ارتباط با دستگاه های حضور و غیاب

بر پایه *EosClocks.dll*

مستند شرکت علم و صنعت در خصوص نحوه اتصال , استخراج و مدیریت دستگاه های
مدل ST-2000 و ST-Pro

شرکت علم و صنعت
دی ۱۳۹۱ – نسخه ۲٫۰

شرکت علم و صنعت به منظور سهولت برقراری ارتباط و کار با سخت افزارها مرتبط با محصولات نرم افزاری خود اقدام به آماده سازی کتابخانه ای در معماری Net. کرده , سعی شده این کتابخانه به گونه طراحی شود که با کمترین آگاهی از مشخصات سخت افزارها بتوان به سهولت با آنها ارتباط برقرار کرد و عملیات استخراج رکوردها و تغییرات در پیکربندی ها به راحتی امکان پذیر باشد.

قبل ار هر چیز باید بدانید که برای شروع اتصال به هر کدام از سخت افزارها باید یک کلاس از جنس Clock ساخته شود. تمامی کلاسهایی که برای اتصال به سخت افزار های طراحی شده از این کلاس پایه به ارث برده شده اند. کلاس Clock در متد سازنده خود یک نمونه از کلاس Connection را برای مشخص کردن نوع پروتوکل ارتباطی می پذیرد , پس قبل از ساختن Clock باید یک Connection ساخته باشیم. برای ساختن Connection نیز باید از متد های Factory تهیه شده در کلاس ConnectionFactory استفاده کنیم

ساخت Connection

```
18      Connection CreateConnection(int portNumber,int baudRate,Parity parity)
19      {
20          try
21          {
22              Connection con = ConnectionFactory.CreateSerialConnection(portNumber, baudRate, parity, 4000, 5000, 500);
23              if (con != null)
24                  return con;
25          }
26          catch
27          {
28              throw new Exception("Error in create connection");
29          }
30          return null;
31      }
```

قطعه کد بالا متدی را نشان میدهد که یک Connection میسازد با پارامترهای ورودی ConnectionFactory توجه کنید , هنگام ساختن یک یک Connection برای ارتباط سریال پارامتر PortNumber شماره درگاه Com را مشخص می کند. متدهای سازنده دیگری برای ارتباط TCP و Modem وجود دارد

سازنده Connection از نوع TCP:

```
62      public static Connection CreateTCPIPConnection(string ip, int port, int waitBeforeRead,
63          int readTimeOut, int writeTimeOut, bool isProtected, string login,
64          string password, NetworkModuleType moduleType)
```

سازنده Connection از نوع Modem:

```
public static Connection CreateModemConnection(int portNumber, int baudRate, System.IO.Ports.Parity parity,
    int readTimeOut, int writeTimeOut, int waitBeforeRead, string telephoneNumber, bool isPulseDialing)
```

پارامترها:

readTimeOut : مدت زمانی که برای خوانده اطلاعات از کانال ارتباطی در هر بار سعی برای خوانده در نظر گرفته می شود (بر حسب میلی ثانیه)

writeTimeOut: مدت زمانی که برای نوشتن اطلاعات روی کانال ارتباطی در هر بار سعی برای خوانده در نظر گرفته می شود (بر حسب میلی ثانیه)

waitBeforeRead: مدت زمان انتظار قبل از خوانده اطلاعات(بر حسب میلی ثانیه)

اتصال و تست برقراری ارتباط

```
var myConnection=CreateConnection(1, 9600, Parity.None);
myConnection.Connect();
if (myConnection.Connected)
{
    //Do something...
}
else
{
    throw  new Exception("Unable connecto to the device");
}
```

همانطور که در تصویر مشاده میکنید بعد از ساختن Connection می توانید با استفاده از متد Connect به دستگاه مورد نظر متصل شوید در صورتیکه عملیات اتصال موفقیت آمیز باشید مقدار ویژپی Connected از کلاس Connection برابر true خواهد بود و در غیر اینصورت مقدار false را دریافت خواهید کرد

حال میتوانید به طریق زیر اقدام به ساخت یک نمونه از کلاس Clock نمایید:

```
Clock CreateClock(Connection connection,ProtocolType protocolType, int trtNumber)
{
    try
    {
        return new Clock(connection, protocolType, trtNumber, ProtocolType.Suprema);
    }
    catch
    {
        throw new Exception("Unable to create an instance of clock");
    }
}
```

همانطور که مشاهده میکنید پارامترهای زیر در هنگام ساختن Clock مورد نیاز می باشد:

protocolType : نوع پروتوکل ارتباطی فعلی دستگاه که معمولاً یکی از حالات RS232و RS485 می باشد را معین می کند

trtNumber: از انجایی که در پروتوکل RS485 میتوان تعداد ۳۲ دستگاه را همزمان توسط یک خط ارتباطی کنترل کرد برای مشخص کردن دتسگاه مورد نظر از یک مشخصه به نام TRT که قبلا روی هر دستگاه به صورت منحصر به فرد تنظیم شده (یک عدد بین ۱تا۳۲) استفاده می شود

حال که به ساعت متصل شده ایم میتوانیم عملیات خوانده رکوردهای را انجام دهیم:

```
31  private List<ClockRecord> StartGetRecords(Clock clock)
32  {
33      var recordList=new List<ClockRecord>();
34      while (clock.IsEmpty())
35      {
36          try
37          {
38              var record = (ClockRecord)clock.GetRecord();
39              recordList.Add(record);
40              clock.NextRecord();
41          }
42          catch (Exception ex)
43          {
44              if ((ex is InvalidRecordException) || (ex is InvalidDataInRecordException))
45              {
46                  // you can find bad record data in ex.Data["RecordRawData"]
47                  MessageBox.Show("Bad Record :" + Environment.NewLine + ex.Data["RecordRawData"]);
48              }
49              else
50                  MessageBox.Show(ex.ToString());
51          }
52      }
53      MessageBox.Show("Operation complete successfully");
54      return recordList;
55  }
```

متد IsEmpty برای تست وجود رکوردی جدید برای خوانده شدن استفاده می شود

متد GetRecord رکورد جاری را خوانده و نتیجه را در قالب Record بر میگرداند که در مثل بالا این نوع ClockRecord تبدیل شده

متد NextRecord : اشاره گر رکورد ها را به رکورد بعدی منتقل می کند

توضیح: در سخت افزار های تولید شده رکوردهای به صورت ترتیبی در حافظه داخلی دستگاه ذخیره می شوند و یک اشاره گر برای خوانده رکوردهای وجود دارد که به رکورد جاری اشاره میکند, بعد از هر بار عملیات خواندن توسط GetRecord بایستی اشاره گر را توسط NextRecord به رکورد بعدی منتقل کرد.

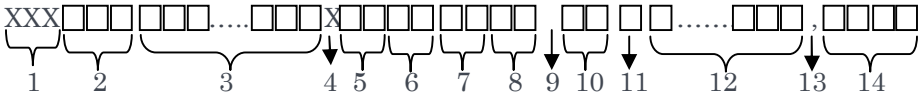
بنابراین توسط قطعه کد بالا تا زمانی که رکورد جدیدی برای خوانده شدن وجود نداشته باشد عملیات خواندن رکوردهای یکی پس از دیگری انجام میشود و در نهایت لیستی از رکوردهای خوانده شه توسط متد بازگردانده میشود

در ادامه به معرفی چند متد کاربردی دیگر از کلاس Clock می پردازیم:

`public override DateTime GetDateTime()` تاریخ و زمان جاری دستگاه را بازمی گرداند
`public override void SetDateTime(DateTime dateTime)...` تاریخ و زمان دستگاه را تنظیم می کند
`public override string GetFirmwareVersion()` نسخه نرم افزار کنترلر دستگاه را باز می گرداند
`public override string GetModel()` مدل (نسل) ساخت دستگاه را باز میگرداند
`public bool TestConnection()` وضعیت اتصال یا عدم اتصال به دستگاه را یک بار دیگر بررسی می کند و نتیجه را باز می گرداند
`public virtual void Disconnect()` اتصال موجود به دستگاه را می بندد

ELMOSANAT Co. Ltd.
شرکت کامپیوتری علم و صنعت

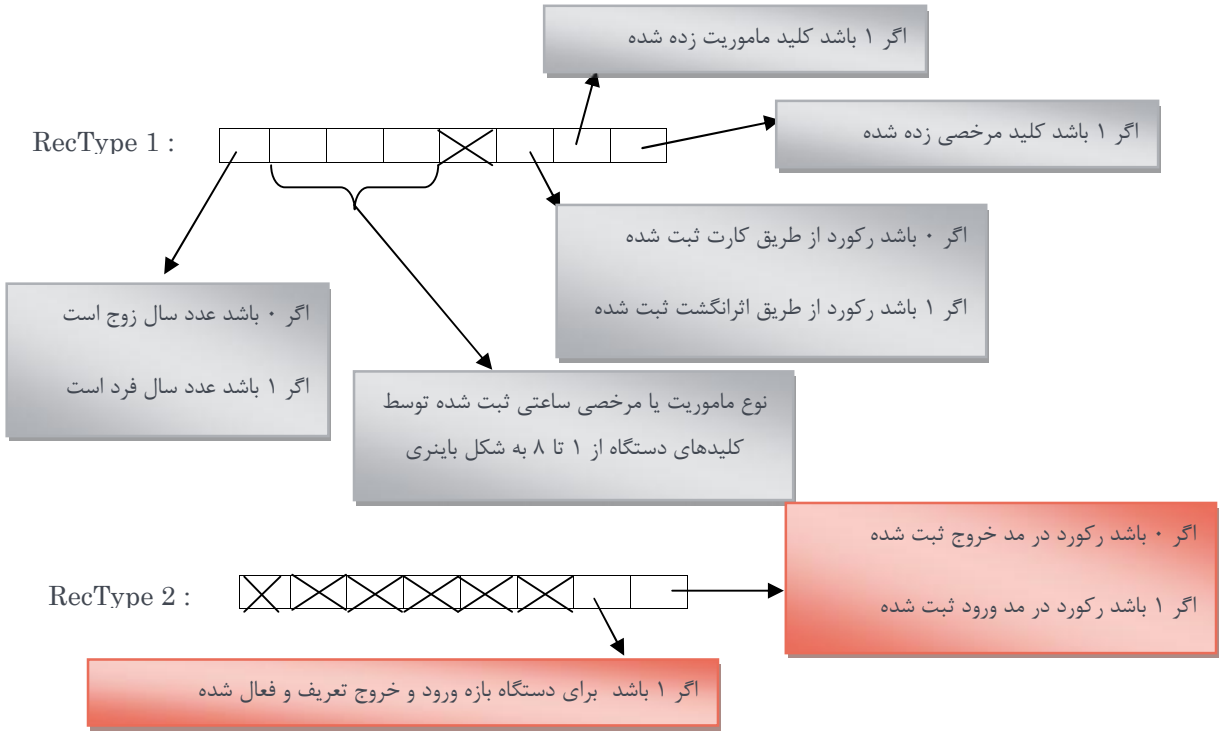
- **فرمت رکورد خام دریافتی از dll فایل مربوط به دستگاه های مدل ST-2000 و ST-Pro علم و صنعت با نام EOSClocks.dll**



طول رکورد در این ۲ مدل معادل ۱۰ بایت است

۱) سه کاراکتر به عنوان don't care که در نظر گرفته نمی شوند.
۲) شماره TRT دستگاه کارتخوان
۳) شماره کارت (۸ یا ۱۰ رقم)
۴) حرف X بزرگ
۵) ساعت ۲ رقم
۶) دقیقه ۲ رقم
۷) ماه ۲ رقم
۸) روز ۲ رقم
۹) اگر رکورد از نوع ورود یا خروج عادی است یک blank ، اگر از نوع مرخصی است حرف E و اگر از نوع ماموریت است حرف F
۱۰) سال ۲ رقم
۱۱) Sequence Code بر اساس حروف الفبای انگلیسی
۱۲) شماره Com یا آدرس IP دستگاه کارتخوان تا کاراکتر کاما
۱۳) کاراکتر کاما
۱۴) RecType 1 و RecType2 برای دریافت پارامترهای مربوط به نوع مرخصی یا ماموریت و ثبت از طریق اثرانگشت یا کارت

(توضیحات مربوط به این قسمت در صفحه بعد)


اگر ۱ باشد کلید ماموریت زده شده
RecType 1 :
اگر ۱ باشد کلید مرخصی زده شده
اگر ۰ باشد رکورد از طریق کارت ثبت شده
اگر ۱ باشد رکورد از طریق اثرانگشت ثبت شده
اگر ۰ باشد عدد سال زوج است
اگر ۱ باشد عدد سال فرد است
نوع ماموریت یا مرخصی ساعتی ثبت شده توسط
کلیدهای دستگاه از ۱ تا ۸ به شکل باینری
اگر ۰ باشد رکورد در مد خروج ثبت شده
RecType 2 :
اگر ۱ باشد رکورد در مد ورود ثبت شده
اگر ۱ باشد برای دستگاه بازه ورود و خروج تعریف و فعال شده