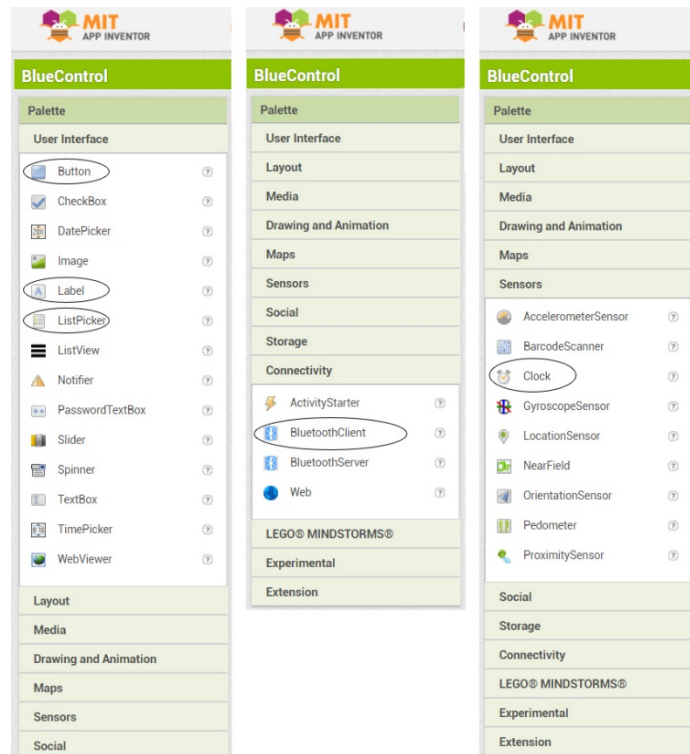


Roboter mit dem Handy steuern

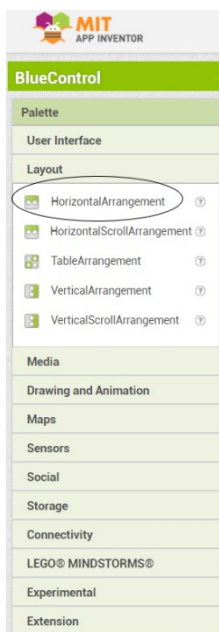
Appdesign

Um einen Roboter mit dem Handy zu steuern, wird eine App benötigt, die auf einfache Weise mit dem „App-Inventor 2“ realisiert werden kann. Dazu muss das Programm im Internet gestartet und aus der „Palette“ die benötigten Komponenten auf den „Viewer“ bzw. den „Screen“ (virtueller Bildschirm) gezogen werden. Der Reihe nach sind das:

1. Ein „ListPicker“, der bluetoothfähige Geräte erkennen und in einer Liste anzeigen kann, aus der das gewünschte ausgewählt werden muss.
2. Ein „Label“, das anzeigen kann, ob eine Verbindung zum gewählten Bluetooth-Modul hergestellt worden ist oder nicht.
3. „Button(s)“, die Steuerbefehle an den Roboter übermitteln.
4. Ein „BluetoothClient“, der die Verbindung zwischen den Geräten überwacht.
5. Üblich ist auch, eine Uhr mit einzubinden.



Als Steuerbefehle sind ▲ vorwärts, ▼ rückwärts, ◀ links, ▶ rechts und STOPP vorgesehen, die wie folgt zusammengestellt werden sollen:



Um die Knöpfe „links“, „rechts“ und „STOPP“ nebeneinanderzulegen, muss in der „Palette“ unter „Layout“ das Feld „Horizontal(Scroll)Arrangement“ auf „Screen1“ platziert werden, in das die „Buttons“ der Reihe nach eingefügt werden. Die „Buttons“ werden über „Properties“ und „Image“ mit passende Grafiken ausgestattet.

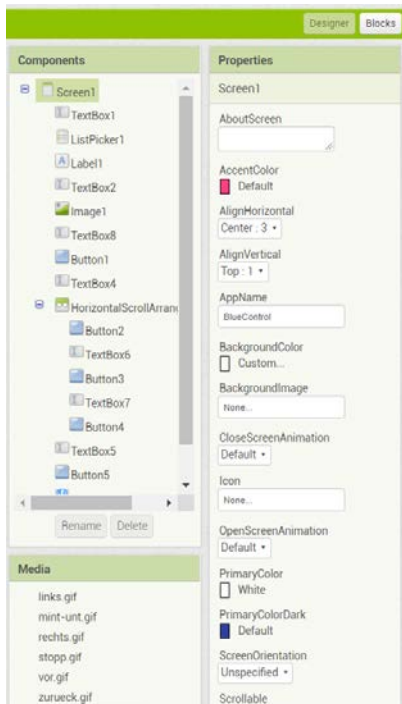
Mit Hilfe leerer „TextBox(es)“ an geeigneten Stellen fungieren, wenn unter „Properties“ die Schriftgröße („Height“ und „Width“) variiert wird. Weitere Grafiken können bei Bedarf über „Palette“, „User-Interfaces“ und „Image“ eingefügt werden.

Rechts das „Design“ im „Viewer“ nach Fertigstellung.



Programmierung der Komponenten

Für die Programmierung im App-Inventor rechts oben auf „Blocks“ klicken und dann in der linken



Spalte unter „Components“ und „Screen1“ den Eintrag „ListPicker1“ anwählen. In dem sich öffnenden Auswahlménú den braunen Block „when ListPicker1.BeforePicking – do“ anklicken. Durch weitere Klicks den dunkelgrünen Block „set ListPicker1.Elements to“ auswählen und in die Lücke des ersten



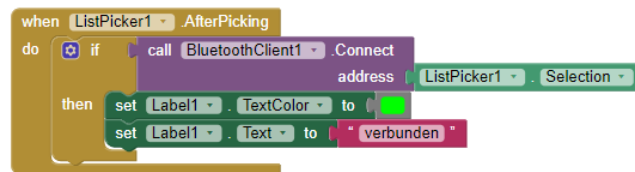
Blocks hinter „do“ einfügen. Der noch fehlende hellgrüne Block „BluetoothClient1.AddressesAndNames“ findet sich unter „BluetoothClient1“ und wird entsprechend der Abbildung oben angefügt.

Dieser Teil der Programmierung sorgt dafür, dass die verfügbaren Bluetoothgeräte angezeigt werden.

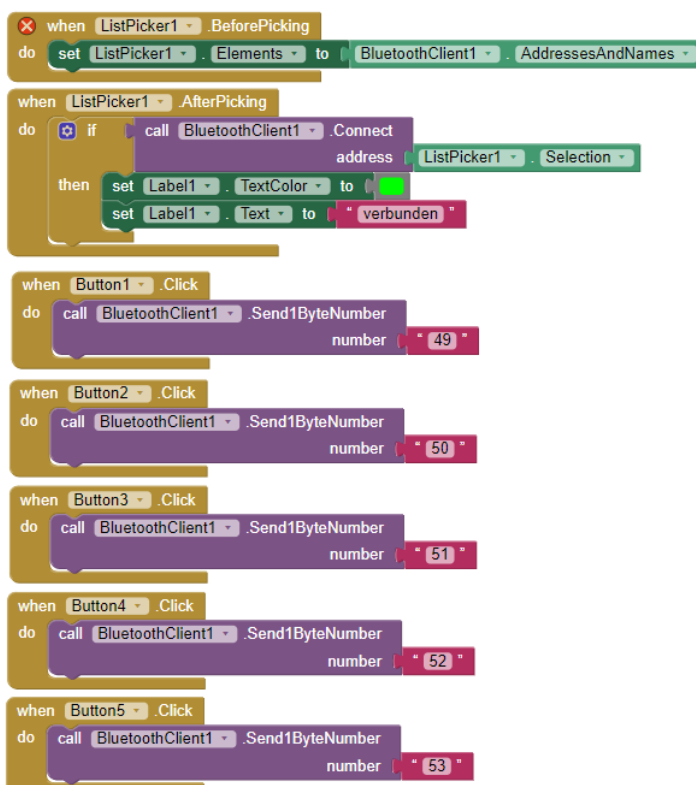
Ein weiterer Programmteil muss dann für dessen Einbindung sorgen.

Dazu durch einen Klick unter „Components“ auf „ListPicker1“ das Auswahlménú öffnen und den braunen Block „when ListPicker1.AfterPicking – do“ auswählen. Danach unter

„Components“, „Built-in“, „Control“ den Block „if – then“ auswählen und in die Lücke des vorherigen Blocks hinter „do“ einfügen. Hinter „if“ wird dann der lila Block „call BluetoothClient1.Connect – address“ eingefügt usw.



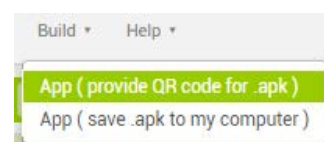
Die weiteren Schritte zur Fertigstellung der Programmierung sollten nach diesen Erklärungen mithilfe der folgenden Abbildung selbst gefunden werden können im Vergleich mit der folgenden



Abbildung, die den kompletten Code umfasst.

Die Zahlen 50 bis 53 entsprechen den Tasten 1 bis 4 auf dem Computer. Das ist bedeutsam, um die Zahlenangaben im Arduino-Quellcode zu verstehen.

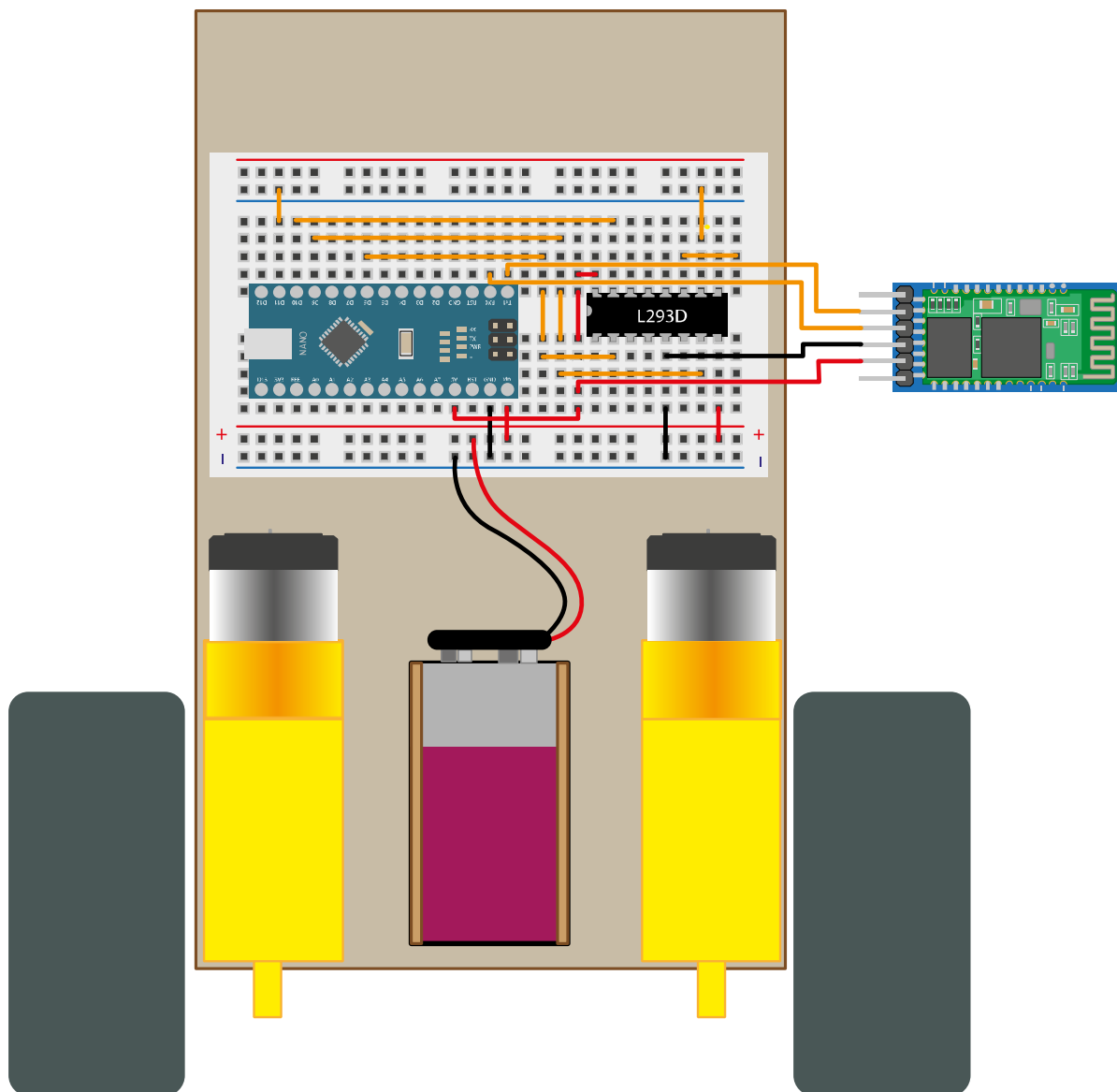
Ist die Programmierung fertiggestellt, wird sie zur Ausführung auf dem Smartphone gespeichert. Dazu im App-Inventor ganz oben auf „Build“ klicken und in dem sich öffnenden Menü „App (save .apk to my computer)“ auswählen.



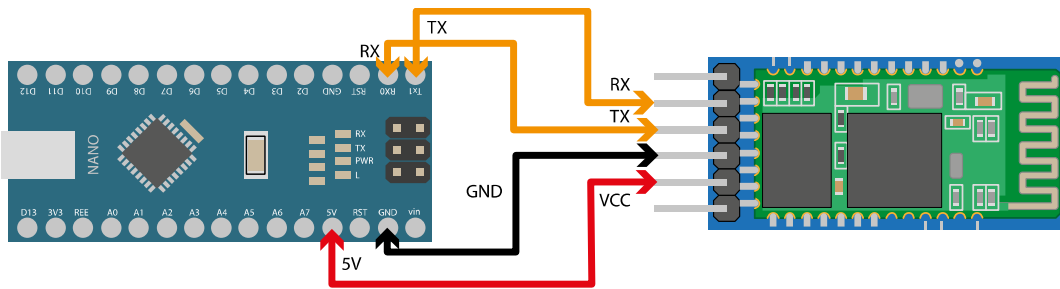
Die Installationsdatei wird damit in den Downloadordner des eigenen Rechners kopiert. Von dort sendet man sie beispielsweise als Anhang an eine Email zum Handy. Ist auf dem Handy die Installation von Fremdsoftware erlaubt, erfolgt die Installation der App nach der Freigabe ohne weiteres Zutun.

Fahrzeugdetails

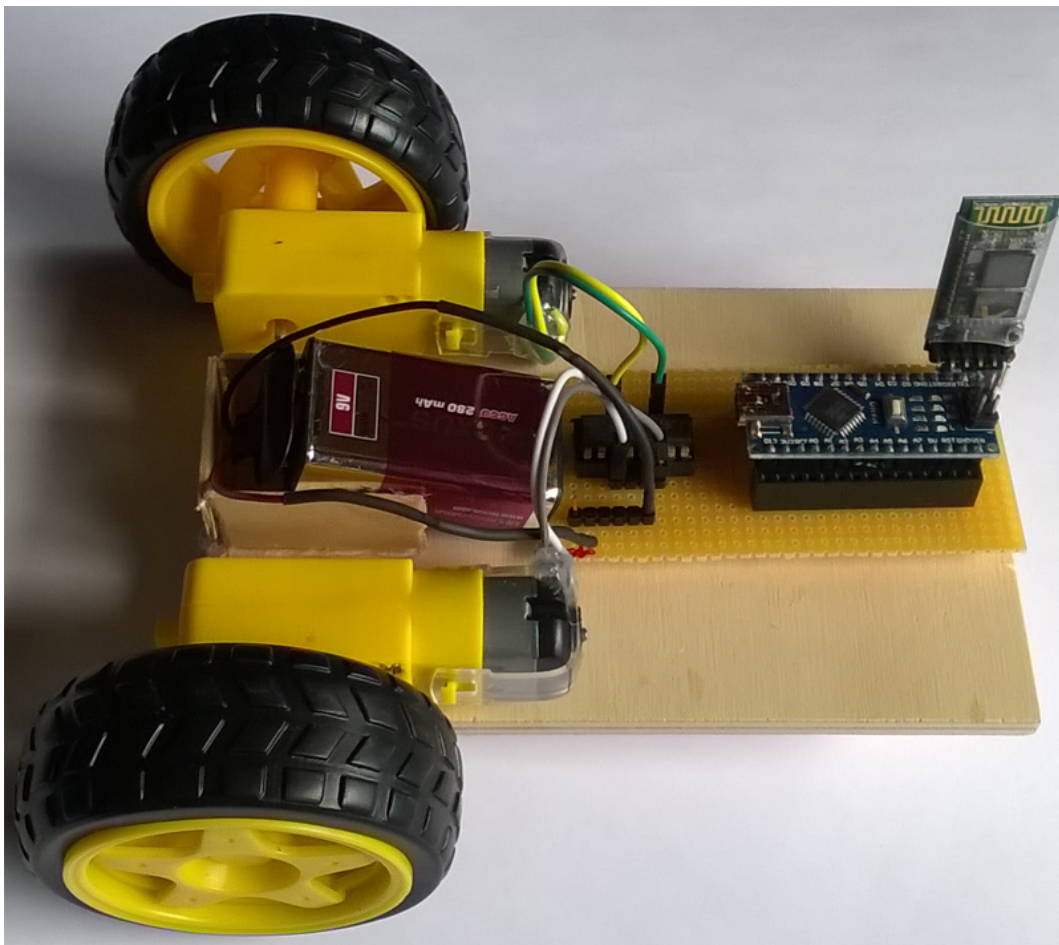
Als Fahrzeug eignet sich der am Ende des Kapitels „[Durstige Motoren](#)“ beschriebene Aufbau, der um ein Bluetooth-Modul (HC-05) erweitert werden muss. Zusätzlich zur den Abbildungen unten können dort weitere Details gefunden werden.



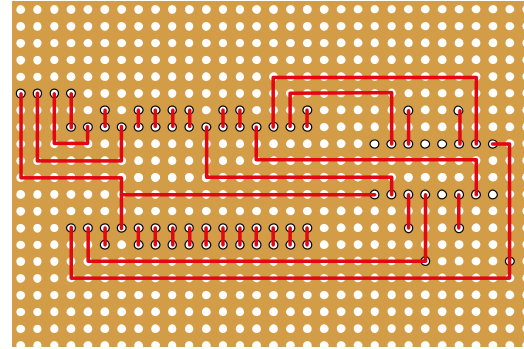
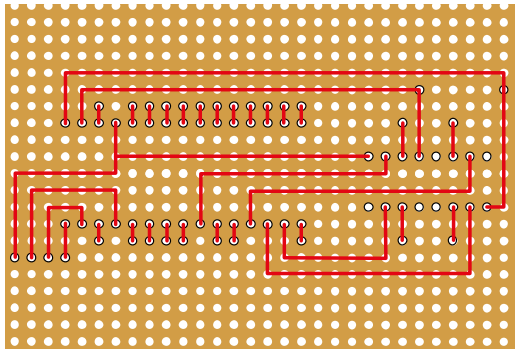
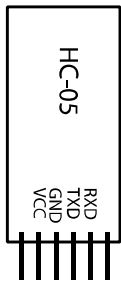
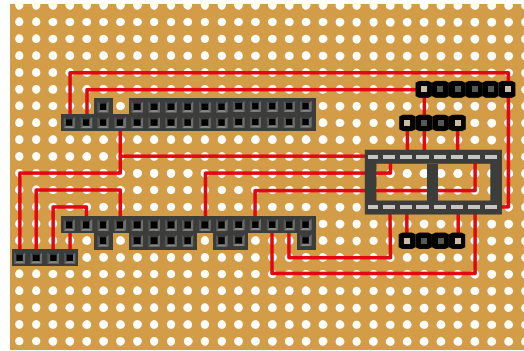
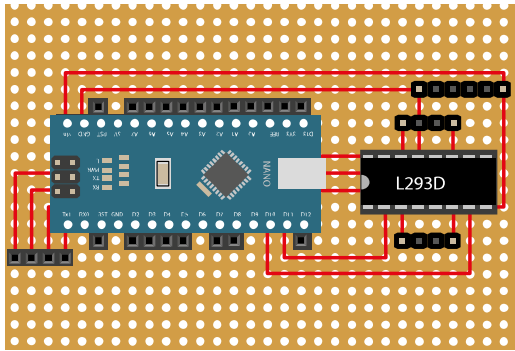
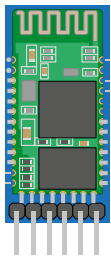
Die Pins zum Ansteuern der Motoren (6, 9, 10, 11) erlauben Drehzahlenregelung, sodass die Geschwindigkeit des Fahrzeugs gedrosselt werden kann und die Fernsteuerung damit einfacher wird. Zum Anschluss des Bluetooth-Moduls orientiert man sich an den Beschriftungen: VCC wird mit 5V, GND mit GND. RX und TX werden überkreuz verbunden, also TX am Nano mit RX am HC-05 und RX am Nano mit TX am HC-05. Vergleiche dazu auch die folgende Abbildung:



Wer Erfahrungen im Lötten hat, dem sei folgender Aufbau empfohlen, der – wie bei der Nutzung eines Breadboards – die freien Pins des Arduino Nanos zugänglich hält:



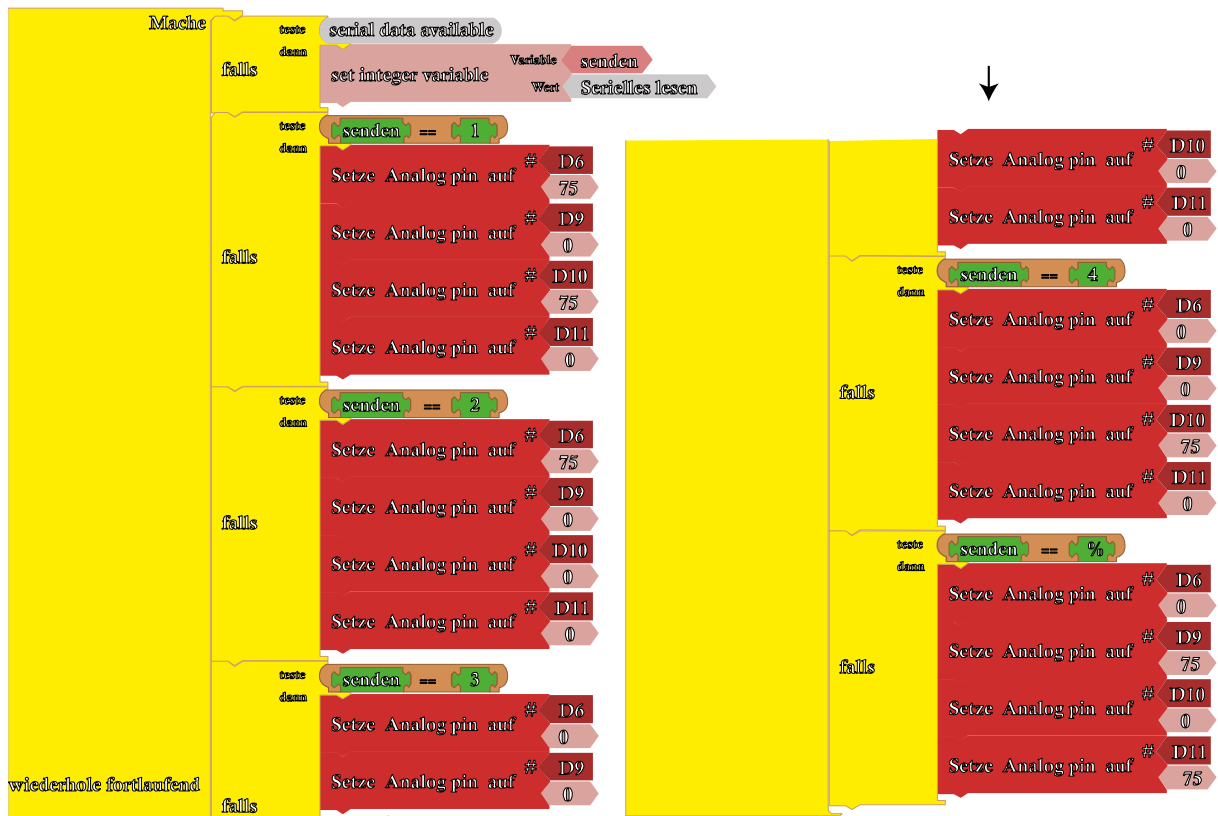
Die Abbildung unten zeigt die mit Nano und L293D bestückte Platine, dann dieselbe ohne Bauteile und schließlich nur die Kabelführung mit den Lötunkten. Da die Kabel auf der Unterseite der Platine verlegt sind, ergibt sich von der Lötseite eine andere Ansicht. Hier ist sie um die Längsachse gedreht (horizontal gespiegelt).



Lötseite

Programmierung

Der Arduino muss so programmiert werden, dass er auf die von der Smartphone-App über Bluetooth ausgegebenen seriellen Steuersignale empfangen und darauf reagieren kann. Hier ein tauglicher Code, der mit Ardublock zusammengestellt werden kann:



Der von dem Arduino-Plugin generierte Code lautet:

```

int _ABVAR_1_senden = 0 ;
void setup() {
  Serial.begin(9600);
}
void loop() {
  if (Serial.available())
  {
    _ABVAR_1_senden = Serial.read();
  }
  if (( ( _ABVAR_1_senden ) == ('1' ) ) )
  {
    analogWrite(6, 75);
    analogWrite(9, 0);
    analogWrite(10, 75);
    analogWrite(11, 0);
  }
  if (( ( _ABVAR_1_senden ) == ('2' ) ) )
  {
    analogWrite(6, 75);
    analogWrite(9, 0);
    analogWrite(10, 0);
    analogWrite(11, 0);
  }
  if (( ( _ABVAR_1_senden ) == ('3' ) ) )
  {
    analogWrite(6, 0);
    analogWrite(9, 0);
    analogWrite(10, 0);
    analogWrite(11, 0);
  }
  if (( ( _ABVAR_1_senden ) == ('4' ) ) ) {
    analogWrite(6, 0);
    analogWrite(9, 0);
    analogWrite(10, 75);
    analogWrite(11, 0);
  }
  if (( ( _ABVAR_1_senden ) == ('5' ) ) )
  {
    analogWrite(6, 0);
    analogWrite(9, 75);
    analogWrite(10, 0);
    analogWrite(11, 75);
  }
}
}

```

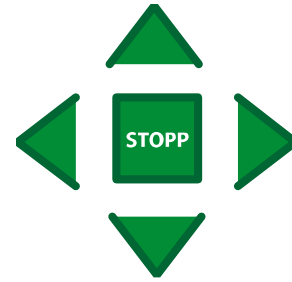
Inbetriebnahme

Wenn alles am Fahrzeug wie oben beschrieben programmiert und zusammengebaut ist und von einem 9V Block mit Strom versorgt wird, sollte die rote LED am Bluetooth-Modul in schneller Folge blinken. In diesem Modus können Modul und Handy miteinander verbunden werden.

Bei Android-Handys muss dazu unter „Einstellungen“ und „Bluetooth“ nach „Pairing-Geräten“ gesucht werden. Das Bluetooth-Modul hat die Bezeichnung „HC-05“. Wird es ausgewählt, muss das Passwort eingegeben werden – in der Regel „1234“. Danach wird das Modul in der Liste der bekannten „Pairing-Geräte“ aufgenommen und kann so von unserer „App“ gefunden werden. Also App starten, auf „Gerät zum Verbinden auswählen“ klicken. Es zeigt sich die Liste der dem Handy bekannten Bluetooth-Geräte, aus der das gewünschte durch Anklicken ausgewählt werden kann.

War die Auswahl erfolgreich, blinkt die LED in längeren Intervallen kurz zweimal und die Motoren lassen sich über die Bedienfelder steuern:

- Das obere Dreieck lässt die Motoren in die Richtung drehen, die das Fahrzeug vorwärtsbewegt
- Das untere Dreieck lässt die Motoren in die entgegengesetzte Richtung drehen, die das Fahrzeug rückwärtsfahren lässt.
- Das linke Dreieck startet den rechten Motor, sodass sich das Fahrzeug nach links bewegt
- Das rechte Dreieck startet den linken Motor, sodass sich das Fahrzeug nach rechts bewegt
- Das mittlere Feld stoppt die Motoren



Betrieb mehrerer Fahrzeuge

Wenn mehrere Fahrzeuge (beispielsweise im Klassenunterricht) in Betrieb genommen werden sollen, ist es hilfreich, wenn die Bluetooth-Module individuelle Namen haben und nicht uniform „HC-05“ heißen.

Es ist deshalb empfehlenswert, die Module umzuprogrammieren und die Namen mit Aufklebern an der Hardware erkennbar zu machen.

Das Umbenennen kann mit „AT-Befehlen“ über den seriellen Monitor der Arduino-IDE erfolgen.

Dazu muss auf den Arduino ein entsprechendes Programm aufgespielt werden, beispielsweise das folgende:

```
#include <SoftwareSerial.h>
SoftwareSerial mySerial(6, 7); // 6 -> TX, 7 -> RX
int i=0;
void setup() {
  mySerial.begin(38400);
  Serial.begin(38400);
  Serial.println("AT-Befehle eingeben:");
}
void loop() {
  if (mySerial.available())
    Serial.write(mySerial.read()); //Befehle lesen
  if (Serial.available())
    mySerial.write(Serial.read()); //Befehle zum HC-05 übertragen
}
```

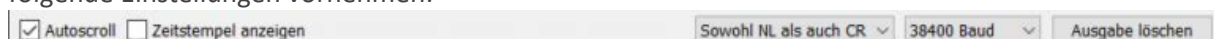
Ist die Programmierung abgeschlossen, das HC-05-Modul an den Arduino anschließen:

Arduino	HC-05	
Pin 6	TX	Die Beschriftung an den Modulen scheint nicht immer verlässlich zu sein. Bei Problemen TX gegen RX tauschen und den Monitor neu starten.
Pin 7	RX	
5V	5V	
GND	GND	

Ist dies geschehen und der Arduino unter Strom gesetzt, blinkt die LED am HC-05-Modul in schneller Folge und zeigt damit seine Pairing-Bereitschaft an.

Zum Umbenennen muss der HC-05 in den Programmiermodus versetzt werden. Dazu die Stromzufuhr unterbrechen und beim erneuten Anschalten den RESET-Knopf am HC-05-Modul kurz gedrückt halten. Die LED blinkt dann langsamer (im 2-Sekunden-Takt).

In diesem Zustand den seriellen Monitor mit der Tastenkombination „Shift+Strg+M“ starten und folgende Einstellungen vornehmen:



Anschließend in die Adresszeile „AT“ eintippen (natürlich ohne Anführungszeichen) und auf „Senden“ klicken. Die erfolgreiche Eingabe wird im Monitor mit einem „OK“ bestätigt. Kommt es zu einer Fehlermeldung, z. B. „ERROR(0)“, die Eingabe wiederholen.

Der Befehl zum Umbenennen lautet:

„AT+NAME=Wieauchimmer“ (max. 20 Zeichen)

Weitere AT-Befehle können dem Datenblatt entnommen werden.

Achtung:

- Es hat sich gezeigt, dass eine Umbenennung für die spätere Nutzung fatal sein kann. Als unkritisch haben sich die Erweiterung des Namens um eine Zahl oder Buchstaben erwiesen, also beispielsweise „HC-05-1“, „HC-05-2“ oder „HC-05-A“.
- Sollte das Bluetoothmodul auf dem Handy (oder einem anderen Gerät der eigenen Wahl) bereits registriert gewesen sein, wird der neue Name nicht angezeigt. Um das zu ändern, muss auf dem Endgerät das Modul aus der Pairing-Liste gelöscht und anschließend mit neuem Namen registriert werden. Das benötigte Passwort kann über den seriellen Monitor mit dem Befehl „AT+PSWD“ abgefragt werden. Es lautet in der Regel „1234“.