

1 Navigation

Die Entwicklungsoberfläche lässt sich durch sechs Buttons am unteren Bildschirmrand bedienen.

Alternativ kann man auch über die Tastatur navigieren.

Hierfür gibt es folgende Abkürzungen:

Strg + ' + ' Heranzoomen

Strg + ' - ' Herauszoomen

Strg + *Pfeiltaste* Ansicht in die entsprechende Richtung bewegen

2 Aufbau der Oberfläche

Die Oberfläche lässt sich grob in drei Bereiche unterteilen - Werkzeugleiste, Programm und Steuerknöpfe.

Die Werkzeugleiste - gelber Kasten - befindet sich am linken Bildschirmrand und beinhaltet alle Programmbausteine, die zum Entwickeln notwendig sind.

Der zweite Teil - roter Kasten - sind die Buttons, die zum Entwickeln bzw. Ausführen benötigt werden.

Hier gibt es einmal den Mülleimer. Dort werden nicht mehr benötigte Programmbausteine abgelegt, die daraufhin gelöscht werden.

Der Testen-Button daneben compiliert das Programm, überträgt es auf das Board und führt es aus. Treten Probleme auf, gibt es einen Test-stoppen-Button, dieser hält das Programm auf dem Board an.

Im grünen Kasten ist das eigentliche Programm zu sehen. Dieses besteht aus mindestens zwei Komponenten:

- Programmstart: Hier werden alle Aktionen aufgelistet, die beim Programmstart ausgeführt werden.
- Ereignis: Hier wird für jedes Ereignis eine Abfolge von Anweisungen definiert. Es ist immer ein leerer Ereignispunkt vorhanden, so dass man immer neue Ereignisse definieren kann.

Der Punkt Ablage dient dazu, um Programmbausteine, die man ggf. gerade nicht verwenden möchte, abzulegen ohne sie zu löschen.

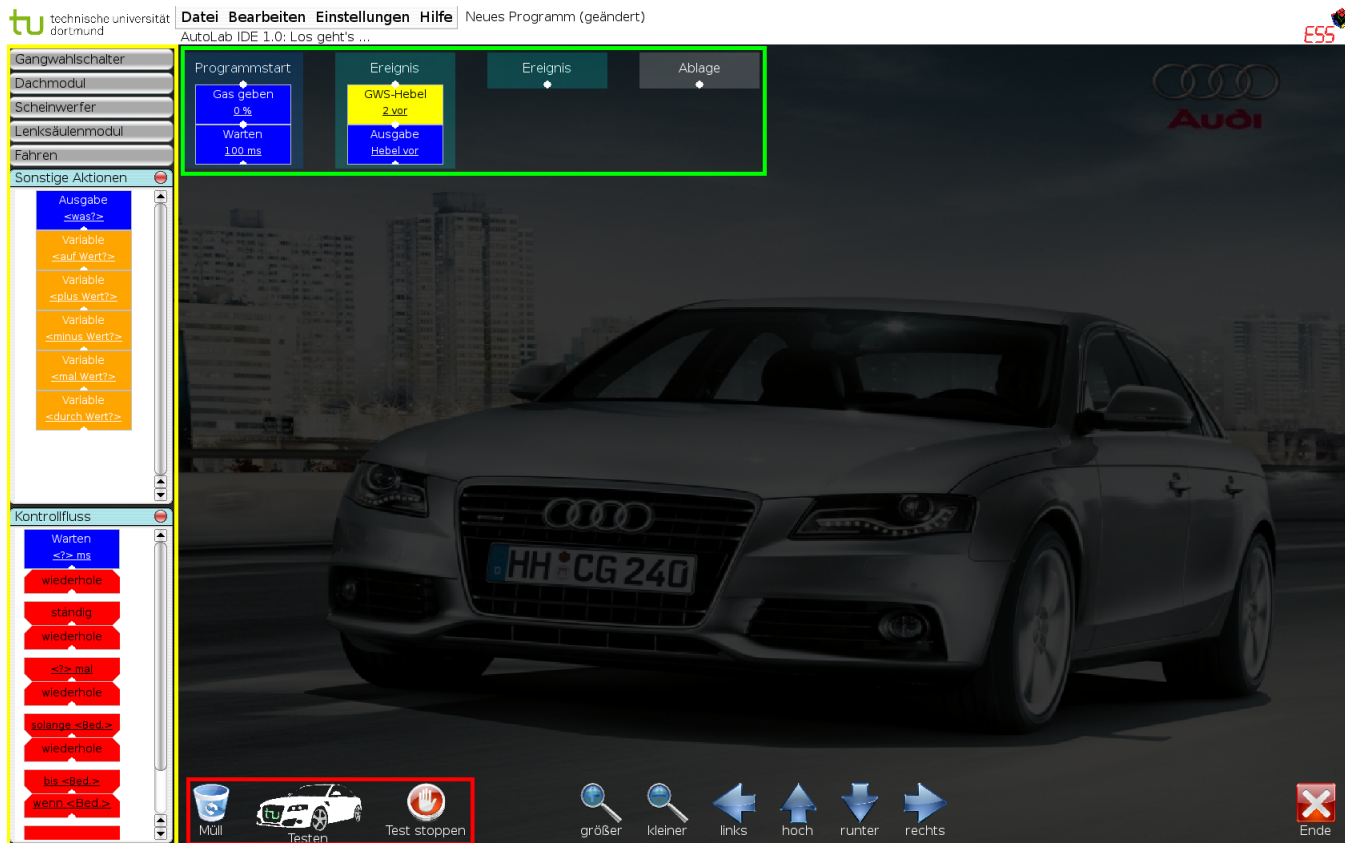


Abbildung 1: Übersicht IDE

3 Werkzeugleiste

In der Werkzeugleiste befinden sich sieben verschiedene Kategorien von Programmbausteinen. Durch klicken auf die jeweilige Kategorie klappt diese auf. Möchte man sie wieder schließen genügt ein Klick auf den kleinen roten Button neben dem Namen.

Es gibt vier Arten von Bausteinen, die durch ihre Farbe identifiziert werden.

- Rot: Kontrollstrukturen, z.B. Schleifen oder bedingte Anweisungen
- Orange: Variablen erstellen oder Operationen auf diesen ausführen
- Blau: Aktionen durchführen, z.B. 10 ms warten
- Gelb: Ereignisse, z.B. Blinkerhebel betätigt

Es ist absolut notwendig, dass ihr nach jedem Befehl - blaue Bausteine - eine kurze Zeit wartet. Dazu baut ihr ein Warten-Baustein ein und stellt die Zeit auf 1-2 ms ein.

4 Wie ist ein Programm aufgebaut?

Unsere Programme bestehen generell aus Ereignissen. Wir können selber bestimmen auf welche Ereignisse wir reagieren wollen, wie das genau geschieht wird in einer späteren Lektion erläutert.

Sobald das Programm auf dem Board läuft, wird beim Eintreten eines Ereignisses der entsprechende Zweig abgearbeitet. Die Ereigniszweige können mehrmals ausgeführt.

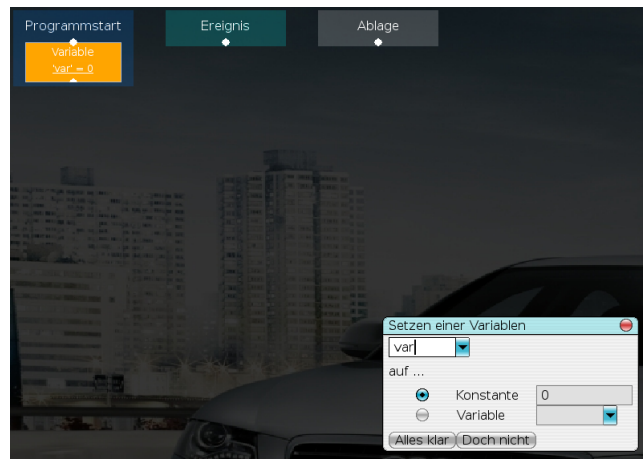
Ein besonderes Ereignis stellt der Programmstart dar. Diese Anweisung werden nur einmal beim Starten des Programmes durchgeführt.

5 Umgang mit der Entwicklungsumgebung

An dieser Stelle wird exemplarisch das Ausgeben eines Textes sowie das Erstellen einer Variable erläutert. Der Umgang mit den anderen Programmbausteinen ist ähnlich.

Zunächst wollen wir uns eine Variable *foo* erzeugen und dieser initial den Wert 42 zuweisen. Hierzu klickt man unter sonstige Aktionen auf „Variable - <auf Wert?>“ und verschiebt das Element nun unter Programmstart. Durch einen erneuten Klick bleibt der **Baustein** nun hängen.

Anschließend erscheint folgender Dialog:



(a) jj

Abbildung 2: Erzeugen einer Variable

Hier wird nun der Variablenname *foo* sowie der Wert 42 festgelegt und mit „Alles klar“ bestätigt.

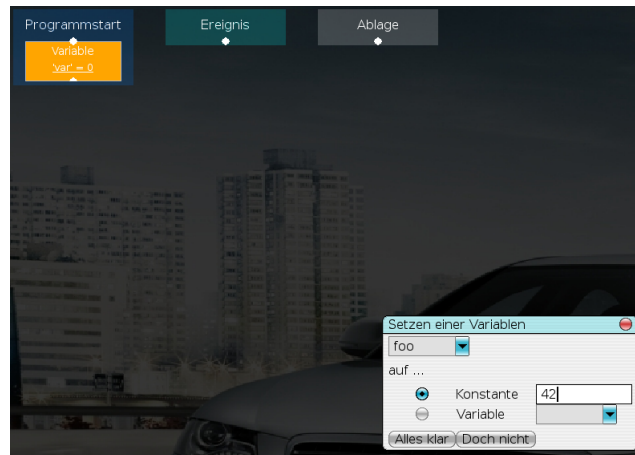


Abbildung 3: Wert der Variablen setzen

Sollte man sich vertippt haben, besteht die Möglichkeit seine Eingabe zu korrigieren. Hierzu klickt ihr mit der Maus auf den Namen der Variablen - der Mauszeiger sollte nun zu einer Raute geworden sein - und es erscheint erneut der Eingabedialog.

Nun wollen wir eine bedingte Anweisung einbauen. Wählt in der Kategorie Kontrollfluss den Baustein „Wenn <Beding.> Dann Sonst“ aus und hängt es unter eure Variable.

In dem Dialog stellt Ihr als Bedingung ein „Wenn Variable foo größer gleich 42“.

D.h. sollte *foo* größer oder gleich 42 sein, wird der linke Zweig dieses Blocks ausgeführt andernfalls der Rechte.

So sollte euer Programm bis jetzt aussehen:

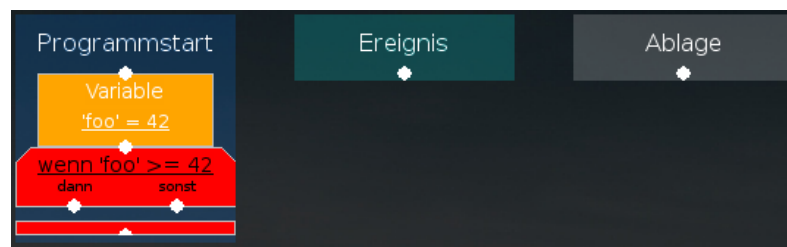


Abbildung 4: Demoprogramm

Nehmt euch jetzt aus der Sektion „sonstige Aktionen“ ein Baustein „Ausgabe“ und hängt es in den Dann-Zweig.

Es soll der Text „foo ist größer gleich als 42“ ausgegeben werden. Wiederholt diese Schritte nun, so dass im Sonst-Fall ausgegeben wird „foo ist kleiner als 42“.

So sieht das fertige Programm aus:

Zum Ausführen klickt auf den Testen-Button.

Es erscheint ein Fenster, dass zum einen eure Ausgaben zum anderen die Nachrichten auf dem CAN-Bus anzeigt.

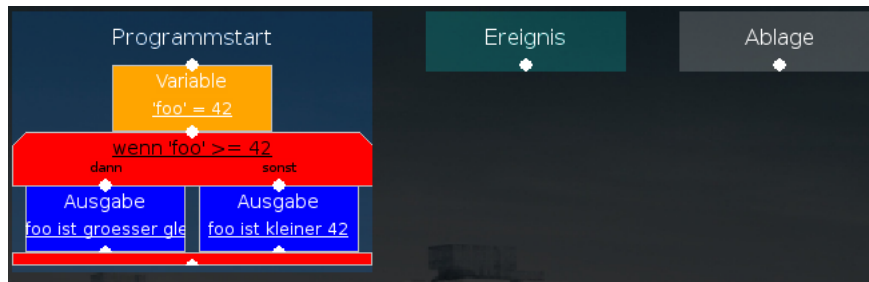


Abbildung 5: Fertiges Demoprogramm

Anmerkung:

Möchtet ihr einen Programmbaustein kopieren, so könnt ihr über Bearbeiten -> Kopieren oder *Strg + 'c'* in den Kopiermodus wechseln. Der Mauszeiger wird nun zu einem Fadenkreuz. Jetzt klickt ihr auf die kleine Verbindungsraute an dem Baustein, ab dem ihr kopieren wollt. Der darunter befindliche Programmbaustein hängt nun als Kopie an der Maus. Nun könnt ihr ihn mittels Klick an die gewünschte Stelle einfügen.

Der Kopiermodus lässt sich über die gleiche Tastenkombination bzw. über das Menü wieder deaktivieren.

6 Ablauf der Entwicklung

Ein typischer Entwicklungszyklus besteht in der Regel aus drei Abschnitten.

Zunächst einmal kommt das eigentliche Entwickeln, sprich dem zusammenbauen des Programms in der Entwicklungsumgebung.

Anschließend wird das Programm übersetzt (compiliert) und auf das Board übertragen.

Jetzt beginnt der spannende und dritte Teil: das Testen.

Hierbei ist es wichtig, dass ihr langsam und Schritt für Schritt vorgeht. Schaut euch genau an was euer Programm z.B. beim Start machen soll, welche Ausgabe ggf. auf der Konsole erscheint oder welche Reaktion sichtbar sein soll, z.B. Licht geht an.

Wenn ihr eure Ereignisbehandlungen testet, achtet darauf, dass ihr immer nur ein Ereignis herbeiführt, z.B. „Gas geben“. Dadurch wird sichergestellt, dass ihr eventuelle Fehler auch eindeutig einem Ereignis zuordnen könnt.

7 Aufgabe

Erstellt ein kleines Programm, dass das Licht - es spielt keine Rolle welches Licht ihr nehmt - einschaltet, eine Sekunde wartet und das Licht wieder ausschaltet.

Zum Schluss soll der Text „Hallo Auto“ ausgegeben werden.