



**Fonds für Unterrichts- und Schulentwicklung
(IMST-Fonds)**

S6 „Anwendungsorientierung und Berufsbildung“

BAU UND PROGRAMMIERUNG EINER COMPUTERGESTEURTEN AMPELANLAGE

ID 1394

Gerhard LECHNER

Hauptschule Imst – Oberstadt,
Franz – Xaver – Renn - Straße 18
6460 Imst

Imst, Mai 2009

INHALTSVERZEICHNIS

ABSTRACT	3
1 EINLEITUNG	4
2 ZIELE/AUFGABENSTELLUNG	4
2.1 Ziel 1	4
2.2 Ziel 2	5
3 DURCHFÜHRUNG/METHODEN	6
3.1 Erarbeitung der Grundlagen der Elektronik.....	6
3.2 Programmieren mit LOGO	7
3.3 Herstellung der Ampelschaltung	12
3.4 Programmieren der Ampelschaltung.....	16
4 ERGEBNISSE/EVALUATION	17
4.1 Die zehn Merkmale guten naturwissenschaftlichen Unterrichts	18
4.2 Evaluation durch Interviews	19
4.3 Erreichte Ziele.....	20
5 DISKUSSION/INTERPRETATION/AUSBLICK	21
6 LITERATUR	21

ABSTRACT

Mit dem Projekt "Bau und Programmierung einer computergesteuerten Ampelanlage" verfolgte ich im Wesentlichen die Verwirklichung zweier Ziele.

Und zwar einerseits den Schüler/innen fächerübergreifend verschiedene Kenntnisse und Fertigkeiten zu vermitteln, wie:

- *einfache Kenntnisse der Grundlagen der Elektronik,*
- *Kennenlernen einer Programmiersprache im Informatikunterricht*
- *Bau einer elektronischen Schaltung*
- *Programmieren der Ampelanlage*
- *Präsentation des fertigen Projektes den Mitschülern*

Das zweite Ziel des Projektes war es, den Schüler/innen neben den klassischen Bereichen des Informatikunterrichts, wie Textverarbeitung, Tabellenkalkulation und Umgang mit dem Internet, auch andere Inhalte zu bieten, zumal sich der Unterricht in Informations- und Kommunikationstechnik teilweise „Office-lastig“ darstellt.

Schulstufe: 7. Schulstufe, 3. Klasse Hauptschule

Fächer: Physik, IKT (= Informations- und Kommunikationstechnik)

Kontaktperson: Gerhard Lechner

Kontaktadresse: Frans – Xaver – Renn – Str. 18, 6460 Imst

E-Mailadresse glechner@hs-imsto.tsn.at

1 EINLEITUNG

Unsere Schule, die Hauptschule Imst-Oberstadt, ist eine von drei Hauptschulen in Imst, im Tiroler Oberland. Wir ca. 30 Lehrer/innen, unterrichten im Schuljahr 2008/09 ca. 300 Schüler/innen in 13 Klassen.

An unserer Schule beschäftigen wir uns schon sehr lange mit dem Informatikunterricht und so haben wir 2002 einen IKT-Schwerpunkt eingeführt. Die Schüler/innen besuchen alle 4 Jahre jeweils 2 Wochenstunden den autonomen Pflichtgegenstand „Informations- und Kommunikationstechnik“, kurz IKT genannt.

Ziel des IKT-Unterrichtes ist es, den Schülern/innen eine umfassende Grundausbildung auf ECDL¹-Niveau zu vermitteln und den Computer als vielseitiges Werkzeug kennen zu lernen. Die meisten Schüler/innen erreichen auch dieses Niveau und erwerben das ECDL-Zertifikat.

Diese Entwicklung führte jedoch dazu, dass der Inhalt im IKT-Unterricht sehr „ECDL-lastig“ wurde.

Es ergab sich für dieses Schuljahr die Situation, dass ich in einer dieser Schwerpunktklassen Mathematik, Physik und Informatik unterrichtete und auch Klassenvorstand bin, was organisatorische Vorteile mit sich bringt.

Dazu kam, dass ich bereits vor ca. 15 Jahren mit meinen damaligen Schülern/innen ein Lauflicht in Form einer elektronischen Kartonschaltung herstellte und auch mit BASIC² programmierte.

2 ZIELE/AUFGABENSTELLUNG

Mit dem Projekt „Bau und Programmierung einer computergesteuerten Ampelanlage“ verfolgte ich im Wesentlichen die Verwirklichung zweier Ziele.

2.1 Ziel 1

Ein Ziel war es, den Schüler/innen fächerübergreifend verschiedene Kenntnisse und Fertigkeiten zu vermitteln, wie:

- einfache Kenntnisse der Grundlagen der Elektronik,

¹ ECDL = European Computer Driving License, der europäische Computerführerschein

² BASIC“ steht für „**B**eginner’s **A**ll-purpose **S**ymbolic **I**nstruction **C**ode“, was so viel bedeutet wie „symbolische Allzweck-Programmiersprache für Anfänger und wurde vor allem im Informatik-Unterricht verwendet

- Kennenlernen einer Programmiersprache im Informatikunterricht
- Bau einer elektronischen Schaltung
- Programmieren der Ampelanlage
- Präsentation des fertigen Projektes den Mitschülern

2.2 Ziel 2

Das zweite Ziel des Projektes war es, den Schüler/innen neben den klassischen Bereichen des Informatikunterrichts wie Textverarbeitung, Tabellenkalkulation und Umgang mit dem Internet auch andere Inhalte zu bieten, da sich der Unterricht in Informations- und Kommunikationstechnik teilweise „Office-lastig“ darstellt.

Dieses zweite Ziel verfolgte noch den Aspekt, dass auch andere Informatik-Kollegen sich motivieren lassen, Programmierunterricht und -techniken in den Informatik-Unterricht einzubauen.

Es ist durchaus denkbar, dass die gebauten Ampelschaltungen und die erstellten Unterlagen zum Programmierunterricht den interessierten Kollegen zur Verfügung gestellt werden könnten.

3 DURCHFÜHRUNG/METHODEN

Die Durchführung des Projektes begann damit, dass ich den Schülern/innen zunächst den Sinn und den Zweck des IMST-Fonds vorstellte und ihnen dann konkret den Ablauf des geplanten Projektes im Laufe des kommenden Schuljahres näher brachte. Das ganze geschah bereits zu Schulanfang im September.

3.1 Erarbeitung der Grundlagen der Elektronik

Der Lehrplan für Hauptschulen der 3. Klassen (7. Schulstufe) sieht im Physikunterricht des Kapitel Elektrotechnik/Elektronik vor, und zwar konkret (Auszug)

Elektrische Phänomene sind allgegenwärtig:

Ausgehend von Alltagserfahrungen sollen die Schülerinnen und Schüler immer intensiver mit grundlegenden elektrischen Vorgängen im technischen Alltag und in Naturvorgängen vertraut gemacht werden.

- verschiedene Spannungsquellen als Energieumformer und einfache Stromkreise verstehen; Gleichstrom und Wechselstrom, Stromstärke, Spannung, Widerstand, das Ohm'sche Gesetz;

-elektrische Erscheinungen in Technik und Natur erklären können.

Den Schülern/innen wurde zunächst der Kernstoff der 3. Klasse vermittelt, wie einfacher Stromkreis, elektrische Ladungen, Gleich- und Wechselstrom, Stromstärke, Spannung, Widerstand, Ohm'sches Gesetz, usw.

Als Erweiterungstoff habe ich dann im Zeitraum von ca. 3 Wochen auch die wichtigsten Halbleiterbausteine erarbeitet und vorgestellt:

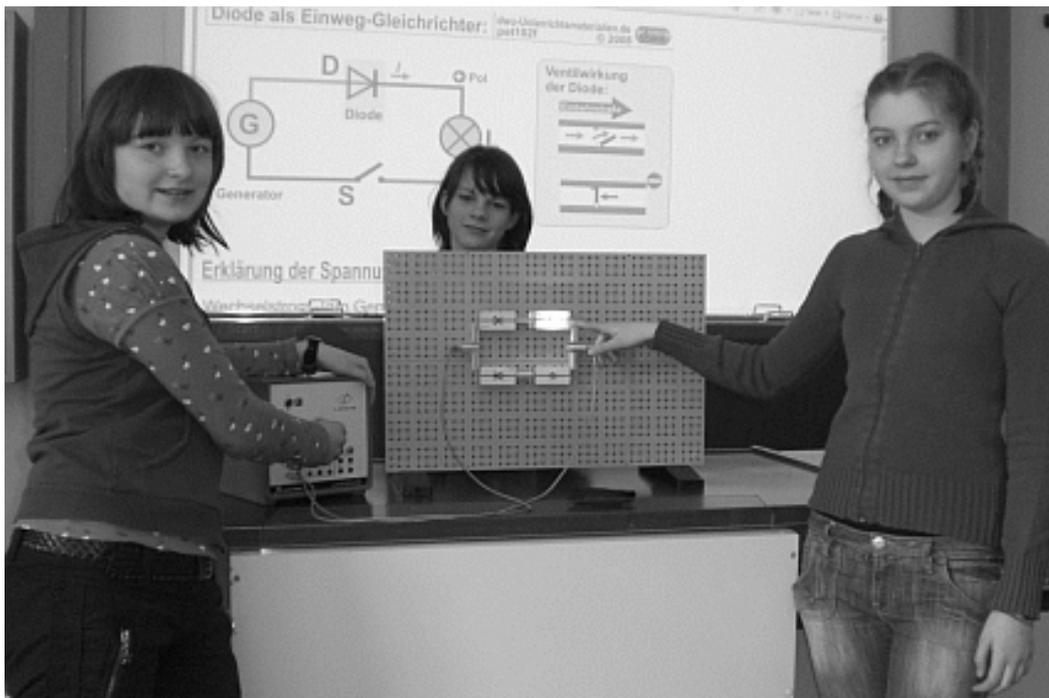
- Funktion und Einsatzmöglichkeiten der verschiedenen Widerstände (z.B. lichtabhängiger Widerstand),
- die Diode als Gleichrichter bzw. als „Sperrventil“ einschließlich der Leuchtdiode, LED
- den Kondensator als Halbleiterbaustein mit „Speicherfunktion“
- den Transistor als vielseitigen Baustein, z.B. Verstärkerwirkung in der Darlington-Schaltung

Als Abschluss der Erarbeitung der elektronischen Grundlagen habe ich den Schülern einen kurzen Überblick über die geschichtliche Entwicklung der Halbleitertechnik gegeben und zwar am Beispiel „Entwicklung der Rechenmaschinen/Computer“

- Blaise Pascal; erfand 1642 eine rein **mechanische** Rechenmaschine
- Konrad Zuse; baute 1941 den ersten funktionsfähige Rechner auf **Relais**-Basis - die Z3 gilt daher als erster funktionsfähiger Computer.

- ENIAC³, der erste Universalrechner mit **Elektronenröhren** der US-Armee
- Heinz Zemanek⁴; ein österreichischer Computerpionier, baute den ersten **volltransistorisierten** Computer namens Mailüfterl.
- um 1970, Aufkommen der ersten Rechner auf Basis der IC, = integrated circuits, also der **integrierten Schaltkreise**, kurz chips genannt

Anschließend erhielten die Schüler/innen den Auftrag über einen Zeitraum von ca. drei Wochen, selbständig im Internet zu recherchieren und eine Zusammenstellung der „Entwicklung der Rechenmaschinen/Computer“ zu erstellen.



(Bild 1: Die Diode als Gleichrichter – Versuchsaufbau und Erklärung)

3.2 Programmieren mit LOGO

Ein wesentliches Ziel des Informatik-Unterrichtes soll es meiner Meinung nach sein, dass die Schüler/innen den Computer als „vielseitig verwendbares und einsetzbares Werkzeug“ kennenlernen, eben ein „Universal-Werkzeug“.

³ Electronic Numerical Integrator and Computer

⁴ Heinz Zemanek, geb. 1. Jänner 1920 in Wien, lebender österreichischer Computerpionier; auch laut Lehrplan (Zitat) soll auf Beiträge österreichischer Wissenschaftlerinnen und Wissenschaftler, Forscherinnen und Forscher, Technikerinnen und Techniker sowie Erfinderinnen und Erfinder besonders einzugehen.

Heinz Zemanek spielt mit dem Namen Mailüfterl auf die in den USA in Betrieb genommenen (Röhren)-Rechner an: „Wenn es auch nicht die rasante Rechengeschwindigkeit amerikanischer Modelle erreichen kann, die ‚Wirbelwind‘ oder ‚Taifun‘ heißen, so wird es doch für ein Wiener ‚Mailüfterl‘ reichen.“ Das Mailüfterl steht heute im Technischen Museum Wien.

Da die Vermittlung der ECDL-Inhalte zwar sehr umfangreich sind aber doch eher klassische Büroanwendungen, wie Textverarbeitung, Tabellenkalkulation, Einsatz von E-Mail als Kommunikationsmittel, usw. beinhalten, ist das Kennenlernen einer einfachen, schülergerechten Programmiersprache eine wertvolle Ergänzung des Informatik-Unterrichtes.

In den ersten beiden Unterrichtsstunden besprach ich mit den Schülern/innen das Grundsätzliche des Programmierens. Zu diesem Zwecke erstellte ich eine kleine HTML-Seite, wo die Schüler alles Wissenswerte nachlesen konnten. (Zitat)

Was bedeutet eigentlich "Programmieren"?

Ein Computer ist dumm. Er kann eigentlich nur eines: Rechnen. Das kann er dafür aber sehr, sehr schnell. Um nun den Computer dazu zu verleiten, etwas Sinnvolles zu tun, benötigt er eine Liste von Anweisungen, die er auszuführen hat. Solch eine Liste von Anweisungen nennt man Programm. Das Erstellen einer solchen Liste wird als Programmieren bezeichnet, Personen, die diese Tätigkeit ausführen, heißen Programmierer.

Programmieren heißt also nicht mehr, als Programme zu schreiben. Im Alltag verwendet man das Wort "Programmieren" oftmals in einem anderen Bezug, denn viele Leute sagen, sie würden ihren DVD-Rekorder "programmieren", obwohl damit lediglich gemeint ist, dass dem DVD-Rekorder Anweisungen gegeben werden, wann er welchen Film aufzunehmen hat.

Und genau hier wird es deutlich, was der Begriff "Programmieren" eigentlich bedeutet.

Programmieren heißt nichts anderes, als Anweisungen (Befehle) erteilen.

Wenn du programmierst, "sagst" du dem Computer was er tun soll. Und wenn du nichts falsch gemacht hast, tut der Computer auch genau das, was du willst.

Du wirst dich jetzt sicherlich nach dem "Wie" fragen. Wie kann ich meinem Computer Befehle erteilen - oder besser: Wie programmiere ich meinen Computer?

Um zu zeigen, wie leicht es der "moderne" Programmierer heute hat, werde ich einen kurzen Rückblick in die "Urzeit" des Computers wagen. Um dir keine falsche Zeitvorstellung zu geben, mit der Urzeit meint man ungefähr die Mitte des zwanzigsten Jahrhunderts.

Wie du ja weißt, kann Strom nur fließen oder nicht fließen. Dies kann man mit zwei Zuständen darstellen: 1 (Eins; Strom fließt) und 0 (Null; Strom fließt nicht). So ein Zahlensystem nennt "Binärsystem".

Aber nun wieder zum Thema zurück. Es ist euch hoffentlich klar, dass man dem Computer alles in binärer Weise mitteilen muss, was er zu tun hat:

(z.B.01100101011110001010101010101111000001110101010)

Glücklicherweise hat man bald versucht, sich die Arbeit zu erleichtern und man begann damit, eine Assembler-Sprache zu entwickeln. Assembler wird oftmals auch als "niedere Programmiersprache" bezeichnet. Der Assembler übersetzt den Code - welcher in

*Textform vorliegt - in Binärcode. Ein typischer Assemblerbefehl wäre "ADD", um Zahlen zu addieren:
ADD 4,9*

Die Assembler-Sprache ist sehr kompliziert, deshalb hat man die "Hochsprachen" entwickelt. Die Anweisungen, also die Programmier-Befehle werden in Textform geschrieben. Damit der Computer etwas mit diesen Anweisungen anfangen kann, braucht man ein Programm, das den Text so "über-

setzt", dass der Computer ihn verstehen kann. Ein solches Programm nennt man "Compiler" oder "Interpreter".

Es gibt sehr viele Programmiersprachen (über 8000), mehr wie "normale" Sprachen (ca. 6500). Einige bekannte Programmiersprachen sind: Basic, PHP, Java, Perl, C++, uva.

Wir werden die Sprache LOGO verwenden, die speziell für den Unterricht entwickelt wurde.

(Übrigens: Auf www.99-bottles-of-beer.net kannst du dir ein und dasselbe Programm in über 1000 verschiedenen Programmiersprachen anschauen.)

Wie funktioniert das "Programmieren"?

Ganz allgemein kann man das Programmieren in folgende Punkte unterteilen:

- *Problemanalyse: Es wird ein Problem daraufhin analysiert, in welche Teilprobleme man es zerlegen kann (so genannt Top-Down, also „vom Groben ins Feine“), da ein riesiges Programm wie z. B. eine Textverarbeitung nie am Stück programmiert werden kann.*
- *Entwerfen von geeigneten Algorithmen: Ein Algorithmus ist eine Handlungsvorschrift, wie z.B. ein Kochrezept oder eine Gebrauchsanweisung. Das bedeutet, man zerlegt die zur Lösung des Problems nötigen Abläufe in kleine Schritte.*
- *Kodierung: Das Übersetzen (Kodieren) dieser Algorithmen (Schritte) in einer Programmiersprache*
- *Testen des fertigen Programmes*

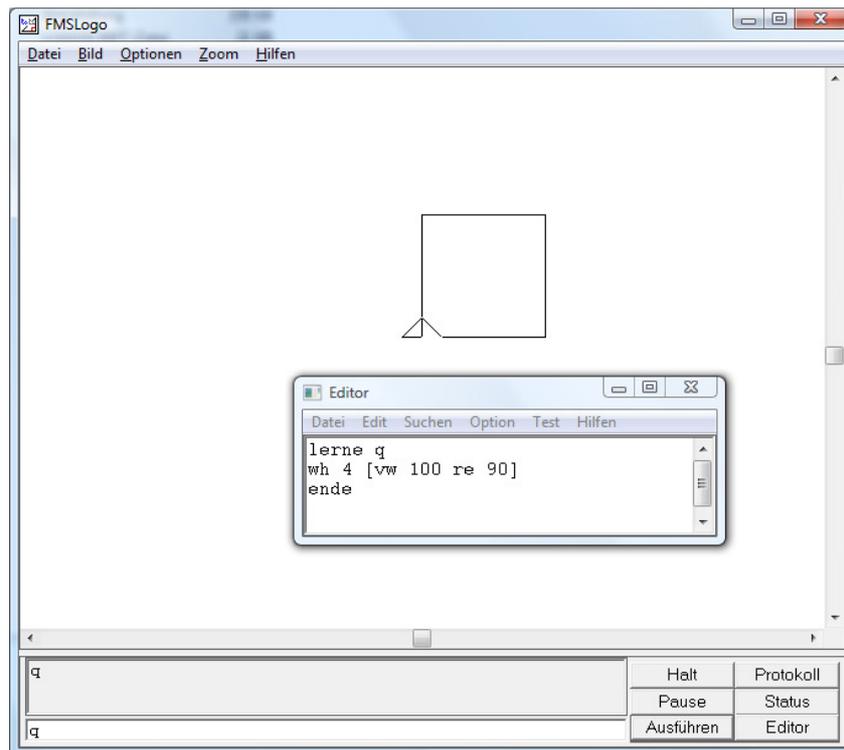
In diesem Zusammenhang erschien es mir auch sehr wichtig, dass die Schüler/innen einen kleinen Einblick in die Berufswelt des Programmierers erhalten.

Anschließend begann ich die Programmiersprache FMSLOGO⁵ vorzustellen. Dabei erstellten wir zuerst einfache Programme mit der TURTLE-Grafik, hier zum Beispiel die Prozedur für das Zeichnen eines Quadrates

⁵ Dabei stehen die Buchstaben FMS nicht für Microsoft, sondern für Fight Multiple Sklerose. Die Entwickler dieser Sprache gründeten einen Sozialfond und werben für ihn mit ihrem Softwareprodukt, um den Menschen, die an dieser schrecklichen Krankheit leiden, zu helfen.

Die Grundzüge der Interpretersprache Logo wurde schon in den 70er Jahren von dem Amerikaner **Seymour Papert** für die Schule entwickelt, um das entdeckende Lernen zu fördern. Heute ist Seymour Papert Wegbereiter der programmierbaren kabellosen LEGO-Technik (Mindstorms).

Die Programmiersprache ist mit einer relativ einfachen Befehlssyntax aufgebaut und steht im Internet als Freeware zur Verfügung (siehe - <http://sourceforge.net/projects/fmslogo>)



(Bild 2: Screenshot – ein mit FMSLogo programmiertes Quadrat)

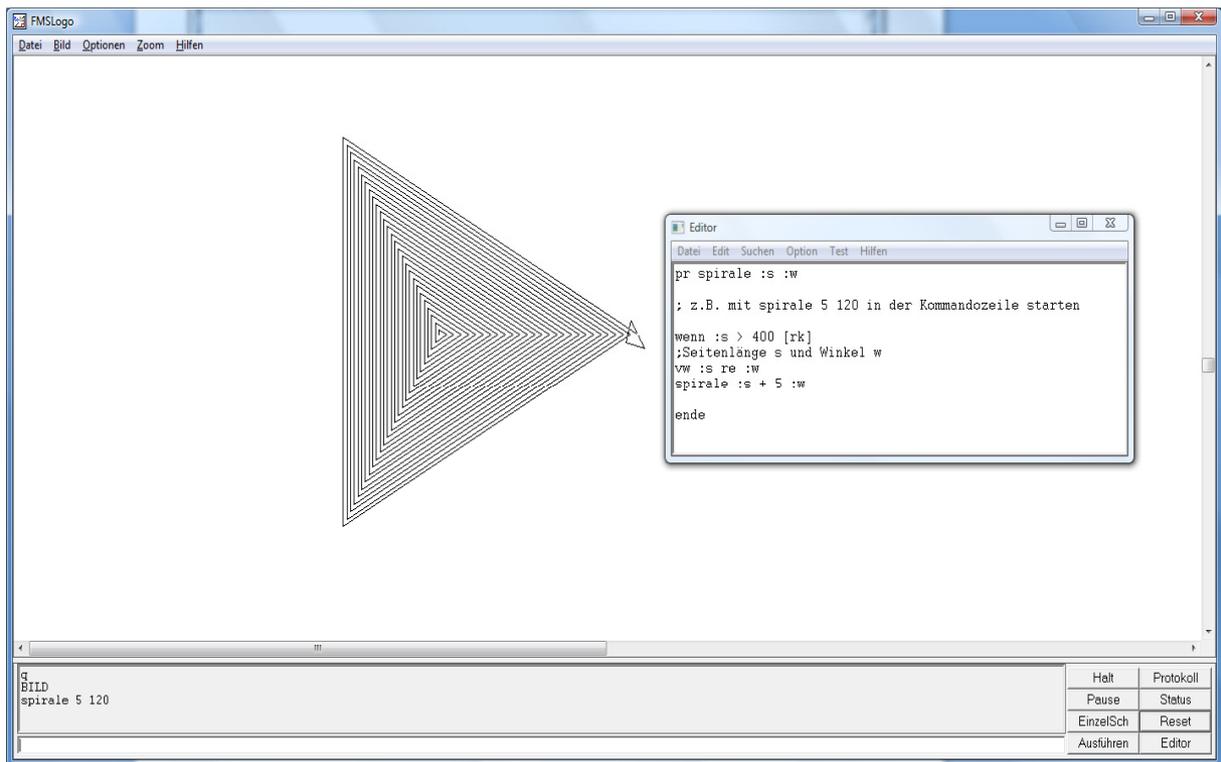
Dabei entwickelten viele Schüler selbständig Ideen, um eigene Figuren und Muster zu programmieren. LOGO stellt damit einen idealen Einstieg in die Welt des Programmierens dar.

Nach dem Erstellen einfacher Figuren kam der nächste Schritt. Nämlich der Einsatz von Variablen, die in LOGO durch einen Doppelpunkt definiert werden. Um ein beliebiges Vieleck zu Zeichnen sieht der Programmcode folgendermaßen aus:

```
lerne Vieleck:Seite :Ecken
wh :Ecken [ vw :Seite re 360/ :Ecken ]
ende
```

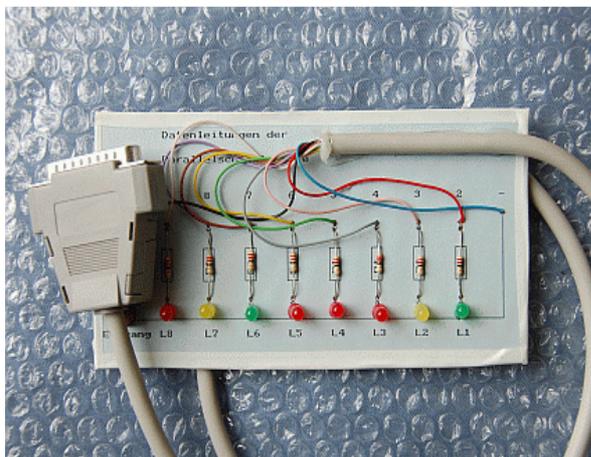
Hierbei sind *:Seite* und *:Ecken* die zwei Variablen, *wh* steht für wiederhole, *vw* für vorwärts und *re* für rechts.

Die nächste Ebene war dann die Einführung des „Rekursionsbegriffes“, heißt also, dass sich ein Befehl immer wieder selber aufruft. Im folgenden Bild wurde mit Hilfe einer Rekursion eine Dreiecksspirale gezeichnet



(Bild 3: Screenshot – eine mit FMSLogo programmierte Spirale als Rekursion)

Nach einer mehrwöchigen Programmierpause begann dann das Programmieren einer Karton-Schaltung, die ich vor ca. 15 Jahren mit einer Schülergruppe herstellte.



(Bild 4: die Kartonschaltung)



(Bild 5: Programmieren am Computer)

Die Kartonschaltung besteht aus 8 LED's, die über die parallele Schnittstelle (LPT1) mit dem Computer verbunden wird. FMSLOGO besitzt mit AUSG⁶ einen eigenen Befehl um jede der 8 LED's anzusteuern.

⁶ die genaue Syntax des Befehls lautet:
AUSG port wert, mit AUSG 888 4 wird also die dritte LED, mit AUSG 888 128 die achte LED angesteuert.

Die Kartonschaltung eignet sich auch sehr gut dazu, den Schülern/innen das binäre Zahlensystem und die ASCII-Codierung zu erklären, da die 8 LED's der binären Zahlendarstellung folgen:

Diode	L8	L7	L6	L5	L4	L3	L2	L1
Wert	128	64	32	16	8	4	2	1
	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0

Als Programmieraufgaben wurden verschiedene Aufgaben gestellt wie:

- Schalte abwechselnd die Lampen 1-4 und 1-8 ein (blinken)
- Schalte abwechselnd die Lampen 1,3,5,7 und dann 2,4,6,8 ein (blinken)
- ein Lauflicht (wie es von Autobahn-Baustellen bekannt ist)
- das Lauflicht soll in eine Richtung langsam und zurück sehr schnell abfolgen
- usw

Das Programmieren dieser Kartonschaltung hat den Vorteil, dass die Schüler/innen ein unmittelbares Feedback auf jeden Programmierschritt erhalten, oder sehr einfach gesagt: die LED „brennt“ oder eben nicht

Nach den verschiedenen Lauflichtvariationen begann die Programmierung einer Ampelschaltung, da ja die 8 LED's in den Farben rot, grün und gelb so angeordnet sind, dass sich eine Ampelschaltung ergibt.

Das Programmieren einer Ampel schafften eigentlich die meisten Schüler problemlos, da es im Vergleich mit den Lauflichtprogrammen keine wesentlichen neuen Schritte braucht. Doch hier stellte sich bald heraus, dass durch die lineare Anordnung der LED's es für die Schüler sehr schwierig war, das Zusammenspiel zweier Ampeln zu programmieren.

Und so kam es zum nächsten Schritt im Projektablauf.

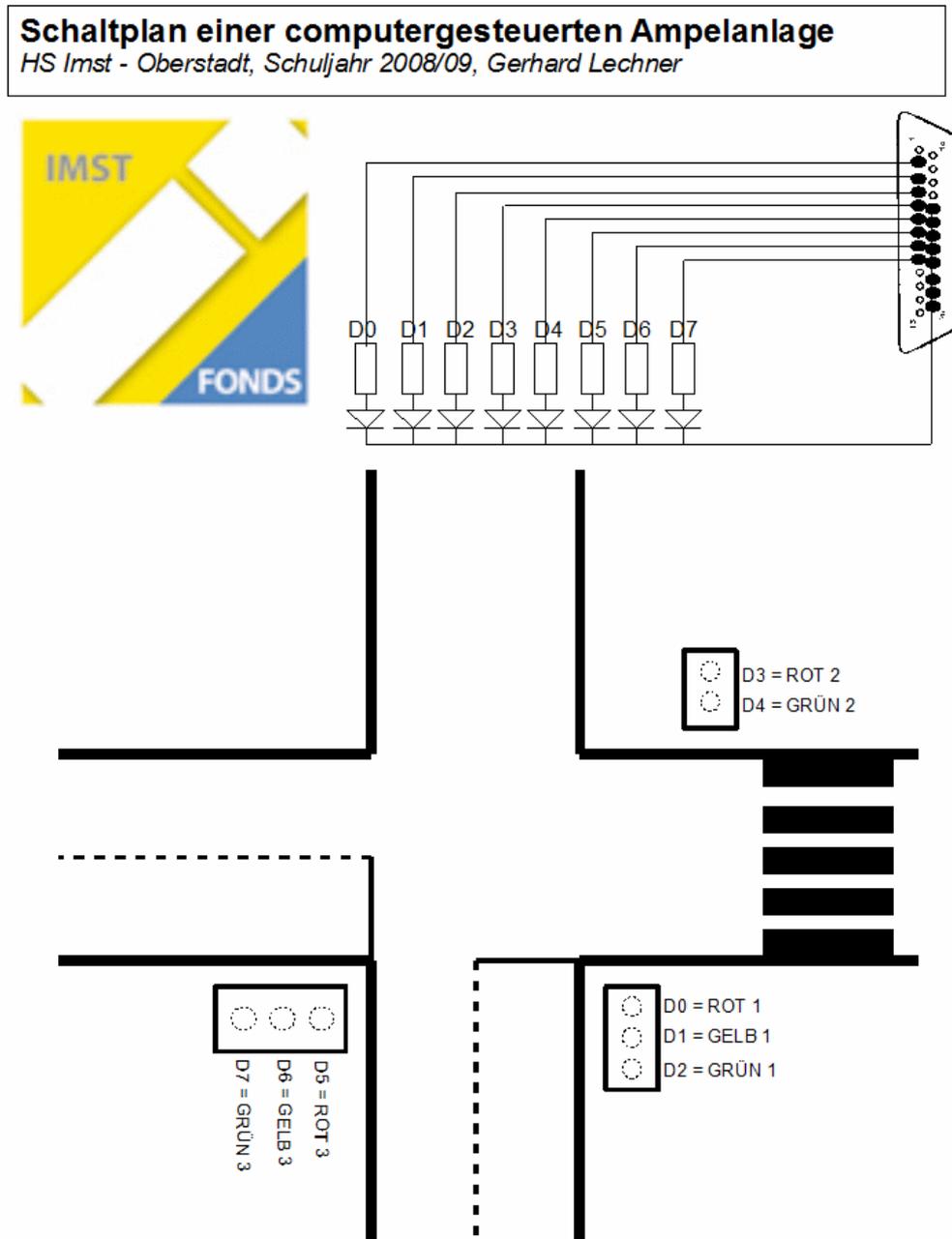
3.3 Herstellung der Ampelschaltung

Das Herstellen von elektronischen Schaltungen ist im Lehrplan der Hauptschulen für Technisches Werken enthalten (Zitat):

Lesen und Verstehen von elektrischen und einfachen elektronischen Schaltplänen. Erarbeiten der Funktionsweise von Bauteilen und Anwenden bei einfachen Grundschaltungen insbesondere zum Regeln und Steuern.

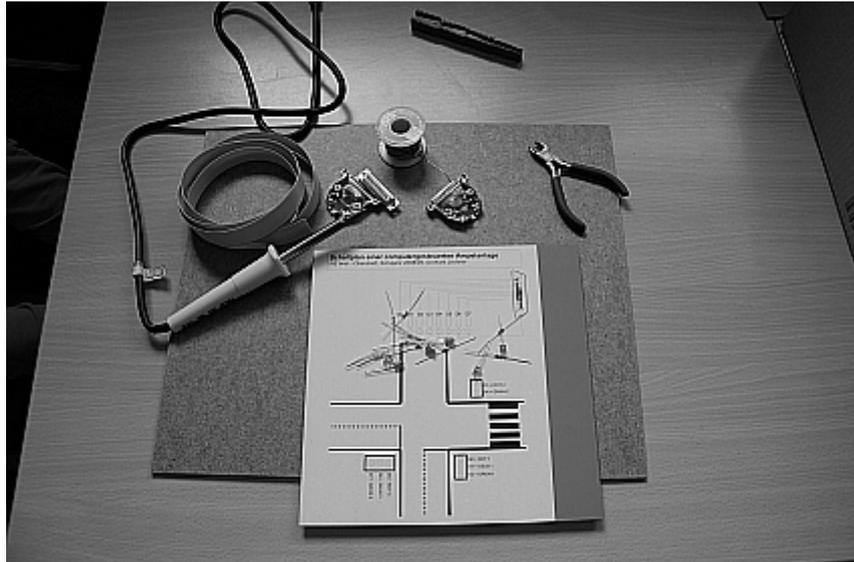
Zunächst galt es alle notwendigen Bauteile bei einem Elektronik-Versand zu bestellen: LED's in drei Farben, Widerstände, 9-polige Datenleitungen, Steckerleiste für die parallele Schnittstelle, Lötcolben, Heißkleber, uvm.

Zusätzlich erstellte ich mit einem Zeichenprogramm einen Schaltplan der Ampelschaltung in der Größe DIN A4 her, den ich ausdruckte, auf Karton klebte und anschließend folierte.

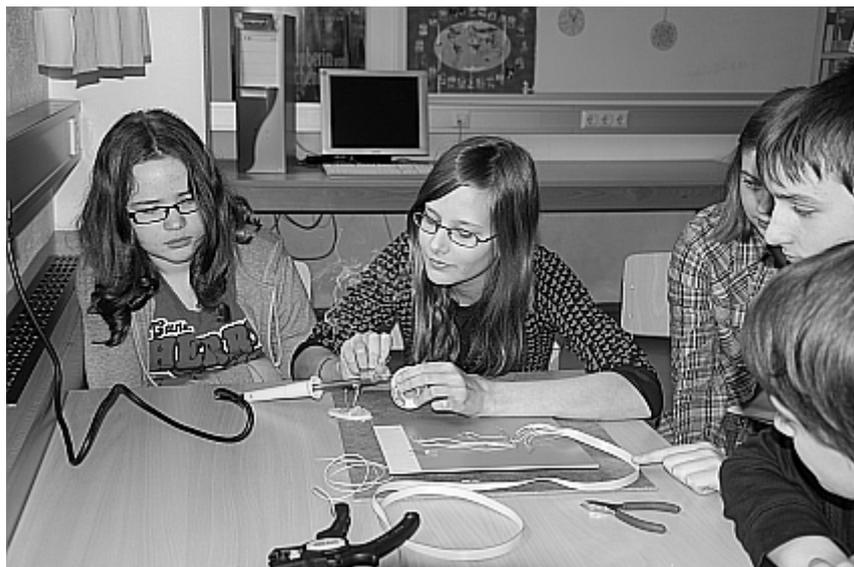


(Bild 6: die Ampelschaltungsvorlage)

Dann ging es daran, die Ampelschaltung „händisch“ herzustellen. In zwei IKT-Stunden zeigte ich den Schülern/innen das Löt- und verdrahten der Schaltung, wobei ich darauf achtete, dass alle Schüler/innen einmal „Löten konnten“. Auch die Feinarbeit, das Einlöten der Leitungen auf die Steckerleiste der parallelen Schnittstelle, wurde gemeinsam durchgeführt, so dass schlussendlich zwei Ampelschaltungen erfolgreich fertig gestellt werden konnten.

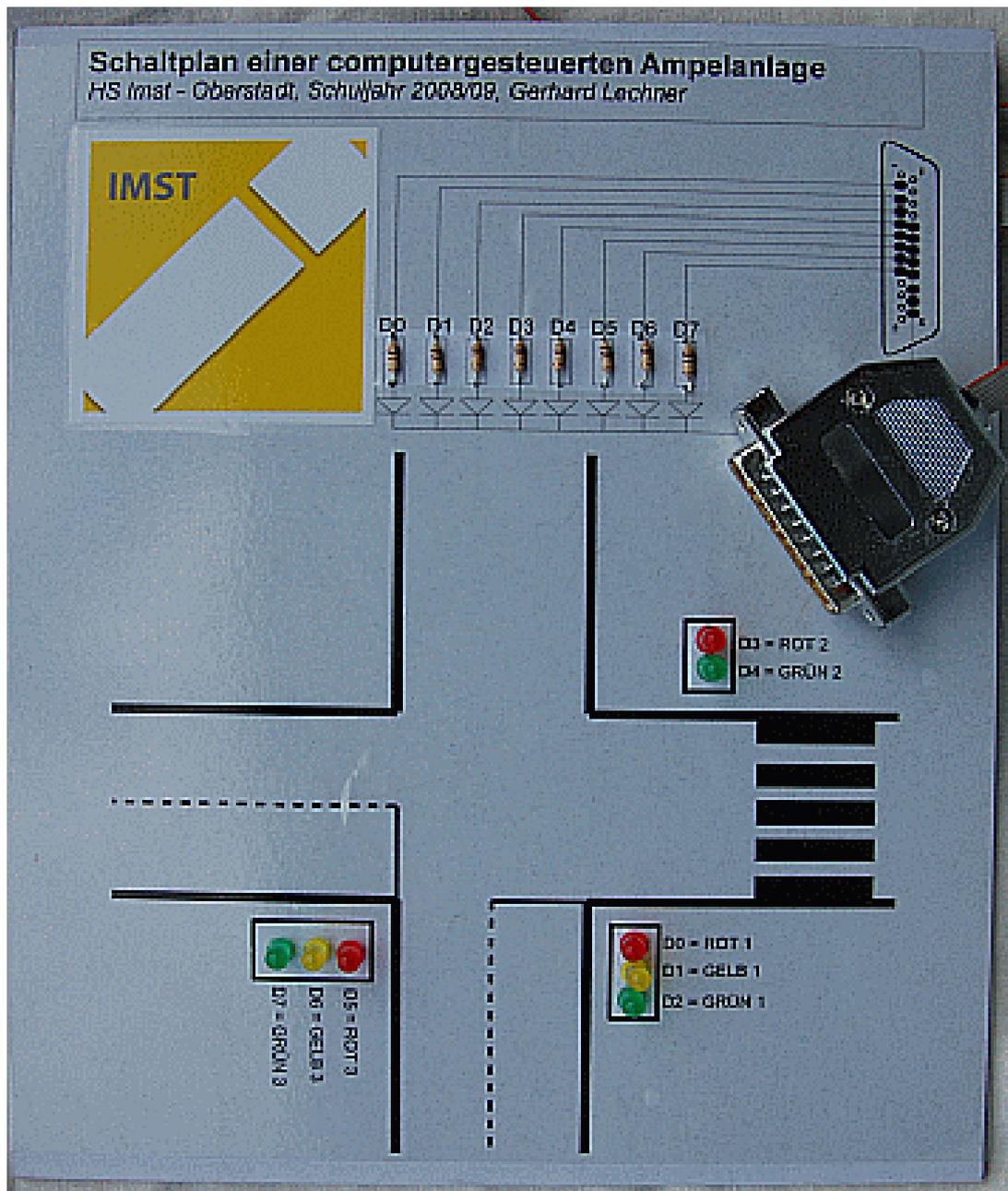


(Bild 7: das benötigte Arbeitsmaterial)



(Bild 8: natürlich versuchten auch die Mädchen zu löten)

Die Schüler/innen hatten natürlich vorher noch nie eine elektronische Schaltung hergestellt bzw. gelötet. So war ich auch nicht allzu sehr überrascht, als sich einige Schüler und auch zwei Schülerinnen spontan bereit erklärten, zu Hause selbständig eine Ampelschaltung herzustellen. Das benötigte Material, Werkzeug und eine detaillierte Anleitung habe ich den Schülern/innen mitgegeben und nach ca. zwei Wochen auch tatsächlich fertig gestellte Ampelschaltungen erhalten. Lediglich zwei der fünf Schaltungen haben kleine Mängel aufgewiesen, so dass nach einem kleinen Reparatüreingriff schlussendlich sieben fertige und funktionstüchtige Ampelschaltungen zur Verfügung standen.



(Bild 9: fertige Ampelschaltung)



(Bild 10: Schüler/innen mit den fertigen Ampelschaltungen)

3.4 Programmieren der Ampelschaltung

Nachdem also sieben funktionsfähige Ampelschaltungen vorhanden waren, ging es am Schluss nur mehr um die Programmierung der Ampel(n). Dabei stellte sich heraus, dass sich die Schüler/innen wesentlich leichter taten, diese Ampelschaltung zu programmieren, als die Ampelschaltung mit 8 linear angeordneten LED's.

Der Grund liegt sicher darin, dass diese Anordnung viel realistischer ist und beim Zusammenspiel von mehreren Ampeln einfacher zu überblicken.

Das Programmieren der Ampelanlagen erfolgte natürlich im IKT-Unterricht. Die sieben Ampeln ergaben also die sieben Gruppen und damit sieben Programmier-Teams.

Die eigentliche Programmerstellung erfolgte stufenweise – vom Einfachen zum Schwierigen – wobei sich folgende Aufgaben ergaben.

- nur Ampel 1
- Ampel 2 und die Fußgängerampel
- Ampel 1 und Ampel 2 (ohne Fußgängerampel)
- alle drei Ampeln zusammen

Die Aufgaben 1 und 2 schafften alle Teams ohne Hilfestellung meinerseits. Bei Aufgabe 3 brauchten 1-2 Teams eine kleine Hilfestellung. Für die schwierigste Aufgabe, das Zusammenspiel aller drei Ampeln, gab ich zuerst eine Hilfestellung für alle Teams und dann starteten wir die ersten Schritte gemeinsam. Schlussendlich hatten alle Teams in 2 Unterrichtsstunden die vier Aufgaben gelöst. Hier das Programm-Listing für die Aufgabe 4 (Zusammenspiel von drei Ampeln):

```
pr ampel :t :w
; t zeit ungefähr 10 und w wiederholungen

ausg 888 0

wh :w [
ausg 888 137
warte :t * 50

wh 4 [ausg 888 9
warte :t * 4
ausg 888 137
warte :t * 4]

ausg 888 75
warte :t * 10

ausg 888 52
warte :t * 50

wh 4 [ausg 888 32
warte :t * 4
ausg 888 52
warte :t * 4]

ausg 888 106
warte :t * 10]

ausg 888 0
ende
```

4 ERGEBNISSE/EVALUATION

Um eine Interpretation der Ergebnisse zu erlangen, möchte ich mich an einen Vortrag im Rahmen des Frühjahrs-Workshop des Schwerpunktes 6 „Anwendungsorientierung und Berufsbildung“ von Hr. Dr. Edwin Scheiber, Sir-Karl-Popper-Schule, Wien mit dem Titel „Bausteine guten naturwissenschaftlichen Unterrichts“⁷ anlehnen.

⁷ Dr. Edwin Scheiber hat im Auftrag des IMST3-MNI-Fonds naturwissenschaftliche Projekte analysiert um ein (Zitat) „Werkzeug zu erarbeiten, das Lehrer/innen der Naturwissenschaften als Planungsunterstützung und Ideenpool für die konkrete Unterrichtsarbeit hilfreich sein soll“, siehe auch <http://work.popperschule.at/publikationen/bausteine>

4.1 Die zehn Merkmale guten naturwissenschaftlichen Unterrichts

- Klare Strukturierung des Unterrichts
- Hoher Anteil echter Lernzeit
- Lernförderliches Klima
- Inhaltliche Klarheit
- Sinnstiftendes Kommunizieren
- Methodenvielfalt
- Individuelles Fördern
- Intelligentes Üben
- Transparente Leistungserwartungen
- Vorbereitete Umgebung

Natürlich waren in meinem Projekt nicht alle Bausteine im gleichen Ausmaß vertreten und darum möchte ich einige Punkte besonders hervorheben:

Diese klare Strukturierung war auf alle Fälle in hohem Maße gegeben, da ja die einzelnen Teilbereiche, wie Erarbeitung der Grundlagen, Erlernen der Programmiersprache LOGO, Herstellung der Ampelschaltung jeweils getrennt und mit klaren Teilzielen stattgefunden hat.

Es war auch ein hoher Anteil echter Lernzeit gegeben, da die Schüler/innen sehr viel selbständig und im Team arbeiten mussten.

Es war auch ein lernförderliches Klima gegeben, da speziell bei der Herstellung der Ampelanlage eine gemeinsam geteilte Arbeitsverantwortung gegeben war.

Es lag auch inhaltliche Klarheit vor, denn die Aufgabenstellung für die Schüler/innen war verständlich und der thematische Ablauf klar.

Ein wesentlicher Punkt für den erfolgreichen guten Unterrichts ist die „vorbereitete Unterrichts- und Lernumgebung“. Dazu zählten:

- die Vorbereitung und Versuchsaufbauten für die physikalischen Experimente (Grundlagen der Elektronik),
- die Erstellung einer Einleitung in das Programmieren speziell eine Einführung in FMSLOGO,
- das Bereitstellen der notwendigen Materialien und Werkzeuge zur eigentlichen Herstellung der Ampelschaltung.
- die Hilfestellung für die Programmiererteams beim eigentlichen Programmieren

Zusammenfassend stelle ich in einigen Punkten große Übereinstimmung mit den von Dr. Scheiber aufgestellten Bausteinen fest.

4.2 Evaluation durch Interviews

Im Laufe des Projektjahres stellte ich den Schülern/innen zweimal Fragen, die sie per E-Mail beantworteten. Zu Beginn des Projektes stellte ich diese vier Fragen:

1. Was versteht man unter "Programmieren"?
2. Wo wird programmiert?
3. Kennst du eine "Programmiersprache"?
4. Aus Physik: Kennst du elektronische Bauteile?

Erwartungsgemäß waren beim ersten Beantworten kaum wesentliche Kenntnisse zu erfahren, da „Programmieren“ nur im Zusammenhang mit DVD-Recordern genannt wurde, und programmieren auch häufig mit „Installieren von Software“ verwechselt wurde.

Am Ende der Projektarbeit befragte ich die Schüler/innen abermals per E-Mail. Erwartungsgemäß waren jetzt die Antworten weitaus zielgerichteter und inhaltlich besser.

Ein paar Zitate will ich als Beispiel anführen:

Wir haben in IKT eine Ampel gebaut. Da haben wir viel über das Programmieren gelernt. Zum Beispiel dass es eine eigene Programmiersprache gibt und dass der Computer unsere Sprache umwandelt in Strom oder kein Strom. Wir haben auch etwas über die elektronischen Bauteile gelernt. Wie zum Beispiel die Halbleiter, die nur unter bestimmten Einwirkungen wie Wärme oder Licht Strom leiten. Auch haben wir über die Programmiersprachen gelernt. Welche es gibt und was diese bewirken. Eine davon ist auch FSM LOGO die wir hier in der Schule verwenden um unsere Ampel zu programmieren. Auch die Ampel haben wir in einer IKT-Stunde selber gebaut. Dort musste man genau und viel Löten. Wir haben auch viel programmieren geübt. Es gibt aber auch noch andere Programmiersprachen wie zum Beispiel C++ oder Pascal usw ...

Wir mussten zuerst einmal die ganzen Befehle für unsere Ampel lernen wie zum Beispiel „ausg 888 1“ was soviel bedeutet wie „schalte die erste Lampe ein“. Das war immer alles sehr interessant und hat sehr viel Spaß gemacht. (Sophia)

Die Schüler der HS Imst-Oberstadt 3b nahmen am Projekt IMST teil. Das Ziel ist am Ende des Schuljahres eine automatisch funktionierende Ampelschaltung zu bauen. Sie lernten dabei die wichtigsten Teile der Elektronik in Physik kennen und deren Wirkungen. Die Schüler lernten in IKT die Programmiersprachen für eine Ampel kennen, um die Ampel zum Leuchten zu bringen.

Schritt für Schritt kamen sie dem Ziel näher. Als erstes versuchten die Schüler das Ganze mit einer einfachen Schaltung, um es ein wenig zu verstehen. In einer IKT Stunde bauten die fleißigen Schüler mit ihren KV eine richtig funktionierende Ampel, nur auf etwas kleiner Art und das ganze von oben gesehen. Einige Schüler hatten sogar den Mut, eine Ampel selber daheim zu bauen, die auch bei allen funktionierten!!! (Niklas)

Im Herbst fingen wir an, das Programmieren zu lernen, indem wir Ampeln programmierten. Wir programmierten die Ampeln im Logo. Es gibt auch C++, Php, Java In einer Ampel sind verschiedene Bauteile vorhanden. Wie zum Beispiel LED's, Widerstände und Drähte. Unser Lehrer hat uns gezeigt wie man eine Ampel baut und einige Schüler haben selber eine gebaut. Man braucht folgendes: Lötkolben, Lötzinn, Kartonplatte, Drähte, Parallelstecker und Kabel. (Florian)

An der HS Imst-Oberstadt nahm die Klasse 3b mit dem Klassenvorstand Gerhard Lechner an einem Projekt teil; in dem sie bis zum Ende des Schuljahres eine vollautomatische Ampel bauen. In IKT lernten sie das Programmieren mit der Programmiersprache LOGO. Sie lernten aber auch andere Sprachen kennen z.B: Java, C++, Pascal u.s.w....;Zuerst mussten sie einmal das Programmieren lernen, danach bauten sie 7 Ampelanlagen selbst im Unterricht. Sie lernten aber auch Roboter zu programmieren, was sie am Tag der offenen Tür darstellen durften. Einige Schüler der 3b waren fleißig und machten in den Semesterferien Ampelanlagen. Die Klasse freut sich, dass wir an diesem Projekt teilnehmen. (Lukas)

4.3 Erreichte Ziele

Wenn ich abschließend und rückblickend den Projektverlauf dahingehend analysiere, ob die geplanten Ziele auch erreicht wurden, so kann ich durchaus auf einen erfolgreichen Ablauf zurückblicken.

Ich konnte den Schülern über die lehrplanmäßig geforderten Lehrinhalte hinaus Grundlagen der Elektronik vermitteln. Speziell der technikgeschichtliche Ablauf der Entwicklung der Rechenmaschinen (vom mechanischen Rechner, über den Relaisrechner, dem „Mailüfterl“ bis zum integrierten Schaltkreis) gab den Schülern einen Einblick, dass Technik nichts „Starres und Festgeschriebenes“ ist, sondern etwas „Fließendes und sich Veränderndes“.

Das Programmieren stellte eine willkommene und interessante Abwechslung zum „normalen“ IKT-Unterricht dar (ECDL, Office). Die Schüler/innen bekamen einen Einblick, wie Programme „gemacht“ werden und konnten sich auch selbst als Programmierer versuchen. Dabei stellte sich heraus, dass das Programmieren von anspruchsvolleren Aufgaben, wie Rekursionen oder das Zusammenspiel von mehreren Ampeln, eine große geistige Herausforderung darstellt.

Das Herstellen der Ampelschaltung mit Dioden, Widerständen, Drähten und Lötkolben erforderte eine handwerklich-praktische Geschicklichkeit, wobei vor allem einige Mädchen beweisen wollten, dass Löten nicht nur „Bubenarbeit“ sein sollte.

Den Informatik-Kollegen gab das Projekt sicher wertvolle Impulse und Anregungen, zukünftig einen Teil der Lehrinhalte für etwas „Office-fremdes“ zu verwenden.

5 DISKUSSION/INTERPRETATION/AUSBLICK

Abschließend noch einen Blick in die Zukunft oder eine Antwort auf die Frage: wie geht es weiter?

Es ist geplant, dieses Projekt auf verschiedene Arten der (Schul)-öffentlichkeit zu präsentieren:

- ein Bericht für die Schulhomepage wird erstellt
- ein Beitrag für den (gedruckten) Jahresbericht wird erstellt
- die Schüler werden in den letzten Schulwochen einigen IKT-Gruppen und den jeweiligen Lehrern/innen ihre Arbeit vorstellen

Persönlich beschäftigen mich noch zwei Gedanken:

Einerseits überlege ich Möglichkeiten, die erstellten Ampelschaltungen und auch Unterlagen (Grundlagen der Elektronik, Einführung in das Programmieren mit LOGO) den IKT-Kollegen und den nachfolgenden Schülergruppen als Unterrichtsmaterial zur Verfügung zu stellen.

Andererseits beschäftigt mich der Gedanke, dieses Projekt in einem IMST-Folgeprojekt weiterzuentwickeln. Es geht darum, mit Hilfe einer LED-Schaltung, einen Einstieg in das Programmieren in der Sekundarstufe I zu erreichen. Denkbar wäre, dass die Schüler/innen einige LED-Schaltungen herstellen, gemeinsam dann einen „Programmierkurs“ entwickeln und dieses Gesamtpaket dann anderen, interessierten Schulen zur Verfügung stellen.

6 LITERATUR

KLAUDT, D.: MSWLogo (Logo für Windows), Institut für Mathematik und Informatik - Pädagogische Hochschule Ludwigsburg, Okt. 1999

MEYER, H.: Was ist guter Unterricht?, Berlin, Cornelsen Verlag

SCHEIBER, E.: Bausteine guten naturwissenschaftlichen Unterrichts, Wien, IMST-Publikation

bm:ukk: Neuer Lehrplan der HS, Bundesgesetzblatt II Nr. 134/2000