

## Am Anfang war das Chaos.

Hoffmann, TGM/NT/3ANK90

Zuerst haben wir einen Zeitplan erstellt, obwohl wir nur in sehr groben Zügen erfahren haben, was überhaupt zu realisieren war: eine Erweiterungsplatine für den  $\mu$ PROFI (ein 8051-Entwicklungssystem), dessen Funktion es sein sollte, nach acht Schalterfunktionen acht LEDs als Lauflicht oder blinkend zu schalten.

Obwohl der 8051 als Microcontroller I/O-Einheiten zur Verfügung stellt, bauten wir unsere doch selber auf (der  $\mu$ PROFI unterstützt den Port-betrieb nicht) und betrieben die CPU extern als normalen Prozessor. Die I/O-Einheit für Schalter und LEDs war unsere Schaltung, bestehend aus einem Gatterfriedhof zur Adreßdecodierung und diversen Latches und Puffer für Lese- und Schreibfunktionen auf dem Datenbus.

Die Schaltung wurde mit PCAD gezeichnet und nach einer Wirelist (Verdrahtungsliste) gefädelt. Das Fädeln, freilich, entpuppte sich nicht nur als eine mühsame, sondern auch als fehlerträchtige Arbeit, zumal wir diese Art des Aufbaues erst neu kennenlernten. Auf das Fädeln folgte die Fehlersuche, denn anfangs pflegte unser Drahtverhau über den  $\mu$ PROFI den PC (dieser bietet dem User einen FullscreenDebugger als Oberfläche zwecks Bedienung und Programmierung) regelmäßig zum Absturz zu bringen.

Auch PCAD bereitete uns Kopfzerbrechen, denn aufgrund unserer mangelnden Kenntnisse war es oft mühsam unsere Vorstellungen einer technischen Zeichnung durchzusetzen. Auch Computer haben ein Eigenleben! Tätigkeiten wie Routen oder Erstellen einer Stückliste entwickelte sich zu einer elektronischen Weltreise und wir kamen erst spät hinter alle Kniffe.

Angesichts der vielen Einzelprogramme von PCAD, die man nach Möglichkeit in der richtigen Reihenfolge aufrufen und auch bedienen können sollte, kam ich mir vor, wie Robin Hood im Datenwald. Letztendlich ist eine noch so schöne softwaregesteuerte Schaltung ohne Software völlig sinnlos. So bestand unsere letzte Hürde darin, auch programmtechnisch unserer Aufgabenstellung gerecht zu werden, wenn eine funktionstüchtige Schaltung vorausgesetzt wird.

Da Blinken oder Lauflicht nicht mit Lichtgeschwindigkeit ablaufen sollten, (man sieht ja nichts!) war es nötig, den internen Timer des 8051 für Verzögerungszwecke heranzuziehen.

Am Ende: Eine nette Bastelarbeit, die den Schülern lehren sollte, die oft schwierige Aufgabe zu meistern, Hard- und Software unter einen Hut zu bringen, natürlich auch bei Fehlfunktionen des Systems bei der Fehlersuche taktisch klug vorzugehen und mit Hilfe der Laborausrüstung und vieler Kniffe, die wir kennenlernten, am Ende Sieger zu bleiben. Hoffmann 3ANK

## HARDWARE-ERWEITERUNG zum $\mu$ PROFI-51

Hoffmann/Scheiber/Vozdecky, TGM/NT/3ANK90

TGM-DSK-175, TGM-LIT-013

1. Programm FTKL im Schuljahr 1990/91:

### SYSTEMZIELE

Die im Wesentlichen geforderten Leistungen des 1. FTKL-Programms liegen im Bereich der Mikroprozessor-Steuerungen. Es soll der Schüler dadurch lernen, einen Mikroprozessor zu verstehen, zu programmieren und eine entsprechende Hardware zu erstellen. Vor Beginn der Arbeit ist ein Projektplan über Gruppeneinteilung, Arbeitszeit und Abgabetermin zu erstellen. Dadurch steckt sich der Schüler selbst ein Ziel und lernt, eine Arbeit zu organisieren. Bei der Entwicklung wird das Nachschlagen in Datenbüchern und Prospekten notwendig, um für spezielle Anwendungen entsprechende Bauteile zu finden. Es wird dadurch die Kreativität und Eigenständigkeit jedes Einzelnen gefördert. Es werden weiters die Grundfunktionen der maschinennahen Programmierung (Assembler) und, in Zusammenhang damit, eine Verbindung zur praktischen Anwendung demonstriert. Zusätzlich wird auch auf die praktische Ausarbeitung Wert gelegt. Das heißt, der Schüler muß sich über verschiedene Fertigungskriterien Gedanken machen (geringe Kosten, einfache Bedienung, Möglichkeiten der Fertigung). Es wird eine vollständige Entwicklung eines Produktes durchgemacht (von Entwicklung über Fertigung zur Wartung und Austestung).

### ANWENDERANFORDERUNGEN

Das komplette System besteht im wesentlichen aus drei Teilen:

- der PC (Personal Computer): er wird verwendet, den  $\mu$ PROFI zu initialisieren und diesen mit den für seine Funktion notwendigen Daten zu versorgen.
- der  $\mu$ PROFI: der  $\mu$ PROFI ist ein kleines, im Prinzip unabhängiges Mikroprozessor-System. In Kombination mit einer entsprechenden Zusatzplatine können kleine Demonstrationen vollzogen werden.
- die Hardware-Erweiterungsplatine: sie stellt für den  $\mu$ PROFI ein geeignetes Ein-/Ausgabegerät dar und kann nur in Zusammenhang damit funktionieren.

### Kurzbeschreibung

In Abhängigkeit der Schalterstellung (immer nur 1 Schalter geschlossen) soll eine entsprechende Zahl auf der 7-Segmentanzeige ausgegeben, und verschiedene Muster an den 8 LED's dargestellt werden:

Schalter 1 geschlossen: es sollten bei richtiger Bedienung alle 8 LED's leuchten und eine "1" an der 7-Segmentanzeige anliegen.

Schalter 2 geschlossen: es entsteht ein langsames Lauflicht und eine "2" am Display.

Schalter 3 geschlossen: diese Schalterkombination ergibt ein schnelles Lauflicht und eine "3" auf der Anzeige.

Schalter 4 geschlossen: es entsteht ein Muster, wo abwechselnd, einmal die "geraden", einmal die "ungeraden" LED's leuchten. Am 7-Segment erscheint eine "4".

Schalter 5 geschlossen: es ergibt sich dabei ein Lauflicht, daß einmal von links nach rechts, und dann von rechts nach links läuft. Die 7-Segmentanzeige bringt die Zahl "5".

Schalter 6 geschlossen: es entsteht ein Muster, wo die Lichter von außen nach innen laufend. Weiters wird eine "6" gezeigt.

Schalter 7 geschlossen: es kommt dabei zu einem Lauflicht, bei dem von rechts beginnend immer eine LED hinzukommt, und einer "7" an der Segment-Anzeige.

Schalter 8 geschlossen: An der 7-Segmentanzeige wird eine "8" ausgegeben. An den LED's ergibt sich ein inverses Lauflicht.

Fehlbedienung: Der  $\mu$ PROFI ist allerdings so programmiert, daß er nur bei Betätigung eines Schalters die entsprechende Ausgabe tätigt. Bei Fehlbetätigung - d.h. bei Betätigen von zwei oder mehreren Schaltern - wird diese durch einen "F" (Fehler) am 7-Segment angezeigt. Dieser Zustand hält solange an, bis wieder eine Taste bedienungsrichtig betätigt wird. Die LED-Reihe zeigt dabei in der Mitte 2 leuchtende Punkte.

## SYSTEMKURZBESCHREIBUNG

Das Projekt stellte die Funktion, Arbeitsweise, Entwicklung und Herstellung eines  $\mu$ P-Systems zur Aufgabe. Es wurde eine Platine, die als Ein-/Ausgabebaustein eines Mikrocontrollers (hier:  $\mu$ PROFI-51) dienen soll, dimensioniert und aufgebaut. Zur Erstellung des Schaltplanes, der Bauteilliste und des Layouts wurde ein Anwenderprogramm (PCAD-Personal Computer Aided Design) zur Verfügung gestellt. Es soll eine achtstellige Schalterkombination (8 Schalter am Print realisiert) vom  $\mu$ PROFI eingelesen werden, und abhängig davon ein Leuchtmuster auf 8 LED's bzw. ein Zeichen an einer 7-Segmentanzeige ausgegeben werden. Dabei wurde vorher über einen PC der  $\mu$ PROFI in Assembler programmiert (unter Verwendung eines Timers zur Verzögerung; in strukturierter Form programmiert, d.h. einzelne, zusammengehörende Teile, z.B. einzelne Schalterstellungen, in einem Unterprogramm realisieren). Die Durchführung des Projekts wurde nach einem festgelegten Zeitplan vollzogen, welcher sich über einen Zeitraum von mehreren Wochen erstreckte.

Kurzbeschreibung der Arbeitsschritte bei der Fertigung:

- Funktion und Arbeitsweise des  $\mu$ PROFI51 anhand von Datenblättern erfassen.
- Problem der Ansteuer- und Durchschaltlogik der zu entwickelten Platine durchdenken.
- mittels PCAD (= Anwenderprogramm) Schaltplan, Layout und sämtliche zur Fertigung wichtigen Listen (cross-referenz-list, net-list) erstellen. Dabei muß allerdings auf

vorhandene und zur Verfügung gestellte Bauteile Rücksicht genommen werden.

- Softwareprogramme entwickeln und programmieren. Als Programmiersprache wird ASSEMBLER verwendet.
- Fertigung der entworfenen Platine und anschließende Kopplung mit dem  $\mu$ PROFI.
- Tests, Optimierung, Wartung

## HARDWAREKONFIGURATION

- a) digitale IC's:
  - 1 x 74LS32 (= 4 OR-Gatter)
  - 1 x 74LS02 (= 4 NOR-Gatter)
  - 1 x 74LS00 (= 4 NAND-Gatter)
  - 2 x 74LS273 (= Latch)
  - 3 x 74LS244 (= LED-Treiber, Puffer)
- b) diskrete Bauteile:
  - 16 x Widerstände à 330E (= Vorwiderstände)
  - 8 x Widerstände à 47k (= Pull-Up-Widerstände)
  - 8 x Tantal-Kondensatoren à 1 $\mu$ F (= Stützkondensatoren)
- c) Elektromechanisches:
  - 1 x 64-polige Steckerleiste (= Schnittstelle)
  - 8 x Umschalter
  - 1 x 7-Segmentanzeige
  - 8 x LED's
- d) Zusätzliches Material:
  - 8 x Sockel für IC's
  - 1 x Lochraster-Platine (100x160mm)
  - dünnere, lackisoliertes Draht zur Verdrahtung des Aufbaues

## FUNKTIONALE BESCHREIBUNG

*Anm.d.Red.: Aus Platzgründen wird hier auf die Wiedergabe der Schaltpläne, Listings und Struktogramme verzichtet. Interessenten können diese über TGM-LIT-013 anfordern. (Bestellkarte am Ende dieses Heftes).*

### Software

Bei Entwurf der Software waren einige Gesichtspunkte zu berücksichtigen. So war es zum Beispiel vorgeschrieben, die Software strukturiert zu gestalten, d.h. es sollten einzelne Programmblöcke zu erkennen sein. Weiters soll ein Timer zur Verzögerung der Sendeimpulse verwendet werden. (Eine Verzögerung ist unbedingt notwendig, da z.B. ein Lauflicht nie als solches zu sehen wäre. Der interne Takt, mit dem normalerweise geschrieben wird, beträgt 12 MHz. Es würde also alle 12  $\mu$ s eine LED leuchten).

Zu Beginn muß der Datenzeiger auf einen bestimmten Anfangswert gelegt werden. Es wurde hier der Wert A000h gewählt und mit dem Befehl ORG fix definiert. Das Hauptprogramm besteht im Wesentlichen aus 3 Teilen:

### EINLESEN

Dieser Programmteil löscht zuerst den Akkumulator-Inhalt, setzt den Datenzeiger auf die gewählte Leseadresse (C004h), um die anliegenden Bitmuster lesen zu können, und gibt dieses in binärer Form in den Akkumulator.

### EINSZÄHL

Hier wird die Anzahl der im eingeholten Bitmuster vorhandenen Einsen gezählt. Dies ist insofern wichtig, um mögliche Fehlbedienungen, wie zum Beispiel die Befätigung mehrerer Schalter gleichzeitig, zu erkennen und entsprechende Maßnahmen zu setzen. Am Ende dieses Teiles steht im Register R4 die Anzahl der "1" im Bitmuster.

## VERZWEIG

Die Aufgabe dieses Programmteils liegt darin, entsprechend der anliegenden Bitmuster Verzweigungen zu kleinen Unterprogrammen (Jobs -> strukturierte Programmierung) durchzuführen und Fehler zu erkennen und ebenfalls entsprechend zu reagieren. Eine Fehlbedienung liegt immer dann vor, wenn die Anzahl der Einsen ungleich 1 ist. Dies ist also hier in 2 Fällen möglich. Entweder es wird überhaupt kein Schalter betätigt (R4=0) oder mehr als einer (R4 > 1). Mit einer doppelten Abfrage wird unterschieden:

Bei richtiger Bedienung kommt es zur bitweisen Abfrage (JNB-Befehl: es wird nur dann auf die angegebene Adresse gesprungen, wenn das angewählte Bit nicht gesetzt ist. LJMP-Befehl: es wird ein unbedingter Sprung über 64k durchgeführt) und entsprechender Verzweigung zu den Jobs. (kleine Arbeitsschritte)

## Fehlersuche

Obwohl zuvor alle Leitungen mit dem Ohmmeter auf ihre elektrische Verbindung überprüft wurden, funtionierte die Schaltung nicht. Der Computer stürzte immer wieder ab. Nach langwierigem Durchmessen der Leitungen der Steckerleiste gegeneinander, wurde festgestellt, daß 4 Stütz-Kondensatoren falsch gepolt eingebaut worden sind. Weiters war zu erkennen, daß der Datenbus nicht vollständig funktionsfähig war (Datenleitungen, Querverbindungen). Es lagen nur 2,5V an, was auf offene Leitungen schließen läßt. Nach Behebung dieser Fehler funktionierte bereits die Ausgabe. Jedoch war das Einlesen nicht möglich. Als Ursache ergab sich, das ein ODER-Gatter-IC defekt war (undefinierte Zustände am Ausgang; Ständig 1,8V). Da der Fehler dadurch nicht behoben werden

konnte, versuchten wir, die Schaltung nach eventuell falschen Verbindungen abzusuchen (nach Verdrahtungsliste).

Es muß ein Fehler bei der PCAD-Programmierung unterlaufen sein. Statt einem ODER-Glied wurde versehentlich ein NAND-Glied eingelötet. Das Testprogramm wurde gestartet. Diesmal funktionierte die Ausgabe auf die 7-Segmentanzeige nicht. Der Grund dafür lag in der Software. Es wurden im Programm die beiden Adressen C003h und C004h verwechselt.

Nun war das Projekt voll funktionstüchtig.

## ZEITPLAN

|   |          |
|---|----------|
| Schaltungsentwurf:                              | 12 Std.  |
| Schaltungsdimensionierung (Bauteilwahl):        | 6 Std.   |
| PCAD  |          |
| - Schaltungsentwurf, Verbesserung, Optimierung: | 30 Std.  |
| - Zuweisung (Cross-Reference-List)              | 4 Std.   |
| - Erstellung sonstiger Listen                   | 4 Std.   |
| - Plazieren der Bauteile (PCPLACE)              | 10 Std.  |
| - Vorbereitung, Entflechtung in PCROUTE         | 3 Std.   |
| - Ausdruck der Pläne, Listen                    | 2 Std.   |
| Printerstellung                                 |          |
| - Plazieren der Bauteile                        | 4 Std.   |
| - Verdrahtung nach Verdrahtungsliste            | 15 Std.  |
| - Inbetriebnahme, Fehlersuche, Optimierung      | 15 Std.  |
| Software  |          |
| - Wissensaneignung (FSD51,ASM51,8051..)         | 40 Std.  |
| - Überlegung, Entwicklung, Erstellung           | 30 Std.  |
| - Tests in Verbindung mit Hardware              | 10 Std.  |
| Dokumentation, Wartung, Tests, Verbesserungen   | 30 Std.  |
|   | -----    |
|   | 215 Std. |
|   | =====    |

Bei diesen Stundenangaben sind nicht nur die verbrachten FTKL-Unterrichtsstunden, sondern auch die im aktuellen Fachgebiet GLP und die privat aufgewendete Arbeitszeit miteinberechnet.

