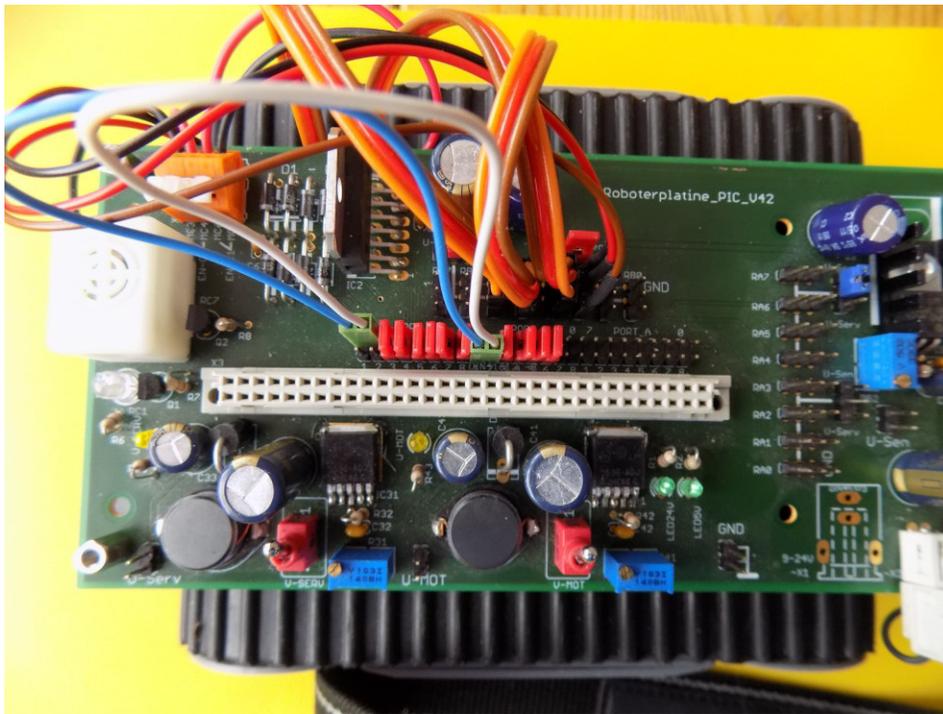


# Anschluss des Roboters an das Uno-Modul-Shield-V2



Das Arduino-Board besitzt 3 Ports von denen leider nicht alle Anschlüsse genutzt werden können.

So sollten die Arduino-Pins 0 und 1 frei bleiben, da diese zum Programmieren des  $\mu\text{C}$  dienen (Serieller Anschluss vom USB-Chip). Die beiden Freigabeanschlüsse des Motortreibers müssen mit einem PWM-Pin verbunden werden.

Die folgende Seite zeigt eine mögliche **Verbindung zwischen Arduino-Modul und Roboter.** (Gilt für die Verwendung mit einem Arduino Uno)

Zum Testen der korrekten Funktion sind zwei Testprogramme abgedruckt.

**Programmlisting 1 testet die Motoren.** Die linken Räder müssen sich in Vorwärtsfahrtrichtung bewegen während die blaue LED leuchtet.

Nach 2 Sekunden müssen sich die rechten Räder vorwärts bewegen während der Summer aktiv ist. Sollte dies nicht der Fall sein sind die Motoren vertauscht bzw. verpolt. An der orangenen Klemmleiste können die Anschlüsse getauscht werden.

**Programmlisting 2 testet die 4 Liniensensoren.** Die Variablen x1 bis x4 für die Reflexlichtschranken werden über die serielle Schnittstelle ausgegeben. Der Roboter erkennt ein schwarzes Isolierband optimal auf einer hellen Tischplatte.

# Mögliche Verbindung Arduino-Board -> Roboter

Arduino PIN	Steckplatz 64pol Stecker	Beschr. Ard.		Jumperleiste Roboter	Funktion Roboter	
A0	B1	PORTC	PORTA	0	• •	analoge Eingänge für optionale Sensoren
A1	B2			1	• •	
A2	B3			2	• •	
A3	B4			3	• •	
A4	B5			4	• — •	Hindernis links
A5	B6			5	• — •	Hindernis rechts
--	B7			6	• •	nicht belegt
--	B8			7	• •	nicht belegt
0	B9	PORTD	PORTB	0	• •	sollte frei bleiben
1	B10			1	• •	sollte frei bleiben
2	B11			2	• — •	Liniensensor 1 links
3 *	B12			3	• — •	Liniensensor 2
4	B13			4	• — •	Liniensensor 3
5 *	B14			5	• — •	Liniensensor 4 rechts
6 *	B15			6	• — •	
7	B16			7	• — •	
8	B17	PORTB	PORTC	0	• — •	blaue LED
9 *	B18			1	• — •	Freigabe Motor links
10 *	B19			2	• — •	Freigabe Motor rechts
11 *	B20			3	• — •	Motor rechts MC3
12	B21			4	• — •	Motor rechts MC4
13	B22			5	• — •	Motor links MC5
--	B23			6	• — •	Motor links MC6
--	B24			7	• — •	Summer

\* PWM fähig

# Programm 1: Testet die Motoren

```
#define blaueled 8
#define motorlinksfreigabe 9
#define motorrechtsfreigabe 10
#define motorrechtsplus 11
#define motorrechtsminus 12
#define motorlinksplus 13
#define motorlinksminus 6
#define summer 7
#define liniensensor1 2 //links
#define liniensensor2 3 //schwarze Linie = HIGH
#define liniensensor3 4 //weiss = LOW
#define liniensensor4 5 //rechts

void setup() {
  pinMode(blaueled, OUTPUT);
  pinMode(motorlinksfreigabe, OUTPUT);
  pinMode(motorrechtsfreigabe, OUTPUT);
  pinMode(motorrechtsplus, OUTPUT);
  pinMode(motorrechtsminus, OUTPUT);
  pinMode(motorlinksplus, OUTPUT);
  pinMode(motorlinksminus, OUTPUT);
  pinMode(summer, OUTPUT);
  pinMode(liniensensor1, INPUT);
  pinMode(liniensensor2, INPUT);
  pinMode(liniensensor3, INPUT);
  pinMode(liniensensor4, INPUT);

  digitalWrite(motorlinksplus, HIGH);
  digitalWrite(motorrechtsplus, LOW);
  digitalWrite(motorrechtsminus, HIGH);
  digitalWrite(motorlinksminus, LOW);

  Serial.begin(9600);
}

void loop() {
  //linke Räder vorwärts LED an
  digitalWrite(blaueled, HIGH);
  digitalWrite(summer, LOW);
  analogWrite(motorlinksfreigabe, 200);
  analogWrite(motorrechtsfreigabe, 0);
  delay(2000);

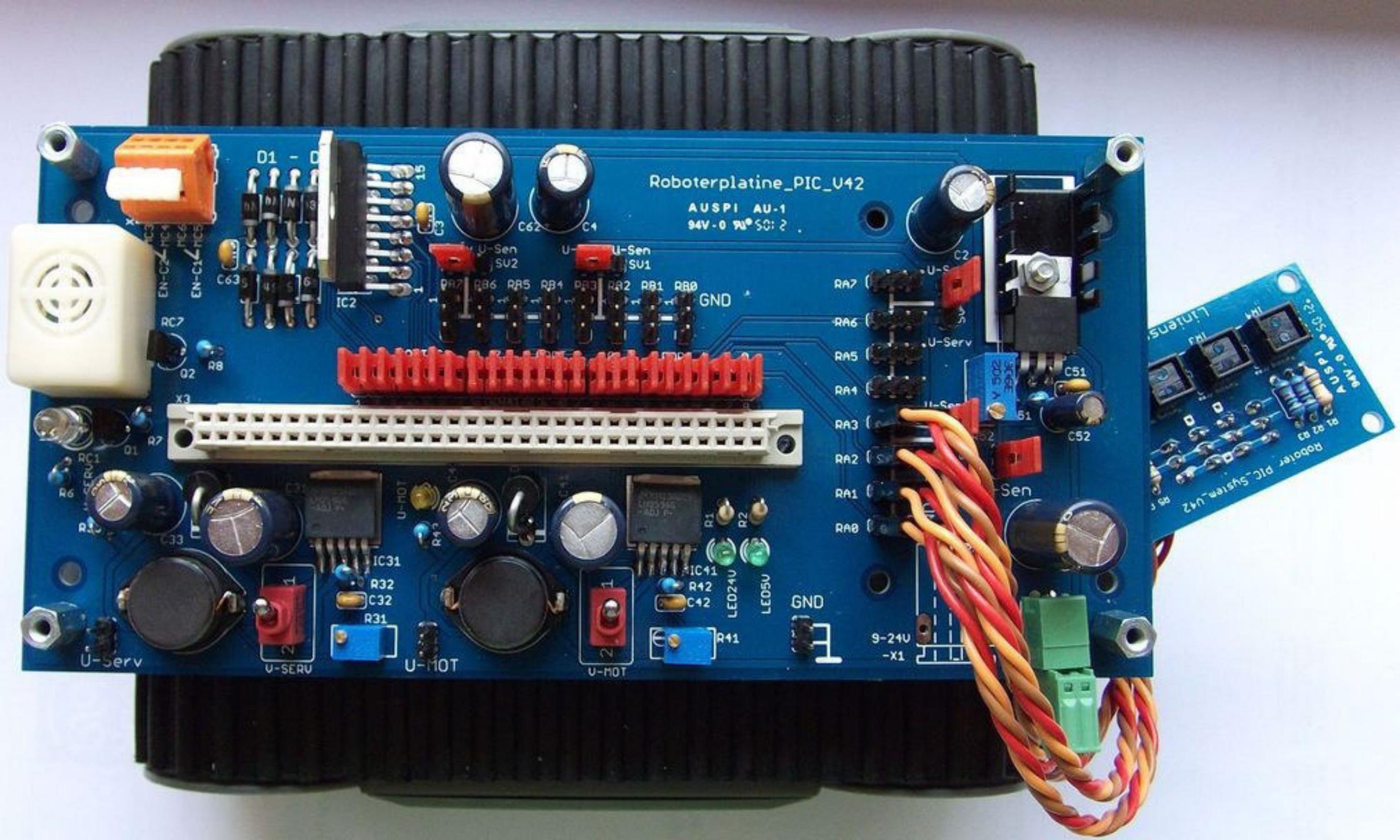
  //rechte Räder vorwärts Summer an
  digitalWrite(blaueled, LOW);
  digitalWrite(summer, HIGH);
  analogWrite(motorlinksfreigabe, 0);
  analogWrite(motorrechtsfreigabe, 200);
  delay(2000);
}
```

# Programm 2: Testet die Liniensensoren

```
#define blaueled 8
#define motorlinksfreigabe 9
#define motorrechtsfreigabe 10
#define motorrechtsplus 11
#define motorrechtsminus 12
#define motorlinksplus 13
#define motorlinksminus 6
#define summer 7
#define liniensensor1 2 //links
#define liniensensor2 3 //schwarze Linie = HIGH
#define liniensensor3 4 //weiss = LOW
#define liniensensor4 5 //rechts

void setup() {
  pinMode(blaueled, OUTPUT);
  pinMode(motorlinksfreigabe, OUTPUT);
  pinMode(motorrechtsfreigabe, OUTPUT);
  pinMode(motorrechtsplus, OUTPUT);
  pinMode(motorrechtsminus, OUTPUT);
  pinMode(motorlinksplus, OUTPUT);
  pinMode(motorlinksminus, OUTPUT);
  pinMode(summer, OUTPUT);
  pinMode(liniensensor1, INPUT);
  pinMode(liniensensor2, INPUT);
  pinMode(liniensensor3, INPUT);
  pinMode(liniensensor4, INPUT);
  Serial.begin(9600);
}

void loop() {
  int x1 = digitalRead(liniensensor1); //linker Sensor
  int x2 = digitalRead(liniensensor2);
  int x3 = digitalRead(liniensensor3);
  int x4 = digitalRead(liniensensor4); //rechter Sensor
  Serial.print("Sensor 1: "); Serial.print(x1); Serial.println(" ganz links");
  Serial.print("Sensor 2: "); Serial.println(x2);
  Serial.print("Sensor 3: "); Serial.println(x3);
  Serial.print("Sensor 4: "); Serial.print(x4); Serial.println(" ganz rechts");
  Serial.println("_____ 1 = HIGH = Sensor dunkel");
  Serial.println();
  delay(2000);
}
```



Roboterplatine\_PIC\_U42

AUSPI AU-1  
94V-0 90° SO: 2

D1 - D

U-Sen  
SU2

U-Sen  
SU1

RA7 RA6 RA5 RA4 RA3 RA2 RA1 RA0

GND

X3

R7

R6

C33

U-Serv

U-SERU

U-MOT

U-MOT

GND

9-24V  
-X1

Roboter PIC-System\_U42

AUSPI 1.05.01

94V-0 90° SO: 2

Roboter PIC-System\_U42

AUSPI 1.05.01

94V-0 90° SO: 2



## Aufbau:

Das Einlöten der Bauelemente kann z.B. in folgender Reihenfolge erfolgen

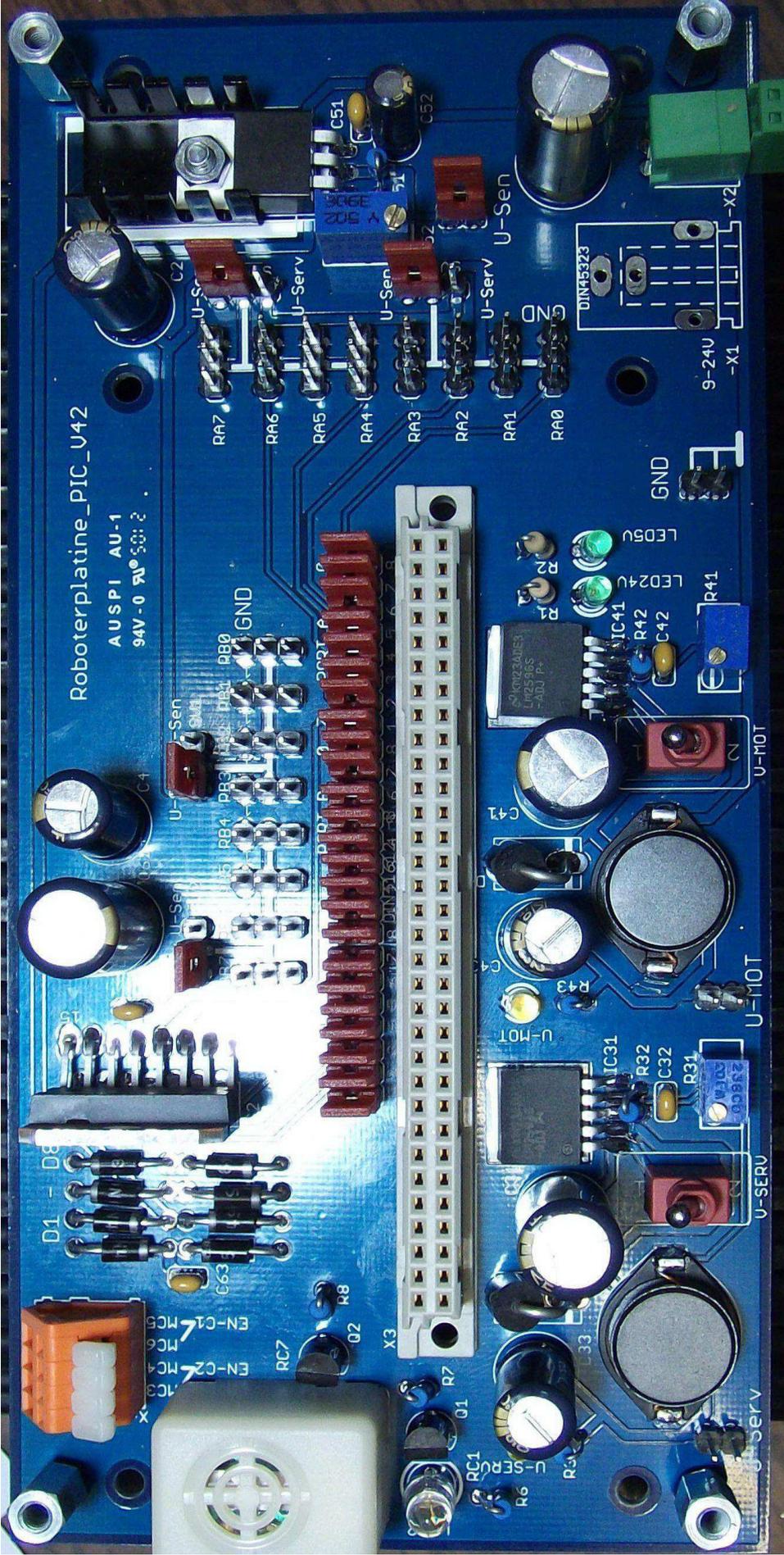
## Roboter

IC31	LM2596S
IC41	LM2596S
R1	6k8
R2	470
R6	1k
R7	10k
R8	10k
R31	10k Trimmer
R32	1k
R33	1K
R41	10k Trimmer
R42	1k
R43	1K
R51	270
R52	5k Trimmer
C3	100n
C32	100n
C42	100n
C51	100n
C63	100n
D1	1N4935
D2	1N4935
D3	1N4935
D4	1N4935
D5	1N4935
D6	1N4935
D7	1N4935
D8	1N4935
LED2	LED blau
LED5V	LED grün
LED24	LED grün
U-MOT	LED gelb
U-SER	LED gelb
D31	1N5822
D41	1N5822
Q1	BC547B
Q2	BC547B
L2	68 $\mu$ H
L3	68 $\mu$ H
X3	FAB64SE
X-MOT	233-504

-X1	Buchse „40 F 140“ -
IC2	L298
IC51	LM317 mit Kühlkörper
SG1	Summer
V-MOT	Kippschalter „04 G 230“
V-SER	Kippschalter „04 G 230“
Alle Jumper wie abgebildet einlöten	
C52	1 $\mu$
C43	220 $\mu$
C2	220 $\mu$
C4	220 $\mu$
C33	220 $\mu$
C62	470 $\mu$
C1	470 $\mu$ /35V
C41	470 $\mu$
C31	470 $\mu$

## Sensorleiste zum Roboter

Bezeichnung	Wert
R1	68
R2	10k
R3	10k
R4	10k
R5	68
R6	10k
IR1	CNY70
IR2	CNY70
IR3	CNY70
IR4	CNY70
JP1	Jumper 3pol
JP2	Jumper 3pol
JP3	Jumper 3pol
JP4	Jumper 3pol



Roboterplatine\_PIC\_U42

AUSPI AU-1  
94V-0 70° 50:2

D1 - D8

EN-C1  
MC6  
MC4  
EN-C2  
MC3

PC7

C63

R8

Q1

R7

PC1

U-SERV

R6

R5

C33

U-SERV

U-SEN

RA7

RA6

RA5

RA4

RA3

RA2

RA1

RA0

U-SERV

C2

C51

C52

C51

C51

U-Sen

9-24V

-X1

-X2

GND

LED5U

LED2

LED2

LED2

LED2

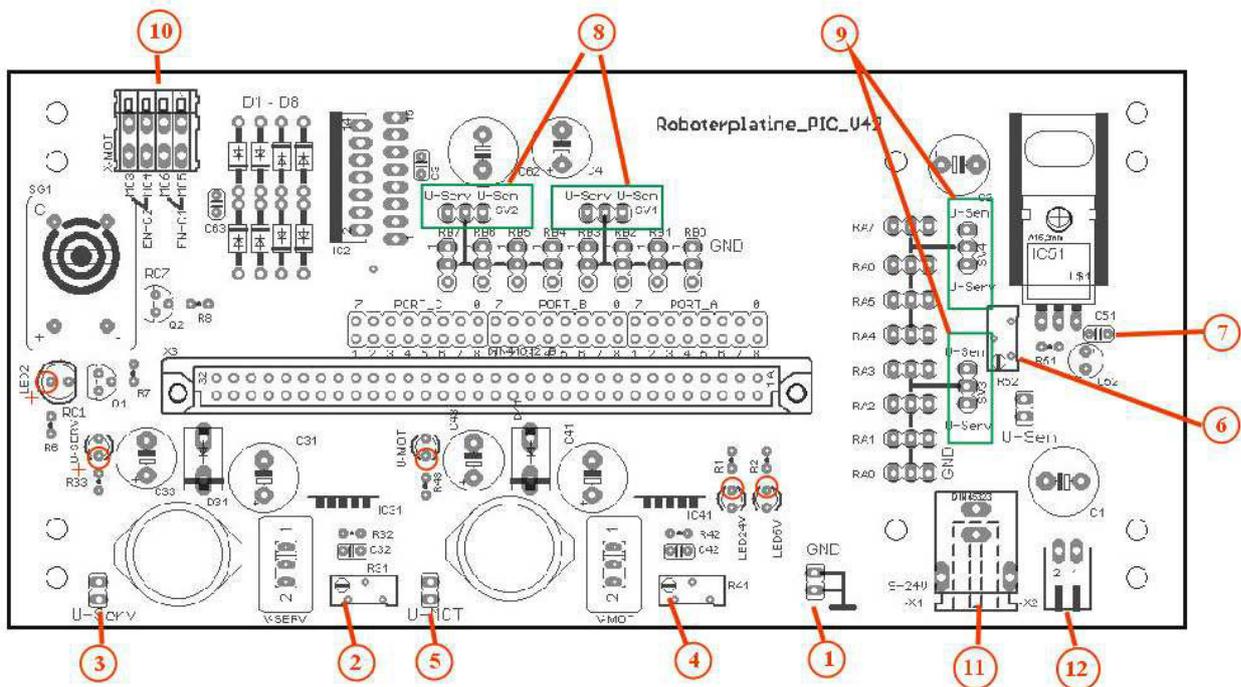
LED2

LED2

LED2

## Anschlüsse und Abgleich:

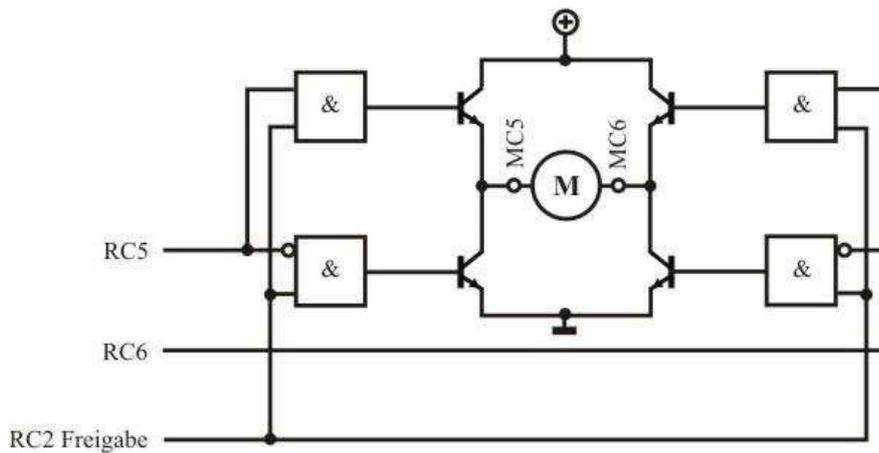
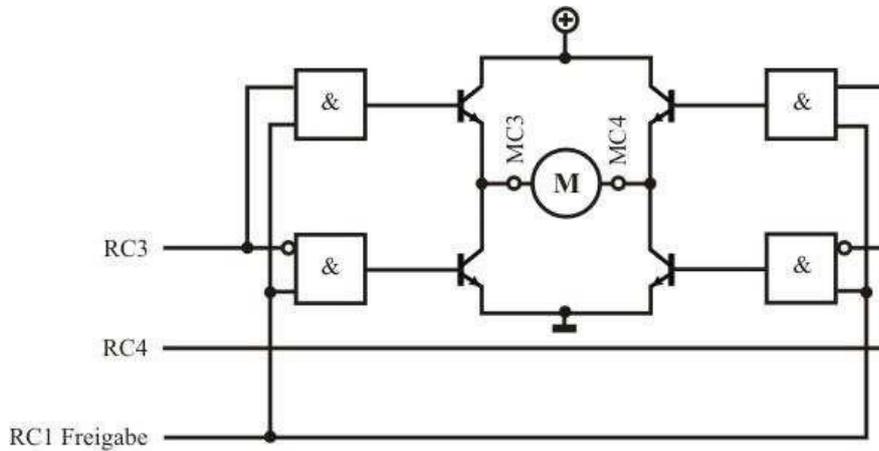
- (1) Die Kontakte ① dienen als Messpunkt – GND
- (2) Die Spannung für die Servos (in der Regel 6V) kann hier eingestellt werden
- (3) Messpunkt für die Servospannung
- (4) Abgleich der Motorspannung – etwa 5V, je nach Ansteuerung
- (5) Messpunkt für die Motorspannung
- (6) Abgleich der Sensorspannung, je nach verwendeten Sensoren (meist 5V)
- (7) Messpunkt für die Sensorspannung
- (8) Jumper zur Umschaltung von jeweils 4 Anschlüssen (PORTB) auf Servospannung (links) oder Sensorspannung (rechts).
- (9) Jumper zur Umschaltung von jeweils 4 Anschlüssen (PORTA) auf Servospannung (links) oder Sensorspannung (rechts).
- (10) Motorklemmen – Belegung je nach Steuerprogramm
- (11) Spannungsversorgung. Die Klemmen 11 und 12 sind verbunden
- (12) Spannungsversorgung 8-24V. Rechter Kontakt +



## Motoransteuerung

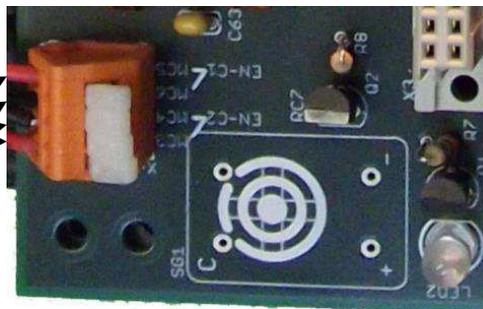
Die Motoren werden über Vollbrückenmotortreiber angesteuert.

Ist z.B. RC3 = 1 und RC4 = 0, so fließt der Strom von MC3 nach MC4 durch den Motor, wenn die Freigabe RC1 ebenfalls 1 ist. Normalerweise wird die Freigabe über PWM angesteuert um den Motorstrom zu steuern.



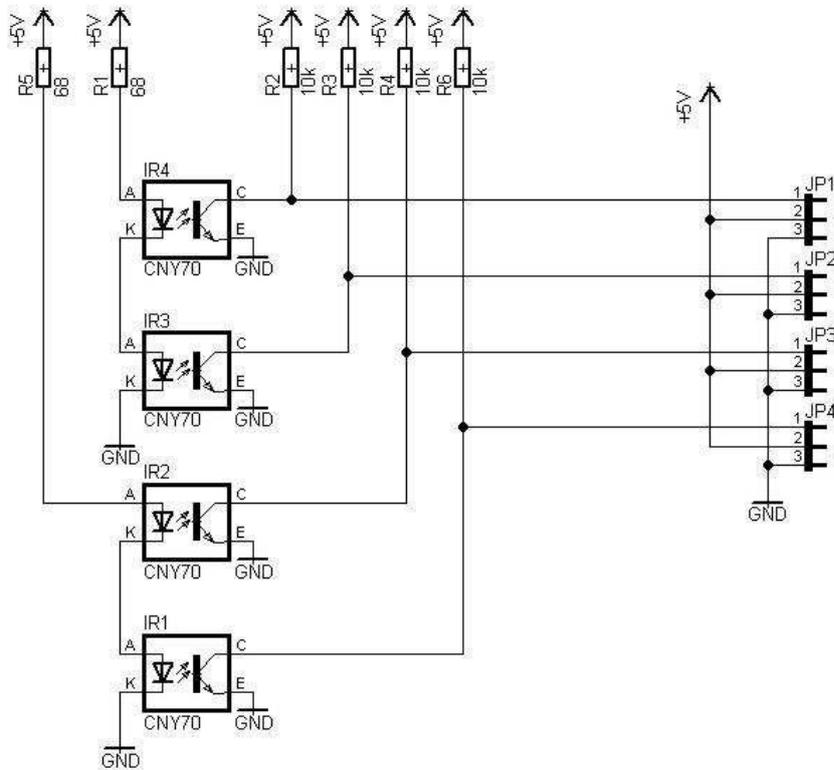
PORTC ist also wie folgt belegt:

- RC0 – Blaue LED
- RC1 – Freigabe M1, M2
- RC2 – Freigabe M3, M4
- RC3 – M2
- RC4 – M1
- RC5 – M3
- RC6 – M4
- RC7 – Summer



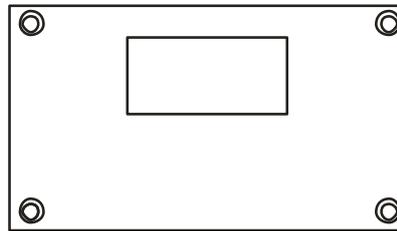
## Sensoren

Zur Linienverfolgung werden vier IR-Reflexkoppler benutzt.  
Die Sensoren können digital oder analog ausgewertet werden.



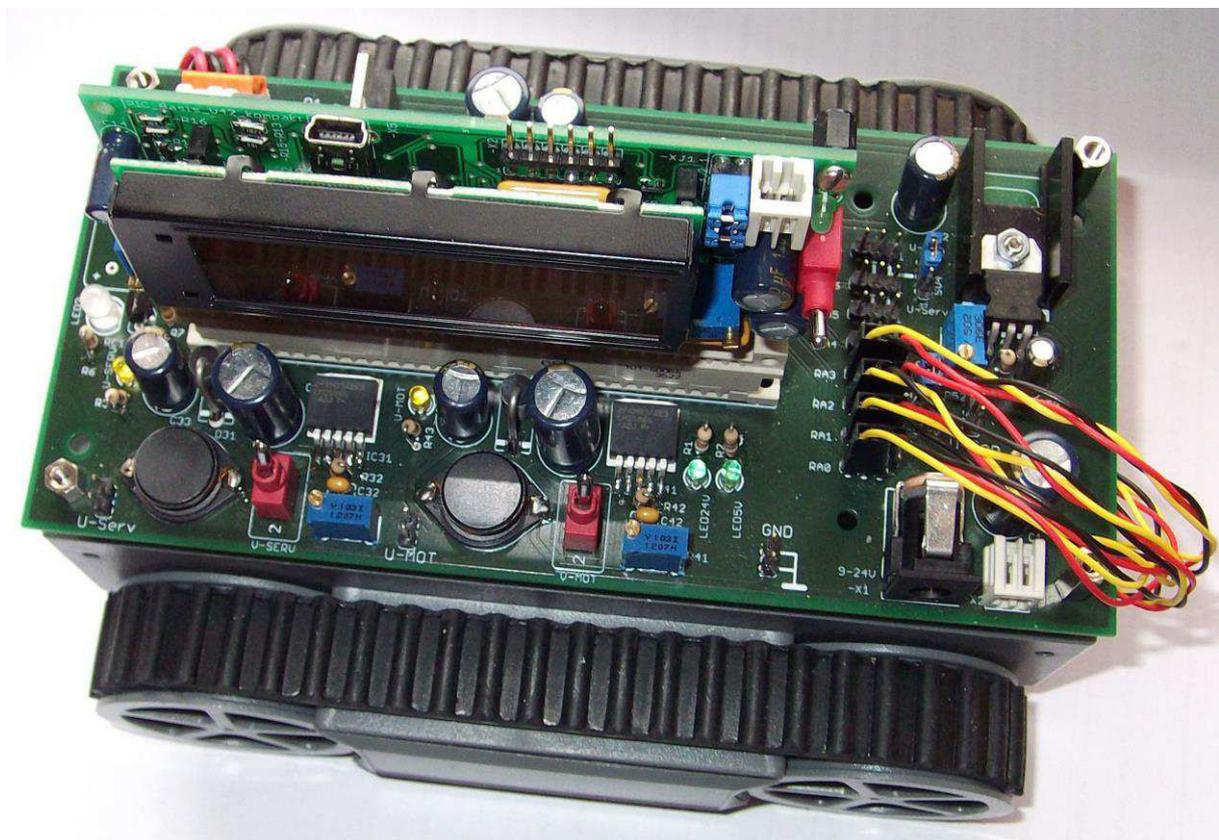
## Hinweise zum Aufbau

Die Liniensensoren sollten im vorderen Teil unter das Chassis geschraubt werden. Zum Bohren und ausschneiden kann die Schablone benutzt werden.

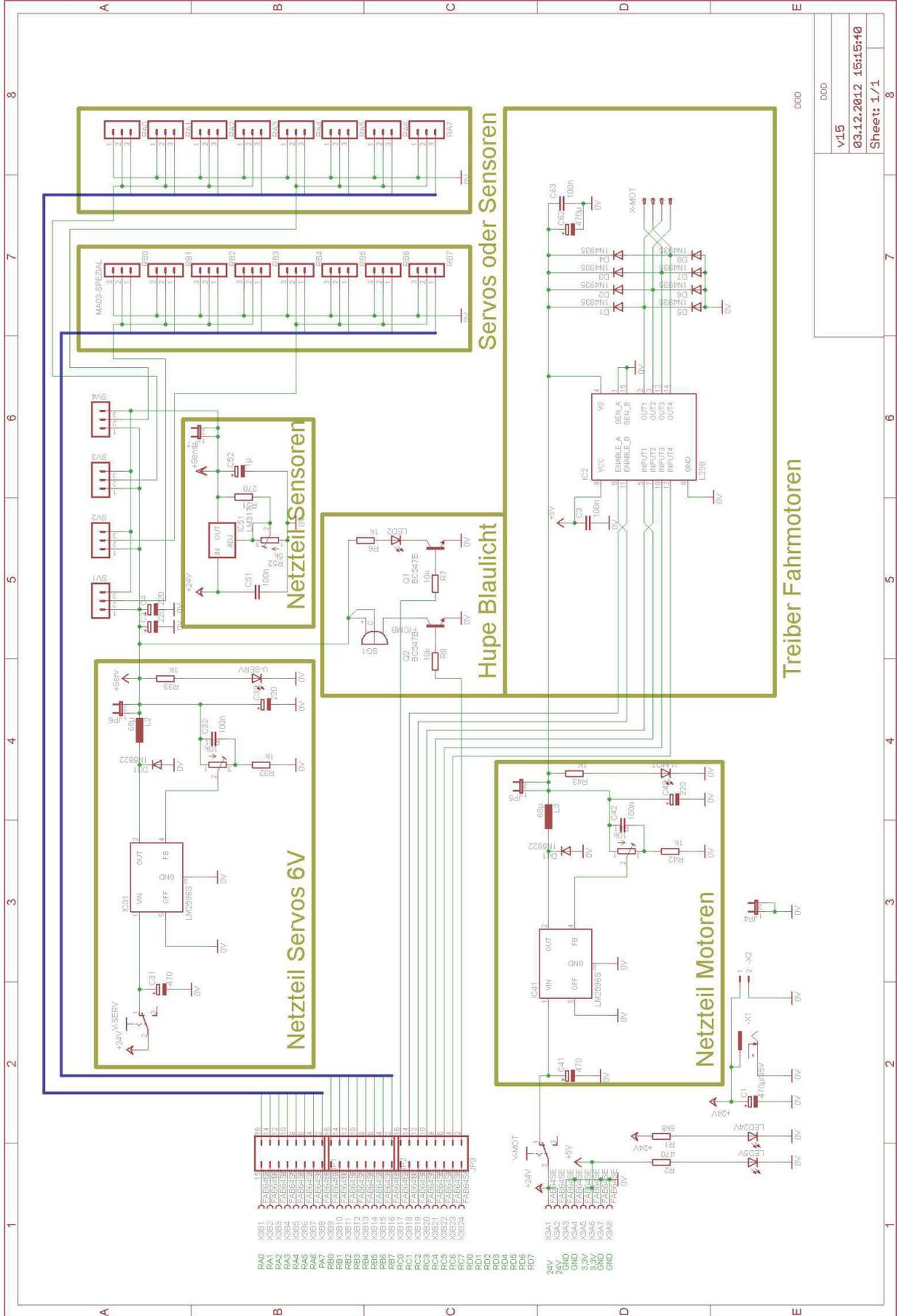


**Bohrschablone für Liniensensoren**

Zum Befestigen der Platine werden vier Abstandsbolzen in das Chassis geschraubt. Dazu wird entweder 3mm Gewinde bohren, oder die Löcher aufbohren und die Bolzen einkleben.

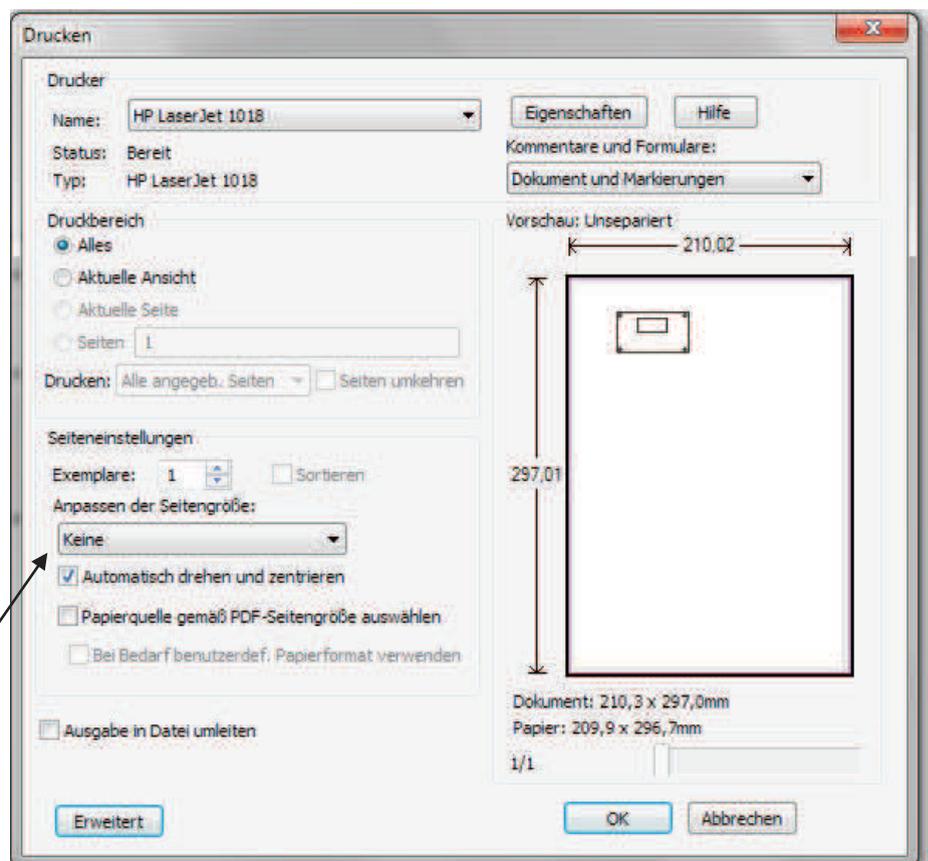
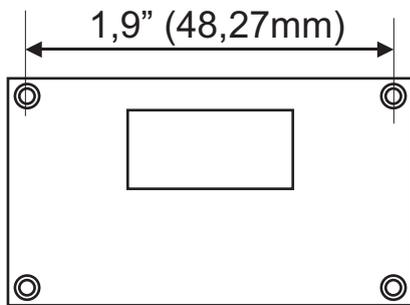


# Schaltplan



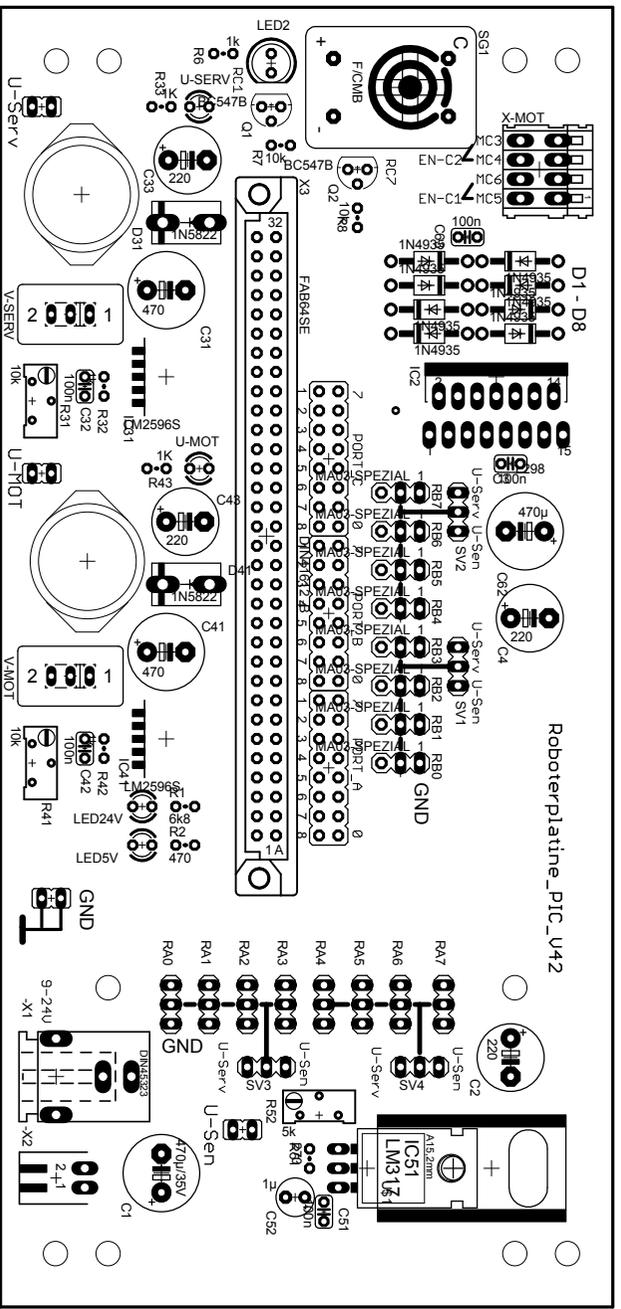
V15	000
03.12.2012 15:15:10	
Sheet: 1/1	

1	2	3	4	5	6	7	8
---	---	---	---	---	---	---	---



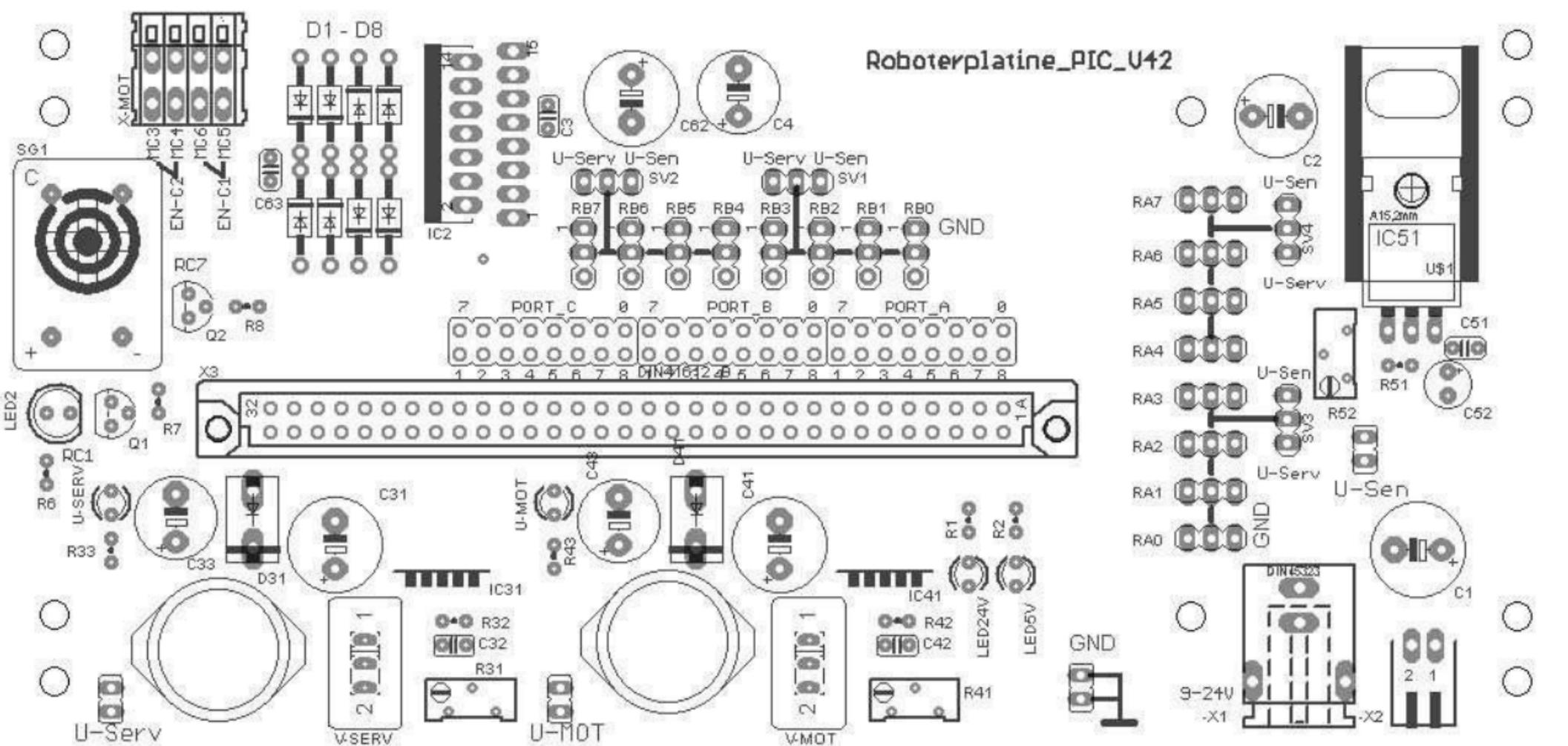
Für maßstäblichen über Acrobat-Reader:  
Anpassen der Seitengröße "keine" einstellen.

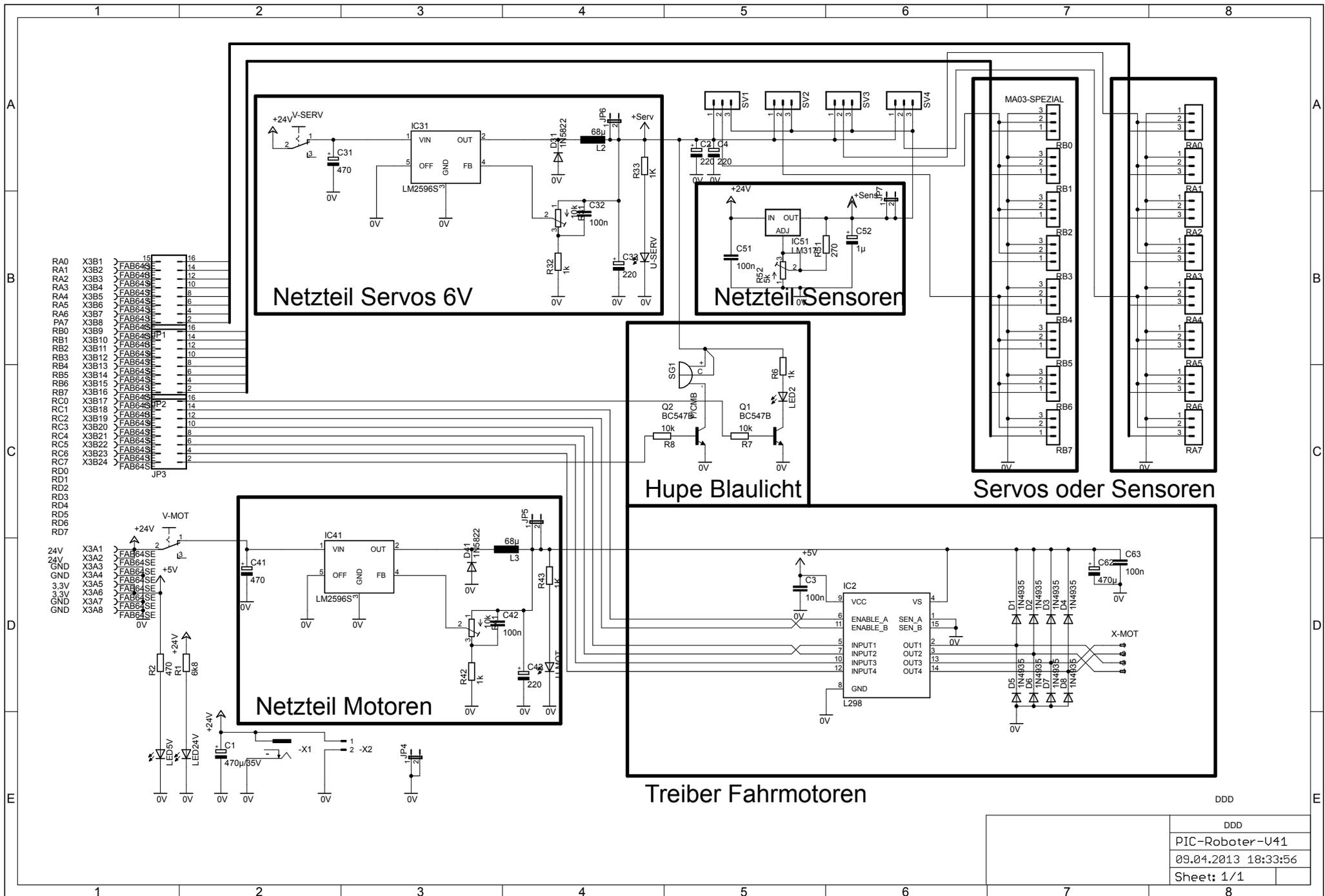






# Roboterplatine\_PIC\_U42





- RA0 X3B1
- RA1 X3B2
- RA2 X3B3
- RA3 X3B4
- RA4 X3B5
- RA5 X3B6
- RA6 X3B7
- PA7 X3B8
- RB0 X3B9
- RB1 X3B10
- RB2 X3B11
- RB3 X3B12
- RB4 X3B13
- RB5 X3B14
- RB6 X3B15
- RB7 X3B16
- RC0 X3B17
- RC1 X3B18
- RC2 X3B19
- RC3 X3B20
- RC4 X3B21
- RC5 X3B22
- RC6 X3B23
- RC7 X3B24
- RD0
- RD1
- RD2
- RD3
- RD4
- RD5
- RD6
- RD7

- X3A1
- X3A2
- X3A3
- X3A4
- X3A5
- X3A6
- X3A7
- X3A8

DDD	
PIC-Roboter-U41	
09.04.2013 18:33:56	
Sheet: 1/1	

