

SendMessageTool

通信テスト用ツール

設定ファイル説明資料

2022/08/06 版

【概要】

通信テスト用ツールとして作成しました。

テスト時に簡易的な操作で予め設定したメッセージを送信するツールがあれば便利だと思い作成しました。

通信テストをされる方は、少し複雑な設定になっても対応いただけたらと思っています。

基本的な操作はマウスのみで簡易的な操作ですが、設定方法は XML 形式で少し複雑になっています。

通信ポートのタイプは、以下の種類が設定可能です。

設定ファイルを変更して EXE を起動してください。

本 EXE は、多重起動が可能であるため、設定を変更して複数の設定で EXE を起動して使用できます。

<< 通信ポートのタイプ >>

- (1) TCP-Client: TCP クライアントポート通信
- (2) TCP-Server: TCP サーバーポート通信
- (3) UDP-Unicast: UDP ユニキャストポート通信
- (4) UDP-Multicast: UDP マルチキャストポート通信
- (5) RS-232C: COM ポート通信

通信データ形式は、以下のフォーマットを設定可能

<< 通信データフォーマット >>

- (A) Text_CrLf: 設定した送信メッセージを ASCII 文字とし、最後に改行コード「CD」「LF」を追加して送信
- (B) Text_Only: 設定した送信メッセージを ASCII 文字として送信
- (C) Binary_Only: 設定した送信メッセージをバイナリー値 (HEX) として送信

本ツールのファイル構成は、以下の通りです。

<< ファイル構成 >>

SendMessageTool.exe :	本ツールの EXE ファイル
PortController.dll :	本ツールが使用する通信ポート制御用の専用ライブラリ
UnitBase.dll :	通信ポートを制御するための基本制御用の専用ライブラリ
UnitTcpClient.dll :	TCP クライアント制御用の専用ライブラリ
SystemSetting.xml :	本ツールの設定ファイル

本ツールは、マイクロソフト製「.NetFramework 4.6.1」以上を使用します。

予め「.NetFramework 4.6.1」以上をパソコンにインストールください。

<< 画面イメージ >>



<< 基本操作方法 >>

(1) 制御機器との接続

ソフトウェア起動後にボタン「接続」を押下して、指定した制御機器と接続します。

接続が完了すると、接続状態表示が「接続完了」に変わります。ボタンも「接続」から「切断」に変わります。

(2) メッセージ送信ボタンで制御機器へメッセージを送信

ボタン「メッセージ送信」を押下することで、予め設定したメッセージを制御機器に送信します。

ボタン「任意送信」は、任意送信メッセージで指定したメッセージを送信します。

任意送信メッセージの内容は、最後に送信したメッセージで更新されます。

メッセージ送信ボタンだけでなく、イベントやスケジュールの設定によりメッセージを送信した場合も、

任意送信メッセージの内容が更新されるので、ご注意ください。

(3) 制御機器と切断

ボタン「切断」を押下して、制御機器との接続を切断します。

<< 設定ファイルの説明 >>

A. ツールのタイトルを設定

`<ToolTitle>Test Device Text Client</ToolTitle>`

画面上部のタイトルとして表示

B. 通信制御用DLLを設定

`<UnitDllFileName>UnitTcpClient.dll</UnitDllFileName>`

指定可能な制御用DLLは、以下の通りです

- | | |
|----------------|------------------------|
| (1) TCPクライアント | : UnitTcpClient.dll |
| (2) TCPサーバー | : UnitTcpServer.dll |
| (3) UDPユニキャスト | : UnitUdpUnicast.dll |
| (4) UDPマルチキャスト | : UnitUdpMulticast.dll |
| (5) RS-232C | : UnitRs232C.dll |

※ 今後、設定可能なDLLを追加する予定

C. 送信メッセージのタイプを設定

`<MessageType>Text_CrLf</MessageType>` `<!-- Text_CrLf / Text_Only / Binary_Only -->`

指定可能な送信メッセージのタイプは、以下の通り

- | | |
|--|---------------|
| (1) テキストで終端コードCR+LF | : Text_CrLf |
| 設定した送信メッセージ（以下参照）の内容をASCIIコードで送信し、
メッセージの終端としてCR+LFを追加する。 | |
| (2) テキストで終端コードなし | : Text_Only |
| 設定した送信メッセージ（以下参照）の内容をASCIIコードで送信する。終端は付加しない。 | |
| (3) バイナリーで終端コードなし | : Binary_Only |
| 設定した送信メッセージ（以下参照）の内容をバイナリーコードで送信する。終端は付加しない。 | |

D. IPアドレスの初期値を設定 （通信ポートのタイプにより設定項目が異なる）

`<TargetIpAddr>127.0.0.1</TargetIpAddr>`

ツール起動時に表示するIPアドレスを設定する。

本IPアドレスは、接続前であれば変更可能

E. ポート番号の初期値を設定 （通信ポートのタイプにより設定項目が異なる）

`<TargetPortNo>60001</TargetPortNo>`

ツール起動時に表示するポート番号を設定する。

本ポート番号は、接続前であれば変更可能

F. メッセージ送信ボタンエリアのサイズを設定

```
<CtrlButtonAreaSize>
    <RowSize>4</RowSize>
    <ColumnSize>8</ColumnSize>
</CtrlButtonAreaSize>
```

メッセージ送信ボタンエリアに設定できる送信ボタンの行数と列数を設定する。

上記の通り 4 行・8 列で設定した場合、送信ボタンは、以下の通り最大32個設定可能となる。



G. 送信メッセージボタンを設定

```
<CtrlButtonList>
    <CtrlButtonData Index="0">
        <ButtonTitle>ABC...</ButtonTitle>
        <SendMessageList>
            <SendMessage>ABCDEFGHIJKLMNOPQRSTUVWXYZ</SendMessage>
        </SendMessageList>
    </CtrlButtonData>
```

Index="0"で、表示する位置を設定する。

Index="0"が左上の位置とし、その右の位置がIndex="1"として設定する。

タグ「CtrlButtonList」内にタグ「CtrlButtonData」を複数記載し、
「Index」で指定した位置に送信メッセージボタンが表示される。

H. 送信メッセージボタンのタイトルを設定

```
<CtrlButtonList>
  <CtrlButtonData Index="0">
    <ButtonTitle> ABC...</ButtonTitle>
    <SendMessageList>
      <SendMessage> ABCDEFGHIJKLMNOPQRSTUVWXYZ </SendMessage>
    </SendMessageList>
  </CtrlButtonData>
```

画面に表示するボタンの名称を設定する。

I-1. 送信メッセージを設定 （例：テキスト）

```
<CtrlButtonList>
  <CtrlButtonData Index="0">
    <ButtonTitle> ABC...</ButtonTitle>
    <SendMessageList>
      <SendMessage> ABCDEFGHIJKLMNOPQRSTUVWXYZ </SendMessage>
    </SendMessageList>
  </CtrlButtonData>
```

ボタン押下時に送信するメッセージを設定する。

I-2. 送信メッセージを設定 （例：バイナリー）

```
<CtrlButtonList>
  <CtrlButtonData Index="0">
    <ButtonTitle> ABC...</ButtonTitle>
    <SendMessageList>
      <SendMessage> 4142434445464748494A4B4C4D4E4F505152535455565758595A</SendMessage>
    </SendMessageList>
  </CtrlButtonData>
```

送信メッセージのタイプをバイナリーで設定した場合は、2文字単位の16進数のバイナリー値を設定する。
上記の場合は、以下のバイナリーメッセージが送信される。

41-42-43-44-45-46-47-...-59-5A: 26バイトのバイナリー値を送信する

I-3. 送信メッセージを設定 （例：複数メッセージの同時送信）

```
<CtrlButtonList>
  <CtrlButtonData Index="4">
    <ButtonTitle>Test</ButtonTitle>
    <SendMessageList>
      <SendMessage>Test Message</SendMessage>
      <SendMessage>0123456789</SendMessage>
    </SendMessageList>
  </CtrlButtonData>
```

上記の通りタグ「SendMessageList」内に複数のタグ「SendMessage」を設定することで、ボタン押下時に複数のメッセージを送信することができる。

上記の設定では、ボタン「Test」押下時に、2パケットを連続送信する。

J. イベント処理を設定

イベントの種類は、以下の通り

== イベント種別 ==

- | | |
|----------------|-------------|
| (1) 接続完了イベント： | OnConnected |
| (2) ボタン操作イベント： | OnClicked |
| (3) タイマーイベント： | OnTimer |
| (4) 受信イベント： | OnReceived |

各イベント時に以下の処理を実施できる

== 処理項目 ==

- (A) ボタン操作時のメッセージ送信
- (B) タイマーの開始
- (C) 送信メッセージ

設定方法は、以下の通り

設定例①： 接続完了時にメッセージ「Connected」を自動送信する

```
<EventList>
  <CtrlEvent Type="OnConnected">
    <SendMessageList>
      <SendMessage>Connected</SendMessage>
    </SendMessageList>
  </CtrlEvent>
```

```
</EventList>
```

設定例②： 接続完了時にタイマー 0 を 1 秒に設定する

タイマー 0 がタイムアップ時にボタン 12 操作時のメッセージを自動送信する

```
<EventList>
```

```
<CtrlEvent Type="OnConnected">
```

```
<CtrlTimerNo>0</CtrlTimerNo>
```

```
<SetTimerValue>1000</SetTimerValue>
```

```
</CtrlEvent>
```

```
<CtrlEvent Type="OnTimer" ID="0">
```

```
<CtrlButtonNo>12</CtrlButtonNo>
```

```
</CtrlEvent>
```

```
</EventList>
```

設定例③： ボタン6操作時にメッセージを送信後、タイマー 1 を 10 秒に設定する

タイマー 1 がタイムアップ時にボタン6操作時のメッセージを自動送信し、

再度タイマー 1 を 10 秒に設定する

(ボタン6操作後、切断するまで10秒間隔でメッセージを自動送信する)

```
<EventList>
```

```
<CtrlEvent Type="OnClicked" ID="6">
```

```
<CtrlTimerNo>1</CtrlTimerNo>
```

```
<SetTimerValue>10000</SetTimerValue>
```

```
</CtrlEvent>
```

```
<CtrlEvent Type="OnTimer" ID="1">
```

```
<CtrlButtonNo>6</CtrlButtonNo>
```

```
<CtrlTimerNo>1</CtrlTimerNo>
```

```
<SetTimerValue>10000</SetTimerValue>
```

```
</CtrlEvent>
```

```
</EventList>
```

設定例④： 「Active」から始まるメッセージを受信した時、メッセージ「Active」を返送する

```
<EventList>
```

```
<CtrlEvent Type="OnReceived">
```

```
<ReceivedMessage>Active</ReceivedMessage>
```

```
<SendMessageList>
```

```
<SendMessage>Active</SendMessage>
```

```
</SendMessageList>
```

```
</CtrlEvent>
```

```
</EventList>
```


上記①～③の全項目を設定

```
<EventList>
  <CtrlEvent Type="OnConnected">
    <CtrlTimerNo>0</CtrlTimerNo>
    <SetTimerValue>1000</SetTimerValue>
    <SendMessageList>
      <SendMessage>Connected</SendMessage>
    </SendMessageList>
  </CtrlEvent>
  <CtrlEvent Type="OnClicked" ID="6">
    <CtrlTimerNo>1</CtrlTimerNo>
    <SetTimerValue>10000</SetTimerValue>
  </CtrlEvent>
  <CtrlEvent Type="OnTimer" ID="0">
    <CtrlButtonNo>12</CtrlButtonNo>
  </CtrlEvent>
  <CtrlEvent Type="OnTimer" ID="1">
    <CtrlButtonNo>6</CtrlButtonNo>
    <CtrlTimerNo>1</CtrlTimerNo>
    <SetTimerValue>10000</SetTimerValue>
  </CtrlEvent>
</EventList>
```

K. スケジュール送信を設定

スケジュールの送信設定は、スケジュールパターンに以下の項目を設定する

== スケジュールパターン設定 ==

- ・タイトル Title
- ・繰り返し設定 IsAutoRepeat
- ・スケジュールイベント EventList

```
<ScheduleEventList>
  <ScheduleEvent>
    <Title>TestList1</Title>
    <IsAutoRepeat>true</IsAutoRepeat>
    <EventList>
      (中省略)
    </EventList>
```

</ScheduleEvent>

<ScheduleEvent>

<Title>TestList2</Title>

<IsAutoRepeat>false</IsAutoRepeat>

<EventList>

(中省略)

</EventList>

</ScheduleEvent>

</ScheduleEventList>

スケジュールイベントには、以下の項目を設定する

== スケジュールイベント設定 ==

- ・送信待ち時間 [秒]
- ・送信メッセージ

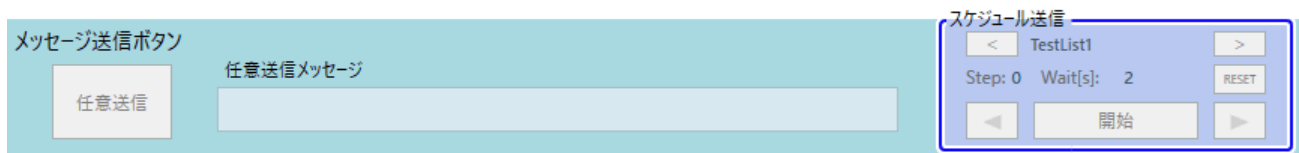
スケジュール設定すると、以下のパネル「スケジュール送信」が表示されます。

スケジュールのタイトル（例：TestList1）が上部に表示され、ステップ「Step」と待ち時間「Wait」が表示されます。

接続完了後にボタン「開始」が選択可能になります。

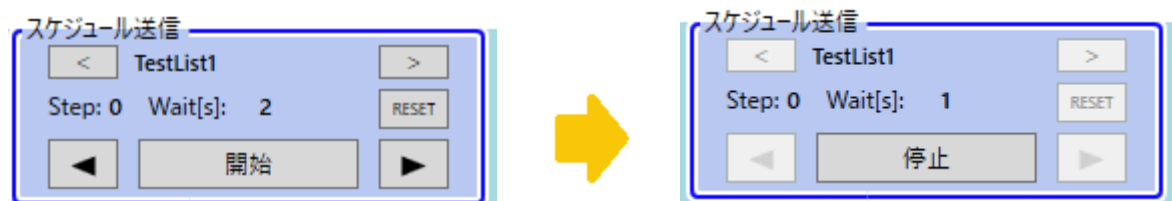
ボタン「開始」を押下すると、「Wait」表示がカウントダウンし、

タイムアップ時に現ステップの送信メッセージを送信します。



スケジュール動作中は、ボタン「開始」が「停止」に変わり、パネルの他のボタンが操作不可能となります。

スケジュール停止中は、【パターン変更】や【ステップ変更】が可能です



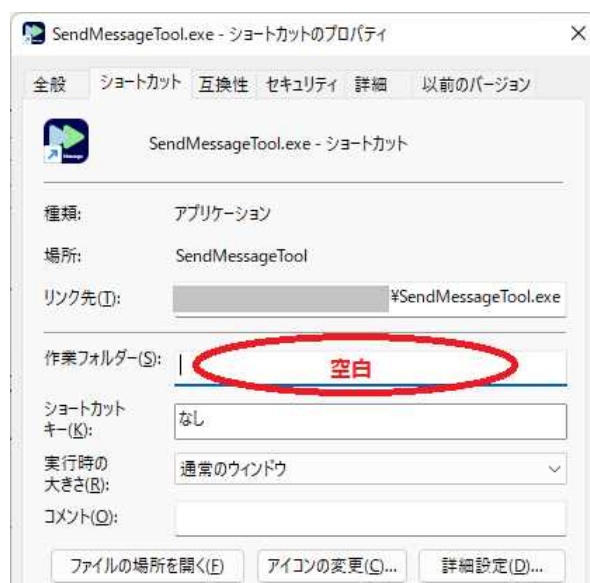
異なる設定でツールを起動する方法

設定ファイルは、以下の順番に読み込みます。

- (1) 作業フォルダー
- (2) EXEフォルダー

EXEのショートカットを作成し、「作業フォルダー」を空白にすることで、ショートカットフォルダーが保存されたフォルダーの設定ファイルを使用します。

1つのEXEファイルで、異なった設定ファイルでツールを起動する場合は、設定ファイルを任意のフォルダーに保存し、ショートカットを設定ファイルのフォルダーに作成し、ショートカットにて本ツールを起動ください。



更新履歴

2022.08.06

制御ユニットを DLL 化に変更し、説明内容を修正

2022.07.18

説明用ドキュメントを新規作成

* SendMessageTool Ver.3.0.0

* 通信テスト用ツール

*

* H.Matsumoto

* machao735.iphone@gmail.com

* 2022/08/06

*
