

# Refleksometr... szachowy

Urządzenie spełnia dwie funkcje: jest zegarem szachowym oraz refleksometrem. Może także odmierzać czas przy myciu zębów czy graniu w gry komputerowe. Okaze się niezwykle przydatne wszystkim szachistom oraz rodzicom dzieci w wieku przedszkolno-szkolnym.

## Do czego to służy?

Przewrotna nazwa tego urządzenia wzięła się stąd, że pełni ono dwie funkcje: zegara szachowego oraz refleksometru. Motywacją do budowy tego urządzenia były rozgrywki szachowe między mną a moim 7-letnim synkiem Szymonem. Konstrukcja urządzenia nie jest skomplikowana, więc średniozaawansowana osoba może zbudować to urządzenie w jeden wieczór.

Urządzenie to będzie przydatne wszystkim fanom szachów, którzy w przerwach między rozgrywkami chcą mierzyć i poprawiać szybkość reakcji na bodźce optyczno-dźwiękowe. Urządzenie sprawdza się również jako odmierzacz czasu mycia zębów przez dzieci czy czasu wyznaczonego na gry komputerowe, dlatego budowę tego urządzenia polecam też wszystkim rodzicom dzieci w wieku przedszkolno-szkolnym.

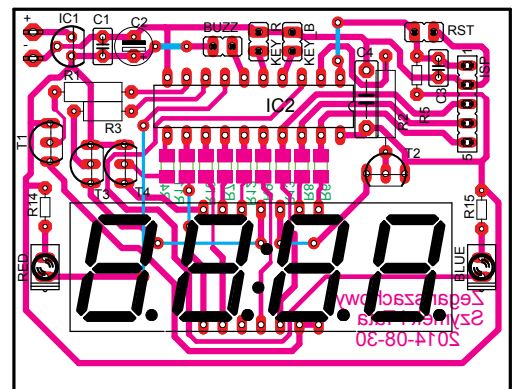
## Jak to działa?

Schemat ideowy zegara szachowego pokazany jest na rysunku 1. Urządzenie zostało zbudowane w oparciu na bardzo popularnym mikrokontrolerze ATtiny2313. Do wyświetlania pozostałego czasu wykorzystano 4-cyfrowy wyświetlacz LED w jednej obudowie. Ze względu na wysoką wydajność prądową wyjście I/O zastosowanego mikrokontrolera wyświetlacz sterowany jest bezpośrednio z portu PB poprzez rezystory R6–R13 ograniczające prąd diod. Wyświetlacz sterowany jest metodą multiplexową poprzez cztery tranzystory T1...T4 oraz rezystory R1...R4. Urządzenie ma dwie diody – czerwoną i niebieską – wskazujące gracza, któremu odmierzany jest aktualnie czas. Zastosowano dwa przyciski do przełączania czasu między graczami. Przyciski podłączono do portów PA0

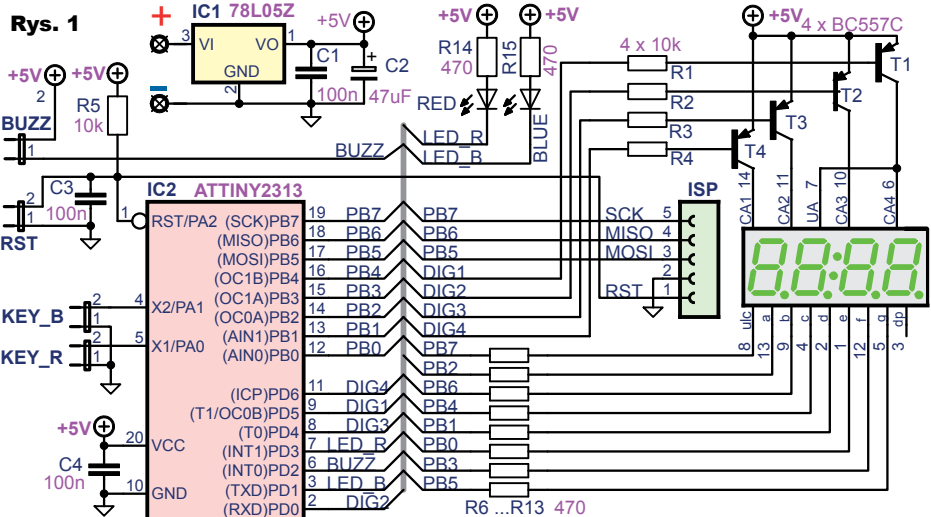
i PA1, współdzielonych z wyprowadzeniami, pod które typowo podłącza się rezonator kwarcowy taktujący mikrokontroler. W prezentowanym układzie zrezygnowano z rezonatora kwarcowego, co zmniejsza nieco jego koszty, ale pogarsza precyzję odmierzania czasu. Ponieważ układ przeznaczony jest do odmierzania krótkich okresów (maksymalnie 99 minut), precyzja nie jest tu niezbędna. Zgodnie z notą aplikacyjną mikrokontrolera ATtiny2313, wewnętrzny generator RC jest wykalibrowany z dokładnością do  $\pm 10\%$  częstotliwości nominalnej, a przy zastosowaniu procedury kalibracji można dokładność poprawić do  $\pm 2\%$ , co przy 99 minutach daje 2 minuty niedokładności. Przy typowych rozgrywkach szachowych odmierzany czas jest rzędu 20 minut, zatem niedokładność odmierzania czasu wyniesie poniżej pół minuty, co w tym zastosowaniu wydaje się być akceptowalne. Ostatecznie można zwiększyć precyzję



Rys. 2



Rys. 1



pomiaru czasu przez zmianę wartości stałej *Timer\_reload* w programie, która domyślnie wynosi 8000. W prototypie po wpisaniu wartości 8070 udało się uzyskać precyzję pomiaru na poziomie 2 sekund opóźnienia na 20 minut.

W układzie zastosowano trzeci przycisk podłączony do wyprowadzenia zerującego mikrokontroler, co umożliwia wyzerowanie zegarów obu graczy i rozpoczęcie nowej rozgrywki.

Układ zasilany jest z baterii 9V przez włącznik podświetlany diodą LED podłączoną do napięcia zasilania.

Program dla mikrokontrolera został napisany w środowisku BASCOM. Pliki źródłowe z komentarzami i pliki wynikowe są umieszczone w Elporcie wśród materiałów dodatkowych do tego numeru EdW. Komentarza wymaga fragment kodu refleksometru generującego losowy bodziec (zaświecenie diody czerwonej, niebieskiej lub generowanie dźwięku) – **listing 1**.

W pierwotnym rozwiązaniu wybór bodźca następował przez sprawdzanie wartości dwóch najmłodszych bitów losowej wartości zmiennej *Losowe\_opoznienie*. Ponieważ jednak funkcja RND() zwracająca wartość losową jest zrealizowana programowo, każdorazowe

**Listing 1**

```

Losowe_opoznienie = Rnd(10000)
Bodziec_swietlny = Losowe_opoznienie.0
, zapamiętaj bit.0
, będzie decydował o kolorze bodźca świetlnego
Falstart = 0 ,wstępne wyzerowanie flagi
,petla losowego opóźnienia między bodźcami
,wewnątrz petli sprawdzanie,
, czy ktoś nie oszukuje naciskając szybko przyciski
, jeśli oszukuje -> falstart = 1 i wynik wynosi 9999ms.
While Losowe_opoznienie > 0
  If Klawisz_czerwony = 0 Then
    Falstart = 1
    Exit While
  End If
  If Klawisz_niebieski = 0 Then
    Falstart = 1
    Exit While
  End If
  Waitms 1
  Decr Losowe_opoznienie
Wend
, jeśli nie było oszustwa,
, wybierany jest bodziec i rozpoczyna się odliczanie
If Falstart = 0 Then
  Stop_refleks = 0 ,uruchomienie zliczania
  If Bit_test = 0 Then
    If Bodziec_swietlny = 1 Then
      Reset Led_czerwony ,bodziec: światło czerwone
      While Klawisz_czerwony = 1 ,oczekiwanie na reakcje gracza
      Wend
      Set Led_czerwony
    Else
      Reset Led_niebieski ,bodziec: światło niebieskie
      While Klawisz_niebieski = 1 ,oczekiwanie na reakcje gracza
      Wend
      Set Led_niebieski
    End If
  Else
    Reset Buzzer ,bodziec: sygnał dźwiękowy
    While Klawisz_czerwony = 1 ,oczekiwanie na reakcje gracza
    Wend
    Set Buzzer
  End If
  Stop_refleks = 1 ,zatrzymaj pomiar
Else
  Refleksometr = 9999 ,przedwczesne wcisnięcie przycisku
End If

```

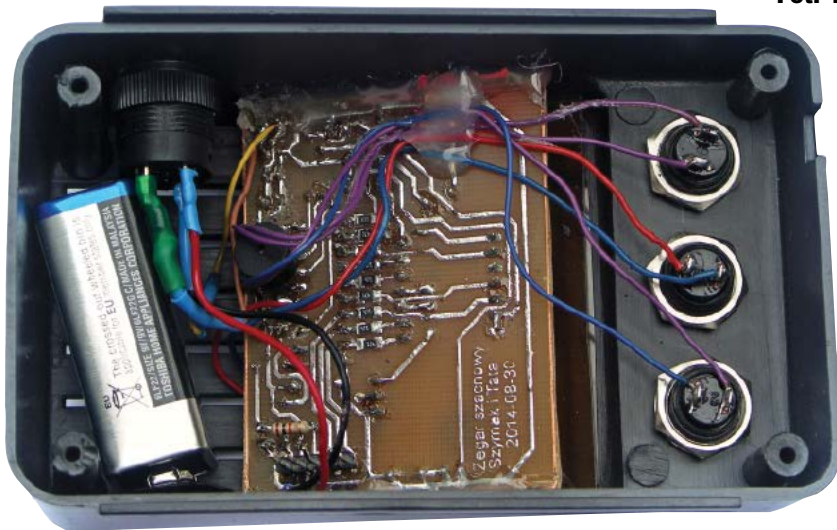
uruchomienie urządzenia wygeneruje ten sam zestaw losowych liczb, zatem i bodźce pojawią się w identycznej kolejności. Aby zwiększyć przypadkowość bodźca, wybór między bodźcem dźwiękowym a świetlnym określany jest na podstawie bitu nr 1 zmiennej *Refleksometr* (zapamiętanego w pomocniczej zmiennej *Bit\_test*) przechowywanej czas reakcji na bodziec z poprzedniej próby.

Bodźce świetlne są natomiast określane na podstawie najmłodszego bitu zmiennej *Losowe\_opoznienie*, przechowywanej w zmiennej pomocniczej *Bodziec\_swietlny*. W ten sposób kolejność bodźców jest bardziej nieprzewidywalna.

## Montaż i uruchomienie

Układ można zamontować na płytce drukowanej, której projekt pokazany jest na **rysunku 2**. Montaż najlepiej rozpocząć od elementów SMD i zwrócić uwagę na największych elementach. W urządzeniu zastosowano brzęczyk z wewnętrznym generatorem zasilany napięciem 5V. Brzęczyk został przyłutowany od strony ścieżek ze względu na swoje gabaryty. Płytkę elektroniki wklejono w obudowę klejem na gorąco. Z tego powodu również złącze do programowania mikrokontrolera zostało przyłutowane od strony ścieżek, gdyż w przeciwnym razie podłączenie programatora po wklejeniu płytki w obudowę byłoby niemożliwe. Fotografia wstępna

Fot. 1



oraz **fotografia 1** pokazują zmontowany model. Ponieważ Szymek bardzo interesuje się elektroniką, dzielnie wlułował wszystkie elementy – również SMD – w prezentowanym prototypie.

Aby układ działał poprawnie, należy najpierw ustawić odpowiednie FUSE BITS w mikrokontrolerze tak, aby ustawić wewnętrzne taktowanie zegarem  $RC=8\text{MHz}$  oraz wyłączyć wewnętrzny dzielnik częstotliwości przez 8. Następnie można przystąpić do zaprogramowania mikrokontrolera. Układ zmontowany ze sprawnych elementów oraz odpowiednio zaprogramowany powinien od razu prawidłowo pracować. Osoby nieoświadczone powinny poprosić kogoś o pomoc w zaprogramowaniu procesora.

Po uruchomieniu układ rozpoczyna pracę jako zegar szachowy. W pierwszej kolejności układ czeka na ustawienie czasu dla gracza czerwonego za pomocą czerwonego przycisku. Zatwierdzenie czasu następuje po wciśnięciu przycisku niebieskiego. Program przechodzi do ustawiania czasu dla gracza niebieskiego. Zatwierdzenie następuje po wciśnięciu przycisku czerwonego. Odliczanie rozpoczyna się po wciśnięciu przycisku przez gracza, który rozpoczyna rozgrywkę. Gdy czas któregoś z graczy dobiegnie końca, układ sygnalizuje to krótkim sygnałem dźwiękowym i kończy pracę. Nowe uruchomienie następuje po wyzerowaniu układu trzecim przyciskiem podłączonym do linii zerującej mikrokontrolera.

Funkcję refleksometru uruchamia się przez przytrzymanie wciśniętego czerwonego przycisku z jednoczesnym wyzerowaniem urządzenia. Na wyświetlaczu pojawi się najlepszy zarejestrowany do tej pory czas reakcji na bodziec. Wartość tę można skasować poprzez

wciśnięcie obu przycisków, tj. czerwonego i niebieskiego i wyzerowanie urządzenia. Skasowanie wyniku sygnalizowane jest długim dźwiękiem oraz wyświetleniem komunikatu „dEL”. Skasowanie najlepszego wyniku oznacza nadpisanie go wartością 9999, i taka wartość się wyświetli tuż po skasowaniu. Przy pierwszym pomiarze reakcji ten wynik zostanie nadpisany. Uwaga! Przy pierwszym uruchomieniu (lub każdorazowo przy przeprogramowaniu mikroprocesora) należy przeprowadzić procedurę kasowania najlepszego wyniku pomiaru refleksu, gdyż po przeprogramowaniu mikrokontrolera wartość rejestrów wynosi 0.

Program refleksometru rozpoczyna odliczanie losowego czasu w zakresie ok. 1...10 sekund, po czym generuje jeden z trzech bodźców: zaświecenie diody czerwonej, niebieskiej lub dźwięk z brzęczyka. Trzeba jak najszybciej wcisnąć przycisk w odpowiednim kolorze. Dla brzęczyka wciska się tylko przycisk czerwony, co jest dodatkowym utrudnieniem. Czas reakcji mierzony w milisekundach wyświetla się na wyświetlaczu przez 4 sekundy, po czym wynik jest zerowany i cykl zaczyna się od nowa.

W prototypowym urządzeniu pobór prądu w normalnym trybie pracy wynosi ok. 60 mA. Przy zastosowaniu baterii litowej o pojemności 1200 mA układ powinien pracować co

najmniej 17 godzin, czyli ok. 25 rozgrywek, licząc po 40 minut na rozgrywkę. W trybie uśpienia prąd pobierany z baterii wynosi ok. 2,5 mA i jest dość wysoki ze względu na zastosowany liniowy regulator typu 78L05, którego prąd spoczynkowy wg specyfikacji może wynieść nawet 5,5 mA. Z tego właśnie powodu zastosowano mechaniczny wyłącznik zasilania, a nie jak pierwotnie planowano przechodzenie układu w tryb wyłączenia (*power down*). Przy zastosowaniu trzech baterii typu AA można by zrezygnować z liniowego regulatora napięcia i zasilic układ bezpośrednio z baterii. Wówczas można by nie stosować mechanicznego wyłącznika i powrócić do pierwotnej koncepcji z wprowadzaniem układu w tryb wyłączenia *power down*. Ponieważ jednak objętość kodu wynosi 2042 bajty (na 2048 B dostępnych), wymagałoby to wprowadzenia pewnych modyfikacji w kodzie tak, aby wygospodarować nieco pamięci na dodatkową funkcję uśpienia układu. Można tego dokonać przez usunięcie niektórych rozwiązań

„estetycznych” zaimplementowanych w kodzie (np. wyświetlanie komunikatu „dEL” czy czterech kresek „----” podczas losowej przerwy w funkcji refleksometru).

**Szymon Bajraszewski**  
**Tomasz Bajraszewski**  
tobajer@poczta.onet.pl



R E K L A M A

#### Wykaz elementów

R6–R13	470Ω SMD 1206
R14, R15	470Ω
R4	10 kΩ SMD 1206
R1–R3, R5	10kΩ
IC1	LM78L05
IC2	ATtiny2313
BLUE	diody LED 3mm niebieska
RED	diody LED 3mm czerwona
C1, C3, C4	100nF
C2	47uF
T1–T4	BC557C
4X7segm. wyświetlacz LED, AF-05643FS	
BUZZER – brzęczyk na napięcie 5V	
5x1 złącze typu goldpin	

**Płytką drukowaną jest dostępna w sieci handlowej AVT jako kit szkolny AVT-3123.**