

SD330自動同調システム ソフトウェア説明書

HOST編(VB.NET)

2016年4月1日

Narichan

1、プログラムの概要

ホスト側のプログラムです。マイクロソフトが無償提供しているExpress2013 for Windows DesktopにてVB.NETで開発しました。理由は無償、使い慣れているなどからです。

プログラムはMPUに対してコマンドを送ります。MPUからの返信を処理します。単純なソフトです。コマンドはボタンに配置しクリックされたらMPUに送るだけです。必ずホストから始まります。

周波数などのデータを送るのはMPUからFコマンドが受信された時です。よく使う周波数などはコンボボックスやリストボックスにセットし操作性を良くしています。

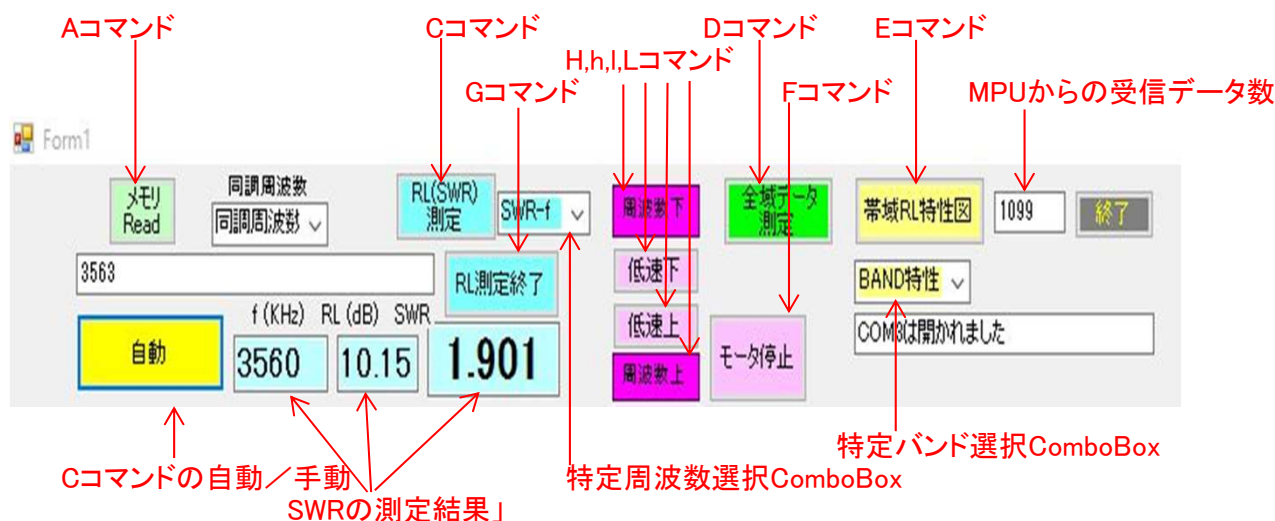
VB.NETのシステムはイベントドリブンなシステムなのでアイドルグループなど記述の必要はありません。

簡単なソフトですが難しいのはserialportの扱いです。中々思うように動いてくれません。

SWR特性はMS-CHARTを使っています。MPUからAD値が送られてきますので電圧(dBm)に変換しリターンロスを計算し、SWRを算出します。HF全帯域の図はdBm値を表示しています。

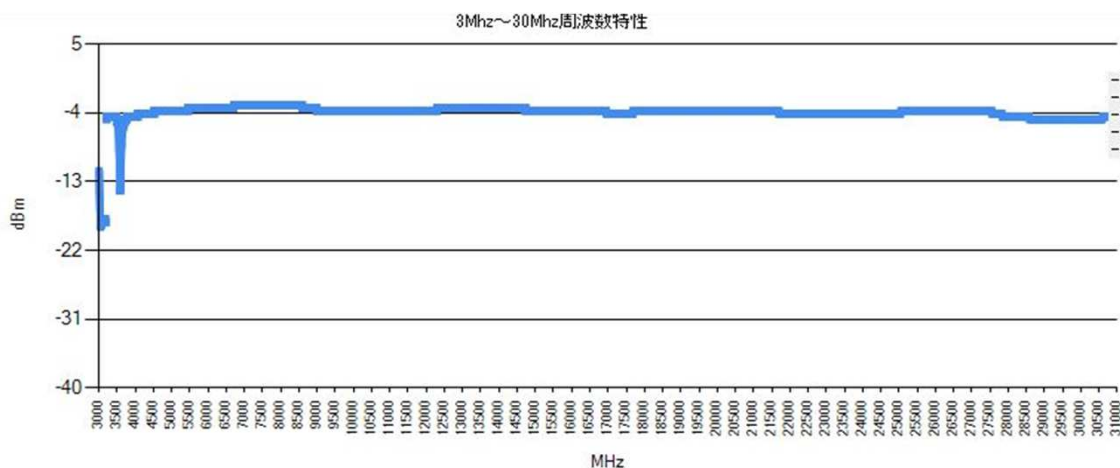
指定周波数の指定帯域の特性図はSWRを表示します。指定帯域と言ってもMPUからは1Kzスパンで1100データ来ますので必要な帯域分だけを表示します。

2、ボタン配置



3、チャート

3. 1、チャート1 (HF全帯域表示)



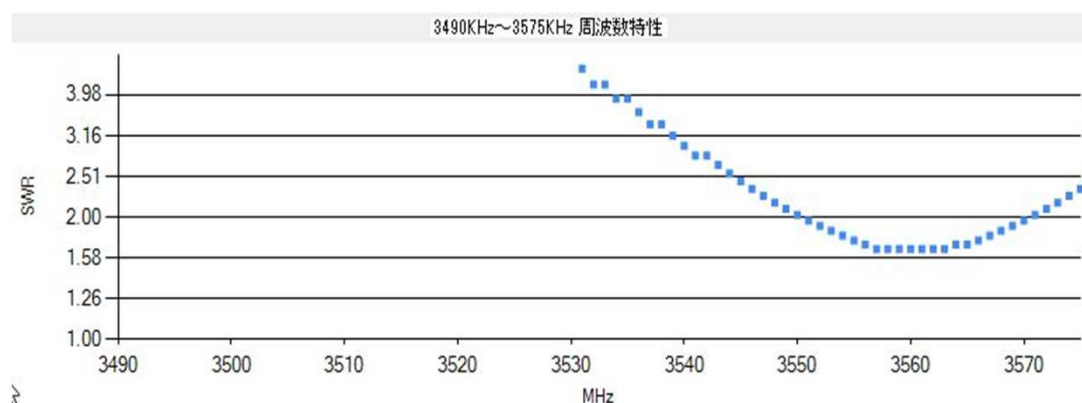
3. 2、チャート1の設定

```

For I As Integer = 0 To 0
    Chart1.Series.Add(CType(I, String))
    Chart1.Series(CType(I, String)).ChartType = DataVisualization.Charting.SeriesChartType.Point
    For k = 0 To 8599
        x = CType(ds.Tables(0).Rows(k).Item(0), Integer)
        y = CType(ds.Tables(0).Rows(k).Item(1), Single)
        Chart1.Series(CType(I, String)).Points.AddXY(x, y) 'データを設定
    Next k
    Chart1.ChartAreas(0).AxisX.MajorGrid.Enabled = False
    Chart1.ChartAreas(0).AxisX.MinorGrid.Enabled = False
    Chart1.ChartAreas(0).AxisX.Maximum = 31000
    Chart1.ChartAreas(0).AxisX.Minimum = 3000
    Chart1.ChartAreas(0).AxisX.Interval = 500
    Chart1.ChartAreas(0).AxisX.Title = "MHz"
    Chart1.ChartAreas(0).AxisY.Minimum = -40
    Chart1.ChartAreas(0).AxisY.Maximum = 5
    Chart1.ChartAreas(0).AxisY.Title = "dBm"
    Chart1.DataBind()
Next I

```

3. 3 チャート2（指定周波数、指定帯域）



3. 4チャート2の設定

```

For I As Integer = 0 To 0
    Chart2.Series.Add(CType(I, String))
    Chart2.Series(CType(I, String)).ChartType = DataVisualization.Charting.SeriesChartType.Point
    For k = 0 To 1099
        x = CType(ds1.Tables(0).Rows(k).Item(0), Integer)
        y = CType(ds1.Tables(0).Rows(k).Item(1), Integer)
        y = CType(ds1.Tables(0).Rows(k).Item(1), Single)
        Chart2.Series(CType(I, String)).Points.AddXY(x, y) 'データを設定
    Next k
    Chart2.ChartAreas(0).AxisX.MajorGrid.Enabled = False
    Chart2.ChartAreas(0).AxisX.MinorGrid.Enabled = False
    Chart2.ChartAreas(0).AxisX.Maximum = f_Tmin + f_Twidth
    Chart2.ChartAreas(0).AxisX.Minimum = f_Tmin
    Chart2.ChartAreas(0).AxisX.Interval = 10
    Chart2.ChartAreas(0).AxisX.Title = "MHz"
    Chart2.ChartAreas(0).AxisY.IsLogarithmic = True
    Chart2.ChartAreas(0).AxisY.Minimum = 1.0
    Chart2.ChartAreas(0).AxisY.Maximum = 5.0
    Chart2.ChartAreas(0).AxisY.Interval = 0.1
    Chart2.ChartAreas(0).AxisY.LabelStyle.Format = "0.##"
    Chart2.ChartAreas(0).AxisY.Title = "SWR"
    Chart2.ChartAreas(0).AxisX.Title = "MHz"
    Chart2.DataBind()
Next I

```

4、ソフトウェア説明

4. 1、主なサブルーチン・関数

(1)Private Sub Form1_Load(…)

概要 フォームロード

機能説明 シリアルポートのオープン。オープンできないときはタイマーで0.5秒ごとにポートを監視

(2)Private Sub Button3_Click(…)

概要 Cコマンド送信

機能説明 自動同調ならCコマンド送出後にモーター駆動ボタンを自動クリック
現在のアンテナ長さと移動後の長さからタイムアウト値を計算

(3)Private Function GetData() As DataSet

概要 チャート1のためのデータセット

機能説明

```

' データの設定
Private Function GetData() As DataSet
    Dim ds As New DataSet
    Dim dt As New DataTable
    Dim dtRow As DataRow
    Dim k As Integer
    Dim ks As Single
    Dim m As Integer
    Dim f_min_ALL As Integer
    dt.Columns.Add("周波数", Type.GetType("System.Int32"))
    dt.Columns.Add("AD値", Type.GetType("System.Single"))
    ds.Tables.Add(dt)
    'データの追加
    f_min_ALL = 0
    k = 1
    For i = 0 To 8599 Step 1
        dtRow = ds.Tables(0).NewRow

        If i < 6000 Then
            k = 2
            ks = (((i * 68719) + 103079214) / 34359.73836) + 0.5
            m = (i * k) + 3000
            dtRow(0) = m
        ElseIf i > 5999 And i < 8000 Then
            k = 5
            ks = (((i - 5999) * 171799) + 515396075) / 34359.73836) + 0.5
            m = ((i - 5999) * k) + 15000
            dtRow(0) = m
        ElseIf i > 7999 And i < 8600 Then
            k = 10
            ks = (((i - 7999) * 343597) + 858993459) / 34359.73836) + 0.5
            m = ((i - 7999) * k) + 25000
            dtRow(0) = m
        End If
        If (f_min_ALL > ddata_int_Fil(i)) And i > 100 Then
            f_min_ALL = ddata_int_Fil(i)
            f_min_ALL_f = dtRow(0)
        End If
        sddata_int(i) = ddata_int_Fil(i)
        dtRow(1) = ddata_int_Fil(i)
        ds.Tables(0).Rows.Add(dtRow)
    Next
    Return (ds)
End Function

```

メモリアドレス
から周波数算出
誤差を最小に
するためDDSの
設定値から算出

ここにdBmが
入っている

チャート用データセット

(4) Private Function GetData1() As DataSet

概要 チャート2のためのデータセット

機能説明

Private Function GetData1() As DataSet

Dim ds1 As New DataSet

Dim dt1 As New DataTable

Dim dt1Row As DataRow

Dim k As Integer

Dim m As Integer

Dim f_min_SML As Integer

Dim t_ As Single '反射係数

'列の作成

dt1.Columns.Add("周波数", Type.GetType("System.Int32"))

'dt1.Columns.Add("SWR値", Type.GetType("System.Int32"))

dt1.Columns.Add("SWR値", Type.GetType("System.Single"))

ds1.Tables.Add(dt1)

'データの追加

k = 1

For i = 0 To 1099 Step 1

dt1Row = ds1.Tables(0).NewRow

m = i + t_f

dt1Row(0) = m

RL_int = ddata_int_Fil(i)

'*****SWRの計算*****

RL_int = -4 - RL_int

t_ = Math.Pow(10, (RL_int / 20))

SWR_int = -1 * ((1 + t_) / (1 - t_))

'*****

'dt1Row(1) = ddata_int_Fil(i)

If SWR_int < 0 Then SWR_int = SWR_int * -1

dt1Row(1) = SWR_int

ds1.Tables(0).Rows.Add(dt1Row)

If (f_min_SML > ddata_int_Fil(i)) And (m > f_Tmin) And (m < f_Tmin + f_Twidth) Then

f_min_SML = ddata_int_Fil(i)

f_min_SML_f = dt1Row(0)

End If

Next

Return (ds1)

End Function

周波数の設定1kHzステップなので
+1していく。

リターンロスの計算

反射係数の計算

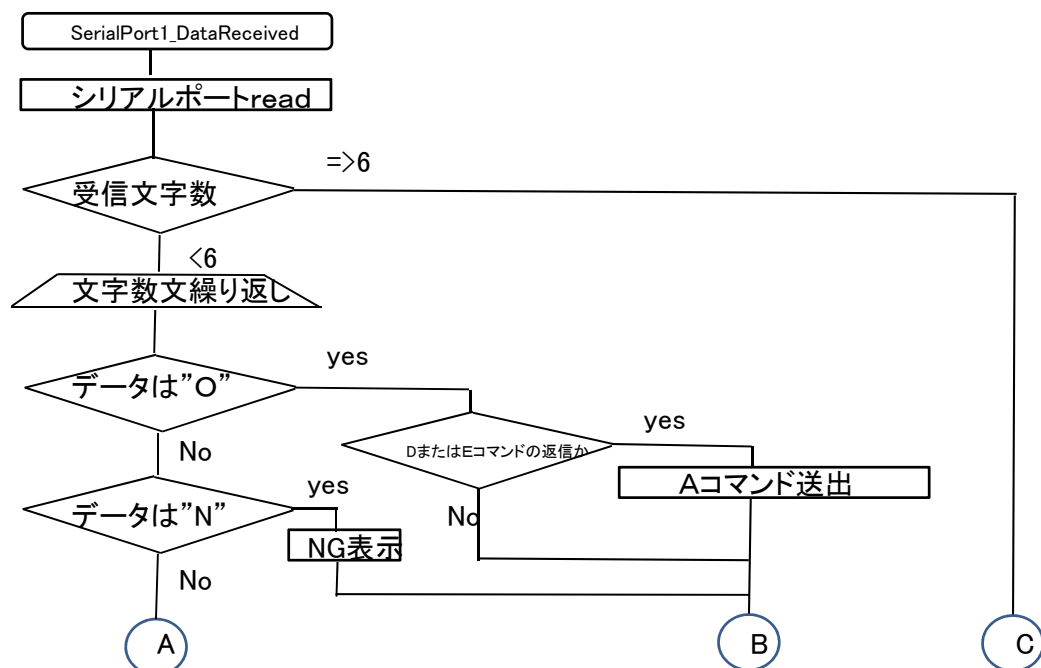
SWRの計算

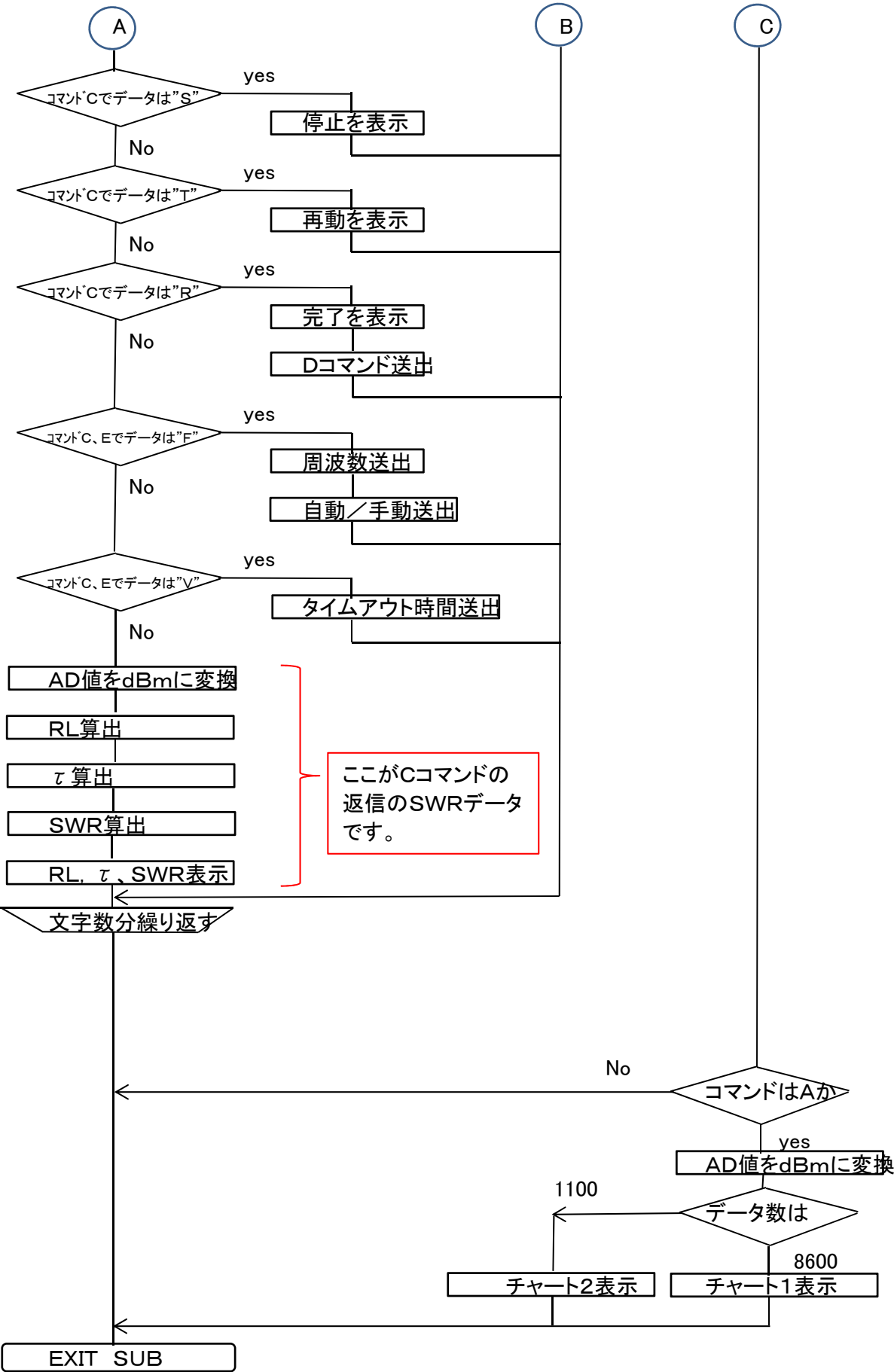
チャート用データセット

(5) Private Sub SerialPort1_DataReceived(...)

概要 シリアルポート受信

フローチャート





機能説明

受信データはWindows の中に組み込まれていてバッファ動作などかなりややこしい。そこは触らないようにして処理します。

AコマンドではMPUのメモリからデータを送信させます。したがって文字数は8600か1100のはずです。ところが、どういうわけか分割されて送られてきます。それも一意ではありません。気まぐれに分割されます。そこでトータルの文字数で処理するように作りました。

なんがかんだやっているうちに纏まって送られるようになりました。それは下記の一行を受信イベントに入れたら時でした。

```
System.Console.Beep(440, 300)
```

何が何だか狐につつまれています。兎に角動いたのでめでたしめでたしです。

Cコマンドで随時AD値を送ってくるのは文字数1ですが、その前後にOK、NGのプロトコル文字をMPUが送っているとタイミングの関係でくっついて文字数が2になったり3になったりします。

そこで6文字以内は一文字ずつ繰り返し処理します。

MPUからのコマンドは

```
// PCへ送出する返信
// 'O': OK送信
// 'N': NG送信 コマンド実行中はN送出後次のコマンド拒否
// 'F': C,Eコマンドで周波数データ(4バイト)要求
//
// 'S': Cコマンドで一旦停止を知らせる
// 'R': Cコマンド移動完了を知らせる
// 'T': Cコマンド再スタートを知らせる
// 'V': 'H,h,L,l'コマンドでタイムアウト時間 要求
// 'X': モータがタイムアウト停止を知らせる
```

これらの文字はCコマンドのADデータと同じ条件で送るためADデータの値が上記だとコマンドと判定されてしまいます。そこでMPU側で上記文字の値の場合値に0xAAを足して送信してきます。

そこでCコマンドのデータ受信で0xF0より大きい値の時は0xAAを引いています。AD値には0xF0のような大きい値は送信されないからです。

これはCコマンドの手動時のことで特段の差し障りはありませんので何もしなくても良いかもしれません。