



Gadająca kostka



Do czego to służy?

Jest to elektroniczna kostka do gry. Służy dokładnie do tego, do czego służy zwykła kostka sześciocienna, tylko, że... gada. Nie dość, że gada to jeszcze odtwarza dźwięki mieszania oraz rzutu kostką. Wszystkie te dźwięki należy nagrać samemu i gdy tylko się znużą można je zmienić.

Kostka ta może się przydać w różnych sytuacjach. Wyobraźmy sobie taki przykład: znajdujemy się w Egipcie, w piramidzie. Nagle gaśnie latarka i zapadają egipskie ciemności, a my akurat teraz mamy wielką ochotę zagrać w jakąś grę planszową. Wszystko jest OK, ale nie widać kostki! W takiej sytuacji uratuje nas kostka, którą słyszać.

Jak to działa?

Schemat ideowy przedstawiony został na rysunku 1. Układ nie należy do skomplikowanych.

Jest to typowa aplikacja magnetofonu cyfrowego ISD1420 (ewentualnie ISD1416) pracującego w trybie operacyjnym. Magnetofonem steruje mikrokontroler AVR AT90S2313. Element ten wybrano z kilku powodów:

- umożliwia on wyjście z trybu uśpienia dzięki przerwaniu zewnętrznemu,
- znikomemu pobór mocy przez procesor oraz ISD w uśpieniu (dzięki temu można zrezygnować z wyłącznika zasilania),
- wewnętrzny układ Power On Reset pozwala na rezygnację z kilku elementów (zmniejszenie układu),
- bardzo ważna sprawa: pakiet BASCOM AVR niewiele różni się od BASCOM 8051, więc nie ma większych problemów z napisaniem programu.

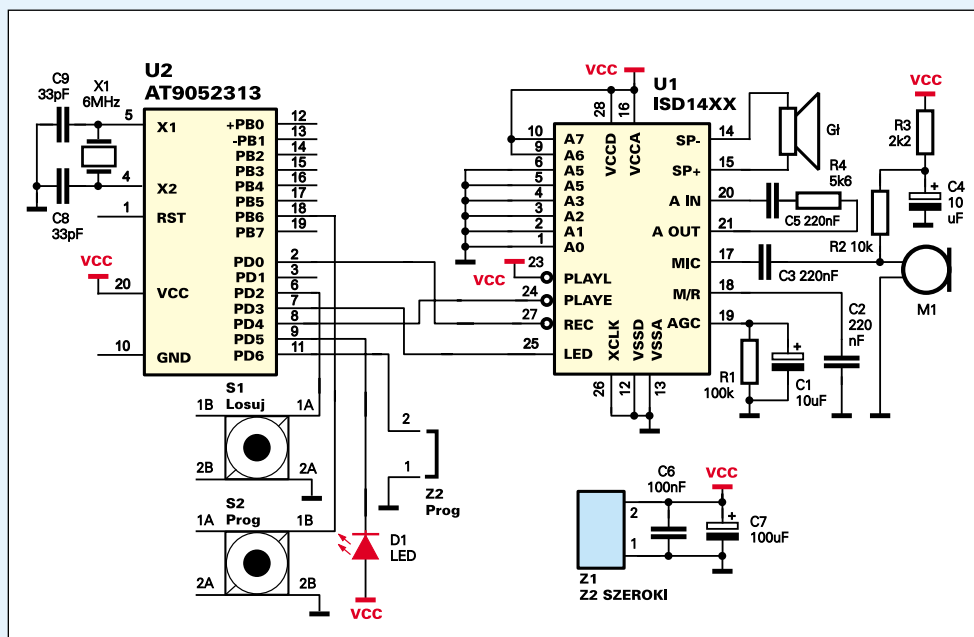
Układ wykorzystuje dziewięć aktywnych wyprowadzeń mikrokontrolera. Pięć z nich steruje procesorem

dźwięku. Do jego sterowania wykorzystano następujące sygnały: *PLAYE* – zbocze opadające na tym wyprowadzeniu inicjuje odtwarzanie; *REC* – stan niski jest warunkiem nagrywania komunikatu; *A0* – gdy na tej końcówce znajduje się stan *H* działanie układu ISD jest przyspieszone około 800 razy, umożliwia to szybkie znalezienie odpowiedniego komunikatu, jeśli znamy tylko kolejność w jakiej wszystkie komunikaty są ułożone; *A4* – gdy *H* - każde odtwarzanie zaczyna się w miejscu gdzie zakończyło się poprzednie; w przeciwnym razie odtwarzanie zaczyna się na początku pamięci; *LED* – to wyjście ma podwójną funkcję: w trybie odtwarzania pojawiają się tutaj ujemne impulsy za każdym razem, gdy dojdziemy do końca aktualnie odtwarzanego komunikatu; w trybie odtwarzania występuje tutaj stan niski,

a pojawienie się stanu wysokiego świadczy o przepełnieniu pamięci.

Na schemacie znajdują się jeszcze dwa przyciski: *Losuj* – w trybie losowania przycisk ten budzi procesor z trybu uśpienia i inicjuje losowanie. Po jego naciśnięciu odtwarzany jest dźwięk losowania. Po puszczeniu kończy się losowanie, odtwarzany jest jeden z dźwięków toczącej się kostki a na końcu słowo określające ilość oczek. W trybie nagrywania jego krótkie naciśnięcie odtwarza aktualnie obrabianą próbkę dźwięku. Jego przytrzymanie dłużej niż sekundę powoduje zatwierdzenie dźwięku i przejście do następnej próbki, co jest sygnalizowane mignięciem diody. Przycisk *Prog* – ma znaczenie tylko w trybie programowania. Jest dostępny

Rys. 1 Schemat ideowy



dopiero po otwarciu obudowy. Jego naciśnięcie powoduje rozpoczęcie nagrywania dźwięku na aktualnej pozycji. Nagrywanie jest sygnalizowane świeceniem diody.

Zwora „Prog.” normalnie powinna być zwarta. Włączenie zasilania przy rozwartej zworze powoduje wejście w tryb programowania, o czym poinformuje nas trzykrotne miganie diody LED.

To tyle części sprzętowej. Cała reszta kryje się w oprogramowaniu. W dalszej części artykułu nie będzie prezentowania kodu źródłowego. Z grubsza opisano tylko zasadę działania programu.

Po starcie i po inicjacji zmiennych, które tego wymagają, sprawdzany jest stan zworki i podejmowana jest decyzja w jakim trybie uruchomić kostkę. Tryb programowania jest względnie łatwy do analizy. Tryb losowania natomiast jest trudniejszy ze względu na wykorzystanie kilku przerwań, bez których niektóre części kodu nie mają sensu. Poniżej wyjaśniono tylko część kodu, pracującą podczas losowania.

Na początku aktywowane jest przerwanie *Int0* i ustawiane jest do reagowania na poziom niski. Po tej czynności procesor jest usypiany. Z uspienia, według dokumentacji, można go wybudzić poprzez: przerwanie *watchdog*, przerwanie zewnętrzne ustawione na *wyzwalanie poziomem* lub poprzez *reset*. Układ *watchdog* jest wyłączany już na starcie programu, wejście *reset* nie jest w ogóle podłączone. Wykorzystujemy więc drugą możliwość – do wejścia *Int0* podłączony jest przycisk *Losuj*. Jego naciśnięcie spowoduje przejście do obsługi przerwania, gdzie w tym momencie nic się nie dzieje. Program od razu opuszcza podprogram, a cała „akcja” spoczywa na pętli głównej. Teraz ustawiona zostaje zmienna *bit_loswanie*, *Timer0* zostaje uruchomiony – dzięki niemu odbywa się losowanie, *Int0* przestawione zostaje tak, aby następna reakcja nastąpiła przy puszczeniu przycisku – zdarzenie to zostanie obsługane w procedurze przerwania dzięki ustawionej zmiennej *bit_loswanie*. Po odtworzeniu dźwięku do mieszania program oczekuje na zwolnienie przycisku, przy czym jeśli stało się to wcześniej natychmiast odtwarzany jest jeden z dźwięków upadającej kostki a następnie słowo określające ilość oczek, która „wypadła”.

Komentarza wymaga jeszcze zakres losowanej liczby, który wynosi 0 – 23. Najmłodszy bit decyduje o tym, który dźwięk stukania będzie odtwarzany. Po podzieleniu przez dwa otrzymujemy numer odtwarzanego komunikatu. Oznacza to, że komunikatów jest 12. Dlaczego? Żeby było ciekawiej: dużo tłumaczy **tabela 1**, w której przedstawione jest to, co powinno znajdować się w pamięci ISD. Powtarzające się numery oczek powodują, że raz usłyszymy na przykład „dwa oczka”, kiedy indziej „dwa”.

tabela1

0	„jeden“
1	„jedyńka“
2	„dwa oczka“
3	„dwójka“
4	„trójka“
5	„trzy oczka“
6	„czwóreczka“
7	„cztery“
8	„pięć“
9	„piątka“
10	„sześć oczek“
11	„szucha“
12	Mieszanie
13	Stuk 1
14	Stuk 2

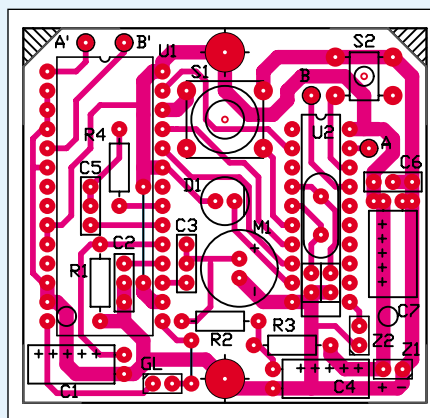
W **tabeli 1** wątpliwość może wzbudzić rozmieszczenie komunikatów – na końcu umieszczone są dźwięki odtwarzane na początku. Powoduje to nieco zakłóceń przez przedłużenie czasu przewijania. Zwiększa jednak wygodę programowania. Po słowach, raczej standardowej długości, możemy eksperymentować z różnymi wersjami dźwięków mieszania oraz toczących się kostek.

Montaż i uruchomienie

Płytką drukowaną jest widoczna na **rysunku 3**. Została ona przystosowana do obudowy KM-22. Obudowa KM-22 składa się z dwóch identycznych części. Jedną z nich potraktujemy jako „część górną”, drugą jako „dolną”. Z górnej części należy usunąć bolce podtrzymujące płytkę oraz miejsca na śruby. Następnie wykonujemy otwory pod przycisk *Losowania* i pod diodę. Można w tym celu posłużyć się płytką drukowaną jako wzorem.

Montaż płytki drukowanej nie jest typowy. Na płytce zamontowano niektóre z elementów między nóżkami scalaków. Podstawki pod układy scalone będą konieczne, trzeba jednak je odpowiednio przygotować – polega to na usunięciu wewnętrznego wzmocnienia. Można też zamiast podstawek DIP użyć złączy SIP (taka jednorzędowa

Rys. 3 Schemat montażowy

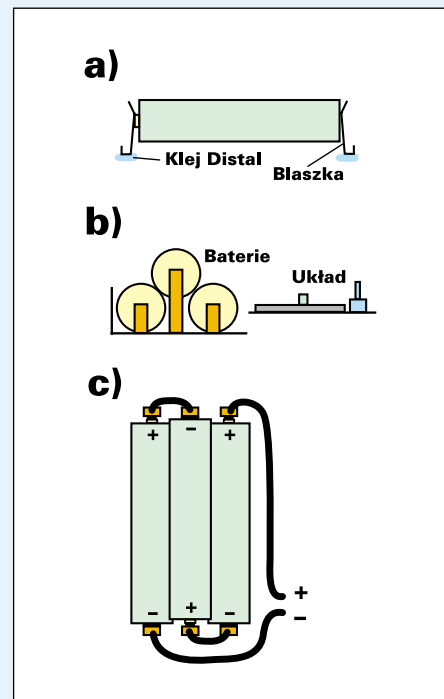


precyzyjna podstawka). Montaż należy rozpocząć od dwóch zworek. Następnie montujemy przygotowane podstawki i prawie do końca działamy standardowo – montując elementy o coraz to większych gabarytach. Kończymy na połączeniu kabelkami punktów A z A' i B z B'. Płytkę mamy gotową.

Gotowa płytką będzie zbyt wysoka, aby dała się zamknąć w obudowie. Bolce oraz wypustki na śruby w części dolnej należy spiliować lub przyciąć o około 2-3mm. Dobrze jest na tym etapie sprawdzić czy wszystko już pasuje. Gdy obudowa daje się już zamknąć, płytkę przykręcamy dwoma blachowkrętami. Z boku należy wypiliować niewielki rowek, przez który „przejdzie” kabelek do głośnika. Głośnik przyklejamy na zewnątrz obudowy. Świetnie do tego nadaje się klej na ciepło z pistoletu.

Układ zasilany jest z trzech baterii R03 - inaczej AAA. Wymaga to jednak samodzielnego stworzenia uchwytów pod baterie, co przedstawia **rysunek 2**. Rysunek a) pokazuje ogólną ideę zamocowania jednej baterii; b) widok z boku na ustawienie baterii względem układu; c) widok z góry razem z układem połączeń. Są to po prostu blaszki przyklejone do obudowy mocnym klejem. Nie wielka ilość miejsca zmusza do umieszczenia jednej baterii wyżej od pozostałych.

Rys. 2 Mocowanie baterii



Nauka mówienia

Jeśli zasilanie jest włączone - wyłączamy je. Czekamy aż kondensatory rozładują się (może to potrwać kilka sekund). Usuwamy zworkę *Prog* i ponownie załączamy układ.

Wejście w tryb programowania zostanie zasygnalizowane trzykrotnym mignięciem diody. Teraz posługując się wcześniej wspomnianą tabelą 1 tworzymy kolejne dźwięki. Interfejs jest w miarę prosty. Przycisk *Prog* – nagrywanie, *Losuj* – (krótkie naciśnięcie) odtwarzanie aktualnego dźwięku, (przytrzymanie aż do mignięcia diody LED) zatwierdzenie i przejście do następnej pozycji, Zworka *Prog* – zerowanie licznika pozycji. Jeśli LED zgaśnie podczas nagrywania, znaczy to o przepełnieniu pamięci. Jeśli podczas próby zatwierdzenia mignie cztery razy oznacza to, że następna próbka jest 15 – a taka nie istnieje, w związku z czym nie można jej obrabiać.

Rada: nie śpiesz się! 20 sekund to dużo jak na 14 komunikatów. Dużo lepiej brzmią słowa wypowiedane wyraźnie i w miarę powoli niż szybko.

Przyjęty sposób programowania umożliwia zmianę próbek od wybranego momentu. Na przykład, jeśli będziemy chcieli zmienić dźwięki losowania i stukania nie musimy w tym celu zmieniać dźwięków określających ilość oczek. Wystarczy wszystkie je „zatwierdzić”.

Uwaga: Odtworzenie komunikatu, który nie kończy się znacznikiem EOM (może się to zdarzyć na przykład jeśli nagrywanie zakończy się z powodu przepełnienia pamięci), lub próba przewinięcia (zatwierdzenia) takiego komunikatu spowoduje zawieszenie się programu, ponieważ będzie on oczekiwać na ujemny impuls na wyjściu LED U1, który nigdy nie nadejdzie. Jedynym wyjściem z tej sytuacji jest wyłączenie i ponowne włączenie zasilania.

W modelu pojawił się pewien problem: gdy zasilanie „padnie” podczas odtwarzania układ może nagrać jakieś bzdury do ISD. Jest to prawdopodobnie spowodowane tym, że zastosowany mikrokontroler AT90S2313-10 pracuje poprawnie przy napięciu 4-6V, a resetuje się przy 2,2V. Aby tego uniknąć można użyć układu AT90S2313-4. Będzie to wymagało zmiany kwarcu na 4MHz, lecz nie powinno to mieć większego wpływu na działanie układu (nie trzeba będzie zmieniać programu).

Parę uwag końcowych (niekoniecznie) na serio

Można spróbować także podjąć się skonstruowania gadającej planszy do gry, gadających pionków oraz gadającej latarki, która poinformuje nas, że zgasła i przynajmniej będziemy mieli w tej sprawie jasność.

Chętnie poznam posiadacza *sluchajacej kostki*.

Radosław Koppel

Uwaga! Plik źródłowy programu i program wynikowy (bas i bin) można ściągnąć ze strony internetowej www.edw.com.pl/library/pliki/gadkosrk.zip

Wykaz elementów

R1	100kΩ
R2	10kΩ
R3	2,2kΩ
R4	5,6kΩ
C1,C4	10μF/16V
C2,C3,C5	220nF
C6	100nF ceramiczny
C7	220μF/16V
C8,C9	33pF
D1	dioda LED
U1	ISD1420 (ew. ISD1416)
U2	AT90S2313
X1	niskoprofilowy rezonator kwarcowy 6MHz
S1	miniaturowy przycisk z klawiszem
S2	μSwitch 1mm
Z2	2x Goldpin z jumperem

Komplet podzespołów z płytką jest dostępny w sieci handlowej AVT jako kit szkolny AVT-3011