www.qfix.de

crash-bobby

Anleitung

Dokument Nr. D0305004R3

Inhaltsverzeichnis

1 Einleitung **4**

1.1 Lieferumfang 5

1.2 Zusätzlich benötigte Komponenten 6

1.3 Interessante Erweiterungen 6

1.4 Referenz 6

2 Aufbau des Crash-Bobby **7**

2.1 Schritt 1 - Räder montieren 7

2.2 Schritt 2 - Motoren montieren 8

2.3 Schritt 3 - Stützrad montieren 9

2.4 Schritt 4 - Sensoren montieren 10

2.5 Schritt 5 - Platinenhalter einschrauben 11

2.6 Schritt 6 - Räder anschrauben 12

2.7 Schritt 7 - Stützrad anschrauben 13

2.8 Schritt 8 - Sensoren anschrauben 14

2.9 Schritt 9 - Controllerboard anbringen 16

2.10 Schritt 10 - Motoren und Sensoren anschließen 17

2.11 Roboter betriebsbereit machen 19

3 Installation der Software **20**

3.1 Voraussetzungen 20

3.2 Inhalt der CD 20

3.3 Installation der WinAVR Software 21

3.4 Installation der qfix Beispiel-Software 24

4 Erste Schritte **25**

4.1 Controller in Betrieb nehmen 25

4.2 Programme in den Editor laden 25
4.3 Programm übersetzen 26
4.4 Programm auf den Controller übertragen 28
4.5 Programme selbst erstellen 28

A Controller Beschaltung 29

B Standard-Programm 30

C Befehlsübersicht 31

C.1 Klasse BobbyBoard 32
C.2 Klasse LCD 33

D Übersicht über die mitgelieferten Programme 34

Kapitel 1

Einleitung

Wir gratulieren herzlich zum Kauf des Crash-Bobby Bausatzes!

Die qfix Bausätze sind speziell dazu ausgelegt, Erfahrungen in den Mechatronik-Bereichen Mechanik, Elektronik und Software zu sammeln. Sie sind damit für die Ausbildung an Schule, Hochschule oder im Beruf gleichermaßen geeigent wie für fortgeschrittene Hobby-Bastler, die sich in das spannende Gebiet der Robotik begeben möchten.

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`.

Wir wünschen viel Spaß mit Crash-Bobby!

Das qfix Team

1.1 Lieferumfang

Bitte kontrollieren Sie den Inhalt der Packung! Falls trotz unserer umfangreichen Qualitätskontrollen einzelne Teile fehlen sollten, wenden Sie sich bitte umgehend an uns. Sie erhalten selbstverständlich die fehlenden Teile unverzüglich zugesandt.

Der Crash-Bobby Bausatz enthält folgende Einzelteile:

Mechanik

1x	Basisplatte, blau
3x	Sensorhalter
3x	Distanzstück
2x	Motorhalter
1x	Platte1 20x20x5
1x	Platte2 30x20x5
2x	Rad-Buchse
4x	Platinenhalter
1x	Stahldraht (für Fähnchen)
1x	Aufnahme für Stahldraht
2x	Rad, Durchmesser 50mm
1x	Stützrad inkl. Gewindestift
11x	Schraube M6x10
3x	Schraube M6x20
10x	Schraube M3x6 (für Platine und Sensoren)
4x	Schraube M2x6 (für Motor)
2x	Senkkopfschraube M4 (für Rad)
2x	Gewindestift (für Rad)
2x	Unterlegring (für Stützrad)

Elektrotechnik

1x	Controllerboard
2x	Motor
3x	Distanz-Sensor
3x	Sensor-Kabel
1x	Download Kabel für den Parallel-Port

Software

1x	CD mit qfix Software

Werkzeug

1x	Sechkant-Schlüssel SW 1.5
1x	Sechkant-Schlüssel SW 2
1x	Sechkant-Schlüssel SW 2.5
1x	Sechkant-Schlüssel SW 4

Sonstiges

1x	Aufkleber “Fähnchen”

1.2 Zusätzlich benötigte Komponenten

Für Crash-Bobby kann ein handelsüblicher Modellbau-Akku mit 7.2, 9.6 oder 12 Volt verwendet werden. Es bietet sich jedoch an, das qfix Akku Set (Best.Nr. QAAC000) zu verwenden, da es bereits das erforderliche Adapterkabel und Klettbänder für die einfache Montage der Akkus beinhaltet.

Um den Akku aufzuladen kann ebenfalls ein handelsübliches Ladegerät, beispielsweise das qfix Stecker-Ladegerät (Best.Nr. RBAD000) verwendet werden.

1.3 Interessante Erweiterungen

Grundsätzlich sind sämtliche qfix Bauteile so konstruiert, dass sie optimal zueinander passen.Insbesondere mit den folgenden Erweiterungen macht Crash-Bobby jedoch noch mehr Spaß:

- Liniensensor Set (Best.Nr. QAAB000) das Entlangfahren an einer Linie.
- Bumper Set (Best.Nr. QAAB005), um auf Kontakt mit Hindernissen zu reagieren.
- LC-Display Board (Best.Nr. QAAF001) für das Anzeigen von Information.

Für Programmier-Anfänger bietet sich die Software “Grape” (Best.Nr. QAAH000) an, mit der der Roboter grafisch programmiert werden kann.

1.4 Referenz

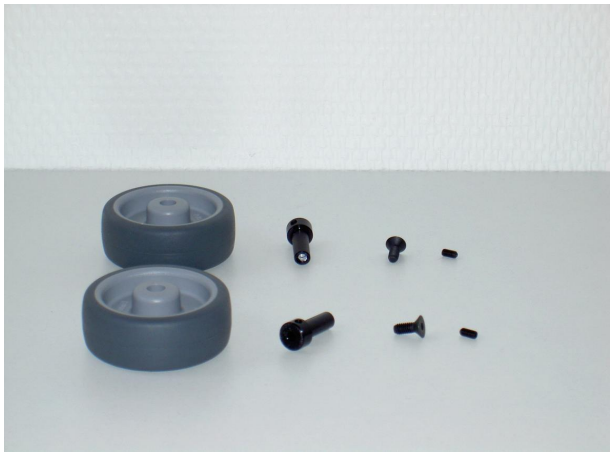
Diese Anleitung bezieht sich auf die Software Version 1.2.

Kapitel 2

Aufbau des Crash-Bobby

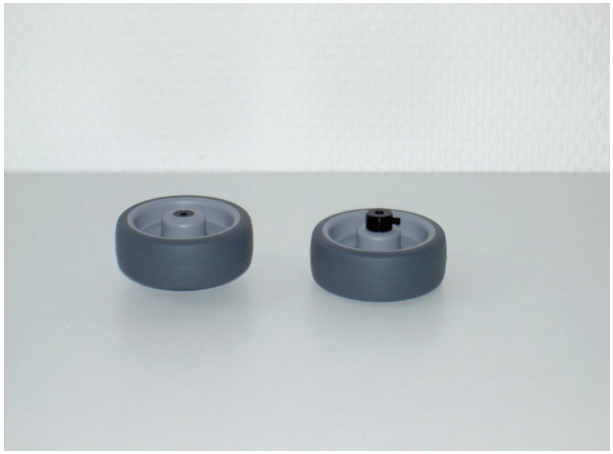
2.1 Schritt 1 - Räder montieren

Für die Montage der Räder werden die Bauteile im folgenden Bild benötigt:



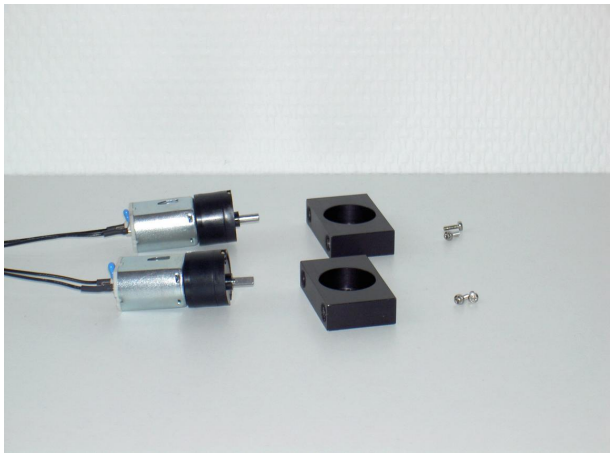
Als erstes wird die kleine Madenschraube seitlich in die Radbuchse eingeschraubt. Dann werden die Radbuchsen so weit wie möglich in die Räder gesteckt, so dass die Senkkopfschraube M4 von der anderen Seite her greift. Diese wird dann mit dem passenden Sechskant-Schlüssel fest angezogen.

Die zusammengesetzten Räder zeigt das nächste Bild:

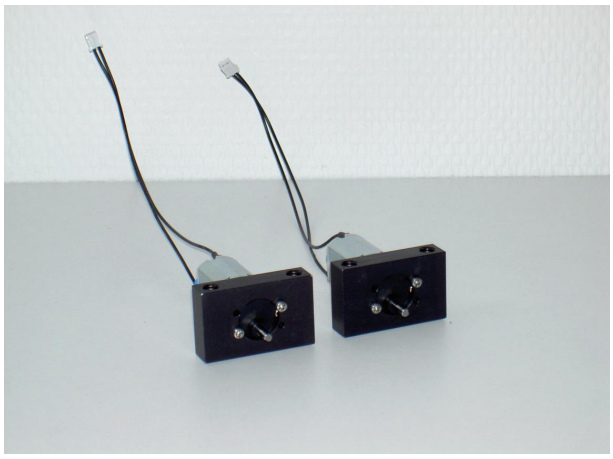


2.2 Schritt 2 - Motoren montieren

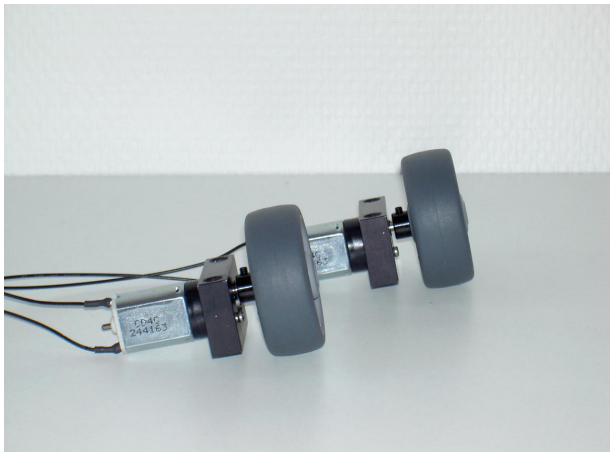
Um die Motoren später an die Basisplatte schrauben zu können, müssen sie vorher an die Motorhalterungen montiert werden. Die benötigten Bauteile zeigt das nächste Bild:



Die Motoren werden in die große Vertiefung der Motorhalterungen gesteckt und mit den kleinen Schrauben M2x6 festgeschraubt. Vorsicht Kunststoffgewinde – die Schrauben also nicht zu fest anziehen!

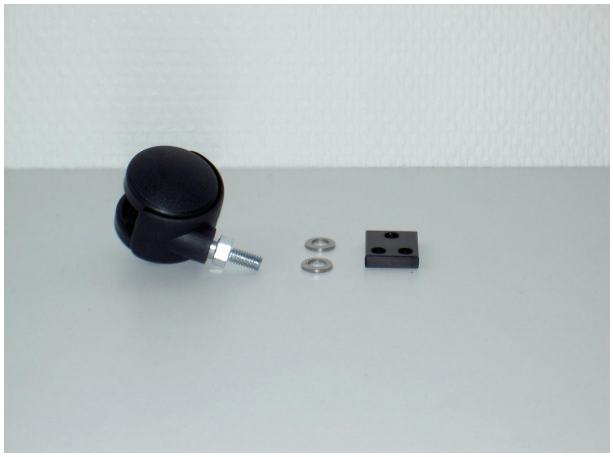


Zuletzt werden die vormontierten Räder auf die Motorachsen gesteckt und die Madenschraube fest angezogen:



2.3 Schritt 3 - Stützrad montieren

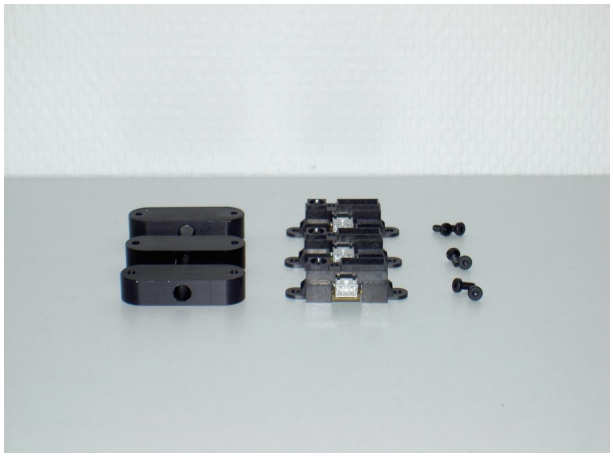
Für die Montage des Stützrades an die rechteckige Halteplatte werden die Bauteile im folgenden Bild benötigt:



Die Unterlegringe werden als Abstandhalter benutzt und einfach auf den Gewindestift gesteckt. Danach wir der Gewindestift fest in das einzeln liegende Gewinde der rechteckigen Platte eingeschraubt.

2.4 Schritt 4 - Sensoren montieren

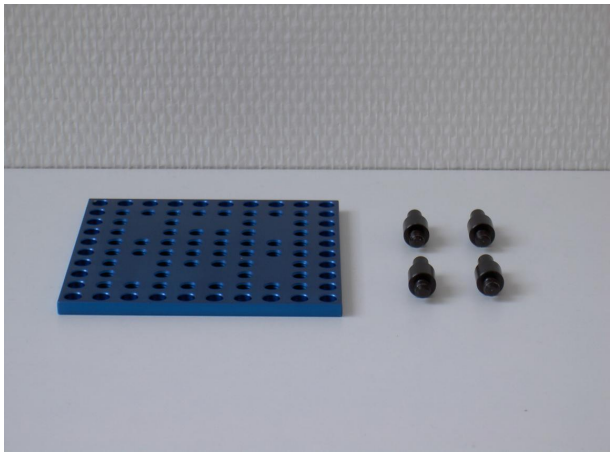
Um die Sensoren später an die Basisplatte schrauben zu können, müssen sie vorher an die Sensorhalterungen montiert werden. Die nötigen Bauteile zeigt das folgende Bild:



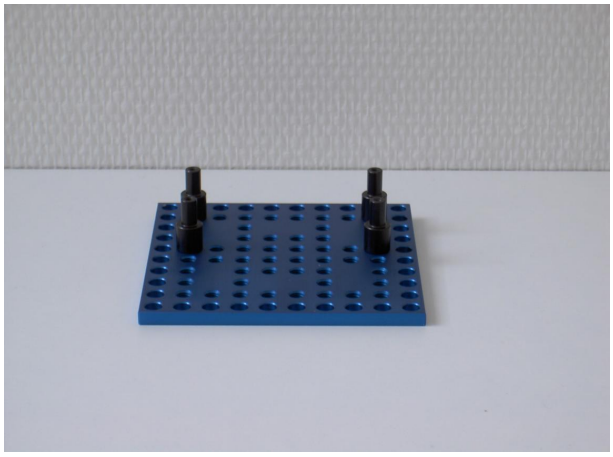
Die drei Sensoren werden mit jeweils zwei Schrauben auf die Halterungen geschraubt. Das nächste Bild zeigt die montierten Sensoren:

2.5 Schritt 5 - Platinenhalter einschrauben

Nun werden die Platinenhalter in die Basisplatte eingeschraubt. Benötigt werden die Basisplatte und vier Platinenhalter:

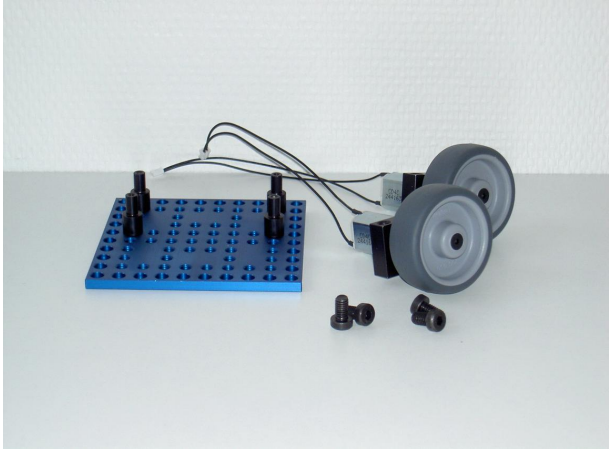


Wie im folgenden Bild zu sehen werden zwei der Halter in die Ecken der äußersten *Gewinde*-Reihen eingeschraubt. Die zwei anderen Halter werden dann mit zwei Gewinden Zwischenraum versetzt befestigt.



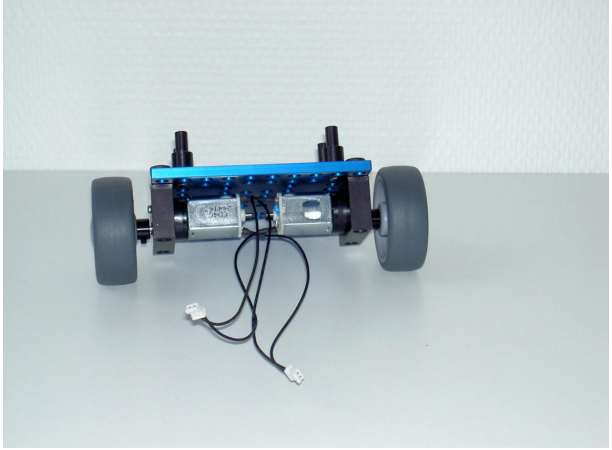
2.6 Schritt 6 - Räder anschrauben

Als nächstes können die vormontierten Räder an der Basisplatte befestigt werden. Es werden folgende Bauteile benötigt:



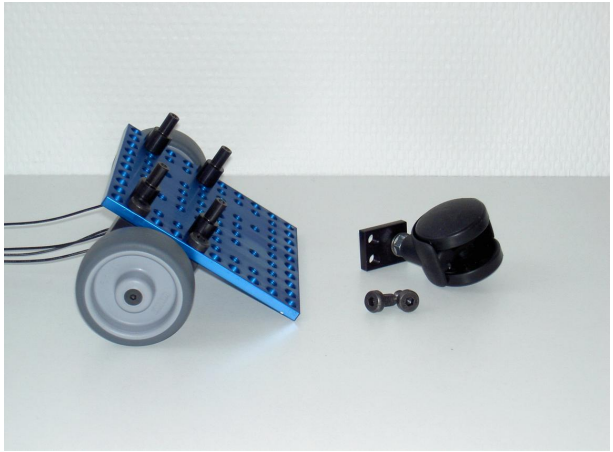
Die Radhalterungen werden direkt neben die Platinenhalter gesetzt und von oben mit den Schrauben M6x10 verschraubt.

Achtung: Die Räder müssen so montiert werden, dass die Rad-Achsen unten aus der Halterung kommen, dass also die “Bodenfreiheit” möglichst groß ist.

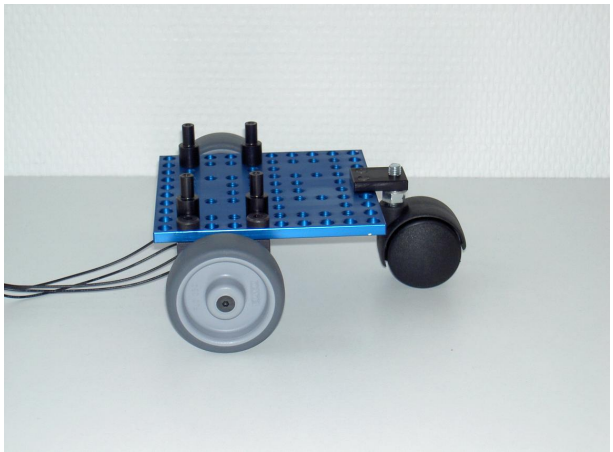


2.7 Schritt 7 - Stützrad anschrauben

Für die Montage des vorbereiteten Stützrades werden ebenfalls noch zwei Schrauben benötigt:

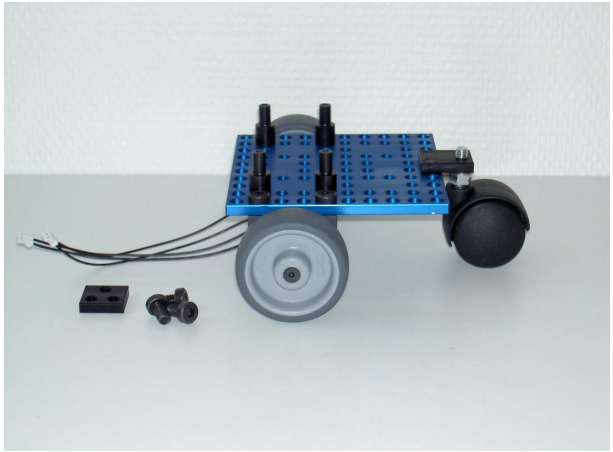


Die Halteplatte wird von oben auf die Basisplatte gelegt und dann von unten mit den Schrauben M6x10 verschraubt.

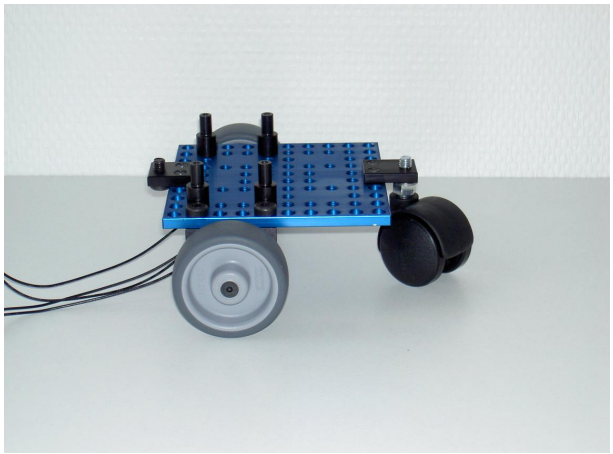


2.8 Schritt 8 - Sensoren anschrauben

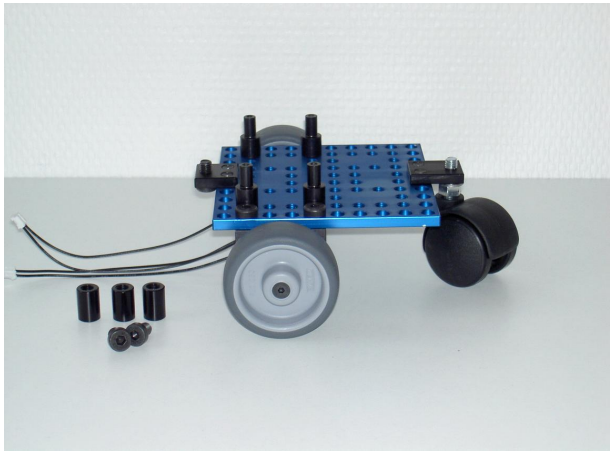
Im nächsten Schritt wird die quadratische Halteplatte verwendet, auf die dann der mittlere Distanzsensor geschraubt wird.



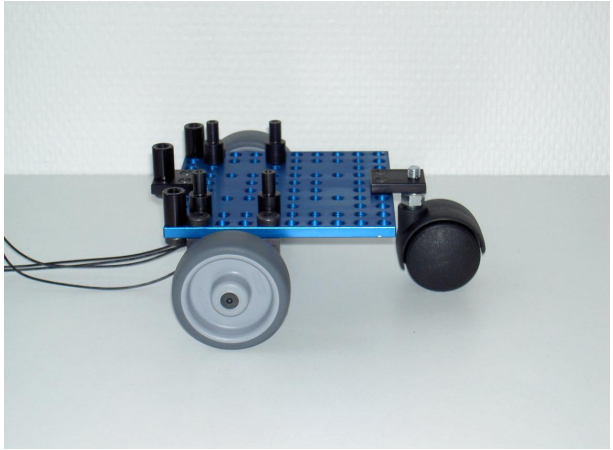
Zuerst wird eine der drei Schrauben M6x10 fest in das einzeln liegende Gewinde der Platte eingeschraubt. Dann wird die Platte gleich wie die Platte des Stützrades oben auf die Basisplatte gelegt und von unten her angeschraubt:



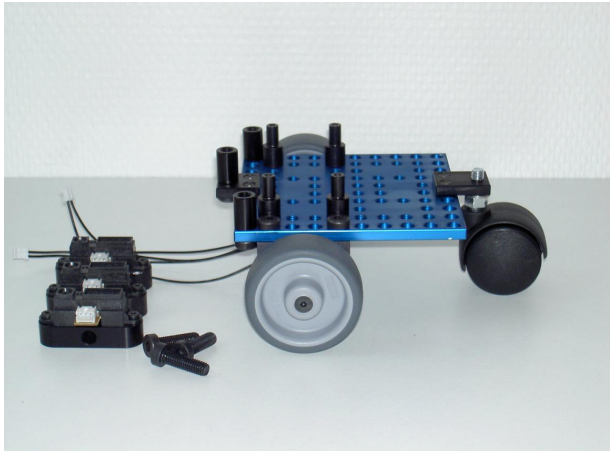
Als nächstes werden die drei Distanzstücke und zwei weitere Schrauben M6x10 benötigt:



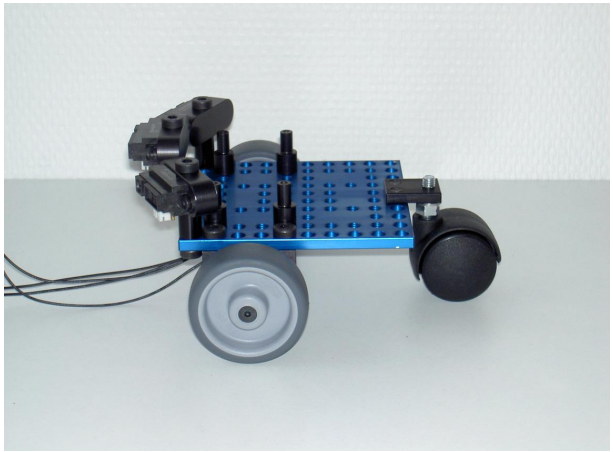
Das Distanzstück für den mittleren Sensor wird auf die nach oben herausstehende Schraube in der quadratischen Platte geschraubt. Die beiden anderen Distanzstücke werden auf die vordere rechte und linke Eckbohrung der Basisplatte gesetzt und von unten verschraubt:



Nun werden die längeren M6x20 Schrauben benötigt, um die vorbereiteten Sensoren auf die Distanzsctücke zu schrauben:

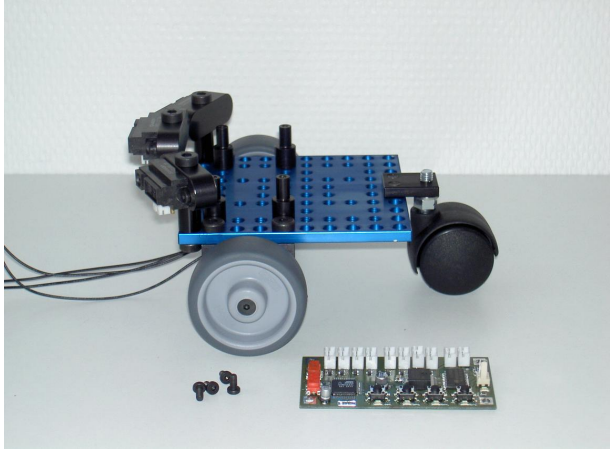


Die Sensoren werden mit dem Anschluss nach unten fest auf die Halterungen geschraubt. Der mittlere Sensor sollte exakt geradeaus, die seitlichen Sensoren in leichtem Winkel nach außen zeigen:

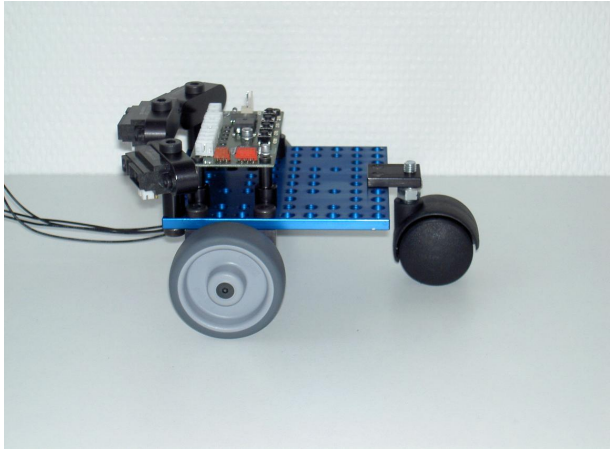


2.9 Schritt 9 - Controllerboard anbringen

Für die Montage des Controllerboards werden 4 Schrauben M3x6 benötigt:

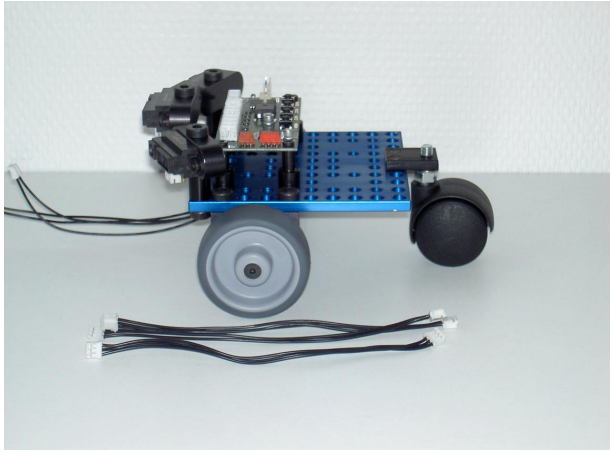


Das Controllerboard wird damit so auf die vier Halterungen geschraubt, dass die Anschlüsse für die Sensoren vorne, also nahe an den Sensoren, liegen:



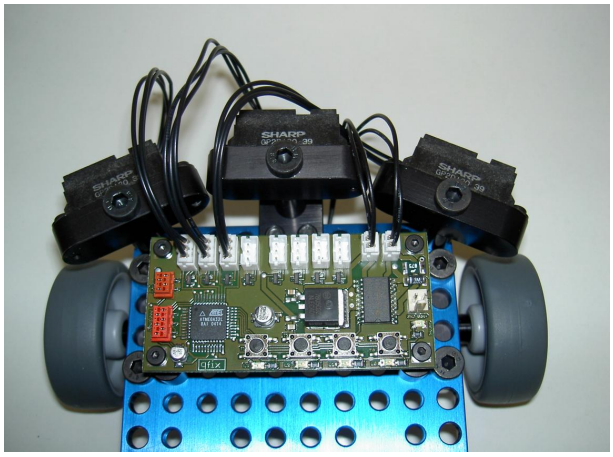
2.10 Schritt 10 - Motoren und Sensoren anschließen

Als letzter Schritt werden nun noch die Motoren und Sensoren an das Controllerboard angeschlossen. Hierzu werden die drei Sensor-Kabel benötigt:

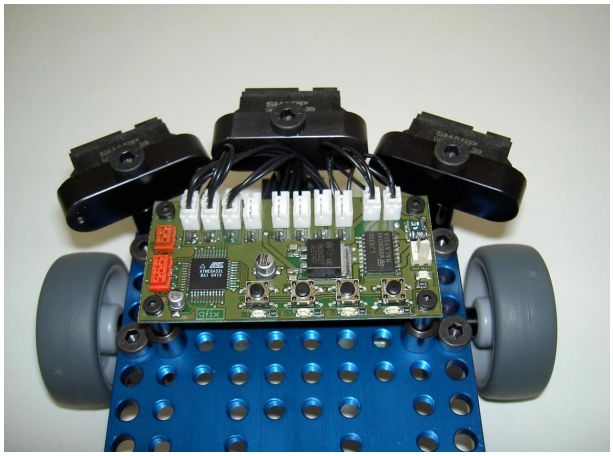


Die Sensoren werden an die Anschlüsse `An0` bis `An3` angeschlossen, die Motoren an die Anschlüsse `Mot0` und `Mot1`. Damit die mitgelieferte Software ohne Änderungen läuft, sollte die Anschlussbelegung folgendermaßen erfolgen:

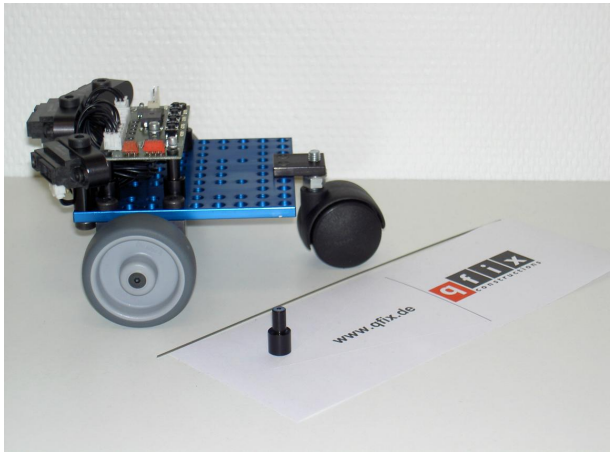
`An0`=Linker Sensor, `An1`=Mittlere Sensor, `An2`=Rechter Sensor
`Mot0`=Linker Motor, `Mot1`=Rechter Motor



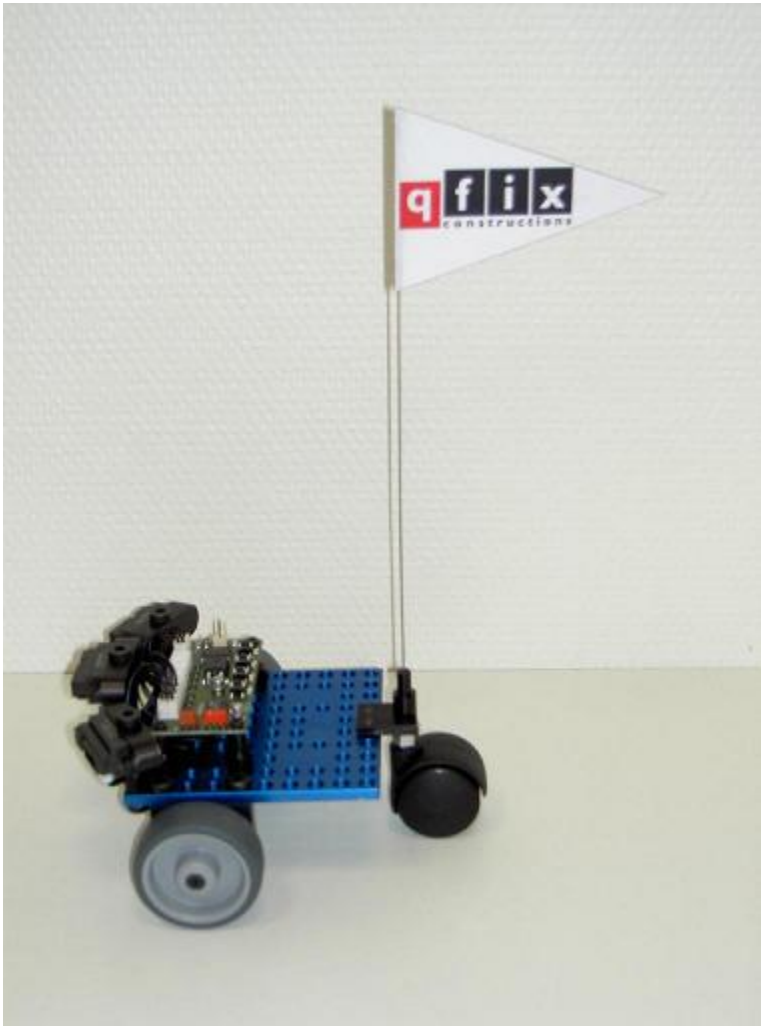
Zuletzt werden die Kabel noch unter der Platine “versteckt”:



Um Crash-Bobby den letzten Schliff zu geben, darf natürlich sein Fähnchen nicht fehlen. Hierzu werden folgende Teile benötigt:

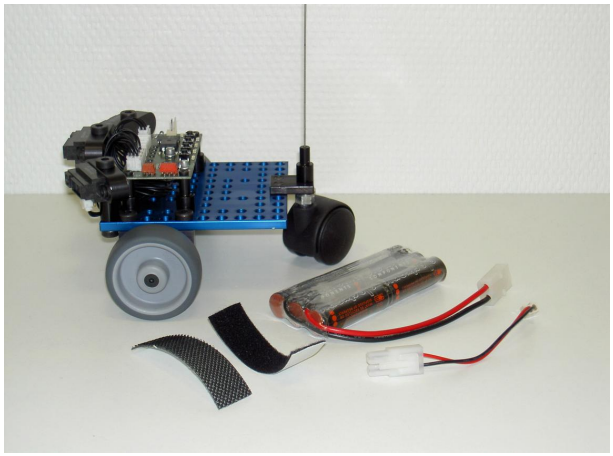


Der Halter für den Stahldraht wir auf das nach oben herausstehende Gewinde des Stützrades geschraubt. Dann kann der Fähnchen-Aufkleber abgezogen und der Stahldraht genau auf die Mittellinie gelegt werden, so dass die beiden Fähnchen-Spitzen exakt aufeinander zu liegen kommen. Nun wird das fertige Fähnchen in den Halter gesteckt - fertig!

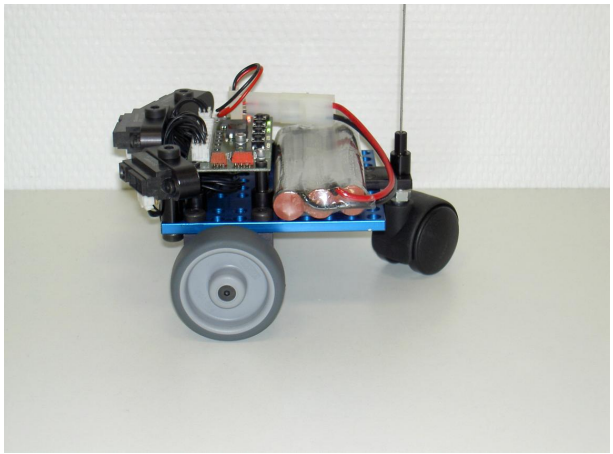


2.11 Roboter betriebsbereit machen

Um den Roboter in Betrieb zu nehmen, wird nun noch ein Akku benötigt. Das folgende Bild zeigt das qfix Akku Set mit seinen Komponenten:



Das Klettband wird einfach auf die Basisplatte geklebt; das Velourband kommt auf die Unterseite des Akkus. Ein Stück Klebeband hilft, das Kabel des Akkus so am Akku zu befestigen, dass es nicht stört. Nach dem Einstecken des Adapterkabels ist der Roboter betriebsbereit.



Kapitel 3

Installation der Software

Vorab: Anleitung und Software dieser CD werden laufend aktualisiert. Bitte prüfen Sie deshalb, ob es sich bei dieser CD um die neueste Version handelt oder ob auf der Internetseite `www.qfix.de` unter “downloads” eine neuere Version zum Herunterladen bereitsteht. Die Versionsnummer der vorliegenden CD entnehmen Sie bitte der Datei `README.txt` bzw. `LIESMICH.txt` im Hauptverzeichnis der CD.

3.1 Voraussetzungen

Die Software kann unter Windows XP oder Windows 2000 installiert werden. Die Verwendung mit älteren Windows-Versionen wurde bisher nicht geprüft.

Für das Laden von Programmen auf den qfix Controller wird der parallele Port (Drucker-Anschluss) benötigt.

3.2 Inhalt der CD

Auf der CD sind folgende Einträge vorhanden:

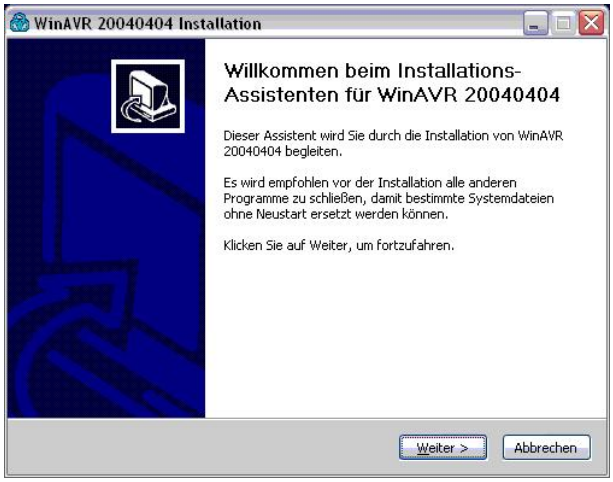
- Datei “README.txt”
- Datei “LIESMICH.txt”
- Datei “install.bat”
- Verzeichnis “software”
- Verzeichnis “doc”
- Verzeichnis “WinAVR”

3.3 Installation der WinAVR Software

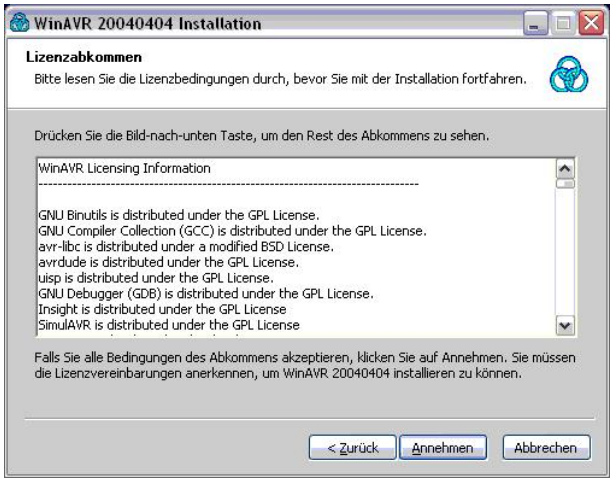
Hinter “WinAVR” verbirgt sich der Compiler für den auf den qfix Boards verwendeten Atmel Controller.

Bitte starten Sie die Installation durch Doppelklicken auf `install.bat`. Als erstes werden Sie dann aufgefordert, die gewünschte Sprache auszuwählen:

Wählen Sie z.B. `Deutsch` und drücken dann `OK`. Wir gehen im folgenden von einer deutschen Einstellung aus. Nun erscheint ein Begrüßungsfenster:



Durch Druck auf `Weiter` kommen Sie zum Lizenzabkommen:

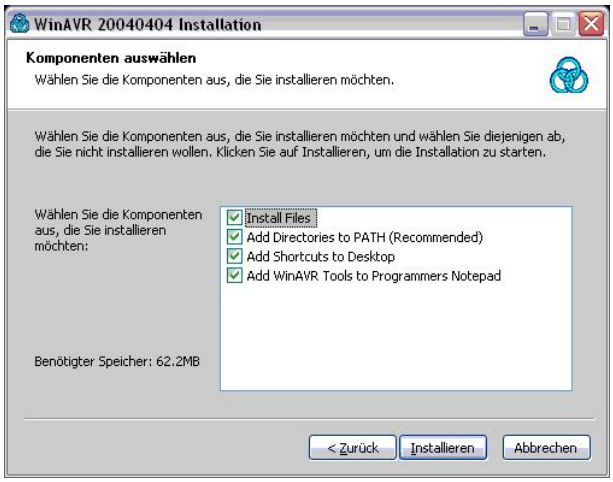


Hier wählen Sie bitte `Annehmen`. Nun erscheint ein Fenster zur Auswahl des Zielverzeichnisses:

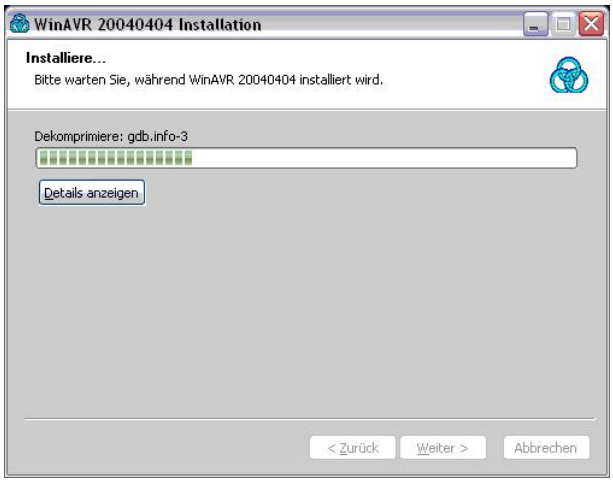
Bitte lassen Sie den vorgeschlagenen Pfad unverändert (C:\WinAVR).

Achtung: Bei falsch gewähltem Pfad compilieren evtl. die mitgelieferten Programme nicht richtig! (WinAVR kann allerdings jederzeit nochmals de-installiert und korrekt installiert werden.)

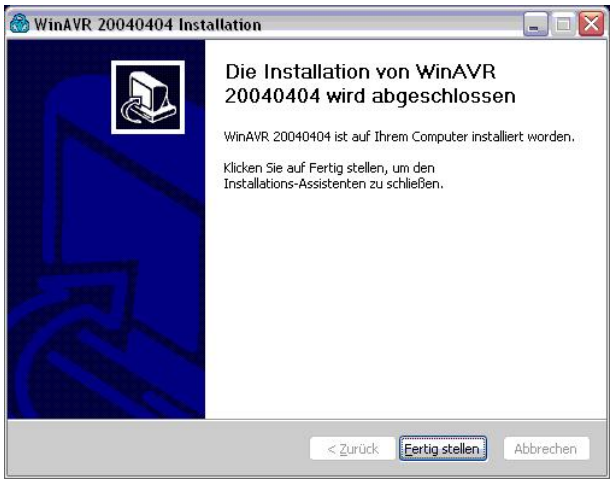
Durch Druck auf `Weiter` bestätigen Sie Ihre Auswahl. Es öffnet sich ein Fenster zur Auswahl der zu installierenden Komponenten:



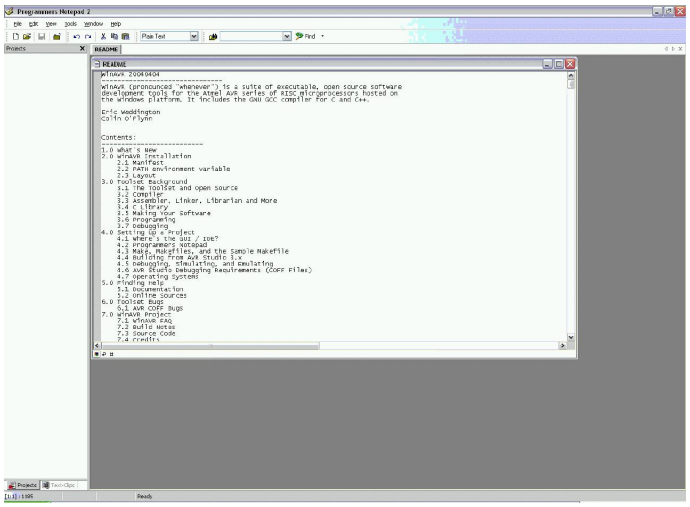
Bitte lassen Sie alle Einträge markiert (bzw. markieren alle Einträge) und drücken Sie dann `Installieren`. Nun öffnet sich ein Fenster, das durch einen Fortschrittsbalken den Status der Installation anzeigt:



Dies kann je nach Rechner ein paar Minuten dauern. Nach der kompletten Installation drücken Sie bitte `Weiter`. Zuletzt öffnet sich noch das letzte Fenster, das Sie mit `Fertig stellen` quittieren können:



In einigen Fällen öffnet sich zusammen mit dem letzten Fenster noch das Programm “Programmers Notepad”, um Ihnen zusätzliche Informationen zu WinAVR anzuzeigen:



Dieses Programm können Sie vor oder nach dem Lesen einfach schließen.

Damit ist der Compiler installiert!

3.4 Installation der qfix Beispiel-Software

Zur Installation der qfix Beispiel-Software genügt es, das komplette Verzeichnis “software” von der CD in ein beliebiges Verzeichnis auf der Festplatte zu kopieren. Sie können so die Software z.B. direkt auf Ihren Desktop ziehen oder aber in Ihrem Ordner “Eigene Dateien” ablegen.

Kapitel 4

Erste Schritte

In diesem Kapitel wird beschrieben, wie man ein Programm zur Bearbeitung in den Editor lädt, das Programm übersetzt und dann vom PC auf den Controller überträgt.

4.1 Controller in Betrieb nehmen

Um den Controller in Betrieb zu nehmen, muss lediglich der Akku angeschlossen werden. Die Akku-Buchse ist verpolungssicher. Nach dem Einstecken des Steckers muss sofort die rote Power-LED leuchten. Leuchtet diese nicht, bitte sofort ausstecken und den Anschluss überprüfen.

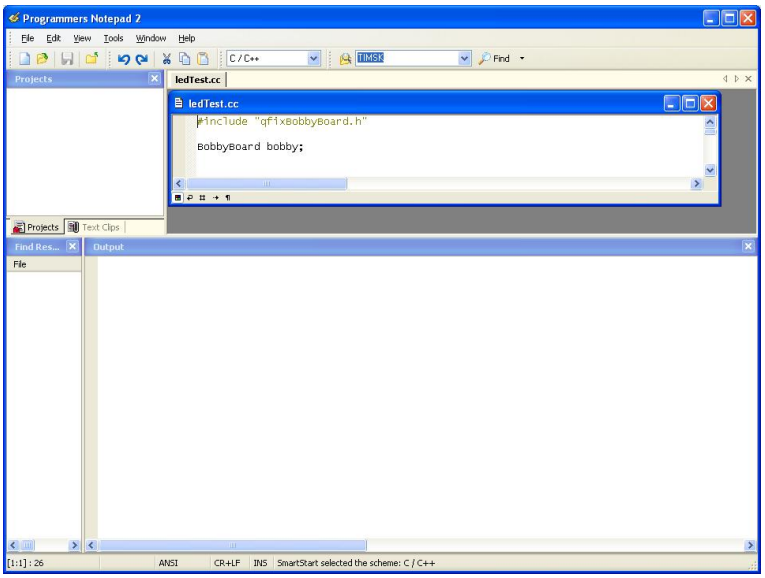
Das Download-Kabel wird an den parallelen Port (Druckeranschluss, LPT1) des PCs gesteckt und der rote Stecker mit der größeren roten Buchse des Controllerboards verbunden. Das Kabel wird nur zum Übertragen eines Programmes auf das Controllerboard benötigt und kann nach dem Übertragen abgesteckt werden.

4.2 Programme in den Editor laden

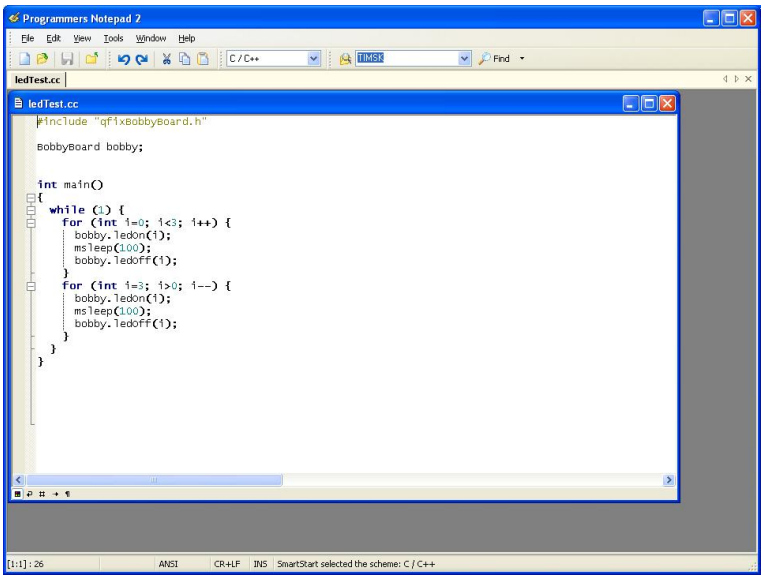
Um mit den mitgelieferten Demo-Programmen arbeiten zu können, sollten Sie das Software-Verzeichnis auf Ihre Festplatte kopiert haben, wie in Kapitel 3.4 beschrieben.

Laden Sie nun eines der Demo-Programme in den Editor, indem Sie die Entwicklungsumgebung “Programmers Notepad” aufrufen (z.B. über Start→WinAVR→Programmers Notepad) und darin über File→Open beispielweise die qfix Demo Datei `ledTest.cc` öffnen.

Beim ersten Start des “Programmers Notepad” zeigt sich dieses mit einer ganzen Reihe überflüssiger Fenster:

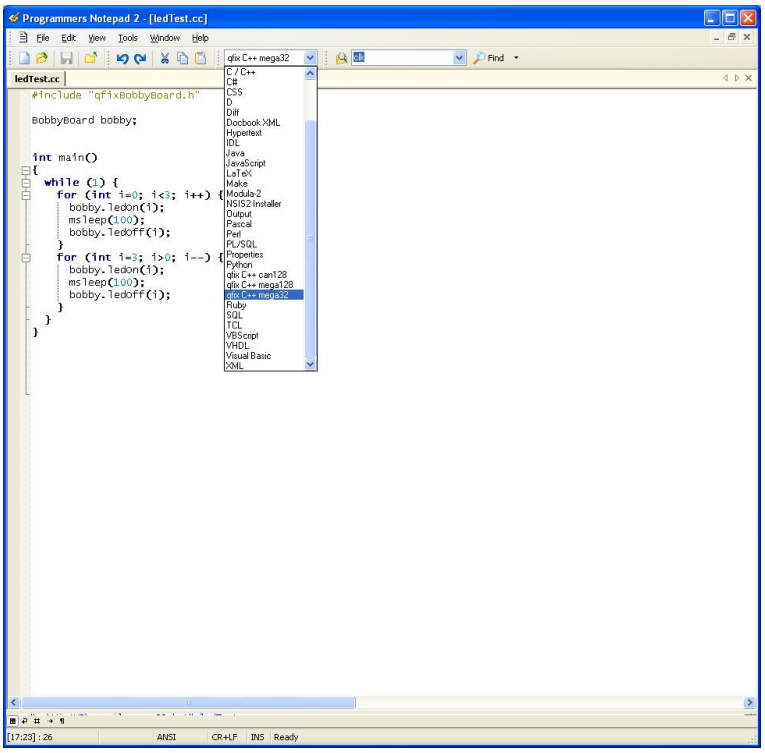


Diese schließen Sie bitte alle, bis auf das Fenster mit dem eigentlichen Programm `ledTest.cc`. Der Editor sieht dann folgendermaßen aus:

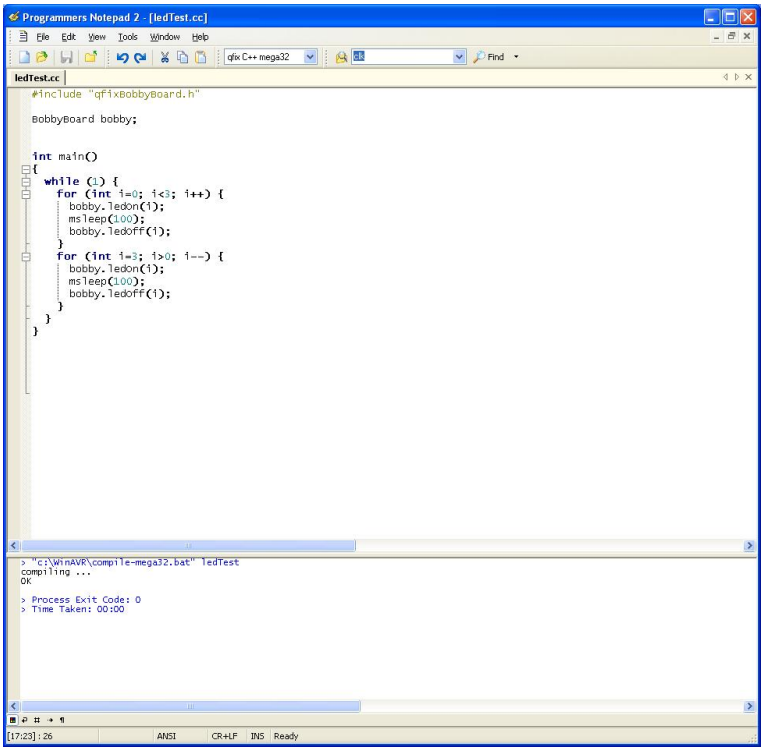


4.3 Programm übersetzen

Bevor das geladene Programm für den Controller übersetzt werden kann, muss der entsprechende Controller ausgewählt werden. Die qfix Bausätze verwenden momentan drei verschiedene Controller der Firma Atmel: mega32, mega128 und AT90can128. Für das “BobbyBoard” des Crash-Bobby Bausatzes wählen Sie bitte in der linken Auswahlliste des Editors den Eintrag `qfix C++ mega32` aus:



Drücken Sie nun die Taste F5, um die Datei zu kompilieren. An der Unterseite des Editor-Fensters öffnet sich ein Ausgabefenster, das den Status der Kompilierung anzeigt:

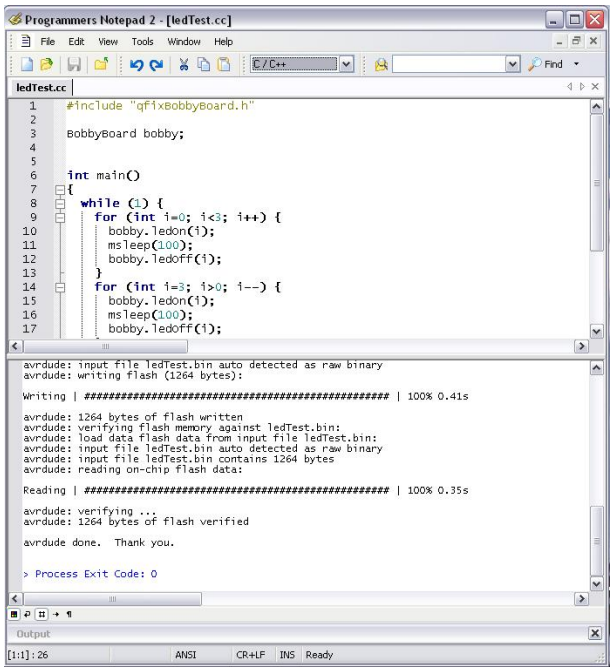


Wie man in der obersten Zeile des Ausgabefensters sieht, wurde für die Kompilierung das Tool `c:→WinAVR→compile-mega32.bat` aufgerufen. Erscheint wie im obigen Beispiel die Meldung `compiling ...` und dann `OK`, war die Übersetzung erfolgreich und das Programm kann auf den Controller übertragen werden. Falls das Programm noch Fehler enthält, werden diese ebenfalls (auf die etwas kryptische Art eines Compilers) hier angezeigt.

4.4 Programm auf den Controller übertragen

Drücken Sie nun die Taste F6, um die kompilierte Datei auf das Controllerboard zu übertragen. Auch hier ist wieder darauf zu achten, dass der korrekte Controller-Typ ausgewählt ist, also `qfix C++ mega32`. Die Übertragung dauert einige Sekunden, während im Ausgabefenster die Meldungen des Übertragungsprogrammes `avrdude` angezeigt werden.

Nach erfolgreicher Übertragung beginnt das Controllerboard sofort, das Programm auszuführen. Im Beispiel blinken die vier LEDs.



Genauso kann auch mit den anderen Beispielprogrammen verfahren werden. Eine Übersicht über die mitgelieferten Programme befindet sich im Anhang.

Sollte das Übertragen der Progamme nicht funktionieren, kann es daran liegen, dass die parallele Schnittstelle im BIOS des Rechners falsch eingestellt ist. Stellen Sie die Schnittstelle LPT1 dann auf den Modus “ECP”.

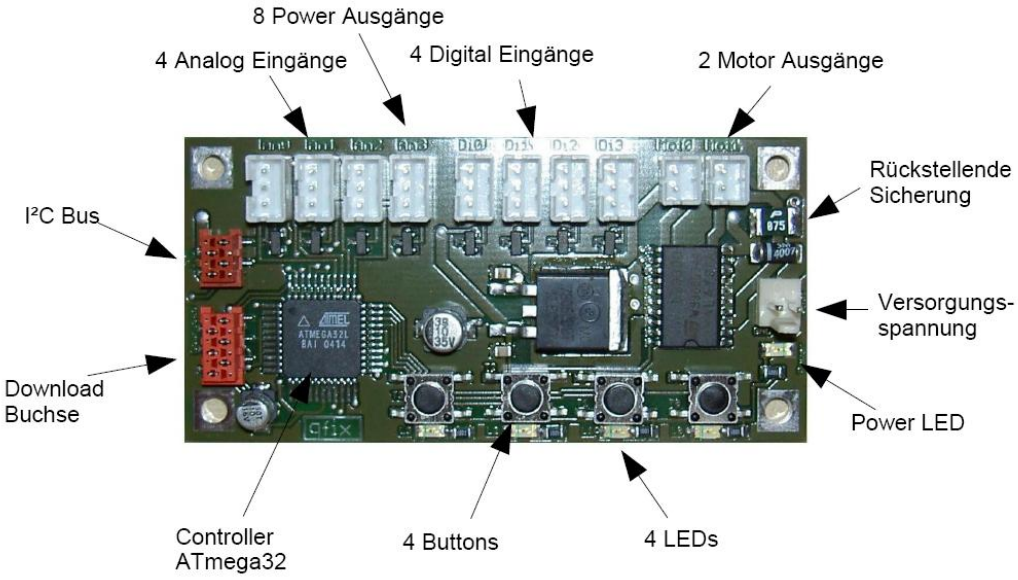
4.5 Programme selbst erstellen

Um selbst Programme zu erstellen, ist es am einfachsten, ein bereits vorhandenes Programm zu kopieren und ausgehend von diesem zu beginnen.

Für die Programmierung des Controllerboards steht die Klasse `BobbyBoard` zur Verfügung, deren Methoden im Anhang C.1 ausführlich erläutert werden.

Anhang A

Controller Beschaltung



Taktfrequenz: 8 MHz

Beschaltung der Ein-/Ausgänge (weiße 3-polige Buchsen):

- Der äußerste Pin ist jeweils der Ausgang. Er kann durch den `powerOn` bzw. `powerOff` Befehl an- und ausgeschaltet werden. Ist er angeschaltet, führt er eine Spannung von 5V und kann bis zu 100mA steuern.
- Der mittlere Pin ist der Massepin.
- Der jeweils innere Pin is der Eingang. Bei den analogen Ports kann hier eine beliebige zwischen 0 und 5V anliegen, bei den digitalen Ports entweder 0 oder 5V.

Anhang B

Standard-Programm

```
#include "qfixBobbyBoard.h"

BobbyBoard bobby;

int main()
{
  /* Hier stehen die Anweisungen */
}
```

Das Standard Programm kann als Ausgangspunkt für alle Übungen benutzt werden. Es lädt zuerst die benötigte Bibliothek `qfixBobbyBoard.h` und legt dann ein Objekt `bobby` an, das für die weiteren Befehle benutzt werden kann.

Durch den Ausdruck
`bobby.`<*Befehl*>;
können so sämtliche Befehle der Klasse `BobbyBoard` (siehe Anhang C) ausgeführt werden.

Anhang C

Befehlsübersicht

Jede der folgenden Tabellen beschreibt jeweils eine Klasse sowie deren Methoden. Um diese Methoden aufrufen zu können, muss ein Objekt (eine Instanz) der Klasse existieren. Als Beispiele hierfür dienen die Demo-Programme auf der CD.

Die Spalte *RWert* beschreibt für jede Methode den Typ des Rückgabewerts. Folgende Typen werden hierbei verwendet:

Wertebereich	*Erklärung*
`void`	Kein Wert (wird z.B. bei Funktionen oder Methoden ohne Rückgabewert benutzt)
`bool`	Boolscher Wert (`true` oder `false`)
`int`	Ganzzahliger Wert (-32767 – 32768)
`char`	Zeichen (z.B. `'A'`)
`char*`	Zeiger auf ein Zeichen. Steht dies in einer Funktion, kann als Parameter ein String übergeben werden (z.B. `"Hallo"`)

C.1 Klasse BobbyBoard

Klasse: `BobbyBoard`		Datei: `qfixBobbyBoard.h`
RWert	*Funktionsname*	*Erklärung*
	`BobbyBoard()`	Diese Methode wird durch das Anlegen des Objektes automatisch aufgerufen ("Konstruktor") und initialisiert das Objekt folgendermaßen: Die Motoren werden auf 0 gesetzt, alle LEDs sind aus, alle Power-Ausgänge sind angeschaltet.
`void`	`ledOn(int i)`	Schaltet die LED mit Index `i` an. `i` muss im Bereich 0 bis 3 liegen.
`void`	`ledOff(int i)`	Schaltet die LED mit Index `i` aus. `i` muss im Bereich 0 bis 3 liegen.
`void`	`ledsOff()`	Schaltet alle LEDs aus.
`void`	`led(int i, bool state)`	Schaltet die LED mit dem Index `i` an, falls `state=true`, bzw. aus, falls `state=false`. `i` muss im Bereich 0 bis 3 liegen.
`bool`	`button(int i)`	Liefert `true` zurück, falls der Button mit dem Index `i` gedrückt ist, ansonsten `false`. `i` muss im Bereich 0 bis 3 liegen.
`void`	`waitForButton(int i)`	Wartet, bis der Button mit dem Index `i` gedrückt und wieder losgelassen wurde. `i` muss im Bereich 0 bis 3 liegen.
`void`	`motor(int i, int speed)`	Setzt den Motor mit dem Index `i` auf den Wert `speed`, der im Bereich -255 bis +255 liegen muss. `i` muss 0 oder 1 sein.
`void`	`motors(int motor0, int motor1)`	Setzt beide Motoren auf die entsprechenden Werte `motor0` und `motor1`, die im Bereich -255 bis +255 liegen müssen.
`void`	`motorsOff()`	Schaltet beide Motoren aus.
`int`	`analog(int i)`	Liefert den Wert des Analog-Eingangs mit dem Index `i` zurück. Der Rückgabewert liegt im Bereich 0-255. `i` muss im Bereich 0 bis 3 liegen.
`bool`	`digital(int i)`	Liefert `true` zurück, falls der Digital-Eingang mit dem Index `i` High ist, ansonsten `false`. `i` muss im Bereich 0 bis 3 liegen.
`void`	`powerOn(int i)`	Schaltet den Power-Ausgang mit Index `i` an. `i` muss im Bereich 0 bis 3 liegen.
`void`	`powerOff(int i)`	Schaltet den Power-Ausgang mit Index `i` aus. `i` muss im Bereich 0 bis 3 liegen.
`void`	`power(int i, bool state)`	Schaltet den Power-Ausgang mit dem Index `i` an, falls `state=true`, bzw. aus, falls `state=false`. `i` muss im Bereich 0 bis 3 liegen.
`void`	`ledMeter(int i)`	Zeigt den übergebenen Wert `i` im Bereich 0-255 mittels der LEDs an.
`void`	`sleep(int s)`	Wartet `s` Sekunden lang.
`void`	`msleep(int ms)`	Wartet `ms` Millisekunden lang.

C.2 Klasse LCD

Die Klasse LCD unterstützt den Betrieb der qfix LC-Displays. Das Display muss über den I2C-Bus an das Controllerboard angeschlossen sein.

Klasse: `LCD`		Datei: `qfixLCD.h`
RWert	*Funktionsname*	*Erklärung*
	`LCD()`	Der Konstruktor initialisiert die Verbindung zum LC-Display, löscht das Display und setzt den Cursor auf (0,0).
`void`	`clear()`	Löscht das Display.
`void`	`print(int i)`	Druckt die Zahl `i` an der aktuellen Cursor-Position aus.
`void`	`print(char* s)`	Druckt den String `s` an der aktuellen Cursor-Position aus.
`void`	`print(int row, int col, int i)`	Druckt die Zahl `i` in Zeile `row` in Spalte `col` aus.
`void`	`print(int row, int col, char* s)`	Druckt den String `s` in Zeile `row` in Spalte `col` aus.
`void`	`lightOn()`	Schaltet die Hintergrundbeleuchtung an.
`void`	`lightOff()`	Schaltet die Hintergrundbeleuchtung aus.

Anhang D

Übersicht über die mitgelieferten Programme

Auf der mit dem Crash-Bobby gelieferten CD befinden sich folgende Programme:

analogTest

Dieses Programm gibt einen Wert am Analog-Port 0 über die LEDs aus. Bei einer Spannung von 0V (interner Wert 0) leuchtet keine LED, bei einer Spannung von 5V (interner Wert 255) leuchten alle vier LEDs. Dazwischen leuchten entsprechend viele LEDs auf.

bobbyTest

Dieses Programm läßt den Crash-Bobby herumfahren und weicht Hindernissen aus. Hierzu müssen die drei Distanzsensoren folgendermaßen mit dem Controller verbunden werden:

- Linker Sensor: An0
- Mittlerer Sensor: An1
- Rechter Sensor: An2

Die Motoren müssen folgendermaßen angeschlossen sein:

- Linker Motor: Mo0
- Rechter Motor: Mo1

Benutzung:

Anfangs leuchtet die LED 0. Nun müssen die Sensoren auf den “kritischen Abstand” kalibriert werden. Hierzu hält man die Hand im Abstand von ca. 10-20 cm vor den mittleren Sensor und drückt Button 0. Nach Druck auf Button 2 fährt der Roboter los.

Mit Button 0 wird er wieder angehalten, Button 2 startet ihn wieder. Für eine Neu-Kalibrierung muss das Akku-Kabel kurz abgesteckt werden.

buttonTest

Dieses Programm läßt bei Druck auf einen der Buttons die entsprechende LED aufleuchten.

digitalTest

Dieses Programm prüft alle vier Digital-Eingänge und läßt bei Anliegen einer Spannung die entsprechende LED aufleuchten.

ledTest

Dieses Programm läßt die vier LEDs ähnlich dem Knight-Rider Auto blinken.

motorTest

Dieses Programm steuert beide Motoren mit 4 unterschiedlichen Geschwindigkeiten in beide Richtungen an. Die 4 Möglichkeiten können über die Buttons gewählt werden.

myTest

Dieses Programm hat keine Funktion und enthält lediglich ein leeres Grundgerüst für eigene Programme. Das ganze Verzeichnis sollte dazu kopiert werden.

powerOutTest

Dieses Programm entspricht dem Programm ledTest, nur dass anstatt der LEDs die Power Ausgänge geschaltet werden.

slaveBoardTest

Dieses Demoprogramm zeigt, wie zwei BobbyBoards auf einfache Weise ber den I2C-Bus miteinander verbunden werden knnen. Ein Board fungiert dann als “Master” und eines als “Slave”. Das Demoprogramm besteht dementsprechend aus zwei Teilen: Dem Masterprogramm `master.cc` und dem Slaveprogramm `slaveBoard.cc`.

Das Slaveprogramm wir compiliert und auf das Slaveboard geladen. Der Slave wartet lediglich auf Anweisungen (z.B. “Motor an”, “Button gedrckt?”), die vom Master ber den Bus bertragen werden und reagiert entsprechend darauf. Der Programmtext muss also nicht verstanden werden (“Blackbox-Prinzip”).

Das Masterprogramm demgegenber ist recht das eigentiche Steuerprogramm. Es legt zwei Objekte an (`bobby` und `slave`) und fragt die Buttons des Masterboards ab, um dann auf dem Slaveboard eine entsprechende Aktion auszufhren:

- Master Button 0 schaltet Slave LED 0
- Master Button 1 schaltet Slave Motor 0
- Master Button 2 fragt Slave Button 0 ab und gibt den Zustand auf Master LED 2 aus.
- Master Button 3 fragt Slave Digital 0 ab und gibt den Zustand auf Master LED 3 aus.

vehikel

Dieses Programm entspricht dem Braitenberg’schen Vehikel. Es fährt umher und weicht Hindernissen aus. Es ähnelt dem Progamm bobbyTest, ist aber sehr viel einfacher aufgebaut und benutzt lediglich 2 Sensoren.

Hierzu müssen die drei Distanzsensoren folgendermaßen mit dem Controller verbunden werden:

- Linker Sensor: An0
- Rechter Sensor: An1

Die Motoren müssen folgendermaßen angeschlossen sein:

- Linker Motor: Mo0
- Rechter Motor: Mo1