

Ansteuerung eines Servos

Programmaufbau

Das Programm liest aus den Argumenten die Benutzereingabe aus. Hierbei stehen dem Nutzer 2 Modi zur Verfügung. Mit `-1` startet er den Automatik-Modus, der den Servo regelmäßig von -90° bis $+90^\circ$ und wieder zurück laufen lässt. Mit einer Eingabe von 0 für 0% bis 100 für 100% startet der Nutzer den manuellen Modus, der den Servo zur gewünschten Stellung fahren lässt.

Intern benutzt das Programm ein PWM Signal zur Einstellung des Servos. Dieses Signal wird über 2 Aufrufe von `nanoSleep` erzeugt. Wobei der erste Aufruf das High Signal ausgibt und anschließend, abhängig von der benötigten Servo Stellung, in ein Low umstellt. Da das High Signal vom PWM und die PWM Periode so exakt wie möglich sein muss, wird eine Real Time Clock verwendet.

QNX Besonderheiten

Unter QNX muss zuerst die Auflösung der Clock erhöht werden, damit das PWM Signal sauber erzeugt werden kann. Siehe hierzu die Funktion `changeSystemTick()`.

Linux Besonderheiten

Unter Linux muss zuerst der GPIO exportiert und seine `direction` auf `out` gesetzt werden. Dies wird über die `system()` Funktion mit dem entsprechenden Linux Befehl erreicht.

Lastuntersuchung

Um das Zeitverhalten unter Last zu testen, haben wir den Servo erst mit einer niedrigeren Priorität gegenüber der Last getestet und anschließend die Priorität höher als die Last eingestellt. Hierbei konnten wir unter QNX ein anderes Verhalten feststellen, als unter Linux.

QNX

Um Last zu erzeugen haben wir ein Programm erstellt, dass 30.000 mal eine Ausgabe auf der Konsole erzeugt. Zudem haben wir die Priorität mit `pthread_setschedprio()` verändert. Beim Programmlauf mit niedrigerer Priorität hat der Servo angefangen stark zu zittern, ist aber trotzdem weitergelaufen. Anschließend haben wir die Priorität vom ServoController

erhöht und beim Programmlauf konnte das LastProgramm den Servo nicht mehr beeinflussen.

Linux

Unter Linux haben wir in der Konsole mit dem Befehl `ls -lR` im `\etc` Verzeichnis Last erzeugt. Die Priorität haben wir mit dem `nice` Befehl angepasst. Hier hat der ServoController bei niedriger Priorität komplett aufgehört zu reagieren und erst nach Fertigstellung des Last-Befehls ist er weitergelaufen. Mit einer höheren Priorität hat der Servo angefangen zu zittern und wurde entgegen dem Lauf unter QNX noch von der Last beeinflusst.