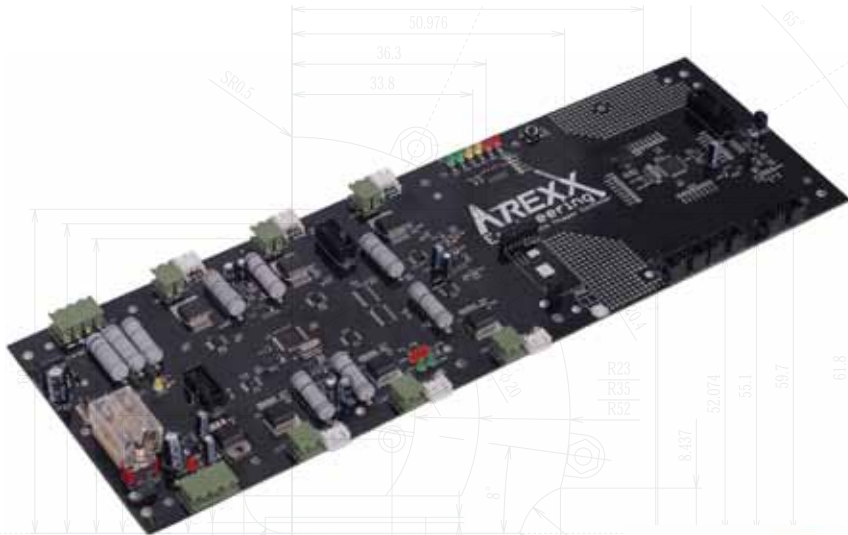


LERNROBOTER Wild Thumper Controller

Aufbauanleitung: Modell WTR-CK1



Inhaltsverzeichnis

1.	Produkt beschreibung WT Kontroller	3
	1.2. Specificationen	2
	1.3. Was kann die Wild Thumper	3
2.	Nötwendige Werkzeuge	5
3.	Teilliste	6
4.	ASEMBLIERUNG von WT	8
5.	Erste Praxis Test	22
6.	Software installation	23
7.	Programmer and Loader	36
	7.1 Robot loader	37
	7.2 Connection of USB interface Windows	38
	7.3 Connection of USB interface LINUX	41
	7.4 Testing the USB interface	42
	7.5 Opening a port Linux	43
	7.6 Selftest	44
	7.7 Calibration	46
	7.8 Keyboard test	48
8.	Programmer und Loader	49
9.	Programmieren von die Wild Thumper	54
10.	Drahtlose Aplications Software	
xx.	APPENDIX	
	A. Circuit diagram Wild Thuimper	63
	B. Circuit diagram Power Supply	64
	C. Circuit diagram Connectors	65
	D. Circuit diagram Keyboard	66
	E- Circuit Programmer	67

AREXX and Wild Thumper are registered trademarks of AREXX Engineering - HOLLAND.

© Deutsche Übersetzung (Juni 2011): AREXX Engineering (NL).

Sämtliche Texte, Bilder, Grafiken dieses Handbuchs unterliegen den geltenden Urheberrechtsbestimmungen und dürfen nicht ohne Genehmigung des europäischen Importeurs verwendet werden.

AREXX Engineering - Zwolle (NL).

Der Hersteller und Vertriebspartner sind nicht haftbar für Schäden als Folge des Misbrauchs, Montagefehlern oder Missachtung der Anweisungen dieses Handbuchs.

Dieses Manuskript kann jederzeit ohne Benachrichtigung geändert werden.



Hersteller:
AREXX Engineering

Europäischer Importeur
AREXX Engineering
ZWOLLE The Netherlands

Technische Unterstützung:

WWW.AREXX.COM
WWW.ROBOTERNETZ.DE

© AREXX Holland

© Deutsche Übersetzung: AREXX - die Niederlande

1. PRODUKTBESCHREIBUNG WTR-CK1

Das Wild Thumper Ansteuerungsmodul bildet die komplette Ansteuerungselektronik für den Wild Thumper "ALLRAD GELÄNDE CHASSIS JSR-6WD". D

Die Steuerungsplatine basiert auf zwei ATMEL Mikroprozessoren, die miteinander kommunizieren. Einer dieser Prozessoren kümmert sich um die Kommunikation und das Ablesen der Sensoren. Der zweite Prozessor ist zuständig für die Ansteuerung der sechs Motoren mittels hocheffizienten H-Brücken.

Zur Ansteuerungselektronik gehören standardmäßig zwei APC-220 Funktransceivers für das 433MHz Frequenzband. Außerdem liefern wir zum System einen Programmieradapter einschließlich der benötigten Loader-Software, sodass beide Prozessoren auf einfacher Weise programmiert werden können. Die Ansteuerungsprogramme werden in C geschrieben, mit Hilfe des WINAVR compiliert und anschließend in eine HEX-Datei verwandelt.

Wie bei allen AREXX-Robotersystemen stehen auch für den Wild Thumper wieder viele Anschlüsse für Ein-/Ausgänge und ein I2C-Bus zur Verfügung, damit sie am System selbst verschiedene Applikationen anschließen können.

Lieferumfang.

- 1 Stück. Hauptplatine
- 1 Stück. Frontplatine mit Berührungsschalter, Licht- und Infrarot-Strahlungssensoren
- 1 Stück. Programmieradapter mit USB- und Funkkommunikation
- 2 Stück. APC-220 Module
- 6 Stück. Rad-Encoder-Module mit HALL Sensoren
- 6 Stück. Encoder-Magnetringe zur Befestigung auf de Radachse
- 1 Stück. CD mit der Betriebsanleitung und Software

Montageteile, Anschlusskabel und Steckverbinder



Warnungen

- * Das Rückgaberecht erlischt nach dem Öffnen der Plastikverpackungen der Bauelementen.
- * Lesen die das Handbuch bitte vor der Aufbauphase vollständig bis zum Schluss.
- * Wir bitten Sie vorsichtig mit den Werkzeugen zu arbeiten.
- * Bauen Sie den Roboter bitte nicht in Anwesenheit von kleinen Kindern auf.
Diese können sich an den Werkzeugen verletzen oder kleine Bauteile verschlucken.
- * Überprüfen Sie die korrekte Polarität beim Einlegen der Batterien.
- * Sehen Sie zu, dass die Batterien und die Batteriehalter immer trocken sind. Falls das System Feuchtigkeit aufnimmt sollten Sie die Batterien entfernen und alle Teile gut trocknen.
- * Entfernen Sie bitte die Batterien falls Sie das Gerät mehr als 1 Woche nicht benutzen.

1.2. Spezifikationen:

Stromversorgung: 7,2 Volt sub C Batterie Pakete (nicht im Lieferumfang enthalten)

2 St. ATMEGA 644 Prozessoren

6 St. Magnet-Drehzahlgeber für Odometrie

6 St. HAL Odometrie Sensoren

Motor-Stromwächter für alle 6 Motoren

Spannungsüberwachung

Lichtsensoren

IR-Sensoren

Tastsensoren

Zahlreiche Externe I/O und I2C-Anschlüsse

12 LEDs

Hauptplatine mit APC-220 drahtlosem Transceiver-Module

USB-Schnittstelle mit APC-220 drahtlosem Transceiver-Module

Beide Prozessoren sind frei programmierbar in der Hochsprache "C"

Die Programme können mit dem mitgelieferten USB-interface und der AREXX Robotloader Software einfach in den Mikroprozessor geladen werden.

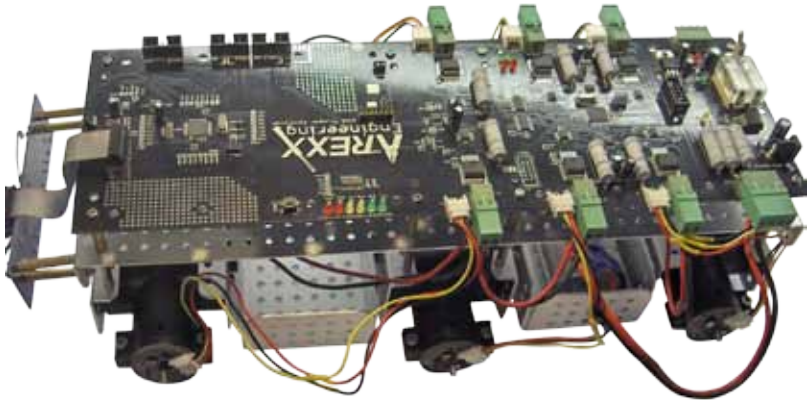
Abmessungen: (LxB) 360 x 120 (mm)

Hinweis: Max 2 St. 7,2V sub C Batteriepakete

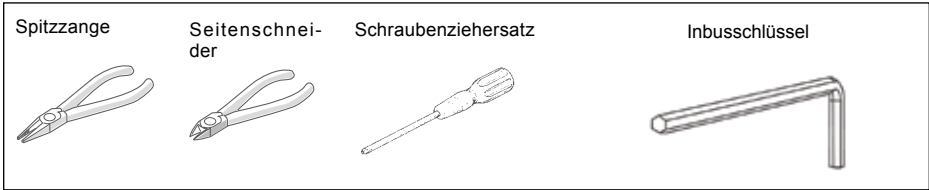


1.3. WILD THUMPER

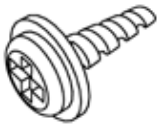
- Selbst Programmieren in C mit WINAVR
- Beispiel- und Neuprogramme in den Wild Thumper übertragen.
- Den Wild Thumper mit einem PC und WTC Software drahtlos kontrollieren
- Ihre eigene drahtlose Kamerabilder in WTC PC Software einfügen
- Den Wild Thumper mit Erweiterungsmodulen ausbauen, so dass er hören, fühlen und sehen kann und somit auf seine Umgebung reagiert
- Mittels I2C Schnittstelle kann der Wild Thumper mit seinem Umfeld und vielen anderen Geräten kommunizieren.
- Künstliche Intelligenz: mit Hilfe von selbstlernender Software kann der Wild Thumper seine Software ständig selbsttätig verbessern.



2. Benötigte Werkzeuge

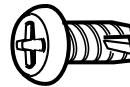


Selbstschneidende Schrauben

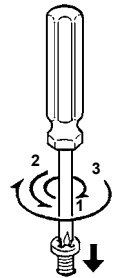


Selbstschneidende Schrauben verhalten sich wie Holzschrauben, d.h. sie schneiden ein spiralförmiges Gewinde ins Material, das sich wie eine Mutter verhält. Zum Ende weist dieser Schraubentyp ein größeres Gewinde und eine schärferen Spitze auf als eine normale Schraube.

Selbstschneidende Schrauben haben einen Spalt an der Spitze, so dass man sie besser ins Material eindrehen kann. Am einfachsten dreht man die Schrauben wie folgt fest:



Kontermutter

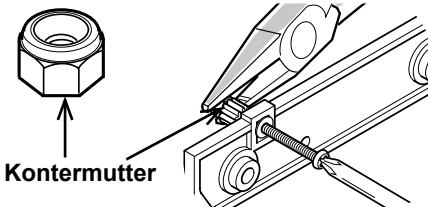


- 1 Drehen Sie die Schraube fest
- 2 Lockern Sie die Schraube
- 3 Jetzt nochmals festschrauben.

Falls die Schrauben zu oft gelöst und wieder festgeschraubt werden, kann das Loch im Laufe der Zeit immer größer werden und passt die Schraube irgendwann nicht mehr richtig.

Kontermutter

Das Festschrauben einer Kontermutter



Kontermutter



Schraubenschlüssel:

Der Bausatz enthält einen Doppelmaulschlüssel, den Sie (anstatt der Zange) für die M2 und M3 Muttern verwenden sollten.



3. TEILLISTE

Hauptplatine



O 1x

Programmieradapter



O 1x

Wild Thumper chassis



O 1x

Magnet-Encoder-Ring



O 6x

Abstandshalter(Scheibe)



O 6x

Encoderplatine



O 6x

Flachbandkabel



O 1x 10 Pol

O 1x 14 Pol

Encoderkabel



O 6x

Frontplatine



O 1x

Power-Anschluss



O 2x

Motorstecker



O 6x

USB-Kabel



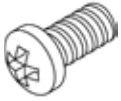
O 1x

APC-220



O 2x

Rundkopf-
Schraube M3x6



○ 24x

Inbus-
schraube



○ 6x

Mutter
M3



○ 4x

Abstandshalter
M3



○ 12x

Vorbereitung des Chassis für den Wild Thumper:

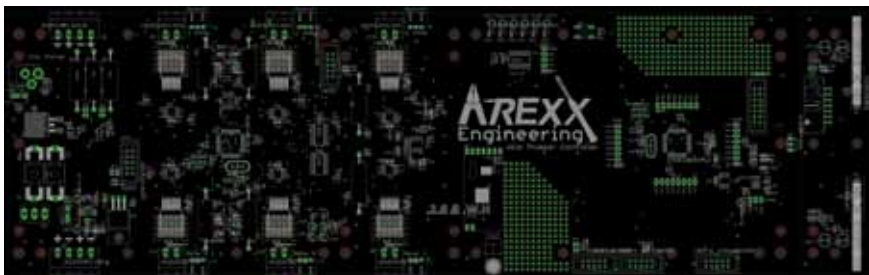
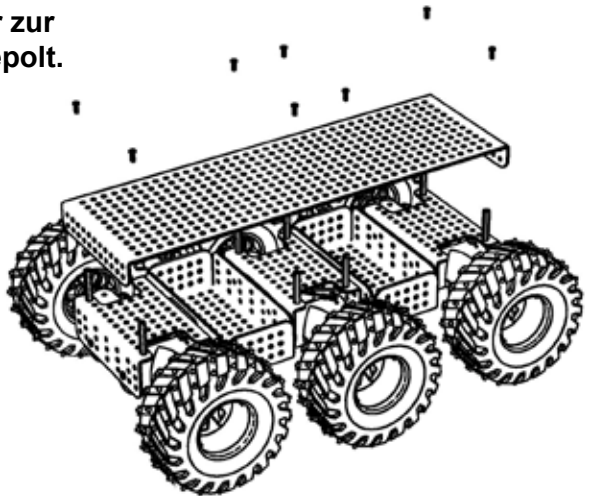
Folgende Teile werden benötigt

1x WT Chassis
6x Motorstecker

Entferne die Abdeckung des WT Chassis.

Befestige die 6 Motorstecker an die Motorkabel.

Die ROTE Ader wird immer zur Frontseite des Steckers gepolt.



4. ZUSAMMENBAU des WT

Vorzugsweise sollte man alle Mechanikteile in folgender Reihenfolge einbauen

- Entferne die Abdeckung
- Befestige die Motorstecker (Rot + gehört zur Frontseite)
- Befestige die Abstandshalter der Leiterplatte
- Befestige die Abdeckung (Befestigen mit Abstandshalter)
- Montiere die Achsen an den Rädern
- Schraube den Inbusschraube vollständig in die radachse
- Befestige die Magnetringe
- Befestige den Abstandshalter und die Hall-Encoderplatine auf den Motoren
- Befestige die Leiterplatten (Front- und Hauptplatine)
- Verdrahte alle Kabel



Montage der WT-CK1 Leiterplatte:

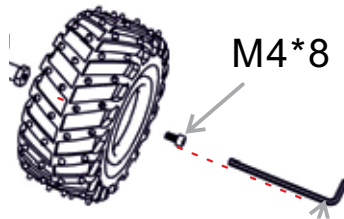
Ehe Sie die Montage der Leiterplatte des Wild Thumpers starten sollten Sie das Chassis des Wild Thumpers teilweise entfernen um einen guten Zugang zu den Motoren zu bewerkstelligen.

Die Montage der Räder:

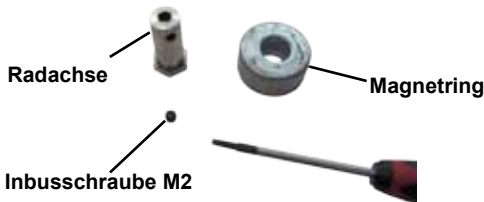
Folgende Teile werden benötigt:

6x Magnetring
6x Radachse
6x Rad
6x Inbusschraube M2
6x Inbusschraube M4x8

Befestige die Radachse zum Rad (6 x).
Befestige auch eine Inbusschraube in die Radachse und drehe die Schraube ganz hinein.
Siehe die Detailzeichnung!



Montage des Magnetings:



So sollte der Magnetring auf die Radachse befestigt werden. Anschließend wird der Magnetring mit Kleber an die Achse oder an das Rad befestigt.

Montiere den Magnetring auf die Radachse und befestige diesen mit einem Kleber oder Siegellack an das Rad

Die Montage der Encoder:

Folgende Teile werden benötigt:

- 6x Encoder-Platine
- 6x Abstandshalterscheibe
- 6x Encoderkabel
- 12x Schraube M2x10

Vor der Befestigung der Räder müssen wir die Encoder einbauen. Die Steckerstifte stecken am Encoder ein wenig heraus und müssen zuerst maßgerecht gestutzt werden. Schneiden Sie mit einem Seitenschneider die Konnektorstifte so kurz wie möglich ab. Die Stifte können nie ganz bis zur Leiterplatte abgeschnitten werden. Deshalb benötigen wir einen Abstandshalter.

Auch der Abstandshalter muß maßgerecht vorbereitet werden. Die Löcher im Abstandshalter sind bereits an der richtigen Stelle vorgebohrt worden. Sie müssen noch etwas nachgebohrt werden. Benutzen Sie zum Nachbohren bitte einen kleinen 2mm Bohrer.

Siehe die Fotoserie für Details zu diesen Arbeiten



Schneiden Sie die Konnektorstifte so kurz wie möglich ab

Abstandshalter



Encoderplatine



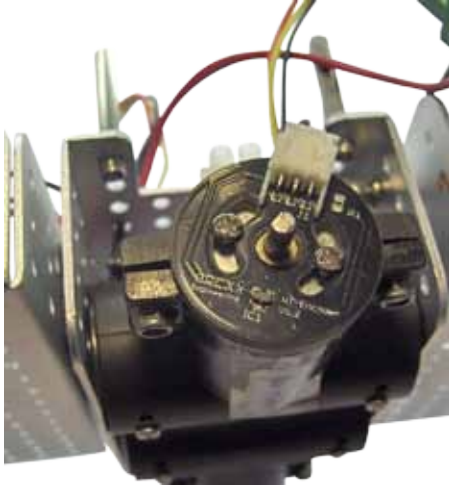
Benutzen Sie zum Nachbohren bitte einen kleinen 2mm Bohrer



Vor dem Einbauen der Encoder und Abstandshalter müssen wir überprüfen ob die Konnektorstifte nicht über den Abstandshalter herausragen und das Motorgehäuse berühren können. Sollte das der Falls sein müssen die Stifte weiter gekürzt werden. Nach dem Einbau des Encoders muss der Magnetring noch auf der Achse befestigt werden.

Befestigung des Encoders:

Folgende Teile werden benötigt: 1x WT Chassis
6x Encoderplatine
6x Abstandshalter
6x Schraube M3x12
6x Encoder Kabelsatz



Befestigen Sie die Encoder zu den Motoren wie auf dem Foto angegeben.

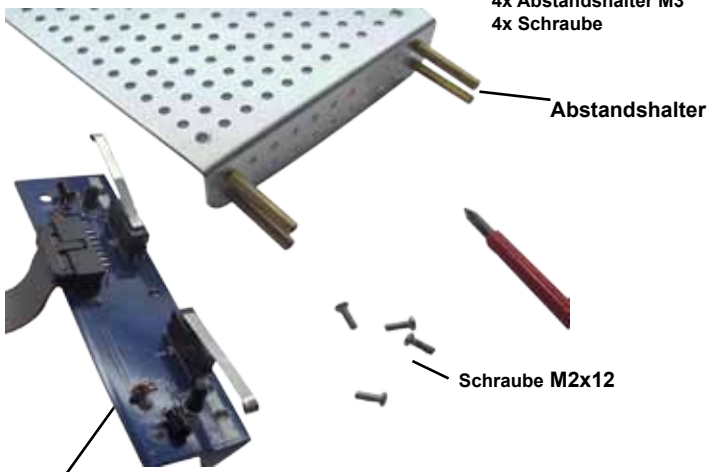
Schließen Sie auch die Encoderkabel an.

Die Abstand von HAL-Sensor nach Magnet muss nicht zu Groß sein!

Einbau der Frontplatte:

Folgende Teile werden benötigt:

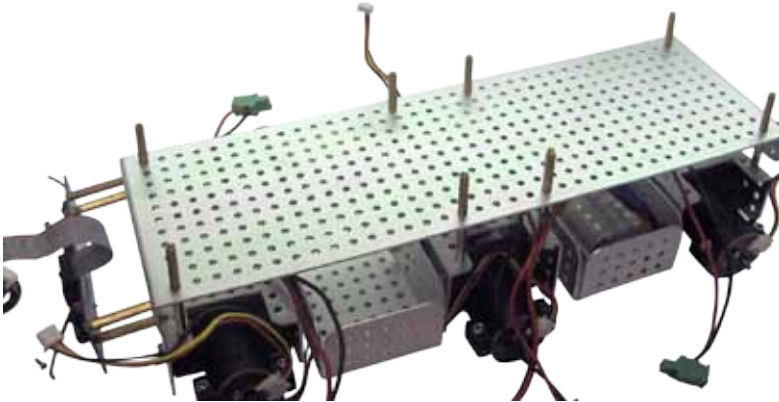
1x Frontplatte
4x Abstandshalter M3
4x Schraube



Befestigen Sie die Frontplatte zur Vorderseite des Chassis des Wild Thumpers wie es auf diesem Bild skizziert wird.

Befestigung der Abstandshalter für die Hauptplatine:

Folgende Teile werden benötigt: 1x WT Chassis
1x Obere Leiterplatte
8x Abstandshalter



Platzieren Sie die obere Leiterplatte auf dem WT Chassis und befestigen Sie dieses Teil mit den 8 Abstandshaltern,

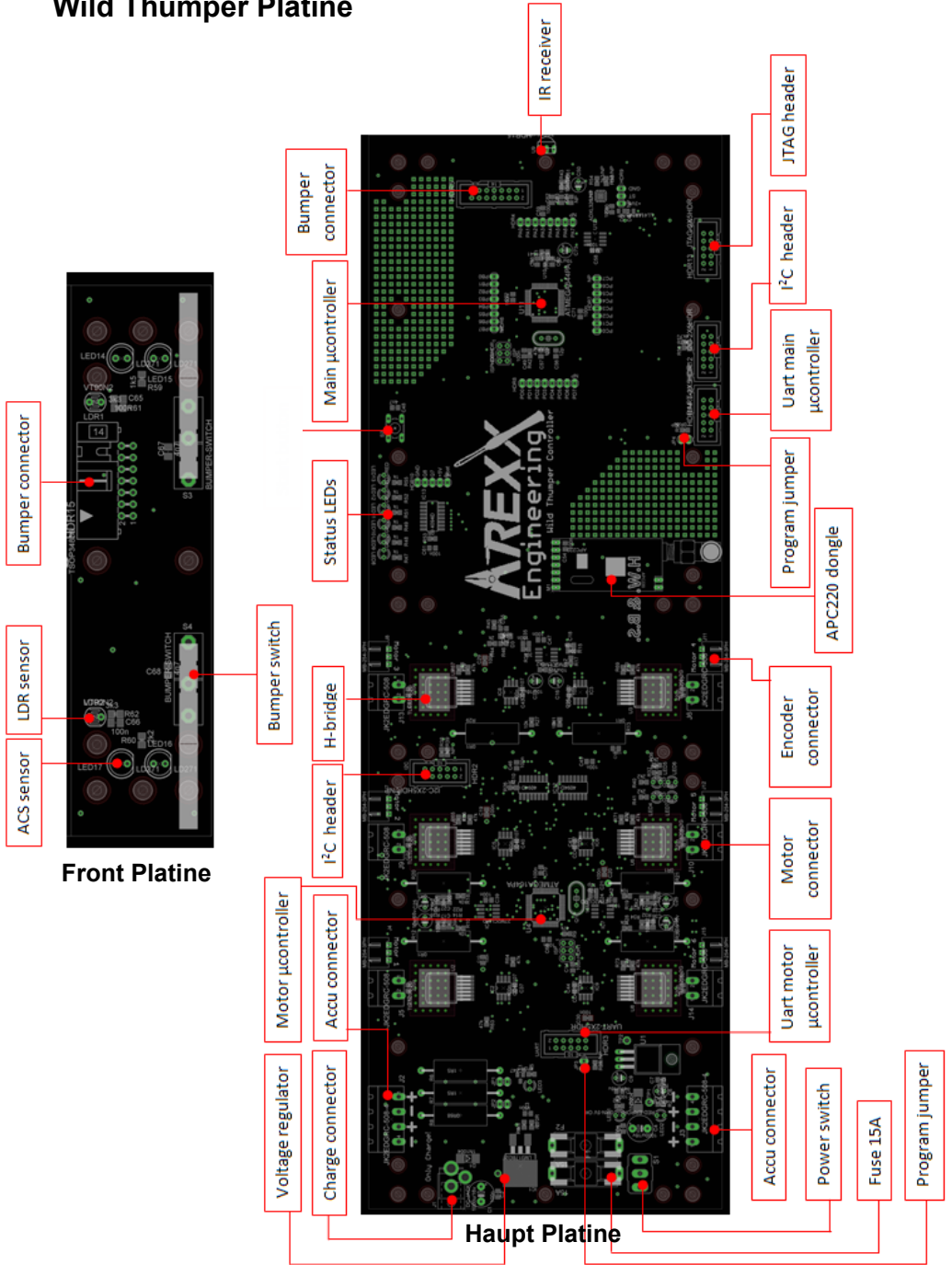
Endmontage:

Folgende Teile werden benötigt: 1x WT Chassis
1x Leiterplatte
1x APC-220
1x Flachbandkabel
(für die Frontplatine)



Befestigen Sie die Leiterplatte auf der oberen Platte. Schließen Sie alle Kabel zur Leiterplatte an und befestigen Sie die Räder auf den Achsen. Bestücken Sie zum Schluss die APC-220 in den Sockel wie es in der Abbildung skizziert wird.

Wild Thumper Platine



5. WILD THUMPER ELECTRONIK BESCHREIBUNG:

5.1. Stromversorgung

Der Wild Thumper soll drahtlos gelenkt werden und deshalb müssen wir das System mit Akkus speisen. Die Batterien liefern eine 7,2 Volt Spannung.

Da die Elektronik nur maximal 5 Volt verträgt müssen wir die Spannung mit einem Spannungsregler auf einen geringeren Wert herabsetzen.

Um den geringen Spannungsbereich optimal zu regeln verwenden wir einen Regler L4940V5 von STMicroelectronics mit einem Spannungsabfall von nur 0,5 Volt.

Die Stromversorgung wird beim Anschließen gegen falscher Verpolung geschützt mit einer Diode (D2). Diese Diode sperrt den Strom bei falscher Polarität. Zur Entstörung wurden Entstörungskondensatoren (C7 und C8) vorgesehen. Auch wurden am Ein- und Ausgang zwei Pufferkondensatoren (C6 und C9) eingebaut.

5.2. Ladeschaltung

Die angeschlossenen Akkus können in eingebautem Zustand aufgeladen werden. Dazu wurde eine Stromquelle auf der Basis eines einstellbaren Reglers LM317 eingebaut. Bei Aufladen wird dieser Regler als Stromquelle geschaltet. Sobald das Akku aufgeladen ist wird der Strom abgeschaltet und wechselt die Schaltung mittels Transistor T1 von Stromquelle in eine Spannungsquelle.

5.3. Motorsteuerung

Die Motorsteuerung enthält sechs identischen Module die jeweils einen Motor ansteuern. Jedes Modul enthält eine H-Brücke, eine Logikschaltung (zwei NAND-Gatter) zur Einsparung der Mikroprozessorausgängen und eine Strommessungsschaltung. Die Schaltung erlaubt die Links- und Rechtsdrehrichtung sowie die Abbremsung. Die H-Brücke verwendet Pulsbreitenmodulation (PBM) zur Geschwindigkeitssteuerung.

5.4. Motorstrommessung

Jedes Modul zur Motorstrommessung enthält einen Leistungswiderstand. Das Ohmsche Gesetz besagt dass der Strom durch den Leistungswiderstand sich proportional zum Spannungsabfall verhalten wird. Zur Begrenzung des Spannungsabfalls wählen wir einen geringen Widerstandswert, der in unserem Entwurf 0,1 Ohm beträgt. Bei diesem Wert ist der Spannungsabfall gering (d.h. 0,1V bei einer Stromstärke von 1A) und dieser muss daher angehoben werden. Diese Verstärkung wird geleistet von einem sogenannten Operationsverstärker (Englisch: „operational amplifier“, abgekürzt: „opamp“). In dieser Schaltung wird die Verstärkung mit den Widerständen R1 und R2 festgelegt. Folgende Formeln werden benötigt um die Schaltung zu dimensionieren:

$$U_{\text{ein}} = I_{\text{motor}} \times R_3$$

$$U_{\text{aus}} = A_{\text{((Verstärkung))}} \times U_{\text{ein}}$$

$$A_{\text{((Verstärkung))}} = 1 + R_1 / R_2$$

Dimensionierung:

$$R_3 = 0,1 \Omega$$

$$I_{\text{(motor_max)}} = 5,5 \text{ A}$$

$$U_{\text{(aus_max)}} = 4 \text{ V bei einer Stromversorgungsspannung von } 5 \text{ V.}$$

$$U_{\text{(ein_max)}} = I_{\text{(motor_max)}} \times R_3 = 5,5 \text{ A} \times 0,1 \Omega = 0,550 \text{ V}$$

$$A = U_{\text{(aus_max)}} / U_{\text{(ein_max)}} = 4 \text{ V} / 0,550 \text{ V} = 7,27$$

$$R_2 / R_1 = A - 1 = 7,27 - 1 = 6,27$$

$$R_2 / R_1 = 6,27$$

Auf der Basis der oben stehenden Formel wählen wir die Werte für R1 und R2. Aus wirtschaftlichen Gründen wählen wir Widerstandswerte aus der E12, beziehungsweise E24-Reihe. Mit folgenden Werten erreicht der Operationsverstärker die optimale Verstärkung:

$$R1 = 6200 \text{ Ohm}$$

$$R2 = 39000 \text{ Ohm}$$

$$A = 1 + R_1 / R_2 = 1 + 39000 / 6200 = 7,29$$

In diesem Fall ist die Abweichung zum Sollwert sehr gering: $(7,29 - 7,27 = 0,02)$, was wir akzeptieren dürfen.

5.5. Encoder

Zur Geschwindigkeitsmessung jedes Einzelrades erhält Rad einen eigenen Encoder. Dieser Encoder besteht aus einem Magnetring mit einem Hall-Sensor. Der Magnetring enthält 6 Magnetpolpaare. Ein drehender Magnetring erzeugt ein magnetisches Wechselfeld.

Sobald der angewandte Hall-Sensor einen Südpol detektiert wechselt der Ausgang zu 5V und bei einem Nordpol zu 0V. Eine Pulsbreitenmessung der eingehenden Impulsen liefert die Geschwindigkeit jedes Einzelrades.

5.6. Motorsynchronisation

Die Spezifikationstoleranzen der Motoren können +/- 10% betragen. Aus diesem Grund erhält jeder Motor ein eigenes Steuerungsmodul, dass eine einheitliche Geschwindigkeit für alle Motoren sicherstellt. Das Steuerungsmodul funktioniert wie folgt:

Der gewünschte Geschwindigkeitswert wird (mittels I²C) aus der Zentralsteuerung vorgegeben und in der Motorsteuerung mit einem PID-Regler in eine Sollvorgabe umgesetzt. Mit diesem Wert berechnet das System einen PBM-(PulsBreitenModulation)-Wert, der den Motor ansteuert. Danach wird die Motorgeschwindigkeit wieder gemessen, wobei ein neuer Steuerungswert festgelegt wird.

5.7. Mikrokontroller

Wie in der Spezifikation bereits erläutert verwendet der Roboter zwei Mikrokontroller. Es handelt sich um zwei Prozessoren des Herstellers Atmel, wobei wir einfachheitshalber den gleichen Typ ATmega644 gewählt haben. Beide Mikrokontroller verwenden I²C zur internen Kommunikation. Eine Übersicht des Systems wird in Abb. 16 abgebildet.



Abb. 16 : Darstellung Motordata Strom

5.7.1. Hauptkontroller

Der Hauptkontroller verwaltet alle Aufgaben im Bereich des Wild Thumpers und trifft alle Entscheidungen, die anschließend gegebenenfalls zu Aktionen führen. Zu den Aufgaben des Hauptkontrollers gehören:

- Verarbeitung der eingehenden PC-Daten
- Zusammensetzung der Berichten, die an den PC versandt werden, wenn der Auftrag dazu vorliegt.
- Kommandos an die Motorkontroller zur Ansteuerung der Motoren.
- Der Betrieb der I²C Schnittstelle.
- Regelmäßige Aktualisierung der Messdaten der Motorkontroller.
- Aktualisierung der Sensordaten.

Der Hauptkontroller arbeitet mit zwei Schnittstellen, und zwar einem extra UART Interface beziehungsweise einem JTAG Interface. Das UART Interface steht zur Verfügung zur Kommunikation mit einem dritten system. Die JTAG Schnittstelle steht zur Verfügung zum Debuggen des Hauptkontrollers.

5.7.2. Motorkontroller

Der Motorcontroller erledigt Aufgaben, die aus dem Hauptkontroller stammen und mittels I²C-Schnittstelle bereitgestellt werden. Aus den Aufträgen wird mit einer in Software aufgebauten PID-Regler für jeden Motor einen Regelwert berechnet. Der Wert stellt sicher, dass jeder Motor mit der gleichen Drehzahl arbeitet.

Abgesehen von diesen Aufgaben aus dem Hauptkontroller misst der Motorcontroller regelmässig den Strom und die Geschwindigkeit. Die Stromwerte und Drehzahlwerte können vom Hauptkontroller jederzeit abgelesen werden.

5.7.3. Kommunikation zwischen Haupt- und Motorkontroller (I2C)

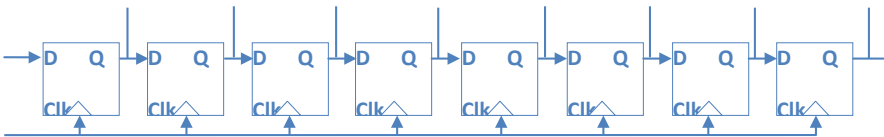
Die Kommunikation zwischen beiden Mikrokontroller verwendet das serielle I2C-Bussystem. Es handelt sich dabei um einem Zweidraht-Systembus, der einen Draht (SDA) für Daten und einen anderen Draht (SCL) für das Systemtaktsignal verwendet.

Beide Datenleitungen sind mit einem Pullup-Widerstand auf der Stromversorgungsspannung angeschlossen. Im passiven Zustand ist der Signalpegel deshalb hoch. Sobald ein Mikrokontroller Daten versenden möchte wird zuerst eine Startbedingung generiert. Danach folgen acht Bits für die Adresse, die gefolgt wird von einer Information für das zuständige IC. Jede 8 Bitreihe wird mit einem Bestätigungs (Acknowledge)-Bit abgeschlossen, womit das System den korrekten Eingang der Daten überprüft. Nach dem Transfer aller Daten generiert das System ein Stoppsignal, womit der Transfer abgeschlossen wird.

Im Motorkontroller stehen zwei Register zum Ablesen zur Verfügung. Diese Register enthalten Messdaten und den Status des Motorkontrollers. Außerdem gibt es einige Kommandos zur Generierung von Steuerungskommandos für den Wild Thumpers.

5.8. Schieberegister

Weil der gewählte Mikrokontroller nicht genügend Ein- und Ausgänge zur Verfügung stellt wurden einige externe Ausgänge hinzugefügt. Dazu werden Schieberegister verwendet. Ein Schieberegister besteht aus einer Reihe zusammenschalteten D-Flipflops womit serielle Daten in Paralleldaten umgesetzt werden. Das D-Datensignal wird bei jeder positiven Flanke des Taktsignals auf den Ausgang übertragen. Indem man nun acht D-Flipflops folgendermassen in Serie zusammenschaltet kann ein serielles Datensignal auf parallel geschalteten Ausgänge bereitgestellt werden.



In diesem Projekt verwenden wir Schieberegister in der Form des ICs HEF4094, das eine seriellen Schnittstelle mit drei Signalen bereitstellt:

- Daten
- Taktsignal
- Latch

Das zusätzliche „Latch“-Signal wird benutzt um Daten in die Ausgangsregister zu übertragen. Solange dieses Signal 0V ist werden Daten intern gespeichert aber noch nicht auf den Ausgang übernommen. Im folgenden Bild wird angedeutet wie die Daten übernommen werden, wobei Ausgang sechs aktiviert wird.



Zur Ansteuerung eines Schieberegisters aus einem Mikrokontroller wird ein sogenannter SPI-bus benutzt, der mit drei Datenleitungen arbeitet:

- MOSI (Master out, Slave in)
- MISO (Master in, Slave out)
- CLOCK (Taktsignal)

Man verwendet einen SPI-bus häufig zur Kommunikation zweier Systemen. Dabei sind zwei Datenleitungen im Einsatz und zwar für jedes Gerät eine Leitung. Wie beim Schieberegister werden die Daten zur positiven Flanke des Taktsignals übertragen.

Weil die Schieberegister nichts zurücksenden ist eine Datenleitung zur Ansteuerung eines Schieberegisters ausreichend, das heißt die Signale MOSI und CLOCK sind in der Lage das Schieberegister zu versorgen.

5.9. Das RF-Modul (d.h. das Radio Frequenz Modul)

Das RF-Modul wird mit Hilfe einer seriellen Schnittstelle (UART) mit dem Hauptcontroller verbunden. Zur Anpassung der Onboard-Einstellungen wurde die SET-Signalleitung des Transceivers auch zum Mikrocontroller geleitet.

5.10. Sensoren

Die Sensoren wurden auf zwei Leiterplatten aufgeteilt und zwar auf der vorderen Stoßstange des Wild Thumpers und auf der Leiterplatte mit dem Rest der Elektronikbaugruppen. Der Wild Thumper verfügt über nachfolgenden Sensoren.

5.10.1. LDR

Auf der vorderen Stoßstange des Wild Thumpers befinden sich sowohl links als rechts ein LDR. Ein LDR ist ein lichtempfindlicher Widerstand. Sobald ein LDR mehr Licht empfängt nimmt sein Widerstand zu. Wie beim Akkuspannungssensor verwenden wir auch in diesem Fall einen Spannungsteiler, nur wird jetzt eine 5V-Spannung anstatt einer Akkuspannung geteilt. Wir verwenden im Teiler einen Festwiderstand und einen LDR, womit wir wie im beigefügten Schaltbild die Lichtintensität messen können. Die Ausgangsspannung des Teilers wird am ADC-Eingang des Mikrocontrollers eingelesen.

5.10.2. Infrarot Kollisionsdetektor

Auf der Frontleiterplatte des Wild Thumpers befinden sich Infrarot LEDs, die Datenpakete versenden. Diese Datenpakete werden an Hindernisse reflektiert und in Infrarotsensoren auf der Hauptplatine empfangen. Falls ein Hindernis entdeckt wird, treffen die Signale der reflektierten Datenpakete ein und kann der Roboter eine geeignete Aktion starten.

5.10.3. Die Stoßstange

Es wurde eine kleine Leiterplatte entworfen, die vorne am Wild thumper befestigt wird. Auf dieser Platine wurden zwei Mikroschalter montiert, die auf Hindernisse reagieren und im Roboter eine geeigneten Aktion auslösen.

6. DER ERSTE WT-PRAXIS TEST

1. Bauen Sie zuerst die Mechanik- und Elektronikmodule des Wild Thumpers mit Hilfe der Bauanleitung zusammen.
2. Schließen Sie ggf. das 7.2 Volt Akkupack (oder ein Netzteil) an.
3. Schalten Sie bitte das System mit dem Hauptschalter ein.

Spannung anschließen

Netzgerät

Es gibt 2 Möglichkeiten zur Stromversorgung.

Die einfachste Lösung ist der Anschluss eines Netzgeräts mit einer Spannung von 8 bis 12 Volt / 2 Ampere am DC Stecker. Dabei wird die Spannung auf den Eingang des Spannungsreglers angeschlossen.

Batterien

Die zweite Möglichkeit ist die Verbindung eines Akkupacks über dem Batterie-Anschluss. ***Dabei wird die Akkuspannung nach dem Ausgang des Spannungsreglers angeschlossen und darf deshalb auch niemals größer werden als 7,2 Volt!!***



WARNUNG!

Die maximale Akku- bzw. Batteriespannung für den Wild Thumper Kontroller ist 7.2 V!

Sobald der Roboter Arm auf einer Stromversorgung angeschlossen wird, können Sie den Roboter mit dem Hauptschalter einschalten und leuchtet die Grüne LED (LED1) auf.

Bei Laden der Batterien leuchtet die Gelbe LED (LED 3) auf.

Nun, das war für den Anfang gar nicht so schwierig und es sieht aus, als ob man jetzt bereits fertig wäre. Jetzt aber fängt die Arbeit erst richtig an !

Jetzt aber folgt zuerst Kapitel 7, in dem wir die Software installieren

7. Software Installation

Als nächstes kommen wir zur Installation der Software. Die korrekt installierte Software wird für alle nachfolgenden Kapitel unbedingt benötigt.

Es sind Administrator-Rechte erforderlich, also melden Sie sich ggf. vorher als Administrator in Ihrem PC an!

Wir empfehlen Ihnen zuerst das gesamte Kapitel in Ruhe durchzulesen und erst dann Schritt für Schritt die Installationsanleitung durchzugehen!

Grundlegende Kenntnis der Bedienung von Computern mit Windows oder Linux Betriebssystemen und den gängigen Programmen wie Dateimanager, Webbrowser, Texteditor, Packer (WinZip, WinRAR, unzip o.ä.) und ggf. Linux-Shell etc. muss vorausgesetzt werden! Wenn Sie sich also nur wenig mit Computern auskennen, sollten Sie sich auf jeden Fall gut damit vertraut machen bevor Sie den Wild Thumper in Betrieb nehmen! Eine Einführung in die Bedienung von Computern ist nicht Ziel dieser Anleitung und würde den Rahmen bei weitem sprengen! Hier geht es nur um den Wild Thumper, dessen Programmierung und die speziell dafür benötigte Software.

Die CD-ROM zum Wild Thumper

Sie haben vermutlich die CD-ROM zum Wild Thumper bereits im Laufwerk Ihres Computers eingelegt – falls doch nicht, legen Sie diese nun bitte ein! Es sollte unter Windows kurz darauf per Autostart das CD Menü erscheinen. Andernfalls können Sie über einen Dateimanager die Datei "start.htm" im Hauptverzeichnis der CD mit einem Webbrowser wie z.B. Firefox öffnen. Die Installationsdateien für Firefox finden Sie übrigens auf der CD im Ordner

<CD-ROM-Laufwerk>:\Software\Firefox

sofern Sie noch keinen aktuellen Webbrowser installiert haben sollten. (Es sollte mindestens Firefox 1.x oder der Internet Explorer 6 sein...)

Nach Auswahl der Sprache finden Sie im CD-Menü neben dieser Anleitung (die es auch zum Download auf unserer Homepage gibt), vielen Informationen, Datenblätter und Fotos auch den Menüpunkt "Software". Hier sind alle Software-Tools, der USB Treiber und die Beispielprogramme mit den Quellcodes für den Wild Thumper zu finden.

Je nach Sicherheitseinstellungen Ihres Webbrowsers können Sie die Installations-Programme direkt von der CD starten!

Wenn Ihr Browser dies aufgrund der Sicherheitseinstellungen nicht erlaubt, müssen Sie die Dateien zunächst in ein Verzeichnis auf Ihrer Festplatte speichern und dann von dort starten. Genauer dazu steht auf der Software Seite des CD Menüs. Alternativ können Sie natürlich auch direkt in einem Dateimanager auf das CD-Laufwerk wechseln und die Software von dort installieren. Die Verzeichnisnamen sind so gewählt, dass sie eindeutig den entsprechenden Softwarepaketen und Betriebssystemen zugeordnet werden können.

WinAVR - für Windows

Als erstes werden wir WinAVR installieren. WinAVR ist aber - wie der Name schon andeutet- nur für Windows verfügbar !

Linux Anwender müssen beim nächsten Abschnitt weiterlesen.

WinAVR (das wird wie das englische Wort "whenever" ausgesprochen) ist eine Sammlung von vielen nützlichen und notwendigen Programmen für die Software Entwicklung für AVR Mikrocontroller in der Sprache C. WinAVR enthält neben dem GCC für AVR (das nennt sich dann insgesamt "AVR-GCC", mehr Infos dazu folgen später) auch den komfortablen Quelltexteditor "Programmers Notepad 2", den wir auch für die Programmentwicklung für den Wild Thumper einsetzen werden!

WinAVR ist ein privat organisiertes Projekt, hinter dem keine Firma o.ä. steht - es ist kostenlos im Internet verfügbar. Neuere Versionen und weitere Informationen finden Sie hier:

<http://winavr.sourceforge.net/>

Inzwischen wird das Projekt aber auch offiziell von ATMEL unterstützt, und der AVRGCC lässt sich in AVRStudio, die Entwicklungsumgebung für AVR's von ATMEL, einbinden. Das werden wir in diesem Handbuch aber nicht beschreiben, für unsere Zwecke ist Programmers Notepad besser geeignet. Die WinAVR Installationsdatei finden Sie auf der CD im Ordner:

<CD-ROM-Laufwerk>:\Software\AVR-GCC\Windows\WinAVR\

Die Installation von WinAVR ist sehr einfach und selbsterklärend - normalerweise brauchen keinerlei Einstellungen geändert werden – also einfach immer auf "Weiter" klicken!

Wenn Sie Windows Vista oder Windows 7 benutzen, müssen Sie auf jeden Fall die neueste Version von WinAVR verwenden! Auch mit Windows 2000 und XP sollte es problemlos klappen. Falls nicht, können Sie eine der beiden älteren Versionen ausprobieren, die ebenfalls auf der CD zu finden sind (vor Neuinstallation immer bereits installierte WinAVR Versionen wieder deinstallieren!). Offiziell wird Win x64 noch nicht unterstützt, aber auf der CD findet sich ein Patch für Win x64 Systeme falls es Probleme geben sollte. Mehr Infos dazu finden Sie auf der Software Seite des CD Menüs!

AVR-GCC, avr-libc und avr-binutils - für Linux

(Windows Anwender können diesen Abschnitt überspringen!)

Unter Linux kann es schon ein wenig aufwändiger werden. Bei einigen Distributionen sind die benötigten Pakete zwar schon vorhanden, aber meist nur veraltete Versionen. Deshalb müssen Sie neuere Versionen kompilieren und einrichten. Wir können hier nicht im Detail auf jede der zahlreichen Linux Distributionen wie SuSE, Ubuntu, RedHat/Fedora, Debian, Gentoo, Slackware, Mandriva etc. pp. in zig verschiedenen Versionen mit ihren jeweiligen Eigenheiten eingehen und beschreiben das daher nur allgemein.

Das gilt auch für alle anderen Linux Abschnitte in diesem Kapitel!

Das hier beschriebene Vorgehen muss also bei Ihnen nicht unbedingt zum Erfolg führen. Oft kann es hilfreich sein, wenn Sie im Internet z.B. nach "<LinuxDistribution> avr gcc" o.ä. suchen (Verschiedene Schreibweisen ausprobieren). Auch das gilt für alle anderen Linux Abschnitte - natürlich mit angepassten Suchbegriffen! Falls Sie Probleme bei der Installation des AVR-GCC haben, können Sie auch mal in unserem oder im Roboternetz Forum nachschauen bzw. in einem der zahlreichen Linux Foren. Zunächst müssen Sie evtl. schon installierte Versionen des avr-gcc, der avr-binutils und der avr-libc deinstallieren – wie schon gesagt sind diese meist veraltet. Das können Sie mit dem jeweiligen Paketmanager ihrer Distribution tun indem Sie nach „avr“ Inbetriebnahme suchen und die drei oben genannten Pakete deinstallieren – sofern diese überhaupt vorhanden sind. Ob der avr-gcc schon installiert ist oder nicht und wenn ja wo, können Sie über eine Konsole z.B. leicht mit > which avr-gcc herausfinden.

Sollte hier ein Pfad angezeigt werden, ist schon eine Version installiert. Geben Sie in diesem Fall einfach mal:

```
> avr-gcc --version
```

ein und schauen Sie sich die Ausgabe an. Sollte eine Versionsnummer kleiner als 3.4.6 angezeigt werden, müssen Sie diese alte Version auf jeden Fall deinstallieren.

Wenn die Versionsnummer zwischen 3.4.6 und 4.1.0 liegt, können Sie erstmal versuchen ob Sie Programme kompilieren können (s. nachfolgende Kapitel) und erst wenn das fehlschlägt, die neuen Tools installieren. Wir installieren im Folgenden die derzeit aktuelle Version 4.1.1 (Stand von März 2011) mit einigen wichtigen Patches.

Werden die oben genannten Pakete nicht im Paketmanager angezeigt, obwohl definitiv schon ein avr-gcc vorhanden ist, müssen Sie die entsprechenden Binärdateien manuell löschen – also die /bin, /usr/bin usw. Verzeichnisse nach allen Dateien, die mit „avr-“ anfangen absuchen, und diese dann löschen (natürlich NUR diese Dateien und sonst nichts anderes!). Eventuell vorhandene Verzeichnisse wie /usr/avr oder /usr/local/avr müssen ebenfalls gelöscht werden.

Achtung: Sie müssen unbedingt sicherstellen, dass die normalen Linux Entwicklungstools wie GCC, make, binutils, libc, etc. installiert sind, bevor Sie mit dem Übersetzen und der Installation beginnen können! Das tun Sie am besten über den Paketmanager Ihrer Distribution. Jede Linux Distribution sollte die benötigten Pakete schon auf der Installations-CD mitliefern bzw. aktuelle Pakete über das Internet bereitstellen.

Stellen Sie sicher, dass das Programm „texinfo“ installiert ist. Installieren Sie bitte ggf. das entsprechende Paket, bevor Sie weitermachen – sonst klappt es nicht!

Ist das erledigt, kann mit der eigentlichen Installation begonnen werden.

Es gibt nun zwei Möglichkeiten, entweder man macht alles von Hand, oder man nutzt ein sehr einfach anzuwendendes Installationskript.

Wir empfehlen es zunächst mit dem Skript zu versuchen. Wenn das nicht klappt, kann man immer noch den Compiler von Hand einrichten!

Achtung:

Sie sollten für die Installation noch genug freien Speicherplatz auf der Festplatte zur Verfügung haben! Temporär werden mehr als 400MB benötigt. Über 300MB davon können nach der Installation wieder gelöscht werden, aber während der Übersetzung braucht man den Platz.

Viele der nachfolgenden Installationsschritte erfordern **ROOT RECHTE**, also loggen Sie sich ggf. mit „su“ als root ein oder führen Sie die kritischen Befehle mit „sudo“ o.ä. aus, wie man es z.B. bei Ubuntu machen muss (das Installationskript, mkdir in /usr/local Verzeichnissen und make install brauchen root Rechte).

Achten Sie im Folgenden bitte auf EXAKTE Schreibweise aller Befehle!

Jedes Zeichen ist wichtig und auch wenn einige Befehle evtl. etwas seltsam aussehen – das ist alles richtig so und kein Tippfehler!

(<CD-ROM-Laufwerk> muss man natürlich trotzdem mit dem Pfad des CD-ROM-Laufwerks ersetzen!)

Alle für uns relevanten Installationsdateien für den avr-gcc, avr-libc und binutils finden Sie auf der CD im Ordner:

```
<CD-ROM-Laufwerk>:\Software\avr-gcc\Linux
```

Zunächst müssen Sie alle Installationsdateien in ein Verzeichnis auf Ihrer Festplatte kopieren – **das gilt für beide Installationsvarianten!** Hier nutzen wir das Home Verzeichnis (übliche Abkürzung für das aktuelle Home Verzeichnis ist die Tilde: „~“):

```
> mkdir ~/Wild Thumper  
> cd <CD-ROM-Laufwerk>/Software/avr-gcc/Linux  
> cp * ~/Wild Thumper
```

Die Dateien können Sie nach der erfolgreichen Installation natürlich wieder löschen um Platz zu sparen!

Automatisches Installationskript

Wenn man das Skript mit `chmod` ausführbar gemacht hat, kann es sofort losgehen:

```
> cd ~/Wild Thumper
> chmod -x avrgcc_build_and_install.sh
> ./avrgcc_build_and_install.sh
```

Die Nachfrage, ob man mit dieser Konfiguration installieren möchte oder nicht, können Sie mit „y“ beantworten.

ACHTUNG: Das Übersetzen und Installieren wird dann je nach Rechenleistung Ihres Systems einige Zeit in Anspruch nehmen. (z.B. etwa 15 min auf einem 2GHz CoreDuo Notebook – bei langsameren Systemen evtl. entsprechend länger)

Das Skript spielt auch einige Patches ein – das sind diese vielen `.diff` Dateien, die in dem Verzeichnis liegen.

Wenn alles klappt, sollte ganz zum Schluss folgendes erscheinen:

```
(./avrgcc_build_and_install.sh)
(./avrgcc_build_and_install.sh) installation of avr GNU tools complete
(./avrgcc_build_and_install.sh) add /usr/local/avr/bin to your path to use the avr GNU tools
(./avrgcc_build_and_install.sh) you might want to run the following to save disk space:
(./avrgcc_build_and_install.sh)
(./avrgcc_build_and_install.sh) rm -rf /usr/local/avr/source /usr/local/avr/build
```

Dann können Sie wie es dort vorgeschlagen wird

```
rm -rf /usr/local/avr/source /usr/local/avr/build
```

ausführen! Das löscht alle temporären Dateien, die Sie normalerweise nicht mehr benötigen.

Jetzt können Sie den nächsten Abschnitt überspringen und noch den Pfad auf die avr tools setzen.

Sollte die Ausführung des Skriptes fehlschlagen, müssen Sie sich genau die Fehlermeldungen ansehen (auch mal in der Konsole etwas hochscrollen) – meist fehlen dann irgendwelche Programme, die man vorher noch installieren muss (wie z.B. das oben erwähnte `texinfo`). Bevor Sie nach einem Fehler weitermachen, sollten Sie die bereits erzeugten Dateien im Standardinstallationsverzeichnis „`/usr/local/avr`“ vorsichtshalber löschen – am besten das ganze Verzeichnis.

Wenn Sie nicht wissen, was da genau falsch gelaufen ist, bitte alle Kommando-zeilenausgaben in einer Datei speichern und damit an den Support wenden. Bitte immer so viele Informationen wie möglich mitsenden! Dann wird es einfacher, Ihnen zu helfen.

GCC für den AVR

Der GCC wird ähnlich wie die Binutils gepatcht, übersetzt und installiert:

```
> cd ~/Wild Thumper> bunzip2 -c gcc-4.1.1.tar.bz2 | tar xf -
> cd gcc-4.1.1
> patch -p0 < ../gcc-patch-0b-constants.diff
> patch -p0 < ../gcc-patch-attribute_alias.diff
> patch -p0 < ../gcc-patch-bug25672.diff
> patch -p0 < ../gcc-patch-dwarf.diff
> patch -p0 < ../gcc-patch-libiberty-Makefile.in.diff
> patch -p0 < ../gcc-patch-newdevices.diff
> patch -p0 < ../gcc-patch-zz-atmega256x.diff
> mkdir obj-avr
> cd obj-avr
> ../configure --prefix=$PREFIX --target=avr --enable-languages=c,c++ \
--disable-nls --disable-libssp --with-dwarf2
> make
> make install
```

Nach dem \ kann man einfach Enter drücken und weiterschreiben – so kann der Befehl auf mehrere Zeilen aufgeteilt werden. Kann man aber auch ganz weglassen.

AVR Libc

Und schließlich noch die AVR libc:

```
> cd ~/Wild Thumper
> bunzip2 -c avr-libc-1.4.5.tar.bz2 | tar xf -
> cd avr-libc-1.4.5
> ./configure --prefix=$PREFIX --build=`./config.guess` --host=avr
> make
> make install
```

Achtung: bei `–build= `./config.guess`` darauf achten auch den Gravis, auch „accent grave“ genannt (à <-- den Strich da auf dem a! Neben der Backspace Taste – rechts oben auf der Tastatur, einmal mit Shift diese Taste drücken und danach die Leertaste) und kein normales Hochkomma oder Anführungszeichen zu benutzen, sonst klappt es nicht.

Pfad setzen

Sie müssen jetzt dafür sorgen, dass das Verzeichnis `/usr/local/avr/bin` auch in der Pfad Variablen eingetragen ist – sonst kann man den `avr-gcc` nicht aus der Konsole bzw. aus Makefiles heraus aufrufen. Dazu müssen Sie den Pfad in die Datei `/etc/profile` bzw. `/etc/environment` o.ä. (variiert von Distribution zu Distribution) eintragen – mit einem Doppelpunkt „:“ getrennt von den anderen schon vorhandenen Einträgen. In der Datei könnte das dann in etwa so aussehen:

```
PATH="/usr/local/bin:/usr/bin:/bin:/usr/X11R6/bin:/usr/local/avr/bin"
```

Jetzt in einer beliebigen Konsole „avr-gcc –version“ eingeben wie weiter oben beschrieben – wenn das funktioniert, ist die Installation gelungen!

Manuelle Installation

Wenn Sie den Compiler lieber von Hand einrichten wollen oder die Installation mit dem Skript nicht klappt, können Sie nach den Anweisungen im folgenden Abschnitt vorgehen.

Die Beschreibung hier orientiert sich an diesem Artikel:

http://www.nongnu.org/avr-libc/user-manual/install_tools.html

der auch in der AVR Libc Dokumentation im PDF Format auf der CD zu finden ist:

<CD-ROM-Laufwerk>:\Software\Documentation\avr-libc-user-manual-1.4.5.pdf

Wir fassen uns hier zwar sehr viel kürzer, spielen aber gleich noch ein paar wichtige Patches ein – ohne diese funktionieren einige Dinge nicht richtig.

Zunächst müssen wir uns ein Verzeichnis erstellen, in das wir alle Tools installieren werden. Das sollte `/usr/local/avr` sein.

Also in einer Konsole ALS ROOT folgendes eingeben:

```
> mkdir /usr/local/avr  
> mkdir /usr/local/avr/bin
```

Es muss nicht unbedingt dieses Verzeichnis sein. Wir legen dafür einfach die Variable `$PREFIX` für dieses Verzeichnis an:

```
> PREFIX=/usr/local/avr  
> export PREFIX
```

Das muss nun noch unbedingt der `PATH` Variable hinzugefügt werden:

```
> PATH=$PATH:$PREFIX/bin  
> export PATH
```

Binutils für AVR

Nun müssen Sie den Quellcode der Binutils entpacken und ein paar Patches einspielen.

Wir nehmen hier an, dass Sie alles ins Home Verzeichnis `~/Wild Thumper` kopiert haben:

```
> cd ~/Wild Thumper
> bunzip2 -c binutils-2.17.tar.bz2 | tar xf -
> cd binutils-2.17
> patch -p0 < ../binutils-patch-aa.diff
> patch -p0 < ../binutils-patch-atmega256x.diff
> patch -p0 < ../binutils-patch-coff-avr.diff
> patch -p0 < ../binutils-patch-newdevices.diff
> patch -p0 < ../binutils-patch-avr-size.diff
> mkdir obj-avr
> cd obj-avr
```

Nun wird das configure Skript ausgeführt:

```
> ../configure --prefix=$PREFIX --target=avr --disable-nls
```

Dieses Skript ermittelt, was auf Ihrem System verfügbar ist und erzeugt dementsprechend passende Makefiles. Jetzt können die Binutils übersetzt und installiert werden:

```
> make
> make install
```

Das kann je nach Rechenleistung Ihres Systems schon ein paar Minuten dauern – das gilt auch für die beiden nächsten Abschnitte – vor allem für den GCC!

Java 6

Der RobotLoader (Infos dazu s.u.) wurde für die Java Plattform entwickelt und ist unter Windows und Linux verwendbar (theoretisch auch andere Betriebssysteme wie OS X, aber hier kann AREXX Engineering leider noch keinen offiziellen Support leisten). Damit das funktioniert, ist es notwendig, ein aktuelles Java Runtime Environment (JRE) zu installieren. Oft haben Sie dies bereits auf dem Rechner, allerdings muss es mindestens Version 1.6 (= Java 6) sein! Falls Sie also noch kein JRE oder JDK installiert haben, müssen Sie zunächst das auf der CD mitgelieferte JRE 1.6 der Firma SUN Microsystems installieren, oder alternativ eine neuere Version von <http://www.java.com> oder <http://java.sun.com> downloaden.

Windows

Das JRE 1.6 befindet sich für Windows in folgendem Ordner:

```
<CD-ROM-Laufwerk>:\Software\Java\JRE6\Windows\
```

Unter Windows ist die Installation von Java sehr einfach - Sie müssen nur den Setup starten und den Anweisungen auf dem Bildschirm folgen - fertig. Den nächsten Abschnitt können Sie überspringen.

Linux

Unter Linux ist die Installation meistens auch relativ problemlos möglich, bei einigen Distributionen kann es aber ein wenig Handarbeit erfordern.

In diesem Ordner:

```
<CD-ROM-Laufwerk>:\Software\Java\JRE6\
```

finden Sie das JRE1.6 als RPM (SuSE, RedHat etc.) und als selbstextrahierendes Archiv „.bin“. Unter Linux ist es besser wenn Sie zunächst im Paketmanager Ihrer jeweiligen distribution nach Java Paketen suchen (Suchbegriffe z.B. „java“, „sun“, „jre“, „java6“ ...) und dann diese distributions-eigenen Pakete verwenden und nicht die auf dieser CD-ROM! Achten Sie aber unbedingt darauf Java 6 (= 1.6) oder ggf. eine neuere Version zu installieren und keine ältere Version!

Unter Ubuntu oder Debian funktioniert das RPM Archiv nicht direkt – hier müssen Sie die Paketmanager Ihrer jeweiligen Distribution bemühen, um an ein passendes Installationspaket zu kommen. Das RPM sollte bei vielen anderen Distributionen wie RedHat/Fedora und SuSE aber problemlos funktionieren. Falls nicht, bleibt noch der Weg das JRE aus dem selbstextrahierenden Archiv (.bin) zu entpacken (z.B.nach /usr/lib/Java6) und dann manuell die Pfade zum JRE zu setzen (PATH und JAVA_HOME etc.).

Bitte beachten Sie hier auch die Installationsanweisungen von Sun – die ebenfalls im oben genannten Verzeichnis und auf der Java Website (s.o.) zu finden sind!

Ob Java korrekt installiert wurde, können Sie in einer Konsole überprüfen, indem Sie den Befehl „java -version“ ausführen. Es sollte in etwa folgende Ausgabe erscheinen:

```
java version "1.6.0"  
Java(TM) SE Runtime Environment (build 1.6.0-b105)  
Java HotSpot(TM) Client VM (build 1.6.0-b105, mixed mode, sharing)
```

Steht dort etwas ganz anderes, haben Sie entweder die falsche Version installiert, oder auf Ihrem System ist noch eine andere Java VM installiert.

Robot Loader

Der Robot Loader wurde entwickelt, um komfortabel neue Programme in den Robot Arm und alle Erweiterungsmodule laden zu können (sofern diese über einen Mikrokontroller mit kompatibelem Bootloader verfügen). Weiterhin sind ein paar nützliche Zusatzfunktionen integriert, wie z.B. ein einfaches Terminal-Programm.

Den Robot Loader selbst braucht man nicht zu installieren – das Programm kann einfach irgendwo in einen neuen Ordner auf die Festplatte kopiert werden. Der Robot Loader befindet sich in einem Zip-Archiv auf der CD-ROM:

```
<CD-ROM-Laufwerk>:\Software\RobotLoader\RobotLoader.zip
```

Dieses müssen Sie nur irgendwo auf die Festplatte entpacken – z.B. in einen neuen Ordner C:\Programme\RobotLoader (o.ä.). In diesem Ordner finden Sie dann die Datei RobotLoader.exe und können sie mit einem Doppelklick starten.

Das eigentliche Robot Loader Programm liegt im Java Archive (JAR) Robot-Loader_lib.jar. Dieses können Sie alternativ auch von der Kommandozeile aus starten.

Unter Windows:

```
java -Djava.library.path=".lib" -jar RobotLoader_lib.jar
```

Linux:

```
java -Djava.library.path=".lib" -jar RobotLoader_lib.jar
```

Diese lange -D Option ist notwendig, damit die JVM auch alle verwendeten Bibliotheken finden kann. Unter Windows braucht man das aber nicht und kann einfach die .exe Datei zum Starten verwenden und für Linux gibt es ein Shell Skript „RobotLoader.sh“. Das Skript muss evtl. zunächst noch ausführbar gemacht werden (chmod -x ./RobotLoader.sh). Danach kann man es in einer Konsole mit „./RobotLoader.sh“ starten.

Es empfiehlt sich, eine Verknüpfung auf dem Desktop oder im Startmenü anzulegen, um den Robot Loader bequem starten zu können. Unter Windows geht das z.B. einfach indem man rechts auf die Datei RobotLoader.exe klickt und dann im Menü „Senden an“ auf „Desktop (Verknüpfung erstellen)“ klickt.

Wild Thumper Library, WT CONTROL Library und Beispielprogramme

Die Wild Thumper Library und die zugehörigen Beispielprogramme befinden sich in einem Zip-Archiv auf der CD:

```
<CD-ROM-Laufwerk>:\Software\Wild Thumper Examples\Wild Thumper Examples  
[PRO].zip
```

Sie können diese einfach direkt in ein Verzeichnis Ihrer Wahl auf die Festplatte entpacken. Am besten entpacken Sie die Beispielprogramme in einen Ordner auf einer Daten Partition. Oder in den „Eigene Dateien“ Ordner in einem Unterordner „Wild Thumper\Examples“ bzw. unter Linux ins Home Verzeichnis. Das steht Ihnen aber völlig frei.

Die einzelnen Beispielprogramme werden noch später im Softwarekapitel besprochen!

8. Programmer und Loader

Zum Laden eines beliebigen HEX Wild Thumper-Programms vom PC in den Wild Thumper verwenden wir den USB Programmier adapter und unsere RobotLoader Software.

Der im Lieferumfang enthaltene lose USB-Port-Adapter Sender/Empfänger (Transceiver) wird einerseits an einem USB-Port des Computers angeschlossen und andererseits an den Prog/UART- Port der Wild Thumper- Platine.

Der Ladevorgang eines Programms in den Wild Thumper überschreibt automatisch das bereits zuvor vorhandene Programm.



USB-Programmieradapter



RobotLoader-Software

8.1. Der RobotLoader

Wie bereits gesagt wurde der RobotLoader entwickelt um komfortabel neue Programme in den Wild Thumper und alle unsere Roboter übertragen (sofern diese über einen Mikrocontroller mit kompatibelem Bootloader verfügen).

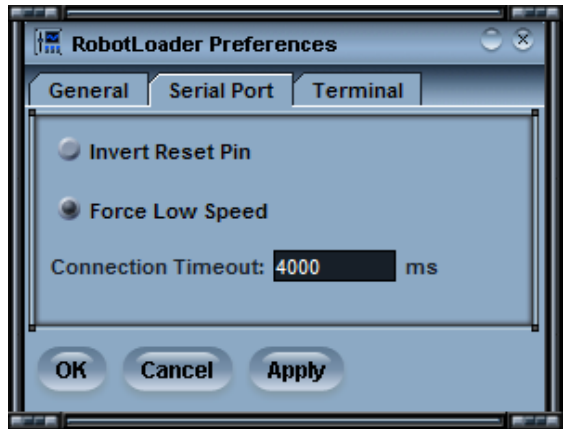
RobotLoader

Mit dem neuen Programmieradapter kann den Wild Thumper drahtlos verbinden.



Weiterhin sind ein paar nützliche Zusatzfunktionen integriert, wie z.B. ein einfaches Terminalprogramm.

Robotloader Einstellungen



Den RobotLoader selbst braucht man nicht installieren – man kann das Programm einfach irgendwo in einen neuen Ordner auf die Festplatte kopieren..

8.2. Anschluss des USB Interfaces – Windows

Linux Anwender können beim nächsten Abschnitt weiterlesen!

Zur Installation des USB Interfaces gibt es mehrere Möglichkeiten. Die einfachste Möglichkeit ist es, **den Treiber VOR dem ersten Anschließen des Geräts zu installieren.**

Auf der CD befindet sich ein Installationsprogramm für den Treiber.

Für **32 und 64 Bit Windows 7, XP, Vista, Server 2003 und 2000** Systeme:

<CD-ROM-Laufwerk>:\Software\USB_DRIVER\Win2k_XP\CDM_Setup.exe

Für alte **Win98SE/Me** Systeme gibt es so ein komfortables Programm leider nicht. Hier muss ein älterer Treiber von Hand installiert werden nachdem man das Gerät angeschlossen hat (s.u.).

Das CDM Installationsprogramm müssen Sie einfach nur ausführen – es gibt nur eine kurze Rückmeldung, sobald der Treiber installiert wurde, sonst passiert nichts weiter.

Dann können Sie das USB Interface an den PC anschließen. **BITTE NOCH NICHT MIT DEM ROBOTER VERBINDEN!** Einfach nur über das USB Kabel mit dem PC verbinden! Dabei sollten Sie darauf achten, die Platine des USB Interfaces nur am Rand oder am USB Stecker bzw. an der Kunststoffwanne des Programmiersteckers anzufassen (siehe Sicherheitshinweise zu statischen Entladungen)! Sie sollten besser keine der Bauteile auf der Platine, Lötstellen oder die Kontakte des Wannensteckers berühren, wenn nicht unbedingt nötig, um statische Entladungen zu vermeiden!

Der zuvor installierte Treiber wird nun automatisch für das Gerät verwendet, ohne dass Sie noch etwas zu tun brauchen. Es erscheinen bei Windows XP/2k kleine Sprechblasen unten über der Taskleiste – die letzte Meldung sollte in etwa „Die Hardware wurde installiert und kann nun verwendet werden!“ lauten!

Wenn Sie das USB Interface doch schon vor der Installation angeschlossen haben (oder Win98/Me benutzen) – auch nicht schlimm. Dann werden Sie von Windows nach einem Treiber gefragt. Auch diese Installationsvariante ist möglich, der Treiber befindet sich auch in entpackter Form auf der CD!

Wenn dies bei Ihnen der Fall ist, erscheint (unter Windows) für gewöhnlich ein Dialog zum Installieren eines neuen Gerätetreibers. Sie müssen dem System dann den Pfad angeben, unter dem es den Treiber finden kann. Bei Windows 2k/XP muss man erst auswählen, den Treiber manuell zu installieren und natürlich keinen Webdienst o.ä. zu suchen. Der Treiber befindet sich in unserem Fall auf der CD in den oben genannten Verzeichnissen.

Also einfach das jeweilige Verzeichnis für Ihre Windows Version angeben und evtl. noch ein paar Dateien, die das System nicht selbstständig findet (sind alle in den weiter unten genannten Verzeichnissen!)

Bei Windows XP oder späteren Versionen folgt oft (hier normalerweise nicht, da die FTDI Treiber signiert sind) noch ein Hinweis das der Treiber nicht von Microsoft signiert/verifiziert worden ist - das ist irrelevant und kann bedenkenlos bestätigt werden.

Inbetriebnahme

Für 32 und 64 Bit Windows 7, XP, Vista, Server 2003 und 2000 Systeme:

<CD-ROM-Laufwerk>:\Software\USB_DRIVER\Win2k_XP\FTDI_CDM2\

Überprüfen, ob das Gerät richtig angeschlossen ist

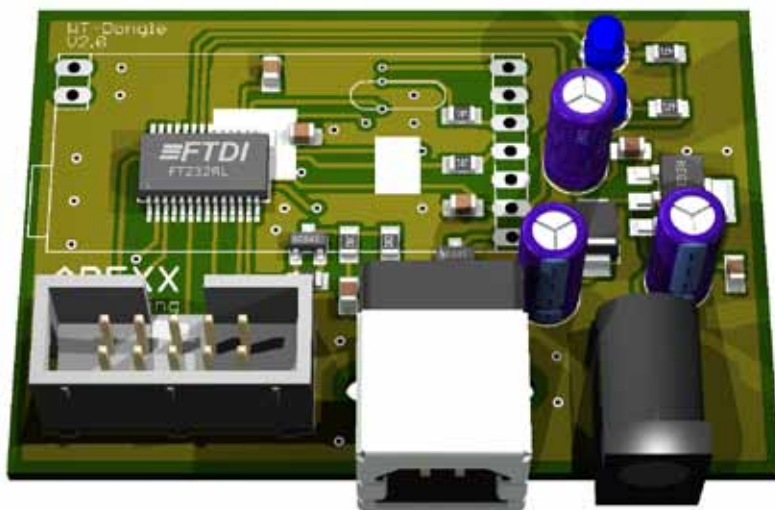
Um zu überprüfen ob das Gerät korrekt installiert worden ist, kann man unter Windows XP, 2003 und 2000 neben dem Robot Loader auch den Gerätemanager verwenden: Rechtsklick auf den Arbeitsplatz --> Eigenschaften --> Hardware --> Gerätemanager

ODER alternativ: Start --> Einstellungen --> Systemsteuerung --> Leistung und Wartung --> System --> Hardware --> Geräte manager und dort in der Baumansicht unter "Anschlüsse (COM und LPT)" nachsehen ob ein "USB-Serial Port (COMX)" zu sehen ist - wobei das X für die Portnummer steht oder unter „USB-Controller“ nach einem „USB Serial Converter“ suchen!

Treiber später wieder Deinstallieren

Sollten Sie den Treiber jemals wieder deinstallieren wollen (Nein, das tun Sie jetzt bitte nicht - ist nur ein Hinweis falls Sie das jemals brauchen sollten): Wenn Sie das CDM Installationsprogramm verwendet haben, können Sie das direkt über Start-->Einstellungen--> Systemsteuerung-->Software tun. In der dortigen Liste finden Sie einen Eintrag des „FTDI USB Serial Converter Drivers“ – diesen auswählen und dort dann auf deinstallieren klicken!

Wenn Sie den Treiber von Hand installiert haben, können Sie das Programm "FTUNIN.exe" im Verzeichnis des jeweiligen USB Treibers für Ihr System ausführen! Achtung: USB-->RS232 Adapter mit FTDI Chipsatz verwenden meist ebenfalls diesen Treiber!



8.3. Anschluss des USB Interfaces – Linux

Windows Anwender können diesen Abschnitt überspringen!

Bei Linux mit Kernel 2.4.20 oder höher ist der benötigte Treiber schon vorhanden (zumindest für das kompatible Vorgängermodell FT232BM des Chips auf unserem USB Interface, dem FT232R), das Gerät wird automatisch erkannt und Sie brauchen nichts weiter zu tun. Falls es doch mal Probleme gibt, erhalten Sie Linux Treiber (und Support und auch evtl. neuere Treiber) direkt von FTDI:

<http://www.ftdichip.com/>

Unter Linux kann man, nachdem man das Gerät angeschlossen hat, mit:
`cat /proc/tty/driver/usbserial`

anzeigen lassen, ob der USB-Serial Port korrekt installiert worden ist. Mehr braucht man hier normalerweise nicht zu tun.

Allerdings sei noch darauf hingewiesen, dass der Robot Loader unter Windows die D2XX Treiber verwendet und dort die vollständigen USB Bezeichnungen in der Portliste auftauchen (z.B. „USB0 | Robot USB Interface | serialNumber“). Unter Linux sind es stattdessen die virtuellen Comport Bezeichnungen `/dev/ttyUSB0`, `/dev/ttyUSB1` etc.. Es werden auch die normalen Comports angezeigt, als „`dev/ttyS0`“ usw.. Hier müssen Sie ausprobieren welcher Port der richtige ist!

Für Linux ist leider kein so einfach zu installierender Treiber verfügbar der beides bereitstellt. Von daher war es hier sinnvoller die Virtual Comport Treiber, die ohnehin schon im Kernel vorhanden sind, zu verwenden. Die D2XX Treiber würden bei der Installation auch noch einiges an Handarbeit erfordern...

Software Installation abschließen

Das war es auch schon mit der Installation der Software und des USB Interfaces! Jetzt könnten Sie noch die wichtigsten Dateien von der CD auf die Festplatte kopieren (vor allem den kompletten Ordner „Documentation“ und, falls nicht schon geschehen, die Beispielprogramme). Dann müssen Sie nicht ständig die CD suchen, wenn Sie diese Dateien benötigen! Die Ordner auf der CD sind alle so benannt, dass sie eindeutig den jeweiligen Softwarepaketen bzw. der jew. Dokumentation zugeordnet werden können!

Sollten Sie die CD einmal „verlegen“, können Sie die wichtigsten Dateien wie dieses Handbuch, den Robot Loader und die Beispielprogramme auch von der AREXX Homepage downloaden. Dort finden Sie auch Links zu den anderen Softwarepaketen, die Sie benötigen.

8.4. USB Interface Testen und RobotLoader starten

Als nächstes testen wir den Programmupload über das USB Interface. Verbinden Sie bitte das USB Interface mit dem PC (Immer zuerst mit dem PC verbinden!) und danach über das 10-pol. Flachbandkabel mit dem "PROG/UART" Anschluss des Wild Thumpers! (Der Wild Thumper MUSS ABGESCHALTET SEIN!). **Es gibt bei WT zwei Prozessoren und zwei PROG/UART Konnektoren! Jumper JP3 und JP4 müssen eingesteckt sein!** Das 10-polige Flachbandkabel ist mechanisch gegen Verpolung geschützt, sofern man es also nicht mit Gewalt behandelt, kann man es gar nicht verkehrt herum anschließen.



Starten Sie danach den RobotLoader. Je nachdem welche Sprache Sie gewählt haben, können die Menüs natürlich etwas anders beschriftet sein. Auf den Screenshots ist die englische Sprache dargestellt, über den Menüpunkt „Options->Preferences“ und dann bei „Language/Sprache“ kann man die Sprache anpassen (Englisch oder Deutsch) und danach auf OK klicken.

Nach Änderung der Sprache muss man den RobotLoader aber erst einmal neu starten bevor sich etwas ändert!

Port öffnen - Windows



Jetzt können Sie den USB Port auswählen. Sofern kein anderer USB->Seriell Adapter mit FTDI Controller am PC angeschlossen ist, sehen sie in der Portliste nur einen einzigen Eintrag den Sie dann bitte auswählen.

Falls doch mehrere Ports vorhanden sind, können Sie den Port anhand des Namens „Robot USB Interface“ identifizieren (oder „FT232R USB UART“). Dahinter wird noch die einprogrammierte Seriennummer angezeigt.

Sollten keine Ports angezeigt werden, können sie im Menü über „RobotLoader-->Refresh Portlist“ („RobotLoader-->Portliste aktualisieren“) die Portliste aktualisieren!



ACHTUNG!

Mit dem neuen Programmieradapter und der APC-220 kann der Wild Thumper drahtlos kommunizieren.

8.5. Port öffnen – Linux

Unter Linux wird der USB-Seriell Adapter wie ein normaler Comport behandelt. Die D2XX Treiberinstallation von FTDI unter Linux wäre nicht ganz so einfach und die normalen Virtual Comport (VCP) Treiber sind in aktuellen Linux Kernen sowieso schon enthalten. Es funktioniert alles fast genauso wie unter Windows, nur muss man erst noch kurz ausprobieren welchen Namen das Wild Thumper-USB Interface hat und darauf achten, den USB Port nicht vom PC zu trennen solange die Verbindung noch offen ist (ansonsten muss der RobotLoader eventuell neu gestartet werden damit die Verbindung wieder klappt). Die Virtual Comports heissen unter Linux „/dev/ttyUSBx“, wobei x eine Nummer ist, z.B. „/dev/ttyUSB0“ oder „/dev/ttyUSB1“. Die normalen Comports heissen unter Linux „/dev/ttyS0“, „/dev/tty- S1“ etc.. Diese tauchen ebenfalls in der Portliste auf sofern vorhanden.

Der RobotLoader merkt sich – wenn es denn überhaupt mehrere Ports gibt - welchen Port Sie zuletzt verwendet haben und selektiert diesen bei jedem Start des Programms automatisch (generell bleiben die meisten Einstellungen und Selektionen erhalten).

Nun können Sie auf den Button „Connect“ (=„Verbinden“) klicken! Der RobotLoader öffnet nun den Port und testet, ob die Kommunikation mit dem Bootloader auf dem Roboter funktioniert. Es sollte dann unten im schwarzen Feld „Status“ die Nachricht „Connected to: Wild Thumper ...“ o.ä. zusammen mit einer Info über die aktuell gemessene Spannung erscheinen. Falls nicht, probieren Sie es nochmal! Wenn es dann immer noch nicht klappt, liegt ein Fehler vor! Schalten Sie in diesem Fall den Roboter am Besten sofort aus und fangen Sie an mit Fehlersuche!

8.6. SELBSTTEST

Sobald der Wild Thumper eingeschaltet wird leuchtet die Spannung-LED (1) Grün.



Wann kein program geladen ist blinkt LED 13.
Mit ein program geladen leuchtet LED 13.

Main Prozessor programieren mit UART HDR 14 und Jumper JP4 eingesteckt
Motor Prozessor programieren mit UART HDR 3 und Jumper JP3 eingesteckt

Die Status-LED 8, 9 und 10 blinkt beim Upload einer HEX Datei in Main Prozessor. Die Status-LED 6 und 7 blinkt beim Upload einer HEX Datei in Motor Prozessor.

Sobald wir ein Programm starten blinken alle Status-LED auf wie ein Laufflicht.

Wenn das geklappt hat, können Sie ein kleines Selbsttestprogramm ausführen, um die Funktionsfähigkeit aller Systeme des Roboters zu überprüfen. Klicken Sie dazu unten im Robot Loader Fenster auf den Button „Add“ („Hinzufügen“) und wählen Sie die Datei Wild Thumper Examples:
„WTEExample_Selftest\WT_Main.hex“ UND „WT_Motor.hex“
im Beispielverzeichnis aus.

***Für die Selbstest Starten Sie dan die WT Software sehe Kapitel 9.
Die Software Anleitung steht in kapitel 10***

Bei Drahtlose Funktion beide APC-220 Installieren!



ACHTUNG!

Beim Wild Thumper gibt es 2 Prozessoren zum Programmieren!

In dieser Datei befindet sich auch das Selbsttestprogramm im hexadezimalen Format – daher werden solche Programmdateien auch „Hexdateien“ genannt. Die eben ausgewählte Datei taucht anschließend in der Liste auf. So können Sie später auch noch andere Hexdateien von Ihren eigenen Programmen und den Beispielprogrammen hinzufügen. Der Robot Loader kann auch verschiedene Kategorien von Hexdateien verwalten.

Damit lassen sich die Dateien übersichtlicher sortieren. Beispielsweise wenn man mehrere programmierbare Erweiterungsmodule auf dem Wild Thumper hat, oder verschiedene Varianten von Programmen verwendet. Die Liste wird immer automatisch beim Beenden des Programms gespeichert! Es werden hier natürlich nur die Pfade zu den Hexdateien gespeichert – nicht die Hexdateien selbst. Wenn Sie an einem Programm arbeiten, brauchen Sie die Hexdatei nur einmal hinzuzufügen und auszuwählen, und können danach sofort nach jedem erneuten Übersetzen des Programms, das neue Programm in den Mikrocontroller laden (auch per Tastenkombination [STRG+D] oder [STRG+Y], um das Programm direkt nach dem Übertragen zu starten). Unter verschiedenen Betriebssystemem sind die Pfadnamen natürlich komplett anders. Sie können den Robot Loader trotzdem ohne weitere Änderungen direkt unter Windows und Linux verwenden, denn es gibt für Windows und Linux jeweils eine extra Liste.

Sie können nun entweder mit den anderen Beispielprogrammen (Examples) des Robot Arm weitermachen oder mit Ihrer eigenen Softwareprogrammierung anfangen.

9.0. Programmierung des WT

Nun kommen wir so langsam zur Programmierung des Roboters.

Einrichten des Quelltexteditors

Erstmal müssen wir uns eine kleine Entwicklungsumgebung einrichten. Der sog. „Quelltext“ (auch Quellcode oder engl. Sourcecode genannt) für unsere C Programme muss ja irgendwie in den Computer eingegeben werden!

Dazu werden wir natürlich auf gar keinen Fall Programme wie OpenOffice oder Word verwenden! Vielleicht ist das nicht für jeden offensichtlich, deshalb wird es hier explizit betont. Damit kann man zwar gut Handbücher wie dieses hier schreiben, aber zum Programmieren ist das absolut ungeeignet. Quelltext ist reiner Text – ohne jegliche Formatierung. Schriftgröße oder Farbe interessieren den Compiler nicht...

Für einen Menschen ist es natürlich übersichtlicher, wenn bestimmte Schlüsselwörter oder Arten von Text automatisch farbig hervorgehoben werden. Das und noch einiges mehr bietet Programmers Notepad 2 (im folgenden kurz „PN2“ genannt), der Quelltexteditor, den wir verwenden werden (ACHTUNG: Unter Linux müssen Sie einen anderen Editor verwenden, der ähnliches wie PN2 bietet. Es sind für gewöhnlich mehrere bereits vorinstalliert! (z.B. kate, gedit, exmcs o.ä.)). Neben dem Hervorheben von Schlüsselwörtern und ähnlichem (sog. „Syntax Highlighting“) gibt es auch eine rudimentäre Projektverwaltung. Man kann so mehrere Quelltextdateien in Projekten organisieren und in einer Liste alle zu einem Projekt gehörenden Dateien anzeigen lassen. Weiterhin kann man aus PN2 komfortabel Programme wie den AVR-GCC aufrufen und so die Programme bequem über einen Menüeintrag übersetzen lassen. Der AVR-GCC ist ja normalerweise ein reines Kommandozeilenprogramm ohne graphische Oberfläche...

Neuere Versionen von Programmers Notepad finden Sie auf der Projekthomepage: <http://www.pnotepad.org/>

Mit den neuesten Versionen von WINAVR ist es nicht mehr notwendig die Menüeinträge zu erstellen!

ACHTUNG:

In diesem Abschnitt beschreiben wir nicht mehr, wie Sie in PN2 die Menüeinträge erstellen müssen! Mit den neuesten WINAVR Versionen sind diese bereits erstellt.

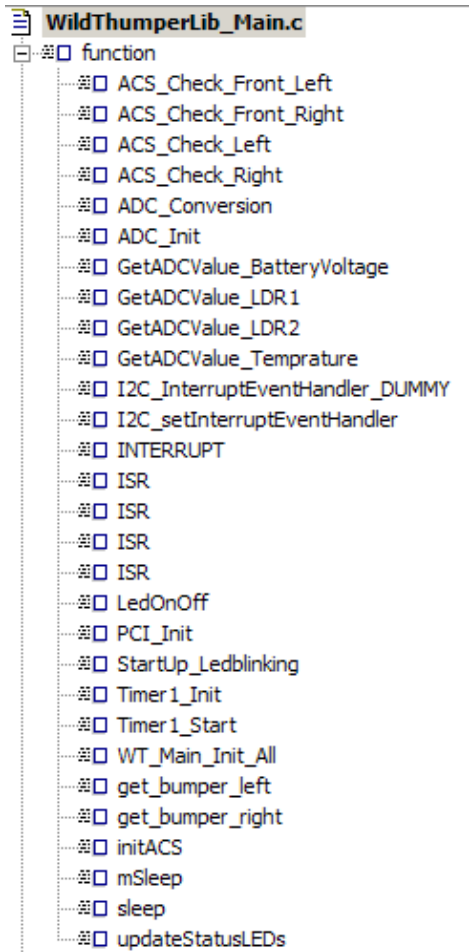
Bei Drahtlose Funktion beide APC-220 Installieren!

Siehe Seite xx „Beispielprojekt öffnen und kompilieren“ wie Sie ein Beispiel Programm öffnen können!

Wenn Sie ein Beispielprojekt geöffnet haben, sollte es im PN2 Schirm etwa so aussehen:

Datei „Wild Thumper Examples.ppg“.

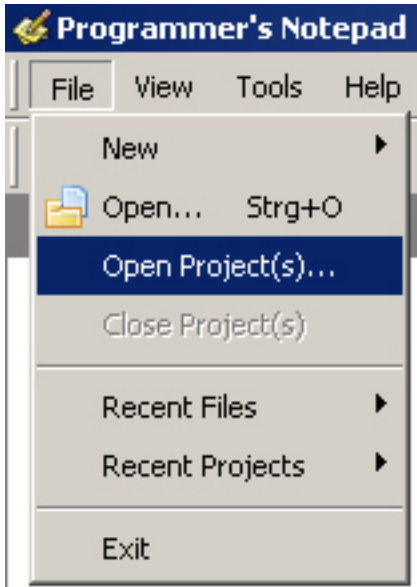
Das ist eine Projektgruppe für PN2, die alle Beispielprogramme sowie die WT Library in die Projektliste („Projects“) lädt.



Links sind alle Beispielprojekte zu sehen, rechts der Quelltexteditor (mit dem angesprochenen Syntax Highlighting) und unten die Ausgabe der Tools (in diesem Fall die Ausgabe des Compilers).

Sie können noch viele andere Sachen in PN2 umstellen und es gibt viele nützliche Funktionen.

Beispielprojekt öffnen und kompilieren



Jetzt probieren wir gleich mal aus, ob auch alles richtig funktioniert und öffnen die Beispielprojekte:

Im Menü „File“ den Menüpunkt „Open Project(s)“ wählen.

Es erscheint ein normaler Dateiauswahl Dialog. Hier suchen Sie bitte den Ordner „WTExamples\“ im Ordner in dem Sie die Beispielprogramme gespeichert haben.

Öffnen Sie nun bitte die Datei „WTExamples .ppg“. Das ist eine Projektgruppe für PN2, die alle Beispielprogramme sowie die WT Library in die Projektliste („Projects“) lädt.

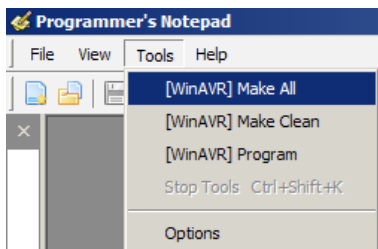
So hat man immer alle Beispielprogramme bequem zur Verfügung und kann sich zu Beginn besser an diesen orientieren oder Funktionen in der WT Library nachschlagen etc..

Öffnen Sie nun das erste Beispielprogramm ganz oben in der Liste („WT_LED“ und die Datei „WT_LED“ selektieren), die am linken Rand des Programmfensters zu sehen ist! Dazu einfach doppelt auf „WT_LED.c“ klicken! Es erscheint ein Quelltexteditor in einem Fenster innerhalb des Programms.

Unten im Programmfenster von PN2 sollte ein Ausgabebereich zu sehen sein – falls nicht, müssen Sie diesen Bereich im Menü mit View->Output aktivieren ODER, falls er zu klein ist, durch „ziehen“ mit der Maus vergrößern (der Mauscursor verändert sich unten im Programmfenster am oberen Rand des grauen Bereichs in dem „Output“ steht in einem recht schmalen Bereich in einen Doppelpfeil...).

Sie können sich das Programm in dem gerade geöffneten Quelltexteditor schonmal kurz anschauen, allerdings müssen Sie hier noch nicht verstehen, was da genau gemacht wird. Das wird weiter unten noch genauer erklärt. Schon mal vorweg: Bei dem grün eingefärbten Text handelt es sich um Kommentare die nicht zum eigentlichen Programm gehören und nur der Beschreibung/Dokumentation dienen.

Darauf gehen wird später noch genauer ein (es gibt auch eine Version dieses Programms OHNE Kommentare – damit man mal sieht, wie kurz das Programm eigentlich ist. Die Kommentare blähen das schon ziemlich auf, sind aber zur Erklärung notwendig. Die unkommentierte Version ist auch praktisch, um den Code in eigene Programme zu kopieren!).



Zunächst wollen wir nur ausprobieren, ob das Übersetzen von Programmen korrekt funktioniert.

Es sollten oben im Tools Menü die beiden eben angelegten Menüeinträge (s. Abb.) vorhanden sein (oder die standardmäßig in PN vorhandenen [WinAVR] Einträge, das ist egal, es funktioniert normalerweise mit beiden).

Klicken Sie jetzt bitte auf „MAKE ALL“!

PN2 ruft nun die oben angesprochene „make_all.bat“ Batch Datei auf. Diese wiederum ruft das Programm „make“ auf. Mehr zu „make“ folgt noch später.

Das Beispielprogramm wird nun übersetzt (das nennt man „kompilieren“ vom englischen „to compile“ bzw. „Compiler“=„Übersetzer“) und eine Hexdatei erzeugt. Diese enthält das Programm in der für den Mikrocontroller übersetzten Form und kann dann später in diesen geladen und ausgeführt werden! Es werden während dem Kompilieren noch viele temporäre Dateien erzeugt (Endungen wie „.o, .lss, .map, .sym, .elf, .dep“). Die brauchen sie alle nicht weiter zu beachten! Mit dem eben angelegten Tool „make clean“ kann man diese bequem löschen. Davon ist nur die Hexdatei für Sie interessant! Die Hexdatei wird beim Aufruf von „make clean“ übrigens nicht gelöscht.

Es sollte nach dem Aufruf des MAKE ALL Menüpunktes folgende Ausgabe erscheinen (hier jedoch stark gekürzt! Einiges kann natürlich etwas anders aussehen):

```
> „make.exe“ all
```

```
avr-gcc -mmcu=atmega644pa -Wall -gdwarf-2 -Os -std=gnu99 -funsigned-char -funsigned-bitfields -fpack-struct -fshort-enums -MD -MP -MT WildThumperLib_Main.o -MF dep/WildThumperLib_Main.o.d -c ../WildThumperLib_Main.c
```

```
avr-gcc -mmcu=atmega644pa -Wall -gdwarf-2 -Os -std=gnu99 -funsigned-char -funsigned-bitfields -fpack-struct -fshort-enums -MD -MP -MT WT_Led.o -MF dep/WT_Led.o.d -c ../WT_Led.c
```

```
avr-gcc -mmcu=atmega644pa -Wl,-Map=WT_LED.map WildThumperLib_Main.o WT_Led.o -o WT_LED.elf
```

```
avr-objcopy -O ihex -R .eeprom -R .fuse -R .lock -R .signature WT_LED.elf WT_LED.hex
```

```
avr-objcopy -j .eeprom --set-section-flags=.eeprom="alloc,load" --change-section-lma .eeprom=0 --no-change-warnings -O ihex WT_LED.elf WT_LED.eep || exit 0
```

```
avr-objdump -h -S WT_LED.elf > WT_LED.lss
```

AVR Memory Usage

Device: atmega644pa

Program: 5514 bytes (8.4% Full)
(.text + .data + .bootloader)

Data: 296 bytes (7.2% Full)
(.data + .bss + .noinit)

```
> Process Exit Code: 0
```

```
> Time Taken: 00:02
```

Wichtig ist ganz unten das „Process Exit Code: 0“. Das bedeutet, dass es beim Übersetzen keinen Fehler gegeben hat. Steht dort ein anderer Code, gibt es einen Fehler im Quellcode, den man korrigieren muss, bevor es klappt. Der Compiler gibt in diesem Fall weiter oben diverse Fehlermeldungen aus, in denen man mehr Infos dazu findet.

Aber bitte beachten Sie, dass „Process Exit Code: 0“ nicht auf ein komplett fehlerfreies Programm hinweist! Denkfehler in Ihrem Programm findet der Compiler natürlich nicht und er kann auch nicht verhindern, dass der Roboter vor die Wand fährt ;-)

WICHTIG: Weiter oben können auch noch Warnungen u.ä. stehen – diese sind oft sehr sehr hilfreich und weisen fast immer auf wichtige Probleme hin! Daher sollten diese immer beseitigt werden! PN2 hebt Warnungen und Fehler farbig hervor, so dass man diese leicht identifizieren kann. Es wird auch die Zeilennummer angegeben, die der Compiler bemängelt. Wenn man auf diese farbig hervorgehobene Meldung klickt, springt PN2 im entsprechenden Editor direkt zu der jew. Zeile.

Auch sehr hilfreich ist die Angabe zum Schluss „AVR Memory Usage“.

Size after:

AVR Memory Usage

AVR Memory Usage

Device: atmega644pa

Program: 5514 bytes (8.4% Full)
(.text + .data + .bootloader)

Data: 296 bytes (7.2% Full)
(.data + .bss + .noinit)

Das bedeutet hier, beim Atmega644 Prozessor, dass unser Programm 5514 Bytes groß ist und 296 Bytes RAM für statische Variablen reserviert sind (dazu kommen noch die dynamischen Bereiche für Heap und Stack, das würde an dieser Stelle aber zu weit führen... halten Sie einfach immer mindestens ein paar hundert Bytes Speicher frei). Wir haben insgesamt 64KB (65536 Bytes) an Flash ROM und 4KB (4096 Bytes) an RAM. Von den 64KB sind 1024 Bytes mit dem Bootloader belegt – also können wir nur 62KB nutzen. Immer darauf achten, dass das Programm auch noch in den verfügbaren Speicher passt! (Der RobotLoader überträgt das Programm nicht wenn es zu groß ist!)

Bei dem Beispielprogramm oben sind also noch 58998 Bytes frei. Das eigentlich recht kurze Beispielprogramm WExample_LED.c ist übrigens nur deshalb schon so groß, weil die WildThumperBaseLibrary mit eingebunden wird! Also keine Sorge, es ist genug Platz für Ihre Programme vorhanden und so kleine Programme brauchen normalerweise nicht so viel Speicher. Die Funktionsbibliothek benötigt alleine nämlich schon mehrere KB vom Flashspeicher, nimmt Ihnen aber auch sehr viel Arbeit ab und daher werden Ihre eigenen Programme meist relativ klein sein im Vergleich zur WildThumperBaseLibrary.

Das eben kompilierte Programm kann nun mit dem RobotLoader in den Roboter geladen werden.

***AUCHTUNG DER WILD THUMPER HAT ZWEI PROZESSOREN !
NICHT VERGESSEN! BEIDE PROZESSOREN PROGRAMMIEREN!!***

Dazu fügen Sie die eben erzeugte Hexdatei in die Liste im RobotLoader mit „Add“ bzw. „Hinzufügen“ ein, selektieren diese und klicken auf den „Upload!“ Button, genau wie Sie es auch schon beim Selbsttestprogramm getan haben. Danach können Sie wieder auf den Terminal wechseln und sich die Ausgabe des Programms anschauen. Die Programmausführung muss dazu natürlich zunächst wieder gestartet werden, im Terminal ist es am bequemsten [STRG]+[S] auf der Tastatur zu drücken oder das Menü zu benutzen (oder einfach ein „s“ senden – nach einem Reset müssen Sie allerdings immer etwas warten, bis die Meldung „[READY]“ im Terminal erscheint!). Auch [STRG]+ [Y] ist eine sehr nützliche Tastenkombination, denn damit wird das aktuell selektierte Programm in den Robot Arm geladen und direkt danach gestartet! Man muss also nicht extra vom Terminal aus wieder auf den Karteireiter „Flash Loader“ wechseln oder das Menü benutzen.

Das Beispielprogramm ist sehr einfach und besteht nur aus einem kleinen LED Lauflicht und etwas Textausgabe.

Bei Drahtlose Funktion beide APC-220 Installieren!

-

9.1. Aktivieren der drahtlosen Steuerung in WT

Es gibt auch bereits vorab kompilierte Programme (HEX Files) auf der CD. Diese Hex-Dateien kann man mit dem RobotLoader direkt in den WT- Prozessor laden.

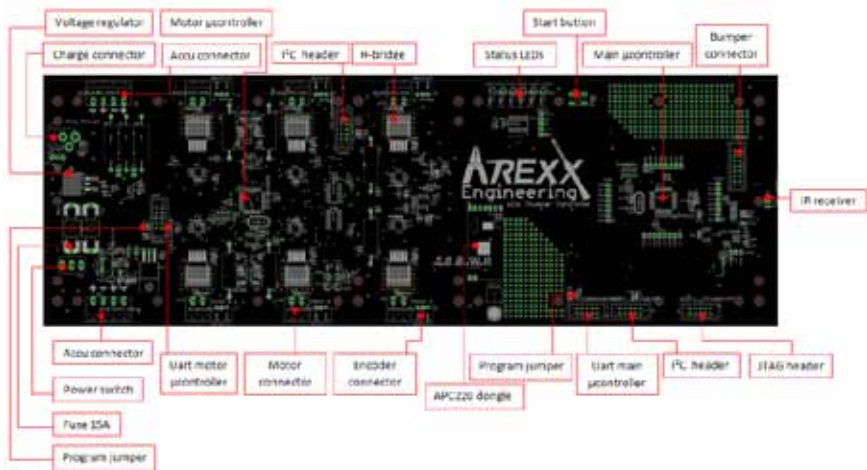


ACHTUNG !
DER WILD THUMPER HAT ZWEI PROZESSOREN !
Vergessen Sie Programjumper JP3 und JP4 nicht die sind Absolut notwendig bei Programmieren!

Als Beispiel laden Sie die HEX-Datei für die Selbst Test in den Hauptprozessor (Wild_Thumper_Main.HEX) und Motorprozessor (Wild_Thumper_Motor.hex) JP3 und JP4 nicht vergessen!.

Starten sie die Wild Thumper Software application und sehe kapitel 10.

Dazu fügen Sie die eben erzeugten Hexdatei in die Liste im RobotLoader mit „Add“ bzw. „Hinzufügen“ ein, selektieren diese und klicken auf die „Upload!“-Taste, genau wie Sie es auch schon beim Selbsttestprogramm getan haben.



Danach können Sie wieder zum Terminal wechseln und sich die Ausgabe des Programms anschauen. Die Programmausführung muss dazu natürlich zunächst wieder gestartet werden, im Terminal ist es am bequemsten [STRG]+[S] auf der Tastatur zu drücken oder das Menü zu benutzen (oder einfach ein „s“ senden – nach einem Reset müssen Sie allerdings immer etwas warten, bis die Meldung „[READY]“ im Terminal erscheint!). Auch [STRG]+ [Y] ist eine sehr nützliche Tastenkombination, denn damit wird das aktuell selektierte Programm in den Wild Thumper geladen und direkt danach gestartet! Man muss also nicht extra vom Terminal aus wieder auf den Karteireiter „Flash Loader“ wechseln oder das Menü benutzen.

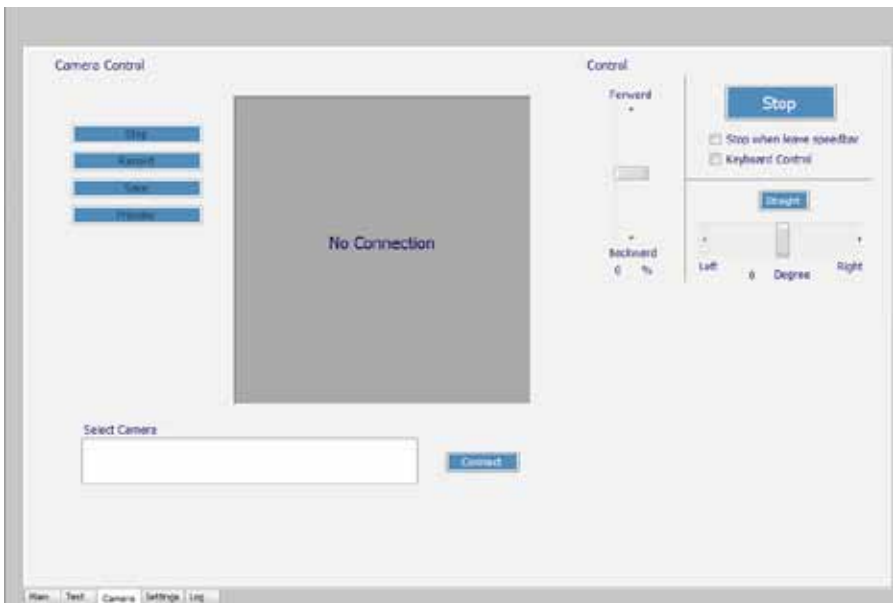
9.1. Installation der drahtlosen Steuerung

In Kapitel 10 wird die komplette Beschreibung der drahtlosen Steuerungsoftware dokumentiert. Vor der Anwendung müssen Sie diese Software auf Ihrem PC installieren.

Kopieren Sie die WT-Anwendung auf Ihre Festplatte oder installieren Sie die Software direkt von der CD.

Start der Anwendung

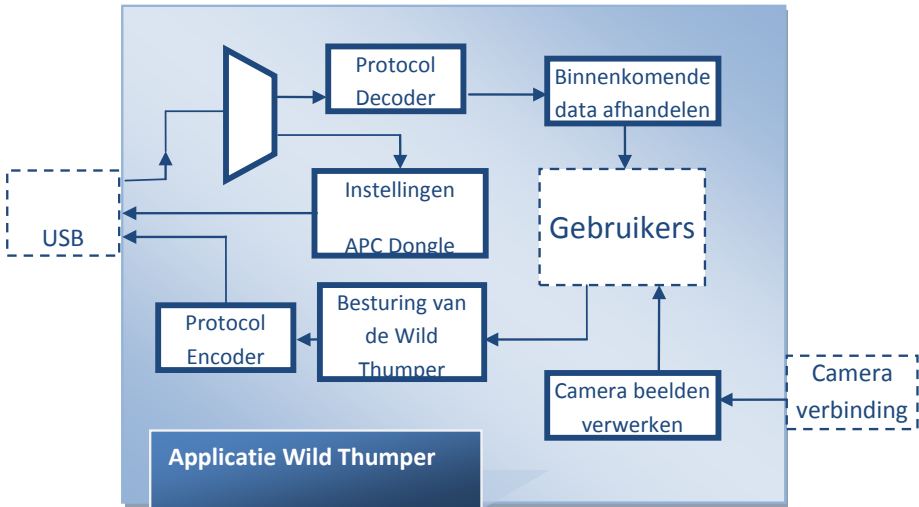
Falls Sie mit einer drahtlosen Kamera arbeiten wollen (Diese muss bereits Installiert sein auf Ihre PC!) sollten Sie diesen der Anwendung hinzufügen. Öffnen Sie dazu den Reiter Kamera und Selektieren Sie die Kamera von Ihre PC.



10. WT DRAHTLOSE STEUERUNG

10.1. Übersicht der drahtlosen Steuerung

Die Anwendung verfügt über zwei Verbindungen zur Außenwelt und zwar die serielle USB-Verbindung mit dem APC-220 Transceiver und die Verbindung zum Kamera-Modul. Die Anwendung selbst besteht aus verschiedenen Funktionen und eine Benutzerschnittstelle, die im nachfolgendem Blockschaltbild erläutert wird.



Weil die Anwendung aus verschiedenen Teilen aufgebaut wurde haben wir Reiterfunktionen zur Auswahl aufgebaut. Jeder Reiter umfasst eigene Funktionen, sodass diese für den Benutzer übersichtlich zugeordnet werden.



Vor dem Testen der drahtlosen Steuerung muss zuerst die zugehörige HEX-Datei mit dem RobotLoader in den Hauptprozessor übertragen werden (Siehe dazu Kapitel 9.1).

Außerdem müssen die beiden APC-220 Module eingesetzt werden.

1. St. auf der Hauptplatine des Wild Thumpers

1. St. auf dem Programmieradapter.

Zudem müssen Sie den Programmieradapter auf eine 5 volt Quelle anschließen.

Nachfolgende Reiter wurden vorbereitet:

Der Reiter Main:

Der Reiter 'main' umfasst folgende Funktionen:

- Die Kommunikationsverbindung mit dem APC Dongle (USB)
- Die Ansteuerung des Wild Thumpers
- Die Darstellung der Messdaten (Stromstärke, Geschwindigkeit, usw..)

Der Reiter Test:

Dieser Reiter wurde zum testen aller Funktionen des Wild Thumpers.

Folgende Tests wurden implementiert:

- Jeder Motor kann einzeln angesteuert werden, wobei die Stromstärke und die Geschwindigkeit des Motors dargestellt werden.
- Alle LEDs können einzeln angesteuert werden.
- Die I²C-Verbindung zwischen Hauptkontroller und Motorkontroller kann getestet werden.
- Die RF-Verbindung kann getestet werden.

Der Reiter Kamera:

Falls am PC eine drahtlose Kamera angeschlossen und auf dem WT installiert wurde können Sie diese vom Wild Thumper stammenden Kamerabilder mit dem „Kamera“-Reiter darstellen. Die Kamerabilder können in der Anwendung auch auf dem PC gespeichert werden. Auch kann man den Wild Thumper aus diesem Reiterbereich ansteuern und dazu die Kamerabilder heranziehen.

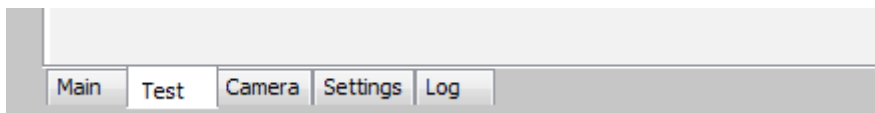
Der Reiter Settings:

Im Bereich des 'Settings'-Reiter können Sie alle Einstellungen vornehmen. Es handelt sich dabei um folgenden Einstellungen:

- Die Parameter für den APC-220
- Die Parameter für die Anwendung

Der Reiter Log:

Zum Rückblick auf die wichtigen Ereignissen wurde der Reiterbereich „LOG“ definiert, in dem die Log-Datei gelesen werden kann.

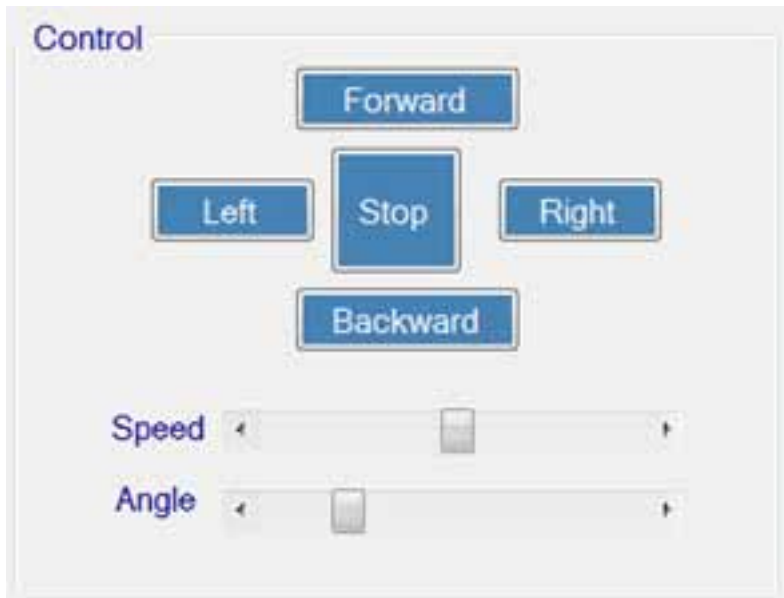


10.2. USB in Visual Basic

Die APC220 kann erst mit der Anwendung kommunizieren nachdem eine Verbindung aufgebaut worden ist. Dazu verwandeln wir die serielle UART-Verbindung des APC220 mit einem FTDI-Chip in eine USB-Verbindung. Der Vorteil dieses Chips liegt nun darin dass dieses neu kreierte „USB-Gerät“ als Gerät behandelt wird, das auf einem virtuellen COM-Port angeschlossen ist. Nach der Inbetriebnahme dieses seriellen Ports können wir die Port-Parameter auf einfachster Weise festlegen.

10.3. Die Steuerung des Wild Thumpers

Wie bereits im vorhergehenden Kapitel beschrieben kann man de Wild Thumper aus den Reiterbereichen 'Main' und 'Kamera' steuern. Zur Ansteuerung verfügt der Wild Thumper über folgendem Steuerpult:



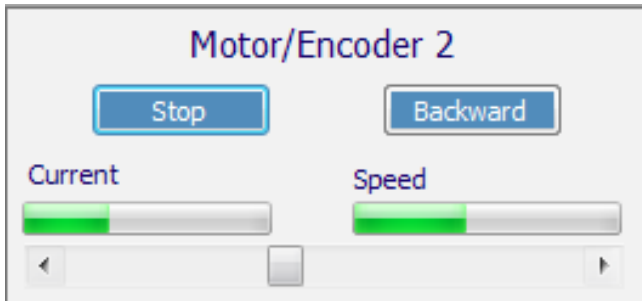
Wie Sie im oben stehenden Bild sehen enthält das Steuerpult fünf Tasten und zwei Schieberegler. Mit den Tasten steuern wir die Richtung des Roboters. Die Schieberegler werden benötigt zur Einstellung der Geschwindigkeit und des Winkels für die Bewegung des Wild Thumpers.

10.5. Testfunktionen

Zum Testen des Wild Thumpers haben wir unter dem Reiter „Test“ ein Testpult entworfen, das Tests für alle Funktionen und Teile des Roboters vorsieht. Nachfolgende Teile des Wild Thumpers können nun getestet werden.

Motortest:

Jeder Motor kann einzeln getestet werden, wobei das Testergebnis die Stromstärke und Geschwindigkeit dieses Motors anzeigt. Sie können den Motor im Vorwärts oder Rückwärtsgang schalten und die Geschwindigkeit regeln.



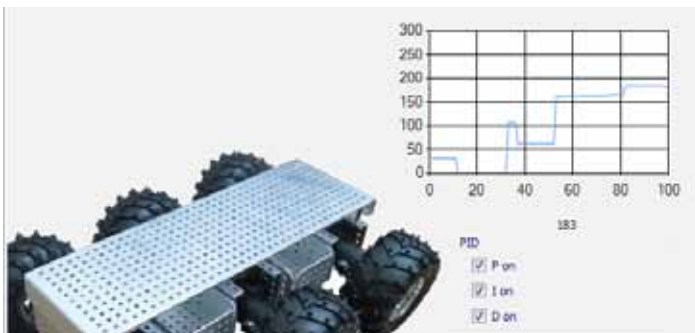
LED-Test:

Alle LEDs können überprüft werden. Nach Ein- oder Abschalten eines LEDs wird dieser Auftrag mit dem WT-Protokoll zum Wild Thumper abgesandt, wo die betreffende LEDs anschließend angesteuert werden.



PID Regler:

Sie können die Verschiedene Funktionen von PID Regler Selbst Einstellen.



Sensortest:

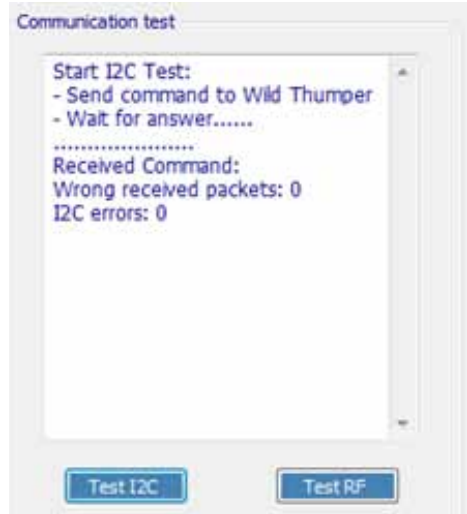
Der Status mehrerer Sensoren wird im Reiterbereich „Test“ dargestellt, so dass Sie als Benutzer den Betrieb aller Sensoren leicht überprüfen können.

Sensors	
ACS Right	Obstacle
ACS Left	Free
Bumper Right	Free
Bumper Left	Free
LDR Right	14
LDR Left	41

Kommunikationstest:

Es stehen zwei Kommunikationstests zur Verfügung: ein RF-Test und ein I²C-Test. Der RF-test testet die drahtlose Verbindung, wobei zunächst ein Datenpaket zur Generierung eines RF-Tests an den Wild Thumper abgesandt wird. Mit dem Paket wird der Roboter beauftragt anschließend zehn Datenpakete mit 100ms Intervall-Längen zu versenden. Sobald alle Bytes einwandfrei eingegangen sind wird der Test als bestanden registriert.

Der I²C-Test wird gestartet indem das System per Funkverbindung das Kommando zum I²C-Test weiter reicht (Siehe Protokollbeschreibung). Anschließend testet der Roboter den I²C durch Übertragung einiger Bytes zum Motorkontroller und zum Ablesen einiger Bytes. Zum Abschluss wird ein I²C-Testbericht generiert und zur Darstellung zurückgesandt zur Anwendung.



10.6. Einstellungen für APPCON APC220

Aus der Anwendung heraus kann man die Einstellungen für das APC220-Modul anpassen beziehungsweise lesen. Dazu steht folgende Schnittstelle zur Verfügung:



Zum Abschluss

Wir hoffen, dass unsere Roboter 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.

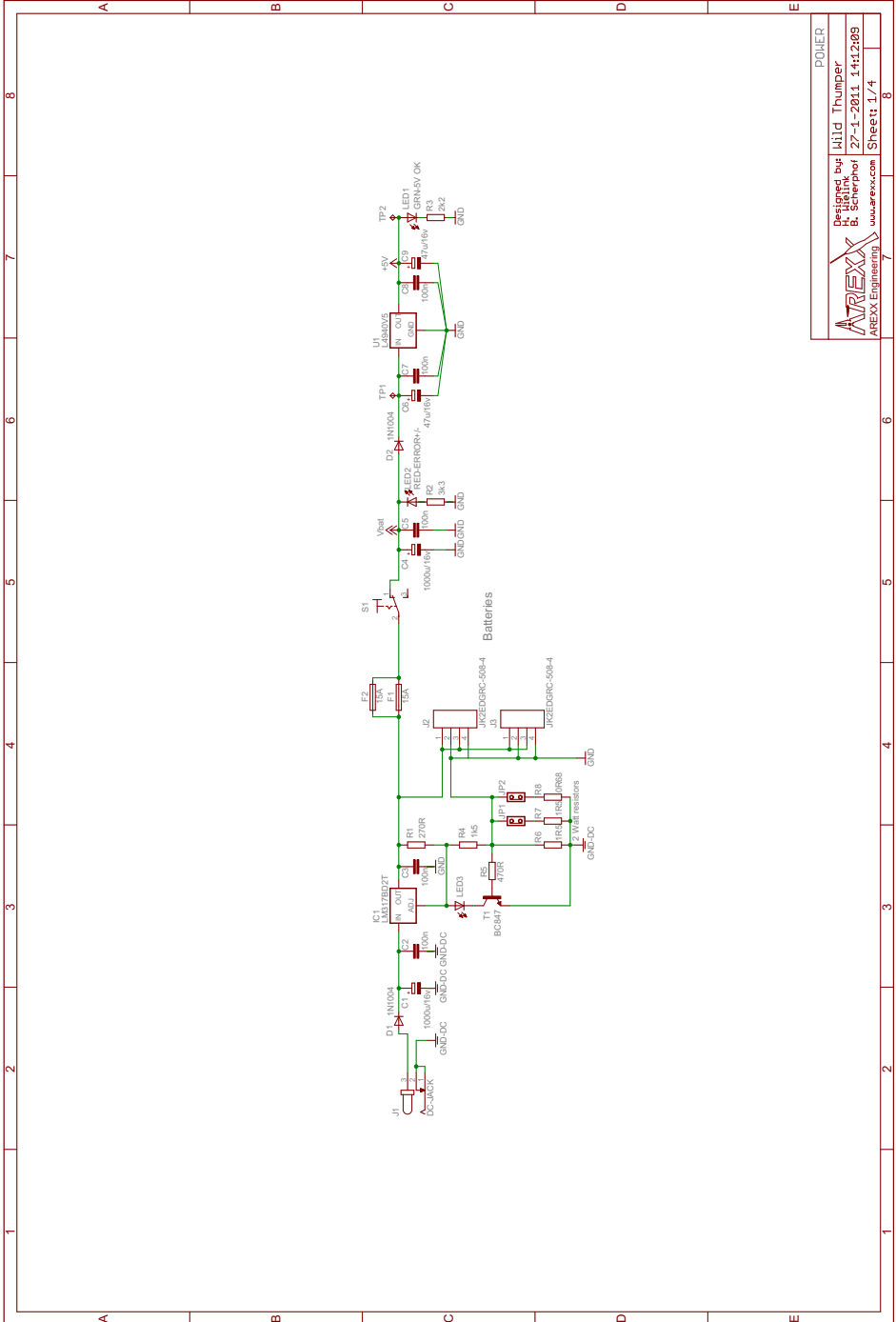
Als Zielsetzung bei der Entwicklung unsere Roboter ASURO, YETI, Caterpillar und Robot Arm haben wir deshalb gewählt:

***AUF ZU EINEM WISSENSCHAFT-
LICHEN GEIST...***



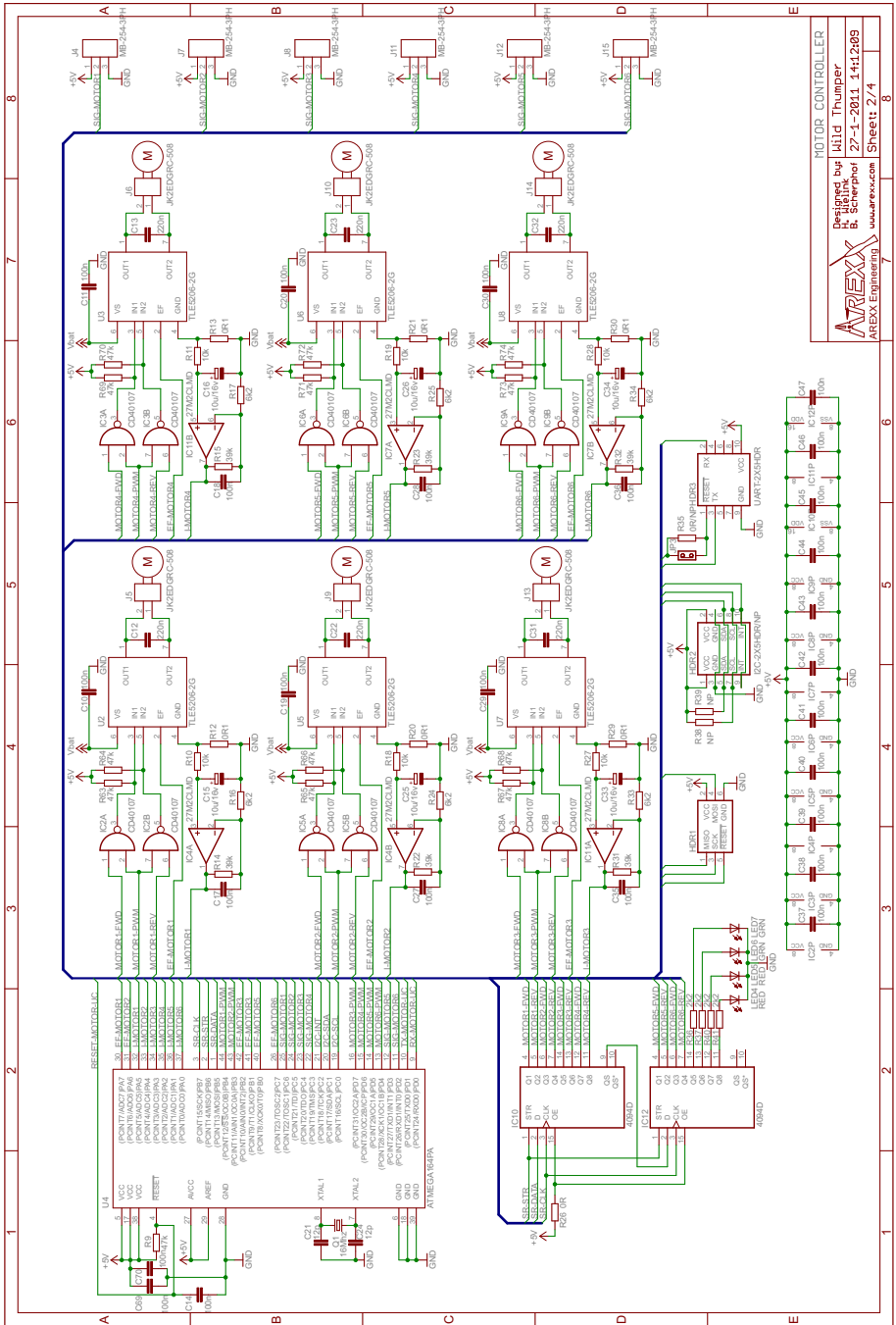
APPENDIX

A. Schaltbild für die STROMVERSORUNG



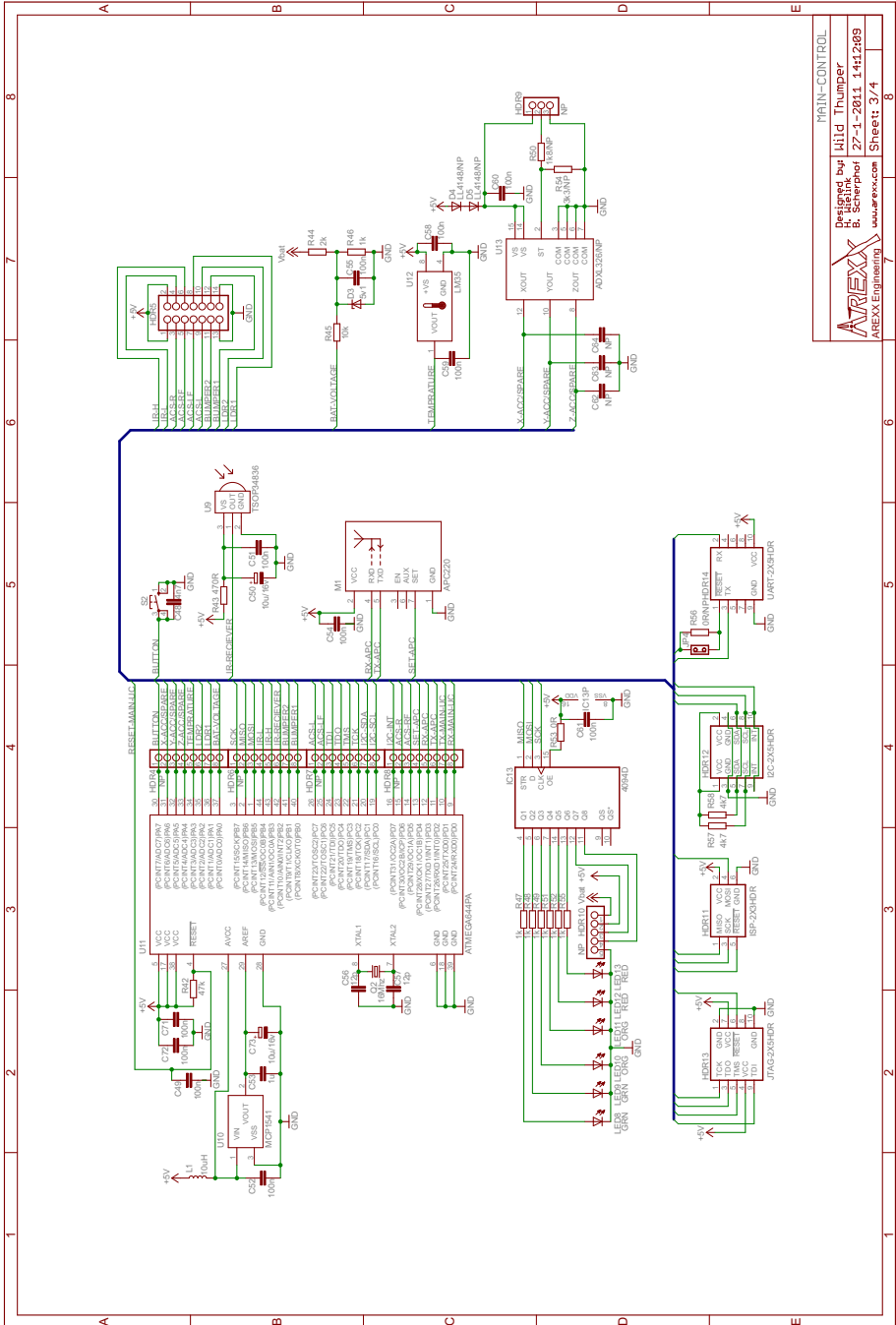
AUREXX Engineering	POWER
	Designed by Wald Thumper B. Schürpfer 27-1-2011, 1411269 www.aurexx.com
Sheet 1/1	8

B. Schaltbild für den WT MOTORSTEUERUNG



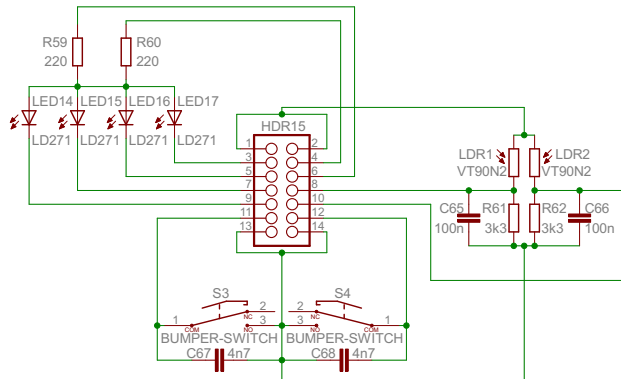
AREXX Engineering
 www.arenx.com
 Resigned by
 Heiko Thumper
 B. Scherphal
 27-1-2011 14:289
 MOTOR CONTROLLER
 Sheet: 2/4

C. Schaltbild für die WT-HAUPTPROZESSOR



MAIN-CONTROL
 Designed by **Hild Thumper**
 B. Scherphof 27-1-2011, 141209
AREXX Engineering
 www.arexx.com
 Sheet: 3/4

D. Schaltbild für die WT-Bumper (Front platine)



E. Programmieradapter für den Wild Thumper

