

Namespace ESL_Protocol_V1

Classes

[EslProtocolGen](#)

[EslProtocolGen.Template](#)

電子紙模板產生模組

[EslProtocolGen.Update](#)

電子紙元素資料內容封包產生

Enums

[EslColorBRW](#)

[EslContentAlignment](#)

[EslFont](#)

[EslLed](#)

Enum EslColorBRW

Namespace: [ESL Protocol V1](#)

Assembly: ESL_Protocol_V1.dll

```
public enum EslColorBRW : byte
```

Fields

```
Black = 0
```

```
Red = 1
```

```
White = 2
```

Enum EslContentAlignment

Namespace: [ESL Protocol V1](#)

Assembly: ESL_Protocol_V1.dll

```
public enum EslContentAlignment : byte
```

Fields

```
LeftBott = 34
```

```
LeftMid = 33
```

```
LeftUp = 32
```

```
MidBott = 18
```

```
MidMid = 17
```

```
MidUp = 16
```

```
RightBott = 2
```

```
RightMid = 1
```

```
RightUp = 0
```

Enum EslFont

Namespace: [ESL Protocol V1](#)

Assembly: ESL_Protocol_V1.dll

```
public enum EslFont : byte
```

Fields

Font12 = 0

Font16 = 1

Font24 = 2

Font32 = 3

Enum EslLed

Namespace: [ESL Protocol V1](#)

Assembly: ESL_Protocol_V1.dll

```
public enum EslLed : byte
```

Fields

```
LED_B = 4
```

```
LED_G = 3
```

```
LED_OFF = 0
```

```
LED_R = 1
```

```
LED_Y = 2
```

Class EslProtocolGen

Namespace: [ESL Protocol V1](#)








Assembly: ESL_Protocol_V1.dll

```
public class EslProtocolGen
```

Inheritance

[object](#)  ← EslProtocolGen

Inherited Members

[object.Equals\(object\)](#)  , [object.Equals\(object, object\)](#)  , [object.GetHashCode\(\)](#)  ,
[object.GetType\(\)](#)  , [object.MemberwiseClone\(\)](#)  , [object.ReferenceEquals\(object, object\)](#)  ,
[object.ToString\(\)](#) 

Methods

[EslRefresh\(\)](#)

標籤螢幕重刷

Method EslRefresh

Namespace: [ESL Protocol V1](#)

Assembly: ESL_Protocol_V1.dll

EslRefresh()

標籤螢幕重刷

```
public IEnumerable<byte> EslRefresh()
```

Returns

[IEnumerable](#) <[byte](#)>

Class EslProtocolGen.Template

Namespace: [ESL Protocol V1](#)

Assembly: ESL_Protocol_V1.dll








電子紙模板產生模組

```
public class EslProtocolGen.Template
```

Inheritance

[object](#)  ← EslProtocolGen.Template

Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) ,
[object.ToString\(\)](#) 

Examples

```
EslProtocolGen.Template EslTemp = new EslProtocolGen.Template();  
EslTemp.Clear();  
EslTemp.Add_Bitmap(0, 0, 50, 6, EslProtocolGen.alignMode.RightUp,  
EslProtocolGen.colorMode.Black); //元素1:BMP  
EslTemp.Add_Bitmap(0, 120, 200, 4, EslProtocolGen.alignMode.RightUp,  
EslProtocolGen.colorMode.Black); //元素2:BMP  
EslTemp.Add_String(52, 22, EslProtocolGen.fontType.Font16, EslProtocolGen.alignMode.RightUp,  
EslProtocolGen.colorMode.Black); //元素3:String  
EslTemp.Add_Num(70, 80, EslProtocolGen.fontType.Font24, EslProtocolGen.alignMode.MidUp,  
EslProtocolGen.colorMode.Black); //元素4:Num  
EslTemp.GenPkg();
```

Properties

[AutoNewTemplate](#)

自動覆蓋設備上原有的模板資料，否則往後添加模板

Methods

[Add_Bitmap\(ushort, ushort, ushort, byte, EslContentAlignment, EslColorBRW\)](#)

添加一個Bitmap元素

[Add_Code39\(ushort, ushort, byte, EslContentAlignment, EslColorBRW\).](#)

添加一個code39元素

[Add_Num\(ushort, ushort, EslFont, EslContentAlignment, EslColorBRW\).](#)

添加一個數字元素

[Add_String\(ushort, ushort, EslFont, EslContentAlignment, EslColorBRW\).](#)

添加一個String元素

[Clear\(\).](#)

清除模板內容及元素

[GenPkg\(\).](#)

模板封包產生(HEX)

Property AutoNewTemplate

Namespace: [ESL Protocol V1](#)

Assembly: ESL_Protocol_V1.dll

AutoNewTemplate

自動覆蓋設備上原有的模板資料，否則往後添加模板

```
public bool AutoNewTemplate { get; set; }
```

Property Value

[bool](#)

Method Add_Bitmap

Namespace: [ESL Protocol V1](#)

Assembly: ESL_Protocol_V1.dll

Add_Bitmap(ushort, ushort, ushort, byte, EslContentAlignment, EslColorBRW)

添加一個Bitmap元素

```
public void Add_Bitmap(ushort posX, ushort posy, ushort heightPixel, byte widthByte, EslContentAlignment align, EslColorBRW color)
```

Parameters

posx [ushort](#)

X座標參考

posy [ushort](#)

Y座標參考

heightPixel [ushort](#)

Bitmap高度

widthByte [byte](#)

Bitmap寬度(單位Byte)

align [EslContentAlignment](#)

對齊方式

color [EslColorBRW](#)

顏色

Method Add_Code39

Namespace: [ESL Protocol V1](#)

Assembly: ESL_Protocol_V1.dll

Add_Code39(ushort, ushort, byte, EslContentAlignment, EslColorBRW)

添加一個code39元素

```
public void Add_Code39(ushort posx, ushort posy, byte heightPixel, EslContentAlignment align, EslColorBRW color)
```

Parameters

posx [ushort](#)

X座標參考

posy [ushort](#)

Y座標參考

heightPixel [byte](#)

條碼高度(pixel)

align [EslContentAlignment](#)

對齊方式

color [EslColorBRW](#)

顏色

Method Add_Num

Namespace: [ESL Protocol V1](#)

Assembly: ESL_Protocol_V1.dll

Add_Num(ushort, ushort, EslFont, EslContentAlignment, EslColorBRW)

添加一個數字元素

```
public void Add_Num(ushort posx, ushort posy, EslFont font, EslContentAlignment align, EslColorBRW color)
```

Parameters

posx [ushort](#)

X座標參考

posy [ushort](#)

Y座標參考

font [EslFont](#)

字體類型(fontType)

align [EslContentAlignment](#)

對齊方式

color [EslColorBRW](#)

顏色

Method Add_String

Namespace: [ESL Protocol V1](#)

Assembly: ESL_Protocol_V1.dll

Add_String(ushort, ushort, EslFont, EslContentAlignment, EslColorBRW)

添加一個String元素

```
public void Add_String(ushort posX, ushort posY, EslFont font, EslContentAlignment align, EslColorBRW color)
```

Parameters

posx [ushort](#)

X座標參考

posy [ushort](#)

Y座標參考

font [EslFont](#)

字體類型(fontType)

align [EslContentAlignment](#)

對齊方式

color [EslColorBRW](#)

顏色

Method Clear

Namespace: [ESL Protocol V1](#)

Assembly: ESL_Protocol_V1.dll

Clear()

清除模板内容及元素

```
public void Clear()
```

Method GenPkg

Namespace: [ESL Protocol V1](#)

Assembly: ESL_Protocol_V1.dll

GenPkg()

模板封包產生(HEX)

```
public IEnumerable<byte> GenPkg()
```

Returns

[IEnumerable](#) <[byte](#)>

回傳為zigbee payload封包資料(不包含標籤短地址的部分)

Class EslProtocolGen.Update

Namespace: [ESL Protocol V1](#)

Assembly: ESL_Protocol_V1.dll








電子紙元素資料內容封包產生

```
public class EslProtocolGen.Update
```

Inheritance

[object](#)  ← EslProtocolGen.Update

Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) ,
[object.ToString\(\)](#) 

Examples

```
//假設模板為[(0)String , (1)Bitmap , (2)String , (3)String , (4)Code39]  
//想要產生更新(1),(3),(4)元素內容的封包  
EslProtocolGen.Update EslUpd = new EslProtocolGen.Update();  
EslUpd.Clear();  
EslUpd.Bitmap(1, ConvStr.ConvertHexStringToList(elem.Content)); //更新元素1:BMP  
EslUpd.String(3, "TEST"); //元素3:String  
EslUpd.Code39(4, "0123456"); //元素4:Code39  
var Datas = EslUpd.GenPkg();
```

Properties

[BmpSplitSize](#)

Bitmap資料分包大小・預設120 byte

[ContentSplitSize](#)

其餘所有內容資料分包大小・預設120 byte

Methods

[Bitmap\(byte, IEnumerable<byte>\).](#)

更新一個Bitmap元素

[Clear\(\).](#)

清除上傳指令內容

[Code39\(byte, string\).](#)

更新一個Code39元素

[GenPkg\(\).](#)

素材上傳指令封包產生(HEX)

[Indicator\(EslLed\).](#)

更新LED狀態

[LASA\(byte\).](#)

更新LASA狀態

[Num\(byte, uint\).](#)

更新一個Num元素

[String\(byte, string\).](#)

更新一個String元素

Property BmpSplitSize

Namespace: [ESL Protocol V1](#)

Assembly: ESL_Protocol_V1.dll

BmpSplitSize

Bitmap資料分包大小・預設120 byte

```
public byte BmpSplitSize { get; set; }
```

Property Value

[byte](#) 

Property ContentSplitSize

Namespace: [ESL Protocol V1](#)

Assembly: ESL_Protocol_V1.dll

ContentSplitSize

其餘所有內容資料分包大小，預設120 byte

```
public byte ContentSplitSize { get; set; }
```

Property Value

[byte](#)

Method Bitmap

Namespace: [ESL Protocol V1](#)

Assembly: ESL_Protocol_V1.dll

Bitmap(byte, IEnumerable<byte>)

更新一個Bitmap元素

```
public bool Bitmap(byte ElemNum, IEnumerable<byte> Bmpdata)
```

Parameters

ElemNum [byte](#)

目標元素序號(模板建立的順序)

Bmpdata [IEnumerable](#) <[byte](#)>

新的Bitmap元素資料

Returns

[bool](#)

Method Clear

Namespace: [ESL Protocol V1](#)

Assembly: ESL_Protocol_V1.dll

Clear()

清除上傳指令內容

```
public void Clear()
```

Method Code39

Namespace: [ESL Protocol V1](#)

Assembly: ESL_Protocol_V1.dll

Code39(byte, string)

更新一個Code39元素

```
public bool Code39(byte ElemNum, string Str)
```

Parameters

ElemNum [byte](#) 

目標元素序號(模板建立的順序)

Str [string](#) 

新的Code39元素資料

Returns

[bool](#) 

Method GenPkg

Namespace: [ESL Protocol V1](#)

Assembly: ESL_Protocol_V1.dll

GenPkg()

素材上傳指令封包產生(HEX)

```
public IEnumerable<IEnumerable<byte>> GenPkg()
```

Returns

[IEnumerable](#) < [IEnumerable](#) < [byte](#) > >

回傳為zigbee payload封包資料(不包含標籤短地址的部分)

Method Indicator

Namespace: [ESL Protocol V1](#)

Assembly: ESL_Protocol_V1.dll

Indicator(EslLed)

更新LED狀態

```
public void Indicator(EslLed Color)
```

Parameters

Color [EslLed](#)

Method LASA

Namespace: [ESL Protocol V1](#)

Assembly: ESL_Protocol_V1.dll

LASA(byte)

更新LASA狀態

```
public void LASA(byte lasa)
```

Parameters

lasa [byte](#)

Method Num

Namespace: [ESL Protocol V1](#)

Assembly: ESL_Protocol_V1.dll

Num(byte, uint)

更新一個Num元素

```
public bool Num(byte ElemNum, uint Num)
```

Parameters

ElemNum [byte](#)

目標元素序號(模板建立的順序)

Num [uint](#)

新的Num元素資料

Returns

[bool](#)

Method String

Namespace: [ESL Protocol V1](#)

Assembly: ESL_Protocol_V1.dll

String(byte, string)

更新一個String元素

```
public bool String(byte ElemNum, string Str)
```

Parameters

ElemNum [byte](#) 

目標元素序號(模板建立的順序)

Str [string](#) 

新的String元素資料

NOTE

資料不可為null

Returns

[bool](#) 