

Videos zu einigen SwDev-Themen ( OOProg, Threads and Forms . . . ) befinden sich  
auf mbed\_hl\ Swdev\_Tinf\_Videos  
und auf swdevhl\SwDev5te\_1617\Videos  
und auf swdevhl\SwDev5te\_1617\Videos1819

Liste der FTP-Verzeichnisse:

|            |            |
|------------|------------|
| mbed_hl    | robo       |
| bulgati_hl | robo       |
| swdevhl    | swdevhlpwd |

Lehrerbox\HL wird mit LB abgekürzt

Grundlagen zu Pointern und dynamischer Speicherverwaltung  
mbed\_hl\SwDev5te\doc \Pointer  
swdevhl\SwDev5te\_1617\DataStruct

**Gallileo Computing C von A bis Z**

Kap. 12 Zeiger und Kap. 14 Dynamische Speicherverwaltung  
Das Buch ist im Internet frei verfügbar  
Liegt auf swdevhl\SwDev5te\_1617\DataStruct

## Concurrent Programming 1

Erstellen Sie eine multithreaded Forms-Anwendung.

Durch ButtonClick sollen von der Form über eine Queue an einen WorkerThread Aufträge vergeben werden. Der WorkerThread soll die Erledigung der Aufträge über Windows-Messages an die Form zurückmelden

1. Aufgabe  
Erklären Sie die Aufgabenstellung und das Design mit einem UML-Diagramm.
2. Aufgabe  
Programmieren Sie die Forms-Anwendung.
3. Aufgabe  
Welche Teile des UML-Diagramm's wurden mit welchen Code-Teilen realisiert.  
Das Diag. Aus meiner Mitschrift

## Concurrent Programming 2

Programmieren Sie die Datenübertragung zwischen 2 C#Threads über einen begrenzten Puffer.

1. Aufgabe  
Erklären Sie die Aufgabenstellung und das Design mit einem UML-Diagramm.
2. Aufgabe  
Programmieren Sie die Datenübertragung
3. Aufgabe  
Erklären Sie die folgenden Fälle im Code und im UML-Diagramm  
- Put() und Get() wenn der Puffer zur Hälfte gefüllt ist.

- `Get()` für einen vollen Puffer

Das ist Producer Consumer

Theorie ist das gesamte *ConcurrentProgramming\_hl.doc* *ConcProcMts.pdf*

### Concurrent Programming 3

Schreiben Sie ein Programm zum Datenaustausch zwischen 2 C#-Threads über eine Mailbox Request / Response

1. Aufgabe  
Erklären Sie die Aufgabenstellung und das Design mit einem UML-Diagramm.
2. Aufgabe  
Programmieren Sie die Datenübertragung.
3. Aufgabe  
Welche Teile des UML-Diagramm's wurden mit welchen Code-Teilen realisiert.

### Concurrent Programming 4

Schreiben Sie einen multithreaded TCP/IP Server

1. Aufgabe  
Erklären Sie die Aufgabenstellung und das Design mit einem UML-Diagramm.
2. Aufgabe  
Programmieren Sie den Server und einen Test-Client
3. Aufgabe  
Warum lässt sich ein Server der dynamisch mit mehrere Clients kommuniziert am besten mit Threads realisieren.

### Echtzeitprogrammierung 1 Pri2

Schreiben Sie eine Ampelsteuerung mithilfe des State-Machine-Entwurfsmusters.

Die Ampel hat die Zustände ROT, GELB und GRÜN.

Das Umschalten zwischen den Zuständen erfolgt Zeitgesteuert mit den im Zustandsdiagramm angegebenen Umschaltzeiten.

Wird der Fußgänger-Knopf ( Btn2 ) gedrückt so wechselt die Ampel zunächst in den Zustand GRÜN-BLINKEN und dann nach ROT.

- a. Aufgabe  
Implementieren Sie die Ampel gemäß Zustandsdiagramm und Spezifikation
- b. Aufgabe  
Zeichnen Sie die Skizze des Zustandsdiagramms neu und erklären Sie jedes Syntaxelement des Zustandsdiagramm's und den dazupassenden Code der Implementierung.

## Echtzeitprogrammierung 2 Pri2

Die verschiedenen Blinkmuster einer Fahrradleuchte sollen mithilfe einer StateMachine programmiert werden.

Die Fahrradleuchte soll die folgenden Blinkmuster anzeigen:

- Blinken mit 2Hz
- Lauflicht links mit 5Hz
- Lauflicht rechts mit 10Hz
- Alle LEDs aus

1) Aufgabe

Zeichnen Sie das State-Diagramm zur Steuerung der Fahrradleuchte mit 2 Buttons.

2) Aufgabe

Programmieren Sie die Fahrradleuchte und erklären Sie welche Codeteile mit welchen Abschnitten des State-Diagramm's korrespondieren

## Echtzeitprogrammierung 3 Pri1

Entwerfen Sie ein Datenübertragungsprotokoll zur Anzeige von Mess und Regelgrößen aus einem embedded System auf einem PC.

1. Aufgabe

Entwerfen Sie das Datenübertragungsprotokoll und diskutieren Sie verschiedene Eigenschaften wie Flusskontrolle, Effizienz, Ausfallssicherheit . . .

2. Aufgabe

Beschreiben Sie mit einem Flussdiagramm den typischen Aufbau einer Robotik Anwendung, welche das Monitoring der Prozessdaten enthält.

3. Aufgabe

Schreiben Sie ein kurzes Testprogramm, welches die Verwendung der Prozessdatenübertragung demonstriert. Ein paar werte übertragen main()-loop uC

## Echtzeitprogrammierung 4 Pri1

Schreiben Sie C++ Funktionen um mehrere Messkanäle mit unterschiedlichen Datentypen über einen Stream ( z.B. serielle Verbindung ) übertragen zu können.

1. Aufgabe

Erklären Sie das Datenübertragungsprotokoll, welches in den C++ Funktionen verwendet wird.

2. Aufgabe

Schreiben Sie die C++ Funktionen zur Datenübertragung für den uC

3. Aufgabe

Erklären Sie mit einer Skizze das Layout der Datentypen **int32**, **float**, **string** auf dem Bytestrom

## Datenstrukturen 1

Programmieren Sie für eine LinkedList die Funktionen **AddHead()** und **Find()**

1. Aufgabe  
Erklären Sie die Funktionsweise von **AddHead()** und **Find()** mithilfe der Skizze einer LinkedList.
2. Aufgabe  
Programmieren Sie die Funktionen **AddHead()** und **Find()**.
3. Aufgabe  
Schreiben Sie einen Testcode um das richtige Funktionieren von **AddHead()** und **Find()** zu verifizieren
4. Aufgabe  
Warum ist DynArray von `IHLContainer` abgeleitet.  
Erklären Sie die Funktionsweise des *IHLContainer-Interface*.

## Datenstrukturen 2

Programmieren Sie für ein dynamisches Array die Funktionen **AddHead()** und **Find()**

1. Aufgabe  
Erklären Sie die Funktionsweise von **AddHead()** und **Find()** mithilfe der Skizze des dynamischen Arrays.
2. Aufgabe  
Programmieren Sie die Funktionen **AddHead()** und **Find()**.
3. Aufgabe  
Schreiben Sie einen Testcode um das richtige Funktionieren von **AddHead()** und **Find()** zu verifizieren
4. Aufgabe  
Warum ist DynArray von `IHLContainer` abgeleitet.  
Erklären Sie die Funktionsweise des *IHLContainer-Interface*.

## Datenstrukturen 3

Programmieren Sie für ein dynamisches Array die Funktionen **InsertSorted()** und **At()**

1. Aufgabe  
Erklären Sie die Funktionsweise von **InsertSorted()** und **At()** mithilfe der Skizze des dynamischen Arrays.
2. Aufgabe  
Programmieren Sie die Funktionen **InsertSorted()** und **At()**.
3. Aufgabe  
Schreiben Sie einen Testcode um das richtige Funktionieren von **InsertSorted()** und **At()** zu verifizieren

4. Aufgabe  
Wie ist es möglich das **InsertSorted()** für die unterschiedlichsten Objekte mit den unterschiedlichsten Sortierkriterien verwendet werden kann.

## Datenstrukturen 4

Programmieren Sie für eine LinkedList die Funktionen **InsertSorted()** und **At()**

1. Aufgabe  
Erklären Sie die Funktionsweise von **InsertSorted()** und **At()** mithilfe der Skizze der LinkedList.
2. Aufgabe  
Programmieren Sie die Funktionen **InsertSorted()** und **At()**.
3. Aufgabe  
Schreiben Sie einen Testcode um das richtige Funktionieren von **InsertSorted()** und **At()** zu verifizieren
4. Aufgabe  
Wie ist es möglich das **InsertSorted()** für die unterschiedlichsten Objekte mit den unterschiedlichsten Sortierkriterien verwendet werden kann.

## Modulare Programmierung 1

C# beinhaltet keine Unterstützung für komplexe Zahlen.

Schreiben Sie daher eine Library für das Rechnen mit komplexen Zahlen

1. Aufgabe  
Programmieren Sie die Complex-Klasse mit den Operationen +,-,\* und schreiben Sie ein geeignetes Testcoding um die Operationen zu testen.
2. Aufgabe  
Schreiben Sie eine Funktion, die es ermöglicht Complex-Objekte mit Console.WriteLine() auszugeben ( Rechteck und Polarkoordinaten). Durch welchen C#-Mechanismus wird dies ermöglicht.
3. Aufgabe  
Schreiben Sie den Divisions-Operator, indem sie die Mult-Methode sinnvoll verwenden.  
Vernünftigen Testcode für Mult  $1/45^\circ * 1/45^\circ$

## Modulare Programmierung 2

Schreiben Sie eine Klasse zur Verwaltung von int Zahlen auf einem Stack.

```
class Stack
{
    protected int[] _ary;
    protected int _sp;

    public Stack17(int aSize)

    // throws StackException
    public int Pop()

    // throws StackException
```

```

    public void Push(int aVal)
}

```

### 1. Aufgabe

Programmieren Sie die Stack-Klasse mit den Operationen Push(), Pop() und Print(). Schreiben Sie ein geeignetes Testcoding um die Operationen zu testen.

### 2. Aufgabe

Implementieren Sie Exception-Handling um bei Overflow und Underflow des Stacks den Benutzer des Stack-Objektes benachrichtigen zu können.

Wozu braucht man Exception-Handling.

Wie könnte man die Fehlerbehandlung ohne Exception-Handling lösen.

Welche Vor-/Nachteile hätte das.

### 3. Aufgabe

Schreiben Sie ein geeignetes Testcoding um das Exception-Handling zu testen.

## Modulare Programmierung 3

Schreiben Sie eine Klasse zur Verwaltung von beliebigen Datentypen in einer Queue

```

class Queue
{
    protected int[] _ary;
    protected int _wrIdx, _rdIdx;

    public Queue(int aSize)

    // throws QueueException
    public void Put(object aVal)

    // throws QueueException
    public object Get(int aVal)
}

```

### 1. Aufgabe

Programmieren Sie die Queue-Klasse mit den Operationen Put(), Get() und Print(). Schreiben Sie ein geeignetes Testcoding um die Operationen zu testen.

### 2. Aufgabe

Zeigen Sie mit einer Skizze wie die Daten in der Queue mit wrIdx und rdIdx verwaltet werden.

### 3. Aufgabe

Implementieren Sie Exception-Handling um bei Overflow und Underflow der Queue den Benutzer des Queue -Objektes benachrichtigen zu können.

Wozu braucht man Exception-Handling.

Wie könnte man die Fehlerbehandlung ohne Exception-Handling lösen.

Welche Vor-/Nachteile hätte das.

### 4. Aufgabe

Schreiben Sie ein geeignetes Testcoding um das Exception-Handling zu testen.

Nennen Sie 2 Anwendungen für Queues in der technischen Informatik.

## Modulare Programmierung 4

Schreiben Sie eine Library für 2D-Vektoroperationen, die für Grafik und Spielprogramme verwendet werden kann.

```
Vect2D A(1,2), B(3,4), C;  
  
C = A*(A - B);  
  
WriteLine („A= B= C=“, A, B, C);  
  
public struct Vect2D  
{  
    public Vect2D(double aX, double aY, bool aPolar)  
    public double VectLength()  
    public static Vect2D operator +(Vect2D aA, Vect2D aB)  
    public Vect2D GetNormalVector()  
}
```

### 1. Aufgabe

Programmieren Sie die Vect2D-Klasse mit den Operationen +,-,\* und CoMul() schreiben Sie ein geeignetes Testcoding um die Operationen zu testen.

### 2. Aufgabe

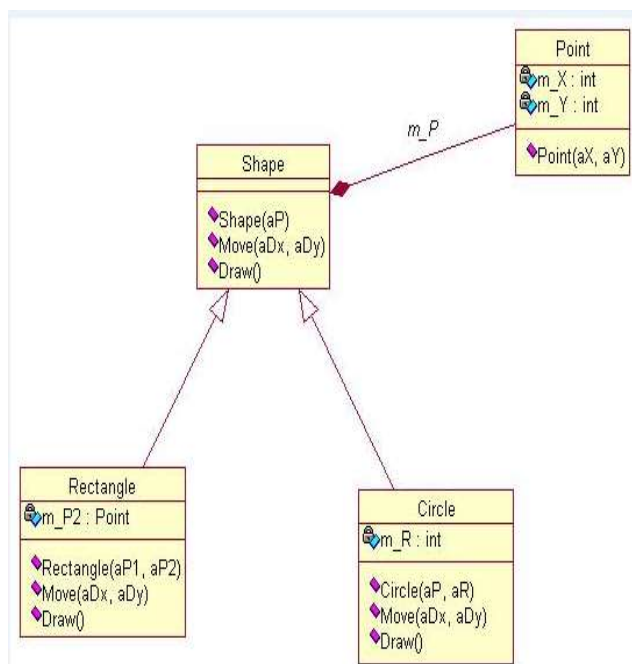
Zeigen Sie wie man CoMul() verwenden kann um Vektoren zu drehen.  
Wie wird das Drehen eines Vektors in einem 2D-Spiel ( Physiksimulation ) verwendet um Spielobjekte ( Auto, Flugzeug, Roboter . . . ) zu drehen.

### 3. Aufgabe

Schreiben und testen Sie eine Methode zur Berechnung des Einheitsvektors.

## Objektorientierte Programmierung 1

Entwerfen Sie eine Klassenhierarchie zur Repräsentation von Grafikobjekten in einem einfachen CAD-Programm.



1. Aufgabe  
Entwerfen Sie die Klassenhierarchie mit den Mitteln der objektorientierter Programmierung
2. Aufgabe  
Testen und erklären Sie das Erzeugen, und Zeichnen verschiedener Shape-Objekte.
3. Aufgabe  
Erklären Sie die folgenden objektorientierten Mechanismen anhand des Shape-Beispiels  
Konstruktor, Vererbung, Polymorphie ( virt. Funktionen ), Properties

## Objektorientierte Programmierung 2

Implementieren Sie das Erzeugen und Löschen von Grafikobjekten in einem einfachen CAD-Programm

1. Aufgabe  
Verschiedene Grafikobjekte ( Rectangle, Circle . . . ) sollen durch Mausklick erzeugt werden
2. Aufgabe  
Implementieren Sie das Löschen von Grafikobjekten.  
Der Code für das Löschen soll zum Großteil im Framework und in der Basisklasse angesiedelt sein.
3. Aufgabe  
Erklären Sie die folgenden objektorientierten Mechanismen anhand der Grafikobjekt  
Klassenhierarchie Konstruktor, Vererbung, Polymorphie.  
Zeigen Sie warum es mithilfe der objektorientierten Programmierung sehr einfach ist neue  
Grafikobjekte zum Minicad-Framework hinzuzufügen.

## Objektorientierte Programmierung 3

Implementieren Sie das Verschieben von Grafikobjekten in einem einfachen CAD-Programm.

1. Aufgabe  
Verschiedene Grafikobjekte ( Rectangle, Circle . . . ) sollen durch Mausklick erzeugt werden
2. Aufgabe  
Implementieren Sie das Verschieben von Grafikobjekten  
Der Code für das Verschieben soll zum Großteil im Framework und in der Basisklasse  
angesiedelt sein.
3. Aufgabe  
Erklären Sie die folgenden objektorientierten Mechanismen anhand der Grafikobjekt  
Klassenhierarchie Konstruktor, Vererbung, Polymorphie.  
Zeigen Sie warum es mithilfe der objektorientierten Programmierung sehr einfach ist neue  
Grafikobjekte zum Minicad-Framework hinzuzufügen.

## Objektorientierte Programmierung 4

Erweitern Sie das MiniCad-Framework um eine Klasse zur Darstellung von Dreiecken.

1. Aufgabe  
Verschiedene Grafikobjekte ( Rectangle, Circle . . . ) sollen durch Mausklick erzeugt werden



2. Aufgabe  
Erweitern Sie das MiniCad-Framework um eine Klasse zur Darstellung von Dreiecken.
3. Aufgabe  
Erklären Sie die folgenden objektorientierten Mechanismen anhand der Grafikobjekt Klassenhierarchie Konstruktor, Vererbung, Polymorphie.  
Zeigen Sie warum es mithilfe der objektorientierten Programmierung sehr einfach ist neue Grafikobjekte zum Minicad-Framework hinzuzufügen.

## Grafik und Spieleprogrammierung 1

Programmieren Sie eine realistische Simulation für einen Ball der sich unter dem Einfluss von Reibung und Gravitation bewegt.

1. Aufgabe  
Erstellen Sie die Bewegungsgleichung für einen Ball der sich unter dem Einfluss von Reibung und Gravitation bewegt.
2. Aufgabe  
Schreiben Sie ein C# Programm, welches die Bewegung des Balls simuliert.

## Grafik und Spieleprogrammierung 2

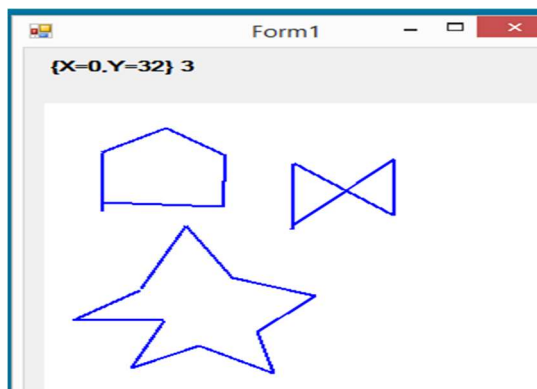
### Leitner

Programmieren Sie eine realistische Simulation für das Zusammenstoßen und Abprallen von 2 Billardkugeln für die Simulation eines Billardspiels.

1. Aufgabe  
Stellen Sie die Bewegungsgleichungen für den nichtzentralen Stoß 2er Kugeln mit unterschiedlicher Masse auf.
2. Aufgabe  
Schreiben Sie ein C# Programm, welches die Bewegung und den Stoß der 2 Kugeln simuliert

## Grafik und Spieleprogrammierung 3

Schreiben Sie einen einfachen Editor der einzelne aus Liniensegmenten zusammengesetzte Figuren erstellen kann.



1. Aufgabe  
Erstellen Sie den Editor und erklären Sie die dazu verwendeten Klassen und Datenstrukturen.

2. Aufgabe  
Implementieren Sie das Speichern und Laden der im Editor erstellten Zeichnungen

## Grafik und Spieleprogrammierung 4

Erstellen Sie eine Klasse für ein beliebiges aus Vektoren zusammengesetztes Objekt ( Roboter, Raumschiff . . . ) welches an jedem Punkt des Koordinatensystems in jeder rotierten Lage gezeichnet werden kann.

1. Aufgabe  
Programmieren und Testen Sie das Rotating-Graphic-Object.
2. Aufgabe  
Verwenden Sie das Rotating-Graphic-Object für die Simualtion eines Rennautos ( MarioCart )

## Pointer und Arrays 1

Schreiben Sie ein C++ Programm welches 2-Arrays von der Konsole einliest Array-A an Array-B anhängt und das Ergebniss auf der Konsole ausgibt.

1. Aufgabe  
Definieren Sie sinnvolle Funktionen, aus welchen das Programm zusammengesetzt wird. Die Funktionen sollen für Arrays mit beliebiger Länge verwendbar sein
2. Aufgabe  
Erklären Sie mithilfe einer Skizze die Funktionsweise der Funktion *AppendArray()*
3. Aufgabe  
Programmieren Sie die Arraybearbeitungs-Funktionen und und das Hauptprogramm

## Pointer und Arrays 2

Schreiben Sie ein C# Programm zum Aufbau, Iterieren und Löschen einer LinkedList

1. Aufgabe  
Definieren Sie eine Klasse, welche ein Element der LinkedList repräsentiert, und schreiben Sie die Funktionen `CreateList()`, `IterateList()`, `RemoveElements()`
2. Aufgabe  
Erklären Sie die Funktionsweise von `CreateList()`, `IterateList()`, `RemoveElements()` mit Skizzen.
3. Aufgabe  
Welche Änderungen sind, notwendig um das Programm in C++ umzuschreiben

## Pointer und Arrays 3

Schreiben Sie ein C++ Programm welches 2-Arrays von der Konsole einliest Array-A auf Array-B spiegelt und das Ergebniss auf der Konsole ausgibt.

1. Aufgabe  
Definieren Sie sinnvolle Funktionen, aus welchen das Programm zusammengesetzt wird.  
Die Funktionen sollen für Arrays mit beliebiger Länge verwendbar sein
2. Aufgabe  
Erklären Sie mithilfe einer Skizze die Funktionsweise der Funktion *MirrorArray()*
3. Aufgabe  
Programmieren Sie die Arraybearbeitungs-Funktionen und das Hauptprogramm

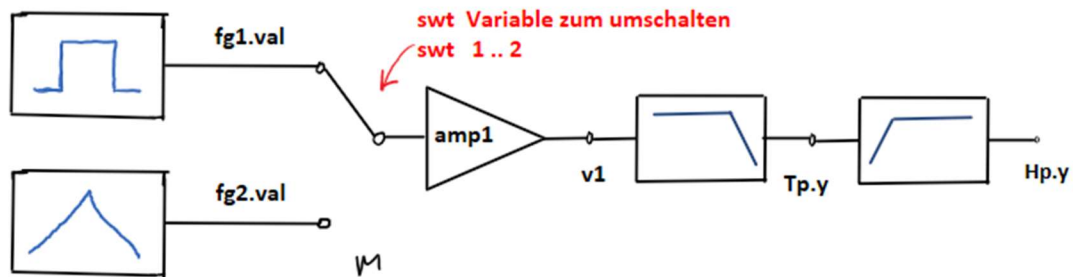
## Pointer und Arrays 4

Schreiben Sie ein C++ Programm welches die Umlaute in einem C-string durch **ae ue oe** ersetzt.

1. Aufgabe  
Erklären Sie die Funktionsweise von ASCII-Strings in C
2. Aufgabe  
Erklären Sie mithilfe einer Skizze die Funktionsweise der Funktion *UmlautErsatz()*
3. Aufgabe  
Programmieren Sie die Stringbearbeitungs-Funktionen und das Hauptprogramm

## Robotik und Signalverarbeitung 1 Pri2

Schreiben Sie die Echtzeit SW für einen digitalen Funktionsgenerator und ein Tiefpassfilter



### 1) Aufgabe

Programmieren Sie einen digitalen Funktionsgenerator mit den Kurvenformen Rechteck, Dreieck. Der Funktionsgenerator soll so effizient wie möglich programmiert werden um hohe Abtastfrequenzen realisieren zu können.

### 2) Aufgabe

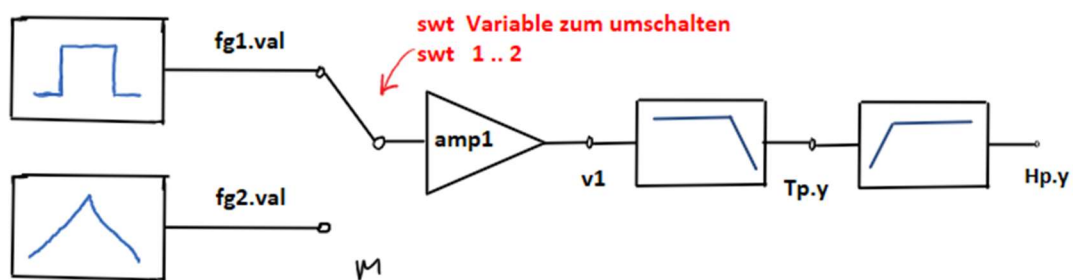
Programmieren Sie einen digitalen Tiefpass 1er Ordnung mit dem das Funktionsgenerator-Signal gefiltert werden kann.

### 3) Aufgabe

Das Funktionsgenerator und das Filter-Signal sollen mit dem Monitoringprogramm RboMon dargestellt werden.

## Robotik und Signalverarbeitung 2 Pri2

Schreiben Sie die Echtzeit SW für einen digitalen Funktionsgenerator und ein Hochpassfilter



### 1) Aufgabe

Programmieren Sie einen digitalen Funktionsgenerator mit den Kurvenformen Sägezahn Sinus. Der Funktionsgenerator soll so effizient wie möglich programmiert werden um hohe Abtastfrequenzen realisieren zu können.

### 2) Aufgabe

Programmieren Sie einen digitalen Hochpass 1er Ordnung mit dem das Funktionsgenerator-Signal gefiltert werden kann.

### 3) Aufgabe

Das Funktionsgenerator und das Filter-Signal sollen mit dem Monitoringprogramm RboMon dargestellt werden.

## Robotik 3 Pri1

Schreiben Sie ein FollowMouse-Programm für das RoboWorld Framework.

1. Aufgabe  
Schreiben Sie ein FollowMouse-Programm für das RoboWorld Framework.
2. Aufgabe  
Erklären Sie das RoboWorld Framework.
3. Aufgabe  
Welchen realen Roboter-Aufbau simuliert das RoboWorld Framework.

## Robotik 4 Pri1

Schreiben Sie ein Kollisionsvermeidungs Programm für das RoboWorld Framework.  
Entweder Rasenmäher oder nur 2 Roboter, die die Plätze tauschen

1. Aufgabe  
Schreiben Sie ein Kollisionsvermeidungs -Programm für das RoboWorld Framework.
2. Aufgabe  
Erklären Sie das RoboWorld Framework.
3. Aufgabe  
Welchen realen Roboter-Aufbau simuliert das RoboWorld Framework.  
Rechts, Links SetPow() : realer Robot  
vs.  
SetV() SetdPhi() : simulierter Robot