Vorbereitung

# 2 Projektdurchführung

Die Durchführung des Projekts lässt sich in vier Phasen gliedern:

1. Projektplanung und Teambildung.

2. Konzeptphase,

3. Entwurfs- und Testphase,

4. Auswertungsphase.

## 2.1 Projektplanung und Teambildung

Die erste Phase dient dazu, das Projekt inhaltlich und zeitlich zu planen. Das Ergebnis dieser Phase ist ein Projektplan, aus dem die einzelnen Arbeitsschritte hervorgehen. Zu jedem Arbeitsschritt gehört der geschätzte

Zeitbedarf und eine Zuordnung zu den Teammitgliedern.

Wie werden die Labortermine genutzt?

-> Excel-Datei

### Grundlage

* Modulare Aufteilung der Projektdateien
* Wiederholte Absprachen vor den einzelnen Terminen

### Termin 1

* Rechnersysteme und Peripherie untersuchen
* Wheel Sensor -> Signalanalyse, Versuche
* Konzept Anpassung

### Termin 2

* Display: grundlegende Funktionen, sinnvolle Darstellungen
* Schnittstellen Definition
* Wheel Sensor -> S1 und S2 analysieren
* Interupt handler zur zeitlichen Abstimmung der Signale

### Termin 3

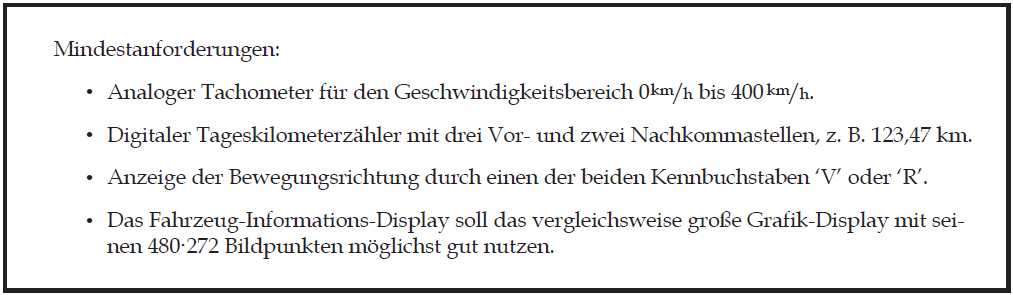
* Wheel Sensor -> S1 und S2 analysieren
* Interupt handler
* Anzeige der analysierten Signale

### Termin 4

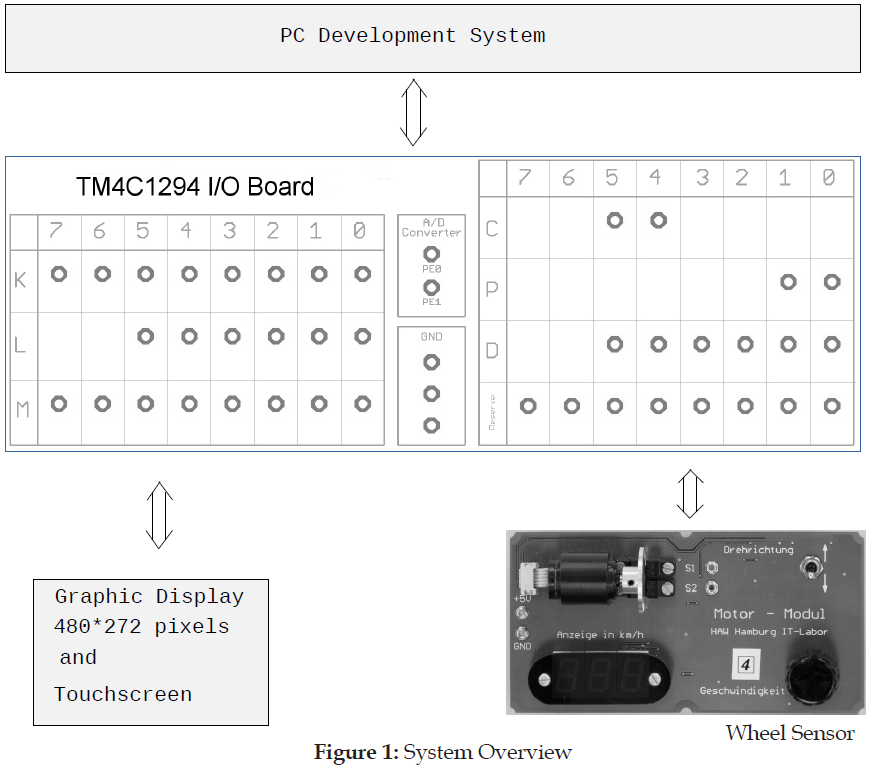
- Systemtest (gesamt)

## 2.2 Konzeptphase

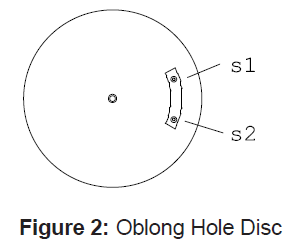
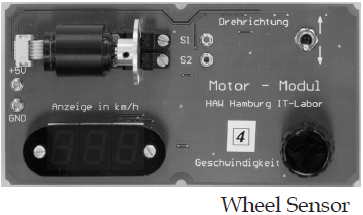
In der zweiten Phase wird das Fahrzeug-Informations-Display konzipiert.



Anschlussübersicht:



### 2.2.1 Wie werden Bewegungsrichtung und Raddrehzahl gemessen?



Das „Wheel Sensor“ – Modul enthält einen Motor, welcher eine Scheibe mit zwei Löchern antreibt.

Die Drehgeschwindigkeit und Drehrichtung kann verändert werden.

Es werden zwei kurze High-Signale (s1 und s2) bei jeder Umdrehung ausgegeben.

Aus der Reihenfolge der High-Impulse lässt sich die Bewegungsrichtung ermitteln.

Die Signale s1 und s2 müssen logisch erfasst werden.

Messung mit dem Oszilloskop, um Zeitverhalten zu analysieren.

#### Drehrichtung

Die High-Signale von s1 und s2 werden von der Interrupt Service Routine einzeln gezählt. In der ISR wird die Drehrichtung gesetzt. Durch das einzelne zählen von s1 und s2 kann die Drehrichtung erkannt werden. Wenn der Zähler von s1 größer als der Zähler von s2 ist kann man von einer Vorwährtsdrehung ausgehen. Wenn der Zähler von s1 kleiner als der Zähler von s2 ist kann man von einer Rückwärtsdrehung ausgehen. Wenn beide Zähler gleich groß sind bleibt man in der aktuellen Richtung. Die ISR beider Signale sind in der Lage die Drehrichtung zu verändern. Da pro Umdrehung das erste Signal um einen größer ist als das zweite Signal, setzt das erste Signal die Drehrichtung. Das Zweite erkannte Signal hat den gleichen Wert wie sein Vorgänger und ändert nichts an der Richtung.

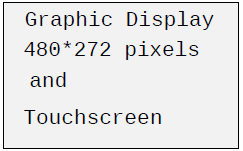
#### Raddrehzahl

Für die Raddrehzahl Erkennung werden die High-Signale von s1 und s2 von der Interrupt Service Routine gezählt. Mit einem Timer werden zu fest definierten Zeiten der Zählerwert der Signale s1 oder s2 abgespeichert. Mit dem festen Zeitabstand und der Anzahl der Umdrehungen kann die Drehzahl/sec berechnet werden.

Drehzahlt/s = (Zählerwert/Timer\_Zeit) \* (Wert, damit aus Timer\_zeit \* Wert = 1s/1000ms)

Geschwindigkeit = Radumfang(1m) \* Drehzal/s

### 2.2.2 Wie soll das Grafik-Display zur analogen und digitalen Ausgabe aufgeteilt werden?



Die Balkenfarbe und die Kasten Farbe für die Fahrtrichtung passen sich an die Fahrtrichtung an: Vorwärts (grün), Rückwärts (gelb).

Durch Berührung lässt sich die Tageskilometerzähler-Anzeige auf die digitale Geschwindigkeits-Anzeige ändern.

#### Vorwärts

**V**

**123,47km**

#### Rückwärts

128,31 km/h

R

#### Start Up

### 2.2.3 Ist es sinnvoll, das Fahrzeug-Informations-Display während der Testphase auf dem PC-Bildschirm auszugeben?

Es wäre sinnvoll für Arbeiten außerhalb des Labors, allerdings könnte sich das Display in dem Labor anders verhalten.

Das Ziel ist es, das System vollständig im Labor zu testen. Mit vorhandener Hardware wäre die Darstellung auf dem PC nicht nötig.

### 2.2.4 Wird ein Interrupt-Handler benötigt?

Die Analyse der Wheel-Sensor Signale S1 und S2 müssen im Zweiten Termin mithilfe von Interrupts und Timern analysiert werden. Interrupts sind sinnvoll, da die Signale der Drehzahlsensoren zeitlich stark variieren. Die Polling variante führt dazu, dass Signale möglicherweise nicht erkannt werden. Da aktiv auf die Signale gewartet werden muss, kann man keine anderen Prozesse ausführen.

Die Interrupts werden von den Sensoren über die Pins direkt getriggert. Die Interrupt Handler zählen die Umdrehungsanzahl.

### 2.2.5 Aufgaben Display

* Bibliothek für „Buchstaben und Zahlen zeichnen“
* Segmentdarstellung
* Wechsel von Tageskilometer zu Digitale Geschwindigkeit durch Touch
* Schnittstelle zur Verfügung stellen
* Start Up Bild einmalig starten und einzelne Segmente überschreiben
* Nur das zu ändernde Feld löschen und neu beschreiben

### 2.2.6 Aufgaben Drehzahlmessung

* Drehrichtung erkennen (s1 u s2)
* Pins Initialisieren (AF)
* Interrupts initialisieren (Pins)
* I-Handler schreiben
* Zählen (Overflow abfangen)
* Drehrichtung Setzen (Vergleich der Zähler)
* Definition von Vorwärts (V) und Rückwärts (R)
* Drehzahl berechnen (m/s)

*unsigned int detect\_frequency (unsigned int s1/s2)*

* Timer initialisieren (one shot, ISR, highest prio)
* Zählerwert von s1/s2 holen nach bestimmter Zeitdauer
* Berechnung der Drehzahl anhand von Umdrehungen pro fester Timerzeit
* Zähler der s1 und s2 Signale zurücksetzten
* Timer neustarten
* Drehgeschwindigkeit berechnen (s1/s2)

*unsigned int detect\_speed (unsigned int s1/s2)*

* Berechnung der Geschwindigkeit anhand von Umdrehung/s und Umfang des Rads.
* Testen von Interrupt und Display Kommunikation (zeitliche Abstimmung)

### 2.2.7 Grundlegende Aufgaben

* Übersichtliche Codierung (Hierarchie der Aufgaben)
* Versionierung mit Git
* Dokumentation von Ergebnisse