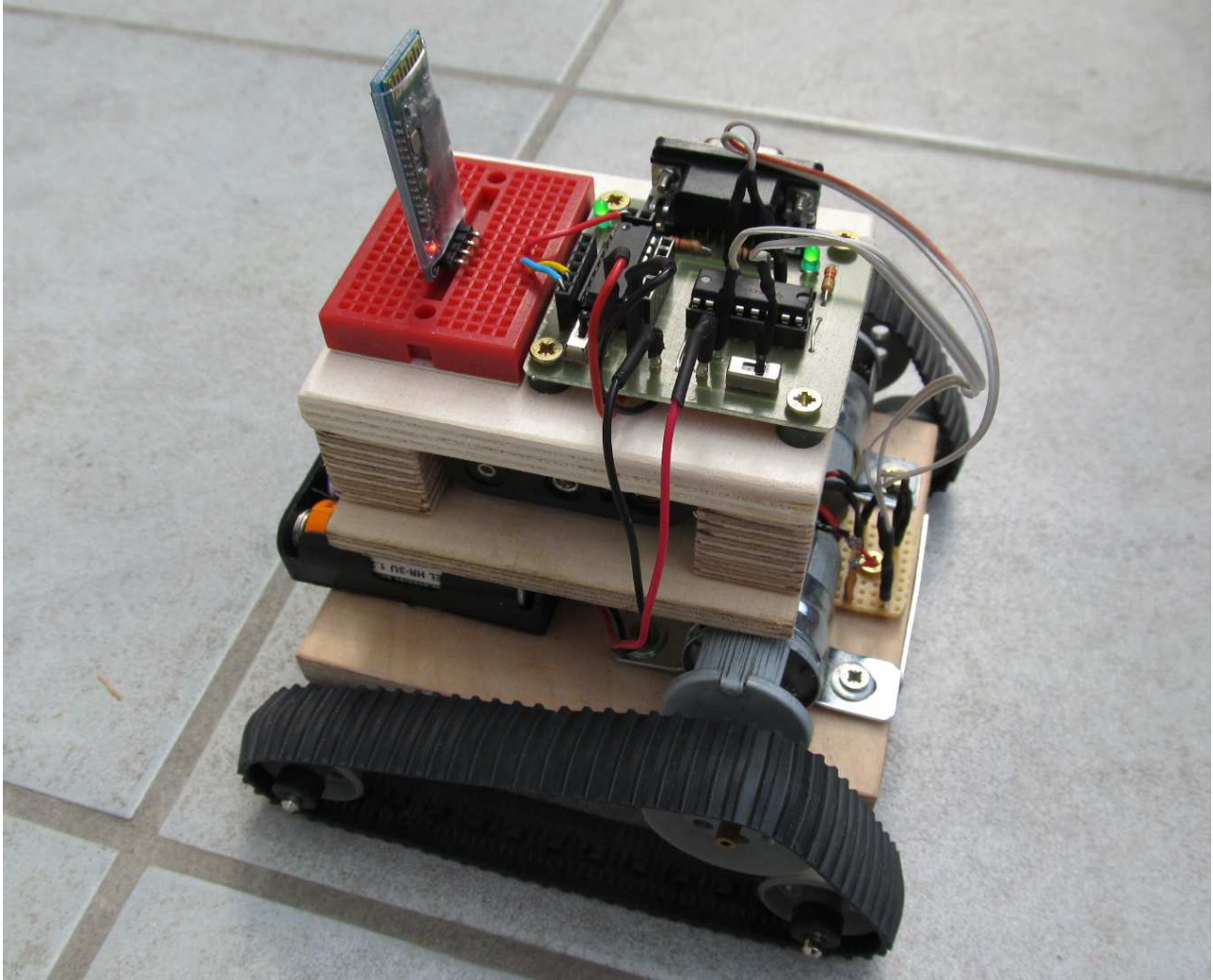


## Das GSR-Picaxe-14M2-Controllerboard (Version 1)



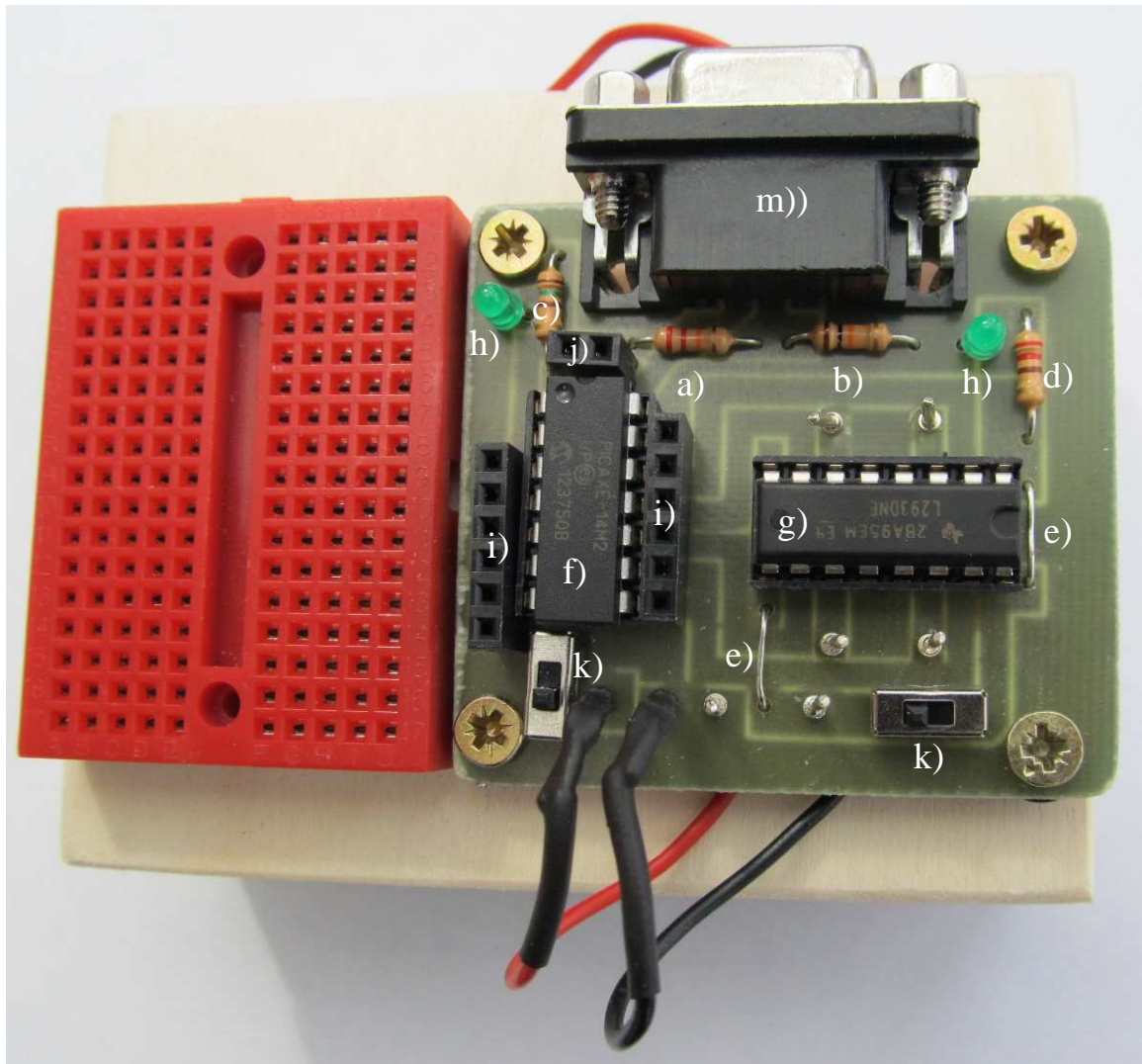
Das GSR-Picaxe-14M2-Controllerboard mit HC-06 Bluetooth-Controller auf unserem GSR-Kettenfahrzeug

## 1. Die Herstellung des GSR-Picaxe-14M2-Controllerboards Version 1

- Die vier Befestigungslöcher an den Ecken mit einem 3 mm Bohrer aufbohren.
- Die beiden Befestigungslöcher für den Sub-D-9-Pol-Stecker mit einem 3 mm Bohrer aufbohren.
- Mit Stahlwolle die Kupferseite der Platine abreiben und die Grate entfernen.

Bauteilliste:

Teil	Menge	Bezeichnung	Hinweis
a)	1	Kohleschichtwiderstand 10 k $\Omega$	R1: braun, schwarz, orange
b)	1	Kohleschichtwiderstand 22 k $\Omega$	R2: rot, rot, orange
c)	1	Kohleschichtwiderstand 150 $\Omega$	R3: braun, grün, braun
d)	1	Kohleschichtwiderstand 220 $\Omega$	R4: rot, rot, braun
e)	2	Drahtbrücken	ca. 20 mm
f)	1	IC-Sockel 14 Pin	Kerbe zeigt nach oben zum Stecker
g)	1	IC-Sockel 16 Pin	Kerbe zeigt nach rechts
h)	2	Farbige LED 3 mm	low current
i)	2	Buchsenleiste 6 Pins	
j)	1	Buchsenleiste 2 Pins	
k)	2	Schiebeschalter (3 Pin)	
l)	6	Lötstifte 1 mm	
m)	1	Sub-D-9-Pol-Stecker	



## 2. Spannungsversorgung und Ausgangsleistung

Der **Picaxe 14M2 Mikrocontroller** sollte mit einer **Spannung von 4,5 V bis 5 V** versorgt werden. Eine höhere Spannung als 5,5V kann den Chip zerstören. Jeder Input oder Output Pin darf mit maximal 20 mA Stromstärke belastet werden. Der gesamte Picaxe-14M2-Chip mit maximal 90 mA.

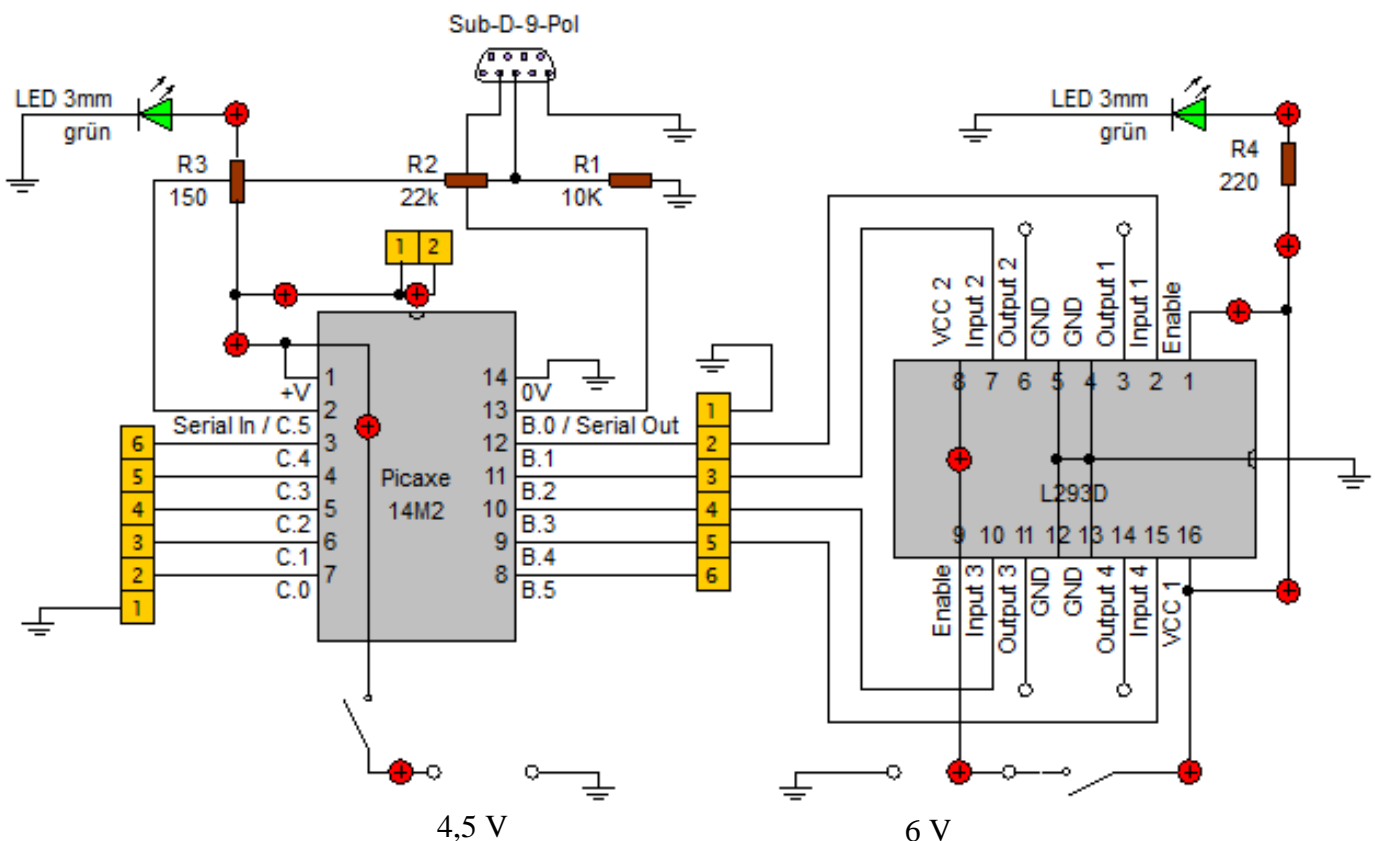
Der **Motortreiberchip L293D** sollte mit **4,5 V bis 7 V** betrieben werden, da Pin 8 und 9 miteinander verbunden sind.

**Wichtig: Es müssen zwei getrennte Spannungsquellen für den Picaxe-Chip und den Motortreiber verwendet werden.**

Drei Vorschläge zur Spannungsversorgung:

- Drei AA-Batterien in einem Batteriehalter und zwei Steckhülsen ( $3 \times 1,5V = 4,5V$ )
- Vier AA-Akkus in einem Batteriehalter und zwei Steckhülsen ( $4 \times 1,2V = 4,8V$ )
- Drei LR44 Knopfzellen ( $3 \times 1,5V$ ) in einem AA-Batteriehalter

## 3. Der Schaltplan des GSR-Picaxe-14M2-Controllerboards



## 4. Nutzung der Pins und Buchsenleisten

Die Versorgungsspannungen sollen jeweils an die beiden Lötstifte neben den Schiebeschaltern angeschlossen werden. Die Pins B.1 bis B.5 und C.0 bis C.4 können genutzt werden. Die beiden Pins B.0 und C.5 stehen nicht zur Verfügung. Die 2-Pin-Buchsenleiste kann als Pluspol genutzt werden. Die Buchse 1 der 6-Pin-Buchsenleiste ist jeweils als Masse (Minuspol) angeschlossen.



## 5. Die Software

Der Picaxe Programming Editor kann kostenlos unter [www.picaxe.com](http://www.picaxe.com) heruntergeladen werden.

## 6. Anschluss des GSR-Picaxe-14M2-Controllerboards an den Computer

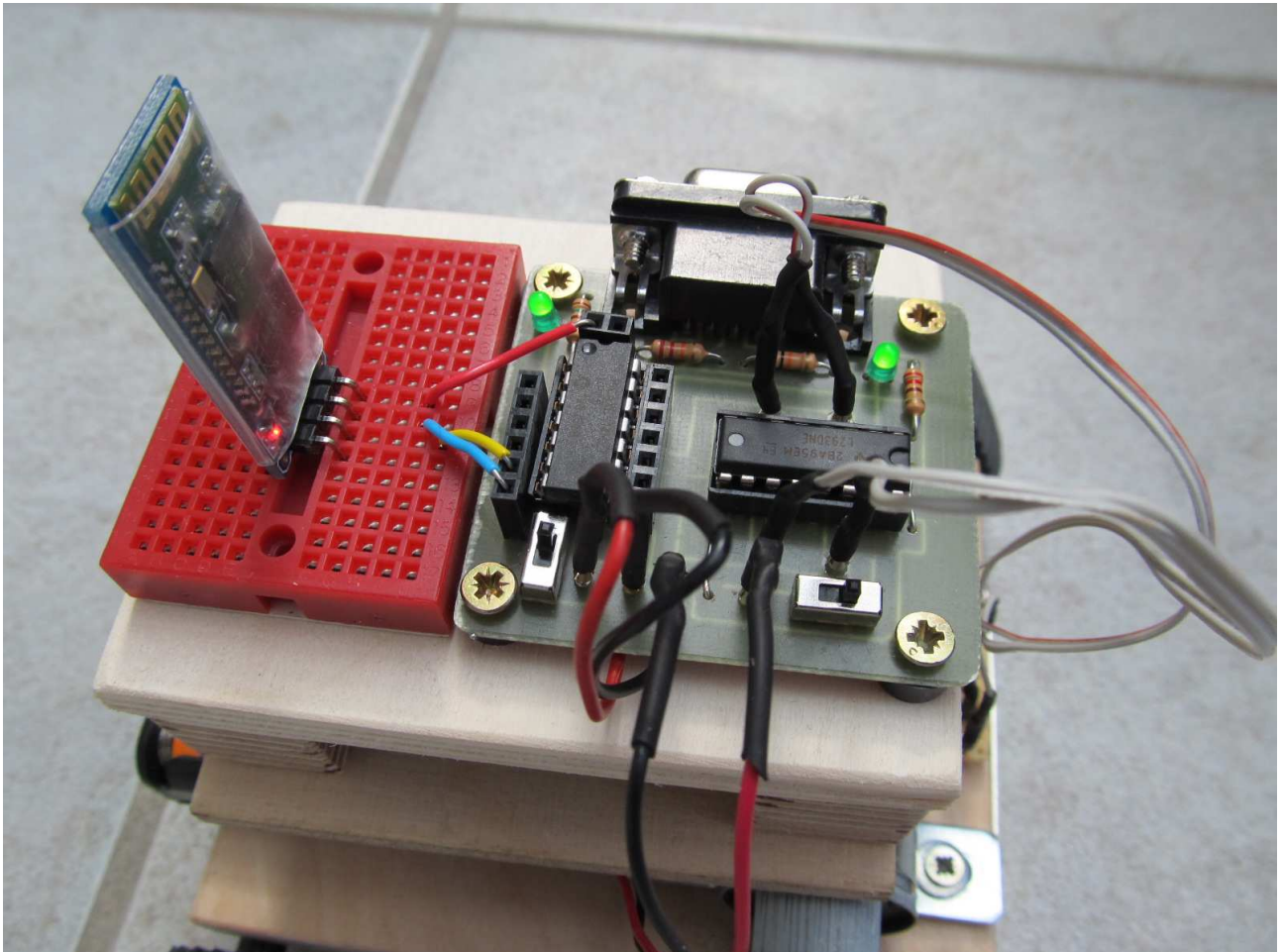


Das Controllerboard kann entweder über ein 9-poliges Sub-D-Verlängerungskabel (Kupplung/Stecker) oder ein USB-Adapterkabel auf Seriell (9-polig Sub-D) angeschlossen werden. Bei den USB-Adapterkabeln kommt es auf den eingebauten Chip an. Picaxe empfiehlt das eigene Kabel mit FTDI-Chip (AXE027 hat aber den kleinen Klinkenanschluss) und bietet noch ein USB-Adapterkabel auf Seriell mit dem günstigeren Prolific-Chip an (USB010).

Alternativ habe ich bei Pollin zwei USB-Adapterkabel auf Seriell getestet. Bis jetzt fehlerfrei. Die günstige Version mit Prolific-Chip hat die Artikelnummer 721034 und kostet 4,95€. Die Variante mit FTDI-Chip ist von Logilink und hat die Artikelnummer 722784 (9,95€).



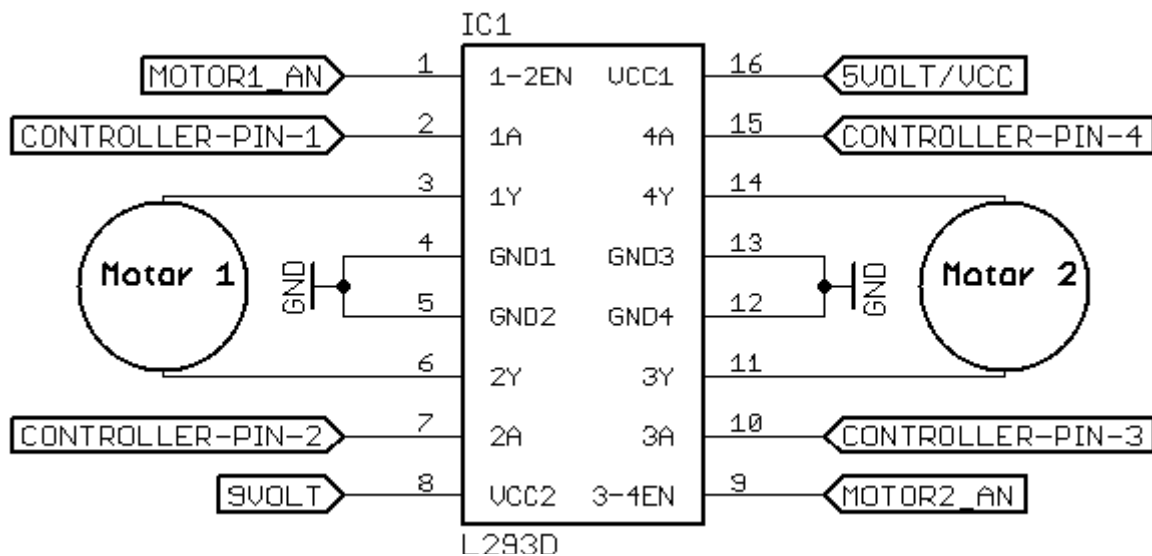
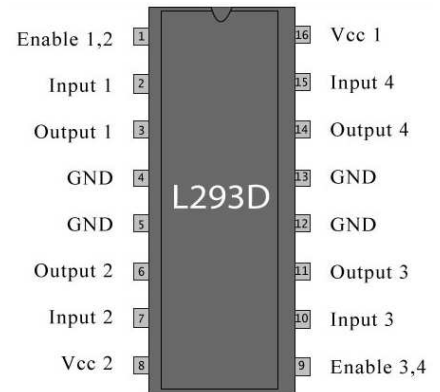
## 7. Das GSR-Picaxe-14M2-Controllerboard als Aufsatz für unser Kettenfahrzeug



Mit dem GSR-Picaxe-14M2-Controllerboard kann man unser Kettenfahrzeug per Infrarot, per Bluetooth, per Funk oder eigenständig (mittels Programmierung und z.B. Taster oder Ultraschallsensor) steuern. Da der Picaxe-14M2-Chip nur mit maximal 90mA belastet werden darf, braucht man für die Steuerung von Motoren einen Motortreiberchip (L293D). Die Pins B.1 bis B.4 sind mit den Input-Pins des L293D verbunden. Somit kann der Picaxe-14M2 über den L293D die beiden Getriebemotoren unseres Kettenfahrzeuges steuern.

## 8. Steuern von Gleichstrommotoren mit dem Motortreiber-IC L293D

Mit dem Motortreiber IC L293DNE lassen sich zwei kleine Motoren (max. 600mA pro Motor) gleichzeitig steuern. Der L293DNE hat 16 Pins. An die Pins 1, 9 und 16 muss eine Versorgungsspannung von 4,5V bis 7V für den IC angelegt werden. An Pin 8 (Vcc 2) kommt die Spannungsversorgung für den Motor (4,5V bis 36V). Die Input-Pins 1 bis 4 sind am Mikrokontroller angeschlossen und steuern den IC. Die Output-Pins gehen zu den Motoren. Die Pins 4, 5, 12 und 13 sind an die Masse / Ground angeschlossen. Das folgende Schaubild soll die Anschlüsse nochmals verdeutlichen:



Quelle: <http://www.wirmachenroboter.com/node/35987>

## 9. Steuern der beiden Getriebemotoren

Zum Steuern der Motoren des GSR-Kettenfahrzeuges brauchen wir die Pins B.1 bis B.4, die jeweils eine Verbindung zu den IC-Pins 1A, 2A, 3A und 4A haben. Schaltet man die Pins B.1 und B.4 auf high und die Pins B.2 und B.3 auf low, dann fährt das Fahrzeug vorwärts.

high B.1  
low B.2  
low B.3  
high B.4

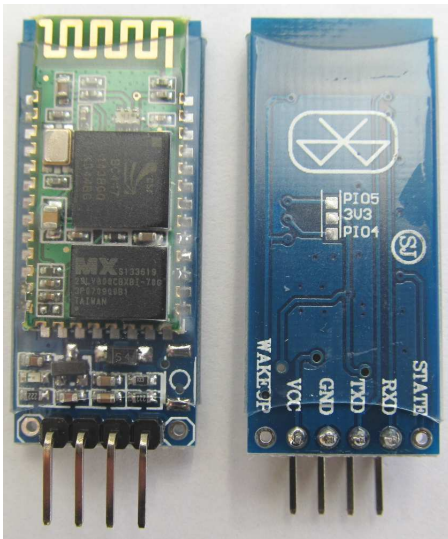
Das geht natürlich mit dem Befehl *let PinsB* schneller! So sieht das kurze Programm aus, wenn das Kettenfahrzeug vorwärts fahren soll:

**Let dirsB = %11111111** ; alle B-Pins werden als Ausgabe-Pins definiert  
**Let pinsB = %00010010** ; die Pins B.1 und B.4 sind high, alle anderen low

Hier eine „**Wahrheitstafel**“ für die Steuerung der beiden Motoren:

	B.7	B.6	B.5	B.4	B.3	B.2	B.1	B.0	Fahrtrichtung:
%	0	0	0	1	0	0	1	0	vorwärts
%	0	0	0	0	1	1	0	0	rückwärts
%	0	0	0	1	0	1	0	0	Drehung rechts
%	0	0	0	0	1	0	1	0	Drehung links
%	0	0	0	0	0	0	0	0	Motoren stoppen
%	0	0	0	1	1	1	1	0	Motoren stoppen

## 10. Steuern mit dem HC-06 Bluetooth-Controller und der App: GSR Drive



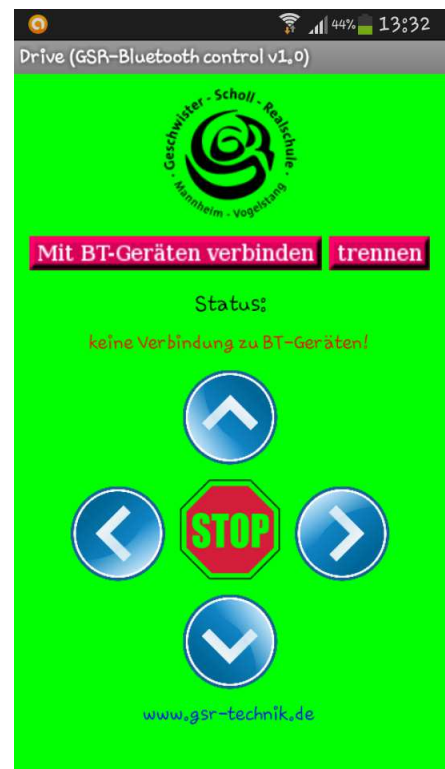
Den HC-06 kann man mit backplane (wichtig) über ebay, banggood, aliexpress oder andere Portale erwerben. Das eigentliche HC-06 Bluetooth Modul ist mit einer blauen Adapterplatte (backplane) verbunden und kann wegen eines Spannungsreglers mit 3,3V bis 6V Spannung versorgt werden. Der HC-05 kann als Master (senden und empfangen) und der HC-06 nur als Slave (empfangen) arbeiten.

Der serielle Bluetooth-Controller HC-06 hat mehrere Anschlüsse, die mit dem **GSR-Picaxe-14M2-Controllerboard** verbunden werden müssen. VCC an den Pluspol (2-Pin-Buchsenleiste), GND an Minuspol (Buchse 1 der 6-Pol-Buchsenleiste), TXD an Pin C.O, RXD (frei). Das Blinken der LED bedeutet, dass der HC-06 betriebsbereit ist aber noch nicht mit einem anderen Gerät (Handy, PC) verbunden ist.

Dann muss der HC-06 mit dem Handy gekoppelt werden. Einfach nach neuen Bluetooth Geräten suchen. Der serielle Bluetooth Controller wird als HC-06 gefunden. Um sich koppeln zu können, muss noch die Pin: 1234 eingegeben werden.

Dann muss der HC-06 mit dem Handy gekoppelt werden. Einfach nach neuen Bluetooth Geräten suchen. Der serielle Bluetooth Controller wird als HC-06 gefunden. Um sich koppeln zu können, muss noch die Pin: 1234 eingegeben werden.

Zum Steuern des **GSR-Picaxe-14M2-Controllerboards** kann man die kostenlose App: **GSR Drive** im Google Play Store herunterladen. Hat man die App gestartet, muss man den Button „Mit BT-Geräten verbinden“ drücken. Anschließend den HC-06 auswählen und los geht's...



Hier das einfache Steuerprogramm für den Picaxe-14M2:

**; Dateiname: 14M2 GSR Drive HC-06 Kettenfahrzeug.BAS**

```

let dirsB=%11111111      ; alle B-Pins werden Ausgabepins
let pinsB=%00000000      ; alle B-Pins werden auf 0V gesetzt (low)
setfreq M8                ; Die Geschwindigkeit des 14M2 wird auf 8 Mhz erhöht

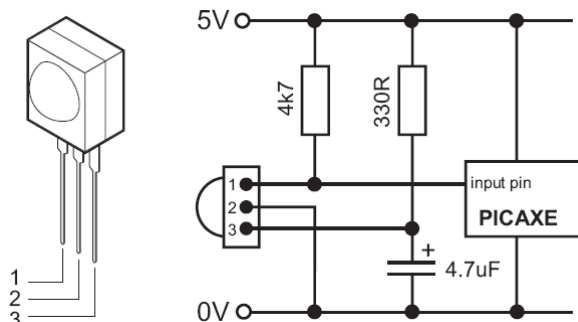
main:                     ; Start des Hauptprogramms
serin C.0,N9600_8,b1      ; C.0 empfängt über HC-06 Signal und speichert es in b1
If b1 = 127 then vw        ; wenn Vorwärtsbutton dann gehe ins Unterprogramm vw
If b1 = 63  then re        ; wenn Rechtsbutton dann gehe ins Unterprogramm re
If b1 = 126 then rw        ; wenn Rückwärtsbutton dann gehe ins Unterprogramm rw
If b1 = 31  then li        ; wenn Linksbutton dann gehe ins Unterprogramm li
If b1 = 125 then stehen    ; wenn Stopbutton dann gehe ins Unterprogramm stehen
goto main

vw:
let pinsB=%00010010        ; B1 und B4 high => L293D Pin 1 und Pin 4 high => Vorwärts
goto main
re:
let pinsB=%00010100        ; B2 und B4 high =>L293D Pin 2 und Pin 4 high => Rechtsdrehung
goto main
rw:
let pinsB=%00001100        ; B2 und B3 high => L293D Pin 2 und Pin 3 high => Rückwärts
goto main
li:
let pinsB=%00001010        ; B1 und B3 high =>L293D Pin 1 und Pin 3 high => Linksdrehung
goto main
stehen:
let pinsB=%00000000        ; alle Pins low =>L293D werden alle Pins low => Stop
goto main

```

## 11. Steuerung des GSR-Picaxe-14M2-Controllerboards per Infrarot

Unser Controllerboard lässt sich mit dem Infrarotsensor LED020 von Revolution Education ([www.tech-supplies.co.uk](http://www.tech-supplies.co.uk)) steuern. Dazu wird eine (Universal)-Fernbedienung benötigt, die Sony-Fernseher steuern kann (z.B. die TVR010 von Revolution Education (ca. 6 €) oder die Universalfernbedienung E+P FB13 von Pollin (Art: 620 186, ca. 2 €)).



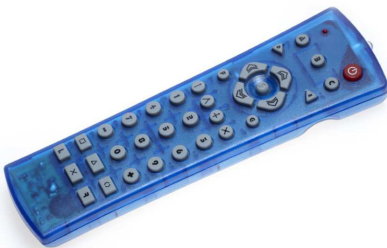
Das nachfolgende Schaltbild zeigt, wie der Infrarotsensor LED020 an unser Projektboard angeschlossen werden muss. Auf der Homepage von Picaxe findet man auch ein Schaltbild ohne den 330-Ohm-Widerstand. Tests haben ergeben, dass die Schaltung auch ohne den Widerstand funktioniert.



Mit dem Befehl **irin** können die Signale der Fernbedienung über den Infrarotsensor empfangen werden. In dem kurzen Programmbeispiel ist der Pin 1 des Infrarotsensors am Pin C.0 des Picaxe-14M2 angeschlossen. Der empfangene „Wert“ wird in der Variablen b1 gespeichert und über **debug** kann man den Wert angezeigt bekommen.

```
main:
    irin C.0, b1
    debug
goto main
```

Die Tabelle zeigt, bei welchem Tastendruck der Fernbedienung welche Werte über den irin-Befehl „erzeugt“ werden.



Symbol KEY_POWER= 21	Symbol KEY_UP = 16
Symbol KEY_1 = 0	Symbol KEY_DOWN = 17
Symbol KEY_2 = 1	Symbol KEY_LEFT = 19
Symbol KEY_3 = 2	Symbol KEY_RIGHT = 18
Symbol KEY_4 = 3	Symbol KEY_BAR = 96
Symbol KEY_5 = 4	Symbol KEY_TENT = 54
Symbol KEY_6 = 5	Symbol KEY_VERT_CROSS = 37
Symbol KEY_7 = 6	Symbol KEY_DIAG_CROSS = 20
Symbol KEY_8 = 7	Symbol KEY_MINUS = 98
Symbol KEY_9 = 8	Symbol KEY_PLUS = 11
Symbol KEY_0 = 9	

Beim Drücken der Taste 1 auf der Fernbedienung wird die LED 1 eingeschaltet.

```
main:
    irin C.0, b1      ; wartet auf ein Signal an Pin C.0; der Wert wird dann in b1 gespeichert
    if b1 = 0 then led1 ; wird die Taste 1 gedrückt (b1 = 0) dann gehe ins Unterprogramm led1
goto main           ; Sprung zum Label main
```

```
led1:                ; Anfang des Unterprogramms led1
    high B.0         ; der Pin B.0 wird high
goto main           ; Sprung zum Label main
```

Der Befehl **irin** kann noch erweitert werden: **IRIN [timeout, address], pin, variable**

**timeout** = eine Variable oder ein Wert, der eine Pause in Millisekunden erzeugt.

**adress** = wird keine Taste gedrückt, springt das Programm in das Unterprogramm address.

```
main:
    irin [100, ledoff], C.0, b1
    if b1 = 0 then led1
goto main
```

Beim linken Programmbeispiel leuchtet die LED 1, wenn die Taste 1 gedrückt wird. Wird keine Taste gedrückt, dann folgt ein Sprung in das Unterprogramm (Label) **ledoff**.

```
led1:
    high B.0
goto main
```

```
ledoff:
    low B.0
goto main
```

**Achtung!!!** Die Schaltung mit dem Infrarotsender muss an den Picaxe-14M2 angeschlossen sein, wenn ein neues Programm aufgespielt werden soll. Sonst kann die Fehlermeldung: **Hardware nicht gefunden ...** kommen.