

USBomat

© 2007/2008

Helmar A. Becker

(spyce@gmx.de)

Dieser Text besteht aus Ausschnitten einer Arbeit über eine mögliche Form der Binnendifferenzierung im Informatik-Unterricht, in deren Rahmen ganz unterschiedliche Programmierumgebungen zum Lösen der gleichen Aufgaben eingesetzt werden sollten. Die Aufgabe bestand in der zustandsorientierten Steuerung einer Modell-Ampelschaltung, die aufgebaut und über eine USB-Schnittstellenkarte an den PC angeschlossen wurde. Eine der Programmierumgebungen für das Steuerprogramm bediente sich des zum Programmieren üblichen Konzepts einer (einfachen) formalen Sprache mit den typischen Schwierigkeiten, die diese symbolische Darstellungsform für Einsteiger mit sich bringt. Die andere Arbeitsumgebung (USBomat) erlaubt das graphische Programmieren und nutzt somit eine ikonische Darstellungsform, in der Zustände und Übergangspfeile durch Maussteuerung erzeugt werden. Wenn man ein Zustandsübergangsdiagramm auf dem Papier entwickelt hat, lässt es sich sehr direkt in diesem Programm nachbilden und über die USB-Schnittstellenkarte am Modell ausprobieren.

Lizenz USBomat darf in öffentlichen hessischen Schulen im Unterricht kostenlos genutzt und zu diesem Zweck auch weitergegeben werden. Jede andere Nutzung, insbesondere die kommerzielle Nutzung oder Verbreitung, bedarf der ausdrücklichen Genehmigung des Autors. Jegliche Haftung ist ausgeschlossen.

Die Datei K8055D.DLL ist zum Betrieb des Programms notwendig, auch wenn keine USB-Karte angeschlossen ist. Diese Datei unterliegt dem Copyright der Firma Velleman und ist deshalb nicht beigelegt. Sie steht unter <http://www.velleman.be/ot/de/download/files/> zum Download bereit.

Automatenmodelle Innerhalb der zustandsorientierten Modellierung gibt es verschiedene Automaten-Modelle wie Endlicher Automat, Keller-Automat, Registermaschine oder Turing-Maschine, von denen es meistens in einer deterministischen und einer nicht-deterministischen Variante gibt. Das einfachste Automatenmodell ist dabei der Deterministische Endliche Automat (DEA). Für praktische Zwecke gibt es davon wiederum zwei Varianten, die in der technischen Informatik von großer Bedeutung sind. Automaten vernetzen also im Sinne kumulativen Lernens verschiedene Bereiche innerhalb der Informatik.

Moore/Mealy Die beiden DEA-Varianten unterscheiden sich in ihrem Ausgabe-Verhalten: Beim einfacheren Moore-Automaten wird die Ausgabe nur vom aktuellen Zustand bestimmt, während sie beim Mealy-Automaten eine Funktion von Zustand und aktuellem Eingabe-Symbol ist, also letztlich vom Zustandsübergang abhängt. Beide Automaten sind im Prinzip gleichwertig und ineinander umformbar, ein Mealy-Automat ist aber aufgrund seiner größeren Vielfalt oft kompakter als ein Moore-Automat, der die gleiche Aufgabe löst. Der Moore-Automat hat demgegenüber jedoch den Vorzug, leichter verständlich zu sein, denn im gleichen Zustand erfolgt immer die gleiche Ausgabe. Im Hinblick auf

Ampelschaltungen ist der Moore-Automat das angemessene Modell, denn dann entspricht jeder Ampelphase (Zustand) eindeutig eine Festlegung, welche Lichter leuchten.

Ampelschaltung In den Empfehlungen des Didaktischen Forums sind Technikmodelle als Additum empfohlen, die auch im Schulcurriculum als Themenkreis 'Steuern und Regeln' gerne behandelt werden – sofern Technikmodelle samt brauchbarer Steuerungssoftware zur Verfügung stehen. Die Ampelschaltungen sollen daher nicht simuliert, sondern als Modell selbst aufgebaut und vom Computer über eine USB-Schnittstellenkarte angesteuert werden. Die Schülerinnen und Schüler erleben so vielleicht zum ersten Mal direkt, dass der Computer auch außerhalb des gewohnten Wirkungskreises (Monitor, Drucker, andere vernetzte Rechner) physisch etwas bewirken kann. Dadurch ist auch eine höhere Motivation bei den Schülerinnen und Schülern zu erwarten. Das Aufbauen der Ampel- und Kreuzungsmodelle erleichtert den Modellbildungsprozess und bietet einen enaktiven Zugang zum Thema. Der Themenkreis Steuern und Regeln wird darüberhinaus für einige der Schüler auch beruflich relevant werden.

Realisiert werden soll die Ampelschaltung zum einen mit selbstgebauten Ampeln aus Leuchtdioden, die auf einem Lego-Brett plaziert und mit ihren Anschlusskabeln an die Buchsen der USB-Karte angeschlossen werden (siehe Technische Aspekte, ??). Zum anderen können die HIBS-Kreuzungen verwendet werden, die eine komplette Kreuzung zweier Straßen samt Fußgängerampeln nachbilden. Die Lego-Variante hat den Vorzug, dass das Ampelmodell selbst aufgebaut wird (enaktiv) und dass sie ganz unterschiedliche Schaltungen erlaubt – für den Einstieg beispielsweise eine einfache Fußgängerampel. Man kann so das Anspruchsniveau beliebig anpassen, und für den Einstieg wirkt die HIBS-Kreuzung recht komplex, insbesondere wenn man nur einen Teil ihrer Funktionalität nutzen möchte.

Fachübergreifende Bezüge bestehen vor allem zur Physik, und es bietet sich auch die Chance zu fächerverbindendem Unterricht. Man kann die Funktionsweise der Leuchtdioden ansprechen, den Zusammenhang zwischen Spannung, Widerstand und Strom, die Schülerinnen und Schüler können den benötigten Vorwiderstand selbst berechnen und sie Ampeln auch selbst löten. Ein Bezug zur Mechanik (Geschwindigkeitskontrolle) findet sich weiter unten.

Graphisches Programmieren Um die Probleme mit der Syntax formaler Sprachen ganz zu vermeiden, wurden Konzepte der graphischen Programmierung entwickelt. Meist werden in diesen Programmierumgebungen Anweisungen und Kontrollstrukturen als graphische Elemente dargestellt und können mit der Maus aneinanderplaziert oder mit Linien verbunden werden, die als 'roter Faden' den Ablauf der Befehls-Elemente festlegen. Grafischen Programmierumgebungen ist gemeinsam, dass sie die Benutzerin oder den Benutzer an die Hand nehmen und ihr/ihm eine begrenzte Anzahl von Möglichkeiten zur Auswahl bieten, die in das graphische Programm eingefügt werden können. Im Gegensatz dazu darf die/der Benutzer/in bei formalen Programmiersprachen im Prinzip alles eintippen, also auch beliebigen 'Unsinn', und welche Worte (Befehle) in welcher Anordnung einen Sinn ergeben, muss

man im Prinzip auswendig wissen¹.

USBomat Die Entwicklung von USBomat ist zum einen geleitet von dem Gedanken, das Programmieren noch einfacher zu gestalten und sprachliche bzw. syntaktische Probleme auszublenden. Es widmet sich außerdem dem Problem, dass die meisten Einstiege ins Programmieren (ob graphisch oder sprachlich) den modellbildenden Charakter nur wenig betonen – mit Ausnahme von Kara², das als Anregung für USBomat diente. Oft neigen die Schülerinnen und Schüler dazu, einfach 'drauf los' zu programmieren und herumzuprobieren. Um aber sinnstiftend Konzepte der Informatik zu erlernen, soll es sich beim Programmieren "um die Implementierung eines vorher entwickelten Modells handeln, also keinesfalls um Stegreifprogrammierung"³. USBomat fordert das Entwickeln eines Zustandsmodells dadurch ein, dass das Zustandsmodell bereits dem graphische Programm entspricht. USBomat kann auch ansatzweise als Werkzeug zur Entwicklung eines Zustandsmodells genutzt werden. Die Zustände lassen sich benennen, und die Ausgabe besteht aus den Schaltzuständen der einzelnen Ausgänge der USB-Karte. Übergänge werden mit der Maus gezogen. Als Eingabe lassen sich die Eingänge der USB-Karte nutzen, aber auch zeitgesteuerte Übergänge sind möglich⁴. USBomat zeigt die aktuelle Ein- und Ausgabe am unteren Bildschirmrand an, und die Eingänge können durch die Tastatur (Tasten 1 bis 5) simuliert werden, so dass man auch ohne angeschlossene Karte schon vieles ausprobieren kann. Widersprüchliche Übergänge werden als Fehler rot hervorgehoben, damit sie sofort erkannt werden können.

Ein Unterschied zum Moore-Automaten besteht darin, dass nicht von jedem Zustand aus ein Übergang für jeden Tastendruck angegeben werden muss. Nicht spezifizierte Eingaben bewirken dann einfach keine Zustandsänderung. Aus diesem Grund sind auch Übergänge von einem Zustand zu sich selbst weder nötig noch implementiert.

USB-Schnittstellenkarte Das technische Herzstück dieses Vorhabens ist die USB-Schnittstellenkarte "USB Experiment Interface Board VM110" (auch als K8055 bezeichnet) von Velleman, die es erlaubt, mit dem PC verschiedene Geräte zu steuern. Die Karte ist bei beispielsweise bei Conrad für 45€ erhältlich – oder als Bausatz zum Selbstlöten für 30€. Letzteres lohnt wohl nur, wenn die Schüler das Löten übernehmen, doch dazu ist bereits etwas Löterfahrung empfehlenswert. Die Karte besitzt 8 digitale und 2 analoge Ausgänge sowie 5 digitale und 2 analoge Eingänge. Über Relais lassen sich auch 230V-Geräte schalten. Es lassen sich bis zu vier Karten an einen PC anschließen und unabhängig voneinander steuern.

Mitgeliefert wird eine DLL, mittels derer man die Karte aus beliebigen Programmiersprachen heraus ansteuern kann. So ist die Ansteuerung über Delphi auch

¹Oder jedesmal in der Hilfe nachlesen, aber das bremst ungemein. Moderne IDEs unterstützen ihre Benutzer/innen durch Codevervollständigung von Befehlen samt ihrer Parameter, so dass man auch hier je nach Kontext aus einer Liste passender Befehle auswählen kann.

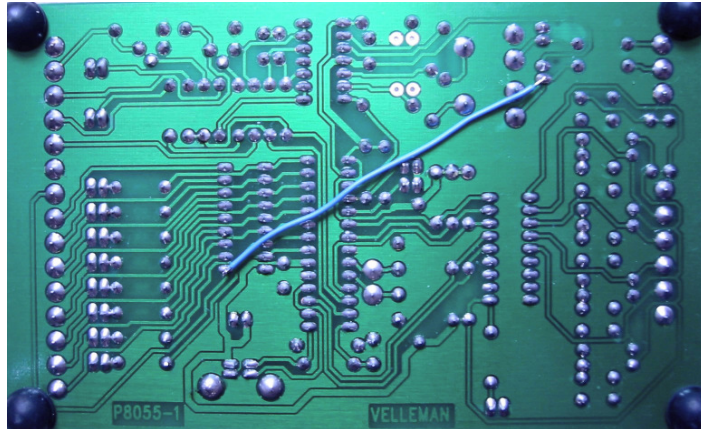
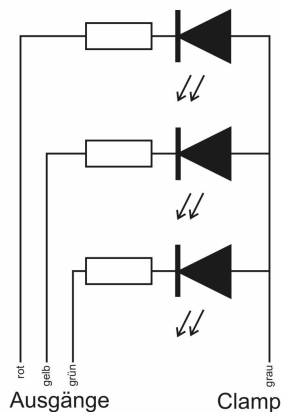
²<http://www.swisseduc.ch/informatik/karatojava/kara/>

³P. Hubwieser: "Didaktik der Informatik", Springer Verlag 2000, S. 86ff

⁴Genau betrachtet wird der Automat durch die Zeitabhängigkeit eigentlich sehr kompliziert, denn die verstrichene Zeit ist im Prinzip Teil des Zustands des Automaten. Diese Betrachtung ist allerdings weder handhabbar noch hilfreich, deshalb wird hier einfach das Verstreichen einer eingestellten Zeit als eine Eingabe modelliert.

für Schülerinnen und Schüler gut handhabbar, wenn man die Schnittstelle zur DLL durch eine Unit mit einfacheren Befehlen implementiert (wie beispielsweise in [?]). Mit netzwerkfähigen Programmen lassen sich die Karte und die angeschlossenen Geräte sogar fernsteuern.

Ampelschaltung (Lego) Die Kreuzungsmodelle werden mithilfe von Lego-Basisplatten und -Bausteinen realisiert, die samt Vorwiderstand (hier 470Ω für Spannungen bis 12V) auf kleine Platinen gelöteten LED-Ampeln (Schaltskizze links) sind mit Heißkleber auf Legosteinen befestigt und lassen sich daher sehr einfach im Modell fixieren und wieder lösen (s.u.).

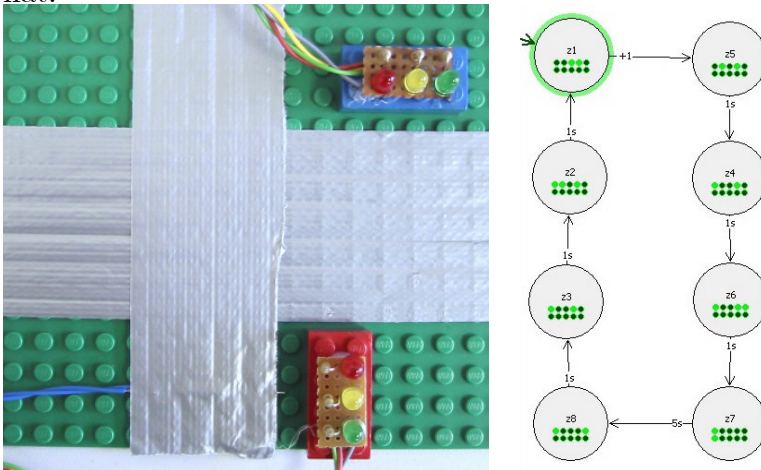


Die Straßen lassen sich sehr gut durch ein breites, undurchsichtiges Klebeband darstellen, auf das man Fußgängerüberwege oder ähnliches leicht aufmalen kann. Für die Fußgänger gibt es extra Drucktaster (ebenfalls auf Legosteine geklebt) und als Modell für die Induktionsschleifen dienen kleine Reed-Kontakte, die zwischen die Noppenreihen der Lego-Platte passen und so wirklich unter der Straße (Klebeband) liegen. Die Reed-Kontakte werden ausgelöst durch Magnetfelder, ein gewöhnlicher Magnet kann als Modell für ein Auto dienen, aber an einem Modellauto kann man auch einen kleinen Magneten befestigen.

Um mit den Anschlusskabeln nicht durcheinanderzukommen, sind alle Kabel farbig; rote, gelbe und grüne Adern steuern die entsprechenden LEDs, der gemeinsame Plus-Pol (grau) wird in den CLAMP-Anschluss der Karte gesteckt. Schalter und Reed-Kontakte haben blaue Kabel (siehe Bild weiter unten).

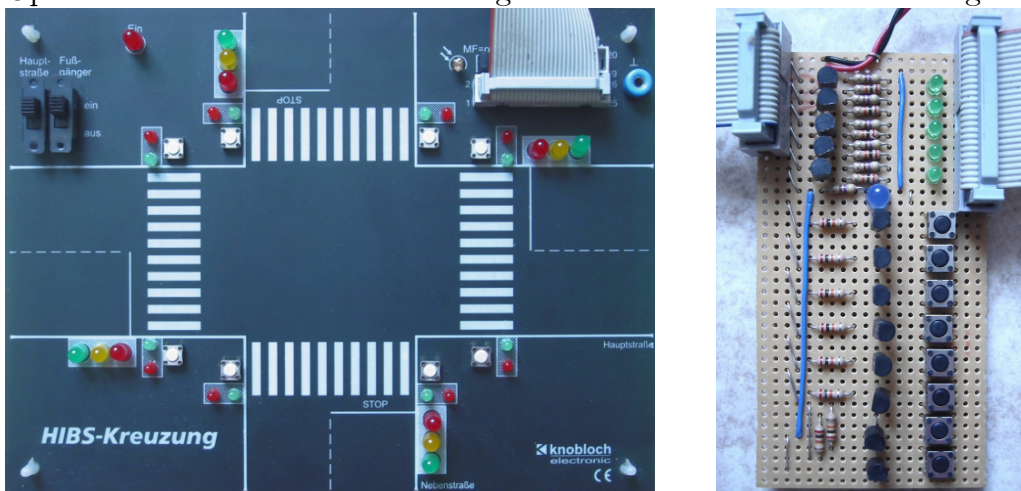
Die Karte liefert keine Spannungsversorgung und verlangt eigentlich eine externe Spannungsquelle, allerdings kann man leicht die 5V-Versorgung des USB-Ports anzapfen und so für ein geringeres Kabel-Durcheinander sorgen. Dazu kann man vorsichtig eine Brücke auf der Rückseite der Karte anlöten, so dass auf CLAMP 5V anliegen (rechtes Bild). Manche Geräte wie die HIBS-Kreuzung oder eine große Sieben-Segment-Anzeige benötigen jedoch eine höhere Spannung, und dann darf keine Verbindung zwischen dieser Spannung (CLAMP) und der 5V-USB-Versorgung liegen. Um hier einfach wechseln zu können, führt die gezeigte Lötbrücke zu einer Steckbrücke auf der Karte, die normalerweise zum Justieren des zweiten Analogeingangs dient; bei gesteckter Lötbrücke werden nun die 5V vom USB-Anschluss an CLAMP angelegt, andernfalls nicht. **Achtung:** Keine Externe Spannungsversorgung bei gesteckter Brücke anschließen! – Der Signalpegel-Regler des zweiten Analogeingangs muss ganz herunter gedreht werden, damit die höhere Spannung den Baustein

nicht gefährdet! Dieser Eingang ist dann nicht nutzbar! Die Lötbrücke läßt sich aber auch leicht und rückstandsfrei wieder entfernen, wenn man sie vorsichtig angelötet hat.



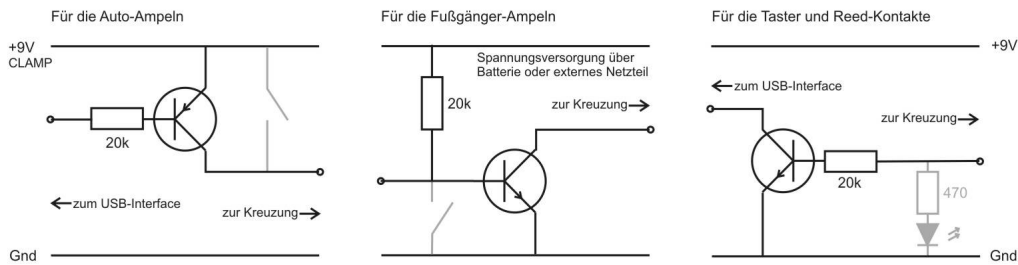
HIBS-Kreuzung An vielen hessischen Schulen existiert wahrscheinlich noch eine alte HIBS-Kreuzung, ein Modell einer Autokreuzung mit Haupt- und Nebenstraße samt Fußgängerüberwegen und allen dabei auftretenden Ampeln, Tastern und Reedkontakten (linkes Bild). Allerdings ist wahrscheinlich meist nur eine vorhanden, und hergestellt wird sie nicht mehr, soweit mir bekannt ist. Damit viele Schülerinnen und Schüler damit arbeiten können, ist entweder das Ausleihen weiterer Kreuzungen von anderen Schulen nötig, oder man nutzt die Lego-Variante.

Wegen der unterschiedlichen Signal-Arten von USB-Karte und Kreuzung ist ein Adapter nötig, dessen Bau ohne geeignetes Werkzeug zum Ätzen einer Platine unangenehm viel Aufwand macht (rechtes Bild). Taster und Kontroll-Leuchten auf dieser Platine erlauben die leichte Identifikation der Funktionen der Ein- und Ausgänge und einen manuellen Betrieb zum Ausprobieren, sind aber nicht unbedingt nötig. Der Adapter ist direkt auf die Kreuzung zugeschnitten, und andere HIBS-Modelle lassen sich damit wahrscheinlich nicht betreiben. Ein Adapter auf der Basis von Operationsverstärkern wäre diesbezüglich wahrscheinlich etwas vielseitiger.



Zum Adapter: Die USB-Karte hat open collector Ausgänge (NPN) und die Eingänge müssen nach Erde geschlossen werden. Die Eingänge der Kreuzung für die Hauptampeln müssen aktiv nach 5V gezogen werden, dafür ist die linke der drei skizzierten Schaltungen in sechsfacher Ausfertigung nötig. Die beiden Fußgängeram-

pele zeigen ohne angelegtes Signal grün und lassen sich durch ziehen auf Erde rot schalten. Sie ließen sich damit direkt an die USB-Karte anschließen, aber es ist eher kontraintuitiv, dass ein ausgeschalteter Ausgang eine grüne Ampel bewirkt. Zum Invertieren dient daher die mittlere Schaltung (zweifache Ausfertigung). Die Taster und Reed-Kontakte schalten nach Plus, und zur Anpassung ist die rechte Schaltung in fünffacher Ausfertigung zuständig. Die Kreuzung hat mehr Ausgänge als die USB-Karte Eingänge, und man muss einige Reed-Kontakte weglassen oder zusammenlegen.



Weitere Einsatzmöglichkeiten Über die Steuerung einer Ampel hinaus bietet die USB-Karte viele Möglichkeiten, von denen einige kurz angerissen werden sollen:

Man kann eine Sieben-Segment-Anzeige anschließen, die von vielen Geräten her bekannt und in Größen bis 10cm erhältlich ist. Damit kann man einen einfachen Zähler aufbauen (leider nur bis 9). Dieses Beispiel läßt sich gut als Einführung in die Zustandsmodellierung, die USB-Karte und die Programme nutzen.

Im thematischen Zusammenhang mit der Ampelsteuerung läßt sich auch eine Geschwindigkeitskontrolle realisieren, indem man in einem bestimmten Abstand zwei Reed-Kontakte unter die Fahrahn legt. Wenn die Zeit zwischen den Auslösungen einen eingestellten Wert unterschreitet, kann eine LED aufblitzen. Dabei müssen die Schülerinnen und Schüler diese Zeit aus Maximalgeschwindigkeit und Kontaktabstand berechnen (Bezug zur Mechanik).

Möglich ist auch der Bau eines elektronischen Zahlenschlosses, das bei Eingabe einer bestimmten Tastenfolge einen elektrischen Türöffner auslösen kann – die Möglichkeiten reichen von einer symbolischen Öffnung durch Aufleuchten einer LED bis zum Öffnen einer wirklichen Tür, deren Öffner über ein Relais angeschlossen wird. Besonders interessant ist an diesem Beispiel der Bezug zur theoretischen Informatik (13.1), denn hierbei handelt es sich um ein einfaches Beispiel zur Erkennung regulärer Sprachen, das hier spielerisch im Sinne vorwegnehmenden Lernens behandelt werden kann, aber auch noch in der Oberstufe.

Sogar die Erkennung von rhythmischen Knopfdruck-Folgen ist realisierbar, wie man sie als geheime Klopfzeichen aus Filmen kennt. Mit einem Mikrofon und ein wenig Elektronik ließe sich sogar wirklich eine Tür durch ein vereinbartes Klopfsignal öffnen – bei einem solchen Projekt wäre ein hohes Motivationspotential zu erwarten.