hotcold

SPECYFIKACJA

Przetwornik wilgotności

HCRH-MODBUS-Ka

1. Wprowadzenie

Przedmiotem niniejszego opracowania jest charakterystyka funkcjonalności przetwornika wilgotności opartego na cyfrowym czujniku firmy HONEYWELL, z interfejsem RS-485 z wbudowanym protokołem MODBUS RTU. Przed przystąpieniem do uruchomienia modułu należy zapoznać się z tekstem zawartym w opracowaniu.

1.1. Funkcje urządzenia

- pomiar wilgotności względnej
- pomiar temperatury
- wyliczenie punktu rosy
- diodowa sygnalizacja pracy urządzenia
- szeregowy interfejs RS-485 (odczyt wartości pomiarowych, konfiguracja parametrów pracy)
    - protokół MODBUS RTU
    - komunikacja w trybie HALFDUPLEX

1.2. Charakterystyka urządzenia

Podstawową funkcją przetwornika HCRH-MODBUS jest wyznaczanie chwilowych wartości wilgotności względnej skompensowanej temperaturowo. Zmierzone za pośrednictwem zintegrowanego czujnika cyfrowego wartości, przeliczone w mikrokontrolerze, dostępne są w jego pamięci (w rejestrach typu HOLDING REGISTERS) zgodnie ze standardem MODBUS. Odczyt rejestrów odbywa się za pomocą funkcji protokołu MODBUS przesyłanych szeregowym interfejsem RS-485.

Ze względu na istotny wpływ pomiaru temperatury na wartość wilgotności, zoptymalizowano działanie urządzenia pod kątem poboru mocy i tym samym wyeliminowano efekt podgrzewania czujnika HIH. Obniżenie poboru mocy uzyskano m.in. dzięki przełączaniu procesora w tryb uśpienia pomiędzy kolejnymi pomiarami i odczytami.

2. Dane techniczne

2.1. Parametry przetwornika

Zasilanie	
- napięciem stałym	24 VDC (20..30 VDC)
- napięciem przemiennym	24 VAC (16..25 VAC)
Pobór prądu	
- minimalny [1)]	10 mA
- typowy [2)]	15 mA
- maksymalny [3)]	25 mA
Sygnalizacja LED	0,2 Hz
Złącze instalacyjne	śrubowe w rastrze 5 mm (do 2,5mm^2)
Wymiary	112 x 62 x 32 (L x H x W)
Waga	-
Montaż [4)]	Naścienny
Stopień ochrony	IP65
Środowisko pracy	bezpyłowe, powietrze, gazy neutralne
Temperatura pracy	-20°C ÷ 85°C
Warunki przechowywania	
- temperatura (szczytowa)	10 ÷ 50°C (0 ÷ 125°C)
- wilgotność względna	20 ÷ 60 %RH

1) Średni pobór prądu urządzenia w warunkach: wyjście analogowe nieobciążone; brak transmisji; zasilanie 24V DC;
2) Średni pobór prądu urządzenia w warunkach: wyjście analogowe obciążone rezystancją 10kΩ; transmisja FULL DUPLEX 10 zapytań na sekundę; prędkość transmisji 9600 b/s; jednoczesny odczyt 25 rejestrów; rezystory terminujące magistralę 2 x 120Ω; zasilanie 24V DC;
3) Maksymalny chwilowy pobór prądu w warunkach jak w punkcie 2); zasilanie 24V DC;
4) Instalacji urządzenia powinien dokonywać wykwalifikowany personel;

2.2. Parametry pomiaru wilgotności

Typ czujnika	HIH6131
Zakres pomiarowy	0 ÷ 100 %RH
Rozdzielczość	12 bitów (0,05 %RH)
Dokładność dla T=25°C	
- w zakresie 20 ÷ 80 %RH	±3 %RH
- w pozostałym zakresie	±(3 ÷ 5) %RH
Histereza	±1 %RH
Częstotliwość próbkowania	0,5 Hz
Czas odpowiedzi [5)]	8 s

5) Warunkiem uzyskania podanych czasów odpowiedzi jest przepływ powietrza > 1m/s; podany czas odpowiedzi jest równy jednej stałej czasowej odpowiadającej 63% wartości ustalonej;

2.3. Parametry pomiaru temperatury

Typ czujnika	HIH6131
Zakres pomiarowy	-40°C ÷ 85°C
Rozdzielczość	14 bitów (0,01 °C)
Dokładność	
- w zakresie 20 ÷ 30 °C	±0,5 °C
- w pozostałym zakresie	±(0,5 ÷ 2,3) °C
Częstotliwość próbkowania	100 Hz
Czas odpowiedzi [1)]	6 ÷ 30 s

1) Warunkiem uzyskania podanych czasów odpowiedzi jest przepływ powietrza > 1m/s; podany czas odpowiedzi jest równy jednej stałej czasowej odpowiadającej 63% wartości ustalonej;

2.4. Parametry interfejsu szeregowego

Warstwa fizyczna	RS-485
Protokół komunikacji	MODBUS RTU
Konfiguracje połączeń [2)]	HALFDUPLEX
Prędkości transmisji	9600 / 19200 / 57600 / 115200 b/s

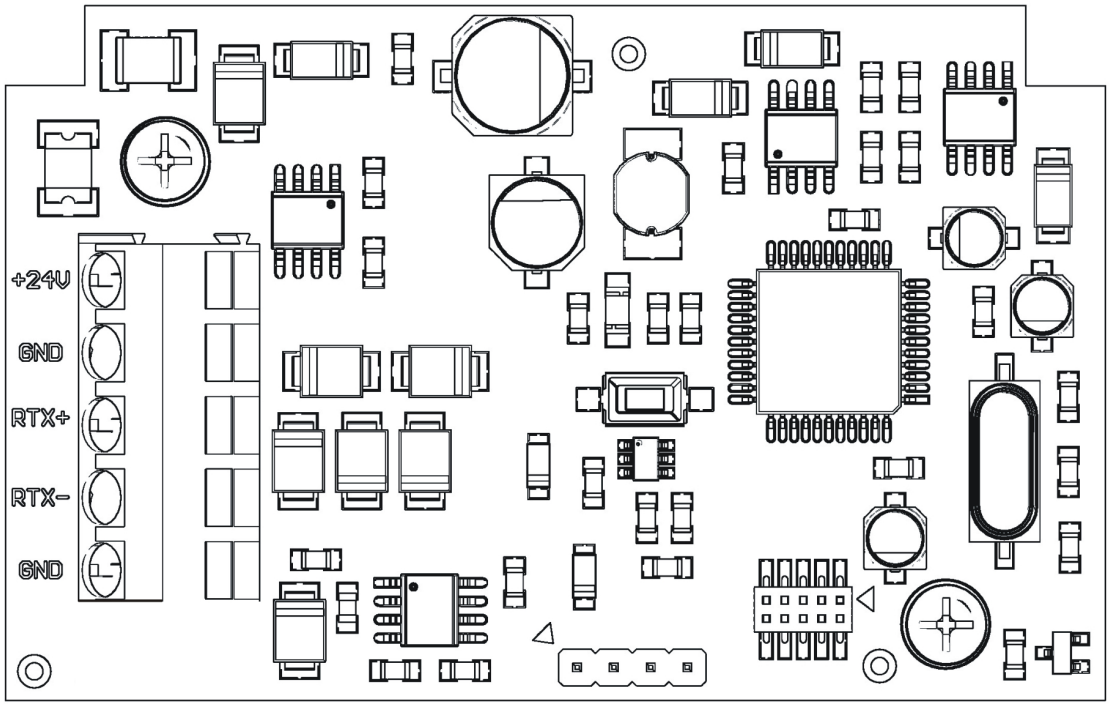
2) HALF DUPLEX – komunikacja dwukierunkowa jedną parą przewodów;

3. Instalacja

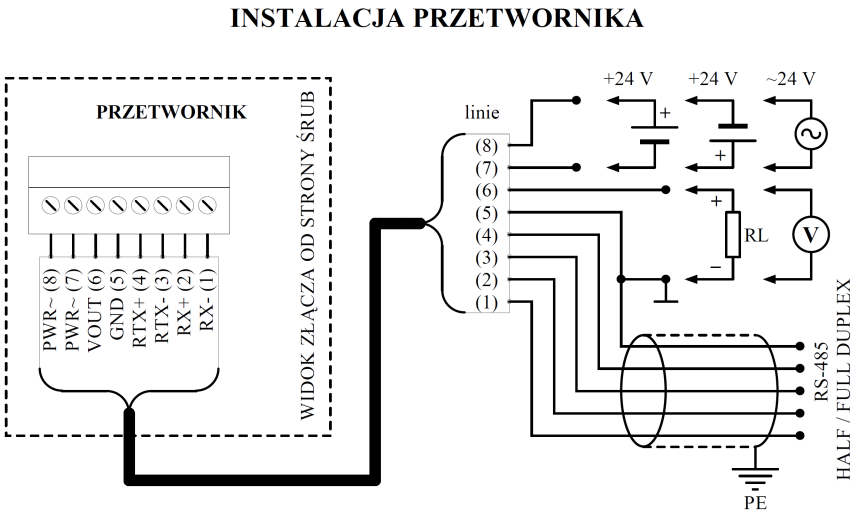
3.1. Bezpieczeństwo

- Instalacji urządzenia powinien dokonywać wykwalifikowany personel!
- Wszystkie podłączenia należy wykonać zgodnie ze schematami elektrycznymi przedstawionymi w niniejszej specyfikacji!
- Przed przystąpieniem do uruchomienia należy sprawdzić wszystkie podłączenia elektryczne!

3.2. Połączenia



Rysunek 1. Obwód drukowany wersji MODBUS-Ka

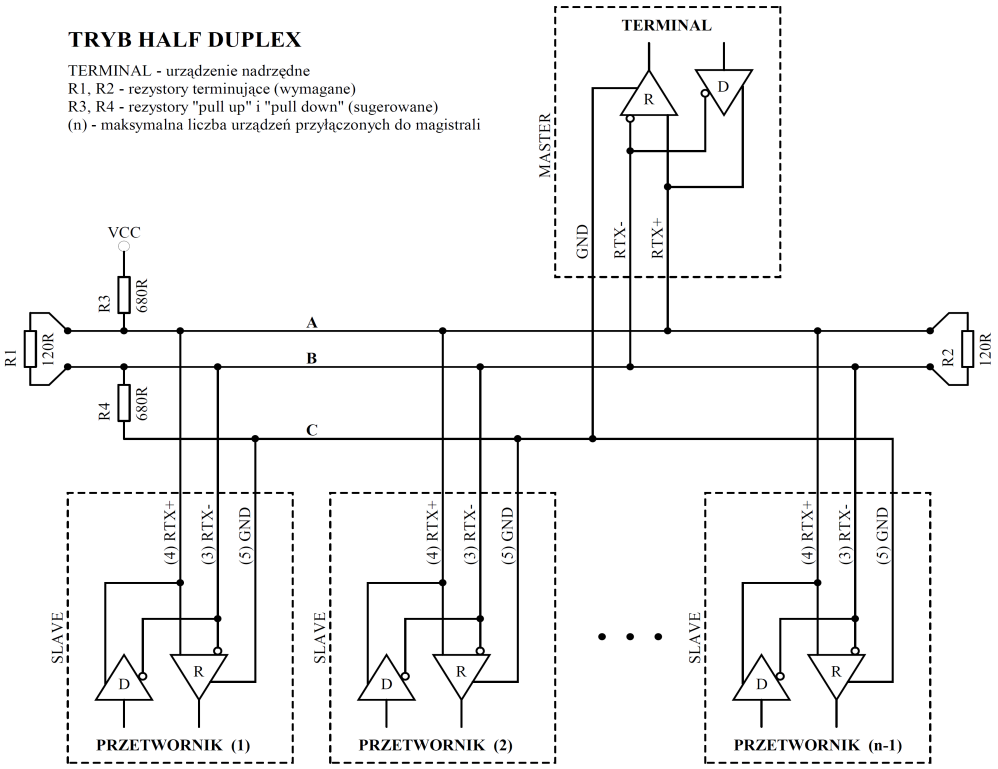


Rysunek 1. Schemat podłączenia przetwornika HCRH-MODBUS.

3.3. Wytyczne

Ważne Instalacja toru transmisji zalecana jest pełną obsadą okablowania, łącznie z przewodem GND.

- W przypadku pracy w otoczeniu dużych zakłóceń, należy zastosować przewody ekranowane.
- Ekran przewodu należy podłączyć do najbliższego punktu PE od strony zasilacza.



Rysunek 2. Sposób podłączenia przetwornika do magistrali RS-485 pracującej w trybie HALF DUPLEX.

4. Protokół MODBUS

4.1. Mapa rejestrów

Tabela rejestrów:

Nr rejestru	Wartości	Opis
1	1÷1000	Wilgotność względna (1 = 0,1%; 1000 = 100%)
2	-4000÷12380	Temperatura [ºC] (1 = 0,01 ºC) ze znakiem
3	-4000÷12380	Punkt rosy [ºC] (1 = 0,01 ºC) ze znakiem
4	1234	Rejestr hasła
5	1. / 2. / 3.	Rejestr poleceń
6	wg tabeli poleceń	Rejestr parametru
7	0÷65535	Licznik poprawnych ramek
8	0÷65535	Licznik wyjątków
9	0÷65535	Licznik błędnych CRC
10	0÷65535	Licznik błędnych bajtów
11	0÷65535	Licznik błędnych adresów

Tabela poleceń:

Nr polecenia	Funkcja	Parametry
1	Ustaw adres urządzenia	1 – 247 (1-wartość domyślna)
2	Ustaw prędkość transmisji	96 – 9600 b/s (wartość domyślna) 192 – 19200 b/s 576 – 57600 b/s 1152 – 115200 b/s
3	Ustaw bity parzystości	0 – NO PARITY; brak bitu parzystości 1– EVEN PARITY; (wartość domyślna) 2 – ODD PARITY,
4	Ustaw bity stopu	1–1 x STOP; 1 bit stopu (wartość domyślna) 2 – 2 x STOP; 2 bity stopu

Uwagi:

- **Podanie błędnej lub spoza zakresu wartości parametru, skutkuje wpisaniem do rejestru poleceń wartości 0xEEEE.**
- **Każdorazowemu wywołaniu polecenia musi towarzyszyć wpisanie hasła (1234 decymalnie).**
- **Wywołanie polecenia poprzez pojedyncze wpisy do rejestrów, musi zostać zakończone wpisaniem hasła.**
- **Przywracanie ustawień fabrycznych poprzez zworkę (przycisk). W dowolnym momencie działania urządzenia należy zewrzeć zworkę na czas około 2 s (sygnalizowane pełnym świeceniem diody statusowej), po upływie 2s dioda statusowa zacznie szybko migać, wtedy należy puścić przycisk, a domyślne nastawy transmisji RS-485 zostaną przywrócone.**

4.2. Funkcje protokołu

W przetworniku HCRH-MODBUS zaimplementowano następujące funkcje standardu MODBUS:

KOD	ZNACZENIE
03 (0x03)	Odczyt N x 16-bitowych rejestrów
16 (0x10)	Zapis N x 16-bitowych rejestrów

4.2.1. Odczyt zawartości grupy rejestrów wyjściowych (0x03)

Format żądania:

Opis	Rozmiar	Wartości
Adres urządzenia	1 bajt	1 – 247 (0xF7)
Kod funkcji	1 bajt	**0x03**
Adres bloku danych	2 bajty	0x0000 – 0xFFFF
Liczba rejestrów (N)	2 bajty	1 – 125 (0x7D)
Suma kontrolna CRC	2 bajty	wg obliczeń

Format odpowiedzi:

Opis	Rozmiar	Wartości
Adres urządzenia	1 bajt	1 – 247 (0xF7)
Kod funkcji1 bajt	1 bajt	**0x03**
Licznik bajtów1 bajty	1 bajt	2 x N
Wartości rejestrów	N x 2 bajty	wg mapy rejestrów
Suma kontrolna CRC	2 bajty	wg obliczeń

Format błędu:

Opis	Rozmiar	Wartości
Adres urządzenia	1 bajt	1 – 247 (0xF7)
Kod funkcji	1 bajt	**0x83**
Kod błędu	1 bajt	0x01 / 0x02 / 0x03 / 0x04
Suma kontrolna CRC	2 bajty	wg obliczeń

4.2.2. Zapis do grupy rejestrów wyjściowych (0x10)

Format żądania:

Opis	Rozmiar	Wartości
Adres urządzenia	1 bajt	1 – 247 (0xF7)
Kod funkcji1 bajt	1 bajt	**0x10**
Adres bloku danych	2 bajty	0x0000 – 0xFFFF
Liczba rejestrów (N)	2 bajty	1 – 123 (0x7B)
Licznik bajtów	1 bajt	2 x N
Wartości	N x 2 bajty	użytkownika
Suma kontrolna CRC	2 bajty	wg obliczeń

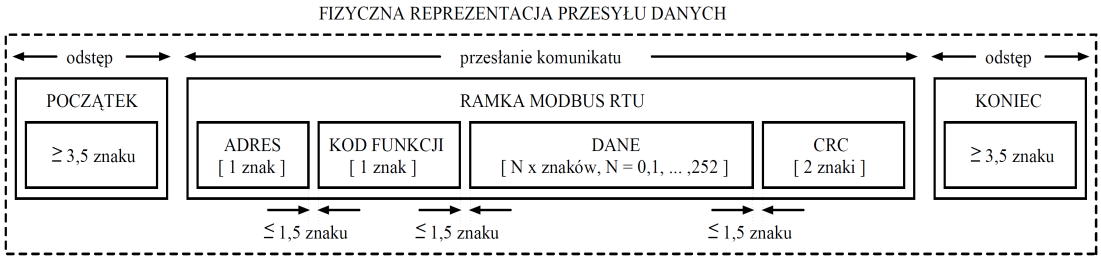
Format odpowiedzi:

Opis	Rozmiar	Wartości
Adres urządzenia	1 bajt	1 – 247 (0xF7)
Kod funkcji1 bajt	1 bajt	**0x10**
Adres bloku danych	2 bajty	0x0000 – 0xFFFF
Liczba rejestrów (N)	2 bajty	1 – 123 (0x7B)
Suma kontrolna CRC	2 bajty	wg obliczeń

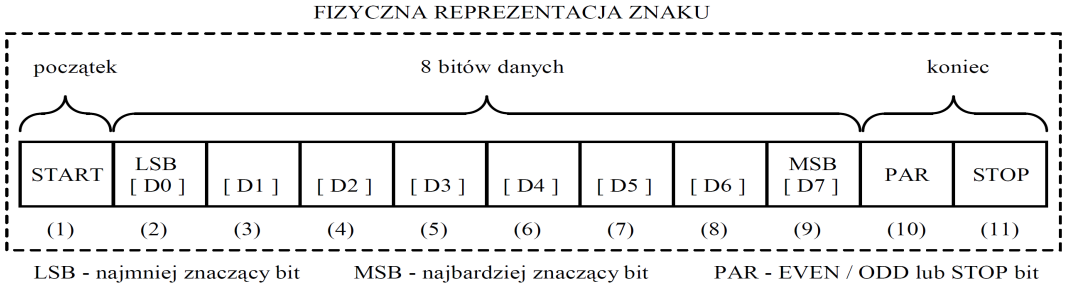
Format błędu:

Opis	Rozmiar	Wartości
Adres urządzenia	1 bajt	1 – 247 (0xF7)
Kod funkcji	1 bajt	**0x90**
Kod błędu	1 bajt	0x01 / 0x02 / 0x03 / 0x04
Suma kontrolna CRC	2 bajty	wg obliczeń

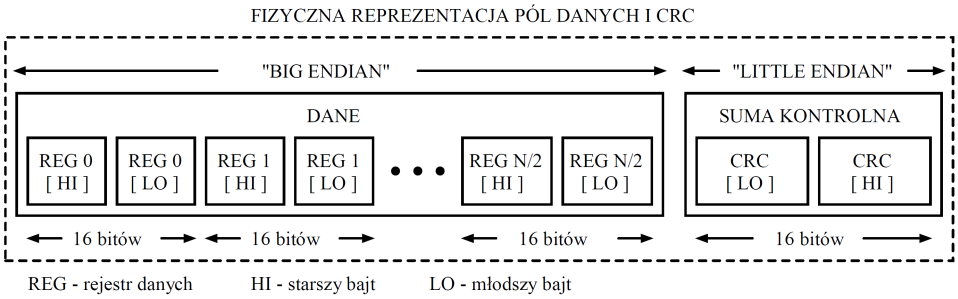
4.3. Format danych



Rysunek 3. Przesył danych w standardzie MODBUS RTU zaimplementowany w przetworniku.



Rysunek 4. Format znaku w standardzie MODBUS RTU zastosowany w przetworniku.



Rysunek 5. Format pól danych i CRC w standardzie MODBUS RTU zastosowany w przetworniku.

4.4. Suma kontrolna CRC

Zgodnie ze standardem MODBUS, do wyliczenia sumy kontrolnej CRC wykorzystano wielomian: X16 + X15 + X2 + 1.

4.4.1. Bitowy algorytm liczenia CRC:

Procedura wyznaczania sumy kontrolnej CRC metodą bitową:

a) załadowanie wartości 0xFFFF do 16-bitowego rejestru CRC;

b) pobranie pierwszego bajta z bloku danych i wykonanie operacji EX-OR z młodszym bajtem rejestru CRC, umieszczenie rezultatu w rejestrze;

c) przesunięcie zawartości rejestru CRC w prawo o jeden bit w kierunku najmniej znaczącego bitu (LSB), wyzerowanie najbardziej znaczącego bitu (MSB);

d) sprawdzenie stanu najmłodszego bitu (LSB) w rejestrze CRC, jeżeli jego stan równa się 0, to następuje powrót do punktu c, jeżeli 1, to wykonywana jest operacja EX-OR rejestru CRC ze stałą 0xA001;

e) powtórzenie punktów c i d do ośmiu razy, co odpowiada przetworzeniu całego bajta;

f) powtórzenie sekwencji b, c, d, e dla kolejnego bajta wiadomości, kontynuacja tego procesu aż do przetworzenia wszystkich bajtów wiadomości;

g) zawartość rejestru CRC po wykonaniu wymienionych operacji jest poszukiwaną wartością sumy kontrolnej CRC;

h) dopisanie sumy kontrolnej CRC do ramki MODBUS RTU musi zostać poprzedzone zamianą miejscami starszego i młodszego bajta rejestru CRC.

4.4.2. Tablicowy algorytm liczenia CRC:

Przykład implementacji procedury wyznaczania sumy kontrolnej CRC metodą tablicową:

```
/* The function returns the CRC as a unsigned short type */
unsigned short CRC16 ( puchMsg, usDataLen )
/* message to calculate CRC upon */
unsigned char *puchMsg ;
/* quantity of bytes in message */
unsigned short usDataLen ;

{
        /* high byte of CRC initialized */
        unsigned char uchCRCHi = 0xFF ;
        /* low byte of CRC initialized */
        unsigned char uchCRCLo = 0xFF ;
        /* will index into CRC lookup table */
        unsigned uIndex ;

        /* pass through message buffer */
        while (usDataLen--)
        {
                /* calculate the CRC */
                uIndex = uchCRCLo ^ *puchMsg++ ;
                uchCRCLo = uchCRCHi ^ auchCRCHi[uIndex] ;
                uchCRCHi = auchCRCLo[uIndex] ;
        }
return (uchCRCHi << 8 | uchCRCLo) ;
}
```

```
/* Table of CRC values for high-order byte */
static unsigned char auchCRCHi[] = {
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81,
0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0,
0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01,
0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41,
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81,
0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0,
0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01,
0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40,
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81,
0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0,
0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01,
0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81,
0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0,
0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01,
0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81,
0x40
} ;

/* Table of CRC values for low-order byte */
static char auchCRCLo[] = {
0x00, 0xC0, 0xC1, 0x01, 0xC3, 0x03, 0x02, 0xC2, 0xC6, 0x06, 0x07, 0xC7, 0x05, 0xC5, 0xC4,
0x04, 0xCC, 0x0C, 0x0D, 0xCD, 0x0F, 0xCF, 0xCE, 0x0E, 0x0A, 0xCA, 0xCB, 0x0B, 0xC9, 0x09,
0x08, 0xC8, 0xD8, 0x18, 0x19, 0xD9, 0x1B, 0xDB, 0xDA, 0x1A, 0x1E, 0xDE, 0xDF, 0x1F, 0xDD,
0x1D, 0x1C, 0xDC, 0x14, 0xD4, 0xD5, 0x15, 0xD7, 0x17, 0x16, 0xD6, 0xD2, 0x12, 0x13, 0xD3,
0x11, 0xD1, 0xD0, 0x10, 0xF0, 0x30, 0x31, 0xF1, 0x33, 0xF3, 0xF2, 0x32, 0x36, 0xF6, 0xF7,
0x37, 0xF5, 0x35, 0x34, 0xF4, 0x3C, 0xFC, 0xFD, 0x3D, 0xFF, 0x3F, 0x3E, 0xFE, 0xFA, 0x3A,
0x3B, 0xFB, 0x39, 0xF9, 0xF8, 0x38, 0x28, 0xE8, 0xE9, 0x29, 0xEB, 0x2B, 0x2A, 0xEA, 0xEE,
0x2E, 0x2F, 0xEF, 0x2D, 0xED, 0xEC, 0x2C, 0xE4, 0x24, 0x25, 0xE5, 0x27, 0xE7, 0xE6, 0x26,
0x22, 0xE2, 0xE3, 0x23, 0xE1, 0x21, 0x20, 0xE0, 0xA0, 0x60, 0x61, 0xA1, 0x63, 0xA3, 0xA2,
0x62, 0x66, 0xA6, 0xA7, 0x67, 0xA5, 0x65, 0x64, 0xA4, 0x6C, 0xAC, 0xAD, 0x6D, 0xAF, 0x6F,
0x6E, 0xAE, 0xAA, 0x6A, 0x6B, 0xAB, 0x69, 0xA9, 0xA8, 0x68, 0x78, 0xB8, 0xB9, 0x79, 0xBB,
0x7B, 0x7A, 0xBA, 0xBE, 0x7E, 0x7F, 0xBF, 0x7D, 0xBD, 0xBC, 0x7C, 0xB4, 0x74, 0x75, 0xB5,
0x77, 0xB7, 0xB6, 0x76, 0x72, 0xB2, 0xB3, 0x73, 0xB1, 0x71, 0x70, 0xB0, 0x50, 0x90, 0x91,
0x51, 0x93, 0x53, 0x52, 0x92, 0x96, 0x56, 0x57, 0x97, 0x55, 0x95, 0x94, 0x54, 0x9C, 0x5C,
0x5D, 0x9D, 0x5F, 0x9F, 0x9E, 0x5E, 0x5A, 0x9A, 0x9B, 0x5B, 0x99, 0x59, 0x58, 0x98, 0x88,
0x48, 0x49, 0x89, 0x4B, 0x8B, 0x8A, 0x4A, 0x4E, 0x8E, 0x8F, 0x4F, 0x8D, 0x4D, 0x4C, 0x8C,
0x44, 0x84, 0x85, 0x45, 0x87, 0x47, 0x46, 0x86, 0x82, 0x42, 0x43, 0x83, 0x41, 0x81, 0x80,
0x40
};
```