

# S7Helper

[Jump to bottom](#)

Federico Barresi edited this page on Jan 12, 2019 · 1 revision

This class allow to read/write an "S7" value into a byte buffer, all new S71200/1500 types are supported.

- Read functions are in format:

```
public static <.NET primitive Type> Get<S7 Type>At(byte[] Buffer, int Pos)
```

They extract a var from the byte buffer starting from the byte number "pos".

- Write functions are in format:

```
public static void Set<S7 Type>At(byte[] Buffer, int Pos, <.NET primitive Type> Value)
```

They insert a .net primitive type var into the byte buffer starting from the byte number "pos".

## Data access example

Now, just a real example of how to read a S7 struct from a PLC into .NET struct using this class in C# and VB. Let's suppose that we make a leak test on a component and that we want to acquire the result of this test. The data of this test consists of some values (Component serial number, Leak Value etc..) stored in DB 100.

Every time a component is tested this struct is filled by the PLC and we want to read it into a .NET struct, picking the fields and adjusting their byte-order (the PLC is Big-Endian, the PC is Little-Endian, so the bytes must be reversed).

This is our C# sample struct:

```
public struct ComponentResult
{
    public String SerialNumber; // Component Serial Number
    public int TestResult;      // Result code 0:Unknown, 1:Good, 2:Scrap
    public double LeakDetected; // Leak value [cc/min]
```

```

    public DateTime TestDateTime; // Test Timestamp
}

```

And this is the C# sample function that fills the struct. Notice that the outcome of any Client function call should be checked, here is skipped for brevity.

```

private ComponentResult LeakResult()
{
    ComponentResult Result = new ComponentResult();
    byte[] Buffer = new byte[26];
    // Reads the buffer.
    Client.DBRead(100, 0, 26, Buffer);
    // Extracts the fields and inserts them into the struct
    Result.SerialNumber = S7.GetCharsAt(Buffer, 0, 12);
    Result.TestResult = S7.GetIntAt(Buffer, 12);
    Result.LeakDetected = S7.GetRealAt(Buffer, 14);
    Result.TestDateTime = S7.GetDateTimeAt(Buffer, 18);
    return Result;
}

```

Same as above in VB.NET

The struct:

```

Private Structure ComponentResult
    Public SerialNumber As String ' Component Serial Number
    Public TestResult As Integer ' Result code 0:Unknown, 1:Good, 2:Scrap
    Public LeakDetected As Double ' Leak value [cc/min]
    Public TestDateTime As DateTime ' Test Timestamp
End Structure

```

...and the function

```

Private Function LeakResult() As ComponentResult
    Dim Result As ComponentResult
    Dim Buffer(26) As Byte
    Client.DBRead(100, 0, 26, Buffer)
    Result.SerialNumber = S7.GetCharsAt(Buffer, 0, 12)
    Result.TestResult = S7.GetIntAt(Buffer, 12)
    Result.LeakDetected = S7.GetRealAt(Buffer, 14)
    Result.TestDateTime = S7.GetDateTimeAt(Buffer, 18)
    Return Result
End Function

```



## S7Client

[Get connected](#)

[I/O functions](#)

[System Info functions](#)

[PLC control functions](#)

[Security functions](#)

[Properties and info](#)

[ErrorCodes](#)

## S7Helper

## S7MultiVar

### Clone this wiki locally

<https://github.com/fbarresi/Sharp7.wiki.git>

