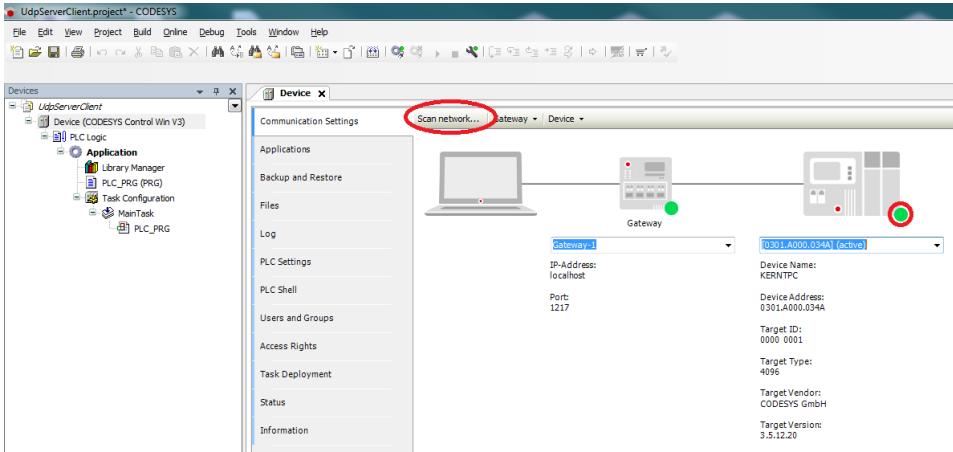


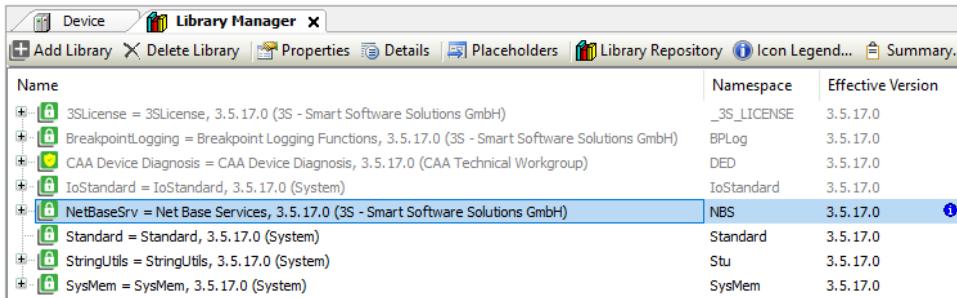
UDP: Example for Server and Client

- Create a "Standard project" and select *CODESYS Control Win V3* as the device.
- Define the target system by means of the *Network scan*.



As of SP17:

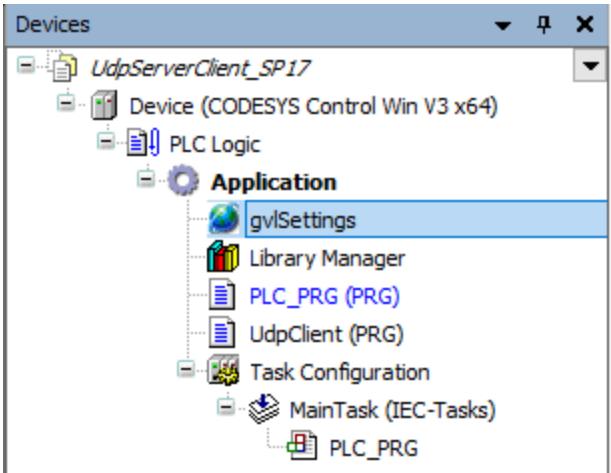
- Open the *Library Manager* and add the following libraries:
Net Base Services
StringUtils



- Create a global variable list named *gvlSettings* gvlSettings and define the following variables:

```
{attribute 'qualified_only'}
VAR_GLOBAL
    // !!! Make sure that the firewall does not block these ports !!!
    uiPort          : UINT          := 50000; // Port for the sender => receiver will set to
uiPort + 1
    sIPAddres      : STRING(19)     := '192.168.99.109';
END_VAR
```

- Add a POU to the project and name is *UdpClient*.



- Adapt the POU *UdpClient* as follows:
-

Declaration

```

VAR
    fbPeerClient      : NBS.UDP_Peer;
    ipAddress        : NBS.IPV4Address;
    xPeerActiv       : BOOL   := TRUE;

    fbSend           : NBS.UDP_Send;
    xSend            : BOOL;

    sSendMsg         : STRING(255) := 'Hello World';
END_VAR

```

Implementation

```

IF xPeerActiv AND NOT fbPeerClient.xBusy THEN
    ipAddress.SetInitialValue(ipAddress := gvlSettings.sIPAddres);
    fbPeerClient(xEnable := TRUE,
                 itfIPAddress := ipAddress,
                 uiPort := gvlSettings.uiPort + 1);
END_IF

fbPeerClient();

fbSend(xExecute := xSend AND fbPeerClient.xBusy,
       itfPeer := fbPeerClient,
       itfIPAddress := ipAddress,
       uiPort := gvlSettings.uiPort,
       pData := ADR(sSendMsg),
       udiSize := DINT_TO_UDINT(Stu.StrLenA(ADR
(sSendMsg))));

IF xSend THEN
    xSend := FALSE;
END_IF

```

- Adapt the POU **PLC_PRG** as follows:
-

Declaration

```

VAR
    fbPeerServer          : NBS.UDP_Peer;
    ipAddress            : NBS.IPV4Address;
    fbReceive             : NBS.UDP_Receive;
    xPeerActiv            : BOOL   := TRUE;

    xRead                : BOOL;
    abyReceive            : ARRAY [0..255] OF BYTE;

    sLastValidReceive     : STRING(255);
    udiIndex              : UDINT;
END_VAR

```

Implementation

```

IF xPeerActiv AND NOT fbPeerServer.xBusy THEN
    ipAddress.SetInitialValue(ipAddress := gvlSettings.sIPAddres);
    fbPeerServer(xEnable := TRUE,
                itfIPAddress := ipAddress,
                uiPort := gvlSettings.uiPort);
END_IF

fbPeerServer();

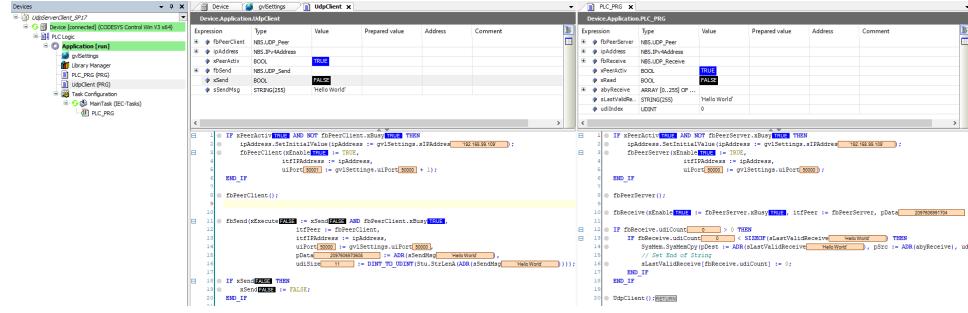
fbReceive(xEnable := fbPeerServer.xBusy, itfPeer := fbPeerServer, pData := ADR(abyReceive),
udiSize := SIZEOF(abyReceive));

IF fbReceive.udiCount > 0 THEN
    IF fbReceive.udiCount < SIZEOF(sLastValidReceive) THEN
        SysMem.SysMemCpy(pDest := ADR(sLastValidReceive), pSrc := ADR(abyReceive),
udiCount := fbReceive.udiCount);
        // Set End of String
        sLastValidReceive[fbReceive.udiCount] := 0;
    END_IF
END_IF

UdpClient();

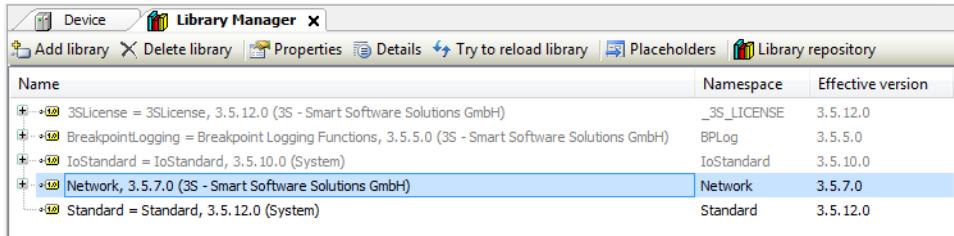
```

-
- Start the project and set the **xSend** variable to **TRUE**.



Up to SP16:

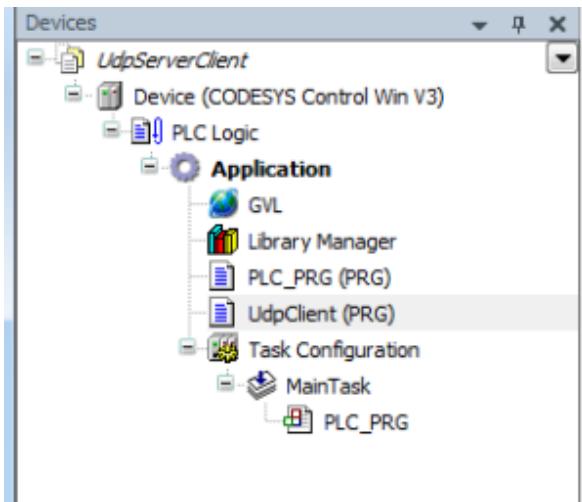
- Open the *Library Manager* and add the *Network* library.



- Add a global variable list to the project and define the following variables:

```
{attribute 'qualified_only'}
VAR_GLOBAL
    sIpAddr      :      NBS.IP_ADDR := (sAddr := '192.168.99.74'); // Change to own IP-Address
    // !!! Make sure that the firewall does not block these ports !!!
    uiPort       :      UINT := 8181; // Port for the sender => receiver will set to uiPort + 1
END_VAR
```

- Add a POU to the project and name is *UdpClient*.



- Adapt the POU *UdpClient* as follows:

Declaration

```

VAR
    fbPeerClient      :   NBS.UDP_Peer;
    fbReceive         :   NBS.UDP_Receive;
    sRcvMsg          :   STRING;
END_VAR

```

Implementation

```

fbPeerClient(xEnable := TRUE, ipAddr:= GVL.stIpAddr, uiPort:= GVL.uiPort + 1);

fbReceive(xEnable := (fbPeerClient.hPeer <> CAA.gc_hINVALID),
           hPeer := fbPeerClient.hPeer,
           szSize := SIZEOF(sRcvMsg),
           pData := ADR(sRcvMsg));

```

- Adapt the POU **PLC_PRG** as follows:

Declaration

```

VAR
    fbPeerServer      :   NBS.UDP_Peer;
    fbSend            :   NBS.UDP_Send;
    xSend             :   BOOL;
    sSendMsg          :   STRING := 'Hello World';
END_VAR

```

Implementation

```

UdpClient();

fbPeerServer(xEnable := TRUE, ipAddr := GVL.stIpAddr, uiPort := GVL.uiPort);

IF fbPeerServer.hPeer <> CAA.gc_hINVALID THEN
    fbSend(xExecute := xSend,
           hPeer := fbPeerServer.hPeer,
           ipAddr := GVL.stIpAddr,
           uiPort := GVL.uiPort + 1,
           szSize := SIZEOF(sSendMsg),
           pData := ADR(sSendMsg));
END_IF

IF xSend THEN
    xSend := FALSE;
END_IF

```

- Start the project and set the `xSend` variable to `TRUE`.

