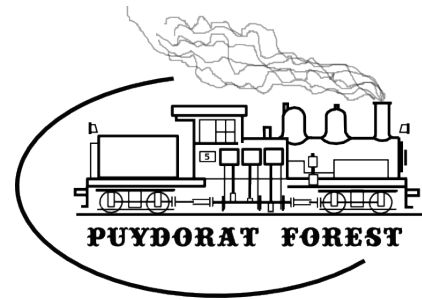


Low Cost PWM für Bürstenmotoren

Eine Anregung für clevere Elektriker, zweiter Teil

Peter Bickel Entwurf 8a



Was bisher geschah

Ich beschreibe den Selbstbau einer Pulsweiten-Steuerung für Gleichstrom-Motoren mit Permanent-Magnet. Das PWM-Signal liefert ein ARDUINO Kleinstcomputer, die Leistungssteuerung übernehmen zwei MOSFET-Bausteine auf einer Platine von Infineon. Im ersten Teil beschreibe ich das Konzept bis zur Gesamtschaltung der Steuerung, in diesem zweiten Teil stelle ich Ihnen das Steuerprogramm vor und dann gehts auf Probefahrt!

Das Steuerprogramm für den Arduino

Sie brauchen jetzt als erstes die Entwicklungsumgebung (IDE).

- in Ihrem HOME-Verzeichnis legen Sie ein Unterverzeichnis /Arduino an
- holen Sie das zu Ihrem Computer passende ZIP-Archiv von der Arduino-Webseite in dieses Verzeichnis
- das Archiv enthält einen Installer, beim Entpacken entsteht das Verzeichnis /Arduino-1.6.8 (Version vom 1.5.2016)
- holen Sie das Steuerprogramm von meiner Webseite, es hat die Endung .ino (wohl inspiriert von arduino) ins Unterverzeichnis /examples
- öffnen Sie jetzt die Entwicklungsumgebung (IDE), klicken Sie auf Datei --> öffnen und gehen ins Unterverzeichnis /examples
- setzen Sie den blauen Balken auf Motorsteuerung.ino und klicken Sie auf öffnen
- die EU beklagt sich: das Programm sollte in einem gleichnamigen Ordner stehen und schlägt Ihnen vor, diese Korrektur gleich zu machen - stimmen Sie zu und das Programm steht auf dem Schirm
- schliessen Sie den ARDUINO ans USB-Kabel an, die grüne LED muss aufleuchten
- klicken Sie auf Werkzeuge --> Platine - Arduino Uno ist bereits mit einem Punkt bezeichnet, wenn nicht, wählen Sie ihn
- klicken Sie auf Werkzeuge --> Port - der Port ist bereits mit einem Häkchen bezeichnet (bei meinem Linux /dev/ttyACM0 (Arduino Uno))
- klicken Sie auf den nach rechts weisenden Pfeil (Hochladen), das Programm wird kompiliert und in den Arduino geladen.
- probieren Sie jetzt Ihre Steuerung aus!

Sowohl die Einstellungen als auch das Hochladen gelingen nur, wenn der Arduino angeschlossen ist, andernfalls bekommen Sie Fehlermeldungen.

Funktion und Einstellungen

Vielleicht wagen Sie jetzt einen Blick ins Programm, das Flussdiagramm in Bild 6 hilft Ihnen dabei. Die Zeilennummern am linken Rand entsprechen denjenigen im Listing, das Sie auf meiner Webseite holen können. ACHTUNG: Dieses Listing ist ein Textfile, den Sie nicht in die ARDUINO-IDE laden können!

ARDUINO-Programme bestehen grob gesagt aus zwei Teilen:

Start: dieser Teil wird nur einmal durchlaufen, hier werden Variablen definiert und ihnen Werte zugewiesen,

Loop: dieser Teil wird ewig durchlaufen, real bis zum Stromausfall, er macht die eigentliche Arbeit: fragt die Schalter ab und reagiert auf ihre Stellung, er hat zwei Teile:

1 der schneller-/langsamer-Teil

Die beiden Stellungen des Schalters *Geschwindigkeit* lassen einen Zähler (count) auf- bzw abzählen, bei jedem Schritt wird der Zählerstand ausgegeben, dann wartet das Programm einen Moment. Diese Wartezeit (delay) bestimmt die Zählgeschwindigkeit. In der Mittelstellung wird nicht gezählt, die Geschwindigkeit bleibt konstant.

Der Zähler zählt von Null bis 255, der grössten mit acht Bit darstellbaren Zahl. Am Ausgang erscheint ein Bruchteil der Versorgungsspannung entsprechend der Formel $\text{Spannung} / 255 * \text{count}$ In Worten: Bei einer Versorgungsspannung von 12V und einem Zählerstand von 128 liegen 6V am Ausgang - Ihr Motor dreht mit halber Geschwindigkeit. Zur Veranschaulichung stellen Sie sich am besten einen Stufenschalter mit 255 Stellungen vor.

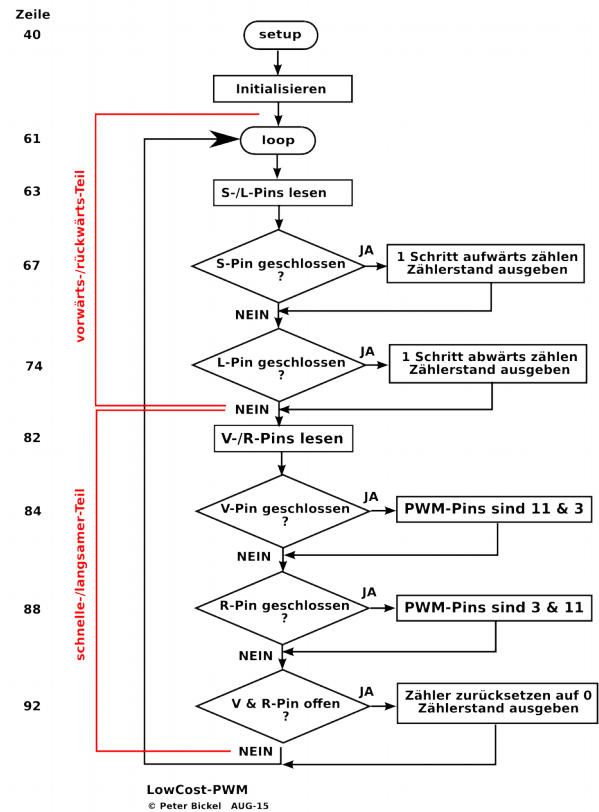


Bild 6

2 der vorwärts-/rückwärts-Teil

Die beiden Schalterstellungen polen die PWM-Pins 3 und 11 des ARDUINO um und ändern so den Stromfluss in der Brücke und damit die Fahrtrichtung. In der Mittelstellung wird schnell auf Null zurückgezählt, das Fahrzeug hält an.

Sie möchten wissen, wie lang ein Durchgang durch *Loop* dauert? Mit der eingebauten Uhren-Funktion lässt sich das feststellen: Werden alle Entscheidungen mit NEIN beantwortet - also ohne jede Schalterbetätigung - macht das Programm 10'000 Durchgänge durch *Loop* in knapp 400 msec, ein Durchgang braucht also rund 0.04 msec oder 0.00004 Sekunden. Ich habe angedeutet, der ARDUINO sei mit dieser Aufgabe bei weitem nicht ausgelastet - hier haben Sie die Bestätigung.

Und so entsteht die Motorspannung

In Bild 7 wird sichtbar, wie das Programm die Ausgangsspannung regelt: Gesteuert wird

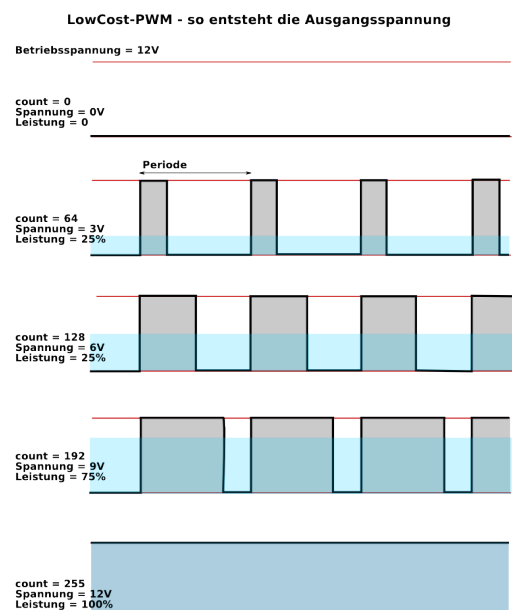


Bild 7

eigentlich die Zeit, während der Strom zum Motor fliesst. Kleine Werte des Zählers *count* bedeuten kurze Impulse und lange Pausen, grosse Werte lange Impulse und kurze Pausen. In Bild 7 ist das durch die grau unterlegten Impulse veranschaulicht. Der Motor hat eine grosse mechanische und elektrische Trägheit und integriert die Impulse zu einer Gleichspannung, veranschaulicht durch die blauen Flächen. Beachten Sie, dass während der Regelung Frequenz und Periodendauer konstant bleiben.

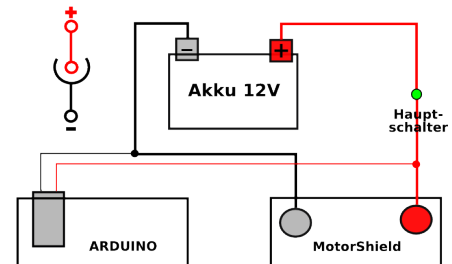
Vorsicht die Mutter des Chipüberlebens

Sowohl der ARDUINO als auch das MotorShield haben sich bei meinen Versuchen als ausgesprochen robust erwiesen. Insbesondere scheinen alle Komponenten gegen die Wirkungen statischer Elektrizität gut geschützt zu sein. Allerdings habe ich weder Spannteppiche noch Plastikböden im Haus. Vergessen Sie aber nie, dass die Anschlüsse des Mikrokontrollers direkt mit den Buchsen am Platinenrand verbunden sind, sie können maximal 40mA liefern. Kurzschlüsse scheinen dem Mikrokontroller nichts auszumachen, was er aber gar nicht mag, sind Überspannungen von mehr als 5V!

Trotz dieser Robustheit ist bei allen Manipulationen Vorsicht am Platz:

- Unterbrechen Sie immer beide Versorgungsspannungen, wenn Sie etwas ändern: Hängen Sie den Akku ab und ziehen Sie den USB-Stecker aus dem ARDUINO, er versorgt die Platine ja mit Strom.
- Kontrollieren Sie lieber einmal zuviel, bevor Sie die Spannungen wieder anlegen.

Beide Boards haben gemäss Datenblatt einen Verpolungsschutz für die Betriebsspannung, wenn Sie mutig sind, probieren Sie natürlich aus, ob er auch wirklich schützt... ☺ Beide Boards sind mehrlagig, mit SMD-Bauteilen bestückt und im Automaten gelötet, sie lassen sich nicht reparieren. Einzig der ATmega328 hat einen Sockel und kann ausgetauscht werden.



ARDUINO UNO steuert Infineon MotorShield
Peter Bickel Juli 2015 © Public Domain

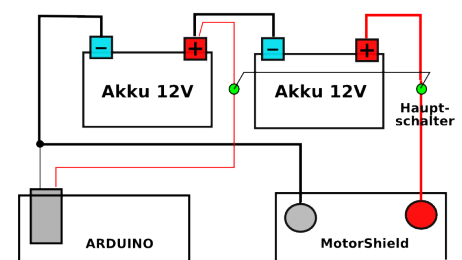


Bild 8

Der Ernstfall

Ich nehme an, dass bei Ihrem Versuchsaufbau auf dem Küchentisch alles zu Ihrer Zufriedenheit funktioniert hat. Dann montieren Sie die fliegende Verdrahtung jetzt auf ein Brettchen und verdrahten sie in Ihrem Fahrzeug. Bild 8 zeigt oben die Verdrahtung mit einem Akku und einem Motor für 12V-Betrieb, unten meine Anordnung mit zwei Akkus für 24V und zwei in Serie betriebenen 12V-Motoren. Dicke Linien sind Leitungen von $\geq 2.5\text{mm}^2$ Querschnitt. Nicht eingezeichnet aber selbstverständlich ist die Sicherung in der $\oplus$ -Leitung. An den Hochstrom-Anschlüssen des MotorShields verwende ich für den Dauerbetrieb 6.3mm-Flachstecker. Der Hauptschalter kommt in die Plus-Leitung, er muss beim Betrieb mit zwei Akkus zweipolig sein, damit auch der ARDUINO abgeschaltet wird.

Für die Stromversorgung des ARDUINO sollten Sie Kabel von wenigstens 0.75mm² vorsehen, was elektrisch zwar nicht nötig ist, Kabelbrüchen durch Vibrationen aber vorbeugt.

Der ARDUINO erhält 12V Betriebsspannung vom Akku, der eingebaute Spannungsregler setzt sie auf 5V herab. Der Pluspol des Steckers liegt auf dem Stift im Zentrum, die Maximalspannung ist 20V!

Jetzt kommt die Prüfung in Ihrem Fahrzeug - bonne chance!

Ändern der Parameter im Programm

Vielleicht haben Sie ja bereits mit den Parametern gespielt - umso besser! Die endgültigen Einstellungen müssen Sie natürlich auf dem Fahrzeug machen, es reagiert anders als ein kleiner freilaufender Motor.

Als Erstes legen Sie eine Sicherungskopie an, zum Beispiel unter dem Namen MotorSteuerung.orig Trennen Sie dann Ihre Schaltung von der Versorgungsspannung, schliessen den ARDUINO wieder über das USB-Kabel an den PC an und starten die Entwicklungs-Umgebung.

Sie finden am Kopf des Programms einen Kasten mit dem Titel *Einstellungen*. Hier können Sie einige Parameter einstellen, indem Sie den Variablen andere Werte geben.

Timer-Einstellung: Dieser Wert bestimmt die PWM-Frequenz, sie sollte wenn immer möglich ausserhalb unseres Hörbereichs liegen.

- mode=1 ergibt eine PWM-Frequenz von rund 31kHz, das liegt weit ausserhalb des Hörbereichs und ist die erste Wahl. Ändern Sie diese Einstellung nur, wenn es nicht anders geht (Motorenprobleme)
- mode=2 legt die Frequenz auf 4kHz und damit in den hörbaren Bereich, ihre Motoren werden singen und das ist nicht angenehm
- Mode=3 ergibt 1kHz und liegt noch unangenehmer im Hörbereich

Geschwindigkeit: Die beiden Werte sind die Grenzen des Zählbereichs.

- countMin = 40: wählen Sie den Anfangswert so, dass die Motoren kräftig aber ohne Ruck anlaufen, Sie werden wohl etwas experimentieren müssen
- countMax = 255: dieser Endwert entspricht der Maximalspannung, er kann tiefer gestellt werden, wenn das Fahrzeug seine volle Geschwindigkeit nicht erreichen soll

Delay: Die Werte von *delay* bestimmen, wie schnell Ihr Fahrzeug auf volle Geschwindigkeit kommt bzw abbremst. Ein Wert von eins bedeutet eine Millisekunde, das Durchzählen der 255 Schritte dauert dann 0.25sec, der



Bild 9

voreingestellte Wert von 30 ergibt also rund 7sec, was ich persönlich sehr angenehm finde. Sie können drei Einstellungen vornehmen:

- $s_delay = 30$: Anstiegszeit, Beschleunigung (s = schneller)
- $l_delay = 30$: Abfallzeit, Verzögerung (l = langsamer)
- $h_delay = 5$: Verzögerung in der Mittelstellung des vor-halt-rück-Schalters (h = halt).

Werte unter 5 können zu hohen Strömen führen und gefährden die Motortreiber, sie werden im Programm auf 5 korrigiert.

Zur Syntax: Die Befehle sind mit einem Strichpunkt abgeschlossen. Was in einer Zeile *hinter einem //* oder *zwischen /* und */* steht, ist Kommentar. Kommentare sind für den Programmierer da, der Compiler überliest sie. Es ist eine gute Gewohnheit, seine eigenen Änderungen im Programm sofort zu dokumentieren, also weitere Kommentare hinzuzufügen.

Nach jeder Änderung sichern Sie das Programm und übersetzen es neu, die Änderungen werden sofort wirksam.

Kühlung

Der BTN8982TA gehört zu den modernsten Bauteilen seiner Art. Er zeichnet sich aus durch minimalen Vorwärtswiderstand und minimale Schaltzeit (Widerstand $8m\Omega$ im durchgeschalteten Zustand, Anstiegs-/Abfallzeit $0.38 \mu sec$), ausserdem ist er mit Sicherungen gegen Übertemperatur und Überstrom ausgestattet. Er sollte sich gemäss Infineon im normalen Betriebszustand auch ohne Kühlkörper nicht überhitzen.



Bild 10

Wenn sich die Bausteine doch überhitzen, schützen sie sich selber durch Abschalten, ein Zeichen dafür, dass sie doch Kühlung brauchen. Setzen Sie dann quer über die Platine einen Kühlkörper auf (Wärmeleitpaste nicht vergessen) und klemmen Sie ihn beidseitig am Rand unverrückbar fest. Einfach wird das nicht sein, denn die Platine ist nicht eingerichtet dafür.

Probefahrt

Bild 9 zeigt die beiden Platinchen auf einem Brettchen montiert und auf meinem Switcher verdrahtet, als Steuerpult dient immer noch die Käseschachtel. Alle Komponenten tun, was sie sollen und den obigen delay-Einstellungen entsprechend ruhig. Die kleinen Kippschalter sind klar nicht ideal, da braucht's etwas Handfesteres. Aber sie reichen aus, wenn Sie jetzt das Feintuning des Programms angehen. Ein tragbarer Computer, auf dem die IDE installiert ist, erleichtert diese Arbeit natürlich enorm.

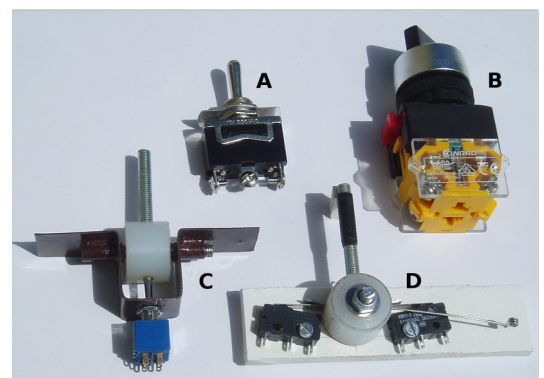


Bild 11

mein Wuppertaler Steuerpult

Ich habe mir in Anlehnung an Bild 2 im ersten Teil mein Steuerpult aufgebaut, Sie sehen es in Bild 10. Das Hauptproblem ist natürlich, dass es Modelle der Originalschalter nicht zu kaufen gibt - verwenden Sie etwas, das Ihnen gefällt, oder wagen Sie eine Eigenkonstruktion. Wegen des oft rauen Betriebs auf unseren Gleisen sind robuste Ausführungen nötig.

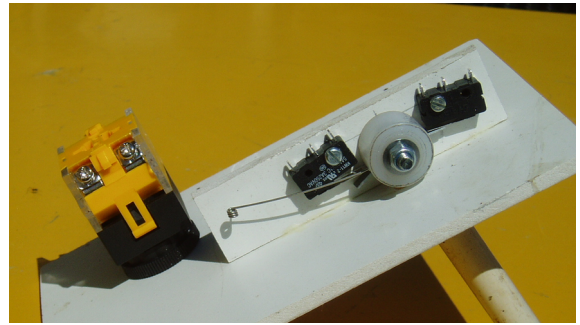


Bild 12

Bild 11 zeigt einige Möglichkeiten:

- A** Diesen Kippschalter gibt es als ON-OFF-ON und (ON)-OFF-(ON) Ausführung, er ist robust, aber etwas schwergängig; er kann für beide Schalter verwendet werden, als Schneller-Langsamer-Schalter ist er mir aber zu hart.
- B** Auch diesen Drehschalter gibt es in beiden Ausführungen, ich habe ihn zum Vorwärts-Rückwärts-Schalter gemacht.
- C** Meine Eigenkonstruktion eines Schneller-Langsamer-Schalters auf der Basis eines Miniatur-Kippschalters: Angenehm weich aber möglicherweise nicht von grosser Lebensdauer...
- D** Meine zweite Eigenkonstruktion mit zwei Mikroschaltern, die Sie in Bild 12 nochmals sehen.

Als in der Steinzeit der Trambahnen noch Direktkontrollen und Noppenschalter die Führerstände beherrschten, war der Vorwärts-Rückwärts-Schalter in der Nullstellung abziehbar und der Wagenführer steckte ihn in die Tasche, wenn er seinen Führerstand verliess: Eine wirklich simple Sicherung gegen Missbrauch war das! Wenn Sie eine Eigenkonstruktion ins Auge fassen, können Sie diese Idee aufnehmen.

FAQs und Hilfe

Es bleiben einige wichtige Fragen zur Beantwortung:

- *Fehler beim Kompilieren?* Am beliebtesten sind natürlich Tippfehler - wundert's? Und: Der ARDUINO muss unbedingt mit dem PC verbunden sein, sonst kann die EU das Programm nicht laden.
- *Darf der ARDUINO gleichzeitig an der Akkuspeisung und am USB-Kabel hängen?* Ja, er darf! Und Sie können das geänderte Programm laden, ohne den ARDUINO vom Akku zu trennen.
- *Kann ein ARDUINO mehrere MotorShields ansteuern?* Ja, er kann! Sie können problemlos eine Vielfachsteuerung einrichten oder jedes Drehgestell mit einem MotorShield ausrüsten.
- *Was heisst ARDUINO UNO r2 oder r3?* Das kleine r bedeutet *Revision* und drückt aus, dass es verschiedene Generationen der Platine gibt. Achten Sie auf Revision 3, dieser Arduino trägt einen 16MHz-Chip. Nur für diesen sind die oben angeführten Timer-Einstellungen gültig.
- *Darf ich das Programm auch für ein anderes Projekt brauchen?* Ja natürlich dürfen Sie, es ist OpenSource! Selbstverständlich werden Sie die Autoren nennen, welche die Vorarbeit geleistet haben.

Weiter unten finden Sie unter «Referenzen» den Link zur Projektseite auf

<http://Puydorat.fr>. Da finden Sie das Programm und die beiden Schemata zum Download, ausserdem die Datenblätter für den MOSFET-Schaltbaustein BTN8982TA und das MotorShield. Auch sammle ich zusätzliche Informationen und gebe Antworten auf oft gestellten Fragen (FAQs). Schauen Sie also zuerst da nach, wenn Sie ein Problem haben. Wenn Sie nicht weiterkommen, senden Sie mir eine Mail. Bedenken Sie dabei, dass ich von Windows absolut nichts verstehe!

Glossar

ATMEL	USA -Hersteller von integrierten Schaltungen
Black Box	Kiste mit unbekanntem Inhalt aber nützlichen Funktionen
Bürstenmotor	Gleichstrommotor mit Permanentmagnet und Kollektor
BDCM	Brushed DC-Motor = GleichstromMotor mit Bürsten/Kollektor
C	Programmiersprache mit grosser Verbreitung
Compiler	übersetzt den Sketch in Arduino-Maschinensprache
Download	Transport von aussen in meinen PC
Halbbrücke	steuert nur die Geschwindigkeit (Vor/Rück mit Umschalter)
IDE, dt EU	Integrated Development Environment, EntwicklungsUmgebung
Infineon	deutscher Halbleiter-Hersteller (ex Siemens)
Loader	kopiert das Programm über USB ins RAM des ARDUINO
Loop	Schleife: darin enthaltene Anweisungen werden wiederholt
Low Cost	billig
PWM	PulsweitenModulation, Art der Gleichstrom-Steuerung
quelloffen	open source: alle Informationen sind öffentlich
RAM	Schreib-/Lesespeicher, enthält die Variablen
ROM	Lesespeicher, enthält das Programm
Shield	Arduino-Slang: Platine aus dem Arduino-Arsenal
Sketch	Arduino-Slang: Programm, Endung .ino
SMD	SurfaceMountedDevice: Miniatur-Bauteile auf dem Board
Upload	Transport von meinem PC nach aussen, zB zum Arduino
USB	Universal Serial Bus, Nachfolger der seriellen Schnittstelle
Vollbrücke	steuert Geschwindigkeit <u>und</u> Vor-/Rückwärtslauf

Bilder

alle Bilder vom Autor

Referenzen

<http://ww1.microchip.com/downloads/en/AppNotes/00905B.pdf> (en)
Grundlagenartikel zu Bürstenmotoren und ihre Ansteuerung mit H-Brücken

<https://de.wikipedia.org/wiki/Vierquadrantensteller> (de)
Verständliche Einführung in die H-Brücken-Technik

https://de.wikipedia.org/wiki/Arduino_%28Plattform%29 (de)
Kurzer Artikel mit Aktualität

<http://www.robotpower.com> (en)
Das ARDUINO-Programm beruht auf einem Beispielsketch des Robot Power Teams für die MegaMoto-Steuerplatine - danke für diese Hilfe!

<http://arduino.cc> (en, teilweise de)
Die ARDUINO-Hauptseite, hier finden Sie die Entwicklungsumgebung für Ihren PC und eine Fülle von Anregungen und Hilfen, wenn Sie selber Projekte machen wollen. Ein Teil der Seite ist übersetzt, vor allem die Tutorials.

<http://playground.arduino.cc/Deutsch/HomePage> (de)
Übersetzung einiger Tutorials von obiger Seite

<http://www.element14.com/community/groups/arduino?ICID=hp-arduino-topgroups> (en)
Farnell element14 vertreibt in seinem Shop das MotorShield

<http://www.infineon.com> (en)

Der Hersteller des Bausteins BTN8982TA, hier gibts Datenblätter und Application Notes

<http://Puydorat.fr> -> Puydorat Forest -> Projekte -> Elektrik+Elektronik -> Low Cost PWM
hier finden Sie das Programm für den ARDUINO, allerhand Datenblätter und Application Notes zum Thema und ergänzende Seiten zu diesem Artikel
eMail: Peter.Bickel@orange.fr

Bücher

MARGOLIS, Michael: Arduino Kochbuch, O'Reilly 2012 € 58.-

Einfache Anwendungen für fast alle Lebenslagen, eine wirklich gute Einführung

http://www.netzmafia.de/skripten/hardware/Arduino/Arduino_Programmierhandbuch.pdf

Seite von Prof. Jürgen Plate, Fachhochschule München

Einführungsbuch (freier Download von dieser Seite)

Material

Decken Sie sich bei den grossen Elektronik-Versendern ein, sie verkaufen auch Kleinmengen. Conrad hat in Frankreich eine Niederlassung, das ist für mich sehr praktisch, aber Reichelt, ELV & Co tun den gleichen Dienst. Und vergessen Sie eBay nicht!

ARDUINO UNO r3

Im Shop von arduino.cc oder beim Elektronik-Versender (zB Conrad 191789, € 28), deutlich billiger sind die Clones aus China bei eBay. Auf dem Board sollte ein ATmega328 sitzen, ob gesteckt oder aufgelötet (SMD) spielt keine Rolle (ab € 10)

MotorShield

Im Shop von [element14.com](http://www.element14.com), keine andere Bezugsquelle bekannt.

Schalter

Verwenden Sie für den Dauerbetrieb keine Mini-Schalter! Für die Fahrtrichtung ist ein Drehschalter mit drei Stellungen nötig ON-OFF-ON (zB 704675 eignet sich gut). Als Kippschalter (ON)-OFF-(ON) eignet sich zB 700596, er ist robust aber etwas schwergängig. Sie bauen sich wohl besser selber etwas mit robusten Mikroschaltern.

Kontaktleisten

Conrad 393491 (männlich) und 733755 (weiblich) in Zehnerstangen zum Abknipsen

Flachstecker 6.3mm

Conrads Angebot ist kaum überschaubar. Unter der Referenz 804983 bekommen Sie zu günstigem Preis ein Sortiment von 95 Stück.



Bild 13