1. Praktikum

# Was ist PWM?

PWM steht für Pulse Width Modulation, Pulsweitenmodulation. Sie beschreibt das Verhältnis zwischen der Einschaltzeit und Periodendauer eines Rechtecksignals bei gleichbleibender Grundfrequenz.

Das Signal x(t) ist für die Periode im Intervall [0,T] definiert:

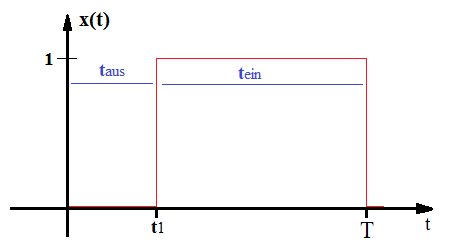


Abbildung 1: PWM (Quelle: http://www.mikrocontroller.net/articles/Datei:Pwmdoc.png)

Dies ist eine Modulationsart, bei welcher eine technische Größe, wie die elektrische Spannung, zwischen zwei festgelegten Werten hin und her wechselt. So wird beispielsweiße bei einem Motor, welcher sehr schnell in gleichbleibenden Abständen kurzzeitig mit Spannung versorgt wird, in einer relativ konstanten Geschwindigkeit laufen.

Die Dauer in welcher der Motor ein und wieder ausgeschaltet wird, wird als Periode bezeichnet. Die Anzahl der Perioden pro Sekunde ergibt dann die Frequenz in Hertz. So würde bei 1000 Perioden in der Sekunde eine Frequenz von 1000Hz ergeben

Die Periodendauer würde bei 1000Hz entsprechend 0.001s, also eine Millisekunde betragen.

Eine ebenfalls wichtige Größe ist der sogenannte Tastgrad, durch ihn lässt sich die die elektrische Spannung steuern. Er lässt sich aus Pulsdauer () und Periodendauer berechnen.

Üblicherweise wird der Tastgrad in Prozent angegeben. Durch den Tastgrad lässt sich der Servo (der Praktikumsaufgabe) steuern. So stellt sich der Servo bei

auf 0° (Tastgrad = 2,5%)

auf 90° (Tastgrad = 7,5%)

auf 180° (Tastgrad = 12,5%)

ein.

# Realisierung unter Linux

Damit die benötigten Files zum Ansprechen der GPIO-Pins verfügbar sind, sind diese mit den Befehlen:

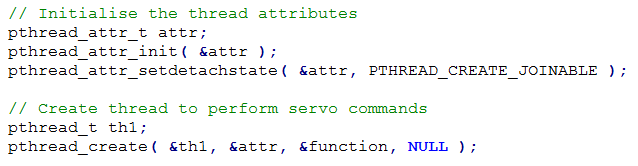
*echo 44 > /sys/class/gpio/export*

*echo out > /sys/class/gpio/gpio44/direction*

freizuschalten. (Es werden die Werte „44“ und „out“ in die Files geschrieben)

Für eine möglichst zeitkonstante Realisierung des PWM, ist es erforderlich das Programm in zwei Module zu unterteilen. Das „Haupt“ oder „Main“ Modul wird die Eingaben des Benutzers entgegennehmen. Wohingegen in das zweite Modul in einem separaten Thread kontinuierlich die Aktualisierungen an der gpio Datei vornimmt.

Durch die Aufteilung des Programms in zwei unabhängige Module, ist es dem Benutzer möglich den Modus des Programms zur Laufzeit zu verändern.



Je nach gewähltem Modus, wird entsprechend vorgegangen.

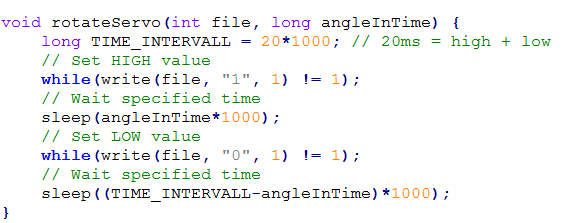
**Manueller Modus:** Hat der Benutzer den manuellen Modus ausgewählt, so hat er danach die Möglichkeit eine Eingabe zu tätigen, um den Servo in eine konstante Ausrichtung zu bewegen. Hierfür muss der Benutzer lediglich eine prozentuale Angabe eingeben. Diese Eingabe entspricht bei 0% 🡪 0° bis hin zu 100% 🡪 180°.

**Automatischer Modus:** Hat der Benutzer den automatischen Modus ausgewählt, werden keine weiteren Eingaben verlangt. Der Servo bewegt sich zwischen 0° und 180° linearer hin und her. Sprich, der Servo wird von 0° auf 180°, in der Schrittweite von 1% (= 1,8°), bewegt und bei dem Erreichen von 180° zurück auf 0°. Dies geschieht solange, bis eine erneute Benutzereingabe in einen anderen Modus erfolgt.

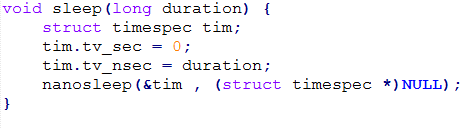
Um das PWM Signal an den Servo zu senden wird während der Dauer einer Periode (20ms) je nach Winkel der HIGH-Value (=1) für eine entsprechende Zeit in das gpio-value-file

*/sys/class/gpio/gpio44/value*

geschrieben und für die restliche Zeit wieder auf den LOW-Value (=0) gesetzt.



Zur erfolgreichen Umsetzung der Aufgabe konnten wir die vorgegebene WasteTime-Funktion nicht verwenden, diese war vermutlich unter Linux zu inkonsistent von der Ausführungszeit, was zu ungültigen PWM-Signalen geführt hat (Anscheinend werden die Threads während der Ausführung sehr häufig vom Scheduler unterbrochen). Dies hatte die Folge, dass die Verbindung zum Servo nach einiger Zeit verloren gegangen ist. Stattdessen wurde die nanosleep Funktion genutzt.



Vor dem „Entdecken“ dieses Fehlers, konnte auch eine auslastungsabhängige Indifferenz festgestellt werden. Dies hat dazu geführt, dass während eines Kopiervorgangs mit SCP von der virtuellen Maschine sogar zu einem sofortigen Abbruch der Verbindung (BeagleBone wurde nicht mehr als USB-Gerät erkannt) geführt hat.