

نسخه یک مورخ 1401/02/19

# راهنمای دستورات DLL

مستند راهنمای نصب برای اجرا

شرکت توزین توان سنجش

واحد فناوری اطلاعات

## فهرست

1	فهرست
2	معرفی اجمالی DLL
2	شرایط اجرایی شدن کتابخانه Dll در Framework های مختلف
3	آشنایی با متغیرهای سیستم
4	مفهوم ScaleId (شناسه ترازو) در برنامه
4	ایجاد یک نمونه جدید از DLL
4	شرایط اجرایی کد نمونه سازی در حالت ارتباط سریال
5	شرایط اجرایی کد نمونه سازی در حالت ارتباط شبکه
6	دریافت فروش در برنامه
8	دستورات در ترازوهای معمولی MDS
13	دستورات در ترازوهای PLUS

## معرفی اجمالی DLL

به منظور برقراری ارتباط ترازو با سیستم کامپیوتر توسط کدهای دستوری C#، شرکت توزین توان سنجش محک اقدام به ارائه کتابخانه Dll کرده است که این کتابخانه به صورت یک Agent کامل، رابطه بین سیستم و ترازو را برقرار می کند. با استفاده از این Dll می توانید دستورات را به سمت ترازو ارسال نمایید و داده های ترازوی خود را مطابق با نیازهای خود مدیریت نمایید.

به منظور شروع استفاده از کتابخانه بعد از ایجاد یک پروژه جدید در ابتدا دو فایل NewDll و MahakScale را از طریق Dependency در قسمت Solution پروژه، Add Reference کنید تا شرایط اولیه برای کد نویسی دستورات فراهم شود.

## شرایط اجرایی شدن کتابخانه Dll در Framework های مختلف

در صورتی که نسخه پروژه ای که استفاده می کنید Net Framework 4.5.2. و یا Net Framework 4.7.2. می باشد نیاز به انجام عملیات خاصی در برنامه ندارید چرا که Framework ها به صورت کامل در Dependency ایجاد می شوند.

در صورتی که از نسخه **.net5** به بالا در پروژه خود استفاده می کنید بایستی دو پکیج System.IO.Ports و System.Text.Encoding.CodePages را نصب نمایید و سپس در Main برنامه از کد زیر استفاده نمایید:

```
public partial class App : Application
{
    protected override void OnStartup(StartupEventArgs e)
    {
        System.Text.Encoding.RegisterProvider(System.Text.CodePagesEncodingProvider.Instance);
    }
}
```

## آشنایی با متغیرهای سیستم

این متغیرها با توجه به نوع ترازویی (MDS یا PLUS) که می‌خواهید با آن ارتباط برقرار کنید متفاوت می‌باشد.

نام متغیر	نوع متغیر	کاربرد	MDS	PLUS
Network	bool	در حالت ارتباط <b>وای فای</b> True بوده و در حالت ارتباط <b>سریال</b> False می‌باشد.	-	-
SeverIP	string	فقط در حالت ارتباط <b>وای فای</b> کاربرد دارد که در واقع همان آی پی کارت شبکه شما می‌باشد و حتما در ترازو نیز باید تنظیم شده باشد. در حالت <b>سریال</b> مقدار آن "" باشد.	-	-
PortNumber	int	فقط در حالت ارتباط <b>وای فای</b> کاربرد دارد و به صورت پیش فرض مقدار آن 5000 می‌باشد. در حالت <b>سریال</b> مقدار آن را 0 قرار دهید.	-	-
ComPort	string	فقط در حالت ارتباط <b>سریال</b> کاربرد دارد و در واقع شماره پورت Com ایست که کابل ترازو را به سیستم متصل کرده است.	COM	COM
BaudRate	int	فقط در حالت ارتباط <b>سریال</b> کاربرد دارد و در واقع پهنای باند ترازو می‌باشد.	9600	115200
isNewProtcol	bool	در هر دو حالت <b>سریال</b> و <b>وای فای</b> کاربرد دارد.	False	True
isDIIWithSale	bool	در هر دو حالت <b>سریال</b> و <b>وای فای</b> کاربرد دارد و فروش های ترازو را دریافت می‌کند.	True	True
isDIIWithoutDatabase	bool	در هر دو حالت <b>سریال</b> و <b>وای فای</b> کاربرد دارد و به این معنی می‌باشد که برنامه به دیتابیس وصل نیست.	True	True
writeLog	bool	در هر دو حالت <b>سریال</b> و <b>وای فای</b> کاربرد دارد و به این معنی می‌باشد که اگر دستورات برنامه با خطایی مواجه شد آن را در Log نشان می‌دهد.	True	True

## مفهوم ScaleId (شناسه ترازو) در برنامه

در حالت ارتباط سریال ScaleId همان شماره پورت Com می باشد . به عنوان مثال اگر کابل ترازوی شما به پورت COM5 سیستم متصل است ScaleId معادل عدد 5 می باشد .

در حالت ارتباط شبکه ScaleId معادل HostID آی پی شما (بخش چهارم آی پی) می باشد . به عنوان مثال اگر آی پی ترازوی شما 192.168.15.125 باشد ScaleId معادل عدد 125 می باشد . بعد از به دست آوردن ScaleId می توانید با استفاده از دستور mahakScale.mahakScales[ScaleId].IsNewProtocol نوع ترازو (MDS یا PLUS) را بدست آورید .

## ایجاد یک نمونه جدید از DLL

حال که با مفهوم متغیرها آشنا شده ایم می توانیم یک نمونه جدید از کلاس NewMahakScale بسازیم . این کلاس مطابق با نمونه کد زیر دارای پارامترهای ورودی که قبلا توضیح داده شد می باشد.

```
mahakScale = new NewMahakScale (Network, ServerIP, PortNumber, ComPort, BaudRate, isNewProtocol,
isDIIWithSale, isDIIWithoutDatabase , writeLog);
```

**نکته:** از ورژن 6.15 کتابخانه به بعد، فیلد IsNewProtocol حذف شده است و می توان از سازنده ایی که یک ورودی کمتر می پذیرد استفاده نمائید. از آن جا که در ارتباط شبکه isNewProtocol کاربردی ندارد می توانید از ورودی زیر استفاده نمائید .

```
mahakScale = new NewMahakScale (Network, ServerIP, PortNumber, ComPort, BaudRate,
isDIIWithSale, isDIIWithoutDatabase , writeLog);
```

## شرایط اجرای کد نمونه سازی در حالت ارتباط با پورت سریال

در حالت ارتباط سریال بعد از اجرا کردن برنامه در صورتی که کد نمونه سازی بتواند پورت Com را شناسایی کند ارتباط برقرار است و یکی از آرایه های MahakScale را پر میکند که شماره این آرایه همان ScaleId می باشد . به منظور حصول اطمینان از برقرار بودن این ارتباط می توانید سه شرط IsActive و IsAvailable و IsPortOpen را مطابق کد زیر چک نمائید :

- mahakScale.mahakScales[ScaleId].IsActive
- mahakScale.mahakScales[ScaleId].IsAvailable
- mahakScale.mahakScales[ScaleId].IsPortOpen

**\*** در صورتی که سه دستور اشاره شده همگی True بود ارتباط کاملا برقرار است .

### شرایط اجرای کد نمونه سازی در حالت ارتباط شبکه

در حالت ارتباط شبکه ( به صورت وای فای) بعد از اجرا کردن کد نمونه سازی ، یک Event فراخوانی می شود که این Event در واقع تمامی آدرس های آی پی ترازوهایی که ServerIp آن ها بر روی کارت شبکه سیستم شما تنظیم شده است را شناسایی می کند و به قسمت Event می فرستد . به منظور فراخوانی Event بعد از ایجاد کد نمونه سازی از کد زیر استفاده نمائید :

```
mahakScale.FindScaleEvent += this.FindScale_ReceiveHandler;
```

در برنامه خود یک Event از نوع Delegate مطابق با کد زیر تعریف نمایید :

```
private delegate void FindScaleDelegate(Object result, Int32 Id);
private void FindScale_ReceiveHandler(Object result, Int32 Id)
{
    // می توانید آی پی های دریافتی را مطابق با نیاز خود مدیریت نمایید
    var IP = _mahakScale.mahakScales[Id].addressIP;
}
```

Id تعریف شده در متد در واقع همان ScaleId می باشد که یکی از خانه های آرایه mahakScales را پر کرده است . همچنین برای بدست آوردن scaleId با توجه به بخش چهارم آی پی (Host Id) می توانید از کد زیر نیز استفاده نمایید :

```
var _ScaleId = Convert.ToInt32(IPAddress.Parse(addressIP).GetAddressBytes()[3]);
```

در صورتی که در این قسمت هیچ آی پی را دریافت نکردید ورودی Server IP در کد نمونه سازی را بررسی نمایید و همچنین ServerIp و دیگر تنظیمات شبکه ترازوهای خود را نیز بررسی نمائید و همچنین از اتصال وای فای شبکه خود و پایداری آن نیز هم مطمئن شوید.

بعد از دریافت آی پی در قسمت FindScale\_ReceiveHandler به منظور حصول اطمینان از برقرار بودن این ارتباط می توانید سه شرط IsActive و IsAvailable و IsAvtiveIpPort را مطابق کد زیر چک نمایید :

- mahakScale.mahakScales[ScaleId].IsActive
- mahakScale.mahakScales[ScaleId].IsAvailable
- mahakScale.mahakScales[ScaleId]. IsAvtiveIpPort

**نکته:** در حالت ارتباط شبکه بر خلاف حالت سریال می توانید به چندین ترازو به صورت همزمان متصل شوید و ارتباط داشته باشید.

#### دریافت فروش در برنامه

- در مرحله اول برای ایجاد کد نمونه سازی فیلد isDllWithSale حتما برابر با True وارد شود .
- توجه داشته باشید روند دریافت فروش در هر دو حالت ارتباط سریال و یا شبکه یکسان می باشد. برنامه از یک Event به منظور دریافت فروش ها استفاده می کند و تمامی فروش های دریافتی را به این قسمت ارسال می کند .
- در صورتی که نیاز است ارسال فروش ترازوی خود را در طول برنامه متوقف کنید می توانید مطابق با کد زیر متغیر isReceivedSalePacket را برابر با True قرار دهید .

```
_mahakScale.mahakScales[_ScaleId].isReceivedSalePacket = true;
```

در صورت false کردن آن ارسال فروش به روال خود بر میگردد.

- بعد از ایجاد کد نمونه سازی در خط بعدی آن از کد زیر استفاده نمایید :

```
_mahakScale.ReceiveSalePacketEvent += this.AppendSale_ReceiveHandler;
```

در برنامه خود یک Event از نوع delegate مطابق با کد زیر تعریف نمایید :

```
private delegate void AppendSaleDelegate(ref List<Object[]> result);
private bool AppendSale_ReceiveHandler(List<Object[]> result)
{
    bool flag = true;
    try
    {
        for (int i = 0; i < result.Count;)
        {
            Object[] row = new object[15];
            row[0] = result[i][0];           // تاریخ فروش
            row[1] = convertTime5(result[i][1].ToString()); // ساعت فروش
            row[2] = result[i][2];           // آی پی شبکه و یا شماره پورت کام
            row[3] = result[i][3];           // شماره حافظه
            row[4] = result[i][4];           // شماره فاکتور فروش
            row[5] = result[i][5];           // فروش وزنی یا فروش تعدادی
            row[6] = result[i][6];           // قیمت واحد
            row[7] = result[i][7];           // فروش وزنی و مقدار وزنی آن
```

```

row[8] = result[i][8];           // قیمت کل
row[9] = result[i][9];           // شناسه ترازو ScaleId
row[10] = result[i][10];          // فروش تعدادی و مقدار تعدادی آن
row[11] = result[i][11];          // itemNumber
row[12] = result[i][12];          // serialNumber
row[13] = result[i][13];          // BsdNew
row[14] = result[i][14];          // در حالت فروش وزنی 0 و در حالت فروش تعدادی 1

result.RemoveAt(i);              // در نهایت فروش دریافتی از آرایه حذف می شود
}
}
catch (Exception ex)
{
    flag = false;
}
return flag;
}

```

\*توجه داشته باشید ترازو به ازای ارسال هر فروش حتما باید Pointer خود را یک واحد افزایش دهد تا فروش های بعدی دریافت شوند این به معنی این است که تنها یک بار فروش ها دریافت می شود. به این منظور در کد بالا یک متغیر flag از نوع bool تعریف می کنیم . در صورتی که روند دریافت موفقیت آمیز بود این flag را به ترازو ارسال می کند . در صورتی که برنامه شما به هر دلیل با خطا مواجه شود flag به حالت false برمیگردد تا این امکان فراهم شود که همان فروش را بتوانید مجددا دریافت نمایید .

متد تبدیل زمان درج فروش convertTime5 به صورت زیر میباشد:

```

private string convertTime5(string strTime)
{
    string intOut = "error";
    int intHour = 0;
    int intMinute = 0;
    int intSecond = 0;
    int intMiliSecond = 0;
    try
    {
        string strTemp = strTime;
        strTemp = strTime.Substring(0, strTime.IndexOf(":"));
        intHour = int.Parse(strTemp);
        strTime = strTime.Substring(strTime.IndexOf(":") + 1);
        if (strTime.IndexOf(":") >= 0)
        {

```



```

strTemp = strTime.Substring(0, strTime.IndexOf(":"));
intMinute = int.Parse(strTemp);
strTime = strTime.Substring(strTime.IndexOf(":") + 1);
if (strTime.IndexOf(":") > 0)
{
    strTemp = strTime.Substring(0, strTime.IndexOf(":"));
    intSecond = int.Parse(strTemp);
    strTime = strTime.Substring(strTime.IndexOf(":") + 1);
    strTemp = strTime;
    intMiliSecond = int.Parse(strTemp);
}
}
intOut = intHour + ":" + intMinute;
return intOut;
}
catch
{
    return "Error";
}
}

```

## دستورات در ترازوهای معمولی MDS

1. دستور قفل مدیریتی MDS :

```

if (mahakScale.mahakScales[scaleID].WriteManagerLockOn())
{
    var Answers = mahakScale.mahakScales[scaleID].Answer;
    Answers[0] = جواب تایید از سمت ترازو
}
else
{
    دستور به درستی اجرا نشد//
}

```

2. دستور باز گشایی قفل مدیریتی MDS:

```

if (mahakScale.mahakScales[scaleID].WriteManagerLockOff())

```

```
{
    var Answers = mahakScale.mahakScales[scaleID].Answer;
    Answers[0] = جواب تایید از سمت ترازو
}
else
{
    // دستور به درستی اجرا نشد
}
```

3. دستور خواندن وزن MDS :

```
if (mahakScale.mahakScales[scaleID].ReadWeight())
{
    var Answers = mahakScale.mahakScales[scaleID].Answer;
    Answers[0] = دریافت عدد وزن

    // دریافت عدد وزن با روش دیگر
    var Weight = _mahakScale.mahakScales[_ScaleID].Weight.ToString();
}
else
{
    // دستور به درستی اجرا نشد
}
```

4. دستور نوشتن تاریخ و ساعت MDS :

نام متغیر	نوع متغیر	توضیحات
CurrentMin	byte	دقیقه
CurrentHour	byte	ساعت
CurrentDay	byte	روز
CurrentMonth	byte	ماه
CurrentYear	ushort	سال

```
if (mahakScale.mahakScales[scaleID].WriteClockDate(CurrentMin, CurrentHour, CurrentDay,
    CurrentMonth, CurrentYear))
{
```

```

var Answers = mahakScale.mahakScales[scaleID].Answer;
Answers[1] = جواب تایید از سمت ترازو
}
else
{
    دستور به درستی اجرا نشد//
}

```

5. دستور خواندن تاریخ و ساعت MDS :

```

if (mahakScale.mahakScales[scaleID].ReadClockDate())
{
    var Answers = mahakScale.mahakScales[scaleID].Answer;
    Answers[0] = تاریخ ;
    Answers[1] = سال ;
    Answers[2] = ماه ;
    Answers[3] = روز ;
    Answers[4] = ساعت ;
    Answers[5] = دقیقه ;
    Answers[6] = ثانیه ;
}
else
{
    دستور به درستی اجرا نشد//
}

```

6. دستور نوشتن سطح دسترسی MDS:

نام متغیر	نوع متغیر	توضیحات
IDAcc	byte	سطح دسترسی تنظیمات اصلی ترازو
BoudRate	byte	سطح دسترسی صفحه کلید اعداد ترازو
RenameMemoryAcc	byte	سطح دسترسی مدیریت کالای ترازو
RenameShopNameAcc	byte	سطح دسترسی تغییر نام فروشگاه ترازو
ChangeClockDateAcc	byte	سطح دسترسی تغییر تاریخ و ساعت ترازو
ChangeDivisionAcc	byte	سطح دسترسی تغییر دقت ترازو

در صورتی که مقدار 1 باشد به معنای True و اگر 0 باشد به معنای False می باشد .

```
if (mahakScale.mahakScales[scaleID].WriteAccessCode(IDAcc, BoudRate, RenameMemoryAcc,
RenameShopNameAcc, ChangeClockDateAcc, ChangeDivisionAcc))
{
    var Answers = mahakScale.mahakScales[scaleID].Answer;
    Answers[1] = جواب تایید از سمت ترازو
}
else
{
    دستور به درستی اجرا نشد
}
```

7. دستور نوشتن پیام ابتدا (نام فروشگاه) و پیام انتهای فاکتور فروش MDS :

نام متغیر	نوع متغیر	توضیحات
shopName	string	نام فروشگاه (پیام ابتدای فاکتور)
endMessage	string	پیام انتهای فاکتور

```
if (mahakScale.mahakScales[scaleID].WriteDescription (shopName, endMessage))
{
    var Answers = mahakScale.mahakScales[scaleID].Answer;
    Answers[1] = جواب تایید از سمت ترازو
}
else
{
    دستور به درستی اجرا نشد
}
```

7. دستور خواندن پیام ابتدا (نام فروشگاه) و پیام انتهای فاکتور فروش MDS :

```
if (mahakScale.mahakScales[scaleID].ReadDescription ())
{
    var Answers = mahakScale.mahakScales[scaleID].Answer;
    Answers[0] = نام فروشگاه (پیام ابتدای فاکتور)
    Answers[1] = پیام انتهای فاکتور
}
else
{
    دستور به درستی اجرا نشد
}
```

## 8. دستور ثبت کالا روی حافظه ترازو MDS:

نام متغیر	نوع متغیر	توضیحات
memoryNumber	ushort	شماره حافظه ترازو
memoryType	byte	وزنی یا تعدادی بودن کالا (وزنی = 0 ، تعدادی = 1)
memoryName	string	نام کالا
unitPrice	uint	قیمت واحد (حداکثر 5 رقم)
gPrice	uint	در حال حاضر کاربردی ندارد (0)
gWeight	uint	در حال حاضر کاربردی ندارد (0)
gChange	bool	در حال حاضر کاربردی ندارد (true)
barcode	string	بارکد کالا

```

if (mahakScale.mahakScales[scaleID].WriteDirectMemory (memoryNumber, memoryType,
memoryName, unitPrice, gPrice, gWeight, gChange, barcode))
{
    var Answers = mahakScale.mahakScales[scaleID].Answer;
    Answers[1] = جواب تایید از سمت ترازو =
}
else
{
    دستور به درستی اجرا نشد//
}

```

**نکته:** نحوه درج بارکد کالا در ترازوهای معمولی MDS را می توانید به تعداد دلخواه و حداکثر تا 13 رقم استاندارد به کالا اختصاص دهید . در صورتی که از 13 رقم کمتر بارکد وارد نمایید Dll به تعداد ارقام باقیمانده تا 13 رقم ، در پشت عدد مورد نظر صفر درج کرده و رقم آخر توسط سیستم ترازو به صورت اتوماتیک محاسبه می شود . به عنوان مثال اگر بارکد 5 رقمی 69874 را در نظر بگیرید خروجی نهایی در ترازو بعد از ثبت ، معادل 0000000698740 خواهد بود که عدد 0 در خروجی نهایی محاسبه خود ترازو می باشد.

## 8. دستور خواندن کالا از حافظه ترازو MDS :

نام متغیر	نوع متغیر	توضیحات
memoryNumber	ushort	شماره حافظه

```

if (mahakScale.mahakScales[scaleID].ReadDirectMemory(memoryNumber))
{
    var Answers = mahakScale.mahakScales[scaleID].Answer;
    Answers[0] = شماره حافظه ;
    Answers[1] = نام کالا ;
    Answers[2] = قیمت واحد ;
    Answers[3] (gPrice) = در حال حاضر کاربردی ندارد ;
    Answers[4] (gWeight) = در حال حاضر کاربردی ندارد ;
    Answers[5] = تعدادی و یا وزنی بودن کالا (وزنی = 0 ، تعدادی = 1) ;
    Answers[60] = ثانیه ;
}
else
{
    دستور به درستی اجرا نشد//
}

```

## دستورات در ترازوهای PLUS

1. دستور قفل مدیریتی PLUS :

```

if (mahakScale.mahakScales[scaleID].SetManagerLock(true))
{
    var Answers = mahakScale.mahakScales[scaleID].Answer;
    Answers[0] = جواب تایید از سمت ترازو
}
else
{
    دستور به درستی اجرا نشد//
}

```

2. دستور باز گشایی قفل مدیریتی PLUS:

```

if (mahakScale.mahakScales[scaleID].SetManagerLock(false))
{
    var Answers = mahakScale.mahakScales[scaleID].Answer;
    Answers[0] = جواب تایید از سمت ترازو
}
else
{
    دستور به درستی اجرا نشد//
}

```

3. دستور خواندن وزن PLUS :

```

if (mahakScale.mahakScales[scaleID]._GetWeight ())
{
    var Answers = mahakScale.mahakScales[scaleID].Answer;
    Answers[0] = دریافت عدد وزن

    دریافت عدد وزن با روش دیگر//
    var Weight = _mahakScale.mahakScales[_ScaleId].Weight.ToString();
}
else
{
    دستور به درستی اجرا نشد//
}

```

4. دستور نوشتن تاریخ و ساعت PLUS :

نام متغیر	نوع متغیر	توضیحات
CurrentYear	ushort	سال
CurrentMonth	byte	ماه
CurrentDay	byte	روز
CurrentMode	byte	نوع تاریخ (شمسی = 0 ، میلادی = 1)
CurrentHour	byte	ساعت
CurrentMin	byte	دقیقه
CurrentSecond	byte	ثانیه

```

if (mahakScale.mahakScales[scaleID]._SetDateClock(CurrentYear, CurrentMonth, CurrentDay,
CurrentMode, CurrentHour, CurrentMin, CurrentSecond))
{
    var Answers = mahakScale.mahakScales[scaleID].Answer;
    Answers[1] = جواب تایید از سمت ترازو
}
else
{
    //دستور به درستی اجرا نشد
}

```

5. دستور خواندن تاریخ و ساعت PLUS :

```

if (mahakScale.mahakScales[scaleID]._GetDateClock())
{
    var Answers = mahakScale.mahakScales[scaleID].Answer;
    Answers[0] = تاریخ ;
    Answers[1] = روز ;
    Answers[2] = نوع تاریخ (شمسی = 0 ، میلادی = 1) ;
    Answers[3] = سال ;
    Answers[4] = ماه ;
    Answers[5] = روز ;
    Answers[6] = ساعت ;
    Answers[7] = دقیقه ;
    Answers[8] = ثانیه ;
}
else { دستور به درستی اجرا نشد }

```

6. دستور نوشتن سطح دسترسی PLUS :

نام متغیر	نوع متغیر	توضیحات
Active	bool	فعال یا غیر فعال کردن سطح دسترسی

سطح دسترسی در ترازوهای PLUS در سه سطح می تواند بررسی شود :

a- سطح دسترسی صفحه کلید ترازو :

```

if(mahakScale.mahakScales[scaleID]._SetKeyboardDigitsAccess(Active))
{
    var Answers = mahakScale.mahakScales[scaleID].Answer;
    Answers[1] = جواب تایید از سمت ترازو
}

```



```

}
else
{
    دستور به درستی اجرا نشد//
}

```

b- سطح دسترسی تنظیمات اصلی ترازو :

```

if(mahakScale.mahakScales[scaleID]._SetSettingsAccess(Active))
{
    var Answers = mahakScale.mahakScales[scaleID].Answer;
    Answers[1] = جواب تایید از سمت ترازو
}
else
{
    دستور به درستی اجرا نشد//
}

```

c- سطح دسترسی مدیریت کالای ترازو :

```

if(mahakScale.mahakScales[scaleID].SetKalaDatabaseAccess(Active))
{
    var Answers = mahakScale.mahakScales[scaleID].Answer;
    Answers[1] = جواب تایید از سمت ترازو
}
else
{
    دستور به درستی اجرا نشد//
}

```

7. دستور نوشتن پیام ابتدای فاکتور فروش PLUS :

نام متغیر	نوع متغیر	توضیحات
LabelFirstMessage	string	پیام ابتدای فاکتور

```

if (mahakScale.mahakScales[scaleID]. _SetLabelFirstMessage(LabelFirstMessage))
{
    var Answers = mahakScale.mahakScales[scaleID].Answer;
    Answers[0] = جواب تایید از سمت ترازو
}
else
{
    دستور به درستی اجرا نشد//
}

```

8. دستور خواندن پیام ابتدای فاکتور فروش PLUS:

```
if (mahakScale.mahakScales[scaleID]._GetLabelFirstMessage())
{
    var Answers = mahakScale.mahakScales[scaleID].Answer;
    Answers[0] = پیام ابتدای فاکتور فروش
    Answers[1] = جواب تایید از سمت ترازو
}
else
{
    //دستور به درستی اجرا نشد
}
```

9. دستور نوشتن پیام انتهای فاکتور فروش PLUS:

نام متغیر	نوع متغیر	توضیحات
LabelLastMessage	string	پیام انتهای فاکتور

```
if (mahakScale.mahakScales[scaleID]. _SetLabelLastMessage(LabelFirstMessage))
{
    var Answers = mahakScale.mahakScales[scaleID].Answer;
    Answers[0] = جواب تایید از سمت ترازو
}
else
{
    //دستور به درستی اجرا نشد
}
```

10. دستور خواندن پیام انتهای فاکتور فروش PLUS:

```
if (mahakScale.mahakScales[scaleID]._GetLabelLastMessage ())
{
    var Answers = mahakScale.mahakScales[scaleID].Answer;
    Answers[0] = پیام انتهای فاکتور فروش
    Answers[1] = جواب تایید از سمت ترازو
}
else
{
    //دستور به درستی اجرا نشد
}
```

# 11 . دستور ثبت کالا روی حافظه ترازو PLUS:

نام متغیر	نوع متغیر	توضیحات
Code	ushort	شماره حافظه ترازو
Active	byte	فعال بودن کالا
Name	string	نام کالا
LenBarcode	uint	طول بارکد (8 و 13 رقمی به صورت استاندارد)
Barcode	uint	بارکد کالا
Vaziat	uint	وزنی یا تعدادی بودن کالا (وزنی = 0 ، تعدادی = 1)
FiForoosh	bool	قیمت واحد کالا
Gorooh	string	گروه کالا
YearTarikhTolid	int	سال تولید کالا
MonthTarikhTolid	byte	ماه تولید کالا
DayTarikhTolid	byte	روز تولید کالا
YearTarikhEngheza	ushort	سال انقضا کالا
MonthTarikhEngheza	byte	ماه انقضا کالا
DayTarikhEngheza	byte	روز انقضا کالا
GoroohTakhfif	byte	گروه تخفیف کالا
Sharh	string	شرح کالا

```

if (mahakScale.mahakScales[scaleID]._SetKala(Code, Active, Name, LenBarcode, Barcode, Vaziat,
FiForoosh, Gorooh, YearTarikhTolid, MonthTarikhTolid, DayTarikhTolid, YearTarikhEngheza,
MonthTarikhEngheza, DayTarikhEngheza, GoroohTakhfif, Sharh))
{
    var Answers = mahakScale.mahakScales[scaleID].Answer;
    Answers[1] = جواب تایید از سمت ترازو
}
else
{
    دستور به درستی اجرا نشد
}

```

**نکته:** نحوه درج بارکد کالا در ترازوهای PLUS به این گونه می باشد که می توانید فقط یا 8 رقم و یا حداکثر تا 13 رقم استاندارد به کالا اختصاص دهید . بر خلاف MDS در صورتی که رقم وارد شده بارکد 8 و 13 رقم وارد نشده باشد مابقی اعداد بارکد در ترازو خالی درج می شود .

12 . دستور خواندن کالا از حافظه ترازو :

نام متغیر	نوع متغیر	توضیحات
memoryNumber	ushort	شماره حافظه

```
if (mahakScale.mahakScales[scaleID]._GetKala(memoryNumber))
{
    var Answers = mahakScale.mahakScales[scaleID].Answer;
    Answers[0] = شماره حافظه ;
    Answers[1] = وضعیت فعال بودن کالا ;
    Answers[2] = نام کالا ;
    Answers[3] = بارکد کالا ;
    Answers[4] = تعدادی و یا وزنی بودن کالا ;
    Answers[5] = قیمت واحد کالا ;
    Answers[6] = گروه کالا ;
    Answers[7] = تاریخ تولید کالا ;
    Answers[8] = تاریخ انقضا کالا ;
    Answers[9] = گروه تخفیف کالا ;
    Answers[10] = شرح کالا ;
}
else { دستور به درستی اجرا نشد // }
```