

Fazit zum PIC Simulator:

Zu Beginn des Projektes war uns nicht klar, wie wir dieses beginnen oder fertigstellen sollten. Es war alles sehr neu und zu komplex um direkt alles zu verstehen. Nachdem wir das Projekt in kleine Aufgaben eingeteilt hatten und einige von diesen dann auch schnell erledigen konnten, sah es machbarer aus. Das größte Problem war unser Unwissen, wie der PIC funktioniert. Somit konnte nicht wirklich exakt geplant werden. Was bekannt war wurde entwickelt und falls neue Erkenntnisse einkamen wurden oft alte Implementierungen verworfen oder mussten stark abgeändert werden. Es war ein ständiges lernen und anpassen in während der Laufzeit des Projektes. Java war keine Erschwernis für uns, nur hatten wir beide noch nie ein GUI in Java erstellt und mit Hilfe von Swing und dem „Intellij-WindowBuilder“ haben wir es zwar geschafft ein funktionsfähiges GUI zu bauen, aber haben wir dort die „best practice“ und gedachte Anwendung nicht eingehalten. Dies hat zu Beginn aber den Fortschritt sehr zurückgehalten. Zuerst wollten wir den Simulator 1:1 wie das Original realisieren. Dies stellte sich aber als ineffizient dar. Deswegen fingen wir frisch an. Dort wurden dann viele Funktionen des PICs abstrahiert um diese in Java um einiges schneller realisieren zu können. Das Fehlen einer genaueren Planung war ein Segen und Fluch zugleich. Eine Grobplanung welche festlegte, wie im Allgemeinen alles funktionieren sollte, was für Klassen es gab und wie die GUI aussehen sollte hatten wir uns erstellt. Genauere Spezifikationen über Methoden und Attribute hatten wir nicht abgemacht. Dadurch konnten wir bei neuen Erkenntnissen oder Ideen flexibel alles anpassen. Die generelle Koordination war aber sehr erschwert, was bei einem kleinen Team nicht so schlimm wie bei Größeren ist. Nachdem die Grundfunktionen des Simulators implementiert waren, lief der restliche Entwicklungsprozess relativ schnell und leicht ab. Bei dem Verständnis des Timers und WatchDog hatten wir zu Beginn Schwierigkeiten, welche aber durch exaktes Recherchieren in dem Datenblattes des Herstellers, gelöst wurde. Vieles wurde später dennoch umgeschrieben oder gelöscht, da durch neue Ideen alte Funktionen obsolet wurden. Einiges konnte aber nicht angepasst werden, da es schon zu tief in der Implementation verankert war und der Aufwand alles anzupassen nicht Wert war. Hier zum Beispiel kommt der RAM, welcher als 128 langes 2d-Array verwirklicht wurde, um die Banken einfacher zu erkennen und zu simulieren. An vielen Stellen mussten aber die Zwei Dimensionen des Arrays hintereinander geknüpft werden um korrektes Verhalten zu erlangen.

Wenn wir das Projekt erneut realisieren würden, würden wir alles zu Beginn genauer Planen und festlegen. Da wir dann auch schon die korrekte Funktion des PICs kennen, wäre dies auch ohne Probleme möglich.