

# Netzwerke und Embedded Software

Unterrichtskonzept für den dritten Jahrgang

## Grundlagen

Basis des Unterrichts ist der aktuelle Lehrplan der Höheren Lehranstalt für Wirtschaftsingenieure – Betriebsinformatik. Dort ist der Lehrstoff in zwei Module unterteilt. Thema des sogenannten fünften Moduls ist die Programmierung eingebetteter Systeme, im sechsten Modul wird auf Echtzeitbetriebssysteme und Multitasking eingegangen. Beurteilungsgrundlage des fünften Moduls ist das Programmierprojekt. Die Beurteilung des sechsten Moduls erfolgt durch eine praktische Leistungsfeststellung.

## Modulüberblick

Einen Überblick über die einzelnen Module finden Sie im Anschluss. Jedes Modul ist in Kompetenzbereiche unterteilt, diese wiederum in einzelne Kapitel. Im Folgenden werden auch diese Einzelkapitel Module genannt. Sie besitzen eine dreiteilige Bezeichnung. Modul 5.1.1 behandelt beispielsweise die Programmierung von Peripheriekomponenten und stellt hierfür Programmierbeispiele zur Verfügung. Wahlmodule sind in der folgenden Übersicht apostrophiert.

### Module 5: Programming Embedded Systems

- Introduction
- 1. Programming
  - 1.1 Peripherals
  - 1.2 Interrupts
  - 1.3' Debugging
- 2. Communication Protocols
  - 2.1 Protocols
  - 2.2 Analysis and Selection
  - 2.2' Implementation
  - 2.3' Examples
- 3. Finite-State Machine
  - 3.1' Implementation
- 4. Coding Theory
  - 4.1' Data Representation
  - 4.2' Number Representation

### Module 6: Real-Time Operating Systems

- 1. Real-Time Operating Systems
  - 1.1 Operating Systems
  - 2.1 Basic Principles of Multitasking
  - 2.2 Inter-Task Communication
  - 3.1 Processes and Threads

## Organisation

Für das Fach stehen drei geteilte Unterrichtsstunden zu Verfügung. Der Unterricht ist offen und eigenverantwortlich organisiert. Das bedeutet, dass sich die Schüler Stoff und Zeit frei einteilen können, sie aber auch selbst sicherstellen müssen, dass die nötigen Kompetenzen erworben werden. Im Zentrum des fünften Moduls steht das Programmierprojekt. Es stehen neun verschiedene Projekte zur Auswahl, die im Zweierteam bearbeitet werden. Eine Übersicht der einzelnen Projekte ist unter <http://www.neff.co.at/courses/nwes/#programming-projects> zu finden. Im Laufe des sechsten Moduls werden die grundlegenden Konzepte des Multitaskings erarbeitet und einfache Multitasking-Anwendungen erstellt.

Der Unterricht ist konstruktivistisch. Die Schüler müssen sich den Stoff in der Regel selbst erarbeiten, erhalten aber alle hierfür nötigen Informationen und Materialien. Dem Lehrer kommt dabei die Rolle des Coaches zu, der die Schüler bei der Stoffbearbeitung unterstützt. Eine ausgeprägte Feedbackkultur ist dabei unabdingbar. Die Schüler sind daher verpflichtet, in jedem Unterrichtsblock mindestens ein kurzes mündliches Feedback zu geben. Dabei soll der momentane Stand und die zukünftig geplanten Tätigkeiten erläutert oder eine konkrete Problemstellung besprochen werden.

## Beurteilung

Für eine positive Beurteilung in diesem Fach ist es notwendig, beide Module positiv abzuschließen. Beurteilungsgrundlage des fünften Moduls ist das Programmierprojekt. Die Beurteilung des sechsten Moduls erfolgt durch eine praktische Leistungsfeststellung.

Im Rahmen des Programmierprojekts ist eine Aufgabenstellung praktische umzusetzen und zu dokumentieren. Nähere Informationen hierzu finden Sie im folgenden Abschnitt. Die Beurteilung erfolgt aufgrund eines begründeten Notenanspruchs. Erläutern Sie darin, warum Sie die beantragte Note als begründet erachten. Geben Sie aber auch Gründe an, die dagegensprechen könnten. Die Fähigkeit seine Leistung kritisch beurteilen zu können ist Lehrziel dieses Faches. Beantragen Sie daher keine Phantasienoten, da sich dies nachteilig auf die Beurteilung auswirkt. Achten Sie darauf, Ihren Antrag angemessen zu begründen. Ein Dreizeiler wird der Komplexität eines Projekts im Umfang eines Semesters sicher nicht gerecht werden.

Die Bearbeitung der Programmieraufgaben im sechsten Modul erfolgt sehr offen auf dem Laptop. Eine verlässliche Feststellung der Mitarbeit im Unterricht ist dabei meist nicht möglich, da sich die Urheberschaft der Programmierleistungen nicht klären lässt. Das sechste Modul wird daher durch eine praktische Leistungsfeststellung abgeschlossen. Multitasking ist ja Teil der praktischen Informatik und das Erstellen von Multitasking-Anwendungen ist im Lehrplan explizit vorgesehen. Die Aufgabenstellung wird den Schülern aus einem Pool zugeteilt. Verpflichtende Nachweise hinsichtlich der zu erwerbenden Kompetenzen sind, im Gegensatz zum vorangegangenen Schuljahr, nicht vorgesehen. Es besteht allerdings die Möglichkeit solche Nachweise auf freiwilliger Basis abzulegen, um die Notenbasis zu erweitern. Die praktische Leistungsfeststellung wird gegen Ende des Schuljahres erfolgen. Rechtzeitig vor diesem Termin wird eine Übungsaufgabe zur Verfügung gestellt, um sich leistungsmäßig einschätzen zu können und um etwaige Defizite beseitigen zu können.

## Das Programmierprojekt

### Allgemeine Hinweise

Für eingebettete Systeme erstellt man im Allgemeinen Speziallösungen, die an den Zweck und die Umgebung optimal angepasst sind. Suchen Sie daher nicht im Internet nach fertigen Lösungen für Ihr Projekt. Sie werden sie entweder nicht finden, sie eignen sich nicht für die konkrete Anwendung oder sind für ein anderes System gemacht. Konzentrieren Sie sich vielmehr darauf, Ihre Speziallösung selbst zu entwickeln.

Überlegen Sie sich dazu ganz genau, wie Ihre Aufgabe gelöst werden kann. Die Lösungsschritte irgendwie zu kennen ist ein guter Anfang, aber Ihr Mikrokontroller muss Schritt für Schritt exakt wissen, was zu tun ist. Ein Ablaufplan oder ein Flussdiagramm kann dabei sehr hilfreich sein. Auch die Lösungsschritte einmal auf einem Blatt Papier durchzuspielen, kann helfen.

Überbewerten Sie die Codierung, also die Erstellung des Programms, nicht. Die Programme sind meist sehr einfach. Am aufwändigsten ist herauszufinden, was zu programmieren ist. Vermutlich werden Sie viel mehr vorausplanen müssen, als Sie es bisher gewohnt waren. Dies liegt daran, dass nun neben der Software auch noch die Hardware ins Spiel kommt. Schnell werden Sie feststellen, dass C, auch wenn Sie es bisher noch nie verwendet haben, das kleinere Problem ist.

Das Programmierprojekt baut auf dem vorherigen Modul auf. Machen Sie sich mit dem Stoff vertraut, falls er Ihnen nicht mehr geläufig ist. Ohne detaillierte Kenntnisse der Bitmanipulation werden Sie keine Zeile Code zustande bringen, denn sie ist die Basis der Programmierung eines Mikrokontrollers. Und ohne Kenntnis der Funktionsweise der eingesetzten Hardwarekomponenten kann die Um-

setzung des Projekts nicht gelingen. Fast jede Komponente wurde übrigens zumindest im Prinzip im Unterricht schon einmal besprochen.

Kommen Sie der Feedbackpflicht nach. Sie müssen die Aufgaben im Team aufteilen. Daher trifft diese Pflicht jeden einzelnen. Feedback kann entweder eine Frage zur konkreten Umsetzung des Projekts sein oder, falls Sie keine Fragen haben, eine Rückmeldung zum momentanen Projektstand. Bedenken Sie, dass das Feedback in die Beurteilung miteinfließt.

Haben Sie Vertrauen in sich selbst. Es ist ganz normal, wenn Sie sich zunächst überfordert fühlen. Versuchen Sie das Problem Schritt für Schritt anzugehen. Bleiben Sie hartnäckig, aber verlieren Sie auch nicht zu viel Zeit, falls Sie einmal hängen. Lassen Sie sich helfen. Warten Sie auf keinen Fall auf bessere Zeiten, denn sie werden schlechter werden! Zeitdruck wird als weiteres Problem hinzukommen.

### Projektbezogene Hinweise

Machen Sie sich mit der Aufgabenstellung vertraut. Starten Sie am besten mit dem Blockdiagramm, das zu Anfang eines jeden Projekts zu finden ist. Erklären Sie es sich einander und einigen Sie sich auf eine Version. Klären Sie sodann ab, ob die Aufgabenstellung vollständig erfasst wurde.

Erstellen Sie eine Liste der eingesetzten Komponenten. Aufgabe und Funktionsweise jeder Komponente müssen bekannt sein. Meist werden Sie nicht umhinkommen, Datenblätter zu lesen. Dies wird Ihnen anfänglich vermutlich schwerfallen. Datenblätter sind eher schwere Kost. Auf keinen Fall sollte man sie wie ein Buch lesen, sondern man überfliegt sie. Die Unterüberschrift verrät oft schon, um welches Bauelement es sich handelt. Was es macht, weiß man entweder aus dem Unterricht oder man schlägt es im Internet nach. Hilfreich sind meist auch das Blockdiagramm (functional diagram), die Pinbeschreibung (pin description), die Pinbelegung (pin configuration) und vor allem die Schalttafel (function table). Die wichtigsten Schaltsymbole sollte man natürlich kennen, aber das ist sicher der Fall. Sie haben sie ja bereits gelernt.

Überlegen Sie sich danach, wie die einzelnen Komponenten zusammenspielen. Dies erfährt man aus dem Schaltplan (schematic). Grundlage Ihrer Überlegungen sollten sowieso immer die Datenblätter und der Schaltplan sein. Nicht umsonst hat sich im Internet das Akronym *RTFM* eingebürgert, wobei man in unserem Fall *Manual* durch *Datasheet* oder *Schematic* ersetzen muss.

Gliedern Sie die Aufgabenstellung in Teilaufgaben. Überlegen Sie sich, welche Teilaufgaben unbedingt nötig sind, um dem Ziel einen Schritt näher zu kommen. Beginnen Sie mit diesen. Wenn Sie beispielsweise eine Uhr zu erstellen haben, wäre es gut, zuerst mit der Ausgabe zu beginnen. Wenn Sie nämlich auf dem Display etwas ausgeben können, haben Sie etwas vorzuweisen, um den Projektfortschritt zu dokumentieren. Wenn Sie hingegen mit der Zeiterfassung beginnen, haben Sie nichts, womit Sie sie kontrollieren könnten. Teilen Sie die Teilaufgaben sinnvoll im Team auf. Vor allem muss sichergestellt werden, dass das Team weiterarbeiten kann, wenn einmal ein Teammitglied fehlt.

Gehen Sie in kleinen Schritten vor und testen Sie regelmäßig. Dies betrifft sowohl die Hardware als auch die Software. Beginnen Sie zunächst mit ersterer. Ohne funktionierende Hardware kann die Software nichts erreichen. Bauen Sie nicht zu viel auf. Schnell haben sich Fehler eingeschlichen, und diese in einem großen System zu finden, ist schwierig und aufwändig. Meist muss dazu ein Teil der Schaltung abgebaut oder ein Teil des Programms wieder entfernt werden. Dies ist nicht effizient. Gehen Sie daher in kleinen Schritten vor und testen Sie lieber zu oft als zu selten.

Jedes Projekt enthält ein modulares Programmgerüst, an dem Sie sich orientieren können. Es besteht aus Header- und C-Dateien. Das Konzept der Modularisierung sollte Ihnen aus dem SWP-Unterricht bekannt sein. Die Header-Dateien sind vorgegeben und definieren die Schnittstellen (interfaces). Die

C-Dateien, die die Funktionalität implementieren, müssen selbst erstellt werden. Die Modularisierung ist vor allem beim Arbeiten im Team von Vorteil. Da alle Schnittstellen bekannt sind, können Funktionen bereits verwendet werden, bevor sie implementiert wurden. Dies fördert das parallele Arbeiten am Projekt, welches für die fristgerechte Abgabe unabdingbar ist.

### Abschluss des Projekts

Zum Abschluss des Projekts ist der Stand zu dokumentieren. Die Dokumentation umfasst folgende Bestandteile:

1. Das Softwareprojekt (Atmel-Studio-Solution) als ZIP-Datei
2. Die schriftliche Dokumentation als DOCX- oder PDF-Datei
3. Das Dokumentationsvideo als Link
4. Die Feedbackblätter als DOCX- oder PDF-Datei

Die Abgabe erfolgt über einen Moodlekurs. Bitte kontrollieren Sie rechtzeitig, ob Sie Zugang zum Kurs haben und Dateien abgeben können. Dort finden Sie auch den Abgabetermin. Dieser ist eine Deadline. Tragen Sie ihn in Ihren Kalender ein, damit Sie ihn nicht vergessen. Verspätete Abgaben können aufgrund des nahen Notenschlusses nicht berücksichtigt werden. Die Abgabe der einzelnen Dokumente hat durch **jedes** Teammitglied zu erfolgen. Die ersten drei Komponenten dürfen zwar gemeinsam erstellt werden, müssen aber von jedem Teammitglied abgegeben werden. Die Feedbackblätter sind von jedem Teammitglied individuell zu erstellen. Als Unikate sollten sie keine zu große Ähnlichkeit zu anderen Feedbackblättern aufweisen.

Halten Sie beim Quellcode folgende Vorgaben ein:

- Das Projekt muss sich ohne Fehler und ohne Warnungen übersetzen lassen.
- Jedes Teammitglied muss mindestens einen Programmteil erstellt haben.
- Die Autorenschaft ist in jedem Programmteil anzugeben<sup>1</sup>.
- Der Quelltext ist korrekt einzurücken.

Halten Sie bei der schriftlichen Dokumentation folgende Vorgaben ein:

- Beschreiben Sie Ihr Projekt kurz<sup>2</sup>.
- Erläutern Sie, in welche Teilaufgaben Sie das Projekt unterteilt haben.
- Geben Sie bei jeder Teilaufgabe an, von welchem Teammitglied sie bearbeitet wurde.
- Begründen Sie gegebenenfalls detailliert, warum einzelne Teile des Projekts nicht umgesetzt werden konnten.
- Fügen Sie, falls Sie das Projekt als Dreiergruppe<sup>3</sup> bearbeitet haben, den Signalverlauf einer typischen Kommunikationssequenz als Screenshot bei. Eine Anleitung hierzu finden Sie unter [Modul 5.3.4](#).

Halten Sie beim Dokumentationsvideo folgende Vorgaben ein:

- Es ist nicht das Video, sondern der Link auf ein YouTube-Video abzugeben<sup>4</sup>.

---

<sup>1</sup> Die Autorenschaft wird im Rahmen eines mündlichen Codereviews überprüft.

<sup>2</sup> Je ausführlicher Sie das Projekt im Video dokumentieren, desto geringer kann die schriftliche Dokumentation ausfallen.

<sup>3</sup> Als zusätzlichen Bonus dürfen diesen Punkt natürlich auch kleinere Gruppen beifügen.

<sup>4</sup> Selbstverständlich kann der Link auch auf ein anderes Videoportal verweisen. Bitte achten Sie darauf, dass auf Ihr Video auch zugegriffen werden kann und es nicht versehentlich auf *privat* gestellt wurde. Das abgegebene Video kann als Best-Practice-Beispiel veröffentlicht werden. Bitte verwenden Sie daher keine copyrightrelevanten Materialien.

- Es muss die Funktionalität des Werks demonstrieren. Die gezeigte Funktionalität muss der des abgegebenen Programmierprojekts entsprechen.
- Auch wenn es wenig zu demonstrieren gibt, muss ein Video abgegeben werden<sup>5</sup>.
- Das Video muss einen Vor- und einen Nachspann besitzen.
- In Vor- oder Nachspann müssen die Teammitglieder, die Klasse, das Fach und das Datum erwähnt werden.
- Im Vorspann muss der Titel des Projekts genannt und das Projektziel kurz erläutert werden.
- Das Ziel des Projekts muss klar dokumentiert werden. Hierzu sind erklärende Erläuterungen einzublenden. Anregungen kann man sich aus den Best-Practice-Beispielen der letzten Jahre holen.
- Das Video ist mit copyrightfreier Musik zu hinterlegen. Deren Quelle ist im Nachspann anzugeben.

Für die Feedbackblätter gelten folgende Vorgaben:

- Es sind drei Feedbackblätter abzugeben<sup>6</sup>.
- Diese sind von jedem Teammitglied individuell zu erstellen.
- Im Projektfeedback
  - reflektieren Sie, was Sie im Rahmen des Projekts gelernt haben.
  - beschreiben Sie kurz, was Ihnen am Projekt gefallen hat.
  - erläutern Sie, was Sie gerne ändert hätten.
- Im Selbstfeedback
  - beantragen Sie eine Note für sich selbst.
  - erläutern Sie, warum Sie diese Note verdient haben.
  - geben Sie Gründe an, die dagegensprechen könnten.
- Im Fremdfedback
  - beantragen Sie eine Note für Ihren Teamkollegen.
  - erläutern Sie, warum er bzw. sie diese Note verdient hat.
  - geben Sie Gründe an, die dagegensprechen.

Bitte beachten Sie, dass ein solches Fremdfedback nur dann sinnvoll ist, wenn es ohne vorherige Absprache verfasst wird.

Bitte achten Sie bei den abgegebenen Dokumenten auf Vollständigkeit und entsprechenden Umfang. Neben der Qualität der Dokumentation gehen auch Grad der Projektumsetzung und Angemessenheit des Selbst- und Fremdfedbacks in die Benotung mit ein.

---

<sup>5</sup> Dafür muss allerdings das Projekt schriftlich umso ausführlicher dokumentiert werden.

<sup>6</sup> Falls Sie das Projekt alleine bearbeitet haben, verringert sich die Anzahl der Blätter auf zwei. Falls Sie in einer Dreiergruppe gearbeitet haben, sind vier Blätter abzugeben.