

C#-MODBUS 接口说明

```
/*
    根据传入的参数，创建 MODBUS RTU 实例
    _____*/
    /// <summary>
    ///     建立 MODBUS RTU 串口业务
    ///     String commum =  串口号
    ///     int buad      =  串口的波特率
    ///     int dataLen   =  数据长度
    ///     String chklen =  检验位
    ///     String stoplen =  停止位
    /// </summary>
public ModbusRTU_Lib(String commum, int buad, int dataLen, string chklen, string stoplen, int
scantime)

/*
    清除创建的 MODBUS RTU 多从站扫描列表
    _____*/
    /// <summary>
    ///     清除所有的从站列表
    /// </summary>
public void clearAllModbusSalve()

    /// <summary>
    ///     删除扫描列表中的指定从站
    /// </summary>
    /// <param name="whichAddr"></param>
public void deletAllModbusSalve(int whichAddr)

/*
    添加一个 MODBUS RTU 扫描从站
    _____*/
    /// <summary>
    ///     增加 MODBUS 从站扫描列表
    /// </summary>
    /// <param name="xDeviceMdEnty"></param>
public void addModbusSlaveScanList(DeviceMdEnty xDeviceMdEnty)
```

```
/*
    打开一个 MODBUS RTU 的通讯串口
    _____ */
/// <summary>
    /// 打开串口
    /// </summary>
    /// <returns></returns>
public Boolean openUartOfModbus()
/*
关闭串口
    _____ */
/// <summary>
    /// 关闭串口
    /// </summary>
    /// <returns></returns>
public Boolean closeUartOfModbus()

/*
启动 MODBUS RTU 的定时扫描业务
    _____ */
/// <summary>
    /// 启动 MODBUS 扫描业务
    /// </summary>
public void startModbusWork()

/*
发送 MODBUS RTU 指令
    _____ */
/*
    发送数据
    *
    int slaveAddr = 从站地址
    int addrTe = 读写的启始地址
    int cmdTe = 功能码
    int lenTe = 读出数据长度
    byte[] msgBuf = 待写入的数据的缓冲期, 每个字占 2BYTE
    int msgLen = 写入的数据长度
    return:
    * true = 已发送数据, false = 未发送
    *
    *****/
/// <summary>
    ///
    /// </summary>
```

```
public Boolean sendModbusMsg(int slaveAddr, int cmdTe, int addrTe, int lenTe, byte[]  
msgBuf, int msgLen)
```

```
/*
```

```
MODBUS RTU 的实例的回调方法，返回所有串口收到的数据
```

```
Result = "E00"; 表示"创建串口失败:"
```

```
Result = "E01"; 表示"串口号未设置:"
```

```
Result = "E02"; 表示"停止位参数错误:"
```

```
Result = "E03"; 表示"校验位参数错误:"
```

```
Result = "F00"; 表示"校验位参数错误:"
```

```
Result = "Y00"; 表示"创建串口成功:"
```

```
Result = "M00"; 表示"接收到串口数据:"
```

```
*/
```

```
//声明关于事件的委托, 参数为要回传的数据类型, 串口接收数据
```

```
public delegate void ModbusRTU_EventHandler(string result, byte[] msg);
```

```
/*
```

```
MODBUS RTU 的实例的回调方法，返回所有串口收到的数据且经过地址和校验码验证的
```

```
Result = "AE0"; 表示"创建串口失败:"
```

```
Result = "CE0"; 表示"串口号未设置:"
```

```
Result = "DE0"; 表示"停止位参数错误:"
```

```
Result = "D00"; 表示"接收到正确解析的串口数据:"
```

```
Result = "D01"; 表示"功能码 0x01--正确应答数据!:"
```

```
Result = "D02"; 表示"功能码 0x02--正确应答数据! "
```

```
Result = "D03"; 表示"功能码 0x03--正确应答数据! "
```

```
Result = "D04"; 表示"功能码 0x04--正确应答数据! "
```

```
Result = "D05"; 表示"功能码 0x05--正确应答数据! "
```

```
Result = "D06"; 表示"功能码 0x06--正确应答数据! "
```

```
Result = "D0F"; 表示"功能码 0x0F--正确应答数据! "
```

```
Result = "D10"; 表示"功能码 0x10--正确应答数据! "
```

```
*/
```

```
//声明关于事件的委托, 参数为要回传的数据类型, 串口接收数据并 MODBUS 验证
```

```
public delegate void ModbusRTU_DecodeHandler(string result, byte[] msg);
```

```
/*
```

```
建立 1 个 MODBUS RTU 从站的列表;
```

```
DLL 会根据设备数自动扫描; 对于没有的功能, 指定长度为 0 即可; 如 16 入 16 出数字量模块则可指  
定输入寄存器和保持寄存器长度为 0
```

```
*/
```

```
/// <summary>
```

```
/// 建立扫描列表
```

```
/// </summary>
```

```
/// <param name="SlaveAddr">设置当前从站的站号</param>
```

东莞市艾莫迅自动化科技有限公司

```
    /// <param name="CoilLen">设置当前从站的输出线圈个数</param>  
    /// <param name="DiscLen">设置当前从站的输入数字量个数</param>  
    /// <param name="InputRegLen">设置当前从站的输入寄存器个数</param>  
    /// <param name="HoldingRegLen">设置当前从站的保持寄存器个数</param>  
public DeviceMdEnty(int SlaveAddr, int CoilLen, int DiscLen, int InputRegLen, int  
HoldingRegLen)
```