

Zegar Nixie z „kukułką”

Zegar Nixie, tykający jak klasyczny zegar mechaniczny, wyposażony w kukulkę lub inne zwierzątko sygnalizujące pełną godzinę. Modułowa budowa ułatwia dostosowanie zegara do własnych upodobań.

Zegary budowane na lampach Nixie od co najmniej kilkunastu lat cieszą się dużą popularnością. Niewątpliwie wpływ na ten stan rzeczy ma urok cyfr wyświetlacza, żarzących się ciepłym, pomarańczowym światłem i wyglądających jak pisanne. W Internecie i prasie, w tym w EdW, pojawiło się już wiele opisów zegarów Nixie. Można odnieść wrażenie, że w tej tematyce nic nowego powiedzieć już nie można. A jednak wszystkie, lub przynajmniej zdecydowana większość opisywanych zegarów, pozbawiona jest elementu, który klasycznym zegarom mechanicznym dodawał uroku. Elementem tym jest

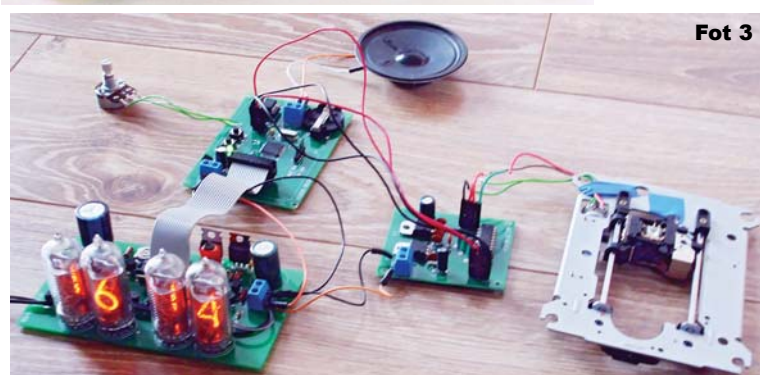
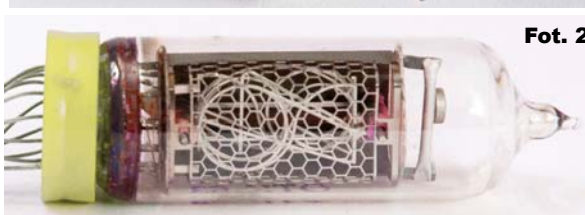
tytułowa „kukułka” (dlaczego pisana w cudzysłowie, o tym w dalszej części tekstu), wychodząca z dziupli i „odśpiewująca” pełną godzinę. Drugim ważnym atrybutem jest charakterystyczne „tykanie” udające pracę wahadła i układu trybów zębatych zegara. Warto zajrzeć do Elportalu, gdzie wśród materiałów dodatkowych do tego numeru udostępnione są m.in. dwa filmiki prezentujące pracę zegara. W ramach tego projektu wykonany został kompletny zegar, pokazany na fotografii tytułowej, zawierający płytki „domowej roboty” i nieduże lampy z odczytem czołowym IN-2 (IH-2) – **fotografia 1**, a potem powstała wersja z większymi lampami IN-14 (IH-14) – **fotografia 2**. Zestaw płytek tej drugiej wersji pokazany jest na **fotografii 3**.

Opis układu

Na schemat blokowy na **rysunku 1** składają się:

1. płyta główna, sterująca zegarem;
2. płyta wyświetlaczy Nixie;
3. płyta wykonawcza sterująca mechanizmem poruszającym kukulkę.

Odpowiednie moduły pokazane są na **fotografiach 4...7**. Budowa moduła ma kilka istotnych zalet. Umożliwia dogodną dla

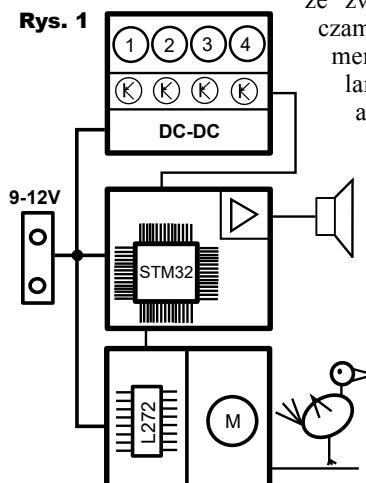


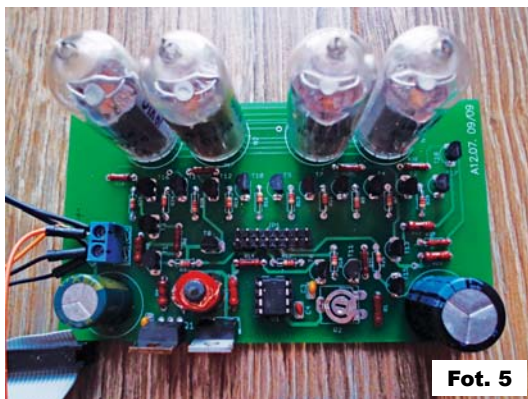
zainteresowanego konfigurację zegara: ktoś, komu nie zależy na kukulce, a np. chciałby mieć efekt tykania i sygnał np. bicia dzwonów, może pominąć moduł mechaniczny z silnikiem krokowym, a do mikrokontrolera wgrać odpowiedni plik dźwiękowy. Z kolei jeżeli ktoś chce mieć koniecznie zegar z kukulką, ale nie zależy mu na lampach Nixie, to może moduł wyświetlacza zastąpić modułem

ze zwykłymi wyświetlaczami LED siedmiosegmentowymi (ponieważ lampa Nixie ma 10 anod, a wyświetlacz LED 8, dwa pozostaną niewykorzystane, oczywiście konieczna będzie modyfikacja programu). Możliwości modyfikacji są bardzo duże.

Podstawowym elementem płyty głównej jest 32-bitowy mikrokontroler typu

Rys. 1



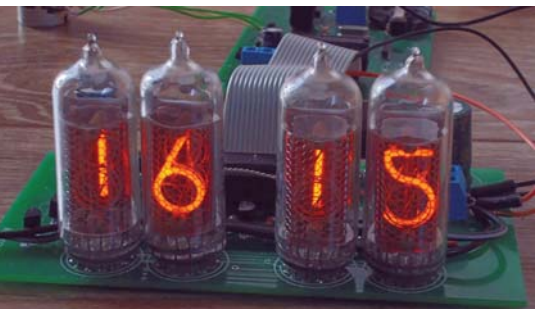


Fot. 5



Fot. 6

STM32F303. Wykorzystanie tak potężnego procesora może początkowo wydać się dziwne, gdyż w zupełności wystarczyłby tutaj układ 8-bitowy. Jednak duża pojemność pamięci Flash pozwala zapisać w niej wszystkie pliki dźwiękowe, bez konieczności stosowania jakichkolwiek zewnętrznych nośników danych jak np. karty SD. Obecność 12-bitowych przetworników DAC pozwala bezpośrednio odtwarzać z pamięci pliki dźwiękowe, bez konieczności stosowania zewnętrznych przetworników lub wykorzystania modulacji PWM. Dodatkowo obecność zegara RTC z kalendarzem zwalnia z potrzeby stosowania zewnętrznego układu. Uwaga – w oryginalnej wersji czas odmierzany jest za pomocą timerów, więc



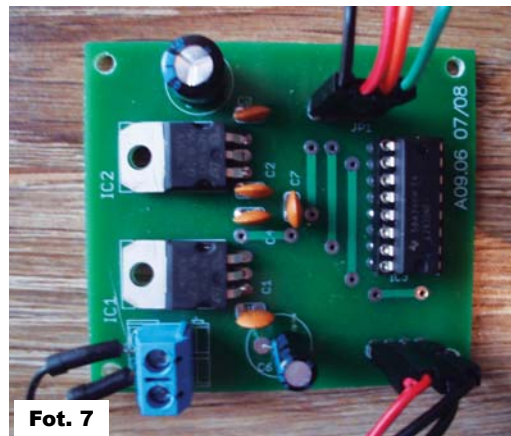
obsługę zegara RTC należy oprogramować samodzielnie. W sumie układ ulega uproszczeniu oraz maleją koszty (mikrokontroler kosztuje nieco ponad 20 zł). Dodatkowo, zaawansowane zasoby sprzętowe umożliwiają rozbudowę zegara według indywidualnych potrzeb, jedynie przez modyfikację oprogramowania. Schemat płyty głównej przedstawiony jest na rysunku 2. Rezystory R4 i R1 dają napięcie wyjściowe IC1 około 3,4V. Regulator jest w wersji SMD, gdyż dość mocno się nagrzewa. Dobrą modyfikacją byłoby zastosowanie regulatora impulsowego. Znaczna liczba kondensatorów ceramicznych 100nF jest wymagana przez kartę katalogową do stabilnej pracy mikrokontrolera. Linie P0-P13 portu PB podłączone są do złącza JP3 (IDC 2x8) i służą do sterowania anodami (PB10-PB13) i katodami (PB0-PB9) lamp Nixie w module wyświetlaczy. Połączenie pomiędzy tym modulem a płytą główną dokonane jest za pomocą przewodu IDC. Mikrokontroler steruje również w module 3 pracą silnika krokowego w połączeniu

bipolarnym (4 przewody) za pomocą linii PA0-PA3 i złącza JP2. Układ L272, podwójny wzmacniacz operacyjny o dużej (do 1A) wydajności prądowej, tworzy wzmacniacz głośnikowy (bez kondensatora wyjściowego). Minimalne napięcie pracy wzmacniacza to 4V, ale w tej aplikacji pracuje prawidłowo już przy napięciu 3,4V.

Do (wewnętrznie podciągniętych) linii PC4 i PC5 mikrokontrolera podłączone są przyciski służące do ustawiania godziny i wyboru trybu pracy (jakie dźwięki mają być generowane).

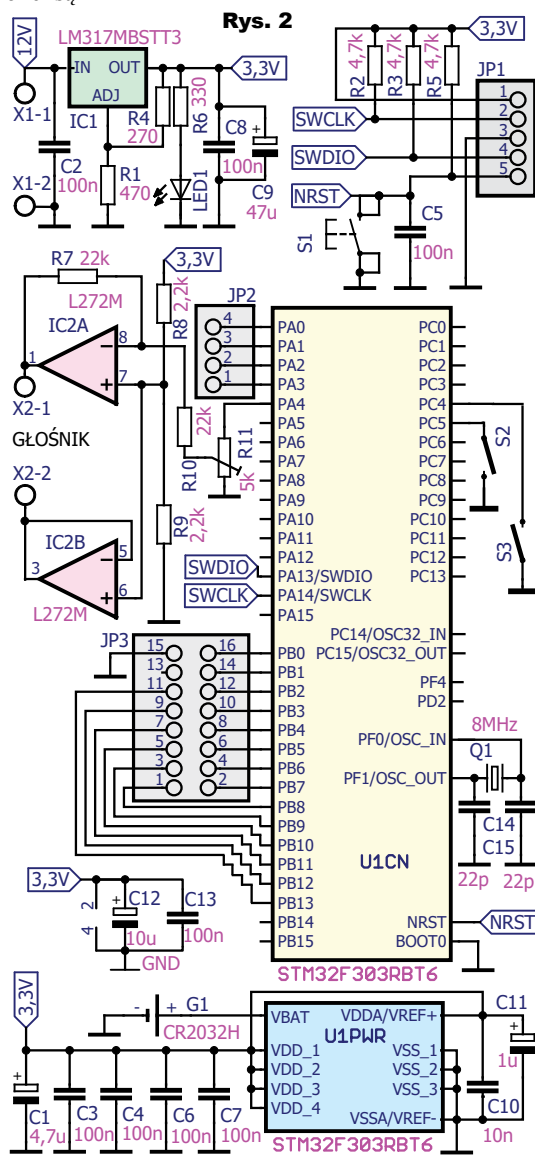
Złącze programatora JP1 korzysta z interfejsu SWD. Na płycie przewidziano miejsce na złącze baterii podtrzymującej działanie wewnętrznego zegara RTC, który jednak w podstawowej wersji nie jest wykorzystany.

Drugim z modułów zegara jest układ wyświetlacza. Jego schemat ideowy przedstawiony jest na rysunku 3. Płytkę zapro-



Fot. 7

jektowano w dwóch wersjach dla lamp IN-2 i IN-14. Wszystkie pozostałe elementy są identyczne. Zawiera on dwa zasadnicze elementy: przetwornicę napięcia generującą napięcie ok. 200V i układ tranzystorowy sterujący pracą lamp Nixie.



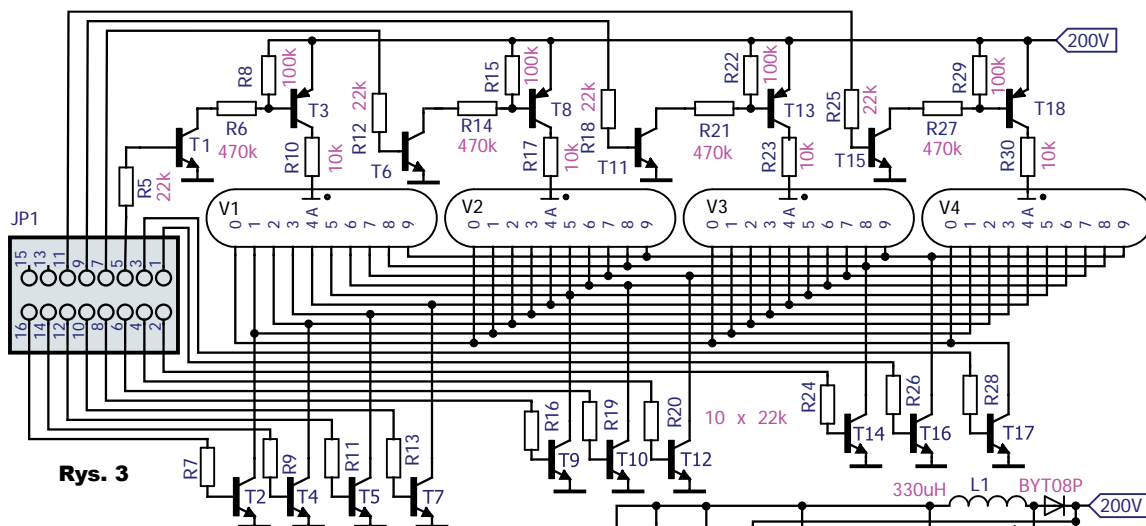
W przetwor-
nicy pracuje
MC34063 (opis
w EP 12/2003).

R2 reguluje
napięcie i jas-
ność świecenia,
a indukcyjność
cewki 330uH
nie jest kry-
tyczna. Cztery
lampy typu IN-2
wyświetlają
godziny i minu-
ty. Sterowanie
lampami jest
multipleksowe
za pomocą 10
„dolnych” tran-
zystorów do
sterowania katodami
oraz ośmiu „górnych”
tranzystorów do ster-
owania anodami. Rezystory
katodo-
we (R10, R17, R23,
R30) ograniczają prąd
do znamionowego
ok. 1,5mA. Zastoso-

wanie tanich i popularnych tranzysto-
rów wysokonapięciowych typu MPSA42
i MPSA92 pozwala uniknąć kłopotów
ze zdobyciem specjalizowanych układów
scalonych sterowników lamp Nixie.

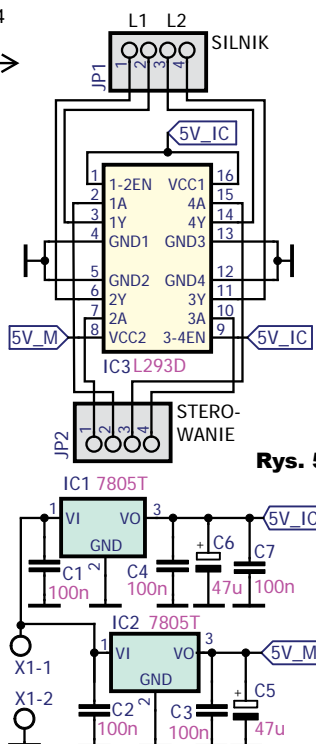
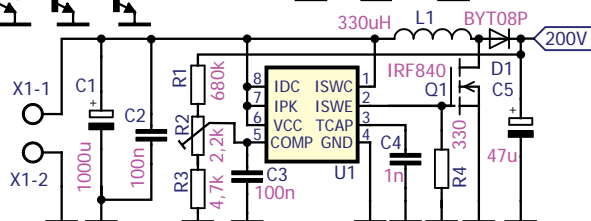
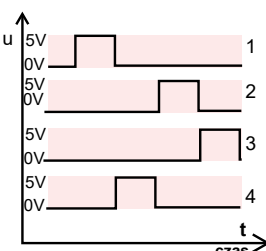
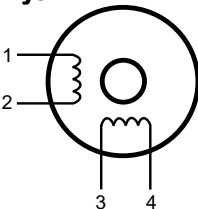
Ostatnim z modułów jest **układ wyko-
nawczy** zapewniający ruch (wsuwanie
i wysuwanie kukułki). Najprostszym
i najłatwiejszym do zdobycia wydaje się
układ przekładni ślimakowej z silnikiem
krokowym, stosowany w napędach stacji
dyskietek i napędach dysków optycz-
nych (CD/DVD) do przesuwania głowicy
magnetycznej w stacjach dyskietek
lub optycznej „lasera” w stacjach CD/
DVD. Umieszczenie na miejscu głowicy
kukułki lub innej figurki, umożliwia jego

przesuwanie w przód
i w tył w zakresie
kilku centymetrów.
Oczywiście ze wzglę-
du na małą moc silni-
ka, jej funkcję mogą
pełnić wyłącznie lek-
kie i małe przedmioty.
Na **rysunku 4** przed-
stawiono w sposób
poglądowy schemat
wewnętrzny silnika
krokowego i sekwen-
cję sterującą. Podanie
na cewki silnika peł-
niej sekwencji powo-
duje wykonanie czte-
rech kroków. Podanie



Rys. 3

Rys. 4



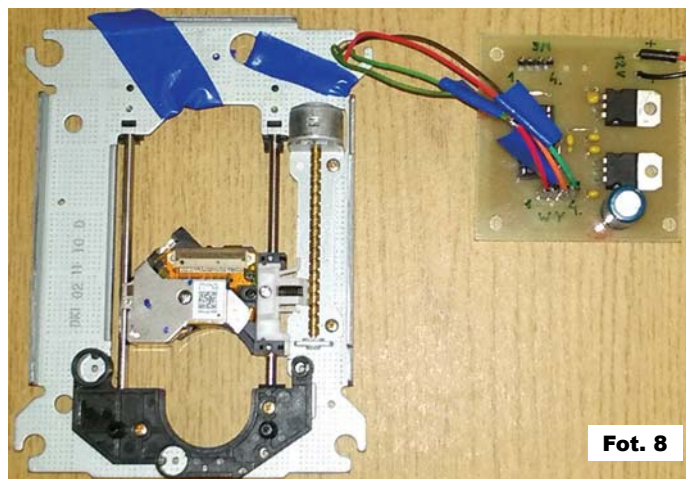
Rys. 5

kolejnych impulsów
w odwrotnej kolejności spo-
woduje obracanie się wału
w drugą stronę. Sterowanie
bipolarne wymaga zastoso-
wania dwóch mostków typu
H. Wykorzystano popularny
układ L293D (cena około
7zł). Bardzo prosty schemat
ideowy zaprezentowany jest
na **rysunku 5**. Karta katalo-
gowa zaleca stosowanie nie-
zależnego zasilania samego
układu scalonego oraz uzwo-
jeń silnika, stąd obecność
dwóch regulatorów napięcia.
Zdecydowana większość sil-
ników w napędach dyskie-
tek i dysków optycznych
wymaga napięcia zasilają-
cego o wartości 5V. Diody
zabezpieczające przed samo-
indukcją uzwojeń są zawarte
wewnątrz układu scalonego
L293D (uwaga, starszy L293
nie ma w swej strukturze
diod zabezpieczających). Na

fotografii 8 przedstawiono
wymontowaną z napędu CD-ROM
przekładnię, wraz z podłąco-
ną do silnika pierwotną wersją płytki
wykonawczej.

W drugiej części artykułu opisa-
ne zostaną montaż i uruchomienie
oraz podane będą dodatkowe infor-
macje o programie.

Michał Raczyński
rm23892@zut.edu.pl



Fot. 8



Zegar Nixie z „kukułką”

Zegar Nixie, tykający jak klasyczny zegar mechaniczny, wyposażony w kukulkę lub inne zwierzątko sygnalizujące pełną godzinę. Modułowa budowa ułatwia dostosowanie zegara do własnych upodobań.

W pierwszej części artykułu przedstawione zostały trzy moduły do prezentowanego zegara. Oto dalsze informacje.

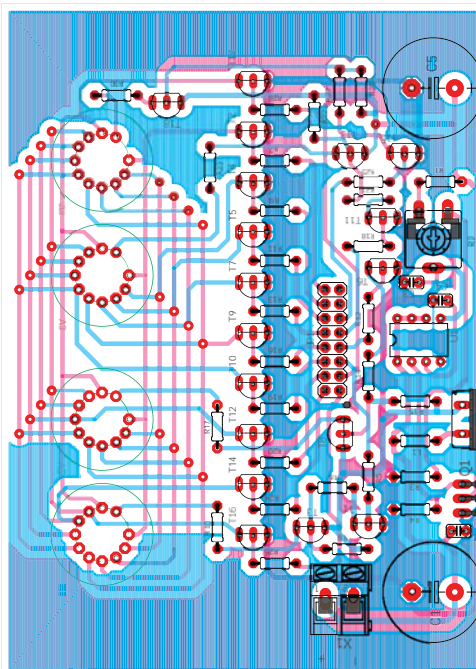
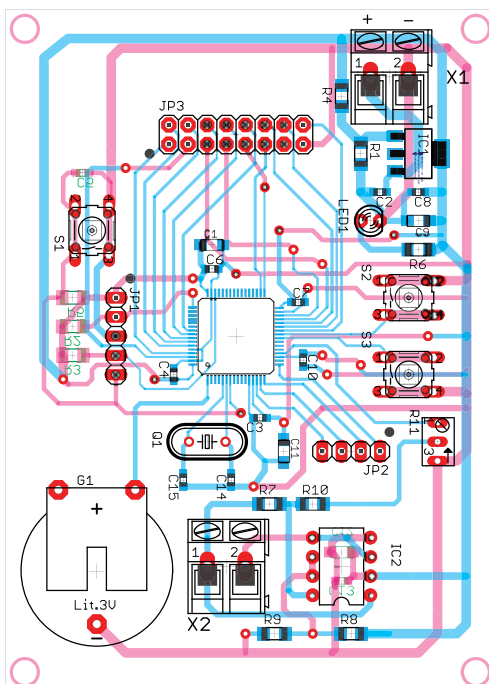
Montaż i uruchomienie

Projekty płytek drukowanych płyty głównej, wyświetlacza i układu wykonawczego pokazane są odpowiednio na **rysunkach 6, 7, 8, 9**. Montaż proponuję zacząć od płytki wyświetlacza (rysunek 7 i 8), standardowo zaczynając od elementów najmniejszych i po podłączeniu napięcia 9–12V sprawdzić za pomocą multimetru, czy na wyjściu jest obecne wysokie napięcie (160–200V). Cewkę wykonano samodzielnie, z wykorzystaniem pręciaka ferrytowego z zasilacza komputerowego i mieści się ona bez problemów na płycie PCB. W przypadku fabrycznych cewek ilość miejsca może być za mała i wtedy można spróbować podłączyć cewkę

za pomocą przewodów, choć nie jest to wskazane. Sygnałem wskazującym na pracę przetwornicy może być także cichy pisk magnetostrykcyjny cewki L1. Oczywiście przy pracującej przetwornicy należy zachować ostrożność, gdyż dotknięcie do elementów lub punktów lutowniczych będących pod napięciem około 200V na pewno nie jest przyjemne, a nawet może być niebezpieczne. Jeżeli przetwornica działa prawidłowo, należy wlotować elementy układu sterowania – jako ostatnie montujemy lampy Nixie – w przypadku lamp IN-2 po przeciwnej stronie niż reszta elementów, a w przypadku lamp IN-14 po tej samej stronie co pozostałe podzespoły (możliwe jest także zamontowanie ich po stronie przeciwnej, ale wtedy należy zmodyfikować fragment programu odpowiadający

Rys. 6 za multipleksowanie).

Rys. 7 Skala 75%



tość napięcia jest prawidłowa (ok. 3,4V). Jeżeli tak jest, to montujemy resztę elementów, na koniec zostawiając mikrokontroler. Tu wskazówka dla niezaawansowanych: po ustawieniu mikrokontrolera we właściwej pozycji lutujemy dwa dowolne wyprowadzenia „po przekątnej”, aby wstępnie unieruchomić układ scalony. Następnie pokrywamy wszystkie wyprowadzenia topnikiem aktywnym. Na grot nakładamy trochę cyny i przeciągamy go po wyprowadzeniach z jednej strony układu. Nadmiar cyny pozostały na pinach ściągamy za pomocą plecionki lutowniczej. Działania te powtarzamy z pozostałych trzech stron układu. Po nałożeniu cyny na grot należy działać szybko, aby spoiwo się nie przegrzało.



część 2

kit

3213

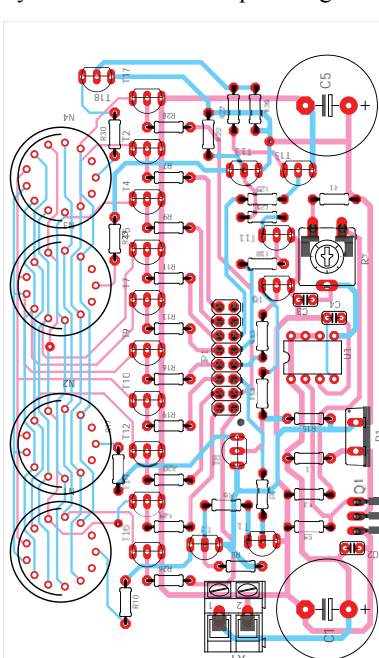
AVT

Trzecia płytki, wykonawcza (rysunek 9) jest zdecydowanie najprostsza. Należy jedynie pamiętać, aby zworkę pod układem scalonym przylutować przed wlutowaniem podstawki. Regulatory napięcia w zasadzie nie potrzebują radiatorów. Jedynym problemem jest lokalizacja uzwojeń w silniku krokowym. Za pomocą omomierza lokalizujemy końcówki jednego uzwojenia. Uzwojenia podłączamy pomiędzy zaciski 1 i 2 oraz 3 i 4 złącza JP1. Nie jest jednak możliwe rozróżnienie między sobą początku i końca uzwojenia. Należy je podłączyć „na chybił trafił”, a po wgraniu programu ewentualnie zamienić zaciski jednej cewki.

Kiedy wszystkie moduły są gotowe, łączymy je ze sobą. Aby ułatwić lokalizację wyprowadzeń, na płytce, przy złączach goldpin, przy pinie nr 1 naniesiono kropkę (warstwa tDocu).

Pozostaje jeszcze wgranie programu do mikrokontrolera. Należy tego dokonać za pomocą zewnętrznego programatora np. st-link, który dostępny jest jako osobne urządzenie lub jako część popularnych płytek ewaluacyjnych (np. seria Discovery lub Nucleo). Darmowy program, dostępny w Internecie, umożliwiający zaprogramowanie układu to

„ST-LINK Utility”. Program i sterowniki możemy pobrać ze strony producenta (konieczne jest podanie swojego adresu e-mail): <http://www.st.com/en/development-tools/st-link-v2.html>. Programowanie odbywa się za pomocą interfejsu SWD, który wykorzystuje trzy linie sygnałowe: SWCLK, SWDIO, NRST oraz linię zasilania 3,3V i masy. Na płytce głównej poszczególne piny złącza JP1 to: 1 – 3,3V, 2 – SWCLK, 3 – GND, 4 – SWDIO, 5 – NRST. Wyprowadzenia te łączymy przewodami z programatorem. W zależności od tego, czy jest on osobnym urządzeniem, czy częścią płytki ewaluacyjnej, musimy w Internecie odnaleźć dokumentację urządzenia i zlokalizować poszczególne linie.

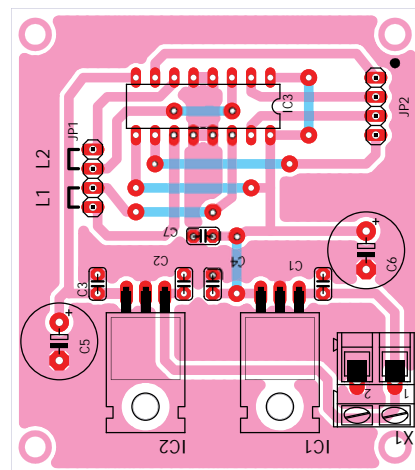


Rys. 8 Skala 75%

Po zainstalowaniu sterowników programatora i programu ST-LINK Utility, podłączamy programator do gniazda USB komputera (jeżeli podłączyliśmy wszystkie pięć linii, czyli także linię zasilania 3,3V, co powinno objawiać się świeceniem diody LED 1 na płytce głównej, to nie jest konieczne dołączanie zewnętrznego zasilania płytki głównej podczas programowania).

Uruchamiamy program ST-LINK Utility, otwieramy odpowiedni plik wynikowy (File→Open File), następnie sprawdzamy, czy programator wykrywa mikrokontroler, klikając: Target→Connect. Jeżeli

nastąpiło połączenie, to wgrywamy plik, klikając Target→Program, Start. Po wgraniu klikamy Target→Disconnect i możemy odłączyć programator. Aby maksymalnie ułatwić uruchomienie zegara, bez konieczności modyfikacji kodu źródłowego, przygotowano aż osiem plików wynikowych, różniących się między sobą trzema parametrami: 1. Rodzaj płytki wyświetlacza (IN-2 lub IN-14) 2. Rodzaj mechanizmu napędowego (CD-ROM lub floppy – różnią się liczbą wykonywanych kroków), 3. Rodzaj dźwięku „kukułki” (kukułka lub owca).



Rys. 9

Gotowe do programowania pliki wynikowe dostępne są w Elportalu wśród materiałów dodatkowych do tego numeru EdW, wraz z projektami płytek, filmem i innymi plikami. Można użyć tego „gotowca”, ale być może celowa i potrzebna będzie modyfikacja programu źródłowego, który zapewnia podstawową funkcjonalność zegara:

- odczytywanie i wyświetlanie czasu;
- nastawianie czasu: jeden przycisk – minuty, drugi – godziny;
- wybór trybu dźwiękowego – zmiana przez jednoczesne naciśnięcie obydwu przycisków. Oto tryby dźwiękowe:
 - a) tryb domyślny (po załączeniu zasilania) – dźwięki kukułki i tykania wyłączzone
 - b) tryb cichy – wszystkie dźwięki i wysuwanie kukułki wyłączone
 - c) tryb tykania – tylko dźwięk tykania, wysuwanie kukułki i kukanie wyłączone
 - d) tryb kukułki – tylko sygnalizacja pełnych godzin przez kukułkę – tykanie zegara wyłączone

W obecnej wersji nie ma zasilania awaryjnego. „Kukułka” w nocy śpi (fotografia 9) i pojawia się o pełnych godzinach od 8 do 22. Wartości te można zmienić, modyfikując program. Zmiany mogą też dotyczyć liczby kroków wykonywanych przez silnik krokowy podczas wysuwania i wsuwania kukułki oraz dźwięków generowanych przez urządzenie.

W trzeciej części artykułu podane zostaną dodatkowe informacje o wykorzystanym oprogramowaniu oraz o kwestiach mechanicznych.

Michał Raczyński
rm23892@zut.edu.pl



Fot. 9



Zegar Nixie z „kukulką”

Zegar Nixie, tykający jak klasyczny zegar mechaniczny, wyposażony w kukulkę lub inne zwierzątko sygnalizujące pełną godzinę. Modułowa budowa ułatwia dostosowanie zegara do własnych upodobań.

W dotychczas przedstawionych dwóch częściach artykułu opisane były elektroniczne aspekty projektu.

Całość projektu została wykonana w darmowym środowisku CoIde CoCox (v 1.7.8). Można je pobrać ze strony: <http://www.cocox.org/>, a dodatkowo należy ściągnąć pliki kompilatora GCC np, z adresu: <https://developer.arm.com/open-source/gnu-toolchain/gnu-rm/downloads>. Po zainstalowaniu oprogramowania uruchamiamy środowisko Coocox CoIde. Możemy wybrać lokalizację, gdzie będzie znajdował się nasz projekt. Następnie musimy ustawić ścieżkę do wcześniej zainstalowanego kompilatora. W tym celu klikamy: „Project”, a następnie „Select Toolchain Path” i wpisujemy przykładowo:

„E:\GNU Tools ARM Embedded\6 2017-q2-update\bin”, w zależności od tego, gdzie mamy zainstalowany kompilator. Teraz przystępujemy do utworzenia nowego projektu. Klikamy Project → New Project i wpisujemy nazwę np. „zegar_3”, klikamy „next”, wybieramy w kolejnej zakładce „chip”, znowu „next”, wybieramy nasz mikrokontroler, czyli w zakładce „ST” szukamy STM32F303VB i klikamy „finish”. Następnie wybieramy zakładkę „Configuration” (jeżeli jej nie widać, to klikamy „View”, i „Configuration”) i zakładkę „Link”. W polu „Linked Libraries” ustawiamy ścieżki do plików „libgcc” i „libm”(przycisk „Add”), przykładowo:

„E:\GNU Tools ARM Embedded\6 2017-q2-update\lib\gcc\arm-none-eabi\6.3.1\libgcc.a” oraz:

„E:\GNU Tools ARM Embedded\6 2017-q2-update\arm-none-eabi\lib\libm.a”

Z kolei w zakładce „compile” zmieniamy ustawienie z „Not use FPU” na „FPU_hard”.

Teraz w zakładce „Repository” zaznaczamy pola przy następujących peryferiach: RCC, GPIO, DAC, TIM, MISC. Z drzewa projektu po lewej stronie wybieramy i otwieramy plik „main.c”, a następnie usuwamy całą jego zawartość i wklejamy zawartość pliku „main.c” przygotowanego w folderze „plik_źródłowy”. Zawarte tam pozostałe trzy pliki (beep.c, kukulka.c, tick.c) kopiujemy do folderu, gdzie znajduje się nasz nowy plik „main.c”.

Po tych zabiegach kompilacja powinna przebiegać prawidłowo. Aby to sprawdzić, klikamy „Project”, a następnie „Rebuild” i w konsoli sprawdzamy, czy kompilacja zakończyła się sukcesem. Pliki wynikowe znajdują się w lokalizacji: „zegar_3\zegar_3\Debug\bin”. Możemy pliki te wgrywać za pomocą wspomnianego wcześniej programu ST LINK Utility, ale możliwe jest także programowanie z poziomu CoCoxa („Flash→Program Download”), możliwe jest także debugowanie. W przypadku błędów podczas prób programowania należy spróbować podmienić plik „STLinkUSBDriver.dll” znajdujący się w lokalizacji: „CooCox\CoIDE\bin” na zawarty w folderze „ST_LINK_driver” (w materiałach do tego numeru). Teraz możemy przystąpić do modyfikacji programu.

W przypadku, gdy po uruchomieniu oryginalnego programu okaże się, że liczba kroków wykonywanych przez silnik krokowy jest za duża lub za mała, należy zmodyfikować deklarację „zakres_wysuniecie” o domyślnej wartości 42. Druga modyfikacja może dotyczyć zmiany dźwięków wydawanych przez zegar i „kukulkę”. Dyskretne wartości sygnałów dźwiękowych są zawarte w jednomiarowych tablicach: „beep” i „tick”, umieszczonych w plikach o takich samych nazwach. W oryginalnej wersji zegara



„kukulka” jest... owcą, więc i wydawany przez nią dźwięk to beczenie. Aby zmienić rodzaj wydawanego dźwięku, należy zastąpić wspomniane wcześniej pliki innymi. W folderze zawierającym plik „main.c” znajduje się dodatkowo plik „kukulka.c” zawierający tablicę z dyskretnymi wartościami dźwięków kukania i bicia dzwonu. Jeżeli chcemy go wykorzystać, to zmieniamy instrukcję „include” w pliku main.c oraz częstotliwość próbkowania i nazwę wykorzystywanej przez przetwornik DAC zmiennej – jest to opisane w komentarzach w kodzie programu. Jeżeli ktoś chce wykorzystać jeszcze inne dźwięki, to musi najpierw znaleźć lub nagrać odpowiedni plik wav (dużo plików, w tym wykorzystane w oryginalnej wersji zegara, można znaleźć na stronie: www.findsounds.com), a następnie przekonwertować go do postaci tablicy. Pamiętać należy o kilku sprawach. Najważniejsze parametry pliku dźwiękowego w postaci cyfrowej to częstotliwość próbkowania i rozdzielczość. Przykładowo plik „kukulka” zapisany jest z częstotliwością próbkowania 22,050kHz i rozdzielczością 8 bitów. Należy jednak pamiętać, że im plik jest zapisany z wyższą częstotliwością próbkowania, tym jego objętość jest większa, a dostępna ilość pamięci Flash w mikrokontrolerze to 128kB. Rozdzielczość przetwornika DAC zawartego w układzie STM32F303 to 12 bitów. Można wykorzystać pliki zapisane z większą rozdzielczością, ale wtedy obniżeniu ulegnie ich jakość, a także pliki o niższej rozdzielczości – wtedy nie ulegnie ona zmianie. Dźwięki tykania zegara



Fot. 10

i wydawane przez kukulkę nie muszą być zapisane z taką samą częstotliwością próbkowania. Szczegóły dotyczące jej ustawiania zawarte są w komentarzach w listingu programu. W folderze ...**obrobka plików** znajduje się plik „wav2hex.m” konwertujący plik typu „wav” na plik zawierający tablicę, gotową do wykorzystania w programie mikrokontrolera. Uwaga! Jest on napisany w programie Matlab, który jest trudno dostępny dla większości użytkowników (choć można poszukać darmowej, ograniczonej czasowo wersji). Napisany program najpierw wczytuje plik typu „wav”, odczytuje częstotliwość próbkowania i rozdzielczość, z jaką został zapisany (w bitach), zapisuje te dane w zmiennych: „y” (wektor próbek), „fs” (częstotliwość próbkowania), „bit” (liczba bitów na próbkę). Następnie skaluje odpowiednio wartości (typowy plik „wav” przyjmuje wartości ułamkowe od -1 do 1, a my potrzebujemy, aby wartości zmieniały się od 0 do 4095,



Fot. 11

co zapewnia maksymalne wykorzystanie rozdzielczości przetwornika) i zapisuje w postaci tablicy. Wzorując się na tym programie, można napisać własną aplikację konwertera np. w języku C.

Inne możliwości modyfikacji programu to np. godziny, w których kukulka wyśpiewuje godzinę oraz częstotliwość z jaką zegar wydaje dźwięki tyknięcia. Wykorzystanie wewnętrznego zegara RTC do odmierzania czasu i tym samym zabezpieczenie przed zanikiem zasilania, wymaga dość istotnej modyfikacji programu, gdyż w obecnej wersji za odmierzanie czasu odpowiadają timery, a po zaniku zasilania następuje reset mikrokontrolera. W związku z tym baterijka nie pełni żadnej funkcji i jeżeli nie chcemy dodawać obsługi RTC, to nie należy jej montować.

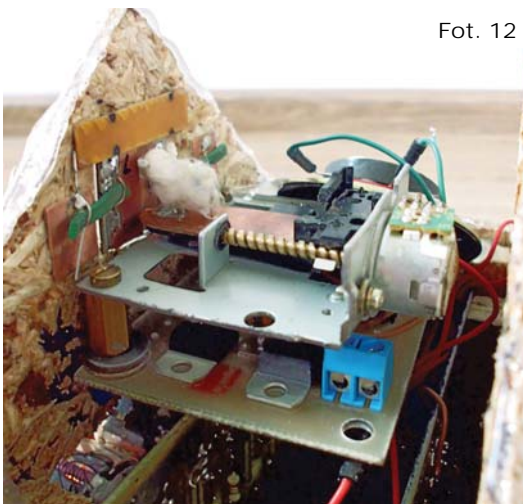
Nie mniej ważną sprawą są kwestie wykonania obudowy i mechanicznego montażu elementów zegara np. drzwiczek kukulki. W wersji pokazanej na **fotografii 10** obudowa wykonana jest z płyty paździerzowej, daszek z panelu podłogowego, drzwiczki z cienkiego laminatu z przylutowanymi „zawiasami”, wykonanymi z obciętych końcówek elementów, zamykają się samoczynnie dzięki gumce recepturce. Z kolei owieczka wykonana została z... kondensatora elektrolitycznego z przylutowaną diodą LED 3mm, pokrytych watą. Pokazują to **fotografie 11...13**. Przyciski sterujące i potencjometr siły głosu umieszczono z tyłu urządzenia.

Na koniec wskazówki dla tych, którzy chcieliby płytki drukowane wykonać samodzielnie: dwie z trzech płytek drukowanych wykonane są jako dwuwarstwowe, zatem ich zrobienie może sprawić problem początkującym. Szczególnie trudna może okazać się płyta główna, która zawiera mikrokontroler

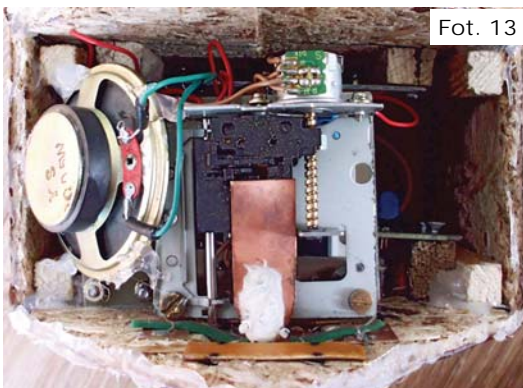


w obudowie TQFP64 i musi być wykonana precyzyjnie. Jest to jednak możliwe w warunkach domowych – amatorskich, metodą termotransferową („żelazkową”) i bardzo niskim kosztem. Taki też sposób wykorzystałem przy konstrukcji pierwszej wersji urządzenia. W tym celu zresztą zastosowane zostały spore otwory na przelotki, aby można je było wykonać standardowym wiertłem bez obawy uszkodzenia miedzi.

Ponieważ tematem artykułu nie jest metodyka wykonywania płytek drukowanych, ograniczę się do kilku wskazówek co do wykonywania płytek dwustronnych. Należy zaopatrzyć się w papier kredowy (możliwie najcieńszy) i wydrukować na dwóch kartkach warstwy „top” i „bottom”. Następnie należy zrobić z wydrukowanych stron „kopertę”, do której włożymy laminat. W tym celu jedną kartkę odkładamy na bok, a drugą docinamy do rozmiarów wydrukowanego obszaru, ale dodajemy margines ok. 5mm z każdego boku. Następnie musimy idealnie spasować dwie kartki (tonerem do środka), można to zrobić pod światło, np. na szybie oświetlonej od drugiej strony. Dopasowanie najlepiej poznać po pokryciu się otworów przelotek leżących w sporej odległości od siebie. Po spasowaniu należy dociętą kartkę przykleić na krawędziach taśmą klejącą do kartki niedociętej, pozostawiając jeden bok nieprzyklejony. Następnie docinamy laminat dwustronny do wymiarów nieco większych niż zaprojektowana płytka (ok. 3mm dodajemy do każdego boku), czyszcimy, odtłuszczamy i wkładamy do koperty. Odprasowywanie polecam wykonać tradycyjnie żelazkiem, a nie za pomocą laminatora, gdyż ruch wałków może spowodować wzajemne przesunięcie wydruków. Przed przystąpieniem do wytrawiania należy sprawdzić prawidłowość przeniesienia ścieżek i przewiercić co najmniej dwie przelotki położone możliwie daleko od siebie. Jeżeli z obydwu



Fot. 12



Fot. 13

Wykaz elementów

Płyta główna:

wszystkie rezystory SMD o wymiarze 1206

R1	470Ω
R2,R3,R5	4,7kΩ
R4	270Ω
R6	330Ω
R7, R10	22kΩ
R8,R9	2,2kΩ
R11	potencjometr 5kΩ (może być o zbliżonej wartości)
C1	4,7u/6,3V (SMD, obudowa A)
C2,C3,C4,C5,C6,C7,C8,C13	100n (SMD, obudowa 0603)
C9	47u/6,3V (SMD, obudowa A)
C10	10n (SMD, obudowa 0603)
C11	1u/6,3V (SMD, obudowa A)
C12	10u/6,3V (SMD, obudowa A)
C14,C15	22p (SMD, obudowa 0603)
U1	STM32F303RBT6
IC1	LM317 (obudowa SOT223)
IC2	L272
LED1	dioda LED 3mm
S1,S2,S3	przyciski

(przycisku S1 „RESET” nie trzeba koniecznie montować)
 Q1 kwarc 8MHz
 X1,X2 złącze ARK podwójne
 JP1 listwa goldpin 1x4
 JP2 listwa goldpin 2x3
 JP3 listwa goldpin 2x8
 G1 złącze baterii CR2032
 Uwaga, przyciski i potencjometr można montować nie bezpośrednio na płycie, ale za pomocą przewodów.

Płyta wyświetlacza:

wszystkie rezystory 0,125W

R1	680kΩ
R2	2,2kΩ
R3	4,7kΩ
R4	330Ω
R5,R7,R9,R11,R12,R13,R16,R18,R19,R20,R24,R25,R26,R28	22kΩ
R6,R14,R21,R27	470kΩ
R8,R15,R22,R29	100kΩ
R10,R17,R23,R30	10kΩ
C1	1000u/25V
C2,C3	100n
C4	1n

C5	47u /350V
T1,T2,T4,T5,T6,T7,T9,T10,T11,T12,T14,T15,T16,T17	MPSA42
T3,T8,T13,T18	MPSA92
Q1	IRF840
U1	MC34063
D1	BYV29/400 (lub podobna)
JP1	listwa goldpin 2x8
X1	złącze ARK podwójne
L1	cewka 300uH, 1,5A
V1,V2,V3,V4	lampa Nixie IN-2
N1,N2,N3,N4	lampa Nixie IN-14

Płyta wykonawcza sterująca silnikiem krokowym:

C1,C2,C3,C4,C7	100n
C5,C6	47u/16V
IC1,IC2	7805
IC3	L293D
JP1,JP2	listwa goldpin 1x4
X1	złącze ARK podwójne

**Płytką drukowaną jest dostępna
w Sklepie AVT jako AVT3213**

stron otwór pojawił się w odpowiednim miejscu, to znaczy, że ścieżki są przeniesione prawidłowo. W przeciwnym wypadku należy całą procedurę powtórzyć.

Początkującym w pierwszej kolejności polecam wykonać płytkę modułu wyświetlacza, ponieważ jest ona prostsza i można nabrać nieco wprawy. Po wytrawieniu płytkę należy pokryć kalafonią rozpuszczoną w denaturacie (lub

pastą lutowniczą), a następnie pocynować i za pomocą rozcieńczalnika usunąć pozostałości kalafonii. W tak przygotowanej płytce trzeba wykonać w pierwszej kolejności wszystkie przelotki. Można wykorzystać w tym celu srebrzanek o średnicy 0,6mm, obcięte końcówki elementów elektronicz-



nych (niestety pasują tylko te najcieńsze) lub dowolny inny rodzaj drutu. Lepiej jest najpierw przylutować przelotkę z obydwu stron, a dopiero później dociąć drukik.

Michał Raczyński
rm23892@zut.edu.pl

R E K L A M A

EDW A10 Elektronika dla juniora (i seniora) - zestaw elementów do kursu EdW2016

W Elektronice dla Wszystkich 1/2016, w związku z jubileuszem 20-lecia czasopisma rozpoczęliśmy elementarny kurs podstaw elektroniki dla najmłodszych i starszych, którzy chcieliby przypomnieć sobie podstawy. Założeniem jest, że młodziutki „uczeń” nie zostanie pozostawiony sam sobie, bo proponowane ćwiczenia ma wykonywać z kimś choć troszkę starszym i doświadczonym.

Oprócz rodziców może to być starsze rodzeństwo albo ktoś z rodziny lub przyjaciół.

kod handlowy:

EDWA10

45 zł

POBIERZ KURS

