1.For PC Com Port Communication Check Com Port Number for PC using Hercules Software

2.First Install TF6340 in Target & XAE system

3.Confirm the com port number (must be Com 1,2,3,4) others are not supported

4.Add serial port in I/O

Graphical user interface, application

Description automatically generated

Graphical user interface, text, application, chat or text message

Description automatically generated

5.Port should be same as step 1

Graphical user interface

Description automatically generated

6.Com Port Mode setting for pc, make sure receiver & Device manager com setting are same as setting in twincat

Graphical user interface, application

Description automatically generated

8.Create 2 Task one for Standard Program & another one for communication make task time is less usually 2ms (Refer manual for further information) & add reference TC2\_SerialCom

A screenshot of a computer

Description automatically generated with medium confidenceText

Description automatically generated

9.Create an BackgroundCommProgram(PRG) call it in CommTask

10.In Back Ground Program Add following Variable

(\* I/O variables for a PC-COM port \*)

stIn\_PcCom AT %I\* : PcComInData; (\* linked to the port in the TwinCAT System Manager \*)

stOut\_PcCom AT %Q\* : PcComOutData; (\* linked to the port in the TwinCAT System Manager \*)

(\* I/O variables for a EL6001 terminal\*)

stIn\_EL6001 AT %I\* : EL6inData22B; (\* linked to the EL6001 in the TwinCAT System Manager \*)

stOut\_EL6001 AT %Q\* : EL6outData22B;(\* linked to the EL6001 in the TwinCAT System Manager \*)

(\* I/O variables for a KL6001 terminal\*)

stIn\_KL6001 AT %I\* : KL6inData; (\* linked to the KL6001 in the TwinCAT System Manager \*)

stOut\_KL6001 AT %Q\* : KL6outData; (\* linked to the KL6001 in the TwinCAT System Manager \*)

11.Add Global Variable for Buffer

(\* Buffer Variable \*)

RxBufferPcCom : ComBuffer; (\* Receive data buffer; used with all receive function blocks \*)

TxBufferPcCom : ComBuffer; (\* Transmit data buffer; used with all receive function blocks \*)

12.Link Background I/O Variable to Hardware

Graphical user interface, text

Description automatically generated

13.Go to com device click on input, select data1 to 64 Right click change multilink & link the background created variable

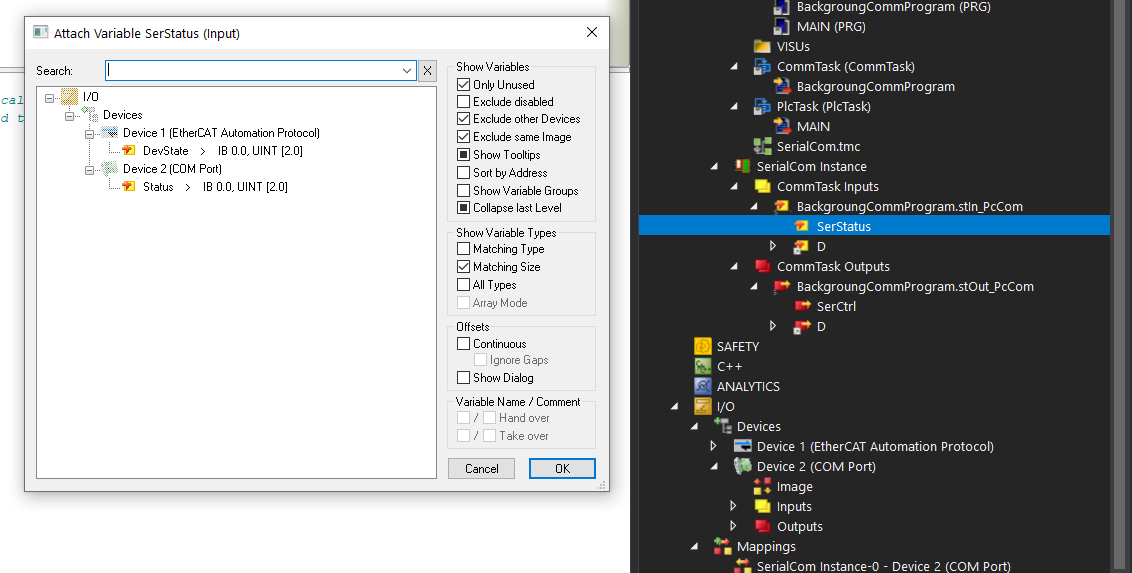
Graphical user interface, application

Description automatically generated

Similar to output also

14. Go to Instance, Task Inputs servStatus to com input device, similar for output

15.Also add serStatus & SerCtrl



16.Add following Program in Background Program

fbPcComCtrl(

Mode:= SERIALLINEMODE\_PC\_COM\_PORT,

pComIn:= ADR(stIn\_PcCom),

pComOut:= ADR(stOut\_PcCom),

SizeComIn:= UINT\_TO\_INT(SIZEOF(stIn\_PcCom)),

Error=> ,

ErrorID=> ,

TxBuffer:= SerialComm.TxBufferPcCom,

RxBuffer:= SerialComm.RxBufferPcCom);

IF fbPcComCtrl.Error THEN

bPcComCtrlError := TRUE;

ePcComCtrlErrorID := fbPcComCtrl.ErrorID;

END\_IF

17.In Main Program

Variable

fbSerialCommInstance1 : SerialCom;

sRecieveString : STRING;

bRecieveError : BOOL;

bSendBusy: BOOL;

eSendError: INT;

sSendData : STRING;

bsendTrigger: BOOL;

STxBuffer: INT;

tSendTime: INT;

sizeoff : UINT;

bRecieveReset: BOOL;

(\* Send String Data \*)

fbSerialCommInstance1(bSendData:= bsendTrigger,

bReset := bRecieveReset,

TxBuffer := SerialComm.TxBufferPcCom,

sSendPrefix:='$01',

sSendSuffix:='$02',

sSendString:=sSendData,

bSendBusy =>bSendBusy,

eSendError =>eSendError,

RxBuffer := SerialComm.RxBufferPcCom,

sReceiveString =>sRecieveString,

sReceivePrefix:='$01',

sReceiveSuffix:='$02',

tReceiveTimeOut := T#1S,

bReset :=bRecieveError,

bReceiveStringData =>,

bReceiveBusy =>,

eReceiveError =>,

bReceiveTimeOut =>

);

18.Then in Main Send sting with sSendData Input variable & once we recieved an string will be displayed in sRecieveString

19 Github Link https://github.com/gokulgbeckhoff/SerialCommunication