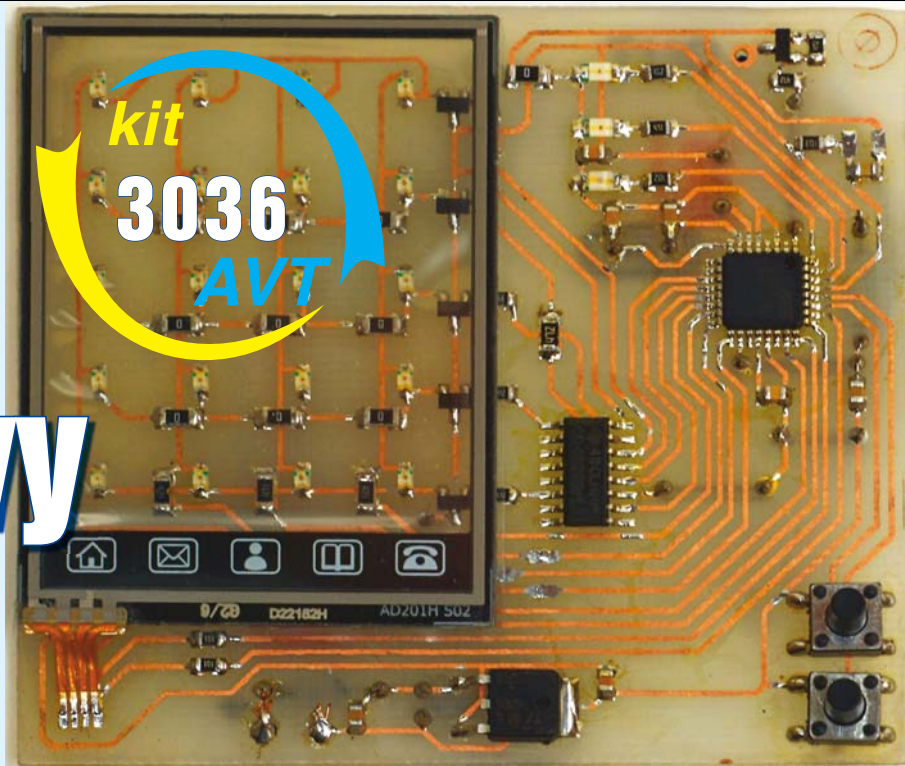


Zamek gestowy

kit
3036
AVT



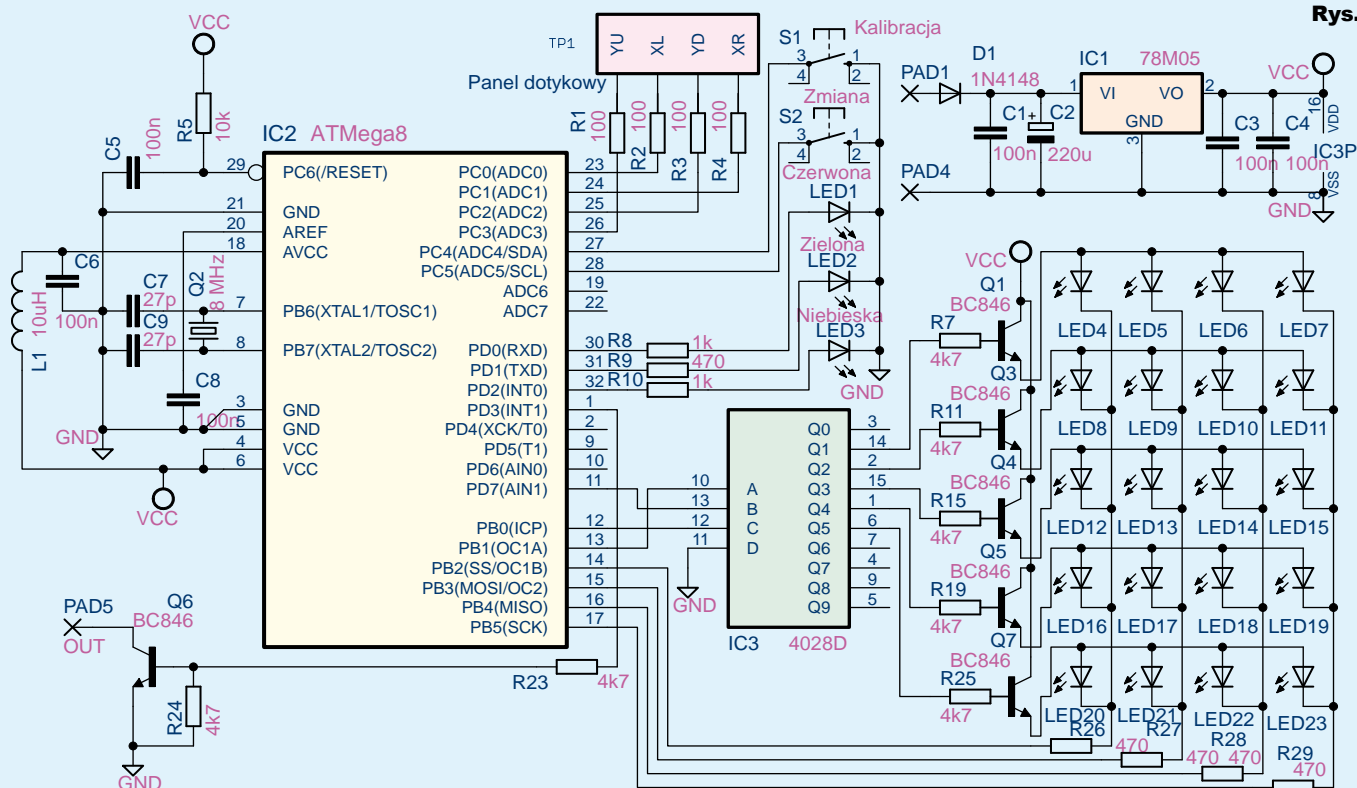
Innowacyjny zamek elektroniczny. Aby uzyskać dostęp, należy wykreślić palcem lub rysikiem odpowiedni gest na płycie dotykowej. Znakomity efekt dopełnia rój maleńkich LED-ów.

Zadanie główne nr 188 Szkoły Konstruktorów dało mi okazję do wypróbowania czegoś nowego – mówię tu o wykorzystaniu panelu dotykowego (który kupiłem 2 miesiące wcześniej) i lutowaniu układów scalonych świeżo zakupionym grotem typu *minifala*.

Pragnę przedstawić zaskakujący i niesamowity elektroniczny system kontroli dostępu. Na łamach EdW opisywanych było mnóstwo przeróżnych zamków/blokad – opartych na pastylkach Dallasa, kartach bezstykowych lub

po prostu kod był wprowadzany za pomocą klawiatury. Ja proponuję coś zupełnie nowego – zamek elektroniczny, który można otworzyć, wykonując na panelu dotykowym odpowiedni gest palcem lub rysikiem. Pod owym panelem znajduje się 20 diod LED, podążających za naszym ruchem, aż zwolnimy nacisk i nastąpi weryfikacja. Nie znajdziemy tu żadnego wyświetlacza – o stanie urządzenia informują nas trzy dodatkowe diody, co dodaje całości specyficznego uroku...

Nad całością „czuwa” mikrokontroler ATmega8L w obudowie TQFP32, a pomaga mu w tym dekodery CMOS 4028 (obudowa SOP16). Oba te maleństwa zostały przylutowane za pomocą wspomnianego wcześniej grota *minifala*, z zastosowaniem roztworu kalafonii w spirytusie jako topnika – po prostu rewelacja! Polecam wszystkim zakup tego narzędzia (koszt ok. 40zł), mnie świetne efekty wyszły już za pierwszym razem (ważne tylko, by nie żałować topnika).



Rys. 1

Jak to działa?

Schemat pokazany jest na **rysunku 1**. Napięcie zasilające jest dołączane do złączy PAD1 i PAD2 i stabilizowane przez układ IC1 78M05. Spokojnie poradziłby sobie mniejszy, 78L05, ale nie miałem takowego w wersji SMD. Dioda D1 chroni przez odwrotną polaryzacją zasilania. Pobór prądu przez układ wynosi 4–30mA.

Urządzenie zostało wykonane z użyciem mikrokontrolera ATmega8L (IC2), taktowanym częstotliwością 8MHz. Do nóżek PC0–PC3 dołączony został oporowy panel dotykowy – taki sam, jaki stosowany jest w dotykowych wyświetlaczach. Do jego obsługi wykorzystany został przetwornik analogowo-cyfrowy wbudowany w ATmegę. Na schemacie widzimy matrycę 20 diod LED (numeracja LED4 – LED23). Katody tych diod (kolumny) sterowane są bezpośrednio przez odpowiednie nogi mikrokontrolera, zaś anody (wiersze) są wybierane przez układ 4028 – dekodler kodu BCD na dziesiątki. Diody te są multipleksowane i zostały umieszczone pod przezroczystym panelem dotykowym. Ich zadanie polega na nasładowaniu naszego ruchu na tym elemencie.

Do procesora dołączone są także trzy diody LED (oznaczone jako LED1–LED3) w kolorach czerwonym, zielonym i niebieskim, które informują nas o stanie układu i wykluzczają konieczność zastosowania drogiego i sporego wyświetlacza. Są także dwa przyciski, umożliwiające wejście w tryb kalibracji płytki dotykowej lub zmianę zapamiętanego gestu. Do wyprowadzenia PD3 został dołączony tranzystor NPN, sterujący ewentualnym przekaźnikiem, rygłem, jakim układem elektronicznym – czymkolwiek, co przyjdzie nam do głowy. Jest to uniwersalne wyjście typu otwarty kolektor (obciążalność 100mA), zwierające do masy.

Filtr L1, C6, C8 zasilają analogową część elektroniki mikrokontrolera, która jest wykorzystywana do określania pozycji nacisku na panel dotykowy. Obwód R5, C5 zapewnia poprawny reset przy włączeniu zasilania, a elementy Q2, C7, C9 służą do taktowania ATmegi częstotliwością 8MHz.

Do przechowywania kodu (gestu) i danych kalibracji wykorzystana została pamięć EEPROM ATmegi. Dzięki temu dane nie zostaną stracone nawet przy zaniku zasilania. Dla zwiększenia niezawodności, nad pracą mikrokontrolera czuwa sprzętowy watchdog, który podczas ewentualnego zawieszenia się układu natychmiast go zresetuje. Tego samego dokona wbudowany w ATmegę brown-out detector, gdy napięcie zasilające spadnie poniżej wartości 4V.

Program na mikrokontroler został napisany w języku BASCOM AVR i jest rozbudowany – zajmuje ok. 64% pamięci układu. Taki wynik jest spowodowany m.in. koniecznością zastosowania zmiennych typu *single precision* do przeliczania ułamkowych części współrzędnych. Ci, którzy mają wersję demo

BASCOM-a, mogą załadować skompilowany kod na pokład ATmegi bez żadnych problemów – ograniczenie dotyczy tylko kompilatora.

Obsługa oporowego panelu dotykowego. Jest on zbudowany z dwóch warstw – poziomej i pionowej, zachowujących się jak oporniki. Do naszej dyspozycji są dwie pary wyprowadzeń – góra–dół oraz prawo–lewo. Nacisk na panel powoduje zwarcie się tych dwóch elektrod w punkcie, w którym dotknęliśmy. Warto podczas tworzenia programu napisać najpierw instrukcje wykrywające nacisk, a dopiero później odczytywać konkretną pozycję. Ja to zrobiłem w następujący sposób: elektrody „góra” i „dół” zwieram do masy (stan niski na odpowiedniej linii procesora), zaś pin procesora podłączony do jednej (którejkolwiek) z pozostałych elektrod konfigurujemy jako wejście i włączamy rezystor podciągający do plusa (pull-up wbudowany w mikrokontroler). Monitorując w pętli stan tego wejścia, oczekujemy, aż wystąpi na nim logiczne zero, co oznacza, że elektrody zostały gdzieś zwarte – dotknięto panelu. Dopiero teraz przechodzimy do procedury odczytu konkretnej pozycji:

Zacznijmy od współrzędnej np. pionowej. W tym celu elektrodę „góra” polaryzujemy dodatnio (na odpowiednim wyjściu ATmegi logiczna jedynka), a elektrodę „dół” ujemnie. Wyprowadzenia procesora podłączone do pary prawo–lewo ustawiamy jako wejścia i wyłączamy wszelkie pull-upy. Z jednego, dowolnego z tych wyprowadzeń, czytujemy wartość napięcia za pomocą przetwornika ADC, wbudowanego w mikrokontroler (wykorzystujemy maksymalną dostępną rozdzielczość). W analogiczny sposób badamy drugą współrzędną. Istotne jest, że nie otrzymamy tu wyniku w zakresie 0–VCC, lecz „coś” pomiędzy, i dlatego konieczna jest kalibracja. Jeżeli wykonujemy urządzenie w jednym egzemplarzu, lub panel jest dość mały, można byłoby te skrajne położenia zbadać i wpisać w program na stałe. Standardowo kalibrację przeprowadza się, wskazując dwa skrajne punkty po przekątnej płytki dotykowej. Dla zainteresowanych podaję gotowe wzory matematyczne, pozwalające obliczyć współrzędne (obie w zakresie 0..1) wskazywanego punktu:

$$X = \frac{U_{ADC} * X_{min}}{X_{max} - X_{min}}$$

$$Y = \frac{U_{ADC} * Y_{min}}{Y_{max} - Y_{min}}$$

gdzie U_{ADC} – odczytana wartość z przetwornika ADC, X_{min} , Y_{min} – wartości zwrócone przez ADC podczas kalibracji dla punktu (0, 0) X_{max} , Y_{max} – jw., ale dla punktu o współrzędnych (1, 1)

Montaż i uruchomienie

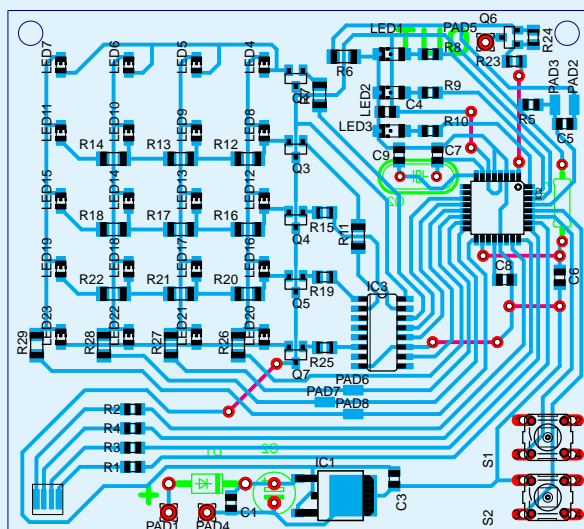
Układ można zmontować na płytce drukowanej, pokazanej na **rysunku 2**. Praktycznie każdy element występuje w wersji SMD. Składanie takiego urządzenia nie jest wcale tak trudne, jak mogłoby się wydawać, ale wymaga odrobiny doświadczenia, skupienia i sprawdzonych, dobrych narzędzi. Pomocą w montażu mogą też być fotografie.

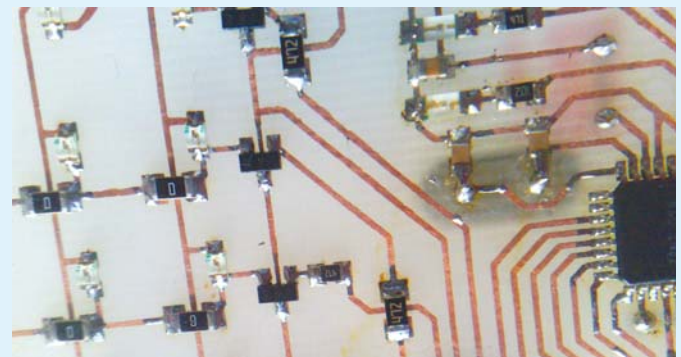
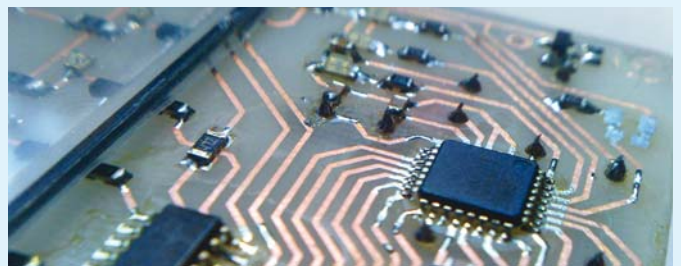
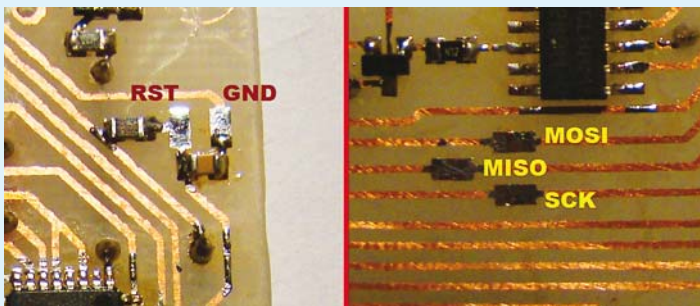
Montaż zaczynamy od układu ATmega8L (może być wersja bez literki L). Najlepiej posłużyć się tutaj grotem typu *minifala* oraz dużą ilością topnika, np. RF800 lub roztworem kalafonii w etanolu. Według internautów najlepszy do SMD (mimo swojej ceny) jest topnik w żelu, ale ja go jeszcze nie wypróbowałem. Na początku, po precyzyjnym ustawieniu układu, lutujemy dwie jego nóżki leżące po przekątnej, aby później móc spokojnie zabrać się za pozostałe. Mnie lutowanie układów scalonych za pomocą *minifali* wyszło bardzo dobrze za pierwszym razem (przed rozpoczęciem warto obejrzeć filmiki na portalu *YouTube*, przedstawiające taki sposób lutowania). Przy odrobieniu cierpliwości można też przylutować układy scalone zwykłym, stożkowym grotem.

Następnie montujemy układ 4028. Jest to łatwiejsze, ponieważ odstęp między jego nóżkami są znacznie większe, niż w ATmega8. Montując ten i poprzedni układ, sprawdzamy kilka razy, czy czasem nie obróciliśmy tego elementu i lutujemy go we właściwym kierunku! W przypadku sklejenia wyprowadzeń cyną, dokładamy topnika i ściągamy nadmiar spoiwa z nóżek. Po zakończeniu sprawdzamy miernikiem, czy nie powstały zwarcia. A często nie widać ich gołym okiem!

Jeżeli wszystko jest OK, montujemy rezystory i kondensatory **oprócz C5!** Najpierw nakładamy niewielką ilość cyny na jeden z padów, na oba dokładamy dodatkowego topnika (tego w druciku cyny jest zdecydowanie za mało), następnie przykładamy element pęsetą i topimy nałożoną wcześniej

Rys. 2





Fot. 1

cyne. Teraz wystarczy zalutować drugi pad z elementem i mamy gotowe, solidne połączenie. Efekty będą coraz lepsze, gdy będziemy nabierać doświadczenia.

Następnie lutujemy diody LED (katoda jest wyraźnie oznaczona kropką), tranzystory, stabilizator 78M05, przyciski microswitch (te montujemy po stronie ścieżek), a następnie elementy przewlekane: zwory z drutu, dławik L1, diodę D1, rezonator kwarcowy Q2 i kondensator elektrolityczny (warto zgiąć mu nóżki i przylutować „na leżąco”, by oszczędzić miejsce). Montaż płytki dotykowej radzę pozostawić na później.

Teraz musimy zaprogramować mikrokontroler. Do tego celu zostały przewidziane specjalne pady, pokazane i opisane na **fotografii 1**. Najpierw montujemy przewód zasilający (dwa przewlekane pady obok diody D1), a następnie krótkie przewody do ww. padów do zaprogramowania procka.

Wypadałoby jeszcze posprawdzać miernikiem, czy nie uchwaloło się przed naszym okiem jakieś zwarcie/przerwa. Jeśli wszystko gra, włączamy zasilanie (9–12V) i wysyłamy program do ATmegi. Po tej czynności odlutowujemy kable programowania i czyścimy po nich pady za pomocą plecionki lutowniczej (dla estetyki). Teraz przylutowujemy „zale-

gły” kondensator C5 (jego obecność mogłaby uszkodzić programator) i (nareszcie!) panel dotykowy. Musimy włożyć trochę wysiłku w równe jego ustawienie. Kiedy uznamy, że jest OK, grzejemy jeden z jego pinów na tasieчке tak, by „chwycił” się płytki.

Dokładamy topnika i lutujemy po kolei pozostałe trzy pola. **Konieczn**ie musimy zachować daleko idącą ostrożność podczas montowania panelu dotykowego! Nawet najdrobniejsze draśnięcie lutownicą krawędzi tego elementu spowoduje, że odczyty będą nieprawidłowe (zaburzone, nieliniowe) i element będzie się nadawał do kosza (żadna kalibracja nie pomoże). Doświadczyłem tego – nie życzę nikomu...

Pozostaje jeszcze mocowanie panelu tak, by nie trzymał się na samej taśmie, która zapewnia mu kontakt elektryczny z procesorem. Najlepiej byłoby zastosować jakąś obudowę (można wykonać z plexiglasu). Ja swojego modelu ostatecznie nie obudowałem i podczas wprowadzania kodu muszę przytrzymać panel palcami za krawędzie. Nie polecam tego rozwiązania, ponieważ taśma może się obruszać i pęknąć.

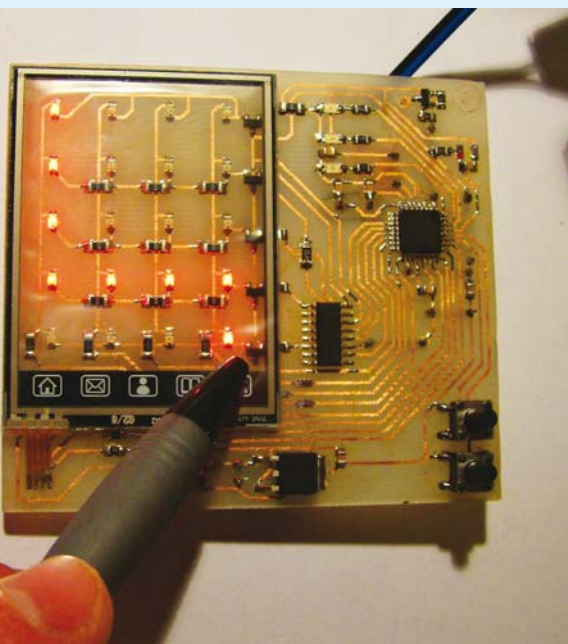
Po skończeniu płytkę myjemy izopropanolem lub spirytusem, aby usunąć szpetne resztki topnika. Układ gotowy! Podłączamy zasilanie (9...12V) i zaczynamy zabawę...

Zamek nie daje oznak „życia”. Podczas czuwania nie świeci ani nie mruga żadna dioda, aby oszczędzać energię i zachować swego rodzaju tajemniczość. Dopiero dotknięcie płytki spowoduje zaświecenie się diody, leżącej pod punktem nacisku. Po kilku sekundach takiej zabawy zauważymy, że diody zaświecają się niedokładnie tam, gdzie powinny. To znak, że panel dotykowy należy skalibrować. W tym celu naciskamy górną przycisk (S1). Zaświeci się duża niebieska dioda oraz mała, znajdująca się w dolnym lewym rogu panelu. Naciskamy na płytkę w miejscu tej czerwonej, po czym powinna ona zgasnąć (w tym momencie zwalniamy nacisk). Po sekundzie zaświeci się ta po prze-

ciwnej stronie panelu, której położenie także należy procesorowi wskazać. Teraz możemy sprawdzić poprawność kalibracji. Nietrudno się domyślić, że przeciągając palcem/rysikiem po płytce dotykowej, spowodujemy zaświecenie wszystkich diod, podążających za naszym ruchem, tak jak na **fotografii 2**. W ten właśnie sposób wprowadzamy do urządzenia gesty. Po zwolnieniu nacisku następuje przetwarzanie/weryfikacja „hasła”. W tym momencie również zaświeca się duża zielona lub czerwona dioda – w zależności od tego, czy wprowadziliśmy poprawny, czy też błędny kod.

Ciąg dalszy na stronie 67

Fot. 2



Wykaz elementów

R1-R4	100Ω SMD 805
R5	10 kΩ SMD 805
R6,R12-R14,R16-R18,R20-R22	0Ω (zwora) SMD 1206 (nie ma na schemacie)
R7, 11	4,7kΩ SMD1206
R8,R10	1kΩ SMD 805
R9	470Ω SMD 805
R15, R19, R23-R25	4,7kΩ SMD 805
R26-R29	470Ω SMD 1206
C1,C3-C6,C8	100 nF SMD 805
C2	220 uF / 16V elektrolityczny
C7,C9	27 nF SMD 805
D1	1N4148 lub podobna
IC1	78M05
IC2	ATmega8(L) TQFP32
IC3	4028 SO16
LED1	Dioda LED SMD 1206 czerwona
LED2	Dioda LED SMD 1206 zielona
LED3	Dioda LED SMD 1206 niebieska
LED4-LED23	Dioda LED SMD 805 czerwona
Q1,Q3-Q7	Tranzystor BC846
Q2	Rezonator kwarcowy 8 MHz
TP1	Oporowy panel dotykowy 2,2"
S1,S2	Microswitch (wys. 6 mm)
L1	Dławik 10 uH

Płytką drukowaną jest dostępna w sieci handlowej AVT jako kit szkolny AVT-3036.

Ciąg dalszy ze strony 52

No właśnie, fajnie by było dowiedzieć się, jak wprowadzić/zmienić gest, którego wykonanie zaświeci tę zieloną. Do tego służy drugi przycisk (S2) w rogu. Po wciśnięciu go, zaświeci się niebieska dioda, a układ oczekuje, aż podamy dotychczasowy kod. Kiedy okaże się on poprawny, mrugnie do nas zielona dioda. W tym momencie musimy dwa razy wprowadzić nowy gest (analogicznie jak zmiana hasła w komputerze). Jeżeli wszystko się zgadza – znów zaświeci się na chwilę dioda zielona, a nowe hasło wędruje do pamięci EEPROM Atmegi. Po pierwszym uruchomieniu urządzenia domyślnym kodem jest naciśnięcie diody w górnym lewym rogu panelu.

Po wprowadzeniu i zapamiętaniu nowego hasła, zamek przejdzie w tryb czuwania. Możemy wprowadzić nowo ustawiony gest – jeśli będzie on poprawny, zaświeci się na kilka sekund zielona dioda i wyjście OUT będzie zwarte do masy. Jeśli chcemy, możemy od razu wrócić do stanu oczekiwania przyciskiem S2 – wyjście przejdzie z powrotem do stanu wysokiej impedancji, a mikrokontroler do stanu oczekiwania.

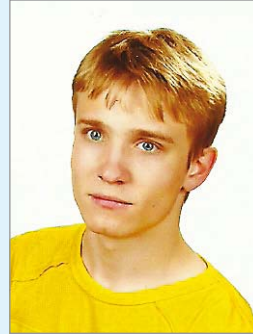
Warto dodać, że weryfikacja wpisanego gestu następuje po upływie ok. 0,5 sekundy po zwolnieniu nacisku, więc jeśli jesteśmy dość szybcy, możemy zaświecić diody pod panelem nieleżące w jednej linii (skutecznie utrudnia

odgadnięcie). Urządzenie nie rozróżnia ścieżki ani kolejności zaświecania diod, ważny jest efekt końcowy. Po zaświeceniu jakiejś diody nie mamy możliwości jej zgasić.

Na koniec policzmy sobie, ile jest wszystkich możliwych hasel. Przy dwudziestu diodach, gdzie każda może przyjmując dwa stany (świeci/nie świeci) mamy w sumie...

$2^{20} - 1 = 1\ 048\ 575$ różnych gestów!

Jeżeli więc potencjalny amator cudzej własności trafi akurat na ten poprawny, to powinien zagrać w „totka”.



Michał Pędzimąż
mpedzimaz@gmail.com