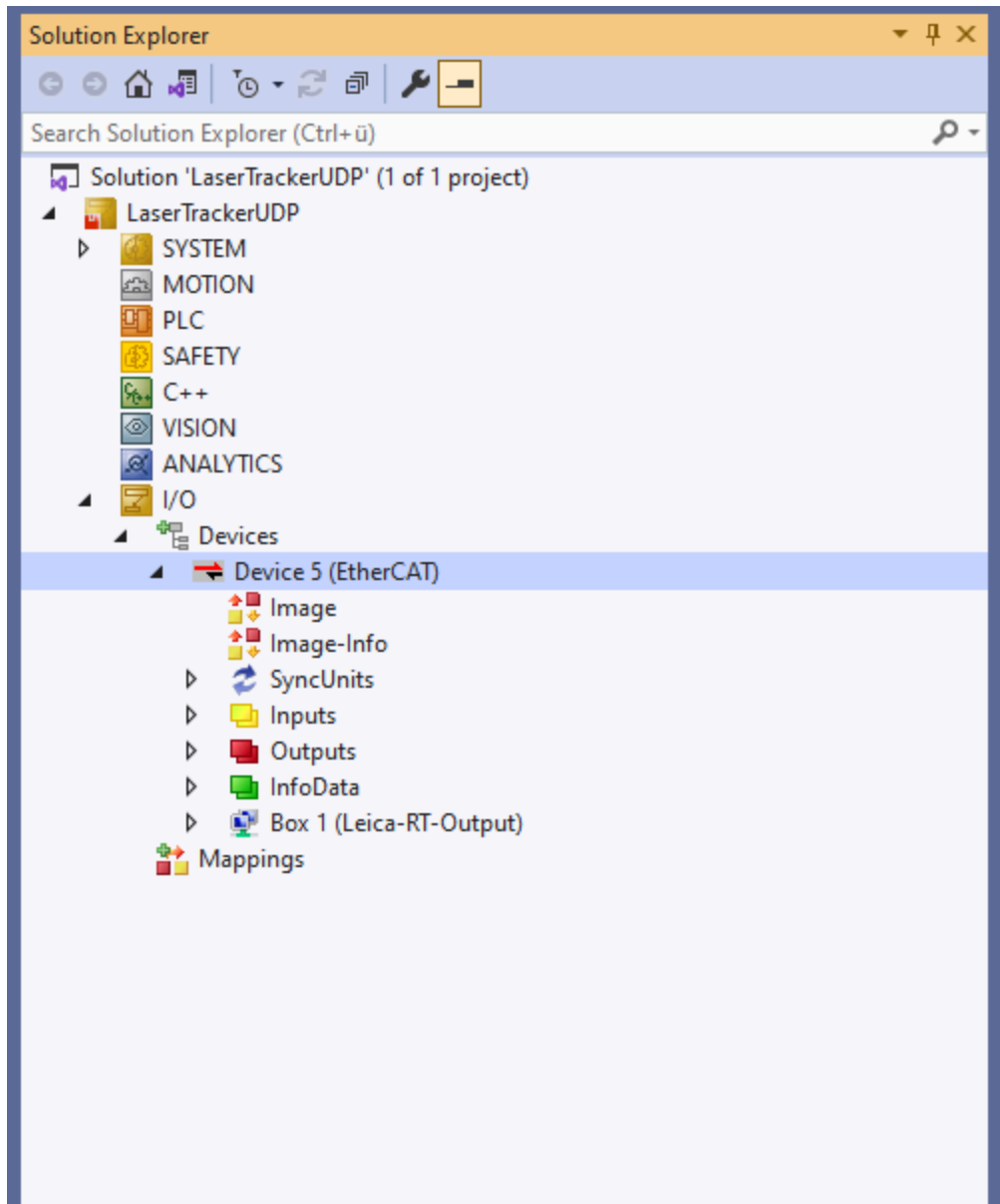


LaserTracker UDP with TwinCAT

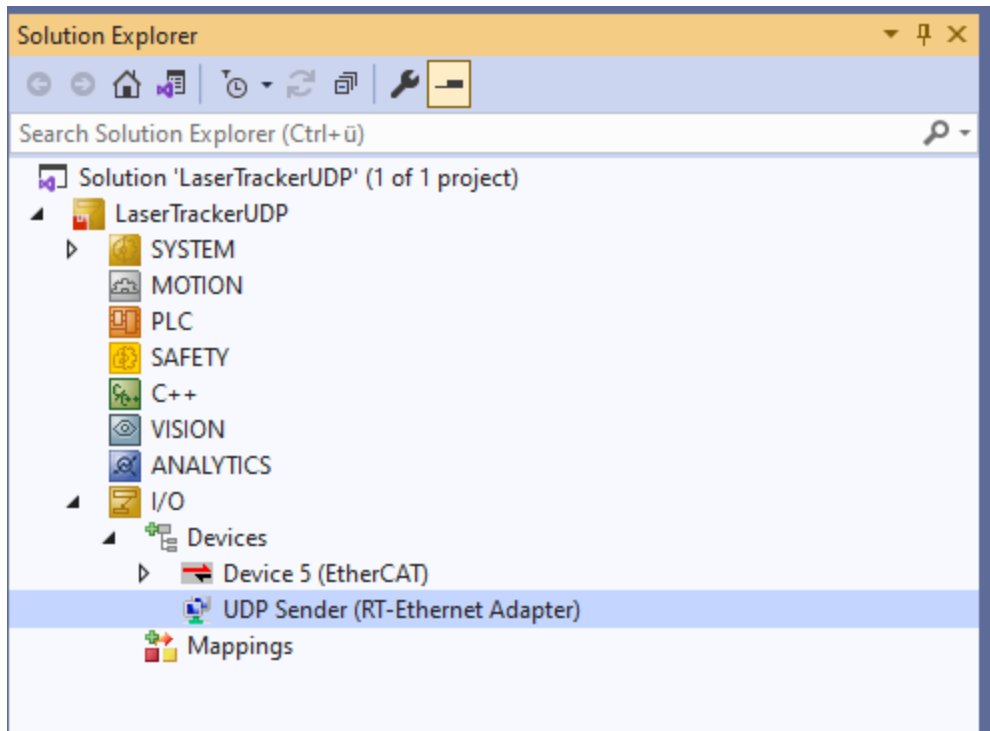
Author: Gerald Ebmer, Minh Nhat Vu

§§ DETAIL STEPS for UDP sender from TwinCAT,

- 1) Create an empty XAE project and scan I/O, the EtherCAT devices will be shown up if it is already connected to the PC (serves a EtherCAT master now)



2) Create an RT-Ethernet Adapter, we can free choose the name, e.g,



- 3) Follow steps here to create a cyclic IO project:
https://infosys.beckhoff.com/english.php?content=../content/1033/tf6105_tc3_opc_ua_pub_sub/16517819531.html&id=
- 4) Adding some codes in the main header file, and also the sending information at the main source file

```

// TODO: Custom variable
UINT m_counter;
ULONG client_ipaddr;
USHORT m_udpPort;
LaserTrackerMeasurement meas_data;

```

Also created struct for meas_data

```

class LaserTrackerMeasurement {
public:
    double angle_hz, angle_vt, distance;
    double quaternion[4];
    LaserTrackerMeasurement() {
        angle_hz = 0.;
        angle_vt = 0.;
        distance = 0.;
        for (int i = 0; i < 4; i++) {
            quaternion[i] = 0.;
        }
        quaternion[0] = 1.;
    }

    void set_position(double dist, double hz, double vt) {
        distance = dist;
        angle_hz = hz;
        angle_vt = vt;
    }

    void set_orientation(double q0, double q1, double q2, double q3) {
        quaternion[0] = q0;
        quaternion[1] = q1;
        quaternion[2] = q2;
        quaternion[3] = q3;
    }
};

///<AutoGeneratedContent id="ImplementationOf_ITcCyclic">
HRESULT CUDP_sender::CycleUpdate(ITcTask* ipTask, ITcUnknown* ipCaller, ULONG_PTR context)
{
    HRESULT hr = S_OK;

    // TODO: Replace the sample with your cyclic code
    //m_counter+=m_Inputs.Value;
    //m_Outputs.Value=m_counter;

    //m_spUdpProt->CheckReceived(); // ADDED
    meas_data.set_position(m_Inputs.Distance, m_Inputs.AngleHz, m_Inputs.AngleVt);
    meas_data.set_orientation(m_Inputs.Quaternion0, m_Inputs.Quaternion1, m_Inputs.Quaternion2, m_Inputs.Quaternion3);

    HRESULT hr2 = S_OK;
    PVOID pData = &meas_data;
    ULONG nData = sizeof(LaserTrackerMeasurement);

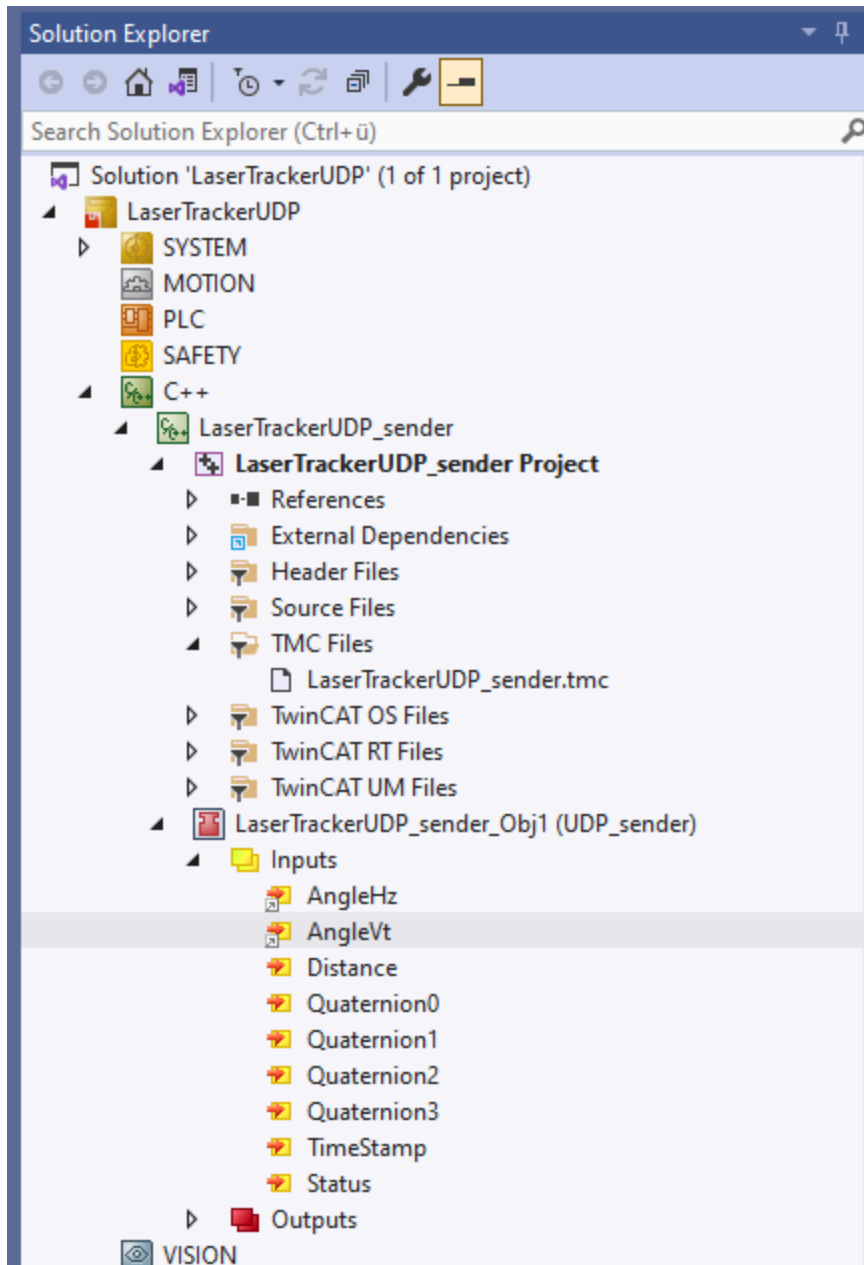
    hr2 = m_spUdpProt->SendData(client_ipaddr, m_udpPort, m_udpPort, nData, pData, true);

    m_Trace.Log(tlInfo, FLEAVEA "send data status hr=0x%08x", hr2);

    return hr;
}
///</AutoGeneratedContent>

```

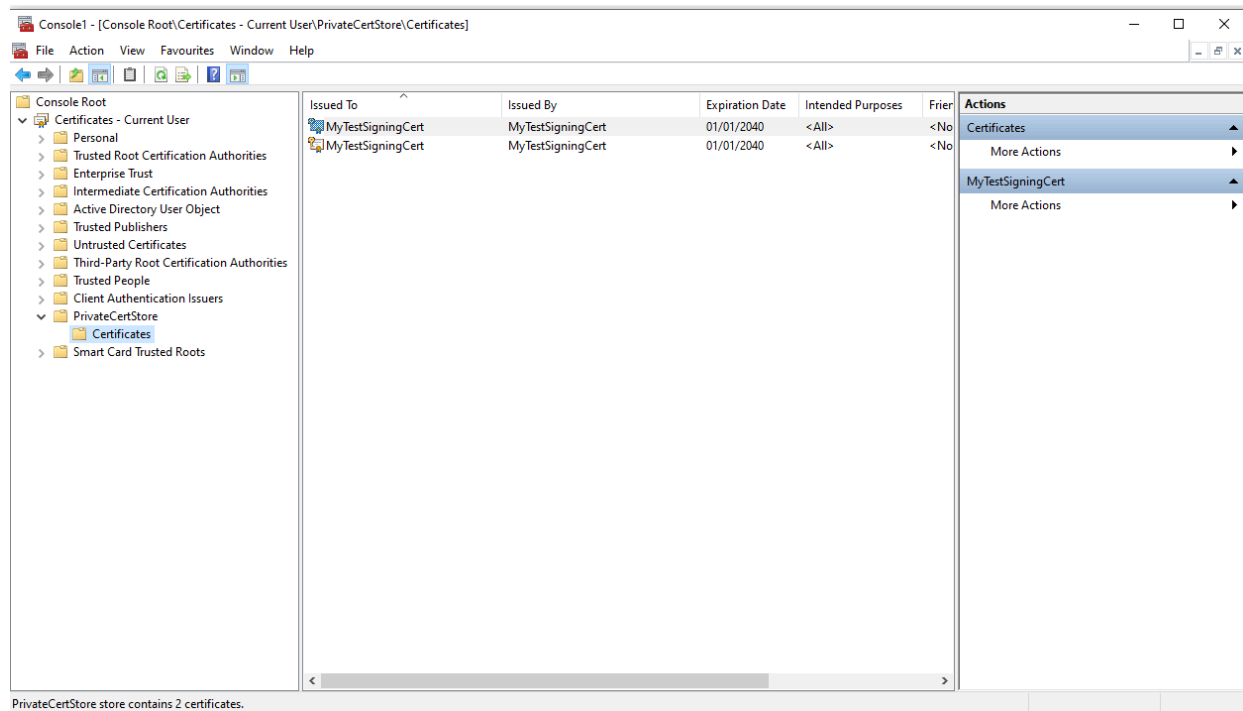
5) Build the project, then add module, and finally link all of the variables.



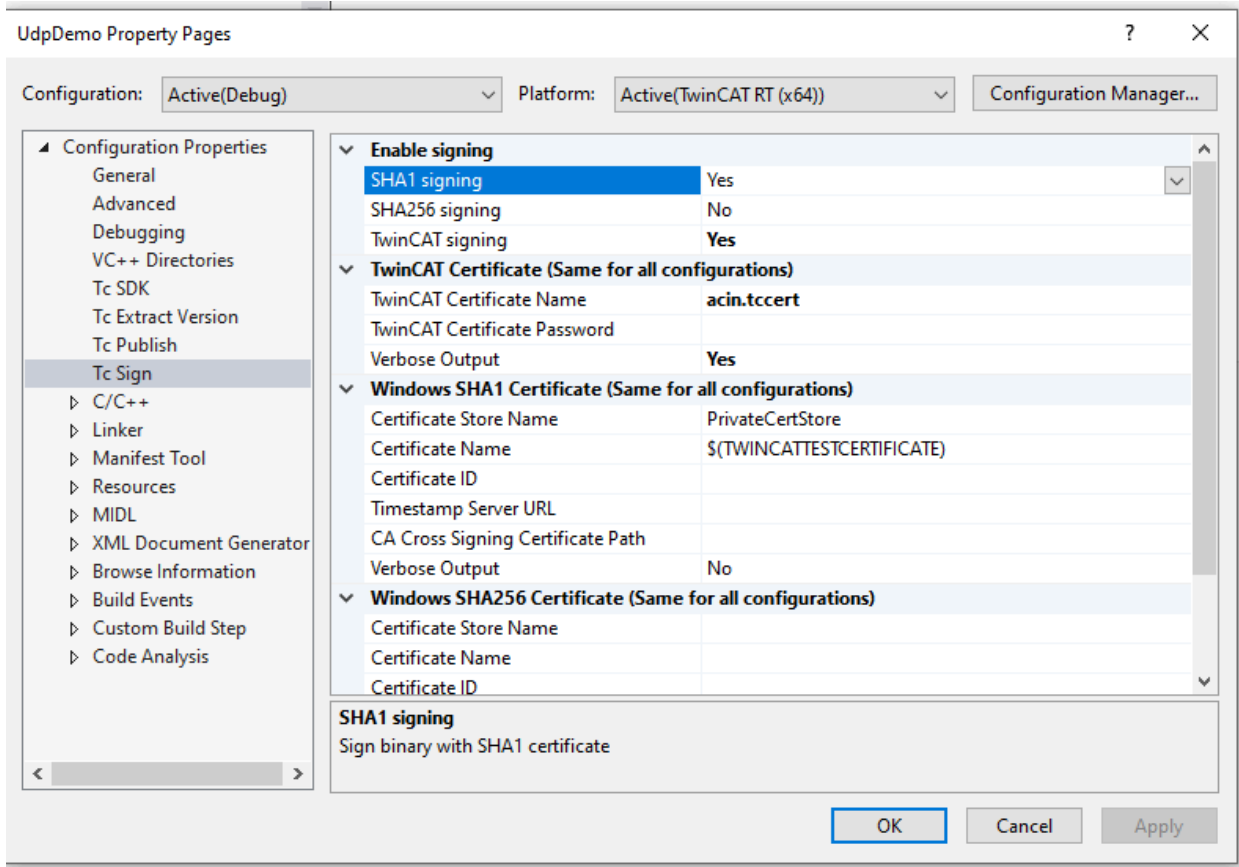
6)

Appendix,

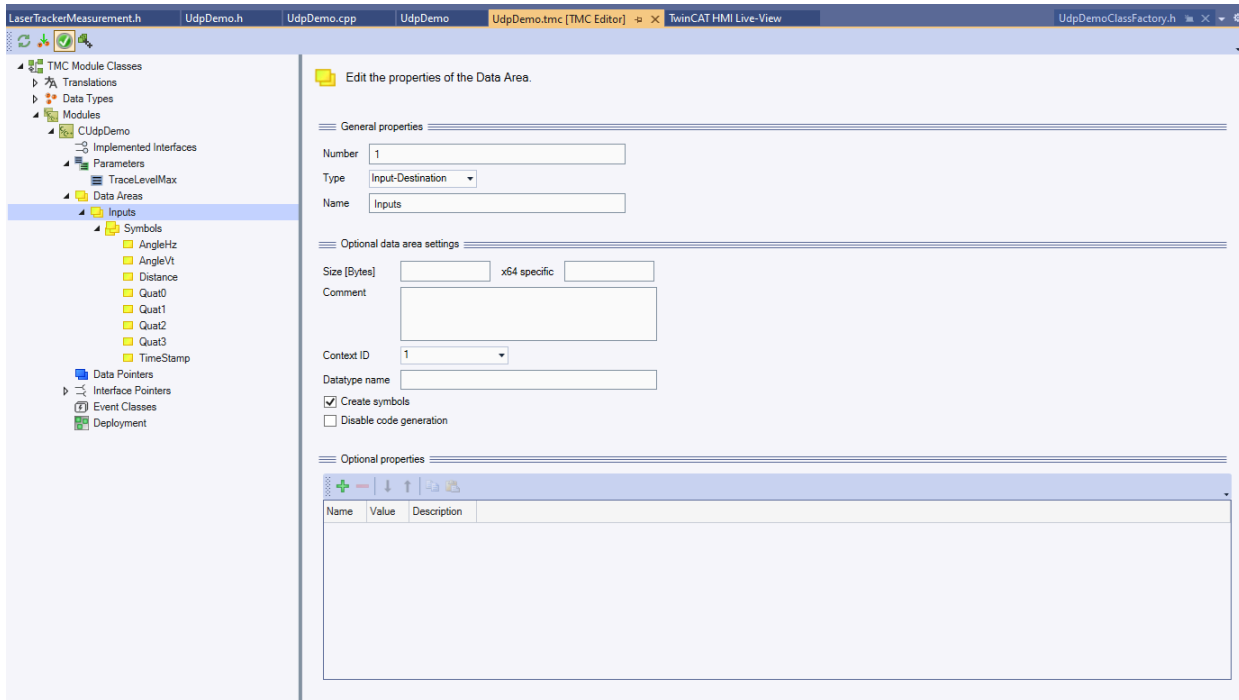
1, Check if you have the correct certificates. Action: Windows + R, “mmc” -> add/remove snap in



2, Make sure Tc Sign is correct (right click on C++ project and see Properties)



3, Setting input-destination



Add license

Order Information (Runtime)
Manage Licenses
Project Licenses
Online Licenses

License Device: Target (Hardware Id) Add...

System Id: E542868F-75DC-EA78-9D3A-8F21957612FC Platform: other (90)

License Request

Provider: Beckhoff Automation Generate File...

License Id: Customer Id:

Comment:

License Activation

7 Days Trial License... License Response File...

Order No	License	Instances	License TAN	Current Status
TC1300	TC3 C++	cpu license		expires on Oct 11, 2024 (...)
TF6311	TC3 TCP/UDP RT	cpu license		expires on Oct 11, 2024 (...)

TwinCAT: Common Error ADS

ADS Error 1792: TcSign nicht richtig konfiguriert
ADS Error 1821 TcCom instance - no context (task) assigned
ADS Error 1836 ? - interface pointer not set (e.g. post cyclic interface pointer in kuka interface module)
ADS Error 1823 Select right network interface (RT-Ethernet Adapter)
ADS Error 1803 data pointer to non existing data area (e.g. disabled tc com module instance where data pointer of other module instance is accessing)