

Felix Rehaag

Rami Chaari

4/1/6

06.11.2019

Prof. Dr. Paweł Buczek

Inhalt

[*Einleitung:* 2](#_Toc24492785)

[*Benutzte Laborgeräte:* 2](#_Toc24492786)

[Aufbau 3](#_Toc24492787)

[*Aufleuchten der LEDs am Anschlag* 4](#_Toc24492788)

[Programm 5](#_Toc24492789)

[Darstellen eines Zeichens 6](#_Toc24492790)

[Programm 7](#_Toc24492791)

[Darstellung einer sinnvollen Zeichenfolge 9](#_Toc24492792)

[Programm 10](#_Toc24492793)

[*Fazit* 11](#_Toc24492794)

# Einleitung:

Im zweiten Laborversuch geht es darum, ein C-Programm zu entwickeln, welches die Steuerung von LEDs an einem LED-Pendel ermöglicht. Im ersten Teil der Aufgabe sollen die LEDs aufleuchten, wenn das Pendel den rechten oder linken Anschlag erreicht. Unter berücksichtigung der Schwingfrequenz des Pendels können beliebige Zeichen dargestellt werden. Im laufe des Versuchs soll ein Programm geschrieben werden, welches ermöglicht eine sinnvolle Zeichenfolge darzustellen.

# Benutzte Laborgeräte:

Für die Laboraufgaben wurden folgende Geräte benutzt:

* Mikrocontroller (Tiva TM4C1294)
* LED-Pendel

# Aufbau

Um die LEDs per Microcontroller zu steuern wird das Pendel folgendermaßen angeschlossen:

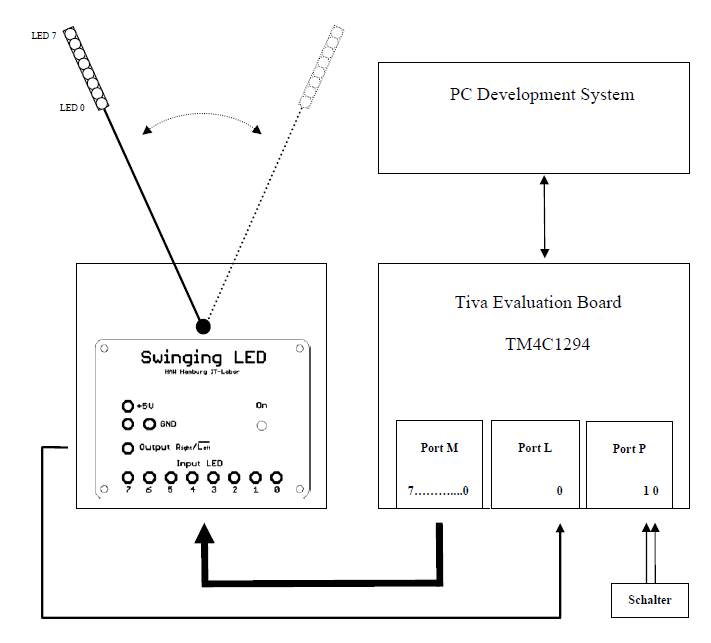


Abbildung 1: Anschlussweise des LED-Pendels

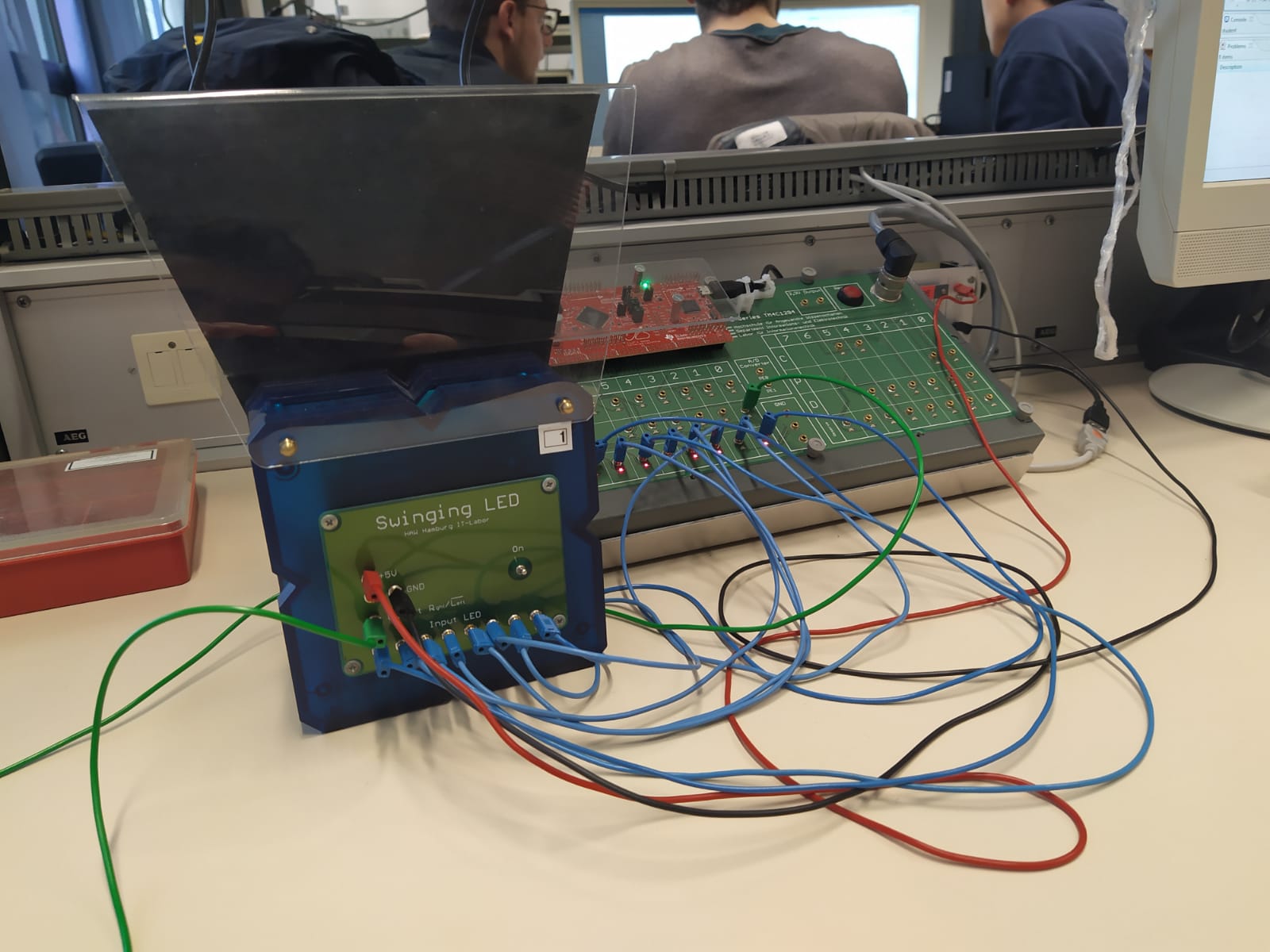


Abbildung 2: Laboraufbau

# Aufleuchten der LEDs am Anschlag

Im ersten Teil des Labors soll ein Programm geschrieben werden, welches die LEDs des Pendels jeweils am rechten und linken Anschlag aufleuchten lässt.

Die Umkehrpunkte des Pendels werden durch die steigende und fallende Flanke des R/L-Signals gekennzeichnet.

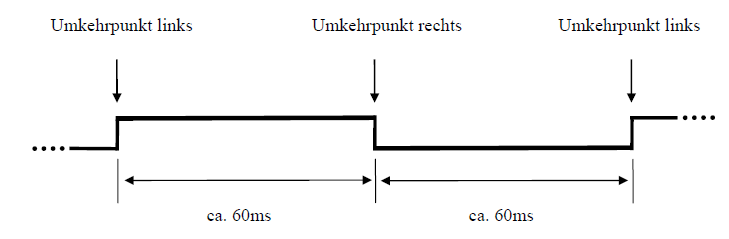
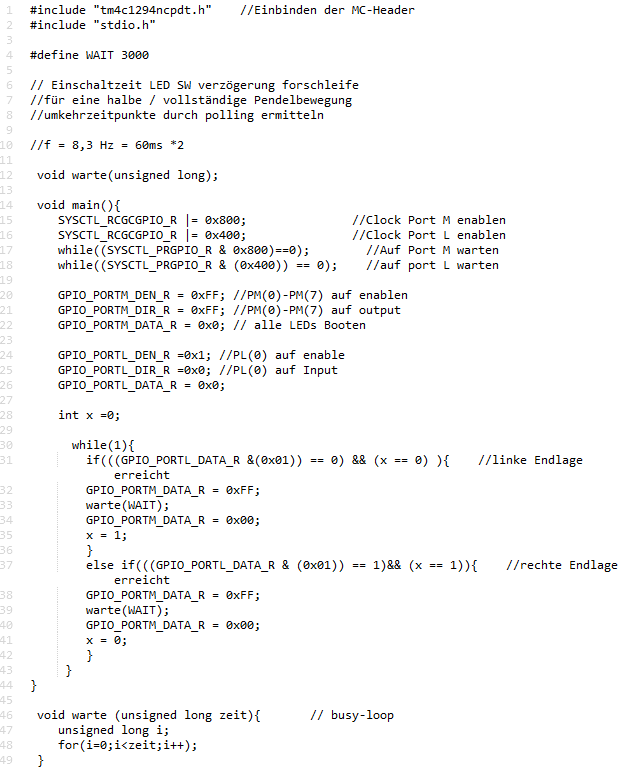


Abbildung 3: Signalverlauf des Pendels

Dieses Signal ermöglicht es, den Zeitpunkt zum Aufleuchten der LEDs festzulegen.

## Programm

Die LEDs werden über folgendes Programm gesteuert:



Realisiert ist die Steuerung hier über eine Dauerschleife, die anläuft nachdem die Ports konfiguriert wurden. In ihr werden, immer nach erreichen der Endlage, alle LEDs für eine definierte Zeit eingeschaltet und im Anschluss wieder ausgeschaltet. Die Ausschaltverzögerung wird durch eine busy-loop-Schleife (Funktion Z. 46-49) gesteuert.

# Darstellen eines Zeichens

Um mit dem Pendel ein statisches Zeichen zu visualisieren müssen verschiedene LEDs in bestimmter Reihenfolge aktiviert und wieder deaktiviert werden. Um herauszufinden welche Ports aktiviert werden müssen, kann anhand des Beispiels, der Visualisierung eines „X“ nachvollzogen werden:

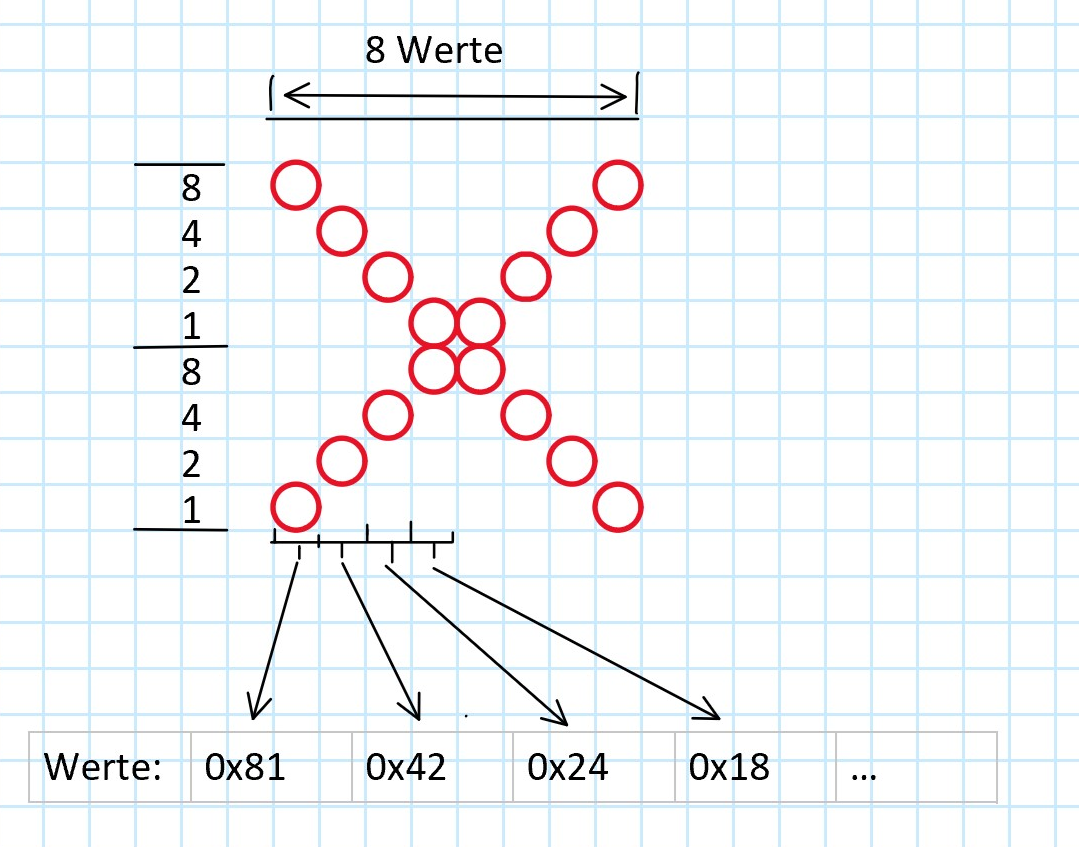


Abbildung 4: Ermittlung der richtigen Ports

Unter Verwendung dieser Werte, entstand folgende Visualisierung:

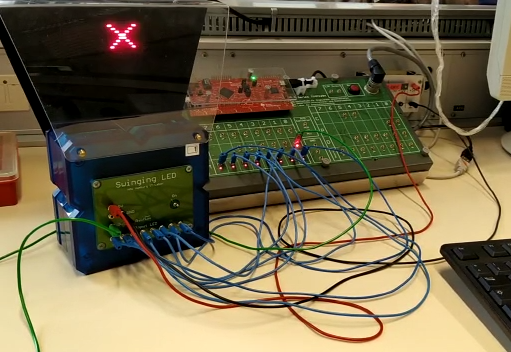
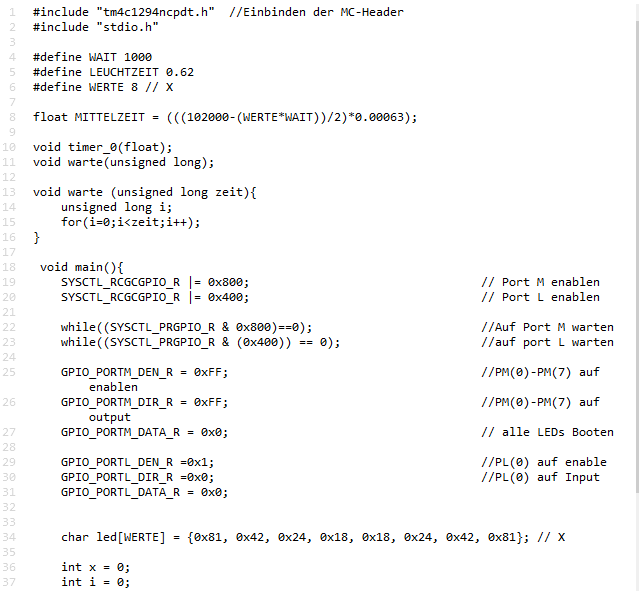


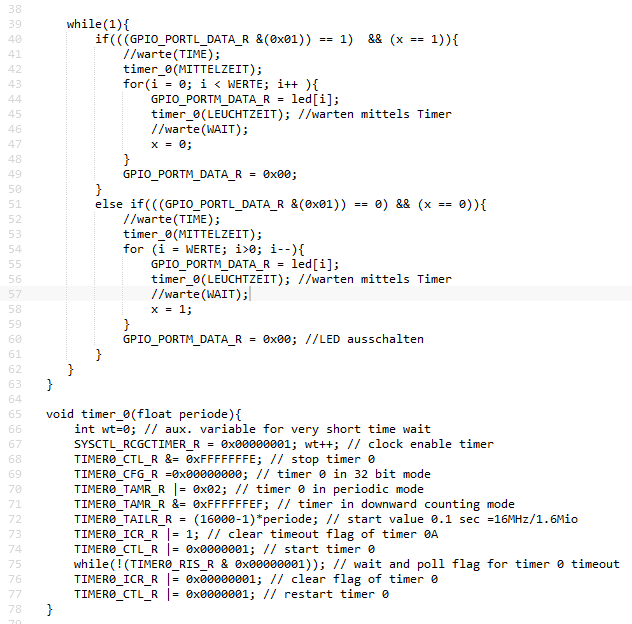
Abbildung 5: Ausgabe von "X" mit dem LED-Pendel

Wie in Abbildung 5 zu sehen, wird das „X“ mittig angezeigt. Um dies zu erreichen müssen die Ports mit einer Zeitverzögerung nach dem Anschlagen am rechten oder linken Rand geschaltet werden. Diese Verzögerung haben wir vorerst mit einer busy-wait Schleife realisiert. Da auf diese Weise eine genaue Einstellung der Verzögerung nur empirisch oder experimentell (wie z.B. im Labor 1 mit einem Oszilloskop) möglich ist, haben wir uns dazu entschieden die busy-wait Schleife durch eine Timer-Funktion zu ersetzen. Der Vorteil der Timer ist es, dass die genaue Zeit in Sekunden und nicht in Schleifendurchgängen angegeben werden kann.

Die Einstellungen für die Aktivierung der Ports haben wir in dem Array „led“ (Z. 34) gespeichert. Die Werte werden dann nacheinander durch eine for-Schleife ausgelesen. Dies ermöglich es, dass das Dargestellte Zeichen, leichter geändert werden kann.

## Programm





# Darstellung einer sinnvollen Zeichenfolge

Im letzten Teil des Labors soll eine sinnvolle Zeichenkette dargestellt werden. In unserem Fall handelt es sich um die Darstellung eines Schriftzugs:

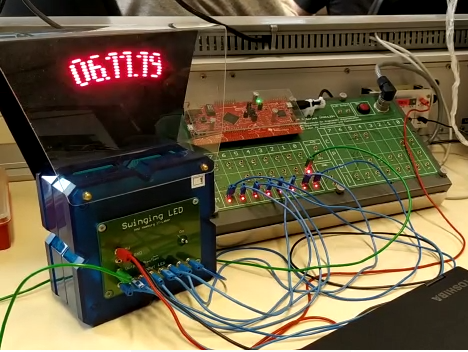
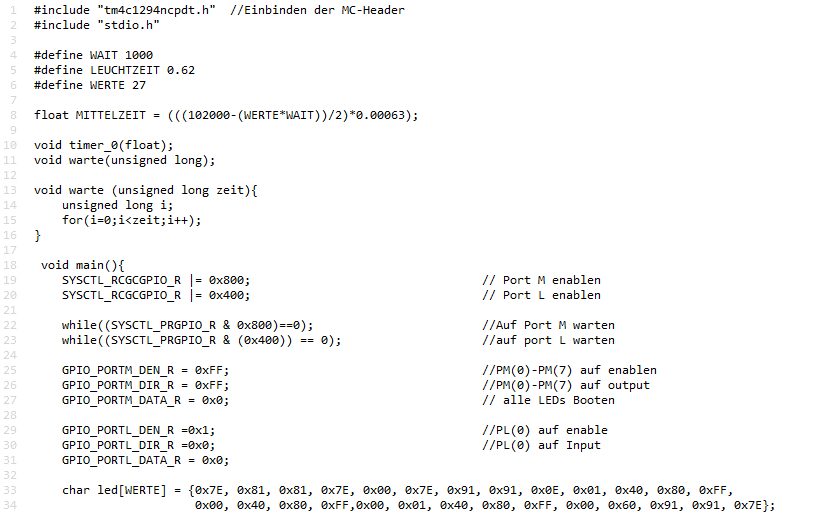
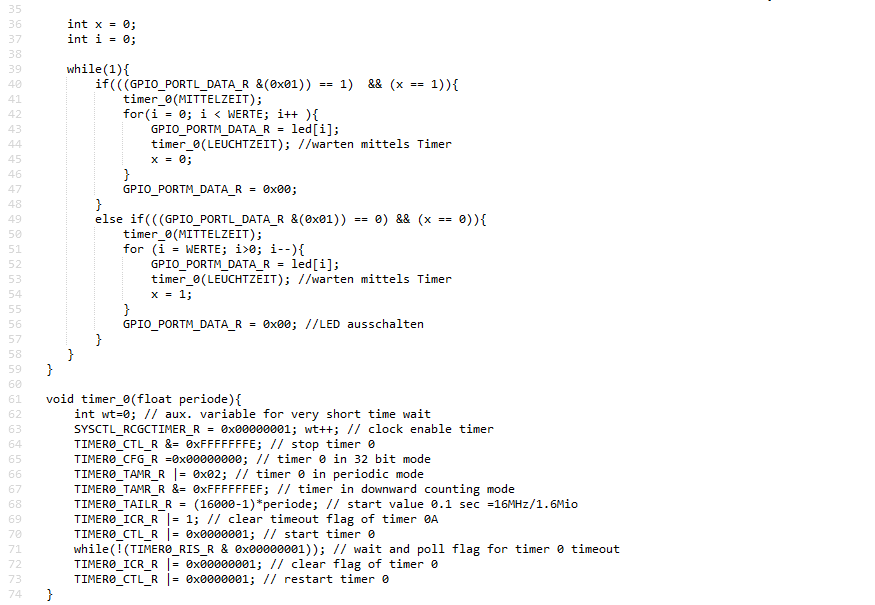


Abbildung 6: Darstellung einer Zeichenfolge

Diese Zeichenfolge ist nicht symmetrisch. Da das Pendel hin und her schwingt, würde die Darstellung beim zurückschwingen invertiert werden. Um dem vorzubeugen muss das Array nach dem Anschlag auf der rechten Seite von rechts nach links ausgelesen werden. Dies geschieht in der zweiten for-Schleife ab Zeile 51.

## Programm

# Fazit

Um sich Schreibaufwand bei sich wiederholenden Funktionen zu ersparen, kann auf for-Schleifen zurückgegriffen werden.

Hardware Timer sind leichter einzustellen als Softwareverzögerungen und eignen sich daher besser für Anwendungsfälle, bei denen eine genaue Zeitverzögerung gebraucht wird.