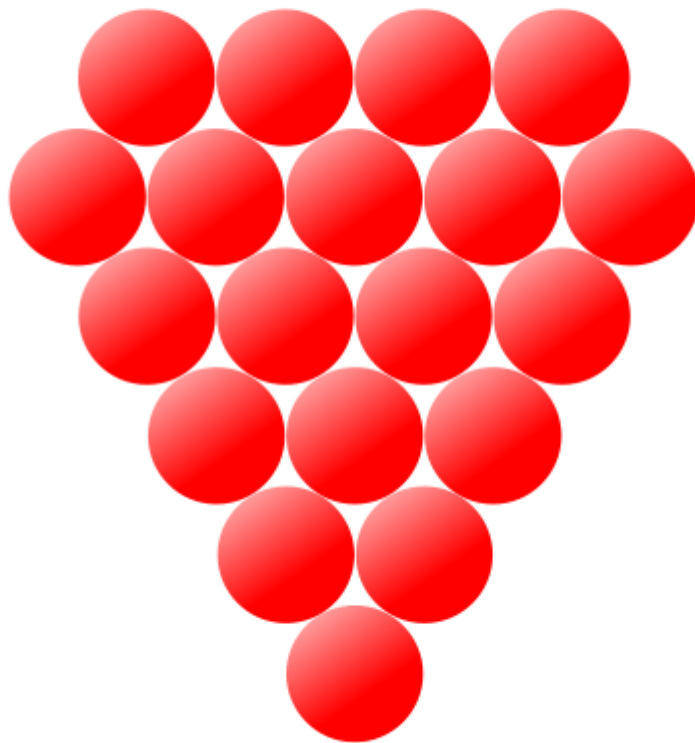




# - Grape - 2.0



**Graphical Programming Environment**

## **Anleitung**

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>3</b>
1.1	Inhalt der CD . . . . .	3
1.2	Zusätzlich benötigte Komponenten . . . . .	3
<b>2</b>	<b>Installation und Start</b>	<b>5</b>
2.1	Voraussetzungen . . . . .	5
2.2	Installation von Grape . . . . .	5
2.3	Grape starten . . . . .	5
<b>3</b>	<b>Programme graphisch erstellen</b>	<b>7</b>
3.1	Neues Projekt starten . . . . .	7
3.2	Programmelemente . . . . .	8
3.3	Elemente einfügen, verschieben, umbenennen und löschen . . . . .	9
<b>4</b>	<b>C++ Code generieren</b>	<b>11</b>
4.1	Code-Generator anschalten . . . . .	11
4.2	Objekte deklarieren . . . . .	12
4.3	Programm übersetzen . . . . .	13
4.4	Programm auf den Controller übertragen . . . . .	14

# Kapitel 1

## Einleitung

Wir gratulieren Ihnen herzlich zu Ihrer grafischen Programmieroberfläche qfix Grape!

Grape steht für "Graphical Programming Environment", also Grafische Programmierumgebung. Das bedeutet, dass man mit Grape programmieren kann, ohne über Erfahrungen in einer textorientierten Programmiersprache, wie beispielsweise C++, zu verfügen. Es ist vielmehr so, dass man beim Befassen mit Grape ganz automatisch Programmieren lernt, da man stets die automatisch erzeugten C++ Programme live betrachten kann, bevor man sie auf das Controllerboard lädt.

Um ständig über Neuigkeiten informiert zu sein, Ersatzteile oder Erweiterungen zu beziehen oder sich mit anderen qfix Benutzern auszutauschen, werfen Sie bitte regelmäßig einen Blick auf die qfix Homepage unter [www.qfix.de](http://www.qfix.de).

Wir wünschen viel Spaß mit Grape!

Ihr qfix Team

### 1.1 Inhalt der CD

Auf der CD sind folgende Einträge vorhanden:

- Datei "setup-Grape.exe" (bzw. "setup-Grape-free.exe")
- Datei "README.txt"
- Datei "LIESMICH.txt"
- Datei "Anleitung-Grape.pdf"(diese Datei)

### 1.2 Zusätzlich benötigte Komponenten

Um Programme nicht nur am PC zu entwickeln, sondern auch in Realität testen zu können, wird mindestens ein qfix Controllerboard, beispielsweise das MiniBoard (Best.Nr.

HBAA011) oder das SoccerBoard (Best.Nr. HBAA009) benötigt. Für Programme mit Motor wird desweiteren ein Akku-Set (z.B. Best.Nr. QAAC000) benötigt.

Noch mehr Spaß macht natürlich die Verwendung eines "Crash-Bobby" Roboters (QAAA000) oder des "Control-Kit" (QAAA002) zum Messen-Steuern-Regeln.

# Kapitel 2

## Installation und Start

### 2.1 Voraussetzungen

Die Software kann ab Windows XP installiert werden. Die Verwendung mit älteren Windows-Versionen wurde nicht geprüft und wird nicht unterstützt.

Für das Laden von Programmen auf ein qfix Controllerboard wird ein freier USB Port benötigt.

**Achtung:** Um den CodeGenerator nutzen zu können, muss die "qfix Software" installiert sein! Dieses Software-Paket ist frei erhältlich und kann von der Internetseite [www.qfix.de](http://www.qfix.de) im Downloadbereich heruntergeladen werden.

### 2.2 Installation von Grape

Zur Installation von Grape starten Sie das Setup-Programm "setup-Grape.exe".

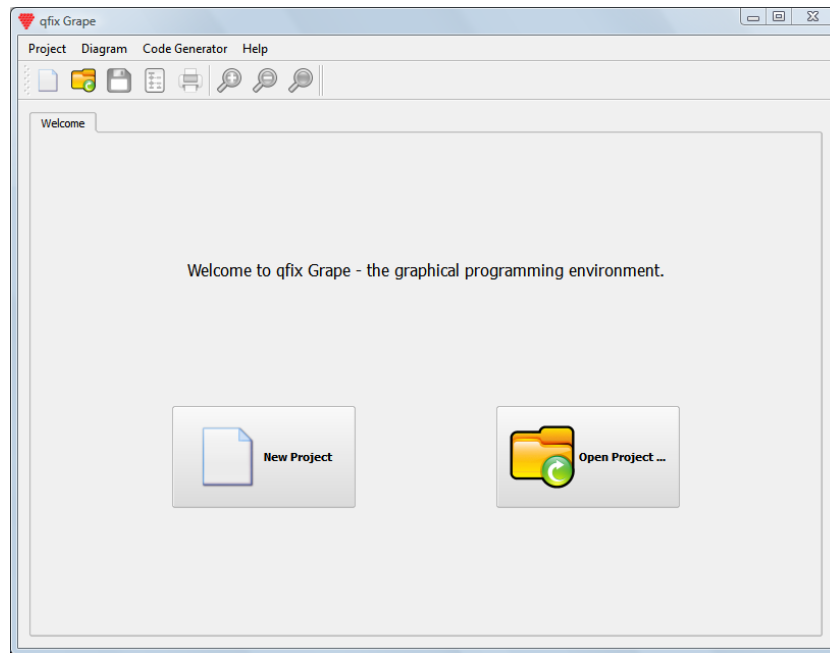
Bitte folgen Sie den Anweisungen des Installationsprogramms.

Es empfiehlt sich, das Programm in das vorgeschlagene Verzeichnis zu installieren.

### 2.3 Grape starten

Um Grape zu starten, wählen Sie das Programm über das Startmenü aus, beispielsweise über `Start→Programme→qfix robotics→Grape→Grape`.

Grape meldet sich mit seinem Begrüßungsbildschirm. Nach kurzer Zeit verschwindet dieser und Sie sehen das Hauptfenster, in dem Sie ein neues Projekt anlegen können oder ein bestehendes Projekt laden können.



# Kapitel 3

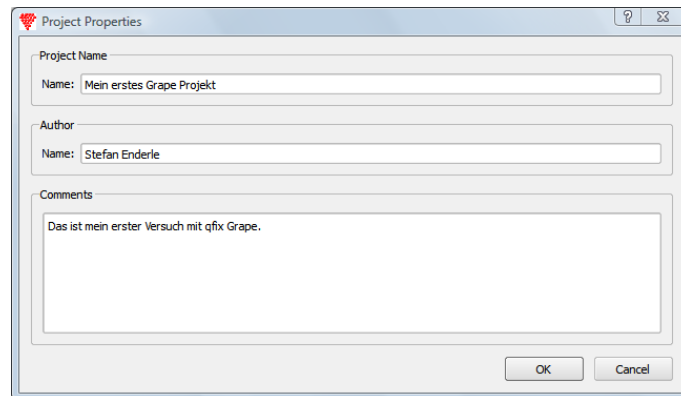
## Programme graphisch erstellen

In diesem Kapitel wird beschrieben, wie man in Grape ein Flussdiagramm erstellt, um die Programmlogik des zu erstellenden Programmes graphisch darzustellen.

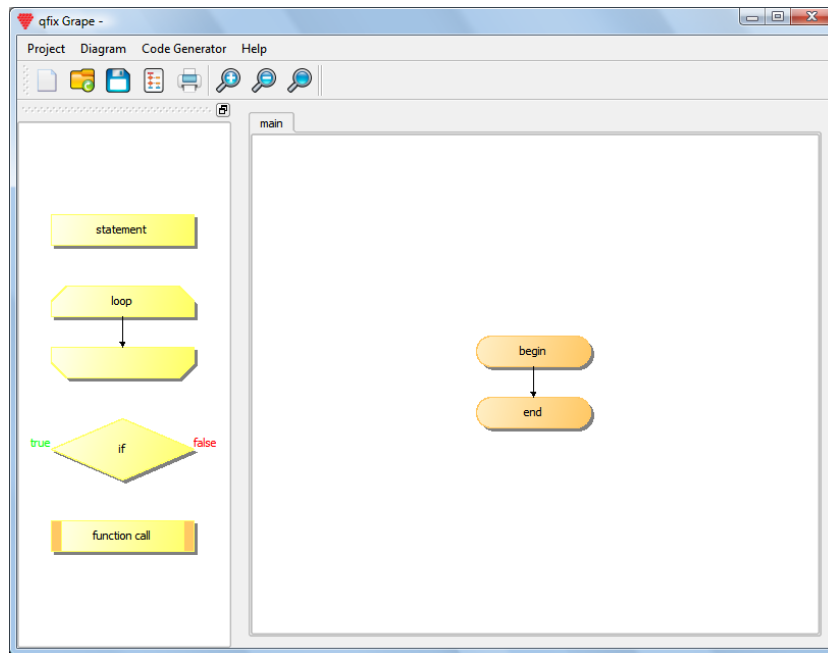
Als Beispielprojekt soll ein einfaches Programm dienen, das andauernd einen Button abfragt und bei Betätigung eine LED an- bzw. ausschaltet.

### 3.1 Neues Projekt starten

Klicken Sie im Startfenster "New Project" an. Es öffnet sich ein Dialog, in dem der Name und Autor des neuen Projekts sowie ein Kommentar eingegeben werden kann.



Drückt man OK, so öffnet sich das Hauptfenster. Dieses enthält anfangs ein nahezu leeres Flussdiagramm, das lediglich zwei abgerundete Kästchen mit "Start" und "End" enthält. Das leere Programm bildet quasi das Grundgerüst jedes grafischen Programmes bestehend aus Anfang und Ende. Den leeren Bereich dazwischen, der im Moment durch den Pfeil symbolisiert wird, gilt es nun mit Anweisungen füllen.



Das Hauptfenster ist in zwei Teile unterteilt: Links die Auswahlliste für Programmelemente, rechts das Programmierfenster.

## 3.2 Programmelemente

In der Auswahlliste am linken Rand des Hauptfenster befinden sich die Symbole, aus denen ein Flussdiagramm erstellt werden kann:

- **statement** (dt. "Anweisung") steht für Befehle wie "mache die LED mit der Nummer 3 an" oder "starte den Motor 1 mit Geschwindigkeit 100".
- **if** (dt. "wenn" oder "falls") bezeichnet eine Fallunterscheidung, also beispielsweise die Aussage "wenn der Button 1 gedrückt ist, dann ..." oder sogar "wenn ..., dann mache die LED an, ansonsten mache sie aus".
- **loop** (dt. Schleife) bezeichnet die wiederholte Ausführung einer Anweisung oder einer ganzen Sequenz von Anweisungen.
- **function call** (dt. Funktionsaufruf) ruft eine Unterfunktion auf.

Bei größeren Flussdiagrammen wird manchmal etwas mehr Platz im Hauptfenster benötigt. Aus diesem Grund kann das Fenster, das die Programmelemente enthält aus dem Hauptfenster herausgezogen werden. Hierzu kann es am oberen Rand gefasst und herausgezogen werden oder alternativ kann der Button oben rechts mit den zwei angedeuteten Fenstern gedrückt werden. Das Fenster wird durch schieben an die entsprechende Position oder durch Doppelklick auf den oberen Fensterrand wieder angedockt.



### 3.3 Elemente einfügen, verschieben, umbenennen und löschen

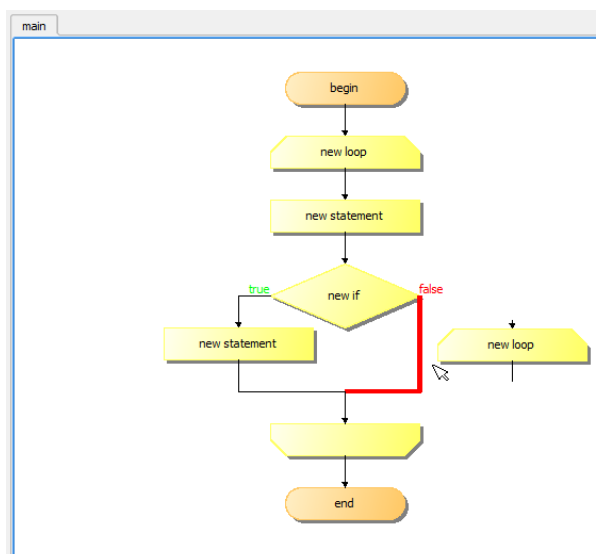
#### Elemente einfügen

Programmelemente werden per Drag&Drop in das bestehende Flussdiagramm eingefügt. Ein Element wird mit der Maus gefasst (linke Maustaste drücken und halten) und an der gewünschten Einfügestelle wieder losgelassen. Das Einfügen ist jeweils an der Stelle eines Pfeiles möglich. Wird ein Element über einen Pfeil des Flussdiagrammes gezogen, so verfärbt sich der Pfeil grün und das Element kann losgelassen werden und wird so an dieser Stelle eingefügt.

#### Elemente und Blöcke verschieben

Es ist möglich, bereits eingefügte Elemente an eine andere Stelle des Flussdiagrammes zu verschieben. Hierzu wird das Element wie oben beschrieben mit der Maus gefasst und über einem grünen Pfeil losgelassen.

Es können auch ganze Blöcke verschoben werden, beispielsweise eine Programmschleife (Loop) samt Inhalt. Hierzu wird das oberste Element gefasst und verschoben. Der gesamte Inhalt verschiebt sich mit. Beachten Sie, dass ganze Blöcke nicht an jede Stelle des Flussdiagrammes verschoben werden können! Insbesondere können Blöcke nicht an eine Stelle innerhalb sich selbst geschoben werden. Ist ein Verschieben nicht möglich, so färbt sich der entsprechende Pfeil rot (siehe Bild).



#### Elemente umbenennen

Zum Umbenennen eines Elements kann dieses mit der linken Maustaste doppelt angeklickt werden. Es öffnet sich ein Eingabefeld innerhalb des Elements und ein neuer Name kann eingegeben werden. Die Eingabe wird mit der Eingabetaste oder durch Mausklick außerhalb des Elements übernommen.

Alternativ öffnet sich per rechtem Mausklick auf ein Element das Popup-Menü dieses Elements. Hierin gibt es einen Eintrag "Rename", über den ebenfalls das Eingabefeld geöffnet werden kann.

### **Elemente löschen**

Über das Popup-Menü lässt sich auch der Eintrag "Delete" auswählen, der ein Element, bzw. einen ganzen Block löscht.

# Kapitel 4

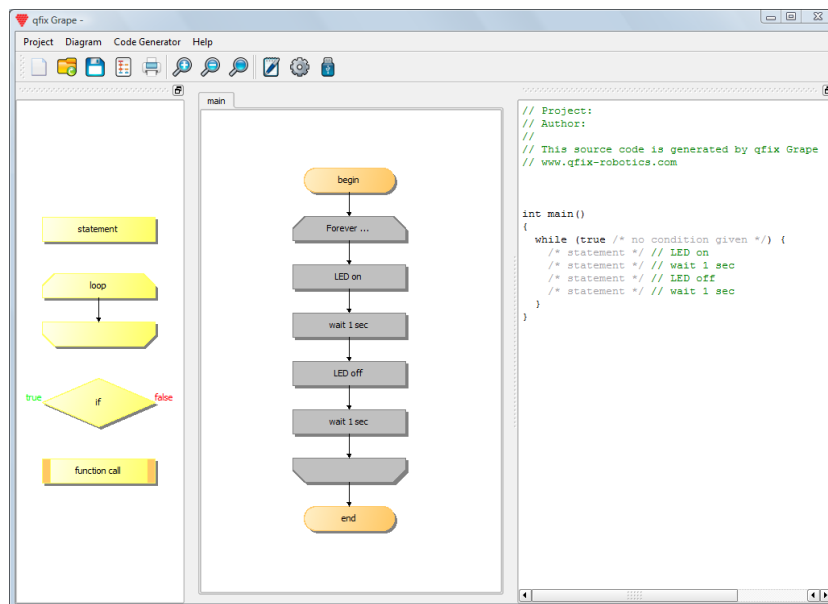
## C++ Code generieren

Der Code-Generator von Grape hat die Aufgabe, das erstellte Flussdiagramm in (lesbaren) Programmcode zu übersetzen. Natürlich ist dies allein auf Basis der Beschreibungen, die auf den Elementen des Flussdiagrammes stehen, nicht möglich. Vielmehr muss nun genau spezifiziert werden, was mit jedem einzelnen Element gemeint ist.

### 4.1 Code-Generator anschalten

Um C++-Code zu generieren, muss erst der Code-Generator von Grape aktiviert werden. Im Menü gibt es hierzu den Punkt Code Generator → Enable Code Generator.

Nach dem Anschalten öffnet sich rechts im Hauptfenster ein weiteres Unterfenster, das den zu erstellenden Code beinhaltet:



Das neue Code-Fenster ist ebenso wie das Fenster mit den Programmelementen aus dem Hauptfenster herausnehmbar und kann so beliebig platziert werden.

Wie man sieht existiert bereits ein Programm-Grundgerüst, das sogar schon durch den Compiler fehlerfrei übersetzt würde, eine Funktion erfüllt es jedoch noch nicht. Diese soll

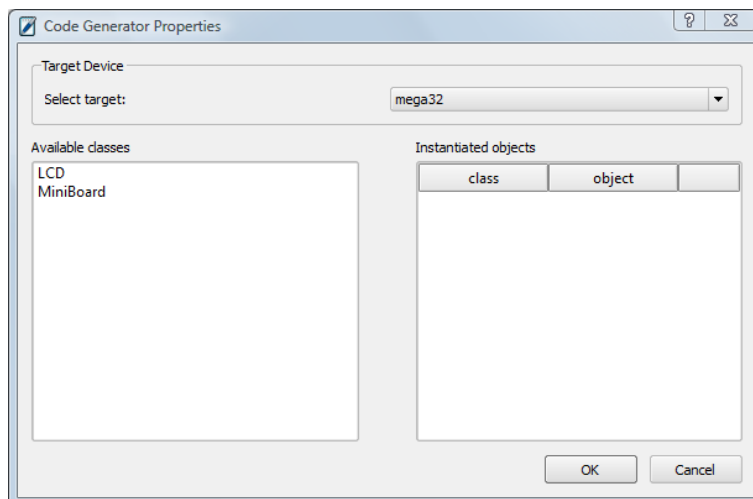
nun im folgenden hinzugefügt werden.

## 4.2 Objekte deklarieren

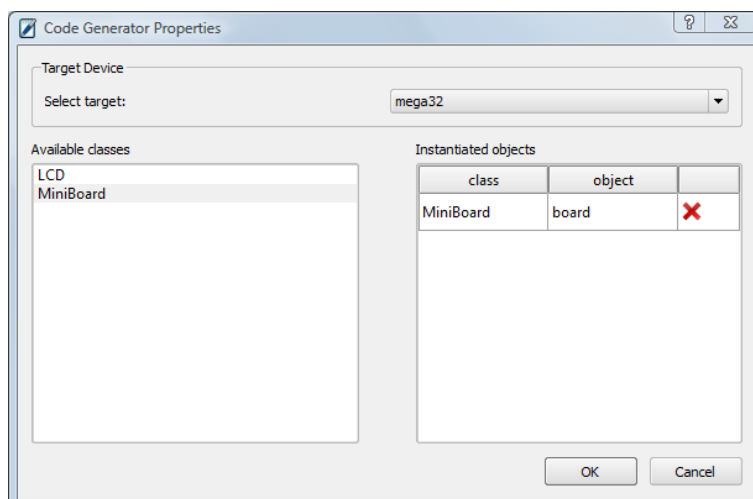
Grape benutzt einen objektorientierten Ansatz. Das bedeutet, dass Objekte aus der Realität, zum Beispiel ein Roboter oder ein Controllerboard, durch entsprechende Software-Objekte repräsentiert werden. Die Daten und Funktionalitäten, die ein solches Objekt bietet, werden durch seine sogenannte *Klasse* festgelegt.

Die Klassen, die benutzt werden können sind in Grape bereits vorgegeben. So gibt es beispielsweise Klassen, die die qfix Controller-Boards oder ein LCD-Display repräsentieren. Möchte man z.B. ein MiniBoard ansprechen, so legt man ein Objekte der Klasse `MiniBoard` an und gibt diesem Objekt einen eindeutigen Namen, etwa `board`.

Hierzu klicken Sie auf den Menüeintrag `Code Generator` → `Code Generator Properties`. Es öffnet sich folgender Dialog:



Doppelklicken Sie nun unter *Available classes* auf den Eintrag `MiniBoard`. Es öffnet sich ein Dialog, um den Objektnamen einzugeben. Geben Sie `board` ein und bestätigen mit OK. Die rechte Tabelle enthält danach den Namen der verwendeten Klasse und des Objektes.



In der Programmierung sagt man zu diesem Vorgang "Deklaration des Objekts 'board' vom Typ 'MiniBoard'".

Im Code-Fenster kamen durch die Deklaration zwei Zeilen hinzu:

1. `#include "qfixMiniBoard.h"`

Diese Zeile bewirkt das Einbinden der C++-Bibliothek für das MiniBoard.

2. `MiniBoard board;`

Die eigentliche Deklaration: Ein Objekt *board* der Klasse *MiniBoard*.

```
// Project:
// Author:
//
// This source code is generated by qfix Grape
// www.qfix-robotics.com

#include "qfixMiniBoard.h"

MiniBoard board;

int main()
{
    while (true /* no condition given */) {
        /* statement */ // LED on
        /* statement */ // wait 1 sec
        /* statement */ // LED off
        /* statement */ // wait 1 sec
    }
}
```

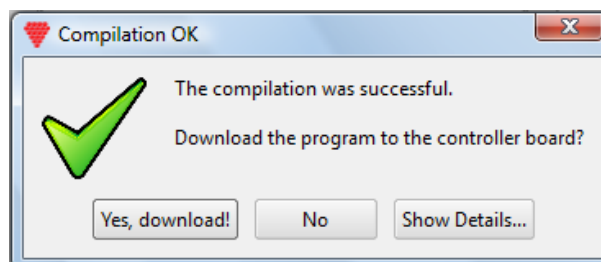
### 4.3 Programm übersetzen

Bemerkung: Vor dem Übersetzen sollten Sie das Projekt speichern. Falls es nicht gespeichert ist, öffnet sich beim Übersetzen zuerst ein "File Save"-Dialog, um es zu speichern.

Um das C++-Programm zu übersetzen drücken Sie die Funktionstaste F5 oder wählen Sie den Menüeintrag Code Generator→Compile.

In diesem Moment wird das angezeigte C++-Programm in das aktuelle Verzeichnis geschrieben. Als Dateiname wird der Dateiname der Projektdatei, jedoch mit Endung ".cc" benutzt.

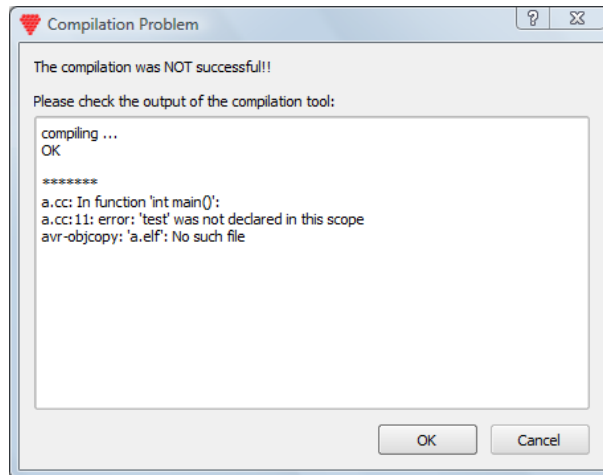
Die Compilierung erfolgt im Hintergrund und dauert einige Sekunden. Nach erfolgreicher Compilierung erscheint folgende Nachricht:



Diese Nachricht zeigt an, dass in Ihrem aktuellen Verzeichnis eine ausführbare Datei ("Binärdatei") mit der Endung `.bin` erzeugt wurde, die nun auf den Controller übertragen werden kann.

## Fehler beim Übersetzen

Grape ist so ausgelegt, dass normalerweise keine Übersetzungsfehler vorkommen können. Die Ausnahme besteht in den Anweisungen, bei denen "von Hand" in den C/C++-Code eingegriffen wird. Hier kann es natürlich sein, dass dem Benutzer ein Schreibfehler unterlaufen ist und der Compiler dann das Programm nicht korrekt übersetzen kann. In diesem Fall erscheint eine Fehlermeldung des Compilers in der folgenden Form:

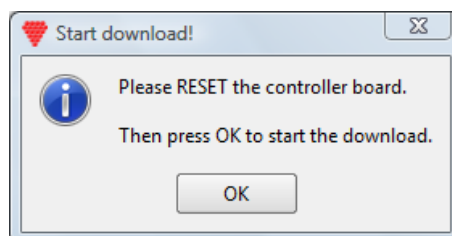


Die Fehlermeldung selbst unterscheidet sich natürlich je nach Fehler. Bitte kontrollieren Sie daraufhin die C/C++-Anweisungen und übersetzen das Programm nach der Korrektur neu, bis der Fehler behoben ist.

## 4.4 Programm auf den Controller übertragen

Bemerkung: Bitte verbinden Sie nun das Controllerboard über das USB-Kabel mit dem PC.

Normalerweise wird ein Programm nach erfolgreicher Kompilierung auf den Controller übertragen. Daher besitzt der Dialog nach der Kompilierung auch den Button "Yes, download!". Drücken Sie diesen Button, oder starten Sie außerhalb dieses Dialogs den Download per F6 bzw. Code Generator → Download, um die erzeugte Binärdatei auf das Controllerboard zu übertragen. Es öffnet sich die folgende Meldung:

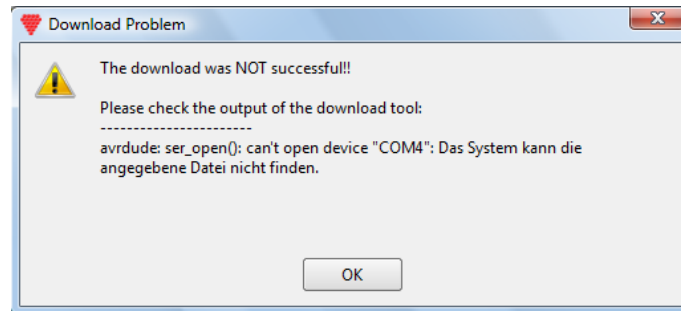


Bitte prüfen Sie, ob sich das Controllerboard im Download-Zustand befindet, d.h. die grüne LED 0 blinkt relativ schnell. Drücken Sie andernfalls den Reset-Button am Controllerboard.

Klicken Sie nun auf "OK". Für die Übertragung wird das Übertragungsprogramm `avrdude` verwendet. Dieses ist Teil der "qfix software" CD. Die Übertragung dauert nur wenige Sekunden. Nach erfolgreichem Übertragen starten Sie das Programm durch Drücken des Start-Buttons (Button 0) am Controllerboard.

### Download Fehler

Falls das Board nicht ordnungsgemäß per USB angeschlossen ist oder über den eingestellten USB-Port nicht gefunden wird, so erscheint folgende Meldung:



Prüfen Sie daraufhin nochmals die USB-Verbindung sowie den per "portSwitch" eingestellten USB Port und starten Sie dann die Übertragung erneut.



**qfix robotics GmbH**  
Erich-Rittinghaus-Straße 2/2  
89250 Senden

Tel.: 07307-800321  
Fax.: 07307-800322  
Mail: [info@qfix.de](mailto:info@qfix.de)  
Web: [www.qfix.de](http://www.qfix.de)

---