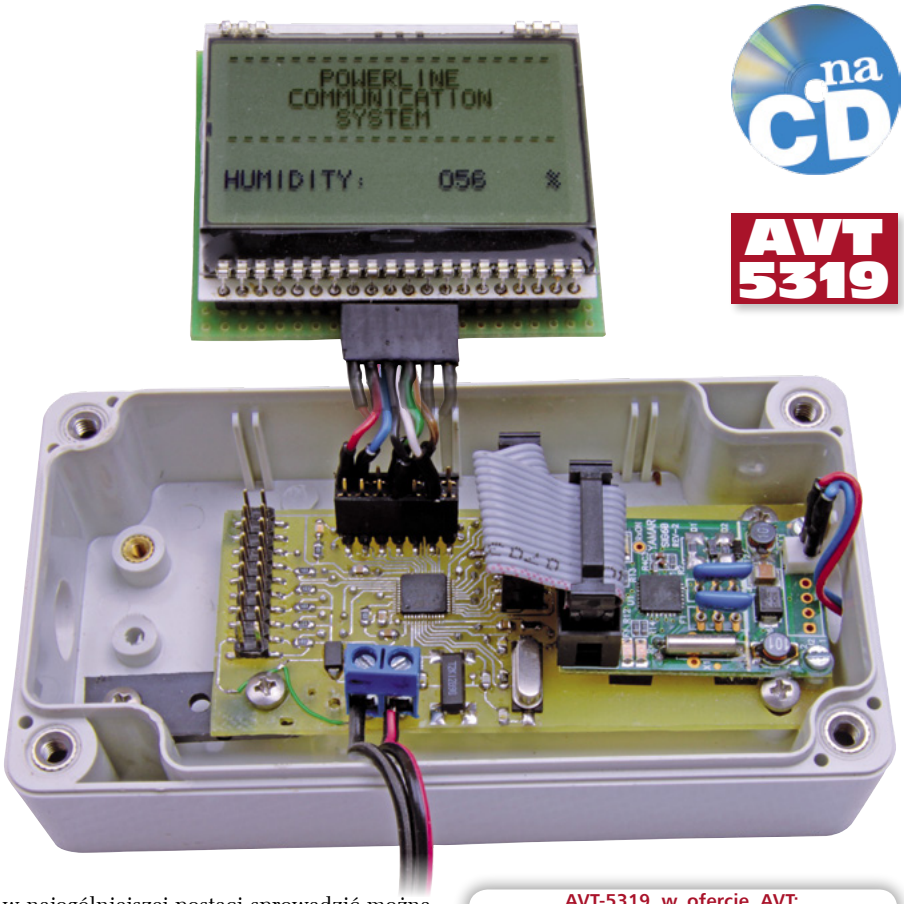
Tor komunikacyjny PLC

Transmisja danych po liniach zasilania



na CD

AVT 5319

Koncepcja transmisji danych z użyciem przewodów zasilających jako kanału komunikacyjnego (Power Line Communication) jest znana od wielu lat. Próby jej implementacji były podejmowane już w pierwszej połowie ubiegłego wieku, jednak z powodu niewystarczającego poziomu ówczesnej techniki, nie mogła ona być zastosowana na masową skalę. Jednak współcześnie możliwość użycia istniejących sieci elektroenergetycznych do wymiany danych przyczyniła się do wdrożenia usług świadczonych za pośrednictwem linii zasilania. Największą zaletą tej technologii jest brak konieczności budowania medium transmisyjnego. W wypadku implementacji PLC np. w samochodzie jest to dodatkowy atut, gdyż z racji istniejących obwodów zasilania proces ten nie wymaga ingerencji w wewnętrzną strukturę tych obiektów, a sygnał jest dostępny niemal w każdym jego punkcie.

Rekomendacje: *System transmisyjny mogący znaleźć zastosowanie w każdym projekcie, w którym istnieje potrzeba połączenia urządzeń w sieć.*

Celem niniejszego projektu było zaprojektowanie oraz wykonanie węzłów komunikacyjnych o funkcjonalności umożliwiającej wykorzystanie współdzielonych linii zasilania jako medium do transmisji danych. Strukturę budowy węzła transmisyjnego w najogólniejszej postaci sprowadzić można do trzech elementów (**rysunek 1**), do których należą:

– moduł komunikacyjny,
– moduł sterujący,
– moduł zasilania.

Moduł komunikacyjny jest odpowiedzialny za cały proces związany z transmisją danych, a więc nadawaniem i odbiorem informacji. Po odebraniu wiadomości nadanej po liniach zasilania jest ona interpretowana i przekazywana w formie cyfrowej do modułu sterującego. Odwrotna sytuacja jest analogiczna – odebranie informacji użytkowej od modułu sterującego inicjuje transmisję danych po liniach zasilania.

Zadaniem modułu sterującego jest poprawna obsługa modułu komunikacyjnego – konfigurowanie, przetwarzanie odbieranych wiadomości oraz sterowanie nadawaniem. Dodatkowo, za pośrednictwem tego modułu do węzła transmisyjnego mogą być dołączone dodatkowe urządzenia lub układy peryferyjne, którymi węzeł może zarządzać.

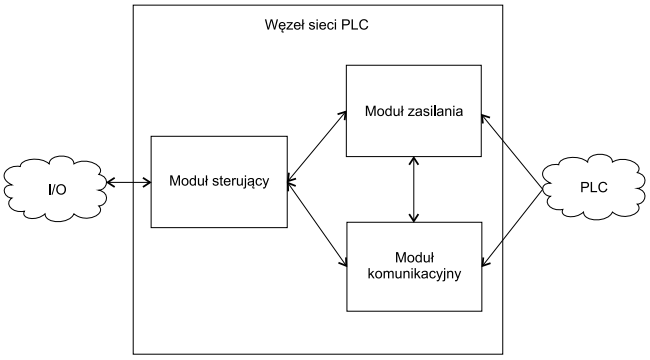
AVT-5319 w ofercie AVT:
AVT-5319A – płytka drukowana

Podstawowe informacje:
- Płytka o wymiarach 90 mm×39 mm.
- Zasilanie 12 V_{DC}.
- Transmisja przez kable zasilające.
- Mikrokontroler STM32F103C8T6.
- Transceiver SIG60 firmy Yamar.

Najważniejsze parametry układu SIG60:
- Napięcie zasilania: 3,0…3,6 V
- Interfejs komunikacyjny: UART (9600 bps…115200 bps)
- Częstotliwość sygnału PLC: 1,75 MHz; 4,5 MHz; 5,5 MHz; 6 MHz; 6,5 MHz; 10,5 MHz; 13 MHz
- Pobór prądu w trybie: nadawania: 40 mA, odbioru: 50 mA, uśpienia: 80 µA,
- Temperatura pracy: –40…+125°C

Dodatkowe materiały na CD/FTP:
ftp://ep.com.pl, user: 15352, pass: 760hp6s5
- wzory płytek PCB
- karty katalogowe i noty aplikacyjne elementów oznaczonych w Wykazie elementów kolorem czerwonym

Ostatni moduł zwany zasilającym odpowiedzialny jest za dostarczenie do pozostałych modułów napięcia umożliwiającego ich poprawną pracę. Odbywa się to poprzez konwersję napięcia dostępnego na liniach transmisyjnych do poziomu wyma-

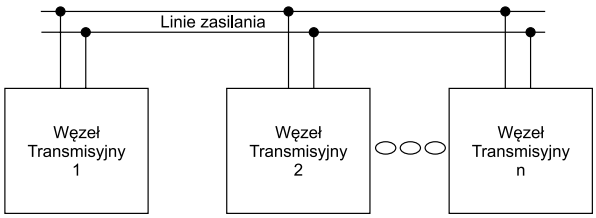


Rysunek 1. Schemat ideowy węzła komunikacyjnego PLC

ganego przez moduł komunikacyjny i sterujący.

Tor komunikacyjny to nie tylko węzły komunikacyjne, ale również pozwalające na ich połączenie medium transmisyjne, którym w tym wypadku są obwody zasilania. Linie zasilania umożliwiają budowę toru transmisyjnego, który ma charakter sieci, dlatego sposób łączenia węzłów za pomocą medium musi być zgodny z obowiązującą topologią, która jest zbiorem zasad fizycznego dołączania węzłów do sieci, jak również określa reguły komunikacji węzłów poprzez dane medium.

Topologią wykonanego w ramach niniejszego projektu toru transmisyjnego jest popularnie stosowana między innymi w sieciach przemysłowych magistrala. Topologia ta charakteryzuje się tym, że wszystkie elementy sieci podłączone są do jednego, współdzielonego przez nie nośnika danych, tutaj przewodu zasilania. Sieć o takiej topologii umożliwia tylko jedną transmisję w danym momencie. Sygnał nadany przez jedną ze stacji odbierany jest przez wszystkie pozostałe, jednakże tylko stacja, do której pakiet został zaadresowany, interpretuje go.



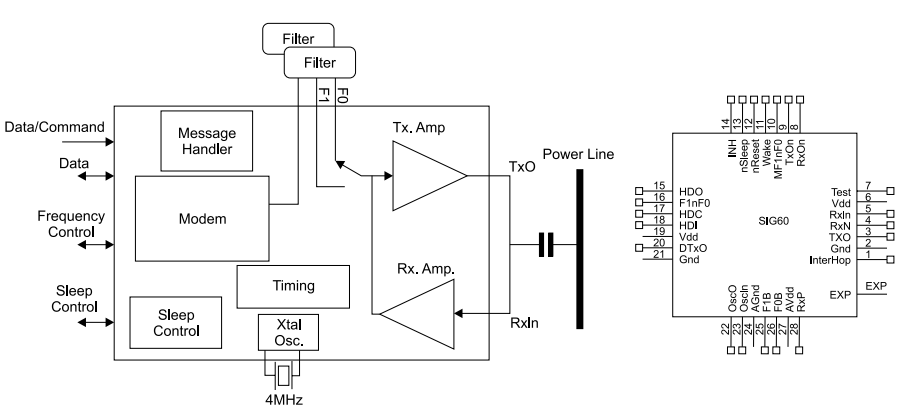
Rysunek 2. Struktura sieci PLC w topologii magistrali

Strukturę modelu sieci PLC o charakterze magistrali przedstawiono na **rysunku 2**. Umożliwia on stworzenie wszechstronnego rozwiązania będącego kompleksowym systemem przesyłowym wykorzystującym linie zasilania.

Dobór podzespołów

Moduł sterujący składa się z układu mikrokontrolera oraz elementów peryferyjnych zapewniających jego poprawną pracę. Myśląc perspektywicznie, warto wybrać układ, który oprócz realizacji funkcji przewidzianych w ramach współpracy modułu sterującego z modułem komunikacyjnym będzie w stanie pełnić w przyszłości również większą liczbę zadań. Zgodnie z tym założeniem na potrzeby projektu wybrany został wszechstronny mikrokontroler STM32F103C8T6 firmy STMicroelectronics oparty na rdzeniu ARM Cortex-M3. Gwarantuje on dużą wydajność (możliwość taktowania do 72 MHz) oraz uniwersalność, o której świadczą jego zasoby sprzętowe.

Kolejnym kluczowym dla urządzenia elementem jest moduł komunikacyjny. Głównym jego podzespołem jest układ transmisyjny. Jego wybór ma fundamentalne znaczenie, gdyż jest to element określający w największym stopniu funkcjonalność całego urządzenia. Wśród obecnych na rynku producentów oferujących rozwiązania w zakresie zintegrowanych układów nadawczo-odbiorczych do transmisji danych po liniach zasilania są takie firmy jak Maxim (MAX298x, MAX299x), Cypress Semiconductor (CY8CPLC10), STMicroelectronics (*ST757x*, ST758x, ST759x), Echelon (PL3120, PL3150, PL3170), Semitech Semiconductor (SM2200). Oprócz wyżej wymienionych producentów, dla których technologia PLC jest tylko jednym z wielu obszarów zainteresowania, istnieją również firmy, które traktują tę dziedzinę priorytetowo i jest to dla nich główny lub nawet jedyny obszar badań. Do takich firm należy Yamar Electronics, która w swojej ofercie posiada całą gamę układów do transmisji po liniach zasilania (SIG60, ISL40, DCB500, DCAN250) wykorzystujących zarówno napięcie przemienne, jak i stałe.



Rysunek 3. Struktura wewnętrzna układu SIG60 oraz jego wyprowadzenia

W zrealizowanym urządzeniu zastosowane zostały układy, których producentem jest właśnie firma Yamar Electronics – konkretnie układy SIG60. Strukturę wewnętrzną układu SIG60 oraz jego wyprowadzenia na **rysunku 3**. Układ składa się z interfejsu komunikacyjnego, mechanizmu zegarowego, modemu, modułu zarządzającego poborem energii, wzmacniaczy wspomagających odbiór i nadawanie danych.

Głównym elementem układu SIG60 jest modem. Zgodnie ze swoją nazwą jego rolą jest modulacja odebranych przez interfejs komunikacyjny danych cyfrowych do postaci sygnału transmitowanego dalej poprzez linie zasilania, jak również demodulacja sygnału pochodzącego z obwodów zasilania do postaci użytkowej przekazywanej następnie do interfejsu użytkownika.

Zaimplementowany w układzie SIG60 moduł zarządzania energią umożliwia pracę w trybie o obniżonym poborze prądu. Odbywa się to poprzez uśpienie układu (wyprowadzenie *nSleep*), który przebywając w tym stanie, pobiera prąd rzędu 80 µA. W tym trybie układ budzi się każdorazowo po czasie 32 ms, po czym sprawdza stan magistrali i w wypadku braku transmisji, ponownie przechodzi w stan uśpienia.

Kolejnym elementem układu SIG60 jest interfejs komunikacyjny, który umożliwia konfigurowanie i obsługę układu. Interfejs

Wykaz elementów

Rezystory: (SMD, 0805)
R1, R2, R5…R9: 10 kΩ
R3: 600 Ω
R4: 100 Ω
R10: 1 MΩ
Kondensatory: (SMD, 0805):
C1…C5, C23: 100 nF
C6, C7: 22 pF
C8, C9: 10 pF
Półprzewodniki:
IC1: STM32F103C8T6
D1: dioda prostownicza (SMD, SOD80)
LED1, LED2: dioda LED (SMD, 0805)
Inne:
SIG60: moduł SIG60
Q1: rezonator kwarcowy 8 MHz (SMD)
Q2: rezonator kwarcowy 32,768 kHz (SMD)
S1: przełącznik suwakowy
SV1: złącze kołkowe proste 2×10 pin SMD
EXTENSION_BOARD: złącze kołkowe proste 2×7 pin SMD
INPUT: gniazdo zasilania

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

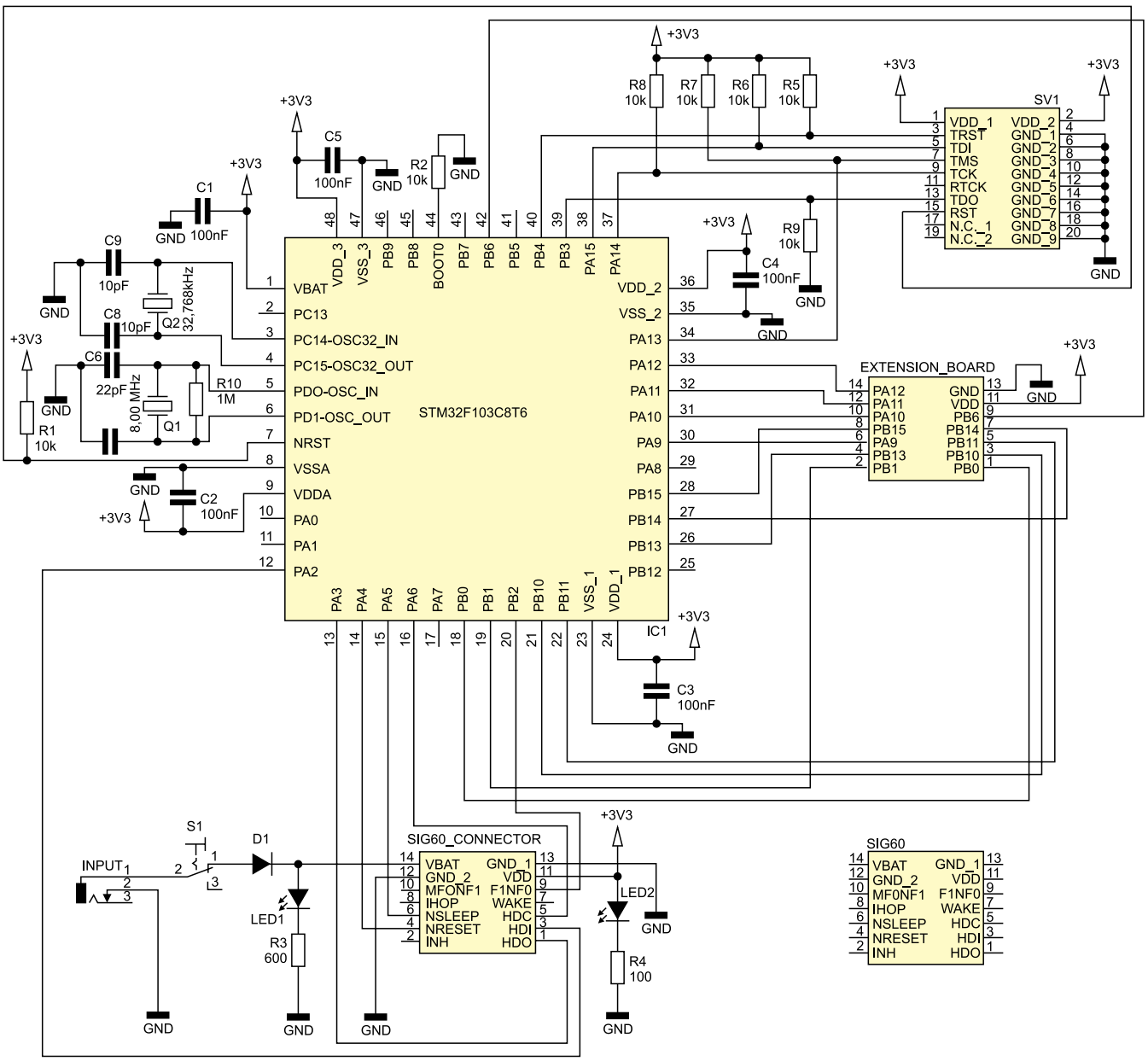
komunikacyjny jest oparty na standardzie transmisyjnym UART (wyprowadzenia układu HDO, HDI) o predefiniowanych parametrach: baudrate 19200 bps, 8 bitów danych, 1 bit stopu, brak parzystości, brak handshakingu. Wybór pomiędzy transmisją danych i konfigurowaniem odbywa się poprzez zmianę stanu logicznego na wyprowadzeniu HDC (stan niski – konfigurowanie, stan wysoki – transmisja danych).

Tryb konfigurowania pozwala na dostęp do dwóch 8-bitowych wewnętrznych rejestrów układu. Procedura zapisu do rejestru polega na wysterowaniu linii HDC stanem niskim, a następnie przesłaniu dwóch bajtów – adresu rejestru i danych do jego wypełnienia. Rejestr Control Register 0 służy do aktywacji opcji dodatkowych np. trybu *loopback mode* i funkcji autousypiania układu po 8 s. Rejestr *Control Register 1* przechowuje informacje o ustawionych parametrach układu – prędkości transmisji UART (możliwe ustawienia: 9600, 19200, 38400, 57600, 115200) oraz częstotliwości nośnej sygnału PLC. Spośród siedmiu dostępnych częstotliwości (1,75 MHz; 4,5 MHz; 5,5 MHz; 6 MHz; 6,5 MHz; 10,5 MHz lub 13 MHz) wybrać należy dwie (zwane jako F0 i F1), między którymi przełączać się można w trakcie pracy układu za pomocą wyprowadzenia F1nF0 (stan wysoki – częstotliwość F1, stan niski – częstotliwość F0).

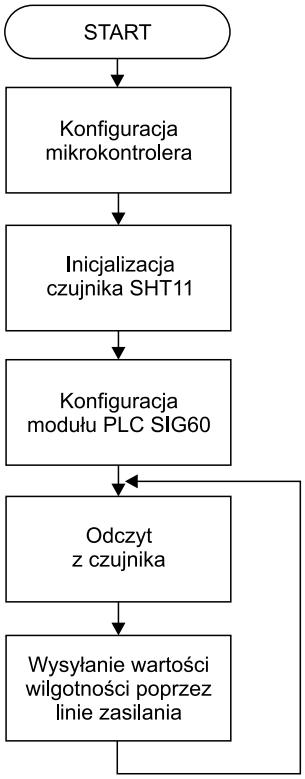
Tryb pracy normalnej umożliwia obsługę transmisji danych. Polega ona na przekazywaniu wszystkich odebranych przez interfejs UART bajtów do modemu, który następnie wysyła je poprzez linie zasilania. Odwrotna sytuacja jest analogiczna – każdy bajt odebrany z linii zasilania jest przekazywany przez modem na interfejs UART.

Interesującą cechą układu SIG60 jest możliwość automatycznej zmiany częstotliwości po wykryciu kolizji na linii transmisyjnej. Aktywacja tej funkcji odbywa się poprzez ustawienie stanu wysokiego na wyprowadzeniu *InterHop*. Wystąpienie kolizji skutkuje przełączeniem przez układ częstotliwości modulacji z wartości F1 na F0 bądź odwrotnie.

Transmisja danych między układami SIG60 jest zapewniona jedynie na podstawowych warstwach modelu OSI/ISO, co oznacza, że układy te nie mają mechanizmu regulującego dostęp do medium. W celu uniknięcia efektu występowania kolizji ramek o tej samej częstotliwości modulacji wysyłanych przez różne moduły, producent zaleca, aby układy pracowały zgodnie z modelem master–slave. Model ten polegać może na nadaniu adresów wszystkim węzłom w sieci oraz wyznaczeniu jednego węzła jako układu nadrzędnego (master), który zarządzałby dostępem do medium poprzez nadawanie pozostałym węzłom (slaves) prawa do transmisji (w danej chwili tylko jeden węzeł może nadawać).



Rysunek 4. Schemat elektryczny węzła komunikacyjnego



Rysunek 5. Schemat blokowy programu węzła nadawczego

Schemat aplikacyjny dla układu SIG60 przewiduje zastosowanie szeregu elementów dodatkowych, których połączenie z układem jest wymagane do jego poprawnej pracy. Firma Yamar Electronics, chcąc ułatwić rozpoczęcie pracy z jej produktem oraz skrócić czas budowy urządzeń na nim opartych, przygotowała specjalne moduły będące zestawami ewaluacyjnymi. Umożliwiają one wykorzystanie pełnej funkcjonalności układu SIG60 i są przeznaczone głównie do zastosowań prototypowych. Płytka modułu SIG60 oprócz układu nadawczo-odbiorczego została wyposażona między innymi w tor sprzęgający układ z liniami zasilania, filtry częstotliwościowe, układ taktujący, gniazda sygnałowe oraz zasilacz. Jej schemat blokowy i ideowy są dostępne w dokumencie *SIG60 Module Manual* dostępnym na stronie internetowej firmy Yamar [6].

Budowa

Urządzenie składa się z trzech modułów: sterującego, komunikacyjnego oraz zasilania. Do realizacji modułu sterującego wybrano mikrokontroler STM32F103C8T6, natomiast funkcję komunikacyjną pełni moduł SIG60, który dzięki zintegrowanemu zasilaczowi może ponadto grać rolę modułu zasilającego. Wybór tych komponentów sprawia, iż urządzenie węzła komunikacyjnego to przede wszystkim mikrokontroler połączony poprzez interfejs komunikacyjny z modułem SIG60.

Schemat elektryczny bloku sterowania pokazano na **rysunku 4**. Do nóżek zasilających mikrokontrolera STM32F103C8T6 (VDD1...3, VBAT, VDDA) doprowadzono napięcie zasilania 3,3 V z przetwornicy. Analogicznie, masy układu (VSS1...3, VSSA) połączono z potencjałem ujemnym. W celu zapewnienia dobrej jakości napięcia zasilania mikrokontrolera do każdej pary wyprowadzeń zasilających dołączono kondensator ceramiczny 100 nF (C1...C5).

Aby wykorzystać w pełnym zakresie potencjału, układu zegarowego mikrokontrolera zastosowano dwa rezonatory zewnętrzne (Q1, Q2). Pierwszy, o częstotliwości 8 MHz, umożliwia zwielokrotnianie częstotliwości taktowania rdzenia oraz peryferii, natomiast drugi (32,768 kHz) jest wymagany do pracy zegara RTC.

Programowanie mikrokontrolera odbywa się za pomocą interfejsu JTAG. Jego linie sygnałowe (PA13...PA15, PB3, PB4, PB6, NRST) zostały wyprowadzone na gnieździe SV1. Do każdej z linii dołączono opornik podciągający 10 kΩ (R5...R8) lub ściągający (R9). Ich włączenie jest opcjonalne (oprócz linii NRST), gdyż producent zadbał o ustalenie odpowiedniego poziomu na linii interfejsu JTAG wewnątrz mikrokontrolera, dzięki czemu użycie dodatkowych oporników nie jest niezbędne.

Mikrokontroler STM32F103C8T6 ma możliwość wykonywania kodu programu z pamięci wewnętrznej FLASH lub SRAM, lub z pamięci zewnętrznej o interfejsie równoległym. Wyboru aktywnej pamięci dokonuje się poprzez ustalenie stanu na wyprowadzeniach BOOT0 oraz BOOT1. W wypadku korzystania z wewnętrznej pamięci FLASH należy dołączyć wyprowadzenie BOOT0 do masy.

Połączenie modułu komunikacyjnego SIG60 z układem mikrokontrolera wymaga użycia interfejsu UART oraz kilku linii ogólnego przeznaczenia. Do tego celu zastosowano wyprowadzenia PA2, PA3 (odpowiednio *USART2_TX* oraz *USART2_RX*), PA4...PA6, PB2. Napięcie wejściowe (może to być napięcie z linii zasilania służących do transmisji danych) jest dołączone do gniazda *INPUT*. Poprzez przełącznik S1 oraz diodę D1 (zabezpieczenie przed odwrotną polaryzacją) napięcie doprowadzone jest do wyprowadzenia VBAT modułu SIG60 (*SIG60_CONNECTOR*), gdzie dalej jest obniżane przez przetwornicę do wartości 3,3 V. Napięcie to wyprowadzono na gniazdo *SIG60_CONNECTOR*, dzięki czemu służy ono również do zasilania mikrokontrolera. Obecność napięcia wejściowego oraz praca przetwornicy 3,3 V są sygnalizowane przez świecenie diod LED1 i LED2.

W celu umożliwienia dołączenia do mikrokontrolera dodatkowych elementów peryferyjnych przewidziano użycie dodatkowego złącza sygnałowego (*EXTENSION_BOARD*). Zostało na nim wyprowadzone zasilanie oraz 12 linii portów I/O (PA9...PA12, PB0, PB1, PB6, PB10, PB11, PB13...PB15), które mogą pełnić funkcję linii ogólnego przeznaczenia, przetwornika A/C, timera oraz interfejsów komunikacyjnych SPI, I²C, CAN, USB, USART.

Listing 1. Konfiguracja zegarów mikrokontrolera STM32

```
ErrorStatus HSEStartUpStatus;

RCC_DeInit();
RCC_HSEConfig(RCC_HSE_ON);
HSEStartUpStatus = RCC_WaitForHSEStartUp();
if(HSEStartUpStatus == SUCCESS)
{
  RCC_HCLKConfig(RCC_SYSCLK_Div1);
  RCC_PCLK2Config(RCC_HCLK_Div1);
  RCC_PCLK1Config(RCC_HCLK_Div2);
  RCC_PLLConfig(RCC_PLLSource_HSE_Div1, RCC_PLLMul_5);
  RCC_PLLCmd(ENABLE);
  while(RCC_GetFlagStatus(RCC_FLAG_PLLRDY) == RESET)
  {
  }
  RCC_SYSCLKConfig(RCC_SYSCLKSource_PLLCLK);
  while(RCC_GetSYSCLKSource() != 0x08)
  {
  }
}
RCC_APB2PeriphClockCmd(RCC_APB2Periph_GPIOA | RCC_APB2Periph_GPIOB, ENABLE);
RCC_APB1PeriphClockCmd(RCC_APB1Periph_USART2, ENABLE);
```

Listing 2. Konfigurowanie interfejsu USART

```
//A.2 - USART2 TX
GPIO_InitStructure.GPIO_Pin = GPIO_Pin_2;
GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz;
GPIO_InitStructure.GPIO_Mode =  GPIO_Mode_AF_PP;
GPIO_Init(GPIOA, &GPIO_InitStructure);
//A.3 - USART2 RX
GPIO_InitStructure.GPIO_Pin = GPIO_Pin_3;
GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz;
GPIO_InitStructure.GPIO_Mode =  GPIO_Mode_IN_FLOATING;
GPIO_Init(GPIOA, &GPIO_InitStructure);
USART_InitStructure.USART_BaudRate = 19200;
USART_InitStructure.USART_WordLength = USART_WordLength_8b;
USART_InitStructure.USART_StopBits = USART_StopBits_1;
USART_InitStructure.USART_Parity = USART_Parity_No ;
USART_InitStructure.USART_HardwareFlowControl = USART_HardwareFlowControl_None;
USART_InitStructure.USART_Mode = USART_Mode_Rx | USART_Mode_Tx;
USART_Init(USART2, &USART_InitStructure);
USART_Cmd(USART2, ENABLE);
```

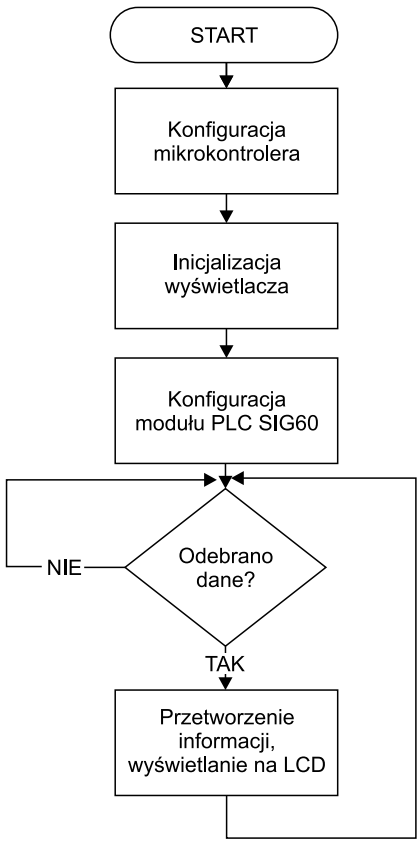
Przykładowy system PLC

Duża funkcjonalność wykonanych węzłów komunikacyjnych PLC pozwala na użycie ich do różnorodnych zastosowań wymagających transmisji danych. W ramach testów urządzenia wykonany został prosty system czujnikowy zbudowany z dwóch węzłów transmisyjnych, które połączono za pomocą linii zasilania, dla których źródłem napięcia był akumulator 12 V_{DC}. Pierwszy węzeł wyposażono w czujnik temperatury i wilgotności Sensirion SHT11, natomiast do drugiego węzła dołączono wyświetlacz LCD EA DOGM128 firmy Electronic Assembly. Taka konfiguracja systemu pozwoliła na wykorzystanie węzła nr 1 do pozyskiwania w wyznaczonym miejscu pomiarów podstawowych parametrów środowiskowych, a następnie przesyłanie ich za pomocą linii napięciowej 12 V do oddalonego węzła nr 2, który wizualizował pomiary na wyświetlaczu LCD.

Jest to przykład bardzo nieskomplikowanego systemu, ponieważ nie tylko składa się z minimalnej liczby węzłów, ale również komunikacja odbywa się w najprostszej formie, a więc tylko w jednym kierunku (węzeł z czujnikiem wysyła dane, węzeł z wyświetlaczem odbiera). Zaletą tego rozwiązania jest brak konieczności implementacji adresacji oraz sposobu dostępu do medium.

Schemat blokowy programu węzła nadawczego przedstawiono na **rysunku 5**. Obejmuje on konfigurowanie mikrokontrolera, inicjalizację czujnika SHT11, konfigurowanie modułu SIG60, a następnie cykliczny odczyt pomiarów z czujnika i wysyłanie ich poprzez obwody zasilania.

W pierwszej kolejności następuje konfiguracja zegarów mikrokontrolera, która obejmuje włączenie taktowania z zewnętrznego źródła będącego kwarcem o częstotliwości 8 MHz. Następnie jest aktywowana pętla PLL, której zwielokrotnienie ustawione zostaje na wartość 5, co w efekcie daje częstotliwość 40 MHz. Włączone zostają ponadto



Rysunek 6. Schemat blokowy programu węzła odbiorczego

Listing 3. Konfigurowanie wyprowadzeń sterujących modułem SIG60 oraz inicjalizacja modułu

```
//A.4 - SIG60 RESET,A.5 - SIG60 SLEEP, A.6 - SIG60 HDC
GPIO_InitStructure.GPIO_Pin = GPIO_Pin_4 | GPIO_Pin_5 | GPIO_Pin_6;
GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz;
GPIO_InitStructure.GPIO_Mode =  GPIO_Mode_Out_PP;
GPIO_Init(GPIOA, &GPIO_InitStructure);

//B.2 - SIG60 F1NF0
GPIO_InitStructure.GPIO_Pin = GPIO_Pin_2;
GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz;
GPIO_InitStructure.GPIO_Mode =  GPIO_Mode_Out_PP;
GPIO_Init(GPIOB, &GPIO_InitStructure);

void SIG60_configurtion(void)
{
  GPIO_SetBits(GPIOA, GPIO_Pin_4);    //reset_pin
  GPIO_SetBits(GPIOA, GPIO_Pin_5);    //sleep_pin
  GPIO_SetBits(GPIOA, GPIO_Pin_6);    //HDC_pin
  GPIO_SetBits(GPIOB, GPIO_Pin_2);    //F1NF0_pin
}

…

SIG60_configurtion();
```

Listing 4. Konfigurowanie wyprowadzeń interfejsu czujnika SHT11 oraz inicjalizacja czujnika

```
  //A.11 - sensor (SHT11) clock line
  GPIO_InitStructure.GPIO_Pin = GPIO_Pin_11;
  GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz;
  GPIO_InitStructure.GPIO_Mode = GPIO_Mode_Out_PP;
  GPIO_Init(GPIOA, &GPIO_InitStructure);

  //A.10 - sensor (SHT11) data line
  GPIO_InitStructure.GPIO_Pin = GPIO_Pin_10;
  GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz;
  GPIO_InitStructure.GPIO_Mode = GPIO_Mode_Out_PP;
  GPIO_Init(GPIOA, &GPIO_InitStructure);
  …
  SHT11_configuration();
```

Listing 5. Pętla główna programu węzła nadawczego

```
while(1)
{
   for(delay = 0; delay <= 300000; delay++) {}
   humidity = SHT11_Read_Humidity();
   USART_SendData(USART2, humidity);
   while(USART_GetFlagStatus(USART2, USART_FLAG_TXE) == RESET)
   {
   }
}
```

Listing 6. Konfigurowanie interfejsu SPI, inicjalizacja wyświetlacza

```
//B.13, B.15 - SPI2 (SCK, MOSI) for LCD (EA DOGM128)
GPIO_InitStructure.GPIO_Pin = GPIO_Pin_13 | GPIO_Pin_15;
GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz;
GPIO_InitStructure.GPIO_Mode = GPIO_Mode_AF_PP;
GPIO_Init(GPIOB, &GPIO_InitStructure);

//B.14 - LCD controll (data/instructions)
GPIO_InitStructure.GPIO_Pin = GPIO_Pin_14;
GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz;
GPIO_InitStructure.GPIO_Mode = GPIO_Mode_Out_PP;
GPIO_Init(GPIOB, &GPIO_InitStructure);

//B.11 - LCD controll (reset)
GPIO_InitStructure.GPIO_Pin = GPIO_Pin_11;
GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz;
GPIO_InitStructure.GPIO_Mode = GPIO_Mode_Out_PP;
GPIO_Init(GPIOB, &GPIO_InitStructure);
SPI_InitStructure.SPI_Direction = SPI_Direction_1Line_Tx;
SPI_InitStructure.SPI_Mode = SPI_Mode_Master;
SPI_InitStructure.SPI_DataSize = SPI_DataSize_8b;
SPI_InitStructure.SPI_CPOL = SPI_CPOL_Low;
SPI_InitStructure.SPI_CPHA = SPI_CPHA_1Edge;
SPI_InitStructure.SPI_NSS = SPI_NSS_Soft;
SPI_InitStructure.SPI_BaudRatePrescaler = SPI_BaudRatePrescaler_2;
SPI_InitStructure.SPI_FirstBit = SPI_FirstBit_MSB;
SPI_InitStructure.SPI_CRCPolynomial = 7;
SPI_Init(SPI2, &SPI_InitStructure);
SPI_Cmd(SPI2, ENABLE);
…
LCD_Init();
```

sygnały taktujące dla wykorzystanych peryferii: GPIOA, GPIOB, USART2 (**listing 1**).

W następnym etapie działania programu skonfigurowane zostają nóżki mikrokontrolera PA2 oraz PA3 do pracy w trybie interfejsu UART, po czym zostają ustawione parametry samego interfejsu zgodnie z tymi obsługiwanymi przez moduł SIG60 (**listing 2**).

Kolejną fazą jest przygotowanie do pracy modułu SIG60. Skonfigurowane zostają wyprowadzenia łączące mikrokontroler z modułem SIG60 (PA4, PA5, PA6, B2), a następnie wystawiony zostaje na nich stan wysoki, co skutkuje zresetowaniem układu, wyjściem ze stanu uśpienia, wyborem częstotliwości modulacji F1 (6,5 MHz) oraz przejściem do trybu transmisji danych (**listing 3**). W celu odczytu pomiarów z czujnika został zaimplementowany kod do jego poprawnej obsługi. Czynność przygotowania czujnika do pracy obejmuje konfigurację pinów, do których został on dołączony (PA10, PA11) oraz wywołanie funkcji ustawiającej parametry pomiarów (**listing 4**). Następnie rozpoczyna się proces wykonywania przewidzianych przez węzeł czynności, które realizowane są cyklicznie w nieskończonej pętli *while(1)*. Co określony odstęp czasu odczytywana jest z czujnika wartość zmierzonej wilgotności, po czym wysyłana jest ona interfejsem UART do modułu SIG60, który modulując otrzymane dane, transmituje je poprzez linie zasilania 12 V (**listing 5**).

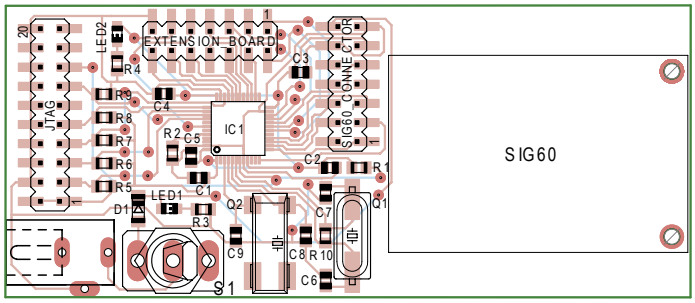
Znaczny fragment kodu węzła odbiorczego pokrywa się z programem węzła nadawczego. Wspólna dla obu węzłów jest konfiguracja zegarów oraz interfejsu UART, jak również sposób przygotowania do pracy modułu SIG60. Elementem nowym w module odbiorczym jest obsługa wyświetlacza LCD. Pełny schemat blokowy programu węzła odbiorczego zamieszczono na **rysunku 6**.

Wyświetlacz LCD w module odbiorczym ma interfejs komunikacyjny SPI. Jego sprzętową realizację w mikrokontrolerze pełnią między innymi wprowadzone na gnieździe *EXTENSION_BOARD* nóżki PB13, PB14 oraz PB15. Dodatkowo, wyświetlacz wymaga sterowania poprzez dwie kolejne linie, które zostały podłączone do pinów PB11 oraz PB14. Polecenia konfigurowania wszystkich wyprowadzeń wyświetlacza LCD, interfejsu SPI oraz wywołanie funkcji inicjalizacji przedstawiono na **listingu 6**.

Działanie węzła odbiorczego polega na oczekiwaniu na dane z modułu SIG60. Każdorazowo po otrzymaniu bajtu danych poprzez interfejs UART informacja jest przetwarzana, a następnie zinterpretowana wartość wilgotności aktualizowana jest na wyświetlaczu LCD (**listing 7**).

Montaż

Schemat montażowy toru transmisji PLC pokazano na **rysunku 7**. Projekt płytki został zoptymalizowany pod kątem małych gabarytów urządzenia. Z tego względu zdecydowano się na zastosowanie głównie elementów w obudowach SMD i umieszczenie ich blisko siebie, co wydatnie przyczyniło się do małej powierzchni płytki (90 mm×39 mm).



Rysunek 7. Rozmieszczenie elementów na płytce drukowanej węzła komunikacyjnego

Listing 7. Pętla główna programu węzła odbiorczego

```
while(1)
{
   if(USART_GetFlagStatus(USART2, USART_FLAG_RXNE) == SET)
   {
      UART_Buffer = USART_ReceiveData(USART2);
      Measurement[0] =  UART_Buffer/100;
      Measurement[1] = (UART_Buffer - 100*Measurement[0])/10;
      Measurement[2] =  UART_Buffer - 100*Measurement[0] - 10*Measurement[1];
      LCD_SetXY(80,6);
      LCD_PutChar(Measurement[0] + 48);
      LCD_PutChar(Measurement[1] + 48);
      LCD_PutChar(Measurement[2] + 48);
   }
}
```

Montaż płytki najwygodniej rozpocząć od przylutowania mikrokontrolera. W następnej kolejności należy przejść do przytwierdzenia pozostałych elementów powierzchniowych: kondensatorów, rezystorów, kwarców oraz diod LED. Kolejna faza montażu to elementy mechaniczne: gniazdo zasilające, szpilkowe gniazda sygnałowe, przełącznik. Ostatnim etapem jest umieszczenie na płytce modułu SIG60 i połączenie go taśmą sygnałową z przeznaczonym mu gniazdem.

Podsumowanie

Technologia PLC to niezwykle obiecująca dziedzina telekomunikacji. Duży potencjał oraz liczne zalety powinny skutkować jej dalszym rozwojem, co może przyczynić się do dalszego dopracowywania i optymalizacji tej metody transmisji danych. Na podstawie przeprowadzonych testów działania dla wykonanych węzłów komunikacyjnych i systemu transmisyjnego na nich opartego można stwierdzić, że opracowany tor transmisyjny może być solidną podstawą dla różnorodnych zastosowań, nie tylko amatorskich, ale również komercyjnych. Jednocześnie trzeba mieć jednak świadomość faktu, iż na obwodach zasilania niejednokrotnie występują różnorodne zakłócenia, co może skutkować wprowadzaniem błędów do transmitowanych danych, które w efekcie dostarczane są z przekłamaniami. Z tego powodu przed podjęciem decyzji o zastosowaniu linii zasilania jako medium przesyłu danych należy odpowiedzieć sobie na pytanie, czy środowisko, w którym będzie pracować taki system, nie zagraża jego poprawnej pracy.

Autor składa podziękowania Panom: Łukaszowi Konarskiemu z firmy Embedded Systems Design Center oraz Yaira Maryanka z firmy Yamar za pomoc w realizacji projektu.

Szymon Panecki
szymon.panecki@pwr.wroc.pl

Bibliografia

[1] Maryanka Y.: Wiring Reduction by Battery Power Line Communication, Maj 2000.
[2] Pavlidou N., VINCK A., YAZDANI J., HONARY B.: Power Line Communications: State of the Art And Future Trends, Kwiecień 2003.
[3] www.lcd-module.com: DOGM Graphic Series.
[4] www.sensirion.com: Datasheet SHT1x.
[5] www.st.com: Datasheet STM32, STM-32F10xxx hardware development: getting started.
[6] www.yamar.com SIG60 Module Manual, 2009, SIG60 – UART over Powerline, for AC/DC-BUS Network, 2009.