

راهنمای استفاده از

Eos.Clock.dll

برای برقراری ارتباط و استفاده از

دستگاههای

رستوران علم و صنعت

لازم به ذکر است برنامه مذکور به زبان C# و تحت .net framework نسخه ۲ نوشته و کامپایل شده است و توسط این نسخه از framework و ویرایش های بالاتر قابل استفاده است.

برای برقراری ارتباط با دستگاه رستوران ابتدا بایستی به ساعت متصل شد به این منظور ابتدا یک ساعت را در کد برنامه تعریف می نماییم و به آن ساعت وصل می شویم که طریقه ساخت ساعت و اتصال به آن به شکل زیر است :

ایجاد یک Connection با استفاده از کلاس ConnectionFactory :

در این حالت با توجه به اینکه طریقه ارتباط با ساعت به چه صورت است از یکی از متدهای زیر استفاده می نماییم :

- ارتباط سریال :

```
public Connection GetSerialConnection(int portNumber, int baudRate,
    Parity parity, int readTimeout, int writeTimeout, int waitBeforeRead)
{
    Connection connection = ConnectionFactory.CreateSerialConnection(
        portNumber, baudRate, parity, readTimeout, writeTimeout, waitBeforeRead);
    return connection;
}
```

- ارتباط شبکه :

```
public Connection GetTcpIpConnection(string ip, int port, int waitBeforeRead,
    int readTimeout, int writeTimeout)
{
    Connection connection = ConnectionFactory.CreateTCPIPConnection(ip, port,
        waitBeforeRead, readTimeout, writeTimeout);
    return connection;
}
```

شرح پارامترهای استفاده شده به شرح زیر است :

portNumber این پارامتر در ارتباط مودمی و سریال شماره پورت سریال را مشخص میکند و در ارتباط شبکه شماره پورت ارتباطی شبکه را مشخص می کند.

baudRate پهنای باند در ارتباط مودمی و سریال میباشد .

parity اطلاعات پریتهی ارتباط که از طریق ساختار System.Io.Ports.Parity قابل تنظیم است.

readTimeout زمان مجاز برای گرفتن جواب زمان خواندن اطلاعات را در ارتباط نشان می دهد .

writeTimeout زمان مجاز برای گرفتن جواب زمان نوشتن اطلاعات را در ارتباط نشان می دهد .

waitBeforeRead مدت زمانی که ارتباط و قبل از هر خواندن اطلاعات بایستی صبر کند را نمایش می دهد.

lp این پارامتر آدرس ساعت را در ارتباط شبکه مشخص می کند.

پس از ایجاد یک شیء connection نوبت به ایجاد ساعت می رسد که روش آن به شرح زیر است :

```
public Clock GetRestaurantClock(Connection connection, ProtocolType protocolType,
                                int trtNumber, ProtocolType sensorType)
{
    Clock device = new RestaurantClock(connection, protocolType, trtNumber, sensorType);
    return device;
}
```

شرح پارامترهای استفاده شده به شرح زیر است :

Connection که شیء ارتباط ساخته شده در قسمت بالا می باشد.
protocolType که پروتکل ارتباطی ساعت را نشان می دهد
trtAddress آدرس trt ساعت را نشان می دهد که در ساعتهای سریال با پروتکل RS486 کاربرد دارد .
sensorType هرگاه دستگاه رستوران شامل ماژول اثر انگشت هم باشد میتوانید نوع آنرا تعیین کنید. اگر در مورد نوع آن اطمینان ندارید میتوانید از ProtocolType.Suprema استفاده کنید.

بعد از این مراحل بایستی به ساعت وصل شویم و مطمئن شویم که اتصال بدرستی برقرار گردیده است :

- با استفاده از دستور Connect :

```
device.Connect();
if (!device.Connected)
{
    MessageBox.Show(@"error in connection");
    return;
}
```

- با استفاده از دستور TestConnection :

```
if (!device.TestConnection())
{
    MessageBox.Show(@"error in connection");
    return;
}
```

در هر دوی این موارد ارتباط با ساعت برقرار میگردد .

لازم بذکر است در ارتباط مودمی ابتدا بایستی شماره گیری انجام گیرد . این کار از طریق دستورات بالا انجام می گیرد ولی زمان بیشتری نسبت به سایر روشها دارد .

لیست عملیات رستوران

```
public override bool SendValidCards(List<string> validList)
```

جهت ارسال لیستی از شماره کارتهای معتبر استفاده میشود. در این حالت تمامی کارتها مجاز به استفاده از تمامی وعده های غذایی تعریف شده در ساعت میباشند.

```
public bool SendValidCards(List<RestaurantValidList> validList)
```

جهت ارسال لیستی از شماره کارتهای معتبر به همراه وعده های مجاز برای استفاده که قابل تعریف برای هر کارت است، استفاده میشود. ساختار RestaurantValidList به این صورت است که پنج وعده بصورت Boolean و یک شماره کارت را در بر میگیرد.

```
public bool SendDayMealsData(RestaurantDayMealsData restaurantDayMealsData)
```

برای ارسال تقویم مربوط به یک روز هفته استفاده میشود. بدیهی است جهت ارسال تقویم یک هفته باید برای تک تک روزهای هفته این تابع را بکار ببرید. ساختار RestaurantDayMealsData به شرح زیر میباشد.

CompanyName

نام شرکت

DayOfWeek

روز از هفته

MealDataCollection

لیست اطلاعات پنج وعده قابل تعریف در روز که به شرح زیر است

MealName

نام وعده

StartTime

زمان شروع وعده

EndTime

زمان پایان وعده

Food1Name

نام غذای اول

Food2Name

نام غذای دوم

Food3Name

نام غذای سوم

Dessert1Name

نام دسر اول

Dessert2Name

نام دسر دوم

```
public bool SendWeekReserveList(List<RestaurantPersonWeekReserveData> weekReserveList)
```

برای ارسال لیست اطلاعات رزرو غذای پرسنل به مدت یک هفته بکار میرود. ساختار RestaurantPersonWeekReserveData به شرح زیر میباشد.

CardNo

شماره کارت

HasMeal1Access

دسترسی به وعده ۱

HasMeal2Access

دسترسی به وعده ۲

MaxRepeatCount

حداکثر غذای مصرفی

PersonDayReserveData

AllowReserve اجازه رزرو دارد یا نه
LimitCardRepeat به حداکثر غذای مصرفی محدود باشد یا نه
Dessert1Reserved دسر اول را رزرو کرده یا نه
Dessert2Reserved دسر دوم را رزرو کرده یا نه
ReservedFood

None غذا قابل انتخاب از روی دستگاه باشد
Food1 غذای ۱ را رزرو کرده
Food2 غذای ۲ را رزرو کرده
Food3 غذای ۳ را رزرو کرده

```
public bool SetReservePcCheck(bool isOk, string message1, string message2, ReserveData reserveData)
```

به هنگام کنترل نرم افزاری صدور فیش در جواب شماره کارت ارسالی از دستگاه رستوران با این تابع تایید و یا عدم تایید شخص، همچنین در صورت نیاز خط اول و خط دوم پیام که بر روی صفحه نمایشگر ساعت نشان داده خواهد شد، و در حالت رزرو اطلاعات رزرو شخص را میتوان به دستگاه ارسال کرد. ساختار ReserveData به شرح زیر می باشد.

AllowSelectFood اجازه انتخاب غذا را داشته باشد یا نه
ReservedFood
None غذا قابل انتخاب از روی دستگاه باشد
Food1 غذای ۱ را رزرو کرده
Food2 غذای ۲ را رزرو کرده
Food3 غذای ۳ را رزرو کرده

Dessert1Reserved دسر اول رزرو شده یا نه
Dessert2Reserved دسر دوم رزرو شده یا نه