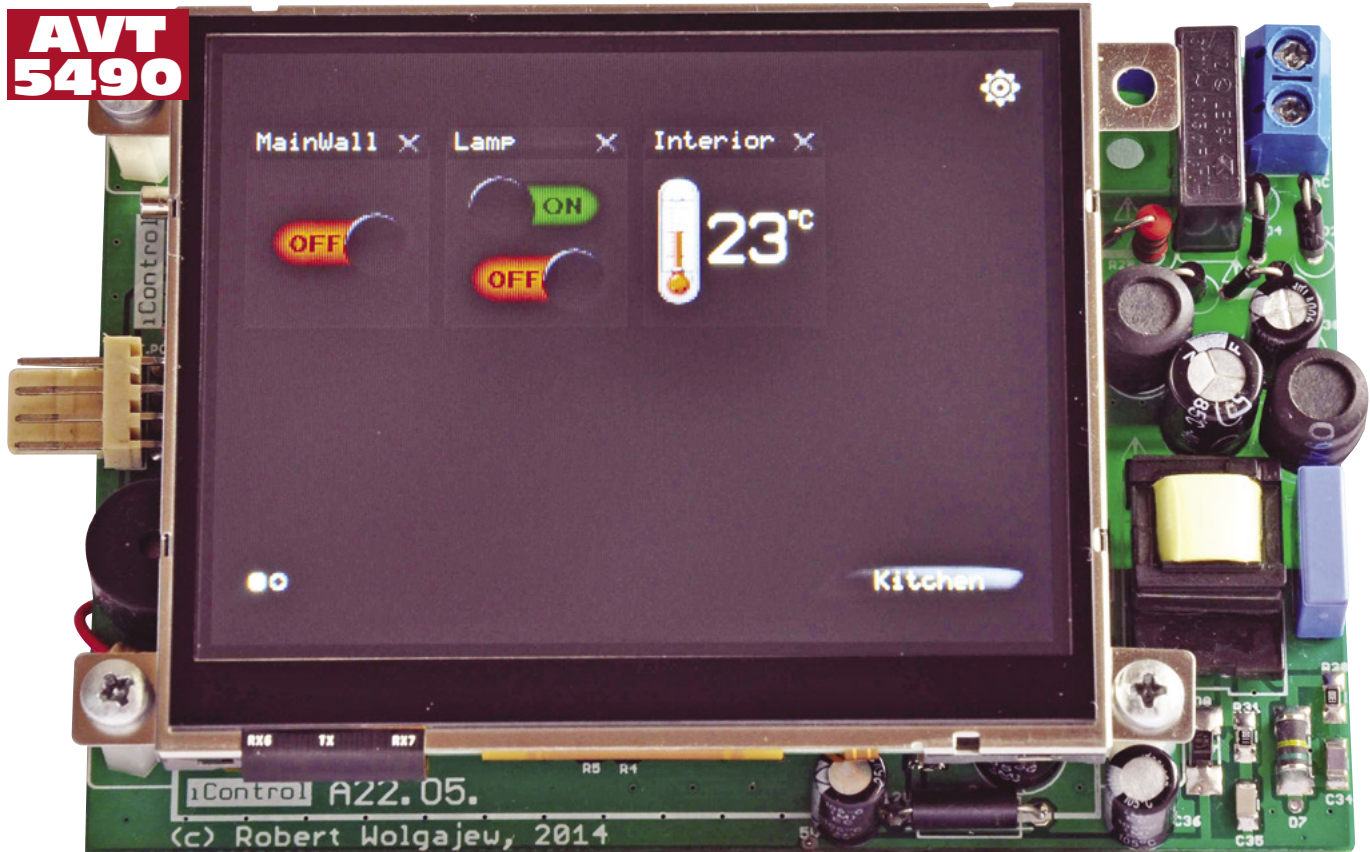


Podziękowania

Autor składa serdeczne podziękowania pani Beacie Stosik z firmy STOFL za dostarczenie wielu darmowych próbek układów CY8CPLC10, panu Sławomirowi Szweda z firmy Unisystem za dostarczenie darmowych modułów wyświetlaczy TFT oraz panu Dariuszowi Kowalczyk za cenne uwagi przekazywane w trakcie projektowania urządzenia.

Author would like to thank Mr Prem Sai V from Cypress Semiconductor customer support team for his huge and inestimable support and commitment while developing the device!



iControl

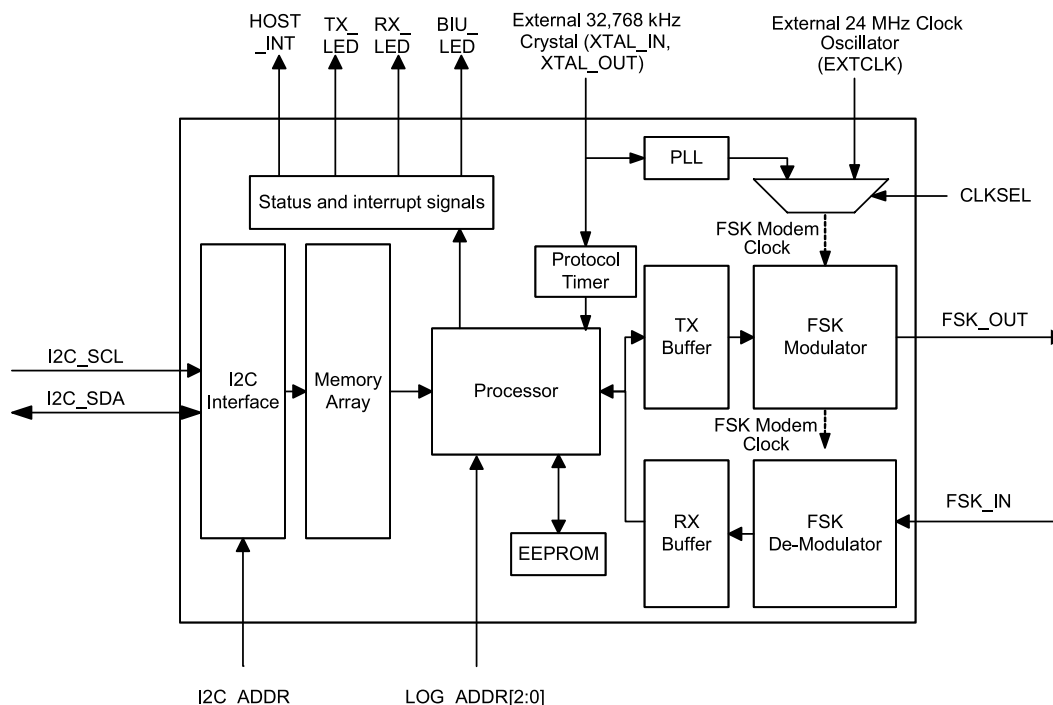
System automatyki domowej (1)



Film demonstrujący działanie systemu iControl:
<http://goo.gl/Q99jyj>

W swojej praktyce konstruktora-programisty kilkakrotnie podejmowałem wyzwanie skonstruowania prostego systemu sterowania i kontroli typu „inteligentny dom”, stosując w tym celu różne interfejsy w roli medium umożliwiającego współpracę specjalizowanych modułów. Jednym z przykładów takiego dość rozbudowanego i nowoczesnego projektu może być system intelli-Dom opublikowany na łamach Elektroniki Praktycznej 10...11/2011, który wykorzystywał specjalizowane i bardzo dobrze wyposażone moduły modemów ZigBee. System ten, mimo nie dawał jednak pełnej elastyczności w zakresie konfigurowania sieci modułów wykonawczych, gdyż korzystał z predefiniowanej funkcjonalności tychże modułów nazywanych tam „modułami pokojowymi”, a poza tym używał transmisji bezprzewodowej o ograniczonym zasięgu, realizowanej w paśmie 2,4 GHz, opartej na standardzie IEEE 802.15.4. Czas więc na projekt zaawansowany, w pełni konfigurowalny i pozbawiony poprzednich ograniczeń a dodatkowo wyposażony w ultranowoczesny interfejs użytkownika.

Rekomendacje: projekt przyda się jako baza dla rozbudowanego systemu zarządzania inteligentnym budynkiem.



Rysunek 1. Schemat blokowy układu CY8CPLC10

Pierwszym, naturalnym wyborem, z którym musiałem się zmierzyć, był oczywiście wybór rodzaju medium transmisyjnego. Tym razem i nie bez dłuższego namysłu, zdecydowałem się na wybór technologii znanej pod nazwą PLC (*Power Line Communication*). Jest to technologia transmisji danych oparta na przesyłaniu równoległe z napięciem zasilającym o częstotliwości 50 (lub 60) Hz sygnału o wiele wyższej częstotliwości (od kilku kiloherców do kilkudziesięciu megaherców) zawierającego dane.

Pomysł wykorzystania medium, którym jest kabel sieci elektrycznej nie jest nowy i zrodził się już wiele lat temu, zaś jego początki związane były z potrzebą zapewnienia taniego sposobu dostępu do Internetu w miejscach, gdzie inne możliwości były niedostępne. Sieci PLC zostały podzielone na PLC wąskopasmowe (NPL – *Narrow Powerline Communication*) i szerokopasmowe (BPL – *Broadband Power Line*). Wąskopasmowe sieci PLC odniosły sukces dzięki wykorzystaniu w systemach telemetrii i inteligentnych sieciach elektroenergetycznych (tzw. Smart Grid). Współcześnie ich głównym zastosowaniem jest zdalny odczyt liczników, sterowanie ruchem ulicznym, oświetleniem itp. Szerokopasmowe sieci PLC wykorzystywano również do transmisji danych w zakresie dostępu do Internetu i w lokalnych sieciach domowych. Mimo pewnych wad tej technologii, jakimi może być zakłócenie środowisko, niezbyt szybki transfer danych i problemy ze spełnieniem wymogów kompatybilności elektromagnetycznej, jest ona nadal rozwijana w wielu krajach, w tym także i w Polsce. Dużym przełomem w rozwoju szerokopasmowego PLC było ratyfikowanie w 2010 r. przez organizację IEEE standardu

1901. W najnowszych generacjach urządzeń opartych na tym standardzie wyeliminowano problemy z wolnym tempem transmisji danych, niewielkim zasięgiem oraz ograniczeniami związanymi z nieprzenikaniem sygnału pomiędzy fazami elektrycznymi w sieciach trójfazowych. Nowoczesne urządzenia wykorzystujące szerokopasmowe PLC mają prędkość fizyczną do 500 Mbit/s i zasięg aż do 500 metrów. Działają niezawodnie bez względu na stosowane fazy energii elektrycznej.

W takim razie pora na wybór rozwiązania układowego. Poszukiwania rozpocząłem od przejrzania oferty producentów półprzewodników w zakresie gotowych układów scalonych pełniących rolę modemów PLC. Szybko okazało się, że nader często stosowanym układem pełniącym rolę interfejsu PLC jest TDA5051 firmy NXP (wcześniej Philips) będący zintegrowanym modemem PLC zapewniającym transmisję danych opartą o modulację ASK (*Amplitude Shift Keying*) z maksymalną prędkością na poziomie 1200 bitów na sekundę. Układ ten, mimo że dość chętnie stosowany (choć już „leciwy”), ma dwie podstawowe wady. Po pierwsze, zastosowana modulacja ASK jest stosunkowo podatna na różne zakłócenia występujące w sieci elektroenergetycznej (zwłaszcza zakłócenia komutacji i generowane przez odbiorniki o charakterze indukcyjnym), a po drugie, nie zaimplementowano w nim żadnego stosu komunikacyjnego zapewniającego chociażby wielodostęp do medium transmisyjnego pozostawiając to zadanie po stronie projektanta systemu. Mimo to, zacząłem projektowanie systemu oraz oprogramowywanie prostego stosu komunikacyjnego z wykorzystaniem kodowania typu Manchester jako remedium na zakłócenia sieciowe.

W ofercie AVT* AVT-5490 A

Podstawowe informacje:

- Maksymalnie 64 modułów wykonawczych (typu Slave) w ramach jednej sieci systemu iControl.
- Maksymalnie 16 modułów sterujących (typu Master) wyposażonych w interfejs użytkownika z wyświetlaczem TFT.
- Adresy logiczne modułów wykonawczych nadawane są automatycznie przez moduły sterujące podczas konfiguracji sieci, zaś adresy logiczne modułów sterujących nadawane są przez użytkownika za pomocą interfejsu użytkownika GUI.
- Każdy moduł sterujący może zapamiętać i zaadresować 64 moduły wykonawcze.
- Wszystkie moduły wykonawcze zapamiętane przez dany moduł sterujący mogą zostać połączone w maksymalnie 8 grup, dowolnie podczas konfiguracji sieci, reprezentujących pomieszczenia, nad którymi moduł ten ma kontrolę (np. pokoje).
- Kilka modułów sterujących może mieć kontrolę nad jednym modułem wykonawczym.
- W ramach graficznego interfejsu użytkownika modułu sterującego każdy moduł wykonawczy jest identyfikowany przez unikalną nazwę.
- Każda z 8 możliwych grup, w które mogą być łączone moduły wykonawcze może mieć nadaną nazwę, aktywowana lub wyłączona.
- Przewidziano 5 rodzajów modułów wykonawczych: wyłącznik 1-biegunowy, wyłącznik 2-biegunowy, ściemniacz, sensor temperatury, sterownik oświetlenia RGB LED.
- System iControl sygnalizuje dołączenie nowych, jeszcze nieskonfigurowanych modułów wykonawczych oraz wystąpienie błędów transmisji.
- System iControl umożliwia usuwanie modułów wykonawczych z sieci, a co za tym idzie – rekonfigurację sieci.

Dodatkowe materiały na FTP:

<ftp://ep.com.pl>, user: 32086, pass: sqz8sawb

• wzory płytek PCB

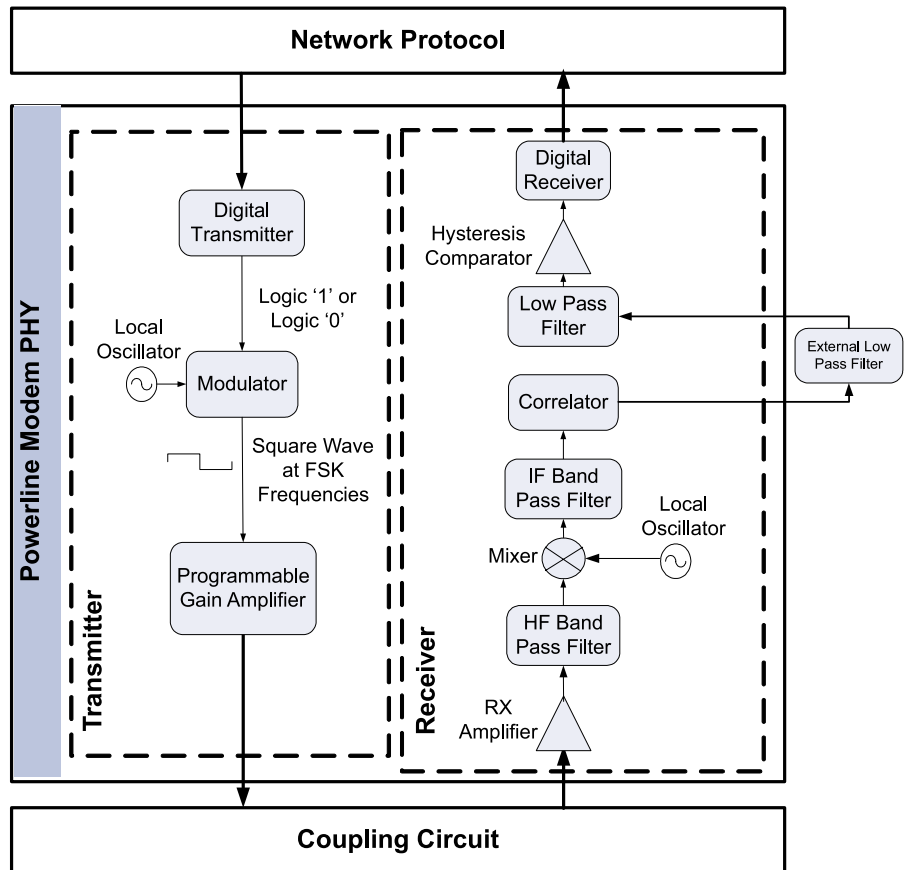
* Uwaga:
Zestawy AVT mogą występować w następujących wersjach:
AVT xxxx UK to zaprogramowany układ. Tylko i wyłącznie. Bez elementów dodatkowych.
AVT xxxx A płytka drukowana PCB (lub płytki drukowane, jeśli w opisie wyraźnie zaznaczono), bez elementów dodatkowych.
AVT xxxx A+ płytka drukowana i zaprogramowany układ (czyli połączenie wersji A i wersji UK) bez elementów dodatkowych.
AVT xxxx B płytka drukowana (lub płytki) oraz komplet elementów wymienionych w załączniku pdf
AVT xxxx C to nic innego jak zmontowany zestaw B, czyli elementy wmontowane w PCB. Należy mieć na uwadze, że o ile nie zaznaczono wyraźnie w opisie, zestaw ten nie ma obudowy ani elementów dodatkowych, które nie zostały wymienione w załączniku pdf
AVT xxxx CD oprogramowanie (nieczęsto spotykana wersja, lecz jeśli występuje, to niezbędne oprogramowanie można pobrać, klikając w link umieszczony w opisie kitu)

Nie każdy zestaw AVT występuje we wszystkich wersjach! Każda wersja ma załączony ten sam plik pdf! Podczas składania zamówienia upewnij się, którą wersję zamawiasz! (UK, A, A+, B lub C). <http://sklep.avt.pl>

Problemem, z którym musiałem się zmierzyć, był wybór mechanizmu zapewniającego wielodostęp do medium transmisyjnego oraz fakt, że w rozwiązaniach wykorzystujących modulację ASK i układy z rodziny TDA5051 użyteczna transmisja danych jest przeprowadzana w krótkim oknie czasowym tuż przed i tuż po przejściu napięcia zasilającego przez zero w celu zminimalizowania wpływu zakłóceń na transmisję użytecznych danych (sposób stosowany na przykład w protokole X10 przeznaczonym do sterowania urządzeniami domowymi). Ta właściwość powoduje, że zmniejsza się użyteczna, wynikowa prędkość transmisji danych, co pogarsza właściwości użytkowe tak skonstruowanej sieci. Mimo że moje prace w tym kierunku były już dość zaawansowane, zacząłem poszukiwania lepszego rozwiązania układowego. I właśnie w tym czasie natknąłem się na doskonale rozwiązanie firmy Cypress, którym jest układ scalony CY8CPLC10. Chip ten jest scalonym, zaawansowanym i kompletnym rozwiązaniem modemu PLC wykorzystującym modulację FSK (*Frequency Shift Keying*), mającym zaimplementowany stos komunikacyjny.

Układ CY8CPLC10 charakteryzuje się następującymi, wybranymi cechami użytkowymi:

- Zintegrowany interfejs fizyczny modemu PLC (*Physical Layer Interface*).
- Modulacja FSK zapewniająca prędkość transmisji na poziomie 2400 bitów na sekundę.
- Kompletny protokół komunikacyjny zoptymalizowany dla sieci PLC zapewniający realizację wszystkich warstw komunikacji (*Data Link, Transport i Network Layers*).
- Komunikacja dwukierunkowa typu half duplex z korekcją błędów (8 bitowe CRC) i sygnałem potwierdzenia (ACK).
- Wbudowany mechanizm wielodostępu do medium komunikacyjnego (CSMA - *Carrier Sense Multiple Access*).
- Zintegrowany interfejs sterujący I²C pracujący w trybie high-speed (400 kHz).
- Wsparcie dla sieci zmiennoprądowych AC 110...240 V i stałoprądowych DC 12...24 V.
- Wsparcie dla wielu trybów transmisji danych w sieci: master-slave, peer-to-peer i multimaster.
- 3 tryby adresowania urządzeń sieciowych: logiczny (8-bitów), rozszerzony logiczny (16-bitów) i fizyczny (64-bity).
- 2 tryby rozgłaszania adresów urządzeń: indywidualny i grupowy (możliwość przyporządkowania urządzeń do jednej z 256 grup i sterowania grupowego).
- Szereg kilkudziesięciu rejestrów kontrolnych/sterujących dających pełną kontrolę nad transmisją danych.
- Wbudowany detektor BIU (*Band-In-Use*) zgodny ze standardem CENELEC EN 50065-1.



Rysunek 2. Schemat funkcjonalny wbudowanego w układ CY8CPLC10 modemu PLC

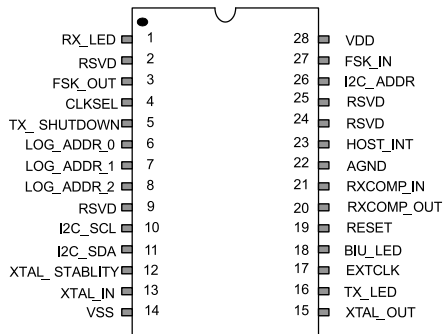
- Zgodność ze standardem CENELEC EN50065-1:2001 i FCC part 15 (dla Ameryki Północnej).

Już z pobieżnej analizy długiej listy możliwości widać, że układ idealnie wpisuje się w potrzeby bieżącego projektu zwalniając nas jednocześnie z konieczności rozwiązania wielu problemów technicznych wynikających z konstrukcji stosu komunikacyjnego i mechanizmów dostępu do medium transmisyjnego.

Na **rysunku 1** pokazano diagram obrazujący schemat blokowy układu CY8CPLC10, natomiast na **rysunku 2** diagram obrazujący schemat funkcjonalny wbudowanego modemu PLC.

Warto na chwilę zatrzymać się nad schematem funkcjonalnym modemu PLC, aby poznać sposób nadawania i odbioru pakietów danych. W konstrukcji nadajnika modemu PLC wykorzystano modulator, który „zasilany” kolejnymi bitami przesyłanych danych generuje sygnał wyjściowy o dwóch możliwych częstotliwościach, zależnych od wartości aktualnie przetwarzanego bitu danych. Dla bitu równego „0” jest to częstotliwość 133,3 kHz, zaś dla bitu o wartości „1” – 131,8 kHz (lub 130,4 kHz – w zależności od ustawień konfiguracyjnych). Wyjście modulatora jest połączone ze wzmacniaczem sygnału o regulowanym programowo wzmacnieniu, a następnie wyprowadzone na zewnątrz układu (pin FSK_OUT) dostarczając w ten sposób sygnał o modulacji FSK.

W części odbiorczej mamy do czynienia z dużo bardziej rozbudowanym układem. Zmodulowany, sieciowy sygnał wejściowy (pin FSK_IN) jest wprowadzany do wewnętrznego filtra pasmowoprzepustowego, który zawęża pasmo sygnału użytecznego do 125 kHz...140 kHz, po czym ten sygnał „zasila” demodulator zbudowany z prostego miksera, lokalnego generatora częstotliwości wzorcowej, filtra częstotliwości pośredniej i korelatora. Sygnał wyjściowy korelatora, który zawiera już zdemodulowany przebieg wyjściowy odpowiadający przesyłanym danym z komponentami o wyższych częstotliwościach, wchodzi na wejście zewnętrznego filtra dolnoprzepustowego (piny RXCOMP_IN i RXCOMP_OUT) o częstotliwości odcięcia równej 7,5 kHz, a następnie na wejście kolejnego, wewnętrznego filtra dolnoprzepustowego, który tłumí wszystkie wyższe składowe częstotliwości generowane w procesie korelacji. Sygnał z wyjścia tego filtra przechodzi jeszcze przez układ komparatora z histerezą, który służy eliminowaniu potencjalnych zakłóceń i korygowania opóźnień wnoszonego przez korelator. Wyjście z komparatora „zasila” rejestr przesuwany, który zamienia strumień szeregowy na wyjściowe pakiety danych poddawane następnie analizie przez mechanizmy warstwy sieciowej stosu komunikacyjnego. Rozmieszczenie wyprowadzeń układu CY8CPLC10 pokazano na **rysunku 3** zaś ich opis funkcjonalny umieszczono w **tabeli 1**.



Rysunek 3. Rozmieszczenie wyprowadzeń układu CY8CPLC10

Można zauważyć jak wiele komponentów warstwy fizycznej jest zaangażowanych w proces dekodowania sygnału PLC, który w dalszej kolejności trafia do części logicznej, odpowiedzialnej za właściwą interpretację danych, ich integralność oraz odpowiednią synchronizację.

Warstwa sieciowa korzysta z ciekawej konstrukcji ramki pakietu danych, której to wygląd pokazano na **rysunku 4**. Można zauważyć, że struktura ramki składa się z nagłówka o zmiennej długości (6 do 20 bajtów) zależnej od typu przesyłanych adresów urządzenia źródłowego (**SA Type**) i urządzenia docelowego (**DA Type**), dołączonych danych o długości od 0 do 31 bajtów oraz sumy kontrolnej CRC8. Pierwszy bajt nagłówka (0x00) zawiera następujące informacje, determinujące logiczny podział całego pakietu danych:

- Pole **SA Type** (bit 7) określa rodzaj przesyłanego adresu urządzenia źródłowego (wysyłającego pakiet danych) według następującej specyfikacji: 0 → adres logiczny (8-bitów), 1 → adres fizyczny (unikalne 64-bity zdefiniowane przez producenta układu).
- Pole **DA Type** (bity 6..5) określa rodzaj przesyłanego adresu urządzenia docelowego (będącego adresem przesyłanego pakietu danych) według następującej specyfikacji: 0 → adres logiczny (8-bitów), 1 → adres grupowy (8-bitów), 2 → adres fizyczny (unikalne 64-bity zdefiniowane przez producenta układu).
- Pole **Service Type** (bit 4) określa konieczność (jeśli ustawione) potwierdzania wysyłanych pakietów danych determinując generowanie sygnału ACK przez strony transmisji.
- Pole **Response** (bit 1) determinuje czy przesyłany pakiet jest odpowiedzią na żądanie przesłania danych (jeśli ustawione) czy też „zwykłą” transmisją danych.

Znaczenie kolejnych pól danych pozostałych bajtów pakietu przedstawia się następująco:

- Bajt(y) **Destination Address** (0x01) określa(ją) adres urządzenia docelowego. Ponieważ dopuszczalne są 3 sposoby adresowania różniące się długością

Nr	Nazwa	Opis
1	RX_LED	Opcjonalna dioda LED sygnalizująca odbiór danych PLC
2		
3	FSK_OUT	Analogowe wyjście sygnału PLC z modulacją FSK
4	CLKSEL	Wybór źródła sygnału taktującego układ: 0→zewnętrzny sygnał zegarowy o częstotliwości 24 MHz doprowadzony do wejścia EXTCLK, 1→rezonator kwarcowy 32768 Hz podłączony do pinów XTAL_IN i XTAL_OUT
5	TX_SHUTDOWN	Wyjście służące do blokowania zewnętrznego układu transmisji (wzmocniacza) podczas odbioru danych: 0→gdy modem transmituje dane, 1→gdy modem nie transmituje danych.
6	LOG_ADDR_0	Bit LA0 adresu logicznego urządzenia, w przypadku, gdy adres ten nie został ustawiony programowo przez kontroler-host (wejście zanegowane).
7	LOG_ADDR_1	Bit LA1 j.w.
8	LOG_ADDR_2	Bit LA2 j.w.
9		
10	I2C_SCL	Sygnal zegarowy magistrali I ² C
11	I2C_SDA	Sygnal danych magistrali I ² C
12	XTAL_STABILITY	Stabilizacja oscylatora zegara taktującego. Między to wyprowadzenie a VSS należy podłączyć kondensator 100 nF
13	XTAL_IN	Wejście do podłączenia rezonatora kwarcowego 32768 Hz
14	VSS	Masa zasilania
15	XTAL_OUT	Wyjście do podłączenia rezonatora kwarcowego 32768 Hz
16	TX_LED	Opcjonalna dioda LED sygnalizująca transmisję danych PLC
17	EXTCLK	Opcjonalny, zewnętrzny sygnał zegarowy o częstotliwości 24 MHz
18	BIU_LED	Opcjonalna dioda LED sygnalizująca zajętość magistrali PLC
19	RESET	Zerowanie układu
20	RXCOMP_OUT	Wyjście analogowe do podłączenia zewnętrznego filtra dolnoprzepustowego (RC) 7.5 kHz
21	RXCOMP_IN	Wejście analogowe do podłączenia zewnętrznego filtra dolnoprzepustowego (RC) 7.5 kHz
22	AGND	Masa zasilania części analogowej. Między to wyprowadzenie a VSS należy podłączyć kondensator 1 μF
23	HOST_INT	Wyjście zgłoszenia przerwania do kontrolera-hosta. Rodzaj zdarzeń generujących przerwanie i polaryzacja tego wyjścia podlega konfiguracji użytkownika.
24		
25		
26	I2C_ADDR	Wejście wyboru 7-bitowego adresu układu na magistrali I ² C: 0→adres 0x7A, 1→adres 0x01
27	FSK_IN	Analogowe wejście sygnału PLC z modulacją FSK
28	VDD	Napięcie zasilania (5 V)

(8 lub 64 bity), offset kolejnych bajtów w zakresie nagłówka pakietu danych może podlegać zmianie.

- Bajt(y) **Source Address** (0x02) określa(ją) adres urządzenia źródłowego. Ponieważ dopuszczalne są 3 sposoby adresowania różniące się długością (8 lub 64 bity), offset kolejnych bajtów w zakresie nagłówka pakietu danych może podlegać zmianie.

- Bajt **Command** (0x03) określa rodzaj przesyłanego rozkazu w ramach predefiniowanych rozkazów stosu komunikacyjnego, które to mogą powodować podjęcie różnych akcji automatycznych lub też powodować zmiany konfiguracji urządzenia

Byte Offset	Bit Offset							
	7	6	5	4	3	2	1	0
0x00	SA Type	DA Type	Service Type				Response	
0x01	Destination Address (8-bit Logical, 16-bit Extended Logical or 64-bit Physical)							
0x02	Source Address (8-bit Logical, 16-bit Extended Logical or 64-bit Physical)							
0x03	Command							
0x04	Payload Length							
0x05	Seq Num				Powerline Packet Header CRC			
0x06	Payload (0 to 31 Bytes)							
	Powerline Transceiver Packet CRC							

Rysunek 4. Budowa ramki pakietu danych układu CY8CPLC10

docelowego, jak też w ramach zdefiniowanych przez użytkownika własnych rozkazów sterujących właściwych dla implementowanej aplikacji.

- Pole **Payload Length** (bity 4...0 bajtu 0x04) określa długość (w bajtach) pakietu dołączonych danych.
- Pole **Seq Num** (bity 7...4 bajtu 0x05) jest używane przez stos komunikacyjny w wypadku retransmisji pakietu danych, na skutek błędów transmisji. Pole to podlega inkrementacji za każdym razem, gdy przesyłany jest zupełnie nowy pakiet danych, natomiast pozostaje bez zmian w przypadku retransmisji pakietu bieżącego. Dopuszczalna liczba retransmisji podlega konfiguracji ze strony użytkownika. Dzięki takiej konstrukcji stosu komunikacyjnego unika się duplikacji odbieranych danych po stronie kontrolera-hosta w przypadkach ich wielokrotnych retransmisji.
- Pole **Powerline Packet Header CRC** (bity 3...0 bajtu 0x05) zawiera sumę kontrolną nagłówka pakietu danych (wykorzystywaną przez mechanizmy stosu komunikacyjnego).
- Bajty **Payload** to użyteczne dane przesyłanego pakietu danych (dane użytkownika).
- Bajt **Powerline Transceiver Packet CRC** zawiera sumę kontrolną CRC8 całego pakietu danych, w tym nagłówka i danych użytecznych (wykorzystywaną przez mechanizmy stosu komunikacyjnego).

Firma Cypress zastosowała bardzo przemyślaną strukturę pakietu danych, co zaowocowało dużą elastycznością funkcjonalną układu zastosowanego w tym projekcie, otwierając nowe możliwości stosu komunikacyjnego. Kilka słów uwagi należy w tym miejscu poświęcić sposobom adresowania urządzeń w tak skonstruowanej sieci PLC.

Każdy układ CY8CPLC10 ma stały, nadany przez producenta w procesie produkcji, unikalny, 64-bitowy adres fizyczny (wzorem adresu MAC), który jednoznacznie identyfikuje go w sieci. Aby jednak zmniejszyć ruch w sieci spowodowany wysyłaniem dłuższych pakietów danych, które zawierają 64-bitowe adresy urządzenia źródłowego i docelowego, możliwe jest nadanie każdemu z modemów unikalnego, 8-bitowego adresu logicznego, co jednocześnie powoduje, że maksymalna liczba urządzeń, które mogą współpracować w tak skonfigurowanej sieci wyniesie wtedy 256. Jako opcjonalny rodzaj adresu, jeśli wymagana przez nas liczba urządzeń przekracza wartość 256, jest możliwe ustawienie rozszerzonego, 16-bitowego adresu logicznego, co powoduje zwiększenie liczby możliwych urządzeń do 65536. Jakby tego było mało, dostępne są 3 sposoby ustawienia 8-bitowego adresu logicznego:

- Przez kontrolera-hosta przy udziale magistrali I²C.
- Poprzez odpowiednie skonfigurowanie wyprowadzeń układu oznaczonych LOG_ADDR_2... LOG_ADDR_0

(jednak wtedy dostępnych jest wyłącznie 8 adresów).

- Zdalnie, za pomocą innego modemu PLC korzystającego z predefiniowanych rozkazów stosu PLC pod warunkiem, że **modem odbierający tego rodzaju rozkaz jest skonfigurowany w taki sposób by nie ignorował go!**

Oprócz możliwości adresowania indywidualnych urządzeń w implementacji stosu komunikacyjnego przewidziano możliwość pracy rozgłoszeniowej, to znaczy możliwość wysyłania pakietów danych do wielu urządzeń docelowych w jednym czasie (tzw. *multicast messages*). Jakby tego było mało, przewidziano dwa sposoby adresowania grupowego, które mogą być aktywne dla każdego urządzenia w tym samym czasie. Każde z urządzeń wyposażonych w układ CY8CPLC10 może być skonfigurowane w taki sposób by było członkiem jednej z 256 dostępnych grup urządzeń lub też może zostać przydzielone do 8 grup w tym samym czasie. Co oczywiste, z założenia przy wysyłaniu wiadomości grupowej wyłączone zostaje potwierdzanie wysyłanych pakietów danych bo w tym wypadku traci ono sens. Imponujące możliwości, nieprawdaż?

Rejestry konfiguracyjne CY8CPLC10

W tym miejscu dysponujemy już sporą dawką wiedzy na temat naszego modemu PLC, jednak by w pełni zrozumieć „drzemiące” w nim możliwości nie sposób nie przedstawić, choćby po krótko, najważniejszych rejestrów konfiguracyjnych, przy udziale których mamy dostęp do pełnej funkcjonalności układu.

Rejestr: INT_ENABLE (0x00)							
D7	D6	D5	D4	D3	D2	D1	D0
INT_CLEAR	INT_POLARITY	INT_UNABLE_TO_TX	INT_NO_ACK	INT_NO_RESP	INT_RX_PACKET_DROPPED	INT_DATA_AVAILABLE	INT_DATA_SENT

Znaczenie poszczególnych bitów rejestru **INT_ENABLE** przedstawia się następująco:

- Bit **INT_CLEAR** jest ustawiany przez układ CY8CPLC10 za każdym razem, gdy zajdzie zdarzenie, dla którego uruchomiono generowanie przerw na wyprowadzeniu HOST_INT. Użytkownik powinien wyzerować ten bit po odczytaniu rejestru statusu INT_STATUS, który przechowuje flagi zdarzeń (spowoduje to wykasowanie flag tegoż rejestru za wyjątkiem flag: STATUS_RX_PACKET_DROPPED i STATUS_RX_DATA_AVAILABLE).
- Bit **INT_POLARITY** decyduje o polaryzacji sygnału na wyjściu HOST_INT: 0 → aktywny stan wysoki, 1 → aktywny stan niski.
- Bit **INT_UNABLE_TO_TX** uruchamia (gdy ustawiony) generowanie przerw w przypadku, gdy modem PLC nie jest

w stanie wysłać pakietu danych (upływa maksymalny czas w jakim modem PLC stara się wysłać pakiet danych sprawdzając ustawicznie zajętość magistrali).

- Bit **INT_NO_ACK** uruchamia (gdy ustawiony i wybrano potwierdzanie pakietów danych) generowanie przerw w przypadku nieotrzymania potwierdzenia odbioru pakietu danych (od adresata wiadomości).
- Bit **INT_NO_RESP** uruchamia (gdy ustawiony) generowanie przerw w przypadku, gdy modem nie otrzymał (w ustalonym czasie) odpowiedzi od adresata, od którego zażądał wysłania danych (wybrane, predefiniowane rozkazy stosu komunikacyjnego).
- Bit **INT_RX_PACKET_DROPPED** uruchamia (gdy ustawiony) generowanie przerw w wypadku przepełnienia (nadpisania) bufora danych odbiorczych.
- Bit **INT_DATA_AVAILABLE** uruchamia (gdy ustawiony) generowanie przerw w przypadku odebrania pakietu danych.
- Bit **INT_DATA_SENT** uruchamia (gdy ustawiony) generowanie przerw w przypadku pomyślnego wysłania pakietu danych.

Rejestr: LOCAL_LA_LSB (0x01)							
D7	D6	D5	D4	D3	D2	D1	D0
LA7	LA6	LA5	LA4	LA3	LA2	LA1	LA0

Wartość rejestru **LOCAL_LA_LSB** określa 8-bitowy adres logiczny modemu PLC lub młodszy bajt 16-bitowego, rozszerzonego adresu logicznego (w przypadku korzystania z tego typu adresacji).

Rejestr: LOCAL_LA_MSB (0x02)							
D7	D6	D5	D4	D3	D2	D1	D0
LA7	LA6	LA5	LA4	LA3	LA2	LA1	LA0

Wartość rejestru **LOCAL_LA_MSB** określa starszy bajt 16-bitowego, rozszerzonego adresu logicznego (w przypadku korzystania z tego typu adresacji).

Rejestr: LOCAL_GROUP (0x03)							
D7	D6	D5	D4	D3	D2	D1	D0
GA7	GA6	GA5	GA4	GA3	GA2	GA1	GA0

Wartość rejestru **LOCAL_GROUP** określa 8-bitowy adres grupy, do której przyporządkowano modem PLC (w przypadku odbioru wiadomości typu multicast).

Rejestr: LOCAL_GROUP_HOT (0x04)							
D7	D6	D5	D4	D3	D2	D1	D0
GH7	GH6	GH5	GH4	GH3	GH2	GH1	GH0

Wartość każdego z bitów rejestru **LOCAL_GROUP_HOT** określa przynależność modemu PLC do dodatkowo zdefiniowanych grup (o numerach od 7 do 0, odpowiednio dla każdego z bitów rejestru). Na przykład, wpisanie do tego rejestru wartości 0b00010001 spowoduje, iż modem PLC należeć będzie do dodatkowych grup o numerach #5 i #1.

Rejestr: PLC_MODE (0x05)							
D7	D6	D5	D4	D3	D2	D1	D0
TX_ENABLE	RX_ENABLE	LOCK_CONFIG	DI-SABLE_BIU	RX_OVERWRITE	SET_EXT_ADDRESS	PROMISCUOUS_MASK	PROMISCUOUS_CRC_MASK

Rejestr **PLC_MODE** określa podstawowe właściwości sprzętowe modemu PLC. Poszczególne bity tego rejestru mają następujące znaczenie funkcjonalne:

- Bit **TX_ENABLE** uruchamia (gdy ustawiony) nadajnik modemu PLC.
- Bit **RX_ENABLE** uruchamia (gdy ustawiony) odbiornik modemu PLC.
- Bit **LOCK_CONFIG** uniemożliwia (gdy ustawiony) zdalną (przy udziale innego modemu PLC) zmianę konfiguracji modemu PLC
- Bit **DISABLE_BIU** wyłącza (gdy ustawiony) detektor zajętości magistrali PLC (mechanizm Band-In-Use).

Ustawienie bitu **RX_OVERWRITE** powoduje, iż nowe dane odebrane przez modem nadpisują dane ostatnio odebrane, nawet, jeśli nie zostały jeszcze odczytane.

Wartość bitu **SET_EXT_ADDRESS** określa rodzaj stosowanego adresu logicznego: 0 → 8-bitowy adres logiczny, 1 → 16-bitowy, rozszerzony adres logiczny.

Wartość bitu **PROMISCUOUS_MASK** określa sposób interpretowania danych odbieranych przez modem PLC: 0 → modem PLC ignoruje wszystkie dane, jeśli adres przeznaczenia nie jest zgodny z jego adresem (logicznym lub fizycznym), 1 → modem PLC odbiera wszystkie dane niezależnie od adresu ich przeznaczenia, jeśli tylko zaopatrzone zostały w poprawną sumę kontrolną CRC8.

Wartość bitu **PROMISCUOUS_CRC_MASK** określa mechanizmy kontroli poprawności przesyłanych danych: 0 → modem PLC ignoruje wszystkie dane, dla których przesłano niepoprawną wartość sumy kontrolnej CRC8, 1 → modem PLC odbiera wszystkie dane niezależnie od wartości sumy kontrolnej CRC8 (jeśli tylko adres przeznaczenia przesyłanych danych zgadza się z adresem modemu).

Rejestr: TX_MESSAGE_LENGTH (0x06)							
D7	D6	D5	D4	D3	D2	D1	D0
SEND_MESSAGE			PAYLOAD_LENGTH_MASK				

Rejestr **TX_MESSAGE_LENGTH** określa długość pakietu danych przeznaczonych do wysłania (użytecznych danych użytkownika) → bity D4...D0 oraz inicjuje, poprzez ustawienie bitu **SEND_MESSAGE**, transmisję danych. Należy mieć na uwadze, iż przed inicjacją transmisji danych należy ustawić wartości wszystkich rejestrów odpowiedzialnych za parametry tejże transmisji jak i wartości samych danych, tj. rejestry: **TX_CONFIG**, **TX_DA**, **TX_COMMAND_ID** i **TX_DATA**.

Rejestr: TX_CONFIG (0x07)							
D7	D6	D5	D4	D3	D2	D1	D0
TX_SA_TYPE		TX_DA_TYPE		TX_SERVICE_TYPE		TX_RETRY	

Rejestr **TX_CONFIG** określa podstawowe parametry nadajnika modemu PLC. Poszczególne bity tego rejestru mają następujące znaczenie:

- Bit **TX_SA_TYPE** określa rodzaj adresu urządzenia źródłowego (wysyłającego dane): 0 → adres logiczny, 1 → adres fizyczny.
- Bity **TX_DA_TYPE** określają rodzaj adresu urządzenia docelowego (urządzenia, dla którego przeznaczone są wysyłane dane): 0 → adres logiczny, 1 → adres grupowy, 2 → adres fizyczny.
- Bit **TX_SERVICE_TYPE** określa konieczność (jeśli ustawiony) potwierdzania wysyłanych pakietów danych determinując generowanie sygnału ACK przez strony transmisji.
- Bity **TX_RETRY** określają liczbę dopuszczalnych retransmisji pakietu danych w przypadku niepowodzenia tejże transmisji.

Rejestr: TX_DA (0x08)							
D7	D6	D5	D4	D3	D2	D1	D0
REMOTE_NODE_DESTINATION_ADDRESS							

Rejestr **TX_DA** określa adres urządzenia docelowego (jeśli wybrany wcześniej rodzaj adresu rządu docelowego to adres logiczny lub grupowy) lub też definiuje najstarszy bajt 64-bitowego, fizycznego adresu urządzenia docelowego. W tym drugim wypadku, kolejne rejestry o adresach 0x09...0x0F definiują pozostałe bajty adresu fizycznego.

Rejestr: TX_COMMAND_ID (0x10)							
D7	D6	D5	D4	D3	D2	D1	D0
TX_COMMAND_ID							

Rejestr **TX_COMMAND_ID** determinuje rodzaj przesyłanego rozkazu do urządzenia docelowego. Predefiniowane rozkazy o numerach 0x01...0x0F powodują podjęcie przez modem docelowy różnych akcji automatycznych lub też generują zmiany jego konfiguracji, np. zdalną zmianę adresu logicznego modemu docelowego, żądanie danych, zdalny odczyt adresu fizycznego modemu docelowego itd i są ściśle określone w specyfikacji stosu komunikacyjnego. Rozkazy o numerach 0x30...0xFF nie są zdefiniowane przez stos komunikacyjny i przeznaczone są do swobodnego wykorzystania przez docelową aplikację użytkownika.

Rejestr: TX_DATA (0x11)							
D7	D6	D5	D4	D3	D2	D1	D0
TX_DATA							

Rejestr **TX_DATA** określa początek obszaru (od 0x11 do 0x2F) przeznaczonego na użyteczne dane przeznaczone do transmisji.

Rejestr: THRESHOLD_NOISE (0x30)							
D7	D6	D5	D4	D3	D2	D1	D0
AUTO_BIU_THRESHOLD				BIU_THRESHOLD_CONSTANT			

Rejestr **THRESHOLD_NOISE** określa parametry dla mechanizmu detekcji zajętości magistrali danych PLC (detektora BIU). Bity **BIU_THRESHOLD_CONSTANT** określają predefiniowany poziom sygnału (w dBμVrms), dla którego przyjmuje się, iż magistrala danych jest zajęta przez inne urządzenie PLC. Wartość domyślna (0x03) odpowiada ustawieniu 87 dBμVrms. Ustawienie bitu **AUTO_BIU_THRESHOLD** wymusza automatyczny dobór poziomu sygnału detektora BIU.

Rejestr: MODEM_CONFIG (0x31)							
D7	D6	D5	D4	D3	D2	D1	D0
TX_DELAY		MODEM_FSKBW_MASK			MODEM_BPS_MASK		

Rejestr **MODEM_CONFIG** określa podstawowe parametry modemu PLC. Poszczególne bity rejestru mają następujące znaczenie funkcjonalne:

- Bity **TX_DELAY** określają opóźnienie, jakie wprowadzane jest przez modem przed każdą transmisją danych: 0 → 7 ms, 1 → 13 ms, 2 → 19 ms, 3 → 25 ms.
- Bit **MODEM_FSKBW_MASK** determinuje wartości częstotliwości modulacji odpowiadające poziomom logicznym „0” i „1”: 0 → 133,3 kHz dla „0” i 131,8 kHz dla „1”, 1 → 133,3 kHz dla „0” i 130,4 kHz dla „1”. Co oczywiste, ustawienie to musi być takie same dla wszystkich urządzeń PLC komunikujących się w obrębie projektowanej sieci.
- Bity **MODEM_BPS_MASK** określają prędkość transmisji modemu PLC: 0 → 600 bps, 1 → 1200 bps, 2 → 1800 bps, 3 → 2400 bps (ustawienie domyślne). Co oczywiste, ustawienie to powinno być takie same dla wszystkich urządzeń PLC w obrębie projektowanej sieci.

Rejestr: TX_GAIN (0x32)							
D7	D6	D5	D4	D3	D2	D1	D0
TX_GAIN							

Rejestr **TX_GAIN** determinuje predefiniowane wzmocnienie sygnału dla wbudowanego w modem PLC nadajnika. Dostępne wartości z przedziału 0x00...0x0E odpowiadają ustawieniom 55 mVp-p...3.5 Vp-p (domyślna wartość 0x0B odpowiada ustawieniu 1.55 Vp-p). Zmiany ustawień tego rejestru niezbędne są w przypadku implementacji dodatkowego, zewnętrznego wzmacniacza sygnału nadajnika jak też dla spełnienia wymagań dotyczących poziomu sygnału transmisji PLC w zakresie odpowiednich norm.

Rejestr: RX_GAIN (0x33)							
D7	D6	D5	D4	D3	D2	D1	D0
RX_GAIN							

Rejestr **RX_GAIN** determinuje predefiniowane wzmocnienie sygnału dla wbudowanego

w modem PLC odbiornika. Dostępne wartości z przedziału 0x00...0x07 odpowiadają ustawieniom 5 mVrms...125 μVrms (domyślna wartość 0x00 odpowiada ustawieniu 5 mVrms). Zmiany ustawień tego rejestru niezbędne są w przypadku implementacji dodatkowego, zewnętrznego wzmacniacza sygnału odbiornika.

Rejestr: RX_MESSAGE_INFO (0x40)							
D7	D6	D5	D4	D3	D2	D1	D0
NEW_RX_MSG	RX_DA_TYPE	RX_SA_TYPE	RX_MSG_LENGTH				

Rejestr **RX_MESSAGE_INFO** udostępnia informacje na temat właściwości odebranego pakietu danych. Poszczególne bity tegoż rejestru mają następujące znaczenie funkcjonalne:

- Bit **NEW_RX_MSG** jest flagą odebrania nowego pakietu danych. Bit ten powinien być wyzerowany przez aplikację użytkownika po odczytaniu przesłanych danych by umożliwić dalszy odbiór pakietów danych. Wyzerowanie tego bitu powoduje także skasowanie flag **STATUS_VALUE_CHANGE**, **STATUS_RX_PACKET_DROPPED** i **STATUS_RX_DATA_AVAILABLE** w rejestrze flag przebrań **INT_STATUS_REGISTER**.
- Bit **RX_DA_TYPE** przechowuje informację o rodzaju adresu urządzenia docelowego, jaki zdefiniowano dla odebranego pakietu danych: 0 → adres logiczny/fizyczny, 1 → adres grupowy.
- Bit **RX_SA_TYPE** przechowuje informację o rodzaju adresu urządzenia źródłowego, jaki zdefiniowano dla odebranego pakietu danych: 0 → adres logiczny, 1 → adres fizyczny.
- Bity **RX_MSG_LENGTH** przechowuje informację o długości danych użytecznych w odebranym pakiecie danych.

Rejestr: RX_SA (0x41)							
D7	D6	D5	D4	D3	D2	D1	D0
REMOTE_NODE_SOURCE_ADDRESS							

Rejestr **RX_SA** określa adres urządzenia źródłowego, które wysłało do naszego modemu PLC pakiet danych. W zależności od wartości bitu **RX_SA_TYPE** w rejestrze **RX_MESSAGE_INFO** adres ten stanowi wartość wyłącznie rejestru 0x41 dla rodzaju adresu typu logicznego, wartości rejestrów 0x41 i 0x42 (jako MSB) dla rozszerzonego adresu logicznego i wartości rejestrów 0x41...0x48 (jako LSB) dla adresu fizycznego.

Rejestr: RX_COMMAND_ID (0x49)							
D7	D6	D5	D4	D3	D2	D1	D0
RX_COMMAND_ID							

Rejestr **RX_COMMAND_ID** przechowuje informację o rodzaju przesłanego rozkazu w bieżącym pakiecie odebranych danych.

Rejestr: RX_DATA (0x4A)							
D7	D6	D5	D4	D3	D2	D1	D0
RX_DATA							

Listing 1. Funkcje obsługujące interfejs TWI

```
void TWI_Init(void)
{
    TWBR = 6; //F_TWI=395kHz @ fosc=11059200Hz
}

void TWI_Start(void)
{
    TWCR = (1<<TWINT)|(1<<TWEN)|(1<<TWSTA);
    while (!(TWCR&(1<<TWINT)));
}

void TWI_Stop(void)
{
    TWCR = (1<<TWINT)|(1<<TWEN)|(1<<TWSTO);
    while (TWCR&(1<<TWSTO));
}

void TWI_WriteByte(uint8_t Byte)
{
    TWDR = Byte;
    TWCR = (1<<TWINT)|(1<<TWEN);
    while (!(TWCR&(1<<TWINT)));
}

uint8_t TWI_ReadByte(uint8_t ACK_NACK)
{
    TWCR = (1<<TWINT)|(ACK_NACK<<TWEA)|(1<<TWEN);
    while (!(TWCR & (1<<TWINT)));
    return TWDR;
}
```

Listing 2. Ciała funkcji odpowiedzialnych za zapis/odczyt rejestrów układu CY8CPLC10

```
void writePLCregister(uint8_t registerNr, uint8_t registerValue)
{
    TWI_Start();
    TWI_WriteByte(PLC_WRITE_ADDR);
    TWI_WriteByte(registerNr);
    TWI_WriteByte(registerValue);
    TWI_Stop();
    _delay_us(BUS_FREE_TIME_BEFORE_STO_STA);
}

uint8_t readPLCregister(uint8_t registerNr)
{
    register uint8_t registerValue;
    TWI_Start();
    TWI_WriteByte(PLC_WRITE_ADDR);
    TWI_WriteByte(registerNr);
    _delay_us(50);
    TWI_Start();
    TWI_WriteByte(PLC_READ_ADDR);
    registerValue = TWI_ReadByte(NACK);
    TWI_Stop();
    _delay_us(BUS_FREE_TIME_BEFORE_STO_STA);
    return registerValue;
}

void writePLCregisters(uint8_t startRegNr, volatile uint8_t *regValues,
uint8_t bytesToWrite)
{
    TWI_Start();
    TWI_WriteByte(PLC_WRITE_ADDR);
    TWI_WriteByte(startRegNr);
    while(bytesToWrite--) TWI_WriteByte(*(regValues++));
    TWI_Stop();
    _delay_us(BUS_FREE_TIME_BEFORE_STO_STA);
}

void readPLCregisters(uint8_t startRegNr, uint8_t *regValues, uint8_t
bytesToRead)
{
    TWI_Start();
    TWI_WriteByte(PLC_WRITE_ADDR);
    TWI_WriteByte(startRegNr);
    _delay_us(50);
    TWI_Start();
    TWI_WriteByte(PLC_READ_ADDR);
    while(bytesToRead--) *(regValues++) = TWI_ReadByte(bytesToRead? ACK:NACK);
    TWI_Stop();
    _delay_us(BUS_FREE_TIME_BEFORE_STO_STA);
}
```

Rejestr **RX_DATA** określa początek obszaru (od 0x4A do 0x68), który zawiera użyteczne dane przesłane w bieżącym pakiecie danych.

Rejestr: INT_STATUS (0x69)							
D7	D6	D5	D4	D3	D2	D1	D0
STA-TUS_VA-LUE_CHAN-GE	STA-TUS_BUSY	STA-TUS_TX_ACK	STA-TUS_TX_NO_RESP	STA-TUS_RX_PACKET_DROP-PED	STA-TUS_RX_DATA_AVAIL-ABLE	STA-TUS_TX_DATA_SENT	

Rejestr **INT_STATUS** jest rejestrem flag przebrań związanych z funkcjonowaniem modemu PLC. Aby skasować flagi w tym rejestrze (za wyjątkiem flag **STATUS_RX_PACKET_DROPPED** i

STATUS_RX_DATA_AVAILABLE) należy wyzerować bit **INT_CLEAR** w rejestrze **INT_ENABLE**. Z kolei, aby skasować flagi **STATUS_RX_PACKET_DROPPED**, **STATUS_RX_DATA_AVAILABLE** i **STATUS_VALUE_CHANGE** należy wyzerować bit **NEW_RX_MSG** w rejestrze **RX_MESSAGE_INFO**.

Listing 3. Ciało funkcji odpowiedzialnych za ustawienie rodzaju adresów urządzenia źródłowego i docelowego modemu PLC oraz funkcji umożliwiającej ustawienie adresu urządzenia docelowego

```
//txSAtype: TX_SA_TYPE_LOGICAL, TX_SA_TYPE_PHYSICAL
//txDAtype: TX_DA_TYPE_LOGICAL, TX_DA_TYPE_GROUP, TX_DA_TYPE_PHYSICAL

void setPLCtXAddrType(uint8_t txSAtype, uint8_t txDAtype)
{
    register uint8_t configRegValue;
    //Odczytanie bieżącej wartości rejestru TX_CONFIG i maskowanie bitów
    //odpowiedzialnych za typy adresów SA i DA
    configRegValue = readPLCregister(TX_CONFIG_REG) & ~(TX_ADDR_MASK);
    //Ustawienie nowych typów adresów będących argumentami funkcji - należy
    //używać definicji typów z pliku nagłówkowego
    writePLCregister(TX_CONFIG_REG, configRegValue|txSAtype|txDAtype);
}

void setPLCtXDA(uint8_t txDAtype, volatile uint8_t *txDA)
{
    //W zależności od typu adresu DA zapisujemy odpowiednia
    //ilość bajtów adresu DA
    switch(txDAtype)
    {
        case TX_DA_TYPE_LOGICAL:
        case TX_DA_TYPE_GROUP:
            writePLCregister(TX_DA_REG, *txDA);
            break;
        case TX_DA_TYPE_PHYSICAL:
            writePLCregisters(TX_DA_REG, txDA, 8);
            break;
    }
}
```

Listing 4. Ciało funkcji umożliwiających ustawienie adresu logicznego i adresu grupowego modemu PLC

```
void setPLCnodeLA(uint8_t logicalAddress)
{
    writePLCregister(LOGICAL_ADDR_LSB_REG, logicalAddress);
}

void setPLCnodeGA(uint8_t groupAddress)
{
    writePLCregister(GROUP_ADDR_REG, groupAddress);
}
```

Listing 5. Ciało funkcji umożliwiających ustalenie typów adresów urządzenia źródłowego i docelowego przesłanego pakietu danych PLC jak i samego adresu urządzenia źródłowego

```
//rxSAtype: RX_SA_TYPE_LOGICAL lub RX_SA_TYPE_PHYSICAL
//rxDAtype: RX_DA_TYPE_LOGICAL_PHYSICAL lub RX_DA_TYPE_GROUP

void getPLCrXAddrType(uint8_t *rxSAtype, uint8_t *rxDAtype)
{
    register uint8_t messageInfo;
    //Odczytanie bieżącej wartości rejestru RX_MESSAGE_INFO_REG
    //i ustalenie typu adresu DA i SA otrzymanej wiadomości
    messageInfo = readPLCregister(RX_MESSAGE_INFO_REG);
    //Zwraca typ DA odebranej wiadomości: RX_DA_TYPE_LOGICAL_PHYSICAL
    //lub RX_DA_TYPE_GROUP
    *rxDAtype = messageInfo & 0b01000000; //Typ zapisany w bicie 6
    //Zwraca typ SA odebranej wiadomości: RX_SA_TYPE_LOGICAL lub
    //RX_SA_TYPE_PHYSICAL
    *rxSAtype = messageInfo & 0b00100000; //Typ zapisany w bicie 5
}

void getPLCrXSA(uint8_t *rxSA)
{
    //Niezależnie od rodzaju SA otrzymanej wiadomości odczytujemy
    //zawsze 8 bajtów. Jeśli adres urządzenia, które przesało nam dane
    //jest typu LA to istotny jest tylko pierwszy bajt odczytanego adresu,
    //w przypadku PA, ważne jest całe 8 bajtów.
    readPLCregisters(RX_SA_REG, rxSA, 8);
}
```

Listing 6. Ciało funkcji umożliwiającej odczytanie przesłanego pakietu danych PLC

```
void readPLCrXPacket(uint8_t *rxCommand, uint8_t *rxData, uint8_t
*rxDataLength)
{
    register uint8_t infoRegister;
    //Odczytanie bieżącej wartości rejestru RX_MESSAGE_INFO_REG i ustalenie
    rozmiaru otrzymanej wiadomości
    *rxDataLength = readPLCregister(RX_MESSAGE_INFO_REG) & 0x1F; //0...31
    //Ustalenie rodzaju otrzymanej komendy
    *rxCommand = readPLCregister(RX_COMMAND_ID_REG);
    //Wczytanie do tablicy rxData będącej argumentem funkcji wszystkich,
    //otrzymanych danych
    readPLCregisters(RX_DATA_REG, rxData, *rxDataLength);
    //Aby skasować flagi STATUS_VALUE_CHANGE, STATUS_RX_PACKET_DROPPED
    //i STATUS_RX_DATA_AVAILABLE w rejestrze INTERRUPT_STATUS_REG musimy
    //wyzerosować flagę NEW_PACKET_RECEIVED w rejestrze RX_MESSAGE_INFO_REG
    //(datasheet)
    infoRegister = readPLCregister(RX_MESSAGE_INFO_REG);
    writePLCregister(RX_MESSAGE_INFO_REG, infoRegister & ~NEW_PACKET_
RECEIVED);
}
```

Poszczególne bity rejestru **INT_STATUS** mają następujące znaczenie funkcjonalne:

- Bit **STATUS_VALUE_CHANGE** ustawiany jest za każdym razem, gdy odebrano nowe dane.
- Bit **STATUS_BUSY** ustawiany jest za każdym razem, gdy nie ma możliwości przesłania pakietu danych z uwagi na zajętość magistrali danych PLC (timeout 1,1...3 s w zależności od ustawień detektora BIU).
- Bit **STATUS_TX_NO_ACK** ustawiany jest za każdym razem, gdy w czasie 500 ms od przesłania pakietu danych nie odebrano potwierdzenia odebrania tegoż pakietu danych po stronie urządzenia docelowego (w przypadku konfiguracji sieci obejmującej potwierdzanie pakietów danych).
- Bit **STATUS_TX_NO_RESP** ustawiany jest w przypadku wybranych rozkazów stosu komunikacyjnego, jak na przykład rozkazów, które wymuszają przesłanie przez urządzenie docelowe danych (żądanie przesłania danych wysłane przez urządzenie źródłowe, np. rozkaz SEND_REMOTE_DATA) lub rozkazu zdalnej zmiany adresu logicznego urządzenia docelowego, a dane takie nie zostaną przez to urządzenie przesłane w czasie 1.5 s od przesłania rozkazu.
- Bit **STATUS_RX_PACKET_DROPPED** ustawiany jest w przypadku nadpisania bufora odbiorczego nowymi danymi, gdy dane poprzednie nie zostały jeszcze odczytane.
- Bit **STATUS_RX_DATA_AVAILABLE** ustawiany jest za każdym razem, gdy modem PLC odbierze nowy pakiet danych.
- Bit **STATUS_TX_DATA_SENT** ustawiany jest wyłącznie wtedy, gdy bieżąca transmisja danych zakończy się powodzeniem.

Rejestr: LOCAL_PA (0x6A)							
D7	D6	D5	D4	D3	D2	D1	D0
PHYSICAL_ADDRESS (MSB)							

Rejestr **LOCAL_PA** określa początek obszaru (od 0x6A do 0x71), który zawiera unikalny, 64-bitowy, fizyczny adres modemu PLC.

Funkcje sterujące pracą modemu CY8CPLC10

Przebrnęliśmy przez szczegółowy opis rejestrów konfiguracyjnych układu CY8CPLC10, dzięki czemu dysponujemy już niezbędną wiedzą by przejść do konfiguracji naszego systemu. Zanim jednak, opiszę szczegóły implementacyjne tego, konkretnego rozwiązania, warto pokazać, choćby po krótko, funkcje, za pomocą których jest możliwe sterowanie pracą modemu. Na początek opiszę podstawowe funkcje, dzięki którym jest możliwa komunikacja przy udziale interfejsu TWI mikrokontrolera, przy czym należy

zaznaczyć, iż są to najprostsze implementacje poszczególnych funkcjonalności niewyposażone np. w mechanizm obsługi błędów, który moim zdaniem, nie jest elementem niezbędnym w tak prostych i niewielkich systemach mikroprocesorowych.

Ciała funkcji odpowiedzialnych za obsługę interfejsu TWI pokazano na **listingu 1**.

Myślę, że nie wymagają one dodatkowego komentarza, gdyż ich nazwy są na tyle wymowne, iż nie pozostawiają wątpliwości, co do realizowanych przez nie funkcjonalności. Przejdźmy, zatem do podstawowych funkcji umożliwiających zapis i odczyt pojedynczych rejestrów modemu PLC, jak i całych grup rejestrów, które to pokazano na **listingu 2**. Oczywiście, można by było zastosować jeden rodzaj funkcji dla odczytu/zapisu zarówno pojedynczych rejestrów jak i całych ich grup, jednak takie rozwiązanie powoduje optymalizację kodu wynikowego zarówno pod względem szybkości wykonania jak i wielkości generowanego kodu.

Pora na przedstawienie kluczowych funkcji umożliwiających konfigurację podstawowych parametrów naszego modemu PLC. Na początek dwie funkcje, jedna przeznaczona do ustawienia typów adresów urządzenia źródłowego i docelowego pakietu danych PLC, zaś druga do ustawienia samego adresu urządzenia docelowego. Funkcje te przedstawiono na **listingu 3**.

Kolejne, dwie, bardzo krótkie funkcje narzędziowe, które pokazano na **listingu 4** służą do ustawienia adresu logicznego modemu PLC jak i adresu grupy, do której przyporządkujemy nasze urządzenie.

Pora na funkcje użyteczne w przypadku odbioru pakietu danych PLC. Pierwsza z nich, przedstawiona na **listingu 5** służy do ustalenia typów adresów urządzenia źródłowego i docelowego, które zostały przesłane w odebranych pakiecie danych PLC, zaś druga z nich służy do odczytania adresu urządzenia źródłowego.

Do „kompletu” brakuje nam funkcji, która umożliwi odczytanie przesłanego pakietu danych, towarzyszącego im rozkazu oraz określenie długości tegoż pakietu danych. Ciało tej funkcji pokazano na **listingu 6**.

Dodatkowo, w przypadku odbioru pakietu danych PLC, przydatna może być funkcja, która pozwala nam na ustalenie zdarzenia, jakie spowodowało wyzwolenie przerwania odpowiedzialnego za obsługę pakietów danych. Ciało tej funkcji pokazano na **listingu 7**.

No i już zupełnie na koniec, funkcja, która pozwala na pełne skonfigurowanie modemu PLC. Ciało tej funkcji pokazano na **listingu 8**.

Dla porządku przedstawiam również listing pliku nagłówkowego modemu PLC, który jak zwykle konstruuję w taki sposób, by bez potrzeby zaglądania do dokumentacji układu, zorientować się w dostępnej liście

Listing 7. Ciało funkcji umożliwiającej ustalenie zdarzenia, jakie spowodowało wyzwolenie przerwania odpowiedzialnego za obsługę pakietów danych

```
uint8_t readPLCIntRegister(void)
{
    register uint8_t intStatusReg, intEnableReg;
    //Odczyt rejestru statusu przerw PLC by ustalić rodzaj zdarzenia
    //jaki zostało zgłoszone
    intStatusReg = readPLCRegister(INTERRUPT_STATUS_REG);
    //Po odczycie powyższego rejestru należy wyzerować bit INT_CLEAR
    //w rejestrze INTERRUPT_ENABLE_REG (datasheet, page 7.)
    intEnableReg = readPLCRegister(INTERRUPT_ENABLE_REG) & ~INT_CLEAR;
    writePLCRegister(INTERRUPT_ENABLE_REG, intEnableReg);
    return intStatusReg;
}
```

Listing 8. Ciało funkcji umożliwiającej kompletną konfigurację modemu PLC

```
void initPLCdevice(uint8_t nodeLA)
{
    //Uruchomienie i podstawowa konfiguracja modemu PLC
    writePLCRegister(PLC_MODE_REG, TX_ENABLE|RX_ENABLE|RX_OVERRIDE|ENABLE_BIU|CHECK_DA|VERIFY_PACKET_CRC8);
    //Ustawienie poziomu sygnału dla mechanizmu CSMA (Carrier Sense Multimaster Access) zapewniającego wielodostęp do medium transmisyjnego
    writePLCRegister(THRESHOLD_NOISE_REG, BIU_THRESHOLD_87DBUV);
    //Konfiguracja parametrów transmisji
    writePLCRegister(MODEM_CONFIG_REG, MODEM_BPS_2400|MODEM_FSK_BAND_DEV_3KHZ);
    //Uruchomienie przerw dla wybranych zdarzeń (aktywny poziom niski //na wyprowadzeniu HOST_INT)
    writePLCRegister(INTERRUPT_ENABLE_REG, INT_POLARITY_LOW|INT_UNABLE_TO_TX|INT_TX_NO_ACK|INT_TX_NO_RESP|INT_RX_DATA_AVAILABLE|INT_TX_DATA_SENT);
    //Ustawienie trybu potwierdzania pakietów danych oraz liczby prób
    //transmisji = 5 (domyślne logiczne typy adresów SA i DA)
    writePLCRegister(TX_CONFIG_REG, TX_SERVICE_ACKNOWLEDGED|0x05);
    //Ustawienie wzmocnienia dla modułu nadajnika PLC
    writePLCRegister(TX_GAIN_REG, TX_GAIN_LEVEL_1550MV);
    //Ustawienie czułości dla modułu odbiornika PLC
    writePLCRegister(RX_GAIN_REG, RX_GAIN_LEVEL_250UV);
    setPLCNodeLA(nodeLA); //Ustawienie numeru LA modemu PLC
    //Ustawienie adresu Grupowego modemu PLC
    setPLCNodeGA(MASTER_GROUP_ADDR);
    //Konfiguracja i aktywacja przerwania INT0 odpowiedzialnego za //obsługę odbioru wiadomości PLC
    HOST_INT_PORT |= (1<<HOST_INT_NR); //Podciągnięcie INT0 do VCC
    EICRA |= (1<<ISC01); //Zbocze opadające na INT0 wyzwala przerwanie
    EIMSK |= (1<<INT0); //Odblokowanie przerwania INT0
}
```

możliwości. Zawartość tego pliku nagłówkowego pokazano na **listingu 9**.

Założenia projektowe, funkcjonalność systemu

Na tą chwilę dysponujemy już kompletną wiedzą, która daje nam możliwość obsługi modemu PLC w związku, z czym pora przejść do konkretnej implementacji systemu iControl. Podstawowe założenia systemu iControl przedstawiają się następująco:

W ramach stawianej sieci systemu iControl (jednego mieszkania lub domu) zakłada się współistnienie maksymalnie 64 modułów wykonawczych (typu Slave) o adresach logicznych z zakresu 0x00..0x3F oraz maksymalnie 16 modułów sterujących (typu Master), wyposażonych w interfejs użytkownika z wyświetlaczem TFT, o adresach logicznych z zakresu 0xF0÷0xFF.

Adresy logiczne modułów wykonawczych nadawane są automatycznie przez moduły sterujące podczas konfigurowania sieci, zaś adresy logiczne modułów sterujących nadawane są przez użytkownika przy pomocy interfejsu użytkownika GUI.

Każdy moduł sterujący może zapamiętać i zaadresować maksymalnie 64 moduły wykonawcze (staje się wtedy tzw. „rodzicem” modułu wykonawczego).

Wszystkie moduły wykonawcze zapamiętane przez dany moduł sterujący mogą

zostać połączone w maksymalnie 8 grup, dowolnie podczas konfiguracji sieci, reprezentujących pomieszczenia, nad którymi moduł ten ma kontrolę (np. pokoje).

Każdy moduł wykonawczy, którego dany moduł sterujący jest „rodzicem” może być współdzielony innemu modułowi sterującemu, dzięki czemu w jednym czasie kilka modułów sterujących może mieć kontrolę nad jednym i tym samym modułem wykonawczym (daje to możliwość sterowania np. oświetleniem w kilku pokojach z jednego modułu sterującego umieszczonego w innym miejscu), przy czym w takim wypadku, moduły sterujące wymieniają się na bieżąco parametrami nadzorowanego modułu sterującego by stan tego modułu był znany każdemu z nich.

W ramach graficznego interfejsu użytkownika modułu sterującego, każdy moduł wykonawczy identyfikowany jest przez unikalną nazwę nadaną przez użytkownika systemu. Podobnie, każda z 8 możliwych grup (pomieszczeń), w które mogą być zebrane moduły wykonawcze podlega konfiguracji (aktywacji/dezaktywacji, nadaniu nazwy) w ramach wspomnianego interfejsu użytkownika.

Przewidziano 5 rodzajów modułów wykonawczych, automatycznie rozpoznawanych przez moduły sterujące, co determinuje wygląd tego modułu w ramach interfejsu użytkownika oraz sposób sterowania:

Listing 9. Plik nagłówkowy modemu PLC

```

#define I2C_ADDR_PIN_FLOATING 1
#define HOST_INT_PORT PORTD
#define HOST_INT_PIN PIND
#define HOST_INT_NR PD2
#define BUS_FREE_TIME_BEFORE_STO_STA 550

#if I2C_ADDR_PIN_FLOATING
#define PLC_READ_ADDR 0x03
#define PLC_WRITE_ADDR 0x02
#else
#define PLC_READ_ADDR 0xF5
#define PLC_WRITE_ADDR 0xF4
#endif

// Rejestry nadajnika modemu PLC
#define INTERRUPT_ENABLE_REG 0x00
#define INT_CLEAR (1<<7)
#define INT_POLARITY_HIGH (0<<6)
#define INT_POLARITY_LOW (1<<6)
#define INT_UNABLE_TO_TX (1<<5)
#define INT_TX_NO_ACK (1<<4)
#define INT_TX_NO_RESP (1<<3)
#define INT_RX_PACKET_DROPPED (1<<2)
#define INT_RX_DATA_AVAILABLE (1<<1)
#define INT_TX_DATA_SENT (1<<0)

#define LOGICAL_ADDR_LSB_REG 0x01
#define LOGICAL_ADDR_MSB_REG 0x02
#define GROUP_ADDR_REG 0x03
#define LOCAL_GROUP_HOT_REG 0x04

#define PLC_MODE_REG 0x05
#define TX_ENABLE (1<<7)
#define RX_ENABLE (1<<6)
#define LOCK_CONFIGURATION (1<<5)
#define DISABLE_BIU (1<<4)
#define ENABLE_BIU (0<<4)
#define RX_OVERRIDE (1<<3)
#define SET_EXT_ADDRESS (1<<2)
#define ACCEPT_ALL_MSGS (1<<1)
#define CHECK_DA (0<<1)
#define DISCARD_PACKET_CRC8 (1<<0)
#define VERIFY_PACKET_CRC8 (0<<0)

#define TX_MESSAGE_LENGTH_REG 0x06
#define SEND_MESSAGE (1<<7)
#define PAYLOAD_LENGTH_MASK 0x1F

#define TX_CONFIG_REG 0x07
#define TX_ADDR_MASK 0b11100000
#define TX_SA_TYPE_LOGICAL (0<<7)
#define TX_SA_TYPE_PHYSICAL (1<<7)
#define TX_DA_TYPE_LOGICAL (0<<5)
#define TX_DA_TYPE_GROUP (1<<5)
#define TX_DA_TYPE_PHYSICAL (2<<5)
#define TX_SERVICE_ACKNOWLEDGED (1<<4)
#define TX_SERVICE_TYPE_NO_ACKNOWLEDGED (0<<4)

#define TX_DA_REG 0x08
#define TX_COMMAND_ID_REG 0x10
#define TX_DATA_REG 0x11

//Rejestry konfiguracyjne modemu PLC
#define THRESHOLD_NOISE_REG 0x30
#define AUTO_BIU_DISABLED (0<<6)
#define AUTO_BIU_ENABLED (1<<6)
#define BIU_THRESHOLD_70DBUV 0x00
#define BIU_THRESHOLD_75DBUV 0x01
#define BIU_THRESHOLD_80DBUV 0x02

#define BIU_THRESHOLD_87DBUV 0x03
#define BIU_THRESHOLD_90DBUV 0x04
#define BIU_THRESHOLD_93DBUV 0x05
#define BIU_THRESHOLD_96DBUV 0x06
#define BIU_THRESHOLD_99DBUV 0x07

#define MODEM_CONFIG_REG 0x31
#define TX_DELAY_7MS (0<<5)
#define TX_DELAY_13MS (1<<5)
#define TX_DELAY_19MS (2<<5)
#define TX_DELAY_25MS (3<<5)
#define MODEM_FSK_BAND_DEV_3KHZ (1<<3) //133.3:130.4
kHz
#define MODEM_FSK_BAND_DEV_1_5KHZ (0<<3)
//133.3:131.8 kHz
#define MODEM_BPS_600 0x00
#define MODEM_BPS_1200 0x01
#define MODEM_BPS_1800 0x02
#define MODEM_BPS_2400 0x03

#define TX_GAIN_REG 0x32
#define TX_GAIN_LEVEL_55MV 0x00
#define TX_GAIN_LEVEL_75MV 0x01
#define TX_GAIN_LEVEL_100MV 0x02
#define TX_GAIN_LEVEL_125MV 0x03
#define TX_GAIN_LEVEL_180MV 0x04
#define TX_GAIN_LEVEL_250MV 0x05
#define TX_GAIN_LEVEL_360MV 0x06
#define TX_GAIN_LEVEL_480MV 0x07
#define TX_GAIN_LEVEL_660MV 0x08
#define TX_GAIN_LEVEL_900MV 0x09
#define TX_GAIN_LEVEL_1250MV 0x0A
#define TX_GAIN_LEVEL_1550MV 0x0B //Default
#define TX_GAIN_LEVEL_2250MV 0x0C
#define TX_GAIN_LEVEL_3000MV 0x0D
#define TX_GAIN_LEVEL_3500MV 0x0E

#define RX_GAIN_REG 0x33
#define RX_GAIN_LEVEL_5000UV 0x00 //Default
#define RX_GAIN_LEVEL_3500UV 0x01
#define RX_GAIN_LEVEL_2500UV 0x02
#define RX_GAIN_LEVEL_1250UV 0x03
#define RX_GAIN_LEVEL_600UV 0x04
#define RX_GAIN_LEVEL_350UV 0x05
#define RX_GAIN_LEVEL_250UV 0x06
#define RX_GAIN_LEVEL_125UV 0x07

//Rejestry odbiornika modemu PLC
#define RX_MESSAGE_INFO_REG 0x40
#define NEW_PACKET_RECEIVED (1<<7)
#define NO_PACKET_RECEIVED (0<<7)
#define RX_DA_TYPE_LOGICAL_PHYSICAL (0<<6)
#define RX_DA_TYPE_GROUP (1<<6)
#define RX_SA_TYPE_LOGICAL (0<<5)
#define RX_SA_TYPE_PHYSICAL (1<<5)
#define RX_MESSAGE_LENGTH_BIT_MASK 0b11111

#define RX_SA_REG 0x41
#define RX_COMMAND_ID_REG 0x49
#define RX_DATA_REG 0x4A

#define INTERRUPT_STATUS_REG 0x69
#define STATUS_VALUE_CHANGE (1<<7)
#define STATUS_BUSY (1<<5) //BIU timeout
#define STATUS_TX_NO_ACK (1<<4)
#define STATUS_TX_NO_RESP (1<<3)
#define STATUS_RX_PACKET_DROPPED (1<<2)
#define STATUS_RX_DATA_AVAILABLE (1<<1)
#define STATUS_TX_DATA_SENT (1<<0)

```

wyłącznik 1-biegunowy, wyłącznik 2-biegunowy, ściemniacz, sensor temperatury, sterownik oświetlenia RGB LED.

System iControl sygnalizuje dołączenie nowych, jeszcze nieskonfigurowanych, modułów wykonawczych w sieci (zapamiętuje do 16 nowych modułów wykonawczych) jak również wystąpienie błędów transmisji (sterowania modułem wykonawczym).

System iControl umożliwia usuwanie modułów wykonawczych z sieci, a co za tym idzie – rekonfigurowanie sieci.

Zakłada się następującą kolejność czynności w ramach konfiguracji sieci: podłączone i skonfigurowane (nadany adres logiczny) wszystkie, planowane moduły sterujące a w każdym z nich zdefiniowana przynajmniej jedna z grup (pomieszczeń). Następnie, dołączamy i konfigurujemy każdy, kolejny moduł sterujący, przy czym dla uproszczenia procedury konfiguracji z punktu widzenia użytkownika, zakłada się, iż kolejny moduł wykonawczy zostanie dołączony do sieci dopiero po zakończeniu konfigurowania

modułu ostatnio dołączonego (pozwala to łatwiej zorientować się, który moduł fizycznie jest aktualnie konfigurowany).

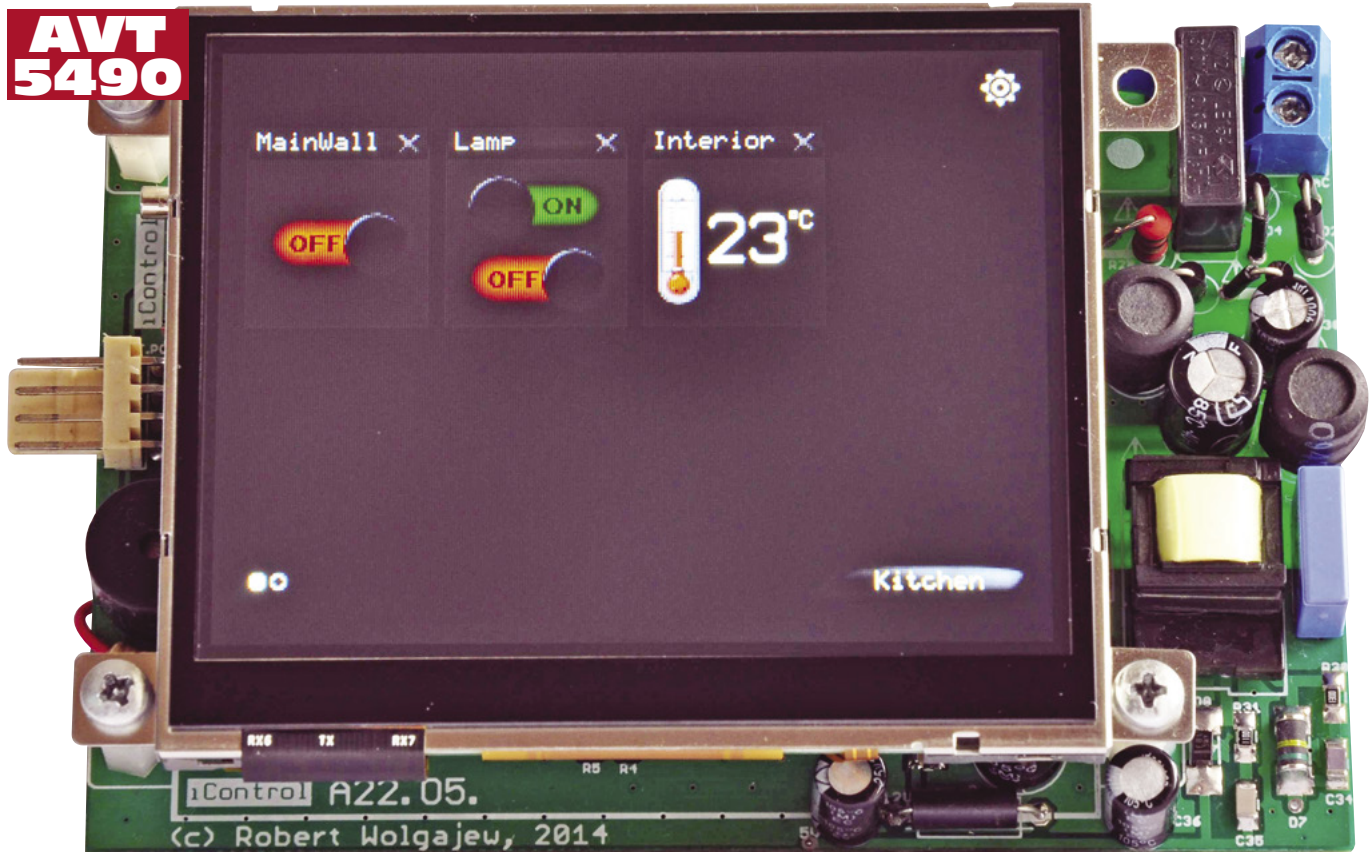
Znając logiczną strukturę systemu iControl pora na przedstawienie konstrukcji pakietów danych (przesyłanych rozkazów i towarzyszących im danych) w ramach konfiguracji sieci jak i jej „normalnego” funkcjonowania. Opiszę je i zaprezentuję schematy i projekty płytek drukowanych w drugiej części artykułu.

Robert Wołgajew, EP

Podziękowania

Autor składa serdeczne podziękowania pani Beacie Stosik z firmy STOFL za dostarczenie wielu darmowych próbek układów CY8CPLC10, panu Sławomirowi Szweda z firmy Unisystem za dostarczenie darmowych modułów wyświetlaczy TFT oraz panu Dariuszowi Kowalczyk za cenne uwagi przekazywane w trakcie projektowania urządzenia.

Author would like to thank Mr Prem Sai V from Cypress Semiconductor customer support team for his huge and inestimable support and commitment while developing the device!



iControl

System automatyki domowej (2)



Film demonstrujący działanie systemu iControl:
<http://goo.gl/Q99yjj>

W swojej praktyce konstruktora-programisty kilkakrotnie podejmowałem wyzwanie skonstruowania prostego systemu sterowania i kontroli typu „inteligentny dom”, wykorzystując w tym celu różne interfejsy w roli medium umożliwiającego współpracę specjalizowanych modułów. Jednym z przykładów takiego dość rozbudowanego i nowoczesnego projektu może być system intelli-Dom opublikowany na łamach Elektroniki Praktycznej 10...11/2011, który wykorzystywał specjalizowane i bardzo dobrze wyposażone moduły modemów ZigBee. System ten, mimo nie dawał jednak pełnej elastyczności w zakresie konfigurowania sieci modułów wykonawczych, gdyż korzystał z predefiniowanej funkcjonalności tychże modułów nazywanych tam „modułami pokojowymi”, a poza tym używał transmisji bezprzewodowej o ograniczonym zasięgu, realizowanej w paśmie 2,4 GHz, opartej na standardzie IEEE 802.15.4. Czas więc na projekt zaawansowany, w pełni konfigurowalny i pozbawiony poprzednich ograniczeń a dodatkowo wyposażony w ultranowoczesny interfejs użytkownika.

Rekomendacje: projekt przyda się jako baza dla rozbudowanego systemu zarządzania inteligentnym budynkiem.

W ofercie AVT* AVT-5490 A

Podstawowe informacje:

- Maksymalnie 64 modułów wykonawczych (typu Slave) w ramach jednej sieci systemu iControl.
- Maksymalnie 16 modułów sterujących (typu Master) wyposażonych w interfejs użytkownika z wyświetlaczem TFT.
- Adresy logiczne modułów wykonawczych nadawane są automatycznie przez moduły sterujące podczas konfiguracji sieci, zaś adresy logiczne modułów sterujących nadawane są przez użytkownika za pomocą interfejsu użytkownika GUI.
- Każdy moduł sterujący może zapamiętać i zaadresować 64 moduły wykonawcze.
- Wszystkie moduły wykonawcze zapamiętane przez dany moduł sterujący mogą zostać połączone w maksymalnie 8 grup, dowolnie podczas konfiguracji sieci, reprezentujących pomieszczenia, nad którymi moduł ten ma kontrolę (np. pokoje).
- Kilka modułów sterujących może mieć kontrolę nad jednym modułem wykonawczym.
- W ramach graficznego interfejsu użytkownika modułu sterującego każdy moduł wykonawczy jest identyfikowany przez unikalną nazwę.
- Każda z 8 możliwych grup, w które mogą być łączone moduły wykonawcze może mieć nadaną nazwę, aktywowana lub wyłączona.
- Przewidziano 5 rodzajów modułów wykonawczych: wyłącznik 1-biegunowy, wyłącznik 2-biegunowy, ściemniacz, sensor temperatury, sterownik oświetlenia RGB LED.
- System iControl sygnalizuje dołączenie nowych, jeszcze nieskonfigurowanych modułów wykonawczych oraz wystąpienie błędów transmisji.
- System iControl umożliwia usuwanie modułów wykonawczych z sieci, a co za tym idzie – rekonfigurację sieci.

Dodatkowe materiały na FTP:

<ftp://ep.com.pl>, user: 54721, pass: qn2jbb4t

- wzory płytek PCB

Projekty pokrewne na FTP:

(wymienione artykuły są w całości dostępne na FTP)

AVT-5490 iControl. System automatyki domowej (EP 1/2015)

* Uwaga:

Zestawy AVT mogą występować w następujących wersjach:

AVT xxxx UK to zaprogramowany układ. Tylko i wyłącznie. Bez elementów dodatkowych.

AVT xxxx A płytka drukowana PCB (lub płytki drukowane, jeśli w opisie wyraźnie zaznaczono), bez elementów dodatkowych.

AVT xxxx A+ płytka drukowana i zaprogramowany układ (czyli połączenie wersji A i wersji UK) bez elementów dodatkowych.

AVT xxxx B płytka drukowana (lub płytki) oraz komplet elementów wymieniony w załączniku pdf

AVT xxxx C to nic innego jak zmontowany zestaw B, czyli elementy wlotowane w PCB. Należy mieć na uwadze, że o ile nie zaznaczono wyraźnie w opisie, zestaw ten nie ma obudowy ani elementów dodatkowych, które nie zostały wymienione w załączniku pdf

AVT xxxx CD oprogramowanie (nieczęsto spotykana wersja, lecz jeśli występuje, to niezbędne oprogramowanie można ściągnąć, klikając w link umieszczony w opisie kitu)

Nie każdy zestaw AVT występuje we wszystkich wersjach! Każda wersja ma załączony ten sam plik pdf! Podczas składania zamówienia upewnij się, którą wersję zamawiasz! (UK, A, A+, B lub C). <http://sklep.avt.pl>

Każdy, nowy, nieskonfigurowany jeszcze moduł wykonawczy co 4 sekundy wysyła pakiet danych rozgłaszając w sieci swój adres fizyczny (*Slave Status = NODE_AVAILABLE*). Jednocześnie sygnalizuje ten stan za pomocą dedykowanej diody LED (oznaczonej *STATUS*), która miga co każdym razem, gdy przesyłany jest taki pakiet danych, czyli co około 4 sekundy.

Adres SA	Adres DA	Rozkaz	Dane
Typ adresu SA	Typ adresu DA	Rozkaz	NEW_NODE
Wartość adresu SA	Wartość adresu DA	Wartość	0x30
Wartość adresu SA	Wartość adresu DA	Wartość	Slave PA, typ Slave, status Slave

*Każdy moduł sterujący ma ustawiony adres GA = 0x01 oraz przydzielony przy pomocy GUI adres LA (z zakresu 0xF0...0xFF)

Pakiet taki odbierany jest przez każdy moduł sterujący dołączony do sieci powodując dodanie nowego modułu wykonawczego do listy modułów przeznaczonych do skonfigurowania (moduł sterujący sprawdza za każdym razem, czy nie dodano już takiego adresu PA do listy) oraz odpowiednią sygnalizację w ramach interfejsu użytkownika GUI. Taki moduł wykonawczy, może następnie zostać wybrany do konfiguracji (wyłącznie z poziomu jednego modułu sterującego), co skutkuje zmianą jego stanu (*Slave Status = NODE_SELECTED*), zaprzestaniem rozgłaszania adresu PA oraz odpowiednią sygnalizacją (dedykowana dioda LED na module wykonawczym zaczyna szybko migać). Aby go wybrać moduł sterujący wysyła następujący pakiet:

Adres SA	Adres DA	Rozkaz	Dane
Typ adresu SA	Typ adresu DA	Rozkaz	SELECT NODE
Wartość adresu SA	Wartość adresu DA	Wartość	0x31
Wartość adresu SA	Wartość adresu DA	Wartość	-

Jeśli użytkownik wybierze moduł wykonawczy z poziomu obsługiwanego modułu sterującego to sterownik ten wysyła do pozostałych

modułów sterujących wiadomość by usunęły zapamiętany wcześniej moduł wykonawczy z listy swoich, jeszcze nieskonfigurowanych modułów, gdyż właśnie został on wybrany do konfiguracji. W tym celu sterownik ten przesyła następujący pakiet danych:

Adres SA	Adres DA	Rozkaz	Dane
Typ adresu SA	Typ adresu DA	Rozkaz	REMOVE_NEW_NODE
Wartość adresu SA	Wartość adresu DA	Wartość	0x38
Wartość adresu SA	Wartość adresu DA	Wartość	Slave PA

Po wyborze modułu wykonawczego, użytkownik może albo porzucić jego konfigurację i moduł ten znowu zacznie rozsyłać swój adres PA (i wróci do statusu *Slave Status = NODE_AVAILABLE*) lub też może powiązać moduł wykonawczy z modułem sterującym stając się jego „rodzicem”. Aby porzucić wybór modułu wykonawczego moduł sterujący wysyła następujący pakiet:

Adres SA	Adres DA	Rozkaz	Dane
Typ adresu SA	Typ adresu DA	Rozkaz	UNSELECT NODE
Wartość adresu SA	Wartość adresu DA	Wartość	0x32
Wartość adresu SA	Wartość adresu DA	Wartość	-

Aby powiązać moduł wykonawczy ze sobą, moduł sterujący wysyła następujący pakiet danych, który de facto przypisuje mu unikalny adres logiczny korzystając z predefiniowanego rozkazu stosu komunikacyjnego (*Slave Status = NODE_ACTIVE*):

Adres SA	Adres DA	Rozkaz	Dane
Typ adresu SA	Typ adresu DA	Rozkaz	SET_LA_ADDR
Wartość adresu SA	Wartość adresu DA	Wartość	0x04
Wartość adresu SA	Wartość adresu DA	Wartość	Unikalny LA

Jednocześnie bieżący moduł sterujący wysyła wiadomość do innych tego typu modułów by zapisały sobie ostatnio przydzielony adres logiczny (by nie mogły go użyć w procesie nadawania nowego LA):

Adres SA	Adres DA	Rozkaz	Dane
Typ adresu SA	Typ adresu DA	Rozkaz	SAVE_LAST_LA
Wartość adresu SA	Wartość adresu DA	Wartość	0x33
Wartość adresu SA	Wartość adresu DA	Wartość	Ostatnio nadany LA

Moduł sterujący może również w każdej chwili (przy użyciu interfejsu użytkownika GUI) porzucić powiązanie z modułem wykonawczym wysyłając następujący pakiet (moduł wykonawczy przejdzie wtedy do stanu *Slave Status = NODE_AVAILABLE*):

Adres SA	Adres DA	Rozkaz	Dane
Typ adresu SA	Typ adresu DA	Rozkaz	UNBIND_NODE
Wartość adresu SA	Wartość adresu DA	Wartość	0x34
Wartość adresu SA	Wartość adresu DA	Wartość	-

W takim przypadku, jeśli bieżący moduł sterujący udostępnił wcześniej swój moduł wykonawczy innym sterownikom to musi ich poinformować, że utracił z nim powiązanie (nie jest już jego „rodzicem”) wysyłając następujący pakiet:

Adres SA	Adres DA	Rozkaz	Dane
Typ adresu SA	Typ adresu DA	Rozkaz	UNBIND_NODE
Wartość adresu SA	Wartość adresu DA	Wartość	0x34
Wartość adresu SA	Wartość adresu DA	Wartość	Slave LA

Moduł sterujący może również żądać od swojego modułu wykonawczego przesłania bieżącego statusu wysyłając do niego następujący pakiet danych:

Adres SA		Adres DA		Rozkaz		Dane	
Typ adresu SA	LA	Typ adresu DA	LA	Rozkaz	GET_NODE_STATUS	Długość	0
Wartość adresu SA	Master LA	Wartość adresu DA	Slave LA	Wartość	0x35	Wartość	-

Moduł sterujący może również, co oczywiste, zmieniać bieżący status modułu wykonawczego (sterować nim) przesyłając następujący pakiet danych:

Adres SA		Adres DA		Rozkaz		Dane	
Typ adresu SA	LA	Typ adresu DA	LA	Rozkaz	SET_NODE_STATUS	Długość	1
Wartość adresu SA	Master LA	Wartość adresu DA	Slave LA	Wartość	0x39	Wartość	nowy Status

Moduł wykonawczy wysła swój status przesyłając następujący pakiet danych:

Adres SA		Adres DA		Rozkaz		Dane	
Typ adresu SA	LA	Typ adresu DA	LA	Rozkaz	SEND_NODE_STATUS	Długość	2
Wartość adresu SA	Slave LA	Wartość adresu DA	Master LA	Wartość	0x36	Wartość	Slave LA, bieżący Status

Taki pakiet danych odbierany jest przez moduł sterujący będący „rodzicem” modułu wykonawczego powodując odpowiednią zmianę stanu interfejsu użytkownika GUI. Gdy moduł sterujący współdziałał wcześniej ten moduł wykonawczy z innym modulem sterującym, to musi go także poinformować o zmianie stanu tego modułu (gdyż moduł wykonawczy poinformował tylko swojego „rodzica”) wysyłając następujący pakiet danych (wysłaniu tego pakietu danych towarzyszy krótkie zaświecenie diody sygnalizacyjnej STATUS umieszczonej na płycie modułu wykonawczego):

Adres SA		Adres DA		Rozkaz		Dane	
Typ adresu SA	LA	Typ adresu DA	GA	Rozkaz	SEND_NODE_STATUS	Długość	2
Wartość adresu SA	Master LA	Wartość adresu DA	0x01	Wartość	0x36	Wartość	Slave LA, bieżący Status

Moduł sterujący, na żądanie użytkownika, może również, co napisano wcześniej, podzielić uprawnienia do sterowania powiązanych modułem wykonawczym wysyłając do innych modułów sterujących następujący pakiet danych:

Adres SA		Adres DA		Rozkaz		Dane	
Typ adresu SA	LA	Typ adresu DA	GA	Rozkaz	SHARE_NODE	Długość	3
Wartość adresu SA	Master LA	Wartość adresu DA	0x01	Wartość	0x37	Wartość	Slave LA, typ modułu, status modułu

Podobnie jak w przypadku rozkazu **NEW_NODE** pakiet taki odbierany jest przez każdy, podłączony do sieci, moduł sterujący powodując dodanie go do listy nowych modułów wykonawczych przeznaczonych do skonfigurowania (moduł sterujący sprawdza za każdym razem, czy nie dodano już takiego adresu LA do listy) oraz odpowiednią sygnalizację w ramach interfejsu użytkownika GUI. Taki moduł wykonawczy, może następnie zostać wybrany do konfiguracji (wyłącznie z poziomu jednego modułu sterującego), lecz w odróżnieniu od scenariusza dla rozkazu **NEW_NODE**, do modułu tego nie są wysyłane już żadne, inne rozkazy zmieniającego jego stan, czy nadające mu nowy adres LA, bo przecież jest to już moduł skonfigurowany wcześniej przez inny sterownik. W takim wypadku bieżący moduł sterujący przyporządkowuje wyłącznie tenże moduł wykonawczy do odpowiedniej grupy w ramach interfejsu użytkownika (do wybranego pokoju) i wysła do innych modułów sterujących rozkaz, który

nakazuje im usunąć wspomniany moduł wykonawczy z listy nowych modułów wykonawczych, gdyż właśnie został on skonfigurowany. W ramach tej czynności wysyłany jest następujący pakiet danych:

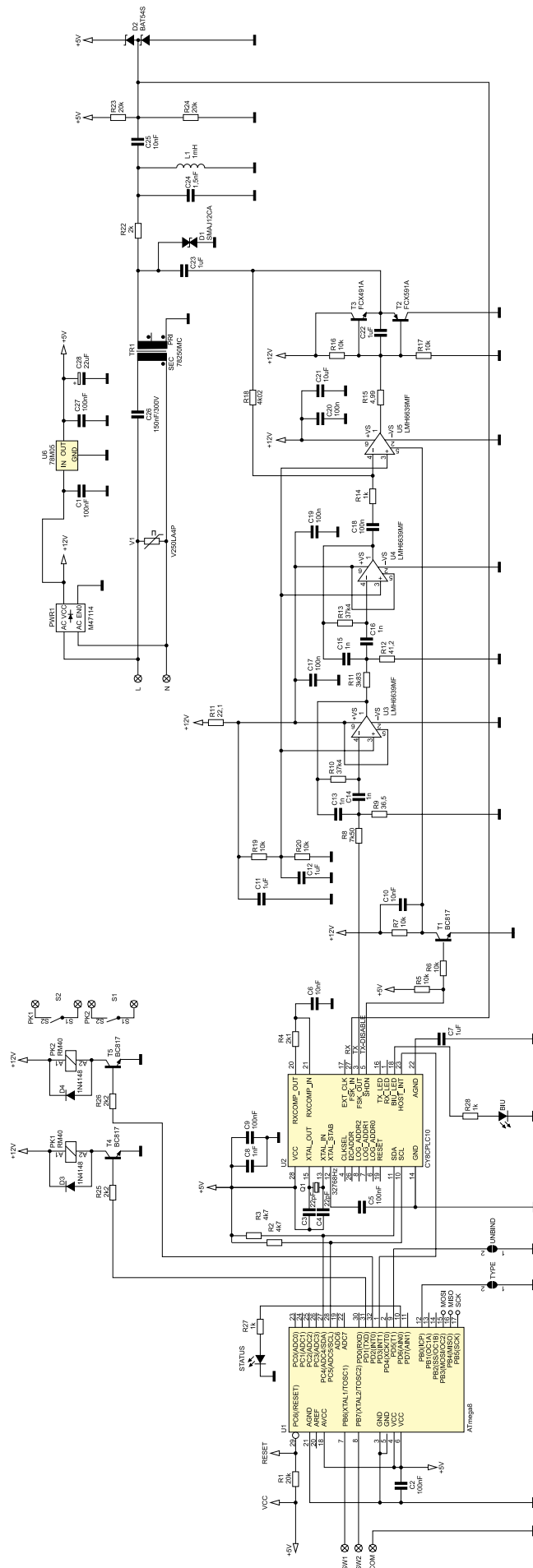
Adres SA		Adres DA		Rozkaz		Dane	
Typ adresu SA	LA	Typ adresu DA	GA	Rozkaz	REMOVE_SHARED_NODE	Długość	1
Wartość adresu SA	Master LA	Wartość adresu DA	0x01	Wartość	0x3A	Wartość	Slave LA

W systemie iControl skorzystaliśmy wyłącznie z jednego, predefiniowanego przez stos komunikacyjny rozkazu sterującego, przy pomocy którego możemy zdalnie nadać adres logiczny konfigurowanemu modułowi wykonawczemu (rozkaz **SET_LOGICAL_ADDRESS**, 0x04). Wszystkie, inne rozkazy sterujące należą do grupy rozkazów definiowanych dowolnie przez użytkownika i zostały zaimplementowane w celu zapewnienia logicznej konstrukcji całego systemu. Co oczywiste, każdy moduł podłączony do tak skonstruowanej sieci, przed wysłaniem pakietu danych, sprawdza każdorazowo zajętość magistrali danych PLC (wykorzystując wbudowany detektor BIU), przez co wysłanie takiego pakietu danych może zająć dłuższą chwilę. Aby wyeliminować potencjalne zjawisko zatrzymywania aplikacji modułu sterującego (w przypadku oczekiwania na rezultat transmisji) zwłaszcza, że korzysta on z graficznego interfejsu użytkownika, zastosowano odpowiednie mechanizmy wysyłania, synchronizacji i kontroli transmisji pakietów danych, o czym później.

Budowa urządzenia Moduł wykonawczy (wyłącznik)

W tym miejscu zakończmy opis części poświęconej logicznej konstrukcji systemu iControl przechodząc w końcu do realizacji praktycznej. Zacniemy od sterownika prostszego, którym jest moduł wykonawczy typu wyłącznik 1-/2-biegunowy (układ realizuje obie funkcjonalności) pokazany na **rysunku 5**.

To system mikroprocesorowy, którego „sercem” jest mikrokontroler ATmega8 odpowiedzialny za realizację całej, założonej funkcjonalności. Mikrokontroler ten, przy pomocy wbudowanego, sprzętowego interfejsu TWI (odpowiednik I²C) steruje pracą modemu CY8CPLC10 a dzięki wykorzystaniu przerwania zewnętrznego INT1, które to wyzwala jest przez modem (jego wyprowadzenie HOST_INT), umożliwia odbiór i odpowiednią analizę wysyłanych pakietów danych. ATmega8 odpowiedzialna jest również za obsługę lokalnego interfejsu wejściowego (zaciski SW1, SW2 i COM) to znaczy, obsługę dołączonego do modułu wykonawczego, typowego wyłącznika 1- lub 2-biegunowego. Jest to niezbędne, gdyż moduł wykonawczy zaprojektowano w taki sposób, aby oprócz sterowania zdalnego za pomocą magistrali PLC i modułu sterującego, zapewnić możliwość sterowania lokalnego zwykłym wyłącznikiem naciennym. Aby zrealizować tę funkcjonalność zaprzęgnięto do pracy wbudowany w mikrokontroler Timer0, którego przerwanie od przepełnienia, generowane co 32 ms, sprawdza czy nie zmieniło się położenie lokalnego wyłącznika i w wypadku detekcji takiej zmiany inicjuje przesłanie odpowiedniego rozkazu sterującego (**SEND_NODE_STATUS**) do modułu sterującego, by bieżący stan pracy pokazany przy udziale jego interfejsu użytkownika tegoż modułu zgadzał się ze stanem faktycznym. Oprócz tego, mikrokontroler steruje pracą diody LED oznaczonej STATUS, która to sygnalizuje bieżący stan pracy modułu wykonawczego, pracą dwóch stopni tranzystorowych sterujących wyjściowymi przełącznikami dużej mocy będącymi wyjściowym interfejsem wyłącznika 1-/2-biegunowego oraz obsługuje dwie zworki oznaczone TYPE i UNBIND. Pierwsza z nich (TYPE) decyduje o typie modułu wykonawczego, co determinuje sposób jego sterowania przez moduł sterujący oraz sposób prezentacji w ramach graficznego interfejsu użytkownika. Pozostawienie tej zworki rozwartej powoduje wybór trybu pracy w roli wyłącznika 2-biegunowego, a wtedy każdy z przełączników wyjściowych jest sterowany niezależnie. Zwarcie zworki powoduje ustawienie trybu wyłącznik 1-biegunowego i jednoczesne sterowanie przełącznikami mocy.

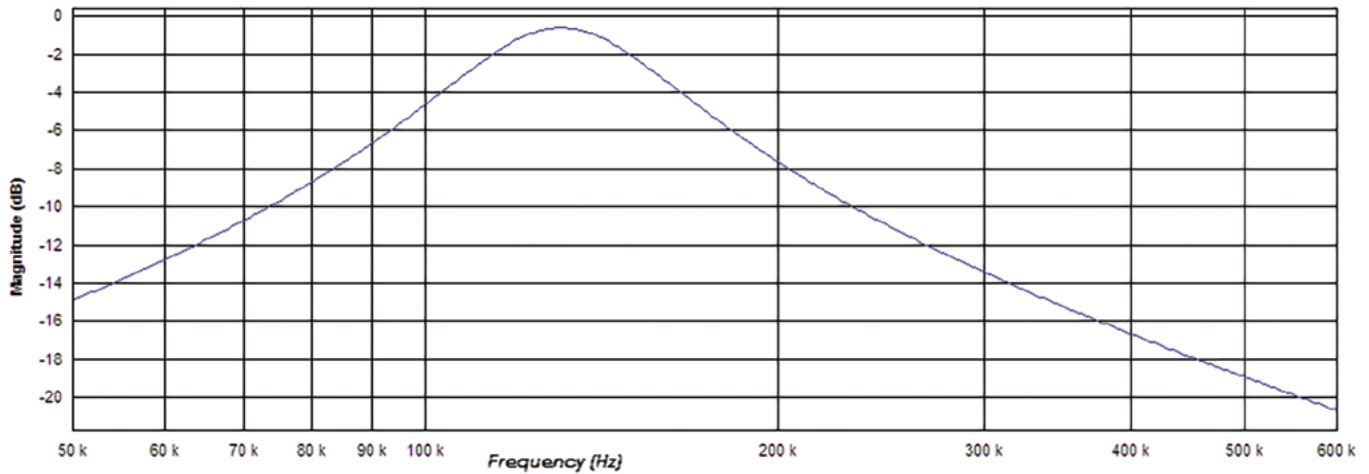


Rysunek 5. Schemat ideowy modułu wykonawczego systemu iControl typu wyłącznik 1/2-biegowy

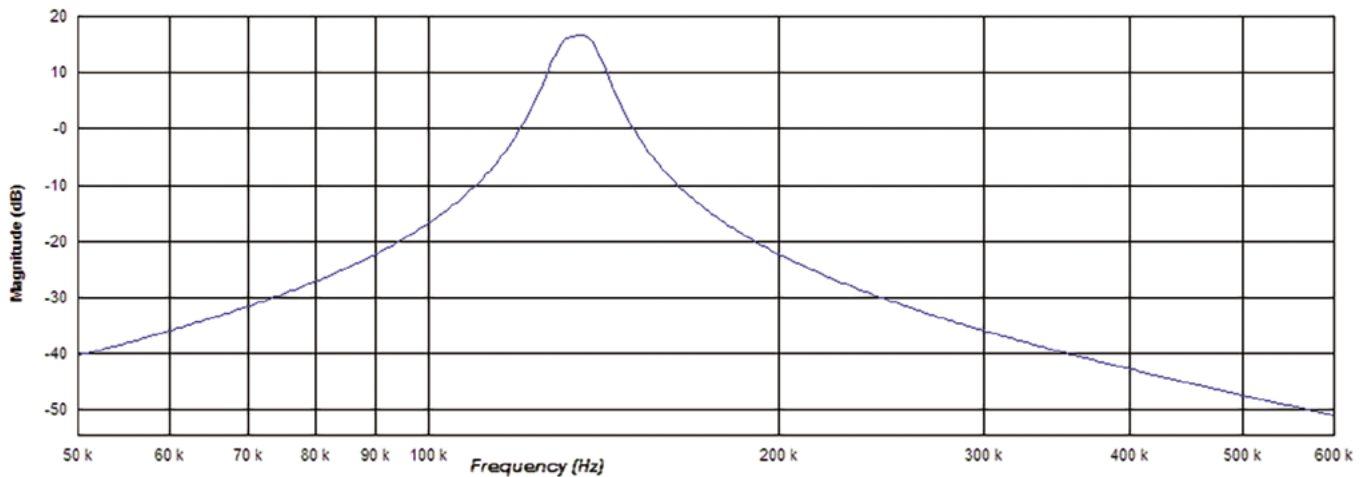
Co oczywiste, raz skonfigurowany i zalogowany do sieci moduł wykonawczy nie powinien być rekonfigurowany w ramach tego ustawienia, gdyż spowoduje to jego błędne działanie oraz nieodpowiednią reprezentację w ramach graficznego interfejsu użytkownika. Ustawienie to może być zmienione dopiero po wylogowaniu modułu z sieci a następnie wyłączeniu i ponownym jego włączeniu.

Druga ze zwrotek, oznaczona jako UNBIND służy do „awaryjnego”, lokalnego zresetowania stanu modułu wykonawczego do stanu wyjściowego (moduł nieskonfigurowany), przy czym jej stan jest sprawdzany wyłącznie podczas włączania urządzenia, zaś potwierdzenie zadziałania tego mechanizmu jest zaświecenie się diody STATUS na czas 0,5 sekundy. Funkcji tej można użyć na przykład w przypadku, gdybyśmy odłączyli zalogowany wcześniej i skonfigurowany moduł wykonawczy od sieci (z nadanym adresem LA) i chcieli go użyć w innej sieci. Wtedy, ponieważ ten moduł miałby już nadany adres LA, nie zgłosiłby się automatycznie w nowej sieci i jedynym sposobem „zmuszenia” go do tej czynności byłoby sprzętowe, lokalne zresetowanie jego stanu. Niemniej jednak, nie polecam tego scenariusza postępowania, gdyż konfiguracja czy rekonfiguracja sieci powinna być procesem przemyślanym i wykonywanym z poziomu modułów sterujących. Dla porządku, należy dodać, iż moduł wykonawczy obsługuje następujące rozkazy sterujące: *SELECT_NODE*, *UNSELECT_NODE*, *SET_LOGICAL_ADDR*, *UNBIND_NODE*, *GET_NODE_STATUS* i *SET_NODE_STATUS*. Przejdźmy, zatem do części analogowej naszego modułu wykonawczego, w konstrukcji której posłużono się schematem aplikacyjnym firmy Cypress (nota CY3274), dzięki czemu całe urządzenie spełnia wymagania norm CENELEC EN50065-1:2001 i FCC part 15. Jak widać, zmodulowany sygnał wyjściowy układu CY8CPLC10 dostępny na wyprowadzeniu FSK_OUT wchodzi na aktywny filtr pasmowo-przepustowy 4-rzędu typu Czebyszewa, zbudowany przy użyciu wysokiej klasy, szerokopasmowych wzmacniaczy operacyjnych U3 i U4 typu LMH6639MF, by następnie trafić na wejście wzmacniacza mocy zbudowanego przy użyciu wzmacniacza operacyjnego U5 i pary komplementarnych tranzystorów mocy T2/T3. Charakterystykę przenoszenia filtra wyjściowego pokazano na **rysunku 6**.

Wspomniany wcześniej wzmacniacz mocy zapewnia dodatkowe wzmocnienie sygnału wyjściowego na poziomie 12 dB i, co ważne, jest zdolny doysterowania obciążenia o niskiej impedancji, jakie to stanowi sieć zasilająca. Wyjście ze wzmacniacza mocy, poprzez kondensator C23, separujący składową stałą, wchodzi na transformator separujący wysokiej częstotliwości TR1 by następnie trafić do sieci zasilającej. Dodatkowo, w budowie obwodu wyjściowego wykorzystano możliwość dezaktywacji końcówek mocy (w momencie, gdy układ CY8CPLC10 nie transmituje danych), a wszystko dzięki wykorzystaniu wyprowadzenia TX_DISABLE modemu. Funkcja ta ma dwojakie znaczenie: po pierwsze zmniejsza pobór mocy przez modem ze źródła zasilania, gdy modem nie transmituje żadnych danych, a pod drugie, powoduje, iż układ wyjściowy (w czasie, gdy jest nieaktywny) przechodzi w stan wysokiej impedancji nie wprowadzając tym samym dodatkowego tłumienia do użytecznego sygnału wejściowego (nie obciąża sygnału wejściowego). Warto w tym miejscu podkreślić, iż jeśli nasz projekt nie musi spełniać wymagań wspomnianych wyżej norm, całą wyjściową część analogową można zwyczajnie pominąć. Jeśli zaś chodzi o układ wejściowy, którego zadaniem jest kondycjonowanie wejściowego sygnału sieciowego, mamy do czynienia ze znacznie prostszą aplikacją. Wejściowy sygnał sieciowy, przez ten sam transformator separujący TR1 wprowadzany jest, poprzez rezystor R22 zapewniający odpowiednią impedancję wejściową, na filtr separujący częstotliwości niepożądane (np. częstotliwości AM) zbudowany z kondensatora C24, a cewki L1 i rezystora R22, aby następnie trafić na tłumik amplitudy w postaci diody Schottky D2 (BAT54S). Dalej, za pomocą rezystorów R23 i R24, sygnał jest polaryzowany połową napięcia zasilającego (+5 V) by trafić na wejście sygnału FSK modemu CY8CPLC10 oznaczone jako FSK_IN. Charakterystykę przenoszenia wspomnianego filtra wejściowego pokazano na **rysunku 7**.



Rysunek 6. Charakterystyka przenoszenia filtra wyjściowego sygnału FSK układu CY8CPLC10



Rysunek 7. Charakterystyka przenoszenia filtra wejściowego sygnału FSK układu CY8CPLC10

Dodatkowo, urządzeniu wykorzystano wyprowadzenie układu CY8CPLC10 oznaczone BIU_LED, do którego dołączono czerwoną diodę LED. Jej zadaniem jest sygnalizowanie zajętości medium transmisyjnego, czyli faktu zajęcia sieci zasilającej przez transmisję danych dokonywaną przez inny modem PLC. Kilka słów uwagi należy się również aplikacji zapewniającej zasilanie modułu wykonawczego.

Podstawowym wymaganiem odnośnie do zasilacza modułu wykonawczego była konieczność zaprojektowania najprostszego, niezawodnego i charakteryzującego się najmniejszymi wymiarami zewnętrznymi źródła napięcia zasilania, ponieważ moduł wykonawczy jest przeznaczony do zamontowania w typowej puszcze podtynkowej o średnicy 60 mm (najlepiej o zwiększonej głębokości). Muszę przyznać, iż nie było to zadanie łatwe, gdyż zastosowanie standardowego transformatora SMD, nawet miniaturowego, powodowało zwiększenie wymiarów obwodu drukowanego, co praktycznie uniemożliwiało zamontowanie płytki w puszcze. Szybko okazało się, iż jedynym sensownym rozwiązaniem jest zastosowanie specjalizowanej przetwornicy napięcia. Ostatecznie zastosowałem specjalizowany układ scalony nieizolowanej (beztransformatorowej) przetwornicy impulsowej typu LNK305 z rodziny LinkSwitch-TN firmy Power Integrations. w jego typowej aplikacji dostarczającej napięcia zasilania 12V przy maksymalnym prądzie obciążenia 225 mA, co zapewniało spory zapas mocy. Niestety, szybko okazało się, że rozwiązanie takie przysparza wielu kłopotów. Po pierwsze, częstotliwość pracy tej przetwornicy równa podwójnej wartości częstotliwości rezonatora kwarcowego modemu PLC (32768 Hz) wpływała na stabilność pracy oscylatora PLL zintegrowanego w układzie CY8CPLC10 powodując problemy z jego uruchomieniem i pracą interfejsu I²C. Po drugie, fakt braku izolacji galwanicznej wymuszał stosowanie zewnętrznego źródła zasilania podczas programowania mikrokontrolera w trakcie uruchamiania i testowania

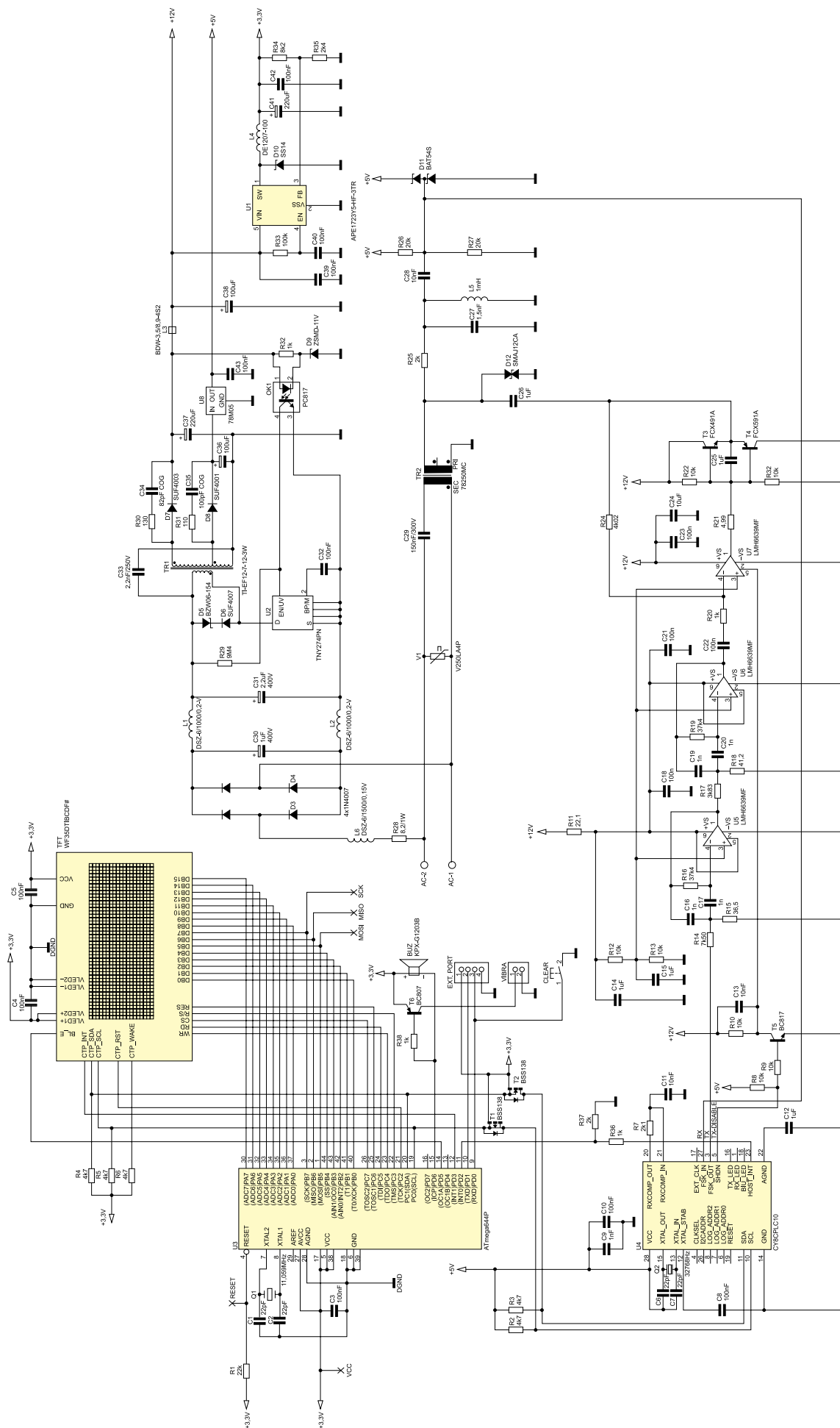
modułu. Niestety, mimo wielu zalet takiego rozwiązania zmuszony byłem porzucić pierwotne rozwiązanie i poszukiwać alternatywy. W tym czasie natknąłem się na gotowe rozwiązanie firmy Myrra pod postacią izolowanej, kompletnej przetwornicy napięcia zabudowanej w typowej, niewielkiej obudowie charakterystycznej dla małych transformatorów przeznaczonych do montażu na płytkach drukowanych. Mowa o przetwornicy M47114 z szerokiej palety produktów tej firmy (serii M47000), która ma następujące parametry:

- Napięcie wejściowe 85...265 VAC.
- Częstotliwość napięcia wejściowego 47...440 Hz.
- Napięcie wyjściowe 12 V DC.
- Maksymalny prąd wyjściowy 200 mA.
- Moc 2,4 W.
- Sprawność 74%,.
- Wymiary zewnętrzne takie, jakie ma najmniejszy transformator do druku – 31 mm 7 mm×26,7 mm.

Rozwiązanie to, mimo że droższe, idealnie nadaje się do wykorzystania w tej aplikacji i dlatego zostało wybrane jako alternatywa dla pierwszej koncepcji. Jako element zapewniający zasilanie układu CY8CPLC10 (+5 V) zastosowano zwykły stabilizator liniowy 78M05, gdyż pobór mocy przez modem PLC jest niewielki i straty mocy na stabilizatorze liniowym są akceptowalne i nie powodują dodatkowego nagrzewania się elementów. Tak oto powstał niewielki zasilacz, który z powodzeniem może być wykorzystywany w tego rodzaju aplikacjach.

Moduł sterujący

Schemat ideowy modułu sterującego pokazano na **rysunku 8**. Jego sercem jest zaawansowany mikrokontroler ATmega644A



Rysunek 8. Schemat ideowy modułu sterującego systemu iControl

Listing 10. Konstrukcje struktur danych przechowujących parametry modułów wykonawczych i ich grup

```
//Deklaracja typu przechowującego informacje o module wykonawczym
typedef struct
{
    volatile uint8_t LA; //NODE_EMPTY = 0xFF
    volatile uint8_t Type;
    volatile uint8_t Status;
    volatile uint8_t Owner; //NODE_OWN = 0xFF, NODE_SHARED_TO = 0x01, NODE_SHARED_FROM = 0x02
    char Name[8];
    uint8_t Room;
} nodeType;

//Deklaracja typu przechowującego informację o grupie modułów wykonawczych
typedef struct
{
    uint8_t Status; //ROOM_ACTIVE = 0x80, ROOM_INACTIVE = 0xFF
    char Name[8];
} roomType;
```

taktowany zewnętrznym rezonatorem kwarcowym o częstotliwości 11,0592 MHz. Mikrokontroler za pomocą wbudowanego, sprzętowego interfejsu TWI (odpowiednik I²C) steruje pracą modemu CY8CPLC10, a dzięki wykorzystaniu przerwania zewnętrznego INT0, które jest wyzwalane przez modem PLC (wyprowadzenie HOST_INT), umożliwia odbiór i odpowiednią analizę wysyłanych pakietów danych. ATmega644A jest również odpowiedzialna za obsługę graficznego interfejsu użytkownika GUI z wyświetlaczem TFT firmy Winstar typu WF35QTIBCDBC0# (sterowanego za pomocą 16-bitowej magistrali danych) ze zintegrowanym kontrolerem pojemnościowego panelu dotykowego CTP. Dodatkowo, za pomocą przerwania zewnętrznego INT1 i interfejsu TWI mikrokontroler komunikuje się z wbudowanym w wyświetlacz TFT kontrolerem panelu dotykowego. Dzięki dużej częstotliwości taktowania mikrokontrolera oraz 16-bitowej magistrali danych wyświetlacza TFT, szybkość wczytywania z pamięci Flash mikrokontrolera obrazków stanowiących elementy graficznego interfejsu użytkownika (zajmują one większą część pamięci programu aplikacji) jest na tyle duża, iż zapewnia płynne i zgodne z założeniami działania wspomnianej platformy współpracy urządzenia z użytkownikiem. Jakby tego było mało, i co pewnie zaskoczy wielu Czytelników, mimo niewielkiej, zdawałoby się, mocy obliczeniowej mikrokontrolerów AVR, w implementacji interfejsu użytkownika udało się nawet zaimplementować mechanizmy zapewniające uzyskanie **przeźroczystości obiektów** (dokładnie okna opcji modułu wykonawczego), co czyni sam interfejs jeszcze bardziej atrakcyjnym i nowoczesnym.

Mikrokontroler jest zasilany napięciem +3,3 V (z uwagi na napięcie zasilania panelu TFT), zaś układ CY8CPLC10, napięciem +5 V, niezbędnym stał się odpowiedni, dwukierunkowy translator poziomu napięcia zbudowany przy użyciu tranzystorów T1 i T2. Ponadto, mikrokontroler steruje pracą buzzera piezoelektrycznego, który sygnalizuje fakt naciśnięcia panelu dotykowego. Jako opcję alternatywną, można zastosować silniczek elektryczny o specjalnej konstrukcji (tzw. *vibra* znana z telefonów komórkowych) i dołączyć go do złącza VIBRA, dający wrażenie wciskania rzeczywistego przycisku mechanicznego dzięki generowaniu wibracji.

Tor analogowej transmisji FSK nie różni się on od tego, który zastosowano w module wykonawczym i dlatego nie będę powtarzał przytoczonych wcześniej założeń dotyczących jego budowy. Zupełnie inaczej mają się sprawy, jeśli chodzi o aplikację zasilacza modułu sterującego, ponieważ ten moduł wymaga kilku napięć zasilających:

- +3,3 V do zasilania mikrokontrolera i panelu TFT.
- +5 V do zasilania modemu CY8CPLC10.
- +12 V do zasilania aplikacji kondycjonującej sygnał wejściowy/wyjściowy modemu PLC.


Ponieważ moje doświadczenie w zakresie projektowania zasilanych, impulsowych przetwornic w układach zasilania jest – mówiąc ogólnie – niewystarczające, zdałem się na aplikację firmy Feryster (w zakresie napięć 5 i 12 V), która udostępnia projekty skomplikowanych układów zasilania o dowolnej konfiguracji. W ten sposób powstał projekt zasilacza pokazany na **rysunku 8**. Zasilacz ten wykorzystuje specjalizowany układ scalony przeznaczony do budowy wysokosprawnych przetwornic impulsowych typu TNY274PN z rodziny

TinySwitch-III firmy Power Integrations. Tym razem do czynienia mamy ze znacznie bardziej skomplikowaną aplikacją, tak naprawdę wygenerowaną przy udziale doskonałego narzędzia *PI Expert Design Software*. Zastosowano rozwiązanie z separacją galwaniczną przy udziale specjalnej konstrukcji transformatora impulsowego wysokiej częstotliwości TR1 (firmy Feryster) i transoptora OK1. Wspomniany zasilacz dostarcza napięć +12 V i +7 V. Napięcie +5 V uzyskujemy poprzez zastosowanie typowej aplikacji, liniowego stabilizatora 78M05, zaś napięcie +3,3 V uzyskujemy z +12 V przy użyciu przetwornicy impulsowej zbudowanej na podstawie standardowej aplikacji układu APE1723Y5-HF-3TR firmy Advanced Power Electronics Corp. Dzięki temu uzyskano wysoką sprawność układu i małe straty mocy przy zachowaniu wymagań dotyczących prądu obciążenia. Dzięki takiej konstrukcji bloku zasilania, zasilacz zajął niewielką część obwodu drukowanego modułu sterującego, mieszcząc się częściowo pod modulem wyświetlacza TFT. Ma przy tym bardzo wysoką sprawność (elementy nie wydzielają w ogóle niekorzystnego ciepła) i niski koszt budowy.

Oprogramowanie modułu sterującego

Moduł sterujący może obsługiwać do 64 modułów wykonawczych nadając im unikalne adresy logiczne w procesie konfiguracji sieci. Moduły te, mogą z kolei zostać podzielone na maksymalnie 8 grup reprezentujących pomieszczenia, w jakich zostały one zamontowane. Wszystkie informacje dotyczące zarówno samych modułów wykonawczych jak i utworzonych grup przechowywane są w nieulotnej pamięci mikrokontrolera typu EEPROM w postaci struktur danych, które pokazano na **listingu 10**.

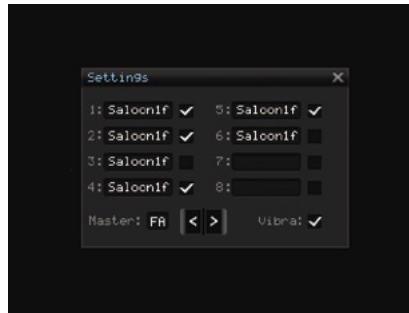
Informacje te wczytywane są do pamięci RAM przez program obsługi aplikacji na samym początku programu obsługi (i według potrzeby aktualizowane później) i wykorzystywane następnie przez funkcje zapewniające obsługę graficznego interfejsu użytkownika GUI. Ponadto, przewidziano dodatkową strukturę danych (**listing 11**), której zadaniem jest przechowywanie parametrów nowych, jeszcze nieskonfigurowanych modułów wykonawczych, które zgłosiły fakt podłączenia do sieci poprzez cykliczne rozgłaszanie swojego adresu PA.

Zarejestrowanie przynajmniej jednego, nowego i jeszcze nieskonfigurowanego modułu wykonawczego powoduje pojawienie się na ekranie graficznego interfejsu użytkownika ikonki  i daje możliwość jego późniejszej konfiguracji tj. nadania mu unikalnej nazwy i przyporządkowania do jednej z utworzonych wcześniej grup modułów (pokojów). Co ważne, program obsługi aplikacji na samym początku sprawdza stan konfiguracji grup modułów wykonawczych i w przypadku, gdy nie została jeszcze zdefiniowana żadna grupa (nie uczyniono jej aktywną i nie nadano nazwy) nie pozwala na przejście do ekranu głównego aplikacji. Aby przybliżyć nieco ideę funkcjonowania graficznego interfejsu użytkownika systemu iControl nie sposób nie opisać kluczowych elementów i ekranów tego interfejsu. Na **rysunku 9** przedstawiono przykładowy ekran główny systemu iControl z wyświetlonymi wszystkimi elementami sterującymi.

Zaprojektowano estetyczny, przejrzysty, graficzny interfejs użytkownika, który na ekranie głównym pokazuje wszystkie, przyporządkowane do danej (wyświetlanej) grupy moduły wykonawcze (z ich



Rysunek 9. Przykładowy ekran główny systemu iControl



Rysunek 10. Ekran konfiguracji modułu sterującego systemu iControl



Rysunek 11. Wygląd okna przeznaczony do edycji nazw w systemie iControl

nazwami własnymi), nazwę grupy (w prawym, dolnym rogu ekranu), liczbę zdefiniowanych grup i znacznik położenia aktualnie wyświetlanej grupy (w lewym, dolnym rogu ekranu). Dwie ikonki (w prawym, górnym rogu ekranu) dają dostęp do konfiguracji nowego modułu wykonawczego, jeśli taki moduł zgłosił się w sieci i konfiguracji modułu sterującego w zakresie definicji grup (ich aktywacji/dezaktywacji, nadania nazw) oraz adresu logicznego w sieci (z zakresu 0xF0...0xFF). Ikona (w lewym, górnym rogu ekranu), która pojawia się w wypadku wystąpienia błędów transmisji daje dostęp do konsoli błędów systemu iControl.

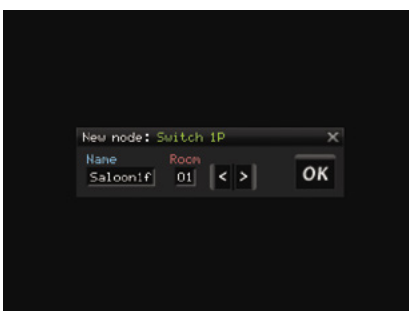
Co oczywiste, graficzna reprezentacja każdego z modułów wykonawczych, jak i możliwości w zakresie jego sterowania, zależą od typu modułu wykonawczego (w tej chwili aplikacja umożliwia obsługę 3 typów modułów wykonawczych: wyłącznik 1- i 2-biegunowy oraz sensor temperatury). Dla przykładu, dla modułu wykonawczego typu wyłącznik 1-2-biegunowy, każde naciśnięcie elementu symbolizującego wyłącznik (jeden lub dwa, niezależne wyłączniki w zależności od typu modułu) powoduje stosowną zmianę na ekranie modułu sterującego jak i zmianę stanu pracy tegoż modułu wykonawczego, zaś dla modułu wykonawczego typu sensor temperatury nie powoduje żadnej akcji ani po stronie modułu sterującego, ani też po stronie modułu wykonawczego (gdyż moduł ten cyklicznie i samodzielnie wysyła bieżący pomiar temperatury do modułu sterującego). Naciśnięcie pola opcji na wybranym module wykonawczym (symbol „krzyżyka” w prawym, górnym rogu pola modułu wykonawczego) powoduje wyświetlenie półprzezroczystego okna opcji i oczekiwanie na ruch po stronie użytkownika. Dostępne są następujące opcje:

- **Share** umożliwiająca współdzielenie modułu wykonawczego z innym modułem sterującym (jeśli tylko moduł ten jest „dzieckiem” bieżącego modułu sterującego).
- **Delete** umożliwiająca usunięcie modułu wykonawczego z listy tychże modułów dla bieżącego modułu sterującego (i wylogowanie z sieci, jeśli bieżący moduł sterujący jest „rodzicem” usuwanego modułu wykonawczego).
- **Cancel** powodująca powrót do ekranu głównego zamykając listę opcji (taka sama akcja zostanie podjęta przez moduł sterujący, gdy

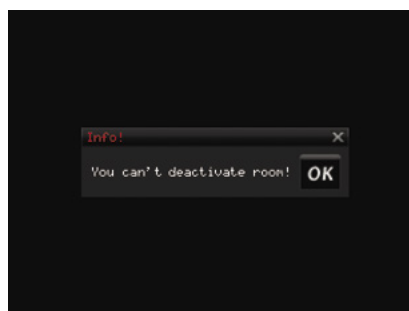
wyberzemy jakkolwiek inny obszar ekranu poza obszarem wyświetlonej listy opcji).

Dodatkowo, w wypadku, gdy moduł wykonawczy nie wykonał przesłanego rozkazu (gdyż został usunięty z sieci bez wylogowania lub też na skutek nieudanej transmisji PLC), aplikacja interfejsu użytkownika zasygnalizuje ten fakt poprzez zmianę barwy modułu wykonawczego, co pokazano dla modułu o nazwie „Lamp” na przykładowym rysunku. Przejdźmy, zatem do ekranu konfiguracji modułu sterującego, który pokazano na **rysunku 10**. Umożliwia on umożliwić aktywację/dezaktywację każdej z grup modułów wykonawczych, edycję nazwy tychże grup (poprzez wybranie pola nazwy edytowanej grupy i pod warunkiem, iż grupa ta jest aktywna – wybrano wcześniej odpowiedni checkbox) oraz umożliwić wybór adresu logicznego naszego modułu sterującego. Adres logiczny każdego z modułów sterujących musi być **unikalny** w ramach jednej i tej samej sieci (z przedziału 0xF0...0xFF). Co ważne, dezaktywacja grupy możliwa jest wyłącznie wtedy, gdy do grupy tej nie przyporządkowano jeszcze żadnego modułu wykonawczego. Nie jest także możliwa dezaktywacja grupy, do której nie przyporządkowano żadnego modułu wykonawczego, ale grupa ta jest jedyną aktywną grupą w ramach modułu sterującego.

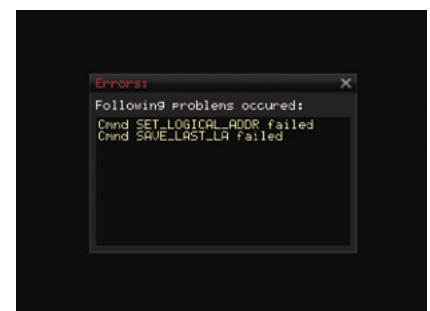
Pole oznaczone „Vibra” służy do włączenia/wyłączenia opcji obsługi silniczka wibracyjnego symulującego efekt naciskania rzeczywistego przycisku. Zaznaczenie tego pola powoduje w rzeczywistości wydłużenie czasu wysterowania buzzera/silniczka zapewniając odpowiednie wrażenia dotykowe. Na koniec opisu tego elementu GUI warto podkreślić, że wybór edycji nazwy grupy modułów wykonawczych powoduje wyświetlenie okna przeznaczony do edycji nazw w systemie iControl (okno to wyświetlane jest za każdym razem w przypadku wyboru edycji jakiegokolwiek nazwy), którego wygląd pokazano na **rysunku 11**. To okno jest zbliżone wyglądem do rozwiązań znanych choćby z interfejsów stosowanych w telefonach komórkowych. Jak łatwo się domyślić, klawisze numeryczne służą do zmiany wartości edytowanego znaku nazwy (każdy, w ramach listy znaków zdefiniowanej dla niego), zaś klawisze oznaczone „<” i „>” służą do zmiany pozycji edytowanego znaku nazwy. Klawisz „OK” zamyka okno edycyjne i zatwierdza wprowadzone dane. Pole oznaczone symbolem „krzyżyka” (w prawym, górnym rogu okna edycyjnego) służy do zamknięcia okna edycyjnego i porzucenia zmian.




Rysunek 12. Wygląd okna edycyjnego pozwalającego na dodanie nowego modułu wykonawczego



Rysunek 13. Wygląd okna decyzyjnego (lub informacyjnego) systemu iControl



Rysunek 14. Wygląd konsoli błędów systemu iControl z przykładową listą błędów

Przejdźmy, zatem do okna edycyjnego pozwalającego na dodanie nowego modułu wykonawczego, które jest wywoływane poprzez naciśnięcie ikonki  z poziomu ekranu głównego systemu iControl, jednak tylko wtedy, gdy system zarejestrował dołączenie nowego modułu. Wygląd wspomnianego okna pokazano na **rysunku 12**. Klawisze

oznaczone „<” i „>” służą do zmiany numeru grupy, do jakiej przyporządkujemy nowy moduł wykonawczy (typ tego modułu pokazany jest na „belce” informacyjnej okna), przy czym dostępna lista grup zależy od konfiguracji przeprowadzonej w ramach okna konfiguracyjnego systemu iControl (pokazywane są wyłącznie numery aktywnych

Wykaz elementów

Moduł wykonawczy wyłącznika 1-/2-biegunowego

Rezystory: (SMD 0805)

R1: 20 kΩ
R2, R3: 4,7 kΩ
R4: 2,1 kΩ/1%
R5...R7, R16, R17, R19, R20: 10 kΩ
R8: 7,5 kΩ/1%
R9: 36,5 Ω/1%
R10, R13: 37,4 kΩ/1%
R11: 3,83 kΩ/1%
R12: 41,2 Ω/1%
R14, R27, R28: 1 kΩ/1%
R15: 4,99 Ω/1%
R18: 4,02 kΩ/1%
R21: 22,1 Ω/1%
R22: 2 kΩ/1%
R23, R24: 20 kΩ
R25, R26: 2.2 kΩ

Kondensatory: (SMD 0805)

C1, C2, C5, C9, C17...C20, C27: 100 nF (X7R)
C3, C4: 22 pF
C6, C10, C25: 10 nF (X7R)
C7, C11, C12, C22, C23: 1 μF (X7R)
C8, C13...C16: 1000 pF/1% (NPO)
C21: 10 μF (SMD 1206, X7R)
C24: 1,5 nF (X7R)
C26: 150 nF/300 V (polipropylenowy X1, raster 15 mm)
C28: 22 μF/10 V (tantal., SMD „A”)

Półprzewodniki:

U1: ATmega8 (TQFP32)
U2: CY8CPLC10 (SSOP28)
U3...U5: LMH6639MF (SOT23-6)
U6: 78M05 (DPAK)
T1, T4, T5: BC817 (SOT23-BEC)
T2: FCX591A (SOT89-BCE)
T3: FCX491A (SOT89-BCE)
D1: SMAJ12CA (DO-214AC)
D2: BAT54S (SOT23)
D3, D4: 1N4148 (MINIMELF)
STATUS: zielona dioda LED SMD 0805
BIU: czerwona dioda LED SMD 0805

Inne:

TR1: transformator separujący SMD Murata 78250MC
L1: dławik SMD 1 mH (L1812)
PWR1: przetwornica do druku Myrra M47114
Q1: rezonator kwarcowy, zegarkowy 32768 Hz
V1: warystor V250LA4P
PK1, PK2: przekaźnik Relpol RM40-3021-85-1012

Moduł sterujący

Rezystory: (SMD 0805)

R1: 22 kΩ
R2...R6: 4,7 kΩ
R7: 2,1 kΩ/1%
R8...R10, R12, R13, R22, R23: 10 kΩ
R11: 22,1 Ω/1%
R14: 7,5 kΩ/1%
R15: 36,5 Ω/1%
R16, R19: 37,4 kΩ/1%
R17: 3,83 kΩ/1%
R18: 41,2 Ω/1%
R20, R36, R38: 1 kΩ
R21: 4,99 Ω/1%
R24: 4,02 kΩ/1%
R25, R37: 2 kΩ/1%
R26, R27: 20 kΩ
R28: 8,2 Ω/1 W (rezystor bezpiecznikowy)
R29: 9,4 MΩ (0,25 W, raster 5 mm)

R30: 130 Ω (SMD 1206)

R31: 110 Ω (SMD 1206)

R32: 1 kΩ (SMD 1206)

R33: 100 kΩ

R34: 8,2 kΩ

R35: 2,4 kΩ

Kondensatory: (SMD 0805)

C1, C2, C6, C7: 22 pF
C3...C5, C8, C10, C18, C21...C23, C39, C40, C43: 100 nF (X7R)
C9, C16, C17, C19, C20: 1000 pF/1% (NPO)
C11, C13, C28: 10 nF (X7R)
C12, C14, C15, C25, C26: 1 μF (X7R)
C24: 10 μF (X7R, SMD 1206)
C27: 1,5 nF (X7R)
C29: 150 nF/300 V (polipropylenowy X1, raster 15 mm)
C30: 1 μF/400 V (elektrolit., 8/3,5 mm)
C31: 2,2 μF/400 V (elektrolit., 8/3,5 mm)
C32, C42: 100 nF (SMD 1206, X7R)
C3: 2,2 nF/250 V (polipropylenowy X1, raster 7,5 mm)
C34: 82 pF COG (SMD 1206, X7R)
C35: 100 pF COG (SMD 1206, X7R)
C36, C38: 100 μF/16 V (elektrolit., 5/2 mm)
C37: 220 μF/16 V (elektrolit., 8/3,5 mm)
C41: 220 μF/10 V (elektrolit., 5/2 mm)

Półprzewodniki:

U1: APE1723Y5-HF-3TR (SOT23-5)
U2: TNY274PN (DIP08)
U3: ATmega644A (TQFP44)
U4: CY8CPLC10 (SSOP28)
U5...U7: LMH6639MF (SOT23-6)
U8: 78M05 (DPAK)
T1, T2: BSS138 (SOT23)
T3: FCX491A (SOT89-BCE)
T4: FCX591A (SOT89-BCE)
T5: BC817 (SOT23-BEC)
T6: BC807 (SOT23-BEC)
D1...D4: 1N4007/400 V (DO41)
D5: BZW06-154 (DO15)
D6: SUF4007 (MELF)
D7: SUF4003 (MELF)
D8: SUF4001 (MELF)
D9: ZSMD-11V (MINIMELF)
D10: SS14 (SMA)
D11: BAT54S (SOT23)
D12: SMAJ12CA (DO-214AC)
OK1: PC817 (DIL04)

Inne:

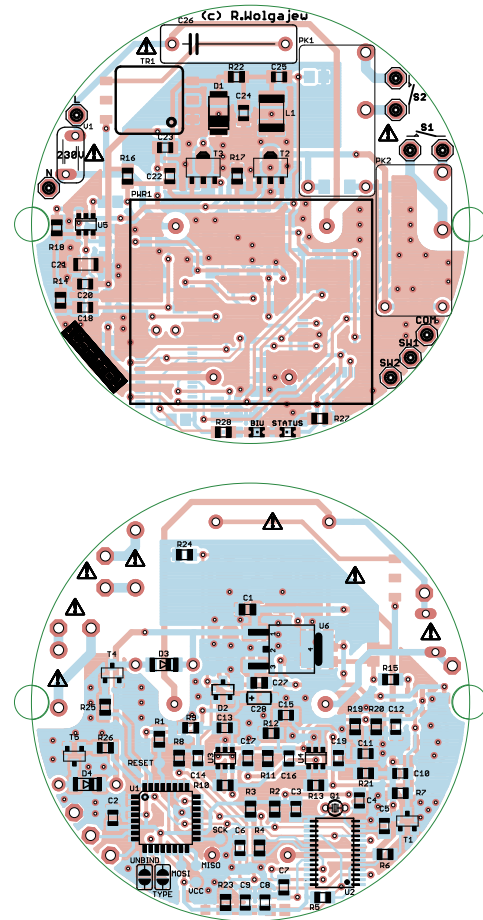
TFT: wyświetlacz TFT Winstar WF35QTIBCDBC0# ze zintegrowanym kontrolerem panelu pojemnościowego CTP (320×240px, złącze ZIFF36 pin)
TR1: transformator Feryster TI-EF12-7-12-3W
TR2: transformator separujący SMD Murata 78250MC
L1, L2: dławik mocy Feryster DSZ-6/1000/0,2-V
L3: koralik ferrytowy 1,5 μH Feryster BDW-3,5/8,9-4S2
L4: dławik mocy SMD 100 μH DE1207-100 (obudowa DE1207)
L5: dławik SMD 1 mH (obudowa L1812)
L6: dławik mocy Feryster DSZ-6/1500/0,15-V
Q1: rezonator kwarcowy 11,0592 MHz, niski
Q2: rezonator kwarcowy, zegarkowy 32768Hz
BUZ: buzzer z generatorem piezo 5 V (raster 5 mm)
EXT.PORT: gniazdo męskie kątowe 90° 4-pin (NSL25-4W)
VIBRA: gniazdo męskie kątowe 90° 2-pin (NSL25-2W)
CLEAR: microswitch SMD typu TACTM-34N-F
V1: warystor V250LA4P
AC: złącze śrubowe AK500/2
Złącze ZIF 36-pin, raster 0.5mm, dolny kontakt, np.: ZIF0536DH
Taśma FFC (Flexible Flat Cable) 36-pin, raster 0.5mm, długość minimum 10cm, np.: FFC0536150: do połączenia wyświetlacza TFT ze złączem ZIF

grup bez ich nazw). Wybór pola nazwy modułu wykonawczego powoduje, jak poprzednio, wyświetlenie okna przeznaczonego do edycji jego nazwy ze wspomnianym wcześniej sposobem obsługi. Klawisz „OK” służy do zamknięcia okna edycyjnego zatwierdzając tym samym dokonaną konfigurację i aktualizując ekran główny modułu sterującego (i powodując wysłanie odpowiednich komunikatów sieciowych), zaś pole oznaczone symbolem „krzyżyka” (w prawym, górnym rogu okna edycyjnego) służy do zamknięcia okna edycyjnego i porzucenia dokonanej edycji (w tym usunięcia modułu wykonawczego z listy nowych i nieskonfigurowanych modułów wykonawczych).

Ostatnim rodzajem okna, z jakim na pewno spotkamy się w czasie obsługi graficznego interfejsu użytkownika systemu iControl jest okno decyzyjne (lub informacyjne) pokazane na **rysunku 13**. Okno takie jest wyświetlane za każdym razem, gdy użytkownik jest informowany o jakimś zdarzeniu lub gdy oczekuje się od niego podjęcia jakiejś akcji (decyzji). Klawisz „OK” służy w tym wypadku do potwierdzenia zapytania wysłanego przez system i zamknięcia okna informacyjnego, zaś pole oznaczone symbolem „krzyżyka” (w prawym, górnym rogu okna edycyjnego) służy do zamknięcia okna informacyjnego i odrzucenia zapytania wysłanego przez system. Ostatnim elementem graficznego interfejsu użytkownika systemu iControl jest, wspomniana wcześniej, konsola błędów, której wywołanie możliwe jest po wystąpieniu błędów transmisji sygnalizowane na ekranie głównym aplikacji i wywołaniu wspomnianej konsoli zainicjowane przyciśnięciem ikonki błędów. Wygląd konsoli błędów systemu iControl pokazano na **rysunku 14**.

Należy zaznaczyć, iż wywołanie wspomnianej konsoli powoduje wyczyszczenie bieżącej listy zarejestrowanych błędów. Pozostała jeszcze jednak kwestia, nad którą zapewne zastanawia się każdy, uważny Czytelnik. W jaki sposób zmieniamy aktualnie wyświetlaną grupę modułów wykonawczych? To proste. Wystarczy, na ekranie głównym, wykonać gest przesunięcia w lewo lub w prawo by przełączać się pomiędzy zdefiniowanymi grupami modułów wykonawczych. Prawda, że proste i intuicyjne? Co ciekawe, w tym mechanizmie, wykorzystano wbudowaną w kontroler pojemnościowego panelu dotykowego, obsługę tzw. gestów znaną chociażby z interfejsów telefonów komórkowych!

Na koniec, kilka słów na temat mechanizmu zapewniającego efektywny sposób transmisji danych na magistrali PLC. Ponieważ transmisja poprzez sieć PLC może zająć dłuższą chwilę (nawet do 3 sekund w skrajnym przypadku), bezcelowe z punktu widzenia interfejsu użytkownika byłoby oczekiwanie na rezultat jej wykonania i tym samym blokowanie programu obsługi aplikacji, w związku z tym konieczne okazało się zastosowanie efektywnego mechanizmu jej przeprowadzenia. W niniejszym rozwiązaniu zastosowano specjalny bufor kołowy pakietów transmisji (z odpowiednimi wskaźnikami miejsca zapisu/odczytu), którego elementem jest struktura danych opisująca kompletną ramkę transmisji oraz 2 funkcje obsługi przerwania: od przepełnienia Timera1, wywoływana cyklicznie co 100ms, której zadaniem jest sprawdzanie czy są jakiegokolwiek dane przeznaczone do wysłania, inicjacja procesu wysłania danych, jego nadzór i obsługa bufora kołowego oraz funkcja obsługi przerwania zewnętrznego INTO, której zadaniem jest zarówno odbiór „zwykłych” danych przesyłanych przez moduły wykonawcze do modułu sterującego jak i odbiór statusów bieżącej i zainicjowanej we wcześniej wspomnianej procedurze obsługi przerwania, transmisji danych. W tym, drugim przypadku, funkcja obsługi przerwania zewnętrznego INTO sprawdza status wysyłania bieżącej transmisji danych i w zależności od jego stanu modyfikuje bieżącą strukturę danych dając asumpt do ponowienia tejże transmisji (w przypadku jej niepowodzenia i nie przekroczenia maksymalnej liczby retransmisji) lub też kończąc jej przebieg poprzez ustawienie odpowiednich flag w strukturze danych (przeprowadzenia transmisji zakończonego powodzeniem lub niepowodzeniem) oraz flag dla funkcji obsługi interfejsu użytkownika, dzięki czemu użytkownik informowany jest na bieżąco o aktualnym stanie systemu (czyli odpowiedzi ze strony sterowanych modułów wykonawczych).



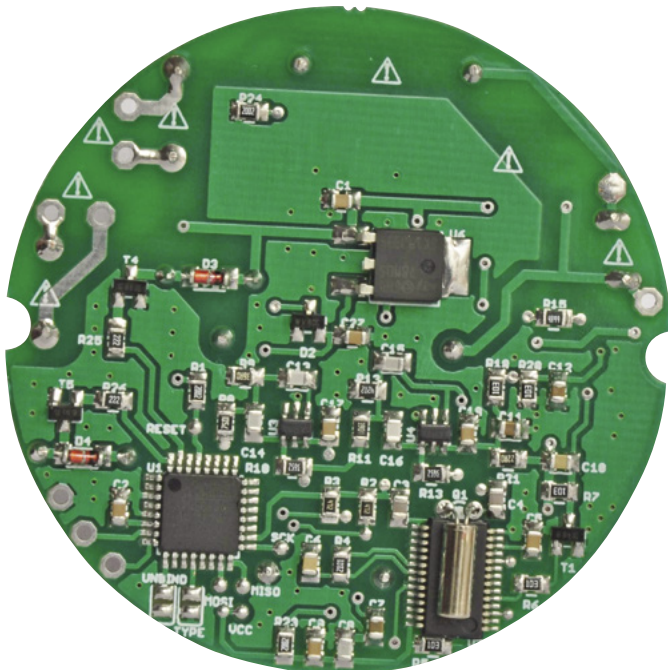
Rysunek 15. Schemat montażowy modułu wykonawczego systemu iControl typu wyłącznik 1-/2-biegunowy.

W ten prosty sposób unika się niepożądanego wstrzymywania pracy graficznego interfejsu użytkownika w przypadku problemów z komunikacją po magistrali PLC.

Montaż

Opis montażu zaczniemy od modułu wykonawczego typu wyłącznik 1-/2-biegunowy – jego schemat montażowy pokazano na **rysunku 15**. Zaprojektowano bardzo zwartą konstrukcję dwustronnego obwodu drukowanego, który w swoich założeniach ma być montowany w typowej puszcze elektroinstalacyjnej o średnicy 60 mm i głębokości 40 mm, przy czym należy zaznaczyć, iż z uwagi na sporą liczbę elementów i małą powierzchnię, zastosowano dość gęsty montaż elementów SMD po obu stronach laminatu. Dla zminimalizowania zakłóceń, na płycie urządzenia poprowadzono obszerne pola masy po obu stronach obwodu drukowanego oraz zastosowano szereg przelotek pomiędzy nimi w celu zmniejszenia pojemności pasożytniczych. Z uwagi na zastosowanie niewielkich elementów SMD, montaż tego typu układu najlepiej jest przeprowadzić z użyciem stacji lutowniczej wyposażonej w grot o niewielkiej średnicy, odpowiedniej jakości topników lutowniczych oraz dysponując sporym doświadczeniem. Jak zwykle, montaż zaczynamy od przylutowania wszystkich półprzewodników w obudowach SMD (po obu stronach laminatu), następnie lutujemy rezystory, kondensatory, elementy indukcyjne, a na samym końcu wszystkie elementy przeznaczone do montażu przewlekane, w tym przetwornicę Myrra M47114. Należy zaznaczyć, iż z uwagi na spore zagęszczenie elementów po obu stronach obwodu drukowanego specjalnego sposobu montażu wymaga rezonator kwarcowy podłączony do układu CY8CPLC10, co najlepiej zobrazuje zdjęcie obwodu drukowanego od strony spodu pokazane na **fotografii 16**.

Poprawnie zmontowany układ nie wymaga żadnych regulacji i powinien działać tuż po włączeniu zasilania. Jediną czynnością,

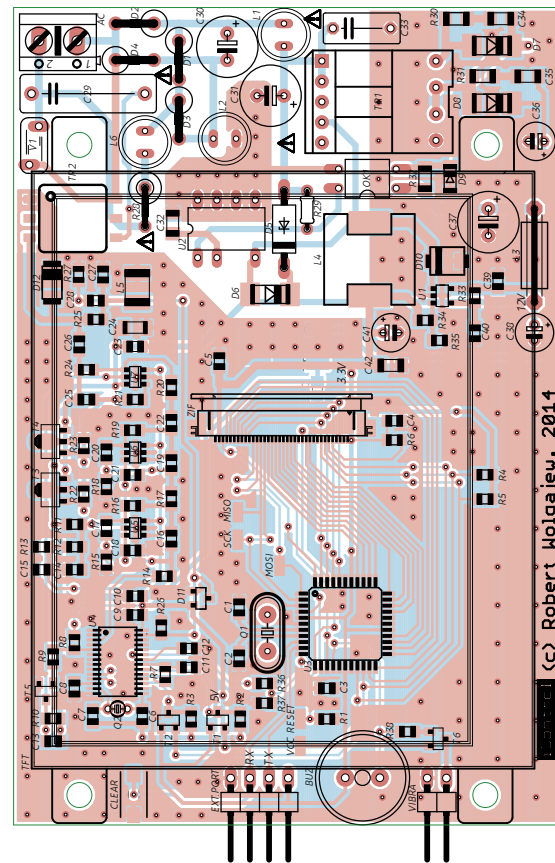


Fotografia 16. Sposób montażu rezonatora kwarcowego podłączonego do układu CY8CPLC10

którą należy wykonać przed użytkowaniem urządzenia jest wybranie rodzaju modułu wykonawczego, jaki ma pełnić uruchamiany układ w systemie iControl, czego dokonujemy za pomocą zworki oznaczonej TYPE (o czym była mowa wcześniej). Przejdźmy, zatem do schematu montażowego modułu sterującego, który pokazano na rysunku 17.

Tym razem, jest to dużo większy, dwustronny obwód drukowany, który swoimi wymiarami przekracza nieco wymiary zastosowanego panelu TFT i choćby z uwagi na ten fakt, można było tym razem zastosować nieco większe elementy SMD montowane wyłącznie po stronie TOP jak i większe odstępki pomiędzy nimi upraszczające nieco proces uruchamiania. Z drugiej strony, z uwagi na fakt, iż na płytce układu zamontowano złącze typu ZIFF przeznaczone do podłączenia wyświetlacza TFT o bardzo gęstym rastrze wyprowadzeń (0.5 mm) montaż tego elementu wymaga pewnej wprawy. Najprostszym sposobem montażu elementów o takim zagęszczeniu wyprowadzeń niewymagającym jednocześnie posiadania specjalistycznego sprzętu jest użycie typowej stacji lutowniczej, dobrej jakości cyny z odpowiednią ilością topnika oraz plecionki rozlutowniczej, która umożliwi usunięcie nadmiaru cyny spomiędzy wyprowadzeń układów. Należy przy tym uważać by nie uszkodzić termicznie tegoż elementu. Następnie lutujemy wszystkie elementy półprzewodnikowe typu SMD, kolejno elementy bierne a na samym końcu pozostałe elementy przeznaczone do montażu przewlekane (w obrębie zasilacza modułu) jak również elementy mechaniczne (złącza, dystanse). Z uwagi na zagęszczenie wyprowadzeń złącza ZIFF jak i układów scalonych, przed pierwszym podłączeniem układu należy jeszcze raz sprawdzić jakość wykonanych połączeń by nie dopuścić do ewentualnych zwarc. Wspomniana kontrola będzie znacznie łatwiejsza, jeśli zmontowana płytkę sterownika przemyjemy alkoholem izopropylowym w celu wypłukania nadmiaru kalafonii lutowniczej. Tuż przed przykręceniem wyświetlacza do płyty naszego układu (przy pomocy czterech, odpowiedniej długości tulei dystansowych), należy go podłączyć korzystając ze złącza ZIFF umieszczonego po stronie elementów i odpowiedniej długości taśmy połączeniowej.

Poprawnie zmontowany układ powinien działać tuż po podłączeniu zasilania. Ewentualnego sprawdzenia mogą wymagać wartości napięć wyjściowych zastosowanej przetwornicy. Widoczne na płytce opcjonalne złącze oznaczone „EXT.PORT” służy do podłączenia modułu interfejsu Bluetooth lub LAN, który daje możliwość dodatkowej, zdalnej kontroli modułu sterującego lub całej sieci PLC (np. z poziomu aplikacji mobilnej), przy czym funkcjonalność ta nie jest

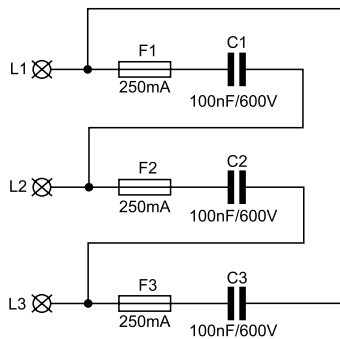


Rysunek 17. Schemat montażowy modułu sterującego systemu iControl

tymczasem zaimplementowana w programie obsługi urządzenia. Opcjonalnie, o czym wspomniano wcześniej, możemy do naszego urządzenia podłączyć także dowolny silniczek typu vibra (z telefonu komórkowego) przy pomocy dedykowanego złącza VIBRA, zaś sam element umocować od spodu modułu wyświetlacza TFT, co zapewni optymalny efekt, jeśli chodzi o odczucia towarzyszące naciskaniu panelu dotykowego.

Problem sieci 3-fazowych

Każdorazowo, pisząc o komunikacji poprzez sieć zasilającą mamy na myśli komunikację w ramach urządzeń transmisyjnych podłączonych do jednej i tej samej fazy tejże sieci, gdyż, jak łatwo się domyślić, komunikacja pomiędzy urządzeniami podłączonymi do różnych faz, w typowych warunkach pracy nie jest możliwa. Jest to spowodowane dużą impedancją między poszczególnymi fazami sieci zasilającej (podyktowaną impedancją poszczególnych urządzeń do niej podłączonych dla tak wysokich częstotliwości) a co za tym idzie niemożliwością przejścia użytecznego sygnału PLC poprzez tę barierę (zbyt dużym tłumieniem sygnału). Z tego samego powodu (impedancji elementów indukcyjnych dla dużych częstotliwości) sygnał PLC nie „przechodzi” do sieci zasilającej znajdującej się za licznikiem energii elektrycznej. Aby wyeliminować ten problem stosuje się różnego rodzaju, mniej lub bardziej rozbudowane, układy sprzęgów 3-fazowych (phase coupler) lub repeaterów (z użyciem technologii bezprzewodowych). Skomplikowanie stosowanego rozwiązania zależy głównie od rozległości sieci zasilającej a co za tym idzie, potrzeby zapewnienia odpowiedniej propagacji sygnału PLC. W tak stosunkowo prostych i niewielkich systemach, jak układ iControl możemy jednak zastosować jedno z najprostszych rozwiązań, które spotyka się w tego rodzaju układach – schemat pokazano na rysunku 18. Pomiędzy każdą z faz sieci zasilającej włączono szeregowo włączony kondensator 22...100 nF (koniecznie na napięcie minimum



Rysunek 18. Trójfazowy sprzęg dla sieci PLC.

600 V) i bezpiecznik 250 mA. Kondensator taki stanowi niewielką impedancję dla częstotliwości rzędu 130...133 kHz, z którą pracują modemy PLC, co zapewnia dobrą propagację sygnału pomiędzy poszczególnymi fazami sieci zasilającej. Z kolei dla częstotliwości sieciowej 50 Hz kondensator taki stanowi dość dużą impedancję, a co za tym idzie niewielkie obciążenie i straty mocy. Bezpiecznik stanowi proste zabezpieczenie na wypadek uszkodzenia (zwarcia) kondensatora szeregowego.

Uruchomienie i konfiguracja systemu

Tak jak wspomniano wcześniej, proces konfiguracji systemu **iControl** wymaga pewnego, wyjściowego stanu systemu. Tym stanem wyjściowym jest taki, w którym do sieci zasilającej zostały podłączone wszystkie, planowane do użycia moduły sterujące, zaś w każdym z nich skonfigurowano unikalny numer adresu logicznego (z zakresu 0xF0÷0xFF) jak i przynajmniej jedną grupę modułów wykonawczych (pomieszczeń). Następnie dołączamy do sieci kolejne moduły wykonawcze, za każdym razem dodając je do wybranego modułu sterującego (czyniąc go ich rodzicem). W przypadku, gdy wybrany moduł wykonawczy zamierzamy współdzielić innemu modułowi sterującemu, operację tę najlepiej przeprowadzić jest przed podłączeniem do sieci i dodaniem kolejnego, nowego modułu wykonawczego. Tak przeprowadzona konfiguracja sieci zapewnia optymalną i bezproblemową obsługę wszystkich urządzeń i jednocześnie skraca czas wdrożenia docelowego systemu. Dla porządku należy dodać, że przewidziano także możliwość wyczyszczenia wszystkich ustawień modułu sterującego, tj. parametrów wszystkich, załogowanych modułów wykonawczych, ustawień grup (pomieszczeń) jak i pamięci zajętych numerów LA bieżącej sieci by moduł sterujący można było przywrócić do stanu wyjściowego (fabrycznego). W tym celu, w trakcie uruchamiania urządzenia należy wcisnąć przycisk oznaczony na obwodzie drukowanym „CLEAR”, co spowoduje, wyczyszczenie pamięci danych modułów wykonawczych, pamięci ustawień grup pomieszczeń oraz pamięci, w której moduł sterujący przechowuje informacje o przyznanym i zajętych numerach adresów LA (oczywiście zostanie wyświetlone stosowne zapytanie). Wygląd interfejsu użytkownika podczas procesu czyszczenia pamięci modułu sterującego pokazano na **rysunku 19**.

Muszę zauważyć, że testy praktyczne systemu **iControl** potwierdziły wręcz doskonałą funkcjonalność rozwiązania

Dodatkowe informacje:

Ustawienia Fuse-bitów (ważniejszych) modułu sterującego:

CKSEL3...0: 1111
SUT1...0: 11
CKDIV8: 1
JTAGEN: 1
EESAVE: 0

Ustawienia fusebitów (ważniejszych) modułu wykonawczego wyłącznika 1-/2-biegunowego:

CKSEL3...0: 0100
SUT1...0: 10
CKOPT: 1
EESAVE: 0



Rysunek 19. Wygląd interfejsu użytkownika podczas procesu czyszczenia pamięci modułu sterującego.

zaproponowanego przez firmę Cypress! Komunikacja z wykorzystaniem sieci zasilającej działała bez jakichkolwiek problemów, a jedyne, jakie zauważono wystąpiły wtedy, gdy do sieci zasilającej podłączono tanią lampę fluorescencyjną (tzw. świetlówkę) z typowym układem zapłonowym. W takim przypadku, by zminimalizować wpływ pracy tego źródła światła na funkcjonowanie systemu **iControl**, należałoby w szereg z zasilaniem lampy fluorescencyjnej dołączyć dławik mocy o indukcyjności rzędu 100...300 μ H (o odpowiedniej mocy), który ograniczyłby wpływ zaburzeń o wysokiej częstotliwości na propagację sygnału PLC w sieci. Zresztą, dość często przyczyną tego stanu rzeczy nie jest właściwie poziom zakłóceń, które generuje odbiornik tego typu (znacznie zmniejszając stosunek S/N), zaś fakt, że na jego wejściu zastosowano kondensator przeciwzakłócenia włączony równolegle do zacisków zasilających, który to dla częstotliwości nośnych sygnału PLC (132 kHz) stanowi obciążenie skutecznie tłumiąc użyteczny sygnał transmisji. Rozwiązaniem tego problemu jest zastosowanie na wejściu (przed przedmiotowym kondensatorem) dławika mocy lub nawet prostego filtra LC. Tematyka ta została zresztą dość ciekawie opisana w nocie aplikacyjnej firmy Cypress oznaczonej symbolem AN58825, której to lekturę szczególnie polecam, gdyż zawiera szereg informacji o tematyce poprawiania jakości sieci zasilającej w świetle zastosowania technologii PLC!

Z uwagi na fakt, iż na płytkach obu układów zbudowano kompletne układy zasilające zasilane napięciem sieciowym 230 V istnieje niebezpieczeństwo porażenia prądem elektrycznym o napięciu 230 V, co stanowić może zagrożenie dla życia i zdrowia użytkowników tychże urządzeń. W związku z tym, montaż układów w tym zakresie powierzyć należy osobie posiadającej uprawnień elektrycznych w zakresie eksploatacji urządzeń o napięciu do 1 kV. Miejsca na obwodach płytek drukowanych, gdzie występuje wysokie napięcie groźne dla życia i zdrowia oznaczone zostały odpowiednimi opisami.

Robert Wołgajew, EP

REKLAMA

ELEKTRONIKA PRAKTYCZNA

Zaprenumeruj na stronie AVT.pl, e-mail: prenumerata@avt.pl
lub telefonicznie pod numerem: 22 257 84 99
Bieżący numer zamów na www.ulubionykiosk.pl

