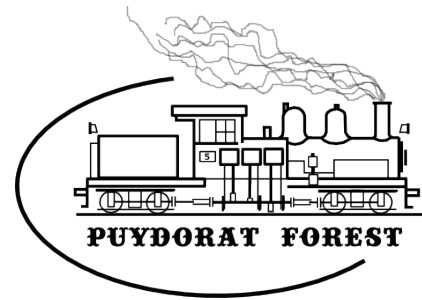


Low Cost PWM für Bürstenmotoren

Eine Anregung für clevere Elektriker, erster Teil

Peter Bickel



Präludium

Meine Fahrzeuge sind mit 12V-GleichstromMotoren mit PermanentMagneten ausgerüstet, sogenannten BürstenMotoren oder *Brushed DC-Motors (BDCM)*. Zur Steuerung verwende ich die Pulsweiten Modulation (PWM) in Vollbrücken- oder H-BrückenSchaltung, sie regelt die Geschwindigkeit sehr feinfühlig, erlaubt mit minimalem Aufwand von Vorwärts auf Rückwärtsfahrt umzuschalten und arbeitet beinahe verlustfrei. Nach mehreren Versuchen mit anderen Steuerungskonzepten meine ich, die mir zusagende Lösung gefunden zu haben.

Zur Steuerung von Bürstenmotoren mit pulsweitenmoduliertem Gleichstrom gibt es im Internet viel zu lesen, ich werde darum nicht in die Details dieser Technik gehen.

Der Anlass zu diesem Projekt

Im Dezember 2014 kündigte Infineon eine Experimentierplatine (*DC Motor Control Shield*, im weiteren kurz MotorShield) mit zwei Halbbrücken-ICs an, im Februar 2015 dieses Jahres war sie dann auch erhältlich. Mit dieser Platine wird der Aufbau einer PWM-Steuerung zum Kinderspiel: Füttern Sie die kleine Platine vorn mit einem PWM-Signal und schliessen Sie hinten Ihren Motor an - fertig! Die Platine ist kompatibel zum Mikrocomputer ARDUINO. Es bietet sich also an, diesen zur Ansteuerung zu verwenden, obwohl das PWM-Signal selbstverständlich auch anders erzeugt werden kann.

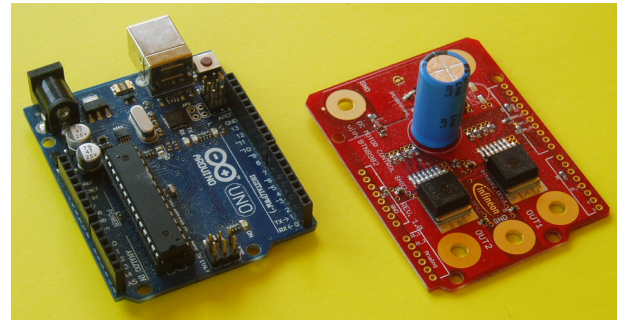
Damit Sie die Katze nicht im Sack kaufen: In der Tabelle finden Sie die Grenzwerte des auf dem MotorShield verbauten Bausteins. Im Dauerbetrieb können Sie bis zu zwei Dritteln dieser Grenzwerte gehen.

BTN8982TA	Maximalwerte
Versorgungsspannung	40 Volt
Logik-Eingänge	5 Volt
Strom kontinuierlich	44 Ampère
Strom PWM	60 Ampère
Strombegrenzung	77 Ampère
PWM-Frequenz	bis 30kHz

Wenn sich hier jetzt Ihre Stirn in Falten legt und Sie Ihren Zeigefinger zum Umblättern netzen, bitte ich Sie um einen Moment Geduld. Ich gehe nämlich davon aus, dass in Ihrer Umgebung bereits viele «ARDUINOs» ganz unauffällig ihre Arbeit tun: Im Geschirrspüler, im Auto, im Fernsehapparat, in Ihrem Smartphone. Wir gehen heute total selbstverständlich mit hochkomplexen Apparaten um, deren Funktionsweise wir kaum verstehen. Warum sollte also in Ihrer Gartenbahn nicht auch ein ARDUINO werkeln?

Der pragmatische Weg

Bild 1 zeigt die beiden Platinchen im Auslieferungszustand, links der ARDUINO, rechts das MotorShield.



Stellen Sie sich nun auf den Standpunkt, dass Sie den Arduino und das MotorShield weder verstehen können noch wollen. Betrachten Sie einfach beides als Black Box, genau wie Ihre anderen elektronischen Geräte, und folgen Sie mir auf den pragmatischen Weg. Sie kommen so für knapp € 60 zu einem Vollbrücken-Steuergerät für 24V/30A Dauerleistung. Wenn das kein Low Cost PWM ist!?

Sie brauchen keinerlei Programmierkenntnisse, denn das im ARDUINO arbeitende Programm stelle ich Ihnen zur Verfügung. Sie brauchen es nur zu laden, es arbeitet sofort los. Ich werde Sie natürlich auf einige Einstellmöglichkeiten hinweisen, die Sie aber nicht zu übernehmen brauchen. Und natürlich brauchen Sie einige Kleinteile sowie eine Lötstation mit feiner Spitze und ein Digital-Multimeter für € 10.-

Der gründliche Weg

Es ist keineswegs meine Absicht, Sie aufs Glatteis der Computerprogrammierung zu führen. In diesem kleinen Arduino-Platinchen liegt aber einige Verführungskraft - sehen Sie sich also vor! Und dieses Projekt kann durchaus Anlass dazu sein, einmal hinter die Kulissen zu schauen. Als erstes werden Sie wohl verstehen wollen, was das Programm eigentlich macht. Und dann werden Sie anfangen, es abzuändern und zu ergänzen. Denn tatsächlich ist der Arduino mit dieser PWM-Aufgabe keineswegs ausgelastet.

Sie können zum Beispiel Erweiterungen schreiben, die die Akkuspannung überwachen, im Tunnel automatisch das Licht einschalten, die Geschwindigkeit Ihres Fahrzeugs messen und anzeigen oder - etwas anspruchsvoller - den Abstand zum vorausfahrenden Fahrzeug messen und vor einem Auffahrunfall warnen. Sie werden sehen, *l'appétit vient en mangeant*: Der Appetit kommt mit dem Essen. Was Sie brauchen, finden Sie im Internet, vieles ist auch auf Deutsch zu lesen. Von *Low Cost* ist dann natürlich keine Rede mehr, denn Sie werden sich einarbeiten müssen und das kostet Zeit und Mühe. Es macht andererseits viel Spass und die Gefahr ist gross, dass Sie zum ARDUINO-Fan mutieren.

Mein persönliches Steuerkonzept

Meine Fahrzeuge sind zum Aufsitzen da und sie werden auch so gesteuert, nämlich wie eine Strassenbahn von einem Steuerstand aus. Und genau so funktioniert auch diese PWM-Steuerung. Ein Fernsteuersender kommt nicht vor meinen Bauch, das überlasse ich den Modellfliegern und den H0ern. Das hier verwirklichte Konzept habe ich in vielen modernen Strassenbahnen gesehen, Bild 2 zeigt



den Bedienbereich der Wuppertaler Schwebbahn: Die Kubel links ist der vorwärts-halt-rückwärts-Schalter, mit dem Knebel rechts wird schneller-langsamer gesteuert.

Genug der Vorbemerkungen, jetzt gehts auf den pragmatischen Weg!

Die PWM-Erzeugung mit dem Arduino

Der ARDUINO erblickte vor zehn Jahren in Italien das Licht der Welt. Die Idee dahinter war, Mikroprozessoren handhabbar zu machen, Barrieren abzubauen und Berührungängste aufzulösen. Hard- und Software sollten bereits Schülern erlauben, einfache Projekte zu verwirklichen, es war auch vorgesehen, einen Teil der Platinen gratis an Schulen zu verteilen. Alle Teile sind quelloffen und unterdessen entwickelt eine riesige Fangemeinde für den ARDUINO. In den zehn Jahren seit seiner Erfindung ist auch die Platinen-Familie stetig grösser geworden: Heute sind neue Platinen mit leistungsfähigen Prozessoren und viele Zusatzplatinen erhältlich. Und im Internet und in zahlreichen Büchern gibt es Programme für alle Lebenslagen.

Ich arbeite hier mit dem *ARDUINO UNO*, bei dem der Prozessor in einem Sockel steckt und ausgewechselt werden kann. Die Hardware besteht aus einer etwa 5x7cm kleinen Platine mit einem Mikroprozessor von ATMEL (ATmega328P) und etwas Zugemüse. Verwenden Sie bitte nur den UNO! Die neueren Boards tragen andere Prozessoren, deren Programmierung von der des UNO abweichen kann.

Die Programme entwickeln Sie auf dem PC in der Sprache C und laden sie dann über ein USB-Kabel in den Speicher des ARDUINO. Das Platinchen kann bei allen Elektronik-Versendern für knapp € 30 bestellt werden. Auf eBay finden Sie es auch für € 10.- *made in China*.

Die Entwicklungsumgebung für den PC

Die Entwicklungsumgebung (EU) ist rund 60MB gross und steht auf der ARDUINO-Webseite zum Download bereit, es gibt sie für Windows, MacOS und Linux, sowie für 32bit- und 64bit-Systeme. Sie enthält einen Editor zum Schreiben des Codes und einen Übersetzer (Compiler), der Ihr Programm auf Fehler untersucht und es dann in Maschinensprache übersetzt. Dann wird es über die USB-Schnittstelle in den Speicher des ARDUINO gefüllt. Ihr Programm startet sofort und läuft so lange, wie der ARDUINO an der Spannungsversorgung liegt.

Auch wenn Sie nicht programmieren wollen, brauchen Sie die EU zum Laden des PWM-Programms in den ARDUINO und allenfalls zum Anpassen des einen oder anderen Parameters an Ihre Wünsche. Weitergehende Änderungen des Programms sind nicht nötig, ausser Sie wollen herumprobieren.

Die Infineon Entwicklungs-Platine

Diese Platine hat das gleiche Format wie der Arduino und trägt zwei hochintegrierte Motorentreiber BTN8982TA. Jeder enthält je zwei MOSFET-Leistungsschalter und die nötige Ansteuerungselektronik in einem Gehäuse. Die beiden Leistungsschalter sind bei mir als H-Brücke geschaltet und steuern Geschwindigkeit und Drehrichtung eines Motors, ein Relais zum Umsteuern der Drehrichtung ist nicht nötig. Die Platine hat einen Schutz gegen Verpolung der Versorgungsspannung. In den ICs eingebaut sind Sicherungen gegen

Übertemperatur und Überstrom. Sie wird von Farnell Elements14 zum Preis von rund € 22 verkauft.

Die Bedienelemente

Der Arduino erhält seine Befehle vom Steuerpult, das zwei einpolige Kippschalter mit je drei Positionen trägt, je einen für Drehrichtung und Geschwindigkeit. Das Programm wertet die Schalterstellungen aus und bildet das PWM-Signal für die Motorentreiber.

1 die Funktion Geschwindigkeit

Dieser Kippschalter hat seine Ruhestellung in der Mitte, die beiden anderen Stellungen rasten nicht ein ((ON)-OFF-(ON)). Die drei Positionen haben die Bedeutung schneller - Geschwindigkeit halten - langsamer.

2 die Funktion Drehrichtung

Dieser Kippschalter hat drei einrastende Positionen: *vorwärts* - *halt* - *rückwärts* (ON-OFF-ON). In der Mittelstellung *halt* ist die Brücke stromlos, das Fahrzeug steht still. Wenn Sie auf eine der beiden anderen Stellungen gehen, steuert der ARDUINO den Stromfluss für Vor- bzw Rückwärtsfahrt. Sobald Sie den Geschwindikeitshebel betätigen, rollt das Fahrzeug an.

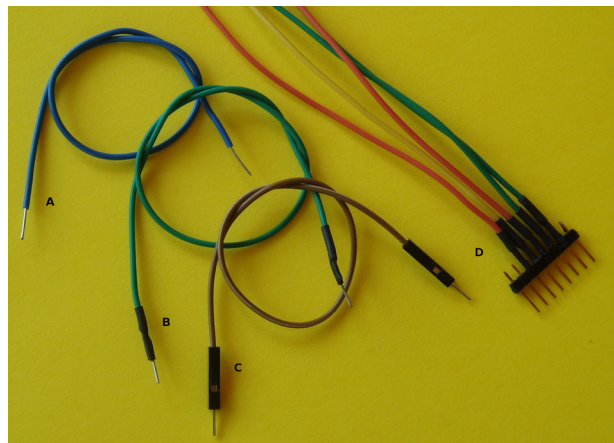
Wenn Sie bei fahrendem Fahrzeug in die *halt*-Stellung gehen oder gar die Drehrichtung ändern, fährt der ARDUINO die Motorspannung herunter und das Fahrzeug hält an. So vermeiden Sie Fehlbedienungen, die rasch zur Überlastung der MOSFET-Transistoren führen können.

Probeaufbau und Funktionskontrolle

1 Material

Im Bild der fabrikneuen Platinen ist gut zu sehen: Das MotorShield hat von Haus aus keine Buchsenleisten. Sie brauchen nur die rechte Seite mit zwei Buchsenleisten von acht bzw zehn Anschlüssen zu bestücken. Natürlich können Sie die Drähte auch direkt auf die Platine löten, ich rate aber ganz entschieden davon ab. Die winzigen Kontaktflächen lösen sich unter der Löthitze rasch vom Isoliermaterial und dann beginnt das Pfriemeln. Was meist mit dem vorzeitigen Ende der Patine ausgeht. Den vier Hochstrom-Anschlüssen habe ich Bananenbuchsen spendiert, Flachstecker 6.3mm sind genau so brauchbar.

Die Buchsenleisten sind für Stecker von 0.5mm gebaut, der schnellste Weg zum Ziel ist isolierter Schalt draht Durchm. 0.5mm, den Sie in verschiedenen Farben erhalten [A]. Etwas aufwendiger sind Stücke von Schaltlitze, die Sie beidseitig mit einem kurzen Stück Schalt draht versehen und mit Schrumpfschlauch isolieren [B]. Und schliesslich finden Sie im Handel fertig konfektionierte Kabel mit Steckern [C]. Für den definitiven Aufbau empfehle ich Ihnen die zu den Buchsenleisten passenden Steckerleisten mit angelöteten Stücken von Schaltlitze [D]. Bild 3 zeigt die vier Möglichkeiten.



Machen Sie sich auch einige Gedanken zur Farbe der Verbindungskabel. Elektrisch spielen diese Farben keine Rolle, Ihnen helfen Sie aber, Fehler zu vermeiden. Bei mir sind rot und schwarz grundsätzlich für die Betriebsspannung vom Akku reserviert, wenn Sie Netzkabel verwenden, werden Sie wohl mit braun und blau vorlieb nehmen müssen. Am Ausgang des MotorShields ist die Polarität je nach Fahrtrichtung wechselnd, wählen Sie also zwei gleichfarbige Leiter. Der mittlere Anschluss GND bleibt offen.

2 Verdrahtung

Bauen Sie sich zunächst ein provisorisches Bedienpult, die Käseschachtel auf dem Bild hat sich prima bewährt. Der Arduino erhält seine Betriebsspannung vorläufig weiterhin vom PC, für den Motor brauchen Sie aber eine unabhängige Spannungsquelle von 12V. Das kann ein Akku sein oder ein Netzteil, ein altes PC-Netzteil tut den Dienst prima. An den Ausgang des MotorShields legen Sie einen kleinen Motor oder fürs Erste Ihr Multimeter, Sie brauchen dann nicht auch noch einen wildgewordenen Motor einzufangen...

Für die Verdrahtung trennen Sie den Arduino vom PC, indem Sie den USB-Stecker abziehen. Auch das 12V-Spannungsquelle bleibt ausgeschaltet. Folgen Sie nun einfach dem Schema in Bild 4 und versuchen Sie, ohne Kurzschlüsse und Wackelkontakte die nötigen Verbindungen zu machen. Ganz wichtig: Die Masse-Kontakte (Ground - GND) von ARDUINO und Motor-Shield müssen miteinander verbunden sein! Die Anschlüsse IN1 und IN2 des Motor-Shields (Pins 3 und 11, Achtung: Die Zählerei beginnt bei 0!) bekommen das PWM-Signal. Die Anschlüsse INH1 und INH2 (Pins 12 und 13) blockieren die beiden Schalter-ICs (inhibit) und müssen mit +5V verbunden werden, damit der Motor läuft. Im Schema habe ich einen «Zündschlüssel» vorgesehen, den Sie zB mit einem Schlüsselschalter realisieren können. Aber ACHTUNG: Dieser Schalter ersetzt natürlich einen Hauptschalter in der Akkuleitung nicht!

Meinen Versuchsaufbau sehen Sie in Bild 5, die Teile sind wie im Schema angeordnet, nur der Schlüsselschalter fehlt. Statt dessen verbindet der blaue Draht die beiden Inhibit-Eingänge mit +5V.

