

Von Eingebetteten Systemen zu Physical Computing: Grundlagen für Informatikunterricht in der digitalen Welt

Mareen Przybylla¹, Ralf Romeike²

Abstract: Der digitale Wandel basiert neben der zunehmenden Durchdringung des täglichen Lebens mit digitalen Systemen auch auf der parallel stattfindenden Entwicklung von untereinander vernetzten allgegenwärtigen, sogenannten cyber-physischen Systemen. Diese sind in der Lage, mit Sensorik ihre Umwelt zu erfassen und mittels Aktorik in die menschliche Erfahrungswelt zurückzuwirken. Für die Schulinformatik wird das Potenzial dieser Innovationen häufig noch unzureichend umgesetzt, was nicht zuletzt in Ermangelung eines geeigneten fachlichen Rahmens und daraus resultierend fehlender Lehr-Lern-Materialien begründet ist. Dieser Beitrag untersucht daher die fachlichen Grundlagen des Themenkomplexes und identifiziert für die Schulinformatik relevante Inhalte und stellt sie strukturiert dar. Anhand einer interaktiven Modellstadt wird exemplarisch gezeigt, wie die Inhalte im Informatikunterricht verankert werden können.

Keywords: Eingebettete Systeme; Physical Computing; Fachstruktur; Unterrichtsbeispiel

1 Einleitung

Immer weniger technische Entwicklungen der heutigen Zeit nutzen klassische Computer mit Bildschirm, Tastatur und Maus. Wir sind vorwiegend von mobilen Geräten und reaktiven Systemen umgeben, die anhand von Sensoren und Aktoren kontinuierlich mit ihrer Umwelt kommunizieren. Mikrocontroller dominieren in Gestalt von eingebetteten Systemen unseren Alltag. Im Schulunterricht werden hingegen meist klassische Desktop-PCs als Beispiel herangezogen, die nur einen Bruchteil aller Informatiksysteme ausmachen. Der Umgang mit eingebetteten Systemen erfordert neues Wissen und neue Kompetenzen, die im traditionellen Unterricht nicht erworben werden. Damit Schüler am gesellschaftlichen Diskurs teilhaben, der Medienberichterstattung folgen und informiert Urteile fällen können, ist es notwendig, die relevanten Aspekte im Unterricht zu thematisieren. Um ihnen darüber hinaus einen produktiven und kreativ-gestalterischen Umgang mit modernen Informatiksystemen zu ermöglichen, muss Unterricht sich an den Gegebenheiten der realen Welt orientieren und entsprechende Kompetenzen ausbilden. Zwar gibt es bereits zahlreiche Beispiele zur Verwendung von Mikrocontrollern im Unterricht, jedoch werden diese häufig rein als Werkzeug und nicht als Unterrichtsgegenstand thematisiert. Hierzu fehlt bisher ein geeigneter fachlicher Rahmen. Dieser Beitrag untersucht daher Literatur der Fachwissenschaft mit

¹ Universität Potsdam, Didaktik der Informatik, Augut-Bebel-Straße 89, 14482 Potsdam przybyll@uni-potsdam.de

² Friedrich-Alexander-Universität Erlangen-Nürnberg, Didaktik der Informatik, Martensstraße 3, 91058 Erlangen, ralf.romeike@fau.de

dem Ziel, für den Informatikunterricht relevante Inhalte zu extrahieren und geeignet zu strukturieren. Die Ergebnisse können dann genutzt werden, um in diesem Bereich angesiedelten Unterricht zu strukturieren, geeignete Unterrichtsmethoden zu etablieren oder Lern- und Lehrmaterialien zu erstellen, die auf die entsprechende Zielgruppe zugeschnitten sind. Anhand einzelner Beispiele aus einem Projekt zur Gestaltung einer interaktiven Modellstadt wird schließlich gezeigt, wie sich ausgewählte Probleme dieses Fachbereichs im Rahmen von Physical Computing im Informatikunterricht thematisieren lassen.

2 Eingebettete Systeme in der informatischen Schulbildung

Unter anderem durch die immer stärkere Präsenz unzähliger Mikrocontroller in autonomen Fahrzeugen, „Smart“-Entwicklungen und Alltagsgegenständen sind eingebettete Systeme seit einiger Zeit einer der wichtigsten und innovativsten Forschungs- und Entwicklungsschwerpunkte der Informatik. Auch die Schulinformatik beschäftigt sich grundsätzlich seit langem mit diesem Themenbereich. Baumann zeigte beispielsweise in einem historischen Abriss zu Entwicklungen in der Schulinformatik im Forschungsgebiet *Eingebettete Systeme*, dass seit etwa 30 Jahren regelmäßig Artikel zu den Themen *Messen – Steuern – Regeln*, *Prozessdatenverarbeitung* und *Automatisierung* veröffentlicht wurden [Ba11]. Im kürzlich erschienenen LOGIN-Themenheft „Eingebettete Systeme“ [Fa16] werden zahlreiche Unterrichtsbeispiele und verschiedenste Ansätze und Werkzeuge für den Unterricht vorgestellt. Aus fachdidaktischer Sicht analysiert z. B. Stechert ausführlich den Informatiksystembegriff und schlussfolgert, dass es zur Kompetenzentwicklung in diesem Bereich nötig sei, „Informatiksysteme in ihrer Einheit aus Hardware, Software und Vernetzung anhand ihres nach außen sichtbaren Verhaltens, der inneren Struktur und Implementierungsaspekten zu analysieren“ und typische Repräsentanten von Informatiksystemen mit Schülern zu untersuchen und sie zu erklären [St09]. In den GI-Bildungsstandards Informatik für die Sekundarstufe II wird empfohlen, zur Entwicklung von Informatiksystemen maschinell verarbeitbare Sachverhalte der realen Welt zu identifizieren und modellieren. Als typische Einsatzbereiche werden u. a. *Robotik*, *Prozesssteuerung* und *Prozessregelung* genannt [Ar16]. Auch in den neueren Curricula einiger Bundesländer wird diese Entwicklung ersichtlich. So empfiehlt der Berlin-Brandenburger Rahmenlehrplan beispielsweise die Realisierung eines eigenen *Physical-Computing-Projektes* und die Verwendung externer Hardware innerhalb des Themenfeldes *Informatiksysteme* [BM15]. Im Wahlthemenfeld Physical Computing wird die *Charakterisierung eingebetteter Systeme*, die *Verarbeitung elektrischer Größen* oder auch die *Verwendung von Mikrocontrollern* in verschiedenen Kontexten verlangt [BM15]. Lehrkräfte zeigen immer wieder reges Interesse an Fortbildungen, Bausätzen und Unterrichtsmaterialien. Verfügbare Ressourcen beschränken sich jedoch häufig auf die Verwendung von Werkzeugen zur motivierenderen Gestaltung des traditionellen Unterrichts und gehen nur selten über die Grundlagen der Sensor-Aktor-Ansteuerung hinaus (z. B. [Ba11; NW16; PR12]), sodass Phänomene des Ubiquitous Computing mit den bisherigen Ansätzen im Unterricht nicht ausreichend erklärt werden.

3 Phänomene der digitalen Welt: Ubiquitous Computing

Informatiksysteme entwickelten sich in den letzten Jahrzehnten immer mehr entsprechend der Vision des *Ubiquitous-Computing*-Pioniers Mark Weiser, der bereits Anfang der 1990er Jahre allgegenwärtige, effiziente und gleichzeitig unsichtbare Computer vorhersagte [We93]. Diese Vision ist durch entsprechende Technologien heute alltäglich und wird durch das *Internet der Dinge* (engl. Internet of Things, IoT) und *Internet von Allem* (engl. Internet of Everything, IoE) noch ausgeweitet. Im IoT werden „Dinge“, die *eingebettete Systeme* enthalten, untereinander und mit Diensten im Internet vernetzt. Geräte und Maschinen erfassen mit Sensorik ihre Kontexte und übermitteln diese Daten. Insbesondere durch den Zusammenfluss der Daten vieler Dinge entstehen neue Möglichkeiten der Informationsgewinnung, die physikalische Welt verschmilzt mit der virtuellen. Für die Industrie ist dies vor allem im Bereich intelligenter Maschinen von Bedeutung, aber auch in der Heimanwendung finden sich Beispiele: Autos, Haushaltsgeräte oder beliebige Einrichtungsgegenstände können mit Diensten aus dem Internet verknüpft werden. Die Daten werden beispielsweise zur Fahrzeugnavigation, automatisierten Ersatzteilbestellung oder Marktanalyse genutzt. Die Einbindung solcher Dienste ermöglicht schließlich dank Echtzeitanalysen, Einfluss auf die reale Welt zu nehmen. Durch das stetige Anwachsen der Informationsvielfalt im Internet und über das Vernetzen der Dinge mit Personen und Prozessen, wird aus dem IoT immer mehr das IoE [BHV14]. Um hierzu inhaltlich fundierten Unterricht anbieten zu können, bedarf es einer fachlichen Klärung unter didaktischen Gesichtspunkten. Die Sichtung von Lehr- und Lernmaterialien zu den genannten Themen zeigte, dass es eine große Fülle an Lehrbüchern der Fachwissenschaft gibt, die jedoch für eine gänzlich andere Zielgruppe konzipiert worden sind (oft Studierende der Ingenieurwissenschaften und Informatik, Hard- und Software-Entwickler in der Praxis). Diese können nicht unmittelbar für den allgemeinbildenden, wissenschaftspropädeutischen Einsatz im Schulunterricht genutzt werden. Es stellt sich die Frage, wie die Fachstruktur zu beschreiben ist, sodass ihre zentralen Inhalte im Unterricht adäquat vermittelt werden können. Daher wurden vorhandene Lehrbücher mit dem Ziel analysiert, die Grundlagen herauszuarbeiten und Inhalte für die Gestaltung entsprechender Unterrichtseinheiten im Informatikunterricht zu extrahieren und geeignet zu strukturieren. Hierzu wurde in einem ersten Analyseschritt aus der Literatur zunächst zusammengetragen, welche informatischen Inhalte zu den jeweiligen Fachgebieten allgemein thematisiert werden. Im zweiten Schritt wurde der daraus entstandene Katalog hinsichtlich zentraler Aspekte analysiert, also solcher Inhalte, Methoden und Konzepte, die in allen Werken als charakteristische Merkmale des betreffenden Gebietes beschrieben und für das Verständnis erforderlich sind. Diese wurden auf ihre wesentlichen Eigenschaften reduziert. Im letzten Schritt wurden Gemeinsamkeiten zusammengefasst. In den folgenden Abschnitten werden die einzelnen Forschungsgebiete näher charakterisiert.

3.1 Eingebettete Systeme

Eingebettete Systeme (ES) vereinen Hard- und Softwarekomponenten, werden im Gegensatz zu Mehrzweckcomputern in größere Produkte eingebettet und erfüllen in der Regel genau

einen Zweck. Sie sind meist mit physischen Prozessen in ihrer Umgebung durch Sensoren und Aktoren verbunden und erfordern oft keine menschlichen Eingaben (vgl. [VG02]). Beispiele für ES finden sich in allen denkbaren Bereichen unseres Lebens: im Straßen- und Bahnverkehr, in medizinischen Geräten oder im „Smart Home“. Der Verband BITKOM definiert ES insbesondere unter Einbezug ihres typischen Zwecks, Systeme zu steuern, regeln oder überwachen. ES erfüllen vordefinierte Aufgaben, oft unter Echtzeit-Anforderungen [BI10]. Sensor- und Aktor-Schnittstellen ermöglichen es, Veränderungen in der Umwelt zu erfassen und Änderungen an der Umwelt vorzunehmen. Im Gegensatz zu transformationellen Systemen, die Eingabe, Verarbeitung und Ausgabe linear durchlaufen und dann ihr Programm beenden, laufen ES kontinuierlich und interagieren stetig mit ihrer Umgebung. Typische Hardware für den Informatikunterricht stellen Mikrocontroller, wie z. B. Arduino dar.

3.2 Robotik

Ein spezielles Anwendungsgebiet von ES ist die Robotik. Roboter sind insbesondere autonom agierende Maschinen, die in einen Bewegungsapparat eingebunden sind und beispielsweise Arme, Beine oder Räder zur Fortbewegung oder für Greiftätigkeiten nutzen. Sie unterstützen und ersetzen Menschen durch physische Aktivität beim Ausführen konkreter Aufgaben und können ihre Umwelt modifizieren [Si10]. Hertzberg et al. betonen zudem die mit der Autonomie von mobilen Robotern verbundene Schwierigkeit, vorhersehbare Programme zu erstellen. Dadurch, dass all ihre Aktionen von ihrer Umgebung abhängig sind, die erst zur Ausführungszeit bekannt ist, müssen Roboter diese mit Sensoren erfassen, die Daten bewerten und passende Handlungsweisen initiieren [HLN12]. Robotik befasst sich eingehend mit Themen wie Kinetik, mechanischen Bewegungsabläufen, Wahrnehmung, Navigation und Hindernisvermeidung. Roboter können auch dort eingesetzt werden, wo es für Menschen schwierig ist zu arbeiten, z. B. im All, in der Wüste oder im Meer. Sie werden zur Effizienzsteigerung in der Lebensmittel- und Güterherstellung, Land- und Forstwirtschaft oder Medizintechnik verwendet. Immer häufiger finden sie sich beispielsweise in Form von Staubsauger- oder Bodenwischrobotern auch in Haushalten wieder. Unterrichtsbeispiele mit LEGO Mindstorms sind überwiegend der Robotik zuzuordnen.

3.3 Cyber-Physische Systeme

Cyber-physische Systeme (CPS) haben viele Gemeinsamkeiten mit ES und werden in der Literatur oft synonym behandelt. In CPS überwachen und steuern untereinander und mit dem Internet vernetzte ES physische Prozesse. Dies geschieht üblicherweise mit Regelungstechnik, indem Daten aus der Umwelt die Berechnungsprozesse beeinflussen und umgekehrt. Neben der physischen Umgebung (z. B. mechanische Teile, chemische Prozesse, menschliche Bediener, also die direkte Umgebung des Systems) bestehen CPS aus Rechnerplattformen, die neben den Mikrocontrollern auch Sensorik und Aktorik bereitstellen und Netzwerkkomponenten, die die Kommunikation der Teilsysteme untereinander und über das Internet ermöglichen [LS14].

3.4 Gemeinsamkeiten von ES, Robotern und CPS

Obwohl die beschriebenen Systemarten jeweils in verschiedene Richtungen spezialisiert sind, gibt es eine Reihe an Eigenschaften und Anforderungen, die übergeordnet als zentral angesehen werden können und aus denen sich zentrale Konzepte und Ideen ableiten lassen. Grundsätzlich werden in allen Systemen *Sensoren* verwendet, um Eingangssignale auszuwerten, zu verarbeiten und entsprechend der darauf basierenden und durch Software bestimmten Entscheidungen *Aktoren* angesteuert, um etwas zu kontrollieren, zu steuern, zu bewegen – also ein Signal in die Außenwelt zu geben. Man unterscheidet hinsichtlich der Datenerfassung allgemein zwischen *zeitkontinuierlichen Systemen*, die, meist als Regelungssysteme, ununterbrochen Signalströme verarbeiten und *diskreten Systemen*, die ereignis- oder zeitgesteuert diskrete Signale verarbeiten. Je nach Anwendungsbereich müssen ES, Roboter und CPS strengen Anforderungen genügen, wie beispielsweise *Echtzeit-Anforderungen*, innerhalb eines vordefinierten Zeitraums korrekte Ergebnisse zu liefern [VG02, pp. 69-74]. Dies ist vor allem in sicherheitskritischen Bereichen von Bedeutung, wie in automatisierten Zugsystemen oder in der computergestützten Chirurgie. Bei Echtzeitsystemen wird zwischen *weicher*, *moderater* und *harter* Echtzeit unterschieden, also danach, ob ein Überschreiten einer Zeitschranke als Systemfehler gewertet wird oder in gewissem Maß tolerabel ist. Während im ersten Fall ein Versagen katastrophale Folgen haben kann, beeinflusst es in letzterem lediglich die Qualität des Systems. Die Laufzeit kann daher kritisch werden und Einfluss auf die Korrektheit nehmen, statt nur auf Performanz oder Bedienkomfort [LS14]. Die meisten ES und CPS lassen die Ausgabe eines physischen Systems einer gegebenen Referenzeingabe folgen (z. B. Thermostat). Im deutschsprachigen Raum werden hierbei weiter Steuern und Regeln unterschieden: Im Steuersystem wird z. B. das Heizungsventil anhand der Temperatureingabe 22°C entsprechend der Berechnung für die Referenzeingabe geöffnet. In Regelungssystemen wird entsprechend der gemessenen Realtemperatur nachreguliert, sollte die erreichte Temperatur nicht 22°C betragen. Solche Systeme sollen sich sinnvoll verhalten und die Referenzeingabe auch bei Rauschen, Modellfehlern oder Störungen verfolgen. Auch hier gibt es wieder spezielle Anforderungen (vgl. [LS14; VG02]):

- *Stabilität*: Ausgangssignale sollen bei gleichbleibenden Eingangssignalen ihre Werte möglichst nicht verändern
- *Performanz*: gewünschte Ausgabe wird möglichst exakt in möglichst kurzer Zeit an Referenzeingabe angepasst
- *Störungsabwehr*: Minimaler Einfluss von Störungen auf das System
- *Robustheit*: Minimaler Einfluss von Modellfehlern auf Stabilität und Performanz

In aller Regel unterliegen ES, Roboter und CPS in der Entwicklung strengen Vorgaben (sogenannten *Entwurfsmetriken*) bezüglich *Kosten*, *Stromverbrauch*, *Formfaktor* und *Performanz*. Auch die Entwicklungszeit spielt in der Wirtschaft eine große Rolle. Unterschieden wird hierbei zwischen *Time-to-Prototype (TTP)* und *Time-to-Market (TTM)*, die häufig ausschlaggebend für den Erfolg oder Misserfolg eines Produktes sind. Alle genannten

Metriken stehen gegenseitig in Konkurrenz [VG02]. So hat erhöhte Performanz z. B. oft auch einen höheren Stromverbrauch und damit sinkende Energieeffizienz zur Folge. Der Entwurf von ES, CPS und Robotern bringt eine Reihe an Problemstellungen mit sich, aus denen sich typische Methoden und Konzepte zur Problemlösung ableiten lassen. Dazu gehören u. a. *Nebenläufigkeit*, *Trade-Offs* und *Heterogenität*:

- *Nebenläufigkeit*: Prozesse in der realen, physischen Welt finden parallel statt, müssen aber in einer sequentiellen Semantik abgebildet werden. Daher werden Konzepte wie *Interrupts*, *Multitasking* oder *Semaphore* genutzt, um beispielsweise *Nachrichtenaustausch* zu ermöglichen und *Deadlocks* zu vermeiden [LS14].
- *Trade-Offs*: Es muss bei der Gestaltung solcher Systeme abgewogen werden zwischen schnell berechneten, aber weniger genauen und langsamer berechneten, aber akkuraten Ergebnissen [Wo09].
- *Heterogenität*: ES, Roboter und insbesondere CPS sind schwieriger zu analysieren und entwerfen als homogene Systeme, da sie neben der Kombination aus Netzwerkkommunikation und der eigentlichen Berechnung zusätzlich dynamische, physische Prozesse einbeziehen, die nie vollumfänglich vorhersehbar sind [LS14].

3.5 Zusammenfassung der Fachstruktur

Es zeigt sich, dass die beschriebenen Forschungsgebiete stark verzahnt sind und dass das Verständnis bestimmter Inhalte und Konzepte Voraussetzung ist, um andere Gebiete zu erschließen. Da sie die fachliche Grundlage für das gesamte Fachgebiet bilden, sollten die vorgestellten Eigenschaften, Anforderungen und Problemstellungen sowie daraus resultierende Methoden und Konzepte eingebetteter Systeme auch im Informatikunterricht thematisiert werden. Die in Abb. 1 dargestellte Struktur des Fachgebietes verortet die einzelnen Forschungsgebiete im größeren Zusammenhang und stellt ihre Unterscheidungsmerkmale dar. Neben den oben beschriebenen Bereichen gibt es noch zahlreiche weitere Technologien im Kontext der *Mensch-Maschine-Interaktion*, beispielsweise *Interaktive Systeme*, *Wearable Computer* oder auch *Elektronische Textilien*, die sich aus weniger technischen Perspektiven mit dem Gegenstandsbereich eingebetteter Systeme auseinandersetzen.

4 Fachdidaktische Aufarbeitung

Die vorgestellte Struktur kann herangezogen werden, um notwendige Vorkenntnisse der Schüler und Zusammenhänge zwischen den Teilgebieten herstellen und sichern zu können. Zentrale Inhalte und Konzepte eingebetteter Systeme werden zur Entwicklung verschiedenster interaktiver Systeme benötigt, sodass je nach Zielgruppe und Interessen unterschiedliche Themengebiete mit verschiedenen Schwerpunkten als Kontexte im Unterricht genutzt werden können. Auf diese Weise können Kompetenzen zur Analyse, Modellierung und Umsetzung von ES schrittweise entwickelt werden. Das Gestalten eingebetteter Systeme stellt zunächst

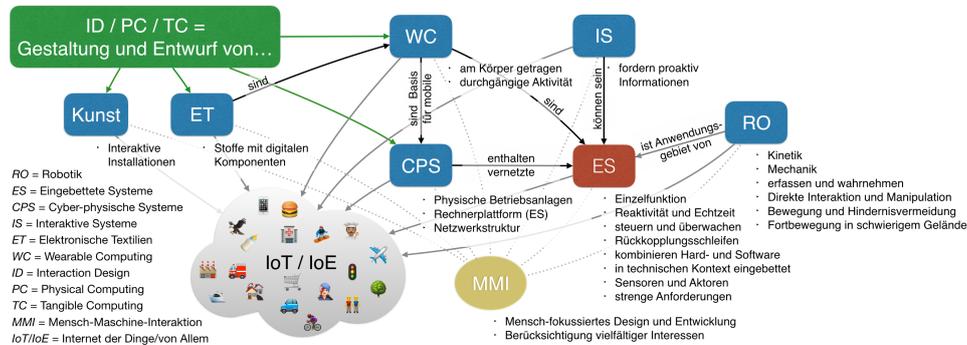


Abb. 1: Überblick über die verschiedenen Forschungsgebiete im Bereich eingebetteter Systeme

eine große Herausforderung dar, die die Beachtung und Bearbeitung zahlreicher Teilaspekte beinhaltet. Daher ist es wichtig, den Lernenden neben der inhaltlichen Auseinandersetzung mit dem Thema auch Werkzeuge und Methoden zur Verfügung zu stellen, die Arbeitsschritte strukturieren und so erfolgreiches Lernen ermöglichen. Eine Möglichkeit hierzu bietet *Physical Computing*: Programmierbare Mikrocontroller bieten als Werkzeug eine attraktive und vielversprechende Heranführung an die Gestaltung eingebetteter Systeme.

4.1 Umsetzung mit Physical Computing

Physical Computing hat sich mit einer stetig wachsenden Gemeinschaft aus Künstlern, Designern und Hobby-Bastlern entwickelt, die Mikrocontroller nutzen, um interaktive Objekte herzustellen. Hierfür werden Sensoren, wie Lautstärke- oder Temperatursensoren und Aktoren, wie Servomotoren oder Lautsprecher verwendet, um kontinuierlich mit ihrer Umwelt zu interagieren. Je nach Ziel des jeweiligen Projektes, werden bestimmte Problemstellungen, Anforderungen und Entwurfsmetriken aus den in Abschnitt 3 analysierten Fachgebieten relevant. Zusammenfassend betrachten wir Physical Computing als die *kreative Gestaltung und Entwicklung interaktiver, physischer Objekte und Systeme*, die als programmierte, greifbare Medien über *Sensoren und Aktoren* mit ihrer Umwelt kommunizieren [PR12]. Physical Computing nutzt Methoden und Ideen eingebetteter Systeme, derer Spezialgebiete und Anwendungen. Es überträgt die traditionell virtuellen kreativen Möglichkeiten der Informatik durch Einbezug von Aspekten aus Kunst und Design in die reale Welt und kann das Fach vielfältiger und damit attraktiver erscheinen lassen. Auch der hohe Motivationswert selbst kreierter Produkte lässt sich für den Informatikunterricht positiv nutzen.

4.2 Beispielumsetzung: Projekttag Smart City

Die nachfolgenden Beispiele von einem Projekttag, der mit 22 Schülerinnen und Schülern der achten bis elften Jahrgangsstufen durchgeführt wurde, illustrieren exemplarisch, wie

Inhalte aus den Bereichen ES, CPS und Robotik in Informatikprojekten verankert werden können. Ziel des Projektes war der Bau einer interaktiven LEGO-Stadt („Smart City“). In einer solchen Stadt gibt es zahlreiche eingebettete Systeme. Diese erfassen an verschiedenen Stellen ihre Umwelt (z. B. Wettereinflüsse, Verkehrsaufkommen, Müllstand) und steuern diese (z. B. Markisen einfahren, Ampelsteuerung anpassen, Müllabholung regulieren).

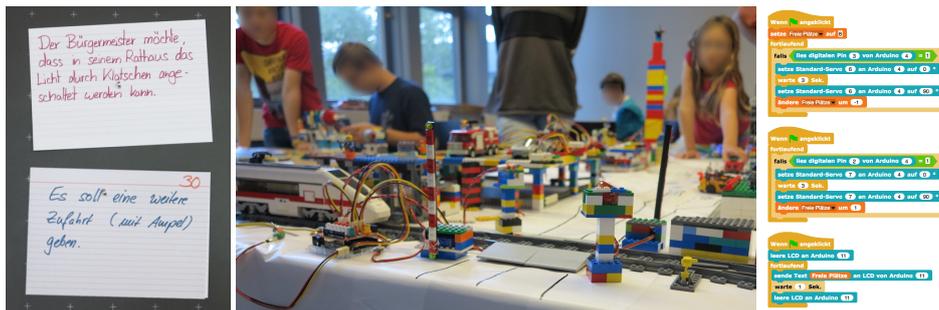


Abb. 2: Schüler beim Bau einer „Smart City“ mit LEGO, Arduino TinkerKit und Snap4Arduino

In kleinen Projektgruppen arbeiteten die Teilnehmer jeweils an Teilprojekten einer interaktiven LEGO-Stadt (siehe Abb. 2). Alle Schüler kamen unabhängig von ihrer konkreten Aufgabe mit gewissen Grundkonzepten eingebetteter Systeme in Berührung, wie beispielsweise der *Kalibrierung* und *Ansteuerung* von *Sensoren* und *Aktoren* oder *zeitkontinuierlichen* und *diskreten* Signalen. Darüber hinaus lernten sie an konkreten, nicht konstruierten Beispielen typische Probleme des Entwurfs eingebetteter Systeme kennen und mit ihnen umzugehen. Durch das Projektziel, die „Smart City“ noch am selben Tag zu präsentieren, wurde eine Situation geschaffen, in der die *Time-to-Market*, wie auch in industriellen Projekten, unausweichlich festgelegt war. Das Projekt wurde mit agilen Methoden umgesetzt, sodass auch die *Time-to-Prototype* begrenzt war, indem die Schüler nach vorab definierten Zeiträumen ihre Prototypen den „Bürgermeistern“ (Betreuern) vorstellen mussten.

Unsere früheren Erfahrungen zeigten, dass es für Schüler häufig eine besondere Herausforderung darstellt, physikalische Prozesse der Umgebung dem Ziel entsprechend zu erfassen. Dies lässt sich mit Physical Computing gut bewerkstelligen, indem sich Unterricht an Leitfragen von O’Sullivan und Igoe [OI04] orientiert, z. B. Was soll das Produkt aus Sicht des Benutzers tun? (*Planung aus Nutzersicht*), Was sind Eingaben, Verarbeitungsschritte und Ausgabe des Produktes aus nicht-technischer Sicht? Welche Ein- und Ausgaben sind diskret, welche zeitkontinuierlich? (*Planung aus Entwicklersicht*). Im Projekt umfasste die Planung aus Nutzersicht eine Beschreibung der Ziele derart, dass sie für die „Bürgermeister“ der Stadt verständlich sein sollte, da von ihnen keinerlei Technikverständnis vorausgesetzt werden durfte. Auch die Bürgermeister selbst konnten Wünsche äußern und an das Projektboard heften. Die Planung aus Entwicklersicht wurde gruppenintern in der Kürze der Zeit nur rudimentär umgesetzt. Die Betreuer regten in den Gruppen entsprechende Diskussionen an.

Während harte Echtzeit in Schulprojekten meist keine Rolle spielen wird, sind *weiche*

Echtzeitanforderungen an die zu erstellenden ES häufig zu erfüllen. Typische Probleme tauchen immer dann auf, wenn Verzögerungen für die Nutzer wahrnehmbar werden. Insbesondere bei gezielten Nutzerinteraktionen ist dies der Fall. In einem Teilprojekt hatten Schüler beispielsweise das Problem zu lösen, dass die Rathausbeleuchtung in der Stadt drahtlos aktiviert werden sollte, jedoch die Übertragungsverzögerung über WLAN deutlich spürbar war. Die Vorhersehbarkeit von Umwelteinflüssen ist ein typisches Problem *heterogener Systeme*. In der Stadt sollten zum Beispiel Fahrzeuge gezählt werden, die einen Parkplatz befahren. Bei der Verwendung eines Helligkeitssensors in der Bodenplatte wurde schnell deutlich, dass Umwelteinflüsse die Zählung verfälschen können. Die Schüler mussten daher eine Lösung für das Problem finden, dass Schwankungen in der Helligkeit über dem Sensor nicht immer bedeuteten, dass ein Fahrzeug die Stelle überfuhr. In Bezug auf Zuverlässigkeit ist *Störungsabwehr* einer der zentralen Aspekte neben Robustheit im Entwurf eingebetteter Systeme. So sollte in der Stadt zum Beispiel ein beschränkter Bahnübergang zuverlässig sensorgesteuert schließen und nach Zugdurchfahrt wieder öffnen. Dabei war es besonders wichtig, keine Fehler in der Durchfahrterkennung zuzulassen.

5 Diskussion

Allgemeinbildender Informatikunterricht steht vor der Herausforderung, zeitbeständige Ideen und Konzepte zu vermitteln und Kompetenzen zu schulen, die in der digitalen Welt von Bedeutung sind. Einige Innovationen der letzten Jahrzehnte und damit zusammenhängende Ideen und Konzepte sind bisher im Schulunterricht jedoch nicht ausreichend verankert: Moderne Informatiksysteme sind immer häufiger eingebettete Systeme, die hochgradig vernetzt und überall zu jeder Zeit verfügbar sind. In diesem Beitrag wurde begründet, dass Themen aus den Bereichen eingebettete Systeme, cyber-physische Systeme und Robotik mit all den dazugehörigen Konzepten (wie z. B. Steuern und Regeln, Nebenläufigkeit oder Trade-Offs) im Informatikunterricht geeignet sind, Schüler dazu zu befähigen, produktiv und kreativ solche Systeme zu gestalten und kompetent mit ihnen umzugehen. Anhand der Erkenntnisse aus den Forschungsgebieten wurde deutlich, dass es zahlreiche gemeinsame Merkmale gibt, die für den Unterricht relevant sind und somit eine fachlich-inhaltliche Grundlage liefern. Die dargestellte Struktur des Fachgebietes kann genutzt werden, um eine inhaltlich logisch aufgebaute Unterrichtsplanung vorzunehmen, Teilgebiete im größeren Zusammenhang zu verorten und geeignete Lehr-Lernszenarien und -materialien für die Schule zu entwerfen. Mit den zugrundeliegenden Konzepten lassen sich auf sehr motivierende und kreative Weise interaktive Objekte und Systeme gestalten, wie z. B. Arbeiten aus der Maker-Kultur zeigen. Es wurde am Beispiel einer Modellstadt aufgezeigt, wie solche Inhalte im Rahmen von Physical Computing kontextualisiert werden können. Unsere Unterrichtserfahrungen zeigen, dass Physical Computing eine attraktive Herangehensweise ist, sich den Inhalten nicht nur auf theoretischer Ebene zu nähern, sondern mit geeigneten Werkzeugen auch praktisch umzusetzen. Beim Entwurf interaktiver, physischer Objekte und Systeme werden Interaktionen zwischen Teilsystemen mit allen dazugehörigen Problemen thematisiert. Diese Inhalte werden mit Physical Computing für den Informatikunterricht auf einer sehr konkreten Ebene zugänglich gemacht.

Literatur

- [Ar16] Arbeitskreis »Bildungsstandards SII« der Gesellschaft für Informatik e.V.: Bildungsstandards Informatik für die Sekundarstufe II, 2016.
- [Ba11] Baumann, R.: Eingebettete Systeme verstehen. Teil 1: Kreatives Experimentieren mit Arduino. LOG IN 171/, S. 33–45, 2011.
- [BHV14] Bojanova, I.; Hurlburt, G.; Voas, J.: Imagineering an Internet of Anything. Computer 47/6, S. 72–77, 2014.
- [BI10] BITKOM: Eingebettete Systeme – Ein strategisches Wachstumsfeld für Deutschland, 2010.
- [BM15] Berliner Senatsverwaltung für Bildung Jugend und Familie; Ministerium für Bildung Jugend und Sport des Landes Brandenburg: Teil C Informatik, 2015.
- [Fa16] Eingebettete Systeme. LOG IN - Informatische Bildung und Computer in der Schule/185/186, hrsg. von Fachbereich Erziehungswissenschaft und Psychologie der Freien Universität Berlin, 2016.
- [HLN12] Hertzberg, J.; Lingemann, K.; Nüchter, A.: Mobile Roboter - Eine Einführung aus Sicht der Informatik. Springer-Verlag, Berlin Heidelberg, 2012.
- [LS14] Lee, E. A.; Seshia, S. A.: Introduction to Embedded Systems - A Cyber-Physical Systems Approach. 2014.
- [NW16] Neutens, T.; Wyffels, F.: Teacher professional development through a physical computing workshop. In: Proceedings of the 11th Workshop in Primary and Secondary Computing Education (WiPSCE '16). S. 108–109, 2016.
- [OI04] O’Sullivan, D.; Igoe, T.: Physical Computing: Sensing and Controlling the Physical World with Computers. Thomson Course Technology, Boston, 2004.
- [PR12] Przybylla, M.; Romeike, R.: My Interactive Garden - A Constructionist Approach to Creative Learning with Interactive Installations in Computing Education. In: Proceedings of Constructionism 2012. Athen, S. 395–404, 2012.
- [Si10] Siciliano, B.; Sciavico, L.; Villani, L.; Oriolo, J.: Robotics: Modeling, Planning and Control. 2010.
- [St09] Stechert, P.: Fachdidaktische Diskussion von Informatiksystemen und der Kompetenzentwicklung im Informatikunterricht, Diss., Universität Siegen, 2009, 45ff., 309.
- [VG02] Vahid, F.; Givargis, T. D.: Embedded System Design: A Unified Hardware/Software Introduction. Wiley, 2002.
- [We93] Weiser, M.: Some Computer Science Issues in Ubiquitous Computing. Communications of the ACM 36/7, S. 75–84, 1993.
- [Wo09] Wolf, W.: Cyber-physical systems. IEEE Computer 42/3, S. 88–89, 2009.