

Technische Informatik II

Motivation

Die Lehrplaneinheit TI-II soll den Schülerinnen und Schülern am Technischen Gymnasium ein grundsätzliches Verständnis der Funktionsweise moderner Mikrocontroller und Mikroprozessoren als informationslogische Schaltungen und damit der digitalen Welt, vermitteln. Die Schülerinnen und Schüler sollen die Möglichkeiten und Grenzen von Mikroprozessoren kennenlernen. Sie lernen in ausgewählten Beispielen die Zerlegung von Aufgaben in elementare Lösungsschritte (Top-Down-Design). Die modulare Denkweise in Funktionseinheiten (Blackbox) wird gefördert. Grundlegende Prinzipien wie z.B. Polling und Unterprogrammtechnik, werden angewendet. Es ist nicht Ziel von TI-II Assemblerprofis auszubilden. Deshalb muss auch nicht der volle Assemblerbefehlssatz des Mikrocontrollers in allen Einzelheiten behandelt werden.

Inhaltliche Einordnung

TI-II stellt das zentrale Bindeglied zwischen Informationslogik auf der einen Seite sowie Hochsprache, Prozessorarchitektur, Netzwerke oder auch IoT auf der anderen Seite, dar. Bereits in der Hardware werden Sprachelemente der Hochsprache C, wie Structs oder auch if-then-Strukturen, abgebildet. Diese Vernetzung ist von zentraler Bedeutung für die inhaltliche Ausgestaltung des Themas. Die Zuordnung zur Eingangsklasse erleichtert es den Schulen die TI-II standortspezifisch auszugestalten.

Anforderungen an die Lernplattform

- Verbreitete, zukunftssichere, leistungsfähige 32-Bit-Prozessorarchitektur
- Professionelle und freie Software-Tools
- Günstige, am Markt verfügbare Hardware
- Reichhaltiges Zubehör für Projekte
- Programmierbaren in Hochsprache C und in Assembler
- EA-Ports
- Timer
- Interrupts
- Breites Spektrum an Schnittstellen
- Onchip Debugger mit folgenden Möglichkeiten:

- Breakpoints
- Singlestepping
- Beobachtung der Prozessorregister, der Hardware und der Variablen

Ausgewählte Mikrocontroller-Plattform

Fast jedes Smartphone auf dem Planet verwendet zur Zeit mit dem ARM, eine günstige und energie-effizienten CPU. Verglichen mit x86 ist ARM wesentlich einfacher aufgebaut und kann leicht lizenziert werden. Die Firma ARM fertigt selbst keine Chips sondern verkauft Lizenzen an Halbleiterhersteller. So konnte sich ARM quasi als Industriestandard etablieren. Wir finden ARM CPUs nicht nur in Smartphones, sondern auch in den Mikrocontrollern vieler Hersteller. Die Anwendungen reichen dabei von der Steuerung einfacher Geräte ("embedded") über IoT bis zu massiv parallelen Supercomputerchips.

Diese Handreichung verwendet einen [STM32L152RET](#) Mikrocontroller der Fa. ST auf einer [STM32 Nucleo Platine](#). Diese Platine ist für unter 15€ fertig bestückt verfügbar. Der Mikrocontroller arbeitet mit einer [ARM Cortex M3 CPU](#). Als Entwicklungsumgebung wird die [SW4STM32](#) System Workbench verwendet. Die Nucleo-Platine kann unter anderem auch mit der Online-Entwicklungsumgebung [MBED](#) programmiert werden. Die Nucleo-Platine wird über USB mit Strom versorgt und programmiert. Sie verfügt über Buchsenleisten zum Aufstecken von Arduino-Shields und über Stiftleisten an denen die 3 16-Bit IO-Ports des Mikrocontrollers zugänglich sind. Die [SW4STM32](#) Entwicklungsumgebung ermöglicht außer der Programmierung in C und Assembler auch ein komfortables Debugging der Programme mittels In-Circuit-Debugger. Für die Handreichung wurde die Nucleo-Platine auf das [Base-Shield](#) der Gewerblichen Schule Öhringen aufgesteckt. Dieses ist nicht zwingend erforderlich, erleichtert die Arbeit allerdings ungemein, denn so erübrigt sich der externe Anschluss von Grund-Bedienungselementen wie Taster, Schalter, LEDs, Display, und Leistungstreiber. Außerdem sind Steckplätze für WiFi, Bluetooth, LAN, Ultraschall und Infrarot vorgesehen.

Struktur der Lerninhalte