

1) Import the TestQEP Project into Code Composer Studio

2) you can edit the QEP\_ERROR\_ON define in globals.h

3) connect the QEP Hardware Simulator to the QEP Pins (A,B,Strobe,Index,Vcc and GND --> you can use the Board 5V supply) and the RS232 Port of your Computer (baudrate is 19200)



4) set first QEP state to state 3 A=1,B=1,I=0,S=1

```
sd - HyperTerminal
Datei Bearbeiten Ansicht Anrufen Übertragung ?
5:A=0 B=0 I=0 S=1
6:A=1 B=0 I=0 S=0
7:A=1 B=1 I=0 S=0
8:A=0 B=1 I=1 S=0
9:A=0 B=0 I=1 S=0
10:A=1 B=0 I=1 S=0
11:A=1 B=1 I=1 S=0
0:A=0 B=1 I=0 S=0
1:A=0 B=0 I=0 S=0
2:A=1 B=0 I=0 S=1
3:A=1 B=1 I=0 S=1
4:A=0 B=1 I=0 S=1
5:A=0 B=0 I=0 S=1
6:A=1 B=0 I=0 S=0
7:A=1 B=1 I=0 S=0
8:A=0 B=1 I=1 S=0
9:A=0 B=0 I=1 S=0
10:A=1 B=0 I=1 S=0
11:A=1 B=1 I=1 S=0
0:A=0 B=1 I=0 S=0
1:A=0 B=0 I=0 S=0
2:A=1 B=0 I=0 S=1
3:A=1 B=1 I=0 S=1
-
Verbunden 00:01:26 Auto-Erkenn. 19200 8-N-1 RF GROSS NUM Aufzeichnen Druckerecho
```

5) Start and debug the Project

6)the QEP Class get the Intestate with the instruction `InitialState = (GpioDataRegs.GPADAT.all & 0x00f00000)>> 20;` - with the `QEP_ERROR_ON` define the Mux is set to read from GPIO and without the define the port is read with default Mux settings

7)uncomment / comment the `QEP_ERROR_ON` define to provoke the Error - you will get a PHE/PCE Error and a wrong counting position after that

--> Set a Breakpoint to line 236 on the `// TI: Breakpoint A` comment

```
cQEP.cpp | main.cpp | globals.h | cQEP.h
227     }
228
229
230
231
232 // Errors
233 if( EQep1Regs.QFLG.bit.PHE || EQep1Regs.QFLG.bit.PCE )
234 {
235     // TI: Breakpoint A
236     EQep1Regs.QCLR.bit.PHE = 1; // simultaneous transition of QEPA and QEPB
237     EQep1Regs.QCLR.bit.PCE = 1; // Position counter error
238 }
```

8) alternatively you can debug via can - you can sent me a email at [heiko.ruth@evomotiv.de](mailto:heiko.ruth@evomotiv.de) and will get a complete CANoe Configuration for easy debugging

