

# Miniaturoboter "Willi"

Aufbau und Inbetriebnahme

Ausgabe 2/98



## Inhaltsverzeichnis

<b>1.</b>	<b>Einleitung</b>	2
<b>2.</b>	<b>Aufbau des Roboters</b>	3
2.1.	Leiterplattenbestückung	3
2.2.	Mechanikaufbau	4
2.3.	Inbetriebnahme	7
<b>3.</b>	<b>Programmierung</b>	9
3.1.	Programminstallation	9
3.2.	Erste Programme	10
<b>4.</b>	<b>Anhang</b>	13
	Befehlsbeschreibung, Fehlermeldungen	
	Stromlaufpläne	
	Anschlußbelegung ausgewählter Bauelemente	
	Haftungsausschluß	
	Stückliste	

### **1. Einleitung**

Die Robotik-Artikelserie in der Elrad (10/97..12/97) stieß auf ein breites Leserinteresse. Das Resultat vieler Anfragen und Wünsche liegt nun vor Ihnen. Der Roboterbausatz. Nicht zu kompliziert, auch für Einsteiger aber trotzdem kein Spielzeug.

Ich habe deshalb versucht die Aufbauanleitung zu einfach wie möglich zu fassen. Auf keinen Fall sollte jedoch vergessen werden, daß der "Willi" trotz allem ein recht komplexes Ding geworden ist.

Lesen Sie bitte die Aufbauanleitung bevor sie beginnen durch. Denjenigen unter Ihnen die häufig mit Lötkolben und Lötzinn umgehen, kommt vieles Bekannt vor, überlesen Sie es einfach.

Alle anderen brauchen nicht zu verzweifeln. Solange der Roboter keine Rauchzeichen gibt, ist das Rennen noch nicht verloren. Wenn irgendetwas nicht funktioniert, bleiben sie ruhig schimpfen sie auf den Konstrukteur (also mich) und auf die Politik (das sind alle anderen) und öffnen sie eine gute Flasche Rotwein.

Ich habe mich bemüht die Anleitung Schritt für Schritt zu erklären. Wenn also etwas nicht "tut", blättern sie zurück und versuchen sie es erneut.

Gerade bei der Kombination von Soft- und Hardware in einem Produkt kann man immer wieder die tollsten Überraschungen erleben. Denken sie beim Experimentieren immer daran: "Der Computer hat immer recht", zumindest aus seiner Sicht.

## 2. Aufbau des Roboters

### 2.1. Leiterplattenbestückung

Der Roboter wird als Bausatz geliefert. Er enthält alle notwendigen Teile. Sie brauchen lediglich Werkzeug und Lötzinn.

Der Aufbau gliedert sich in drei Abschnitte. Der erste Abschnitt umfaßt die Leiterplattenbestückung, der zweite Abschnitt den Mechanikaufbau und im dritten Teil wird die Elektronik abgelichen.

Im Anhang finden Sie die Stückliste zur Leiterplatte. Die Einzelteile sind mit Positionsnummern gekennzeichnet.

Bevor Sie mit der Arbeit beginnen senken Sie mit einem das Bohloch das später unter der Batterie liegt (es ist die einzige nichtdurchkontaktierte Bohrung) mit einem 8 mm Spiralbohrer. Die Senkung ist zur Befestigung des "dritten" Rades notwendig.

Dann werden die Widerstände bestückt. Die Anschlußdrähte werden vorsichtig abgewinkelt und dann durch die Bohrungen geschoben. Danach werden die Abschlüsse gekürzt.

Zum Löten verwenden Sie bitte einen ElektroniklötKolben (ca. 20 Watt) oder eine Lötstation. Die LötKolbenspitze sollte zunderfrei und gut verzinkt sein. Eine flache Meißelform der Spitze hat sich gut bewährt. Lötwasser oder andere säurehaltige Löthilfen dürfen nicht benutzt werden. Selbst beim vermeintlich sorgfältigen Entfernen der Rückstände bleibt oft genug der aggressiven Substanzen zurück, um die Leiterzüge zu zerstören. Besonders empfindlich reagieren die IC-Fassungen auf eingedrungene Löthilfen. Also bitte nur nur Kolophonium verwenden.

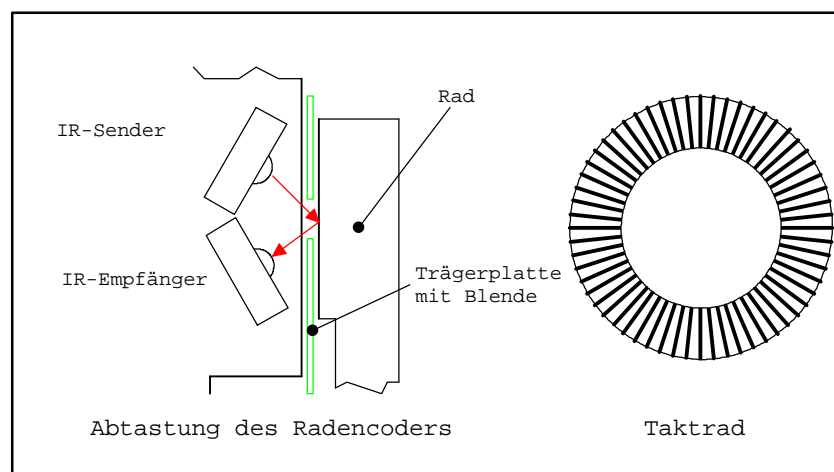
Nach den Widerständen werden die Dioden bestückt. Hier bitte die Drähte vorsichtig mit einer kleinen Zange abwinkeln. Brüche oder Risse im Glaskörper der Dioden sind Fehler, die oft erst nach längerer Betriebszeit zutage treten.

Die weiteren Bauelemente werden dann entsprechend ihrer Zugänglichkeit und Höhe (die kleinen zuerst) aufgelötet.

Zuallerletzt werden die optoelektronischen Bauelemente bestückt.

Es werden zwei Fototransistoren geliefert. Sie müssen entscheiden ob Sie die Transistoren nach unten oder nach vorn "blickend" montieren. Sie können auch zwei zusätzliche Fototransistoren z.B. SFH309 (bei Conrad-Electronic erhältlich) bestücken; die Auswahl erfolgt dann über die Kurzschlußbrücken an JP3.

Besondere Sorgfalt erfordert die Montage der Reflexkoppler Q5 und Q6. Die optische Achse soll so gut wie möglich durch die Blende der Motorträger weisen.



Um die Position besser zu fixieren, wird die Trägerplatte provisorisch mit Hilfe der Distanzbolzen an der Leiterplatte befestigt. Die beiden Bauelemente können nun durch die Blende beobachtet, fixiert und angelötet werden. Die Position der Sendediode ist mit einem "D" auf der L-Seite der Leiterplatte gekennzeichnet.

Wenn alle Bauelemente bestückt sind, wird die Leiterplatte sehr sorgfältig auf Lötbrücken oder Zinnspritzer untersucht.

Ist dies zur Zufriedenheit erfolgt, wird die Schaltung einem kurzen Einschalttest unterzogen. Wenn eine Laborstromversorgung zur Verfügung steht, sollte sie verwendet werden; 9-Volt und Strombegrenzung auf ca. 10 mA einstellen. Ansonsten wird die Batterie angeschlossen.

Nach dem Einschalten wird die 5-Volt am 78L05 überprüft. Die Schaltkreise sind zu diesem Zeitpunkt nicht gesteckt.

Der Stromfluß sollt nur einige mA betragen. Sollte dies nicht der Fall sein, die Platte nochmals auf Lötbrücken untersuchen. Das gilt auch, wenn die Spannung am Regler außerhalb des Bereiches von 5 Volt (+/- 10%) ist.

Bei einem erfolgreichen Test wird die Platte vorerst zur Seite gelegt.

## 2.2 Mechanikaufbau

Dem Bausatz liegen die beiden Motoren und Räder bereits vorgefertigt bei. Ihre Aufgabe ist das Aufkleben der Taktscheiben auf den Innenrand der Räder. Bitte beachten Sie, daß es eine rechte bzw. linke Taktscheibe gibt.

Die Taktscheiben sind wischfeste Klebeetiketten. Sie werden sorgfältig ausgeschnitten und angepaßt. Auch der Innenteil der Taktscheibe wird ausgeschnitten (die Scheibe, eigentlich Ring, muß über den Motor gehen).

Danach wird der Trägerstreifen entfernt. Die Klebeflächen der Räder werden vorher von anhaftenden Fett- oder Schmutzteilen gereinigt.

Eine Pinzette erleichtert das Fixieren auf den Rädern. Die Taktscheibe wird gleichmäßig ohne Luftblasen oder Falten angedrückt. Die Genauigkeit der Bewegungen des Roboters wird hier stark beeinflusst.

Motor und Rad können nun auf der Trägerplatte befestigt werden.

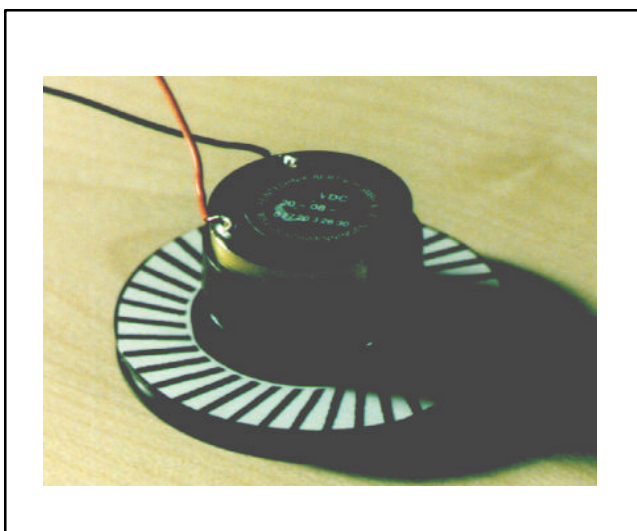


Bild 2: Motor mit Rad und Taktscheibe

Zur Montage wird das Rad vorsichtig gedreht, so daß die Bohrungen im Rad genau über den Befestigungspunkten des Motors an der Trägerplatte liegen.

Mit einem passenden Schraubenzieher werden die Schrauben M1,4 x 5 aufgenommen und in die jeweilige Bohrung am Motor eingeschraubt. Die drei Schrauben werden geheftet und erst zum Schluß angezogen. Bitte hier mit großer Vorsicht drehen (nach fest kommt ab !!).

Nicht vergessen, es gibt einen rechten und einen linken Motor. Die schwarzen Striche der Taktscheibe müssen die Blende in der Trägerplatte beim Durchdrehen des Motors vollständig ausfüllen.

Sind beide Motoren montiert, wird sicherheitshalber der Rundlauf und der Seitenschlag des Rades überprüft. Eine Spannung von ca. 1,5 Volt (Mignon-Zelle) versetzt den Motor in langsame Drehung.

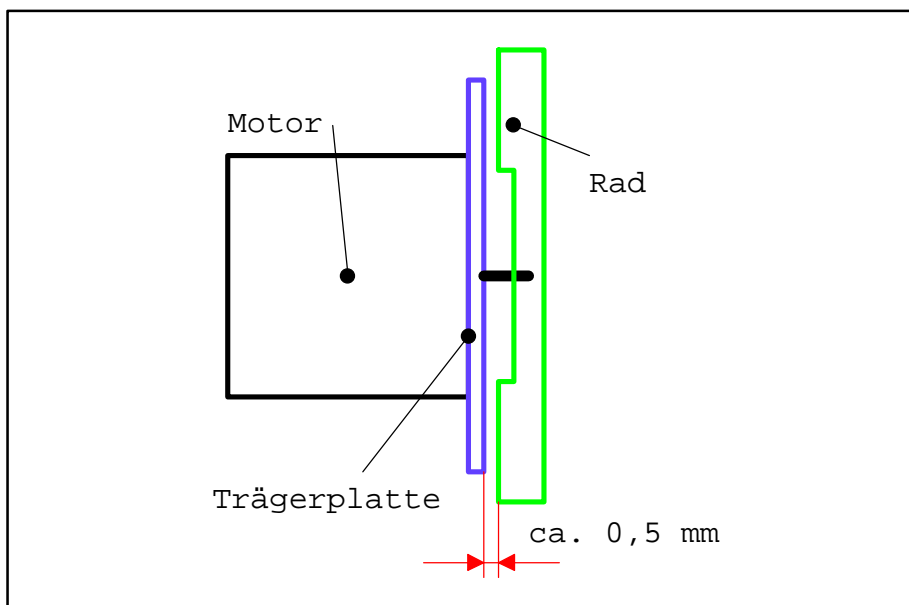


Bild 3: Maßbild Motorbefestigung

Das Rad soll sich gleichmäßig, ohne Berührungen an der Trägerplatte, drehen. Der Luftspalt zwischen Rad und Trägerplatte beträgt ca. 0,5 mm. Falls das Rad doch "eiern" sollte, prüfen Sie den richtigen Sitz des Motors auf der Trägerplatte.

Das einzige Einzelteil, das sie völlig selbst bauen, ist der Stoßkontakt. Die Gestaltung des Teiles obliegt ihrer Fantasie. In der Elrad-Serie hatte "Willi" zwei Kontakte in Form von Fühlern. Das sieht zwar dekorativ aus, ist jedoch etwas unpraktisch.

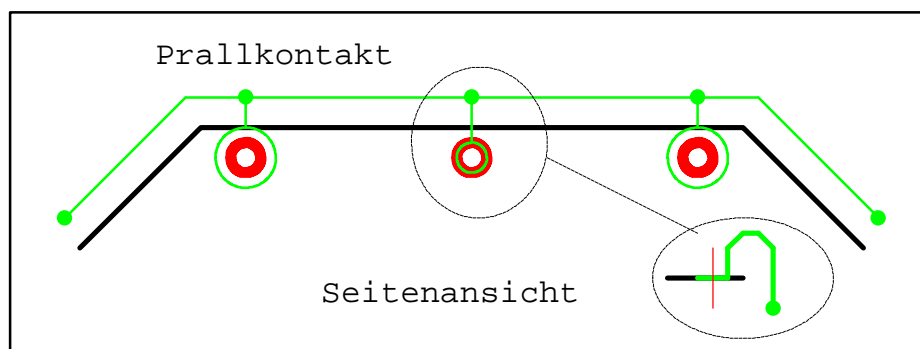


Bild 4: Befestigung des Prallkontaktes

Besser erscheint die oben abgebildete Variante. Der beiliegende Draht wird zu einem Bügel gebogen. An diesen Bügel werden drei Ösen angelötet. Die beiden äußeren sind groß genug, um ohne anzustoßen um die beiden Kontaktstifte (M2-Schrauben) gelegt werden zu können.

Die mittlere Öse wird an der Leiterplatte verschraubt und mit dem Bügel verlötet. Das Verbindungsstück zwischen Leiterplatte und Bügel wird dabei wegen der besseren Federwirkung U-förmig gebogen (siehe Seitenansicht im Bild).

Damit ragt der Prallkontakt nicht zu weit über die Kontur des Roboters hinaus. Der Kontakt verhakt sich dann bei Drehungen des Roboters nicht so leicht.

Die Gesamtmontage ist somit beinahe fertig. Die O-Ringe werden als Reifen auf die Räder gezogen. Danach werden die beiden Motoren mit den Distanzbolzen an der Leiterplatte befestigt. An der hinteren Bohrung wird das Stützrad angeschraubt; M3 Schraube hindurchstecken, mit M3 Mutter sichern und die Kugel festziehen.

Das Klettband wird passend zugeschnitten und an den Akku (nicht im Lieferumfang) und die Leiterplatte angeklebt.

Ein letzter Blick, die beiden Abdeckfolien an den Räder fehlen noch (bitte aufkleben), dann ist der Roboter fertig (die Abdeckplatte kommt erst nach dem Abgleich auf den Roboter).

## 2.3 Inbetriebnahme

Jetzt soll er fahren, der "Willi".

Doch bevor es richtig losgeht, müssen die Sensoren abgeglichen werden.

Dazu hat die Firmware des Roboters ein internes Testprogramm. Die Leiterplatte haben sie bereits geprüft, wenn nicht bitte nachholen (Abschnitt 2.1).

Zum Abgleich kommt der Roboter auf eine Streichholzschachtel oder ein Holzklötzchen. Die Räder sollen sich frei drehen können.

Dann wird der Roboter eingeschaltet. Während des Einschaltens halten sie den Startknopf gedrückt. Wenn sie den Knopf loslassen, piept der Roboter einmal und es beginnt sich das rechte Rad zu drehen. Die Leuchtdiode LED1 ist dabei entweder ein oder aus. Mit einem Schraubenzieher verstellen sie den Regler R11 solange bis die LED im Rythmus der Striche auf der Taktscheibe blinkt.

Mit der erneuten Betätigung der Taste wird der nächste Punkt des Abgleichprogrammes aktiviert. Die nachfolgende Tabelle erläutert alle notwendigen Schritte zum Einstellen des Roboters.

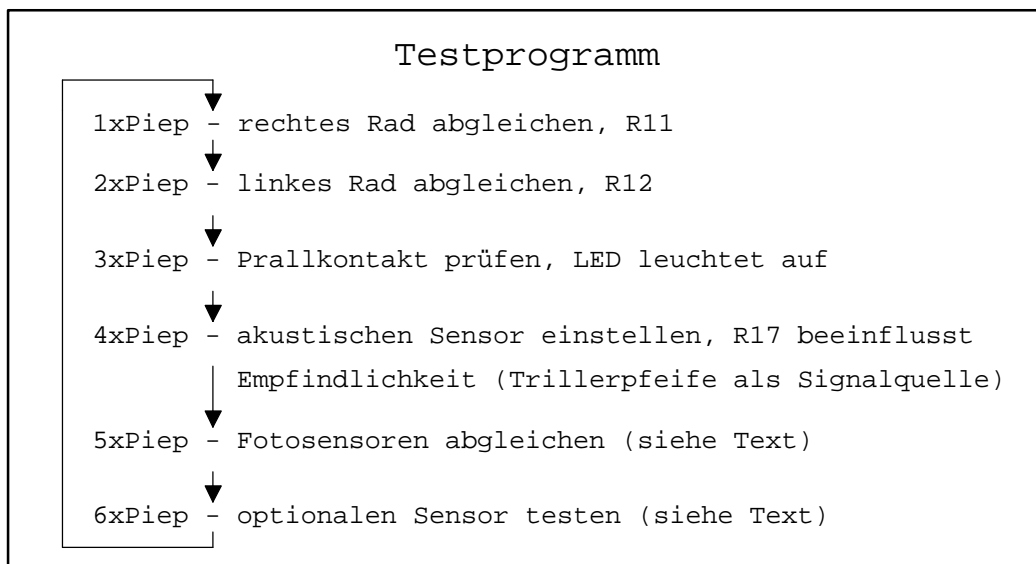
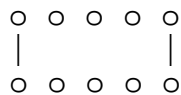


Bild 6: Programmablauf Selbsttest

Der Abgleich der Fotosensoren ist nicht ganz einfach. Die Einstellvorschrift hängt von der Konfiguration ab (Leitstrahl oder Leitlinienerkennung).

Konfiguration 1 (Leitstrahlerkennung):

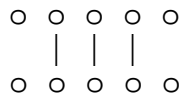
- die Jumper sind außen gesteckt, der mittlere bleibt offen



- ein Lichtquelle (Taschenlampe) wird nun aus einiger Entfernung auf die Foto-transistoren gerichtet
- mit einem Meßgerät werden die Spannungen am Pin 10 und 13 des ST6 (U1) gemessen
- mit R10 wird die Spannung an Pin 10 der an Pin 13 angeglichen
- im Selbsttest (5xPiep) ist die LED ein (kein Licht auf den Sensoren), bzw. aus beide Sensoren beleuchtet
- ist nur ein Sensor beleuchtet, blinkt die LED

### Konfiguration 2 (Leitlinienerkennung):

- die Jumper sind innen gesteckt



- der Roboter wird auf die Fläche gestellt, auf der er später fahren soll, dann werden

die Fototransistoren durch Bewegen des Gefährtes über der Leitlinie abgeglichen dabei ist auf genügenden Kontrast zu achten.

Die Schaltschwelle der Eingänge des ST6 liegen bei ca. 2,5 Volt. Damit eine Lichtquelle erkannt wird, muß die Pegeländerung genügend groß werden. Direktes Sonnenlicht ist zu vermeiden.

Der letzte Abgleichpunkt behandelt den optional anschließbaren Sensor. Sie können dort eigene Sensoren anklemmen. Die Schaltschwelle liegt auch hier bei 2,5 Volt.

Auf der Leiterplatte sind zum Anschluß des eigenen Sensors drei Kontakte vorgesehen (5V, GND, Signal). Erweiterungen sollten jedoch nur erfahrene Bastler vornehmen. Für Beschädigungen des Roboters durch zusätzliche Sensoren wird keine Garantie übernommen.

Sind sie mit diesem Abgleichvorgang fertig, können sie das "mitgelieferte" Testprogramm starten. Sie schalten den Roboter dazu wieder ein und drücken dann den Startknopf. "Willi" fährt ein kurzes Stück, piept, dreht dann, fährt ein kurzes Stück, piept, dreht dann, fährt .....

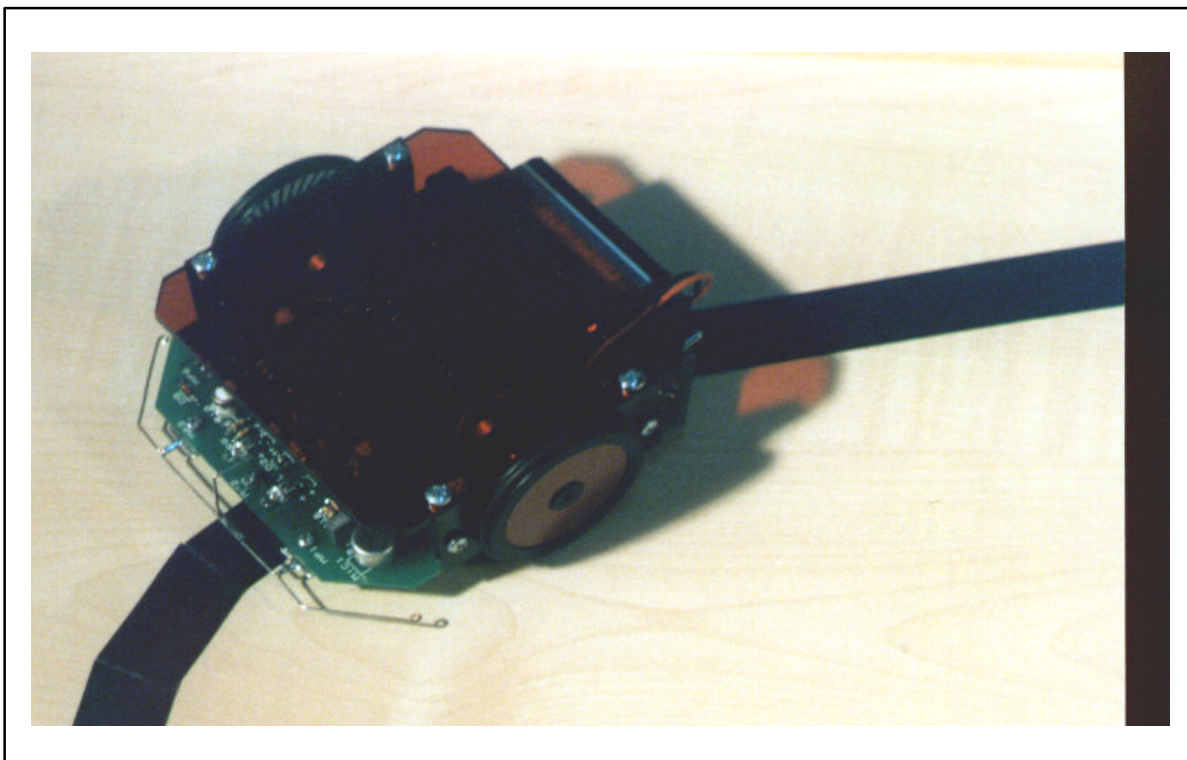


Bild 7: Willi auf dem Strich

### 3. Programmierung

#### 3.1 Installation des Mikrobotassemblers

Die beiliegende Diskette beinhaltet den Mikrobot-Assembler. Dieses Programm erzeugt den notwendigen Steuercode für "Willi".

Sie legen die Diskette in ihr Computerlaufwerk und starten die Installationsprozedur.

**A:\>setup <ENTER>**

Das Programm wird damit auf ihrer Festplatte installiert. Durch Doppelclick auf das Programmsymbol wird es gestartet.

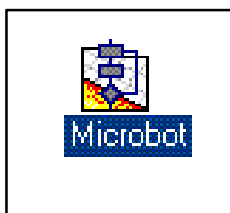


Bild 8: Mikrobot-Assembler Icon

Nach dem Programmstart erscheint der Eröffnungsbildschirm.



Bild 9: Mikrobot-Assembler

Die in der Menüleiste zu erkennenden Programmpunkte entsprechen weitgehend den Windows Konventionen.

Mit **Datei** werden die einzelnen Programme geladen, gespeichert und gedruckt.

**Bearbeiten** und **Suchen** beinhalten die üblichen Funktionen zur Editierung von Texten.

Über **Hilfe** können Erläuterungen zu Programmfunktionen bekommen werden.

Die beiden wichtigsten Funktionen verbergen sich hinter dem Menüpunkt **Programm**.

Zum einen handelt es sich um die Anweisung, das eingegebene Programm zu assemblieren und zum anderen das fertige Programm in den Controller des Roboters zu laden.

## 3.2 Erste Programme


Mit einigen Beispielen werden sie mit den Funktionen des Mikrobot-Assemblers am schnellsten vertraut.

Beginnen wir mit einer einfachen Aufgabenstellung. "Willi" soll einfach geradeaus fahren und am Ende seines Weges piepsen.

Dafür werden vorerst drei Befehle benötigt: `step`, `port` und `speed`.


Sie öffnen das Programm durch Doppelklick auf das Icon. Dann tippen sie los (für die Schreibmüden unter Ihnen, laden Sie die Datei `example1.st6`) :

```
start:
  speed 1      ; Geschwindigkeit 1 genügt - der Roboter soll nicht rasen
  step  47     ; sie können auch andere Zahlen einsetzen
  port  %1110 ; jetzt piepst "Willi"
```

Das eingetippte Programm wird assembliert. Dazu klicken sie am einfachsten das entsprechende Kurzsymbol an. 

Sie erhalten eine Meldung über den Vorgang und im Fehlerfall auch eine passende Fehlerbeschreibung. Probieren sie diesen Mechanismus ruhig aus (z.B. unsinnige Geschwindigkeitsangaben wie 10000 oder ähnlich).

Das übersetzte Programm können sie auch anschauen. Die Datei-Extension lautet `.LST`.

Damit "Willi" mit dem Programm arbeiten kann, muß es "geladen" werden. Der PC und "Willi" werden dazu mit einem Kabel verbunden. "Willi" ist bei dieser Prozedur ausgeschaltet. Die starten die Laderoutine mit Klick auf das entsprechende Icon. Sie können natürlich auch die Menüpunkte aus **Programm** bzw. die Funktionstasten **F9**  und **F10** verwenden.

Ist die Verbindung zwischen PC und "Willi" hergestellt, können Sie mit der Datenübertragung beginnen. Schalten Sie bitte den "Willi" erst dann ein, wenn Sie vom Programm dazu aufgefordert werden. Die Datenübertragung erfolgt nicht über RS232-Protokoll obwohl der entsprechende Anschluß am PC benutzt wird. Es kommt die synchrone Schnittstelle (SPI) des ST6 zum Einsatz.

Nach der Aufforderung zum Einschalten des "Willi" quittiert der Roboter das mit zwei Pieptönen. Gleichzeitig leuchtet die LED auf. Sollte das nicht passieren, schalten Sie den Roboter aus und wieder ein. Falls es dann immer noch nicht funktioniert prüfen sie die Kabelverbindung.

Während der Übertragung flackert die LED (eigentlich blinkt sie bei jedem empfangenen Zeichen). Die erfolgreiche Übernahme des Programmes quittiert "Willi" mit einem erneuten Piepton.

Sie ziehen dann die Klinkenbuchse ab und drücken den Startknopf. Gehorsam setzt sich "Willi" in Bewegung, absolviert seine 47 (oder wieviel auch immer) Schritte, piept und macht dann Unfug. Wieso tut er das ?

Nach dem `port`-Befehl haben sie zwar nichts angeordnet, das kann aber "Willi" nicht wissen. Er holt den nächsten Befehl aus seinem Speicher und führt ihn aus. Und irgendetwas steht immer im Speicher.

Eine kleine Programmänderung hilft schnell aus der Patsche:

```
start:
  speed 1      ; Geschwindigkeit 1 genügt - der Roboter soll nicht rasen
  step  47     ; sie können auch andere Zahlen einsetzen
  port  %1110 ; jetzt piepst der "Willi"
  jp    start ; alles von vorn
```

Jetzt läuft das Programm rund. Fahren, piepen, fahren, piepen .... Wenn das nichts ist :-)! Jedoch spätestens am ersten Hindernis werden die Grenzen der Maschinenintelligenz deutlich. Auch das Schieben und Drängen bis der Akku leer ist, bewegt den Kühlschrank auch nicht um einen Millimeter von der Stelle.

"Willi" muß auf Reize reagieren können.

Doch zuvor ein Programmbeispiel zur Leitlinienabtastung. Sie benötigen dazu eine Experimentierfläche. Idealerweise ist diese Fläche einfarbig, möglichst hell und gut reflektierend. Darauf werden mit schwarzem Isolierband "Leitlinien" aufgeklebt.

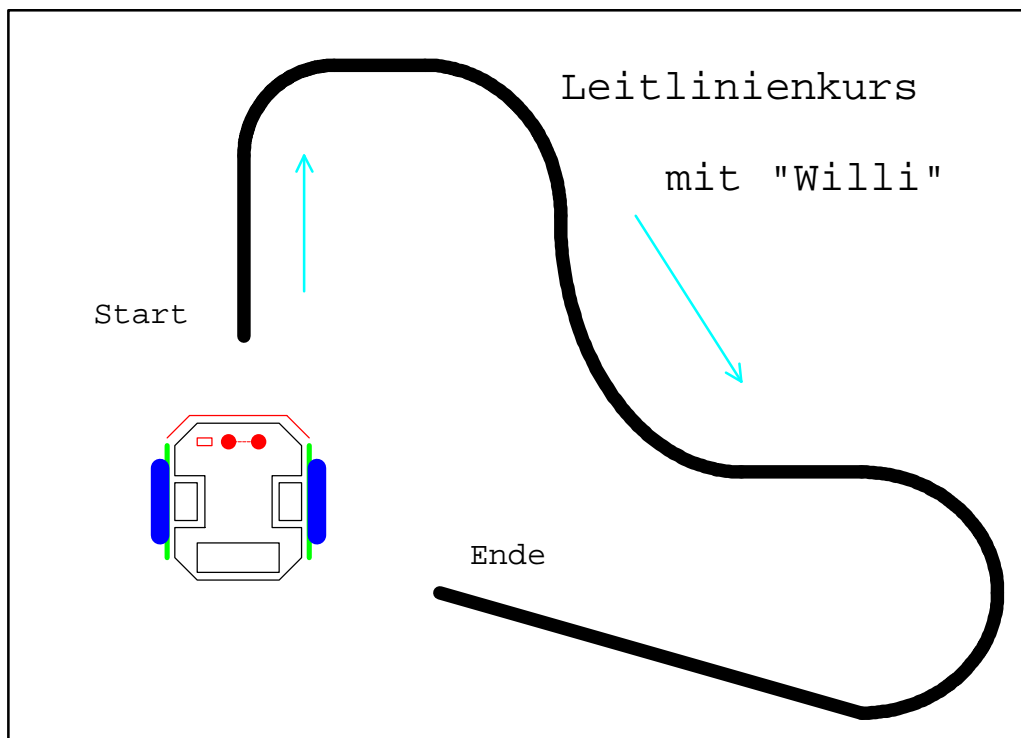


Bild 10: Elchtest für Willi

Der Roboter soll diesen Kurs abfahren. Dazu braucht er ein Programm. Die Sensoren müssen abgefragt und entsprechend der Abweichung vom Kurs müssen Korrekturbewegungen eingefügt werden.

Auch hier gibt es ein Beispiel (example2.st6):

```

;*****
;* Leitstrahlabtastung mit "Willi" *
;*****
#define main ; MainProzess definieren
speed 1 ; Geschwindigkeit einstellen

start:
step 10 ; 10 Schritte nach vorn

st1:
jband %00011,start ; kein Licht auf den Sensoren -> weiter
jband %00001,lk ; Korrektur links notwendig
jband %00010,re ; Korrektur rechts notwendig
jband %00000,ende ; Leitstrahl verloren

lk:
mturn 1 ; Korrekturbewegung links
jp st1 ; Test ob ausreichend

```

```

re:
    mturn -1          ; Korrektur rechts
    jp     st1       ; Test ob ausreichend

ende:
port  %1110          ; Piepton
e1:
port  %1101          ; LED ein
pause 1              ; 2 Sekunden Pause
port  %1100          ; LED aus
pause 1
jp     e1            ; Endlosschleife

;     ***     ENDE

```

Die Programmfunktion ist schnell erläutert. Der Roboter wird auf den Anfang der Leitlinie gesetzt und gestartet. Er folgt nun der Linie. Sobald einer der Sensoren nicht mehr über der Linie ist, wird eine Korrekturbewegung ausgelöst. Der Roboter prüft, ob die Korrektur ausreichend war und fährt dann weiter. Wenn die Linie zu Ende ist, bzw. der Roboter durch irgendein Ereignis vollends die Linie verliert, bleibt er stehen, piept kurz und verharrt dann blinkend.

Richtig perfekt wird das Programm erst, wenn der Roboter die Leitlinie selbständig wieder auffinden kann. Denken Sie sich das notwendige Programm dazu aus.

Problematisch wird es auch, wenn die Teststrecke mit Schikanen versehen wird. Kleine Holzklöße eignen sich dazu hervorragend.

Der nichtsahnende Roboter kommt nun des Wegs und prallt gegen ein Hindernis. Darauf muß er reagieren.

Damit er das kann, ist jedem Sensor ein eigener Prozeß zugeordnet. Trifft das vom Sensor erkannte Ereignis ein, startet das erforderliche Programm.

Ein Beispiel erläutert dies (example3.st6):

```

;*****
;*   Leitstrahlabtastung mit "Willi"           *
;*   mit Reaktion auf Hindernis              *
;*****
#define main                                ; Hauptprogramm wie example2.st6

#define bump
speed -1          ; Geschwindigkeit 1, rückwärts
step  5           ; 5 Schritte zurück
speed 1           ; vorwärts
turn  1           ; 90° Grad Drehung
step  10
turn -1          ; zurückdrehen
step  25
turn -1          ; weitere 90° Drehung
step  10
turn  1           ; ursprüngliche Richtung wieder erreicht
port  %1110      ; piepton
end              ; Ausweichbewegung beendet

```

Auch hier wieder eine kurze Erläuterung. Das Hindernis wird erkannt, der Roboter stößt zurück und versucht es zu umfahren. Das funktioniert natürlich nur, wenn das Hindernis sehr klein ist und die Leitlinie idealerweise geradeaus geht.

In allen anderen Fällen findet der Roboter die Leitlinie nicht wieder und bleibt entnervt stehen.

Soweit die kurze Einführung zur Roboterprogrammierung.

## 4. Anhang

Die Befehle bestehen jeweils aus einem Schlüsselwort und einer Datenangabe (außer end). Der Wertebereich der Daten richtet sich nach dem Befehl. Die Daten können als Dezimal- bzw. als Binärzahlen eingegeben werden. Binärwerten wird ein % vorangestellt.

Zur besseren Lesbarkeit ist bei den beiden Befehlen, die auf Register wirken (store, djnz) eine Adressierung der Register mit Buchstaben (A..D) möglich.

Sprungmarken werden vom Assembler in absolute Adressen umgerechnet.

<b>STEP</b>	<b>n</b>	Bewegung um n-Schritte; {0<n<4096} - Schritt = Teilung des Taktrades	2 Byte [0000nnnn] [nnnnnnnn]
step	10	; 10 Schritte mit der vorher eingestellten ; Geschwindigkeit	
<b>TURN</b>	<b>r</b>	Drehung in 22.5° - Schritten; {-7<r<8} - Vorzeichen bestimmt die Dreh- richtung	1Byte [0001rrrr]
turn	-2	; Drehung des Roboters um 45°Grad ; die Genauigkeit wird von der Taktscheibe bestimmt	
<b>SPEED</b>	<b>r</b>	Geschwindigkeit in 8 Stufen; {-7<r<8} - Vorzeichen definiert die Fahrtrichtung	1 Byte [0010rrrr]
speed	-1	; kleine Geschwindigkeit, rückwärts ; die Bewegung wird erst mit Eingabe eines step ; Befehles ausgeführt ; Defaultwert für speed ist 1	
<b>MTURN</b>	<b>r</b>	Drehung um r Takte; {-7<r<8} - Vorzeichen bestimmt die Drehrichtung	1 Byte [0011rrrr]
mturn	1	; "Mikrodrehung" des Roboters um einen Schritt ; wichtig für Suchprogramme	
<b>WAIT</b>	<b>r</b>	auf Tastenbetätigung warten r = 0 Roboter bleibt dabei aktiv r = 1 Power-Down-Mode bei WAIT	1 Byte [0100rrrr]
wait	1	; der Roboter wartet auf einen Tastendruck, der ; Power-Down-Mode ist aktiv	
<b>PAUSE</b>	<b>r</b>	Unterbrechung der Befehlsabarbeitung für 2 * r Sekunden; {1<r<16}	1 Byte [0101rrrr]
pause	16	; die Programmabarbeitung wird für 32 Sekunden ; unterbrochen	
<b>PORT</b>	<b>r</b>	Zugriff auf die Peripherie-Funktionen %xxx0 LED (rot) aus %xxx1 LED (rot) ein %xx1x Piepton (ca. 0,5 s) %x1xx IR-LED ein %x0xx IR-LED aus %1xxx Power Motoren ein %0xxx Power Motoren aus	1 Byte [0111rrrr]
port	%1100	; aktiviert die Motorstromversorgung und die IR-LED ; es müssen immer alle Bits des Befehls gesetzt werden, ; die Angabe %10 erzeugt einen Signalton, gleichzeitig ; wird jedoch auch die Stromversorgung der IR-LED und ; der Motoren abgeschaltet (der Assembler ersetzt die ; fehlenden Bits mit 0)	

```

JPAND d,adr Test des Statusbytes und Sprung          2 Byte
              d      Maske für Statusbyte          [1000dddd]
              adr    Adresse, MSB optionaler Sensor [d  adr  ]
jpan d, st    ; d wird als Maske auf den Inhalt des Statusbyte
              ; interpretiert, ein Sprung erfolgt wenn die logische
              ; Vernüpfung wahr wird
              ; Inhalt Status : %xxx00011
              ;           d :   %00011
              ;
              ;           ---> Sprung
              ;           d :   %00010
              ;
              ;           ---> kein Sprung

JPOR d,adr Test des Statusbytes und Sprung          2 Byte
              d      Maske                          [1001dddd]
              adr    Adresse, MSB optional          [d  adr  ]
jpor s, st    ; d wird als Maske auf den Inhalt des Statusbyte
              ; interpretiert, ein Sprung erfolgt wenn die logische
              ; Vernüpfung wahr wird
              ; Inhalt Status : %xxx00011
              ;           d :   %00010
              ;
              ;           ---> Sprung
              ;           d :   %10000
              ;
              ;           ---> kein Sprung

END         Beenden eines Prozesses                1 Byte
              [1010xxxx]
end           ; Verlassen eines aktivierten Prozesses

DJNZ d,adr Dekrementieren der Variablen, Sprung    2 Byte
              solange nicht 0                       [1011dddd]
              d = {A,B,C,D} 0...255                [  adr  ]
djnz a, st    ; Register A wird dekrementiert, Sprung solange A un-
              ; gleich 0 ist
              ; Verwendung dieses Befehles als Schleifenzähler
              ; der Befehl dekrementiert das Register und prüft erst
              ; dann den Inhalt

JP adr      unbedingter Sprung                    2 Byte
              [1100....]
              [  adr  ]
jp      st    ; unbedingter Sprung zur Marke st

STORE d     Setzen der Zählvariablen A,...,D      1 Byte
              d = 0..255                            [1101....]
store a,10    ; setzen der Schleifenvariable
              ; der Defaulwert ist 0, d.h. ein djnz ohne store erzeugt
              ; 255 Schleifendurchläufe

#DEFINE main Grundprozeß, nach dem Start immer aktiv

#DEFINE bump wird vom Prallkontakt gestartet

#DEFINE talk wird vom Akustiksensord gestartet

#DEFINE field wird vom optionalen Sensor gestartet, der Schwellwert
              dieses Sensors kann seperat eingestellt werden

#DATA Wert Wert zwischen 0...255, Schwellwert für optionalen Sensor

```

**Statusbyte** Dieses Byte wird von der Firmware des ST6 ständig aktualisiert  
 Der Inhalt dient zur Programmverzweigung bei JP-Befehlen  
 Es werden nur Bit0..4 bitweise verglichen

Bit 0 - LS1  
 Bit 1 - LS2  
 Bit 2 - Prallkontakt  
 Bit 3 - akustischer Sensor  
 Bit 4 - optionaler Sensor

Das assemblierte Roboterprogramm wird in den EEPROM des ST6 geladen. Der Speicher umfaßt 128 Byte. 6 Byte werden für die Adressen der Prozesse und das Schwellwertbyte reserviert. Die maximale Programmlänge beträgt demzufolge 122 Byte. Das sind ca. 80 Programmzeilen.

Es wird immer der gesamte EEPROM beschrieben. Der Hersteller garantiert 300000 Schreibzyklen.

Ein Hinweis noch zum Statusbyte. Die Bit's 0..4 repräsentieren immer den gegenwärtigen Zustand (Status) des Roboters. Die Meßwerte der Sensoren beeinflussen die ihnen zugeordneten Bit's. Über die Masken der bedingten Sprungbefehle ist damit eine Programmverzweigung möglich.

Um mehrfache Programmdurchläufe zu ermöglichen, sind vier Variable (A..D) definiert. Diese können mit dem Befehl djnz decremientiert und abgefragt werden.

Beispiel:

```
start:
    store A,25          ; Variable A setzen
    mturn 1            ; Mikrodrehung
    jband %00011,weit ; Ziel erkannt, weiter in Programm
    djnz A,start
weit:
    ... weiter im Programm
```

### Fehlermeldungen

Error1 Beep, 1 x Blinken  
 wird bei gesetztem Error-Bit in der proc-Abfrage angesprungen

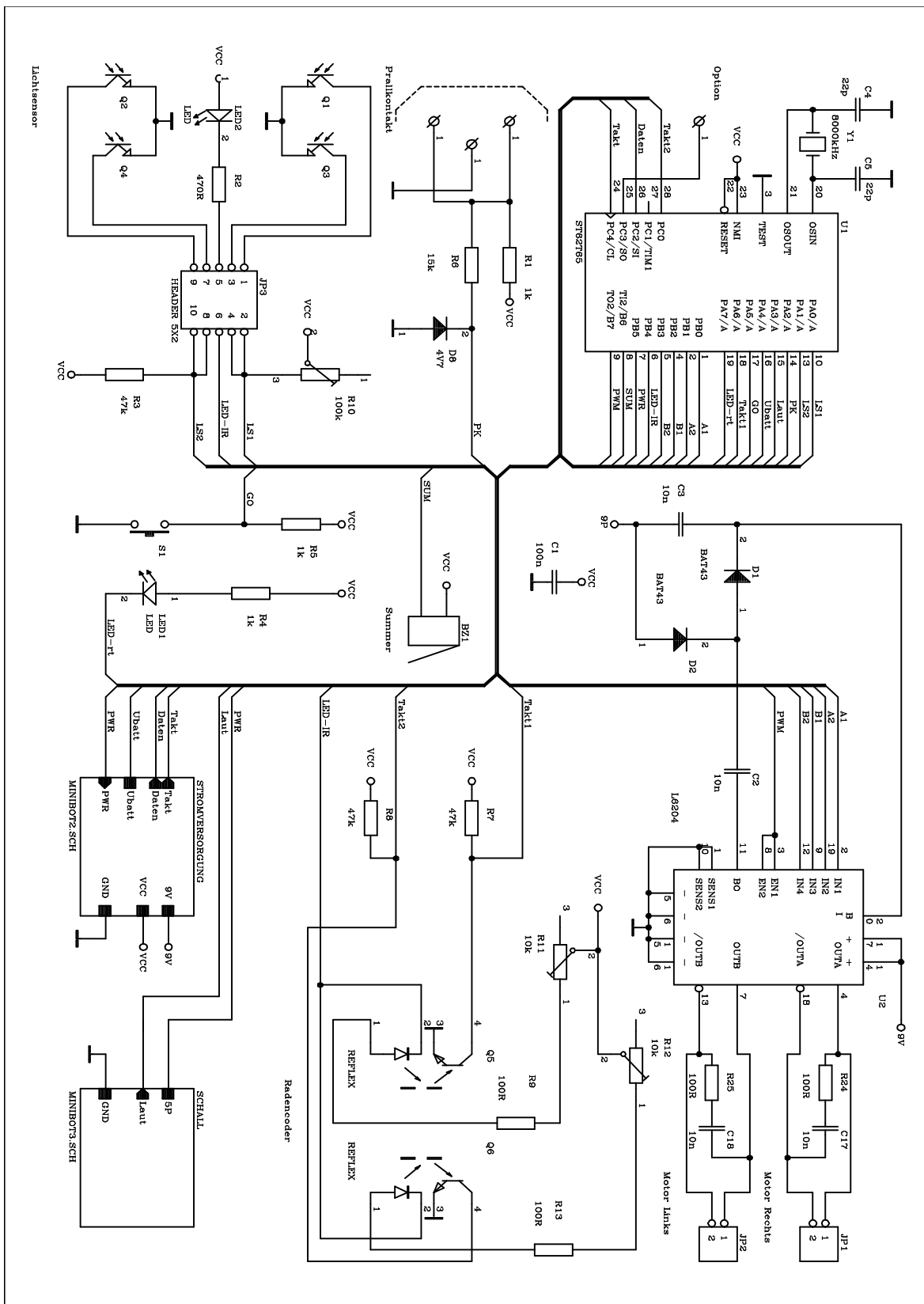
Error2 Beep, 2 x Blinken  
 wird bei fehlerhafter DFÜ angesprungen

Error3 Beep, 3 x Blinken  
 keinen gültigen Befehlscode gefunden

Error4 Beep, 4 x Blinken  
 Timeoutzeit bei der Motordrehung überschritten

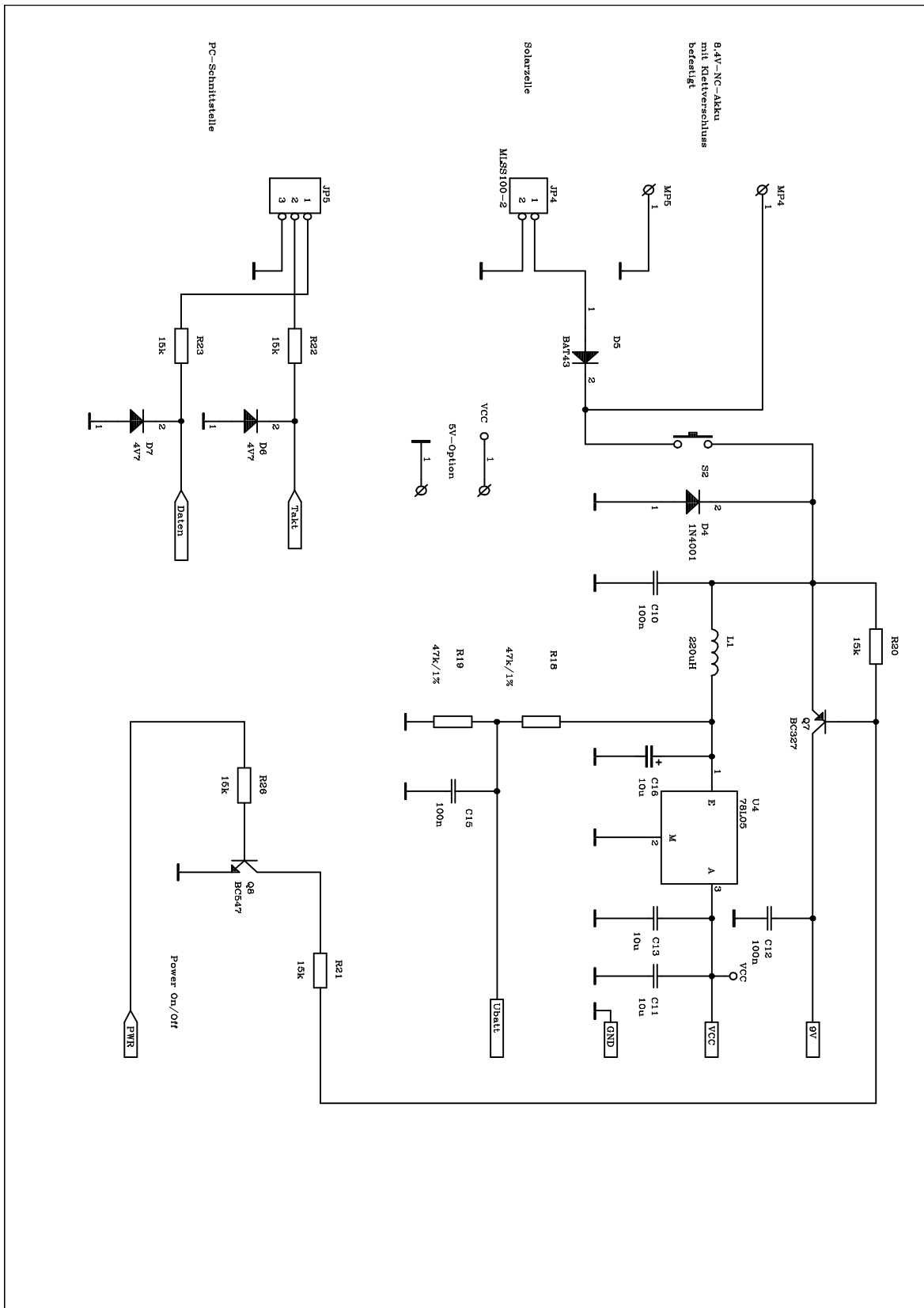
Error5 Beep, 5 x Blinken  
 Prozeßstack Overflow

# Stromlaufpläne Teil 1



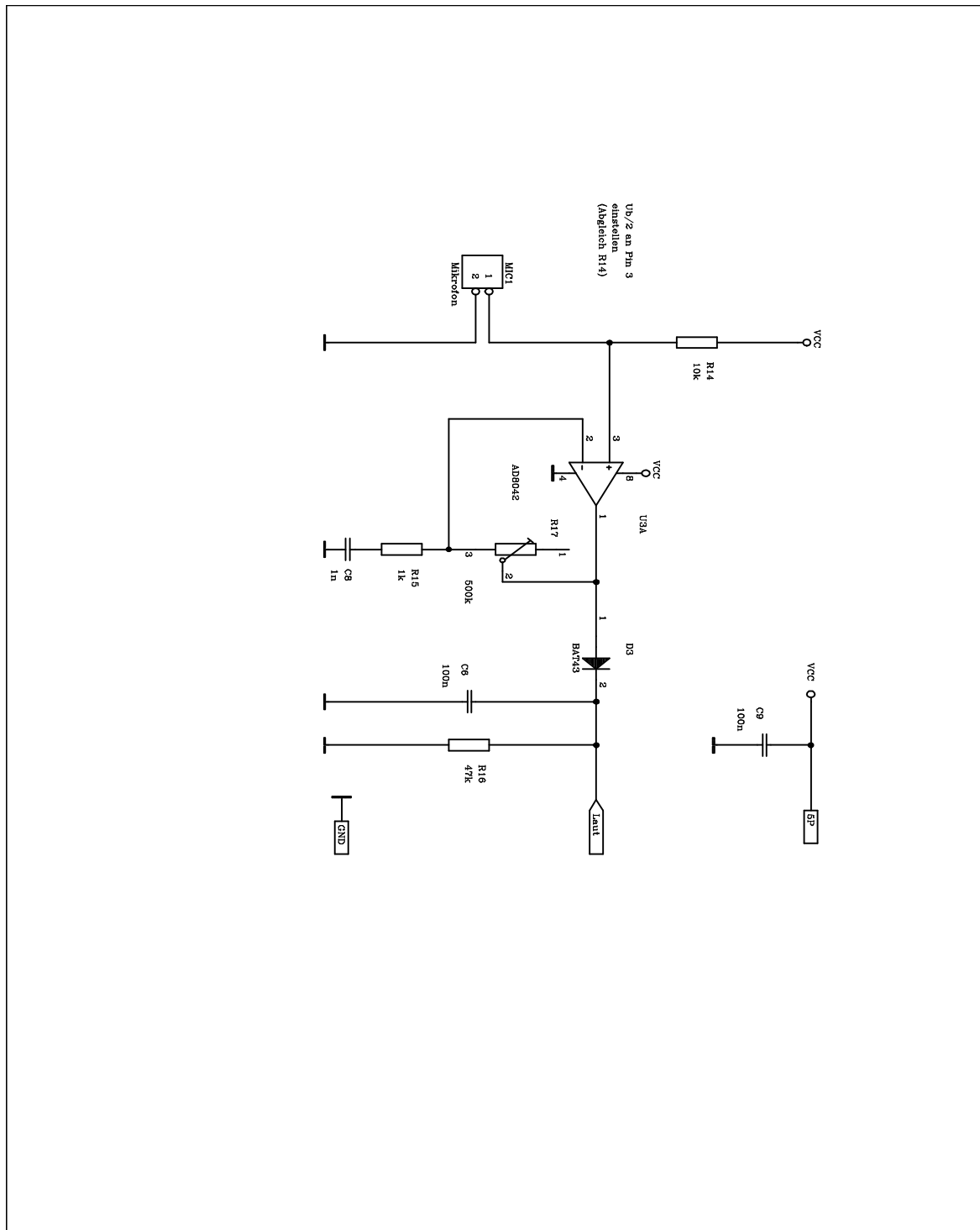
## Prozessoreinheit und Motortreiber

# Teil 2



Spannungsstabilisierung, Takt- und Dateneingang vom PC

### Teil 3



Akustischer Sensor

#### Wichtig:

Die Werte der Bauelemente in den Stromlaufplänen sind nur Richtwerte, verbindlich sind die Angaben in der Stückliste !

## Anschlußbelegung ausgewählter Bauelemente

### Kondensatoren

Bitte beachten Sie beim Einlöten der Bauelemente die Polarität der Kondensatoren. Im Zweifelsfalle ziehen Sie den Stromlaufplan zu Rate.

Bei den Kondensatoren sollte auf der Leiterseite ein + die Polariät kennzeichnen. Leider ist dies gründlich schief gegangen. Bei C16 ist das + außerhalb der Kontur gelandet, und bei C13 falsch. Die Lage der Kondensatoren bitte anhand des Stromlaufplanes (der an dieser Stelle richtig ist) festlegen.

### Fototransistoren und Leuchtdioden

Diese Bauelemente haben zwei unterschiedlich lange Anschlußbeinchen. Der längere Anschluß kommt in die Bohrung mit dem quadratischen Lötauge.

### Motoren

Damit der Drehsinn der Motoren mit den Programmanweisungen korreliert, kommt der schwarze Anschlußdraht in das quadratische Lötauge

### Programmierskabel

Das Kabel mit dem 3,5mm Klinkenstecker wird an die 9-polige Buchse angelötet.

Buchse	Kabel
Pin 4	weiße Leitung
Pin 5	Massegeflecht
Pin 7	rote Leitung

Alle anderen Pins bleiben unbenutzt.

### Haftungsausschluß

Dieses Handbuch wurde sorgfältig erstellt und geprüft. Trotzdem können Fehler und Irrtümer nicht ausgeschlossen werden. Es wird keinerlei Verantwortung für die uneingeschränkte Richtigkeit und Anwendbarkeit des Handbuches und des beschriebenen Produktes übernommen. Die Eignung des Produktes für einen spezifischen Verwendungszweck wird nicht zugesichert. Die Haftung des Herstellers ist in jedem Fall auf den Kaufpreis des Produktes beschränkt. Eine Haftung für eventuelle Mangelfolgeschäden wird ausgeschlossen.

Die in diesem Handbuch erwähnten Soft- und Hardwarebezeichnungen sind in den meisten Fällen auch eingetragene Warenzeichen und unterliegen als solche den gesetzlichen Bestimmungen. Es kann aus dem Fehlen einer besonderen Kennzeichnung nicht darauf geschlossen werden, daß die Bezeichnung ein freier Warenname ist.

# Stückliste

Position	Stück	Bezeichnung	Bemerkung
1	2	Motor mit Rad	scharzer Anschluß an quadratisches Pad
2	2	Motorträger	
3	1	Klettband f. Batt.	
4	1	Batterieclip	
5	1	Taktscheiben	
6	1	Diskette	
7	2	Abdeckfolie, rot	erst zum Schluß montieren
8	1	Cu-Sb-Draht, 130mm	für Stoßkontakt
9	3	Schraube M2x10	für Stoßkontakt
10	3	Mutter M2	für Stoßkontakt
11	4	Distanzbolzen	
12	4	Schraube M2,5x8	Befestigung Motorträger
13	8	Schraube M3x8	
14	1	Abdeckplatte	
15	1	Leiterplatte	
16	2	O-Ring	
17	8	U-Scheibe 3,2mm DIN 433	
18	5	Kondensator 100n	C1, C9, C10, C11, C15
19	3	Kondensator 10µ	C12, C13, C16
20	4	Kondensator 10n	C2, C3, C17, C18
21	2	Kondensator 22p	C4, C5
22	1	Kondensator 330n	C6
23	1	Kondensator 1n	C8
24	4	Diode 1N4148	D1, D2, D3, D5
25	1	Diode 1N4001	D4
26	3	Z-Diode 4V7	D6, D7, D8
27	1	Siftleiste 5x2	JP3
28	1	3,5mm Klinkenbuchse	JP5
29	1	Drossel 220µH	L1
30	1	Mikrofon	MIC1, Signalleitung an quadratisches Pad
31	1	Leuchtdiode	LED1
32	1	IR-Leuchtdiode	LED2
33	2	Fototrans. SFH309	Q1, Q2 (Q3, Q4)
34	2	Fototrans. LPT85A	Q5T, Q6T

35	2	IR-Diode IRL81A	Q5D, Q6D, "D" auf der Leiterseite
36	1	Transistor BC327	Q7
37	1	Transistor BC547	Q8
38	1	Taste	S2
39	1	Einschalter	S1
40	1	IC ST62T65	U1
41	1	IC L6204	U2
42	1	IC TL062	U3
43	1	IC 78L05	U4
44	1	Quarz 8000kHz	Y1
45	4	Widerstand 100R	R9,R13,R24,R25
46	1	Widerstand 470R	R2
47	4	Widerstand 1k	R1,R4,R5,R15
48	7	Widerstand 10k	R6,R14,R20,R21,R22,R23,R26
49	6	Widerstand 47k	R3,R7,R8,R16,R18,R19
50	2	Einstellregler 10k	R11,R12
51	1	Einstellregler 100k	R10
52	1	Einstellregler 1M	R17
53	1	Sub-D9 Gehäuse	für Programmierkabel
54	1	Sub-D9 Buchse	für Programmierkabel
55	1	Signalgeber	+ an die Seite von "BZ1"
56	3	Kodierbrücken	
57	1	Kabel mit Stecker	Programmierkabel
58	1	Kugelrad	Befestigung freisenken
59	1	Sockelstreifen	für Schaltkreise

---