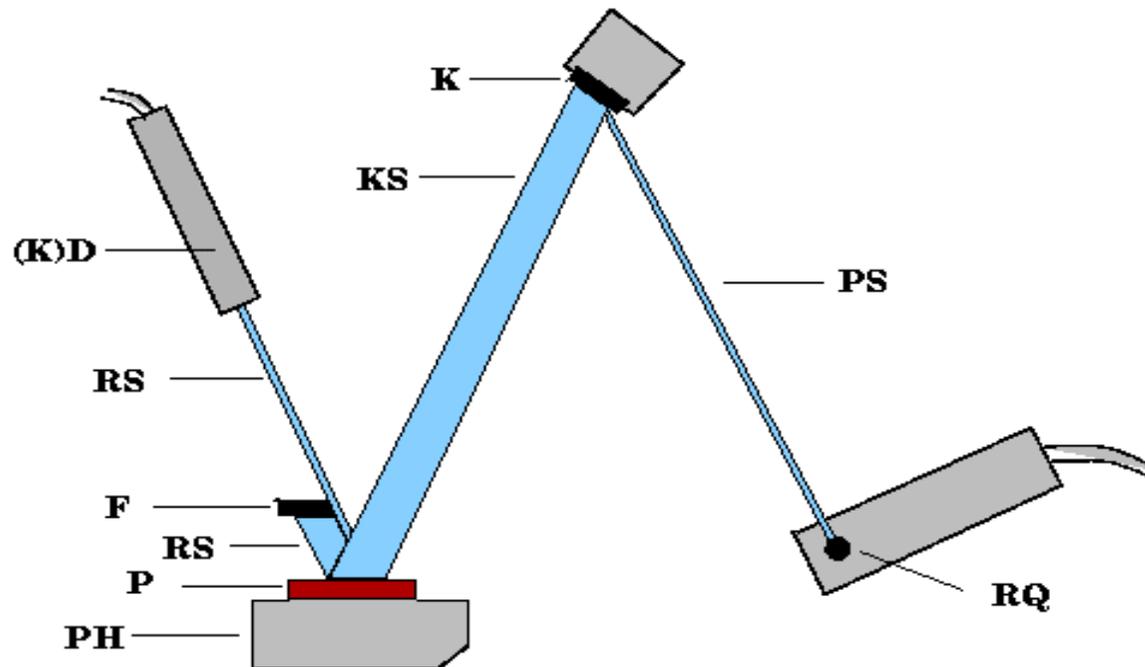


# XCTL - Portierung

# Einleitung

- XCTL – X-ray Control
- Einsatz im Institut für Physik der HU-Berlin
- Steuerung von Motoren/Detektoren an Labor-Messplätzen zur Untersuchung von kristallinen Halbleiterstrukturen

# Einleitung



**Abb.: Schema der Zwei-Kristall-Topographiekamera**

**RQ: Roentgenquelle; PS: Primaerstrahl; K: Kollimator;  
KS: kollimierter Strahl; PH: Probenhalter; P: Probe;  
RS: reflektierter Strahl; F: Fotoplatte; (K)D: (Kontroll-)Detektor**

# Motivation – Portierung nach Windows NT-Plattform

- technologischer Fortschritt
  - zeitgemäße Erweiterbarkeit der Messplätze
  - Nutzung moderner 32-Bit Betriebssystemarchitekturen
  - Verbesserung der grafischen Nutzerschnittstelle
- Support durch Microsoft erst wieder ab Windows 2000

# Motivation – Portierung nach Microsoft Visual Studio 6

- Breitere Toolunterstützung
- Subjektiv bessere Entwicklungsumgebung
  - Debugger
  - Versionsverwaltung

# Ist - Zustand

- Hardware
- Betriebssystem
- Entwicklungsumgebung
- XCTL

# Hardware – Ist-Zustand

- Messplatzhardware (Basis)
  - Steuerungs-PC
  - Motorsteuerkarten & Präzisionsmotoren
  - Detektorsteuerkarten & Röntgendetektoren

# Hardware – Ist-Zustand

- Steuerungs-PC
  - Intel/AMD 486er
  - 8 bis 32 MByte Hauptspeicher
  - Festplattenkapazität zwischen 100 MB und 2 GB
  - ISA-Erweiterungsbus

# Hardware – Ist-Zustand

- Motorsteuerkarten
  - C-812
    - programmierbarer Mikrocontroller
    - Ansteuerung von bis zu 4 Motoren
    - 8-Bit ISA-Schnittstelle
    - IEEE488-Schnittstelle

# Hardware – Ist-Zustand

- Motorsteuerkarte
  - C-832
    - 2 spezielle, nicht programmierbare Mikrocontroller (ASIC)
    - Ansteuerung von max. 2 Motoren
    - 16-Bit ISA-Schnittstelle

# Hardware – Ist-Zustand

- Detektoren
  - 0-dimensional
    - Zu einem Zeitpunkt ein einzelner, positionsunabhängiger Messwert
  - 1-dimensional
    - Zu einem Zeitpunkt eine Menge von positionsabhängigen Messwerten (ein Messwert je Kanal)
  - 2-dimensional
    - Zu einem Zeitpunkt positionsabhängige Messwerte einer Fläche

# Hardware – Ist-Zustand

- 0-dimensional
  - Radicon SCSCS (Single Channel Scintillation Crystal Spectrometer)
    - 8-Bit ISA, programmierbarer Mikrocontroller
  - Russischer SCSCS (BDS-6-06)
    - generische Controllerkarte AX5216/TAm9513
- 1-dimensional
  - Braun PSD (Position Sensitive Detector)
    - 8-Bit ISA, programmierbarer Mikrocontroller
  - Stoe PSD (eigenes Controllersystem)

# Hardware – Ist-Zustand

- 2-dimensional
  - Proscan CCD-Kamera (eigener Controller HSSC-1)
    - PCI, DSPs

# Betriebssystem – Ist-Zustand

- Windows 3.11
  - 16-Bit Ausführungsumgebung
  - Grafische Nutzeroberfläche (max. 256 Farben/gerastert)
  - Kooperatives Multitasking
  - Segmentierte Speicheradressierung
  - DLL-Instanzen nutzen gemeinsamen Speicher
  - Hardwarezugriffe direkt möglich (rudimentäres Treiberkonzept)
  - Keine Verwaltung von Hardwareressourcen

# Entwicklungsumgebung – Ist-Zustand

- Borland C++ 4.5
  - 16-Bit Compiler/Linker/Debugger
  - Borland-spezifische Sprachelemente/  
Bibliotheksfunktionen
  - Borland-spezifischer Ressourceneditor

# Entwicklungsumgebung – Ist-Zustand

- Borland C++ 5.02
  - 32-Bit Compiler/Linker/Debugger
  - 16-Bit Compiler/Linker
  - Borland-spezifische Sprachelemente/  
Bibliotheksfunktionen
  - Borland-spezifischer Ressourceneditor

# XCTL – Ist-Zustand

- 16-Bit Applikation
- ca.140 Dateien
- ca. 56000 LOC (ca. 16000 LOCmt)
- 7 Module
- 14 Subsysteme

# XCTL - Subsysteme

- Motorsteuerung
  - logische Abstraktionsschicht zur Ansteuerung und Kontrolle der Motoren des Messplatzes
- Detektornutzung
  - logische Abstraktionsschicht zur Ansteuerung und Kontrolle der Detektoren des Messplatzes
- Repäsentation und Darstellung der Messdaten
  - Funktionen/Strukturen für Datenhaltung und -darstellungen

# XCTL - Subsysteme

- Topographie
  - Funktionalität/Oberfläche des Messvorganges  
Topographie
- Diffraktometrie/Reflektometrie
  - Funktionalität/Oberfläche der Messvorgänge  
Diffraktometrie/Reflektometrie
- Ablaufsteuerung
  - Funktionen zur Steuerung des Messvorgangs über eine Skriptsprache und Funktionen zur Messungsvorbereitung (z.B. manuelle Justage)

# XCTL - Subsysteme

- Online-Hilfe
- Nutzerinteraktion
- Interne Funktionalität & allg. Definitionen
  - Programmrumppf
- Automatische Justage
  - Funktionen zur automatischen Probenjustierung

# XCTL

- Windows-Ressourcen
- Protokollbuch
  - Elektronisches Messprotokoll
- Allgemeine Einstellungen
  - Erfassung/Verwaltung der allgemeinen Messungseinstellungen (Xcontrol.ini)
- Hardwareansteuerung
  - Funktionen für den Zugriff auf Hardwarekomponenten

# XCTL – Ist-Zustand

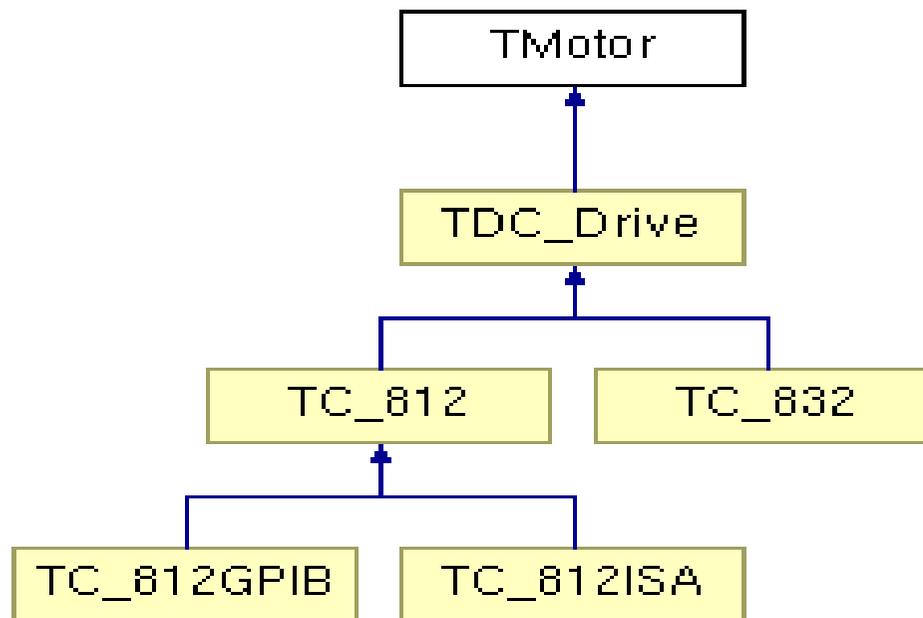
- Arten von Hardwarezugriffen
  - Port I/O
    - Kommunikation über speziellen I/O-Bus
    - Register der Hardware werden an Ports/Adressen dieses Busses gebunden
    - 16-Bit Adressen, 8 Bit Datenbreite
    - Lesen und Schreiben
  - Memory mapped I/O
    - Kommunikation über ‘normale‘ Speicheradressen
    - Register der Hardware werden auf Speicheradressen gemappt
    - 16-Bit Adressen (Segment:Offset)
    - Lesen und Schreiben

# XCTL – Ist-Zustand

- Arten von Hardwarezugriffen
  - IEEE 488
    - Externer Bus
    - 8-Bit
    - Ähnlich LPT-Schnittstelle

# XCTL – Ist-Zustand

- Klassenstruktur Motoren



# XCTL – Ist-Zustand

- Hardwarezugriff Motoren

- C-812

- Memory mapped I/O

```
char* Register = 0xD000;  
char Value = *Register; //lesen  
*Register = Value; //schreiben
```

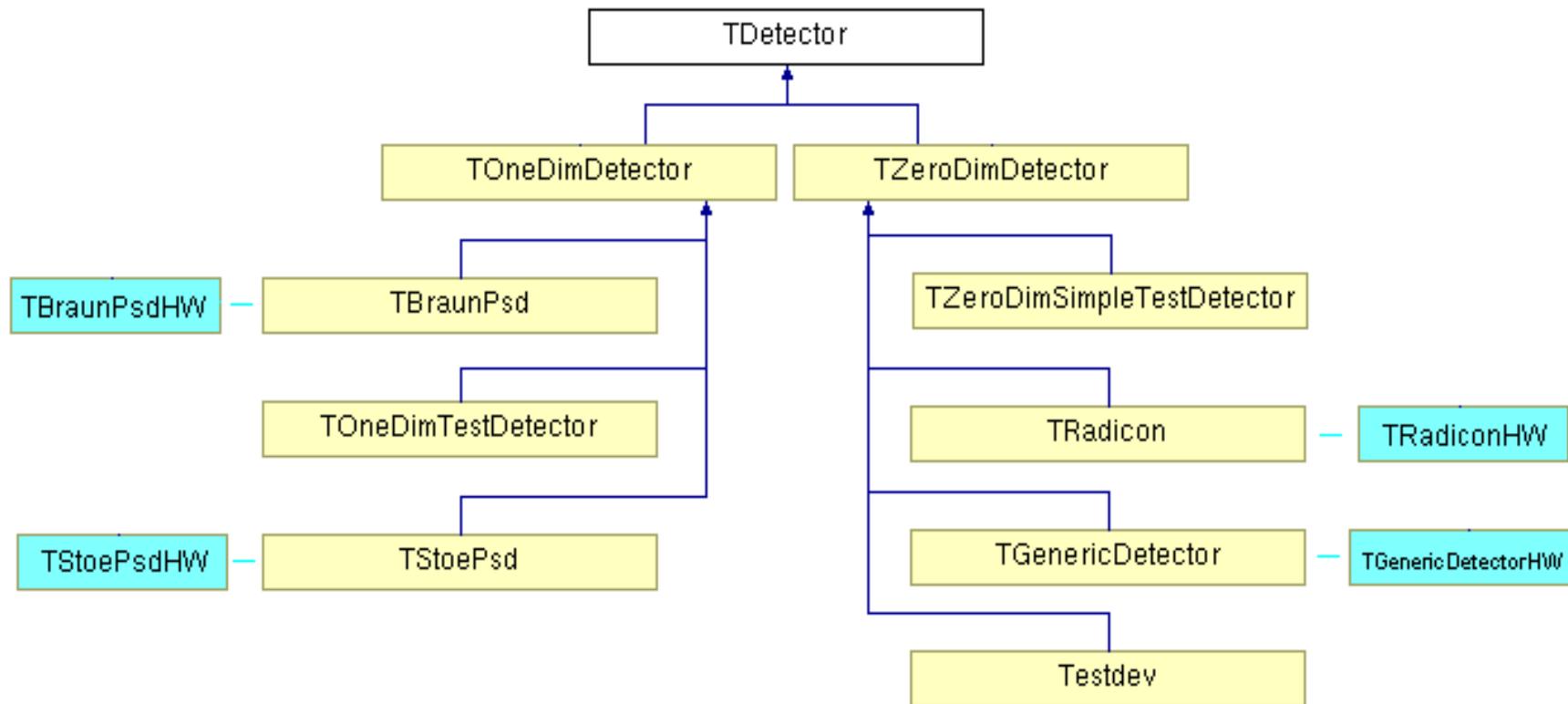
- C-832

- Port I/O

```
int Value = inp(Port); //lesen  
outp(Port, Value); //schreiben
```

# XCTL – Ist-Zustand

- Klassenstruktur Detektoren



# Soll - Zustand

- Hardware
- Betriebssystem
- Entwicklungsumgebung
- XCTL

# Hardware – Soll-Zustand

- Detektoren und Motoren unverändert zu Ist-Zustand
- Steuerungs-PCs werden aufgerüstet
  - min. Pentium 133
  - min. 128 MByte Hauptspeicher
  - Festplattenkapazität durchschnittlich 6 GB
  - min. 3 ISA-Slots

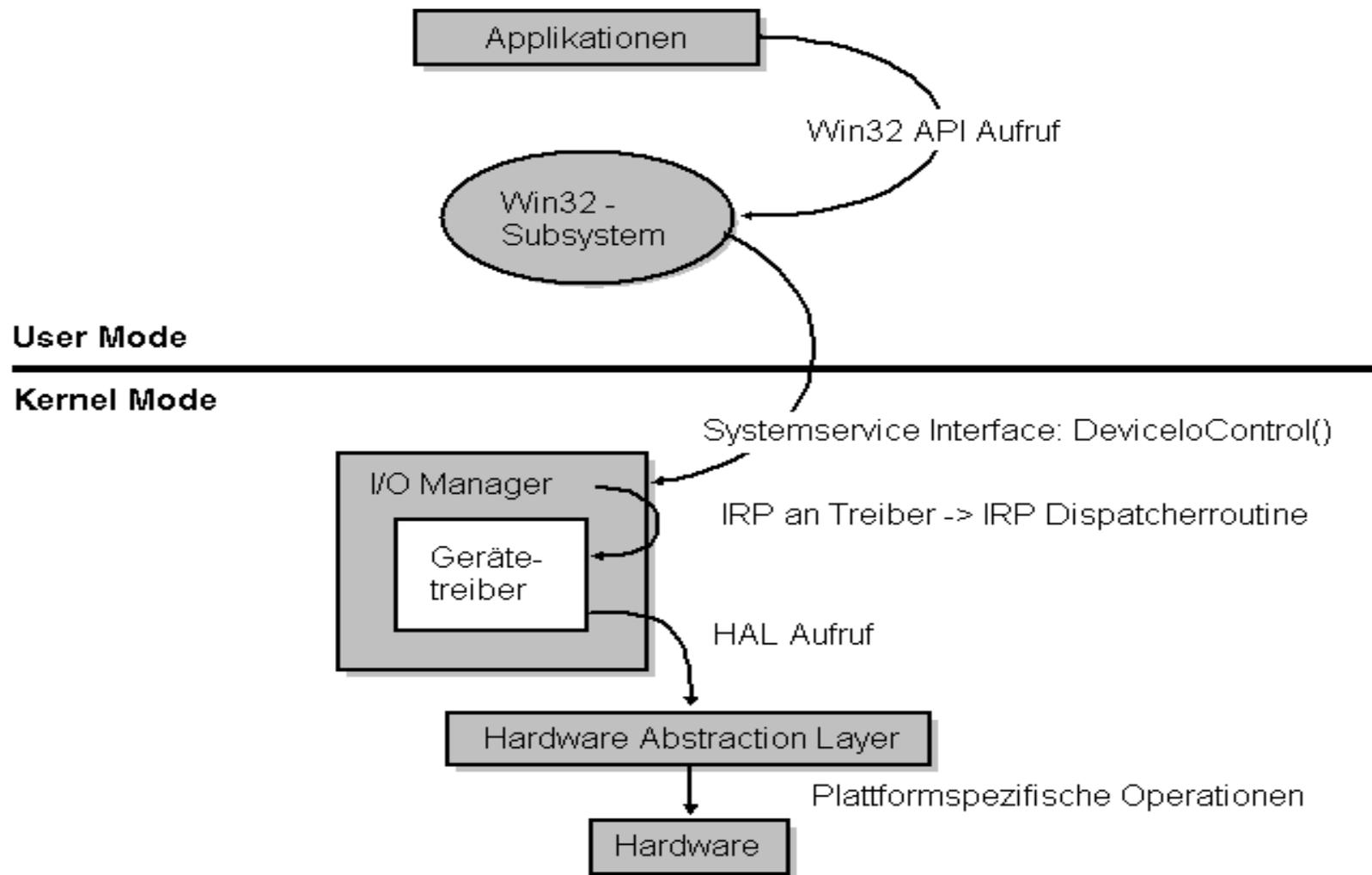
# Betriebssystem – Soll-Zustand

- Windows 2000
  - Echte 32-Bit Ausführungsumgebung
  - Grafische Nutzeroberfläche (bis 32-Bit Farbtiefe)
  - Preemptives Multitasking
  - Lineare Speicheradressierung
  - Speicherschutz/DLL-Instanzen nutzen getrennten Speicher
  - Hardwarezugriffe nicht direkt möglich (nur für privilegierte Software/Treiber)
  - Zentrale Verwaltung der Hardwareressourcen
  - Nutzerprofile mit Rechteverwaltung

# Betriebssystem – Soll-Zustand

- Möglichkeiten für Hardwarezugriffe unter Windows 2000
  - Alle Ports freischalten
    - Ausschalten der Sicherheitsmechanismen
  - Generischer Porttreiber
    - Memory mapped I/O ???
  - WDM-Treiber
    - W2k - konform

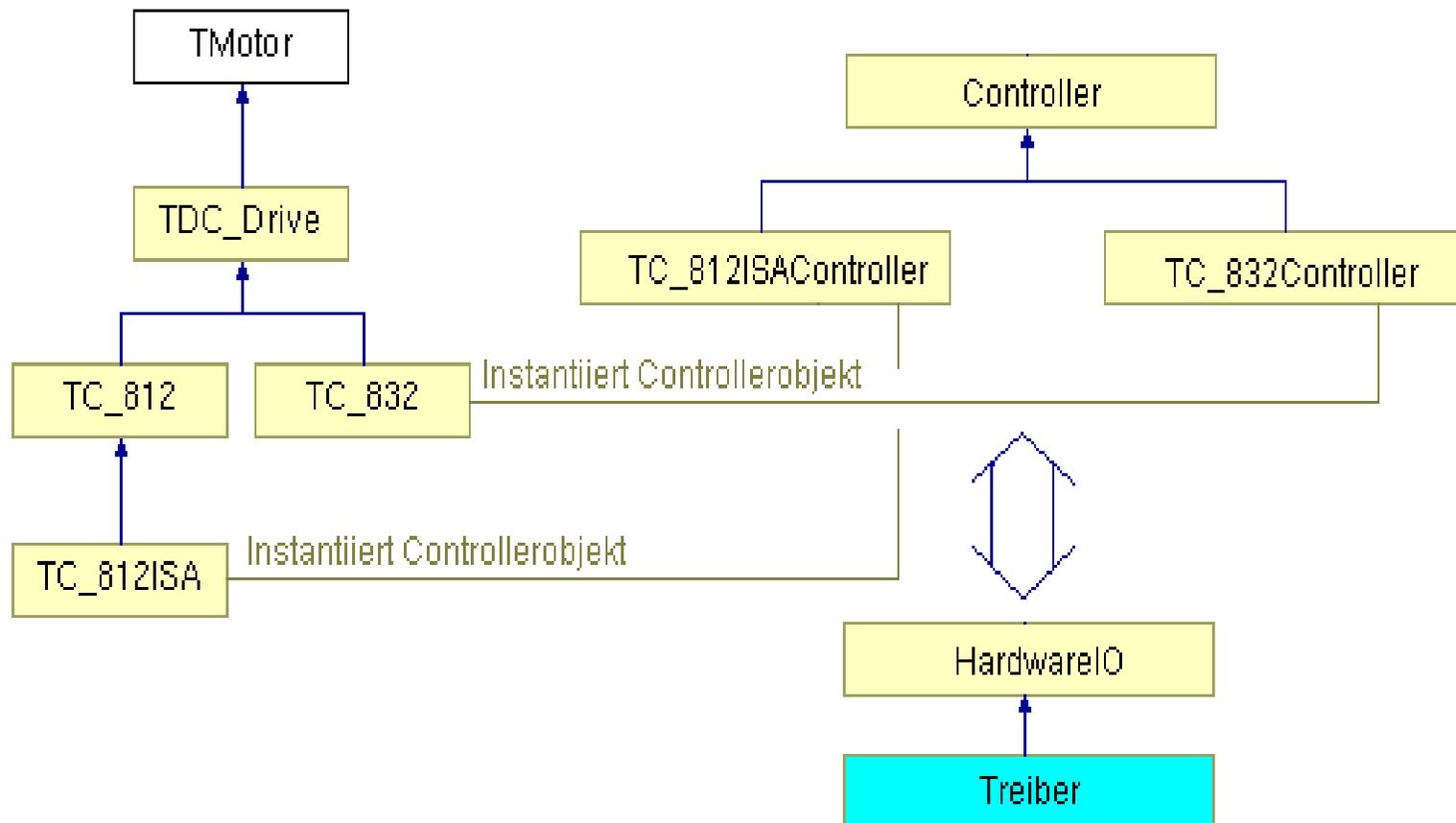
# Betriebssystem – Soll-System



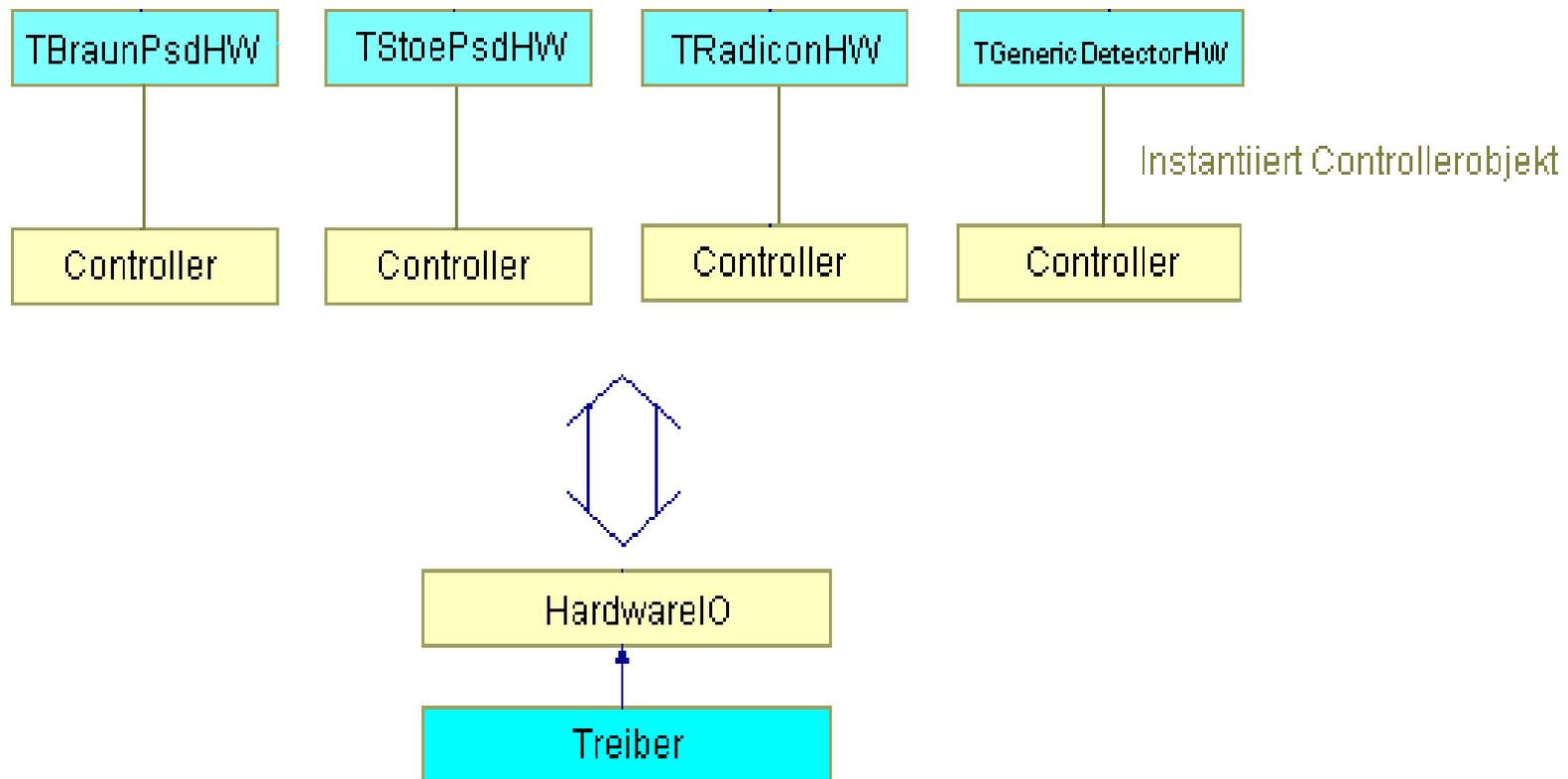
# Entwicklungsumgebung – Soll-Zustand

- Microsoft Visual Studio (C++) 6
  - 32-Bit Compiler/Linker/Debugger
  - Microsoft-spezifische Sprachelemente/  
Bibliotheksfunktionen
  - Microsoft-spezifischer Ressourceneditor
  - Breite Toolunterstützung
  - Komfortable Entwicklungsumgebung

# XCTL – Soll-Zustand



# XCTL – Soll-Zustand



# Treiber – Prinzipieller Aufbau

- DriverEntry(...)
  - {
    - Treiber Einstiegspunkt
    - Initialisierung sämtlicher Verwaltungsfunktionen,  
da DriverEntry einzige Funktion ist die im Namen festgelegt ist.
    - IOCTL Anforderungen bearbeiten - *McDispatchDeviceControl*;
    - IRP Anforderungen bearbeiten - *McDispatchPNP*;
    - Treiber entladen - *McUnload*;
    - Gerät hinzufügen - *McAddDevice*;
  - }

# Treiber – Prinzipieller Aufbau

- McAddDevice(...)
  - { ...
  - ein neues Gerät anlegen
  - den geräteeigenen Speicher initialisieren
  - das Kommunikationsinterface registrieren
  - das Gerät an den Gerätestack anfügen
  - ...
  - }

# Treiber – Prinzipieller Aufbau

- McDispatchPNP(...)
  - {
    - ...
    - Auswertung des IRP-Codes der vom I/O-Manager gesendet wurde
    - Steuerung des Treiberhaltens im System
      - Powemanagement
      - Start
      - Stop
      - Gerät entfernen
    - ...
  - }

# Treiber – Prinzipieller Aufbau

- McDispatchDeviceControl(...)
  - {
    - ...
    - eigentliche Treiberfunktionalität
    - den angeforderten IOCTL-Code prüfen und entsprechend behandeln z.B.
      - Byte/Word lesen/schreiben
      - ID der Treiberinstanz (Controller-ID) übermitteln
    - ...

# Vor/Nachteile

- Vorteile
  - Beibehaltung der Sicherheitsmechanismen
  - leichte Erweiterbarkeit
  - Wiederverwendbarkeit durch andere Komponenten
  - Vereinfachung der Hardwaresimulation (Testtreiber)
- Nachteile
  - Einarbeitungsaufwand
  - Relativ aufwendig in Implementation und Test