

Laufroboter YETI

BAUANLEITUNG: Modell YT-3000



© AREXX - DIE NIEDERLANDE 2006

2.	YETI Allgemeine Info	4
3.	Software	11
4.	Wie laufen wir eigentlich?	17
5.	Hardware	19
6.	Aufbau Elektronik	22
7.	Teileliste Mechanik	30
8.	Bauanleitung für die Mechanikteile	33
9.	YETI Akkus aufladen	45
10.	Installation der Software	46
11.	Inbetriebnahme und Test	61
12.	Kalibrierung des YETIs	64
13.	YETI Programmieren	67
14.	Erweiterungen	83
xx.	APPENDIX	99
	A. Übersicht der YETI-funktionen	100
	B. Schaltbild YETI	104
	C. Schaltbild Display Module	105
	D. Schaltbild US Module	106
	E. Schaltbild RS232 IR	107
	F. Schaltbild USB IR	108
	G. Flachband Anschlußbelegung	109
	H. Fehlersuche	110
	I. Montage Erweiterungs kits	112
	J. Kalibrierungs und test programm	113

AREXX und YETI sind registrierte Warenzeichen von AREXX Engineering - HOLLAND.

© Deutsche Übersetzung/German translation (March 2006): AREXX Engineering (NL).

Diese Beschreibung ist urheberrechtlich geschützt. Der Inhalt darf auch nicht teilweise kopiert oder übernommen werden ohne schriftlicher Zustimmung des europäischen Importeurs:

AREXX Engineering - Zwolle (NL).

Hersteller und Vertreiber sind nicht haftbar oder verantwortlich für die Folgen unsachgemäßer Behandlung, Einbaufehler und oder Bedienung dieses Produkts bei Mißachtung der Bauanleitung.

Der Inhalt dieser Gebrauchsanleitung kann ohne vorheriger Ankündigung unsererseits geändert werden.



Fabrikant:
AREXX Engineering
JAMA Oriental



Europäischer Importeur:
AREXX Engineering
ZWOLLE Die Niederlande

**Technische Unterstützung beim Bauen
des Roboters:**

WWW.AREXX.COM
WWW.ROBOTERNETZ.DE

© AREXX Holland und JAMA Taiwan

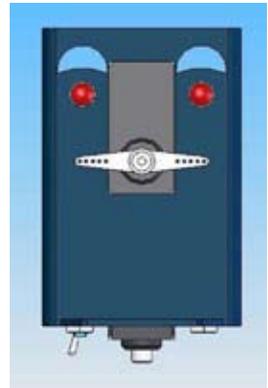
© Deutsche Übersetzung: AREXX - Die Niederlande

1. PRODUKTBESCHREIBUNG YETI

1.1. Zu welcher Roboterfamilie gehört YETI?

YETI ist ein gehender Roboter. Sein Name stammt vom “Schnee-Menschen”, einem vermeintlich im Himalaja lebenden, behaarten Riese. Genauso wie dieses legendäre Ungeheuer bewegt sich unser YETI auf zwei riesigen Füßen, aber darin erschöpft sich auch schon die Ähnlichkeit dieser Namensvetter.

Der Antrieb für die Beine und Füße unseres YETIs stammt von Servomotoren, von einem Mikroprozessor und wird gesteuert von einem Computerprogramm, das wir zuvor im Gehirn



1.2. Spezifikationen:

Motoren	2 Servomotoren (5 Volt)
Prozessortyp	ATmega8L
Programmiersprache	C
Spannung	4 St. AA Akku 4,8 - 6 Volt
Strom	Min. 10 mA Max. 600 mA
Kommunikation	Infrarot und I2C Bus
Erweiterung	2 Erweiterungen möglich mittels Flachbandkabel

2. YETI ALLGEMEINE INFO

2.1. Wer oder was ist ein YETI?

Wie bereits am Anfang beschrieben ist ein YETI ein "Schneemensch", ein vermeintlich im Himalaja lebender, behaarter Riese. Mit seinem Riesenkörper und seinen großen Füßen kann er sich nur ziemlich schwerfällig bewegen.

Unser YETI ist ein hochgewachsener Roboter, der ebenfalls auf großen Füßen läuft. Er kann vorwärts und rückwärts gehen und sich sogar links oder rechts herum drehen.

Bei jedem Schritt vorwärts oder rückwärts muß YETI sich zuerst auf einem Fuß abstützen und dann den anderen Fuß versetzen. Dazu benutzt er zwei Servomotoren. Ein Servomotor verwendet ein Getriebe und ist deshalb sehr kräftig. Außerdem enthält der Servomotor eine Impulsgesteuerte, elektronische Regelung. Die Elektronik erlaubt den Servo genau nur eine bestimmte Schwenkung durchzuführen.

Der YETI verwendet einen Servo an der Vorderseite und einen Servo an der Unterseite. Der Servo an der Vorderseite zieht die Füße hoch zum verschieben und wird Fußservo genannt. Der Servo an der Unterseite versetzt die Beine (und damit auch die Füße) eins nach dem anderen und wird Beinservo genannt.

2.2. Was können wir mit dem YETI anfangen?

- Neue (Beispiel-) Programme in den YETI übertragen.
- Selbsterstellte Programme in den YETI übertragen.
- YETI mit gebrauchsfertigen Erweiterungsmodulen erweitern, so dass der YETI zum Beispiel Gegenstände ausweichen oder Entfernungen messen kann.
- YETI mit selbstgebauten Erweiterungsmodulen erweitern.
- YETI mittels Infrarotsignalen mit dem PC kommunizieren lassen.
- YETI mittels Infrarotsignalen aus dem PC oder aus der Fernbedienung eines Fernsehers steuern.
- YETI zur Abgabe einer Tonfolge oder Geräuschfolge stimulieren.
- Seine "Augen"-LEDs ein- und ausschalten.
- Den Körper des YETIs erweitern, zum Beispiel um einen Display oder einem LED-Mund.

2.3. YETI wird mit drei Handgriffen zum Leben erweckt:

1. Setzen Sie zuerst die Mechanik- und Elektronikmodule des YETIs mit Hilfe der Bauanleitung zusammen.
2. Laden Sie ggf. die Akkus auf.
3. Schalten Sie den YETI mit dem Hauptschalter auf der Unterseite des Geräts ein.

Nach einigen Sekunden wird YETI seine Beine und Füße strecken und anschließend (mittels einem standard Beispielprogramm im Prozessor-hirn) ein Beispiel seines Könnens geben.

Nun, das war zunächst gar nicht schwierig, und es sieht aus, als ob man jetzt bereits fertig wäre.

Jetzt aber fängt die Arbeit erst an ! Wir werden uns jetzt mit dem Entwurf und Schreiben eigener Programme beschäftigen und können so in kreativer Weise das Äußere unseres YETIs anpassen.

2.4. Laden eines (Beispiel-) Programms in den YETI

Zum Laden eines beliebigen YETI-Programms vom Computer in den YETI verwenden wir unsichtbare Lichtsignale. Der im Lieferumfang enthaltene lose COM-Port-Adapterstecker ist ein RS232-Infrarot-Sender/Empfänger (Transceiver). Dieser wird an einem COM-Port des Computers angeschlossen. YETI verfügt in den beiden kleinen Öffnungen seines Rückens über einen eingebauten RS232-Infrarot-Sender/Empfänger. Der COM-Port-Adapterstecker ist auch als USB-Adapter verfügbar. Es handelt sich dabei um den gleichen IR-Transceiver, der auch für den ASURO benutzt wird. ASURO ist ein anderer, programmierbarer Roboter aus unserer Produktpalette.

Der Ladevorgang eines Programms in den YETI überschreibt automatisch das bereits zuvor vorhandene Programm. Das ursprünglich vorhandene, standard Beispielprogramm wird deshalb auch beim ersten Ladevorgang verschwinden. Das ist jedoch nicht schlimm, denn wir können dieses Beispielprogramm später immer wieder in YETI übertragen.

2.5. Laden eines Programms in den YETI

- Schließen Sie den COM-Port-Adapter, bzw. USB-Adapter auf den Computer an.
- Starten Sie bitte das Computerprogramm Flash.
- Selektieren Sie den zuvor angeschlossenen COM-Port in Flash.
- Selektieren Sie ein zu übertragendes YETI-Programm in Flash.
- Sehen Sie zu, dass die Öffnungen im YETI-Rücken auf der Oberseite des COM-Port-Adapters gerichtet sind.
- Schalten Sie bitte den YETI aus.
- Drücken Sie den Button „Programmieren“ im Flash-Programm.
- Schalten Sie bitte den YETI (innerhalb von 10 Sekunden) ein.
- Normalerweise wird das YETI-Programm nun in den YETI-Prozessor übertragen.
- Warten Sie bitte, bis der Ladevorgang des YETI-Programms abgeschlossen ist.
- Schalten Sie den YETI aus und wieder ein.
- Warten Sie bitte 3 Sekunden.

YETI wird nun sein neues Programm starten und ausführen.

2.6. Erweiterungsmodule (Kits)

Sie können YETI erweitern mit zusätzlichen (nicht im Lieferumfang enthaltenen) Baugruppen, welche die Leistungsfähigkeit des Roboters erheblich steigern. So können Sie YETI mit einem Ultraschall-Sender/Empfänger ausstatten, der ihm ermöglicht mit Schallwellenechos die Distanz zu entfernten Gegenstände zu messen und diesen auszuweichen.

Auch können Sie YETI mit einem Display ausstatten auf darin Daten oder Nachrichten anzeigen. Ein Erweiterungsmodul besteht aus einer kleinen Leiterplatte, die wahlweise mit oder ohne Bauelementen geliefert wird. Die Erweiterungsmodule passen genau in YETIs Kopf und werden sozusagen in seiner Schädeldecke genau oberhalb der Augen festgeschraubt.

Sie können aber auch Ihre eigenen Module mit Experimentierleiterplatten entwerfen. Diese werden auf YETIs Kopf anstelle des kleinen Vordachs platziert.

Mit einem Flachbandkabel verbinden Sie die Erweiterungsmodule über einer I2C Bus-Schnittstelle mit der Hauptplatine (und somit automatisch auch mit dem I2C Bus des Mikroprozessors im YETIs Kopf.

2.7. Der Kommunikationsprozess zwischen YETI und PC

Ein Druck auf die „Programmieren“-Taste im Flashprogramm aktiviert das Flashprogramm zehn Sekunden lang zur Kontaktaufnahme mit dem YETI. Falls die Kommunikation zustande kommt, wird das Programm in YETI übertragen. YETI wird sich nur innerhalb den ersten drei Sekunden nach dem Einschalten des YETIs beim Computer melden. Falls YETI innerhalb dieser drei Sekunden keinen Kontakt zum Computer herstellen kann, wird er einfach sein bereits geladenes Programm starten. Das Flash-programm wird bei fehlender Kontaktaufnahme nach 10 Sekunden eine Fehlermeldung absetzen.

Die Erfahrungen mit dem ASURO ergeben, dass der Daten-transfer manchmal etwas schwerfällig abläuft. Insbesondere bei Verwendung der RS-232 Adapterschnittstelle werden manchmal Fehlermeldungen erzeugt.

Man kann diese Fehlermeldungen mit folgenden Maßnahmen verhindern:

- Sorgen Sie für eine gute Sichtverbindung zwischen IR Sender und Empfänger.
- Verwenden Sie bitte die aktuellsten Flash Softwareversion.
- Verwenden Sie frische Batterien.
- Schalten Sie störende Lichtquellen (insbesonder Leuchtstoffröhren) ab. Verwenden Sie den USB-Adapter (insbesonder dann, wenn die RS-232 Spannung gering ist).

3. SOFTWARE

3.1. Schreiben Sie ein eigenes YETI-Programm

Für die Unerfahrenen, die mit dem Programmieren noch wenig Erfahrung haben, folgt hier eine kurze Einführung. Auch werden wir einige Hintergrunddetails zum Schreiben eigener Programme liefern.

Das Kapitel ist sicherlich nicht einfach und Sie werden einige neue Begriffe begegnen. Der Vorteil dieses Kapitels (der auch als kleiner Crashkurs betrachtet werden kann) ist auch ein besseres Verständnis für unser Vorgehen. Aber auch für Frage und Antwort oder für Diskussionen ist das Vorhandensein eines gemeinsamen Wortschatzes und gemeinsames Basiswissen vorteilhaft.

Sie können also den YETI mit einem selbstgeschriebenen Programm ausstatten!

Wie machen wir das?

- Sie schreiben ein Programm in einer Sprache, die C genannt wird (z.B. "test.c").
- Sie compilieren das Programm und erzeugen dabei eine Hex-Datei (z.B. "test.hex").
- Sie übertragen diese Hex-Datei in den YETI.

Der Ablauf besteht tatsächlich nur aus drei einfachen Schritten und ist für erfahrene Programmierer einfach. Den unerfahreneren Anfänger werden wir nun schrittweise voranbringen.

3.2. Stufe 1 Die Erstellung eines "C"-Programms.

Sie schreiben ein YETI-Programm in einer vorgegebenen Programmiersprache. Wir haben für diese Aufgabe die mächtige und gängige Programmiersprache mit dem ausgefallenen Namen "C" gewählt. Sie schreiben das YETI-Programm mit einem speziellen Textverarbeitung (einen Editor). Dieses Programm heißt "Programmers Notepad 2" (PN2).

Prinzipiell wäre es zwar möglich Programme mit einer Textverarbeitung wie "MS Word" zu verfassen, jedoch raten wir Ihnen dringend davon ab.

"C" Quelltexte sind reiner ASCII Text - ohne jegliche Formatierungen! Programmers Notepad ist dafür deutlich besser geeignet und vereinfacht die Programmerstellung immens!

Das PN2-Programm ist nämlich speziell auf die Programmierung in mehreren Sprachen zugeschnitten. Die Sprache, zum Beispiel "C", "Visual Basic" oder "HTML" wird in PN2 einmal definiert. Mit dieser Festlegung werden alle spezielle Sprachelemente des Programmtextes mit einer abweichenden Farbe markiert. Dieses Hilfsmittel bietet dem Programmierer eine hervorragende Unterstützung bei der Programmentwicklung.

Im PN2 können Sie zudem selbstdefinierte Aufträge hinzufügen. Das ist mit einem normalen Textverarbeitung nicht möglich. Unabhängig vom gewählten Textverarbeitung entsteht jedoch zunächst als erste Stufe eines selbstgeschriebenen Programms eine normale Textdatei.

3.3. Stufe 2 Der Compilervorgang bei einem “C”-Programm

Ein selbsterstelltes YETI-Programm, das in der erste Phase auch Programmquelle oder “Sourcecode” genannt wird, ist eine normale Textdatei, zum Beispiel die Datei “test.c”.

Ein sogenannter “Compiler”, in unserem Fall das Programm GCC.exe, transformiert die in der C-Sprache geschriebene Programmquelle “test.c” um in eine Objektdatei “test.o”. Eine Objektdatei enthält sogenannte Assemblerkommandos, die der YETI-Prozessor versteht und ausführen kann, aber in der Objektdatei noch als lesbarer Text vorliegen.

Eine “.o”-Objektdatei enthält somit Ihre eigene “C”-Programmquelle, angereichert mit Prozessorkommandos (für den speziellen Prozessor, den Sie verwenden werden).

Anschließend setzt ein sogenannter “Linker” diese Kommandos um in die für den Prozessor benötigten, ausführbaren sogenannten (HEXadezimalen) Kommandos und fügt diese Daten zusammen in eine Datei “test.hex”.

Theoretisch müßten Sie bei jeder Änderung eines YETI-Programms zuerst den Compiler und anschließend das Linkerprogramm starten. Wir können jedoch diese Prozedur mit einer Batch-Datei (.bat) und einer “makefile”-Datei automatisieren.

Eine Batch-Datei kann eine Kommandoreihe in einer Serie abarbeiten. In einer Makedatei legen Sie fest, wie und welche Dateien compiliert und gelinkt werden sollen. Siehe zum Beispiel die Information auf <http://www.gnu.org/software/make> und <http://www.gnu.org/software/make/manual>.

Diese Prozedur verwendet eine große Menge neuer Dateien und Dateinamen, ist kompliziert und vielleicht auch verwirrend. Die Beschreibung der genauen Prozedur ist jedoch nur interessant für diejenigen, die das Verfahren wirklich genau verstehen wollen. In der Praxis wird das Verfahren stark vereinfacht!

Wie vereinfachen wir die Prozedur Nun, wir starten eine kleine Batchdatei, die ihrerseits eine Makefile-Datei aufruft. Und wenn wir unserem Hauptprogramm PN2 mit dieser Kommandofolge bestücken, wird die ganze Arbeit automatisch erledigt!

Die Compiler- und Linkervorgänge werden jetzt gestartet mit einem Menubutton im PN2 oder mit einer bestimmten Tastenkombination im PN2. Einfacher geht es nicht mehr! Im Hintergrund führt PN2 die vielen kleinen Programmschritten in den Einzelprogrammen durch und liefert als Endergebnis eine HEX-Datei, die Sie in den YETI-Prozessor übertragen können.

PN2 wurde als Universalwerkzeug entworfen für eine Vielzahl an Compilern und Linkerprogramme. Dadurch ist die Schnittstelle zwischen PN2 und den Compilern beziehungsweise Linkern etwas komplexer geraten.

Hört sich das immer noch komplex an? Das ist es in gewisser Hinsicht auch. Aber um in "C" zu programmieren, braucht uns das Verfahren nicht zu belasten.

Diese Schnittstelle und Ankopplung der Compiler-/Linkersoftware wird stufenweise in diesem Handbuch beschrieben.

Warum haben wir genau diesen Compiler gewählt?

Nun, zunächst weil es sich um ein sehr professionelles und leistungsfähiges Werkzeug handelt und weiterhin, weil es sich dabei um Open Source Software handelt, die jeder frei benutzen kann ohne dafür erst teure Lizenzgebühren zahlen zu müssen!

Die "GNU" Organisation entwickelt den GCC neben vielen weiteren Unix Programmen ständig weiter. Mehr Informationen erhalten Sie unter <http://www.gnu.org/> bzw. <http://gcc.gnu.org/>

Der GCC ist also für Unix Betriebssysteme wie Linux entwickelt worden und die Installation wäre unter Windows normalerweise nicht ganz einfach. Glücklicherweise gibt es aber das Projekt WinAVR (wird wie das englische Wort "whenever" ausgesprochen), welches die Installation und die Mikrocontroller Programmentwicklung mit dem GCC unter Windows sehr stark erleichtert.

Im WinAVR Paket befindet sich unter anderem der speziell für AVR Mikrocontroller angepasste Compiler AVR-GCC, viele weitere Unix Tools wie z.B. 'make', ein Debugger, Simulator und vieles andere - allesamt fertig für die Verwendung unter Windows eingerichtet! Weiterhin wird ein komfortabler Quelltext Editor 'Programmers Notepad 2' (PN2) mitgeliefert, welchen wir auch für die Programmentwicklung für YETI empfehlen!

Neben der WinAVR Version 20060421 auf der CD finden Sie stets eine aktuellste Version von WinAVR unter <http://winavr.sourceforge.net/> zum Download!

Bei der Installation von WinAVR werden hunderte von Dateien in den WinAVR Ordner kopiert, allerdings benötigen wir für unsere Zwecke nur sehr wenige davon.

Sie müssen also nicht alle Tools und Dateien von WinAVR verstehen um YETI zu programmieren!

3.4. Stufe 3 Der Transfer eines YETI-Programms

Im DLR (Deutsches Zentrum für Luft- und Raumfahrt) wurde ein spezielles Werkzeug entwickelt um das YETI-Programm mit dem beiliegenden COM-Port oder mit dem USB-Adapter in den YETI zu übertragen. Dieses Werkzeug mit dem Namen Flash transferiert die vom Benutzer selektierte Hex-Datei, zum Beispiel "test.hex", in den YETI Mikroprozessor.

Im YETI befindet sich eine kleine, fest eingebaute Kommunikationssoftware, die "Bootloader" genannt wird. Dieses Mini-Programm kann vom Benutzer nicht geändert oder entfernt werden und gehört im YETI mehr oder weniger zur Hardware.

Nach dem Einschalten des YETIs wird immer zuerst der "Bootloader" gestartet. Dieses fest eingebaute Kommunikationsprogramm horcht drei Sekunden lang, ob irgendwelche Infrarotsignale aus der Flash-Software im Computer empfangen werden. Wenn ja, dann registriert YETI diese Daten und schreibt das empfangene Programm in seinen Programmspeicher.

Wenn nein, dann startet YETI anschließend das in seinem Programmspeicher vorhandene Programm.

YETI verfügt über eine besondere Art Speicher, der auch nach dem Ausschalten des Roboters erhalten bleibt. Dieser Speichertyp heißt "Flash"-Speicher. Sicherlich werden Sie jetzt auch den Grund für den Namen "Flash"-Programm verstehen.

4. WIE LAUFEN WIR EIGENTLICH?

4.1. Zusammenfassung

YETI balanciert auf einem Fuß und versetzt anschließend den anderen Fuß. Dazu zieht YETI die Außenseite seines abstützenden Fußes hoch und drückt zur gleichen Zeit die Außenseite seines zu versetzenden Fußes herunter. Dabei neigt sich YETI zum abstützenden Fuß herüber und verlagert den Großteil seines Gewichts auf diesen abstützenden Fuß. Anschließend schiebt YETI seinen zu versetzenden Fuß nach vorne und schließt somit den Schritt ab. Der abstützende Fuß wird nun der zu versetzende Fuß und umgekehrt.

Dieser Bewegungsablauf muß sich beim Gehen natürlich ständig wiederholen.

4.2. Ausführliche Erläuterung

Sobald der YETI aus der Stillstandposition laufen soll, dreht er seinen Fußservo von vorne betrachtet im Uhrzeigersinn. Dabei passieren zwei Dinge gleichzeitig.

Zum einen hebt sich die rechte Seite des Fußservos und zieht die Außenseite des rechten Fußes ein wenig hoch. Er könnte seinen Körper nun rechtswärts bewegen, tut das aber nicht, weil ein Großteil des Gewichts sich immer noch auf der Innenseite des rechten Beines befindet.

Zweitens senkt sich die linke Seite des Fußservos und drückt gleichzeitig die Außenseite des linken Fußes ein wenig herunter. Dabei wird das linke Bein hochgedrückt. Beide Bewegungen zusammen bewegen den Körper nach rechts und verlagern das Gewicht von den linken auf den rechten Fuß.

Falls YETI seinen rechten Fuß etwas zu hoch anheben würde, drückt der linke Fuß den Körper des YETIs über die Gleichgewichtsposition des rechten Fußes hinweg. Der Roboter verliert dann sein Gleichgewicht und würde nach Rechts umkippen. YETI soll deshalb seinen rechten Fuß gerade soweit hochheben, dass er, wenn sein linkes Bein ihn nach Rechts drückt, gerade nicht sein rechter Gleichgewichtspunkt erreicht. Je dichter der Körper den Gleichgewichtspunkt annähert, desto größer ist der Anteil des Gesamtgewichts auf dem rechten Fuß und desto leichter läßt sich der linke Fuß versetzen.

Jetzt müssen noch die Beinservos sich bewegen. Mit dem Beinservo versetzt YETI seinen linken Fuß möglichst weit nach vorne. Beim Versetzen des linken Fußes von hinten nach vorne bewegt sich auch der Körper bis zur halben Strecke des linken Fußes, aber immer noch in der Spur des rechten Fußes. Er stützt sich also immer noch überwiegend auf dem rechten Fuß. Wenn nun der linke Fuß sich weit nach vorne gestreckt hat, zieht er die Außenseite seines linken Fußes hoch und die Außenseite seines rechten Fußes herunter. Dadurch verschiebt er seinen Körper, sein Körpergewicht und damit auch den Schwerpunkt des Körpers vom rechten Fuß zum linken Fuß. Ab diesem Zeitpunkt ruht das Gewicht hauptsächlich auf den linken Fuß. Auch hier gilt die Regel, daß YETI nicht zu weit nach links balancieren darf, damit er nicht nach links umkippt.

Beim nächsten Schritt wiederholt sich die Prozedur mit dem gegenüberliegenden Fuß.

5. HARDWARE

5.1. YETI Basis Leiterplatte

Die Hauptleiterplatte des YETIs enthält einen Atmega8L Mikrocontroller Chip. Dieser Mikrocontroller ist verbunden mit dem großen, runden Piepser und mit den beiden roten LEDs an der Vorderseite des Roboters. Zahlreiche andere Mikrocontrolleranschlüsse, z. B. der I2C Bus, werden direkt zum 20-Pins Steckeranschluß auf der Rückseite weitergeleitet. Dieser Steckverbinder erlaubt eine Systemerweiterung mit weiterer Hardware über Flachbandkabel. Die beiden schwarzen Steckverbinder neben der roten LED versorgen den Anschluss der beiden Servos. Die drei weißen Steckverbinder versorgen den Ein/Aus-Schalter und die beiden Akkupacks.



5.2. Flachbandanschluss

Vier der zwanzig Flachbandadern sind für die Stromversorgung reserviert. Pin 19 für VCC und Pin 7, 8 und 20 für GND.

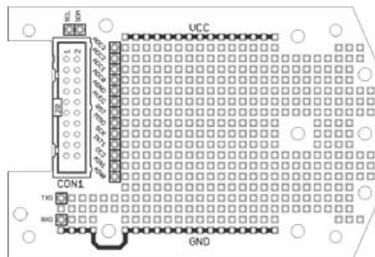
Alle übrigen Flachbandanschlüsse sind direkt mit einem festgelegten Anschluß des Mikrocontrollers verbunden. Direkt neben den Anschlußnummern sind die Mikrocontrollerpinfunktionen spezifiziert. Ein Mikrocontrollerpin kann je nach Anwenderprogramm auf verschiedenster Art verwendet werden. Überprüfen Sie bitte im Datenblatt bzw. Handbuch zum Mikrocontroller die genaue Definition der Pinfunktion.

5.3. Erweiterungsanschlussbelegung

Alle Erweiterungsmodule werden grundsätzlich mit dem gleichen 20-adrigen Flachbandkabel auf der YETI-Hauptplatine angeschlossen. Das Flachbandkabel enthält auch die Stromversorgungsanschlüsse der Erweiterungsmodule und den I2C Bus von und zu den Erweiterungsmodulen.

Info Flachbandanschluss: (Mehr Info sehe Seite 20)

Pin 1	SCL
Pin 2	SDA
Pin 3	PC3(ADC3)
Pin 4	PC2(ADC2)
Pin 5	PC1(ADC1)
Pin 6	PC0(ADC0)
Pin 7	GND
Pin 8	GND
Pin 9	AVCC
Pin 10	PC6(RESET)
Pin 11	PB5(SCK)
Pin 12	PB4(MISO)
Pin 13	PB3(MOSI/OC2)
Pin 14	PD3(INT1)
Pin 15	PD6(AIN0)
Pin 16	D7(AIN1)
Pin 17	PD0(RXD)
Pin 18	PD1(TXD)
Pin 19	VCC
Pin 20	GND



5.4. YETI Experimentiermodul

Im Experimentiermodul können Sie einen eigenen Entwurf einer Elektronikbaugruppe aufbauen und dieses Modul auf unterschiedlichster Art aus dem Mikrocontroller ansteuern.

5.5. Flachbandanschlussbelegung

Vier der zwanzig Flachbandadern sind für die Stromversorgung reserviert. Pin 19 für VCC und Pin 7, 8 und 20 für GND.

Alle übrigen Flachbandanschlüsse sind direkt mit einem festgelegten Anschluß des Mikrocontrollers verbunden.

Direkt neben den Anschlußnummern sind die Mikrocontroller-pinfunktionen spezifiziert. Ein Mikrocontrollerpin kann je nach Anwenderprogramm auf verschiedenster Art verwendet werden. Überprüfen Sie bitte im Datenblatt bzw. Handbuch zum Mikrocontroller die genaue Definition der Pinfunktion.

Pin 1	SCL	Serial Clock (für die I2C Datenkommunikation)
Pin 2	SDA	Serial Data (für die I2C Datenkommunikation)
Pin 3	PC3(ADC3)	Digitaler Eingang/Ausgang oder analoger Messeingang
Pin 4	PC2(ADC2)	Digitaler Eingang/Ausgang oder analoger Messeingang
Pin 5	PC1(ADC1)	Digitaler Eingang/Ausgang oder analoger Messeingang
Pin 6	PC0(ADC0)	Digitaler Eingang/Ausgang oder analoger Messeingang
Pin 7	GND	GND (Mehrere Anschlüsse zur Entstörung vorgesehen)
Pin 8	GND	GND (Mehrere Anschlüsse zur Entstörung vorgesehen)
Pin 9	AVCC	Analoge Referenzspannung für die AD-Konverter
Pin 10	PC6(RESET)	Mikrocontroller Reset Anschluss
Pin 11	PB5(SCK)	Digitaler Eingang/Ausgang
Pin 12	PB4(MISO)	Digitaler Eingang/Ausgang oder I2C Funktion Pin
Pin 13	PB3(MOSI/OC2)	Digitaler Eingang/Ausgang oder I2C function pin oder Timer2 Pin
Pin 14	PD3(INT1)	Digitaler Eingang/Ausgang oder externer Interrupt
Pin 15	PD6(AIN0)	Digitaler Eingang/Ausgang oder analoger Testeingang
Pin 16	D7(AIN1)	Digitaler Eingang/Ausgang oder analoger Testeingang
Pin 17	PD0(RXD)	Digitaler Eingang/Ausgang oder RS232 Eingang
Pin 18	PD1(TXD)	Digitaler Eingang/Ausgang oder RS232 Ausgang
Pin 19	VCC	VCC
Pin 20	GND	GND (Mehrere Anschlüsse zur Entstörung vorgesehen)

6. AUFBAU ELEKTRONIK

Überprüfen Sie zunächst anhand der Teilleiste der Elektronik-Komponenten, ob alle Bauteile vorhanden sind.

6.1. Die Lötarbeit

Die Beschriftung der Leiterplatte zeigt genau, wo Sie die Bauteile bestücken müssen. Sollten Sie sich die Position genauer ansehen wollen, steht Ihnen ein Bild der Bestückplan zur Verfügung.

Beim Bestücken einer Leiterplatte beginnen wir vorzugsweise mit den niedrigen Bauteilen. Das sind normalerweise die Widerstände. Schneiden Sie die Drahtenden kurz nach dem Einlöten ab, so daß Sie immer genug Platz zum Arbeiten haben.

Ehe Sie mit dem Löten beginnen, sollten Sie die integrierten Schaltungen probeweise kurz auf der Leiterplatte einsetzen und notfalls die Beinchen genau ausrichten mit einer Flachzange. Meistens sind die Beinchen etwas zu weit angewinkelt. Zum Schluß montieren Sie das IC in den IC-Sockel.

WICHTIG

Der Elko und das IC müssen mit korrekter Orientierung eingelötet werden!

ACHTUNG

Sie sollten die integrierten Schaltungen (IC) jetzt noch nicht einlöten!!

TIP

Die IC-Beinchen können Sie einfach auf einer harten Tischfläche ausrichten! Legen Sie dazu die Beinchen auf die Tischfläche und drücken Sie diese vorsichtig in die richtige Position.

Bei technischen Fragen oder Problemen können Sie in folgenden Foren um Hilfe fragen:

www.arexx.com --> Forum

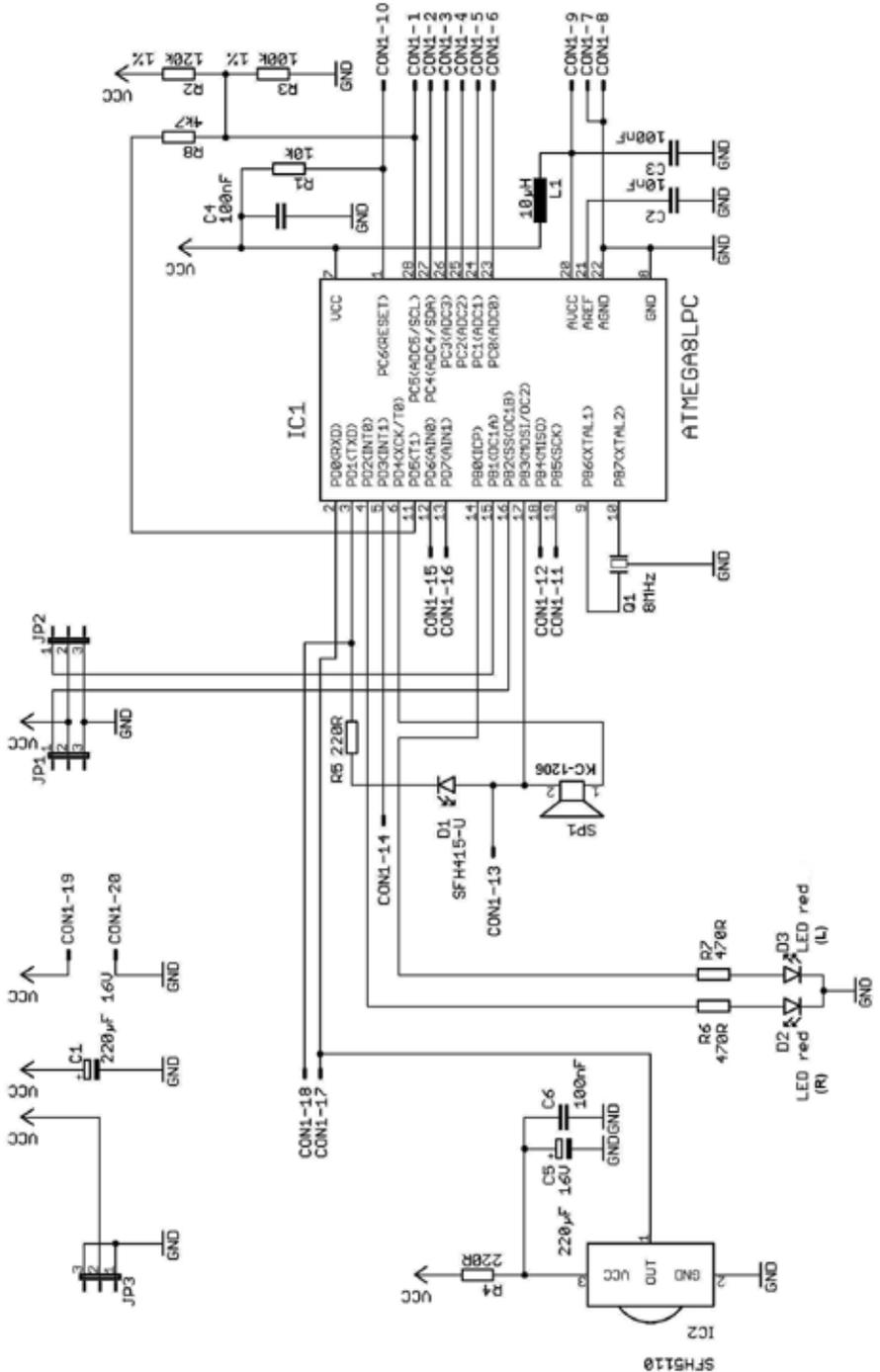
www.roboternetz.de --> Forum

6.2. Bestückung Hauptplatine

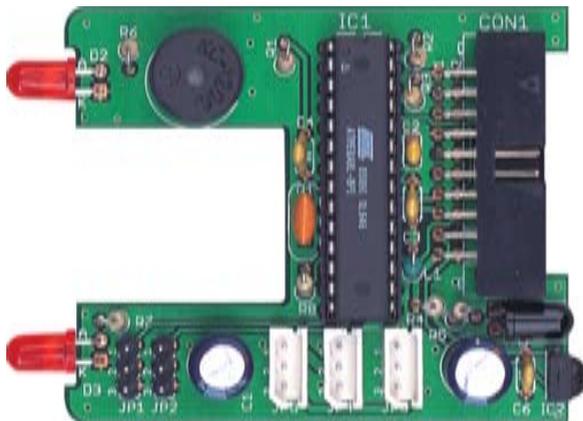
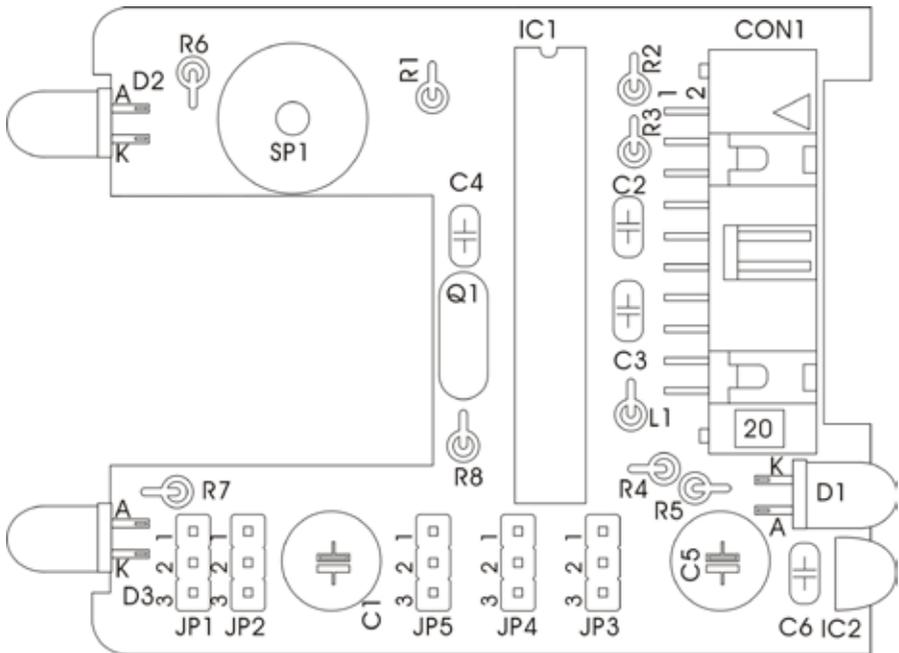
YETI Teileliste

Nr.	Name	St.
PCB1	Platine	1
IC1	ATmega8-I (auf richtige Polung achten)	1
IC2	SFH5110 IR-Empfänger-IC (auf richtige Polung achten)	1
R1	10K / 0.25W / 5% (braun, schwarz, orange, gold)	1
R2	120K / 0.25W / 1% (braun, rot, schwarz, orange, braun)	1
R3	100K / 0.25W / 1% (braun, schwarz, schwarz, orange, braun)	1
R4	220R / 0.25W / 5% (rot, rot, braun, gold)	1
R5	220R / 0.25W / 5% (rot, rot, braun, gold)	1
R6	470R / 0.25W / 5% (gelb, violett, braun, gold)	1
R7	470R / 0.25W / 5% (gelb, violett, braun, gold)	1
R8	4K7 / 0.25W / 5% (gelb, violett, rot, gold)	1
L1	10uH (braun, schwarz, schwarz, silber)	1
C1	220uF/16V (auf richtige Polung achten)	1
C2	10nF Aufdruck: 103	1
C3	100nF Aufdruck: 104	1
C4	100nF Aufdruck: 104	1
C5	220uF/16V (auf richtige Polung achten)	1
C6	100nF Aufdruck: 104	1
D1	SFH415-U IR-LED (auf richtige Polung achten)	1
D2	LED Rot, 5 mm (auf richtige Polung achten)	1
D3	LED Rot, 5 mm (auf richtige Polung achten)	1
Q1	Quarz, 8Mhz / 3 PIN	1
SP1	Piepser, 5V, (KC1206) (auf richtige Polung achten)	1
IC socket	28 PIN, IC Fuß (auf richtige Polung achten)	1
JP1	3 PIN, Platinensteckverbinder, schwarz	1
JP2	3 PIN, Platinensteckverbinder, schwarz	1
JP3	3 PIN, Platinensteckverbinder, weiß	1
JP4	3 PIN, Platinensteckverbinder, weiß	1
JP5	3 PIN, Platinensteckverbinder, weiß	1
CON1_PCB	Konnektor, male, 20 pins, für Flachband/ 90° gewinkelt	1
Battery-Hal	Batteriehalter mit Konnektor	2
Klett-Band	(male)	2
Klett-Band	(female)	2
Servo	Servo Motor	2

6.3. Schaltbild YETI



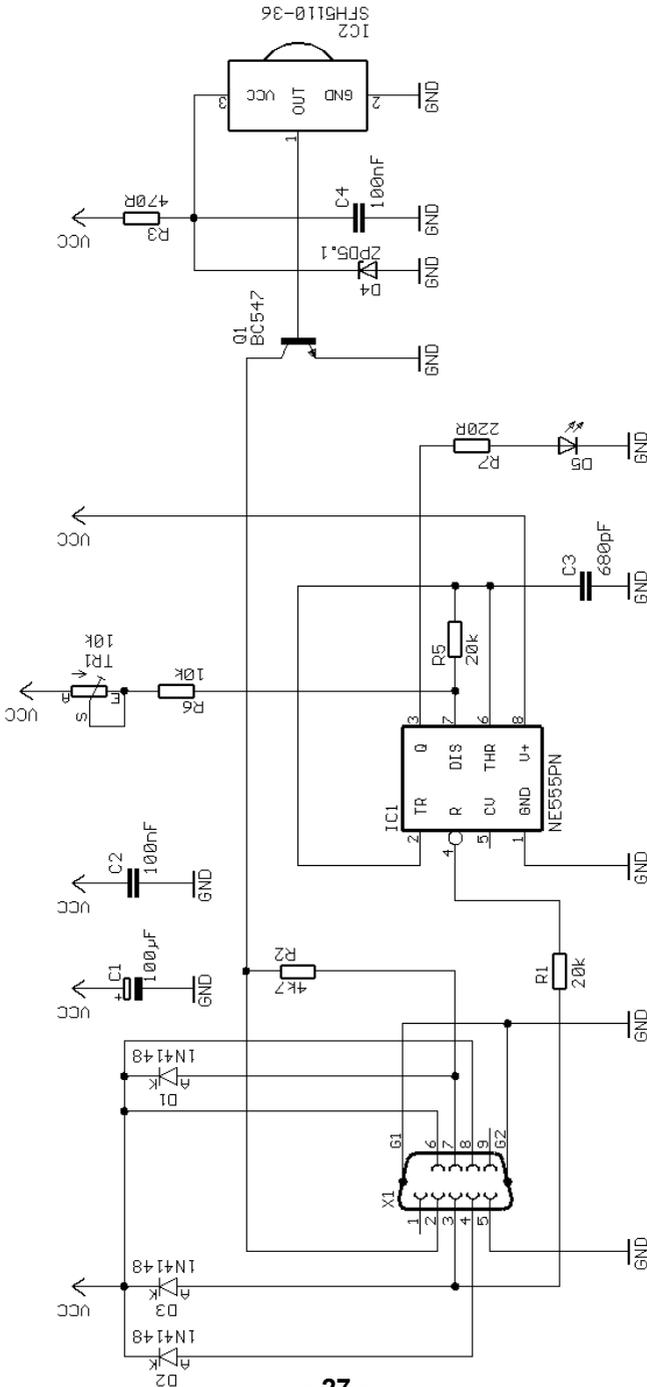
6.4. Hauptplatine



6.5. Bestückung des RS232-Infrarot-Transceivers

- **IC1:** Hier wird zunächst nur der 8-polige Sockel eingelötet. Dieser besitzt eine Richtungsmarkierung, welche mit der Markierung auf der Platine übereinstimmen muss.
- **D1, D2, D3:** 1N4148, auf richtige Polung achten!
Nicht verwechseln mit ZPD5.1 oder BZX55-C5V1 (Aufdruck)!
- **D4:** ZPD5.1 oder BZX55-C5V1, auf richtige Polung achten!
Nicht verwechseln mit 1N4148 (Aufdruck)!
- **C2, C4:** 100nF keramisch, Aufdruck: 104
- **C3:** 680pF keramisch, Aufdruck: 681
- **Q1:** BC547 (A,B oder C) oder BC548 (A,B oder C)
- **R1, R5:** 20k Ohm, 5% (rot, schwarz, orange, gold)
- **R2:** 4.7k Ohm, 5% (gelb, violett, rot, gold)
- **R3:** 470 Ohm, 5% (gelb, violett, braun, gold)
- **R6:** 10k Ohm, 5% (braun, schwarz, orange, gold)
- **R7:** 220 Ohm (rot, rot, braun, gold)
- **C1:** 100µF/ mindestens 16V, auf richtige Polung achten!
- **TR1:** 10k Ohm Trimmer
- **D5:** SFH 415-U IR-LED (schwarzes Gehäuse), auf richtige Polung achten!
Gehäuse sollte auf der Platine aufliegen!
- **IC2:** SFH5110-36 Infrarot-Empfänger-IC, Beinchen mit Zange abwinkeln!
Auf richtige Polung achten (Seite mit Wölbung muss nach oben weisen), Achtung:
elektrostatisch gefährdet und - Hinweis für die Hobbyschweißer - hitzeempfindlich!
- **X1:** 9pol. SUB-D Buchse, Gehäuse muss auf der Platine aufliegen, auch die Befestigungsglaschen müssen angelötet werden!
- **IC1:** NE555P einstecken, Richtungsmarkierung (Nase oder Kreis) beachten!

6.6. Schaltbild RS-232 IR-Transceiver



6.7. Bestückung des RS232-Infrarot-Transceivers

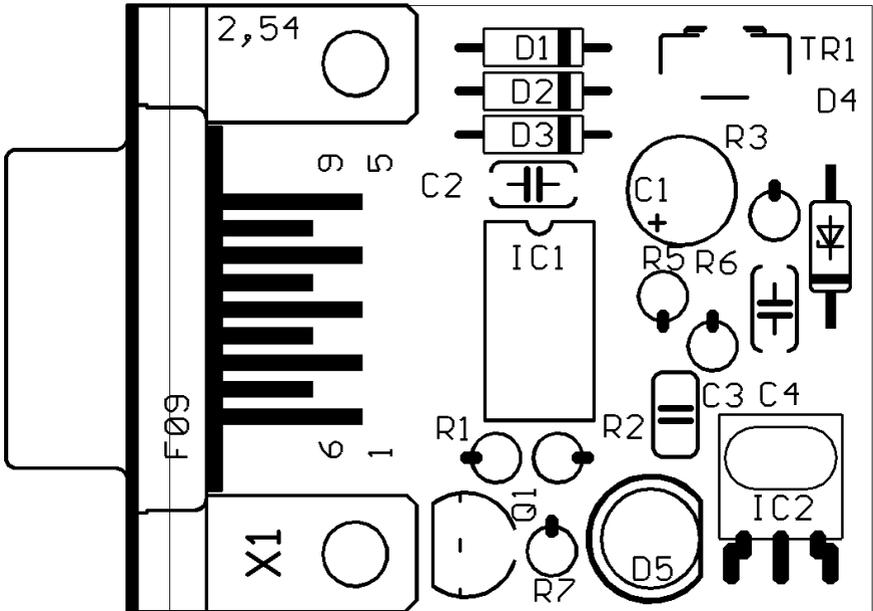
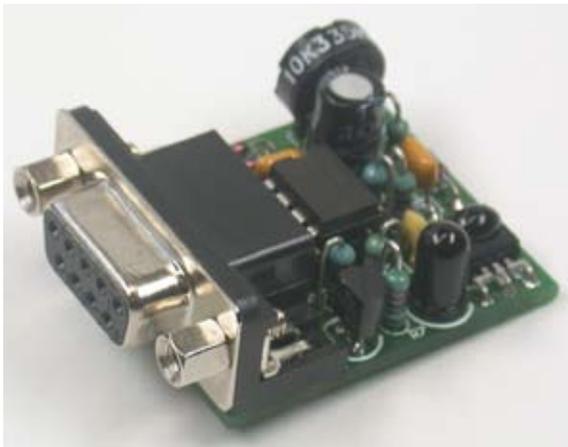


Abbildung 6.1.: Bestückung des RS232- Infrarot-Transceivers



Nun nochmal mit kritischem Blick die Lötstellen auf gute Verbindung oder Kurzschlüsse überprüfen und ggf. nachbessern.

6.8. Info Fertiggerät USB-Infrarot-Transceiver

Optional ist ein USB-IR-Transceiver als Fertiggerät lieferbar.

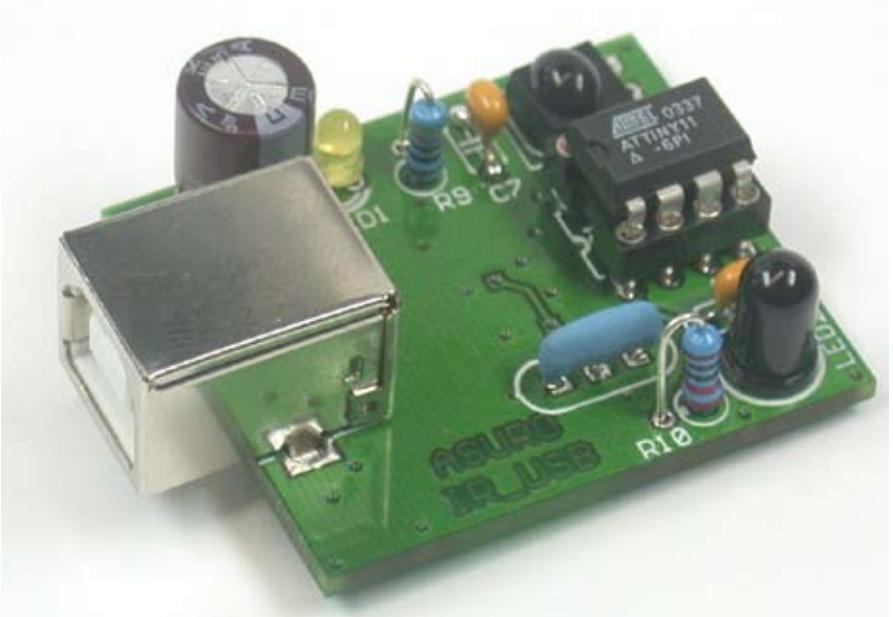


Abbildung 6.2.: USB Infrarot-Transceiver

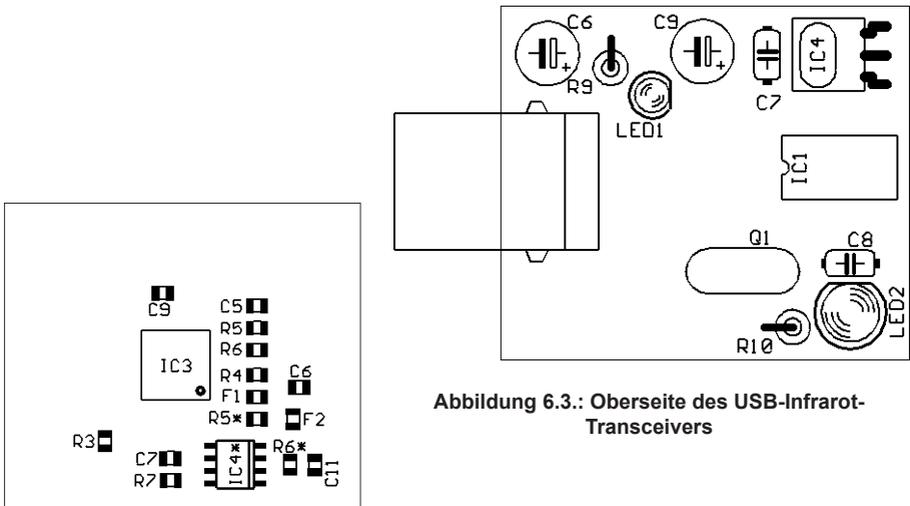


Abbildung 6.3.: Oberseite des USB-Infrarot-Transceivers

Abbildung 6.4.: Unterseite des USB-Infrarot-Transceivers

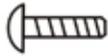
7. TEILELISTE MECHANIK

Mutter M3



O 8 St.

Linsenkopfschraube
M3x8mm



O 8 St.

Stellring
Groß



O 4 St.

Stellring
Klein



O 4 St.

Maden-
Schraube



O 10 St.

Mutter M2



O 10 St.

Buchse



O 4 St.

Niete



O 4 St.

Gestänge-
anschluss



O 2 St.

Servo
Schraube



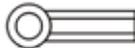
O 2 St.

Kugelpopf-
schraube



O 4 St.

Gelenkkopf mit
Innengewinde



O 4 St.

Kugelpopf-
mutter



O 4 St.

Sechskant-Winkel-
schraubendreher



O 1 St.

Drahtachse lang



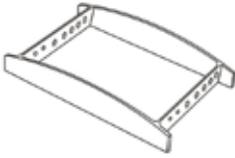
O 2 St.

Drahtachse kurz



O 4 St.

Fuß



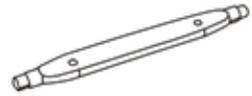
O 2 St.

Fußachse
5x80mm



O 2 St.

Fuß Kuppelstange



O 2 St.

Vorderbein



O 2 St.

Hinterbein



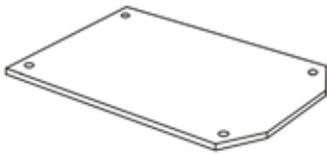
O 2 St.

Stelldraht



O 2 St.

Obere Abdeckung (meist oben)

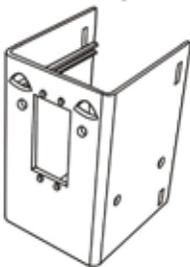


O 1 St.

Obere Abdeckung (unteres Teil)

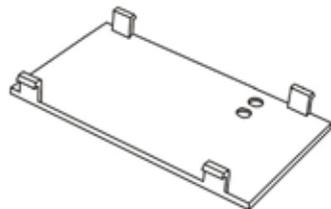
O 1 St.

Kopfteil

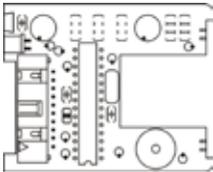
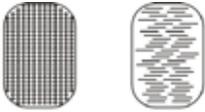
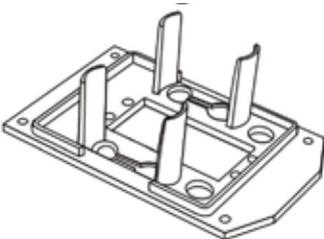
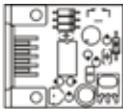


O 1 St.

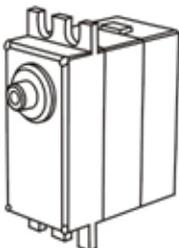
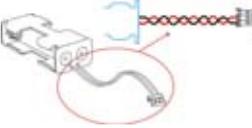
Hintere Abdeckung



O 1 St.

<p>YETI Hauptplatine</p>  <p>O 1 St.</p>	<p>Klett-Band</p>  <p>O 2 St. Male O 2 St. Female</p>	<p>Bodenteil</p>  <p>O 1 St.</p>
<p>IR/Transceiverplatine</p>  <p>O 1 St.</p>	<p>RS-232 Kabel</p>  <p>O 1 St.</p>	

7.1. Wichtige Elektronikteile

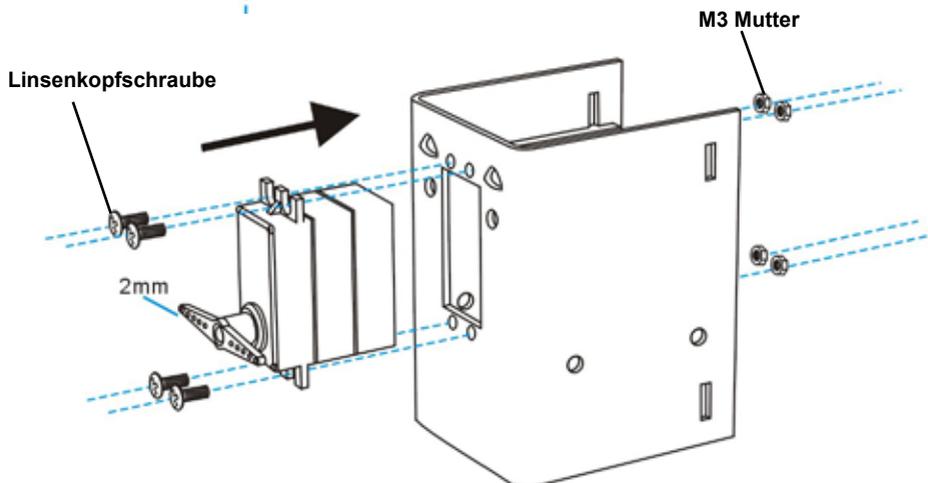
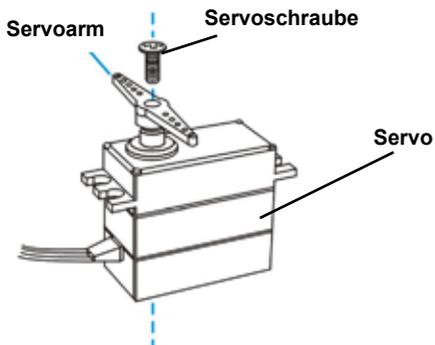
<p>Servo Motor</p>  <p>O 2 St.</p>	<p>Servo Achse</p>  <p>O 2 St.</p>	<p>Schalter</p>  <p>O 1 St.</p>	<p>Batteriehalter</p>  <p>O 2 St.</p>
<p>Flachbandkabel</p>  <p>O 1 St.</p>	<p>Kabelsatz</p>  <p>O 1 St.</p>	<p>DC-Buchse</p>  <p>O 1 St.</p>	

8. Bauanleitung für die Mechanikteile

Montage des Kopfservos:

Zur Montage des Vorderservos wird folgendes benötigt;

- 1 St. Kopfteil
- 1 St. Servo
- 4 St. Linsenkopfschraube
- 4 St. Mutter M3
- 1 St. Servoarm
- 1 St. Servoschraube



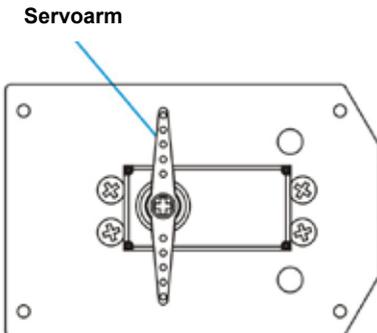
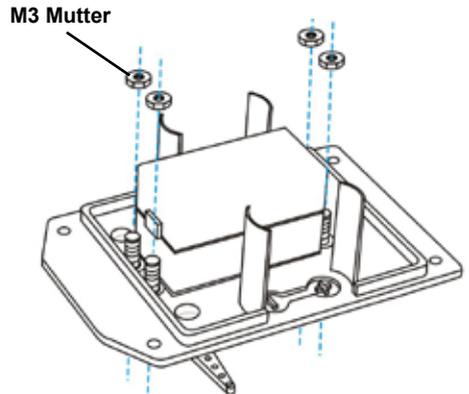
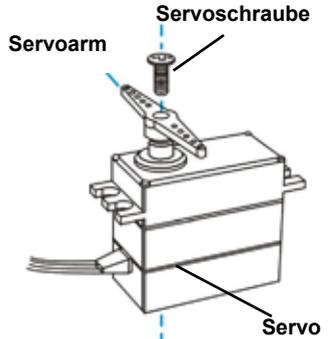
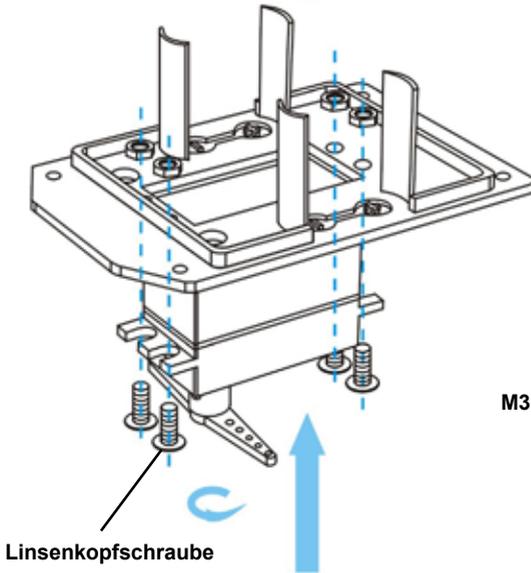
Befestigen Sie den Servo genau so, wie es in den Zeichnungen beschrieben wird.

Montieren Sie die Servoachse auf dem Servo, beachten Sie dabei bitte die kleine Detailskizze!

Montage des Bodenservos:

Zur Montage des Bodenservos wird folgendes benötigt;

- 1 St. Bodenteil
- 1 St. Servo
- 4 St. Linsenkopfschraube
- 4 St. Mutter M3
- 1 St. Servoarm
- 1 St. Selbstschneidende Schraube



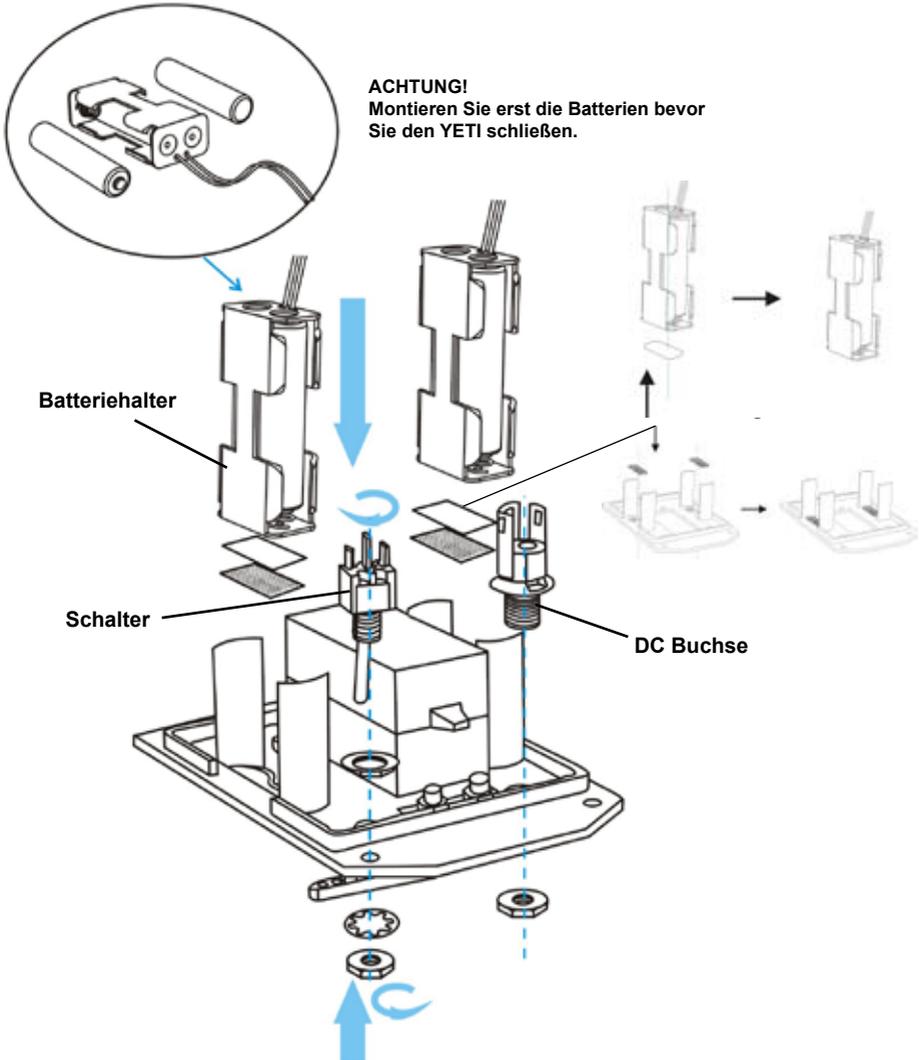
Befestigen Sie den Servo genau so, wie es in den Zeichnungen beschrieben wird.

Montieren Sie die Servoachse auf dem Servo, beachten Sie dabei bitte die kleine Detailskizze!

Endmontage der Bodenplatte:

Zur Endmontage des Bodenservos wird folgendes;

- 1 St. Montiertes Bodenteil
- 1 St. Schalter
- 1 St. DC Buchse
- 2 St. Batteriehalter
- 4 St. AA Akku
- 2 St. Klett-Band male
- 2 St. Klett-Band female

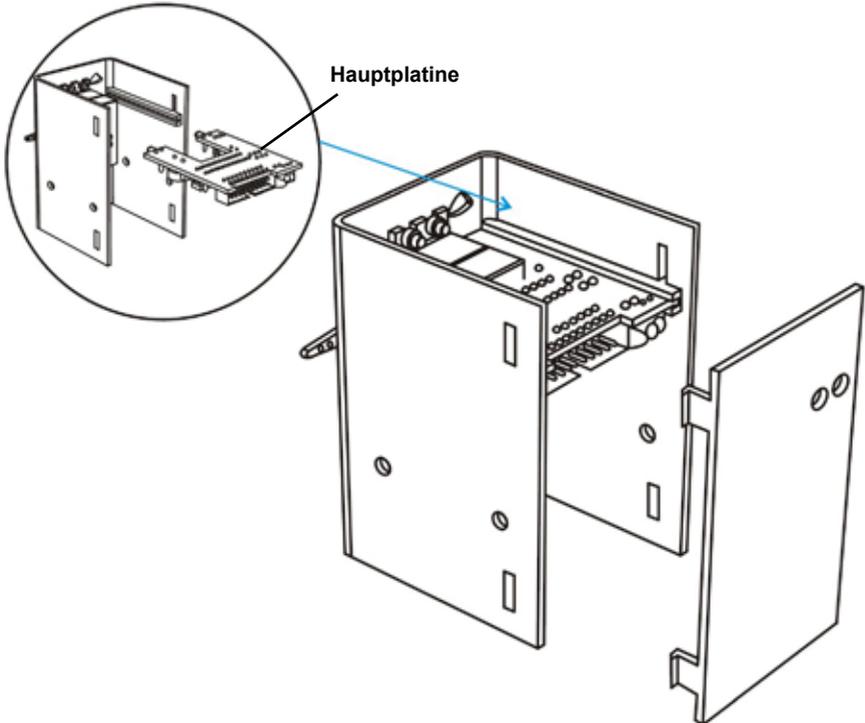


Montieren Sie die Bodenplatte genauso, wie es in die Zeichnung beschrieben wird.

Montage des YETI Kopfes:

Zur Montage des Kopfes wird folgendes benötigt;

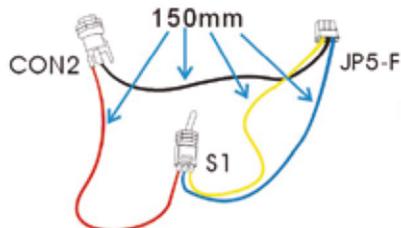
- 1 St. Montiertes Bodenteil
- 1 St. Montiertes Kopfteil
- 1 St. Montierte Hauptplatine
- 1 St. Hinterabdeckung



ACHTUNG!

Montieren Sie erst die komplette Verkablung bevor Sie die Hinterabdeckung schließen.

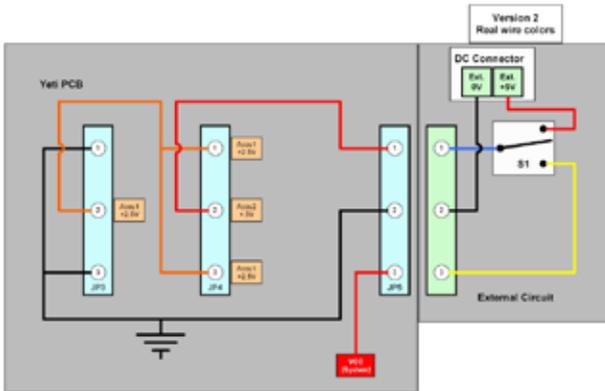
Den Schaltplan für den Kabelanschluß finden Sie auf den Seiten 37 und 38



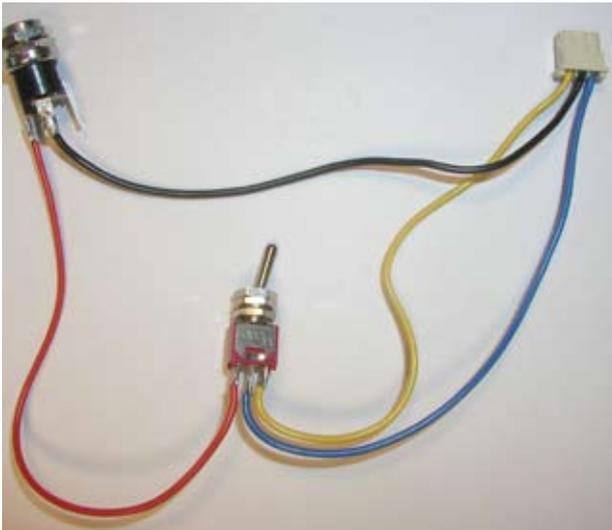
Kabel Endmontage:

Zur Endmontage der Kabel wird folgendes benötigt;

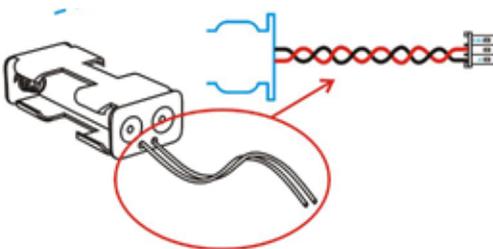
- 1 St. Montiertes Bodenteil
- 1 St. Montiertes Kopfteil
- 1 St. Montierte Hauptplatine
- Montierte Kabelsatz mit;
 - 1 St. Kabel schwarz
 - 1 St. Kabel blau
 - 1 St. Kabel gelb
 - 1 St. Kabel rot



Montieren Sie zunächst die Verkabelung genau so wie es in die Zeichnung beschrieben wird.

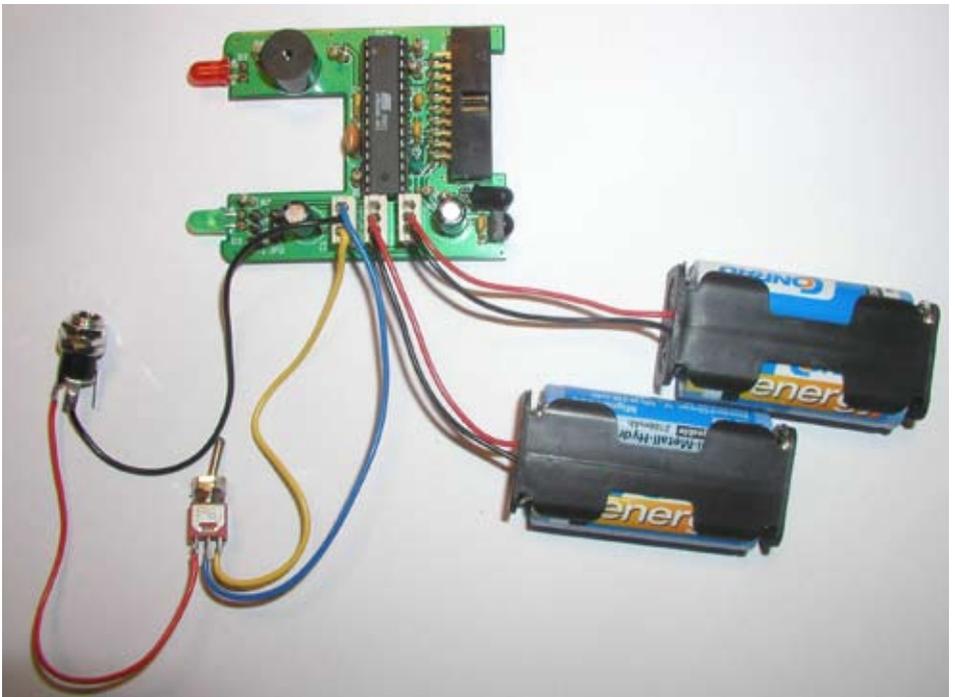
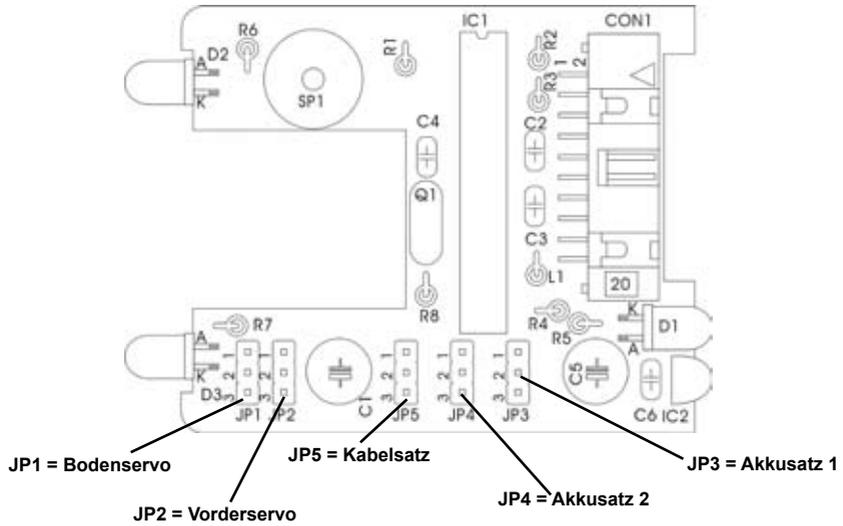


Fertig montierter Kabelsatz



Montierter Akku Kabelsatz

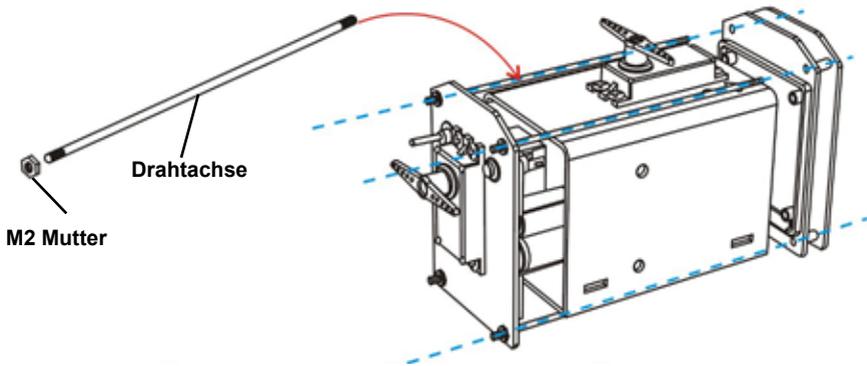
Anschlussbelegung der Hauptplatine



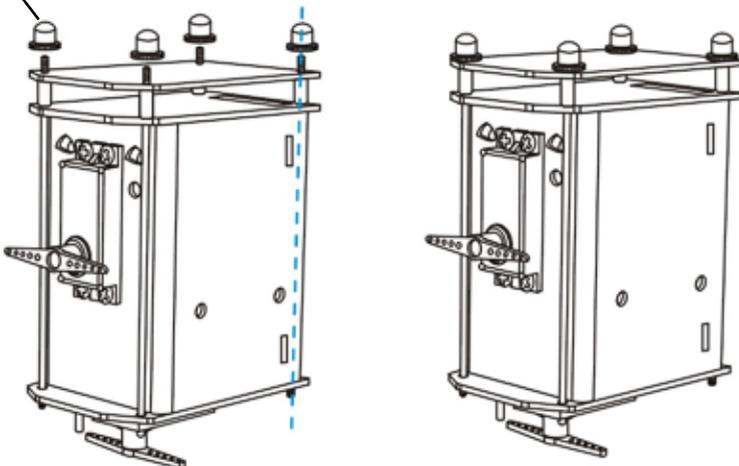
Endmontage des YETI Kopfs:

Zur Endmontage des Kopfs wird folgendes benötigt;

- 1 St. YETI Kopf
- 4 St. Drahtachse
- 4 St. M2 Mutter
- 4 St. Kugelkopfmutter



Kugelkopfmutter

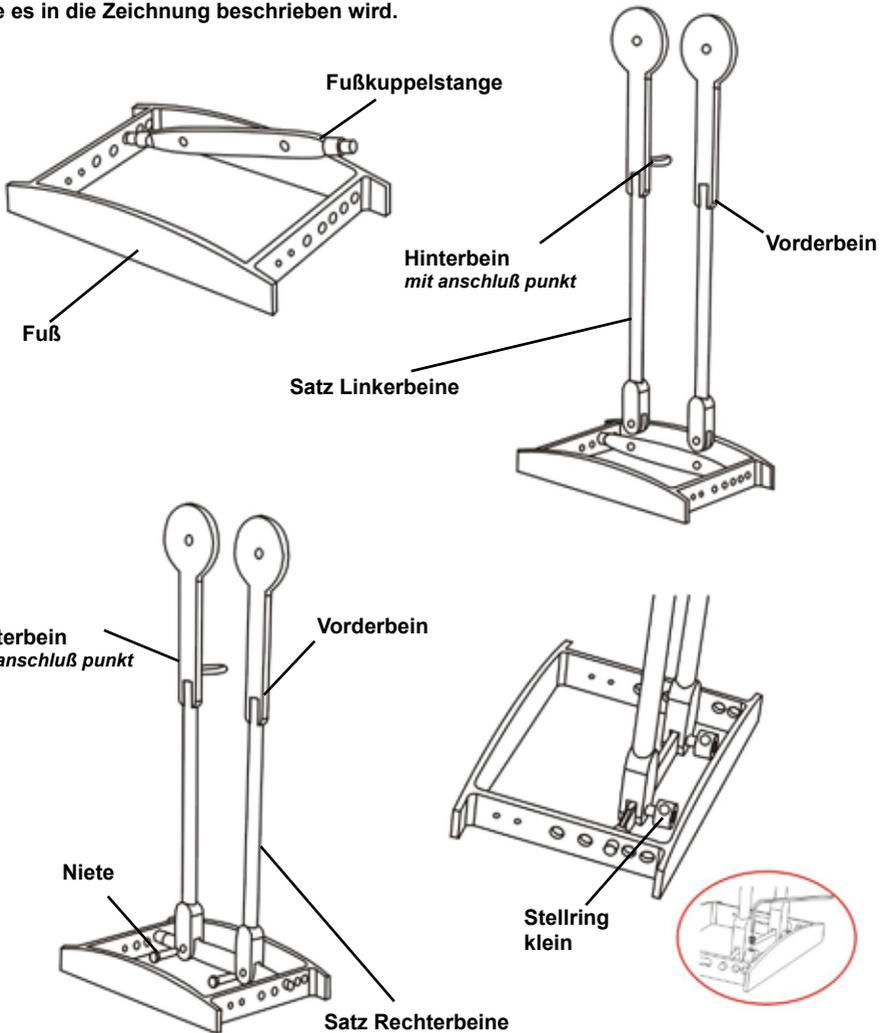


Bein- und Fußmontage:

Zur Bein- und Fußmontage wird folgendes benötigt;

- 2 St. Fuß
- 2 St. Vorderbein
- 2 St. Hinterbein
- 2 St. Fußkuppelstange
- 2 St. Niete
- 2 St. Stelling klein

Montieren Sie Beine und Fuß genauso, wie es in die Zeichnung beschrieben wird.

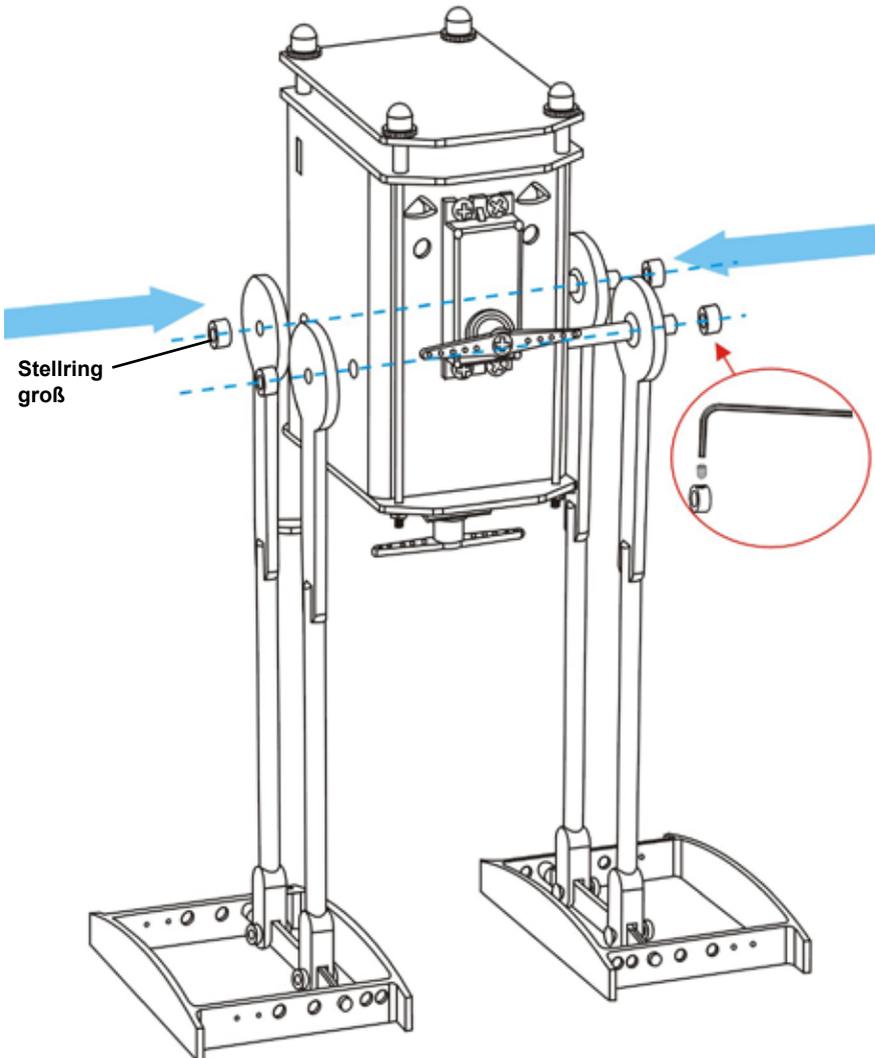


Montage der Beine I:

Zum ersten Teil der Montage der Beine wird folgendes benötigt;

- 1 St. Montiertes Chassis
- 1 St. Montierte Beinesatz Links
- 1 St. Montierte Beinesatz Rechts
- 2 St. Achse
- 4 St. Stellring

Montieren Sie die Beine an am Chassis genau so, wie es in die Zeichnung beschrieben wird.

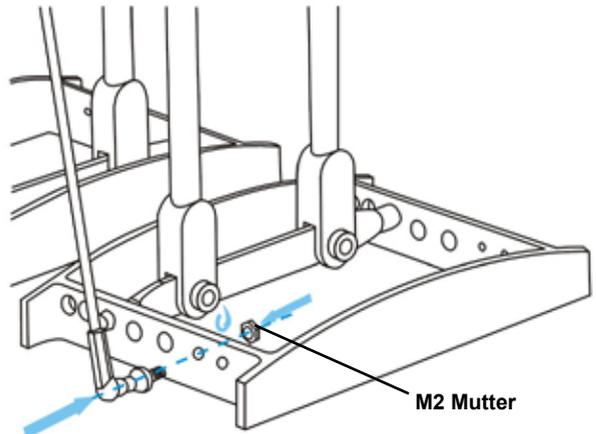
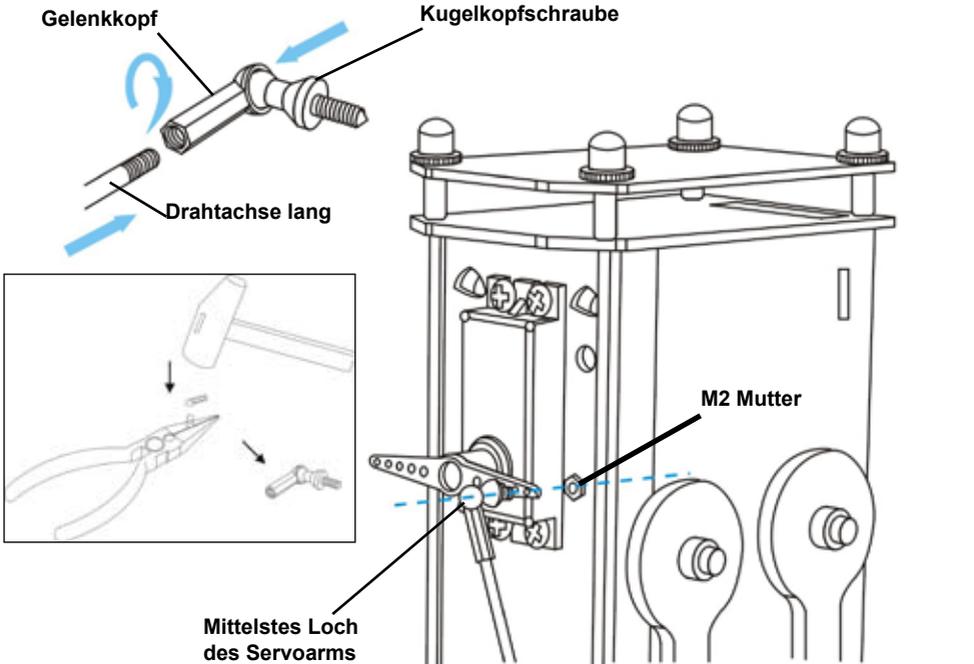


Montage der Beine II:

Zum zweiten Teil der Montage der Beine wird folgendes benötigt;

- 1 St. Montiertes Chassis
- 4 St. Gelenkkopf
- 4 St. Kugelschraube
- 2 St. Drahtachse lang
- 4 St. Mutter M2

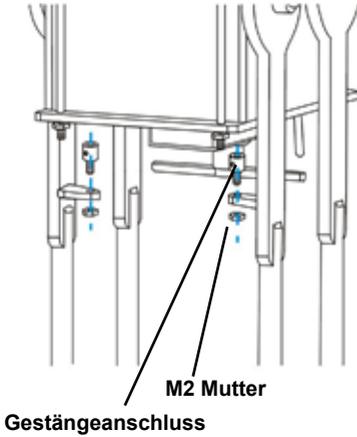
Montiere das Drahtende mit Kugelkopf genauso, wie es in die Zeichnung beschrieben wird.



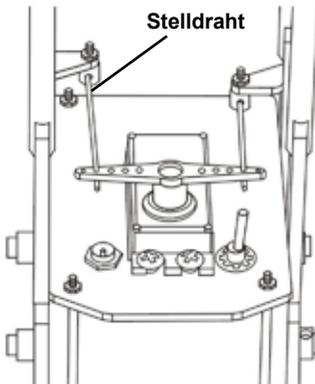
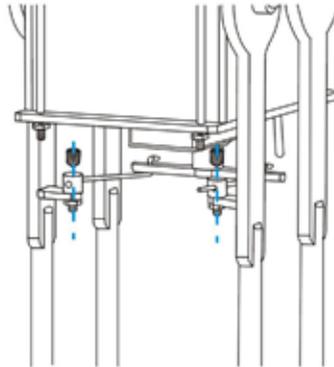
Endmontage der YETI Beine:

Zur Endmontage der YETI Beine wird folgendes benötigt;

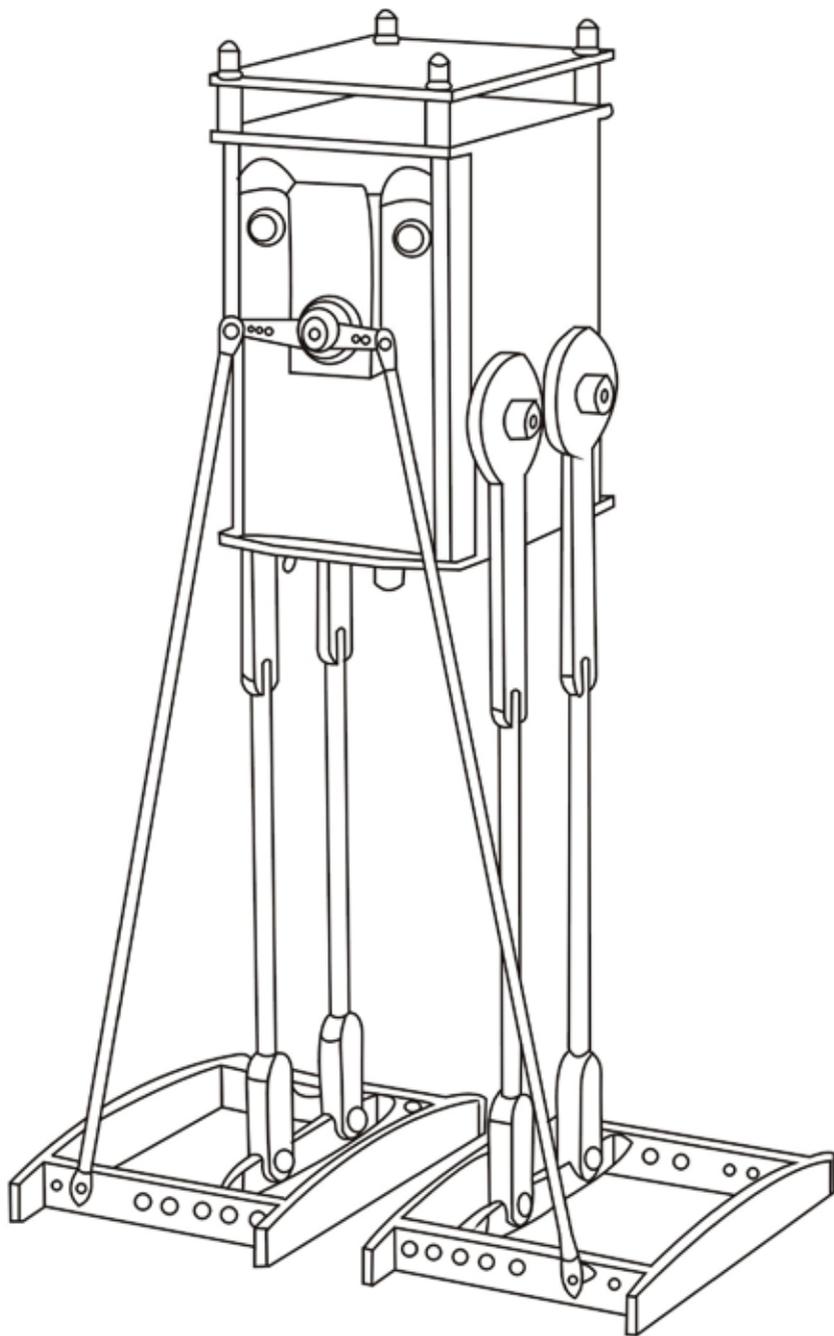
- 1 St. Montiertes Chassis
- 2 St. Mutter M2
- 2 St. Gestängeanschluss
- 2 St. Stelldraht



Montieren Sie die Beine genau an der Servo, wie es in die Zeichnungen beschrieben wird.



DER FERTIGE YETI !



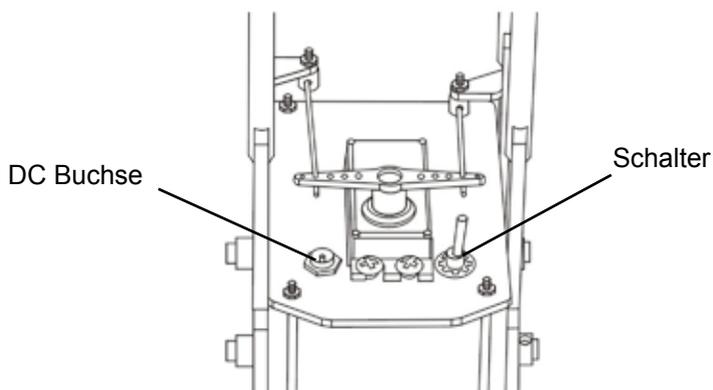
9. YETIS AKKUS AUFLADEN

Als Versorgungsspannung ist eine Gleichspannung von 4,8 Volt vorgesehen, die mit 4 NiMH (1,2V) Akkumulatoren erzeugt wird.
ACHTUNG: Die Akkus sind nicht geschützt mit ein Widerstand!

ACHTUNG: Die Akkus sind nicht durch eine Widerstand oder eine Sicherung geschützt!

Verwenden Sie zum Laden nur geprüfte Ladegeräte am besten ein Mikroprozessorgesteuertes!

Alternativ funktioniert auch ein stabilisiertes Stecknetzgerät mit sehr niedrigem Ladestrom, z.B. 5 Volt / 300 mA.



Akkus laden:

1. Den Schalter auf Aus stellen!
2. Ladegerät an die DC-Buchse anschließen.

YETI mit externer Spannungsversorgung betreiben:

1. Stabilisierte Stromversorgung (5V) an die DC-Buchse anschließen.
2. YETI Anschalten!

10. INSTALLATION DER SOFTWARE

Die YETI-CD ins CDROM-Laufwerk einlegen, sie ist autostartfähig. Falls Autostart deaktiviert ist, kann man die CD auch mit dem Windows-Explorer öffnen. Nach der Auswahl der Sprache findet sich im Abschnitt "Software" alles, was für den Betrieb von YETI erforderlich ist. Diese Programme müssen zunächst installiert werden. Für die Installation des Compilers sind Administrator-Rechte erforderlich. Falls der aktuelle Benutzer diese Rechte nicht besitzt, abmelden und als Administrator wieder anmelden.

Während der Softwareinstallation werden folgende Schritte durchgeführt:

1. Das Flash-Tool zum Übertragen der eigenen Programme auf YETI wird installiert
2. Ein Programmeditor (Programmers Notepad 2, PN2) und ein Compiler (WinAVR) wird installiert.
3. Ein Beispielprogramm wird von CDROM auf Festplatte kopiert.
4. Im Programmeditor (PN2) wird je ein Menüeintrag für MAKE und für CLEAN eingerichtet.

10.1 Windows

10.1.1 Flash-Tool

Das Flash-Tool kann entweder in ein Verzeichnis auf der Festplatte kopiert (z.B.:C:\Programme\Fly) oder später direkt von CD ausgeführt werden. In jedem Fall ist es hilfreich, wenn man sich einen Link auf den Desktop einrichtet, um das Flash-Tool einfach starten zu können.

10.1.2 Installation des Programmeditors und des Compilers

Für die Installation des Compilers sind Administratorrechte erforderlich (weil bei der Installation die Registry geändert wird). Sollte der aktuelle Benutzer diese nicht besitzen, abmelden und als Administrator wieder anmelden!

Klick auf [Install]

COMPILER WinA VR (20060421)

Folgendes Fenster erscheint:



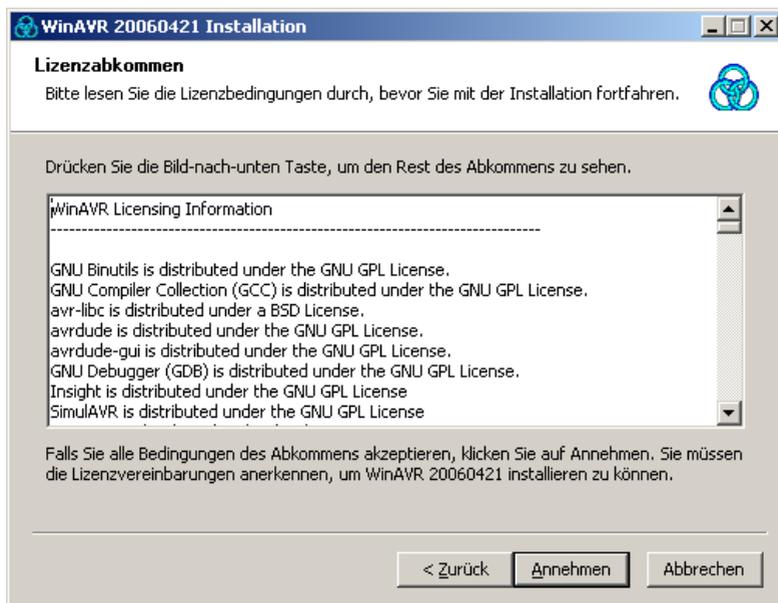
Klick auf OK [I Agree]

Folgendes Fenster erscheint:



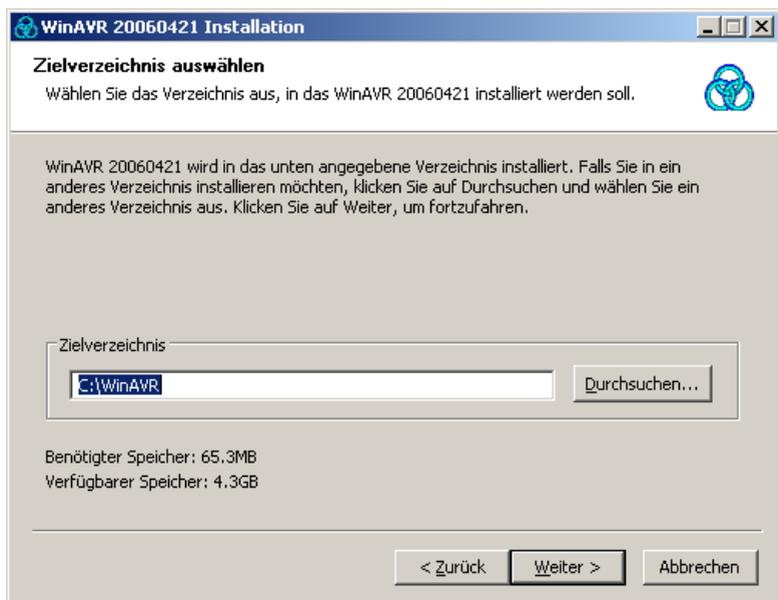
Klick auf Weiter [Next]

Folgendes Fenster erscheint:

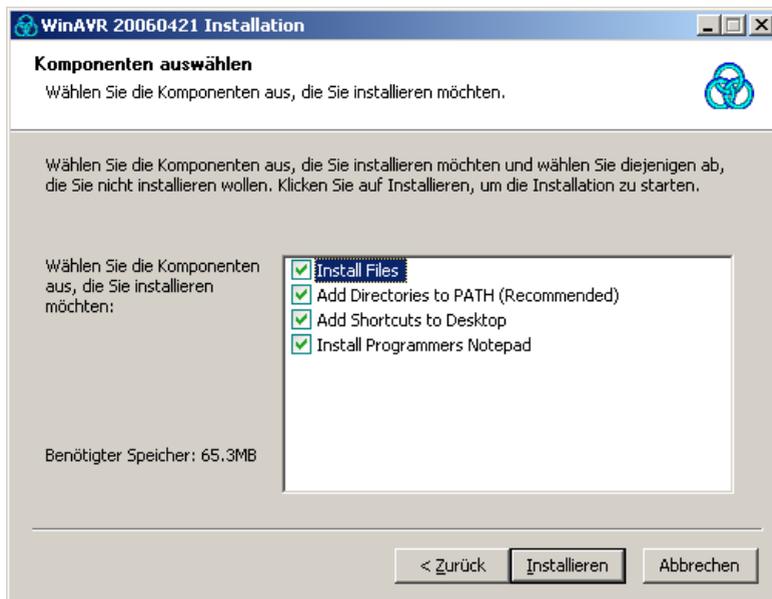


Klick auf Annehmen [Agree]

Folgendes Fenster erscheint:

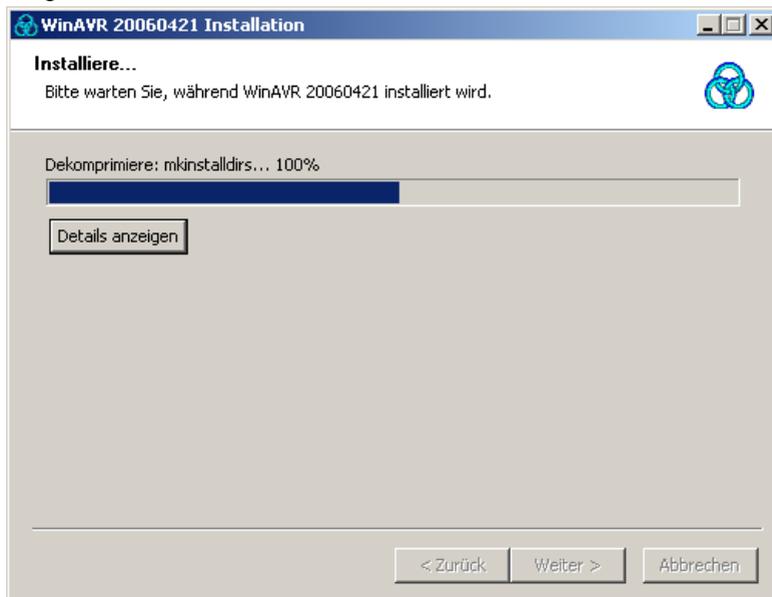


Folgendes Fenster erscheint:



Klick auf Installieren [instal]

Folgendes Fenster erscheint:

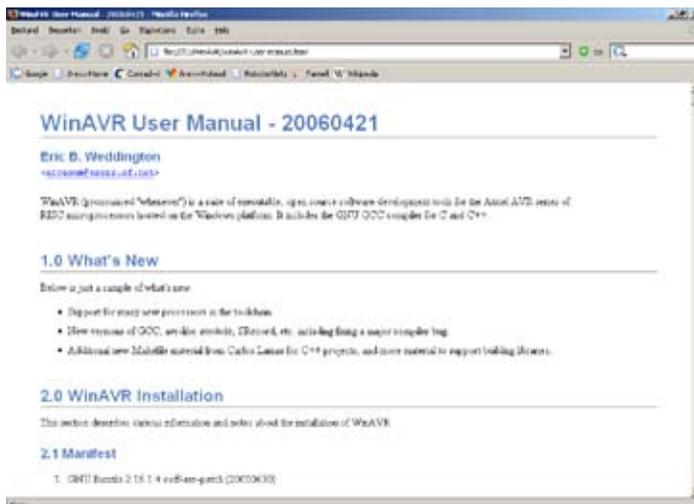


Folgendes Fenster erscheint:



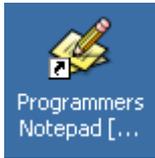
Klick auf Fertig Stellen [safe]

Folgendes Fenster erscheint:



Schliessen Sie dieses Fenster

Auf dem Desktop erscheint das 'Programmiers Notepad 2' Symbol:



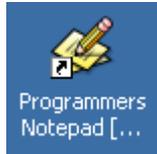
Der Programmierer und der Compiler sind jetzt installiert.

10.1.3. Beispielprogramme von CDROM auf die Festplatte kopieren

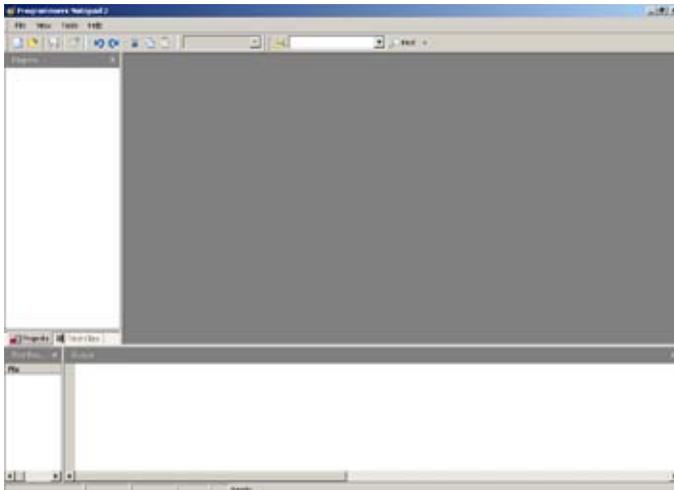
Von der CD den Ordner YETI_src' in einen beliebigen Ordner (Z.B. 'C:\YETI_src') auf der Festplatte kopieren.

Dass YETI Program öffnen und compilieren.

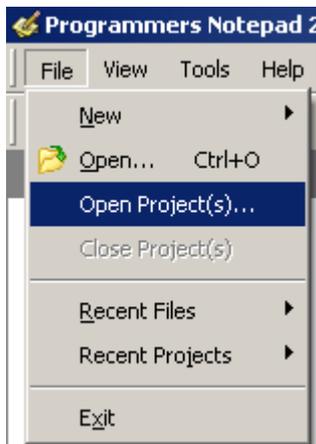
Doppelklicken Sie auf das 'Programmiers Notepad 2' Symbol auf dem Desktop.



Folgendes Fenster erscheint:



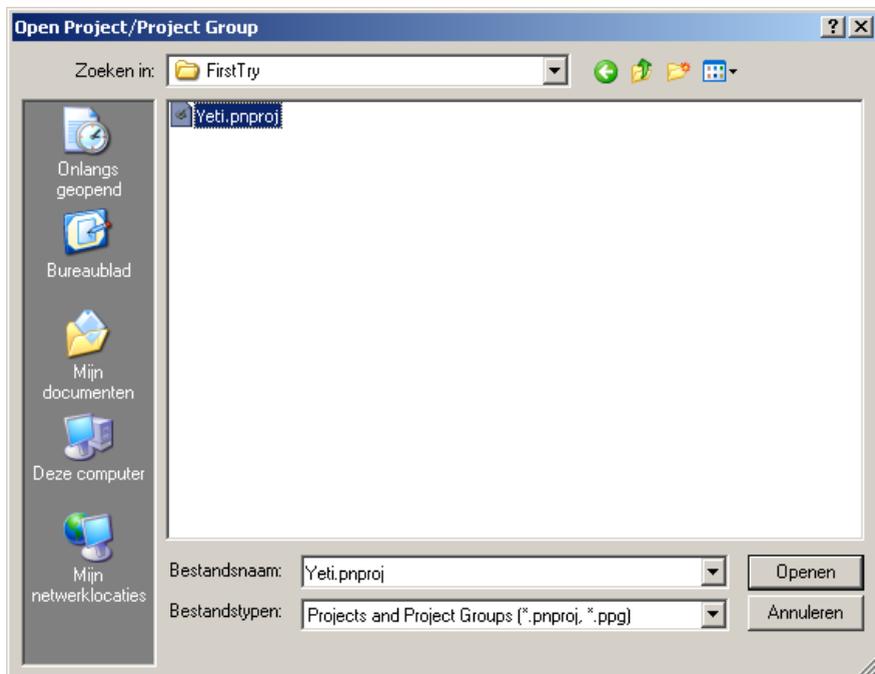
Selektieren: Open Project(s)...



Finden Sie die Datei:

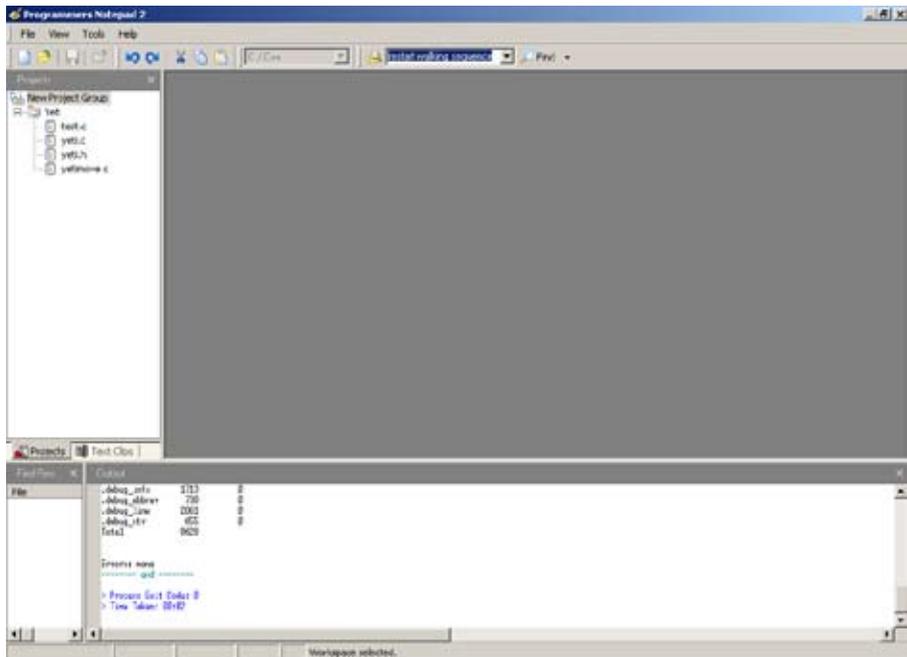
C:\YETI_scr\Firsttry\YETI.pnproj

Folgendes Fenster erscheint:

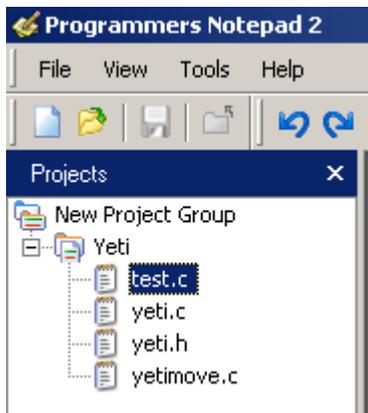


Und klicken Sie auf Offnen

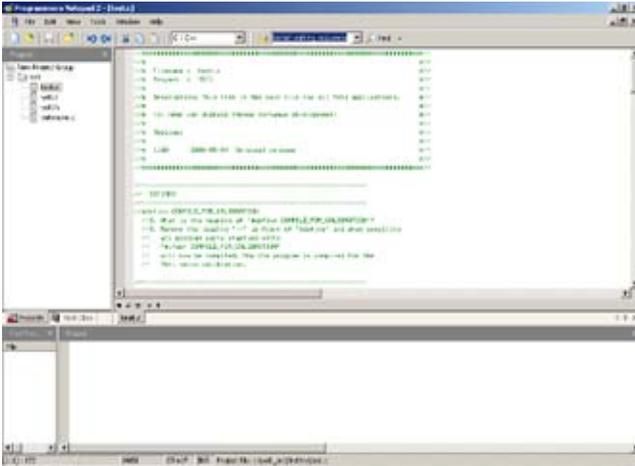
Folgendes Fenster erscheint:



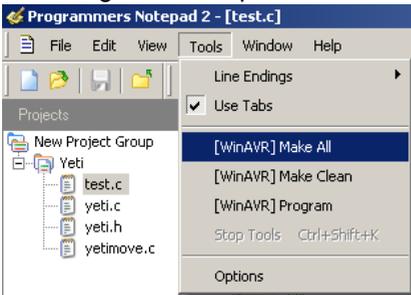
Doppelklicken auf 'test.c':



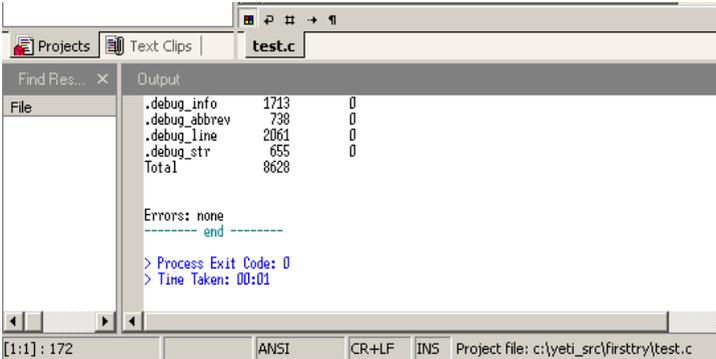
Folgendes Fenster erscheint:



Das Programm kompilieren:



Warten bis...



...Errors: 'none' und 'Process Exit Code: 0' erscheint.

Damit ist das Kompilieren fertig und die Datei 'test.hex' generiert worden.

...und wenn das Programm keine Fehler enthält (was zu erwarten ist, da gerade das Beispielprogramm geladen ist), erscheint unten die Meldung:
Errors: none.

Was ist passiert?

Aus der Datei test.c (und YETI.c) ist eine Datei test.hex generiert worden. Diese Datei stellt das in Maschinencode übersetzte Programm dar, welches nun in YETI Speicher geladen (geflasht) werden kann. Das Programm selbst macht noch gar nichts, wir werden es später aber zum Ausprobieren des Flash-Tools benötigen.

Wie hat's funktioniert?

Der Menüeintrag 'make' ruft die Batch-Datei Test-all.bat auf (eine Batch-Datei enthält eine Liste mit Kommandozeilenbefehlen, die der Reihe nach ausgeführt werden).

In Test-all.bat wird der Befehl 'make all' ausgeführt. 'make' führt immer ein makefile aus, das sich (bei der YETI-Programmierung) im gleichen Ordner befinden muss, wie Test-all.bat .

Ein makefile ist eine Textdatei, die festlegt, wie ein oder mehrere Programm kompiliert werden müssen. Bei Programmen, die nur aus einer Datei übersetzt werden ist das noch recht übersichtlich. Nachdem aber ganze Betriebssysteme in C geschrieben werden und der Code auf mehrere Dateien aufgeteilt ist, die alle in einer bestimmten Reihenfolge übersetzt und zusammengebunden (gelinkt) werden müssen, kann auch ein makefile sehr aufwändig werden. Das 'all' ruft den Eintrag im makefile namens 'all' auf, was bedeutet, dass ein komplettes Projekt und nicht nur einzelne Teile übersetzt werden sollen.

Das zu unserem Beispielprogramm gehörige makefile ist so geschrieben, dass es eine Datei mit Namen test.c zusammen mit YETI.c (das einige vordefinierte Funktionen enthält) kompiliert und eine .hex-Datei erzeugt, die so auf YETI geladen werden kann.

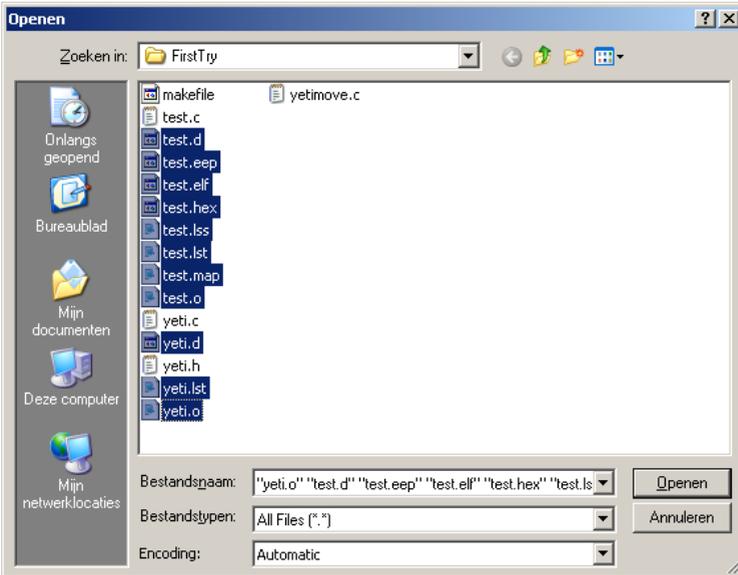
Achtung! Das heißt auch, dass – solange das makefile nicht geändert, sondern nur kopiert wird – das das eigene Programm immer test.c heißen muss.

Wer makefiles komplett verstehen will (was aber für erste Schritte nicht erforderlich ist) kann beispielsweise unter <http://www.gnu.org/> die Dokumentation zu Make ziehen.

Die Grundlagen der YETI-Programmierung werden in Kapitel 13 erklärt.

Beim Compilieren eines Programms werden einige "Nebendateien" erzeugt, die nur während der Übersetzung benötigt werden und später überflüssig sind. Diese können mit dem neu eingerichteten 'clean'-Tool gelöscht werden.

Mit 'File -> Open' kann man die generierten Dateien anschauen. Hier sind die neue Dateien markiert mit u.a. die Datei 'test.hex'



Mit 'Tools -> [WinAVR] Make Clean' lassen sich diese markierten Dateien wieder löschen.

FERTIG

Mehr WINAVR info sehe; <http://www.kreatives-chaos.com> --> software

10.2. LINUX

Für die Installation der Software sind root-Rechte erforderlich. Entweder ausloggen und als root einloggen oder eine shell öffnen und mit "su" root-Rechte erlangen.

10.2.1 Flash-Tool

Die YETI CD-ROM einlegen, ggf. mounten und die beiden Flash-Tools "YETIflash" und "YETIcon" aus dem Verzeichnis "/Linux/Tools/" in das Verzeichnis "/usr/local/bin" kopieren.

Danach noch das Ausführen mit "chmod a+x /usr/local/bin YETIcon YETIflash" erlauben.

Wird ein in einer Shell eingetipptes "YETIflash" nicht gefunden, muss der Pfad "/usr/local/bin" noch der %PATH-Variable hinzugefügt oder das Programm mit vollem Pfad aufgerufen werden.

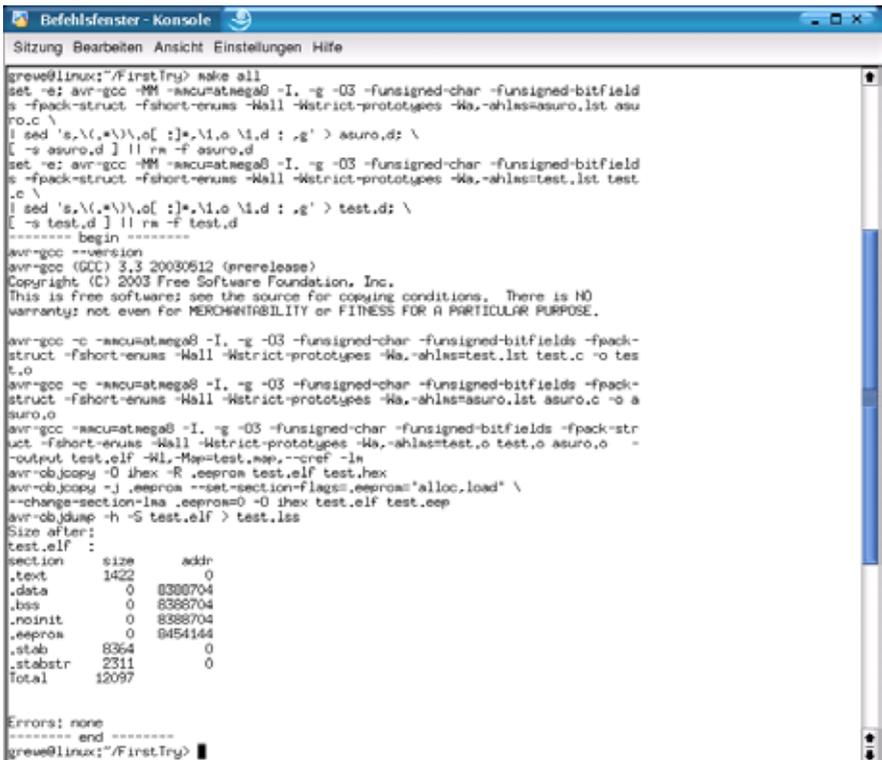
10.2.2 Compiler

Zur Installation des Gnu-Compilers für AVR-Prozessoren die YETI CD-ROM einlegen und aus dem Verzeichnis "/Linux/Compiler/" mindestens die folgenden Pakete in der angegebenen Reihenfolge installieren:

1. **avr-binutils-... .rpm**
2. **avr-gcc-... .rpm**
3. **avr-libc-... .rpm**

Die Installation ist denkbar einfach! Einfach in der Konsole mit root-Rechten den Befehl : rpm -i <paket>.rpm eingeben. **Fertig!**

Als Editoren eignen sich zum Beispiel Exmacs, Kate oder Kedit. Zum Ausprobieren kopiert man sich (als normaler User) die Beispieldateien von der CD aus dem Verzeichnis "/ASURO_src/FirstTry/" ins home-Verzeichnis beispielsweise unter "~/ASURO/". Danach öffnet man eine Shell, wechselt in obiges Verzeichnis und gibt "make" ein. Ist alles richtig installiert, ergibt sich etwa folgendes Bild (siehe Abbild. 10.1):



```
Befehlsfenster - Konsole
Sitzung Bearbeiten Ansicht Einstellungen Hilfe

greue@linux:~/FirstTry> make all
set -e; avr-gcc -MM -acu=atmega8 -I. -g -O3 -funsigned-char -funsigned-bitfield
s -fpack-struct -fshort-enums -Wall -Wstrict-prototypes -Wa,-ahlas=asuro.lst asu
ro.c \
| sed 's,\(.*\)\\.o[ ]:[ ]*,\1.o \1.d : ;g' > asuro.d; \
[ -s asuro.d ] || rm -f asuro.d
set -e; avr-gcc -MM -acu=atmega8 -I. -g -O3 -funsigned-char -funsigned-bitfield
s -fpack-struct -fshort-enums -Wall -Wstrict-prototypes -Wa,-ahlas=test.lst test
.c \
| sed 's,\(.*\)\\.o[ ]:[ ]*,\1.o \1.d : ;g' > test.d; \
[ -s test.d ] || rm -f test.d
----- begin -----
avr-gcc --version
avr-gcc (GCC) 3.3 20030612 (prerelease)
Copyright (C) 2003 Free Software Foundation, Inc.
This is free software; see the source for copying conditions. There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

avr-gcc -c -acu=atmega8 -I. -g -O3 -funsigned-char -funsigned-bitfields -fpack-
struct -fshort-enums -Wall -Wstrict-prototypes -Wa,-ahlas=test.lst test.c -o tes
t.o
avr-gcc -c -acu=atmega8 -I. -g -O3 -funsigned-char -funsigned-bitfields -fpack-
struct -fshort-enums -Wall -Wstrict-prototypes -Wa,-ahlas=asuro.lst asuro.c -o a
suro.o
avr-gcc -acu=atmega8 -I. -g -O3 -funsigned-char -funsigned-bitfields -fpack-str
uct -fshort-enums -Wall -Wstrict-prototypes -Wa,-ahlas=test.o test.o asuro.o -
-output test.elf -Wl,-Map=test.map,--cref -ln
avr-objcopy -O ihex -R .eeprom test.elf test.hex
avr-objcopy -j .eeprom --set-section-flags=.eeprom="alloc,load" \
--change-section-lma .eeprom=0 -O ihex test.elf test.eep
avr-objdump -h -S test.elf > test.lss
Size after:
test.elf :
section      size      addr
.text        1422      0
.data         0      8388704
.bss         0      8388704
.noinit      0      8388704
.eeprom      0      0454144
.stab        8364      0
.stabstr     2311      0
Total       12097

Errors: none
----- end -----
greue@linux:~/FirstTry>
```

Abbildung 10.1.: Make all

10.3. Flash - das YETI-Programmier-Tool

Hierzu wird das Programm Flash (siehe Abb. 10.2) benötigt.

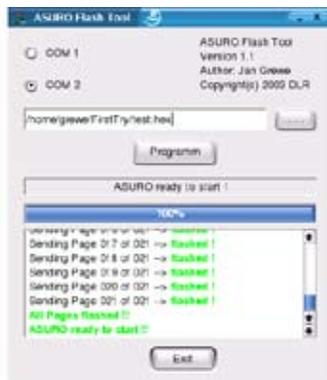


Abbildung 10.2.:
Flash-Tools für
Windows und LINUX

Damit das Flashen auch richtig gut klappt, muss natürlich der RS232- oder USB-IR-Transceiver angeschlossen werden. Danach das Programm starten und die Schnittstelle auswählen, welche auch bei der Inbetriebnahme funktioniert hat.

Die Datei "Test.hex" aus dem Verzeichnis "C:\YETI_src\FirstTry" (bzw. ~/YETI/) auswählen.

Den fertig zusammengebauten und getesteten YETI bereit halten und beim Flash-Tool den Knopf Programm anklicken. YETI muss Sichtkontakt (ca. 50cm Abstand zwischen IR-Transceiver und YETI, beide Bestückungsseiten zeigen zueinander und nichts ist im Lichtweg) zum IR-Transceiver haben und wird jetzt eingeschaltet (S1 auf ON), bevor der Statusbalken ganz rechts angekommen ist. Sollte man mal nicht schnell genug gewesen sein oder sollte die Kontaktaufnahme nicht geklappt haben, einfach YETI ausschalten, erneut Programm drücken und YETI einschalten.

Hat der Verbindungsaufbau geklappt, sieht man an der Statusanzeige und dem Protokollfenster, dass nun die Datei Test.hex zu YETI übertragen wird. Dort wird das Programm im Flash-Speicher des Prozessors abgelegt, sodass das Programm auch nach einem Abschalten der Versorgungsspannung immer wieder zur Verfügung steht.

War der Vorgang erfolgreich, muss YETI aus- und erneut eingeschaltet werden, um das Programm zu starten. Das vor kurzem geschriebene Programm wird ausgeführt und die grüne LED erstrahlt in hellem Glanz.

10.3.1. Wie funktioniert das Flashen?

Sobald das Programm Flash ausgeführt wird, versucht der Rechner 10 Sekunden lang eine Verbindung mit YETI herzustellen. Schaltet man YETI ein, leuchtet die Status-LED für ca. 1 Sekunde zweifarbig auf. Dies ist die "Boot-Phase". YETI schaut nach, ob der PC neue Software für ihn bereitstellt. Diese wird gegebenenfalls geladen. Nach Aus- und Wiedereinschalten wird diese dann gestartet.

10.4. Flash Fehler

Folgende Fehler können während des Flashens auftreten:

- "c" Checksum Error. Es sind andere als die vom PC geschickten Daten bei YETI angekommen.
Das kann durch Störlicht (wie Leuchtstofflampen) kurze Unterbrechungen in der Sichtverbindung oder ähnliches passieren.
- "t" Timeout. Die Sichtverbindung zu YETI ist abgerissen.
- * "v" Verify Error. YETI hat falsche Daten in seinen Flashspeicher geschrieben. Das sollte normalerweise nicht passieren und ist ein Zeichen dafür, dass der nichtflüchtige Programmspeicher (Flash-EPROM) das Ende seiner Lebenszeit erreicht hat, was typischerweise erst nach 10.000 Programmierungen passiert.

Es wird bis zu zehnmal versucht den Fehler zu beheben. Gelingt dies nicht, wird der Flashvorgang abgebrochen.



Treten beim Flashen gehäuft Checksum Errors auf, hilft es oftmals das Raumlicht abzuschalten bzw. etwas abzuschatten, besonders wenn es sich um Leuchtstofflampen handelt.



Immer erst den Programm-Knopf drücken, dann ASURO einschalten, sonst ist kein Softwaredownload möglich!

11. INBETRIEBNAME UND TEST

Endlich ist alles zusammengebaut und der Spaß kann beginnen. Zunächst müssen aber noch die zuvor eingebauten Fehler gesucht, gefunden und beseitigt werden, ohne dabei allzu großen Schaden anzurichten.

11.1. RS232-Infrarot-Transceiver

Diese Inbetriebnahme gilt nur für den RS232-IR-Transceiver.

Als erstes sollte der RS232-IR-Transceiver auf seine volle Funktionsfähigkeit überprüft werden, da dieser später für den Selbsttest des Fahrzeugs benötigt wird.

Dazu wird der RS232-IR-Transceiver über das mitgelieferte serielle Verlängerungskabel an eine freie serielle Schnittstelle angeschlossen.

Danach startet man das Terminal-Programm von Windows "Hyperterminal" (unter Linux beispielsweise "Minicom"). Normalerweise findet man es unter Start --> Programme --> Zubehör --> Kommunikation --> Hyperterminal. Falls es nicht vorhanden ist, muss man es von der Windows-CD nachinstallieren. Terminalprogramme stammen eigentlich noch aus der Zeit des Modems und davor, als man häufiger über die serielle Schnittstelle mit anderen Rechnern kommunizierte. Heutzutage werden sie vorwiegend dann benutzt, wenn man sich über das Internet rein textbasiert auf einem anderen Rechner einloggen will.

Nach dem Start von Hyperterminal fragt das Programm nach einem Namen für die neue Verbindung. Hier kann man YETI eingeben und ein beliebiges Symbol auswählen. Im nächsten Fenster wählt man bei "Verbinden über:" die COM-Schnittstelle aus, an der der Transceiver angeschlossen worden ist. Nach Drücken auf "OK" wählt man

- Bits pro Sekunde: 2400
- Datenbits: 8
- Parität: keine
- Stoppbits: 1
- Flusststeuerung: kein



Danach wieder bestätigen mit "OK".

Nun den IR-Transceiver ca. 10 cm über ein weißes Blatt Papier halten. Die Bauteile zeigen zum Papier. Nun munter ein paar Tasten auf der PC-Tastatur gedrückt und das Terminal-Programm sollte diese Tasten anzeigen. Der IR-Transceiver sendet dabei den Tastendruck über die IR-Diode (D5), das am Papier reflektierte Signal trifft auf das Empfänger-IC (IC2) und wird zum PC übertragen. Kommen gar keine oder falsche Zeichen an, so kann man mit einem kleinen Schraubenzieher vorsichtig den Trimmer zwischen seinem linken und rechten Anschlag verdrehen und wieder ein paar Tasten drücken, bis korrekte Zeichen erscheinen. Das Ganze funktioniert nicht so wie beschrieben? Schade, hier muss wohl ein Fehler vorliegen, der behoben werden sollte (siehe. WWW.AREXX.COM --> Forum). Zur Sicherheit kann man abschließend den IR-Transceiver wieder abstecken und nochmal ein paar Tasten drücken. Nun dürfen keine Zeichen mehr erscheinen.

11.2. USB-Infrarot-Tranceiver

Diese Inbetriebnahme gilt nur für den USB-Infrarot-Transceiver.

Achtung! Der ungehäusete USB-Infrarot-Transceiver ist empfindlich gegen elektrostatische Entladungen. Vor der Benutzung muss man sich an einem metallischen Körper (Heizung, Computergehäuse) entladen um Schäden zu vermeiden. Alternativ kann man den Transceiver auch in ein für Infrarotlicht durchsichtiges Gehäuse einbauen.

11.2.1 Windows

Der USB-Transceiver wird mit dem USB-Kabel an einer freien USB-Buchse angeschlossen. Es erscheint die Meldung:

“Neue Hardware gefunden: **AREXX YETI USB-IR-Transceiver**”

Danach den USB-Treiber von der YETI-CD installieren. Sollte der Treiber nicht automatisch gefunden werden, das Verzeichnis (D: steht hier für das CD-ROM-Laufwerk) D:\Windows\USB_Driver“ auswählen. Eventuell sind hierzu Administratorrechte erforderlich. Dann abmelden und als Administrator erneut anmelden. Es wird nun ein Treiber installiert, damit man unter Windows den USB-Transceiver wie eine normale serielle Schnittstelle ansprechen kann.

Hat dies Fehlerfrei geklappt, so startet man zum Ausprobieren auch hier das Terminalprogramm “Hyperterminal”, gibt YETIUSB für den Verbindungsnamen an und wählt ein beliebiges Symbol aus. Beim nächsten Fenster “Verbinden über:” wählt man die letzte verfügbare COM-Schnittstelle aus. Nach Drücken auf “OK” wählt man:

- Bits pro Sekunde: 2400
- Datenbits: 8
- Parität: keine
- Stoppbits: 1
- Flusssteuerung: kein



Danach wieder bestätigen mit “OK”.

Nun hält man den Transceiver mit der Seite der Leuchtdiode nach unten 10 cm über ein weißes Blatt Papier. Falls der Transceiver ohne Gehäuse betrieben wird, darf die Platine nur am Stecker oder am Platinenrand gehalten werden, um die Schaltung nicht zu stören. Nun drückt man am Terminalprogramm einige Tasten. Dabei muss die gelbe Leuchtdiode auf der Platine blinken und die Tastendrucke auf dem Bildschirm erscheinen.

11.2.2 Linux

Der USB-Transceiver wird mit dem USB-Kabel an einer freien USB-Buchse angeschlossen.

Ein kurzes "Piep" ertönt, wenn Linux den Transceiver erkannt hat. Um zu überprüfen, ob das Gerät korrekt erkannt wurde, kann man sich den entsprechenden Eintrag im proc-Verzeichnis anschauen:

```
foo@bar:~>cat /proc/tty/driver/usb-serial
```

Was eine Ausgabe produzieren muss, die mindestens folgende Einträge aufweist (statt der "0:" kann auch "1:", "2:" usw. stehen):

```
usbserinfo:1.0 driver:v1.4  
0: module:ftdi_sio name:"FTDI 8U232AM Compatible" vendor:0403 product:6001  
num_ports:1 port:1 path:usb-00:11.2-1
```

Zum Ausprobieren konfiguriert man Minicom auf die Schnittstelle /dev/ttyUSB0 (oder 1, 2 usw...) und folgende Parameter:

- Bits pro Sekunde: 2400
- Datenbits: 8
- Parität: keine
- Stoppbits: 1
- Flusststeuerung: kein



Danach wieder bestätigen mit "OK".

Dazu sind eventuell root-Rechte erforderlich.

Eventuell muss man dem gewünschten User oder der gewünschten Gruppe noch Lese- und Schreibrechte auf dem Device /dev/ttyUSB? einräumen. Das kann mit einem `chmod u+rw /dev/ttyUSB0` (oder 1, 2...) oder `chmod g+rw /dev/ttyUSB0` (auch mit root-Rechten) erfolgen.

Nun hält man den Transceiver mit der Seite der Leuchtdiode nach unten ca. 10cm über ein weißes Blatt Papier. Falls der Transceiver ohne Gehäuse betrieben wird, darf die Platine nur am Stecker oder am Platinenrand gehalten werden, um die Schaltung nicht zu stören.

Nun drückt man am Terminalprogramm einige Tasten. Dabei muss die gelbe Leuchtdiode auf der Platine blinken und die Tastendrucke auf dem Bildschirm erscheinen.

Funktioniert das nicht, bei www.arexx.com --> Forum weiterlesen oder www.roboternetz.de --> Forum.

12. KALIBRIERUNG

Eine genaue Kalibrierung der Servomotoren und YETIs Beinmuskeln garantiert eine stabile Gangart. Eine Softwarejustierung wird benötigt damit die Servomotoren bei einer softwaretechnischen Positionsmittle auch tatsächlich die mechanische Positionsmittle einnehmen. Eine Hardwarejustierung stellt sicher, dass YETIs Beine zur Mittelposition der Servomotoren die genaue Sollposition einnehmen.

12.1. Softwarekalibrierung

Eine Kalibrierung der Servomotoren ist für eine stabile Gangart unerlässlich. Diese Justierung wird einmalig durchgeführt. Wir benötigen dazu das Programm Hyperterminal. Version 6.3 steht für einen Abruf kostenlos im Internet zur Verfügung.

Starten Sie Hyperterminal.

Programmieren Sie Hyper-terminal für COM (X) --> 2400 Baud, 8-Bit, ohne Parität und ohne Hardware Steuerung.

Schließen Sie den USB-Adapter auf den PC an.

Legen Sie YETI kopfüber auf den Tisch und sehen Sie zu, dass der USB-Adapter und YETIs Hinterkopf für die Infrarotlichtübertragung Sichtkontakt haben.

Schalten Sie YETI ein.

Nach etwa 3 Sekunden ertönt eine Reihe Piepstöne.

Drücken Sie jetzt innerhalb 3 Sekunden eine willkürliche Taste der PC-Tastatur.

Mit dem Tastefeld in Tabelle 1 wird YETI jetzt kalibriert.

Drücken Sie zuerst [-]. Beide Servomotoren werden jetzt ihre Mittenpositionen einnehmen.

Justieren Sie jetzt den „vorderen“ Servo-Arm so, dass dieser möglichst genau quersteht zum Servo. Dabei dürfen Sie die Servomotorachse jedoch nicht verdrehen.

Bewegen Sie den „vorderen“ Servomotor mit den Tasten [7] und [9] in eine möglichst exakte Querposition.

MEHR INFO UBER KALIBRIERUNGSOFTWARE

Drücken Sie Taste [1] um diese Position zu speichern.
 Bewegen Sie zum Test jetzt mit Taste [7] oder [9] die Servomotor-position und drücken Sie Taste [8] um den Servomotor wieder in die justierte Nullposition zurück zu fahren.

Wiederholen Sie diese Prozedur für das untere Servomotorsystem mit den Tasten [4], [5], [6] und [2]. Die [+]-Taste wurde programmiert als eine Kombination der Tasten [8] und [5].

Mit der [ENTER] wird die Kalibrierungsprozedur beendet.
 YETI wird jetzt bei jedem Startvorgang seine Servomotoren zuerst in die exakt justierte Nullposition bringen.

ACHTUNG!

In YETI's standard mitgeliefertes Program befindet sich eine funktion 'vCalibrateServos()' mit der die Kalibrierung ausgeführt wird.

Numerisches Tastenfeld am PC			
			[-] Bewege 2 Servos in ihre unkalibrierte Nullposition
[7] 'Vorderer' Servomotor Linksdrehung	[8] "Vorderer" Servomotor Kalibrierte Nullposition	[9] 'Vorderer' Servomotor Rechtsdrehung	[+] Bewege 2 Servos in ihre kalibrierte Nullposition (wie [8] & [5] gleichzeitig)
[4] 'Unterer' Servomotor Linksdrehung	[5] 'Unterer' Servomotor Kalibrierte Nullposition	[6] "Unterer" Servomotor Rechtsdrehung	[ENTER] Ende der Kalibrierungs-Prozedur
[1] 'Vorderer' Servomotor Position Speichern	[2] 'Unterer' Servomotor Position Speichern		

12.2. Hardware Kalibrierung

Die Servomotoren des YETIs sind jetzt justiert. Nun sind die Beine an der Reihe.

Lösen Sie bitte die Schrauben für Befestigung der „Beinmuskeln“ an den Hinterbeinen. Die Beinmuskeln sollten sich dabei leicht durch die Führung bewegen können.

Legen Sie YETI auf den Rücken auf einer CD-Schachtel. Die „Fersen“ liegen nun etwas (d.h. eine CD-Schachteldicke) tiefer als der Rücken. (Siehe Abb. 12.1).

Kleben Sie einen Streifen Klebeband fest um den Roboterkörper und dessen Beine, so dass sich die Beine nicht mehr bewegen können.

Starten Sie Hyperterminal. Legen Sie YETI bitte kopfüber auf den Tisch und sehen Sie zu, dass der USB-Adapter und YETIs Hinterkopf für die Infrarotlichtübertragung Sichtkontakt haben.

Schalten Sie YETI ein. Warten Sie bis eine Piepstonreihe ertönt und YETI seine Beine in die Nullposition bringt.

Drücken Sie jetzt innerhalb 3 Sekunden eine willkürliche Taste der PC-Tastatur. YETI piepst einmal und wird innerhalb die Kalibrationsfunktion bleiben, damit die Servos aktiviert bleiben.

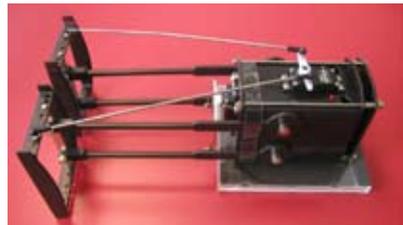
Befestigen Sie die Schrauben der Beinmuskeln zu den Hinterbeinen mit einem Imbus-Schlüssel.

Schalten Sie YETI aus.

Entfernen Sie das Klebeband.

Und jetzt, bitte sehr, ist YETI zum ersten Laufgang bereit.

Abb. 12.1
YETI Hardware Kalibrierungsposition



13. PROGRAMMIEREN

Und jetzt?

So, unser YETI ist betriebsbereit und funktionsfähig. Das war's wohl, oder?

Vergiß es, die wirkliche Arbeit fängt jetzt erst an!

Erfahrene C-Programmierer können nun direkt mit der Programmierung beginnen. Anfänger sollten jedoch lieber zuerst dieses Kapitel lesen, auch wenn das eine oder andere Thema Ihnen vielleicht bekannt vorkommt.

Ein Minicomputer

Wir fangen mal mit einer einfachen Einführung an. Die Hauptplatine des YETIs enthält einen Minicomputer. Einen solchen Kleinstrechner nennt man auch 'Mikrocontroller' und wird in einem IC eingebaut. Ein 'IC' ist ein Integrated Circuit (Integrierter Schaltkreis) und wird manchmal auch kurzweg Chip genannt. Unser 'Mikrocontroller' befindet sich nun in der kleinen schwarzen Box mit den 28 Beinchen, welche die elektrische Anschlüsse zu den roten Lämpchen, zum Lautsprecher, und zu den Antriebsmotoren in den YETI-Muskeln und Infrarot Sender/Empfänger des YETIs versorgen.

Diese Kurzbeschreibung reicht zur Einführung, wird jedoch in den nachfolgenden Kapiteln für die Anfängergruppe Schritt für Schritt erläutert. Beim Lesen werden Sie feststellen, dass die Lektüre langsam anspruchsvoller wird und Sie unbemerkt mit der Programmierung angefangen haben.

Rote Augenlichter

Die roten Augenlichter des YETIs sind sogenannte LEDs. LED ist eine verkürzte Schreibweise für Light Emitting Diode oder Leuchtdiode. Eine Diode ist ein Elektronikteil, das den elektrischen Strom nur in eine Richtung leitet. Eine LED ist eine spezielle Diode, die dabei auch noch leuchtet. In der Leuchtdiode befindet sich kein Glühdraht. Die LED wird deshalb nicht heiß und verbraucht relativ wenig Energie. Ausführliche Information über LEDs finden Sie im Internet, z.B. unter www.wikipedia.org.

Motoren

Der YETI verfügt über zwei Spezialmotoren. Diese Motoren befinden sich zusammen mit einigen Zahnrädern und mit einiger Steuerelektronik jeweils in einem kleinen Gehäuse. Ein solches Antriebsmodul nennt man auch 'Servomotor'. Die Zahnräder bilden ein Getriebe. In der Praxis setzt das Servomotorgetriebe die Drehzahl herab und das Ausgangszahnrad dreht sich erheblich langsamer als der Motor selbst. Tatsächlich kann das Ausgangszahnrad nicht einmal eine volle Umdrehung durchführen und erreicht nur eine Dreivierteldrehung von etwa 220 Grad. Die Herabsetzung der Drehzahl führt jedoch zu einer erheblichen Drehkraft.

Der Mikrocontroller und sein Befehlssatz

Der Befehlssatz für den Mikrocontroller umfaßt 120 einzelne Basis-kommandos, die in der englischen Fachsprache der Programmierer auch 'Instructions' genannt werden. Der Mikrocontroller kann jedoch Hunderte von Basiskommandokombinationen verarbeiten. Eine Reihe von Basiskommandos wird ein Programm genannt. Zur Ausführung eines Programms muß der Mikrocontroller diese Reihe in seinen Speicher laden, nimmt dann ein einzelnes Basiskommando aus dem Speicher und führt es aus. Dann nimmt er das nächste Kommando der Reihe und führt auch dieses aus. Der Prozessor wiederholt diese Prozedur so lange, bis er in der Reihe ein 'STOP'-Kommando begegnet. Erst danach wird das Programm beendet.

Der Flashspeicher

Bei einem normalen Computer müssen Sie immer zuerst ein Programm, wie zum Beispiel ein Spielprogramm, starten, damit Sie es benutzen können. Beim Starten kopiert der Computer das Programm von der Festplatte oder CD in den Arbeitsspeicher des Computers. Falls Sie den Computer danach abschalten, verschwindet das Programm aus dem Arbeitsspeicher und Sie müssen das Programm zum Benutzen wiederum starten.

In einem Mikrocontroller ist das Programm in einem Speicherspeicher (im sog. 'Flash'-Speicher) abgelegt. Beim Stromausfall und Abschalten bleibt das Programm im 'Flash'-Speicher erhalten. Um ein solchermassen gespeichertes Programm zu löschen, muß der Mikrocontroller zuerst eingeschaltet werden.

Im Mikrocontroller ist der 'Flash'-Speicher sozusagen der Arbeitsspeicher, der ständig bereits das betriebsbereite Programm enthält und es beim Abschalten auch nicht verliert. Sie können diesen Speicher als Kombination eines Festplattenspeichers und eines Arbeitsspeichers betrachten.

Das Laden eines Programms

Wie können wir ein Computerprogramm in den Flashspeicher eines Mikrocontrollers übertragen?

Normalerweise würden wir dazu ein spezielles Programmiergerät benötigen mit einem Sockel, in dem man das Mikrocontroller-IC einsteckt. Danach lädt man das Programm direkt in den Flashspeicher des Mikrocontrollers. In der englischen Fachsprache nennen wir ein solches Programmiergerät einen 'Programmer'.

Dieses würde bedeuten, dass wir den YETI mit einem Spezialstecker an einem Programmiergerät anschließen und dann mit Hilfe des Programmiergeräts Ihr Programm in den YETI übertragen müßten. Soweit wird es jedoch nicht kommen, denn einige fachkundige Entwickler beim DLR (www.dlr.de) haben dazu eine trickreiche Alternativmethode erfunden.

Der Trick besteht aus einem kleinen Programm im Flashspeicher des Mikrocontrollers. Dieses Miniprogramm wird 'Bootloader' genannt und der Name bedeutet: 'Ladeprogramm zum Starten'.

Unmittelbar nach dem Einschalten des Mikrocontrollers untersucht der Mikrocontroller immer den Inhalt des Flashspeichers, ob darin ein Bootloaderprogramm enthalten ist. Wenn ja, wird der Mikrocontroller grundsätzlich zuerst das Bootladerprogramm ausführen.

Achtung, jetzt wird es spannend:

Das Bootloaderprogramm wurde so entworfen, dass es mit dem Infrarot Sende-/Empfangsmodul im YETI, d.h. mit unsichtbaren Lichtstrahlen, Information austauschen kann. Auf dieser Weise können wir mit einem Spezialprogramm auf einem normalen Computer sozusagen mit dem YETI 'reden'. Dieses Spezialprogramm heißt 'Flash' und dieser Name kommt Ihnen vielleicht schon bekannt vor. Selbstverständlich muß der Computer auch mit einem eigenen Infrarot Sende-/Empfangsmodul verbunden werden.

Das Bootloaderprogramm und das Flashprogramm starten nun einen Dialog. Anschließend überträgt das Flashprogramm Kommando nach Kommando ein YETIprogramm zum Bootloaderprogramm. Das Bootloaderprogramm erhält diese Einzelkommandos und legt diese nacheinander in den internen Flashspeicher. Dieser Speichervorgang überschreibt dabei natürlich niemals den Speicherbereich, der das Bootloaderprogramm selbst enthält und beginnt mit dem darauf folgenden Speicherbereich.

Sobald der Programmtransfer abgeschlossen ist, stoppt der Mikrocontroller die Ausführung des Bootladerprogramms und startet die Ausführung des zuletzt geladenen Programms.

Das Bootloaderprogramm beansprucht somit ein fester Bereich im normalen Flashspeicher des Mikrocontrollers und wird derart abgesichert, dass man es weder lesen noch löschen kann. Es ist vielleicht besser so, dass das Bootloaderprogramm, abgesehen vom Löschvorgang im speziellen Programmiergerät, nicht einfach gelöscht werden kann. So brauchen Sie keine Angst haben, dieses Programm irrtümlicherweise zu löschen oder zu überschreiben.

Programmieren

Wie aber erstellen wir jetzt ein Programm?

In aller Kürze:

Sie schreiben ein YETI-Programm in einem Texteditor in einer ganz bestimmten 'Sprache'. Anschließend übersetzen Sie diese Textdatei in sog. HEX-Kommandos, die der Mikrocontroller versteht und in einer HEX-Datei abgelegt werden. Diese HEX-Datei können Sie nun mit Hilfe des im vorangegangenen Kapitel bereits beschriebenen Flashprogramms in den YETIprozessor übertragen.

Theoretisch könnten Sie auch Programme für den Mikrocontroller direkt als Folge von HEX-Kommandos schreiben. Das ist jedoch sehr kompliziert und unübersichtlich und für Laien nahezu unmöglich.

Programme schreiben

In gewisser Weise ähnelt die Programmerstellung oft dem Schreiben eines Briefes und im Idealfall sind Fachleute in der Lage, die Programmquelle auch als Text fließend zu lesen. Zum Erstellen und Ändern (Editieren) eines Programms verwenden wir einen speziellen Textverarbeiter mit dem Namen 'Programmers Notepad2' oder in der Kurzform PN2. Tatsächlich könnten wir zum Schreiben eines YETI-Programms auch einen anderen Textverarbeiter wie zum Beispiel Notepad oder Microsoft Word verwenden, aber das Werkzeug 'Programmers Notepad2' wurde speziell entwickelt zum Editieren der Programmquellen. Dazu sind zum Beispiel bequeme Hilfsmittel, wie Spezialfarben für Schlüsselwörter, Variablen und Kommentare, verfügbar.

Ein Programm besteht aus einer Reihe normaler Textzeilen, die natürlich einer Sprachkonvention entsprechen und einige Randbedingungen erfüllen müssen. Der Programmtext heißt auch 'Programmquelle' oder 'Sourcecode' in der englischen Fachsprache. Sprachkonvention und Randbedingungen bilden die 'Programmiersprache' zusammen. Genau wie bei den menschlichen Fremdsprachen gibt es im Computersprachbereich ebenfalls verschiedene Sprachen und Übersetzungssoftware, die eine Textdatei in eine Kommandodatei für einen ganz bestimmten Mikrokontroller übersetzen kann. Es gibt zahllose Programmiersprachen, aber eines der bekanntesten Sprachen für Mikrokontrollerprogramme ist eine Sprache mit dem kurzen Namen 'C'. Diese Sprache 'C' ist auch die Sprache, die wir für die Programmierung des YETIs ausgewählt haben.

Das Übersetzen eines Programms

Ein Mikrocontroller kann mit einem in der 'C'-Sprache geschriebene Textdatei nichts anfangen. Zum Übersetzen einer in der 'C'-Sprache geschriebenen Textquelle in eine Datei mit HEX-Kommandos für den Mikrocontroller benötigen wir ein Übersetzungsprogramm. Ein solches Übersetzungsprogramm ist zunächst ein ganz normales Computerprogramm, das in Fachkreisen üblicherweise 'Compiler' genannt wird. Für den YETI benötigen wir daher einen 'C'-Compiler. Das dazu geeignete Programm heißt 'gcc.exe'. Dieser Compiler erwartet zur Übersetzung eine Textdatei mit einer Dateinamensendung '.c'. Der Compiler kann somit keine Datei 'test.txt', dafür jedoch 'test.c' sehr wohl übersetzen. Wenn der Textverarbeiter Programmers Notepad2 mit dem korrekten 'C'-Compiler ausgestattet wird, kann der Menüpunkt Tools->Make automatisch die Übersetzung einer Programmquelle in eine HEX-Datei durchführen.

Die Programmübertragung oder der 'Programm upload'

Der Compiler hat jetzt zum Beispiel das 'C'-Programm 'test.c' übersetzt in eine HEX-Datei 'test.hex' mit den Mikrocontrollerkommandos. Das Flashprogramm kann diese HEX-Datei mittels Infrarotlichtübertragung in den YETI-Mikrocontroller transferieren. Der Transport heißt in der englischen Fachsprache auch 'Programm upload', weil das Flashprogramm die HEX-Datei in den YETI-Roboter überträgt.

Konfiguration der Compilersoftware

Sollten Sie beabsichtigen, einen 'C'-Compiler zu kaufen, dann sollten Sie schon einige Hundert Euro beiseite legen. Es gibt jedoch eine Gruppe professionelle und begeisterte Programmierer, die einen ausgezeichneten und kostenlosen 'C'-Compiler entwickelt haben, der zudem noch frei konfigurierbar ist.

Die Softwarehersteller WINAVR hat den 'C'-Compiler und den Programmiers-Notepad2 bereits vorkonfiguriert. Der Benutzer kann diese Einstellung jedoch anpassen. Die Voreinstellung erfordert, dass das Hauptprogramm grundsätzlich als 'test.c' abgelegt werden muß. Außerdem sollte eine Datei 'YETI.c' verfügbar sein, in der ergänzende Funktionen festgelegt werden können. In diesen beiden Dateien werden Hauptprogramm und ergänzende Funktionen ordentlich getrennt verwaltet.

Der Programmiers-Notepad2 übernimmt die Aufgabe, die Quelldateien 'test.c' und 'YETI.c' immer zusammen zu compilieren, so dass man sich als Benutzer auf wichtigeren Aufgaben konzentrieren kann.

Das habe ich doch schon mal gelesen, werden Sie jetzt vielleicht denken. Stimmt, aber beim zweiten Mal Lesen sieht alles viel einfacher aus und ist es auch leichter zu verstehen.

Die Erstellung eines 'C'-Programms für den YETI.

Ist die Erstellung eines 'C'-Programms schwierig?

Die Antwort ist sowohl positiv als negativ. Negativ, weil ein Steuerungsprogramm für den YETI ziemlich einfach sein kann. Andererseits aber auch positiv, weil man in 'C' auch sehr komplizierte Programme schreiben kann.

Wenn man erstmalig mit 'C' in Berührung kommt, ähnelt alles zunächst Zauberei, aber nach einer Weile werden die Programme durchschaubar und übersichtlich. Man muß nicht einmal alles verstehen. Beim Weiterlesen wird Ihnen dieses einleuchten.

Hauptprogrammstruktur

Eine 'C'-Hauptprogramm hat eigentlich immer nachfolgende Grundstruktur:

```
int main(void){
    return 0;
}
```

'int' beschreibt den Typ der Hauptfunktion
'main' ist der Name der Hauptfunktion
'void' bedeutet 'leer' oder 'unbesetzt'
'return 0' bedeutet: liefere eine Zahl Null zurück an die Variable,
 womit die Funktion 'main' aufgerufen wurde.

Grundsätzlich sollte, jede 'C'-Kommandozeile immer mit einem Semikolon, d.h. ';', abgeschlossen werden. Ausgenommen sind nur Zeilen, die mit einer Funktionsklammer '}' enden.

Programmers Notepad2 kann bestimmte Textteile automatisch mit einer speziellen, frei wählbaren Farbe darstellen. So kann man zum Beispiel syntax oder die speziellen 'C'-Kommandocodewörter grün darstellen. Zahlen werden in Rot, Kommentar in Blau geschrieben.

Man kann fast überall Kommentar einfügen, nach '/' oder zwischen '/*' und '*/'

```
int main/*Beliebiger Kommentartext*/(void){ /*Irgendein Kommentar*/
    //Kommentarzeile
    /* Textzeilen
    */
    return 0; // Beliebige Textzeile
/*
weiterer Text
*/
}
```

Nachfolgendes Programm generiert eine Fehlermeldung beim Kompilieren, denn 'Main' wurde mit einem Großbuchstaben geschrieben. Die 'C'-Sprache unterscheidet die Wörter, die mit Großbuchstaben oder Kleinbuchstaben geschrieben werden.

```
int Main(void){  
    return 0;  
}
```

Ein Programm darf zum Beispiel auch wie folgt zusammengescriben werden, aber das Ergebnis ist nicht so gut lesbar:
int main(void){return 0;}

#Kommentare können zum Beispiel am Ende einer Programmzeile eingefügt werden:

```
int main(void){  
    return 0;  
} //Beginn der main-Funktion  
//verlasse die Hauptfunktion unter  
//Rückgabe des Werts 0  
//Ende der main-Funktion
```

#Kommentare können aber auch vor der Programmzeile eingefügt werden:

```
//Beginn der main-Funktion  
int main(void){  
    //verlasse die Hauptfunktion unter Rückgabe des Werts 0  
    return 0;  
} //Ende der main-Funktion
```

#Die 'C'-Sprache setzt immer eine und nur eine Hauptfunktion mit dem Namen 'main' voraus und daneben ggf. noch eine oder mehrere Zusatzfunktionen mit beliebigen Namen.

#Nachfolgendes Programm enthält nur eine Hauptfunktion mit dem Namen 'main', führt jedoch keine Kommandos aus und bildet sozusagen nur ein Gerüst:

```
int main(void){           //Beginn der main-Funktion
    return 0;             //verlasse die Hauptfunktion unter Rück-
                           gabe des Werts 0
}                          //Ende der main-Funktion
```

'C'-Kurs

Wie bereits gesagt, wäre es nicht schlecht an dieser Stelle einen kompletten 'C'-Programmierkurs ein zu fügen. Auf diesem Bereich sind jedoch viele Handbücher im Handel und ausführliche Webseiten verfügbar, die 'C' sehr detailliert dokumentieren. Deshalb werden wir in diesem Manuskript nur die für den YETI benötigten 'C'-Begriffe und Funktionen beschreiben.

Nachdem wir die Grundbegriffe verstanden haben, fangen wir die Programmierung mal vorsichtig an. Wir werden dazu keinen kompletten 'C'-Programmierkurs anbieten, aber die auf der CD mitgelieferten Beispielprogramme beschreiben.

OK, es soll also kein 'C'-Programmierkurs werden, denn diese gibt es in Hülle und Fülle und Information zum 'C'-Programmieren gibt es auch in Büchern und im Internet. Unsere Information bildet jedoch einen guten Grundstock für eine Wiederholung des Wesentlichen und ein Leitfaden für Anfänger mit Praxisbeispielen.

WICHTIG

Der Buchstaben **v** am Anfang der YETI-Funktionen ist ein Hilfsmittel des Programmierers! Der Buchstaben **v** (**void**) am Anfang der YETI-Funktionen bedeutet, dass die Funktion keinen Wert zurückliefert.

YETI schaltet sein rechtes 'Auge' ein

```
#Dieses Programm schaltet das rechte YETI-'Auge' ein
#include "YETI.h"           //Lade Definitionen und Funktionen
int main(void){           //Beginn der main-Hauptfunktion
    vInitYETI();           //Initialisiere alle
                           //Mikrocontrollerfunktionen des YETIs
    vFrontLEDs(RIGHT);    //Schalte LED im rechten YETI-Auge ein
    return 0;              //Verlasse main-Hauptfunktion mit
                           //Rückgabe Wert 0
}                           //Ende der main-Hauptfunktion
```

#include "YETI.h"

An dieser Stelle wird die Datei 'YETI.h' eingefügt, welche die Definitionen der benötigten Funktionen enthält, z.B. die Funktion 'vYETIInit()'. Die vollständigen Funktionen befinden sich in der Datei 'YETI.c'.

vInitYETI();

Diese Spezialfunktion legt die Startbedingungen für alle Mikrocontrollerfunktionen des YETIs fest. Die Funktion wird als erster Programmschritt 'vInitYETI();' aufgerufen. Die tatsächlich auszuführende Funktionssteile befinden sich in der Funktionsdefinition in der Datei 'YETI.c'.

vFrontLEDs(LEFT);

Diese Funktion schaltet das linke YETI-‘Auge’ ein.

Gleichwertige Funktionen sind:

vFrontLEDs(RIGHT);

Diese Funktion schaltet das rechte YETI-‘Auge’ ein.

vFrontLEDs(OFF);

Diese Funktion schaltet beide YETI-‘Auge’ aus.

vFrontLEDs(BOTH);

Diese Funktion schaltet beide YETI-‘Auge’ ein.

Wir lassen das rechte YETI-Auge 5-Mal blinken

#Dieses Programm lässt das rechte YETI-Auge 5-Mal im Sekundentakt blinken

```
#include "YETI.h" //Lade Definitionen und
//Funktionen
int main(void){ //Beginn der main-Hauptfunktion
  int i; //Deklariere Variable 'i' als Integer
  vInitYETI(); //Initialisiere alle Mikrocontroller
//funktionen des YETIs
  for(i=0 ;i<5 ;i++){ //Wiederhole 5 Mal
    vFrontLEDs(RIGHT); //Schalte LED im rechten
//YETI-Auge ein
    vWaitMilliseconds(500); //Warte eine halbe Sekunde
    vFrontLEDs(OFF); //Schalte LED im rechten
//YETI-Auge aus
    vWaitMilliseconds(500); //Warte eine halbe Sekunde
  } //Ende der 5-maligen 'for'-Schleife
  return 0; //Verlasse main-Hauptfunktion
}
```

Der Programmteil innerhalb den geschweiften 'for'-Klammern '{' und '}' heißt 'Schleife' oder 'Loop'.

```
for(...){  
}
```

Int i;

An dieser Stelle wird eine Variable mit dem beliebigen Namen 'i' definiert.

```
for(i=0 ;i<5 ;i++){
```

Definiere zuerst 'i' als Zahl 0 und wiederhole alle Programmteile zwischen der geschweiften 'for'-Öffnungsklammer '{' und der zugehörigen 'for'-Abschlussklammer '}', solange Variable 'i' kleiner ist als 5. Dabei wird jedes Mal, wenn das Programm an der 'for'-Zeile vorbeikommt, die Variable 'i' mit 1 erhöht. Beim ersten Vorbeikommen ist dieser Wert daher 0.

Beim 6en Mal hat 'i' den Wert 5 erreicht und ist somit nicht mehr kleiner als 5. Das Programm wird daher keine weitere 'for'-iSchleife durchlaufen und seinen Weg mit der erstfolgenden Zeile nach der 'for'-Abschlussklammer verfolgen, d.h. mit der Zeile return 0;

Die Schleife wurde unserem Wunsch entsprechend insgesamt 5 Mal ausgeführt.

vWaitMilliseconds(500);

Diese Zeile ruft die Funktion vWaitMilliSeconds mit dem Übergabewert 500 auf.

Die Funktion befindet sich in der Datei 'YETI.c'. Sie können die Datei 'YETI.c' mit dem Programmers Notepad2 und die Funktionsbeschreibung durchlesen. Die Funktion verwendet für die Warteschleife einen internen Zähler des Mikrocontrollers und kehrt anschließend aus der Funktion zurück zum Programm.

Als Eingabewert akzeptiert die Funktion Zeitangaben in Millisekunden. Das sind Tausendstel Sekunden. 500 Tausendstel Sekunden entsprechen daher einer halben Sekunde.

Der YETI soll Geräusche produzieren

#Dieses Programm lässt YETI Geräusche produzieren

```
#include "YETI.h"           //Lade Definitionen und Funktionen
int main(void){           //Beginn der main-Hauptfunktion
    vInitYETI();           //Initialisiere alle Mikrocontroller-
                           //funktionen des YETIs
    vBeep(400,80);         //lässt YETI einen Piepston produzieren
    vBeep(520,120);       //lässt YETI einen Piepston produzieren
    vBeep(360,80);        //lässt YETI einen Piepston produzieren
    vBeep(580,160);       //lässt YETI einen Piepston produzieren
    return 0;              //Verlasse main-Hauptfunktion
}                           //Ende der main-Hauptfunktion
```

vBeep(400,80);

Aufruf der Funktion 'vBeep' mit einer Tonhöhe 400 und mit einer Tondauer von 80 Millisekunden.

Der YETI soll tänzeln

#Dieses Programm veranlasst YETI, seine Füße in einem Tanz zu bewegen.

```
#include "YETI.h"           //Lade Definitionen und Funktionen
#include "YETImove.c"       //Einfügen der YETI-Bewegungs-
funktionen
int main(void){            //Beginn der main-Hauptfunktion
    int i;                 //Deklariere Variable 'i' als Integer
    vInitYETI();          //Initialisiere alle Mikrocontroller-
                           //funktionen des YETIs
    for(i=0 ;i<3 ;i++){    //Wiederhole 3 Mal
        vMoveBody(16,10);  //Bewege den Körper nach links
        vMoveBody(-16,10); //Bewege den Körper nach rechts
    }                      //Ende der 3-maligen 'for'-Schleife
    vMoveBody(0,10);       //Bewege den Körper in die Ruheposition
    return 0;              //Verlasse main-Hauptfunktion
}                          //Ende der main-Hauptfunktion
```

#include "YETImove.c"

Einfügen der Datei 'YETImove.c' mit den YETI-Bewegungsfunktionen.

vMoveBody(16,10);

Bewege den Server für die YETI-Körperbewegungen um 16 Schritte nach links und warte bei jedem Schritt 10 Millisekunden.

vMoveBody(-16,10);

Bewege den Server für die YETI-Körperbewegungen um 16 Schritte nach rechts und warte bei jedem Schritt 10 Millisekunden.

Sie können die Schrittzahl für die YETI-Körperbewegungen zwischen -58 und +58 wählen. Die Wartezeiten werden immer auf ein Vielfach von 10 aufgerundet. So wird für jede Zahl zwischen 1-10 Millisekunden eine Wartezeit von 10 Millisekunden und für jede Zahl zwischen 11-20 Millisekunden eine Wartezeit von 20 Millisekunden angehalten.

vMoveBody(0,10);

Bringe den YETIkörper wieder in die Ruheposition.

Mit einer artverwandten Funktion 'vMoveLegs()' können Sie die YETI-beine nach vorne oder nach hinten bewegen.

YETI soll laufen

#Dieses Programm lässt YETI drei Schritte laufen.

```
#include "YETI.h"           //Lade Definitionen und Funktionen
#include "YETImove.c"       //Einfügen der YETI-Bewegungs-
funktionen
int main(void){            //Beginn der main-Hauptfunktion
    vInitYETI();           //Initialisiere alle Mikrocontrol-
                           //lerfunktionen des YETIs
    vMoveForwardXSteps(3); //Bewege den Körper 3 Schritte vorwärts
    vStandUpright();       //Zurück zur Ruheposition
    return 0;              //Verlasse main-Hauptfunktion
}                           //Ende der main-Hauptfunktion
```

vMoveForwardXSteps(3);

YETI bewegt sich drei Schritte nach vorne. Als Anfangsschritt wird YETI immer mit dem rechten Fuß beginnen.

vStandUpright();

YETI bewegt seinen Körper zurück in die Ruheposition

14. ERWEITERUNGEN

Erweiterungsmodule (Kits)

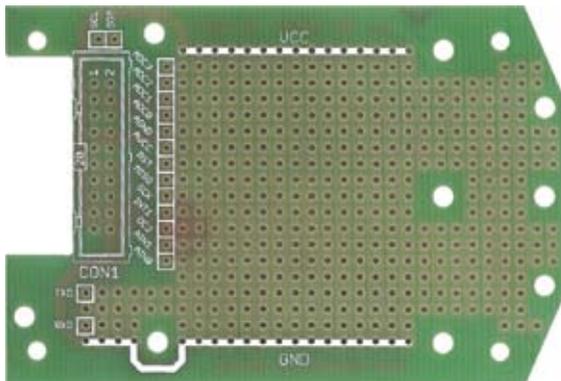
Sie können YETI erweitern mit zusätzlichen (nicht im Lieferumfang enthaltenen) Baugruppen, welche die Leistungsfähigkeit des Roboters erheblich steigern. So können Sie YETI mit einem Ultraschall-Sender/ Empfänger ausstatten, der ihm ermöglicht mit Schallwellenechos die Distanz zu entfernten Gegenstände zu messen und diese beim Gehen auszuweichen.

14.1.2. YETI Experimentiermodul YT-EXP1

Im Experimentiermodul können Sie einen eigenen Entwurf einer Elektronikbaugruppe aufbauen und dieses Modul auf unterschiedlichster Art aus dem Mikrocontroller ansteuern.

14.1.2. Teilleiste Experimentiermodul YT-EXP1

PCB-DSP		YETI Experimentier platine
CON1-PCB		Platine konnektor, male, 20 pins, für flachband
CON1-FC	(2 St.)	Flatchband Konnektor, 20-pins
F1		Flachbandkabels, 20-Adern, 10cm



14.2. YETI Display Erweiterungsmodul YT-DSP2

Allgemeine Beschreibung

Das Display Erweiterungsmodul enthält 4 Stück 8-Segmentanzeigeblöcke. Ein ebenfalls im Erweiterungsmodul anwesender 24-pins I2C driver chip versorgt die Ansteuerung des Displays.

Der YETI Mikrocontroller steuert seinerseits den I2C Treiberchip. I2C (buchstabiert als I-Quadrat-C) ist eine offizielle Standardmethode für die gegenseitige Kommunikation zwischen Elektronikmodule mit nur zwei Anschlussadern SCL (Serial Clock) und SDA (Serial Data). Wir benötigen für dieses Kommunikationsprotokoll deshalb nur 2 der 20 Adern des Flachbandkabels.

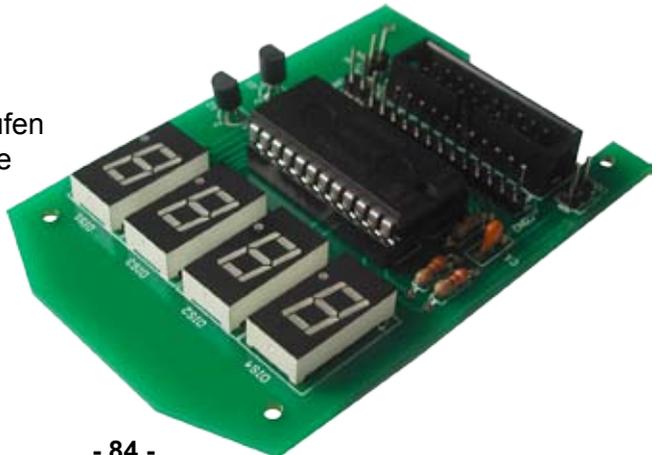
Das Display bietet eine Anzeigeoption für diverse Nachrichten oder Daten. Die einfache Ansteuerung des I2C Chips erlaubt die Darstellung vielerlei selbstdefinierten Symbole auf dem Display.

Hardware Beschreibung

Das Displaymodul enthält vier 8-Segment Displays, einen I2C Displaychip und einige Hilfskomponenten.

Der Chip kommuniziert über dem I2C-Interface mit einem Steuerungsmodul, in der Regel einem Mikrocontroller. Der Chip liefert selbstständig alle Steuerungssignale zur Ansteuerung der Displays. Der Anwender braucht sich um diese Details nicht mehr zu kümmern und muss nur festlegen welches Symbol in jedem einzelnen Zeichen angezeigt werden soll. Dazu wird selbstverständlich der I2C-Bus verwendet.

Auch die Helligkeit der Anzeige lässt sich in 8 Stufen regeln. Zudem verfügt jede Displayeinheit über 8 Anschlüsse für die Einzelsegmente: sieben Datensignale für die Einzelsegmente und ein Dezimalpunkt.



Jedes Segment besteht aus einer Leuchtdiode oder LED. Außerdem wird noch ein gemeinsamer Pol für den Stromversorgungsanschluß der Leuchtdioden in jedem Displayelement benötigt. Der Hersteller hat dazu vier Segmente jeweils an einem Anschluss verknüpft und die Anschlüsse 3 und 14 intern zu einen gemeinsamen Stromversorgungsanschluß zusammengelegt.

Zur Ansteuerung der vier Displays mit jeweils $8+2=10$ Anschlüssen würde man theoretisch ein 40-poliges IC benötigen. Diese Verbindungsfülle können wir jedoch mit dem Multiplexertrick beseitigen. Der Chip benötigt dazu 2 Sätze 8-Segment-Anschlüsse: P1-P8 und P9-P16.

Zunächst betrachten wir den Satz P1-P8, die sowohl mit Display 1 als mit Display 2 verbunden sind. Falls wir nun eine bestimmten Bitkombination auf P1-P8 schalten, dann werden beide angeschlossenen Displays das gleiche Symbol anzeigen. Der Trick besteht nun darin, mit den Transistoren Q1 und Q2 Display 1 ein- und Display 2 abzuschalten. Mit dieser Schaltung wird zuerst nur Display 1 das Symbol anzeigen. Nun jedoch schalten wir Display 1 mit Transistor Q2 aus, erzeugen auf den Datenleitungen P1-P8 eine neue Bitkombination für ein weiteres Symbol und schalten Display 2 mit Transistor Q1 ein. Das zweite Symbol wird nun auf Display 2 angezeigt. Dieses Multiplexerprinzip wird auch für Display 3 und 4 angewandt. Falls wir nun den Multiplexertrick mit höheren Schaltgeschwindigkeit betreiben, wird das menschliche Auge nicht registrieren, daß die Displays eigentlich die Hälfte der Zeit gar nicht aufleuchten.

Wir können zwei der vier Displays auch ohne Multiplexermodul ansteuern. Diese beiden Displays sind ständig eingeschaltet, falls ein Display (z.B. Display 1) auf P1-P8 und das zweite Display (Display 3) auf P9-P16 angeschlossen wird. Die Beispielsoftware basiert auf die Randbedingung, daß die Displays folgendermaßen geschaltet werden:

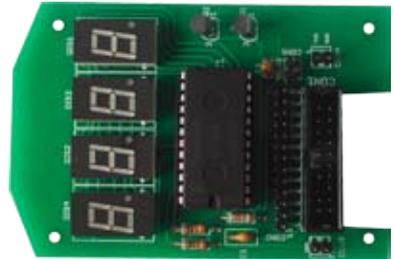
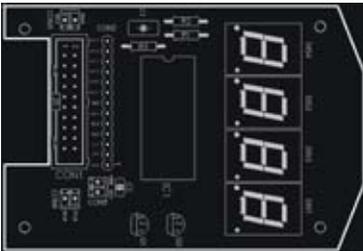
Displayschaltung:

Display 4 Display 2 Display 3 Display 1

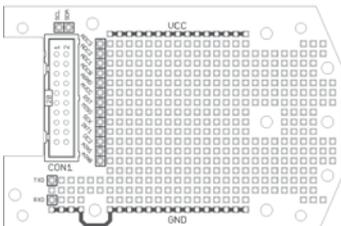
Diese Schaltung basiert auf die grundlegende Idee, daß zum Betrieb zweier Displays (Display 1 und 3) im statischen Modus diese Module nebeneinander stehen sollten..

14.2.1. Teilleiste YETI Displaysatz YT-DSP2 KIT

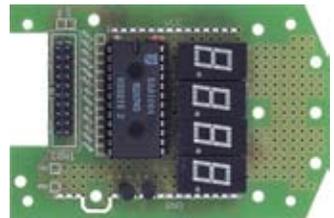
PCB-DSP	YETI Display platine
R1	330R (orange, orange, braun, gold)
R2	330R (orange, orange, braun, gold)
R3	18K (braun, grau, orange, gold)
C1	2,7nF (Aufdruck: 272)
C2	100nF (Aufdruck: 104)
Q1	BC547B/C (auf richtige Polung achten)
Q2	BC547B/C (auf richtige Polung achten)
D1	8-Segment display, common anode
D2	8-Segment display, common anode
D3	8-Segment display, common anode
D4	8-Segment display, common anode
IC1	SAA1064 (auf richtige Polung achten)
S1	IC-Fuß, 24-pins, 600mil (auf richtige Polung achten)
CON2	Kontaktleiste 13 -polig, platine montage
CON, 3, 4 und 5 (3. St.)	Kontaktleiste 2 -polig, platine montage
CON1-PCB	Platine konektor, male, 20 polig, für flachband
CON1-FC (2 St.)	Flatchband Konnektor, 20-polig
F1	Flachbandkabels, 20-Adern, 10cm



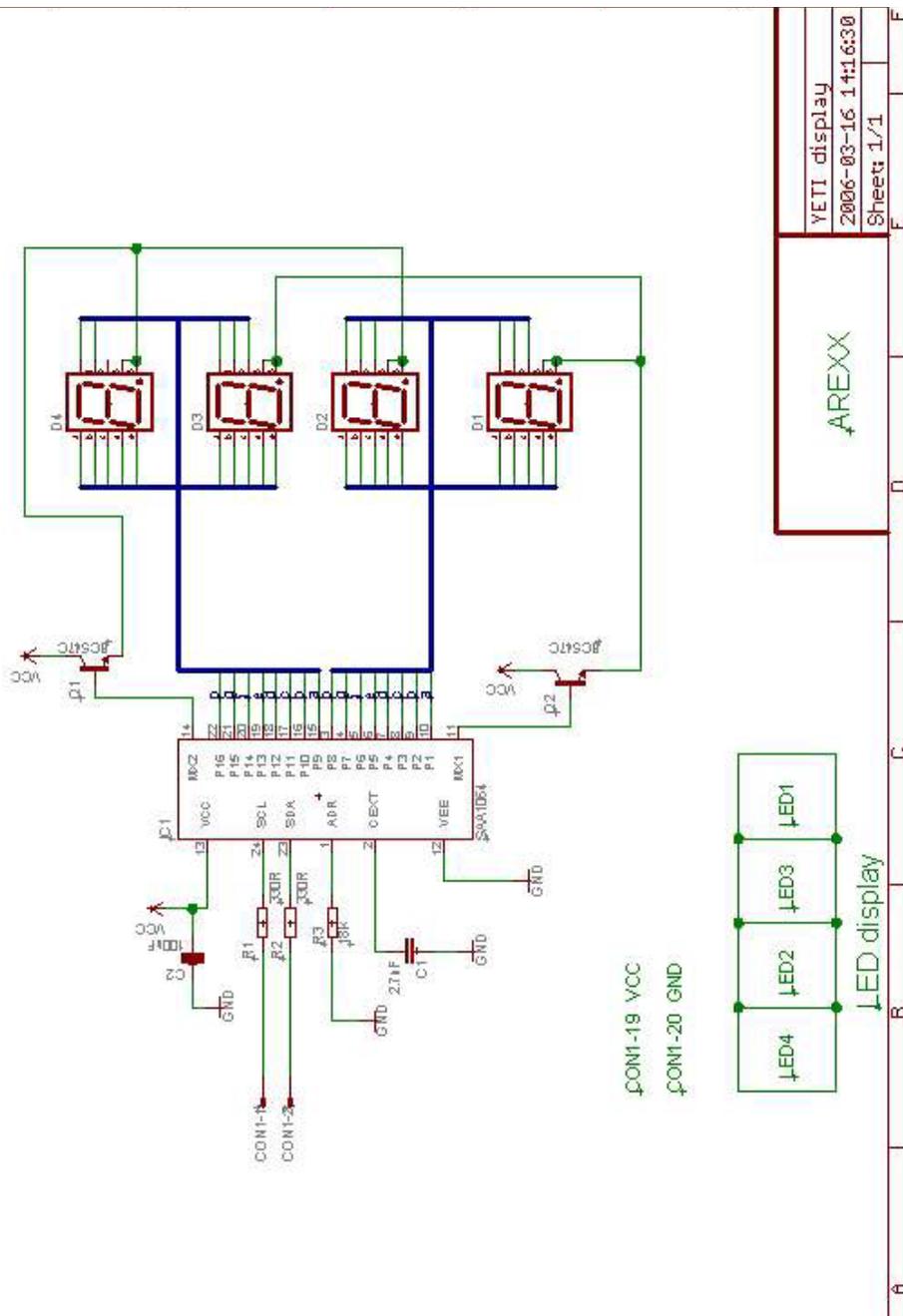
Display auf original Display Platine YT-DSP2 KIT



Display auf Experimentierplatine



14.2.2. Display Modul Schaltplan



14.3. YETI Ultraschall Erweiterungsmodul YT-ULT3

Der Ultraschall Erweiterungssatz enthält einen Ultraschallsender und einen Ultraschallempfänger. Ultraschall bedeutet: Schallwellen mit einer so hohen Frequenz, dass Menschen diesen Ton nicht hören können.

Fledermäuse senden zum Beispiel Ultraschallwellen aus, damit sie im Dunkeln fliegen und jagen können. Man nennt dieses Verfahren Echo-Ortung. Gegenstände reflektieren die Schallwellen, die dann im Ohr empfangen werden.

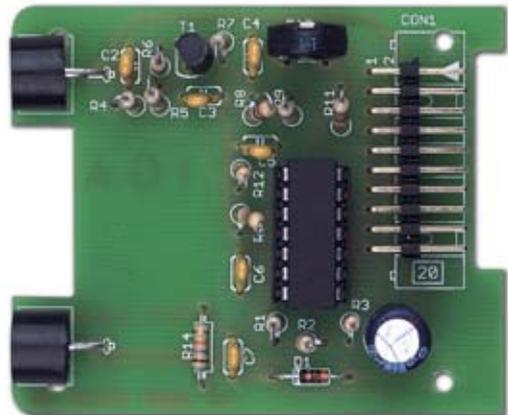
Beim YETI wird ein Mikrofon zum Schallwellenempfang benutzt.

Der Sender erzeugt Schallwellen mit einer Frequenz von 40.000Hz. Der Empfänger registriert die eventuell von einem Gegenstand reflektierten Signale und auch die Zeitverzögerung zwischen Sende- und Empfangsimpuls.

Aus der Zeitverzögerung zwischen Sende- und Empfangsimpuls können wir den Abstand zum Gegenstand berechnen. Das Ultraschallmodul verwandelt die Zeitverzögerung in eine elektrische Spannung. Das Flachbandkabel führt diese elektrische Spannung zur Analog/Digitalwandlereingang des Mikroprozessors im YETI.

Ein im YETI verfügbares Programm kann die Spannung messen und aufgrund der Daten eine Aktion ausführen.

Im YETI ist zur Aufnahme des Ultraschallmoduls bereits ein Platz am Deckel zum Innenraum des Kopfes reserviert. Hinter den beiden "Augenbrauen"-Öffnungen in der Stirn befinden sich der Sender, beziehungsweise Empfänger.



14.3.1. Hardware Beschreibung

Das Ultraschallmodul besteht aus 5 Komponenten:

1. Der Sender
2. Der Empfänger
3. Der Empfängerverstärker
4. Eine konstante Referenzspannung
5. Eine variable Referenzspannung

Der Mikrocontroller generiert das benötigte Ultraschallsignal. Ein Senderlautsprecher (TX) erzeugt das akustische Signal und ein Empfängermikrofon (RX) empfängt die reflektierte Schallwelle. Der Empfängerverstärker erhöht den schwachen Empfangspegel, wobei der Verstärkungsfaktor manuell mit R10 einstellbar ist. Die konstante Referenzspannung liefert eine Spannung, die auf genau 50% der Stromversorgungsspannung eingestellt und zur Ansteuerung des Senders und zur Erzeugung der variablen Referenzspannung benötigt wird. Ein spezieller Regelkreis läßt mit Hilfe der variablen Referenzspannung den Mikrocontroller je nach Entfernung der Reflexionsquelle nach jeder Schallwelle immer genauer zuhören. Diese Regelung ist sehr wichtig, denn bei großer Entfernung der Reflexionsquellen nimmt der Empfangspegel rapide ab.

Der Mikrocontroller erzeugt die benötigten Ultraschallwellen und führt diese Signale über CON1-13 in das Ultraschallmodul. Das reflektierte und im Ultraschallmikrofon empfangene Signal wird über CON1-6 in den Mikrocontroller zurückgeführt. Anschluß CON1-15 liefert nach jedem Senderimpuls eine abnehmende Spannungssignal an den Mikrocontroller.

Das im Mikrocontroller erzeugte 'Ultraschall'-Signal ist zunächst natürlich keine Schallwelle, sondern ein elektrisches Signal. Genau genommen darf man diese Signale erst 'Ultraschallwellen' nennen, nachdem der Lautsprecher das Signal in Schallwellen verwandelt hat.

Dieses im Mikrocontroller generierte 'Ultraschall'-Signal wird zunächst über CON1-13 und über Widerstand R3 in den Sender eingespeist. Der Sender besteht aus 2 (von 4) Verstärker, die einzelnes IC (Integrated Circuit = Chip) mit dem Namen IC1 zur Verfügung stellt. Fachleute nennen diese Verstärker oft Opamp, d.h. Operational Amplifier oder Operationsverstärker. Es handelt sich dabei um Differenzverstärker mit zwei Eingängen: einen Plus- und einen Minuseingang. Wie der Name Differenzverstärker andeutet, verstärkt das Verstärkermodul die Differenzspannung zwischen den beiden Eingängen.

Das Ultraschallsignal erreicht nun einen Eingang eines der beiden Opamps. Die übrigen Eingänge der beiden Opamps werden mit einer festen Referenzspannung in Höhe von 50% der Stromversorgungsspannung verbunden. Beide Opamps haben zur Aufgabe den Lautsprecher mit Energie zu versorgen und das vom Mikrocontroller geringfügig geschwächt und verzerrt aufbereitete Signal zu regenerieren. Das Ultraschallmikrofon (RX) empfängt das reflektierte Signal und wandelt es in ein elektrisches Signal. Nach einem einfachen Filter erreicht das Signal den Empfangsverstärker Opamp IC1B, dessen Verstärkungsfaktor mit einem Regelwiderstand einstellbar ist.

Nach jedem Senderimpuls steigt die Spannung auf CON1-15 sprunghaft auf 50% der Stromversorgungsspannung VCC. Dazu wird das elektrische 'Ultraschallsignal' nicht nur über R3 an den Verstärker, aber auch über Diode D1 an Kondensator C7 weitergeleitet. Nachdem diese Spannung C7 zunächst sofort aufgeladen hat, entlädt der Kondensator sich langsam über Widerstand R14.

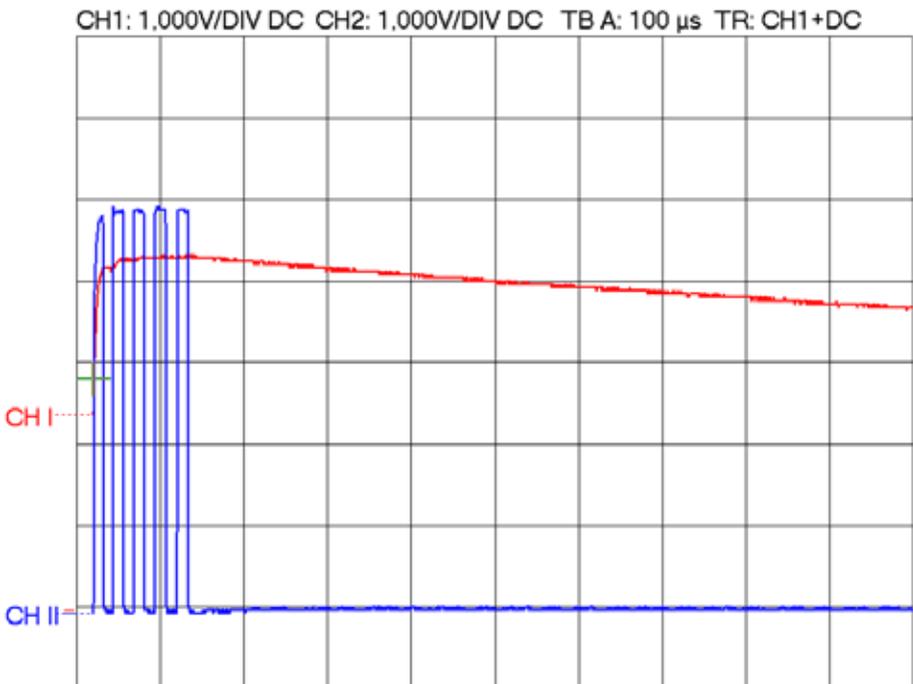
Der Mikrocontroller enthält einen analogen Komparator, d.h. einen Vergleichsmodul, das zwei elektrische Spannungspegel vergleicht. Sowohl der Empfangspegel als diese Entladungsspannung werden nun auf CON1-6 beziehungsweise CON1-15 dem analogen Komparator zum Vergleich angeboten. Der Mikrocontroller vergleicht den Empfangspegel und die Entladungsspannung. Falls der Empfangspegel die Entladungsspannung übersteigt, betrachtet der Mikrocontroller den Empfangspegel als ein zuverlässiges Empfangssignal. Unmittelbar nach dem Senderimpuls muß ein Empfangspegel daher schon relativ hoch sein, um als reflektiertes Signal akzeptiert zu werden. Je länger der Senderimpuls jedoch zurückliegt, desto geringer darf der Empfangspegel sein, um als reflektiertes Signal akzeptiert zu werden.

14.3.2. Abgleich Ultraschall-Zusatzmodul

Die Ultraschallplatine ist in YETI's Kopf eingebaut. Allerdings wird das Ultraschallmodul in dieser Position durch folgende Dinge negativ beeinflusst:

1. Unerwünschte Reflektionen des Ultraschalltons innerhalb des YETI Chassis.
2. Unerwünschte Reflektionen des Ultraschalltons durch die Aussenhülle von YETI.

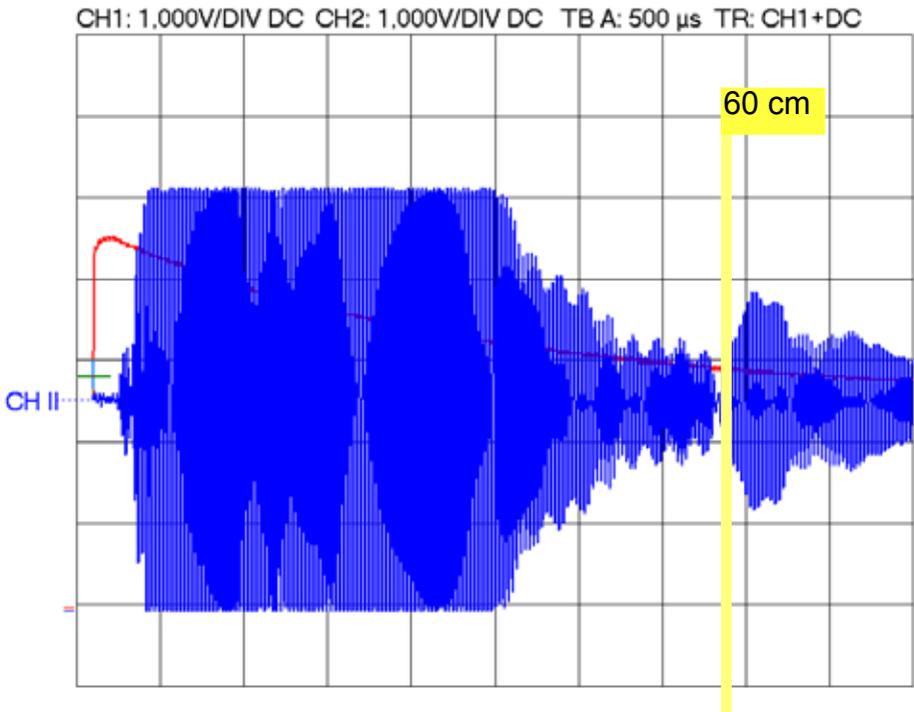
In folgender Grafik sehen Sie ein Oszillogramm der Referenzspannung (rot, gemessen an CON1-15) und des Ultraschallsignals des Senders (blau, CON1-13, 5 Impulse von etwa 4.5V, vom Mikrocontroller erzeugt):



Das erste Signal (CON1-13) zeigt kurz 5 Pulse von etwa 4.5 Volt.

Das Signal kommt sofort vom Mikroprozessor und geht zu der Ultraschallplatine.

Die rote Kurve an CON1-15 ist eine Entladekurve, welche als Referenzsignal dient. Diese wird auf der Ultraschallplatine erzeugt und zurück zum Mikrocontroller geleitet. Erzeugt der Empfängerteil des Ultraschallsensors eine Spannung die höher ist als das Referenzsignal, wird dies vom Mikrocontroller als Reflektion an einem Objekt erkannt.



Nun eine weitere Messung, diesmal jedoch neben der Referenzspannung (rot) auch das empfangene Echo (blau, CON1-6 - wird zum Mikrocontroller geleitet).

Wie man leicht erkennen kann gibt es zu Beginn eine sehr starke Fehlmessung! Das Objekt welches das Signal eigentlich reflektieren sollte befindet sich bei der Markierung in 60 cm Entfernung. Dies wird durch Schallreflektionen innerhalb von YETI verursacht.

Im Folgenden werden wir beschreiben wie man dies stark verbessern kann!

Wie man aus der Beschreibung des Messverfahrens unseres Ultraschallsensors herauslesen kann, reagiert der Empfänger sehr empfindlich auf jegliche Arten von Reflektion des Ultraschallsignals. So auch auf Reflektionen innerhalb von YETI!

Um diesen zu verringern, müssen wir das Innere von YETI unterhalb des Ultraschallsensors vollständig mit Watte auffüllen (s. Abbildung 1)!

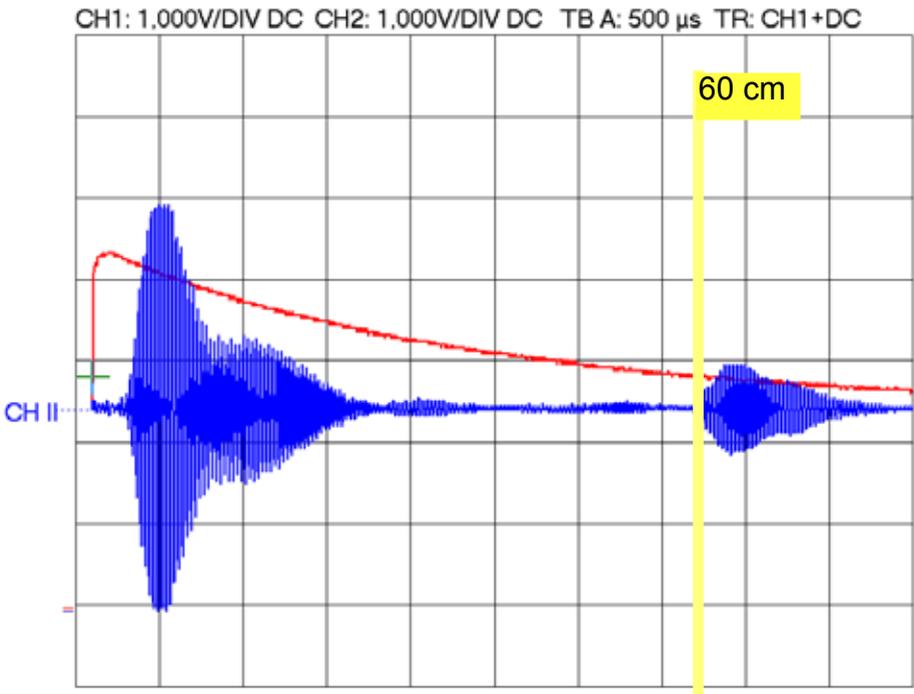
Abbildung 1. YETI vollständig mit Watte ausgefüllt.



Schritt 1.

Abdämmen der unerwünschten Schallreflektionen innerhalb von YETI.

Folgendes Oszillogramm zeigt die neue Situation (YETI vollständig mit Watte aufgefüllt):



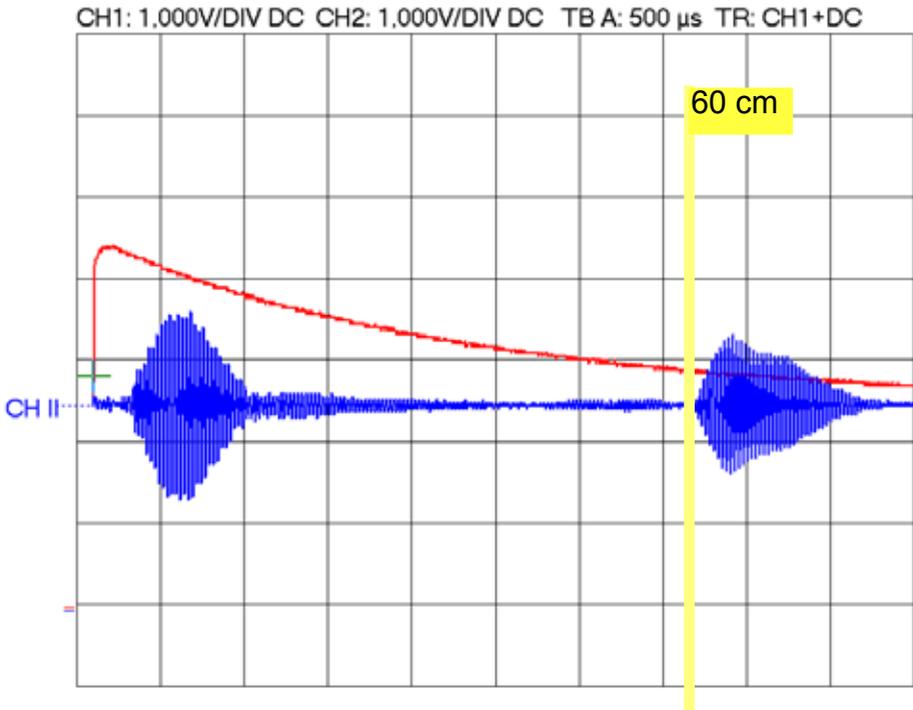
Es ist leicht zu erkennen das es nun deutlich geringere unerwünschte Reflektionen gibt!

Doch noch immer gibt es einen kleinen Bereich zu Beginn der Messung, der höher als das Referenzsignal liegt. Dies wird durch Reflektionen direkt vor YETIs Kopf verursacht.



Schritt 2.

Dies lässt sich mit einem weiteren Stück Watte (oder ähnlichem, evtl. etwas festeres Material verwenden!) vermeiden, welches so ausserhalb von YETI angebracht wird, wie es auf dem nebenstehenden Bild zu sehen ist!



Auf dem Oszillogramm erkennt man das Ergebnis unserer Bemühungen: Alle zu Beginn der Messung empfangenen Signale bleiben unterhalb der Referenzspannung! Erst das tatsächlich zu detektierende Objekt erzeugt eine höhere Spannung am Empfänger Ausgang!

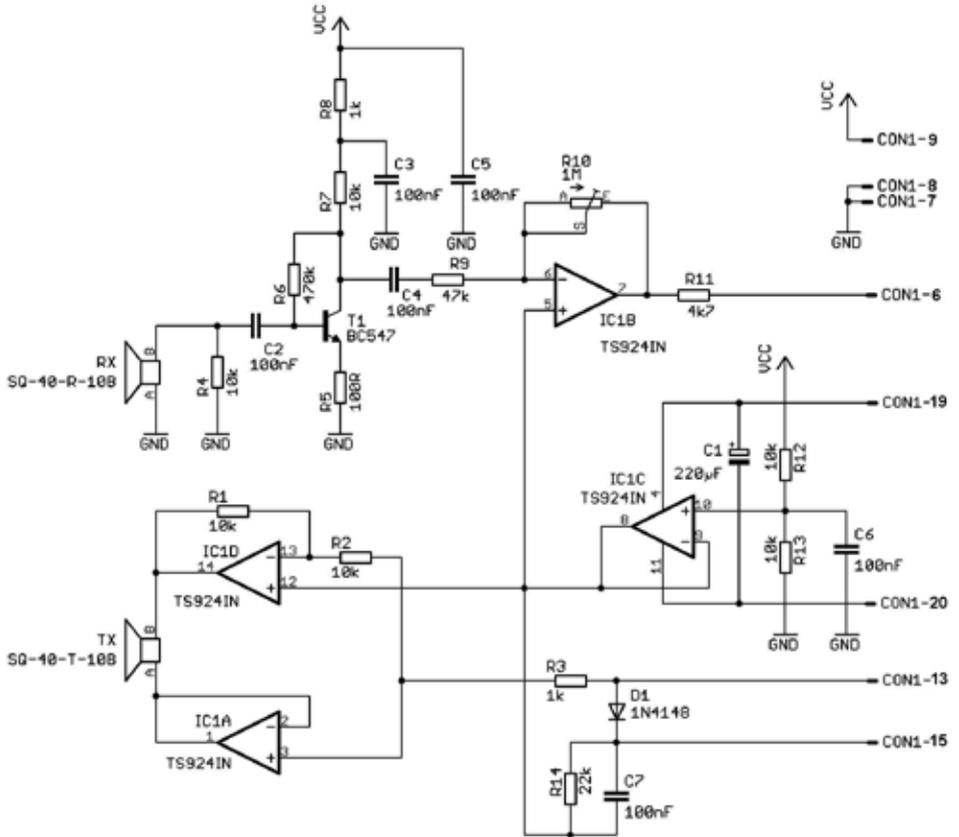
Die Zeitspanne zwischen den 5 übertragenen Ultraschallpulsen und dem Reflektierten Signal wird vom Mikrocontroller gemessen und in den Abstand umgerechnet. Diesen Abstand kann man dann z.B. auf dem YETI Display anzeigen!

Es ist sehr wichtig, dass Sie beim Abgleich dafür sorgen das die Reflektion zu Beginn der Messung unterhalb der Referenzspannung bleiben.

14.3.3. Teileliste YETI Ultrasschallsatz YT-ULT3

PCB-UTS		Ultraschall Platine
R1		10K (braun, schwarz, orange, gold)
R2		10K (braun, schwarz, orange, gold)
R3		1K (braun, schwarz, rot, gold)
R4		10K (braun, schwarz, orange, gold)
R5		100R (braun, schwarz, braun, gold)
R6		470K (gelb, violett, orange, gold)
R7		10K (braun, schwarz, orange, gold)
R8		1K (braun, schwarz, rot, gold)
R9		47K (gelb, violett, rot, gold)
R10 TRIMMER		1M Trimmer
R11		4K7 (gelb, violett, rot, gold)
R12		10K (braun, schwarz, orange, gold)
R13		10K (braun, schwarz, orange, gold)
R14		22K (rot, rot, orange, gold)
C1		220uF (auf richtige Polung achten)
C2		100nF (Aufdruck: 104)
C3		100nF (Aufdruck: 104)
C4		100nF (Aufdruck: 104)
C5		100nF (Aufdruck: 104)
C6		100nF (Aufdruck: 104)
C7		100nF (Aufdruck: 104)
IC1		TS924IN (Quad Opamp) (auf richtige Polung achten)
S1		IC-socket, 14-pins (auf richtige Polung achten)
TX		400ST100 (Ultraschall transmitter)
RX		400SR100 (Ultraschall receiver)
T1		BC547B oder BC547C (auf richtige Polung achten)
D1		1N4148 (auf richtige Polung achten)
CON1-PCB		PCB Konnektor, male, 20 pins, für Flachbandkabel
CON1-FC	(2. St.)	Flachbandkabel Konnektor, 20-pins
F1		Flachbandkabel, 20-adrigen, etwa 10cm

14.3.4. Schaltbild YETI Ultraschallsatz



Zum Abschluss

Wir hoffen, dass unsere Roboter ASURO und YETI Ihnen auf den Weg in die Roboterwelt geholfen haben! Wie unsere japanischen Freunde glauben auch wir, dass Roboter nach den Computern und Mobiltelefonen die nächste technologische Revolution bilden werden. Diese Revolution wird auch neue wirtschaftliche Impulse auslösen. Leider haben Japan, andere asiatische Länder und auch die USA, Europa dabei längst überholt. Im Gegensatz zu Europa beginnt der Technikunterricht in Asien bereits in der Grundschule und ist ein wichtiger Bestandteil der Ausbildung. Als Zielsetzung bei der Entwicklung der Roboter ASURO und YETI haben wir deshalb gewählt:

TO TRAIN A SCIENTIFIC MIND



APPENDIX

A. ÜBERSICHT YETI-FUNKTIONEN

Basisfunktionen.

Nachfolgende Funktionen gehören zu den Basisfunktionen des YETIs und befinden sich in 'YETI.c'.

vInitYETI()

Initialisieren des YETIs.

vFrontLEDs(x)

x = ON, LEFT, RIGHT, OFF

Steuerung der LED im YETI-Auge.

Beispiel:

```
vFrontLEDs(LEFT);
```

vServo1ToPosition(x)

x = 0-65635

Platziere den Front(körper)servomotor in Stand x.

WARNUNG: Diese Funktion kann den Servomotor in eine beliebige Position bewegen. Dabei kann der Motor auch Positionen ansteuern, die in der YETImechanik gar nicht vorgesehen oder gar erreichbar sind. Diese Kommandos können den YETI ggf. auch beschädigen. Verwenden Sie stattdessen deshalb vorzugsweise die sichere Funktion 'vMoveBody()'.

Beispiel:

```
vServo1ToPosition(35);
```

vServo2ToPosition(x)

x = 0-65635

Platziere den Boden(Beine)servomotor in Stand x.

WARNUNG: Diese Funktion kann den Servomotor in eine beliebige Position bewegen. Dabei kann der Motor auch Positionen ansteuern, die in der YETImechanik gar nicht vorgesehen oder gar erreichbar sind. Diese Kommandos können den YETI ggf. auch beschädigen. Verwenden Sie stattdessen deshalb vorzugsweise die sichere Funktion 'vMoveLegs ()'.

Beispiel:

```
vServo2ToPosition(35);
```

vRs232Write(x,y)

x = Text

y = Textlänge, 0-255

Beispiel:

```
void vRs232Write ("Hello World",11) ;
```

vRs232Read(x,y,z)

x = Zeiger (Englisch: Pointer) auf ein Array zur Speicherung des empfangenen Textes

y = Erwartete Zeichenzahl

z = timeout

Beispiel:

```
char RxData[10] ;
```

```
vRs232Read(&RxData[0],4,0);
```

Sobald 4 Zeichen eingegangen sind, wird die Funktion beendet.

vWaitMilliseconds(x)

x = Wartezeit in Millisekunden, 0 - 65535

Die Funktion rundet in Zehnerschritten auf, d.h. 23 wird zu 30 aufgerundet.

Beispiel: warte 1 Sekunde.

```
vWaitMilliseconds(1000);
```

vBeep(x,y)

x = Tonhöhe, 1-3906

y = Tondauer in Millisekunden.

Beispiel:

```
vBeep(200,100);
```

Abgeleitete Funktionen.

Untengenannte Funktionen sind abgeleitete Funktionen, die eine oder mehrere Basisfunktionen enthalten und sich in der Datei 'YETImove.c' befinden.

vStandUpright()

bewegt YETI in seine Ruheposition. Sowohl der 'body'-Servomotor als auch der 'legs'-Servomotor bewegen sich in die Ruheposition.

vMoveBody(x,y)

x = YETI Körperposition nach links oder rechts neigend, ab -58 (Neigung nach rechts) bis einschließlich +58 (Neigung nach links).

y = Bewegungstempoverzögerung in Millisekunden pro Schritt, 0 – 65635.

Die Funktion rundet die Verzögerungszeiten in Zehnerschritten auf, d.h. 17 wird zu 20.

Beispiel:

```
vMoveBody(-25,20);
```

Falls YETI aufrecht steht, wird vMoveBody(-25,20) etwa $25 \cdot 20 = 500 \text{ms} = 0,5 \text{ Sekunde}$ benötigen um YETI in eine Rechtsneigung mit Position -25 zu bewegen.

vMoveLegs(x,y)

x = YETI Beinposition rechts nach vorne oder nach hinten bewegen, ab -58 (Rechtes Bein nach vorne) bis einschließlich +58 (Linkes Bein nach vorne). Das linke Bein bewegt sich immer entgegengesetzt.

y = Bewegungstempoverzögerung in Millisekunden pro Schritt, 0 – 65635.

Die Funktion rundet die Verzögerungszeiten in Zehnerschritten auf, d.h. 17 wird zu 20.

Beispiel:

```
vMoveLegs(-25,20);
```

Falls YETI aufrecht steht, wird vMoveLegs(-25,20) etwa $25 \cdot 20 = 500 \text{ms} = 0,5 \text{ Sekunde}$ benötigen um YETIs rechte Bein nach vorne in Stand -25 und das linke Bein entgegengesetzt nach hinten zu bewegen.

vMoveForwardXSteps(x)

x = Anzahl Vorwärtsschritte, 0-255

Beispiel:

```
vMoveForwardXSteps(3);
```

YETI macht 3 Schritte vorwärts. Aus der Ruheposition beginnt YETI immer mit dem rechten Bein.

vMoveBackwardXSteps(x)

x = Anzahl Rückwärtsschritte, 0-255

Beispiel:

```
vMoveBackwardXSteps(4);
```

YETI macht 3 Schritte rückwärts. Aus der Ruheposition beginnt YETI immer mit dem rechten Bein.

vTurnLeftXSteps(x,y)

x = Anzahl Schritte, 0-255

y = 'true' oder 'false', true = Vorwärtsgang, false = Rückwärtsgang.

Beispiel:

```
vTurnLeftXSteps(2,false);
```

YETI dreht sich im Rückwärtsgang 2 Schritte linksherum.

vTurnRightXSteps(x,y)

x = Anzahl Schritte, 0-255

y = 'true' oder 'false', true = Vorwärtsgang, false = Rückwärtsgang

Beispiel:

```
vTurnLeftXSteps(3,true);
```

YETI dreht sich im Vorwärtsgang 3 Schritte rechtsherum.

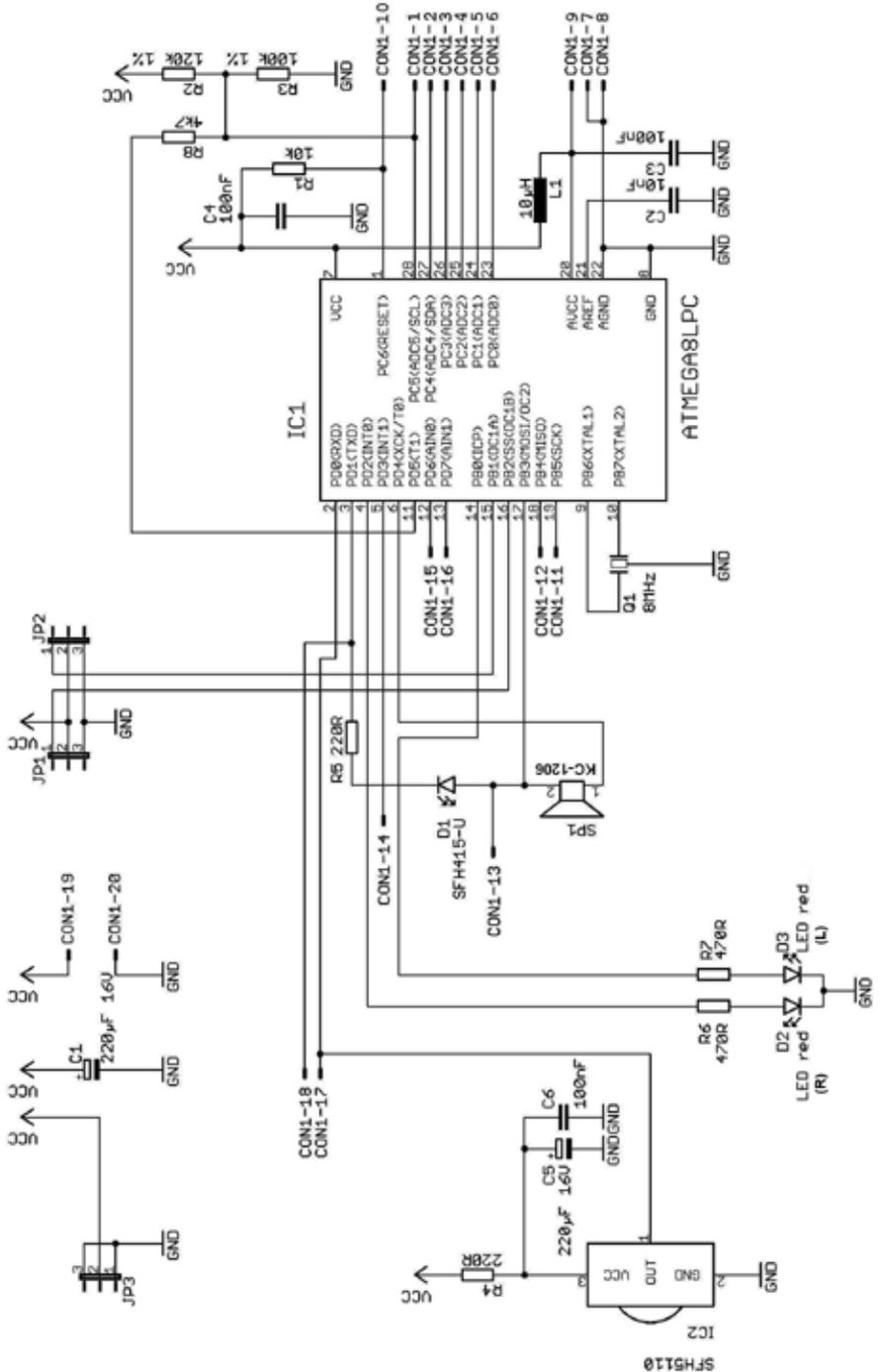
vCalibrateServos()

Wir werden diese Funktion (in Kombination mit dem kostenlosen Kommunikationsprogramm Hyperterminal) jedesmal benötigen, wenn wir die YETI-Servomotoren oder YETI-Servoglieder neu befestigt haben. Die Funktion übernimmt dann die genaue Kalibrierung des YETI-Servosystems. Man kann die Funktion auch benutzen, um eine Fernbedienung mit dem Hyperterminal und mit den Tastaturtasten 'W', 'A', 'S' und 'D' auf zu bauen.

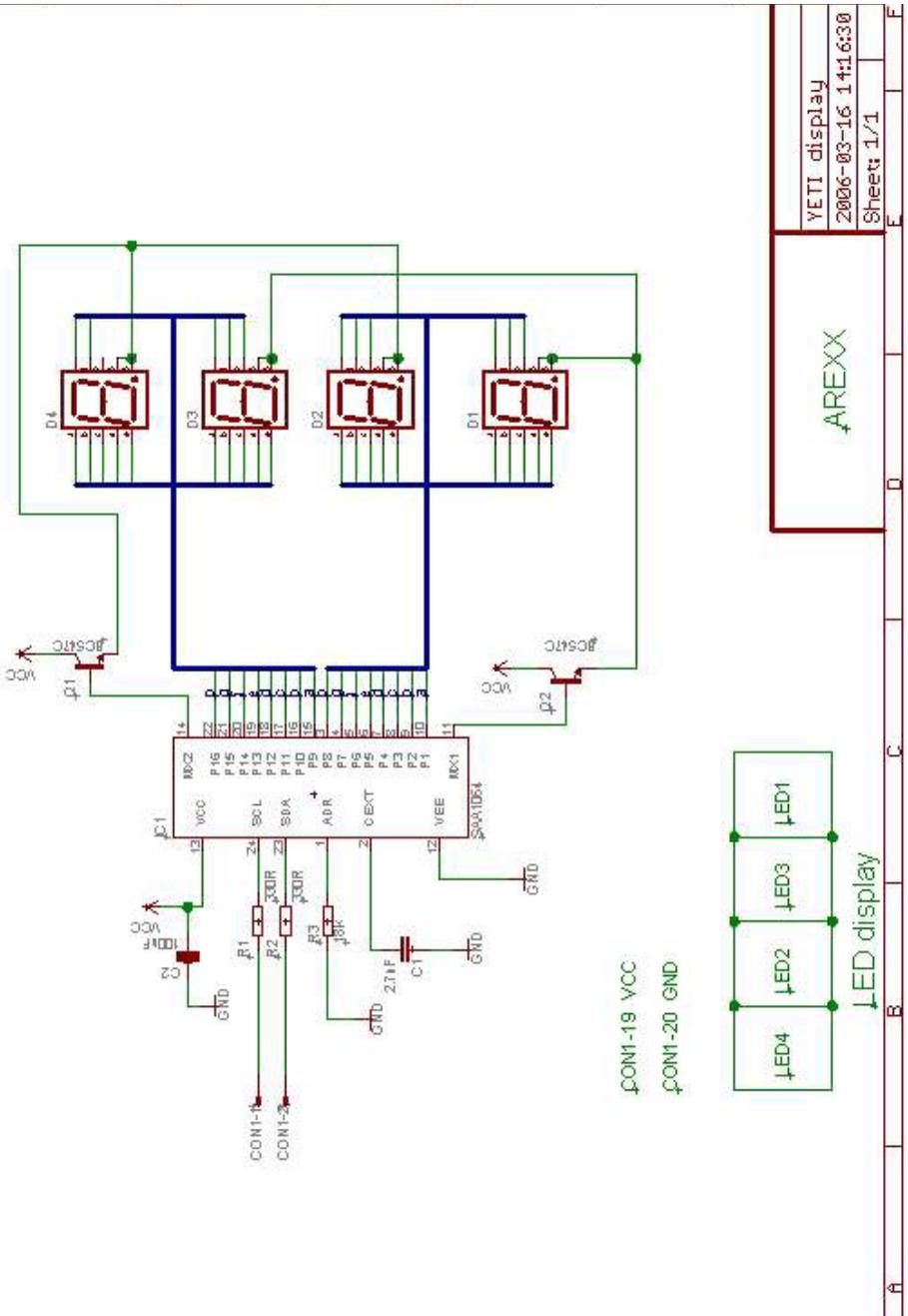
Hinweis:

Nach dem Abgleichvorgang des YETI-Servomotorsystems kann man die Funktion aus der Datei entfernen und dadurch den Umfang des zu ladenden YETIprogramms erheblich verringern.

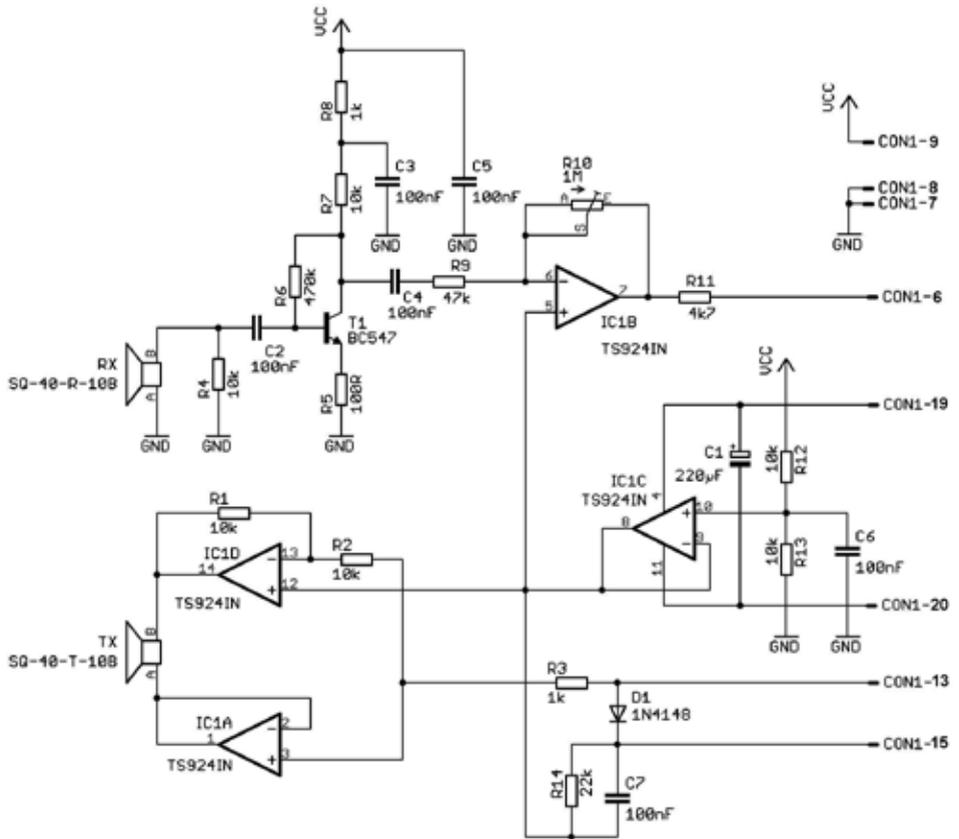
B. SCHALTBILD YETI



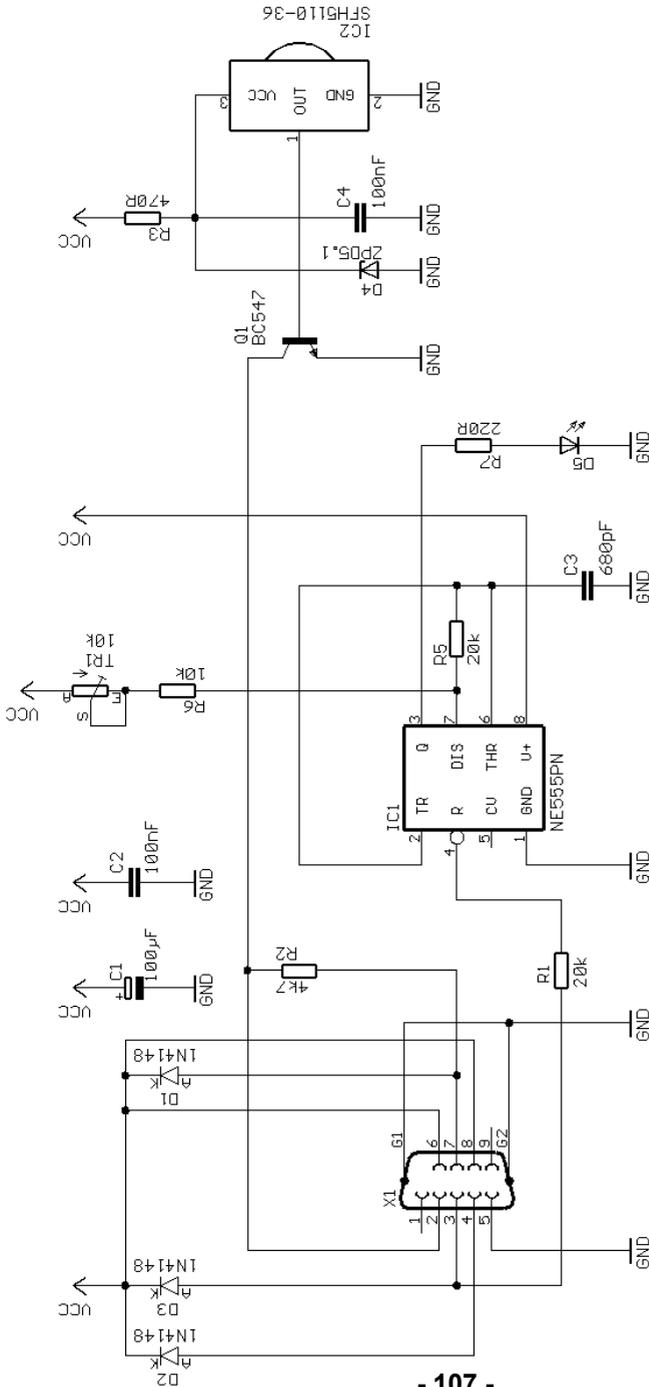
C. SCHALTBILD DISPLAY MODUL



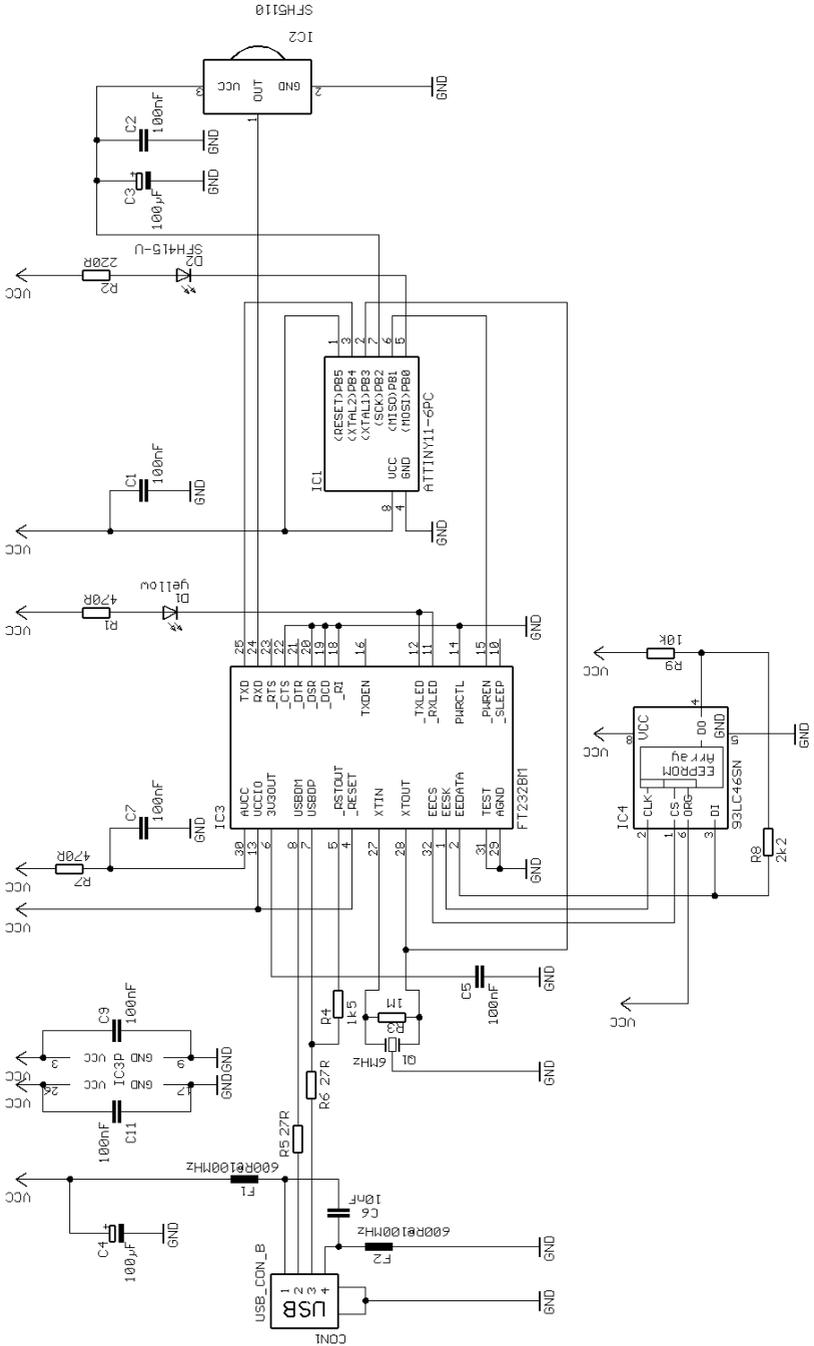
D. SCHALTBILD US MODUL



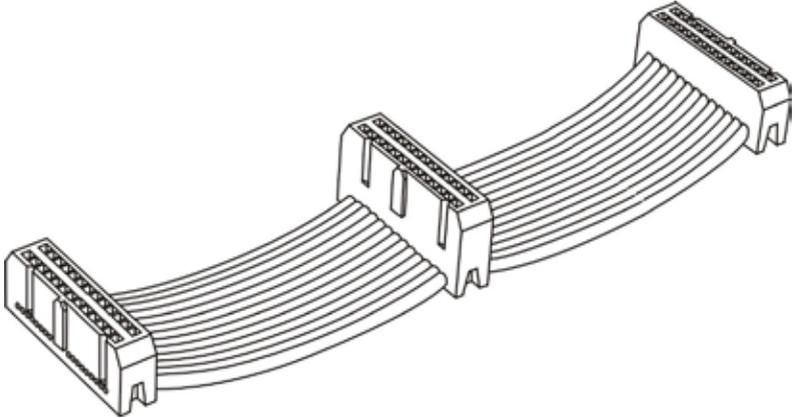
E. SCHALTBILD RS-232 IR-TRANSCEIVER



F. SCHALTBILD USB IR-TRANSCEIVER



G. FLACHBANDANSCHLUSSBELEGUNG



Pin 1	SCL	Serial Clock (für die I2C Datenkommunikation)
Pin 2	SDA	Serial Data (für die I2C Datenkommunikation)
Pin 3	PC3(ADC3)	Digitaler Eingang/Ausgang oder analoger Messeingang
Pin 4	PC2(ADC2)	Digitaler Eingang/Ausgang oder analoger Messeingang
Pin 5	PC1(ADC1)	Digitaler Eingang/Ausgang oder analoger Messeingang
Pin 6	PC0(ADC0)	Digitaler Eingang/Ausgang oder analoger Messeingang
Pin 7	GND	GND (Mehrere Anschlüsse zur Entstörung vorgesehen)
Pin 8	GND	GND (Mehrere Anschlüsse zur Entstörung vorgesehen)
Pin 9	AVCC	Analoge Referenzspannung für die AD-Konverter
Pin 10	PC6(RESET)	Mikrocontroller Reset Anschluss
Pin 11	PB5(SCK)	Digitaler Eingang/Ausgang
Pin 12	PB4(MISO)	Digitaler Eingang/Ausgang oder I2C Funktion Pin
Pin 13	PB3(MOSI/OC2)	Digitaler Eingang/Ausgang oder I2C function pin oder Timer2 Pin
Pin 14	PD3(INT1)	Digitaler Eingang/Ausgang oder externer Interrupt
Pin 15	PD6(AIN0)	Digitaler Eingang/Ausgang oder analoger Testeingang
Pin 16	D7(AIN1)	Digitaler Eingang/Ausgang oder analoger Testeingang
Pin 17	PD0(RXD)	Digitaler Eingang/Ausgang oder RS232 Eingang
Pin 18	PD1(TXD)	Digitaler Eingang/Ausgang oder RS232 Ausgang
Pin 19	VCC	VCC
Pin 20	GND	GND (Mehrere Anschlüsse zur Entstörung vorgesehen)

H. Fehlersuche

H1. ALLGEMEIN

Alle Bauteile auf richtige Einbaulage (Polung) und korrekten Wert prüfen. Lötstellen auf Kurzschlüsse bzw. kalte Lötstellen nachsehen. Hat sich irgendwo ein Lötauge gelöst? Sind diese Kontrollen durchgeführt ohne einen Fehler zu finden, muss mit Hilfe des Schaltplanes und einem geeigneten Messgerät (Multimeter bzw. Oszilloskop) das defekte Bauteil gesucht werden. IC's, transistoren, dioden und LEDs sind oft die wahrscheinlichsten Kandidaten für einen Defekt.

H.2. RS232-IR-Transceiver geht nicht !

H.2.1. Tastendruck und Zeichenausgabe stimmen nicht überein

Solange am Trimmer TR1 drehen bis Tastendruck und Zeichenausgabe Übereinstimmen.

H.2.2. Das Terminal-Programm gibt keine Zeichen aus

Ist das Timer-IC (IC1) montiert bzw. richtig herum (Kerbe zeigt in Richtung der drei Dioden) eingesteckt?

Eine Infrarot-Fernbedienung eines beliebigen Gerätes (Videorecorder, Fernseher, Tuner, ...) nehmen, diese in Richtung des IR-Transceivers halten und einige Tasten drücken. Passiert gar nichts, nochmal die COM-Port-Einstellungen kontrollieren. Gibt das Terminal-Programm wirre Zeichen aus, funktioniert der Empfangsteil (IC2, R3, C4, D4, T1). Alle anderen Teile sind zu überprüfen.

H.2.3. Es geht immer noch nicht

Sind Lötstellen Kontrollen durchgeführt ohne einen Fehler zu finden, muss mit Hilfe des Schaltplanes und einem geeigneten Messgerät (Multimeter bzw. Oszilloskop) das defekte Bauteil gesucht werden. (IC1, IC2, Q1, D4 sind die wahrscheinlichsten Kandidaten für einen Defekt)

H.3. USB-Infrarot-Transceiver funktioniert nicht

H.3.1 Windows

Ist der Treiber ordnungsgemäß installiert? Teilweise werden andere COM-Port-Nummern zugewiesen als die letzte. Hier hilft im Hyperterminal auch mal andere Ports einzustellen und den Test zu wiederholen. Ggf. kann man auch in der Systemsteuerung nachsehen, welcher Port zugewiesen wurde.

H.3.2 Linux

Hier kann es helfen, den USB-Transceiver abzustecken und nach etwas Warten wieder zu kontaktieren. Taucht dann der Eintrag im proc-Verzeichnis immer noch nicht auf, kann es nützen, einen neueren Kernel zu installieren.

H4. IR-Schnittstelle

H.4.1. YETI sendet keine Zeichen

SFH 415-U, IR-LED richtig eingebaut?

Polung der IR-LED prüfen.

Widerstand R7 (RS-232 IR-transceiver) und Widerstand R5 (YETI) richtig?

220 Ω (ro,ro,br,gld)

H.4.2. YETI empfängt keine Zeichen

Zwischen IR-Transceiver und YETI muss Sichtverbindung bestehen (Abstand ca. 50cm) und der IR-Transceiver muss voll funktionsfähig sein.

SFH5110 IR-Empfänger-IC richtig eingebaut in YETI und RS-232 IR-Transceiver?

Transistor Q1, Diode D4, Widerstand R3 und C4 überprüfen von RS-232 IR-transceiver und Widerstand R4 und C6 von YETI hauptplatine !

470 Ω (br,sw,br,gld)

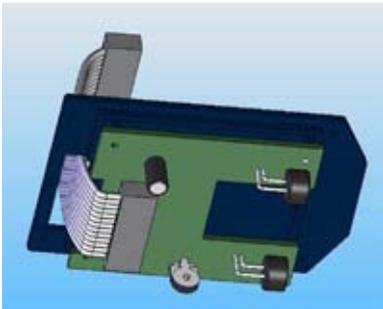
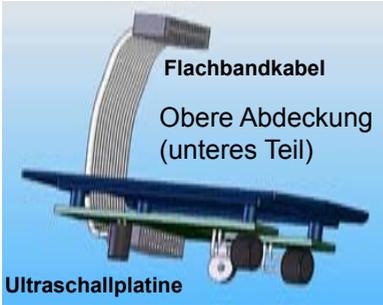
100nF (Aufdruck104)

Wer den Fehler bisher noch nicht gefunden hat, der möge sich überlegen, ob er das SFH5110 IR-Empfänger-IC eingelötet oder "eingeschweißt" hat. SFH5110 IR-Empfänger-IC ist ein wenig hitzeempfindlich und ist evtl. beim Einbauen kaputt gegangen, dann neues IC (SFH 5110-36) besorgen und einbauen.

H.4.3. Es geht immer noch nicht so richtig

Kommt es bei der Übertragung von Daten vom PC zu YETI immer wieder zu Schwierigkeiten, so muss ein wenig am Trimmer TR1 des Transceivers gedreht werden.

I. MONTAGE ERWEITERUNGSKITS



Ultraschallplatine

Hauptplatine

Akkusatz

SERVO



Montage
Display Platine

Montage Ultraschallplatine



Montage Experimentiplatine



J. KALIBRIERUNGS- UND TESTPROGRAMM

Der Hersteller hat den YETI-Prozessor bereits mit einem Selbsttest und Kalibrierungsprogramm ausgestattet.

Dieses Programm mit dem Namen 'test.hex' umfasst:

1. einen Kalibrierungsmodus
2. einen Laufmodus

Selbsttestverfahren

Schalten Sie den YETI ein.

Das YETI 'BOOTLOADER'-Programm wird gestartet

- Beide LEDs werden sofort aufleuchten
- Beide Servos werden kurz in Uhrzeigersinn eine Umdrehung durchführen
- Nach 3 Sekunden folgt ...

#ENDE des 'BOOTLOADER'-Programms

→ Das 'TEST.HEX'-Programm wird gestartet

Das 'TEST.HEX'-Programm wird gestartet

- Beide LEDs werden abgeschaltet
- Beide Servos bewegen sich (nahezu) in die Mittelposition
- Der Roboter erzeugt eine Reihe Beep-Töne
- Das Programm wartet nun 3 Sekunden bis eine Taste im Hyperterminalprogramm am PC gedrückt wird. Anschließend wechselt YETI in den Kalibrierungsmodus.

Nach dem Aktivieren der ersten Taste im Hyperterminalprogramm am PC erzeugt der YETI einen kurzen Beep-Ton, der bedeutet:

- dass YETI jetzt in den Kalibrierungsmodus gewechselt hat.
- dass YETI ein Infrarotsignal empfangen hat und der Infrarotsender offensichtlich korrekt funktioniert.
- Falls nun eine Taste im Hyperterminalprogramm gedrückt wurde und der YETI sich im Kalibrierungsmodus befindet, können Sie die ENTER-Taste am Hyperterminalprogramm drücken und den YETI aus dem Kalibrierungsmodus in den Laufmodus wechseln lassen.
- Falls Sie jedoch keine Taste im Hyperterminalprogramm betätigen, wird YETI automatisch nach 3 Sekunden aus dem Kalibrierungsmodus in den Laufmodus wechseln.

#ENDE des Kalibrierungsmodus

→ START des YETI-Laufmodus

START des YETI-Laufmodus

- Der Modus startet mit einem einzigen Beep-Ton
- Der linke LED des YETIs wird eingeschaltet
- YETI verneigt sich nach links und schiebt seinen rechten Fuß nach vorne. Anschließend werden 3 Rechts-/Linksschritte durchgeführt.
- YETI dreht sich nach links und macht 4 Schritte
- YETI macht 3 Schritte rückwärts
- YETI dreht sich nach rechts und macht 4 Schritte
- YETI schüttelt sich dreimal links und rechts
- YETI wiederholt diese og. Bewegungsfolge ununterbrochen

Nach jeder Bewegungsfolge erzeugt YETI Beep-Töne und wechselt jedes mal die aufleuchtende LED-Kombination.

#YETI bleibt ununterbrochen in diesem Laufmodus

#Hinweise zu den zusammengebauten YETIs

Eine wichtiger Kontrollschritt überprüft, ob der YETI seinen Laufgang mit dem rechten Fuß beginnt. Andernfalls wurde der Anschluss der Servomotoren zur Steuerungselektronik versehentlich vertauscht.