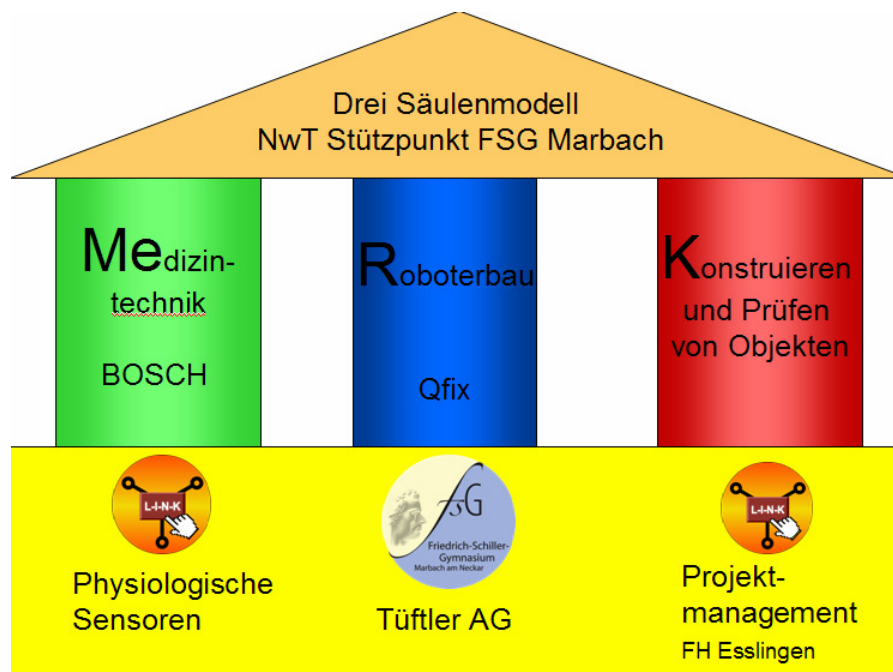


NwT – Stützpunkt MeRK

Erste Schritte zum Roboterbau



Vorbemerkungen

Das vorliegende Manuskript ist sowohl für Lehrkräfte als auch für Lernende, die mit dem Gerätesatz arbeiten wollen, gedacht. Das Papier kann das Unterrichtsmaterial nur ergänzen und soll eine Hilfe für den Aufbau und die Programmierung der Roboter sein. Insbesondere folgt es keinem Unterrichtsablauf, sondern der Fachsystematik des Baukastens.

Für die Programmierung sind elementare Kenntnisse in der Sprache C++ nötig. Hier wird nur auf die Spezialbefehle, die zur Steuerung des Roboters benötigt werden, eingegangen.

Inhaltsverzeichnis

1. Bauanleitung Fahrzeug (ohne Sensoren).....	3
Räder auf Motoren montieren	3
Stützrad montieren	3
Aufbau der mechanischen Teile des Fahrzeugs	4
Controllerboard anbringen	5
2. Softwareinstallation	6
Reihenfolge.....	6
Programmierung.....	6
3. Tastsensor	7
Sensoren montieren und anschließen	7
Sensoren testen	7
4. Abstandsensor.....	8
Sensoren montieren und anschließen	8
Sensoren testen	8
5. IR-Reflex-Sensor	9
Sensoren montieren und anschließen	9
Sensoren testen	9
6. Bestückungsplan	10
Schraubentüte M2 / M3	11
Schraubentüte M6.....	11
Koffer für den Klassensatz	11
7. Übersicht über die Bauteile	12
8. Datenblatt Miniboard.....	14
9. Grundelemente von C++	15
10. Programmierung Methoden der Klasse MiniBoard	16

1. Bauanleitung Fahrzeug (ohne Sensoren)

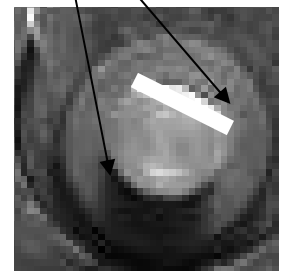
Räder auf Motoren montieren

Bauteile

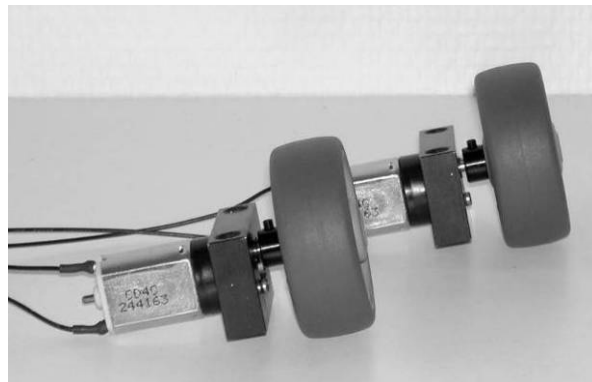
- 2 Motore
- 2 Madenschrauben M3x5

Montage

1. Die Madenschrauben werden auf die Achsen aufgesetzt und mit drei Umdrehungen fixiert.
2. Dann werden die Räder so auf die Achsen gesteckt, dass die Madenschrauben auf die abgeflachte Seite der Achse treffen.
3. Die Madenschrauben werden angezogen.



Ergebnis



Stützrad montieren

Bauteile

- 1 Halteplatte 20 x 25
- 1 Stützrad
- 2 Beilagscheiben D6

Zusatzwerkzeug:
Gabelschlüssel

Montage

1. Die Beilagscheiben werden als Abstandhalter benutzt und auf den Gewindestift gesteckt.
2. Danach wird der Gewindestift fest in das einzeln liegende Gewinde der rechteckigen Halteplatte eingeschraubt.

Ergebnis



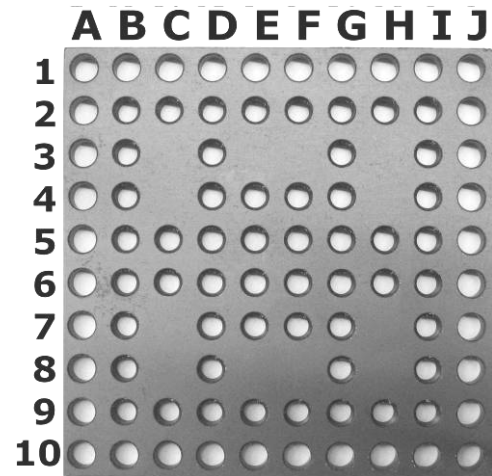
Aufbau der mechanischen Teile des Fahrzeugs

Bauteile

- 2 vormontierte Motoren
- 1 vormontiertes Stützrad
- 1 Basisplatte
- 4 Platinenhalter



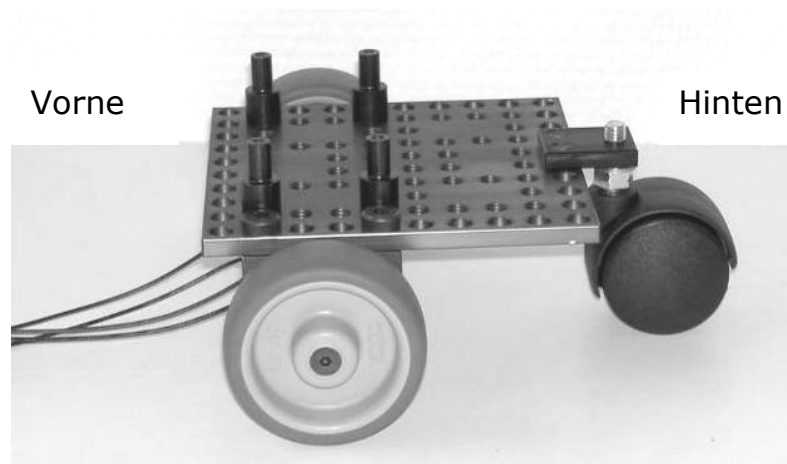
- 6 Inbusschrauben M6x10



Montage

1. Die Platinenhalter werden in die Gewinde B2, B5, I2 und I5 eingeschraubt.
2. Die Motoren werden von unten mit den Schrauben M6x10 in die Bohrungen A2 & A5 bzw. J2 & J5 neben den Abstandhaltern angeschraubt. Achtung: Die Räder müssen so montiert werden, dass die Radachsen unten aus der Halterung kommen, dass also die "Bodenfreiheit" möglichst groß ist.
3. Die Anschlusskabel sollten von vorne frei zugänglich sein.
4. Die Halteplatte des Stützrades wird von oben mittig auf die Basisplatte bei E10 & F10 gelegt und dann von unten mit den Schrauben M6x10 verschraubt.

Ergebnis

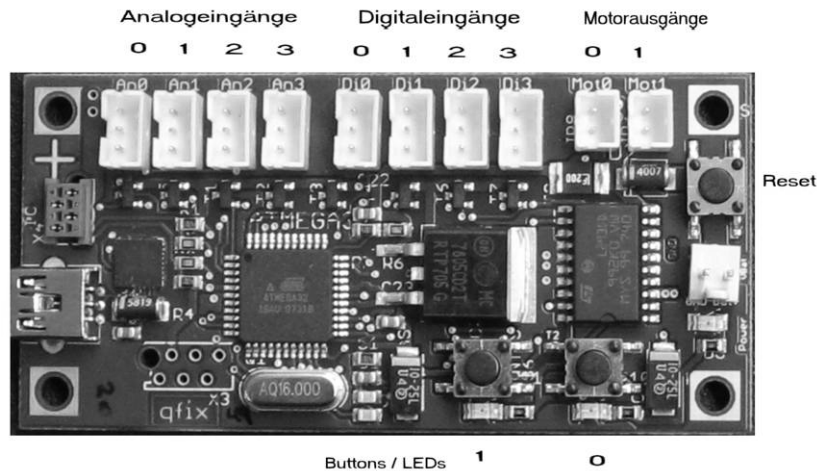


Controllerboard anbringen

Motoren anschließen und testen

Bauteile

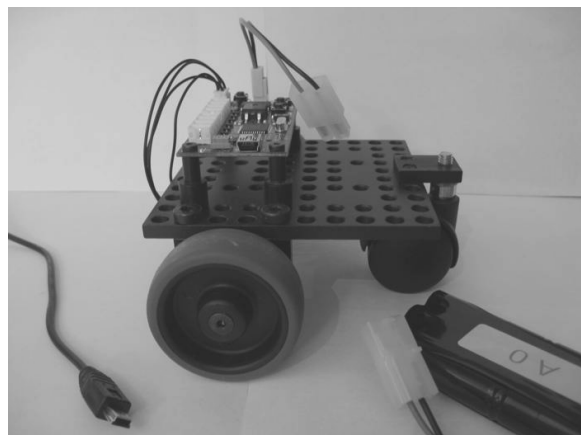
- 1 Controllerboard
- 4 Inbusschrauben M3x6
- 1 12V Versorgungskabel
- 1 Akku



Montage

1. Das Controllerboard wird mit den Inbusschrauben M3x6 so auf die vier Platinenhalter geschraubt, dass die Buttons in der Fahrzeugmitte liegen.
2. die Motoren an die Anschlüsse Ausgänge anschließen
Motorausgang 0= linker Motor,
Motorausgang 1= rechter Motor
3. Software und Treiber nach Anleitung installieren
4. Rechner und Board mit dem USB-Kabel verbinden
5. Die Datei **MotorTestMeRK.cc** öffnen mit der Taste F5 in eine ausführbare Datei übersetzen (compilieren) und mit F6 aufs Board übertragen. Der Programmstart und das Überspielen eines Programms sind nur nach Drücken der Reset-Taste möglich. Dieser Zustand wird durch das schnelle Blinken von LED 0 angezeigt.
6. das 12 V Versorgungskabel wird mit dem Akku verbunden und am Controllerboard eingesteckt.
7. Test der Motoren durch Drücken von:
 - Button 0 linker Motor vorwärts
 - Button 1 rechter Motor vorwärts
 - Button 0 und 1 beide Motore vorwärts

Ergebnis



Programmierung:

Der Befehl `car.button(i)` liefert den Wert „true“, wenn der Button mit der Nummer `i` gedrückt ist.

2. Softwareinstallation

Wichtig: USB Adapter erst in Schritt 4 anschließen.

Dir hier beschriebene Software inklusive aller Handbücher ist downloadbar (Adresse in Moodle, Umfang ca. 60MB)

Reihenfolge

1. Start des Programmes **qfixSoftware-1.4.0.msi**.
2. Im Startmenue unter → **Alle Programme** → **qfix Software** den Menuepunkt → **install USB driver** aufrufen
3. NEU-Start des PC
4. USB Adapter anschließen
5. Im **Gerätemanager** unter **Anschlüsse (COM und LPT)** den Eintrag **silicon Labs CP210x USB to UART bridge** auswählen und **COM-Port** ablesen.
6. Im Startmenue unter → **Alle Programme** → **qfix Software** den Menuepunkt **portswitch** aufrufen und der USB Schnittstelle den eben ermittelten COM-Port zuweisen.
(bei PC mit mehreren USB Anschlüssen, kann jeder USB eine andere Einstellung erfordern, deshalb bitte immer denselben USB nutzen oder ab Schritt 5 erneut durchführen.)

Programmierung

Lege im Heft eine Übersicht der Programmbefehle (Glossar) an. Achte dabei auf Groß-Kleinschreibung.

Rufe im Startmenue unter → **Alle Programme** → **qfix Software** → **examples** → **MiniBoard** die Datei myTest.cc auf (sie wird mit dem Programm **Programmers Notepad** bearbeitet).

Speichere sie **sofort** unter einem neuen Namen (ohne Umlaute und ohne Sonder- oder Leerzeichen) in deinem FSG-Verzeichnis ab.

Durch das neu zu schreibende Programm wird der Abschnitt ersetzt.

```
// my program  
// for example:  
// ...
```

hinter jeder Zeile soll ein Kommentar eingefügt werden Dieser beginnt mit // und wird nicht übersetzt.

Mit der Taste F5 wird das Programm in eine dem, Microcontroller verständliche Datei übersetzt (compiliert).

Bei Erfolg erscheint am Bildschirm die Meldung **Process Exit Code: 0**

Nach Reset am Miniboard (Button 0 blinkt schnell) wird mit der Taste F6 das Programm an den Microcontroller übertragen.

Bei Erfolg erscheint am Bildschirm die Meldung **Process Exit Code: 0**

3. Tastsensor

Sensoren montieren und anschließen

Bauteile

- 2 Tastsensoren
- 2 Halteplatten (2 Gewinde M6, 2 Gewinde M2)
- 4 Kreuzschlitzschrauben M2x10
- 4 Inbusschrauben M6x10

Zusatzwerkzeug:

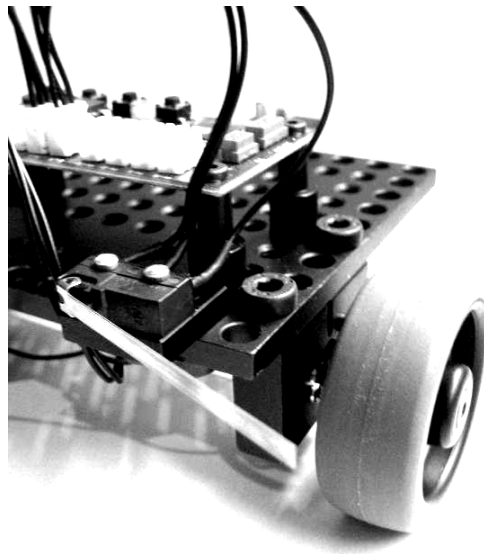
Kreuzschlitzschraubendreher

Montage

1. Die Tastsensoren mit den Kreuzschlitzschrauben auf den Halteplatten montieren.
2. Die Halteplatten werden in der äußersten Lochreihe (B1,C1 & H1, I1) mit M6x10 Schrauben an die Basisplatte geschraubt.
3. Anschluss der Sensoren an die digitalen Eingänge:

Linker Tastsensor	an Digitaleingang 0
Rechter Tastsensor	an Digitaleingang 1

Ergebnis



Sensoren testen

Der Taster, der am Digitaleingang *i* angeschlossen ist, kann mit folgendem Befehl abgefragt werden:

car.digital(i); übergibt „true“ wenn der Taster **nicht** gedrückt ist. Ein gedrückter Taster den Wert 0. Vorsicht: die Buttons sind gerade umgekehrt programmiert.

Programm Sensor-Taster.cc:

keine Taster gedrückt	keine LED
linker Taster gedrückt	LED 0
rechter Taster gedrückt	LED 1
beide Taster gedrückt	beide LED

Programm Sensor-Taster-Motor.cc:

Fahrzeug „parkt ein“

4. Abstandsensor

Sensoren montieren und anschließen

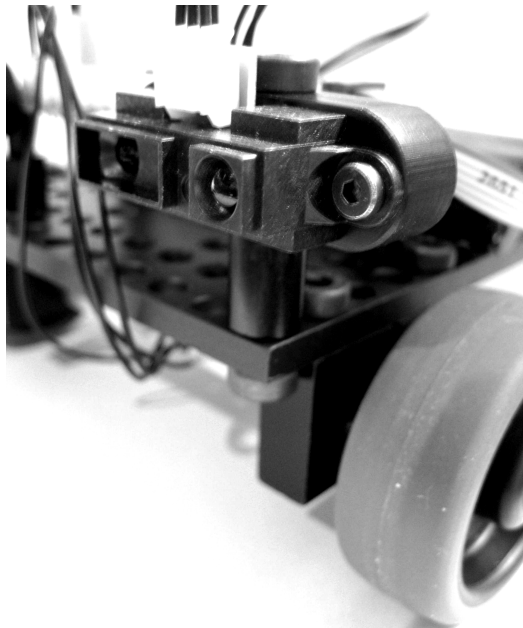
Bauteile

- 2 Abstandsensoren
- 2 Grundplatten
- 4 Inbusschrauben M3x6
- 2 Gewindehülsen 15 mm
- 2 Inbusschrauben M6x20
- 2 Inbusschrauben M6x10
- 2 Signalkabel

Montage

1. Die Sensoren werden mit Schrauben M3x6 auf ihre Grundplatten geschraubt.
2. Durch die Durchgangslöcher werden die Schrauben M6x20 mit den Gewindehülsen verschraubt.
3. Diese werden von unten mit M6x10 an der äußersten Reihe der Bohrungen in der Basisplatte befestigt.
4. Anschluss der Sensoren mit Signalkabeln:
linker Sensor an Analogeingang 0
rechten Sensor an Analogeingang 1

Ergebnis



Sensoren testen

Programm **Sensor-Abstand.cc**

Wenn der Abstand 0 geringer ist als Abstand 1, dann leuchtet LED0, sonst LED1

Nutzung

Der Rückgabewert der Sensoren liegt zwischen 0 und 255, wobei zu großen Entfernungen kleine Rückgabewerte gehören.

Werden die Abstandsensoren etwas seitlich ausgerichtet, so kann mit dem Programm **vehicle.cc** ein fertiges Ausweichprogramm getestet werden.

5. IR-Reflex-Sensor

Sensoren montieren und anschließen

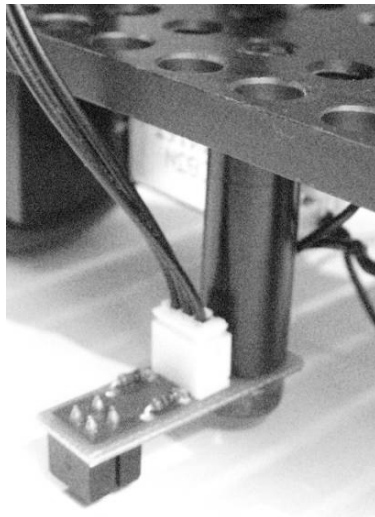
Bauteile

- 2 IR-Reflex-Sensoren
- 2 Verlängerung 30mm
- 2 Inbus-Schraube M6x10
- 2 Signalkabel

Montage

1. Die Verlängerung wird in der zweiten Lochreihe von unten in die Basisplatte geschraubt.
2. Mit der M6x10 Schraube wird der IR-Sensor an die Verlängerung geschraubt.
3. Das Signalkabel wird an die Analogeingänge 2 bzw. 3 angeschlossen.

Ergebnis



Sensoren testen

Test des an Analogeingang 2 angeschlossenen Sensors:

Programm **Sensor_IR_reflex.cc**

Kalibrieren:

Heller Hintergrund

Button 1: speichern

Dunkler Hintergrund

Button 1: speichern

Messen:

Nach Drücken von Button 1: Messung durch die Sensoren bei langsamer Vorwärtsfahrt.

Anzeige:

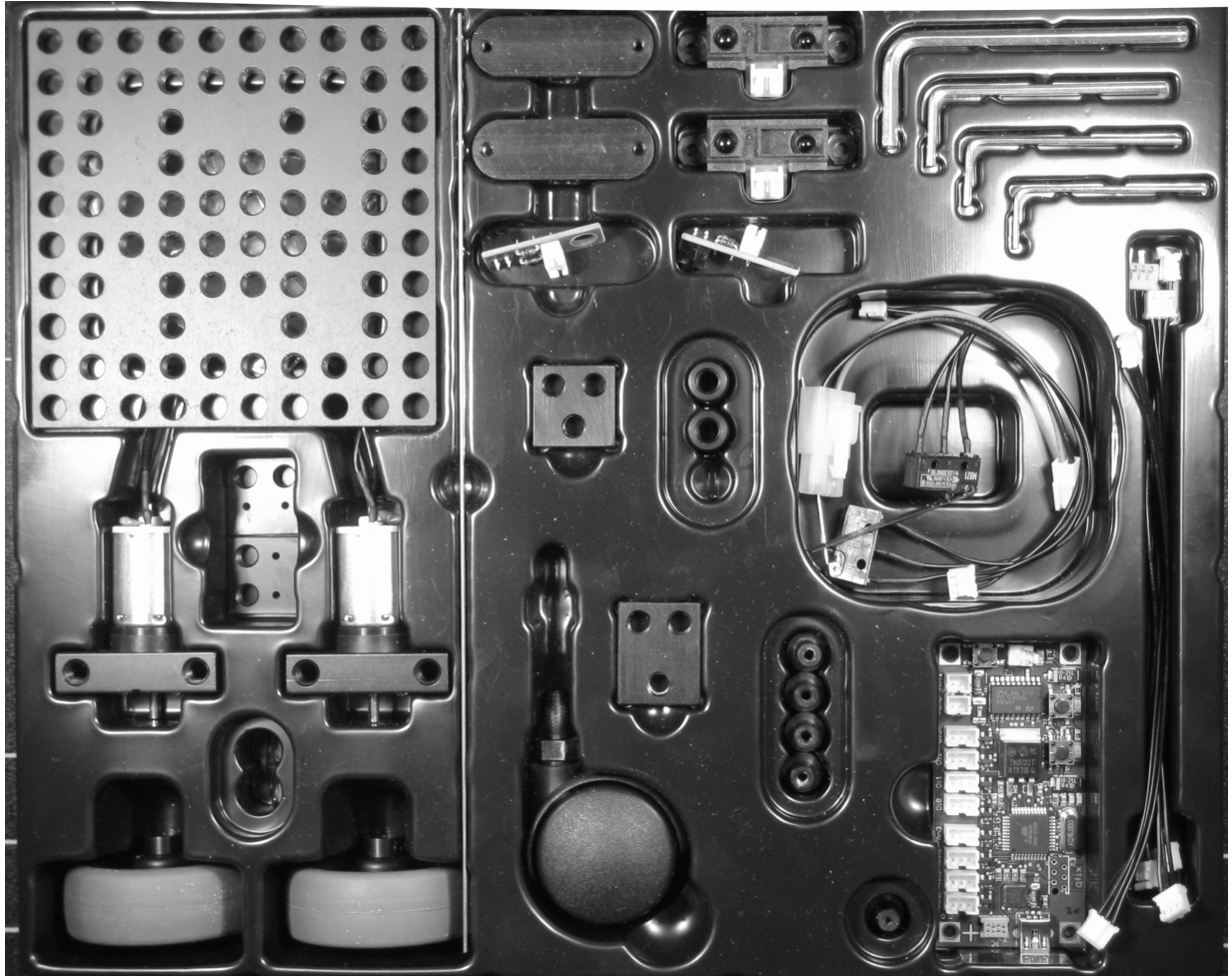
„Hell“ => LED 0

„Dunkel“ => LED 1


Nutzung

Der Rückgabewert der Sensoren liegt zwischen 0 und 255, wobei zu hellen Flächen (hoher IR- Reflexionsgrad) kleine Rückgabewerte gehören.

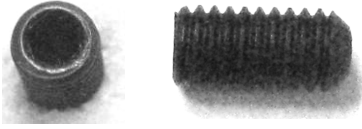

6. Bestückungsplan



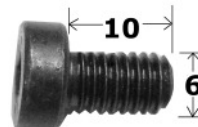
Unter der Basisplatte befinden sich zwei Tüten mit Schrauben. Der Inhalt kann durch wiegen kontrolliert werden und ist auf der Folgeseite beschrieben.

<p>Schraubentüte M2 / M3</p> <p>Wägekontrolle ergibt: 8,4 g</p> <p>Madenschrauben 3 M3x5</p> <p>Kreuzschlitzschrauben 1 M2x6 4 M2x10</p> <p>Inbusschrauben 10 M3x6</p>	<p>Schraubentüte M6</p> <p>Wägekontrolle ergibt: 90,8g</p> <p>Inbusschrauben 2 M6x8 15 M6x10 3 M6x20</p> <p>Beilagscheiben 2 D6</p> <p>2 Verlängerung 30mm</p> 
--	--

Bezeichnungen der Schrauben

Madenschrauben: Gewindestift ohne Kopf	Inbusschrauben: Innensechskantkopf
	

Bemassung der Schrauben:
 Erste Zahl: Durchmesser (in mm)
 Zweite Zahl: Gewindelänge (in mm)



M6x10

Schon verbaut sind:
 Kreuzschlitzschrauben
4 M2x6
 Senkkopfschrauben
2 M4x10

Koffer für den Klassensatz

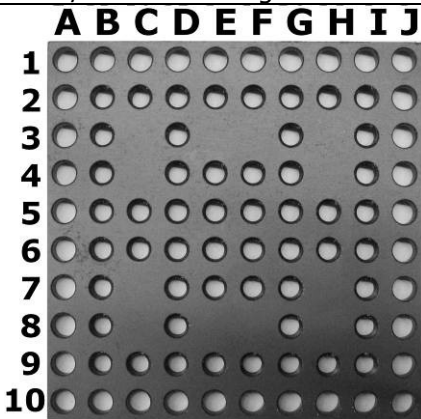
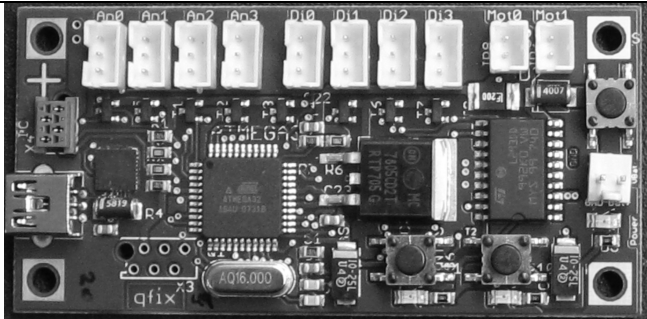
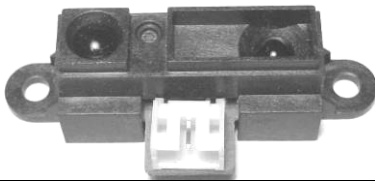

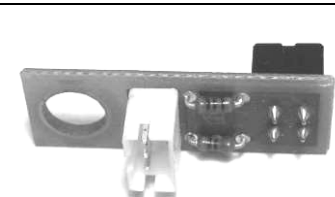

- 11** Kleinteilschalen
- 11** USB-Kabel







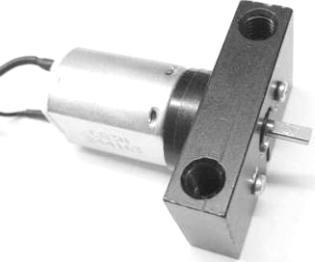
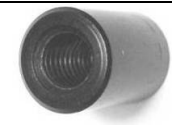


- 2** Schraubendreher Kreuzschlitz
- 2** Gabelschlüssel 10mm

- 2** Steckdosenleisten (6-fach)
- 11** Netzgeräte
- 11** Doppelpacks Akku

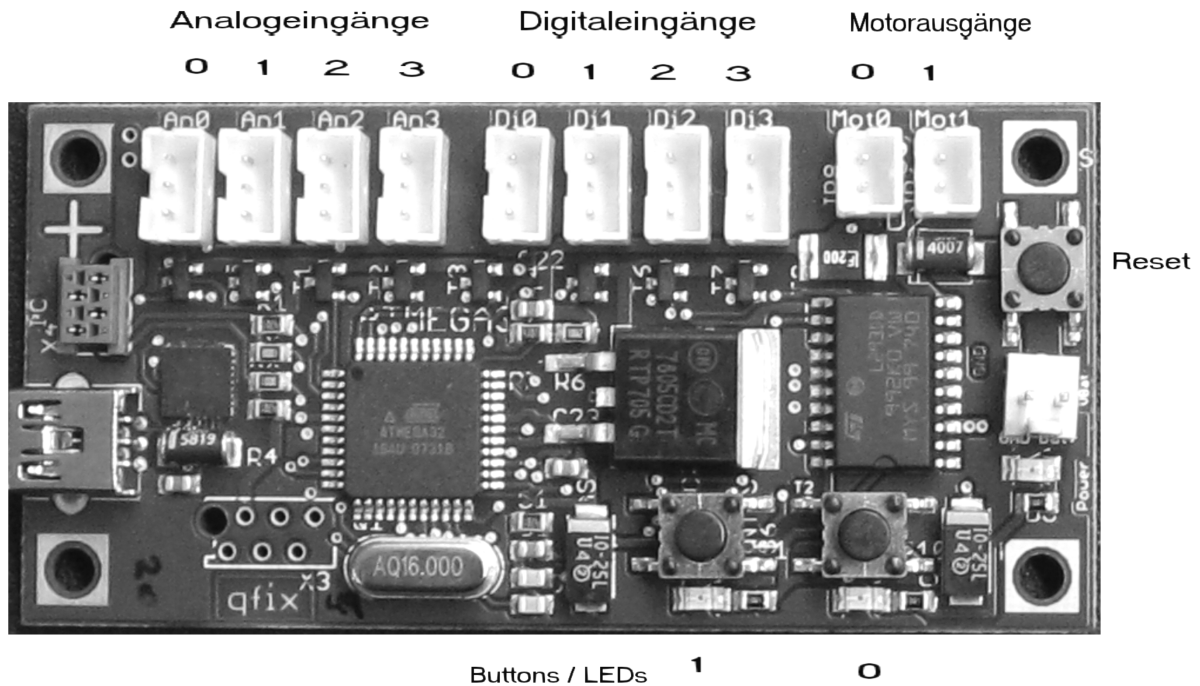
- 2** CD-ROM

7. Übersicht über die Bauteile

Anzahl	Bezeichnung	Foto / Beschreibung	Position
1	Basisplatte		Formteil links
1	Controllerboard		Formteil rechts
2	Abstandsensoren		Formteil rechts
2	Grundplatten für Abstandsensoren		Formteil rechts
2	IR-Reflex-Sensoren		Formteil Rechts (beim Abstandsensoren)
2	Tastensensoren		CD-Fach

1	Halteplatte 20x25 (3 Gewinde M6)		Formteil Rechts
1	Halteplatte 20x20 (3 Gewinde M6)		Formteil rechts
2	Halteplatte 20x20 (2 Gewinde M6)		tiefes Formteil links
1	12V Versorgung		CD-Fach
4	Signalkabel (3adrig)		Formteil rechts
2	Räder auf Achse montiert		Formteil links
2	Motore in Halte- platte montiert		Formteil links
1	Haltegummi für Geräte		Um die BOX her- um
2	Gewindehülse 15mm		Formteil rechts (eines bleibt frei)
4	Platinenhalter 20mm		Formteil rechts
1	Fahnenhalter		Formteil rechts unten

8. Datenblatt Miniboard



I/O Ports

- 4 Analogeingänge z.B. für Abstandssensoren oder Lichtsensoren
- 4 Digitaleingänge z.B. für Tastsensoren

diese sind auch nutzbar als

- 8 Powerausgänge mit 5 V (geregelt) bis 100 mA (TTL-Pegel)
- 2 Motorausgänge für Gleichstrommotoren bis 600 mA
- 2 Buttons (on board)
- 2 LED (on board)
- 1 Reset-Taste

Sensor-Eingänge / TTL-Ausgänge



Ausgang +5V (schaltbar)
 GND (Masse)
 Sensoreingang



Button



LED

LED: eine dunkle LED hat den Wert 0, eine leuchtende LED hat den Wert 1.

Button: ein **nicht** gedrückter Button hat den Wert **0**, ein gedrückter Button hat den Wert 1. Vorsicht: die Tastsensoren sind gerade umgekehrt programmiert.

Reset-Taste

Programmstart und Überspielen eines Programms sind nur nach Drücken der Reset-Taste möglich.

Dieser Zustand wird durch das schnelle Blinken von LED 0 angezeigt.

Hinweis: Falls man das Drücken des Resetknopfs vergessen hat, reicht das ein- und ausstecken des USB-Kabels. Danach nochmals versuchen das Programm zu überspielen.

9. Grundelemente von C++

Am unteren rechten Eck des Editorfensters kann die Hervorhebung von Befehlen die Zeilennummerierung, Leer- und Steuerzeichen sichtbar gemacht werden.

Im linken Fenster sind Befehlsstrukturen abgelegt. Unter C++ sind z.B. die Strukturen für Schleifen nutzbar.

Zähler erhöhen ++i	Erhöht den Wert von i um 1
Zähler senken --i	Senkt den Wert von i um 1
Wiederholungsschleifen	
while (Bedingung) { Anweisung; }	Die Befehle innerhalb der geschweiften Klammer werden so lange ausgeführt wie Bedingung = true erfüllt ist. Die Abfrage wird geprüft bevor der Anweisung ausgeführt wird. Hinweis: Bei der while Schleife kann in der runden Klammer nicht gerechnet werden.
do { Anweisung; } while (Bedingung);	Die Befehle innerhalb der geschweiften Klammer werden so lange ausgeführt wie Bedingung = true erfüllt ist. Die Anweisung wird ausgeführt, bevor die Abfrage geprüft wird. Somit wird mindestens einmal die Anweisung innerhalb der Schleife ausgeführt.
Zählschleifen	
for (Startbedingung; Stoppbedingung; Veränderung) { Anweisung; }	Der Zähler wird von der Startbedingung so lange erhöht bis die Stoppbedingung erreicht ist. Bei jedem Zählschritt wird der Anweisung ausgeführt.
Verzweigung	
if (Bedingung) { Anweisung1; } else { Anweisung2; }	Eine If-Schleife wird verwendet, um zu entscheiden, ob Befehle ausgeführt werden. Wenn Bedingung = true ist, dann wird Anweisung1 ausgeführt, andernfalls Anweisung2.
switch (Variable) { case (Wert1): Anweisung1; break; case (Wert2): Anweisung2; break; default: Anweisungx; break; }	Verzweigung in Abhängigkeit vom Wert einer Variable: Wenn sie den Wert1 annimmt wird Anweisung1 ausgeführt (bis break) Wenn sie den Wert2 annimmt wird Anweisung2 ausgeführt (bis break) Andernfalls Anweisungx Hinweis: Bei beim case(Wert) kann der Wert nicht ausgerechnet werden. Es muss dort eine feste Zahl stehen.

10. Programmierung Methoden der Klasse MiniBoard

Die Tabelle beschreibt die Klasse MiniBoard sowie deren Methoden.

Um diese Methoden aufrufen zu können, muss ein Objekt (eine Instanz) der Klasse existieren. Durch den Befehl **Miniboard car;** wird ein Objekt „car“ der Klasse MiniBoard erzeugt, dieses erhält damit die Methoden der Klasse MiniBoard. (car könnte durch eine beliebige andere Bezeichnung wie z.B. MeRK ersetzt werden). Durch das Anlegen des Objektes wird es automatisch aufgerufen und initialisiert. Die Motorgeschwindigkeiten werden auf 0 gesetzt, alle LEDs sind aus, alle Power-Ausgänge sind angeschaltet.

Einige Methoden geben einen Wert zurück. In der Tabelle ist der Typ des Rückgabewertes angegeben.

Methoden der Klasse MiniBoard (Auf Groß- und Kleinschreibung achten!!)	Typ des Rückgabewertes	Erklärung
ledOn(i)	void	Schaltet die LED mit Index i an
ledOff(i)	void	Schaltet die LED mit Index i aus
ledsOff()	void	Schaltet alle LEDs aus.
led(i, Bedingung)	void	Schaltet die LED mit dem Index i an, falls Bedingung=true, bzw. aus, falls Bedingung=false. i muss 0 oder 1 betragen.
button(i)	bool	Liefert true zurück, falls der Button mit dem Index i gedrückt ist, ansonsten false. i muss 0 oder 1 betragen.
waitForButton(i)	Void	Wartet, bis der Button mit dem Index i gedrückt und wieder losgelassen wurde
motor(i, speed)	Void	Setzt den Motor mit dem Index i auf den Wert speed, speed liegt im Bereich -255 bis 255.
motors(speed0,speed1)	Void	Setzt beide Motoren auf die entsprechenden Werte speed0 und speed1
motorsOff()	Void	Schaltet beide Motoren aus.
analog(i)	int	Liefert den Wert des Analog-Eingangs mit dem Index i zurück. Der Rückgabewert liegt im Bereich 0 bis 255.
digital(i)	bool	Liefert true zurück, falls der Digital-Eingang mit dem Index i high ist, ansonsten false
powerOn(i)	void	Schaltet den Power-Ausgang mit Index i an
powerOff(i)	void	Schaltet den Power-Ausgang mit Index i aus
power(i, Bedingung)	void	Schaltet den Power-Ausgang mit dem Index i an, falls Bedingung=true, bzw. aus, falls Bedingung=false.
ledMeter(i)	void	Zeigt den übergebenen Wert i im Bereich 0 - 255 mittels der LEDs an.
sleep(s)	void	Wartet s Sekunden lang.
msleep(ms)	void	Wartet ms Millisekunden lang.

Folgende Datentypen werden hierbei verwendet:

Typ	Kürzel / Variable	Erklärung
void		Kein Wert (wird z.B. bei Funktionen oder Methoden ohne Rückgabewert benutzt)
bool	Bedingung	Boolescher Wert (true oder false)
int		ganzzahliger Wert (-32767 - 32768)
	i	gibt die Nummer des Ein-, oder Ausganges an. Sie muss bei den Eingängen im Bereich 0 bis 3 und bei den Ausgängen im Bereich 0 bis 7 liegen.
	Speed	gibt die Geschwindigkeit an, und darf zwischen -255 und 255 liegen.
	s, ms	Zeit in s bzw. ms