

Rysunek 2. Schemat montażowy bateryjnego banku energii

Zegar binarny

W rodzinie „binarnych” urządzeń znalazły się już: termometr (AVT1698), higrometr (AVT5426) oraz barometr. Teraz prezentujemy opis kolejnego również ciekawego gadżetu – zegara z wyświetlaczem binarnym.

W ofercie AVT*
AVT-1873 A
Wykaz elementów:
 R1-R3, R11-R17: 4,7 kΩ SMD1206
 R4, R18: 0 Ω SMD1206
 R5-R10: 180 Ω SMD1206
 C1: 220 μF/16 V elektrolityczny THT
 C2-C4: 180 pF SMD1206
 C5, C6: 100 nF SMD1206
 C7: 22 μF/16 V tantalowy SMD B
 D1, D2: BYS10-45 lub podobne
 T1-T4: BC847
 LED1-LED5: czerwone 5 mm
 LED6-LED11: zielone 5 mm
 LED12-LED17: żółte 5 mm
 US1: ATmega48V-10AU TQFP32
 B1: bateria CR2032 + koszyk poziomy THT
 J1: ARK2 5 mm
 J2: goldpin kątowy 5pin 2,54 mm
 Q1: kwarc 4 MHz THT
 SW1, SW2: tact switch 6x6 mm
Dodatkowe materiały na FTP:
<ftp://ep.com.pl>, user: 66465, pass: td79fqh6
 • wzory płytek PCB
Projekty pokrewne na FTP:
 (wymienione artykuły są w całości dostępne na FTP)
 --- Barometr binarny (EP 8/2014)
 AVT-5426 Higrometr binarny (EP 12/2013)
 AVT-1698 Dwupunktowy termometr binarny (EP 8/2012)

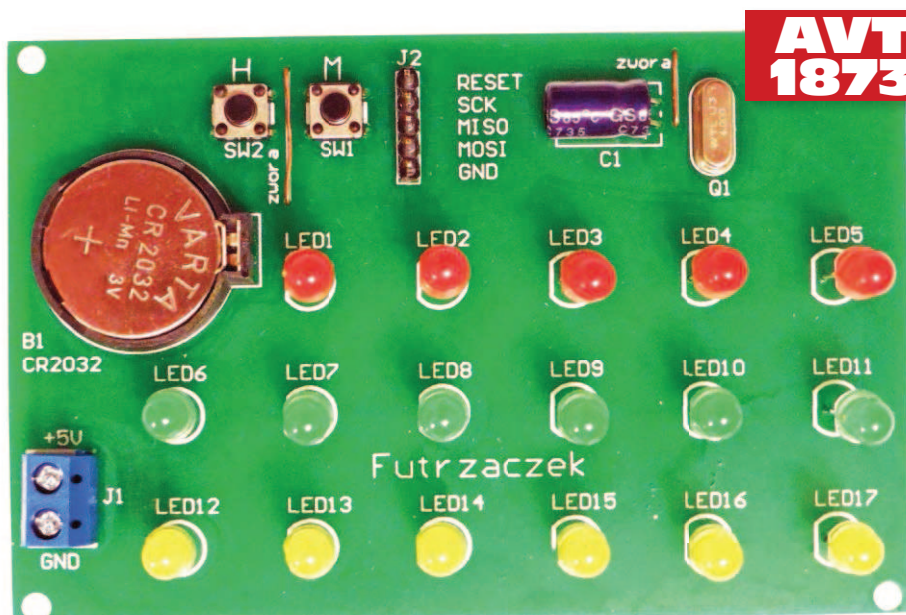
* Uwaga:
 Zestawy AVT mogą występować w następujących wersjach:
 AVT xxxx UK to zaprogramowany układ. Tylko i wyłącznie. Bez elementów dodatkowych.
 AVT xxxx A płytka drukowana PCB (lub płytki drukowane, jeśli w opisie wyraźnie zaznaczono), bez elementów dodatkowych.
 AVT xxxx A+ płytka drukowana i zaprogramowany układ (czyli połączenie wersji A i wersji UK) bez elementów dodatkowych.
 AVT xxxx B płytka drukowana (lub płytki) oraz komplet elementów wymienionych w załączniku pdf w złączniku pdf.
 AVT xxxx C to nic innego jak zmontowany zestaw B, czyli elementy wmontowane 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 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>

USBA. Kondensatory C1 i C2 odsprężają zasilanie, CE1, CE2 filtrują wyjściowe napięcie przetwornicy. Ze względu na dużą częstotliwość kluczkowania, dla zapewnienia odpowiedniej filtracji i stabilności układu, zostały zastosowane dwa kondensatory tantalowe o małym ESR. Dioda LD1 sygnalizuje załączenie napięcia wyjściowego +5 V. TPS6103x mają wbudowane dodatkowe obwody monitorowania zbyt niskiej wartości napięcia zasilania układu (LBI) oraz obwód zdalnego wyłączenia (EN), które w modelu nie są wykorzystane.

Układ zmontowano na niewielkiej, jednostronnej płytce drukowanej, której schemat montażowy pokazano na rysunku 2.

Montaż nie wymaga opisu. Po włożeniu baterii i załączeniu przetwornicy należy tylko skontrolować napięcie wyjściowe. W zależności od jakości baterii jest możliwe uzyskanie napięcia wyjściowego +5 V i prądu z zakresu 600...1500 mA już przy napięciu baterii rzędu 2...3 V. **Uwaga: bank energii jest przystosowany do zasilania jedynie z baterii AA. Niedopuszczalne jest używanie do zasilania akumulatorów Ni-Cd, Ni-MH i innych ze względu na brak zabezpieczenia przed ich nadmiernym rozładowaniem.**

Adam Tatuś, EP



Czy pamiętasz, drogi Czytelniku, panele sterujące pierwszych maszyn obliczeniowych? Dzisiaj można je zobaczyć wyłącznie na filmach, lecz nadal przyciągają wzrok. Długie rzędy żarówek, które migwały na pozór chaotycznie – pasjonatowi elektroniki nie może oderwać wzroku od tego widoku. Ten zegar, zwłaszcza w zestawieniu z resztą podobnych mu układów, stworzy w pokoju namiastkę wrażenia z dawnych lat.

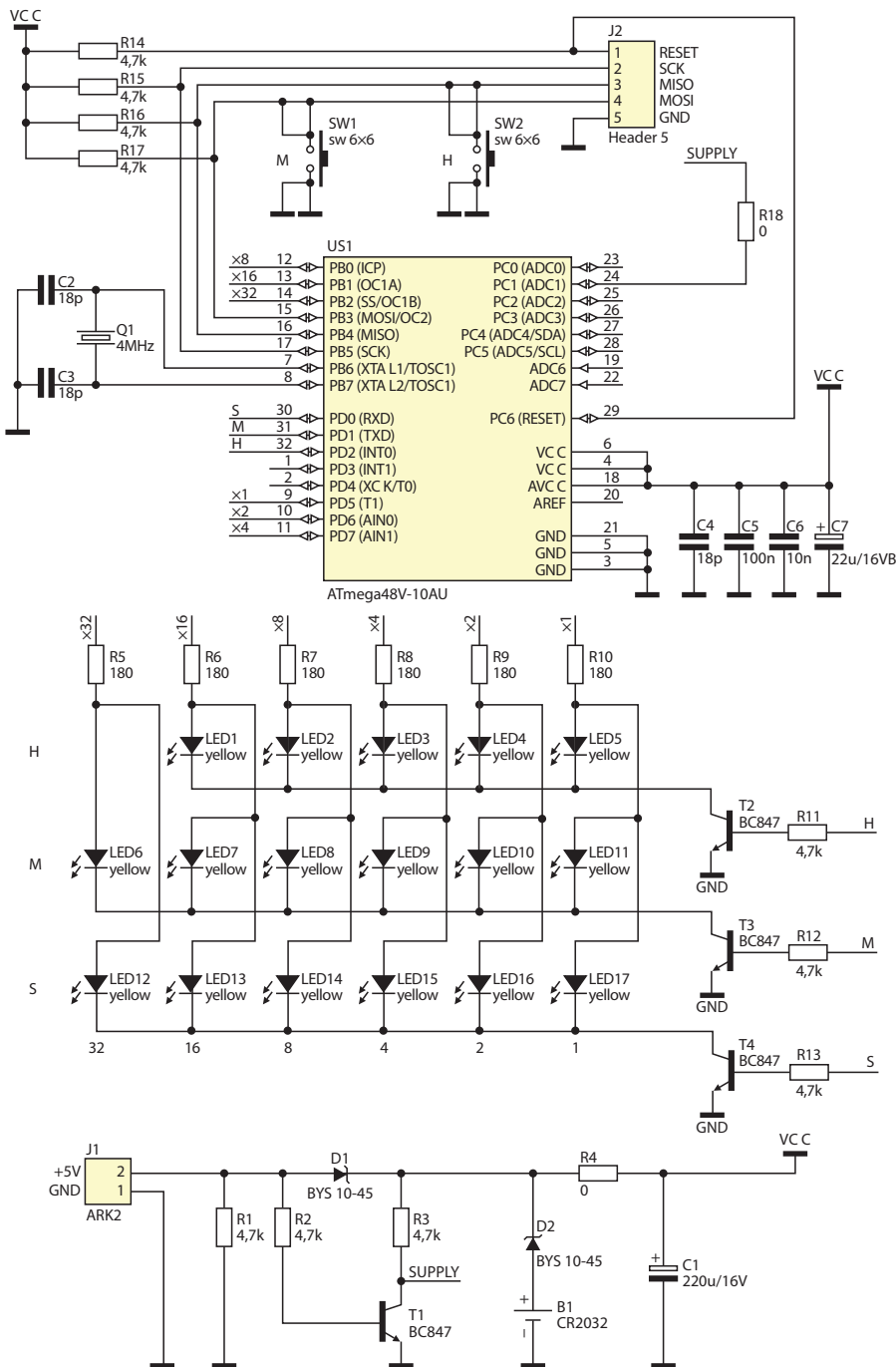
Schemat ideowy zegara binarnego pokazano na rysunku 1. Został on zrealizowany na mikrokontrolerze ATmega48. Nie ma tu zewnętrznego zegara czasu rzeczywistego (RTC), co obniża cenę urządzenia. Po zaniku zasilania, odliczanie jest podtrzymywane bateryjnie, a diody wyłączane.

Do wykrycia zasilania z zewnętrznego zasilacza służy obwód z tranzystorem T1. Jeżeli na zaciskach złącza J1 występuje napięcie, to otwiera się dioda D1, zatyka D2 i tranzystor wchodzi w stan nasycenia – wejście PC0 jest wyzerowane. Po zaniku zasilania, napięcie na kondensatorach filtrujących spada. Jednocześnie zatyka się D1 i otwiera D2. Zastosowane diody Schottky mają prąd

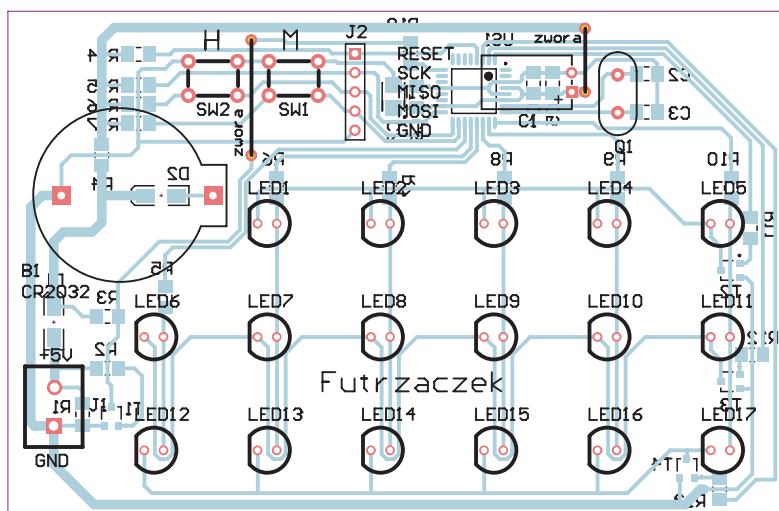
wsteczny na tyle duży, że jest on w stanie otworzyć wprowadzić T1 w stan nasycenia. Z tego powodu dodano rezystor R1, który odprowadza ten prąd do masy i nie pozwala na wystąpienie takiej sytuacji. Użycie diod Schottky było jednak konieczne ze względu na niski spadek napięcia w kierunku przewodzenia.

Wyświetlanie aktualnej liczby godzin, minut i sekund zostało zrealizowane na trzech liniijkach diod świecących. Sterowane są multipleksowo, co ogranicza pobór prądu oraz zmniejsza liczbę używanych wyprowadzeń mikrokontrolera. Wyświetlanie odbywa się wyłącznie w systemie 24-ro godzinnym. Do wyświetlenia godzin i minut potrzebne było 6 diod, a godzin 5 diod. W naturalnym kodzie binarnym, który został użyty do zakodowania wyświetlanych informacji, odpowiada to możliwym wskazaniom, odpowiednio, 0...63 i 0...31. Część sekwencji pozostanie niewykorzystana. Sposób odczytu zostanie opisany dalej.

Wykorzystany mikrokontroler może pracować przy napięciu zasilania nie niższym niż 1,8 V, co jest dużą przewagą



Rysunek 1. Schemat ideowy zegara binarnego



Rysunek 2. Schemat montażowy zegara binarnego

w porównaniu do starszej wersji tego układu, czyli ATmega8. Nawet niskonapięciowa wersja ATmega8L nie gwarantuje poprawnej pracy przy napięciu niższym niż 2,7 V, co przy zasilaniu napięciem 3 V jest niedopuszczalne. Ponadto, ATmega48 pobiera mniejszy prąd. Częstotliwość sygnału zegara jest stabilizowana rezonatorem kwarcowym 4 MHz, który jednocześnie stanowi wzorzec do odmierzania czasu.

Ustawianie aktualnej godziny i minuty odbywa się przyciskami, odpowiednio, SW2 i SW1. Sekundnik zeruje się po wciśnięciu któregośkolwiek z nich. Warto nadmienić, iż przyciski te są nieaktywne podczas pracy z baterii, aby zapobiec przypadkowemu przestawieniu godziny.

Zegar został zmontowany na jednostronnej płytce drukowanej o wymiarach 103 mm×67 mm, której schemat montażowy przedstawia rysunek 2. Szerokość jest dostosowana do pozostałych, wymienionych we wstępie, projektów „binarnych”. W pierwszej kolejności należy przylutować elementy montowane powierzchniowo, potem drwie zwory z drutu, a dalej elementy przewlekane, od najniższych poczynając. Na sam koniec należy przylutować diody, najlepiej na długich wyprowadzeniach – będą wtedy wystawały ponad pozostałe podzespoły. Nic nie stoi na przeszkodzie, aby zastosować inne kolory diod, lub całkiem je pomieszać, tworząc efekt całkowitego chaosu.

Przed zaprogramowaniem mikrokontrolera, należy zmienić bity zabezpieczające CKSEL na takie, które pozwolą na pracę z zewnętrznym rezonatorem kwarcowym 4 MHz oraz wyłączyć podział częstotliwości zegara przez 8 (bit ten nosi nazwę CKDIV8). Po poprawnym zaprogramowaniu, bezbłędnie zmontowany układ startuje od razu, rozpoczynając pracę od godziny 00:00:00.

Zasilanie układu odbywa się napięciem stałym, stabilizowanym o wartości +5 V. Pobór prądu jest zmienny i zależy od liczby włączonych aktualnie diod, ale nie przekracza 60 mA. Bateria typu CR2032 jest opcjonalna i niewymagana do poprawnej pracy układu – podtrzymuje ona jedynie odliczanie czasu po zaniku zasilania sieciowego. Pobór prądu z niej to ok. 1,5 mA. Przy pojemności baterii rzędu 200 mAh, powinna wystarczyć na ponad 5 dni pracy mikrokontrolera, co jest wystarczające w typowych sytuacjach.

Jak wspomniano, wyświetlanie czasu odbywa się w NKB (Naturalnym Kodzie Binarnym). Bity najstarsze znajdują się po lewej stronie, a najmłodsze po prawej. Celowo na płytce nie umieszczono opisów, aby osobom niewtajemniczonym trudniej było odgadnąć zasadę działania. Wbrew pozorom, po pewnym czasie, posługiwanie się NKB staje się tak samo intuicyjne, jak w przypadku systemu dziesiętnego, którym

Listing 1.

```

//#####
// Michał Kurzela @ 2015
// Zegar binarny
// ATmega48V-10AU, taktowanie kwarcem 4MHz + wyłączony CKDIV8
//#####

#include <avr/io.h>
#include <util/delay.h>
#include <avr/interrupt.h>
#define SW_M bit_is_clear(PINB,PINB3) // ustawianie minut
#define SW_H bit_is_clear(PINB,PINB4) //ustawianie godzin
#define SUPPLY bit_is_clear(PINC,PINC1) //sprawdzanie, czy mamy zasilanie, TRUE jeśli tak

//#####
//#####
//#####

//deklaracja zm ienych globalnych
volatile uint8_t godziny = 0;
volatile uint8_t minuty = 0;
volatile uint8_t sekundy = 0; //zmienne „czasowe”
volatile uint8_t wiersz = 0; //aktualnie uruchomiony wiersz
//0 - sekundy, 1 - minuty, 2 - godziny
volatile uint8_t licznik = 0; //zmienna licząca przerwania od Timer0 - zlicza ich 250
volatile uint8_t supply = 0; //0 - jest zasilanie, 1 - brak zasilania

//#####
//#####
//#####

void podziel_liczbe(uint8_t liczba){ //najpierw załączamy diody, a potem cały wiersz
PORTD &= 0b11111000; //wyłączenie wiersza, aby nie migał podczas ustawiania diod

//porównywanie z wagami bitów i ewentualne ich odejmowanie, aż do zera
if(liczba >= 32){ PORTB |= 0b00000100; liczba -= 32;} else { PORTB &= 0b11111011;}
if(liczba >= 16){ PORTB |= 0b00000010; liczba -= 16;} else { PORTB &= 0b11111101;}
if(liczba >= 8){ PORTB |= 0b00000001; liczba -= 8;} else { PORTB &= 0b11111110;}
if(liczba >= 4){ PORTD |= 0b10000000; liczba -= 4;} else { PORTD &= 0b01111111;}
if(liczba >= 2){ PORTD |= 0b01000000; liczba -= 2;} else { PORTD &= 0b10111111;}
if(liczba >= 1){ PORTD |= 0b00100000; liczba -= 1;} else { PORTD &= 0b11011111;}

//zostało to wykonane instrukcjami IF aby było bardziej przejrzyste
if(wiersz == 0){PORTD |= 0b00000001; } //załączony wiersz sekund
if(wiersz == 1){PORTD |= 0b00000010; } //załączony wiersz minut
if(wiersz == 2){PORTD |= 0b00000100; } //załączony wiersz godzin
}

//#####
//#####
//#####

ISR(TIMER0_OVF_vect){
TCNT0 = 6; //odliczanie 250
licznik++;
if(licznik >= 250){licznik = 0; sekundy++;}
if(sekundy >= 60){sekundy = 0; minuty++;}
if(minuty >= 60){minuty = 0; godziny++;}
if(godziny >= 24){godziny = 0;}

if(supply == 0){ //kiedy jest zasilanie, to wyświetlaj
wiersz++; //inkrementacja w każdym przerwaniu
if(wiersz >= 3){ wiersz = 0;}

if(wiersz == 0){ podziel_liczbe(sekundy); } //wyświetlenie jednej wartości
if(wiersz == 1){ podziel_liczbe(minuty); }
if(wiersz == 2){ podziel_liczbe(godziny); }}

if(supply == 1){ //kiedy nie ma zasilania
PORTD &= 0; //wyłącz wiersze i kolumny
PORTB &= 0b00011000;
}}

//#####
//#####
//#####

int main (void){
//sprawdzanie obecności zasilania
//aby wyświetlanie w przerwaniu się przypadkowo nie uaktywniło
if(SUPPLY){supply = 0;} else {supply = 1;}

DDRB = 0b00000111; //konfiguracja we/wy
DDRC = 0b11111101;
DDRD = 0b11111111;

PORTB = 0b00011000;
PORTC = 255;
PORTD = 0;

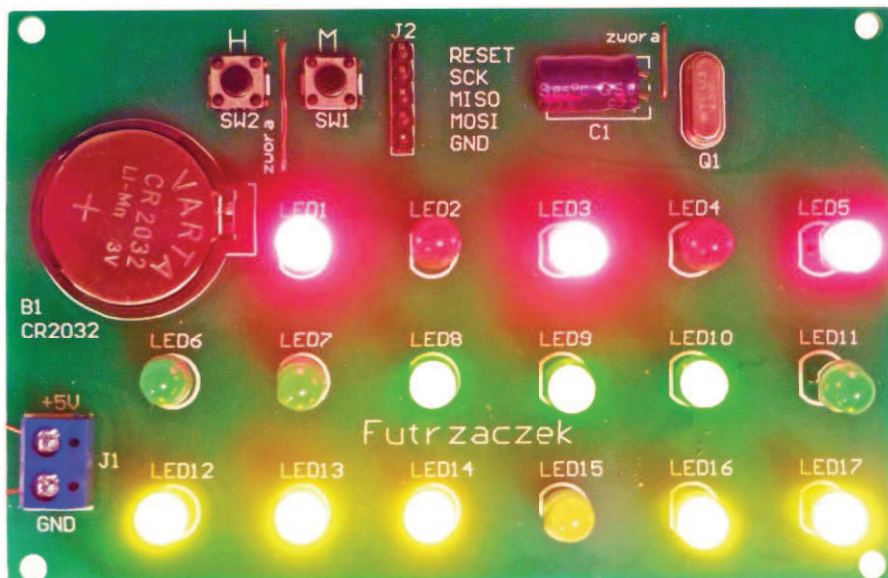
TCCR0A = 0;
TCCR0B = 0b00000011; //przeskaler przez 64
TCNT0 = 6; //250 taktów dla częstotliwość przerwań 250Hz
TIMSK0 = 1; //zezwoleń na przerwania od przepełnienia
TIFR0 = 1;
sei(); //aktywacja systemu przerwań

while(1){

if(SUPPLY){supply = 0;} else {supply = 1;} //sprawdzanie obecności zasilania

if(supply == 0){ //nastawy czasu tylko przy zasilaniu
if(SW_M){minuty++; sekundy = 0; licznik = 0; if(minuty >= 60){minuty = 0;} _delay_ms(120);}
if(SW_H){godziny++; sekundy = 0; licznik = 0; if(godziny >= 24){godziny = 0;} _delay_ms(300);}
}}
}

```

Fotografia 3. Wskazywana godzina to 21:14:59

posługujemy się na co dzień. Przykładową sytuację prezentuje **fotografia 3**, zaś

rysunek 4 ilustruje zasadę odczytu. Górny wiersz to godzina, środkowy minuta, a dolny

H		●	○	●	○	●
M	○	○	●	●	●	○
S	●	●	●	○	●	●
	x32	x16	x8	x4	x2	x1

Rysunek 4. Sposób odczytu

sekunda. Jeżeli dioda świeci się, należy wagę kolumny pomnożyć przez 1, a jeśli jest wygaszona, to przez 0. Na końcu poszczególne wartości kolumn w wierszu sumuje się.

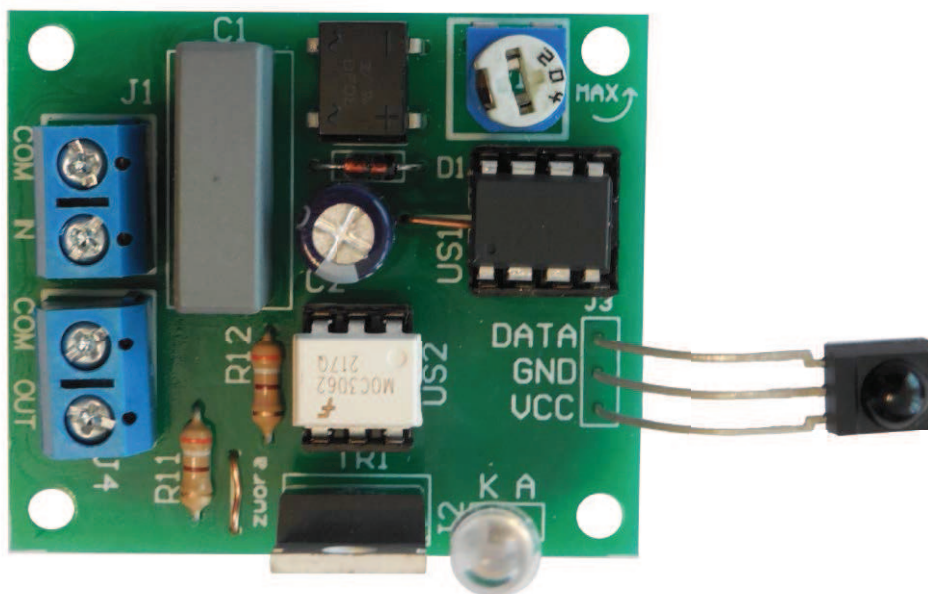
Dla dociekliwych

Na **listingu 1** znajduje się kod źródłowy programu, który został napisany w języku C. Odmierzanie czasu odbywa się w przerwanach od Timer0, który przepelnia się 250 razy na sekundę. Przy każdym wywołaniu przerwania załączany jest jednocześnie jeden wiersz. Częstotliwość odświeżania wynosi około 80 Hz, więc migotanie diod jest dla ludzkiego oka niewidoczne.

Zbliżeniowy włącznik czasowy

Popularne bariery podczerwieni reagują przy przecięciu wiązki światła pomiędzy nadajnikiem a odbiornikiem. Jednak w pewnych sytuacjach, montaż dwóch modułów naprzeciw siebie jest utrudnione lub wręcz niemożliwy. Przedstawione urządzenie eliminuje ten problem. Przyda się szczególnie w miejscach, w których załączenie oświetlenia lub wentylacji jest potrzebne na niedługi czas.

Układ włącznika jest przystosowany do zasilania napięciem 230 V AC. Działa na zasadzie wykrycia wiązki podczerwieni odbitej od obiektu. Schemat urządzenia jest przedstawiony na **rysunku 1**. Za funkcjonalność włącznika jest odpowiedzialny mikrokontroler ATtiny13. Okresowo, co 10 ms, generuje 30 impulsów załączających diodę IR. Impulsy te mają częstotliwość ok. 36 kHz i wypełnienie 2%, dzięki czemu zużycie energii jest niewielkie. Kondensator C5 poprawia szybkość zmian napięcia na diodzie, w szczególności przy wychodzeniu tranzystora T1 ze stanu nasycenia. Jeżeli część wyemitowanego światła odbije się od znajdującego się niedaleko obiektu, scalony odbiornik TSOP4836 poinformuje o tym zerując swoje wyjście. W tym momencie następuje



odczytanie położenia suwaka potencjometru (poprzez sprawdzenie zawartości rejestrów przetwornika A/C) i załączenie diody w optotriaku. Czas załączenia jest regulowany z krokiem jednej sekundy, w zakresie 10...1033 s, czyli do ok. 17 minut.

Rolą tego ostatniego nie jest tutaj separacja galwaniczna, ponieważ cały układ i tak jest zasilany beztransformatorowo z sieci, lecz poprawne załączenie elementu wykonawczego. Tym elementem wykonawczym jest triak BT137, choć można zastosować inny. Jako optotriak wybrano MOC3063, ponieważ wymaga on najmniejszego (z całej

rodziny) prądu diody świecącej, wynoszącego 5 mA. Testy wykazały, że MOC3062 (wymagający 10 mA) również działa poprawnie. Zasilacz beztransformatorowy został zaprojektowany tak, aby dostarczał napięcie ok. 5 V niezbędne do poprawnego działania mikrokontrolera. Rezystory R1 i R2 rozładowują kondensator C1 po wyłączeniu zasilania, zaś R3...R5 ograniczają udar prądowy w momencie włączenia zasilania. Połączenie ich szeregowo zwiększa wytrzymałość napięciową. Dodatkowo, w razie przebicia kondensatora C1, któryś z rezystorów R3...R5 zadziała jak bezpiecznik i przerwie obwód.

**AVT
1874**