

サンプルプログラム (簡易モニタープログラム)

1. 概要

このアプリケーションノートでは、かそくドリ・グラビー (MODEL GT-N456) と PC アプリケーションの通信方法を、Visual Basic にて作成したサンプルプログラムを例に説明したものです。

※ 以下、製品名は GT-N456 として記載しています

基本的な Visual Basic によるプログラミングについて理解されている方を対象に GT-N456 から COM 通信にてデータを取得する方法について説明を行っています。

プログラムの作成およびサンプルプログラムのプロジェクトファイルを開くにはマイクロソフト社のソフトウェア統合開発環境 Visual Studio 2008 Express Edition のインストールが必要です。インストール方法と関数・メソッドの詳細や注意事項につきましては、Microsoft 社ホームページ (MSDN <http://msdn.microsoft.com>) を参照してください。

サンプルプログラムは以下の開発ツール、環境で開発、動作確認を行っています。

OS	Windows XP Professional Service Pack2
Windows アプリケーション開発環境	Microsoft Visual Studio 2008 Express Edition Version 9.0.20729.1 SP Microsoft .NET Framework Version 3.5 SP1

※ マイクロソフト社が無償提供するプログラミングツールです

● 内容

簡易受信ログ表示のサンプルプログラムの説明は

- ・サンプルプログラムのダウンロード
- ・プロジェクトの構成
- ・サンプルプログラムの機能
- ・プログラムの内容
- ・プログラムコードの内容

となっています。

2. サンプルプログラム

サンプルプログラムについて説明します。

■サンプルプログラムのダウンロード

サンハヤト株式会社ホームページの「かそくドリ・グラビー (MODEL GT-N456) のご紹介ページ」(http://www.sunhayato.co.jp/gt-n456/gt-n456.html) より、ホームページ画面下段付近の「★ダウンロード」>「アプリケーションノートサンプルプログラム (簡易モニタープログラム)」のリンクをクリックしサンプルプログラムのプロジェクトファイルをダウンロードしてください。

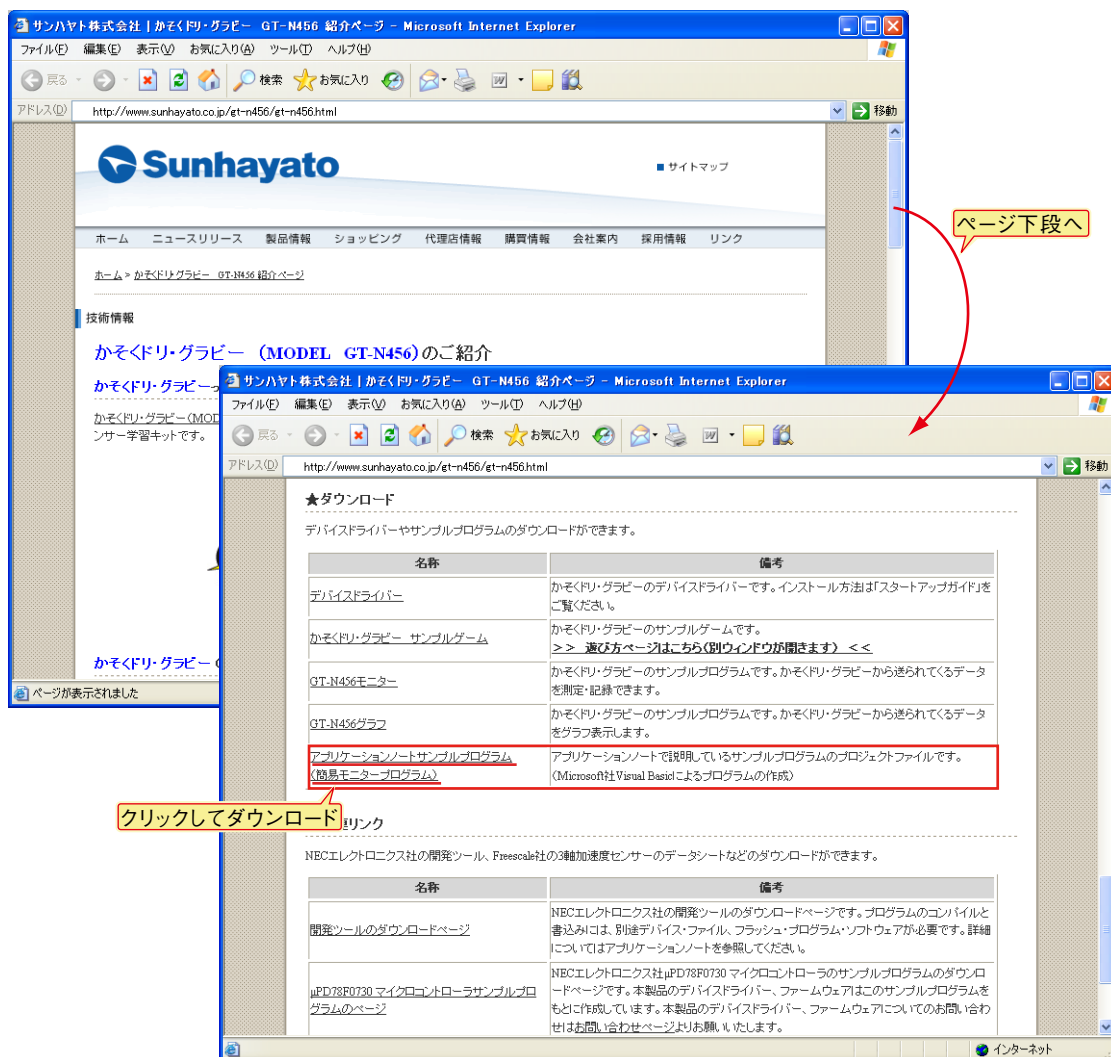


図2-1 サンプルプログラムのプロジェクトファイルをダウンロード

■ プロジェクトの構成

ダウンロードしたファイルを解凍すると「GTN456_Sample」フォルダが出力されます。

「GTN456_Sample」フォルダ以下の構成は以下のツリー図のようになっています。

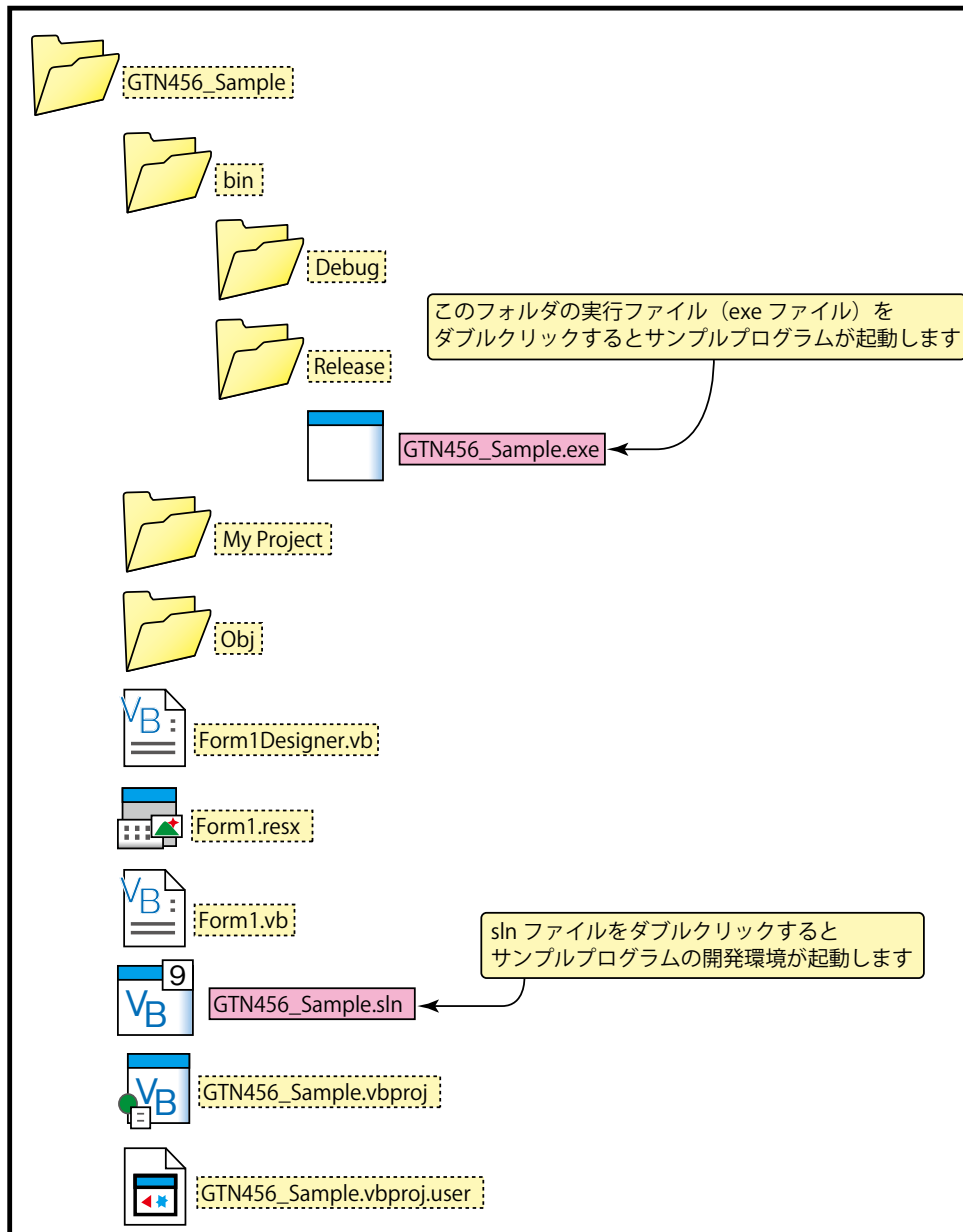


図2-2 フォルダ構成

GTN456_Sample/bin/Release フォルダ内の GTN456_Sample.exe をダブルクリックするとサンプルプログラムが起動します。

また、Visual Basic 2008 Express Edition をインストールしている PC で「GTN456_Sample.sln」をダブルクリックするとサンプルプログラムのプロジェクトが起動します。

Sunhayato

■ サンプルプログラムの機能

サンプルプログラムの機能は以下の通りです。

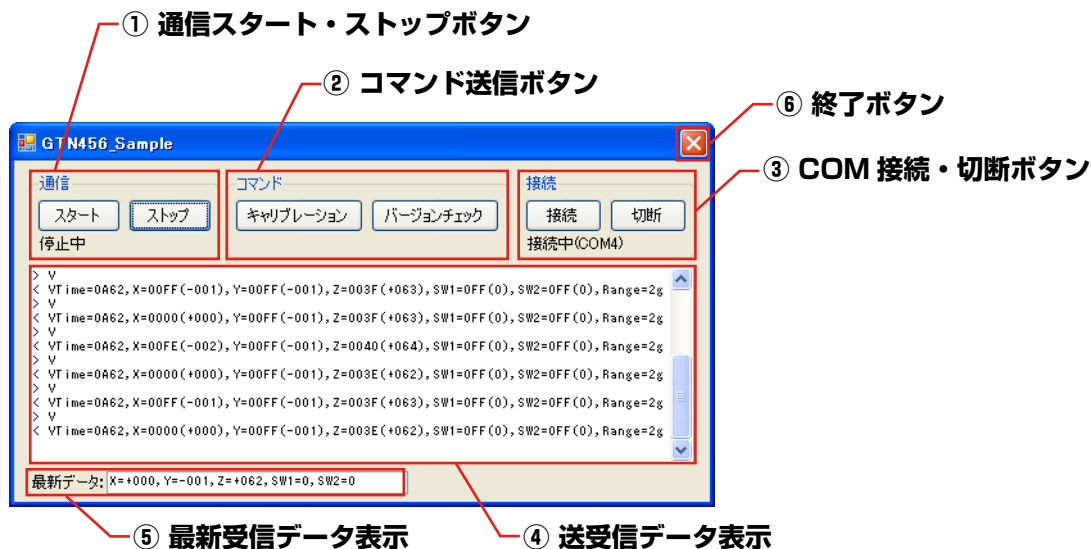


図2-3 サンプルプログラムの機能

① 通信スタート・ストップボタン

通信の開始 / 停止を行うボタンです。

「スタート」ボタン	データ要求コマンド「V」をGT-N456へ100ms周期で送信開始します。
「ストップ」ボタン	データ要求コマンド「V」の送信を停止します。

※ 接続状態が接続中の場合のみコマンドが送信されます

② コマンド送信ボタン

コマンドの送信を行うボタンです。

「キャリブレーション」ボタン	キャリブレーション要求コマンド「cal」をGT-N456へ送信します。
「バージョンチェック」ボタン	バージョンチェック要求コマンド「ver」をGT-N456へ送信します。

※ 接続状態が接続中の場合のみコマンドが送信されます

③ COM 接続・切断ボタン

通信ポートの接続（オープン）・切断（クローズ）を行うボタンです。

「接続」ボタン	COM 通信をオープン（接続）しGT-N456との通信を開始します。
「切断」ボタン	COM 通信をクローズ（切断）しGT-N456との通信を終了します。

④ 送受信データ表示

GT-N456 との通信データを表示するテキストボックスです。

受信データは行頭に "<"、送信データは ">" を付加して表示します。

⑤ 最新受信データ表示

GT-N456 から送られてきた「加速度」と「スイッチ状態」の最新データを表示します。

⑥ 終了ボタン

「×」ボタンをクリックするとプログラムを終了します。

■ プログラムの内容

以下はサンプルプログラムのプロジェクト開発画面です。

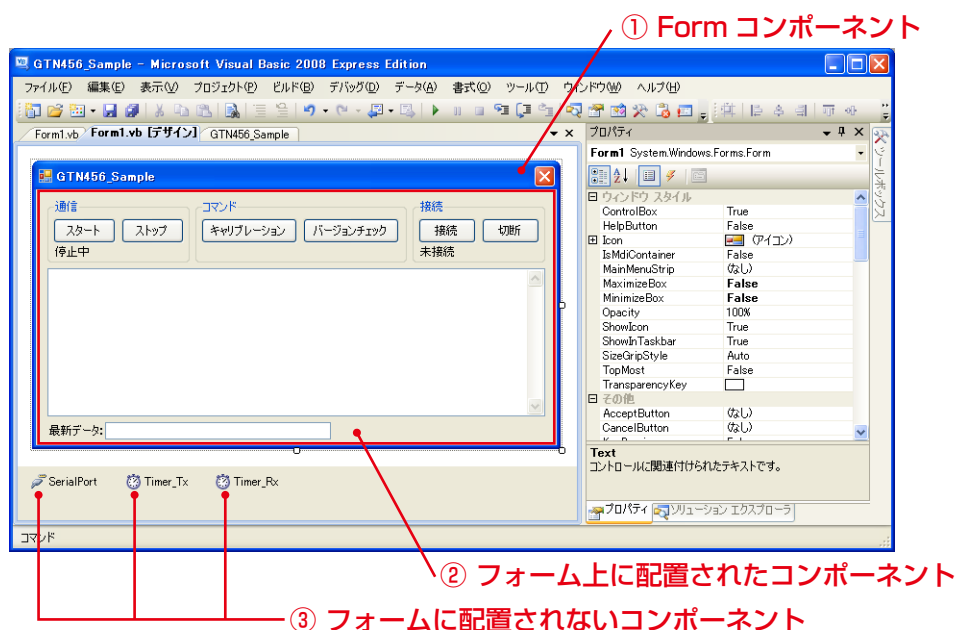


図2-4 プロジェクト開発画面 (Form1.vb[デザイン])

使用しているコンポーネントの設定概要は以下の通りです。

① Form コンポーネント

プロパティ	MaximizeBox	False	最大化ボタンを非表示
	MinimizeBoxText	False	最小化ボタンを非表示
	Text	GTN456_Sample	プログラム名の表示

② フォーム上に配置されたコンポーネント

・ ボタン : 操作用のボタンとして以下の6つを配置しています。

ボタン名	プロパティ	
	Name	Text
[スタート]ボタン	Button_Start	スタート
[ストップ]ボタン	Button_Stop	ストップ
[キャリブレーション]ボタン	Button_Cal	キャリブレーション
[バージョンチェック]ボタン	Button_VerChk	バージョンチェック
[接続]ボタン	Button_Connect	接続
[切断]ボタン	Button_DisConnect	切断

・ グループボックス : ボタンのグループ分け表示用として以下の3つを配置しています

グループボックス名	プロパティ	
	Name	Text
[通信]グループ	GroupBox1	通信
[コマンド]グループ	GroupBox2	コマンド
[接続]グループ	GroupBox3	接続

・ ラベル : フォーム上の説明として以下の3つを配置しています。

ラベル名	プロパティ	
	Name	Text
[通信状態]説明用ラベル	Label_ComState	停止中
[接続状態]説明用ラベル	Label_ConnectState	未接続
[最新データ]説明用ラベル	Label_RcvData	最新データ :

・ テキストボックス : 通信データの表示用として以下の2つを配置しています。

テキストボックス名	プロパティ	
	Name	Multiline
[ログ表示]テキストボックス	TextBox_Log	True
[最新受信データ表示]テキストボックス	TextBox_RcvData	False

Sunhayato

③ フォームに配置されないコンポーネント

以下はフォーム画面上にオブジェクトが表示されないコンポーネントです。ツールボックスから選択しフォーム上へドラッグ&ドロップすると開発画面の下部に機能のアイコンが配置されます。

- ・シリアルポート : GTN456 との通信用として1つを配置しています。

プロパティ	
シリアルコンポーネント名	Name
[シリアルポート]コンポーネント	SerialPort

- ・タイマー : 送信・受信処理をそれぞれ定期的に行うために2つを配置しています。

プロパティ			
タイマーコンポーネント名	Name	Enabled	Interval[msec]
[送信タイマ]コンポーネント	Timer_Tx	False	100
[受信タイマ]コンポーネント	Timer_Rx	True	50

■ プログラムコードの内容

Visual Basic では各コンポーネントに対応する処理を記述することによりプログラム全体の作成を行います。
以下のコードはプログラムコードを一覧表示したものです。

プログラムコード1：一覧表示

```
Public Class Form1
    '+++++
    ' 「接続」 ボタンクリック時の処理
    '+++++
    Private Sub Button_Connect_Click ...

    '+++++
    ' 「切断」 ボタンクリック時の処理
    '+++++
    Private Sub Button_DisConnect_Click ...

    '+++++
    ' 「スタート」 ボタンクリック時の処理
    '+++++
    Private Sub Button_Start_Click ...

    '+++++
    ' 「ストップ」 ボタンクリック時の処理
    '+++++
    Private Sub Button_Stop_Click ...

    '+++++
    ' 「キャリブレーション」 ボタンクリック時の処理
    '+++++
    Private Sub Button_Cal_Click ...

    '+++++
    ' 「バージョンチェック」 ボタンクリック時の処理
    '+++++
    Private Sub Button_VerChk_Click ...

    '+++++
    ' 送信タイマー(100ms周期)の処理
    '+++++
    Private Sub Timer_Tx_Tick ...

    '+++++
    ' 受信タイマー(50ms周期)の処理
    '+++++
    Private Sub Timer_Rx_Tick ...

    '-----
    ' 接続処理
    '-----
    Function SerialPort_Connect ...

    '-----
    ' 切断処理
    '-----
    Private Sub SerialPort_DisConnect ...

    '-----
    ' コマンド送信関数
    '-----
    Function Tx_Cmd ...
End Class
```

各コンポーネントに対応するイベントのプログラムコードを記述します。

特定の処理をまとめた関数です。上記の各処理で使用します。

以下は各処理のプログラムコードと処理内容の説明です。

● 「接続」 ボタンクリック時の処理

プログラムコード2:「接続」 ボタンクリック時の処理

```

+++++
'「接続」 ボタンクリック時の処理
+++++
Private Sub Button_Connect_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button_Connect.Click
    If SerialPort_Connect() = True Then
        Label_ConnectState.Text = "接続中(" & SerialPort.PortName & ")"
    End If
End Sub

```

「接続」 ボタンがクリックされた際は以下の処理を行います。

- ①. シリアルポートの接続を実施する
- ②. [接続成功]:「接続状態」ラベルに " 接続中 (COM 番号) " を表示する
[接続失敗]: 処理無し

● 「切断」 ボタンクリック時の処理

プログラムコード3:「接続」 ボタンクリック時の処理

```

+++++
'「切断」 ボタンクリック時の処理
+++++
Private Sub Button_DisConnect_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button_DisConnect.Click
    SerialPort_DisConnect()
    Label_ConnectState.Text = "未接続"

    '同時に通信動作を停止させる
    Timer_Tx.Stop() '送信タイマ動作停止
    Label_ComState.Text = "停止中"
End Sub

```

「切断」 ボタンがクリックされた際は以下の処理を行います。

- ①. シリアルポートの接続を切断する
- ②. 「接続状態」ラベルに " 未接続 " を表示する
- ③. 送信タイマ処理を停止する
- ④. 「通信状態」ラベルに " 停止中 " を表示する

● 「スタート」 ボタンクリック時の処理

プログラムコード4:「スタート」 ボタンクリック時の処理

```

+++++
'「スタート」 ボタンクリック時の処理
+++++
Private Sub Button_Start_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button_Start.Click
    If SerialPort.IsOpen = True Then
        '接続中
        Timer_Tx.Start() '送信タイマ動作開始
        Label_ComState.Text = "通信中"
    Else
        '未接続
        '処理無し
    End If
End Sub

```

「スタート」 ボタンがクリックされた際は以下の処理を行います。

- ①. シリアルポートの接続の状態を確認する
- ②. [接続中]: 送信タイマーを開始。「通信状態」ラベルに " 通信中 " を表示する
[未接続]: 処理無し

Sunhayato

● 「ストップ」 ボタンクリック時の処理

プログラムコード5:「ストップ」 ボタンクリック時の処理

```
+++++  
'「ストップ」 ボタンクリック時の処理  
+++++  
Private Sub Button_Stop_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button_Stop.Click  
    Timer_Tx.Stop() '送信タイマ動作停止  
    Label_ComState.Text = "停止中"  
End Sub
```

「ストップ」 ボタンがクリックされた際は以下の処理を行います。

- ①. 送信タイマーを停止する
- ②. 「通信状態」ラベルに " 停止中 " を表示する

● 「キャリブレーション」 ボタンクリック時の処理

プログラムコード6:「キャリブレーション」 ボタンクリック時の処理

```
+++++  
'「キャリブレーション」 ボタンクリック時の処理  
+++++  
Private Sub Button_Cal_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button_Cal.Click  
    If SerialPort.IsOpen = True Then  
        '接続中  
        Tx_Cmd("cal" & vbCrLf) 'キャリブレーションコマンドを送信  
    Else  
        '未接続  
        '処理無し  
    End If  
End Sub
```

「キャリブレーション」 ボタンがクリックされた際は以下の処理を行います。

- ①. シリアルポートの状態を確認する
- ②. [接続中] : コマンド送信処理を呼び出して "cal" コマンドを送信する
[切断中] : 処理無し

● 「バージョンチェック」 ボタンクリック時の処理

プログラムコード7:「バージョンチェック」 ボタンクリック時の処理

```
+++++  
'「バージョンチェック」 ボタンクリック時の処理  
+++++  
Private Sub Button_VerChk_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button_VerChk.Click  
    If SerialPort.IsOpen = True Then  
        '接続中  
        Tx_Cmd("ver" & vbCrLf) 'バージョンチェックコマンドを送信  
    Else  
        '未接続  
        '処理無し  
    End If  
End Sub
```

「バージョンチェック」 ボタンがクリックされた際は以下の処理を行います。

- ①. シリアルポートの状態を確認する
- ②. [接続中] : コマンド送信処理を呼び出して "ver" コマンドを送信する
[切断中] : 処理無し

● 「送信タイマー（100ms 周期）」の処理

プログラムコード8：「送信タイマー（100ms 周期）」の処理

```

'+++++
'送信タイマー(100ms周期)の処理
'+++++
Private Sub Timer_Tx_Tick(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Timer_Tx.Tick
    Tx_Cmd("V") 'Vコマンドを送信
End Sub

```

「送信タイマー」処理は「スタート」ボタンのクリックによりスタートが要求されている間、100ms 周期で処理を行います。処理内容はコマンド送信処理を呼び出して "V" コマンドの送信を行います。

● 「受信タイマー（50ms 周期）」の処理

プログラムコード9：「受信タイマー（50ms 周期）」の処理

```

'+++++
'受信タイマー(50ms周期)の処理
'+++++
Private Sub Timer_Rx_Tick(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Timer_Rx.Tick
    If SerialPort.IsOpen = True Then '接続中
        If SerialPort.BytesToRead < 0 Then '受信データ有り ①
            Dim rx_buf As String = ""
            Do
                '受信データの取得
                Try
                    rx_buf = SerialPort.ReadLine '受信データを1行取り出す
                Catch ex As Exception
                    '受信エラー時、エラーログを出力してDo-Loopを抜ける
                    TextBox_Log.AppendText(vbNewLine & "< 受信エラー" & vbNewLine)
                    Exit Do
                End Try

                TextBox_Log.AppendText("< " & rx_buf & vbNewLine) '受信した文字列をLog表示に追加

                If SerialPort.BytesToRead = 0 Then '全データを取り出し完了?
                    Exit Do 'Do-Loopを抜ける
                End If
            Loop

            If rx_buf.Contains("Time") = True Then '受信文字列に"Time"が含まれる場合は
                '受信文字列からX,Y,Z,SW1,SW2情報を抽出
                '()で囲まれた各データの抽出 ③
                Dim wk1() As String = rx_buf.Split(",")
                Dim wk2() As String
                Dim x, y, z, sw1, sw2 As String

                wk2 = wk1(1).Split(",")
                x = wk2(0)
                wk2 = wk1(2).Split(",")
                y = wk2(0)
                wk2 = wk1(3).Split(",")
                z = wk2(0)
                wk2 = wk1(4).Split(",")
                sw1 = wk2(0)
                wk2 = wk1(5).Split(",")
                sw2 = wk2(0)

                TextBox_RcvData.Text = "X=" & x & ",Y=" & y & ",Z=" & z & ",SW1=" & sw1 & ",SW2=" & sw2
            End If
        End If
    End Sub

```

「受信タイマー」処理は 50ms 周期で以下の処理を行っています。

- ①. シリアルポートの状態と受信データの有無を確認する
- ②. 受信データが存在する間、1行ずつ文字列の取得を行いログ表示用テキストボックスへ出力する
(改行コードを判定し1行ずつ取り出しを行う Visual Basic の関数を使用しています)
- ③. 最新の受信文字列に "Time" の文字列が含まれていることを確認する
("Time" の文字列が含まれている場合、XYZ 軸加速度とスイッチ情報が含まれています)
- ④. XYZ 軸加速度とスイッチ情報の抽出を行い、最新受信データ表示テキストボックスへ出力する

※ GT-N456 から送信される改行コードは「CR (キャリッジリターン)」 + 「LF (ラインフィード)」となっています

Sunhayato

● 接続処理

プログラムコード10：接続処理

```
'-----  
'接続処理  
'-----  
Function SerialPort_Connect() As Boolean  
    Dim Serial As Object  
    Dim SerialSet As Object  
    Dim ObjWMIService As Object  
    Dim StrComputer As String  
    Dim Com_Number As String = "Unknown"  
  
    SerialPort_Connect = True '返値初期値を「True（正常）」にセット  
  
    'WMIを使用してPnPデバイスの中からCOMポートデバイスの一覧を取得する  
    StrComputer = "."  
    ObjWMIService = GetObject("winmgmts:%%" & StrComputer & "%root%cimv2")  
    SerialSet = ObjWMIService.ExecQuery("Select * from Win32_PnpEntity Where " _  
        & "Name like '%(COM%)%'")  
  
    'GT-N456のCOM番号を探してオープンを行う  
    For Each Serial In SerialSet  
        If Serial.Caption.Contains("Sunhayato GT-N456 3-Axis Accelerometer") = True Then  
            Dim wk() As String = Serial.Caption.Split("(") 'COM番号の文字列を抽出する  
            Com_Number = wk(1).Split("%") 'COM番号を記憶  
            Exit For 'For-Nextを抜ける  
        End If  
    Next  
  
    If Com_Number = "Unknown" Then 'GT-N456が未接続の場合  
        MsgBox.Show("GT-N456が接続されていません", "エラー", MessageBoxButtons.OK, MessageBoxIcon.Error)  
        SerialPort_Connect = False '返値初期値を「False（異常）」にセット  
        Exit Function  
    End If  
  
    'COMの接続処理  
    If SerialPort.IsOpen = False Then '切断中の場合  
        SerialPort.PortName = Com_Number 'COM番号をセット  
        SerialPort.NewLine = vbCrLf 'GT-N456の仕様に合わせて、改行を\r\n(CRLF)で登録する  
        SerialPort.ReadTimeout = 500 'タイムアウト500ms  
        SerialPort.WriteTimeout = 500  
        SerialPort.Open() 'ポートオープン（接続）  
    End If  
End Function
```

「接続処理」はPCとGT-N456のCOM接続を行います。GT-N456のCOM番号をPnP(プラグアンドプレイ)デバイスの一覧から取得しポートオープン処理を実施します。

関数を使用する側で接続の成功/失敗がわかるように「True=成功」「False=失敗」を返します。

また、COM接続の設定で改行を「CR(キャリッジリターン)」+「LF(ラインフィード)」としてGT-N456の動作に合わせて設定しています。

● 切断処理

プログラムコード11：切断処理

```
'-----  
'切断処理  
'-----  
Private Sub SerialPort_DisConnect()  
    If SerialPort.IsOpen = True Then '接続中の場合のみ  
        SerialPort.Close() 'ポートクローズ（切断）  
    End If  
End Sub
```

- ①. シリアルポートの状態を確認する
- ②. [接続中]: シリアルポートの接続を切断する
[未接続]: 処理無し

● コマンド送信関数

プログラムコード12：コマンド送信関数

```
-----  
'コマンド送信関数  
-----  
Function Tx_Cmd(ByVal cmd As String) As Boolean  
    Tx_Cmd = True                                '返値に正常応答をセット  
    If SerialPort.IsOpen = True Then           'ポートがオープンされている場合のみ処理する  
        '送信処理  
        TextBox_Log.AppendText("> " & cmd & vbCrLf)    '送信する文字列をLog表示に追加  
        Try  
            SerialPort.Write(cmd)                'コマンドを送信  
        Catch ex As Exception                    '例外判定  
            TextBox_Log.AppendText(vbCrLf & "> 送信エラー" & vbCrLf)  
            Tx_Cmd = False                        '返値を異常で上書き  
        End Try  
    Else  
        'ポート非オープンの場合  
        Tx_Cmd = False  
    End If  
End Function
```

「コマンド送信関数」は任意の文字列を GT-N456 へ送信を行う処理です。

文字列を送信する際にログ表示用テキストボックスへの出力とシリアルポートの状態の確認を行います。また、関数を使用する側で送信の成功 / 失敗がわかるように「True= 成功」「False = 失敗」を返します。

本資料について

- 本資料は、電子工作や電子回路、パーソナルコンピュータの操作について一般的な知識をお持ちの方を対象としています。
- Microsoft®、Windows® は米国 Microsoft 社の米国およびその他の国における登録商標です。
- その他、記載されている製品名は各社の商標または登録商標です。

本資料のご利用にあたって

- 本資料に掲載している内容は、お客様が用途に応じた適切な製品をご購入頂くことを目的としています。その使用により当社及び第三者の知的財産権その他の権利に対する保証、又は実施権の許諾を意味するものではありません。また、権利の侵害に関して当社は責任を負いません。
- 本資料に記載した情報を流用する場合は、お客様のシステム全体で充分評価し適用可能かご判断願います。当社では適用可能判断についての責任を負いません。

- 本資料に記載してある内容は、一般的な電子機器（学習教材、事務機器、計測機器、パーソナル機器、コンピュータ機器など）に使用されることを目的としています。高い品質や信頼性が要求され、故障や誤作動が直接人命を脅かしたり人体に危害を及ぼす恐れのある、医療、軍事、航空宇宙、原子力制御、運輸、移動体、各種安全装置などの機器への使用は意図も保証もしていません。
- 本資料の一部、又は全部を当社の承諾なしで、いかなる形式でも転載又は複製されることは堅くお断りします。
- 全ての情報は本資料発行時点のものであり、当社は予告なしに本資料に記載した内容を変更することがあります。
- 本資料の内容は慎重に制作しておりますが、万一記述誤りによってお客様に損害が生じても当社はその責任を負いません。
- 本資料に関してのお問合せ、その他お気付きの点がございましたら、当社ホームページのお問い合わせページ (<https://www.sunhayato.co.jp/inquiry/>) よりお問合せください。
- 本資料に関する最新の情報はサンハヤト株式会社ホームページ (<http://www.sunhayato.co.jp/>) に掲載しております。

