W bascomowym „kąciku" staramy się w miarę przystępnie przedstawiać rozwiązania problemów napotykanych przez naszych Czytelników podczas pisania programów w Bascomie. Rubryka ta powstała z myślą o rozwiązywaniu trudności, jakie najczęściej napotykają programiści, zatem zachęcamy wszystkich Czytelników do zgłaszania przeszkód, na jakie się natknęli podczas tworzenia własnych programów.

Obsługa RS485 w Bascomie, część 2

Interfejs RS485 cieszy się coraz większą popularnością, którą zawdzięcza głównie możliwości przesyłania danych na duże odległości z dużymi prędkościami. W artykule przedstawiamy przykłady w Bascomie ilustrujące wymianę danych za pomocą tego interfejsu i specjalnie opracowanego protokołu.

W przypadku *Slave 2*, tak jak w przypadku *Slave 1*, dane z magistrali RS485 są odbierane w przerwaniu i ładowane do 15-bajtowego bufora odbiorczego. Na **list. 2** przestawiono program obsługi układu wykonawczego (*Slave 2*). Timer 1 skonfigurowany został jako generator sygnału PWM o rozdzielczości 8 bitów i częstotliwości ok. 245 Hz, co w przypadku wentylatora jest wystarczające do sterowania jego prędkością. Stała *adr* ma wartość „8", która jest adresem tegoż układu. Po uruchomieniu układu, *Slave 2* pracuje w trybie odbiornika, a przekaźniki oraz wentylator (PWM=0) są wyłączone. Pętla główna programu zajmuje się interpretacją czterech powyżej opisanych komend i zapewnieniem odpowiedniej reakcji na nie układów dołączonych do mikrokontrolera. W porównaniu do *Slave 1*, instrukcje interpretujące komendy dla *Slave 2* są bardziej rozbudowane, gdyż mają za zadanie zinterpretować 4 rozkazy, a nie 1 jak to było w przypadku *Slave 1*. Jeżeli moduł *Slave* odebrał znak „BS", to następnie czeka na adres (tak samo jak to było w przypadku *Slave 1*). Jeśli otrzymany adres jest z zakresu od 0 do 99, a w tym przypadku równy „8", to następuje dalsza interpretacja otrzymywanych znaków. W przeciwnym razie komenda zostaje odrzucona, gdyż przeznaczona jest dla układu o innym adresie. W tym przypadku układ będzie oczekiwał następnej komendy rozpoczynającej się znakiem „BS" (*Back Space*). Jeśli otrzymany adres zgadza się z adresem tego układu, sprawdzane jest, czy znak po adresie jest znakiem „r" (komenda odczytu), czy znakiem „w" (komenda zapisu). W przypadku, gdy otrzymany znak to „r" oraz kolejny znak to „p", zostanie wysłany – po przełączeniu układu w tryb nadajnika – stan rejestru wypełnienia przebiegu PWM, po czym układ ponownie przejdzie w stan odbiornika. W przypadku, gdy po „r" zostanie odebrane „o", zostanie wysłany – po przełączeniu układu

List. 2. Program sterujący układem wykonawczym (*Slave 2*)

```
'Przykład drugiego urzadzenia slave sieci RS485.
'Urzadzenie steruje dwoma przekaznikami oraz wypełnieniem jednego 8 bitowego sygnału PWM
'na rzadanie mastera
'Urzadzenie ma mozliwosc otrzymania od mastera nowych ustawien przekaznikow oraz pwm.
'Ma takze mozliwosc wysłania do mastera stanów przekazników oraz wypelnienia pwm
'Adres tego urzadzenia wynosi 8

$regfile = "m8def.dat"              'informuje kompilator o pliku dyrektyw wykorzystywanego
                                    'mikrontrolera
$crystal = 8000000                  'informuje kompilator o czestotliwosci rezonatora kwarcowego
$baud = 9600                        'informuje kompilator o predkosci transmisji RS232

Config Pinb.1 = Output              'linia pb.1 jako wyjscie
Config Pinb.2 = Output              'linia pb.2 jako wyjscie
Config Pinb.3 = Output              'linia pb.3 jako wyjscie
Config Pind.2 = Output              'linia pd.2 jako wyjscie
Config Serialin = Buffered , Size = 15
                                    'konfiguracja by interfejs rs232 uzywal przy odbiorze transmisji
                                    'buforowej (bufor o wielkosci 15 znakow)
Config Timer1 = Pwm, Pwm = 8, Compare A Pwm = Clear Down, Compare B Pwm = Disconnect, Prescale = 64
                                    'konfiguracja timer1 jako generator pwm o rozdzielczosci 8 bit
                                    'generowany przebieg PWM o czestotliwosci ok 245 Hz
Const Adr = 8                       'adres ukladu - od 0 do 99
Dim Zn As String * 1                'zmienna przecowujaca odebrany znak z rs232
Dim Zm_lan As String * 3            'zmienna w ktorej skladany jest w calosc otrzymany adres
                                    'oraz dane o szerokosci impulsu PWM
Dim Adres As Byte                   'zmienna ktora przechowuje przekonwertowany na dziesietnie
                                    'otrzymany adres
Dim Il_zn As Byte                   'zmienna liczy ilosc otrzymanych znakow przy skladaniu adresu
Dim St_pk1 As Byte                  'przechowuje tymczasowo stan przekaznika pk1
Dim St_pk2 As Byte                  'przechowuje tymczasowo stan przekaznika pk2
Dim Wart_pwm As Word                'przechowuje otrzymana od mastera wartosc pwm

Re_de Alias Portd.2                 'przypisanie aliasu linii pd.2, re_de steruje kierunkiem
                                    'transmisji konwertera
Pk1 Alias Portb.2                   'przypisanie aliasu pk1 linii pb.2
Pk2 Alias Portb.3                   'przypisanie aliasu pk2 linii pb.3

Enable Interrupts                   'odblokowanie globalnych przerwan
Reset Re_de                         'przelaczenie konwertera rs485 na odbiornik
Pwm1a = 0                           'szerokosc impulsu pwm 0
Reset Pk1                           'wylaczenie pk1
Reset Pk2                           'wylaczenie pk2

Do                                  'poczatek petli programu
  Zn = Inkey()                      'odczyt znaku z portu rs232
  If Zn = Chr(8) Then               'jesli otrzymany znak to BS (backspace - kod ascii 8) to
    Zm_lan = ""                     'czysc zmienna zn_lan
    Il_zn = 0                       'oraz zeruj zmienna il_zn
    Do                              'poczatek petli do-loop
      Zn = Inkey()                  'odczyt znaku z portu rs232
      If Zn >= "0" And Zn <= "9" Then     'jesli otrzynany znak jest cyfra (0..9) to
        Incr Il_zn                  'zwieksz o jeden il_zn
        Zm_lan = Zm_lan + Zn        'dodaj otrzymany znak cyfry do zmiennej zm_lan
      Else                          'w przeciwnym razie
        If Zn > Chr(0) Then         'jesli otrzymany znak ma kod ascii wiekszy od 0 to
          Exit Do                   'opusc petle do-loop
        End If
      End If
    Loop                            'koniec wewnetrznej petli do-loop
    If Il_zn > 0 And Il_zn < 3 Then       'jesli il_zn wynosi 1 lub 2 to
      Adres = Val(zm_lan)           'zamien znakowa wartosc otrzymanego adresu na postac liczbowa
                                    'i umiesc ja w adres
      If Adres = Adr Then           'jesli otrzymany adres od mastera rowna sie 8
                                    '(adres tego urzadzenia) oraz
        If Zn = "r" Then            'jesli nastepny otrzymany znak to "r"
          Do                        'poczatek wewnetrznej petli do-loop
            Zn = Inkey()            'odczyt znaku z portu rs232
          Loop Until Zn > Chr(0)    'opusc petle gdy otrzymany znak wiekszy od kodu 0 ascii
          If Zn = "p" Then          'jesli nastepny otrzymany znak to "p" to
            Do                      'poczatek wewnetrznej petli do-loop
              Zn = Inkey()          'odczyt znaku z portu rs232
            Loop Until Zn = Chr(13)       'jesli otrzymany znak to cr - enter (kod ascii 13) to
            Set Re_de               'przelaczenie konwertera rs485 na nadajnik
            Waitus 200              'czekaj 200 us
            Print Pwm1a             'wyslij wartosc wypelnienia przebiegu pwm
            Waitms 1                'czekaj 1 ms
            Reset Re_de             'przelaczenie konwertera rs485 na odbiornik
```

w tryb nadajnika – stan przekaźnika PK1, a po przecinku stan przekaźnika PK2. Po zakończeniu wysyłania stanu przekaźników układ *Slave 2* zostaje z powrotem przełączony w tryb odbiornika. Gdy znakiem odebranym po adresie jest „w" oraz po nim znak „o", to otrzymana następnie od *Mastera* wartość wypełnienia będzie wpisywana po zamianie na wartość cyfrową do rejestru wypełnienia przebiegu PWM generowanego na wyjściu OC1A mikrokontrolera. W przypadku, gdy po znaku „w" jest znak „o", otrzymane dalej znaki będą określały stan przekaźników PK1 i PK2. Wartość „0" będzie oznaczała wyłączenie przekaźnika, a „1" jego włączenie. Wartość znaku po przecinku zawsze będzie przeznaczona dla przekaźnika PK2. Choć instrukcje programu odpowiedzialne za interpretacje komend dla układu *Slave 2* są bardziej rozbudowane niż dla *Slave 1*, to warto zauważyć, że są one rozszerzeniem instrukcji interpretujących komendę (jedną) dla układu *Slave 1*. Oczywiście podobnie jak w przypadku układu *Slave 1*, tak i w przypadku układu *Slave 2* wszystkie dane nadawane przez *Mastera*, jak i wysyłane przez *Slave 2* muszą być potwierdzone znakiem CR (*enterem*).

Został jeszcze do przedstawienia najważniejszy układ prezentowanego systemu, którym jest *Master*, czyli nadrzędny układ komunikujący się z układami *Slave* dołączonymi do magistrali RS485. *Master* umożliwia sterowanie prędkością obrotową wentylatora, odczytuje z układu *Slave 1* temperaturę i wyświetla ją na LCD, realizuje prosty termostat z wykorzystaniem układów *Slave 1* i *2* oraz umożliwia sterowanie drugim przekaźnikiem układu *Slave 2* za pomocą przycisku. Każde naciśnięcie tego przycisku będzie powodować naprzemienne włączenie i wyłączenie jednego przekaźnika układu *Slave 2*, gdyż drugi przekaźnik wykorzystuje *Mastera* do realizacji termostatu. Na **rys. 5** przedstawiono schemat elektryczny sterownika systemu (*Mastera*).

Konwerter MAX485 spełnia takie samo zadanie jak w przypadku układów *Slave 1* i *Slave 2*. Dołączono go także do tych samych linii mikrokontrolera. Podczas spoczynku, gdy wszystkie układy dołączone do magistrali przełączone są w tryb odbiorników (także *Master*), na magistrali RS485 panują stany nieustalone, które mogą być błędnie interpretowane przez odbiorniki. Aby temu zapobiec, linie magistrali są wstępnie ustawiane przez rezystory R1...R3 tak, by napięcia na niej interpretowane były przez odbiorniki jako stan wysoki. W sterowniku przycisk S1 umożliwia załączanie oraz wyłączanie zdalnego przekaźnika PK1, w który wyposażono *Slave 2*. Za pomocą potencjometru P1, z którego napięcie jest konwertowane na postać cyfrową przez przetwornik A/C zawarty w mikrokontrolerze, możliwa jest regulacja prędkości wentylatora dołączonego do układu *Slave 2*. Wszystkie parametry odczytywane z układów *Slave 1* i *2* są ukazywane na wyświetlaczu LCD. Potencjometr P2 umożliwia regulację kontrastu wyświetlacza. Na wyświetlaczu prezentowane są: wartość wypełnienia PWM (od 0 do 255), stan przekaźników PK1 i PK2, zmierzona temperatura oraz zadana temperatura, która ma być utrzymywana przez termostat. Funkcja termostatu w sterowniku jest bardzo prosta i w celach przykładowych temperatura zadana została ustawiona na stałym poziomie 25°C. Przekroczenie tej temperatury będzie powodować wyłączenie zdalnego przekaźnika PK2, a gdy temperatura zmierzona będzie niższa niż 25 stopni, przekaźnik PK2 zostanie załączony. Histereza termostatu została ustalona na poziomie 1°C. Oczywiście działanie sterownika (*Mastera*) ma pokazać sposób obsługi wielu układów dołączonych do magistrali RS485, a nie jest realizacją konkretnego urządzenia. Funkcja termostatu

List. 2. cd.

```
        Elseif Zn = "o" Then    'w przeciwnym razie jesli otrzymano znak "o" to
          Do                    'poczatek wewnetrznej petli do-loop
            Zn = Inkey()        'odczyt znaku z portu rs232
          Loop Until Zn = Chr(13)       'jesli otrzymany znak to cr - enter (kod ascii 13) to
          St_pk1 = Pk1          'zapisz do st_pk1 stan przekaznika pk1
          St_pk2 = Pk2          'zapisz do st_pk2 stan przekaznika pk2
          Set Re_de             'przelaczenie konwertera rs485 na nadajnik
          Waitus 200            'czekaj 200 us
          Print St_pk1 ; "," ; St_pk2   'wyslij stan przekaznikow pk1 oraz pk2
          Waitms 1              'czekaj 1 ms
          Reset Re_de           'przelaczenie konwertera rs485 na odbiornik
        End If
      End If
      If Zn = "w" Then          'jesli otrzymany znak to "w"
        Do                      'poczatek wewnetrznej petli do-loop
          Zn = Inkey()          'odczyt znaku z portu rs232
        Loop Until Zn > Chr(0)  'opusc petle gdy otrzymany znak wiekszy od kodu 0 ascii
        If Zn = "p" Then        'jesli otrzymany znak to "p" to
          Il_zn = 0             'zeruj zmienna il_zn
          Zm_lan = ""           'czysc zmienna zm_lan
          Do                    'poczatek wewnetrznej petli do-loop
            Zn = Inkey()        'odczyt znaku z portu rs232
            If Zn >= "0" And Zn <= "9" Then       'jesli otrzynany znak jest cyfra (0..9) to
              Incr Il_zn        'zwieksz o jeden il_zn
              Zm_lan = Zm_lan + Zn      'dodaj otrzymany znak cyfry do zmiennej zm_lan
            End If
          Loop Until Il_zn > 3 Or Zn = Chr(13)     'petla do-loop wykonywana az il_zn wieksze
                                'niz 3 lub otrzymano znak CR (ascii 13)
          If Il_zn > 0 And Il_zn < 4 Then         'jesli il_zn wynosi 1, 2 lub 3 to
            Wart_pwm = Val(zm_lan)      'zapisz do wart_pwm wartosc otrzymanego wypelnienia
                                'przekonwertowanego na postac cyfrowa
            If Wart_pwm < 256 Then      'jesli otrzymana wartosc pwm mniejsza niz 256 to
              Pwm1a = Wart_pwm  'zapisz otrzymana wartosc pwm do timera1
            End If
          End If
        Elseif Zn = "o" Then    'w przeciwnym przypadku jesli otrzymano znak "o" to
          Il_zn = 0             'zeruj zmienna il_zn
          Do                    'poczatek wewnetrznej petli do-loop
            Zn = Inkey()        'odczyt znaku z portu rs232
            If Zn = "0" Or Zn = "1" Then 'jesli otrzymano znak o wartosci 0 lub 1 to
              Incr Il_zn        'zwieksz o jeden il_zn
              St_pk1 = Asc(zn)  'zapisz zamieniona na postac cyfrowa znak do zmiennej stanu
                                'przekaznika st_pk1
            Else                'w przeciwnym razie
              If Zn > Chr(0) Then       'jesli otrzymany znak ma kod ascii wiekszy od 0 to
                Exit Do         'opusc petle do-loop
              End If
            End If
          Loop                  'koniec wewnetrznej petli do-loop
          If Il_zn = 1 And Zn = "," Then   'jesli il_zn wynosi 1 oraz otrzymano znak "," to
            Il_zn = 0           'zeruj zmienna il_zn
            Do                  'poczatek wewnetrznej petli do-loop
              Zn = Inkey()      'odczyt znaku z portu rs232
              If Zn = "0" Or Zn = "1" Then         'jesli otrzymano znak o wartosci 0 lub 1 to
                Incr Il_zn      'zwieksz o jeden il_zn
                St_pk2 = Asc(zn)        'zapisz zamieniona na postac cyfrowa znak do zmiennej
                                'stanu przekaznika st_pk2
              Else              'w przeciwnym razie
                If Zn > Chr(0) Then     'jesli otrzymany znak ma kod ascii wiekszy od 0 to
                  Exit Do       'opusc petle do-loop
                End If
              End If
            Loop                'koniec wewnetrznej petli do-loop
            If Il_zn = 1 And Zn = Chr(13) Then     'jesli il_zn rowne 1 oraz otrzymano znak CR
                                '(ascii 13) to
              Pk1 = St_pk1.0    'nadanie stanu najmlodszego bitu st_pk1 linii portu sterujacej
                                'przekaznikiem pk1
              Pk2 = St_pk2.0    'nadanie stanu najmlodszego bitu st_pk2 linii portu sterujacej
                                'przekaznikiem pk2
            End If
          End If
        End If
      End If
    End If
  End If
Loop                            'koniec glownej petli do-loop
End                             'koniec programu
```

jest jednym z przykładów bardziej praktycznego wykorzystania układów *Slave 1* i *Slave 2* dołączonych do magistrali RS485. Na **list. 3** przedstawiono program sterujący sterownikiem (*Masterem*).

Jest to program, który w pętli wysyła dane do układów *Slave 1* i *2*, następnie dane odczytuje i wyświetla na wyświetlaczu LCD. *Master* także odbiera znaki z magistrali w przerwaniu i umieszcza je w 15-bajtowym buforze. W programie po wysłaniu do układów *Slave* komendy odczytu, *Master* na otrzymanie danych od układów *Slave* czeka ok. 100 ms, nie wstrzymując działania programu. Do odbioru danych od układów *Slave* nie użyto instrukcji *input*, gdyż wstrzymuje ona działanie programu aż do odebrania znaku „CR". W przypadku uszkodzenia lub braku któregoś z układów *Slave*, od którego oczekiwane są dane, program uległby zawieszeniu. Odbieraniem danych od układów *Slave* zajmuje się funkcja *inkey*, która nie wstrzymuje programu, ale odczyt wielu danych za jej pomocą składa się z kilku instrukcji. Zatem przy braku któregoś z układów *Slave* nie zostanie wstrzymana praca układu sterownika. Stan przekaźników PK1 i PK2 układu *Slave 2* jest prezentowany w postaci graficznej. W programie zostały zdefiniowane dla LCD znaki załączonego i wyłączonego przekaźnika oraz znaku „stopnia" dla temperatury. W przypadku braku odpowiedzi od *Slave* przez co najmniej 100 ms, przy odczytywanym z niego parametrze umieszczany jest na LCD znak „?", który może świadczyć o uszkodzeniu danego układu *Slave* lub jego braku w systemie. W programie do odczytu znaków od układów *Slave* służy procedura *odczyt_zn*. Procedura ta odbiera jedynie znaki cyfr od 0 do 9, aż do otrzymania znaku potwierdzenia „CR – LF". W programie głównym, na samym jego początku, jest mierzone napięcie na potencjometrze P1 regulującym obroty zdalnego wentylatora (dołączonego do *Slave 2*) przez przetwornik A/C. Wartość z przetwornika jest konwertowana do wartości 8-bitowej (przetwornik jest 10-bitowy), przez co skrajnym położeniom potencjometru będą przypadać wartości z przetwornika 0 i 255, które wysłane do układu *Slave 2* mogą być wprost wpisane do rejestru PWM (PWM ma rozdzielczość 8 bitów). Należy zwrócić uwagę, że napięcie odniesienia przetwornika zostało ustalone na 5 V. Po przekonwertowaniu wartości odczytanej z P1 jest ona wysyłana do rejestru PWM układu *Slave 2*. Następnie jest badany stan przycisku „S1". Jeśli został naciśnięty, wywołany zostaje podprogram *p_s1*, w którym następuje odwrócenie bitu 0 zmiennej *ST_pk1* określającego stan przekaźnika PK1 układu *Slave 2*. Następnie jest wysyłany aktualny stan przekaźników do *Slave 2*. Należy zwrócić uwagę, że przy wysyłaniu danych do układów *Slave*, *Master* jest na tę chwilę przełączany z trybu odbiornika w tryb nadajnika. Kolejne instrukcje programu głównego odczytują temperaturę od układu *Slave 1* realizują funkcję termostatu z uwzględnieniem histerezy. Wartość temperatury zadanej zawiera stałą *temp_ust*, a histerezy stałą *hist*. W zależności od temperatury odczytanej (ze *Slave 1)* i zadanej sterowany jest przekaźnik PK2, którego stan jest wysyłany do układu *Slave 2*. Pozostałe instrukcje pętli głównej programu prezentują w pierwszej linii wyświetlacza temperaturę odczytaną z układu *Slave 1* i temperaturę zadaną.

W drugiej linii LCD prezentowane są odczytane z układu *Slave 2* wypełnienia sygnału PWM (0...255) oraz graficzny stan przekaźników PK1 i PK2. Pętla główna programu jest powtarzana

List. 3. Program sterownika *Master*

```
'Przykład urzadzenia (sterownika) master sieci RS485.
'Urzadzenie realizuje funkcje termostatu - odczytuje temperature
'ze slave 1 i steruje ukladem wykonawczym (przekaznikiem PK2) slave 2
'Za posrednictwem potencjometru, master ma takze mozliwosc sterowania
'obrotami wentylatora doalczonego do slave 2
'Master umozliwia takze sterowanie drugim przekaznikiem (PK1) urzadzenia slave 2 za
'posrednictwem przycisku

$regfile = "m8def.dat"              'informuje kompilator o pliku dyrektyw wykorzystywanego
                                    'mikrontrolera
$crystal = 8000000                  'informuje kompilator o czestotliwosci rezonatora kwarcowego
$baud = 9600                        'informuje kompilator o predkosci transmisji RS232

Config Pind.7 = Output              'linia pd.7 jako wyjscie
Config Pind.2 = Output              'linia pd.2 jako wyjscie
Config Lcd = 16 * 2                 'konfiguracja organizacji znakow wyswietlacza LCD
Config Lcdpin=Pin, Db4=Portb.3, Db5=Portb.2, Db6=Portb.1, Db7=Portb.0, E=Portb.4, Rs=Portb.5
                                    'konfiguracja pinow mikrokontrolara do
Config Adc = Single , Prescaler = Auto , Reference = Avcc
                                    'konfiguracja wewnetrznego przetwornika
Config Serialin = Buffered , Size = 15
                                    'konfiguracja by interfejs rs232 uzywal przy odbiorze transmisji
                                    'buforowej (bufor o wielkosci 15 znakow)

Const Temp_ust = 25                 'ustawiona temperatura termostatu
Const Hist = 1                      'histereza teprmostatu

Declare Sub Odczyt_zn               'procedura odczytu otrzymanych danych od ukladow slave

Dim Zn As String * 1                'zmienna przecowujaca odebrany znak z rs232
Dim Zm_lan As String * 5            'zmienna w ktorej skladany jest w calosc otrzymany adres
Dim Wyp_pwm As Byte                 'przechowuje wartosc wypelnienia PWM
Dim Wart_adc As Word                'przechowuje wartosc odczytana z przetwornika A/C
Dim St_pk1 As Byte                  'przechowuje stan przekaznika pk1
Dim St_pk2 As Byte                  'przechowuje stan przekaznika pk2
Dim Czekaj As Word                  'zmienna liczaca czas na odpowiedz ukladow slave
Dim Temp_odcz As Integer            'zmienna przechowujaca odczytana temperature od slave 1
Dim Temp_pom As Integer             'zmienna przechowuje obliczana temperature przy realizacji
                                    'termostatu

Re_de Alias Portd.2                 'przypisanie aliasu linii pd.2, re_de steruje kierunkiem
                                    'transmisji konwertera
S1 Alias Pind.7                     'przypisanie aliasu S1 (dolaczony przycisk S1) linii pd.7

Deflcdchar 0 , 8 , 4 , 19 , 32 , 14 , 14 , 14 , 14    'definicja znaku rozlaczonego przekaznika
Deflcdchar 1 , 32 , 32 , 31 , 32 , 14 , 14 , 14 , 14  'definicja znaku zalaczonego przekaznika
Deflcdchar 2 , 2 , 5 , 2 , 32 , 32 , 32 , 32 , 32     'definicja znaku stopnia

Set Portd.7                         'dolaczenie do linii wejsciowej d.7 rezystora podciagajacego
Start Adc                           'uruchamia wbudowany przetwornik A/C
Enable Interrupts                   'odblokowanie globalnych przerwan
St_pk1 = 0                          'zeruje zmienna okreslajaca stan przekaznika pk1
St_pk2 = 0                          'zeruje zmienna okreslajaca stan przekaznika pk2
Cursor Off                          'kursor lcd wylaczony
Cls                                 'czysc LCD

Do                                  'poczatek glownej petli programu
  Wart_adc = Getadc(0)              'pomiar napiecia na wejtciu PC0 mikrokontrolera
  Shift Wart_adc , Right , 2        'przesuniecie bitow wartosci odczytanej z A/C o dwa miejsca
                                    'w prawo (pozbycie sie 2 najmlodszych bitow)
  Wyp_pwm = Low(wart_adc)           'zapisz mniej znaczacej czesci slowa wart_adc do zmiennej
                                    'bajtowej wyp_pwm
  Set Re_de                         'przelaczenie konwertera rs485 na nadajnik
  Waitms 1                          'czekaj 1ms
  Print "{008}8wp" ; Wyp_pwm       'zapisuje do slave 2 wartosc wypelnienia PWM, ktory steruje
                                    'wentylatorem
  Waitms 1                          'czekaj 1ms
  Reset Re_de                       'przelaczenie konwertera rs485 na odbiornik
  Debounce S1 , 0 , P_s1 , Sub      'jesli nacisnieto S1 to wywolanie podprogramu P_s1
                                    'obslugujacego S1
  Set Re_de                         'przelaczenie konwertera rs485 na nadajnik
  Waitms 1                          'czekaj 1ms
  Print "{008}5r"                   'wysyla do slave 1 rozkaz rzadania otrzymania zmierzonej
                                    'temperatury
  Waitms 1                          'czekaj 1ms
  Reset Re_de                       'przelaczenie konwertera rs485 na odbiornik
  Call Odczyt_zn                    'wywolanie procedury odczytu znakow od slave
  Temp_odcz = Val(zm_lan)           'zamiana odczytanej temperatury (znakowo) na postać numeryczna
  Temp_pom = Temp_ust - Hist        'obliczenie temp_pom poprzez odjecie od temperatury zadanej
                                    'wartosci histerezy
  If Temp_odcz <= Temp_pom Then     'jesli temp_odcz mniejsza lub rowna temp_pom to
    St_pk2 = 1                      'zalaczenie przekaznika pk2
  Else                              'w przeciwnym razie
    Temp_pom = Temp_ust + Hist      'obliczenie temp_pom poprzez dodanie do temperatury
                                    'zadanej wartosci histerezy
    If Temp_odcz >= Temp_pom Then   'jesli temp_odcz wieksza lub rowna temp_pom to
```

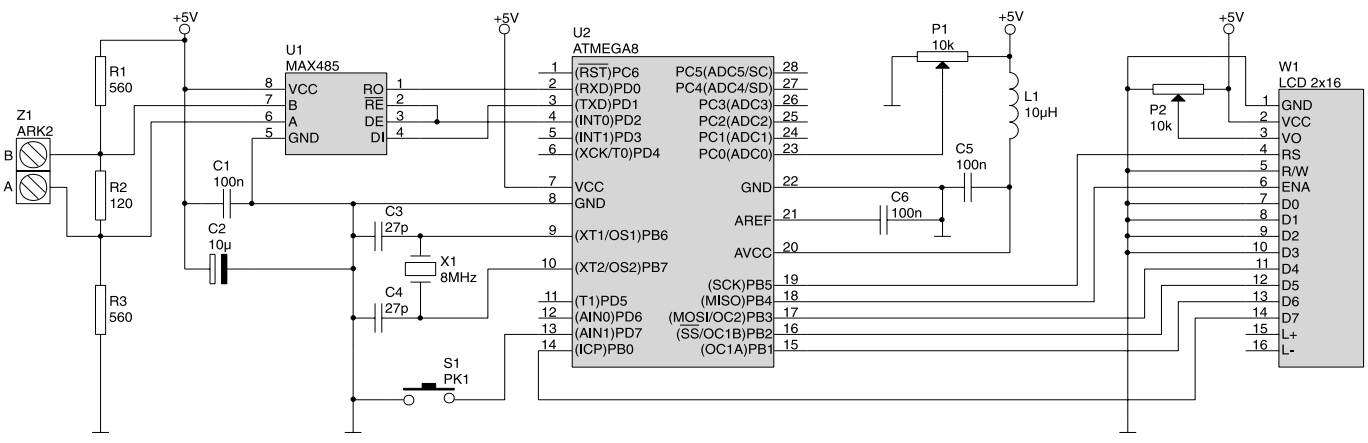
List. 3. cd.

```
      St_pk2 = 0                  'wyłaczenie przekaznika pk2
    End If
  End If
  Set Re_de                       'przelaczenie konwertera rs485 na nadajnik
  Waitms 1                        'czekaj 1ms
  Print "{008}8wo" ; St_pk1 ; "," ; St_pk2     'wysyla do ukladu slave 2 aktualny stan
                                  'przekaznikow pk1 i pk2
  Waitms 1                        'czekaj 1ms
  Reset Re_de                     'przelaczenie konwertera rs485 na odbiornik
  Home                            'kursor na poczatej pierwszej linii LCD
  If Czekaj > 100 Then            'jesli czekaj wieksze od 100 to
    Lcd "Tz=?     "               'wyswietl na lcd ? - brak odczytanej temp (brak komunikacji
                                  'ze slave 1)
  Else                            'w przeciwnym razie
    Lcd "Tz=" ; Temp_odcz ; Chr(2) ; "C  "      'wyswietlenie otrzymanej od slave 1 temperatury
                                  'z dodatkowym znakiem stopnia
  End If
  Locate 1 , 10                   'kursor w linii 1 i na pozycji 10 LCD
  Lcd "Tu=" ; Temp_ust ; Chr(2) ; "C  "   'wyswietlenie temperatury zadanej z dodatkowym
                                  'znakiem stopnia
  Set Re_de                       'przelaczenie konwertera rs485 na nadajnik
  Waitms 1                        'czekaj 1ms
  Print "{008}8rp" ; Wyp_pwm      'odczytuje od slave 2 wartosc wypelnienia PWM, ktory
                                  'steruje wentylatorem
  Waitms 1                        'czekaj 1ms
  Reset Re_de                     'przelaczenie konwertera rs485 na odbiornik
  Call Odczyt_zn                  'wywolanie procedury odczytu znakow od slave
  Lowerline                       'kursor na poczatek drugiej linii LCD
  If Czekaj > 100 Then            'jesli czekaj wieksze od 100 to
    Lcd "PWM=?  "                 'wyswietl na lcd ? - brak odczytanej wartosci PWM (brak
                                  'komunikacji ze slave 2)
  Else                            'w przeciwnym razie
    Lcd "PWM=" ; Zm_lan ; "  "    'wyswietlenie odczytanej wartosc PWM ze slave 2
  End If
  Set Re_de                       'przelaczenie konwertera rs485 na nadajnik
  Waitms 1                        'czekaj 1ms
  Print "{008}8ro" ; Wyp_pwm      'odczytuje od slave 2 stan przekaznikow PK1 oraz PK2
  Waitms 1                        'czekaj 1ms
  Reset Re_de                     'przelaczenie konwertera rs485 na odbiornik
  Call Odczyt_zn                  'wywolanie procedury odczytu znakow od slave
  Locate 2 , 12                   'kursor w linii 2 i na pozycji 12 LCD
  Zn = Mid(zm_lan , 1 , 1)        'odczyt ze zmiennej lancuchowej zm_lan, znakowej wartosci
                                  'stanu przekaznika PK1
  St_pk1 = Val(zn)                'zamiana znakowej postaci stanu PK1 na postac numeryczna
                                  'i zapis jej do zmiennej st_pk1
  Zn = Mid(zm_lan , 2 , 1)        'odczyt ze zmiennej lancuchowej zm_lan, znakowej wartosci
                                  'stanu przekaznika PK2
  St_pk2 = Val(zn)                'zamiana znakowej postaci stanu PK2 na postac numeryczna
                                  'i zapis jej do zmiennej st_pk2
  Locate 2 , 12                   'kursor w linii 2 i na pozycji 12 LCD
  If Czekaj > 100 Then            'jesli czekaj wieksze od 100 to
    Lcd "?  ?"                    'wyswietl na lcd ? - brak odczytanego stanu przekaznikow PK1
                                  'i PK2 (brak komunikacji ze slave 2)
  Else                            'w przeciwnym razie
    Lcd Chr(st_pk1) ; "  " ; Chr(st_pk2)   'wyswietlenie graficznej reprezentacji odczytanej
                                  'wartosci stanow przekaznikow PK1 i PK2 ukladu slave 2
  End If
  Waitms 200                      'czekaj 200 ms
Loop                              'koniec glownej petli do-loop
End                               'koniec programu

Sub Odczyt_zn                     'procedura odczytu znakow nadawanych przez uklady slave
  Czekaj = 0                      'zerowanie zmiennej czekaj
  Zm_lan = ""                     'czyszczenie zmiennej znakowej zm_lan
  Do                              'poczatek petli do-loop
    Incr Czekaj                   'zwieksz o jeden zmienna czekaj
    Zn = Inkey()                  'odczyt znaku z bufora rs232
    If Zn >= "0" And Zn <= "9" Then       'jesli odczytany znak na kod ascii liczby
                                  'z zakresu 0..9 to
      Zm_lan = Zm_lan + Zn        'dodaj odczytany znak do zmiennej zm_lan
    End If
    Waitms 1                      'czekaj 1ms
  Loop Until Zn = Chr(13) Or Czekaj > 100    'petla wykonywana az otrzymano znak CR (ascii 13)
                                  'lub zmienna czejaj> niz 100
End Sub                           'koniec procedury - odczytane dane zamiera zmienna zm_lan

P_s1:                             'podprogram obslugi przycisku S1
  St_pk1 = St_pk1 Xor &B00000001  'zamienia bit 0 zmiennej st_pk1 na przeciwny (zmiana stanu pk1
                                  'na przeciwny)
  Set Re_de                       'przelaczenie konwertera rs485 na nadajnik
  Waitms 1                        'czekaj 1 ms
  Print "{008}8wo" ; St_pk1 ; "," ; St_pk2    'wysyla do ukladu slave 2 aktualny stan
                                  'przekaznikow pk1 i pk2
  Waitms 1                        'czekaj 1 ms
  Reset Re_de                     'przelaczenie konwertera rs485 na odbiornik
Return                            'powrot z podprogramu
```

co ok. 200 ms.

Liczne komentarze w przedstawionych programach pomogą w ich dokładnym zrozumieniu i rozbudowie o nowe funkcje. Przedstawiony system jest tylko przykładem, który miał ukazać aspekty komunikacji z wykorzystaniem magistrali RS485 na dość duże odległości. W przypadku magistrali RS485 można dołączyć do niej wiele układów *Slave* tego samego typu, ale dołączone układy powinny mieć różne adresy. Na przykład, dołączone dwa czujniki temperatury mogą mieć adresy 1 oraz 2 lub dowolnie inne z zakresu od 0 do 99. W przypadku potrzeby szybkich transferów danych na magistrali RS485 w obu kierunkach, można do tego celu stworzyć magistralę RS485 z dwoma parami przewodów, w których jedna służy do odbioru, a druga do nadawania danych. W tym przypadku należy wykorzystać po dwa konwertery w każdym układzie *Master* lub *Slave*. Dostępne są także w jednej obudowie podwójne konwertery RS485. Jeżeli transmisje danych mają przebiegać bez błędów, wysyłane dane można opatrzyć np. w sumę kontrolną według algorytmu CRC. Obliczenie sumy kontrolnej po odbiorze danych umożliwi sprawdzenie poprawności danych z otrzymaną od nadajnika sumą kontrolną CRC. W przedstawionym systemie protokół transmisji w formie znaków „BS – adres układu – zapis lub odczyt-dane" może być całkowicie inny, dopasowany do danego systemu. Może być wyposażony we wspomnianą sumę kontrolną CRC, którą nadajnik wysyła z danymi, a odbiornik oblicza i porównuje z otrzymaną od nadajnika wraz z danymi. Zaprezentowany system może być punktem wyjściowym do budowy bardziej rozbudowanych systemów w oparciu o magistralę RS485.

Marcin Wiązania, EP
marcin.wiazania@ep.com.pl



Rys. 5. Schemat elektryczny sterownika (*Mastera*) systemu RS485