



Digitale Funkuhr

Dokumentation

Projekt: Was ist Zeit?

F. Böhler, A. Fischer, D. Gut

Betreuung

B. Piller & G. Kopacsy

Baden – November 2002

1	Zusammenfassung.....	4
2	Einleitung	5
2.1	Das Produkt	5
2.1.1	Funktionen	5
2.1.2	Qualitätsmerkmalen	5
2.2	Teamregeln.....	5
2.2.1	Regel 1	6
2.2.2	Regel 2	6
2.2.3	Regel 3	6
2.3	Fächerübergreifen.....	6
2.4	Geplante Hilfsmittel.....	6
3	Hauptteil.....	7
3.1	Hardwareentwicklung.....	7
3.1.1	Übersicht nach dem EVA-Prinzip.....	7
3.1.2	Uhrlogik	7
3.1.3	Funksignalempfang	8
3.1.4	Anzeige.....	8
3.1.5	Das Schema	9
3.2	Herstellung.....	10
3.2.1	Ätzen	10
3.2.2	Zusammenbau.....	12
3.3	Signal / Zeitübertragungsdienste	13
3.3.1	Zeitzeichensender DCF77	13
3.3.2	Was wird gesendet?	13
3.3.3	Kodierung des Zeitsignals	13
3.4	Software Computer (SyncTime).....	15
3.4.1	Funktion.....	15
3.4.2	Programmiersprache	16
3.4.3	Kommunikation zwischen dem Computer und der Uhr.....	17
4	Schlussteil/Reflexion	18
4.1	Hardwareentwicklung.....	18
4.2	Software Mikrokontroller	18
4.3	Software Computer (SyncTime).....	18
4.4	Reflexion zum gesamten Projekt	19

5	Verdankung.....	20
	Anhang A: Installationsanleitung der Uhr	
	Anhang B: Herstellungsanleitung der Uhr	
	Anhang C: Flussdiagramm AT90S4433	
	Anhang D: Quellcode AT90S4433	
	Anhang E: Flussdiagramm PC-Software	
	Anhang F: Quellcode PC-Software	
	Anhang G: Arbeitsjournal	
	Anhang H: Selbsteinschätzung	
	Anhang I: Projektbeurteilung durch die Lehrer	

1 Zusammenfassung

Im Vordergrund dieses Projekts steht das Produkt, welches eine digitale Funkuhr ist. Das Produkt wird anhand dieser Dokumentation und der Anhänge dementsprechend auch vorgestellt. Die Zielsetzung dieses Projekts war, neben eigentlichen Produkt, die Erweiterung des Wissen im Bereich Elektronik und Informatik, sowie das Sammeln von Erfahrung in Bezug auf Projekt- und Teamarbeit.

Das Projekt wurde mit Hilfe aller Gruppenmitglieder realisiert, wobei die einzelnen Arbeiten so aufgeteilt wurden, dass man möglichst effektiv arbeiten konnte. Die jeweilige Planung geschah gemeinsam in der Gruppe, so dass jedes Gruppenmitglied seine Meinung und Begründung hervorbringen, aber auch allen Mitgliedern die neuen Ideen und Probleme erläutern konnte. Zusätzlich wurden drei Teamregeln eingeführt, welche während des ganzen Projekts ihre Gültigkeiten hatten und somit auch eingehalten werden mussten. Wir sind mit unseren Resultaten aus dem Projekt zufrieden und haben unser Ziel erreicht. Das Produkt erfüllt die Zielsetzungen und die Erfahrungen, sowie das Wissen, wurde auch erweitert. Aus dieser Sicht ein gelungenes Projekt.

Das Team

F. Böhler, A. Fischer, D. Gut

2 Einleitung

Das Projekt stand unter dem Titel: "Was ist Zeit?". Unsere Gruppe, die aus Interesse an der Idee eine Funkuhr zu bauen entstand, definierte das Endprodukt dieses Projekts wie unten beschrieben.

2.1 Das Produkt

Das Produkt ist eine digitale Funkuhr, welche am Schluss über die unten aufgelisteten Funktionen und Qualitätsmerkmalen verfügt. Eine Funkuhr ist eine Uhr, welche sich nach der Zeit, die über ein Funksignal auf einer bestimmten Frequenz gesendet wird, automatisch stellen kann.

Zudem soll es möglich sein die Zeit des Computers an die Zeit der Funkuhr abzugleichen, was man dann unter dem Begriff „Synchronisieren“ versteht. Die Voraussetzung ist ein Computer, welcher Microsoft Windows als Betriebssystem hat.

2.1.1 Funktionen

- Sie stellt sich automatisch über das offizielle Zeitsignal, das per Funk übertragen wird.
- Sie gibt die Zeit auf einer Anzeige aus.
- Sie wird über das normale Stromnetz (230V) gespeisen.
- Sie bietet die Möglichkeit, das Zeitsignal über den PC zu synchronisieren

2.1.2 Qualitätsmerkmalen

- Die Uhr funktioniert stabil und muss nicht manuell neu gestartet werden
- Die Uhr sieht anschaulich aus und macht nicht den Anschein eines Bastelobjekts

2.2 Teamregeln

Sie mussten, wie schon in der Zusammenfassung erwähnt, über die ganze Zeitspanne des Projekts eingehalten werden, folgende drei wurden ausgewählt und eingehalten:

2.2.1 Regel 1

Alle Behauptungen und alle zur Verteidigung eine Behauptung herangezogenen Aussagen müssen begründet werden.

2.2.2 Regel 2

Jeder Teilnehmer muss bereit sein, alle seine für die Begründungen wichtigen Überzeugungen – wie sehr er auch an ihnen hängen mag- überprüfen zu lassen und ggf. aufzugeben.

2.2.3 Regel 3

Wenn eine Argumentation nach bestem Wissen aller Beteiligten zu einem begründeten Ergebnis gelangt ist, sollte geprüft werden, ob jedermann diesem Ergebnis zustimmen kann.

2.3 *Fächerübergreifen*

Das Projekt greift auf die Fächer wie Informatik, Elektrotechnik und Mathematik über, wobei sich Mathematik auf die Logik bezieht.

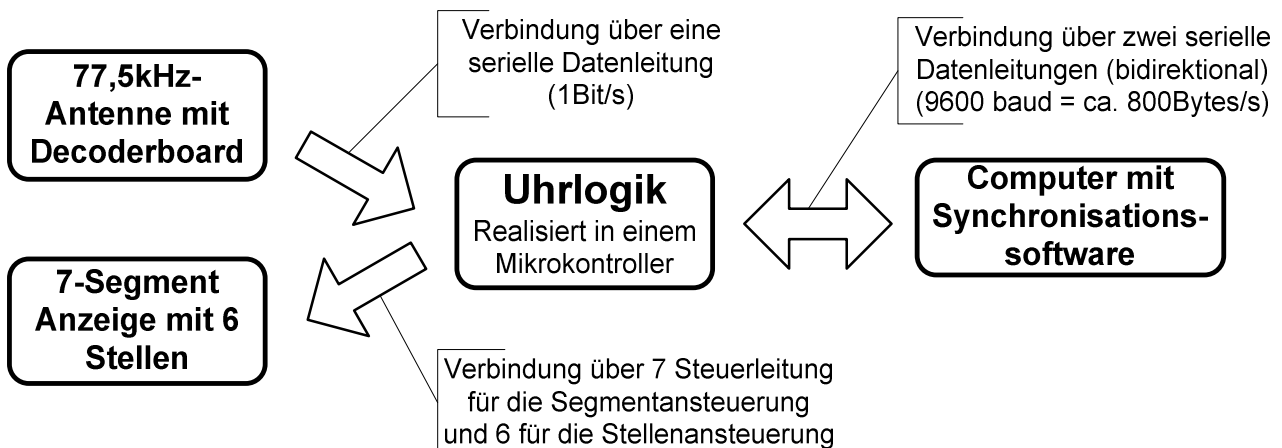
2.4 *Geplante Hilfsmittel*

Das Internet spielt bei der Informationsbeschaffung eine grosse Rolle, weil es die besten Möglichkeiten bietet, schnell an Informationen zu gelangen. Bei Bedarf werden Fachpersonen zu Rate gezogen.

3 Hauptteil

3.1 Hardwareentwicklung

3.1.1 Übersicht nach dem EVA-Prinzip



Die erste Überlegung bei der Herstellung war die Gesamtfunktion der Digitaluhr:

- Uhrlogik, realisiert in einem Mikrocontroller
- Funksignalantenne mit Decoderboard, verbunden über eine serielle Datenleitung
- 7-Segmentanzeig mit 6 Stellen
- Serielle Schnittstelle als Verbindung zum PC

3.1.2 Uhrlogik

Für die Realisation der Uhrlogik haben sich grundsätzlich zwei Möglichkeiten angeboten: Die eine bestand darin, eine Uhr ausschliesslich mit diskreten Bauteilen zu bauen. Da dies aber sehr grosse elektronische Kenntnisse fordert und sich nur bei grossen Stückzahlen des Endprodukts lohnt, haben wir von dieser Lösung abgesehen.

Die andere Möglichkeit war, einen Mikrokontroller einzusetzen. Diese gibt es aber in verschiedensten Variationen (mit CISC- oder RISC-Architektur, mit verschiedensten Arten von Programm- und Arbeitsspeichern, mit integrierten Eingabe-/Ausgabebausteinen, usw.).

Nach einigen Abklärungen fiel die Entscheidung auf einen Kontroller der Atmel AVR8-Serie. Aus dieser Serie gibt es verschieden Kontroller mit derselben Architektur aber verschiedenen Optionen, das heisst der Kontroller ist entsprechend den Anforderungen wählbar. Zudem ist bei dieser Serie der Herstellersupport ausgezeichnet: Atmel bietet eine Entwicklungsumgebung mit einen integrierten Simulator zum freien Download an.

Da für die gemultiplexte Anzeige 15, für den Empfang der Zeitsignale eine und für die serielle Schnittstelle zwei¹ – also insgesamt 18 – PIO-Leitungen und zudem ein UART und zwei Timer notwendig sind, fiel die Auswahl auf den AT90S4433. Dieser bietet unter anderem zwei 6-Bit und ein 8-Bit parallel Port, ein UART, einen 8-Bit und ein 16-Bit Timer, 4KBytes integrierter Flashspeicher und 128Bytes flüchtiger Arbeitsspeicher. Nur die unterstützten Quarzfrequenzen sind etwas hoch für dieses Projekt, doch es schadet sicherlich nicht, mehr Ressourcen als gerade nötig zur Verfügung zu haben.

3.1.3 Funksignalempfang

Die interessanteste Lösung wäre für dieses Teilproblem eine wäre eine mit diskreten Bauteilen gewesen. Doch Herr Obrist, der sich schon eingehend mit diesem Problem befasst hat und über ein riesiges Wissen in der Elektronik verfügt, hat dringend davon abgeraten. Aus diesem Grund kam nur ein Fertigbaustein in Frage.

Von diesen gibt es zwei Varianten die für das Projekt in Frage gekommen sind: Einer mit einem „normalen“ seriellen Signal als Ausgabe und einer der nur das Signal Decodiert. Um das Signal besser kontrollieren zu können und um die Aufgabe interessant zu halten fiel die Entscheidung auf den reinen Decoder der über eine Steuerleitung mit dem Controller verbunden wird. In diesem kann das Signal dann gesampelt und umgewandelt werden.

3.1.4 Anzeige

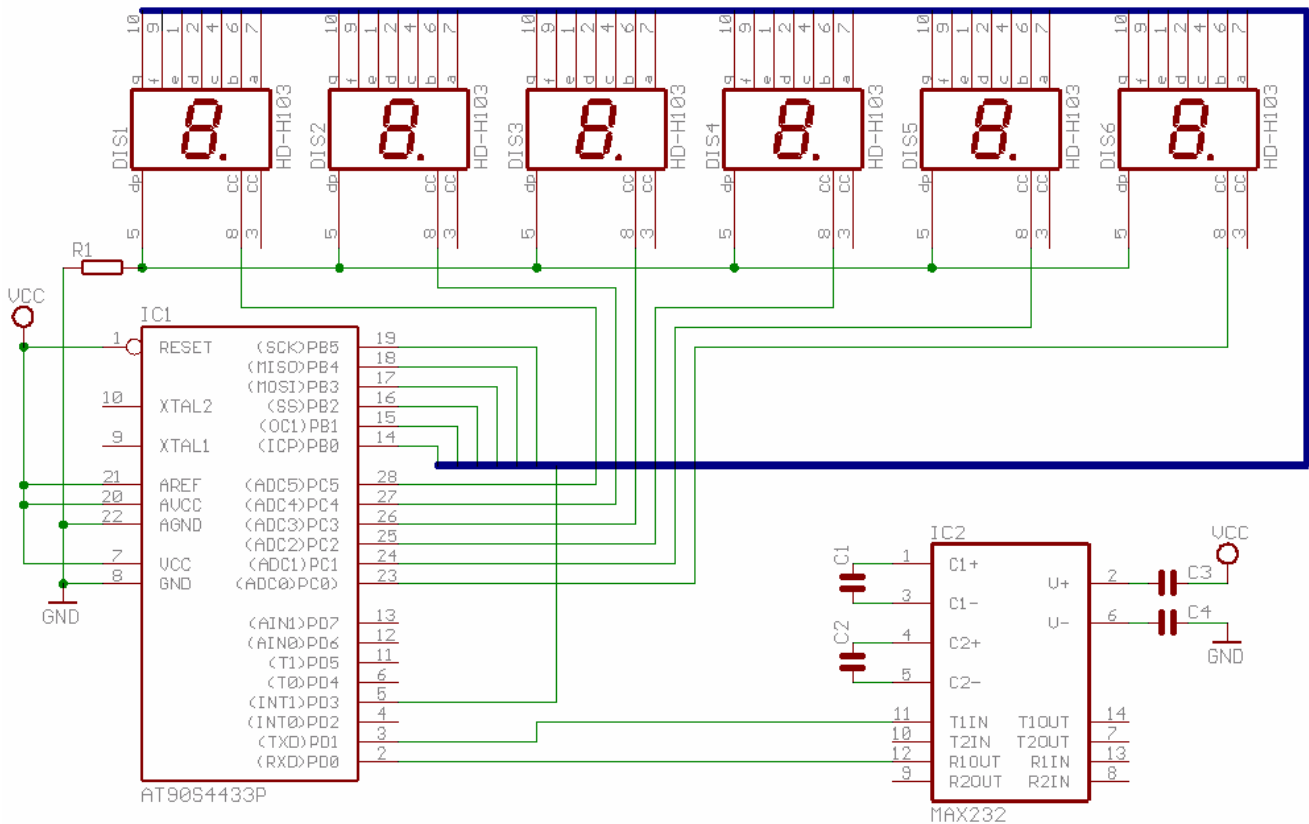
Die vielfältigste Lösung wäre ein Display mit integrierter Logik gewesen. Diese Displays bieten viele Möglichkeiten was die Ausgabe anbelangt, sind aber aufwändig anzusteuern. Etwas weniger vielfältig und weniger kompliziert wäre die Möglichkeit einer LED-Matrix gewesen. Doch auch diese fiel weg, da sie einen hohen Programmier- und Hardwareaufwand fordert.

Schlussendlich durchgesetzt hat sich die Lösung mit einer 7-Segment-Anzeige, angesteuert mit Multiplexing. Klassisch, simpel und funktionierend. Auf einem Codierbaustein wurde verzichtet, da dieser zwar PIO-Leitungen spart, aber die Ausgabe von anderen Zeichen als Zahlen verunmöglicht. Die Umrechnung der Binärzahl in das Bitmuster für die Anzeige wird stattdessen softwaremässig gelöst.

¹ Bei der AVR8-Serie verfügen die meisten PIO-Leitungen über eine Doppelfunktion

3.1.5 Das Schema

Diese Überlegungen galt es nun in das Schema umzusetzen. Der erste Entwurf dessen sah wie folgt aus:



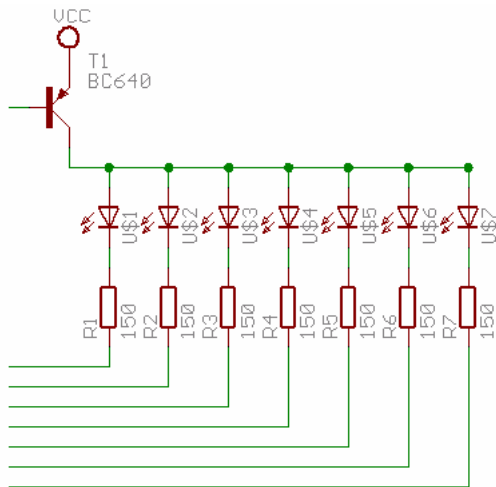
Fehlende Teile auf dem Schema:

- Stabilisator für die Stromversorgung
- Quarz mit Anschluss an XTAL1 und XTAL2
- Lötkontakte für die serielle Schnittstelle, für das Decoderboard und für den Stromanschluss
- Die Versorgungsspannung auf dem Board sollte mit Kondensatoren stabilisiert werden

Fehler auf dem Schema:

- Wird die Strombegrenzungsdiode für die Diode wie R1 angebracht leuchtet eine Angezeigte 1 heller als eine 8 da sich das Verhältnis von R1 zum Ersatzwiderstand der Diode ändert.
- Obwohl die Ausgabeports für die Ansteuerung der Dioden gedacht sind, übersteigt in dieser Konstellation der Strom der über die Ports fließt die Maximalwerte => Deshalb ist der Einbau einer Verstärkenden Schaltstufe notwendig
Da die auf 1 geschalteten PIO-Leitungen aber nur ein Strom von 3mA liefern können, ohne dass die Spannung zusammenbricht, muss diese Schaltstufe

invertierend sein (gegen 0 können bis 20mA pro PIO-Leitung fließen). Hier liegt eine Lösung mit einem Transistor auf der Hand:



Das fertig Schema und das daraus resultierende Board-Layout befinden sich im Anhang diese Dokuments.

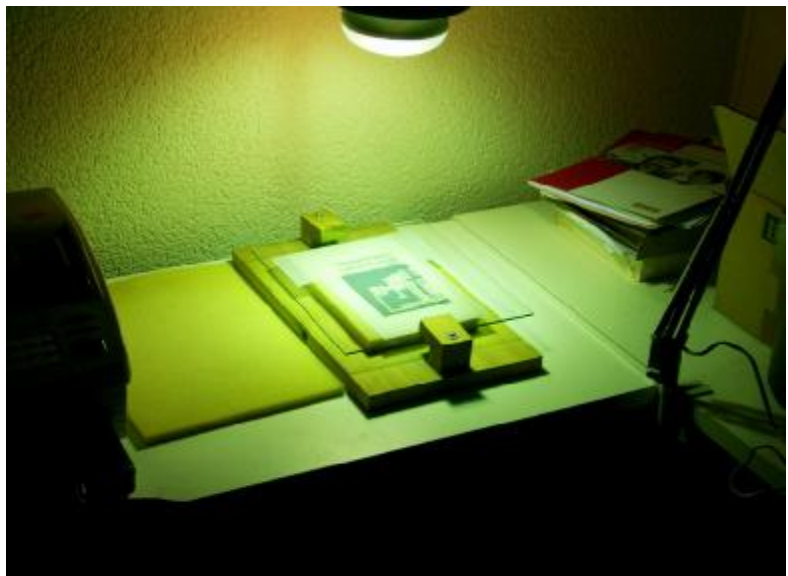
3.2 Herstellung

Als erstes wird die Platine hergestellt, wobei dieser Vorgang in Punkt „Ätzen“ genauer beschrieben wird. Der weitere Verlauf der Erstellung der Funkuhr wird in „Zusammenbau“ erläutert. (Alle im Abschnitt „Herstellung“ verwendeten Bilder wurden während der Produktion der Uhr erstellt)

3.2.1 Ätzen

Als erstes wird das Layout der Platine, welches wir erstellt haben, auf eine Klarsichtfolie ausgedruckt und die Platinen aus dem entsprechenden Kasten geholt, welche auf UV-Licht empfindlich ist. Das Layout für die Platine befindet sich im Anhang.

Die Folie wird nun dementsprechend unter einer Glasscheibe auf die Platine gelegt. Falls noch irgendwelche Ungereimtheiten auf der Folie sind, dann soll die Folie mit einem Korrekturstift korrigiert werden. Oberhalb wird die UV-Lampe im richtigen Abstand installiert und dann für die

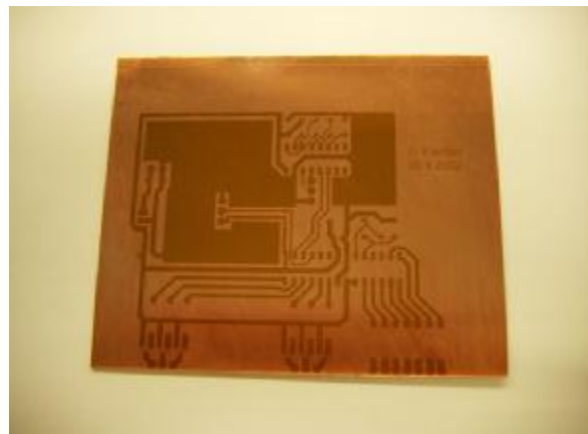


benötigte Zeit angezündet. Die Platine wird somit belichtet.

Nachdem die Platine belichtet wurde, muss man sie sofort in ein Entwicklerbad eintauchen. Als Entwickler wurde „RP25 Positiv Entwickler“ verwendet. Nach dem Eintauchen in den Entwickler löst sich der Lack und das Layout der Platine erscheint.

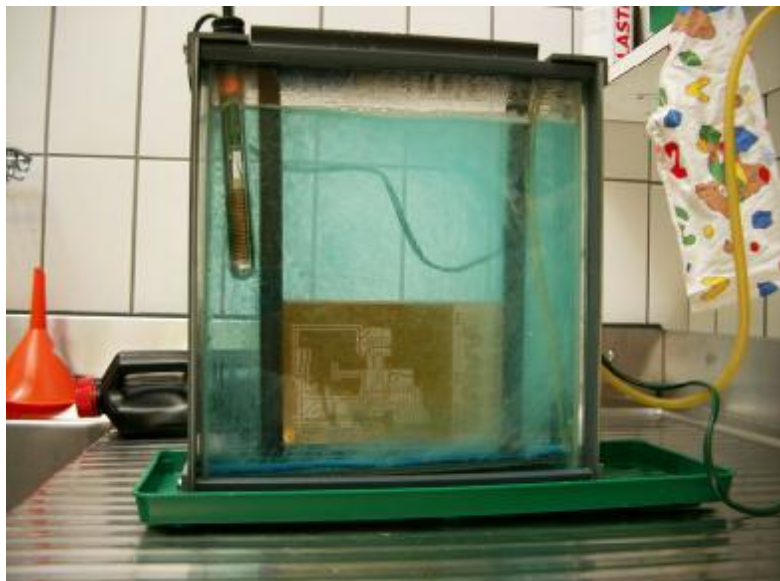


Nach der sich der Lack gelöst hat, wird die Platine mit Wasser abgespült. Das Resultat sollte dann so aussehen, wie im Bild rechts von diesem Text, man sieht den zurückbleibenden Lack. An seiner Stelle werden die Leitbahnen zurückbleiben. Die Platine muss jetzt zuerst getrocknet werden, bevor an ihr weiter gearbeitet werden kann.

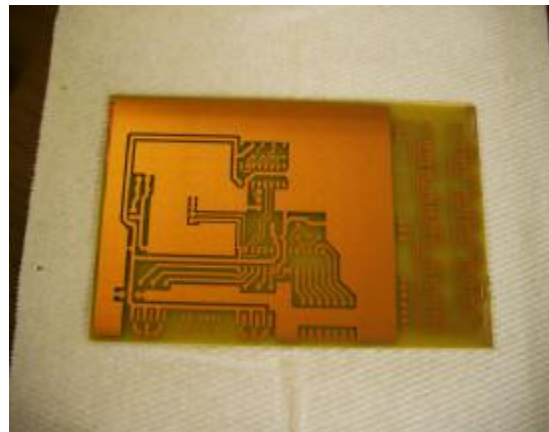
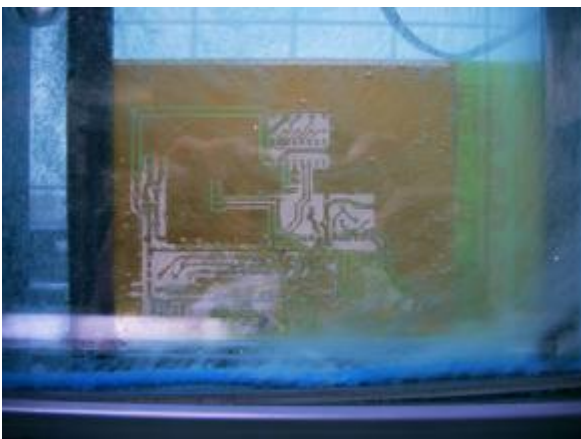


Im nächsten Arbeitsschritt kommen wir zum eigentlichen Ätzen. Wir nahmen als Ätzmittel „Natriumpersulfat“ (Bild unten links). Dieses wurde nach Angaben des Herstellers entsprechend vorbereitet und in den unten gezeigten Behälter eingefüllt. Die trockene Platine wird, wie auf dem rechten Bild unten, im Behälter installiert. Das ganze wird nun in Betrieb genommen.

Quelle: <http://www.qsl.net/dg5dbz/platine/platine.html>



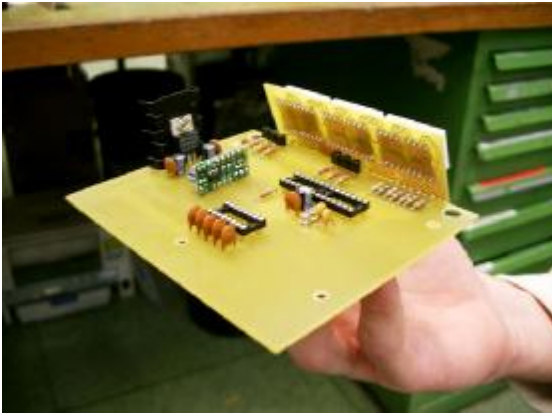
Das Wasser wird sich blau färben. Wo sich nun kein Lack mehr befindet, da wird das Kupfer nun entfernt. Die Platine wird solange im Behälter gelassen, bis nur noch die entsprechenden Leiterbahnen existieren. Sobald dies eintrifft, wird die Platine entfernt und abgetrocknet. Die Platine wird mit einem Lappen, welcher mit Verdünner getränkt ist, gereinigt. Das Resultat kann man unten rechts begutachten. Somit wäre die Platine zur weiteren Verarbeitung bereit.



3.2.2 Zusammenbau

Nachdem die Platine erstellt wurde, beginnt man mit der Arbeit am Gehäuse, welches nach Vorgabe, die sich im Anhang befindet, erstellt. Das gleiche geschieht mit der Platine, man muss zum Beispiel die Löcher für die Bauteile bohren oder die Platine in die jeweiligen beiden Stücken teilen.





Wenn dies erledigt ist, bestückt man die Platine mit den Bauelementen, wie sie im Anhang beschrieben wurden und lötet diese an. Schlussendlich muss noch der Baustein mit dem Programm beschrieben werden und eingesetzt. Die Uhr ist nun voll funktionstüchtig, das Zuschrauben der Uhr ist die letzte Arbeit, welche noch erledigt werden muss.

3.3 Signal / Zeitübertragungsdienste

3.3.1 Zeitzeichensender DCF77

Die Sendestation befindet sich Mainflingen im deutschen Bundesstaat Hessen. Er wird seit 1959 betrieben.

3.3.2 Was wird gesendet?

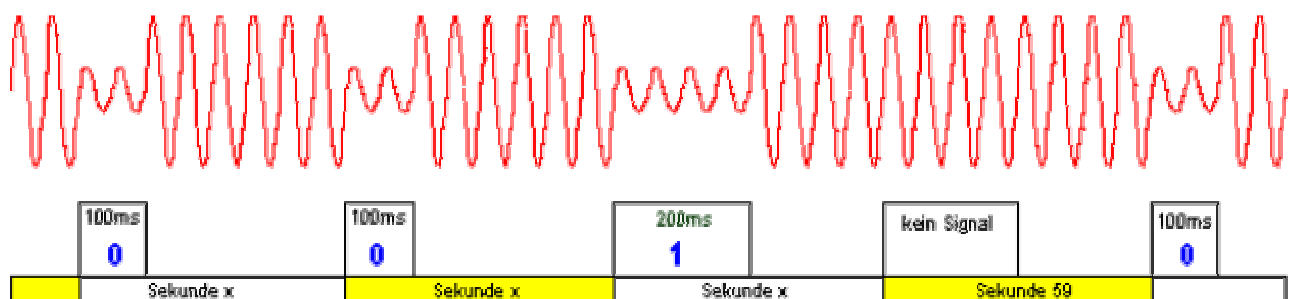
Der Zeitzeichensender erhält die exakte Zeit von einer Atomuhr der Physikalisch Technischen Bundesanstalt und sendet diese mit Datum über Kurzwelle aus.

Im Umkreis von 500km um Mainflingen ist die Bodenwelle dominant. Zwischen 500km und 1000km sind die Bodenwelle und die an den oberen Atmosphäreschichten reflektierte Welle etwa gleich stark, was wegen den unterschiedlichen Laufzeiten zu Interferenzen führen kann, so dass sich die zwei Wellen im schlechtesten Fall fast vollständig aufheben. Da das Signal über Kurzwelle gesendet wird kann Niederschlag zu einer Störung des Empfangs führen.

3.3.3 Kodierung des Zeitsignals

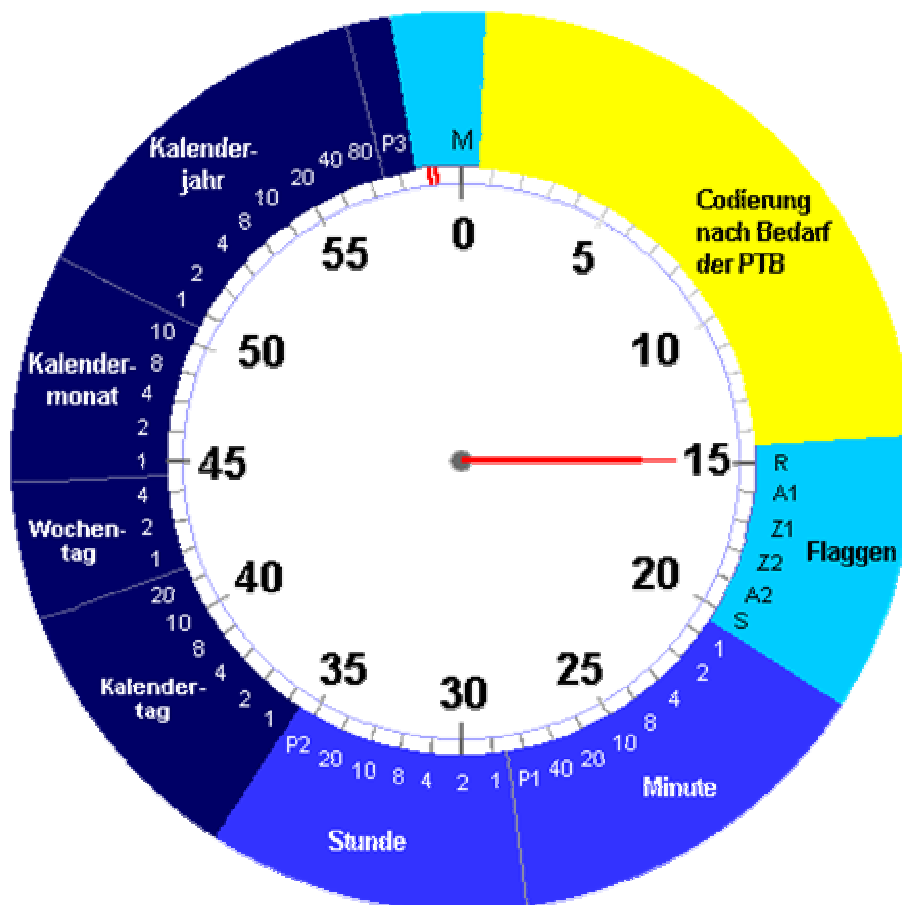
Das Signal wird amplitudenmoduliert auf einer Trägerfrequenz von 77,5kHz gesendet. 60 Mal in der Minuten wird die Signalstärke auf 25% abgesenkt. Eine Absenkung für 100ms bedeutet eine logische 0, eine Absenkung für 200ms eine logische 1.

Für die Minutensynchronisation wird in der 59. Sekunde kein Signal gesendet.



Innerhalb der 21. und 58. Sekunde wird die Minute, die Stunde, der Kalendertag, der Wochentag, der Monat und das Jahr übertragen. Die Informationen werden jeweils im BCD-Format übertragen.

Dezimal-System	BCD-Code
1	0001
2	0010
3	0011
...	...
9	1001
10	0001 0000
11	0001 0001
...	...



Sekunde	Bit	Bemerkungen	
00		Minutenmarke	
01-14		reservierte Bits, nur für die PTB	
15	R	Antennenbit, Betriebs- Reserveantenne	0: Betr - 1: Reserve
16	A1	Ankündigung MEZ/MESZ bzw. MESZ/MEZ	
17	Z1	Zonenzeitbit 1	0: MEZ - 1: MESZ
18	Z2	Zonenzeitbit 2	0: MESZ - 1: MEZ
19	A2	Ankündigung Schaltsekunde (eine Std. zuvor)	
20		Startbit der Zeitinformation.	1: immer 1
21-27		BCD-codierte Minuten (7Bit)	
28	P1	Paritätsbit der Minuten (evenParity, sek 21-27)	
29 - 34		BCD-codierte Stunden (6 Bit)	
35	P2	Paritätsbit der Stunden (evenParity, sek 29-34)	
36 - 41		BCD-codierter Tag (6 Bit)	
42 - 44		BCD-codierte Nummer des Wochentages (4 Bit)	1- Mo ... 7- So
45 - 49		BCD-codierter Monat (5 Bit)	
50 - 57		BCD-codierte Jahreszahl (ohne Jahrhundert, 8 Bit)	
58	P3	Paritätsbit des Datums (even Parity, sek.36-57)	
59		keine Absenkung der Trägerfrequenz	

Quellen: <http://www.metas.ch/de/labors/official-time/hgb/index.html>
http://www.metas.ch/de/labors/official-time/pdf/hgb_info_d.pdf

3.4 Software Computer (SyncTime)

3.4.1 Funktion

Zusätzlich zur Hardware Funkuhr haben wir eine Computer-Software entwickelt.

Mit Hilfe dieser Software ist es möglich die Informationen, welche die Uhr empfängt, mit dem Computer auszutauschen. Dies geschieht über die Serielle Schnittstelle, welche schon seit einigen Jahren standardmässig zu jedem Computer gehört. Somit ergibt dies eine einfache Alternative zum 1995 eingeführten SNTP-Standard (**S**imple **N**etwork **T**ime **P**rotocol), welches über das Internet die Zeit nach speziellen Zeit-Servern richtet. Der Vorteil unserer Lösung liegt darin, auch Benutzer ohne Internet-Zugang mit der aktuellen Zeit zu versorgen.

3.4.2 Programmiersprache

Die Applikation wurde vollständig in der Programmiersprache C# entwickelt. C# wurde in den letzten 2 Jahren von Microsoft eigens für das .Net Framework entwickelt, welches im Februar 2002 in der ersten öffentlichen Version 1.0 erschien. Mit dem .Net Framework setzt Microsoft eine neue Strategie um, die dem Entwickler eine grössere Flexibilität bietet. Der Vor- und zugleich der Nachteil des .Net ist die Virtual Machine. Um Anwendungen, die in einer der zurzeit 22 .Net Sprachen geschrieben wurden, ausführen zu können muss das .Net Framework installiert sein. Die Anwendungen sind nun nicht mehr fest in den Maschinencode übersetzt sondern dies geschieht erst mit dem Programmstart. Der Vorteil darin liegt, dass die Umwandlung in den Maschinencode automatisch, abhängig vom Computer, optimal umgesetzt wird. Da die Umwandlung erst vom Framework übernommen wird ermöglicht es auch den Code auf anderen Plattformen und optimiert auf deren Möglichkeiten zu kompilieren. Einzige Voraussetzung ist, dass zu diesem Betriebssystem ein .Net Framework vorhanden ist. Zurzeit entwickelt Corel eine Portierung in das Unix-Derivat BSD (**B**erkeley **S**oftware **D**istribution) zudem findet in der Open-Source Community die Portierung nach Linux, einem weiteren Unix-Derivat, statt. Beide Portierungen sind jedoch noch nicht vollständig beendet.

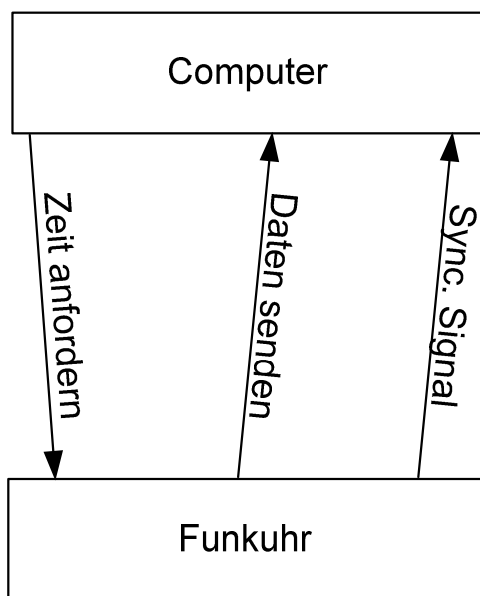
Der Nachteil, wie schon erwähnt, liegt bei der Voraussetzung vom .Net Framework. Zur heutigen Zeit ist dies noch nicht allzu gross Verbreitet und der Benutzer muss es manuell nachinstallieren. Dies wird sich aber in den nächsten Jahren drastisch ändern, da es zumindest jedes neue Windows direkt implementiert haben wird. Das erste OS wird der Windows .Net Server sein, der im Frühjahr 2003 erscheinen wird. Zudem wird Microsoft mit den nächsten Service Packs das Framework auch bei den älteren Windows Versionen installieren, falls es noch nicht vorhanden ist.

Eine ähnliche Abhängigkeit war vor einigen Jahren auch mit C++ und Visual Basic so. Doch in der heutigen Zeit bemerkt dies kaum ein Benutzer, da es schon zum einem Standard Microsoft Betriebssystem gehört.

Quellen: <http://www.microsoft.com/netframework>
<http://www.microsoft.com/net>
<http://www.microsoft.com/vstudio>

3.4.3 Kommunikation zwischen dem Computer und der Uhr

Um eine Kommunikation zu ermöglichen setzt es gewisse Grundregeln voraus, welche in einem so genannten Protokoll spezifiziert werden. Da die Kommunikation zwischen der Uhr und dem Computer nicht einen besonders grossen Datenaustausch erfordert, gelten relativ einfache Regeln.



Somit sehen die Kommunikations-Schritte so aus, dass der Computer zuerst in, vom Benutzer der Software, bestimmten Intervallen eine Anforderung (Hex-Code: 00) an die Funkuhr sendet um der Uhr die Bereitschaft mitzuteilen. Sobald dies geschehen ist, sendet die Uhr die Informationen in der Reihenfolge: Sekunde, Minute, Stunde, Tag, Monat und als letztes das Jahr zurück. Um die Daten digital zu übertragen wurde auf die BCD-Codierung (**B**inary **C**oded **D**igit / **D**ecimals) gesetzt. Am Ende der Übertragung wird zur Synchronisation ein

Synchronisations-Byte (Hex-Code: FF) gesendet. Die Software synchronisiert exakt zu diesem Zeitpunkt die Windows System-Uhr nach den empfangenen Daten.

4 Schlussteil/Reflexion

4.1 Hardwareentwicklung

Bei der Entwicklung des Schemas wäre es sicher hilfreich gewesen, vorgängig einige Tests mehr durchzuführen. So haben beim Prototyp nicht bemerkt, dass der Schnittstellentreiberbaustein für die serielle Schnittstelle Störsignale produziert, die den sauberen Empfang je nach Ausrichtung der Antenne im Bezug zum Schnittstellentreiber stört. Dieser Fehler hätte sich vermeiden lassen, wenn wir den Prototyp ausgiebiger getestet hätten.

Es ist zwar so, dass an gewissen Orten der Empfang sehr schlecht funktioniert, dies ist aber auch bei anderen Funkuhren der Fall und könnte durch eine Präzisionsantenne und ein genau abgestimmtes Decodierboard erreicht werden. Und die bedingt entweder grosse Geldausgaben oder aber einen hohen Entwicklungsaufwand für den Eigenbau.

Leider bringt aber schon das Sendeverfahren des Zeitsignals einige Probleme mit sich.

Quelle: www.metas.ch

Abgesehen von dem nicht 100%ig zufrieden stellendem Empfang sind wir mit der Uhr vollumfänglich zufrieden. Die Anzeige und die Schnittstelle zum PC funktionieren sauber und fehlerfrei. (A. Fischer)

4.2 Software Mikrokontroller

Für die Software des Mikrokontrollers war es ein riesiger Vorteil das wir zuerst ein Flussdiagramm erstellt haben. Ohne dieses wäre die Fehlersuche extrem erschwert gewesen und spätestens ab der 500sten Programmzeile wäre die Übersicht ganz verloren gegangen.

Wie oben bereits angetönt sind beim Mikrokontrolle noch einige unverwendete Ressourcen vorhanden, die eine Erweiterung des Codes (z.B. bessere Fehlererkennung und -korrektur, Datumsanzeige an der Uhr), ermöglicht hätten. Leider blieb für die Verwirklichung dieser Ideen keine Zeit mehr übrig.

Abschliessend blicken wir auf eine fehlerfreie und funktionierende Software zurück, deren Entwicklung einige Stunden Arbeit und viel Durchhaltevermögen gekostet hat. (A. Fischer)

4.3 Software Computer (SyncTime)

Bei der Auswahl der Programmiersprache entschloss ich mich für eine Sprache, die ich zu diesem Zeitpunkt noch nicht vollkommen beherrschte. Zu diesem Entschluss kam ich, da

ich mich schon seit längerem für diese Sprache interessierte aber bis jetzt noch kein Projekt hatte um mich stärker damit auseinander zu setzen. Dies führte logischerweise zu mehr Aufwand dafür habe ich in dieser Zeit auch viel neues dazu gelernt. Zudem stiess ich beim Programmieren teilweise auf unlösbare Ansätze. Und musste den Aufbau zum Teil ändern. Zum jetzigen Zeitpunkt läuft das Programm stabil und beherrscht alle geplanten Features. Jedoch hätte ich gerne noch ein paar Sachen zusätzlich eingebaut, doch die Zeit war zu knapp um diese auch noch nach Stabilität zu prüfen können. Da ich das Programm selber einsetze werde ich vermutlich nach dem Projekt weitere Neuerungen einbauen und auf Wunsch auch veröffentlichen. (D. Gut)

4.4 Reflexion zum gesamten Projekt

Wenn man so das Arbeitsjournal betrachtet, dann fällt vor allem auf, dass A. Fischer wohl am meisten gearbeitet hat. Dies ist gut möglich, weil die Aufteilung der einzelnen Projektteile etwas ungeschickt gewählt wurde. Auf der anderen Seite konnte man den Teil von A. Fischer auch nicht weiter aufteilen, weil sonst der Zusammenhang verloren gegangen wäre. Dieser Punkt führte auch dazu, dass sich eine kleine Demotivation unter den weitere Gruppenmitgliedern ausbreitete, diese wurde aber mit Hilfe von einem intensiven Gespräch in der Gruppe wieder wettgemacht.

Ein weiterer negativer Punkt in unserem Projekt war der Aufwand. Dieser übersteige unsere Erwartungen um ein mehrfaches. Das wirkte sich dann auf die Dokumentation aus, welche die Seitenzahl um einiges höher steigen lies, als vorgegeben ist. Es mussten darum auch eigene Interessen, wie zum Beispiel Hobbys dem Projekt weichen. Das Projekt war aus unserer Sicht zu gross, bzw. wir hätten um einiges mehr Zeit benötigt, um nicht in Stress zu geraten.

Das Ergebnis lässt sich aber sehen, wir haben eine funktionierende Funkuhr, über die der PC, mit Hilfe eines Programms und der Seriellen Schnittstelle, die aktuelle Zeit abgleichen kann. Uns ist aufgefallen, dass wir das Budget, welches wir am Anfang geplant haben, nicht überschritten haben. Das benötigte Material & die Kosten sind im Anhang der Dokumentation beschrieben. Zusätzlich haben wir weitere Erfahrungen in Bezug zu Gruppen- und Projektarbeiten gesammelt und werden diese auch dementsprechend im nächsten Projekt nutzen.

(F. Böhler)

5 Verdankung

Wir möchten Josef Obrist ganz herzlich für seine Unterstützung danken. Herr Obrist hat schon selbst mit einem Projekt begonnen, bei dem er das Funksignal nur mit diskreten Bauteilen einfängt. Er hat grosse Erfahrung mit Prozessoren und Kontrollern, und hat vor uns schon viele andere Berufslernende unterstützt.

Auf die Frage was er gerne möge, hat er mit einem lächeln geantwortet: „Alles was ungesund ist.“ Einige Sekunden hat er angefügt, dass es nicht nötig sei, ihm ein Präsent zu machen, es sei sein Beruf als Lehrlingsbetreuer.

Es wäre schön, wenn jeder Lehrlingsbetreuer seinen Beruf so ernst nähme!

Aus diesem Grund haben wir uns entschieden, Herr Obrist nach der Projektabgabe die gesamte Dokumentation und ein kleines Dankeschön zu übergeben.