



Praktisches Beispiel zur Nutzung der PWM des ATTiny84

Das Projekt ist für Anfänger und Lernenden gedacht, stellt aber auch einen praktischen Nutzen dar.

Grundlegende Kenntnisse in Elektronik, Löten und Programmierung mit Arduino sind hilfreich.

Das Projekt baut auf ein Kit von AZ-Delivery „DIY LED Weihnachtsbaum Kit: Weihnachtsbaum Elektronik Bausatz zum Löten - Lötbausatz für einen drehenden Weihnachtsbaum mit LEDs und USB-Anschluss“ auf.

Materialliste:

- 1 1 Weihnachtsbaum Kit
- 2 1 ATTiny84
- 3 3 Transistor BD139
- 4 3 R1-R3 Potenziometer 10kOhm
- 5 3 R4-R6 330Ohm
- 6 3 R7-R9 10kOhm
- 7 1 C1 100nF
- 8 1 C2 4,7μF
- 9 1 C3 10μF
- 10 M1 –M6 und Test sind Steckbuchsen
- 11 S1 –S3 Jumper
- 12 Universalleiterplatte
- 13 18 Flacker-LEDs aus flackernde Teelichtern

Bild2 bestückte Leiterplatte

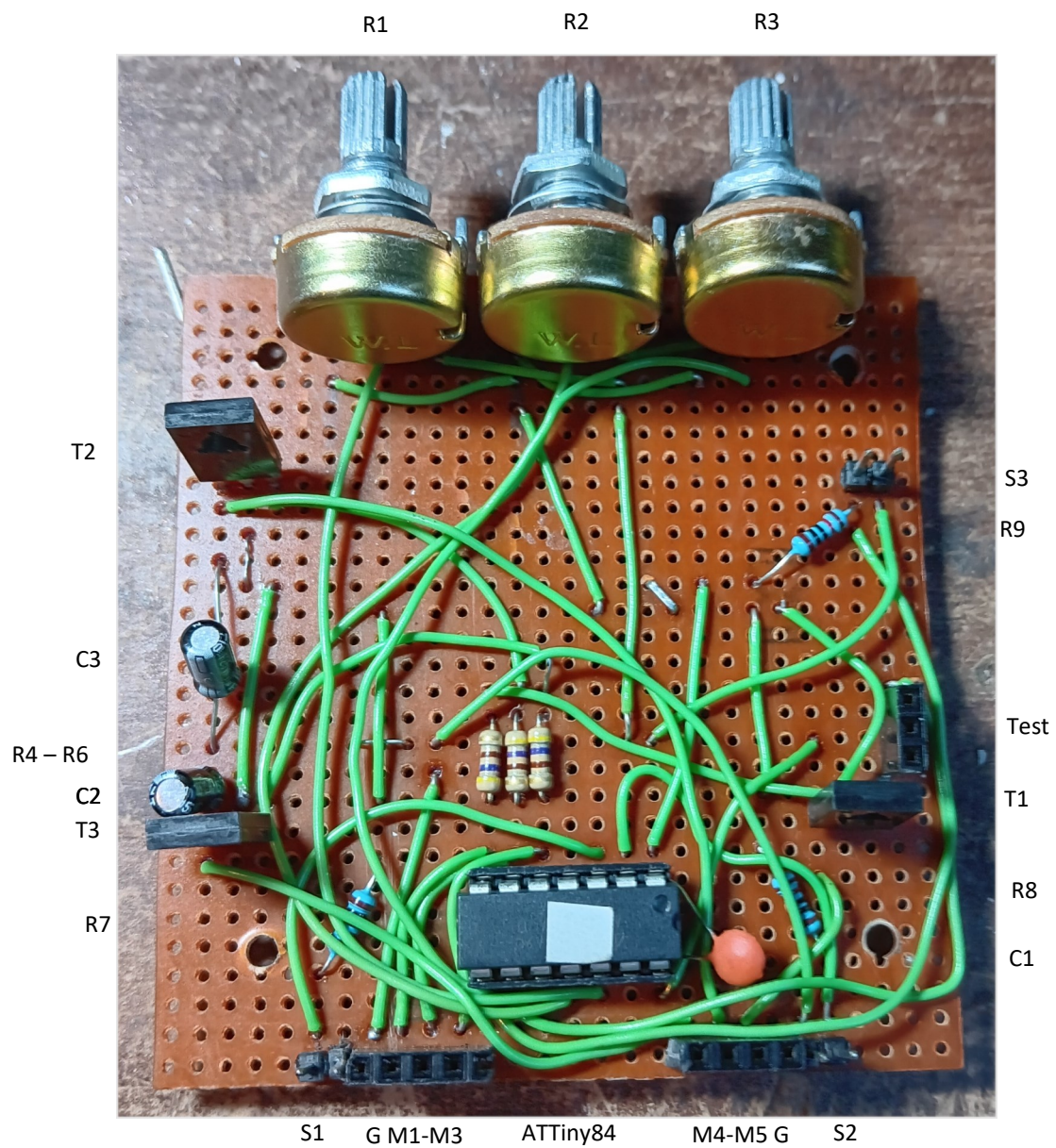
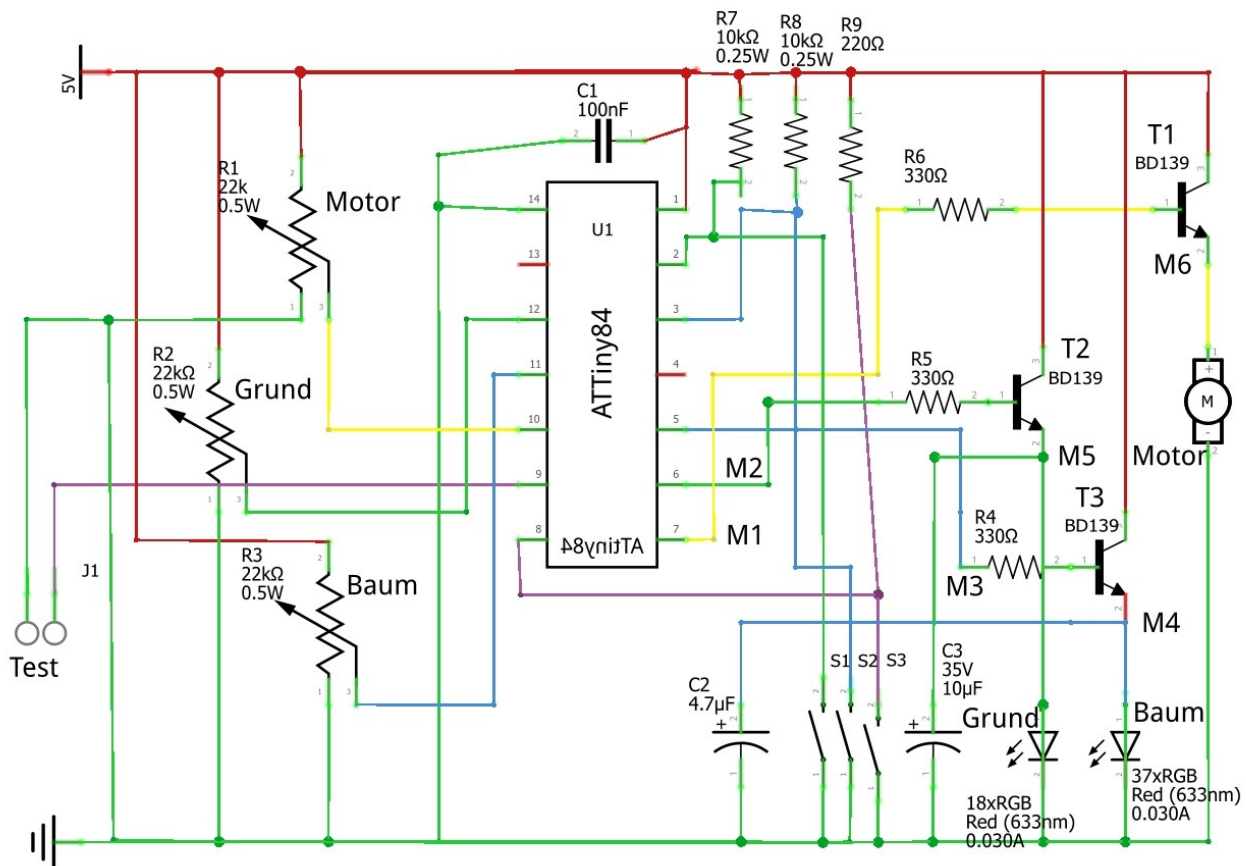


Bild3 Weihnacht_schaltplan.jpg



fritzing

Der Baum wird nach Anleitung aufgebaut. Bei der Grundplatte wurden die blauen LEDs durch Flacker-LEDs ersetzt. Möglich sind auch RGB Farbwechsel-LEDs. Weiter wird die Grundplatte nicht bestückt. Die Verbindung Emitter T3 zum Baum VCC der Grundplatte geschieht über Anschluß Motor+ der Grundplatte. Emitter T2 geht auf Kollektor T1 der Grundplatte. Motor + direkt auf Emitter T1.

Die Kondensatoren C2 und C3 sind empirisch ermittelt. An den Emittern T2 bzw. T3 muß eine Spannung von $V_{min}=1\text{ bis }1,2\text{V}$ bleiben, so das die LEDs weiterarbeiten und nicht in den LOW-Phasen der PWMs neu starten. Abhängig sind diese von der Art und Menge der LEDs = Stromstärke. Siehe Bild 3.

Bild3 Oszillograph an M5 mit duty 8%

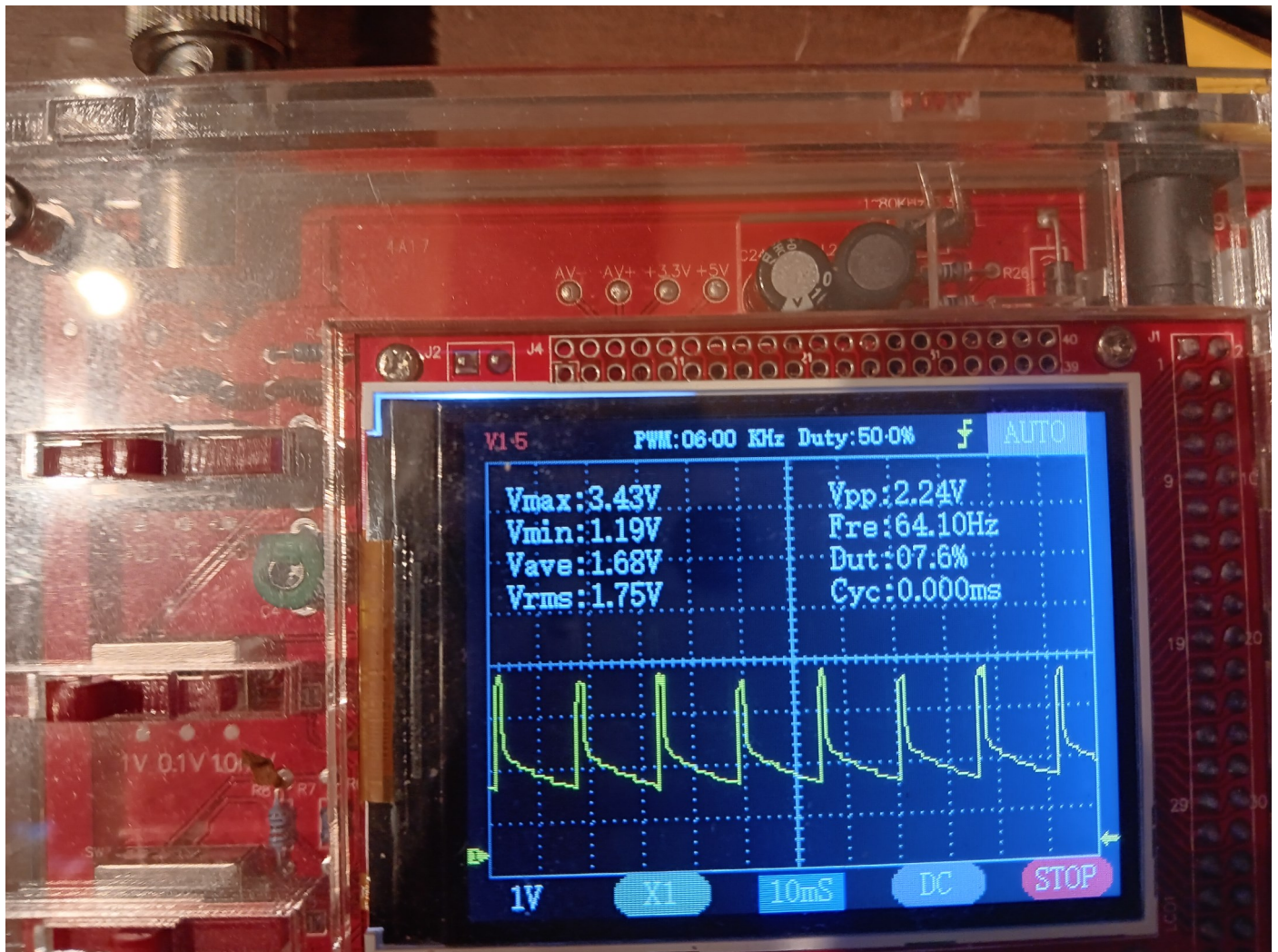
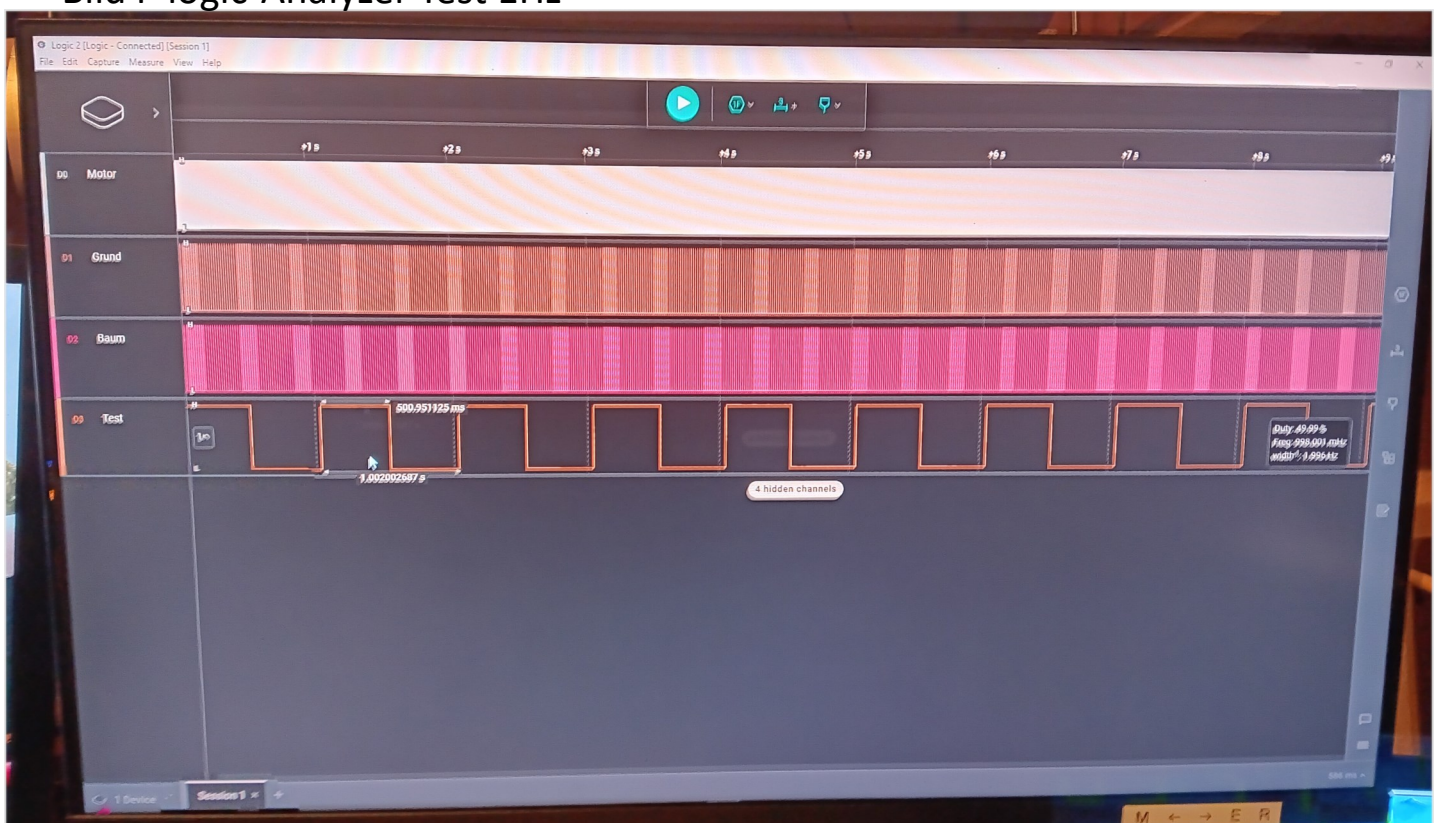


Bild4 logic-Analyzer Test 1Hz



Bemerkungen zum Programmcode

Das Programm wurde mit Arduino 1.8.19 geschrieben, da unter 2.2.1 kein Programmer gefunden wird.

Das Board wird mit:

http://drazzy.com/package_drazzy.com_index.json

In den Voreinstellungen eingebunden.

Board: attiny by David A. Mellis

Unter Werkzeuge dann folgende Einstellungen:

Board: ATTiny24/44/84

Prozessor: ATTiny84

Clock „Internal 1MHz“

Programmer: Arduino as ISP

Im Prinzip werden zwei Modis zur Verfügung gestellt. Modus1 S3 geöffnet ohne Zeitsteuerung, Modus2 S3 geschlossen mit Zeitsteuerung. Ein Neustart ist erforderlich. Die Jumper S1 und S2 stellen ein kontinuierliches Dunkel zu Hell und wieder zu Dunkel zur Verfügung. Die Geschwindigkeit wird mit den Potenziometer R2 und R3 eingestellt. Bei offenen Jumper bestimmen diese die Helligkeit der LEDs. R1 bestimmt die Motorgeschwindigkeit. Sie beträgt ca. 1.4 bis 4 U/min, darunter bleibt der Motor stehen. Über den verwendeten Motor kam ich und auch AZ-Delivery keine Aussage machen (China). Die entsprechenden Werte sind empirisch ermittelt. Das gilt auch für die Ladekondensatoren C2 und C3.

Am ATTiny84 können die Pins PB3 und PA0 nicht verwendet werden. PB3 ist mit RESET belegt und PA0 ist AREF bei Verwendung der ADCs – PA1 bis PA3—.

Einiges zur Zeitsteuerung: Der ATTiny84 arbeitet zum Einen nicht mit genau 1MHz und die PWMs an PA6 und PA7 haben einen Einfluss auf delay() und delayMicroseconds(). Meine Einstellungen erreichten im Langzeittest von 5 Tagen eine Genauigkeit von minus einer Minute. Je nach eingestellter PWM an Baum und Grund kürzer oder länger.

Bild4 zeigt das pseudo PWM an Test Pin von 1Hz.