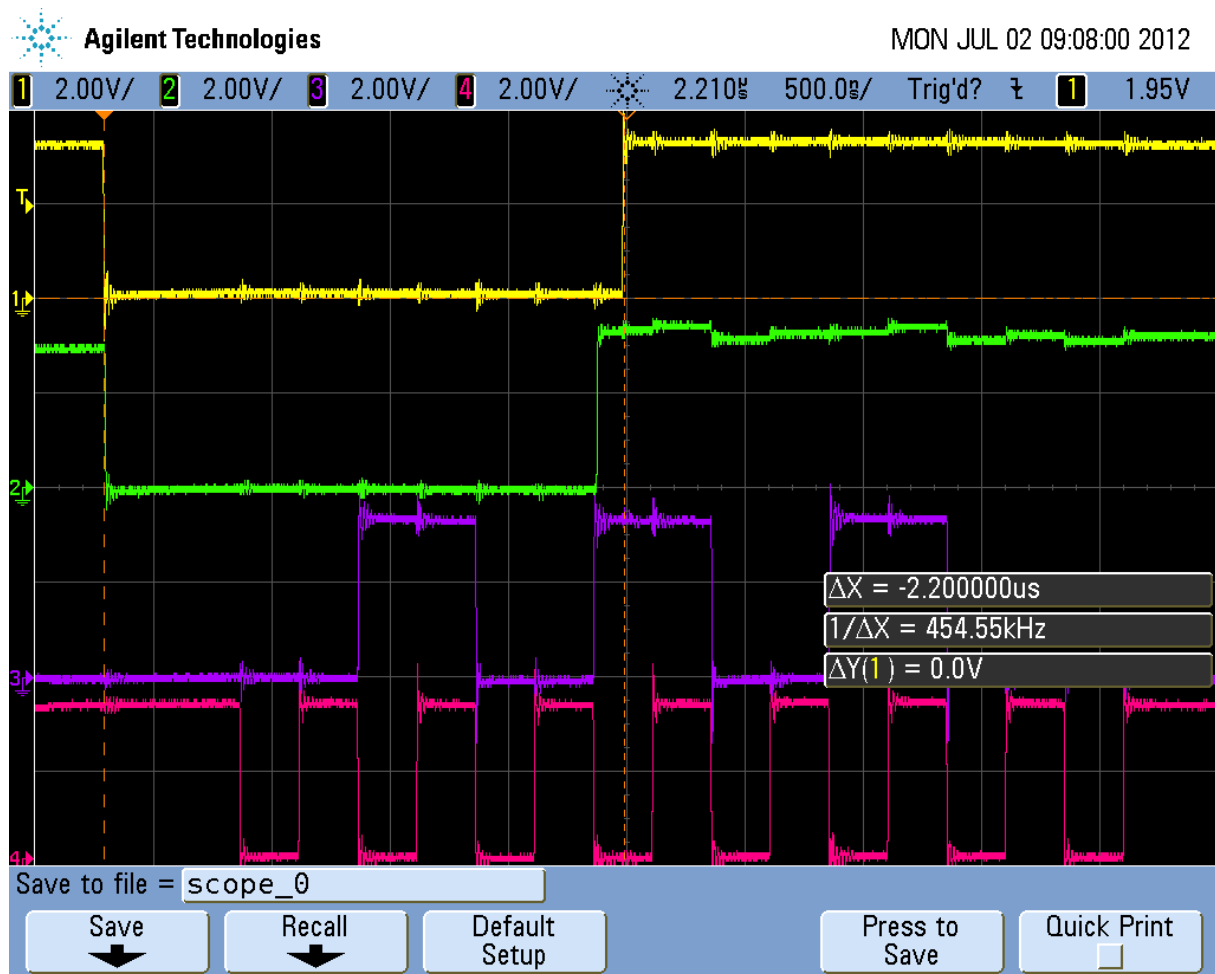


Probleme SPI-Kommunikation zwischen STM32 uP und ADT7310 Temp.Sensor !

Versuch 1 (/CS geht auf high bei Set TXE-Flag vom STM32)

Die Kommunikation funktioniert ab und zu in Kombination mit dem Keil Debugger
mehr zufälligerweise (Auch ohne Debugger) !

Schritt 1: Setzte mit dem Command-Byte 0x054 den ADT7310 in den continuous read mode
(Datenblatt ADT7310 Seite 20)



Gelb = /CS Grün = Dout (MISO) Violette = Din (MOSI) Rot = CLK (500ns = 2MHz)

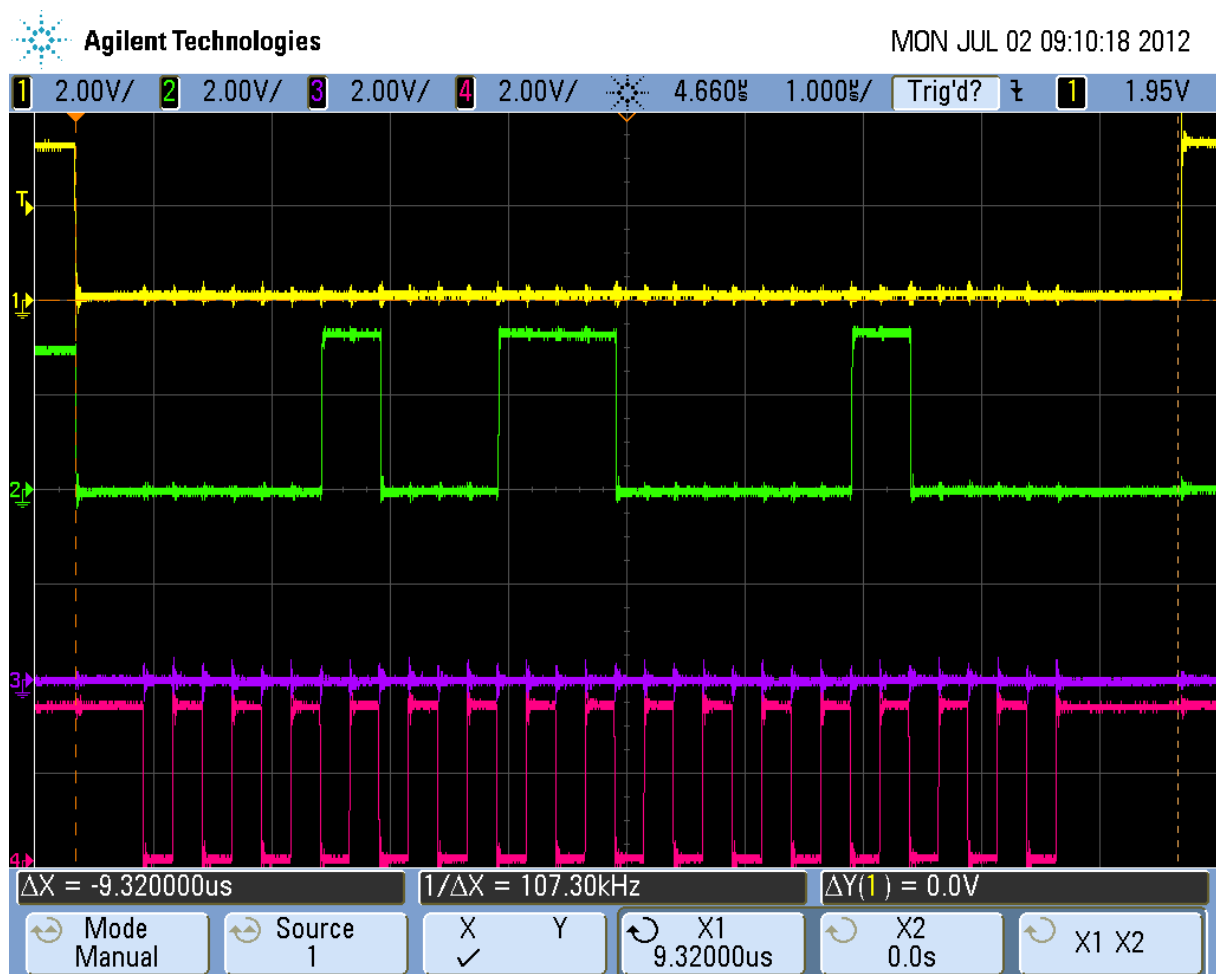
SPI timing spez. (Datenblatt ADT7310 Seite 4)

T1	Soll	0ns min.	Ist	580ns
T2	Soll	100ns min.	Ist	250ns
T6	Soll	0ns min. 80ns max.	Ist	-2.1us !!!

Bem: /CS Steuerung

```
while(SPI_I2S_GetFlagStatus(SPI2, SPI_I2S_FLAG_TXE)==RESET)
IO_SET_PIN(TSens_SPI2_GPIO, TSens_SPI2_nSS_PIN);
```

Schritt 2: Lese Daten (durch erzeugen von 16 clocks) vom ADT7310 im continuous read mode (Datenblatt ADT7310 Seite 20)



Gelb = /CS Grün = Dout (MISO) Violette = Din (MOSI) Rot = CLK (500ns = 2MHz)

SPI timing spez. (Datenblatt ADT7310 Seite 4)

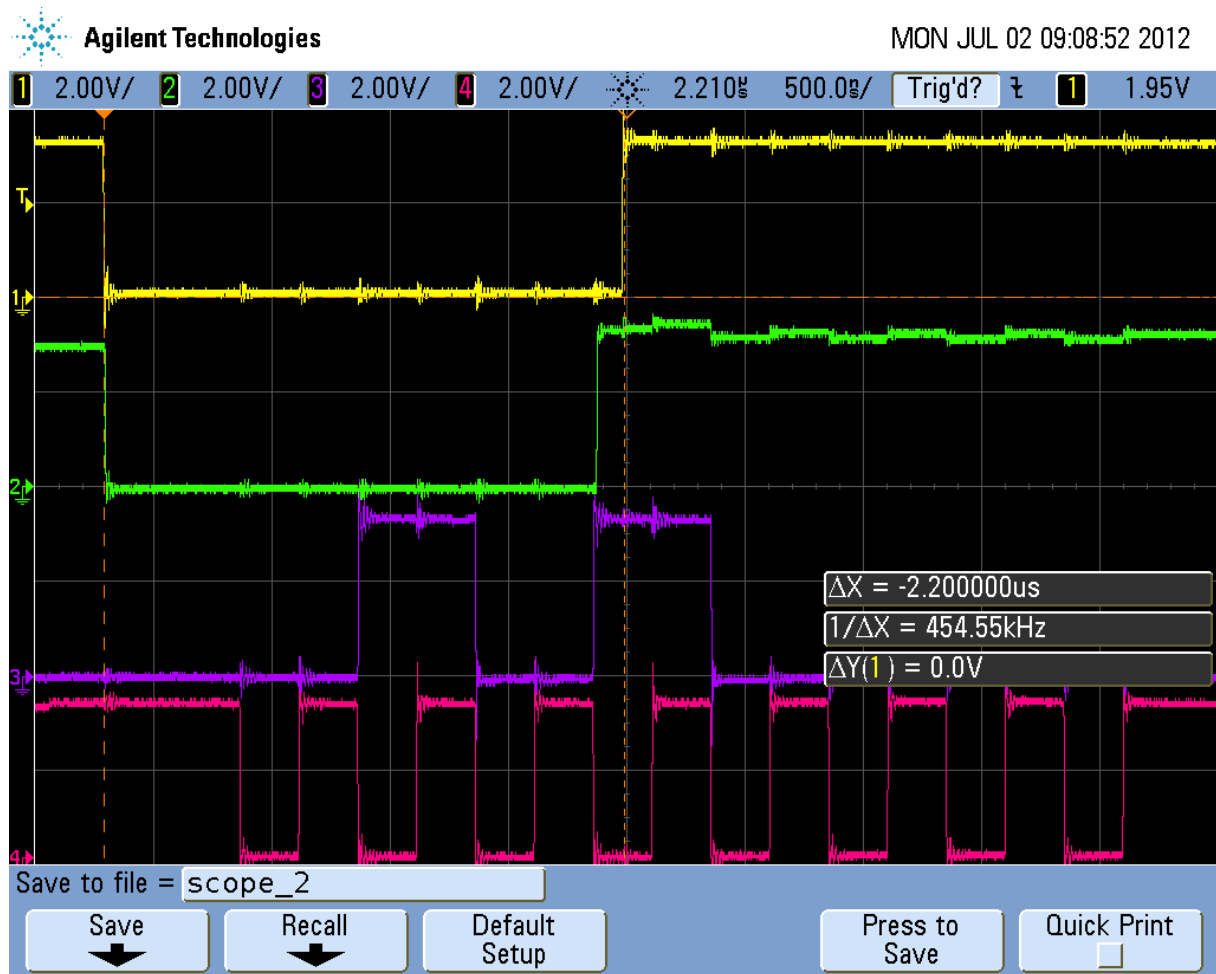
T1	Soll	0ns min.	Ist	580ns
T2	Soll	100ns min.	Ist	250ns
T6	Soll	0ns min. 80ns max.	Ist	1us

Durch mehrmaliges senden des Schritt 1 (0x54) lässt sich der Chip in den cont.read mode versetzen. Danach können die Daten korrekt ausgelesen werden, obwohl der /CS erst nach 1us ab letztem clock wieder auf high geschaltet wird.

Bem. (/CS geht auf high bei Set RXNE-Flag vom STM32)

```
while(SPI_I2S_GetFlagStatus(SPI2, SPI_I2S_FLAG_RXNE) == RESET)
IO_SET_PIN(TSens_SPI2_GPIO, TSens_SPI2_nSS_PIN);
```

Schritt 3: Deaktiviere mit dem Command-Byte 0x050 den ADT7310 den continuous read mode (Datenblatt ADT7310 Seite 20)



Gelb = /CS Grün = Dout (MISO) Violette = Din (MOSI) Rot = CLK (500ns = 2MHz)

SPI timing spez. (Datenblatt ADT7310 Seite 4)

T1	Soll	0ns min.	Ist	580ns
T2	Soll	100ns min.	Ist	250ns
T6	Soll	0ns min. 80ns max.	Ist	-2.1us!!!

Durch mehrmaliges senden des Schritt 3 (0x50) kann der cont.read mode deaktiviert werden. Danach können die Daten nicht mehr nur durch erzeugen von 16 clocks ausgelesen werden. (Korrektes Verhalten)

Wird der /CS in Abhängigkeit von TXE und RXNE Flag vom STM32 auf High gesetzt, kann man den ADT7310 durch mehrmaligen Senden der Command Bytes (0x54 rsp. 0x50) in den cont. read mode gesetzt oder deaktiviert werden.

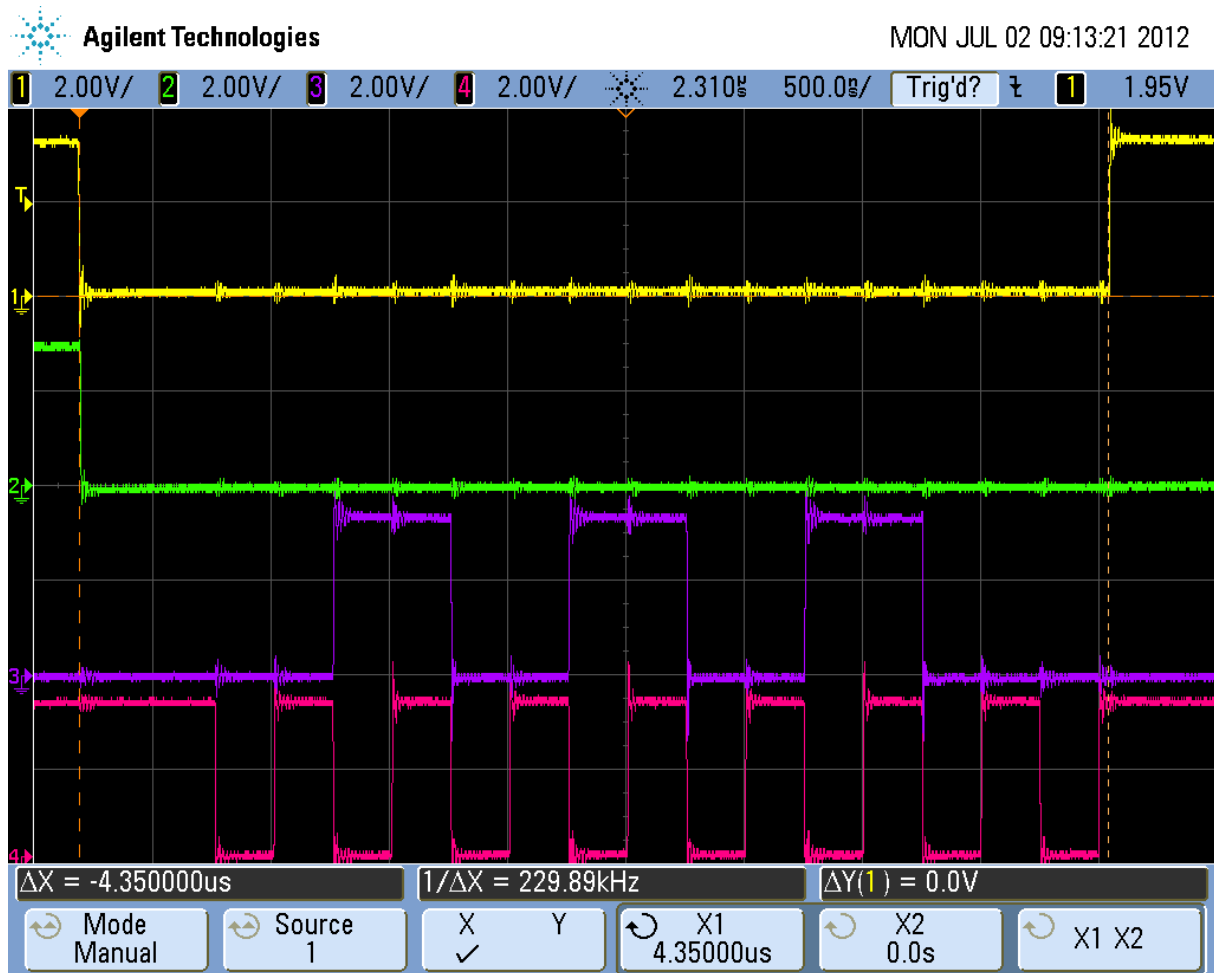
Obwohl dies nicht der timing spez. Entspricht !?!?

Ist der Chip aber einmal (Zufälligerweise) im cont read mode lassen sich auch die Daten zuverlässig auslesen.

Versuch 2 (/CS geht auf high durch Einhaltung der timing spez.)

Die Kommunikation funktioniert ab und zu in Kombination mit dem Keil Debugger
Ohne Debugger aber kein auslesen der Temperatur möglich !

Schritt 1: Setzte mit dem Command-Byte 0x054 den ADT7310 in den continuous read mode
(Datenblatt ADT7310 Seite 20)



Gelb = /CS Grün = Dout (MISO) Violette = Din (MOSI) Rot = CLK (500ns = 2MHz)

SPI timing spez. (Datenblatt ADT7310 Seite 4)

T1	Soll	0ns min.	Ist	580ns
T2	Soll	100ns min.	Ist	250ns
T6	Soll	0ns min. 80ns max.	Ist	46ns

Bem: /CS Steuerung

```
while(SPI_I2S_GetFlagStatus(SPI2, SPI_I2S_FLAG_TXE)==RESET)
```

```
delay = 4.35us;
```

```
IO_SET_PIN(TSens_SPI2_GPIO, TSens_SPI2_nSS_PIN);
```

Schritt 2: Lese Daten (durch erzeugen von 16 clocks) vom ADT7310 im continuous read mode (Datenblatt ADT7310 Seite 20)



Gelb = /CS Grün = Dout (MISO) Violette = Din (MOSI) Rot = CLK (500ns = 2MHz)

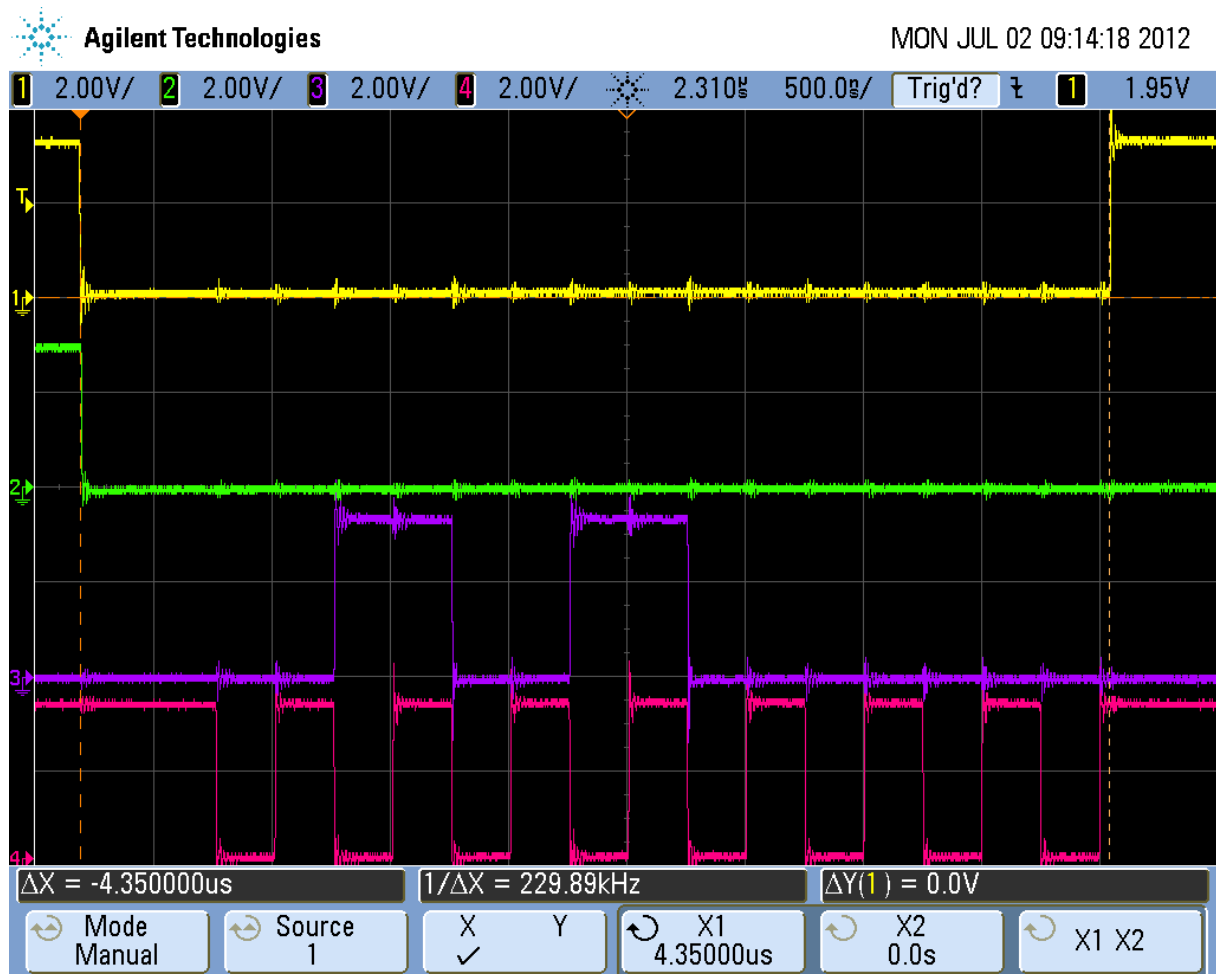
SPI timing spez. (Datenblatt ADT7310 Seite 4)

Parameter	Type	Spec	Actual
T1	Soll	0ns min.	580ns
T2	Soll	100ns min.	250ns
T6	Soll	0ns min. 80ns max.	1µs

Ohne Debugger lässt sich der Chip nicht in den cont read mode versetzen und dementsprechend können auch keine Daten ausgelesen werden ? Und dies Obwohl die timing spez. Eingehalten werden. Mit Debugger lässt sich der Chip aber (Wohl eher Zufällig) in den cont. read mode versetzen und die Daten können korrekt ausgelesen werden !

Bem. (/CS geht auf high bei Set RXNE-Flag vom STM32)

Schritt 3: Deaktiviere mit dem Command-Byte 0x050 den ADT7310 den continuous read mode (Datenblatt ADT7310 Seite 20)



Gelb = /CS Grün = Dout (MISO) Violette = Din (MOSI) Rot = CLK (500ns = 2MHz)

SPI timing spez. (Datenblatt ADT7310 Seite 4)

T1	Soll	0ns min.	Ist	580ns
T2	Soll	100ns min.	Ist	250ns
T6	Soll	0ns min. 80ns max.	Ist	46ns

Durch mehrmaliges senden des Schritt 3 (0x50) kann der cont.read mode deaktiviert werden. Danach können die Daten nicht mehr nur durch erzeugen von 16 clocks ausgelesen werden. (Korrektes Verhalten)

Obwohl nun die timing spez. Eingehalten werden kann der Chip mit Debugger eher zufällig ohne Debugger gar nicht korrekt betrieben werden.

Was mache ich falsch ??