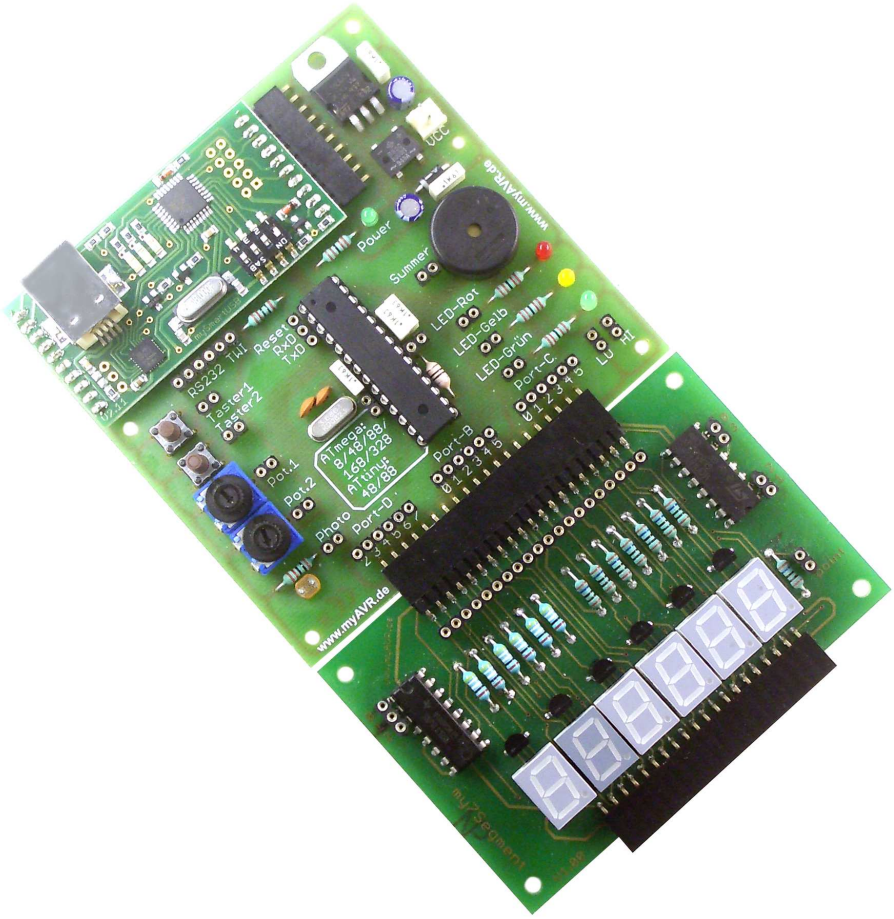
StD Udo John
Dipl. Ing. Päd. Alexander Huwaldt

myAVR Projekt
7-Segment-Anzeige

Interessante Experimente mit einer 7-Segment-Multiplexanzeige

- Digitalvoltmeter
- Stoppuhr
- Echtzeituhr



Leseprobe

2. Auflage: Juli 2010

© Laser & Co. Solutions GmbH
www.laser-co.de
www.myavr.de
service@myavr.de
info@udojohn.de
Tel: ++49 (0) 3585 470 222
Fax: ++49 (0) 3585 470 233

Inhaltsverzeichnis

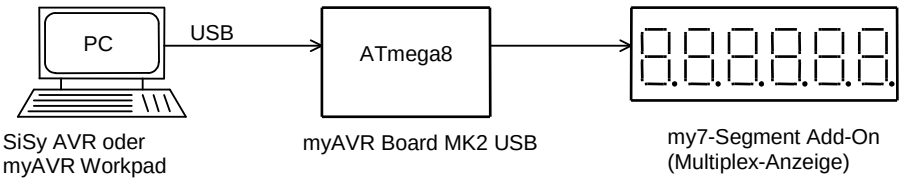
1 Einleitung

Mit dem Projekt „myAVR Projekt 7-Segment-Anzeige“ wird das Ziel verfolgt, eine mehrstellige numerische Anzeige für Mikrocontroller-Systeme zu realisieren. Diese 7-Segment-Anzeige, nachfolgend auch als Multiplex-Anzeige benannt, stellt eine preiswerte und ansprechende Möglichkeit dar, um zum Beispiel Messergebnisse gut sichtbar darzustellen. Sie lässt sich beispielsweise im Gegensatz zu einer LCD-Anzeige leicht programmieren und ist mit 6 Stellen für viele Anwendungen einsetzbar.
In dieser Dokumentation werden zunächst die Wirkungsweise, die schaltungstechnische Realisierung und ein Testprogramm beschrieben.
In weiteren Kapiteln werden beispielhaft drei Projekte vorgestellt:

- Projekt Digitalvoltmeter
- Projekt Stoppuhr
- Projekt Echtzeituhr

Ausgehend von den Programmstrukturen wird die Programmentwicklung bis zum vollständigen Assemblerprogramm dargestellt. Somit soll diese Dokumentation auch mit dazu beitragen, die Kenntnisse in der Programmierung zu vertiefen und zu erweitern. So werden hier besonders der Analog/Digital-Wandler und die Timer des ATmega8 mit den zugehörigen Interrupts an konkreten Beispielen beschrieben.

Zur Programmentwicklung wird SiSy AVR oder alternativ myAVR Workpad für diese Projekte verwendet. Als Hardware wird ein myAVR Board MK2 USB (oder wahlweise myAVR Board MK1 LPT) und ein my7-Segment Add-On für die Multiplex-Anzeige eingesetzt.



Abschließend werden in dieser Dokumentation einige Anregungen für weitere mögliche Projekte gegeben, bei denen der Anwender die Möglichkeit hat, selbstständig Programme zu entwickeln und den Lernerfolg zu überprüfen.

Die Autoren wünschen viel Erfolg beim Studium!

2 Allgemeine Beschreibung des my7-Segment Add-On

2.1 Aufbau

Das my7-Segment Add-On ist ein anschlussfertiges Erweiterungsmodul, welches direkt über die standardisierte Steckerleiste mit einem myAVR Systemboard verbunden werden kann. Über ein solches Systemboard kann die Ansteuerung erfolgen.

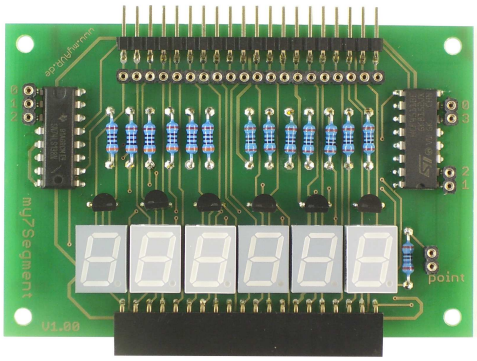


Abbildung 2.1: fertig bestücktes my7-Segment Add-On

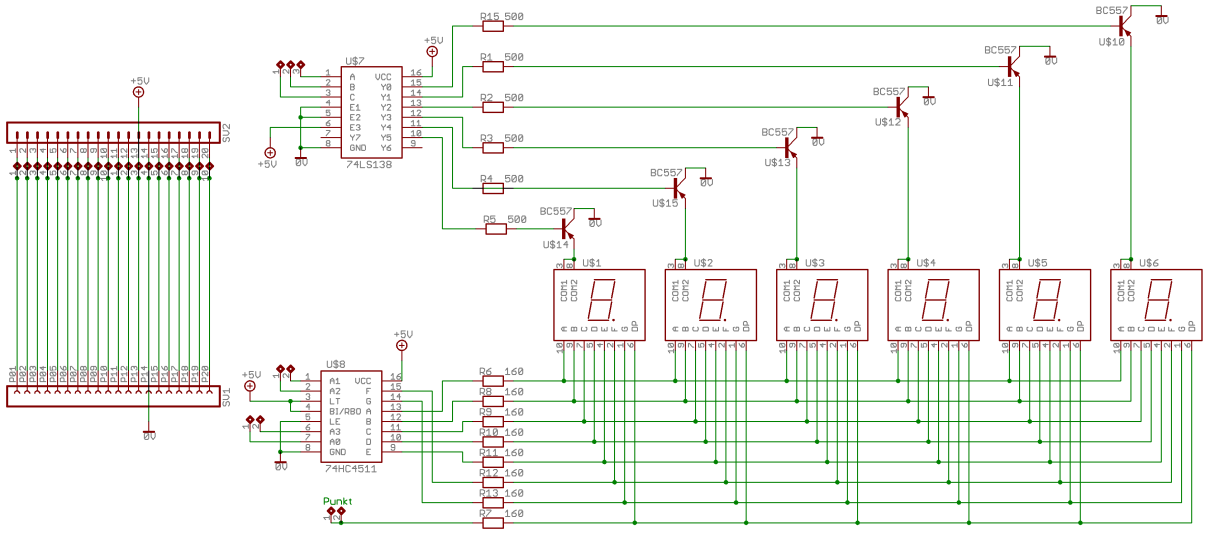


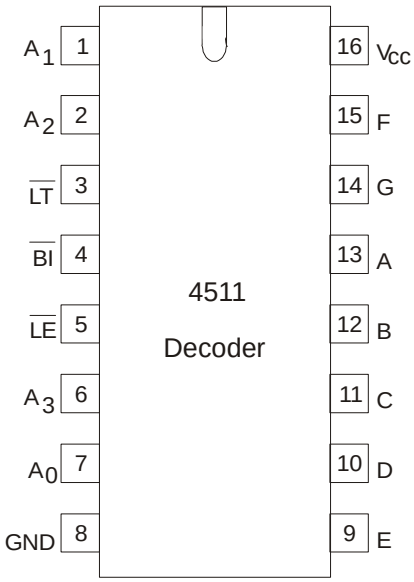
Abbildung 2.2: Schaltplan zu my7-Segment Add-On

2.2 Funktionsweise

Eine 7-Segment-Anzeige besteht aus 8 einzelnen LED's, welche einzeln angesteuert werden. Die Segmente a bis g gestatten eine Anzeige der Zahlen von 0 bis 9 (siehe Abbildung 2.3). Eine weitere LED ist für einen Dezimalpunkt vorgesehen.

Abbildung 2.3: Darstellung der Ziffern auf einer 7-Segment-Anzeige

Alle LED's haben einen gemeinsamen Kathodenanschluss. Im Schaltplan sind die Segmente a bis g aller Anzeigen U$1 bis U$6 parallel miteinander verbunden. Jede einzelne 7-Segment-Anzeige liegt mit ihrem Kathodenanschluss über einen Transistor U$10 bis U$15 an Masse. Ein Low-Signal auf den entsprechenden Transistor schaltet die Kathoden nach Masse und lässt die entsprechende Anzeigenstelle leuchten.
Die erforderliche BCD-7-Segment-Codewandelung erfolgt durch den IC-Baustein 74HC4511. Die Pinbelegung dieses Bausteins ist in Abbildung 2.4 dargestellt und die Wahrheitstabelle in Abbildung 2.5.



Pin Nummer	Symbol	Name und Funktion
3	$\overline{LT}$	Funktionstest (LOW-Aktiv)
4	$\overline{BI}$	Adressdeaktivierung (LOW-Aktiv)
5	$\overline{LE}$	Taktfrequenzeingang
7, 1, 2, 6	A_0 to A_3	BCD Adressierung
8	GND	Masse (0 V)
13, 12, 11, 10, 9, 15, 14	A to G	Segmentausgabe
16	V_{CC}	Positive Versorgungsspannung

Abbildung 2.4: Pinbelegung und –beschreibung des 74HC4511

An den Eingängen A0 bis A3 dieses Bausteines wird eine BCD-Zahl angelegt. Das sind die Dualzahlen von 0b0000 bis 0b1001, entsprechend die Dezimalzahlen von 0 bis 9. An den Ausgängen A bis G wird der 7-Segment-Code ausgegeben. In der Wahrheitstabelle sind die Ein- und Ausgänge mit A0…A3 bzw. A…G bezeichnet.

INPUTS							OUTPUTS							DIS-PLAY
$\overline{LE}$	$\overline{BI}$	$\overline{LT}$	A_3	A_2	A_1	A_0	A	B	C	D	E	F	G	
X	X	L	X	X	X	X	H	H	H	H	H	H	H	8
X	L	H	X	X	X	X	L	L	L	L	L	L	L	blank
L	H	H	L	L	L	L	H	H	H	H	H	H	L	0
L	H	H	L	L	L	H	L	H	H	L	L	L	L	1
L	H	H	L	L	H	L	H	H	L	H	H	L	H	2
L	H	H	L	L	H	H	H	H	H	H	L	L	H	3
L	H	H	L	H	L	L	L	H	H	L	L	H	H	4
L	H	H	L	H	L	H	H	L	H	H	L	H	H	5
L	H	H	L	H	H	L	L	L	H	H	H	H	H	6
L	H	H	L	H	H	H	H	H	H	L	L	L	L	7
L	H	H	H	L	L	L	H	H	H	H	H	H	H	8
L	H	H	H	L	L	H	H	H	H	L	L	H	H	9
L	H	H	H	L	H	L	L	L	L	L	L	L	L	blank
L	H	H	H	L	H	H	L	L	L	L	L	L	L	blank
L	H	H	H	H	L	L	L	L	L	L	L	L	L	blank
L	H	H	H	H	L	H	L	L	L	L	L	L	L	blank
L	H	H	H	H	H	L	L	L	L	L	L	L	L	blank
L	H	H	H	H	H	H	L	L	L	L	L	L	L	blank
H	H	H	X	X	X	X	(1)							(1)

1. Ist abhängig vom verwendeten BCD-Code während der LOW-zu-HIGH Übertragung von LE.

L = LOW voltage Level
H = HIGH voltage Level
X = Don't Care

Abbildung 2.5: Wahrheitstabelle des IC-Bausteins 74HC4511

Der 7-Segment-Code liegt jetzt zwar an allen Anzeigen an, leuchten wird jedoch nur die Stelle, bei welcher der zugehörige Transistor (U$10…U$15) den Kathodenanschluss an Masse durchschaltet. Das Durchschalten eines Transistors erfolgt mit einem Low-Pegel an der Basis.
Der IC-Baustein 74LS138 ist ein 1-aus-8-Demultiplexer/Decoder. In Abhängigkeit von den Eingangssignalen A, B und C wird nur einer der 8 Ausgänge Y0 bis Y7 auf Low-Pegel geschaltet. Liegt an den Eingängen A, B und C beispielsweise 0b000 an, so wird der Ausgang Y0 auf Low geschaltet und bringt damit U$6 (Stelle 1) zur Anzeige. Die Widerstände R6 bis R13 begrenzen den Strom für die Segmente auf die zulässigen ca. 20 mA.

Für das Verständnis der 7-Segment-Anzeige ist zu beachten, dass zu einer Zeit immer nur eine Anzeige aktiv ist. Wählt man die Stellen genügend schnell nacheinander aus (Multiplex-Betrieb) so erscheint für das menschliche Auge eine ‚stehende' Anzeige. Um eine flimmerfreie Anzeige zu erhalten, sollten alle Stellen ca. 50 mal pro Sekunde ausgewählt werden.
Das bedeutet: Alle 1/50 s = 20 ms sollte die komplette Anzeige erneuert werden. Das heißt wiederum bei einer 6-stelligen Anzeige, dass jede Stelle für 20 ms/6 = 3,3ms (oder weniger) anzusteuern ist.

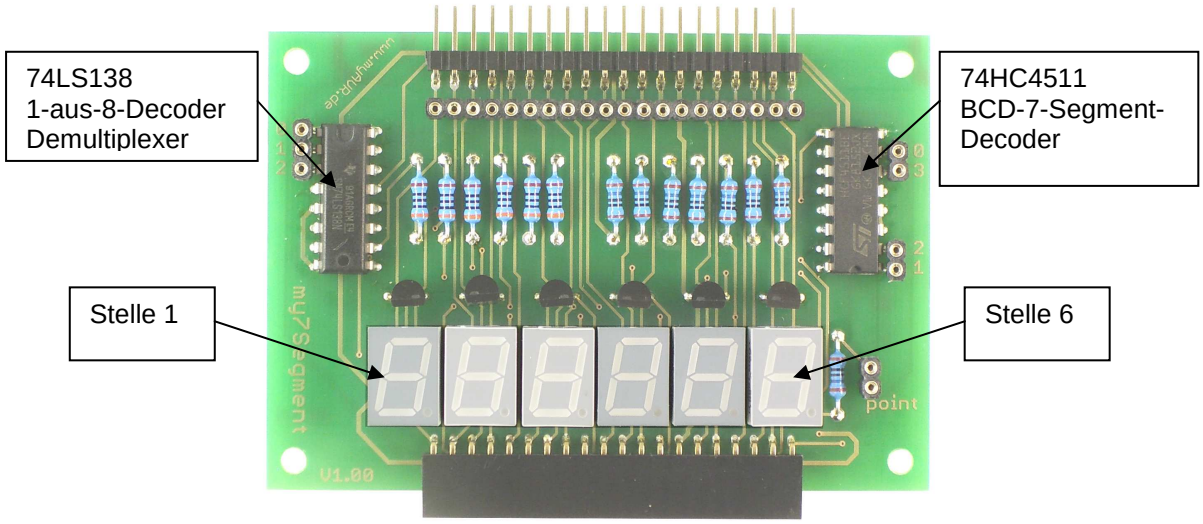


Abbildung 2.6: Anordnung der Bauelemente auf der Platine

3 Bauanleitung

Das in diesem Projekt verwendete my7-Segment-Add-On liegt Ihnen als Bausatz vor. Das Bauelementesortiment wurde gewissenhaft zusammengestellt und auf Vollzähligkeit überprüft.

3.1 Lieferumfang

Die nachfolgend aufgeführten Bauelemente sind im Bausatz enthalten. Sie benötigen noch Lötwerkzeug und geeignetes Messmittel.

Material	Typ	Stück
BCD zu 7-Segment Decoder	74HC4511	1
Demultiplexer	74LS138	1
Widerstand	METALL 500 Ohm	6
Widerstand	METALL 160 Ohm	8
Transistor	BC557	6
7-Segment-Anzeige	TDSR10	6
Sockelleiste	BL 1x20W 2,54	1
Sockelleiste	BL 1x10W 2,54	1
Buchsenleiste	1x20; 2,54	1
Stiftleiste	1x20; 2,54	1
Leiterplatte	my7-Segment Add-On	1

3.2 Vorgehensweise

Beim Bestücken wird in der Regel mit den Bauteilen begonnen, welche die kleinste Bauteilhöhe besitzen. Dann werden die Bauelemente in der Reihenfolge ihrer Bauhöhe aufgesetzt und eingelötet.
Vermeiden Sie beim Umgang mit integrierten Schaltkreisen elektrostatische Aufladungen z.B. durch die Bekleidung.

Wichtig :
Teile müssen sich ohne große Kraftanwendung einstecken lassen.
Beachten Sie bei nachfolgend aufgeführten Bauelementen die Einbaurichtung.

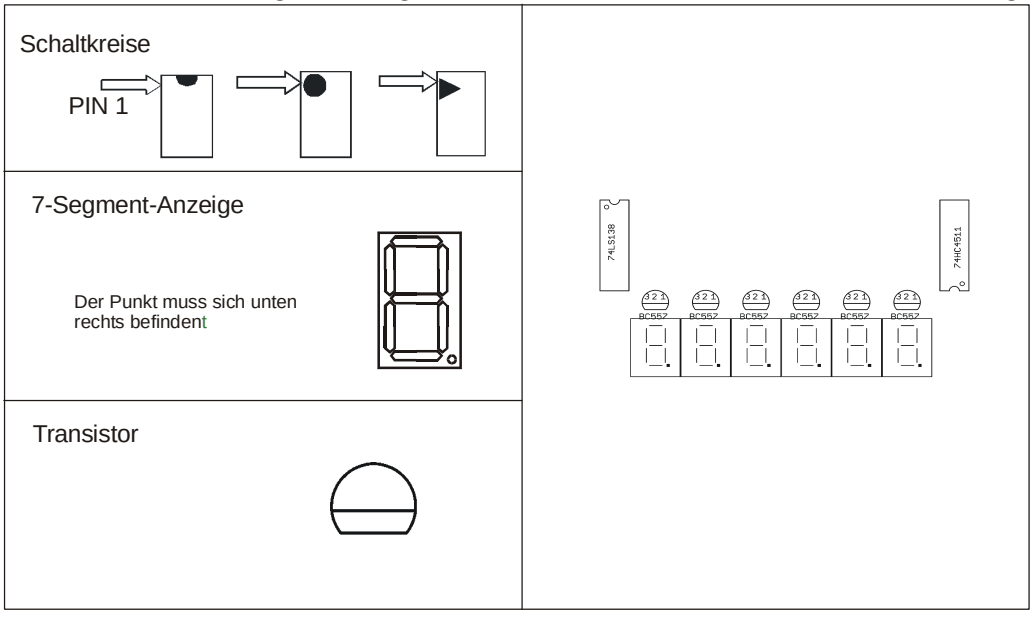


Abbildung 3.1: Bauteile, bei denen auf die Polarität zu achten ist

3.3 Bestückungsplan

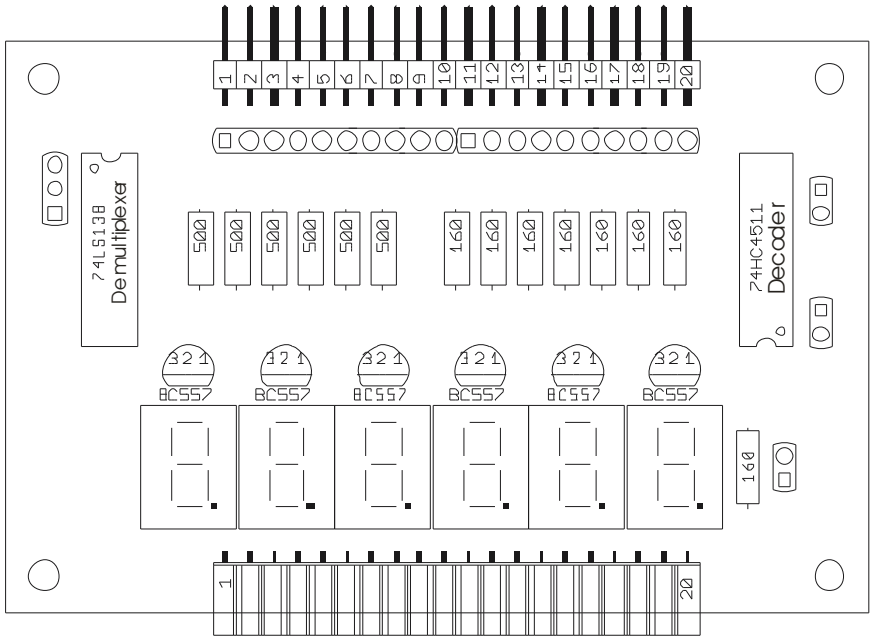


Abbildung 3.2: Bestückungsplan my7-Segment Add-On

3.4 Bestückungsbeispiel

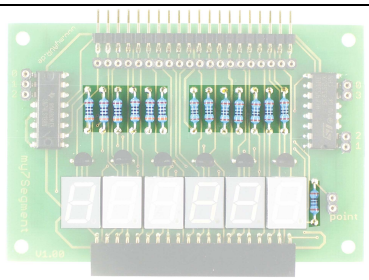
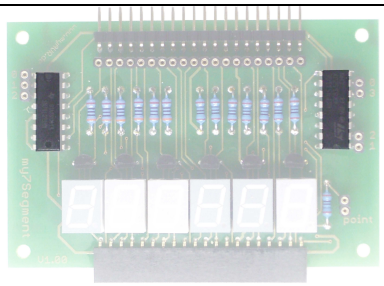
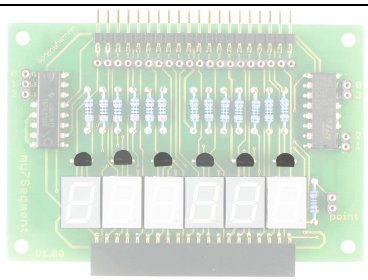
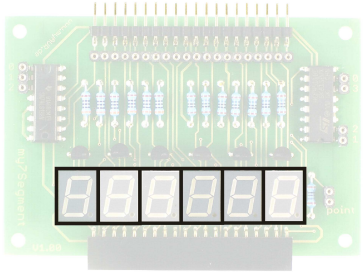
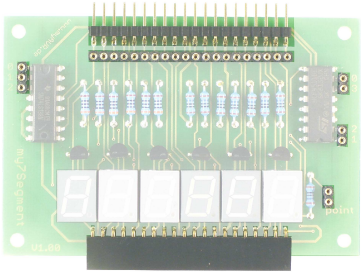
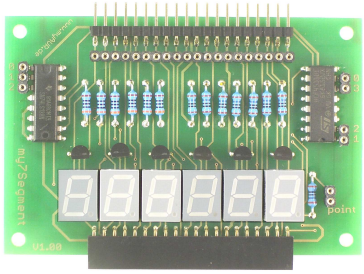
Widerstände	IC's	Transistoren
		
7-Segment-Anzeigen	Stiftleiste, Buchsenleiste, Sockelleisten	Fertig bestücktes my7-Segment Add-On
		

Abbildung 3.3: mögliche Reihenfolge der Bestückung

4 Ein Testprogramm für die 7-Segment-Anzeige

Das hier vorgestellte Testprogramm „Multiplex-Anzeige“ wählt nacheinander alle Stellen der 7-Segment-Anzeige aus. Bei jeder Stelle werden nacheinander beginnend mit dem Dezimalpunkt alle Zahlen von 0 bis 9 ausgegeben. Zwischen den einzelnen Ausgaben verstreicht eine Zeit von ca. 0,5 Sekunden. Für einen vollständigen Durchlauf des Testes wird eine Zeit von ca. 30 Sekunden benötigt. Nach einem vollständigen Durchlauf beginnt der Test von vorne.
Als erstes wird ein Struktogramm des Testprogramms entworfen, Abbildung 4.1

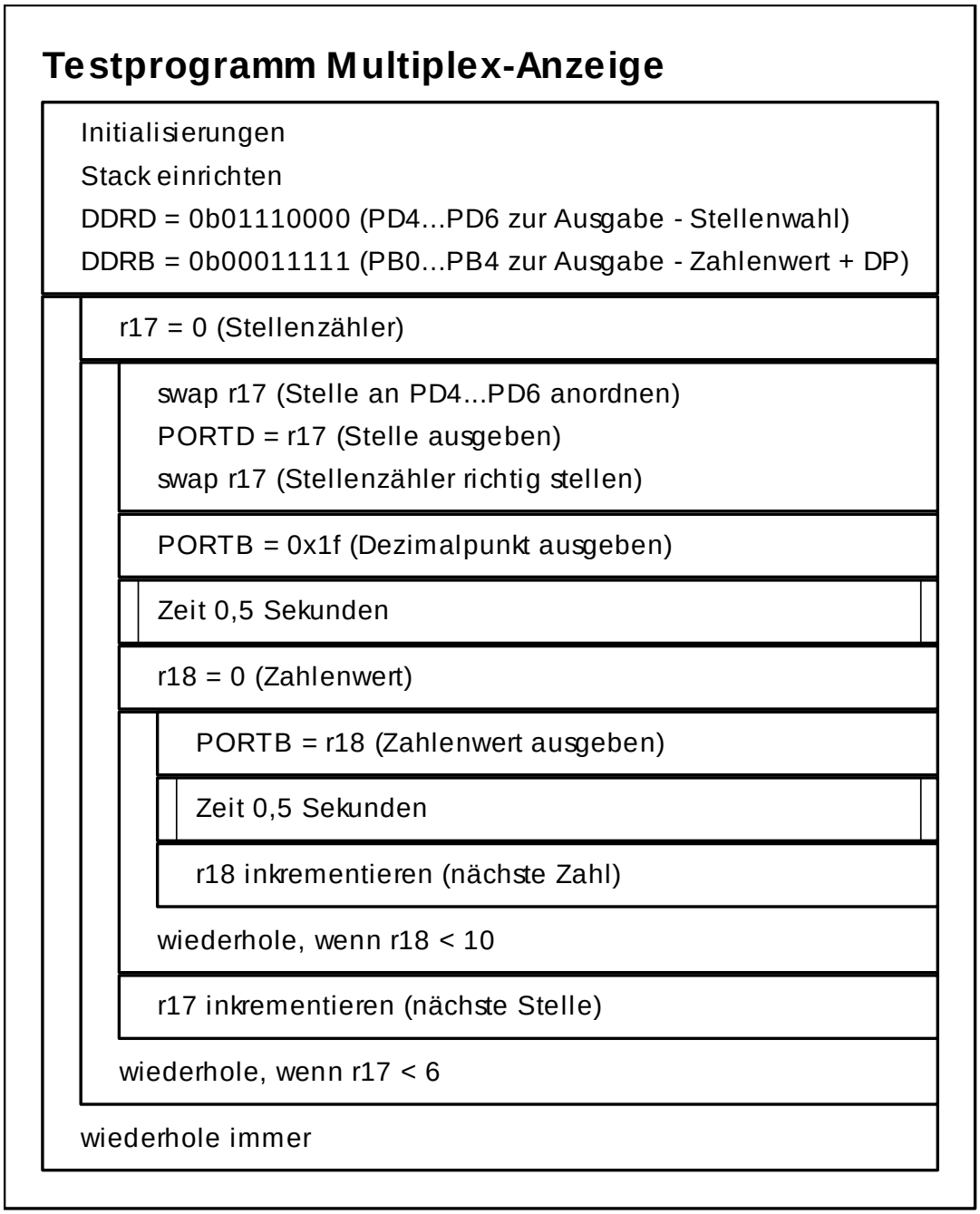


Abbildung 4.1: Struktogramm zum Testprogramm

Abbildung 4.2 zeigt die Schaltung mit dem myAVR Board MK2 USB und dem my7-Segment Add-On.

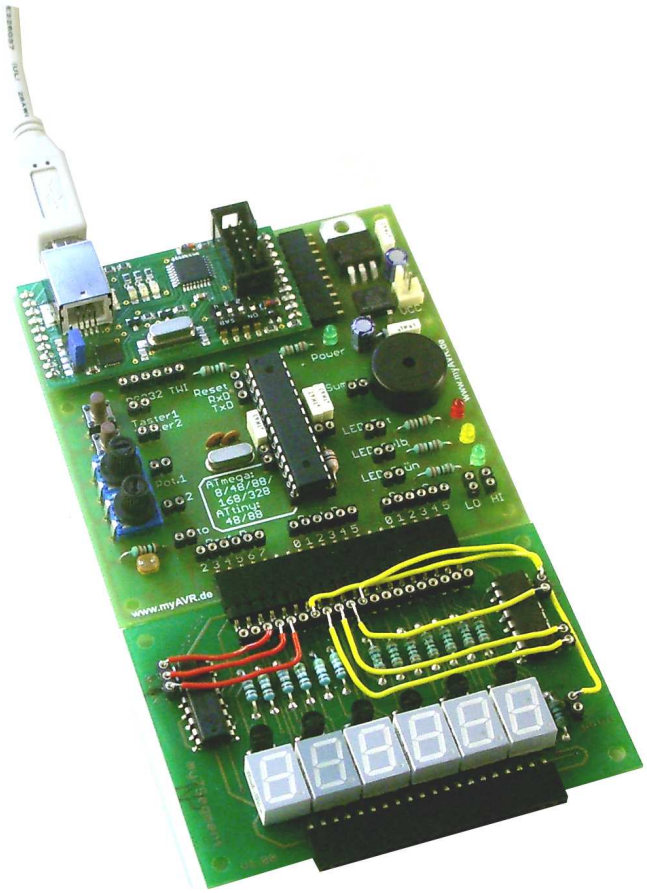


Abbildung 4.2: Schaltung für das Testprogramm

5 Projekt „Digitalvoltmeter“

Eine analoge Eingangsspannung an PC0 von 0…5 Volt soll fortlaufend gemessen und auf einer dreistelligen Multiplex-Anzeige dargestellt werden; d.h. auf dem my7-Segment Add-On werden die Stellen 4, 5 und 6 mit einer Anzeige belegt.
Bei einer Auflösung des A/D-Wandlers von 8 Bit sind 256 Schritte nötig. Die Genauigkeit der Messung bei anliegenden 5 Volt ergibt sich somit aus
5 Volt/256 Schritte ≈ 20mV/Schritt.
Die Darstellung der Spannung erfolgt mit einer Stelle vor und zwei Stellen nach dem Dezimalpunkt.
Die Referenzspannung von ca. 5 V wird extern zugeführt (VCC).

5.1 Multiplex-Anzeige im Interrupt-Betrieb

Unabhängig von den anzuzeigenden Daten muss für eine flimmerfreie Darstellung ca. alle 3 ms die Stellenanzeige mit dem zugehörigen Zahlenwert aktualisiert werden (vgl. Kapitel 2).

Für dieses und alle folgenden Projekte wird deshalb eine Software-Schnittstelle eingerichtet, so dass das Hauptprogramm, welches die numerischen Werte berechnet, und die eigentliche Anzeige, voneinander getrennt laufen. Für die Anzeige wird eine Interrupt-Routine eingerichtet, welche ca. alle 3 ms durch einen Überlauf vom Timer0 des ATmega8 ausgelöst wird. Die Interrupt-Routine benötigt als globale Informationen die anzuzeigenden Zahlenwerte und die Nummer der nächsten anzuzeigenden Stelle. Diese Informationen werden im Datensegment abgelegt.
Für die Programmierung der Interrupt-Routine ist zu beachten, dass sich die 1.Stelle der Anzeige links in Blickrichtung befindet. Werden weniger als 6 Stellen benötigt, soll die Anzeige jedoch rechtsbündig erfolgen. Deshalb wird als erstes der Stellenzähler mit 0x05 (das ist 6. Stelle) initialisiert.
Die Ablage der Werte erfolgt ebenfalls beginnend mit der 6. Stelle. Zum Einlesen der Werte wird ein Zeiger mit dem z-Register auf den Anfang der Werte eingerichtet.
Das Hauptprogramm kann die Werte beliebig initialisieren.

In der weiteren Beschreibung folgen Hinweise zu Taktfrequenz, das Struktogramm und der Quellcode in Assembler zur Multiplex-Anzeige im Interrupt-Betrieb.

5.2 Realisierung der digitalen Anzeige

Zum Verständnis dieses Kapitels werden Kenntnisse über die Funktionsweise des A/D-Wandlers beim ATmega8 vorausgesetzt. Hilfestellung dazu bietet das Datenblatt.
Nach der Wandelung des Eingangswertes an PC0 steht das Ergebnis in den I/O-Registern ADCL und ADCH (insgesamt 10 Bit). Bei linksbündiger Ausgabe (ADLAR=1) befindet sich das Ergebnis für die 8 höherwertigen Bits in ADCH. Da hier diese Genauigkeit ausreichen soll, wird nur dieses Ergebnis ausgewertet.

Nach weiteren theoretischen Erklärungen wird das vollständige Assemblerprogramm zum „Digitalvoltmeter“ gelistet.

Die Abbildung 5.1 zeigt die Schaltung für das Projekt „Digitalvoltmeter“. Dabei wird an PORTC0 (PC0) die Eingangsspannung gemessen. Dieser Wert wird auf der 7-Segment-Anzeige dargestellt. Es ist eine dreistellige Zahl, die zwei Nachkommastellen hat.

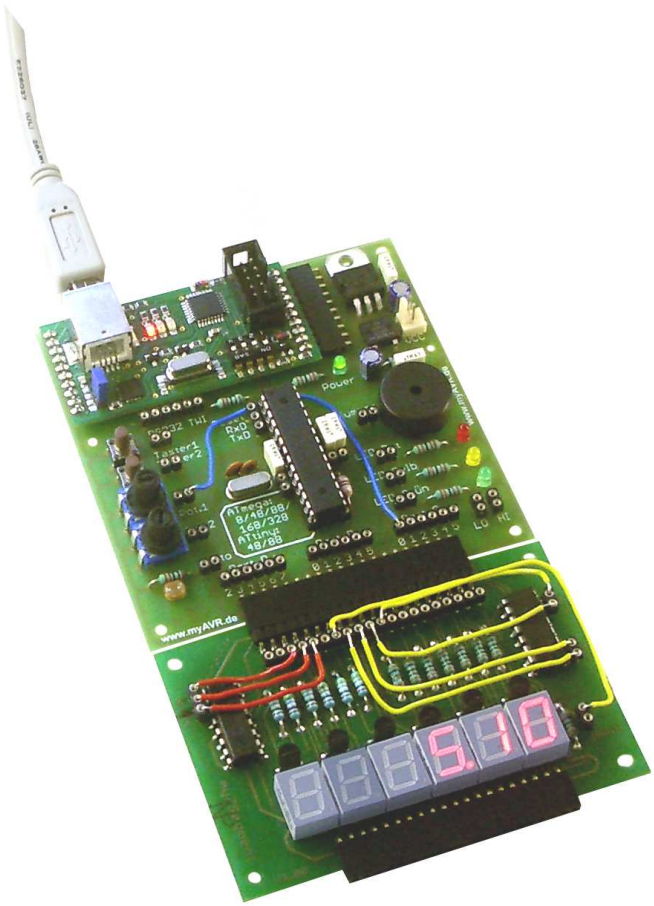


Abbildung 5.1: Schaltung und Darstellung des Projekts "Digitalvoltmeter"

6 Projekt „Stoppuhr“

Die in diesem Projekt vorgestellte Stoppuhr misst Zeiten von 0,01…9999,99 Sekunden mit einer Genauigkeit von 1/100 Sekunde. Die Zeitmessung beginnt mit Betätigung von Taster 1 (Start) und endet bei Betätigung von Taster 2 (Stopp).
Die Stoppuhr ist im Prinzip ein 6-stelliger Dezimalzähler, welcher alle 0,01 Sekunden inkrementiert wird.
Für die Zeitbasis wird der Timer1 des ATmega8 verwendet, welcher alle 0,01 s einen Overflow-Interrupt auslöst. Die erforderliche Interrupt-Routine inkrementiert die Werte im Datenspeicher. Für die Anzeige der Werte wird die im vorigen Kapitel beschriebene Interrupt-Routine für Timer0 verwendet.
Grundsätzlich gilt für jede Stelle, dass bei Erreichen eines Wertes größer als 9 diese Stelle auf 0 zurückgestellt und die nächsthöhere Stelle inkrementiert werden muss. Zu diesem Zweck wird ein globaler Zeiger y eingerichtet und mit 5 (6.Stelle) initialisiert. Es wird also zunächst die Hundertstel-Stelle inkrementiert. Wenn diese den Wert 9 überschreitet wird diese auf 0 zurückgesetzt und die Zehntel-Stelle inkrementiert. Wenn die Zehntel-Stelle 9 überschreitet wird diese zurückgesetzt und die Einer-Stelle inkrementiert, usw… . Dieser Vorgang lässt sich rekursiv in einer Schleife programmieren.

Das Hauptprogramm gestaltet sich relativ einfach. Mit Betätigung von Taster 1 werden Stelle und Zählwerte zurückgesetzt und der `tim1_ovf-Interrupt` zugelassen.
Die Zeitmessung beginnt. Wenn Taster 2 betätigt wird, wird dieser Interrupt gesperrt und die Anzeige bleibt bei dem gemessenen Wert bis zum nächsten Start stehen.
Bei der Interrupt-Routine für Timer0 muss zudem noch die Dezimalpunktanzeige für Stelle 4 eingefügt werden.

In der weiteren Beschreibung folgen Hinweise zur Taktfrequenzberechnung, zum Struktogramm und dem Quellcode in Assembler für die Stoppuhr.

Abbildung 6.1 zeigt die Schaltung für das Projekt „Stoppuhr“. Bei diesem Projekt wird die Zeit ab Betätigung des Tasters 1 bis zur Beendigung durch Drücken des Tasters 2 gezählt. Dieser Wert wird auf der 7-Segment-Anzeige dargestellt.

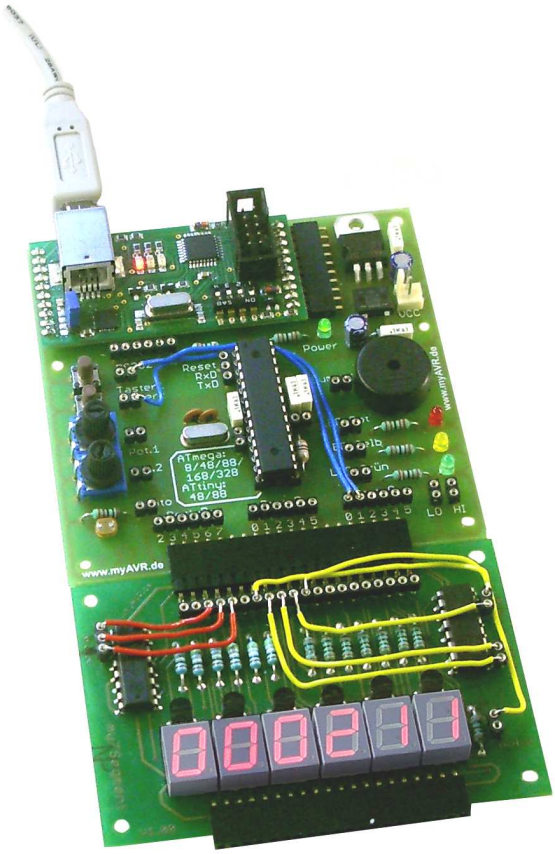


Abbildung 6.1: Schaltung und Darstellung des Projekts "Stoppuhr"

7 Projekt „Echtzeituhr“

7.1 Funktionsweise der Uhr

Bei diesem Projekt sind alle 6 verfügbaren 7-Segment-Anzeigen wertmäßig belegt.
Die Uhr mit der 6-stelligen Anzeige zeigt die Zeit im 24-Stunden-Format in der Form `HH.MM.SS`

`HH` = 00…23 Stunden

`MM` = 00…59 Minuten

SS = 00…59 Sekunden

In der 2. und in der 4. Stelle wird ein Dezimalpunkt angezeigt. Nach einem `Reset` werden alle Werte mit 0 initialisiert.
Zum Stellen der Uhr sind zwei Taster an PC0 (Taster 1) und PC1 (Taster 2) vorgesehen.
Bei Betätigung von Taster 1 wird in den Modus „Uhr stellen“ geschaltet. Der Dezimalpunkt zeigt, welche Stelle eingestellt werden kann. Mit Betätigung von Taster 2 wird die entsprechende Stelle bis zum möglichen Maximalwert inkrementiert. Mit jeder Betätigung von Taster 1 wird zur nächsten Stelle weiter geschaltet. Nach Erreichen der letzten Stelle geht die Anzeige wieder auf die fortlaufende Uhrzeit.

7.1.1 Die Programmentwicklung

Das Gesamtprogramm besteht aus drei Modulen:

1. `Timer0-Overflow-Interrupt-Routine` mit der Zeit 3 ms für die Ansteuerung der Anzeige.
2. `Timer1-Overflow-Interrupt-Routine` mit der Zeit von einer Sekunde für die Aktualisierung der Uhrzeit.
3. Das Hauptprogramm zum Stellen der Uhr.

Alle Module greifen auf eine gemeinsame Software-Schnittstelle im SRAM des ATmega8 zu, in welcher die auszugebenden Werte, die Maximalwerte für jede Stelle und die Position der Dezimalpunkte abgelegt sind.

Jedes Modul hat die Möglichkeit auf diese Werte zum Lesen oder Schreiben zuzugreifen. Die Werte werden von `tim1_ovf-Interrupt` jede Sekunde aktualisiert. Die `tim0_ovf-Interrupt-Routine` gibt diese Werte zur Anzeige.

Bei der Zeitanzeige wird der Dezimalpunkt an den Stellen 2 und 4 angezeigt. Beim Stellen der Uhr wird der Dezimalpunkt nur auf der Stelle angezeigt, welche gerade eingestellt wird. Die `tim0_ovf-Interrupt-Routine` wertet diese Information aus und zeigt entsprechend dem Dezimalpunkt.

7.2 Das Hauptprogramm zum Stellen der Uhr

Mit Betätigung von Taster 1 geht die Anzeige in den Modus „Uhr stellen“ über. Zur besseren Verständlichkeit wird die Programmierung an Hand eines etwas vereinfachten Programmablaufplanes dargestellt.
Nach Betätigung von Taster 1 wird der `tim1_ovf-Interrupt` gesperrt, so dass die Uhr für die Dauer des Stellens stehen bleibt. Der Zeiger y wird auf die erste Stelle eingerichtet, um auf die Werte der gemeinsamen Software-Schnittstelle zuzugreifen. Die Anzeige der Dezimalpunkte für die Stellen 2 und 4 wird ausgeschaltet.
Auf Grund der nachfolgenden Vergleichsoperationen muss `r18` dekrementiert werden (der tatsächliche Maximalwert ist um 1 kleiner als der gespeicherte Wert!).
Als nächstes kann entweder Taster 2 oder Taster 1 betätigt werden. Bei Betätigung von Taster 2 wird der aktuelle Stellenwert nach `r17` geladen.

- Wenn `r17` (aktueller Wert) kleiner ist als `r18` (maximal zulässiger Wert), wird der Inhalt des Stellenwertes inkrementiert, andernfalls wird er auf ‚0‘ zurückgestellt.
- Wenn die Stelle 1 (Zehner-Stunden) gestellt wird, ist u.U. eine Korrektur für Stelle 2 (Einer-Stunden) erforderlich.
- Wenn der Wert für Stelle 1 größer als 1 wird, muss der Maximalwert für Stelle 2 auf 4 gesetzt werden, ansonsten auf 10.
- Wird Stelle 1 auf 2 erhöht, wird Stelle 2 automatisch auf 0 zurückgestellt, wenn der Stellenwert von Stelle 2 größer als 3 ist.

Die Abbildung 7.1 zeigt die Schaltung des Projekts „Echtzeituhr“. Dabei wird die Zeit vom Benutzer eingegeben. Durch Betätigung des Taster 1 wird der Dezimalpunkt um jeweils eine Stelle nach links verrückt. Die Werte werden über den Taster 2 eingegeben. Nach der Einstellung der Zeit werden die Sekunden automatisch hoch gezählt.

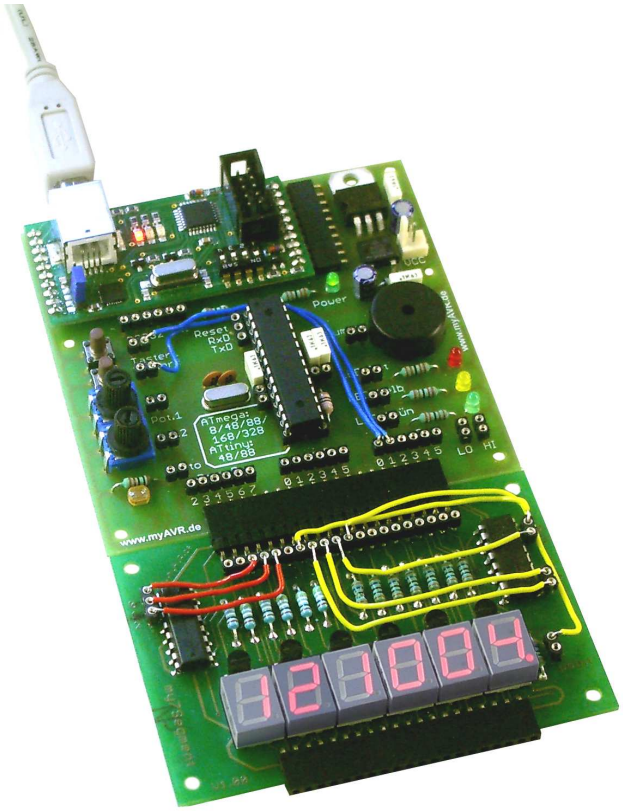


Abbildung 7.1: Schaltung und Darstellung des Projekts "Echtzeituhr"

8 Projektvorschläge

Die 6-stellige 7-Segment-Multiplexanzeige lässt dem Leser viel Raum für eigene Entwicklungen. Die hier vorgestellten Beispiele mögen als Anregung dienen.

8.1 Vorschlag 1 – „4-stelliges Digitalvoltmeter“

Das hier in Kapitel 5 vorgestellte Digitalvoltmeter misst mit einer Genauigkeit von 20 mV. Der ATmega8 besitzt tatsächlich einen A/D-Wandler mit einer Auflösung von 10 Bit. Bei einer Referenzspannung von ca. 5 V lässt sich damit eine Genauigkeit von 5 mV erreichen. Zur Anzeige muss die Stellenzahl auf 4 erhöht werden.
Das Programm ist entsprechend zu ergänzen.

8.2 Vorschlag 2 – „Temperaturmessung“

Mit einem einfachen Temperatursensor (z.B. KTY81-1 von Philips Semiconductors) mit nachgeschaltetem Elektrometerverstärker lässt sich das Digitalvoltmeter zur Temperaturmessung verwenden. Eine Ergänzung wäre eine Temperaturüberwachung, welche z.B. bei einer beliebig einstellbaren unteren Temperatur eine Heizung automatisch einschaltet.

8.3 Vorschlag 3 – „Frequenzmessung“

An dem Eingang für den externen Interrupt INT0 (PD2) wird ein Rechtecksignal eingespeist. Für die Dauer von 1 Sekunde werden die Impulse gezählt und danach auf der Anzeige dargestellt. Die Anzeige dürfte für die Frequenzmessung im Audiobereich ausreichend sein. In Verbindung mit einem geeigneten Mikrophonverstärker könnte das Gerät beispielsweise für die Stimmung der Saiten einer Gitarre verwendet werden.

8.4 Vorschlag 4 – „Uhr mit Datumsanzeige und Weckfunktion“

Mit einer Echtzeituhr lässt sich eventuell mit zusätzlichen Tastern um sinnvolle Funktionen erweitern. Beispielsweise könnte die Anzeige zwischen der Anzeige der Uhrzeit und dem aktuellen Datum in regelmäßigen Zeitabständen wechseln. Das Programm wäre um die Möglichkeit, das Datum einzustellen, zu erweitern.
Eine andere Erweiterung ist die Möglichkeit, die Uhr als Wecker zu verwenden. Bei einer beliebig einstellbaren Zeit, soll ein Summer ertönen.

9 Quellenverzeichnis

Datenblatt 74HC/HCT4511 BCD to 7-segment latch/decoder/driver
http://www.nxp.com/documents/data_sheet/74HC_HCT4511_CNV.pdf

Datenblatt DM74LS138 • DM74LS139 Decoder/Demultiplexer
http://www.futurlec.com/74LS/74LS138.shtml

http://www.datasheetcatalog.org/datasheet/motorola/SN74LS138N.pdf

