



Podstawowe parametry:

- komunikacja z potwierdzaniem poprawności transmisji,
- moduł nadawczy zasilany z baterii CR2016 (teoretyczny czas pracy na jednej baterii: 10 lat),
- maksymalny prąd obciążenia (tryb uśpienia/nadawanie): 0,2 μ A / 22,6 mA,
- zasilanie odbiornika: 230 V AC, ok. 2 W,
- 12 wyjść przełącznikowych,
- maksymalne obciążenie wyjść: 5 A/250 VAC
- częstotliwość pracy transceiwera: 868 MHz
- zasięg w terenie otwartym: ok. 100 m

* **Uwaga!** Elektroniczne zestawy do samodzielnego montażu. Wymagana umiejętność lutowania! Podstawową wersją zestawu jest wersja **[B]** nazywana potocznie KIT-em (z ang. zestaw). Zestaw w wersji **[B]** zawiera elementy elektroniczne (w tym **[UK1]** – jeśli występuje w projekcie), które należy samodzielnie wylutować w dołączoną płytkę drukowaną (PCB). Wykaz elementów znajduje się w dokumentacji, która jest podlinkowana w opisie kitu. Mając na uwadze różne potrzeby naszych klientów, oferujemy dodatkowe wersje:

Dodatkowe materiały do pobrania ze strony www.ulubionykiosk.pl/media

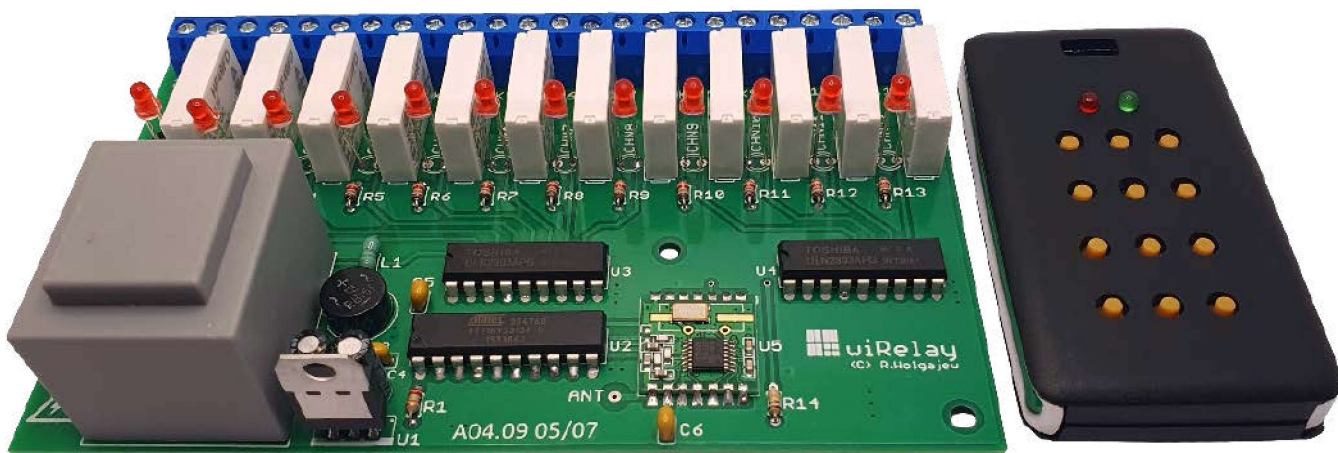
- AVT5960 Przełącznik elektromagnetyczny sterowany optoelektronicznie (EP 11/2022)
- AVT5876 Energooszczędny przełącznik bistabilny (EP 8/2021)
- AVT5794 Moduł przełącznikowy z gasikami (EP 8/2020)
- AVT5710 8-kanatowy moduł przełącznikowy z USB (EP 8/2019)
- AVT5682 Przełącznik elektromagnetyczny 230 V sterowany optoelektronicznie (EP 6/2019)
- AVT5632 Moduł przełączników z interfejsem USB (EP 3/2019)
- AVT5588 Sterownik-timer z 8 przełącznikami (EP 6/2017)
- AVT1916 Konfigurowalny przełącznik 4-kanatowy (EP 8/2016)
- AVT1890 Moduł przełączników z USB (EP 6/2016)
- AVT1815 4-kanatowy przełącznik sterowany dowolnym pilotem IR (EP 8/2014)
- AVT5368 Programowalny moduł przełączników (EP 11/2012)
- AVT5353 Moduł przełączników z interfejsem USB (EP 7/2012)
- AVT1691 Uniwersalny moduł przełącznikowy (EP 8/2012)
- AVT1659 8-kanatowy miniaturowy moduł przełączników (EP 1/2012)
- AVT1656 Uniwersalny moduł wykonawczy (EP 12/2011)
- AVT1560 8-kanatowa karta przełączników (EP 2/2010)

- wersja **[C]** – zmontowany, uruchomiony i przetestowany zestaw **[B]** (elementy wylutowane w płytkę PCB),
 - wersja **[A]** – płytka drukowana bez elementów i dokumentacji.
- Kity, w których występuje układ scalony wymagający zaprogramowania, mają następujące dodatkowe wersje:
- wersja **[A+]** – płytka drukowana **[A]** + zaprogramowany układ **[UK1]** i dokumentacja,
 - wersja **[UK]** – zaprogramowany układ.

Nie każdy zestaw AVT występuje we wszystkich wersjach! Każda wersja ma załączony ten sam plik PDF! Podczas składania zamówienia upewnij się, którą wersję zamawiasz! <http://sklep.avt.pl>

W przypadku braku dostępności na stronie sklepu osoby zainteresowane zakupem płytek drukowanych (PCB) prosimy o kontakt via e-mail: kity@avt.pl.

W ofercie AVT*
AVT5966



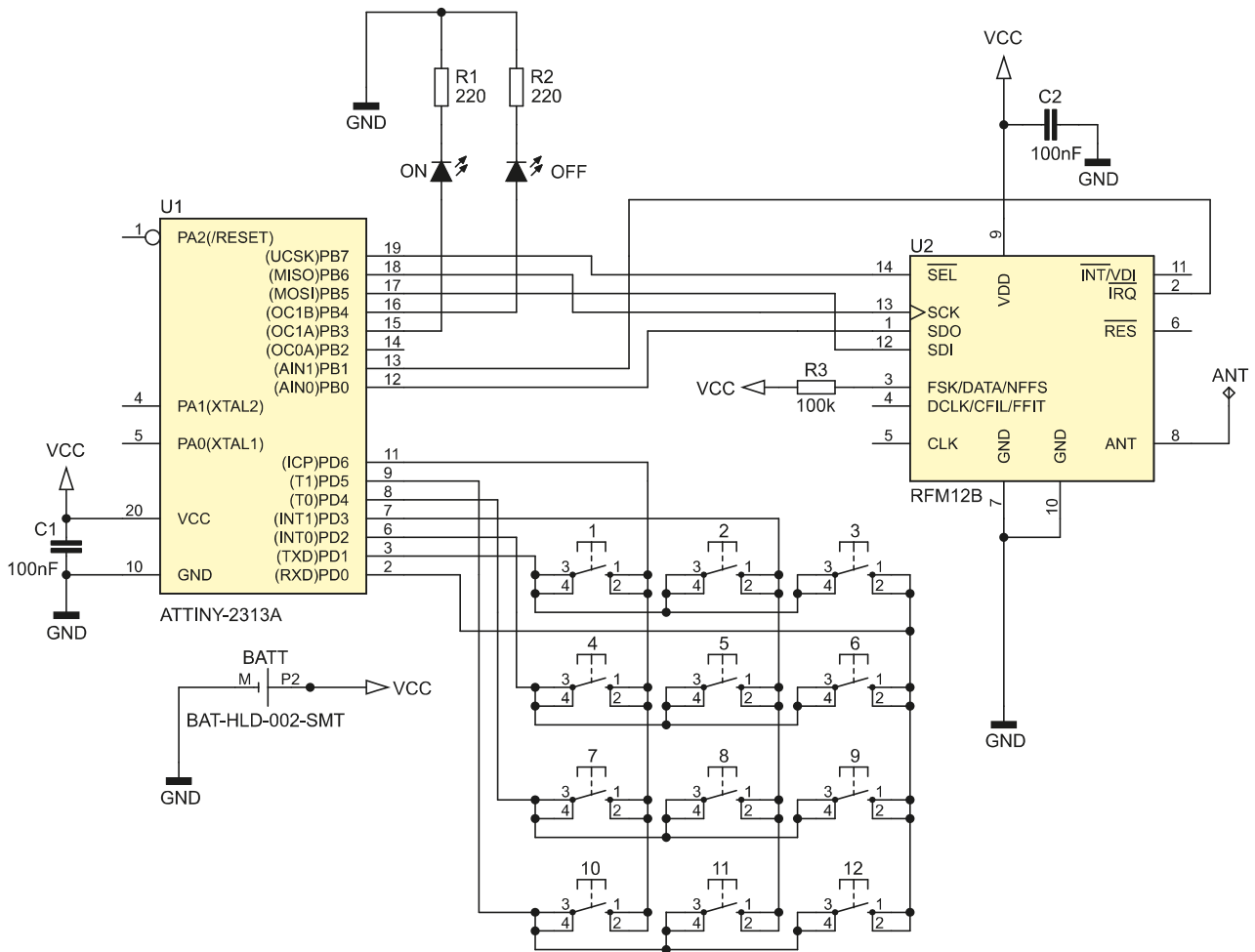
wiRelay – bezprzewodowa, 12-kanalowa karta przełączników

Tym razem, zupełnie inaczej, niż miało to miejsce zazwyczaj, pomysł na niniejszy projekt narodził się w głowie...mojego kolegi z redakcji. Zaproponowano mi zaprojektowanie 12-kanalowej, bezprzewodowej karty przełączników, gdyż, jak się okazuje, tego rodzaju urządzenia dość często znajdują zastosowanie w gospodarstwach domowych, zwłaszcza wtedy, gdy zamieszkujemy domek jednorodzinny. Co prawda w swoim portfolio mam już urządzenie o podobnej funkcjonalności, lecz tym razem postanowiłem zbudować system o większej uniwersalności i lepszych cechach użytkowych.

Kompletny system składa się z nadajnika i odbiornika rozkazów sterujących. Głównym problemem implementacyjnym było znalezienie sprawdzonego i energooszczędnego modułu do bezprzewodowej transmisji danych. Co oczywiste, zależało mi na tym,

aby nadajnik tego systemu pracował na zasilaniu bateryjnym i charakteryzował się minimalnym poborem mocy zapewniającym długą pracę urządzenia. Ponieważ w moim poprzednim projekcie energooszczędnego systemu pomiaru temperatury z EP 8/2018 [1]

i EP 9/2018 [2] zdobyłem duże doświadczenie w zakresie obsługi bardzo ciekawych modułów RFM12B pracujących w paśmie 433, 868 lub 915 MHz (w zależności od wersji), należących do bogatej oferty modułów radiowych produkowanych przez firmę HopeRF, zdecydowałem się na ich zastosowanie również w tej konstrukcji. Moduły, o których mowa stanowią kompletne rozwiązanie toru radiowego nadawczo-odbiorczego. Dostarczają wygodny interfejs komunikacyjny SPI pozwalający na przeprowadzenie pełnej konfiguracji elementu w ramach dostępnej szerokiej palety ustawień jak i sterowanie komunikacją radiową. Nie będę powtarzał informacji dotyczących specyfikacji i obsługi tych peryferiów, gdyż takowe



Rysunek 1. Schemat ideowy nadajnika systemu wiRelay

zamieściłem w ramach wspomnianych wcześniej artykułów. Zainteresowanych tymi szczegółami Czytelników odsyłam do wskazanych wcześniej materiałów. Zanim jednak przejdę do schematów urządzenia kilka słów napiszę na temat specyfikacji całego systemu.

Tak, jak wspomniano powyżej, system wiRelay składa się z dwóch modułów komunikacyjnych: nadawczego (pilota), który obsługuje klawiaturę sterującą i odbiorczego (karty przekaźników), który steruje wbudowanymi przekaźnikami dużej mocy. Moduł nadawczy pracuje na zasilaniu bateryjnym w postaci pastylki CR2016 i większość swojego czasu spędza w trybie uśpienia (dla ograniczenia poboru mocy) czekając na zmianę stanu obsługiwanej klawiatury matrycowej. Wspomnianej zmianie stanu towarzyszy wybudzenie

układu (mikrokontrolera i modułu RF), wysłanie komunikatu do karty przekaźników, odbiór odpowiedzi i ponowne uśpienie urządzenia. W ten prosty sposób ograniczamy do minimum pobór energii ze źródła zasilania pozwalając na wieloletnią pracę urządzenia.

Uważny Czytelnik dostrzeże pewne ograniczenia i sformułuje związane z nimi zapytania. Otóż bateria CR2016 (3 V) zastosowana w urządzeniu przeznaczona jest do zasilania urządzeń cechujących się bardzo niskim poborem prądu – rzędu ułamków mA, do pojedynczych mA. Nasz układ po wybudzeniu aktywuje nadajnik modułu RFM12B, który w czasie transmisji pobiera prąd rzędu 22,6 mA, co stanowi bardzo duże obciążenie dla niewielkiej baterii zasilającej. Na szczęście transmisja trwa około

20 ms, w związku z czym pobrana energia jest bardzo mała, niemniej jednak takie obciążenie skromnej baterii ma pewne reperkusje. Po pierwsze – z czasem spada jej znamionowa pojemność, napięcie znamionowe oraz wzrasta rezystancja wewnętrzna. Spadek pojemności nie jest jakiś drastycznie wielki, ale można go szacować na 25%, przy spadku napięcia baterii do 2,2 V. Zagadnienie jest naprawdę bardzo ciekawe, w związku z czym zachęcam ambitnych Czytelników do zgłębienia tematu, który został omówiony na stronie [3], gdzie inżynierowie firmy Energizer i Nordic Semiconductor bardzo drobiazgowo wyjaśnili ten interesujący temat w dokumencie o nazwie „High pulse drain impact on CR2032 coin cell battery capacity”. Na szczęście główne podzespoły nadajnika

Wykaz elementów, kupuj na stronie sklep.avt.pl (Warszawa, ul. Leszczyńska 11, tel. +48222578451, e-mail: handlowy@avt.pl)

Karta przekaźników

Rezystory: (miniaturowe 1/8 W, raster 0,2")

R1: 22 kΩ
R2...R13: 1,8 kΩ
R14: 100 kΩ

Kondensatory:

C1, C3, C5, C6: 100 nF ceramiczny (raster 0,1")
C2, C4: 100 µF/16 V elektrolityczny (5 mm, raster 2,5 mm)

Półprzewodniki:

U1: LD1117V33 (TO-220)
U2: ATtiny2313A (DIL20)
U3, U4: ULN2803 (DIL18)
U5: RFM12B-866MHz (SMD)

B1: mostek prostowniczy 1 A (okrągły, raster 0,2")

CHN1...CHN12: czerwona dioda LED 3 mm

Pozostałe:

L1: dławik osiowy 10 µH (raster 0,2")
TR1: transformator do druku TEZ2.5/D/9V
K1...K12: przekaźnik AZ921-1A-12DE ZETTLER (JZC-49F-12V)
AC: złącze śrubowe AK500/2
OUT1...OUT3: złącze śrubowe AK500/8

Pilot

Rezystory: (SMD0805)
R1, R2: 220 Ω
R3: 100 kΩ

Kondensatory: (SMD0805)

C1, C2: 100 nF ceramiczny X7R

Półprzewodniki:

U1: ATtiny2313A (SO20)
U2: RFM12B-866MHz (SMD)
ON: czerwona dioda LED (SMD1206)
OFF: zielona dioda LED (SMD1206)

Pozostałe:

BATT: koszyk baterii CR2016 SMT typu BAT-HLD-002-SMT
1...12: microswitch SMD typu KSC411J 70 SH LFS C&K (wysokość 5,2 mm)
OBUDOWA: plastikowa do pilota typu 13120.44 TEKO

pracują już przy niewielkich napięciach zasilających (moduł RF: 2,2 V, mikrokontroler: 1,8 V) i nawet przy 25% spadku pojemności, tego typu aplikacja powinna zapewnić wieloletnią pracę urządzenia.

Aby ocenić jak długo nadajnik pracował będzie na pojedynczej baterii CR2016 należy zastanowić się z jakich etapów składa się jego praca i jakie są wtedy prądy pobierane ze źródła napięcia zasilającego. Przystępując do obliczeń przyjąłem następujący podział cyklu pracy urządzenia:

- etap trybu power-down (uśpienie), który trwa z dużym przybliżeniem 24 h/dobę i podczas którego pobierany jest prąd rzędu 0,2 µA,
- etap transmisji modułu RFM12B (po wybudzeniu), który trwa średnio 20 ms i podczas którego pobierany jest prąd rzędu 22,6 mA.

Założono ponadto, iż wybudzenie nadajnika, a więc zmiana stanu podłączonej klawiatury następuje 15 razy na dobę. Przy tych założeniach otrzymano teoretyczny, ponad 62-letni czas pracy na pojedynczej baterii CR2016 (o zredukowanej o 25% pojemności) co wydaje się wartością niespotykaną i grubo przekraczającą czas życia samej baterii, standardowo przyjmowany, jako 10 lat.

Uważny Czytelnik dostrzeże pewną subtelność w opisie sposobu działania nadajnika (pilota) systemu wiRelay. Otóż napisałem, że pilot, oprócz nadania rozkazu, odbiera również odpowiedź wysłaną przez kartę przekaźników. Tak, to prawda. Oba moduły w istocie przeprowadzają dwustronną komunikację wyłącznie po to, aby po stronie pilota sterującego pozyskać informację na temat wykonania rozkazu sterującego (zmiany stanu sterowanego przekaźnika). Do tego rodzaju potwierdzenia służą diody LED zamontowane na obwodzie drukowanym nadajnika (pilota). Oczywiście tego rodzaju rozwiązanie nie daje 100% pewności co do faktu wykonania rozkazu sterującego, gdyż nie trudno wyobrazić sobie sytuację, gdzie rozkaz wysłany przez nadajnik został w istocie wykonany, zaś stosowne potwierdzenie wysłane przez odbiornik nie dotarło do nadajnika przez co zasygnalizował on (zupełnie błędnie) niewykonanie tegoż rozkazu sterującego, lecz moim zdaniem lepsze jest tego rodzaju rozwiązanie, niż brak jakiegokolwiek potwierdzenia po stronie nadajnika. Z resztą mechanizm ten jest niezbędny wyłącznie wtedy, gdy naocznie nie jesteśmy w stanie przekonać się o tym, iż sterowany przez nas przekaźnik (a więc podłączony do niego odbiornik) nie zmienił stanu pracy. Sama karta przekaźników pozostaje, co oczywiste, w nasłuchu i po otrzymaniu rozkazu sterującego zmienia stan związanego z nim przekaźnika (w przypadku braku błędów) po czym odsyła potwierdzenie do odbiornika (pilota). Warto również podkreślić, iż transmisja pakietów danych

między stronami komunikacji oparta jest stosowną sumą kontrolną CRC8 minimalizując ryzyko potencjalnych błędów. Tyle w kwestii funkcjonowania systemu.

Budowa i działanie

Pora na zaprezentowanie schematu ideowego nadajnika, który został pokazany na **rysunku 1**. Jak widać, zaprojektowano bardzo prosty system mikroprocesorowy, którego sercem jest niewielki mikrokontroler ATtiny2313A (niskonapięciowy) taktowany wewnętrznym oscylatorem RC o częstotliwości 1 MHz sterujący pracą modułu transceivera, dzięki realizacji programowej obsługi interfejsu SPI oraz obsłudze przerwania zewnętrznego Pin Change Interrupt 0 (wyprowadzenie PB1) odpowiedzialnego za mechanizm wysyłania i odbierania danych. Ponadto mikrokontroler obsługuje klawiaturę matrycową utworzoną z przycisków 1...12, której obsługa realizowana jest z użyciem przerwania zewnętrznego Pin Change Interrupt 2 (wyprowadzenia PD0...PD6) inicjującego wybudzenie mikrokontrolera (i modułu RF) oraz obsługuje dwie diody LED (ON i OFF) służące sygnalizacji wykonania wysłanych rozkazów sterujących.

Zgodnie z tym, co napisano wcześniej, nasz nadajnik powinien charakteryzować się minimalnym zapotrzebowaniem na energię elektryczną, jako że jest zasilany niewielką baterią CR2016. W związku z powyższym zastosowano następujące mechanizmy programowo-sprzętowe:

- wyłączono wszystkie nieużywane peryferia mikrokontrolera (komparator analogowy, TIMER1, TIMER0, USI, USART),

Listing 1. Fragment pliku nagłówkowego związany z obsługą klawiatury matrycowej

```
//Definicje portu klawiatury
#define KEY_PORT PORTD
#define KEY_PIN PIND
#define KEY_DDR DDRD
#define COL1 (1<<PD6)
#define COL2 (1<<PD3)
#define COL3 (1<<PD0)
#define ROW1 (1<<PD1)
#define ROW2 (1<<PD2)
#define ROW3 (1<<PD4)
#define ROW4 (1<<PD5)

//Definicje portu led
#define LED_PORT PORTB
#define LED_DDR DDRB
#define LED_RED PB3
#define LED_GREEN PB4
#define LED_RED_ON
    LED_PORT |= (1<<LED_RED)
#define LED_GREEN_ON
    LED_PORT |= (1<<LED_GREEN)
#define LED_BOTH_ON
    LED_PORT |= ((1<<LED_GREEN)|
        (1<<LED_RED))
#define LED_BOTH_OFF
    LED_PORT &= ~(1<<LED_GREEN)|
        (1<<LED_RED)

//Definicje rozkazów sterujących
#define RELAY_UNDEFINED 0x00
#define RELAY_ON 0b11101011
#define RELAY_OFF 0b10101111

//Timeout - ms
#define TIMEOUT 100
```

Listing 2. Fragment funkcji main w zakresie konfiguracji obsługi klawiatury matrycowej

```
//Podciągnięcie wierszy klawiatury pod VCC
KEY_PORT |= ROW1|ROW2|ROW3|ROW4;
//Kolumny klawiatury jako wyjścia
//ze stanem 0 - przygotowanie do przerwania
KEY_DDR |= COL1|COL2|COL3;
//Konfiguracja i uruchomienie
//przerwania zewnętrznego obsługi klawiatury
//Zmiana stanu na wierszach klawiatury
//powoduje przerwanie
PCMSK2 = ROW1|ROW2|ROW3|ROW4;
GIMSK |= (1<<PCIE2);
```

Listing 3. Funkcja obsługi przerwania odpowiedzialna za obsługę klawiatury matrycowej

```
//Dane przeznaczone do wysyłki: Rozkaz i numer przekaźnika
volatile char dataToSend[2];
//Tablica kolumn
const uint8_t Cols[3] = {COL1, COL2, COL3};
//Tablica wierszy
const uint8_t Rows[4] = {ROW1, ROW2, ROW3, ROW4};

ISR(PCINT2_vect){
    //Wyłączenie przerwania klawiatury
    GIMSK &= ~(1<<PCIE2);
    //Wszystkie kolumny ustawiamy
    //na początek jako wejścia
    KEY_DDR &= ~(COL1|COL2|COL3);

    //Ustalamy numer wciśniętego klawisza
    for(uint8_t Col=0; Col<3; Col++){
        //Wybraną kolumnę ustawiamy,
        //jako wyjściową ze stanem 0
        KEY_DDR |= Cols[Col];
        //Szukamy czy nie zwarto jej
        //do wybranego wiersza
        for(uint8_t Row=0; Row<4; Row++){
            if(!(KEY_PIN & Rows[Row]))
                dataToSend[1] = Col + (Row*3);
        }
        //Wybraną kolumnę ustawiamy,
        //jako wejściową
        KEY_DDR &= ~Cols[Col];
    }

    //Wszystkie kolumny ustawiamy ponownie,
    //jako wyjścia ze stanem 0 - przygotowanie do przerwania
    KEY_DDR |= COL1|COL2|COL3;

    //Opóźnienie by ocenić czy krótkie,
    //czy długie naciśnięcie klawisza
    _delay_ms(300);
    //Sprawdzamy czy klawisz nadal jest naciśnięty
    if(!(KEY_PIN & ROW1) || !(KEY_PIN & ROW2) ||
        !(KEY_PIN & ROW3) || !(KEY_PIN & ROW4))
        dataToSend[0] = RELAY_ON;
    else dataToSend[0] = RELAY_OFF;
}
```

Ustawienia Fuse-bitów obu modułów:

```
CKSEL3...0: 0100
SUT1...0: 10
CKDIV8: 0
CKOUT: 1
```

- wprowadzono mikrokontroler w tryb niskiego poboru mocy Power-down, z którego wybudzany jest wyłącznie poprzez zmianę stanu klawiatury matrycowej. Wybudzony mikrokontroler przeprowadza transmisję danych, po czym przechodzi ponownie w tryb Power-down,
- nieużywany transceiver RFM12B wprowadzany jest każdorazowo w tryb niskiego poboru mocy.

Dzięki tym rozwiązaniom otrzymano wspomniane wcześniej wartości zapotrzebowania na energię dla całego urządzenia oraz teoretyczne czasy jego działania. Na koniec kilka słów komentarza należy się implementacji obsługi klawiatury matrycowej realizowanej w funkcji przerwania zewnętrznego *Pin Change Interrupt 2* (wyprowadzenia PD0...PD6). Zaczniemy zatem od pliku nagłówkowego porządkującego ustawienia sprzętowe w tym zakresie, który to pokazano na **listingu 1**. Dalej, na **listingu 2** pokazano fragment funkcji *main* w zakresie podstawowej konfiguracji obsługi klawiatury matrycowej. I na koniec funkcja obsługi przerwania odpowiedzialna za obsługę klawiatury matrycowej, której ciało pokazano na **listingu 3** (wraz z deklaracją niezbędnych zmiennych globalnych upraszczających ciał funkcji). Funkcja ta ustawia stosowne wartości zmiennej *dataToSend[]*, które transmitowane są do karty przekaźników.

Za samą transmisję danych do karty przekaźników, jak i odbiór i sygnalizację potwierdzenia odpowiedzialny jest fragment funkcji *main* pokazany na **listingu 4**. Z kolei za sygnalizację potwierdzenia wysłanego przez kartę przekaźników odpowiada prosta funkcja pokazana na **listingu 5**.

Tyle w kwestii nadajnika, przejdźmy zatem do schematu ideowego karty przekaźników, który pokazano na **rysunku 2**. Tak jak poprzednio, zaprojektowano bardzo prosty system mikroprocesorowy, którego sercem jest niewielki mikrokontroler ATtiny2313A (niskonapięciowy) taktowany wewnętrznym oscylatorem RC o częstotliwości 1 MHz sterujący pracą modułu transceiwera dzięki realizacji programowej obsługi interfejsu SPI oraz obsłudze przerwania zewnętrznego *Pin Change Interrupt 2* (wyprowadzenie PD6) odpowiedzialnego za mechanizm odbierania i wysyłania danych. Ponadto mikrokontroler ten steruje pracą 12 przekaźników dużej mocy a dokonując tego dzięki zastosowaniu scalonych sterowników pod postacią układów ULN2803 (U3, U4). W odróżnieniu jednak od układu nadajnika, w ramach aplikacji odbiornika zaprojektowano niewielki, kompletny układ

zasilający przeznaczony do podłączenia do sieci 230 VAC. Było to niezbędne z uwagi na fakt, iż moduł odbiornika steruje pracą przekaźników, z których każdy w czasie załączenia pobiera prąd rzędu 10 mA. Pomimo tego sumaryczny pobór mocy urządzenia w przypadku załączenia wszystkich przekaźników nie przekracza 2 W.

Przejdźmy zatem to szczegółów implementacyjnych, jeśli chodzi o program obsługi aplikacji. Prawdę mówiąc nie ma tego zbyt wiele, gdyż wspomniany program jest niezmiernie prosty. Na początek fragment funkcji *main* odpowiedzialny za odbiór rozkazu sterującego i odesłanie odpowiedzi pokazany na **listingu 6**. I na sam koniec funkcja odpowiedzialna za sterowanie przekaźnikami, której ciało pokazano na **listingu 7**.

Listing 4. Fragment funkcji *main* odpowiedzialny za transmisję danych do karty przekaźników, jak i odbiór i sygnalizację potwierdzenia

```
//Wychodzimy z trybu powerDown modułu RFM12B
RFM12bPowerUp();

//Uruchomienie przerwania odpowiedzialnego
// za nadawanie i odbieranie
GIFR |= (1<<PCIF0);
GIMSK |= (1<<PCIE0);

//Wysyłamy ramkę danych do odbiornika - 11ms
RFM12bStartTx((char*) dataToSend, 2, 0xFF);
//Czekamy na zakończenie transmisji
while(RFM12B.Status != PACKET_SENT);
_delay_us(100);

//Przygotowujemy się na odbiór potwierdzenia
RFM12bStartRx();

Timeout = 0;
while(1){
    //Odebrano poprawną ramkę danych:
    //Size|Rozkaz|CRC8
    if(RFM12B.Status == NEW_PACKET){
        //Sprawdzamy, czy odebrane potwierdzenie
        //jest takie samo, jak wysłany rozkaz
        if(RFM12B.Buffer[1] == dataToSend[0])
            blinkLED(dataToSend[0]);
        else blinkLED(RELAY_UNDEFINED);
        break;
    }
    //Odebrano błędną ramkę danych: sygnalizujemy błąd
} else if(RFM12B.Status == FAULTY_PACKET){
    blinkLED(RELAY_UNDEFINED);
    break;
}
//Sprawdzamy czy upłynął maksymalny czas
//na odbiór potwierdzenia
if(++Timeout > TIMEOUT){
    blinkLED(RELAY_UNDEFINED);
    break;
}
_delay_ms(1);
}

//Wyłączenie przerwania odpowiedzialnego
//za nadawanie i odbieranie
GIMSK &= ~(1<<PCIE0);

//Wprowadzamy moduł RFM12B
//w tryb powerDown
RFM12bPowerDown();
```

Listing 5. Funkcja odpowiedzialna za sygnalizację potwierdzenia wysłanego przez kartę przekaźników

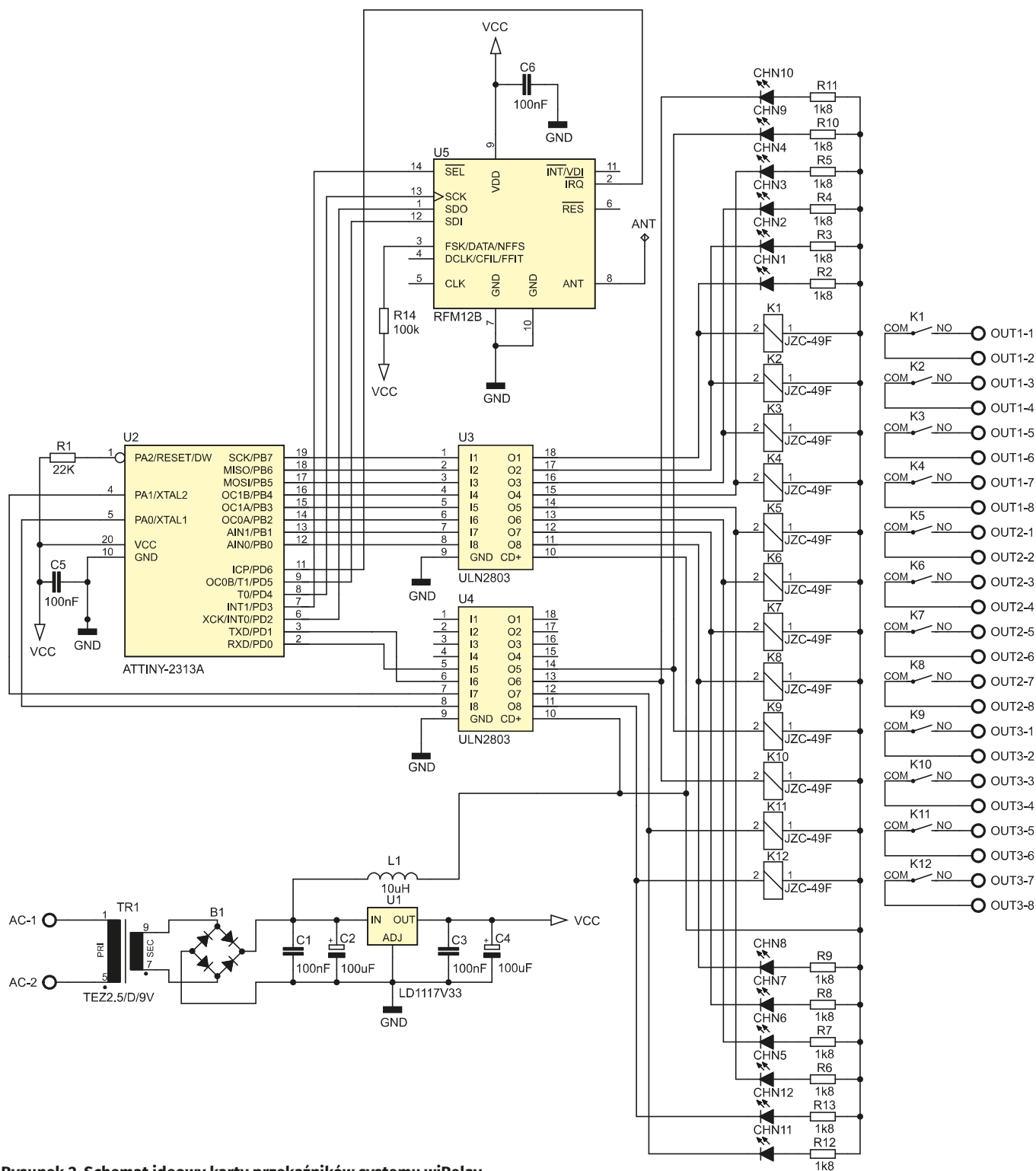
```
void blinkLED(uint8_t Led){
    switch(Led){
        case RELAY_ON: LED_RED_ON; break;
        case RELAY_OFF: LED_GREEN_ON; break;
        case RELAY_UNDEFINED: LED_BOTH_ON; break;
    }
    _delay_ms(200);
    LED_BOTH_OFF;
}
```

Listing 6. Fragment funkcji *main* odpowiedzialny za odbiór rozkazu sterującego i odesłanie odpowiedzi

```
//Uruchomienie i konfiguracja RFM12B,
//w tym interfejsu SPI oraz przerwania nadawania/odbioru
RFM12bInit(0xFF);
//Przygotowujemy moduł na odbiór danych
RFM12bStartRx();

sei();
while(1){
    //Odebrano poprawną ramkę danych:
    //Size|Rozkaz|Przekaźnik|CRC8
    if(RFM12B.Status == NEW_PACKET){
        //Sterujemy przekaźnikiem
        ctrlRelay(RFM12B.Buffer[1], RFM12B.Buffer[2]);
        //Odsyłamy potwierdzenie
        //w postaci odebranego rozkazu
        Ack = RFM12B.Buffer[1];

        _delay_ms(1);
        //Wysyłamy potwierdzenie
        //do odbiornika - 10ms
        RFM12bStartTx(&Ack, 1, 0xFF);
        //Czekamy na zakończenie transmisji
        while(RFM12B.Status != PACKET_SENT);
        _delay_ms(1);
        //Przygotowujemy moduł
        //na kolejny odbiór danych
        RFM12bStartRx();
    }
    //Odebrano błędną ramkę danych:
    //restartujemy procedurę odbiorczą,
    //bo inaczej nastąpiłaby blokada odbioru
    else if(RFM12B.Status == FAULTY_PACKET)
        RFM12bStartRx();
}
}
```



Rysunek 2. Schemat ideowy karty przełączników systemu wiRelay

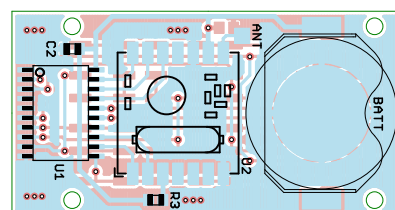
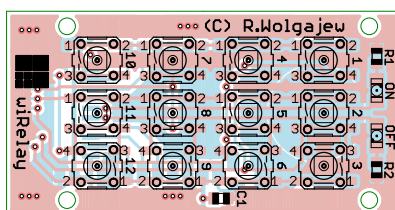
Tyle w kwestiach implementacyjnych dotyczących obu układów.

Montaż i uruchomienie

Przejdźmy zatem do schematu montażowego nadajnika, który pokazano na rysunku 3. Jak widać zaprojektowano bardzo niewielki obwód drukowany z wyłącznym montażem elementów SMD po obu stronach laminatu. Projektując nadajnik systemu wiRelay, jak i jego obwód drukowany chciałem, aby docelowe urządzenie wyposażone było w gęstą i niewielką obudowę,

przez co etapem wyjściowym w procesie projektowania było znalezienie atrakcyjnej wizualnie i łatwo dostępnej obudowy. Zdecydowałem się na zastosowanie smukłej,

plastikowej obudowy do pilota typu 13120.44 firmy TEKO w wersji bez żadnych przycisków sterujących, gdyż zakładałem zastosowanie dedykowanych switchy i nawiercenie



Rysunek 3. Schemat płytki PCB nadajnika systemu wiRelay

Uwaga! Na płycie modułu karty przekaźników zamontowano kompletny zasilacz łącznie z transformatorem zasilanym napięciem sieciowym 230 V AC oraz zamontowano elementy będące na potencjale tego napięcia. Istnieje niebezpieczeństwo porażenia prądem elektrycznym, co może stanowić zagrożenie dla życia i zdrowia użytkowników. W związku z tym, montaż układu oraz jego uruchomienie należy wykonywać pod nadzorem osoby z odpowiednimi uprawnieniami lub mającej niezbędną wiedzę i doświadczenie. Natomiast gotowe urządzenie musi być właściwie zabezpieczone – np. umieszczone w nieprzewodzącej obudowie.

stosownych otworów (na osie switchy i diody LED) w płycie czołowej (górnej) obudowy. W związku z powyższym cały projekt laminatu podporządkowany został wymiarom zastosowanej obudowy. Co więcej, z uwagi na fakt, iż zastosowany typ obudowy umożliwia zamontowanie w nim płytki z elementami o maksymalnej, sumarycznej grubości ok. 7,2 mm musiałem zdecydować się na zastosowanie bardzo niskich switchy SMD oraz wyjątkowo niskiego koszyczka baterii zasilającej. Poniekąd z tego tytułu wynika konieczność zastosowania cienkiej baterii CR2016, zamiast zdecydowanie bardziej popularnej CR2032 (o ponad 2 krotnie większej pojemności). Osoby, które nie planują zastosowania obudowy, o której mowa powyżej, mogą nie zważać na poniższe ograniczenia.

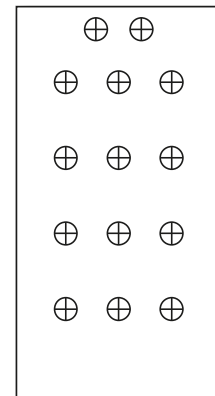
Przejdźmy zatem do szczegółów montażowych dotyczących nadajnika. Montaż obudowy drukowanego nadajnika rozpoczynamy od warstwy BOTTOM, na której przylutujemy mikrokontroler, następnie lutujemy moduł RFM12B, dalej elementy bierne a na końcu koszyczek baterii zasilającej. Następnie przechodzimy na warstwę TOP, gdzie w pierwszej kolejności przylutujemy elementy bierne i diody LED a na końcu switchy SMD. Jeśli chodzi o typ zastosowanych diod LED i kwestie obudowy to zakładam dwie możliwości rozwiązania tego problemu. Pierwsza to zastosowanie diod LED SMD (tak, jak przewidziano na płycie nadajnika) i umieszczenie w wywierconych otworach w obudowie samych główek zwykłych

diod LED o średnicy 3 mm podświetlanych niejako przez wlotowane pod spodem diody SMD, zaś drugie rozwiązanie to wlotowanie w miejsce diod SMD typowych diod LED o średnicy 3 mm z odpowiednio przyciętymi i wygiętymi końcówkami. Na koniec, do tak przygotowanej płytki przylutujemy antenę nadawczą (punkt lutowniczy po stronie BOTTOM) w postaci kawałka przewodu o długości około 17 cm (może być odpowiednio zwinięty). Wygląd zmontowanego nadajnika zarówno od strony warstwy TOP jak i strony warstwy BOTTOM pokazano na **fotografii 1**.

Na koniec kilka słów na temat przygotowania wybranej obudowy. Jak można się domyślić patrząc na obwód drukowany nadajnika, w górnej części jego obudowy (tzn. tej bez otworu na śrubę montażową) należy wywiercić 14 otworów o średnicy 3 mm: 12 otworów dla osi zastosowanych switchy oraz dwa otwory na diody LED. Aby usprawnić proces trasowania otworów przygotowano specjalny szablon, który pokazano na **rysunku 4** (w skali 1:1). Zanim jednak skorzystamy z szablonu, o którym mowa powyżej z wnętrza górnej części obudowy

Listing 7. Funkcja odpowiedzialna za sterowanie przekaźnikami

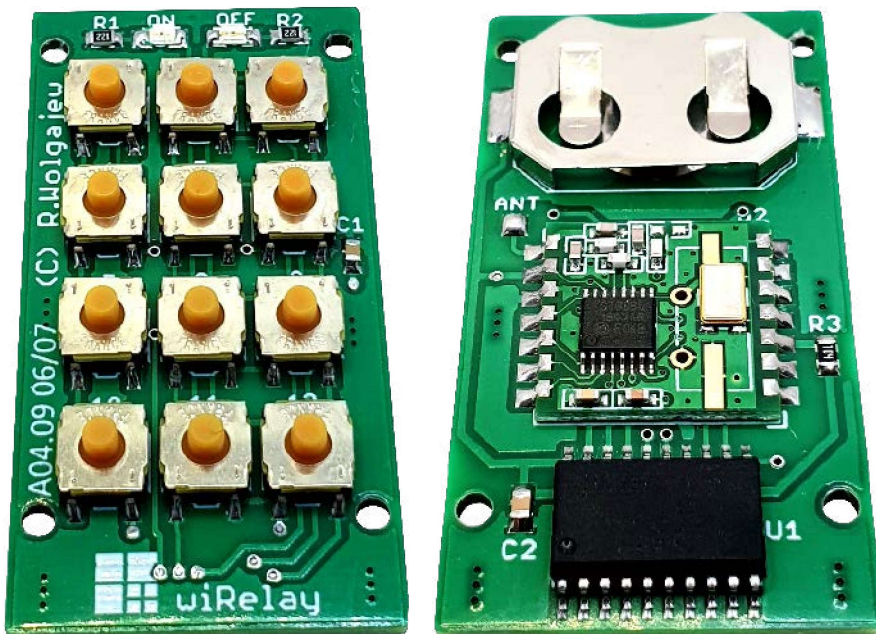
```
void ctrlRelay(uint8_t Command, uint8_t Relay){
  //0...7
  if(Relay < 8){
    if(Command == RELAY_ON)
      PORTB |= (1<<(7-Relay));
    else PORTB &= ~(1<<(7-Relay));
  }
  //8, 9
  }else if(Relay < 10){
    if(Command == RELAY_ON)
      PORTD |= (1<<(Relay-8));
    else PORTD &= ~(1<<(Relay-8));
  }
  //10, 11
  }else{
    if(Command == RELAY_ON)
      PORTA |= (1<<(11-Relay));
    else PORTA &= ~(1<<(11-Relay));
  }
}
```



Rysunek 4. Szablon ułatwiający wykonanie otworów w obudowie nadajnika

(tzn. tej bez otworu montażowego pod śrubę) należy usunąć (przy pomocy ostrego nożyka) wszystkie nadlewki pokazane na **fotografii 2** po to, aby ułatwić umieszczenie szablonu wewnątrz obudowy w celu dokładnego trasowania otworów, oraz aby zwolnić dodatkowe miejsce na obwód drukowany wraz z elementami. Po wykonaniu wspomnianych czynności szablon umieszczamy od środka górnej części obudowy, następnie trasujemy przy jego pomocy miejsca otworów, a na końcu wiercimy je wiertłem do metalu o średnicy 3 mm. Z kolei we wnętrzu dolnej części obudowy (tzn. tej z otworem montażowym pod śrubę) musimy również dokonać pewnych modyfikacji, które mają na celu zwiększenie miejsca na obwód drukowany wraz z elementami. W tym przypadku, podobnie jak poprzednio, usuwamy nadlewki pokazane na **fotografii 3**, które to stanowią podporę zintegrowanych plastikowych tulejek dystansowych uniemożliwiając głębsze osadzenie obudowy drukowanego.

Po takich modyfikacjach obudowa nasza jest gotowa do umieszczenia w niej zmontowanego obudowy drukowanego nadajnika systemu wiRelay. Co prawda po takich



Fotografia 1. Wygląd zmontowanego nadajnika systemu wiRelay od strony TOP i BOTTOM

zabiegach osie zastosowanych mikroprzełączników będą wystawać ponad płaszczyzną obudowy w sposób minimalny, jednak jest to jak najbardziej pożądane, gdyż ograniczy to możliwość przypadkowego wciśnięcia przycisku sterującego. Można jednak nieco ułatwić ich obsługę poprzez nawiercenie stosownych otworów montażowych nieznacznie większym wiertłem, na przykład o średnicy 4 mm.

Przechodzimy do schematu montażowego karty przełączników, który pokazano na **rysunku 5**. Tym razem zaprojektowano dużo większy obwód drukowany, który integruje w sobie kompletny, transformatorowy zasilacz sieciowy i zbudowany jest w zdecydowanej większości z elementów przewlekanych. Montaż obwodu drukowanego karty przełączników rozpoczynamy od przylutowania modułu RFM12B, następnie lutujemy pozostałe elementy półprzewodnikowe, dalej elementy bierne, zaś na samym końcu elementy mechaniczne oraz transformator do druku. Diody LED możemy umieścić na stosownej długości tulejach dystansowych by ich główki wystawały ponad płaszczyznę obudowy przełączników. Do tak przygotowanej płytki przylutowujemy antenę odbiorczą w postaci kawałka przewodu o długości około 17 cm (może być odpowiednio zwinięty). Wygląd zmontowanej karty przełączników od strony warstwy TOP pokazano na **fotografii 4**.

Obsługa urządzenia

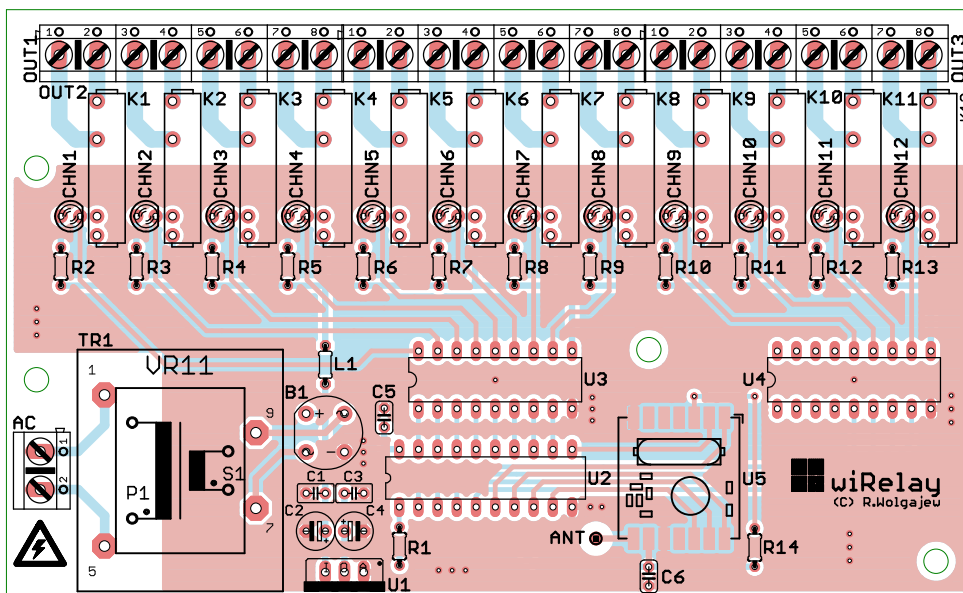
Sposób obsługi systemu wiRelay tak, jak można by tego oczekiwać, jest niezmiernie prosty. Włączeniu pilota systemu wiRelay towarzyszy krótkie (1 s) zaśnięcie obu diod LED, które ma na celu sprawdzenie tych elementów sygnalizacyjnych. Dalej, krótkie naciśnięcie wybranego switcha umieszczonego na nadajniku powoduje wyłączenie powiązanego z nim przełącznika (na karcie przełączników), co powinno zostać potwierdzone poprzez chwilowe zapalenie się diody OFF (zielonej). Długie naciśnięcie wybranego switcha umieszczonego na nadajniku powoduje z kolei włączenie powiązanego z nim przełącznika (na karcie przełączników), co powinno zostać potwierdzone poprzez chwilowe zapalenie się diody ON (czerwonej). Nieodebranie potwierdzenia ze strony karty przełączników (w maksymalnym czasie 100 ms) powoduje chwilowe zapalenie się obu diod LED. W takim wypadku nie mamy pewności, co do stanu sterowanego przełącznika, chyba że możemy wzrokowo ocenić stan karty przełączników korzystając z zamontowanych tam diod LED, których świecenie jest równoznaczne z załączeniem



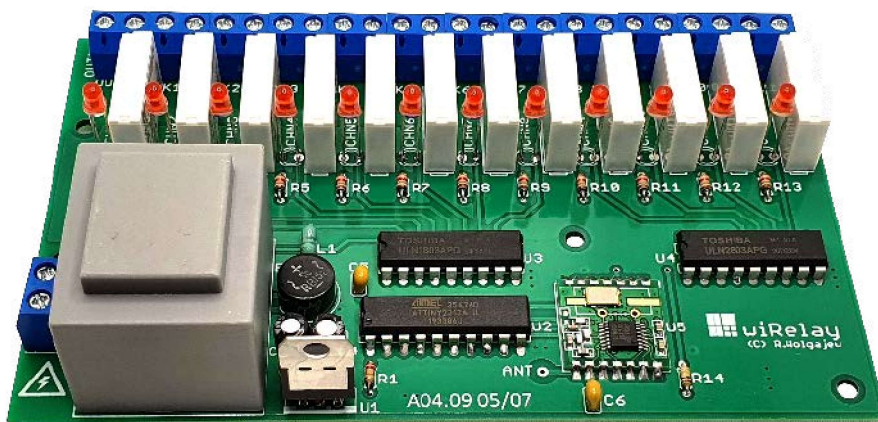
Fotografia 2. Wygląd wnętrza górnej części obudowy z zaznaczeniem nadlewek przeznaczonych do usunięcia



Fotografia 3. Wygląd wnętrza dolnej części obudowy z zaznaczeniem nadlewek przeznaczonych do usunięcia



Rysunek 5. Schemat płytki PCB karty przełączników systemu wiRelay



Fotografia 4. Wygląd zmontowanej karty przełączników systemu wiRelay od strony warstwy TOP

odpowiedniego przełącznika lub zwyczajnie mamy możliwość wzrokowej oceny faktu załączenia do pracy sterowanego urządzenia.

Robert Wołgajew, EP

Odnośniki:

- [1] <https://bit.ly/3C8HLB1>
- [2] <https://bit.ly/3jAKj4H>
- [3] <https://bit.ly/3Ol42zY>