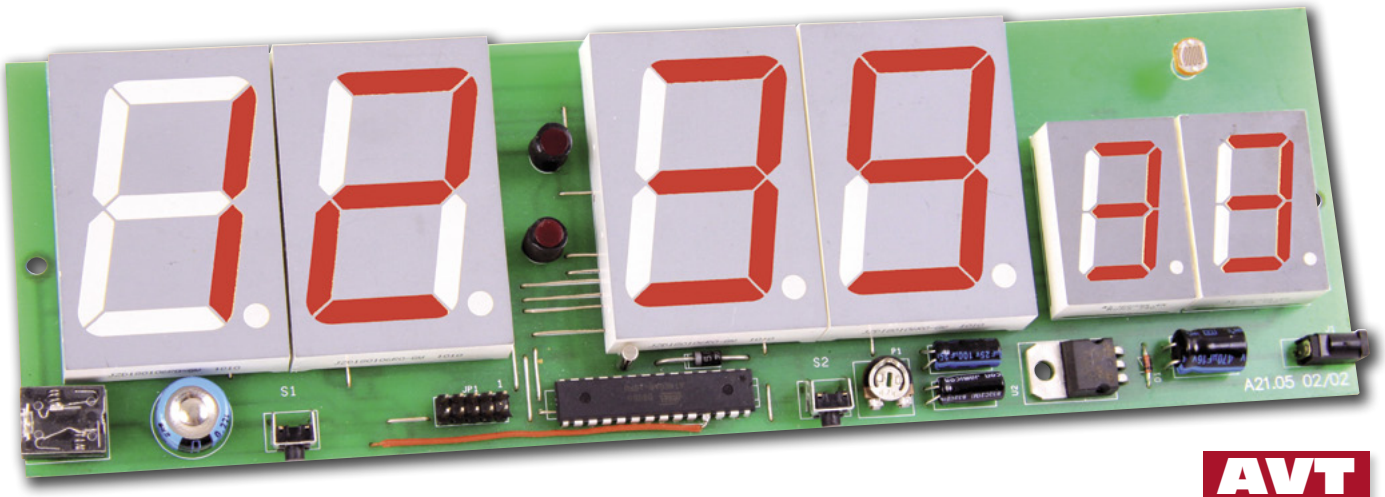


AVT 5281

„Inteligentny" zegar z wyświetlaczami LED

Chyba każdy elektronik zaprojektował i wykonał choć jeden zegar. Ja zbudowałem ich kilka, ale ostatnio znów wróciłem do tego tematu, a to z powodu specjalnych wymagań, których zazwyczaj nie spełniają komercyjne wyroby. Zegar powstał w odpowiedzi na zapotrzebowanie na zegarek naścienny z dużymi cyframi, widocznymi zarówno w dzień jak i w nocy, ale nieoświetlającymi w nocy pokoju. Tak powstał zegar, który mimo prostej budowy ma kilka bardzo przydatnych funkcji.

Rekomendacje: *ze względu na unikatowe funkcje, zegar przyda się w niejednym domu.*

Fotografia 1. Pierwszy prototyp zegara w wersji 4-cyfrowej

AVT-5281 w ofercie AVT:
AVT-5281A – płytka drukowana
AVT-5281B – płytka drukowana + elementy

Podstawowe informacje:
- Wyświetlanie czasu na dużym, czytelnym, 6-cyfrowym wyświetlaczu 7-segmentowym
- Wskazania minut, godzin i sekund.
- Możliwość wyświetlania temperatury.
- Jasność świecenia regulowana automatycznie w zależności od natężenia światła zewnętrznego.
- Zasilanie 8...20 VDC, pobór prądu do ok. 600 mA.
- Mikrokontroler ATmega8.
- Możliwość programowej korekty chodu zegara

Dodatkowe materiały na CD i FTP:
ftp://ep.com.pl, user: 10460, pass: 0646g3n0
- wzory płytek PCB
- karty katalogowe i noty aplikacyjne elementów oznaczonych w wykazie elementów kolorem czerwonym

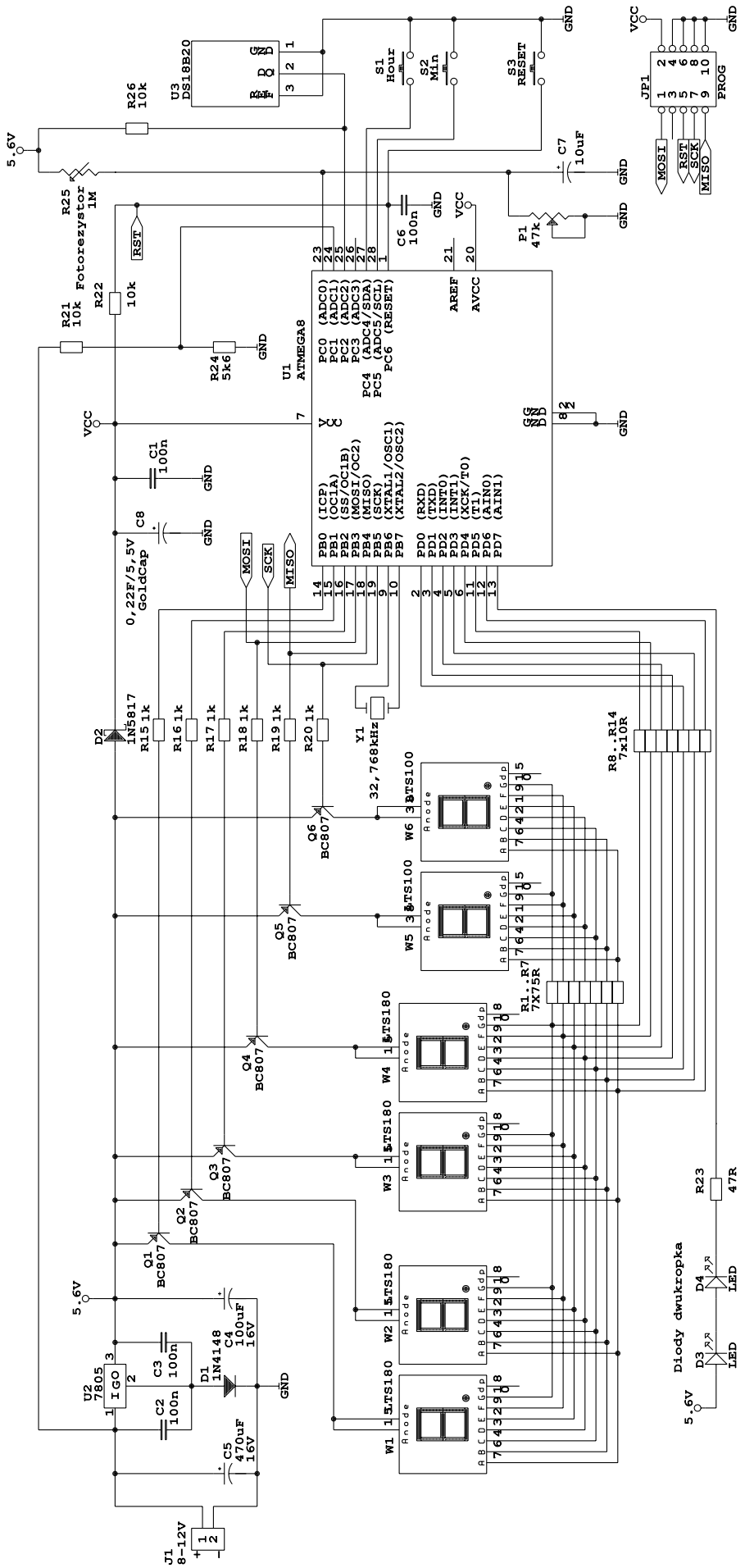
Projekty pokrewne na CD i FTP:
(wymienione artykuły są w całości dostępne na CD)

AVT-5273	Zegar cyfrowy z analogowym sekundnikiem (EP 1/2011)
AVT-5245	Zegar widmowy (EP 7/2010)
AVT-5145	Zegar retro na lampach NIXIE (EP 9/2008)
AVT-513	Zegar ze 100-letnim kalendarzem (EP 10-11/2003)
AVT-5048	Zegar z kalendarzem zasilany bateryjnie (EP 2/2002)
AVT-5022	Programowany zegar z DCF77 (EP 6-7/2001)
AVT-5002	Zegar cyfrowy z wyświetlaczem analogowym (EP 3/2001)
AVT-217	Zegar DCF (EP 7/1994)
AVT-2849	Tiny clock (EdW 1/2008)
AVT-2825	Zegar z budzikiem (EdW 5/2007)
AVT-2721	Mikroprocesorowy zegar (EdW 4/2004)
AVT-2632	Gigantyczny zegar (EdW 5/2002)

Początkowo przewidywałem wykonanie zegara w wersji 4-cyfrowej (**fotografia 1**), lecz dość szybko powstała kolejna wersja rozszerzona o obsługę sekundnika oraz z możliwością pomiaru i wyświetlania temperatury. Po pewnym czasie w oprogramowaniu przybyła możliwość obsługi dodatkowych driverów wyświetlaczy oraz analogowego sekundnika złożonego z 60 diod świecących. Całością zarządza mikrokontroler ATmega8. Oprogramowanie mikrokontrolera zostało napisane w języku C z wykorzystaniem darmowego kompilatora GCC w wersji 3.4.5 (WinAVR-20060125).

Opis układu i jego podstawowe funkcje

Najważniejszą funkcją każdego zegara jest wyświetlanie aktualnego czasu. Do odmierzania czasu nie został użyty żaden specjalizowany układ RTC. Czas jest mierzony z wykorzystaniem wewnętrznych mechanizmów mikrokontrolera ATmega8. Jako podstawę czasu użyto timer 2 mikrokontrolera w trybie asynchronicznym, który jest taktowany sygnałem z oscylatora z kwarcem „zegarkowym" 32,768 kHz dołączonym do pinów OSC1 i OSC2. Timer generuje przerwania co sekundę i są one zastosowane do aktualizacji zawartości programo-



Rysunek 2. Schemat ideowy zegara 6-cyfrowego

Wykaz elementów

Rezystory: (SMD, 0805)
R1...R7: 75 Ω
R8...R14: 10 Ω
R15...R20: 1 kΩ
R21, R22, R26: 10 kΩ
R23: 47 Ω
R24: 5,6 kΩ
R25: 1 MΩ (fotorezystor np. FR48/1M)
P1: 47 kΩ (pot. montażowy)
Kondensatory:
C1, C2, C3, C6: 100 nF (SMD, 0805)
C4: 100 µF/16 V
C5: 470 µF/16 V
C7: 10 µF/25 V
C8: ColdCap 0,22 F/5,5 V
Półprzewodniki:
U1: ATMega8 (DIL28)
U2: 7805 (TO220)
U3: DS18B20, DS18S20 lub DS1822 (opcjonalnie)
Q1...Q6: BC807 (SOT23)
D1: 1N4148
D2: 1N5817
D3,D4: LED (czerwona, 5 mm, z płaskim czołem)
W1...W4: JZD180106RO-GW (1,8”, czerwony)
W5,W6: JZD100106RO-GW (1,0”, czerwony)
Inne:
Y1: Kwarc 32,768 kHz
S1, S2: mikroprzycisk kątowy h=7,35 mm
S3: mikroprzycisk SMD INT-1187
JP1: Złącze szpilkowe 2×5 raster 2,54 mm
J1: Gniazdo DC 4,5×1 mm
J2: Gniazdo Jack 3,5 mm (dla czujnika temperatury)

wych liczników sekund, minut i godzin. Sam mikrokontroler taktowany jest z wewnętrznego generatora RC ustawionego na 2 MHz.

Dokładność zliczania czasu jest korygowana programowo. Jest to najefektywniejszy i najprostszy sposób na uzyskanie bardzo dokładnego zegara, z błędem mniejszym niż 2...3 sekundy na miesiąc (!), a jednocześnie umożliwia bardzo prostą kalibrację.

Cały układ, którego schemat ideowy przedstawiono na **rysunku 1** składa się ze stabilizatora zasilania, mikrokontrolera, driverów anod wyświetlaczy oraz wyświetlaczy z rezystorami ograniczającymi prąd. Do ustawiania godzin i minut służą dwa przyciski. Jasność świecenia jest regulowana na podstawie sygnału z fotorezy-

R E K L A M A

Na CD: karty katalogowe i noty aplikacyjne elementów oznaczonych w wykazie elementów kolorem czerwonym

stora. Dodatkowo zegar ma układ podtrzymania zasilania oraz opcjonalny czujnik temperatury.

Do wyświetlania godzin i minut użyto czerwonych wyświetlaczy o wielkości 1,8 cala (46 mm), natomiast sekund trochę mniejszych, o wysokości 1 cala (25,4 mm). W tym miejscu muszę zaznaczyć, że wybrany przeze mnie typ wyświetlaczy 1,8" jako jeden z niewielu ma w każdym segmencie dwie diody połączone szeregowo. Większość zamienników ma trzy diody i ich użycie zaowocuje kiepską jasnością świecenia. Przed ewentualnym zastosowaniem zamienników należy koniecznie sprawdzić ich kartę katalogową i konfigurację diod.

Wyświetlacze uzupełniają dwie diody świecące o średnicy 5 mm z płaskim czołem, włączone jako dwukropek. Ze względu na to, że zastosowane wyświetlacze mają spadek napięcia na pojedynczym segmencie wynoszący 4,2 V, to zastosowano prosty sposób na niewielkie podwyższenie napięcia zasilania wyświetlaczy poprzez zastosowanie diody krzemowej w obwodzie masy stabilizatora 7805, co podnosi jego napięcie wyjściowe do wartości około 5,6 V. Anody wyświetlaczy są sterowane za pomocą kluczy z tranzystorami PNP. W obwodach katod zastosowano rezystory ograniczające prąd wpływający do wyjść portu mikrokontrolera. Katody wyświetlaczy sekund mają dodatkowe rezystory wyrównujące ich jasność, ponieważ spadek napięcia na segmentach tych wyświetlaczy wynosi ok. 2,1 V. Podłączony do wyprowadzeń OSC1 i OSC2 rezonator kwarcowy nie potrzebuje dodatkowych kondensatorów od jego wyprowadzeń do masy, ponieważ programując odpowiednio bity procesora możemy włączyć wewnętrzne kondensatory 36 pF znajdujące się w procesorze.

Odmierzanie czasu

Odmierzanie czasu odbywa się w obsłudze przerwania po przepełnieniu timera 2, które następuje dokładnie co sekundę. Na **listingu 1** przedstawiono procedurę zliczania sekund wykonywaną w przerwaniu „OVERFLOW" timera 2.

To przerwanie jest wykonywane zawsze, nawet w przypadku braku zasilania sieciowego, gdy procesor pracuje z zasilaniem awaryjnym. Struktura „t" zawiera pola, w których jest przechowywana bieżąca wartość czasu. Jego wyświetlanie jest wykonywane w pętli głównej, w oparciu o aktualne wartości struktury „t"

Regulacja Jasności

Do wejścia PC0 (ADC0) jest przyłączony dzielnik napięcia z fotorezystorem R25, a wewnętrzny przetwornik A/D mierzy napięcie z tego dzielnika i odpowiednio dostosowuje jasność świecenia wyświetlaczy do natężenia oświetlenia zewnętrznego. Fotorezystor jest dołączony do zasilania +5,6 V, aby dzielnik nie pobierał prądu z kondensatora podtrzymującego zasilanie mikrokontrolera (backup) w przypadku zaniku zasilania sieciowego. Zastosowany fotorezystor powinien mieć rezystancję na ciemno około 1 MΩ. Potencjometr P1 służy do ustawienia czułości regulatora jasności. Odpowiednie ustawienie należy przeprowadzić w zależności od typu użytego fotorezystora oraz jasności filtru, za którym są schowane wyświetlacze i fotorezystor. Do regulacji jasności świecenia wyświetlaczy zastosowano sprzętowo-programowy PWM, niezbędny do multipleksowania wyświetlaczy. Napisałem „sprzętowo-programowy", ponieważ wykorzystano w nim sprzętowy mechanizm *output compare* timera 1 do ustalania wartości PWM i do ustalania momentu gaszenia wyświetlaczy, lecz samo gaszenie jest realizowane programowo w przerwaniu *compare match*. Multipleksowanie wyświetlaczy następuje w przerwaniu *overflow* tego samego timera.

Listing 1. Przerwanie obsługujące zliczanie czasu i korekcję dokładności.

```
SIGNAL(SIG_OVERFLOW2)  //RTC, przerwanie co 1 sekundę
{
  sekunda++;  //zwiększ pomocniczy licznik sekund
  if (++t.second == 60)  //zwiększ licznik sekund i sprawdź czy 60
  {
    t.second=0;  // zeruj licznik sekund
    if (++t.min == 60)  //zwiększ licznik minut i sprawdź czy 60
    {
      t.min = 0;  //zeruj licznik minut
      if (++t.hour == 24)  //zwiększ licznik godzin i sprawdź czy 24
        t.hour=0;  //zeruj licznik godzin
    }
    t.corr++;  // co minutę zwiększ również licznik korekcji
  }
}
```

Na **listingu 2** zamieszczono procedury obsługi przerwania służące do sterowania wyświetlaczem.

Przerwanie *SIG_OVERFLOW1* jest wywoływane z częstotliwością ok. 977 Hz (2 MHz/8/256). Najpierw gaszona jest poprzednia cyfra i jest wyłączane sterowanie anodami. Zmienne *neg_anod* i *neg_katod* ustawiane są na początku programu po przeprowadzeniu testów obciążenia pinów procesora, zapewniając odpowiednie wysterowanie wyjść niezależnie od ewentualnej negacji sygnałów na dodatkowych buforach. W ten sposób układ sam dostosowuje się do ewentualnie podłączonych dodatkowych driverów anod lub/i katod wyświetlacza. Następnie zwiększany jest licznik cyfr, i jeśli zrówna się on z liczbą za-

Listing 2. Procedury obsługi wyświetlaczy i regulacji jasności.

```
//Przerwanie overflow timera 1,
//obsługa multipleksowania wyświetlaczy

SIGNAL(SIG_OVERFLOW1)  //Multiplex LED
{
  static u08 cyfra,anod;
  u08 tmp,tmp2;

  if(licz)
    licz--;  //pomocniczy licznik czasu do wielu zastosowań
  KATODY = neg_katod;  //zgaś wyświetlacz
  anod <<= 1;  //wartość dla wybrania kolejnej anody
  tmp = 4;
  if(opt.sekundnik)  //maks. licznika cyfr to 4 lub 6
      tmp = 6;
  if(++cyfra == tmp)  //zwiększ licznik cyfr, maksimum?
  {
    cyfra = 0;  //zeruj licznik cyfr
    anod = 1;  //wartość dla wybrania pierwszej anody
  }
  ANODY = neg_anod^anod;  //wybierz kolejną anodę (XOR umożliwia
                  //sterowanie niezależnie od negacji na anodach)

  if( !opt.sekundnik && opt.dwukPB5 )  // oddzielna obsługa
                          //dwukropka na PB5
  {
    if(dwukropek)
    ANODY |= 1<<PB5;  //„1" na PB5, niezależnie od negacji anod
      else
        ANODY &= ~(1<<PB5);  //„0" na PB5
  }
  tmp = ledbuf[cyfra];  //pobierz znak do wyświetlenia
  if(cyfra == 0 && tmp == 0)  //jeśli na pozycji 10 godz.
                      //ma być wyłwietlone 0
    tmp2 = 0xFF;  //to wygaś dziesiątki godzin
      else
        tmp2 = pgm_read_byte(GENZNAK+tmp);  //lub pobierz stan katod
                  //z generatora znaków

  if(dwukropek||opt.dwukPB5)  //obsługa dwukropka na katodach
   tmp2 &= ~(seg_dp);  //zaświeć dwukropek lub użyj PD7 tylko do PWM
  KATODY = (neg_katod) ? tmp2 : ~tmp2;    // wyświetl cyfrę
}

//Przerwanie Compare A timera 1 - obsługa regulacji jasności
//minimalizacja czasu trwania obsługi przerwania
void SIG_OUTPUT_COMPARE1A(void) __attribute__ ((naked));
SIGNAL(SIG_OUTPUT_COMPARE1A)  //Softwarowy PWM do wyświetlacza
{
  asm volatile("PUSH R24");  //Zachowaj R24
  KATODY = neg_katod;  //zgaś wyświetlacz jeżeli TCNT1==OCR1
  asm volatile("POP R24");  //Odtwórz R24
  asm volatile("RETI");  //Powrót z ISR (atrybut "naked"
                      //nie generuje powrotu)
}
```

stosowanych wyświetlaczy (warunkowa instrukcja kompilacji *#ifdef*), to jest on zerowany. Dalej jest adresowana kolejna anoda (z tablicy, żeby było szybciej), z bufora wyświetlacza pobierany jest kolejny znak do wyświetlenia, przekodowany według tablicy *GENZNAK* zawierającej układ segmentów dla danego znaku. Teraz następuje sprawdzenie czy wyświetlane dziesiątki godzin są równe 0 (aby wygasić 0 na pierwszej pozycji), określenie stanu dwukropka i wysłanie odpowiedniego bajtu do portu sterującego katodami wyświetlacza, z uwzględnieniem ewentualnej negacji wszystkich bitów.

Przerwanie *SIG_OUTPUT_COMPARE1A*, wywoływane w momencie zrównania licznika timera 1 z zawartością rejestru OCR1AL, służy do regulacji jasności wyświetlaczy. Jego zadaniem jest zagaszenie w odpowiednim momencie aktualnie wyświetlanej cyfry. Kilka asemblerowych instrukcji oraz deklarację *__attribute__ ((naked))* wstawiono aby zminimalizować czas jego obsługi – kompilator nie traktuje tej procedury jak przerwania, co normalnie skutkuje odłożeniem zawartości standardowego zestawu rejestrów na stosie. Jedyny używany w tej procedurze rejestr jest zachowywany i odtwarzany „ręcznie".

Schemat czynnościowy obsługi tego przerwania jest następujący:

1. Zaświeć cyfrę.
2. Zaczekaj aż TCNT1 zrówna się z OCR1.
3. Zgaś cyfrę.
4. Zaczekaj do czasu zaadresowania kolejnej cyfry.

Czas pomiędzy pierwszym i ostatnim zadaniem jest stały, a zmienia się tylko moment zgaszenia cyfry. W ten sposób wszystko dzieje się autonomicznie, a poziom jasności zmieniamy zawartością rejestru OCR1A.

Pomiar temperatury

Pomiar temperatury jest wykonywany za pomocą cyfrowego termometru firmy Dallas DS18B20, DS18S20 lub DS1822. Program sam wykrywa podłączenie czujnika temperatury i w przypadku jego wykrycia odblokowuje opcję wyświetlania temperatury. Ponieważ wymienione powyżej układy nie są między sobą w 100% kompatybilne, oprogramowanie mikrokontrolera automatycznie rozpoznaje typ podłączonego czujnika. Czujnik podłącza się dwoma przewodami pomiędzy masę a linię portu PC2. Jeśli użyjemy czujnika w obudowie TO92, to zwieramy jego dwa skrajne wyprowadzenia i zwieramy do masy, a środkowe podłączamy do linii PC2 portu mikrokontrolera. W wersji 4-cyfrowej temperatura jest pokazywana z rozdzielczością 1°, a w wersji 6 cyfrowej – 0,1°. Dla czujników DS18S20 i DS1822 zastosowano procedurę zwiększającą rozdzielczość pomiaru temperatury. Temperatura jest wyświetlana na zmianę z aktualnym czasem, przy czym można ustawić zarówno okres wyświetlania czasu jak i temperatury.

Zasilanie

Zegar może być zasilany z niestabilizowanego zasilacza 9 V/300 mA, lub z jakiejś impulsowej ładowarki do telefonu komórkowego. Użyta przeze mnie ładowarka do telefonu Nokia 6210 daje na wyjściu około 9 V i jest idealna do tego celu. Gniazdo zasilania J1 ma wymiary przystosowane do wtyku takiej właśnie ładowarki.

Jako zasilanie awaryjne użyto kondensatora (C8) typu *GoldCap* o pojemności 0,22 F, podtrzymującego zasilanie samego procesora w przypadku zaniku napięcia sieciowego. Przy braku zasilania sieciowego zasilanie pozostałych obwodów jest odcięte przez diodę D2. Sygnał pobrany sprzed stabilizatora U2 jest podany poprzez dzielnik napięcia na linię portu PC1 procesora i informuje o występowaniu zasilania sieciowego, a w przypadku jego braku następuje wyłączenie wszystkich zbędnych obwodów mikrokontrolera. Jedynie jest realizowana funkcja odmierzania czasu (pętla pokazana na **listingu 3** oraz przerwanie z listingu 1). W tym stanie procesor pobiera tylko około 15 μA prądu, a więc użyty do podtrzymania *GoldCap* wystarcza na kilka-kilkanaście godzin pracy bez zasilania sieciowego. Po restarcie mikrokontrolera (pierwsze uruchomienie,\ lub zbyt długi zanik zasilania) zegar pokazuje **„--:-- --"** aż do momentu ręcznego ustawienia czasu. Na płytce drukowanej przewidziano miejsce na przycisk *RESET*, dostępny np. przez otwór w tylnej ściance zegara. Ręczny restart mikrokontrolera jest przydatny w przypadku zbyt długiej przerwy w zasilaniu sieciowym, co spowoduje spadek napięcia zasilania mikrokontrolera poniżej poziomu, przy którym może jeszcze pracować, lecz jednocześnie zbyt krótkiej do całkowitego rozładowania kondensatora podtrzymania i tym samym zadziałania wewnętrznych obwodów zerujących mikrokontroler. W takim przypadku może się zdarzyć, że po włączeniu zasilania sieciowego zegar będzie ciemny, więc musimy go zrestartować ręcznie, lub odczekać jeszcze kilka godzin bez zasilania sieciowego. Czytelnicy znający procesory rodziny AVR zapewne stwierdzą, że można wykorzystać wbudowany w procesor układ BOD, czyli czuwak napięcia zasilania procesora. Niestety, załączenie tego układu za pomocą fusebitów powoduje drastyczny wzrost poboru prądu w trybie *Power save* i bardzo szybkie rozładowanie kondensatora podtrzymującego zasilanie.

Ustawianie czasu

Przycisk K1 służy do ustawiania godzin, a K2 do ustawiania minut. Oba przyciski mają autorepetycję z dynamiczną zmianą szybkości repetycji. Przy zmianie zawartości licznika minut, licznik sekund jest automatycznie zerowany oraz jest kasowany licznik poprawki dokładności zegara. Od momentu zmiany minut poprawka jest więc liczona od początku. Zmiana zawartości licznika godzin nie ma wpływu na liczniki minut i sekund oraz poprawkę. Ponadto, pierwsze naciśnięcie klawisza zmiany minut umożliwia tzw. korektę czasu w zakresie ±30 s, co oznacza, że jeśli w momencie jego naciśnięcia licznik sekund wskazuje mniej niż 30 sekund, to jest zerowany bez zwiększania licznika minut. Natomiast, jeśli licznik sekund wskazuje 30 lub więcej albo jest to kolejne naciśnięcie przycisku bądź jego przytrzymanie w celu repetycji, to oprócz zerowania licznika sekund jest inkrementowany licznik minut.

Zegar nie ma pełnego kalendarza, więc nie ma możliwości automatycznej zmiany czasu zimowy - letni.

Ustawianie parametrów zegara

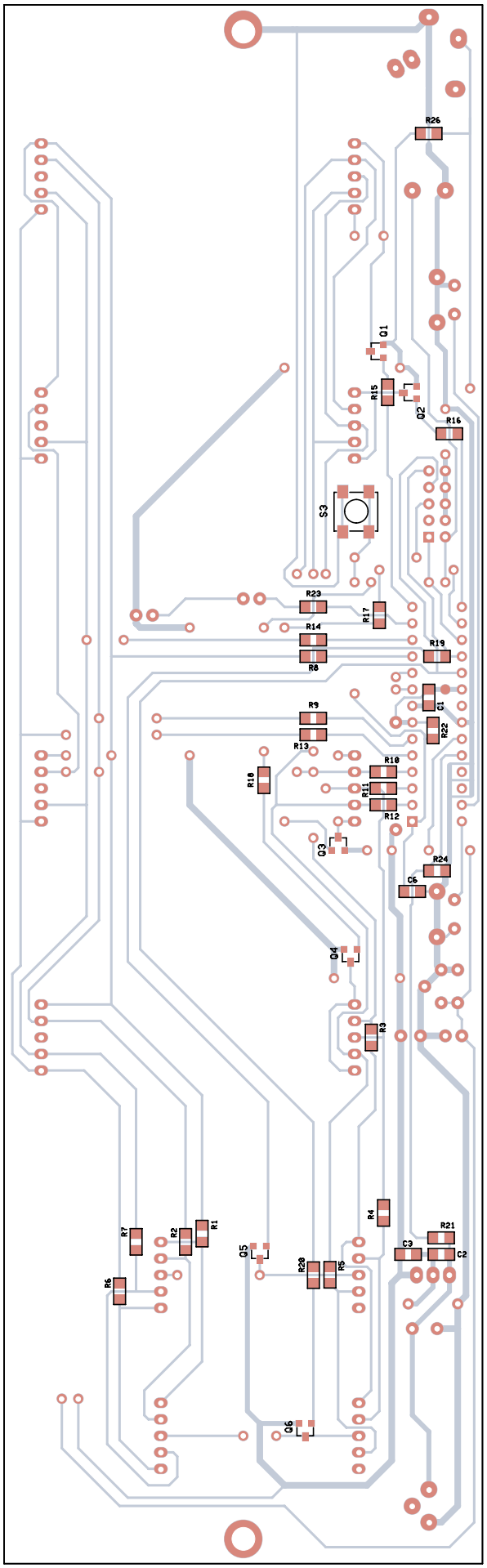
Oprogramowanie zegara napisano tak, aby można go było łatwo dostosować do własnych potrzeb. Aby wejść w tryb konfigurowania należy odłączyć zasilanie sieciowe, nacisnąć jednocześnie oba przyciski i trzymając je wciśnięte włączyć zasilanie sieciowe. Wówczas zegar wyświetli komunikat „SEt", a po puszczeniu klawiszy numer programu, w tym przypadku „P1". W tym trybie klawiszem ustawiania godzin zmieniamy po kolei numer programu (P1 -> P2 -> P3 -> P4 -> P5 -> P6 -> P7 -> P1 ...), a klawiszem ustawiania minut wchodzimy w odpowiedni program. Poszczególne programy umożliwiają:

- **P1** - ustawianie okresu (w sekundach) wyświetlania czasu; w przypadku niestosowania czujnika temperatury, wpisana wartość jest nieistotna, ponieważ zegar i tak będzie cały czas pokazywał aktualną godzinę,
- **P2** - ustawianie okresu (w sekundach) wyświetlania temperatury; wpisanie wartości „0" powoduje wyłączenie wyświetlania temperatury,
- **P3** - ustawianie korekcji dokładności pracy zegara,
- **P4** - ustawianie opóźnienia (filtru) reakcji na zmianę oświetlenia otoczenia,
- **P5** - ustawianie minimalnej jasności wyświetlacza,
- **P6** - ustawienie częstotliwości multipleksowania wyświetlaczy,
- **P7** - wybór opcji i trybu pracy zegarka.

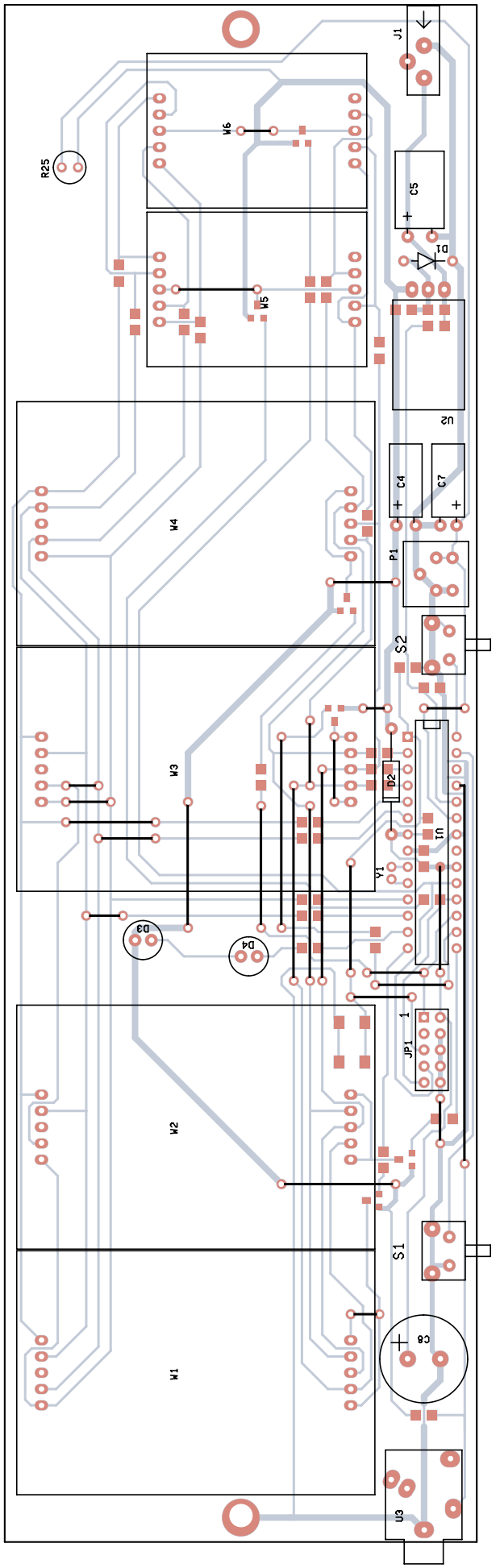
W każdym z programów wartość parametru zwiększa się klawiszem „minuty" a zmniejsza klawiszem „godziny". Wyjście z programu następuje automatycznie po 5 sekundach od ostat-

Listing 3. Pętla wykonywana przy zasilaniu awaryjnym.

```
do {
  MCUCR = 0xB0;                          //Wybierz tryb power save
  asm volatile(„SLEEP");                 //Wejdź w power save
  asm volatile("NOP");                   //Czekaj na zakończenie wybudzenia z p.s.
  TCCR2=0x05;                            //Zapisz coś do TCCR2 (wymagane dla
                                         //synchronizacji)
  while(ASSR&0x07);                      //czekaj na aktualizacje rejestru TCCR2
}while(bit_is_clear(PINC,PWR_FAIL)); //sprawdź status zasilania,
                                         //jeśli dalej brak, powtórz
```



Rysunek 3. Rozmieszczenie elementów SMD



Rysunek 4. Rozmieszczenie elementów przewlekanych

Listing 4. Sposób korekcji dokładności pracy zegara

```
if(t.second == 30)                                  // jeśli wartość sekund = 30
{
  if(poprawka != 0 && t.corr >= absint(poprawka))   //jeśli poprawka !=0 i licznik
  {                                                 // t.corr doszedł do wartości poprawki
    t.corr -= absint(poprawka);                     // odejmij poprawkę od licznika (po obcięciu znaku)
    if(poprawka < 0)
      t.second--;                                   //jeśli wartość poprawki jest ujemna to odejmij sekundę
    else
      t.second++;                                   //jeśli poprawka dodatnia to dodaj sekundę
  }
}
```

niego naciśnięcia klawisza, a zegar wraca do trybu wyboru programu. Wyjście z trybu wyboru programu, czyli powrót do wyświetlania czasu następuje automatycznie po 10 sekundach od ostatniego naciśnięcia klawisza.

Program „P3" wymaga nieco szerszego omówienia. Oprócz zliczania sekund, minut i godzin, wprowadzono dodatkowy licznik minut służący do korekcji dokładności pracy zegara (zmienna *t.corr* na list. 1). Co minutę jest on inkrementowany a jego wartość jest porównywana z ustawioną w programie P3 poprawką. Jeśli wartość tego licznika zrówna się lub przekroczy wartość poprawki, to w 30 sekundzie kolejnej minuty zegar zostanie „popchnięty" lub „cofnięty" o jedną sekundę, a licznik poprawki wyzerowany. Właściwie to nie wyzerowany, tylko od jego zawartości zostanie odjęta wartość poprawki, bo testy poprawki są wykonywane tylko wtedy, gdy zegar jest zasilany sieciowo i mogłoby wystąpić zgubienie jakiejś poprawki, ale efekt końcowy jest praktycznie taki sam, czyli co określoną jako „poprawka" liczbę minut, jest dodawana lub odejmowana jedna sekunda od aktualnego czasu. Na **listingu 4** pokazano ten fragment programu. Poprawka może być ustawiona w zakresie od –1999...+1999. Ustawienie poprawki na wartość dodatnią powoduje przyspieszanie pracy zegara, a na wartość ujemną – jego opóźnianie. W celu ustalenia właściwej wartości poprawki należy uruchomić zegarek, ustawić poprawkę na zero (co powoduje wyłączenie korekcji), jak najdokładniej ustawić bieżący czas (najlepiej według zegara z DCF), po czym umieścić zegarek w miejscu docelowym, (bo temperatura otoczenia ma również wpływ na częstotliwość kwarcu a więc na dokładność zegarka). Następnie odczekujemy około 6 dni (raczej nie należy przekraczać 6,5 dnia bo w tym czasie zegar zlicza czas pracy i go wyświetla, ale może wyświetlić maksymalnie do 9999 minut, tj. niecałe 7 dni). Około 6 dnia pracy zegara sprawdzamy błąd jego wskazań, znów porównując wskazywany czas z zegarem z DCF lub jakimś innym wzorcem czasu. Najprawdopodobniej będzie późnił. Przyjmijmy, że zegar spóźnił się o 1 minutę i 7 sekund, czyli o 67 sekund. Teraz należy na chwilę wyłączyć zasilanie sieciowe i załączyć je ponownie. Przez około sekundę po załączeniu zasilania zegar wyświetli zawartość licznika poprawki (w minutach), czyli czas pracy od momentu ostatniego ustawiania. Wartość poprawki, którą należy wpisać w programie P3 obliczymy jako stosunek czasu pracy zegara w minutach do błędu wskazań wyrażonego w sekundach. Przyjmijmy, że po załączeniu zasilania zegar wyświetlił 8960. Dzielimy 8960 przez 67, co daje 133,73. W zaokrągleniu jest to 134. Właśnie tę wartość (134) ustawiamy w programie P3 jako wartość poprawki, w wyniku czego co 134 minuty program doda do aktualnego czasu 1 sekundę, co spowoduje że odchyłka zostanie zniwelowana. Jeśli zegar zamiast spóźniać się będzie się śpieszył, to wyliczoną poprawkę ustawiamy jako wartość ujemną.

Program „P4" umożliwia ustawienie opóźnienia reakcji regulatora jasności świecenia wyświetlacza na zmianę natężenia oświetlenia zewnętrznego. Czym mniejsza wartość, tym szybsza reakcja. Ustawienie zbyt małej wartości może spowodować „sprzęganie" się fotorezystora z wyświetlaczem i migotanie wyświetlacza.

Program „P5" umożliwia ustawienie minimalnej luminancji wyświetlacza. W trakcie regulacji tej wartości, luminancja wyświetlacza zmienia się na bieżąco, więc możemy w ciemności ustawić ją tak, aby wyświetlacz był dobrze widoczny i jednocześnie nie raził oczu i nie oświetlał całego pokoju.

Program „P6" umożliwia zmianę częstotliwości taktowania mikrokontrolera w górę lub w dół, a co za tym idzie również i częstotliwość multipleksowania wyświetlaczy. Może to być przydatne w przypadku, gdyby zegar zakłócał pracę np. pobliskiego odbiornika DCF lub byłyby widoczne zdudnienia świecenia wyświetlacza ze świetlówkami kompaktowymi bądź jarzeniówkami. W takim przypadku należy doświadczalnie dobrać odpowiednią wartość częstotliwości taktowania. Zmienia się ona na bieżąco, podczas regulacji jej wartości.

Program „P7" umożliwia zmianę konfiguracji pracy zegara. Wartość ustawiona w programie P7 jest sumą wartości poniższych opcji (dodajemy do siebie wartości opcji, które mają być aktywne):

1 - wersja 6 cyfrowa,

2 - migotanie dwukropka przy wyświetlaniu czasu,

4 - w wersji 4 cyfrowej termometr wyświetla dziesiąte części stopnia i nie wyświetla znaczka °C,

Montaż i uruchomienie:

Zegar został zmontowany na płytce drukowanej o wymiarach 240 mm×75 mm. Rozmieszczenie elementów na płytce drukowanej pokazano na **rysunku 2** i **rysunku 3**.

Montaż rozpoczynamy od przylutowania wszystkich elementów montowanych powierzchniowo, a następnie wszystkich mostków znajdujących się po stronie elementów. Mostek od nóżki 25 mikrokontrolera wykonujemy izolowanym przewodem. Przed wlutowaniem mikrokontrolera należy sprawdzić działanie stabilizatora zasilania mierząc napięcie na jego wyjściu. Powinno ono wynosić od 5,6 do 5,8 V. Na samym końcu montujemy wyświetlacze, kondensator GoldCap C8, fotorezystor i diody dwukropka, które powinny być zamontowane na takiej wysokości, aby ich czoła były na równi z wyświetlaczami.

Po zmontowaniu całości ustawiamy potencjometr P1 w połowie zakresu i załączamy zasilanie. Jeśli mikrokontroler jest wstępnie zaprogramowany, to zegar powinien wyświetlić „**--:-- --**". Przy pierwszym załączeniu zasilania zegar może „wystartować" z kilkusekundowym opóźnieniem potrzebnym na naładowanie kondensatora C8.

Jeśli nasz mikrokontroler jest „czysty", to do złącza JP1 podłączamy programator STK200 (lub kompatybilny) i programujemy mikrokontroler plikiem „zegarek.hex" lub „zegarek.bin". Oprócz wgrania pliku bardzo ważne jest prawidłowe ustawienie fusebitów procesora zgodnie z opisem w pliku źródłowym albo w pliku „fuse.txt". Zgodnie z logiką Atmela, w opisie „1" oznacza bit niezaprogramowany a „0" zaprogramowany. Po odłączeniu programatora zegar powinien wystartować.

Przyciskami S1 i S2 wstępnie ustawiamy czas i sprawdzamy poprawność pracy zegara. Jeśli podłączymy czujnik temperatury, to co 10 sekund na 2 sekundy powinna być wyświetlana aktualna temperatura. Możemy także sprawdzić działanie regulacji jasności, poprzez zakrycie ręką fotorezystora. Wyświetlacz powinien się przyciemnić. Po tych próbach możemy rozpocząć ustawianie parametrów zegara oraz jego dokładności.

W egzemplarzu prototypowym obudowę wykonano z pasków miedziowanego laminatu zlutowanych na kształt ramki, do której od wewnątrz przylutowano dwie tulejki M3 umożliwiające przykręcenie płytki drukowanej zegara, a od czoła przyklejono filtr wykonany z kawałka dymnej pleksi. Potencjometr P1 ustawiamy tak wyświetlacz był odpowiednio widoczny przy danej jasności otoczenia, ale tę regulację można wykonać dopiero po umieszczeniu całości w obudowie z filtrem dla wyświetlaczy.

Romuald Biały
romek_b@o2.pl