

---

# Allgemeine Hinweise zu den SPiC-Übungen

- Sie benötigen einen Login für die CIP-Pools der Informatik, um die Aufgaben bearbeiten zu können. Falls Sie noch keinen Login besitzen, erstellen Sie bitte einen unter `https://account.cip.cs.fau.de`.
- Alle an der Übung Teilnehmenden erhalten für SPiC jeweils ein spezielles Projektverzeichnis mit dem Namen `/proj/i4spic/<login>/`, wobei `<login>` für den eigenen Login-Namen steht. Die Projektverzeichnisse werden für alle Teilnehmer angelegt, die sich im Waffel-System angemeldet haben. Eine Anmeldung im Waffel-System ist daher zwingend zur Übungsteilnahme erforderlich! Die Projektordner werden auch in die SPiC-IDE eingebunden.
- Der Verzeichnisbaum für die Aufgaben ist folgendermaßen aufzubauen:  
`/proj/i4spic/<login>/aufgabe1`  
`/proj/i4spic/<login>/aufgabe2`  
...
- Die Aufgaben sind bis spätestens zum Abgabetermin durch die SPiC-IDE abzugeben. Alternativ kann auch per  
`/proj/i4spic/bin/submit aufgabeX`  
(mit  $X = 1 \dots n$ ) abgegeben werden. Dieses Programm kopiert die in der Aufgabenstellung verlangten Dateien aus dem entsprechendem Verzeichnis. Bis zum Abgabetermin kann ein Programm beliebig oft abgegeben werden – es gilt der letzte, vor dem Abgabetermin vorgenommene Aufruf des Abgabeprogramms.
- Um die letzte (und damit gültige) Abgabe zu überprüfen, kann die SPiC-IDE genutzt werden oder per  
`/proj/i4spic/bin/show-submission aufgabeX`  
die Abgabe betrachtet werden. Um nur die Unterschiede zwischen der letzten Abgabe und dem aktuellen Stand im Projektverzeichnis anzuschauen, kann die Option `-d` übergeben werden.  
`/proj/i4spic/bin/show-submission -d aufgabeX`
- Die Abgabezeitpunkte können Sie durch die SPiC-IDE abfragen oder durch Aufruf des Programms  
`/proj/i4spic/bin/get-deadline aufgabeX`  
Zusätzlich stehen die Termine auf den Aufgabenblättern.
- Eine Wertung bei Abgabe nach dem Abgabezeitpunkt kann **nur in begründeten Ausnahmefällen** nach Rücksprache mit Ihrem Übungsleiter erfolgen, der über den Einzelfall entscheidet. Eine frühere, fristgerechte Abgabe wird durch eine verspätete Abgabe *nicht* überschrieben. Im Zweifelsfall wird also erstere gewertet.
- Dieses Semester dient sowohl das SPiCsim als auch das SPiCboard als Referenz für die Korrektur. Bitte stellen Sie vor einer Abgabe sicher, dass Ihre Lösung auf einer der beiden Plattformen gemäß der Angabe funktioniert.
- Verwenden Sie für den Namen der C-Quelldatei, soweit in der Aufgabenstellung nicht anders angegeben, den Namen des Programms entsprechend dem Titel der jeweiligen Aufgabenstellung. Ist der Titel der Aufgabenstellung also z. B. *blink*, so legen Sie den Quellcode in einer Datei `blink.c` ab.
- Weitere Informationen finden Sie auf den Vorlesungsseiten:  
`https://sys.cs.fau.de/lehre/ss26/spic/`
- Die Dokumentation der `libspicboard` finden Sie ebenfalls auf der Vorlesungsseite:  
`https://sys.cs.fau.de/lehre/ss26/spic/uebung/spicboard/libapi`

---

# SPiC-Aufgabe #1: blink

(3 Punkte, keine Gruppen)

Programmieren Sie zum Kennenlernen der benötigten Programmierwerkzeuge zunächst ein C-Programm `blink` für das SPiCboard, welches abwechselnd die LEDs `YELLOW0` und `YELLOW1` (siehe SPiCboard-Skizze) an- bzw. ausschaltet. Es soll also immer eine der beiden LEDs aktiviert sein und die jeweils andere deaktiviert, wie es etwa bei einem Warnlicht für Baufahrzeuge der Fall ist. Die LEDs sollen in Zeitintervallen von etwa einer halben Sekunde blinken. Die Realisierung der Wartezeit soll durch eine aktive Warteschleife erfolgen; die zum Erreichen der Wartedauer notwendige Schleifenanzahl soll von Ihnen empirisch ermittelt werden.

Verwenden Sie zur Ansteuerung der LEDs die SPiCboard-Bibliothek und suchen Sie in der Onlinedokumentation der Bibliothek nach den benötigten Funktionen. Speichern Sie den Programmquellcode in einer Datei `blink.c` ab.

## Hinweise:

- Begründen Sie die Verwendung von allen `volatile` Variablen. Wenn für mehrere Variablen die selbe Begründung gilt, dürfen Sie diese gemeinsam begründen.
- Im Verzeichnis `/proj/i4spic/pub/aufgabe1/` befindet sich die Datei `blink.elf`. Mit dieser flashbaren Beispielimplementierung können Sie die gewünschte Verhaltensweise des Programms nachvollziehen.
- Achten Sie darauf, dass bei Ihrer abgegebenen Lösung der Compiler keine Warnungen oder Fehler beim Übersetzungsvorgang ausgibt.

## Abgabezeitpunkt

T01	27.04.2026	18:00:00
SLP	29.04.2026	18:00:00

## Eigenständiges Lösen

Alle Aufgaben müssen **selbstständig** bearbeitet werden. Die generative Verwendung von KI-Tools jeglicher Art ist untersagt.