1. Praktikum
2. 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 ein Signal x(t) ist für die Periode im Intervall [0,T] definiert:

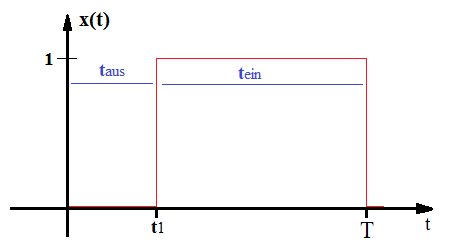


Abbildung 1: PWM (Quelle: http://www.mikrocontroller.net/articles/Datei:Pwmdoc.png)

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 = 5%)

auf 90° (Tastgrad = 7.5%)

auf 180° (Tastgrad = 10%)

ein.

1. Realisierung unter Linux

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

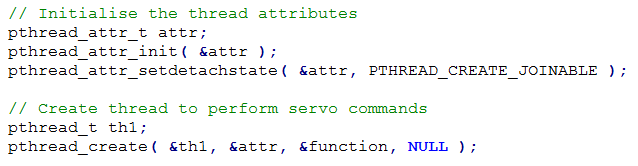
*echo 44 > /sys/class/gpio/export*

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

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

Das Programm kann sich abhängig vom Userinput, in den automatischen oder in den manuellen Modus befinden.

Um eine Benutzereingabe während des Betriebs zu ermöglichen, muss das Programm in mindestens zwei Threads ablaufen.



Je nach gewählten Modus, wird entsprechend vorgegangen.

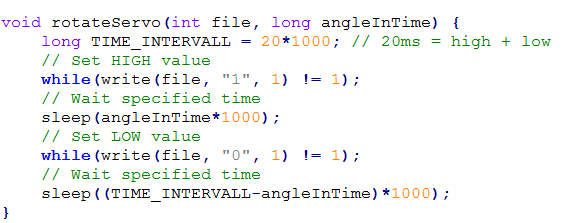
**Manueller Modus:** Je nach gemachter Angabe in Prozent lässt sich der Servo zwischen 0° (0%) und 180° (100%) einstellen.

**Automatischer Modus:** Der Servo bewegt sich zwischen 0° und 180° hin und her.

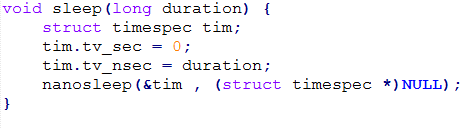
Um das PWM Signal an den Servo zu senden wird während der Dauer einer Periode (20ms) je nach Winkel der HIGH-value für eine entsprechende Zeit in das gpio-value-file

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

geschrieben und für die restliche Zeit wieder auf den LOW-value 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.