

Moduł Komunikacyjny SER2I2C

Instrukcja obsługi

IO-E369-001

Edycja 1.0

2017.03.06

Przed przystąpieniem do eksploatacji wyrobu należy zapoznać się z treścią niniejszej instrukcji.

Firma POLON-ALFA nie ponosi odpowiedzialności za szkody powstałe w wyniku użytkowania wyrobu.



Wyeksploatowany wyrób nie nadający się do dalszego użytkowania należy przekazać do jednego z punktów zajmujących się zbiórką zużytego sprzętu elektrycznego i elektronicznego.

Uwaga: Zastrzega się prawo do wprowadzania zmian.

Spis treści

1	Wstęp	1
2	Ukompletowanie	2
3	Parametry techniczne	3
4	Budowa i podłączenie urządzeń	4
4.1	Zasilanie urządzenia	4
4.2	Zworki i konfiguracja urządzenia	5
4.3	Diody sygnalizacyjne	5
4.4	Przełączniki	5
4.5	Podłączanie magistrali I ² C	5
4.6	Podłączanie RS-232, USB	5
4.7	Podłączanie RS-485/422	6
4.8	Podłączanie urządzenia do komputera PC poprzez USB	6
4.9	Podłączanie ZR-2	6
5	Tryby pracy	7
5.1	I ² C ↔ RS-232	7
5.2	I ² C ↔ RS-485	7
5.3	I ² C ↔ RS-422	8
5.4	I ² C ↔ USB	8
5.5	USB ↔ RS-232	8
5.6	USB ↔ RS-485	9
5.7	USB ↔ RS-422	9
5.8	RS-232 ↔ RS-485	10
5.9	RS-232 ↔ RS-422	10
6	Protokoły transmisji	11
6.1	Protokół wymiany danych	11
6.1.1	Format pakietu	11
6.1.2	Kody błędów	12
6.1.3	Komendy	12
6.2	Protokół sterowania magistrali	17
6.2.1	Pakiet	18
6.2.2	Komendy	19
6.2.3	Algorytm	19
6.2.4	Stałe wbudowane protokołu	20
7	Przykładowe wykorzystanie urządzenia	21

1 Wstęp

Moduł Komunikacyjny SER2I2C zwany dalej modułem jest urządzeniem realizującym pomost pomiędzy interfejsami komunikacyjnymi takimi jak I²C, USB, RS-232, RS-485/RS-422 (patrz tabela 3.1). Przyrząd wyposażony jest w dwa wyjścia przekaźnikowe (przekaźniki SSR), z czego jedno jest sterowane sygnałem zewnętrznym (patrz rozdział 4.4). Zasilanie może być doprowadzone z przewodu USB, z zewnętrznego napięcia 5 V lub napięciem z zakresu (9 ÷ 30) V (patrz rozdział 4.1). Konfiguracja trybu pracy odbywa się za pomocą zworek, przy czym ich stan odczytywany jest każdorazowo po uruchomieniu modułu.

W urządzeniu zaimplementowane są protokoły umożliwiające sterowanie interfejsem I²C oraz kierowanie danych do poszczególnych elementów (umożliwia to zbudowanie magistrali master-multi slave na bazie RS-485).

Moduł został zaprojektowany jako narzędzie umożliwiające komunikację Zestawu Radiometrycznego ZR-2 z komputerem PC. Niemniej jednak może również działać jako samodzielne urządzenie lub jako element systemu niepowiązanego z ZR-2.

2 Ukompletowanie

W zestawie handlowym znajdują się następujące elementy:

- moduł SER2I2C
- komplet złącz MC
- płyta DVD z:
 - sterownikami USB dla systemów Windows XP i nowszych,
 - instrukcją obsługi,
 - przykładowym programem napisanym w C++ z wykorzystaniem WinAPI demonstrującym sposób korzystania z mechanizmów zaimplementowanych w SER2I2C.

3 Parametry techniczne

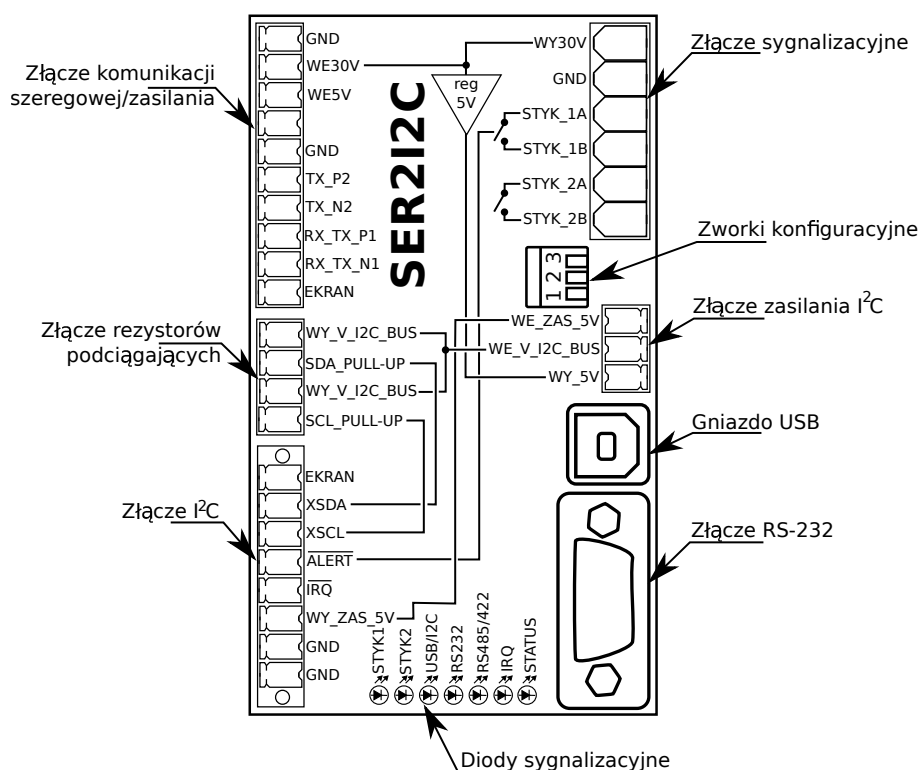
Parametr	Wartość	Jednostka
Parametry mechaniczne		
Wymiary płytki (DŁ × SZ × W)	105 × 60 × 26	mm
Parametry elektryczne		
Zewnętrzne napięcie zasilania	USB 5 (9 ÷ 30)	VDC
Maksymalny pobór prądu	50	mA
Maksymalne obciążenie styków przekaźnika	100	mA
Maksymalne napięcie na styku przekaźnika	350	VDC
Parametry transmisji I²C		
Napięcie magistrali	dobierane przez użytkownika	V
Zewnętrzne rezystory podciągające	dobierane przez użytkownika	
Prędkość transmisji	1000 400 100 50 31	kHz
Parametry transmisji RS-232, RS-485/422		
Prędkość transmisji	zależnie od trybu (Tab. 3.2)	
Bity danych	8	
Bity stopu	1	
Bity parzystości	brak	
Kontrola przepływu	brak	
Parametry transmisji USB		
Klasa	CDC	
Zgodność ze specyfikacją	2.0 wzwyż	
Prędkość transmisji	10	Mb/s
Parametry środowiskowe		
Zakres temperatur pracy	(0 ÷ 40)	°C
Wilgotność	< 80%	%Rh

Tab. 3.1: Parametry urządzenia.

Tryb	Układ zworek ¹	Prędkość transmisji szeregowej	Opis
I ² C – RS-232	1 – 1 – X ²	19200 lub 115200	Roz. 5.1
I ² C – RS-485 ³	1 – 1 – 0	19200 lub 115200	Roz. 5.2
I ² C – RS-422 ³	1 – 1 – 1	19200 lub 115200	Roz. 5.3
I ² C – USB	1 – 1 – X	nie dotyczy	Roz. 5.4
USB – RS-232	0 – 0 – X	19200 lub 115200	Roz. 5.5
USB – RS-485	1 – 0 – 0	19200 lub 115200	Roz. 5.6
USB – RS-422	1 – 0 – 1	19200 lub 115200	Roz. 5.7
RS-232 – RS-485 ⁴	0 – 1 – 0	19200	Roz. 5.8
RS-232 – RS-422	0 – 1 – 1	dowolna	Roz. 5.9

Tab. 3.2: Tryby pracy urządzenia.

4 Budowa i podłączenie urządzeń



Rys. 4.1: Budowa SER2I2C.

Przyrząd SER2I2C istnieje w postaci płytki drukowanej nieumieszczonej w obudowie. Posiada wbudowane złącza USB oraz RS-232, zestaw zworek konfiguracyjnych, diody sygnalizacyjne, dwa przekaźniki oraz gniazda MC do podłączenia przewodów.

4.1 Zasilanie urządzenia

Urządzenie może być zasilane na trzy sposoby:

- poprzez doprowadzenie napięcia zasilającego z zakresu (9 ÷ 30) V do wejścia WE30V, które zasilają wszystkie układy wewnętrzne. Jest bezpośrednio doprowadzone do wyjścia WY30V oraz po obniżeniu napięcia do 5 V do wyjścia WY_5V ułatwiającego ustawianie napięć na magistrali I²C,
- poprzez doprowadzenie zasilania o napięciu 5V do wejścia WE5V. W tym przypadku zasilane są układy wewnętrzne, natomiast wyjścia WY_5V oraz WY30V pozostają niezasilone,
- poprzez port USB, efekt jest identyczny jak w wyżej wymienionym podpunkcie.

Zasilanie magistrali I²C wymaga doprowadzenia dodatkowych napięć (patrz rozdział 4.5).

¹ „1” oznacza zworkę wciśniętą w dół, „0” oznacza zworkę w pozycji górnej, zapis określa pozycje zworek w kolejności 1-2-3.

² „X” oznacza dowolną pozycję zworki.

³ Początkowy interfejs aż do otrzymania pierwszego sygnału.

⁴ W trybie RS-232 – RS-485 występuje echo po stronie RS-232 (dane przekazane na linię Tx portu RS-232 przekazywane są na linię Rx).

4.2 Zworki i konfiguracja urządzenia

Wybór trybu pracy odbywa się poprzez odpowiednie ustawienie zworek (patrz tabela 3.2). Stan zworek odczytywany jest każdorazowo po uruchomieniu urządzenia, więc zmiana trybu wymaga restartu płytki. Dokładny opis trybów pracy znajduje się w rozdziale 5.

4.3 Diody sygnalizacyjne

W urządzeniu znajduje się 7 diod sygnalizujących stan SER2I2C oraz podłączonych układów. Poszczególne diody oznaczają:

- STYK1 (alarm) – dioda świeci na czerwono gdy przełącznik nr 1 jest włączony (wejścia STYK1_A i STYK1_B są zwarte), patrz rozdział 4.4,
- STYK2 (status) – dioda świeci na zielono gdy przełącznik nr 2 jest włączony (wejścia STYK2_A i STYK2_B są zwarte), patrz rozdział 4.4,
- USB/I2C – dioda świeci w sposób ciągły na zielono gdy urządzenie poprawnie przeszło proces enumeracji USB, następnie przygasa w momentach transmisji USB. Na żółto świeci w trakcie aktywności I²C,
- RS232 – dioda sygnalizuje stan transmisji RS-232, kolor zielony oznacza nadawanie, natomiast kolor żółty odbieranie transmisji,
- RS485/422 – dioda sygnalizuje stan transmisji RS-485/422, kolor zielony oznacza nadawanie, natomiast kolor żółty odbieranie transmisji,
- IRQ – dioda świeci na pomarańczowo gdy na wejściu IRQ ustawiony jest stan niski,
- STATUS – dioda sygnalizuje stan urządzenia, sposób świecenia zależy od trybu pracy. W trybach I²C – XXX dioda świeci na zielono i okresowo dwukrotnie przygasa, w pozostałych trybach działa odwrotnie, jest zgaszona i okresowo podwójnie miga.

4.4 Przełączniki

W SER2I2C wbudowane są dwa przełączniki, ich wyjścia połączone są z parami gniazd STYK1_A – STYK1_B oraz STYK2_A – STYK2_B. Ich maksymalna obciążalność została przedstawiona w tabeli 3.1. Startowanie przełącznikami odbywa się w następujący sposób:

- przełącznik nr 1 jest załączany gdy na wejściu ALERT pojawi się logiczne „0”,
- przełącznik nr 2 jest załączany zawsze gdy urządzenie pracuje w prawidłowy sposób w jednym z pierwszych czterech trybów (w trybie translacji z I²C na dowolne medium).

4.5 Podłączanie magistrali I²C

Linie magistrali I²C należy podłączyć do wejść XSDA oraz XSCL. W miarę potrzeby rezystory podciągające wpina się pomiędzy wejścia SDA_PULL-UP – WY_V_I2C_BUS oraz SCL_PULL-UP – WY_V_I2C_BUS. Napięcie podciągające doprowadza się do wejścia WE_V_I2C_BUS (przy odpowiednim zasilaniu wygodnie jest użyć mostka z wyjściem WY_5V).

4.6 Podłączanie RS-232, USB

Na płycie SER2I2C umieszczone jest złącze DB9 żeńskie oraz gniazdo USB typu B.

4.7 Podłączanie RS-485/422

W przypadku transmisji RS-485 parę różnicową należy podłączyć do wejść RX_TX_N1 (linia odwracająca) oraz RX_TX_P1. Jeżeli urządzenie jest skonfigurowane do pracy z RS-422 parę wyjściową z przyrządu stanowią linie TX_N2 i TX_P2 natomiast parę wejściową stanowią RX_TX_N1 oraz RX_TX_P1. Jeżeli zaistnieje taka potrzeba, pomiędzy linie różnicowe należy wpiąć terminatory (rezystory) o rezystancji dopasowanej do impedancji falowej wykorzystanego przewodu (przeważnie ok. 100 Ω).

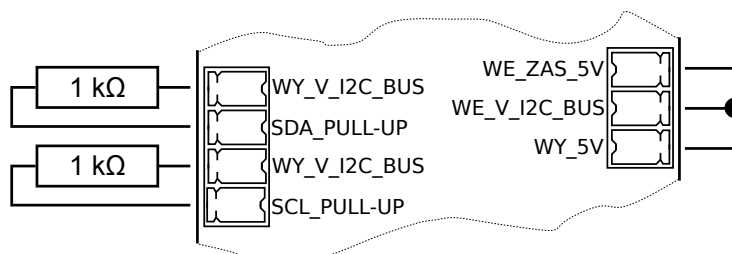
4.8 Podłączanie urządzenia do komputera PC poprzez USB

Po podłączeniu może być konieczna instalacja dostarczonych wraz z urządzeniem sterowników. Urządzenie wykorzystuje klasę CDC specyfikacji USB, tak więc generyczne sterowniki są obecne w większości systemów operacyjnych. Dla systemów Windows niezbędnym jest jednak przekazanie informacji o tym, że należy użyć wbudowanego w system sterownika klasy i tylko temu służy dostarczony z przyrządem sterownik.

Urządzenie jest widoczne w systemie jako dodatkowy port szeregowy.

4.9 Podłączanie ZR-2

Aby urządzenie prawidłowo współpracowało z ZR-2, musi być odpowiednio skonfigurowane. Należy wpiąć rezystory podciągające o wartości 1 k Ω oraz wykonać mostki w celu zasilania ZR-2 (sposób podłączenia rezystorów i mostków pokazano na rysunku 4.2). Zasilanie musi być doprowadzone do wejścia WE30V (patrz rozdział 4.1).



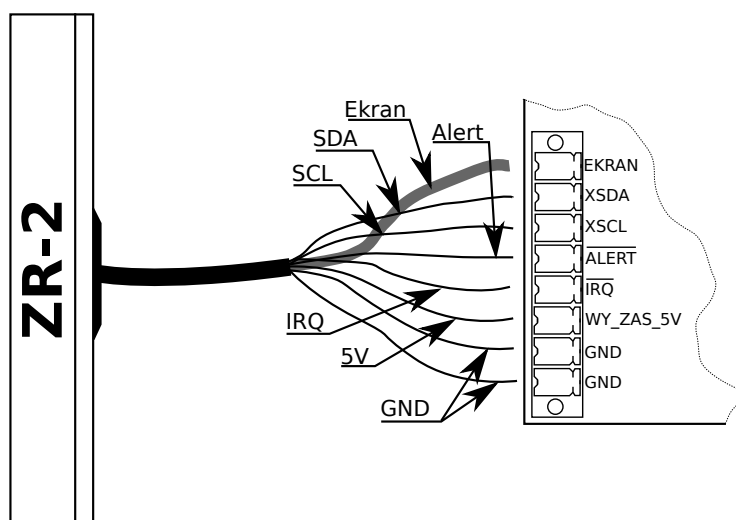
Rys. 4.2: Podłączenie rezystorów oraz mostków do pracy z ZR-2.

ZR-2 należy podłączyć do płytki SER2I2C w sposób jaki pokazano na 4.3.

SER2I2C z podłączonym ZR-2 można podłączyć bezpośrednio do komputera (przez USB lub RS-232) lub transmitować na dłuższe odległości za pomocą RS-485/422 (można wykorzystać drugą płytkę SER2I2C do tłumaczenia RS-485/422 na USB przy komputerze).



Połączenie ZR-2 ↔ SER2I2C(I²C – RS-485) ↔ SER2I2C(RS-485 – RS-232) ↔ PC uniemożliwi funkcjonowanie oprogramowania ZR-2 z powodu echa na konwerterze. Aby uzyskać dostęp do magistrali poprzez wejście RS-232 należy zastosować inny konwerter, np. MOXA TCC-100I.



Rys. 4.3: Podłączenie ZR-2 do SER2I2C.

5 Tryby pracy

Urządzenie może pracować w 9 trybach realizujących różne połączenia, sposób działania determinowany jest poprzez odpowiednie ustawienie trzech zworek (tabela 3.2). Stan zworek odczytywany jest każdorazowo po uruchomieniu urządzenia.

Niektóre ustawienia zworek nie definiują jednoznacznie trybu (np. układ 1 – 1 – 1 pasuje do trybów I²C – RS-232, I²C – RS-422 oraz I²C – USB), w takim przypadku urządzenie nasłuchuje na wszystkich możliwych interfejsach i za aktywny przyjmuje ten, na którym odczytał pierwszą transmisję. Domyślnym interfejsem (zanim wykryje aktywność na jakimkolwiek z nich) jest RS-485/RS-422 i to na niego początkowo będą przesyłane dane z I²C.

W transmisji na portach szeregowych parametry (nie licząc prędkości) są stałe (patrz tabela 3.1), nie jest realizowane żadne sterowanie przepływem.

5.1 I²C ↔ RS-232

Tryb ten realizuje translację pomiędzy interfejsem I²C a RS-232. Aby urządzenie pracowało w ten sposób należy ustawić zworki w trybie 1 – 1 – X, z powodu niejednoznaczności takiego ustawienia zworek pierwsza transmisja (nie licząc I²C) musi nadejść z portu szeregowego RS-232.

W tym trybie aktywne są protokoły opisane w rozdziale 6. Transmisja na porcie szeregowym odbywa się z prędkością 19200 *bps* lub 115200 *bps*, ustawiana jest za pomocą odpowiednich komend protokołu wymiany danych (patrz rozdział 6.1.3).

Prędkość transmisji I²C (31 *kHz*, 50 *kHz*, 100 *kHz*, 400 *kHz* lub 1 *MHz*) ustawiana jest za pomocą odpowiednich komend protokołu wymiany danych (patrz rozdział 6.1.3).

Jeżeli płytką SER2I2C działa prawidłowo dioda sygnalizacyjna świeci na zielono i okresowo dwukrotnie przygasa. W tym trybie styki przekaźnika 2 (STYK2_A i STYK2_B) są zwarte.

5.2 I²C ↔ RS-485

Tryb ten realizuje translację pomiędzy interfejsem I²C a RS-485. Aby urządzenie pracowało w ten sposób należy ustawić zworki w trybie 1 – 1 – 0, z powodu niejednoznaczności takiego ustawienia zworek pierwsza transmisja (nie licząc I²C) musi nadejść z interfejsu szeregowego RS-485.

W tym trybie aktywne są protokoły opisane w rozdziale 6. Transmisja na porcie szeregowym odbywa się z prędkością 19200 *bps* lub 115200 *bps*, ustawiana jest za pomocą odpowiednich komend protokołu wymiany danych (patrz rozdział 6.1.3).

Przy korzystaniu z transmisji RS-485 należy pamiętać, że urządzenie SER2I2C nie zapewnia wykrywania ani eliminowania kolizji.

Prędkość transmisji I²C (31 *kHz*, 50 *kHz*, 100 *kHz*, 400 *kHz* lub 1 *MHz*) ustawiana jest za pomocą odpowiednich komend protokołu wymiany danych (patrz rozdział 6.1.3).

Jeżeli płytko SER2I2C działa prawidłowo dioda sygnalizacyjna świeci na zielono i okresowo dwukrotnie przygasa. W tym trybie styki przekaźnika 2 (STYK2_A i STYK2_B) są zwarte.

Praca w tym trybie umożliwia zbudowanie magistrali łączącej wiele płytek SER2I2C pod kontrolą oprogramowania komputera PC z wykorzystaniem protokołu sterowania magistrali (patrz rozdział 6.2).

5.3 I²C ↔ RS-422

Tryb ten realizuje translację pomiędzy interfejsem I²C a RS-422. Aby urządzenie pracowało w ten sposób należy ustawić zworki w trybie 1 – 1 – 1, z powodu niejednoznaczności takiego ustawienia zworek pierwsza transmisja (nie licząc I²C) musi nadejść z interfejsu szeregowego RS-422.

W tym trybie aktywne są protokoły opisane w rozdziale 6. Transmisja na porcie szeregowym odbywa się z prędkością 19200 *bps* lub 115200 *bps*, ustawiana jest za pomocą odpowiednich komend protokołu wymiany danych (patrz rozdział 6.1.3).

Prędkość transmisji I²C (31 *kHz*, 50 *kHz*, 100 *kHz*, 400 *kHz* lub 1 *MHz*) ustawiana jest za pomocą odpowiednich komend protokołu wymiany danych (patrz rozdział 6.1.3).

Jeżeli płytko SER2I2C działa prawidłowo dioda sygnalizacyjna świeci na zielono i okresowo dwukrotnie przygasa. W tym trybie styki przekaźnika 2 (STYK2_A i STYK2_B) są zwarte.

5.4 I²C ↔ USB

Tryb ten realizuje translację pomiędzy interfejsem I²C a USB. Aby urządzenie pracowało w ten sposób należy ustawić zworki w trybie 1 – 1 – X, z powodu niejednoznaczności takiego ustawienia zworek pierwsza transmisja (nie licząc I²C) musi nadejść z portu USB.

W tym trybie aktywne są protokoły opisane w rozdziale 6. Po podłączeniu płytki poprzez USB do komputera PC, urządzenie powinno być widoczne jako dodatkowy port szeregowy, parametry z jakimi port zostanie otwarty nie wpływają w żaden sposób na transmisję. Do prawidłowego działania USB mogą być wymagane sterowniki (patrz rozdział 4.8). Po podłączeniu USB (jeżeli proces enumeracji przebiegł poprawnie) dioda USB/I²C powinna świecić się na zielono i przygasać w czasie transmisji po USB. Jeżeli po podłączeniu płytki przez USB do komputera dioda sygnalizacyjna się nie zaświeciła należy upewnić się czy system widzi urządzenie w prawidłowy sposób (jako port szeregowy), pomocne może być odłączenie i ponowne podłączenie przewodu.

Prędkość transmisji I²C (31 *kHz*, 50 *kHz*, 100 *kHz*, 400 *kHz* lub 1 *MHz*) ustawiana jest za pomocą odpowiednich komend protokołu wymiany danych (patrz rozdział 6.1.3).

Jeżeli płytko SER2I2C działa prawidłowo dioda sygnalizacyjna świeci na zielono i okresowo dwukrotnie przygasa. W tym trybie styki przekaźnika 2 (STYK2_A i STYK2_B) są zwarte.

5.5 USB ↔ RS-232

Tryb ten realizuje translację pomiędzy interfejsem USB a RS-232. Aby urządzenie pracowało w ten sposób należy ustawić zworki w trybie 0 – 0 – X.

W tym trybie nie są aktywne żadne dodatkowe protokoły, dane są przesyłane bez ich analizowania. Po podłączeniu płytki poprzez USB do komputera PC, urządzenie powinno być widoczne jako dodatkowy port szeregowy. Do prawidłowego działania USB mogą być wymagane sterowniki (patrz rozdział 4.8). Po podłączeniu USB (jeżeli proces enumeracji przebiegł poprawnie) dioda USB/I2C powinna świecić się na zielono i przygasać w czasie transmisji po USB. Jeżeli po podłączeniu płytki przez USB do komputera dioda sygnalizacyjna się nie zaświeciła należy upewnić się czy system widzi urządzenie w prawidłowy sposób (jako port szeregowy), pomocne może być odłączenie i ponowne podłączenie przewodu.

Transmisja na porcie szeregowym odbywa się z prędkością 19200 *bps* lub 115200 *bps*. Prędkość tą dobiera się poprzez otwarcie portu USB z odpowiednią konfiguracją (wybranie nieobsługiwanej prędkości spowoduje ustawienie na interfejsie RS-232 baud rate 19200 *bps*).

Jeżeli płytka SER2I2C działa prawidłowo dioda sygnalizacyjna jest zgaszona i okresowo podwójnie miga na zielono. W tym trybie styki przekaźnika 2 (STYK2_A i STYK2_B) są rozwarne.

5.6 USB ↔ RS-485

Tryb ten realizuje translację pomiędzy interfejsem USB a RS-485. Aby urządzenie pracowało w ten sposób należy ustawić zworki w trybie 1 – 0 – 0.

W tym trybie nie są aktywne żadne dodatkowe protokoły, dane są przesyłane bez ich analizowania. Po podłączeniu płytki poprzez USB do komputera PC, urządzenie powinno być widoczne jako dodatkowy port szeregowy. Do prawidłowego działania USB mogą być wymagane sterowniki (patrz rozdział 4.8). Po podłączeniu USB (jeżeli proces enumeracji przebiegł poprawnie) dioda USB/I2C powinna świecić się na zielono i przygasać w czasie transmisji po USB. Jeżeli po podłączeniu płytki przez USB do komputera dioda sygnalizacyjna się nie zaświeciła należy upewnić się czy system widzi urządzenie w prawidłowy sposób (jako port szeregowy), pomocne może być odłączenie i ponowne podłączenie przewodu.

Transmisja na porcie szeregowym odbywa się z prędkością 19200 *bps* lub 115200 *bps*. Prędkość tą dobiera się poprzez otwarcie portu USB z odpowiednią konfiguracją (wybranie nieobsługiwanej prędkości spowoduje ustawienie na interfejsie RS-232 baud rate 19200 *bps*).

Przy korzystaniu z transmisji RS-485 należy pamiętać, że urządzenie SER2I2C nie zapewnia wykrywania ani eliminowania kolizji.

Jeżeli płytka SER2I2C działa prawidłowo dioda sygnalizacyjna jest zgaszona i okresowo podwójnie miga na zielono. W tym trybie styki przekaźnika 2 (STYK2_A i STYK2_B) są rozwarne.

5.7 USB ↔ RS-422

Tryb ten realizuje translację pomiędzy interfejsem USB a RS-422. Aby urządzenie pracowało w ten sposób należy ustawić zworki w trybie 1 – 0 – 1.

W tym trybie nie są aktywne żadne dodatkowe protokoły, dane są przesyłane bez ich analizowania. Po podłączeniu płytki poprzez USB do komputera PC, urządzenie powinno być widoczne jako dodatkowy port szeregowy. Do prawidłowego działania USB mogą być wymagane sterowniki (patrz rozdział 4.8). Po podłączeniu USB (jeżeli proces enumeracji przebiegł poprawnie) dioda USB/I2C powinna świecić się na zielono i przygasać w czasie transmisji po USB. Jeżeli po podłączeniu płytki przez USB do komputera dioda sygnalizacyjna się nie zaświeciła należy upewnić się czy system widzi urządzenie w prawidłowy sposób (jako port szeregowy), pomocne może być odłączenie i ponowne podłączenie przewodu.

Transmisja na porcie szeregowym odbywa się z prędkością 19200 *bps* lub 115200 *bps*. Prędkość tą dobiera się poprzez otwarcie portu USB z odpowiednią konfiguracją (wybranie nieobsługiwanej prędkości spowoduje ustawienie na interfejsie RS-232 baud rate 19200 *bps*).

Jeżeli płytką SER2I2C działa prawidłowo dioda sygnalizacyjna jest zgaszona i okresowo podwójnie miga na zielono. W tym trybie styki przekaźnika 2 (STYK2_A i STYK2_B) są rozwarne.

5.8 RS-232 ↔ RS-485

Tryb ten realizuje translację pomiędzy interfejsem RS-232 a RS-485. Aby urządzenie pracowało w ten sposób należy ustawić zworki w trybie 0 – 1 – 0.

W tym trybie nie są aktywne żadne dodatkowe protokoły, dane są przesyłane bez ich analizowania. Transmisja na porcie szeregowym odbywa się z prędkością 19200 *bps*, prędkość taką należy ustawić na urządzeniach podłączonych do płytki. Ograniczenie prędkości wynika z czasu aktywacji nadajnika RS-485 dopasowanego do baud rate 19200 *bps*. Ustawienie innej prędkości może doprowadzić do powstawania błędów transmisji na magistrali RS-485.

Przy korzystaniu z transmisji RS-485 należy pamiętać, że urządzenie SER2I2C nie zapewnia wykrywania ani eliminowania kolizji.

W tym trybie powstaje sprzętowe echo widoczne od strony RS-232.

Jeżeli płytką SER2I2C działa prawidłowo dioda sygnalizacyjna jest zgaszona i okresowo podwójnie miga na zielono. W tym trybie styki przekaźnika 2 (STYK2_A i STYK2_B) są rozwarne.

5.9 RS-232 ↔ RS-422

Tryb ten realizuje translację pomiędzy interfejsem RS-232 a RS-422. Aby urządzenie pracowało w ten sposób należy ustawić zworki w trybie 0 – 1 – 1.

W tym trybie nie są aktywne żadne dodatkowe protokoły, dane są przesyłane bez ich analizowania. Transmisja na porcie szeregowym odbywa się z dowolną prędkością (dane są przesyłane z identycznymi parametrami jak zostały odebrane)

Jeżeli płytką SER2I2C działa prawidłowo dioda sygnalizacyjna jest zgaszona i okresowo podwójnie miga na zielono. W tym trybie styki przekaźnika 2 (STYK2_A i STYK2_B) są rozwarne.

6 Protokoły transmisji

W urządzeniu (tylko w trybach wykorzystujących I²C) zaimplementowano dwa protokoły spełniające różne zadania. Każdy z protokołów działa identycznie niezależnie od wykorzystywanego interfejsu.



Opisane protokoły transmisji są aktywne tylko w trybach wykorzystujących interfejs I²C.

6.1 Protokół wymiany danych

Protokół wymiany danych służy do komunikacji komputera PC z płytką SER2I2C oraz do zlecenia transakcji I²C.

6.1.1 Format pakietu

Żądanie sprowadza się do wysłania następującego pakietu:

```
[0x00]
[0xFF]
[bajt komendy]
[rozmiar ładunku w bajtach]
...ładunek
[zanegowany bajt komendy]
```

Po wysłaniu żądania przyrząd przystępuje do wykonania polecenia a następnie, zależnie od rezultatu udziela odpowiedzi:

```
[0x00]
[0xFF]
[0 ooo oooo] - kod odpowiedzi
[rozmiar ładunku w bajtach]
...ładunek
[zanegowany bajt kodu odpowiedzi]
```

Lub, jeżeli nie przewiduje się odpowiedzi, potwierdza wykonanie odsyłając:

```
[0x00]
[0xFF]
[bajt komendy]
[ 0 ]
[zanegowany bajt komendy],
```

gdzie „bajt komendy” jest komendą którą się potwierdza.

W razie wystąpienia problemu informuje o błędzie:

```
[0x00]
[0xFF]
[1 eee eeee] – kod błędu
[ 0 ]
[zanegowany bajt kodu błędu]
```

6.1.2 Kody błędów

Nazwa	Wartość	Opis
evtERROR_SYNTAX	0x80	Wysłany, gdy odebrane przez urządzenie SER2I2C polecenie jest odebrane poprawnie, rozpoznane ale zawiera nieprawidłowe parametry, np. <code>cmTRANSACTION</code> z całkowitym rozmiarem bufora do zapisu przekraczającym dopuszczalny lub zerowym przeterminowaniem.
evtERROR_COMMAND	0x82	Wysyłany, gdy nie rozpoznano przesłanej komendy.
evtE_TIMEOUT	0x83	Wysłany, gdy operacja na magistrali I ² C nie mogła zakończyć się w żądanym czasie, np. z powodu zbyt długiego rozciągnięcia zegara przetrzymaniem linii SCL.
evtE_NOACKNOWLEDGE_STAGE_1	0x84	Wysłane, gdy zaadresowane w transakcji I ² C urządzenie nie potwierdziło zaadresowania w pierwszej części transakcji. Informacja: Zazwyczaj jeżeli urządzenie jest zajęte zamiast nie potwierdzać zaadresowania zatrzyma ono linię SCL w stanie niskim co ewentualnie może doprowadzić do <code>evtE_TIMEOUT</code> .
evtE_NOACKNOWLEDGE_STAGE_2	0x85	Jak wyżej, ale dotyczy drugiej części transakcji.

Tab. 6.1: Kody błędów protokołu wymiany danych.

6.1.3 Komendy

cmdIDENT 0x00

Komenda odczytuje informacje o wersji urządzenia.

Ładunek:

brak ładunku

Odpowiedź:

[wersja_protokołu]
[kod_urządzenia]

Dla urządzenia opisanego w tym dokumencie:

- wersja_protokołu – 2
- kod_urządzenia – 1

cmdTRANSACTION 0x01

Podstawowa komenda, zleca wykonanie transakcji na magistrali I²C.

Ładunek:

[w1]
[r1]
[w2]
[r2]
[timeout, 7...0]
[timeout, 15..8]
... do 250 bajtów.

Gdzie:

- $w1$ – rozmiar pierwszej operacji zapisu, w bajtach, zero, jeżeli pierwszą operację należy pominąć
- $r1$ – rozmiar odczytu następującego po pierwszej operacji zapisu, zero gdy operację pominąć
- $w2$ – rozmiar drugiej operacji zapisu, w bajtach, zero gdy operację pominąć. Między pierwszą a drugą operacją urządzenie wystawi warunek *repeated start*. Przy czym $w1+w2$ nie może być większe niż 250
- $r2$ – rozmiar drugiej operacji odczytu, zero gdy operację pominąć. Przy czym $r1+r2$ nie może być większe niż 255
- *timeout* – przeterminowanie operacji w jednostkach $16 \mu s$. Wartość nie może być zerem.
- następnie łącznie do 250 bajtów jakie zapisać w kolejnych operacjach.

Odpowiedź:

Jeżeli transakcję wykonano poprawnie ładunek pakietu zawiera:

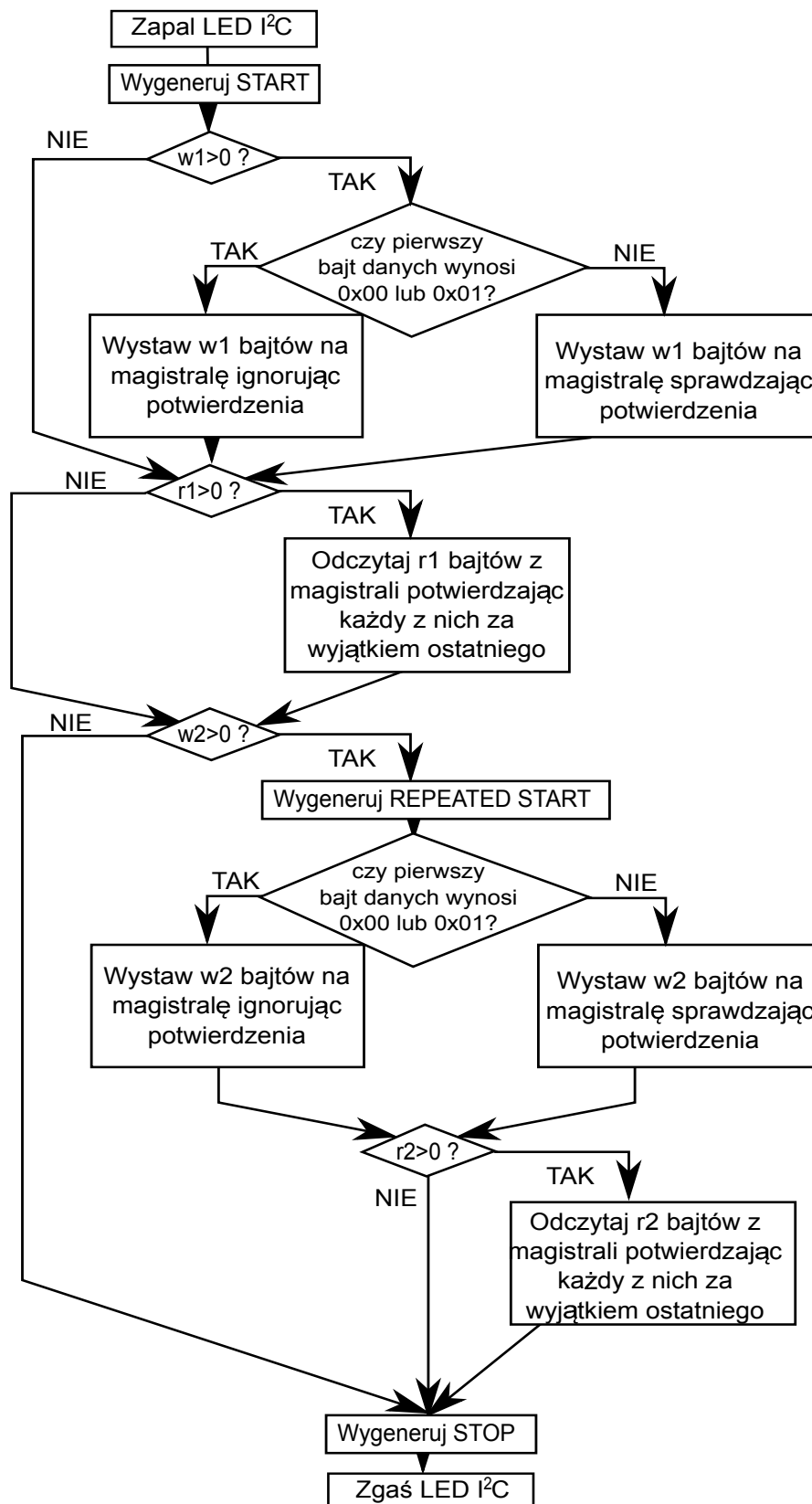
.... do 255 bajtów

Odpowiedź składa się z maksymalnie 255 bajtów odczytanych w kolejnych operacjach, przy czym oczywiście cały pakiet wygląda:

```
[0x00]
[0xFF]
[0x01]
[ ? ] - rozmiar
.... dane
[~0x01]
```

Jeżeli wykryto błędy zamiast odpowiedzi generowane są wymienione w rozdziale 6.1.2 kody błędów.

Bardziej szczegółowo sposób działania opisuje algorytm przedstawiony na rysunku 6.1.



Rys. 6.1: Algorytm polecenia cmdTRANSACTION. Rysunek nie uwzględnia wykrycia błędów transmisji i przeterminowań.

Przykład:

Zapisanie 4-rech bajtów z urządzenia pod adresem 7:

```
[0x00]
[0xFF]
[0x01]
[0x0B] – rozmiar ładunku
[ 5 ] – w1 = 5, zapis bajtu adresowego + 4 bajty danych
[ 0 ]
[ 0 ]
[ 0 ]
[0xFF] – timeout
[0x00]
[0x7<<1 + 0x00] – bajt adresowy, najmłodszy bit to kod operacji
[0xAA] – dane
[0xBB]
[0xCC]
[0xDD]
[~0x01]
```

Odpowiedź:

```
[0x00]
[0xFF]
[0x01]
[0x00]
[~0x01]
```

Przykład:

Odczytanie 4-rech bajtów z urządzenia pod adresem 7:

```
[0x00]
[0xFF]
[0x01]
[0x07] – rozmiar ładunku
[ 1 ] – w1 = 1, zapis bajtu adresowego
[ 4 ] – r1 = 4 – odczyt danych
[ 0 ]
[ 0 ]
[0xFF] – timeout
[0x00]
[0x7<<1 + 0x01] – bajt adresowy, najmłodszy bit to kod operacji
[~0x01]
```

Odpowiedź:

```
[0x00]
[0xFF]
[0x01]
[0x04]
[ . . . ] – x4 – odczytane dane.
[~0x01]
```

Przykład:

Zapis dwu bajtów do urządzenia pod adresem 7 po czym odczyt trzech bajtów z warunkiem REPEATED_START.

[0x00]
[0xFF]
[0x01]
[0x0A] – rozmiar ładunku
[3] – w1=3, zapis bajtu adresowego + dane
[0]
[1] – w2=1, zapis bajtu adresowego + dane
[3] – r2=3, odczyt
[0xFF] – timeout
[0x00]
[0x7<<1 + 0x00] – bajt adresowy, najmłodszy bit to kod operacji
[0xAA] – dane do zapisania
[0xBB]
[0x7<<1 + 0x01] – bajt adresowy z kodem odczytu
[~0x01]

Odpowiedź:

[0x00]
[0xFF]
[0x01]
[0x03]
[. . .] – x3 – odczytane dane.
[~0x01]

cmdBAUD_1MHz 0x02

Komenda ustala zegar magistrali I²C na 1 MHz.

Ładunek:

brak ładunku

Odpowiedź:

Tylko potwierdzenie.

cmdBAUD_400kHz 0x03

Komenda ustala zegar magistrali I²C na 400 kHz.

Ładunek:

brak ładunku

Odpowiedź:

Tylko potwierdzenie.

cmdBAUD_100kHz 0x04

Komenda ustala zegar magistrali I²C na 100 kHz.

Ładunek:

brak ładunku

Odpowiedź:

Tylko potwierdzenie.

cmdBAUD_50kHz 0x05

Komenda ustala zegar magistrali I²C na 50 kHz.

Ładunek:

brak ładunku

Odpowiedź:

Tylko potwierdzenie.

cmdBAUD_31kHz 0x06

Komenda ustala zegar magistrali I²C na 31 *kHz*.

Ładunek:

brak ładunku

Odpowiedź:

Tylko potwierdzenie.

cmdSerialBaudrate19200 0x08

Komenda przełącza transmisję szeregową na prędkość 19200 *bps*, 8 bitów danych, 1 stopu, bez parzystości.

Ładunek:

brak ładunku

Odpowiedź:

Brak odpowiedzi.

cmdSerialBaudrate115200 0x09

Komenda przełącza transmisję szeregową na prędkość 115200 *bps*, 8 bitów danych, 1 stopu, bez parzystości.

Ładunek:

brak ładunku

Odpowiedź:

Brak odpowiedzi.

cmdGetBAUD 0x0A

Komenda odczytuje bieżące nastawy zegara magistrali I²C.

Ładunek:

brak ładunku

Odpowiedź:

[b0]

[b1]

Gdzie:

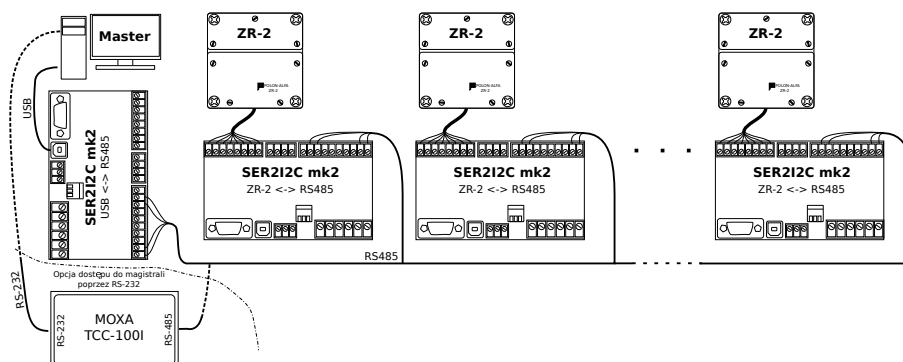
- b0 – młodszy bajt
- b1 – starszy bajt, razem prędkość zegara magistrali I²C w kHz.

6.2 Protokół sterowania magistrali

Protokół został zaimplementowany jako warstwa nadrzędna dla protokołu opisanego w punkcie 6.1. Protokół sterowania magistrali umożliwia zbudowania sieci master-multi slave jak na

rysunku 6.2. Zapewnia mechanizm wykrywania urządzeń oraz adresowania informacji do poszczególnych odbiorników. Protokół jest opcjonalny i jego pominięcie w żaden sposób nie wpłynie na działanie sieci (każda informacja będzie odbierana przez wszystkie odbiorniki). Urządzenia SER2I2C identyfikowane są za pomocą numerów fabrycznych indywidualnych dla każdego przyrządu.

Płytki SER2I2C implementują tylko część dotyczącą urządzeń slave, jako master musi pracować odpowiednie urządzenie zewnętrzne (np. aplikacja na komputerze PC).



Rys. 6.2: Magistrala urządzeń z wykorzystaniem SER2I2C.



Z powodu ograniczeń technicznych urządzenia SER2I2C, nie może ono działać w wyżej pokazanym schemacie jako konwerter RS485-RS232. Aby uzyskać dostęp do magistrali poprzez wejście RS-232 należy zastosować urządzenie MOXA TCC-100I.

Adresowanie danych za pomocą protokołu starowania magistrali polega na poprzedzeniu pakietu protokołu wymiany danych odpowiednią ramką (patrz rozdział 6.2.1). Urządzenia odbiorcze odczytują z niej adres docelowy, jeżeli nie jest on zgodny z adresem odbiornika lub z adresem ogłoszeniowym kolejny odebrany pakiet protokołu wymiany danych zostanie zignorowany.

Budowanie mapy urządzeń następuje według algorytmu opisanego w rozdziale 6.2.3.

6.2.1 Pakiet

Pakiet protokołu składa się z samodzielnej ramki pokazanej na Rys. 2.1 (sygnały sterujące magistrali), lub ramki sklejonej z pakietem protokołu wymiany danych (przesyłanie danych do konkretnego urządzenia).

start	cmd	adres	tail
2B	1B	2B	1B

Rys. 6.3: Ramka protokołu sterowania magistrali.

- **start** - dwa bajty startowe, stałe o wartości 0x0FF0,
- **cmd** - komenda sterująca o długości 1B, dokładny spis komend znajduje się w rozdziale 6.2.2,
- **adres** - adres docelowy urządzenia (młodszy bajt pierwszy), gdy pakiet nie jest kierowany do konkretnego odbiorcy należy wysłać wartość 0xFFFF,
- **tail** - zakończenie ramki będące negacją bitową komendy.

6.2.2 Komendy

Discovery 0x90 : rozpoznanie sieci

Sygnał wysyłany przez urządzenie master na adres 0xFFFF w celu wywołania urządzeń w sieci.

Response 0x91 : odpowiedź na discovery

Sygnał wysyłany przez urządzenie slave ze swoim adresem jako odpowiedź na zapytanie discovery. Wysyłanie odpowiedzi jest uwarunkowane założeniami opisanymi w rozdziale 6.2.3.

Accept 0x92 : zaakceptowanie urządzenia

Sygnał wysyłany przez urządzenie master na adres konkretnego odbiorcy po prawidłowym jego wykryciu. Slave po otrzymaniu tej komendy przestaje odsyłać sygnały response.

Reset 0x93 : reset sieci

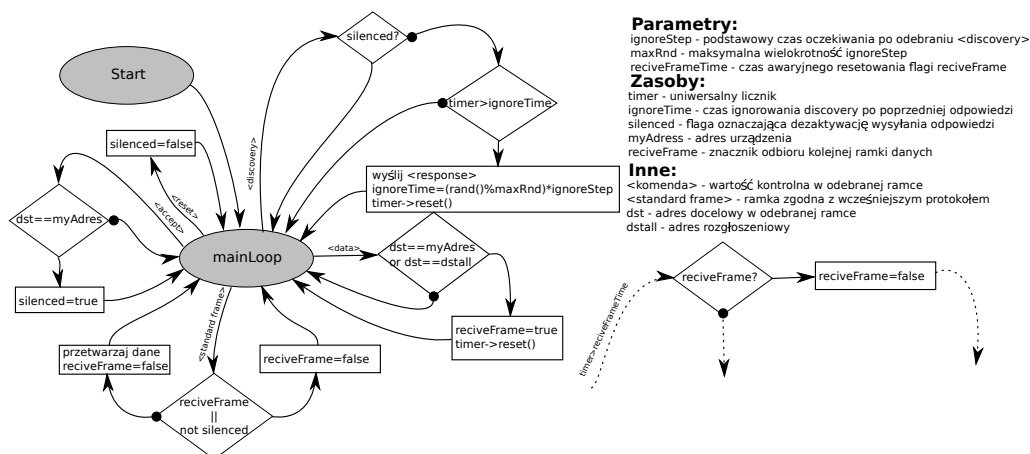
Sygnał wysyłany przez urządzenie master na adres 0xFFFF w celu zresetowania stanu odbiorników.

Data 0x94 : przesył danych

Sygnał wysyłany przez urządzenie master na adres konkretnego urządzenia lub na adres rozgłoszeniowy (0xFFFF), służy do kierowania pakietu bazowego do wskazanego odbiornika. Ramka powinna być wklejona bezpośrednio przed pakiet protokołu przesyłu danych. Odbiornik odczytuje ramkę z komendą data, jeżeli przekazany adres nie jest identyczny z adresem urządzenia lub z adresem rozgłoszeniowym, kolejny pakiet bazowy zostanie zignorowany.

6.2.3 Algorytm

Protokół sterowania magistrali działa według algorytmu pokazanego na rysunku 6.4.



Rys. 6.4: Algorytm działania protokołu sterowania magistrali po stronie urządzeń slave.

Przesył danych

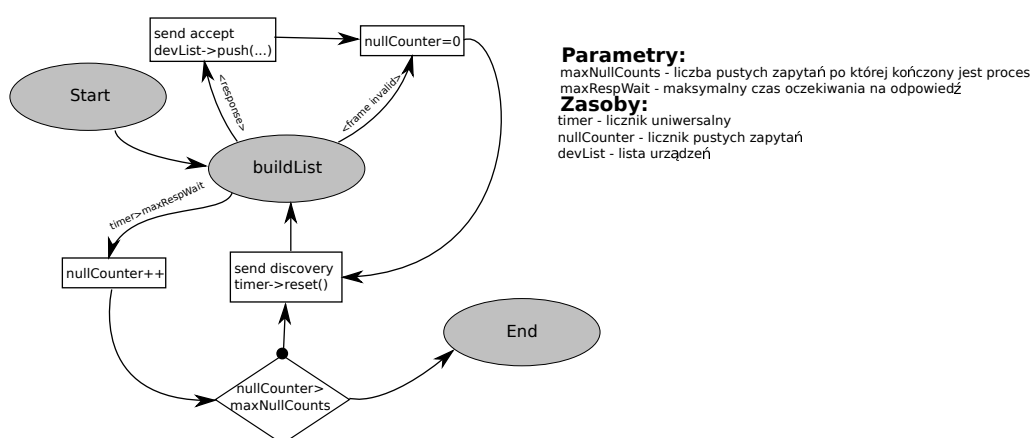
Pakiet z danymi powinien składać się z ramki protokołu sterowania magistrali (patrz rozdział 6.2.1) z rozkazem „data” oraz doklejonego bezpośrednio za nią pakietu zgodnego z protokołem wymiany danych (patrz rozdział 6.1). Odbiorniki odczytują z pierwszej części sygnału adres docelowy, jeżeli adres jest zgodny z adresem urządzenia lub adresem rozgłoszeniowym (0xFFFF) ustawiana jest flaga `recvFrame`. Druga część pakietu jest analizowana na podstawie standardowych

mechanizmów, różnicą jest tylko sprawdzanie flag `recvFrame` oraz `silenced` przed wykonaniem rozkazu zawartego w pakiecie. Jeżeli urządzenie nie jest zaadresowane przez PC (flaga `silenced` jest wyzerowana) odbierany będzie każdy pakiet, jeżeli urządzenie jest zaadresowane przetwarzane są tylko pakiety adresowane do danego odbiornika (sprawdzana jest flaga `recvFrame`). Flaga `recvFrame` jest resetowana przed wykonaniem rozkazu lub po upływie czasu od jej ustawienia (patrz rozdział 6.2.4).

Protokół jest opcjonalny i pominięcie go nie spowoduje żadnych błędów. Odbiorniki rozpoznają typ pakietu na podstawie bajtów startowych, jeżeli urządzenie odbierze pakiet bez dodanego wyżej opisanego wstępu, przejdzie bezpośrednio do wykonywania rozkazów zawartych w wiadomości.

Budowanie mapy urządzeń

Za zbudowanie listy urządzeń SER2I2C dostępnych na magistrali odpowiedzialne jest oprogramowanie komputera PC realizujące algorytm przedstawiony na rysunku 6.5.



Rys. 6.5: Algorytm wykrywania urządzeń po stronie PC master.

Master cyklicznie wysyła sygnał `discovery` i nasłuchuje odpowiedzi, jeżeli otrzyma prawidłowy pakiet `response` dodaje urządzenie o odczytanym adresie do listy, jeżeli nie otrzyma żadnej odpowiedzi lub wykryje uszkodzony pakiet (kolizja transmisji) powtarza zapytanie. Algorytm jest zatrzymywany po określonej liczbie (patrz rozdział 6.2.4) pustych zapytań pod rząd.

Urządzenia SER2I2C odpowiadają na sygnał `discovery` pod warunkiem, że flaga `silenced` nie jest ustawiona oraz że od ostatniej odpowiedzi minął odpowiedni czas. Czas ten jest ustalany każdorazowo po wysłaniu sygnału `response` na podstawie równania $ignoreTime = (rand() \% maxRnd) \cdot ignoreStep$, dokładne wartości podano w rozdziale 6.2.4. Flaga `silenced` ustawiana jest po odebraniu sygnału `accepted` z odpowiednim adresem, natomiast zerowana jest poprzez sygnał `reset` lub ponowne uruchomienie urządzenia.

6.2.4 Stałe wbudowane protokołu

Parametr	Znaczenie	Wartość
<code>ignoreStep</code>	bazowy czas ignorowania po odebraniu <code>discovery</code>	10 ms
<code>maxRnd</code>	maksymalna krotność <code>ignoreStep</code>	24
<code>recvFrameTime</code>	czas po jakim flaga <code>recvFrame</code> zostanie wyzerowana	130 ms
<code>maxNullCounts</code>	liczba pustych zapytań kończąca proces wyszukiwania	6
<code>maxRespWait</code>	maksymalny czas oczekiwania na <code>response</code>	40 ms

Tab. 6.2: Parametry protokołu sterowania magistrali.

7 Przykładowe wykorzystanie urządzenia

Aby ułatwić zrozumienie protokołów i zaprezentować wykorzystanie SER2I2C przygotowano przykładowy program napisany w języku C++. Do komunikacji poprzez port RS-232 wykorzystano WinAPI, struktura kodu składa się z następujących plików:

- main.cpp – główny plik z funkcją main,
- SER2I2Cfunc.hpp/cpp – pliki z funkcjami sterującymi SER2I2C,
- serialdevice.hpp/cpp – pliki z klasą realizującą połączenie poprzez port szeregowy.

Do skompilowania kodu wymagane są:

- kompilator C++ zgodny ze standardem c++11,
- Microsoft Windows SDK.

W kodzie przedstawiono korzystanie z SER2I2C połączonych na jednej magistrali RS-485 (patrz rysunek 6.2) pracujących w trybie I²C – RS-485 (nie licząc listowania SER2I2C na magistrali, przedstawione funkcja można wykorzystać w pozostałych trybach SER2I2C – XXX).

W programie zaprezentowano sposób wykrywania urządzeń na magistrali, zmiany parametrów I²C i RS oraz przesyłania danych poprzez I²C (na podstawie wykrywania i identyfikowania ZR-2).