



Zegarek nareczny z wyświetlaczem binarno-widmowym



Elektronika to hobby, które daje dużo satysfakcji. Satysfakcja ta jest oczywiście większa, jeśli możemy się komuś pochwalić zbudowaną przez siebie ciekawą i oryginalną konstrukcją. Urządzenie, które pragnę zaprezentować, świetnie nadaje się do tego celu. Można je mieć zawsze przy sobie, aby w stosownym momencie mimochodem zaprezentować, wzbudzając przy tym nieuchronnie zainteresowanie. Bo jest to samodzielnie wykonany elektroniczny zegarek nareczny.

Wyświetlacz

Kiedyś każdy zegarek był droгим, eleganckim dziełem sztuki jubilerskiej, ale również prezentacją sztuki mechaniki precyzyjnej. Zegarek był raczej niezawodny. W razie awarii oddawało się go do zakładu zegarmistrzowskiego, gdzie potrafiąco za pomocą precyzyjnych narzędzi i przy użyciu lupy na nowo tchnąć weń życie. Dzisiaj większość sprzedawanych zegarków to tanie i niestety dość szybko psujące się urządzenia. Takich tanich czasomierzy nikt nie naprawia. Ale jeśli posiadasz zepsuty zegarek, możesz pokusić się o jego reaktywację w nowej formie. I niczym zegarmistrz, za pomocą lupy i innych precyzyjnych narzędzi na nowo tchnąć weń życie.

Tradycyjne zegarki miały wskazówki, później pojawiły się wyświetlacze numeryczne – LED-owe i ciekłokrystaliczne. Ostatnio modne stają się konstrukcje z wyświetlaczami binarnymi oraz widmowe. A taki właśnie wyświetlacz – zarówno binarny, jak i widmowy – będzie miał nasz zegarek.

Jak powszechnie wiadomo, „jest tylko 10 rodzajów ludzi – ci, którzy rozumieją system dwójkowy i ci, którzy nie rozumieją”. Zrozumienie systemu binarnego

(dwójkowego) jest potrzebne, aby umieć odczytać czas z wyświetlacza naszego zegarka. Jako że system dwójkowy jest podstawą działania wszystkich cyfrowych układów elektronicznych, a w szczególności tych opartych na mikrokontrolerach, warto sobie tę wiedzę przypomnieć. Zatem dlaczego system dwójkowy? Bo są w nim tylko dwie cyfry: jeden i... zero oczywiście. Wszystkie liczby przedstawione są jako kombinacja tych dwóch cyfr – więc po kolei licząc, będzie to 0, 1, 10, 11, 100, 101 ... itd. (0, 1, 2, 3, 4, 5, dziesięć), jak pokazuje tabela 1.

Liczbę z systemu dwójkowego można łatwo przeliczać na dziesiętny, sumując wartości dla ustawionych bitów, np. liczba dwójkowa 00010101 to dziesięć 16 + 4 + 1 = 21.

I właśnie dwójkowy sposób odczytu wykorzystujemy w naszym zegarku. Na jego tarczy znajdują się dwa rzędy diod LED. Górny, mający cztery diody, wyświetla godziny: 8 + 2 + 1 = 11, minut: 32 + 16 + 4 + 1 = 53.

(godzin: 8 + 2 + 1 = 11, minut: 32 + 16 + 4 + 1 = 53).

Zdecydowanie bardziej spektakularny jest widmowy tryb wyświetlania. Aby w tym trybie odczytać czas, należy poruszyć zegarkiem. Napis przedstawiający aktualny czas w formacie HH:MM wyświetli się w powietrzu (fotografia 2).

Zatem wiemy już, że czas może być wyświetlany na dwa sposoby.

Opis układu

Schemat zegarka jest zaprezentowany na rysunku 1. Diody LED sterowane są poprzez rezystory ograniczające prąd bezpośrednio z wyprowadzeń mikrokontrolera. Właśnie mikrokontroler ATmega8 w obudowie TQFP jest głównym elementem systemu. Dlatego też zajmuje on centralne miejsce na okrągłej tarczy naszego zegarka. Oprócz mikrokontrolera i diod z rezystorami, z przodu płytki

znajduje się jeszcze kondensator odsprzęgający 100nF. Pozostałe 3 elementy znajdują się z drugiej strony płytki (fotografia 3). A są to: 3-woltowa bateria CR2032, mikroprzełącznik oraz element, który występuje w każdym zegarku elektronicznym – tzw. kwarc zegarkowy 32kHz w swojej charakterystycznej rurkowej obudowie. Impulsy od kwarcu zliczane są w trybie asynchronicznym, a procesor w swoich krótkich chwilach aktywności jest taktowany wewnętrznym

Tabela 1

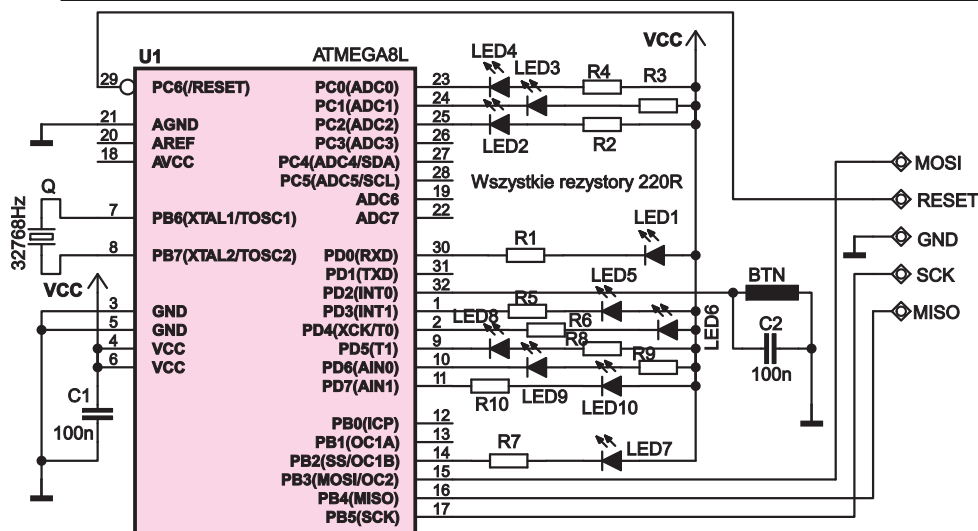
Liczba dziesiętna	Liczba dwójkowa	Zapis potęgowy
1	00000001	2 ⁰
2	00000010	2 ¹
4	00000100	2 ²
8	00001000	2 ³
16	00010000	2 ⁴
32	00100000	2 ⁵
itd.		



Fot. 1 Zegarek z przodu

Fot. 2 Wyświetlanie widmowe





Rys. 1 Schemat ideowy

Fot. 3 Wnętrze – pod tylną pokrywą



generatorem 1MHz. Więcej o działaniu zegarka w części Oprogramowanie.

Montaż i uruchomienie

Montaż należy rozpocząć od przygotowania obudowy – koperty starego zegarka. Nie każda koperta się nada. Średnica wewnętrzna musi wynosić co najmniej 28mm, ale nie więcej niż 35mm, bo wtedy zegarek nie wygląda już atrakcyjnie. Ponadto koperta musi być odpowiednio gruba, aby zmieściła się w niej płytkę z przylutowanymi częściami i uchwyt z baterijką CR2032 na plecach. Do koperty musimy zamontować ruchomo przycisk (gałeczkę – pokrętko). Można to zrobić na kilka sposobów. Na obciążoną ośkę z gałką można od środka przylutować kropelkę cyny lub nakleić tulejkę (fotografia 4). Można też zamiast osi użyć małej śrubki, którą wkładamy od środka, a gałkę dokleić z zewnątrz klejem pociopól lub distal.

Płytkę drukowaną trzeba dopasować do koperty. Najpierw należy zgrubnie obróbić ją na szlifierce, a następnie ręcznie oszlifować papierem ściernym. Laminat powinien wchodzić do koperty na lekki wcisk. Dla ułatwienia szlifowania, płytkę ma oznakowane trzy współśrodkowe okręgi o średnicach 28mm, 30mm i 33mm (zobacz płytkę na rysunku 2). W razie potrzeby można wykonać lekkie nacięcia dla ułatwienia przełożenia przycisku. Po dopasowaniu do koperty i powierceniu otworów, przystępujemy do montażu elektroniki.

Fot. 4 Sposób zamontowania gałeczki przycisku



Najpierw należy starannie pocynować całą płytkę. Najprościej zrobić to, cienko rozcierając cynę gorącym grottem lutowniczym. Płytkę należy pokryć przednio preparatem RF800, a cyny używać naprawdę niewiele. Pocynowane ścieżki będą trwałe i estetyczne. Następnie wlotujemy zworki – styk (-) baterii. Najlepiej wykonać je z odciętych nóżek rezystora (jak najcieńsze). Zworki te wlotować jak najbardziej płasko przy samym laminacie. Następnie lutujemy uchwyt baterii, wykonany ze srebrzanki o średnicy 0,8–1mm. Jeśli nie mamy srebrzanki, możemy użyć odciętych wyprowadzeń diody prostowniczej 1A. Uchwyt baterii powinien być wygięty w kształt ostrej litery C tak, aby nie zwiększać i tak już sporej grubości urządzenia (fotografia 5). Taką kłamrę należy wlotować, podkładając pod spód zużytą baterię. Bateria powinna dać się wysuwać, ale jednocześnie wchodzić dość ciasno. Jej pewne zamocowanie to gwarancja stabilnej pracy zegarka. Wystające końce kłamry przed lutowaniem należy zagiąć płasko tak, aby zabezpieczyć uchwyt przed wyrwaniem. Teraz wracamy na frontową stronę. Niestety płytkę z wystającym od spodu pałąkiem nie jest zbyt wygodna do dalszych prac. Jest na to rada: w kawałku deseczki robimy brzeszczotem nacięcie, do którego wciskamy wystającą z płytki kłamrę baterii. Taka deseczka stanowi stabilną i wygodną podstawkę do dalszych prac.

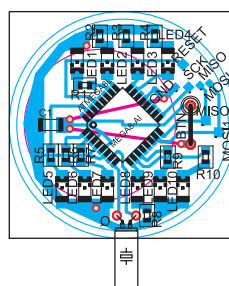
Do operacji lutowania SMD należy przygotować się solidnie: cienki grot (0,4mm), cienka cyna, lupa na wysięgniku, specjalna pęseta i plecionka do

odciągania nadmiaru cyny. Na pady pod układ scalony nakładamy kropelki cyny, a sam układ scalony lutujemy, kolejno dociskając wszystkie wyprowadzenia i nakładając na nie jeszcze po kropelce cyny. Oczywiście zanim przylutujemy wszystkie wyprowadzenia, dokładnie pozycjonujemy chip – późniejsza korekta jego położenia jest praktycznie niemożliwa. Następnie lutujemy rezystorki (trzeba uważać, aby nie odfrunęły ☹), a na koniec kondensator i diody LED. Diody należy uważnie pozycjonować – od ich równego ustawienia będzie zależeć późniejszy efekt przy wyświetlaniu. Nie należy używać zbyt dużo lutowni, aby nie porobić zwarców – dlatego niezbędna jest cienka cyna, którą łatwiej dozować. Po sprawdzeniu połączeń musimy przylutować jeszcze kwarc zegarkowy 32kHz oraz mikroprzełącznik. Te elementy montujemy tradycyjnym sposobem przewlekającym. Sam mikroprzełącznik musimy przygotować, wyginając jego wyprowadzenia. Jeśli nie mamy wersji przewlekanej, nada się też wersja SMD, ale wcześniej trzeba dolutować wyprowadzenia (wykonane z nóżek rezystora). Przełącznik najlepiej wlotować, pasując go do przycisku obudowy. Całość powinna pracować lekko i pewnie. Płytkę po zakończonym lutowaniu jest widoczna na fotografii 6.

Następnie należy zaprogramować procesor i ewentualnie usunąć błędy montażu, które się pojawiają.

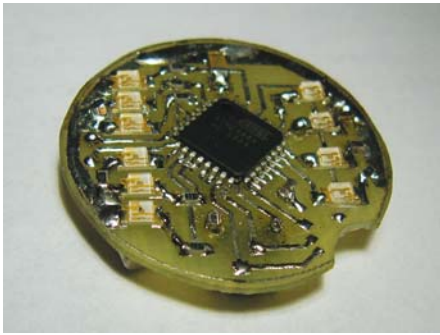
Po uruchomieniu zegarka pozostaje jego wykończenie. Musimy wykonać cyferblat. Element ten odgrywa istotną rolę – ułatwia odczyt czasu, dodaje profesjonalnego wyglądu, a posiadając dość duże okienko, gustownie eksponuje elementy elektroniczne. Pełni również funkcję izolatora, oddzielając obwód od metalowej obudowy. Papierowy cyferblat możemy wykonać sami, umieszczając na nim np. dedykację. Można też posłużyć się jednym z gotowych wzorów dostępnych w materiałach dodatkowych w Elportalu ([cyferblat.pdf](#)). Wybrany wzór drukujemy na drukarce (najlepiej laserowej, ale niekoniecznie). Następnie żyłką, skalpelem lub ostrymi nożyczkami starannie wycinamy okienko. Krawędź papieru po wycięciu możemy zamaskować czarnym flamastrem. Tak spreparowany cyferblat przyklejamy do płytki zegarka. Ja użyłem kleju magik, ale może to być też wikol. Po przyklejeniu musimy poczekać kilka godzin, aż

Rys. 2 Płytkę drukowaną



Fot. 5 Zmontowana płytkę od tyłu – mocowanie baterii





Fot. 6 Zmontowana płytka od przodu

klej zaschnie. Na czas klejenia wyciągamy baterię, gdyż mokry klej przewodzi prąd. Po zaschnięciu kleju obcinamy z zewnątrz nadmiar papieru, montujemy baterię i wkładamy całość do koperty. Zamykamy wieczko i podziwiamy efekt ☺.

Kolejne fazy montażu możemy obejrzeć na fotografii 7.

Oprogramowanie

Po zakończonym montażu wypadaloby zaprogramować procesor. Oczywiście na tak małej płytce nie ma miejsca na tradycyjne złącze programatora. Można było na czas programowania dolutować się do odpowiednich wyprowadzeń procesora, jak na fotografii 8. Ja chciałem mieć możliwość łatwego poprawiania programu w trakcie użytkowania zegarka. Zastosowałem patent podpatrzony w Internecie – małą taśmę elastyczną. Wykorzystałem tasiemkę wymontowaną z oryginalnym gniazdem z starej komputerowej stacji dyskieta 3,5 cala. Tasiemka służyła tam do połączenia głowicy odczytującej. Tasiemka taka jest dość długa. Dlatego uciąłem stosowny odcinek z częścią wtykową. Drugi koniec musiałem „zarobić” tak, aby można go było przylutować. Najpierw usunąłem lakier drobnym papierem ściernym, następnie pocynowałem odkryte miedziane ścieżki – topnik RT800 znów się przydał. Na koniec odpowiednio porozcinałem tasiemkę żyłką tak, aby pojedyncze ścieżki można było przylutować do odpowiednich padów. Aby taką tasiemką połączyć z programatorem, przygotowałem specjalną płytkę-prześciówkę zaopatrzoną w dwa gniazda: tasiemki oraz tradycyjne 10-pinowe gniazdo programowania ISP. Płytkę prototypu zegarka wraz z prześciówką do programowania jest widoczna na fotografii 9. Schemat i płytkę adaptera jest przedstawiony na rysunkach 3 i 4



Fot. 7 Kolejne fazy montażu

Program dla mikrokontrolera został napisany w języku C (środowisko AVR-GCC). Jego uproszczona, ale działająca wersja (uclock_short.c) została zamieszczona na listingu 1 i 2.

Procesor jest taktowany wewnętrznym generatorem 1MHz. W funkcji *main()* następuje inicjalizacja rejestrów mikrokontrolera, w tym rejestrów timera 2, umożliwiającego jego pracę jako asynchronicznego licznika. Będzie on zliczał impulsy z zewnętrznego kwarcu 32768Hz, a dzięki zastosowaniu preskalera o wartości 256, co 2 sekundy będzie wywoływane przerwanie.

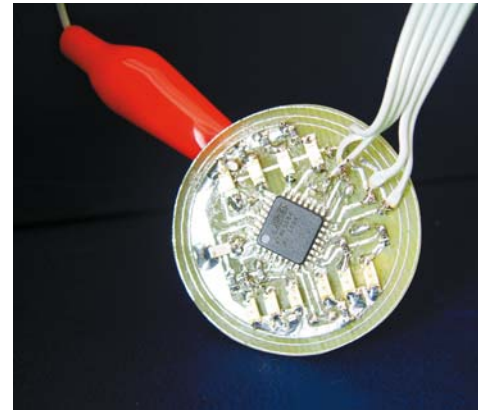
Druga część funkcji *main()* to nieskończona pętla *for(;;)*, w której następuje sprawdzenie, czy przycisk jest naciśnięty. Jeśli nie jest, to wyświetlacz jest wygaszany, a procesor przechodzi w oszczędzający energię tryb uśpienia *set_sleep_mode(SLEEP_MODE_PWR_SAVE)*. Z uśpienia procesor jest wybudzany jedynie przez przerwanie:

- *SIG_OVERFLOW2* od wspomnianego asynchronicznego timera 2 – w funkcji jego obsługi następuje zliczanie czasu.

- *SIG_INTERRUPT0* wywoływane naciśnięciem przycisku. Wtedy funkcja obsługi wyświetla bieżący czas w formie binarnej, wywołując funkcję *ShowDisplay()*.

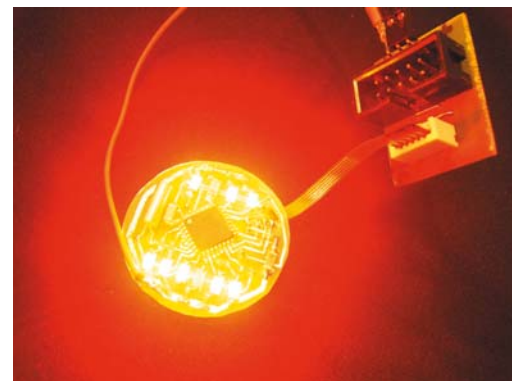
Oczywiście właściwy program jest dużo bardziej złożony. Dochodzi w nastawianiu czasu, korekcji czasu, wyświetlanie widmowe czy tryb wzbudzenia uwagi. Kompletny kod źródłowy jest dostępny w materiałach dodatkowych w Elportalu (*uclock.c*). Za wyświetlanie widmowe odpowiada funkcja *ShowDisplayPov()*, a wzory cyfr są zapisane w pamięci programu *prog_uint8_t g_Digit[41]*. Do generowania opóźnień potrzebnych w trybie wyświetlania widmowego oraz obsłudze przycisku wykorzystany został timer 0.

Dla osób niechących zaprzątać sobie głowy programem w języku C, dostępny jest gotowy wsad (*uclock.hex*). Przy progra-



Fot. 8 Wlutowane przewody programowania

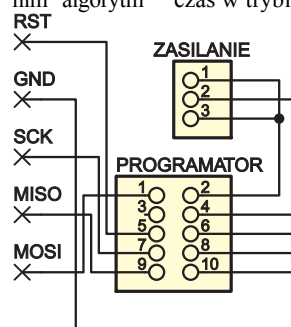
Fot. 9 Tasiemka do programowania wraz z prześciówką



owaniu nie należy zapomnieć o właściwym ustawieniu fuse-bitów – zwłaszcza *CKOPT* – odpowiedzialnego za dołączenie wewnętrznych kondensatorów na wyjściach kwarcu (*XTAL*). Właściwe ustawienie fuse-bitów prezentuje rysunek 5.

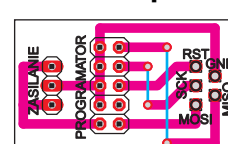
Obsługa

Zegarek przez większość czasu pozostaje uśpiony. Pojedyncze naciśnięcie i zwolnienie przycisku w trybie uśpionym powoduje, że zegarek przez około 4 sekundy wyświetla czas w trybie widmowym. Aby coś zobaczyć, należy w trakcie wyświetlania przesunąć zegarek. W praktyce wystarczy machnąć ręką (nie za szybko, aby kogoś nie uderzyć). Mrugający napis przedstawiający aktualny czas w formacie HH:MM dosłownie pojawi się w powietrzu, podążając za ruchem nadgarstka. Korzystanie z tego typu wyświetlania wymaga nieco wprawy, ale jest szalenie efektowne. Naciśnięcie i przytrzymanie przycisku w trybie uśpienia wyświetla czas w trybie binarnym opisanym wcześniej. Aby wejść w tryb ustawień, należy przycisk nacisnąć 3 razy. Zacznie wtedy migać górna część wyświetlacza, odpowiedzialna za wskaza-



Rys. 3 Schemat adaptera

Rys. 4 Płytkę adaptera



nia godzin. Teraz naciskając krótko przycisk, można zmieniać w pętli wyświetlaną godzinę (1,2,3...12). Naciskając i przytrzymując przycisk dłużej, przechodzi się w tryb ustawiania dziesiątek minut. W tym trybie, na górnej części wyświetlacza migają dwie diody z lewej strony oraz dolna część wyświetlacza z bieżącym ustawieniem dziesiątek minut. Naciskając krótko, można zmieniać ustawienie (0,10,20...50). Przyciskając dłużej, przechodzi się w tryb ustawiania jednostek minut – migają dwie prawe diody z górnej części, a na dolnej ustawiana wartość minut (0,1,2...9).

```
#include <avr\io.h>
#include <avr\interrupt.h>
#include <avr\sleep.h>
#include <util\delay.h>
// Definicje wyprowadzeń portu C
#define H01 0
#define H02 1
#define H04 2
#define H08 3
// Definicje wyprowadzeń portu D
#define KEY 2
#define M32 1
#define M16 3
#define M08 4
#define M04 5
#define M02 7
#define M01 6
// typ czas
typedef struct
{
    uint8_t Sec;
    uint8_t Min;
    uint8_t Hour;
} Time;
// czas
volatile Time g_CurrTime;
// wyczyść wyświetlacz
void ClearDisplay(void)
{
    PORTC = _BV(H01) | _BV(H02) | _BV(H04) | _BV(H08);
    PORTD = _BV(M01) | _BV(M02) | _BV(M04) | _BV(M08) |
            _BV(M16) | _BV(M32) | _BV(KEY);
}
//funkcja main
int main(void)
{
    //konfiguracja portów - wyjść
    DDRC = _BV(H01) | _BV(H02) | _BV(H04) | _BV(H08);
    DDRD = _BV(M01) | _BV(M02) | _BV(M04) | _BV(M08) | _BV(M16) | _BV(M32);
    //wyłączenie komparatora analogowego (oszczędność prądu)
    ACSR = _BV(ACD);
    //ustawienie timera 2 do obsługi zliczania czasu z zewnętrznego kwarcu
    w trybie asynchronicznym
    TIMSK = _BV(TOIE2); //Obsługa T/C Overflow Interrupt Enable
    ASSR = _BV(AS2);
    TCNT2 = 0x00;
    TCCR2 = _BV(CS22) | _BV(CS21); //prescaler/256 -2sek
    //umożliwienie usypiania
    MCUCR = _BV(SE);
    //czas początkowy
    g_CurrTime.Sec=0;
    g_CurrTime.Min=50;
    g_CurrTime.Hour=9;
    ClearDisplay();
    //globalne zezwolenie na przerwanie
    sei();
    for(;;)
    {
        //obsługa klawisza
        if(!(PIND & _BV(KEY))) //czy wcisnięty
        {
            delay_ms(10);
            //nadał wcisnięty
            while (!(PIND & _BV(KEY)))
            {
                _delay_ms(10);
            }
        }
        //przygotowanie do usypienia
        //ustawienie obsługi przerwania Int0...
        GICR = _BV(INT0);
        //zgaszenie wyświetlacza
        ClearDisplay();
        //przejdźcie w usypienie
        set_sleep_mode(SLEEP_MODE_PWR_SAVE);
        sleep_mode();
    }
    return 0;
}
```

```
// Ustaw wyświetlacz
void ShowDisplay(void)
{
    uint8_t h08,h04,h02,h01;
    uint8_t m32,m16,m08,m04,m02,m01;
    h08 = ((g_CurrTime.Hour & (1<<3))>>3);
    h04 = ((g_CurrTime.Hour & (1<<2))>>2);
    h02 = ((g_CurrTime.Hour & (1<<1))>>1);
    h01 = ((g_CurrTime.Hour & (1<<0))>>0);
    m32 = ((g_CurrTime.Min & (1<<5))>>5);
    m16 = ((g_CurrTime.Min & (1<<4))>>4);
    m08 = ((g_CurrTime.Min & (1<<3))>>3);
    m04 = ((g_CurrTime.Min & (1<<2))>>2);
    m02 = ((g_CurrTime.Min & (1<<1))>>1);
    m01 = ((g_CurrTime.Min & (1<<0))>>0);
    PORTC = ~(h01*_BV(H01) | h02*_BV(H02) |
              h04*_BV(H04) | h08*_BV(H08));
    PORTD = ~(m01*_BV(M01) | m02*_BV(M02) |
              m04*_BV(M04) | m08*_BV(M08) |
              m16*_BV(M16) | m32*_BV(M32));
}
// Obsługa przycisku
SIGNAL(SIG_INTERRUPT0)
{
    //wyłączenie obsługi przerwania INTO
    GICR = 0;
    //wyswietlenie czasu
    ShowDisplay();
}
// Obsługa upływu czasu
SIGNAL(SIG_OVERFLOW2)
{
    //inkrementacja bieżącego czasu
    g_CurrTime.Sec+=2;
    if(g_CurrTime.Sec>=60)
    {
        g_CurrTime.Sec=g_CurrTime.Sec-60;
        g_CurrTime.Min++;
    }
    if(g_CurrTime.Min>=60)
    {
        g_CurrTime.Min=g_CurrTime.Min-60;
        g_CurrTime.Hour++;
    }
    if(g_CurrTime.Hour>12)
    {
        g_CurrTime.Hour=g_CurrTime.Hour-12;
    }
}

```

Listing 1

Listing 2

Zapamiętanie nowego czasu następuje po dłuższym przytrzymaniu przycisku. Zegarek potwierdza nowy czas, wyświetlając go przez 4 sekundy. Jeśli w trybie ustawiania czasu nic nie zostanie naciśnięte przez 5 sekund, zegarek porzuca zmiany i przechodzi w tryb usypiania. W materiałach dodatkowych dostępna jest szczegółowa instrukcja obsługi (*instrukcja.pdf*). Została ona przygotowana w taki sposób, aby po obustronnym wydruku i złożeniu na pół, powstała otwie-

rana broszurka. Instrukcja będzie niezbędna, jeśli zechcemy komuś nasz zegarek podarować.

Jasność świecenia diod została tak dobrana, aby możliwy był odczyt czasu nawet w słoneczny dzień na zewnątrz. Ale także odczyt w kompletnej ciemności nie sprawia problemu. Światło diod odbija się od wewnętrznej powierzchni szybki i gustomiennie oświetla napisy na cyferblacie oraz elementy elektroniczne wnętrza.

W zegarku zastosowano baterię CR2032. Jest to standardowa, łatwo dostępna baterijka o znamionowej pojemności 200mAh. Zegarek w trybie usypionym pobiera ok. 7µA, co wystarczyłoby na ponad 3 lata pracy zegarka. Niestety każde wyświetlanie czasu skracza znacznie żywotność baterii – 1 sekunda odczytu czasu to około 35 minut pracy w trybie czuwania. Przy standardowym użyciu bateria powinna wystarczyć na około pół roku.

Propozycje zmian

Zegarek jest dość wdzięcznym tematem do wszelkich udoskonalień. Najłatwiej zrobić zmiany w samym oprogramowaniu, dodając np. funkcję stroboskopu dyskotekowego lub

wyświetlania różnych napisów w trybie widmowym. Ja z założenia zrobiłem jak najprostszą konstrukcję, ale zmieniając lub projektując od nowa płytkę można zegarek znacznie rozbudować np. dodając membranę piezoelektro (budzik) czy fotoelement sterujący jasnością diod LED w zależności od jasności otoczenia.

Sławomir Węgrzyn
bsw@poczta.onet.pl

Wykaz elementów

- R1-R10 220Ω-470Ω (SMD 0603)
– wartość rezystorów należy dobrać w zależności od koloru zastosowanych diod LED: czerwonych – 470Ω, niebieskich – 330Ω, zielonych – 220Ω
 - C1 100nF (SMD 1206)
 - D1-D10 diody LED (SMD 1206)
– kolor wg uznania
 - U1 ATmega8A lub ATmega8L
(w obudowie TQFP)
 - Q kwarc zegarkowy 32768Hz
 - BTN mikroswitch prostokątny
(przewlekany lub SMD)
- Bateria CR2032

Płytką drukowaną jest dostępna w sieci handlowej AVT jako kit szkolny AVT-2983.

Rys. 5 Ustawienie Fuse-bitów

Fuse bits							
BODLEVEL	BODEN	SUT1	SUTO	CKSEL3	CKSEL2	CKSEL1	CKSELO
1	1	1	0	0	0	0	1
RSTDISBL	WDTON	CKOPT	EESAVE	BOOTSZ1	BOOTSZ0	BOOTRST	
1	1	0	0	1	0	0	1