

Als erstes eine neues Projekt starten und einen Verweis auf MoBaSRCP.dll erstellen

Auf den namespace verweisen mit: using MoBaSRCP;

So könnte der Code zum Verbinden mit dem Server aussehen:

```
private TMoBaSRCPClient client = null;
client = new TMoBaSRCPClient();
client.Host = "localhost";
client.Port = 4303;
client.Connect();
```

und zum Beenden der Verbindung:

```
client.Disconnect();
```

Der Code zum Erstellen eines Lok Objektes:

```
private TMoBaAdvancedLoco floco = null;
private TMoBaLocoProtocol lp = null;

lp = new TMoBaLocoProtocol();
lp.ProtocolBase = TMoBaProtocolBase.mpbMotorola;
lp.MaxFunctions = 5;
lp.MinSteps = 0;
lp.MaxSteps = 14;
lp.ProtocolExtension = 2;
floco = new TMoBaSRCPLoco();
floco.Protocol = lp;
floco.Address = 3;
floco.MaxSpeed = 100;
floco.Speed = 0;
floco.DriveMode = TMoBaDriveMode.mdmForward;
floco.Description = "Testlock zum Prüfen der MoBa Funktionen!";
```

Der Code zum Erstellen eines Solenoid Objektes:

```
private TMoBaAdvancedSolenoid fsol = null;
private TMoBaSolenoidProtocol sp = null;

sp = new TMoBaSolenoidProtocol();
sp.ProtocolBase = TMoBaProtocolBase.mpbMotorola;
sp.MaxPort = 2;
fsol = new TMoBaAdvancedSolenoid();
fsol.Description = "Dies ist ein Testobjekt!";
fsol.Protocol = SP;
fsol.ActivationTime = 220;
fsol.Address = 13;
fsol.SetPorts(1, 1); oder fsol.SetPorts(0, 1);
```

Der Code zum Übermitteln der Lok Daten an den Server:

```
TMoBaSRCPBus ABus = null;
```

```

int i = 0;
bool found = false;
int MMbusnr = 1;

if (null != srcpClient)
{
    if (srcpClient.Connected)
    {
        ABus = srcpClient.GetBusses(MMbusnr);
        if (null != ABus.GLs)
        {
            if (null != ABus.GLs)
            {
                for (i = 0; i < ABus.GLs.Count; i++)
                {
                    if (ABus.GLs.GetItems(i).Address ==
                        ALoco.Address)
                    {
                        found = true;
                        break;
                    }
                }
            }
            if (!found)
            {
                ABus.GLs.Add(ALoco);
            }
            else
            {
                ABus.GLs.Change(ALoco);
            }
        }
    }
}
}

```

Die Loco-Events:

```

client.OnLocoInitEvent += new
    TMoBaSRCPLocoDel(srcpClient_OnLocoInitEvent);
client.OnLocoTermEvent += new
    TMoBaSRCPLocoDel(srcpClient_OnLocoTermEvent);
client.OnLocoChangeEvent += new
    TMoBaSRCPLocoDel(srcpClient_OnLocoChangeEvent);
client.OnLocoLockEvent += new
    TMoBaSRCPLocoDel(srcpClient_OnLocoLockEvent);
client.OnLocoUnlockEvent += new
    TMoBaSRCPLocoDel(srcpClient_OnLocoUnlockEvent);

```

Der Code zum Übermitteln der Solenoid Daten an den Server:

```

TMoBaSRCPBus Bus = null;
int Busnr = 0;
if ((fsol != null) &&(fsol.Protocol != null))

```

```

{
    if (fsol.Protocol.ProtocolBase ==
        TMoBaProtocolBase.mpbMotorola)
    {
        Busnr = 5;
    }
    else
    {
        Busnr = 6;
    }

    if (Busnr <= srcpClient.BusCount)
    {
        Bus = srcpClient.GetBusses(Busnr);
        if (null != Bus)
        {
            if (null != Bus.GAs)
            {
                if (null == Bus.GAs.GetSolenoids(fsol.Address))
                {
                    Bus.GAs.Add(FSol);
                }
                else
                {
                    //Befehl senden
                    Bus.GAs.Change(fsol);
                }
            }
        }
    }
}
}
}

```

Die Solenoid-Events:

```

client.OnAccessoireInitEvent += new
    TMoBaSRCPAccessoireDel(srcpClient_OnAccessoireInitEvent);
client.OnAccessoireTermEvent += new
    TMoBaSRCPAccessoireDel(srcpClient_OnAccessoireTermEvent);
client.OnAccessoireChangeEvent += new
    TMoBaSRCPAccessoireDel(srcpClient_OnAccessoireChangeEvent);
client.OnAccessoireLockEvent += new
    TMoBaSRCPAccessoireDel(srcpClient_OnAccessoireLockEvent);
client.OnAccessoireUnlockEvent += new
    TMoBaSRCPAccessoireDel(srcpClient_OnAccessoireUnlockEvent);

```

Code zum Starten der Zeitfunktion

```

TMoBaSRCPBus Bus = null;

if (null != client)
{
    if (client.Connected)
    {
        Bus = client.GetBusses(0);
    }
}

```

```

        if (null != Bus.Time)
        {
            Bus.Time.Multiplicator = 5;
            Bus.Time.Divisor = 1;
            Bus.Time.Date_Time = DateTime.Now;
            Bus.Time.Enabled = true;
        }
    }
}

```

Code zum Beenden der Zeitfunktion:

```

TMoBaSRCPBus Bus = null;
if (null != client)
{
    if (client.Connected)
    {
        Bus = client.GetBusses(0);
        if (null != Bus.Time)
        {
            Bus.Time.Enabled = false;
        }
    }
}

```

Die Zeit-Events:

```

client.OnTimeInitEvent += new
    TMoBaSRCPSimpleDel(srcpClient_OnTimeInitEvent);
client.OnTimeTermEvent += new
    TMoBaSRCPSimpleDel(srcpClient_OnTimeTermEvent);
client.OnTimeChangeEvent += new
    TMoBaSRCPSimpleDel(srcpClient_OnTimeChangeEvent);

```