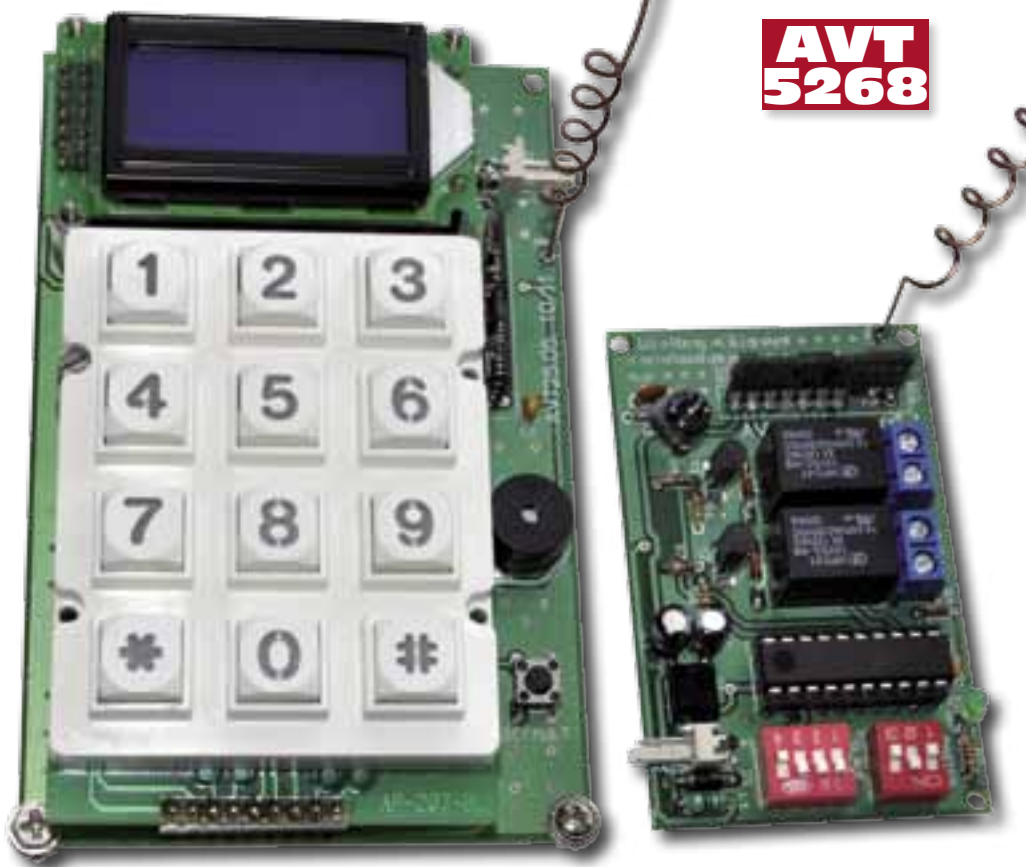


Zamki szyfrowe o tradycyjnej konstrukcji są coraz rzadziej stosowane w systemach kontroli dostępu, ponieważ element wykonawczy w postaci przełącznika najczęściej jest umieszczony w obrębie tego samego modułu i przez to łatwo dokonać sabotażu. Rozwiązaniem może być wykonanie klawiatury i elementu wykonawczego jako oddzielnych modułów, a następnie połączenie ich za pomocą fal radiowych.

**Rekomendacje:** zamek jest w pełni profesjonalną konstrukcją, mogącą spełnić oczekiwania wymagających użytkowników.



# Wikey

## Bezprzewodowy zamek szyfrowy

W EP 6/2010 prezentowałem projekt zamka szyfrowego wyposażonego w dwa niezależne przekaźniki sterujące i interfejs użytkownika z wyświetlaczem LCD. Jeden z moich znajomych zajmujący się systemami zabezpieczeń wyraził opinię, że zamki szyfrowe są coraz rzadziej stosowane we wszelkiego rodzaju układach kontroli dostępu, ponieważ element wykonawczy jest umieszczony w obrębie tego samego modułu, co interfejs użytkownika, a to powoduje, że takie zabezpieczenie jest łatwe do sforsowania. Postanowiłem skonstruować zamek szyfrowy złożony z klawiatury sterującej i modułu przekaźników, oba sprzęgnięte za pomocą fal radiowych w paśmie ISM 433 MHz. Należy jednak szczególnie podkreślić, iż prezentowane urządzenie nie rozwiązuje problemu ogólnie pojętego bezpieczeństwa, gdyż nie korzysta się w nim z żadnych metod szyfrowania transmisji danych. Ideałem byłoby zastosowanie modułów ZigBee, które wspierają autentykację, szyfrowanie transmisji

i mają wbudowany mechanizm unikania kolizji, lecz moim zdaniem byłoby to rozwiązanie przewymiarowane biorąc pod uwagę cenę i przeznaczenie.

### Aurel RTX-MID-3V

W urządzeniu zastosowano zintegrowany moduł transceivera firmy Aurel typu RTX-MID-3V pracujący w paśmie 433 MHz. Moduł ten charakteryzuje się następującymi parametrami:

- praca w trybie half-duplex,
- modulacja ASK,
- krótki czas przełączania pomiędzy trybami nadawania i odbioru,
- moc wyjściowa ok. 10 mW,
- niski pobór mocy w trybie PowerDown (rzędu 1  $\mu$ A),
- maksymalna prędkość transmisji 9600 bps,
- czułość odbiornika -106 dBm.

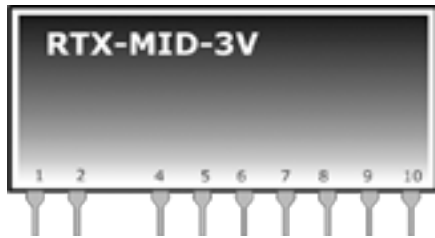
Wygląd modułu transceivera oraz opis jego wyprowadzeń pokazano na **rysunku 1**.

**AVT-5268 w ofercie AVT:**  
AVT-5268A – płytka drukowana  
AVT-5268B – płytka drukowana + elementy

**Podstawowe informacje:**  
Moduł klawiatury (Wikey Master):  
• Napięcie zasilania: 8...12 VDC  
• Pobór mocy (tryb bezczynności/praca): 0,42/0,54 W  
• Liczba możliwych adresów układu Master: 8  
• Częstotliwość pracy transceivera: 433,92 MHz  
• Zasięg w terenie otwartym: ok. 100 m  
Moduł wykonawczy (Wikey Slave):  
• Napięcie zasilania: 12 VDC  
• Pobór mocy (tryb bezczynności/załączenie przekaźnika): 0,24/0,66 W  
• Liczba możliwych adresów układu Slave: 16  
• Częstotliwość pracy transceivera: 433,92 MHz  
• Zasięg w terenie otwartym: ok. 100 m  
Ustawienia ważniejszych FUSE BIT'ów Wikey-Master:  
• CKSEL3..0: 0100  
• SUT1..0: 10  
• EESAVE: 0  
Ustawienia ważniejszych FUSE BIT'ów Wikey-Slave:  
• CKSEL3..0: 0100  
• SUT1..0: 10  
• EESAVE: 1  
• CKDIV8: 1

**Dodatkowe materiały na CD i FTP:**  
<ftp://ep.com.pl>, user: 16195, pass: 4k17u606  
• 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-5241 Zamek szyfrowy (EP 6/2010)  
AVT-5042 Zamek szyfrowy do sejfu (EP 1/2002)  
AVT-1309 Zamek szyfrowy z systemem alarmowym (EP 7/2001)  
AVT-869 Zamek szyfrowy z jednym przyciskiem (EP 10/2000)  
AVT-377 Sezam na cztery sposoby (EP 3/1998)  
AVT-311 Uniwersalny zamek szyfrowy (EP 1/1997)  
Projekt 065 Zamek szyfrowy typu „SEJF” (EP 8/1999)



Rysunek 1. Wygląd modułu transceivera oraz funkcje jego wyprowadzeń

Pin	Nazwa	Opis
1	Antena	Wyjście antenowe transceivera. Impedancja 50 Ω
2	GND	Masa
4	Data Input	Wejście nadajnika transceivera (1=emisja nośnej, 0=brak nośnej)
5	TX/RX	Tryb pracy transceivera (1=nadajnik, 0=odbiornik). Podłączone rezystorem pull-down do masy
6	Enable	Aktywacja transceivera (0= tryb PowerDown o niskim poborze mocy rzędu 1 μA, 1=transceivera aktywny)
7	GND	Masa
8	Analog	Pin testowy. Wykorzystywany dla realizacji funkcji „squelch”
9	Data Out	Wyjście odbiornika transceivera. Podłączone rezystorem pull-down do masy
10	VCC	Napięcie zasilania: 2,1...3,6 V. Zalecany kondensator odsprzedający 100 nF

Jak wspomniano wyżej, RTX-MID-3V (dostępny również w wersji o napięciu zasilania 5 V oznaczonej RTX-MID-5V) charakteryzuje się krótkimi czasami przełączania pomiędzy funkcjami nadawania i odbioru oraz krótkim czasem wyjścia z trybu *Power Down*. Należy jednak zachować określoną kolejność sygnałów sterujących, która zapewni krótkie czasy ładowania/rozładowania wewnętrznych obwodów RC transceivera, jak i uwzględnić czas stabilizacji wewnętrznego oscylatora PLL. Na **rysunku 2** zilustrowano zależności czasowe dla 4 przypadków zmian trybów pracy.

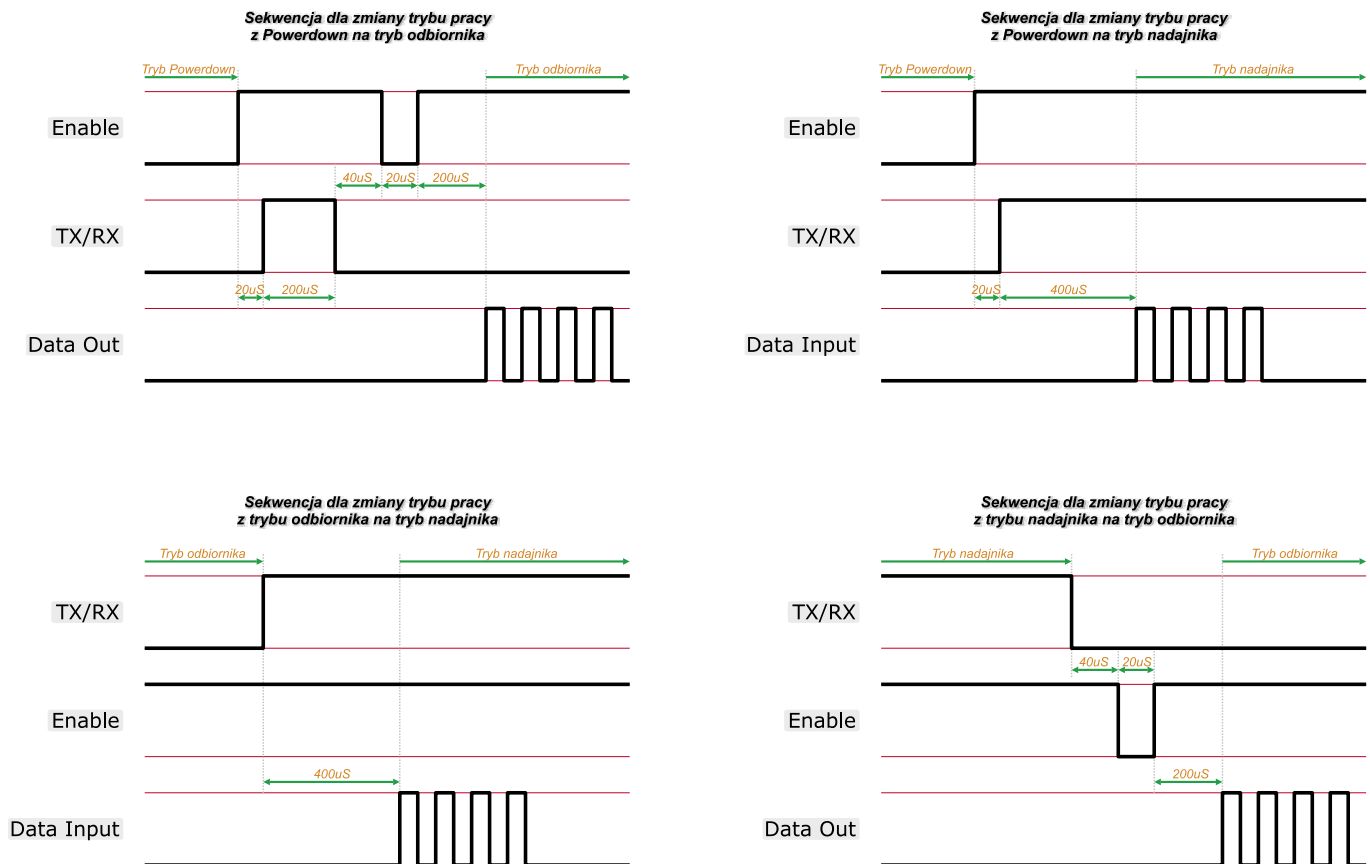
### Oprogramowanie sterujące

Wydawać by się mogło, że użycie modułu transceivera nie jest trudne. Wystarczy dołączyć jego wejścia i wyjścia do wyprowadzeń interfejsu USART mikrokontrolera oraz wyprowadzeń, dzięki którym można sterować modułem. Jednak system z nadrzędny musi odróżnić zakłócenia od ważnych ramek danych. Pomocna w tym może być funkcja *Squelch* (pin 8 transceivera), która powoduje zwarcie do masy wyjścia odbiornika w przypadku braku odbioru sygnału o częstotliwości nośnej. Realizacja tej funkcji

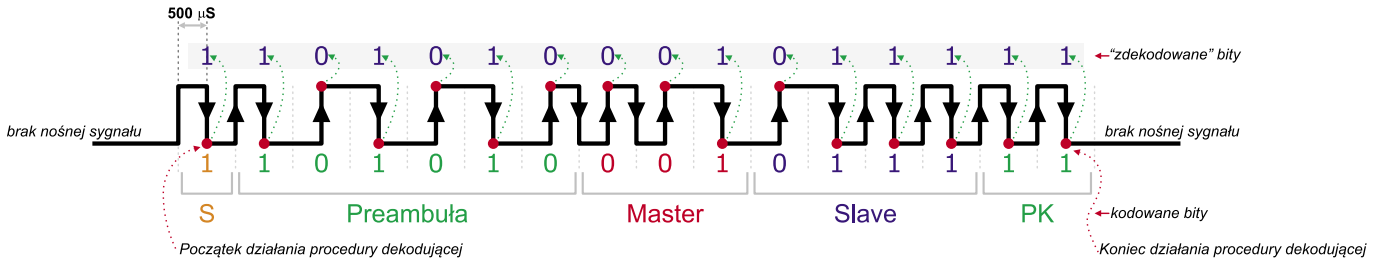
jest możliwa dzięki zastosowaniu prostego dzielnika rezystorowego i wykorzystaniu stanu wyprowadzenia *Enable* (pin 6 transceivera), jednak zmniejsza czułość odbiornika o 3 dB, jak również nie rozwiązuje wszystkich problemów związanych z zakłóceniami. W takiej sytuacji rozwiązaniem może być zastosowanie sprawdzonej, odpornej na błędy metody kodowania transmisji. Jedną z nich jest kodowanie typu Manchester, szeroko stosowane np. w układach sterowania z wykorzystaniem podczuwieni. Każdy bit jest kodowany w postaci zmiany poziomów logicznych, tzw. półbitów, a zmiana ta następuje w środku czasu przeznaczanego na przesłanie pojedynczego bitu. I tak, logiczna „1” jest przesyłana jako zmiana poziomu z wysokiego na niski, natomiast logiczne „0” z niskiego na wysoki. Dla naszych celów przyjmujemy, iż czas trwania bitu wynosi 1000 μs, co daje prędkość transmisji równą 1000 bps. Mieści się to w zakresie możliwości transceivera. Dodatkowo przyjmujemy pewne założenia dotyczące formatu ramki danych. Pełna ramka transmisji powinna mieć następującą strukturę:

1	101010	M2 M1 M0	S3 S2 S1 S0	P1 P0
Bit startowy	Preambuła	Adres układu Master	Adres układu Slave	Kod przełącznika do załączenia

Ramka transmisji jest złożona z 16 bitów, a czas jej trwania wynosi 16 ms. Bit pierwszy jest bitem startowym i służy do synchroni-



Rysunek 2. Zależności czasowe dla 4 przypadków zmiany trybów pracy transceivera



- ▼▲ - aktywne zbocze sygnału powodujące przechwycenie stanu licznika Timer1
- - moment "zaliczenia" kolejnego bitu ramki danych

Rysunek 3. Przykładowa, kompletna ramka danych

zacji odbiornika z nadajnikiem. Następnie jest nadawana preambuła zwiększająca odporność danych na przekłamania. Kolejne 3 bity to adres urządzenia *Master* (modułu klawiatury), który nadaje dane. Za nimi są umieszczone 4 bity adresu urządzenia *Slave* (karty przekaźników), do którego adresowane są dane. Ostatnie dwa bity kodują numer załączanego przekaźnika (01: przekaźnik pierwszy, 11: przekaźnik drugi). Przewidziano możliwość pracy do 8 układów typu *Master* (klawiatur) i do 16 układów typu *Slave* (kart przekaźnikowych). Sparowane ze sobą klawiatury i moduły przekaźnikowe muszą mieć ustawione takie same adresy *Master/Slave*. Niedopuszczalne jest sparowanie dwóch modułów przekaźnikowych z jednym modułem klawiatury, lecz można sterować pojedynczym modułem wykonawczym za pomocą wielu modułów klawiatury. Ograniczenie to wynika z faktu, iż moduł przekaźnikowy każdorazowo potwierdza otrzymanie ważnej dla niego ramki danych poprzez odsłanie jej do modułu klawiatury. W związku z tym połączenie więcej niż jednej karty przekaźnikowej z jedną i tą samą klawiaturą spowodowałoby konflikt transmisji. Dodatkowo, klawiatura, która nie otrzymała potwierdzenia od karty przekaźnikowej lub też otrzymała potwierdzenie z błędem, powtarza wysyłanie rozkazu trzykrotnie, po czym, je-

śli nie otrzymała odpowiedzi, wyświetla na wyświetlaczu LCD komunikat o błędzie.

Przyjęte rozwiązania logiczno-układowe nie zapewniają 100% skuteczności. Możliwe są kolizje transmisji, ponieważ można wyobrazić sobie hipotetyczną sytuację, w której dwa sparowane ze sobą komplety klawiatura-karta przekaźnikowa znajdujące się w zasięgu transmisji próbują nadawać własne ramki danych dokładnie w tym samym czasie i powodują kolizję. Jest to możliwe, jednak mało prawdopodobne, ponieważ:

- transmisja całej ramki danych zajmuje zaledwie 16 ms,
- mało prawdopodobne jest zdarzenie, w którym dwaj użytkownicy sparowanych kompletów klawiatura-karta przekaźnikowa dokładnie w tym samym czasie wprowadzą prawidłowy kod inicjujący transmisję,
- nawet w przypadku wystąpienia kolizji transmisji każdy układ *Master* powtarza transmisję ramki, a czas odstępu pomiędzy poszczególnymi powtórzeniami dobierany jest losowo z zakresu 16...96 ms z krokiem co 16 ms.

Znając zasady kodowania sygnału typu Manchester, strukturę ramki danych i algorytm postępowania, warto przybliżyć jedną z metod stosowanych do dekodowania tego rodzaju transmisji (samo kodowanie jest na

tyle proste, iż komentarz wydaje się zbędny).

Dekodowanie transmisji Manchester można wykonać na wiele sposobów, ale wydaje się, że najlepiej znanym (stosowanym też przez Bascom dla procedury GetRC5) jest użycie przerwania zewnętrznego i jednego z układów czasowo-licznikowych. Dla celu procedury dekodowania użyłem układu Timer1 oraz dwóch generowanych przez niego przerwań: od przechwycenia zawartości licznika (wyzwalanego zboczem sygnału na wejściu ICP mikrokontrolera i używanego do pomiaru czasu trwania półbitów) oraz przerwania od przepełnienia zawartości (zabezpieczające procedurę dekodującą przed niepełnymi lub błędnymi ramkami danych). Na **rysunku 3** pokazano przykładową, kompletną ramkę danych wraz z charakterystycznymi punktami procedury dekodującej.

Procedura dekodująca rusza z chwilą wystąpienia pierwszego, opadającego zbocza sygnału na wejściu ICP mikrokontrolera, które jest dołączone do wyjścia odbiornika transceivera. To pierwsze zbocze sygnału stanowi zarazem bit startowy transmisji o wartości logicznej „1” (początkowa wartość zmiennej Current\_bit=1 procedury dekodującej). Na początku jest zerowany licznik Timer1 i zostaje zmieniony rodzaj zbocza sygnału powodującego kolejne przechwyce-

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

**Wykaz elementów Wikey Master**

- Półprzewodniki:**  
 U1: 78L05  
 U2: Atmega8 (obudowa DIP28)  
 U3: NCP1117DT33G (obudowa DPACK)  
 D1: 1N4004  
 T1: BC560C  
 DISPLAY: wyświetlacz LCD Artronic 6x1 BIG  
 TSC: transceiver Aurel RTX-MID-3V
- Rezystory:**  
 R1...R7: 470 Ω (lub zworki)  
 R8: 100 Ω  
 R9: 1 kΩ  
 R10: 10 Ω  
 R11: 22 kΩ  
 R12: 47 kΩ  
 P1: potencjometr montażowy 10 kΩ  
 P2: potencjometr montażowy 47 kΩ
- Kondensatory:**  
 C1, C3, C5, C6: ceramiczny 100 nF

- C2, C4: elektrolityczny 100 µF/25 V (niski profil)  
 C7: ceramiczny 1 nF  
 C8: elektrolityczny 10 µF/25V (niski profil)
- Inne:**  
 PWR: gniazdo męskie kątowe 90° 2pin (NSL25-2W)  
 DEFAULT: microswitch z krótką ośką  
 BUZ: buzzer piezoelektryczny 5 V  
 KEYBOARD: klawiatura matrycowa (specyfikacja w tabeli 1)

**Wikey-Slave**

- Półprzewodniki:**  
 U1: NCP1117DT33G (obudowa DPACK)  
 U2: Attiny2313 (obudowa DIL20)  
 D1: 1N4004  
 D2...D3: 1N4148  
 T1...T2: BC547  
 TSC: transceiver Aurel RTX-MID-3V  
 CMD: dioda LED 3 mm, zielona

**Rezystory:**

- R1: 47 kΩ  
 R2: 150 Ω  
 R3: 22 kΩ  
 R4, R6: 1 kΩ  
 R5, R7: 4,7 kΩ  
 P1: potencjometr montażowy 47 kΩ

**Kondensatory:**

- C1, C2: elektrolityczny 10 µF/25 V (niski profil)  
 C3, C5: ceramiczny 100 nF  
 C4: ceramiczny 1 nF

**Inne:**

- REL1, REL2: przekaźnik HM4100F/012-NS  
 PWR: gniazdo męskie kątowe 90° 2pin (NSL25-2W)  
 RL1, RL2: złącze śrubowe typu AK500/2  
 L1: dławik 10 µH  
 MASTER: przełącznik dip-switch 3-pozycyjny  
 SLAVE: przełącznik dip-switch 4-pozycyjny



nie: tym razem na zbrocze narastające. Dzieje się tak za każdym razem, gdy zachodzi zdarzenie przechwycenia zawartości licznika Timer1. W ten sposób procedura dekodująca mierzy czas trwania półbitów sygnału. Kolejnym krokiem jest sprawdzenie czy zmierzony czas trwania półbitu mieści się w dopuszczalnym zakresie tj. 250...1250  $\mu$ s (0.25T do 1.25T, T=1000  $\mu$ s) oraz, jeśli sprawdzenie dało wynik pozytywny, czy bieżący półbit jest półbitem „krótkim” czy „długim” (granica podziału jest w tym przypadku wartość 750  $\mu$ s czyli 0.75T). Dwa „krótkie” półbity klasyfikują odebrany bit przebiegu Manchester jako bit o bieżącej (określonej zmienną Current\_bit) wartości, zaś pojedynczy „długi” półbit klasyfikuje odebrany przebieg jako bit o wartości przeciwnej (Current\_bit=not Current\_bit) jednocześnie modyfikując wartość zmiennej Current\_bit. Działanie procedury dekodującej kończy się z chwilą odebrania wszystkich 16-tu bitów ramki transmisji. Fakt ten jest sygnalizowany programowi głównemu aplikacji poprzez ustawienie specjalnej zmiennej – flagi programowej (Ready=1). Dodatkowo, procedura obsługi przzerwania od przepelnienia zawartości licznika Timer1 zabezpiecza algorytm dekodujący w przypadkach odebrania niepełnych lub niemieszczących się w zadanym czasie ramek danych, powodując po czasie około 65 ms bezczynności ustawienie wszystkich zmiennych biorących udział w procedurze dekodującej do ich wartości startowych. Procedura ta jest w naszym przypadku cały czas aktywna, choć tak naprawdę można zezwalać na jej obsługę wyłącznie po odebraniu pierwszego, startowego bitu ramki transmisji, a blokować ją po odebraniu kompletu bitów.

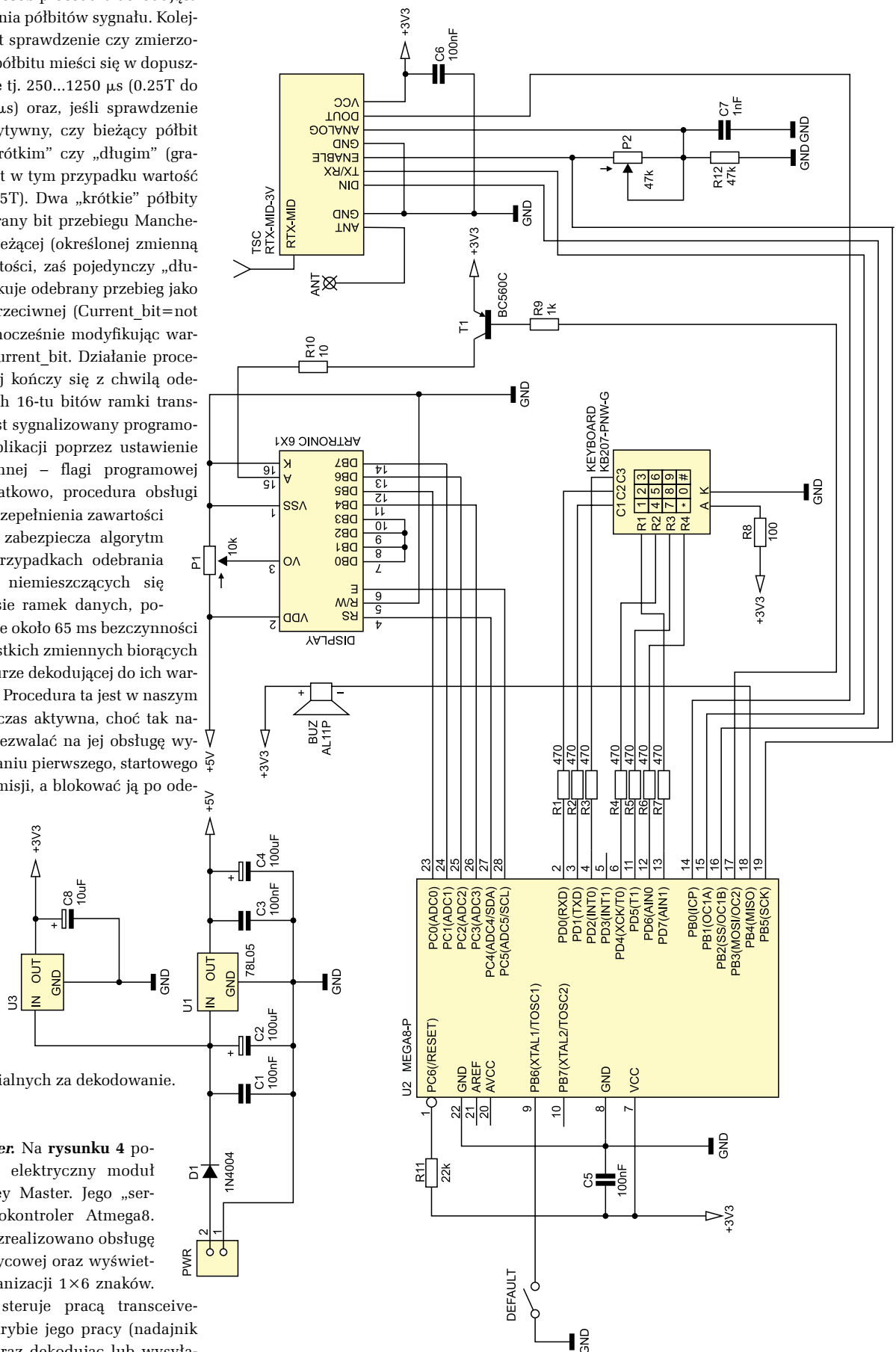
Na **listingu 1** zamieszczono część programu dekodującego przebiegi typu Manchester i treść procedur obsługi przerwania odpowiedzialnych za dekodowanie.

**Budowa**

**Wikey Master.** Na **rysunku 4** pokazano schemat elektryczny modułu klawiatury Wikey Master. Jego „sercem” jest mikrokontroler Atmega8. Za jego pomocą zrealizowano obsługę klawiatury matrycowej oraz wyświetlacza LCD o organizacji 1x6 znaków. Mikrokontroler steruje pracą transceivera decydując o trybie jego pracy (nadajnik lub odbiornik) oraz dekodując lub wysyłając dane. Dodatkowo, dzięki zastosowaniu

wbudowanego w strukturę mikrokontrolera reukładowi czasowo-licznikowemu Timer2 pracującym w trybie generatora PWM

(końcówka OC2/PB3), możliwa stała się realizacja funkcji automatycznej regulacji jasności podświetlenia wyświetlacza alfanu-



Rysunek 4. Schemat ideowy układu Wikey Master

merycznego. Intensywność tego podświetlenia jest ograniczana po około 10 sekundach bezczynności. Przewidziano możliwość wprowadzenia dwóch niezależnych, 4-cyfrowych kodów użytkownika, każdy dedykowany do załączania odpowiedniego przekaźnika sparowanego układu *Slave*. Dodatkowo, dla zapominalskich, wprowadzono przycisk oznaczony „DEFAULT”, służący do przywrócenia domyślnych kodów urządzenia (obydwa kody przyjmują wtedy wartość

Tabela 1. Lista klawiatur dla modułu Wikey Master

Symbol elementu	Opis
KB207-MAS-WP	Klawiatura metalowa, alfanumeryczna, szczelna
KB207-MNS-WP	Klawiatura metalowa
KB207-MNS-WP-G	Klawiatura metalowa, alfanumeryczna, podświetlana LED (zielone podświetlenie)
KB207-PNB	Klawiatura plastikowa, czarna, alfanumeryczna
KB207-PNB-WP	Klawiatura plastikowa, czarna, alfanumeryczna, szczelna
KB207-PNW	Klawiatura plastikowa, biała, alfanumeryczna
KB207-PNW-WP	Klawiatura plastikowa, biała, alfanumeryczna, szczelna
KB207-PNW-WP-G	Klawiatura plastikowa, alfanumeryczna, podświetlana LED (zielone podświetlenie)

## Listing 1. Program obsługi odpowiedzialny za dekodowanie Manchester

```
,Konfiguracja układu czasowo-licznikowego Timer1 służącego dekodowaniu danych typu Manchester
Config Timer1 = Timer , Capture Edge = Falling , Prescale = 8 , 1 tick= 1 uS
,Przerwanie od przechwycenia zawartości licznika Timer1 - oblicza długość każdego impulsu na wejściu ICP
mikrokontrolera
On Capture1 Capture_edges
Enable Capture1
,Przerwanie od przepełnienia zawartości licznika Timer1 - zachodzi, gdy Timer1 zostanie przepełniony w skutek odbioru
niepoprawnych danych
On Ovfl Timeout
Enable Ovfl
Enable Interrupts

Dim First_edge As Byte ,1 -> wystąpiło pierwsze zbocze na wejściu ICP
Dim Current_bit As Bit ,Wartość aktualnego bitu ramki danych - Received
Dim Bit_nr As Byte ,Bieżąca liczba bitów pozostałych do detekcji (na starcie=15)
Dim Ready As Byte ,Wskaźnik gotowej ramki danych - flaga dla pętli głównej aplikacji
Dim Received As Word 'Odebrana ramka danych
Dim Buffered_received_data As Word 'Bufor danych odbieranych w OVFl
Dim Short_pulse_nr As Byte ,Liczba krótkich półbitów o czasie trwania 1/2 T = 500uS

,1 T = 1000 uS -> 1000 taktów Timer1
Const Too_short = 250 '1/4 T
Const Too_long = 1250 '1.25 T
Const Border_line = 750 '3/4 T czyli środek pomiędzy Too_short a Too_long
'Procedura ustawia wartości startowe zmiennych jak i parametry startowe układu Timer1
Sub Reset_received_settings
Tccr1b.ices1 = 0 ,Wyzwolenie przechwycenia przy opadającym zboczach na wejściu ICP
Tifr.icf1 = 1 ,Skasowanie flagi po zmianie aktywnego zbocza - wymaga tego dokumentacja mikrokontrolera
First_edge = 0
Short_pulse_nr = 0
Current_bit = 1
Bit_nr = 15
End Sub

Capture_edges:
If First_edge = 0 Then
,Pierwsze zbocze na wejściu ICP
First_edge = 1
Timer1 = 0
Tccr1b.ices1 = 1 ,Wyzwolenie przechwycenia przy narastającym zboczach na wejściu ICP
Tifr.icf1 = 1 ,Skasowanie flagi po zmianie aktywnego zbocza - wymaga tego dokumentacja mikrokontrolera
Received.15 = 1
Else
,Mierzmy czas impulsu, który upłynął od ostatniego przechwycenia - każde przechwycenie zachodzi przy innym zboczach
Timer1 = 0
Tccr1b.ices1 = Not Tccr1b.ices1 ,Zmiana zbocza powodującego kolejne przechwycenie na odwrotne niż aktualne
Tifr.icf1 = 1 ,Skasowanie flagi po zmianie aktywnego zbocza - wymaga tego dokumentacja mikrokontrolera

,Teraz sprawdzamy czy zmierzony impuls ma długość krótkiego czy długiego półbitu czy też jest spoza zakresu
If Capture1 > Too_short And Capture1 < Too_long Then
If Capture1 < Border_line Then
'Impuls o czasie trwania krótkiego półbitu = 500uS
Incr Short_pulse_nr
If Short_pulse_nr = 2 Then
Decr Bit_nr
Received.bit_nr = Current_bit
Short_pulse_nr = 0
End If
Else
,Impuls o czasie trwania długiego półbitu = 1000uS
Decr Bit_nr
Current_bit = Not Current_bit
Received.bit_nr = Current_bit
Short_pulse_nr = 0
End If
If Bit_nr = 0 Then
,Kompletna ramka danych -> Received
,Zmienna odebranego rozkazu (Received) przepisujemy do zmiennej globalnej używanej w pętli głównej gdyż
,może być ona w każdej chwili zmieniona w procedurze OVFl przez nowo nadchodzące dane
Buffered_received_data = Received
Ready = 1
Call Reset_received_settings
End If
Else
,Impuls o czasie trwania spoza zakresu przyjętego standard
Call Reset_received_settings
End If
End If
Return

Timeout:
Call Reset_received_settings
Return
```

„0000”). Przycisk należy wcisnąć podczas włączania modułu. Przywrócenie kodów domyślnych sygnalizuje napis „C:0000” na wyświetlaczu LCD. Do sygnalizacji stanu

pracy służy brzęczyk piezoelektryczny (tzw. buzzer).

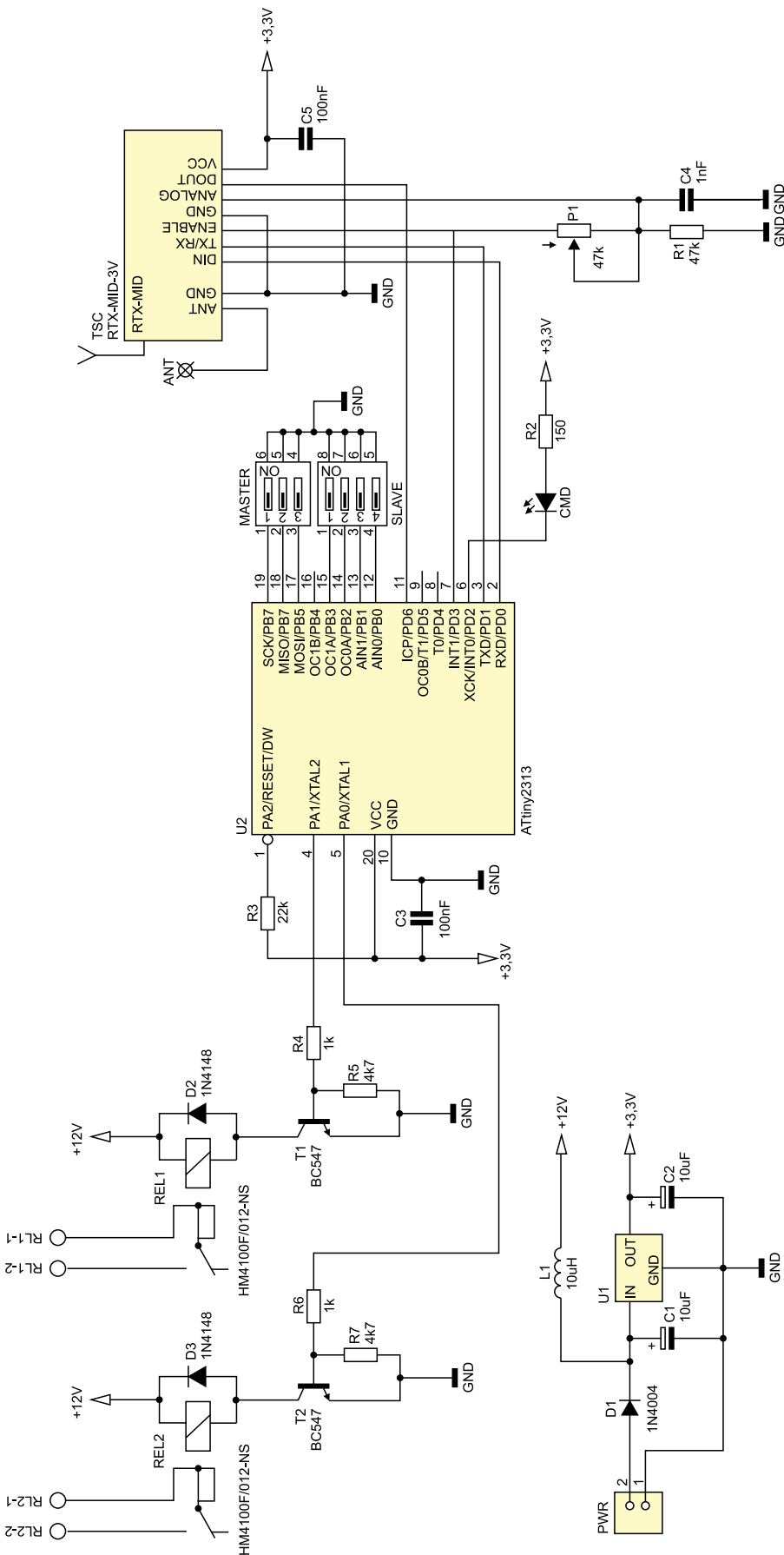
Projekt płytki modułu wykonano w sposób umożliwiający dołączenie różnych klawiatur różniących się położeniem złącza jak i liczbą zacisków (dla wersji z lub bez podświetlenia). Listę typów klawiatur, których można użyć do budowy modułu bez wykonywania zmian płytki drukowanej zawiera **tabela 1**.

Po załączeniu mikrokontroler domyślnie przełącza transceiver do pracy w trybie nadajnika (pin TX/RX=1) przygotowując go tym samym do wysłania ramki danych sterujących. Jedynie po wyemitowaniu wspomnianej ramki danych, moduł transceivera przełączany jest na chwilę w tryb odbiornika (pin TX/RX=0) w celu odebrania oczekiwanej odpowiedzi. Za każdym razem jest jednak zachowywana odpowiednia kolejność podawania sygnałów sterujących oraz czasów ich trwania, co jest ważne z punktu widzenia zapewnienia krótkich czasów przełączania pomiędzy trybami pracy modułu (rysunek 2). Jak wspomniano, wysłanie ramki danych inicjuje prawidłowo wprowadzony kod użytkownika, zaś sam proces powtarzany jest trzykrotnie, jeśli pierwsza próba zakończy się niepowodzeniem (nie odebrano potwierdzenia z układu Slave).

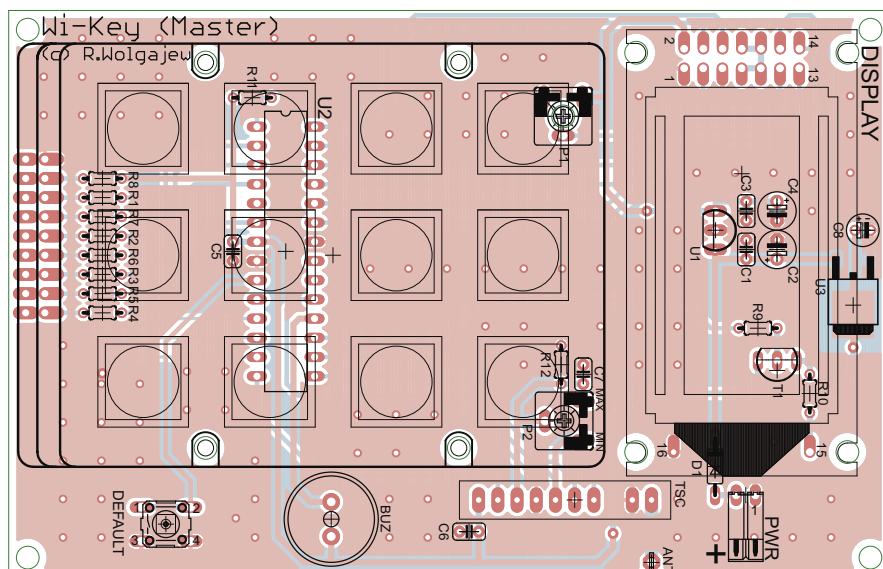
**Wikey Slave.** Schemat modułu wykonawczego pokazano na **rysunku 5**. Jego „sercem” jest mikrokontroler ATtiny2313 odpowiedzialny za obsługę transceivera domyślnie pracującego w trybie odbiornika oraz sterowanie pracą dwóch przekaźników. Do konfiguracji adresów *Master/Slave* przewidziano typowe przełączniki DIP-SWITCH: 3-segmentowy dla adresu *Master* (zakres adresów 0...7) i 4-segmentowy dla adresu *Slave* (zakres 0...15). Przełączenie każdego z segmentów do pozycji ON powoduje wyzerowanie bitu adresu, odpowiednio: M2...M0 dla adresu układu *Master* i S3...S0 dla adresu układu *Slave*. Do sygnalizacji odbioru komendy i jej realizacji przeznaczono diodę świecącą CMD.

Wikey Slave domyślnie pracuje w trybie odbiornika i oczekuje na ważną dla niego ramkę danych sterujących (zawierającą zgodne adresy *Master* i *Slave*). Po otrzymaniu takiej ramki danych, układ *Slave* przełącza transceiver w tryb nadajnika, a następnie natychmiast przesyła potwierdzenie zawierające tę samą ramkę danych, przełącza transceiver w tryb odbiornika i załącza na 3 sekundy odpowiedni przekaźnik mocy (w zależności od bitów numeru przekaźnika P1 i P0).

**Obsługa**  
Ta część opisu systemu Wikey dotyczy wyłącznie układu *Master*, gdyż tylko on ma interfejs użytkownika. Tworząc oprogramowanie starałem się maksymalnie uprościć jego obsługę. Kierując się tym założeniem przyjąłem pewne, jasne reguły funkcjonowania interfejsu użytkownika, które to określają sposób wprowadzania i zmiany kodów urządzenia jak i tryby jego pracy.



Rysunek 5. Schemat ideowy układu Wikey Slave



Rysunek 6. Schemat montażowy układu Wikey Master

Operacja wprowadzania kodu użytkownika rozpoczyna się w chwili przyciśnięcia dowolnego klawisza numerycznego („0”...„9”). Na wprowadzenie całego kodu jest przeznaczony czas 5 sekund. Jeśli w tym czasie nie zostanie wprowadzone 4 cyfry kodu, układ przechodzi do trybu bezczynności wyświetlając komunikat powitalny „Hello!”. Czas ten można skrócić poprzez anulowanie operacji wprowadzania kodu, inicjowane przyciśnięciem klawisza „\*” (tylko i wyłącznie wtedy, gdy wprowadzono co najmniej jedną cyfrę kodu). Wprowadzenie prawidłowego kodu użytkownika powoduje wysłanie ramki sterującej oraz (jeśli zostanie ona prawidłowo odebrana przez sparowany układ *Slave*) załączenie odpowiedniego przekaźnika na czas 3 sekund. Wprowadzenie błędnego kodu użytkownika powoduje zablokowanie modułu *Master* na 20 sekund.

Kolejną operacją, możliwą do wykonania w przypadku układu *Master* jest aktualizacja kodu użytkownika. Operacja ta rozpoczyna się w chwili przyciśnięcia klawisza „\*” w trakcie bezczynności pracy urządzenia. Użytkownik ma następnie 5 sekund na wprowadzenie kodu, który podlega aktualizacji. Jeśli w tym czasie nie zostaną wprowadzone 4 cyfry kodu, układ przechodzi do trybu bezczynności wyświetlając napis powitalny „Hello!”. Po wprowadzeniu prawidłowego, starego kodu użytkownika, układ przechodzi do trybu wpisywania nowego kodu użytkownika, przy czym dopuszczalny czas wprowadzania (5 s) jest liczony od nowa.

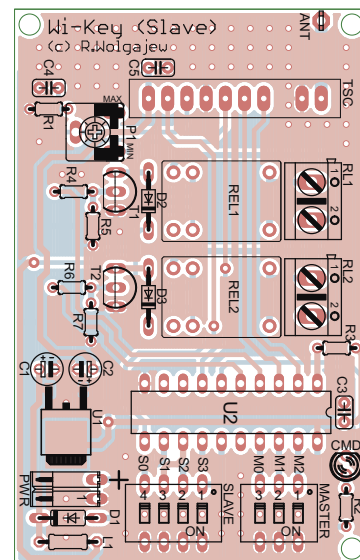
Ostatnią operacją, którą możemy wykonać korzystając z interfejsu użytkownika jest definicja adresów *Master* i *Slave*. Adres *Master* jest adresem naszego modułu klawiatury, natomiast adres *Slave* jest adresem karty przekaźnikowej, do której adresowane będą przesyłane dane sterujące. Warto przypomnieć, iż niedopuszczalne jest istnienie dwóch układów *Slave* o tym samym adre-

sie będących w zasięgu transmisji. Operacja definicji adresów inicjowana jest w chwili przyciśnięcia klawisza „#” w trybie bezczynności pracy urządzenia. Następnie użytkownik musi za pomocą klawiszy numerycznych wprowadzić jedną cyfrę adresu *Master* (0...7) oraz dwie cyfry adresu *Slave* (00...15). Po wprowadzeniu żądanych wartości układ *Wikey-Master* zapisuje je w nieulotnej pamięci mikrokontrolera, a następnie przechodzi do trybu bezczynności wyświetlając napis powitalny „Hello!”. Jak poprzednio, użytkownik ma 5 sekund na wprowadzenie wszystkich informacji. Jeśli w tym czasie nie zostaną wprowadzone 3 cyfry adresu *Master/Slave* układ przechodzi do trybu bezczynności wyświetlając napis powitalny „Hello!”. Czas ten można skrócić poprzez anulowanie operacji wprowadzania kodu inicjowane przyciśnięciem klawisza „\*” (tylko i wyłącznie, gdy wprowadzono co najmniej jedną cyfrę adresu).

### Montaż

Na **rysunku 6** pokazano schemat montażowy modułu klawiatury *Wikey Master*, natomiast na **rysunku 7** modułu wykonawczego *Wikey Slave*.

Opis sposobu montażu dotyczy obu modułów, ponieważ jest wykonywany podobnie. Jak zwykle rozpocząć należy od wlotowania zworek, następnie zamontowania: rezystorów, kondensatorów, przekaźników, złącz, przycisków a na końcu półprzewodników. Zależnie od typu zastosowanego zamka, może być konieczne pocynowanie ścieżek wiodących do zestyków wykonawczych przekaźników REL1 i REL2 modułu *Wikey Slave*. Wyświetlacz LCD i klawiaturę należy zamocować w odpowiedniej odległości od obwodu drukowanego układu *Wikey-Master*, najlepiej za pomocą tulei dystansowych wykorzystując przewidziane w tym celu otwory, zaś same połączenie należy



Rysunek 7. Schemat montażowy układu Wikey Slave

wykonać przy użyciu listwy goldpin (gniazdo-wtyk) lub taśmy wieloprzewodowej. Długość tulei dystansowych dla klawiatury matrycowej to 9 mm, natomiast dla wyświetlacza 14 mm. Niezależnie od zastosowanych rozwiązań montażowych należy pamiętać by wspomniane elementy ustawić w takiej odległości od obwodu drukowanego, by górna powierzchnia wyświetlacza była na równi z powierzchnią klawiatury, co umożliwi dopasowanie do płyty czołowej panelu użytkownika.

W większości przypadków zadowalającą pracę klawiatury (odporność programu obsługi na drgania styków) zapewniają rezystory R1...R7 o wartościach 0  $\Omega$  (zwykle zworki). W przypadku niepotrzebnych zadziałań klawiatury wartość tych rezystorów należy dobrać z zakresu 470  $\Omega$ ...1 k $\Omega$ .

Do wykonania anteny można użyć kawałka drutu miedzianego o przekroju 0,5 mm<sup>2</sup> i długości około 17 cm. Długość takiej anteny można zmniejszyć do około 8,3 cm formując cewkę powietrzną położoną u podstawy anteny. Anteny nadajnika i odbiornika powinny znajdować się w położeniu prostokątym do powierzchni masy poszczególnych płytek drukowanych. Wzajemne umiejscowienie anten wpływa na zasięg kompletu urządzeń. Można także skorzystać z innych, dostępnych w handlu anten o impedancji 50  $\Omega$  przeznaczonych do pracy w paśmie ISM 433 MHz.

Potencjometry P1 (na płytce układu *Slave*) i P2 (na płytce układu *Master*) służą do korekty czułości toru odbiornika scalonego transceivera RTX-MID-3V w przypadku pracy na obszarze o dużych zakłóceniach. Należy jednak dążyć do uzyskania maksymalnej czułości obu układów, gdyż ma to bezpośredni wpływ na zasięg i pewność transmisji.

**Robert Woigajew, EP**  
robert.woigajew@ep.com.pl