通信仕様書

放射温度計（TMHX、FTKX各シリーズ用）

サーモスポットセンサー

ＲＳ２３２Ｃ

センサヘッド／温度変換器ソフト　　Ｖｅｒ４．２４以降

ジャパンセンサー　株式会社

2013年10月28日版

目次

概要

・TMHX、FTKX シリーズは RS232C インターフェースによる通信機能を備えており、
弊社の表示設定器、パラメーター設定セット等との通信が可能です。
通信プロトコルはバイナリデータ通信による MODBUS RTU プロトコルを使用しております。

通信仕様

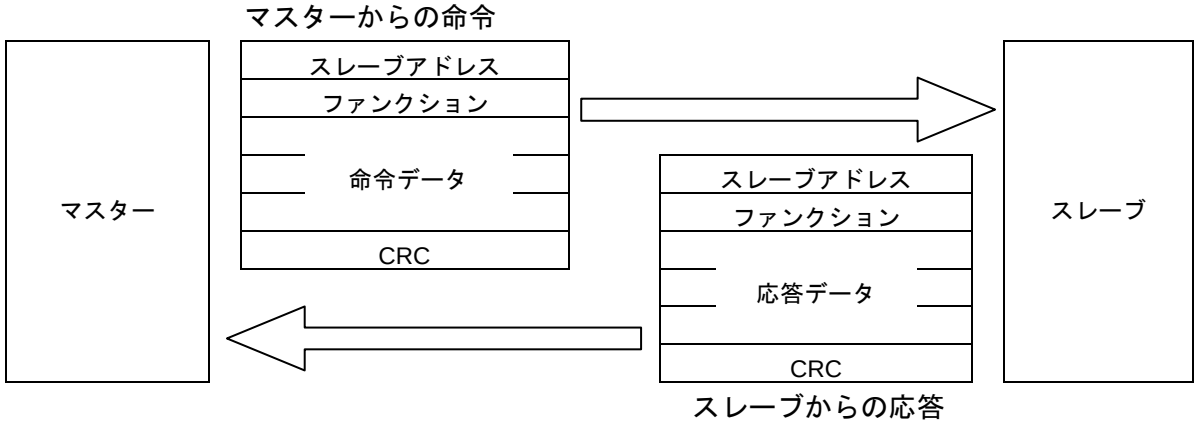
通信方式	ＲＳ２３２Ｃ準拠
接続方式	ﾎﾟｲﾝﾄ ﾄｩ ﾎﾟｲﾝﾄ
通信方式	３線式半二重
通信距離	５０ｍ（ｍａｘ）
出力スイング幅	±４Ｖ
信号内容	ＴＸＤ（送信データ）およびＲＸＤ（受信データ）
最大接続数	１
同期方式	調歩同期式
通信速度(BPS)	４８００，９６００、１９２００，３８４００，５７６００，１１５２００
データ長	８ｂｉｔ
パリティー	偶数、奇数、パリティーなし
ストップビット長	１ｂｉｔまたは ２ｂｉｔ
通信符号	バイナリ
プロトコル	ＭＯＤＢＵＳ　ＲＴＵ
通信データ	温度測定値、アラーム出力状態、パラメータ設定値

工場出荷時設定

通信速度（BPS）	9600BPS
パリティービット	なし
ストップビット	2bit
データ長	8bit

MODBUS RTU プロトコル

・MODBUS RTU プロトコルの通信方式はシングルマスター/マルチスレーブ方式です。
マスター（表示設定器等）だけがスレーブ（温度計）に命令を送信できます。スレーブはマスターからの命令に従って指定された機能を実行し、応答メッセージをマスターに返します。TMHX、FTKX の通信媒体は RS232C なので、マスターとスレーブで 1 対 1 通信を行います。



通信データフォーマット

メッセージの構成

Start	Slave Address	Function	Data	CRC	End
※1	8bit	8bit	N*8bit	16bit	※1

※1　Start と End は３．５文字分の無通信時間

Slave Address・ スレーブアドレスを指定します。TMHX、FTKX は 1 対 1 通信なので
アドレスは常に“０１”（HEX）です。

Function・・・・Funcion はスレーブに対する動作の種類を指示するコードです。
TMHX、FTKX では以下の２つを使用します。

Function コード（HEX）	詳細
03	スレーブの設定値、情報の読み取り
06	スレーブの書き込み

Data・・・・・・Function コードに関するデータを送信する場合に使用します。
レジスタアドレス、設定値、受信データ、エラーコード等

CRC・・・・・・ModBus RTU のエラーチェックは CRC（周期冗長検査）と言われる計算方法で計算されます。計算された 16bit のデータは 2 つの 8bit データで表します。CRC の算出方法は 13 ページ 『CRC-16 の算出』を参照下さい。

通信手順

・リードコマンドは単一のレジスタあるいは連続する複数のレジスタを一度に読み込むことができます。
・セットコマンドは単一のレジスタを書き込みます。複数のレジスタ書き込みには対応していません。

単一レジスタのリードコマンド / レスポンス

例）レジスタアドレス 0100 H（測定温度）をリードする例です。

リードコマンド			
データ番号	フィールド名	説明	データ例(HEX)
0	Slave Address	スレーブアドレス （1 対 1 通信なので常に 01H）	01
1	Function	Function コード （読込 03 H）	03
2	Starting Register Address　Hi	先頭レジスタアドレス　上位	01
3	Starting Register Address　Lo	先頭レジスタアドレス　下位	00
4	No. of Registers　Hi	読込レジスタ数　上位	00
5	No. of Registers　Lo	読込レジスタ数　下位	01
6	CRC16　Lo	0～5 のデータから計算した CRC　下位	85
7	CRC16　Hi	0～5 のデータから計算した CRC　上位	F6

温度計の応答：測定温度 23.5℃（00EB H）

レスポンス			
データ番号	フィールド名	説明	データ例(HEX)
0	Slave Address	スレーブアドレス （1 対 1 通信なので常に 01 H）	01
1	Function	Function コード （読込 03 H）	03
2	Byte Count	応答データ数(データ番号 3～4 迄)	02
3	測定温度　Hi	0100H に対する応答　上位	00
4	測定温度　Lo	0100H に対する応答　下位	EB
5	CRC16　Lo	0～8 のデータから計算した CRC　下位	F8
6	CRC16　Hi	0～8 のデータから計算した CRC　上位	0B

数字末尾の H は 16 進数を表しています。

複数レジスタのリードコマンド / レスポンス

例）レジスタアドレス 0100Hから 3 レジスタをリードする例です。

リードコマンド			
データ番号	フィールド名	説明	データ例(HEX)
0	Slave Address	スレーブアドレス （1 対 1 通信なので常に 01H）	01
1	Function	Function コード （読込 03H）	03
2	Starting Register Address　Hi	先頭レジスタアドレス　上位	01
3	Starting Register Address　Lo	先頭レジスタアドレス　下位	00
4	No. of Registers　Hi	読込レジスタ数　上位	00
5	No. of Registers　Lo	読込レジスタ数　下位	03
6	CRC16　Lo	0～5 のデータから計算した CRC　下位	04
7	CRC16　Hi	0～5 のデータから計算した CRC　上位	37

温度計の応答 : 測定温度 23.5℃（00EBH）、ステータス 0000H、測定温度ホールド値 23.5℃（00EBH）

レスポンス			
データ番号	フィールド名	説明	データ例(HEX)
0	Slave Address	スレーブアドレス （1 対 1 通信なので常に 01H）	01
1	Function	Function コード （読込 03H）	03
2	Byte Count	応答データ数(データ番号 3～8 迄)	06
3	測定温度　Hi	0100H に対する応答　上位	00
4	測定温度　Lo	0100H に対する応答　下位	EB
5	ステータス　Hi	0101H に対する応答　上位	00
6	ステータス　Lo	0101H に対する応答　下位	00
7	測定温度　ホールド値　Hi	0102H に対する応答　上位	00
8	測定温度　ホールド値　Lo	0102H に対する応答　下位	EB
9	CRC16　Lo	0～8 のデータから計算した CRC　下位	45
10	CRC16　Hi	0～8 のデータから計算した CRC　上位	2D

数字末尾の Hは 16 進数を表しています。

単一レジスタのセットコマンド / レスポンス

例）レジスタアドレス 0300ｈ（放射率設定）に 0.950（03B6ｈ）をセットする例です。

リードコマンド			
データ番号	フィールド名	説明	データ例(HEX)
0	Slave Address	スレーブアドレス （1 対 1 通信なので常に 01ｈ）	01
1	Function	Function コード （書込 06ｈ）	06
2	Register Address　Hi	書込するレジスタアドレス　上位	03
3	Register Address　Lo	書込するレジスタアドレス　下位	00
4	Set Value　Hi	設定する値　上位	03
5	Set Value　Lo	設定する値　下位	B6
6	CRC16　Lo	0～5 のデータから計算した CRC　下位	08
7	CRC16　Hi	0～5 のデータから計算した CRC　上位	C8

正常レスポンスはセットコマンドと同一電文になります。

レスポンス			
データ番号	フィールド名	説明	データ例(HEX)
0	Slave Address	スレーブアドレス （1 対 1 通信なので常に 01ｈ）	01
1	Function	Function コード （書込 06ｈ）	06
2	Register Address　Hi	書込するレジスタアドレス　上位	03
3	Register Address　Lo	書込するレジスタアドレス　下位	00
4	Set Value　Hi	設定する値　上位	03
5	Set Value　Lo	設定する値　下位	B6
6	CRC16　Lo	0～5 のデータから計算した CRC　下位	08
7	CRC16　Hi	0～5 のデータから計算した CRC　上位	C8

数字末尾の ｈ は 16 進数を表しています。

リードコマンド / セットコマンドに対する異常レスポンス

例）リードコマンドに対して Illegal register address エラーが発生した例です。

異常レスポンス			
データ番号	フィールド名	説明	データ例(HEX)
0	Slave Address	スレーブアドレス （1 対 1 通信なので常に 01 H）	01
1	Function ｜ 80H	Function コードに 80 H を 加算します	83 （03 ｜80）
2	Error Code	エラーの種類 12 ページを参照下さい	02
3	CRC16 Lo	0～5 のデータから計算した CRC 下位	C0
4	CRC16 Hi	0～5 のデータから計算した CRC 上位	F1

数字末尾の H は 16 進数を表しています。

レジスター覧

データの表現

・測定値など温度データはすべてオフセットバイナリ（10 倍データ）の「℃」でやり取りします。

例）30000 カウント　→　30000÷10＝3000.0℃

16 進数表現　0000 H＝0.0℃　7 FFF H＝3276.7℃　8000 H＝－3276.8℃　FFFF H＝－0.1℃

レジスタアドレス	対応 Function	機能	データ範囲
0040 H	03 H	温度計 F/W バージョン	
0100 H	03 H	測定温度	最高表示温度～最低表示温度 最高表示温度以上の場合 7FFFH を返信 最低表示温度以下の場合 8000H を返信
0101 H	03 H	ステータス	ステータス割り付け表(12 ページ)参照
0102 H	03 H	測定温度ホールドされた値	最高表示温度～最低表示温度 最高表示温度以上の場合 7FFFH を返信 最低表示温度以下の場合 8000H を返信
0103 H	03 H	放射率設定	50 (0.050) ～ 1000 (1.000)
0120 H	03 H / 06 H	アラーム H 設定値	最高表示温度～アラーム L 設定値
0121 H	03 H / 06 H	アラーム L 設定値	アラーム H 設定値～最低表示温度
0122 H	03 H / 06 H	ピークホールド ON/OFF	0 (OFF) 1 (ON)
0123 H	03 H / 06 H	サンプルホールド ON/OFF	0 (OFF) 1 (ON)
0124 H	03 H / 06 H	スムージング時間	0 (0.0001sec) 1 (0.0002) 2 (0.0005) 3 (0.001) 4 (0.002) 5 (0.005) 6 (0.01) 7 (0.02) 8 (0.05) 9 (0.1) 10 (0.2) 11 (0.5) 12 (1) 13 (2) 14 (5)
0125 H	03 H / 06 H	照準ランプ ON/OFF	0 (OFF) 1 (ON)

数字末尾の Hは 16 進数を表しています。

レジスタ アドレス	対応 Function	機能	データ範囲
0201 H	03 H / 06 H	外部制御ピン	0 (アラーム出力) 1 (外部照準制御)
0202 H	03 H / 06 H	照準ランプ制御	0 (常時 OFF) 1 (ON/OFF) 2 (常時 ON)
0208 H	03 H / 06 H	室温反射補正有無	0 (OFF) 1 (ON)
0209 H	03 H / 06 H	アラームヒステリシス幅	0 (0.0) ～ 999 (99.9)
020A H	03 H / 06 H	アラームモード	0 (アラームモード 1) 1 (アラームモード 2) 2 (アラームモード 3) 3 (アラームモード 4) 4 (アラームモード 5) 5 (アラームモード 6) 6 (アラームモード 7) 7 (アラームモード 8)
020C H	03 H / 06 H	アラーム判定切換	0 (リアル) 1 (ホールド)
020D H	03 H / 06 H	ピークホルードリセット方式	0 (時間) 1 (外部) 2 (放電)
020E H	03 H / 06 H	ピークホールドリセット時間 または放電時間	1 (0.01 秒) ～ 1000(10.00 秒)
020F H	03 H / 06 H	ピークホールド放電レベル	20 (0.20) ～ 100(1.00)
0210 H	03 H / 06 H	ピークホールド出力	0 (表示のみ) 1 (表示＋アナログ出力) 2 (全部) 3 (アナログ出力のみ)
0211 H	03 H / 06 H	ピークホールド 外部タイミング入力の極性	0 (＋) 1 (－)
0212 H	03 H / 06 H	サンプルホールド出力	0 (表示のみ) 1 (表示＋アナログ出力) 2 (全部) 3 (アナログ出力のみ)
0213 H	03 H / 06 H	サンプルホールド 外部タイミング入力の極性	0 (＋) 1 (－)

数字末尾の Hは 16 進数を表しています。

レジスタアドレス	対応Function	機能	データ範囲
0214H	03H / 06H	アナログ出力　出力タイプ	0 (4～20mA) 1 (0～20mA) 2 (0～1V) 3 (mV/℃) 4 (なし)
0215H	03H / 06H	アナログ出力　上限温度	3276.7℃～アナログ出力下限温度
0216H	03H / 06H	アナログ出力　下限温度	アナログ出力上限温度～－200℃
0217H	03H / 06H	エラー時アナログ出力極性	0 (無処理) 1 (High) 2 (Low)
021BH	03H / 06H	センサー補正スパン	500 (0.500) ～ 2000 (2.000)
021CH	03H / 06H	センサー補正オフセット	－500 (－50.0) ～ 500 (50.0)
021EH	03H / 06H	通信速度	0 (1200bps) 1 (2400bps) 2 (4800bps) 3 (9600bps) 4 (19200bps) 5 (38400bps) 6 (57600bps) 7 (115200bps)
021FH	03H / 06H	パリティービット	0 (なし) 1 (偶数) 2 (奇数)
0220H	03H / 06H	ストップビット	0 (1bit) 1 (2bit)
0300H	03H / 06H	放射率設定 （コマンド毎にメモリに保存)	50 (0.050) ～ 1000 (1.000)
0301H	03H / 06H	放射率設定 （メモリに保存しない)	50 (0.050) ～ 1000 (1.000)
0302H	06H	自動放射率設定	最高表示温度～最低表示温度 (最低表示温度が 50℃以下の場合は 50℃迄)
0303H	03H / 06H	放射率反射補正値 （コマンド毎にメモリに保存)	－1999 ～ 9999
0304H	03H / 06H	放射率反射補正値 （メモリに保存しない)	－1999 ～ 9999
0305H	06H	自動放射率反射補正値	最高表示温度～最低表示温度
0306H	06H	イニシャライズ	1(イニシャライズ実行)

数字末尾の H は 16 進数を表しています。

ステータス割り付け表

MSB LSB

8 4 2 1	8 4 2 1	8 4 2 1	8 4 2 1
ピークホールド状態	サンプルホールド状態	アラーム状態	±OVER の有無

4bit づつ割り付け

・ピークホールド状態・・・・・(0：なし　1：表示ホールド発生中　2：アナログ出力ホールド発生中　3：1+2 発生中)

・サンプルホールド状態・・・・(0：なし　1：サンプルホールド発生中)

・アラーム状態・・・・・・・・(0：なし　1：アラーム発生中)

・±OVER の有無　・・・・・・(0：なし　1：温度が最高表示温度以上　2：温度が最低表示温度以下)

例)

0000　0000　0001　0010（0012ｈ）

ピークホールドなし、サンプルホールドなし、アラーム発生中、 - OVER 発生中

エラーコード

通信エラー時の応答

エラーコード	名称	内容
01ｈ	Illegal function	Function コードが未定義
02ｈ	Illegal register address	レジスタ番号が未定義
03ｈ	Illegal data value	設定値が設定範囲外
80ｈ	自動設定エラー	目標温度に設定できなかった

温度計エラー時の応答

エラーコード	名称	内容
13ｈ	内部電圧異常	温度計内部の電圧低下

数字末尾の ｈ は 16 進数を表しています。

CRC-16 の算出

エラーチェックは、スレーブアドレスからデータの最後までの CRC - 16 を計算し、
算出した 16 ビットデータを下位上位の順にデータの後ろにセットします。

CRC - 16 の計算方法

CRC のデータを X とします。

1）X に FFFFH を代入します。
2）１つ目のデータと X の排他的論理和（XOR）を取り、X に代入します。
3）X の右端のビットが１の場合、X を右に１ビットシフトした後 A001H で XOR を取り、X に代入します。
　X の右端のビットが０の場合、X を右に１ビットシフトして ４）に進みます。
4）８回シフトするまで３）を繰り返します。
5）次のデータと X の XOR をとり、X に代入します。
6）最後のデータまで３）～５）を繰り返します。
7）X を CRC - 16 としてデータの後ろに下位上位の順でセットします。

CRC-16 計算サンプルプログラム

RxData[]には受信データが格納されているものとします。

```
uint8_t RxData[];
uint16_t crc16;
uint8_t *cptr1
uint16_t NoOfByte = 6;
int16_t carry;
int16_t i;
int16_t j;

cptr1 = RxData;
crc16 = 0xffff;
for(i=0;i<NoOfByte;i++){
   crc16=crc16^(uint8_t)(*(cptr1+i));
   for(j=0;j<8;j++){
      if(crc16 & 0x0001){
         carry=1;
      }else{
         carry=0;
      }
      crc16 = crc16>>1;
      if(carry){
         crc16=crc16^0xa001;
      }
   }
}
```

RS232C 通信サンプルコード　VB.NET

```
'*********************************************************************************************
'*                                                                                           *
'*      ＴＭＨＸ　ＲＳ２３２Ｃ通信 サンプルコード　VB.NET                                    *
'*                                                                                           *
'*********************************************************************************************
Imports System.IO.Ports          'serial port で使う

Public Class Form1

    Private port As New SerialPort("COM4", 9600, 0, 8, 2) 'シリアルポート宣言

    '/*******************************************************************************************
    '/        測定温度データ読込
    Private Sub TempRead()
      '送信処理
        Dim CRC As Long
        Dim SendByte() As Byte = New Byte(7) {}
        SendByte(0) = &H1 'スレーブアドレス
        SendByte(1) = &H3 'Functionコード 読込
        SendByte(2) = &H1 '先頭レジスタアドレス　測定温度　上位
        SendByte(3) = &H0 '先頭レジスタアドレス　測定温度　下位
        SendByte(4) = &H0 '読込レジスタ数　上位
        SendByte(5) = &H1 '読込レジスタ数　下位
        CRC = CrcY(SendByte, 6) '送信データのCRCを計算
        SendByte(6) = CByte(CRC And &HFF) 'CRC Highバイト
        SendByte(7) = CByte(CRC >> 8) 'CRC Lowバイト
        Try
            port.Open() '通信ポートを開く
            port.DiscardOutBuffer() '送信バッファのクリア
            port.DiscardInBuffer() '受信バッファのクリア
            port.Write(SendByte, 0, 8) '送信する
            Debug.WriteLine(BitConverter.ToString(SendByte)) 'イミディエイトウィンドウに送信データを表示する
        Catch ex As Exception
            MessageBox.Show(ex.Message)
            Exit Sub
        End Try
      '受信処理
        Dim length As Integer = 7 '受信バイト数
        Dim ReceivedByte() As Byte = New Byte(length - 1) {}   '受信データ用変数
        Dim ct As Integer 'タイムアウト用カウンタ
        Do
            System.Threading.Thread.Sleep(10) '10mSタイマ
            ct += 1
            If ct > 10 Then Exit Do '100mS以上ならタイムアウト
            length = port.BytesToRead '受信バイト数を確認
        Loop Until length >= 7 '受信バイト数が 7 になるまで待つ
        Try
            System.Threading.Thread.Sleep(10) '10mSタイマ
            length = port.Read(ReceivedByte, 0, length) '受信データのReceivedByteへの読込
            If length <= 0 Then
                port.Close() '通信ポートを閉じる
                'エラー処理　省略
                Exit Sub
            End If
            'イミディエイトウィンドウに送信データを表示する
            Debug.WriteLine(BitConverter.ToString(ReceivedByte))
         Catch ex As Exception
            MessageBox.Show(ex.Message)
        End Try
        port.Close() '通信ポートを閉じる
```

```
        '受信データ処理
        CRC = CrcY(ReceivedByte, length - 2) '受信データのCRCを計算
        If CByte(CRC >> 8) <> ReceivedByte(length - 1) Or CByte(CRC And &HFF) <> ReceivedByte(length - 2) Then
            'CRCエラー処理　省略
            Exit Sub
        End If
        'コンピューターのアーキテクチャが、リトル エンディアンかビッグ エンディアンかにより
        '上位バイトと下位バイトの並び順を変える。
        Dim Temp As Byte() = New Byte(1) {}
        If BitConverter.IsLittleEndian Then
            Temp(0) = ReceivedByte(4)
            Temp(1) = ReceivedByte(3)
        Else
            Temp(0) = ReceivedByte(3)
            Temp(1) = ReceivedByte(4)
        End If
        Dim readValue As Short = BitConverter.ToInt16(Temp, 0)
        '受信された測定温度を整数⇒小数変換し、イミディエイトウィンドウに表示
        Debug.WriteLine(readValue / 10)
        If readValue = 32767 Then
            '最高表示温度以上(&H7FFF)のエラー処理　省略
            Exit Sub
        End If
        If readValue = -32768 Then
            '最低表示温度(&H8000)以下のエラー処理　省略
            Exit Sub
        End If
        If ReceivedByte(1) >= &H80 Then 'ファンクションは、エラーか？
            'エラー処理　省略
        End If
    End Sub

    '/*************************************************************************************************
    '/        '放射率書込

    Private Sub EmissWrite()
      '送信処理
        Dim CRC As Long
        Dim SendByte() As Byte = New Byte(7) {}
        SendByte(0) = &H1 'スレーブアドレス
        SendByte(1) = &H6 'Functionコード 書込み
        SendByte(2) = &H3 '書込するレジスタアドレス　放射率　上位
        SendByte(3) = &H0 '書込するレジスタアドレス　放射率　下位
        Dim Emiss As Single = 0.95 '書き込み値をセット 放射率 0.95
        Dim writeValue As Short = Emiss * 1000 '小数⇒整数変換
        Dim writeByte As Byte() = BitConverter.GetBytes(writeValue)
        If BitConverter.IsLittleEndian Then
            SendByte(4) = writeByte(1) '上位
            SendByte(5) = writeByte(0) '下位
        Else
            SendByte(4) = writeByte(0) '上位
            SendByte(5) = writeByte(1) '下位
        End If
        CRC = CrcY(SendByte, 6) '送信データのCRCを計算
        SendByte(6) = CByte(CRC And &HFF) 'CRC Highバイト
        SendByte(7) = CByte(CRC >> 8) 'CRC Lowバイト
        Try
            port.Open() '通信ポートを開く
            port.DiscardOutBuffer() '送信バッファのクリア
            port.DiscardInBuffer() '受信バッファのクリア
            port.Write(SendByte, 0, 8) '送信する
```

```
            'イミディエイトウィンドウに送信データを表示する
            Debug.WriteLine(BitConverter.ToString(SendByte))
        Catch ex As Exception
            MessageBox.Show(ex.Message)
            Exit Sub
        End Try
    '受信処理
        Dim length As Integer = 8 '受信バイト数
        Dim ReceivedByte() As Byte = New Byte(length - 1) {}    '受信データ用変数
        Dim ct As Integer 'タイムアウト用カウンタ
        Do
            System.Threading.Thread.Sleep(10) '10mSタイマ
            ct += 1
            If ct > 10 Then Exit Do '100mS以上ならタイムアウト
            length = port.BytesToRead '受信バイト数を確認
        Loop Until length >= 8 '受信バイト数が 8 になるまで待つ
        Try
            System.Threading.Thread.Sleep(10) '10mSタイマ
            length = port.Read(ReceivedByte, 0, length) '受信データのReceivedByteへの読込
            If length <= 0 Then
                port.Close() '通信ポートを閉じる
                'エラー処理　省略
                Exit Sub
            End If
            'イミディエイトウィンドウに送信データを表示する
            Debug.WriteLine(BitConverter.ToString(ReceivedByte))
            txtRecieve.Text = BitConverter.ToString(ReceivedByte)
            txtInput.Text = Mid(txtRecieve.Text, 13, 5)
        Catch ex As Exception
            MessageBox.Show(ex.Message)
        End Try
        port.Close() '通信ポートを閉じる
        '送信データと受信データが同じでなければ、エラー処理
        If System.Linq.Enumerable.SequenceEqual(SendByte, ReceivedByte) = False Then
            'エラー処理　省略
        End If
End Sub

Private Const CRC_POLYNOM As Long = &HA001&         'CRC多項式 = x16+x12+x5+1
Private Const CRC_PRESET As Long = &HFFFF&          'CRCプリセット

'/*****************************************************************************************
'                   CRC16 計算
Public Function CrcY(ByRef dat() As Byte, No As Long) As Long
    Dim i As Long
    Dim j As Long

    CrcY = CRC_PRESET
    For i = 0 To No - 1
        CrcY = CrcY Xor CLng(dat(i))
        For j = 0 To 7
            If (CrcY And &H1&) <> 0 Then
                CrcY = (CrcY ¥ 2) Xor CRC_POLYNOM
            Else
                CrcY = CrcY ¥ 2
            End If
        Next j
    Next i
End Function
```