

EPIC

Electronic Programmable Intelligent Calculator

Michael Friesenhengst, Florian Weinzerl

22. März 2017

Inhaltsverzeichnis

1	Einleitung	3
1.1	Gedanken zur Umsetzung	3
2	Hardware	5
3	Software	6
3.1	GUI - Das Graphical User Interface	6
3.2	Das Mathematik-Paket	6
4	Marketing	7

1 Einleitung

Bereits als wir uns erstmalig mit dem Thema, eine Diplomarbeit zu schreiben, auseinandersetzen, war es eindeutig, dass sie nicht bloß sinnhaft und innovativ, sondern darüber hinaus noch eine Herausforderung für uns sein müsste. Ein solches Projekt zu finden, stellte sich als nicht einfach heraus, bis schlussendlich ein Problem aus dem Schulunterricht auftrat. Das Wort "Problem" soll hier weniger für eine Störung stehen, wie es in unserem Sprachgebrauch oft verwendet wird, viel mehr ist damit "Problemstellung" gemeint.

In unseren jeweiligen Schulkarrieren wurde denkbar auf Papier begonnen und allmählich würde ein Fach nach dem anderen auf digitale Mitschrift durch Laptop oder ähnliches umsteigen. Heute und vermutlich noch sehr lange Zeit besteht dies aus einer Mischform: Wer bestmöglich im Unterricht dabei sein will, sollte Texte notieren, möglicherweise auf Dateien aus dem Internet zugreifen, Freihandzeichnungen von der Tafel auf die eigenen Zettel übertragen und die Lösung des einen oder anderen Gleichungssystems nachvollziehbar dokumentieren können und wenn möglich alles so speichern respektive hinterlegen, dass es einfach wieder auffindbar ist. Unserer Meinung nach, sind das zwar Aufgaben, die bewältigbar sind, aber wenn es möglich wäre, all diese Aufgabenbereiche in ein Gerät zu kombinieren, das demnach im Unterricht, in Vorlesungen, aber auch in manchen technischen Berufen genutzt werden kann, so ist das Ergebnis nicht nur eine Entlastung, sondern schließt womöglich auch eine Marktlücke.

Und genau das haben wir uns zur Aufgabe gemacht. Wenn Sie als Leser eine kurze Werbung erlauben, hier ist die fertig verfasste Aufgabenstellung an uns selbst während der Arbeit:

EPIC ist ein fortschrittlicher Taschenrechner in Form eines Tablets, das mit einem Stylus bedient werden kann. EPIC kann sowohl Berechnungen durchführen als auch Funktionsgraphen erstellen sowie Gleichungen lösen. Er soll die Mitschrift händisch sowie am Computer ersetzen und somit alles in einem Gerät vereinen. Das gesamte System befindet sich auf einer austauschbaren SD-Karte, wodurch einfache Software-Updates bzw. Resets durchgeführt werden können. Der langlebige Akku garantiert Arbeiten ohne Unterbrechung.

Dinge, die uns außerdem wichtig sind, umfassen unter anderem Modularität und das korrekte Marketing, was wir uns anschließend auch zur Aufgabe machten.

1.1 Gedanken zur Umsetzung

Nun stellt sich also die Frage: Wie wollen wir all das bewerkstelligen? Die hardwaretechnische Einschränkung auf ein Tablet erleichtert die Entscheidung schon um Einiges. Im Hinblick darauf fassten wir den Entschluss, ein **...Friesi pls...**

Bezüglich der Software konnte wohl nicht viel aus fremden Komponenten verwendet werden, vor

allem, da das dem Sinn dieser Arbeit widersprechen würde. Wir waren also selbst dafür zuständig und das bedeutete, sowohl die Benutzer-Oberfläche als auch die Schreib- und Mathematik-Einheit waren so zu realisieren, dass auch nach Abschluss und Veröffentlichung Entwickler auf dem Code aufbauen können. Das hat zusätzlich den taktischen Vorteil, dass Benutzer bei Bedarf auch schnell eigene Funktionen hinzufügen und diese dann für andere zugänglich machen können. Diese Funktionalitäten sind zwar von den üblichen Taschenrechnern bekannt, allerdings nicht in einer plattformunabhängigen Sprache, die darüber hinaus objektorientiertes Programmieren ermöglicht, wie Java. Potenzielle Kunden hätten also den Vorteil, dass sie selbst darüber verfügen können, welche Software-Pakete sie neben den Grundfunktionalitäten integriert haben möchten. Das bezieht sich auch darauf, dass sie vom Programmier talent Anderer profitieren können, falls eine eher ausgefallene oder fachspezifische Funktion per Softwaremodul von einer Privatperson ins Internet hoch geladen wurde.

Betreffend der Oberfläche ist sicher ein Menü am oberen Fensterrand sinnvoll. Dieses sollte Menüpunkte in Kategorien einteilen, und dem Benutzer die häufigst verwendeten Funktionen schnell zugreifbar machen. Dies sind möglicherweise keine überraschenden Festlegungen, aber zu irgendeinem Zeitpunkt müssen Design-Entscheidungen getroffen werden. Außerdem weist nicht jedes Menü diese Flexibilität auf, ob sie sich bewährt wird man sehen, und diese Eigenschaft kann eventuell noch nachjustiert werden. Darüber, was die Mathematik-Einheit zu bieten haben sollte, lässt sich wohl kaum streiten. Die Vorgaben dafür beschränken sich, bis auf, dass sie funktionsfähig sein soll, auf Null.

Dafür ist das Marketing eine äußerst interessante, um nicht zu sagen eine schwungvolle Angelegenheit. Dabei wollen wir uns in unsere potenziellen Kunden hinein versetzen und erforschen, warum sie unser Produkt kaufen, wie der Entscheidungsprozess stattfindet und in wie fern wir uns das für unseren Imageaufbau und Vermarktung des Produkts zu Nutzen machen können.

2 Hardware

Von Anfang an war klar, dass die Hauptrecheneinheit unseres Prototypen fähig sein musste, den Rechenaufwand unserer Software problemlos zu bewältigen. Ein weiteres wichtiges Kriterium war, einen kleinen Formfaktor einhalten zu können, um zu garantieren, dass unsere Einheit in einem handlichen Gehäuse Platz findet. Beiden Punkten zugleich wird lediglich ein *single-board-computer* gerecht, er vereint hohe Rechenleistung mit minimalem Platzaufkommen.

Das *Display* ist dafür konzipiert dem Benutzer ... Mittels dem *resistiven Touchscreen* ist es möglich, Berechnungen einzugeben sowie Mitschriften zu tätigen. Er wird mithilfe eines *Stylus* bedient.

Da der *single-board-computer* und *Display* einen relativ hohen Energiebedarf aufweisen, musste eine Hardware-Komponente her, welche beide je nach Bedarf aus-und-einschalten kann. Bei dieser Komponente war darauf zu achten, dass diese dem Gerät so wenig Energie wie möglich entzieht um eine lange *Standby-Zeit* zu garantieren. Die Komponente kann auf Knopfdruck *Display* und Recheneinheit getrennt von einander stromlos machen.

3 Software

Besonders, wenn es um große Projekte wie dieses geht, ist Software tatsächlich kein einmaliges Produkt, sondern etwas, was sich über den Lauf dessen und darüber hinaus verändert. Aus diesem Grund ist es unserer Meinung nach äußerst wichtig, nur Lösungen zu publizieren, die zukunftstauglich und offen für Erweiterungen sind. Im folgenden Kapitel möchten wir darauf eingehen, wie diese Denkweise in unserer Arbeit umgesetzt wurde und warum davon auch langfristig profitiert werden kann.

Als wir begannen, uns über die Software-Strukturen Gedanken zu machen, war es klar, dass es eine Komponente bräuchte, die sich mit der Benutzer-Interaktion beschäftigt, und eine, in der mathematische Prozesse durchlaufen, aber auch abgebildet werden sollten. Es musste also abgewogen werden, ob besser eines eine vorrangige Stellung haben sollte, oder ob im Endeffekt zwei Systeme, die sich komplementieren, am Laufen sein sollten. An den Beispielen existierender Software erkannten wir, dass zwei parallele Systeme, um ideal zu funktionieren, andauernd in Synchronisation gehalten werden müssten. Um diesen doppelten Aufwand zu umgehen, entschieden wir, das Mathematik-Paket in die grafische Oberfläche zu integrieren, da der mathematische Teil eben nur eine der Funktionalitäten unseres Produkts ist, wenn auch eine sehr wichtige. Das Grundgerüst stellt also das *Graphical User Interface* oder kurz *GUI* dar.

3.1 GUI - Das Graphical User Interface

3.2 Das Mathematik-Paket

Mathematik-Paket ist der interne Name für diejenige Software Komponente, die alles in Bezug auf mathematische Vorgänge verarbeitet. Aus dem Versuch, diese Komponente in Programmcode zu fassen, erlangte ich eine wesentliche Erkenntnis. In der Mathematik sieht man Konstrukte als abstrakt, soweit sollte die mathematische Sicht der Dinge ja bekannt sein. Durch Definitionen schafft sich ein Mathematiker Freiraum, wodurch auch der Level der Abstraktion steigt. Zum Beispiel ist uns möglicherweise gar nicht mehr bewusst, dass wir beim 'mit x multiplizieren', tatsächlich x Mal zusammenzählen. Beim 'hoch x rechnen' wird eigentlich x Mal multipliziert, also wie oft addiert? Natürlich stellen die Grundrechenarten programmtechnisch kein Problem dar, jedoch ist bereits diese einfache Frage ohne mehr Information nicht zu beantworten, was zeigen soll, wie tief wir schon drin stecken, ohne jemals angefangen zu haben.

4 Marketing