFF;
        //=====
        if(godzina == godzinal_on && minuta >= minatural_on)
            RELAY1_ON;
        if(godzina == godzinal_on && minuta < minatural_on)
            RELAY1_OFF;
        if(godzina == godzinal_off && minuta < minatural_off)
            RELAY1_ON;
        if(godzina == godzinal_off && minuta >= minatural_off)
            RELAY1_OFF;
        //=====
        if(godzina < godzinal_on || godzina > godzinal_off)
            RELAY1_OFF;
        if(godzina > godzinal_on && godzina < godzinal_off)
            RELAY1_ON;
    }
    if(godzinal_on == godzinal_off)
    {
        if(godzina != godzinal_on)
            RELAY1_OFF;
        if(godzina == godzinal_on)
        {
            if(minuta >= minatural_on && minuta < minatural_off)
                RELAY1_ON;
            else
                RELAY1_OFF;
        }
    }
    if(godzinal_on > godzinal_off)
    {
        if(godzina == godzinal_on && minuta == minatural_on)
            RELAY1_ON;
        if(godzina == godzinal_off && minuta == minatural_off)
            RELAY1_OFF;
        //=====
        if(godzina == godzinal_on && minuta >= minatural_on)
            RELAY1_ON;
        if(godzina == godzinal_on && minuta < minatural_on)
            RELAY1_OFF;
        if(godzina == godzinal_off && minuta < minatural_off)
            RELAY1_ON;
        if(godzina == godzinal_off && minuta >= minatural_off)
            RELAY1_OFF;
        //=====
        if(godzina < godzinal_on && godzina > godzinal_off)
            RELAY1_ON;
        if(godzina > godzinal_on || godzina < godzinal_off)
            RELAY1_OFF;
    }
    break;
}
```





Rysunek 4. Struktura menu użytkownika

należy je rozkodować w przypadku odczytu lub zakodować w przypadku zapisu danych.

Urządzenie jest sterowane za pomocą 5 przycisków oraz przełącznika blokowania i odblokowania urządzenia. Program obsługi przycisku jest napisany w taki sposób, aby uniknąć efektu wielokrotnego wciśnięcia klawisza w sytuacjach, gdy jest to zjawisko niepożądane.

Każdy przycisk ma własną flagę, która jest zerowana wtedy gdy dany przycisk jest zwolniony. Po przyciśnięciu przycisku sprawdzany jest warunek czy jego flaga jest równa zero. Przycisk był wcześniej zwolniony, więc flaga jest wyzerowana i wykonywane są instrukcje obsługi przycisku. Na końcu wykonywania instrukcji, flaga przycisku jest ustawiana. Zakładamy, że przycisk cały czas jest wciśnięty. Po ponownym przejściu programu nie zostaną wykonane instrukcje obsługi przycisku, ponieważ flaga nie została wyzerowana. Dopiero po zwolnieniu przycisku flaga się wyzeruje i będzie można ponownie wcisnąć przycisk. Zapobiega to sytuacjom tego typu, że naciśniemy raz przycisk, a program zamiast jednej pozycji w menu przeskoczy o kilka. Dodatkowo wykorzystałem funkcję sprawdzania stanu przycisku (**listing 1**). Funkcja sprawdza stan przycisku, następnie odczytuje 100 ms sprawdza stan ponownie. Jeśli przycisk jest wciśnięty funkcja zwraca wartość 1, natomiast jeżeli przycisk jest zwolniony, zwraca wartość 0. Czas 100 ms jest założonym, domniemanym czasem drgania styków przycisku.

Do komunikacji z użytkownikiem służyć wyświetlacz LCD. Jest on podświetlany, gdy urządzenie jest odblokowane. Jeśli włączymy blokadę, to wyświetlacz jest podświetlany przez czas 10 sekund, po czym podświetlenie jest wyłączone. Wyprowadzenie sterujące podświetleniem możemy zmienić w pliku *config.h*. Nie musimy przejmować się w tym wypadku zmianą ustawień trybu pracy wyprowadzeń portów oraz poziomami wyprowadzeń, ponieważ jest to robione automatycznie. Do włączania oraz wyłączania podświetlenia napisałem krótką funkcję (**listing 2**). W połączeniu z definicjami, według których ON = 1 a OFF = 0, możemy w bardzo przyjazny dla użytkownika sposób włączać i wyłączać podświetlenie wyświetlacza, np. `lcdLight(ON)` – włączanie, `lcdLight(OFF)` – wyłączenie.

Odliczaniem czasu, po którym ma się wyłączyć podświetlenie zajmuje się `TIMER1`, którego procedurę obsługi możemy zobaczyć na **listingu 3**. `Timer1` jest skonfigurowany tak, aby wykonywał daną procedurę z częstotliwością 10 Hz. Taka częstotliwość pozwala na uzyskanie przerwania od timera co 100 ms. Gdy włączamy blokadę, zostaje ustawiona zmienna *led\_off\_flaga*, co pozwala na uruchomienie procedury odliczania czasu. Zmienna *led\_off\_cnt* jest inkrementowana przy każdym wejściu w przerwanie aż do wartości 100, co jest odpowiednikiem 10 sekund. Po tym czasie flaga jest zerowana, a w dalszym kroku jest wyłączone podświetlenie LCD. W procedurze obsługi przerwania Timera jest też zmienna *blink\_cnt*, która odpowiada za czas mrugania aktualnie zmienianych wartości w niektórych miejscach menu (ustawianie czasu, daty itp.).

## Tryby pracy przełączników

Każdy przełącznik ma przypisaną zmienną, która wskazuje na jego aktualny stan. Ta zmienna to *stan\_pk1* dla przełącznika pierwszego. W naszym układzie przełączniki mogą być w jednym z trzech stanów: przełącznik włączony – odpowiada temu wartości 1 zmiennej *stan\_pk*; wyłączony – *stan\_pk* równy 2; przełącznik wykonuje pracę programową – *stan\_pk* równy 3. Podczas wykonywania pracy programowej każdy z przełączników może przyjąć jeden z dwóch trybów pracy. Tryb cykliczny lub tryb chwilowy.

Oprogramowanie trybu cyklicznego sprowadza się w zasadzie do ustalenia sposobu działania przełącznika. W tym wypadku rozpisałem procedurę krok po kroku analizując każdy możliwy przypadek. Takie rozwiązanie pozwoliło na bezproblemową obsługę tego trybu pracy. W wypadku gdyby użyto flag przełączających przełącznik tylko w konkretnych, granicznych momentach, wystąpiłby problem np. z sytuacjami typu: jest godzina 10:00, a my ustawiamy czas włączenia na godzinę 9:00 i wyłączenia na 11:00. Efekt jest taki, że w dniu dzisiejszym przełącznik się nie włącza. Dopiero kolejnego dnia o godzinie 9:00 zostanie włączony, a o 11:00 wyłączony. Nam zależy na tym, aby przełącznik włączył się także tego dnia, w którym go ustawiamy w wyżej opisanej sytuacji. Warunki, które to zapewniają można prześledzić na **listingu 4**.

Tryb chwilowy natomiast, składa się z kilku etapów. Obsługa tego trybu dla przełącznika numer 1 przedstawiona jest na **listingu 5**. Ustawiając czas włączenia a następnie wyłączenia przełącznika, przypisujemy te

### Listing 5. Obsługa włączenia/wyłączenia przełącznika w trybie chwilowym

```
case chwilowy: //Tryb chwilowy
{
    switch(odlicz_flaga1)
    {
        case 1:
        {
            odlicz_minuty1 = włącz_minuty1*60;
            RELAY1_OFF;
            odlicz_flaga1 = 2;
            break;
        }
        case 2:
        {
            if(odlicz_minuty1 == 0)
            {
                RELAY1_ON;
                odlicz_flaga1 = 3;
            }
            break;
        }
        case 3:
        {
            odlicz_minuty1 = wyłącz_minuty1*60;
            odlicz_flaga1 = 4;
            break;
        }
        case 4:
        {
            if(odlicz_minuty1 == 0)
            {
                RELAY1_OFF;
                odlicz_flaga1 = 0;
                stan_pk1 = 1; //Przelacz stan na wyłączony
            }
            break;
        }
    }
    break;
}
```

**Listing 6. Obsługa przerwania INT0**

```
ISR(INT0_vect)
{
    flaga_int0 = 1;
    if(odlicz_minuty1 != 0)        odlicz_minuty1--;
    if(odlicz_minuty2 != 0)        odlicz_minuty2--;
}
```

**Listing 7. Obsługa menu użytkownika**

```
enum menu
{
    m_ekran_glowny,
    m_pk1,
        m_pk1_wybor,
            m_pk1_wylacz_o,
            m_pk1_wlacz_o,
            m_pk1_wylacz_za,
            m_pk1_wlacz_za,
            m_pk1_program,
            m_pk1_tryb_pracy,
            m_pk1_potwierdz,
    m_pk2,
        m_pk2_wybor,
            m_pk2_wylacz_o,
            m_pk2_wlacz_o,
            m_pk2_wylacz_za,
            m_pk2_wlacz_za,
            m_pk2_program,
            m_pk2_tryb_pracy,
            m_pk2_potwierdz,
    m_czas,
        m_czas_ustaw,
        m_czas_potwierdz,
    m_data,
        m_data_ustaw,
        m_data_potwierdz,
    m_reset,
        m_reset_potwierdz
};
```

wartości do zmiennych *wlacz\_minuty1* oraz *wylacz\_minuty1*, gdzie numer na końcu nazwy zmiennej oznacza numer przełącznika. Gdy zatwierdzamy ustawione czasy zmienna o nazwie *odlicz\_flaga1* jest ustawiana na wartość 1. Powoduje to wystartowanie funkcji *switch*, która w czterech etapach obsłuży nasz tryb. Pierwszy etap to przypisanie do zmiennej *odlicz\_minuty1* wartości zmiennej *wlacz\_minuty1* pomnożonej przez 60. Mnożenie przez 60 wykonuje się po to, aby w dalszych krokach móc łatwo odliczać kolejne sekundy. Następuje także wyłączenie przełącznika oraz przejście do drugiego etapu poprzez ustawienie *odlicz\_flaga1* na wartość 2. W drugim kroku oczekujemy, aż wartość wcześniej ustalonej zmiennej będzie równa zero. Dzieje się to w krótkiej obsłudze przerwania INT0, **listing 6**. Kiedy zmienna *odlicz\_minuty1* była równa zero wtedy nie była wykonywana żadna operacja. Natomiast po przypisaniu jej wartości jest zliczana w dół dokładnie co 1 sekundę. Efektem tego jest zliczenie do 0 po czasie załączenia przełącznika. Wracamy do obsługi trybu.

Nasza zmienna dekrementowana przez ustawiony czas osiąga 0. Następuje wtedy włączenie przełącznika oraz przejście do trzeciego etapu. Trzeci etap jest analogiczny do pierwszego etapu z tą różnicą, że teraz do zmiennej *odlicz\_minuty1* przypisujemy czas w sekundach, po którym urządzenie ma się wyłączyć. Przechodzimy do etapu czwartego, ostatniego. W tym etapie ponownie czekamy aż zmienna odliczająca minuty będzie równa zero. Kiedy już to się stanie, czyli minie czas, po którym urządzenie ma się wyłączyć, przełącznik jest wyłączany. Stan prze-

łącznika jest zmieniany na wyłączony, czyli *stan\_pk1=1*. Zmienna *odlicz\_flaga1* jest zerowana, co oznacza, że ponowne włączenie tego trybu jest możliwe tylko poprzez ustawienie czasów włączenia/wyłączenia, a następnie zatwierdzenie tych ustawień. Tryb ten można w prosty sposób zapętlić, ustawiając w ostatnim kroku zmienną *odlicz\_flaga1* na wartość 1 oraz kasując linijkę zmiany stanu przełącznika. Spowodowałoby to ciągłe wykonywanie się programu. Po ustawieniu czasu włączenia na 15 minut a wyłączenia po 30 minutach, nasze urządzenie zamiast wykonać program raz i na tym zakończyć, wykonywałoby program w nieskończoność. Czyli po 15 minutach przełącznik się włącza, po 30 minutach od włączenia wyłącza, po 15 minutach od wyłączenia włącza itd.

Interakcja z użytkownikiem zajmuje dużą część programu. Menu wykonano z użyciem funkcji *switch*. Bardzo pomocna okazała się możliwość enumeracji, za pomocą której kod stał się bardziej czytelny (**listing 7**). Po menu poruszamy się za pomocą przycisków LEWA oraz PRAWA, które ustalają wartość zmiennej *cursor*. W każdym podmenu, w którym mamy do wyboru jedną z kilku opcji, każdej opcji przypisana jest konkretna wartość kursora. Jeśli staniemy na danej opcji i wciśniemy ENTER, zostaniemy przeniesieni do kolejnego poziomu menu. Jak już opisałem wyżej, przyciski nawigacji i zatwierdzania działają raz po naciśnięciu. W przypadku zmiany wartości godziny, lat, dni itp. jest to raczej mało praktyczne, dlatego też przyciski zmieniające wartości, czyli GORA oraz DOL reagują cały czas po naciśnięciu aż do momentu puszczenia. Ułatwia to usta-

wianie dowolnych wartości. Każda zmiana ustawień, oprócz włączenia lub wyłączenia przełącznika, jest zabezpieczona przed przypadkową zmianą poprzez wprowadzenie potwierdzeń. Komunikaty te, pojawiają się na koniec wszelkich ustawień i aby potwierdzić zmiany należy po raz drugi wcisnąć ENTER. Można anulować wykonanie zmian, stosując się do instrukcji w danym komunikacie.

**Montaż i uruchomienie**

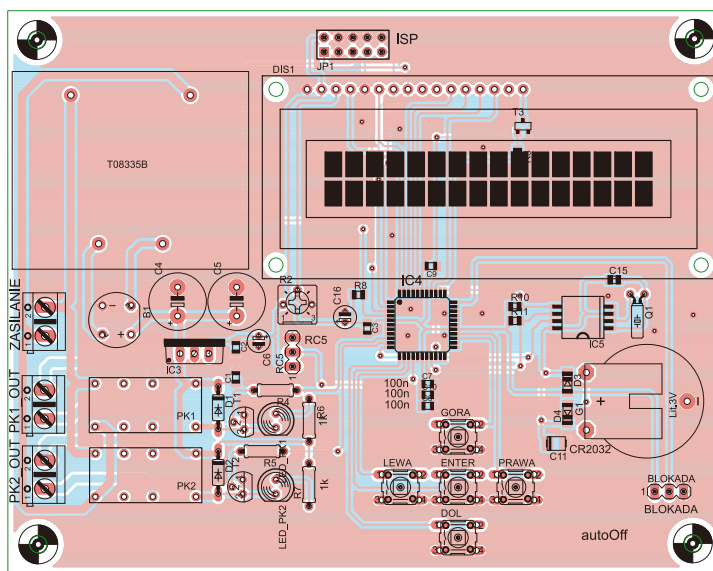
Schemat montażowy sterownika pokazano na **rysunku 5**. Uwaga! Przy uruchamianiu urządzenia oraz podczas jego użytkowania należy zachować szczególną ostrożność i pamiętać o tym, że sterownik jest zasilany napięciem 230 V AC!

Pierwszym i chyba najważniejszym krokiem, który należy wykonać jest sprawdzenie układu zasilania. Przyłutowanie w pierwszej kolejności mostka prostowniczego, dużych kondensatorów, stabilizatora czy transformatora z pewnością utrudni nieco montaż innych elementów. Jednak dzięki takiej kolejności działań możemy przetestować czy nasz układ zasilania dostarczy odpowiednie napięcie do procesora. Uważam, że lepiej spędzić 5 minut dłużej z lutownicą i miernikiem niż uszkodzić mikrokontroler.

Ze względu na niebezpieczne napięcie zasilania i wynikające z tego spore niebezpieczeństwo, na płytce zostały umieszczone cztery otwory montażowe o średnicy 3 mm. Dzięki temu można umieścić układ w obudowie, a co za tym idzie, zmniejszyć ryzyko porażenia prądem. Można użyć dowolnej obudowy, według własnego uznania. Oczywiście, zworka BLOKADA i przyciski sterujące mogą być zastąpione przez inne przełączniki lub przyciski.

Mikrokontroler można zaprogramować po zmontowaniu urządzenia lub użyć zaprogramowanego wcześniej.

**Tomasz Piechowicz**



**Rysunek 5. Schemat montażowy włącznika programowanego**