EtherCAT DeviceDesigner + Microchip LAN9252 UART Example

Table of Contents

1 Introduction........3
2 Prerequisites........3
3 Step – by – Step........3
    3.1 EtherCAT DeviceDesigner........4
    3.2 MPLAB X........6
    3.3 Program the EEPROM........7
4 Using the example........8
    4.1 Description........8
    4.2 Connecting UART........8
    4.3 Using the example........9

1 Introduction

This document contains a description how to get the LAN9252 UART example easily working with the emtas EtherCAT DeviceDesigner and MPLAB X.

2 Prerequisites

The following tools and files are needed to get this UART example working.

- MPLAB X IDE
  (http://www.microchip.com/pagehandler/en-us/family/mplabx/)

- MPLAB XC Compiler
  (http://www.microchip.com/pagehandler/en-us/devtools/mplabxc/home.html)

- SSC v5.10 or higher from EtherCAT.org
  You need to be a member of the ETG to be allowed to download and use the SSC.
  You do not have to extract the SSC, just remember where the downloaded .zip file is saved.

- EtherCAT DeviceDesigner
  (Included in this archive or available at http://www.emtas.de/download/ethercat-devicedesigner)

3 Step – by – Step

The following guide expects that you have downloaded and installed the MPLAB X IDE, the MPLAB XC Compiler and the EtherCAT DeviceDesigner. Furthermore it is necessary that the SSC was downloaded.

3.1 EtherCAT DeviceDesigner

1. To generate all necessary files just launch the EtherCAT DeviceDesigner and open the bundled project “LAN9252_UART.eddp” from the sub folder “EDD”.

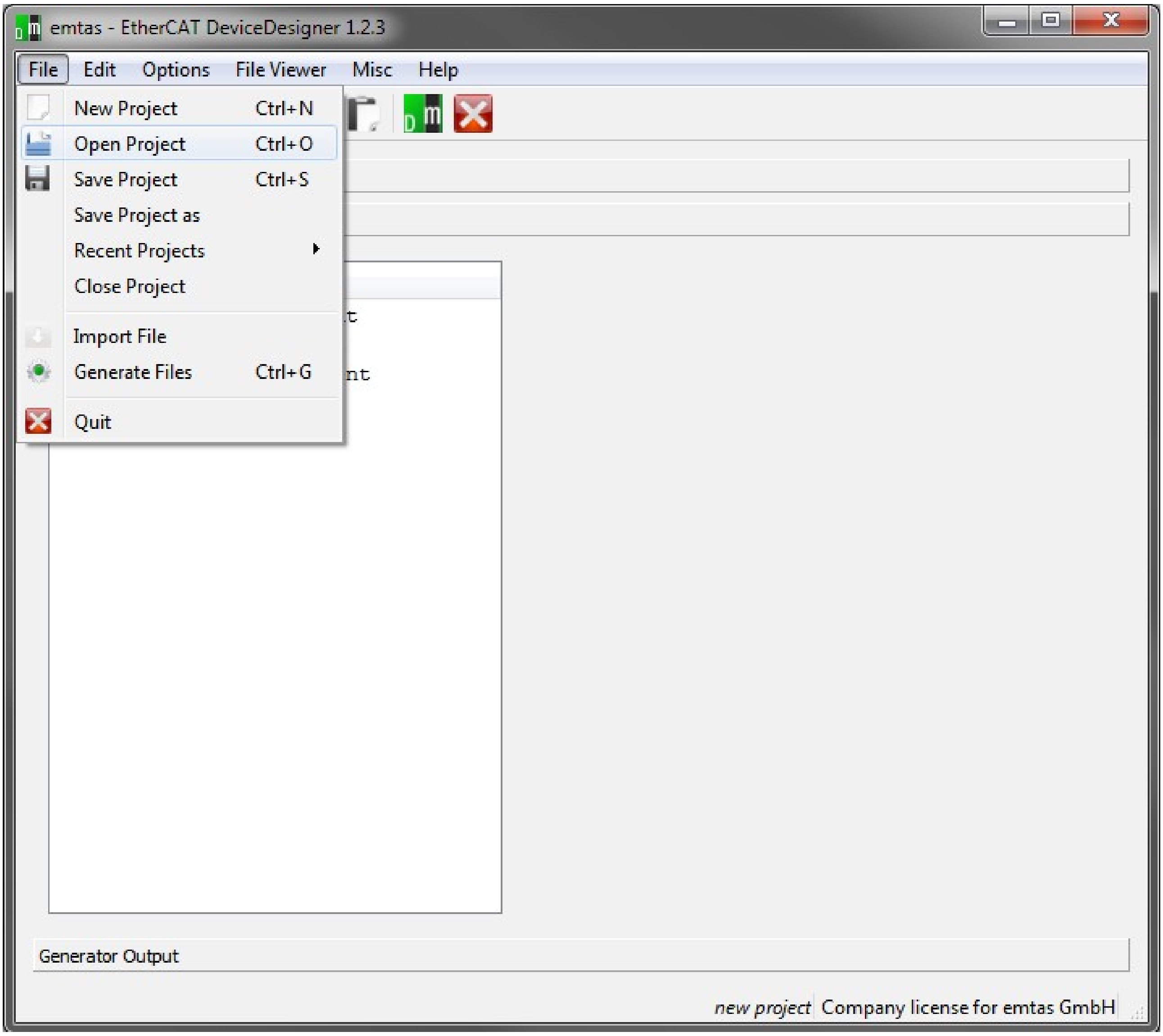
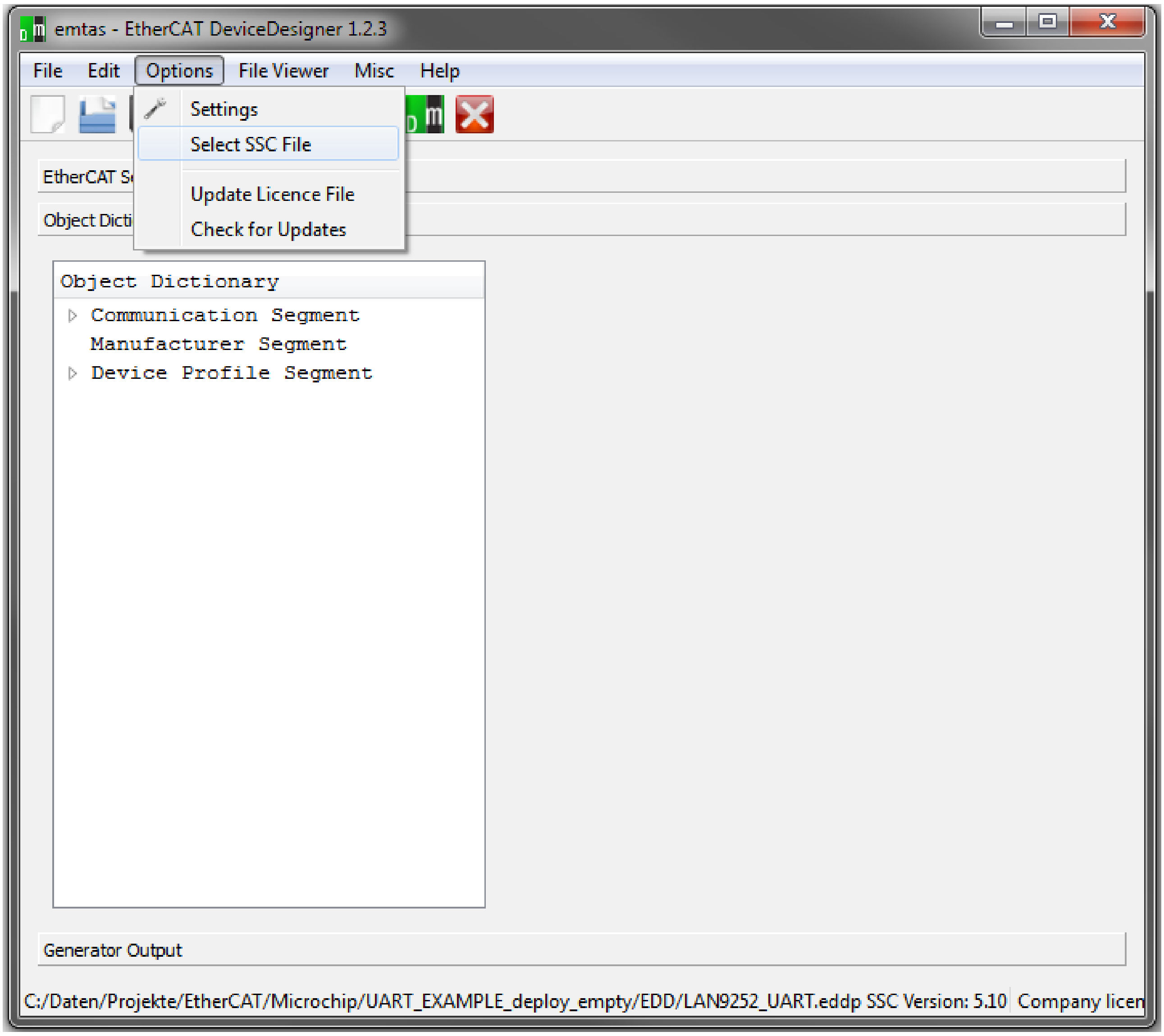


Figure 1: EtherCAT DeviceDesigner Open existing project

2. Select the downloaded SSC zip file via "Options -> Select SCC File" as shown in "Figure 2". Just point to the .zip file.



C:/Daten/Projekte/EtherCAT/Microchip/UART_EXAMPLE_deploy_empty/EDD/LAN9252_UART.eddp SSC Version: 5.10 Company licen

Figure 2: Select the SSC zip file

3. Generate all necessary files by clicking the "Generate files" button.

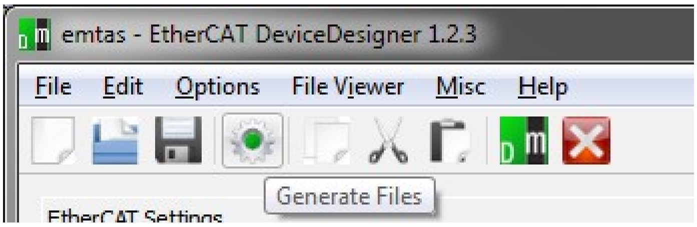
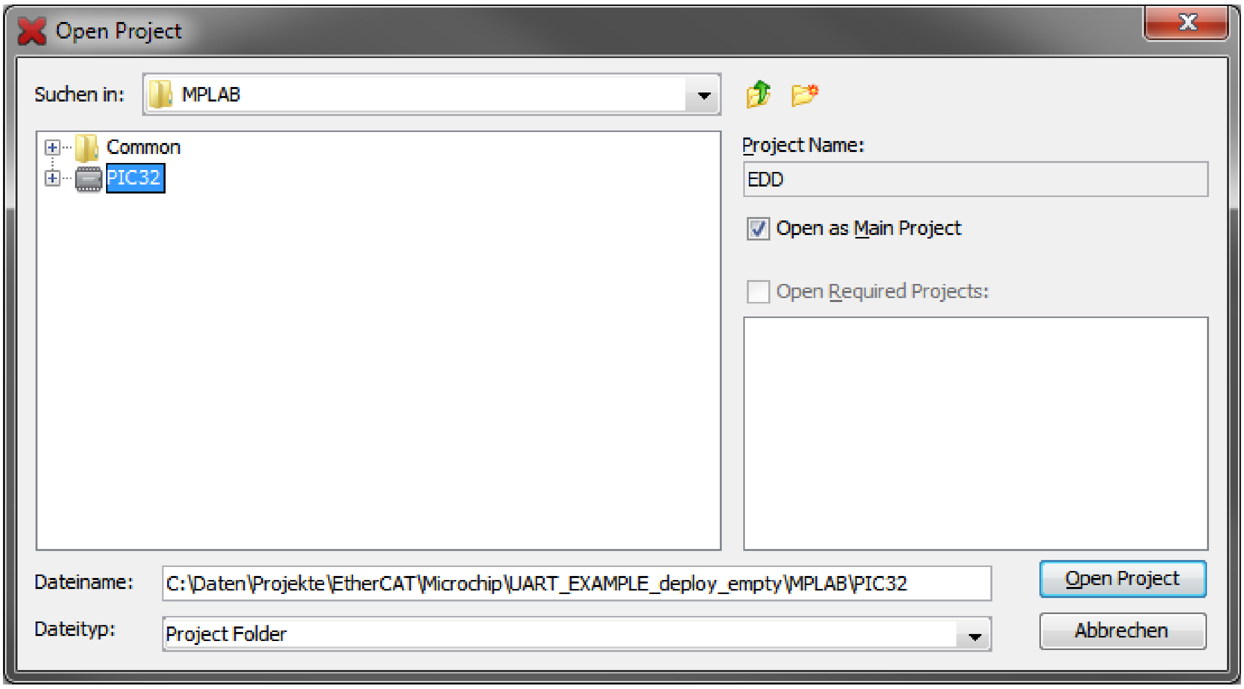


Figure 3: Generate Files

3.2 MPLAB X

1. Start MPLAB X and open the project in the “MPLAB” folder.



Suchen in: MPLAB

Project Name: EDD

☑ Open as Main Project

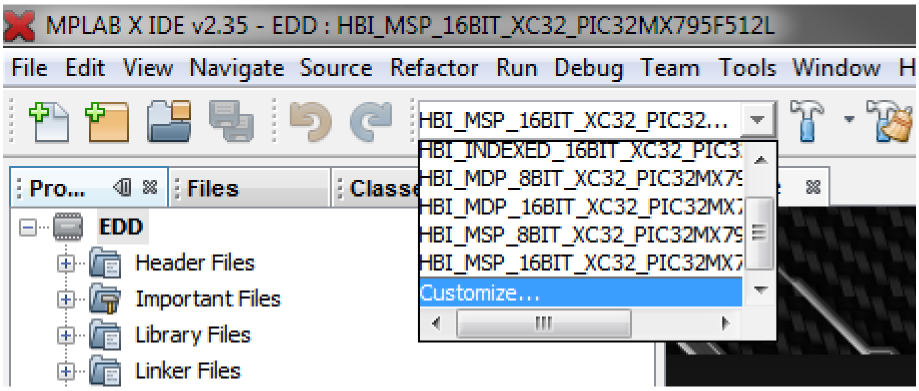
☐ Open Required Projects:

Dateiname: C:\Daten\Projekte\EtherCAT\Microchip\UART_EXAMPLE_deploy_empty\MPLAB\PIC32

Dateityp: Project Folder

Figure 4: MPLAB X - Open Project

2. Select your programming hardware via “Customize” in the Configurations Combo Box, see “Figure 5”.



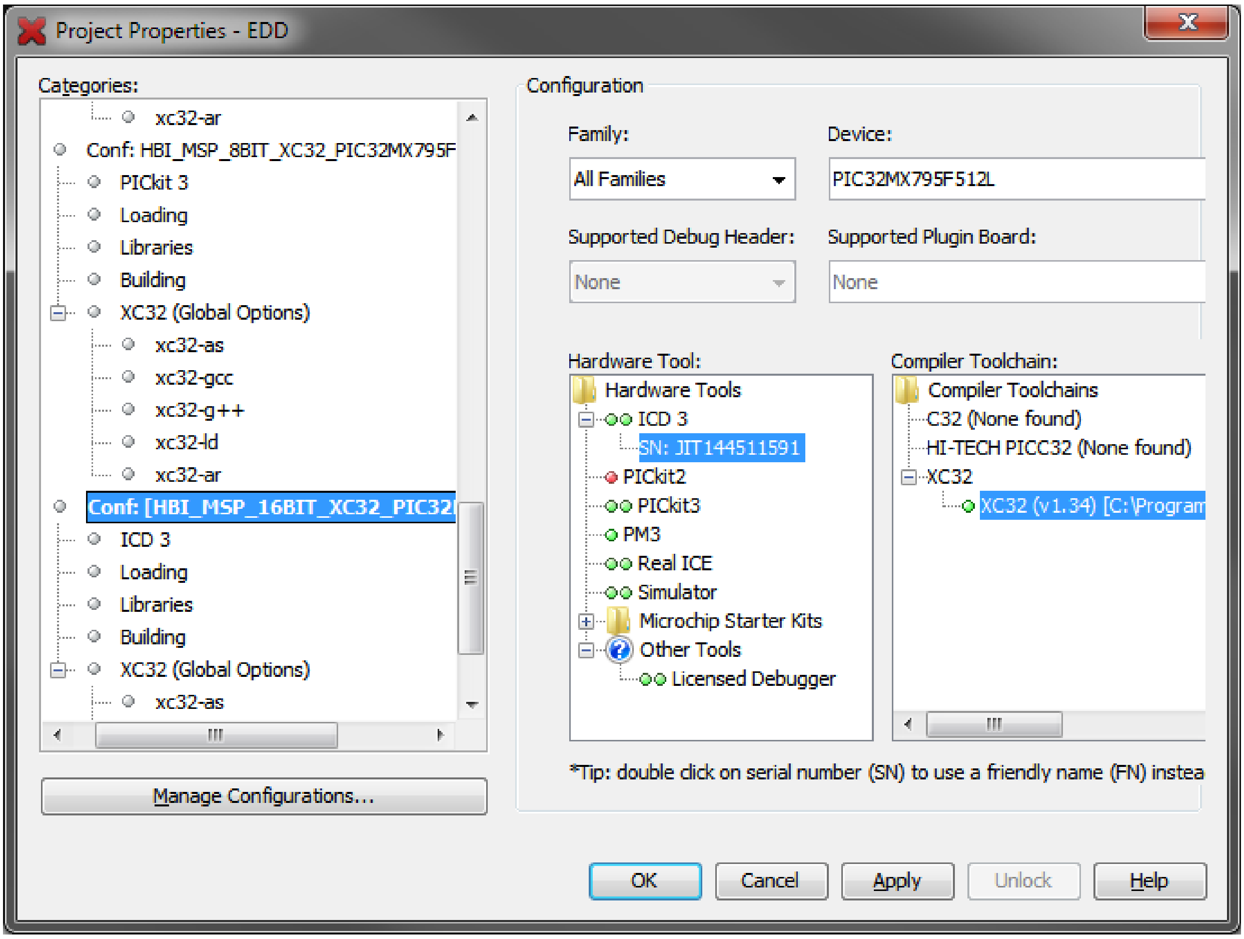
MPLAB X IDE v2.35 - EDD : HBI_MSP_16BIT_XC32_PIC32MX795F512L

HBI_MSP_16BIT_XC32_PIC32...

HBI_INDEXED_16BIT_XC32_PIC3
HBI_MDP_8BIT_XC32_PIC32MX79
HBI_MDP_16BIT_XC32_PIC32MX7
HBI_MSP_8BIT_XC32_PIC32MX79
HBI_MSP_16BIT_XC32_PIC32MX7
Customize...

Figure 5: MPLAB X - Customize

3. The next dialog lets you choose your hardware.



Family: All Families

Device: PIC32MX795F512L

Supported Debug Header: None

Supported Plugin Board: None

*Tip: double click on serial number (SN) to use a friendly name (FN) instea

Figure 6: Selecting Hardware

4. Build the project and program the device.

3.3 Program the EEPROM

It is necessary to program the EEPROM of your LAN9252. You can do this by using the „EEPROM Programmer“, „TwinCAT“ or the „EtherCAT DeviceExplorer“ by emtas.

The necessary files are generated by the EtherCAT DeviceDesigner. If you use the „EEPROM Programmer“ or TwinCAT you can use the LAN9252_UART.xml file. You can find this file at „\EDD\esi“.

If you use the „EtherCAT DeviceExplorer“ you can directly use the EEPROM file at “\EDD\eeprom\eeprom.bin”.

4 Using the example

4.1 Description

This example demonstrates how EtherCAT can interact simply with the MCU. Furthermore it shows some basic functions of the EtherCAT framework by emtas. After connecting the EVB2-LAN9252 board via UART with a PC you can send and receive values with a terminal program.

4.2 Connecting UART

The example uses the UART3 pins of the PIC32 on the EVB-LAN9252 board. You can use a simple USB to serial cable to connect your PC to the board.

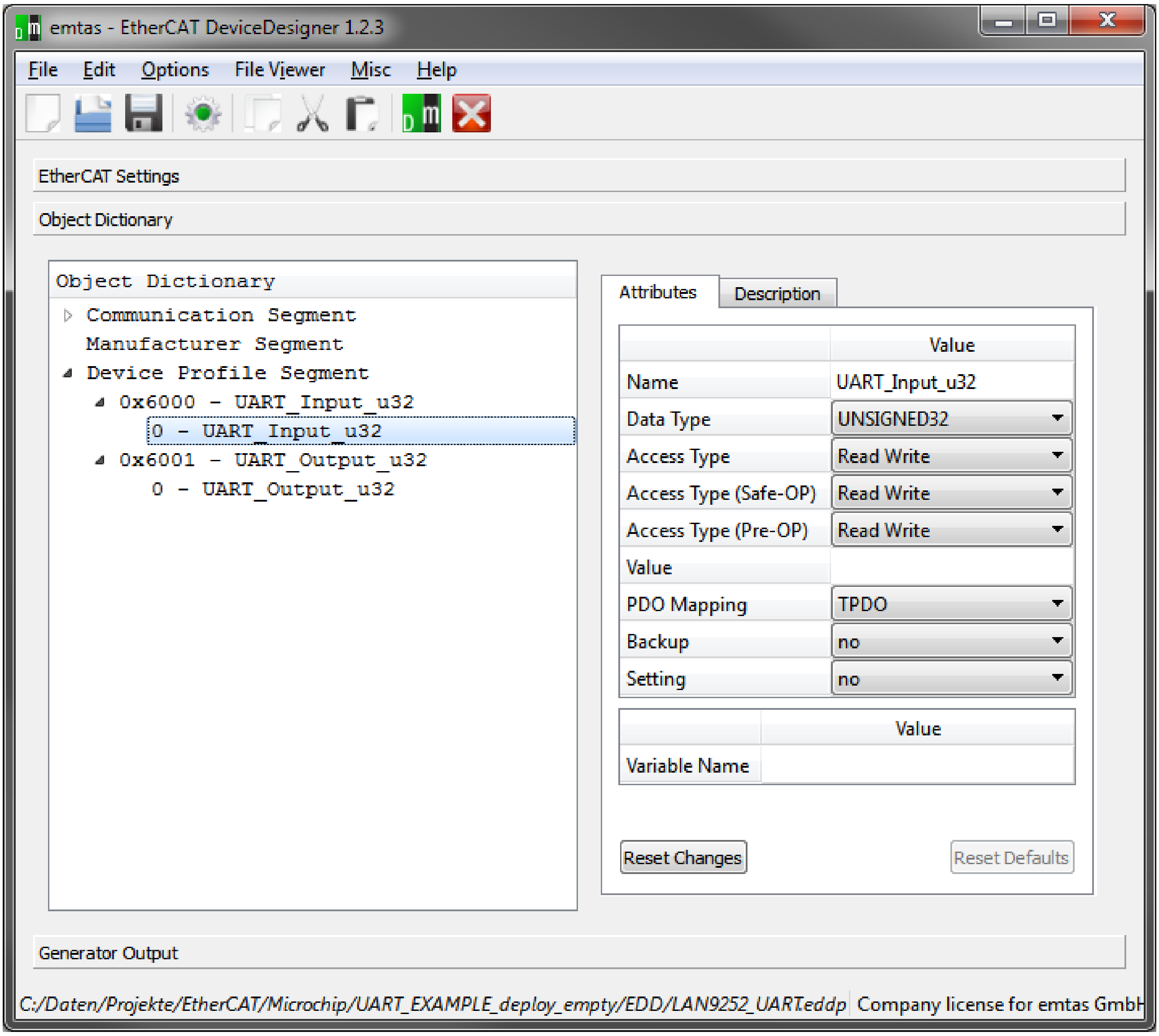
The pins of UART3 are accessible at header P1. pin 44 is the UART TX and pin 41 is the UART RX.

After connecting your PC with the board use a simple terminal programm like „HyperTerminal“, „HTerm“ or „Putty“.

The used settings are:

- 9600 baudrate
- 8 Databits
- 1 Stopbit
- No parity, no flow control

4.3 Using the example



Object Dictionary

- Communication Segment
- Manufacturer Segment
- Device Profile Segment
  - 0x6000 – UART_Input_u32
    - 0 – UART_Input_u32
  - 0x6001 – UART_Output_u32
    - 0 – UART_Output_u32

	Value
Name	UART_Input_u32
Data Type	UNSIGNED32
Access Type	Read Write
Access Type (Safe-OP)	Read Write
Access Type (Pre-OP)	Read Write
Value	
PDO Mapping	TPDO
Backup	no
Setting	no

	Value
Variable Name	

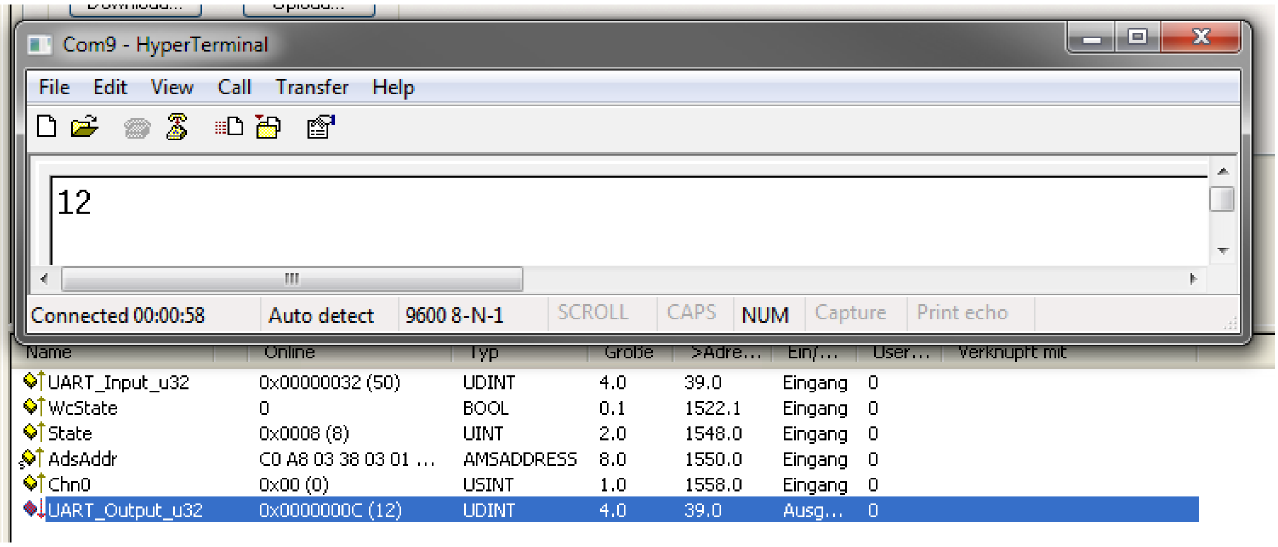
C:/Daten/Projekte/EtherCAT/Microchip/UART_EXAMPLE_deploy_empty/EDD/LAN9252_UART.eddp Company license for emtas GmbH

Figure 7: Input- and Output variables

As you can see in the EtherCAT DeviceDesigner there exists two unsigned32 variables which are mapped as PDO. “UART_Input_u32” as transmit-PDO and “UART_Output_u32” as receive-PDO.

Values received via EtherCAT are mapped to “UART_Output_u32”. In this case the EtherCAT framework function “pdoReceivceInd()” in main.c is called. This function writes the received value as string via UART.

If a value is entered in a terminal program the interrupt service routine “IntUart3Handler()” in serial.c is triggered. This function add the available byte to a buffer. If the byte is a line feed the received value is mapped to the variable “UART_Input_u32” and so transmitted via PDO.



```
12
```

Name	Online	Typ	Größe	>Adre...	Ein/...	User...	Verknüpft mit
UART_Input_u32	0x00000032 (50)	UDINT	4.0	39.0	Eingang	0	
WcState	0	BOOL	0.1	1522.1	Eingang	0	
State	0x0008 (8)	UINT	2.0	1548.0	Eingang	0	
AdsAddr	C0 A8 03 38 03 01 ...	AMSADDRESS	8.0	1550.0	Eingang	0	
Chn0	0x00 (0)	USINT	1.0	1558.0	Eingang	0	
UART_Output_u32	0x0000000C (12)	UDINT	4.0	39.0	Ausg...	0	

Figure 8: HyperTerminal Example

In „Figure 7“ you can see a screenshot containing the „HyperTerminal“ window and a part of TwinCAT.

As you can see I changed the value of „UART_Output_u32“ in TwinCAT to 12 and this value is outputted in „HyperTerminal“. After typing 50 and ENTER in „HyperTerminal“ this value is read via UART and transmitted as „UART_Input_u32“.