

Praktikum Kommunikationstechnik

Versuch: Digitale Vermittlungstechnik 1

1 Versuchsziele

Kennen lernen:

- einer Vermittlungshardware,
- des Protokolls zur Ansteuerung beziehungsweise Abfrage,
- eines Software-Moduls für Schleifenerkennung,
- eines Software-Moduls für die Ruf- und Hörtonsteuerung.

Versuchsaufbau

Die Vermittlung ist in einem Kompaktgehäuse untergebracht und enthält:

- Netzteile zur Erzeugung der Betriebsspannungen (2,5 V, 3,3 V und 20 V),
- ein Taktmodul zur Erzeugung notwendiger PCM-Takte,
- einen LAN- und USB-Anschluss zur Kommunikation mit dem Steuerrechner,
- einen Mikrokontroller für die Koordination des Hardwarezugriffs,
- vier a/b-Teilnehmersätze,
- vier RJ45-Anschlüsse um die Endgeräte an der Vermittlung anzuschließen.

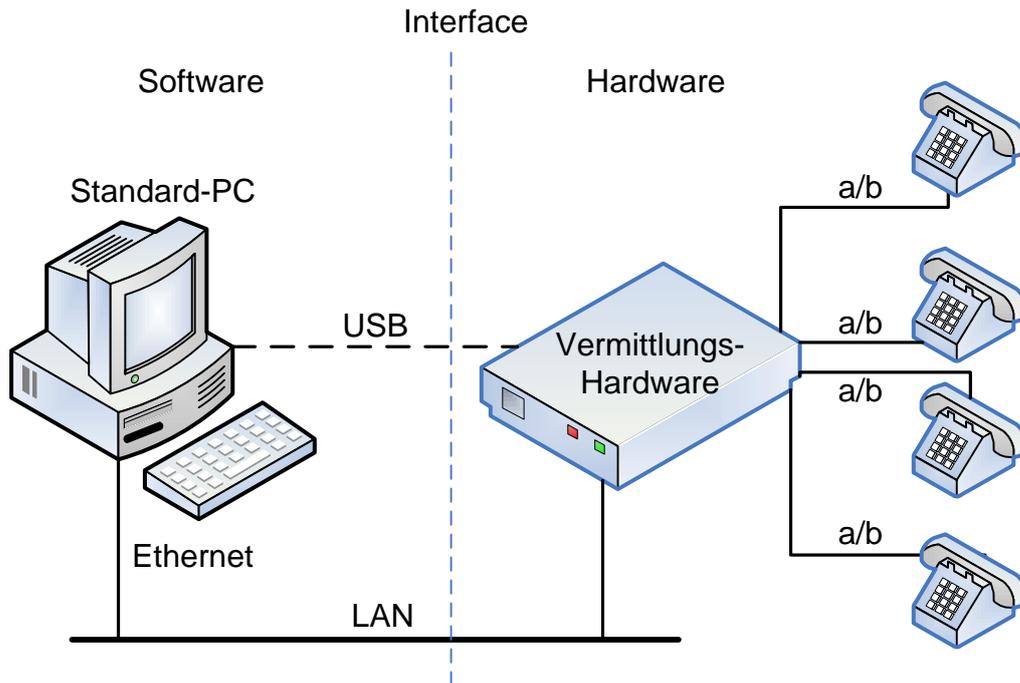


Abbildung 1: Versuchsaufbau

1.1 Übersicht zur Hardware der Vermittlung

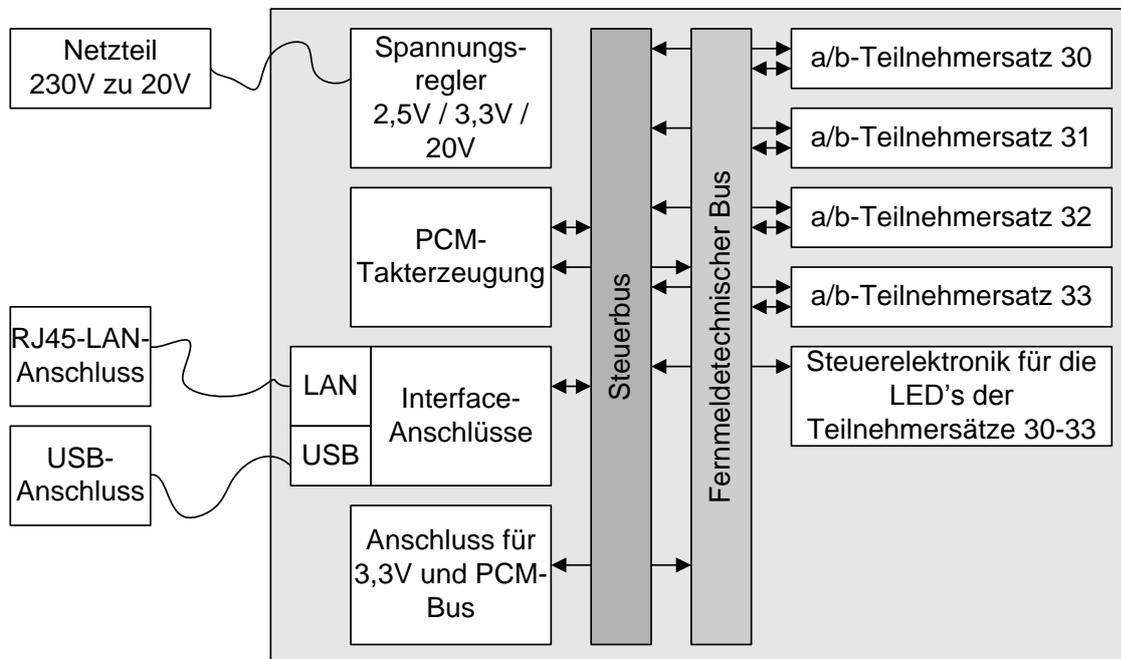


Abbildung 2: Hardwareübersicht der Vermittlung

Bussystem: Jede Leiterplatte ist über einen 68-poligen Steckverbinder mit dem internen Bussystem verbunden. Über das Bussystem werden der fernmeldetechnische Bus (FMT-Bus) und der Steuerbus geführt. Im FMT-Bus werden die Signale des PCM-Busses und die 3,3 V und 20 V Betriebsspannung für die SLIC's verteilt. Über den Steuerbus kann auf die Teilnehmersätze zugegriffen werden. Dies geschieht über entsprechende Commands.

Spannungsregler: Die Spannungsregler versorgen die internen Systeme mit 2,5 V und 3,3 V für digitale Komponenten und 3,3 V und 20 V für die Teilnehmersätze. Die beiden Spannungen werden über Schaltregler vom Typ LM2576, der Firma National Semiconductor, aus den vom Netzteil kommenden 20 V Betriebsspannung gewonnen.

PCM-Takterzeugung: Jede Digitalvermittlung benötigt eine PCM-Taktversorgung. Hier werden, basierend auf einem genauen Quarzgenerator mit einem Grundtakt von 16,384 MHz der Bittakt von 256 kHz bis 4096 kHz und der Rahmentakt (FSYNC) von 8 kHz erzeugt. Die Taktversorgung kann auch von extern über den PCM-Anschluss erfolgen. Die Taktversorgung muss sich normalerweise auf die Phase einer ankommenden PCM-Primärfolge synchronisieren können. Es werden dafür üblicherweise digitale Phasenregler (DPLL) benutzt. Diese Funktionalität ist in der Vermittlungshardware hier nicht realisiert.

Interface-Anschluss: Das interne Bussystem ist über einen Mikrokontroller mit dem Interface-Anschluss verbunden. Der Mikrokontroller übernimmt an dieser Stelle verschiedene Protokoll-/Signalanpassungen und Steuerungsaufgaben.

Anschluss für PCM-Bus: Hier können die PCM-Bussignale abgegriffen und dargestellt werden. Es ist ebenso eine Stromversorgung von 3,3 V vorhanden.

a/b-Teilnehmersätze: Jede Karte enthält vier a/b-Teilnehmersätze. Die Funktionalität wird im folgenden Abschnitt beschrieben.

Steuerelektronik für LED der Teilnehmersätze: Mit Hilfe dieser Baugruppe werden die Leuchtdioden auf den Teilnehmersätzen angesteuert.

1.2. Hardware der analogen Teilnehmerschaltungen

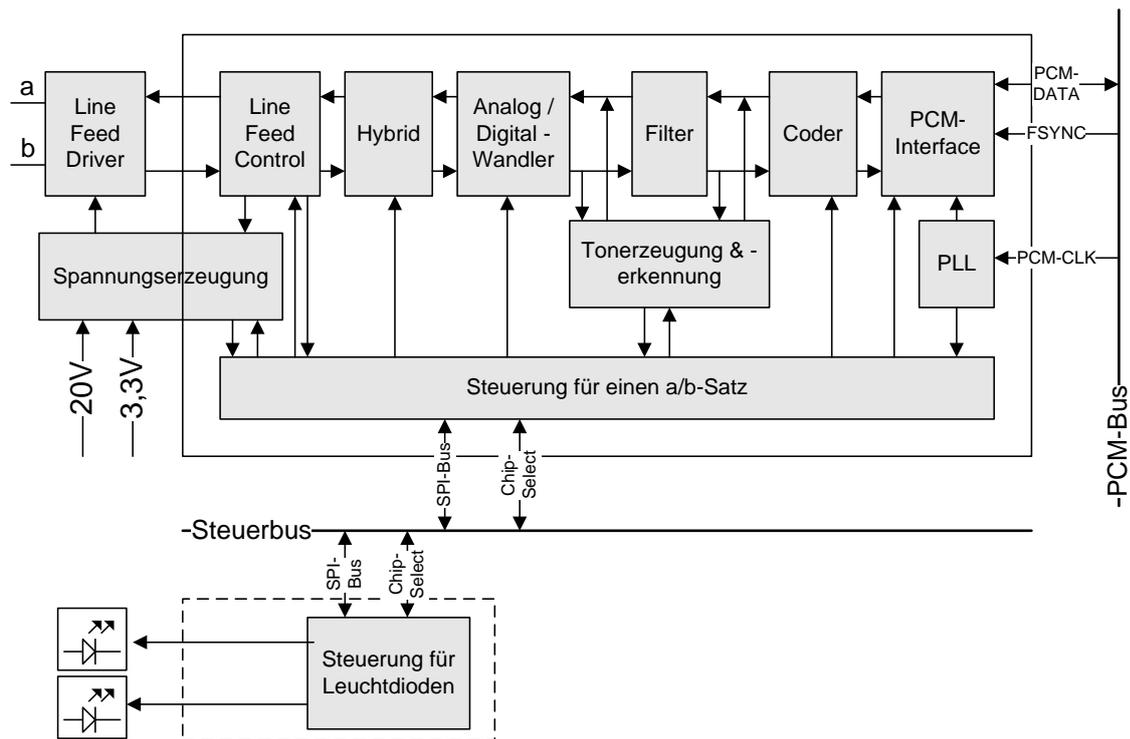


Abbildung 3: Blockschaltung eines Teilnehmersatzes

Auf jeder Leiterkarte sind vier a/b-Sätze und die zugehörige Steuerelektronik untergebracht. Jede Platine hat eine über Lötbrücken einstellbare Basisadresse (z.B. %x30 oder %x34). Die vier Teilnehmer der Karte mit der Basisadresse %x30 haben die einzelnen Teilnehmeradressen %x30, %x31, %x32, %x33. Die Steuerelektronik für die LED hat dazu die Adresse %x38.

Schutzschaltung

Am Eingang der Teilnehmerschaltung sollte ein Überlastungsschutz integriert sein, auf den an dieser Stelle aus Platzgründen verzichtet wurde. In der Regel wird der Überspannungsschutz in mehreren Stufen ausgeführt, z.B. Grob- und Feinschutz. Der Feinschutz wird meistens als eine Diodenschaltung, der Grobschutz in Form von Überspannungsgasableitern im Hauptverteiler (HVT) ausgeführt.

Rufsteuerung

Über den Line-Feed-Control wird der Line-Feed-Driver angesteuert, der steuerbare Stromquellen beinhaltet. Durch entsprechende Ansteuerung wird die Rufspannung auf der a/b-Leitung erzeugt. Die An- und Abschaltung der Rufwechselspannung erfolgt auch hier ausschließlich in Nulldurchgang, um die spektralen Komponenten der Span-

nung möglichst gering zu halten. Einen mechanischen Verschleiß gibt es bei dieser Art der Rufaufschaltung nicht, da es keine mechanischen Kontakte gibt. Der SLIC realisiert durch die gesteuerten Stromquellen auch die Speisung der Schleife, den Leitungsabschluss (Hybrid), die Signalwandlung und Kodierung. Über die Steuerung wird die Schleifenspannung ausgewertet. Bei einer Spannung kleiner 16 V und einem Stromfluss, wird eine Schleife signalisiert. Es ist an dieser Stelle eine Spannungsauswertung nötig, da die Stromquelle versucht den Strom auf ca. 20 mA konstant zu halten.

Die Hörtöne werden, von digitalen Oszillatoren erzeugt, direkt in den digitalen Datenstrom eingespeist.

Der CODEC ist frei einstellbar auf 8/16 Bit lineare, 8 Bit A-Law und μ -Law Kodierung.

Die Sende und Empfangszeitlage sind bitgenau einstellbar.

Die Steuerung für den SLIC hat eine Besonderheit. Bei einem Lesevorgang ist das MSB der Adresse immer auf logisch 1 zusetzen, bei einem Schreibvorgang immer auf logisch 0.

2 PBX-Protokoll

2.1 Übersicht über die Steuer-PDUs

Die wichtigsten PDUs zur Ansteuerung der SLIC's sind in nachfolgenden Tabellen aufgeführt: (Angabe der PDUs in hexadezimalen Zahlen.)

Bedeutung	PDU	Initialisierungs-Reihenfolge
<u>Get-Loop</u> (Schleifenzustand der Teilnehmer 30-37 zyklisch lesen beenden)	81 53 00 00 00 FE	1
<u>Basic-Init</u> (Takterzeugung initialisieren)	81 68 00 00 00 FE	2
<u>SLIC-Reset</u> (Teilnehmersätze zurücksetzen)	81 59 00 00 00 FE	3
<u>Set-PCM-Takte</u> (PCM-Bittakt auf 2048 kHz und Rahmentakt auf 8 kHz festlegen)	81 54 00 04 00 FE	4
<u>SLIC-Init</u> (Teilnehmersatz initialisieren)	81 49 <Adresse> 00 00 FE	5

Tabelle 1:Initialisierungs-PDU

Nach Übermittlung der fünf genannten PDUs können die nachfolgenden Steuer-PDUs angewendet werden.

Bedeutung	PDU
<u>Set-LED-on</u> (LED am Teilnehmersatz einschalten) Nr.: 1= Gelb, 2 = Grün	81 4C <Adresse> <Nr.> 00 FE
<u>Set-LED-off</u> (LED am Teilnehmersatz ausschalten) Nr.: 1= Gelb, 2 = Grün Ansonsten wie Nummerierung	81 4D <Adresse> <Nr.> 00 FE
<u>Set-LED-blink</u> (LED am Teilnehmersatz langsam blinken lassen) Nr.: 1= Gelb, 2 = Grün Ansonsten wie Nummerierung	81 4F <Adresse> <Nr.> 00 FE

<u>Set-LED-flash</u> (LED am Teilnehmersatz schnell blinken lassen) Nr.: 1= Gelb, 2 = Grün Ansonsten wie Nummerierung	81 4E <Adresse> <Nr.> 00 FE								
<u>Set-State</u> (Schleifenspeisung für Teilnehmer ausschalten, Ruf abschalten)	81 52 <Adresse> 00 00 FE								
<u>Set-State</u> (Schleifenspeisung für Teilnehmer einschalten, Ruf wird abschalten)	81 52 <Adresse> 02 00 FE								
<u>Set-State</u> (Ruf für Teilnehmer einschalten, Teilnehmer wird gerufen)	81 52 <Adresse> 04 00 FE								
<u>Set-SZL</u> (Zeitlage für das Senden festlegen) Zeitlage: 0-FF, abhängig vom PCM-Takt	81 5A <Adresse> <Zeitlage> 00 FE								
<u>Set-EZL</u> (Zeitlage für das Empfangen festlegen) Zeitlage: 0-FF, abhängig vom PCM-Takt	81 5B <Adresse> <Zeitlage> 00 FE								
<u>Set-PCM-Mode</u> (PCM-Codierung festlegen und Übertragung aktivieren) Art: 0 = deaktiviert, Art: 3C = 16-Bit linear,	81 51 <Adresse> <Art> 00 FE								
<table border="1"> <thead> <tr> <th>Bitbreite</th> <th>A-Law</th> <th>μ-Law</th> <th>linear</th> </tr> </thead> <tbody> <tr> <td>8</td> <td>Art = 20</td> <td>Art = 28</td> <td>Art = 38</td> </tr> </tbody> </table>	Bitbreite	A-Law	μ -Law	linear	8	Art = 20	Art = 28	Art = 38	
Bitbreite	A-Law	μ -Law	linear						
8	Art = 20	Art = 28	Art = 38						
<u>Set-Tone</u> (440Hz-Dauerton zum Teilnehmer einschalten)	81 50 <Adresse> 01 00 FE								
<u>Set-Tone</u> (440Hz-Dauerton zum Teilnehmer ausschalten)	81 50 <Adresse> 00 00 FE								
<u>Get-Loop</u> (Schleifenzustand der Teilnehmer 30-37 zyklisch aller 10ms automatisch lesen)	81 53 01 00 00 FE								
<u>Get-Loop</u> (Schleifenzustand der Teilnehmer 30-37 zyklisch lesen beenden)	81 53 00 00 00 FE								
<u>Get-Loop</u> (Schleifenzustand der Teilnehmer 30-37 einmalig lesen)	81 53 02 00 00 FE								
<u>Set-3PTY</u> (3'er-Konferenzfunktion aktivieren)	81 4A 00 01 00 FE								
<u>Set-3PTY</u> (3'er-Konferenzfunktion deaktivieren)	81 4A 00 00 00 FE								
<u>Set-PCM-Stream</u> (PCM-Kanal 23 lesen aktivieren und	81 4B 00 01 00 FE								

PCM-Kanal 24 schreiben aktivieren)	
<u>Set-PCM-Stream</u> (PCM-Kanal lesen & schreiben deaktivieren)	81 4B 00 00 00 FE

Tabelle 2: Steuer-PDU

Geräteadressen

Adresse	Gerät / Bezeichnung
30	Teilnehmersatz 1 auf Platine 1
31	Teilnehmersatz 2 auf Platine 1
32	Teilnehmersatz 3 auf Platine 1
33	Teilnehmersatz 4 auf Platine 1
34	Teilnehmersatz 1 auf Platine 2
35	Teilnehmersatz 2 auf Platine 2
36	Teilnehmersatz 3 auf Platine 2
37	Teilnehmersatz 4 auf Platine 2
38	Steuerkontroller für Teilnehmersatz-LED auf Platine 1
39	Steuerkontroller für Teilnehmersatz-LED auf Platine 2
40	Mikrokontroller (CPU) auf CPU-Platine
41	FPGA-IC auf CPU-Platine (XILINX XC2S50)
42	Flash-ROM auf CPU-Platine (M25P10)

Tabelle 3: Geräteadressen

2.1.1. MSC zur Initialisierung der Hardware

Dieser Ablauf wird nach jedem Neustart der Vermittlungshardware zur Initialisierung benötigt.

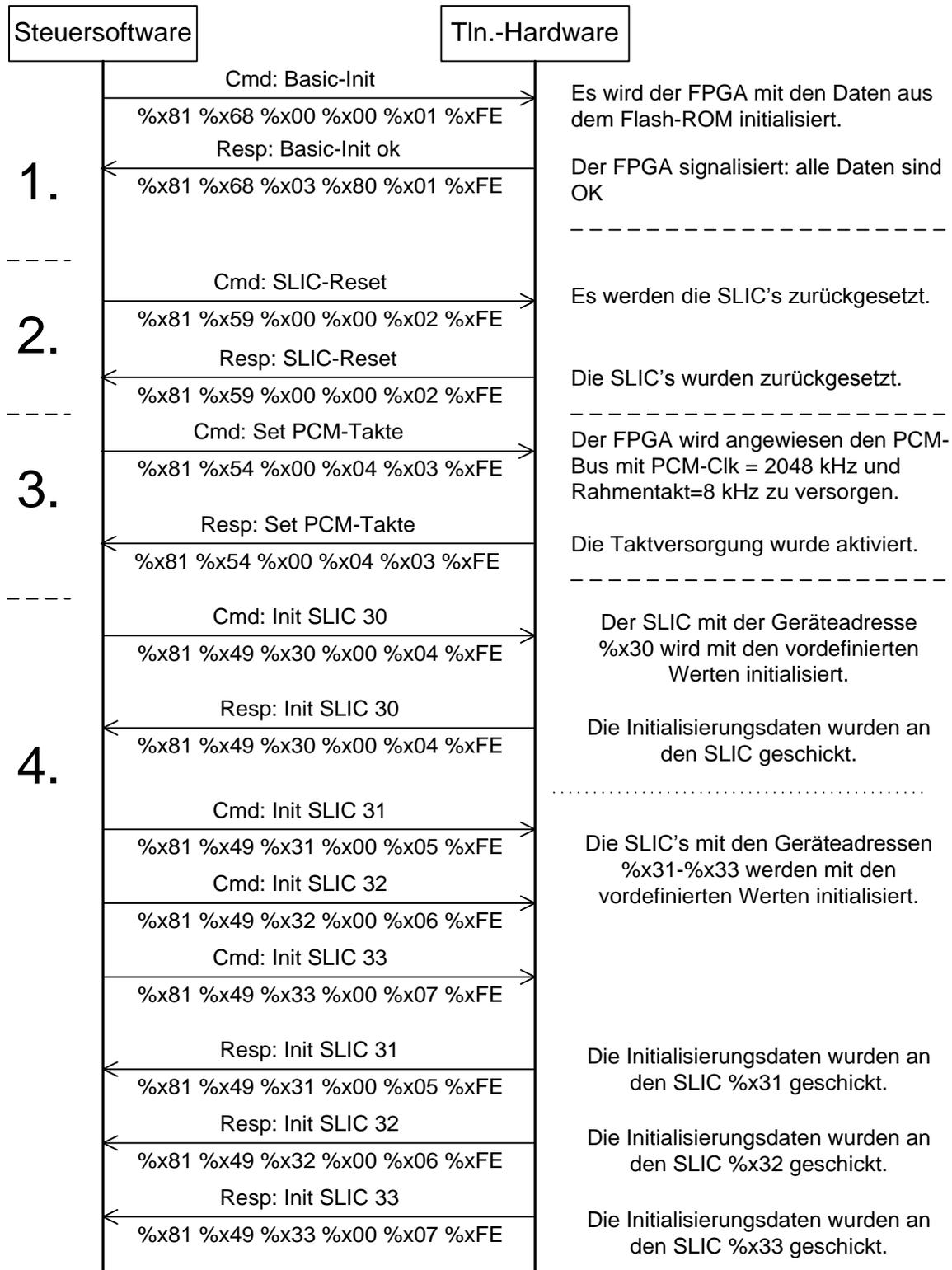


Abbildung 4: Initialisierung der Vermittlungshardware (MSC)

2.1.2. MSC zum Lesen eines Direktregisters eines SLIC

Dieser Ablauf wird benötigt, wenn ein Wert eines Register in dem SLIC gelesen werden soll.

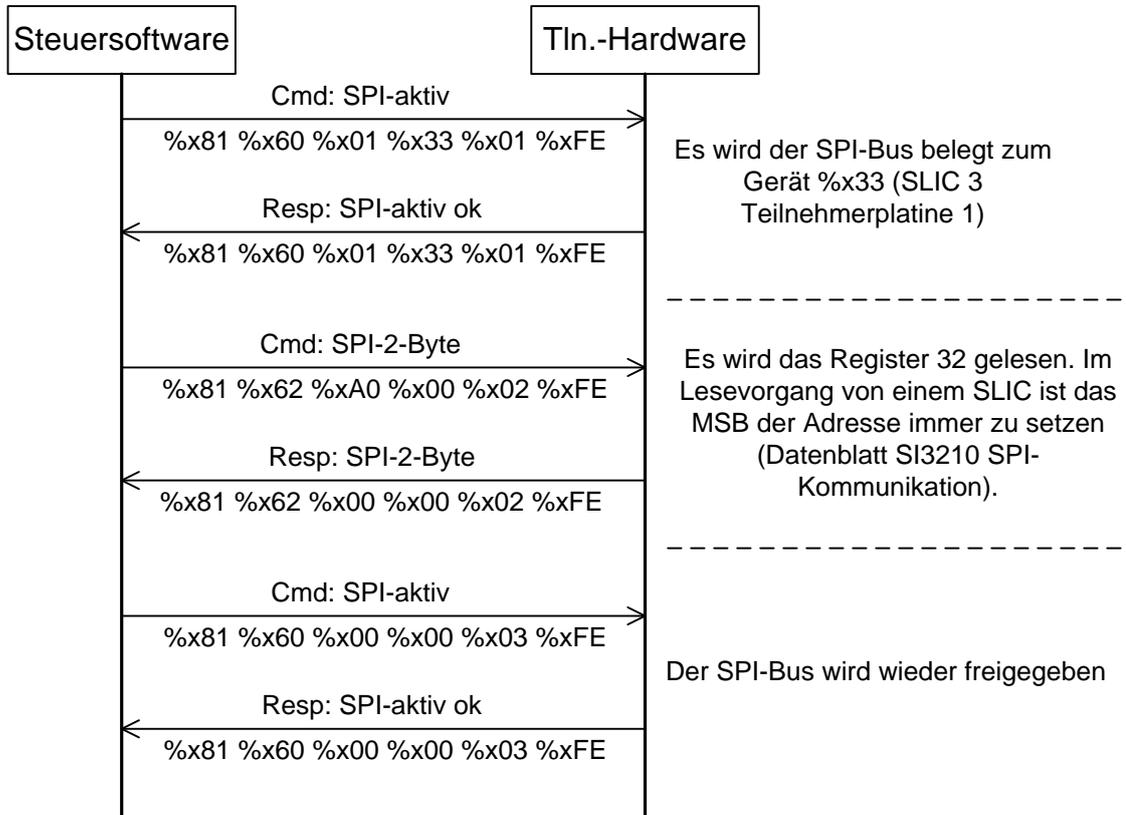


Abbildung 5: Lesen eines Direktregisters eines SLIC (MSC)

2.1.3. MSC zum Schreiben eines Direktregisters eines SLIC

Diese Ablauf wird benötigt, wenn ein Wert eines Register in dem SLIC geschrieben werden soll.

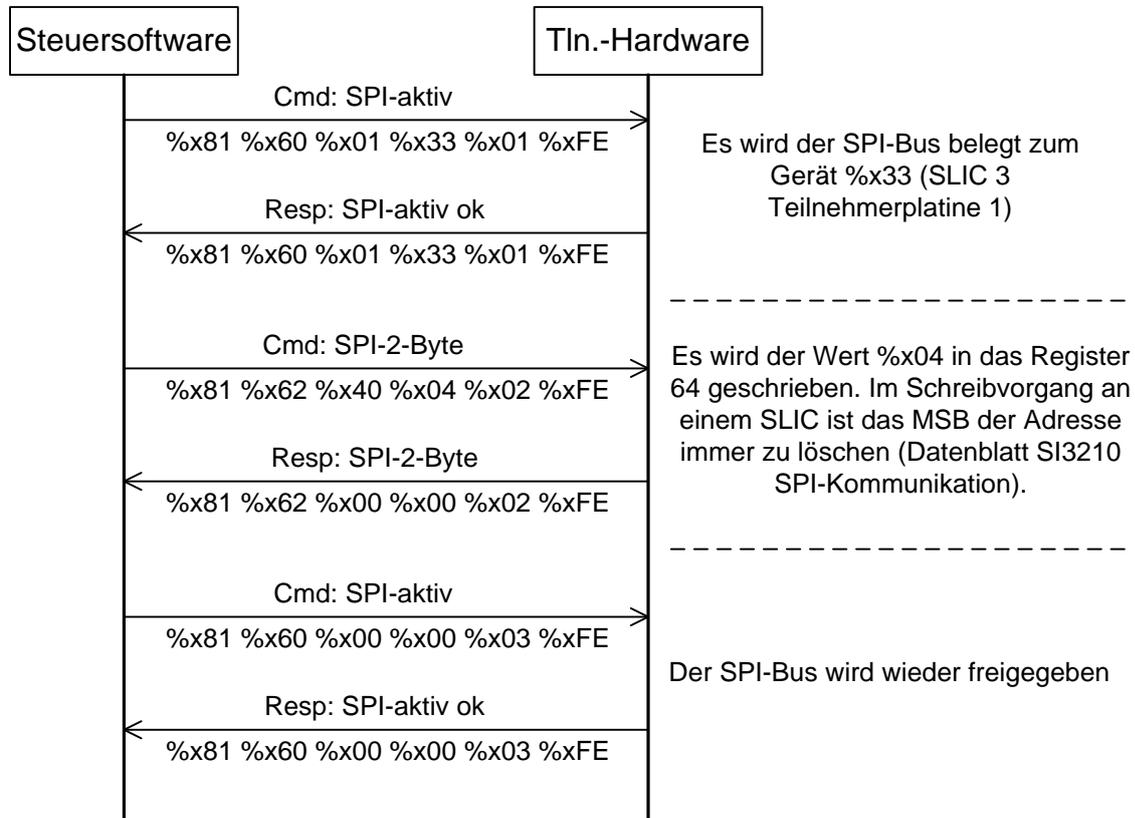


Abbildung 6: Schreiben eines Direktregisters eines SLIC (MSC)

2.1.4. MSC zum Lesen eines indirekten Registers eines SLIC

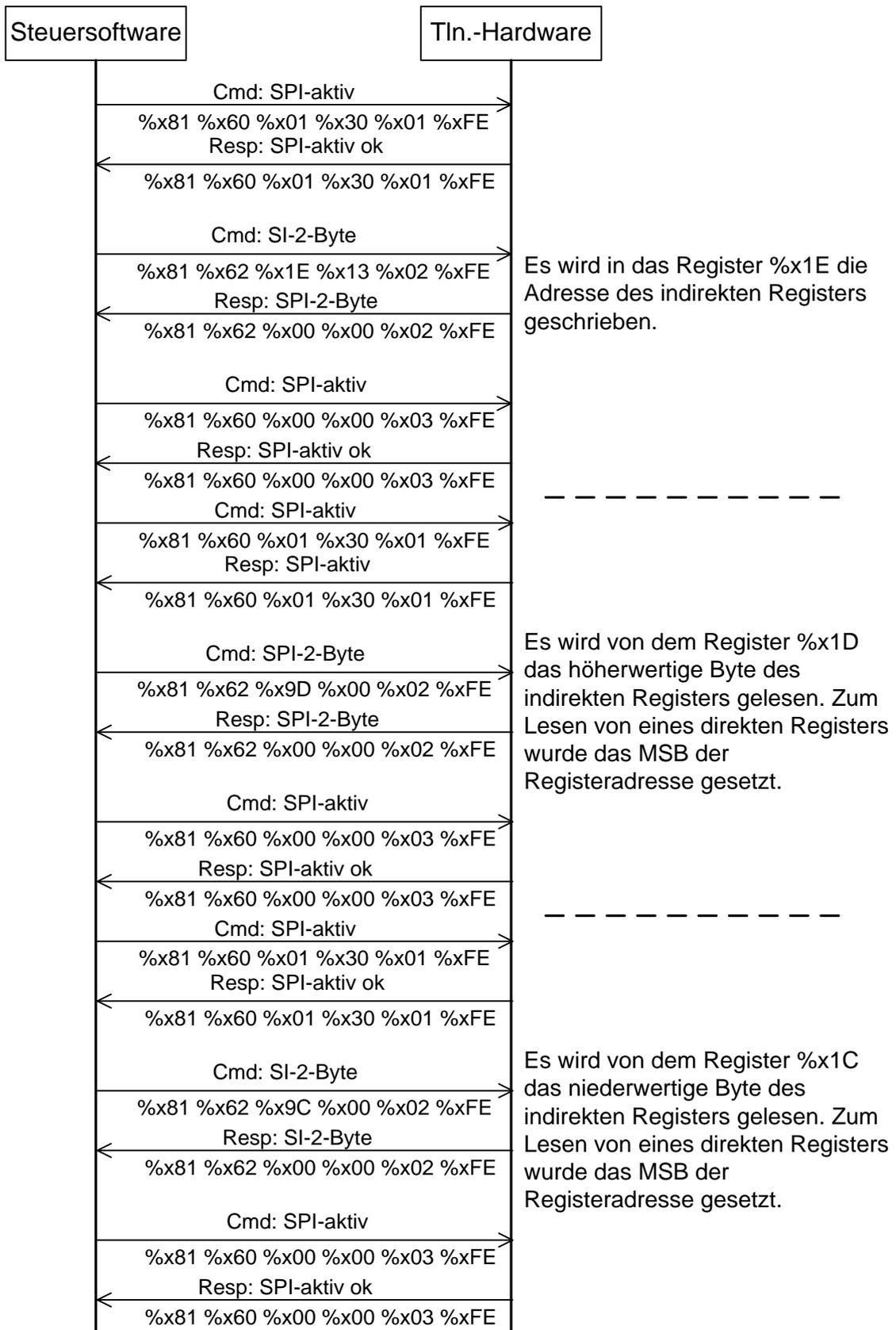


Abbildung 7: Lesen eines indirekten Registers eines SLIC (MSC)

2.1.5. MSC zum Schreiben eines indirekten Registers eines SLIC

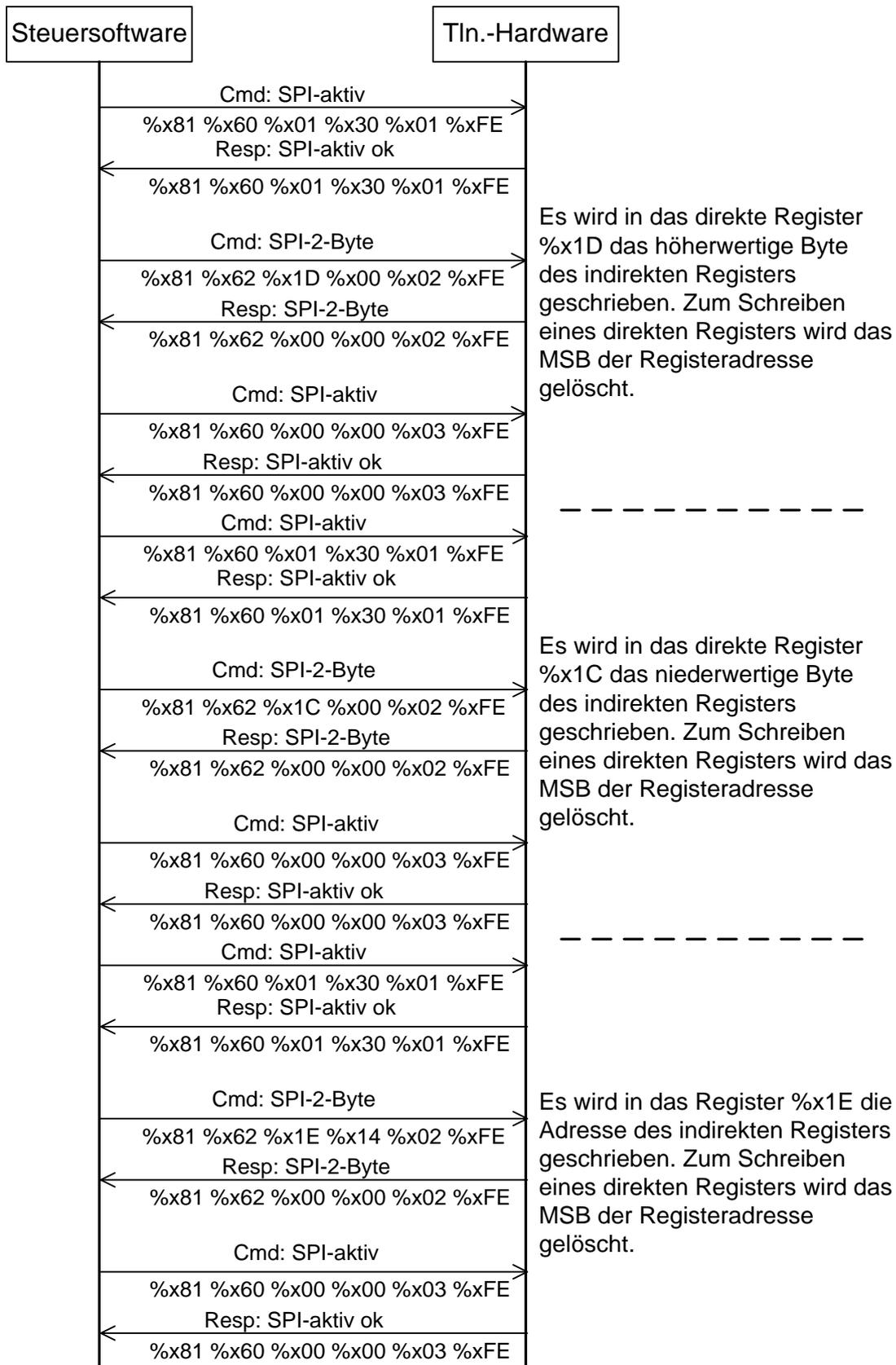


Abbildung 8: Schreiben eines indirekten Registers eines SLIC (MSC)

3 Versuchsumfeld

3.1 Entwicklungsumgebung

Als Entwicklungsumgebung wird Borland Developer Studio 2006 (BDS 2006) eingesetzt. Die Software bietet die Möglichkeit, Programme in Delphi, C++ und C# zu entwickeln. Für die Programmierung der PBX-Steuerungssoftware wird die Programmiersprache C++ eingesetzt. Hierfür soll eine VCL-Formularanwendung verwendet werden. Diese können Sie in BDS 2006 unter Datei → Neu → VCL-Formularanwendung erstellen.

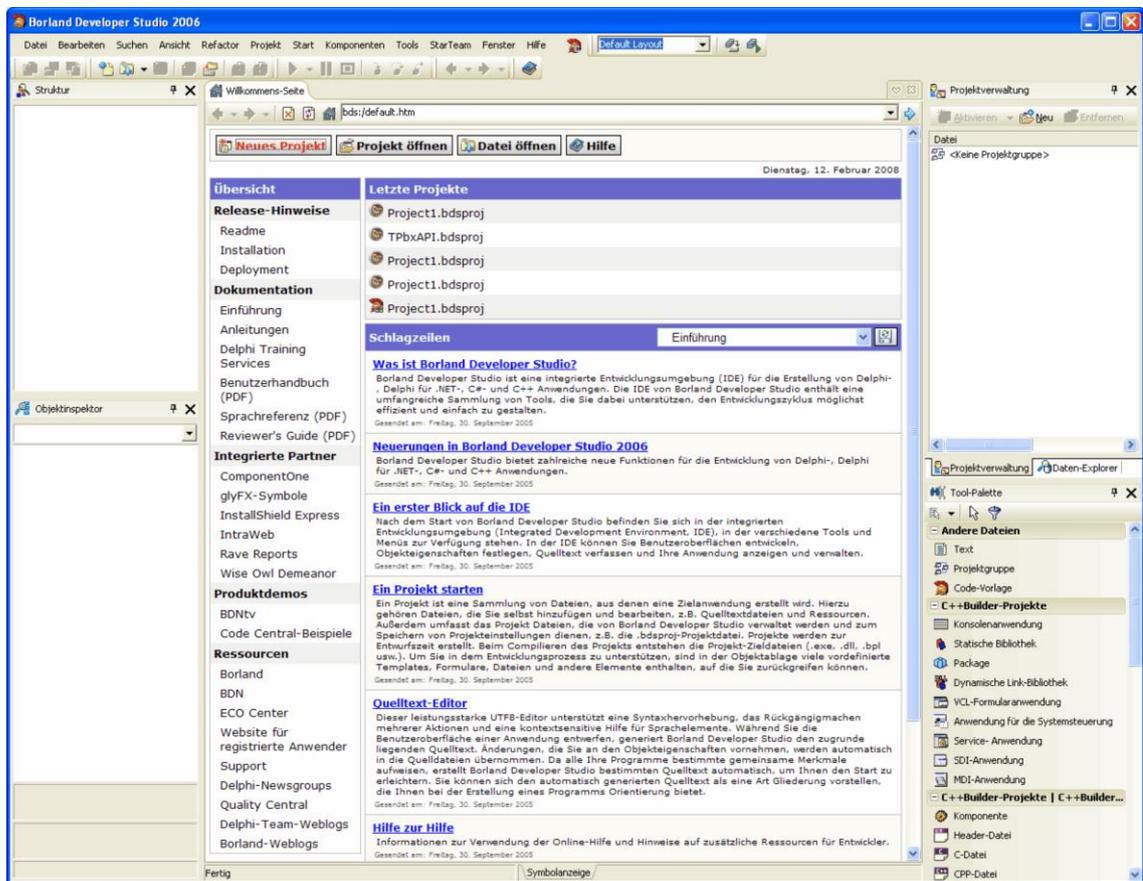


Abbildung 9: BDS 2006

3.2 Kommunikation mit der Versuchshardware

Wie bereits in Kapitel 0 erwähnt, bietet die Versuchshardware als Steuerschnittstellen USB und Ethernet an, wobei für die Versuchsdurchführung die Ethernet-Schnittstelle verwendet wird. Dies deutet darauf hin, dass zur Kommunikationsrealisierung eine Socketprogrammierung notwendig ist. Wie Sie bereits in der Vorlesung „Software für Kommunikationssysteme“ feststellen konnten, würde dies zu einem erheblichen Mehr-

aufwand zur Realisierung der eigentlichen Aufgabe führen. Aus diesem Grund steht Ihnen für die Kommunikation mit der Versuchshardware eine C++-Komponente „TPbxAPI“ zur Verfügung, welche die Socketverbindung zur Vermittlungshardware bereits integriert hat und ggf. eine Fehlerbehandlung durchführt. Sie finden diese Komponente in der Toolbox von BDS 2006, unter der Registerkarte „PbxPraktikum“.

3.2.1 Verwendung der Komponenten „TPbxAPI“

Zur Verwendung der Komponente „TPbxAPI“ können Sie diese auf Ihrem Formular platzieren. Die Einstellungen für die IP-Adresse und den Port können Sie über den Objektinspektor unter der Registerkarte „Eigenschaften“ festlegen.

Entnehmen Sie bitte die zu verwendende IP-Adresse und die Portnummer der folgenden Tabelle.

Hostname	IP-Adresse	Port
tc-pabx1.it.htwm.de	141.55.243.219	8000
tc-pabx2.it.htwm.de	141.55.243.220	8000

Tabelle 4: IP-Adresse und Port für die Kommunikation

Neben den Eigenschaften bietet die Komponenten weiterhin noch die in Tabelle 5 dargestellten Ereignisse an, welche zur Auswertung und Funktionskontrolle verwendet werden können.

OnPbxConnected	Verbindung zur PBX wurde erfolgreich hergestellt.
OnPbxDisconnected	Verbindung zur PBX wurde beendet.
OnPbxError	Während der Kommunikation trat ein Fehler auf. Der Fehlertext befindet sich in der übergebenen String-Variable.
OnPbxInitComplete	Initialisierung der PBX durchgeführt.
OnPbxRead	Daten zum Lesen sind verfügbar. Die Daten müssen aus dem Socket gelesen werden.
OnPbxSendBuf	Gesendete Daten werden zurück gemeldet. Die Daten befinden sich in der Char-Variable.

Tabelle 5: Ereignisse von TPbxAPI

Weiterhin bietet die Komponente noch zahlreiche Methoden an, welche die Kommunikation mit der PBX erheblich erleichtern.

PbxConnect()	Verbindung zur PBX herstellen
PbxDisconnect()	Verbindung zur PBX trennen
PbxInit()	PBX initialisieren
PbxSpeisungAn(char port,char referenz)	Speisung anschalten
PbxSpeisungAus(char port,char referenz)	Speisung ausschalten
PbxRufAn(char port,char referenz)	Rufton einschalten
PbxRufAus(char port,char referenz)	Rufton ausschalten
PbxLedAn(char port,char referenz,char farbe)	LED leuchten lassen farbe=0x01 – gelb farbe=0x02 – grün
PbxLedAus(char port,char referenz,char farbe)	LED ausschalten
PbxLedBlinkLangsam(char port,char referenz,char referenz,char farbe)	LED langsam blinken lassen
PbxLedBlinkSchnell(char port,char referenz,char referenz,char farbe)	LED schnell blinken lassen
PbxSZL(char port,char zeitlage,char referenz)	Sende-Zeitlage festlegen
PbxEZL(char port,char zeitlage,char referenz)	Empfangszeitlage festlegen
PbxPcmMode(char port, char mode, char referenz)	PCM-Mode festlegen
PbxTonAn(char port, char referenz)	Ton anschalten
PbxTonAus(char port, char referenz)	Ton ausschalten
PbxSchleifeZykLesenAn(char referenz)	Schleife aller 10ms senden
PbxSchleifeZykLesenAus(char referenz)	Schleife abschalten
PbxSchleifeEinmalLesenAn(char referenzv)	Schleife einmal senden
PbxKonferenzAn(char referenz)	Konferenz anschalten
PbxKonferenzAus(char referenz)	Konferenz ausschalten
PbxPcmStreamAn(char referenz)	PCM-Stream einschalten
PbxPcmStreamAus(char referenz)	PCM-Stream ausschalten
PbxSendePDUBytes(char startbyte,char command_response,char param1,char param2,char referenz,char stopbyte)	Selbsterstellte PDU als Byte-Werte versenden
PbxSendePDUString(AnsiString PDU)	PDU als String versenden

Tabelle 6: Methodenreferenz von TPbxAPI

4 Versuchsvorbereitung und -durchführung

4.1 Versuchsvorbereitung

1. Ermitteln Sie alle notwendigen PDUs und deren Abfolge um die Vermittlungshardware in einen definierten Grundzustand zu bringen.

Schreiben Sie dafür entsprechenden C-Quellcode. Die direkte Kommunikation wird dabei durch die gedachten Funktionen „SEND“ und „RECEIVE“ verdeckt. Beiden Funktionen wird ein Zeiger auf das erste Character eines Arrays und die Anzahl der zu lesender beziehungsweise zu schreibender Characters übergeben.

2. Bestimmen Sie, welche Schritte nötig sind, um nach dem Strom einschalten die Speisung für einen Teilnehmer zu aktivieren und um Sprachdaten über einen PCM-Bus zu übertragen?

Schreiben Sie auch dafür einen entsprechenden Quelltext.

3. Auf einem 4-Kanal-Speicheroszilloskop soll ein gesamter PCM-Rahmen einschließlich zugehöriger Takte dargestellt werden. Legen Sie in einer Skizze fest, welche Leitungen Sie anzeigen und welche Einstellungen Sie bezüglich Amplitude und Zeitbasis vorsehen müssen.

Am Versuch werden Sie an einem Stecker alle benötigten Signale vorfinden.

4. Wenn Sie von der Vermittlungshardware über die Interfaces Daten abholen, haben diese eine festliegende Struktur:

```
- Startbyte      =  %x81
- Cmd|Resp      =  %x00 ... %xFF
- Parameter 1   =  %x00 ... %xFF
- Parameter 2   =  %x00 ... %xFF
- Referenz      =  %x00 ... %xFF
- Stoppbyte     =  %xFE
```

Beim Lesen kann es aber passieren, dass Meldungen verfälscht sind, Start- oder Stoppbyte fehlen oder Meldungen noch nicht komplett übertragen wurden. Es könnte also vorkommen, dass in einem Lesezyklus von einer Meldung zum Beispiel nur das Startbyte enthalten ist und die restlichen Bytes im nächsten Zyklus von der Vermittlungshardware geholt werden.

Entwickeln Sie eine Software, die aus einem Character-String Messages der oben gezeigten Struktur zerlegt und dabei fehlerhafte Meldungen (Start- oder Stoppbyte beziehungsweise wenn andere Bytes fehlen) erkennt.

4.2 Versuchsdurchführung - Grundaufgaben

Schreiben und testen Sie ein Programm mit folgenden Aufgaben und Aussehen:

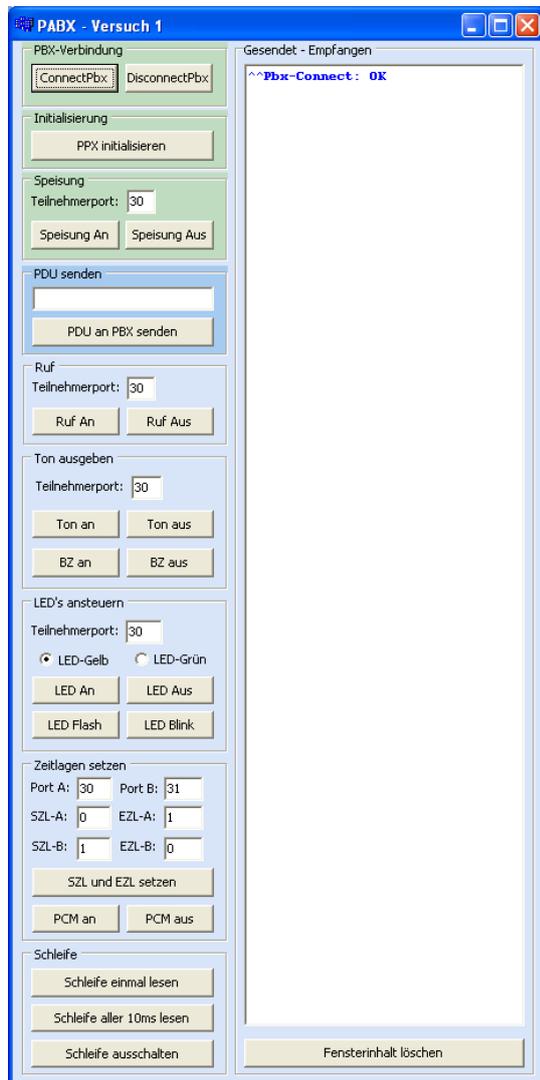


Abbildung 10: Anwendung für Versuch PABX1

1. Nehmen Sie die Basis-Initialisierung der Vermittlungshardware inklusive der Takterzeugung vor.
2. Initialisieren Sie die Teilnehmersätze %x30 bis %x33, schalten Sie die Speisung ein und als Quittung die grüne LED an, wobei die gelbe LED aus sein soll.
3. Aktivieren Sie den Ruf für einen beliebigen Teilnehmer und lassen Sie bei diesem Teilnehmer während des Rufes die gelbe LED langsam blinken.
4. Sorgen Sie dafür, dass die Speisung der Teilnehmer eingeschaltet ist. Messen Sie mittels eines Voltmeters die Spannungen auf der Teilnehmerleitung bei aufliegendem Hörer (On-Hook) und bei abgenommenem Hörer (Off-Hook).
5. Geben Sie an einen beliebigen Teilnehmer ein Besetztzeichen aus.

6. Stellen Sie zwei beliebige Teilnehmersätze so ein, dass einer auf Zeitlage 0 und der andere auf Zeitlage 2 sendet. Stellen Sie PCM-Highway und die entsprechenden Takte mittels Oszilloskop dar.
7. Stellen Sie einen weiteren Teilnehmer so ein, dass seine Empfangszeitlage 0 ist, er also die Sendedaten des Teilnehmers empfängt, der auf Zeitlage 0 sendet. Prüfen Sie, ob diese Simplexverbindung funktioniert. Ist dies der Fall, realisieren Sie für diese beiden Teilnehmer eine Duplexverbindung.

4.3 Versuchsdurchführung - Wahlbewertung

Schreiben Sie ein Programm zur Wahlbewertung für einen Teilnehmer.

1. Nehmen Sie alle notwendigen Initialisierungen des SLICs vor.
2. Programmieren Sie die Interface-Schaltung so, dass diese aller 10ms das Schleifensignal überträgt.
3. Implementieren Sie den unter 4.1 – 4 entwickelten Algorithmus zur Auswertung der von der PBX gesendeten Meldungen. Nutzen Sie zum Erhalt des Datenstromes von der PBX die Ereignisfunktion OnPbxRead().
4. Die Meldungen bestehen wie bereits erläutert wieder aus 6 Bytes.

```
%x81 %x53 %x01 <Schleifenprobe> <Referenz> %xFE
```

In dem Byte Schleifenprobe können die Schleifenzustände von acht Teilnehmern abgebildet werden. Im hier vorliegenden Fall sind nur die unteren vier Bits relevant.

Bit 0 = Schleife Teilnehmer %x30, Wert = 0 TIn. passiv, Wert = 1 TIn. aktiv

Bit 1 = Schleife Teilnehmer %x31, Wert = 0 TIn. passiv, Wert = 1 TIn. aktiv

Bit 2 = Schleife Teilnehmer %x32, ...

Bit 3 = Schleife Teilnehmer %x33, ...

Schreiben Sie eine Automatensoftware entsprechend der Vorlesung „Softwareversorgung analoger Teilnehmersätze“ womit die Zustands- und Wahlbewertung realisiert wird. Ihre Software soll die 10-ms-Schleifenproben so verarbeiten, dass die vermittlungstechnischen Ereignisse (Teilnehmer aktiv,passiv, wzi0...wzi9) für die Call-Control erzeugt werden.